A WINNING STRATEGY

| OCTOBER 09, 2016 | *Russ Conte* | *Kaggle name: russconte* |

Table of Contents:

1. Introduction
2. The Goal
3. Data Exploration

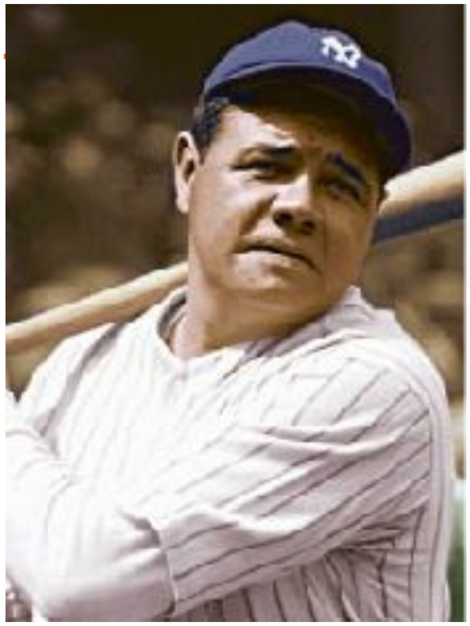
3. Data Preparation
4. Building Models
5. Selecting Models

6. Model Deployment Code
7. A Winning Strategy For Our Team

Introduction

This analysis is the first in a series that will allow our business to expand the solutions we offer to our customers. This report will show solid results in a clear presentation.

Once we prove we can do this analysis, then our business can submit bids for many other types of analyses. A similar method of analysis can yield profitable results in every field from insurance to education. I am very proud to submit this analysis for your review.



The Goal

The data are not the actual data from 1900-1950 Major League Basebll, but are transformed and modified in several ways.

The data consists of two parts. The first part contains 2276 rows of data and 13 rows of data. However, the data are missing 259 rows of data that we will use to test our predictions.

The goal is to use the first part of data to create a model to predict the number of wins, and test that model using the 259 rows from the test data.

A WINNING STRATEGY

| OCTOBER 09, 2016 | *Russ Conte* | *Kaggle name: russconte* |

Honus Wagner 1910 baseball card. Approximate value: $2,100,000.

He's one of the players in our data set.

Data Exploration:

Our data consists of 2276 rows of data. This is *not* the actual data, but manipulated data from Major League Baseball from 1900-1950. We can create some simple summary data from the maximum and minimum values of our data:

Table 1:

Variable	Label	Maximum	Minimum
INDEX		2535.00	1.0000000
TARGET_WINS		146.0000000	0
TEAM_BATTING_H	Base Hits by batters	2554.00	891.0000000
TEAM_BATTING_2B	Doubles by batters	458.0000000	69.0000000
TEAM_BATTING_3B	Triples by batters	223.0000000	0
TEAM_BATTING_HR	Homeruns by batters	264.0000000	0
TEAM_BATTING_BB	Walks by batters	878.0000000	0
TEAM_BATTING_SO	Strikeouts by batters	1399.00	0
TEAM_BASERUN_SB	Stolen bases	697.0000000	0
TEAM_BASERUN_CS	Caught stealing	201.0000000	0
TEAM_BATTING_HBP	Batters hit by pitch	95.0000000	29.0000000
TEAM_PITCHING_H	Hits allowed	30132.00	1137.00
TEAM_PITCHING_HR	Homeruns allowed	343.0000000	0
TEAM_PITCHING_BB	Walks allowed	3645.00	0
TEAM_PITCHING_SO	Strikeouts by pitchers	19278.00	0
TEAM_FIELDING_E	Errors	1898.00	65.0000000
TEAM_FIELDING_DP	Double Plays	228.0000000	52.0000000

The most critical factor is to find which of these variables contribute to TARGET_WINS. This is a correlation table that shows the correlation of each variable with TARGET_WINS:

Table 2:

A WINNING STRATEGY

| OCTOBER 09, 2016 | *Russ Conte* | *Kaggle name: russconte* |

Pearson Correlation Coefficients Prob > |r| under H0: Rho=0 Number of Observations	
	TARGET_WINS
TEAM_BATTING_H Base Hits by batters	0.38877 <.0001 2276
TEAM_BATTING_2B Doubles by batters	0.28910 <.0001 2276
TEAM_BATTING_3B Triples by batters	0.14261 <.0001 2276
TEAM_BATTING_HR Homeruns by batters	0.17615 <.0001 2276
TEAM_BATTING_BB Walks by batters	0.23256 <.0001 2276
TEAM_BASERUN_SB Stolen bases	0.13514 <.0001 2145
TEAM_BASERUN_CS Caught stealing	0.02240 0.3853 1504
TEAM_BATTING_HBP Batters hit by pitch	0.07350 0.3122 191
TEAM_PITCHING_H Hits allowed	-0.10994 <.0001 2276
TEAM_PITCHING_HR Homeruns allowed	0.18901 <.0001 2276
TEAM_PITCHING_BB Walks allowed	0.12417 <.0001 2276
TEAM_PITCHING_SO Strikeouts by pitchers	-0.07844 0.0003 2174
TEAM_FIELDING_DP Double Plays	-0.03485 0.1201 1990

What we learn in the correlation table is that most of the variables are correlated with wins, but a few are not. The values with a positive number are positively correlated with our team winning, and the ones with a negative number are negatively correlated. In other words, the numbers that are negative increase our chases of losing.

A few of the results are surprising. For example, Team_Pitching_Strike_Outs are *negatively* correlated with winning games. In other words, when our pitchers strike out batters we are more likely to *lose* games. Not by a lot, but that's what the data show. It's the same story with double plays by the fielding team. There is a slight negative correlation between pulling off double plays and losing the game. That's not what we would expect, but it's what the data show.

Some very odd points in the data:

The Proc Means (prior page) shows some numbers that are clearly impossible in Major League Baseball. For example, the training data shows at least one team having 146 wins in a season, and another team having exactly zero wins in a season. Neither of those have actually happened in MLB, so the data will be adjusted to adjust for those anomolies in the data.

Another huge anomoly is Strikeouts by Pitchers. The maximum is 19,278, clearly impossible. Even Cy Young wasn't that good!

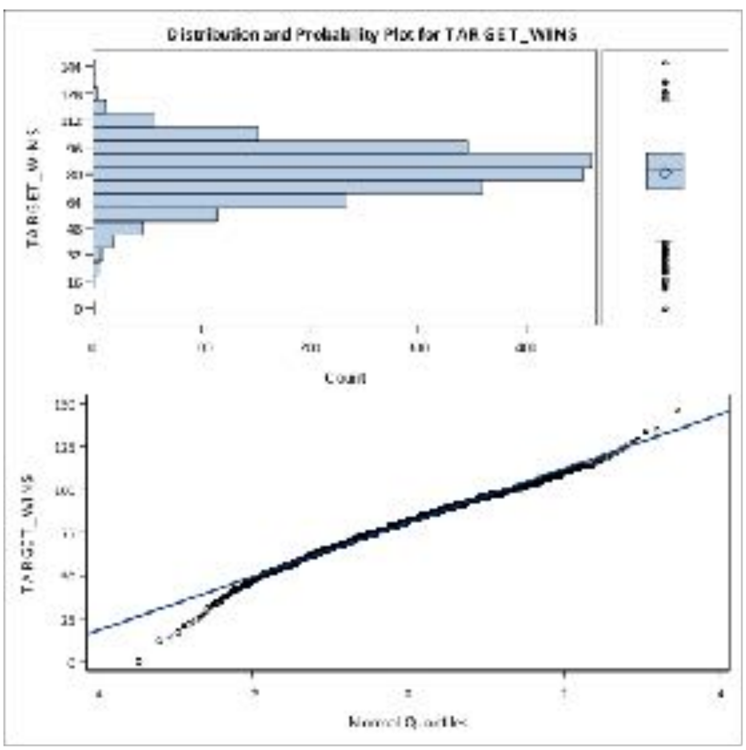
The count of the Missing data:

As we explore the data, we find a number of missing data points. Actually a lot of them. There are literally hundreds of missing data points in the Training data set. We will accommodate these missing data points in our analysis. Here are the counts of the missing data points in the Training data set:

Table 3:

Missing data in the Training data set	
Strikeouts by batters	102
Stolen Bases	131
Runners Caught Stealing	772
Batters Hit by Pitch	2085
Strikeouts by Pitchers	102
Double Plays	286

One other noteworthy point in the exploratory data analysis: The average number of wins is 82, exactly 50% of the games out of our 182 game season. The vast majority of the results line with a bell shaped curve.



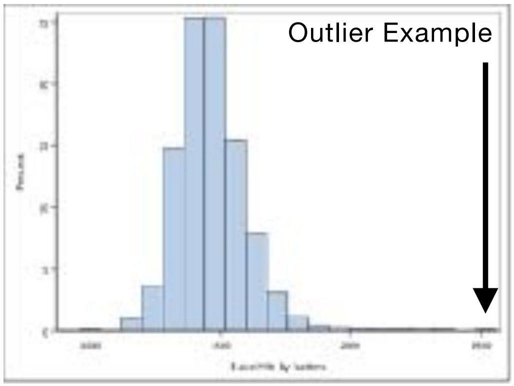
Part 4 Data Preparation

What we are going to do in this part is clean up the mess we received. There are a number of serious problems with the data as we noted above.

We're going to take several steps to clean up the data. If the testing data set was a dog it would need a bath, mainly due to the missing data.

The strategy with the missing data will be to replace it by the average value for that value. For example, the average number of hits in our data set is 1,469. If a data point is missing, we'll put 1,469 in that spot. We'll do that for every variable in our training and testing data set.

The other big problem with the training data set are outliers. Those are points that are so extreme they literally have never happened in the entire history of Major League Baseball. Our job will be to get rid of these outliers. Let's look at a few:



Part 5 Building Models including a model with $R^2 > 0.50$

A linear regression was run using SAS, and the method was forward. The results are:

Variable	Parameter Estimate
Intercept	13.91565
Hits	0.04801
Doubles	-0.01172
Triples	0.05256
Base on Balls	0.01482
Strikeouts	-0.00769
Stolen Bases	0.03051
Caught Stealing	-0.02230
Hit by pitch	0.10279
Hits off pitcher	-0.00061774
Homeruns off pitcher	0.05925
Base on balls off pitcher	-0.00413
Strikeouts off our pitcher	0.00320
Errors by fielders	-0.02144
Double play by fielder	-0.12041

What we learn from this analysis is that the baseline is 13 wins, and that number can go up or down depending on the other variables. The strongest factor that can help us win is when our batters get hits. Second after that is when our fielding team gets a double play or when our fielding team makes an error. The next two most significant factors are when there is a homerun off our pitcher, and a successful stolen base.

It should be noted that a couple of these results are very counterintuitive. For example, the analysis says that we win more games when our batter strikes out, or when there is a hit off our pitcher. Neither of these are intuitively obvious, and merit more investigation.

Model #2: Backward Regression

The same data were used to run the regression using a Backward Methodology. The results are:

Results of Backwards Analysis	
Intercept	14.43474
Batting team hits	0.04516
Batting team triples	0.05236
Batting team base on balls	0.01042
Batting team strike outs	-0.00723
Batting team baserunners stolen bases	0.02883
Batting team hit by pitch	0.10578
Pitching hits	-0.00084263
Pitching home runs	0.059263
Pitching team getting strike outs	0.00243
Fielding errors	-0.02045
Fielding double plays	-0.12078

The values in model #2 are similar to model #1 in their strengths. The most significant factor is getting hits, followed by getting a double play. The next three are fielding errors (negatively correlated), when the pitcher gives up a home run, and stolen bases.

Both of these models have correlations around 0.31, meaning the model accounts for around 31% of the results.

Model #3: An adjusted R^2 of 0.4698, much higher than the other two models

For our third model, I used the R programming language. This is a different programming language than SAS, and can do some of the same things as SAS, in particular it does linear regression. Here are the results from the analysis performed in R, which only takes five lines of code:

```
library(MASS)
baseball_train=read.csv(file = 'baseball1.csv',header = TRUE,sep = ',')
fit=lm(TARGET_WINS~.,data = baseball_train)
fit_best=stepAIC(object = fit,direction = "backward")
summary(fit_best)
```

That's it. Just five lines of code, no imputation, no working on missing values or transformations or calculating other variables (R does all of that for me), and we have an adjusted R^2 of 0.4698 and other very good results, all statistically significant at the $p<0.10$ level (and most are significant at much more than $p<0.10$), as the results show below:

```
Call:
lm(formula = TARGET_WINS ~ TEAM_BATTING_HR + TEAM_BATTING_BB +
    TEAM_BATTING_HBP + TEAM_PITCHING_H + TEAM_PITCHING_SO + TEAM_FIELDING_DP,
    data = baseball_train)

Coefficients:
                     Estimate     Std. Error    t value    Pr(>|t|)
(Intercept)          40.727939    19.494101     2.089      0.038061 *
TEAM_BATTING_HR      0.080567     0.024956      3.228      0.001474 **
TEAM_BATTING_BB      0.062121     0.009773      6.356      1.58e-09 ***
TEAM_BATTING_HBP     0.092116     0.050808      1.813      0.071457 .
TEAM_PITCHING_H      0.032024     0.010500      3.050      0.002627 **
TEAM_PITCHING_SO     -0.038856    0.007401      -5.250     4.16e-07 ***
TEAM_FIELDING_DP     -0.130519    0.036722      -3.554     0.000482 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.822 on 184 degrees of freedom
  (2085 observations deleted due to missingness)
Multiple R-squared:  0.4865, Adjusted R-squared:  0.4698
F-statistic: 29.05 on 6 and 184 DF,  p-value: < 2.2e-16
```

Here is the Sum of Squares, RSS and AIC for this model:

```
Step:  AIC=838.57
TARGET_WINS ~ TEAM_BATTING_HR + TEAM_BATTING_BB + TEAM_BATTING_HBP +
    TEAM_PITCHING_H + TEAM_PITCHING_SO + TEAM_FIELDING_DP
```

	Df	Sum of Sq	RSS	AIC
<none>			14320	838.57
- TEAM_BATTING_HBP	1	255.82	14576	839.96
- TEAM_PITCHING_H	1	723.94	15044	845.99
- TEAM_BATTING_HR	1	811.14	15131	847.10
- TEAM_FIELDING_DP	1	983.19	15303	849.26
- TEAM_PITCHING_SO	1	2145.42	16466	863.24
- TEAM_BATTING_BB	1	3144.36	17464	874.49

Part 5 Select the Model

The three models we chose all have positive qualites, but model #3 has much more adjusted R^2 than the other two models, or any other model I could come up with using SAS. Model #3 is also much easier to understand since it has the fewest terms, and we can convey this to our front office easier than the other two models. It's important to give front office a model that they can use, and model #3 is easier to use than the other two.

Model #3 also has higher AIC scores, and runs much faster than the other two models in SAS. Model #3 in R runs in well under one second, but models 1 and 2 take quite a lot longer to run.

Another advantage to Model #3 is that it is only five lines of code. The SAS model runs close to 200 lines of code to run just one model, so the R model is much simpler. To be fair each model in SAS is not almost 200 lines of code, but it would require the nearly the entire 200 lines to run and debug.

For all of these reasons we Model #3 is the top choice.

Part 6 Stand Alone Scoring Program

The SAS code to use the results from Model #3 to create an output file are:

```
P_TARGET_WINS = 40.727939 + 0.080567*TEAM_BATTING_HR + 0.062121*TEAM_BATTING_BB
+0.092116*TEAM_BATTING_HBP + 0.032024*TEAM_PITCHING_H
-0.038856*TEAM_PITCHING_SO -0.130519*TEAM_FIELDING_DP;

*Check for missing values in scoredfile*;
proc means data=scoredfile nmiss mean min max;
run;

keep INDEX P_TARGET_WINS;
run;
proc print data=scoredfile;
run;
proc means data=scoredfile N NMISS MIN MAX;
```

```
var P_TARGET_WINS;
run;
```

Conclusion: A World Series Victory for Our Team

As you know, we have an uphill battle winning the championship next year. That's not news to anyone, from our players to the fans who buy tickets to our advertisers. My very strong recommendation is to expand our analytics capabilities, such as the Boston Red Sox did a few years ago, and the Chicago Cubs are doing this year. Analytics played a huge role in the World Series victory for the Red Sox, and has given the Cubs over 100 wins this year and the best record in baseball. We can go down the same path, and do our best to improve our results.

If we've learned anything from this assignment, it's that regression is just the beginning. Winning the World Series is the goal, not just a set of equations and graphs. The equations can help, but I hope to be part of the group that crafts our team into World Series Champions. I know analytics can help, I'm looking forward to using my skills in analytics to help us achieve our common goal!