Predict 411 Unit3 Wine Sales

Jennifer Wanat

Contents

Introduction

In this project our objective is to be able to provide a regression model to predict the number of wine cases ordered based upon wine characteristics. The data set contains records of commercially available wines and their physicochemical properties such as acidity, sulfur dioxide, sulphates and residual sugar. In order to build a model to provide a prediction of the number of wine cases ordered, an exploratory data analysis (EDA) of the data set was completed prior to this analysis, which allowed for an evaluation and selection of the most promising predictor variables.

The generated model summaries and parameters were provided, and assessed for predictive accuracy by examing model selection criteria. Then a model was selected, and used to score new data to predict the number of wine cases ordered. The analysis was conducted with the R programming language.

Data

The data was entered in an Excel spreadsheet and saved as a comma separated value file (.csv). The .csv file was opened in R and saved into a data frame called `wine`. The data frame contains 12795 observations of 16 variables for wine. Refer to the data dictionary for variable definition and theoretical effect (positive or negative impact, if known) on wine sales. The variable *INDEX* is an identification variable and was not used for modeling purposes. An EDA was conducted on the remaining 15 variables.

A second data set was entered in an Excel spreadsheet and saved as a comma separated value file (.csv). The .csv file was opened in R and saved into a data frame called `wine_test`. The `wine_test` data frame contains 3335 observations of the same 16 variables for wine. The `wine_test` set does not contain values for the variable, TARGET, which is the number of wine cases purchased. The best model selected from the `wine` set will be used to score the `wine_test` data.

VARIABLE.NAME	DEFINITION	THEORETICAL.EFFECT
INDEX	Identification Variable (do not use)	None
TARGET	Number of Cases Purchased	None
AcidIndex	Proprietary method of testing total acidity of wine by using a weighted average	
Alcohol	Alcohol Content	
Chlorides	Chloride content of wine	
CitricAcid	Citric Acid Content	
Density	Density of Wine	
FixedAcidity	Fixed Acidity of Wine	
FreeSulfurDioxide	Sulfur Dioxide content of wine	
LabelAppeal	Marketing Score indicating the appeal of label design for consumers. High numbers suggest customers like the label design. Negative numbers suggest customes don't like the design.	Many consumers purchase based on the visual appeal of the wine label design. Higher numbers suggest better sales.
ResidualSugar	Residual Sugar of wine	
STARS	Wine rating by a team of experts. 4 Stars = Excellent, 1 Star = Poor	A high number of stars suggests high sales
Sulphates	Sulfate conten of wine	
TotalSulfurDioxide	Total Sulfur Dioxide of Wine	
VolatileAcidity	Volatile Acid content of wine	
pH	pH of wine	

Figure 1:

Section 1: Data Exploration

A basic statistical summary of the variables was prepared. The descriptive summary includes the number of observations (n), mean, standard deviation (sd), median, trimmed mean (trimmed), median absolute deviation from the mean (mad), minimum value (min), maximum value (max), the difference between the minimum and maximum values (range), skewness (skew), kurtosis, and standard error (se). Eight variables were noted to have missing values (NA), as determined by a n value less than 12795, and were examined further. These variables were: Residual Sugar, Chlorides, Free Sulfur Dioxide, Total Sulfur Dioxide, pH, Sulphates, Alcohol, and STARS.

The number and percentage of missing values for these variables is examined in the table below. All variables have 9.46% or less missing values, except for STARS which was missing 26.25% of the total values. The missing values will need to be imputed in order to build a logistic regression model. This will be discussed in Section 2.

The quantiles were examined for all variables except STARS, TARGET and INDEX. There is some difference between the 0th and 1st, and the 99th and 100th percent quantiles, but the skewness of the variables was minimal. Variables would not undergo a truncation transformation.

A histogram of the TARGET variable was prepared. There are a large number of wines with a zero TARGET value. TARGET values greater than zero displayed a normal distribution, with most wines having a TARGET value of 4.

A dot plot of TARGET versus Label Appeal was prepared. The data points were colored according to their STARS ranking and jittered to help with visualization. Most wines that were missing (NA) STARS had a zero TARGET value. Generally, the TARGET value increased with increasing STARS and Label Appeal.

Boxplots of TARGET versus STARS and Label Appeal versus TARGET were prepared. The resulting plots supported the conclusions drawn from the dot plot.

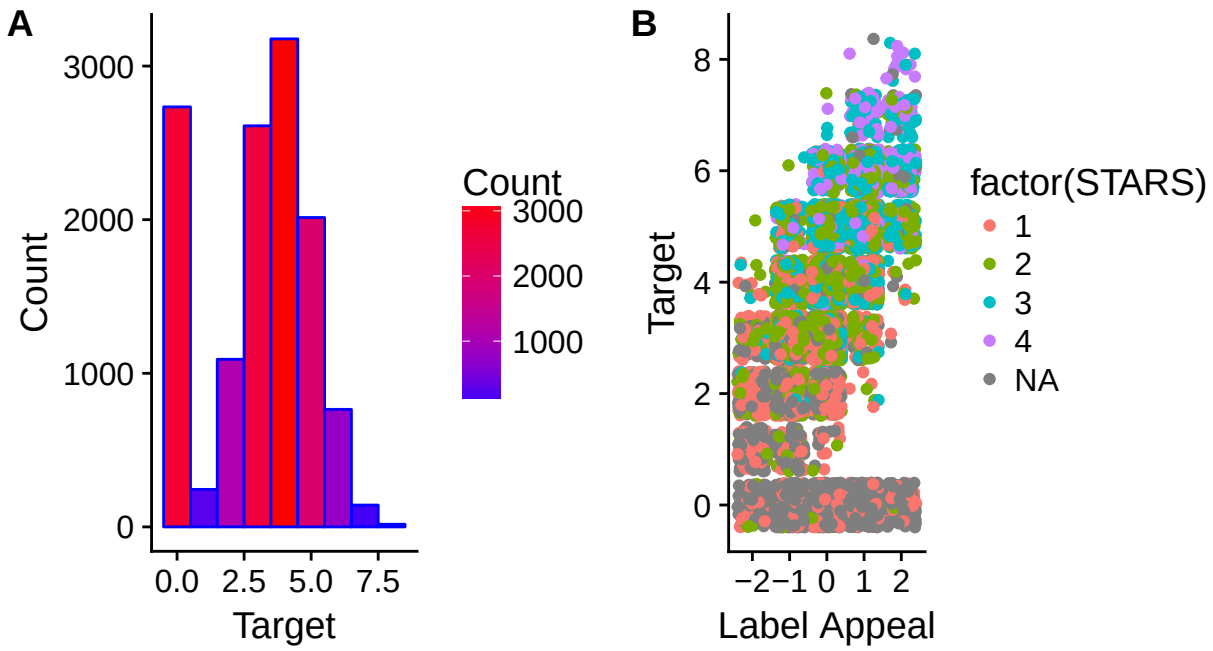


Figure 2: A: Histogram of TARGET. B: Dot plot of TARGET, label appeal and STARS.

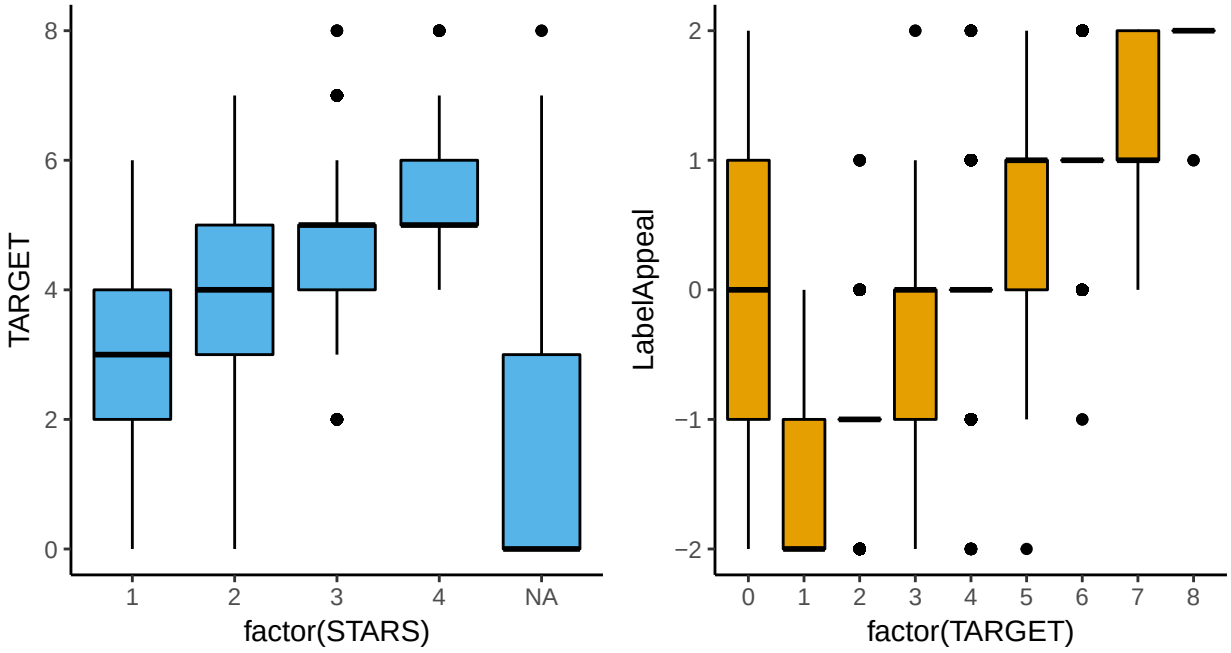


Figure 3: Boxplot of Label Appeal and TARGET.

Table 1: Percentage of Missing Values for Affected Variables.

	Number of missing values	Percent of missing values
Residual Sugar	616	4.81
Chlorides	638	4.99
Free Sulfur Dioxide	647	5.06
Total Sulfur Dioxide	682	5.33
pH	395	3.09
Sulphates	1210	9.46
Alcohol	653	5.10
STARS	3359	26.25

The frequency, proportion and cumulative proportion of the label appeal and STARS was examined. Wines were approximately normally distributed in the label appeal. Of the wines that had STARS ranking, most were ranked with 2 STARS (3570) and 4 STARS had the least number of wines (612). Wines with a higher STARS ranking generally had a higher label appeal.

Table 2: Frequency, Proportion and Cumulative Proportion of Label Appeal.

X	Freq	Prop	CumProp
-2	504	0.0394	0.0394
-1	3136	0.2451	0.2845
0	5617	0.4390	0.7235
1	3048	0.2382	0.9617
2	490	0.0383	1.0000

Table 3: Frequency, Proportion and Cumulative Proportion of STARS.

X	Freq	Prop	CumProp
1	3042	0.3224	0.3224
2	3570	0.3783	0.7007
3	2212	0.2344	0.9351
4	612	0.0649	1.0000

Table 4: Frequency of Wines According to STARS and Label Appeal.

	1	2	3	4	Sum
-2	203	70	21	0	294
-1	1008	849	262	29	2148
0	1334	1669	1011	192	4206
1	448	873	766	310	2397
2	49	109	152	81	391
Sum	3042	3570	2212	612	9436

Table 5: Quantiles of Variables of Interest.

	0%	1%	5%	10%	25%	50%	75%	90%	95%	99%	100%
Fixed Acidity	-18.1	-10.9	-3.6	-1.2	5.2	6.9	9.5	15.6	17.8	24.3	34.4
Volatile Acidity	-2.8	-1.9	-1.0	-0.7	0.1	0.3	0.6	1.4	1.6	2.6	3.7
Citric Acid	-3.2	-2.2	-1.2	-0.8	0.0	0.3	0.6	1.4	1.8	2.7	3.9
Residual Sugar	-127.8	-90.6	-52.7	-39.7	-2.0	3.9	15.9	49.7	62.7	98.8	141.2
Chlorides	-1.2	-0.9	-0.5	-0.4	0.0	0.0	0.2	0.5	0.6	1.0	1.4
Free Sulfur Dioxide	-555.0	-388.0	-224.0	-171.0	0.0	30.0	70.0	230.0	284.0	469.0	623.0
Total Sulfur Dioxide	-823.0	-530.9	-273.0	-185.0	27.0	123.0	208.0	421.8	513.4	766.4	1057.0
Density	0.9	0.9	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.1	1.1
pH	0.5	1.3	2.1	2.3	3.0	3.2	3.5	4.1	4.4	5.1	6.1
Sulphates	-3.1	-2.1	-1.1	-0.7	0.3	0.5	0.9	1.8	2.1	3.2	4.2
Alcohol	-4.7	0.1	4.1	5.7	9.0	10.4	12.4	15.2	16.7	20.3	26.5
Label Appeal	-2.0	-2.0	-1.0	-1.0	-1.0	0.0	1.0	1.0	1.0	2.0	2.0
Acid Index	4.0	6.0	6.0	7.0	7.0	8.0	8.0	9.0	10.0	13.0	17.0

[a] 0% = minimum, 25% = Q1, 50% = median, 75% = Q3, 100% = maximum

The correlation matrix was constructed into a graphical display. The correlation coefficient is proportional to the color and size of the square. Positive correlation is blue and negative correlation is red. The large dark blue square visualized diagonally across the plot represents a correlation of 1, and it is the correlation of the variable against itself.

It was noted that Label Appeal and STARS had a strong positive correlation with each other and the TARGET variable. Acid Index was noted to have a negative correlation with STARS and TARGET. Models generated with these variable will be monitored for multicollinearity.

The variance inflation factor (VIF) estimates how much the variance of a predictor is inflated and a high value reflects multicollinearity. A predictor with high correlation to many predictors will have a high VIF value and conversely, a predictor with low correlation to many predictors will have a low VIF value. High VIF values contribute to model instability. Small values of VIF are preferred. The VIF values will be monitored for models when possible.

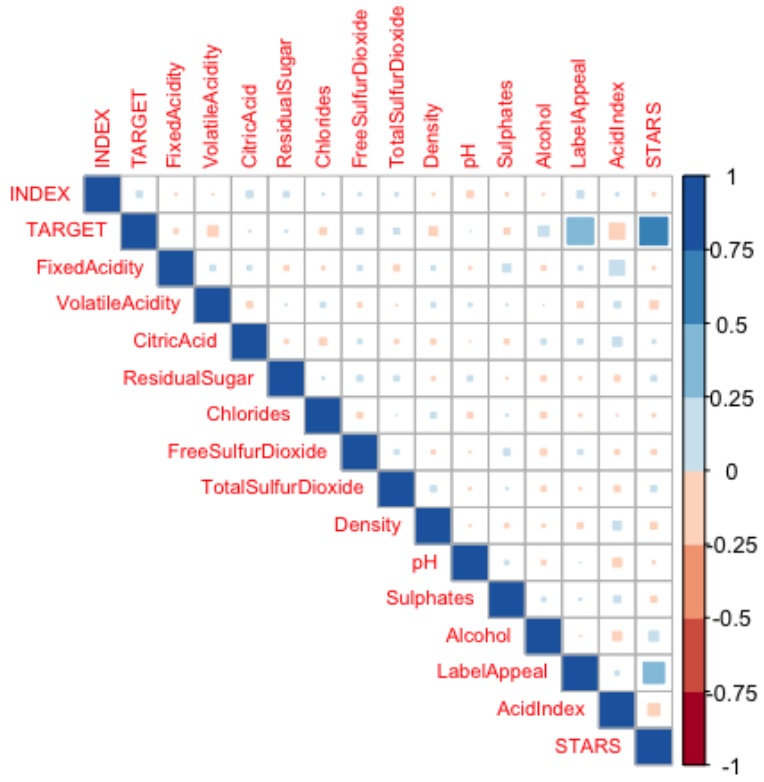


Figure 4:

Table 6: Wine TARGET Mean and Variance.

	TARGET
mean	3.029074
variance	3.710895

The mean and variance of the TARGET variable were examined. The variance is greater than the mean, which indicates overdispersion. If the overdispersion is not taken into account when creating predictive models using the binomial or Poisson regression, then the standard errors and confidence intervals will be too small. Additionally, a variable may be considered significant (z-value) when it is not. Overdispersion in the binomial or Poisson regression can be assessed with the ratio of the residual deviance to the residual degrees of freedom. If the ratio is much larger than 1, then overdispersion is a concern.

Section 2: Data Preparation

Before creating a predictive model, the data needed to be prepared. Missing values would need to be imputed, and variables with outliers would need a type of transformation in order to reduce the effect of the outlier on the model accuracy. After data preparation has been completed, then predictive models may be generated.

Section 2.1: Data Transformations

Variables with missing values were imputed with the median value for each variable. In order to do this, a new version of each variable was created but the new version of each variable name was distinguished

with a "_IMP" tag (e.g. ResidualSugar_IMP). This would allow the missing values to be imputed in the "_IMP" variable and the original variable would be unchanged. The mean and median were identical for pH, sulphates, and STARS.

Variables with missing values also had another new version created that was labeled with a "_IMP_Flag" tag (e.g. ResidualSugar_IMP_Flag). This variable would indicate missing values as a 1, and all other values as a 0. A new indicator variable was created called FINAL_REDFLAG. This new variable was based upon the summation of new indicator variables ("_IMP_REDFLAG") for volatile acidity, residual sugars, total sulfur dioxide, density. According to Cortez (2009), red wine generally had higher volatile acidity and pH, but lower residual sugars and total sulfur dioxide compared to white wines. Wines with a volatile acidity or pH value greater than the mean value for the respective variable were assigned a 1 and all other values as a 0. Wines with a residual sugar or total sulfur dioxide value less than the mean value for the respective variable were assigned a 1 and all other values as a 0. If the total tally of these indicator variables was greater than the mean of the total tally, then the observation was considered a red wine.

All data transformations applied to the `wine` set were also applied to the `wine_test` set.

Section 2.2: Outlier Transformation

Based upon the EDA, no outlier transformations were conducted on the `wine` and `wine_test` data set.

Section 3: Models

Utilizing the data set, seven models were created with predictor variables selected from EDA. The models generated were with a combination of manual or stepwise selection. Models were assessed with various metrics in Section 4.

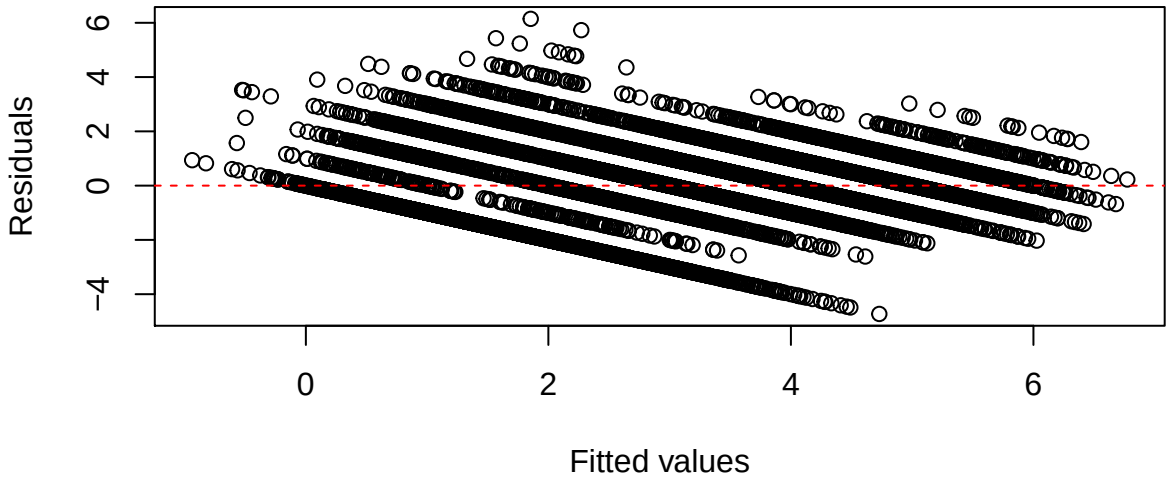
Section 3.1: Model 1 - Linear Regression

A linear regression model was created using selected variables from the `wine` set. The following model was generated:

```
Call:
lm(formula = TARGET ~ VolatileAcidity + Alcohol_IMP + LabelAppeal +
    STARS_IMP + STARS_IMP_Flag + AcidIndex + FreeSulfurDioxide_IMP +
    TotalSulfurDioxide_IMP + Chlorides_IMP + Sulphates_IMP, data = wine)

Residuals:
    Min      1Q  Median      3Q     Max
-4.7289 -0.8565  0.0295  0.8462  6.1457

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             3.477e+00  8.656e-02  40.171  < 2e-16 ***
VolatileAcidity        -9.722e-02  1.481e-02  -6.563 5.50e-11 ***
Alcohol_IMP             1.248e-02  3.199e-03   3.900 9.67e-05 ***
LabelAppeal             4.663e-01  1.367e-02  34.113  < 2e-16 ***
STARS_IMP               7.799e-01  1.567e-02  49.755  < 2e-16 ***
STARS_IMP_Flag         -2.246e+00  2.695e-02 -83.352  < 2e-16 ***
AcidIndex              -1.995e-01  8.943e-03 -22.309  < 2e-16 ***
FreeSulfurDioxide_IMP   2.837e-04  8.007e-05   3.543 0.000397 ***
TotalSulfurDioxide_IMP  2.237e-04  5.143e-05   4.350 1.37e-05 ***
```

```
Chlorides_IMP          -1.177e-01  3.735e-02  -3.152 0.001625 **
Sulphates_IMP          -3.094e-02  1.307e-02  -2.367 0.017939 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.31 on 12784 degrees of freedom
Multiple R-squared:  0.5379,    Adjusted R-squared:  0.5375
F-statistic:  1488 on 10 and 12784 DF,  p-value: < 2.2e-16
```

Term	VIF
VolatileAcidity	1.01
Alcohol_IMP	1.01
LabelAppeal	1.11
STARS_IMP	1.1
STARS_IMP_Flag	1.05
AcidIndex	1.04
FreeSulfurDioxide_IMP	1
TotalSulfurDioxide_IMP	1
Chlorides_IMP	1
Sulphates_IMP	1



The VIF for each predictor variable was calculated. The VIF is a measurement of collinearity between predictors. Small values of VIF are preferred. All VIF values were low.

All predictors were significant at the 95th significance level or higher. A scatterplot of the residuals versus predicted values indicated that a linear regression model is not ideal for the data.

Section 3.2: Model 2 - Stepwise Variable Selection and Linear Regression

A second linear regression model was created using variables from the `wine` data set. The `stepAIC` function in R was used to add and remove predictor variables until a model with the lowest AIC was achieved. The following model was generated with the stepwise method:

```
Call:
lm(formula = TARGET ~ ResidualSugar_IMP + Chlorides_IMP + FreeSulfurDioxide_IMP +
    Density + Sulphates_IMP + Alcohol_IMP + LabelAppeal + AcidIndex +
```

```
    STARS_IMP + VolatileAcidity_IMP_REDFLAG + ResidualSugar_IMP_REDFLAG +
    TotalSulfurDioxide_IMP_REDFLAG + Density_IMP_REDFLAG + STARS_IMP_Flag,
    data = wine)

Residuals:
    Min      1Q  Median      3Q     Max
-4.6653 -0.8386  0.0221  0.8348  6.0046

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     1.182e+00  5.747e-01   2.058 0.039646 *
ResidualSugar_IMP              -9.170e-04  4.651e-04  -1.972 0.048677 *
Chlorides_IMP                  -1.172e-01  3.714e-02  -3.156 0.001602 **
FreeSulfurDioxide_IMP           2.627e-04  7.963e-05   3.299 0.000974 ***
Density                         2.545e+00  5.812e-01   4.379 1.20e-05 ***
Sulphates_IMP                  -2.860e-02  1.299e-02  -2.201 0.027755 *
Alcohol_IMP                     1.159e-02  3.196e-03   3.627 0.000287 ***
LabelAppeal                     4.693e-01  1.359e-02  34.534  < 2e-16 ***
AcidIndex                      -1.787e-01  9.033e-03 -19.783  < 2e-16 ***
STARS_IMP                       7.673e-01  1.562e-02  49.133  < 2e-16 ***
VolatileAcidity_IMP_REDFLAG    -2.202e-01  2.352e-02  -9.361  < 2e-16 ***
ResidualSugar_IMP_REDFLAG      -1.214e-01  3.124e-02  -3.884 0.000103 ***
TotalSulfurDioxide_IMP_REDFLAG -1.372e-01  2.337e-02  -5.870 4.46e-09 ***
Density_IMP_REDFLAG            -2.794e-01  3.150e-02  -8.870  < 2e-16 ***
STARS_IMP_Flag                 -2.219e+00  2.687e-02 -82.578  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.302 on 12780 degrees of freedom
Multiple R-squared:  0.5437,    Adjusted R-squared:  0.5432
F-statistic:  1088 on 14 and 12780 DF,  p-value: < 2.2e-16
```

Term	VIF
ResidualSugar_IMP	1.77
Chlorides_IMP	1
FreeSulfurDioxide_IMP	1
Density	1.8
Sulphates_IMP	1
Alcohol_IMP	1.02
LabelAppeal	1.11
AcidIndex	1.08
STARS_IMP	1.11
VolatileAcidity_IMP_REDFLAG	1.03
ResidualSugar_IMP_REDFLAG	1.82
TotalSulfurDioxide_IMP_REDFLAG	1.02
Density_IMP_REDFLAG	1.87
STARS_IMP_Flag	1.06

All VIF values were low (<2). All predictors were significant at the 95th significance level or higher. Model 1 demonstrated that a linear regression was not ideal for the data, however the stepwise regression would be useful for predictor selection in subsequent models.

Section 3.3: Model 3 - Poisson Regression

A third model was created using the Poisson regression with the selected variables from the stepwise model in Section 3.2 above. The Poisson distribution assumes that the mean and variance are equal, and that the events are independent. The following model was generated:

```
Call:
glm(formula = TARGET ~ ResidualSugar_IMP + Chlorides_IMP + FreeSulfurDioxide_IMP +
    Density + Sulphates_IMP + Alcohol_IMP + LabelAppeal + AcidIndex +
    STARS_IMP + VolatileAcidity_IMP_REDFLAG + ResidualSugar_IMP_REDFLAG +
    TotalSulfurDioxide_IMP_REDFLAG + Density_IMP_REDFLAG + STARS_IMP_Flag,
    family = poisson(link = "log"), data = wine)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.1635  -0.6510   0.0063   0.4566   3.6662

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                     6.867e-01  2.549e-01   2.694 0.007053 **
ResidualSugar_IMP              -3.773e-04  2.051e-04  -1.839 0.065864 .
Chlorides_IMP                  -3.647e-02  1.646e-02  -2.215 0.026738 *
FreeSulfurDioxide_IMP           9.321e-05  3.510e-05   2.655 0.007924 **
Density                         8.594e-01  2.575e-01   3.338 0.000845 ***
Sulphates_IMP                  -1.119e-02  5.755e-03  -1.944 0.051882 .
Alcohol_IMP                     3.254e-03  1.416e-03   2.299 0.021507 *
LabelAppeal                     1.596e-01  6.130e-03  26.030  < 2e-16 ***
AcidIndex                      -7.373e-02  4.558e-03 -16.177  < 2e-16 ***
STARS_IMP                       1.839e-01  6.112e-03  30.091  < 2e-16 ***
VolatileAcidity_IMP_REDFLAG    -7.594e-02  1.050e-02  -7.233 4.74e-13 ***
ResidualSugar_IMP_REDFLAG      -4.671e-02  1.369e-02  -3.411 0.000647 ***
TotalSulfurDioxide_IMP_REDFLAG -4.830e-02  1.033e-02  -4.676 2.92e-06 ***
Density_IMP_REDFLAG            -9.442e-02  1.394e-02  -6.772 1.27e-11 ***
STARS_IMP_Flag                 -1.015e+00  1.700e-02 -59.666  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 22861  on 12794  degrees of freedom
Residual deviance: 13674  on 12780  degrees of freedom
AIC: 45646

Number of Fisher Scoring iterations: 6
```

Term	VIF
ResidualSugar_IMP	1.76
Chlorides_IMP	1
FreeSulfurDioxide_IMP	1
Density	1.8
Sulphates_IMP	1
Alcohol_IMP	1.02
LabelAppeal	1.13

Term	VIF
AcidIndex	1.05
STARS_IMP	1.15
VolatileAcidity_IMP_REDFLAG	1.02
ResidualSugar_IMP_REDFLAG	1.8
TotalSulfurDioxide_IMP_REDFLAG	1.02
Density_IMP_REDFLAG	1.88
STARS_IMP_Flag	1.03

```
Ratio of residual deviance to residual degrees of freedom: 1.069972

Calculated 95th percentile of the Chi-Squared distriubtion: 13044.1

Degrees of freedom: 12780
```

All VIF values were low (<2). All predictors were significant at the 95th significance level or higher, except for ResidualSugar_IMP and Sulphates_IMP that were significant at the 90th significance level. The ratio of the residual deviance to residual degrees of freedom was close to 1, which indicates that overdispersion was not an issue.

The calculated 95th percentile of the Chi-Squared distribution with 12780 degrees of freedom is 13044.1. The model residual deviance (13674) is greater than the 95th percentile, which means that the model does not fit the data well.

Section 3.4: Model 4 - Negative Binomial Regression

A fourth model was created using the negative binomial regression model with the selected variables from the stepwise model in Section 3.2 above. The output was the same as the Poisson regression. In an effort to create a different model, the ResidualSugar_IMP and Sulphates_IMP variables were removed. The following model was generated:

```
Call:
glm.nb(formula = TARGET ~ Chlorides_IMP + FreeSulfurDioxide_IMP +
    Density + Alcohol_IMP + LabelAppeal + AcidIndex + STARS_IMP +
    VolatileAcidity_IMP_REDFLAG + ResidualSugar_IMP_REDFLAG +
    TotalSulfurDioxide_IMP_REDFLAG + Density_IMP_REDFLAG + STARS_IMP_Flag,
    data = wine, init.theta = 40994.97073, link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.1910  -0.6484   0.0085   0.4524   3.7048

Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                  6.824e-01  2.548e-01   2.678 0.007405 **
Chlorides_IMP               -3.638e-02  1.646e-02  -2.210 0.027113 *
FreeSulfurDioxide_IMP        9.239e-05  3.510e-05   2.633 0.008475 **
Density                      8.488e-01  2.573e-01   3.298 0.000973 ***
Alcohol_IMP                  3.192e-03  1.416e-03   2.255 0.024125 *
LabelAppeal                  1.595e-01  6.129e-03  26.030  < 2e-16 ***
AcidIndex                   -7.393e-02  4.556e-03 -16.226  < 2e-16 ***
STARS_IMP                    1.840e-01  6.112e-03  30.100  < 2e-16 ***
VolatileAcidity_IMP_REDFLAG -7.633e-02  1.050e-02  -7.270 3.59e-13 ***
ResidualSugar_IMP_REDFLAG   -3.009e-02  1.032e-02  -2.914 0.003568 **
```

```
TotalSulfurDioxide_IMP_REDFLAG -4.913e-02  1.032e-02  -4.760 1.94e-06 ***
Density_IMP_REDFLAG            -9.308e-02  1.392e-02  -6.687 2.27e-11 ***
STARS_IMP_Flag                 -1.015e+00  1.700e-02 -59.728  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(40994.97) family taken to be 1)

    Null deviance: 22860  on 12794  degrees of freedom
Residual deviance: 13681  on 12782  degrees of freedom
AIC: 45652

Number of Fisher Scoring iterations: 1

              Theta:  40995
          Std. Err.:  34959
Warning while fitting theta: iteration limit reached

 2 x log-likelihood:  -45623.77
```

Term	VIF
Chlorides_IMP	1
FreeSulfurDioxide_IMP	1
Density	1.8
Alcohol_IMP	1.02
LabelAppeal	1.13
AcidIndex	1.05
STARS_IMP	1.15
VolatileAcidity_IMP_REDFLAG	1.02
ResidualSugar_IMP_REDFLAG	1.03
TotalSulfurDioxide_IMP_REDFLAG	1.02
Density_IMP_REDFLAG	1.87
STARS_IMP_Flag	1.03

```
Ratio of residual deviance to residual degrees of freedom: 1.070318
```

The ratio of the residual deviance to residual degrees of freedom was close to 1, which indicates that overdispersion is not an issue. VIF values are low (<2). All predictors were significant at the 95th significance level or higher.

The message `Warning while fitting theta: iteration limit reached` was displayed with the model summary. This model does not appear to be a good fit for the data.

Section 3.5: Model 5 - Zero Inflated Poisson Regression

A fifth model was created using zero inflated Poisson regression with the selected variables from the stepwise model in Section 3.2 above. The following model was generated:

```
Call:
zeroinfl(formula = TARGET ~ ResidualSugar_IMP + Chlorides_IMP +
    FreeSulfurDioxide_IMP + Density + Sulphates_IMP + Alcohol_IMP +
    LabelAppeal + AcidIndex + STARS_IMP + VolatileAcidity_IMP_REDFLAG +
    ResidualSugar_IMP_REDFLAG + TotalSulfurDioxide_IMP_REDFLAG +
    Density_IMP_REDFLAG + STARS_IMP_Flag, data = wine)

Pearson residuals:
      Min        1Q    Median        3Q       Max
-2.328321 -0.412451 -0.005701  0.371889  6.832997

Count model coefficients (poisson with log link):
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                  7.769e-01  2.626e-01   2.959 0.003090 **
ResidualSugar_IMP            6.293e-06  2.101e-04   0.030 0.976099
Chlorides_IMP               -2.431e-02  1.688e-02  -1.441 0.149659
FreeSulfurDioxide_IMP        2.593e-05  3.540e-05   0.732 0.463925
Density                      4.154e-01  2.657e-01   1.564 0.117919
Sulphates_IMP               -1.638e-04  5.909e-03  -0.028 0.977879
Alcohol_IMP                  6.399e-03  1.446e-03   4.426 9.61e-06 ***
LabelAppeal                  2.326e-01  6.319e-03  36.812  < 2e-16 ***
AcidIndex                   -1.616e-02  4.864e-03  -3.322 0.000894 ***
STARS_IMP                    1.023e-01  6.418e-03  15.942  < 2e-16 ***
VolatileAcidity_IMP_REDFLAG -3.092e-02  1.072e-02  -2.885 0.003911 **
ResidualSugar_IMP_REDFLAG    6.080e-03  1.400e-02   0.434 0.664078
TotalSulfurDioxide_IMP_REDFLAG  9.361e-03  1.055e-02   0.887 0.374942
Density_IMP_REDFLAG         -5.606e-02  1.428e-02  -3.927 8.61e-05 ***
STARS_IMP_Flag              -1.824e-01  1.857e-02  -9.821  < 2e-16 ***

Zero-inflation model coefficients (binomial with logit link):
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                  1.4112031  1.7418683   0.810  0.41784
ResidualSugar_IMP            0.0035193  0.0013832   2.544  0.01095 *
Chlorides_IMP                0.0951516  0.1094904   0.869  0.38482
FreeSulfurDioxide_IMP       -0.0006296  0.0002417  -2.604  0.00920 **
Density                     -3.5657834  1.7381711  -2.051  0.04022 *
Sulphates_IMP                0.1246913  0.0386574   3.226  0.00126 **
Alcohol_IMP                  0.0273919  0.0095460   2.869  0.00411 **
LabelAppeal                  0.7211668  0.0427039  16.888  < 2e-16 ***
AcidIndex                    0.3764362  0.0257241  14.634  < 2e-16 ***
STARS_IMP                   -3.7899797  0.3314532 -11.434  < 2e-16 ***
VolatileAcidity_IMP_REDFLAG  0.4301474  0.0689453   6.239 4.40e-10 ***
ResidualSugar_IMP_REDFLAG    0.4695802  0.0946526   4.961 7.01e-07 ***
TotalSulfurDioxide_IMP_REDFLAG  0.5325183  0.0693356   7.680 1.59e-14 ***
Density_IMP_REDFLAG          0.3701774  0.0933890   3.964 7.38e-05 ***
STARS_IMP_Flag               5.8482517  0.3318195  17.625  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Number of iterations in BFGS optimization: 41
Log-likelihood: -2.033e+04 on 30 Df
```

The zero inflated Poisson regression contains two models - one for the whether the wine is purchased (zero-inflation model), and one that predicts how many cases are purchased if you exclude the wines that are not purchased (count model). Given the two model output, the VIF was not calculated.

Table 11: Model Performance.

	Probability of Zeros
Observed	0.2136772
Poisson	0.1112855
Negative Binomal	0.1112349
Zero Inflated Poisson	0.2203398

```
Calculated 95th percentile of the Chi-Squared distriubtion: 13028.95

Degrees of freedom: 12765
```

The number of zeroes observed in the `wine` data set was 21.3%. The number of zeroes predicted by both the Poisson model and the negative binomial was 11.1%. The Poisson and negative binomial models underestimate the probability of zero counts. The zero inflated Poisson model assumes that the `wine` data is composed of two populations - one where the counts are always zero and the other where the counts have a Poisson distribution. The number of zeroes predicted by the zero inflated Poisson model was 22.0%, which was much closer to the observed value in the `wine` data set.

Section 3.6: Model 6 - Zero Inflated Negative Binomial Regression

A sixth model was created using zero inflated negative binomial regression with the selected variables from the stepwise model in Section 3.2 above. The following model was generated:

```
Call:
zeroinfl(formula = TARGET ~ ResidualSugar_IMP + Chlorides_IMP +
    FreeSulfurDioxide_IMP + Density + Sulphates_IMP + Alcohol_IMP +
    LabelAppeal + AcidIndex + STARS_IMP + VolatileAcidity_IMP_REDFLAG +
    ResidualSugar_IMP_REDFLAG + TotalSulfurDioxide_IMP_REDFLAG +
    Density_IMP_REDFLAG + STARS_IMP_Flag, data = wine, dist = "negbin",
    EM = TRUE)

Pearson residuals:
      Min        1Q    Median        3Q       Max
-2.328325 -0.412450 -0.005697  0.371886  6.833056

Count model coefficients (negbin with log link):
                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                 7.769e-01  2.626e-01   2.959 0.003091 **
ResidualSugar_IMP           6.291e-06  2.101e-04   0.030 0.976110
Chlorides_IMP              -2.431e-02  1.688e-02  -1.441 0.149649
FreeSulfurDioxide_IMP       2.593e-05  3.540e-05   0.732 0.463897
Density                     4.154e-01  2.657e-01   1.564 0.117907
Sulphates_IMP              -1.641e-04  5.909e-03  -0.028 0.977848
Alcohol_IMP                 6.399e-03  1.446e-03   4.426 9.61e-06 ***
LabelAppeal                 2.326e-01  6.319e-03  36.812  < 2e-16 ***
```

```
AcidIndex                       -1.616e-02  4.864e-03  -3.322 0.000894 ***
STARS_IMP                        1.023e-01  6.418e-03  15.942  < 2e-16 ***
VolatileAcidity_IMP_REDFLAG     -3.092e-02  1.072e-02  -2.885 0.003911 **
ResidualSugar_IMP_REDFLAG        6.079e-03  1.400e-02   0.434 0.664127
TotalSulfurDioxide_IMP_REDFLAG   9.360e-03  1.055e-02   0.887 0.374968
Density_IMP_REDFLAG             -5.606e-02  1.428e-02  -3.927 8.61e-05 ***
STARS_IMP_Flag                  -1.824e-01  1.857e-02  -9.821  < 2e-16 ***
Log(theta)                       1.233e+01  3.902e+00   3.160 0.001577 **

Zero-inflation model coefficients (binomial with logit link):
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                      1.4113797  1.7418969   0.810  0.41779
ResidualSugar_IMP                0.0035193  0.0013832   2.544  0.01095 *
Chlorides_IMP                    0.0951457  0.1094912   0.869  0.38486
FreeSulfurDioxide_IMP           -0.0006296  0.0002417  -2.604  0.00920 **
Density                         -3.5656276  1.7381843  -2.051  0.04023 *
Sulphates_IMP                    0.1246911  0.0386577   3.226  0.00126 **
Alcohol_IMP                      0.0273926  0.0095461   2.870  0.00411 **
LabelAppeal                      0.7211760  0.0427043  16.888  < 2e-16 ***
AcidIndex                        0.3764368  0.0257243  14.633  < 2e-16 ***
STARS_IMP                       -3.7903329  0.3315575 -11.432  < 2e-16 ***
VolatileAcidity_IMP_REDFLAG      0.4301482  0.0689458   6.239 4.41e-10 ***
ResidualSugar_IMP_REDFLAG        0.4695771  0.0946533   4.961 7.01e-07 ***
TotalSulfurDioxide_IMP_REDFLAG   0.5325208  0.0693361   7.680 1.59e-14 ***
Density_IMP_REDFLAG              0.3701715  0.0933897   3.964 7.38e-05 ***
STARS_IMP_Flag                   5.8486161  0.3319239  17.620  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Theta = 226389.7308
Number of iterations in BFGS optimization: 1
Log-likelihood: -2.033e+04 on 31 Df
```

Table 12: Model Performance.

	Probability of Zeros
Observed	0.2136772
Poisson	0.1112855
Negative Binomal	0.1112349
Zero Inflated Poisson	0.2203398
Zero Inflated Negative Binomal	0.2203388

The zero inflated negative binomial regression contains two models - one for the whether the wine is purchased (zero-inflation model), and one that predicts how many cases are purchased if you exclude the wines that are not purchased (count model). Given the two model output, the VIF was not calculated. The model generated with zero inflated negative binomial regression was very similiar to the model generated with zero inflated Poisson regression.

The estimated theta parameter is significant ($\Pr(>|z|) = 0.001577$), which indicates that the zero-inflated negative binomial regression model is not appropriate.

The number of zeroes predicted by the zero inflated negative binomial model was 22.0%, which was very close to the observed value in the `wine` data set.

Section 3.7: Model 7 - Zero Inflated Poisson Regression II

A seventh model was created using zero inflated Poisson regression with variables that were at the 95th significance level or higher from the zero inflated Poisson regression model in Section 3.5 above. The following model was generated:

```
Call:
zeroinfl(formula = TARGET ~ Alcohol_IMP + LabelAppeal + AcidIndex +
    STARS_IMP + VolatileAcidity_IMP_REDFLAG + Density_IMP_REDFLAG +
    STARS_IMP_Flag | -1 + ResidualSugar_IMP + FreeSulfurDioxide_IMP +
    Density + Sulphates_IMP + Alcohol_IMP + LabelAppeal + AcidIndex +
    STARS_IMP + VolatileAcidity_IMP_REDFLAG + ResidualSugar_IMP_REDFLAG +
    TotalSulfurDioxide_IMP_REDFLAG + Density_IMP_REDFLAG + STARS_IMP_Flag,
    data = wine)

Pearson residuals:
      Min        1Q    Median        3Q       Max
-2.322738 -0.412061 -0.001281  0.376130  6.813668

Count model coefficients (poisson with log link):
                             Estimate Std. Error z value Pr(>|z|)
(Intercept)                  1.188675   0.043137  27.556  < 2e-16 ***
Alcohol_IMP                  0.006656   0.001441   4.619 3.86e-06 ***
LabelAppeal                  0.232401   0.006316  36.796  < 2e-16 ***
AcidIndex                   -0.016403   0.004861  -3.375 0.000739 ***
STARS_IMP                    0.102692   0.006411  16.018  < 2e-16 ***
VolatileAcidity_IMP_REDFLAG -0.030843   0.010712  -2.879 0.003985 **
Density_IMP_REDFLAG         -0.042755   0.010536  -4.058 4.95e-05 ***
STARS_IMP_Flag              -0.182565   0.018566  -9.833  < 2e-16 ***

Zero-inflation model coefficients (binomial with logit link):
                                Estimate Std. Error z value Pr(>|z|)
ResidualSugar_IMP               0.0035243  0.0013782   2.557  0.01055 *
FreeSulfurDioxide_IMP          -0.0006548  0.0002407  -2.721  0.00652 **
Density                        -2.2112119  0.4078205  -5.422 5.89e-08 ***
Sulphates_IMP                   0.1252413  0.0384829   3.254  0.00114 **
Alcohol_IMP                     0.0275389  0.0095482   2.884  0.00392 **
LabelAppeal                     0.7229808  0.0426990  16.932  < 2e-16 ***
AcidIndex                       0.3812378  0.0253030  15.067  < 2e-16 ***
STARS_IMP                      -3.7356077  0.3122708 -11.963  < 2e-16 ***
VolatileAcidity_IMP_REDFLAG     0.4309145  0.0688521   6.259 3.89e-10 ***
ResidualSugar_IMP_REDFLAG       0.4664949  0.0942546   4.949 7.45e-07 ***
TotalSulfurDioxide_IMP_REDFLAG  0.5282336  0.0690830   7.646 2.07e-14 ***
Density_IMP_REDFLAG             0.3233992  0.0726342   4.452 8.49e-06 ***
STARS_IMP_Flag                  5.8000556  0.3132911  18.513  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Number of iterations in BFGS optimization: 30
Log-likelihood: -2.033e+04 on 21 Df
```

The zero inflated Poisson regression contains two models - one for the whether the wine is purchased (zero-inflation model), and one that predicts how many cases are purchased if you exclude the wines that are not purchased (count model). The zero-inflation model was specified without an intercept. Given the two model

output, the VIF was not calculated.

Table 13: Model Performance.

	Probability of Zeros
Observed	0.2136772
Poisson	0.1112855
Negative Binomal	0.1112349
Zero Inflated Poisson	0.2203398
Zero Inflated Negative Binomal	0.2203388
Zero Inflated Poisson II	0.2202885

The number of zeroes predicted by the zero inflated Poisson model II was 22.0%, which was very close to the observed value in the `wine` data set.

Section 4: Model Selection

For each of the models the deviance, log likelihood, Akaike's Information Criterion (AIC), and Bayesian Information Criterion (BIC) were calculated. These metrics are used to evaluate the model fit. A small value of AIC and BIC is desired. The deviance is equal to -2 times the log likelihood of the model. Smaller values of deviance are preferred.

The MSE measures the average of the squares of the residuals, and the MAE measures the average of the absolute value of the residuals. A smaller value of the MSE and MAE is preferred.

The number of zeroes predicted by the non-linear regression models was compared against the observed value in the `wine` data set. A predicted value closest to the observed value is preferred.

The BIC value was not calculated for the zero inflated models.

Table 14: Model Performance.

	AIC	BIC	MSE	MAE	Deviance	logLik
Linear	43235	43325	2	1	43307	-21606
Stepwise	43081	43201	2	1	43157	-21525
Poisson	45646	45758	1	1	45616	-22808
Negative Binomal	45652	45756	1	1	45624	-22812
Zero Inflated Poisson	40717	NA	2	1	40657	-20329
Zero Inflated NB	40719	NA	2	1	40657	-20329
Zero Inflated Poisson II	40707	NA	2	1	40665	-20333

Table 15: Model Performance.

	Probability of Zeros
Observed	0.2136772
Poisson	0.1112855
Negative Binomal	0.1112349
Zero Inflated Poisson	0.2203398
Zero Inflated Negative Binomal	0.2203388
Zero Inflated Poisson II	0.2202885

Conclusion

The zero inflated Poisson regression model presented in Section 3.5 was selected to predict the TARGET value in the `wine_test` data set. This model had the lowest deviance and logLik, and the second lowest AIC. The model predicted the number of zeroes at 22.0%, which was very close to the observed value (21.3%) in the `wine` data set.

References

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems, 47, 547 - 553.

Fox, J. and Weisberg, S. (2011). An R Companion to Applied Regression, Second Edition. Thousand Oaks, CA: Sage Publications, Inc.

Hoffmann, J.P. (2004). Generalized Linear Models: An Applied Approach. Boston, MA: Pearson Education, Inc.

Kabacoff, R. (2015). R in Action, Second Edition. Shelter Island, NY: Manning Publications Co.

Lander, J. (2014). R for Everyone. Upper Saddle River, NJ: Addison-Wesley.

Rodriguez, G. (2017). Generalized Linear Models: 4.A Models for Over-Dispersed Count Data. Retrieved from http://data.princeton.edu/wws509/r/overdispersion.html

Stowell, S. (2014). Using R for Statistics. New York, NY: Apress.

Wickham, H. (2016). ggplot2: elegant graphics for data analysis (Use R!). Switzerland: Springer International Publishing AG.

##Code

```
knitr::opts_chunk$set(echo = FALSE, fig.pos = 'h')
#knitr::opts_chunk$set(dev = 'pdf')
#options(knitr.table.format = 'markdown')
library(knitr)

#call libraries
library(ggplot2) # For graphical tools
library(MASS) # For some advanced statistics
library(pscl) # For "counting" models (e.g., Poisson and Negative Binomial)
library(dplyr) # For general needs and functions
library(readr)
library(corrplot)

# Note, some of these libraries are not needed for this template code.
library(zoo)
library(psych)
library(ROCR)
library(car)
library(InformationValue)
library(rJava)
library(pbkrtest)
library(car)
library(leaps)
library(glm2)
library(aod)
library(GGally)
library(magrittr)
library(kableExtra)
library(RColorBrewer)
library(broom)
library(cowplot)
library(stargazer)
library(gridExtra)


#Read File in from your working directory
setwd("~/Desktop/R/")
wine = read.csv("Wine_Training.csv")  # read csv file

#take a look at the high level characteristics of the wine data
summary(wine)
str(wine)

#Read data dictionary in from working directory
dictionary = read.csv("DataDictionary_Wine.csv")
str(dictionary)


#grid.table(dictionary, rows=NULL)
kable(dictionary,
      caption = "Data Dictionary of Variables.",
      format = "latex", booktabs = T ) %>%
```

```
  kable_styling(latex_options = c("striped", "scale_down"))


#examine the target variable
target.hist <- ggplot(data=wine, aes(wine$TARGET)) +
  geom_histogram(binwidth =1,
                 col="BLUE",
                 aes(fill=..count..))+
  scale_fill_gradient("Count", low = "blue", high = "red") +
  xlab("Target") +
  ylab("Count")
#examine label appeal with target and stars
label.stars.jitter <- ggplot(wine, aes(LabelAppeal, TARGET, colour = factor(STARS))) +
  geom_jitter() +
  xlab("Label Appeal") +
  ylab("Target")

plot_grid(target.hist, label.stars.jitter, labels = "AUTO")


ggplot(data=wine, aes(wine$VolatileAcidity)) +
  geom_histogram(binwidth =1,
                 col="BLUE",
                 aes(fill=..count..))+
  scale_fill_gradient("Count", low = "blue", high = "red")

ggplot(wine, aes(FixedAcidity, TARGET, colour = STARS)) +
  geom_jitter()

ggplot(wine, aes(FixedAcidity, TARGET, colour = STARS)) +
  geom_jitter()

ggplot(wine, aes(AcidIndex, FixedAcidity, colour = STARS)) +
  geom_jitter()

ggplot(wine, aes(Density, Sulphates, colour = STARS)) +
  geom_jitter()

ggplot(wine, aes(ResidualSugar, TARGET, colour = STARS)) +
  geom_jitter()

ggplot(wine, aes(STARS, TARGET, colour = STARS)) +
  geom_jitter()

ggplot(data=wine, aes(wine$LabelAppeal)) +
  geom_histogram(binwidth =1,
                 col="BLUE",
                 aes(fill=..count..))+
  scale_fill_gradient("Count", low = "blue", high = "red")

ggplot(wine, aes(STARS, LabelAppeal, colour = STARS)) +
  geom_jitter()
```

```r
ggplot(wine, aes(AcidIndex, TARGET, colour=STARS)) + geom_jitter()


#boxplot of useful variables
#boxplot(wine$TARGET~wine$STARS, col = "green", main = " ")
plot1 <- ggplot(wine, aes(x=factor(STARS), TARGET)) +
  geom_boxplot(fill='#56B4E9', color="black") + theme_classic()


#boxplot of useful variables
#boxplot(wine$LabelAppeal~wine$TARGET, col = "red", main = " ")
plot2 <- ggplot(wine, aes(x=factor(TARGET), LabelAppeal)) +
  geom_boxplot(fill='#E69F00', color="black") + theme_classic()
grid.arrange(plot1, plot2, ncol=2)

#function to describe variables with missing data
myNA.variable.summary <- function(variable){
  NAsum.summary <- sum(is.na(variable))
  NAmean.summary <- (round(mean(is.na(variable)), 4)*100)
  return(c(NAsum.summary, NAmean.summary))
}

#table of data variables with missing data
NA.RS <- myNA.variable.summary(wine$ResidualSugar)
NA.C <- myNA.variable.summary(wine$Chlorides)
NA.FSD <- myNA.variable.summary(wine$FreeSulfurDioxide)
NA.TSD <- myNA.variable.summary(wine$TotalSulfurDioxide)
NA.pH <- myNA.variable.summary(wine$pH)
NA.S <- myNA.variable.summary(wine$Sulphates)
NA.A <- myNA.variable.summary(wine$Alcohol)
NA.STARS <- myNA.variable.summary(wine$STARS)

overview.NA <- rbind(NA.RS, NA.C, NA.FSD, NA.TSD, NA.pH, NA.S, NA.A, NA.STARS)
colnames(overview.NA) <- c("Number of missing values", "Percent of missing values")
rownames(overview.NA) <- c("Residual Sugar", "Chlorides", "Free Sulfur Dioxide",
                           "Total Sulfur Dioxide", "pH", "Sulphates", "Alcohol",
                           "STARS")
kable(overview.NA, caption = 'Percentage of Missing Values for Affected Variables.')
#grid.table(overview.NA)


# Frequencies, proportions and cumulative proportions side-by-side.
My.Table <- function(X){
  Table <- data.frame(table(X))
  Table$Prop <- round(prop.table(Table$Freq), 4)
  Table$CumProp <-  cumsum(Table$Prop)
  Table
}


kable(My.Table(wine$LabelAppeal),
      caption = "Frequency, Proportion and Cumulative Proportion of Label Appeal.")
```

```
kable(My.Table(wine$STARS),
      caption = "Frequency, Proportion and Cumulative Proportion of STARS.")


kable(addmargins(table(wine$LabelAppeal, wine$STARS)),
      caption = "Frequency of Wines According to STARS and Label Appeal.")


quantile.FA <- quantile(wine$FixedAcidity,
                        probs = c(0,1,5,10,25,50,75,90,95,99,100)/100, na.rm=TRUE)
quantile.VA <- quantile(wine$VolatileAcidity,
                        probs = c(0,1,5,10,25,50,75,90,95,99,100)/100, na.rm=TRUE)
quantile.CA <- quantile(wine$CitricAcid,
                        probs = c(0,1,5,10,25,50,75,90,95,99,100)/100, na.rm=TRUE)
quantile.RS <- quantile(wine$ResidualSugar,
                        probs = c(0,1,5,10,25,50,75,90,95,99,100)/100, na.rm=TRUE)
quantile.Cl <- quantile(wine$Chlorides,
                        probs = c(0,1,5,10,25,50,75,90,95,99,100)/100, na.rm=TRUE)
quantile.FSD <- quantile(wine$FreeSulfurDioxide,
                        probs = c(0,1,5,10,25,50,75,90,95,99,100)/100, na.rm=TRUE)
quantile.TSD <- quantile(wine$TotalSulfurDioxide,
                        probs = c(0,1,5,10,25,50,75,90,95,99,100)/100, na.rm=TRUE)
quantile.Den <- quantile(wine$Density,
                        probs = c(0,1,5,10,25,50,75,90,95,99,100)/100, na.rm=TRUE)
quantile.pH <- quantile(wine$pH,
                        probs = c(0,1,5,10,25,50,75,90,95,99,100)/100, na.rm=TRUE)
quantile.Sul <- quantile(wine$Sulphates,
                        probs = c(0,1,5,10,25,50,75,90,95,99,100)/100, na.rm=TRUE)
quantile.Alc <- quantile(wine$Alcohol,
                        probs = c(0,1,5,10,25,50,75,90,95,99,100)/100, na.rm=TRUE)
quantile.LA <- quantile(wine$LabelAppeal,
                        probs = c(0,1,5,10,25,50,75,90,95,99,100)/100, na.rm=TRUE)
quantile.AI <- quantile(wine$AcidIndex,
                        probs = c(0,1,5,10,25,50,75,90,95,99,100)/100, na.rm=TRUE)

overview.quantile <- round(rbind(quantile.FA, quantile.VA, quantile.CA, quantile.RS,
                          quantile.Cl, quantile.FSD, quantile.TSD, quantile.Den,
                          quantile.pH, quantile.Sul, quantile.Alc, quantile.LA,
                          quantile.AI), 1)
colnames(overview.quantile) <- c("0%","1%","5%","10%","25%","50%",
                                 "75%", "90%", "95%", "99%", "100%")
rownames(overview.quantile) <- c("Fixed Acidity", "Volatile Acidity", "Citric Acid",
                                 "Residual Sugar", "Chlorides", "Free Sulfur Dioxide",
                                 "Total Sulfur Dioxide", "Density", "pH", "Sulphates",
                                 "Alcohol", "Label Appeal", "Acid Index")
#grid.table(overview.quantile)
kable(overview.quantile, caption = 'Quantiles of Variables of Interest.',
      format = "latex", booktabs = T) %>%
  kable_styling(latex_options = c("striped", "scale_down")) %>%
  add_footnote(c("0% = minimum, 25% = Q1, 50% = median, 75% = Q3, 100% = maximum"))


#Correlation numbers
```

```
wine_clean <- na.omit(wine)
round(cor(wine_clean[sapply(wine_clean[c(-1)], is.numeric)]), 4)

c <- cor(wine_clean[sapply(wine_clean[c(-1)], is.numeric)])

#Correlation plot
corrplot::corrplot(c, method = "square",
                   col=brewer.pal(n=8, name="RdBu"),
                   diag = TRUE, tl.cex = 0.7, number.cex = 0.7,
                   type= "upper")


#what type of dispersion does sample have?
TARGET.mean <- mean(wine$TARGET)
TARGET.var <- var(wine$TARGET)

#The variance is greater than the mean
#this means overdispersion
#> mean(wine$TARGET)
#[1] 3.029074
#> var(wine$TARGET)
#[1] 3.710895


fit.compare <- rbind(TARGET.mean, TARGET.var)
colnames(fit.compare) <- c('TARGET')
rownames(fit.compare) <- c('mean', 'variance')
kable(fit.compare, caption = 'Wine TARGET Mean and Variance.')


######################################################################
###################### Data Preparation ##############################################
#create IMP versions of each independent variable
#The IMP columns are copies of the original columns
wine$FixedAcidity_IMP <- wine$FixedAcidity
wine$VolatileAcidity_IMP <- wine$VolatileAcidity
wine$CitricAcid_IMP <- wine$CitricAcid
wine$ResidualSugar_IMP <- wine$ResidualSugar
wine$Chlorides_IMP <- wine$Chlorides
wine$FreeSulfurDioxide_IMP <- wine$FreeSulfurDioxide
wine$TotalSulfurDioxide_IMP <- wine$TotalSulfurDioxide
wine$Density_IMP <- wine$Density
wine$pH_IMP <- wine$pH
wine$Sulphates_IMP <- wine$Sulphates
wine$Alcohol_IMP <- wine$Alcohol
wine$LabelAppeal_IMP <- wine$LabelAppeal
wine$AcidIndex_IMP <- wine$AcidIndex
wine$STARS_IMP <- wine$STARS

#replace NA's in each IMP column with median
wine$FixedAcidity_IMP[which(is.na(wine$FixedAcidity_IMP))] <-
  median(wine$FixedAcidity_IMP,na.rm = TRUE)
wine$VolatileAcidity_IMP[which(is.na(wine$VolatileAcidity_IMP))] <-
```

```
  median(wine$VolatileAcidity_IMP,na.rm = TRUE)
wine$CitricAcid_IMP[which(is.na(wine$CitricAcid_IMP))] <-
  median(wine$CitricAcid_IMP,na.rm = TRUE)
wine$ResidualSugar_IMP[which(is.na(wine$ResidualSugar_IMP))] <-
  median(wine$ResidualSugar_IMP,na.rm = TRUE)
wine$Chlorides_IMP[which(is.na(wine$Chlorides_IMP))] <-
  median(wine$Chlorides_IMP,na.rm = TRUE)
wine$FreeSulfurDioxide_IMP[which(is.na(wine$FreeSulfurDioxide_IMP))] <-
  median(wine$FreeSulfurDioxide_IMP,na.rm = TRUE)
wine$TotalSulfurDioxide_IMP[which(is.na(wine$TotalSulfurDioxide_IMP))] <-
  median(wine$TotalSulfurDioxide_IMP,na.rm = TRUE)
wine$Density_IMP[which(is.na(wine$Density_IMP))] <-
  median(wine$Density_IMP,na.rm = TRUE)
wine$pH_IMP[which(is.na(wine$pH_IMP))] <-
  median(wine$pH_IMP,na.rm = TRUE)
wine$Sulphates_IMP[which(is.na(wine$Sulphates_IMP))] <-
  median(wine$Sulphates_IMP,na.rm = TRUE)
wine$Alcohol_IMP[which(is.na(wine$Alcohol_IMP))] <-
  median(wine$Alcohol_IMP,na.rm = TRUE)
wine$LabelAppeal_IMP[which(is.na(wine$LabelAppeal_IMP))] <-
  median(wine$LabelAppeal_IMP,na.rm = TRUE)
wine$AcidIndex_IMP[which(is.na(wine$AcidIndex_IMP))] <-
  median(wine$AcidIndex_IMP,na.rm = TRUE)
wine$STARS_IMP[which(is.na(wine$STARS_IMP))] <-
  median(wine$STARS_IMP,na.rm = TRUE)

#flag NA values with new field
#first, create new field
#second, replace NA's with 1 else 0

wine$ResidualSugar_IMP_Flag <- wine$ResidualSugar
wine$Chlorides_IMP_Flag <- wine$Chlorides
wine$FreeSulfurDioxide_IMP_Flag <- wine$FreeSulfurDioxide
wine$TotalSulfurDioxide_IMP_Flag <- wine$TotalSulfurDioxide
wine$pH_IMP_Flag <- wine$pH
wine$Sulphates_IMP_Flag <- wine$Sulphates
wine$Alcohol_IMP_Flag <- wine$Alcohol
wine$STARS_IMP_Flag <- wine$STARS


#NEW BINARY FIELDS TO SHOW na's
wine$ResidualSugar_IMP_Flag <-
  ifelse(is.na(wine$ResidualSugar_IMP_Flag)==TRUE, 1,0)
wine$Chlorides_IMP_Flag <-
  ifelse(is.na(wine$Chlorides_IMP_Flag)==TRUE, 1,0)
wine$FreeSulfurDioxide_IMP_Flag <-
  ifelse(is.na(wine$FreeSulfurDioxide_IMP_Flag)==TRUE, 1,0)
wine$TotalSulfurDioxide_IMP_Flag <-
  ifelse(is.na(wine$TotalSulfurDioxide_IMP_Flag)==TRUE, 1,0)
wine$pH_IMP_Flag <-
  ifelse(is.na(wine$pH_IMP_Flag)==TRUE, 1,0)
wine$Sulphates_IMP_Flag <-
  ifelse(is.na(wine$Sulphates_IMP_Flag)==TRUE, 1,0)
```

```
wine$Alcohol_IMP_Flag <-
  ifelse(is.na(wine$Alcohol_IMP_Flag)==TRUE, 1,0)
wine$STARS_IMP_Flag <-
  ifelse(is.na(wine$STARS_IMP_Flag)==TRUE, 1,0) #LOOK FOR MISSING STAR OBSERVATIONS

#make new indicator that indicates red vs white based on volatile acidity
wine$VolatileAcidity_IMP_REDFLAG <-
  ifelse(wine$VolatileAcidity_IMP > mean(wine$VolatileAcidity_IMP),1,0)
wine$ResidualSugar_IMP_REDFLAG <-
  ifelse(wine$ResidualSugar_IMP < mean(wine$ResidualSugar_IMP),1,0)
wine$TotalSulfurDioxide_IMP_REDFLAG <-
  ifelse(wine$TotalSulfurDioxide_IMP < mean(wine$TotalSulfurDioxide_IMP),1,0)
wine$Density_IMP_REDFLAG <-
  ifelse(wine$Density_IMP > mean(wine$Density_IMP),1,0)
wine$TallyUp <- wine$VolatileAcidity_IMP_REDFLAG + wine$ResidualSugar_IMP_REDFLAG +
  wine$TotalSulfurDioxide_IMP_REDFLAG + wine$Density_IMP_REDFLAG
wine$Final_REDFLAG <- ifelse(wine$TallyUp > mean(wine$TallyUp),1,0)

#Add Target Flag for 0 sale scenarios
wine$TARGET_Flag <- ifelse(wine$TARGET >0,1,0)
wine$TARGET_AMT <- wine$TARGET - 1
wine$TARGET_AMT <- ifelse(wine$TARGET_Flag == 0,NA,wine$TARGET-1)



##################### Part 3: Model Creation ############################################
#Function for Mean Square Error Calculation
mse <- function(sm)
  mean(sm$residuals^2)

#Function for Mean Absolute Error Calculation
mae <- function(sm)
  mean(abs(sm$residuals))

#########################################################
#########################################################
## FIRST MODEL ... REGULAR LINEAR REGRESSION MODEL#####
lm_fit <- lm(TARGET~ VolatileAcidity + Alcohol_IMP + LabelAppeal +
               STARS_IMP + STARS_IMP_Flag + AcidIndex +
               FreeSulfurDioxide_IMP + TotalSulfurDioxide_IMP +
               Chlorides_IMP + Sulphates_IMP, data = wine)


summary(lm_fit)
coefficients(lm_fit)
wine$fittedLM <-fitted(lm_fit)
AIC(lm_fit)
vif(lm_fit)
glance(lm_fit)



summary(lm_fit)
```

```r
#Table of model VIF
lm_fit.vif <- round(vif(lm_fit), 2)
lm_fit.vif.term <- names(lm_fit.vif)
lm_fit.vif.numbers <- unname(lm_fit.vif)
lm_fit.vif.final <- cbind(lm_fit.vif.term, lm_fit.vif.numbers)
colnames(lm_fit.vif.final) <- c("Term", "VIF")
kable(lm_fit.vif.final)

plot(lm_fit$fitted, lm_fit$residuals,
     xlab = 'Fitted values', ylab = 'Residuals')
abline(h=0, col="red", lty =2)


##############################################################################################
##############################################################################################
## SECOND MODEL ... REGULAR LINEAR REGRESSION MODEL USING STEPWISE VARIABLE SELECTION (AIC)
##############################################################################################
full.model <- lm(TARGET ~ FixedAcidity + VolatileAcidity + CitricAcid + ResidualSugar_IMP +
                   Chlorides_IMP + FreeSulfurDioxide_IMP + TotalSulfurDioxide_IMP + Density +
                   pH_IMP + Sulphates_IMP + Alcohol_IMP + LabelAppeal + AcidIndex + STARS_IMP +
                   VolatileAcidity_IMP_REDFLAG + ResidualSugar_IMP_REDFLAG +
                   TotalSulfurDioxide_IMP_REDFLAG + Density_IMP_REDFLAG +
                   Final_REDFLAG +STARS_IMP_Flag, data=wine)
stepwise_lm <- stepAIC(full.model, direction="both")
stepwise_lm$anova


lm_fit_stepwise <- lm(TARGET ~ ResidualSugar_IMP + Chlorides_IMP + FreeSulfurDioxide_IMP +
                        Density + Sulphates_IMP + Alcohol_IMP + LabelAppeal + AcidIndex +
                        STARS_IMP + VolatileAcidity_IMP_REDFLAG + ResidualSugar_IMP_REDFLAG +
                        TotalSulfurDioxide_IMP_REDFLAG + Density_IMP_REDFLAG + STARS_IMP_Flag,
                      data=wine)

summary(lm_fit_stepwise)
coefficients(lm_fit_stepwise)
wine$fittedLMStepwise <-fitted(lm_fit_stepwise)
AIC(lm_fit_stepwise)
vif(lm_fit_stepwise)
glance(lm_fit_stepwise)


summary(lm_fit_stepwise)


#Table of model VIF
lm_fit_stepwise.vif <- round(vif(lm_fit_stepwise), 2)
lm_fit_stepwise.vif.term <- names(lm_fit_stepwise.vif)
lm_fit_stepwise.vif.numbers <- unname(lm_fit_stepwise.vif)
lm_fit_stepwise.vif.final <- cbind(lm_fit_stepwise.vif.term, lm_fit_stepwise.vif.numbers)
colnames(lm_fit_stepwise.vif.final) <- c("Term", "VIF")
kable(lm_fit_stepwise.vif.final)
```

```
#############################################################################################
#############################################################################################
## THIRD MODEL ... POISSON#################################################################
#############################################################################################

poisson_model <- glm(TARGET ~ ResidualSugar_IMP + Chlorides_IMP + FreeSulfurDioxide_IMP +
                       Density + Sulphates_IMP + Alcohol_IMP + LabelAppeal + AcidIndex +
                       STARS_IMP + VolatileAcidity_IMP_REDFLAG + ResidualSugar_IMP_REDFLAG +
                       TotalSulfurDioxide_IMP_REDFLAG + Density_IMP_REDFLAG + STARS_IMP_Flag,
                     family="poisson"(link="log"), data=wine)


coef(poisson_model)
wine$poisson_yhat <- predict(poisson_model, newdata = wine, type = "response")


summary(poisson_model)


#Table of model VIF
poisson_model.vif <- round(vif(poisson_model), 2)
poisson_model.vif.term <- names(poisson_model.vif)
poisson_model.vif.numbers <- unname(poisson_model.vif)
poisson_model.vif.final <- cbind(poisson_model.vif.term, poisson_model.vif.numbers)
colnames(poisson_model.vif.final) <- c("Term", "VIF")
kable(poisson_model.vif.final)

poisson_model.ratio <- deviance(poisson_model)/df.residual(poisson_model)
cat("Ratio of residual deviance to residual degrees of freedom:", poisson_model.ratio)

#1 - pchisq(summary(poisson_model)$deviance,
#           summary(poisson_model)$df.residual)

#nd = data.frame(Trt = c("A","B"))
#cbind(nd,
#      Mean = predict(poisson_model, newdata = nd, type = "response"),
#      SE = predict(poisson_model, newdata = nd, type = "response", se.fit = T)$se.fit)

poisson_model.95 <- qchisq(0.95, df.residual(poisson_model))
cat("Calculated 95th percentile of the Chi-Squared distriubtion:", poisson_model.95)
cat("Degrees of freedom:", df.residual(poisson_model))

#############################################################################################
#############################################################################################
## FOURTH MODEL ... NEGATIVE BINOMIAL DISTRIBUTION#########################################
#############################################################################################

NBR_Model<-glm.nb(TARGET ~ Chlorides_IMP + FreeSulfurDioxide_IMP +
                    Density + Alcohol_IMP + LabelAppeal + AcidIndex +
                    STARS_IMP + VolatileAcidity_IMP_REDFLAG + ResidualSugar_IMP_REDFLAG +
                    TotalSulfurDioxide_IMP_REDFLAG + Density_IMP_REDFLAG +
                    STARS_IMP_Flag, data=wine)
```

```r
glance(NBR_Model)
wine$NBRphat <- predict(NBR_Model, newdata = wine, type = "response")


summary(NBR_Model)


#Table of model VIF
NBR_Model.vif <- round(vif(NBR_Model), 2)
NBR_Model.vif.term <- names(NBR_Model.vif)
NBR_Model.vif.numbers <- unname(NBR_Model.vif)
NBR_Model.vif.final <- cbind(NBR_Model.vif.term, NBR_Model.vif.numbers)
colnames(NBR_Model.vif.final) <- c("Term", "VIF")
kable(NBR_Model.vif.final)

NBR_Model.ratio <- deviance(NBR_Model)/df.residual(NBR_Model)
cat("Ratio of residual deviance to residual degrees of freedom:", NBR_Model.ratio)

#1 - pchisq(summary(NBR_Model)$deviance,
#           summary(NBR_Model)$df.residual)


#munb <- exp(predict(NBR_Model))
#theta <- NBR_Model$theta
#znb <- (theta/(munb+theta))^theta

# also dnbinom(0, mu=munb, size=theta)
#mean.znbr <- mean(znb)
#compare this to the numbers generated in section 3.5

#deviance(NBR_Model)


##############################################################################################
##############################################################################################
## FIFTH MODEL ... ZERO INFLATED POISSON (ZIP)################################################
##############################################################################################

ZIP_Model<-zeroinfl(TARGET ~ ResidualSugar_IMP + Chlorides_IMP + FreeSulfurDioxide_IMP +
                      Density + Sulphates_IMP + Alcohol_IMP + LabelAppeal + AcidIndex +
                      STARS_IMP + VolatileAcidity_IMP_REDFLAG + ResidualSugar_IMP_REDFLAG +
                      TotalSulfurDioxide_IMP_REDFLAG + Density_IMP_REDFLAG + STARS_IMP_Flag,
                    data=wine)

AIC(ZIP_Model)
wine$ZIPphat <- predict(ZIP_Model, newdata = wine, type = "response")


summary(ZIP_Model)

#From the wine data set, determine the number of zero observations for TARGET
zobs <- wine$TARGET == 0
```

```r
#expected zeros from the Poisson model
zpoi <- exp(-exp(predict(poisson_model))) # or dpois(0,exp(predict(mp)))

#expected zeros from the Negative Binomial model
znb <- exp(-exp(predict(NBR_Model)))

#Numbers of zeros in data set compared against predicted zeros in Poisson
#c(obs=mean(zobs), poi=mean(zpoi), nbr=mean(znb))

##       obs       poi       nbr
## 0.2136772 0.1112855 0.1112349
#The Poisson  and NBR models underestimate the number of zeros

#verify that the ZIP model solves the problem of excess zeros
#predict the zeros with the ZIP model
pr <- predict(ZIP_Model,type="zero")

#predict the counts with the ZIP model
mu <- predict(ZIP_Model,type="count")

#predict the combined probability of zero TARGET with ZIP model
zip <- pr + (1-pr)*exp(-mu) # also predict(mzip,type="prob")[,1]

mean.obs <- mean(zobs) #from wine data set
mean.poi <- mean(zpoi) #poisson
mean.znb <- mean(znb) #negative binomial
mean.zip <- mean(zip) #zero inflated poisson
#0.2203398
#this value is much closer to the observed value (zobs)


zero.obs.compare <- rbind(mean.obs, mean.poi, mean.znb, mean.zip)
colnames(zero.obs.compare) <- c('Probability of Zeros')
rownames(zero.obs.compare) <- c("Observed","Poisson",
                            "Negative Binomal", "Zero Inflated Poisson")
kable(zero.obs.compare, caption = 'Model Performance.')

ZIP_model.95 <- qchisq(0.95, df.residual(ZIP_Model))
cat("Calculated 95th percentile of the Chi-Squared distriubtion:", ZIP_model.95)
cat("Degrees of freedom:", df.residual(ZIP_Model))


##############################################################################################
##############################################################################################
## 6TH MODEL ... ZERO INFLATED NEGATIVE BINOMIAL REGRESSION (ZINB)#########################
##############################################################################################

ZINB_Model<-zeroinfl(TARGET ~ ResidualSugar_IMP + Chlorides_IMP + FreeSulfurDioxide_IMP +
                       Density + Sulphates_IMP + Alcohol_IMP + LabelAppeal + AcidIndex +
                       STARS_IMP + VolatileAcidity_IMP_REDFLAG + ResidualSugar_IMP_REDFLAG +
                       TotalSulfurDioxide_IMP_REDFLAG + Density_IMP_REDFLAG +
                       STARS_IMP_Flag, dist = "negbin", EM=TRUE,
                     data=wine)
```

```
AIC(ZINB_Model)
wine$ZINBphat <- predict(ZINB_Model, newdata = wine, type = "response")



summary(ZINB_Model)


#verify that the ZIP model solves the problem of excess zeros
#predict the zeros with the ZINB model
pr2 <- predict(ZINB_Model,type="zero")

#predict the counts with the ZINB model
mu2 <- predict(ZINB_Model,type="count")

#predict the combined probability of zero TARGET
zip2 <- pr2 + (1-pr2)*exp(-mu2) # also predict(mzip,type="prob")[,1]

mean.zinb <- mean(zip2) #zero inflated NBR

zero.obs.compare2 <- rbind(mean.obs, mean.poi, mean.znb, mean.zip, mean.zinb)
colnames(zero.obs.compare2) <- c('Probability of Zeros')
rownames(zero.obs.compare2) <- c("Observed","Poisson",
                            "Negative Binomal", "Zero Inflated Poisson",
                            "Zero Inflated Negative Binomal")
kable(zero.obs.compare2, caption = 'Model Performance.')



##############################################################################################
##############################################################################################
## SEVENTH MODEL ... ZERO INFLATED POISSON (ZIP) II ##########################################
##############################################################################################

ZIP_Model2<-zeroinfl(TARGET ~  Alcohol_IMP + LabelAppeal + AcidIndex +
                       STARS_IMP + VolatileAcidity_IMP_REDFLAG + Density_IMP_REDFLAG +
                       STARS_IMP_Flag | -1 + ResidualSugar_IMP + FreeSulfurDioxide_IMP +
                       Density + Sulphates_IMP + Alcohol_IMP + LabelAppeal + AcidIndex +
                       STARS_IMP + VolatileAcidity_IMP_REDFLAG + ResidualSugar_IMP_REDFLAG +
                       TotalSulfurDioxide_IMP_REDFLAG + Density_IMP_REDFLAG + STARS_IMP_Flag ,
                     data=wine)

AIC(ZIP_Model2)
wine$ZIP2phat <- predict(ZIP_Model2, newdata = wine, type = "response")


summary(ZIP_Model2)

#verify that the ZIP model solves the problem of excess zeros
#predict the zeros with the zero inflated poisson II
pr3 <- predict(ZIP_Model2,type="zero")
```

```r
#predict the counts with the zero inflated poisson II
mu3 <- predict(ZIP_Model2,type="count")

#predict the combined probability of zero TARGET
zip3 <- pr3 + (1-pr3)*exp(-mu3) # also predict(mzip,type="prob")[,1]

mean.zip.2 <- mean(zip3) #zero inflated poisson II

zero.obs.compare2 <- rbind(mean.obs, mean.poi, mean.znb, mean.zip,
                           mean.zinb, mean.zip.2)
colnames(zero.obs.compare2) <- c('Probability of Zeros')
rownames(zero.obs.compare2) <- c("Observed","Poisson",
                            "Negative Binomal", "Zero Inflated Poisson",
                            "Zero Inflated Negative Binomal", "Zero Inflated Poisson II")
kable(zero.obs.compare2, caption = 'Model Performance.')


######################### Section 4: Model Selection ###################################
#Model Performance function
Model.fit <- function(model){
  AIC.fit <- round(AIC(model), 0)
  BIC.fit <- round(BIC(model), 0)
  MSE.fit <- round(mean(model$residuals^2), 0)
  MAE.fit <- round(mean(abs(model$residuals)), 0)
  Deviance.fit <- round(-2*logLik(model, REML = TRUE))
  logLik.fit <- round(logLik(model), 0)
  return(c(AIC.fit, BIC.fit, MSE.fit, MAE.fit, Deviance.fit, logLik.fit))
}

#Model Comparison
#Table of AIC, BIC, MSE, MAE, deviance, logLik for each Model
lm_fit.fit <- Model.fit(lm_fit)
lm_fit_stepwise.fit <- Model.fit(lm_fit_stepwise)
poisson_model.fit <- Model.fit(poisson_model)
NBR_Model.fit <- Model.fit(NBR_Model)
ZIP_Model.fit <- Model.fit(ZIP_Model)
ZINB_Model.fit <- Model.fit(ZINB_Model)
ZIP_Model2.fit <- Model.fit(ZIP_Model2)

fit.compare <- rbind(lm_fit.fit, lm_fit_stepwise.fit, poisson_model.fit,
                     NBR_Model.fit, ZIP_Model.fit, ZINB_Model.fit, ZIP_Model2.fit)
colnames(fit.compare) <- c('AIC', 'BIC', 'MSE', 'MAE', 'Deviance', 'logLik')
rownames(fit.compare) <- c("Linear", "Stepwise", "Poisson",
                            "Negative Binomal", "Zero Inflated Poisson",
                           "Zero Inflated NB", "Zero Inflated Poisson II")
kable(fit.compare, caption = 'Model Performance.')

kable(zero.obs.compare2, caption = 'Model Performance.')


lm_fit.glance <- round(glance(lm_fit), 4)
lm_fit_stepwise.glance <- round(glance(lm_fit_stepwise), 4)
poisson_model.glance <- round(glance(poisson_model), 4)
```

```r
NBR_Model.glance <- round(glance(NBR_Model), 4)

Model.glance1 <- rbind(lm_fit.glance, lm_fit_stepwise.glance)
rownames(Model.glance1) <- c("Linear", "Stepwise")
kable(Model.glance1,
      format = "latex", booktabs = T) %>%
  kable_styling(latex_options = c("striped", "scale_down"))

Model.glance2 <- rbind(poisson_model.glance, NBR_Model.glance)
rownames(Model.glance2) <- c("Poisson", "Negative Binomal")
kable(Model.glance2,
      format = "latex", booktabs = T) %>%
  kable_styling(latex_options = c("striped", "scale_down"))

#glance does not work with zero inflated models


#AIC with rank component
lm_fit$rank <- length(coef(lm_fit)) # add a rank component
lm_fit_stepwise$rank <- length(coef(lm_fit_stepwise)) # add a rank component
poisson_model$rank <- length(coef(poisson_model)) # add a rank component
NBR_Model$rank <- length(coef(NBR_Model)) # add a rank component
ZIP_Model$rank <- length(coef(ZIP_Model)) # add a rank component
ZINB_Model$rank <- length(coef(ZINB_Model)) # add a rank component
ZIP_Model2$rank <- length(coef(ZIP_Model2)) # add a rank component

aic <- function(model) -2*logLik(model) + 2*model$rank

sapply(list(lm_fit, lm_fit_stepwise, poisson_model, NBR_Model,
            ZIP_Model, ZINB_Model, ZIP_Model2), aic)


#################### Score Test Data ##########################
#### Part 5:  Score Model on Test Data set and output csv file
#Read File in from your working directory
setwd("~/Desktop/R/")
wine_test = read.csv("Wine_Test.csv")  # read csv file

###################### Data Preparation ########################################
#create IMP versions of each independent variable
#The IMP columns are copies of the original columns
wine_test$FixedAcidity_IMP <- wine_test$FixedAcidity
wine_test$VolatileAcidity_IMP <- wine_test$VolatileAcidity
wine_test$CitricAcid_IMP <- wine_test$CitricAcid
wine_test$ResidualSugar_IMP <- wine_test$ResidualSugar
wine_test$Chlorides_IMP <- wine_test$Chlorides
wine_test$FreeSulfurDioxide_IMP <- wine_test$FreeSulfurDioxide
wine_test$TotalSulfurDioxide_IMP <- wine_test$TotalSulfurDioxide
wine_test$Density_IMP <- wine_test$Density
wine_test$pH_IMP <- wine_test$pH
wine_test$Sulphates_IMP <- wine_test$Sulphates
wine_test$Alcohol_IMP <- wine_test$Alcohol
wine_test$LabelAppeal_IMP <- wine_test$LabelAppeal
```

```
wine_test$AcidIndex_IMP <- wine_test$AcidIndex
wine_test$STARS_IMP <- wine_test$STARS

#replace NA's in each IMP column with median
wine_test$FixedAcidity_IMP[which(is.na(wine_test$FixedAcidity_IMP))] <-
  median(wine_test$FixedAcidity_IMP,na.rm = TRUE)
wine_test$VolatileAcidity_IMP[which(is.na(wine_test$VolatileAcidity_IMP))] <-
  median(wine_test$VolatileAcidity_IMP,na.rm = TRUE)
wine_test$CitricAcid_IMP[which(is.na(wine_test$CitricAcid_IMP))] <-
  median(wine_test$CitricAcid_IMP,na.rm = TRUE)
wine_test$ResidualSugar_IMP[which(is.na(wine_test$ResidualSugar_IMP))] <-
  median(wine_test$ResidualSugar_IMP,na.rm = TRUE)
wine_test$Chlorides_IMP[which(is.na(wine_test$Chlorides_IMP))] <-
  median(wine_test$Chlorides_IMP,na.rm = TRUE)
wine_test$FreeSulfurDioxide_IMP[which(is.na(wine_test$FreeSulfurDioxide_IMP))] <-
  median(wine_test$FreeSulfurDioxide_IMP,na.rm = TRUE)
wine_test$TotalSulfurDioxide_IMP[which(is.na(wine_test$TotalSulfurDioxide_IMP))] <-
  median(wine_test$TotalSulfurDioxide_IMP,na.rm = TRUE)
wine_test$Density_IMP[which(is.na(wine_test$Density_IMP))] <-
  median(wine_test$Density_IMP,na.rm = TRUE)
wine_test$pH_IMP[which(is.na(wine_test$pH_IMP))] <-
  median(wine_test$pH_IMP,na.rm = TRUE)
wine_test$Sulphates_IMP[which(is.na(wine_test$Sulphates_IMP))] <-
  median(wine_test$Sulphates_IMP,na.rm = TRUE)
wine_test$Alcohol_IMP[which(is.na(wine_test$Alcohol_IMP))] <-
  median(wine_test$Alcohol_IMP,na.rm = TRUE)
wine_test$LabelAppeal_IMP[which(is.na(wine_test$LabelAppeal_IMP))] <-
  median(wine_test$LabelAppeal_IMP,na.rm = TRUE)
wine_test$AcidIndex_IMP[which(is.na(wine_test$AcidIndex_IMP))] <-
  median(wine_test$AcidIndex_IMP,na.rm = TRUE)
wine_test$STARS_IMP[which(is.na(wine_test$STARS_IMP))] <-
  median(wine_test$STARS_IMP,na.rm = TRUE)

#flag NA values with new field
#first, create new field
#second, replace NA's with 1 else 0

wine_test$ResidualSugar_IMP_Flag <- wine_test$ResidualSugar
wine_test$Chlorides_IMP_Flag <- wine_test$Chlorides
wine_test$FreeSulfurDioxide_IMP_Flag <- wine_test$FreeSulfurDioxide
wine_test$TotalSulfurDioxide_IMP_Flag <- wine_test$TotalSulfurDioxide
wine_test$pH_IMP_Flag <- wine_test$pH
wine_test$Sulphates_IMP_Flag <- wine_test$Sulphates
wine_test$Alcohol_IMP_Flag <- wine_test$Alcohol
wine_test$STARS_IMP_Flag <- wine_test$STARS


#NEW BINARY FIELDS TO SHOW na's
wine_test$ResidualSugar_IMP_Flag <-
  ifelse(is.na(wine_test$ResidualSugar_IMP_Flag)==TRUE, 1,0)
wine_test$Chlorides_IMP_Flag <-
  ifelse(is.na(wine_test$Chlorides_IMP_Flag)==TRUE, 1,0)
wine_test$FreeSulfurDioxide_IMP_Flag <-
```

```
  ifelse(is.na(wine_test$FreeSulfurDioxide_IMP_Flag)==TRUE, 1,0)
wine_test$TotalSulfurDioxide_IMP_Flag <-
  ifelse(is.na(wine_test$TotalSulfurDioxide_IMP_Flag)==TRUE, 1,0)
wine_test$pH_IMP_Flag <-
  ifelse(is.na(wine_test$pH_IMP_Flag)==TRUE, 1,0)
wine_test$Sulphates_IMP_Flag <-
  ifelse(is.na(wine_test$Sulphates_IMP_Flag)==TRUE, 1,0)
wine_test$Alcohol_IMP_Flag <-
  ifelse(is.na(wine_test$Alcohol_IMP_Flag)==TRUE, 1,0)
wine_test$STARS_IMP_Flag <-
  ifelse(is.na(wine_test$STARS_IMP_Flag)==TRUE, 1,0) #LOOK FOR MISSING STAR OBSERVATIONS

#make new indicator that indicates red vs white based on volatile acidity
wine_test$VolatileAcidity_IMP_REDFLAG <-
  ifelse(wine_test$VolatileAcidity_IMP > mean(wine_test$VolatileAcidity_IMP),1,0)
wine_test$ResidualSugar_IMP_REDFLAG <-
  ifelse(wine_test$ResidualSugar_IMP < mean(wine_test$ResidualSugar_IMP),1,0)
wine_test$TotalSulfurDioxide_IMP_REDFLAG <-
  ifelse(wine_test$TotalSulfurDioxide_IMP < mean(wine_test$TotalSulfurDioxide_IMP),1,0)
wine_test$Density_IMP_REDFLAG <-
  ifelse(wine_test$Density_IMP > mean(wine_test$Density_IMP),1,0)
wine_test$TallyUp <- wine_test$VolatileAcidity_IMP_REDFLAG + wine_test$ResidualSugar_IMP_REDFLAG +
  wine_test$TotalSulfurDioxide_IMP_REDFLAG + wine_test$Density_IMP_REDFLAG
wine_test$Final_REDFLAG <- ifelse(wine_test$TallyUp > mean(wine_test$TallyUp),1,0)

#Add Target Flag for 0 sale scenarios
wine_test$TARGET_Flag <- ifelse(wine_test$TARGET >0,1,0)
wine_test$TARGET_AMT <- wine_test$TARGET - 1
wine_test$TARGET_AMT <- ifelse(wine_test$TARGET_Flag == 0,NA,wine_test$TARGET-1)

# Again, double-checking to make sure we don't have any NA's in our Test Data Set
summary(wine_test)

#############################################################################################
#############################################################################################
## CHAMPION MODEL ... ZERO INFLATED POISSON (ZIP)#############################################
#############################################################################################

ZIP_Model<-zeroinfl(TARGET ~ ResidualSugar_IMP + Chlorides_IMP + FreeSulfurDioxide_IMP +
                      Density + Sulphates_IMP + Alcohol_IMP + LabelAppeal + AcidIndex +
                      STARS_IMP + VolatileAcidity_IMP_REDFLAG + ResidualSugar_IMP_REDFLAG +
                      TotalSulfurDioxide_IMP_REDFLAG + Density_IMP_REDFLAG + STARS_IMP_Flag,
                    data=wine)

wine_test$P_TARGET <- predict(ZIP_Model, newdata = wine_test, type = "response")

summary(wine_test)

select <- dplyr::select

# Scored Data File
scores <- wine_test[c("INDEX","P_TARGET")]
write.csv(scores, file = "U3_Scored1.csv", col.names = TRUE)
```

```
#write.xlsx(scores, file = "U3_Scored1.xlsx",
#           sheetName = "Scored Data File", col.names = TRUE)
```