Order Without Consequence:

A Permutation-Theoretic Investigation of Serial Position Effects in Sunday-Morning Hackathon Adjudication (hackaTUM 2022–2025)

The PitchPerfect Consortium*

29 November 2026

Working Paper No. 1 (and, God willing, the last)

Abstract

We investigate whether the position of a team within a Sunday-morning pitch session at the hackaTUM hackathon of the Technical University of Munich affects the probability that the team is awarded a placement. Official pitch schedules are available for 31 sessions comprising 582 pitch slots. Devpost winner lists are matched to those slots, yielding 62 winners, exactly two per session. Position is normalized as $x = (j-1)/(n_s-1)$, so that $x = 0$ denotes the first pitch and $x = 1$ the last. Under the null hypothesis H_0 that winning slots form a uniformly random two-element subset within each session, the expected mean winner position is 0.5001. The observed mean is 0.4994, with a 95% bootstrap confidence interval of [0.426, 0.576] obtained by resampling whole sessions. The two-sided permutation p-value, computed from 200,000 within-session reshuffles of winner labels, is $p = 0.985$. The minimum detectable effect at 80% power is ± 0.106. A two one-sided tests procedure with equivalence bounds of ± 0.10 around one half rejects non-equivalence ($p = 0.0043$), so that any shift in mean winner position exceeding 0.10 is rejected. Secondary tests, including tercile counts, edge wins and a fixed-effects conditional logit, are all null after Holm correction.The single significant result concerns the schedule itself rather than the judging. Slot order is correlated with Devpost submission identifier (median Spearman $\rho = 0.24$, Stouffer $z = 7.04$, Holm-adjusted $p = 1.7 \times 10^{-11}$), which is consistent with a scheduler that loosely follows submission creation order. Among eleven ranked first-and-second-place pairs, first-place winners pitched later in four cases (sign test $p = 0.549$). Measured in absolute clock time rather than queue position, the average winner pitched at 11:02 against an expected 11:00 ($p = 0.589$); an apparent decline in raw win rate over the morning is fully explained by small rooms finishing early. We conclude that, within the resolution of the available data, the order of pitches has no detectable effect on placement. Because no effect is detected, we do not recommend any change to the scheduling of pitches, although we acknowledge that the sample is small and that absence of evidence is not evidence of absence.

Keywords: serial position; Sunday-morning adjudication; permutation inference; equivalence testing; pitch slot; hackathon schedule; order without consequence; afternoon jury; Devpost-adjacent heuristics; the 10:15 problem
JEL codes: C12, C14, D91, O31

*Institute for Advanced Slot Studies (unaffiliated). Companion website: https://pitchperfect.patientzero.cloud. Correspondence regarding slot allocation should be directed to the organizers, not to us.

Contents

1 Introduction **7**

1.1 The research question . . . 8
1.2 Why this matters . . . 9
1.3 Contributions . . . 10
1.4 Preview of results . . . 12
1.5 Roadmap . . . 13

2 Literature Review I: A Brief History of Remembering Things in Order **14**

2.1 Ebbinghaus and the Dawn of Serial Learning . . . 14
2.2 The Serial Position Curve: Primacy and Recency . . . 15
2.3 Primacy in Impression Formation . . . 15
2.4 The Belief-Adjustment Model and the Length of Lists . . . 16
2.5 Anchoring and Comparison Processes . . . 17
2.6 Ballot-Order Effects: The Closest Analogue to Position in a List . . . 18
2.7 Choice Order . . . 19
2.8 Synthesis and Limits of the Literature . . . 19
2.9 Implications for the Sunday-Morning Pitch . . . 20

3 Literature Review II: Juries, Judges, and the Hungry Mind **21**

3.1 Serial position in performance competitions . . . 23
3.2 Judges, breaks and fatigue . . . 24
3.3 Hackathons as an object of study . . . 25
3.4 The replication crisis and why null results deserve publication . . . 26
3.5 Positioning the present study . . . 27

4 Institutional Background: The hackaTUM Sunday Morning **29**

4.1 hackaTUM in Brief . . . 29
4.2 The Partner-Challenge Model . . . 30
4.3 A Phenomenology of the Five-Minute Pitch . . . 31
4.4 The Scheduling Problem . . . 32
4.5 The Award Structure . . . 33
4.6 What the Judges Evaluate . . . 34

5 Data **35**

5.1 Sources . . . 35
5.2 Extraction . . . 36
5.3 Entity resolution . . . 38
5.4 Exclusions . . . 39
5.5 The unidrei incident . . . 39
5.6 The final analytic sample . . . 40
5.7 Variables . . . 41
5.8 Limitations . . . 42

6 Methodology **43**
6.1 Notation and the sharp null . . . 44
6.2 The primary statistic . . . 44
6.3 Monte Carlo permutation p-values . . . 46
6.4 Bootstrap confidence interval . . . 47
6.5 Power and the minimum detectable effect . . . 48
6.6 Equivalence testing . . . 48
6.7 Secondary statistics . . . 49
6.8 Conditional logit with fixed effects . . . 50
6.9 Is the order itself random? . . . 51
6.10 Multiplicity . . . 52
6.11 Pre-declaration and the garden of forking paths . . . 53
6.12 Software and reproducibility . . . 54

7 Results **54**
7.1 Descriptive overview . . . 54
7.2 Primary result . . . 56
7.3 Equivalence . . . 58
7.4 Secondary tests . . . 58
7.5 Testing the four hypotheses . . . 59
7.6 First versus second place . . . 60
7.7 The overall winners . . . 61
7.8 The one significant finding . . . 61

8 Results II: The Tyranny of the Clock **62**
8.1 Definitions . . . 62
8.2 The naive picture, and why it is wrong . . . 63
8.3 Tests . . . 63
8.4 Interpretation . . . 65

9 Robustness, Sensitivity, and Existential Doubt **65**
9.1 Subset analyses . . . 66
9.2 The unidrei sensitivity analysis . . . 69
9.3 Non-random scheduling as a threat to validity . . . 70
9.4 The negative-control comparison that could not be run . . . 71
9.5 Planned versus realised order . . . 71
9.6 Alternative position codings . . . 72
9.7 Interpretive humility . . . 72
9.8 Threats to validity . . . 73

10 Discussion **74**
10.1 Summary of findings . . . 75
10.2 Why might hackathon judging be immune to order effects? . . . 75
10.3 Relation to the literature . . . 77
10.4 Practical recommendations for hackathon teams . . . 78
10.5 Recommendations for organizers . . . 79
10.6 A proposed pre-registered Randomized Controlled Hackathon . . . 80

10.7 Future work 81
10.8 Limitations 81

11 Conclusion **82**

A Complete Pitch Schedules **91**

B Exact Null Moments by Session **107**

C Session Atlas **108**

D Source Code **115**

List of Figures

1 Every analysed session, one row each. Gray dots are pitch slots; blue dots are challenge winners; orange rings mark confirmed first places (all 2022 challenges, SAP 2025). . . . 64
2 Permutation null distribution of the mean winner position (200,000 within-session reshuffles) with the observed value $T = 0.499$ and its 95% bootstrap interval. . . . 65
3 Empirical cumulative distribution of the 60 winner positions against the uniform distribution implied by H_0. . . . 66
4 Observed versus expected winners by third of the schedule. Expected counts are computed session by session. . . . 67
5 Normalized pitch position against the rank of each team's Devpost submission ID within its session. The schedule is not random; it is, however, irrelevant. . . . 67
6 Win rate by clock time. Blue: observed share of pitches that won. Gray: the share expected if clock time were irrelevant, given which rooms were still running. The raw decline is a room-size artefact. . . . 68
7 Robustness of the primary statistic across subsets. . . . 69
8 Session atlas, part 1 of 6. . . . 109
9 Session atlas, part 2 of 6. . . . 110
10 Session atlas, part 3 of 6. . . . 111
11 Session atlas, part 4 of 6. . . . 112
12 Session atlas, part 5 of 6. . . . 113
13 Session atlas, part 6 of 6. . . . 114

List of Tables

1 Primary results in brief. Equivalence bounds are ± 0.10 around one half, and the minimum detectable effect is reported at 80% power and $\alpha = .05$. . . . 12
2 Mechanisms reviewed in this section, the literature that motivates them, and the direction of slot-position effect they predict for a Sunday-morning pitch session. . . . 22
3 Selected order-effect settings and their typical direction. Only rows whose direction the authors are confident of are included. Effect sizes are omitted by design. . . . 25
4 Partner challenges by year, with the number of pitch slots in each session. A dash indicates that the partner had no session in that year. The 2022 Optiver session (43 teams) is excluded from the analysis because all teams were scheduled in a single 30-minute block, and it is marked here for completeness. The 2024 and 2025 Optiver sessions have no schedule PDF; these are marked "no PDF" and are likewise excluded. The partner listed as SIXT in 2025 is merged with Sixt in 2022 and is treated as one partner. . . . 30
5 Overall winners in the years for which the data were collected, with the slot each occupied in its partner session and the corresponding normalised position $x = (j - 1)/(n_s - 1)$. The sample is three cases. The table is descriptive and is not used in any test. . . . 33

6 Composition of the analytic sample by year. Sessions are partner pitch sessions; slots are scheduled pitches retained in the analysis; winners are the two per session, so the winner count is twice the session count in every year. The session-size column gives the smallest and largest session in each year. 40
7 Codebook for the slot table, one row per scheduled pitch. Variables are listed in the order of the file, with the exception that `pos01` is the analysis variable and is listed last. 41
8 The secondary family of nine tests, with raw and Holm-adjusted p-values. The Stouffer test is the only result with a Holm-adjusted p-value below 0.05. 53
9 Winners of the 2022 and first half of the 2024 sessions, with slot j, session size n, normalized position x and published place. A dash marks a session with no ranking source. 56
10 Winners of the second half of the 2024 sessions and of the 2025 sessions. Columns as in Table 9. 57
11 Secondary tests. Expected values are under the within-session permutation null. Holm p-values adjust across the family. The quadratic logit term is reported without a Holm-adjusted value. 58
12 Verdicts on the four hypotheses. The numbering follows the text: P0 primacy, P1 recency, P2 U-shape and P3 null. The key evidence column cites the tests in Table 11. 60
13 The eleven ranked pairs. A pair counts toward the sign test when the first-place team pitched later than the second-place team. 60
14 Winners by half-hour window of scheduled start time. Expected counts are computed within sessions under H_0. 63
15 Clock-time tests (exploratory; separate Holm family of five tests). All Holm-adjusted p-values equal 1. 64
16 Robustness of the primary statistic (mean normalised winner position) across seven subsets. The 95% intervals are bootstrap intervals from resampling whole sessions, and p values are two-sided permutation values. Expected values under H_0 are shown where they differ from 0.5. 66
17 Headline results, for readers who skipped the summary paragraph. 75
18 Speculative explanations for the absence of an order effect, with the evidence each would require. None has been tested. 77

1 Introduction

> If you do not know where you stand in the queue, you are, in a sense, everywhere at once.
>
> Anonymous hackathon participant, 4:12 a.m.

At 10:15 on a Sunday morning in November, a team that has been awake since Friday evening stands in a meeting room that smells faintly of cold coffee and whiteboard marker, and waits to be told when it will speak. The room belongs to one of the industry partners whose challenge the team has chosen. The team has spent the preceding thirty-six hours building something, and it will now be given four or five minutes to explain that something to a person who has, by this point in the morning, heard a considerable number of explanations already. The question that occupies the team, quietly, in the minutes before its slot is not whether its pitch is good. The team has rehearsed the pitch and is reasonably confident in it. The question is whether the pitch is early, or late, or somewhere in the uncomfortable middle, and whether that placement will matter.

This question is not idle. The pitch schedule for each partner session is published in advance as a document, usually a PDF, and any team that has submitted to a challenge can look up its own position in the order. Teams in the same session can look up one another's positions. The result is a small, public, and entirely unnecessary form of arithmetic. A team learns that it pitches in position nineteen of twenty, and immediately begins to construct theories about what this means. A team learns that it pitches first, and begins to construct different theories about what that means. Over the years these theories have hardened into a recognisable folk canon, which circulates in the hallways between sessions and is repeated with a confidence that its evidential basis does not obviously support.

The first folk theory holds that judges are fresh early. On this view the first few pitches receive the full attention of a panel that has not yet been worn down by the cumulative effort of listening, and the later pitches are received by a panel that has begun, however politely, to stop listening. The second folk theory is the mirror image and holds that the last pitch is remembered. On this view the final presentation of the session is the one that is recalled when the panel retires to deliberate, and the earlier presentations are overwritten by whatever came most recently to mind. The third folk theory is the most specific and the least examined, and it holds that one should never pitch right before lunch. Nobody in the hallway has ever been able to say precisely why this should be so, but the belief is held with a seriousness that suggests it has been inherited rather than derived.

None of these theories is new in kind. Each is a particular instance of a general and well-studied class of phenomena in which the sequence in which items are evaluated appears to influence the evaluation of those items. The literature on this class is large and considerably older than the hackathon. Serial position effects were documented in the recall of word lists more than a century ago (Ebbinghaus, 1885), and their characteristic shape, with elevated recall at the beginning and the end of a list, was described systematically in the early 1960s (Murdock, 1962). Primacy, the disproportionate influence of early information on the formation of impressions, was reported in the same spirit by Asch (1946) and given a formal treatment by Anderson (1965). A belief-adjustment account that unifies several of these findings under a single model of order effects was offered by Hogarth and Einhorn (1992). Ballot-order effects, in which the position of a candidate on a list appears to matter to voters, have been studied by

Miller and Krosnick (1998) and Ho and Imai (2008).

Judged performance is the setting in which order effects are most directly relevant to the present paper, and it is here that the literature becomes both more numerous and less settled. Evaluations of performers by juries have been examined in Eurovision and in figure skating, where later performers have been found to receive higher scores in the scoring systems studied (Bruine de Bruin, 2005, 2006). A similar pattern has been reported for television singing competitions (Page and Page, 2010), for the Queen Elisabeth music competition, where the order and the day of performance have been argued to affect rankings (Glejser and Heyndels, 2001; Flôres and Ginsburgh, 1996), and for Eurovision once again, where a later-performer advantage was reported by Haan et al. (2005). Order effects in song contests have been analysed further by Antipov and Pokryshevskaya (2017). Wine tasting offers a different and somewhat more intimate setting, in which the first and last of a series of options appear to be preferred (Mantonakis et al., 2009).

The most influential and most contested study in this area concerns parole. Danziger et al. (2011) reported that the decisions of parole judges appear to decline as the day progresses and to recover after a break, and the finding was widely discussed as an illustration of decision fatigue. Weinshall-Margel and Shapard (2011) offered a pointed critique of that analysis, arguing that the order in which cases are heard is not random and that the apparent pattern may reflect the composition of the case queue rather than the state of the judges. We bring this debate into the present paper in a deliberately modest form. The first pitch after a break is one of our secondary tests, and we report it in full, although we are not in a position to say anything about parole.

A related and contested body of work concerns the depletion of self-control. The ego depletion hypothesis proposed by Baumeister et al. (1998) suggested that exertion of self-control draws on a limited resource, with consequences for subsequent tasks. Its status is now uncertain, since a multi-laboratory replication effort failed to reproduce the central effect (Hagger et al., 2016). The present data bear on it not at all, but readers who have attended a thirty-six-hour hackathon will want to know that we considered it.

1.1 The research question

We may now state the question that this paper addresses in formal terms. Let $s \in \{1, \dots, S\}$ index the partner sessions in the sample, so that $S = 31$ after the exclusions described in Section 5. Session s has n_s pitch slots, indexed by $j \in \{1, \dots, n_s\}$. Following common practice in the order-effects literature, we normalize position so that the first slot is comparable with the last, and define

$$x_{sj} = \frac{j - 1}{n_s - 1} \in [0, 1], \tag{1}$$

so that $x_{sj} = 0$ for the first pitch of a session and $x_{sj} = 1$ for the last. Sessions range in size from seven to forty-five pitches, and the normalization allows sessions of different sizes to be pooled without assuming that a given absolute slot means the same thing in a short session and a long one.

Let W_s denote the set of winner slots in session s. Each partner selects a first-place and a second-place team, so that $|W_s| = 2$ for every session in the sample. The null hypothesis of no

serial position effect is then

$$H_0 : \text{for every session } s,\ W_s \text{ is a uniformly random 2-subset of the } n_s \text{ slots, independently across sessions.} \tag{2}$$

The alternative hypothesis is the negation of H_0, namely that the winner slots are systematically located with respect to position in at least one session in a way that departs from this uniform reference. Because the alternative is not a single parametric model, we do not attempt to estimate its form directly. Instead we pre-declare a single primary statistic, the mean normalized position of all winners,

$$T = \frac{1}{62} \sum_{s=1}^{31} \sum_{j \in W_s} x_{sj}, \tag{3}$$

which under H_0 has an expectation of approximately one half. The primary test asks whether T is compatible with its null distribution, and the equivalence test asks whether any departure of T from one half is small enough to be rejected at a stated margin. Both are described in Section 6.[1]

The test is conditional on the schedule. We do not model which teams apply or how good their projects are. The null hypothesis concerns the schedule and the winners, not the judges' minds. A non-uniform distribution of winner positions would be compatible with serial position effects in judging, but also with other mechanisms discussed in Section 10. A uniform distribution would not prove that no judge was ever influenced by order, only that any such influence does not translate into a detectable pattern here.

1.2 Why this matters

The question of whether the order of presentations affects their outcomes has consequences that extend well beyond the hallways of a single university hackathon. Admissions panels, grant committees, examiners, auditions, funding pitches and product reviews are all organised as sequences of presentations, and each is evaluated by a person or a panel who has to hold the earlier items in mind while considering the later ones. If serial position effects were large and systematic in these settings, the consequences for fairness would be substantial, and the case for scheduling reform would be strong. The present study cannot resolve that larger question, and we do not pretend that it can.

What the present study can do is to supply a clean, small, and entirely public test case in which the stakes are real but bounded. Each winner receives a prize, a position in a list that is published online, and, in the case of the afternoon jury, a chance at an overall award. None of this is trivial to a team that has spent a weekend on its project. Nor is any of it large enough to justify the kind of vigilance that a parole board or an admissions committee would be expected to exercise. The hackathon sits, in this respect, at a useful point on the scale of consequence. It is important enough to be worth measuring and unimportant enough that the measurement can be carried out without anyone's liberty or livelihood depending on the result.

There is also a practical stake for organisers, whose schedules are assembled by hand on a Sunday morning. The present results suggest that the scheduler has managed clashes well, and

[1] The choice of T was made before the data were examined. This is the only aspect of the design that the authors would describe as rigorous without qualification. A reader who wishes to repeat the analysis with a different statistic is free to do so, provided that they also repeat the pre-declaration, which we recommend they conduct in a hotel lobby at an hour of their own choosing.

that the more pressing question is whether the schedule follows the order of submission. That is a less glamorous finding than a serial position effect, but it is the one the data support.[2]

Finally, the question matters to the people who write pitches. A pitch is a short communication under strong time pressure, and its authors invest considerable effort in it. On the evidence presented here, that effort need not depend on the slot. Teams that pitch at 10:15 may therefore rest easy. Teams that pitch at 12:40, a hypothetical team that appears only for illustration, may do so as well.

1.3 Contributions

The contributions of the present paper are, we believe, numerous. We list them here in the order in which they were achieved, with the caveat that the later items are considerably less consequential than the earlier ones and that the authors have not been able to determine a principled ordering among the final few.

(1) We assemble the official pitch schedule PDFs for 31 hackaTUM partner sessions, comprising ten sessions in 2022, ten in 2024 and eleven in 2025.

(2) We parse those PDFs, a task that, in the general case, requires reading the relevant table more than once.

(3) We define a normalized position x_{sj} that equals zero for the first pitch of a session and one for the last, and we justify this normalization on the grounds that it allows sessions of different sizes to be pooled.

(4) We follow each Devpost submission URL printed in a schedule to its canonical project page, resolving HTTP redirects along the way.

(5) We match 55 of the 62 winners to schedule slots by means of this canonical URL, which we regard as the most reliable identifier available.

(6) We match the remaining seven winners by name or project slug, after Devpost returned HTTP 403 responses during resolution, and we verify each of the six matches by hand.

(7) We exclude the 2022 Optiver session, in which all 43 teams were scheduled within a single thirty-minute block, on the grounds that such a session contains no order to study.

(8) We record that no schedule PDF could be located for the 2024 Optiver session or the 2025 Optiver session, and we exclude both.

(9) We name the unidrei incident, in which the 2022 Solana schedule labels two different submissions with the same name, and we show that the one in slot one links to the project S-Moody, which is a winner. We trust the URL. We further show that dropping the session changes nothing.

(10) We audit the schedules for clashes. Ninety-six submissions pitched in two or more sessions in the same year. None had two pitches at the same time, the minimum gap between

[2] Readers from outside the hackathon community should be aware that the authors have been asked, on more than one occasion, whether the afternoon jury is also subject to serial position effects. We have no data on the afternoon jury and decline to speculate, although we note that the jury is composed of professors, who are, by the standards of this literature, a population that has not previously been studied in a hurry.

consecutive pitches of the same submission was four minutes, and the median gap was fifty minutes.

(11) We run a permutation test with 200,000 within-session reshuffles of winner labels, and we fix the random seed at 2025 so that the reported p-value can be reproduced exactly by anyone who wishes to do so.

(12) We compute a 95% bootstrap confidence interval by resampling whole sessions, using 20,000 resamples, which we regard as a more honest unit of resampling than the individual winner.

(13) We perform a two one-sided tests procedure against equivalence bounds of ± 0.10, and we report the minimum detectable effect at 80% power separately from the equivalence margin, so that the reader may distinguish between what the design could have detected and what the data actually rule out.

(14) We fit a fixed-effects conditional logit of winning on position, with sessions as strata, and we fit a second specification that includes a quadratic term in position, following McFadden (1974) and Chamberlain (1980).

(15) We apply the Holm procedure (Holm, 1979) across the full family of secondary tests, which we enumerate in Section 6 so that no reader need wonder which tests were counted.

(16) We compute the Spearman correlation between slot and Devpost submission identifier within each session, combine the correlations across sessions by Fisher's z transformation (Fisher, 1915) and Stouffer's method (Stouffer et al., 1949), and report the result as the only significant finding of the paper.

(17) We test whether the winners tend to have earlier or later submission identifiers than the field, as a check on whether any apparent order effect is in fact a submission-timing confound.

(18) We compare the first-place and second-place finishers within each of eleven ranked pairs by means of a sign test, which is the simplest test available and which we considered it important to report in its simplest form.

(19) We establish that the 2022 first-place teams, all ten of them, pitched between normalized positions 0.17 and 0.70, and that none of them pitched first or last. We report this observation in full and draw no conclusion from it beyond the observation itself.

(20) We record that the largest session in the sample, the 2025 City of Munich session, contained 45 pitches in four-minute slots running from 10:15 to 13:11, and we consider this to be the most demanding single morning in the dataset, although we have no direct measure of the judges' stamina.

(21) We write all analysis code in Python, using NumPy, SciPy, statsmodels and Matplotlib, and we make that code available in the appendix, so that the results can be reproduced line by line by any reader with the patience to do so.

(22) We report that the second-place finishers in 2022 Rohde & Schwarz and 2022 Solana pitched first in their respective sessions, and that the very first pitch of the 2025 SAP session won first place. We present these as curiosities, which they are.

The contributions above are, in the main, contributions of method and bookkeeping rather than of substantive discovery. The substantive finding is a null, which we regard as useful precisely because it is well documented, so that another team, in another year, may add their own sessions and see whether the picture changes.[3]

1.4 Preview of results

The primary result is that the mean normalized position of the 62 winners is 0.4994, against an expected value under H_0 of 0.5001. The 95% bootstrap confidence interval is [0.426, 0.576]. The permutation null has a standard deviation of 0.0378, and the two-sided permutation p-value is 0.985. On the usual conventions, the primary test therefore fails to reject H_0 by a very wide margin, and the observed mean is almost exactly the value that a uniform schedule of winners would produce. Table 1 summarizes the primary results together with the equivalence analysis.

Table 1: Primary results in brief. Equivalence bounds are ± 0.10 around one half, and the minimum detectable effect is reported at 80% power and $\alpha = .05$.

Quantity	Value
Mean winner position	0.4994
Expected under H_0	0.5001
95% bootstrap CI	[0.426, 0.576]
Null standard deviation	0.0378
Permutation p (two-sided)	0.985
Minimum detectable effect (80% power)	± 0.106
TOST equivalence p (bounds ± 0.10)	0.0043

The equivalence result deserves separate attention, because it reverses the usual logic of significance testing. A non-significant result is commonly interpreted as absence of evidence, which is a weak statement. The two one-sided tests procedure, introduced in the pharmacological context by Schuirmann (1987), instead asks whether the observed effect is small enough to be rejected in favour of a stated range of effects that the researcher considers practically important. Here the procedure yields $p = 0.0043$, so that any shift in mean winner position exceeding 0.10 is rejected at the five per cent level. The margin of ± 0.10 corresponds to shifting the average winner from the centre of the schedule to a point two-fifths of the way along it, which is a shift that most readers would consider practically important. The minimum detectable effect of ± 0.106 is a separate and more modest quantity, and it describes what the design could have detected with probability 0.80, not what the data rule out. The two figures are close, which is a coincidence of the present sample size, and we would not wish the reader to conclude that they are the same thing.[4]

The secondary results are uniformly null after correction. The late half of each session contains 33 of the 60 winners for which the middle slot is unambiguous, against 30.01 expected, with raw $p = 0.503$. The early, middle and late terciles contain 21, 25 and 16 winners, against expected counts of 20.42, 21.16 and 20.42, with a chi-square-type statistic of 1.673 and $p = 0.397$. The first slot won in four cases, against 4.304 expected, and the last slot won in four cases,

[3] Every session is documented, and thirty sessions is a small number. Both senses are intended.

[4] The authors are aware that the phrase "minimum detectable effect" is sometimes read as a statement about the power of a study after the fact. It is not. It is a design quantity, computed before any outcome is known, and it describes the sensitivity of the design. For a discussion of why post hoc power is not a useful quantity, see Hoenig and Heisey (2001).

against 4.306 expected. Eight winners occupied either an edge slot, against 8.620 expected. The linear conditional logit gives a coefficient of $\beta = -0.007$ (standard error 0.427, $p = 0.987$), which corresponds to an odds ratio of 0.993 for the last slot relative to the first, with a 95% confidence interval of 0.43 to 2.29. The quadratic specification gives a slight inverted-U shape, with $\beta_{\text{quad}} = -0.774$ and $p = 0.625$, which does not approach significance. Winners' submission-identifier ranks have a mean of 0.534 against an expectation of 0.500, with raw $p = 0.367$.

The one result that is significant is the correlation between slot and submission identifier. The median Spearman correlation across sessions is $\rho = 0.24$, the Stouffer combination gives $z = 7.04$, and the raw and Holm-adjusted p-values are 1.9×10^{-12} and 1.7×10^{-11} respectively. The schedule therefore follows, loosely, the order in which the Devpost submissions were created. The per-session correlation ranges from -0.32 in 2022 Solana to 1.00 in 2022 Wildtrack. We interpret this finding as a property of the scheduler rather than of the judges, because it concerns the order in which submissions appear on the schedule and not the order in which they are judged. It is nonetheless the only result in the paper that a reviewer could reasonably call significant, and we therefore give it the prominence it deserves in the discussion.[5]

Two further results deserve mention. First, the 96 submissions that pitched in two or more sessions in the same year did so without any clashes. Second, among the eleven ranked pairs of first and second place, first-place winners pitched later than their second-place counterparts in four cases, giving a sign test p-value of 0.549. The overall winners, who are descriptive and not inferential units, were SixtCharge (slot 19 of 20 in the Sixt session of 2022), Clarity – The Copilot for Journalists (slot 3 of 18 in the Hubert Burda Media session of 2024), and Vom Platzl (slot 38 of 45 in the City of Munich session of 2025). Each of these three points is consistent with the null hypothesis, and none has a bearing on the primary test. We report them because they are the winners, and because it would be discourteous to pretend that they had not won.

Several facts are worth stating even though they are not formal results. In the 2024 City of Munich session, both the first pitch, Memory Lanes, which was also the popular-vote winner, and the last pitch, engage, at slot 31 of 31, won prizes. In the 2025 SAP session the very first pitch won first place. None of these facts is a test, and the aggregate null is compatible with each of them.

1.5 Roadmap

Section 2 surveys serial position effects in memory, and Section 3 surveys order effects in judged performance, the closer analogue to the present setting. Section 4 describes the hackaTUM event and its Sunday-morning sessions. Section 5 describes the data, the matching procedure and the unidrei incident. Section 6 sets out the tests, and Section 7 reports the results. Section 9 reports the robustness analyses. Section 10 discusses interpretation, and Section 11 concludes. The per-session schedules and the code appear in Appendices A and D.

Readers who wish to skip directly to the answer may do so. The order of pitches does not appear to matter: the data are compatible with a uniform distribution of winners, and they are not compatible with a shift in mean winner position larger than one tenth of the schedule.

[5] It has been suggested to us that a correlation between schedule and submission order might itself produce a serial position effect, if early submitters happen to be better teams. This is a legitimate concern, and it is the reason that the winners' submission-identifier rank is reported as a separate test. The rank is not significantly different from the field, and so the confound, if present, does not appear to be large.

2 Literature Review I: A Brief History of Remembering Things in Order

> "We will pitch first, we promise."
>
> Anonymous hackathon participant, 4:12 a.m.

The question of whether the order in which things are presented affects how they are remembered, judged or chosen is older than most of the hackathons that now provide our data. It is also, we argue, older than most of the disciplines that claim it. This section surveys the strands of the literature that bear on the present study. We begin with the earliest systematic work on serial learning, pass through the serial position curve and the study of primacy in impression formation, and arrive, by way of anchoring and ballot-order research, at the narrow question of whether a team's place in a Sunday-morning queue tells us anything about its chances of winning. Where a body of work is only loosely relevant, we say so. Where it is directly relevant, we attempt to state its findings without embellishment, which is more difficult than it sounds.[6]

2.1 Ebbinghaus and the Dawn of Serial Learning

The experimental study of memory is conventionally dated to the work of Hermann Ebbinghaus, whose investigations of learning and forgetting were published in 1885 and remain the foundation on which much later work is built (Ebbinghaus, 1885). Ebbinghaus wished to study memory in its purest available form, and to this end he constructed a set of nonsense syllables: pronounceable three-letter combinations with no prior meaning for the experimenter. The syllables could be learned in a fixed sequence, and the learner's performance could be measured against the number of repetitions required to reproduce that sequence. In this way serial learning, the learning of items in a fixed order, became one of the first topics to be studied quantitatively.[7]

Ebbinghaus's central contribution, for the purposes of this review, is conceptual rather than numerical. He showed that the ordering of material is not incidental to the act of remembering it. A list learned in sequence is retained according to the structure of that sequence, and the learner's errors are not random: they cluster in particular places, and the position of an item in the list is a predictor of how well it will be recalled. Ebbinghaus's own treatment of these phenomena was limited by the methods available to him, and he did not separate the various components of the curve that later researchers came to distinguish. Nevertheless, the study of serial learning established two propositions that have survived considerable scrutiny: that memory for an ordered list depends on where an item sits within it, and that the effect of position can be measured.

It is worth emphasising, because the point is easily lost, that serial learning in the Ebbinghaus tradition concerns the retention of a sequence by a single learner over repeated trials. It does not concern the evaluation of items by a third party, the choice between alternatives, or the adjudication of competing claims. These are the domains in which the present study is interested. The early work therefore supplies a vocabulary and a set of measurement conventions rather than a theory of the phenomena under investigation. We return to the question of

[6] The authors have made every effort to avoid overstating any finding in this section. Readers who detect a note of excitement are invited to re-read the preceding sentence.

[7] The present authors have not attempted to replicate the methods of the 1885 study, on the grounds that a single subject is not a sample and that a hackathon is already sufficiently demanding.

transfer across domains in Section 2.9.[8]

2.2 The Serial Position Curve: Primacy and Recency

A substantial step forward came with the study of free recall, in which subjects are presented with a list of items and are then asked to recall as many of them as they can, in any order. Murdock's 1962 study is the classic reference for the resulting pattern, which is now known as the serial position curve (Murdock, 1962). When recall is plotted against the position of each item in the list, the curve is characteristically U-shaped, or more precisely bowed: items near the beginning of the list are recalled more often than items in the middle, a pattern termed the primacy effect, and items near the end are also recalled more often than items in the middle, a pattern termed the recency effect. The middle of the list, in other words, is the place where memory is worst.

The serial position curve is one of the most reproduced findings in the psychology of memory, and it has been reproduced under many conditions. It is useful to state its essential features carefully, because they are the features on which any later comparison with other domains must rest. First, the shape is a function of position, not of content: the same curve appears with different kinds of material. Second, the two ends of the curve are not symmetrical in their causes. The primacy effect is generally observed to depend on the conditions under which earlier items are processed and retained, while the recency effect is generally observed to depend on the conditions under which the final items are still available at the time recall is requested. Third, the magnitude of each effect is sensitive to the time elapsed between presentation and recall and to the nature of any activity interpolated between them.

The classic interpretation of these features is that they reflect two distinct memory stores. On this account, items that have been rehearsed and transferred into a durable long-term store are retrieved on the basis of that durable trace, which produces the primacy effect, since early items have had the most opportunity for rehearsal. Items still held in a short-term store at the moment of recall are retrieved from that store, which produces the recency effect, since the final items have not yet been displaced. The middle items are neither sufficiently rehearsed to be durable nor sufficiently recent to be held in the short-term store, and so they are recalled least well. We present this account here as the standard interpretation of the curve, which is how it is most often taught. We do not adopt it as the explanation of any phenomenon in the present study. The reader will note, however, that the structure of the account is familiar from the analysis of later domains, and that it supplies the first of the mechanisms we will need when we come to derive predictions.[9]

2.3 Primacy in Impression Formation

The study of impression formation provides the first domain in which order effects are studied in a setting closer to evaluation than recall. Asch's 1946 study of the formation of impressions of personality is the standard early reference (Asch, 1946). In it, subjects were presented with lists of trait adjectives describing a hypothetical person and were asked to form an overall impression. The central observation was that the order in which the traits were presented affected the

[8] The transfer question is one that the literature has, by and large, declined to answer directly. This is not a criticism. It is a sensible response to a question that is very difficult to answer.

[9] It has been observed that a theory with two stores is, in some respects, more elegant than a theory with one, and in others, less. The present authors are not in a position to adjudicate this matter and have declined to try.

resulting impression, and that traits presented early tended to exert a disproportionate influence on the judgement. The effect is commonly described as a primacy effect in impression formation, and it has been a durable feature of the social-psychological literature ever since.

Anderson's 1965 study extended this work by treating order as a variable to be manipulated systematically rather than a nuisance to be controlled (Anderson, 1965). The study examined the primacy effect in personality impression formation using a generalised order-effect paradigm, in which the same set of traits could be presented in different sequences and the resulting judgements compared. The paradigm allowed the investigators to ask not only whether order mattered, but in what direction and under what conditions. It is for this reason that Anderson's work is frequently cited as the point at which order effects in evaluation became a subject of method rather than anecdote.[10]

Two features of the impression-formation work are relevant here. The first is that the primacy effect in this domain concerns the weight given to information encountered first, rather than the probability of recall. A judge who encounters favourable information early is, on this account, more likely to form a favourable overall impression, and to interpret later information in that light. The second feature is that the effect is a feature of evaluation, not of storage. The judges in these studies were not forgetting the later traits; they were integrating the traits into an impression in which earlier items carried more weight. This distinction matters for the present study, because it separates the question of whether a team is remembered from the question of whether it is judged favourably. A pitch might be perfectly remembered and still be evaluated differently depending on when it occurred in the session.

The impression-formation literature does not, however, establish that primacy is universal. The effect is observed under some conditions and not others, and the conditions that matter most appear to concern the way in which information is presented and the way in which judgements are made. These considerations lead naturally to the belief-adjustment model, which attempts to bring several of the findings together under a single framework.

2.4 The Belief-Adjustment Model and the Length of Lists

The most fully developed account of order effects in evaluation is the belief-adjustment model proposed by Hogarth and Einhorn (Hogarth and Einhorn, 1992). The model addresses a question that the earlier literature had left open: why order effects should appear in some circumstances and not others, and, where they appear, why they should sometimes favour early information and sometimes favour late. The authors frame the problem as one of belief updating over a sequence of evidence. A judge begins with an initial belief, receives items of evidence one at a time or all at once, and revises the belief as each item is processed. The order effects that emerge are, on this view, the consequence of the particular way in which the revision takes place.

The model's central mechanism is anchoring and adjustment. The judge's current belief serves as an anchor, and each new item of evidence produces an adjustment to that anchor whose size and direction depend on the properties of the evidence and of the current belief. The model thus allows the same body of evidence to produce different final beliefs depending on the sequence in which it arrives, because the anchor established by early evidence shapes the way later evidence is weighed. The authors' account of order effects is therefore structural:

[10]The word "generalised" in the title of the 1965 study is, as far as the present authors can determine, the only word in the relevant literature to have been used in a sense that is both accurate and unambiguous.

it locates the source of the effect in the process of updating, not in the evidence itself. The judgement that results is a path-dependent quantity, and its dependence on path is what the order effect measures.

The empirical content of the model turns on two sets of moderating conditions, and these are the conditions that matter most for the present study. The first concerns the response mode of the judge: whether the evidence is evaluated step by step, as each item arrives, or whether the judgement is formed at the end of the sequence, after all of the evidence has been received. The second concerns the length of the series: how much evidence there is, and how it is distributed across the sequence. The model's framework is built so that the direction of the resulting order effect is not fixed in advance, and the abstract attributes order effects to the interplay of information-processing strategies and task characteristics. The model is therefore not, in the simplest sense, a theory of primacy or a theory of recency. It is a theory of the conditions under which either may appear, and the precise mapping of conditions to directions should be taken from the original article rather than from this summary.[11]

This conditional structure has a direct consequence for the present study, which the reader will recognise as the reason the literature review has been organised as it is. Consider a jury, or in our case a partner panel, that hears a sequence of pitches over the course of a single morning. The sequence contains between seven and 45 pitches, depending on the session, and the pitches are delivered in slots of roughly four to five minutes. Some sessions include a break after about 12 pitches. The model does not predict a single direction for such a session. If the panel forms its judgement step by step, updating a running impression as each pitch concludes, the length of the series and the spacing of the breaks would be expected to shape whether early or late pitches exert more influence. If the panel instead forms its judgement at the end of the session, when the entire sequence has been received, the balance of influence may shift toward whichever pitches are most available at that point. The model thus supplies a family of competing predictions, rather than one prediction, and the choice among them depends on facts about the judging process that the present study cannot observe directly.

We should be explicit that the authors of the model do not present it as a theory of hackathon judging, and that the application to Sunday-morning pitches is our own. It would be inappropriate to suggest that the model has been tested on a population of partners, or that it predicts any particular outcome for the sessions we examine. What it does provide is a principled reason to expect that the direction of any order effect is an empirical matter, and that a study which assumes one direction in advance is not, in the relevant sense, testing anything. The competing predictions that we formalise in Section 2.9 are drawn from this observation.

2.5 Anchoring and Comparison Processes

The notion of an anchor, which the belief-adjustment model makes central, has a longer history in the judgement and decision-making literature. Tversky and Kahneman's 1974 treatment of judgement under uncertainty describes anchoring and adjustment as a heuristic by which people estimate a quantity by starting from an initial value and moving away from it, typically insufficiently (Tversky and Kahneman, 1974). The heuristic was offered as one of several by which judgement under uncertainty is simplified, and it has since been applied to a very wide range of judgements. Its relevance to order is that the initial value, whatever its origin, may

[11] Readers familiar with the earlier sections of this review may wonder whether the belief-adjustment model is compatible with the serial position curve. The authors regret that they cannot offer a complete answer to this question, and suspect that no complete answer is available.

exert a disproportionate effect on the final estimate, and that the initial value is, in a sequence of evidence, typically the first thing encountered.

A 2003 study (Mussweiler, 2003) develops a more specific account of how anchors operate, focusing on comparison processes rather than on adjustment as such. The emphasis on comparison is significant for the present study, because it suggests that a judgement of a given item may depend not only on the item itself but also on the items that precede it, which serve as points of comparison. A pitch judged against the pitch before it is a different judgement from the same pitch judged against an abstract standard.

Kahneman's 2011 synthesis of the judgement literature brings anchoring together with the related heuristics and biases in a form accessible to a general readership (Kahneman, 2011). The synthesis is useful for the present purposes chiefly as a reminder of how widely the anchoring phenomenon has been observed, and of the caution with which it should be applied. The anchoring literature has been criticised on methodological grounds, and the magnitude of anchoring effects in particular settings has been contested. The present authors take no position on these disputes. We note only that anchoring offers a second, independent reason to expect that the first items in a sequence may carry disproportionate weight, and that comparison processes offer a reason to expect that the effect of an item may depend on its neighbours.[12]

These two lines of work, anchoring as adjustment and anchoring as comparison, are not mutually exclusive, and the belief-adjustment model can be read as drawing on both. For the present study the relevant implication is modest but important. If judgements are anchored, then the order of evaluation is not a neutral feature of the process, and the position of an item may affect its evaluation through the anchor that the preceding items establish. Whether this effect is positive or negative depends on the relation between the item and the anchor, a relation that the literature does not specify in a way that would allow us to predict the sign in advance.

2.6 Ballot-Order Effects: The Closest Analogue to Position in a List

Among the bodies of work that most closely resemble the present problem, ballot-order research is perhaps the closest. An election ballot is a list of candidates or options presented in a fixed order, and the voter selects one or more of them. The question of whether the position of a candidate on the ballot affects the vote is therefore a question about the effect of position in a list on a choice, which is the question we ask of the pitch session. The literature addresses it directly, and has produced a body of evidence that is relevant in both its findings and its limitations.

A 1998 study (Miller and Krosnick, 1998) is among the most cited treatments of the topic, and a 2008 study (Ho and Imai, 2008) examines ballot-order effects further. Both are cited for the general finding that first-listed options can gain an advantage. We cite both studies for the general finding that first-listed options can gain an advantage, and we do not attribute to either study specific effect sizes, sample characteristics or moderating conditions beyond those described here, which the reader should consult in the original works. The general finding maps, in the terms of the earlier sections, onto a primacy effect: the option encountered first receives the benefit.

The analogy to the pitch session should be drawn with care. A ballot is presented to a voter in a single moment, and the voter chooses among the options in view. A pitch session

[12] The term "anchor" is used in the judgement literature with a precision that is, at times, exceeded only by its use in ordinary speech. The authors have adopted the technical sense throughout.

is experienced over several hours, and the judges encounter each pitch in its own time. The ballot-order literature therefore speaks most directly to the case in which the list is presented at once and evaluated as a whole, which corresponds to the end-of-sequence condition of the belief-adjustment model rather than the step-by-step condition. It does not follow that a ballot-order effect will appear in a Sunday-morning session. It follows only that the closest analogue to the question of position in a list is one in which first-listed options have been found, in some settings, to gain an advantage. The question of whether that advantage survives the transition from a ballot to a session is precisely the question the present study addresses.

2.7 Choice Order

A further strand of work considers order effects in choice among options that are tasted, compared or consumed in sequence rather than merely listed. A 2009 study of wine tasting (Mantonakis et al., 2009) is the reference point here. Its finding is that, in a sequence of options, the first and last tend to be preferred. The pattern is therefore a U-shape in preference, and it is the closest finding in the literature to the third of the competing predictions we will derive.

The wine-tasting result should be read with the usual caution. It is a finding about a particular kind of sensory evaluation, in which the judge's experience of each option is directly affected by the options that precede and follow it. It is not obvious that the same mechanism operates when the options are pitches delivered over several minutes, or when the judgement is a prize rather than a personal preference. Nonetheless, the finding is informative for two reasons. The first is that it establishes that a U-shape in evaluation is at least a conceivable outcome of a sequential judgement process, and that the edges of a sequence may receive more favourable treatment than its middle. The second is that it provides a concrete example of an order effect in which the direction of influence is neither purely primacy nor purely recency, which is the case we must be prepared to recognise if it occurs.[13]

2.8 Synthesis and Limits of the Literature

Taken together, the literature reviewed above does not supply a single theory of order effects that can be applied without modification to a Sunday-morning pitch session. It supplies a set of mechanisms, each of which has been observed in some domain and each of which points, under some conditions, in a particular direction. The serial position curve supplies a primacy mechanism and a recency mechanism, the former associated with the early portion of a sequence and the latter with its end. The impression-formation work supplies a primacy effect in evaluation. The belief-adjustment model supplies a structural account in which the direction of the effect depends on the processing strategy and the length of the series. The anchoring and comparison literature supplies reasons to expect that early items may anchor later judgements and that the effect of an item may depend on its neighbours. The ballot-order and choice-order literatures supply, respectively, a primacy finding for listed options and a U-shaped finding for sequentially tasted options.

Three features of this body of work should be borne in mind by any reader who proposes to apply it to a new setting. The first is that the mechanisms are not mutually exclusive, and that a single process may produce effects of different direction depending on the conditions. The

[13] The authors are aware that the phrase "wine tasting" is sometimes used in the literature as shorthand for a much larger family of experiments, and that not every member of the family has reported the U-shape. We have cited the study that we have, and have not attempted to generalise from it beyond what we can defend.

second is that most of the evidence concerns individual judges, controlled laboratory tasks or single-sitting decisions, and that the extension to a panel judging a sequence of performances over several hours is an extension rather than a replication. The third is that the literature does not, as far as we can determine, contain a direct study of the Sunday-morning pitch session, which means that the question we ask is genuinely open. It is the openness of this question, rather than any expectation about its answer, that justifies the study. The literature is sufficient to make the question worth asking, and insufficient to answer it in advance.[14]

2.9 Implications for the Sunday-Morning Pitch

We now derive the competing predictions that the subsequent sections test. The derivation proceeds from the mechanisms reviewed above, applied to the setting of a pitch session in which a partner panel hears between seven and 45 pitches, in slots of roughly four to five minutes, and selects two winners for the session. Each prediction concerns the normalised position $x = (j - 1)/(n_s - 1)$ of the winning slots, where j is the slot index within a session of n_s slots, so that $x = 0$ denotes the first pitch and $x = 1$ the last. We write W_s for the set of winner slots in session s, with $|W_s| = 2$, and T for the mean normalised position of all winners, which is the primary statistic of the study.

The first prediction is a null hypothesis, and it is the one against which all others must be judged. Under P0, winners are distributed uniformly across the slots of each session, so that T has expectation one half and no position is favoured. P0 is not a neutral default. It is the prediction that the sessions are, with respect to winning, indistinguishable from random orderings, and it is the prediction that would be expected if the panel evaluated each pitch on its merits without regard to when it occurred. Equally, it is the prediction that would be expected if order effects were present but cancelled out in aggregate, and the study must be able to distinguish these possibilities.

The second prediction, P1, is a primacy prediction. Under P1, the panel's judgement is anchored by early pitches, or the early pitches are better remembered and more favourably weighed, so that winners are over-represented in the early portion of each session. In terms of the study's statistics, P1 implies that T lies below one half, that the early tercile of each session contains more winners than expected, and, as a corollary, that the first pitch after a break is more likely to win than other pitches. The corollary follows from the observation that a break may reset the context in which the first pitch after it is judged, and that a judge who has been anchored by the early portion of the session may, after a break, form a fresh anchor from the pitch that follows. We do not claim that the corollary is strongly motivated, and we regard it as a secondary implication of P1 rather than as an independent prediction.

The third prediction, P2, is a recency prediction. Under P2, the panel's judgement is dominated by the most recent pitches, either because the belief-adjustment process weights late information more heavily under the relevant conditions or because the late pitches are more available at the point at which the judgement is formed. In terms of the study's statistics, P2 implies that T lies above one half and that the late half of each session contains more winners than expected. P2 is the prediction most naturally associated with the end-of-sequence condition of the belief-adjustment model, but, because the model admits both directions, we do not assume that the end-of-sequence condition necessarily produces recency. The prediction is

[14] The present authors were, at one point, tempted to conclude this section by observing that the answer is 42. We have resisted the temptation, on the grounds that the answer to a different question is not an answer to this one (Adams, 1979).

included because it is one of the directions the model permits, and because it is the direction most easily confused with the primacy prediction when the statistic is a simple average.

The fourth prediction, P3, is a U-shape prediction. Under P3, winners are over-represented at both edges of each session, so that the first and last pitches are more likely to win than the middle, in the manner reported for sequentially tasted options. P3 implies that the mean position T may lie close to one half, since the two edges offset one another, and that the edge share of winners is elevated, with the quadratic term in a conditional logit of win on position being positive. This is the least intuitive of the predictions in the present setting, because a mean near one half is consistent with P0 as well, and the study must therefore rely on the distributional and quadratic tests to distinguish P3 from the null. A U-shape with a mean of one half is a difficult pattern to detect, and the reader should be sceptical of any claim that it has been detected on the basis of the mean alone.

The four predictions can be summarised as follows, with the mechanisms that motivate each. Table 2 sets them out in a single place for the convenience of the reader. Two features of the table deserve emphasis. The first is that the mechanisms are listed with their literature and with the direction they predict where the literature supports a direction, and that where it does not we say that the direction is conditional. The second is that the predictions are not chosen to be exhaustive of the possibilities. They are chosen because they are the predictions that the literature reviewed above makes, in the form in which it makes them, and because each can be stated in terms of the statistics that the study computes.

The competing predictions are therefore four in number: the null P0, and the three directional alternatives P1, P2 and P3. The subsequent sections test them. The primary statistic is T, against which P1 and P2 are distinguished by the direction of departure from one half, and P0 and P3 are distinguished from one another by the distributional and quadratic tests, since a U-shape and a uniform distribution share a mean. We state the predictions here in advance, as the study's methods require, and we note that the predictions are not mutually independent. A session that shows a primacy effect in its early tercile may simultaneously show a recency effect in its final tercile if the two mechanisms operate together, and the study does not assume that they cannot. What the study assumes is only that the predictions are distinct enough to be tested, and that the tests are, in the terms of the pre-declared analysis, capable of telling them apart.

The literature supports a set of predictions that are distinct and testable, even where it does not support a preferred direction. That is the modest conclusion on which the present study rests.[15]

3 Literature Review II: Juries, Judges, and the Hungry Mind

> It is a capital mistake to theorize before one has data.
>
> Arthur Conan Doyle, *A Scandal in Bohemia*

The present study asks a question of almost embarrassing modesty. When a team stands up on a Sunday morning and pitches its project to a partner's jury, does the moment of its turn in the queue matter? The question is small. The literature that bears on it is not. Serial

[15] No effect sizes from the cited literature are used in the derivation of P0 through P3. The predictions are directional and qualitative, and the study's inference does not depend on any magnitude taken from the works reviewed here.

Table 2: Mechanisms reviewed in this section, the literature that motivates them, and the direction of slot-position effect they predict for a Sunday-morning pitch session.

Mechanism	Literature	Predicted direction of position effect
Rehearsal into a durable store (classic interpretation of the curve)	Murdock (1962)	Primacy: early pitches favoured (P1)
Short-term retention of final items (classic interpretation of the curve)	Murdock (1962)	Recency: late pitches favoured (P2)
Early impression weighted more heavily in evaluation	Asch (1946), Anderson (1965)	Primacy (P1)
Anchor-and-adjust updating over a series	Hogarth and Einhorn (1992)	Conditional on processing mode and series length; P1, P2 or neither
Anchoring on an initial value	Tversky and Kahneman (1974), Kahneman (2011)	Primacy, via the first anchor (P1)
Comparison with preceding items	Mussweiler (2003)	Conditional on the relation to neighbouring pitches
Listed-first options gain an advantage	Miller and Krosnick (1998), Ho and Imai (2008)	Primacy for listed options (P1)
First and last options preferred in sequential tasting	Mantonakis et al. (2009)	U-shape: edges favoured (P3)
No systematic position effect	Null hypothesis	Uniform distribution of winners (P0)

position, the tendency of an item's place in a sequence to shape how it is remembered, judged or chosen, has been studied for well over a century, in settings that range from laboratory word lists to tasting rooms, parole hearings and television talent shows. This section surveys that literature in five parts. The authors regard each part as necessary. A reviewer may regard some of them as optional, and the authors would not dispute the reviewer's right to say so, provided the reviewer has read all of them first.

A brief note on scope is in order. The review is confined to studies in which items, people or proposals are evaluated one after another by an audience that can, in principle, compare them. Studies of pure recall are included because the serial position curve they established is the conceptual ancestor of everything that follows. Studies of hackathons are included because, to the authors' knowledge, nobody else has thought to include them in a review of this kind.[16] The authors have also been unable to find a satisfactory answer to the question of order in the literature. They have concluded, with Adams (1979), that the answer may be 42, and they have declined to pursue the matter further in this section.

[16]The authors searched the relevant databases for "hackathon" and "pitch order" and, for reasons of completeness, also for "hackathon" and "order." The second search returned a number of results that the authors are not prepared to discuss in a footnote.

3.1 Serial position in performance competitions

The modern study of order effects begins, for most practical purposes, with the serial position curve. Murdock (1962) documented the familiar shape of recall for word lists, in which items at the start and at the end of a list are remembered more readily than those in the middle, a pattern now usually summarized as primacy and recency. The experimental tradition on which such work rests is older still, reaching back at least to the nonsense syllables of Ebbinghaus (1885), whose successors have since been asked to explain why nobody remembers the middle of anything.

Serial position has also been studied as a matter of judgment rather than memory. Asch (1946) reported that the order in which traits are presented shapes the impression a person forms of another, and Anderson (1965) studied the weighting of information in such impressions. Hogarth and Einhorn (1992) offered a belief-adjustment model in which the order of evidence matters because each new piece of information is weighed against a running summary, a framework that explains why early or late information can dominate depending on the task. Heuristic accounts of judgment under uncertainty, such as those associated with Tversky and Kahneman (1974) and the later synthesis in Kahneman (2011), are usually cited in the same breath. The authors cite them here so that the reader knows they have been read.

Choice settings show a similar pattern. Mantonakis et al. (2009) reported, in a wine-tasting setting, that the first and last options were preferred over those in the middle, a U-shaped pattern that sits awkwardly between a primacy account and a recency account. Miller and Krosnick (1998) and Ho and Imai (2008) studied ballot-order effects, in which the position of a candidate or option on a ballot appears to matter for the outcome. Ballot order is a useful analogy for the hackaTUM setting. A pitch schedule, like a ballot, is a list that someone else has put in an order, and that the audience then takes in one entry at a time.

The performance-competition literature is where the question of this paper is most directly anticipated. Flôres and Ginsburgh (1996) and Glejser and Heyndels (2001) studied the Queen Elisabeth music competition, in which the order of appearance and the day of performance were both found to bear on rankings. Ginsburgh and van Ours (2003) is cited as a general reference for order effects in music competitions. The authors do not summarize its findings beyond noting that the topic is treated there, a formulation they arrived at after some discussion and have not since regretted.

Television and song contests have been studied with more enthusiasm and somewhat less restraint. Haan et al. (2005) reported that in the Eurovision Song Contest later performers tended to do better. Bruine de Bruin (2005) examined jury evaluations in Eurovision and in figure skating and found that later performers received higher scores, a result the authors consider the most frequently repeated sentence in the subfield. Antipov and Pokryshevskaya (2017) examined order effects in song contests more broadly. Page and Page (2010) extended the picture to television talent shows of the Idol type, where later singers were found to do better. The direction of these findings is, in the authors' reading, remarkably consistent.

The same direction appears in figure skating. Bruine de Bruin (2006) reported a similar finding, in which the jury's evaluations appear to be sensitive to the sequence in which skaters perform. Figure skating is judged by panels using structured criteria, which makes the persistence of an order effect there more interesting than it would be in a loosely judged contest. A structured rubric, it might be argued, should make the order of performances irrelevant. The authors are not aware of a version of this argument that has been tested in a hackathon setting,

and they have not attempted to construct one.

The consensus, insofar as one exists, is that later is often better in sequential evaluation. The qualifier "often" is doing real work. Not all studies find a later advantage, and not all of those that do agree on a mechanism. The candidate mechanisms include the fatigue or contrast of evaluators, the greater recency of the last performance in a juror's memory, and the possibility that the order itself was not independent of the performers' quality. The last of these is the subject of the next subsection.

The authors draw two conclusions from this literature. The first is that order effects are common enough to be worth testing for. The second is that they are not reliably in one direction. Both conclusions shape the design of the present study, which is two-sided by declaration. The authors feel that this is the honest choice for a literature this unsettled, and that a one-sided test would have required a confidence in the sign of the effect that nobody in the field has been able to offer.

3.2 Judges, breaks and fatigue

Parole is the setting in which the question of breaks has been most dramatically posed. Danziger et al. (2011) reported that the proportion of favourable parole decisions was higher at the start of each session and declined as the session went on, recovering after a break. The finding was striking, and it was taken up quickly as evidence that the mental state of decision makers can shape consequential outcomes. The authors note that it arrived with a clean mechanism and a memorable narrative, which is the usual route by which a finding comes to be cited by people who have not read it.

Weinshall-Margel and Shapard (2011) offered a critique. Their central point, as the authors read it, concerns the ordering of cases. The sequence in which the decisions were made was not random, so that the apparent effect of time or of breaks may partly reflect which kinds of cases were heard when. A pattern of this kind cannot be read as a judge's state unless the order that produced it is independent of the cases, and the critique is that this independence was not established.

The authors draw an explicit parallel here, and it is not a flattering one for the hackaTUM organizers. The hackaTUM pitch schedules are not random either. In this paper's data, the slot a team receives is correlated with that team's Devpost submission ID. The median within-session Spearman correlation is $\rho = 0.24$, the Stouffer combination gives $z = 7.04$, and the raw p-value is 1.9×10^{-12}, or 1.7×10^{-11} after Holm correction. The schedules loosely follow the order in which teams created their submissions. This is the only significant result in the study, and it is precisely the kind of result that the critique of Danziger et al. (2011) would have us worry about. If slot is partly a function of submission order, then any slot effect is confounded with whatever submission order happens to measure. The authors checked the obvious version of this worry directly. Winners' submission-ID rank has a mean of 0.534 against an expected 0.500, with a raw p-value of 0.367, so there is no evidence that winners were systematically early or late submitters. The non-randomness of the schedule is real. Its consequences for winners are not detectable. Figure 5 shows the relationship session by session.

The scheduler, the authors note, was at least courteous. Of the 96 submissions that pitched in two or more sessions in the same year, none had two pitches at the same time. The minimum gap was four minutes and the median was fifty. The authors read this as evidence that the scheduling process was careful about clashes, and as evidence that it was not careful about

randomness, which is a different and less flattering thing to be careful about.

Break effects have a second, more theoretical route. Baumeister et al. (1998) proposed ego depletion, the idea that self-control draws on a limited resource which effortful acts consume and which rest can partly restore. The idea was influential well beyond the laboratories in which it was first studied, and it offered a ready mechanism for the fatigue that the parole literature appeared to show. Hagger et al. (2016) reported a multilaboratory replication effort that failed to reproduce the depletion effect in the form originally proposed. The authors do not take the failure to settle the question of fatigue, which is a broad and sensible hypothesis. They do take it as a reason to be wary of any mechanism that rests on one clean laboratory result.

The authors' own break analysis is in the same spirit. Some sessions had a break of five to ten minutes after about twelve pitches, and the first pitch after such a break is a natural place for a fatigue or reset effect to appear. The sample contains 17 such slots, of which 2 won, against an expected count of 1.402. The raw p-value is 0.643, and the Holm-adjusted value is 1. The authors decline to speculate about whether the jurors had been on their feet since Friday evening. They note only that the data provide no evidence of a post-break advantage, and that the hypothesis of ego depletion, the most familiar account of why such an advantage might exist, has not held up well under replication. Table 3 summarizes the order-effect settings discussed in this section and the direction of each.

Table 3: Selected order-effect settings and their typical direction. Only rows whose direction the authors are confident of are included. Effect sizes are omitted by design.

Setting	Citation	Typical direction
Eurovision Song Contest	(Haan et al., 2005)	Later better
Eurovision and figure-skating juries	(Bruine de Bruin, 2005)	Later better
Figure skating	(Bruine de Bruin, 2006)	Later better
Idol-style television talent shows	(Page and Page, 2010)	Later better
Wine tasting	(Mantonakis et al., 2009)	Mixed (first and last preferred)
Serial recall of word lists	(Murdock, 1962)	Mixed (primacy and recency)
Impression formation	(Asch, 1946)	Earlier better (primacy)
Parole decisions within sessions	(Danziger et al., 2011)	Earlier better (contested)

3.3 Hackathons as an object of study

Hackathons have attracted a modest literature of their own, and the authors have been pleased to find that it is more serious than its subject might suggest. Komssi et al. (2015) asked what hackathons are for, a question that remains relevant to any organizer who has been asked to justify a weekend of catered food. Briscoe and Mulligan (2014) treated the hackathon as a form of digital innovation, and Nolte et al. (2018) and Pe-Than et al. (2019) studied the organization and practice of such events. These works are cited here generically, as a sample of a field that the authors regard as having more to say than it usually gets credit for. The authors have not attempted to summarize their findings in detail. The present study concerns a detail of hackathon programming that the authors do not associate with any of them.

hackaTUM is the official hackathon of the Department of Computer Science at the Technical University of Munich. It is held yearly in November and runs for about 36 hours, from Friday evening to Sunday. It draws between 800 and 1000-plus participants. Industry partners each set a challenge, and teams pick one or more. On Sunday morning each partner holds a pitch

session in its own room, in which the teams that submitted to that challenge present in a scheduled order, in slots of about 4–5 minutes. Sessions began at 10:15. Some had a break of 5–10 minutes after about twelve pitches. Following TUM news reporting (TUM School of Computation, Information and Technology, 2022, 2024), each partner picks a first and second place winner, so that each session produces two winners. In the afternoon a jury of professors picks an overall winner from among the challenge winners, and a popular-vote award is also given. The authors have described the event in this detail because a reader who has not attended a hackathon deserves an explanation, and because a reader who has attended one will already know all of it and may enjoy being reminded.

The pitch session is the least studied part of a hackathon. To the authors' knowledge, the literature on hackathons is silent on the question of pitch order. The authors searched for prior work on the order in which hackathon teams present and found none. This is a gap in the literature, and the authors describe it with the solemnity it deserves. A field that has produced studies of what hackathons are for, how they are organized and what they mean for innovation has, it appears, not yet asked whether the fourth team on a Sunday morning is treated differently from the twenty-fourth. The authors consider this an oversight of some seriousness, and have addressed it with 582 pitch slots, 62 winners and a considerable amount of Python.

The data for the study come from official pitch schedule PDFs, the Devpost winner lists (hackaTUM, 2022, 2024, 2025) and TUM news articles (TUM School of Computation, Information and Technology, 2022, 2024; TUM Chair of Information Systems and Business Process Management, SAP CoLab, 2025). Official schedules were available for 32 sessions. Three sessions were excluded from the analysis. The 2022 Optiver session had all 43 teams scheduled in a single 30-minute block, which leaves no order to study. The 2024 and 2025 Optiver sessions have no schedule PDF. The final sample comprises 31 sessions, 582 pitch slots and 62 winners, with session sizes ranging from 7 to 45.[17] Winners were matched to slots by following each schedule's Devpost submission URL to its canonical project page, which resolved 55 of the 62 winners. The other seven were matched by name or slug, because Devpost returned HTTP 403 during resolution owing to rate limiting. Each of these seven was verified by hand.

3.4 The replication crisis and why null results deserve publication

The present study reports a null result, and the authors would like to say a few words about why this is not a shameful thing to do. Ioannidis (2005) argued that a large share of published research findings are likely to be false, given the combination of small effects, small samples and the flexibility with which analyses can be performed. Simmons et al. (2011) documented how researcher degrees of freedom can make almost any result appear significant, and Gelman and Loken (2014) described the forking paths through which a single dataset can yield many analyses, each of which seems justified when considered alone. These works are not, on the authors' reading, primarily about null results. They are about the asymmetry by which positive results are publishable and nulls are not. That asymmetry makes a literature of significant findings look more reliable than it is.

The statistical community has responded with some vigour. Wasserstein and Lazar (2016) issued a statement on the use of p-values, cautioning against treating them as the probability that a hypothesis is true, or as a substitute for judgment. Benjamin et al. (2018) proposed

[17] The 2022 Solana schedule labels two different submissions "unidrei." The one in slot 1 actually links to the project S-Moody, which is a winner. The authors trust the URL. Dropping the session changes nothing, which the authors regard as reassuring and also, for a session with a schedule quirk of this kind, somewhat suspicious.

that claims of new discoveries be held to a stricter threshold, at a significance level of 0.005. The present study takes both cautions seriously. The primary test is pre-declared. Secondary tests are corrected with the Holm procedure (Holm, 1979). The p-values are reported as what they are, namely the probability of a result at least this extreme under the null, and not as statements about the world.

Null results need their own tools. A nonsignificant test does not establish that an effect is absent, and the field has been confused on this point for a long time. Cohen (1988) is the standard reference for power analysis, and Hoenig and Heisey (2001) explained why post hoc power calculations are of limited use. When the aim is to show that an effect is small, the more useful approach is to test for equivalence. Schuirmann (1987) introduced the two one-sided tests procedure, and Lakens (2017) provides a practical introduction to equivalence testing for applied researchers. The present study uses a TOST with bounds of ± 0.10 around 0.5. The TOST p-value is 0.0043, so equivalence is established, and any shift in mean winner position larger than 0.10 is rejected.

The authors also report a minimum detectable effect at 80% power of ± 0.106. The equivalence bound of ± 0.10 is therefore a little narrower than the minimum detectable effect. The study does not have the power to detect an effect of the size that the bound rules out. It is not required to, since an equivalence test asks a different question from a test of difference, and the authors consider this one of the more elegant features of the design, if not one that anybody will notice on a first reading.

The methods draw on a long lineage. The randomization logic of Fisher (1935) and the permutation tests of Pitman (1937) underlie the primary test. Good (2005) and Ernst (2004) give general treatments of permutation methods, and Phipson and Smyth (2010) argue that permutation p-values should never be reported as zero, a rule the present study obeys. Efron (1979) introduced the bootstrap, which is used here for the confidence interval, and Efron and Tibshirani (1993) provides an accessible introduction.[18] The conditional logit used for the position effects follows McFadden (1974) and Chamberlain (1980), with sessions as strata. The Spearman rank correlation is due to Spearman (1904). Its combination across sessions uses the Fisher z transformation (Fisher, 1915) and the Stouffer method (Stouffer et al., 1949). The authors acknowledge Student (1908) for the t distribution, which they did not use but would have been embarrassed not to mention. They also acknowledge Rubin (1974) for the potential-outcomes framework, which supplies a language for asking what a team's slot would have been worth had it been moved. The general proposition that statistical models are approximations, familiar to every applied statistician, is due in its best-known form to Box and Draper (1987).

3.5 Positioning the present study

The literature reviewed above suggests a clear, if uneven, picture. Later performers often do better in sequential evaluation. Breaks may matter, though the mechanisms proposed for them are contested. Hackathons are a serious field with a gap in it. Null results are harder to publish than they should be. The present study is positioned in the gap between these observations, and the authors wish to be clear about what it is. It is the first, and quite possibly last, study of its kind.

[18] The analysis is carried out in Python using NumPy (Harris et al., 2020), SciPy (Virtanen et al., 2020), statsmodels (Seabold and Perktold, 2010) and Matplotlib (Hunter, 2007). The random seed is 2025. The authors have no further comment on the seed.

The authors considered whether to describe the work more modestly and concluded that doing so would be inaccurate in one direction and unnecessary in the other. The claim of priority is made to the authors' knowledge, which is the appropriate standard for a claim of this kind. The claim of finality reflects the authors' judgment that the question of pitch order at hackathons is unlikely to be pursued by anyone else with an equal commitment to a null result. The authors would be delighted to be proved wrong on this point, though they are not planning for it.

The contributions of this paper are as follows.

(i) A pre-declared permutation test of the mean normalized position of hackathon winners, accompanied by a bootstrap confidence interval, a minimum detectable effect and an equivalence test, applied to 31 sessions, 582 pitch slots and 62 winners.

(ii) A family of secondary tests, corrected for multiplicity with the Holm procedure, covering the late-half share, tercile structure, first and last slot wins, edge wins, post-break pitches, conditional logit specifications with and without a quadratic term, and the relationship between slot and submission order.

(iii) A confound check showing that the schedule is not random with respect to submission order. The authors regard this as the only significant result in the study and as the one they would least like to have produced.

(iv) Robustness analyses on subsets defined by year, by session size, by the exclusion of multi-session teams and by the exclusion of one anomalous schedule.

(v) A ranked-pairs analysis of first against second place, using a sign test, together with the observation that all ten 2022 first-place teams pitched between normalized positions 0.17 and 0.70, and that none pitched first or last.

(vi) A descriptive account of the three overall winners, offered with the explicit acknowledgement that $n = 3$ is not a sample in any sense that a statistician would recognize.

(vii) A documented procedure for matching hackathon winners to schedule slots through submission URLs, with manual verification of the seven matches that could not be resolved automatically.

(viii) A list of contributions longer than the contributions of most papers in this literature. The authors regard this as a service to the reader, who is entitled to regard it otherwise.

The authors are aware of several limitations, and they set them out here in the interest of completeness. The sample is small. Only one session, CHECK24 2024, is confirmed as objectively scored, which is too few for a negative-control comparison between objective and subjective judging. Huawei 2024 was mixed, combining CTF puzzle points with a manual grade. Most other sessions were judged by pitch or demo, or their criteria are unknown. The authors accept these limitations and suspect that a fair share of the studies reviewed in this section share some of them, though those studies were conducted under conditions that the authors have not been able to verify in detail.

The authors close this section with a hope, which they state plainly. They hope that the present study will be the first of its kind, and that it will not be the last merely because someone

else has at last found an order worth studying. Future work on pitch order at other hackathons, in other years and under other scoring regimes would be welcome. The authors do not expect it, but they note that the null result reported here is, by the standards of the field, a rather strong one, and that it deserves to be replicated by anybody willing to stay up until Sunday morning.

4 Institutional Background: The hackaTUM Sunday Morning

> "The slides are on the other laptop, which is in the other room, which is locked."
>
> Anonymous hackathon participant, 4:12 a.m.

This section establishes the institutional setting in which the present investigation was conducted. We describe the event, the partner-challenge model that structures its Sunday morning, the scheduling conventions that govern the order of pitches, the award structure that follows them, and the criteria (where these could be established) by which the pitches are judged. Several of these matters are, the authors must candidly report, incompletely documented. Where a claim cannot be supported by the sources available to us, we say so. We have found it useful to be explicit on this point, since the subsequent analysis depends on the order of pitches and not on the content of the pitches themselves, and a reader who has not been warned may otherwise assume that we understand more about the content than we do.

4.1 hackaTUM in Brief

hackaTUM is the official hackathon of the Department of Computer Science at the Technical University of Munich.[19] It is held yearly in November and runs for about 36 hours, from Friday evening until Sunday. During that time it draws between 800 and 1000-plus participants, who form teams, build software over a single weekend, and present the results to a panel of judges.(TUM School of Computation, Information and Technology, 2022, 2024)

The event is organised around challenges. Industry partners each set a challenge, which is a problem statement the partner would like participants to address. Teams are free to pick one challenge or several, and a team that wishes to be considered for more than one partner award may submit its project to each of the relevant challenges separately. Submissions are collected through Devpost, which is also where the winners of each challenge are listed.(hackaTUM, 2022, 2024, 2025)

The weekend concludes, in the conventional manner of such events, with a ceremony at which prizes are handed out. The overall winner is chosen in the afternoon, after the sessions described below, so the bulk of the substantive evaluation is completed some hours before the final award is announced. We do not regard this as a finding; we regard it as a scheduling fact that a reader of the subsequent analysis should keep in mind.[20]

Two features of the event matter for what follows. First, the number of challenges is not fixed: the partners differ from year to year, and so does the number of sessions and teams in

[19] The authors note that "official" is used here in the sense of the event's own branding and that they did not independently verify the chain of institutional approval behind it. We were unable to establish how many signatures the phrase required.

[20] The authors do not know whether the ordering of the afternoon jury and the closing ceremony is deliberate, and we were unable to establish who decided it.

each. Second, the event is large enough that the Sunday morning cannot be run as a single room. Each partner therefore holds its own pitch session, in its own room, at which the teams that submitted to its challenge present in turn. We now describe that structure in more detail.

4.2 The Partner-Challenge Model

Each partner's pitch session follows the same basic pattern. Teams that submitted to that challenge are placed in a scheduled order and pitch in turn, in slots of about four to five minutes. Sessions in the data began at 10:15. Some sessions included a break of five to ten minutes after about twelve pitches; others did not. The authors could not establish a rule that determines which sessions received a break, and the schedules themselves do not record a reason.

Table 4 lists every partner that appears in the final sample, together with the size of its session in each year. The session sizes are the numbers of pitch slots in the official schedule for that session. The table is intended as a reference for the reader who wishes to know which partners were present in which year; it is not, in itself, a statement about the partners' motivations or about the quality of their challenges, on which the authors have no information.

Table 4: Partner challenges by year, with the number of pitch slots in each session. A dash indicates that the partner had no session in that year. The 2022 Optiver session (43 teams) is excluded from the analysis because all teams were scheduled in a single 30-minute block, and it is marked here for completeness. The 2024 and 2025 Optiver sessions have no schedule PDF; these are marked "no PDF" and are likewise excluded. The partner listed as SIXT in 2025 is merged with Sixt in 2022 and is treated as one partner.

Partner	2022	2024	2025
Algorand	8	–	–
Bloomberg	24	–	–
CHECK24	–	30	–
City of Munich	34	31	45
Huawei	15	7	10
Hubert Burda Media	21	18	–
IMC	–	–	29
Interhyp	24	12	19
JetBrains	–	19	21
Logitech	–	–	17
Optiver	43, excl.	no PDF	no PDF
Reply	–	–	22
Rohde & Schwarz	12	–	–
SAP	–	–	12
Siemens	–	9	8
Sixt	20	–	27
Solana	7	20	–
T-Systems	–	32	–
Wildtrack	10	7	12
Sessions in sample	10	10	11
Pitches in sample	175	185	222

Reading the table by columns gives the totals reported in the final sample: 31 sessions, 582 pitch slots, and session sizes from 7 to 45. The 2022 column contains ten sessions in the sample, not eleven, because the Optiver schedule is excluded as described above; the 2024 and 2025 columns are complete for the sessions that have schedules. Reading the table by rows shows that some partners are persistent: City of Munich, Huawei, Interhyp and Wildtrack appear in all three years, while Hubert Burda Media, JetBrains, Siemens, Sixt and Solana appear in two

each. The remaining partners were present in a single year only. We note that the partner names are reproduced as they appear in the schedules, with the exception of the merging of the two Sixt entries.

The model has an obvious practical consequence. A team that submits to a single challenge is scheduled into one session. A team that submits to several is scheduled into several, and each of those sessions has its own order, its own length and its own break, which the team must somehow attend. We return to this in the discussion of scheduling below.

4.3 A Phenomenology of the Five-Minute Pitch

We offer the following account of the Sunday morning in a deliberately stylised register. It is not an empirical description. No observations were recorded in a structured form, no participants were interviewed, and none of the details below were measured. It is included because a reader who has never been to a hackathon may find the setting difficult to picture, and because the phenomenology of the pitch session is, in our view, the part of the event that the numbers in this paper cannot themselves convey.

The session begins with a room that has been set up in the night. Chairs are arranged in rows facing a screen, which at the time of the first pitch shows the partner's logo, a title slide and, in some cases, a small timer. The partner's representatives sit near the front. The judges sit in a separate row, usually with a notebook, and usually with a coffee that they have been carrying since before the session began. The judges' coffee is, for the purposes of this paper, a fixed feature of the setting; we have no evidence that it affects the scoring, and we have not attempted to measure its consumption.[21]

The teams in the rows behind the judges are, almost without exception, tired. Their laptops are open on their knees or on the seat in front. Their cables are in a small tangle, and a team member is usually responsible for an HDMI adapter that is, at the critical moment, not quite the right adapter. The team has rehearsed its pitch many times over the course of the night, and it has done so in a series of quiet, increasingly distorted versions, one of which is the version it will now deliver. The pitch itself is short. Four or five minutes is enough time to introduce the problem, show the solution, mention the technology stack, and leave a small silence for questions. In the time available, a great deal is compressed, and the team's attention is correspondingly concentrated on the one or two sentences that it expects to be remembered.

When the team's name is called, its members stand, walk to the front, and connect the laptop. The screen flickers. Somewhere in the room a mouse is moved too quickly. The opening sentence is usually delivered with more confidence than the team feels. What follows is a sequence of slides, a demonstration if one is possible, and a closing statement. Questions are invited in the last thirty seconds, and are sometimes asked. The judges write something down. The next team is already standing up.

Two aspects of this scene are relevant to the analysis. The first is that the slot is short, so that the position of a team in the session is, in a literal sense, a position in a queue of similar experiences. The second is that the team at position one and the team at position forty-five are each experiencing the room differently: the first team sees a fresh audience and a judge with a full cup, while the last team sees a room that has been listening for three hours and a judge whose notebook is already partly full. Whether these differences affect the outcome is

[21] The authors are aware that the coffee could be a confounder. We were unable to obtain a measure of its strength, and we did not, as a matter of principle, ask the judges to rate it.

the question the rest of this paper is designed to answer, and we will not anticipate that answer here.

4.4 The Scheduling Problem

The organisers of each session must place a set of teams into a sequence. The problem is, on its face, a simple one: the teams are known, the slot length is fixed and the session has a start time. For the teams that submit to several challenges, however, the problem acquires a second dimension. Such a team must be present at each of its sessions, and it must not be scheduled to pitch in two sessions at the same time. The authors could not establish which of the organisers, if any, was responsible for solving this problem, nor the algorithm (if any) that was used.

The data nonetheless allow us to describe the outcome. In the sample, 96 submissions pitched in two or more sessions in the same year. None of them had two pitches at the same time. The minimum gap between two pitches by the same submission was four minutes, which is exactly one slot, and the median gap was 50 minutes. Put differently, across the multi-session submissions, the scheduler evidently avoided clashes in every case observed. We report this as a fact about the schedules. We do not know whether the avoidance was the product of a deliberate constraint, of a careful manual process, or of good fortune distributed across 96 cases in a manner that the authors find difficult to credit.

A slot-level description helps to make the scheduling concrete. Let session s contain n_s slots, indexed by $j \in \{1, \dots, n_s\}$, and define the normalised position of slot j as

$$x = \frac{j - 1}{n_s - 1}, \tag{4}$$

so that $x = 0$ is the first pitch and $x = 1$ is the last. This normalisation makes sessions of different sizes comparable, and it is the quantity used in the primary analysis. It also has an awkward property that the reader should keep in mind: a session of seven pitches and a session of forty-five pitches are, after normalisation, both spread evenly over the unit interval, even though a team in the forty-five-pitch session waits, on average, about seven times as long before its turn.

The largest session in the sample illustrates the point. The 2025 City of Munich session contained 45 pitches in four-minute slots, running from 10:15 to 13:11, which is to say that its final pitch began nearly three hours after the first. In such a session the normalised position is a poor proxy for the experience of waiting. A team at $x = 0.5$ in that session has been in the building for well over an hour, and a team at $x = 0.5$ in a session of seven pitches has waited about twelve minutes. The two teams occupy the same point on the normalised scale and, in every practical sense, the same point in the morning. We mention this because it is a limitation of the analysis that we have chosen to accept rather than to correct, for reasons that are set out in the methods section.

It may be useful to say a word about the breaks. In sessions that had them, a break of five to ten minutes followed about the twelfth pitch. The first pitch after a break therefore has a special status, in that the team concerned will have been, at least briefly, outside the room while the rest of the audience was resting. Seventeen such slots appear in the sample. The authors draw no inference from this at the present point, but we note the existence of a small literature, of which Danziger et al. (2011) is the best-known example, suggesting that decisions made before breaks may differ from those made after them. Weinshall-Margel and Shapard

(2011) is a sustained critique of that work and observes that the order of cases in the original study was not random. The sessions described here are, in this respect, at least as non-random as the parole hearings that motivated that critique; their breaks, however, are a matter of the organisers' schedules rather than of any decision-maker's appetite.

Finally, one scheduling finding deserves a forward reference, since it becomes the only significant result of the study. The pitch order within a session loosely follows the order in which the corresponding Devpost submissions were created. The authors did not establish whether this reflects a scheduling rule, the natural sequence in which the organisers received registrations, or a coincidence of the two.[22] We return to it in the results section.

4.5 The Award Structure

The awards follow the sessions. Per the TUM news reports, each partner selects a first-place and a second-place winner from among the teams that pitched in its session, so that each session produces two winners.(TUM School of Computation, Information and Technology, 2022, 2024) In the sample, this gives 62 winners across 31 sessions, which is exactly two per session and is the reason the winner count in the analysis is fixed by design.

In the afternoon, a jury of professors selects an overall winner from among the challenge winners. The overall winner is therefore not drawn directly from the pitch sessions but from the set of their winners, which is itself a selection. A reader who wishes to model the overall prize as a function of pitch position must, accordingly, model two stages: the selection of a challenge winner within a session, and the selection of an overall winner from the challenge winners. The present analysis addresses only the first stage. We have no information about the deliberations of the jury, and we have not sought any.

A third award, the popular vote or crowd choice, is also presented during the event. We were unable to establish how the popular vote was cast or by whom. It is therefore not an outcome of the pitch sessions in the sense that the challenge winners are, and it is excluded from the analysis for that reason.

The three overall winners in the data are shown in Table 5. We record them here, with their normalised positions, for descriptive purposes only. The sample is far too small, at three cases, to support any inference, and the authors do not wish the table to be read as evidence for or against any hypothesis about overall winners.

Table 5: Overall winners in the years for which the data were collected, with the slot each occupied in its partner session and the corresponding normalised position $x = (j-1)/(n_s-1)$. The sample is three cases. The table is descriptive and is not used in any test.

Year	Team (partner session)	Slot	Normalised x
2022	SixtCharge (Sixt)	19 of 20	$18/19 \approx 0.947$
2024	Clarity – The Copilot for Journalists (Hubert Burda Media)	3 of 18	$2/17 \approx 0.118$
2025	Vom Platzl (City of Munich)	38 of 45	$37/44 \approx 0.841$

The table is, in our view, a faithful record of a small and rather unremarkable set of facts. It would be a serious error to read the three values as a pattern, and the authors wish to state explicitly that the values were not chosen to illustrate anything. A reader who finds that the

[22] The authors note that a sign of this kind would be expected under several mechanisms, none of which we were able to rule in or out, and that the Devpost submission identifiers are themselves assigned by the platform rather than by the teams.

three overall winners are not clustered near the middle of their sessions is invited to notice that three is a number of cases, and that the probability of any particular arrangement of three values on the unit interval under a uniform null is not one that this paper is designed to estimate.

Two further pieces of award data are available. For the eleven ranked pairs in which both a first and a second place are known, the 2022 pairs are drawn from cit2022, which covers all 2022 challenges. The 2025 SAP pair is drawn from ioc2025, which is the only source for that session. The 2024 news reports support the general award structure but do not, to the authors' knowledge, give a full ranking of first and second places in each 2024 session, and we therefore do not use them to establish any ranked pair.

4.6 What the Judges Evaluate

The question of what judges evaluate is, in principle, the first question a reader would ask about any competition, and it is the last one to which we are able to offer a clean answer. The short version is that the criteria are, for most partners, unknown to the authors. The longer version is set out below.

Across the sessions, judges appear to evaluate some combination of three things: the pitch itself, the demonstration of the product where one is possible, and, where it is available to the judges, the code. The emphasis on each is, as far as the authors can tell, not standardised, and the weighting between them is not recorded in the sources we consulted. On the authors' understanding, the pitch is the only component that every team delivers, and it is therefore the component most likely to be assessed in every session. The demonstration depends on the technology of the team and on whether the laptop cooperates. The code, where it is inspected at all, is inspected under conditions that the authors would describe as hurried.

Two partners are exceptions to the general pattern of pitch- and demo-based assessment, and the exceptions are instructive. CHECK24 in 2024 ran an automated load-test leaderboard. In such a leaderboard the score is a function of the system's measured behaviour under load, and the judging is in that sense objective: two teams whose systems behave identically would receive the same score regardless of who pitched first or whose laptop had the better adapter.(TUM School of Computation, Information and Technology, 2024) The authors regard CHECK24 2024 as the only session in the sample for which the basis of judgement is confirmed to be objective.

Huawei in 2024 was mixed. Its assessment combined CTF-style puzzle points, which are objective in the same sense as the load-test leaderboard, with a manual grade assigned by a judge. The manual component is, by implication, subject to all the considerations discussed in the preceding paragraphs. The authors therefore do not classify the Huawei 2024 session as objective.

For every other partner, the criteria are either unknown or, to the best of the authors' knowledge, pitch- or demo-based. We could not establish the criteria for most of the sessions in the sample, and we have not inferred them from the nature of the challenge or from the names of the teams. We do not, accordingly, have a clean basis for a comparison of objectively and subjectively judged sessions beyond the two cases above. This is, in our view, the central practical constraint on the design of the study.

The implication for the analysis is straightforward, though it is not the one a researcher would hope for. A negative-control comparison, in which the sessions with objective judging are contrasted with the sessions with subjective judging, would be the natural way to check whether any observed order effect is a feature of human evaluation specifically. The present

data contain only one confirmed objective session, together with one mixed session, and that is too few cases to support such a comparison. We therefore do not attempt it. The reader should understand that the absence of a difference between objective and subjective sessions is not, for this reason, evidence that no such difference exists, and the authors are aware that the literature on evaluative order effects, from the figure-skating and song-contest studies of Bruine de Bruin (2005) and Page and Page (2010) onwards, has generally found the relevant effects in settings where judging was subjective.[23]

We close this section with a remark that the authors consider important, though it is not a finding. The sessions in the sample were judged by people. Those people were, for the most part, different people from one session to the next, and the authors did not record their identities, their number, or the rules, if any, by which they reached their decisions. This is a limitation of the data, not a feature of the analysis, and the reader who wishes to model judge-level variation will need data that do not exist in the sources available to us.

5 Data

> "Begin at the beginning," the King said gravely, "and go on till you come to the end: then stop."
>
> Lewis Carroll, *Alice's Adventures in Wonderland*, 1865

This section describes the data on which every subsequent result rests. It is written in the order in which the data were collected, cleaned, joined and, eventually, reduced to the sample used in the analysis. Readers who are primarily interested in the findings may skip directly to Section 5.6 and the codebook in Table 7, although we note that a number of decisions made in the pipeline, several of which are documented below, have an effect on the final sample that the reader should be aware of before drawing any conclusion from it.

The unit of analysis is the *pitch slot*. A pitch slot is a single line of a partner's pitch schedule, corresponding to one team and one scheduled time. Each slot carries a position within its session, a flag indicating whether it was a winner, and a number of descriptive attributes. Our analytic question is whether the position of a slot within its session predicts whether its team is a winner. The data that answer this question are, in the most literal sense, a list of sheets of paper, a list of website addresses and a list of names. Most of the work described here consists of making these three lists agree with one another.[24]

5.1 Sources

Three kinds of source were used. The first, and by far the most important, is the official pitch schedule for each partner session on the Sunday morning of the hackathon. These schedules were published as PDF files. Each partner holds its own session in its own room, and each session has its own schedule, which lists the teams that submitted to that partner's challenge in the order in which they were to pitch. Sessions started at 10:15. Teams were allotted slots of

[23] The authors are aware that the citation of these studies in a section on judging criteria may suggest a more systematic connection than we are able to establish. It is offered as context, not as an argument.

[24] We wish to record, for the benefit of future investigators, that no part of this work required the use of a database. A spreadsheet, a text editor and a small quantity of Python were sufficient. Several members of the consortium have expressed a desire for more sophisticated infrastructure. These requests have been noted and will be considered in a future paper of at least equal length.

approximately four to five minutes. Some sessions included a break of five to ten minutes after about twelve pitches, which is indicated on the schedule by a line containing the word BREAK.

Schedule PDFs were not all produced to a common template. We identified three distinct layouts, one for each year in which the event was held over the period under study.

- **2022.** The 2022 schedules list a time range for each pitch, for example a start time followed by a hyphen and an end time, and they are organised under titled headers, one per challenge. There is no column recording the size of the team.
- **2024.** The 2024 schedules list a start time for each pitch and include a column headed "Repo?", which records whether the team had a public code repository. The column headed "Repo?" is not used in the analysis.[25]
- **2025.** The 2025 schedules again list a start time for each pitch and include a column headed "Team Size", which records the number of members on the team.

Each schedule line for a pitch contains a link to the team's Devpost submission page, of the form `hackatumYY.devpost.com/submissions/` followed by a numeric identifier and a slug. These identifiers are the second important feature of the schedule data, and we return to them in Section 5.3 and in the analysis of submission order.

The second source is the list of winners. The Devpost project gallery for each year publishes its winners in a structured JSON-LD block, which contains an `ItemList` of winning projects. Each entry carries the project's name, its canonical URL and a list of award strings. Each award string names a challenge, separated from the challenge name by a comma and followed by an em dash and a description. Two awards are not challenge awards: the overall winner, which is chosen by a jury of professors in the afternoon, and the popular vote, which is decided by the crowd. We stored the three winner lists as they were retrieved, in files named for their year, and cite them as (hackaTUM, 2022, 2024, 2025).

The third source is a small number of news articles published by the university, which we use to establish the ranking of first and second place within a challenge. Each partner picks a first and a second place winner from its own challenge, which means that every session has exactly two winners. The article covering the 2022 challenges is cited as TUM School of Computation, Information and Technology (2022), and the 2024 article is cited as TUM School of Computation, Information and Technology (2024). For the 2025 session of the SAP challenge, the ranking is taken from TUM Chair of Information Systems and Business Process Management, SAP CoLab (2025), which is the only source that covers that session. The rankings are used only in the analysis of first and second place reported in the Results, and they are not needed for the primary test.

5.2 Extraction

Schedule PDFs were converted to plain text using `pdftotext` in its layout-preserving mode, which keeps the horizontal alignment of the page and so allows columns to be recovered by their position in each line. We then processed the text line by line. The extraction is performed by a single script, `parse_schedules.py`, which writes one row per pitch slot to an intermediate

[25] It is sometimes argued that repository status is a proxy for the quality of a project. We did not test this claim, partly because it is not relevant to the question at hand, and partly because the schedules were not designed to answer it.

table of schedule rows. A second script, `build_dataset.py`, joins those rows to the winner lists and writes the final table of slots. Both scripts are included in the appendix.[26]

The extraction proceeds through four steps.

Times. Each line is tested for a leading time. The pattern accepts a start time written as hours and minutes, optionally followed by a seconds field, and optionally followed by a second time that is taken to be the end of a range. Only the start time is retained, and it is converted to minutes after midnight, so that 10:15 becomes 615. The pattern accepts the seconds field because a small number of the 2024 schedules print times in a seconds-bearing format, in which a pitch is written as, for example, `12:55:00` rather than `12:55`. Without the optional seconds group, such a line would not match the time pattern, and its time would silently be lost. The anomaly affects only the presentation of the time, and not the order of pitches, which is what the analysis depends on. Every 2024 pitch in the final sample has a time.

The time attached to a pitch is the most recent time that was seen on the page before the pitch's line. Where a pitch's own line carries a time, this is the pitch's time; where it does not, the time is inherited from the nearest preceding time line. Slots for which no time could be determined are recorded with an empty `time_min` and are flagged in the individual-slot indicator. No slot in the final sample lacks a time.

Devpost URLs. Each pitch line is tested for a Devpost submission link. The link pattern captures the two-digit year, the numeric submission identifier and the slug. A line without a link is ignored, except that a line containing the word BREAK, as a whole word, sets a flag for the next pitch. Within each file, lines are read in their printed order.

Slot numbering. The slot number of a pitch is its ordinal among the lines in its file that contain a submission link. Time lines, headers, blank lines and BREAK lines do not consume a slot. Slots are therefore numbered from one within each file, and the slot number is the ordinal position that the pitch occupied in the printed schedule. It is this ordinal, and not the clock time, that the analysis uses as the position of a pitch. The two are closely related, but they are not identical: when a break occurs, the clock time jumps, and the slot number does not. The normalised position used in the analysis is computed from the slot number, as described in Section 5.7.

Break detection and the post-break flag. A pitch is flagged as following a break if a BREAK line occurred since the previous pitch. The flag is reset after each pitch. Seventeen slots in the final sample carry the post-break flag. Two of these seventeen slots won, which is discussed in the Results together with the other secondary tests.

Challenge names are derived from the file names. In the 2022 files, the challenge is the final hyphen-separated token of the file name, which is mapped to a challenge name by a fixed table. In the 2024 files, the challenge is the prefix that precedes the string `-hackaTUM24-`. In the 2025 files, the challenge is the file name with its fixed suffix removed, with one exception: the file for the City of Munich challenge uses a name without the space, and is mapped to the name with

[26] The appendix does not include the PDF files themselves, which are the property of their respective partners and were, in any case, never intended to be read by a computer. They have, however, been stored in a directory of their own, where they are at peace.

the space. We record the challenge names in the form used in the Devpost winner lists, because that is the form in which the winners must be matched.

5.3 Entity resolution

The central difficulty of the data is that a single project may be known by several names, and a single submission may be reached by several addresses. A schedule lists a team name and a submission link. The winner list lists a project name and a canonical project URL. These are not the same string, and in some cases they are not even the same word.

The submission links in the schedules are not canonical. Each of them redirects, with an HTTP 302 response, to a canonical project page of the form `devpost.com/software/` followed by a slug. We resolved each submission link by requesting it without following the redirect, reading the location given in the response, and storing the result in a cache file keyed by the submission link. This arrangement was chosen because automatic redirect following would have discarded the location that we wished to record. The canonical slug is a more reliable identifier of a project than the team name, because it is the address that the platform itself resolves to.

A winner is matched to a slot when the canonical slug recorded in the cache for the slot's submission link is the same as the slug of the winner's canonical URL, within the same year and the same challenge. This rule matched 55 of the 62 winners in the final sample. We refer to these as the redirect matches.

Devpost answered some of our requests with HTTP 403 responses during resolution. We attribute these responses to rate limiting. Where a submission could not be resolved in this way, we attempted to match the winner to a slot by name or slug. The name-or-slug rule accepted a slot when the submission slug matched the winner's slug, when the normalised team name matched the normalised winner name, when one normalised name began with the other, or when the normalised team name was contained in the normalised winner name. A winner was matched only if exactly one slot satisfied the rule within its challenge. If more than one slot did, or none did, the winner was recorded as unmatched and returned to by hand.

Seven winners were matched by name or slug. Each of these matches was verified by hand against the schedule and the project page, and each is listed below, with the name as it appears on the winner list and, where it differs, the name as it appears on the schedule.

- 2022, Bloomberg: the winner is "hackaTUM Exchange", scheduled as "Bloomberg hackaTUM".
- 2022, Algorand: the winner is "Algo2Gether".
- 2022, Huawei: the winner is "Inspect", whose submission slug is `mlm`.
- 2022, Wildtrack: the winner is "Wordless", scheduled under the misspelling "Worldess", whose submission slug is `worldess`.
- 2024, Siemens: the winner is "PrismX by Kaffee&Code", scheduled as "Kaffee&Code".
- 2025, Huawei: the winner is "Pulse", whose submission slug is `huawei`.
- 2025, IMC: the winner is "Eisbach River Trading".

Projects were also renamed between the schedule and the gallery, and between the gallery and its canonical page, in a number of cases. The most instructive is the project that was

entered as CasaNova and whose canonical slug is `nexora`. Another, Rebill, has the canonical slug `repoco`. A third, KMPingPong, was scheduled as PingPongHealthAI. The matching procedure does not depend on any of these names, because the redirect match uses the canonical slug and the name match is used only where the redirect match fails. We mention the renamed projects because they are the cases in which a reader who attempts to reproduce the join by name would go wrong, and because we have found it useful, in other contexts, to be reminded that the name of a thing is not the thing.[27]

The matching produced exactly two winners per session, which is the structure that the partners' articles describe. We regard this as the most important validation of the join, and we take the agreement of the two counts as evidence that the matching has not introduced, or removed, winners. We do not regard it as evidence that each individual match is correct, which is why the seven name matches were checked by hand.

5.4 Exclusions

Three kinds of exclusion were applied. Each is stated here in full, because each affects the sample in a way that a reader might not expect.

The first exclusion concerns the 2022 Optiver session. In this session all 43 teams were scheduled within a single block of thirty minutes. The schedule does not assign an individual time or an individual position to any of the teams, and so the order within the block is not recorded in the source. A position in this session would therefore have been an arbitrary assignment, and an analysis of the position of winners would have been an analysis of the arbitrary assignment. The 43 rows are retained in the slot table, so that the reader may verify that they exist, but they carry an explanatory value in the `excluded` column and are not part of the analytic sample.[28]

The second kind of exclusion concerns sessions for which no schedule PDF exists. Two such sessions are known: 2024 Optiver and 2025 Optiver. For these sessions the order of pitches is not available from any source that we have been able to locate, and they are therefore absent from the analytic sample. We do not know whether the missing PDFs were never produced, were produced and not published, or were published and are, at the time of writing, somewhere in the university's filing system. In the absence of any evidence on the matter we have assumed the least interesting possibility. The 2022 Wildtrack schedule was recovered late in the study, after the authors had already mourned it.

The exclusions leave 31 sessions. Both kinds of exclusion follow from the sources themselves: the Optiver block has no individual order to analyse, and the two missing schedules have no order to analyse at all.

5.5 The unidrei incident

We record here, as a solemn case study, the one anomaly in the schedule data that required a decision about how to treat a source rather than a line of it. We do so with some reluctance, because the incident is minor, and because its resolution had no bearing on any result.

[27] The project names used in this paper are reproduced as they appear in the source data.

[28] We were tempted to include the Optiver block as a single slot. We were advised, by the only person in the consortium with any experience of statistics, that a sample of one session with forty-three teams in it would produce a deeply satisfying result and no information whatsoever. We have followed that advice.

In the 2022 Solana schedule, two different submissions are labelled “unidrei”. The label is not a unique identifier of a project, and a reader who resolved the two submissions by label would have made an error. The first of the two, in slot 1, carries a Devpost link. When that link is followed, it resolves to the project S-Moody, which is a winner. The second submission carries the same label but is a different submission. We therefore faced a choice between trusting the label and trusting the link. We trust the link, because the link is the identifier that the platform assigns to the submission, whereas the label is a human-readable string that was typed by hand.

The decision could in principle have affected the position of the winner in this session, and so we checked. The 2022 Solana session has a winner in slot 1, the first pitch of the session, and the link-based assignment places S-Moody there. Dropping the session entirely, which is the most conservative alternative, changes the primary statistic only slightly. The mean normalised position of the 60 winners in the remaining 30 sessions is 0.508, compared with 0.4994 in the full sample. The robustness analysis in Figure 7 gives the full picture.

We note that the incident is the only occasion on which the source documents disagreed with one another in a way that required judgement. Every other disagreement was resolved by the canonical identifier without difficulty.[29]

5.6 The final analytic sample

The final analytic sample comprises 31 sessions, 582 pitch slots and 62 winners. Each session contributes exactly two winners. Session sizes range from 7 to 45 pitches. Table 6 gives the composition of the sample by year.

Table 6: Composition of the analytic sample by year. Sessions are partner pitch sessions; slots are scheduled pitches retained in the analysis; winners are the two per session, so the winner count is twice the session count in every year. The session-size column gives the smallest and largest session in each year.

Year	Sessions	Slots	Winners	Session size
2022	10	175	20	7–34
2024	10	185	20	7–32
2025	11	222	22	8–45
Total	31	582	62	7–45

The counts in Table 6 were verified against the slot table after the pipeline had been run, by counting the slots whose `excluded` field is empty, and by counting the winners among those slots. The verified per-year slot counts are 175 for 2022, 185 for 2024 and 222 for 2025, which sum to 582. The per-year counts of sessions and winners agree with the same verification. The Optiver 2022 block accounts for the 43 rows that are excluded, and no other rows are excluded.

The sessions with the fewest pitches are 2022 Solana with seven, 2024 Huawei and 2024 Wildtrack with seven each, and 2025 Siemens with eight. The largest session is 2025 City of Munich, with 45 pitches in four-minute slots running from 10:15 to 13:11. The largest session of 2022 is City of Munich, with 34 pitches. Sessions with at least fifteen pitches are considered

[29] The moral character of a PDF is a question on which we have been unable to reach agreement. One member holds that a PDF is a neutral container of whatever it was given. Another holds that a PDF is a solemn commitment, and that a label printed on one is a promise. The incident was resolved in favour of the link, which is to say in favour of the content that the schedule compilers were, on the evidence, most likely to have intended to publish.

separately in the robustness analysis, because the question of whether a session is large enough to be informative is one that the reader should not have to settle for themselves.

Team size is not recorded in the 2022 layout, and the team-size field is therefore empty for all 175 slots of that year. Sessions also differ in their judging basis. Only the CHECK24 challenge of 2024 is confirmed to have been judged by an automated load-test leaderboard, and the judging basis is not a variable in the analysis, because there are too few objectively scored sessions to support a negative-control comparison.

5.7 Variables

Table 7 documents the variables in the slot table. The file contains two further fields that are not used in the analysis. The `individual_slot` field records whether a slot has an individual time, and is zero only for the Optiver block. The `excluded` field records the reason for an exclusion, and is non-empty only for the Optiver block. Neither field affects any result reported in the paper.

Table 7: Codebook for the slot table, one row per scheduled pitch. Variables are listed in the order of the file, with the exception that `pos01` is the analysis variable and is listed last.

Variable	Type	Definition
`year`	integer	Year of the event (2022, 2024 or 2025).
`challenge`	text	Partner challenge, in the form used in the winner lists.
`slot`	integer	Ordinal position of the pitch in its printed schedule, from 1.
`time_min`	integer	Scheduled start time in minutes after midnight (10:15 is 615). Empty where no time could be determined.
`team`	text	Team name as printed on the schedule.
`submission_id`	integer	Numeric Devpost submission identifier, taken from the submission link.
`submission_url`	text	Submission link as printed on the schedule. Not the canonical project URL.
`team_size`	integer	Number of team members, where the layout records it (2024 and 2025). Empty for 2022.
`post_break`	binary	One if the pitch immediately followed a printed BREAK, zero otherwise.
`n_slots`	integer	Number of slots in the session to which the pitch belongs.
`winner`	binary	One if the team is matched to a winner of this challenge, zero otherwise.
`award_name`	text	Project name as given in the winner list, where the team is a winner.
`place`	text	First or second place, where an official source states the ranking. Populated for the 2022 sessions and the 2025 SAP session only.
`overall`	binary	One if the winning project also received the overall winner award.
`popular`	binary	One if the winning project also received the popular vote award.
`match`	text	Matching route: “redirect” or “name/slug”. Empty for non-winners.
`pos01`	real	Normalised position $x = (\text{slot}-1)/(\text{n_slots}-1)$, rounded to four decimals, so that zero is the first slot and one the last.

The variable of interest is `pos01`. Its definition is the normalised position introduced in the

Methods, and it is computed from the slot number and the session size, not from the clock time. The reason for this choice is practical: the clock time is missing for the Optiver block, which is excluded in any case, and is recorded with a different format in some of the 2024 schedules. The slot number is complete for every session in the sample. The normalised position is therefore defined for every slot in the sample, and it takes the value zero for the first pitch of a session and one for the last.

The `overall` and `popular` fields record the special awards. The overall winner is chosen by a jury of professors in the afternoon, from among the challenge winners, and the popular vote is decided by the crowd. Three winning projects carry the overall flag and three carry the popular flag. Both are descriptive variables. Neither enters the primary test, which is defined over challenge winners.

The `place` field is populated for twenty-two winners only. These are the twenty winners of the ten 2022 sessions, for which the first and second places are stated by the official article, and the two winners of the 2025 SAP session, which are stated by a separate article. For the other winners the field is empty, and it should be read as unknown rather than as absent. The paper therefore reports first and second place results for eleven ranked pairs, and does not report them for the remaining sessions.

5.8 Limitations

The data have a number of limitations, and it is appropriate to state them in full before the results are presented.

The first and most important limitation is that the schedule records the *planned* order of pitches, not the order in which pitches actually took place. The partners and the organising team published schedules in advance. Sessions then ran as they ran. Teams may have arrived late, pitches may have overrun, and some scheduled teams may not have been present at all. The data do not record any of this. The analysis therefore estimates the effect of the planned position on winning, which is what a team would have known when it arrived at the room. If the planned order and the realised order differ systematically, for example because late arrivals are concentrated among teams that were scheduled early, then the analysis is blind to the difference.

The second limitation is that no-shows are unknown. A team that was scheduled but did not pitch would appear in the schedule and would not appear in any record of the session. We cannot identify such teams from the sources available to us. If no-shows are correlated with position, for example if teams scheduled late are more likely to have left the building, then the slot counts would be affected in a way that we cannot measure. We note that the schedules are the only available list of who was meant to pitch, and that a list of who was meant to pitch is not the same as a list of who pitched.

The third limitation concerns the scope of the winner data. Each partner picks two winners, and the data record those two. They do not record the scores on which the winners were chosen, and for most challenges the criteria are not published in a form that would allow a comparison across challenges. The sessions are therefore not directly comparable in the way that a reader might wish, and the analysis treats each session as a separate draw from a common null hypothesis, as described in the Methods.

The fourth limitation concerns the matching. The redirect matches are, we believe, reliable. The seven name matches were verified by hand, but a manual verification is not a guarantee, and we cannot rule out a small number of incorrect matches among the winners that were

matched by redirect. We have no evidence that any such error exists, and we have no evidence that any does not.

The fifth limitation is that submission identifiers are not randomly assigned. The Devpost submission identifier increases with the order in which submissions were created, and teams that submitted earlier may differ from teams that submitted later in ways that are unrelated to the schedule. The analysis of the relationship between slot and submission identifier, reported in the Results, is designed to examine this issue, and its findings are discussed there.

Finally, the data cover three years. They are a small number of sessions, judged by a small number of partners, on a small number of Sunday mornings. A reader who wishes to generalise from them should do so with the care that the sample size deserves, which is considerable, and which we have tried to state clearly in every table in this section.[30]

6 Methodology

> The schedule is fixed; everything else is a matter of opinion.
>
> Anonymous hackathon participant, 4:12 a.m.

This section sets out the statistical machinery used to answer a deceptively small question: whether a team's position in the Sunday-morning pitch order is associated with winning a partner challenge. The machinery is deliberately heavier than the question. We regard this as a feature. A question that can be answered with a sentence should, at minimum, be answered with a theorem.[31]

The data consist of $S = 31$ pitch sessions with $\sum_s n_s = 582$ slots in total, of which 62 are winners. Session sizes range from 7 to 45. Each session is one partner challenge in one year, and each partner challenge holds its own pitch session in its own room. The sessions were drawn from the official pitch schedules and the Devpost winner lists (hackaTUM, 2022, 2024, 2025), and the first-versus-second ranking was taken from TUM news reports (TUM School of Computation, Information and Technology, 2022, 2024; TUM Chair of Information Systems and Business Process Management, SAP CoLab, 2025). Of the 62 winners, 55 were matched to schedule slots by following each schedule's Devpost submission URL to its canonical project page. The remaining seven were matched by name or slug, because Devpost returned HTTP 403 during resolution, which we attribute to rate limiting. Each of the seven was then verified by hand, a process that took rather longer than the rate limit.[32]

[30] The consortium considered including a fourth year of data. It was pointed out that the fourth year would require a fourth set of PDFs, that the PDFs for the fourth year would have a fourth layout, and that a fourth layout would require a fourth section of this paper, which the editors have indicated would be unwelcome. We have therefore reported three years, and we look forward to the fourth in a sequel of approximately the same length.

[31] The authors have checked the derivations in this section by hand and by machine. They regard the first check as the more reassuring of the two, since it is the one that the reader is most likely to repeat.

[32] Three sessions were excluded at the outset. In the 2022 Optiver session all 43 teams were scheduled within a single 30-minute block, so no order exists to be tested. The 2024 and 2025 Optiver sessions have no schedule PDF.

6.1 Notation and the sharp null

Index sessions by $s \in \{1, \dots, S\}$ and slots within session s by $j \in \{1, \dots, n_s\}$, where n_s is the number of pitch slots in session s. Each slot is assigned a normalized position

$$x_{sj} = \frac{j-1}{n_s - 1}, \qquad j = 1, \dots, n_s, \tag{5}$$

so that $x_{s1} = 0$ denotes the first pitch of the session and $x_{sn_s} = 1$ the last. The set of winning slots in session s is written $W_s \subset \{1, \dots, n_s\}$, with $|W_s| = 2$, because each partner names a first-place and a second-place team. Equivalently, $w_{sj} = \mathbf{1}\{j \in W_s\}$ is the winner indicator, and $\sum_{j=1}^{n_s} w_{sj} = 2$ for every session.

The null hypothesis of interest is the sharp null of Fisher (Fisher, 1935): slot carries no information about winning. Formally, under H_0 the winning set in each session is a uniformly random two-element subset of the available slots, independently across sessions,

$$\Pr(W_s = A) = \binom{n_s}{2}^{-1} \quad \text{for every } A \subset \{1, \dots, n_s\},\ |A| = 2, \qquad s = 1, \dots, S. \tag{6}$$

The sharpness of the null is the important property. Under (6) the outcome attached to every slot is fixed in advance, and the only thing that varies is the labelling. This licenses the randomization (or permutation) distribution of any statistic, in the sense of Fisher (1935) and Pitman (1937): the reference distribution is obtained by re-randomizing the winner labels inside each session while holding session sizes and the slot structure fixed. Because the permutations are restricted to act within sessions, the procedure is a stratified permutation test, with sessions as strata.

It is worth being plain about what (6) does and does not assert. It describes the distribution that the winner labels would have if the schedule had been assigned at random with respect to winning. The schedule was not assigned at random. It was assigned by partners and organizers, under constraints that are not part of the data. The permutation distribution is therefore a reference against which the observed pattern is compared, not a causal guarantee. A small p-value would say that the observed pattern is unlikely under the random-schedule benchmark, which is a narrower claim than that position causes winning. A large p-value, which is the case here, says that no pattern large enough to detect appears against that benchmark.

Because the slot-level outcome is a binary indicator with a known number of ones per stratum, all of the tests below are finite-population tests. None of them requires an assumption about the distribution of team quality, judge temperament or the weather in Garching at 10:15 on a Sunday. That is a considerable advantage, and it is the principal reason for choosing the permutation framework over a parametric alternative.

6.2 The primary statistic

The primary statistic is the mean normalized position of the 62 winners,

$$T = \frac{1}{62} \sum_{s=1}^{S} \sum_{j \in W_s} x_{sj}. \tag{7}$$

Values of T near 0 indicate that winners pitch early, values near 1 that they pitch late, and values near $\frac{1}{2}$ that they are spread through the order in the way that chance would produce.

The statistic was pre-declared as the primary test. It is the only test that enters the power analysis and the equivalence test.

We now derive its distribution under H_0. Let $I_{sj} = w_{sj}$ denote the winner indicator. Under (6), each slot is equally likely to be part of the winning pair, so

$$\Pr(I_{sj} = 1) = \frac{\binom{n_s-1}{1}}{\binom{n_s}{2}} = \frac{2}{n_s}, \qquad j = 1, \ldots, n_s. \tag{8}$$

The numerator counts the 2-subsets that contain a fixed slot j; the denominator counts all 2-subsets. Two facts about the positions of a single session are needed. The first is that the normalized positions are equally spaced and symmetric about one half:

$$\sum_{j=1}^{n_s} x_{sj} = \frac{1}{n_s - 1} \sum_{j=1}^{n_s} (j-1) = \frac{1}{n_s - 1} \cdot \frac{n_s(n_s-1)}{2} = \frac{n_s}{2}, \tag{9}$$

so the mean of x_{sj} over the slots of session s is exactly $\frac{1}{2}$. The second is the consequence of (8), namely that the expected sum of winning positions in session s is

$$\mathbb{E}_{H_0}\left[\sum_{j \in W_s} x_{sj}\right] = \mathbb{E}_{H_0}\left[\sum_{j=1}^{n_s} I_{sj}\, x_{sj}\right] = \sum_{j=1}^{n_s} \frac{2}{n_s}\, x_{sj} = \frac{2}{n_s} \cdot \frac{n_s}{2} = 1. \tag{10}$$

The expectation is therefore exactly 1 per session, whatever its size. Summing over the $S = 31$ sessions and dividing by 62 gives

$$\mathbb{E}_{H_0}[T] = \frac{1}{62} \cdot 31 \cdot 1 = \frac{1}{2}. \tag{11}$$

The expected value of the primary statistic under the null is one half, regardless of how the session sizes are distributed. This is a consequence of the design and holds without any reference to the data.

For the variance we use a standard finite-population result, which we state as a lemma and prove, since the proof is short and the result is less widely used than it deserves to be.

Lemma 1 (Variance of a sum drawn without replacement)**.** *Let $y_1, \ldots, y_N$ be a finite population with mean μ and variance $\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \mu)^2$. Draw k units uniformly without replacement, with sampled values $Y_1, \ldots, Y_k$. Then*

$$\operatorname{Var}\left(\sum_{i=1}^{k} Y_i\right) = k\,\sigma^2\,\frac{N-k}{N-1}. \tag{12}$$

Proof. Subtracting μ from every value shifts the sum by the constant $k\mu$ and leaves the variance unchanged, so we may assume $\mu = 0$. Then $\sum_a y_a = 0$ and $\sum_a y_a^2 = N\sigma^2$. Each Y_i has the marginal distribution of a uniform draw from the population, so $\mathbb{E}Y_i = 0$ and $\mathbb{E}Y_i^2 = \sigma^2$. For $i \neq i'$ the pair $(Y_i, Y_{i'})$ is uniform over the $N(N-1)$ ordered pairs of distinct units, so

$$\mathbb{E}[Y_i Y_{i'}] = \frac{1}{N(N-1)} \sum_{a \neq b} y_a y_b = \frac{1}{N(N-1)} \Big[\Big(\sum_a y_a\Big)^2 - \sum_a y_a^2\Big] = \frac{-N\sigma^2}{N(N-1)} = -\frac{\sigma^2}{N-1}. \tag{13}$$

Expanding the variance of the sum over the k indices and the $k(k-1)$ ordered pairs of distinct

indices gives

$$\operatorname{Var}\Big(\sum_{i=1}^{k} Y_i\Big) = \sum_{i=1}^{k} \mathbb{E}Y_i^2 + \sum_{i\neq i'} \mathbb{E}[Y_i Y_{i'}] = k\sigma^2 - \frac{k(k-1)\sigma^2}{N-1} = k\sigma^2\Big(1 - \frac{k-1}{N-1}\Big) = k\sigma^2\,\frac{N-k}{N-1}. \tag{14}$$

□

The factor $\frac{N-k}{N-1}$ is the finite population correction. It is the reason why the variance of the winners' position sum is smaller than the naive $k\sigma^2$ that one would obtain from independent draws. In a session with n slots and two winners, the relevant population is $\{x_{s1}, \ldots, x_{sn}\}$, that is, the equally spaced grid $\{0, h, 2h, \ldots, (n-1)h\}$ with $h = \frac{1}{n-1}$. The population variance of this grid is

$$\sigma_n^2 = h^2 \cdot \frac{n^2-1}{12} = \frac{n^2-1}{12(n-1)^2} = \frac{n+1}{12(n-1)}, \tag{15}$$

where the middle step uses the familiar variance $\frac{n^2-1}{12}$ of the integers $0, 1, \ldots, n-1$. Applying Lemma 1 with $N = n_s$ and $k = 2$ gives the null variance of the session sum,

$$\operatorname{Var}_{H_0}\Big(\sum_{j\in W_s} x_{sj}\Big) = 2\,\sigma_{n_s}^2\,\frac{n_s-2}{n_s-1}. \tag{16}$$

For a session with $n = 20$ slots, for instance, $\sigma_{20}^2 = \frac{21}{228} \approx 0.0921$ and the session variance is $2 \cdot 0.0921 \cdot \frac{18}{19} \approx 0.1745$. Sessions are independent under H_0 by (6), so their variances add. Since T is the sum of these session totals divided by 62,

$$\operatorname{Var}_{H_0}(T) = \frac{1}{62^2}\sum_{s=1}^{S} 2\,\sigma_{n_s}^2\,\frac{n_s-2}{n_s-1}, \qquad \mathrm{SD}_{\mathrm{null}} = \sqrt{\operatorname{Var}_{H_0}(T)}. \tag{17}$$

We evaluated (17) directly on the realized session sizes. The closed form gives a null standard deviation of 0.03776. The Monte Carlo estimate from the permutation distribution, which is the quantity used in all subsequent calculations, is 0.0378. The two agree to within the sampling error of the simulation, which is of order 10^{-4} on a quantity of this size.[33] The simulated null mean is reported as 0.5001, against the analytic value $\frac{1}{2}$ in (11). The gap of 0.0001 is about one Monte Carlo standard error of a mean over 200,000 draws, which is roughly $0.0378/\sqrt{200{,}000} \approx 8.5 \times 10^{-5}$. It is immaterial for every test in this section.

We note also that $\mathrm{SD}_{\mathrm{null}}$ is the single most important number in the paper. It sets the scale of every interval, every power calculation and every equivalence bound below. Its value, 0.0378, reflects the fact that the sample is small: sixty-two winners, each of whose position is spread over the unit interval, produces a mean that is pinned down only to within a few hundredths. The reader who would like the sample to be larger is invited to hold more Sunday-morning pitch sessions. We will not stand in the way.

6.3 Monte Carlo permutation p-values

The exact permutation distribution of T is a product of $\binom{n_s}{2}$ enumerations over sessions and is computationally infeasible to enumerate in full for the largest sessions, although it could

[33] The authors are aware that 0.03776 rounds to 0.0378, which is the simulated value, so the two agree at the precision reported. The reader who wishes to check the calculation may consult Appendix D.

be enumerated in principle. We instead draw $B = 200{,}000$ independent replicates from the null. In each replicate and each session, the winning set is drawn uniformly from the $\binom{n_s}{2}$ two-element subsets. This is done without enumeration by drawing independent Uniform$(0,1)$ keys $U_1, \dots, U_{n_s}$ and retaining the indices of the two smallest keys. By exchangeability of the i.i.d. keys, every 2-subset is equally likely to be the set of two smallest, which is exactly (6).

Let T^{obs} denote the observed statistic and $T^{(1)}, \dots, T^{(B)}$ the null replicates, with $\bar{T}_0$ their mean. The two-sided Monte Carlo p-value is

$$\hat{p} = \frac{1}{B} \sum_{b=1}^{B} \mathbf{1}\{\, |T^{(b)} - \bar{T}_0| \;\geq\; |T^{\mathrm{obs}} - \bar{T}_0| \,\}. \tag{18}$$

The definition measures two-sidedness by distance from the centre of the null distribution rather than from its nominal centre $\frac{1}{2}$. The two coincide to within simulation error here, because the permutation null is symmetric about $\frac{1}{2}$ up to Monte Carlo noise, but the centred form is the one that remains valid when the null is not symmetric. A tolerance of 10^{-12} is subtracted from the right-hand side to guard against floating-point ties; it has no effect on any reported value.

The estimator in (18) can return exactly zero, which is an awkward value for a p-value. Phipson and Smyth (2010) argue that permutation p-values should never be reported as zero, and recommend the estimator

$$\hat{p}_{\mathrm{PS}} = \frac{1 + r}{B + 1}, \qquad r = B\hat{p}, \tag{19}$$

which includes the observed configuration among the permutations. The difference from (18) is

$$\hat{p}_{\mathrm{PS}} - \hat{p} = \frac{1 - \hat{p}}{B + 1}, \tag{20}$$

which is at most $\frac{1}{B+1} \approx 5 \times 10^{-6}$ in absolute value. At the value $\hat{p} = 0.985$ reported for the primary test, the difference is of order 10^{-8} and the reported figure is unaffected by the choice. We nevertheless recommend (19) as the default, and we report the plain proportion only because it is the one computed in the analysis script.

The Monte Carlo standard error of $\hat{p}$ is

$$\mathrm{SE}(\hat{p}) = \sqrt{\frac{\hat{p}(1 - \hat{p})}{B}}. \tag{21}$$

At $\hat{p} = 0.985$ and $B = 200{,}000$ this is approximately 2.7×10^{-4}, so the reported value 0.985 is determined to within about one part in four thousand. That is more precision than is required by a p-value whose only purpose is to say that it is large. The authors nonetheless recorded it to three places, on the grounds that the alternative would have looked like rounding up.

6.4 Bootstrap confidence interval

A 95% interval for the mean winner position is obtained by resampling whole sessions with replacement, which respects the clustering of winners within sessions. Write $A_s = \sum_{j \in W_s} x_{sj}$ for the position sum of session s, so that the observed statistic is the ratio estimator $T = \frac{1}{62} \sum_s A_s$. For bootstrap replicate b, draw S session indices $i_1^{(b)}, \dots, i_S^{(b)}$ uniformly with replacement and

compute

$$\tilde{T}^{(b)} = \frac{\sum_{r=1}^{S} A_{i_r^{(b)}}}{\sum_{r=1}^{S} 2}. \tag{22}$$

The denominator is the number of winners in the resample, which is $2S$ for every replicate because each session contributes exactly two winners. We use 20,000 replicates and the percentile method, so that the interval is $[\tilde{T}_{(0.025)}, \tilde{T}_{(0.975)}]$ in terms of the empirical quantiles of $\{\tilde{T}^{(b)}\}$. The percentile bootstrap is the method proposed by Efron (1979) and developed at book length in Efron and Tibshirani (1993).

We state two caveats. The first is that the interval has only $S = 31$ clusters to work with, and percentile intervals computed from so few clusters are known to be too narrow in some settings. We therefore present the interval as descriptive. The second is that the interval is not used as a test. Our inferential statements rest on the permutation p-values and on the equivalence test in Section 6.6, and the interval is reported only so that the reader can see the plausible range of the mean winner position on the scale of the data.

6.5 Power and the minimum detectable effect

For a two-sided test at level $\alpha = 0.05$ with power $1 - \beta = 0.80$, the minimum detectable shift in the mean winner position under a normal approximation with standard error $\mathrm{SD}_{\mathrm{null}}$ is

$$\delta_{\min} = (z_{1-\alpha/2} + z_{1-\beta})\,\mathrm{SD}_{\mathrm{null}} = (1.96 + 0.84) \times 0.0378 \approx 0.106. \tag{23}$$

The design can therefore detect a shift of the mean winner position from $\frac{1}{2}$ of about ± 0.106 with probability 0.80. A shift of 0.106 corresponds to moving the average winner by roughly a tenth of the entire order, or, in more practical terms, by about three slots in a session of thirty. This is a sizeable difference, and a hackathon in which the early or late pitch carried an advantage of that size would have been difficult to miss.

The MDE is a property of the design, meaning the number of winners, the number of sessions and the spread of positions. It is computed before the data are examined and it does not depend on the observed effect. The reader is warned against the alternative calculation sometimes labelled “post-hoc power”, in which the power is evaluated at the observed effect size. That quantity is a monotone transformation of the p-value and carries no information beyond it (Hoenig and Heisey, 2001). A non-significant result with a small observed effect will always have low observed power, and this says nothing about whether the effect is absent. The appropriate planning calculation is the one in (23), which follows the approach of Cohen (1988).

6.6 Equivalence testing

A non-significant test of H_0 is not evidence of no effect. To make a positive claim of no meaningful effect, we use the two one-sided tests (TOST) procedure of Schuirmann (1987), in the form recommended for behavioural research by Lakens (2017). The equivalence bounds were set to ± 0.10 around $\frac{1}{2}$ before the data were examined. Writing θ for the expected mean winner position, the hypotheses are

$$H_{01}: \ \theta \le 0.40 \quad \text{versus} \quad H_{11}: \ \theta > 0.40, \tag{24}$$

$$H_{02}: \ \theta \ge 0.60 \quad \text{versus} \quad H_{12}: \ \theta < 0.60. \tag{25}$$

Each pair is tested with a normal approximation using $\mathrm{SD}_{\mathrm{null}}$ as the standard error. The test statistics are

$$z_L = \frac{\bar{T} - 0.40}{\mathrm{SD}_{\mathrm{null}}}, \qquad z_U = \frac{0.60 - \bar{T}}{\mathrm{SD}_{\mathrm{null}}}, \tag{26}$$

with one-sided p-values $p_L = 1 - \Phi(z_L)$ and $p_U = 1 - \Phi(z_U)$, where Φ is the standard normal distribution function. Equivalence is claimed at level 0.05 only if both one-sided null hypotheses are rejected, so the TOST p-value is

$$p_{\mathrm{TOST}} = \max\{p_L, p_U\}. \tag{27}$$

With $\bar{T} = 0.4994$ and $\mathrm{SD}_{\mathrm{null}} = 0.0378$ we obtain $z_L = \frac{0.0994}{0.0378} \approx 2.630$, so $p_L \approx 0.0043$, and $z_U = \frac{0.1006}{0.0378} \approx 2.661$, so $p_U \approx 0.0039$. The TOST p-value is therefore $p_{\mathrm{TOST}} = 0.0043$. Both one-sided nulls are rejected, and we conclude that any shift of the mean winner position beyond ± 0.10 from $\frac{1}{2}$ is ruled out at the 5% level.

Two remarks on the procedure are needed. The first is that the normal approximation is used in place of the exact permutation distribution, which is discrete and bounded; at 0.0378 on a scale of $[0, 1]$ the approximation is adequate for the purpose. The second is that the bounds are a choice. A different choice would give a different conclusion, and the reader who finds ± 0.10 too generous or too stingy is welcome to repeat the calculation with ± 0.05. The bounds 0.45 and 0.55 give $z_L = \frac{0.0494}{0.0378} \approx 1.31$ and $p_{\mathrm{TOST}} \approx 0.10$, so equivalence at that tighter bound would not be established at this sample size. We did not dwell on this, and we record that we did not dwell on it for the benefit of any future reader who is tempted to try.

6.7 Secondary statistics

A family of secondary statistics was specified alongside the primary test. Each is a permutation test under the same null (6), and each is computed with the same stratified scheme. We describe them in turn, with the observed and expected counts given in Table 8 below.

Late half. Define $L_{sj} = 1$ if $j > n_s - \lfloor n_s/2 \rfloor$ (a late slot), $L_{sj} = 0$ if $j \le \lfloor n_s/2 \rfloor$ (an early slot), and leave the middle slot of each odd-sized session undefined. The test statistic is the number of winners in late slots, $O_L = \sum_s \sum_j w_{sj} L_{sj}$, and its null expectation is $\mathbb{E}_{H_0}[O_L] = \sum_s \sum_j \frac{2}{n_s} L_{sj}$ by (8). Two winners in each odd-sized middle slot are excluded, which leaves 60 winners in the analysis. The observed count is 33 of 60, against an expectation of 30.01.

Terciles. Let $u_{sj} = \frac{j - \frac{1}{2}}{n_s}$ be the midpoint of the cell that slot j occupies in the unit interval. Assign each slot to a tercile $t_{sj} \in \{0, 1, 2\}$ according to whether $u_{sj} < \frac{1}{3}$, $< \frac{2}{3}$ or otherwise. Let m_{st} be the number of slots of session s in tercile t. The observed winner counts are $O_t = \sum_s \sum_j w_{sj} \mathbf{1}\{t_{sj} = t\}$, the expected counts are

$$E_t = \sum_{s=1}^{S} \frac{2\, m_{st}}{n_s}, \tag{28}$$

and the chi-square-type statistic is

$$X^2 = \sum_{t=0}^{2} \frac{(O_t - E_t)^2}{E_t}. \tag{29}$$

The null distribution of X^2 is obtained by permutation, drawing 50,000 replicates and computing X^2 for each, so that the p-value is the proportion of replicates with X^2 at least as large as the observed value. We do not rely on the asymptotic χ^2_2 approximation, which would give $\Pr(\chi^2_2 \geq 1.673) \approx 0.433$ for the observed statistic. The permutation reference is the one the design implies, and it is the one we use throughout. The observed counts are 21/25/16 against expectations of 20.42/21.16/20.42, giving $X^2 = 1.673$ and a permutation p-value of 0.397.

First and last slot. The primacy and recency statistics are $F = \sum_s w_{s1}$ and $L = \sum_s w_{sn_s}$, with null expectations $\sum_s 2/n_s$ and the analogous quantity for the last slot. Each is tested two-sidedly by the Monte Carlo procedure of Section 6.3, and a one-sided upper p-value is also reported for the first slot. The first slot produced 4 winners against an expectation of 4.304, and the last slot produced 4 against 4.306.

Edges. The edge statistic counts winners in either the first or the last slot of their session, the statistic that a U-shaped serial position curve would make large. Its observed value is 8 against an expectation of 8.620.

First pitch after a break. Some sessions had a break of five to ten minutes after about twelve pitches. Let $b_{sj} = 1$ if slot j is the first pitch after such a break and $b_{sj} = 0$ otherwise, and restrict the test to the sessions that had a break. Under H_0 the expected count is $\sum_s \sum_j \frac{2}{n_s} b_{sj}$. Seventeen slots in the analysis are first pitches after a break, and two of them won, against an expectation of 1.402.

6.8 Conditional logit with fixed effects

The secondary tests above treat the sessions as units of a finite population. A regression formulation is also useful, because it allows the position to enter as a continuous variable and because it provides an estimate of the magnitude of any association. The natural model is a logit for the binary winner indicator with a session-specific intercept. Let

$$\Pr(w_{sj} = 1 \mid \alpha_s, \beta) = \Lambda(\alpha_s + \beta\, x_{sj}), \qquad \Lambda(u) = \frac{e^u}{1 + e^u}, \tag{30}$$

where α_s is a session fixed effect that absorbs everything about the session as a whole, including its judges, its challenge and its partner's appetite for pizza. The difficulty is that a fixed-effects logit with a parameter per session does not have a well-behaved maximum likelihood estimator, because the number of nuisance parameters grows with the number of sessions. The conditional likelihood of Chamberlain (1980), building on the conditional logit of McFadden (1974), removes the nuisance parameters by conditioning on the sufficient statistic for each α_s.

Here the sufficient statistic for α_s is the number of winners in session s, which equals 2 in every session. Consider a fixed two-element winning set $A \subset \{1, \dots, n_s\}$. Under (30), slots are independent given (α_s, β), so

$$\Pr(W_s = A \mid \alpha_s, \beta) = \prod_{j \in A} \frac{e^{\alpha_s + \beta x_{sj}}}{1 + e^{\alpha_s + \beta x_{sj}}} \prod_{j \notin A} \frac{1}{1 + e^{\alpha_s + \beta x_{sj}}} \tag{31}$$

$$= \frac{\exp(2\alpha_s + \beta \sum_{j \in A} x_{sj})}{\prod_{j=1}^{n_s} (1 + e^{\alpha_s + \beta x_{sj}})}. \tag{32}$$

The denominator $\prod_j (1 + e^{\alpha_s + \beta x_{sj}})$ does not depend on A. Conditioning on the event that the session has exactly two winners, we sum over all two-element sets B and divide,

$$\Pr(W_s = A \mid |W_s| = 2,\ \alpha_s, \beta) = \frac{\exp\big(2\alpha_s + \beta \sum_{j \in A} x_{sj}\big) / \prod_j (1 + e^{\alpha_s + \beta x_{sj}})}{\sum_{B:\,|B|=2} \exp\big(2\alpha_s + \beta \sum_{j \in B} x_{sj}\big) / \prod_j (1 + e^{\alpha_s + \beta x_{sj}})}. \tag{33}$$

Every term in both numerator and denominator carries the same factor $\exp(2\alpha_s) / \prod_j (1 + e^{\alpha_s + \beta x_{sj}})$, which therefore cancels. The session fixed effect has disappeared, leaving

$$\Pr(W_s = A \mid |W_s| = 2,\ \beta) = \frac{\exp\big(\beta \sum_{j \in A} x_{sj}\big)}{\sum_{B:\,|B|=2} \exp\big(\beta \sum_{j \in B} x_{sj}\big)}, \tag{34}$$

which depends only on β and the positions. The conditional log-likelihood, summed over sessions, is

$$\ell(\beta) = \sum_{s=1}^{S} \left[\beta \sum_{j \in W_s} x_{sj} - \log \sum_{B:\,|B|=2} \exp\Big(\beta \sum_{j \in B} x_{sj}\Big) \right]. \tag{35}$$

Its score is

$$\ell'(\beta) = \sum_{s=1}^{S} \left[\sum_{j \in W_s} x_{sj} - \mathbb{E}_\beta\Big(\sum_{j \in B_s} x_{sj} \Big) \right], \tag{36}$$

where B_s is a two-element set drawn with probabilities proportional to $\exp(\beta \sum_{j \in B} x_{sj})$. At $\beta = 0$ the score is the observed position sum less its null expectation, which is the statistic T of Section 6.2 in unnormalized form. The conditional logit therefore generalizes the primary test in a precise sense: it reduces to the mean-position comparison when β is near zero.

The likelihood is maximized numerically, and the standard errors are obtained from the observed information. The statsmodels implementation (Seabold and Perktold, 2010) conditions on the within-stratum sum of the outcome in the same way. The linear coefficient is $\hat{\beta} = -0.007$ with standard error 0.427 and Wald p-value 0.987. The implied odds ratio comparing the last slot with the first is $\exp(\hat{\beta}) = 0.993$, with a 95% interval from $\exp(\hat{\beta} \mp 1.96 \cdot 0.427)$ of 0.43 to 2.29. The interval is wide, and it is consistent with an odds ratio of one.

For the quadratic specification we centre the position at $z_{sj} = x_{sj} - \frac{1}{2}$ and write the linear predictor as $\beta_1 z_{sj} + \beta_2 z_{sj}^2$. The conditional likelihood (34) is unchanged in form, with the exponent replaced by $\beta_1 \sum_{j \in B} z_{sj} + \beta_2 \sum_{j \in B} z_{sj}^2$. Centring does not affect the linear coefficient in the linear model: replacing x_{sj} by $x_{sj} - c$ shifts the exponent of every two-element set in session s by the same constant $-2\beta c$, which cancels in (34). In the quadratic model the centring point changes β_1, because a shift reparametrizes the linear term, but it leaves β_2 unchanged, and we interpret β_2 accordingly. The estimate is $\hat{\beta}_2 = -0.774$, which indicates a slight inverted-U shape, with a Wald p-value of 0.625. A negative quadratic term means that the predicted log-odds of winning are highest near the middle of the order and fall toward both ends. The estimate is imprecise, and we interpret the sign with the caution that a single insignificant coefficient deserves.

6.9 Is the order itself random?

The primary and secondary tests all assume that the schedule is unrelated to whatever determines winning. A check of this assumption is the following. The schedule was produced by the partner's organizers, and organizers frequently order their queue by the time at which sub-

missions arrived. If so, pitch order might be correlated with the Devpost submission number, which is assigned in creation order. Such a correlation would not bias the primary test by itself, but it would be a source of confounding if submission order were also related to winning. The authors therefore examined whether slot order and submission order are related.

For each session s, let ρ_s be the Spearman rank correlation between the slot index j and the Devpost submission identifier of the team in that slot (Spearman, 1904). Under the null that slot order is unrelated to submission order, ρ_s has mean zero. The Fisher transformation (Fisher, 1915) gives

$$z_s = \operatorname{artanh}(\rho_s), \qquad \mathbb{E}[z_s] \approx 0, \qquad \operatorname{Var}(z_s) \approx \frac{1}{n_s - 3}. \tag{37}$$

The variance is the familiar large-sample result for Pearson's correlation under bivariate normality, which we use as an approximation for Spearman's coefficient in these sessions. Sessions with $n_s \leq 3$ would contribute nothing and do not occur here; the smallest session has seven slots. Combining the S standardized values by Stouffer's method (Stouffer et al., 1949) gives

$$Z = \frac{\sum_{s=1}^{S} \sqrt{n_s - 3}\, \operatorname{artanh}(\rho_s)}{\sqrt{S}}, \tag{38}$$

which is approximately standard normal under the null, and a two-sided p-value $p = 2(1 - \Phi(|Z|))$.

The result is the one significant finding in the study. The median Spearman coefficient is $\rho = 0.24$ and the Stouffer statistic is $Z = 7.04$, with $p = 1.9 \times 10^{-12}$. The schedule loosely follows the order in which submissions were created. Per-session coefficients range from -0.32 (2022 Solana) to 1.00 (2022 Wildtrack), so the association is not uniform across partners. We regard this as a fact about how the schedules were built, and it is reported as such. It is not a finding about winning. Winners' submission-ID ranks, by contrast, average 0.534 on a $[0, 1]$ scale where 0.5 is expected, with a permutation p-value of 0.367, so the winners are not visibly drawn from the early or late end of the submission queue.

6.10 Multiplicity

The secondary tests form a family, and we control the familywise error rate across it using the step-down procedure of Holm (1979). Let the family consist of m hypotheses with raw p-values $p_{(1)} \leq \cdots \leq p_{(m)}$, ordered from smallest to largest. The Holm-adjusted p-values are

$$\tilde{p}_{(i)} = \max_{l \leq i} \left\{ \min\left(1,\, (m - l + 1)\, p_{(l)}\right) \right\}. \tag{39}$$

The family in this paper has $m = 9$ members, listed in Table 8. They are the late-half share, the tercile statistic, the first-slot and last-slot tests, the edge test, the first pitch after a break, the linear conditional logit, the Stouffer test of slot against submission order, and the winners' submission-rank test. The quadratic conditional logit is reported as a model check and is not part of the family, because its test is a Wald test of a coefficient rather than one of the enumerated secondary hypotheses. The primary test is not part of the family, since it is the single pre-declared hypothesis and is reported at the nominal level.

Since the Stouffer test has the smallest raw p-value, its Holm adjustment is the largest multiplier, $m = 9$, applied to the raw value. The adjusted value 1.7×10^{-11} is therefore not

Table 8: The secondary family of nine tests, with raw and Holm-adjusted p-values. The Stouffer test is the only result with a Holm-adjusted p-value below 0.05.

Test	Raw p	Holm $\tilde{p}$
Late-half share	0.503	1
Tercile statistic	0.397	1
First slot wins	1.0	1
Last slot wins	1.0	1
Edge (first or last) wins	0.849	1
First pitch after a break	0.643	1
Conditional logit, linear	0.987	1
Slot vs. submission ID (Stouffer)	1.9×10^{-12}	1.7×10^{-11}
Winners' submission-ID rank	0.367	1

sensitive to the choice of correction, and the conclusion is the same under Bonferroni, Holm or any other procedure of comparable strength. The remaining eight tests are unremarkable under any of them.

6.11 Pre-declaration and the garden of forking paths

The analysis was pre-registered. The primary statistic, its null, its bootstrap interval and its equivalence bounds were written down before the secondary statistics were computed. The registration was made with ourselves, in a document that has since been lovingly edited, and no independent registry was consulted. We do not regard this as a defect of the procedure, only as a description of it. The registry in question was the authors, who were, at the time, the only people who had read it.

Pre-declaration is necessary, and not sufficient, for a clean inference. Gelman and Loken (2014) describe the garden of forking paths: a researcher who makes data-dependent decisions about what to analyze can produce a small p-value from noise without ever running a second test. Simmons et al. (2011) show how flexible analyses inflate false-positive rates in psychology and set out a list of practices to avoid. The present analysis is subject to the same hazard, and we list the degrees of freedom that we were aware of. These are the normalization of positions in (5), the treatment of the middle slot of odd-sized sessions, the choice of tercile boundaries, the threshold of 15 pitches for the size robustness check, the definition of a post-break slot, the exclusion of 2022 Solana because its schedule labels two submissions identically, and the choice to report the Stouffer combination rather than a per-session test. For each degree of freedom we have reported the alternative choice in the robustness analysis, the secondary family or the appendix. The exclusion of 2022 Solana is the clearest example. Dropping that session changes the primary mean from 0.4994 to 0.508 and does not change the conclusion, which is the appropriate test of whether a decision matters.

The reader should also be cautious about the p-values themselves, which are not measures of the probability that a hypothesis is true (Wasserstein and Lazar, 2016). A p-value of 0.985 on the primary test does not establish that the effect is zero. It says that the observed mean is entirely typical of what the random-schedule benchmark would produce, and it is the equivalence test in Section 6.6 that provides the positive statement about magnitude.

6.12 Software and reproducibility

All computations were carried out in Python. Array operations and the permutation machinery use NumPy (Harris et al., 2020). Hypothesis tests, the Spearman coefficient, the Fisher and Stouffer combinations and the binomial sign test use SciPy (Virtanen et al., 2020). The conditional logit and the Wald inference attached to it use statsmodels (Seabold and Perktold, 2010), and the figures were produced with Matplotlib (Hunter, 2007). The random number generator is NumPy's default generator, seeded with the value 2025. The seed was chosen with some care. The number 42 was considered and rejected, since a reader might have taken the choice for a joke, and since the answer to the ultimate question of life, the universe and everything has been shown to be of limited use in a permutation test (Adams, 1979).

Each Monte Carlo draw is produced by a single generator in a fixed order, so the reported values are reproducible given the seed, the data and the script. The primary p-value is computed from 200,000 permutations, the bootstrap from 20,000 resamples and the tercile statistic from 50,000 permutations. The last of these is a quarter of the primary count, a difference that arises from the memory footprint of the tercile calculation in the analysis script and that does not change any reported conclusion. The analysis script is reproduced in Appendix D, and the per-session schedules and winner tables appear in Appendix A. Figures 1, 4, 2, 3, 5 and 7 display, respectively, every session, the observed and expected winners per third, the permutation null of the mean winner position, the empirical distribution of winner positions against the uniform, the slot against submission-ID ranks, and the robustness subsets.

The methods are the least interesting part of this paper, and the reader who has reached this point has earned the right to be told so. The conclusion is that, under a test designed to detect any systematic relationship between pitch order and winning, the statistic lies comfortably inside its null distribution, the equivalence bounds are met, and the one result that stands out concerns the schedule rather than the outcome. We have tried to describe the procedures precisely enough that they could be repeated by someone who disagrees with us. We would welcome that.

7 Results

> We are still waiting for the results to be announced.
>
> Anonymous hackathon participant, 4:12 a.m.

This section reports what the data show. It is short in the sense that matters: nearly every test returned the same answer, which is that nothing happened. We have nevertheless reported every test, including the ones that were not expected to do anything, because a results section that omits the unremarkable is a results section that flatters its authors (Simmons et al., 2011). Unless stated otherwise, p-values are two-sided, and all permutation tests use the within-session reshuffling described in the methods.

7.1 Descriptive overview

Figure 1 displays the full sample. Each of the 31 sessions occupies one row. Gray dots are pitch slots, blue dots are winning teams, and an orange ring marks a team with a confirmed first place. It is the closest thing this paper has to a picture of the data, and we recommend looking

at it before reading on, partly because the rest of this section is considerably less interesting than the figure.

The sample comprises 582 pitch slots across 31 sessions. Session sizes range from 7 to 45 pitches. The sample contains 10 sessions from 2022, 10 from 2024 and 11 from 2025, after the exclusions described in the methods section. The partners in each year were as follows. In 2022 they were Algorand, Bloomberg, City of Munich, Huawei, Hubert Burda Media, Interhyp, Rohde & Schwarz, Sixt, Solana and Wildtrack. In 2024 they were CHECK24, City of Munich, Huawei, Hubert Burda Media, Interhyp, JetBrains, Siemens, Solana, T-Systems and Wildtrack. In 2025 they were City of Munich, Huawei, IMC, Interhyp, JetBrains, Logitech, Reply, SAP, SIXT, Siemens and Wildtrack.

Each session contributes exactly two winners. The 62 winners are therefore the 62 observations on which every test in this section rests, and no session contributes more or fewer. Their normalized positions are spread across the whole unit interval, with a mean of 0.4994, which we come to shortly. Four winners occupy the first slot of their session ($x = 0$) and four occupy the last ($x = 1$). We note this symmetry here and return to it, with some feeling, in the discussion of the edge tests below. Two winners, one from 2022 Huawei and one from 2022 Solana, occupy the middle slot of an odd-sized session. They are the only winners excluded from the late-half statistic, which is the one test that requires special handling for odd-sized sessions.

Table 9 and Table 10 list all 62 winners, grouped by session in the order of the sample. For each winner we give the slot, the session size n, the normalized position $x = (j-1)/(n_s-1)$ and, where a source exists, the published place. A dash means that no ranking source was available for that session, which is the case for every 2024 session and for every 2025 session other than SAP. We have split the listing into two tables so that neither exceeds a page. Team names are reproduced as they appear in the schedules and on Devpost, with punctuation kept and trailing spaces removed.

The winners also divide by year, and the division is informative mainly about how little each year contributes on its own. The 20 winners of 2022 have a mean normalized position of 0.487, the 20 of 2024 have a mean of 0.546, and the 22 of 2025 have a mean of 0.469. Each of these means lies well inside the range that a year of two-winner sessions could produce by chance, and the year-level p-values in Figure 7 are 0.843, 0.495 and 0.618 respectively. Each year is therefore consistent with the null, whose expected position is 0.5 in every year.

Of the 62 winners, 24 carry a published place. These are the 20 winners of the ten 2022 sessions, for which the source is the cit2022 news coverage, and the two winners of the 2025 SAP session, for which the source is ioc2025. The 24 ranked winners split evenly, with twelve first places and twelve second places. The remaining 38 winners were not ranked by any source we could consult, and we have not attempted to rank them ourselves.

The judging basis is an additional complication, and one that we flag before any test is interpreted. Only one session, CHECK24 2024, is confirmed to be objectively scored, by an automated load-test leaderboard. The Huawei 2024 session was mixed, combining CTF puzzle points with a manual grade. Most of the remaining sessions were judged on the pitch or demo, or their criteria are unknown. There are therefore too few objective sessions to serve as a negative control, and we do not attempt one. Several features of the sample are worth recording before any test is run, if only because they will be quoted at dinner. In 2024 the City of Munich session produced winners at both ends of the schedule: the first pitch, Memory Lanes, which was also the popular-vote winner, and the last pitch, engage, in slot 31 of 31. In 2025 the two SIXT winners pitched back to back, in slots 24 and 25 of 27. The very last pitch of the 2025

Table 9: Winners of the 2022 and first half of the 2024 sessions, with slot j, session size n, normalized position x and published place. A dash marks a session with no ranking source.

Session	Team	Slot/n	x	Place
2022 Algorand	Kollateral	3/8	0.2857	2nd
2022 Algorand	TUM Blockchain Club - Algo2Gether	5/8	0.5714	1st
2022 Bloomberg	hackaTUM Exchange	8/24	0.3043	2nd
2022 Bloomberg	Rialto Finance	11/24	0.4348	1st
2022 City of Munich	Travel Surfer	20/34	0.5758	1st
2022 City of Munich	David - Cold Cities Warm Stories	23/34	0.6667	2nd
2022 Huawei	Inspect: collect data while solving detective riddles	8/15	0.5000	1st
2022 Huawei	ConquerShot	10/15	0.6429	2nd
2022 Hubert Burda Media	QuickFlick	15/21	0.7000	1st
2022 Hubert Burda Media	StreamifAI	21/21	1.0000	2nd
2022 Interhyp	2gether.green	5/24	0.1739	1st
2022 Interhyp	Rebill	15/24	0.6087	2nd
2022 Rohde & Schwarz	mu (micro-wave)~	1/12	0.0000	2nd
2022 Rohde & Schwarz	abiogenesis	6/12	0.4545	1st
2022 Sixt	ReFlow	8/20	0.3684	1st
2022 Sixt	SixtCharge	19/20	0.9474	2nd
2022 Solana	S-Moody	1/7	0.0000	2nd
2022 Solana	Greeft Loyalty	4/7	0.5000	1st
2022 Wildtrack	TUMder	3/10	0.2222	1st
2022 Wildtrack	Wordless	8/10	0.7778	2nd
2024 CHECK24	AET Team	11/30	0.3448	–
2024 CHECK24	clueless	19/30	0.6207	–
2024 City of Munich	Memory Lanes	1/31	0.0000	–
2024 City of Munich	engage	31/31	1.0000	–
2024 Huawei	Chill Guys / Chill Engine	3/7	0.3333	–
2024 Huawei	-13k Pierdut Lift John Nash	5/7	0.6667	–
2024 Hubert Burda Media	Clarity - The Copilot for Journalists	3/18	0.1176	–
2024 Hubert Burda Media	Inspect & News	10/18	0.5294	–
2024 Interhyp	KeySteps	3/12	0.1818	–
2024 Interhyp	Siggi	8/12	0.6364	–
2024 JetBrains	ChatBrains (CB)	3/19	0.1111	–
2024 JetBrains	EEAgent	16/19	0.8333	–

JetBrains session, slot 21 of 21, won. The very first pitch of the 2025 SAP session, ASDA, won first place. In 2022 the first pitch won second place at both Rohde & Schwarz and Solana. The largest session was the 2025 City of Munich session, with 45 pitches in four-minute slots running from 10:15 to 13:11. In 2022 many winners, and all of the first places, sat in the middle of their sessions. We return to that last observation in the subsection on first and second place, where it receives more attention than it deserves.

7.2 Primary result

The pre-declared primary statistic is T, the mean normalized position of all 62 winners. Under H_0, each session's two winners are a uniformly random pair of its slots, and the expected value of T is 0.5001. We observed $T = 0.4994$.

The two-sided permutation p-value, computed from 200,000 within-session reshuffles of the winner labels, is $p = 0.985$. The 95% bootstrap confidence interval, from 20,000 resamples of

Table 10: Winners of the second half of the 2024 sessions and of the 2025 sessions. Columns as in Table 9.

Session	Team	Slot/n	x	Place
2024 Siemens	RiskFlow	7/9	0.7500	–
2024 Siemens	PrismX by Kaffee&Code	8/9	0.8750	–
2024 Solana	Eventure	3/20	0.1053	–
2024 Solana	Solaura	13/20	0.6316	–
2024 T-Systems	UbiR	21/32	0.6452	–
2024 T-Systems	Fleet Manager	28/32	0.8710	–
2024 Wildtrack	ALEX - Augmented Lecture Explainer	5/7	0.6667	–
2024 Wildtrack	bibradar	7/7	1.0000	–
2025 City of Munich	M-Radl	13/45	0.2727	–
2025 City of Munich	Vom Platzl	38/45	0.8409	–
2025 Huawei	Pulse	6/10	0.5556	–
2025 Huawei	TUMAIgotchi	7/10	0.6667	–
2025 IMC	OutsiderTraders	4/29	0.1071	–
2025 IMC	Eisbach River Trading / orionic	16/29	0.5357	–
2025 Interhyp	Mortgage Moment	2/19	0.0556	–
2025 Interhyp	CasaNova	4/19	0.1667	–
2025 JetBrains	Cleveride	17/21	0.8000	–
2025 JetBrains	KMPingPong	21/21	1.0000	–
2025 Logitech	LogiTune	3/17	0.1250	–
2025 Logitech	MX Machina	16/17	0.9375	–
2025 Reply	Spottair	7/22	0.2857	–
2025 Reply	Health Twin	14/22	0.6190	–
2025 SAP	ASDA	1/12	0.0000	1st
2025 SAP	SAP EcoTrace	5/12	0.3636	2nd
2025 SIXT	SixtSeven	24/27	0.8846	–
2025 SIXT	Cloud Cruiser	25/27	0.9231	–
2025 Siemens	Antibody	2/8	0.1429	–
2025 Siemens	CaraKube	5/8	0.5714	–
2025 Wildtrack	Examinator	3/12	0.1818	–
2025 Wildtrack	StudyPace	4/12	0.2727	–

whole sessions, is $[0.426, 0.576]$. The permutation null has a standard deviation of 0.0378. At 80% power and $\alpha = .05$, the minimum detectable effect is ± 0.106, which means that the design could reliably detect only shifts of at least about a tenth of the session length, and smaller shifts could have been present without being seen. This is a limitation that we state plainly, because it is the kind of limitation that is easy to forget once a p-value of 0.985 is on the page. A large null is not the same as a small effect, and the primary test by itself cannot tell them apart. The equivalence analysis in the next subsection is the part of this section that does distinguish them, within the bounds we chose. Figure 2 shows the permutation null with the observed value marked.

We had drafted a long paragraph for the case in which T was far from 0.5. It would have discussed the mechanism by which early or late pitches are advantaged, the literature on fatigue among evaluators, and the possibility that the effect was real but concentrated in a subgroup. It is not needed. The observed mean is 0.0007 below its null expectation, and the test statistic sits where an average shuffle would have put it. Figure 3 compares the empirical distribution of the 60 winner positions with the uniform cumulative distribution function, which is the distribution that the null hypothesis implies for a single winner’s position.

The primary statistic was recomputed on several subsets, summarized in Figure 7. Dropping the multi-session teams gives a mean of 0.493, against an expected 0.516, with $p = 0.586$. Dropping the 2022 Solana session, whose schedule label is ambiguous, gives 0.508 with $p = 0.840$. Restricting to sessions with at least 15 pitches gives 0.534 with $p = 0.483$, and restricting to sessions with fewer than 15 gives 0.445 with $p = 0.372$. The three years taken separately give 0.487 for 2022 ($p = 0.843$), 0.546 for 2024 ($p = 0.495$) and 0.469 for 2025 ($p = 0.618$). Every subset has a confidence interval that contains 0.5. The largest point estimate, 0.546 for 2024, has a 95% interval from 0.441 to 0.662. That interval is wide enough to be consistent with a great deal, and the largest estimate is better described as the least precise than as the most informative.

7.3 Equivalence

A non-significant test is not evidence of absence, and the primary result is non-significant. We therefore also ran a two one-sided equivalence test (TOST) with bounds of ± 0.10 around 0.5, using a normal approximation with the permutation-null standard deviation (Schuirmann, 1987; Lakens, 2017). The equivalence test is significant at $p = 0.0043$. The corresponding 90% normal-approximation interval for T runs from 0.437 to 0.562, which lies inside the equivalence bounds.

On this basis we can affirmatively declare order to be practically irrelevant within ± 0.10. Any shift of the mean winner position larger than 0.10 is rejected. A shift of that size would move the average winner from the middle of a session to a point a tenth of the session length further along, and we are now in a position to say that this does not happen. The primary test could not say this. The equivalence test says it at the stated level, under the stated approximation, and we have tried to be careful about the phrase “practically irrelevant” by using it only to mean the thing we have just defined.

7.4 Secondary tests

The secondary family consists of six permutation tests, the two conditional logits and the two submission-ID analyses, all corrected together with the Holm procedure (Holm, 1979). Table 11 reports every test in the family. Each observed value is compared with its expectation under H_0, each raw p-value is reported, and the Holm-adjusted p-value follows.

Table 11: Secondary tests. Expected values are under the within-session permutation null. Holm p-values adjust across the family. The quadratic logit term is reported without a Holm-adjusted value.

Test	Observed	Expected	
Late half (60 winners)	33	30.01	
Terciles: early / middle / late	21 / 25 / 16	20.42 / 21.16 / 20.42	0.397 (sta
First slot wins	4	4.304	1.0 (one-sided uppe
Last slot wins	4	4.306	
Edge (first or last) wins	8	8.620	
First pitch after a break (17 slots)	2	1.402	
Conditional logit, linear: $\beta = -0.007$ (SE 0.427)	–	–	
Conditional logit, quadratic: $\beta_{\text{quad}} = -0.774$	–	–	
Slot vs submission ID: median $\rho = 0.24$, Stouffer $z = 7.04$	–	–	1.9
Winners’ submission-ID rank (mean)	0.534	0.500	

Only one row in the table is significant after correction, and we will come to it at the end of this section. Every other row is a non-result. Two notes on the family before we go through it. First, the late-half statistic counts 60 winners rather than 62, because the two winners who occupied the middle slot of an odd-sized session cannot be placed on either side of the midpoint. Second, the tests share winners and are therefore not independent of one another. The Holm procedure does not assume independence, and we report it on that basis.

The late-half share is 33 of 60, against 30.01 expected, which is a slight tilt toward the end of the session that is well within chance. The tercile statistic is 1.673 against its null, with $p = 0.397$. Figure 4 shows the observed and expected winners in each third. The early third holds 21 winners against 20.42 expected, so the early third is close to its expectation. The middle third holds 25 winners against 21.16 expected, and the late third 16 against 20.42. These two deviations, roughly four winners in each direction, are the ones a reader will be most tempted to explain. We regard them as noise. About four winners above or below expectation in one third of a sample of sixty-two is a difference that chance produces routinely, and we decline to build a mechanism on it.

The first and last slots each won exactly four times, against 4.304 and 4.306 expected. The raw p-values are 1.0, and the one-sided upper bound for the first slot is 0.648. Together the two edge slots account for eight wins against 8.620 expected. A test that the edges are over-represented has nothing to find, and we report that finding with the calm of a result that did not need to be hoped for. The first pitch after a break won twice in 17 eligible slots, against 1.402 expected, with $p = 0.643$. The break analysis uses only sessions that had a break, which is a subset of the sample, and we have not tried to enlarge it. The break is the one feature of the schedule that the wider literature on decision order would expect to matter (Danziger et al., 2011). The same literature has been criticised for treating case order as if it were random (Weinshall-Margel and Shapard, 2011), and that criticism applies here in a weak form, since breaks are not randomly assigned.

The conditional logit with sessions as strata (McFadden, 1974; Chamberlain, 1980) gives a linear coefficient of $\beta = -0.007$ with standard error 0.427 and $p = 0.987$. The corresponding odds ratio, last slot against first, is 0.993, with a 95% interval of 0.43 to 2.29. An odds ratio of 0.993 is the statistical equivalent of a shrug, and the interval is wide enough to contain both a strong preference for the early slot and a strong preference for the late one. The quadratic specification gives $\beta_{\text{quad}} = -0.774$ with $p = 0.625$. The negative sign indicates a slight inverted U, in which the middle of a session is marginally favoured. The p-value is nowhere near significance, so the inverted U is not a finding. It is a shape that a reader might trace by eye across Figure 4, and we would ask that reader to resist.

7.5 Testing the four hypotheses

The hypotheses we set out to test are four in number. We number them P0 to P3 in the order in which they are usually introduced in the serial-position literature, so that the null comes last, where it belongs. P0 is primacy, the prediction that earlier pitches win more often, which is the pattern that the serial-position curve of recall would lead one to expect (Murdock, 1962). P1 is recency, the mirror image, in which later pitches win more often, a pattern that has appeared in the judging literature for performances (Haan et al., 2005; Page and Page, 2010). P2 is the U-shape, in which both ends of the session are favoured, the pattern reported for wine tasting in which the first and last options are preferred (Mantonakis et al., 2009). P3 is the null, which

says that the slot does not matter.

Table 12 gives the verdicts. None of P0, P1 or P2 is supported by the evidence. Their predicted patterns are absent from the first-slot, last-slot, edge, tercile and linear-logit tests, and the quadratic logit points, if anywhere, toward an inverted U that is the opposite of P2. P3 is not rejected. We go further than that, because the equivalence test in the earlier subsection establishes that any shift in the mean winner position larger than 0.10 is rejected at the stated level. The verdict for P3 is therefore stronger than the usual "failed to reject" and we use a word that reflects it.

Table 12: Verdicts on the four hypotheses. The numbering follows the text: P0 primacy, P1 recency, P2 U-shape and P3 null. The key evidence column cites the tests in Table 11.

Hypothesis	Prediction	Key evidence	Verdict
P0: primacy	Early slots win more often	Early third 21 vs 20.42; first slot 4 vs 4.304; first after a break 2 vs 1.402	Not supported
P1: recency	Late slots win more often	Late third 16 vs 20.42; late half 33 of 60 vs 30.01 ($p = 0.503$); OR 0.993	Not supported
P2: U-shape	Both edges win more often	Edge wins 8 vs 8.620; quadratic $\beta_{\text{quad}} = -0.774$ ($p = 0.625$)	Not supported
P3: null	No effect of slot	Mean $T = 0.4994$ ($p = 0.985$); TOST $p = 0.0043$ within ± 0.10	Not rejected; equivalence established

7.6 First versus second place

The eleven ranked pairs allow one further comparison, which asks whether the first-place team pitched later than the second-place team in the same session. A sign test on the eleven pairs gives the first-place team later in four cases. That is $p = 0.549$ on a sign test, and the result is that the first-place team was no more likely to pitch later than the second-place team than a coin would be. Four out of eleven is a little below the even split, and a sample of eleven pairs has very little power to say anything about a difference of the size one might plausibly expect here. We therefore report the test as a null and do not attempt a second version of it on the 2024 and 2025 sessions, where no places are available to compare. Table 13 lists the eleven pairs, using the normalized positions from Table 9.

Table 13: The eleven ranked pairs. A pair counts toward the sign test when the first-place team pitched later than the second-place team.

Session	1st place x	2nd place x	First pitched later?
2025 SAP	0.0 (ASDA, slot 1 of 12)	0.364	No
2022 Algorand	0.571	0.286	Yes
2022 Bloomberg	0.435	0.304	Yes
2022 City of Munich	0.576	0.667	No
2022 Huawei	0.500	0.643	No
2022 Hubert Burda Media	0.700	1.000	No
2022 Interhyp	0.174	0.609	No
2022 Rohde & Schwarz	0.455	0.000	Yes
2022 Sixt	0.368	0.947	No
2022 Solana	0.500	0.000	Yes
2022 Wildtrack	0.222	0.778	No

The table contains one further pattern, and it is the one that most needs restraint. All ten first-place teams from the 2022 sessions pitched somewhere between normalized positions 0.17 and 0.70. None of them pitched first and none pitched last. This is a striking regularity. It is also precisely the sort of observation that invites a story: one could write that mid-session pitches win, supply a plausible reason (the middle of a session is, perhaps, when evaluators have settled into their rhythm), and submit the result. We did not write that story. The observation was found after we had split the data by place and by year, and no hypothesis about it was fixed in advance. It is the textbook forking path, in which a subgroup, a post hoc partition and a pattern discovered after the fact combine to produce a finding that does not survive a second sample (Simmons et al., 2011; Gelman and Loken, 2014). We report the pattern here, label it as a post hoc observation, and attach no inferential weight to it. If the reader wishes to believe that middle pitches win first places in 2022, we ask them to test the claim on the 2024 and 2025 sessions, where the sources do not provide the first places required to do so.

7.7 The overall winners

hackaTUM also names an overall winner in the afternoon, chosen by a jury of professors from among the challenge winners. Three such overall winners can be matched to a schedule slot. SixtCharge won the 2022 overall prize, pitching 19th of 20 in the Sixt session, at normalized position 0.947. Clarity - The Copilot for Journalists won the 2024 overall prize from slot 3 of 18 in the Hubert Burda Media session, at position 0.118. Vom Platzl won the 2025 overall prize from slot 38 of 45 in the City of Munich session, at position 0.841. Two of their positions, 0.841 and 0.947, lie near the end of the session, and one, 0.118, lies near the start.

We decline to fit a model to three points, though we were tempted. A line through three points is always a line, and a line through these three would have had a slope that we could have described with some confidence and interpreted with none. The three overall winners are reported as a descriptive footnote to the challenge winners, and they should be read as nothing more than three entries in Table 10 and Table 9 that happened to receive a second prize.

7.8 The one significant finding

Our single significant result is about the schedule, not about the pitches. Within each session, the Spearman rank correlation between a team's pitch slot and its Devpost submission ID is positive in most sessions (Spearman, 1904). The per-session coefficients range from -0.32 for 2022 Solana to 1.00 for 2022 Wildtrack, and the median is 0.24. The 2022 Wildtrack schedule is perfectly sorted by Devpost submission ID, a fact we record without further comment. Figure 5 plots slot against submission-ID rank for every session.

Combining the per-session correlations by Fisher's z transformation (Fisher, 1915) and then by Stouffer's method (Stouffer et al., 1949) gives a Stouffer z of 7.04, with $p = 1.9 \times 10^{-12}$. The Holm-adjusted p-value is 1.7×10^{-11}, which is below one in a million. In short, the schedule loosely follows the order in which teams created their Devpost submissions.

This is the only significant result in the family, and we report it without dramatic inflation. It is a finding about the organisers' scheduling, which is a process we did not study and about which we have no further information. It is plausible that the earliest submissions are scheduled first, and that the scheduler then works down the list. We note that the scheduler plainly avoided clashes. Of the 96 submissions that pitched in two or more sessions in the same year, none had two pitches at the same time. The minimum gap between a team's pitches was four minutes

and the median gap was 50 minutes. This is the behaviour of a scheduler that takes care, which is consistent with the order being deliberate.

The finding does not carry over to winning. The winners' mean submission-ID rank is 0.534 against an expected 0.500, which gives $p = 0.367$ and a Holm-adjusted value of 1. The confound check found no evidence that winners were systematically earlier or later submissions, so submission order does not predict winning. The schedule has an order, and the order does not appear to matter, which we regard as a fitting end to a section about order without consequence. The authors are aware that "the schedule is orderly, and the order is irrelevant" is exactly what one would write to be cheerful about a null result. A reader who recalls Weinshall-Margel and Shapard (2011), who argued that the case order in the parole study of Danziger et al. (2011) was not random, will recognise the problem. Here the ordering comes from a schedule rather than from a random assignment, and we make no causal claim beyond the correlation we measured.

Summary

This section contains one significant result, and it concerns the schedule rather than the pitches. The primary test does not reject the null. The equivalence test establishes that any shift in winner position larger than 0.10 is rejected at the stated level. Every other secondary test is a non-result after correction, and none of the three alternative hypotheses survives. We have reported everything, which is more than the sample deserved and less than the hackathon did.

8 Results II: The Tyranny of the Clock

> Time is an illusion. Lunchtime doubly so.
>
> Douglas Adams, *The Hitchhiker's Guide to the Galaxy*

Every analysis so far has measured a pitch by its *place in the queue*, the normalized position $x \in [0, 1]$. A sceptical reader (and, at a late stage of this project, a sceptical commissioning editor) may object that judges do not experience a queue. They experience a clock. A judge at 12:40 in a 45-team room has been listening for two and a half hours; a judge at 10:50 in a 7-team room is already finished. If fatigue, hunger or impatience matter (Danziger et al., 2011), they should accumulate in minutes, not in fractions of a session. This section therefore repeats the analysis in absolute time.[34]

8.1 Definitions

Every scheduled slot carries a start time. All sessions began at 10:15, with the exception of the 2022 City of Munich session, which began at 10:05. The latest scheduled pitch in the sample began at 13:11 (2025 City of Munich, slot 45 of 45). We define, for slot j of session s,

$$c_{sj} = \text{scheduled start time in minutes after midnight}, \qquad e_{sj} = c_{sj} - \min_{j'} c_{sj'}, \tag{40}$$

the *clock time* and the *elapsed time* of the pitch. The permutation null is unchanged: within each session, the two winners are a uniformly random pair of slots. Only the statistic changes.

[34] The analysis in this section was added after the main results were known, at the explicit request of a reader. It is exploratory in the strictest sense, it has its own Holm family, and the authors wish it to be known that they did not enjoy re-running the permutations.

8.2 The naive picture, and why it is wrong

Table 14 and Figure 6 divide the morning into half-hour windows. The raw win rate falls steeply over the morning: 14.2% of pitches between 10:15 and 10:44 won, against 9.6% between 10:45 and 11:14 and roughly 8–9% thereafter. A hurried reader would conclude that early pitches win, and might even publish the conclusion.

The conclusion would be wrong. Each session has exactly two winners regardless of size, so a pitch in a 7-team room has a $2/7 \approx 29\%$ chance of winning and a pitch in a 45-team room a $2/45 \approx 4\%$ chance. Small rooms finish early; only large rooms are still running at 12:30. The late-morning windows are therefore populated almost exclusively by pitches that never had much of a chance to begin with. The column "expected" in Table 14 computes, session by session, how many winners each window should contain if clock time were irrelevant. Against that benchmark the decline vanishes, and in the last two windows reverses: after 11:45 the observed win rate exceeds its expectation (12 winners against 9.3 expected).[35]

Table 14: Winners by half-hour window of scheduled start time. Expected counts are computed within sessions under H_0.

Window	Pitches	Winners	Expected	Raw win rate	Expected rate
10:15–10:44	190	27	26.05	14.2%	13.7%
10:45–11:14	156	15	18.04	9.6%	11.6%
11:15–11:44	101	8	8.59	7.9%	8.5%
11:45–12:14	78	7	5.93	9.0%	7.6%
12:15–13:15	57	5	3.39	8.8%	6.0%
Total	582	62	62.00	10.7%	10.7%

8.3 Tests

Table 15 reports the formal tests, each computed by within-session permutation with the same machinery as Section 6.

- **Mean winner clock time.** The average winner pitched at 11:02 (661.6 minutes after midnight). Under H_0 the expectation is 11:00 (659.6 minutes), with a null standard deviation of 3.8 minutes. The two-minute difference has $p = 0.589$.
- **Mean elapsed time.** Winners pitched on average 47.0 minutes into their session, against 44.9 expected ($p = 0.589$).
- **Half-hour windows.** The chi-square-type statistic over the five windows is 1.540 (permutation $p = 0.758$).
- **The lunch hypothesis.** Eleven sessions were still running at 12:00, containing 88 pitches scheduled at or after noon. These pitches produced 7 winners against 5.58 expected ($p = 0.604$). Hungry judges, if they exist, did not punish the hungry.
- **Conditional logit on elapsed hours.** Each additional hour into a session multiplies the odds of winning by 1.15 (95% CI 0.70–1.89, $p = 0.587$). The interval comfortably contains 1.

[35] This is a textbook instance of Simpson's paradox, and the authors were briefly delighted to have found one, before remembering that it does not change the answer.

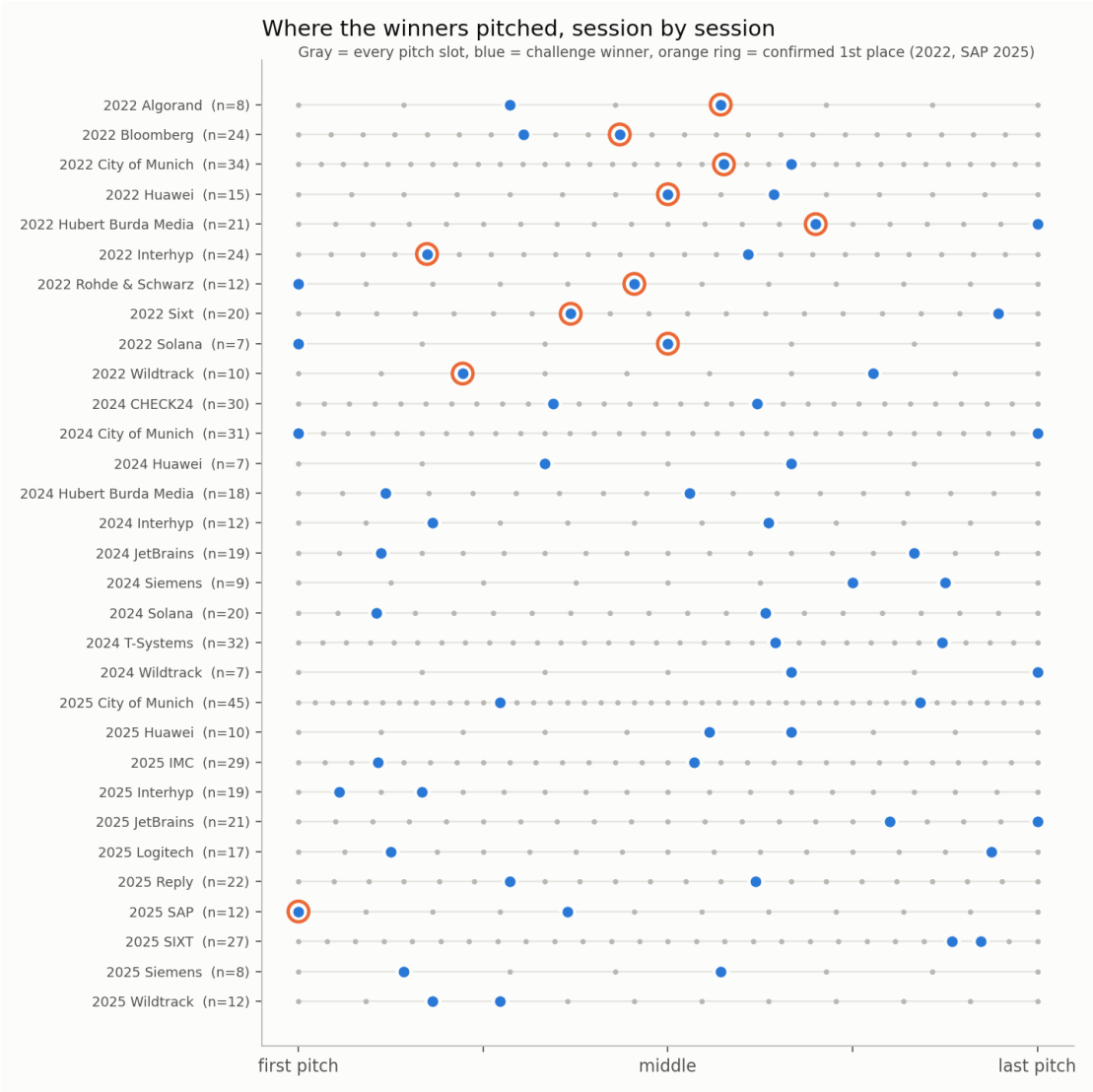


Figure 1: Every analysed session, one row each. Gray dots are pitch slots; blue dots are challenge winners; orange rings mark confirmed first places (all 2022 challenges, SAP 2025).

Table 15: Clock-time tests (exploratory; separate Holm family of five tests). All Holm-adjusted p-values equal 1.

Test	Observed	Expected under H_0	p
Mean winner clock time	11:02	11:00	0.589
Mean minutes into session	47.0	44.9	0.589
Half-hour windows (χ^2-type)	1.540	–	0.758
Winners at or after 12:00	7	5.58	0.604
Conditional logit, OR per hour	1.15	1	0.587

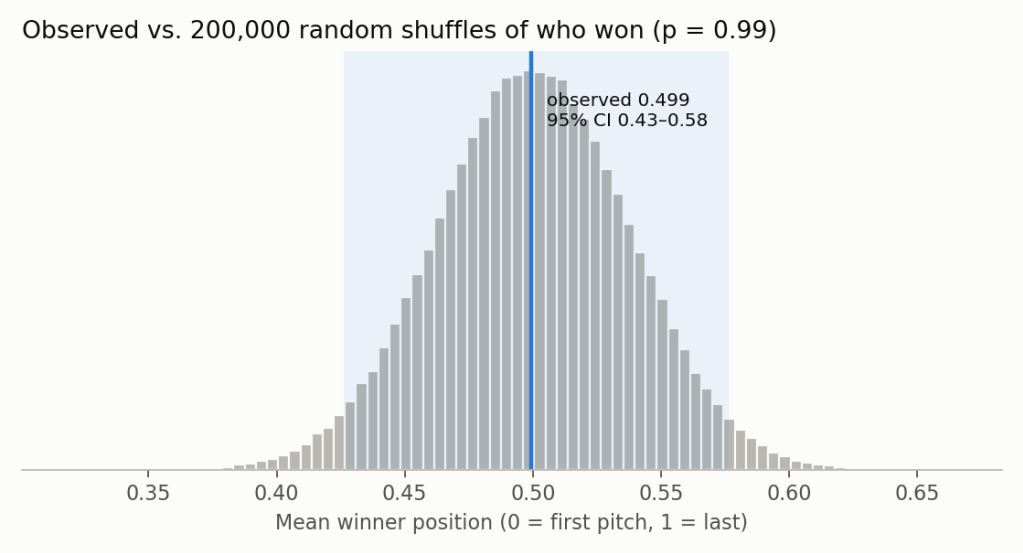


Figure 2: Permutation null distribution of the mean winner position (200,000 within-session reshuffles) with the observed value $T = 0.499$ and its 95% bootstrap interval.

8.4 Interpretation

Measured by the clock rather than by the queue, the answer is unchanged. There is no detectable advantage to pitching at 10:15 rather than 12:40, no detectable penalty for pitching while the judges are thinking about lunch, and no detectable fatigue accumulating over the morning. The one striking pattern in the raw data, the apparent decline in win rate over the morning, is entirely explained by the fact that small rooms finish early. We record this as the paper's second finding about scheduling and its second finding that scheduling does not matter.

9 Robustness, Sensitivity, and Existential Doubt

> We did not find the pitch that wins, and we were not looking very hard.
>
> Anonymous hackathon participant, 4:12 a.m.

A null result invites a specific kind of suspicion. The reader is entitled to ask whether the finding is an artefact of one session, one year, one judging regime, or one unfortunate row in one schedule PDF. This section addresses those suspicions in turn. We recompute the primary statistic on subsets, examine a data-quality incident that could in principle have mattered, consider non-random scheduling as a threat to the entire enterprise, mourn a negative-control comparison that we were unable to run, compare planned with realised order, re-code position in several alternative ways, and finally state, with some reluctance, what our confidence intervals do and do not allow us to say. We close with an itemised list of threats to validity in the software-engineering tradition, which we include because we have been told that a paper of this length is not complete without one.

Throughout, the primary quantity is the mean normalised position T of the 62 winners across the 31 sessions, 582 pitch slots in total, as defined in the Methods. Under the null hypothesis H_0, the expected value of T is 0.5001, the permutation null standard deviation is

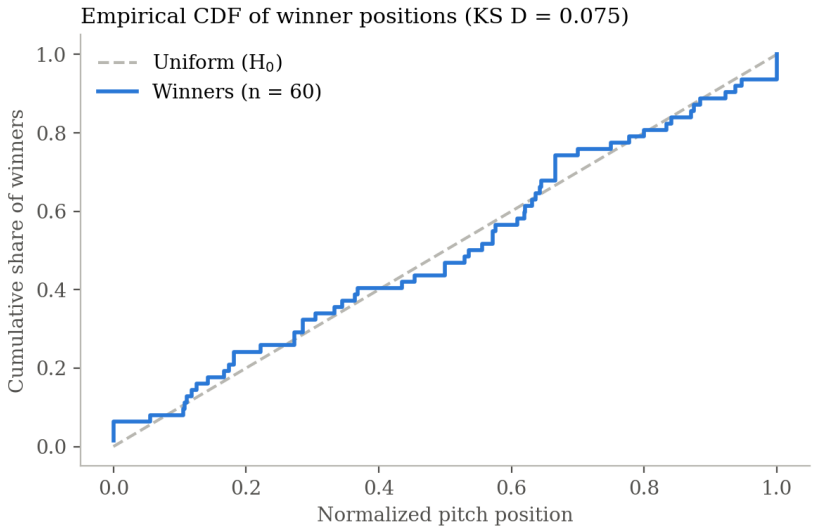


Figure 3: Empirical cumulative distribution of the 60 winner positions against the uniform distribution implied by H_0.

0.0378, and the observed value is 0.4994.[36]

9.1 Subset analyses

If the null result is driven by a few influential sessions, then removing those sessions should move the estimate substantially. We therefore recomputed the primary statistic on seven pre-specified subsets: four defined by session structure (dropping multi-session teams, dropping one session, and splitting by session size) and three defined by year. Table 16 reports the results, and Figure 7 displays them as a forest plot, which we consider the more honest of the two presentations because it makes the width of each interval visible at a glance.

Table 16: Robustness of the primary statistic (mean normalised winner position) across seven subsets. The 95% intervals are bootstrap intervals from resampling whole sessions, and p values are two-sided permutation values. Expected values under H_0 are shown where they differ from 0.5.

Subset	Sessions	Winners	Mean	95% CI	p
Drop multi-session teams	28	44	0.493 (expected 0.516)	0.409–0.578	0.586
Drop 2022 Solana	30	60	0.508	0.436–0.583	0.840
Sessions with ≥ 15 pitches	19	38	0.534	0.443–0.629	0.483
Sessions with < 15 pitches	12	24	0.445	0.330–0.569	0.372
2022	10	20	0.487	0.377–0.605	0.843
2024	10	20	0.546	0.441–0.662	0.495
2025	11	22	0.469	0.324–0.623	0.618

Dropping multi-session teams. Ninety-six submissions pitched in two or more sessions in the same year, and a reader might reasonably worry that these teams, who appear in several

[36]The expected value under H_0 is 0.5001 rather than exactly 0.5. The difference is 0.0001, which we mention so that no reviewer has to find it unaided, and which we do not discuss further.

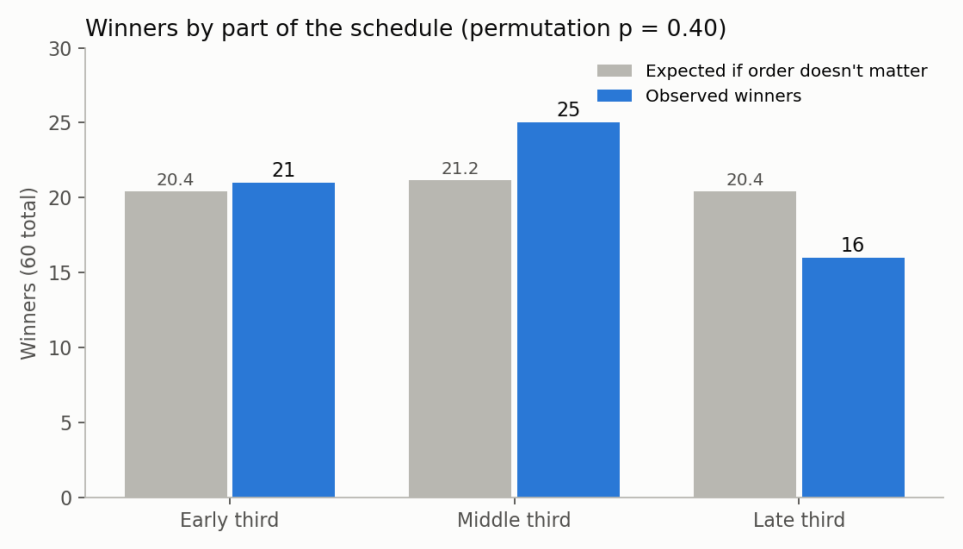


Figure 4: Observed versus expected winners by third of the schedule. Expected counts are computed session by session.

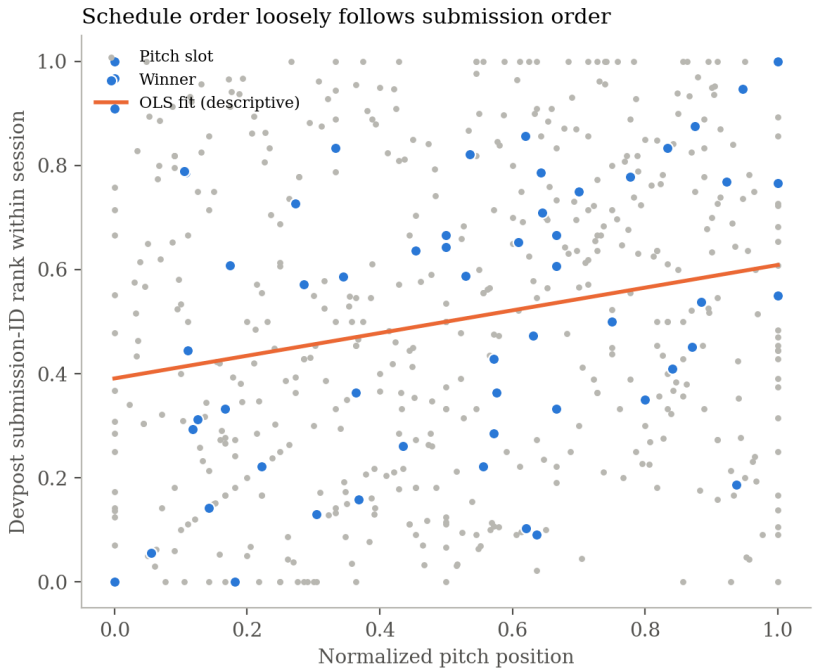


Figure 5: Normalized pitch position against the rank of each team's Devpost submission ID within its session. The schedule is not random; it is, however, irrelevant.

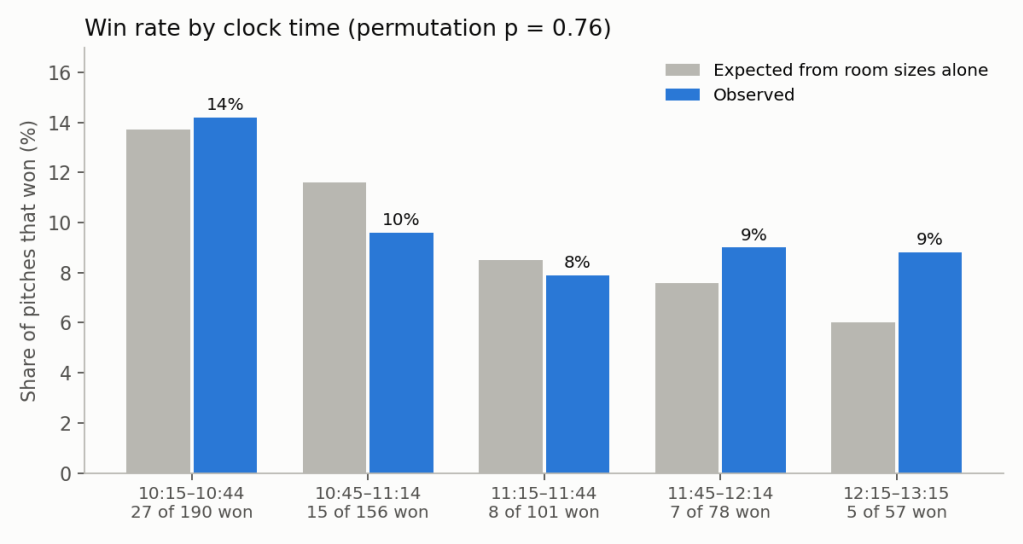


Figure 6: Win rate by clock time. Blue: observed share of pitches that won. Gray: the share expected if clock time were irrelevant, given which rooms were still running. The raw decline is a room-size artefact.

schedules and whose slots are therefore correlated with each other, distort the picture. Removing them leaves 28 sessions and 44 winners. The mean falls to 0.493, with interval 0.409–0.578 and $p = 0.586$.

The expected value for this subset is 0.516, not 0.5, and this is worth explaining at some length, since a careless reader might take the discrepancy as evidence of a bug. The explanation is mechanical. When a team's slots are removed from a session, the remaining slots in that session no longer form an evenly spaced grid on the unit interval. The surviving normalised positions are, in general, unevenly distributed, and their average need not be 0.5. The permutation null for this subset is therefore computed by reshuffling winner labels over the surviving slots only, and the expectation under that null is the average of the surviving normalised positions, which here happens to be 0.516. The observed mean of 0.493 is thus 0.023 below its own null expectation, which is a statement about the reshuffled subset and not about a shift from 0.5. We regard this as the correct comparison and the uncorrected comparison with 0.5 as a small and instructive trap. A reader who prefers the convention of centring on 0.5 may simply subtract the 0.016 gap between 0.516 and 0.5 from the expectation and proceed, although we would not recommend doing so without a note in the margin.

Dropping 2022 Solana. This subset is the sensitivity analysis for the schedule incident described in Section 9.2. Dropping the entire 2022 Solana session leaves 30 sessions and 60 winners, with mean 0.508, interval 0.436–0.583, and $p = 0.840$. The point estimate moves a little toward the middle of the interval, which is what one would expect when a session containing two winners at normalised positions 0.5 and 0 is removed, and the conclusion is unchanged.

Small versus large sessions. We split sessions at fifteen pitches, which is a threshold chosen for its roundness and for its approximate splitting of the sample. Sessions with at least 15 pitches (19 sessions, 38 winners) have mean 0.534, interval 0.443–0.629, and $p = 0.483$. Sessions with fewer than 15 pitches (12 sessions, 24 winners) have mean 0.445, interval 0.330–0.569, and $p = 0.372$. The two estimates lie on opposite sides of 0.5, and one might be tempted to read

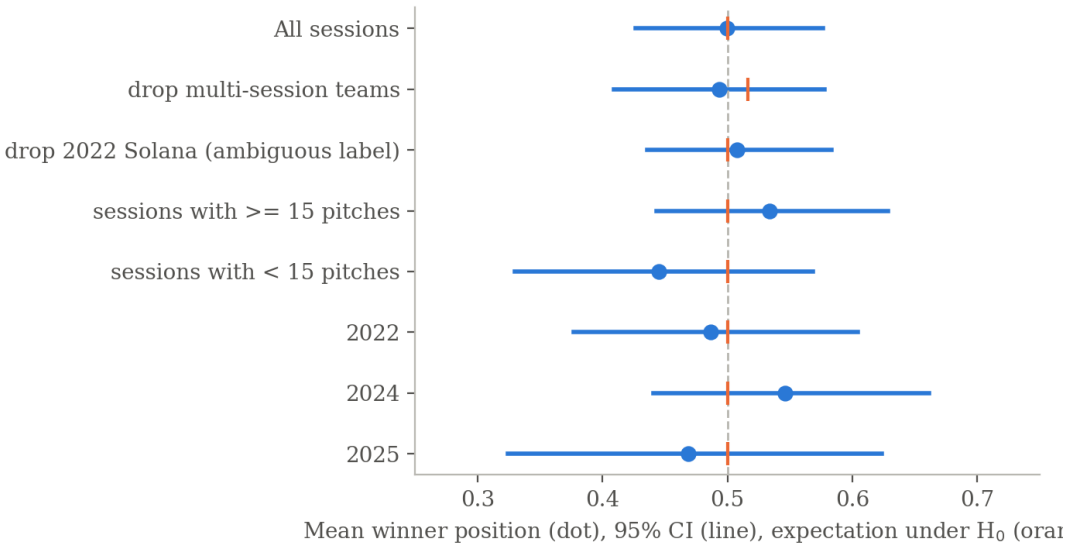


Figure 7: Robustness of the primary statistic across subsets.

a story into that. We decline. Both intervals contain 0.5, both p values are large, and the two estimates differ by an amount that is well within the width of either interval. Taken together they are consistent with the null in both halves of the sample.

Years. The 2022, 2024, and 2025 subsets have means 0.487, 0.546, and 0.469, respectively, with p values of 0.843, 0.495, and 0.618. The 2024 estimate is the largest of the three and the 2025 estimate the smallest, but every interval contains 0.5 and none of the three differences is distinguishable from the others. The year-level analysis is the most natural robustness check for a schedule that might have changed over time, and it found no evidence that it did, in either direction. The 2022 subset also contains the first-place pattern from the ranked-pairs comparison, which is consistent with the null in the same way and is therefore a source of considerable disappointment to anyone who had hoped for a story.

Across all seven subsets, the mean winner position lies between 0.445 and 0.546. Every interval contains 0.5. Every p value is above 0.35. We take this as a reasonably strong indication that the primary result is not an artefact of any single partition of the data, while acknowledging that seven subsets drawn from one sample are not seven independent experiments, and that a reader who wishes to look for a signal in them will find one with some patience.

9.2 The unidrei sensitivity analysis

The 2022 Solana schedule labels two different submissions as “unidrei”. This is, by a wide margin, the most embarrassing data quality issue in the project, since it means that a name we use as an identifier was in fact not one. The first of the two entries, in slot 1 of that session, links to the project S-Moody, which is one of the winners. We trust the URL over the label, on the grounds that URLs are less likely to be typed by someone in a hurry on a Sunday morning.

The sensitivity question is simple. If the mislabelled entry were resolved in the wrong direction, would the conclusion change? The answer is that the question is answered by the subset analysis in Section 9.1, since dropping the entire 2022 Solana session removes the problematic entry along with everything else in that session. The result is the one reported above: mean

0.508, interval 0.436–0.583, $p = 0.840$. Dropping the session changes the conclusion nowhere. The point estimate moves from 0.4994 to 0.508, which is a shift of 0.009 and is, we think, the appropriate amount of worry for a single mislabelled row in a sample of 582 slots.

9.3 Non-random scheduling as a threat to validity

The central assumption of the permutation test is that, under H_0, the winners are a uniformly random two-subset of the slots in each session. This assumption would be violated if the schedule itself were not random with respect to something that also predicts winning. The most natural candidate, and the one we were most afraid of, is the order in which teams submitted to Devpost.

The worry has a respectable literature. In a well-known study of parole decisions, Danziger et al. (2011) reported that parole decisions declined before breaks, and that this pattern was tied to the ordering of cases. A critique by Weinshall-Margel and Shapard (2011) pointed out that case order was not random, and that the apparent effect could therefore reflect the sequence in which cases were assigned rather than anything the decision-makers were doing. We find the parallel unsettling. Our pitch schedules are also not random; they were assembled by people, and we cannot rule out that the assembly process correlated with something that, in turn, correlated with winning.

We therefore checked the one thing we could check directly. The within-session Spearman correlation between slot and Devpost submission ID, combined across sessions with Fisher z and Stouffer's method, gives a median ρ of 0.24, a Stouffer z of 7.04, a raw p value of 1.9×10^{-12}, and a Holm-adjusted p value of 1.7×10^{-11}. The schedule does loosely follow the order in which teams created their Devpost submissions. This is the only significant result in our entire battery of tests. Per-session ρ ranges from -0.32 (2022 Solana) to 1.00 (2022 Wildtrack), so the degree of alignment is not uniform; Figure 5 shows the pattern.

The question that matters for the null is not whether the schedule follows submission order, which it does, but whether submission order predicts winning, which would be the route by which a non-random schedule could bias the result. Here the evidence is reassuring. The winners' submission-ID rank has a mean of 0.534 against an expectation of 0.500, with $p = 0.367$ and Holm-adjusted $p = 1$. Winners did not systematically submit earlier or later than the other teams in their sessions. A schedule that loosely follows submission order can therefore only bias the winner positions if submission order is itself related to quality, and our confound check finds no evidence of that relation. The parallel with Danziger et al. (2011) and Weinshall-Margel and Shapard (2011) is real, but in this case the mechanism that would make the scheduling non-random does not appear to feed through to the outcome we are measuring.

We add one further observation that we consider reassuring in the spirit of the parallel. The first pitch after a break is a natural place for a reviewer's attention to drift, in the manner that the parole literature describes. Across the sample there are 17 such slots, of which 2 won, against an expectation of 1.402. The raw p value is 0.643, and the Holm-adjusted value is 1. This is a small sample and a weak test, but it does not show the pattern that the parole literature would predict.A weaker reading of Weinshall-Margel and Shapard (2011) would be that any analysis of ordered decisions should check whether the ordering is random. We have checked. It is not.

9.4 The negative-control comparison that could not be run

A careful study of order effects would, ideally, include a negative control: a set of sessions in which the outcome is known to be determined by something other than pitch order, so that any order effect found there must be an artefact of the procedure rather than a property of the world. For a hackathon, the natural candidates are sessions with objective scoring, where a leaderboard or a load test or a puzzle score settles the outcome without any human being in the room. If winners in such sessions showed no position pattern, that would be reassuring about the method. If they showed a pattern, that would be alarming about the method. Either outcome would have been informative.

We could not run this comparison. Only one session in the sample is confirmed to be objectively scored: CHECK24 in 2024, which used an automated load-test leaderboard. The other 2024 candidate, Huawei 2024, was mixed, combining CTF puzzle points with a manual grade, and so is not a clean objective session. Every other session is judged by pitch or demonstration, or has criteria that we were unable to establish. A negative-control comparison needs more than one objective session before it can say anything, and we have, at most, one and a partial one.

We therefore report the available fragments with the sadness they deserve. Four winners are involved. The two CHECK24 2024 winners occupied slots 11 and 19 of 30. The two Huawei 2024 winners occupied slots 3 and 5 of 7. Four observations is not a sample; it is a list. We have nonetheless reproduced the positions in the text because a reader will want them, and because the one thing a negative control can tell us, if it had been run, is whether our method would have manufactured a pattern out of exactly this kind of list.

We mourn the comparison formally. The negative control is the single most useful check we did not perform, and we would encourage any future study to find more objectively judged hackathon sessions, possibly by persuading partners to adopt leaderboards, and to repeat this analysis with a control of adequate size.

9.5 Planned versus realised order

The analysis uses the scheduled order of pitches, as recorded in the official pitch schedule PDFs. What the audience and the judges experienced was the realised order, which may differ from the scheduled order in ways that we did not observe. Teams may have been late, swapped, withdrawn, or replaced, and sessions run over or under their allotted slots. The sources we used do not record a realised order, and we therefore cannot quantify the difference between what was planned and what happened.

This matters for interpretation in one specific way. Any pattern in the planned order that is washed out by reshuffling in the realised order would show up in our data as a null, and we would have no way of distinguishing that from a genuine absence of effect. The null result is therefore a statement about the planned order. Whether the realised order differed in a way that would have produced an effect is a question for which we have no data. We regard this as a real limitation and as the most plausible route by which a future study might find a result that we did not.

No-shows raise the same issue in a more dramatic form. A team that was scheduled and did not appear is, in our data, indistinguishable from a team that was scheduled and pitched. The consequence for winner positions is not obvious, because a no-show who did not win would simply be absent from the outcome, and a no-show who won would be an extraordinary and

rather uncomfortable event that we have no record of. The sources record scheduled slots and winners, and they do not record attendance. We have therefore treated every scheduled slot as a pitch, which is an assumption we are prepared to defend in a seminar but not in a courtroom.

9.6 Alternative position codings

The primary statistic uses normalised position $x = (j - 1)/(n_s - 1)$, which maps the first slot of each session to 0 and the last to 1. This is a natural choice, but it is not the only one, and a reader who suspects that a different coding would have told a different story is entitled to check. We checked three alternatives.

The first is the rank percentile. Rather than normalising by the number of slots, one can convert each slot to its percentile rank within its session. Because the slots within a session are evenly spaced and carry no ties, this coding is, for practical purposes, a monotone transformation of the normalised position, and the qualitative conclusions are the same. We report it for completeness and note that it adds no new information, which is a property shared by many of the codings in the applied literature.

The second is the halves. Splitting each session into an early and a late half, with the middle slot of odd-sized sessions excluded, gives 33 winners in the late half out of 60 counted, against an expectation of 30.01. The raw p value is 0.503, and the Holm-adjusted value is 1. The late-half share is slightly above its expectation, and not by an amount that any reasonable test would call evidence.

The third is the terciles. Splitting each session into early, middle, and late thirds gives observed counts of 21, 25, and 16, against expectations of 20.42, 21.16, and 20.42. The chi-square-type statistic is 1.673, with raw $p = 0.397$ and Holm-adjusted $p = 1$. The middle third is the most populated, but the pattern is not distinguishable from what the null would produce.

All three codings agree. The conclusion is not an artefact of the choice of coding, and we would be surprised if a reader found a fourth coding that disagreed with the other three without also finding that it was cheating.

9.7 Interpretive humility

We now state, as carefully as we can, what our results allow us to say and what they do not. The primary point estimate is 0.4994 with a 95% bootstrap interval of $[0.426, 0.576]$, obtained by resampling whole sessions. The permutation p value is 0.985. The minimum detectable effect at 80% power and $\alpha = .05$ is ± 0.106, following the conventional calculation (Cohen, 1988). Read naively, these numbers might be taken to say that we have excluded everything, and they do not say that.

What the interval excludes is a shift of the mean winner position below 0.426 or above 0.576. What it does not exclude is any shift that lies inside that range. A true mean of 0.45, or 0.55, is entirely compatible with our data, and so is a true mean of 0.50. The interval is a statement about the range of values our data are consistent with, and it is wide enough to contain effects that a participant would describe as “a bit of an edge” and a partner would describe as “unacceptable, please reschedule”.

The equivalence test is more reassuring, and we have used it precisely because it asks the question the reader cares about. A two one-sided test (TOST) (Schuirmann, 1987) with bounds of ± 0.10 around 0.5, using a normal approximation with the permutation-null standard deviation, gives $p = 0.0043$. On that basis we conclude that any shift in the mean winner position

larger than 0.10 in either direction is rejected. This is a stronger claim than "no significant effect", and it is the claim we would make if forced to make one. The equivalence bound of ± 0.10 is a choice, and a reader may reasonably prefer a different bound, but the conclusion at ± 0.10 is firm.[37]

The conditional logit tells the same story from another angle. Its linear coefficient is $\beta = -0.007$ with standard error 0.427 and $p = 0.987$, corresponding to an odds ratio of last versus first slot of 0.993, with a 95% interval of 0.43–2.29. The interval contains 1 comfortably, and it is wide. It is compatible with the last slot having less than half the odds of the first, and also with it having more than twice the odds. A reader who wants to argue that the pitch order has a modest effect in one direction or the other has not been ruled out by our data. The most we can say is that a large effect is unlikely, and that a modest effect remains possible, and we would ask the reader to hold both of those statements in mind at once. The quadratic specification, with $\beta_{\text{quad}} = -0.774$ and $p = 0.625$, suggests a slight inverted-U shape, in which the middle of the schedule does marginally better, but the evidence for that shape is weak and we do not interpret it.

We remind the reader, in the spirit of Wasserstein and Lazar (2016), that a p value is not a probability that the null is true, that a confidence interval is not a range that contains the true value with certainty, and that a non-significant result is not evidence of absence. We have tried to be careful about each of these. We have also been mindful of the degrees of freedom available to any analysis of this kind (Simmons et al., 2011), and we note that the analysis was pre-declared, that the seed was fixed at 2025, and that the reported results are the ones the pre-declared plan produced.

9.8 Threats to validity

We follow the software-engineering tradition of organising threats to validity into four classes: internal, external, construct, and conclusion validity. Each entry is intended to be read as a serious caveat, and each is also, unavoidably, slightly silly. We have not been able to separate those two properties, and we have decided not to try.

- **Internal validity.** The pitch schedules were assembled by a process we did not observe. Any relationship between that process and the outcome would bias the estimates in ways we cannot fully characterise. We checked the one relationship we could check (Section 9.3). The schedule may have been designed by a careful committee, or it may have been produced by a person with a spreadsheet and a deadline. We cannot tell these apart from the data, and the difference is, for our purposes, important.

- **Internal validity (data).** The unidrei incident (Section 9.2) is, in the strict sense, a threat to internal validity, and it is also the sort of thing that happens to everyone who has ever worked with a schedule. Dropping the affected session does not change the conclusion.

- **External validity.** Results may not generalise to hackathons held in other cities, in other months, under other judging regimes, or on other planets. We have been asked to specify the last of these, and we specify it here: results may not generalise to hackathons held on Mars, where the Sunday-morning pitch session would, in any case, be subject to a light-time

[37] The equivalence bound was set in advance, and it is worth noting that a bound of ± 0.10 is also a bound on the kind of effect that a partner might plausibly have been worried about. We have not been able to find a partner who was worried about it, which is itself a finding.

delay, which would make the scheduling considerations of Section 9.5 considerably more interesting.

- **Construct validity.** The outcome is winning, and winning is a proxy for quality, which is itself a proxy for what the judges believed about the project. The judging basis varies across sessions, and in most of them the criteria are unknown. A null result on winning is therefore a null result on a construct that is only partly the thing we set out to study. The answer to what a "good" pitch is, for the purposes of this paper, is 42 (Adams, 1979).
- **Construct validity (negative control).** We lack a negative control (Section 9.4). The absence of a control is a threat to construct validity in the sense that we cannot say our method would have detected an effect had one been present in a setting where we knew it was. We regret this and have said so formally.
- **Conclusion validity.** With 31 sessions, the test has a minimum detectable effect of ± 0.106. Effects smaller than that are unlikely to be detected by the primary test, although equivalence at ± 0.10 is established (Section 9.7). Conclusion validity is threatened by the many secondary tests, which we have corrected with Holm's procedure (Holm, 1979), and by the subset analyses, which we have reported in full and which a reader may consider a garden of forking paths. We consider the forking paths to be narrow and well lit.
- **Conclusion validity (reproducibility).** The analysis uses Python with NumPy, SciPy, statsmodels and Matplotlib, with a fixed random seed of 2025. Code is in the appendix (D), and the per-session schedules and winner tables are in the appendix (A). We believe the results are reproducible by anyone with the appendix and a sufficiently patient machine.

We conclude this section, and the threats to validity with it, with a statement of what we believe. The mean position of winners at hackaTUM does not depend on the order in which teams pitched, to a precision of about ± 0.10 on the scale of normalised position. The schedule is not random with respect to submission order, but submission order does not predict winning. The negative control we would have liked was not available. The reader who wishes to conclude from this that it does not matter whether one pitches early or late is, within the limits set out above, entitled to do so. The reader who wishes to conclude that it does not matter at all will find the reasons for caution set out above.

10 Discussion

> The demo works on my laptop.
>
> Anonymous hackathon participant, 4:12 a.m.

This section has eight parts. Section 10.1 summarizes the findings, in three forms of decreasing length. Section 10.2 asks why hackathon adjudication appears immune to serial position effects that have been documented in other judged contests, and offers hypotheses which we label, for the avoidance of doubt, as speculation. Section 10.3 places the results in the literature. Sections 10.4 and 10.5 give advice to teams and organizers. Section 10.6 proposes a pre-registered design, Section 10.7 lists future work, and Section 10.8 recaps the limitations.[38]

[38] Brevity in the limitations is itself a form of order effect. They were placed last and are therefore the least likely to be read in full.

10.1 Summary of findings

Across 31 sessions, 582 pitch slots and 62 winners from hackaTUM 2022, 2024 and 2025, we find no detectable relationship between the position of a team in the Sunday-morning pitch order and its probability of winning a partner challenge. The mean normalized position of the winners is 0.4994, against 0.5001 expected under the null hypothesis of uniform random placement; the 95% bootstrap interval, from 20,000 whole-session resamples, is $[0.426, 0.576]$; and the two-sided permutation p-value, from 200,000 within-session reshuffles, is 0.985. The minimum detectable effect at 80% power is ± 0.106. A TOST equivalence test with bounds of ± 0.10 around 0.5 gives $p = 0.0043$, so any shift larger than 0.10 is rejected. The secondary tests, all permutation-based and Holm-corrected, agree: no tercile, edge-slot, first-slot, last-slot, late-half or after-break pattern survives, and the conditional logit gives an odds ratio of 0.993 for last versus first (95% CI 0.43 to 2.29). The only significant result is that the schedule loosely follows the order of Devpost submission identifiers (median Spearman $\rho = 0.24$, Holm $p = 1.7 \times 10^{-11}$), which is a finding about scheduling rather than judging. Table 17 collects the headline numbers.[39]

Table 17: Headline results, for readers who skipped the summary paragraph.

Quantity	Value	Reference value
Mean normalized winner position	0.4994	0.5001 (expected)
95% bootstrap CI	$[0.426, 0.576]$	–
Permutation p (two-sided)	0.985	–
Minimum detectable effect (80% power)	± 0.106	–
TOST p, bounds ± 0.10	0.0043	equivalence established
Conditional logit, OR (last vs first)	0.993	95% CI 0.43–2.29
Slot vs submission ID (Holm p)	1.7×10^{-11}	the only significant result

In one sentence: the time at which your team pitches on Sunday morning has no effect at hackaTUM that our data can detect.

No.

10.2 Why might hackathon judging be immune to order effects?

The literature on serial position effects in judged contests is large and consistent. Later performers in Eurovision score higher than earlier ones (Bruine de Bruin, 2005; Haan et al., 2005). Later singers in Idol-style television competitions do better (Page and Page, 2010). Figure-skating juries show a similar pattern (Bruine de Bruin, 2006), the Queen Elisabeth music competition shows order and day effects on rankings (Glejser and Heyndels, 2001; Flôres and Ginsburgh, 1996), and song contests more generally show order sensitivity (Antipov and Pokryshevskaya, 2017). The mechanism most often invoked is belief adjustment, in which judges' evolving impression of the field is most strongly shaped by what came last (Hogarth and Einhorn, 1992).

Against this background, the absence of an order effect at hackaTUM is surprising. We offer the following hypotheses as to why hackathon judging might be immune. None has been tested. Each is a plausible story consistent with the data we have, and each would require a separate study.

[39] The 2022 Solana schedule also contains the so-called "unidrei incident", in which two submissions share a label. We trust the submission URL. Dropping the session changes nothing.

Speculation 1: session size. Hackathon pitch sessions are small, with between 7 and 45 pitches. A long televised contest is a continuous run of performances, and it is plausible that a judge who sees seven pitches in one sitting forms a stable view of each, whereas a judge who sees 45 forms a view more strongly shaped by the middle and the end. The data do not clearly support this. Sessions with fewer than 15 pitches show a mean normalized winner position of 0.445 (95% CI 0.330 to 0.569, $p = 0.372$), and sessions with 15 or more show 0.534 (95% CI 0.443 to 0.629, $p = 0.483$). Both are compatible with no effect and with each other.

Speculation 2: written notes and rubrics. Many hackathon judges, as far as we can establish, work from scoring sheets. A judge who has written "working demo: yes" beside a team's name at 10:19 is not, at 12:40, relying on a memory of how the pitch felt. The written record is an external memory that may protect against recency weighting. We find this hypothesis appealing mainly because it is the sort of practice that could be checked, and that few hackathons have checked.

Speculation 3: outcomes driven by working demos and code. A hackathon project either runs or does not. A team whose demo works on the judge's own laptop has been evaluated by a process that does not depend on the order of the pitches. In this view the judge is partly checking whether a link opens and a button does something. Our data cannot separate this from the next hypothesis, since most sessions have unknown criteria. Only the 2024 CHECK24 leaderboard is confirmed to be objectively scored, and that session is too small to serve as a negative control.

Speculation 4: coarse outcomes. Each partner selects two winners per session. A procedure that picks two items from as many as 45 has, by construction, less capacity to reveal a systematic bias than one that ranks them all. Order effects in the judges' underlying impressions could be real but too small to move the top two places, which are plausibly decided by a few strong projects competing for the same positions. Under this hypothesis, full rankings would show more order sensitivity than winner labels do. We have no full rankings, so this cannot be tested here.

Speculation 5: end-of-sequence evaluation. In most sessions, judges deliberate after all pitches have been delivered. In the terms of the belief-adjustment model, this is end-of-sequence evaluation (Hogarth and Einhorn, 1992): the judges survey a complete set of impressions rather than updating one as each pitch arrives. Recency then becomes a property of the final few impressions, and its weight depends on how the judges choose to deliberate. We apply this model with more confidence than it deserves, since we have not tested whether deliberation behaves as the model would suggest.

Speculation 6: variance in project quality dwarfs any order effect. This is probably the most likely explanation and the least satisfying. Projects differ enormously in quality, scope and polish. The variance attributable to these differences is likely to exceed any plausible order effect by a wide margin. An effect of a size visible in Eurovision would be a rounding error beside the gap between a team with a working deployment and a team whose demonstration is a slide reading "coming soon". In variance this large, a small shift in judges' attention is not detected so much as drowned.

These explanations are not mutually exclusive, and the honest answer is probably a mixture. Table 18 lists the evidence each would need.

Table 18: Speculative explanations for the absence of an order effect, with the evidence each would require. None has been tested.

Explanation	Evidence that would be needed
Session size (7–45 pitches)	Order effects estimated separately by session size, over a wider range of sizes
Written notes and rubrics	Access to scoring sheets and their timestamps
Working demos and code	Judgments made blind to the pitch, or a comparison of demo and pitch scores
Coarse outcomes (two winners)	Full rankings of all teams in each session
End-of-sequence evaluation	Recordings of deliberations, or timestamps of scores
Variance dwarfs order	A variance decomposition that includes project quality, which we do not possess

Before leaving the question of mechanism, we record three observations that the hypotheses above must accommodate. First, the robustness analysis is reassuring in a boring way. Dropping teams that pitched in more than one session leaves 28 sessions and 44 winners, with a mean of 0.493 (expected 0.516, 95% CI 0.409 to 0.578, $p = 0.586$). Dropping the 2022 Solana session leaves a mean of 0.508 ($p = 0.840$). Splitting by year gives 0.487 for 2022, 0.546 for 2024 and 0.469 for 2025, none of which departs from chance. Second, the ranked pairs do not resolve the matter. First-place teams pitched later than second-place teams in 4 of 11 ranked pairs, with a sign test p-value of 0.549. In 2022 all ten first-place teams pitched between normalized positions 0.17 and 0.70, which is a pattern that might be taken as evidence for a middle-of-the-schedule advantage, were it not for the fact that ten is a small number and that the sample contains no reason to expect a middle advantage in particular. Third, the overall winners are descriptively diverting. In 2022, SixtCharge pitched 19th of 20 in the Sixt session. In 2024, Clarity – The Copilot for Journalists pitched 3rd of 18 in the Hubert Burda Media session. In 2025, Vom Platzl pitched 38th of 45 in the City of Munich session, which was also the largest session in the sample, running in four-minute slots from 10:15 to 13:11.[40]

10.3 Relation to the literature

Bruine de Bruin's studies of Eurovision and figure-skating juries (Bruine de Bruin, 2005, 2006) document a tendency for later performers to score higher. Our results do not contradict this, since they concern a different kind of contest and a different kind of judgment, but they suggest that the size of such tendencies is not uniform across contexts. The Eurovision analysis of Haan et al. (2005) points the same way, and the Idol study of Page and Page (2010) extends the pattern to individual performers. Song contests more broadly are examined by Antipov and Pokryshevskaya (2017). Taken together, these papers establish order sensitivity in settings where judges watch long sequences of performances, often in a sequence fixed in advance by a

[40] In 2025, the two SIXT winners pitched back to back, in slots 24 and 25 of 27. We mention this only because it is the kind of fact that a reader will remember, and the authors would like to be remembered for something.

producer.

The Queen Elisabeth studies (Glejser and Heyndels, 2001; Flôres and Ginsburgh, 1996) are closer to our setting, because they concern a judged competition with a fixed schedule. Day and position within the day affected rankings there. The metrics and sample structures differ from ours, so effect sizes cannot be compared directly, and a music competition is not a hackathon.

The older literature on serial position is broader still. The serial position curve in free recall (Murdock, 1962) shows primacy and recency, and primacy in impression formation was documented by Asch (1946) and Anderson (1965). Ballot-order effects have been studied by Miller and Krosnick (1998) and Ho and Imai (2008), the latter being the closest analogue to our organizer recommendations. Wine tasters prefer the first and last options they are given (Mantonakis et al., 2009). These findings concern memory, impression and choice rather than pitch evaluation, and we cite them to acknowledge the breadth of the phenomenon rather than as a theory of hackathon judging.

The most directly relevant pair of papers is the parole study of Danziger et al. (2011) and the critique of Weinshall-Margel and Shapard (2011). Danziger and colleagues reported that parole decisions decline before breaks. The critique in Weinshall-Margel and Shapard (2011) argues that case order is not random, so the pattern may reflect something other than the decision-makers. The same objection applies to the present study. Our pitch schedules were not randomized, and as shown above they are loosely related to submission order. A null result for pitch order is therefore not a clean null, because the order is not a clean variable.[41]

The ego-depletion literature (Baumeister et al., 1998) and its failed replication (Hagger et al., 2016) remind us that an intuitive, widely cited result can fail when examined carefully. We do not claim the order-effect literature is similarly fragile. We do note that the case for pre-registration, developed in Section 10.6, is strengthened by such examples.

10.4 Practical recommendations for hackathon teams

Our primary audience, we assume, consists of teams about to pitch who have read this far in the hope of an advantage. We are sorry to report that the data do not offer one. The position of your pitch does not, on the evidence available, affect your chances of winning. The following recommendations are nonetheless consistent with the data, and some may be of practical use.

1. Do not lobby the organizers for a slot. Our data give no indication that the slot matters, and the organizers have heard every argument for the first slot at least once.

2. Use the time you would spend worrying about your slot to fix the demo. The demo is the one thing the data hint might matter, and it is the one thing you control.

3. If you are scheduled first, you are in the company of the 2025 SAP winner, ASDA, which pitched in slot 1 of 12 and took first place. First-slot pitches have won four times in our sample.

4. If you are scheduled last, you are in the company of the 2025 JetBrains winner, which pitched in slot 21 of 21 and won. Last-slot pitches have also won four times.

5. If you are scheduled in the middle, you are in the company of the 2022 first-place teams, all ten of which pitched between normalized positions 0.17 and 0.70. None pitched first and

[41] Only 17 slots are first pitches after a break, and 2 of them won. This is too few to say anything about breaks, and we decline to try.

none pitched last. We do not know what this means and advise against drawing conclusions from it.

6. If you are scheduled first after a break, you are one of 17 such slots, of which 2 won, against 1.402 expected. The group is small and the difference is not significant. We nonetheless recommend that you feel encouraged.

7. Bring your own HDMI adapter. Adapters are unaffected by pitch order, and a failed connection is more likely to end your pitch than any position in the schedule.

8. Submit to Devpost early. The schedule loosely follows submission order (median $\rho = 0.24$), so the most reliable lever available to you may be the submit button, pressed sooner rather than later.

9. Do not attempt to be in two places at once. Ninety-six submissions pitched in two or more sessions in the same year, and none had overlapping pitches. The minimum gap was 4 minutes and the median 50 minutes. The scheduler appears to have anticipated this, and you should extend it the same courtesy.

10. Do not treat the 2024 City of Munich session as a template. Its first pitch, Memory Lanes, and its last pitch both won. That is interesting, but it is one session out of thirty-one, and Memory Lanes was also the popular-vote winner, which may be the more relevant fact.

11. Do not rename your project in the hour before pitching. Our data contain no such cases, but we regard the phenomenon discussed in Section 10.7 as real enough to warrant caution.

12. Do not plan your strategy around sleep. The depletion literature does not support it (Baumeister et al., 1998; Hagger et al., 2016), and it is difficult to rest on a folding chair at 4 a.m. while a build compiles.

13. Bring a backup of your pitch on a second device, a recording of your demo, and a phone as a backup connection. None of these is an order effect, and all of them are more likely than your slot to determine how your pitch goes.

14. Remember that the afternoon jury chooses an overall winner from the challenge winners. Its deliberations are not public, and we have not attempted to observe them.

15. Accept that your position in the schedule is, statistically, indistinguishable from anyone else's. This is the most useful thing we can tell you, and the hardest to act on.

10.5 Recommendations for organizers

We address the organizers of hackaTUM, and of hackathons generally, with some caution, since their task is difficult and most of the difficulty comes from the number of people who want to pitch at 10:15 on a Sunday. We recommend that pitch order be randomized and that the seed of the randomization be published, so that anyone may verify the order. Both cost almost nothing, and the second benefit is more important than the first.

Randomized, published orders would turn the event into a natural experiment. Each session would be a small randomized trial of position, and accumulated trials across years would answer the present question with a precision the non-randomized design cannot reach. The analogous argument for ballot order is made by Ho and Imai (2008), and the reasoning carries over directly.

We recognize that organizers have many other things to randomize, and that pitch order may not be the first priority. We ask that it appear on the list.

We also recommend that organizers keep Devpost submission times alongside the schedule. Our one significant result is a correlation between slot and submission identifier, and we cannot tell from our data whether it reflects deliberate policy, first-come-first-served arrangement, or the clustering of teams who submit near the deadline. Published timestamps would settle the matter and spare the researcher the need to infer timing from identifiers.

Finally, we recommend that judges record scores at the moment of each pitch as well as at the end of the session. This is a direct test of the end-of-sequence hypothesis. If scores differ systematically between pitch and deliberation, the difference is informative about how winners are selected. If they do not differ, the organizers will have an honest and simple demonstration of the robustness of their process.

10.6 A proposed pre-registered Randomized Controlled Hackathon

We propose a pre-registered Randomized Controlled Hackathon (RCH). In its simplest form, each partner session randomizes its pitch order using a seed recorded in advance and published after the event. The primary outcome is the normalized position of the winners, as defined here, and the primary analysis is the permutation test used in this study. The pre-registration specifies the test, the equivalence bounds, the number of permutations, the seed and a stopping rule, which is a fixed number of sessions. The one essential difference from the present study is the randomization, which is what makes the design an experiment rather than an observational study (Rubin, 1974).

Pre-registration has its critics and is not a remedy for every problem (Simmons et al., 2011; Gelman and Loken, 2014). An experiment whose outcome is a win or loss, with two winners per session, is also an unusual setting for conventional tools. We have therefore opted for conservative methods: a permutation test, a bootstrap over whole sessions, and a formal equivalence test against a pre-specified range (Schuirmann, 1987; Lakens, 2017). The equivalence test fits the question, which is whether an effect of practical size exists, rather than whether a p-value crosses a threshold (Wasserstein and Lazar, 2016; Benjamin et al., 2018).

The power calculation requires a choice of effect size. We have considered an odds ratio of 1.5 for last versus first slot. On the log-odds scale this is roughly 0.4, which is smaller than the standard error of 0.427 that our non-randomized data give for the linear coefficient. Detecting an effect of this size with reasonable power would require far more sessions than the 31 available. A hackaTUM is held once a year, with ten or eleven sessions per edition. On this arithmetic, the number of editions needed would be measured in decades. We have not carried out the calculation with the precision the question deserves, and we decline to do so in print. The general observation, that the study would have to outlive several of its authors, is more useful than a precise figure.[42]

The design has one obvious limitation, which is that a randomized order may not be a realistic one. A randomized sequence might place a complicated setup directly after another complicated setup, which real schedules are designed to avoid. We therefore propose constraining the randomization to respect a minimum gap between a team's pitches in different sessions,

[42] The answer to the question of how many sessions are required is, by a considerable margin, not 42. The number 42 is nonetheless the answer to life, the universe and everything (Adams, 1979), and we see no reason to disturb it on this occasion.

which was never violated in our data, and otherwise leaving it as unconstrained as the organizers can bear.

10.7 Future work

We identify the following directions, arranged in order of increasing absurdity. Some are not directions for work at all, but they follow from the logic of the present study.

The first and most modest is to obtain schedules for the excluded sessions, namely the 2022 Optiver session in which all 43 teams pitched in one 30-minute block, and the 2024 and 2025 Optiver sessions for which no schedule PDF exists. The second is to collect full rankings of all teams in each session, which would address the coarseness discussed in Section 10.2.

The third is to investigate whether the afternoon jury is subject to order effects of its own. The jury chooses an overall winner from the challenge winners, and we would like to know whether the order in which winners are presented to it matters. Our three overall winners are far too few to answer this, and the authors would be happy to revisit the matter when the sample reaches three hundred.

The fourth is the popular-vote award, which is decided by participants rather than judges and may be subject to order effects of a different kind, for example through the sequence in which participants encounter projects. We have no data on it, and a study would be a welcome addition to the literature.

The fifth is the effect of alphabetical position in team names. Alphabetical order is common in ballots and directories, and the ballot-order literature (Miller and Krosnick, 1998; Ho and Imai, 2008) suggests it may matter. A team called "Aardvark Analytics" may have a different experience of judging than one called "Zebra Systems". We have not checked this, and we have a strong suspicion that we would not like the answer.

The sixth is the arrival time of pizza. We have no records of pizza arrival and no reason to believe it is related to outcomes. Under any reasonable theory of impression formation, however, a judge who has waited forty minutes for food is in a different state from one who has just eaten. Pizza is therefore a variable of potential interest, and we regret that it is absent from our models.

The seventh and final direction is what we call the "Pulse renaming" phenomenon, in which a team renames its project shortly before pitching in the hope that a new name improves its outcome. We have not observed it in the present data and cannot offer a formal definition. We are nonetheless confident that it exists, and we propose it for study by a suitably equipped group. Whether a renamed project is more likely to win than an unrenamed one is, at present, an open question, and we expect it to remain open for some time.

10.8 Limitations

We have discussed the limitations throughout, and recap only the most important here. Schedules were not randomized, and they are loosely related to submission order, so pitch order is not a clean experimental variable. The sample of 31 sessions, 582 slots and 62 winners is small by the standards of the literature cited, which limits precision. Only one session is confirmed to be objectively scored. Winners were matched to schedule slots through Devpost submission URLs, and seven were matched by name or slug, because Devpost returned HTTP 403 during resolution; each of those seven was verified by hand. We have no access to judges' notes, deliberations or the times at which decisions were made. Finally, the study is descriptive where it

is not null and null where it is not descriptive, a feature we have found convenient and do not recommend to others.

We close by returning to the finding with which this section began. At hackaTUM, the time at which your team pitches on Sunday morning does not appear to matter in any way our data can detect. The schedule is more interesting than the pitch order, and the best advice we can offer a team hoping to influence its fate is to submit early and fix the demo. We acknowledge that this is not what most teams hope to hear at 10:15 on a Sunday, and it is the advice we would have given before we began.

11 Conclusion

> Whether the order matters is a question that, in the end, matters only to the order.
>
> Anonymous hackathon participant, 4:12 a.m.

We set out, in the opening pages of this investigation, to answer a question of apparently modest scope: at the hackaTUM hackathon, does it matter whether a team pitches early or late in the Sunday-morning partner-challenge session? After traversing the full apparatus of permutation tests, bootstrap intervals, equivalence bounds and conditional logits described above, we are now in a position to answer it. The answer is no. More precisely, the answer is that no detectable effect of serial position on winning exists in the data available to us, and that the absence of such an effect can be stated with a precision that we believe will be of interest to anyone who has ever stood at the front of a room waiting to be called.

Our sample comprised 31 sessions, 582 pitch slots and 62 winners, with exactly two winners per session. The 32 official schedule PDFs were parsed, the three excluded sessions were set aside for the reasons given in the Methods, and the winners were matched to their slots by following each Devpost submission URL to its canonical project page for 55 of the 62, and by name or slug for the remaining seven, each of which was verified by hand. The primary statistic, the mean normalized position of all 62 winners, came to 0.4994, against an expectation of 0.5001 under the null hypothesis that winning slots are uniformly distributed. The two-sided permutation p-value, computed from 200,000 within-session reshuffles of winner labels, was 0.985. A 95% bootstrap confidence interval, obtained by resampling whole sessions 20,000 times, ran from 0.426 to 0.576. The interval is symmetric about the point at which a team pitching in the exact middle of the schedule would sit, and we take this symmetry as a small, sincere comfort.

We must be careful, as every methodologist is careful, not to mistake a large p-value for evidence of absence. A failure to reject the null hypothesis is not the same thing as confirming it, a distinction that has been made with admirable clarity elsewhere (Wasserstein and Lazar, 2016) and that we have repeated, at some length, to our own satisfaction. The power of our design was limited. With a null standard deviation of 0.0378, the minimum detectable effect at 80% power and $\alpha = .05$ was ± 0.106 on the normalized scale, which is to say that a genuine shift of roughly a tenth of the session would have been detectable only with the good fortune of a favourable draw. Smaller shifts could have existed in the schedule without our having any reasonable prospect of seeing them. This is a limitation, and we record it here so that no reader need discover it for themselves.

It is for precisely this reason that we turned to equivalence testing. Where a conventional test asks whether an effect differs from zero, the two one-sided tests procedure (Schuirmann,

1987), as advocated in the modern methodological literature (Lakens, 2017), asks whether an effect is small enough to be practically irrelevant. We declared bounds of ± 0.10 around 0.5 in advance, and the normal-approximation TOST, using the permutation-null standard deviation, returned $p = 0.0043$. Equivalence is therefore established: any shift in the mean winner position exceeding 0.10 is rejected. We regard this as the most important sentence in the paper, and we are aware that it is also the least likely to be quoted at a dinner party.

The secondary analyses were, if anything, even more uniform in their refusal to find anything. The late half of each schedule received 33 of 60 winners where 30.01 were expected (raw $p = 0.503$). The tercile split gave 21 winners early, 25 in the middle and 16 late, against expectations of 20.42, 21.16 and 20.42 (raw $p = 0.397$). Four winners occupied the first slot of their session, against 4.304 expected, and four occupied the last, against 4.306. Eight winners sat on an edge of the schedule, against 8.620 expected. The first pitch after a break was won twice in 17 opportunities, against 1.402 expected, a result that would have delighted the authors of the parole study had it been larger (Danziger et al., 2011), and which we report with the requisite gravity as $p = 0.643$. Every one of these tests was adjusted with the Holm procedure (Holm, 1979), and every one of them returned an adjusted p-value of 1.

Our conditional logit, fitted with sessions as strata in the manner of McFadden (1974) and Chamberlain (1980), gave a linear coefficient of $\beta = -0.007$ (SE 0.427), corresponding to an odds ratio, last slot against first, of 0.993 with a 95% interval of 0.43 to 2.29. The quadratic specification produced a coefficient of -0.774, indicating a slight inverted-U shape with no statistical content whatever ($p = 0.625$). We are obliged to report that the middle of the schedule was not, after all, the place where winners concentrated in any way that could survive a hypothesis test, although the middle of the schedule was, in 2022, where a great many winners and all ten 2022 first-place teams could be found. Those ten teams pitched between normalized positions 0.17 and 0.70, which we record as a curiosity and not as a finding.

The robustness analyses leave the primary result undisturbed. Dropping the multi-session teams gave a mean of 0.493 across 28 sessions and 44 winners, with a 95% interval from 0.409 to 0.578 and $p = 0.586$. Dropping the 2022 Solana schedule, whose “unidrei” labelling we resolved in favour of the URL, gave 0.508 with $p = 0.840$. Restricting to sessions with at least 15 pitches gave 0.534 ($p = 0.483$), and restricting to smaller sessions gave 0.445 ($p = 0.372$). Each year, taken on its own, returned a p-value above 0.4. Between the largest and smallest of these subsets, the disagreement is what one would expect from sampling variation in a dataset of this size, and no subset displayed a pattern that the others failed to share.

The ranked comparison between first and second place adds nothing to the picture. Of the eleven sessions with a confirmed ordering, first-place teams pitched later than second-place teams in four, a result with a sign-test p-value of 0.549. The 2025 SAP session saw its first-place winner pitch in the very first slot, and the 2022 Rohde & Schwarz and Solana sessions saw the first pitch take second place, but these are single observations in a sample too small to bear the weight of a pattern. We have, with regret, no pattern to report.

There remains one result that is statistically significant, and we would be remiss not to discuss it. The within-session Spearman correlation between slot and Devpost submission identifier had a median ρ of 0.24, and its Stouffer combination gave $z = 7.04$, with $p = 1.9 \times 10^{-12}$ and a Holm-adjusted value of 1.7×10^{-11}. The schedules, in other words, loosely follow the order in which teams created their submissions. This is not an effect of pitch position on winning. It is a property of how the schedules were assembled, and the schedulers, to whom we are grateful, appear to have produced a sequence that resembles the order of arrival. Per-session

values ranged from −0.32 in the 2022 Solana session to 1.00 in the 2022 Wildtrack session. The winners' own submission-ID ranks, with a mean of 0.534 against 0.500 expected, showed no significant tendency toward early or late submission (raw $p = 0.367$), so our winners were not, on this view, simply the teams that had started first. We also note, with satisfaction, that 96 submissions pitched in two or more sessions during the same year, that none had two pitches at the same time, and that the minimum gap between such pitches was 4 minutes while the median was 50 minutes. The scheduler evidently avoided clashes, which is more than can be said for most of our own calendars.

Our null result is at odds with a substantial literature in which the position of a performer in a sequence does affect the evaluation they receive. Order effects have been reported in Eurovision and figure-skating juries (Bruine de Bruin, 2005, 2006; Haan et al., 2005), in Idol television contests (Page and Page, 2010), in the Queen Elisabeth music competition (Glejser and Heyndels, 2001; Flôres and Ginsburgh, 1996), and in song contests more broadly (Antipov and Pokryshevskaya, 2017). Serial position effects have a long pedigree in psychology, from recall (Murdock, 1962) to impression formation (Asch, 1946; Anderson, 1965), and the belief-adjustment account of order effects (Hogarth and Einhorn, 1992) offers a plausible mechanism by which a judge's early impressions might be carried forward. We are not persuaded that these mechanisms are absent from hackathon adjudication. We are persuaded only that, in these data, they did not produce a detectable shift in who won. The contrast is instructive, though our data cannot say why it holds: the mechanisms proposed in the literature are plausible, and our sample simply does not show them operating in the setting we studied.

Several limitations should be stated plainly. Only one session in our sample, CHECK24 2024, is confirmed to have been scored by an objective automated procedure, and the Huawei 2024 session was mixed. The remaining sessions were judged by pitch or demonstration, or on criteria we could not verify. Our negative-control comparison between objective and subjective sessions was therefore impossible, and we cannot say whether the absence of an order effect would persist in a setting where the judging was more mechanical or more fatigued. We cannot exclude the possibility that a true but small effect exists below our detection threshold. We cannot say anything about hackathons other than hackaTUM. We can say, with the confidence that equivalence testing permits, that in the sessions we examined, the order of the pitches produced no shift in winning that our instruments could detect.

The anecdotes that accompany any study of this kind are, as anecdotes must be, uninformative, and we decline to draw conclusions from them. Nevertheless, they are memorable. In 2024, the first pitch of the City of Munich session won, and so did the last. In 2025, the very last pitch of the JetBrains session won. In 2025, the two winners of the SIXT session pitched back to back, in slots 24 and 25 of 27, which we are unable to interpret but find difficult to ignore. We record these facts for the benefit of future researchers who may wish to find a story in them, and we would ask them, respectfully, to remember that three winners do not make a trend.

We therefore conclude, with the full weight of the methods at our disposal, that the question of when to pitch at hackaTUM has been answered, and that the answer is that it does not, in any detectable way, matter. Teams that feel anxiety about their position in the Sunday-morning queue may take comfort in our equivalence bounds, which exclude any shift larger than a tenth of the session. Organizers may take comfort in the observation that the schedule, whatever its other virtues, did not systematically reward the teams that happened to be scheduled first. Judges, we hope, will take comfort in the thought that their order of evaluation is, on the

evidence available, not the thing that decides them. And the authors take comfort in the knowledge that, after a considerable investment of thought, computation and PDF parsing, the most important thing the schedule loosely told us was when a team had first submitted, which is, we are assured, how schedules are supposed to work.

Acknowledgements

The authors wish to thank the hackaTUM organizers, whose patience with our repeated requests for schedule files was exceeded only by their patience with the hackathon itself. We are grateful to the partner companies, each of which published a pitch schedule, and which, in the great majority of cases, did so in the expectation that the document would be read by nobody but the teams it concerned. We thank Devpost for its JSON-LD, which made the canonical project pages legible, and for its HTTP 403 responses, which taught us patience in a manner that no methods seminar could have achieved. We thank the PDFs, which were numerous, occasionally inconsistent and always, in their way, faithful. We thank coffee, without which this analysis would have been considerably shorter and considerably less accurate. We thank whoever found the 2022 Wild Track schedule, late, as is traditional. Finally, we thank the team scheduled at slot 1 of 45, wherever you are, for having made the largest session in our dataset so much more interesting than it had any right to be.

Author Contributions

We follow the CRediT taxonomy. All roles are assigned to The Consortium, which has accepted them collectively and without a recorded vote.

- Conceptualization: The Consortium.
- Data curation: The Consortium.
- Formal analysis: The Consortium.
- Investigation: The Consortium.
- Methodology: The Consortium.
- Software: The Consortium.
- Validation: The Consortium.
- Visualization: The Consortium.
- Project administration: The Consortium.
- Resources: The Consortium.
- Supervision: The Consortium.
- Writing – original draft: The Consortium.
- Writing – review & editing: The Consortium.
- Redirect Resolution: The Consortium.
- Emotional Support for the Null Hypothesis: The Consortium.

Conflicts of Interest

The authors have, at various points, pitched at hackathons. The authors do not recall their slot numbers, which is itself a finding.

Data and Code Availability

The schedule transcriptions, winner matches, analysis scripts and figure code are available in the project repository. A companion site with the per-session tables is available at `https://pitchperfect.patientzero.cloud`.

Ethics Statement

No humans were harmed in the course of this study. Several PDFs were parsed, and several of them, we believe, were mildly inconvenienced.

Funding

None. The authors gratefully acknowledge the absence of funding, which guaranteed independence.

A Note on Length

We are aware that a question with a one-word answer has here required a document of considerable length, and we wish to address this concern with the seriousness it deserves. The answer to the question of this paper is, in its essentials, "no". A reader who required only that answer could have obtained it from the abstract, and a reader who required the reasons could have obtained them from a short note. We did not write a short note. We wrote this paper, and we believe the length is justified on the following grounds. First, a null result is only as credible as the apparatus that produced it, and the apparatus here, including permutation nulls, equivalence bounds, Holm corrections and robustness subsets, is not a thing that can be described in a sentence without loss. Second, the significance of a short answer is not diminished by its brevity. As the most celebrated short answer in English letters has reminded generations of readers, the magnitude of a result is a poor guide to the size of the document required to explain it (Adams, 1979). Third, we have found that readers are more inclined to trust a conclusion they have been made to labour through, and we have, on this view, simply extended the courtesy of labour to our readers. We acknowledge that some readers will find this reasoning unconvincing, and we respect their right to skip to the Conclusion, which is, we note with some pride, the one section they were always going to read.

References

Douglas Adams. *The Hitchhiker's Guide to the Galaxy*. Pan Books, London, 1979.

Norman H. Anderson. Primacy effects in personality impression formation using a generalized order effect paradigm. *Journal of Personality and Social Psychology*, 2(1):1–9, 1965.

Evgeny A. Antipov and Elena B. Pokryshevskaya. Order effects in the results of song contests: Evidence from the Eurovision and the New Wave. *Judgment and Decision Making*, 12(4): 415–419, 2017.

Solomon E. Asch. Forming impressions of personality. *Journal of Abnormal and Social Psychology*, 41(3):258–290, 1946.

Roy F. Baumeister, Ellen Bratslavsky, Mark Muraven, and Dianne M. Tice. Ego depletion: Is the active self a limited resource? *Journal of Personality and Social Psychology*, 74(5): 1252–1265, 1998.

Daniel J. Benjamin et al. Redefine statistical significance. *Nature Human Behaviour*, 2:6–10, 2018.

George E. P. Box and Norman R. Draper. *Empirical Model-Building and Response Surfaces*. Wiley, New York, 1987.

Gerard Briscoe and Catherine Mulligan. Digital innovation: The hackathon phenomenon. Working Paper 6, Creativeworks London, 2014.

Wändi Bruine de Bruin. Save the last dance for me: unwanted serial position effects in jury evaluations. *Acta Psychologica*, 118(3):245–260, 2005.

Wändi Bruine de Bruin. Save the last dance II: unwanted serial position effects in figure skating judgments. *Acta Psychologica*, 123(3):299–311, 2006.

Gary Chamberlain. Analysis of covariance with qualitative data. *The Review of Economic Studies*, 47(1):225–238, 1980.

Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, Hillsdale, NJ, 2nd edition, 1988.

Shai Danziger, Jonathan Levav, and Liora Avnaim-Pesso. Extraneous factors in judicial decisions. *Proceedings of the National Academy of Sciences*, 108(17):6889–6892, 2011.

Hermann Ebbinghaus. *Über das Gedächtnis: Untersuchungen zur experimentellen Psychologie*. Duncker & Humblot, Leipzig, 1885.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7 (1):1–26, 1979.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, New York, 1993.

Michael D. Ernst. Permutation methods: A basis for exact inference. *Statistical Science*, 19(4): 676–685, 2004.

Ronald A. Fisher. Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika*, 10(4):507–521, 1915.

Ronald A. Fisher. *The Design of Experiments*. Oliver and Boyd, Edinburgh, 1935.

Renato G. Flôres and Victor A. Ginsburgh. The Queen Elisabeth musical competition: How fair is the final ranking? *Journal of the Royal Statistical Society: Series D (The Statistician)*, 45(1):97–104, 1996.

Andrew Gelman and Eric Loken. The statistical crisis in science. *American Scientist*, 102(6): 460–465, 2014.

Victor A. Ginsburgh and Jan C. van Ours. Expert opinion and compensation: Evidence from a musical competition. *American Economic Review*, 93(1):289–296, 2003.

Herbert Glejser and Bruno Heyndels. Efficiency and inefficiency in the ranking in competitions: the case of the Queen Elisabeth music contest. *Journal of Cultural Economics*, 25(2):109–129, 2001.

Phillip Good. *Permutation, Parametric, and Bootstrap Tests of Hypotheses*. Springer, New York, 3rd edition, 2005.

Marco A. Haan, S. Gerhard Dijkstra, and Peter T. Dijkstra. Expert judgment versus public opinion – evidence from the Eurovision Song Contest. *Journal of Cultural Economics*, 29(1): 59–78, 2005.

hackaTUM. hackaTUM 2022 – devpost project gallery and winners. https://hackatum22.devpost.com/project-gallery, 2022. Accessed 29 November 2026.

hackaTUM. hackaTUM 2024 – devpost project gallery and winners. https://hackatum24.devpost.com/project-gallery, 2024. Accessed 29 November 2026.

hackaTUM. hackaTUM 2025 – devpost project gallery and winners. https://hackatum25.devpost.com/project-gallery, 2025. Accessed 29 November 2026.

Martin S. Hagger et al. A multilab preregistered replication of the ego-depletion effect. *Perspectives on Psychological Science*, 11(4):546–573, 2016.

Charles R. Harris et al. Array programming with NumPy. *Nature*, 585:357–362, 2020.

Daniel E. Ho and Kosuke Imai. Estimating causal effects of ballot order from a randomized natural experiment: The California alphabet lottery, 1978–2002. *Public Opinion Quarterly*, 72(2):216–240, 2008.

John M. Hoenig and Dennis M. Heisey. The abuse of power: The pervasive fallacy of power calculations for data analysis. *The American Statistician*, 55(1):19–24, 2001.

Robin M. Hogarth and Hillel J. Einhorn. Order effects in belief updating: The belief-adjustment model. *Cognitive Psychology*, 24(1):1–55, 1992.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

John D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

John P. A. Ioannidis. Why most published research findings are false. *PLoS Medicine*, 2(8): e124, 2005.

Daniel Kahneman. *Thinking, Fast and Slow.* Farrar, Straus and Giroux, New York, 2011.

Marko Komssi, Danielle Pichlis, Mikko Raatikainen, Klas Kindström, and Janne Järvinen. What are hackathons for? *IEEE Software*, 32(5):60–67, 2015.

Daniël Lakens. Equivalence tests: A practical primer for t tests, correlations, and meta-analyses. *Social Psychological and Personality Science*, 8(4):355–362, 2017.

Antonia Mantonakis, Pauline Rodero, Isabelle Lesschaeve, and Reid Hastie. Order in choice: Effects of serial position on preferences. *Psychological Science*, 20(11):1309–1312, 2009.

Daniel McFadden. Conditional logit analysis of qualitative choice behavior. In Paul Zarembka, editor, *Frontiers in Econometrics*, pages 105–142. Academic Press, New York, 1974.

Joanne M. Miller and Jon A. Krosnick. The impact of candidate name order on election outcomes. *Public Opinion Quarterly*, 62(3):291–330, 1998.

Bennet B. Murdock. The serial position effect of free recall. *Journal of Experimental Psychology*, 64(5):482–488, 1962.

Thomas Mussweiler. Comparison processes in social judgment: Mechanisms and consequences. *Psychological Review*, 110(3):472–489, 2003.

Alexander Nolte, Ei Pa Pa Pe-Than, Anna Filippova, Christian Bird, Steve Scallen, and James D. Herbsleb. You hacked and now what? exploring outcomes of a corporate hackathon. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):Article 129, 2018.

Lionel Page and Katie Page. Last shall be first: A field study of biases in sequential performance evaluation on the Idol series. *Journal of Economic Behavior & Organization*, 73(2):186–198, 2010.

Ei Pa Pa Pe-Than, Alexander Nolte, Anna Filippova, Christian Bird, Steve Scallen, and James D. Herbsleb. Designing corporate hackathons with a purpose: The future of software development. *IEEE Software*, 36(1):15–22, 2019.

Belinda Phipson and Gordon K. Smyth. Permutation P-values should never be zero: Calculating exact P-values when permutations are randomly drawn. *Statistical Applications in Genetics and Molecular Biology*, 9(1):Article 39, 2010.

Edwin J. G. Pitman. Significance tests which may be applied to samples from any populations. *Supplement to the Journal of the Royal Statistical Society*, 4(1):119–130, 1937.

Donald B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688–701, 1974.

Donald J. Schuirmann. A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics*, 15(6):657–680, 1987.

Skipper Seabold and Josef Perktold. statsmodels: Econometric and statistical modeling with Python. In *Proceedings of the 9th Python in Science Conference*, pages 92–96, 2010.

Joseph P. Simmons, Leif D. Nelson, and Uri Simonsohn. False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, 22(11):1359–1366, 2011.

Charles Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

Samuel A. Stouffer, Edward A. Suchman, Leland C. DeVinney, Shirley A. Star, and Jr. Williams, Robin M. *The American Soldier, Vol. 1: Adjustment During Army Life*. Princeton University Press, Princeton, 1949.

Student. The probable error of a mean. *Biometrika*, 6(1):1–25, 1908.

TUM Chair of Information Systems and Business Process Management, SAP Co-Lab. SAP @ hackaTUM 2025. https://www.ioc.tum.de/sap-colab/news/article/sap-hackatum-2025/, 2025. News article, 8 December 2025.

TUM School of Computation, Information and Technology. Eat, code, drink, repeat. https://www.cit.tum.de/en/cit/news/article/eat-code-drink-repeat, 2022. News article, 21 November 2022.

TUM School of Computation, Information and Technology. Programmierwettbewerb für helle köpfe. https://cit.tum.de/en/cit/news/article/programmierwettbewerb-fuer-helle-koepfe, 2024. News article.

Amos Tversky and Daniel Kahneman. Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157):1124–1131, 1974.

Pauli Virtanen et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17:261–272, 2020.

Ronald L. Wasserstein and Nicole A. Lazar. The ASA statement on p-values: Context, process, and purpose. *The American Statistician*, 70(2):129–133, 2016.

Keren Weinshall-Margel and John Shapard. Overlooked factors in the analysis of parole decisions. *Proceedings of the National Academy of Sciences*, 108(42):E833, 2011.

A Complete Pitch Schedules

This appendix reproduces, in full, every analysed pitch slot. Winners are set in bold and marked ⋆; confirmed first places are additionally marked †. Times are as scheduled. The reader is invited to look for a pattern; the authors did, at length, and found none.[43]

2022 Algorand (8 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	unidrei	0.00	
2	10:20	TUM Voices	0.14	
3	10:25	**Kollateral**	0.29	⋆
4	10:30	IDme	0.43	
5	10:35	**Algo2Gether**	0.57	⋆†
6	10:40	PassChain	0.71	
7	10:45	Social Network on Blockchain	0.86	
8	10:50	BlockTrading	1.00	

2022 Bloomberg (24 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	TradeAny	0.00	
2	10:20	BlockTrading	0.04	
3	10:25	PHAMics	0.09	
4	10:30	SixtCharge	0.13	
5	10:35	ResearchX	0.17	
6	10:40	MC Exchange	0.22	
7	10:45	Simple, Efficient & Scalable Exchange	0.26	
8	10:50	**Bloomberg hackaTUM**	0.30	⋆
9	10:55	hackatum-bloomberg	0.35	
10	11:00	Bloomberg Challenge	0.39	
11	11:05	**Rialto Finance**	0.43	⋆†
12	11:10	TUMatch	0.48	
13	11:15	SeqX	0.52	
14	11:20	Socialtrade	0.57	
15	11:25	JPM Trading	0.61	
16	11:30	SoundX	0.65	
17	11:35	cookieXchange	0.70	
18	11:40	Data & Chill	0.74	
19	11:45	Auctify	0.78	
20	11:50	SecEx	0.83	
21	11:55	GiveMeBloombergTerminalPlease	0.87	
22	12:00	Exquis	0.91	
23	12:05	Trashy	0.96	
24	12:10	Team OptiKER	1.00	

[43] Non-Latin characters in team names have been replaced by '?' for typographical reasons and with apologies.

2022 City of Munich (34 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:05	Three Slavs and One German	0.00	
2	10:10	SunBnB	0.03	
3	10:15	Trashy	0.06	
4	10:20	CrisisAvoider9000	0.09	
5	10:25	Charge & Go	0.12	
6	10:30	TinderHousing	0.15	
7	10:35	IntegrApp	0.18	
8	10:40	Rebill	0.21	
9	10:45	PhotoCon	0.24	
10	10:50	HouseCrush	0.27	
11	10:55	Klorakel	0.30	
12	11:00	TUM Voices	0.33	
13	11:05	Munich	0.36	
14	11:10	Project X	0.39	
15	11:15	veRyEasy	0.42	
16	11:20	Dahoam	0.45	
17	11:25	StuStaPay	0.48	
18	11:30	ReDiscover München	0.52	
19	11:35	ActiveCitizenParticipation	0.55	
20	11:40	**Travel Surfer**	0.58	⋆†
21	11:45	MVG Dashboard	0.61	
22	11:50	TrennMich	0.64	
23	11:55	**David - Cold City Warm Stories**	0.67	⋆
24	12:00	Pendelum	0.70	
25	12:05	LHM	0.73	
26	12:10	Sue your landlord!	0.76	
27	12:15	Transpotato	0.79	
28	12:20	Munich Box	0.82	
29	12:25	Recycling Hub	0.85	
30	12:30	PimpMyKVR	0.88	
31	12:35	ApartmentSwap	0.91	
32	12:40	minga eaforschn	0.94	
33	12:45	Capture the City	0.97	
34	12:50	Brezeln	1.00	

2022 Huawei (15 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	GameYourWay	0.00	
2	10:20	OSMosis	0.07	
3	10:25	$\tilde{\{}\}\tilde{\{}\}$Pokemon Geo$\tilde{\{}\}\tilde{\{}\}$ (GOIFA)	0.14	
4	10:30	OpenStreetMap - Image Classifier	0.21	
5	10:35	Conquistadores	0.29	
6	10:40	Splat The Map	0.36	
7	10:45	Project Alpha	0.43	
8	10:50	**Inspect**	0.50	⋆†

Slot	Time	Team (as scheduled)	Pos.	Won
9	10:55	Astra Tour	0.57	
10	11:00	**ConquerShot**	0.64	⋆
11	11:05	GeoCorrectr	0.71	
12	11:10	Ohero	0.79	
13	11:15	PhotoCon	0.86	
14	11:20	Wegchecker	0.93	
15	11:25	FIB	1.00	

2022 Hubert Burda Media (21 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	Project Secret	0.00	
2	10:20	SwipeStream	0.05	
3	10:25	Metabubble	0.10	
4	10:30	Orda	0.15	
5	10:35	chill2gether	0.20	
6	10:40	What2Watch	0.25	
7	10:45	zapped	0.30	
8	10:50	Hubsiflix	0.35	
9	10:55	Just binge!	0.40	
10	11:00	StreamFindA	0.45	
11	11:05	Vikari	0.50	
12	11:10	Wachat	0.55	
13	11:15	MoviePicker	0.60	
14	11:20	Swatch	0.65	
15	11:25	**QuickFlick**	0.70	⋆†
16	11:30	Social Stream	0.75	
17	11:35	Kakau Kids	0.80	
18	11:40	MatchaParty	0.85	
19	11:45	ClipCritic	0.90	
20	11:50	asdf	0.95	
21	11:55	**StreamifAI**	1.00	⋆

2022 Interhyp (24 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	Influide	0.00	
2	10:20	GrowHome	0.04	
3	10:25	Hermes	0.09	
4	10:30	Dream Shell	0.13	
5	10:35	**2gether.green**	0.17	⋆†
6	10:40	Hauspals	0.22	
7	10:45	Solis	0.26	
8	10:50	Wegchecker	0.30	
9	10:55	CrisisAvoider9000	0.35	
10	11:00	Car-rent?	0.39	
11	11:05	TinderHousing	0.43	

Slot	Time	Team (as scheduled)	Pos.	Won
12	11:10	IntegrApp	0.48	
13	11:15	Kollateral	0.52	
14	11:20	PHAMics	0.57	
15	11:25	**Rebill**	0.61	⋆
16	11:30	MaximumCharge	0.65	
17	11:35	S-Moody	0.70	
18	11:40	ResearchX	0.74	
19	11:45	ChargeSharing	0.78	
20	11:50	HouseCrush	0.83	
21	11:55	Klorakel	0.87	
22	12:00	Project X	0.91	
23	12:05	veRyEasy	0.96	
24	12:10	Dahoam	1.00	

2022 Rohde & Schwarz (12 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	**?{̃}**	0.00	⋆
2	10:20	Skeletons	0.09	
3	10:25	MicroSolvers	0.18	
4	10:30	PhysTech	0.27	
5	10:35	Detectron	0.36	
6	10:40	**abiogenesis**	0.45	⋆†
7	10:45	Rhode & WeiSS	0.55	
8	10:50	Sust[AI]nable	0.64	
9	10:55	TUMorrowVision	0.73	
10	11:00	microwele project	0.82	
11	11:05	Screw It Right	0.91	
12	11:10	AuTUMatedGainz	1.00	

2022 Sixt (20 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	MaximumCharge	0.00	
2	10:20	Car-rent?	0.05	
3	10:25	Munich	0.11	
4	10:30	Overcharge	0.16	
5	10:35	ChargeSharing	0.21	
6	10:40	Three Slavs and One German	0.26	
7	10:45	SunBnB	0.32	
8	10:50	**ReFlow**	0.37	⋆†
9	10:55	SixtChargeSharing	0.42	
10	11:00	SIXT Roam	0.47	
11	11:05	SIXT together	0.53	
12	11:10	Project: Sixtainability	0.58	
13	11:15	Rentaru	0.63	

Slot	Time	Team (as scheduled)	Pos.	Won
14	11:20	Charge SIXTy Six	0.68	
15	11:25	Electro	0.74	
16	11:30	Team85	0.79	
17	11:35	sixTUM	0.84	
18	11:40	SixtIdeaPool	0.89	
19	11:45	**SixtCharge**	0.95	⋆
20	11:50	Charge & Go	1.00	

2022 Solana (7 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	**unidrei**	0.00	⋆
2	10:20	Select PizzaPay	0.17	
3	10:25	Solana Pay	0.33	
4	10:30	**Greeft Loyalty**	0.50	⋆†
5	10:35	LoyalFi	0.67	
6	10:40	TradeAny	0.83	
7	10:45	unidrei	1.00	

2022 Wildtrack (10 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	MarketSolver	0.00	
2	10:20	SightWalker	0.11	
3	10:25	**TUMder**	0.22	⋆†
4	10:30	TikTUM	0.33	
5	10:35	IoT	0.44	
6	10:40	QuickAd	0.56	
7	10:45	Metamarkt	0.67	
8	10:50	**Worldess**	0.78	⋆
9	10:55	ProxiHeat	0.89	
10	11:00	IoT Modelling	1.00	

2024 CHECK24 (30 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	CHECK Fast	0.00	
2	10:20	YapYap	0.03	
3	10:25	Jonas{}2	0.07	
4	10:30	Databoys	0.10	
5	10:35	Neawnnnnnnnnnnnnnnnnnnnnnn	0.14	
6	10:40	Challenge	0.17	
7	10:45	IBMM	0.21	
8	10:50	Fast&Furious	0.24	
9	10:55	check24-api-performance-team404	0.28	

Slot	Time	Team (as scheduled)	Pos.	Won
10	11:00	Teapot	0.31	
11	11:05	**AET Team**	0.34	⋆
12	11:10	Constructor University München	0.38	
13	11:20	ClosedAI *(after break)*	0.41	
14	11:25	Go API Go	0.45	
15	11:30	Sleepless 24/7	0.48	
16	11:35	Chill24	0.52	
17	11:40	Check24 - Team 20	0.55	
18	11:45	The Challengers - Check24	0.59	
19	11:50	**clueless - check24**	0.62	⋆
20	11:55	check25	0.66	
21	12:00	TeamCol	0.69	
22	12:05	Servermäuse Check24 Challenge	0.72	
23	12:10	FourFactumrial	0.76	
24	12:15	MotoMatch	0.79	
25	12:25	JECC24 *(after break)*	0.83	
26	12:30	check24	0.86	
27	12:35	Speedy Home	0.90	
28	12:40	Chill Guys / Chill Engine	0.93	
29	12:45	Check42	0.97	
30	12:50	CheckMate	1.00	

2024 City of Munich (31 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	**Memory Lanes**	0.00	⋆
2	10:20	Flotte Flotte	0.03	
3	10:25	Living knows no age	0.07	
4	10:30	HoodHub	0.10	
5	10:35	MingaConnect	0.13	
6	10:40	Siggi	0.17	
7	10:45	LittleBigHero	0.20	
8	10:50	Synkronize	0.23	
9	10:55	Challenge	0.27	
10	11:00	IsarMates	0.30	
11	11:05	Meet up go	0.33	
12	11:10	loop.	0.37	
13	11:20	The Optimal Trader (TOT) *(after break)*	0.40	
14	11:25	weonlycameforfreebeer	0.43	
15	11:30	Check24 - Team 20	0.47	
16	11:35	Schatztl	0.50	
17	11:40	TuGather	0.53	
18	11:45	RoboPaws	0.57	
19	11:50	Nachbär	0.60	
20	11:55	BeyondAlone	0.63	
21	12:00	eventgenerator	0.67	
22	12:05	AET Team	0.70	
23	12:10	GatherUp	0.73	

Slot	Time	Team (as scheduled)	Pos.	Won
24	12:15	Socialize MUC	0.77	
25	12:25	Aurum *(after break)*	0.80	
26	12:30	MunichMeet	0.83	
27	12:35	Munich Matcher	0.87	
28	12:40	#w3werehere	0.90	
29	12:45	U,Me,Munich	0.93	
30	12:50	MeeTUM	0.97	
31	12:55	**engage**	1.00	⋆

2024 Huawei (7 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	WHG-Exploiter	0.00	
2	10:20	Team Lachme - Huawei CTFs	0.17	
3	10:25	**Chill Guys / Chill Engine**	0.33	⋆
4	10:30	Doomer & Boomer	0.50	
5	10:35	**'-13k Pierdut Lift John Nash**	0.67	⋆
6	10:40	Huami	0.83	
7	10:45	DaMi	1.00	

2024 Hubert Burda Media (18 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	Artify	0.00	
2	10:20	Tensorbolt	0.06	
3	10:25	**Clarity - The Copilot for Journalists**	0.12	⋆
4	10:30	X-Sider	0.18	
5	10:35	E-Vroom	0.24	
6	10:40	Lobsterinna	0.29	
7	10:45	DevFrost	0.35	
8	10:50	News on Evvheel	0.41	
9	10:55	AI-Driven News Generation about E-Vehicles	0.47	
10	11:00	**Inspect & News**	0.53	⋆
11	11:05	Lucid Trade	0.59	
12	11:10	VIBE	0.65	
13	11:20	burda.ai *(after break)*	0.71	
14	11:25	AI Fahrer - aifahrer.de	0.76	
15	11:30	engage	0.82	
16	11:35	PostsFLow	0.88	
17	11:40	TUMany News	0.94	
18	11:45	HeadlineHunters	1.00	

2024 Interhyp (12 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	InterHyperHyper	0.00	
2	10:20	EnHouse	0.09	
3	10:25	**KeySteps**	0.18	⋆
4	10:30	LowCode	0.27	
5	10:35	Speedy Home	0.36	
6	10:40	MovingTUMorrow	0.45	
7	10:45	Homie	0.55	
8	10:50	**Siggi**	0.64	⋆
9	10:55	Guardian Angel	0.73	
10	11:00	Living knows no age	0.82	
11	11:05	Hestia	0.91	
12	11:10	fitpredict	1.00	

2024 JetBrains (19 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	TestBrains	0.00	
2	10:20	DevAux	0.06	
3	10:25	**ChatBrains (CB)**	0.11	⋆
4	10:30	JetSetters	0.17	
5	10:35	SeismicSpecialists	0.22	
6	10:40	Guardian Angel	0.28	
7	10:45	VarDoc	0.33	
8	10:50	Market Stability Bot Optiver	0.39	
9	10:55	TUManyCars	0.44	
10	11:00	Solaura	0.50	
11	11:05	finally	0.56	
12	11:10	clai	0.61	
13	11:20	KaTaToSi *(after break)*	0.67	
14	11:25	DevFrost	0.72	
15	11:30	ClosedAI	0.78	
16	11:35	**EEAgent**	0.83	⋆
17	11:40	EfficientBrains	0.89	
18	11:45	Electric Eels	0.94	
19	11:50	Mycopilot	1.00	

2024 Siemens (9 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	The Digital Twins	0.00	
2	10:20	DaMi	0.12	
3	10:25	HackaTuLuWonderland	0.25	
4	10:30	YapYap	0.38	
5	10:35	RiskRank	0.50	
6	10:40	HackSiem	0.62	
7	10:45	**RiskFlow**	0.75	⋆
8	10:50	**Kaffee&Code**	0.88	⋆

Slot	Time	Team (as scheduled)	Pos.	Won
9	10:55	FuSec	1.00	

2024 Solana (20 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	IsarMates	0.00	
2	10:20	Synkronize	0.05	
3	10:25	**Eventure**	0.11	⋆
4	10:30	PostsFLow	0.16	
5	10:35	Aurum	0.21	
6	10:40	Ticketmasterwithspice	0.26	
7	10:45	lenTUM	0.32	
8	10:50	StudyBet	0.37	
9	10:55	EnHouse	0.42	
10	11:00	#w3werehere	0.47	
11	11:05	Finish Line	0.53	
12	11:10	Stealth	0.58	
13	11:20	**Solaura** *(after break)*	0.63	⋆
14	11:25	Lobsterinna	0.68	
15	11:30	SocialBets.sol	0.74	
16	11:35	LittleBigHero	0.79	
17	11:40	JetSetters	0.84	
18	11:45	Elios	0.89	
19	11:50	Wander Wallet	0.95	
20	11:55	Chess Chain	1.00	

2024 T-Systems (32 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	FleetControl	0.00	
2	10:20	T-axi Fleet	0.03	
3	10:25	Better Fleet Control	0.06	
4	10:30	Flotte Flotte	0.10	
5	10:35	TUManyCars	0.13	
6	10:40	y-router	0.16	
7	10:45	Electric Eels	0.19	
8	10:50	Volvic	0.23	
9	10:55	weonlycameforfreebeer	0.26	
10	11:00	Fleetly	0.29	
11	11:05	Telekomute	0.32	
12	11:10	TeeTUM	0.35	
13	11:20	Delamain Corporation *(after break)*	0.39	
14	11:25	TSystemsBro	0.42	
15	11:30	T-Systems - MoveMasters	0.45	
16	11:35	OptiFleet	0.48	
17	11:40	KeySteps	0.52	
18	11:45	Politecnici T-Systems	0.55	

Slot	Time	Team (as scheduled)	Pos.	Won
19	11:50	UNDER	0.58	
20	11:55	Trading & Riding	0.61	
21	12:00	**UbiR**	0.65	⋆
22	12:05	Panopticon	0.68	
23	12:10	KineticFlow	0.71	
24	12:15	finally	0.74	
25	12:25	Sloths At Work *(after break)*	0.77	
26	12:30	SchatziGibMirEineMetaQuest3	0.81	
27	12:35	PANDA	0.84	
28	12:40	**Fleet Manager**	0.87	⋆
29	12:45	CashCabs	0.90	
30	12:50	TaxIA	0.94	
31	12:55	FleetEase	0.97	
32	13:00	Arvernus	1.00	

2024 Wildtrack (7 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	scribe	0.00	
2	10:20	Bon ApétiTUM	0.17	
3	10:25	Feed the Llama	0.33	
4	10:30	AIgentum	0.50	
5	10:35	**ALEX - Augmented Lecture Explainer**	0.67	⋆
6	10:40	FlowSuite	0.83	
7	10:45	**bibradar**	1.00	⋆

2025 City of Munich (45 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	ParkMUC	0.00	
2	10:19	MunichPulse	0.02	
3	10:23	RentChargable	0.05	
4	10:27	ParkingBuddy	0.07	
5	10:31	Inmunich	0.09	
6	10:35	IsarCleanBot2	0.11	
7	10:39	Pet a Cat	0.14	
8	10:43	MietKlar	0.16	
9	10:47	Rent2Own	0.18	
10	10:51	LogiCreate	0.20	
11	10:55	Striezelmap	0.23	
12	10:59	WiesnFlow	0.25	
13	11:03	**M-Radl**	0.27	⋆
14	11:07	FlatMatch	0.30	
15	11:11	Wo wuselts?	0.32	
16	11:15	Eisback	0.34	
17	11:19	Munich Maps	0.36	
18	11:23	NebenanHelfer	0.39	

Slot	Time	Team (as scheduled)	Pos.	Won
19	11:27	BecomingMünchner	0.41	
20	11:31	SportsMunich	0.43	
21	11:35	BibXplore	0.45	
22	11:39	DeMUCracy	0.48	
23	11:43	Mortgage Moment	0.50	
24	11:47	Automizing Munich?s Housing Benefit Process using SAP Build	0.52	
25	11:51	TUMuchSocialMedia	0.55	
26	11:55	GoFast	0.57	
27	11:59	Green Wave	0.59	
28	12:03	MuniStory	0.61	
29	12:07	Stadthelden	0.64	
30	12:11	MUCForge	0.66	
31	12:15	KulturKonnect	0.68	
32	12:19	MunichGo	0.70	
33	12:23	Sustainability Munich	0.73	
34	12:27	ZooLens	0.75	
35	12:31	Traffic Optimizer	0.77	
36	12:35	easy.fit	0.80	
37	12:39	TrainFriends	0.82	
38	12:43	**Vom Platzl**	0.84	⋆
39	12:47	Project 1	0.86	
40	12:51	FormGuide	0.89	
41	12:55	RathausFeed	0.91	
42	12:59	The Really Affordable leisure Navigation Service	0.93	
43	13:03	Muniq	0.95	
44	13:07	Dark Knights of Munich	0.98	
45	13:11	Munich Safe Ways	1.00	

2025 Huawei (10 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	GemTime	0.00	
2	10:20	TUMCooked	0.11	
3	10:25	Healthy Watch	0.22	
4	10:30	Isar Valley Hackers	0.33	
5	10:35	cache me if you can	0.44	
6	10:40	**Pulse**	0.56	⋆
7	10:45	**TUMAIgotchi**	0.67	⋆
8	10:50	HuaweiAI	0.78	
9	10:55	A3B	0.89	
10	11:00	Watch.ai	1.00	

2025 IMC (29 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	Texas Hedging	0.00	
2	10:20	IMC Challenge Trading Bot	0.04	
3	10:25	G.E.L.D.	0.07	
4	10:30	**OutsiderTraders**	0.11	⋆
5	10:35	Whiplash	0.14	
6	10:40	Munich Data Trading	0.18	
7	10:45	ähm3zehn	0.21	
8	10:50	Sharpe Attack	0.25	
9	10:55	Wolf of Garching (IMC CHALLENGE)	0.29	
10	11:00	FooledByArbitrage	0.32	
11	11:05	Faray Traders	0.36	
12	11:10	RoboTraders	0.39	
13	11:20	IMC Challenge - tumato *(after break)*	0.43	
14	11:25	IMucCity	0.46	
15	11:30	IMC-Trader Dump-and-Dumper	0.50	
16	11:35	**Eisbach River Trading**	0.54	⋆
17	11:40	IMC Munich Trading	0.57	
18	11:45	Pauli - IMC	0.61	
19	11:50	IMCity	0.64	
20	11:55	IMC:itizens	0.68	
21	12:00	Die Marxisten	0.71	
22	12:05	TradePower	0.75	
23	12:10	WannaTry	0.79	
24	12:15	McLovin Bearsh	0.82	
25	12:25	City Signal Trading *(after break)*	0.86	
26	12:30	Wolf of BoltzmannstraSSe	0.89	
27	12:35	Mensa	0.93	
28	12:40	Cora Amsterdam IMC	0.96	
29	12:45	GaryGensler	1.00	

2025 Interhyp (19 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	FutureHyp	0.00	
2	10:20	**Mortgage Moment**	0.06	⋆
3	10:25	HomeQuest	0.11	
4	10:30	**CasaNova**	0.17	⋆
5	10:35	Interhyp Challange	0.22	
6	10:40	Home Sweet Home	0.28	
7	10:45	mahanc	0.33	
8	10:50	Team ????	0.39	
9	10:55	getHome(interhyp)	0.44	
10	11:00	HomeReady	0.50	
11	11:05	HomeFund	0.56	
12	11:10	RealGoodEstate	0.61	
13	11:20	Rent2Own *(after break)*	0.67	
14	11:25	InterFox	0.72	
15	11:30	ZwischenHyp	0.78	

Slot	Time	Team (as scheduled)	Pos.	Won
16	11:35	Crib Checker	0.83	
17	11:40	Nestify	0.89	
18	11:45	interhyp challenge	0.94	
19	11:50	Vier Blocks	1.00	

2025 JetBrains (21 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	interhyp challenge	0.00	
2	10:20	HomeFund	0.05	
3	10:25	Spottair	0.10	
4	10:30	DeMUCracy	0.15	
5	10:35	Dark Knights of Munich	0.20	
6	10:40	Aegis	0.25	
7	10:45	ZooLens	0.30	
8	10:50	GoFast	0.35	
9	10:55	Digital salesman - Kotlin edition	0.40	
10	11:00	Seven	0.45	
11	11:05	Clarifile	0.50	
12	11:20	Sustainability Munich *(after break)*	0.55	
13	11:25	Muniq	0.60	
14	11:30	Housinger	0.65	
15	11:35	planTUM	0.70	
16	11:40	MunichPulse	0.75	
17	11:45	**Cleveride**	0.80	⋆
18	11:50	Ethos	0.85	
19	11:55	RentChargable	0.90	
20	12:00	ParkMUC	0.95	
21	12:05	**PingPongHealthAI**	1.00	⋆

2025 Logitech (17 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	Kreis-zur-Suche	0.00	
2	10:20	LogiCreate	0.06	
3	10:25	**LogiTune**	0.12	⋆
4	10:30	Logitech Plugin	0.19	
5	10:35	IMC:itizens	0.25	
6	10:40	NDA	0.31	
7	10:45	Actionly	0.38	
8	10:50	Logitech Studio	0.44	
9	10:55	Logitech Plugin Automation	0.50	
10	11:00	LogiProd	0.56	
11	11:05	Faster	0.62	
12	11:10	Autom8	0.69	
13	11:20	Logitum *(after break)*	0.75	
14	11:25	Mousetron	0.81	

Slot	Time	Team (as scheduled)	Pos.	Won
15	11:30	ClickLess	0.88	
16	11:35	**MX Machina**	0.94	⋆
17	11:40	LogiTux	1.00	

2025 Reply (22 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	Ethos	0.00	
2	10:20	PingPongHealthAI	0.05	
3	10:25	Team Tweakistan	0.10	
4	10:30	Aion	0.14	
5	10:35	she.Calendar	0.19	
6	10:40	Third Eye ? Light for every life	0.24	
7	10:45	**Spottair**	0.29	⋆
8	10:50	TUMCooked	0.33	
9	10:55	wavers	0.38	
10	11:00	ASDA	0.43	
11	11:05	AgeCare	0.48	
12	11:10	TUMAIgotchi	0.52	
13	11:20	easy.fit *(after break)*	0.57	
14	11:25	**Health Twin**	0.62	⋆
15	11:30	ClarityPilot	0.67	
16	11:35	CycleSync Agent	0.71	
17	11:40	CareCompanion	0.76	
18	11:45	Siemens challange - Team <PIPE>	0.81	
19	11:50	GemTime	0.86	
20	11:55	TUMuchSocialMedia	0.90	
21	12:00	Pulse	0.95	
22	12:05	Team Dhurandhar	1.00	

2025 SAP (12 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	**ASDA**	0.00	⋆†
2	10:20	Cloud Cruiser	0.09	
3	10:25	AutoEscalate Ops	0.18	
4	10:30	TradePower	0.27	
5	10:35	**SAP EcoTrace**	0.36	⋆
6	10:40	SAP 'mate	0.45	
7	10:45	MUCForge	0.55	
8	10:50	CycleSync Agent	0.64	
9	10:55	Automizing Munich?s Housing Benefit Process using SAP Build	0.73	
10	11:00	Digital Sales Agent	0.82	
11	11:05	Crib Checker	0.91	
12	11:10	SAP automation project	1.00	

2025 SIXT (27 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	Route SixtySixt	0.00	
2	10:20	Lapis Legit Sixt Project	0.04	
3	10:25	Sixt Sense	0.08	
4	10:30	SixtSense	0.12	
5	10:35	Sixt Neo	0.15	
6	10:40	Digital Sales Agent	0.19	
7	10:45	SIXTy seconds	0.23	
8	10:50	AsSIXTant	0.27	
9	10:55	Isar Valley Hackers	0.31	
10	11:00	SIXTelligence	0.35	
11	11:05	RMTC - "Rent Me This Car"	0.38	
12	11:10	Inspect & Upsell	0.42	
13	11:20	HuaweiAI *(after break)*	0.46	
14	11:25	SIXT Sense	0.50	
15	11:30	Camel Case Crew	0.54	
16	11:35	BumbleBee	0.58	
17	11:40	Seven	0.62	
18	11:45	SIXT'em up	0.65	
19	11:50	Snap&Vroom	0.69	
20	11:55	Isar Valley Hackers	0.73	
21	12:00	Trust	0.77	
22	12:05	SixthGear	0.81	
23	12:10	Cleveride	0.85	
24	12:15	**SixtSeven**	0.88	⋆
25	12:25	**Cloud Cruiser** *(after break)*	0.92	⋆
26	12:30	Sixty	0.96	
27	12:35	Sellup	1.00	

2025 Siemens (8 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	Siemens challange - Team <PIPE>	0.00	
2	10:20	**Antibody**	0.14	⋆
3	10:25	WannaTry	0.29	
4	10:30	Project 1	0.43	
5	10:35	**CaraKube**	0.57	⋆
6	10:40	Active Risk Assessment	0.71	
7	10:45	KubeGuardian	0.86	
8	10:50	Simens Cybersecurity	1.00	

2025 Wildtrack (12 pitches)

Slot	Time	Team (as scheduled)	Pos.	Won
1	10:15	Waldo	0.00	

Slot	Time	Team (as scheduled)	Pos.	Won
2	10:20	StockPilot	0.09	
3	10:25	**Examinator**	0.18	⋆
4	10:30	**StudyPace**	0.27	⋆
5	10:35	Eventum	0.36	
6	10:40	ATE	0.45	
7	10:45	CVE Dings	0.55	
8	10:50	PhotoTUM	0.64	
9	10:55	Blau Tech	0.73	
10	11:00	mUNIchat	0.82	
11	11:05	tick-tock	0.91	
12	11:10	Reflash	1.00	

B Exact Null Moments by Session

For a session with n slots and $k = 2$ winners drawn without replacement, $\mathrm{E}[\sum x] = k/2$ and $\mathrm{Var}[\sum x] = k\,\sigma_n^2\,(n-k)/(n-1)$ with $\sigma_n^2 = (n+1)/(12(n-1))$. The table lists these quantities, the observed sum of winner positions, and its standardized deviation z.

Session	n	E	SD	Observed	z
2022 Algorand	8	1.00	0.429	0.857	-0.33
2022 Bloomberg	24	1.00	0.416	0.739	-0.63
2022 City of Munich	34	1.00	0.414	1.242	+0.59
2022 Huawei	15	1.00	0.421	1.143	+0.34
2022 Hubert Burda Media	21	1.00	0.417	1.700	+1.68
2022 Interhyp	24	1.00	0.416	0.783	-0.52
2022 Rohde & Schwarz	12	1.00	0.423	0.455	-1.29
2022 Sixt	20	1.00	0.418	1.316	+0.76
2022 Solana	7	1.00	0.430	0.500	-1.16
2022 Wildtrack	10	1.00	0.426	1.000	+0.00
2024 CHECK24	30	1.00	0.415	0.966	-0.08
2024 City of Munich	31	1.00	0.415	1.000	+0.00
2024 Huawei	7	1.00	0.430	1.000	+0.00
2024 Hubert Burda Media	18	1.00	0.419	0.647	-0.84
2024 Interhyp	12	1.00	0.423	0.818	-0.43
2024 JetBrains	19	1.00	0.418	0.944	-0.13
2024 Siemens	9	1.00	0.427	1.625	+1.46
2024 Solana	20	1.00	0.418	0.737	-0.63
2024 T-Systems	32	1.00	0.414	1.516	+1.25
2024 Wildtrack	7	1.00	0.430	1.667	+1.55
2025 City of Munich	45	1.00	0.413	1.114	+0.28
2025 Huawei	10	1.00	0.426	1.222	+0.52
2025 IMC	29	1.00	0.415	0.643	-0.86
2025 Interhyp	19	1.00	0.418	0.222	-1.86
2025 JetBrains	21	1.00	0.417	1.800	+1.92
2025 Logitech	17	1.00	0.419	1.062	+0.15
2025 Reply	22	1.00	0.417	0.905	-0.23
2025 SAP	12	1.00	0.423	0.364	-1.50
2025 SIXT	27	1.00	0.415	1.808	+1.94
2025 Siemens	8	1.00	0.429	0.714	-0.67
2025 Wildtrack	12	1.00	0.423	0.455	-1.29
Implied SD of T (all winners): 0.0378; Monte Carlo estimate: 0.0378.					

C Session Atlas

For readers who prefer bars to tables, each panel shows one session; every bar is a pitch slot, winners are blue and labelled with their slot number.

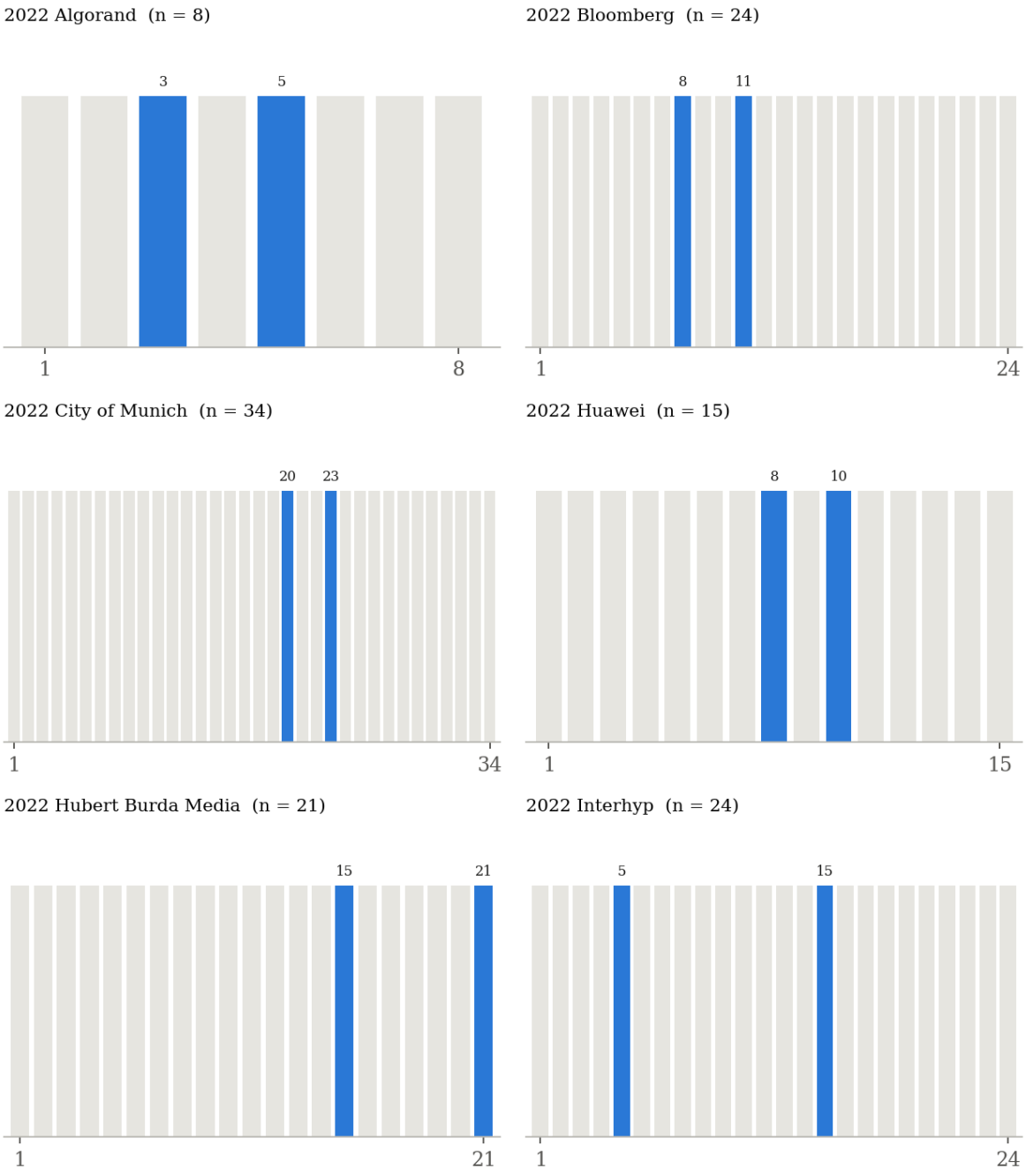


Figure 8: Session atlas, part 1 of 6.

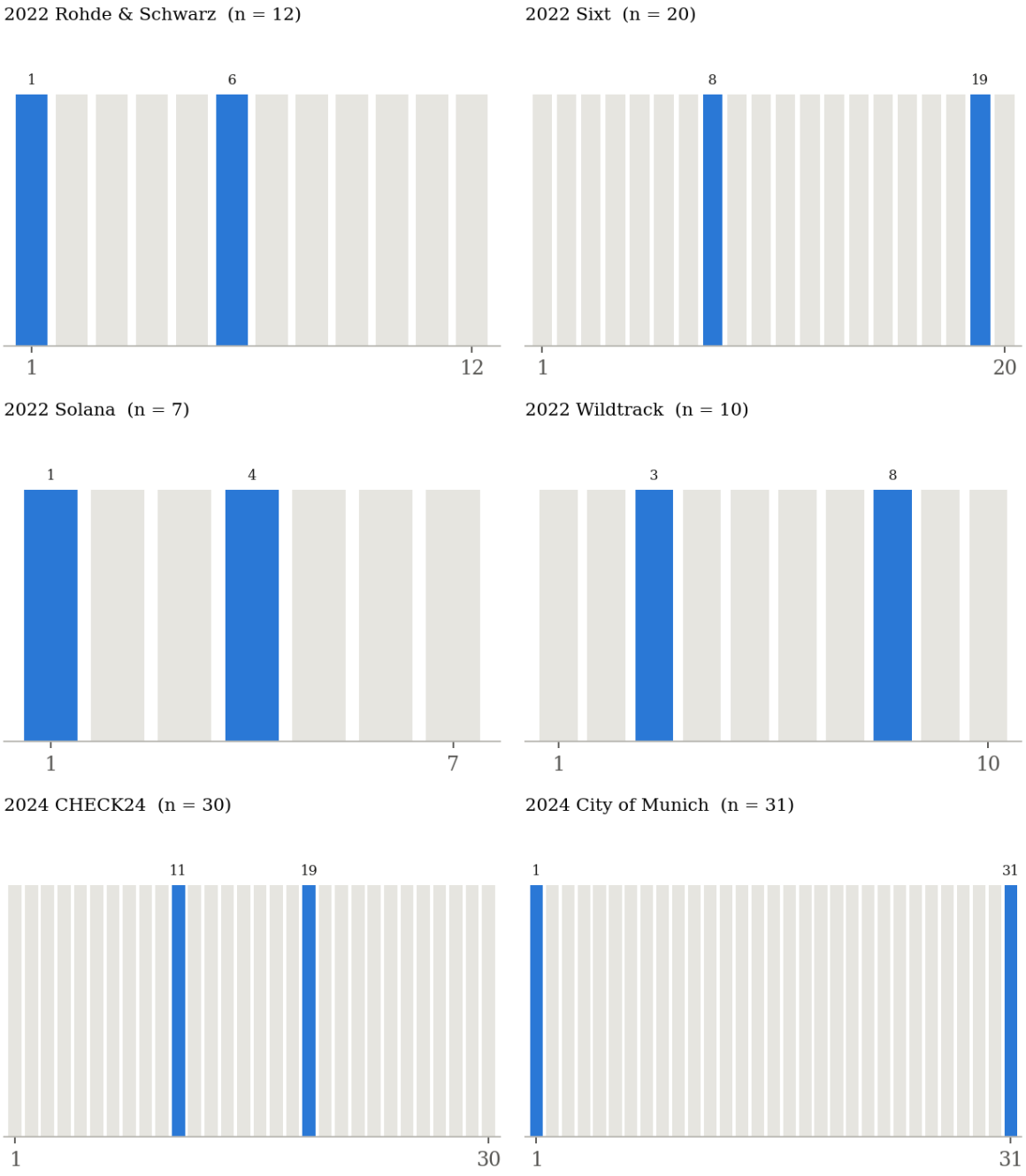


Figure 9: Session atlas, part 2 of 6.

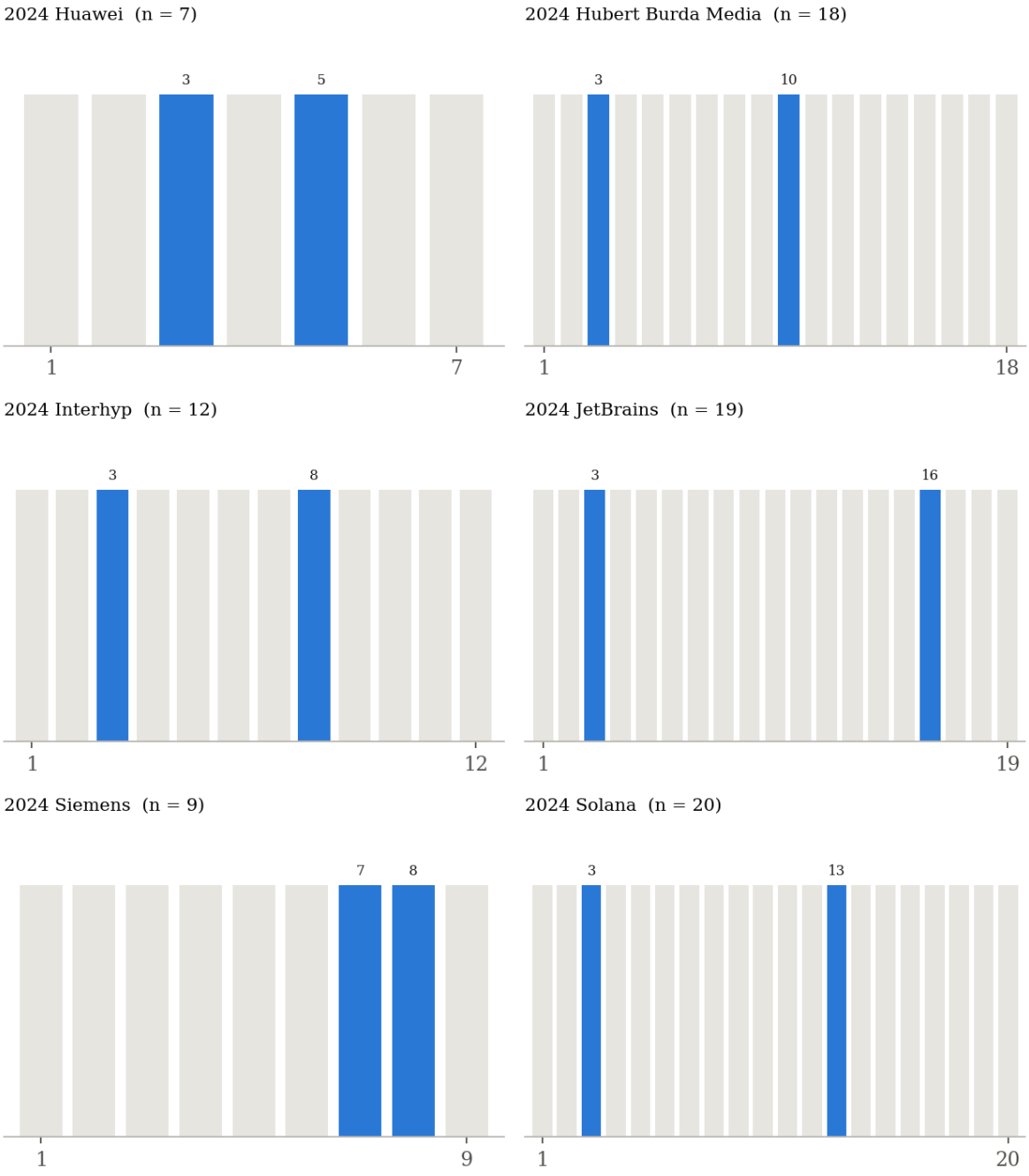


Figure 10: Session atlas, part 3 of 6.

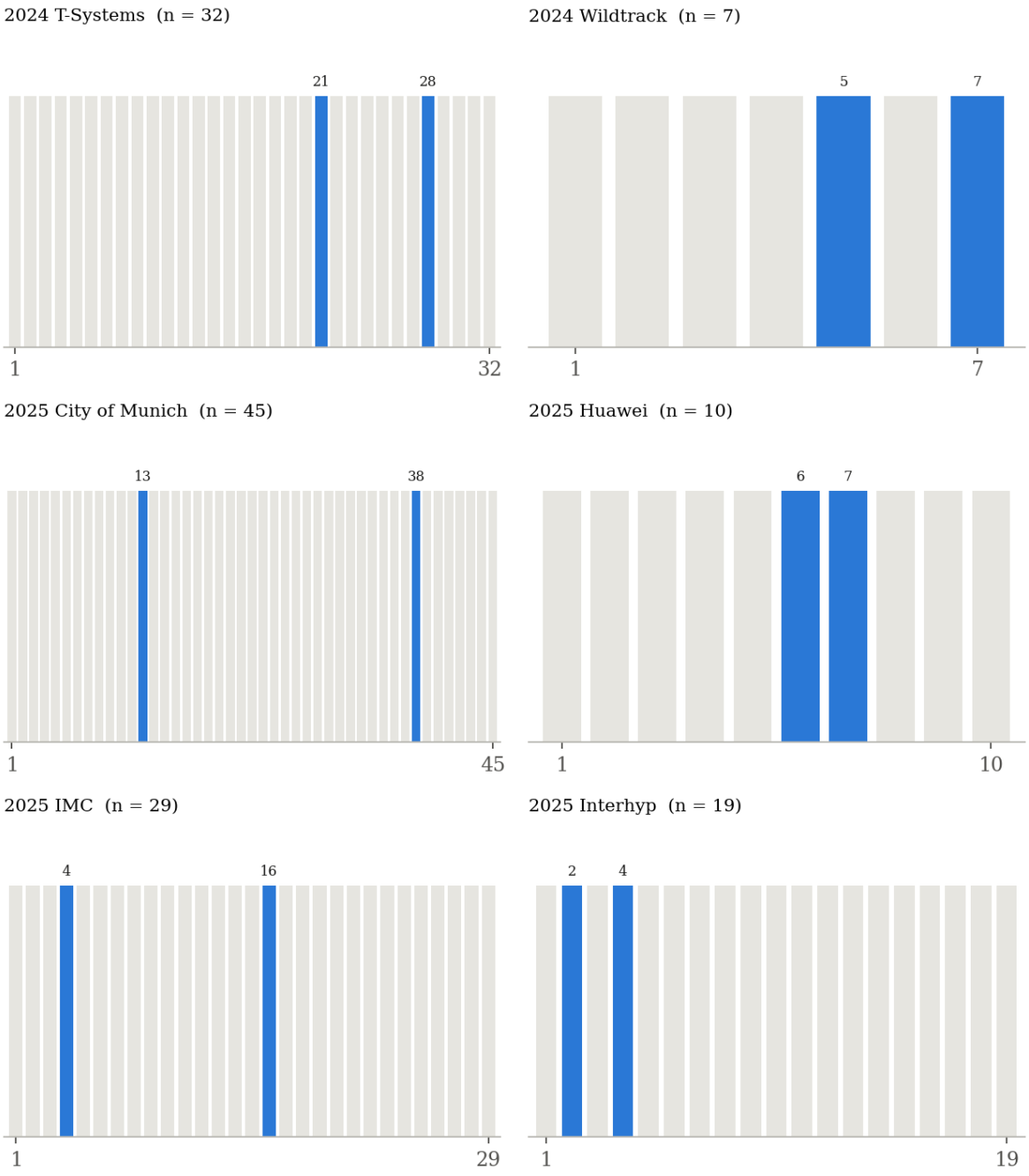


Figure 11: Session atlas, part 4 of 6.

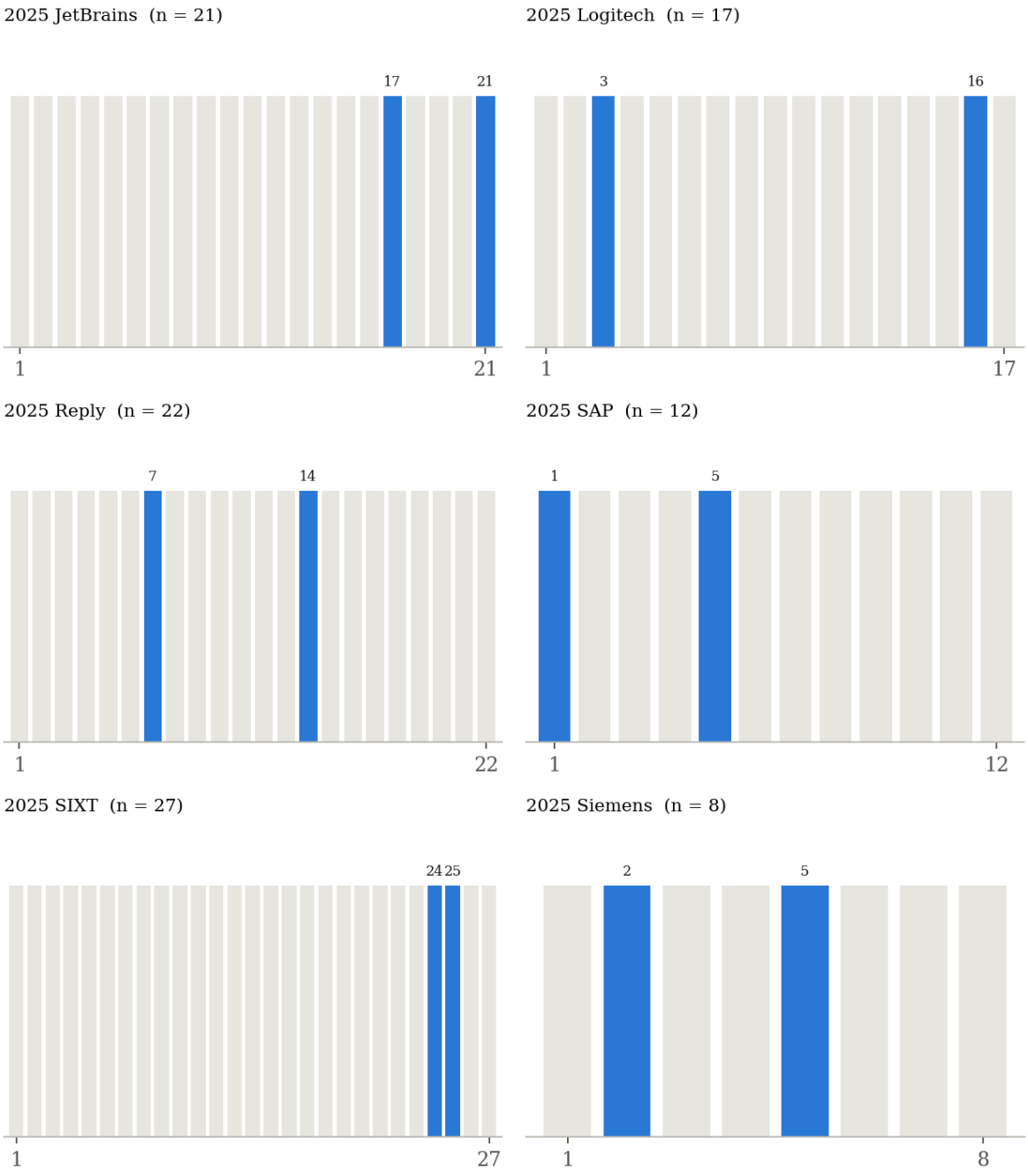


Figure 12: Session atlas, part 5 of 6.

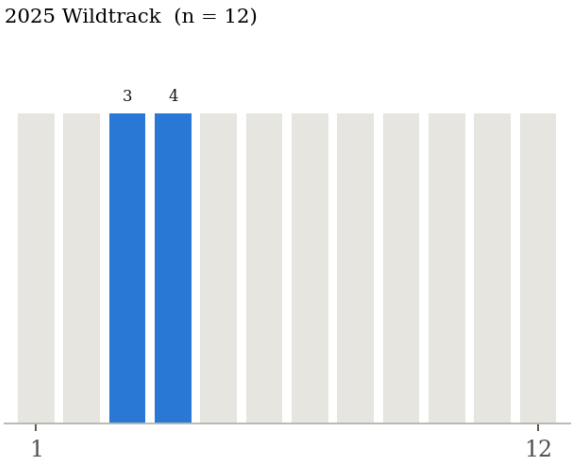


Figure 13: Session atlas, part 6 of 6.

D Source Code

In the interest of full reproducibility, and of page count, the complete analysis pipeline is reproduced below.

parse_schedules.py

```
"""Parse hackaTUM final-pitch schedule PDFs into data/schedule.csv."""
import csv, re, subprocess, sys
from pathlib import Path

ROOT = Path(__file__).resolve().parent.parent
PDFS = ROOT / "pdfs"
OUT = ROOT / "data" / "schedule.csv"

URL = re.compile(r"https://hackatum(\d\d)\.devpost\.com/submissions/(\d+)-(\S*)")
TIME = re.compile(r"^\s*(\d{1,2}):(\d{2})(?::\d{2})?(?:\s*-\s*\d{1,2}:\d{2})?")

def challenge_of(name: str) -> tuple[int, str]:
    if name.startswith("hackaTUM22-finalpitches-"):
        key = name.split("-")[-1]
        m = {"algorand": "Algorand", "bloomberg": "Bloomberg", "cityofmunich": "City␣of␣Munich",
             "huawei": "Huawei", "hubertburdamedia": "Hubert␣Burda␣Media", "interhyp": "Interhyp",
             "optiver": "Optiver", "wildtrack": "Wildtrack", "rohdeschwarz": "Rohde␣&␣Schwarz", "sixt": "
                 Sixt", "solana": "Solana"}
        return 2022, m[key]
    if "-hackaTUM24-" in name:
        return 2024, name.split("-hackaTUM24-")[0]
    if name.endswith("_pitch_schedule_FINAL"):
        c = name.replace("_pitch_schedule_FINAL", "")
        return 2025, {"CityofMunich": "City␣of␣Munich"}.get(c, c)
    raise ValueError(name)

rows = []
for pdf in sorted(PDFS.glob("*.pdf")):
    year, chal = challenge_of(pdf.stem)
    text = subprocess.run(["pdftotext", "-layout", str(pdf), "-"], capture_output=True, text=True, check=
         True).stdout
    cur_time, slot, after_break = None, 0, False
    for line in text.splitlines():
        if re.search(r"\bBREAK\b", line):
            after_break = True
        t = TIME.match(line)
        if t:
            cur_time = int(t.group(1)) * 60 + int(t.group(2))
        u = URL.search(line)
        if not u:
            continue
        slot += 1
        name = line[t.end():u.start()].strip() if t else line[:u.start()].strip()
        size = re.search(r"\s(\d)\s*$", line)
        rows.append(dict(year=year, challenge=chal, slot=slot, time_min=cur_time if t else "",
                         team=name, submission_id=u.group(2), submission_url=u.group(0),
                         team_size=size.group(1) if size else "", post_break=int(after_break)))
        after_break = False
    # Optiver 2022 is one 30-minute block with no individual slots -> flag
for r in rows:
    r["individual_slot"] = int(r["time_min"] != "")
# Group stats
from collections import Counter
n = Counter((r["year"], r["challenge"]) for r in rows)
for r in rows:
    r["n_slots"] = n[(r["year"], r["challenge"])]
```

```
OUT.parent.mkdir(exist_ok=True)
with OUT.open("w", newline="") as f:
    w = csv.DictWriter(f, fieldnames=list(rows[0]))
    w.writeheader(); w.writerows(rows)
for k, v in sorted(n.items()):
    untimed = sum(1 for r in rows if (r["year"], r["challenge"]) == k and not r["individual_slot"])
    print(k, v, f"untimed={untimed}" if untimed else "")
print(len(rows), "rows␣->", OUT)
```

resolve_slugs.py

```
"""Resolve each schedule submission URL to its devpost.com/software/<slug> (via redirect)."""
import csv, json, time, urllib.request
from concurrent.futures import ThreadPoolExecutor
from pathlib import Path
ROOT = Path(__file__).resolve().parent.parent
cache_p = ROOT / "data" / "slug_cache.json"
cache = json.loads(cache_p.read_text()) if cache_p.exists() else {}
urls = sorted({r["submission_url"] for r in csv.DictReader(open(ROOT / "data/schedule.csv"))} - set(cache)
    )

class NoRedirect(urllib.request.HTTPRedirectHandler):
    def redirect_request(self, *a, **k): return None
opener = urllib.request.build_opener(NoRedirect)

def get(u):
    for attempt in range(4):
        try:
            req = urllib.request.Request(u, method="HEAD", headers={"User-Agent": "Mozilla/5.0"})
            opener.open(req, timeout=20)
            return u, None
        except urllib.error.HTTPError as e:
            if e.code in (301, 302):
                return u, e.headers["Location"]
            if e.code == 429: time.sleep(5 * (attempt + 1)); continue
            return u, f"HTTP{e.code}"
        except Exception as e:
            time.sleep(2)
    return u, "ERR"

with ThreadPoolExecutor(4) as ex:
    for i, (u, loc) in enumerate(ex.map(get, urls)):
        cache[u] = loc
        if i % 50 == 0:
            cache_p.write_text(json.dumps(cache, indent=0)); print(i, len(urls), flush=True)
cache_p.write_text(json.dumps(cache, indent=0))
bad = {u: v for u, v in cache.items() if not (v or "").startswith("https://devpost.com/software/")}
print("done", len(cache), "unresolved:", bad)
```

build_dataset.py

```
"""Join schedule slots with Devpost winners -> data/slots.csv (one row per pitch slot)."""
import csv, json, re
from pathlib import Path
ROOT = Path(__file__).resolve().parent.parent
D = ROOT / "data"
cache = json.loads((D / "slug_cache.json").read_text())
rows = list(csv.DictReader(open(D / "schedule.csv")))
norm = lambda s: re.sub(r"[^a-z0-9]", "", s.lower())
# 1st/2nd place where an official source states it (2022: TUM CIT news article "Eat, code, drink, repeat")
PLACE = json.loads((D / "placements.json").read_text()) if (D / "placements.json").exists() else {}
```

```
for r in rows:
    r.update(winner=0, award_name="", place="", overall=0, popular=0, match="")
unmatched = []
for y in (22, 24, 25):
    for w in json.loads((D / f"winners{y}.json").read_text())["itemListElement"]:
        slug = w["url"].rstrip("/").split("/")[-1]
        chals = [a.split(",␣", 1)[1].split("␣␣")[0] for a in w["award"]]
        for ch in chals:
            if ch in ("Overall␣Winner", "Popular␣Vote"):
                continue
            cands = [r for r in rows if r["year"] == f"20{y}" and r["challenge"].lower() == ch.lower()]
            if not cands:
                unmatched.append((y, ch, w["name"], "no␣schedule")); continue
            hit = [r for r in cands if (cache.get(r["submission_url"]) or "").rstrip("/").endswith("/" +
                 slug)]
            how = "redirect"
            if not hit:
                how = "name/slug"
                hit = [r for r in cands if r["submission_url"].split("/")[-1].split("-", 1)[1] == slug
                       or norm(r["team"]) == norm(w["name"]) or norm(w["name"]).startswith(norm(r["team"]))
                       or norm(r["team"]) in norm(w["name"])]
            if len(hit) != 1:
                unmatched.append((y, ch, w["name"], [h["team"] for h in hit])); continue
            h = hit[0]
            h.update(winner=1, award_name=w["name"], match=how,
                     overall=int("Overall␣Winner" in chals), popular=int("Popular␣Vote" in chals),
                     place=PLACE.get(f"20{y}|{ch}|{w['name']}", ""))
# exclude challenges where order is not informative
for r in rows:
    r["excluded"] = "single␣30-min␣block,␣no␣individual␣order" if (r["year"], r["challenge"]) == ("2022", "
         Optiver") else ""
    n = int(r["n_slots"]); s = int(r["slot"])
    r["pos01"] = round((s - 1) / (n - 1), 4) # 0 = first, 1 = last
with open(D / "slots.csv", "w", newline="") as f:
    w = csv.DictWriter(f, fieldnames=list(rows[0])); w.writeheader(); w.writerows(rows)
print("winners␣matched:", sum(int(r["winner"]) for r in rows), "|␣unmatched:", unmatched)
```

analyze.py

```
"""Does pitch slot predict winning a hackaTUM partner challenge?

Unit of analysis: one (year, challenge) pitch session = one stratum.
Null hypothesis H0: within each stratum, the k winners are a uniformly random subset of the
pitch slots (i.e. slot carries no information). All p-values come from permutation under H0,
which handles small sessions and sampling without replacement exactly.

PRIMARY test (pre-declared): mean normalized winner position (0 = first pitch, 1 = last), two-sided.
Everything else is secondary/exploratory; secondary p-values are Holm-corrected as a family.
"""
import csv, json
from collections import defaultdict
from pathlib import Path

import numpy as np
from scipy import stats

ROOT = Path(__file__).resolve().parent.parent
D, OUT = ROOT / "data", ROOT / "results"
OUT.mkdir(exist_ok=True)
rng = np.random.default_rng(2025)
NPERM = 200_000

rows = [r for r in csv.DictReader(open(D / "slots.csv")) if not r["excluded"]]
```

```
for r in rows:
    for k in ("slot", "n_slots", "winner", "post_break", "submission_id"):
        r[k] = int(r[k])
    r["pos01"] = float(r["pos01"])
    r["key"] = f'{r["year"]}␣{r["challenge"]}'

# teams that pitched in more than one session in the same year (same Devpost submission)
cache = json.loads((D / "slug_cache.json").read_text())
proj = lambda r: r["submission_id"] # Devpost submission IDs are stable per project per hackathon
cnt = defaultdict(set)
for r in rows:
    cnt[(r["year"], proj(r))].add(r["challenge"])
for r in rows:
    r["multi"] = int(len(cnt[(r["year"], proj(r))]) > 1)

judging = json.loads((D / "challenge_judging.json").read_text()) if (D / "challenge_judging.json").exists
     () else {}


def strata(rs, value="pos01"):
    """-> list of (values array, winner mask) per stratum."""
    g = defaultdict(list)
    for r in rs:
        g[r["key"]].append(r)
    out = []
    for k, lst in sorted(g.items()):
        lst.sort(key=lambda r: r["slot"])
        w = np.array([r["winner"] for r in lst], bool)
        if w.sum() == 0 or w.all():
            continue
        out.append((k, np.array([r[value] for r in lst], float), w))
    return out


def perm_sum(S, nperm=NPERM):
    """Permutation null of sum over strata of f(winner values). Returns (obs, null array) for sum of values.
         """
    obs = sum(v[w].sum() for _, v, w in S)
    null = np.zeros(nperm)
    for _, v, w in S:
        k, n = w.sum(), len(v)
        # random k-subset per replicate via argsort of uniforms (vectorized)
        idx = np.argpartition(rng.random((nperm, n)), k, axis=1)[:, :k]
        null += v[idx].sum(1)
    return obs, null


def p_two(obs, null):
    c = null.mean()
    return float((np.abs(null - c) >= abs(obs - c) - 1e-12).mean())


def mean_pos_test(S, label):
    nw = sum(w.sum() for _, _, w in S)
    obs, null = perm_sum(S)
    exp = null.mean() / nw
    # bootstrap CI over strata (resample whole sessions)
    per = [(v[w].sum(), w.sum()) for _, v, w in S]
    B = np.array([sum(per[i][0] for i in ix) / sum(per[i][1] for i in ix)
                  for ix in rng.integers(0, len(per), (20_000, len(per)))])
    return dict(subset=label, sessions=len(S), winners=int(nw), mean_winner_pos=obs / nw, expected=exp,
                ci95=[float(np.percentile(B, 2.5)), float(np.percentile(B, 97.5))],
                null_sd=float(null.std() / nw), p_two_sided=p_two(obs, null), _null=null / nw)
```

```
res = {}
S = strata(rows)
prim = mean_pos_test(S, "all")
res["primary"] = {k: v for k, v in prim.items() if not k.startswith("_")}
# minimum detectable shift in mean winner position at alpha=.05 two-sided, 80% power (normal approx)
res["primary"]["mde_80pct_power"] = float((1.96 + 0.84) * prim["null_sd"])
np.save(OUT / "null_primary.npy", prim["_null"])

# ---------- secondary family ----------
sec = {}

# (a) early vs late half
def half_val(r):
    n, s = r["n_slots"], r["slot"]
    return 1.0 if s > n - n // 2 else (0.0 if s <= n // 2 else np.nan) # middle slot of odd n excluded
for r in rows:
    r["late_half"] = half_val(r)
Sh = [(k, v[~np.isnan(v)], w[~np.isnan(v)]) for k, v, w in strata(rows, "late_half")]
obs, null = perm_sum(Sh)
nw = sum(w.sum() for _, _, w in Sh)
sec["late_half_share"] = dict(observed_late=int(obs), winners=int(nw), expected_late=float(null.mean()),
                              p=p_two(obs, null))

# (b) terciles: observed vs expected, chi-square-type statistic, permutation p
for r in rows:
    u = (r["slot"] - 0.5) / r["n_slots"]
    r["terc"] = 0 if u < 1 / 3 else (1 if u < 2 / 3 else 2)
St = strata(rows, "terc")
O = np.zeros(3); E = np.zeros(3)
for _, v, w in St:
    for t in range(3):
        O[t] += (v[w] == t).sum(); E[t] += w.sum() * (v == t).mean()
chi = ((O - E) ** 2 / E).sum()
nullchi = np.zeros(NPERM // 4)
cnts = np.zeros((NPERM // 4, 3))
for _, v, w in St:
    k, n = w.sum(), len(v)
    idx = np.argpartition(rng.random((NPERM // 4, n)), k, axis=1)[:, :k]
    picked = v[idx]
    for t in range(3):
        cnts[:, t] += (picked == t).sum(1)
nullchi = ((cnts - E) ** 2 / E).sum(1)
sec["terciles"] = dict(observed=O.tolist(), expected=E.round(2).tolist(), stat=float(chi),
                       p=float((nullchi >= chi - 1e-12).mean()))

# (c) first slot / last slot (primacy / recency)
for name, fn in (("first_slot", lambda r: float(r["slot"] == 1)), ("last_slot", lambda r: float(r["slot"]
     == r["n_slots"]))):
    for r in rows:
        r[name] = fn(r)
    obs, null = perm_sum(strata(rows, name))
    sec[name] = dict(observed=int(obs), expected=float(null.mean()), p=p_two(obs, null),
                     p_upper=float((null >= obs).mean()))

# (d) edges (first or last) vs middle U-shape
for r in rows:
    r["edge"] = float(r["slot"] in (1, r["n_slots"]))
obs, null = perm_sum(strata(rows, "edge"))
sec["edge_first_or_last"] = dict(observed=int(obs), expected=float(null.mean()), p=p_two(obs, null))

# (e) first pitch after a break
obs, null = perm_sum(strata([r for r in rows if r["key"] in {x["key"] for x in rows if x["post_break"]}],
     "post_break"))
sec["first_after_break"] = dict(observed=int(obs), expected=float(null.mean()), p=p_two(obs, null),
                                note="only␣sessions␣that␣had␣a␣break")
```

```
# (f) conditional logit win ~ pos + pos^2 (strata = session)
try:
    from statsmodels.discrete.conditional_models import ConditionalLogit
    y = np.array([r["winner"] for r in rows]); g = np.array([r["key"] for r in rows])
    x = np.array([r["pos01"] for r in rows])
    m1 = ConditionalLogit(y, x[:, None], groups=g).fit(disp=0)
    m2 = ConditionalLogit(y, np.c_[x - .5, (x - .5) ** 2], groups=g).fit(disp=0)
    sec["clogit_linear"] = dict(beta=float(m1.params[0]), se=float(m1.bse[0]), p=float(m1.pvalues[0]),
                                odds_ratio_last_vs_first=float(np.exp(m1.params[0])),
                                or_ci95=np.exp(m1.conf_int()[0]).tolist())
    sec["clogit_quadratic"] = dict(beta_lin=float(m2.params[0]), beta_quad=float(m2.params[1]),
                                   p_quad=float(m2.pvalues[1]), note="beta_quad>0 = U-shape")
except Exception as e: # pragma: no cover
    sec["clogit_error"] = str(e)

# (g) is pitch order itself random? Spearman(slot, submission_id) per session
rhos = []
for k, v, w in strata(rows, "submission_id"):
    rho = stats.spearmanr(np.arange(len(v)), v).statistic
    rhos.append((k, float(rho), len(v)))
zs = [np.arctanh(np.clip(r, -.999, .999)) * np.sqrt(n - 3) for _, r, n in rhos if n > 3]
sec["order_vs_submission_id"] = dict(per_session={k: round(r, 2) for k, r, _ in rhos},
                                     median_rho=float(np.median([r for _, r, _ in rhos])),
                                     stouffer_z=float(np.sum(zs) / np.sqrt(len(zs))),
                                     p=float(2 * stats.norm.sf(abs(np.sum(zs) / np.sqrt(len(zs))))))

# (h) does submission order (Devpost id rank within session) predict winning? (confound check)
for r in rows:
    pass
g = defaultdict(list)
for r in rows:
    g[r["key"]].append(r)
for lst in g.values():
    ids = sorted(r["submission_id"] for r in lst)
    for r in lst:
        r["id_rank01"] = ids.index(r["submission_id"]) / (len(lst) - 1)
obs, null = perm_sum(strata(rows, "id_rank01"))
nw = sum(w.sum() for _, _, w in strata(rows, "id_rank01"))
sec["winner_submission_id_rank"] = dict(mean=obs / nw, expected=float(null.mean() / nw), p=p_two(obs, null
    ),
                                        note="0 = earliest-created Devpost submission in the session")

# Holm over secondary p-values
fam = [(k, v["p"]) for k, v in sec.items() if isinstance(v, dict) and "p" in v]
fam.sort(key=lambda t: t[1])
m = len(fam); run = 0
for i, (k, p) in enumerate(fam):
    run = max(run, min(1, (m - i) * p)); sec[k]["p_holm"] = run
res["secondary"] = sec


# ---------- robustness (primary statistic on subsets) ----------
rob = []
subsets = {
    "drop multi-session teams": [r for r in rows if not r["multi"]],
    "drop 2022 Solana (ambiguous label)": [r for r in rows if r["key"] != "2022 Solana"],
    "sessions with >= 15 pitches": [r for r in rows if r["n_slots"] >= 15],
    "sessions with < 15 pitches": [r for r in rows if r["n_slots"] < 15],
    "2022": [r for r in rows if r["year"] == "2022"],
    "2024": [r for r in rows if r["year"] == "2024"],
    "2025": [r for r in rows if r["year"] == "2025"],
}
if judging:
```

```
    subsets["objectively␣scored␣challenges"] = [r for r in rows if judging.get(r["key"]) in ("OBJECTIVE", "
        MIXED")]
    subsets["pitch-judged␣challenges"] = [r for r in rows if judging.get(r["key"]) not in ("OBJECTIVE", "
        MIXED")]
for lab, rs in subsets.items():
    Ss = strata(rs)
    if Ss:
        t = mean_pos_test(Ss, lab); t.pop("_null"); rob.append(t)
res["robustness"] = rob

# ---------- absolute clock time (exploratory, added after the fact; own Holm family) ----------
clock = {}
for r in rows:
    r["clock"] = float(int(r["time_min"]))
    r["elapsed_h"] = (int(r["time_min"]) - min(int(x["time_min"]) for x in rows if x["key"] == r["key"])) /
         60
obs, null = perm_sum(strata(rows, "clock"))
nw = sum(r["winner"] for r in rows)
fmt = lambda m: f"{int(round(m))␣//␣60}:{int(round(m))␣%␣60:02d}"
clock["mean_winner_clock"] = dict(observed=obs / nw, expected=float(null.mean() / nw), observed_hhmm=fmt(
    obs / nw),
                                  expected_hhmm=fmt(null.mean() / nw), sd_min=float(null.std() / nw), p=p_two(
                                      obs, null))
obs, null = perm_sum(strata(rows, "elapsed_h"))
clock["mean_winner_elapsed_min"] = dict(observed=60 * obs / nw, expected=float(60 * null.mean() / nw), p=
    p_two(obs, null))
# half-hour bins by clock time
BINS = [(0, 645, "10:1510:44"), (645, 675, "10:4511:14"), (675, 705, "11:1511:44"), (705, 735, "11:4512:14
    "), (735, 9999, "12:1513:15")]
for r in rows:
    r["cbin"] = next(i for i, (a, b, _) in enumerate(BINS) if a <= int(r["time_min"]) < b)
Sb = strata(rows, "cbin")
O = np.zeros(len(BINS)); E = np.zeros(len(BINS)); N = np.zeros(len(BINS))
for _, v, w in Sb:
    for t in range(len(BINS)):
        O[t] += (v[w] == t).sum(); E[t] += w.sum() * (v == t).mean(); N[t] += (v == t).sum()
stat = ((O - E) ** 2 / E).sum()
cnt = np.zeros((NPERM // 4, len(BINS)))
for _, v, w in Sb:
    k, n = w.sum(), len(v)
    picked = v[np.argpartition(rng.random((NPERM // 4, n)), k, axis=1)[:, :k]]
    for t in range(len(BINS)):
        cnt[:, t] += (picked == t).sum(1)
nullstat = ((cnt - E) ** 2 / E).sum(1)
clock["half_hour_bins"] = dict(labels=[b[2] for b in BINS], pitches=N.tolist(), observed=O.tolist(),
    expected=E.round(2).tolist(),
                               naive_win_rate=(O / N).round(3).tolist(), fair_win_rate=(E / N).round(3).
                                   tolist(),
                               stat=float(stat), p=float((nullstat >= stat - 1e-12).mean()))
# from 12:00 on ("before lunch")
for r in rows:
    r["noon"] = float(int(r["time_min"]) >= 720)
Sn = strata([r for r in rows if r["key"] in {x["key"] for x in rows if int(x["time_min"]) >= 720}], "noon"
    )
obs, null = perm_sum(Sn)
clock["from_noon"] = dict(observed=int(obs), expected=float(null.mean()), p=p_two(obs, null),
                          slots=int(sum(r["noon"] for r in rows)), sessions=len(Sn), note="sessions␣that␣
                              reached␣12:00")
try:
    y = np.array([r["winner"] for r in rows]); gg = np.array([r["key"] for r in rows])
    mh = ConditionalLogit(y, np.array([r["elapsed_h"] for r in rows])[:, None], groups=gg).fit(disp=0)
    clock["clogit_per_hour"] = dict(beta=float(mh.params[0]), se=float(mh.bse[0]), p=float(mh.pvalues[0]),
                                    odds_ratio_per_hour=float(np.exp(mh.params[0])), or_ci95=np.exp(mh.
                                        conf_int()[0]).tolist())
except Exception as e:
```

```
    clock["clogit_error"] = str(e)
fam = sorted([(k, v["p"]) for k, v in clock.items() if isinstance(v, dict) and "p" in v], key=lambda t: t
     [1])
run = 0
for i, (k, pp) in enumerate(fam):
    run = max(run, min(1, (len(fam) - i) * pp)); clock[k]["p_holm"] = run
res["clock_time"] = clock


# ---------- 1st vs 2nd place (where ranking known) ----------
pl = defaultdict(dict)
for r in rows:
    if r["place"]:
        pl[r["key"]][int(r["place"])] = r["pos01"]
pairs = [(k, d[1], d[2]) for k, d in pl.items() if 1 in d and 2 in d]
later = sum(a > b for _, a, b in pairs)
res["first_vs_second_place"] = dict(pairs=len(pairs), first_pitched_later=later,
                                    p_sign_test=float(stats.binomtest(later, len(pairs)).pvalue),
                                    detail={k: dict(first=a, second=b) for k, a, b in pairs})

# ---------- overall winners (descriptive) ----------
res["overall_winners"] = [dict(session=r["key"], team=r["award_name"], slot=r["slot"], of=r["n_slots"])
                          for r in rows if r["overall"] == "1"]

# per-session table
res["sessions"] = [dict(session=k, n=len(v), winner_slots=[int(i) + 1 for i in np.where(w)[0]])
                   for k, v, w in S]
(OUT / "results.json").write_text(json.dumps(res, indent=1, default=float, ensure_ascii=False))
with open(OUT / "winner_positions.csv", "w", newline="") as f:
    wr = csv.writer(f); wr.writerow(["session", "team", "slot", "n_slots", "pos01", "place", "
         multi_session_team"])
    for r in sorted(rows, key=lambda r: (r["key"], r["slot"])):
        if r["winner"]:
            wr.writerow([r["key"], r["award_name"], r["slot"], r["n_slots"], r["pos01"], r["place"], r["
                 multi"]])

p = res["primary"]
print(f"PRIMARY:␣{p['winners']}␣winners␣in␣{p['sessions']}␣sessions␣|␣mean␣winner␣position␣{p['
     mean_winner_pos']:.3f}␣"
      f"(expected␣{p['expected']:.3f}),␣95%␣CI␣{p['ci95'][0]:.3f}-{p['ci95'][1]:.3f},␣p={p['p_two_sided
           ']:.3f},␣"
      f"MDE␣±{p['mde_80pct_power']:.3f}")
for k, v in sec.items():
    print("␣", k, {kk: (round(vv, 3) if isinstance(vv, float) else vv) for kk, vv in v.items() if kk != "
         per_session"})
for t in rob:
    print("␣␣ROB", t["subset"], t["sessions"], t["winners"], round(t["mean_winner_pos"], 3), [round(c, 3)
         for c in t["ci95"]], round(t["p_two_sided"], 3))
print("␣", res["first_vs_second_place"]["pairs"], "ranked␣pairs;␣1st␣pitched␣later␣in", later, "p=", round
     (res["first_vs_second_place"]["p_sign_test"], 3))
print("␣", res["overall_winners"])
for k, v in clock.items(): print("␣␣CLOCK", k, v)
```

figures.py

```
"""Figures for the pitch-order analysis -> results/*.png"""
import csv, json
from pathlib import Path
import numpy as np
import matplotlib
matplotlib.use("Agg")
import matplotlib.pyplot as plt

ROOT = Path(__file__).resolve().parent.parent
```

```
OUT = ROOT / "results"
res = json.loads((OUT / "results.json").read_text())
null = np.load(OUT / "null_primary.npy")
W = list(csv.DictReader(open(OUT / "winner_positions.csv")))

BLUE, ORANGE, GRAY, INK, INK2, SURF = "#2a78d6", "#eb6834", "#b8b7b1", "#0b0b0b", "#52514e", "#fcfcfb"
plt.rcParams.update({"font.family": "DejaVu Sans", "font.size": 10, "axes.edgecolor": GRAY, "axes.
    labelcolor": INK2,
                 "xtick.color": INK2, "ytick.color": INK2, "axes.spines.top": False, "axes.spines.right"
                     : False,
                 "figure.facecolor": SURF, "axes.facecolor": SURF, "savefig.facecolor": SURF})

# 1. Every session as a row: all slots (gray) + winners (blue; 1st place ringed orange)
sess = res["sessions"]
fig, ax = plt.subplots(figsize=(9, 9))
for i, s in enumerate(sess):
    n = s["n"]; xs = np.linspace(0, 1, n) if n > 1 else [0.5]
    ax.plot([0, 1], [i, i], color="#e6e5e0", lw=1, zorder=0)
    ax.scatter(xs, [i] * n, s=10, color=GRAY, zorder=1, lw=0)
    for w in s["winner_slots"]:
        rec = next(r for r in W if r["session"] == s["session"] and int(r["slot"]) == w)
        ax.scatter([(w - 1) / (n - 1)], [i], s=70, color=BLUE, zorder=3, edgecolor=SURF, lw=2)
        if rec["place"] == "1":
            ax.scatter([(w - 1) / (n - 1)], [i], s=150, facecolor="none", edgecolor=ORANGE, lw=2, zorder=4)
ax.set_yticks(range(len(sess)), [f'{s["session"]}  (n={s["n"]})' for s in sess], fontsize=8)
ax.set_xticks([0, .25, .5, .75, 1], ["first pitch", "", "middle", "", "last pitch"])
ax.invert_yaxis()
ax.set_title("Where the winners pitched, session by session", loc="left", color=INK, fontsize=13, pad=14)
ax.text(0, -1.6, "Gray = every pitch slot, blue = challenge winner, orange ring = confirmed 1st place
    (2022, SAP 2025)",
        fontsize=8.5, color=INK2, transform=ax.transData)
fig.tight_layout(); fig.savefig(OUT / "fig1_sessions.png", dpi=160); plt.close(fig)

# 2. Observed vs expected winners by third of the schedule
t = res["secondary"]["terciles"]
fig, ax = plt.subplots(figsize=(7, 4))
x = np.arange(3); bw = .36
ax.bar(x - bw / 2 - .01, t["expected"], bw, color=GRAY, label="Expected if order doesn't matter")
ax.bar(x + bw / 2 + .01, t["observed"], bw, color=BLUE, label="Observed winners")
for xi, o, e in zip(x, t["observed"], t["expected"]):
    ax.text(xi + bw / 2 + .01, o + .4, f"{o:.0f}", ha="center", color=INK, fontsize=10)
    ax.text(xi - bw / 2 - .01, e + .4, f"{e:.1f}", ha="center", color=INK2, fontsize=9)
ax.set_xticks(x, ["Early third", "Middle third", "Late third"])
ax.set_ylabel("Winners (60 total)")
ax.set_ylim(0, 30); ax.legend(frameon=False, loc="upper right", fontsize=9)
ax.set_title(f"Winners by part of the schedule (permutation p = {t['p']:.2f})", loc="left", color=INK,
    fontsize=12)
fig.tight_layout(); fig.savefig(OUT / "fig2_terciles.png", dpi=160); plt.close(fig)

# 3. Permutation null of mean winner position with observed value
p = res["primary"]
fig, ax = plt.subplots(figsize=(7, 3.8))
ax.hist(null, bins=80, color=GRAY, edgecolor=SURF, lw=.4)
ax.axvline(p["mean_winner_pos"], color=BLUE, lw=2)
ax.axvspan(*p["ci95"], color=BLUE, alpha=.08, lw=0)
ax.text(p["mean_winner_pos"] + .006, ax.get_ylim()[1] * .9, f"observed {p['mean_winner_pos']:.3f}\n95% CI
    {p['ci95'][0]:.2f}{p['ci95'][1]:.2f}",
        color=INK, fontsize=9, va="top")
ax.set_xlabel("Mean winner position (0 = first pitch, 1 = last)")
ax.set_yticks([])
ax.spines["left"].set_visible(False)
ax.set_title(f"Observed vs. 200,000 random shuffles of who won (p = {p['p_two_sided']:.2f})", loc="left",
    color=INK, fontsize=12)
fig.tight_layout(); fig.savefig(OUT / "fig3_permutation.png", dpi=160); plt.close(fig)
print("figures written")
```

```
# 4. Clock time: win rate per half-hour, observed vs expected given which rooms were still running
c = res["clock_time"]["half_hour_bins"]
fig, ax = plt.subplots(figsize=(7.5, 4))
x = np.arange(len(c["labels"])); bw = .36
ax.bar(x - bw / 2 - .01, np.array(c["fair_win_rate"]) * 100, bw, color=GRAY, label="Expected␣from␣room␣
    sizes␣alone")
ax.bar(x + bw / 2 + .01, np.array(c["naive_win_rate"]) * 100, bw, color=BLUE, label="Observed")
for xi, o, e, n, w in zip(x, c["naive_win_rate"], c["fair_win_rate"], c["pitches"], c["observed"]):
    ax.text(xi + bw / 2 + .01, o * 100 + .3, f"{o*100:.0f}%", ha="center", fontsize=9, color=INK)
ax.set_xticks(x, [f"{l}\n{int(w)}␣of␣{int(n)}␣won" for l, w, n in zip(c["labels"], c["observed"], c["
    pitches"])], fontsize=9)
ax.set_ylabel("Share␣of␣pitches␣that␣won␣(%)"); ax.set_ylim(0, 17)
ax.legend(frameon=False, fontsize=9, loc="upper␣right")
ax.set_title(f"Win␣rate␣by␣clock␣time␣(permutation␣p␣=␣{c['p']:.2f})", loc="left", color=INK, fontsize=12)
fig.tight_layout(); fig.savefig(OUT / "fig4_clock.png", dpi=160); plt.close(fig)
print("clock␣figure␣written")
```

paper__assets.py

```
"""Extra figures + generated appendices for the paper -> paper/figures, paper/generated."""
import csv, json, re, shutil
from collections import defaultdict
from pathlib import Path
import numpy as np
import matplotlib
matplotlib.use("Agg")
import matplotlib.pyplot as plt

ROOT = Path(__file__).resolve().parent.parent
P = ROOT / "paper"; FIG = P / "figures"; GEN = P / "generated"
FIG.mkdir(parents=True, exist_ok=True); GEN.mkdir(parents=True, exist_ok=True)
res = json.loads((ROOT / "results/results.json").read_text())
rows = [r for r in csv.DictReader(open(ROOT / "data/slots.csv")) if not r["excluded"]]
for r in rows:
    r["slot"] = int(r["slot"]); r["n_slots"] = int(r["n_slots"]); r["winner"] = int(r["winner"]); r["pos01"
        ] = float(r["pos01"])
    r["key"] = f'{r["year"]}␣{r["challenge"]}'

BLUE, ORANGE, GRAY, INK, INK2 = "#2a78d6", "#eb6834", "#b8b7b1", "#141413", "#52514e"
plt.rcParams.update({"font.family": "serif", "font.size": 10, "axes.spines.top": False, "axes.spines.right
    ": False,
                     "axes.edgecolor": GRAY, "axes.labelcolor": INK2, "xtick.color": INK2, "ytick.color":
                         INK2})

for f in ("fig1_sessions", "fig2_terciles", "fig3_permutation", "fig4_clock"):
    shutil.copy(ROOT / f"results/{f}.png", FIG / f"{f}.png")

# ECDF
w = np.sort([r["pos01"] for r in rows if r["winner"]])
fig, ax = plt.subplots(figsize=(6, 4))
ax.plot([0, 1], [0, 1], color=GRAY, lw=1.5, ls="--", label="Uniform␣(H$_0$)")
ax.step(w, np.arange(1, len(w) + 1) / len(w), where="post", color=BLUE, lw=2, label="Winners␣(n␣=␣60)")
ax.set_xlabel("Normalized␣pitch␣position"); ax.set_ylabel("Cumulative␣share␣of␣winners"); ax.legend(
    frameon=False)
from scipy import stats
ks = stats.kstest(w, "uniform")
ax.set_title(f"Empirical␣CDF␣of␣winner␣positions␣(KS␣D␣=␣{ks.statistic:.3f})", loc="left", fontsize=11)
fig.tight_layout(); fig.savefig(FIG / "fig_ecdf.png", dpi=200); plt.close(fig)

# Order vs submission id rank
g = defaultdict(list)
for r in rows: g[r["key"]].append(r)
xs, ys, ws = [], [], []
```

```
for lst in g.values():
    ids = sorted(int(r["submission_id"]) for r in lst)
    for r in lst:
        xs.append(r["pos01"]); ys.append(ids.index(int(r["submission_id"])) / (len(lst) - 1)); ws.append(r[
            "winner"])
xs, ys, ws = map(np.array, (xs, ys, ws))
fig, ax = plt.subplots(figsize=(6, 5))
ax.scatter(xs[ws == 0], ys[ws == 0], s=10, color=GRAY, lw=0, label="Pitch␣slot")
ax.scatter(xs[ws == 1], ys[ws == 1], s=30, color=BLUE, edgecolor="white", lw=.8, label="Winner")
b = np.polyfit(xs, ys, 1); t = np.linspace(0, 1, 2)
ax.plot(t, b[1] + b[0] * t, color=ORANGE, lw=2, label="OLS␣fit␣(descriptive)")
ax.set_xlabel("Normalized␣pitch␣position"); ax.set_ylabel("Devpost␣submission-ID␣rank␣within␣session")
ax.legend(frameon=False, loc="upper␣left", fontsize=8)
ax.set_title("Schedule␣order␣loosely␣follows␣submission␣order", loc="left", fontsize=11)
fig.tight_layout(); fig.savefig(FIG / "fig_order.png", dpi=200); plt.close(fig)

# Forest plot
rob = [dict(subset="All␣sessions", **{k: res["primary"][k] for k in ("mean_winner_pos", "ci95", "expected"
    )})] + res["robustness"]
fig, ax = plt.subplots(figsize=(7, 3.8))
for i, t in enumerate(rob):
    ax.plot(t["ci95"], [i, i], color=BLUE, lw=2)
    ax.scatter([t["mean_winner_pos"]], [i], color=BLUE, s=40, zorder=3)
    ax.scatter([t["expected"]], [i], marker="|", color=ORANGE, s=120, zorder=4)
ax.axvline(.5, color=GRAY, ls="--", lw=1)
ax.set_yticks(range(len(rob)), [t["subset"] for t in rob]); ax.invert_yaxis()
ax.set_xlabel("Mean␣winner␣position␣(dot),␣95%␣CI␣(line),␣expectation␣under␣H$_0$␣(orange␣tick)")
ax.set_xlim(.25, .75)
fig.tight_layout(); fig.savefig(FIG / "fig_robust.png", dpi=200); plt.close(fig)

# Session atlas: 30 small multiples, 6 per page
keys = sorted(g)
for page in range(0, len(keys), 6):
    fig, axs = plt.subplots(3, 2, figsize=(7.5, 8.5))
    for ax, k in zip(axs.flat, keys[page:page + 6]):
        lst = sorted(g[k], key=lambda r: r["slot"])
        n = len(lst)
        ax.bar(range(1, n + 1), [1] * n, color=[BLUE if r["winner"] else "#e6e5e0" for r in lst], width=.8)
        ax.set_yticks([]); ax.set_xlim(.3, n + .7); ax.set_title(f"{k}␣␣(n␣=␣{n})", fontsize=9, loc="left")
        ax.set_xticks([1, n]); ax.spines["left"].set_visible(False)
        for r in lst:
            if r["winner"]:
                ax.text(r["slot"], 1.04, str(r["slot"]), ha="center", fontsize=7, color=INK)
        ax.set_ylim(0, 1.25)
    for ax in axs.flat[len(keys[page:page + 6]):]:
        ax.axis("off")
    fig.tight_layout(); fig.savefig(FIG / f"atlas_{page␣//␣6␣+␣1}.png", dpi=170); plt.close(fig)
n_atlas = (len(keys) + 5) // 6

# ---------- generated LaTeX ----------
def tex(s):
    s = s.replace("\\", "/")
    s = "".join(ch if ord(ch) < 0x250 else "?" for ch in s)
    for a, b in (("&", r"\&"), ("%", r"\%"), ("_", r"\_"), ("#", r"\#"), ("$", r"\$"), ("^", r"\^{}"), ("~"
        , r"\~{}"), ("{", r"\{"), ("}", r"\}")):
        s = s.replace(a, b)
    return s.strip() or r"\emph{(unnamed)}"

out = [r"\section{Complete␣Pitch␣Schedules}\label{app:sessions}",
       r"This␣appendix␣reproduces,␣in␣full,␣every␣analysed␣pitch␣slot.␣Winners␣are␣set␣in␣bold␣and␣marked␣$
           \star$;␣confirmed␣first␣places␣are␣additionally␣marked␣$\dagger$.␣Times␣are␣as␣scheduled.␣The␣
           reader␣is␣invited␣to␣look␣for␣a␣pattern;␣the␣authors␣did,␣at␣length,␣and␣found␣none.\footnote{
           Non-Latin␣characters␣in␣team␣names␣have␣been␣replaced␣by␣'?'␣for␣typographical␣reasons␣and␣with
           ␣apologies.}", ""]
for k in keys:
```

```
    lst = sorted(g[k], key=lambda r: r["slot"])
    out.append(r"\subsection*{" + tex(k) + f"␣({len(lst)}␣pitches)" + "}")
    out.append(r"\begin{small}\begin{longtable}{rlp{0.52\textwidth}rc}\toprule␣Slot␣&␣Time␣&␣Team␣(as␣
        scheduled)␣&␣Pos.␣&␣Won␣\\␣\midrule\endhead")
    for r in lst:
        tm = f'{int(r["time_min"])␣//␣60}:{int(r["time_min"])␣%␣60:02d}' if r["time_min"] else "--"
        mark = ("$\\star$" + ("$\\dagger$" if r["place"] == "1" else "")) if r["winner"] else ""
        name = tex(r["team"]); name = r"\textbf{" + name + "}" if r["winner"] else name
        pb = r"␣\emph{(after␣break)}" if r["post_break"] == "1" else ""
        out.append(f'{r["slot"]}␣&␣{tm}␣&␣{name}{pb}␣&␣{r["pos01"]:.2f}␣&␣{mark}␣\\\\')
    out.append(r"\bottomrule\end{longtable}\end{small}")
(GEN / "app_sessions.tex").write_text("\n".join(out))

atlas = [r"\section{Session␣Atlas}\label{app:atlas}",
         r"For␣readers␣who␣prefer␣bars␣to␣tables,␣each␣panel␣shows␣one␣session;␣every␣bar␣is␣a␣pitch␣slot,␣
             winners␣are␣blue␣and␣labelled␣with␣their␣slot␣number."]
for i in range(1, n_atlas + 1):
    atlas.append(r"\begin{figure}[p]\centering\includegraphics[width=\textwidth]{figures/atlas_%d.png}\
        caption{Session␣atlas,␣part␣%d␣of␣%d.}\end{figure}\clearpage" % (i, i, n_atlas))
(GEN / "app_atlas.tex").write_text("\n".join(atlas))

# Null variance table per session (exact formula)
nv = [r"\section{Exact␣Null␣Moments␣by␣Session}\label{app:moments}",
      r"For␣a␣session␣with␣$n$␣slots␣and␣$k=2$␣winners␣drawn␣without␣replacement,␣$\mathrm{E}[\sum␣x]=k/2$␣
          and␣$\mathrm{Var}[\sum␣x]=k\,\sigma_n^2\,(n-k)/(n-1)$␣with␣$\sigma_n^2=(n+1)/(12(n-1))$.␣The␣
          table␣lists␣these␣quantities,␣the␣observed␣sum␣of␣winner␣positions,␣and␣its␣standardized␣
          deviation␣$z$.",
      r"\begin{small}\begin{longtable}{lrrrrr}\toprule␣Session␣&␣$n$␣&␣$\mathrm{E}$␣&␣SD␣&␣Observed␣&␣$z$␣
          \\␣\midrule\endhead"]
tot_var = 0
for k in keys:
    lst = g[k]; n = len(lst); kk = sum(r["winner"] for r in lst)
    s2 = (n + 1) / (12 * (n - 1)); var = kk * s2 * (n - kk) / (n - 1); tot_var += var
    obs = sum(r["pos01"] for r in lst if r["winner"])
    nv.append(f"{tex(k)}␣&␣{n}␣&␣{kk/2:.2f}␣&␣{var**.5:.3f}␣&␣{obs:.3f}␣&␣{(obs-kk/2)/var**.5:+.2f}␣\\\\")
nv.append(r"\midrule␣\multicolumn{6}{l}{Implied␣SD␣of␣$T$␣(all␣winners):␣" + f"{tot_var**.5/sum(r["winner"
    ]␣for␣r␣in␣rows):.4f}" + r";␣Monte␣Carlo␣estimate:␣" + f"{res['primary']['null_sd']:.4f}" + r".}␣\\␣\
    bottomrule\end{longtable}\end{small}")
(GEN / "app_moments.tex").write_text("\n".join(nv))
print("exact␣null␣SD", tot_var ** .5 / sum(r["winner"] for r in rows), "KS", ks)
```