PRACTICE_baseball_data

Andrew Crena

2023-05-28

R Markdown

I prefer to set wd's through the user interface by either using the upper left hand quadrant, or the bottom right.

```
## [1] "C:/Users/adcre/OneDrive/Documents/Desktop_RStudio"
```

Baseball Data

Example of me importing a baseball data set, cleaning anything that needs it, and performing some exploratory analysis. Hopefully we will be able to use visualization, but my priority with this file is just to practice importing!

Import from loacal files, still zipped

```
unzip_baseball <- unzip("C:\\Users\\adcre\\OneDrive\\Documents\\Desktop_RStudio\\baseball_data_zipped.z
baseball_data <- read.csv(unzip_baseball)

View(baseball_data)
```

Import from URL

```
library(utils)

# assign a variable to your url for future ref
url <- "https://www.kaggle.com/datasets/mathchi/hitters-baseball-data/download?datasetVersionNumber=1"
download.file(url, destfile = "dataset.zip")
unzpd_baseball <- unzip("dataset.zip")
baseball_data_url <- read.csv("C:\\Users\\adcre\\OneDrive\\Documents\\Desktop_RStudio\\Hitters.csv")

View(baseball_data_url)
```

Clean/Subset

```
baseball_data <- read.csv("Hitters.csv")
Column_indexes <- colnames(baseball_data)
Column_indexes
```

Let's see how we can subset this data. But first, take a look at your variables !

```
##  [1] "AtBat"     "Hits"      "HmRun"     "Runs"      "RBI"       "Walks"
##  [7] "Years"     "CAtBat"    "CHits"     "CHmRun"    "CRuns"     "CRBI"
## [13] "CWalks"    "League"    "Division"  "PutOuts"   "Assists"   "Errors"
## [19] "Salary"    "NewLeague"
```

```
# It looks like we have to create a subject column, here is one way of doing this
baseball_data$subject_id <- row.names(baseball_data)
View(baseball_data)

# Let's subset players with over 200 at-bats
manyabs_data <- baseball_data[baseball_data$AtBat > 200, ]
View(manyabs_data)

# After looking through the data, there appears to be some missing values, especially in certain columns
noNA_baseball <- subset(baseball_data, !is.na(baseball_data$Salary))
View(noNA_baseball)

# Let's combine these two conditions for our next subset, just to show how specialized you can make your
new_baseball <- baseball_data[baseball_data$AtBat>250 & baseball_data$HmRun<5, ]
View(new_baseball)

# After observing this graph, I see there are columns for 'Hits' and 'AtBats', but no batting average c
baseball_data$batting_avg <- ((baseball_data$Hits)/(baseball_data$AtBat))
which(colnames(baseball_data) == "batting_avg")
```

```
## [1] 22
```

```
bat_avg <- baseball_data[ ,22]

# Removing outliers
# grep() is being used here to remove an outlier, assuming we know the characters/numbers etc of the
# data. For the # sake of example, I will remove a case where a hitter had not enough at-bats (19) for
# me to feel confident in their data.
badid <- grep('65', baseball_data$subject_id)
baseball_data_sub <- baseball_data[-badid,]
View(baseball_data_sub)
```

```
# download new data set. There are two csv files I will be looking at. One will be hitting stats from t
# postseason in the past half-century for all players that played in the playoffs, as well as another
# data set that contains pitching stats. Let's se what we can learn!
hitting_data <- read.csv("BattingPost.csv")
```

```
# This is clearly too much, so let's just extract the last three years, 2012-2015 (info only kept until
# 2015)
new_hitting <- hitting_data[hitting_data$yearID > 2012, ]
str(new_hitting)
```

Thankfully, baseball has become a very data-oriented game in the past decade. This allows us to have descriptive equations or statistics that can be easily visualized. Let's find a different baseball dat set online to practice on.

```
## 'data.frame':    1180 obs. of  22 variables:
##  $ yearID  : int  2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
##  $ round   : chr  "ALCS" "ALCS" "ALCS" "ALCS" ...
##  $ playerID: chr  "albural01" "avilaal01" "benoijo01" "berryqu01" ...
##  $ teamID  : chr  "DET" "DET" "DET" "BOS" ...
##  $ lgID    : chr  "AL" "AL" "AL" "AL" ...
##  $ G       : int  6 6 3 1 4 4 2 6 3 4 ...
##  $ AB      : int  0 16 0 0 6 0 0 22 5 0 ...
##  $ R       : int  0 2 0 0 4 0 0 3 0 0 ...
##  $ H       : int  0 3 0 0 3 0 0 6 0 0 ...
##  $ X2B     : int  0 0 0 0 3 0 0 0 0 0 ...
##  $ X3B     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ HR      : int  0 1 0 0 0 0 0 1 0 0 ...
##  $ RBI     : int  0 3 0 0 0 0 0 4 0 0 ...
##  $ SB      : int  0 0 0 1 0 0 0 1 0 0 ...
##  $ CS      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ BB      : int  0 4 0 0 3 0 0 2 0 0 ...
##  $ SO      : int  0 6 0 0 1 0 0 7 2 0 ...
##  $ IBB     : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ HBP     : int  NA 1 NA NA NA NA NA NA NA NA ...
##  $ SH      : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ SF      : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ GIDP    : int  NA NA NA NA NA NA NA 1 1 NA ...
```

```
# Now, let's use the info given in the columns to create a new column for 'slugging percentage' a
# baseball stat that is meant to reflect how damging it is when a player does get a hit. For example,
# Barry Bonds had a high Slugging percentage because he did damage much of the time when we successfull
# got a hit. I'll give you a hint; Barry did not like hitting singles!
#
# Per the MLB website, the formula for slugging percentage is: (1B + 2Bx2 + 3Bx3 + HRx4)/AB.
new_hitting$slugging <- (((new_hitting$H) + ((new_hitting$X2B)*2) + ((new_hitting$X3B)*3) + ((new_hitti
View(new_hitting)

# After observing how many NAs were in my new slugging column, I will remove them so we just have cases
noNA_slug <- subset(new_hitting, !is.na(new_hitting$slugging))
View(noNA_slug)
```

DURING-Session REFLECTION

POST-Session REFLECTION

TBD..