Project

JARED YU

Introduction

The choice of dataset for the final project is Table B.1 from the Textbook. The table is titled "National Football League 1976 Team Performance." It contains data on various American football teams that art part of the National Football League (NFL). The target (a.k.a. response) variable is related to the number of games won in a 14-game season. The attributes of the dataset are related to various statistics for each of the 28 teams listed in the table. A reason for choosing this dataset over the other option of Table B.12 is that it has both more rows and fewer columns. This could make modeling more straightforward. However, the more noteworthy reason is the fact that this dataset has fairly easy to understand variables. American football is a sport that I understand on a decent level, and so that makes doing analysis on this data is both more exciting and more interpretable. There is the option also of choosing a dataset from some online source, however, the pattern thus far throughout the class has been to work with these table-sized datasets rather than the typical ones found online with hundreds or thousands if not more rows of data. Often times, these datasets will also have a mixture of categorical variables and potentially missing values. For these reasons, the choice will be to stick with Table B.1.

After having selected a dataset, the next steps are to utilize the flowchart seen in Figure 1.8 of the Textbook. This will allow for a more straightforward modeling process. This figure combines together data and theory, model specification, parameter estimation, model adequacy checking, model validation, and model use. It also explains how an analyst can go from one step to the next in a cyclical fashion until he or she finds a suitable model. It is emphasized that this is not necessarily a robotic process, instead, the analyst must think creatively and make decisions based on experience.

Exploration

The first stage of analyzing the dataset will revolve around trying to understand its variables. The dimensions of the dataset are such that there are 28 rows and 10 columns. Of these columns, they are labeled $y, x_1, x_2, \cdots, x_9$. The y column corresponds with the target variable, whereas before it was mentioned that this is the number of games won by an NFL football team in 1976 out of a total of 14 games played. The other $x_j, j = 1, \cdots, 9$ correspond to the following statistics: Rushing yards (season), Passing yards (season), Punting average (yards/punt), Field Goal Percentage (FGs made/FGs attempted), Turnover differential (turnovers acquired - turnovers lost), Penalty yards (season), Percent rushing (rushing plays/total plays), Opponents' rushing yards (season), and Opponents' passing yards (season).

Most of these variables are pretty straightforward to anyone with a basic understanding of football. The only one that was of confusion is turnover differential. Doing a quick Google search yields the following, "Turnover Differential is calculated by subtracting the total number of giveaways (interceptions & fumbles lost) from the total number of takeaways (interceptions & opponent fumble recoveries) [1]." Therefore, the understanding that a high value of this in general should lead to a higher response value in y.

Something else worth noting is that all of the attributes, $x_j, j = 1, \cdots, 9$, are numeric variables. This can help to simplify the process, since it will not be necessary to handle categorical variables through dummy variables. There is a mixture of both discrete and continuous numeric variables amongst the attributes. The response variable itself is within the

range of 0 to 13, showing how one team won zero games (Tampa Bay) and how one team lost only one game (Oakland).

A simple step to better understand the dataset is to plot all of them individually. Figure 1 in the Appendix shows the density histogram of each variable, where a smoothed density curve is included to help make the plots easier to understand. Looking at these plots, it is apparent that many of them do not have the bell-curved shape of the normal distribution, including the response y. Looking closer at y, it has a noticeable bimodal distribution, where there is a gap in the middle of the range of values. Looking at x_1, it seems fairly bell-shaped. This seems to indicate that the total number of rushing yards is possibly close to a normal distribution across all the teams. The x_2 plot seems to show instead that there is an almost uniform distribution to the data. Looking closely, it seems that most teams seem to be in the low to mid-range of passing yards, with a spike in the upper end for other teams. Looking at x_3, it shows almost a bell-shape, but there is a noticeable peak on the left-tail. Looking at x_4, the shape is also roughly normal. However, it is apparent that there is only a small density towards the left-tail, with the other teams falling close to the middle or high range. The x_5 variable shows a balance around the center of zero. This makes sense, since the variable is calculated by subtracting negative turnovers from positive turnovers. Furthermore, if a team gains a turnover, then another team gets a (negative) turnover. Looking at x_6 it shows a roughly bell-shaped curve, with a peak around the 700 range. The x_7 plot seems to show a concentration in the mid and high ranges of the data. The x_8 variable shows a concentration below the mid-range, while being imbalanced around the middle. The x_9 variable shows a large concentration around a center, which is unlike what is seen in other plots.

Something important about the numeric attributes is how they are related to each other. To understand this visually, a pairs matrix can be done, which shows the pair-wise scatterplot between all the variables in the dataset. This can be seen in Figure 2 of the Appendix. This plot is symmetric around the diagonal, and so it is only necessary to look at either the upper or lower triangle. Here, the focus will be on the upper triangle. Looking first at the relationship between y and the other attributes, it is evident that there is some linear relationship with many of them. Some standout examples are x_1, x_7, and x_8. The concern however is primarily among the attributes themselves, which can result in multicollinearity. This can be thought of as correlation amongst the attributes, which can possibly be identified visually by looking for linear relationships between two attributes. Looking at the same three x_1, x_7, and x_8, it is evident that the pairwise relationships between these variables seem to themselves show some sort of linearity. A noticeable problem with the pairs matrix is that there are a relatively few number of points in each plot. This makes it more difficult to identify obvious linear patterns via some straight line.

An alternative to the pairs matrix for understanding multicollinearity is to simply find the correlation matrix for all the variables. This can be seen in Table 1 of the Appendix. Like the pairs matrix, it is symmetric about the diagonal and so for simplicity only the upper triangle is focused on. The correlation between variables is shown, which ranges from 0 to 1. Large values indicate a high level of correlation, while low levels indicate low correlation. Ideally, we want to see low correlation between the attributes. The first row of this table shows the correlation between y and the x_j. It is not necessarily bad that these values are high, and if they are high then

that could possibly mean that they will be effective in trying to model y. Some color coding has been done, where values larger than 0.7 (and below 0.9) are colored red and those between 0.5 and 0.7 are colored blue. Red indicates that there is high correlation and blue indicates moderate correlation [2]. Amongst the attributes, there is high correlation between x_1 and x_7. This indicates that teams with that have many plays where they rush the ball also often many rushing yards. This is quite easy to see, but it can be problematic for a model when one variable is a simple function of the other. The other pairs of moderate correlation include: (x_1, x_5), (x_1, x_8), (x_5, x_7), and (x_7, x_8). The correlation between x_1 and x_5 shows that teams with many rushing yards also possibly have a good turnover rate. The correlation between x_1 and x_8 shows that teams with many rushing yards have a low number of opponents' rushing yards. Looking back at Figure 2, we can see that this is a negative (correlation) relationship. The correlation between x_5 and x_7 indicates that teams with a higher turnover differential tend to spend more plays rushing. The correlation between x_7 and x_8 indicate that when teams spend more time rushing their opponents tend to gain less yards rushing.

Another way to understand multicollinearity is to look at the variance inflation factor (VIF) for different attributes. The general rule of thumb for VIFs is that when the VIF for an attribute is above 5, that is a bad sign. When it is higher than 10, then that is an even worse sign. The can VIF help to indicate that multicollinearity exists in the data when more than one of them have large VIF values. The VIF values can be seen in Table 2 of the Appendix. Only one of them, x_7, exceed 5. However, two of them, x_1 and x_8, are relatively close to 5, each being larger than 4. These are the same three attributes pointed out in the pairs matrix for being more obviously related. These three variables are all related to the concept of rushing within the game of football. Therefore, it makes sense that there is concern over the multicollinearity of these variables. It would perhaps make sense to remove at least one of them in the final model, given the issue of correlation amongst them.

Modeling

There are different possible approaches to modeling such a dataset. However, the initial model can simply be the fitted multiple linear regression model, regression y on the x_j, $j = 1, \cdots, 9$. After fitting this model, the first step can be to perform the test for the significance of regression. This test checks the hypothesis $H_0: \beta_j = 0$ vs. $H_1:$ at least one β_j is nonzero. The result of this test yields test statistic $F_0 = 8.8458$ with a corresponding p-value of 5.303×10^{-5}, so we can reject the null at the 99% confidence level. From here, it is evident that there is potential for modeling the response variable based on the given predictors. An issue arises however when performing the test on the individual regression coefficients. The hypothesis test is as follows, $H_0: \beta_j = 0$ vs. $H_1: \beta_j \neq 0$ for each $j = 1, \cdots, 9$. Only x_2 has a p-value less than 0.05. The variable x_8 has a p-value below 0.1. However, the p-value for every other variable is over 0.2. The p-values for the test on the individual regression coefficients can be seen in Table 3 of the Appendix. This seems to indicate that removing almost any other variable from the current model, we can call it "model_1," leaves little difference in the result. It is essentially being compared to the intercept-only model, which will be denoted "model_0."

From here, there are some possible choices. It is possible to try some sort of transformation on the response variable, through Box-Cox. Another possibility is to add

quadratic terms to the model so that it becomes polynomial. This includes squared terms and interaction terms. A more realistic option is to perform variable selection through forward selection, backward elimination, stepwise regression, and all possible regression. The reason why these options sound like good ideas is that apparently from model_1, there are many possibly unhelpful terms in the model and so the model size can be dramatically reduced. This could also potentially help to reduce bias in the model by eliminating extra terms.

An issue with transforming the response variable is that the interpretation no longer holds. Instead, we would be looking at the impact on for example the natural log of y, $\ln y$, which is the natural log the number of games won. This is not an intuitive measure and inverting the transformation would lead to the model looking at the median response rather than the mean response value, which is not as desirable in this case.

However, with modern software and computers, it is fairly easy to at least quickly check the results from the previously mentioned ideas. The "check" here will be to see if the hypotheses tests on the individual regression coefficients yield more ideal results, where it is not keeping just one or two regressor variables that have any statistical significance to them. A simple transformation method is Box-Cox. An issue with using Box-Cox on this dataset however is that one of the response variables has a value of 0. The "boxcox()" function in R that is part of the "MASS" library requires positive values for the response. Therefore, to fit the initial Box-Cox the step was to ignore this one observation where $y = 0$. The first call of boxcox() on the dataset can be seen in Figure 3 (a) of the Appendix. The graph indicates that the dataset could possibly benefit from a power transformation, since the vertical center dotted line is not directly above 1. The first transformation to try however, is the natural log transformation, where the result of calling boxcox() again on the log transformed response, $\ln y$, can be seen in Figure 3 (b). Trying this transformation yields poor results, as the center dotted line is no longer visible, since ideally the peak of the curve will be over 1. In Figure 3 (c), it shows the result of using the power transformation on y, where the result is $y^{\frac{3}{4}}$. In this plot, it is evident that the center dotted line is much more directly above 1. Using the $y^{\frac{3}{4}}$ transformation, a new model called "model_2" was fit. Again, the test for the significance of regression was done, where the result is a test statistic of $F_0 = 8.4087$ and a corresponding p-value of 7.459×10^{-5}. The results however for the hypotheses tests on the individual regression coefficients were not too different from what was seen in model_1. The p-values can be seen below in Table 4 of the Appendix. Once again, only x_2 has a p-value below 0.05, and x_8 has a p-value below 0.1.

Now, the approach will be to try and use variable selection. As mentioned, there are three methods to be attempted. These methods are forward selection, backward elimination, stepwise regression, and all possible regression. We can call the forward selection model "forward_1," the backward elimination model "backward_1," the stepwise regression model "step_1," and the all possible regression model "best_1." The results of forward selection, backward elimination, and stepwise regression can be discussed more briefly. The methods work by using an algorithm to either build a model from an intercept-only model, take it apart from a full model using all terms, or use some combination of the two. In these cases, the algorithm will run on its own and there is no need to make decisions. In all three cases, forward_1, backward_1, and step_1 happen to agree on the same choice of a model. The resulting model chosen is,

$$y = \beta_0 + \beta_1 x_2 + \beta_2 x_7 + \beta_3 x_8 + \beta_4 x_9.$$

The results for each of these algorithms can be seen in Table 5 (forward selection and stepwise regression), Table 6 (backward elimination) in the Appendix (*Note: The algorithms' steps in forward selection and stepwise regression are identical so their results can be seen in the same Table 5.*). It can be seen that in all models, the goal is for the algorithm to minimize the AIC criterion by choosing the subset of variables that lead to the optimal AIC.

The results of using all possible regression is slightly more complicated. The function in R is called "regsubsets()" and is part of the "leaps" library. The function will first find the optimal subset of regressors for a model with 1, 2, …, 9 regressors in the model (here 9 is associated with the full model utilizing all regressors). This method is not problematic in this dataset with such a limited number of variables. It is problematic only when there are many variables and so the program can take quite a while to run. Each of these models are judged based on the residual sums of squares for each subset of models. The criterion that are used to judge the chosen subsets in this case are: R^2_{Adj}, MS_{Res}, C_p, AIC, and BIC. The results can be seen in Table 7 of the Appendix. It can be seen that C_p and BIC both agree on the model of size three, while the others agree on the model of size four. The former consists of $y \sim x_2 + x_7 + x_8$, while the latter consists of $y \sim x_2 + x_7 + x_8 + x_9$. So, it can be seen that the majority of the criteria agree on the same model as chosen by forward selection, backward elimination, and stepwise regression. It is interesting to note however that two criteria do choose an alternative model. This model does exclude x_9, which does not have as serious of a multicollinearity problem. It does have a slight correlation however with x_8, another regressor in the model. This variable is related to the opponent's passing yards and could possibly be indicative of the defense for a team. Perhaps it is the case that the defense for most teams is roughly similar, and so it is not too useful in prediction. A possibility is however that there are certain teams with extremely good (or poor) defense and this helps to indicate their winning (or losing) rate. In either case, we can try to explore both to see their potential. We can call the model of size three "best_1a" and the model of size four "best_1b."

So, from the variable selection methods, we are left with two models, best_1a and best_1b. We can analyze both of these again side-by-side. Using the hypothesis to test the significance of regression, we can see that both models still reject the null that $\beta_j = 0$ for all $j \neq 0$. The best_1a model gives the test statistic $F_0 = 29.437$ with a corresponding p-value of 3.273×10^{-8}, while the best_1b model gives the test statistic $F_0 = 23.172$ with a corresponding p-value of 8.735×10^{-8}. So, both models are significant up to the 99% confidence level. Now, going back to the hypothesis tests for the significance of individual regression coefficients, we can see the difference now between best_1a, best_1b, and model_1. As a recap, model_1 had issues when examining the significance of individual coefficients. The results of the same test can be seen in Table 8 shows the p-values for both best_1a and best_1b. The difference this time is that nearly all the regressors listed in the model are shown to be significant up to the 95% confidence level. It is noteworthy to see that in both models, the intercept is seen to not be useful. The difference here is that in best_1b, the regressor x_9 is also shown not to be useful, since it has a corresponding p-value of 0.2074. However, for both models, x_2, x_7, and x_8 are shown to be nonzero up to the 95% confidence level. Since the only difference between these two models is x_9, the choice will then be to drop this regressor and go with best_1a. Usually, the

intercept is kept in a model, so it will not be dropped despite having a high p-value. The coefficients for best_1a can be seen below in Table 9 of the Appendix.

Residuals

Having selected a model, we can do model adequacy checking with some residual analysis. When doing regression analysis, it is important to check that some assumptions of the linear regression model are being met. There are five main assumptions. The first is that the relationship between y and the regressors is linear. The second through fifth are about the error term. It should have zero mean, constant variance, uncorrelated, and have be normally distributed. Importantly, the hypotheses tests performed earlier are possibly not accurate if the errors are not normal. Through residual analysis, we can try to check how well our model conforms to these main assumptions.

We can understand the residuals as simply the difference between the true y values and their predicted values from model best_1a. The residuals and their scaled values in Table 10 of the Appendix. This table includes the following types of residuals: residual (unscaled residuals), standardized residuals, studentized residuals, R-Student residuals, and PRESS residuals. Looking at the residuals, it seems many of them are within a narrow range around zero. However, three of them have an absolute value greater than 2.5 and they have been highlighted in yellow. It does not necessarily make sense to call them outliers. The reason is that there are so few observations, and all the data is most likely carefully evaluated and measured correctly. Given that this is data from the NFL, it is not too likely that this data is incorrect or contains noticeable errors. The highlighted rows are 1, 3, and 21 from the dataset. These rows correspond with Washington, New England, and New York Giants respectively. The first two are in the upper range of the number of games won, while the last is in the lower range. However, none of them are the minimum or maximum number of games won.

A basic initial plot of the residuals is the normal probability plot. This plot shows the residuals sorted and plotted against their cumulative probability. Ideally, we want to see the points lie fairly close to the straight line. However, it is evident that there is possibly some heavy-tailed distribution to the residuals on the right-hand side. The upper points seem to have the pattern of lying below the line. In Figure 5 of the Appendix, it shows the plot of the different residuals against the fitted values. Amongst the four scaled (and unscaled) residuals, they all remain relatively similar in appearance, so no distinction seems to be needed amongst them. They all show a "good" pattern, where the points all fall within some horizontal band. That is, there is no alternate funnel, circular, or bowed shape for example. This is a good sign and shows that perhaps no further transformation is required on the regressors.

Furthermore, we can also look at the residuals plotted against the regressors. As with the Textbook, the type of residuals used will be the R-Student residuals, or the externally rotated studentized residuals. The plots can be seen in Figure 6 of the Appendix. In the top-left, for the plot related to x_2, it seems that this plot shows a fairly balanced distribution of points which is ideal. However, looking at x_7 in the top-right it shows a much different behavior. It seems instead that there is some fan-out, which indicates some increasing variance. However, much of this seems to be associated to a single observation on the far left, which makes the analysis less ideal. Ignoring this observation, there still seems to be a type of fanning out pattern in the data.

Looking at the bottom-left is the plot for x_7. Here, there is a different, seemingly double-bow shape to the points. This is also not too ideal. It is worth noting however, that these patterns exist in relationship to the residuals and the regressors instead of the residuals and the predicted values. The latter case may lead to a more urgent reason for performing further transformations. However, a drawback of transformations is that they distort the variables and lead to a more difficult interpretation.

Another idea is to examine the response against each of the regressors in a scatterplot. This can reveal if there are for example any influential or leverage points. The plots can be seen in Figure 7 of the Appendix. Looking at the top-left shows y against x_2. From here, the points seem fairly evenly distributed and none seem to be sticking out. Looking at the top-right is y against x_7. It is evident that there is a single point which is an influential point in the bottom left of the plot. It is also identified via a red cross over the point. This point corresponds with the penalty yards for the Seattle football team. It seems that in particular, this team is unique in having few penalty yards in that season. The plot of y against x_8 can be seen in the bottom-right. In this plot, it seems that the points are fairly evenly distributed along a diagonal pattern with no noticeable leverage or influential points. It seems then that overall, for the most part these regressors are balanced in terms of the distribution of points relative to y. In other words, there are not many noticeable influential or leverage points, with the exception of one observation in x_7. This is a good sign, as these influential and leverage points can skew the estimated values of the coefficients.

Model Validation

Model validation is slightly different from model adequacy checking (done previously). Here, the goal is to see if it will perform well for its intended purposes. A possible purpose here is to do prediction, for example if it will predict well the future number of games won in the next NFL season. Another possibility is simply for better understanding the relationship between the response and the regressors. Here, there is not necessarily data available for the next season. Also, there are limited samples, so any test split will have questionable results. Also, understanding football or sports in general will show that different teams (observations) will have varying characteristics based on the teams' skill that season. Also, it is well understood that the performance of teams varies over seasons, so it must be accounted for that next season could have major changes in team performance with reasons independent of the model itself.

Performing validation can involve analyzing the model based on theory about the data or even something like simulation. However, parts of this is not something feasible for the course project, as such data and knowledge is not easily available. It is also possible to try and perhaps collect more data through online resources to for example find out the data for the next few seasons in NFL. Another option is to perform data splitting to see if the predictive performance does well. However, as mentioned there are few samples and so splitting the data will not be simple.

A first simple step is to reanalyze the multicollinearity amongst the regressors in best_1a. There was some initial concern, due to the conceptual relationship between x_7 and x_8. They are both related to the concept of "rushing" in football. To do this, the VIFs can be analyzed again for the new model. The VIFs can be seen in Table 11 of the Appendix. The results are good for

the model, since all the values are less than 5. This is indicative that there are no more multicollinearity issues with the model.

By building on the previous residuals generated from the PRESS residuals, we can utilize data splitting. In calculating these residuals, we ignore one of the points at a time. From these residuals, we can take their sums of squares to get the PRESS statistic. Using the PRESS statistic, we can then calculate $R^2_{Prediction} = 1 - \frac{PRESS}{SS_T}$, which is another form of the R^2 statistics. The resulting value is $R^2_{Prediction} = 0.7325$, which is not the best. Preferably, we would like a value over 0.9, but this value is not too bad either. This would indicate that the model is good at predicting new observations. However, at the same time, using only three regressors it seems unrealistic to think that the model would perform too well. A quick test was done afterwards to try possible a polynomial model based on this reduced best_1a model. However, there was a similar issue of the linear dependence between the columns and so the result is many NA's. Therefore, this model was not approached.

The project instructions do however say that we should test the model's ability to perform prediction, so this will be done despite the limited size of the dataset. We can further build on the idea of prediction by doing some alternative data splitting. For example, we can fit the chosen model on half the data and test on the other half of the data. However, in regression the loss function is simply the sum of squares of the difference between the predicted and true values. Therefore, looking at the loss for a single model in isolation is meaningless since there is no other score to compare it to. So, for this example, we will include two other models. Let "model_base" be the intercept-only model and "model_full" be the full model including all regressors. The loss for the model_base, model_full, and model_3 are 185, 295.5793, and 62.4047 respectively. Clearly, the reduced model outperforms the other models by a significant amount. Surprisingly, the full model even underperforms against the intercept-only model which simply looks at the average of y. It seems then that this reduced model, model_3, is comparatively better than the base and full model versions. The likely reason for the underperformance of the full model is that it has too much bias and incorporates too much information that is not useful for prediction. The base model contains too much variance and is unable to accurately predict for new observations. The reduced model, model_3, seems to be able to be slim enough to predict well new observations without having too much bias or variance. It is worth noting too though that the test observations are for the "current" 1976 NFL season. It would be more logical to try and predict on the next season of statistics, given that they are available.

Discussion

The resulting model seems decent given the dataset being the size it is. I think that in general outside of textbooks we are used to thinking about "big data," or at least data that is comparatively larger. This dataset seems more akin to the famous Iris dataset with its limited size. These are good for learning to implement basic statistical models like linear or logistic regression but trying to extract too many insights or expect grand results is probably unrealistic for the most part.

We can further do some analysis of this final model, best_1a. Table 12 in the Appendix shows the 95% confidence intervals for each of the coefficients in the best_1a model. Looking closely, it is apparent that many of the values are quite close to zero, with the exception of the intercept term. This can be thought of in different ways. For one, the response variable itself only lies between 0 and 13. Therefore, it wouldn't make sense to have too large of coefficient values (we can standardize the coefficients, which will be discussed later). Table 13 in the Appendix shows the range (defined as the min and max values) of each regressor in the model. Looking at these, it is apparent that they are relatively large compared to the response variable's range. Therefore, trying to get a good interpretation of the model in the current state can be problematic.

Alternatively, we can scale the coefficients by applying a standardization technique. Here, we will utilize unit normal scaling to scale all the variables to have roughly mean zero and unit variance. This is done by subtracting the sample mean and dividing by the sample standard deviation for both the response and the regressors. We can call this scaled model "model_3." Doing a quick analysis of this model, we can likewise perform the hypotheses tests for the significance of regression and for the individual regression coefficients. Comparing this to a base intercept-only model using the scaled response variable, the result gives a test statistic of $F_0 = 29.437$ with a corresponding p-value of 3.273×10^{-8}. The set of p-values for each of the coefficients can be seen in Table 14 of the Appendix. The results are similar to before, where there is statistical significance for all the terms except for the intercept.

After having scaled best_1a into model_3, the confidence interval for each of the coefficients can be seen in Table 15 of the Appendix. Here, it is simpler to look at the coefficients in the context of each other. It is worth noting also here that none of the confidence intervals contain zero (except for the intercept term), as expected due to the p-value results of hypotheses on the individual regression coefficients. Here, x_8 is slightly larger than x_2 in terms of magnitude. The regressor x_8 is related to the opponent's rushing yards. This is the number of yards gained by opponents whenever they are executing plays the involve "rushing." The coefficient itself has a negative value, so this indicates that as the opponents gain more yards rushing, it has a negative relationship with the number of games won. This is intuitive, but the connection is that teams with stronger defense who can protect against rushing plays do better. The other coefficient with the next largest magnitude is x_2, which is the number of passing yards. This implies that teams that gain more yards passing the ball (rather than rushing perhaps) is a strong indicator of their ability to win games. This also is common sense, since it is expected that teams which win often are capable of making plays where they are successfully throwing the ball. The more interesting relationship seems to be that the model favors a stronger defense more than a strong offense, albeit the difference is extremely minor or possibly even negligible. The other regressor, x_7, is related to the ratio of rushing plays for a team. It seems that in 1976, teams that executed a large number of rushing plays tended to be successful.

It is worth reiterating how small the model is compared to the original dataset size. Here, in the selected model, only three variables are chosen. The original dataset included nine regressors. It seems then that there is perhaps an issue with some of these other statistics regarding their ability to help model the number of games won in that season. An initial problem was multicollinearity, where if looking closely at the regressors, some of them have similar names and conceptually are not too distinct. For example, some variables are related to rushing,

while others are related to passing the ball and gaining yards. This possibly indicates that the major factors for winning teams is simply a strong defense and the ability to gain yards against their opponents. Other factors such as penalty yards and field goals are less deterministic of a team's ability to win games.

Conclusion

The project itself was fairly open ended. The goal primarily is to apply the theory and techniques studied throughout the course onto a chosen dataset. The NFL dataset was useful in that it was of a manageable size and the data was comparatively easy to interpret. Fitting several different models led to a surprisingly reduced model. It had a noticeable improvement over the intercept-only and full models in terms of the predictive capabilities.

Applying the core concepts to the dataset led to a model that could be thoroughly understood and discussed. It helped to better understand the NFL statistics within the context of how they are associated with the number of games won (in the 1976 NFL season). There could perhaps be more done, for example trying to further explore polynomial regression or some general least squares method. However, the former was problematic in terms of coding in R, while the latter was never done in an applied sense from classwork. Therefore, the simpler model chosen through the all possible regressions seemed in the end to be quite suitable. I think that if for example more statistics that are less correlated with each other were included, then the model could grow larger. Using some other more complex transformations could possibly lead to linearly independent regressors with effective predictive power. However, the tradeoff of these more complex approaches towards modeling the data is that there is less ability to discuss in a straightforward sense the meaning or interpretation of the final model. For example, applying some Box-Cox or log transformation could potentially improve something like the $R^2_{Prediction}$, however there would be a great loss in the ability to make direct sense of the results. To elaborate, with this model we can see that a strong defense against opponent rushing is helpful. However, it would be difficult to say exactly what to do in the case of x_8^2, since there is no direct meaning for the square of the opponents' rushing yards.

In the end, I am quite satisfied with the final model. Applying scaling to the resulting model made for an easy comparison between the regressors. The understanding of the model is fairly in line with an interpretation of football or sports in general. That is, a combination of a strong offense and defense lead to a team winning more games. There are likely more nuanced pieces of information but given the limited dataset size and a limited knowledge of the game it is not expected to uncover too many new pieces of information. Furthermore, applying the techniques learned in class helped to reinforce the understanding of the process and the results. I think that for future work, it would be interesting to work with a larger dataset consisting of thousands of rows of data and possibly a hundred or more attributes. This would make for the ability to do more variable exploration and complex model development. At this level, only more superficial insights could be derived, but to find more valuable information I believe would require more data.

Reference

[1] *2020 NFL Turnover Differential | The Football Database*. (n.d.). FootballDB.Com. Retrieved December 8, 2020, from https://www.footballdb.com/stats/turnovers.html

[2] Calkins, K. G. C. (2005, July 18). *Correlation Coefficients*. https://www.andrews.edu/. https://www.andrews.edu/%7Ecalkins/math/edrm611/edrm05.htm

Appendix

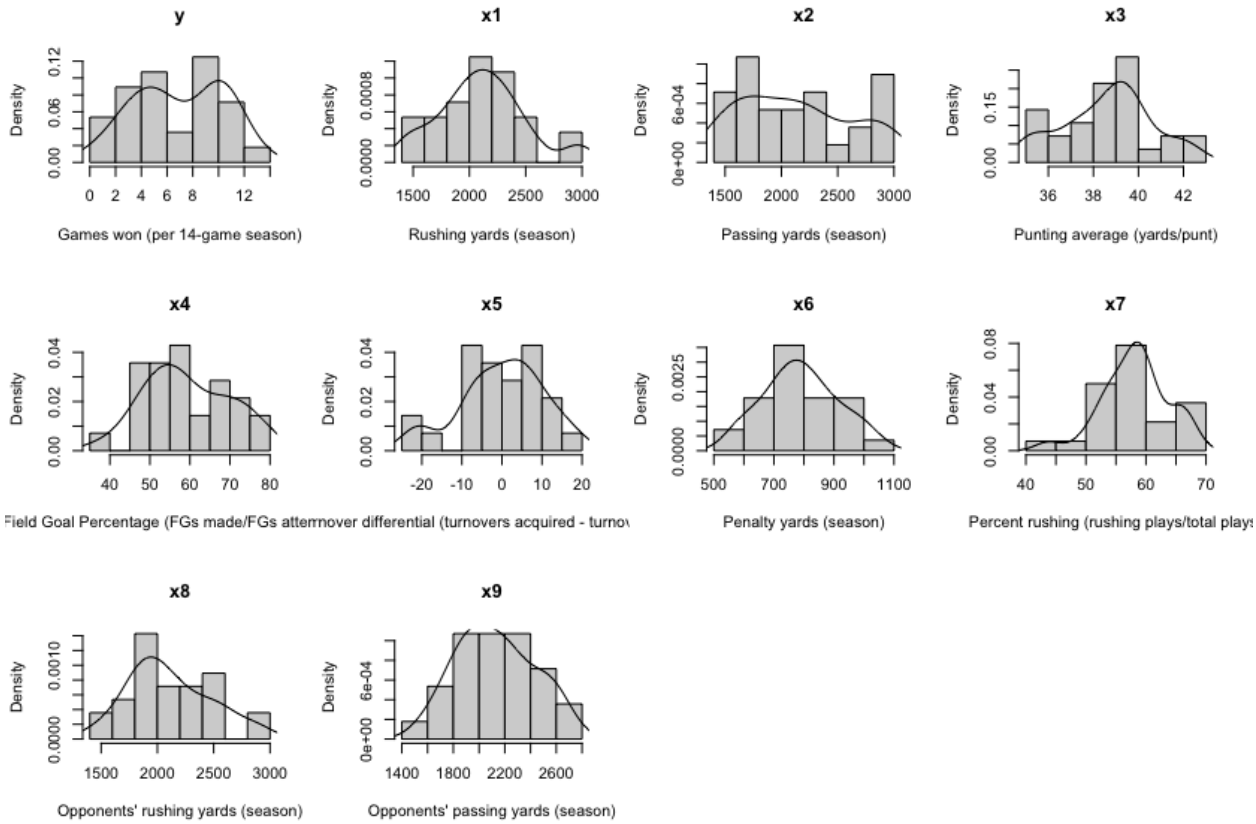

Figure 1 The above figure shows the density plots for each of the variables in the dataset.

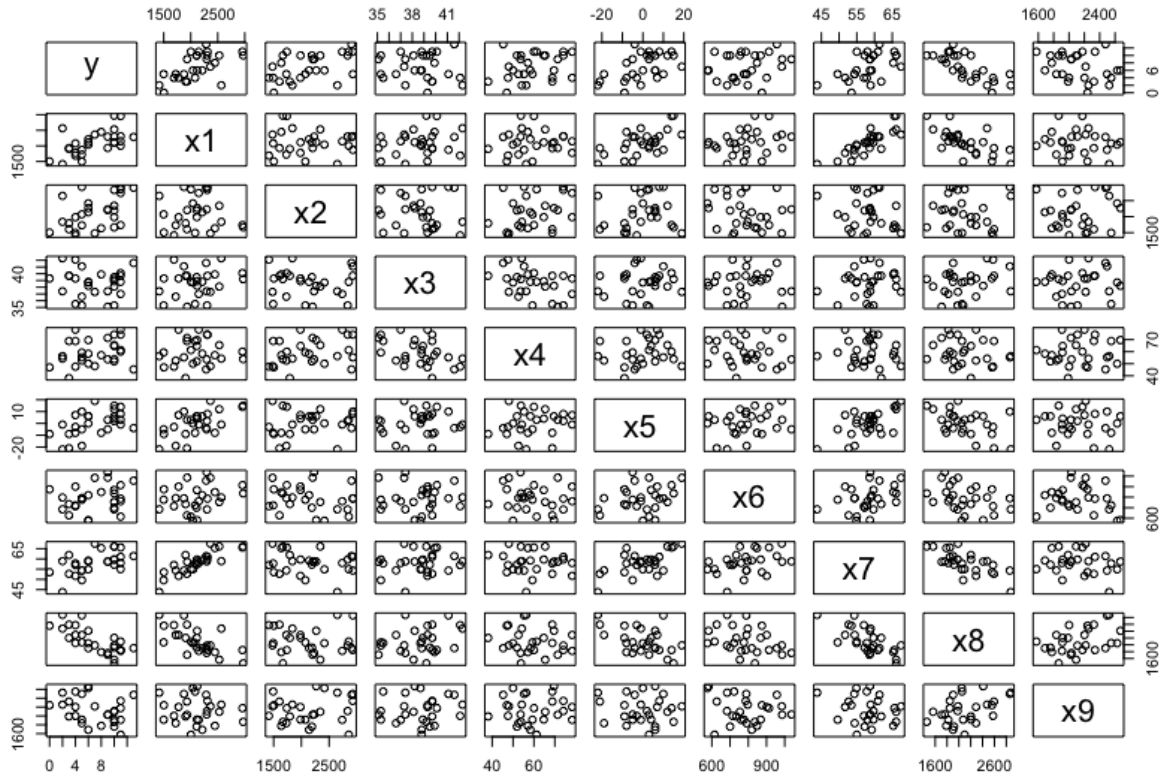

Figure 2 The above figure shows the pairwise scatterplot for each of the variables in the dataset.

Table 1 The table below shows the correlation matrix between all the variables in the dataset. The upper and lower diagonals are symmetric and so the lower triangle has been zeroed out along with the values being rounded to the second decimal place. Cells colored in red indicate high correlation (0.7-0.9) and cells in blue indicate moderate correlation (0.5-0.7).

	y	x1	x2	x3	x4	x5	x6	x7	x8	x9
y	1	0.59	0.48	0.08	0.26	0.51	0.22	0.55	0.74	0.3
x1	0	1	0.04	0.21	0.07	0.6	0.25	0.84	0.66	0.11
x2	0	0	1	0.07	0.3	0.13	0.19	0.2	0.05	0.15
x3	0	0	0	1	0.41	0.12	0	0.16	0.29	0.09
x4	0	0	0	0	1	0.15	0.13	0.1	0.16	0.06
x5	0	0	0	0	0	1	0.26	0.61	0.47	0.09
x6	0	0	0	0	0	0	1	0.37	0.35	0.17
x7	0	0	0	0	0	0	0	1	0.69	0.2
x8	0	0	0	0	0	0	0	0	1	0.42
x9	0	0	0	0	0	0	0	0	0	1

Table 2 The below table shows the variance inflation factor (VIF) for each of the attributes.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
4.8276	1.4202	2.1266	1.5661	1.9240	1.2760	5.4146	4.5356	1.4234

Table 3 The below table shows the p-values for the test on individual regression coefficients for model_1. The x_0 term indicates the intercept term in the multiple linear regression model. Only x_2 has a p-value below 0.05, and x_8 has a p-value below 0.1.

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
0.5763	0.6903	0.0004	0.6427	0.4533	0.9997	0.6303	0.3235	0.0738	0.2225

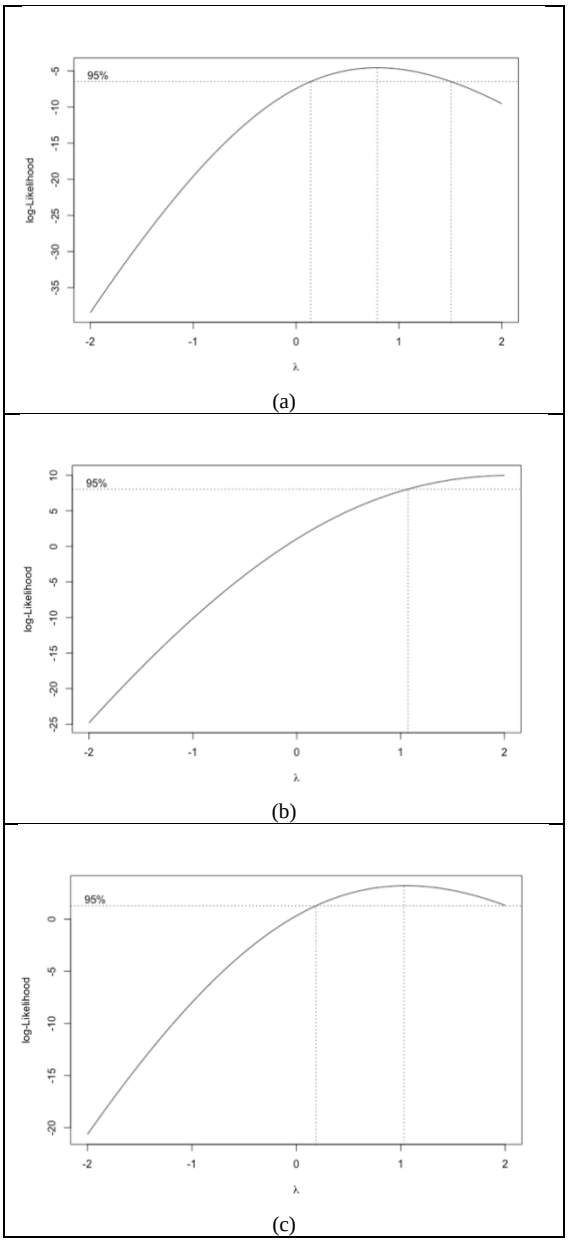

Figure 3 The above figure shows the results from using the "boxcox()" function from the "MASS" library on the dataset. The first figure (a), shows the result from calling it on the raw dataset, excluding the observation with $y = 0$. The second figure (b),

shows the result from doing the same command in R, but also applying natural log so that the formula regresses $\ln y$ on the regressors. The third figure (c), instead uses the power transformation on y where the power is 3/4.

Table 4 The below table shows the p-values for the test on individual regression coefficients for model_2. The x_0 term indicates the intercept term in the multiple linear regression model. Only x_2 has a p-value below 0.05, and x_8 has a p-value below 0.1.

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
0.8048	0.6502	0.0007	0.7984	0.4462	0.8817	0.8184	0.3369	0.0928	0.2121

Table 5 The table below shows the results of using forward selection and stepwise regression leading to the forward_1 and step_1 models. The left column shows the potential model at each step of forward selection and stepwise regression. The right column shows the AIC score for each corresponding model.

Model	AIC
$y \sim 1$	70.81
$y \sim x_8$	50.78
$y \sim x_8 + x_2$	36.74
$y \sim x_8 + x_2 + x_7$	33.6
$y \sim x_8 + x_2 + x_7 + x_9$	33.58

Table 6 The table below shows the results of using backward elimination leading to the backward_1 model. The left column shows the potential model at each step of backward elimination. The right column shows the AIC score for each corresponding model.

Model	AIC
$y \sim x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9$	41.48
$y \sim x_1 + x_2 + x_3 + x_4 + x_6 + x_7 + x_8 + x_9$	39.48
$y \sim x_2 + x_3 + x_4 + x_6 + x_7 + x_8 + x_9$	37.73
$y \sim x_2 + x_3 + x_4 + x_7 + x_8 + x_9$	36.05
$y \sim x_2 + x_4 + x_7 + x_8 + x_9$	34.82
$y \sim x_2 + x_7 + x_8 + x_9$	33.58

Table 7 The table below shows the result from using all possible regression to create the best_1 model. The columns indicate the different criterion that can be used to judge a model. The second row indicates the size of the model that corresponds to the optimized value of each criterion. The model sizes themselves correspond to a specific model of that size generated by using the regsubsets() function in R.

Criterion	R^2_{Adj}	MS_{Res}	C_p	AIC	BIC
Model size	4	4	3	4	3

Table 8 The table below shows the p-values for the individual coefficients from models best_1a (top row) and best_1b (bottom row). The columns indicate which regressor the p-values are associated with. The N/A indicates how there is no p-value for x_9 in best_1a, since it does not contain that corresponding regressor.

	x_0	x_2	x_7	x_8	x_9
best_1a	0.8209	0.0000	0.0378	0.0009	N/A
best_1b	0.8170	0.0000	0.0225	0.0086	0.2024

Table 9 The table below shows the estimated coefficients for model best_1a.

$\hat{\beta}_0$	$\hat{\beta}_2$	$\hat{\beta}_7$	$\hat{\beta}_8$
-1.8084	0.0036	0.1940	-0.0048

Table 10 The table below shows the residuals and scaled residuals for the dataset based on model best_1a. The observation number corresponds with the i'th observation from the dataset in order that it is presented in the Textbook. The residuals are all rounded to the 4th decimal place. Residuals with an absolute value greater than 2.5 have been highlighted in yellow.

Observation	Residual	Standardized Residuals	Studentized Residuals	R-Student	PRESS
1	3.7049	2.1714	2.2319	2.4544	3.9141
2	1.9613	1.1495	1.2256	1.2392	2.2296
3	2.729	1.5994	1.7026	1.7776	3.0926
4	1.6107	0.944	1.0298	1.0311	1.9166
5	0.0094	0.0055	0.0061	0.006	0.0116
6	-0.6557	-0.3843	-0.4189	-0.4116	-0.779
7	-1.904	-1.1159	-1.2068	-1.219	-2.2269
8	0.4798	0.2812	0.2993	0.2936	0.5437
9	2.0745	1.2158	1.338	1.3616	2.5125
10	-2.306	-1.3515	-1.4418	-1.4768	-2.6243
11	-0.0551	-0.0323	-0.0365	-0.0357	-0.0702
12	2.0618	1.2084	1.2511	1.2668	2.2101
13	-0.1365	-0.08	-0.0839	-0.0821	-0.15
14	-0.2582	-0.1513	-0.1607	-0.1574	-0.2911
15	-2.2197	-1.3009	-1.3354	-1.3587	-2.3388
16	1.0501	0.6155	0.645	0.637	1.1533
17	-0.2896	-0.1697	-0.1969	-0.1929	-0.3899
18	-0.4853	-0.2844	-0.365	-0.3583	-0.7993
19	-0.1274	-0.0746	-0.079	-0.0773	-0.1427
20	-0.3317	-0.1944	-0.2065	-0.2023	-0.3742
21	-3.037	-1.7799	-1.8699	-1.9805	-3.3519
22	1.2899	0.756	0.8173	0.8114	1.5075
23	-0.8846	-0.5185	-0.5511	-0.5429	-0.9993
24	-0.4417	-0.2589	-0.2765	-0.2712	-0.504
25	-1.6708	-0.9793	-1.0186	-1.0194	-1.8077
26	-0.1504	-0.0881	-0.0941	-0.0921	-0.1712
27	-0.369	-0.2163	-0.2621	-0.257	-0.5421
28	-1.6487	-0.9663	-1.0487	-1.051	-1.9421

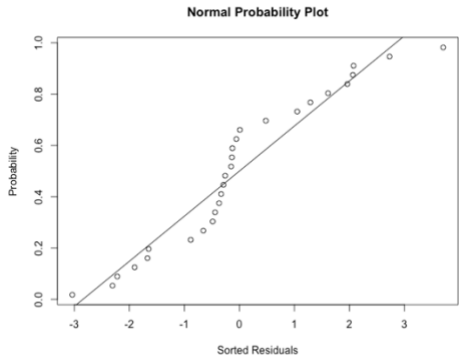

Figure 4 The above figure shows the normal probability plot of the residuals based on model best_1a.

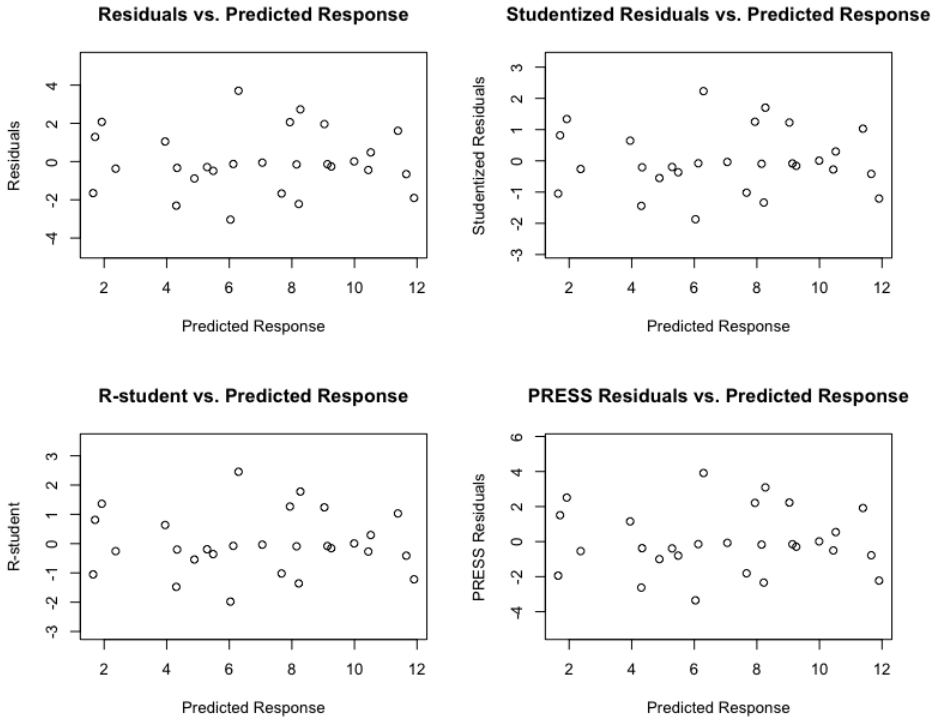

Figure 5 The above plot shows the different scaled (and unscaled) residuals plotted against the predicted response. Clockwise form the top-left shows: Residuals, Studentized Residuals, R-student, and PRESS residuals.

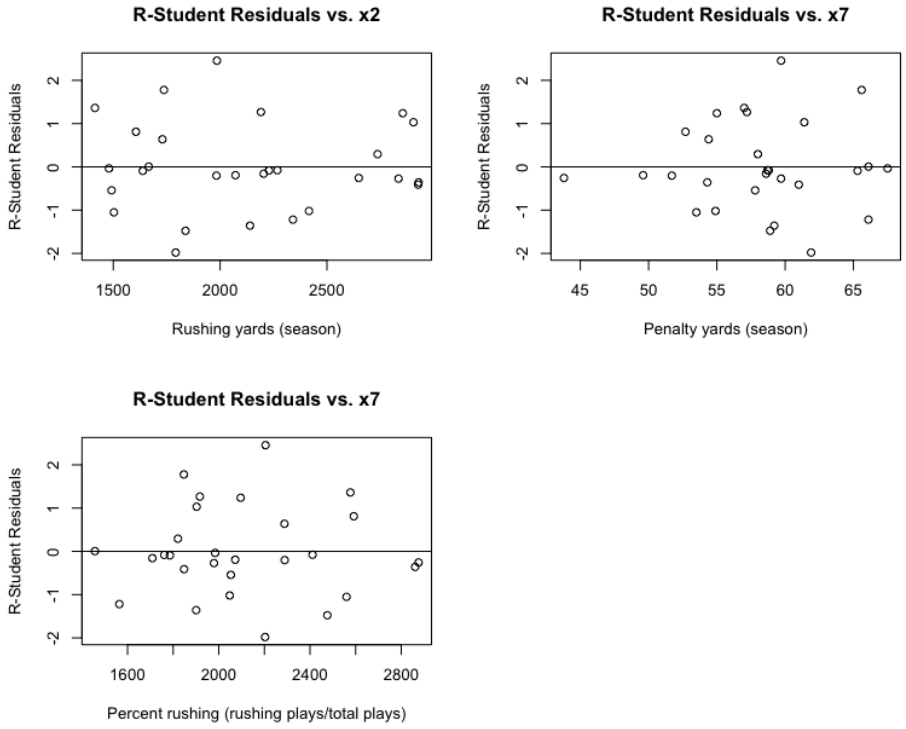

Figure 6 The above figure shows the R-Student residuals plotted against each of the regressors. Clockwise from the top-left it shows x_2, x_7, and x_8.

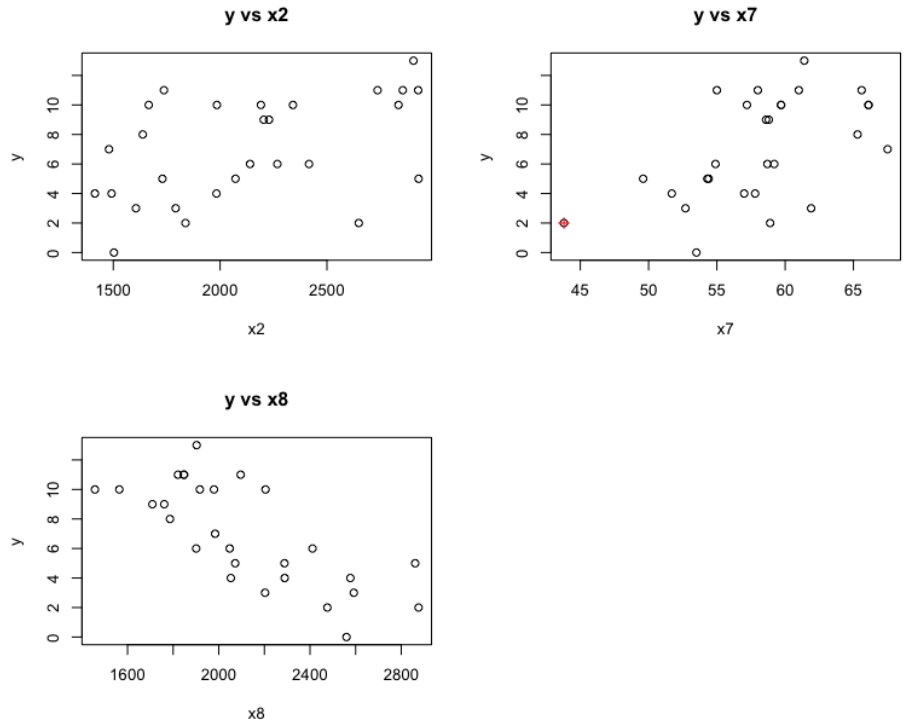

Figure 7 The above figure plots the response against each of the chosen regressors in best_1a. Clockwise from the top-left it shows y against x_2, x_7, and x_8. A red cross in the top-right plot indicates an influential point.

Table 11 The below table shows the VIFs for the regressors in model best_1a.

Regressor	x_2	x_7	x_8
VIF	1.1160	2.0973	2.0213

Table 12 The table below shows the 95% confidence intervals for each of the coefficients from the best_1a model.

Coefficient	$\hat{\beta}_0$ (intercept)	$\hat{\beta}_1$ (x_2)	$\hat{\beta}_2$ (x_7)	$\hat{\beta}_3$ (x_8)
Confidence Interval	(-18.1149, 14.4982)	(0.0022, 0.0050)	(0.0119, 0.3761)	(-0.0075, -0.0022)

Table 13 The table below shows the range (i.e., min and max) of the variables in the best_1a model.

Regressor	x_0	x_2	x_7	x_8
Range (Min, Max)	(1,1)	(1414, 2929)	(43.8, 67.5)	(1457, 2876)

Table 14 The table below shows the corresponding p-values for each of the regressors in model_3.

Regressors	x_0	x_2	x_7	x_8
p-values	1.0000	0.0000	0.0378	0.0009

Table 15 The below table shows the 95% confidence intervals for each of the coefficients from model_3.

Coefficient	$\hat{\beta}_0$ (intercept)	$\hat{\beta}_1$ (x_2)	$\hat{\beta}_2$ (x_7)	$\hat{\beta}_3$ (x_8)
Confidence Interval	(-0.1912, 0.1912)	(0.3103, 0.7218)	(0.0184, 0.5824)	(-0.7828, -0.2290)

Code Appendix

```
library(MASS) # Load library

### Load data
df <- MPV::table.b1
head(df)
col_names <- c('Games won (per 14-game season)',
  'Rushing yards (season)',
  'Passing yards (season)',
  'Punting average (yards/punt)',
  'Field Goal Percentage (FGs made/FGs attempted)',
  'Turnover differential (turnovers acquired - turnovers lost)',
  'Penalty yards (season)',
  'Percent rushing (rushing plays/total plays)',
  'Opponents\' rushing yards (season)',
  'Opponents\' passing yards (season)')

team_names <- c("Washington", "Minnesota", "New England", "Oakland", "Pittsburgh", "Baltimore",
  "Los Angeles", "Dallas", "Atlanta", "Buffalo", "Chicago", "Cincinnati", "Cleveland",
  "Denver", "Detroit", "Green Bay", "Houston", "Kansas City", "Miami", "New Orleans",
  "New York Giants", "New York Jets", "Philadelphia", "St. Louis", "San Diego",
  "San Francisco", "Seattle", "Tampa Bay")

### Graphical data displays
x_labels <- c('Games won (per 14-game season)',
              'Rushing yards (season)',
              'Passing yards (season)',
              'Punting average (yards/punt)',
              'Field Goal Percentage (FGs made/FGs attempted)',
              'Turnover differential (turnovers acquired - turnovers lost)',
              'Penalty yards (season)',
              'Percent rushing (rushing plays/total plays)',
              'Opponents\' rushing yards (season)',
              'Opponents\' passing yards (season)')
var_names <- c('y', 'x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7', 'x8', 'x9')
par(mfrow = c(3,4))
for (i in 1:ncol(df)) {
  hist(df[,i], probability = TRUE,
       main = var_names[i],
       xlab = x_labels[i])
  lines(density(df[,i]))
}
dev.off()

pairs(df) # pairs matrix

# Reference: https://stackoverflow.com/questions/9439619/replace-all-values-in-a-matrix-0-1-with-0/9439694
# Reference: https://stackoverflow.com/questions/3192791/find-indices-of-non-zero-elements-in-matrix/3193207
### correlation matrix
cor_df <- abs(cor(df))
# diag(cor_df) <- 0
cor_df[lower.tri(cor_df)] <- 0
cor_df <- as.matrix(cor_df)
cor_df <- ifelse(cor_df < 0.5, 0, cor_df)
which(cor_df != 0, arr.ind = TRUE)
# correlation > 0.5
write.table(round(cor_df, 2), file = 'correlation.txt', sep = ',')

# Reference: https://stackoverflow.com/questions/39731068/how-to-let-a-matrix-minus-vector-by-row-rather-than-by-column
# Reference: https://stackoverflow.com/questions/3444889/how-to-use-the-sweep-function
# Reference: http://www.sthda.com/english/articles/39-regression-model-diagnostics/160-multicollinearity-essentials-and-vif-in-r/
### VIF pp.117-118, pp.296-297 in Textbook
ones <- rep(1, nrow(df))
X <- as.matrix(cbind(ones, df[,2:ncol(df)]))
```

```r
sample_means <- colMeans(X)
sample_sd <- sqrt(diag(cov(X)))
X_centered <- sweep(X, 2, sample_means)
X_standardized <- sweep(X_centered, 2, sample_sd, FUN = "/")
W <- X_standardized[,2:ncol(X_standardized)]

C <- solve(t(W) %*% W)
# 1 or more large VIFs indicate multicollinearity
diag(C) # All VIF's are fairly controlled
car::vif(model_1) # x1, x7, and x8 are relatively large (around 5)

### Modeling
# often the variables aren't entirely normal
# there are limited observations
# the response seems to have two peaks
model_0 <- lm(y~1, df)
model_1 <- lm(y~., df)

# Significance of regression
# F: 8.8458
# p-value: 5.303e-05
anova(model_0, model_1)

# Check contribution of each term
# most aren't significant (only x2 > 0.05)
summary(model_1)
round(summary(model_1)$coefficients[,4], 4)

# BoxCox
boxcox(df[-28,]$y~., data=df[-28,]) # one response is 0 (28th)

# natural log is not so good
boxcox(log(df[-28,]$y)~., data=df[-28,])

# 3/4 also works
boxcox(df[-28,]$y^(3/4)~., data=df[-28,])
model_2 <- lm(y^(3/4)~., df)
model_2b <- lm(y^(3/4)~1, df)
anova(model_2, model_2b)
summary(model_2) # same issue
round(summary(model_2)$coefficients[,4], 4)

# forward selection
forward1 <- MASS::stepAIC(model_0,
                          scope = list(upper=model_1, lower=model_0),
                          direction = c('forward'))
summary(forward1) # lm(formula = y ~ x8 + x2 + x7 + x9, data = df)

# backward elimination
backward1 <- MASS::stepAIC(model_1, direction = c('backward'))
summary(backward1) # lm(formula = y ~ x2 + x7 + x8 + x9, data = df)

# stepwise regression
step1 <- MASS::stepAIC(model_0,
                       scope = list(upper=model_1, lower=model_0),
                       direction = c('both'))
summary(step1) # lm(formula = y ~ x8 + x2 + x7 + x9, data = df)

# all possible regressions
best1 <- leaps::regsubsets(x = y~., data = df, nvmax = 10)
best1_sum <- summary(best1)
p.m <- 2:10
n <- nrow(df)
MS_Res <- best1_sum$rss / (n-p.m)
aic <- n * log(best1_sum$rss / n) + 2 * p.m
data.frame(
  adjrsq = which.max(best1_sum$adjr2),
  rsq = which.max(best1_sum$rsq),
```

```
  CP = which.min(best1_sum$cp),
  MSRes = which.min(MS_Res),
  bic = which.min(best1_sum$bic),
  aic = which.min(aic)
)

# voted 3: CP, bic
# voted 4: adjrsq, MSRes, aic
# 3: x2, x7, x8
best_1a <- lm(y~x2 + x7 + x8, data = df)
summary(best_1a)
round(summary(best_1a)$coefficients[,4],4)
anova(model_0, best_1a)

# 4: x2, x7, x8, x9
best_1b <- lm(y~x2 + x7 + x8 + x9, data = df)
summary(best_1b)
round(summary(best_1b)$coefficients[,4],4)
anova(model_0, best_1b)

### Residual analysis
beta_hat_calc <- function(X, y) {
  X <- as.matrix(X)
  beta_hat <- solve(t(X) %*% X) %*% t(X) %*% y
  return(beta_hat)
}
SS_Res_calc <- function(y, beta_hat, X) {
  SS_Res <- (t(y) %*% y) - (t(beta_hat) %*% t(X) %*% y)
  return(SS_Res)
}
H_calc <- function(X) {
  X <- as.matrix(X)
  H <- X %*% solve(t(X) %*% X) %*% t(X)
  return(H)
}
ones <- rep(1, nrow(df))

# residuals
e <- best_1a$residuals

# standardized residuals
standardized_res <- sapply(1:n, function(x) e[x] / sqrt(MS_Res))

# studentized residuals
beta_hat <- best_1a$coefficients
X <- cbind(ones, df[,c('x2','x7','x8')])
H <- H_calc(X)
SS_Res <- SS_Res_calc(y = df$y, beta_hat = beta_hat, X = X)
p <- ncol(X)
MS_Res <- SS_Res / (n - p)
H_diag <- diag(H)
studentized_residuals <- sapply(1:n, function(x) {
  e[x] / sqrt(MS_Res * (1 - H_diag[x]))
})

# PRESS residuals
PRESS_res <- sapply(1:n, function(x) e[x] / (1 - H_diag[x]))

# R-Student residuals
S_squared <- sapply(1:n, function(x) {
  ((n - p) * MS_Res - ((e[x]^2) / (1 - H_diag[x]))) / (n - p - 1)
})

R_student_res <- sapply(1:n, function(x) {
  e[x] / sqrt(S_squared[x] * (1 - H_diag[x]))
})

# residual table
```

```
res_table <- data.frame(
  Observation = seq(1, n),
  Residual = e,
  Standardized_Residuals = standardized_res,
  Studentized_Residuals = studentized_residuals,
  R_Student = R_student_res,
  PRESS = PRESS_res
)
res_table <- round(res_table, 4)
abs(res_table)
write.table(res_table, file = 'res_table.txt', sep = ',')

norm_prob_plot <- function(residual_var, x_label,
                           main_title = 'Normal Probability Plot',
                           y_label = 'Probability', n_size=n) {
  ones <- rep(1, n)
  sorted_residuals <- sort(residual_var)
  cumulative_probability <- (1:n_size - 0.5) / n_size
  plot(sorted_residuals, cumulative_probability, main = main_title,
       xlab = x_label,
       ylab = y_label)
  X_temp <- cbind(ones, sorted_residuals)
  beta_hat_temp <- beta_hat_calc(X=X_temp,y=cumulative_probability)
  abline(beta_hat_temp)
}
norm_prob_plot(residual_var = best_1a$residuals, x_label = 'Sorted Residuals')

order(e, decreasing = FALSE)
e[order(e, decreasing = FALSE)]

# res vs fitted
y_hat <- best_1a$fitted.values
res_vs_fitted_plot <- function(residual_var,
                               main_title,
                               y_label,
                               x_label = 'Predicted Response',
                               pred_response = y_hat) {
  plot(pred_response, residual_var, main = main_title,
       xlab = x_label,
       ylab = y_label,
       ylim = c(min(residual_var)-sd(residual_var),
                max(residual_var)+sd(residual_var)))
}
par(mfrow = c(2,2))
res_vs_fitted_plot(residual_var = e,
                   main_title = 'Residuals vs. Predicted Response',
                   y_label = 'Residuals')
res_vs_fitted_plot(residual_var = studentized_residuals,
                   main_title = 'Studentized Residuals vs. Predicted Response',
                   y_label = 'Studentized Residuals')
res_vs_fitted_plot(residual_var = R_student_res,
                   main_title = 'R-student vs. Predicted Response',
                   y_label = 'R-student')
res_vs_fitted_plot(residual_var = PRESS_res,
                   main_title = 'PRESS Residuals vs. Predicted Response',
                   y_label = 'PRESS Residuals')

# res vs regressor
# R-student
res_vs_regressor <- function(residual_var,
                             main_title1, main_title2, main_title3,
                             ylabel,
                             X_df=X) {
  x2 <- X_df[,2]; x7 <- X_df[,3]; x8 <- X_df[,4]
  plot(x2, residual_var,
       main = main_title1,
       ylab = ylabel,
       xlab = col_names[2])
```

```
  abline(h = 0)

  plot(x7, residual_var,
       main = main_title2,
       ylab = ylabel,
       xlab = col_names[7])
  abline(h = 0)

  plot(x8, residual_var,
       main = main_title2,
       ylab = ylabel,
       xlab = col_names[8])
  abline(h = 0)
}

par(mfrow = c(2,2))
res_vs_regressor(residual_var = R_student_res,
                 main_title1 = 'R-Student Residuals vs. x2',
                 main_title2 = 'R-Student Residuals vs. x7',
                 main_title3 = 'R-Student Residuals vs. x8',
                 ylabel = 'R-Student Residuals',
                 X_df = X)
par(mfrow = c(2,2))
plot(df$x2, df$y,
     main = 'y vs x2',
     xlab = 'x2', ylab = 'y')
plot(df$x7, df$y,
     main = 'y vs x7',
     xlab = 'x7', ylab = 'y')
points(43.8, 2, pch = 3, col = 'red')
plot(df$x8, df$y,
     main = 'y vs x8',
     xlab = 'x8', ylab = 'y')

### model validation
# VIFs
car::vif(best_1a)

# data splitting
SS_T_calc <- function(y) {
  n <- length(y)
  SS_T <- (t(y) %*% y) - ((sum(y)^2) / n)
  return(SS_T)
}
SS_T <- SS_T_calc(y = df$y)
PRESS <- sum(PRESS_res^2)
R_squared_pred <- 1 - PRESS / SS_T

# data split
df_split <- data.frame(y = df$y, x2 = df$x2, x7 = df$x7, d8 = df$x8)
n_train <- n * 0.5
n_test <- n * 0.5
idx <- 1:n
set.seed(1)
train_idx <- sort(sample(idx, n_train))
test_idx <- idx[!(idx %in% train_idx)]
model_test <- lm(y~x2+x7+x8, data = df_split[train_idx,])
y_pred <- predict(model_test, df_split[test_idx,])
test_error <- sum((y_pred - df_split[test_idx,1])^2) # 62.40468

# compare to intercept-only
model_base <- lm(y~1, data = df_split[train_idx,])
y_pred <- predict(model_base, df_split[test_idx,])
test_error <- sum((y_pred - df_split[test_idx,1])^2) # 185

# compare to full model
model_full <- lm(y~., data = df[train_idx,])
y_pred <- predict(model_full, df[test_idx,])
```

```r
test_error <- sum((y_pred - df[test_idx,1])^2) # 295.5793


### polynomial
# Reference: https://www.ics.uci.edu/~jutts/201-F13/Lecture13.pdf
# Reference: https://stats.stackexchange.com/questions/25975/how-to-add-second-order-terms-into-the-model-in-r/25977
X_poly <- as.data.frame(cbind(y=df$y, poly(X[,2:ncol(X)], degree = 2)))
model_6 <- lm(y~., data=X_poly)
summary(model_6)

### CI's
X <- as.matrix(X)
C <- solve(t(X) %*% X)
SS_Res <- SS_Res_calc(y = df$y, beta_hat = coef(best_1a), X = X)
sigma_hat_squared <- SS_Res / (n - p)

standard_errors <- sqrt(as.vector(sigma_hat_squared) * diag(C))
alpha <- 0.05
t_value <- qt(p = 1 - alpha / 2, df = n - p)
round(coef(best_1a) + t_value * standard_errors, 4)
round(coef(best_1a) - t_value * standard_errors, 4)
apply(X, 2, function(x) c(min(x), max(x)))

### Unit normal scaling
sample_means <- colMeans(df)
sample_sd <- sqrt(diag(cov(df)))
df_centered <- sweep(df, 2, sample_means)
df_normal_scaled <- sweep(df_centered, 2, sample_sd, FUN = "/")

model_3b <- lm(y~1, data = df_normal_scaled)
anova(model_3b, model_3)
model_3 <- lm(y~x2+x7+x8, data = df_normal_scaled)
round(summary(model_3)$coefficients[,4], 4)

# CI
X <- df_normal_scaled[,c('x2', 'x7', 'x8')]
X <- as.matrix(cbind(ones, X))
C <- solve(t(X) %*% X)
SS_Res <- SS_Res_calc(y = df_normal_scaled$y, beta_hat = coef(model_3), X = X)
sigma_hat_squared <- SS_Res / (n - p)

standard_errors <- sqrt(as.vector(sigma_hat_squared) * diag(C))
alpha <- 0.05
t_value <- qt(p = 1 - alpha / 2, df = n - p)
round(coef(model_3) + t_value * standard_errors, 4)
round(coef(model_3) - t_value * standard_errors, 4)
```