DATA 621 Homework 6 - Count Regression Models

Erik Nylander

May 1, 2016

1 Data Exploration

In this analysis we will be looking at data on ~12,000 commercially available wines. The predictor variables are primarily the chemical properties for the wine and the Target variable is the number of cases purchased by wine distributors after sampling the wine. The theory is that the number of cases ordered will ultimately determine the amount of wine purchased at high end restaurants. Our objective is to determine what characteristics will lead to a large number of cases purchased so that a winery can adjust the properties of their wines to increase sales. To facilitate our analysis we have been given a training data set that will be split into a training set containing 70% of the data and a testing data set that contains 30% of the data. We have also been given an evaluation data set that we will use to predict the number of cases sold for a new set of wines.

1.1 Explanation of Variables

The data that we will be analyzing consists of a single target variable, the number of cases of wine that were sold, and a number of different predictor variables. One of the things that we noticed when looking at a summary for the variables is that there are a number of missing values in the data. This lead us to construct a number of dummy variables to see if the fact that the data was missing is predictive in some way. The following variables are used in this analysis.

- Target Variable
  - TARGET: The number of cases of wine sold from 0 to 8.
- Predictor Variables
  - AcidIndex: Proprietary method for testing total acidity of wine using a weighted average.
  - Alcohol: Alcohol Content (653 NA's)
  - Chlorides: Chloride content of Wine (638 NA's)
  - CitricAcid: Citric Acid Content
  - Density: Density of the Wine
  - FixedAcidity: Fixed Acidity of Wine
  - FreeSulfurDioxide: Sulfur Dioxide content of Wine (647 NA's)
  - LableAppeal: Marketing score indicating the label appeal to consumers. High scores indicate customers like the label. Negative numbers indicate that consumers do not like the label.
  - ResidualSugar: Residual Sugar of Wine (616 NA's)
  - STARS: Wine rating by a team of experts. 4 Stars = Excellent, 1 Star = Poor (3359 NA's)
  - Sulphates: Sulfate content of Wine (1210 NA's)
  - TotalSulfurDioxide: Total Sulfur dioxide of Wine (682 NA's)
  - VolatileAcidity: Volatile Acid content of Wine
  - pH: pH of Wine (395 NA's)
- Dummy Variables (1 = missing, 0 = not missing)
  - STARSMissing: Was the STARS variable missing?
  - AlcoholMissing: Was the Alcohol variable missing?
  - SulphatesMissing: Was the Sulphates variable missing?
  - pHMissing: Was the pH variable missing?
  - TotalSulfurDioxideMissing: Was the TotalSulfurDioxide variable missing?

- FreeSulfurDioxideMissing: Was the FreeSulfurDioxide variable missing?
- ChloridesMissing: Was the Chlorides variable missing?
- ResidualSugarMissing: Was the ResidualSugar variable missing?

1.2 Exploring the TARGET Variable

The target variable for this analysis is count data in that the only possible values that the variable can assume are 1 - 8 which indicate the number of cases of wine sold. Given that we are looking at count data we will be constructing both Poisson and negative binomial models. One of the assumptions of the Poisson model is that the mean an variance of the data should be approximately equal. When we calculate these values we get the results in Table 1. We see that the mean and variance are fairly close but and that the variance is slightly higher then the mean indicating that we may have some over-dispersion in the model which could be improved by the negative binomial model but it would not surprise us if the Poisson fit better.

Table 1: Mean and Variance of the Cases Sold

Mean	Variance	Mean/Variance
3.0291	3.7109	0.8163

In the Figure 1 below we look at the basic shape of our the TARGET variable. Given that we have count data we are looking to see if there is an over-inflation of the zeros.If we find this over-inflation then we may be better off fitting a zero-inflated Poisson model. From the histogram of the data below we can see that there is evidence for zero-inflation in the data as we have almost as many 0's as our mean value of 3.0291. This may also be causing the over dispersion that we saw in the table above since the number of zeros may be pulling the mean lower. Given these findings we will fit a zero-inflation Poisson model to the data as well.

Figure 1: Histogram of the TARGET Variable

1.3 Exploring the Predictor Variables

Our predictor variables can be broken into two basic groups. The first are the continuous variables that indicate the measurements of some property in the wine, such as the residual sugars variable plotted in Figure 2 below. Interestingly all of the continuous variables show very similar pattern. The data has a strong central peak and is symmetrically distributed on each side of the peak. When we look at the boxplot divided up by number of cases sold we see that there are large numbers of outliers that are symmetrically distributed about the median. We explored what happens with the removal of the outliers from the data set but felt that with the removal of outliers and the number of missing observations we no longer had a good representation of the original data. We also explored the prospect of transforming the data but were unable to develop an adequate mathematical transform that improved the data.

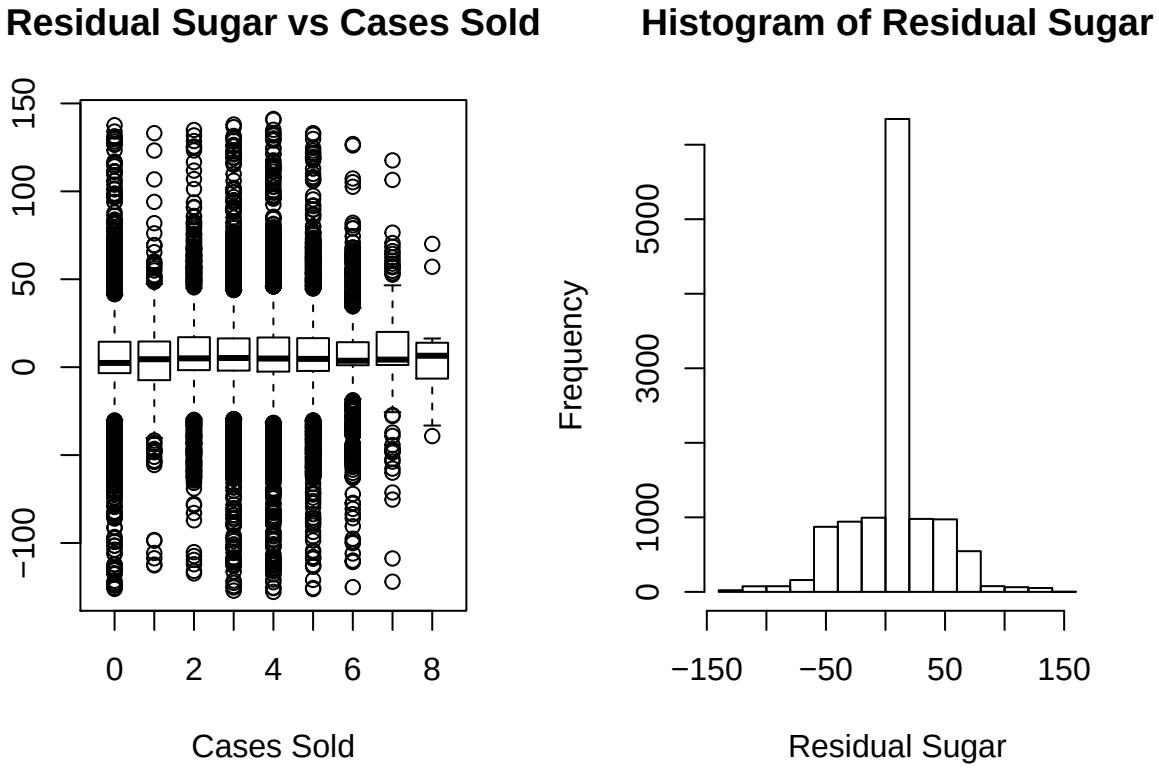


Figure 2: Summary Plots for Residual Sugars

The second group are the categorical and dummy variables. With these we elected to plot the categories vs the number of cases sold. We can see the results in the Figure 3 below. For the STARS variable we see that a high rating from the experts is correlated with more cases sold. The remaining plots can be found in Appendix A.

One final note about the data, as we look across the various predictor variables broken down by the number of cases sold we see that most of these variables change very little. This leads us to infer that they variables show little correlation with the number of cases sold. However we do see strong correlations in the STARS and LabelAppeal categories. We also see the evidence for an effect on the number of cases sold when we look at the Density, FreeSulfurDioxide, and Alcohol. We also see that there appears to be some correlation in the missing variables with the value missing relating to a slight decrease in cases sold.

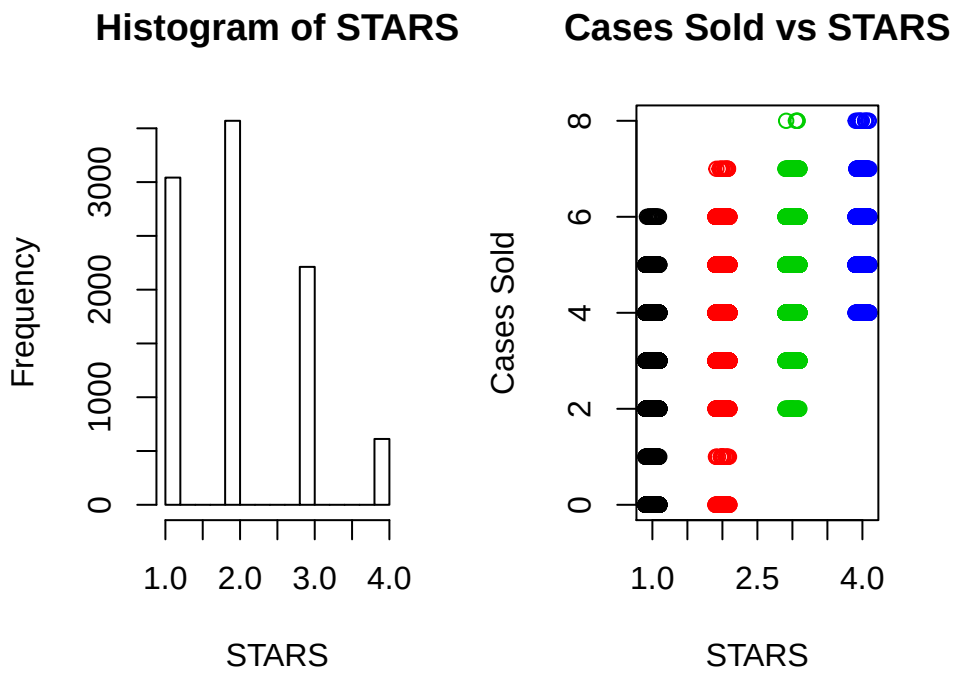


Figure 3: Summary Plots for the STARS Variable

2 Data Preperation

For this analysis we have two primary issues with our data that we would like to address. The first is the potential that a variable being missing may predictive in the model. We could easily believe that a wine that people showed little interest would not be tested in this way we may be able to predict sales from the fact that there was no data for that wine. To accomplish this we created a second data set that contained all of the dummy variables for the missing data as described above. In the model building process below we have fit one model to the original data and a second model to the new data set containing the dummy variables.

The second issue that we see with the data is the odd shape of each of the predictor variables. Each of the continuous predictor variables showed heavy tails with a very tall central peak. We tried a number of different transformations that are not included in this analysis since they had little impact on the overall shape of the data given the symmetry to begin with. Given this we felt that it would be best to move forward with the data in the original state knowing that this increased variation in each of the predictor variables will add variance to our models. We do however feel that this decision can still lead to good results given the symmetry of the predictor variables.

3 Building the Models

We will now take our data and build a series of different models for our training data. We will generate two Poisson models, two negative binomial models, two multiple linear regression models and finally a zero-inflated Poisson regression model to the data. We will also calculate the Akaike Information Criterion (AIC) on each of the models and the mean square error (MSE) of the model on the test data set that we have subset from the original training data.

3.1 Poisson Regression Models

The first two models that we will develop are the Poisson regression models for count data. For both models we used a combination of the robust residuals using the *sandwich()* package in R and a stepwise selection method to find our models.

Poisson Regression on the Full Data

The first Poisson model that we will develop is based on the full data set without the dummy variables indicating missing values. Table 2 below contains the results of the model building process including the robust residuals for predictive variables and 95% confidence intervals for each of the coefficients. All of the coefficients in the model are significant at the $\alpha = 0.05$. This model generates and AIC of 16138.06.

Table 2: Poisson Regression Model on the Full Data

	Estimate	Robust SE	Pr(>abs(z))	LL	UL
(Intercept)	1.6716142	0.1826190	0.0000000	1.3136809	2.0295475
VolatileAcidity	-0.0290096	0.0060587	0.0000017	-0.0408847	-0.0171345
Chlorides	-0.0382590	0.0148092	0.0097815	-0.0672850	-0.0092329
FreeSulfurDioxide	0.0000745	0.0000319	0.0194774	0.0000120	0.0001371
Density	-0.4656073	0.1812870	0.0102186	-0.8209298	-0.1102847
Alcohol	0.0031284	0.0012648	0.0133831	0.0006494	0.0056074
LabelAppeal	0.1778982	0.0062312	0.0000000	0.1656850	0.1901114
AcidIndex	-0.0459095	0.0055297	0.0000000	-0.0567477	-0.0350712
STARS	0.1838418	0.0063655	0.0000000	0.1713655	0.1963180

From the model outlined in Table 2 we can infer the following about this model. The volatile acidity, chlorides, density, and acid index all have a negative impact on the number of cases of wine sold. The free sulfur dioxide variable has a coefficient that is significant however we see from the value of this coefficient that the effect is minimal. The alcohol content, label appeal, and Stars rating variable all have a positive impact on the number of cases of wine sold.

Finally lets take a look at the summary plots for the residuals to see what we can about the above model. The Figure 4 below contains the diagnostic plots of the residuals. We note that there are some issues with the residuals the first is that there appears to be a second data set that is poorly predicted by the model in the residuals. We can see this as the horizontal line in the Residuals vs. Fitted plot below the rest of the residuals and again in the Scale-Location plot and finally as the second small clump of residuals below the main group in the Residuals vs Location plot. We believe that this may be an effect from the zero-inflation that we believe is in the TARGET data we also have some unusual patterns in the residuals that may have to do with the count nature of the TARGET variable.

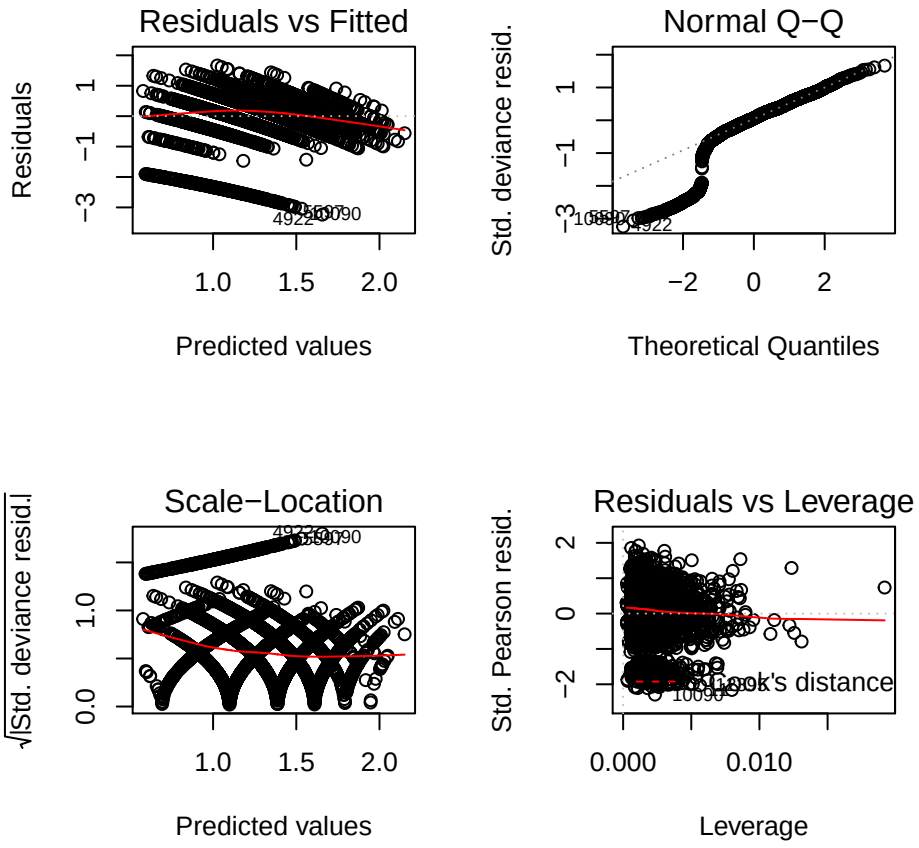


Figure 4: Diagnostic Plots of the Poisson Model fit the Full Data

Poisson Regression on the Dummy Variables

The second Poisson model that we will be fitting is the Poisson regression on the data that contains dummy variables that indicate if one of the measured predictor variables is missing. Table 3 below contains the results of the model building process including the robust residuals for predictive variables and 95% confidence intervals for each of the coefficients. All of the coefficients in the model are significant at the $\alpha = 0.05$. This model generates and AIC of 32627.82.

Table 3: Poisson Regression Model on the Dummy Data

	Estimate	Robust SE	Pr(>abs(z))	LL	UL
(Intercept)	1.9713885	0.0412890	0	1.8904621	2.0523148
VolatileAcidity	-0.0390555	0.0063441	0	-0.0514899	-0.0266210
AcidIndex	-0.0901564	0.0055830	0	-0.1010991	-0.0792137
LabelAppeal	0.2191457	0.0057313	0	0.2079123	0.2303791
STARSMissing	-1.0413420	0.0295045	0	-1.0991708	-0.9835131

From the model outlined in Table 3 we can infer the following about this model. The volatile acidity, acid index, and stars missing all have a negative impact on the number of cases of wine sold. The label appeal

rating variable has a positive impact on the number of cases of wine sold.

Lets now take a look at the diagnostic plots of the residuals for the model outlines in Table 3. In Figure 5 we can see the output from these plots. We see that the results of the diagnostic plots are very similar to what we saw in the Poisson model fit to the full data. There is still the evidence for zero-inflation in these diagnostics.

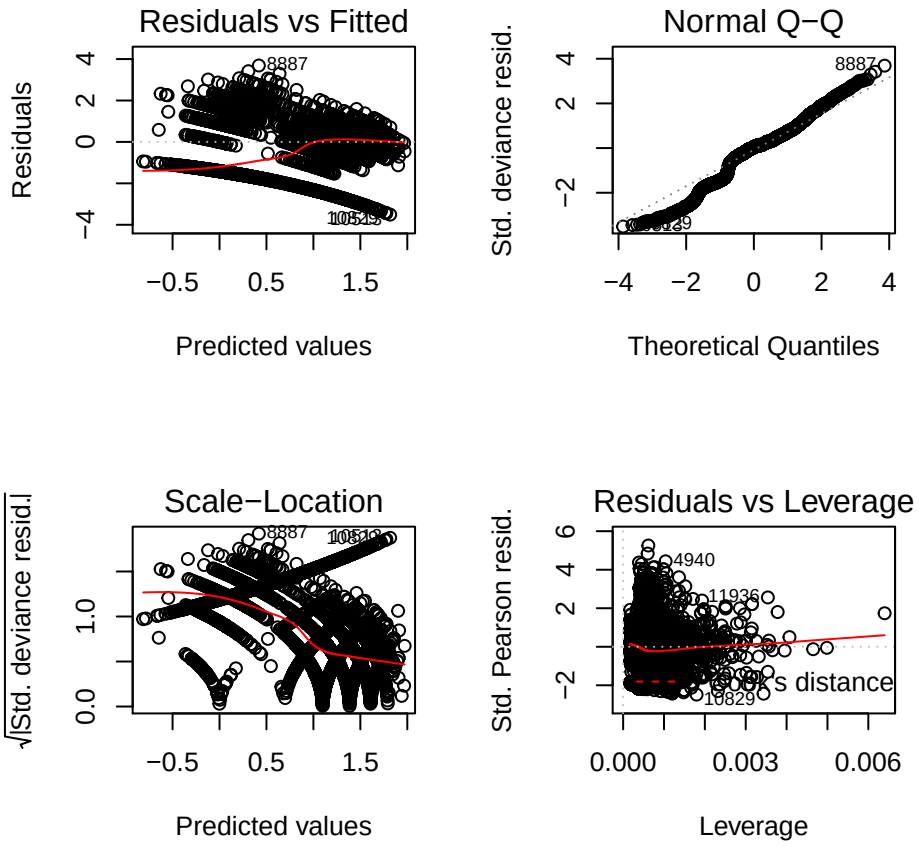


Figure 5: Diagnostic Plots of the Poisson Model fit to the Dummy Data

3.2 Negative Binomial Regression Models

The next set of models that we will fit to the data are a negative binomial model. This model helps to deal with over dispersion which may exist in the data. However we found when fitting these models that we had issues in the model fit that are consistent with the negative binomial being a good choice for the data. Specifically the glm.nb() function failed to converge to a link value. Therefore we had to set the %k=1% to get the model to run. For each of the models we used stepwise selection and robust residuals to decide which variables we should include in the model.

Negative Binomial Regression on the Full Data

For our first negative binomial model we will fit to the full data. The resulting model is outline in Table 4 below. All of the predictors are significant at the $\alpha = 0.5$ level and the AIC generated from this model is 21553.61.

Table 4: Negative Binomial Regression Model on the Full Data

	Estimate	Robust SE	Pr(>abs(z))	LL	UL
(Intercept)	1.7523791	0.2061129	0.0000000	1.3483977	2.1563604

	Estimate	Robust SE	Pr(>abs(z))	LL	UL
VolatileAcidity	-0.0329811	0.0069547	0.0000021	-0.0466123	-0.0193500
Chlorides	-0.0460980	0.0166075	0.0055078	-0.0786487	-0.0135473
FreeSulfurDioxide	0.0000813	0.0000360	0.0238228	0.0000108	0.0001518
Density	-0.4963564	0.2043215	0.0151284	-0.8968265	-0.0958863
Alcohol	0.0029488	0.0014385	0.0403789	0.0001293	0.0057684
LabelAppeal	0.1843997	0.0069696	0.0000000	0.1707392	0.1980602
AcidIndex	-0.0555012	0.0064771	0.0000000	-0.0681964	-0.0428060
STARS	0.1956394	0.0068549	0.0000000	0.1822038	0.2090751

We can see from Table 4 above that we get a very similar model to the Poisson model. This is a sign that there may not actually be over dispersion in the model and that we are getting the same model as we would in a Poisson regression. Since we have gotten this result we will elect to fit a zero inflated model to data. Figure 6 below contains the diagnostic plots of the residuals to see if they also are similar to what we found before. From the summary plots we see a very similar result to what we found when analyzing the Poisson regression and we see the same evidence for a second data set as we did before in the data.

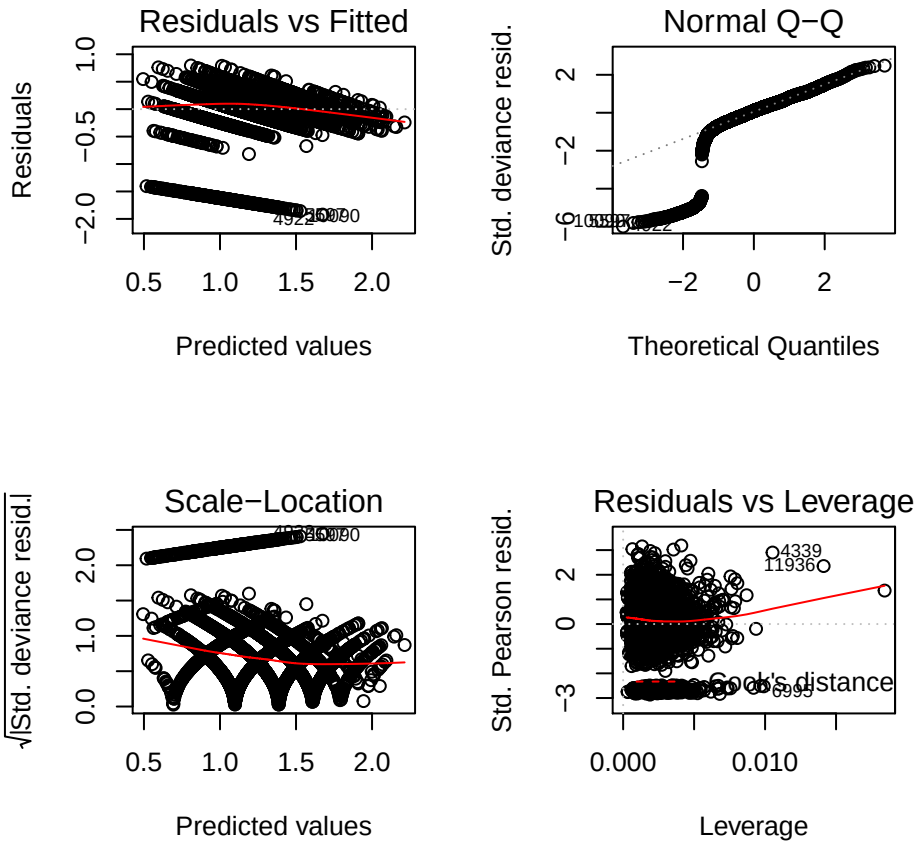


Figure 6: Diagnostic Plots of the Negative Binomial fit to the Full Data

Fitting a Negative Binomial Model to the Dummy Data

For our next model we will fit the negative to the data set containing the dummy variables that indicate that the data was missing. The resulting model is described in Table 5 below. We can see that all of the coefficients are significant at the $\alpha = 0.5$ level and we get an AIC of 38825.25.

Table 5: Negative Binomial Regression Model on the Dummy Data

	Estimate	Robust SE	Pr(>abs(z))	LL	UL
(Intercept)	2.1904333	0.0554602	0	2.0817313	2.2991353
VolatileAcidity	-0.0470726	0.0084143	0	-0.0635647	-0.0305805
LabelAppeal	0.1995211	0.0077134	0	0.1844028	0.2146394
STARSMissing	-1.0339159	0.0288021	0	-1.0903681	-0.9774636
AcidIndex	-0.1183557	0.0074463	0	-0.1329503	-0.1037610

Once again we see a that the model is very similar to the model that we got in the Poisson regression with the STARTSMissing, AcidIndex, and VolatileAcidity having a negative impact on the number of cases sold and the LabelAppeal variable having a positive impact on the number of cases sold. In Figure 7 below we see the diagnostic plots for the residuals of this model.

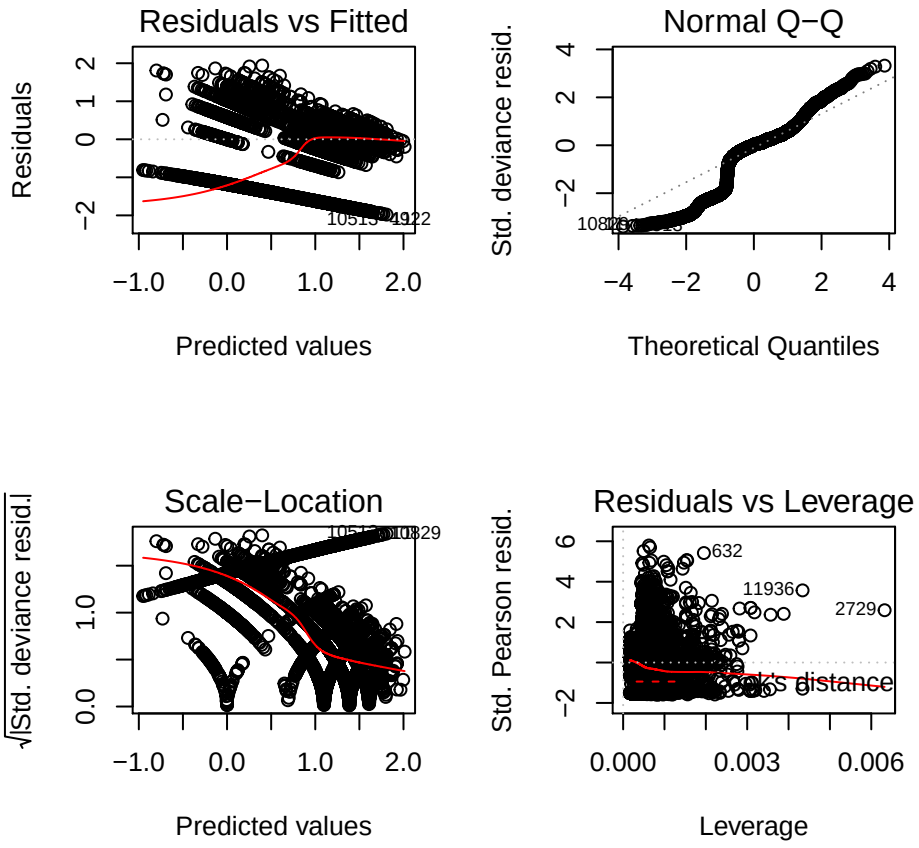


Figure 7: Diagnostic Plots of the Negative Binomial fit to the Dummy Data

3.3 Fitting a Multiple Linear Regression Models

For our next set of models we will fit a multiple linear regression model to the full data and dummy data. We will use a similar technique of stepwise selection and evaluation of the robust residuals to determine the variables to leave in the model.

Fitting a Multiple Linear Regression to th Full Data

We start by fitting a multiple linear regression to the full data set. Table 6 below contains the results of fitting the model. The model results in an AIC of 16682.74.

Table 6: Multiple Linear Regression Model on the Full Data

	Estimate	Robust SE	Pr(>abs(z))	LL	UL
(Intercept)	4.5078978	0.6110895	0.0000000	3.3101625	5.7056331
TotalSulfurDioxide	0.0001348	0.0000685	0.0489478	0.0000006	0.0002690
FreeSulfurDioxide	0.0002528	0.0001030	0.0141144	0.0000509	0.0004547
Chlorides	-0.1036315	0.0491545	0.0350068	-0.1999743	-0.0072886
Density	-1.2399010	0.6087801	0.0416804	-2.4331100	-0.0466920

	Estimate	Robust SE	Pr(>abs(z))	LL	UL
Alcohol	0.0136691	0.0040892	0.0008295	0.0056543	0.0216838
VolatileAcidity	-0.0948843	0.0203152	0.0000030	-0.1347021	-0.0550664
AcidIndex	-0.1624996	0.0161564	0.0000000	-0.1941661	-0.1308331
LabelAppeal	0.6517637	0.0214649	0.0000000	0.6096926	0.6938349
STARS	0.7253594	0.0210630	0.0000000	0.6840759	0.7666429

We can see from Table 6 that in the linear model there is the addition of one more variable, TotalSulfurDioxide, to the model. However if we look at the coefficient on this variable it has a negligible effect on number of cases sold. In Figure 8 below we can see the diagnostic plot of the residuals. Once again we see very similar patterns to what we have seen in the other residuals.

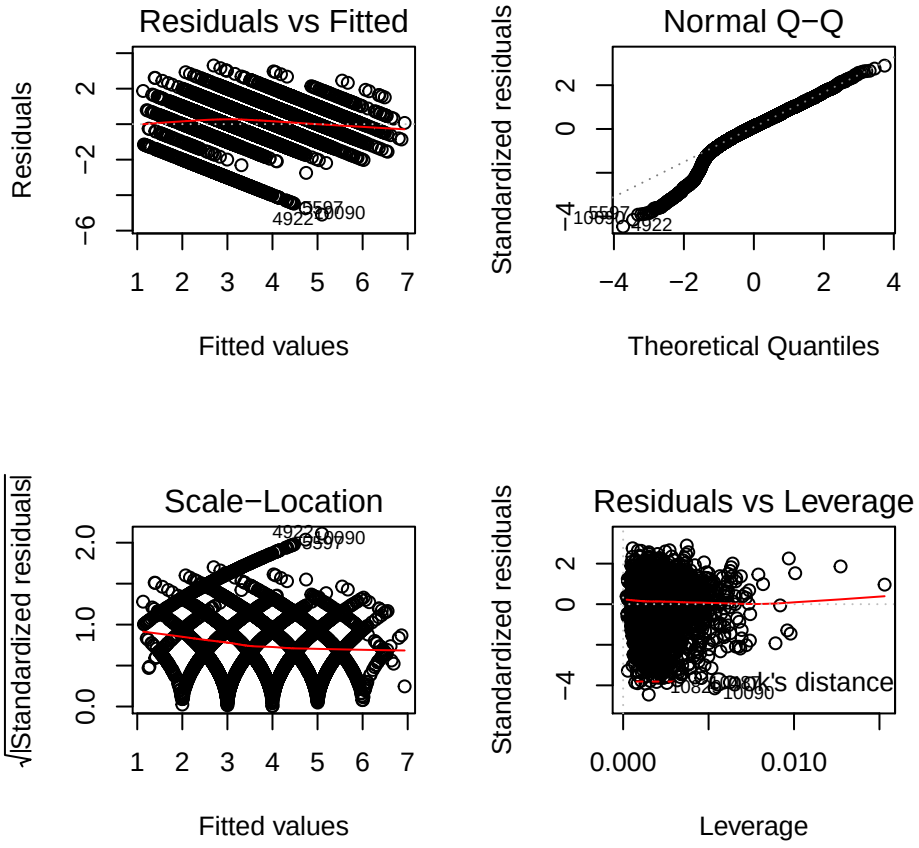


Figure 8: Diagnostic Plots of the Multiple Linear fit to the Full Data

Fitting a Multiple Linear Regression to the Dummy Data

For our final linear model we will fitting a multiple regression to the dummy variables. Table 7 below contains the coefficients of the model constructed. All variables are significant at the $\alpha = 0.05$. The model has an AIC of 31817.54.

	Estimate	Robust SE	Pr(>abs(z))	LL	UL
(Intercept)	5.4601370	0.0980060	0	5.2680453	5.6522287
VolatileAcidity	-0.1247381	0.0197432	0	-0.1634348	-0.0860414
AcidIndex	-0.2311027	0.0129713	0	-0.2565263	-0.2056790
LabelAppeal	0.6608784	0.0186503	0	0.6243238	0.6974331
STARSMissing	-2.2253331	0.0415851	0	-2.3068398	-2.1438263

From the table above we see that we have a similar set of predictor variables as what we got with the Poisson and negative binomial regressions. Overall all of our models have indicated that a similar set of variables are predictive in this model. In Figure 9 below we see that the residuals from this model are also very similar to the residuals that we have seen before indicating that there may be zero inflation in the data.

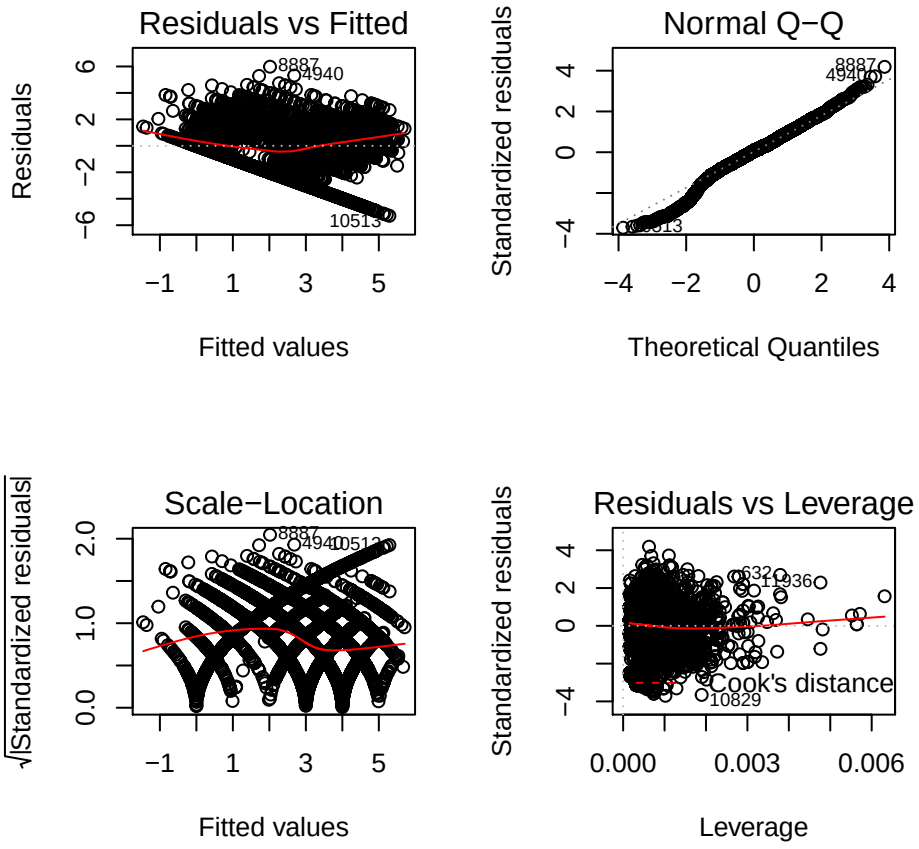


Figure 9: Diagnostic Plots of the Multiple Linear fit to the Dummy Data

3.4 Fitting a Zero-Inflation Poisson Model

For our final model we will fit a zero-inflation Poisson model. We believe from the results above that there is the potential for more zeros in the data then we would expect. The zero-inflated Poisson fits two different models to the data. The first is a logistic model to determine which of the TARGET zeros are real zeros and which are inflated. A model is then fit to the remaining data. We will use the zeroinfl function of the *pscl()* package to fit the model. The Table 8 and 9 below contains the results from fitting the model to the data. We then used an AIC based selection criterion to pick the variables. All predictors are significant at the $\alpha = 0.05$ level and the model has an AIC value of 15590.05.

Table 8: Logistic Model for Determining Zeros

	Estimate	Std. Error	z value	Pr(>abs(z))
(Intercept)	-2.9030280	0.6995798	-4.149674	0.0000333
VolatileAcidity	0.2983481	0.1129693	2.640967	0.0082670
FreeSulfurDioxide	-0.0014749	0.0005957	-2.475995	0.0132866
TotalSulfurDioxide	-0.0010847	0.0003898	-2.782969	0.0053864
Sulphates	0.2032546	0.0952078	2.134853	0.0327731
Alcohol	0.0666338	0.0245115	2.718468	0.0065585
LabelAppeal	0.6859433	0.1130794	6.066030	0.0000000
AcidIndex	0.5129925	0.0615603	8.333166	0.0000000
STARS	-3.6395560	0.4035989	-9.017755	0.0000000

Table 9: Poisson Model for the Cases of Wine Sold

	Estimate	Std. Error	z value	Pr(>abs(z))
(Intercept)	1.1524578	0.0644601	17.878630	0.0000000
Alcohol	0.0057025	0.0021383	2.666860	0.0076564
LabelAppeal	0.2086732	0.0097072	21.496739	0.0000000
AcidIndex	-0.0170298	0.0072313	-2.355006	0.0185224
STARS	0.1128689	0.0094658	11.923914	0.0000000

The interesting things that we see in this model are that once we have a way of dealing with the extra zeros in the data we need the alcohol content, the label appeal, the acid index, and the number of stars to determine the number of cases sold. The other notable thing is that the label appeal and the experts rating have the largest positive impact impact on the number of cases sold. The alcohol content of the wine has a much smaller impact on the number of cases sold and finally the acidity of the wine has a negative impact on the number of cases sold.

3.5 Discussion of the Coefficients

Finally lets discuss what we found in the coefficient of the variables we will only discuss the variables that were used in at least one of the models. We will look at the models fit to the dummy variables first and then the full model.

Models fit to the Dummy Variables

In Table 10 below we have the variables used by each of the models along with their coefficients. We note that for all of the variables we maintain a similar sign so that all of them impact the number of cases of wine sold in a similar way. The linear coefficients have a different value but this makes sense in that the Poisson

and negative binomial models include a log effect while the coefficients in the linear model indicate a unit increase. We can also interpret that wines that are more acidic and have not been rated are less likely to be bought and wines that have an appealing label are more likely to be purchased.

Table 10: Coefficients for the models fit to the Dummy Variables

Variable	Poisson	Negative Binomial	Multiple Linear
VolatileAcidity	-0.0391	-0.0471	-0.1247
AcidIndex	-0.0902	-0.1184	-0.2311
LabelAppeal	0.2191	0.1995	0.0661
STARSMissing	-1.0413	-1.1339	-2.2253

Models fit to the Full Variables

In Table 11 we have the variables used by each of the models along with their coefficients. We note that the signs on all of the coefficients are consistent through all of the models we also see similar differences between the linear model and the other models as we saw in the dummy models. It is interesting to see that the Logistic model used to distinguish the zeros has different coefficients and values that are not used by any of the models. However once we get to the count regression portion of the zero-inflated Poisson it requires a much smaller variable set to distinguish the number of cases sold.

Table 11: Coefficients for the models fit to the Full Variables

Variable	Poisson	Negative Binomial	Multiple Linear	Zero-Inf Poisson
VolatileAcidity	-0.0391	-0.033	-0.0949	-
Chlorides	-0.0383	-0.0461	-0.1036	-
FreeSulfurDioxide	0.0001	0.0001	0.0003	-
TotalSulfurDioxide	-	-	0.0001	-
Density	-0.4656	-0.4964	-1.2399	-
Alcohol	0.0031	0.0029	0.0137	0.0057
AcidIndex	-0.0459	-0.0555	-0.1625	-0.017
LabelAppeal	0.1779	0.1844	0.6518	0.2087
STARS	0.1838	0.1956	0.2753	0.1129

4 Selecting a Model

We will now use the models that we have built to select a model and use this model to predict the number of cases of wine sold for our evaluation data set. To select between all of the models that we have generated we will use the Akaike Information Criterion (AIC) and the Mean Square Error (MSE). The selection criteria can be found in Table 12 below. From these criteria we can see that our best choice for a model is the zero-inflated Poisson model. It gives us an AIC better than the Poisson model and gives us a mean square error better than the multiple linear regression model. This model while slightly more complicated then the other gives us the best of both worlds and we will elect to move forward with this model.

Table 12: Evaluation Criteria for the Models

Model	AIC	MSE
Poisson Full	16138.06	7.74043
Poisson Dummy	32627.82	6.86914
Neg. Binomial Full	21553.61	7.72159
Neg. Binomial Dummy	38825.25	6.86822

Model	AIC	MSE
Multiple Linear Full	16682.74	1.35342
Multiple Linear Dummy	31817.54	2.10676
Zero-Inflated Poisson	15590.05	1.32552

Lets next take a look at our predicted values on the training set plotted against our actual values. We expect to see a correlation between these two observations in Figure 10 below.

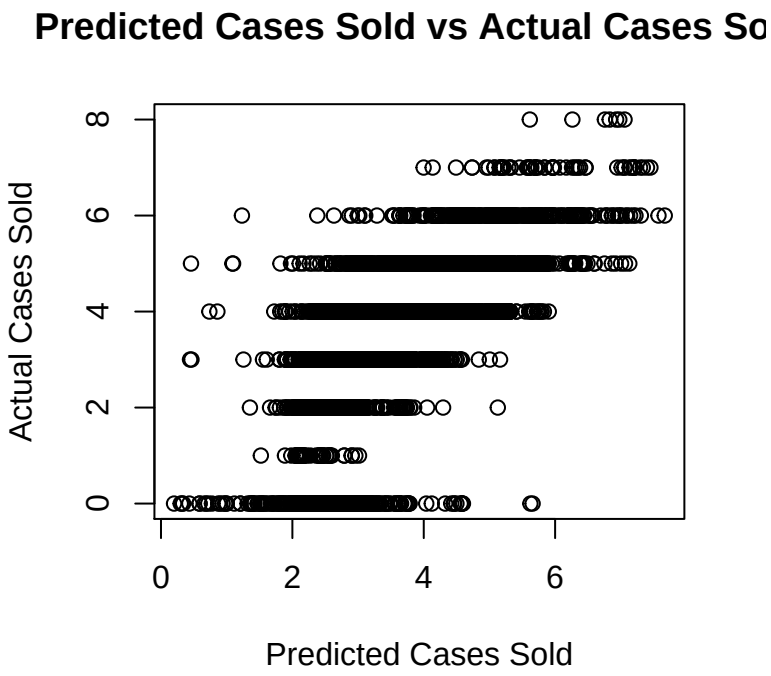


Figure 10: Predicted Values vs Actual Values for the Training Data Set

Now that we have selected our model and we see that it appears to be predicting the values we will use it to predict the number of cases of wine sold for our evaluation data. Table 13 and 14 below contain the first 6 and last 6 observations of the evaluation data set. We do note that there are some NA's in the set. These indicate that one of the required predictor variables was missing in that row of the observations.

Table 13: First 6 Observations

Observation	1	2	3	4	5	6
TARGET	NA	3.9	2.6	2.4	NA	5.7

Table 14: Last 6 Observations

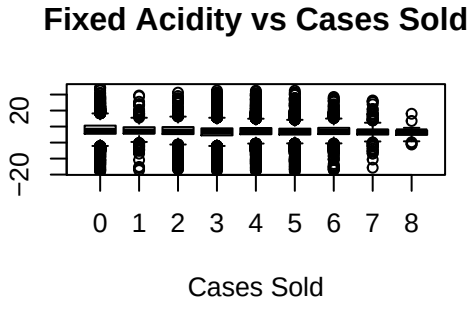
Observation	3330	3331	3332	3333	3334	3335
TARGET	2.4	3.6	5.7	3.6	NA	4.1

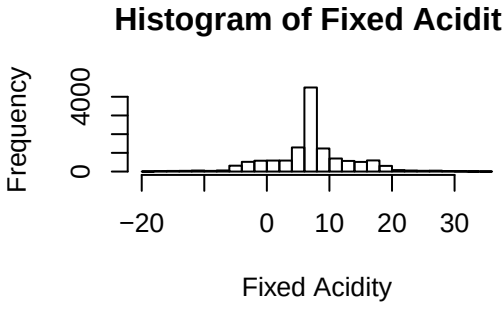
5 Concluding Remarks

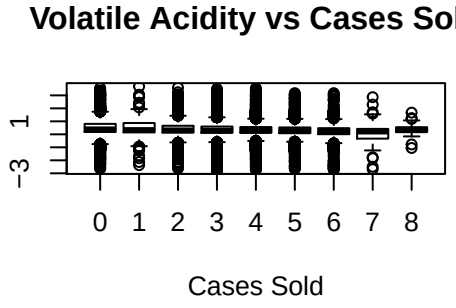
In this analysis we were able to generate a model that we feel does a reasonable job of predicting the number of cases of wine sold. From our analysis we would advise the winery to make sure that they have a wine with a reasonable high alcohol content that scores lower on the acidity index. The wine should have a very appealing label and should also be well rated by expert reviewers. As an aside as a non-wine person I do find it a little funny that in all of the models having an appealing label is great importance to selling wine. Ultimately it would appear that marketing is the most important thing after all.

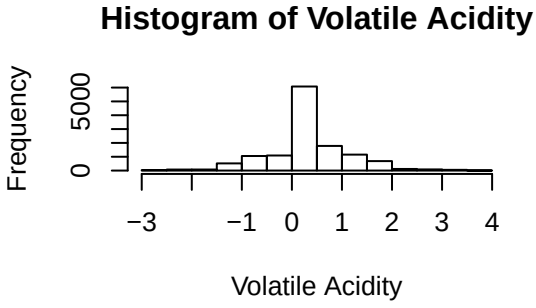
Appendix A: Plots of the Predictor Variables

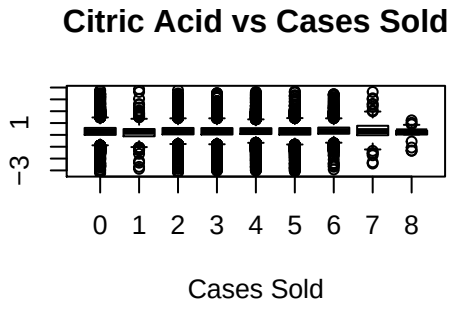
Plotting the Predictor Variables

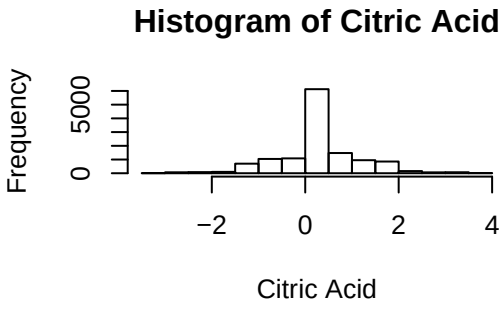


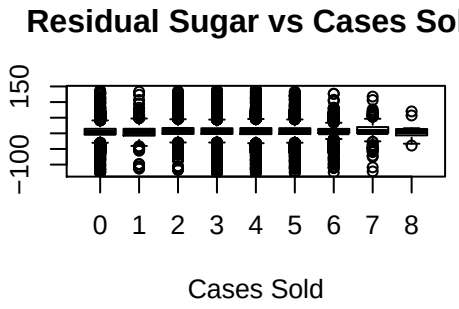


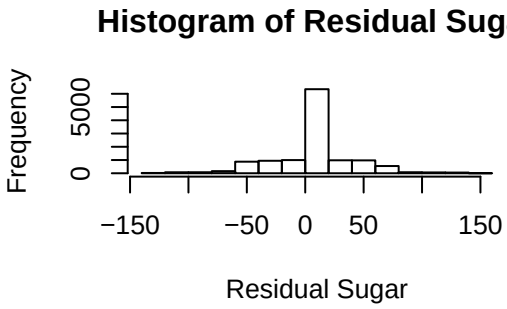


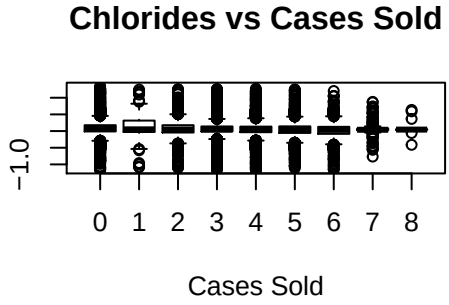


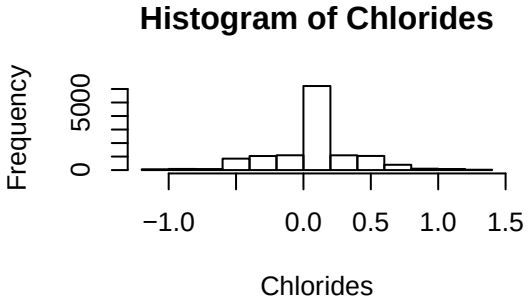


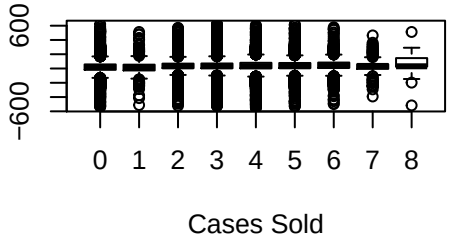


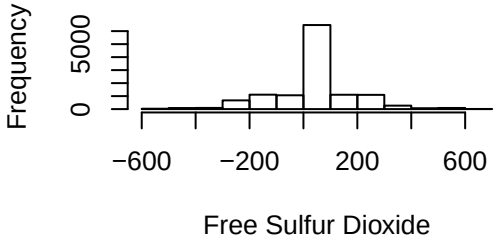


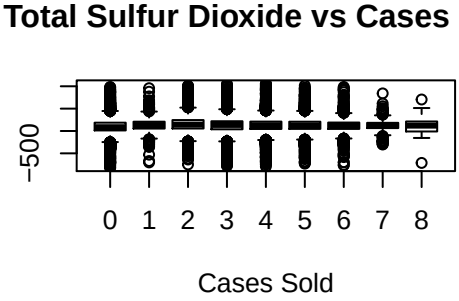


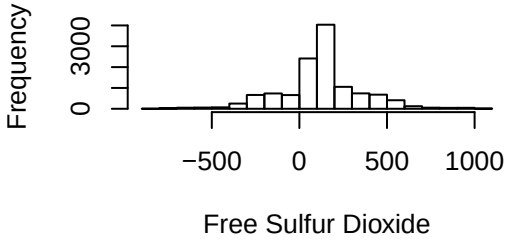

Chlorides vs Cases Sold
−1.0
0 1 2 3 4 5 6 7 8
Cases Sold

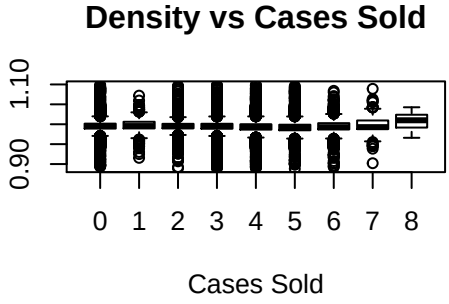

Histogram of Chlorides
Frequency
0 5000
−1.0 0.0 0.5 1.0 1.5
Chlorides

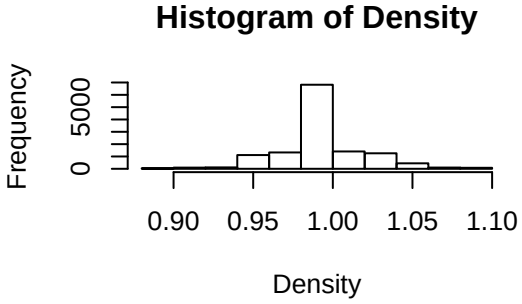

Free Sulfur Dioxide vs Cases Sold
−600 600
0 1 2 3 4 5 6 7 8
Cases Sold

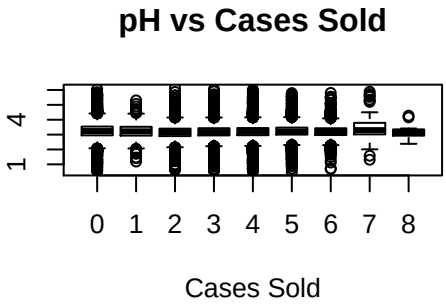

Histogram of Free Sulfur Dioxide
Frequency
0 5000
−600 −200 200 600
Free Sulfur Dioxide

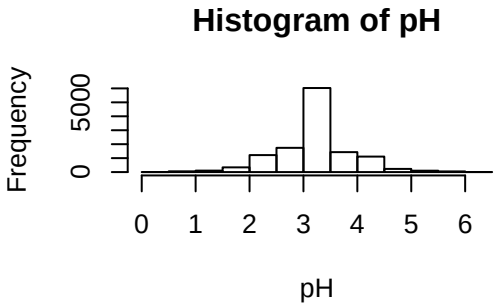

Total Sulfur Dioxide vs Cases Sold
−500
0 1 2 3 4 5 6 7 8
Cases Sold

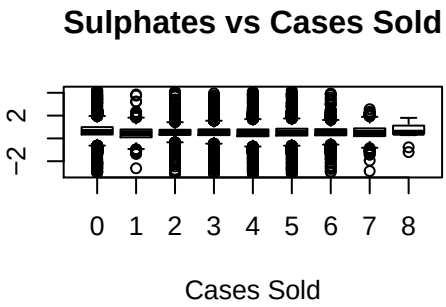

Histogram of Total Sulfur Dioxide
Frequency
0 3000
−500 0 500 1000
Free Sulfur Dioxide

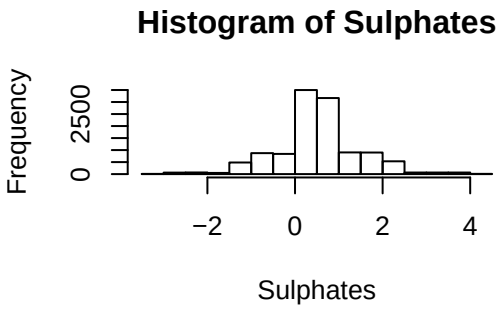

Density vs Cases Sold
0.90 1.10
0 1 2 3 4 5 6 7 8
Cases Sold

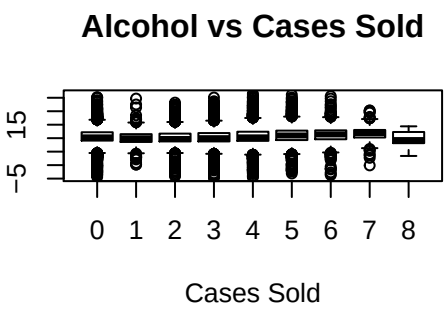

Histogram of Density
Frequency
0 5000
0.90 0.95 1.00 1.05 1.10
Density

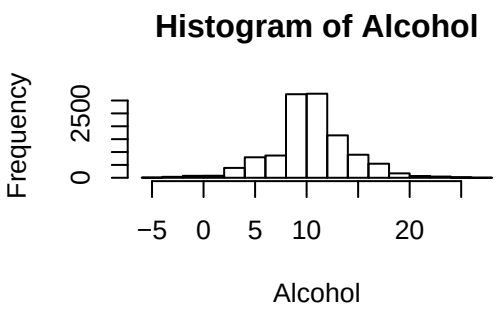

pH vs Cases Sold
1 4
0 1 2 3 4 5 6 7 8
Cases Sold

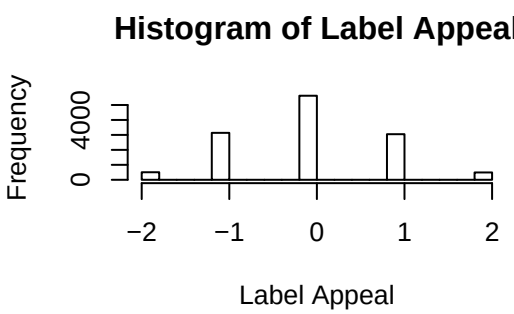

Histogram of pH
Frequency
0 5000
0 1 2 3 4 5 6
pH

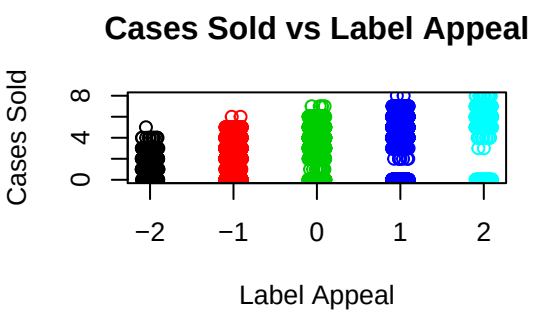

Sulphates vs Cases Sold
−2 2
0 1 2 3 4 5 6 7 8
Cases Sold

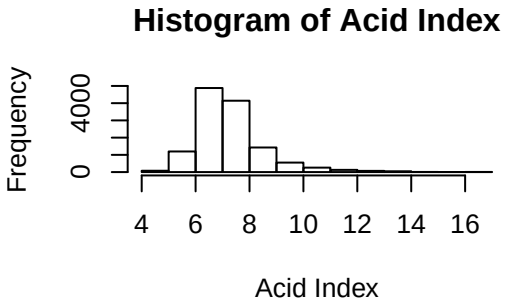

Histogram of Sulphates
Frequency
0 2500
−2 0 2 4
Sulphates

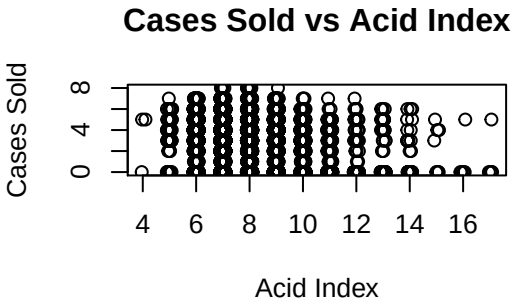

Alcohol vs Cases Sold
−5 15
0 1 2 3 4 5 6 7 8
Cases Sold

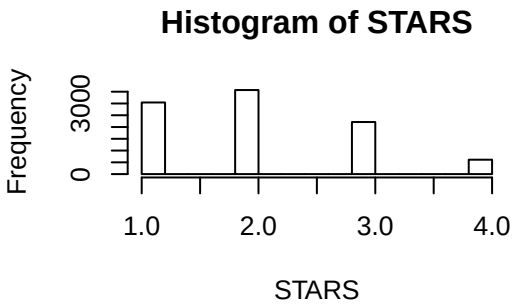

Histogram of Alcohol
Frequency
0 2500
−5 0 5 10 20
Alcohol

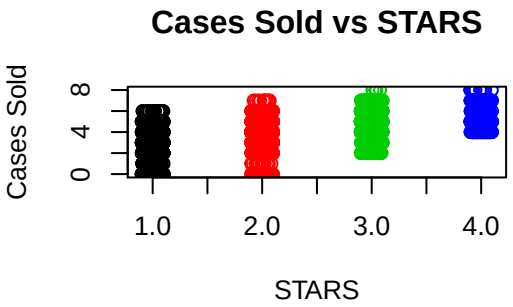

Histogram of Label Appeal
Frequency
0 4000
−2 −1 0 1 2
Label Appeal


Cases Sold vs Label Appeal
Cases Sold
0 4 8
−2 −1 0 1 2
Label Appeal

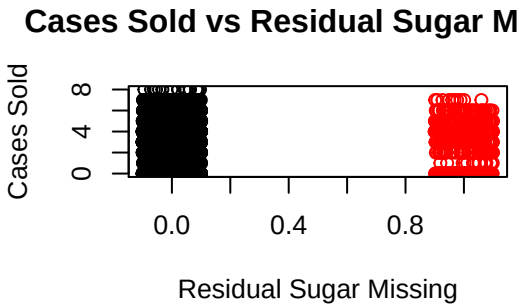

Histogram of Acid Index
Frequency
0
4000
4
6
8
10
12
14
16
Acid Index

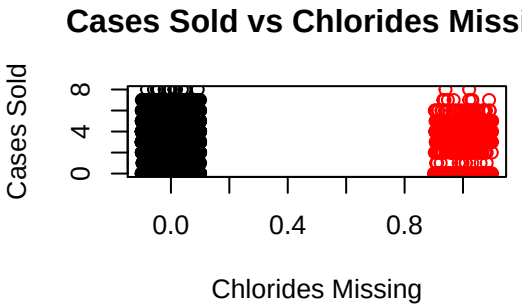

Cases Sold vs Acid Index
Cases Sold
0
4
8
4
6
8
10
12
14
16
Acid Index

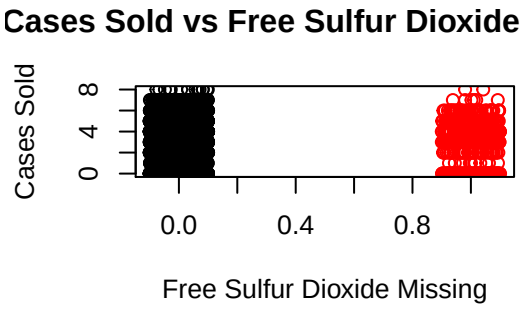

Histogram of STARS
Frequency
0
3000
1.0
2.0
3.0
4.0
STARS

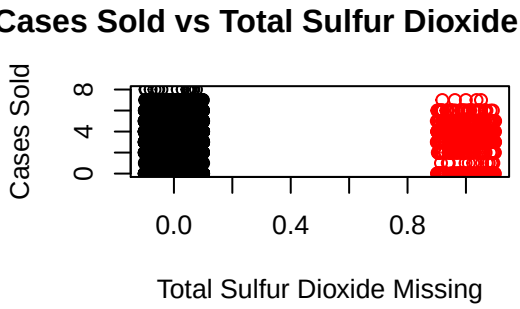

Cases Sold vs STARS
Cases Sold
0
4
8
1.0
2.0
3.0
4.0
STARS

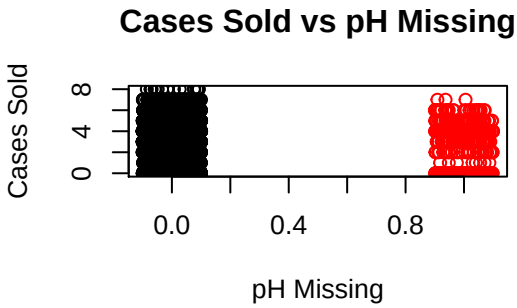
Ploting the Dummy Variables











Appendix B: R Code

The R code for the above analysis is contained below.

```
#### Loading Data ####
wine <- read.csv("DATA621/wine-training-data.csv")
wine.eval <- read.csv("DATA621/wine-evaluation-data.csv")
library(knitr)
summary(wine)

# Creating a simplified data frame and dropping the index#
wine <- wine[,2:16]
wine.simple <- wine

summary(wine.simple)

#### Adding Dummy Variables for Missing Values ####
wine <- within(wine, {
  ResidualSugarMissing <- ifelse(is.na(ResidualSugar),1,0)
  ChloridesMissing <- ifelse(is.na(Chlorides),1,0)
  FreeSulfurDioxideMissing <- ifelse(is.na(FreeSulfurDioxide),1,0)
  TotalSulfurDioxideMissing <- ifelse(is.na(TotalSulfurDioxide),1,0)
  pHMissing <- ifelse(is.na(pH),1,0)
  SulphatesMissing <- ifelse(is.na(Sulphates),1,0)
  AlcoholMissing <- ifelse(is.na(Alcohol),1,0)
  STARSMissing <- ifelse(is.na(STARS),1,0)
})

wine.dummy <- wine[,c(1:4,9,13,14,16:23)]
summary(wine.dummy)

#### Examining the Data ####
# FixedAcidity Variable #
boxplot(FixedAcidity ~ TARGET, data = wine, main = "Fixed Acidity vs Cases Sold",
        xlab = "Cases Sold") # Wide variance all medians are near equal
hist(wine$FixedAcidity, breaks = 25, main = "Histogram of Fixed Acidity",
     xlab = "Fixed Acidity") # Histogram tall peak with wide shoulders

# VolatileAcidity Variable #
boxplot(VolatileAcidity ~ TARGET, data = wine,
        main = "Volatile Acidity vs Cases Sold",
        xlab = "Cases Sold") # Wide variance might be a slight difference in medians
hist(wine$VolatileAcidity, main = "Histogram of Volatile Acidity",
     xlab = "Volatile Acidity") # Tall peak with wide shoulders

# CitricAcid Variable #
boxplot(CitricAcid ~ TARGET, data = wine,
        main = "Citric Acid vs Cases Sold",
        xlab = "Cases Sold") # Wide variance with nearly equal medians
hist(wine$CitricAcid, main = "Histogram of Citric Acid",
     xlab = "Citric Acid") # Tall peak with wide shoulders

# Residual Sugar Variable #
boxplot(ResidualSugar ~ TARGET, data = wine,
```

```
        main = "Residual Sugar vs Cases Sold",
        xlab = "Cases Sold") # Wide variance with little chnge in medians
hist(wine$ResidualSugar, main = "Histogram of Residual Sugar",
     xlab = "Residual Sugar") # Tall peak with wide shoulders

# Chlorides Variable #
boxplot(Chlorides ~ TARGET, data = wine,
        main = "Chlorides vs Cases Sold",
        xlab = "Cases Sold") # Wide variance with equal medians, less variance on the
                             # larger orders
hist(wine$Chlorides, main = "Histogram of Chlorides",
     xlab = "Chlorides") # Tall peak with wide shoulders

# FreeSulferDioxide Variable #
boxplot(FreeSulfurDioxide ~ TARGET, data = wine,
        main = "Free Sulfur Dioxide vs Cases Sold",
        xlab = "Cases Sold") # Wide variance with slight change to median
hist(wine$FreeSulfurDioxide, main = "Histogram of Free Sulfur Dioxide",
     xlab = "Free Sulfur Dioxide") # Tall peak with wide shoulders

# TotalSulfurDioxide Variable #
boxplot(TotalSulfurDioxide ~ TARGET, data = wine,
        main = "Total Sulfur Dioxide vs Cases Sold",
        xlab = "Cases Sold") # Wide variance with slight chnge to medians
hist(wine$TotalSulfurDioxide, main = "Histogram of Total Sulfur Dioxide",
     xlab = "Free Sulfur Dioxide") # Tall peak slight more normal

# Density Variable #
boxplot(Density ~ TARGET, data = wine,
        main = "Density vs Cases Sold",
        xlab = "Cases Sold") # Wide Variance, more dense on higher cases
hist(wine$Density, main = "Histogram of Density",
     xlab = "Density") # Tall peak with large shoulders

# pH Variable #
boxplot(pH ~ TARGET, data = wine,
        main = "pH vs Cases Sold",
        xlab = "Cases Sold") # Wide variance 7 case higher then rest
hist(wine$pH, main = "Histogram of pH",
     xlab = "pH") # Tall peak, good distribution

# Sulphates Variable #
boxplot(Sulphates ~ TARGET, data = wine,
        main = "Sulphates vs Cases Sold",
        xlab = "Cases Sold") # Wide Variance and the 8 case higher
hist(wine$Sulphates, main = "Histogram of Sulphates",
     xlab = "Sulphates") # Tall peak with wide shoulders

# Alcohol Variable #
boxplot(Alcohol ~ TARGET, data = wine,
        main = "Alcohol vs Cases Sold",
        xlab = "Cases Sold") # Wide variance may be predictive,
hist(wine$Alcohol, main = "Histogram of Alcohol",
```

```
     xlab = "Alcohol") # Tall peak good distribution

# Label Appeal Variable # Has effect on cases sold
boxplot(LabelAppeal ~ TARGET, data = wine,
        main = "Label Appeal vs Cases Sold",
        xlab = "Cases Sold") # Categorical, very predictive
hist(wine$LabelAppeal, main = "Histogram of Label Appeal",
     xlab = "Label Appeal")

plot(wine$TARGET ~ jitter(wine$LabelAppeal, .5),
     col = as.factor(wine$LabelAppeal),
     main = "Cases Sold vs Label Appeal",
     xlab = "Label Appeal",
     ylab = "Cases Sold") # Might be better than the boxplot

# AcidIndex Variable #
boxplot(AcidIndex ~ TARGET, data = wine,
        main = "Acid Index vs Cases Sold",
        xlab = "Cases Sold") # Acid index, categorical may be predictive
hist(wine$AcidIndex, main = "Histogram of Acid Index",
     xlab = "Acid Index") # Log normal distribution?

plot(wine$TARGET ~ jitter(wine$AcidIndex, .5),
     main = "Cases Sold vs Acid Index",
     xlab = "Acid Index",
     ylab = "Cases Sold") # Might be better than the boxplot

# STARS Variable # High Stars are more likely to be sold
boxplot(STARS ~ TARGET, data = wine,
        main = "STARS vs Cases Sold",
        xlab = "Cases Sold") # Categorical, highly predictive
hist(wine$STARS, main = "Histogram of STARS",
     xlab = "STARS")

plot(wine$TARGET ~ jitter(wine$STARS, .5),
     col = as.factor(wine$STARS),
     main = "Cases Sold vs STARS",
     xlab = "STARS",
     ylab = "Cases Sold") # Might be better than the boxplot

# Plotting the Dummy Variables #
# Residual Sugar Missing #
plot(wine$TARGET ~ jitter(wine$ResidualSugarMissing, .5),
     col = as.factor(wine$ResidualSugarMissing),
     main = "Cases Sold vs Residual Sugar Missing",
     xlab = "Residual Sugar Missing",
     ylab = "Cases Sold") # The more missing values the lower number sold

# Chlorides Missing #
plot(wine$TARGET ~ jitter(wine$ChloridesMissing, .5),
     col = as.factor(wine$ChloridesMissing),
     main = "Cases Sold vs Chlorides Missing",
     xlab = "Chlorides Missing",
```

```
     ylab = "Cases Sold") # The more missing values the lower number sold

# Free Sulfur Dioxide Missing #
plot(wine$TARGET ~ jitter(wine$FreeSulfurDioxideMissing, .5),
     col = as.factor(wine$FreeSulfurDioxideMissing),
     main = "Cases Sold vs Free Sulfur Dioxide Missing",
     xlab = "Free Sulfur Dioxide Missing",
     ylab = "Cases Sold") # The more missing values the lower number sold

# Total Sulfur Dioxide Missing #
plot(wine$TARGET ~ jitter(wine$TotalSulfurDioxideMissing, .5),
     col = as.factor(wine$TotalSulfurDioxideMissing),
     main = "Cases Sold vs Total Sulfur Dioxide Missing",
     xlab = "Total Sulfur Dioxide Missing",
     ylab = "Cases Sold") # The more missing values the lower number sold

# pH Missing #
plot(wine$TARGET ~ jitter(wine$pHMissing, .5),
     col = as.factor(wine$pHMissing),
     main = "Cases Sold vs pH Missing",
     xlab = "pH Missing",
     ylab = "Cases Sold") # The more missing values the lower number sold

# Sulphates Missing #
plot(wine$TARGET ~ jitter(wine$SulphatesMissing, .5),
     col = as.factor(wine$SulphatesMissing),
     main = "Cases Sold vs Free Sulphates",
     xlab = "Sulphates Missing",
     ylab = "Cases Sold") # The more missing values the lower number sold

# Alcohol Missing #
plot(wine$TARGET ~ jitter(wine$AlcoholMissing, .5),
     col = as.factor(wine$AlcoholMissing),
     main = "Cases Sold vs Alcohol Missing",
     xlab = "Alcohol Missing",
     ylab = "Cases Sold") # The more missing values the lower number sold

# STARS Missing #
plot(wine$TARGET ~ jitter(wine$STARSMissing, .5),
     col = as.factor(wine$STARSMissing),
     main = "Cases Sold vs STARS Missing",
     xlab = "STARS Missing",
     ylab = "Cases Sold") # The more missing values the lower number sold

summary(wine)

#### Creating a Training and Test Data sets ####
n <- dim(wine)[1]
set.seed(101010) # 42!

test <- sample(n, round(n * .3))
wine.simple.train <- wine.simple[-test,]
wine.simple.test <- wine.simple[test,]
```

```
wine.dummy.train <- wine.dummy[-test,]
wine.dummy.test <- wine.dummy[test,]

#### Poisson Regression ####
# Base Assumption mean == variance
mean(wine$TARGET)
sd(wine$TARGET)^2
mean(wine$TARGET) / sd(wine$TARGET)^2

# Mean and Variance differ may be solvable with dispersion or negative binomial.
hist(wine$TARGET, breaks = 9)

# Fitting the regression to all data points without dummy variables #
wine_poisson1 <- glm(TARGET ~ ., family = "poisson", data = na.omit(wine.simple.train))
summary(wine_poisson1)

coefficients(wine_poisson1)

# Using Robust Residuals to identify significant variables #
library(sandwich)
cov.wine_poisson1 <- vcovHC(wine_poisson1, type="HC0")
std.err <- sqrt(diag(cov.wine_poisson1))
r.est <- cbind(Estimate= coef(wine_poisson1), "Robust SE" = std.err,
               "Pr(>abs(z))" = 2 * pnorm(abs(coef(wine_poisson1)/std.err),
                                         lower.tail=FALSE),
               LL = coef(wine_poisson1) - 1.96 * std.err,
               UL = coef(wine_poisson1) + 1.96 * std.err)

kable(r.est)

step(wine_poisson1)
# Robust Residuals suggest that the following are significant
# Intercept, VolatileAcidity, Chlorides, FreeSulfurDioxide, Density, Alcohol,
# LabelAppeal, AcidIndex, STARS

# Fitting the reduced model
wine_poisson2 <- update(wine_poisson1, .~. - FixedAcidity - CitricAcid -
                          ResidualSugar - TotalSulfurDioxide - pH -
                          Sulphates)

summary(wine_poisson2)

# Constructing New set of robust residuals
cov.wine_poisson2 <- vcovHC(wine_poisson2, type="HC0")
std.err <- sqrt(diag(cov.wine_poisson2))
r.est <- cbind(Estimate= coef(wine_poisson2), "Robust SE" = std.err,
               "Pr(>abs(z))" = 2 * pnorm(abs(coef(wine_poisson2)/std.err),
                                         lower.tail=FALSE),
               LL = coef(wine_poisson2) - 1.96 * std.err,
               UL = coef(wine_poisson2) + 1.96 * std.err)

kable(r.est)
par(mfrow=c(2,2))
```

```
plot(wine_poisson2)

AIC(wine_poisson2) # AIC 16138.06
mean((wine.simple.test$TARGET - predict(wine_poisson2, wine.simple.test))^2,
     na.rm = TRUE) # MSE = 7.740432


# Fitting the regression to the dummy variables with the primary variables missing
wine_pdummy1 <- glm(TARGET ~ ., family = "poisson", data = na.omit(wine.dummy.train))
summary(wine_pdummy1)

# Constructing a set of robust residuals
cov.wine_pdummy1 <- vcovHC(wine_pdummy1, type="HC0")
std.err <- sqrt(diag(cov.wine_pdummy1))
r.est <- cbind(Estimate= coef(wine_pdummy1), "Robust SE" = std.err,
               "Pr(>abs(z))" = 2 * pnorm(abs(coef(wine_pdummy1)/std.err),
                                         lower.tail=FALSE),
               LL = coef(wine_pdummy1) - 1.96 * std.err,
               UL = coef(wine_pdummy1) + 1.96 * std.err)

kable(r.est)

drop1(wine_pdummy1, test = "F") #Checking the significant predictors.
step(wine_pdummy1) # The stepwise selection suggests the following variables
# VolatileAcidity, AcidIndex, LabelAppeal, STARSMissing

wine_pdummy2 <- glm(TARGET ~ VolatileAcidity + AcidIndex + LabelAppeal +
                      STARSMissing,
                    family = "poisson", data = na.omit(wine.dummy.train))
summary(wine_pdummy2)

# Robust Residuals suggest the following variables
cov.wine_pdummy2 <- vcovHC(wine_pdummy2, type="HC0")
std.err <- sqrt(diag(cov.wine_pdummy2))
r.est <- cbind(Estimate= coef(wine_pdummy2), "Robust SE" = std.err,
               "Pr(>abs(z))" = 2 * pnorm(abs(coef(wine_pdummy2)/std.err),
                                         lower.tail=FALSE),
               LL = coef(wine_pdummy2) - 1.96 * std.err,
               UL = coef(wine_pdummy2) + 1.96 * std.err)
kable(r.est)

AIC(wine_pdummy2) #32627.82
mean((wine.dummy.test$TARGET - predict(wine_pdummy2, wine.dummy.test))^2) # 6.869139


#### Building the Negative Binomial Models ####
# Negative Model on full simplified data set #
wine.na.train <- na.omit(wine.simple.train)
library(MASS)
wine_nb1 <- glm(TARGET ~ ., negative.binomial(1), wine.na.train)
summary(wine_nb1)

step(wine_nb1, na.rm = TRUE)
```

```
# Step reccomends the following variables: VolatileAcidity, Chlorides,
# FreeSulfurDioxide, TotalSulfurDioxide, Density, ph, sulphates, Alcohol,
# LabelAppeal, AcidIndex, STARS

wine_nb2 <- glm(TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide +
                  Density + Alcohol +
                  LabelAppeal + AcidIndex + STARS,
                negative.binomial(1), wine.na.train)
summary(wine_nb2)

# Robust Residuals suggest the following variables
cov.wine_nb2 <- vcovHC(wine_nb2, type="HC0")
std.err <- sqrt(diag(cov.wine_nb2))
r.est <- cbind(Estimate= coef(wine_nb2), "Robust SE" = std.err,
               "Pr(>abs(z))" = 2 * pnorm(abs(coef(wine_nb2)/std.err),
                                         lower.tail=FALSE),
               LL = coef(wine_nb2) - 1.96 * std.err,
               UL = coef(wine_nb2) + 1.96 * std.err)
kable(r.est)

AIC(wine_nb2) # 21553.61
mean((wine.simple.test$TARGET - predict(wine_nb2, wine.simple.test))^2,
     na.rm = TRUE) # MSE = 7.721589

# Fitting the model to the dummy data
wine_nbdummy1 <- glm(TARGET ~ ., family = negative.binomial(1),
                     data = wine.dummy.train)
summary(wine_nbdummy1)

# Robust Residuals suggest the following variables
cov.wine_nbdummy1 <- vcovHC(wine_nbdummy1, type="HC0")
std.err <- sqrt(diag(cov.wine_nbdummy1))
r.est <- cbind(Estimate= coef(wine_nbdummy1), "Robust SE" = std.err,
               "Pr(>abs(z))" = 2 * pnorm(abs(coef(wine_nbdummy1)/std.err),
                                         lower.tail=FALSE),
               LL = coef(wine_nbdummy1) - 1.96 * std.err,
               UL = coef(wine_nbdummy1) + 1.96 * std.err)
kable(r.est)
# The robust residuals suggest that we use: VolatileAcidity, LabelAppeal, AcidIndex,
# STARSMissing

step(wine_nbdummy1)
# Step Reccomends VolatileAcidity, LabelAppeal, AcidIndex, STARSMissing, pHMissing

wine_nbdummy2 <- glm(TARGET ~ VolatileAcidity + LabelAppeal + STARSMissing +
                       AcidIndex, family = negative.binomial(1),
                     data = wine.dummy.train)
summary(wine_nbdummy2)

# Robust Residuals suggest the following variables
cov.wine_nbdummy2 <- vcovHC(wine_nbdummy2, type="HC0")
std.err <- sqrt(diag(cov.wine_nbdummy2))
r.est <- cbind(Estimate= coef(wine_nbdummy2), "Robust SE" = std.err,
```

```
               "Pr(>abs(z))" = 2 * pnorm(abs(coef(wine_nbdummy2)/std.err),
                                         lower.tail=FALSE),
               LL = coef(wine_nbdummy2) - 1.96 * std.err,
               UL = coef(wine_nbdummy2) + 1.96 * std.err)
kable(r.est)

AIC(wine_nbdummy2) # 38825.25
mean((wine.dummy.test$TARGET - predict(wine_nbdummy2, wine.dummy.test))^2,
     na.rm = TRUE) # MSE = 6.86822

#### Fitting a Linear Model ####
# Fitting the linear model to the simple test data #
wine_lm1 <- lm(TARGET ~ ., data = wine.na.train)
summary(wine_lm1)
plot(wine_lm1)
# Issues with the residuals

step(wine_lm1)


# stepwise selection reccomends the following variables: Sulphates,
# TotalSulfurDioxide, FreeSulfurDioxide, Chlorides, Density, Alcohol,
# VolatileAcidity, AcidIndex, LabelAppeal, STARS

wine_lm2 <- lm(TARGET ~ TotalSulfurDioxide + FreeSulfurDioxide + Chlorides +
                 Density + Alcohol + VolatileAcidity + AcidIndex + LabelAppeal +
                 STARS, data = wine.simple.train)
summary(wine_lm2)

# Robust Residuals suggest the following variables
cov.wine_lm2 <- vcovHC(wine_lm2, type="HC0")
std.err <- sqrt(diag(cov.wine_lm2))
r.est <- cbind(Estimate= coef(wine_lm2), "Robust SE" = std.err,
               "Pr(>abs(z))" = 2 * pnorm(abs(coef(wine_lm2)/std.err),
                                         lower.tail=FALSE),
               LL = coef(wine_lm2) - 1.96 * std.err,
               UL = coef(wine_lm2) + 1.96 * std.err)
kable(r.est)

plot(wine_lm2)

AIC(wine_lm2) # 16682.74
mean((wine.simple.test$TARGET - predict(wine_lm2, wine.simple.test))^2,
     na.rm = TRUE) # 1.353.417

# Fitting a linear model to the dummy data set #
wine_lmdummy1 <- lm(TARGET ~ ., data = wine.dummy.train)
summary(wine_lmdummy1)
plot(wine_lmdummy1)
#Paterns in the Residuals

step(wine_lmdummy1)
# Step reccomends that we use: VolatileAcidity, AcidIndex, LabelAppeal, STARSMissing
```

```r
wine_lmdummy2 <- lm(TARGET ~ VolatileAcidity + AcidIndex + LabelAppeal +
                      STARSMissing, data = wine.dummy.train)
summary(wine_lmdummy2)

# Robust Residuals suggest the following variables
cov.wine_lmdummy2 <- vcovHC(wine_lmdummy2, type="HC0")
std.err <- sqrt(diag(cov.wine_lmdummy2))
r.est <- cbind(Estimate= coef(wine_lmdummy2), "Robust SE" = std.err,
               "Pr(>abs(z))" = 2 * pnorm(abs(coef(wine_lmdummy2)/std.err),
                                         lower.tail=FALSE),
               LL = coef(wine_lmdummy2) - 1.96 * std.err,
               UL = coef(wine_lmdummy2) + 1.96 * std.err)
kable(r.est)

plot(wine_lmdummy2)
AIC(wine_lmdummy2) # 31817.54
mean((wine.dummy.test$TARGET - predict(wine_lmdummy2, wine.dummy.test))^2,
     na.rm = TRUE) # 2.106764

#### Fitting a zero inflated poisson ####
library(pscl)
wine_zip <- zeroinfl(TARGET ~ ., data = wine.na.train)
summary(wine_zip)

wine_zip1 <- zeroinfl(TARGET ~ Alcohol + LabelAppeal + AcidIndex + STARS |
                        VolatileAcidity + FreeSulfurDioxide +
                        TotalSulfurDioxide + Sulphates + Alcohol +
                        LabelAppeal + AcidIndex + STARS, data = wine.na.train)

summary(wine_zip1)
coefficients(wine_zip1)
kable(summary(wine_zip1)$coefficients$count)


AIC(wine_zip1) #15590.05
mean((wine.simple.test$TARGET - predict(wine_zip1, wine.simple.test))^2,
     na.rm = TRUE) # 1.325523

# Favorite Model!

pred.train <- predict(wine_zip1, wine.na.train)
plot(pred.train, wine.na.train$TARGET,
     main = "Predicted Cases Sold vs Actual Cases Sold",
     xlab = "Predicted Cases Sold",
     ylab = "Actual Cases Sold")

pred.eval <- predict(wine_zip1, wine.eval)
head(pred)
tail(pred)
```