Final Project

Jacob M. Lundeen

2021/05/03

Abstract

National Football League (NFL) teams traditional methods of player evaluation is outdated, especially at the quarterback position, which leads to teams overpaying for sub-par talent. Traditional methods include completion percentage, total passing yards, total passing touchdowns, and others. The objective of this project is to shed light on better ways for teams to evaluate quarterback play.

I will create an R Shiny application to show the correlations between modern metrics, completion percentage over expectation and expected points added, and wins. I will utilize multivariate linear regression to show which metric, or both, correlate and predict with more accuracy.

My results show that the metrics correlate highly with each other, so completion percentage over expectation was dropped from the model, and expected points added was shown to be statistically significant to Wins with an R^2 of 0.932.

In conclusion, the more modern metric of expected points added correlates well with Wins and is an accurate predictor.

Introduction

In American football, the Quarterback (QB) is king. In all of professional sports, it is the single most important position bar none. No other position in a team sport is as responsible for the success and failure of their team as the QB is. Because of this fact, a lot of research has started up over the past decade into this broad topic. Therefore, the focus of my project is to look at just how important the QB is to a team's success and create an R Shiny application to display that.

Background

Previous Work

For my project #2, I did some basic visualizations using a subset of the data set I am using for this project. In particular, I looked at how Expected Points Added (EPA) compares to Completion Percentage Over Expected (CPOE). To better understand these, we need some definitions. The two following definitions come from Advanced Football Analytics:

> **Expected Points (EP)** – The value of the current down, distance, and field position situation in terms of future expected net point advantage. In other words, it is the net point value a team can expect given a particular combination of down, distance, and field position. First and goal at the one represents an EP near 6, while 3rd and 20 at a team's own one yard line represents an EP of about -2. EP differs from Win Probability (WP) in that it does not take into account the game score and time remaining.("Glossary" 2015)
>
> **Expected Points Added (EPA)** – The difference between the Expected Points (EP) at the start of a play and the EP at the end of the play. EPA is the measure of a play's impact on the score of the game. An individual player's EPA is the sum of the EPA of the plays in which that player was directly involved. Being directly involved is defined as an offensive player who ran, threw, or kicked the ball, was targeted by a pass, or flagged for a penalty.("Glossary" 2015)

For CPOE, Josh Hermsmeyer said this:

> Completion percentage is the performance measurable that best translates from college to the NFL. The metric's shortcomings — players can pad their completion percentage with short, safe passes, for instance — are well-known. But even in its raw form, it's a useful predictive tool.
>
> Its kissing cousin in the pantheon of stats that translate from college to the pros is average depth of target: Passers who throw short (or deep) in college tend to continue that pattern in the NFL. These two metrics can be combined to create an expected completion percentage, which helps correct the deficiencies in raw completion percentage. If you give more credit to players who routinely complete deeper passes — and dock passers who dump off and check down more frequently — you can get a clearer picture of a player's true accuracy and decision-making.
>
> Another important adjustment is to account for the level of competition a player faced. ESPN's Total Quarterback Rating does this, and we're doing it, too. For instance, passes in the Big Ten are completed at a lower rate than in the Big 12 and the Pac-12. We should boost players from conferences where it is tougher to complete a pass and ding players whose numbers are generated in conferences where passing is easier.
>
> When we make these adjustments, and then subtract expected completion percentage from a QB's actual completion percentage, we get a new metric: completion percentage over expected, or CPOE. An example: In 2011 at Wisconsin, Russell Wilson had a raw completion percentage of 73 percent. We would expect an average QB in the same conference who attempted the same number of passes at the same depths that Wilson attempted to have a completion percentage of just 57 percent. So Wilson posted an incredible CPOE of +16 percentage points in his last year of college. CPOE translates slightly less to the NFL than either raw completion percentage or average depth of target, but it does a substantially better job of predicting on-field production. In stat nerd parlance, we've traded a little stability for increased relevance.(Hermsmeyer 2019)

Basically, CPOE is a measure of a QB's completion percentage compared to all other QB's in similar situations with strength of opponent taken into account. Josh Hermsmeyer's premise from that article is that NFL teams are looking at the wrong things when projecting a college QB's game to the next level. CPOE is less stable but overall more relevant because it is better at predicting on-field production.

Another leader in looking at QB efficiency is the company ProFootballFocus (PFF). In 2020, they posted an article looking at how a combination of their internal grades and EPA, combined with Bayesian Inference, helps them predict future production and rank the active QB's. In their research, they found, that for second-year QB's, the effect size in their model is up to 0.025 EPA per play (the difference between the seventh- and 13th-best passing offense in 2019).(Cole and Riske 2020) This helps illustrate that the higher a QB's EPA per play is, in general the more efficient they are.

Data

For my project, I will be using the data set created by Mr. Ben Baldwin called 'nflfastR' (www.nflfastr.com). It is a database containing NFL play-by-play data back to 1999. It includes the advanced metrics I discussed previously in EPA and CPOE. The data set contains 361 variables and, depending on how many seasons you're are looking at, hundreds of thousands of observations. For this project, I will be using data from the years 2011-2020 (well over 400,000 observations). Of the 361 variables, I will be using a very small subset, only 6 of the variables (one of which, 'Wins,' will be a mutation of the 'result' variable).

Approach

Extract, Transfer & Load

First, I needed to get the data into a usuable format. Once the 'nflfastR' package is loaded up, you can pull the data from it by X years (either single or multiple). From there, I had to distill it down into Name, EPA, CPOE, Attempts, Wins, and Number of Games played. I utilized the Tidyverse package to accomplish this: first by determining the number of wins each QB accumulated. There is a variable called 'result' that gives the difference in score between the home team and the away team from the home team's perspective. So a positive result is a win, a negative result is a loss, and a zero is a tie. The issue is that every observation is an individual play and each observation contains the same result value for a game. With some help from Mr. Ben Baldwin himself on his Discord server, I was able to get the number of wins by QB (a tie is counted as half a win).

With the wins calculated, it was a simple matter to calculate the CPOE and EPA of each QB. The two data frames were then combined and filtered for a minimum of 500 attempts to keep out backup QBs who might have much higher variance due to their low sample size. A print out of this code will be available in the Appendix.

Exploratory Data Analysis

Next is my EDA. Below is an overview of the final data set I used.

Table 1: Data summary

Name	results
Number of rows	74
Number of columns	6
Column type frequency:	
character	1
numeric	5
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
name	0	1	6	16	0	74	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Wins	0	1	35.86	29.29	1.00	13.12	29.00	48.62	139.00
Num_Games	0	1	69.59	45.59	15.00	37.25	54.50	86.50	182.00
CPOE	0	1	-0.40	2.54	-7.34	-1.87	-0.12	1.21	5.66
EPA	0	1	0.15	0.10	-0.15	0.07	0.16	0.22	0.39
Atts	0	1	2238.96	1672.98	509.00	1022.25	1520.50	3017.25	6842.00

So I am working with 6 variables and 74 observations, or QB's, that played between 2011 and 2020 and had at least 500 passing attempts. I know the data isn't normally distributed, but given this is just a cursory look at the relationships, I did not go in depth into investigating the distributions and if any transforms needed to happen.

R Shiny

With my data loaded and ready to use, I got to creating my application.

I didn't want the user to be able to change the axis variables, so I refrained from including any sort of input from the user. Instead I have six tabs: CPOE, EPA, Model, Residuals, Summary, and Table and a brief overview of what the application entails. Each tab is as follows:

- CPOE - This tab contains the plot of Wins vs. CPOE
- EPA - This tab contains the plot of Wins vs. EPA
- Model - This tab contains the printouts relevant to the models, including a correlation matrix
- Residuals - This tab contains the residual plots for both models
- Summary - This tab contains the descriptive statistics for the data
- Table - This tab contains a printout of the data frame

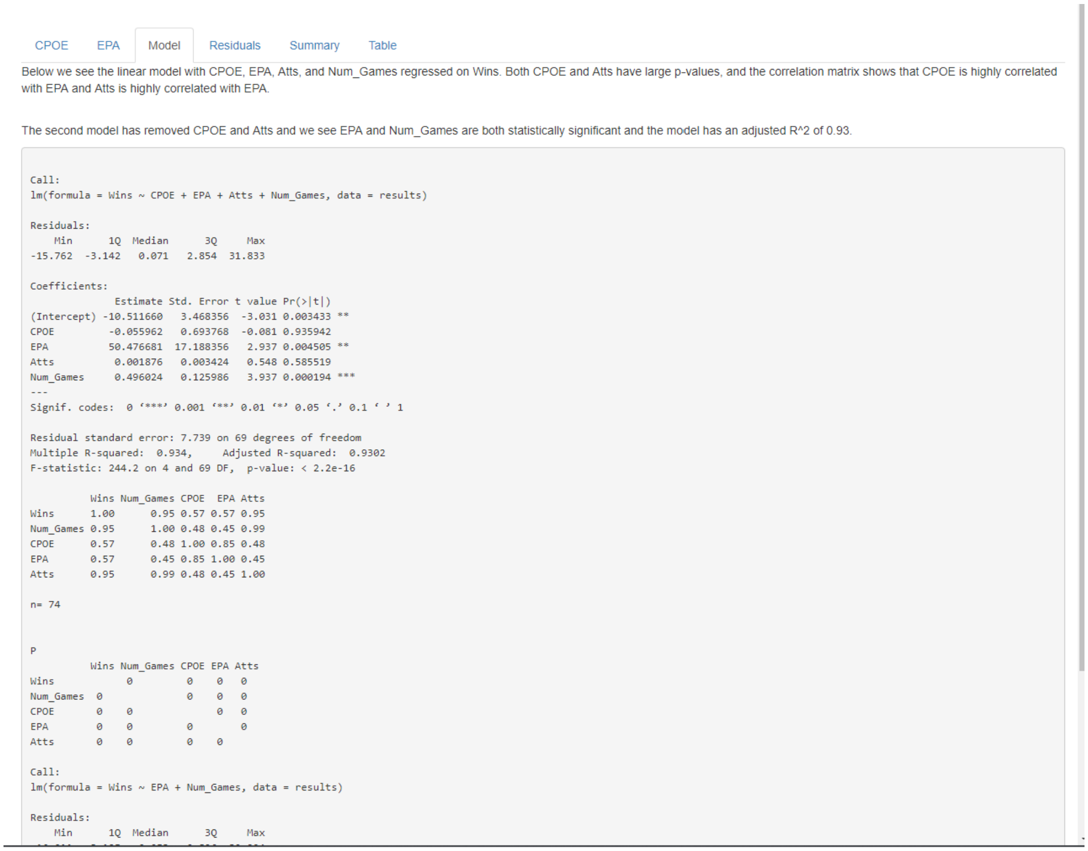
The description of the app on the left side is done in the sidebarPanel() using fluidrow() to handle the text. The image for the website nflfastR is included for flair. Each tab is created in the mainPanel() using tabPanel() and the text is output using p().

Both the CPOE and EPA tabs use plotly() with ggplot to create the tooltip that appears when the user hovers over the data points. The Model, Summary, and Table tabs uses renderPrint() and the Residuals tab uses renderPlot() since they are not interactive.

Results

Model

For the exploration of QB efficiency, I found there to be high correlation between CPOE and EPA and Attempts and Number of Games. I was not surprised by the latter, but I was a little surprised by the former. Neither metric is used in the calculation of the other, and CPOE is solely a measure of completion whereas EPA is a measure of everything the QB does, not just passing. With removing CPOE and Attempts, the R^2 improves slightly from 0.93 to 0.932, but all features are now statistically significant (see Figure 1 below).



Below we see the linear model with CPOE, EPA, Atts, and Num_Games regressed on Wins. Both CPOE and Atts have large p-values, and the correlation matrix shows that CPOE is highly correlated with EPA and Atts is highly correlated with EPA.

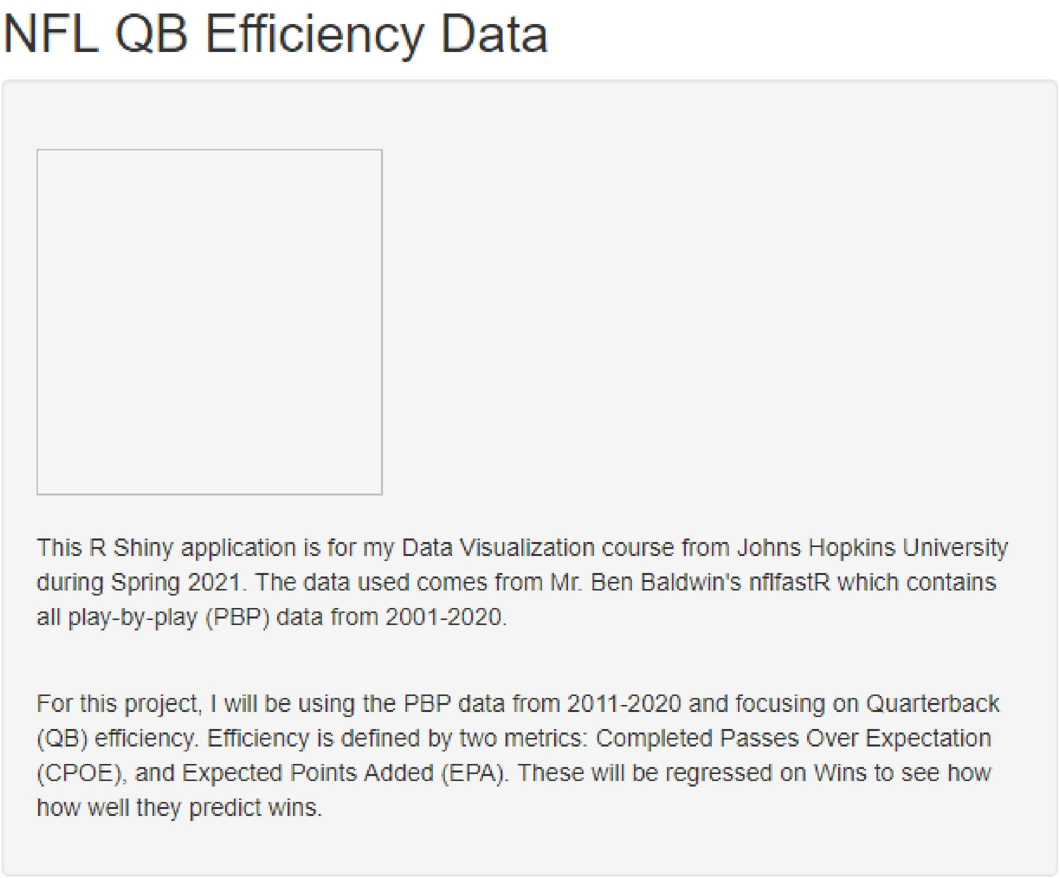
The second model has removed CPOE and Atts and we see EPA and Num_Games are both statistically significant and the model has an adjusted R^2 of 0.93.

```

Call:
lm(formula = Wins ~ CPOE + EPA + Atts + Num_Games, data = results)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.762  -3.142   0.071   2.854  31.833 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -10.511660   3.468356  -3.031 0.003433 ** 
CPOE         -0.055962   0.693768  -0.081 0.935942    
EPA          50.476681  17.188356   2.937 0.004505 ** 
Atts          0.001876   0.003424   0.548 0.585519    
Num_Games     0.496024   0.125986   3.937 0.000194 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.739 on 69 degrees of freedom
Multiple R-squared:  0.934,	Adjusted R-squared:  0.9302 
F-statistic: 244.2 on 4 and 69 DF,  p-value: < 2.2e-16

          Wins Num_Games CPOE  EPA Atts
Wins      1.00      0.95 0.57 0.57 0.95
Num_Games 0.95      1.00 0.48 0.45 0.99
CPOE      0.57      0.48 1.00 0.85 0.48
EPA       0.57      0.45 0.85 1.00 0.45
Atts      0.95      0.99 0.48 0.45 1.00

n= 74 


P
          Wins Num_Games CPOE EPA Atts
Wins            0         0    0   0  
Num_Games  0              0    0   0  
CPOE       0    0              0   0  
EPA        0    0         0        0  
Atts       0    0         0    0      

Call:
lm(formula = Wins ~ EPA + Num_Games, data = results)

Residuals:
    Min      1Q  Median      3Q     Max 
```

Figure 1: Model Tab

R Shiny

The application overall came together quite nicely. My application properly displays multiple plots comparing the correlation of CPOE and EPA to Wins, builds a simple linear model to show the correlations, plots the residuals, and then prints out the descriptive statistics and the data frame itself. As seen below in Figures 02 - 07:

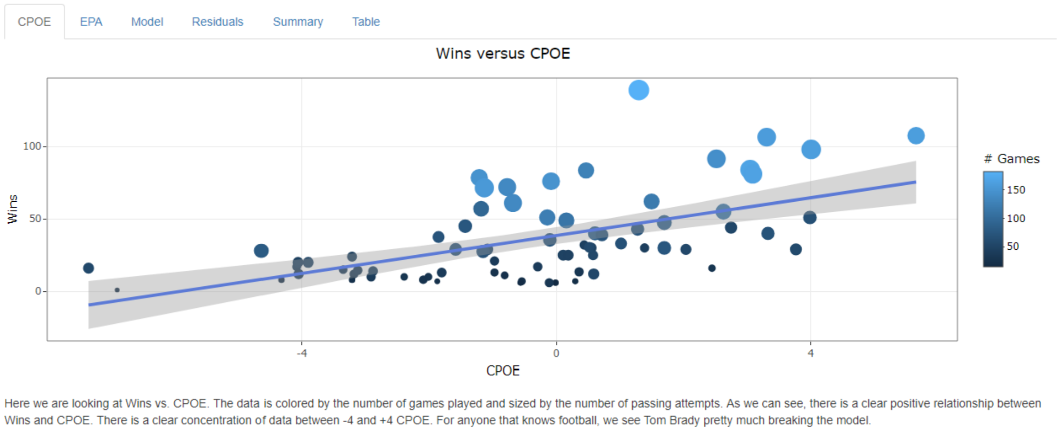


NFL QB Efficiency Data

This R Shiny application is for my Data Visualization course from Johns Hopkins University during Spring 2021. The data used comes from Mr. Ben Baldwin's nflfastR which contains all play-by-play (PBP) data from 2001-2020.

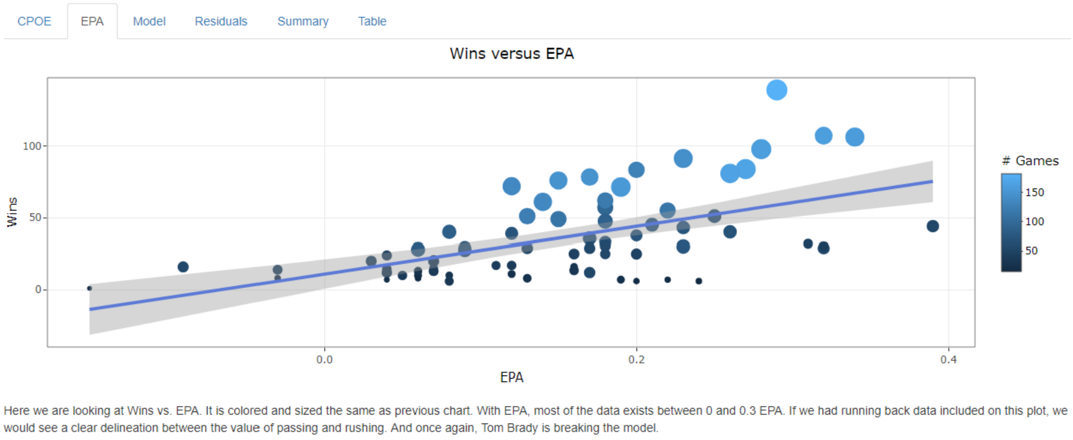
For this project, I will be using the PBP data from 2011-2020 and focusing on Quarterback (QB) efficiency. Efficiency is defined by two metrics: Completed Passes Over Expectation (CPOE), and Expected Points Added (EPA). These will be regressed on Wins to see how how well they predict wins.

Figure 2: Introduction



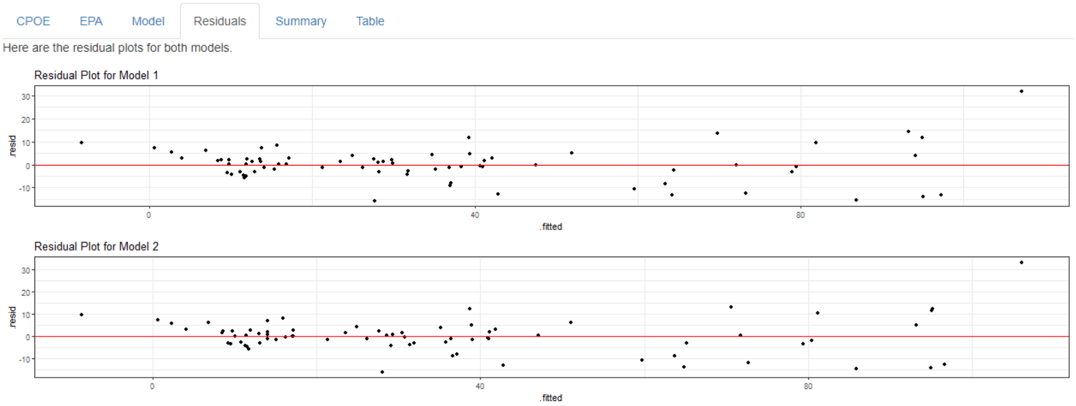
Here we are looking at Wins vs. CPOE. The data is colored by the number of games played and sized by the number of passing attempts. As we can see, there is a clear positive relationship between Wins and CPOE. There is a clear concentration of data between -4 and +4 CPOE. For anyone that knows football, we see Tom Brady pretty much breaking the model.

Figure 3: Wins vs. CPOE



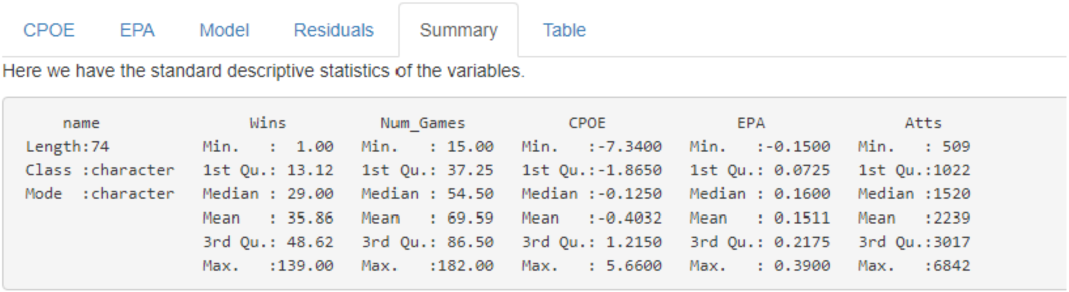
Here we are looking at Wins vs. EPA. It is colored and sized the same as previous chart. With EPA, most of the data exists between 0 and 0.3 EPA. If we had running back data included on this plot, we would see a clear delineation between the value of passing and rushing. And once again, Tom Brady is breaking the model.

Figure 4: Wins vs. EPA



Here are the residual plots for both models.

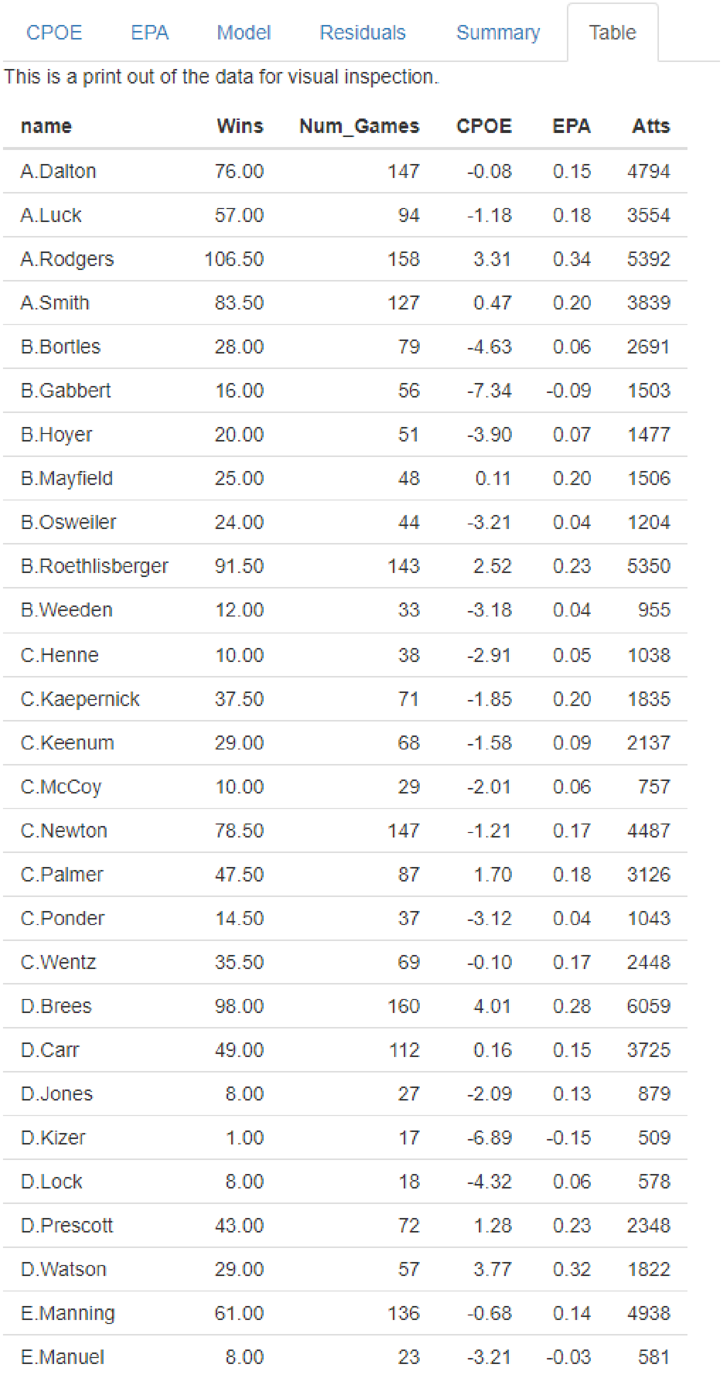
Figure 5: Residual Plots



Here we have the standard descriptive statistics of the variables.

```
     name                Wins          Num_Games          CPOE              EPA              Atts     
 Length:74          Min.   :  1.00   Min.   : 15.00   Min.   :-7.3400   Min.   :-0.1500   Min.   : 509  
 Class :character   1st Qu.: 13.12   1st Qu.: 37.25   1st Qu.:-1.8650   1st Qu.: 0.0725   1st Qu.:1022  
 Mode  :character   Median : 29.00   Median : 54.50   Median :-0.1250   Median : 0.1600   Median :1520  
                    Mean   : 35.86   Mean   : 69.59   Mean   :-0.4032   Mean   : 0.1511   Mean   :2239  
                    3rd Qu.: 48.62   3rd Qu.: 86.50   3rd Qu.: 1.2150   3rd Qu.: 0.2175   3rd Qu.:3017  
                    Max.   :139.00   Max.   :182.00   Max.   : 5.6600   Max.   : 0.3900   Max.   :6842  
```

Figure 6: Summary Statistics



This is a print out of the data for visual inspection.

name	Wins	Num_Games	CPOE	EPA	Atts
A.Dalton	76.00	147	-0.08	0.15	4794
A.Luck	57.00	94	-1.18	0.18	3554
A.Rodgers	106.50	158	3.31	0.34	5392
A.Smith	83.50	127	0.47	0.20	3839
B.Bortles	28.00	79	-4.63	0.06	2691
B.Gabbert	16.00	56	-7.34	-0.09	1503
B.Hoyer	20.00	51	-3.90	0.07	1477
B.Mayfield	25.00	48	0.11	0.20	1506
B.Osweiler	24.00	44	-3.21	0.04	1204
B.Roethlisberger	91.50	143	2.52	0.23	5350
B.Weeden	12.00	33	-3.18	0.04	955
C.Henne	10.00	38	-2.91	0.05	1038
C.Kaepernick	37.50	71	-1.85	0.20	1835
C.Keenum	29.00	68	-1.58	0.09	2137
C.McCoy	10.00	29	-2.01	0.06	757
C.Newton	78.50	147	-1.21	0.17	4487
C.Palmer	47.50	87	1.70	0.18	3126
C.Ponder	14.50	37	-3.12	0.04	1043
C.Wentz	35.50	69	-0.10	0.17	2448
D.Brees	98.00	160	4.01	0.28	6059
D.Carr	49.00	112	0.16	0.15	3725
D.Jones	8.00	27	-2.09	0.13	879
D.Kizer	1.00	17	-6.89	-0.15	509
D.Lock	8.00	18	-4.32	0.06	578
D.Prescott	43.00	72	1.28	0.23	2348
D.Watson	29.00	57	3.77	0.32	1822
E.Manning	61.00	136	-0.68	0.14	4938
E.Manuel	8.00	23	-3.21	-0.03	581

Figure 7: Table

Issues / Future Work

The primary issue I ran up against was getting the tooltip function to work with selectable data. As the charts currently stand, when you move the cursor over a data point on the CPOE and EPA plots, a tooltip pops up that displays the ‘X’ and ‘Y’ data, plus some additional data made possible by plotly. I wanted this functionality along with allowing the user to select the data to display on the ‘X’ and ‘Y’ axes, but I couldn’t get that to work. I chose to go with the working tooltip since I was able to get the changeable axes to work in Project #2.

My secondary issue is data storage. As of right now, the application has to pull the PBP data every time I run the application. I’d like to find a way to store that data somewhere so it will not have to do that every time and will run much faster.

For future work, I’d like to do go much more in depth with the aesthetics of the application. There are apparently pre-designed layouts and color schemes that would be fun to use.

Conclusion

In conclusion, I had a lot of fun making this application. Shiny has a lot of potential on making interactive dashboards and I plan on exploring those in the future. Next would be to publish my application and see it work from the web.

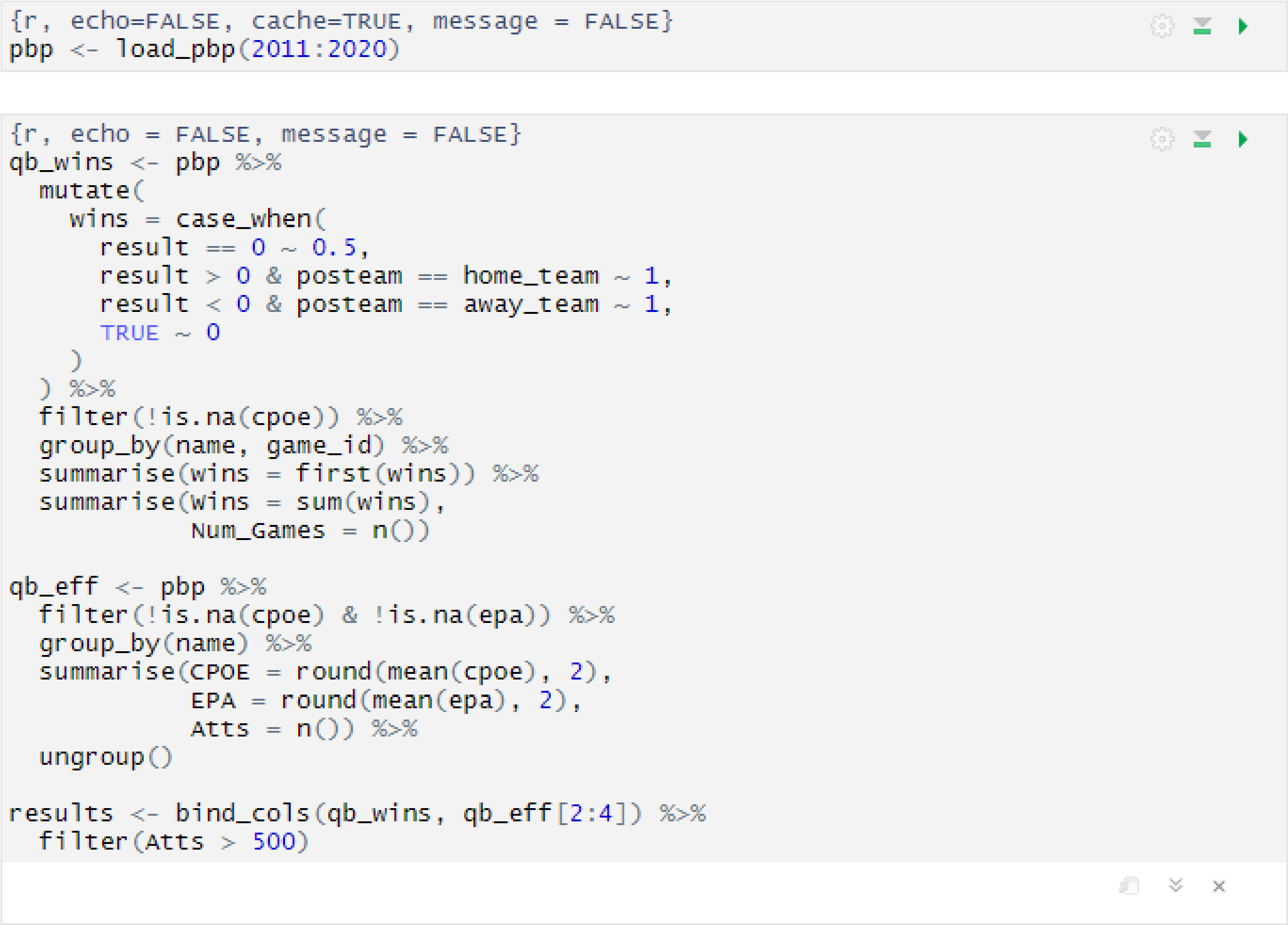
References

Cole, Kevin, and Timo Riske. 2020. "2020 NFL QB Rankings: Using Bayesian Updating to Rank All 32 Projected Starters." https://www.pff.com/news/nfl-2020-quarterback-rankings.

"Glossary." 2015. https://www.advancedfootballanalytics.com/index.php/home/stats/stats-explained/glossary#:~:text=Expected%20Points%20Added%20(EPA)%20%E2%80%93,that%20player%20was%20directly%20involved.

Hermsmeyer, Josh. 2019. "The NFL Is Drafting Quarterbacks All Wrong." https://fivethirtyeight.com/features/the-nfl-is-drafting-quarterbacks-all-wrong/.

Appendix



```
{r, echo=FALSE, cache=TRUE, message = FALSE}
pbp <- load_pbp(2011:2020)
```

```
{r, echo = FALSE, message = FALSE}
qb_wins <- pbp %>%
  mutate(
    wins = case_when(
      result == 0 ~ 0.5,
      result > 0 & posteam == home_team ~ 1,
      result < 0 & posteam == away_team ~ 1,
      TRUE ~ 0
    )
  ) %>%
  filter(!is.na(cpoe)) %>%
  group_by(name, game_id) %>%
  summarise(wins = first(wins)) %>%
  summarise(wins = sum(wins),
            Num_Games = n())

qb_eff <- pbp %>%
  filter(!is.na(cpoe) & !is.na(epa)) %>%
  group_by(name) %>%
  summarise(CPOE = round(mean(cpoe), 2),
            EPA = round(mean(epa), 2),
            Atts = n()) %>%
  ungroup()

results <- bind_cols(qb_wins, qb_eff[2:4]) %>%
  filter(Atts > 500)
```

Figure 8: Tidy Code for ETL