Hendon Mob Analysis

Brendon Kaufman

2/5/2019

Hendon Mob Database Analysis

Background: The game of poker is arguably America's most popular card game. Based on the variant, players hold various numbers of cards, trying to make the strongest 5-card combination. The strengths of these combinations are determined by a ranking system which is uniform across all variants. Poker's biggest attractions are its unique mix of luck and skill and the fact that oit's one of few games where players regularly bet money as part of the game. These factors draw a wide variety of players, from professionals who make their living solely from playing the game, to recreational players who enjoy a challenge or a gamble. Although there are clear differences between these populations, there has yet to be rigorous analysis using existing databases which would demonstrate how these differences manifest themselves.

Goal of this analysis: I propose an analysis of poker's only public database of player performance, The Hendon Mob tournament database, to reveal insights about different populations playing tournament poker.

Data : The data analyzed come from The Hendon Mob, a tournament poker database which displays all live cashes for individual players. If a player made money in an official tournament, it is recorded on www.thehendonmob.com. I used the package `rvest` to scrape data from individual player pages. Then, I created functions to extract and summarize the most important statistics for each player and create a summary dataframe, where each row contains one player and his defining statistics and information. This is done with the functions in the script *01_scrape_hendon_mob* located in my Github. The script *02_analyze_hendon_mob* allows the user to convert the summary dataframe into a format suitable for analysis, and provides some sample analyses.

———Loading in packages and the data————

```
library(tidyverse)
library(lubridate)

hendon_summaries_df <- read_csv("hendon_summaries.csv")

hendon_summaries_df <- hendon_summaries_df %>%
  mutate(new_name = case_when(is.na(name) == TRUE ~ nationality, is.na(name) == FALSE ~ name)) %>%
  mutate(new_nationality = case_when(is.na(name) == TRUE ~ "Unlisted", is.na(name) == FALSE ~ nationali
  mutate(average_cash = round(sum_of_cashes/number_of_cashes, 2)) %>%
  mutate(average_placement = round(average_placement), 2) %>%
  mutate(average_buy_in = round(average_buy_in), 2) %>%
  mutate(last_date = as_date(last_date)) %>%
  mutate(first_date = as_date(first_date)) %>%
  mutate(years_played = round((last_date - first_date)/365.25,2)) %>%
  mutate(average_time_btwn_cash = round((last_date - first_date)/number_of_cashes), 2) %>%
  mutate(average_time_btwn_cash = as.numeric(average_time_btwn_cash)) %>%
  mutate(binks_proportion = round(number_of_binks/number_of_cashes * 100, 2)) %>%
  select(name = new_name, nationality = new_nationality, average_buy_in,
         number_of_cashes, sum_of_cashes, average_cash, average_placement, number_of_binks,
         binks_proportion, number_of_countries_cashed, first_date, last_date, years_played,
```

```r
         average_time_btwn_cash, unique_views) %>%
  arrange(desc(sum_of_cashes)) %>%
  mutate(quantile = ntile(sum_of_cashes, 4))
```