Baseball Salary Analysis

Edgar Hernandez, Ryan Sims, Jessica Tomas

2022-08-04

Contents

Introduction 1

Introduction

We will be examining what statistics related to a baseball player's performance have a significant effect on the salary of a player. This is particularly interesting as it is sort of the flip-side of the coin to the traditional Sabermetrics employed by the Oakland Athletics in the 1990's to analyze player statistics to try to assemble a winning team. We will be instead be leveraging the dataset to instead work for the players, hopefully determining which statistic(s) a player should focus on improving in order to increase their compensation.

Before we begin our analysis, we must first load the data using the `Lahman` library:

```
library(Lahman)
```

We then join the Salaries, Fielding, and Batting tables:

Methods

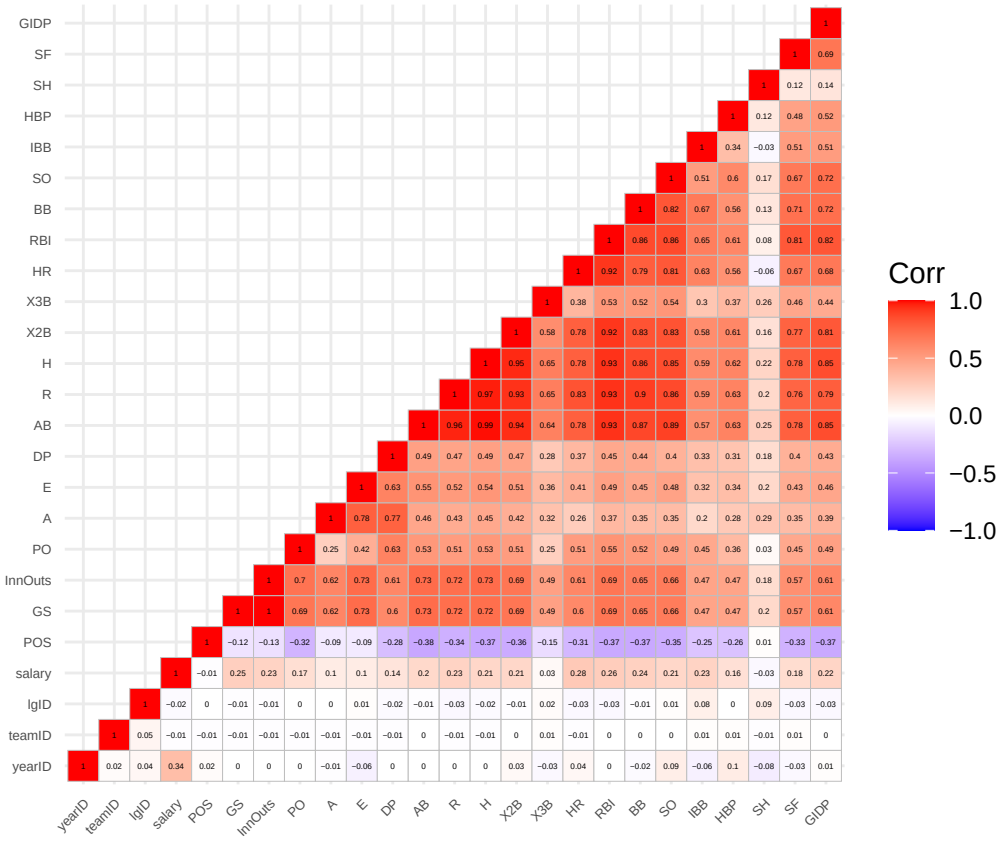
First, we visualize pairwise correlations between `Salary` and other variables:

- Calculating the correlations:

```
salary_data_plot = as.data.frame(lapply(salary_data, as.integer))
salary_data_cor = cor(salary_data_plot)
```

- Plotting the relationships:

```
library(ggcorrplot)
my_plt = ggcorrplot(salary_data_cor, lab = TRUE, lab_size = 1, show.diag = TRUE, type = "lower")
my_plt + theme(axis.text.x = element_text(size = 5), axis.text.y = element_text(size = 5))
```



- As expected, certain batting predictors such as `Doubles (X2B)` and `Triples (X3B)` have a very high correlation. We will keep these correlations under consideration as we refine our model.

Before generating models, we begin by splitting the data into a test and train dataset. For this project we will split 60% of the data into a training set and 40% of the data into a testing set.

```
salary_trn_idx = sample(nrow(salary_data), size = trunc(0.60 * nrow(salary_data)))
salary_trn = salary_data[salary_trn_idx, ]
salary_tst = salary_data[-salary_trn_idx, ]
```

Next, we create a simple additive model with all of the predictors:

```
simple_add = lm(salary ~ ., data = salary_data)
```

We then use a backwards AIC stepwise procedure to remove predictors:

```
simple_back_aic = step(simple_add, direction = "backward", trace = 0)
```