Exploring the 2021 MLB ban on grip enhancers using non-parametric methods and Lasso regression

Shea Frantz
frantzs2@wwu.edu

David Tabakian
tabakid@wwu.edu

Eric Folsom
folsome@wwu.edu

March 18, 2022

1 Introduction

We seek to use techniques of non-parametric statistics to analyze MLB data gathered with baseball savant[1] and baseball reference[5] via the baseballr[4] package. For our initial research question, we explore how the MLB policy change on "sticky stuff"[6] impacted various pitching statistics of qualified (minimum 100 plate appearances for each half of the season) MLB pitchers throughout the 2021 season. Our methodology includes hypothesis testing about a change in medians using the Wilcoxon Signed-Rank test to see if there were statistically significant changes which occurred in relevant pitching statistics from before the policy change compared to afterwards. Notably, this will be a paired test since they will be the same players before and after. We are investigating how the ban on "stick stuff" impacted pitchers' spin rate because of the impact that spin rate has on the trajectory of the baseball. A higher spin rate will lead to more movement[3], so we would like to investigate the effects of the "stick stuff" ban so that we can better understand how the ban affects pitcher performance.

Additionally, we investigate the validity of a linear model using characteristics of pitching mechanics (i.e spin rate, velocity) to predict metrics measuring pitchers' performance (i.e ERA (earned run average), WHIP (walks and hits per inning pitched)), namely the lasso regression. Once we've built our model, we may identify significant predictors for pitching performance, and investigate differences in these predictors before the MLB policy change and afterwards.

2 Methodology

2.1 Hypotheses

We set up the following hypotheses to test:

H_0 : MLB pitchers gained no significant advantage from the use of grip enhancers in the 2021 MLB season. (i.e $\theta_{pre} = \theta_{post}$)

H_1 : MLB pitchers gained a significant advantage from the use of grip enhancers in the 2021 MLB season. (i.e $\theta_{pre} \neq \theta_{post}$)

The inequality of H_1 can vary depending on the statistic in question, for some (namely spin rate) we expect a higher spin rate to be more beneficial[3] to the pitcher. As such, our alternative hypothesis would be $H_1 : \theta_{pre} > \theta_{post}$. Likewise, for woba (which measures weighted on base average of the batter the pitcher is facing) we would expect a higher woba to indicate a worse pitching performance, the inverse of spin rate. As such, our alternative hypothesis would be $H_1 : \theta_{pre} < \theta_{post}$.

To test these hypotheses we gathered the pitching data for the 2021 MLB season from MLB savant. We then split that into two data sets, one from games played prior to June 21, 2021 and one after that date, as that's the date that the ban went into effect. The specific hypothesis test we use is the Wilcoxon Signed Rank test, as it allows us to make minimal

2.2 Lasso Regression

To begin talking about our linear regression, we need to delve into the nature of our statistics a bit more. As mentioned in the introduction, there are two categories of pitching statistics we're dealing with, pitching mechanics and pitching performance. Pitching performance statistics are more traditional sports statistics which measure how good the pitcher did in an appearance. How many runs they gave up, how many runners they allowed on base, how many batters they strike out, these and more are counted and used to analyze a pitchers overall performance. Pitching

mechanics statistics are a lot newer, as they use new Statcast technologies[1] to analyze the pieces that comprise each individual pitch the pitcher throws. How far does the pitcher extend off of the mound when they're throwing? How fast is the ball spinning? These sorts of questions are answered by this new Statcast technology.

In our regression model we seek to predict important pitching performance statistics using relevant pitching mechanics statistics. Specifically, we want to use the Lasso regression so we can identify significant beta coefficients for the regression model, in order to help inform our choice of relevant statistics for the hypothesis test outlined earlier. This will allow our regression to serve two functions. First, the model will allow us to plug in pitchers' pitching mechanics statistics from before and after the policy change, to see if their mechanics as a whole indicate that they were gaining an advantage from foreign substances before the ban. Second, one of the characteristics of the Lasso regression is that it drops insignificant predictors from its final model. As such, we can fit 100 Lasso regressions, then use the rate at which predictors are accepted in the final model to determine their overall significance. This provides a mathematical basis in addition to the conventional wisdom we used in picking the statistics for our Wilcoxon Rank Sum tests.

3 Results

First we'll be going into detail regarding the results of our Lasso regression. We were interested in fitting our regression over the entire 2021 MLB season, to provide a full picture of pitching throughout the season. To do so, we did various data manipulations with a few different data sets to get access to what we wanted, which we choose not to go over here but is roughly described in the R code comments. We experimented with many different models, which helped in choosing the best final model for our analysis, along with providing a clear picture of significant predictors to use in our analysis. Ultimately we achieved our best model for trying to predict strikeouts per nine innings (SO9), or how many strikeouts the pitcher is able to get averaged over 9 innings. For example, if a pitcher had 40 strikeouts over 30 innings pitched, they would have a SO9 of 6.8. Following the simulation of 100 Lasso regression to predict SO9, here was our output:

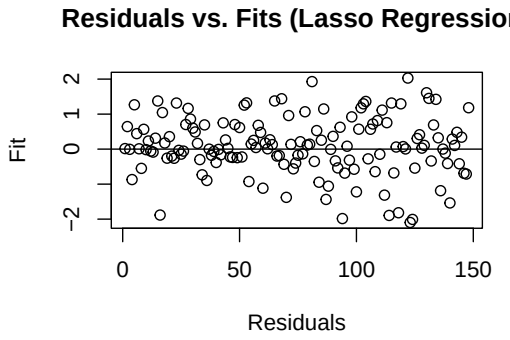
```
> # Showing the probability of selecting each predictor for the Lasso
> colMeans(lasso01mat)
       spin_rate         velocity release_extension      launch_angle             avg_x             avg_z             BAbip 
            1.00             1.00             1.00             0.99             0.73             1.00             0.86 
> # Showing the mean of the beta coefficients for each predictor for the Lasso
> colMeans(lassobetamat)
       spin_rate         velocity release_extension      launch_angle             avg_x             avg_z             BAbip 
      0.22841834       0.31831522       0.14250421       0.19326050       0.03380225      -0.16202228       0.07642666 
> # Mean of the MSE values
> mean(mse.lasso)
[1] 0.7434878
```

To provide context to the output, the first results (colmeans(lasso01mat)) are the rates at which those corresponding predictors were accepted in the final Lasso regression. For example, "launch_angle" as a predictor was a part of our final model 99% of the time in our simulations, indicating that it's a significant predictor. For SO9, we identified spin rate, velocity, release extension, launch angle, average horizontal movement from the catcher's perspective (avg_x), average vertical movement from the catcher's perspective (avg_z), and batting average on balls in play (BAbip) as relevant predictors. The second listed results (colmeans(lassobetamat)) are the β coefficients for our linear model. As such, our linear model would take the form:

$$SO9 = 0.2284x_{sr} + 0.3183x_v + 0.1425x_{re} + 0.1932x_{la} + 0.0338x_{avg_x} - 0.162x_{avg_z} + 0.0764x_{BAbip} \quad (1)$$

However, this must be taken with a grain of salt. Due to the nature of our statistics, namely the different scales at which each of them operate, we scaled our data before conducting the regression. As such, this regression model is only suited to providing a picture of how significant individual predictors in relation to each other. For example, the β coefficient for x_{re} (0.1425) is much smaller than that of x_v (0.3183), which indicates that the velocity of a pitch is more impactful on the pitchers' SO9 than their release extension. As such, while the scales are inexact, it can still serve to provide a very rough picture of SO9 performance.

The last aspect of our output is the mean square error (MSE) of our regression model. The average MSE over our 100 simulated regressions was 0.7435, a very solid value indicating our Lasso regression did a good job. Further reinforcing this was an example residual plot from one of the 100 simulated Lasso regressions:



From the residual plot we can see that model assumptions are satisfied (constant variance, no heteroskedasticity), and that our model describes the data reasonably well. From here, we wanted to see if we could use our model to show that there was a difference in what it would predict for pitchers using their first half of the season predictors compared to the latter half, as a precursor to our hypothesis tests. Again, this doesn't actually tell us what the pitcher's actual predicted SO9 was, because the model was made using scaled data. However, it does provide a very rough picture to tell us what we should expect going forward:

```
> # average prediction for first half of season (pre-policy change)
> mean(pred_pre, na.rm=TRUE)
[1] 565.2032
> # average prediction for latter half of season (post-policy change)
> mean(pred_post, na.rm=TRUE)
[1] 554.9652
```

From this output, we see that our model predicts a "higher SO9" for the first half of the season than the latter half. Again, this isn't even close to an exact estimate, only an approximation to give us a general idea. This gives us the general thought that pitchers' predictive statistics were higher in the first half compared to the latter half, indicating they may be recipients of some sort of advantage.

Now that we've identified our parameters of interest, we may carry out our Wilcoxon Signed-Rank tests to see if they were significantly different between the first and latter half of the season. As such we carried out Wilcoxon Signed-Rank tests on statistics our model indicated could be helpful to look at, as well as statistics we thought would be helpful because of prior knowledge. Our results were somewhat surprising, as there were a few of the variables we expected to be significant were not. The table below shows the p-values of our different predictors for the above hypothesis tests.

Statistic:	SR	WOBA	BA	Velocity	RE	LA	avg_x	avg_z
p-value:	0.0125	0.0001431	0.000007	0.6159	0.4938	0.9474	0.5607	0.4477

Statistic:	BAbip	FIP	ERA	WHIP	SO9
p-value:	0.4815	0.3627	0.5698	0.5214	0.693

Interpreting these values, we were very surprised how few of them were actually significant. The three statistics that indicated that pitchers were gaining a statistically significant advantage from the presence of sticky stuff prior to the ban were spin rate, WOBA, and batting average against (the percentage of batters faced which ended up in a hit). Meanwhile, every other predictor indicated by our model, or that we thought would be significant ended up being insignificant.

4 Discussion

Our p-values weren't what we expected them to be, however they still provide immense insight into the question we wanted to answer. We were interested in whether or not pitchers were gaining a significant advantage as a result of using sticky stuff, and whether or not the policy change would significantly change pitching statistics as a result of it's implementation. The answer to our question ended up being a bit ambiguous. The minority of statistics we thought may be significant ended up giving us a significant p-value, but those statistics are relatively important. Spin rate is one of the most commonly pointed to "new" analytics to indicate movement on pitches, and the fact that spin rate was significantly different between the two halves of the season is reasonably strong evidence that there was perhaps some advantage being gained in the first half of the season. This is similar for WOBA (weighted on base average) and BA (batting average).

However, where this is called into question are the remaining statistics. Not a single one signalled that pitchers were recipients of a significant competitive advantage during the first half of the season, prior to the policy change. Some of these weren't as surprising, such as LA (launch angle) and BAbip (batting average on balls in play), as the pitcher inherently has much less control

over these statistics than others. Although this wasn't the case for all of these statistics, as WHIP (walks and hits per inning pitched) is very similar to WOBA and BA in what it measures, and avg_x and avg_z are somewhat similar to spin rate in that they're measurements of a specific pitch characteristic. As such, we cannot really come close to definitively saying that pitchers were gaining a significant advantage from sticky stuff prior to the MLB policy change.

5 Conclusion

There are a lot of things which could be done to extend our research. First, we realized after we had completed our analysis that the initial restriction we imposed on qualified pitchers was perhaps unnecessary. We limited our sample to pitchers with over 100 plate appearances in each half of the season so that we would only be looking at pitchers who pitched a lot over both parts of the season. However, in doing so, we're only looking at a fraction of overall pitchers in the MLB, with many relief pitchers not reaching that qualification. In hindsight, the statistics of these relievers is still valuable, and may perhaps be even more valuable than that of the starters since they're often pitching in higher leverage situations which could induce them to be more willing to use sticky stuff to augment their pitches. Additionally, work could be done to expand our model we build using Lasso regression to the point where it could perform accurate predictive analytics, rather than rough comparisons as a result of the scaling.

Going forward, there are quite a few interesting ideas to explore for this data set. Namely, time-series analysis could be helpful to implement alongside linear regression and hypothesis testing, as the composition of our data lends itself to time-series analysis, with games taking place at semi-regular intervals throughout the course of the season. This could provide further insight into overall pitching trends throughout the course of a season, and assist the regression in making predictions regarding the future. We also think after working closely with the data that the nature of our data may lend itself to parametric methods of analysis more than non-parametric methods. Our sample size is very large, so the central limit theorem helps us make relatively accurate assumptions about the normality of our data, and the data is all demonstrably independent by nature of the game of baseball.

Lastly, to truly answer the question accurately we believe that waiting for more data to surface would be valuable. We fit our regression and did our hypothesis tests over one season, as it was the only season with games played before and after the policy change. Once the 2022 MLB season is played we'll have a full season of games with the policy change in effect, which would allow us build a model from 2020-2022 similar to what we have, with half of the games occurring prior to the ban and half occurring after. This would triple our sample size, giving us a significantly clearer picture of what pitching looked like before the policy change, and what potentially changed as a result. Pitchers could adapt going forward, and it would be very interesting to see what the future of pitching holds.

6 References

[1] https://www.mlb.com/glossary/statcast
[2] https://www.smartfantasybaseball.com/tools/ - Player ID Map Excel Spreadsheet
[3] https://pubmed.ncbi.nlm.nih.gov/22923374/
[4] https://billpetti.github.io/baseballr/
[5] https://www.baseball-reference.com/
[6] https://www.mlb.com/news/faq-sticky-stuff-and-new-rule-enforcement

7 Appendix

All of our R code was relevant to our project, so we included it all. Let us know if you'd like an email containing the .R file as well as the 5 .csv files we use in our analysis so you can run it yourself if you're curious. Additionally, the baseballr package cannot be downloaded via conventional means, if you follow the link in source [4] in references, then that will provide instructions for downloading the package.

```
library(dplyr)
library(baseballr)
library(glmnet)

# comment this out if you aren't shea, or substitute whatever folder you've got the data in
setwd("~/winter 2022/math 446/project")

# initializing data
pre <- read.csv("pre_ban_min100pa.csv", header = TRUE, stringsAsFactors = FALSE)
post <- read.csv("post_ban_min100pa.csv", header = TRUE, stringsAsFactors = FALSE)
# filtering our data to pitchers who reached the prerequisite pitching threshold
# both before and after the policy change
overlap <- intersect(pre$player_id, post$player_id)
# the %>% character is indicating running a function on a dataset, so in this case we're filtering
# the pre d.f for player_id's *in* the overlap vector, then storing that new d.f in true_pre
true_pre <- pre %>%
  filter(player_id %in% overlap) %>%
  mutate(whiff_rate = whiffs / swings)
true_post <- post %>%
  filter(player_id %in% overlap) %>%
  mutate(whiff_rate = whiffs / swings)

# getting bref data from baseballr
bref_pre <- bref_daily_pitcher("2021-04-01", "2021-06-21")
bref_post <- bref_daily_pitcher("2021-06-21", "2021-10-03")
bref_pre <- fip_plus(bref_pre)
bref_post <- fip_plus(bref_post)
# making our map
map <- read.csv("id_map.csv", header = TRUE, stringsAsFactors = FALSE)
# note that when making the map the name column selected impacts how many NA's show up in
# the final product. Current name selection has 6 na's in final df. most NA's are dealt with later
colnames(map)[39] <- "Name"
f <- select(bref_pre, bbref_id, Name)
map <- left_join(map, f, by = "Name")
map <- map %>%
  filter(!is.na(bbref_id)) %>%
  select(MLBID, Name, bbref_id)
colnames(map)[1] <- "player_id"
# using the map to combine the baseball savant data with the bref data
true_pre <- left_join(true_pre, map, by = "player_id")
true_pre <- left_join(true_pre, bref_pre, by = "bbref_id")
true_pre <- true_pre[!duplicated(true_pre), ]
true_post <- left_join(true_post, map, by = "player_id")
true_post <- left_join(true_post, bref_pre, by = "bbref_id")
true_post <- true_post[!duplicated(true_post), ]
```

```
######################
# LASSO REGRESSION
######################

#initializing the full season data from baseball savant
full_season <- read.csv("full_season_data.csv", header = TRUE, stringsAsFactors = FALSE)
full_season <- left_join(full_season, map, by = "player_id")
full_season <- left_join(full_season, bref_pre, by = "bbref_id")
full_season <- full_season[!duplicated(full_season), ]
#creating our whiff_rate stat
full_season <- full_season %>% mutate(whiff_rate = whiffs / swings)

#getting full data from statcast
statcast_data <- read.csv("full_season_statcast.csv", header = TRUE)
# we want average pfx_x and pfx_z for each individual pitcher for fastballs and breaking balls
# adding those to our full_season data frame:
pitches <- c("FA", "FT", "FC", "SI", "SL", "CU", "KC", "KN", "EP")
statcast_data <- statcast_data %>%
  filter(pitcher %in% full_season$player_id) %>%
  filter(pitch_type %in% pitches) %>%
  group_by(pitcher) %>%
  mutate(avg_x = mean(pfx_x), avg_z = mean(pfx_z))
colnames(statcast_data)[9] <- "player_id"
statcast_data <- select(statcast_data, player_id, avg_x, avg_z)
statcast_data <- statcast_data %>% distinct(player_id, .keep_all = TRUE)
full_season <- left_join(full_season, statcast_data, by = "player_id")

# everything we're interested in for the lasso regression
baseball_trim <- full_season %>% select("woba", "BAbip", "spin_rate", "velocity", "whiff_rate",
                                        "WHIP", "FIP", "SO9", "release_extension", "launch_angle"
                                        "avg_x", "avg_z")
baseball_trim <- baseball_trim[complete.cases(baseball_trim), ]
# one variable we wish to estimate, the others are predictors. in this case, SO9 is what we wish
# to estimate, and we seek to do so using the other variables.
baseball_trim <- baseball_trim %>% select("SO9", "spin_rate", "velocity", "release_extension",
                                          "launch_angle", "avg_x", "avg_z", "BAbip")

baseball2 <- data.frame(scale(as.matrix(baseball_trim)))
B <- 100

# Matrices and vector storing results
lasso01mat <- c()
lassobetamat <- c()
mse.lasso <- c()

for (b in 1:B)
{
  # set.seed(b)
  print(paste("Iteration #", b, sep = ""))
  train <- sample(1:dim(baseball2)[1], dim(baseball2)[1] / 2)
  test <- -train
  baseball2.train <- baseball2[train, ]
  baseball2.test <- baseball2[test, ]

  train.mat <- model.matrix(SO9 ~ ., data = baseball2.train)[, -1]
  test.mat <- model.matrix(SO9 ~ ., data = baseball2.test)[, -1]

  # alpha = 1 corresponds to the lasso regression
  fit.lasso <- glmnet(train.mat, baseball2.train$SO9, alpha = 1)
  cv.lasso <- cv.glmnet(train.mat, baseball2.train$SO9, alpha = 1)
```

```
  bestlam.lasso <- cv.lasso$lambda.min
  bestlam.lasso

  pred.lasso <- predict(fit.lasso, s = bestlam.lasso, newx = test.mat)
  mse.lasso[b] <- mean((pred.lasso - baseball2.test$SO9)^2)

  # List of selected/dropped variables with their respective beta estimates
  beta.lasso <- predict(fit.lasso, s = bestlam.lasso, type = "coefficients")
  lassobetamat <- rbind(lassobetamat, beta.lasso[-1, 1])

  # Convert the variables to 0 (not selected) or 1 (selected)
  lasso01 <- as.integer(abs(beta.lasso[-1, 1]) > 0)
  lasso01mat <- rbind(lasso01mat, lasso01)
}
colnames(lasso01mat) <- colnames(baseball2)[-1]
colnames(lassobetamat) <- colnames(baseball2)[-1]
# Showing the probability of selecting each predictor for the Lasso
colMeans(lasso01mat)
# Showing the mean of the beta coefficients for each predictor for the Lasso
colMeans(lassobetamat)
# Mean of the MSE values
mean(mse.lasso)

#plots the residuals for ONE (the last) of the above simulated lasso regressions
#interpretation: the closer the dots are to 0 the better the model is
#pdf(file = "residuals.pdf", width = 4, height = 3)
plot((pred.lasso-baseball2.test$SO9), main = "Residuals vs. Fits (Lasso Regression)",
     ylab = "Fit", xlab = "Residuals")
abline(h=0)
#dev.off()


# Using model to predict the response variable for players in the first and latter halves of
# the season
# KINDA WORKS, KINDA DOESN'T. LASSO IS FIT ON SCALED DATA, SO COEFFICIENTS DON'T WORK EXACTLY
# 1:1 WITH THE REAL DATA, SO THIS JUST PROVIDES A ROUGH PICTURE OF HOW THE PREDICTED VARIABLE
# WOULD CHANGE WITH PLUGGED IN VALUES, NOT AN ACTUAL PREDICTION FOR SO9
# WE FIND THAT THE PREDICTED SO9 FOR PRE IS 598.8457, AND PREDICTED FOR POST IS 587.9489
# WHICH IS CONSISTENT WITH WHAT WE FOUND BEFORE, OUR LASSO REGRESSION ALSO INDICATES INCREASED
# PERFORMANCE FOR PITCHING MECHANICS FROM THE FIRST HALF OF THE SEASON COMPARED TO THE LATTER.
x <- colnames(lassobetamat)
y <- colMeans(lassobetamat)
model_df <- as.data.frame(rbind(x,y))

true_pre <- left_join(true_pre, statcast_data, by = "player_id")
true_pre <- true_pre %>% filter(!is.na(avg_x))
true_post <- left_join(true_post, statcast_data, by = "player_id")
true_post <- true_post %>% filter(!is.na(avg_x))

pred_pre <- double(nrow(true_pre))
for(i in 1:length(colnames(true_pre))){
 current <- colnames(true_pre)[i]
 if(current %in% model_df[1,]){
   x <- model_df[ , current]
   pred_pre <- pred_pre + (as.numeric(x[[2]])*true_pre[,current])
 }
}
#mean(pred_pre, na.rm=TRUE)

pred_post <- double(nrow(true_post))
for(i in 1:length(colnames(true_post))){
  current <- colnames(true_post)[i]
```

```
  if(current %in% model_df[1,]){
    x <- model_df[ , current]
    pred_post <- pred_post + (as.numeric(x[[2]])*true_post[,current])
  }
}
#mean(pred_post, na.rm=TRUE)

# average prediction for first half of season (pre-policy change)
mean(pred_pre, na.rm=TRUE)
# average prediction for latter half of season (post-policy change)
mean(pred_post, na.rm=TRUE)

###################
#Wilcoxon Tests
###################

# significant, p = 0.0125
sr_test <- true_pre$spin_rate - true_post$spin_rate
wilcox.test(sr_test, alternative = "greater")
# pre-avg = 2335.85
# mean(true_pre$spin_rate)
# post-avg = 2290.661
# mean(true_post$spin_rate)
# going off the dip in averages as well as the significant p-value, we can say that the sticky
# stuff ban resulted in a statistically significant dip in spin rate for the average mlb starter

# significant, p = 0.0001431
woba_test <- true_pre$woba - true_post$woba
wilcox.test(woba_test, alternative = "less")
# p-value of 0.0001431, indicates a significant difference in woba pre and post ban
# mean(true_pre$woba)
# mean(true_post$woba)
# higher avg. woba post ban indicates that there's a significantly lower woba pre ban,
# so we can conclude that pitchers were gaining a significant advantage pre ban.

#very significant, p = 7.107x10^-5
ba_test <- true_pre$ba - true_post$ba
wilcox.test(ba_test, alternative = "less")

# each of these tests gave us an insignificant p-value, some of which were surprising, some of
# which weren't as surprising
# not surprising, velocity isn't really impacted by improved grip p = 0.6159
velo_test <- true_pre$velocity - true_post$velocity
wilcox.test(velo_test, alternative = "greater")
# not surprising, release extension doesn't really have much impact p=0.4938
re_test <- true_pre$release_extension - true_post$release_extension
wilcox.test(re_test, alternative = "greater")
# not surprising, launch angle doesn't really depend on pitcher, but induced launch angle makes
# it make sense in the context of our lasso regression p = 0.9474
la_test <- true_pre$launch_angle - true_post$launch_angle
wilcox.test(la_test, alternative = "greater")
# avg_x: surprising p = 0.5607
ax_test <- true_pre$avg_x - true_post$avg_x
wilcox.test(ax_test, alternative = "greater")
# avg_z: surprising p = 0.4477
az_test <- true_pre$avg_z - true_post$avg_z
wilcox.test(az_test, alternative = "greater")
# not surprising, babip doesnt depend on pitcher p = 0.4815
babip_test <- true_pre$BAbip - true_post$BAbip
wilcox.test(babip_test, alternative = "greater")
# fip : surprising p = 0.3627
fip_test <- true_pre$FIP - true_post$FIP
```

```
wilcox.test(fip_test, alternative = "less")
# era: surprising p = 0.5698
era_test <- true_pre$ERA - true_post$ERA
wilcox.test(era_test, alternative = "less")
# whip: surprising p = 0.5214
whip_test <- true_pre$WHIP - true_post$WHIP
wilcox.test(whip_test, alternative = "less")
# SO9: surprising p = 0.693
so9_test <- true_pre$SO9 - true_post$SO9
wilcox.test(so9_test, alternative = "greater")

# CODE FOR GETTING BASEBALL SAVANT DATA FOR 2021 MLB SEASON
# YOU WILL GET AN ERROR MESSAGE AT THE END BUT IT'S FINE ALL THE DATA IS IN DATA
# fastball <- read.csv("full_season_big_data_fastball.csv", header = TRUE,
                       stringsAsFactors = FALSE)
# #fastballr <- statcast_search(start_date = "2021-04-01", end_date = "2021-04-01",
                               player_type = "pitcher")
#
# month <- c("04", "05", "06", "07", "08", "09", "10")
# days <- c(30, 31, 30, 31, 31, 30, 31)
# day <- c("01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12", "13", "14",
#          "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28",
#          "29", "30", "31")
#
# data <- data.frame(matrix(ncol = length(colnames(fastball)), nrow = 0))
# colnames(data) <- colnames(fastball)
#
# for(i in 1:length(month)){
#   current_month <- month[i]
#   num_days <- days[i]
#   for(j in 1:num_days){
#     date <- toString(paste0("2021-", current_month, "-", day[j]))
#     if(date == "2021-07-12" || date == "2021-07-13" || date == "2021-07-14" ||
#        date == "2021-07-15"){
#       print("allstar break woohoo")
#     } else{
#       new_data <- statcast_search(start_date = date, end_date = date, player_type = "pitcher")
#       data <- rbind(data, new_data)
#     }
#   }
# }
# write.csv(data, "full_season_statcast.csv")
```