Predicting MLB Team Win Percent and Playoff Status (1997-2016)

Group 3: Eashan Kaw, Michael Macfarlan, Wyatt Scott, Victor Teelucksingh

1. Executive summary

This project explored the ability to predict a Major League Baseball (MLB) team's win percentage and playoff status using baseball stats from 1997 through 2016 from the Lahman Baseball Database. This group intended to focus, in particular, on variables that a team owner or manager could have some level of influence over, either by offensive strategy decisions, defensive strategy decisions, player stats to focus on when making signing decisions, or other intentional strategies. The intention was to create predictive models for use in real-world decision-making scenarios for team managers and owners to improve outcomes. Right from the beginning, the objectives of this group may seem similar to those outlined in Michael Lewis' renowned book, Moneyball. However, the group delved further into the subject matter, expanding the focus to identify other crucial variables that could inform team owners' and managers' decisions and improve outcomes.

The Lahman Baseball database contains several team and player statistics across MLB seasons from 1871 to 2021. In our work, we pre-processed, then combined several tables from the database, including Teams, Salaries, and SeriesPost tables, to create a final dataframe with several useful variables. While exploring the datasets, this working group inspected, calculated, and combined these variables to create useful predictors in a single dataframe.

The group's first question of interest explored which team stats are the most useful for accurately predicting a team's winning percentage for the seasons spanning 1997 to 2016. Starting with exploratory data analysis using data visualizations, this working group began to form hypotheses, then put these to the test by constructing and testing a high-performing linear regression model.

In summary, the variables that were both influential and statistically significant enough to make it into our predictive model were (in order of influence) fielding percentage, on-base percentage, steal percentage, saves, home runs allowed, home runs, walks allowed, outs pitched, strikeouts by pitchers, and strikeouts by batters. On-base percentage and field percentage stand out as the most influential predictors. For every percentage point increase in OBP, the team's win percentage increases by an estimated 1.95%, and for every percentage point increase in field percentage, the team's win percentage increases by an estimated 2.23%, assuming all other variables remain constant.

The group's second question of interest explored the ability to successfully predict a team's ability to make it to the playoffs based on a handful of predictor variables. Again, by starting with analyzing the data with visualizations, the group was able to begin to understand relationships between variables, test hypotheses about those relationships, and eventually create a high-performing logistic regression model. We trained the final model using observations from the 1997 through 2015 seasons, then tested it on data from the 2016 season. It achieved an accuracy of 86.67% in testing.

During this team's work, we discovered the most powerful predictor for a team making it to the playoffs was the team's On-base Percentage (OBP) – how frequently a team's batters reach base per plate appearance. In our research, this group discovered that for every additional percentage point increase in OBP, the estimated odds of that team progressing to the playoffs is multiplied by an estimated 3.49, assuming all other variables remain constant.

Our analysis confirms that both our models meet all necessary statistical criteria for validity. This means that these models are reliable in predicting team success both in win percentage and prediction of playoff appearance because the relationships between the predictor variables and response variables are not simply due to chance or flawed experimentation. Nevertheless, it is important to note the nuances associated with this group's work.

First, since our data spans the years from 1997 to 2016, these models' reliability is limited to this time frame, and to use it for analysis outside of this timeframe, one must also consider changes that may occur in the rules of MLB baseball, trending team strategy changes, regulation changes in the sport (and so on) that may have an impact on model performance. The second notable nuance to consider is that one could argue that creating separate and unique observations for each team-year combination potentially violates the independence assumption if, for example, a team were to employ a successful strategy across multiple years. Correcting for this level of nuance falls outside of the scope of this project but is an important characteristic to consider.

Although there are nuances to the models created in our research, this group is confident that these models would be useful to MLB team owners and managers in making trustworthy predictions that would be useful in real-world decision-making scenarios. While it is important to recognize limitations, it is also important to remind ourselves that no model is perfect, and the ability to predict a Major League Baseball (MLB) team's win percentage and playoff status using known stats can be a powerful tool if taken advantage of properly.

---

2. Data description:

a. Data source

The final dataframes we use to fit our models, *Data_TeamsIn* and *Data_Teams*, cover the 1997 through 2016 seasons and treat each team-season combination as individual observations.

Data for this project come from the Lahman Baseball Database, an R package containing several team and player statistics across MLB seasons from 1871 to 2021. The database is named after the creator, Sean Lahman, and now a team of researchers maintains it. It covers a broad range of information and, as a relational database, has links across different tables with unique codes (e.g., playerID).

We combine several tables from the database, including:

- Teams – Annual statistics and standings for teams.
- Salaries – Player salary data.
- SeriesPost – postseason series information.

We use the Teams table as our main data frame. In addition, we use the Salaries table to create a variable for total spend `(Spend)` per team by summing player salaries for each team and each season; and the SeriesPost table to create the binary variable `(playoffs)` for logistic regression: 0 if a team did not make it to the postseason, and 1 if a team did make it to the postseason.

In addition to merging these three tables, we create variables to combine predictors and reduce model complexity.

- Team batting average `(Bavg)` = Hits / At Bats
- Winning percentage `(WinP)` = Wins / Games
- On Base Percentage `(OBP)` = (Hits + Walks + Hit by Pitch) / (At Bats + Walks + Hit by Pitch + Sacrifice Flies)

b. Variable descriptions

```
## # A tibble: 56 × 2
##    Variable       Description
##    <chr>          <chr>
##  1 yearID         Year
##  2 lgID           League; a factor with levels AA AL FL NL PL UA
##  3 teamID         Team
##  4 franchID       Franchise
##  5 divID          Team’s division; a factor with levels C E W
##  6 lgIDdivID      Concat League and Division ID's
##  7 team_yr_ID     Concat team ID and year ID
##  8 salary         Team salary for a season
##  9 Spend          Total team salary for 1997 to 2016
## 10 Rank           Position in final standings
## 11 G              Games played
## 12 Ghome          Games played at home
## 13 W              Wins
## 14 L              Losses
## 15 DivWin         Division Winner (Y or N)
## 16 WCWin          Wild Card Winner (Y or N)
## 17 LgWin          League Champion (Y or N)
## 18 WSWin          World Series Winner (Y or N)
## 19 R              Runs scored
## 20 AB             At bats
## 21 H              Hits by batters
## 22 X2B            Doubles
## 23 X3B            Triples
## 24 HR             Homeruns by batters
## 25 BB             Walks by batters
## 26 SO             Strikeouts by batters
## 27 SB             Stolen bases
## 28 CS             Caught stealing
## 29 HBP            Batters hit by pitch
## 30 SF             Sacrifice flies
## 31 RA             Opponents runs scored
## 32 ER             Earned runs allowed
## 33 ERA            Earned run average
## 34 CG             Complete games
## 35 SHO            Shutouts
## 36 SV             Saves
## 37 IPouts         Outs Pitched (innings pitched x 3)
## 38 HA             Hits allowed
## 39 HRA            Homeruns allowed
## 40 BBA            Walks allowed
## 41 SOA            Strikeouts by pitchers
## 42 E              Errors
## 43 DP             Double Plays
## 44 FP             Fielding percentage
## 45 name           Team's full name
## 46 park           Name of team's home ballpark
## 47 attendance     Home attendance total
## 48 BPF            Three-year park factor for batters
## 49 PPF            Three-year park factor for pitchers
## 50 teamIDBR       Team ID used by Baseball Reference website
## 51 teamIDlahman45 Team ID used in Lahman database version 4.5
## 52 teamIDretro    Team ID used by Retrosheet
## 53 Bavg           Team batting average
## 54 WinP           Winning percentage
## 55 OBP            On Base Percentage
## 56 StealP         Stealing Percentage
```

---

3. First question of interest involving linear regression

a. Introduction:

i. First question of interest

Which team stats are useful for accurately predicting a team’s winning percentage for the seasons spanning 1997 to 2016?

ii. Why it's worth exploring this question

Considering a team owner's perspective, are there any predictors a team can develop to improve the outcome of its season? The book Moneyball by Michael Lewis documents how Oakland Athletics' General Manager Billy Beane emphasized the importance of OBP over many other stats when considering which players to sign. There was a focus on finding players that could "get on base." Our first question of interest expands that line of thinking to consider if there are other so-called 'low-hanging fruit' or predictors that significantly impact a team's winning percentage. Exploring unexpected predictors may provide interesting insights for this question, which we will examine in the EDA. For example, does the number of left-handed hitters on a team or average team height correlate with the winning percentage or improve our predictive model? Additionally, knowing whether money predicts success, however indirect or unknown the mechanism of its impact, is an object of general interest for owners, league officials, and spectators, even if determining its causal impact is outside the scope of this study. These are qualities that make the question worth exploring.

b. Data Visualizations:

i. Data-wrangling

Several data-wrangling processes were necessary to produce visualizations for this section. First, we imported the *Teams* table from the *Lahman* package to create our main dataframe, *Data_Teams*. We then filtered the dataframe to seasons between 1997 and 2016 using the `filter` function on the **`yearID`** column. At this stage of the data-wrangling process, *Data_Teams* had 49 variables. We significantly reduce this number by removing unnecessary variables, including:

- **`Ghome`**: The number of home games played.
- **`WSWin`**: A factor-type variable for whether a team won the World Series.
- **`WCWin`**: A factor-type variable for whether a team is a Wild Card winner.
- **`DivWin`**: A factor-type variable for whether a team won its division.
- **`name`**: The name of the team.
- **`franchID`**: An identifier for franchise name.
- **`park`**: The name of the team's home stadium.
- **`BPF`**: Three-year park factor for batters.
- **`PPF`**: There-year park factor for pitchers.
- **`teamIBDR`**: Another team identifier for a certain website.
- **`teamIDretro`**: Another team identifier for an older version of the database.
- **`teamIDlahman45`**: Another team identifier for an older version of the database.
- **`attendance`**: Home attendance total.
- **`LgWin`**: An factor-type variable for whether the team won their league.
- **`Rank`**: A team's position in the final standings for a given season.

This reduced the *Data_Teams* dataframe to 34 variables. We further reduced *Data_Teams* based on certain assumptions that are contextually dependent on our question of interest and based on a review of the relevant literature. Further, we combined several of the indicator variables as explained below.

We created a unique identifier for each team and season combination, **`team_year_ID`**, and calculated each observation's **`OBP`**, **`WinP`**, and **`StealP`** and added these variables to *Data_Teams*.

With these newly created variables, we further reduced the *Data_Teams* dataframe to avoid overfitting our models.

- We removed **`H`**, **`BB`**, **`HBP`**, **`AB`**, and **`SF`**, given that **`OBP`** effectively combines these variables into a single metric using the formula (**`H`** + **`BB`** + **`HBP`**) / (**`AB`** + **`BB`** + **`HBP`** + **`SF`**).
- We removed **`W`**, **`L`**, **`CG`**, and **`G`**, given that **`WinP`** combines these using the formula (**`W`** / (**`W`** + **`L`**)).
- We removed **`SB`** and **`CS`**, given that **`StealP`** combines these using the formula (**`SB`** / (**`SB`** + **`CS`**)).
- We removed **`SHO`** because shutouts (e.g., the opponent scores no runs) may be too closely related to winning percentage (you win a game with a shutout). This predictor may dominate other predictors in determining win percentage.

At this point in the data-wrangling process, the *Data_Teams* dataframe has 25 variables. We further reduce *Data_Teams* using several techniques. For example, we used the `corr_cross` function from the *lares* package to create a horizontal bar chart measuring the correlation between predictor variables to examine multicollinearity issues. This initial plot shows that **`E`** and **`FP`** are almost perfectly negatively correlated at -.988. We chose to drop **`E`** as a predictor and retain **`FP`** based on the findings of James, M., and Wells, A. (2008). In short, **`FP`** is a more comprehensive metric for team defensive performance. The initial correlation bar chart also shows that **`ER`** and **`ERA`** are highly positively correlated with **`HA`**. This makes sense contextually given that the number of earned runs allowed and earned run average would increase alongside the number of hits allowed. Along similar lines, **`ER`** and **`ERA`** are highly positively correlated with **`HRA`**, which also makes sense, given that more home runs allowed would mean more earned runs and a higher earned run average. For these reasons, we removed **`ER`**, **`ERA`**, and **`HA`**.

At this point in the data-wrangling process, there are 20 variables, including our response variable, in the *Data_Teams* dataframe. Now that we have significantly reduced our dataframe, we created a second horizontal bar chart using the `corr_cross` function from the *lares* package to examine if there are other highly correlated indicators that we should consider removing. This second correlation bar chart shows that **`X2B`** and **`OBP`** are positively correlated. This makes sense, given that a batter hitting a double would advance that batter onto base (second base). Based on Pinheiro, R. and Szymanski, S. (2022) and the context of our question of interest, we chose not to include **`X2B`** and **`X3B`**; double and triples are relatively rare events and thus not a proper measure of player (or, in the case of this study, team) offensive ability. The breakdown of the rate of occurrence for each of these is listed below.

Percent of **`AB`** for the 1997-2016 seasons (*these do not sum to 100%; batters can strikeout or be hit by a pitch*):

- **`H`**: 26.20%
- **`X2B`**: 5.20%
- **`X3B`**: 0.55%
- **`HR`**: 3.04%
- **`BB`**: 9.52%

The next step in wrangling the data for our linear regression visualizations, and the models, was to import the *Salaries* table to create the *Data_sal* dataframe. We then used the `filter` function to include only seasons between 1997 and 2016. We then created a vector, *Spend*, from the *Data_sal* dataframe and used the `group_by` function to group by **`yearID`** and **`teamID`** and the `summarize` function to sum the salaries of each observation, creating the **`Spend`** variable. We then used the `paste` function to create the **`team_yr_ID`** column for this vector. Finally, we used the `subset` function to remove **`yearID`** and **`teamID`**.

We combined the *Total_spend* vector with the *Data_Teams* dataframe using the `inner_join` function based on **`team_year_ID`**.

At this point in the data-wrangling process, *Data_Teams* has 13 variables, including the response variable, **`WinP`**.

ii. Data visualizations

	HR	SO	SV	IPouts	HRA	BBA	SOA	DP	FP	OBP	WinP	StealP	Spend
HR	1.00	0.06	0.14	0.07	0.28	0.05	-0.01	0.04	0.02	0.48	0.40	-0.01	0.12
SO	0.06	1.00	0.00	0.13	-0.11	-0.16	0.43	-0.17	0.05	-0.43	-0.13	0.08	0.12
SV	0.14	0.00	1.00	0.38	-0.32	-0.34	0.29	-0.24	0.22	0.08	0.65	0.03	0.21
IPouts	0.07	0.13	0.38	1.00	-0.33	-0.27	0.40	-0.14	0.30	0.04	0.48	0.09	0.16
HRA	0.28	-0.11	-0.32	-0.33	1.00	0.31	-0.36	0.12	-0.18	0.16	-0.38	-0.07	-0.18
BBA	0.05	-0.16	-0.34	-0.27	0.31	1.00	-0.33	0.29	-0.33	0.17	-0.41	-0.11	-0.37
SOA	-0.01	0.43	0.29	0.40	-0.36	-0.33	1.00	-0.49	0.31	-0.24	0.34	0.15	0.48
DP	0.04	-0.17	-0.24	-0.14	0.12	0.29	-0.49	1.00	-0.07	0.13	-0.20	-0.12	-0.23
FP	0.02	0.05	0.22	0.30	-0.18	-0.33	0.31	-0.07	1.00	-0.01	0.36	0.14	0.37
OBP	0.48	-0.43	0.08	0.04	0.16	0.17	-0.24	0.13	-0.01	1.00	0.49	0.00	-0.03
WinP	0.40	-0.13	0.65	0.48	-0.38	-0.41	0.34	-0.20	0.36	0.49	1.00	0.14	0.32
StealP	-0.01	0.08	0.03	0.09	-0.07	-0.11	0.15	-0.12	0.14	0.00	0.14	1.00	0.21
Spend	0.12	0.12	0.21	0.16	-0.18	-0.37	0.48	-0.23	0.37	-0.03	0.32	0.21	1.00

Figure 1.a Correlation Matrix

The table above shows the correlations of all variables used to develop our models.

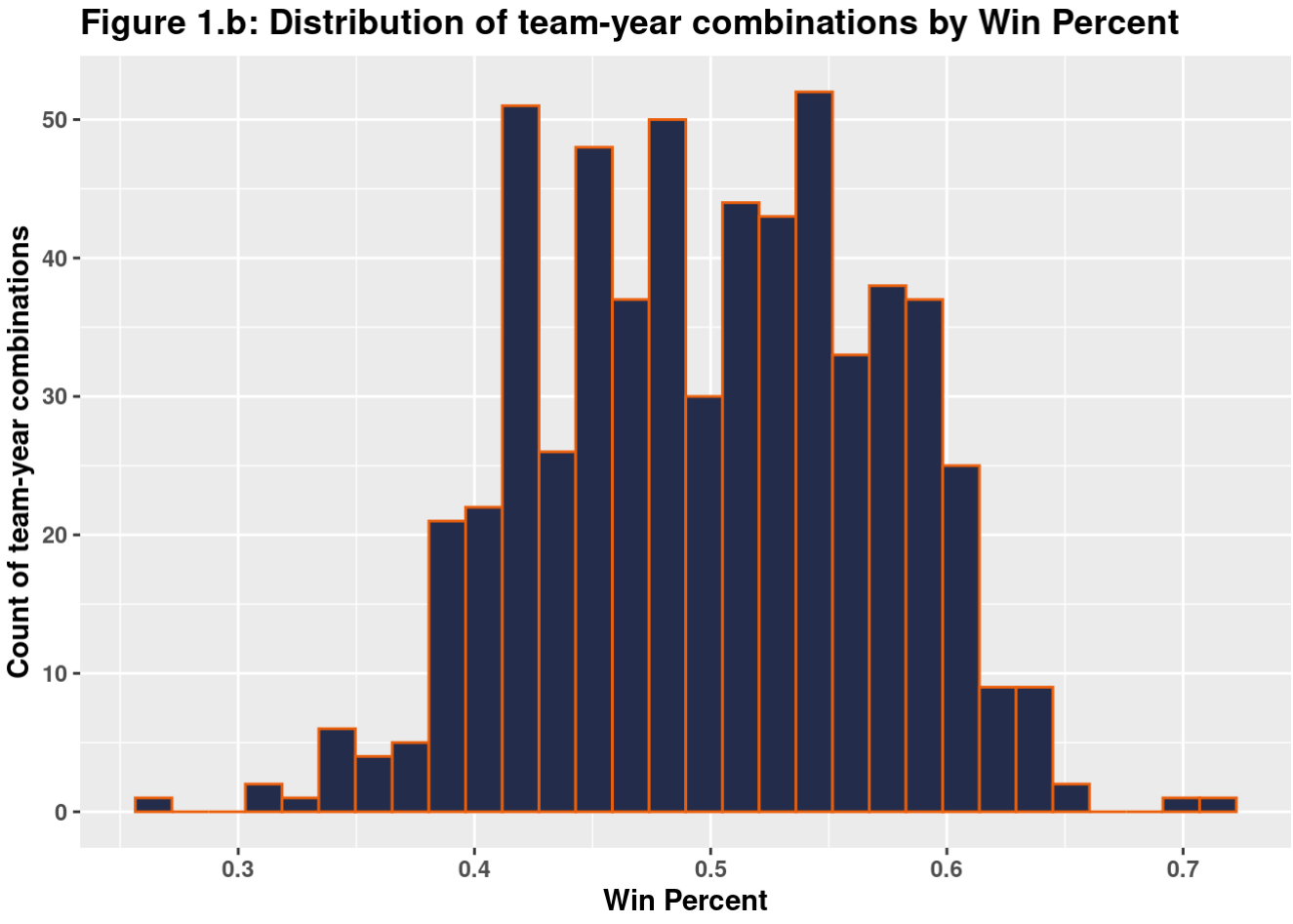


Figure 1.b is a histogram of **`WinP`** for the 1997-2016 seasons. Although the histogram shows that **`WinP`** does not follow a perfect standard normal distribution, it does tell us that the observations in our dataframe generally follow a bell-shaped curve in terms of **`WinP`**.

Figure 1.c: WinP against each Predictor by divIDlgID

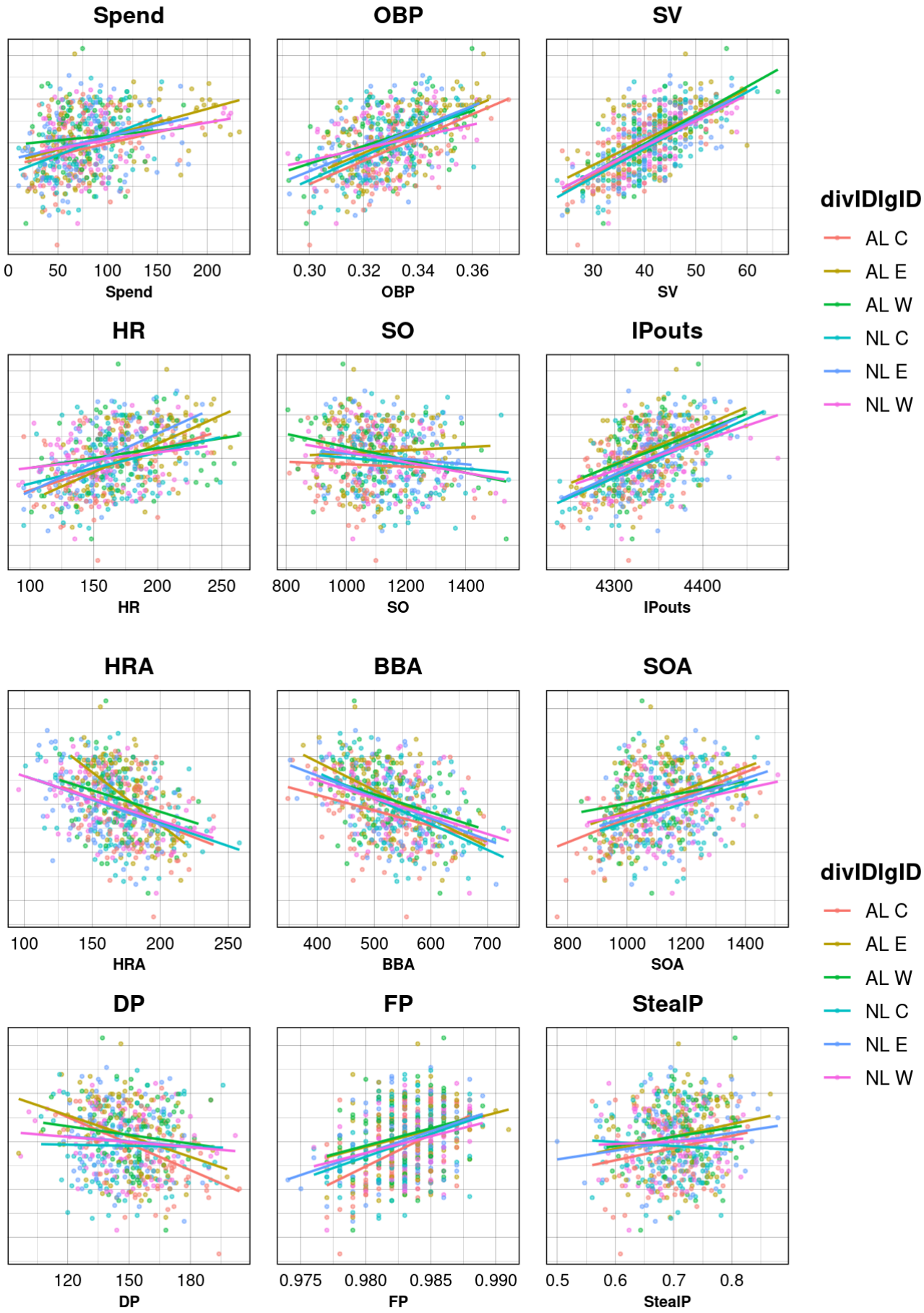


Figure 1.c is a combined scatterplot of **`WinP`** against each of the predictors colored by each combination of **`divID`** and **`lgID`** (which we call **`lgIDdivID`**). We used this scatterplot to guide our hypothesis testing for whether to include an interaction term for **`lgIDdivID`** and any of the predictors in our models. The regression lines between each predictor and **`WinP`** appear close to parallel for **`lgIDdivID`** for all predictors except **`HR`**, **`DP`**, **`FP`**, and **`StealP`**.

Figure 1.d: Impact of SV and OBP on WinP by lgIDdivID

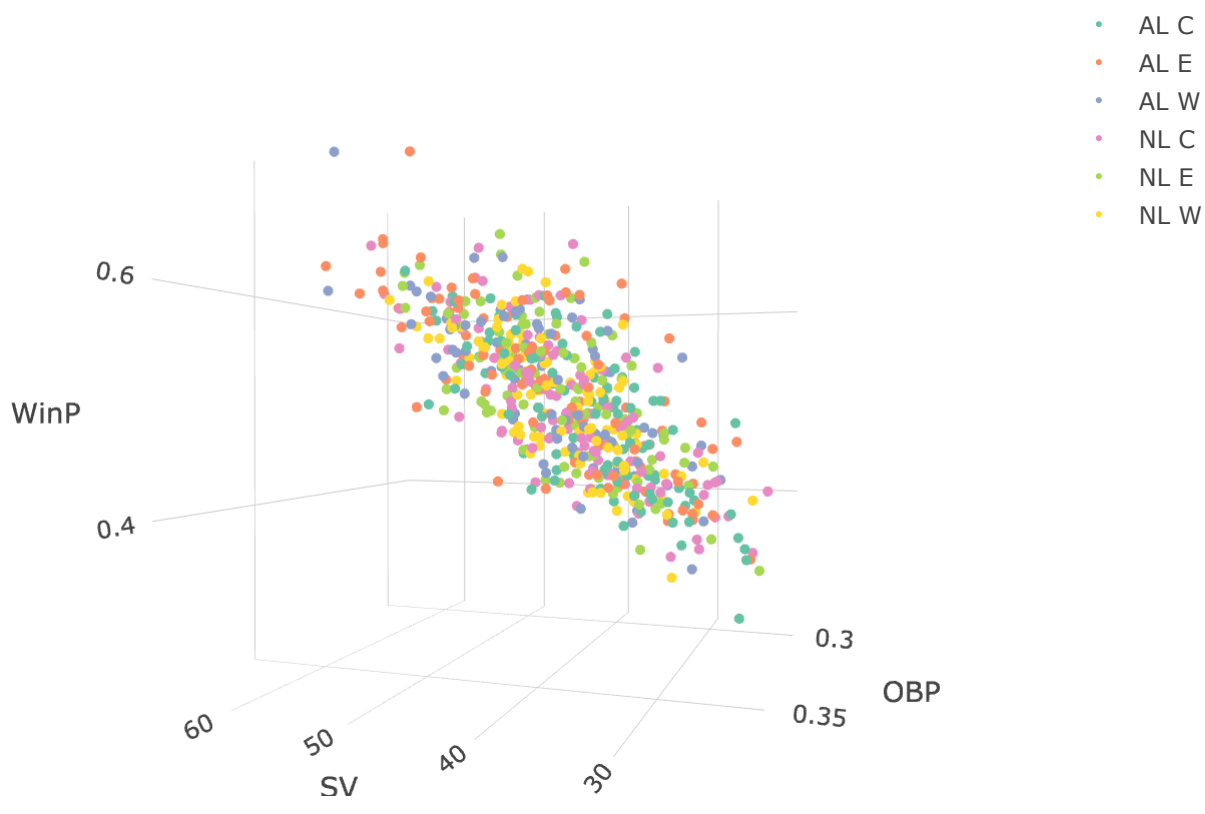


Figure 1.d is a 3D scatterplot of **`WinP`** against both **`OBP`** and **`SV`** and colored by each **`lgIDdivID`**. We used this scatterplot to assess the relationship between the two predictors we assumed to have the most impact on our response and to help inform our model-building regarding interaction terms. We can see that as both OBP and SV increase, WinP increases.

c. Model Building:

We ran a few models with interaction terms between **`lgIDdivID`** and each of the predictors identified in Figure 1.c as potentially having an interaction, including **`HR`**, **`DP`**, **`FP`**, and **`StealP`**.

Testing Interaction Terms:

None of the t-tests for the interaction terms were significant when including an interaction term between **`HR`** and **`lgIDdivID`** or between **`DP`** and **`lgIDdivID`**. This indicates that the intercepts and slopes for American League East (AL_E), American League West (AL_W), National League Central (NL_C), National League East (NL_E), and National League West (NL_W) are not significantly different from the reference class, American League Central (AL_C), for **`HR`** or **`DP`**. The model summaries for those two models are shown below.

Note: American League Central (AL_C) is the reference class for **`lgIDdivID`**.

Model summary with an interaction term between **`lgIDdivID`** and **`HR`**:

```
##
## Call:
## lm(formula = WinP ~ HR * lgIDdivID + SV + SO + IPouts + HRA +
##     BBA + SOA + DP + FP + StealP + Spend, data = Data_Teamsln)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.102533 -0.022535 -0.001326  0.022901  0.113874
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -2.981e+00  5.873e-01  -5.076 5.20e-07 ***
## HR                9.991e-04  1.047e-04   9.546  < 2e-16 ***
## lgIDdivIDAL E     2.170e-02  2.690e-02   0.807 0.420169
## lgIDdivIDAL W     3.782e-02  2.536e-02   1.491 0.136472
## lgIDdivIDNL C     5.238e-04  2.589e-02   0.020 0.983868
## lgIDdivIDNL E    -2.806e-03  2.560e-02  -0.110 0.912754
## lgIDdivIDNL W    -4.277e-04  2.601e-02  -0.016 0.986887
## SV                3.223e-03  2.390e-04  13.485  < 2e-16 ***
## SO               -1.553e-04  1.249e-05 -12.428  < 2e-16 ***
## IPouts            2.383e-04  4.326e-05   5.510 5.43e-08 ***
## HRA              -7.484e-04  7.154e-05 -10.461  < 2e-16 ***
## BBA              -1.796e-04  2.697e-05  -6.659 6.45e-11 ***
## SOA               6.848e-05  1.731e-05   3.957 8.54e-05 ***
## DP               -3.212e-05  9.920e-05  -0.324 0.746193
## FP                2.442e+00  6.018e-01   4.058 5.63e-05 ***
## StealP            9.892e-02  2.588e-02   3.822 0.000147 ***
## Spend            -2.052e-05  4.565e-05  -0.449 0.653258
## HR:lgIDdivIDAL E -1.419e-04  1.509e-04  -0.941 0.347228
## HR:lgIDdivIDAL W -2.151e-04  1.462e-04  -1.472 0.141694
## HR:lgIDdivIDNL C  1.159e-05  1.524e-04   0.076 0.939416
## HR:lgIDdivIDNL E  4.513e-05  1.554e-04   0.290 0.771624
## HR:lgIDdivIDNL W  2.346e-05  1.557e-04   0.151 0.880267
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03506 on 576 degrees of freedom
## Multiple R-squared:  0.7616, Adjusted R-squared:  0.7529
## F-statistic: 87.64 on 21 and 576 DF,  p-value: < 2.2e-16
```

Model summary with an interaction term between **`lgIDdivID`** and **`DP`**:

```
##
## Call:
## lm(formula = WinP ~ DP * lgIDdivID + SV + HR + IPouts + HRA +
##     BBA + SOA + SO + FP + StealP + Spend, data = Data_Teamsln)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.116140 -0.022418 -0.000867  0.021726  0.120239
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -3.028e+00  5.942e-01  -5.095 4.74e-07 ***
## DP               -3.119e-04  2.109e-04  -1.479 0.139765
## lgIDdivIDAL E    -2.066e-02  4.361e-02  -0.474 0.635823
## lgIDdivIDAL W    -4.212e-02  4.863e-02  -0.866 0.386795
## lgIDdivIDNL C    -6.120e-02  4.150e-02  -1.475 0.140871
## lgIDdivIDNL E    -8.305e-02  4.756e-02  -1.746 0.081317 .
## lgIDdivIDNL W    -3.104e-02  4.241e-02  -0.732 0.464434
## SV                3.127e-03  2.397e-04  13.045  < 2e-16 ***
## HR                9.521e-04  4.884e-05  19.492  < 2e-16 ***
## IPouts            2.509e-04  4.312e-05   5.819 9.85e-09 ***
## HRA              -7.215e-04  7.106e-05 -10.154  < 2e-16 ***
## BBA              -1.804e-04  2.710e-05  -6.658 6.47e-11 ***
## SOA               6.613e-05  1.729e-05   3.824 0.000146 ***
## SO               -1.550e-04  1.254e-05 -12.358  < 2e-16 ***
## FP                2.488e+00  6.062e-01   4.104 4.64e-05 ***
## StealP            9.887e-02  2.568e-02   3.850 0.000131 ***
## Spend            -2.034e-05  4.568e-05  -0.445 0.656251
## DP:lgIDdivIDAL E  1.017e-04  2.850e-04   0.357 0.721268
## DP:lgIDdivIDAL W  2.763e-04  3.150e-04   0.877 0.380909
## DP:lgIDdivIDNL C  4.185e-04  2.686e-04   1.558 0.119785
## DP:lgIDdivIDNL E  5.839e-04  3.158e-04   1.849 0.064998 .
## DP:lgIDdivIDNL W  2.200e-04  2.760e-04   0.797 0.425791
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03506 on 576 degrees of freedom
## Multiple R-squared:  0.7616, Adjusted R-squared:  0.7529
## F-statistic: 87.63 on 21 and 576 DF,  p-value: < 2.2e-16
```

The t-tests for several interaction terms are statistically significant when running models with interaction terms between **`lgIDdivID`** and **`FP`** and **`lgIDdivID`** and **`StealP`**.

Model summary with an interaction term between **`lgIDdivID`** and **`FP`**:

```
##
## Call:
## lm(formula = WinP ~ FP * lgIDdivID + SV + HR + IPouts + HRA +
##     BBA + SOA + SO + DP + StealP + Spend, data = Data_Teamsln)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.116416 -0.022850 -0.000603  0.021678  0.116656
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -5.946e+00  1.350e+00  -4.405 1.26e-05 ***
## FP                5.431e+00  1.373e+00   3.955 8.61e-05 ***
## lgIDdivIDAL E     3.406e+00  1.845e+00   1.846 0.065362 .
## lgIDdivIDAL W     3.578e+00  1.945e+00   1.839 0.066375 .
## lgIDdivIDNL C     3.067e+00  1.791e+00   1.712 0.087386 .
## lgIDdivIDNL E     3.959e+00  1.751e+00   2.261 0.024134 *
## lgIDdivIDNL W     2.977e+00  1.858e+00   1.602 0.109641
## SV                3.178e-03  2.378e-04  13.366  < 2e-16 ***
## HR                9.424e-04  4.838e-05  19.478  < 2e-16 ***
## IPouts            2.475e-04  4.296e-05   5.760 1.37e-08 ***
## HRA              -7.353e-04  7.086e-05 -10.377  < 2e-16 ***
## BBA              -1.817e-04  2.681e-05  -6.777 3.04e-11 ***
## SOA               6.678e-05  1.738e-05   3.843 0.000135 ***
## SO               -1.577e-04  1.254e-05 -12.572  < 2e-16 ***
## DP               -2.382e-05  9.836e-05  -0.242 0.808713
## StealP            1.018e-01  2.585e-02   3.939 9.17e-05 ***
## Spend            -1.811e-05  4.532e-05  -0.400 0.689624
## FP:lgIDdivIDAL E -3.469e+00  1.877e+00  -1.849 0.065028 .
## FP:lgIDdivIDAL W -3.640e+00  1.979e+00  -1.839 0.066417 .
## FP:lgIDdivIDNL C -3.118e+00  1.823e+00  -1.710 0.087742 .
## FP:lgIDdivIDNL E -4.023e+00  1.781e+00  -2.258 0.024290 *
## FP:lgIDdivIDNL W -3.025e+00  1.890e+00  -1.601 0.109956
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03503 on 576 degrees of freedom
## Multiple R-squared:  0.7621, Adjusted R-squared:  0.7534
## F-statistic: 87.86 on 21 and 576 DF,  p-value: < 2.2e-16
```

The model summary for the model with interaction between **`lgIDdivID`** and **`FP`** shows that the t-tests for β2 and β17, β3 and β18, β4 and β19, and β6 and β21 are insignificant, which indicates that the intercepts and slopes for American AL_E, AL_W, NL_C, and NL_W are not significantly different from the reference class, AL_C. However, we note a significant t-test for β5 and β20 (NL_E). We conduct a partial F-test to determine whether we can drop the interaction terms β17, β18, β19, and β21.

- **Ho**: β17 = β18 = β19 = β21 = 0
- **Ha**: at least one of the coefficients in Ho is not zero

```
## Analysis of Variance Table
##
## Model 1: WinP ~ FP + lgIDdivID + SV + HR + IPouts + HRA + BBA + SOA +
##     SO + DP + StealP + Spend
## Model 2: WinP ~ FP * lgIDdivID + SV + HR + IPouts + HRA + BBA + SOA +
##     SO + DP + StealP + Spend
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1    581 0.71407
## 2    576 0.70663  5 0.0074408 1.2131 0.3015
```

The resulting F-statistic is (1.2131), and its associated p-value is high (0.3015). So, we fail to reject our null hypothesis and determine that we can drop the interaction term.

When running models with interaction terms between **`lgIDdivID`** and **`StealP`**, we note that the t-tests for several interaction terms are statistically significant.

Model summary with an interaction term between **`lgIDdivID`** and **`StealP`**:

```
##
## Call:
## lm(formula = WinP ~ StealP * lgIDdivID + SV + HR + IPouts + HRA +
##     BBA + SOA + SO + DP + FP + Spend, data = Data_Teamsln)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.119833 -0.021036 -0.000778  0.022210  0.111130
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -3.181e+00  5.876e-01  -5.414 9.07e-08 ***
## StealP                2.260e-01  6.290e-02   3.593 0.000355 ***
## lgIDdivIDAL E         3.961e-02  6.291e-02   0.630 0.529160
## lgIDdivIDAL W         7.475e-02  6.801e-02   1.099 0.272242
## lgIDdivIDNL C         1.894e-01  6.147e-02   3.081 0.002162 **
## lgIDdivIDNL E         1.191e-01  5.791e-02   2.057 0.040131 *
## lgIDdivIDNL W         9.856e-02  6.359e-02   1.550 0.121692
## SV                    3.151e-03  2.372e-04  13.284  < 2e-16 ***
## HR                    9.559e-04  4.825e-05  19.812  < 2e-16 ***
## IPouts                2.518e-04  4.282e-05   5.881 6.93e-09 ***
## HRA                  -7.411e-04  7.039e-05 -10.528  < 2e-16 ***
## BBA                  -1.844e-04  2.679e-05  -6.884 1.53e-11 ***
## SOA                   6.155e-05  1.756e-05   3.506 0.000491 ***
## SO                   -1.537e-04  1.249e-05 -12.305  < 2e-16 ***
## DP                   -6.419e-05  9.760e-05  -0.658 0.510984
## FP                    2.519e+00  5.989e-01   4.206 3.01e-05 ***
## Spend                -2.882e-05  4.674e-05  -0.617 0.537677
## StealP:lgIDdivIDAL E -6.402e-02  8.922e-02  -0.718 0.473326
## StealP:lgIDdivIDAL W -1.073e-01  9.633e-02  -1.114 0.265718
## StealP:lgIDdivIDNL C -2.686e-01  8.826e-02  -3.044 0.002442 **
## StealP:lgIDdivIDNL E -1.644e-01  8.210e-02  -2.003 0.045670 *
## StealP:lgIDdivIDNL W -1.369e-01  9.113e-02  -1.502 0.133539
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03488 on 576 degrees of freedom
## Multiple R-squared:  0.7641, Adjusted R-squared:  0.7555
## F-statistic: 88.85 on 21 and 576 DF,  p-value: < 2.2e-16
```

Next, we perform a similar partial F-test for the model with interaction between **`lgIDdivID`** and **`StealP`**. The model summary for this model shows insignificant t-tests for β2 and β17, β3 and β18, and β6 and β21, indicating that the intercepts and slopes for AL_E, AL_W, and NL_W are not significantly different from the reference class, AL_C. However, we note a significant t-test for β4 and β19, and β5 and β20, which indicates that the intercepts and slopes for NL_C and NL_E are significantly different from AL_C. We conduct a partial F-test to determine whether we can drop the interaction terms β17, β18, and β21.

- **Ho**: β17 = β18 = β21 = 0
- **Ha**: at least one of the coefficients in Ho is not zero

```
## Analysis of Variance Table
##
## Model 1: WinP ~ StealP + lgIDdivID + SV + HR + IPouts + HRA + BBA + SOA +
##     SO + DP + FP + Spend
## Model 2: WinP ~ StealP * lgIDdivID + SV + HR + IPouts + HRA + BBA + SOA +
##     SO + DP + FP + Spend
##   Res.Df     RSS Df Sum of Sq     F Pr(>F)
## 1    581 0.71407
## 2    576 0.70066  5  0.013411 2.205 0.0524 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The resulting F-statistic is (2.205), and its associated p-value is high (0.0524). Although the p-value is very close to significance, it's larger than 0.05, so we fail to reject our null hypothesis and determine that we can drop the interaction term.

After testing interaction terms and determining that we could drop them, we chose to drop **`lgIDdivID`** and fit an initial model, *full_ln*, with all predictors that remained in *Data_Teams*.

Model summary for the *full_ln* model:

```
##
## Call:
## lm(formula = WinP ~ HR + SO + SV + IPouts + HRA + BBA + SOA +
##     FP + OBP + StealP + DP + Spend, data = Data_Teamsln)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.067781 -0.019781 -0.001772  0.017655  0.079220
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.289e+00  4.583e-01  -7.175 2.20e-12 ***
## HR           5.027e-04  4.330e-05  11.611  < 2e-16 ***
## SO          -6.257e-05  1.076e-05  -5.817 9.86e-09 ***
## SV           3.130e-03  1.855e-04  16.876  < 2e-16 ***
## IPouts       1.820e-04  3.386e-05   5.375 1.11e-07 ***
## HRA         -6.867e-04  5.438e-05 -12.629  < 2e-16 ***
## BBA         -2.193e-04  2.029e-05 -10.810  < 2e-16 ***
## SOA          9.299e-05  1.356e-05   6.859 1.77e-11 ***
## FP           2.337e+00  4.688e-01   4.984 8.21e-07 ***
## OBP          1.953e+00  1.033e-01  18.910  < 2e-16 ***
## StealP       7.041e-02  1.998e-02   3.525 0.000457 ***
## DP          -2.372e-05  7.583e-05  -0.313 0.754487
## Spend       -3.024e-05  3.513e-05  -0.861 0.389659
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02759 on 585 degrees of freedom
## Multiple R-squared:  0.8501, Adjusted R-squared:  0.847
## F-statistic: 276.4 on 12 and 585 DF,  p-value: < 2.2e-16
```

After fitting the *full_ln* model, we examined each predictor's Variance Inflation Factor (VIF). The VIFs, displayed below, are low for all predictors in the *full_ln* model.

```
##     HR     SO     SV IPouts    HRA    BBA    SOA     FP    OBP StealP     DP
##   1.71   1.63   1.43   1.45   1.49   1.47   2.42   1.32   1.84   1.07   1.45
##  Spend
##   1.57
```

The model summary for *full_ln*, however, shows that **`DP`** and **`Spend`** are not statistically significant in the presence of the other predictors, so we conduct a partial F-test to determine whether **`DP`** and **`Spend`** have a significant impact on **`WinP`** when controlling for the other predictors in *full_ln*. First, we fit a reduced model, *reduced1_ln*, using all predictors except **`DP`** and **`Spend`**.

Model summary for the *reduced1_ln* model:

```
##
## Call:
## lm(formula = WinP ~ HR + SO + SV + IPouts + HRA + BBA + SOA +
##     FP + OBP + StealP, data = Data_Teamsln)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.066608 -0.020033 -0.001662  0.017867  0.081397
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.207e+00  4.484e-01  -7.151 2.58e-12 ***
## HR           4.957e-04  4.258e-05  11.640  < 2e-16 ***
## SO          -6.166e-05  1.066e-05  -5.783 1.19e-08 ***
## SV           3.141e-03  1.835e-04  17.114  < 2e-16 ***
## IPouts       1.851e-04  3.338e-05   5.546 4.42e-08 ***
## HRA         -6.828e-04  5.376e-05 -12.700  < 2e-16 ***
## BBA         -2.166e-04  1.939e-05 -11.168  < 2e-16 ***
## SOA          9.075e-05  1.131e-05   8.026 5.51e-15 ***
## FP           2.234e+00  4.544e-01   4.917 1.14e-06 ***
## OBP          1.954e+00  1.032e-01  18.942  < 2e-16 ***
## StealP       6.834e-02  1.971e-02   3.468 0.000563 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02756 on 587 degrees of freedom
## Multiple R-squared:  0.8499, Adjusted R-squared:  0.8473
## F-statistic: 332.3 on 10 and 587 DF,  p-value: < 2.2e-16
```

Next, we used the `anova` function to compare the fit of the *full_ln* model with that of the *reduced1_ln* model.

- **Ho**: β11 = β12 = 0
- **Ha**: at least one of the coefficients in Ho is not 0.

```
## Analysis of Variance Table
##
## Model 1: WinP ~ HR + SO + SV + IPouts + HRA + BBA + SOA + FP + OBP + StealP
## Model 2: WinP ~ HR + SO + SV + IPouts + HRA + BBA + SOA + FP + OBP + StealP +
##     DP + Spend
##   Res.Df     RSS Df  Sum of Sq      F Pr(>F)
## 1    587 0.44596
## 2    585 0.44531  2 0.00065705 0.4316 0.6497
```

The resulting F-statistic was (0.4316), and its associated p-value is high (0.6497). Given these results, we fail to reject the null hypothesis; the partial F-test suggests that we can drop **`DP`** and **`Spend`** and go with the *reduced1_ln* model.

We then cross-validated our model selection process using Automated Search Procedures based on three criteria: Adjusted R-squared, Bayesian Information Criterion (BIC), and Mallow's CP (cp). Each criterion returned models with the same predictors and coefficients as *reduced1_ln*. The results are displayed below.

Coefficients for the models produced using R-squared adjusted, Mallow's CP, and BIC as criteria:

```
##   (Intercept)            HR            SO            SV       IPouts
## -1.0897401692  0.0004916967 -0.0000641435  0.0031353368  0.0002123053
##           HRA           BBA           SOA           OBP
## -0.0006795149 -0.0002430788  0.0001046424  1.9887523370
```

iii. Recommended linear regression model(s)

We recommend the *reduced1_ln* model because it was the most performant model based on several different statistical assessments, including the partial F-test, and automated search procedure methods based on Adjusted R-squared, Bayesian Information Criterion, and Mallow's CP.

The final equation for the recommended linear regression model is:

$$\begin{aligned}\text{WinP} \sim 04.957\text{e}^{-04}(\text{HR}) - 6.166\text{e}^{-05}(\text{SO}) + 3.141\text{e}^{-03}(\text{SV}) + 1.851\text{e}^{-04}(\text{IPouts}) - 6.828\text{e}^{-04}(\text{HRA}) \\ - 2.166\text{e}^{-04}(\text{BBA}) + 9.075\text{e}^{-05}(\text{SOA}) + 2.234(\text{FP}) + 1.954(\text{OBP}) + 6.834\text{e}^{-02}(\text{StealP}) + \epsilon\end{aligned}$$

iv. Assessing regression assumptions and the presence of influential data points

Predictive Performance:

The model might be able to explain 84.41% of the variability in the new observations. The R-squared is 0.8499. Both values are fairly high and close to each other, so the model has good predictive ability.

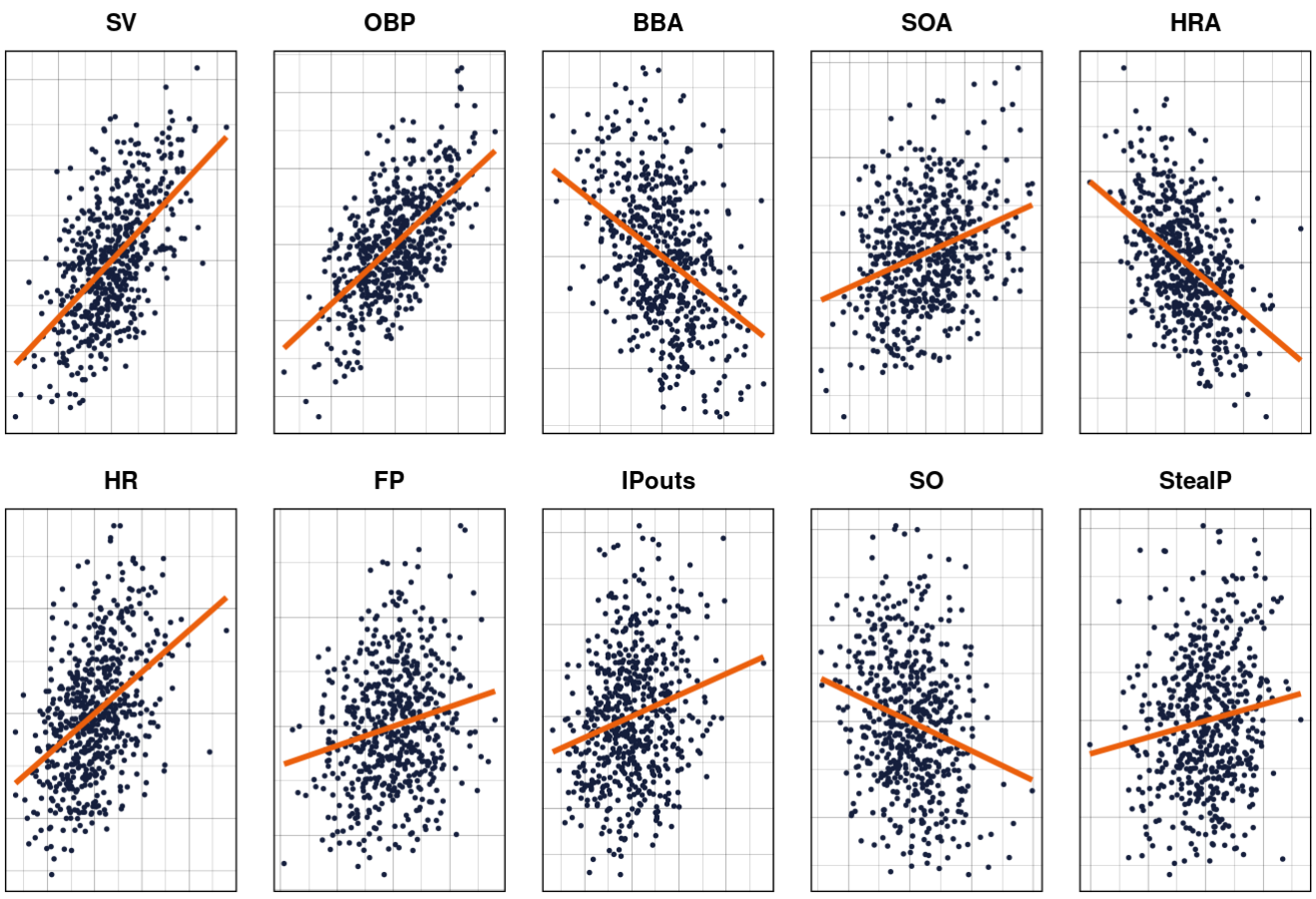
Outlying Observations:

We identified no outlying observations using the Bonferroni procedure to review externally studentized residuals.

Influential Observations:

We identified no influential observations using Cook's distance; however, we identified 27 influential observations using DFFITS.

Figure 1.e: Partial Regression Plots for all Predictors



Per Figure 1.e., the partial regression plots for each predictor, we see that a linear term for each predictor is appropriate based on the patterns of each scatterplot. In particular, `SV`, `OBP`, and `HR` have strong positive linear patterns, and `BBA` and `HRA` show strong negative linear patterns.

d. Conclusions:

i. Discuss how our model(s) answers our question of interest

Based on a review of available predictor variables, which included analyzing correlation, potential interaction terms, partial F-tests, and various automated search procedures, we developed a statistically significant model for predicting baseball win percentage that identified `HR` (Homeruns by batters), `SO` (Strikeouts by batters), `SV` (Saves), `IPouts` (Outs Pitched), `HRA` (Homeruns allowed), `BBA` (Walks allowed), `SOA` (Strikeouts by pitchers), `FP` (Fielding percentage), `OBP` (On-base percentage), and `StealP` (Stolen Base success rate) as significant predictors. This means that these baseball statistics together create a useful model for predicting win percentages, which implies that owners and managers alike can focus on improving these team statistics together to best improve their win percentages.

ii. Provide interesting insights gained about the data

Interestingly, double plays appeared to have a slight negative correlation with wins for all divisions in the MLB based on our initial scatterplot visualizations. We initially believed double plays would be an important predictor for win percentage, but looking deeper, it could be the case that teams with high double plays in a season may indicate poor pitching and/or fielding performance because a team wouldn't even have the opportunity to make a double play if they were consistently striking out batters, denying hits, and making less fielding errors. During the model-building process, we eventually dropped `DP` because this predictor was not statistically significant in our model.

Likewise, we dropped `Spend` (team spend in a season) from our regression model. `Spend` appears to be positively correlated with win percentage based on initial visualizations. `Spend` is a very controversial statistic in baseball news, as there can be a large difference across the MLB in team payroll for a given season. During the 1997 - 2016 period (as well as currently in 2023), the MLB has no salary cap or salary floor. Some teams spend very little, whether because of the owner's decisions, the general manager's strategic decisions, finance issues, lack of income from a strong fanbase, or other reasons. However,

teams with high payrolls can afford higher-valued players, so they are expected to consistently outperform teams with low payrolls. However, we see that the correlation between `Spend` and `WinP` is not as strong as other predictors in our analysis and is eventually deemed statistically insignificant in the presence of the other predictors. In context, some reasoning for this may be that the MLB overvalues "good" players who may be paid more than they are worth in terms of contributing to the team's winning percentage. Also, the influence of young, cheap talent from the draft might weaken the impact of `Spend` on `WinP`. Teams with low payrolls and strong, cheap draft picks may be able to win more than teams with highly compensated star players.

We also dropped `lgIdDivID` (identifier for league + division). The American League East (AL_E) is consistently the strongest MLB division; some data support this based on the trend lines in our initial scatterplots for each predictor. There also seems to be some variation across all divisions in `WinP` across our predictors based on the trend lines in these scatterplots. However, `lgIdDivID` is eventually dropped from the model and does not appear to be a significant predictor of wins.

iii. Challenges we faced

The Lahman MLB database contains 32 tables with dozens of categorical and quantitative variables. We spent significant time researching these variables to understand their context in the game of baseball holistically and their context in potentially being used as predictors for win percentage. We eventually developed a simple yet useful model for predicting win percentage using 10 quantitative predictors based on our understanding of these predictors and our use of various statistical methods. We also were wary of overfitting the model given all the many (often related) predictors at our disposal from the main tables we used from the database.

4.

a. Introduction:

i. Second question of interest

Using the 19 seasons spanning 1997-2015, can we develop an accurate model to predict a team making the 2016 playoffs?

ii. Why this question is worth exploring

Making it to the playoffs, or advancing to the postseason, is one of many measures of an MLB team's success and competitiveness. Knowing which predictors best influence a team's odds of advancing to the postseason is useful for owners, players, and fans alike in understanding how to best allocate resources and strategize to gain the best odds of advancing.

b. Data Visualizations:

i. Data-wrangling

Several data-wrangling processes were necessary to produce visualizations for this section. We completed much of the data-wrangling in the linear regression section, but a few extra steps were necessary to add our response variable, `playoffs`, for the logistic section.

We imported the *SeriesPost* table to get the data necessary to identify whether each observation made it to the playoffs. We filtered the dataframe, *Data_playoffs*, to seasons between 1997 and 2016. We used this dataframe to create our response variable, `playoffs`. To do so, we created two new columns in the *Data_playoffs* dataframe to store the unique identifiers of team-year combinations that made it to the playoffs called `team_yr_ID_w` and `team_yr_ID_l`; we created these two separate columns in the dataframe because our question of interest is about making it to the playoffs generally, not whether an observation (team-year combination) won or lost a given round. Next, we used the `mutate` function to create two variables in the *Data_Teams* dataframe called `playoffs_win` and `playoffs_lose`, and matched `team_yr_ID_w` from the *Data_playoffs* dataframe with `playoffs_win` in the *Data_Teams* dataframe, and did the same for `team_yr_ID_l` and `playoffs_lose`. The next step was to create the actual response variable, `playoffs`, using the `paste` function to combine the `playoffs_win` column with the `playoffs_lose` column in the *Data_Teams* dataframe. At this point, the `playoffs` column in the *Data_Teams* dataframe contains "TRUE TRUE," "FALSE TRUE," or "TRUE FALSE" if the observation made it to the playoffs, and "FALSE FALSE" if the observation did not make it to the playoffs. The final step to create our predictor variable, `playoffs`, was to use Boolean indexing to change instances of "TRUE TRUE," "FALSE TRUE," and "TRUE FALSE" to 1 and instances of "FALSE FALSE" to 0. We removed `playoffs_lose` and `playoffs_win` from the *Data_Teams* dataframe and changed `playoffs` to a factor.

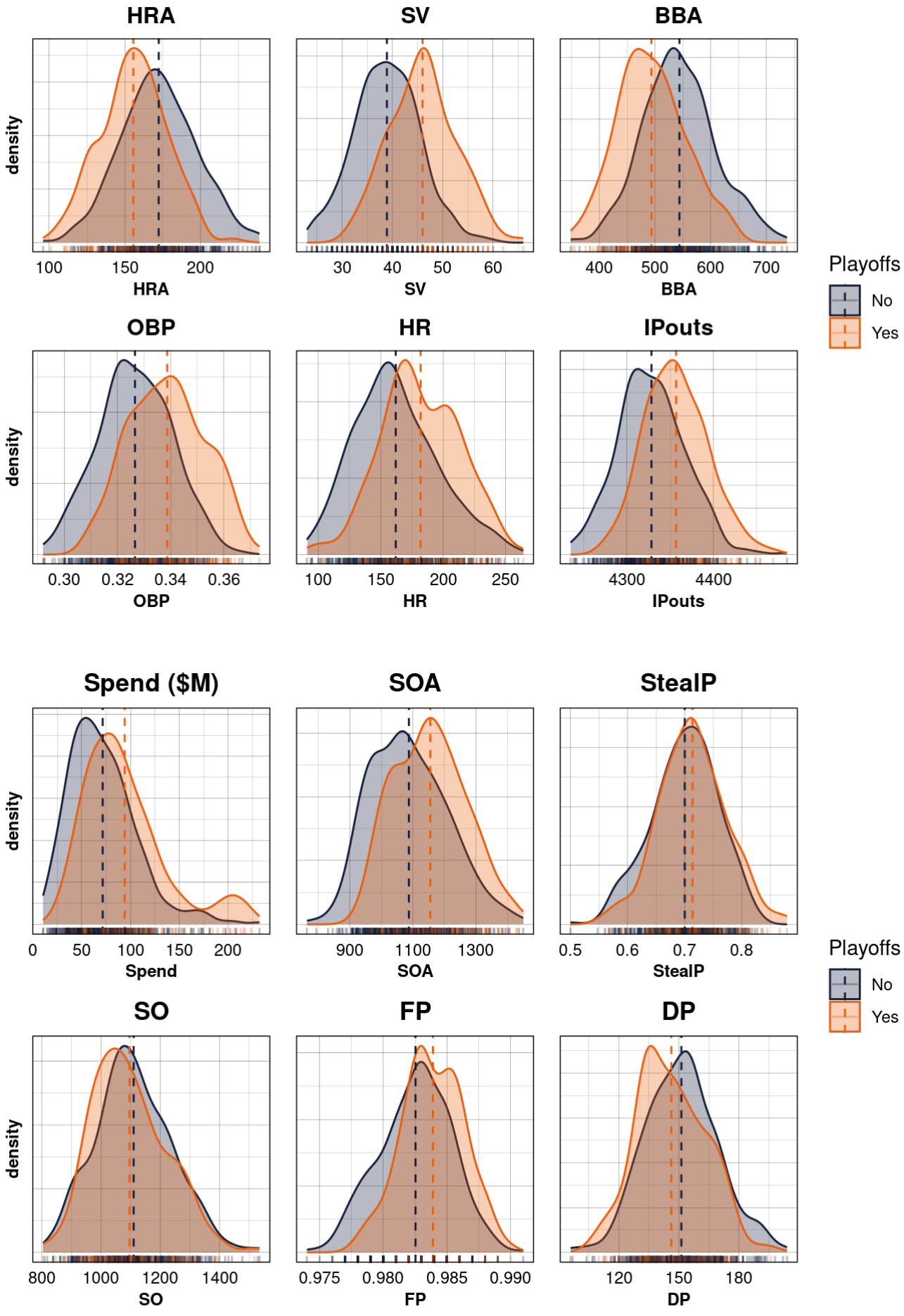
In addition, to help us visualize the data, we added categorical variables that bin teams into "High" and "Low" categories for each numeric predictor based on whether the observation was above or below the sample mean. For example, the `OBP_category` column has categories for "High OBP" and "Low OBP" based on whether the team's `OBP` is higher or lower than the sample mean `OBP`.

The next step was to separate the *Data_Teams* dataframe into two parts; one dataframe for team-year combinations between 1997 and 2015, which we use as training data, and the second dataframe for team-year combinations for 2016, which we use as the test data. We set the test and train data in this way due to the context of our question of interest; we want to predict teams that make it to the playoffs in 2016 using data from 1997-2015.

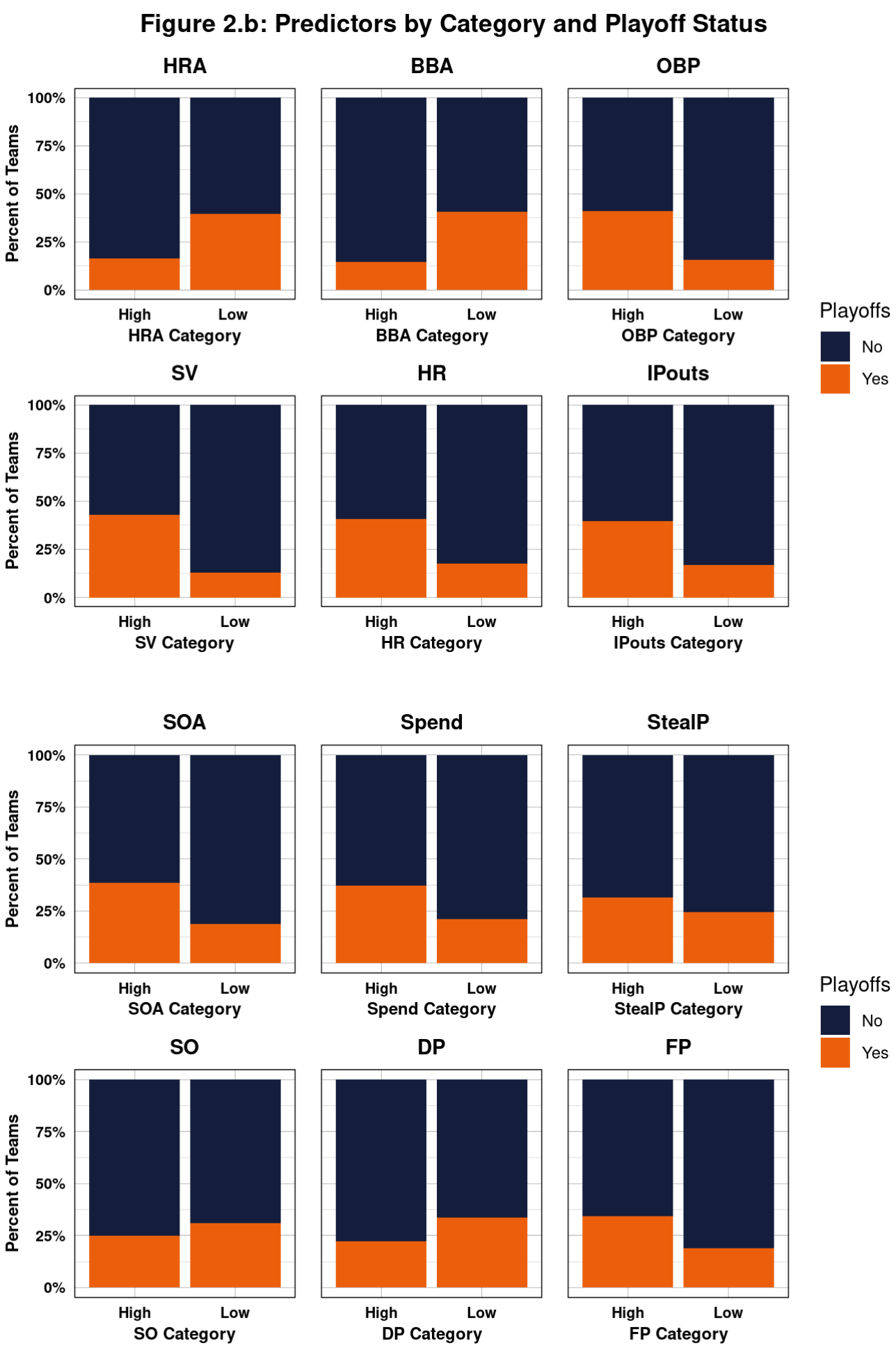
ii. Data visualizations

First, we examine the percentage of teams that make it to the Playoffs while grouping them into higher-than-average and lower-than-average for each predictor.

Figure 2.a: Predictors by Playoff Status



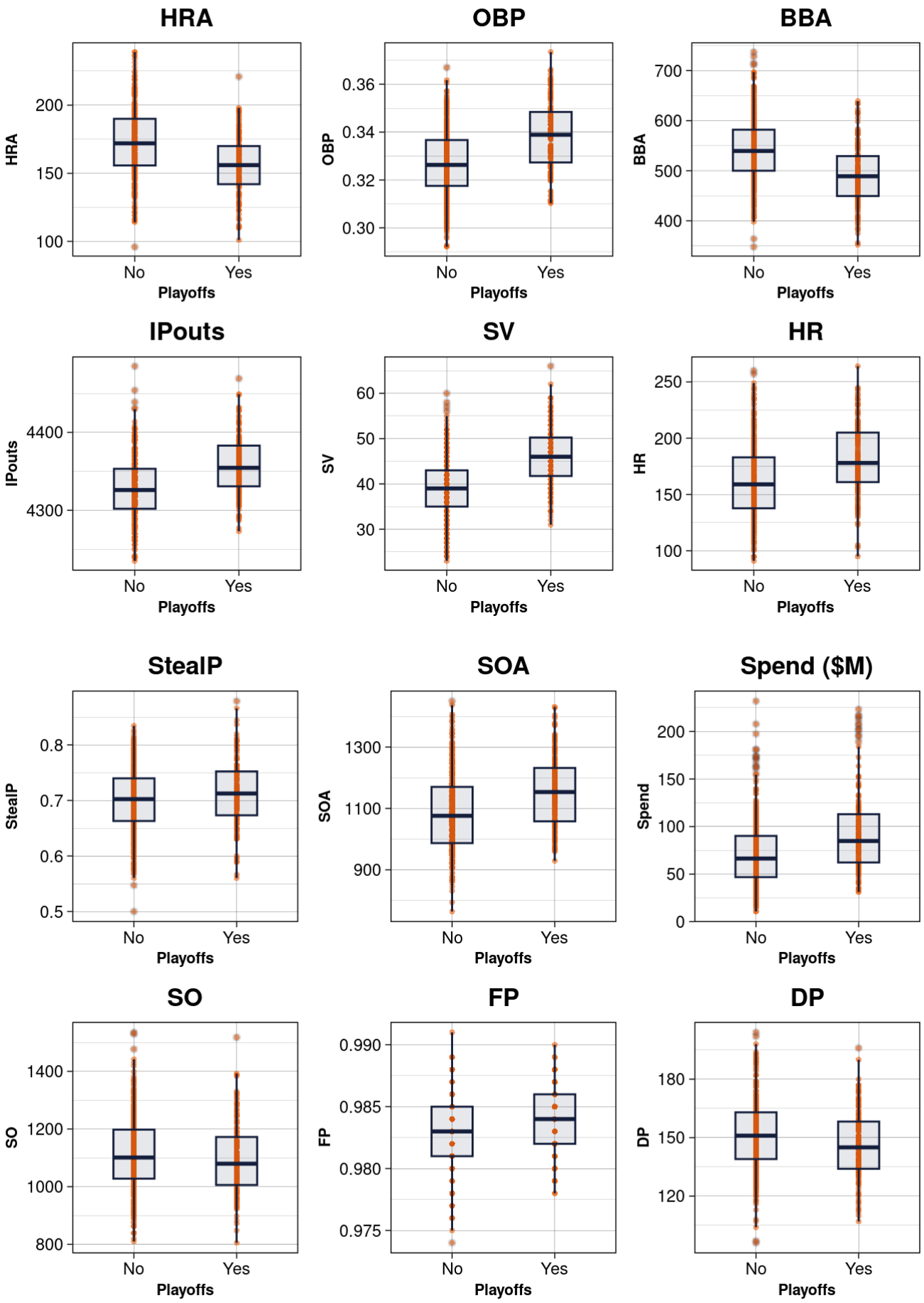
Next, we use the categorical variables to better visualize the difference in the percentage of teams making the playoffs by comparing teams with higher-than-average and lower-than-average values per variable.



The bar charts in Figure 2.c show higher shares of teams that make it to the playoffs have lower-than-average `HRA` and `BBA` but higher-than-average `OBP` and `SV`.

Again, while not as prominent, we see that higher shares of teams that make it to the playoffs have higher-than-average `HR`, `IPouts`, `SOA`, and `Spend`.

Figure 2.c: Predictors by Playoff Status



The boxplots in Figure 2.c visualize the difference between teams who do and do not make the playoffs across the quantitative variables.

Here, we still see noticeable differences between teams that do and do not make the playoffs across these variables.

iii. Contextual interpretations of data visualizations

Based on the visualizations, **`HRA`**, **`BBA`**, **`IPouts`**, **`SV`**, and **`OBP`** are the most powerful predictors of whether a team will make the playoffs. The boxplots show the difference in values between teams that make the playoffs and those that do not; there are differences between playoff and non-playoff teams for all predictors. Based on the above visualizations, teams that make it to the playoffs tend to have higher **`HR`**, **`OBP`**, and **`FP`** and fewer **`BBA`** and **`HRA`**.

`OBP`, **`HR`**, **`HRA`**, and **`BBA`** appear to be the biggest factors in whether a team plays in the postseason because their distributions are the most different. Teams that make the playoffs tend to have a slightly higher **`FP`** and, interestingly, tend to have slightly less **`DP`** than their non-playoff counterparts. In contrast, distributions for offensive statistics seem to differ more drastically for playoff and non-playoff teams, suggesting that a strong offense may be more predictive of playoff status than a strong defense.

Next, we create visualizations to see whether there are differences in **`HR`** and **`OBP`** by league and division.

Figure 2.d: OBP by Playoffs Status

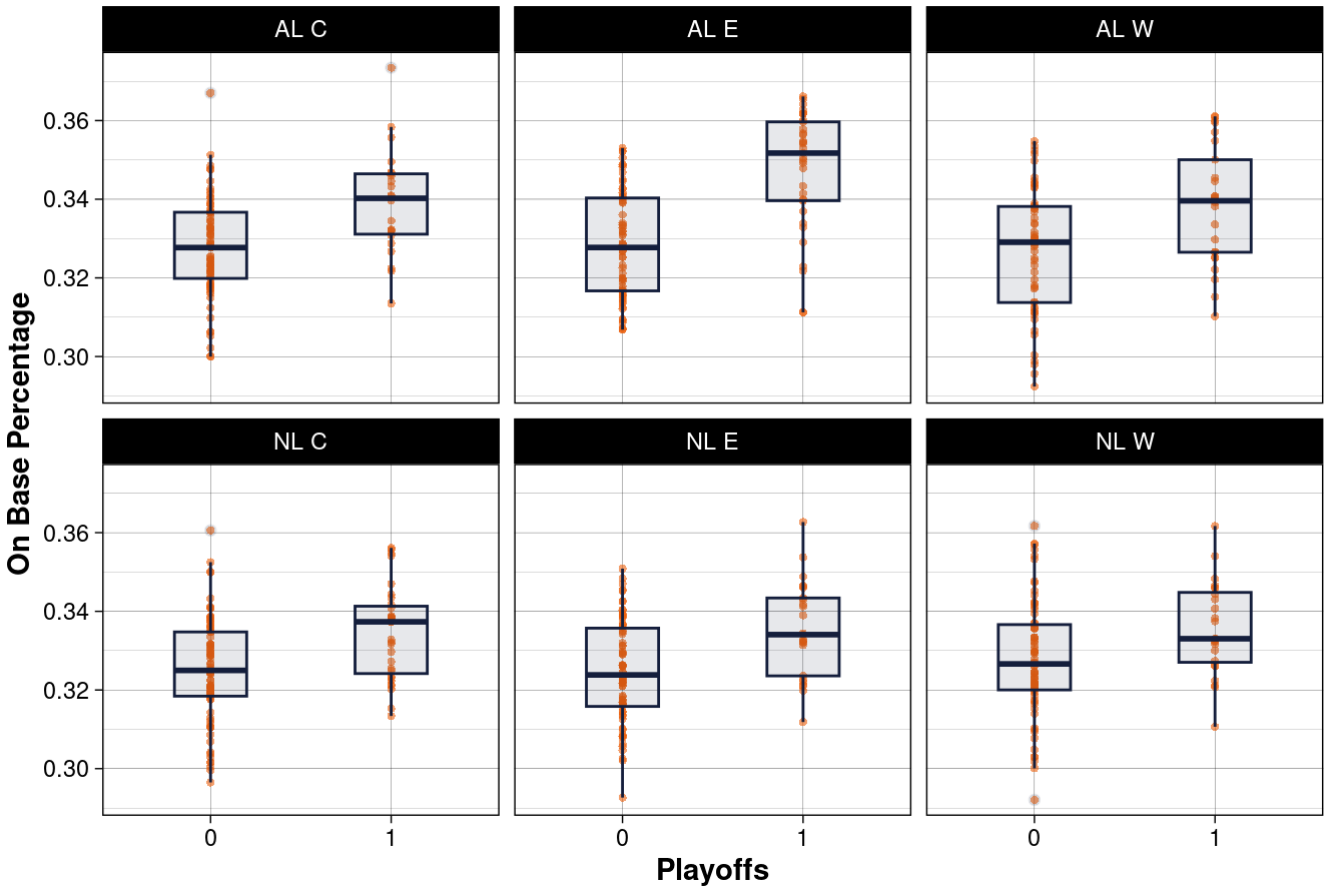
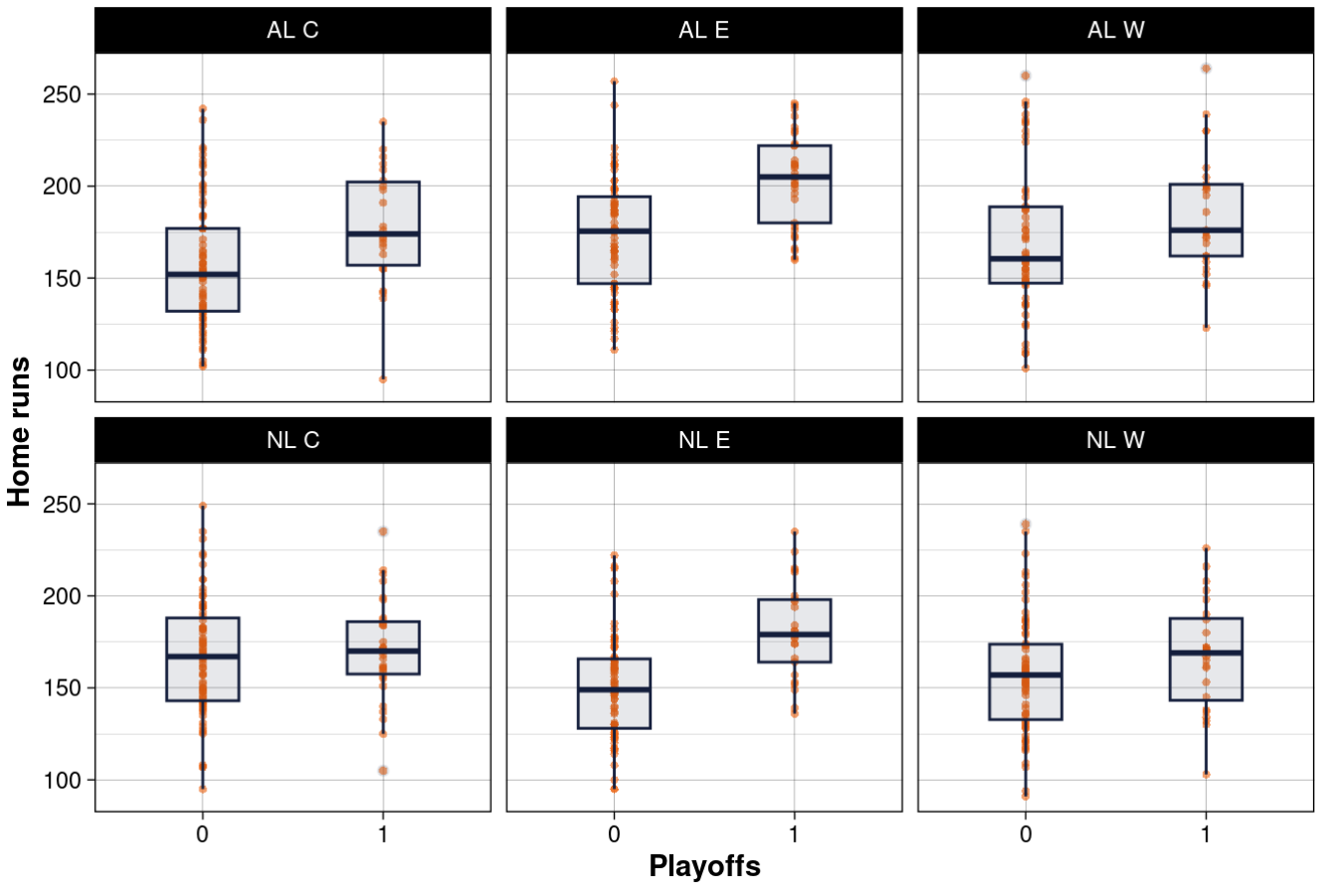


Figure 2.e: Home runs by Playoffs Status



Based on Figure 2.d and Figure 2.e, we can see some differences in `OBP` and `HR` across divisions and playoff status. In particular, AL East appears to have the largest difference in distribution between non-playoff teams and playoff teams for both `OBP` and `HR`, suggesting that there may be some imbalance for teams in this league. Moreover, AL East playoff teams tend to have higher `OBP` and more `HR` by a large margin compared to other divisions, and AL East non-playoff teams tend to have more `HR` than other divisions. This suggests that AL East teams tend to outperform other divisions, regardless of playoff status. When we compare the distributions for the other five divisions considering `OBP` and `HR`, there tends to be less variation across divisions for both playoff and non-playoff teams. This suggests that division ID may not be a significant predictor in our model.

c. Model Building:

i. How we chose our initial logistic regression model

We chose to include predictors that a team owner or general manager has the power to influence, including various offensive and defensive statistics and team spend. We also included `lgIDdivID` to examine if a team's division has a statistically significant influence on its likelihood of making the playoffs in the presence of the other predictors. Given the nature of our question, we also excluded some predictors from our models. In this context, several predictors obviously influence playoff statuses, such as wins (`W`) and shutouts (`SHO`), that we have excluded. The relationship between wins and playoff status per the MLB regulations is well understood; we are more interested in the less obvious relationship between playoff status and other predictors.

The predictors we ultimately decided to examine as possible model inputs include `HR`, `HRA`, `IPouts`, `OBP`, `Spend`, `BBA`, `DP`, `SV`, `SOA`, `StealP`, `FP`, and `lgIDdivID`.

A team owner may be able to increase a team's home runs (`HR`) by strategically placing players with the highest likelihood of hitting home runs in certain spots in the batting order or by aligning the batting order in a specific sequence against certain pitchers. Similarly, an owner may reduce the number of home runs allowed (`HRA`) by employing certain pitching strategies against — perhaps purposefully walking — opponent batters with high likelihoods of hitting home runs. And, as is the idea behind "Moneyball," an owner may influence a team's chances of winning by maximizing its On-base percentage (`OBP`); for example, stacking a batting lineup with players that, despite not having great batting averages, are more likely to get on base, whether that be achieved by actually hitting the ball (`H`), being hit by a pitch (`HBP`), or being walked (`BB`). The box plots from our EDA show meaningful differences between teams that did and did not make it to the `playoffs` in terms of `OBP`, `HR`, and `HRA`.

We also know contextually that team owners and general managers influence the amount of money a team spends in a given season (`Spend`), and in our EDA, we see a difference in median `Spend` by `playoff` status.

Coaches may reduce the number of times an opponent catches their team stealing (`CS`) or increase the number of times a team successfully steals a base (`SB`) by calculating each player's percent chance of successfully stealing a base and determining a threshold below which players are simply barred from stealing. We include `StealP` as a team's percent chance of successfully stealing.

The previously mentioned predictors are largely offensive. Some defensive predictors we chose to examine include `BBA`, `DP`, `FP`, `HRA`, `SV`, `SOA`, `IPouts`. A team owner may increase or decrease a team's `BBA` based on opponent batters' capabilities; for example, as previously mentioned, a pitcher might purposefully walk a batter with greater chances of hitting a home run. A pitcher might also purposefully not walk slower batters, assuming that doing so increases the chances of getting that batter out on his way to first.

Initial model:

```
##
## Call:
## glm(formula = Playoffs ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.59459  -0.35318  -0.09375   0.23706   2.99145
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.552e+02  6.124e+01  -2.535 0.011251 *
## HR             2.049e-02  5.823e-03   3.518 0.000435 ***
## SO             9.733e-04  1.504e-03   0.647 0.517430
## SV             1.631e-01  2.762e-02   5.904 3.56e-09 ***
## IPouts         4.314e-03  4.397e-03   0.981 0.326530
## HRA           -4.402e-02  8.347e-03  -5.274 1.33e-07 ***
## BBA           -1.666e-02  3.060e-03  -5.444 5.21e-08 ***
## SOA            4.791e-03  1.822e-03   2.629 0.008559 **
## DP             4.845e-03  1.030e-02   0.471 0.637961
## FP             8.762e+01  6.189e+01   1.416 0.156891
## OBP            1.338e+02  1.677e+01   7.974 1.53e-15 ***
## StealP         3.839e+00  2.583e+00   1.486 0.137278
## Spend          2.087e-03  4.738e-03   0.441 0.659548
## lgIDdivIDAL E -9.928e-01  5.592e-01  -1.775 0.075855 .
## lgIDdivIDAL W  1.669e-01  5.209e-01   0.320 0.748645
## lgIDdivIDNL C  4.085e-01  5.280e-01   0.774 0.439171
## lgIDdivIDNL E -3.688e-01  5.304e-01  -0.695 0.486827
## lgIDdivIDNL W -1.548e-01  5.180e-01  -0.299 0.765121
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 675.40  on 567  degrees of freedom
## Residual deviance: 308.22  on 550  degrees of freedom
## AIC: 344.22
##
## Number of Fisher Scoring iterations: 7
```

ii. How we tried to improve our initial logistic regression model

The *full* model shows that `HR`, `HRA`, `SV`, `BBA`, `SOA`, and `OBP` are the only statistically significant predictors based on their respective p-values.

Before removing any predictors, we use the *glmulti* package to perform all-subset logistic regressions to find the best model based on two separate penalized-fit criteria; the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). AIC uses a penalty term proportional to the number of parameters in the model, while BIC uses a penalty term proportional to the logarithm of the sample size multiplied by the number of parameters. AIC makes sense as a criterion in the context of our dataset and question of interest because the sample size of the training data (568 observations) is large compared to the number of parameters under consideration (12). BIC makes sense as a criterion because it balances between goodness-of-fit and model complexity; further, given that we split our training and testing data in a 95-5 split, the sample size of our test data is much smaller than the training data, so only using AIC as the criteria could result in overfitting when we test the model.

The different criteria yield different models, the formulas and summaries of which are shown below:

AIC as criterion:

```
## [[1]]
## Playoffs ~ 1 + HR + SV + IPouts + HRA + BBA + SOA + OBP + StealP
## <environment: 0x55efec943318>
```

```
##
## Call:
## fitfunc(formula = as.formula(x), family = ..1, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8982  -0.3626  -0.1038   0.2307   2.7344
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -71.741421  18.977654  -3.780 0.000157 ***
## HR            0.019888   0.005181   3.839 0.000124 ***
## SV            0.150591   0.025646   5.872 4.30e-09 ***
## IPouts        0.006217   0.004207   1.478 0.139469
## HRA          -0.043200   0.007818  -5.526 3.28e-08 ***
## BBA          -0.017080   0.002796  -6.109 1.00e-09 ***
## SOA           0.004444   0.001407   3.158 0.001590 **
## OBP         125.993318  14.962192   8.421  < 2e-16 ***
## StealP        3.778957   2.387534   1.583 0.113470
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 675.40  on 567  degrees of freedom
## Residual deviance: 318.45  on 559  degrees of freedom
## AIC: 336.45
##
## Number of Fisher Scoring iterations: 7
```

BIC as criterion:

```
## [[1]]
## Playoffs ~ 1 + HR + SV + HRA + BBA + SOA + OBP
## <environment: 0x55efee315c68>
```

```
##
## Call:
## fitfunc(formula = as.formula(x), family = ..1, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9075  -0.3883  -0.1110   0.2455   2.7517
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -42.568375   5.612132  -7.585 3.32e-14 ***
## HR            0.020194   0.005146   3.924 8.71e-05 ***
## SV            0.156404   0.024668   6.340 2.29e-10 ***
## HRA          -0.043264   0.007721  -5.604 2.10e-08 ***
## BBA          -0.017359   0.002772  -6.261 3.82e-10 ***
## SOA           0.005079   0.001361   3.731 0.000191 ***
## OBP         125.023473  14.755563   8.473  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 675.40  on 567  degrees of freedom
## Residual deviance: 323.18  on 561  degrees of freedom
## AIC: 337.18
##
## Number of Fisher Scoring iterations: 7
```

The AIC of the model produced using BIC as the criterion is slightly higher (337.18) than that of the model produced using AIC as the criterion (336.45).

We fit two reduced models: *reduced1*, where we use the model recommended when using AIC as the criterion, and *reduced2*, where we use the model recommended when using BIC as the criterion.

reduced1:

```
##
## Call:
## glm(formula = Playoffs ~ HR + SV + HRA + BBA + SOA + OBP + StealP,
##     family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9529  -0.3756  -0.1103   0.2363   2.7677
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -45.434997   5.990859  -7.584 3.35e-14 ***
## HR            0.020168   0.005153   3.914 9.10e-05 ***
## SV            0.160373   0.024977   6.421 1.36e-10 ***
## HRA          -0.044021   0.007765  -5.669 1.44e-08 ***
## BBA          -0.017091   0.002785  -6.137 8.41e-10 ***
## SOA           0.004880   0.001372   3.558 0.000374 ***
## OBP         125.754969  14.881003   8.451  < 2e-16 ***
## StealP        3.748044   2.378615   1.576 0.115089
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 675.40  on 567  degrees of freedom
## Residual deviance: 320.65  on 560  degrees of freedom
## AIC: 336.65
##
## Number of Fisher Scoring iterations: 7
```

reduced2:

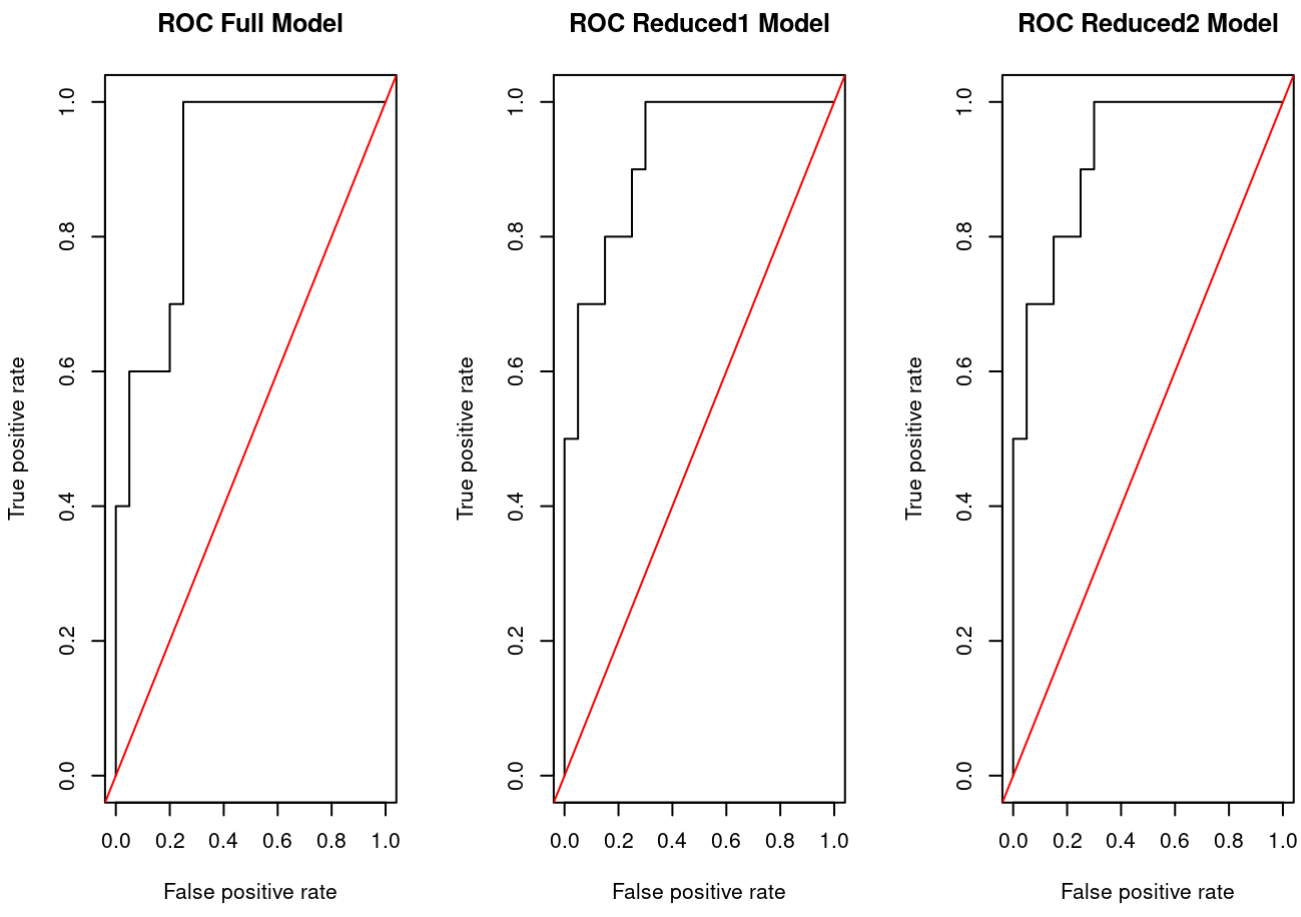
```
##
## Call:
## glm(formula = Playoffs ~ HR + SV + HRA + BBA + SOA + OBP, family = "binomial",
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9075  -0.3883  -0.1110   0.2455   2.7517
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -42.568375   5.612132  -7.585 3.32e-14 ***
## HR            0.020194   0.005146   3.924 8.71e-05 ***
## SV            0.156404   0.024668   6.340 2.29e-10 ***
## HRA          -0.043264   0.007721  -5.604 2.10e-08 ***
## BBA          -0.017359   0.002772  -6.261 3.82e-10 ***
## SOA           0.005079   0.001361   3.731 0.000191 ***
## OBP         125.023473  14.755563   8.473  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 675.40  on 567  degrees of freedom
## Residual deviance: 323.18  on 561  degrees of freedom
## AIC: 337.18
##
## Number of Fisher Scoring iterations: 7
```

We will compare three models: the initial, full model, *result*, and the reduced models, *reduced1* and *reduced2*.

iii. Comparing model performance by ROC curve, AUC, and Accuracy rate:

ROC curve

Based on the ROC curves for the full and reduced models, we see that these models perform better at predicting playoff status than random guessing.



The Area Under the Curve (AUC)

full Model: (0.895)

reduced1: (0.925)

reduced2: (0.920)

The *full* model has a slightly lower AUC than both reduced models; however, the reduced models are much simpler. The *reduced2* model has slightly less AUC than *reduced1*. Next, we will check model accuracy.

Accuracy rate

full: The *full* model is 73.33% accurate at a threshold of 0.05.

reduced1: The *reduced1* model is 86.67% accurate at a threshold of 0.05.

reduced2: The *reduced2* model is 86.67% accurate at a threshold of 0.05.

Other ways of comparison, if appropriate

Given that our *reduced2* model is a subset of our *reduced1* model, we will also compare using a Likelihood Ratio Test (LRT).

```
## Analysis of Deviance Table
##
## Model 1: Playoffs ~ HR + SV + HRA + BBA + SOA + OBP + StealP
## Model 2: Playoffs ~ HR + SV + HRA + BBA + SOA + OBP
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       560     320.65
## 2       561     323.18 -1  -2.5252    0.112
```

The LRT compares the fit of the two models by testing the null hypothesis (Ho) that the *reduced1* model and the *reduced2* model fit the data equally well versus the alternative hypothesis (Ha) that the *reduced1* model fits the data significantly better than the *reduced2* model. We use the `anova` function to perform the LRT and the Chisq argument to specify the test type.

The p-value is greater than 0.05, so we fail to reject the null hypothesis. From the LRT, we conclude that we can go with the *reduced2* model because adding **`StealP`** and **`IPouts`** does not significantly improve the model fit.

The model summary for *reduced1* showed that **`StealP`** and **`IPouts`** were not statistically significant, which the LRT confirms. We will validate this with a hypothesis test, that is, adding **`StealP`** and **`IPouts`** as predictors to the *reduced2* model:

- *Ho:* β7= β8 = 0.
- *Ha:* at least one of the coefficients in Ho is not zero.

We calculate a ΔG2 test statistic and compare it with a χ2 distribution with 2 degrees of freedom. The ΔG2 test statistic is the difference in the residual deviance of the *reduced1* model and that of the *reduced2* model.

The test statistic is (-4.7309), and the corresponding p-value is (0.0939).

The results of this test confirm the result of the LRT, so we conclude that **`StealP`** and **`IPouts`** do not significantly improve the model fit; we can go with the *reduced2* model.

iv. Recommended logistic regression model(s)

We recommend the *reduced2* model given:

- The *reduced2* model is much simpler than both the *full* and *reduced1* models.
- The *reduced2* model has greater AUC than the *full* and similar AUC to the *reduced1* models.
- The *reduced2* model has the same accuracy as the *reduced1* and better accuracy than the *full* model.

In short, *reduced2* is highly accurate in its predictive capabilities and a relatively simple model.

The final recommended logistic regression equation is:

$$\log(\frac{\pi}{1-\pi}) = -42.568375 + 0.020194(\text{HR}) + 0.156404(\text{SV}) - 0.043264(\text{HRA}) \\ -0.017359(\text{BBA}) + 0.005079(\text{SOA}) + 125.023473(\text{OBP}) + \epsilon$$

d. Conclusions:

i. How our model(s) answers our question of interest

Our recommended model for predicting 2016 playoff status includes the following predictors: **`HR`**, **`SV`**, **`HRA`**, **`BBA`**, **`SOA`**, and **`OBP`**.

Of these variables, **`OBP`** is by far the most powerful predictor variable when reviewing the coefficients of our recommended model. For every additional percentage point increase in OBP, the estimated odds of that team progressing to the playoffs get multiplied by an estimated 3.49, assuming all other variables remain constant. Based on our data, this implies that general managers and coaches should have prioritized improving team OBP for the best playoff chances in 2016. Playoff status is consistent with the OBP theory in popular culture's Moneyball.

ii. Provide interesting insights gained about the data

Similar to our recommended model for the linear regression in Question 1, our logistic regression model also drops **`Spend`** as a predictor, despite our initial visualizations showing a generally higher team spend for playoff teams compared to non-playoff teams. This remains a hotly debated topic in MLB today, with proponents for team salary caps and team salary floors to create some consistency in payroll across MLB teams to keep the league fair. However, at least for predicting 2016 playoff status, **`Spend`** is not statistically significant in the presence of our other predictors (in particular, **`OBP`**).

Our recommended logistic regression model keeps much of the same predictors that we see have a large quantitative difference across our initial between playoff and non-playoff status teams across our initial set of predictors based on our preliminary visualizations.

iii. Challenges we faced

Like the work done for the linear regression, the group spent a significant amount of time upfront researching and understanding these variables to understand their context in the game of baseball holistically and their context in potentially being used as predictors of a team's playoff appearance.

In addition, it was challenging to narrow our scope to predictor variables that an owner or team manager would have some influence over, but the group was determined to create a model that would be useful in decision-making scenarios around team strategy to increase the likelihood of a team's playoff appearance. Ultimately, our model predicts the likelihood of playoff appearance using six quantitative predictors based on our understanding of these predictors and our use of various statistical methods.

Sources:

- Lewis, M. (2003). Moneyball: *The art of winning an unfair game*. New York, NY: Norton.
- James, M., and Wells, A. (2008). Evaluating baseball performance: A study of the best measures for evaluating team performance. *Journal of Sports Economics*, 9(6), pp. 598-626.
- Pinheiro, R. and Szymanski, S. (2022). All runs are created equal: Labor market efficiency in Major League Baseball. *Journal of Sports Economics*, 23(8), pp. 1046-1075.