Lab 1

Jackson Janes

Due on 02/03 at 11:59 pm

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(broom)
library(Lahman)
library(retrosheet)
```

```
##
## For Retrosheet data obtained with this package:
##
## The information used here was obtained free of charge from
## and is copyrighted by Retrosheet. Interested parties may
## contact Retrosheet at "www.retrosheet.org"
```

```
library(ggplot2)
```

Instructions: This lab report needs to be professional. Only report relevant and finalized code. Your writing should be concise and void of spelling errors. Use code chunk options to hide unnecessary messages/warnings. Your report should be reproducible. Reports that involve simulations need to have the random seed specified so that simulation results are reproducible. You are allowed to work on this lab assignment in groups of 2-3. You still need to submit an individual lab report if you do work in a group, and you need to list your collaborators.

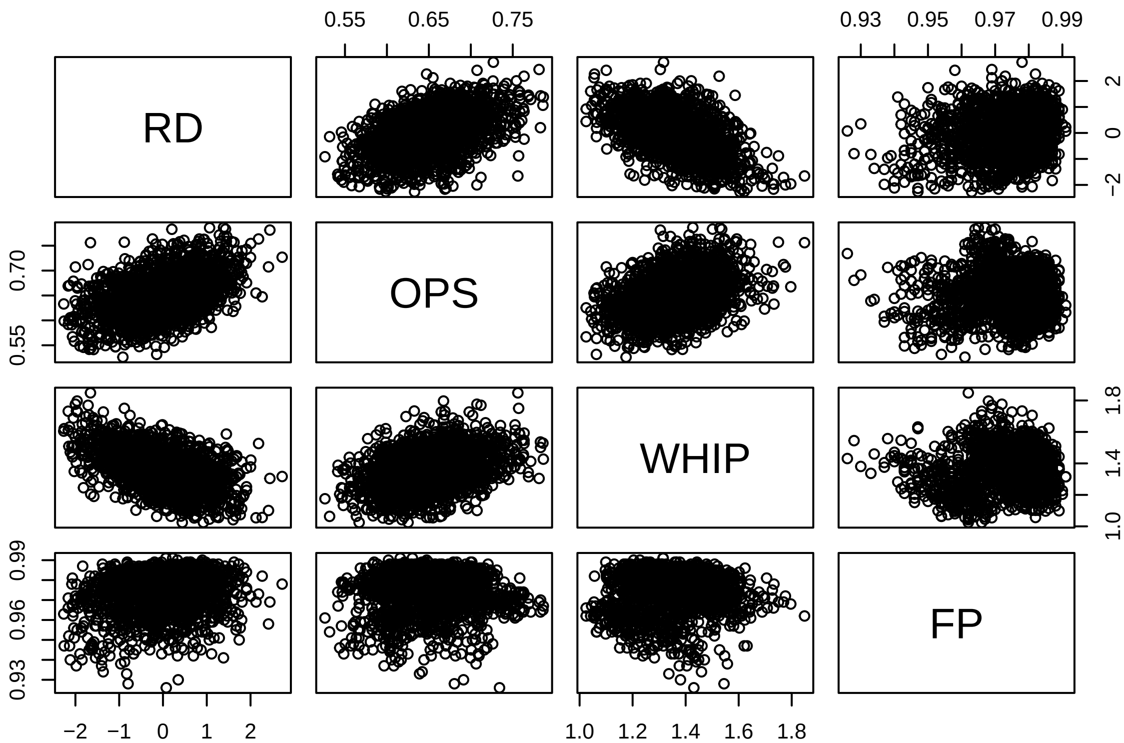
Question 1 In lecture it was demonstrated that baseball is a game of offense, pitching, and defense with a regression model that considered expected run differential as a function of explanatory variables OPS, WHIP, and FP. Do the following:

- Fit a similar regression model with runs as the response variable. Report problems with this model. Investigate problematic residuals to discover what went wrong. Fix the problem with this model by adding categorical variable(s) to the list of explanatory variables. Briefly explain what went wrong.

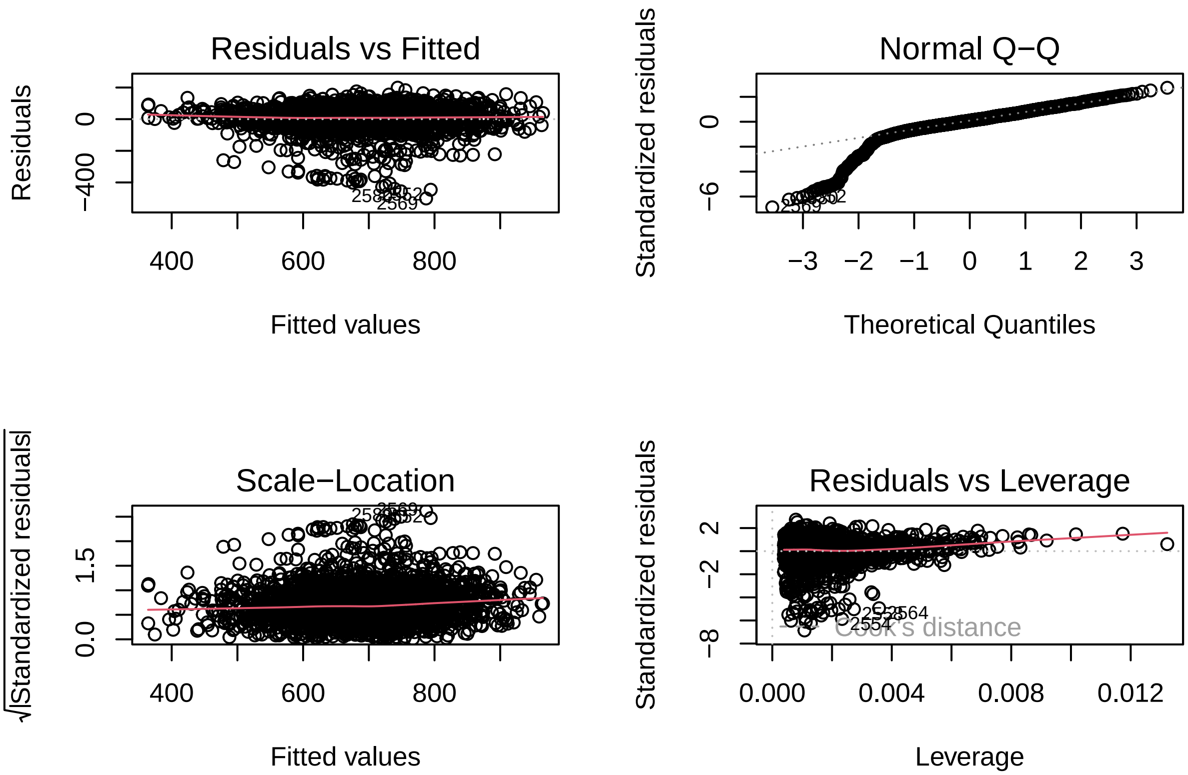
```r
#OPS calculation
#OBP = (H + BB + HBP) / (AB + BB + HBP + SF)
#SLG = ((X1B + 2*X2B + 3*X3B + 4*HR) / AB)

#OPS = OBP + SLG

#WHIP calculation
#WHIP = 3* (HA + BBA/IPouts)

#FP calculation

dat <- Teams %>%
  dplyr::select(yearID, franchID, W, L, AB, H, X2B, X3B, HR, BB, HBP, SF, HA, HRA, BBA, SOA, IPouts, FP
  filter(yearID >= 1900) %>%
  replace_na(list(HBP = 0, SF = 0)) %>%
  mutate(X1B = H - (X2B - X3B - HR)) %>%
  mutate(RD = (R - RA) / (W + L), X1B = H - (X2B + X3B + HR)) %>%
  mutate(OBP = (H + BB + HBP) / (AB + BB + HBP + SF)) %>%
  mutate(SLG = (X1B + 2*X2B + 3*X3B + HR) / AB) %>%
  mutate(OPS = OBP + SLG) %>%
  mutate(WHIP = 3*(HA + BBA) / IPouts)

question1 <- lm(R ~ OPS + WHIP + FP, data = dat)

summary(question1)
```

```
##
## Call:
## lm(formula = R ~ OPS + WHIP + FP, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -501.20  -27.50    6.29   40.79  199.08
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3876.770    153.558 -25.246   <2e-16 ***
## OPS          2194.394     39.325  55.801   <2e-16 ***
## WHIP           -7.735     13.433  -0.576    0.565
## FP           3231.696    154.993  20.851   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 73.05 on 2606 degrees of freedom
## Multiple R-squared:  0.6006, Adjusted R-squared:  0.6001
## F-statistic:  1306 on 3 and 2606 DF,  p-value: < 2.2e-16
```

```r
pairs(dat %>% select(RD, OPS, WHIP, FP))
```




```r
par(mfrow = c(2,2))
plot(question1)
```




```
dat_aug <- augment(question1, data = dat)
dat_aug %>%
  mutate(rmse = sqrt((mean(.resid^2)))) %>%
  summarize(N = n(),
            within_1rmse = sum(abs(.resid) < rmse),
            within_2rmse = sum(abs(.resid) < 2 * rmse)) %>%
  mutate(within_1rmse_pct = within_1rmse / N,
         within_2rmse_pct = within_2rmse / N)
```

```
## # A tibble: 1 x 5
##       N within_1rmse within_2rmse within_1rmse_pct within_2rmse_pct
##   <int>        <int>        <int>            <dbl>            <dbl>
## 1  2610         2106         2507            0.807            0.961
```

```
m_glm <- glm(RD ~ OPS + WHIP + FP, data = dat)
pchisq(m_glm$deviance, m_glm$df.residual, lower = FALSE)
```

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

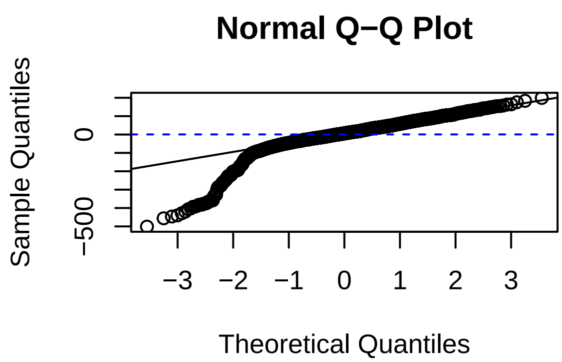
```
dat_aug %>% filter(abs(.resid) >= 1) %>%
  select(yearID, franchID, R, OPS, WHIP, FP, .resid, .fitted) %>%
  mutate(across(3:8, round, 3)) %>%
  arrange(desc(.resid))
```

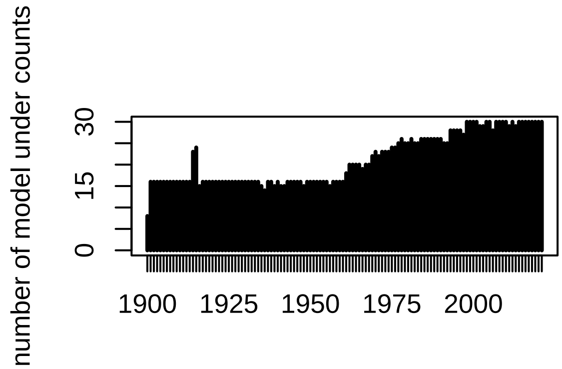
```
## # A tibble: 2,553 x 8
##    yearID franchID     R   OPS  WHIP    FP .resid .fitted
##     <int> <fct>    <dbl> <dbl> <dbl> <dbl>  <dbl>   <dbl>
##  1   2019 NYY        943 0.664  1.30 0.982   199.    744.
##  2   2019 MIN        939 0.671  1.30 0.981   183.    756.
##  3   2021 TBD        857 0.629  1.17 0.986   176.    681.
##  4   2000 OAK        947 0.688  1.50 0.978   164.    783.
##  5   2018 NYY        851 0.635  1.24 0.984   163.    688.
##  6   1931 NYY       1067 0.754  1.42 0.972   158.    909.
##  7   2019 LAD        886 0.657  1.10 0.982   155.    731.
##  8   2017 TEX        799 0.619  1.40 0.982   155.    644.
##  9   2021 LAD        830 0.628  1.10 0.985   153.    677.
## 10   1996 BAL        949 0.687  1.50 0.984   151.    798.
## # ... with 2,543 more rows
```

```
dat_aug %>% filter(.fitted >= 2) %>%
  select(yearID, franchID, R, OPS, WHIP, FP, .resid, .fitted)
```

```
## # A tibble: 2,610 x 8
##    yearID franchID     R   OPS  WHIP    FP .resid .fitted
##     <int> <fct>    <int> <dbl> <dbl> <dbl>  <dbl>   <dbl>
##  1   1900 LAD        816 0.726  1.45 0.948   47.0    769.
##  2   1900 ATL        778 0.686  1.39 0.953   81.5    697.
##  3   1900 CHC        635 0.639  1.34 0.933  105.     530.
##  4   1900 CIN        703 0.652  1.40 0.945  107.     596.
##  5   1900 SFG        713 0.680  1.54 0.928  110.     603.
##  6   1900 PHI        810 0.716  1.53 0.945   73.0    737.
##  7   1900 PIT        733 0.679  1.24 0.945   75.1    658.
##  8   1900 STL        744 0.709  1.37 0.943   28.5    716.
##  9   1901 NYY        760 0.734  1.43 0.926   44.1    716.
## 10   1901 BOS        759 0.688  1.21 0.943   86.8    672.
## # ... with 2,600 more rows
```

```
qqnorm(resid(question1)); qqline(resid(question1))
abline(a=0.5, b=0, lty =2, col = "blue")

plot(table(dat_aug %>% filter(abs(.resid) >= 0.5) %>%
             pull(yearID)), ylab = "number of model under counts")
```





- We can significantly improve the regression model in the notes through a principled rescaling of OPS, WHIP, and FP. Split the Teams data frame by {yearID} and, for each year, create variables {OPSscale = OPS/avgOPS}, {WHIPscale = avgWHIP/WHIP}, and {FPscale = avgFP/FP} which require you to first create league average variables {avgOPS}, {avgWHIP}, and {avgFP}. Fit the linear regression model with runs differential as the response and explanatory variables {OPSscale}, {WHIPscale}, and {FPscale}, and report relevant output. Why does this model perform so much better than the model in the notes? Support your answer. Hint: functions {split}, {do.call}, and {lapply} are useful.

```
q1b <- lm(RD ~ OPSscale + WHIPscale + FPscale, data = newmodel)

summary(q1b)
```

```
##
## Call:
## lm(formula = RD ~ OPSscale + WHIPscale + FPscale, data = newmodel)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.65929 -0.27836  0.00616  0.28471  1.57567
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  25.3774     2.6208   9.683   <2e-16 ***
## OPSscale      5.5841     0.1559  35.815   <2e-16 ***
## WHIPscale     7.1250     0.1398  50.953   <2e-16 ***
```

```
## FPscale     -37.6021     2.5403 -14.803   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4238 on 2606 degrees of freedom
## Multiple R-squared:   0.69,  Adjusted R-squared:  0.6897
## F-statistic:  1934 on 3 and 2606 DF,  p-value: < 2.2e-16
```

This model performs much better because it focuses on each season as a whole, which makes the outliers less impactful to the model.

Question 2 Choose 3 batters and 3 pitchers that have played in at least 10 seasons and do the following:

- Display the seasonal statistics for these players. The following statistics should be included for batters (derivations of unconventional statistics are in parentheses): year, G, AB, R, H, X2B, X3B, HR, RBI, SB, CS, SBpct (SB / (SB + CS)), BB, SO, OBP, SLG, OPS. The following statistics should be included for pitchers: year, W, L, IPouts, H, ER, HR, BB, HBP, SO, ERA, WHIP, SOper9 (SO / IP * 9), SOperBB (SO / BB). These statistics can be found in or computed from statistics that are found in the `Batting` and `Pitching` dataframes in the `Lahman` package.

Alfonso Soriano

Dexter Fowler

Jason Heyward

Kerry Wood

Carlos Zambrano

Jeff Samardzija

- Create career stat lines for each of the players that you selected. Be careful about how these statistics are calculated.

```
# Combined career batting stats for Alfonso Soriano, Dexter Fowler, and Jason Heyward
Batter_Career <- bind_rows(also, defo, jahe); Batter_Career
```

```
##              Name CarYrs CarG CarAB CarR CarH CarX2B CarX3B CarHR CarRBI CarSB
## 1 Alfonso Soriano     16 1975  7750 1152 2095    481     31   412   1159   289
## 2   Dexter Fowler     14 1460  5040  817 1306    253     82   127    517   149
## 3   Jason Heyward     12 1531  5390  766 1394    264     38   158    631   117
##   CarCS   CarSBpct CarBB CarSO    CarOBP    CarSLG    CarOPS
## 1    84 0.7747989   496  1803 0.3192225 0.4998710 0.8190935
## 2    68 0.6866359   740  1326 0.3579109 0.4174603 0.7753712
## 3    40 0.7452229   633  1069 0.3408197 0.4096475 0.7504672
```

```
# Combined career pitching stats for Kerry Wood, Carlos Zambrano, and Jeff Samardzija
Pitcher_Career <<- bind_rows(kewo, caza, jesa); Pitcher_Career
```

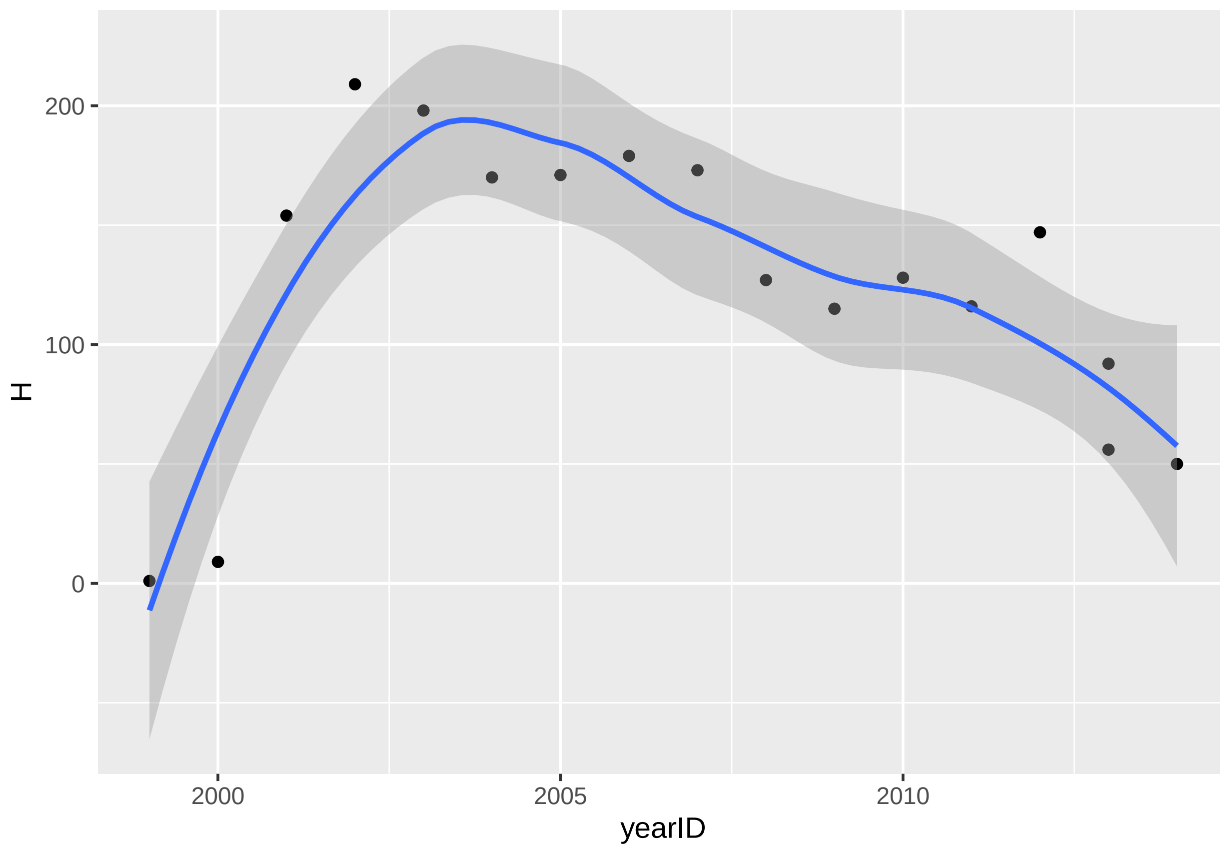
```
##              Name CarYrs CarW CarL CarIPouts CarH CarER CarHR CarBB CarHBP
## 1      Kerry Wood     14   86   75      4140 1083   563   148   666     99
## 2 Carlos Zambrano     12  132   91      5877 1709   797   161   898    102
## 3 Jeff Samardzija     13   80  106      4936 1555   759   205   491     60
##   CarSO   CarERA  CarWHIP CarSOper9 CarSOperBB
```

```
## 1  1582 3.671739 1.267391 10.317391   2.375375
## 2  1637 3.661562 1.330781  7.520674   1.822940
## 3  1449 4.151742 1.243517  7.926053   2.951120
```

- Provide a plot for career trajectories for one batting and one pitching statistic of your choice. These are two separate graphics, one for the batters and one for the pitchers. The graphics that you produce should display the trajectories of the 3 batters and the 3 pitchers. Provide interesting commentary on your graphic.

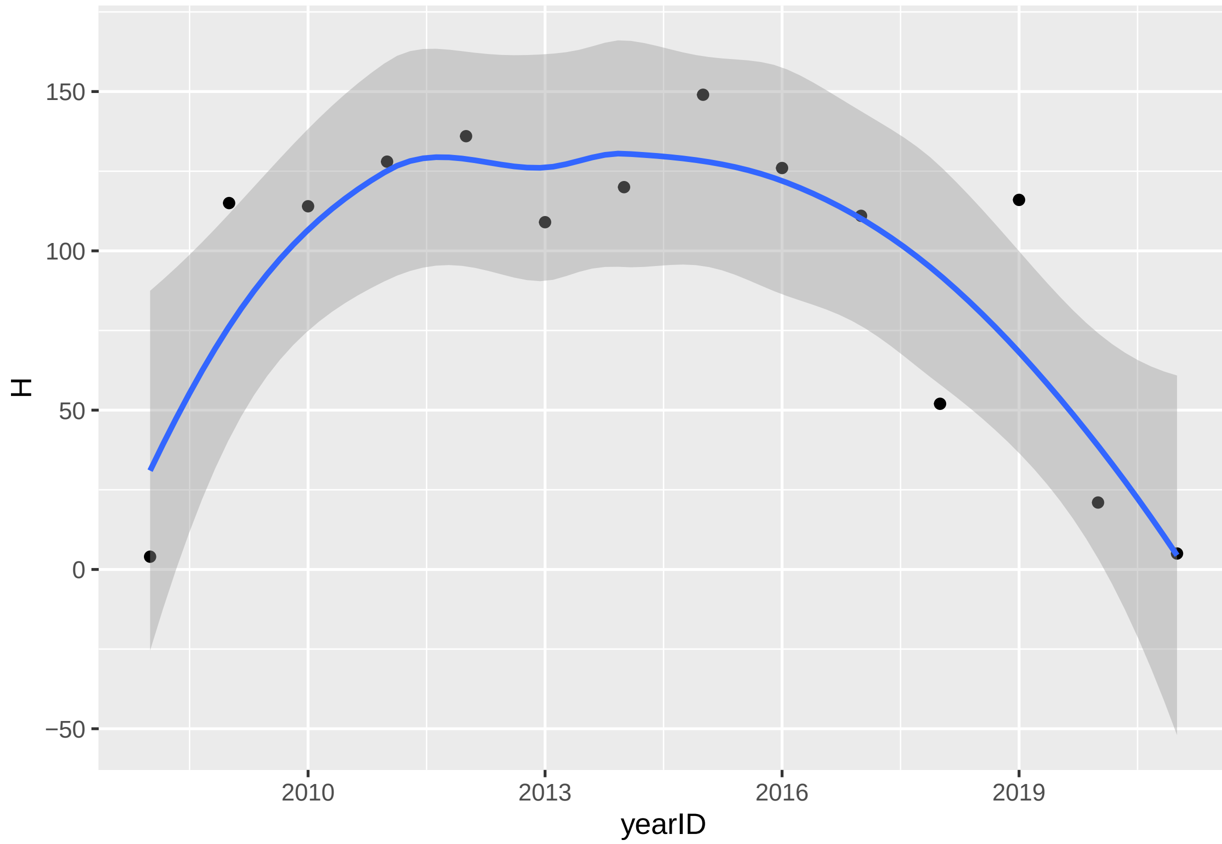
```
#Batting Statistic: Hits

ggplot(data = soriano, mapping = aes(x = yearID, y = H)) + geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



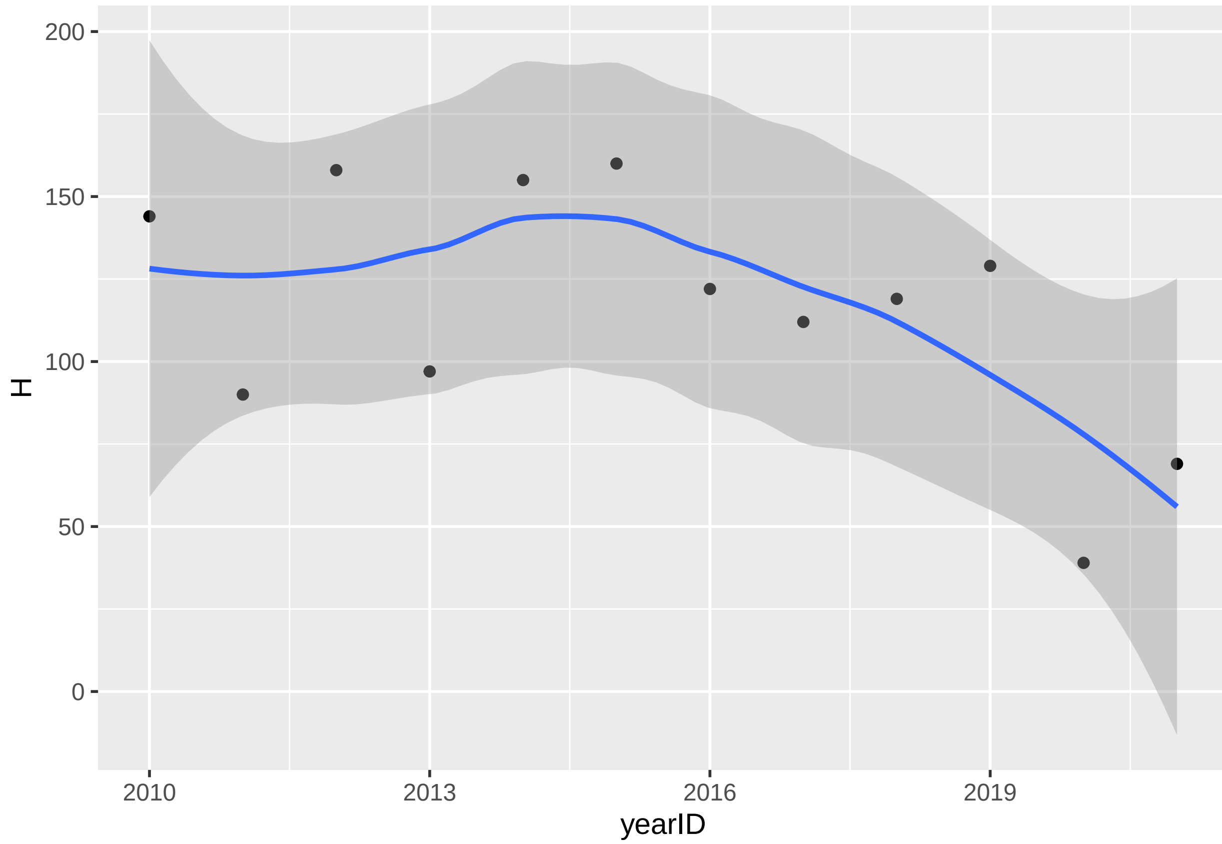

```
ggplot(data = fowler, mapping = aes(x = yearID, y = H)) + geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



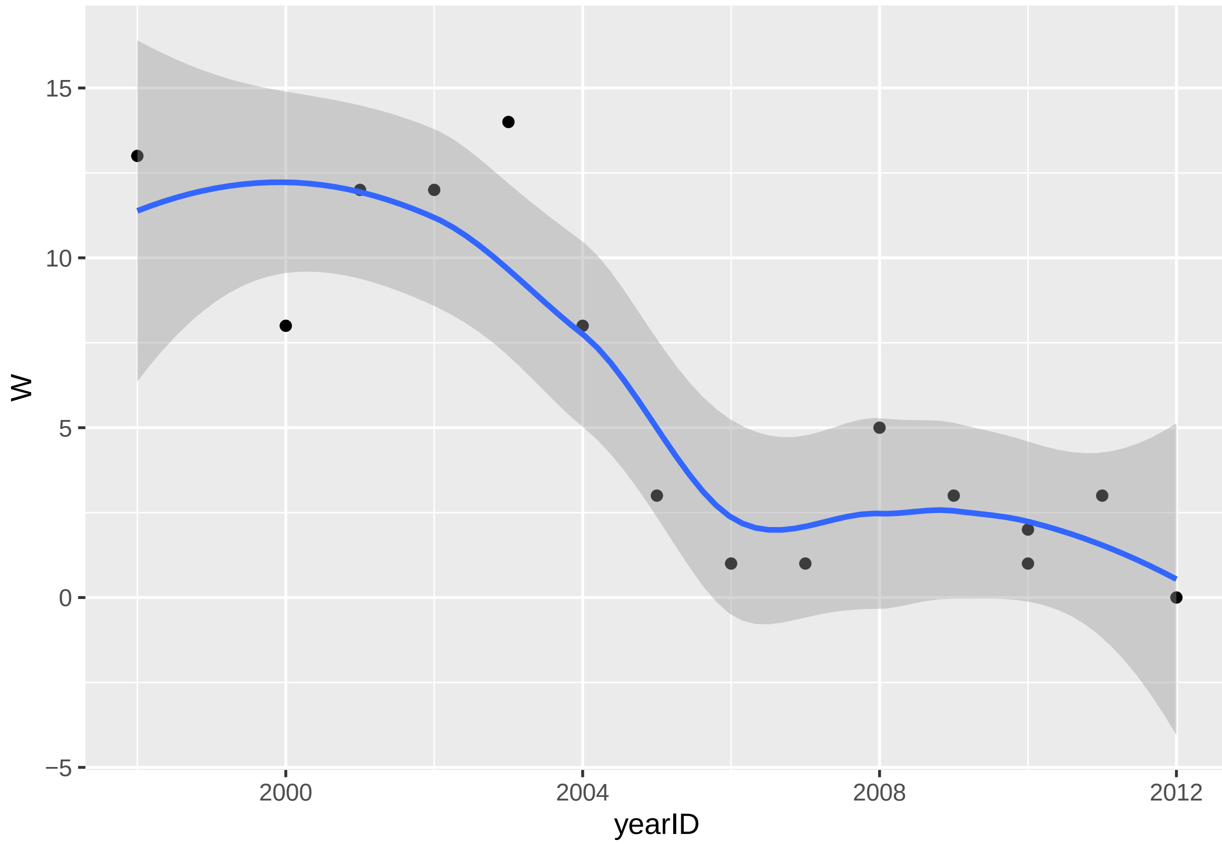

```
ggplot(data = heyward, mapping = aes(x = yearID, y = H)) + geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



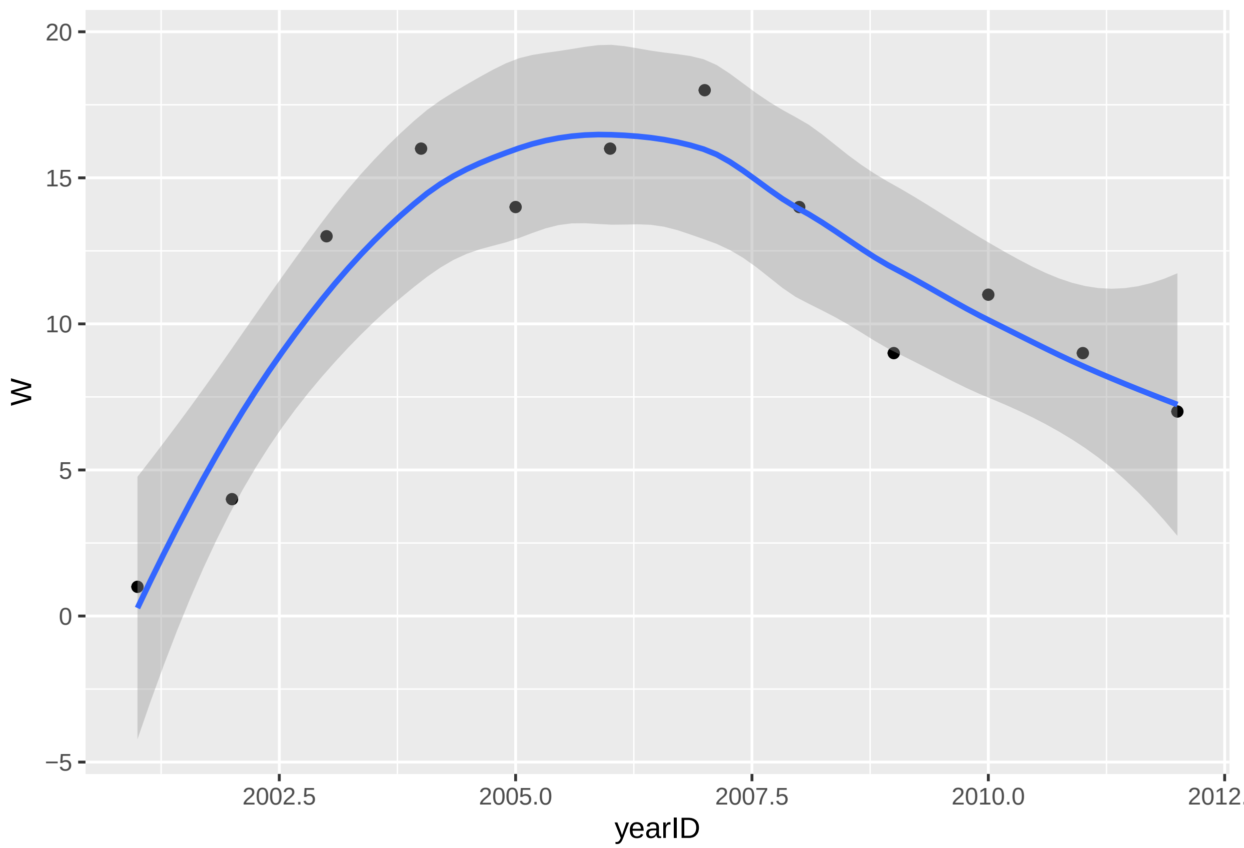

```
#Pitching Statistic: Wins

ggplot(data = wood, mapping = aes(x = yearID, y = W)) + geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



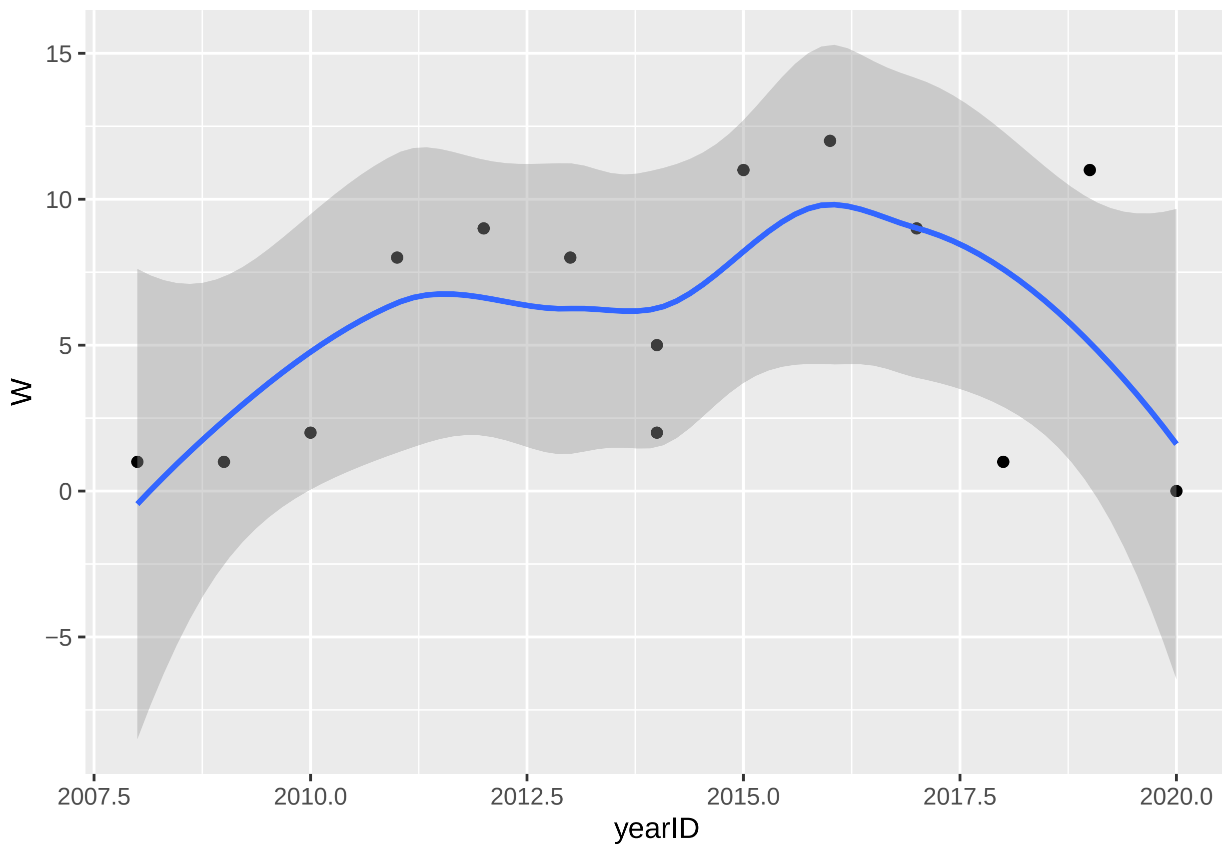

```
ggplot(data = zambrano, mapping = aes(x = yearID, y = W)) + geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
ggplot(data = samardzija, mapping = aes(x = yearID, y = W)) + geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



All three hitters seemed to be in their "prime" in roughly the middle of their careers. There is a clear decline from this point on in their careers, and the dropoff in hits is pretty dramatic for all three players.

The plots of Carlos Zambrano and Jeff Samardzija are roughly what I would expect a career trajectory to look like: They struggled at first, built up a few really solid years in the middle of their careers, and then declined in their final years in the MLB. Kerry Wood's plot is rather interesting, as he had his best years very early and then experienced a fairly steep decline after that point. I did not expect that dropoff to be as dramatic as depicted above, but it is interesting how consistent and how little variation that was later in his career.

Question 3 Problem 2 on page 28 of Analyzing Baseball Data with R

(a) Gibson started 34 games for the Cardinals in 1968. What fraction of these games were completed by Gibson?

```
q3a <- Pitching %>%
 select(playerID, yearID, teamID, G, CG) %>%
  filter(playerID == "gibsobo01") %>%
  filter(yearID == "1968") %>%
  mutate(CGpct = CG / G) %>%
  select(CGpct); q3a
```

```
##       CGpct
## 1 0.8235294
```

(b) What was Gibson's ratio of strikeouts to walks this season?

```r
q3b <- Pitching %>%
  select(playerID, yearID, SO, BB) %>%
  filter(playerID == "gibsobo01") %>%
  filter(yearID == "1968") %>%
  mutate(KBBratio = SO/ BB) %>%
  select(KBBratio); q3b
```

```
##   KBBratio
## 1 4.322581
```

(c) One can compute Gibson's innings pitched by dividing IPouts by three. How many innings did Gibson pitch this season?

```r
q3c <- Pitching %>%
  select(playerID, yearID, IPouts) %>%
  filter(playerID == "gibsobo01") %>%
  filter(yearID == "1968") %>%
  mutate(IP = IPouts / 3) %>%
  select(IP); q3c
```

```
##         IP
## 1 304.6667
```

(d) A modern measure of pitching effectiveness is WHIP, the average number of hits and walks allowed per inning. What was Gibson's WHIP for the 1968 season?

```r
q3d <- Pitching %>%
  select(playerID, yearID, H, BB, IPouts) %>%
  filter(playerID == "gibsobo01") %>%
  filter(yearID == "1968") %>%
  mutate(WHIP = 3*(H + BB) / IPouts) %>%
  select(WHIP); q3d
```

```
##        WHIP
## 1 0.8533917
```

Question 4 Problem 3 on page 29 of Analyzing Baseball Data with R

(Retrosheet Game Log) Jim Bunning pitched a perfect game on Father's Day on June 21, 1964. Some details about this particular game can be found from the Retrosheet game logs.

(a) What was the time in hours and minutes of this particular game?

```r
q4a <- getRetrosheet(type = "game", year = 1964) %>%
  select(Date, Duration, WinPNm) %>%
  filter(WinPNm == "Jim Bunning") %>%
  filter(Date == "19640621") %>%
  mutate(Hrs = Duration / 60) %>%
  mutate(Hours = as.integer(Hrs)) %>%
  mutate(Minutes = (Hrs - Hours) * 60) %>%
  select(Hours, Minutes); q4a
```

```
##   Hours Minutes
## 1     2      19
```

(b) Why is the attendance value in this record equal to zero?

```
q4b <- getRetrosheet(type = "game", year = 1964) %>%
  select(Date, Attendance, WinPNm) %>%
  filter(WinPNm == "Jim Bunning") %>%
  filter(Date == "19640621") %>%
  select(Attendance); q4b
```

```
##   Attendance
## 1          0
```

The attendance value in this record is equal to zero because it was the first half of a doubleheader. Fans likely only needed one ticket to attend both games, so `0` was recorded for the first game, and the actual attendance was entered for the second game of the doubleheader.

(c) How many extra base hits did the Phillies have in this game? (We know that the Mets had no extra base hits this game.)

```
q4c <- getRetrosheet(type = "game", year = 1964) %>%
  select(Date, WinPNm, VisD, VisT, VisHR, HmD, HmT, HmHR) %>%
  filter(WinPNm == "Jim Bunning") %>%
  filter(Date == "19640621") %>%
  mutate(TotXBH = VisD + VisT + VisHR) %>%
  select(TotXBH); q4c
```

```
##   TotXBH
## 1      3
```

(d) What was the Phillies' on-base percentage in this game?

```
q4d <- getRetrosheet(type = "game", year = 1964) %>%
     select(WinPNm, Date, VisH, VisBB, VisHBP, VisAB, VisSF) %>%
     filter(WinPNm == "Jim Bunning") %>%
     filter(Date == "19640621") %>%
     mutate(OBP = (VisH + VisBB + VisHBP) / (VisAB + VisBB + VisHBP + VisSF)) %>%
     select(OBP); q4d
```

```
##         OBP
## 1 0.3333333
```