Homework 7

Question 1

First, read the TED talk data `TED_Talks.csv` into R as a data frame named `ted`.

```
ted <- read.csv("TED_Talks.csv",
                stringsAsFactors = FALSE,
                header = TRUE)
```

Next, create a subset called `ted2`, subsetting when Hans Rosling was the speaker.

```
ted2 <- ted[ted$speaker == "Hans Rosling",]
```

```
ted$headline[ted$speaker == "Hans Rosling"]
```

```
## [1] "The best stats you’ve ever seen"
## [2] "New insights on poverty"
## [3] "Insights on HIV, in stunning data visuals"
## [4] "Let my dataset change your mindset"
## [5] "Asia’s rise -- how and when"
## [6] "Global population growth, box by box"
## [7] "The good news of the decade? We’re winning the war against child mortality"
## [8] "The magic washing machine"
## [9] "Religions and babies"
```

The above list represents the headlines for the TED Talks delivered by Hans Rosling.

Using `ted2`, replace the `speaker` with your name **(1 pt)**. Remember `R`’s recycling rules, you should only have to type your name once in order to replace Hans Rosling with your own name for every occurrence.

After having made this modification, you can check the effectiveness of your code using the following:

```
knitr::kable(ted[ted$speaker == "Hans Rosling", 2:3])

knitr::kable(ted2[, 2:3])
```

This code should produce a data frame with 9 rows and 2 columns. The first column should contain 9 instances of your name. The second column should contain the 9 headlines for Hans Rosling’s TED Talks. Can you explain why the above command works?

Logical subsetting is very powerful because it allows you to quickly and easily identify, extract, and modify individual values in your data set.

Lets try to exercise our logical subsetting skills a bit:

- Create a new data set called `ted_17` that contains only the TED Talks that *occurred in 2017*. How many talks does this represent? **(2 pts)**.

- In the ted2 data frame, create a new variable called popular that contains a Y if the talk exceeded a million views as of 6/16/17 and N if the talk did not **(2 pts)**. (Hint: you may want to start by creating a new column and setting all values to N. You can then update the column in a second command to place the values of Y where appropriate).

Print out the results:

```
knitr::kable(ted2[, c("headline", "views_as_of_06162017", "popular")])
```

Question 2

Using the AirBnB dataset, airbnb_los_angeles_2017_03_10.csv, use tapply to calculate mean price *by neighborhood* and write the results of this function out to a new data frame called avg.price **(3 pts)**.

```
airbnb <- read.csv("airbnb_los_angeles_2017_03_10.csv",
                   stringsAsFactors = FALSE,
                   header = TRUE)
```

Question 3

The dplyr package is an extremely powerful and useful library for both data manipulation and summarization. Using the AirBnB data set, calculate *mean overall satisfaction by room type* using the summarize function from dplyr **(3 pts)**.

Major League Baseball Data

To practice summarizing data in R, we will familiarize ourselves with a new data set containing team information by year for each of the existing 30 teams from 1876 to 2016. This data set was originally compiled for an analysis of coaching records and to attempt to answer the question of why managers change jobs. The data was originally extracted from https://www.baseball-reference.com.

The following variables are included in the baseballdata.csv file:

Variable	Description
Year	Calendar year
Tm	Team name in the calendar year
Lg	League
G	Total games played
W	Total games won
L	Total games lost
Ties	Total games tied
WL	Win-Loss Percentage
Finish	Standing at the end of season
GB	Games back relative to team in first place
Playoff	Information about how the team finished the playoffs, if they participated
R	Total runs earned
RA	Total runs allowed
Attendance	Annual attendance at games
BatAge	Average age of all batters on the team
PAge	Average age of all pitchers on the team

Variable	Description
`TopPlayer`	The best player on the team in that calendar year
`Managers`	The team's manager or managers in that calendar year

Import the data into R into a data frame named `baseball`.

```r
baseball <- read.csv("baseballdata.csv",
                     sep = ",",
                     header = TRUE,
                     stringsAsFactors = FALSE)
```

Question 4

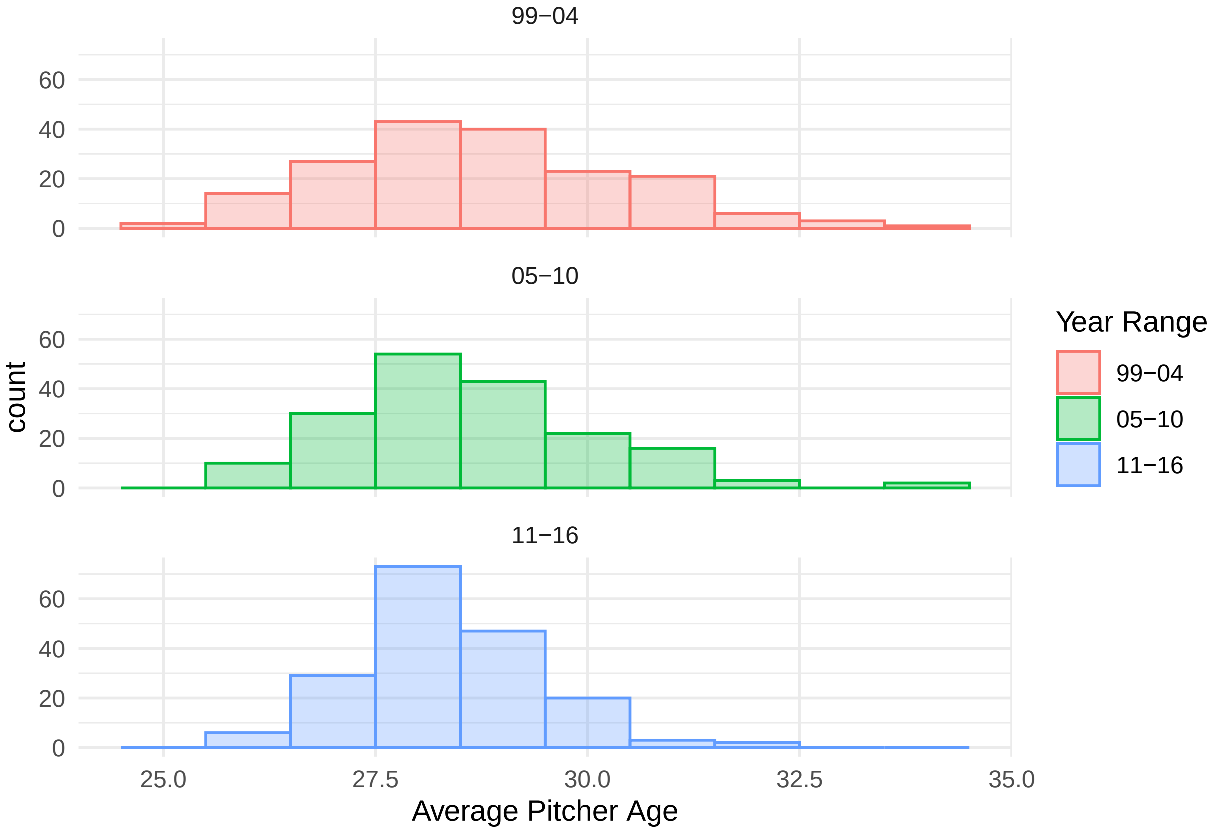
First, subset the baseball data to only include the 1999 to 2016 seasons.

- Create a variable called `Year_cut` using the `cut()` functions. The breaks should be `c(1999, 2004, 2010, 2016)`, make sure to include 1999 data by setting `include.lowest = TRUE` and appropriately label each factor level **(2 pts)**.
- Create a variable called `RA_avg` that calculates the **average number of runs allowed per game (1 pt)**.

Using the `ggplot2` package, we can easily create complex graphical summaries of the data, for example, in the code below, we look at the distribution of average pitcher ages across each level of `Year_cut`.

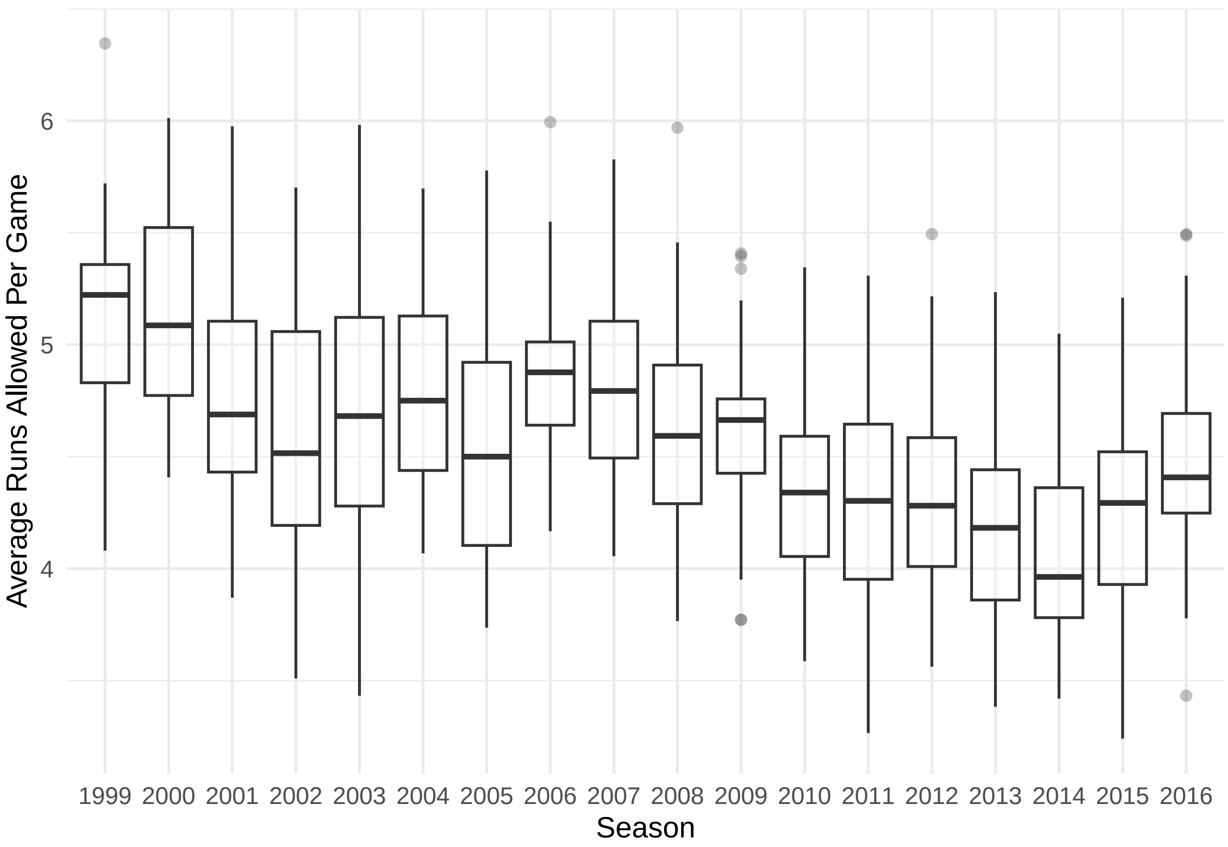
```r
library(ggplot2)

ggplot(baseball_sub, aes(x = PAge, fill = Year_cut, color = Year_cut)) +
  geom_histogram(binwidth = 1, alpha = 0.3) +
  facet_wrap(vars(Year_cut), ncol = 1) +
  scale_x_continuous(breaks = c(25, 27.5, 30, 32.5, 35), minor_breaks = NULL) +
  labs(x = "Average Pitcher Age",
       fill = "Year Range",
       color = "Year Range") +
  theme_minimal()
```



Using `ggplot2`, graphically summarize the the distribution of average runs allowed per game by `Year`. Make sure that the plot is appropriately labelled. Do you see any trends? **(3 pts)**

As a hint, the correct plot is below, you need to write the code to reproduce it.



Next, create a scatterplot of average runs allowed per game by average pitcher age, stratified by `Year_cut`. Include a best fit linear regression line for each year range along with a 95% confidence band. Interpret the plot. **(3 pts)**