A. The Ultimate Halloween Candy Power Ranking

1. Find the top 5 best rated and top 5 worst rated candy.

Code:

```
# top 5 worst rated candy

candy_rankings %>%

  group_by(competitorname) %>%

  summarize(winpercent) %>%

  arrange(winpercent) %>%

  top_n(-5)
```

Result:

	competitorname	winpercent
	<chr>	*<dbl>*
1	Nik L Nip	22.4
2	Boston Baked Beans	23.4
3	Chiclets	24.5
4	Super Bubble	27.3
5	Jawbusters	28.1

Code:

```
# top 5 best rated candy

candy_rankings %>%

  group_by(competitorname) %>%

  summarize(winpercent) %>%

  arrange(desc(winpercent)) %>%

  top_n(5)
```

Result:

	competitorname	winpercent
	<chr>	*<dbl>*
1	Reese's Peanut Butter cup	84.2
2	Reese's Miniatures	81.9
3	Twix	81.6
4	Kit Kat	76.8
5	Snickers	76.7

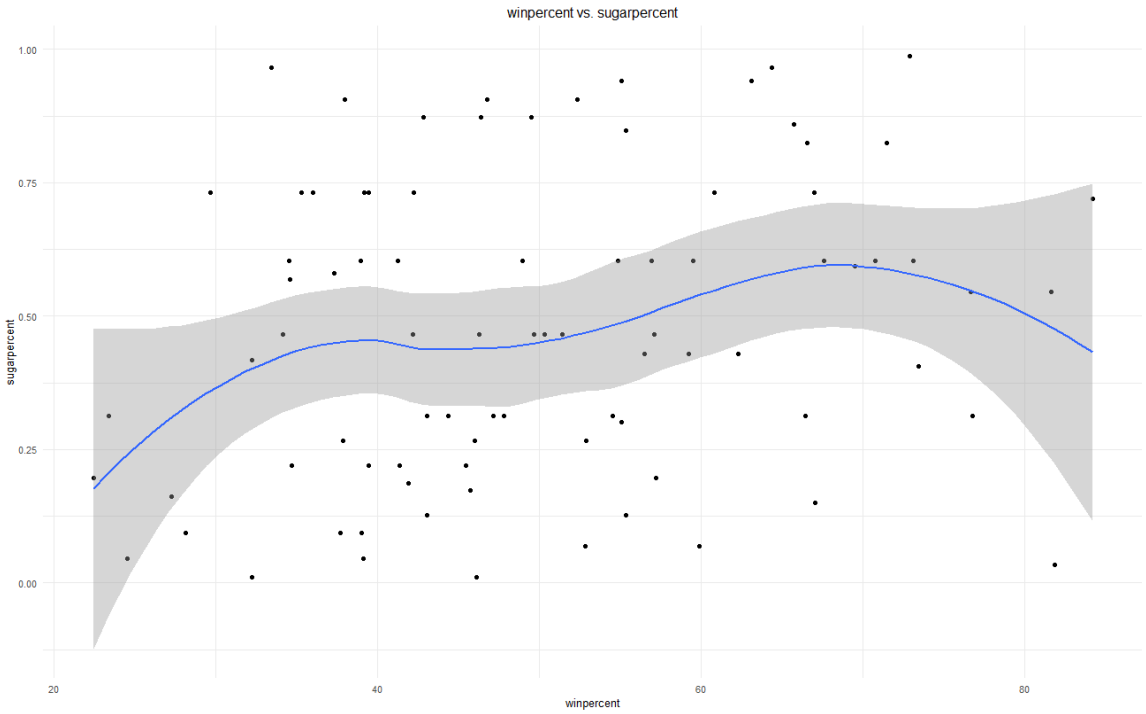
2. Plot winpercent against sugarpercent. Do you see any association? Now, plot winpercent against pricepercent. Do you see any association?

Code:

```
qplot(x = winpercent, y = sugarpercent, data = candy_rankings) +
```

```
  geom_smooth() +
  ggtitle("winpercent vs. sugarpercent") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(text=element_text(size=10))
```

Result:

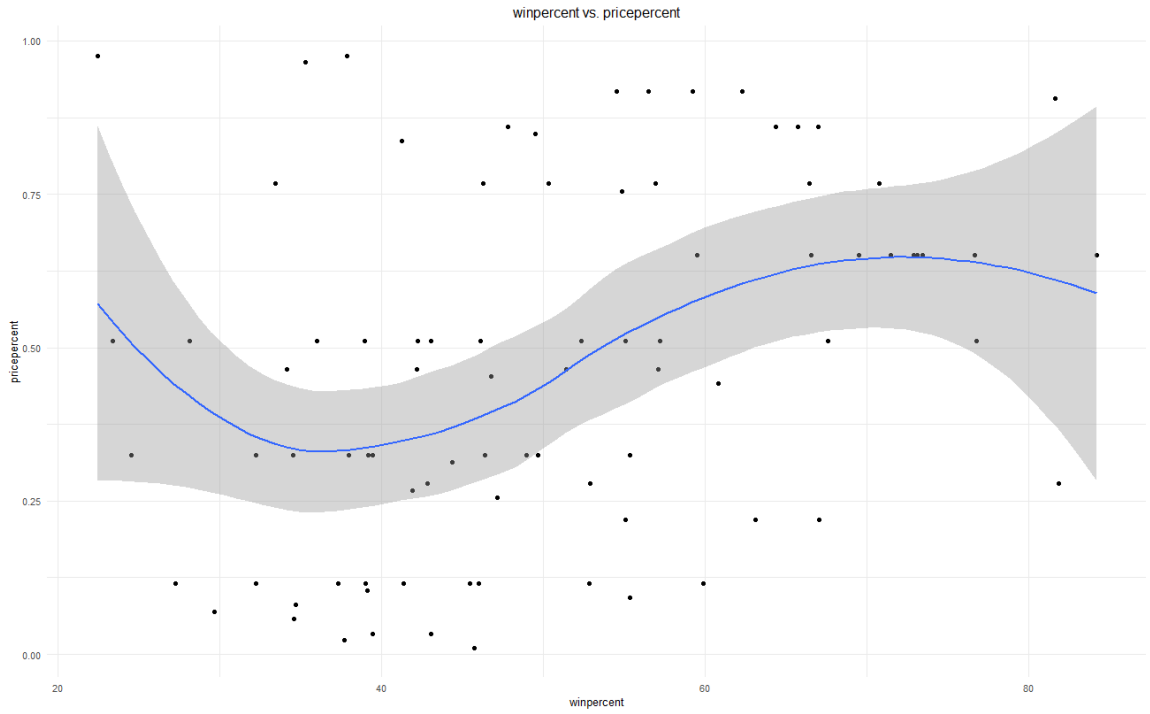


Yes. There's weak a non-linear correlation between winpercent and sugarpercent. The correlation is positive when winpercent is lower than 40; the correlation is negative when winpercent is larger than 70; there's no obvious correlation when winpercent is within the range of 40 and 70.

Code:

```
qplot(x = winpercent, y = pricepercent, data = candy_rankings) +
  geom_smooth() +
  ggtitle("winpercent vs. pricepercent") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(text=element_text(size=10))
```

Result:



The correlation between winpercent and pricepercent looks stronger compared to the correlation between winpercent and sugarpercent. When winpercent is lower than 40, the correlation between winpercent and pricepercent is negative and when winpercent is larger than 40, the correlation appears to be positive.

3. Consider all the logical-type variables in the dataset. For each logical variable, find the average difference in winpercent between the treats that satisfy the condition and the treats that don't satisfy it. Which logical variable seems to have the strongest effect on winpercent?

Code:

```
candy_rankings %>% group_by(chocolate) %>% summarize(avgWinPerc = mean(winpercent, na.rm = TRUE))

candy_rankings %>% group_by(fruity) %>% summarize(avgWinPerc = mean(winpercent, na.rm = TRUE))

candy_rankings %>% group_by(caramel) %>% summarize(avgWinPerc = mean(winpercent, na.rm = TRUE))

candy_rankings %>% group_by(peanutyalmondy) %>% summarize(avgWinPerc = mean(winpercent, na.rm = TRUE))

candy_rankings %>% group_by(nougat) %>% summarize(avgWinPerc = mean(winpercent, na.rm = TRUE))
```

candy_rankings %>% group_by(crispedricewafer) %>% summarize(avgWinPerc = mean(winpercent, na.rm = TRUE))

candy_rankings %>% group_by(hard) %>% summarize(avgWinPerc = mean(winpercent, na.rm = TRUE))

candy_rankings %>% group_by(bar) %>% summarize(avgWinPerc = mean(winpercent, na.rm = TRUE))

candy_rankings %>% group_by(pluribus) %>% summarize(avgWinPerc = mean(winpercent, na.rm = TRUE))

Result:

```
> candy_rankings %>%
+   group_by(chocolate) %>%
+   summarize(avgWinPerc = mean(winpercent, na.rm = TRUE))
# A tibble: 2 x 2
  chocolate avgWinPerc
  <lgl>          <dbl>
1 FALSE           42.1
2 TRUE            60.9
>
> candy_rankings %>%
+   group_by(fruity) %>%
+   summarize(avgWinPerc = mean(winpercent, na.rm = TRUE))
# A tibble: 2 x 2
  fruity avgWinPerc
  <lgl>       <dbl>
1 FALSE        55.3
2 TRUE         44.1
>
> candy_rankings %>%
+   group_by(caramel) %>%
+   summarize(avgWinPerc = mean(winpercent, na.rm = TRUE))
# A tibble: 2 x 2
  caramel avgWinPerc
  <lgl>        <dbl>
1 FALSE         48.9
2 TRUE          57.3
>
> candy_rankings %>%
+   group_by(peanutyalmondy) %>%
+   summarize(avgWinPerc = mean(winpercent, na.rm = TRUE))
# A tibble: 2 x 2
  peanutyalmondy avgWinPerc
  <lgl>                <dbl>
1 FALSE                 47.7
2 TRUE                  63.7
>
> candy_rankings %>%
+   group_by(nougat) %>%
+   summarize(avgWinPerc = mean(winpercent, na.rm = TRUE))
# A tibble: 2 x 2
  nougat avgWinPerc
  <lgl>       <dbl>
1 FALSE        49.4
2 TRUE         60.1
>
> candy_rankings %>%
+   group_by(crispedricewafer) %>%
```

```
+   summarize(avgWinPerc = mean(winpercent, na.rm = TRUE))
# A tibble: 2 x 2
  crispedricewafer avgWinPerc
  <lgl>                  <dbl>
1 FALSE                   48.9
2 TRUE                    66.2
>
> candy_rankings %>%
+   group_by(hard) %>%
+   summarize(avgWinPerc = mean(winpercent, na.rm = TRUE))
# A tibble: 2 x 2
  hard  avgWinPerc
  <lgl>      <dbl>
1 FALSE       52.4
2 TRUE        40.5
>
> candy_rankings %>%
+   group_by(bar) %>%
+   summarize(avgWinPerc = mean(winpercent, na.rm = TRUE))
# A tibble: 2 x 2
  bar   avgWinPerc
  <lgl>      <dbl>
1 FALSE       46.7
2 TRUE        61.3
>
> candy_rankings %>%
+   group_by(pluribus) %>%
+   summarize(avgWinPerc = mean(winpercent, na.rm = TRUE))
# A tibble: 2 x 2
  pluribus avgWinPerc
  <lgl>          <dbl>
1 FALSE           54.1
2 TRUE            46.8
```

As we can see from the results, chocolate seems to have the strongest effect on winpercent.

B. College admissions dataset

1. Find the percentage of men who applied and got in and the percentage of women who applied and got in. What do you see?

Code:

```
colAdm = read.csv("http://vicpena.github.io/admin.csv")

str(colAdm)

colAdm2 = colAdm %>% uncount(Freq)

100*round(prop.table(table(colAdm2$Gender, colAdm2$Admit)),4)
```

Result:

```
        Admitted Rejected
  Female    12.31    28.24
  Male      26.47    32.99
```

It looks like there're more female got admitted than male if we only focus on the overall admitted percentage. However, that's because the number of female who applied is larger than the one of male. To discuss further:

Code:

```
# men applied and got in

maleIn = colAdm %>% filter(Admit == "Admitted" & Gender ==
"Male")

male = colAdm %>% filter(Gender == "Male")

100*round(sum(maleIn$Freq)/sum(male$Freq),4)


# women applied and got in

femaleIn = colAdm %>% filter(Admit == "Admitted" & Gender ==
"Female")

female = colAdm %>% filter(Gender == "Female")

100*round(sum(femaleIn$Freq)/sum(female$Freq),4)
```

Result:

```
> 100*round(sum(maleIn$Freq)/sum(male$Freq),4)
[1] 44.52
> 100*round(sum(femaleIn$Freq)/sum(female$Freq),4)
[1] 30.35
```

As we can see from the two percentages above, the percentage of women who got in is higher than men, but that's just because there're more women applied. Actually, male is easier to get admitted compared to women according to the data. 44.52% male who applied got admitted while only 30.35% female got admitted.

2. Now, find the percentage of men who applied and got in by department. Do the same with women. Compare the results with what you found in part 1.

Code:

```
colAdm2 = colAdm %>% uncount(Freq)

100*round(prop.table(table(colAdm2$Gender, colAdm2$Dept,
colAdm2$Admit),2),4)

male2 = male %>% uncount(Freq)

100*round(prop.table(table(male2$Dept, male2$Admit),1),4)

female2 = female %>% uncount(Freq)

100*round(prop.table(table(female2$Dept, female2$Admit),1),4)
```

Result:

```
> colAdm2 = colAdm %>% uncount(Freq)
> 100*round(prop.table(table(colAdm2$Gender, colAdm2$Dept, colAdm2$Admi
t),2),4)
, ,  = Admitted


             A     B     C     D     E     F
  Female  9.54  2.91 22.00 16.54 16.10  3.36
  Male   54.88 60.34 13.07 17.42  9.08  3.08

, ,  = Rejected


             A     B     C     D     E     F
  Female  2.04  1.37 42.59 30.81 51.20 44.40
  Male   33.55 35.38 22.33 35.23 23.63 49.16

>
> male2 = male %>% uncount(Freq)
> 100*round(prop.table(table(male2$Dept, male2$Admit),1),4)

    Admitted Rejected
  A    62.06    37.94
  B    63.04    36.96
  C    36.92    63.08
  D    33.09    66.91
  E    27.75    72.25
  F     5.90    94.10
>
> female2 = female %>% uncount(Freq)
> 100*round(prop.table(table(female2$Dept, female2$Admit),1),4)

    Admitted Rejected
  A    82.41    17.59
  B    68.00    32.00
  C    34.06    65.94
  D    34.93    65.07
  E    23.92    76.08
  F     7.04    92.96
```

As we can see from the tables above, men are easier to get admitted into department A and B, supported by the fact that the admitted percentages of these two departments are above the average male admitted percentage, which is 44.52. Women are easier to get into department A, B, C and D in the fact that the admitted percentages of these four departments are above the average female admitted percentage, which is 30.35.

3. Explain what is going on in this dataset. Do you see any evidence of gender discrimination?

Not really. Although the average admitted percentage of female is lower than male, each department's admitted percentages of male and female are almost the same (except for department A) according to the tables we got in question 2. Then the only explanation is that the percentage of women who applied department E and F, which are the top 2 department with the lowest admitted rate, are higher than men; the percentage of women who applied department A and B, which are the top 2 department with the highest admitted rate, are lower than the one of men. To support my conjecture:

Code:

```
male2 %>% group_by(male2$Dept) %>% summarise(malePerc =
100*n()/sum(male$Freq))

female2 %>% group_by(female2$Dept) %>% summarise(femalePerc =
100*n()/sum(female$Freq))
```

Result:

```
> male2 %>% group_by(male2$Dept) %>% summarise(Perc = 100*n()/sum(male
$Freq))
# A tibble: 6 x 2
  `male2$Dept`  Perc
  <fct>        <dbl>
1 A             30.7
2 B             20.8
3 C             12.1
4 D             15.5
5 E              7.10
6 F             13.9
> female2 %>% group_by(female2$Dept) %>% summarise(femalePerc = 100*n()
/sum(female$Freq))
# A tibble: 6 x 2
  `female2$Dept` femalePerc
  <fct>                <dbl>
1 A                     5.89
2 B                     1.36
3 C                    32.3
4 D                    20.4
5 E                    21.4
6 F                    18.6
```

C. Fandango movie ratings

1. Identify the Top 5 best rated and Top 5 worst rated movies in the dataset. Average over different platforms.

Code:

```
# create a new variable that represents the average score of a
film over different platforms

fandango$avgScore <- (fandango$fandango_ratingvalue +

  fandango$rt_norm +

  fandango$metacritic_norm +

  fandango$imdb_norm)/4

# top 5 worst rated movies

fandango %>%

  group_by(film) %>%

  summarize(avgScore) %>%
```

```
  select(film, avgScore) %>%
  arrange(avgScore) %>%
  top_n(-5)
# top 5 best rated movies
fandango %>%
  group_by(film) %>%
  summarize(avgScore) %>%
  select(film, avgScore) %>%
  arrange(desc(avgScore)) %>%
  top_n(5)
```

Result:

```
  film                   avgScore
  <chr>                     <dbl>
1 Fantastic Four             1.62
2 Paul Blart: Mall Cop 2     1.64
3 The Gallows                1.85
4 Hot Tub Time Machine 2     1.92
5 The Vatican Tapes          1.92
```

```
   film               avgScore
   <chr>                 <dbl>
 1 Inside Out             4.6
 2 Mad Max: Fury Road     4.44
 3 Selma                  4.44
 4 Song of the Sea        4.41
 5 Amy                    4.38
```

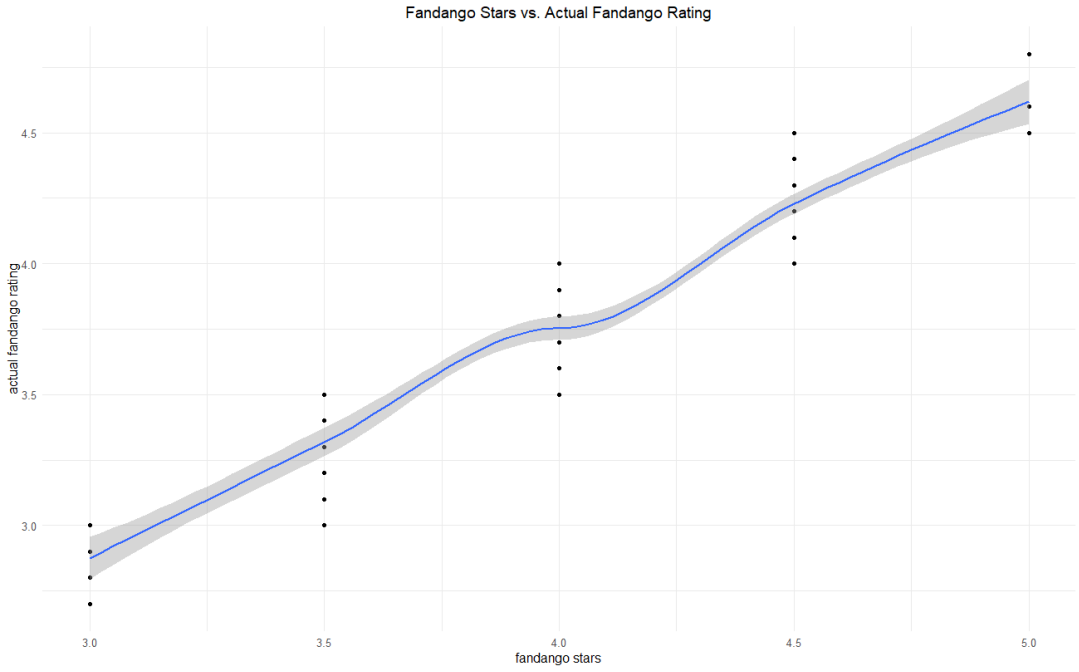
2. Visualize the difference between Fandango stars and actual Fandango ratings. Comment on what you see.

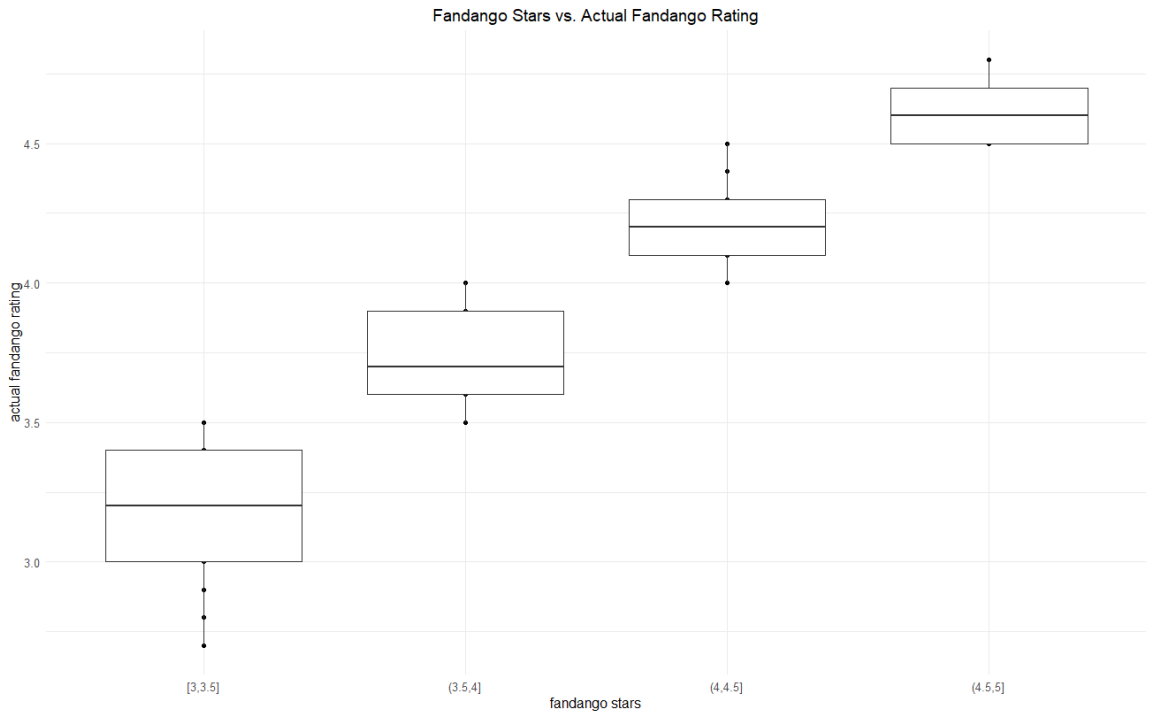
Code:

```
qplot(x = fandango_stars, y = fandango_ratingvalue, data =
fandango) +
  geom_smooth() +
  xlab("fandango stars") +
  ylab("actual fandango rating") +
  ggtitle("Fandango Stars vs. Actual Fandango Rating") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
```

```
  theme(text=element_text(size=12))


# another plot
# convert fandango stars to a categorical variable
quantile(fandango$fandango_stars)
fandango$fandango_stars = cut(fandango$fandango_stars,
                          breaks=quantile(fandango$fandango_stars),
                          include.lowest = TRUE)


qplot(x = fandango_stars, y = fandango_ratingvalue, data =
fandango) +
  geom_boxplot() +
  xlab("fandango stars") +
  ylab("actual fandango rating") +
  ggtitle("Fandango Stars vs. Actual Fandango Rating") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(text=element_text(size=12))
```

Result:





From the line plot, we can see there's a positive correlation between fandango stars and actual fandango ratings; from the boxplot, we can see that when fandango stars are below 4.5 or 5, the average score of actual fandango ratings are usually lower than the average score of fandango stars.

3. Some movies are loved by the critics, but hated by the audience (and sometimes, it's the other way around). Given the data you have, create a metric to measure discrepancies between user and critic ratings. Create a table that contains the Top 5 movies that seem to appeal to critics but not the audience, and another table with the Top 5 movies that users seem to like more than critics do.

Code:

```
# Note:

# IMDb doesn't have critic rating

# I didn't use fandango rating here because it doesn't seperate
critics' rating and users' rating

# Also didn't use rottentomatoes' user rating because the dataset
doesn't provide the number of rottentomatoes users, so it's hard
to normalize

fandango$criticScore <- (fandango$rt_norm +
fandango$metacritic_norm)/2

fandango$userScore <- (fandango$metacritic_user_nom *
fandango$metacritic_user_vote_count +
```

```
                          fandango$imdb_norm *
fandango$imdb_user_vote_count)/(fandango$metacritic_user_vote_cou
nt + fandango$imdb_user_vote_count)

fandango$diff <- fandango$criticScore - fandango$userScore


# top 5 movies users like more than critics

fandango %>%

  group_by(film) %>%

  summarize(diff) %>%

  select(film, diff) %>%

  arrange(diff) %>%

  top_n(-5)


# top 5 movies critics like more than users

fandango %>%

  group_by(film) %>%

  summarize(diff) %>%

  select(film, diff) %>%

  arrange(desc(diff)) %>%

  top_n(5)
```

Result:

```
  film              diff
  <chr>            <dbl>
1 Little Boy       -2.45
2 The Loft         -2.27
3 Taken 3          -2.17
4 Hitman: Agent 47 -2.05
5 The Longest Ride -2.00
```

```
  film                         diff
  <chr>                       <dbl>
1 Mr. Turner                   1.35
2 Timbuktu                     1.15
3 Phoenix                      1.15
4 The Diary of a Teenage Girl  1.06
5 It Follows                   1.02
```

D. Lahman Baseball Dataset

Some questions about home advantage

1. Create a statistic that quantifies "home advantage". You'll use this statistic for the next few questions. There is more than one reasonable choice here. Propose 2 different statistics and justify why you picked the one you'll use from now on.

```
lb$AdvW <- lb$HomeW - lb$AwayW
```

```
lb$AdvL <- lb$HomeL - lb$AwayL
```

I'd like to pick either AdvW or AdvL in order to contrast the players' performance when they're home and when they're away.

2. Find home advantage statistics for the American League (AL) and National League (NL) in the 2017-2019 period. Comment on the results. Do you see any differences between leagues? Do you see any evidence of home advantage at all? What are the years where there seems to be more of a home advantage, and those where the effect might not be as strong (or doesn't seem to be there)?

Code:

```
al = lb %>% filter(League == "AL" & Year <= 2019 & Year >=
2017) %>% select(Team, Year, AdvW)
```

```
nl = lb %>% filter(League == "NL" & Year <= 2019 & Year >=
2017) %>% select(Team, Year, AdvW)
```

```
lb %>% group_by(League) %>% summarize(HomeAdv = mean(AdvW, na.rm
= TRUE))%>% arrange(desc(HomeAdv))
```

```
lb %>% group_by(Year) %>% summarize(HomeAdv = mean(AdvW, na.rm =
TRUE)) %>% arrange(Year)
```

Result:

```
> lb %>% group_by(League) %>% summarize(HomeAdv = mean(AdvW, na.rm = TR
UE))%>% arrange(desc(HomeAdv))
# A tibble: 2 x 2
  League HomeAdv
  <fct>    <dbl>
1 NL        5.96
2 AL        4.53
> lb %>% group_by(Year) %>% summarize(HomeAdv = mean(AdvW, na.rm = TRU
E)) %>% arrange(Year)
# A tibble: 3 x 2
   Year HomeAdv
  <int>   <dbl>
1  2017    6.47
2  2018    4.5
3  2019    4.77
```

We can see from the tables above that there's a clear evidence of the existence of home advantage because each year and each league's average AdvW is positive. We can also see that NL teams tend to have a stronger home advantage compared to AL teams. As for years, home advantage effect seems to be stronger in 2017, weaker in 2018 and 2019.

3. Find the teams that had the highest and lowest home advantage effect by league in 2017, 2018, and 2019 separately. Comment on the results.

Code:

```
# the teams that has the highest home advantage:

# AL:

al %>% filter(Year == 2017) %>% group_by(Team) %>%

  summarize(alHomeAdv = mean(AdvW, na.rm = TRUE)) %>%
arrange(desc(alHomeAdv)) %>% top_n(1)

al %>% filter(Year == 2018) %>% group_by(Team) %>%

  summarize(alHomeAdv = mean(AdvW, na.rm = TRUE)) %>%
arrange(desc(alHomeAdv)) %>% top_n(1)

al %>% filter(Year == 2019) %>% group_by(Team) %>%

  summarize(alHomeAdv = mean(AdvW, na.rm = TRUE)) %>%
arrange(desc(alHomeAdv)) %>% top_n(1)


# NL:

nl %>% filter(Year == 2017) %>% group_by(Team) %>%

  summarize(nlHomeAdv = mean(AdvW, na.rm = TRUE)) %>%
arrange(desc(nlHomeAdv)) %>% top_n(1)

nl %>% filter(Year == 2018) %>% group_by(Team) %>%

  summarize(nlHomeAdv = mean(AdvW, na.rm = TRUE)) %>%
arrange(desc(nlHomeAdv)) %>% top_n(1)

nl %>% filter(Year == 2019) %>% group_by(Team) %>%

  summarize(nlHomeAdv = mean(AdvW, na.rm = TRUE)) %>%
arrange(desc(nlHomeAdv)) %>% top_n(1)


# the teams that has the lowest home advantage:

# AL:
```

```
al %>% filter(Year == 2017) %>% group_by(Team) %>%

  summarize(alHomeAdv = mean(AdvW, na.rm = TRUE)) %>%
arrange(desc(alHomeAdv)) %>% top_n(-1)

al %>% filter(Year == 2018) %>% group_by(Team) %>%

  summarize(alHomeAdv = mean(AdvW, na.rm = TRUE)) %>%
arrange(desc(alHomeAdv)) %>% top_n(-1)

al %>% filter(Year == 2019) %>% group_by(Team) %>%

  summarize(alHomeAdv = mean(AdvW, na.rm = TRUE)) %>%
arrange(desc(alHomeAdv)) %>% top_n(-1)


# NL:

nl %>% filter(Year == 2017) %>% group_by(Team) %>%

  summarize(nlHomeAdv = mean(AdvW, na.rm = TRUE)) %>%
arrange(desc(nlHomeAdv)) %>% top_n(-1)

nl %>% filter(Year == 2018) %>% group_by(Team) %>%

  summarize(nlHomeAdv = mean(AdvW, na.rm = TRUE)) %>%
arrange(desc(nlHomeAdv)) %>% top_n(-1)

nl %>% filter(Year == 2019) %>% group_by(Team) %>%

  summarize(nlHomeAdv = mean(AdvW, na.rm = TRUE)) %>%
arrange(desc(nlHomeAdv)) %>% top_n(-1)
```

Result:

```
> # the teams that has the highest home advantage:
> # AL:
> al %>% filter(Year == 2017) %>% group_by(Team) %>%
+   summarize(alHomeAdv = mean(AdvW, na.rm = TRUE)) %>% arrange(desc(al
HomeAdv)) %>% top_n(1)
Selecting by alHomeAdv
# A tibble: 2 x 2
  Team      alHomeAdv
  <fct>         <dbl>
1 Athletics        17
2 Orioles          17
> al %>% filter(Year == 2018) %>% group_by(Team) %>%
+   summarize(alHomeAdv = mean(AdvW, na.rm = TRUE)) %>% arrange(desc(al
HomeAdv)) %>% top_n(1)
Selecting by alHomeAdv
# A tibble: 1 x 2
  Team  alHomeAdv
  <fct>     <dbl>
1 Twins        20
> al %>% filter(Year == 2019) %>% group_by(Team) %>%
+   summarize(alHomeAdv = mean(AdvW, na.rm = TRUE)) %>% arrange(desc(al
HomeAdv)) %>% top_n(1)
Selecting by alHomeAdv
# A tibble: 1 x 2
```

```
  Team   alHomeAdv
  <fct>      <dbl>
1 Astros        13

> # NL:
> nl %>% filter(Year == 2017) %>% group_by(Team) %>%
+   summarize(nlHomeAdv = mean(AdvW, na.rm = TRUE)) %>% arrange(desc(nl
HomeAdv)) %>% top_n(1)
Selecting by nlHomeAdv
# A tibble: 1 x 2
  Team   nlHomeAdv
  <fct>      <dbl>
1 Padres        15
> nl %>% filter(Year == 2018) %>% group_by(Team) %>%
+   summarize(nlHomeAdv = mean(AdvW, na.rm = TRUE)) %>% arrange(desc(nl
HomeAdv)) %>% top_n(1)
Selecting by nlHomeAdv
# A tibble: 1 x 2
  Team     nlHomeAdv
  <fct>        <dbl>
1 Phillies        18
> nl %>% filter(Year == 2019) %>% group_by(Team) %>%
+   summarize(nlHomeAdv = mean(AdvW, na.rm = TRUE)) %>% arrange(desc(nl
HomeAdv)) %>% top_n(1)
Selecting by nlHomeAdv
# A tibble: 1 x 2
  Team  nlHomeAdv
  <fct>     <dbl>
1 Cubs         18
> # the teams that has the lowest home advantage:
> # AL:
> al %>% filter(Year == 2017) %>% group_by(Team) %>%
+   summarize(alHomeAdv = mean(AdvW, na.rm = TRUE)) %>% arrange(desc(al
HomeAdv)) %>% top_n(-1)
Selecting by alHomeAdv
# A tibble: 1 x 2
  Team   alHomeAdv
  <fct>      <dbl>
1 Astros        -5
> al %>% filter(Year == 2018) %>% group_by(Team) %>%
+   summarize(alHomeAdv = mean(AdvW, na.rm = TRUE)) %>% arrange(desc(al
HomeAdv)) %>% top_n(-1)
Selecting by alHomeAdv
# A tibble: 1 x 2
  Team   alHomeAdv
  <fct>      <dbl>
1 Astros       -11
> al %>% filter(Year == 2019) %>% group_by(Team) %>%
+   summarize(alHomeAdv = mean(AdvW, na.rm = TRUE)) %>% arrange(desc(al
HomeAdv)) %>% top_n(-1)
Selecting by alHomeAdv
# A tibble: 1 x 2
  Team  alHomeAdv
  <fct>     <dbl>
1 Twins        -9
> # NL:
> nl %>% filter(Year == 2017) %>% group_by(Team) %>%
+   summarize(nlHomeAdv = mean(AdvW, na.rm = TRUE)) %>% arrange(desc(nl
HomeAdv)) %>% top_n(-1)
Selecting by nlHomeAdv
# A tibble: 1 x 2
  Team      nlHomeAdv
  <fct>         <dbl>
1 Nationals        -3
```

```
> nl %>% filter(Year == 2018) %>% group_by(Team) %>%
+   summarize(nlHomeAdv = mean(AdvW, na.rm = TRUE)) %>% arrange(desc(nl
HomeAdv)) %>% top_n(-1)
Selecting by nlHomeAdv
# A tibble: 2 x 2
  Team   nlHomeAdv
  <fct>      <dbl>
1 Braves        -4
2 Padres        -4
> nl %>% filter(Year == 2019) %>% group_by(Team) %>%
+   summarize(nlHomeAdv = mean(AdvW, na.rm = TRUE)) %>% arrange(desc(nl
HomeAdv)) %>% top_n(-1)
Selecting by nlHomeAdv
# A tibble: 1 x 2
  Team   nlHomeAdv
  <fct>      <dbl>
1 Giants        -7
```

What we can see is that home advantage effect is very unstable. The teams who has the largest home advantage in this year might be the same team who has the worst home advantage effect, just like team Astros, Twins, and Padres.

4. Which franchise had the highest average home advantage in the 2017-2019 period? Which one had the lowest average home advantage effect?

Code:

```
lb %>% group_by(Team) %>% summarize(HomeAdv = mean(AdvW, na.rm =
TRUE)) %>% arrange(desc(HomeAdv)) %>% top_n(1)

lb %>% group_by(Team) %>% summarize(HomeAdv = mean(AdvW, na.rm =
TRUE)) %>% arrange(desc(HomeAdv)) %>% top_n(-1)
```

Result:

```
> lb %>% group_by(Team) %>% summarize(HomeAdv = mean(AdvW, na.rm = TRU
E)) %>% arrange(desc(HomeAdv)) %>% top_n(1)
Selecting by HomeAdv
# A tibble: 1 x 2
  Team     HomeAdv
  <fct>      <dbl>
1 Phillies      13
> lb %>% group_by(Team) %>% summarize(HomeAdv = mean(AdvW, na.rm = TRU
E)) %>% arrange(desc(HomeAdv)) %>% top_n(-1)
Selecting by HomeAdv
# A tibble: 1 x 2
  Team   HomeAdv
  <fct>    <dbl>
1 Astros      -1
```

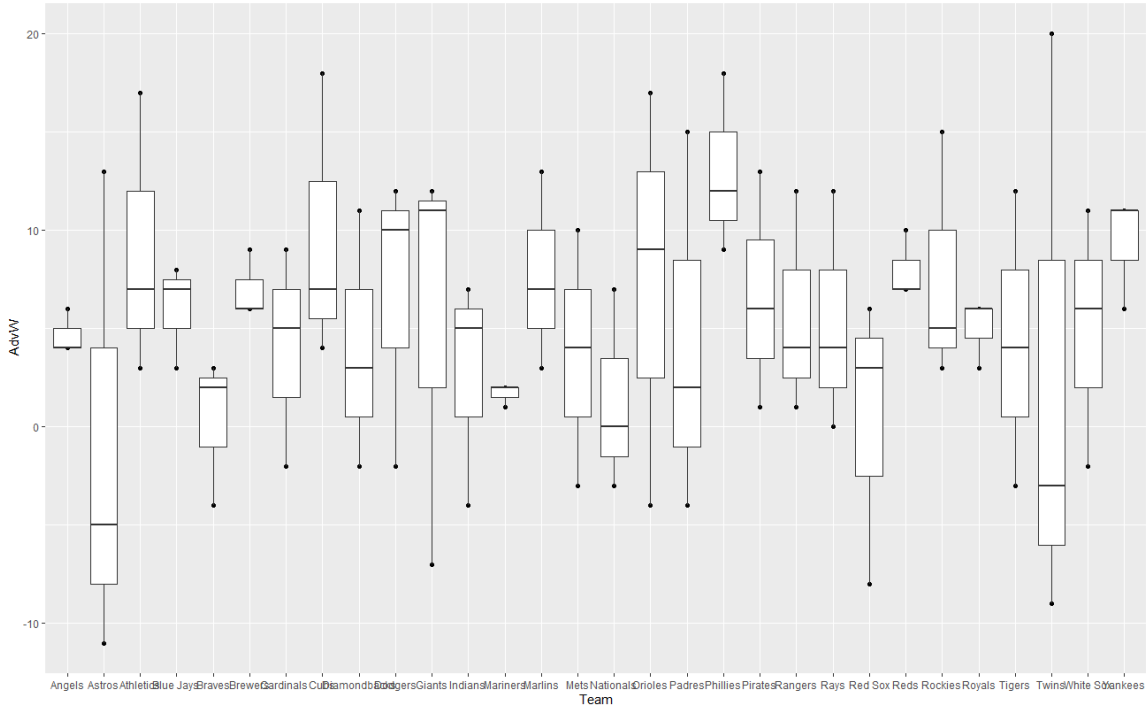
5. After completing these exercises, what did you learn about home advantage effect in the MLB? You're welcome to try out a few new queries to illustrate your points.

I feel like there's a correlation between a team's stability and its average home advantage effect. To prove whether it's true or not:

Code:

```
qplot(x = Team, y = AdvW, data = lb) + geom_boxplot()
```

Result:



We can see that when a team has a large range of home advantage effect, this team's average home advantage score tends to be negative or low. For example, team Astros and Twins' average home advantage effect are negative and their range are the top 2 largest.

Aging in pitchers and batters

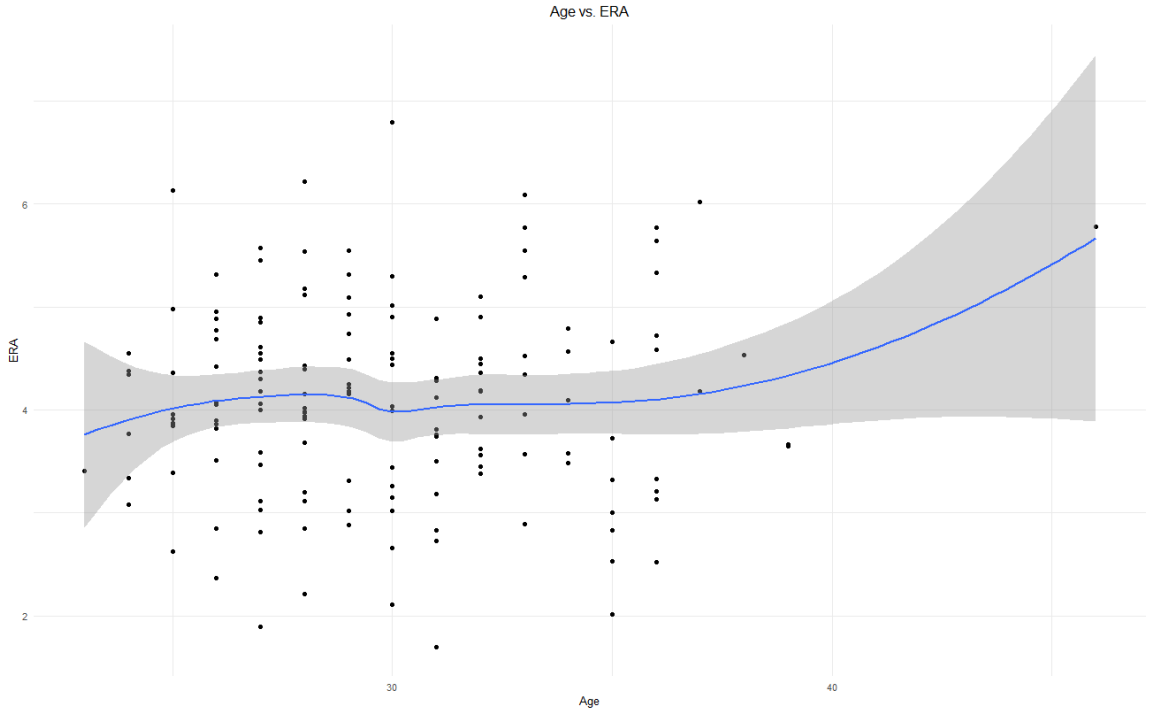
1. Let's consider data from 2018 only and look at the subset of pitchers who pitched more than 250 outs. Plot the earned run average (ERA; small values are good and big ones are bad) of the pitchers against their age. Do you see any patterns? Now, find a table with the average ERAs by age. Do you see any patterns?

Code:

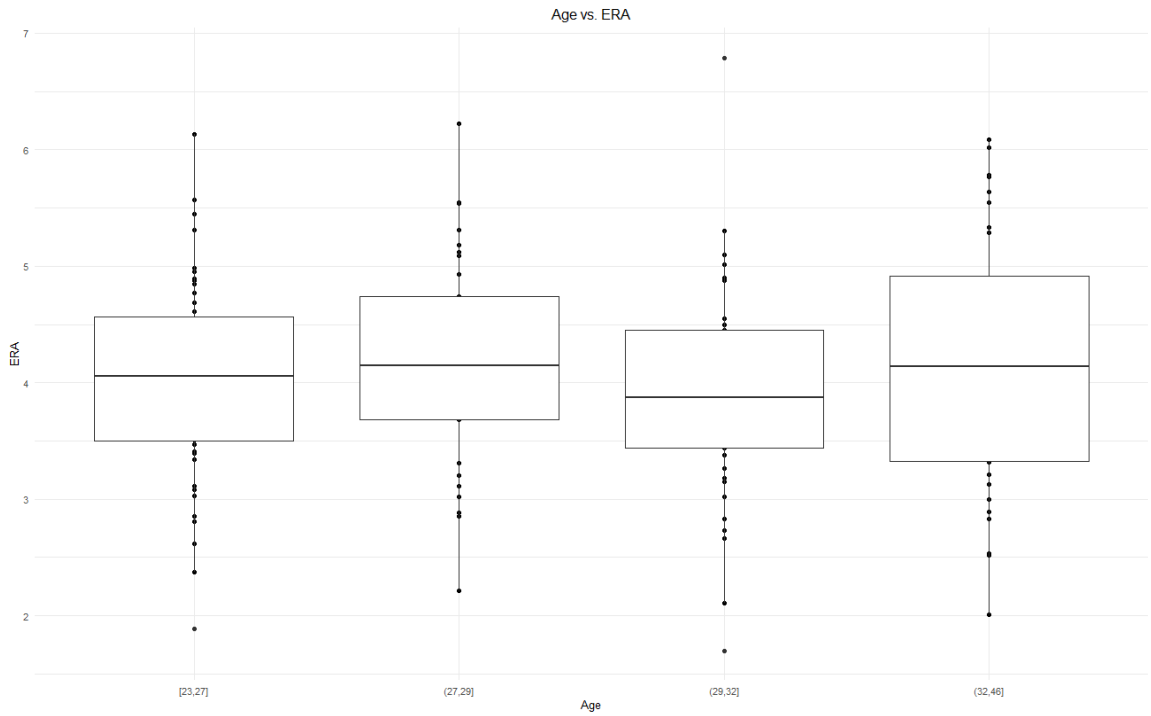
```
data(Pitching)
pitching <- Pitching %>%
  filter(IPouts > 250 & yearID == 2018) %>%
  left_join(People, by = "playerID") %>%
  select(nameFirst, nameLast, birthYear, ERA)

pitching$Age = 2019 - pitching$birthYear
```

```
qplot(x = Age, y = ERA, data = pitching) +
  geom_point() +
  geom_smooth() +
  ggtitle("Age vs. ERA") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(text=element_text(size=10))
# table
pitching %>% group_by(Age) %>% summarise(avgERA = mean(ERA, na.rm
= TRUE))


# another plot
# categorize age variable
pitching$Age = cut(pitching$Age,
                        breaks=quantile(pitching$Age),
                        include.lowest = TRUE)


pitching %>% group_by(Age) %>% summarise(avgERA = mean(ERA, na.rm
= TRUE))
qplot(x = Age, y = ERA, data = pitching) +
  geom_boxplot() +
  ggtitle("Age vs. ERA") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(text=element_text(size=10))
```

Result:




```
> # table
> pitching %>% group_by(Age) %>% summarise(avgERA = mean(ERA, na.rm = T
RUE))
# A tibble: 18 x 2
     Age avgERA
   <dbl>  <dbl>
 1    23   3.41
 2    24   3.91
 3    25   4.12
 4    26   4.10
 5    27   4.07
 6    28   4.05
 7    29   4.31
 8    30   4.08
 9    31   3.61
10    32   4.14
11    33   4.66
12    34   4.10
13    35   3.15
14    36   4.25
15    37   5.1
16    38   4.53
17    39   3.66
18    46   5.78

> # categorize age variable
> pitching %>% group_by(Age) %>% summarise(avgERA = mean(ERA, na.rm = T
RUE))
# A tibble: 4 x 2
  Age     avgERA
  <fct>    <dbl>
1 [23,27]   4.05
2 (27,29]   4.16
3 (29,32]   3.93
4 (32,46]   4.18
```



From the line plot we could see that there seems to be a correlation between age and ERA, with age going up, ERA goes up. However, if we categorize the age and see the tables and the boxplot, there's no clear relationship between age and ERA.

2. Again, let's look at pitchers who pitched more than 250 outs in 2018. Identify the top 5 best and worst pitchers, in terms of ERA.

Code:

```
pitching <- Pitching %>%
  filter(IPouts > 250 & yearID == 2018) %>%
  left_join(People, by = "playerID") %>%
  select(playerID, nameFirst, nameLast, ERA)


# top 5 best pitchers
pitching %>% group_by(playerID) %>%
  summarise(avgERA = mean(ERA, na.rm = TRUE)) %>%
arrange(avgERA) %>%
  top_n(-5) %>% left_join(People, by = "playerID") %>%
  select(nameFirst, nameLast, avgERA)
```

```
# top 5 worst pitchers

pitching %>% group_by(playerID) %>%

  summarise(avgERA = mean(ERA, na.rm = TRUE)) %>%
arrange(desc(avgERA)) %>%

  top_n(5) %>% left_join(People, by = "playerID") %>%

  select(nameFirst, nameLast, avgERA)
```

Result:

```
> # top 5 best pitchers
  nameFirst nameLast avgERA
  <chr>     <chr>     <dbl>
1 Jacob     deGrom     1.7
2 Blake     Snell      1.89
3 Clay      Buchholz   2.01
4 Chris     Sale       2.11
5 Trevor    Bauer      2.21

> # top 5 worst pitchers
  nameFirst nameLast avgERA
  <chr>     <chr>     <dbl>
1 Matt      Moore      6.79
2 Martin    Perez      6.22
3 Lucas     Giolito    6.13
4 Homer     Bailey     6.09
5 Jason     Hammel     6.02
```

3. Consider the best pitcher (in terms of ERA) that you found in part 2. Find his ERA by season throughout his career. Based on this alone, do you think he's already "peaked"? If you like baseball, you're welcome to share your opinion here as well.

Code:

```
pitching %>% group_by(playerID) %>%

  summarise(avgERA = mean(ERA, na.rm = TRUE)) %>%

  arrange(avgERA) %>%

  top_n(-1) %>% select(playerID) %>%

  inner_join(Pitching, by = "playerID") %>%

  select(playerID, yearID, ERA) %>%

  left_join(People, by = "playerID") %>%

  select(playerID, nameFirst, nameLast, yearID, ERA)
```

Result:

```
  playerID  nameFirst nameLast yearID   ERA
  <chr>     <chr>     <chr>     <int> <dbl>
1 degroja01 Jacob     deGrom     2014  2.69
2 degroja01 Jacob     deGrom     2015  2.54
3 degroja01 Jacob     deGrom     2016  3.04
```

```
4 degroja01 Jacob     deGrom     2017  3.53
5 degroja01 Jacob     deGrom     2018  1.7
```

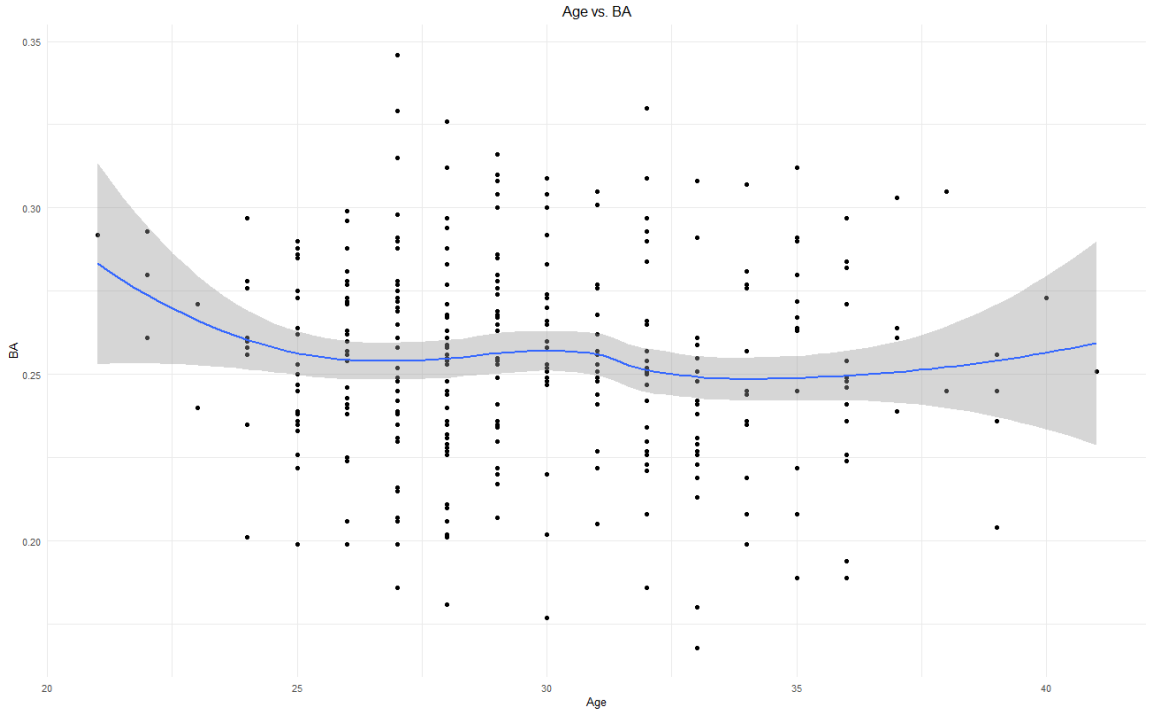
According to the data we have, we can say that he's peaked, only considering ERA.

4. Let's do a similar exercise, but now with batting average (BA; more is better). Use the battingStats function in Lahman to find BAs. Consider data from 2018 only and look at players that have more than 200 at bats (AB). Plot BA against age. Do you see any patterns? Find a table with average BAs by age. Explain what you see.

Code:

```
data(Batting)


batting <- Batting %>%
  filter(AB > 200 & yearID == 2018) %>%
  left_join(People, by = "playerID")


BAs = battingStats(data = batting,
             idvars = c("playerID", "yearID", "stint", "teamID",
"lgID"),
             cbind = TRUE)
BAs$Age = 2019 - batting$birthYear


qplot(x = Age, y = BA, data = BAs) +
  geom_point() +
  geom_smooth() +
  ggtitle("Age vs. BA") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(text=element_text(size=10))


# table
BAs %>% group_by(Age) %>% summarise(avgBA = mean(BA, na.rm =
TRUE))
```

Result:




```
   Age avgBA
   <db1> <db1>
 1    21 0.292
 2    22 0.278
 3    23 0.256
 4    24 0.260
 5    25 0.254
 6    26 0.255
 7    27 0.257
 8    28 0.249
 9    29 0.260
10    30 0.259
# ... with 11 more rows
```

Looks like there's no clear relationship between age and BA when age is above 25; there's a negative correlation between age and BA when age is under 25, that is to say, when age goes up, BA tends to go down before a player reach 25.

5. Again, let's look at players with more than 200 ABs in 2018. Find the top 5 best and worst players in terms of BA.

Code:

```
# top 5 best players

BAs %>% group_by(playerID) %>%

  summarise(avgBA = mean(BA, na.rm = TRUE)) %>%
arrange(desc(avgBA)) %>%
```

```
  top_n(5) %>% left_join(People, by = "playerID") %>%
  select(nameFirst, nameLast, avgBA)


# top 5 worst players
BAs %>% group_by(playerID) %>%
  summarise(avgBA = mean(BA, na.rm = TRUE)) %>%
arrange(avgBA) %>%
  top_n(-5) %>% left_join(People, by = "playerID") %>%
  select(nameFirst, nameLast, avgBA)
```

Result:

```
> # top 5 best players
  nameFirst nameLast avgBA
  <chr>     <chr>    <dbl>
1 Mookie    Betts    0.346
2 J. D.     Martinez 0.33
3 Jeff      McNeil   0.329
4 Christian Yelich   0.326
5 Jose      Altuve   0.316


> # top 5 worst players
  nameFirst nameLast avgBA
  <chr>     <chr>    <dbl>
1 Chris     Davis    0.168
2 Sandy     Leon     0.177
3 Dexter    Fowler   0.18
4 Aaron     Altherr  0.181
5 Logan     Morrison 0.186
6 Gary      Sanchez  0.186
```

6. Consider the best player (in terms of BA) that you found in part 5. Find his BA by season throughout his career. Based on this alone, do you think he's already "peaked"? If you like baseball, you're welcome to share your opinion here as well.

Code:

```
BattingAllYear = battingStats(data = Batting,
              idvars = c("playerID", "yearID", "stint", "teamID",
"lgID"),
              cbind = TRUE)


BAs %>% group_by(playerID) %>%
  summarise(avgBA = mean(BA, na.rm = TRUE)) %>%
  arrange(avgBA) %>%
```

```
  top_n(1) %>% select(playerID) %>%
  inner_join(BattingAllYear, by = "playerID") %>%
  select(playerID, yearID, BA) %>%
  left_join(People, by = "playerID") %>%
  select(playerID, nameFirst, nameLast, yearID, BA)
```

Result:

```
  playerID  nameFirst nameLast yearID    BA
  <chr>     <chr>     <chr>     <int> <dbl>
1 bettsmo01 Mookie    Betts      2014 0.291
2 bettsmo01 Mookie    Betts      2015 0.291
3 bettsmo01 Mookie    Betts      2016 0.318
4 bettsmo01 Mookie    Betts      2017 0.264
5 bettsmo01 Mookie    Betts      2018 0.346
```

I think he's already peaked if we only look at the BA scores.