

# Using Time Series to Model American Football Teams' Yearly Winning Percentage

Samuel Greeman

April 29, 2021

## 1 Introduction

Previously, I have researched if a Major League Baseball team's yearly winning percentage could be modeled and forecasted by time series, and suprisingly, it was rather effective at doing so. Today, our goal shifts to finding if the same process and models will work for the NFL/American Football. In our baseball analysis, we found that a Holt-Winters forecast method and an ARIMA(0, 1, 1) model (meaning 0 autoregressive terms, 1 difference term, and 1 moving average term) were the best and most accurate models, so we will use those on our NFL data [1] in addition to the best ARIMA model, determined by lowest AIC. The team whose data we will use is the New York Giants. As far as the data itself, a few observations and modifications need to be made. First, teams play 16 games per year, with some exceptions. Due to those exceptions, I found it would be better to use winning percentage as I did with the baseball analysis, in order to normalize the number of games played per year. Typically in the NFL, winning percentages vary far more than in other leagues, as playing only 16 games means there is potential for teams to have a 0 winning percentage, a 1.000 winning percentage, and anywhere in between. Another decision I made was in which year to start the data. 1966 seemed like the perfect year to start, as it marks the merge of the AFL and the NFL, and in turn, the year of the first Super Bowl. With everything introduced, let's hop into the analysis.

## 2 Initial Analysis

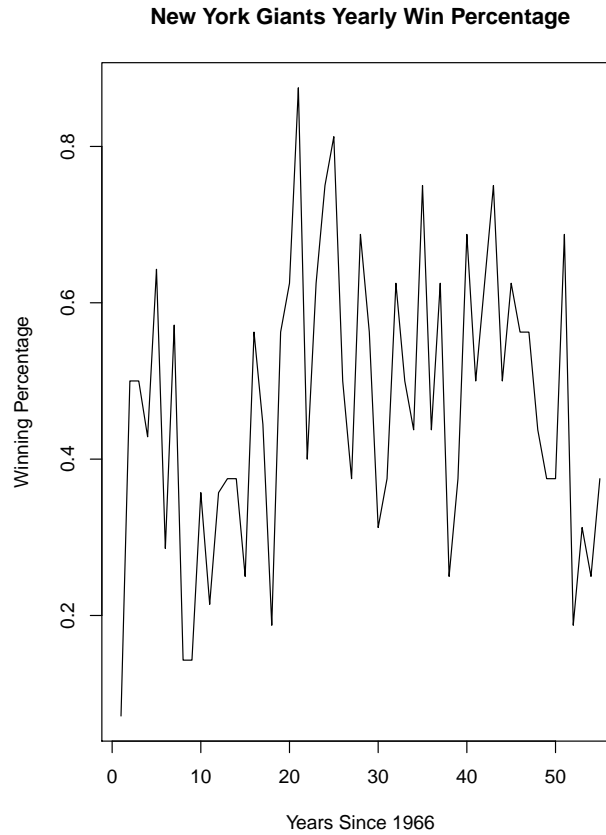


Figure 1: Plot of Time Series Data

Figure 1 shows us that the Giants have typically been in the range of 0.200 to 0.800 winning percentage, with a mean of 0.467. Since this is close to the 50 percent mark, it tells us that we can perhaps use this analysis to represent other NFL teams as well, with different means and parameter values. A problem, however, is stationarity. For our models to be applicable to the data, we have to have stationarity, meaning our mean and variance don't change over time, and from Figure 1, we can see that the former is not necessarily true in our case so we will attempt to eliminate this trend via differencing.

### 3 Differencing

If we difference our data, we can eliminate the trend of the mean that we saw in Figure 1. The procedure is to have the difference between consecutive terms be our data points and if we choose our order,  $k$ , to be 1, the data points will be  $d(X_1) = X_2 - X_1$ ,  $d(X_2) = 0$ , and  $d(X_t) = X_t - X_{t-1}$  for the rest of the data points. Applying this transformation, our new data looks like this:

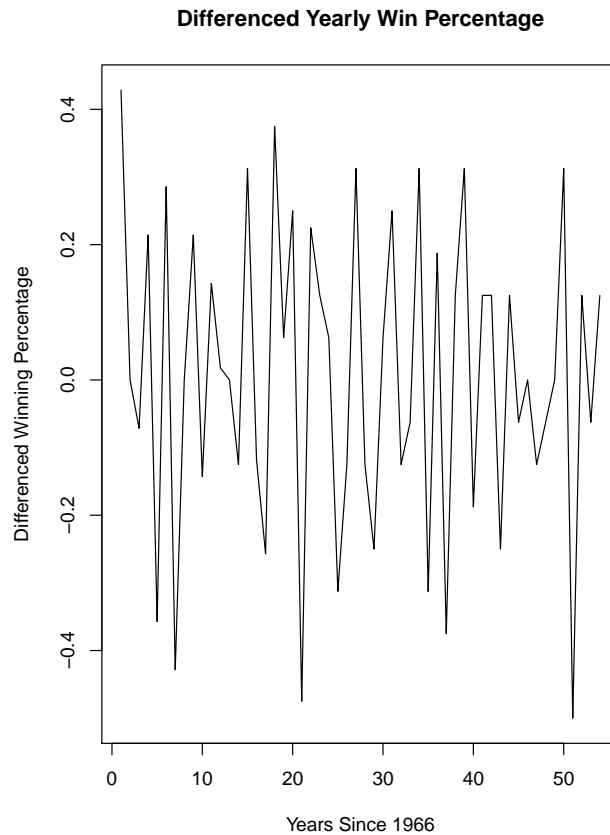


Figure 2: Plot of Differenced Time Series Data

Now we can see that the mean stays relatively constant over time, so in our next section, we can try to create an ARIMA model with a difference order of 1.

## 4 Box-Cox Transformation

We already have the outline for three separate models. We have an ARIMA(p, 1, q) model, which we need to find the p and q for. Next, we plan to use Holt-Winters forecasting as a model, which we will discuss later. Lastly, assuming that the ARIMA(p, 1, q) model is not the optimal model, then we will find a third model that has a lower AIC. One final check before we do those things is to make sure that differencing our data was enough to make it stationary. Clearly, looking at Figure 2, we can see that the mean is stationary, meaning it is constant over time. We have yet to check the stationarity of the variance, and one way to do this is by using a Box-Cox transformation on our data. Box-Cox transformation will assign a value of  $\lambda$  to the data. This  $\lambda$  is optimized to keep the variance constant over time. If the value of  $\lambda$  is 0, then the transformation is a logarithmic transformation. If it is not 0, then the data,  $X_t$  is transformed to  $\frac{(X_t^\lambda - 1)}{\lambda}$ . Using the R function `BoxCox.lambda`, we find that the optimal value of  $\lambda$  for our data is 0.7191. However, optimal is a relative term. In Table 1, we ran Dickey-Fuller tests on our original data and on both of our transformations. The Dickey-Fuller test tests the null hypothesis, which is the data is nonstationary, against the alternative hypothesis, which is the data is stationary. It generates a t-statistic and corresponding p-value, and in our case, if the p-value is less than 0.05, we reject the null hypothesis.

Transformation	DF value	p-value	Reject?
None	-1.808	0.652	No
Differenced	-5.759	< 0.01	Yes
Box-Cox ( $\lambda = 0.7191$ )	-1.883	0.622	No

Table 1: Stationarity of Transformed Data

Table 1 tells us that applying the optimal Box-Cox transformation does not result in stationarity for the data, but differencing does. This means that Box-Cox transformation is not needed, and we can move on to finding our 2 ARIMA models.

## 5 ARIMA Model Selection

One of the simplest ways to select an appropriate ARIMA model for our data is to look at both the autocorrelation function plot, and the partial autocorrelation function plot. These plots are likely to have distinguishing features that help point us to the correct order for our model. Since we know that one of our models will have difference order of 1, we can look at the ACF and PACF plots of the differenced data in Figure 3 to find out what order of AR(p) and MA(q) we should use.

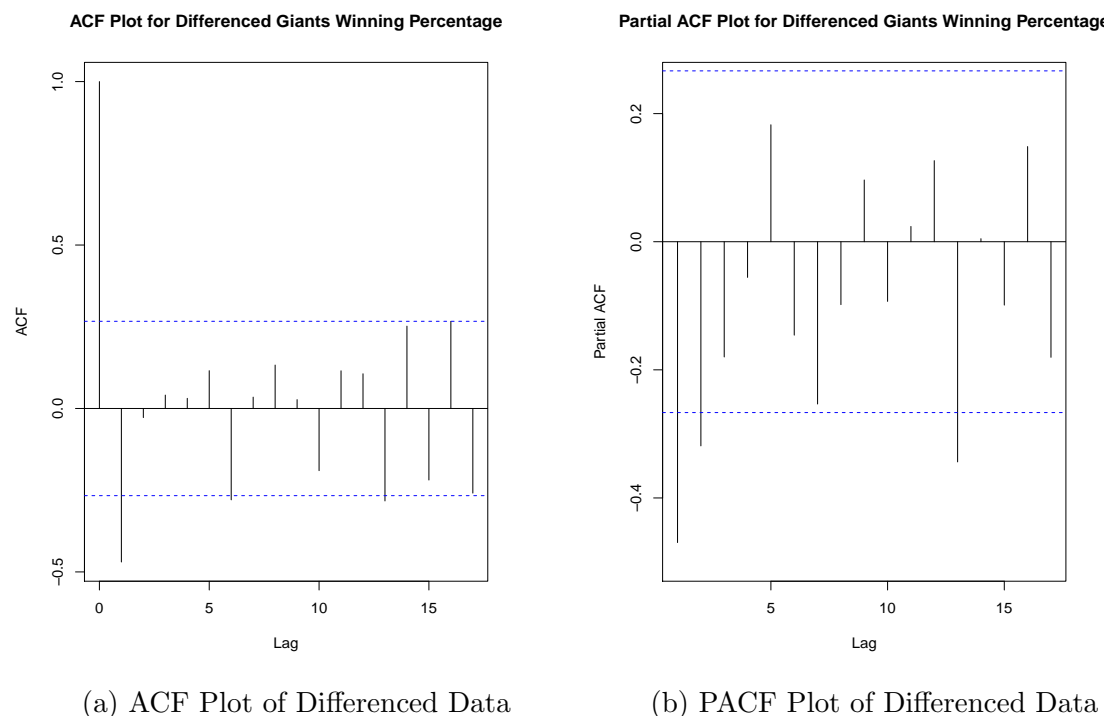


Figure 3: PACF and ACF Plots for Differenced Data

We see in the ACF plot in Figure 3 that the ACF at lag 1 is around -0.47, after which it cuts off to 0. For the PACF plot, the PACF value at lag 1 is also -0.47, and it slowly decays to 0 after lag 1. This pattern is consistent with an MA(1) model, meaning our AR order,  $p$ , is 0, and our MA order,  $q$ , is 1. We can even approximate the value of the MA parameter using the following relation:  $\rho_1 = \frac{\theta}{1+\theta^2}$ . Here, our  $\rho_1$ , which is the ACF at lag 1, is approximately -0.47. Solving for  $\theta$ , our parameter value, we get -0.701. This value is close to R's more precise value of -0.7725, and

we will use the value that R gives us for that reason. So, our first ARIMA(p, d, q) model is ARIMA(0, 1, 1), with the MA parameter equal to -0.7725. The same ARIMA order was found to be the optimal model order in the analysis of baseball teams, but to check if this is the case for the football analysis, we use the `auto.arima` function in R to generate the best order for our ARIMA model. It tells us that the optimal model is ARIMA(1, 0, 1), with mean of 0.4549, AR parameter  $\phi = 0.8241$ , and MA parameter  $\theta = -0.6467$ . R uses AIC to select the best order for an ARIMA model, but, as Table 2 shows, BIC and AIC disagree.

Model	AIC	BIC
ARIMA(0, 1, 1)	-26.89	-22.91
ARIMA(1, 0, 1)	-27.89	-19.86

Table 2: AIC and BIC for ARIMA(0, 1, 1) and ARIMA(1, 0, 1)

Lower is better for both AIC and BIC, and we can see in Table 2 that AIC favors ARIMA(1, 0, 1), but BIC favors ARIMA(0, 1, 1), so we will use both models when testing.

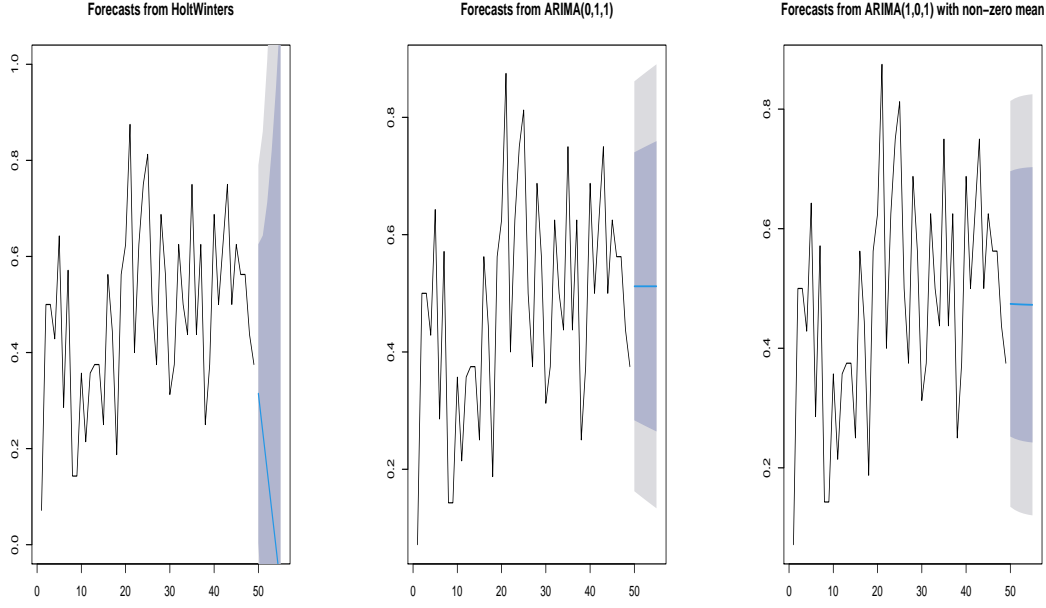
## 6 Model Testing

With our 2 ARIMA models complete, we need to introduce our third and final model to be tested with forecasting. We tested the Box-Cox transformation earlier in order to make our variance more constant over time. We found that while there was possible room for improvement, it wasn't room for significant improvement. The Holt-Winters method, which is the basis for our third model, uses exponential smoothing in addition to forecasting the trend of the mean to account for the possibility of both nonconstant mean, and nonconstant variance. The way in which we test these models is by using the first 49 years of our data as training data, and we use the models that are produced by the three methods to forecast the final 6 years of our data. We can compare these forecasts to the actual values of the final 6 years in order to determine the effectiveness of each model. All of these forecasts, including their 95 percent forecast intervals, are summarized in Table 3 below.

Year	Actual	Holt-Winters	ARIMA(0, 1, 1)	ARIMA(1, 0, 1)
2015	0.375	0.315, (-0.161, 0.790)	0.512, (0.163, 0.861)	0.474, (0.135, 0.814)
2016	0.688	0.234, (-0.392, 0.860)	0.512, (0.157, 0.867)	0.474, (0.130, 0.818)
2017	0.188	0.154, (-0.700, 1.008)	0.512, (0.151, 0.873)	0.473, (0.126, 0.821)
2018	0.313	0.074, (-1.066, 1.214)	0.512, (0.145, 0.879)	0.473, (0.123, 0.823)
2019	0.250	-0.007, (-1.477, 1.464)	0.512, (0.139, 0.885)	0.473, (0.122, 0.824)
2020	0.375	-0.087, (-1.925, 1.751)	0.512, (0.133, 0.890)	0.473, (0.120, 0.825)

Table 3: Forecasted Win Percentages from our 3 models

A few things are worth mentioning: First, the Holt-Winters method creates wide forecast intervals and as such, it forecasts the last 2 years as having negative win percentages, in addition to most of the forecast intervals having negative lower bounds and/or upper bounds greater than 1, both of which are not in the possible range of values for winning percentage. For our ARIMA(0, 1, 1) model, all forecast values are constant based on previous values, and the only changes in their forecasts are increasing the forecast intervals. Our ARIMA(1, 0, 1) model seems to make the most logical sense, as it predicts a downward trend based on the training data, and its forecasted values do not leave the range of possible values. Figure 4 shows the forecasts and their intervals in graphical form.



(a) Holt-Winters Forecast (b) ARIMA(0, 1, 1) Forecast (c) ARIMA(1, 0, 1) Forecast

Figure 4: Forecast Plots

## 7 Model Selection and Conclusion

As far as deciding on which model performed the best, we chose 2 error terms to compare for the three forecasts. The first, RMSE or root mean squared error is calculated as follows:  $\sqrt{\frac{\sum_{i=1}^n (x_i - \hat{x}_i)^2}{n}}$  where  $x_i$  is the observed value and  $\hat{x}_i$  is the predicted value. The second, MAE (mean absolute error) is calculated by:  $\frac{\sum_{i=1}^n |\hat{x}_i - x_i|}{n}$ . Values of these measures for all three models are in Table 4.

Model	RMSE	MAE
Holt-Winters	0.5008	0.2508
ARIMA(0, 1, 1)	0.3839	0.2059
ARIMA(1, 0, 1)	0.3298	0.1800

Table 4: Accuracy of Our Forecast Models



Once again, lower is better, and by both measures, it appears our best model is our ARIMA(1, 0, 1) model. While this is a different model than what was best for baseball teams, it is still a lower order ARIMA model, and surprisingly, instead of using a difference term, this ARIMA(1, 0, 1) model instead assumed a non-zero mean at lag 0. While we may have proven that there is a difference between sports when it comes to optimal ARIMA models for predicting winning percentage, I still believe that this model could be used as a basis for predicting winning percentage of all NFL teams, just with different parameter values and different means. The underlying logic behind our model for football definitely makes sense. Firstly, it assumes a general mean at lag 0 or "starting point" and uses previous data and trends to forecast future winning percentage. While I don't think these models will be accurate on a yearly basis, as these trends are often unpredictable, I do think this model and possible other models for other teams can be effective at forecasting the beginning and end of prolonged success or losing based on previous trends. As a result, I think that time series models for forecasting winning percentages of football teams are effective and accurate, especially in the long run or over a multiple-year period.

## 8 References

- [1] <https://www.pro-football-reference.com/teams/nyg/>