Executive summary

In this study, we conducted exploratory data analysis and built multiple regression models to investigate the relationship between various offensive statistics and the number of runs scored in baseball. We evaluated the models using metrics such as mean squared error and Bayesian information criterion and found that a Poisson regression model with total hits, plate appearances, singles, doubles, triples, and walks as predictor variables performed the best.

We analyzed the coefficients of the model to determine the relative importance of each offensive statistic in predicting the number of runs scored. Some variables, such as singles and walks, had unexpected signs, suggesting complex relationships between these variables and the response variable in a team sport like baseball.

Our findings indicate that certain offensive statistics are more important than others in predicting the number of runs scored, and that a Poisson regression model can be an effective tool for this type of analysis. However, further research is needed to fully understand the complex factors that can affect the relationships between these variables.

Problem

The data set we choose for this project is 2020-2022 offensive baseball statistics. Our source of the data is Baseball Savant, a major league baseball affiliated data repository for everything professional baseball. The goal of the report is to make a model using limited offensive stats to predict runs. Runs are a good measure of offensive performance as they are the points scored by players. In baseball the primary goal is to have more runs at the end of the game than the opposing team.

The family of regression analysis we are using is poisson. The baseball regular season is 162 games every year for each team (even if games are canceled they are made up eventually before the season starts). Poisson distribution is a statistical distribution that models the probability of rare events occurring in a fixed interval of time or space, given the average rate at which these events occur. This is consistent with how baseball is played and the data collected for it and is more optimal than a general linear model. This is due to baseball data being non-negative. Linear models have the potential to give negative numbers which are not in the feasible region for prediction.

Baseball rule set summary:

The goal of baseball is to outscore the opposing team by striking a ball with a bat and consecutively traveling around four bases that are arranged symmetrically in a diamond configuration.The teams have a total of nine players, assigned to either defensive or offensive roles. While one group assumes the defensive position on the field, the other team takes on the

role of the offensive party, positioned at bat.A baseball game typically comprises nine innings. Each inning affords an opportunity for both teams to undertake positions of offensive (batting) and defensive (fielding) play.

In the game of baseball, scoring transpires when a member of the offensive team makes contact with the ball and subsequently proceeds to touch each of the four designated bases in a sequential manner, resulting in a run, which is our target variable. The act of batting entails the batter's concerted effort to strike the ball. The batter is afforded the opportunity to make three unsuccessful attempts at hitting the ball before incurring an out.

Pitching involves the throwing of the baseball towards the home plate, with the objective of achieving a strikeout or inducing the batter to make contact resulting in a ground or fly ball which can be effectively fielded by teammates on the field. The defensive team tries to seize the ball and label out the offensive participants who are in the process of running the bases. In the event that a fielder successfully catches a fly ball struck by the batter prior to its surface contact, the batter is deemed as having been"out." The term "outs" refers to a situation in which the defense successfully executes one of the three actions: obtaining three strikes on a given batter, catching a fly ball hit by said batter, or tagging a runner with the ball before they reach a base. Once the defense has 3 outs the teams switch positions, and / or innings change.

Variable Descriptions:
Our data source baseball savant has 100’s of statistics for each player most of which or ratios built by countable stats. This being said we wanted a more manageable data set so we used the “basic stats” filter to only give that main statistics that would be a newspaper for example. This being said we still have multiple independent variables that are ratios based off of other independent variables. This can cause problems regarding correlation and will be explored in our modeling building approach. This being said we still have multiple independent variables that are ratios based off of other independent variables. This can cause problems regarding correlation and will be explored in our modeling building approach. .

- r_run: : The number of runs the player scored during the season (**Dependant Variable**)
- last_name: The last name of the player
- first_name: The first name of the player
- player_id: A unique identifier for the player
- year: The year of the season being analyzed
- player_age: The age of the player during the season
- b_ab: The number of at-bats the player had during the season
- b_total_pa: The total number of plate appearances the player had during the season
- b_total_hits: The total number of hits the player had during the season
- b_single: The number of singles the player had during the season

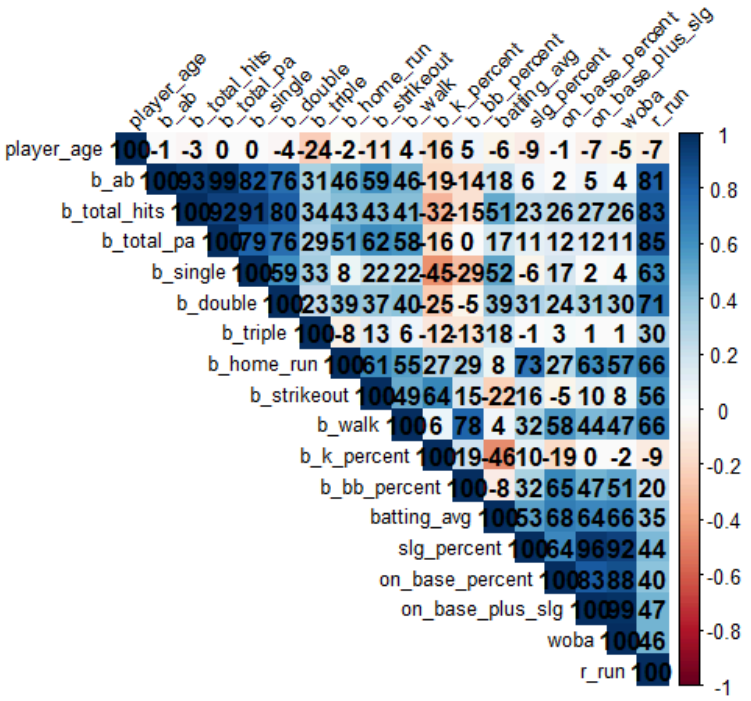
- b_double: The number of doubles the player had during the season
- b_triple: The number of triples the player had during the season
- b_home_run: The number of home runs the player had during the season
- b_strikeout: The number of times the player struck out during the season
- b_walk: The number of walks the player had during the season
- b_k_percent: The percentage of plate appearances that resulted in a strikeout for the player (B_STRIKEOUT/B_TOTAL_PA)
- b_bb_percent: The percentage of plate appearances that resulted in a walk for the player (B_WALK/B_TOTAL_PA)
- batting_avg: The player's batting average, which is the number of hits divided by the number of at-bats (B_TOTAL_HITS/B_AB)
- slg_percent: The player's slugging percentage, which is the total number of bases the player has divided by the number of at-bats ((B_SINGLE + 2 * B_DOUBLE + 3 * B_TRIPLE + 4 * B_HOME_RUN)/B_AB)
- on_base_percent: The player's on-base percentage, which is the percentage of plate appearances that result in the player reaching base ((B_TOTAL_HITS + B_WALK)/(B_AB + B_WALK))

- on_base_plus_slg: The player's on-base plus slugging percentage, which is the sum of the player's on-base percentage and slugging percentage(ON_BASE_PERCENT + SLG_PERCENT)
- r_run: The number of runs the player scored during the season
- woba: The player's weighted on-base average, which is a more advanced metric that measures a player's overall offensive performance.(wOBA = (0.69 × uBB + 0.72 × HBP + 0.89 × B_single + 1.27 × B_double + 1.62 × B_triple + 2.10 × b_home_run) / (b_ab + b_bb – IBB + SF + HBP) (IBB: intentional walks, SF: sacrifice flies, uBB: unintentional walks, HBP: times hit by pitch)

Data Cleaning / Visualization

Our data set coming from baseball savant has been pre cleaned and formatted. Their site lets you choose the statistics you want and directly makes a csv file for use.

Whenever making a complex model, it's a good rule of thumb to create a correlation matrix for the independent variables. This can show us the relationships between the independent variables and the dependent variable for a model building, and if the multicollinearity between the independent variables is significant.



As the correlation table shows, OB+SLG is highly correlated with SLG and woba which are also correlated with each other. This can be explained by stats themselves being ratios based on the same count data in different forms (variable list formulas).

AB and total PA are highly correlated because they are almost the same statistic. AB represents the number of times a batter has faced a pitcher and attempted to hit the ball, while total PA includes ABs as well as other plate appearances such as walks, hit by pitch, sacrifices, and catcher's interferences. Due to their similarity, including both AB and total PA in the same model would result in high multicollinearity.

Analysis / Model Selection

The first model we created is the base model. It uses all of the independent variables except for first and last name. The base model contains categorical data like First Name, Last Name, Player ID. These are necessary for data organization, however are not useful for the type of model we are creating. As part of the data cleaning we removed these variables so the AIC algorithm does not confuse it for continuous data.

Model 1 and 2 build methodology.

When building these models, we used the step AIC and step BIC algorithmic model building technique. The base model that we created the models from includes all independent variables except the previously mentioned categorical data.

Model 3 and 4 build methodology

When building these models, we used the base model 2. Base model 2 takes into consideration the multicollinearity of the independent variables and has been truncated to 9 independent variables. The first variable that was excluded was b_ab which is 99% correlated with b_total_pa which is included. We also excluded any variables that are ratios based off of other count data which can lead to multicollinearity as well. Model 3 uses the step AIC algorithm and model 4 uses BIC algorithm for model building.

Variables all models excluded

In the context of our regression analysis, it is pertinent to note the variables that were excluded from both models, as their absence suggests that they may not have a significant impact on the response variable. These variables, specifically player age, b_k_percent, batting average, and b_home_run, were found to have relatively low p-value significance in both base models.

The exclusion of player age, in particular, is noteworthy as it may imply that player experience and skill level may not have a strong correlation with the number of runs scored by a team. Similarly, the exclusion of variables such as b_k_percent and batting average may suggest that these metrics are not as influential in determining a team's offensive output as other variables, such as the number of hits or walks.

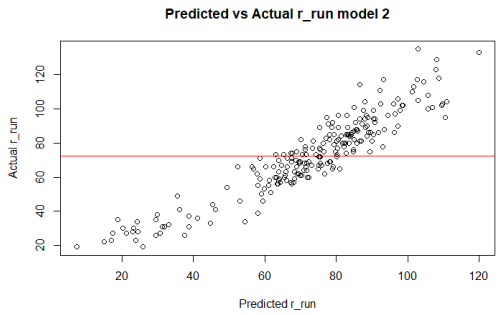
It should be noted, however, that the exclusion of these variables does not necessarily imply that they are not important in a broader context or that they should be disregarded entirely. Rather, it is indicative of their relative importance in the specific regression models that were constructed for this analysis.

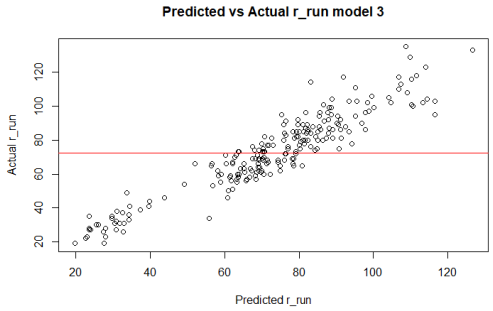
Model Cross Validation

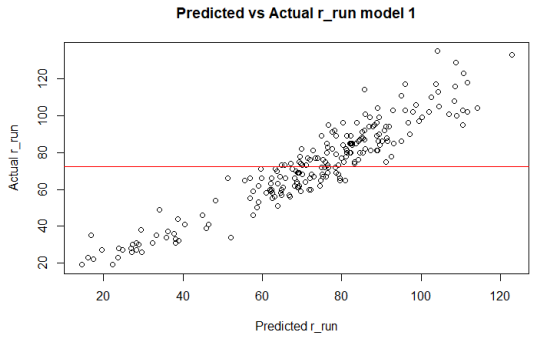
In order to test the predictions of our model, we split our data into training and test data. We used an 80/ 20 split, 80% training 20% test. After making the predictions we calculated the SQRT MSE of each of the predictions (results of the test are in the Model comparison matrix below. Another method of cross validation is BIC. BIC has an advantage penalizing the complexity of the model which can lead to overfitting. We measured the log likelihood of each of the models and compared the individual BIC (results in model comparison matrix).

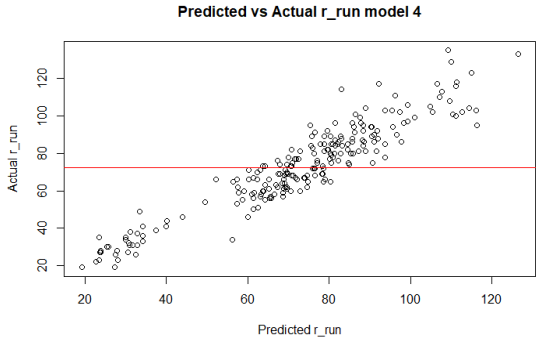
Predicted vs Actual Plots

The red line is the mean of runs scored as a null hypothesis. These graphs show that all of the models are better than the coefficients being 0. The redline









Model Comparison Matrix

Models	Independent variables (k)	Sqrt MSE	BIC	List of Variables
Model 1	12	8.128592	7858	b_total_hits / b_total_pa / b_single / b_double / b_triple / b_strikeout / b_bb_percent / batting_avg / slg_percent / on_base_percent / on_base_plus_slg / woba
Model 2	5	8.586367	7827	b_total_pa / b_triple / b_strikeout / slg_percent / woba
Model 3	7	8.1773	7951	b_total_pa / b_total_hits / b_single / b_double / b_triple / b_strikeout / b_walk
Model 4	6	8.158577	7951	b_total_pa / b_total_hits / b_single / b_double / b_triple / b_walk

This is a breakdown of our cross validation methods and the selection process. We can see here that the model 1 has the lowest SQRT MSE. This means that the model has the least amount of raw errors when compared to our test data. However, this can be the result of over fitting since

this model has the most variables. Also, the variables used in model 1 have high correlation and multicollinearity being ratios based on other independent variables.

Model 2 does use the ratio based variables like Woba. b_triple is used in the formula for woba, however they are almost not correlated at all being 1%. The model 2 used BIC selection process so it has an internal measure to avoid multicollinearity by giving a higher penalty factor to k than AIC. This is also why the number of k is lower in model 2 as compared to model 1.

Model 3 uses the base model 2 which has ratio variables removed before the model building process. This actually lowers the total MSE and brings clarity to the model. Model 3 BIC is also higher compared to model 2 because of the extra independent variables.

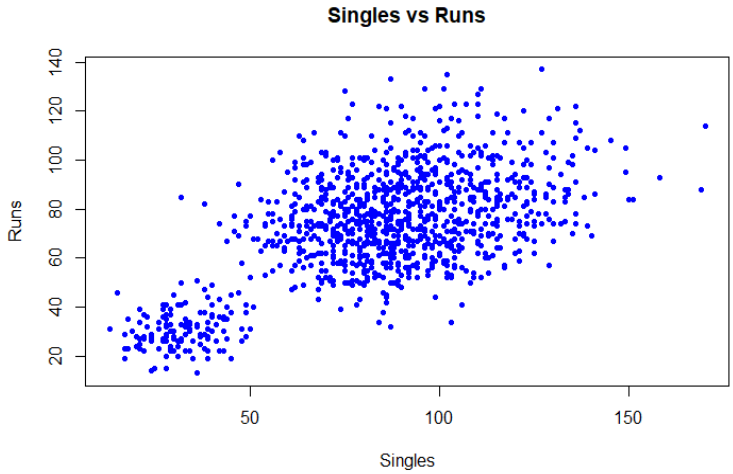
Model 4 also uses the base model 2 for its initial variables. However we used the BIC model build approach for model 4 instead of AIC. This gives a higher penalty factor for the more independent variables in the model. As you can see the models are very similar except for the b_strikeout variable. With removing b_strikeout the model has better clarity and less errors when applying it to our test data set. B_strike out can be determined to not be an important coefficient in the context of the other r model 4 coefficients.

In conclusion, for our data set and our assumptions we select model 4. Our selection criteria in mind being BIC and MSE, model 4 has more clarity than model 3 while also not risking as much multicollinearity being made of count data not ratio based data.

Model 4 coefficient analysis

Intercept	b_total_hits	b_total_pa	**b_single**	b_double	b_triple	b_walk
2.935e+00	1.066e-02	8.437e-04	**-7.238e-03**	-5.276e-03	7.578e-03	2.695e-03

It is interesting looking at our model and how it comes to its prediction. In real baseball, a single directly leads to scoring more runs for yourself s]and your team as a whole.Theoretically this should always be positive because when translated to real the sum bases run would increase your runs scored statistics. However in our model these coefficient estimates are always in the context of the other coefficients and our target variable. It would make sense that in our model scoring singles would have a negative effect on our target variables value. This might be due to the fact that the players with higher ratios of singles compared to other hits would result in less runs scored even though literally this is not true due to baseball being a team sport. Every run single score has a positive effect on runs as shown in the graphic below.



When creating analyzing coefficients we have to keep in mind that it is in context of all the other independent variables.

Summary

This study examined the relationship between offensive statistics and the number of runs scored by a baseball team. Regression models were constructed using various combinations of offensive statistics as predictor variables and the number of runs scored as the response variable. The effectiveness of the models was assessed using metrics such as mean squared error and Bayesian information criterion. A Poisson regression model with total hits, plate appearances, singles, doubles, triples, and walks as predictor variables was found to perform the best.

Coefficients of this model were examined to determine the relative importance of each offensive statistic in predicting the number of runs scored. It was observed that the coefficients of certain variables, such as singles and walks, had unexpected signs, likely due to the complex relationships between these variables and the response variable in a team sport like baseball.

Visualization techniques were employed to explore the relationships between specific variables and the number of runs scored. A scatterplot of singles vs. runs scored revealed a positive relationship between the two variables, which is in line with expectations.

The study concludes that certain offensive statistics are more influential than others in predicting the number of runs scored by a baseball team, and that a Poisson regression model is an effective tool for such analyses.

However, we also found that there are many complex factors that can affect the relationships between these variables, and further research may be needed to fully understand these relationships.

Appendix:

```
library(MASS)   # For fitting the Poisson regression model
library(car)    # For the stepAIC function
library(caret)
library(corrplot)
library(reshape2)
library(DataExplorer)
#inserting data and naming it data
data = `baseballdatafinal`
#practice 3 variable manual model poison regression
#model <- glm(r_run ~ player_age + b_ab + b_total_hits, family = poisson(link = "log"), data =
data)
#making a corr matrix
# Select independent variables
vars <- data[, c("player_age" , "b_ab" , "b_total_hits" , "b_total_pa", "b_single", "b_double",
"b_triple", "b_home_run", "b_strikeout" , "b_walk" , "b_k_percent" , "b_bb_percent" ,
"batting_avg" , "slg_percent" , "on_base_percent" , "on_base_plus_slg" , "woba" , "r_run" )]

# Calculate correlation matrix
cor_matrix <- cor(vars)

# Plot correlation matrix
corrplot(cor_matrix, type = "upper", method = "color",
         tl.col = "black", tl.srt = 45, tl.cex = 0.8,
         addCoef.col = "black", addCoefasPercent = TRUE)

create_report(baseballdatafinal)
#Below models are made with all variables from data sets
#Every Variable Model
base_model <- glm(r_run ~ player_age + b_ab + b_total_hits + b_total_pa + b_single + b_double
+ b_triple + b_home_run + b_strikeout + b_walk + b_k_percent + b_bb_percent + batting_avg +
slg_percent + on_base_percent + on_base_plus_slg + woba, family = poisson(link = "log"), data
= data)
summary(base_model)

# AIC Building Form
step_model1 <- stepAIC(base_model, direction = "both")
summary(step_model1)
predict(step_model1,type="response")
```

```
# BIC model building form.
step_model2 <- stepAIC(base_model, direction = "both",k=log(1112))
summary(step_model2)
predict(step_model2,type="response")

# split data and create testing and training set

#below models are made after a corr table was made and ratio based stats, and the b_ab is
removed
base_model2 <- glm(r_run ~ player_age + b_total_hits + b_total_pa + b_single + b_double +
b_triple + b_home_run + b_strikeout + b_walk, family = poisson(link = "log"), data = data)
summary(base_model2)
#step aic model made with base model corr table truncated variables
step_model3 <- stepAIC(base_model2, direction = "both")
summary(step_model3)
predict(step_model3,type="response")
#step bic model made with base model with corr table truncated Variables
step_model4 <- stepAIC(base_model2, direction = "both",k=log(1112))
summary(step_model4)
predict(step_model4,type="response")
#anova of models
anova(step_model1,step_model2)


set.seed(222) # set seed

# creates a 80% selection for training from the dataset

train_index <-createDataPartition(data$r_run, p=0.80, list=F)



data_train<-data[train_index,] # select training set

data_test<-data[-train_index,] # select testing set
#predicting training data
attach(data_train)
#model 1 training
train_step_model1 <- glm(step_model1, data = data_train)
summary(train_step_model1)
```

```
#model 2 Training
train_step_model2 <- glm(step_model2, data = data_train)
summary(train_step_model2)
#model 3 training
train_step_model3 <- glm(step_model3, data = data_train)
summary(train_step_model3)
#model 4 training
train_step_model4 <- glm(step_model4, data = data_train)
summary(train_step_model4)
detach(data_train)
attach(data_test)
#predictions for model 1
predictions_M1=predict(train_step_model1,newdata = data_test)
summary(predictions_M1)
#prediction for model 2
predictions_M2=predict(train_step_model2,newdata= data_test)
summary(predictions_M2)
#predictions for model 3
predictions_M3=predict(train_step_model3,newdata= data_test)
summary(predictions_M3)
#prediction for model 4
predictions_M4=predict(train_step_model4,newdata= data_test)
summary(predictions_M4)


detach(data_test)
# comparing sqrt (MSE) of all models
MSE_step_model1=sum((data_test$r_run-predictions_M1)^2)/length(data_test$woba)
sqrtMSE1 <- sqrt(MSE_step_model1)

MSE_step_model2=sum((data_test$r_run-predictions_M2)^2)/length(data_test$woba)
sqrtMSE2 <- sqrt(MSE_step_model2)

MSE_step_model3=sum((data_test$r_run-predictions_M3)^2)/length(data_test$woba)
sqrtMSE3 <- sqrt(MSE_step_model3)

MSE_step_model4=sum((data_test$r_run-predictions_M4)^2)/length(data_test$woba)
sqrtMSE4 <- sqrt(MSE_step_model4)
#BIC test for model comparison to make penalize model complexity which isn't taken into
account in MSE testing
```

```
#creating the logLik for each model
log_likelihood1 <- logLik(step_model1)
log_likelihood2 <- logLik(step_model2)
log_likelihood3 <- logLik(step_model3)
log_likelihood4 <- logLik(step_model4)

# Getting the parameters
numpara1 <- length(coefficients(step_model1))
numpara2 <- length(coefficients(step_model2))
numpara3 <- length(coefficients(step_model3))
numpara4 <- length(coefficients(step_model4))
# calculate BIC get n value
n <- nrow(data)
BIC1 <- -2*log_likelihood1 + numpara1 *log(n)
BIC2 <- -2*log_likelihood2 + numpara2 *log(n)
BIC3 <- -2*log_likelihood3 + numpara3 *log(n)
BIC4 <- -2*log_likelihood4 + numpara4 *log(n)
#output of BIC
BIC1
BIC2
BIC3
BIC4

# create a base test mean of r runs
mean_r_run <- mean(data$r_run)
me# create actual r_run values
actual_r_run <- data_test$r_run
# create a scatter plot of predicted vs actual values

plot(predictions_M1, actual_r_run, xlab = "Predicted r_run", ylab = "Actual r_run", main =
"Predicted vs Actual r_run model 1")
plot(predictions_M2, actual_r_run, xlab = "Predicted r_run", ylab = "Actual r_run", main =
"Predicted vs Actual r_run model 2")
plot(predictions_M3, actual_r_run, xlab = "Predicted r_run", ylab = "Actual r_run", main =
"Predicted vs Actual r_run model 3")
plot(predictions_M4, actual_r_run, xlab = "Predicted r_run", ylab = "Actual r_run", main =
"Predicted vs Actual r_run model 4")
abline(h = mean_r_run, col = "red")

#creating run to single plot to show that a=they are not actually negatively correlated
```

```
plot(x = data$b_single, y = data$r_run, main = "Singles vs Runs",
     xlab = "Singles", ylab = "Runs", col = "blue", pch = 20)
```