Andys MLB Download

Author - Andy Ouellette willandy14@gmail.com (mailto:willandy14@gmail.com)

6/19/2024

READ ME

How to execute

Run the lines of code to install the packages and execute the script on 00_Download_MLB_Master.R

Sections

1. Todays MLB lines and prices for h2h, spreads and totals
2. Historical lines and prices for h2h, spreads and totals
3. Historical game scores
4. Combines #2 and #3 into a final dataframe

Dataframes_% of relevance

- sportsbook_%
- historical_%
- final_%

Information

- This is an R script that can be used to download historical MLB game scores, closing lines and prices.
- The main data sources are the baseballR package and the-odds-API.
- The script can be run with a free API key obtained through the-odds-API website. If you plan to use the script more frequently , you will have to get a paid API license.
- The script is setup to run from the start of the 2024 MLB season until yesterday for all downloads.
- The script downloads todays betting lines and prices for h2h aka moneyline, spreads and totals
- The script is using the default sportsbooks of DraftKings and Pinnacle

- The prices are downloaded as American and converted to Decimal for convenience and further calculations
- The script downloads historical prices that reflect the closing line, can be modified to download for certain time intervals before game time.
- The historical game scores returns game rows with home and away team scores, winning percentages and other game identifiers.
- The vigorish % is calculated for Pinnacle on h2h, spreads and totals is saved into variables avg_vig_%
- The final dataframe final_combined_df_wide should have 3 rows for each game, one for h2h spreads and totals with the respective line information.

The total vig is calculated using the formula:

$$\text{vig} = \left(\frac{1}{\text{decimal\_outcome\_1}} + \frac{1}{\text{decimal\_outcome\_2}} \right) - 1$$

Where: - decimal_outcome_1 is the decimal odds for the first outcome. - decimal_outcome_2 is the decimal odds for the second outcome.

The average vig for each market key is then calculated by grouping the data by `market_key` and taking the mean of the vig values within each group. # print(avg_vig_h2h) # 0.04159131 # print(avg_vig_spreads) # 0.04273564 # print(avg_vig_totals) # 0.04582104

Packages

```
# List of packages to check and install if necessary
packages <- c("baseballr", "dplyr", "data.table",  "httr2", "tidyverse", "lubridate",
              "jsonlite", "ggplot2", "GGally", "purrr", "tibble")
```

What to update

```
## Paste your API key here
api_key <- "e0d0cebccd1f3b630b762f64b521867d"
# API parameters
regions <- c('us', 'eu')  # Change if needed
books <- c('pinnacle', 'draftkings')  # Change if needed
```

```r
# All of the sections write the dataframes to files.
# Update the variables to your own folder paths or document names

# Generate date and time strings
today_date <- format(Sys.Date(), "%Y-%m-%d")  # Formats today's date as 'YYYY-MM-DD'
current_time <- format(Sys.time(), "%H-%M-%S")  # Formats current time as 'HH-MM-SS'

# Specify folder path
folder_path <- "C:/Users/WINDOWSUSER/Documents/%"  # Update the path as needed

# Generate filenames for each data frame
filename_h2h <- paste0("historical_h2h_", today_date, "_", current_time, ".csv")
filename_spreads <- paste0("historical_spreads_", today_date, "_", current_time, ".csv")
filename_totals <- paste0("historical_totals_", today_date, "_", current_time, ".csv")

# Combine folder path and filenames
full_path_h2h <- file.path(folder_path, filename_h2h)
full_path_spreads <- file.path(folder_path, filename_spreads)
full_path_totals <- file.path(folder_path, filename_totals)
```