Classwork

Immanuel Williams Ph.D.

8/18/2020

Every Time You Create A Dataframe, Explore It

```
install.packages("devtools")
devtools::install_git("https://github.com/jamesijw23/stat_calpoly_package.git")

library(statcalpolypackage) ## to gain access to dataframe
```

HW1. There are two dataframes named, **basic_info_adult_df** and **game_info_adult_df**. Combine the columns of these dataframes, provide an appropriate name for the dataframe, and specify the variable types in the combined dataframe.

HW2. There are two dataframes named, **game_info_teen_df** and **game_info_adult_df.** Combine the rows of these dataframes, provide an appropriate name for the dataframe, and specify the variable types in the combined dataframe.

HW3. There are two dataframes named, **basic_info_adult_f1_df** and **basic_info_adult_f2_df**. Combine the rows of these dataframes based on the common ids, provide an appropriate name for the dataframe, and specify the variable types in the combined dataframe.

HW4. There are two dataframes named, **basic_info_adult_f1_df** and **basic_info_adult_f2_df**. Combine the rows of these dataframes based on the common ids where your resulting dataframe should have the rows from **basic_info_adult_f2_df**, provide an appropriate name for the dataframe, and specify the variable types in the combined dataframe.

HW5. There are two dataframes named, **basic_info_adult_f1_df** and **basic_info_adult_f2_df.** Combine the rows of these dataframes based on the common ids where your resulting dataframe should have the rows from both original dataframes, provide an appropriate name for the dataframe, and specify the variable types in the combined dataframe.

Tuesday: Join

Run your code from yesterday.

Extraction of Data

Make sure you have access to yesterday’s Michael Jordan clean dataframe. Continue work from the same R Markdown file.

Q1. Extract LeBron James’s 2012 dataframe from the NBA website using the following website: http://www.basketball-reference.com/players/j/jamesle01/gamelog/2012/

Name this dataframe lj_2012_df.

Q2. Clean up the data by doing the following:

1. Create the *Game_Location* and *Game_Outcome* variables using the same process for Michael Jordan (Remember you have to modify the column name.)
2. Within the Game_Location variable, change the rows that have the ‘@’ symbol to be ‘Away’ and the ” to be ‘Home’
3. Remove the rows that does not have data
4. Turn the same quantitative variables to be numeric as was done for the Michael Jordan dataframe
5. Save this dataframe as **mod1_lf_2012_df**

Q3. There is a variable labelled ‘+/-’ in the **mod1_lj_2012_df** this needs to be removed. Remove this variable and save this dataframe as **mod2_lf_2012_df** (*Hint: You need to use tick marks.*)

Replication

The *rep()* function allows you to create replications of a number, letter or vector.

```
rep('happy',5)

## [1] "happy" "happy" "happy" "happy" "happy"
```

Dimension

The *dim()* function show the number of rows and columns a dataframe has.

```
dim(mtcars)

## [1] 32 11
```

Q4. Part a: Create a dataframe named **mj_info_df** that has two columns

- Variable Name: *Player* that says 'Jordan' for each row based on the number of rows from **mod4_mj_1991_df**
- Variable Name: *Year* that says 'year_1991' for for each row based on the number of rows from **mod4_mj_1991_df**

Q4. Part b: Create a dataframe named **lj_info_df** that has two columns

- Variable Name: *Player* that says 'James' for each row based on the number of rows from **mod2_lj_2012_df**
- Variable Name: *Year* that says 'year_2012' for for each row based on the number of rows from **mod2_lj_2012_df**

Q5. Part a: Bind the **mj_info_df** to **mod3_mj_1991_df**, save resulting dataframe as **mj_df**. *Hint: You have two options for binding: column binf or row bind. What makes the most sense in this scenario.*

Q5. Part b: Bind the **lj_info_df** to **mod2_lj_2012_df**, save resulting dataframe as **lj_df**. *Hint: You have two options for binding: column bind or row bind. What makes the most sense in this scenario. Be consistent with what the order as Q5*

Q6. Bind **mj_df** and **lj_df** together, save resulting dataframe as **great_players_df**. *Hint: You have two options for binding: column bind or row bind. What makes the most sense in this scenario.* Also remove the *Binary_GmSc* variable.

Q7. Download *teams_abbreviation.csv* from the website. Save the csv file where you saved this R markdown file is saved. Use the *read.csv()* function to read-in in this csv, seen below. The *stringAsFactors = FALSE* ensures that strings are treated as strings in R.

```r
## Change Directory by using setwd()
team_abbrev_df = read.csv('teams_abbreviation.csv',stringsAsFactors = FALSE)
```

Q8. Explore the **team_abbrev_df**. What extra information is in this dataframe that is not in the **great_players_df**? How could you use this information to analyze these players better?

Rename Variable Name (This is another way)

We can rename variables using the rename function. We use the following

```
mod1_mtcars = mtcars %>%
  rename(Miles_Per_Gallon = mpg)
head(mod1_mtcars)

##                   Miles_Per_Gallon cyl disp Horse_Power drat    wt  qsec vs am
## Mazda RX4                     21.0   6  160         110 3.90 2.620 16.46  0  1
## Mazda RX4 Wag                 21.0   6  160         110 3.90 2.875 17.02  0  1
## Datsun 710                    22.8   4  108          93 3.85 2.320 18.61  1  1
## Hornet 4 Drive                21.4   6  258         110 3.08 3.215 19.44  1  0
## Hornet Sportabout             18.7   8  360         175 3.15 3.440 17.02  0  0
## Valiant                       18.1   6  225         105 2.76 3.460 20.22  1  0
##                   gear carb
## Mazda RX4            4    4
## Mazda RX4 Wag        4    4
## Datsun 710           4    1
## Hornet 4 Drive       3    1
## Hornet Sportabout    3    2
## Valiant              3    1

head(mtcars)

##                    mpg cyl disp Horse_Power drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160         110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160         110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108          93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258         110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360         175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225         105 2.76 3.460 20.22  1  0    3    1
```

Q9. Rename the *Opp* variable in the **great_players_df** to be *opp_team_abbrev*. Keep the **great_players_df** as the dataframe name.

Q10. Join **great_players_df** and **team_abbrev_df**, using *opp_team_abbrev* as the joining variable (like the id variable in the videos). Save this dataframe as **mod1_great_players_df**. *Hint: You have three options for joining: inner join, left join or full join. What makes the most sense in this scenario.*

Q11. Create a plot to visualize the relationship between *Conference*, and *PTS* (Points). Facet this relation on *Player*. Use the **mod1_great_players_df**. Make sure to label the x and y axes as well as the title. Make sure the title is centered. What pattern do you notice in this plot?

Q12. Create a plot to visualize the relationship between *AST* (Assists) and *PTS* (Points). Facet this relationship on *Conference*. Use the **mod1_great_players_df**. Make sure to label the x and y axes as well as the title. Make sure the title is centered. What pattern do you notice in this plot?

Q13. Create plot using one quantitative and two categorical variables. Use variables that are different from **Q10.** and **Q11.** and make sense in the context in basketball. Make sure to label the x and y axes as well as the title. Make sure the title is centered. What pattern do you notice in this plot?