Homework 7

Question 1

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

```
ted <- read.csv("/Users/hanbeixiong/Desktop/UCLA_courses/Biostat203A/HW7/TED_Talk
        s.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 mortalit
y"
## [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.

```
ted2$speaker = "Hanbei Xiong"
```

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])
```

	speaker	headline
88	Hans Rosling	The best stats you’ve ever seen
123	Hans Rosling	New insights on poverty
441	Hans Rosling	Insights on HIV, in stunning data visuals
497	Hans Rosling	Let my dataset change your mindset
561	Hans Rosling	Asia’s rise – how and when
730	Hans Rosling	Global population growth, box by box
787	Hans Rosling	The good news of the decade? We’re winning the war against child mortality
896	Hans Rosling	The magic washing machine
1241	Hans Rosling	Religions and babies

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

	speaker	headline
88	Hanbei Xiong	The best stats you’ve ever seen
123	Hanbei Xiong	New insights on poverty
441	Hanbei Xiong	Insights on HIV, in stunning data visuals
497	Hanbei Xiong	Let my dataset change your mindset
561	Hanbei Xiong	Asia’s rise – how and when
730	Hanbei Xiong	Global population growth, box by box

	speaker	headline
787	Hanbei Xiong	The good news of the decade? We're winning the war against child mortality
896	Hanbei Xiong	The magic washing machine
1241	Hanbei Xiong	Religions and babies

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?

Answer: For command "knitr::kable(ted[ted$speaker == "Hans Rosling", 2:3])", Command "knitr::kable()" creates a nice table of what being included in the bracket. Command "ted[ted.speaker == "Hans Rosling", 2:3]" select speakers who are Hans Rosling and display the second and third columns of filtered ted dataframe. Command "knitr::kable(ted2[, 2:3])" works similarly. It used the dataframe we created previously which includes my name as speakers. It selects the 2nd and 3rd columns of the dataframe we created and display it nicely with command "knitr::kable()".

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)**.

```
ted_17=ted[ted$year_filmed==2017,]
nrow(ted_17)
```

```
## [1] 39
```

- 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).

```
ted2$popular <- ifelse(ted2$views_as_of_06162017 > 1e6, "Y", "N")
```

Print out the results:

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

	headline	views_as_of_06162017	popular
88	The best stats you've ever seen	11783283	Y
123	New insights on poverty	3203013	Y
441	Insights on HIV, in stunning data visuals	887145	N
497	Let my dataset change your mindset	1438347	Y
561	Asia's rise – how and when	1717975	Y
730	Global population growth, box by box	2864937	Y
787	The good news of the decade? We're winning the war against child mortality	731062	N
896	The magic washing machine	2359215	Y
1241	Religions and babies	2100043	Y

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("/Users/hanbeixiong/Desktop/UCLA_courses/Biostat203A/HW7/airbn
        b_los_angeles_2017_03_10.csv",
                   stringsAsFactors = FALSE,
                   header = TRUE)
```

```
#tapply(airbnb$price, airbnb$neighborhood, mean)
mean_prices = tapply(airbnb$price, airbnb$neighborhood, mean)
avg.price = data.frame(
  neighborhood = names(mean_prices),
  mean_price = mean_prices
)
head(avg.price)
```

```
##                     neighborhood mean_price
## Adams-Normandie Adams-Normandie   75.92105
## Agoura Hills       Agoura Hills  138.61111
## Alhambra               Alhambra   87.36424
## Alondra Park       Alondra Park  113.60000
## Altadena               Altadena  145.86391
## Arcadia                 Arcadia   94.32768
```

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)**.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
result <- airbnb %>%
  group_by(room_type) %>%
  summarize(mean_satisfaction = mean(overall_satisfaction, na.rm = TRUE))
result
```

```
## # A tibble: 3 × 2
##   room_type       mean_satisfaction
##   <chr>                       <dbl>
## 1 Entire home/apt              3.03
## 2 Private room                 2.69
## 3 Shared room                  2.08
```

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 (https://www.baseball-reference.com).

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

Variable	Description
`Year`	Calendar year

Variable	Description
`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
`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`.

```
baseball <- read.csv("/Users/hanbeixiong/Desktop/UCLA_courses/Biostat203A/HW7/bas
        eballdata.csv",
                    sep = ",",
                    header = TRUE,
                    stringsAsFactors = FALSE)
```

Question 4

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

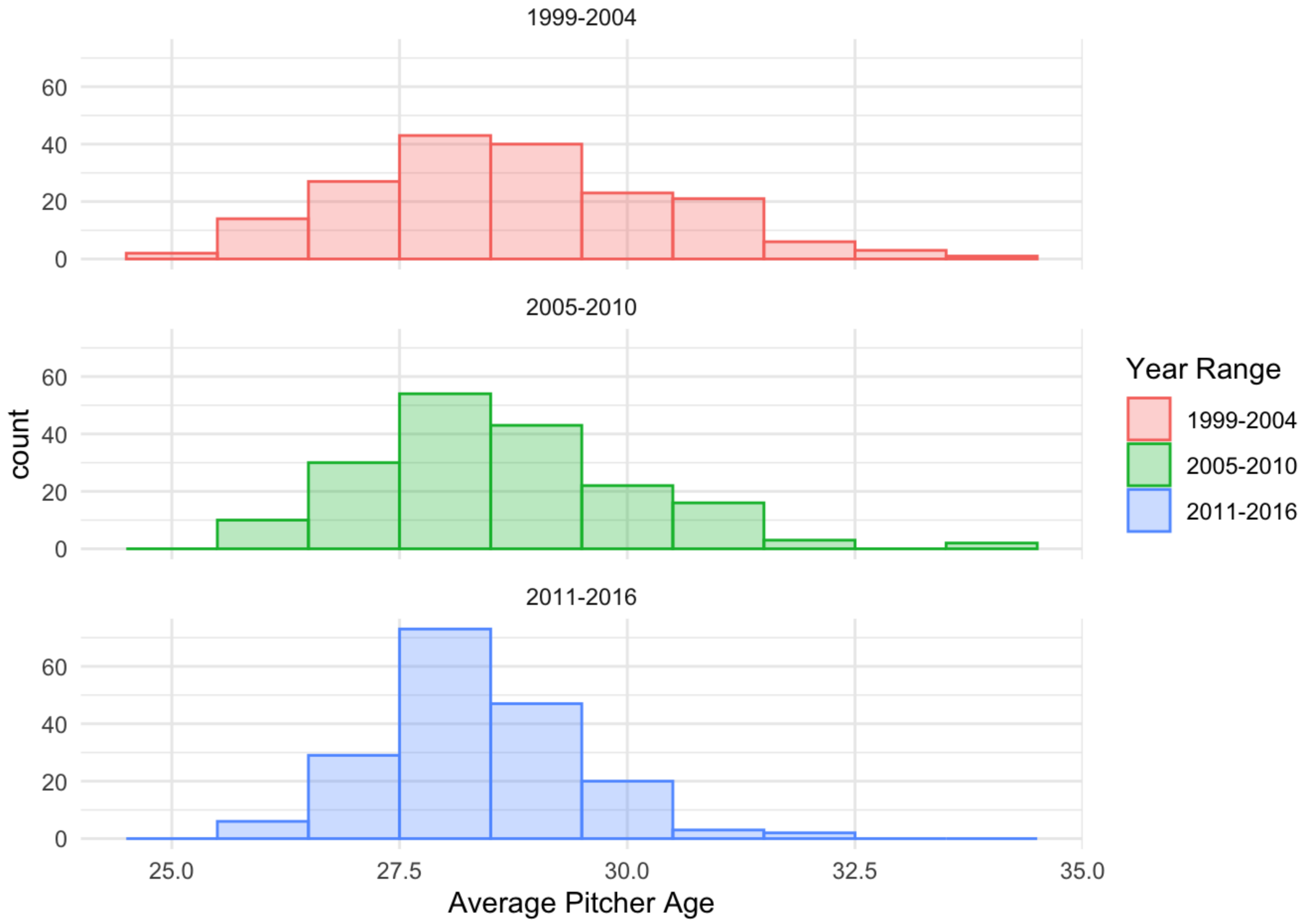
- Create a variable called `Year_cut` using the `cut()` functions. The breaks should be `c(1999, 2004, 2010, 2016)`, make sure to include 1999 data by setting `include.lowest = TRUE` and appropriately label each factor level **(2 pts)**.
- Create a variable called `RA_avg` that calculates the **average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

```
baseball_sub=baseball[baseball$Year>=1999 & baseball$Year<=2016,]
Year_cut = cut(baseball_sub$Year,breaks=c(1999, 2004, 2010, 2016),include.lowest
        = TRUE,labels = c("1999-2004", "2005-2010", "2011-2016"))
RA_avg = baseball_sub$RA / baseball_sub$G
baseball_sub$Year_cut = Year_cut
baseball_sub$RA_avg = RA_avg
```

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`.

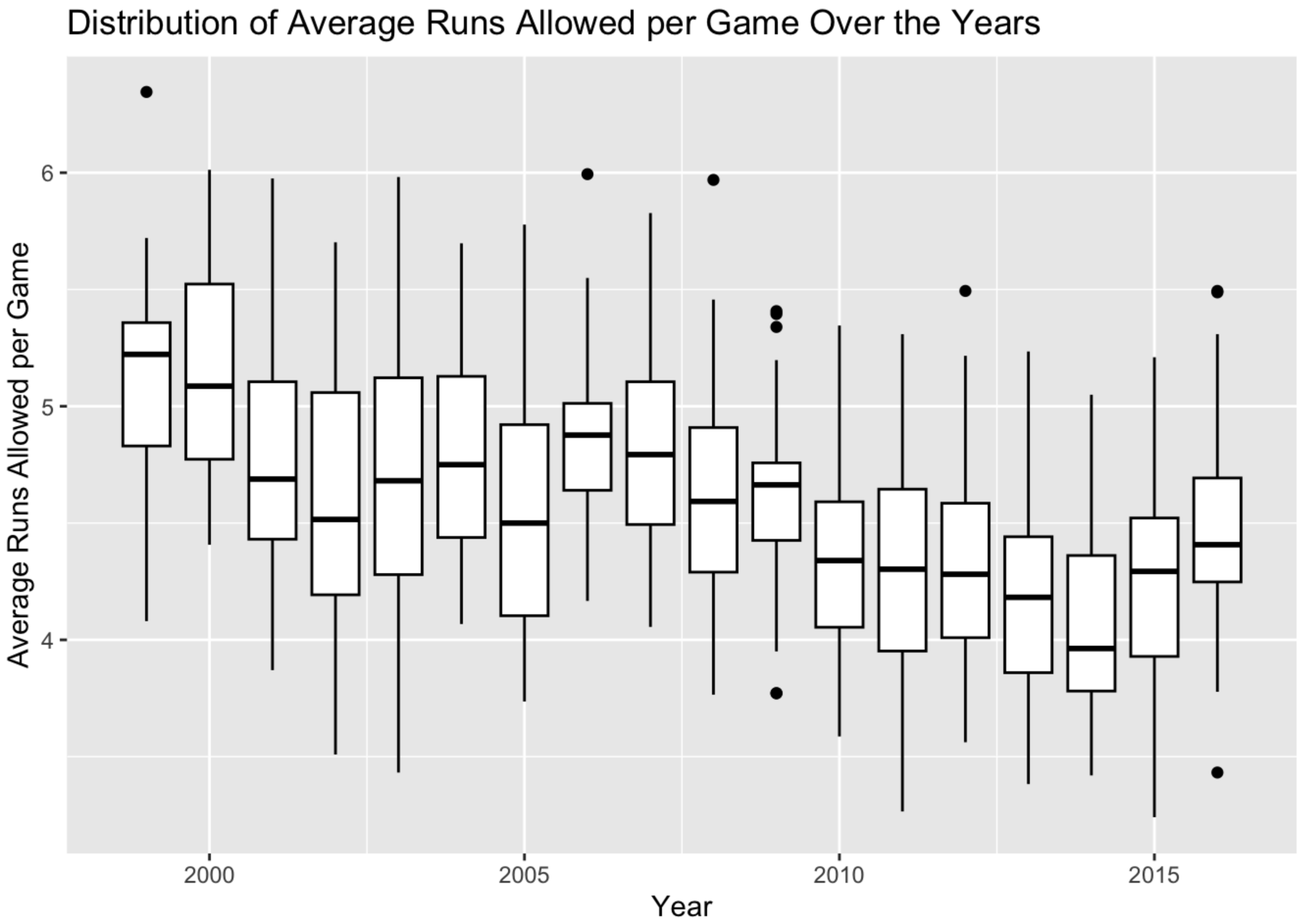
```
library(ggplot2)

ggplot(baseball_sub, aes(x = PAge, fill = Year_cut, color = Year_cut)) +
  geom_histogram(binwidth = 1, alpha = 0.3, position = "identity") +
  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 average runs allowed per game by `Year`. Make sure that the plot is appropriately labelled. Do you see any trends? **(3 pts)**

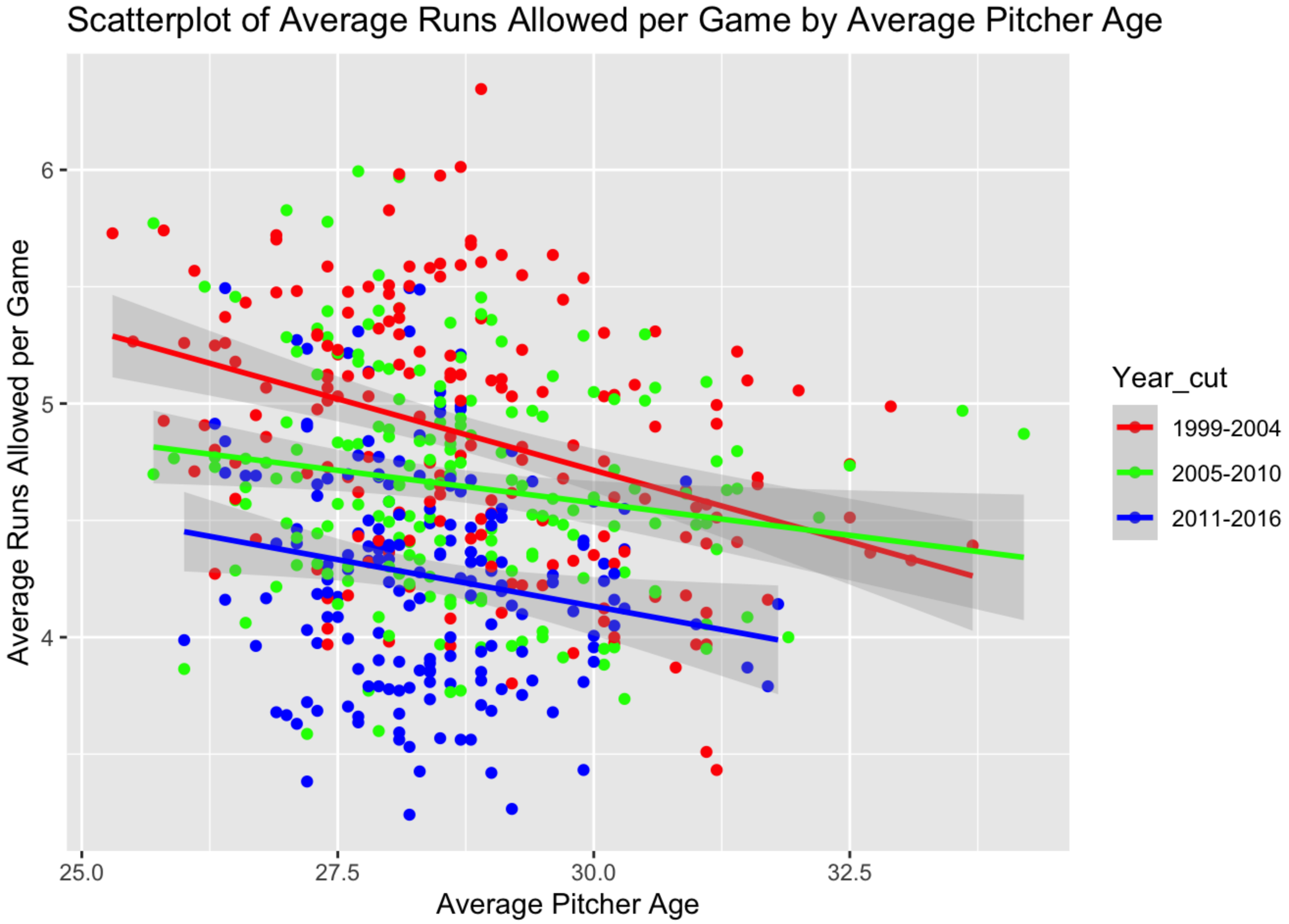
```
ggplot(baseball_sub, aes(x = Year, y = RA_avg,group=Year)) +
  geom_boxplot(color = "black") +
  labs(title = "Distribution of Average Runs Allowed per Game Over the Years",
       x = "Year",
       y = "Average Runs Allowed per Game")
```



Answer: By observation, as year gets larger, the average runs allowed per game decreases.

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 along with a 95% confidence band. **(3 pts)**

```
ggplot(baseball_sub, aes(x = PAge, y = RA_avg, color = Year_cut)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE, alpha = 0.3) +
  labs(title = "Scatterplot of Average Runs Allowed per Game by Average Pitcher A
        ge",
       x = "Average Pitcher Age",
       y = "Average Runs Allowed per Game") +
  scale_color_manual(values = c("1999–2004" = "red", "2005–2010" = "green", "2011
        –2016" = "blue"))
```



Interpret: All three lines show decrease patterns that as average picher Age increases, the average runs allowed per game decrease. In the year cut between 1999-2004, the slope is more extreme than the other two lines.