# STA 141A Final Project

## Work was split between Ryan Cosgrove and Daniel Soriano Only, Atsushi has been unresponsive and did not contribute to the final project

**Ryan Cosgrove, Daniel Soriano, and Atsushi Higuchi**

**2023-12-15**

## Introduction

In this project we attempt evaluate the hitting statistics used in baseball and to answer the question of what hitting statistics help a baseball team win and what hitting statistics belong in the regression model for winning percentage. We will analyze team statistics for every Major League Baseball team in the 21st century that had a full season, so we will filter the teams from the year 2000 to the present, excluding the year 2020 because it was a short season due to COVID-19.

## Our Data Explained

The data we are using is from a baseball database called Lahman Database which can be imported from installing the package “Lahman” and we then use the Teams dataframe.

We will use the following team statistics:

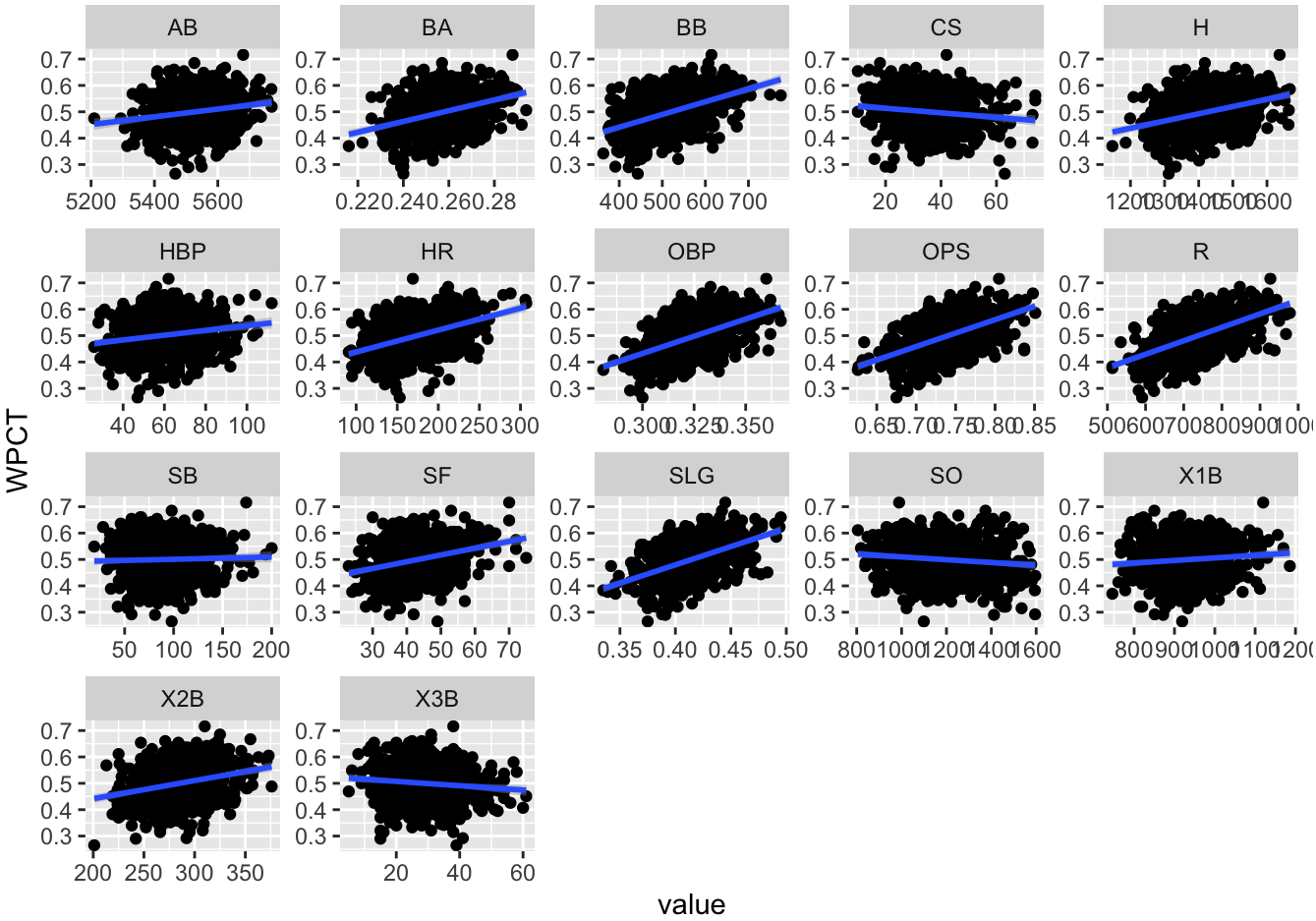
- `WPCT (Winning Percentage)`: Wins / Games
- `BA (Batting Average)`: Hits / at bats
- `OBP (On Base Percentage)`: (H + BB + HBP) / (AB + BB + HBP + SF)
- `SLG (Slugging Percentage)`: (X1B + 2 * X2B + 3 * X3B + 4 * HR) / AB
- `OPS (On Base Plus Slugging Percentage)`: OBP + SLG
- `R` : Runs scored
- `AB` : At-Bats
- `H` : Hits
- `X1B` : Singles hit
- `X2B` : Doubles hit
- `X3B` : Triples hit
- `HR` : Home runs hit
- `BB` : Bases on balls or also called walks
- `SO` : Strikeouts
- `SB` : Stolen bases
- `CS` : Caught stealing
- `HBP` : Times hit by a pitch
- `SF` : Sacrifice flies

```
##    WPCT    BA   OBP   SLG   OPS   R   AB    H  X1B X2B X3B  HR  BB   SO  SB CS
## 1 0.506 0.280 0.352 0.472 0.825 864 5628 1574  995 309  34 236 608 1024  93 52
## 2 0.525 0.265 0.333 0.429 0.763 792 5527 1466  961 282  44 179 535  975  97 44
## 3 0.586 0.271 0.346 0.429 0.775 810 5489 1490 1011 274  26 179 595 1010 148 56
## 4 0.457 0.272 0.341 0.435 0.776 794 5549 1508  992 310  22 184 558  900 126 65
## 5 0.525 0.267 0.341 0.423 0.764 792 5630 1503  988 316  32 167 611 1019  43 30
## 6 0.586 0.286 0.356 0.470 0.826 978 5646 1615 1041 325  33 216 591  960 119 42
##   HBP SF
## 1  47 43
## 2  59 58
## 3  59 45
## 4  49 54
## 5  42 48
## 6  53 61
```

# Regression/Correlation Coefficient

To see what statistics have the best relationship with Winning percentage, we can plot all the statistics against WPCT and add a regression line to see the relationship pattern. We also can calculate the correlation coefficient for all the statistics against WPCT and arrange them by strongest positive to strongest negative relationship.

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



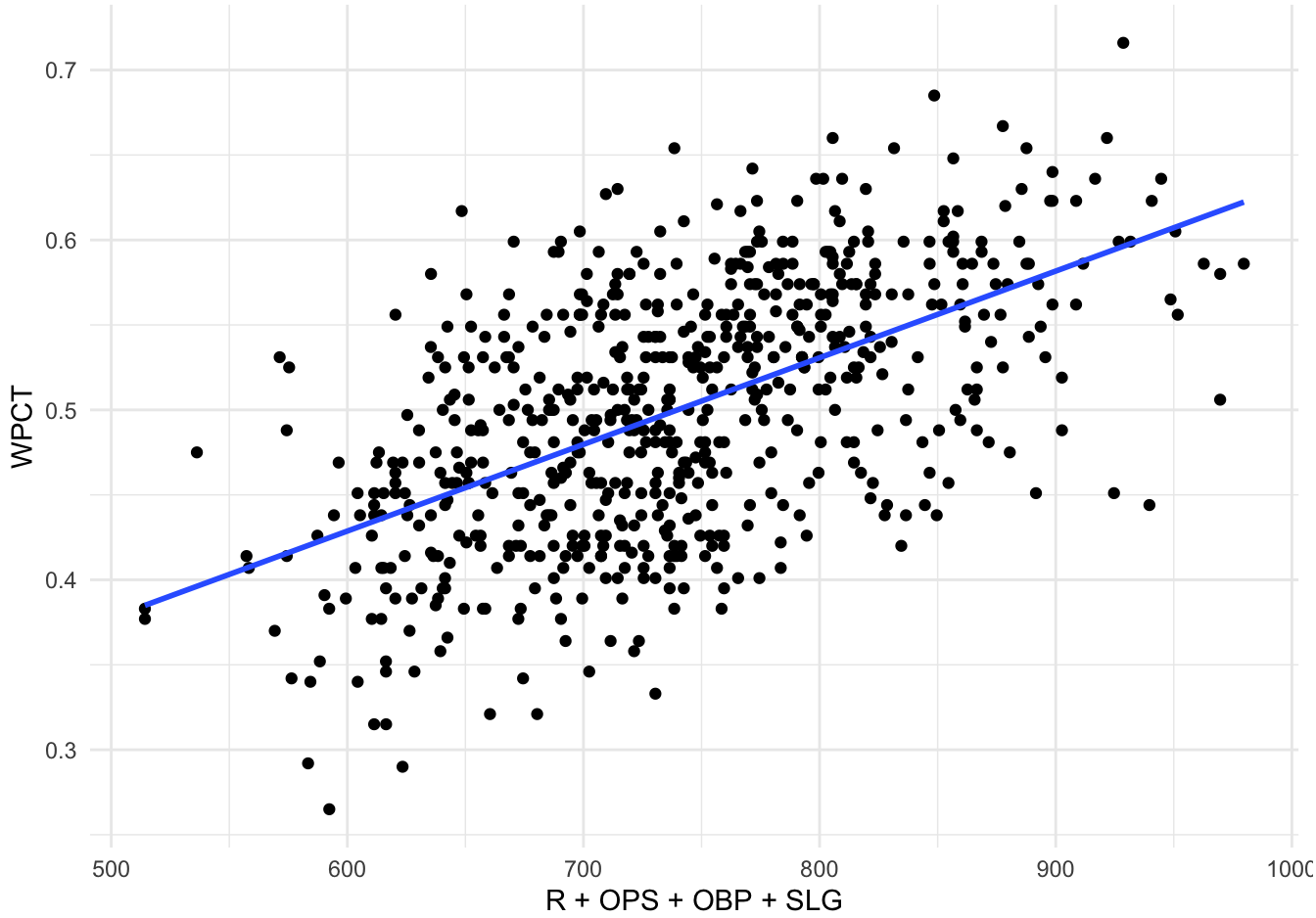
```
##     Cor with WPCT
## R      0.56838745
## OPS    0.53668488
## OBP    0.51457489
## SLG    0.50167314
## BB     0.45130803
## HR     0.41199494
## BA     0.35431896
## H      0.32714509
## SF     0.29768774
## X2B    0.26485907
## HBP    0.17601691
## AB     0.16729716
## X1B    0.09673114
## SB     0.03472038
## X3B   -0.10041498
## SO    -0.12128276
## CS    -0.12795377
```

We can conclude that from the plots and Correlation coefficients that the statistics R, OPS, OBP, SLG have the strongest positive relationships with winning percentage of all these statistics but they aren't necessarily close to 1 indicating that none of these statistics connect perfectly to how many wins a team has. The statistics that have almost a neutral relationship with winning are AB, singles, SB, triples, strikeouts, and caught stealings, where we can conclude that just because a team performs well by these statistics doesn't mean they have a high winning percentage.

```
##
## Call:
## lm(formula = WPCT ~ ., data = teams_21stcent)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.163247 -0.039090  0.002368  0.039201  0.175193
##
## Coefficients: (1 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.687e+00  2.230e+00  -2.102 0.035914 *
## BA          -7.528e-01  6.835e+00  -0.110 0.912337
## OBP          3.477e+00  6.448e+00   0.539 0.589844
## SLG          5.913e+00  6.326e+00   0.935 0.350216
## OPS          3.389e+00  5.104e+00   0.664 0.506988
## R            4.164e-04  9.908e-05   4.202 3.02e-05 ***
## AB           7.899e-04  3.896e-04   2.027 0.043051 *
## H           -7.478e-03  3.115e-03  -2.401 0.016642 *
## X1B          5.002e-03  2.676e-03   1.869 0.062028 .
## X2B          3.197e-03  1.794e-03   1.782 0.075215 .
## X3B          7.459e-04  9.269e-04   0.805 0.421271
## HR                  NA         NA      NA       NA
## BB          -6.300e-04  6.329e-04  -0.995 0.319914
## SO          -1.001e-05  2.113e-05  -0.473 0.636052
## SB           3.607e-04  9.816e-05   3.675 0.000258 ***
## CS          -1.262e-03  2.752e-04  -4.584 5.48e-06 ***
## HBP         -5.346e-04  6.515e-04  -0.821 0.412144
## SF           9.179e-04  4.573e-04   2.007 0.045166 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05771 on 643 degrees of freedom
## Multiple R-squared:  0.4104, Adjusted R-squared:  0.3957
## F-statistic: 27.97 on 16 and 643 DF,  p-value: < 2.2e-16
```

Here is a plots of the statistics R, OPS, OBP, SLG that have the strongest positive relationships with winning percentage of all these statistics

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




# $R^2$/P-values

From looking full model, we can create an adjusted model with the variables where the p-values are less than $\alpha = 0.05$ since the p-values for each independent variable tests the $H_0$ that there is no correlation between the independent and the dependent variable. In other words, we are trying to see if we can reject the null hypothesis, $H_0 : \beta_i = 0$ Those variables would be runs (R), hits(H), at bats(AB), stolen bases (SB), caught stealing (CS), and sacrifice flies (SF).

Adjusted Model without the lowest P-Values from the full model

```
##
## Call:
## lm(formula = WPCT ~ BA + OBP + SLG + OPS + X1B + X2B + X3B +
##     HR + BB + SO + HBP, data = teams_21stcent)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.187533 -0.042923  0.002266  0.042668  0.163794
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.895e-02  1.789e-01  -0.385 0.700039
## BA          -3.552e+00  6.524e+00  -0.544 0.586318
## OBP         -6.689e+00  5.965e+00  -1.121 0.262538
## SLG          1.211e+01  5.985e+00   2.023 0.043484 *
## OPS         -1.026e+00  5.080e+00  -0.202 0.839956
## X1B          3.036e-04  4.461e-04   0.681 0.496299
## X2B         -1.704e-03  4.420e-04  -3.857 0.000126 ***
## X3B         -4.427e-03  1.226e-03  -3.610 0.000330 ***
## HR          -5.231e-03  1.945e-03  -2.690 0.007332 **
## BB           1.108e-03  4.439e-04   2.495 0.012835 *
## SO           1.526e-08  2.170e-05   0.001 0.999439
## HBP          1.200e-03  4.774e-04   2.513 0.012218 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05998 on 648 degrees of freedom
## Multiple R-squared:  0.3581, Adjusted R-squared:  0.3472
## F-statistic: 32.86 on 11 and 648 DF,  p-value: < 2.2e-16
```

Model with top 4 correlation coefficients (Runs, OPS, OBP, SLG)

```
##
## Call:
## lm(formula = WPCT ~ R + OPS + OBP + SLG, data = teams_21stcent)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.161478 -0.044489  0.003589  0.041897  0.163610
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0801789  0.0888735   0.902    0.367
## R            0.0005227  0.0000922   5.669 2.15e-08 ***
## OPS         -2.1725042  4.8742513  -0.446    0.656
## OBP          2.6141241  4.8672173   0.537    0.591
## SLG          1.9071148  4.8847539   0.390    0.696
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06107 on 655 degrees of freedom
## Multiple R-squared:  0.3274, Adjusted R-squared:  0.3232
## F-statistic: 79.69 on 4 and 655 DF,  p-value: < 2.2e-16
```

If we compare the adjusted model to the model with the highest correlation coefficients, both models do have a relatively low adjusted R squared value. But this is to be expected as R squared values measure how well the model is fitting the actual data, and from our different adjusted R-squared values, we see that the model with high the highest correlation coefficients has the lowest R-squared value, the adjusted model having a slightly higher R-squared value, and our full model having the highest R-squared value. For our 3 different model's R-squared values, roughly 32-40% of the variance found in the response variables, win percentage, can be explained by all these different predictor variables. Retrospectively, if you were able to choose any metric to predict winning percentage, it looks to be that offensive stats seem to only account for 32-40% of the game. As an athlete, this does make sense in the sport of baseball as there are many other aspects of the game where winning a game is dictated by other aspects of the game such as defense (defensive/pitching stats) or possibly even external factors such as weather conditions or different stadiums since no two MLB stadiums are the same. What we did discover is that the MLB games are not won with 50-50 offense and defense, there are more aspects to the game, as offensive stats historically only really account for at most 40% of your likely hood of winning games.

# Akaike Information Criteria/AIC

Since new predictors will always improve the model fit, even if they just marginally explain variation in the response. To penalize too many predictors in the model, an information criteria considers the fit in terms of likelihood minus the number of parameters.

Here, $\ell$ is the log-likelihood of the data.

```
## AIC of the full model:  -1873.296
```

```
##
## AIC of the adjusted model:  -1827.186
```

```
##
## AIC of the highest correlation coefficient model:  -1810.341
```

```
##
##
## BIC of the full model:  -1792.435
```

```
##
## BIC of the adjusted model:  -1768.786
```

```
##
## BIC of the highest correlation coefficient model:  -1783.387
```

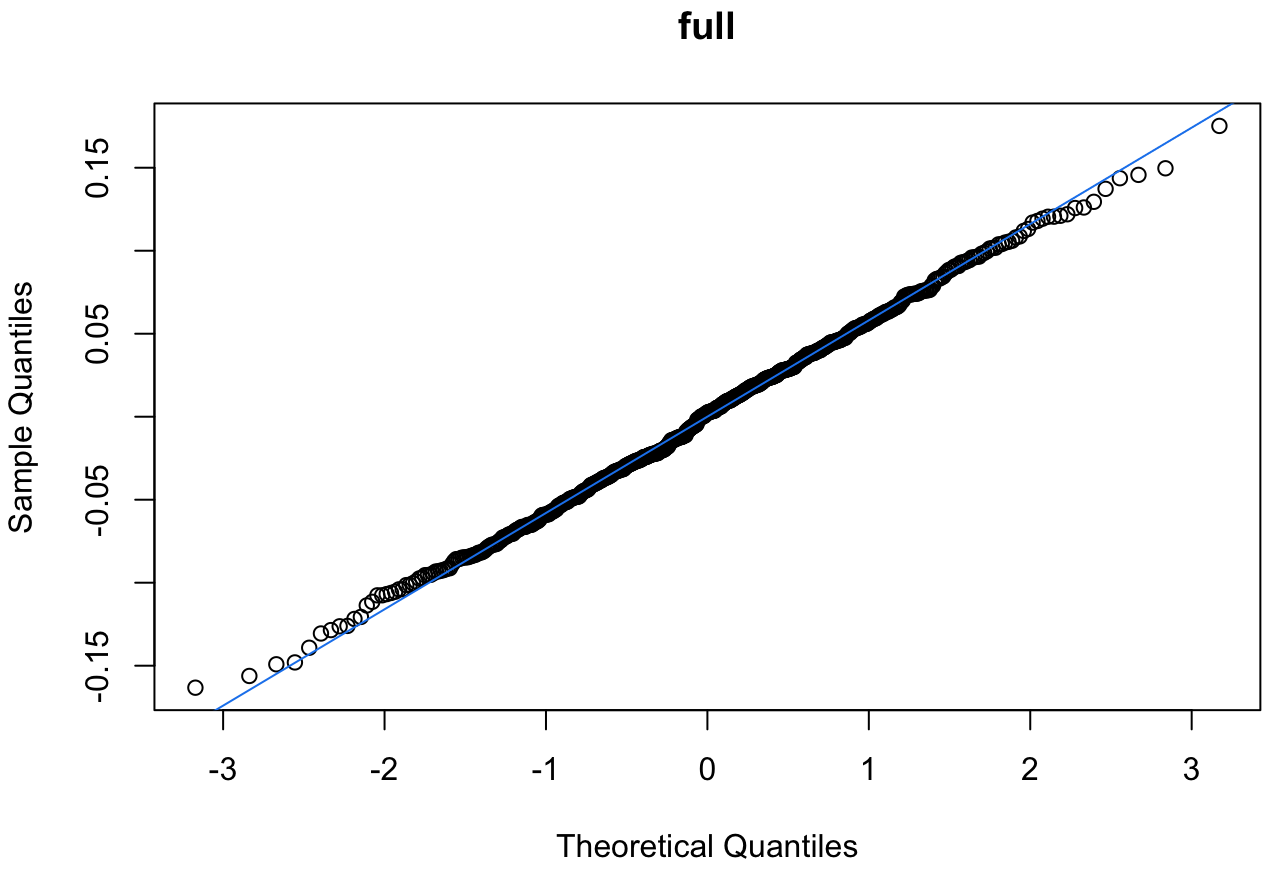
After looking at log-likelihood of the data, we see that the full model had both the lowest AIC and the lowest BIC, suggesting that we would still want to use our adjusted model. We can also confirm our answer by using cross validation.

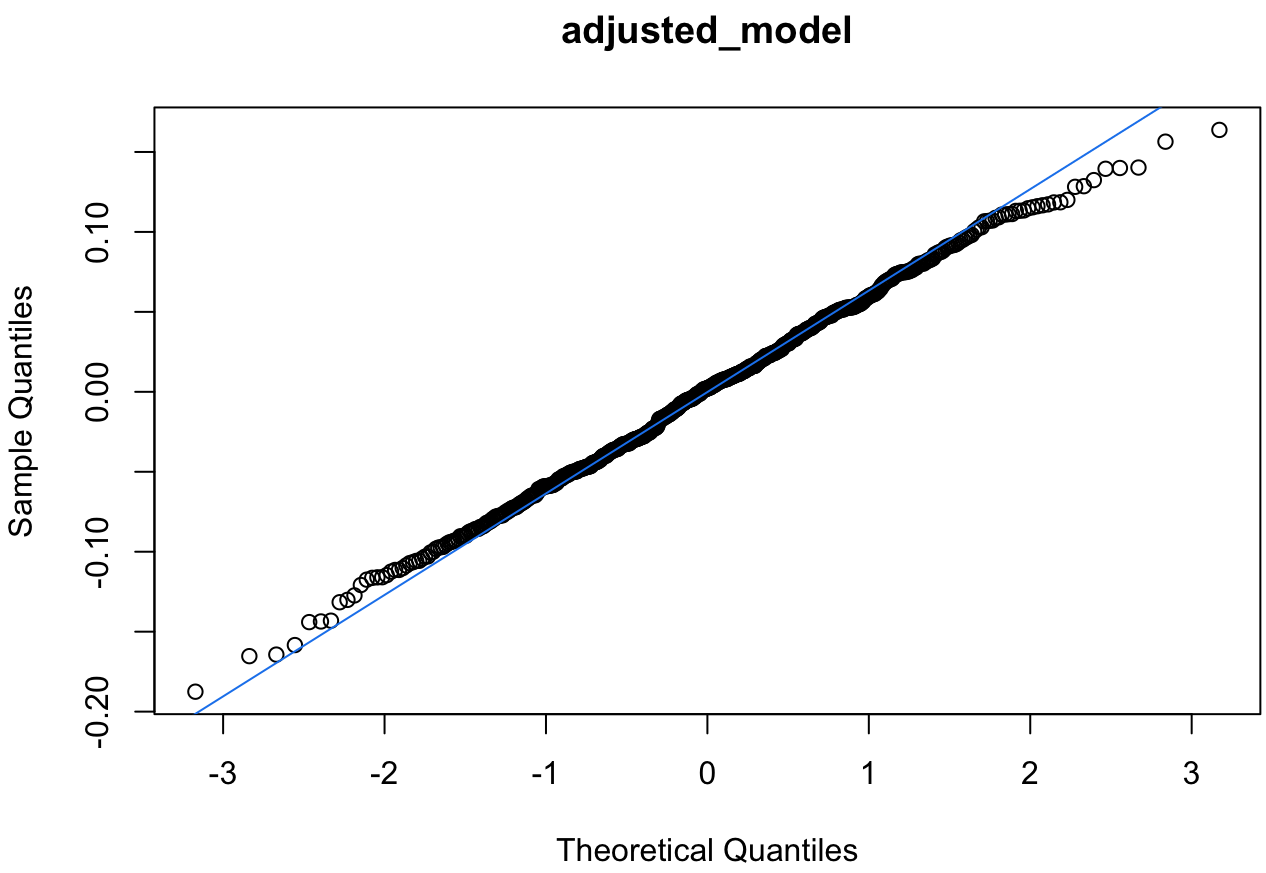
# Leave Out One Cross Validation/LOOCV

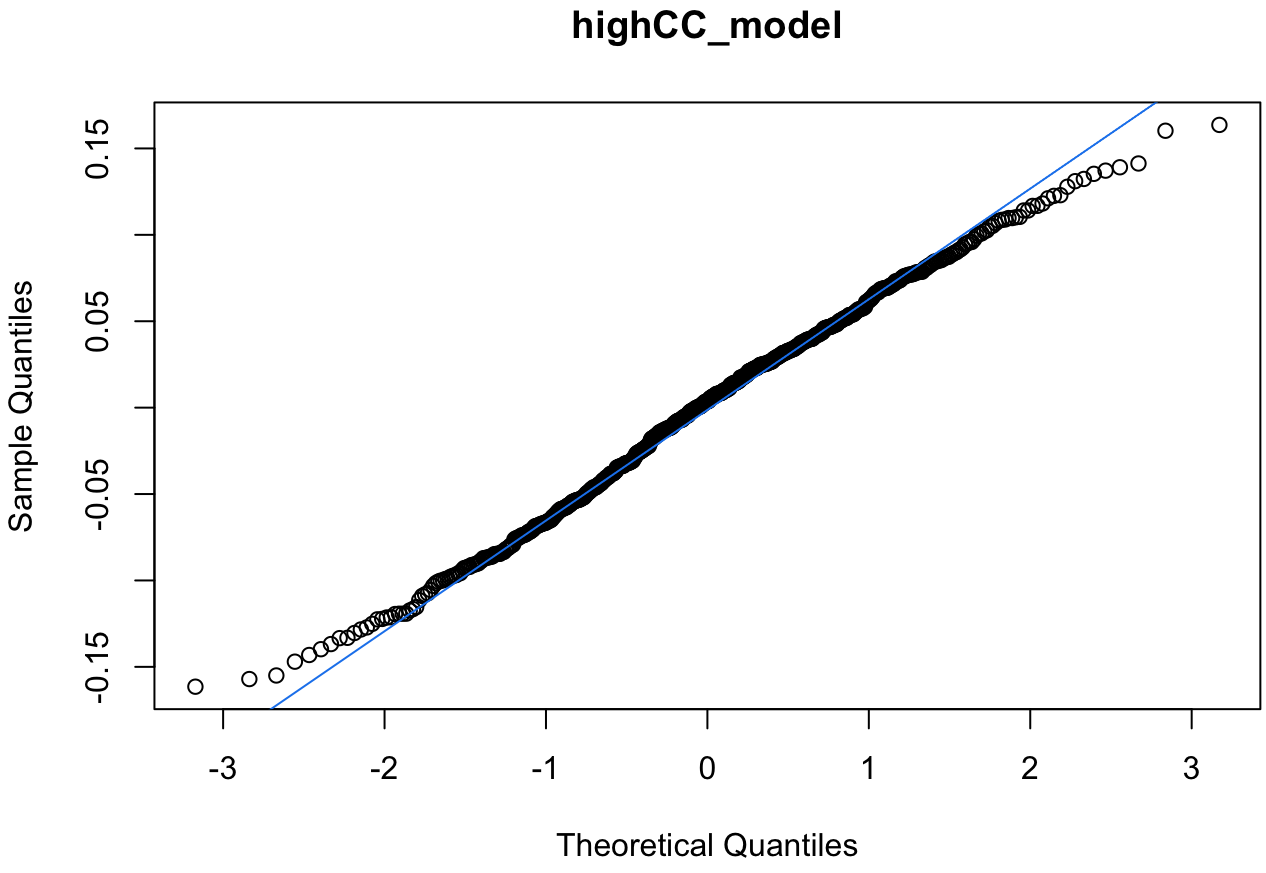
```
## $full
## [1] 0.003419155
##
## $adjusted_model
## [1] 0.003475353
##
## $highCC_model
## [1] 0.003758501
```

Since our MSE for the full model was marginally smaller than the adjusted model, we can say that the full model is performing slightly better than the other models.

# Residual Analysis (QQ Plots)




adjusted_model
Sample Quantiles
Theoretical Quantiles
-0.20
-0.10
0.00
0.10
-3
-2
-1
0
1
2
3


highCC_model
Sample Quantiles
Theoretical Quantiles
-0.15
-0.05
0.05
0.15
-3
-2
-1
0
1
2
3

For the full residual analysis, we can see that our best model is the full model and close in second is our adjusted model. Both of which seem to be more normally distributed that our high correlation coefficient model. In the highCC_model, we see that the points are close to the line, but tail off towards the edges more than the other two plots meaning there must be a peak or a spike in our data, more specifically, we would see a spike commonly around the median values. But it probably isn't too noticeable as the highCC_model is still a very good QQ plot.

# F-Tests

Looking at our F-Tests for our two different models:

```
## F-Statistic:  79.69229
```

```
## Numerator Degrees of Freedom:  4
```

```
## Denominator Degrees of Freedom:  655
```

```
## Adjusted model critical value:  2.385533
```

High CC Model null hypothesis $\mathrm{H}_0$: The high cc model fits the data as well as the full model. High CC Model alternative hypothesis $\mathrm{H}_\mathrm{a}$: The full model fits the data significantly better than the high cc model.

Conclusion: With an F-statistic of 79.69229, 4 degrees of freedom in the numerator, and 655 in the denominator for the full model, compared to an adjusted model critical value of 2.385533, the F-statistic substantially surpasses the critical value for the adjusted model, signaling strong statistical significance. Consequently, we reject the null hypothesis $\mathrm{H}_0$ asserting that the adjusted model fits the data as well as the full model. Instead, robust evidence supports the alternative hypothesis $\mathrm{H}_\mathrm{a}$ that the full model significantly surpasses the adjusted model in capturing and explaining the observed variability within the data set.

```
## F-Statistic:  32.85943
```

```
## Numerator Degrees of Freedom:  11
```

```
## Denominator Degrees of Freedom:  648
```

```
## Adjusted model critical value:  1.803418
```

Adjusted Model null hypothesis $\mathrm{H}_0$: The adjusted model fits the data as well as the full model. Adjusted alternative hypothesis $\mathrm{H}_\mathrm{a}$: The full model fits the data significantly better than the adjusted model.

Conclusion: Given the calculated F-statistic of 67.16695 exceeding the critical value of 2.112446, the outcome leads to rejecting the null hypothesis $\mathrm{H}_0$ in favor of the alternative hypothesis $\mathrm{H}_\mathrm{a}$ that the adjusted model is statistically significant at the significance level $\alpha = 0.05$. The adjusted model substantially explains the data set's variation. This stark difference confirms that the chosen variables wield considerable collective influence on the dependent variable. The statistical evidence strongly supports the relevance of these predictors, affirming their crucial role in modeling the data effectively.

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## ℹ Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(variable)
##
##   # Now:
##   data %>% select(all_of(variable))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Critical value for all models:  1.659253
```

```
## $BA
##       statistic df1 df2
## value  94.46603   1 658
##
## $OBP
##       statistic df1 df2
## value  236.9791   1 658
##
## $SLG
##       statistic df1 df2
## value  221.2982   1 658
##
## $OPS
##       statistic df1 df2
## value  266.1971   1 658
##
## $R
##       statistic df1 df2
## value  314.0273   1 658
##
## $AB
##       statistic df1 df2
## value  18.94661   1 658
##
## $H
##       statistic df1 df2
## value  78.86183   1 658
##
## $X1B
##       statistic df1 df2
## value  6.215003   1 658
##
## $X2B
##       statistic df1 df2
## value  49.64127   1 658
##
## $X3B
##       statistic df1 df2
## value  6.702305   1 658
##
## $HR
##       statistic df1 df2
## value  134.5227   1 658
##
## $BB
##       statistic df1 df2
## value  168.2999   1 658
##
## $SO
##       statistic df1 df2
## value  9.823353   1 658
##
```

```
## $SB
##       statistic df1 df2
## value 0.7941794   1 658
##
## $CS
##       statistic df1 df2
## value   10.9522   1 658
##
## $HBP
##       statistic df1 df2
## value  21.03792   1 658
##
## $SF
##       statistic df1 df2
## value  63.98046   1 658
```

If we wanted to look at each stat individually for their effect on Winning Percentage, it's evident that across the various offensive baseball statistics we observed, a majority of the the calculated F-statistics significantly surpass the critical value of 1.659253. This compelling finding indicates strong statistical significance for these statistics within their respective models in which they are the only variable. In essence, each of these individual statistics demonstrates substantial explanatory power in relation to the variability observed in the data set. Such significance underscores the importance of these offensive baseball metrics in understanding and predicting performance outcomes, thereby emphasizing their relevance in baseball analytics and player performance evaluation.

Of all the offensive baseball statistics, the F-statistic for the Stolen Bases ($\mathrm{SB}$) model falls below the critical value. It suggests that, in this particular context, the model might not possess enough collective explanatory power to significantly predict or explain the variability in the data set pertaining to stolen bases. This outcome indicates that the variables or factors included in this model, when considered together, might not adequately capture the nuances or influences affecting stolen base outcomes in the analyzed data set. It's essential to note that while this model might not exhibit significant explanatory capability in this specific instance, it doesn't diminish the importance of stolen bases as a metric in baseball.

```
knitr::opts_chunk$set(echo = TRUE)
knitr::opts_knit$set(root.dir = "/Users/danielsoriano/Downloads/Random School Files/Stat
s 141")
library(ggplot2)
library(dplyr)
library(Lahman)
library(tidyr)
library(boot)
library(gridExtra)
library(grid)
teams_21stcent <- Teams %>%
  filter(yearID > 1999 & yearID != 2020)%>%
  mutate(X1B = H - (X2B+X3B+HR),
         BA = H/AB,
         OBP = (H + BB + HBP) / (AB + BB + HBP + SF),
         SLG = (X1B + 2 * X2B + 3 * X3B + 4 * HR) / AB,
         OPS = OBP + SLG,
         WPCT = W / G,
         BA = round(BA, digits=3),
         OBP = round(OBP, digits=3),
         SLG = round(SLG, digits=3),
         OPS = round(OPS, digits=3),
         WPCT = round(WPCT, digits=3))%>%
  dplyr::select(WPCT, BA, OBP, SLG, OPS, R, AB, H, X1B, X2B, X3B, HR, BB, SO, SB, CS, HB
P, SF)

head(teams_21stcent)
attach(teams_21stcent)
ggplot(data = tidyr::pivot_longer(teams_21stcent %>%
                                    dplyr::select(WPCT, BA, OBP, SLG, OPS, R, AB, H, X1
B, X2B, X3B, HR, BB, SO, SB, CS, HBP, SF), -1) ,
       aes(value, WPCT)) +
  geom_point() + facet_wrap('name', scales = 'free')+
  geom_smooth(method = 'lm')

Correlations <- data.frame(c(cor(WPCT,BA), cor(WPCT,OBP), cor(WPCT,SLG), cor(WPCT,OPS),
cor(WPCT,R), cor(WPCT,AB), cor(WPCT,H), cor(WPCT,X1B), cor(WPCT,X2B), cor(WPCT,X3B), cor
(WPCT,HR), cor(WPCT,BB), cor(WPCT,SO), cor(WPCT,SB), cor(WPCT,CS), cor(WPCT,HBP), cor(WP
CT,SF)))
rownames(Correlations) = c('BA', 'OBP', 'SLG', 'OPS', 'R', 'AB', 'H', 'X1B', 'X2B', 'X3
B', 'HR', 'BB', 'SO', 'SB', 'CS', 'HBP', 'SF')
colnames(Correlations) = c('Cor with WPCT')

Correlations %>% arrange(desc(Correlations))
full = lm(WPCT ~ . , data = teams_21stcent)
summary(full)
require('ggplot2')
ggplot(teams_21stcent, aes(R + OPS + OBP + SLG, WPCT)) +
    theme_minimal() +
    geom_point() +
    geom_smooth(method = 'lm', se = F)
adjusted_model = lm(WPCT ~ BA + OBP + SLG + OPS + X1B + X2B + X3B + HR + BB + SO + HBP,
```

```
data = teams_21stcent)
summary(adjusted_model)
highCC_model = lm(WPCT ~ R + OPS + OBP + SLG, data = teams_21stcent)
summary(highCC_model)
x <- AIC(full)
y <- AIC(adjusted_model)
z <- AIC(highCC_model)
cat("AIC of the full model: ", x)
cat("\nAIC of the adjusted model: ", y)
cat("\nAIC of the highest correlation coefficient model: ", z)

a <- BIC(full)
b <- BIC(adjusted_model)
c <- BIC(highCC_model)
cat("\n\nBIC of the full model: ", a)
cat("\nBIC of the adjusted model: ", b)
cat("\nBIC of the highest correlation coefficient model: ", c)
# LOOCV function
loocv <- function(model_formula, data) {
  predictions <- numeric(nrow(data))
  for (i in 1:nrow(data)) {
    data_leave_one_out <- data[-i, ]
    model <- lm(model_formula, data = data_leave_one_out)
    prediction <- predict(model, newdata = data[i, ])
    predictions[i] <- prediction
  }
  actual <- data$WPCT
  mse_loocv <- mean((actual - predictions)^2)
  return(mse_loocv)
}
models <- list(
  full = "WPCT ~ .",
  adjusted_model = "WPCT ~ R + AB + H + SB + CS + SF",
  highCC_model = "WPCT ~ R + OPS + OBP + SLG"
)

results <- lapply(models, function(formula) loocv(formula, teams_21stcent))
names(results) <- names(models)
results
residuals_full <- residuals(full)
residuals_adjusted <- residuals(adjusted_model)
residuals_highCC <- residuals(highCC_model)
#par(mfrow = c(1, 3))
qqnorm(residuals_full, main = "full")
qqline(residuals_full, col = "dodgerblue2")

qqnorm(residuals_adjusted, main = "adjusted_model")
qqline(residuals_adjusted, col = "dodgerblue2")

qqnorm(residuals_highCC, main = "highCC_model")
qqline(residuals_highCC, col = "dodgerblue2")
F_statistic_highcc = summary(highCC_model)$fstatistic[1]
```

```
df1_cc = summary(highCC_model)$fstatistic[2]
df2_cc = summary(highCC_model)$fstatistic[3]

cat("F-Statistic: ", F_statistic_highcc)
cat("Numerator Degrees of Freedom: ", df1_cc)
cat("Denominator Degrees of Freedom: ", df2_cc)

alpha <- 0.05

critical_value_highcc <- qf(1 - alpha, df1_cc, df2_cc)
cat("Adjusted model critical value: ", critical_value_highcc)
F_statistic_adjusted = summary(adjusted_model)$fstatistic[1]
df1_a = summary(adjusted_model)$fstatistic[2]
df2_a = summary(adjusted_model)$fstatistic[3]

cat("F-Statistic: ", F_statistic_adjusted)
cat("Numerator Degrees of Freedom: ", df1_a)
cat("Denominator Degrees of Freedom: ", df2_a)

alpha <- 0.05
critical_value_adjusted <- qf(1 - alpha, df1_a, df2_a)
cat("Adjusted model critical value: ", critical_value_adjusted)
perform_F_test <- function(variable) {
 full = lm(WPCT ~ . , data = teams_21stcent %>% select(variable, WPCT))
 f_test_result = data.frame(statistic = summary(full)$fstatistic[1],
                            df1 = summary(full)$fstatistic[2],
                            df2 = summary(full)$fstatistic[3])
 return(f_test_result)
}


f_test_results = list()

variables = c('BA', 'OBP', 'SLG', 'OPS', 'R', 'AB', 'H', 'X1B', 'X2B', 'X3B', 'HR', 'B
B', 'SO', 'SB', 'CS', 'HBP', 'SF')

for (variable in variables) {
 f_test_results[[variable]] = perform_F_test(variable)
}

df1_z = summary(full)$fstatistic[2]
df2_z = summary(full)$fstatistic[3]
critical_value_adjusted <- qf(0.95, df1_z, df2_z)
cat("Critical value for all models: ", critical_value_adjusted)
cat("\n")
print(f_test_results)
```