Report

Jaume Boned

2025-11-13

Maze Game Data Analysis

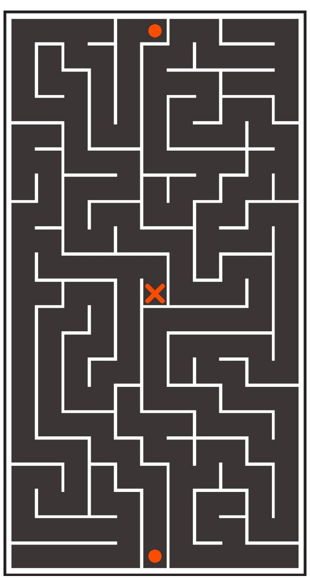
1. Introduction

1.1 Explanation of the Game

- Brief description of the mechanics (maze generation, objective, scoring).

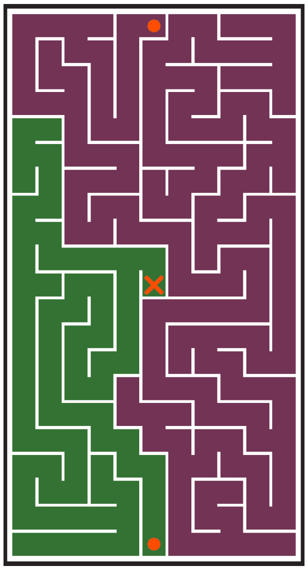
aMaze is a mobile puzzle game where players must navigate procedurally generated mazes under time pressure to earn the greatest score possible. Each maze in the game is unique, ensuring replayability, unpredictability and a challenge to any player in a short time game.

- *Maze:* When starting a session, the player enters a **block** of 10 **mazes**. In each maze, players are presented with a new randomly generated grid of 11 cells width and 21 cells height. At the central cell of the maze there is always an orange X that marks the treasure of the maze, as well as two orange dots that mark the two entrances centered at the top and bottom rows.

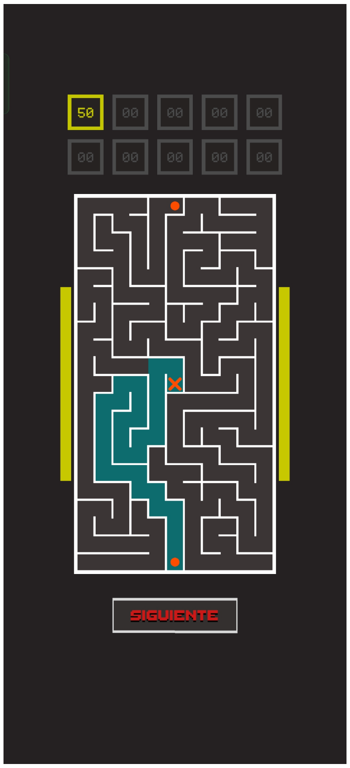


- *Algorithms:* Each maze is generated with two simultaneous backtracking algorithms starting at each entrance cell. This generates to fully sepparated paths, with only one connected with the central cell.

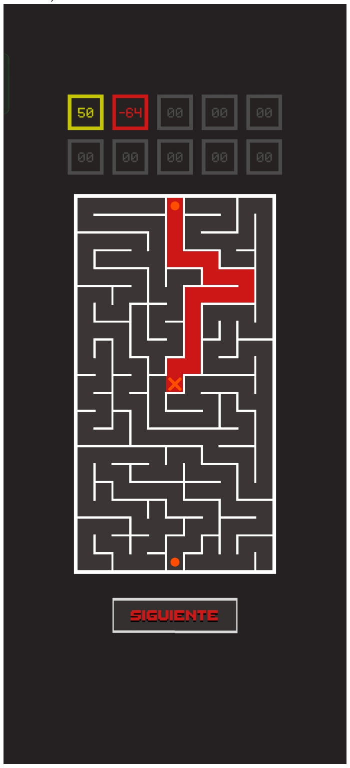
  See an **example** with the top algorithm painted in purple and the bottom algorithm painted in green:



- *Task:* In each maze, the player must find out which entrance is connected to the central cell as fast as possible. The procedure of each maze is as follows:
  1. The maze is revealed and a 10 seconds countdown starts. During each maze, the player receives continuous feedback. Two shrinking bars on the sides of the screen visually indicate the remaining time. At the top of the screen, small squares summarize the scores obtained in previous mazes within the block. The square corresponding to the current maze displays a live countdown of the exact points available, decreasing second by second. This dual feedback system allows players to monitor their progress both in the moment and across the block.
  2. The player should tap the upper half of the maze if the central cell connects with the top entrance, or the lower half if it connects to the bottom entrance.
  3. Whether the player gives a response or the 10 second timer is over, the maze ends and the score is registered at the corresponding score squares located at the top of the screen.
     The scores are determined as follows:
     * **Correct answer**: They gain an amount of points equal to the percentage of remaining time (from 100 to 0).



* **Wrong answer**: Negative points equal to the percentage of remaining time (from -100 to 0).



* **Time up:** Alternatively, if participants didn't give any response when the time is over, the score will become automatically -100.

4. Each score is stored at the top of the mazes, and after 10 mazes are finished the block ends

and participants get the final score as the average score of all 10 mazes (with a minimum of 0).

This method of scoring encourages players to always try to give a response, but spend as much time as needed to have enough certainty that the side they select will be the correct one, balancing between the **probability** of reward (as the information gathered through time) and the **amount** of reward. The faster they respond, the greater will be the reward if they succeed, but also it will be more risky. This tension between speed and accuracy, added to the fast nature of the game (as each block usually takes between 30 seconds and 1.5 minutes to complete) makes the game perfect for short but engaging play sessions.

Finally, with this score, players may find a personal goal both individually by improving one's own scores across sessions and/or competitively, through leaderboards that rank players globaly.

1.2 Aim of the Analysis

The aim of this report is to provide insight into how players interact with the game, identify any aspects that influence their success, and allow for predictions about how well they will perform and for how long will they play.

Therefore, we focus on player behavior and a bit of UX insight.

1.3 Research Questions / Hypotheses

- List of main questions (learning, difficulty, churn, strategies, etc.).
- Hypotheses if any (e.g., "Players will improve over blocks", "More difficult mazes increase error rate").

---

2. Methodology and Data

2.1 Data Collection

The game creates 2 different files locally, which are updated at the end of each maze:

- **datapack.json** contains all the information of each cell, of each maze, of each block, along with participants' information. This is the data that we will analyze.
- **summary.json** contains only the information of each participants' block, without the specifics of the maze and cell data. This alternative file is used to more easily keep progression of each participants with only local files.

2.2 Main Variables

- Player-level variables:
  - **PlayerID:** Nickname that each participant chose personally.
  - **DeviceID:** Random 4-character identifier of each individual device.
- Maze-level variables:
  - **TrialID:** Identifier of the maze order within a block (from 1 to 10).
  - **TrialTimeStamp:** Time stamp of each individual maze.
  - **Score:** Each maze score goes from -100 to 100. It starts at 100 as the maze starts and decreases towards 0 across the 10 seconds players have to answer. If the response is correct the value is stored as positive and if the response is wrong the value becomes negative. If players do not respond before the 10 seconds, the score becomes automatically -100.
  - **RT:** The time in seconds players took to give a response. If a player does not respond, the RT is stored as NA.
  - **Correct:** Whether participants selected the correct side (TRUE) or the wrong side (FALSE).
  - **CorrectSide:** Which is the side of the correct path.
  - **PathLength:** How many steps is the path long.
- Block-level variables:
  - **BlockID:** Identifier of the number of blocks that the player performed (increasing from 1).
  - **BlockTimeStamp:** Time stamp of the start of the block of mazes.
  - **AvgScore:** Average of the scores of all completed mazes within a block.
  - **IsBlockComplete:** Whether the block reached the end (TRUE) or the player closed before finishing (FALSE).

2.3 Data Cleaning

- Handling missing data (incomplete blocks, NA values).
- ~~How identifiers were handled (deviceID vs playerID).~~
- Basic preprocessing steps. *

---

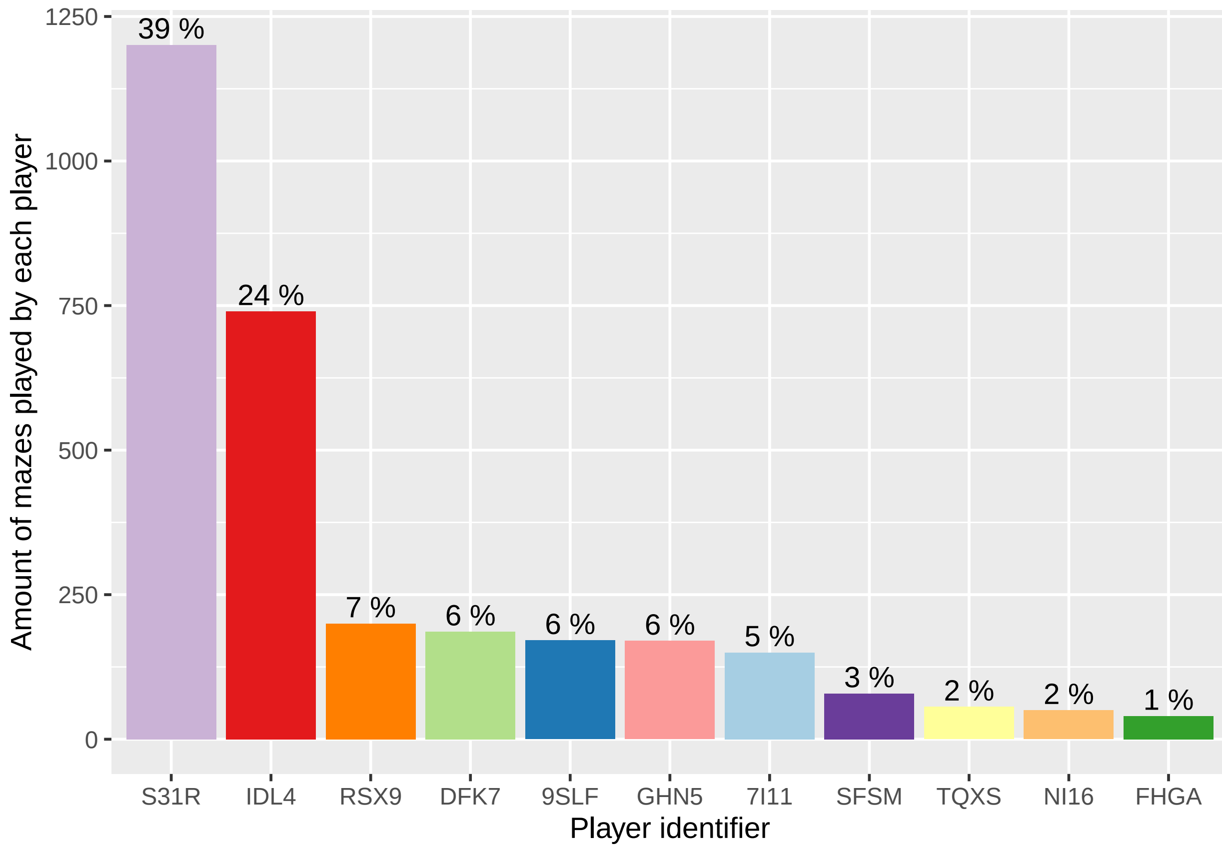
3. Analysis and Results

3.1 Summary Data

- **How many players played the game?**

We gathered data from a total of **3043 mazes** across **11 players** during an interval of **24** days.

- **How much did they play?**
  - We observe an uneven amount of mazes played between participants, with a few participants owning the majority of the games played.



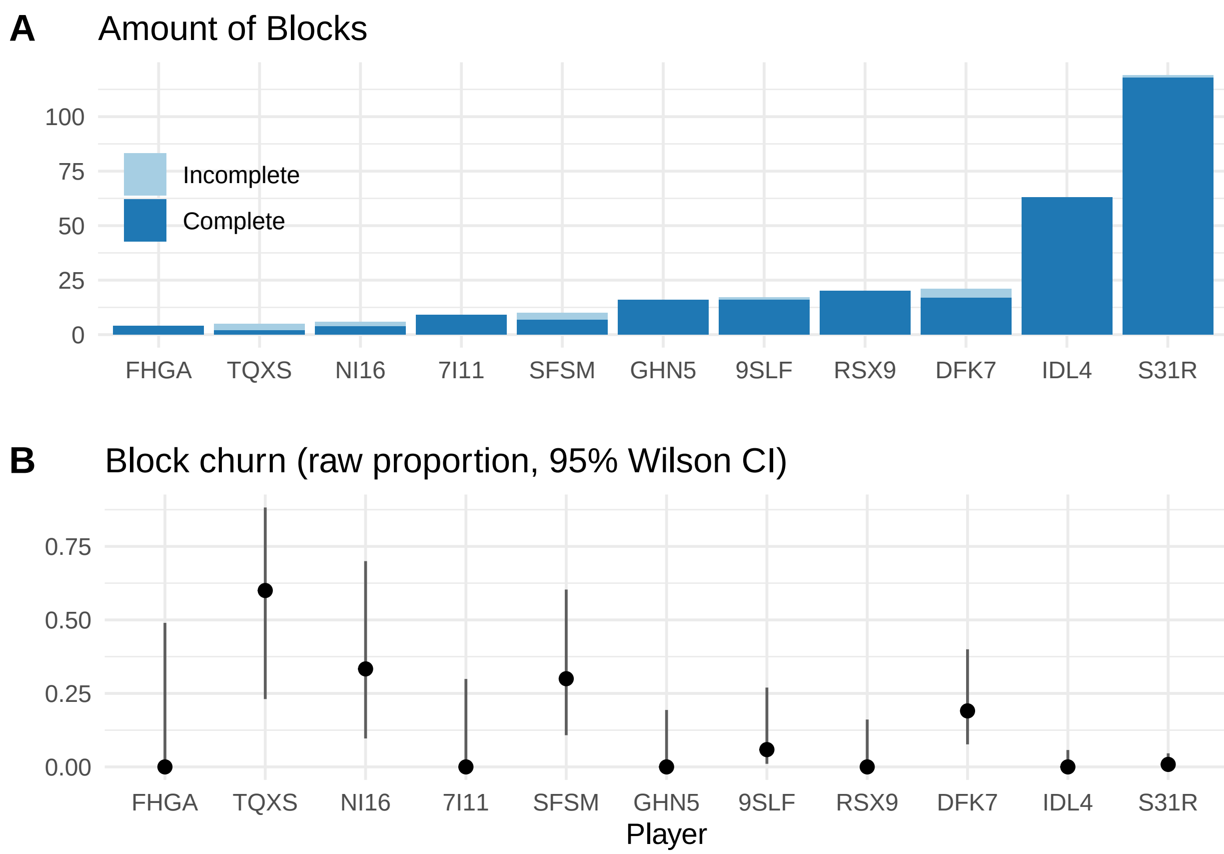
3.2 Engagement

A key aspect of the experience is not only *how much* people play, but *where/when* do they stop. We look at disengagement through two lenses. First, **block churn**: blocks of mazes that were started but not completed. Second, **session churn**: players who finish a block and then stop playing for a while (end of a session).

3.2.1 Incomplete blocks (Block churn) One example of this situation is when players are performing a block of mazes but decide to stop playing midway through the block.

We quantify *block churn* as the fraction of started blocks that ended incomplete. In our dataset, incomplete blocks are rare (~*5%*) and unevenly distributed across players.

Panel A displays each player's *volume* of complete vs incomplete blocks. To correct for the lack of a greater sample, Panel B shows each player's *incomplete rate* by a shrunk estimate (that gently pulls very small samples towards the overall average, leaving well-measured players essentially unchanged) with 95% Wilson confidence intervals (which is more robust for small samples).



From this, we can conclude that more hardcore players (those that played more games overall) almost never leave a block unfinished, while higher incomplete rates are observed mostly on more casual players (with less mazes played overall) and come with wider uncertainty. However, although block churn exists in our sample, it doesn't seem to be the dominant disengagement pattern.

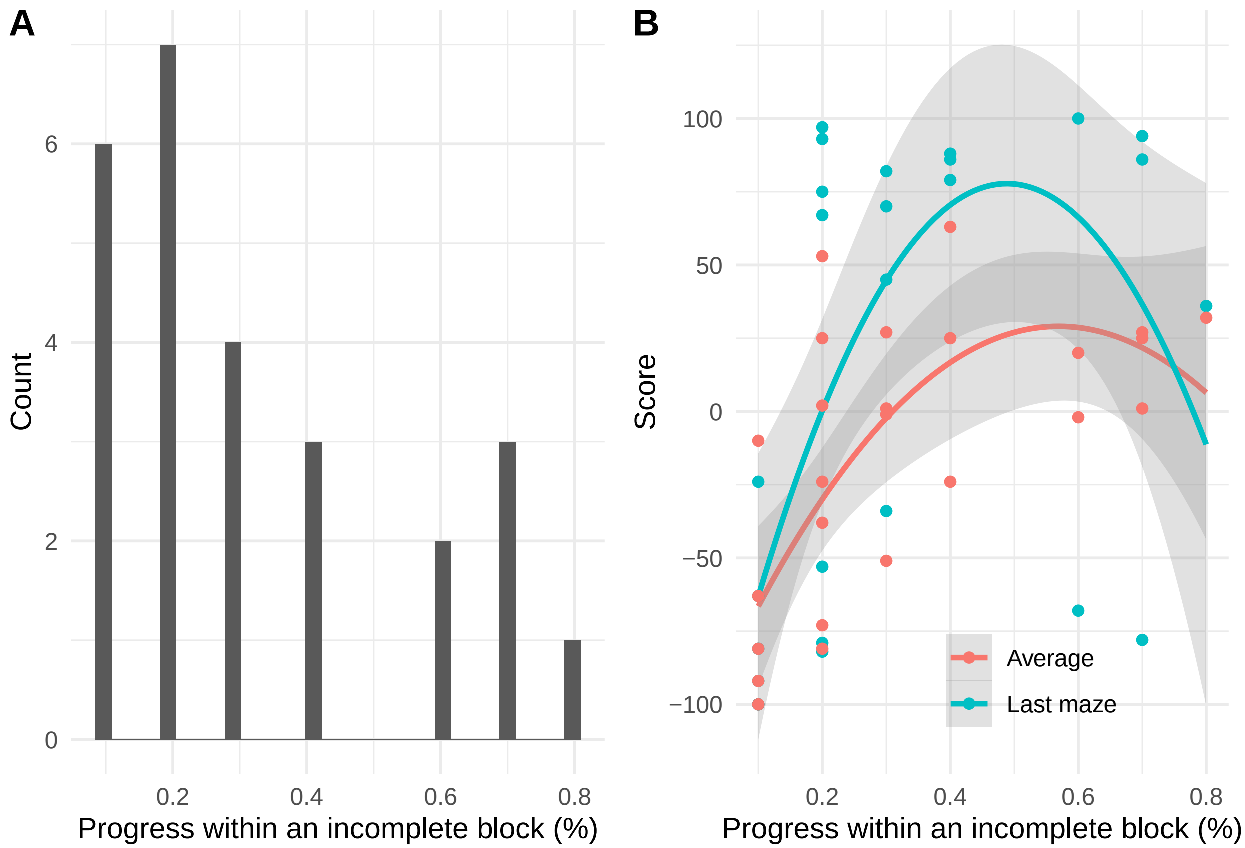
How far into the block do players quit?

This block churn could happen at different moments, for example a player might start with very low scores in the first couple of mazes and decide it not worth to carry on with the block, or they could try to get to the end and maybe predict a low final score and decide to quit from frustration.

Given the small number of incomplete blocks (N=*26*) we can only report descriptive evidence.

A bimodal distribution of churn can be identified (Panel A), one with few mazes (1 or 2) and a smaller one closer to the end of the block, at 7 trials. No blocks missing only one trial are found.

In terms of its relation to the score of the mazes played (Panel B), we find that churn in early mazes is always associated to negative score, which represents blocks that start with wrong responses. With churn later in the block, we find that although participants got a very good score in the last maze, the average score was still low, which implies that early mazes had to have a negative score (hence a wrong answer). This shows that the most important mazes to predict whether participants will churn are the first ones of the block.



3.2.2 Spacing between blocks (Sessions churn) Another conceptualization of disengagement could be the moment at which players stop playing successive games and leave the game.

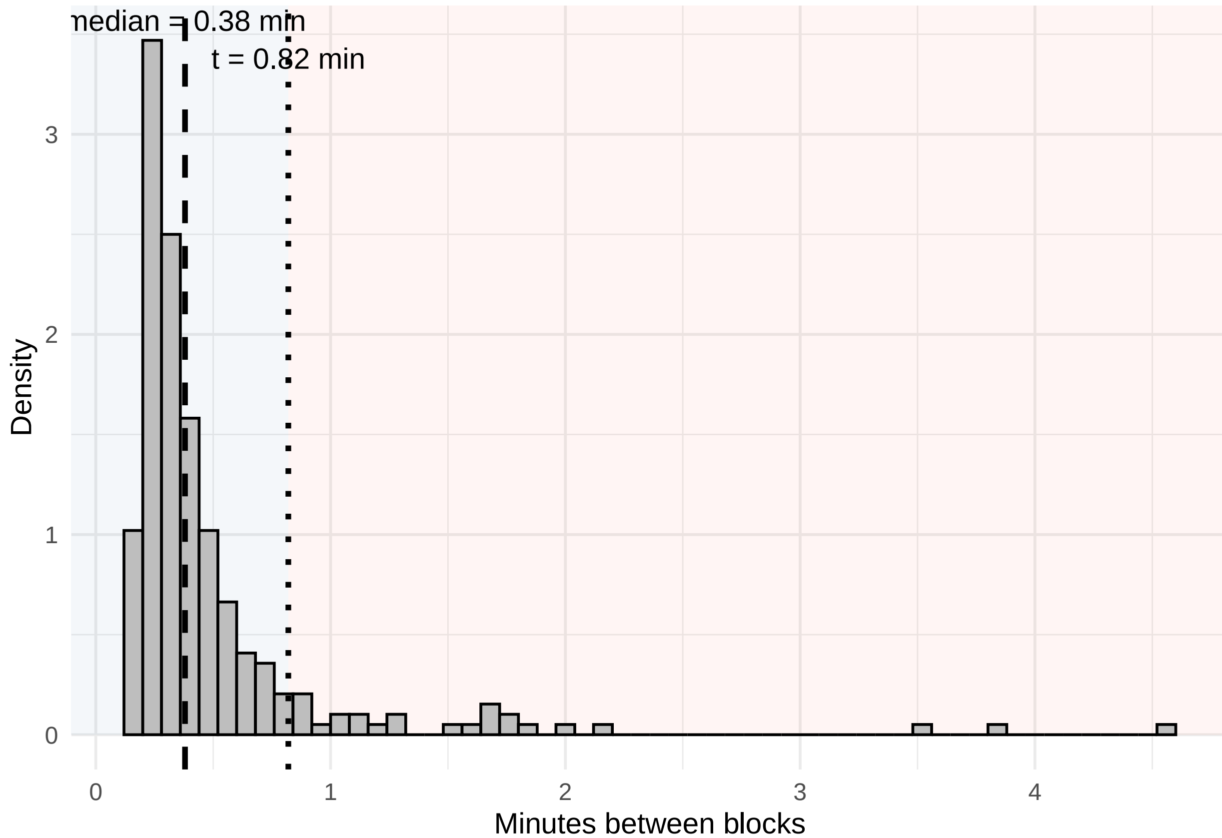
Here, the criteria is the gap between blocks. However, interpreting the time between blocks can be tricky. Players could be having a short rest between blocks of the same "gaming session", or they could completely separate their attention from the game, coming back at a later (although sometimes short) time to begin with a new session.

The key to define when a session has really ended relies on finding an adequate time threshold, after which we expect (at least in most cases) that the following block will be part of a new session. Once we identify this we can pinpoint the games where, although completing the block, led to players leave the game afterwards. This in turn can allow to study what factors can influence the disengagement to the game.

We compute inter-block gaps from the **end time** of a block (last maze start + RT) to the **start time** of the next block.

To differentiate between short within-sessions rests from session ends we use a robust, data-driven threshold (t) based on the Median and the Median Absolute Deviation. Gaps below the threshold keep the session as continued, gaps above the threshold are considered the division between two sessions. With this, we can pinpoint the moments at which players stopped playing.

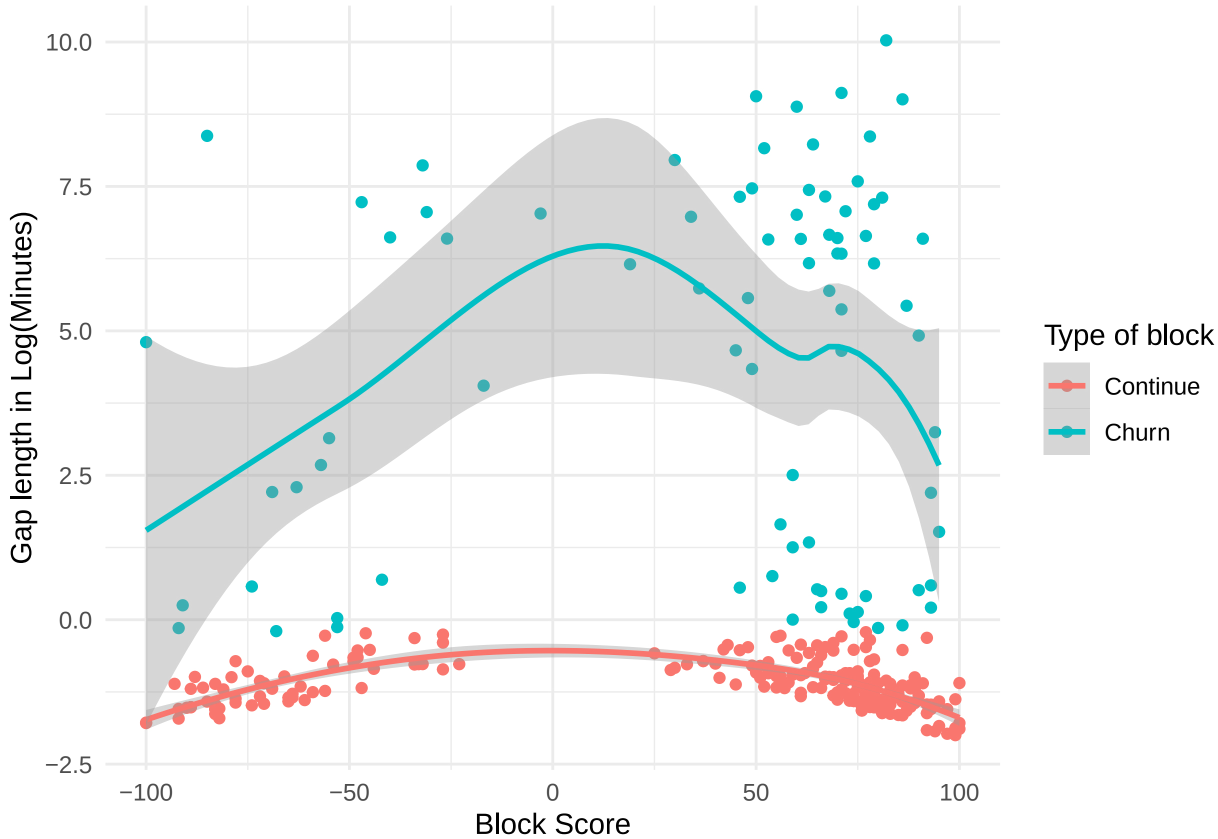
Here we show the distribution of the shortest gaps between blocks (longer gaps, such as hours or days are not even considered for representing the threshold). The vertical dashed line marks the **median** (0.38 s), and the dotted line marks the **threshold** (0.82 s).



Due to the huge scale difference in gaps between blocks, where most of them lie below 3 minutes but there are some observations to up to days, we applied a logarithmic transformation to the gaps length in minutes.

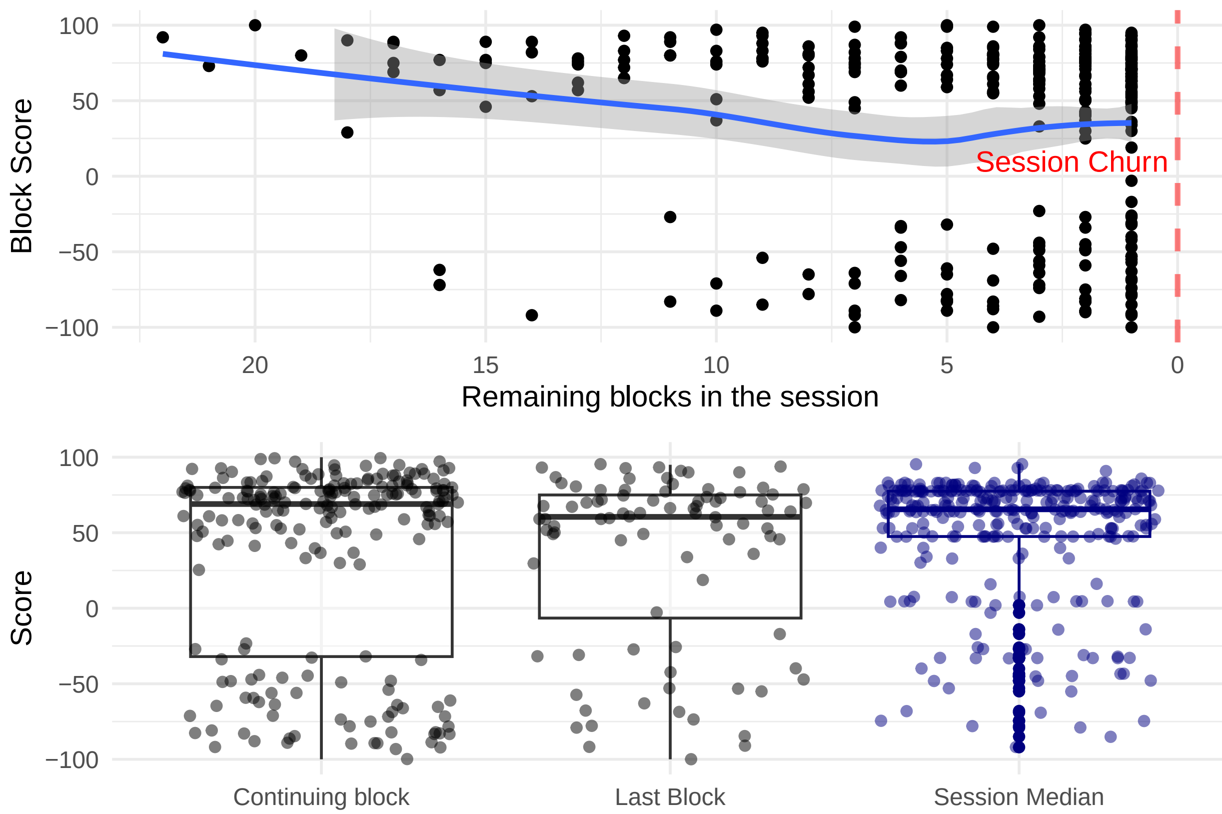
We find that within-session rests (gaps after blocks that do not end the session), show a weak inverted-U relationship with score. Pauses are shortest after a very low and very high scores, and slightly longer around mid-range scores. One possible explanation for this finding is that after a big success players are reinforced to keep playing as quickly as possible, while a very bad score drives them to improve that last game and surpass their own score. In both cases, players **engage** more with the game because of the **extreme scores**. In terms of session ends (by definition longer gaps), they follow a similar pattern, but with gaps more evenly distributed after higher scores.

Note that given the sample size and the difficulty at establishing a reliable threshold, these patterns are descriptive rather than inferential.

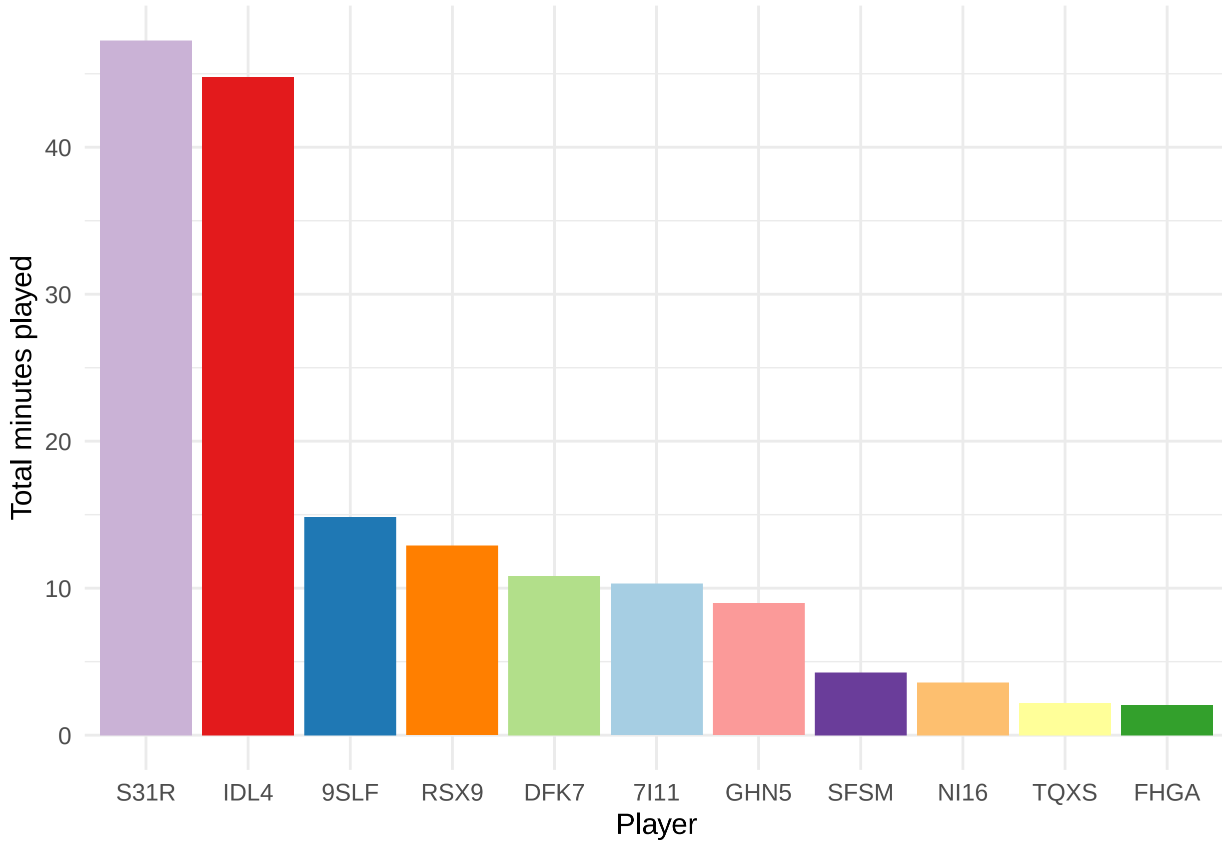


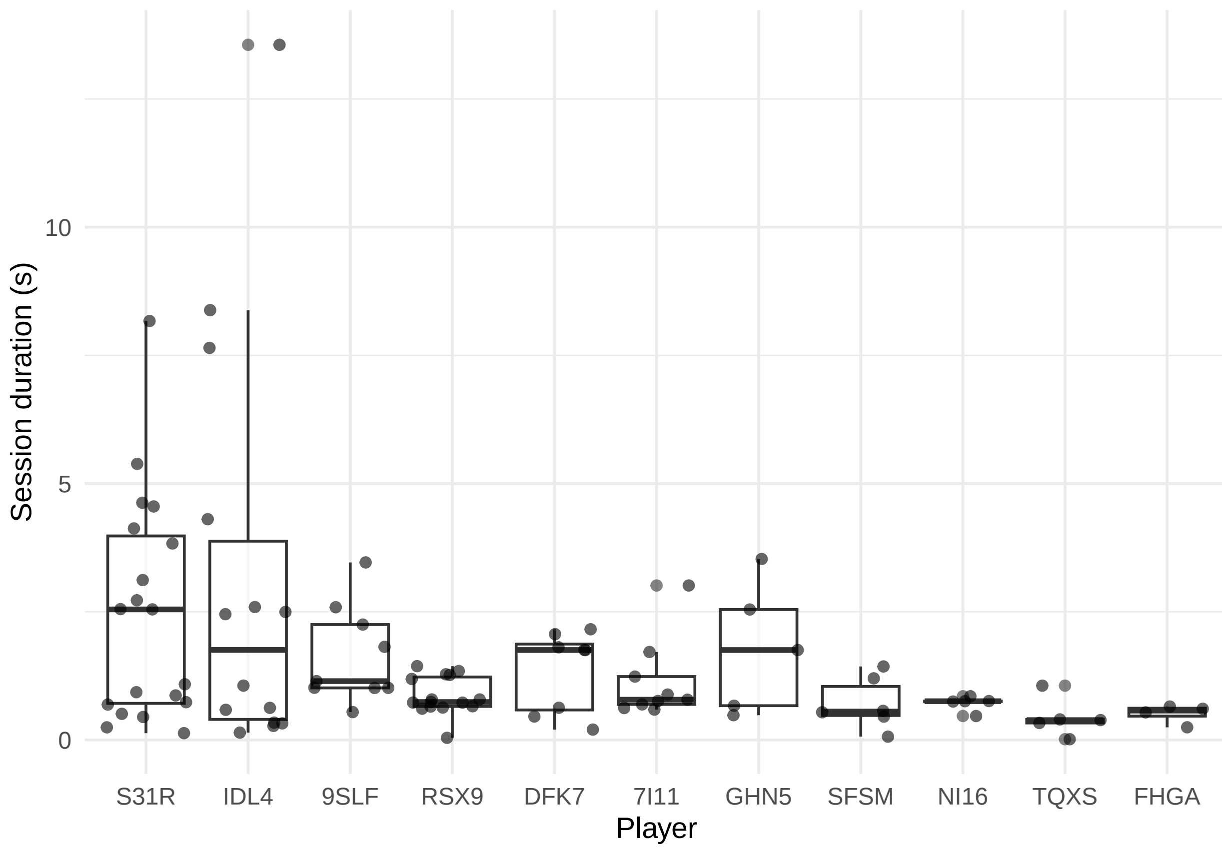
We also find that early blocks in a session tend to show better performance, that decays with later blocks presenting more varied scores, with increased prevalence of negative values. This shows that players might be getting tired after repeated blocks, and once their performance has worsened enough, they leave the game until a future session. Although our data is not sufficient, with enough amount of data this could be used to get a tailored prediction of when each player might abandon the game according to their evolution in performance.

Also, although there are not significant differences between the last block and other blocks or even the session median, it is clear that the variability in scores is greater at these last blocks. Probably tiredness makes these games less representative of the real player level.

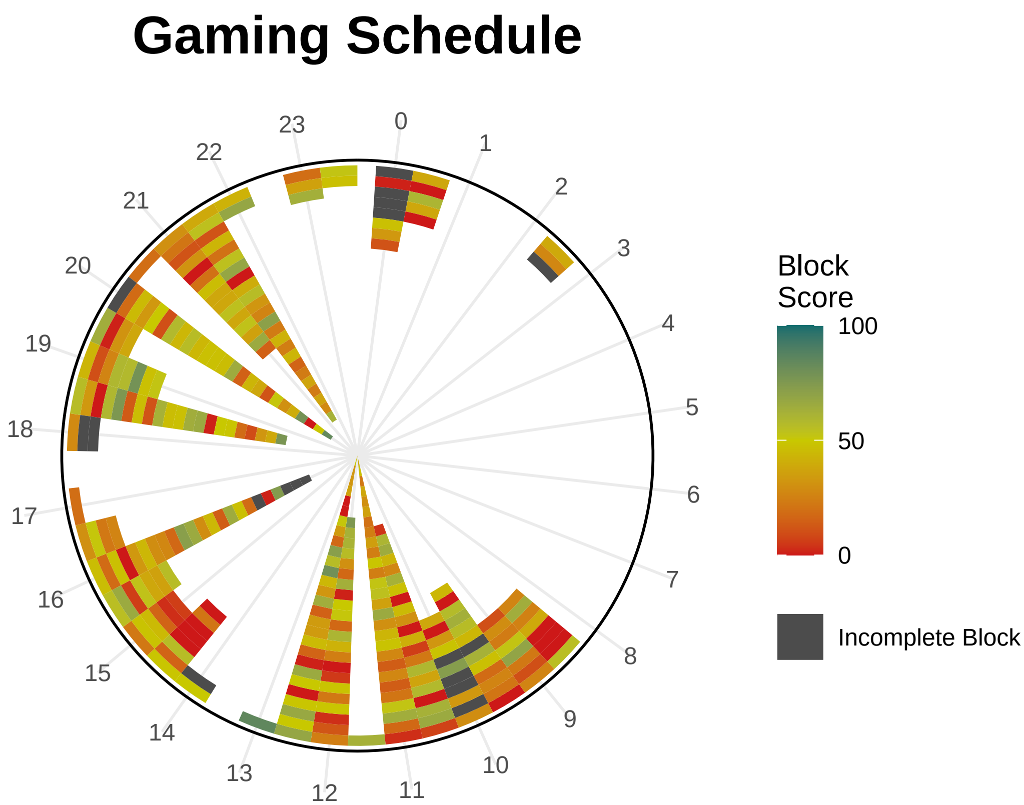


3.2.3 Time played Total time varies widely across players. Pairing minutes with sessions and average session length reveals distinct engagement patterns (few long sessions vs. many short ones).


Total minutes played
0
10
20
30
40
S31R
IDL4
9SLF
RSX9
DFK7
7I11
GHN5
SFSM
NI16
TQXS
FHGA
Player



Session duration (s)
10
5
0
S31R
IDL4
9SLF
RSX9
DFK7
7I11
GHN5
SFSM
NI16
TQXS
FHGA
Player




```r
knitr::knit_exit()
```