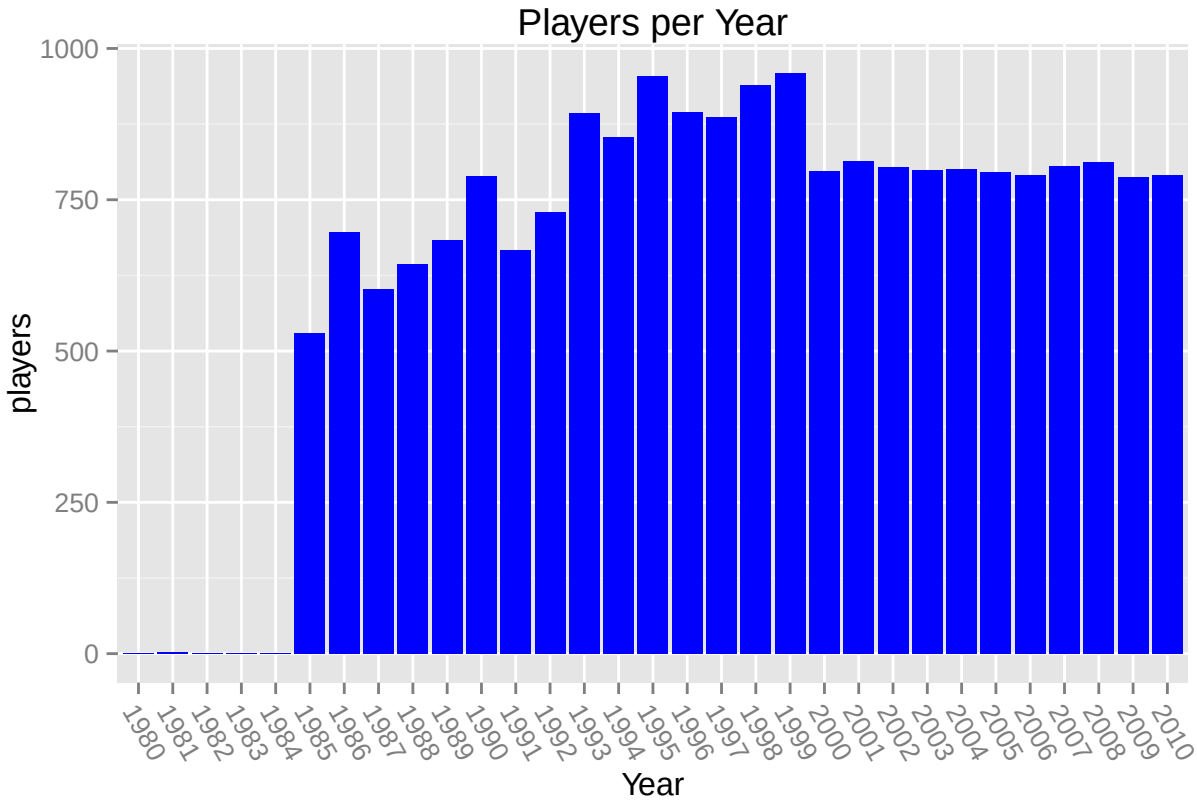
Baseball Data Analysis

Exploratory analysis of the data

The data set contains baseball player data from 1980 to 2010. Records are maintained on each player, for each year, team, league, stint, and fielding position they played. Overall, 4055 unique players are recorded in the data set, covering 20524 man years.

```
ggplot(by_year, aes(x = as.factor(yearID), y = players)) +
  geom_bar(stat = "identity", fill = "Blue") +
  xlab("Year") +
  ggtitle("Players per Year") +
  theme(axis.text.x = element_text(angle = -60, hjust = -0))
```



Batting Data Exploration

Batting data was recorded including:

- b_G -
- b_G_batting
- b_AB - At Bats
- b_R - Runs
- b_H - Hits
- b_2B - Doubls

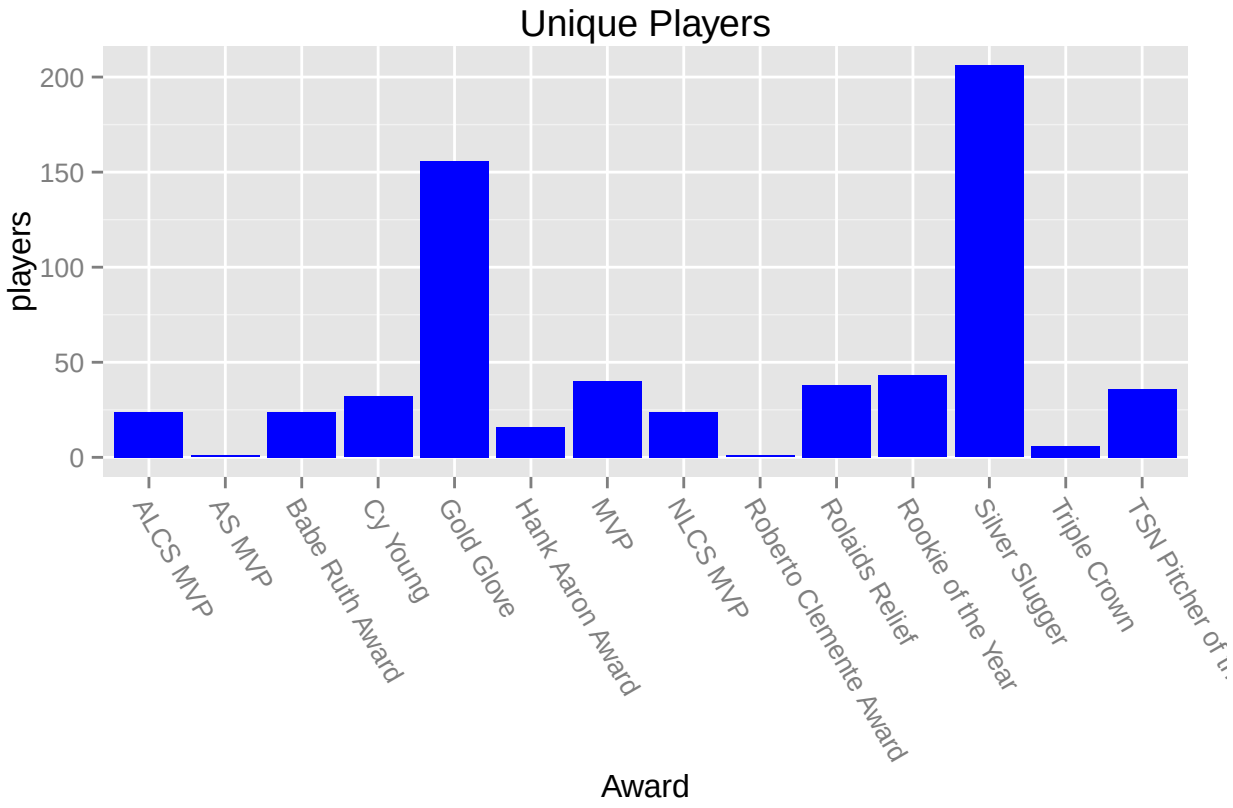
- b_3B - Tripples
- b_HR - Home Runs
- b_RBI - Runs Batted In
- b_SB - Stolen Bases
- b_CS - Caught Stealing
- b_BB - Bases on Balls (Walks)
- b_SO - Strikeouts
- b_IBB - Intentional Bases on Balls (Walks)
- b_HBP - Hit By Pitch
- b_SH - Sacrifice Hits (Bunts)
- b_SF - Sacrifice Flies
- b_GIDP
- b_G_old

In addition to the baseball statistics provided, several additional ratios were computed to normalise the data:

- b_hits_per_AB - Hits per At Bat
- b_runs_per_AB - Runs per At Bat
- b_runs_per_H - Runs per Hit
- b_home_runs_per_AB - Home Runs per At Bat
- b_balls_per_AB - Balls per At Bat
- b_RBI_per_H - Runs Batted In per Hit
- b_HBP_per_AB - Hit By Pitch per At Bat

Finaly, award data was considered:

```
ggplot(filter(award_winners, !is.na(awardID)), aes(x = as.factor(awardID), y = players)) +
  geom_bar(stat = "identity", fill = "Blue") +
  xlab("Award") +
  ggtitle("Unique Players") +
  theme(axis.text.x = element_text(angle = -60, hjust = -0))
```



As we can see, the silver slugger award has the most winners contained in the data set. As a result of this, it was chosen as the dependent variable.

The dataset also contained team, league, & fielding data that was not considered.

Model selection

To model the likelyhood that a player would recived the Silver Slugger Award a Logistic Regression was performed. Since the Silver Slugger is a batting award recognising players of high offensive value, the batting data set was used. According to wikipedia, the baseball coaches vote for players on other teams to win based on several batting ratios that we were able to derive from our raw count data.

The full model contained:

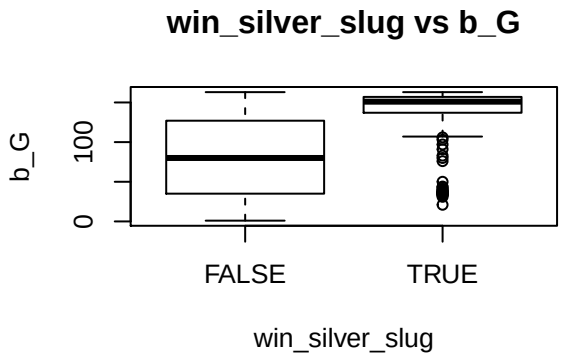
win_silver_slug = salary + b_G + b_G_batting + b_AB + b_R + b_H + b_2B + b_3B + b_HR + b_RBI + b_SB + b_CS + b_BB + b_SO + b_IBB + b_HBP + b_SH + b_SF + b_GIDP + b_G_old + b_hits_per_AB + b_runs_per_AB + b_runs_per_H + b_home_runs_per_H + b_balls_per_AB + b_RBI_per_H + b_HBP_per_AB

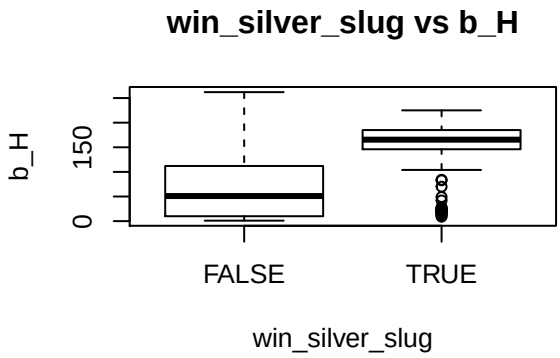
Model Selection

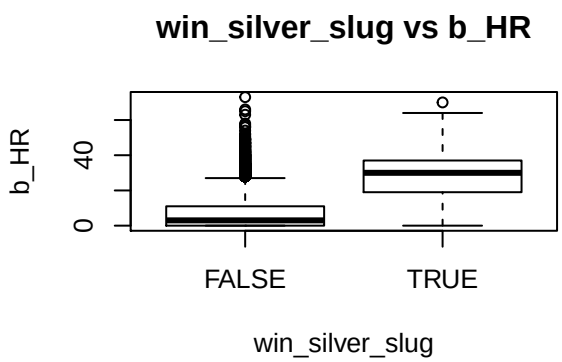
After backward selection the model that was chosen was: **log(odds(win_silver_slug)) = -5.10697989512792 + -0.0437813071772097 * b_G + 0.0298363646334512 * b_H + 0.0353237814406278 * b_HR + 0.0235952761309906 * b_RBI + 0.03690307794777 * b_CS + 0.0283786990755013 * b_BB + -0.00719867486183699 * b_SO + 0.115025302963441 * b_HBP + 1.90522965303078 * b_hits_per_AB + 2.46316604034702 * b_home_runs_per_H + -10.2429175488287 * b_balls_per_AB + -46.0399398528793 * b_HBP_per_AB**

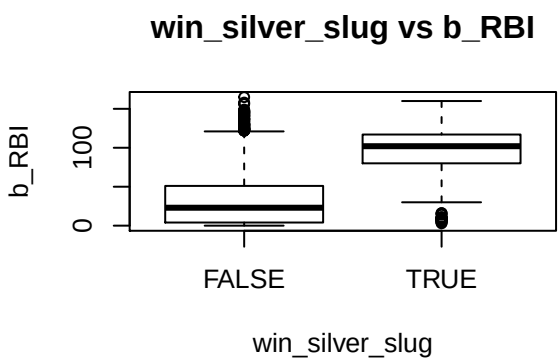
Check Fit

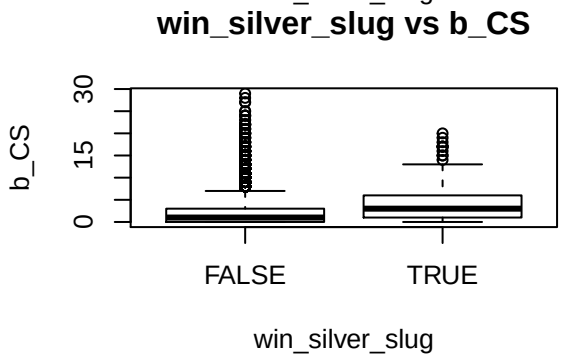
```
batting_data[which(colnames(batting_data) %in% c("win_silver_slug", names(model_backwards_selection$coe
plot_all_box(which(colnames(.) == "win_silver_slug"), .)
```

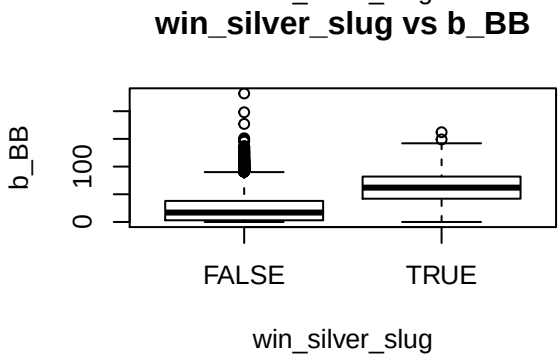

win_silver_slug vs b_G
b_G
0
100
FALSE
TRUE
win_silver_slug

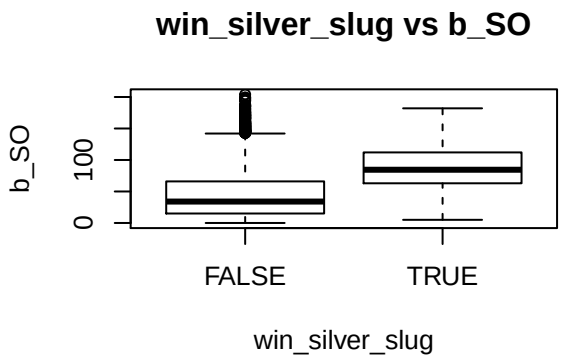

win_silver_slug vs b_H
b_H
0
150
FALSE
TRUE
win_silver_slug

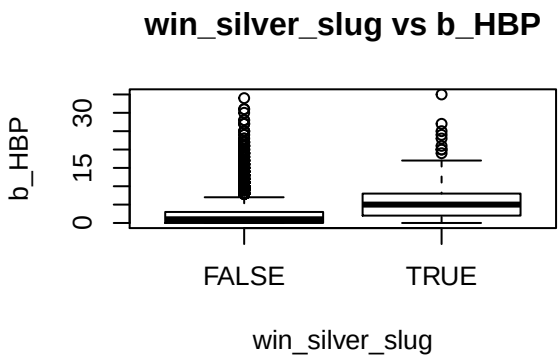

win_silver_slug vs b_HR
b_HR
0
40
FALSE
TRUE
win_silver_slug

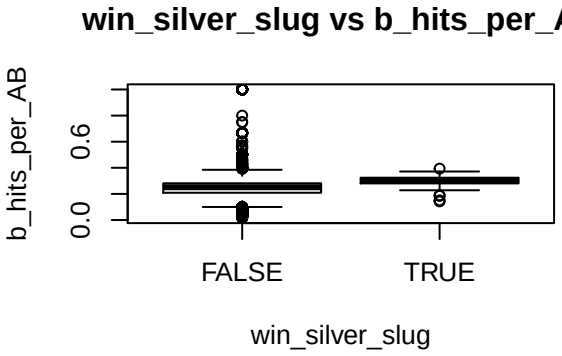

win_silver_slug vs b_RBI
b_RBI
0
100
FALSE
TRUE
win_silver_slug

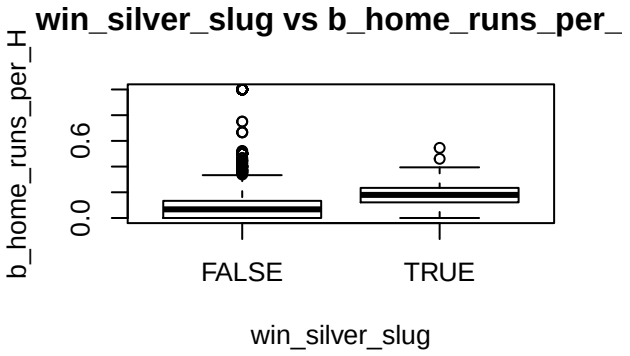

win_silver_slug vs b_CS
b_CS
0
15
30
FALSE
TRUE
win_silver_slug

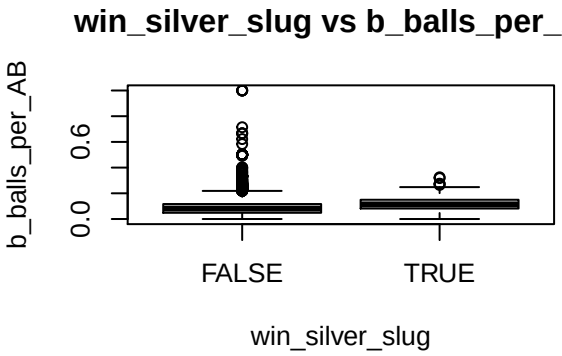

win_silver_slug vs b_BB
b_BB
0
100
FALSE
TRUE
win_silver_slug

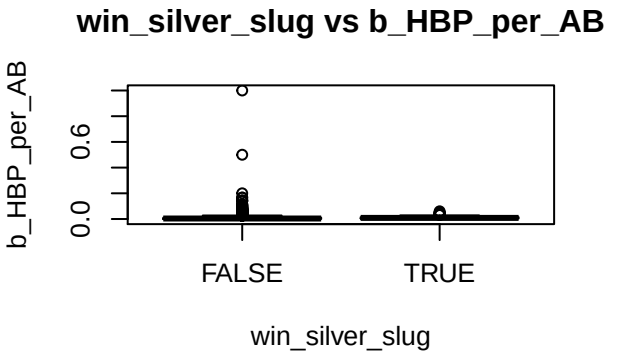

win_silver_slug vs b_SO
b_SO
0
100
FALSE
TRUE
win_silver_slug


win_silver_slug vs b_HBP
b_HBP
0
15
30
FALSE
TRUE
win_silver_slug










```
vif(model_backwards_selection)
```

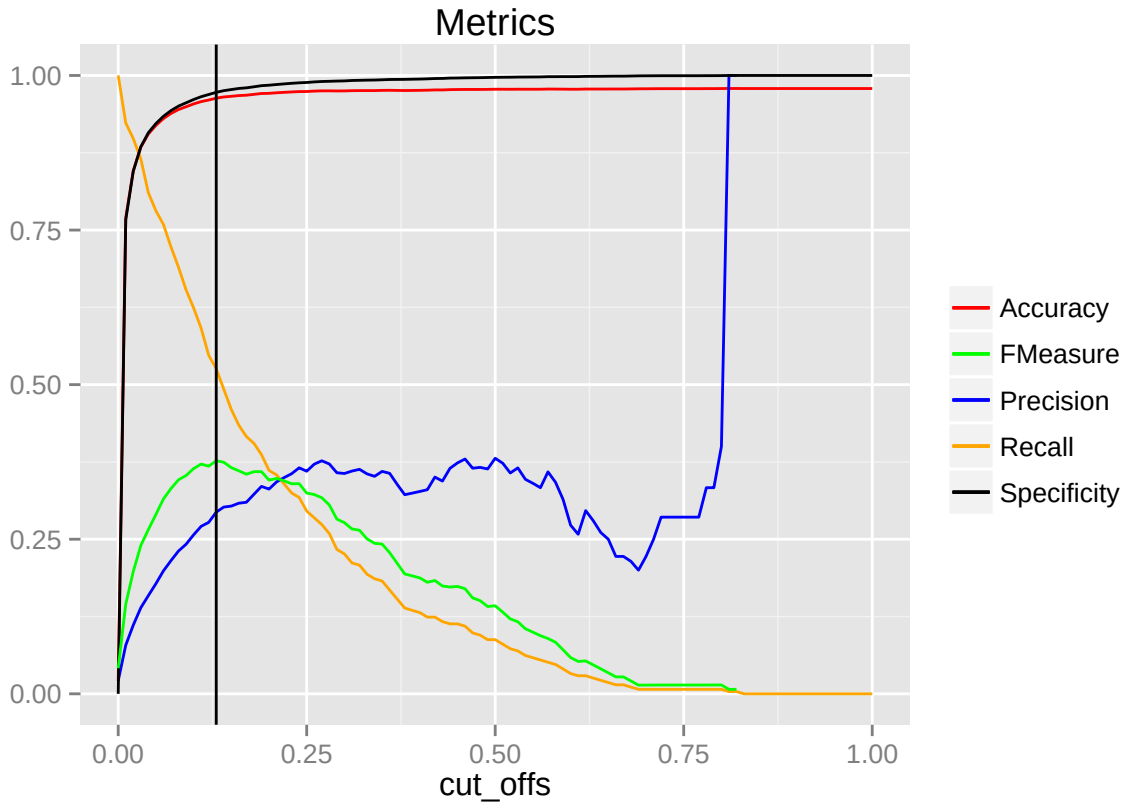
```
##              b_G              b_H             b_HR            b_RBI
##        11.457211        11.286429         7.332786        10.183930
##             b_CS             b_BB             b_SO            b_HBP
##         1.370876        17.147657         2.581509        19.221669
##     b_hits_per_AB b_home_runs_per_H    b_balls_per_AB      b_HBP_per_AB
##         1.164119         1.841251        12.686127        17.834542
```

```
lm.beta(model_backwards_selection)
```

```
##
## Call:
## glm(formula = win_silver_slug ~ b_G + b_H + b_HR + b_RBI + b_CS +
##     b_BB + b_SO + b_HBP + b_hits_per_AB + b_home_runs_per_H +
##     b_balls_per_AB + b_HBP_per_AB, family = binomial(), data = batting_data)
##
## Standardized Coefficients::
##       (Intercept)              b_G              b_H             b_HR
##         0.0000000      -14.7591199       12.2880563        2.3867899
##             b_RBI             b_CS             b_BB             b_SO
##         5.3694099        0.8367252        5.0736515       -1.8274445
##             b_HBP     b_hits_per_AB b_home_runs_per_H    b_balls_per_AB
##         2.5745607        1.2437720        1.6719370       -4.4004964
##      b_HBP_per_AB
##        -4.6642504
```

Finding a Optimal Cut Off

```
plot_of_cut_offs
```



After searching for a good cut off value, 0.13 was chosen.

```
best_cut_off
```

```
##   cut_offs true_positive true_negative false_positive false_negative
## 1     0.13           144         12345            346            130
##      recall  accuracy precision specificity f_measure      prior     lift
## 1 0.5255474 0.9632858 0.2938776   0.9727366 0.3769634 0.02113382 13.90556
```