Predicting Win/Loss Totals of NFL Teams in 2019 and 2020

Dom Pizzano

1/4/2021

Predicting Total Wins and losses of NFL Teams from the 2019 and 2020 Seasons using Linear Regression

Overview:

Predicting Wins and losses for NFL teams given the total points scored and total points allowed. Then adding Turnover (Offense) and Takeaway (Defense) Data to see if those can be used to help better predict a team's total wins and losses for a Season. Will be using mainly the 2019 data to make a linear regression model predicting the Wins and Losses, and then compare that model to the 2020 Season to see if it is applicable.

Methods/Analysis:

First is to get the data, the NFL season data is from https://www.pro-football-reference.com/years/2019/index.htm, the data will be retrieved using web scraping tools and functions to parse out the data from the HTML.

Here is the data from the 2019 season, which comes in 2 HTML tables, one for the AFC and one for the NFC, so it is necessary to parse each table then combine them into one data frame. Also, unnecessary data rows will need to be removed from the data set.

```
#make call to the website you want to parse data from
url <- "https://www.pro-football-reference.com/years/2019/index.htm"
#read in the html
h <- read_html(url)
#get the table element from the html
tab <- h %>% html_nodes("table")
# read the two tables in (one for AFC and one for NFC) using html_table()
afc <- html_table(tab[1])
nfc <- html_table(tab[2])
# convert to data frames
afc <- as.data.frame(afc)
nfc <- as.data.frame(nfc)
# get rid of the extra division names in the table and for some reason the OR function is not working
afc <- afc %>% filter(Tm!="AFC North") %>%
  filter(Tm!="AFC East") %>%
  filter(Tm!="AFC South") %>%
  filter(Tm!="AFC West")
nfc <- nfc %>% filter(Tm!="NFC North") %>%
  filter(Tm!="NFC East") %>%
  filter(Tm!="NFC South") %>%
  filter(Tm!="NFC West")
#combine the afc and nfc data frames
nfl_df <- rbind(afc,nfc)
```

There is also a need to convert the columns into numeric data types, and get rid of special characters in the Team name column.

	Tm	W	L	T	W.L.	PF	PA	PD	MoV
	<chr>	<dbl>	<dbl>	<chr>	<chr>	<dbl>	<dbl>	<chr>	<chr>
1	NewEnglandPatriots	12	4	0	.750	420	225	195	12.2
2	BuffaloBills	10	6	0	.625	314	259	55	3.4
3	NewYorkJets	7	9	0	.438	276	359	-83	-5.2
4	MiamiDolphins	5	11	0	.313	306	494	-188	-11.8
5	BaltimoreRavens	14	2	0	.875	531	282	249	15.6

5 rows | 1-10 of 14 columns

Need to replace team name changes, since the Oakland Raiders became the Las Vegas Raiders and the Washington R****** became the Washington Football Team.

```
nfl_df$Tm<- str_replace_all(nfl_df$Tm, "OaklandRaiders", "LasVegasRaiders")
nfl_df$Tm<- str_replace_all(nfl_df$Tm, "WashingtonRedskins", "WashingtonFootballTeam")
```

There are 32 rows in the 2019 dataset, one row for each NFL team.

Now that the 2019 Data for each NFL team is available in a data frame, the data frame will be used to build linear models used to predict wins and losses.

The first NFL 2019 Linear regression model is labeled nfl_lm for the wins and nfl_lm_l for the losses.

```
nfl_lm <- lm(W ~ PF + PA , data = nfl_df)
nfl_lm_l <- lm(L ~ PF + PA , data = nfl_df)
```

With the linear models created from the 2019 data, we are now going to bring in 2020 NFL data, using the same URL from above and similarly processing the data.

	Tm	W	L	T	W.L.	PF	PA	PD	MoV
	<chr>	<dbl>	<dbl>	<chr>	<chr>	<dbl>	<dbl>	<chr>	<chr>
1	BuffaloBills	13	3	0	.813	501	375	126	7.9
2	MiamiDolphins	10	6	0	.625	404	338	66	4.1
3	NewEnglandPatriots	7	9	0	.438	326	353	-27	-1.7
4	NewYorkJets	2	14	0	.125	243	457	-214	-13.4
5	PittsburghSteelers	12	4	0	.750	416	312	104	6.5

5 rows | 1-10 of 14 columns

There are 32 rows in the 2020 dataset, one row for each NFL team, matching the 2019 dataset.

The points allowed and points scored totals for the 2020 and 2019 season have been accumulated, next is getting the Offensive Turnover and Defensive Takeaway totals into a data frame.

For this, the data needs to be copied and pasted from the same website above and put into a .csv file. My files are in the GitHub repository.

The offense and defense data is imported, then stripped of Takeaway and Turnover data in two different tables, then using an inner join function to pair the data correctly. TO are turnovers (Offensive) and TA are takeaways (Defensive)

```
# load in the csv with the pathways of the files on you computer
off_19 <- read.csv("C:/Users/dom/Documents/education/data-science/r/nfl-analysis/off-data-2019.csv",header = T)

def_19 <- read.csv("C:/Users/dom/Documents/education/data-science/r/nfl-analysis/def-data-2019.csv",header = T)

# change column names to the first row
colnames(off_19)<-off_19[1,]
colnames(def_19)<-def_19[1,]
# remove the first row
off_19 <- off_19[-1,]
def_19 <- def_19[-1,]

#remove spaces in Tm (team) to make sure we can use it to join the team win/loss data
off_19$Tm <-  str_replace_all(off_19$Tm,"[^[:alnum:]]", "")
def_19$Tm <-  str_replace_all(def_19$Tm,"[^[:alnum:]]", "")

# now we are going to get the turn over data data from the offense and defense by sub setting the data frames int
o select columns, then joining them
off_to <- off_19 %>% select(Tm,TO)
def_to <- def_19 %>% select(Tm,TO)
TO_df <- inner_join(off_to,def_to,by = "Tm")
# here TO.x is offensive turnover lost, TO.y is defensive takeaways
colnames(TO_df) <- c("Tm","TO","TA")

#make sure numeric columns are numbers
cols.num <- c("TO","TA")
TO_df[cols.num] <- sapply(TO_df[cols.num],as.numeric)

# now lets join this data with the original NFL data frame of the 2019 data
nfl_df3 <- inner_join(nfl_df,TO_df,by = "Tm")
head(nfl_df3 %>% select(Tm,W,L,PF,PA,TO,TA))
```

	Tm	W	L	PF	PA	TO	TA
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	NewEnglandPatriots	12	4	420	225	15	36
2	BuffaloBills	10	6	314	259	19	23
3	NewYorkJets	7	9	276	359	25	21
4	MiamiDolphins	5	11	306	494	26	16
5	BaltimoreRavens	14	2	531	282	15	25
6	PittsburghSteelers	8	8	289	303	30	38

6 rows

Then using the turnover and takeaway data from 2019 to make an updated linear model with Point Scored and Points Allowed. Calling it tot_lm for total (all variables (that is planned to be used)) linear model for predicting wins with 2019 data. totL_lm is for predicting losses with 2019 data

```
tot_lm <- lm(W ~ PF + PA + TO + TA, data=nfl_df3)
totL_lm <- lm(L ~ PF + PA + TO + TA, data=nfl_df3)
```

Then importing 2020 data that includes the nfl totals of takeaways and turnovers to see how the linear model works at predicting wins and losses for the 2020 season, using the linear models trained with the 2019 data.

	Tm	W	L	PF	PA	TO	TA
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	BuffaloBills	13	3	501	375	22	26
2	MiamiDolphins	10	6	404	338	20	29
3	NewEnglandPatriots	7	9	326	353	19	22
4	NewYorkJets	2	14	243	457	19	19
5	PittsburghSteelers	12	4	416	312	18	27
6	ClevelandBrowns	11	5	408	419	16	21

6 rows

Results:

All the data and models are created, now to use predict() function and newdata from 2019 and 2020 NFL seasons to see how well the linear models can predict:

(Linear model using Points Allowed and Points Scored Variables)

1.) Total Wins for each NFL Team in 2019

2.) Total Losses for each NFL Team in 2019

3.) Total Wins for each NFL Team in 2020

4.) Total Losses for each NFL Team in 2020

(Linear model using Points Allowed, Points Scored, Takeaways, Turnovers Variables)

5.) Total Wins for each NFL Team in 2019

6.) Total Losses for each NFL Team in 2019

7.) Total Wins for each NFL Team in 2020

8.) Total Losses for each NFL Team in 2020

(Linear model using Points Allowed and Points Scored Variables)

First, to see if the P values are significant enough to use the linear model in the calculation

```
Call:
lm(formula = W ~ PF + PA, data = nfl_df)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.9001 -0.8180  0.0385  0.5708  3.3685 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  8.347078   2.731297   3.056  0.00478 ** 
PF           0.025537   0.004514   5.657 4.09e-06 ***
PA          -0.026574   0.004662  -5.701 3.63e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.609 on 29 degrees of freedom
Multiple R-squared:  0.7647,	Adjusted R-squared:  0.7485 
F-statistic: 47.12 on 2 and 29 DF,  p-value: 7.737e-10
```

```
Call:
lm(formula = L ~ PF + PA, data = nfl_df)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.3598 -0.5900 -0.1053  0.7554  2.9245 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  8.023898   2.628659   3.052  0.00482 ** 
PF          -0.025663   0.004344  -5.907 2.06e-06 ***
PA           0.025512   0.004486   5.687 3.78e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.548 on 29 degrees of freedom
Multiple R-squared:  0.772,	Adjusted R-squared:  0.7563 
F-statistic: 49.11 on 2 and 29 DF,  p-value: 4.884e-10
```

With Both of these linear models, the variable Points For and Points Against have a high significance with a P-value under .001. Meaning these variables can be used for predicting total wins.

Create a newdata frame with the 2019 data to run the predictions for the 2019 Season, then adding the predictions to a compare table to see how it matches up to the actuals from the 2019 season. (w_p is predicted wins and l_p is predicted losses)

	Tm	W	L	PF	PA	w_p	l_p
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	NewEnglandPatriots	12	4	420	225	13.093617	2.985585
2	BuffaloBills	10	6	314	259	9.483167	6.573292
3	NewYorkJets	7	9	276	359	5.855381	10.099699
4	MiamiDolphins	5	11	306	494	3.034045	12.773934
5	BaltimoreRavens	14	2	531	282	14.413545	1.591162
6	PittsburghSteelers	8	8	289	303	7.675493	8.337402
7	ClevelandBrowns	6	10	335	393	6.458570	9.452983
8	CincinnatiBengals	2	14	279	420	4.310996	11.578946
9	HoustonTexans	10	6	378	385	7.769259	8.145370
10	TennesseeTitans	9	7	402	331	9.817133	6.151802

1-10 of 10 rows

Now to run a Root Means Square Error function comparing the actuals to the predicted so see how accurate the predictions were.

1.) 2019 Wins vs 2019 Predicted Wins:

```
[1] 1.53148
```

- Meaning on average the prediction of wins for 16 win season is off by 1.53 wins.

2.) 2019 losses vs 2019 Predicted Losses:

```
[1] 1.47393
```

- Meaning on average the prediction of losses for 16 game season is off by 1.47 losses.

Now to run the same 2019 model on the 2020 data to see if it compares.

Here is a sample of the actuals vs the predicted wins and losses for the 2020 season:

	Tm	W	L	PF	PA	w_p	l_p
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	BuffaloBills	13	3	501	375	11.176073	4.733680
2	MiamiDolphins	10	6	404	338	9.682191	6.279060
3	NewEnglandPatriots	7	9	326	353	7.291684	8.663469
4	NewYorkJets	2	14	243	457	2.408429	13.446767
5	PittsburghSteelers	12	4	416	312	10.679555	5.307789
6	ClevelandBrowns	11	5	408	419	7.631869	8.242886
7	BaltimoreRavens	11	5	468	303	12.246653	3.743695
8	CincinnatiBengals	4	11	311	424	5.021891	10.859773
9	IndianapolisColts	11	5	451	362	10.244671	5.685181
10	TennesseeTitans	11	5	491	439	9.219982	6.623085

1-10 of 10 rows

3.) 2020 Actual Wins vs Predicted wins

```
[1] 1.497605
```

- Meaning on average was off by predicting actual wins by 1.49, .04 more accurate on average than 2019 predictions.

4.) 2020 Actual losses vs Predicted Losses

```
[1] 1.457914
```

- Meaning on average was off by predicting actual losses by 1.45, .03 more accurate on average than 2019 predictions. Very similar to one another.

(Linear model using Points Allowed, Points Scored, Takeaways, Turnovers Variables)

Let's compare the four linear models to see if the new linear model is significant enough to use with the anova() function.

Comparing Wins Linear Models

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	29	75.05380	NA	NA	NA	NA
2	27	51.71617	2	23.33763	6.092059	0.006552766

2 rows

Comparing Losses Linear Models

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	29	69.51900	NA	NA	NA	NA
2	27	44.09852	2	25.42048	7.782042	0.002144502

2 rows

Both comparisons show that the new linear model is more accurate to use with the P-value under .05.

```
Call:
lm(formula = W ~ PF + PA + TO + TA, data = nfl_df3)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.95651 -1.04253  0.03943  0.99068  2.43326 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  5.422536   2.730096   1.986  0.05725 .  
PF           0.021571   0.004088   5.277 1.45e-05 ***
PA          -0.014371   0.005321  -2.701  0.01179 *  
TO          -0.135302   0.044525  -3.039  0.00522 ** 
TA           0.131630   0.052277   2.518  0.01804 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.384 on 27 degrees of freedom
Multiple R-squared:  0.8379,	Adjusted R-squared:  0.8138 
F-statistic: 34.88 on 4 and 27 DF,  p-value: 2.653e-10
```

```
Call:
lm(formula = L ~ PF + PA + TO + TA, data = nfl_df3)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.36248 -0.90834  0.01073  0.95082  2.13996 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 10.845994   2.521023   4.302 0.000198 ***
PF          -0.021648   0.003775  -5.735 4.27e-06 ***
PA           0.012814   0.004913   2.608 0.014660 *  
TO           0.145443   0.041115   3.537 0.001483 ** 
TA          -0.129806   0.048274  -2.689 0.012133 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.278 on 27 degrees of freedom
Multiple R-squared:  0.8554,	Adjusted R-squared:  0.834 
F-statistic: 39.93 on 4 and 27 DF,  p-value: 5.766e-11
```

With the *** indication in the P-values, one can see that the added variables are significant enough to be used in a model, all variables being at least under .05 P-value.

Now to predict 2019 data with the updated more accurate linear model.

5.) 2019 Actual wins vs Predicted total wins with Updated model

```
[1] 1.271271
```

- Meaning that on average the prediction was off by 1.27, .2+ more accurate than the original linear model.

6.) 2019 Actual losses vs Predicted total losses with Updated model

```
[1] 1.173916
```

- Meaning that on average the prediction was off by 1.17 losses, not bad, again an improvement from the original linear model by .3.

Now using it on the 2020 Season data that now includes takeaways and turnovers.

7.) 2020 Actual wins vs Predicted total wins with Updated model

```
[1] 1.507926
```

- Meaning this is less accurate than the original model by .01 with an RMSE of 1.5

8.) 2020 Actual losses vs Predicted total losses with Updated model

```
[1] 1.526682
```

- Meaning that the predicted wins are off by 1.52 wins, again an increase from the original linear model using just Points for and Points allowed.

Conclusion:

The overall conclusion from our data is that for the 2020 Season, using just Points allowed and Points for is more accurate at predicting total season wins and losses than adding turnover and takeaway variables to the linear model. For the 2020 NFL Season Data:
Original Model RMSEs: (W:1.49, L:1.45) vs Updated Model RMSEs: (W: 1.5,L: 1.52)

For the 2019 season, it is more accurate to use a linear model with Points allowed, Points for, Turnovers, and Takeaways in predicting the total wins and losses for the season. And that updating the model with the Turnover and Takeaway variable significantly improved the RMSE for predicting both wins and losses by nearly .3. For the 2019 Season Data:
Original Model RMSEs: (W:1.53, L:1.47) vs Updated Model RMSEs: (W: 1.27,L: 1.17)

The 2019 Linear Model with Points For and Points Against variable better predicted the Win/Loss totals for the 2020 data than the 2019 data, interesting because the model was trained and developed using the 2019 data but better predicted the 2020 season win/loss totals. However the updated model with Point For, Points Against, Turnovers, and Takeaways only significantly improved the predictions with the 2019 data but decreased the accuracy of predictions in the 2020 season, but by less than .1 RMSE.

For future considerations, getting more data and seeing if 2019 was an anomaly where turnover and takeaways factored more into wins/losses, or if usually, they do not contribute heavily towards overall wins/losses for the whole season.