David Stern
DATA 621: Business Analytics and Data Mining
MSc Data Analytics, CUNY SPS

Data Exploration

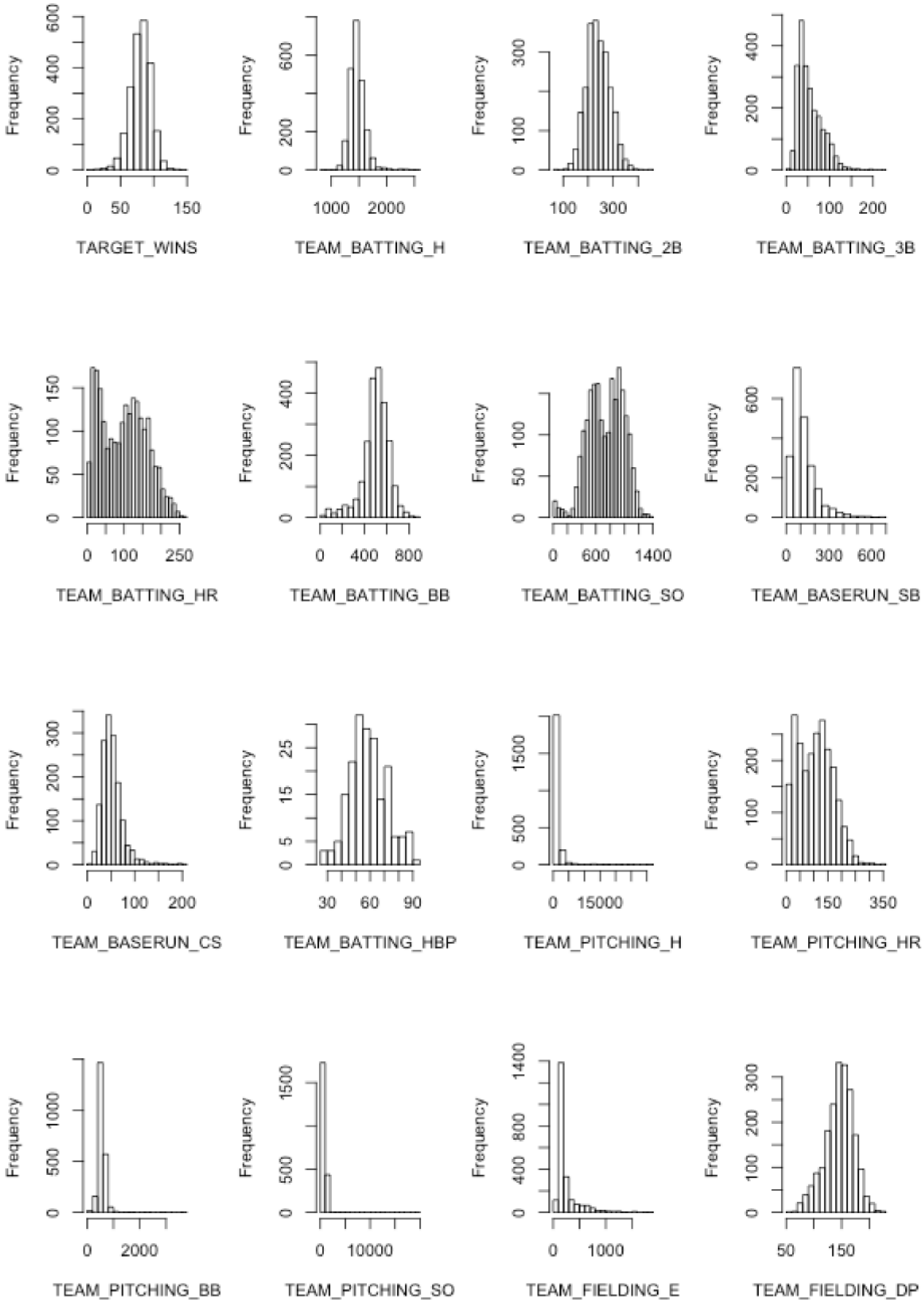
Our dataset consists of 16 variables detailing the performance of professional baseball teams. Each record represents a team's performance for a given season from 1871-2006. The variables include offensive and defensive metrics such as base hits by batters, fielding errors, and strikeouts by pitchers that we will use to predict the team's wins for the season. The data is split into training and evaluation set with the total wins for the season removed from the latter. There are 2271 records in the training set and 259 records in the evaluation set.

Here a quick overview of the variables in the dataset:

Variable	***Abbrev.***	***Description***
TARGET_WINS	W	Number of wins
TEAM_BATTING_H	B_H	Base Hits by batters (1B,2B,3B,HR)
TEAM_BATTING_2B	B_2B	Doubles by batters (2B)
TEAM_BATTING_3B	B_3B	Triples by batters (3B)
TEAM_BATTING_HR	B_HR	Homeruns by batters (4B)
TEAM_BATTING_BB	B_BB	Walks by batters
TEAM_BATTING_HBP	B_HBP	Batters hit by pitch
TEAM_BATTING_SO	B_SO	Strikeouts by batters
TEAM_BASERUN_SB	B_SB	Stolen bases
TEAM_BASERUN_CS	B_CS	Caught stealing
TEAM_FIELDING_E	F_E	Errors
TEAM_FIELDING_DP	F_DP	Double Plays
TEAM_PITCHING_BB	P_BB	Walks allowed
TEAM_PITCHING_H	P_H	Hits Allowed
TEAM_PITCHING_HR	P_HR	Homeruns Allowed
TEAM_PITCHING_SO	P_SO	Strikeouts by pitcher

We will first look at the distribution of the variables by examining the histograms of each one.

Histograms of Variables in Training Set



We see above that the variables do vary somewhat in their distributions. The very first plot shows that our response variable, wins, seems normally distributed around it's mean – 80.8 – and is not skewed. It is interesting to note that the wins distribution proves Tommy LaSorda's axiom incorrect. Not all teams win 60 games and lose 60 games – about 15% of seasons actually fall outside the range of 60-102 wins.

Some of the variables appear normally distributed with a small amount of skew: triples by batter, walks by batter, and caught stealing. A few others - strikeouts by batter, homeruns allowed, homeruns by batter – appear bimodal. We should also note that a few of the variables seem to be distorted by unusually highly values so we will take a closer look at the distributions for fielding errors, and hits, walks, and strikeouts allowed.

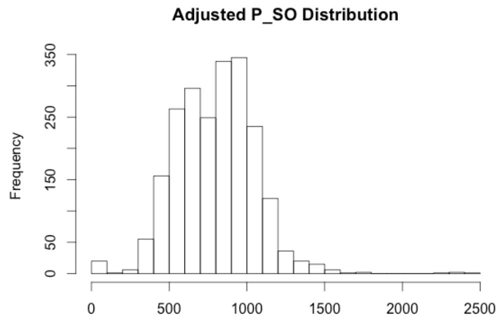
Summary Statistics Table

	W	*B_H*	*B_2B*	*B_3B*	*B_HR*	*B_BB*	*B_HBP*	*B_SO*	*B_SB*	*B_CS*	*F_E*	*F_DP*	*P_BB*	*P_H*	*P_HR*	*P_SO*
Min.	0	891	69	0	0	0	29	0	0	0	65	52	0	1137	0	0
1st Quartile	71	1383	208	34	42	451	50.5	548	66	38	127	131	476	1419	50	615
Median	82	1454	238	47	102	512	58	750	101	49	159	149	537	1518	107	813.5
Mean	80	1469	241	55.3	99.6	502	59.36	736	125	52.8	247	146	553	1779	106	817.7
3rd Quartile	92	1537	273	72	147	580	67	930	156	62	249	164	611	1682	150	968
Max	146	2554	458	223	264	878	95	1399	697	201	1898	228	3645	30132	343	19278
NA Count	0	0	0	0	0	0	2085	102	131	772	0	286	0	0	0	102

This table demonstrates that the four variables under suspicion are in fact distorted by improbably high values. For fielding errors, it seems improbable that a team could average over 10 errors per game over the course of a season, but the distribution shows that this the maximum is not an outlier, but the right tail of exponential decay. The shape of these distributions will be important later on when deciding how to impute data for variables with missing values.

Data Preparation

While examining the outliers for strikeouts, I identified five records – rows 1, 282, 1342, 1826, 2136 - with more than 2500 strikeouts by pitcher. Some of the strikeout counts were actually impossible to achieve even if a team struck out every batter for every out over the course of a season. I decided to delete these records from our training data. They also contained improbably high values for walks allowed, hits allowed, and fielding errors allowed, so these were very likely data entry errors, or incorrectly scaled for those seasons with fewer than 162 games. Since we will want to choose the best method for imputing the missing values for each variable, I plotted strikeouts by pitcher again without the extreme outliers. The distribution seems much more normally distributed around the mean of 799.



Next I looked into the six variables with missing values. These variables and the proportion of the data that is missing for each is:

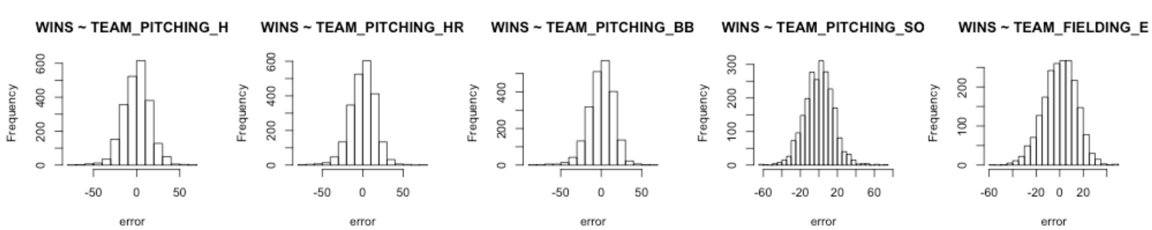
Predictor	***Pct. NA***
TEAM_BATTING_SO	4.5%
TEAM_BATTING_SB	5.8%
TEAM_BASERUN_CS	33.9%
TEAM_BATTING_HBP	91.6%
TEAM_PITCHING_S0	4.5%
TEAM_FIELDING_DP	12.6%

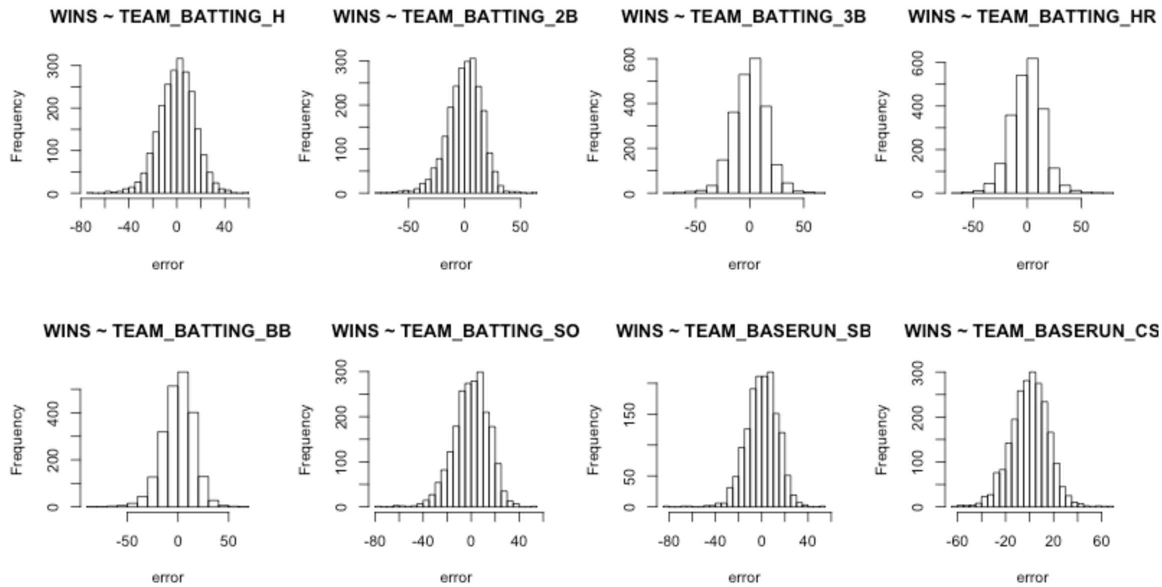
Here we see that some variables are missing much more data than others. We will explore the effectiveness of imputing missing values with some of the variables with a smaller proportion of data missing: strikeouts by batters, stolen bases, pitching strikeouts, and perhaps also fielding double plays. The percentage of those missing values for caught stealing and batters hit by pitch, seem very high and we may be better off deleting these variables altogether. If we perform exploratory data analysis in R without removing missing values, our linear regression function *lm* and correlation matrix *cor(data, use=na.or.complete)* will not include any records that have one or more missing values. This means that at least 91.6% of our training data will be discarded. To determine how best to proceed, I eliminated the batters hit by pitch variable from the dataset and created a correlation matrix of the remaining variables. Although records with missing values were excluded from the correlation data, we know from the percentages from the table above that the correlations will still be calculated from at least 40% of our training data. We should however be careful in drawing conclusions from the correlation matrix. We will only use it to identify possible collinearities.

	W	B_H	B_2B	B_3B	B_HR	B_BB	B_SO	B_SB	B_CS	P_H	P_HR	P_BB	P_SO	F_E	F_DP
W	1	0.36	0.19	0.08	0.28	0.35	-0.06	0.12	-0.01	0.22	0.28	0.29	-0.07	-0.25	-0.05
B_H		1	0.68	0.42	0.12	0.17	-0.37	0.01	0.01	0.69	0.14	0.19	-0.33	0.12	0.15
B_2B			1	0.04	0.31	0.20	0.08	0.07	-0.12	0.45	0.32	0.19	0.09	-0.19	0.05
B_3B				1	-0.55	-0.10	-0.69	0.05	0.37	0.36	-0.53	-0.02	-0.64	0.64	0.01
B_HR					1	0.29	0.64	-0.14	-0.47	0.00	0.97	0.20	0.60	-0.62	0.07
B_BB						1	0.02	-0.08	-0.21	0.13	0.30	0.87	0.03	-0.16	0.16
B_SO							1	0.13	-0.26	-0.34	0.61	-0.06	0.93	-0.63	-0.14
B_SB								1	0.65	0.02	-0.13	-0.06	0.13	-0.01	-0.26
B_CS									1	0.07	-0.45	-0.14	-0.23	0.43	-0.21
P_H										1	0.17	0.49	-0.06	0.14	0.12
P_HR											1	0.32	0.65	-0.60	0.08
P_BB												1	0.13	-0.10	0.16
P_SO													1	-0.59	-0.12
F_E														1	-0.08
F_DP															1

The correlation matrix shows us that there are a number of moderate correlations between the variables, $0.3 \le |x| \le 0.5$ (in light red for negative, light blue for positive) and strong correlations $|x| \ge 0.5$ (dark red for negative, dark blue for positive). This gives us an early hint that there we will very likely see the effect of multicollinearity in a full multiple linear regression fit, but that we might also be able to predict the missing values for some predictor variables by performing linear regression on some subset of the other predictors.

Before I started fitting models, I fit each of the predictor variables against our predictor variable, target wins, to make sure that the errors were normally distributed. They are each normally distributed around a mean of 0. I also examined scatterplots of the squared residuals for each of the simple regression models and did not identify any pattern in the size of the residuals for any predictor. Each appears to be homoscedastic.





Building Models

Before dealing with the missing values in the training data, I wanted to build a model that used only the records with data for each variable.

First, I fit each of the predictor variables individually to the response variable, wins, and found that the highest r-squared value was 0.15 for hits by batters (B_H). The p-value for the model was very significant with a value of 2.2e-16.

Model 1

Next, I built the first multiple linear regression model by fitting all of the variables and working backwards, updating the model by subtracting one variable in order of descending p-value. After subtracting, in order, walks allowed, homeruns allowed, and strikeouts by batter, the p-values for model and for the individual predictors all dropped below 0.05. The goodness of fit measures and coefficients are:

Model p-value	***f-stat***	***R-squared***	***RSE***
<2.2e-16	104.6 on 11 and 1474 df	0.4384	9.548

Intercept	***B_H***	***B_2B***	***B_3B***	***B_HR***	***B_BB***	***B_SB***	***B_CS***	***P_H***	***P_SO***	***F_E***	***F_DP***
58.4	0.03	-0.07	0.16	0.10	0.04	0.04	0.05	0.01	-0.02	-0.16	-0.11

The R-squared value shows us that the model explains just about 44% of the variation in the model. Some of the coefficients are counterintuitive. We would assume that strikeouts by pitcher (P_SO), fielder double plays (F_DP), and doubles by batter (B_2B) to have a positive effect on wins, but the coefficients are negative. We would also expect batter caught stealing (B_CS), hits allowed (P_H) to be negative, but the coefficients are positive. We can certainly improve on this model.

Model 2

We can improve this model by imputing the missing values. Since the linear regression function does not include records with NA values, 785 records (or 34.5%) of our training data set is essentially discarded. Based on the shape of the distributions, I imputed the missing values for strikeouts by batter (B_S) and stolen bases (B_SO) with the median values for each variable. I imputed the missing values for the more normally distributed variables – caught stealing (B_CS), strikeouts by pitcher (P_SO) and double plays fielded (F_DP) – with the mean for each variable. I then fit the full model and again worked backwards. After subtracting, in order, homeruns allowed, hits allowed, and caught stealing, the p-values for model and for the individual predictors all dropped below 0.05. The goodness of fit measures and coefficients are:

Model p-value	***f-stat***	***R-squared***	***RSE***
<2.2e-16	97.04 on 11 and 2259 df	0.3209	12.96

Intercept	***B_H***	***B_2B***	***B_3B***	***B_HR***	***B_BB***	***B_SO***	***B_SB***	***P_BB***	***P_SO***	***F_E***	***F_DP***
22.64	0.04	-0.03	0.07	0.07	0.03	-0.02	0.03	-0.01	0.01	-0.02	-0.12

With a much lower r-squared value, this model does not seem as good a fit as the previous one. Aside from the intercept, the magnitude of the coefficients did not change drastically. The only variable in both of the models that changes sign is strikeouts by pitcher (P_SO). This is a good sign, although it has a relatively small effect in the model and the difference in coefficients is only 0.03.

Model 3

For my third model, I hoped to see if I could improve the model by also imputing the zero values for each of the variables. Given that the non-occurrence of any of the events described by the variables during a baseball season is extremely improbable, we might want to treat these as missing values: zeroes entered in data where none was available. I imputed the zero values for each of the variables with the mean value and worked backwards from the full model. After the p-values for each of the predictors dropped below 0.05, the measures and coefficients are:

Model p-value	***f-stat***	***R-squared***	***RSE***
<2.2e-16	87.59 on 12 and 2258 df	0.3176	13

Intercept	***B_H***	***B_2B***	***B_3B***	***P_HR***	***B_BB***	***B_SO***	***B_SB***	***P_BB***	***P_SO***	***F_E***	***F_DP***	***P_H***
9.07	0.06	-0.03	0.07	0.05	0.03	-0.01	0.03	-0.01	0.01	-0.02	-0.11	0

Based on the R-squared value, this model seems to be a slightly worse fit than the previous. The coefficients are virtually the same – for those that did change, the difference is negligible.

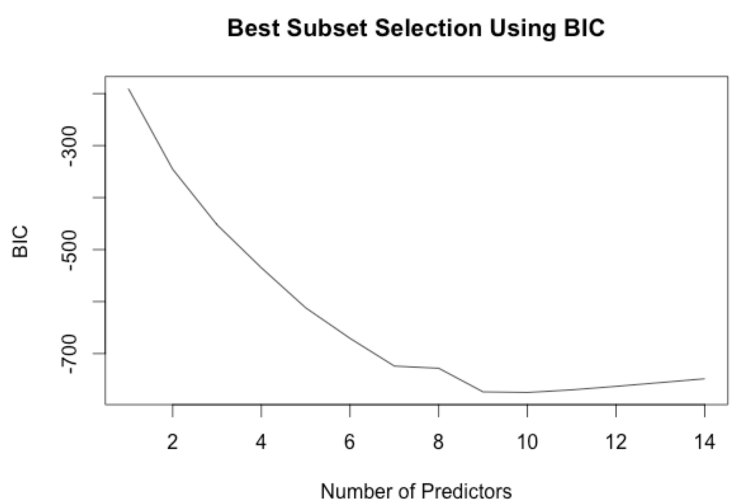
Select Models

Of the three multiple linear regression models, I think our first model provided the best fit to our data. Although the model was trained on a much smaller portion of the data, it provides the highest R-squared value and lowest residual standard error (RSE) than the two models with the missing values and zeroes imputed with mean and median values. Since we are fitting many variables, we should use the RSE as the goodness of fit measure, since the R-squared value will increase for each predictor that we add. This was demonstrated in the backwards selection process as the R-squared value dropped - sometimes by very small numbers - for each predictor that was removed from the model. The variance inflation factors (VIF) for the predictors in this model also returned relatively low values compared to the other models. Each of the VIFs is below 10 so we can conclude that there is less effect of multicollinearity in this model.

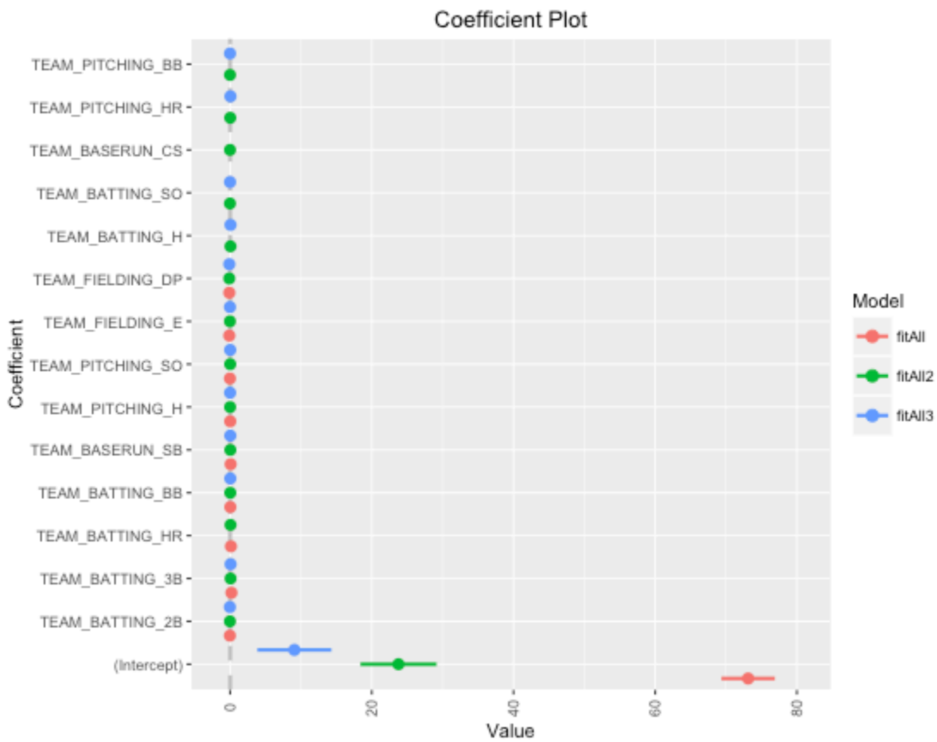
Model 1 VIF

B_H	***B_2B***	***B_3B***	***B_HR***	***B_BB***	***B_SB***	***B_CS***	***P_H***	***P_SO***	***F_E***	***F_DP***
6.04	2.48	2.78	3.43	1.12	2.41	2.82	2.71	3.91	2.43	1.16

Finally, I used Bayesian information criterion (BIC) to determine the best number of predictors to use in the subset. In the plot below we see that the BIC is minimized between 9 and 11 predictors, so our model is within the proper range.

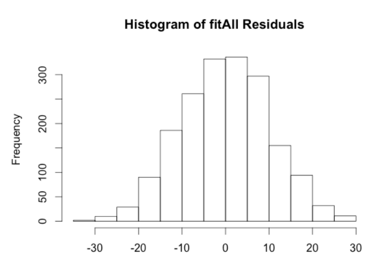


Although the signs for some of the coefficients for our first model seem counterintuitive, if we compare them between models, they coefficients do not vary much in magnitude. The major difference between the models is the magnitude of the intercept coefficient. It is highest for the first model, where it is about the same as the first quartile for the distribution of wins.



Summary

After comparing the three models, I would use the first. Although there is room for improvement in goodness-of-fit measures, the p-value (<2.2e-16) and f-statistic (104.6 on 11 variables and 1474 degrees of freedom) indicate that the model is statistically significant. The residuals for the fit are normally distributed around a mean of zero. I used the first model to predict the number of wins for the records in the evaluation set. Our predictions, with wins rounded to the nearest whole number, along with their prediction and confidence intervals are included in the appendix.



Appendix

index	*fit*	*c-l*	*c-u*	*p-l*	*p-u*	*in.*	*fit*	*c-l*	*c-u*	*p-l*	*p-u*	*in.*	*fit*	*c-l*	*c-u*	*p-l*	*p-u*
1	61	59	62	41	81	***43***	74	73	75	54	94	***85***	72	71	74	52	92
2	67	65	68	47	87	***44***	75	74	77	55	96	***86***	89	88	90	69	109
3	71	70	72	51	91	***45***	69	68	70	49	89	***87***	91	90	92	71	111
4	84	83	85	64	104	***46***	78	77	79	58	98	***88***	91	89	93	71	111
5	64	62	66	44	84	***47***	92	89	95	72	112	***89***	95	93	96	75	115
6	73	72	75	53	94	***48***	77	75	79	57	97	***90***	91	89	92	70	111
7	70	69	72	50	91	***49***	92	90	93	72	112	***91***	79	78	80	59	99
8	63	62	65	43	83	***50***	83	81	85	63	103	***92***	76	75	77	56	96
9	84	83	85	64	104	***51***	87	86	88	67	107	***93***	84	83	85	64	104
10	87	85	88	67	107	***52***	85	83	86	65	105	***94***	83	82	84	63	103
11	83	81	84	63	103	***53***	85	84	87	65	105	***95***	67	66	69	47	87
12	87	86	89	67	108	***54***	58	55	60	37	78	***96***	70	68	71	50	90
13	76	74	77	56	96	***55***	64	62	66	44	84	***97***	83	81	85	63	103
14	71	70	72	51	91	***56***	93	90	95	72	113	***98***	84	83	86	64	105
15	76	75	77	56	96	***57***	90	88	93	70	110	***99***	76	74	77	56	96
16	87	85	88	66	107	***58***	73	71	74	52	93	***100***	92	90	93	71	112
17	85	84	87	65	106	***59***	82	81	84	62	102	***101***	87	86	89	67	107
18	82	80	83	62	102	***60***	94	92	96	74	114	***102***	80	79	81	60	100
19	84	82	85	64	104	***61***	69	67	70	48	89	***103***	81	80	83	61	102
20	71	69	72	50	91	***62***	77	75	78	56	97	***104***	74	73	75	54	94
21	78	77	79	58	98	***63***	89	88	90	69	109	***105***	80	79	82	60	101
22	84	83	86	64	105	***64***	82	81	83	62	102	***106***	91	90	92	71	111
23	67	66	69	47	88	***65***	81	79	82	60	101	***107***	78	77	79	58	98
24	81	79	83	61	101	***66***	84	82	85	64	104	***108***	75	74	76	55	95
25	63	61	65	43	84	***67***	93	91	94	72	113	***109***	93	91	96	73	114
26	92	90	93	72	112	***68***	71	69	72	51	91	***110***	80	80	81	60	100
27	89	88	90	69	109	***69***	88	86	89	67	108	***111***	51	49	53	31	71
28	85	84	87	65	105	***70***	79	78	81	59	100	***112***	92	91	94	72	112
29	84	82	86	64	104	***71***	86	85	88	66	106	***113***	68	66	69	48	88
30	80	78	81	59	100	***72***	85	84	86	65	105	***114***	76	75	77	56	96
31	86	84	87	66	106	***73***	96	94	98	76	116	***115***	74	73	75	54	94
32	77	76	77	57	97	***74***	91	89	92	70	111	***116***	75	74	76	55	95
33	91	89	92	71	111	***75***	65	63	68	45	86	***117***	81	80	82	61	101
34	82	79	85	62	102	***76***	96	94	98	76	116	***118***	78	76	79	58	98
35	87	86	88	67	107	***77***	99	97	101	79	119	***119***	84	83	85	64	104
36	81	79	82	61	101	***78***	88	86	89	68	108	***120***	83	82	84	63	103
37	92	91	94	72	112	***79***	91	89	93	71	111	***121***	78	77	79	58	98
38	73	71	74	53	93	***80***	78	77	80	58	98	***122***	61	59	62	41	81
39	65	63	66	45	85	***81***	73	72	74	53	93	***123***	77	76	78	57	97
40	79	78	80	59	99	***82***	82	81	83	62	102	***124***	68	67	70	48	89
41	74	72	75	54	94	***83***	87	86	89	67	108	***125***	92	90	93	72	112
42	81	80	83	61	101	***84***	74	72	75	54	94	***126***	81	77	84	60	101

in.	fit	c-l	c-u	p-l	p-u	in.	fit	c-l	c-u	p-l	p-u
127	106	104	109	86	127	***169***	81	80	82	61	101
128	114	112	116	94	134	***170***	78	76	80	58	98
129	100	98	102	80	120	***171***	96	94	98	76	116
130	108	106	110	88	128	***172***	75	74	76	55	95
131	103	101	105	83	123	***173***	82	81	83	62	102
132	100	98	101	79	120	***174***	72	70	73	51	92
133	87	85	88	67	107	***175***	69	68	71	49	89
134	86	85	88	66	107	***176***	77	75	78	56	97
135	72	71	73	52	92	***177***	69	67	71	49	89
136	79	78	80	59	99	***178***	81	80	82	61	101
137	93	92	95	73	113	***179***	78	77	79	58	98
138	83	82	84	63	103	***180***	78	76	79	57	98
139	93	91	94	73	113	***181***	84	83	85	64	104
140	79	77	80	58	99	***182***	96	95	98	76	117
141	79	77	80	59	99	***183***	86	85	87	66	106
142	84	83	86	64	105	***184***	88	87	89	68	108
143	69	67	70	48	89	***185***	80	79	80	60	100
144	74	73	75	54	94	***186***	76	75	77	56	96
145	81	80	82	61	101	***187***	75	74	76	55	95
146	100	97	103	80	120	***188***	81	80	83	61	102
147	90	88	91	70	110	***189***	72	70	73	52	92
148	87	86	88	67	107	***190***	87	86	88	67	107
149	86	85	87	66	106	***191***	90	89	92	70	110
150	67	66	69	47	88	***192***	82	81	83	62	102
151	70	68	71	50	90	***193***	79	78	80	59	99
152	66	64	68	46	86	***194***	60	59	62	40	81
153	61	59	63	41	81	***195***	82	80	84	62	102
154	72	71	73	52	92	***196***	76	75	78	56	97
155	95	93	97	75	115	***197***	84	83	85	64	104
156	85	84	86	65	105	***198***	74	73	75	54	94
157	84	83	85	64	104	***199***	84	83	86	64	104
158	72	70	73	52	92	***200***	78	77	80	58	99
159	79	78	80	59	99	***201***	70	69	72	50	90
160	76	75	78	56	96	***202***	80	78	81	60	100
161	96	93	99	76	116	***203***	83	82	85	63	104
162	81	80	82	61	101	***204***	86	85	88	66	106
163	89	88	90	69	109	***205***	71	69	73	51	91
164	74	73	76	54	94						
165	76	75	78	56	96						
166	88	86	90	68	108						
167	64	62	65	44	84						
168	70	69	72	50	90						

My data analysis was performed using R. The packages and code used for my plots and analysis can be found here:

```
library(psych)
library(car)
library(coefplot)

training <- read.csv("moneyball-training-data.csv", header=T)
evaluation <- read.csv("moneyball-evaluation-data.csv", header=T)
training <- training[,-1] # remove index

# plot histograms of all variables

par(mfrow=c(2,4))
for (i in 1:8){
  hist(training[,i], breaks=20, xlab=colnames(training)[i],main=NA)
}

par(mfrow=c(2,4))
for (i in 9:16){
  hist(training[,i], breaks=20, xlab=colnames(training)[i],main=NA)
}

# summary stats
summary(training)

# pct of seasons with fewer than 60 losses or more than 102 wins

nrow(training[training$TARGET_WINS<60 | training$TARGET_WINS>102,])*100/nrow(training)

# examine strikeout outliers

subset(training,TEAM_PITCHING_SO>2500)

# delete from dataframe

training <- training[-c(1,282,1342,1826,2136),]

# fixed SO distribution

hist(training$TEAM_PITCHING_SO, breaks=20, main="Adjusted P_SO Distribution",xlab=NA)

# check for NA values and measure proportion

na_count <-sapply(training, function(y) sum(length(which(is.na(y)))))
na_pct <- na_count*100/nrow(training)
na_pct <- data.frame(na_pct)
na_pct

#next look at a correlation matrix

training <- training[,-10] # remove HBP
corrMatrix <- cor(training, use = "na.or.complete")
#pVal <- corr.test(corrMatrix,y=NULL)
#pVal <- data.frame(pVal)
roundedCM <- round(corrMatrix,2)

# Fit all predictors individually and plot error distributions

fitList <- list()
```

```
par(mfrow=c(2,4))

for(i in 2:15){
  fitName <- colnames(training)[i]
  fitList[[ fitName ]] <- lm(TARGET_WINS~training[,i],training)
}

par(mfrow=c(2,4))
for (i in 2:9){
  hist(summary(fitList[[i]])$residuals,xlab="error",main=paste("WINS ~",names(fitList)[i-1]),breaks=20)
}

par(mfrow=c(1,5))
for (i in 10:14){
  hist(summary(fitList[[i]])$residuals,xlab="error",main=paste("WINS ~",names(fitList)[i-1]),breaks=20)
}


# Repeat for square of residuals to check for homoscedasticity

par(mfrow=c(2,4))
for (i in 2:9){
  plot(summary(fitList[[i]])$residuals^2,ylab="Residual Squared",main=paste("WINS ~",names(fitList)[i-1]))
}

par(mfrow=c(1,5))
for (i in 10:14){
  plot(summary(fitList[[i]])$residuals^2,ylab="Residual Squared",main=paste("WINS ~",names(fitList)[i-1]))
}

# find best r-squared of single predictors
# high value is fitList[[1]]

for (i in 1:length(fitList)){
  print(summary(fitList[[i]])$r.squared)
}

fitAll <- lm(TARGET_WINS ~., training)
fitAll <- update(fitAll, . ~ . -TEAM_PITCHING_BB) #R-squared 0.4386
fitAll <- update(fitAll, . ~ . -TEAM_PITCHING_HR) #R-squared 0.4386
fitAll <- update(fitAll, . ~ . -TEAM_BATTING_SO) #R-squared 0.4384

summary(fitAll)$r.squared
summary(fitAll)$coefficients

# impute missing values

tmeans <- apply(training, 2, mean, na.rm=T)
tmedian <- apply(training, 2, mean, na.rm=T)
trainingImputed <- training
for (i in c(9,13,15)){
  for (j in 1:nrow(trainingImputed)){
    if (is.na(trainingImputed[j,i])){
      trainingImputed[j,i] <- tmeans[i]
    }
  }
}
for (i in c(7,8)){
  for (j in 1:nrow(trainingImputed)){
    if (is.na(trainingImputed[j,i])){
```

```
    trainingImputed[j,i] <- tmedian[i]
   }
  }
}

# backwards selection, model 2

fitAll2 <- lm(TARGET_WINS ~., trainingImputed)

summary(fitAll2)  #R-squared 0.3213, repeat for each

fitAll2 <- update(fitAll2, . ~ . -TEAM_PITCHING_HR) # R-squared 0.3212
fitAll2 <- update(fitAll2, . ~ . -TEAM_PITCHING_H) # R-squared 0.3211
fitAll2 <- update(fitAll2, . ~ . -TEAM_BASERUN_CS) # R-squared 0.3209

summary(fitAll2)$coefficients

# model 3

trainingImputed2 <- trainingImputed
for (i in 2:15){
  for (j in 1:nrow(trainingImputed2)){
   if (trainingImputed2[j,i]==0){
    trainingImputed2[j,i] <- tmeans[i]
   }
  }
}

fitAll3 <- lm(TARGET_WINS ~., trainingImputed2) #R-squared 0.318
fitAll3 <- update(fitAll3, . ~ . -TEAM_BATTING_HR) #R-squared 0.318
fitAll3 <- update(fitAll3, . ~ . -TEAM_BASERUN_CS) #R-squared 0.318

# check variance inflations factors

vif(fitAll)
vif(fitAll2)
vif(fitAll3)

# BIC plot

full <- regsubsets(TARGET_WINS ~ ., training, nvmax = 15)
par(new=True)
plot(summary(full)$bic, xlab = "Number of Predictors", ylab = "BIC", type = "l",
    main = "Best Subset Selection Using BIC")

# predict evaluation set, format table in excel

predict(fitAll,evaluation,na.action = na.exclude,interval = "confidence")
predict(fitAll,evaluation,na.action = na.exclude,interval = "prediction")

# compare coefficients for all models

multiplot(fitAll,fitAll2,fitAll3)

# plot of residuals for fitAll

hist(summary(fitAll)$residuals,xlab="error",main="Histogram of fitAll Residuals",breaks=20)
```