Money-Ball Project

Kush Jani

6/19/2020

Let’s get started!

Follow the steps outlined in bold below using your new R skills and help the Oakland A’s recruit under-valued players!

Use R to open the Batting.csv file and assign it to a dataframe called batting using read.csv

```
here I mentioned my path in code , You have to write your path as you have saved your fie in your PC
```

```
batting <- read.csv('C:/Users/Kush/Documents/R/R-Course-HTML-Notes/R-Course-HTML-Notes/R-for-Data-Science-and-Machine-Learning/Training Exercises/Capstone and Data Viz Projects/Capstone Project/Batting.csv')
```

Use head() to check out the batting

```
head(batting)
```

Use str() to check the structure. Pay close attention to how columns that start with a number get an ‘X’ in front of them! You’ll need to know this to call those columns!

```
str(batting)
```

Call the head() of the first five rows of AB (At Bats) column

```
head(batting$AB)
```

Call the head of the doubles (X2B) column

```
head(batting$X2B)
```

Feature Engineering

We need to add three more statistics that were used in Moneyball! These are

```
Batting Average  On Base Percentage  Slugging Percentage
```

Batting Average is equal to H (Hits) divided by AB (At Base). So we’ll do the following to create a new column called BA and add it to our data frame:

```
batting$BA <- batting$H / batting$AB
```

After doing this operation, check the last 5 entries of the BA column of your data frame and it should look like this:

```
tail(batting$BA,5)
```

Now do the same for some new columns! On Base Percentage (OBP) and Slugging Percentage (SLG)

```
HINT:- For SLG, you need 1B (Singles), this isn't in your data frame.
However you can calculate it by subtracting doubles,triples, and home runs from total hits (H): 1B = H-2B-3B-HR
```

Create an OBP Column & Create an SLG Column On Base Percentage

```
batting$OBP <- (batting$H + batting$BB + batting$HBP)/(batting$AB + batting$BB + batting$HBP + batting$SF)
```

Creating X1B (Singles)

```
batting$X1B <- batting$H - batting$X2B - batting$X3B - batting$HR
```

Creating Slugging Average (SLG)

```
batting$SLG <- ((1 * batting$X1B) + (2 * batting$X2B) + (3 * batting$X3B) + (4 * batting$HR) ) / batting$AB
```

Check the structure of your data frame using str()

```
str(batting)
```

Merging Salary Data with Batting Data

```
We know we don't just want the best players, we want the most undervalued players,
meaning we will also need to know current salary information! We have salary information in the csv file 'Salaries.csv'.
```

Complete the following steps to merge the salary data with the player stats!

Load the Salaries.csv file into a dataframe called sal using read.csv

```
sal <- read.csv('C:/Users/Kush/Documents/R/R-Course-HTML-Notes/R-Course-HTML-Notes/R-for-Data-Science-and-Machine-Learning/Training Exercises/Capstone and Data Viz Projects/Capstone Project/Salaries.csv')
```

Use summary to get a summary of the batting data frame and notice the minimum year in the yearID column. Our batting data goes back to 1871! Our salary data starts at 1985, meaning we need to remove the batting data that occured before 1985 & Use subset() to reassign batting to only contain data from 1985 and onwards

```
summary(batting)
batting <- subset(batting,yearID >= 1985)
```

Now use summary again to make sure the subset reassignment worked, your yearID min should be 1985

```
summary(batting)
```

Use the merge() function to merge the batting and sal data frames by c('playerID','yearID'). Call the new data frame combo

```
combo <- merge(batting,sal,by=c('playerID','yearID'))
```

Use summary to check the data

```
summary(combo)
```

Analyzing the Lost Players

```
As previously mentioned, the Oakland A's lost 3 key players during the off-season.
We'll want to get their stats to see what we have to replace.
The players lost were: first baseman 2000 AL MVP Jason Giambi (giambja01) to the New York Yankees,
outfielder Johnny Damon (damonjo01) to the Boston Red Sox and infielder Rainer Gustavo "Ray" Olmedo ('saenzol01').
```

Use the subset() function to get a data frame called lost_players from the combo data frame consisting of those 3 players. Hint: Try to figure out how to use %in% to avoid a bunch of or statements!

```
lost_players <- subset(combo,playerID %in% c('giambja01','damonjo01','saenzol01') )
lost_players
```

```
Since all these players were lost in after 2001 in the offseason, let's only concern ourselves with the data from 2001.
```

Use subset again to only grab the rows where the yearID was 2001.

```
lost_players <- subset(lost_players,yearID == 2001)
```

Reduce the lost_players data frame to the following columns: playerID,H,X2B,X3B,HR,OBP,SLG,BA,AB

```
lost_players <- lost_players[,c('playerID','H','X2B','X3B','HR','OBP','SLG','BA','AB')]
head(lost_players)
```

Replacement Players

Now we have all the information we need! Here is your final task - Find Replacement Players for the key three players we lost!

However, you have three constraints: (1)The total combined salary of the three players can not exceed 15 million dollars. (2)Their combined number of At Bats (AB) needs to be equal to or greater than the lost players. (3)Their mean OBP had to equal to or greater than the mean OBP of the lost players

Example Solution

Note: There are lots of correct answers and ways to solve this!

First only grab available players from year 2001

```
library(dplyr)
avail.players <- filter(combo,yearID==2001)
```

Then I made a quick plot to see where I should cut-off for salary in respect to OBP:

```
library(ggplot2)
ggplot(avail.players,aes(x=OBP,y=salary)) + geom_point()
```

Looks like there is no point in paying above 8 million or so (I'm just eyeballing this number). I'll choose that as a cutt off point. There are also a lot of players with OBP==0. Let's get rid of them too.

```
avail.players <- filter(avail.players,salary<8000000,OBP>0)
```

The total AB of the lost players is 1469. This is about 1500, meaning I should probably cut off my avail.players at 1500/3= 500 AB.

```
avail.players <- filter(avail.players,AB >= 500)
```

Now let's sort by OBP and see what we've got!

```
possible <- head(arrange(avail.players,desc(OBP)),10)
```

Grab columns I'm interested in:

```
possible <- possible[,c('playerID','OBP','AB','salary')]
possible
```

Can't choose giambja again, but the other ones look good (2-4). I choose them!

```
possible[2:4,]
```

Great, looks like I just saved the 2001 Oakland A's a lot of money! If only I had a time machine and R, I could have made a lot of money in 2001 picking players!

Great Job! Here we completed MoneyBall project!

The End!