Player Retention

Mobile game data over 365 days

Project Introduction

- The project data was provided by a mobile game company.
- We worked with four tables spanning 365 days of user data including registration numbers, player info such as location, in game purchases and frequency of game play.
- First we were asked to create a table counting the rolling retention rates and fractional retention, based on players who played a game within 30 days of joining.
- As a further investigation we looked at retention specific to the location of players as well as the average in game spend per player, specific to their location.

Approach

- To create the first table we combined two sets of data in a relational database.
- We created multiple SQL subqueries to aggregate the two data sets, analyzing the player retention, and fractional retention, looking at these grouped by day the player joined.
- We used the structure of the first queries to create the second table, analyzing the retention and average in-game spends per player, grouped by the player location.
- Once we had created the two tables we moved the files to Google Sheets to analyze the data and create visualizations.
- Note: The last 30 days of data provided regarding player retention was incomplete, as we were not able to determine if those who joined played a game within 30 days. For this reason we have omitted days 335 - 365.

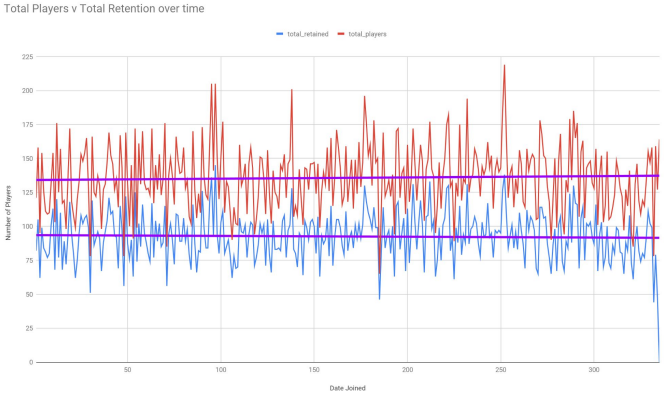
Queries

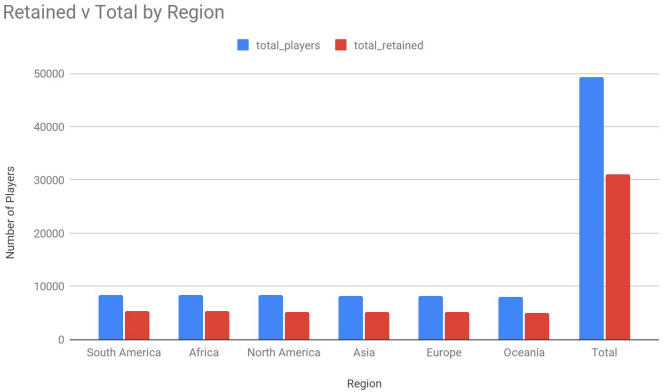
```sql
--Create a table with date joined, total number of players joined per day, total players retained
and fractional retention
SELECT
  date_joined,-- day in question
  total_players, -- total players joined per day
  total_retained, --total players retained from those joined
  ROUND((total_retained / total_players*100), 2) AS percentage --Fractional retention

FROM (
  SELECT
    date_joined,
    COUNT(player ) AS total_players,--counting total players by date joined
    SUM (retained) AS total_retained --Suming the players who returned a 1 in the case state

  FROM (
    SELECT
      joined AS date_joined,
      player_id AS player,
      CASE
        WHEN last_game - joined >30 THEN 1 --finding players who were retained, played a gam
within 30 days of joining
      ELSE
      0 --will return 0 for players outside 30 days
    END
      AS retained
    FROM (
      SELECT
        joined, -- date players joined 1-365
        p.player_id, --unique player id
        MAX(day) AS last_game --Because some players played multiple games, will return the last
game they played
      FROM
        `juno2110.gamecodata.matches_info` AS m -- contains match info, needed to find retention
      JOIN
        `juno2110.gamecodata.player_info` AS p --contains player info including join date
      ON
        m.player_id = p.player_id
      GROUP BY
        p.player_id, joined)) --group by unique player id to remove duplicate rows and joined day
to get closer to retention numbers

  GROUP BY
    date_joined) -- group by date joined to return the total number of players and retention per
day
```

Visualizations

- Trend finds retention to be relatively stable over time

- When you compare across regions, player retention remains relatively the same.





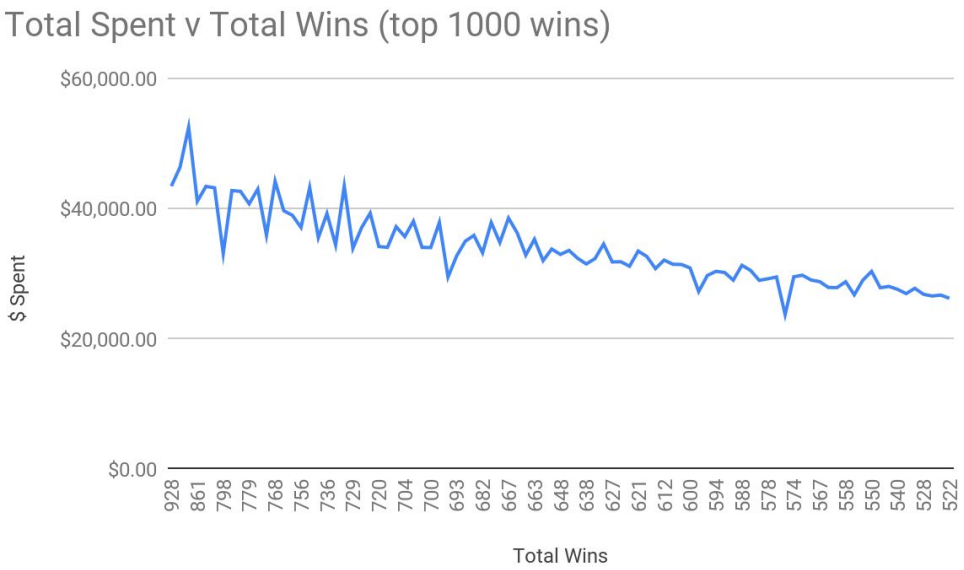
Query

Query used for Top 1000 players and how much they spent.

```sql
/*Grab the top 1000 players with wins and see how much they spend on items*/
SELECT DISTINCT
count(*) as total_wins, m.player_id as player, SUM(item.price) as cost
FROM
  `pro-tracker-329514.gamecompanydata.matches_info` as m
  JOIN
  `pro-tracker-329514.gamecompanydata.player_info` as p
  ON
    m.player_id = p.player_id
  JOIN
  `pro-tracker-329514.gamecompanydata.purchase_info` as pID
  ON
  p.player_id = pID.player_id
  JOIN
  `pro-tracker-329514.gamecompanydata.item_info` as item
  ON
  pID.item_id = item.item_id
    WHERE m.outcome = "win"
    GROUP BY m.player_id
    ORDER BY total_wins DESC
Limit 1000
```

Visualizations

- Players with more wins spend more, and this tracks across the top 1000 winners



Findings & Conclusion