Predicting the 2023 Seattle Mariners Wins and Playoff Odds - A Regression Approach

2023

Yi Kai Chen

Table of Content

1 Intro

Abstract

Dynamic Statcast databases of MLB empowers amateur baseball enthusiasts to better analyze baseball than ever before. As do many baseball analysts, we will utilize Baseball Savant's databases, and concepts such as multiple linear regression and logistic regression for prediction in this project. Then, we build multiple linear regression models to predict how many games will the Seattle Mariners win in the 2023 season. Moreover, logistic regression allows us to predict the probability of advancing to the postseason. Afterward, the model outcomes will tell us what's the potential weakness of the team and how can baseball management improve their recruitment of players. Noted that the datasets are collected from 2017~2022 season, but I excluded 2020 season, a shrinking season that will affect our data completeness.

Inspiration of the project

Inspired by one of the MIT's open learning platforms called "The analytics edge", I used similar approach in some proportion of the project. The approach is also similar to what the Oakland A's innovatively used to build their team in 2002, the story known as the Moneyball. Additionally, the Boston Red Sox utilized similar statistics approach to recruit their roster and eventually won the world series champion in 2004.

"When you have fun, it changes all the pressure into pleasure." – Ken Griffey. Jr, MLB Hall Of Famer, former Mariners.

The project is designed to be fun. We will input actual Mariners players' data to the model we built and then observe interesting implications. Besides, I also displayed my R code in the project and provided data visualizations. Thus, anyone who is appealed by the approach can also learn and design their own data analytic project. This project is for self-learning and for people who love baseball, so don't feel pressured. Let's begin our journey!!

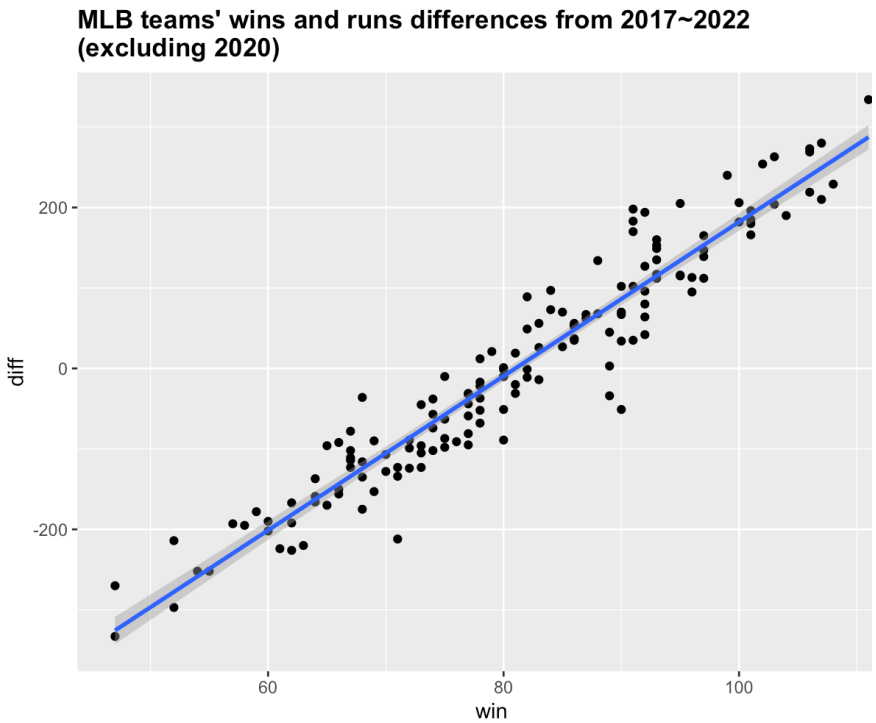
Logo of the Seattle Mariners


Pre-Season Prediction
Pre-Season Prediction
Pre-Season Prediction

2 How to win a ball game?

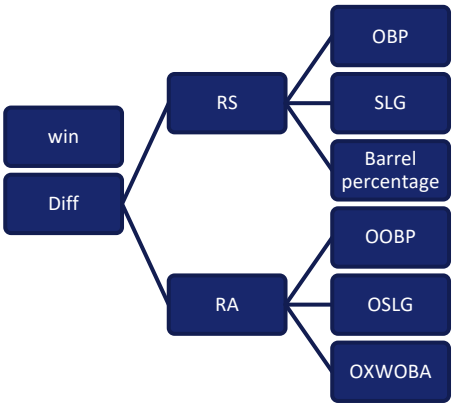
Run differential is the key.

To win a ball game, a team's runs scored(rs) must be more than their runs allowed(ra). The value between runs scored and runs allowed is called runs differential(rd). In fact, runs differential and wins have strong 0.959 linear correlation (as shown below), indicating that the more runs differential a team have the more ball games they win. Further, according to Moneyball's philosophy, if we want to predict runs difference in a season, we first need to predict its runs scored(rs) and runs allowed(ra).



Ability to attack = Run scored

Intuitively, runs scored(rs) indicates how well a team's batters can attack. Likewise, runs allowed(ra) showed how well its pitcher can handle the batters they faced. For runs scored, there are plentiful features collected by Statcast such as on-base percentage (OBP), slugging percentage (SLG), barrels percentage, xwOBA, launch angle, exit velocity and more, all measuring a team's ability to get on the base, how hard can they hit the ball and even scored. In our project, I used those features to build multiple linear regression, with runs scored as dependent variables and with OBP, SLG, Barrel percentage as independent variables.



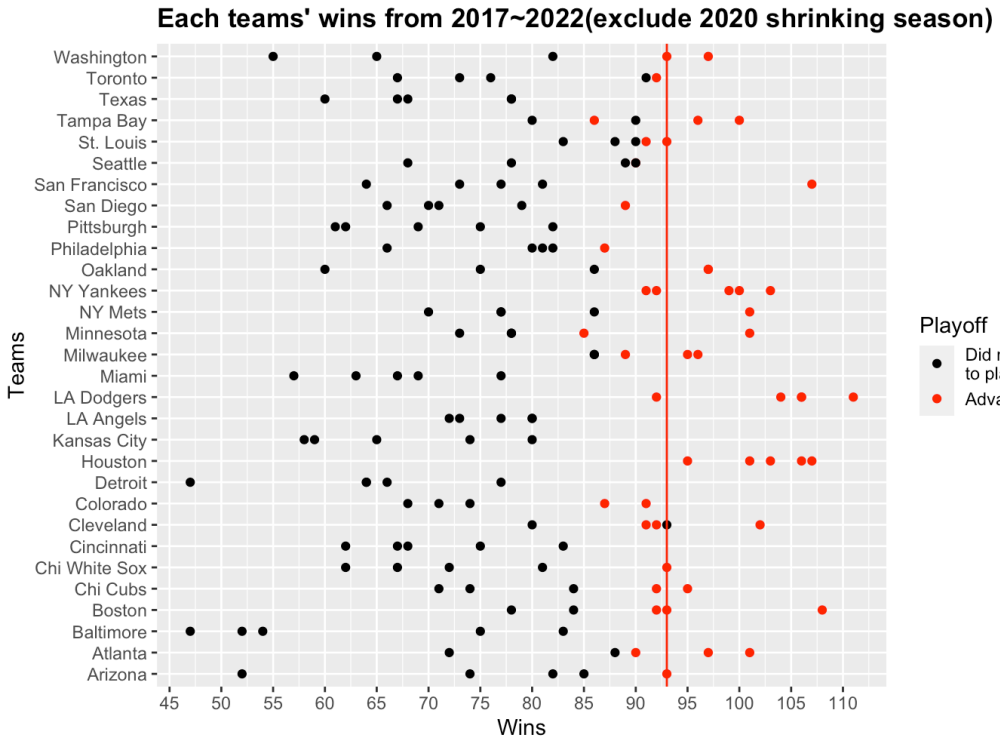
The relationship between each attribute.

3

To make the playoff, what's the threshold of winning?

At least 94 wins can guarantee a team for making it.

After knowing how to win ball games, we can better understand how many to win in order to get it to the playoff. To answer that question, I utilize MIT open resource's method. That is, as shown below, we visualize how many wins a team historically must get for entering the playoff. The red line shows that playoff teams generally won at least 94 wins in their regular seasons. In other words, if the mariners want to attend the playoff, winning at least 94 games can almost guaranteed them to do so.



Formula for Multiple Linear Regression

$$Y = a + b_1X_1 + b_2X_2$$

Above is multiple linear regression's formula. "Y" is the dependent variable, which is the thing we want to predict. "a" is the y intercept. "b_1" and "b_2" are the regression coefficient, and (b_1) is the first independent variable (X_1). If there is only one independent variable, it is called simple linear regression. For instance, in our case, if we want to predict wins, Y will be wins and we use runs differential as b_1X_1(as shown below) . Ex:

Win= a + b_1*(runs differential)

R will do the math!

By using R, we can get "a", *b_1(coefficient)* easily. Below is the example for this case, but don't worry I will explain the output later in the project.

The Mariners clinched the postseason after 21 years by winning 90 games in the 2022 season.

4 Basic baseball terms

To let you better understand the independent variables that we will use later in the projects, take a look at the below definitions. Note: Definitions are gathered from MLB's website.

OBP: On-base percentage refers to the frequency of a player reaches base. Players can get on base through walks, hits and hit by pitch. The MLB average OBP is roughly 0.32. It is calculated as below:

$$OBP = \frac{H + BB + HBP}{AB + BB + HBP + SF}$$

SLG: Slugging percentage is calculated as total bases divided by at bats, through the following formula, where *AB* is the number of at bats for a given player, and *1B*, *2B*, *3B*, and *HR* are the number of singles, doubles, triples, and home runs, respectively:

$$\text{SLG} = \frac{(1B) + (2 \times 2B) + (3 \times 3B) + (4 \times HR)}{AB}$$

It gives more weight to extra-base hits such as doubles and home runs, relative to singles. Generally, player with 0.45 of SLG will be considered good and above 0.55 is considered outstanding.

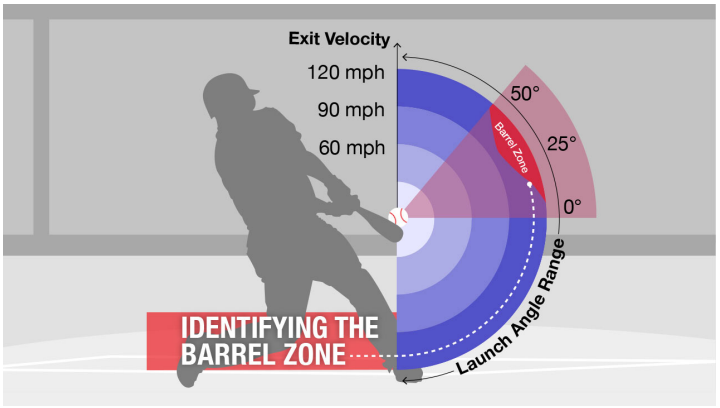
xwOBA: Expected Weighted On-base Average (xwOBA) is formulated using exit velocity(how hard the batter hits the ball), launch angle and, on certain types of batted balls, Sprint Speed. Below is the formula:

xwOBA = (xwOBAcon + wBB x (BB-IBB) + wHBP x HBP)/(AB + BB — IBB + SF + HBP)

For xwOBAcon, con means contact. In short, xwOBA is the most important statistic in evaluating a player's overall expected contribution on offense. The MLB average of xwOBA in 2022 season is 0.309 and an elite batter generally have a xwOBA higher than 0.4.

In the 2022 season, Julio Rodriguez has 0.345 OBP and 0.509 SLG, which are both the highest on the team.

Barrel: A Barrel is a batted ball whose comparable hit types (in terms of exit velocity and launch angle) have led to a minimum .500 batting average and 1.500 slugging percentage.



The red-shaded area in the graphic above illustrates the exit velocity and launch angle combinations that yield Barreled status in 100 percent of cases.

5 Runs Scored Prediction

Exploring Correlation

To predict anything, we first want to have a clear view of the correlation between runs scored(rs) and other independent variables. As shown below, OBP, SLG, WOBA, XWOBA all have approximately 90% of correlation with rs, which means they significantly affect run scored and are good for our prediction. However, when independent variables have too high correlation among each other, we will encounter a common problem that is called multicollinearity

Choosing independent variables

The main reason we have multicollinearity is that WOBA and XWOBA are both calculated based on OBP and SLG, so they all have high correlation, which is approximately 0.8.

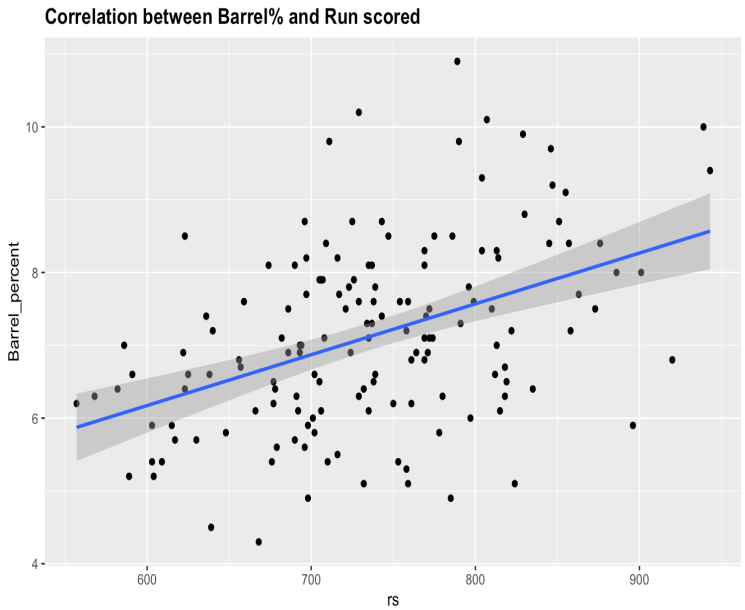
	rs
rs	1.0000000
ra	-0.3278344
diff	0.7813243
playoff	0.6037962
HR	0.7951342
BA	0.7225370
OBP	0.8841453
SLG	0.9276723
WOBA	0.9390875
WOBACON	0.7993299
Barrels	0.5481148
Barrel_percent	0.4345811
Hard_Hit_percent	0.3200827
Exit_Velocity	0.4038926
Launch_Angle	0.2958865
XBA	0.6388098
XSLG	0.7960603
XWOBA	0.8307494
XWOBACON	0.6574413

Problem of multicollinearity

When two or more independent variables in a multiple regression model have high intercorrelations, multicollinearity will happen. With correlation above 0.7, it might cause analysts to misinterpret the model's outcome. For example, when OBP is already in the model, adding xwOBA won't contribute too much to the determination of runs scored because both independent variables are already similar. This also inflates the standard errors of some or all of the regression coefficients.

Why do I choose barrel%?

In our baseball analytics case, we avoid multicollinearity by choosing barrel%. Barrel has only 0.4 of correlation with both OBP and SLG. Additionally, we have 0.91 of adjusted R-squared when adding barrel% instead of XWOBA, which is better than the one using XWOBA. Besides, barrel% is better than barrels, since barrel% is a probability form and ignore the factor of player appearance. Player appearance sometimes underestimates a player's performance. Barrel% also have moderate linear relationship with run scored if we exclude outliers.



6 Statistical Meaning of Regression Model

Do we get the right variables for the model?

```
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     -754.513     52.743 -14.305  < 2e-16 ***
OBP             2553.352    280.700   9.096 6.31e-16 ***
SLG             1513.628    141.106  10.727  < 2e-16 ***
Barrel_percent     6.962      1.768   3.939 0.000127 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’

Residual standard error: 23.92 on 146 degrees of freedom
Multiple R-squared:  0.9118,    Adjusted R-squared:   0.91
F-statistic: 503.3 on 3 and 146 DF,  p-value: < 2.2e-16
```

Above is the summary of runs scored linear regression model after calculating by R. First, at the upper section, I input rs as dependent variable with OBP, SLG and Barrel% as independent variables. Second, we can see that each independent variables have three stars at the right-hand side. The more stars there are, the more influential and significant these attributes are to the model and has lower p-value. Thus, we get variables that matter the most.

Coefficients and R-squared

The “Estimate” column shows the coefficients of each attribute. With positive coefficient, the higher the attributes’ values we input to the equation, the higher the prediction outcome will be. Lastly, the Adjusted R-squared section tells us how well a set of predictor variables is able to explain the variation in the dependent variable. Value closer to 1 is better than value is closer to 0. In this case, we get 0.91, indicating that this is a good model.

The equation for RS

Eventually, we can get the equation as shown below, in which we will further input mariners’ data to predict runs scored for 2023 season.

Rs=-754.51+2553.35(OBP) +1513.63*(SLG)+696.24*(Barrel%)*

Confidence Interval of coefficients

In terms of confidence interval, below is the value we get by using R and all values are within acceptable ranges. For example, we have 95% of confidence saying that the coefficients of SLG will be within 1792 and 1234. In other words, if the actual value of inputted OBP data is increased by 0.1, the runs scored outcome will increase in the range between 179.2 and 123.4, which is quite reasonable.

```
                    2.5 %     97.5 %
(Intercept)     -858.751759 -650.27327
OBP             1998.591958 3108.11168
SLG             1234.754789 1792.50205
Barrel_percent     3.468686   10.45616
```

For runs allowed prediction, we will do the exact same calculation as shown above with only different independent variables. Thus, we will eventually have two equations for predicting both run scored and run allowed, which will help us to get 2023 run differential and for predicting the number of wins.

7 Input Real Life Mariners Batters Data

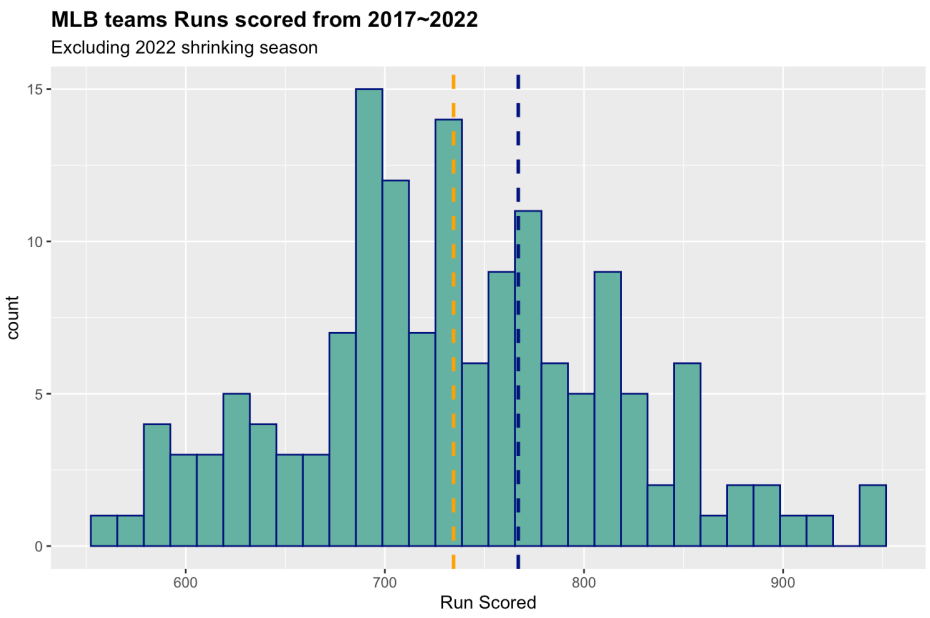
I predict that the mariners will score 767 points in the 2023 season.

Linear Regression Equation: Rs=-754.51+2553.35(OBP) +1513.63*(SLG)+696.24*(Barrel%)*
With the above equation, we will input 2022 mariners' data to the model and predict the runs scored for 2023. I calculated weighted OBP, weighted SLG and weight barrel% by using 2022 at bats as weighting factor. With weighting factor, prediction will be more accurate because we add the factor of players' appearing frequency. As shown below, the average weighted values at the bottom of each yellow columns are the data we input. Thus, we get the following calculation:
Rs=-754.51+2553.35*0.318 +1513.63*0.424+696.24*0.096. Then we get **RS= 767.0942**

Player	obp	at bats	weighting factor	weighted_obp	slg	weighted_SLG	barrel_per	weighted_barrel_per
Murphy, Tom	0.439	33	0.006672058	0.002929034	0.455	0.003035786	0.2	0.001334412
Hernández, Teoscar	0.315	499	0.100889608	0.031780226	0.491	0.049536797	0.15	0.015133441
Raleigh, Cal	0.284	370	0.074807926	0.021245451	0.489	0.036581076	0.154	0.011520421
Rodríguez, Julio	0.34	511	0.103315811	0.035127376	0.509	0.052587748	0.131	0.013534371
Suárez, Eugenio	0.332	543	0.109785685	0.036448848	0.459	0.05039163	0.148	0.016248281
Pollock, AJ	0.292	489	0.098867772	0.028869389	0.389	0.038459563	0.094	0.009293571
France, Ty	0.335	551	0.111403154	0.037320057	0.436	0.048571775	0.054	0.00601577
Moore, Dylan	0.363	205	0.041447634	0.015045491	0.385	0.015957339	0.129	0.005346745
Trammell, Taylor	0.284	102	0.020622725	0.005856854	0.402	0.008290336	0.099	0.00204165
Wong, Kolten	0.337	430	0.086938941	0.029298423	0.43	0.037383744	0.054	0.004694703
Kelenic, Jarred	0.221	163	0.032955924	0.007283259	0.313	0.010315204	0.136	0.004482006
Crawford, J.P.	0.339	518	0.104731096	0.035503841	0.336	0.035189648	0.02	0.002094622
Hummel, Cooper	0.274	176	0.035584311	0.009750101	0.307	0.010924383	0.071	0.002526486
Haggerty, Sam	0.325	176	0.035584311	0.011564901	0.403	0.014340477	0.04	0.001423372
La Stella, Tommy	0.282	180	0.036393045	0.010262839	0.35	0.012737566	0.002	7.27861E-05
avg	0.31747	4946	1	0.31828609	0.41027	0.424303073	0.0988	0.095762636

Comparing to MLB teams' median

On your left, to give you a better concept, I compare MLB teams' run scored median, which is 734.5(orange line), to our prediction for the mariners this year, 767.09(navy blue line). As the result, we predict that mariners attacking ability in 2023 will be better than league's median.

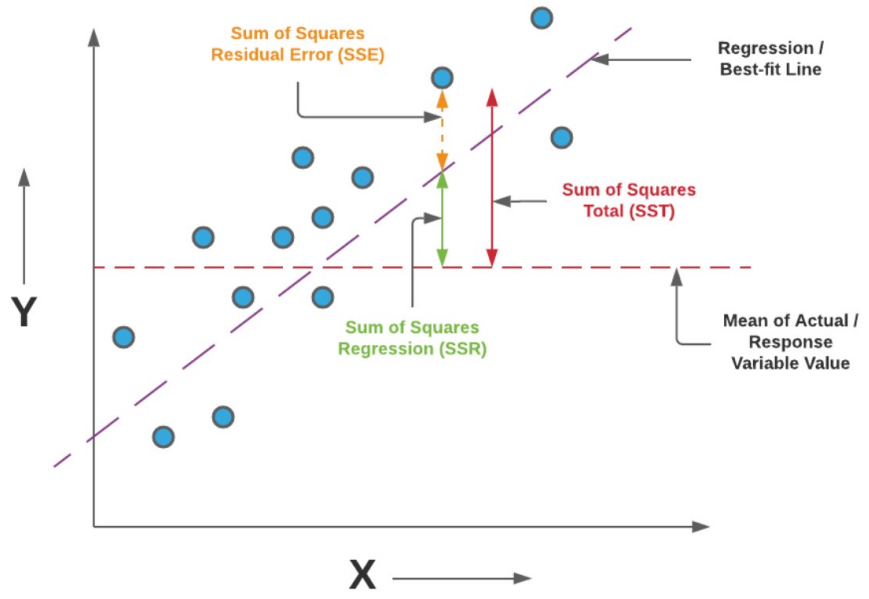


8 Model Diagnostic - Error of the predicting model

Sum of squared errors (SSE)

The difference between predicted value and observed value is called sum of squared errors or sse. The smaller the sse is the more accurate the prediction is. The example figure below visualizes SSE to you. To understand how good our model can predict RS, I used R to calculate the SSE. I then get the number 83507.72. This number isn't intuitive enough and can hardly give us a good insight of whether there is a large gap between the predicted and observed values. Therefore, I then use the **root mean square error (RMSE).**

$$SSE = \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$



SSE plot example (Source: Data Analytics)

Explanation of the SSE plot

For each point, there is a residual between the point and the regression line. We want to add those values up and measure whether we can accept the added value or is there a better regression model that can provide lower SSE.

Average error between actual runs scored and the predicated run scored is 23.59.

Root-mean-square error (RMSE)

The difference between SSE and RMSE is that RMSE is the square root of average SSE. It provides us with a better view of the error in our model. After calculating it through R with the equation below, I find out that the square root and average of 83507.72 is 23.59487.

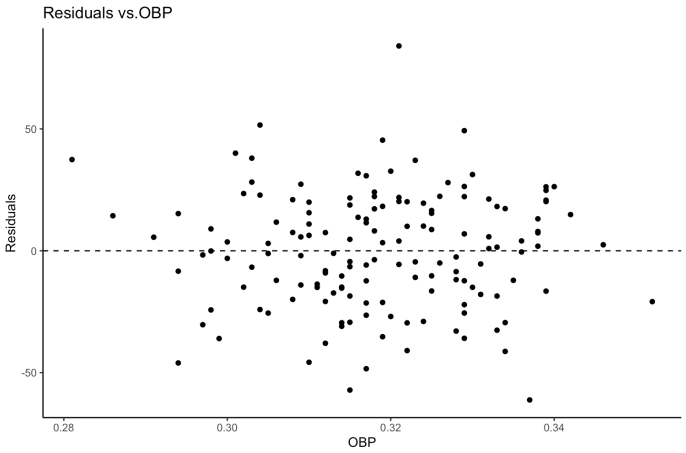
$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

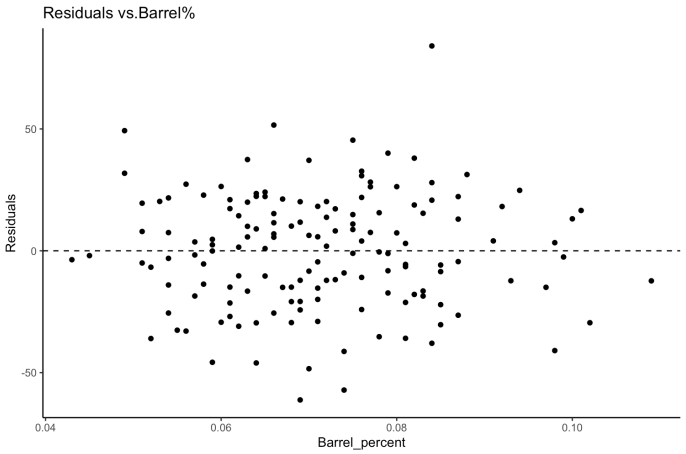
We can then interpret the number 23.59 as the average error between actual runs scored and the predicated run scored, which is acceptable given that a team generally score 600~900 points in a season and the runs scored of MLB league median is 734.5 points. In other words, we might only have 23 points error by predicting through this model.

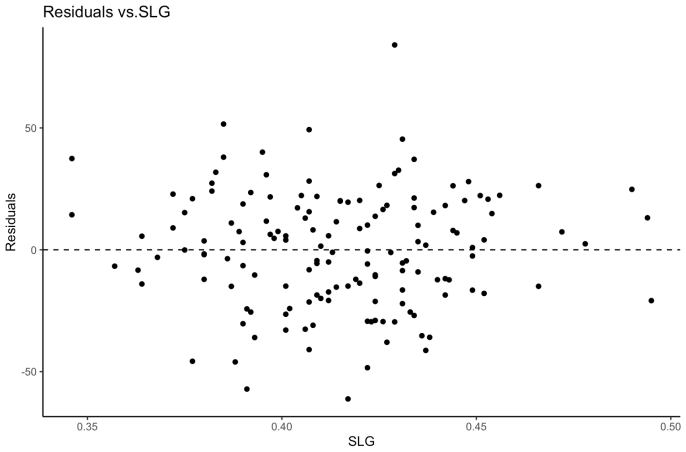
9 Model Diagnostic -Residual Analysis

(1) Linearity

First, as we already checked whether the relationship between rs and OBP, SLG and Barrel% are linear by using scatterplots, we should also verify this condition with plots of the residuals vs. OBP, residuals vs. SLG and residuals vs. Barrel%. After plotting, I found out that there is no apparent pattern in the residuals plot, which is a good sign for our linear model.

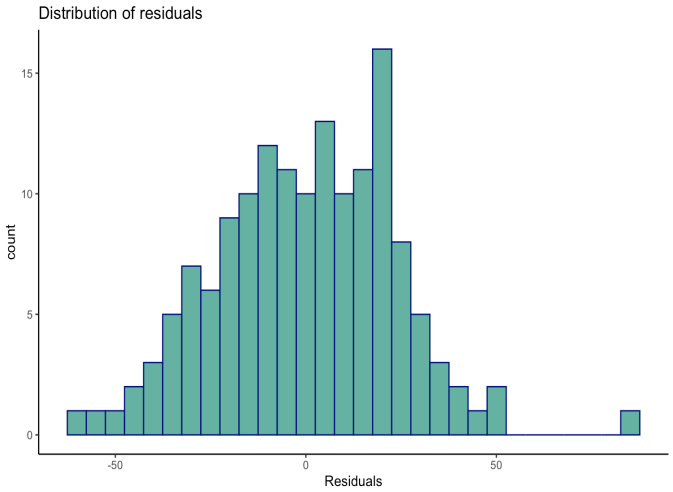






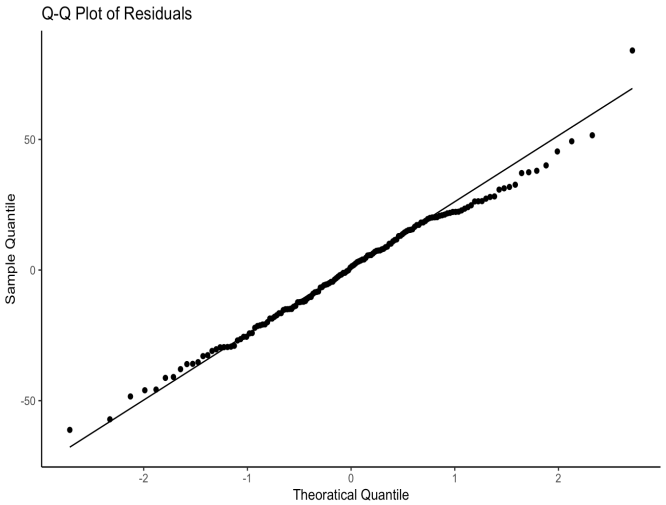
(2) Nearly Normal Residuals

Second, we can also use histogram to show the distribution of residual. After plotting, I found out that the residual has normal distribution, indicating that there is no skewness and that our model is good for linear modeling.



(3) Normal Q-Q Plot

Lastly, QQ plot (Qunatile Quantile plot) is a scatterplot for comparing two probability distributions by plotting their quantiles against each other. If the distributions are linearly related, the points in the Q–Q plot will approximately lie on a line. In our case, The qq plot nearly form a line, showing that the residuals are distributed normally.



10 Runs Allowed Prediction

Exploring Correlation

Likewise, for runs allowed, we have the exact same analysis procedure as we did for runs scored prediction. I choose oOBP(Opponent OBP), oSLG(Opponent SLG) and oxwOBA(Opponent xwOBA). You might wonder why we don't choose opponent barrel% for runs allowed prediction just as we select barrel% for runs scored prediction. The main reason is that the correlation between ra and opponent barrel% is only 0.36, which is low (as shown below). Thus, I instead pick oxwOBA as one of the independent variables. You may also question about the problem of multicollinearity, which I explain at your right.

	ra
ra	1.00000000
HR.g	0.75839873
OBA	0.89776791
OOBP	0.91879476
OSLG	0.93967050
OWOBA	0.95851177
OWOBACON	0.82646118
OBarrels	0.57625200
OBarrel_percent	0.36468572
OHard.Hit_percent	0.29672405
OExit.Velocity	0.37207984
OLaunch.Angle	0.09109807
OXWOBA	0.85947693
OXSLG	0.82238170
OXBA	0.79111578
OXWOBACON	0.69119377
WHIP	-0.10316838

Reality of multicollinearity

In reality, decision makers rarely avoid multicollinearity since it is difficult. In our case, we can only select OOBP, OSLG and OXWOBA simultaneously because all other attributes have either low correlation with runs allowed or was already calculated in oxwOBA. To illustrate, I don't pick owOBA, owOBACON, oxSLG and oxwBA as independent variables since they are already all highly associate with oxwOBA. As we mentioned before, oxwOBA was calculated based on the following:

oxwOBA = (xwOBAcon + wBB x (BB-IBB) + wHBP x HBP)/(AB + BB — IBB + SF + HBP)

Thus, owOBA, owOBACON, oxSLG and oxWBA are already part of oxwOBA. Their correlations are shown below.

	OXWOBA
ra	0.8594769
HR.g	0.6388454
OBA	0.8448740
OOBP	0.8769034
OSLG	0.8320065
OXWOBA	1.0000000
OWOBA	0.8897813
OWOBACON	0.6092123
OBarrels	0.6793508
OBarrel_percent	0.4628505
OHard.Hit_percent	0.4161628
OExit.Velocity	0.4947748
OLaunch.Angle	0.1171995
OXSLG	0.9412356
OXBA	0.9503030
OXWOBACON	0.8076344
WHIP	-0.1236550

11 Statistical Meaning of Regression Model

Summary of Model

Likewise, for runs allowed, we have the model with 0.931 adjusted R-squared value. oxwOBA have high p-value because of the multicollinearity we mentioned before. In addition, we get a rmse of 24.17, which is also an acceptable error.

```
Call:
lm(formula = ra ~ OXWOBA + OOBP + OSLG, data = pitcher)

Residuals:
    Min      1Q  Median      3Q     Max
-59.718 -17.691   2.508  16.548  58.289

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -734.5       44.3 -16.581  < 2e-16 ***
OXWOBA         203.4      284.6   0.715    0.476
OOBP          2232.9      278.4   8.020 3.14e-13 ***
OSLG          1677.2      131.2  12.782  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 24.5 on 146 degrees of freedom
Multiple R-squared:  0.9324,    Adjusted R-squared:  0.931
F-statistic:   671 on 3 and 146 DF,  p-value: < 2.2e-16
```

```
> confint(pitcherreg)
                2.5 %    97.5 %
(Intercept) -822.0846 -646.9852
OXWOBA      -358.9535  765.8276
OOBP        1682.7032 2783.1583
OSLG        1417.8770 1936.5504
```

Input Mariners Pitchers Data

As shown below, I also calculated each variable by using pitching innings as weighting factor, which shows a pitcher's appearing frequency. After viewing the summary, we get reasonable confidence interval and the following equation:

Rs=-734.5+203.4*(oxwOBA)+2232.9*(OOBP)+1677.2*(OSLG)

Then, I input the average weighted values at the bottom of each yellow columns. Thus, we get the following calculation:

Rs=-734.5+203.4*0.2649 +2232.9*0.2918+1677.2*0.3791

Then we get **Predicted Runs Allowed= 606.8619**

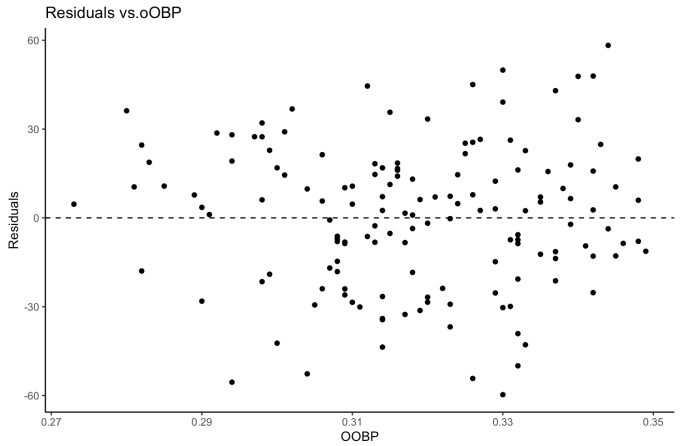
Pitchers Table:

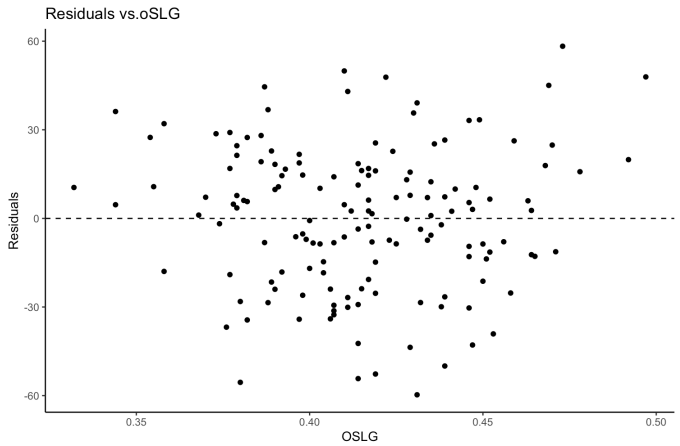
Player	oslg	innings	weighting factor	weighted_OSLG	oobp	weighted_OOBP	oxwoba	weighted_OXWOBA
Brash, Matt	0.328	50.2	0.037226548	0.012210308	0.364	0.013550463	0.237	0.008822692
Castillo, Diego	0.299	54.1	0.04011865	0.011995476	0.275	0.011032629	0.269	0.010791917
Castillo, Luis	0.332	150.1	0.111308862	0.036954542	0.286	0.031834334	0.156	0.017364182
Flexen, Chris	0.408	137.2	0.101742677	0.041511012	0.312	0.031743715	0.331	0.033676826
Gilbert, Logan	0.39	185.2	0.137337783	0.053561735	0.293	0.04023997	0.237	0.032549055
Gonzales, Marc	0.457	183	0.13570634	0.062017798	0.321	0.043561735	0.33	0.044783092
Gott, Trevor	0.392	45.2	0.033518725	0.01313934	0.273	0.009150612	0.269	0.009016537
Kirby, George	0.393	130.2	0.096551724	0.037944828	0.299	0.028868966	0.332	0.032055172
Munoz, Andres	0.273	65	0.048201706	0.013159066	0.241	0.011616611	0.211	0.01017056
Murfee, Penn	0.32	69.1	0.051242121	0.016397479	0.247	0.012656804	0.303	0.015526363
Ray, Robbie	0.423	189	0.140155729	0.059285873	0.299	0.041906563	0.254	0.035599555
Sewald, Paul	0.301	64	0.047460141	0.014285502	0.21	0.00996663	0.197	0.009349648
Speier, Gabe	0.352	19.1	0.014163886	0.004985688	0.269	0.003810085	0.26	0.00368261
Topa, Justin	0.323	7.1	0.005265109	0.00170063	0.353	0.001858584	0.303	0.001595328
avg	0.3565	1348.5	1	0.379149277	0.28871	0.291797701	0.2635	0.264983537

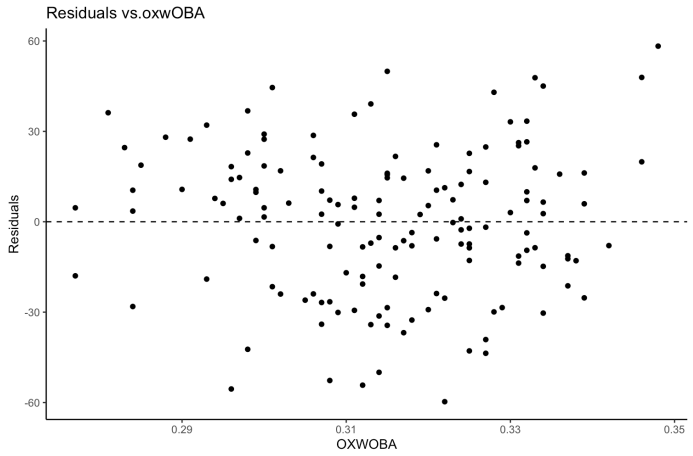
12 Model Diagnostic-Residual Analysis

(1) Linearity

Like residual analysis for RS, there is no apparent pattern in the RA residuals plot, which is a good sign for our linear model.

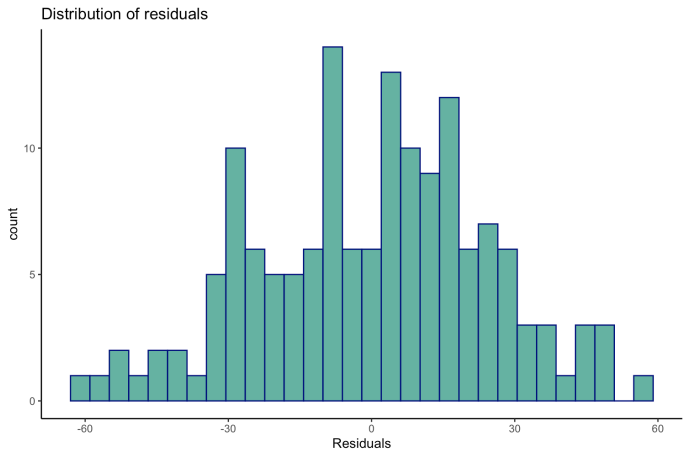






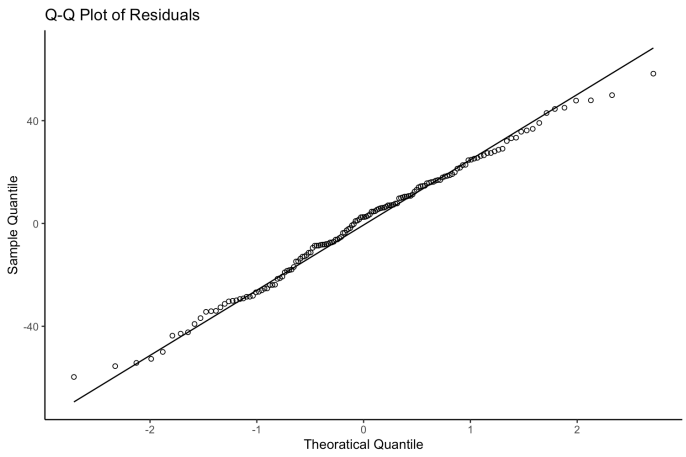
(2) Nearly Normal Residuals

Second, we can use histogram to show the distribution of RA's residual. After plotting, I found out that the residual has normal distribution, indicating that there is no skewness and that our model is good for linear modeling.



(3) Normal Q-Q Plot

Lastly, QQ plot for RA's residual shows that the distributions are linearly related and the points in the Q–Q plot approximately lie on a line, showing that the residuals are distributed normally.



13 Let's predict wins!!

Predicted Run Differential=160.2323

By subtracting 606.8619(RA prediction) from 767.0942(RS prediction), we get 160.2323, which will be our run differential prediction.

Below is the summary of win and runs differential's formula. Based on the summary, we can also get the following formula.

WIN=80.933+0.096*(diff)

```
Call:
lm(formula = win ~ diff, data = baseball)

Residuals:
    Min      1Q  Median      3Q     Max
-9.5345 -2.7124 -0.0139  2.6929 13.9067

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 80.993333   0.327237  247.51   <2e-16 ***
diff         0.096078   0.002326   41.31   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.008 on 148 degrees of freedom
Multiple R-squared:  0.9202,    Adjusted R-squared:  0.9197
F-statistic:  1707 on 1 and 148 DF,  p-value: < 2.2e-16
```

Plug in Run Differential to the model

Then, we plug 160.2323 to the above formula. Finally, we get the outcome, which is 96.38813, indicating that the mariners were predicted to win **96 games**, or 59% of winning probability, in the 2023 season!! Last year, they won 90 games, so this looks like a great improvement.

Comparing prediction with the performance of 2022

As the following table shows, we see that both run scored and runs allowed are predicted to be improved. Part of the reason why runs scored improves so much might be due to the signing of Teoscar Hernandez and AJ Pollock, both have good OBP, SLG and barrel%. Moreover, they have 499 and 489 at bats, respectively, which contribute a lot to the weighting factors.

	RS	RA	RD	Wins
2022	690	623	67	90
Prediction	**767**	**606**	**160**	**96**

Conclusion and Limitation

There are limitations in our model. For example, the players data we inputted assume that in 2023 those players will perform similarly as 2022, won't injured and won't be replaced by others throughout the season. Otherwise, the prediction will be inaccurate. But this model still gives us a great insight for preseason prediction. In conclusion, the Seattle Mariners began a competitive new era of their franchise since last year. As I mentioned previously, a team with more than 94 wins generally advance to the playoff, so, with 96 predicted wins, the mariners might appear to it again this year!!

Mariners' slogan of season

Some Numbers for ....

The summary of predicting Mariners' game wins with multiple linear regression

767
Predicted Runs Scored

606
Predicted Runs Allowed

160.2323
Predicted Runs Differential

96
Predicted Wins for 2023 Season

0.91
Adjusted R-Squared for Runs Scored Prediction

0.931
Adjusted R-Squared for Runs Allowed Prediction

23.59
RMSE for Runs scored

S

Mid-Season Prediction

PREDICTING WITH LOGISTIC REGRESSION

What will we do in this session?

- **Learn Logistic Regression**
- **Use On-going Data to predict**
- **Predict chance to make to the postseason**
- **Learn how the team can improve**

How will we use Logistic Regression?

Logistic regression is categorized as supervised machine learning, which allows us to predict whether an event will happen or not. In our case, suppose that we are at the mid-season, and we are part of the mariners' management team. We want to predict whether our team will clinch the postseason based on the team's current performance. To illustrate we will use both runs scored and runs allowed as independent variables and whether a team making playoff as dependent variable.

Formula of Logistic Regression

a.k.a. **Log Odds**

or **Logit**

Intercept

$$\log\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 X$$

Source: QUANTIFYING HEALTH

P is the probability of having the outcome and (P / (1-P)) is the odds of the outcome. When X = 0, the intercept β0 is the log of the odds of having the outcome. If we exponentiate both sides, we get the following:

$$P = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

When X = 0, the probability of having the outcome is $P = e^{\beta_0} / (1 + e^{\beta_0})$. If the intercept, e^{β_0}, has a **positive sign**, the probability of having the outcome will be > 0.5.

Logistic Regression

Using RS and RA to make Logistic Regression

In this session, we will utilize both "Run scored per game" and "Runs allowed per game" as independent variables.

Data

The data I used to build models are shown below, which shows the MLB teams' rs/game and ra/game from 2017~2022. Totally, there will be 150 rows. Each row represents a team's performance of that year. If they advance to the playoff, they were noted with 1 in the playoff column and vice versa.

	rs_g	ra_g	diff_g	playoff
1	4.333333	4.567901	-0.23456790	0
2	4.191358	5.512346	-1.32098765	0
3	5.012346	4.067901	0.94444444	1
4	5.018519	4.586420	0.43209876	0
5	4.277778	3.975309	0.30246914	0
6	4.870370	3.759259	1.11111111	1
7	4.876543	4.049383	0.82716049	0
8	4.518519	5.067901	-0.54938272	0
9	5.277778	4.586420	0.69135802	1
10	4.685185	4.055556	0.62962963	1
11	4.160494	4.246914	-0.08641975	0
12	4.067901	5.901235	-1.83333333	0
13	4.586420	5.191358	-0.60493827	0
14	4.500000	6.055556	-1.55555556	0
15	3.839506	5.506173	-1.66666667	0
16	4.537037	4.858025	-0.32098765	0
17	5.117284	4.623457	0.49382716	1
18	4.845679	4.123457	0.72222222	1
19	5.561728	5.111111	0.45061728	0
20	5.407407	3.993827	1.41358025	1
21	4.055556	4.512346	-0.45679012	0

Data sample(Total: 150 rows)

We will split this dataset to **train data and test data** by using R's caTools function, with 0.67 of split ratio. That said, we will train data with 150*0.67 equals 100 rows as training data. The reason for using 0.67 will be explained later.

```
w1=select(win2,rs_g,ra_g,diff_g,playoff)
str(w1)
library(caTools)
set.seed(88)
split=sample.split(w1$playoff,SplitRatio = 0.67)
train=subset(w1,split==TRUE)
test=subset(w1,split==FALSE)
nrow(train)
```

Building Model

We use R's glm(generalize linear model) function to establish the model, with playoff as our dependent variable.

```
logit2<- glm(playoff~rs_g+ra_g, data=train,family='binomial')
summary(logit2)
```

Then, we get the following result that is similar to how linear regression works. RS/game and RA/game significantly affect whether a team can advance to playoff because they are both noted with three stars. If ra/g increase, probability of making playoff decrease because its coefficient is negative.

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.785      5.521  -0.142 0.886928
rs_g           4.615      1.161   3.975 7.04e-05 ***
ra_g          -4.907      1.270  -3.864 0.000112 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 129.033  on 100  degrees of freedom
Residual deviance:  50.574  on  98  degrees of freedom
AIC: 56.574

Number of Fisher Scoring iterations: 7
```

With following code, we can use the above model to predict.

```
predict2=predict(logit2,type='response')
summary(predict2)
tapply(predict2, train$playoff,mean)
```

Outcome:

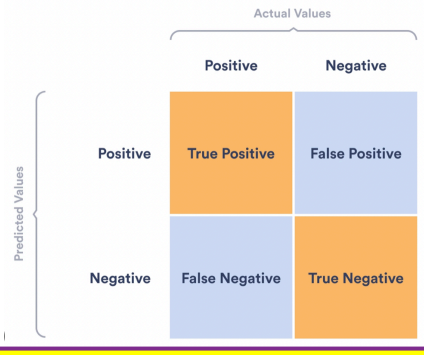
```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
0.0000066 0.0085501 0.1126951 0.3366337 0.7163669 0.9995921
```

Above shows prediction in different quantile, and below is the average prediction of true cases. Our case is good because we predict making playoff, 0.766, with higher probability.

```
        0         1
0.1185752 0.7663371
```

Confusion Matrix

Confusion matrix allows us to know the accuracy of our model, such as false positive rate etc.



Model Accuracy And Thresholding

Model Accuracy

We will then input the testing data to this model for prediction and use the result to get the confusion matrix, with 0.4 threshold value. The code is shown at below:

```
table(train$playoff, predict2>0.4)
```

As the result, we get the following matrix:

```
   FALSE TRUE
0     61    6
1      5   29
```

The accuracy of the model is (61+29)/(61+6+5+29)= 0.8653846

Then, we compare the prediction's accuracy with the baseline model. Baseline models calculates the average outcome of all data point and the most frequent outcome. In our case, the most frequent outcome is that teams are not advancing to playoff, which is denoted as 0 at below. Thus, our baseline model is calculated as 100/(100+50)=0.67, which is also our split ratio used in the previous page. Thus, our prediction's accuracy, 0.86, is better than that of baseline model, 0.67, and this is a good sign.

To make testing set be representative, 0.67 of split ratio allows us to get the training set with 67% of teams didn't get to the playoff and make it representative to the true situation.

```
  0   1
100  50
```

Table for Baseline Model

Sensitivity and Specificity

No matter how good the model is, the prediction will make some errors. For example, when we predict that a team will make to the playoff, but it actually didn't(False positive). In fact, different threshold value will generate different true positive rate. So, when choosing threshold value, we dependents on how much error we are willing to accept. To know those errors, we measure by using sensitivity and specificity.

Sensitivity(true positive rate) is the ability of a test to correctly identify teams making to playoff, which is cases of true positive divided by true positive plus false negative. Likewise, the **specificity**(true negative rate) is the ability of a test to correctly identify teams that didn't make to the playoff.
The formula is shown at below:
Sensitivity=TP/(TP+FN)
Specificity=TN/(TN+FP)

Typically, when threshold value is lower, sensitivity will increase. When threshold is larger, specificity will go up. But what if you don't have a preference in higher sensitivity(true positive rate) or the other one? You can choose 0.5 as the threshold value. But in our model, I prefer higher true positive rate. Thus, I use 0.4 threshold value and use the previous confusion matrix to calculate as below:

- Sensitivity=(29)/(5+29)= **0.8529412**
- Specificity=(61)/(61+6)= **0.9104478**

New logo on the City Connect jersey: Pacific Northwest(PNW)

ROC Curve

Why ROC Curve?

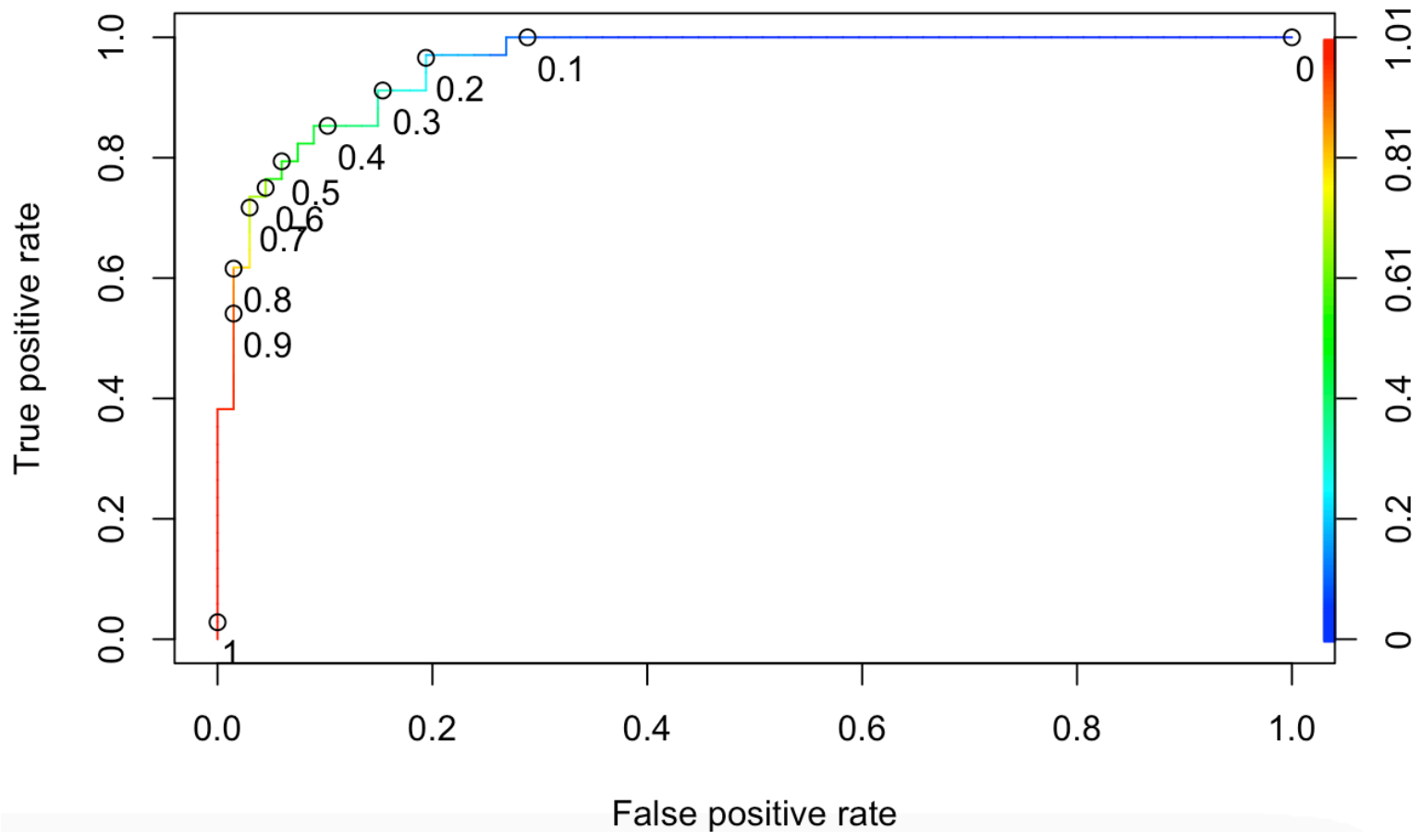
It helps us to select *threshold value* and further the sensitivity or specificity we prefer.

Receiver Operating Characteristic(ROC) curve is typically plotted by true positive rate and false positive rate. It allows us to visualize the relation among those rates and threshold values. Hence, we can better choose threshold value based on our preference on the type of false positive rate we can accept. Below is the code for plotting ROC curve and its explanation:

R Code(With Colorization):

```
library(ROCR)
roc=prediction(predict2,train$playoff)
perf=performance(roc,'tpr','fpr')
plot(perf)
plot(perf,colorize=TRUE)
plot(perf,colorize=TRUE, print.cutoffs.at=seq(0,1,0.1),text.adj=c(-0.2,1.7))
```

Outcome:



Interpretation of ROC Curve

Interpretation of The Plot

As we mentioned previously, when threshold value is lower, sensitivity will increase. When threshold is larger, specificity will go up. Thus, in our plot above, if you prefer high sensitivity, you can choose 0.3 as your threshold value because it has 0.9 of true positive rate.

Likewise, if you like lower false positive rate, then 0.7 of threshold value will be great since it only has 0.1 of false positive rate. The color legend at the right tells us the threshold value in a similar way.

AUC(Area Under Curve)

The area under ROC curve is called AUC. More importantly, the closer AUC is to 1, the better the model's ability to distinguish between making playoff or don't make playoff, and also the better its predicting ability. In our case, we have a great predicting model since the AUC is **0.9569** by running the following R code:

```
as.numeric(performance(roc,'auc')@y.values)
```

Inputting Live Data for 2022 season

What's the chances of getting to the playoff based on current performance?

After building the model, we can input the Mariners' current rs/game and ra/game into it. By doing so, we can know their probability of getting to the playoff, how can they improve to reach the standard of making playoff and learn to increase their probability. The scale of this model is not just limited for the Mariners, but it can also predict 29 other teams' probabilities when you collect each of their own's rs/game and ra/game data. If the probability is greater than 0.5, I consider that a team is predicted to make the playoff. Before we test 2023's probability, I first wish to test the accuracy of the model by inputting MLB teams' 2022 data to it. Then, I compared that predicted probability to whether the team actually made the playoff in 2022, thereby confirming our model's accuracy. The process is shown below:

1. Data of 2022 season(RS/Game, RA/Game)

Note: team IDs with yellow color are inaccurate prediction that will be explained later.

	Team	rs_g	ra_g
1	Arizona	4.33	4.57
2	Atlanta	4.83	3.81
3	Baltimore	4.16	4.25
4	Boston	4.54	4.86
5	Chi Cubs	4.06	4.51
6	Chi Sox	4.23	4.43
7	Cincinnati	4	5.03
8	Cleveland	4.23	3.88
9	Colorado	4.31	5.39
10	Detroit	3.44	4.4
11	Houston	4.51	3.17
12	Kansas City	3.95	5
13	LA Angels	3.85	4.12
14	LA Dodgers	5.17	3.18
15	Miami	3.62	4.17
16	Milwaukee	4.48	4.25
17	Minnesota	4.3	4.22
18	NY Mets	4.73	3.77
19	NY Yankees	4.89	3.5
20	Oakland	3.51	4.75
21	Philadelphia	4.59	4.15
22	Pittsburgh	3.65	5.04
23	San Diego	4.34	4.05
24	Seattle	4.27	3.86
25	SF Giants	4.42	4.3
26	St. Louis	4.73	3.93
27	Tampa Bay	4.07	3.76
28	Texas	4.36	4.59
29	Toronto	4.78	4.23
30	Washington	3.72	5.28

2. Inputting above data to model through R

```
#Input live data to predict 2022 and test accuracy:0.9
runs2022=read.csv('/Users/chenyikai/Documents/runs2022.csv')
runs2022
predict=predict(logit2,type='response',newdata=runs2022)
```

3. Receiving following result

```
          1           2           3           4           5           6           7
0.03812716 0.94314742 0.08000852 0.02455741 0.01507496 0.04731508 0.00090375
          8           9          10          11          12          13          14
0.42467614 0.00064605 0.00149978 0.98871024 0.00083138 0.03787300 0.99942994
         15          16          17          18          19          20          21
0.01054335 0.27577902 0.16125092 0.92713977 0.99011515 0.00037233 0.50820693
         22          23          24          25          26          27          28
0.00017123 0.34747235 0.49478157 0.18426204 0.85301438 0.38861594 0.03963307
         29          30
0.62646518 0.00007286
```

4. Interpreting the result

In step 3, we get 5 rows, and, in each row, we have both team ID and predicted probability. For example, team ID 2 is the Atlanta braves and we predicted that they have 0.943 of playoff-making probability. In fact, in 2022 the braves did advance to the playoff, so we actually get an accurate prediction in the braves' case. Note that the index of team ID in step 3 is also as same as team ID in step 1.

Then, after summarized and compared all my prediction to the actual results, I get an 90% of accuracy. Inaccurate predictions are the Guardians', Padres' and Rays'. We predicted these team won't make the playoff because of their probabilities below 0.5. However, those teams actually advanced to the playoff in 2022. In the next page, we will see the prediction for the 2023 season.

Inputting Live Data for 2023 season

What's the probability of advancing to playoff this year?

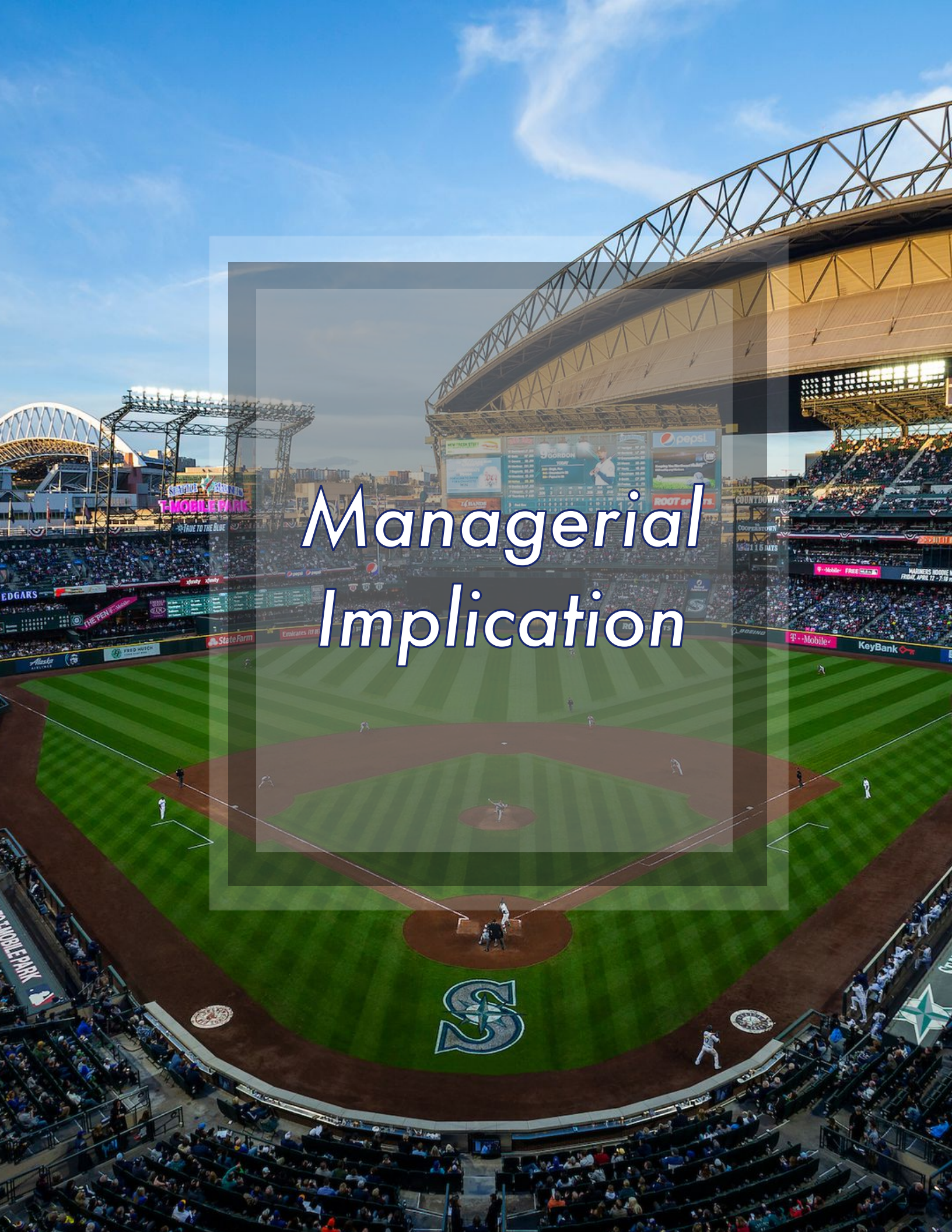
Below table contains each team's runs scored per game and runs allowed per game until 6/4. The contribution of my approach is that the data is on-going, meaning that people can update the probability as long as they collect the newest data.

teamid	Team	rs_g	ra_g
1	Arizona	4.95	4.58
2	Atlanta	4.98	3.95
3	Baltimore	4.9	4.47
4	Boston	5.22	4.95
5	Chi Cubs	4.39	4.39
6	Chi Sox	4.27	5.12
7	Cincinnati	4.66	5.19
8	Cleveland	3.62	4.17
9	Colorado	4.55	5.47
10	Detroit	3.66	4.73
11	Houston	4.59	3.57
12	Kansas City	3.91	5.24
13	LA Angels	4.83	4.8
14	LA Dodgers	5.59	4.61
15	Miami	3.83	4.49
16	Milwaukee	4.07	4.47
17	Minnesota	4.51	3.73
18	NY Mets	4.31	4.47
19	NY Yankees	4.72	4.07
20	Oakland	3.35	6.85
21	Philadelphia	4.16	4.91
22	Pittsburgh	4.42	4.26
23	San Diego	4.16	3.93
24	Seattle	4.4	4.19
25	SF Giants	4.53	4.43
26	St. Louis	4.76	4.69
27	Tampa Bay	5.82	3.85
28	Texas	6.39	3.88
29	Toronto	4.64	4.15
30	Washington	4.31	4.81

The probability is **26%.**

The team id of the Mariners is 24, so we can find out that the chance is 0.26. To increase that probability, the team might need to improve their scoring ability by trading for batters that has good oxwOBA, Slugging percentage and etc.

```
         1          2          3          4          5          6          7
0.39750973 0.94342772 0.47331984 0.27181239 0.11226531 0.00201787 0.00859952
         8          9         10         11         12         13         14
0.01054335 0.00131974 0.00082033 0.94679295 0.00021303 0.11413656 0.91609287
        15         16         17         18         19         20         21
0.00580743 0.01913150 0.84871863 0.05574800 0.73600640 0.00000001 0.00339907
        22         23         24         25         26         27         28
0.21560674 0.29484001 0.26109975 0.16547859 0.13794868 0.99923977 0.99993650
        29         30
0.56551378 0.01100945
```



Managerial Implication

Pitchers' Performance

After knowing that oOBP, oXWOBA and oSLG are all critical factors that correlate a lot with a team's runs allowed, I compare each Mariners pitcher's 2022 performance in such fields to MLB's teams' median. By doing so, we will have a clearer view of how to further improve player recruitment or to specify training menu.

In figure 1, the orange vertical line shows the median of oOBP in MLB, which is 0.318. The histogram in green shows the distribution of MLB teams' oOBP. If a player's oOBP is higher than the median, which is not a good sign, the player's respective data will be turned into a red dot.

Likewise, figure 2 and figure 3 have the same mechanism, allowing us to summarize each Mariners pitchers' pitching performance.

In summary of those three figures, I find out that the dots of Marco Gonzales have all appeared in red. In fact, this observation doesn't mean that the team should replace this player. However, instead, the pitching coaches can be more aware of this phenomenon, thereby finding solutions such as altering pitching strategies to enhance his performance.

Figure 1.

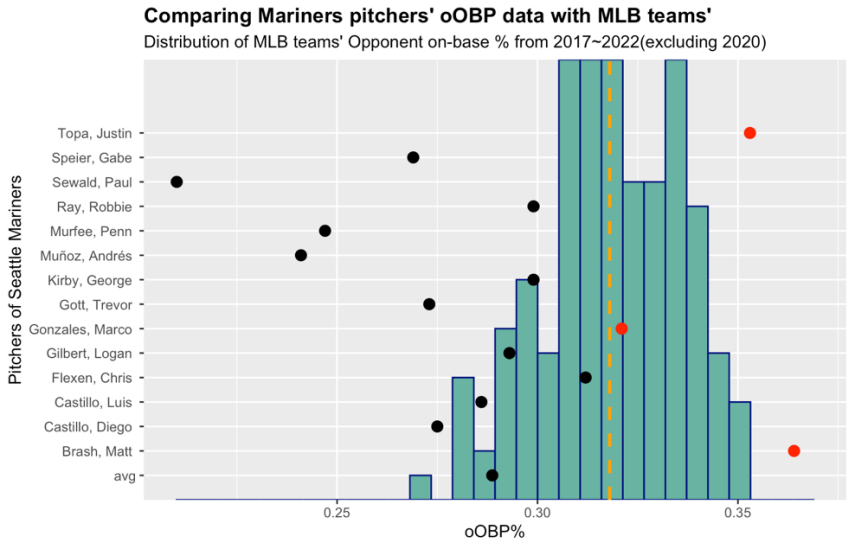


Figure 2.

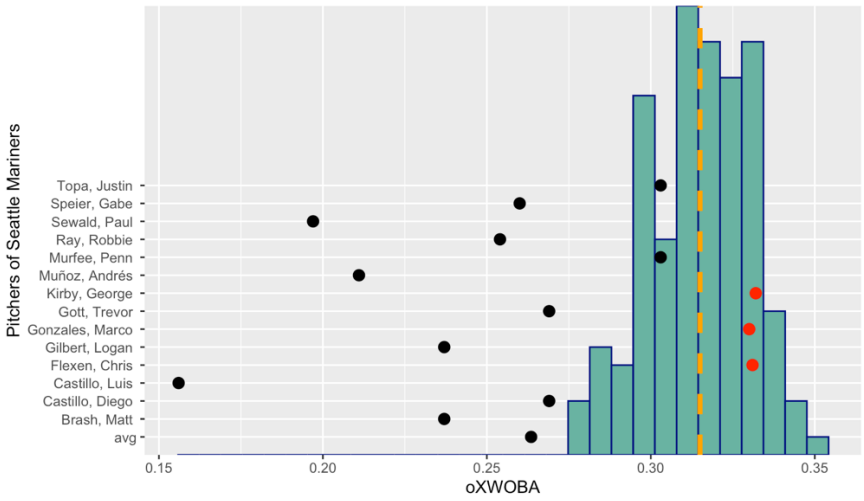
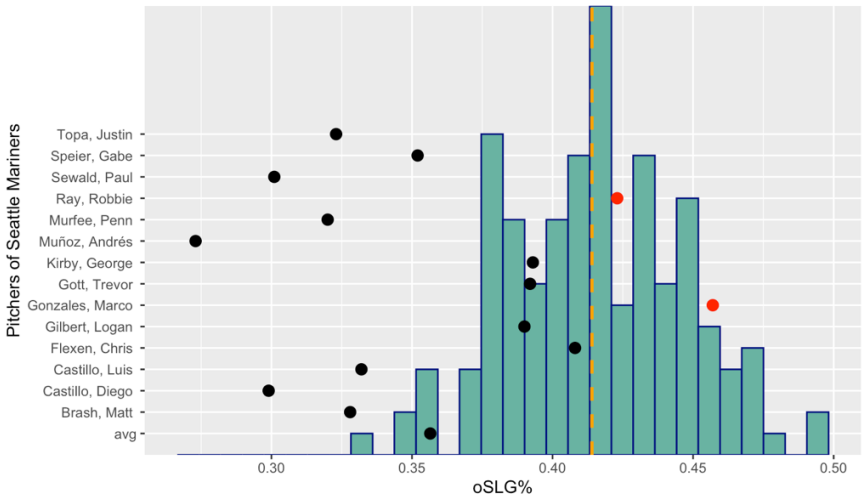


Figure 3.



Conclusion

Sabermetrics

We get 2 reasonable outcomes by using the multiple linear regression and also the logistic regression. In fact, our project showcases the power of the combination of statistic and baseball, which is also known as the Sabermetrics.

Strategic Decision

One of the utilities of Sabermetrics is that the management team can use the numbers to improve their decisions not only in player recruitment but also in every single scenario occurred in a game. To illustrate, based on our observation, the Mariners can increase their run scoring ability for the playoff odds by acquiring players with good SLG and oxwOBA.

Limitation

Both two predictions we have are bit contradict. We predict that the Mariners will get 96 wins eventually, but we only predict that the playoff odds are 26% based on current performance. This might be counterintuitively because teams with 93+ wins all advance to the playoff. However, I believe that the Mariners will bounce back strongly by the end of the season, thus increasing their playoff odds.

Source of Information

1. https://en.wikipedia.org/wiki/Logistic_regression
2. https://baseballsavant.mlb.com
3. https://ocw.mit.edu/courses/15-071-the-analytics-edge-spring-2017/
4. https://www.teamrankings.com/mlb/stat/opponent-runs-per-game
5. https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression/
6. https://rstudio-pubs-static.s3.amazonaws.com/524350_909ddc890c3042b8a3e7950019d2774d.html
7. https://rpubs.com/Roxi/369394