DATA 621 Week 5

Vinicio Haro

CUNY SPS

DATA 621 Week 5

Vinicio Haro

July 9, 2018

We are given a dataset containing information on commercially avilable wines. The variables describe the chemical attributes of wine being sold. Our response variable in this case is the number of wine cases purchased by wine distribution companies. Each record in the data represents the number of cases sold for a wine with the specific chemical attributes.

Our goal is to model the data and predict the number of cases sold as a function of wine's chemical attributes. The use case is for a wine manufacturer to adjust their wine offerings in order to maximize sales. The model of use for this case study will be a count regression model.

Read the data in

```
##    INDEX TARGET FixedAcidity VolatileAcidity CitricAcid
ResidualSugar
## 1      1      3          3.2           1.160      -0.98
54.20
## 2      2      3          4.5           0.160      -0.81
26.10
## 3      4      5          7.1           2.640      -0.88
14.80
## 4      5      3          5.7           0.385       0.04
18.80
## 5      6      4          8.0           0.330      -1.26
9.40
## 6      7      0         11.3           0.320       0.59
2.20
## 7      8      0          7.7           0.290      -0.40
21.50
## 8     11      4          6.5          -1.220       0.34
1.40
## 9     12      3         14.8           0.270       1.05
11.25
## 10    13      6          5.5          -0.220       0.39
1.80
##    Chlorides FreeSulfurDioxide TotalSulfurDioxide Density   pH
Sulphates
## 1     -0.567                NA                268 0.99280 3.33     -
0.59
## 2     -0.425                15               -327 1.02792 3.38
0.70
## 3      0.037               214                142 0.99518 3.12
0.48
```

```
## 4     -0.425                22                115 0.99640 2.24
1.83
## 5         NA              -167                108 0.99457 3.12
1.77
## 6      0.556               -37                 15 0.99940 3.20
1.29
## 7      0.060               287                156 0.99572 3.49
1.21
## 8      0.040               523                551 1.03236 3.20
NA
## 9     -0.007              -213                 NA 0.99620 4.93
0.26
## 10    -0.277                62                180 0.94724 3.09
0.75
##    Alcohol LabelAppeal AcidIndex STARS
## 1      9.9           0         8     2
## 2       NA          -1         7     3
## 3     22.0          -1         8     3
## 4      6.2          -1         6     1
## 5     13.7           0         9     2
## 6     15.4           0        11    NA
## 7     10.3           0         8    NA
## 8     11.6           1         7     3
## 9     15.0           0         6    NA
## 10    12.6           0         8     4
```

I) EDA

How many records and variables?

```
##  [1] "TARGET"             "FixedAcidity"       "VolatileAcidity"
##  [4] "CitricAcid"         "ResidualSugar"      "Chlorides"
##  [7] "FreeSulfurDioxide"  "TotalSulfurDioxide" "Density"
## [10] "pH"                 "Sulphates"          "Alcohol"
## [13] "LabelAppeal"        "AcidIndex"          "STARS"

## 'data.frame':    12795 obs. of  15 variables:
##  $ TARGET            : int  3 3 5 3 4 0 0 4 3 6 ...
##  $ FixedAcidity      : num  3.2 4.5 7.1 5.7 8 11.3 7.7 6.5 14.8 5.5
...
##  $ VolatileAcidity   : num  1.16 0.16 2.64 0.385 0.33 0.32 0.29 -
1.22 0.27 -0.22 ...
##  $ CitricAcid        : num  -0.98 -0.81 -0.88 0.04 -1.26 0.59 -0.4
0.34 1.05 0.39 ...
##  $ ResidualSugar     : num  54.2 26.1 14.8 18.8 9.4 ...
##  $ Chlorides         : num  -0.567 -0.425 0.037 -0.425 NA 0.556 0.06
0.04 -0.007 -0.277 ...
##  $ FreeSulfurDioxide : num  NA 15 214 22 -167 -37 287 523 -213 62
...
##  $ TotalSulfurDioxide: num  268 -327 142 115 108 15 156 551 NA 180
...
```

```
##  $ Density           : num  0.993 1.028 0.995 0.996 0.995 ...
##  $ pH                : num  3.33 3.38 3.12 2.24 3.12 3.2 3.49 3.2
4.93 3.09 ...
##  $ Sulphates         : num  -0.59 0.7 0.48 1.83 1.77 1.29 1.21 NA
0.26 0.75 ...
##  $ Alcohol           : num  9.9 NA 22 6.2 13.7 15.4 10.3 11.6 15
12.6 ...
##  $ LabelAppeal       : int  0 -1 -1 -1 0 0 0 1 0 0 ...
##  $ AcidIndex         : int  8 7 8 6 9 11 8 7 6 8 ...
##  $ STARS             : int  2 3 3 1 2 NA NA 3 NA 4 ...
```

For this particular study, we will be implimenting new features into our EDA using a package called DataExplorer. More information can be found here https://datascienceplus.com/blazing-fast-eda-in-r-with-dataexplorer/

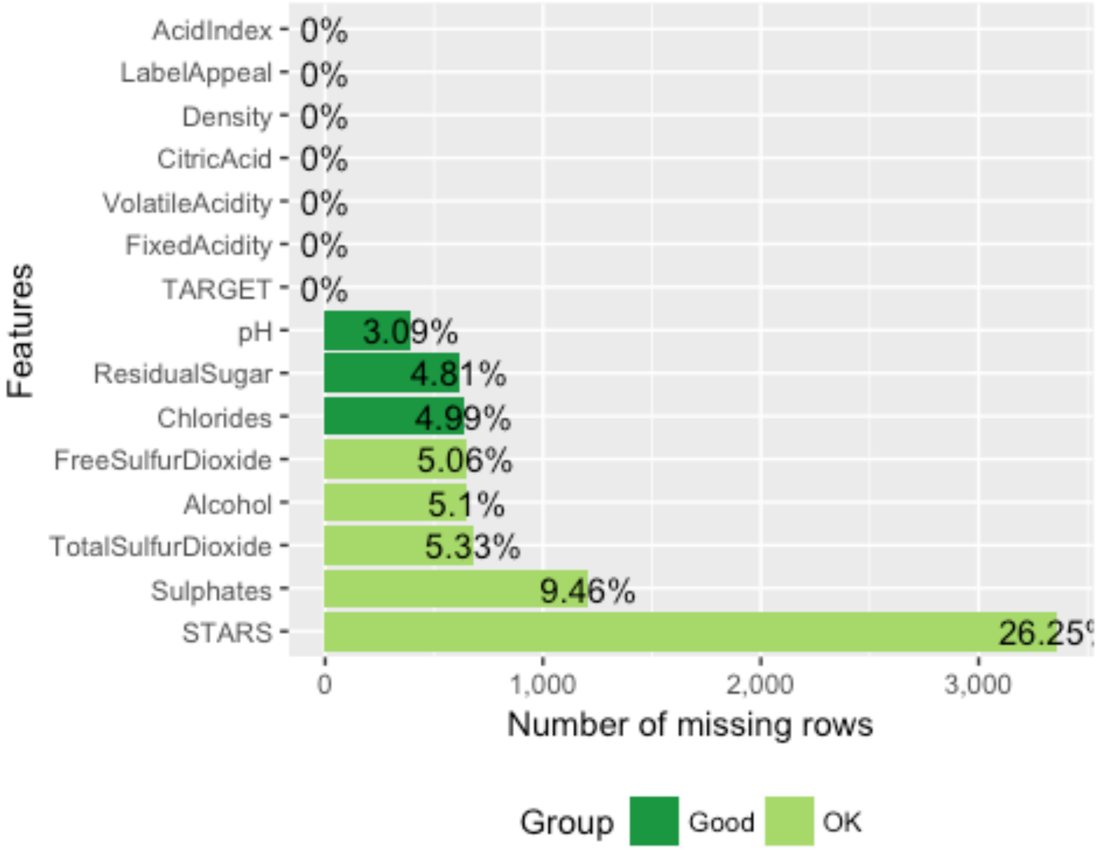
Using DataExplorer, we can examine the dimensions of each variable.

```
## Warning: package 'DataExplorer' was built under R version 3.4.4
```

This interactive chart details the data type for each variable in addition to number or rows and variables within the dataset.

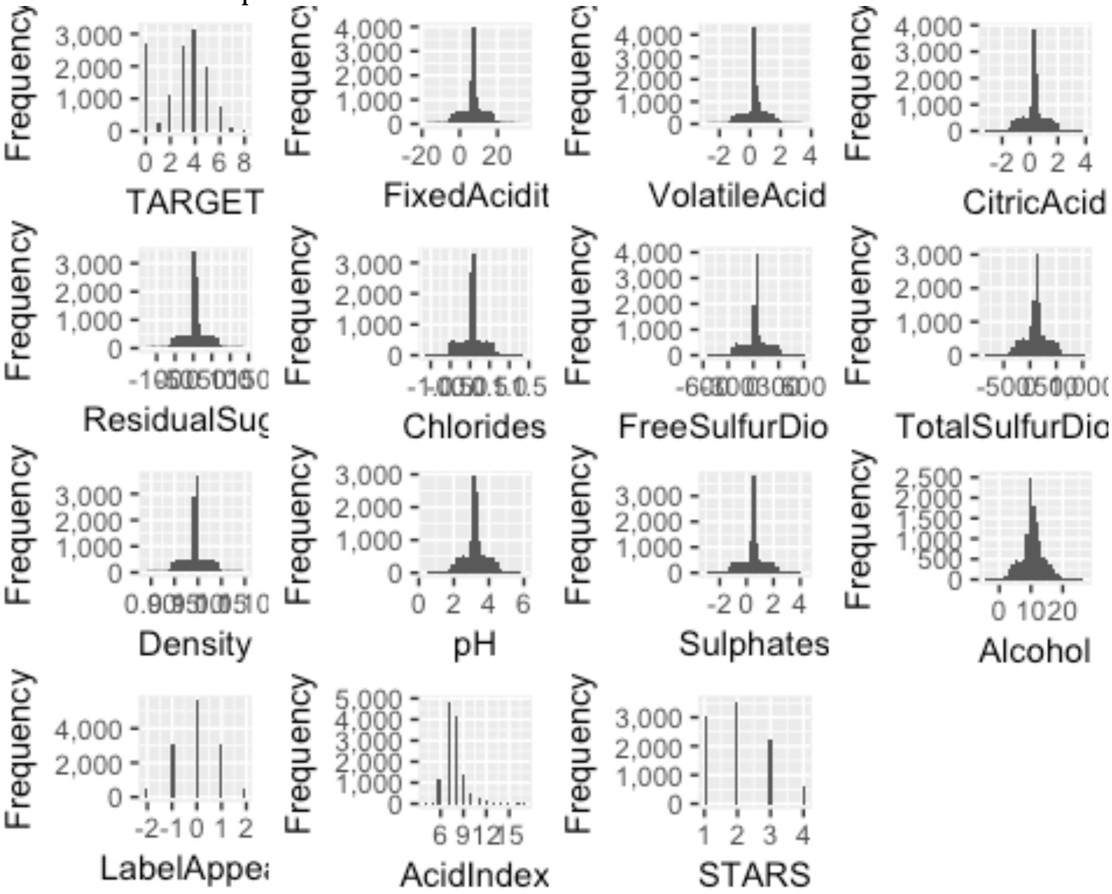
There are 12,795 records and 15 variables. The index variable can be removed all together since it only serves as a row number. The data types seem have correct data types.

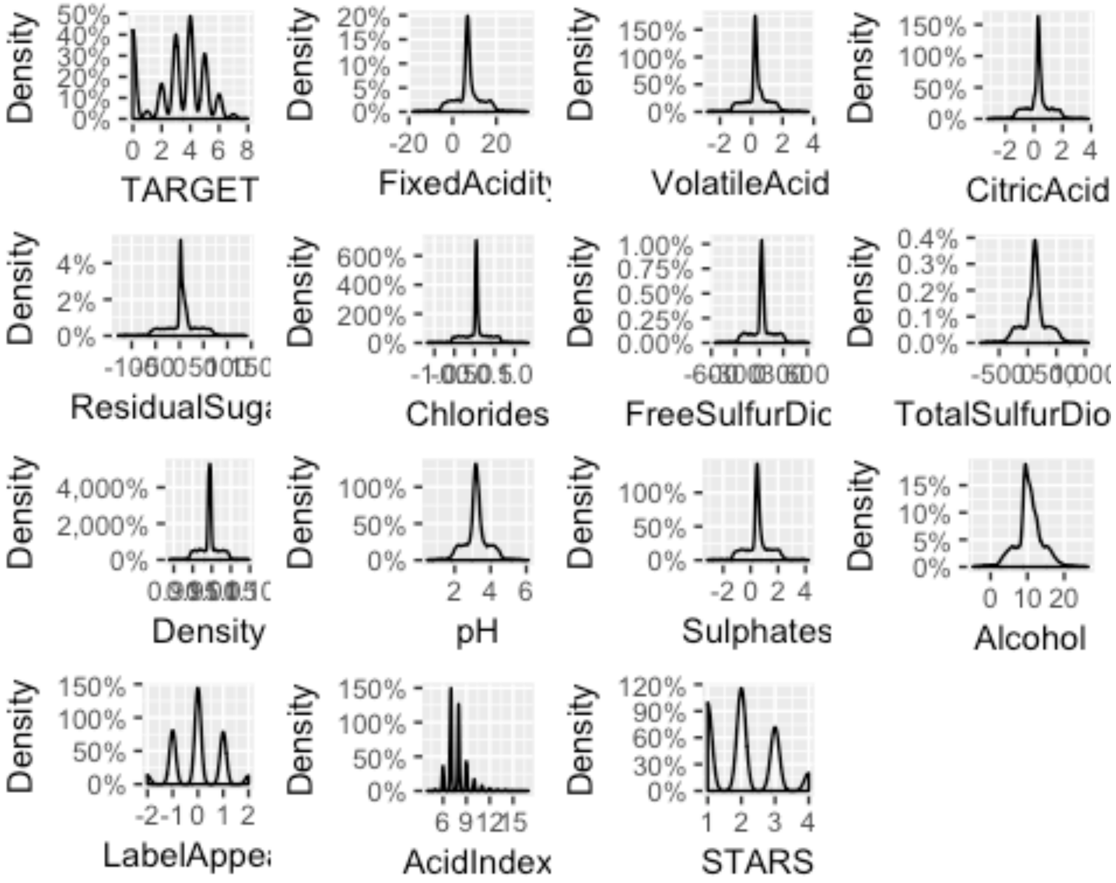
Missing value analysis



STARS is missing over 25% of its data. Sulphates has 9% missing data. The missing variables will be treated in the data preperation step.

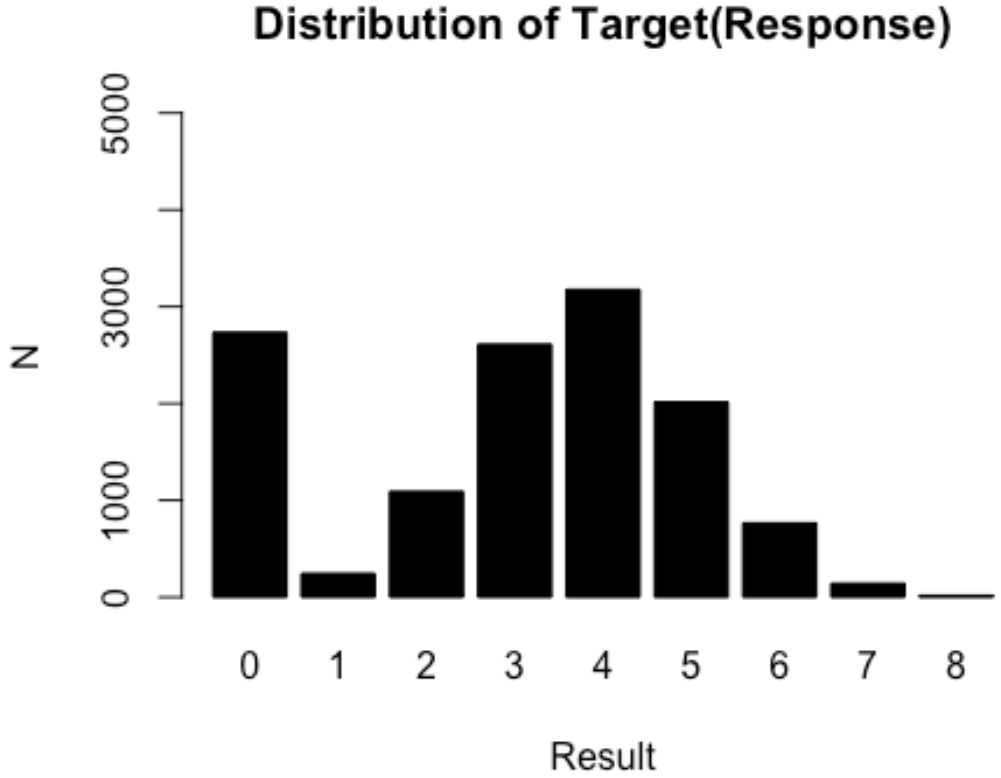
Lets examine the spread of each variable





It looks like the spread of most variables are even, with a near normal distribution. AcidIndex has a right skew. Most variables have significant peaks. This will be closely analyzed with an outlier analysis.

Lets closely examine the distribution of the response variable



The distribution of the response variable is close to normal, however there are a significant number of records that have the number 0.

Overall Summary of the data

```
##      TARGET       FixedAcidity     VolatileAcidity     CitricAcid
##  Min.   :0.000   Min.   :-18.100   Min.   :-2.7900   Min.   :-3.2400
##  1st Qu.:2.000   1st Qu.:  5.200   1st Qu.: 0.1300   1st Qu.: 0.0300
##  Median :3.000   Median :  6.900   Median : 0.2800   Median : 0.3100
##  Mean   :3.029   Mean   :  7.076   Mean   : 0.3241   Mean   : 0.3084
##  3rd Qu.:4.000   3rd Qu.:  9.500   3rd Qu.: 0.6400   3rd Qu.: 0.5800
##  Max.   :8.000   Max.   : 34.400   Max.   : 3.6800   Max.   : 3.8600
##
##  ResidualSugar        Chlorides       FreeSulfurDioxide
TotalSulfurDioxide
##  Min.   :-127.800   Min.   :-1.1710   Min.   :-555.00   Min.   :-
823.0
##  1st Qu.:  -2.000   1st Qu.:-0.0310   1st Qu.:   0.00   1st Qu.:
27.0
##  Median :   3.900   Median : 0.0460   Median :  30.00   Median :
123.0
##  Mean   :   5.419   Mean   : 0.0548   Mean   :  30.85   Mean   :
```

```
120.7
##  3rd Qu.:  15.900   3rd Qu.: 0.1530   3rd Qu.:  70.00   3rd Qu.:
208.0
##  Max.   : 141.150   Max.   : 1.3510   Max.   : 623.00   Max.   
:1057.0
##  NA's   :616        NA's   :638        NA's   :647        NA's   :682
##     Density             pH          Sulphates          Alcohol
##  Min.   :0.8881   Min.   :0.480   Min.   :-3.1300   Min.   :-4.70
##  1st Qu.:0.9877   1st Qu.:2.960   1st Qu.: 0.2800   1st Qu.: 9.00
##  Median :0.9945   Median :3.200   Median : 0.5000   Median :10.40
##  Mean   :0.9942   Mean   :3.208   Mean   : 0.5271   Mean   :10.49
##  3rd Qu.:1.0005   3rd Qu.:3.470   3rd Qu.: 0.8600   3rd Qu.:12.40
##  Max.   :1.0992   Max.   :6.130   Max.   : 4.2400   Max.   :26.50
##                   NA's   :395     NA's   :1210      NA's   :653
##   LabelAppeal          AcidIndex          STARS
##  Min.   :-2.000000   Min.   : 4.000   Min.   :1.000
##  1st Qu.:-1.000000   1st Qu.: 7.000   1st Qu.:1.000
##  Median : 0.000000   Median : 8.000   Median :2.000
##  Mean   :-0.009066   Mean   : 7.773   Mean   :2.042
##  3rd Qu.: 1.000000   3rd Qu.: 8.000   3rd Qu.:3.000
##  Max.   : 2.000000   Max.   :17.000   Max.   :4.000
##                                        NA's   :3359
```

There are several variables that should not contain any negative entires. This brings the problem of data collection into consideration. This will have to be furthur investigated in the data perperation step.

Lets point out the summary of the response variable

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   2.000   3.000   3.029   4.000   8.000

## [1] 3.710895
```

The mean is nearly equal to the variance. This is a strong indicator

How many negative values does each variable contain?

```
##             TARGET       FixedAcidity    VolatileAcidity
##                  0               1621               2827
##         CitricAcid      ResidualSugar          Chlorides
##               2966                 NA                 NA
##  FreeSulfurDioxide TotalSulfurDioxide            Density
##                 NA                 NA                  0
##                 pH          Sulphates            Alcohol
##                 NA                 NA                 NA
##        LabelAppeal          AcidIndex              STARS
##               3640                  0                 NA
```

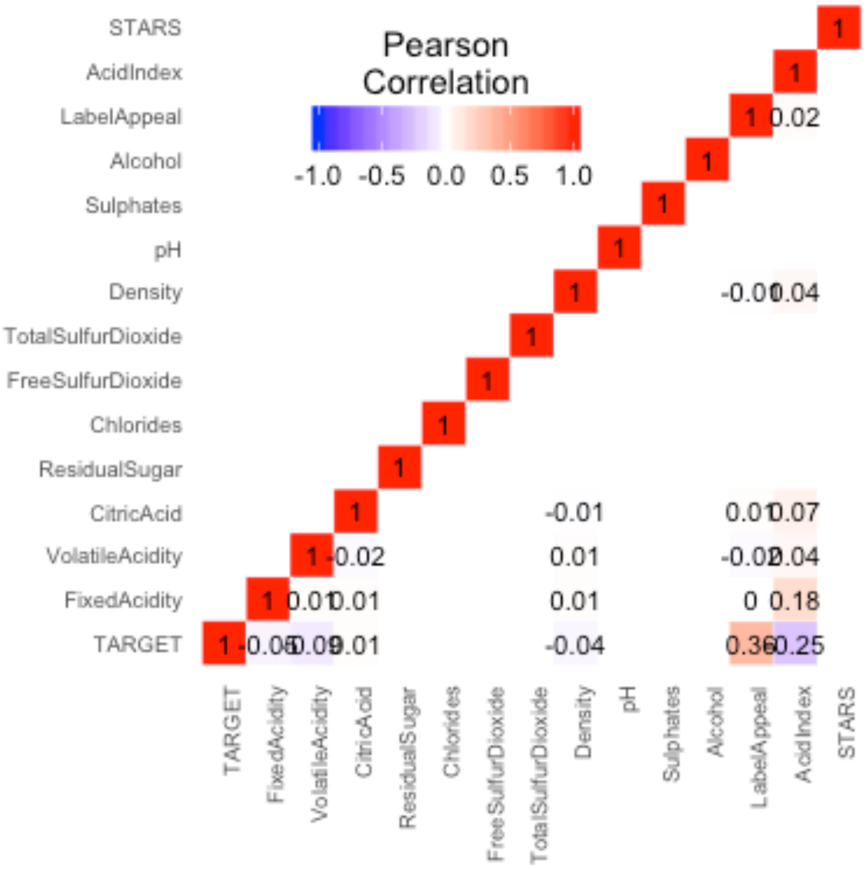
Label Appeal can have negative values in its domain. A negative value means customers do not like the design. Several chemical attributes have negative numbers. 23% of the values in Citric Acid are negative. In our data prep, we will confirm if negative numbers are in the domain of these variables based on the definition of their chemical attributes.

How does each variable correlate to the response variable?

```
##             TARGET       FixedAcidity    VolatileAcidity
##        1.000000000       -0.049010939       -0.088793212
##         CitricAcid      ResidualSugar          Chlorides
##        0.008684633                 NA                 NA
##  FreeSulfurDioxide TotalSulfurDioxide            Density
##                 NA                 NA       -0.035517502
##                 pH          Sulphates            Alcohol
##                 NA                 NA                 NA
##        LabelAppeal          AcidIndex              STARS
##        0.356500469       -0.246049449                 NA
```

There are several variables that do not have a correlation with the number of sales. The highest positive correlation is label appeal which is the marketing score. This correlation makes sense as one would imagine a higher marketing score leads to more sales. Acid index has the strongest negative correlation however it is difficult to see the connection without prior knowledge of wine attributes.

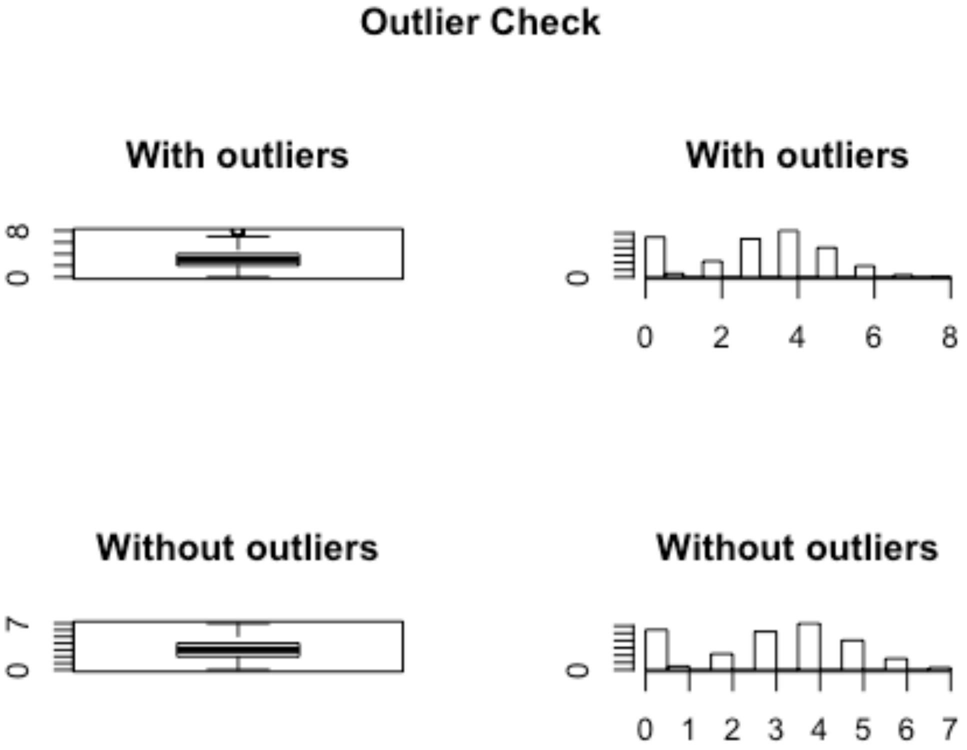
How do the other variables correlate with each other?



There are numerous variables that do not have any correlation at all with any other variable. Through modeling, we will determine if they are significant. STARS has no correlation with any variable except its self. STARS also has the highest percentage of missing data. It's hard to say if the missing data is a reason why it does not correlate with other predictors. This only gives evidence in support of removing STARS all together.

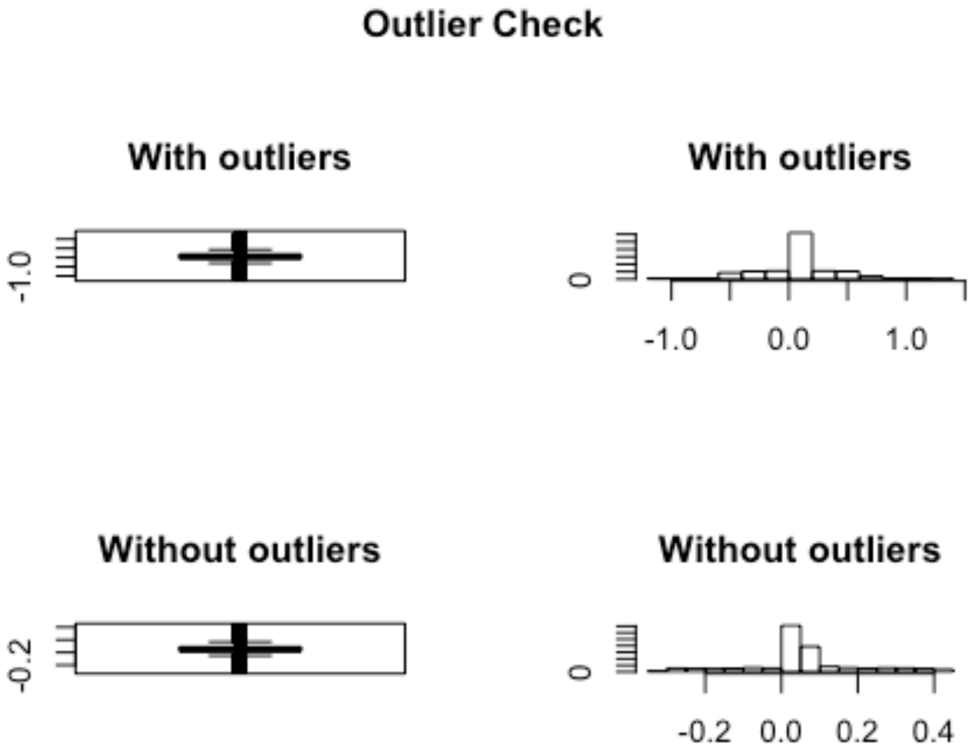
Outlier Analysis

Target



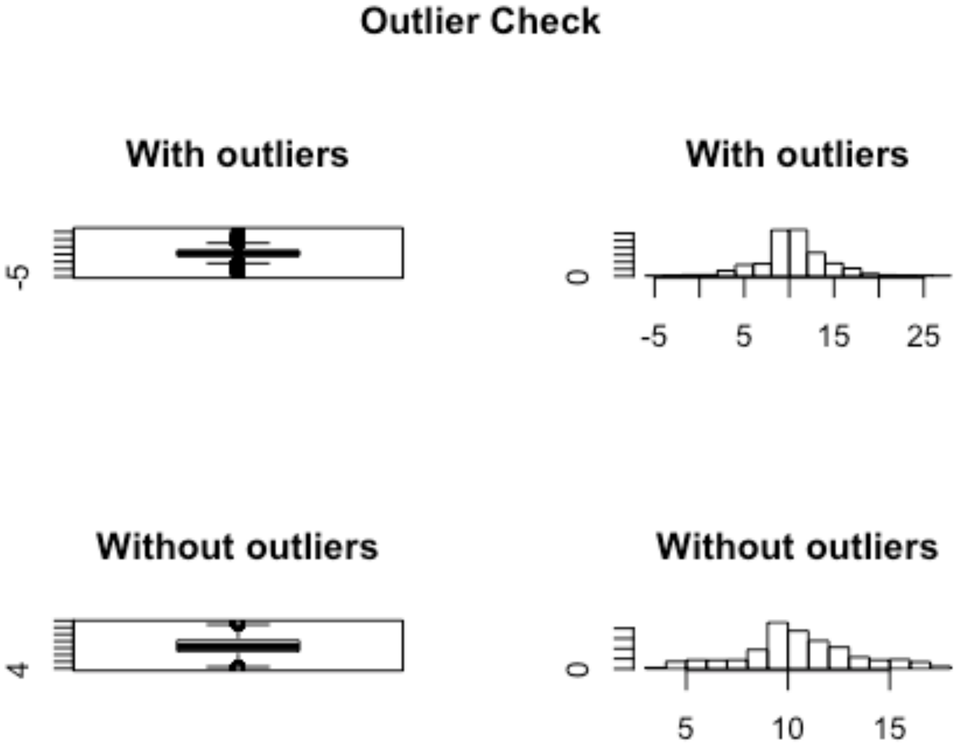
```
## Outliers identified: 17 nPropotion (%) of outliers: 0.1 nMean of the
outliers: 8 nMean without removing outliers: 3.03 nMean if we remove
outliers: 3.02 nDo you want to remove outliers and to replace with NA?
[yes/no]:
## Nothing changed n
```

Chlorides



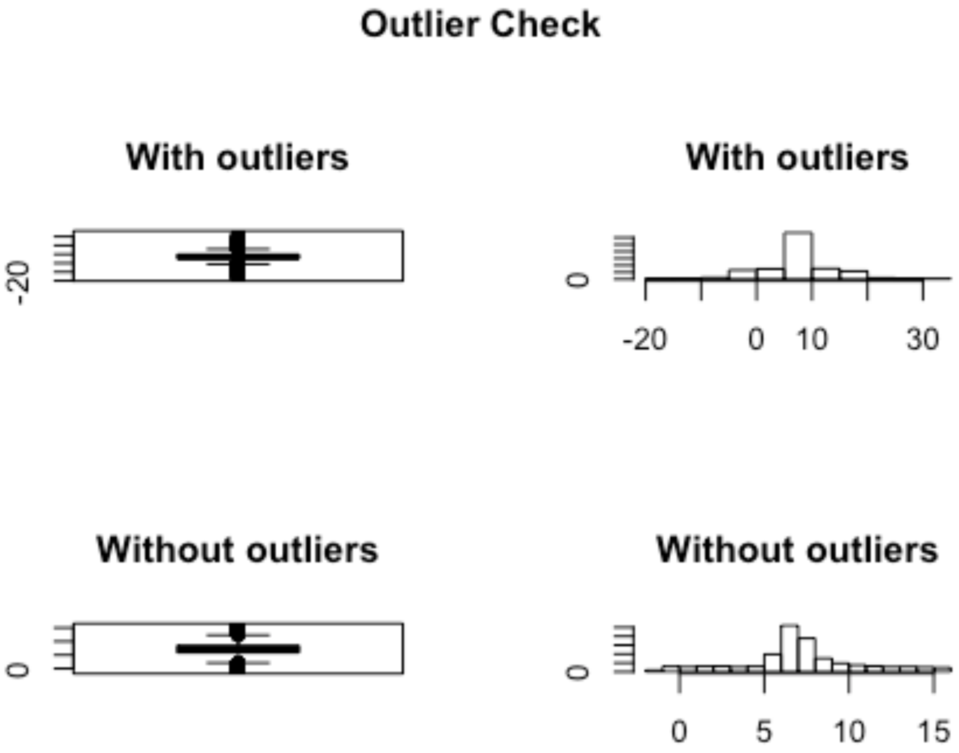
```
## Outliers identified: 3021 nPropotion (%) of outliers: 33.1 nMean of
the outliers: 0.05 nMean without removing outliers: 0.05 nMean if we
remove outliers: 0.06 nDo you want to remove outliers and to replace
with NA? [yes/no]:
## Nothing changed n
```

Alcohol



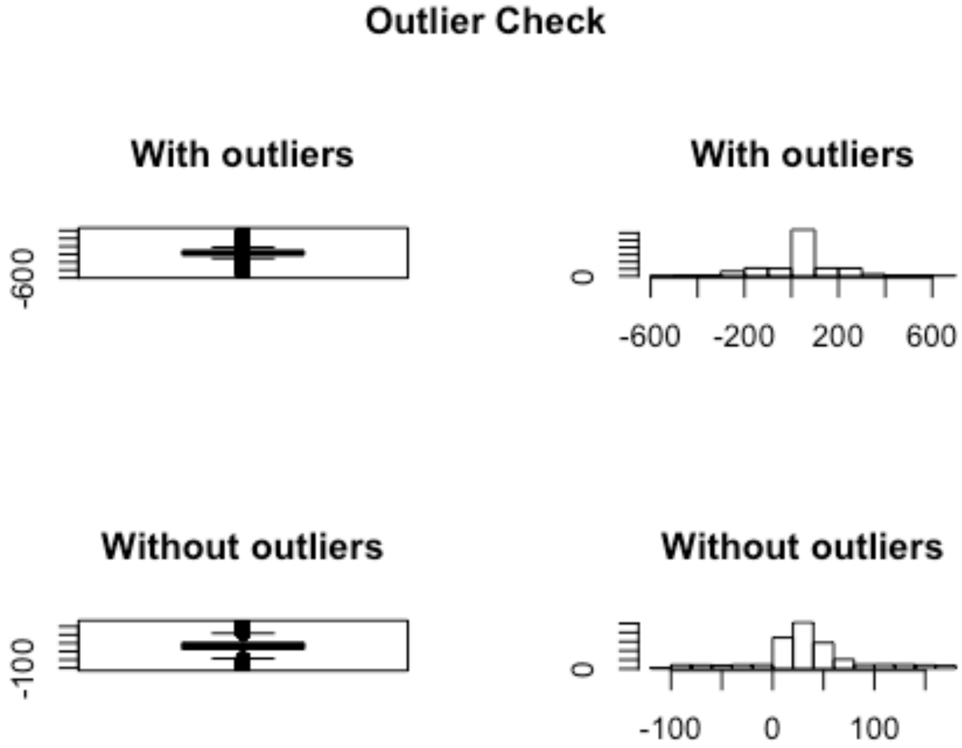
```
## Outliers identified: 928 nPropotion (%) of outliers: 8.3 nMean of
the outliers: 9.47 nMean without removing outliers: 10.49 nMean if we
remove outliers: 10.57 nDo you want to remove outliers and to replace
with NA? [yes/no]:
## Nothing changed n
```

Fixed Acidity



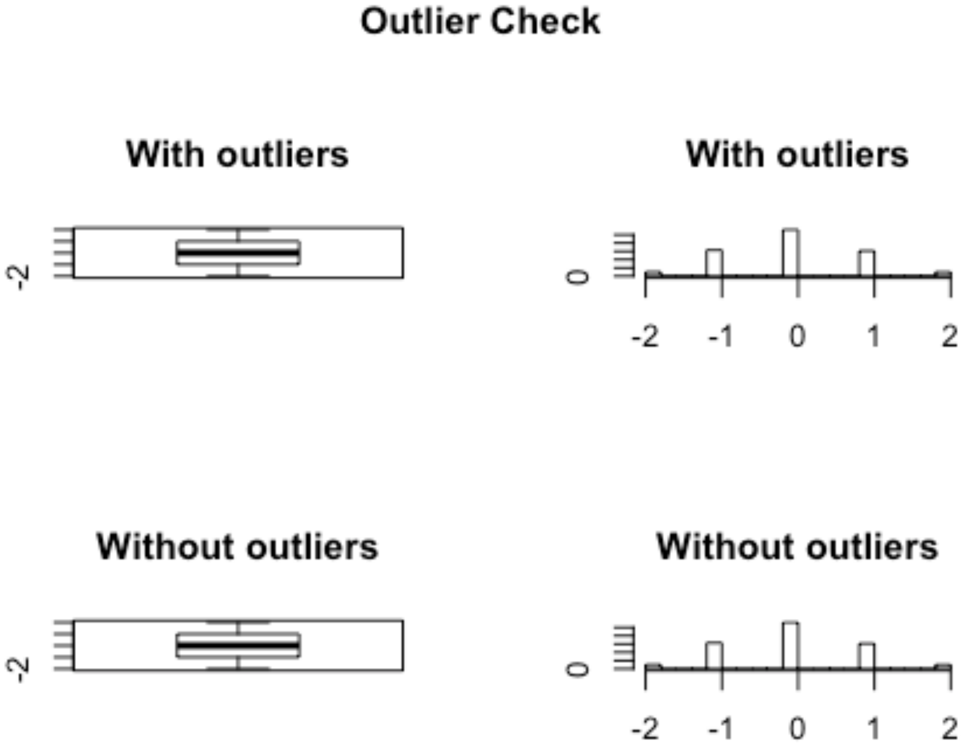
```
## Outliers identified: 2455 nPropotion (%) of outliers: 23.7 nMean of
the outliers: 6.76 nMean without removing outliers: 7.08 nMean if we
remove outliers: 7.15 nDo you want to remove outliers and to replace
with NA? [yes/no]:
## Nothing changed n
```

Free Sulfur Dioxide



```
## Outliers identified: 3712 nPropotion (%) of outliers: 44 nMean of
the outliers: 24.17 nMean without removing outliers: 30.85 nMean if we
remove outliers: 33.78 nDo you want to remove outliers and to replace
with NA? [yes/no]:
## Nothing changed n
```

Label Appeal


Outlier Check

With outliers

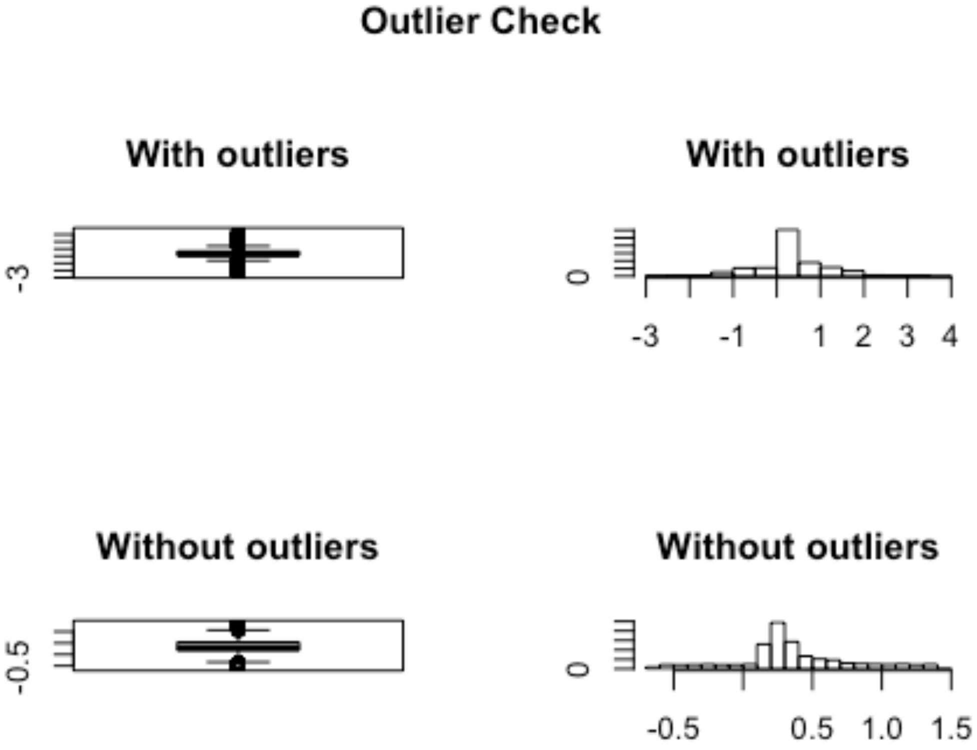
With outliers

Without outliers

Without outliers

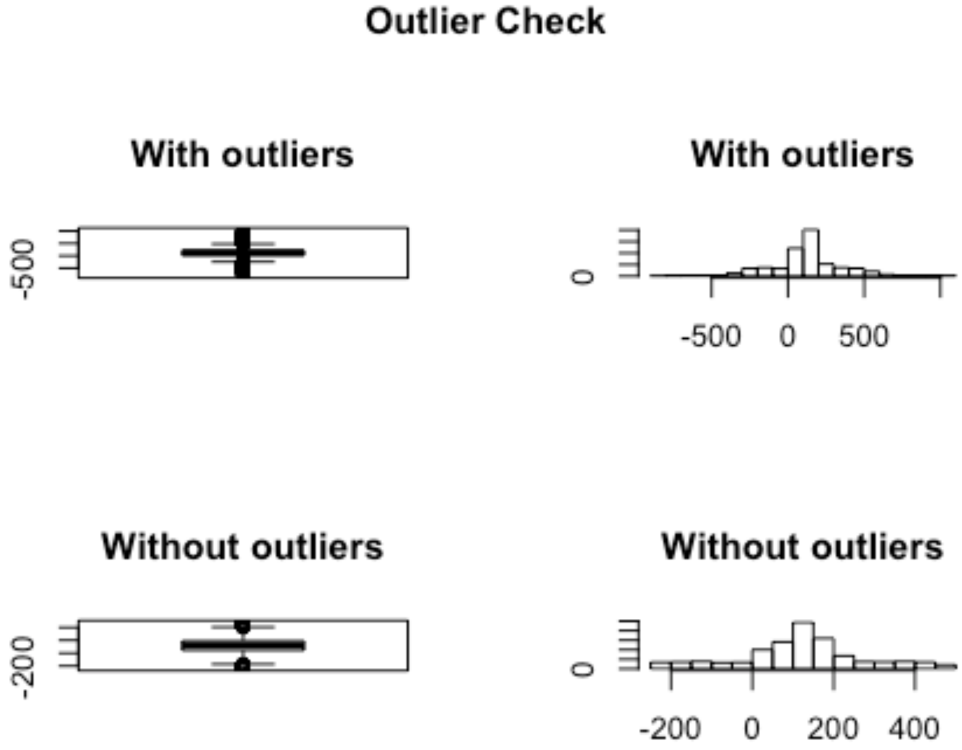
```
## Outliers identified: 0 nPropotion (%) of outliers: 0 nMean of the
outliers: NaN nMean without removing outliers: -0.01 nMean if we remove
outliers: -0.01 nDo you want to remove outliers and to replace with NA?
[yes/no]:
## Nothing changed n
```

Volatile Acidity



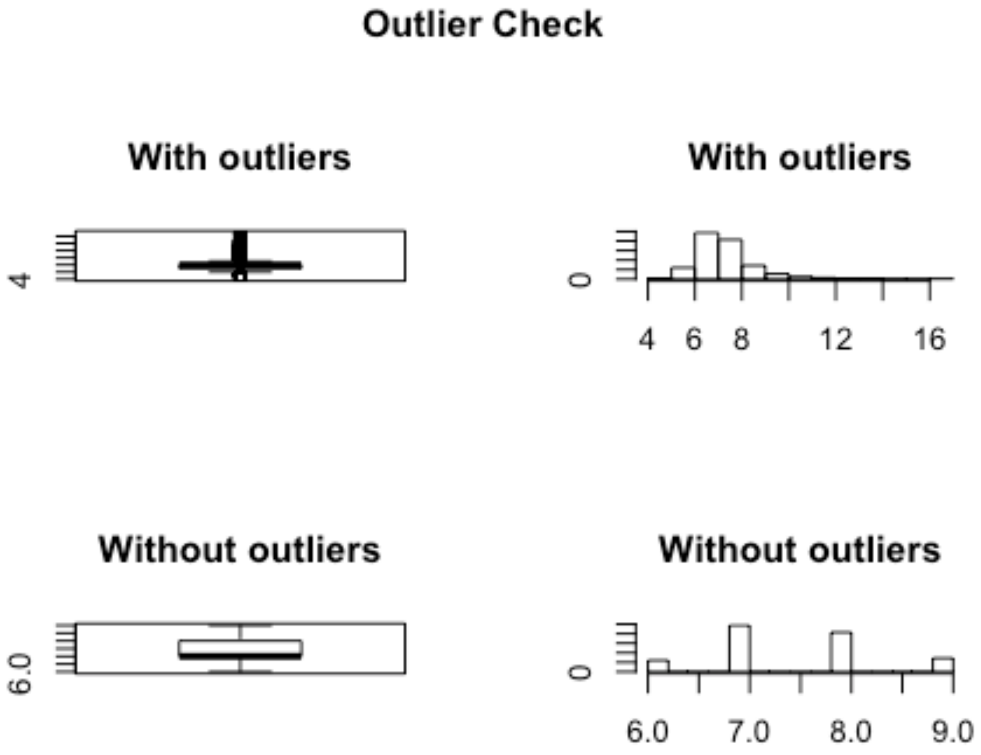
```
## Outliers identified: 2599 nPropotion (%) of outliers: 25.5 nMean of
the outliers: 0.21 nMean without removing outliers: 0.32 nMean if we
remove outliers: 0.35 nDo you want to remove outliers and to replace
with NA? [yes/no]:
## Nothing changed n
```

Total Sulfur Dioxide



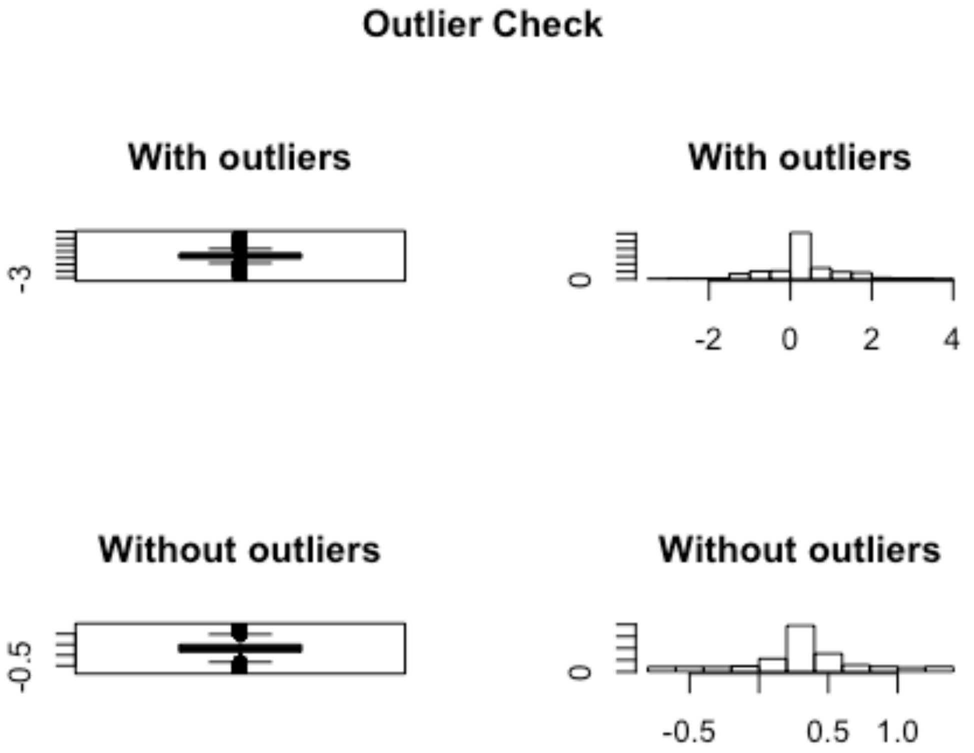
```
## Outliers identified: 1590 nPropotion (%) of outliers: 15.1 nMean of
the outliers: 127.61 nMean without removing outliers: 120.71 nMean if
we remove outliers: 119.67 nDo you want to remove outliers and to
replace with NA? [yes/no]:
## Nothing changed n
```

Acid Index



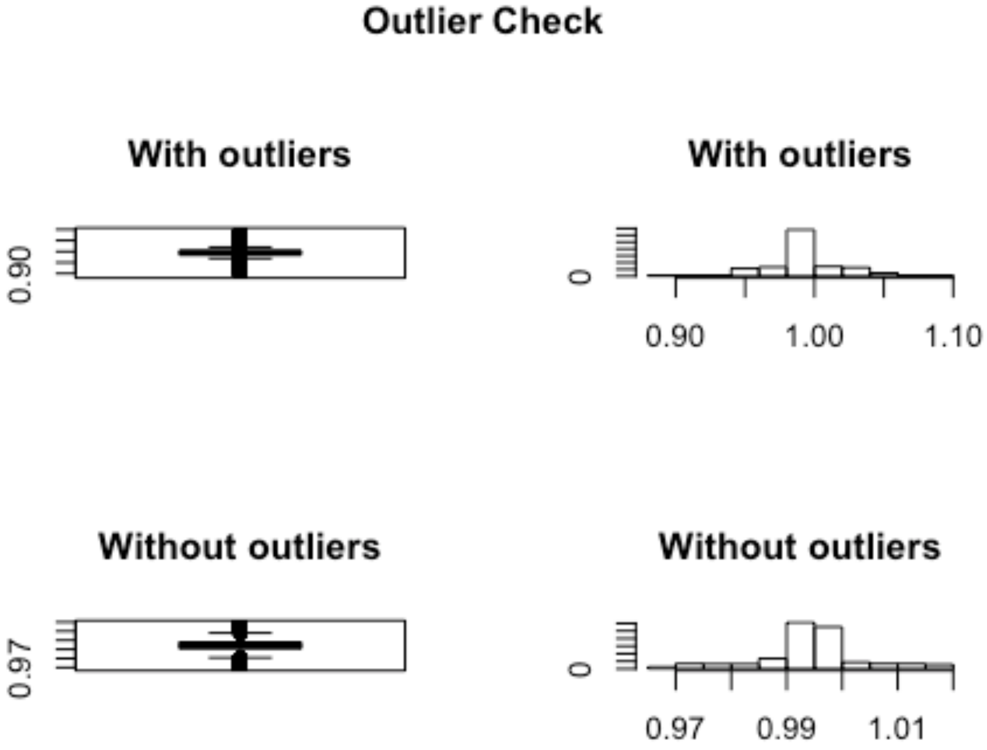
```
## Outliers identified: 1151 nPropotion (%) of outliers: 9.9 nMean of
the outliers: 10.55 nMean without removing outliers: 7.77 nMean if we
remove outliers: 7.5 nDo you want to remove outliers and to replace
with NA? [yes/no]:
## Nothing changed n
```

Citric Acid



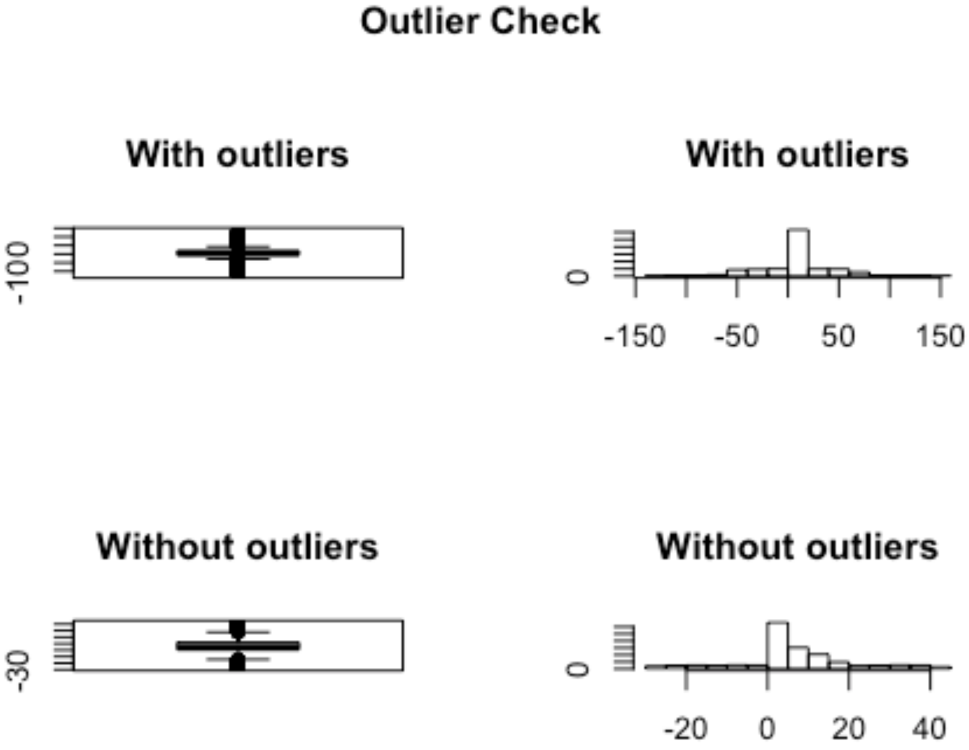
```
## Outliers identified: 2688 nPropotion (%) of outliers: 26.6 nMean of
the outliers: 0.29 nMean without removing outliers: 0.31 nMean if we
remove outliers: 0.31 nDo you want to remove outliers and to replace
with NA? [yes/no]:
## Nothing changed n
```

Denisty



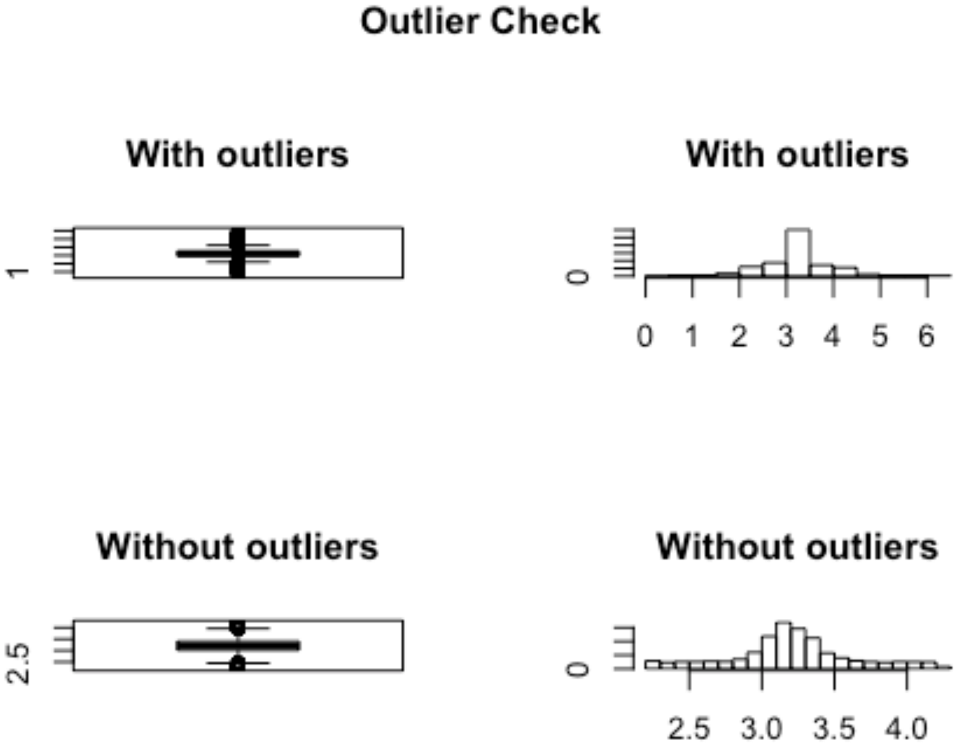
```
## Outliers identified: 3823 nPropotion (%) of outliers: 42.6 nMean of
the outliers: 0.99 nMean without removing outliers: 0.99 nMean if we
remove outliers: 0.99 nDo you want to remove outliers and to replace
with NA? [yes/no]:
## Nothing changed n
```

Residual Sugar



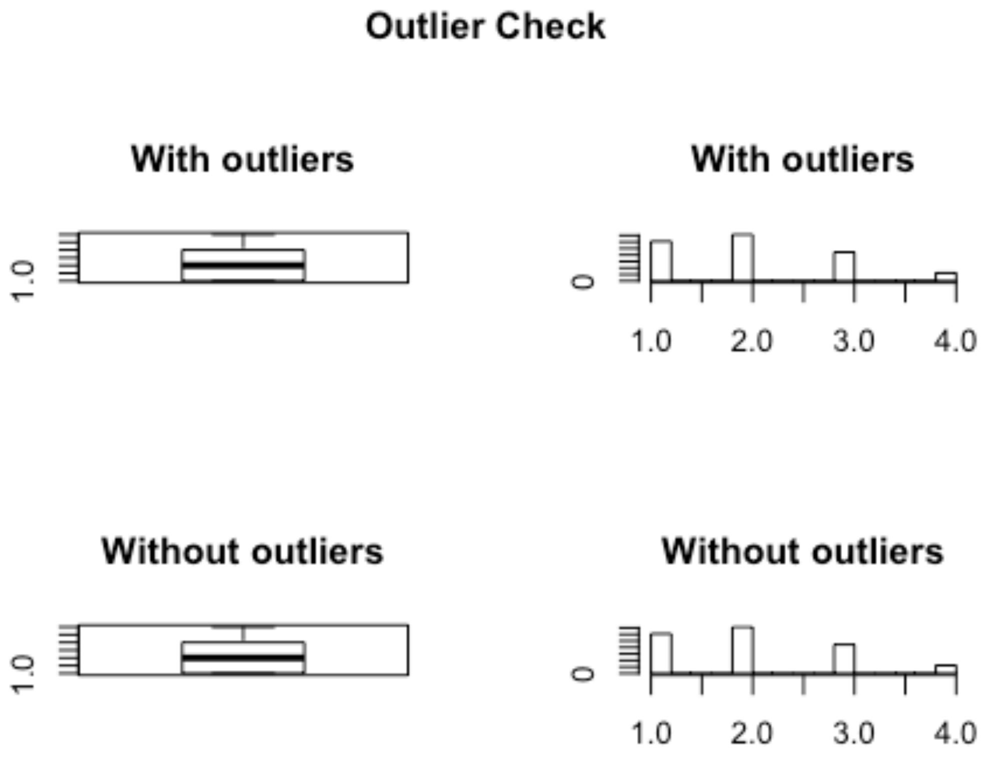
```
## Outliers identified: 3298 nPropotion (%) of outliers: 37.1 nMean of
the outliers: 3.5 nMean without removing outliers: 5.42 nMean if we
remove outliers: 6.13 nDo you want to remove outliers and to replace
with NA? [yes/no]:
## Nothing changed n
```

pH



```
## Outliers identified: 1864 nPropotion (%) of outliers: 17.7 nMean of
the outliers: 3.2 nMean without removing outliers: 3.21 nMean if we
remove outliers: 3.21 nDo you want to remove outliers and to replace
with NA? [yes/no]:
## Nothing changed n
```

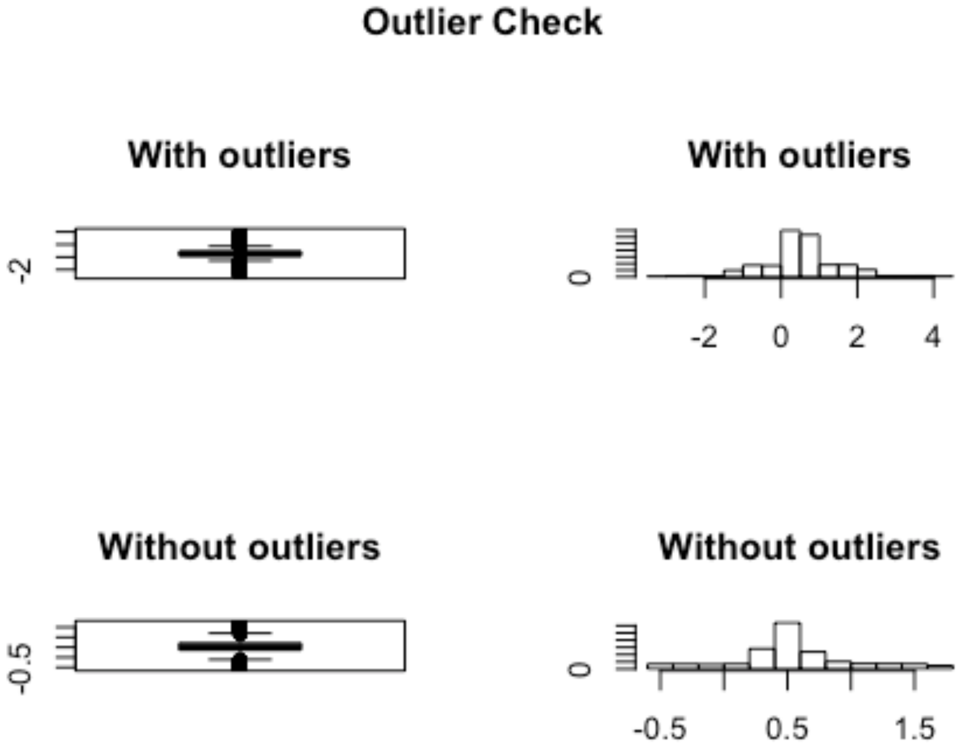
STARS



```
## Outliers identified: 0 nPropotion (%) of outliers: 0 nMean of the
outliers: NaN nMean without removing outliers: 2.04 nMean if we remove
outliers: 2.04 nDo you want to remove outliers and to replace with NA?
[yes/no]:
## Nothing changed n
```

Sulphates

```
outlierKD(wine_training2, Sulphates)
```



```
## Outliers identified: 2606 nPropotion (%) of outliers: 29 nMean of
the outliers: 0.46 nMean without removing outliers: 0.53 nMean if we
remove outliers: 0.55 nDo you want to remove outliers and to replace
with NA? [yes/no]:
## Nothing changed n
```

The removal of outliers seems to tighten the overall spread of certain variables such as pH. Removing outliers for other variables does not really make a change to the overall distribution. Through modeling, we will have a better idea if we should remove outliers or not.

II) Data Preperation Recall some of our findings regarding the data Number of missing values

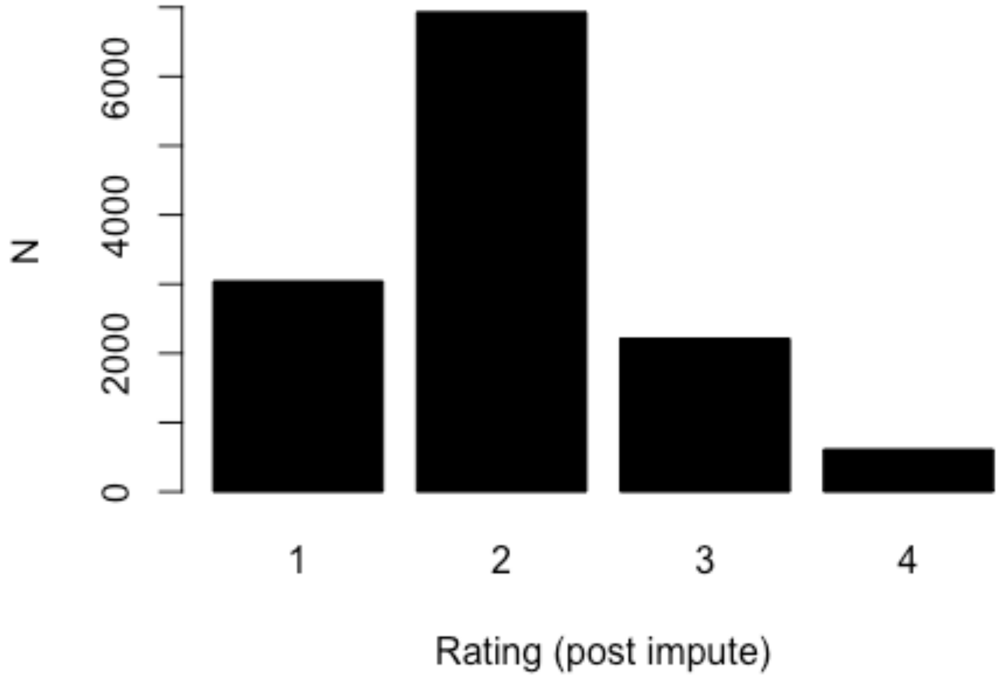
```
##             TARGET       FixedAcidity    VolatileAcidity
##                  0                  0                  0
##         CitricAcid      ResidualSugar          Chlorides
##                  0                616                638
##  FreeSulfurDioxide TotalSulfurDioxide            Density
##                647                682                  0
##                 pH          Sulphates            Alcohol
##                395               1210                653
##        LabelAppeal          AcidIndex              STARS
##                  0                  0               3359
```

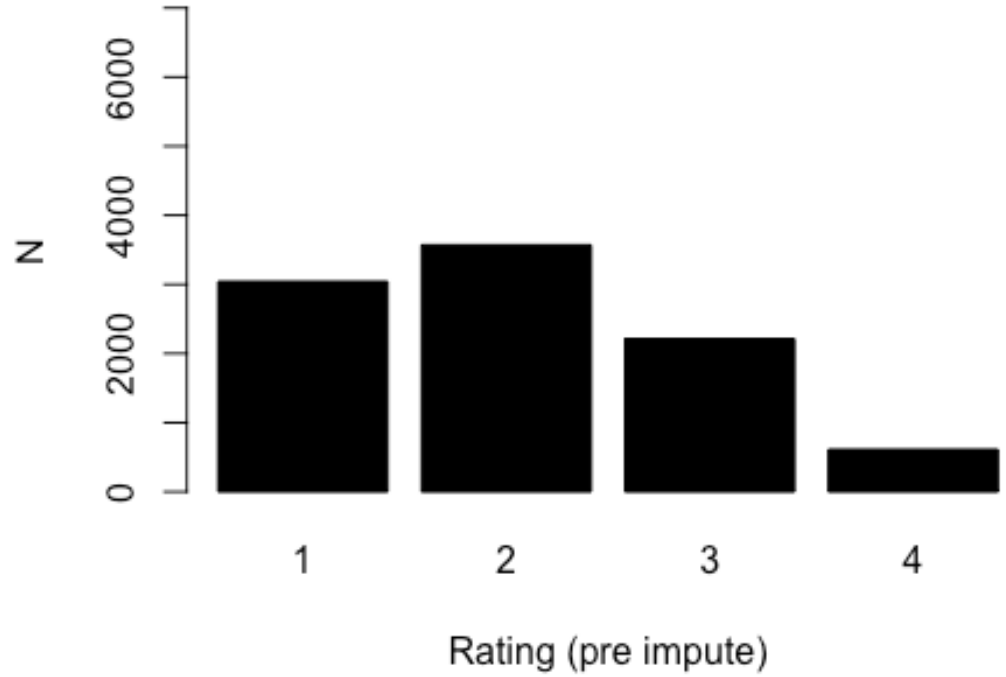
Stars is defined as the rating given by wine experts. One could assume that a high rating should be correlated or related to the response variable. We are going to impute the STARS variable with its median value and recheck the correlation number.

Summary of STARS after we impute missing values with the median vs non imputed values

```
## 
##  3359 values imputed to 2

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   1.000   2.000   2.000   2.031   2.000   4.000

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   1.000   1.000   2.000   2.042   3.000   4.000    3359
```

Plots of Spread





Before we impute the other variables, we need to check if negative values belong in their domin. If negative values do not make sense in the context of the variable, then some potential fixes are to consider the absolute value or drop them from the data frame all together.

```
##             TARGET       FixedAcidity    VolatileAcidity
##                  0               1621               2827
##         CitricAcid      ResidualSugar          Chlorides
##               2966                 NA                 NA
##  FreeSulfurDioxide TotalSulfurDioxide            Density
##                 NA                 NA                  0
##                 pH          Sulphates            Alcohol
##                 NA                 NA                 NA
##        LabelAppeal          AcidIndex              STARS
##               3640                  0                  0

##             TARGET       FixedAcidity    VolatileAcidity
##                  0                  0                  0
##         CitricAcid      ResidualSugar          Chlorides
##                  0                616                638
##  FreeSulfurDioxide TotalSulfurDioxide            Density
##                647                682                  0
##                 pH          Sulphates            Alcohol
##                395               1210                653
##        LabelAppeal          AcidIndex              STARS
##                  0                  0                  0
```

After scanning several documents regarding the chemical properties of wine,is it reasonable to transform the variables with negative values into positive for chemical attributes?

We need to provide justification for transforming chemical attributes into positive only values.

Residual sugar concentration is a measure of the amount of sugar solids in a given volume of wine following the end of fermentation and any sugar addition when making a sweet wine.

Volatile Acidity and citric is also a measurement that can be expressed in g/l or mg/l.

Fixed Acidity levels found in wine can vary greatly but in general one would expect to see 1,000 to 4,000 mg/L tartaric acid, 0 to 8,000 mg/L malic acid, 0 to 500 mg/L citric acid, and 500 to 2,000 mg/L succinic acid

Based on some understanding on these chemical attributes, it makes sense to treat negative values. Any acidity variable is a measure and measures cannot be negative unless we examine a rate of change. However rate of change is not needed for this case study.

Documentation: https://winemakermag.com/501-measuring-residual-sugar-techniques http://waterhouse.ucdavis.edu/whats-in-wine/volatile-acidity https://en.wikipedia.org/wiki/Acids_in_wine

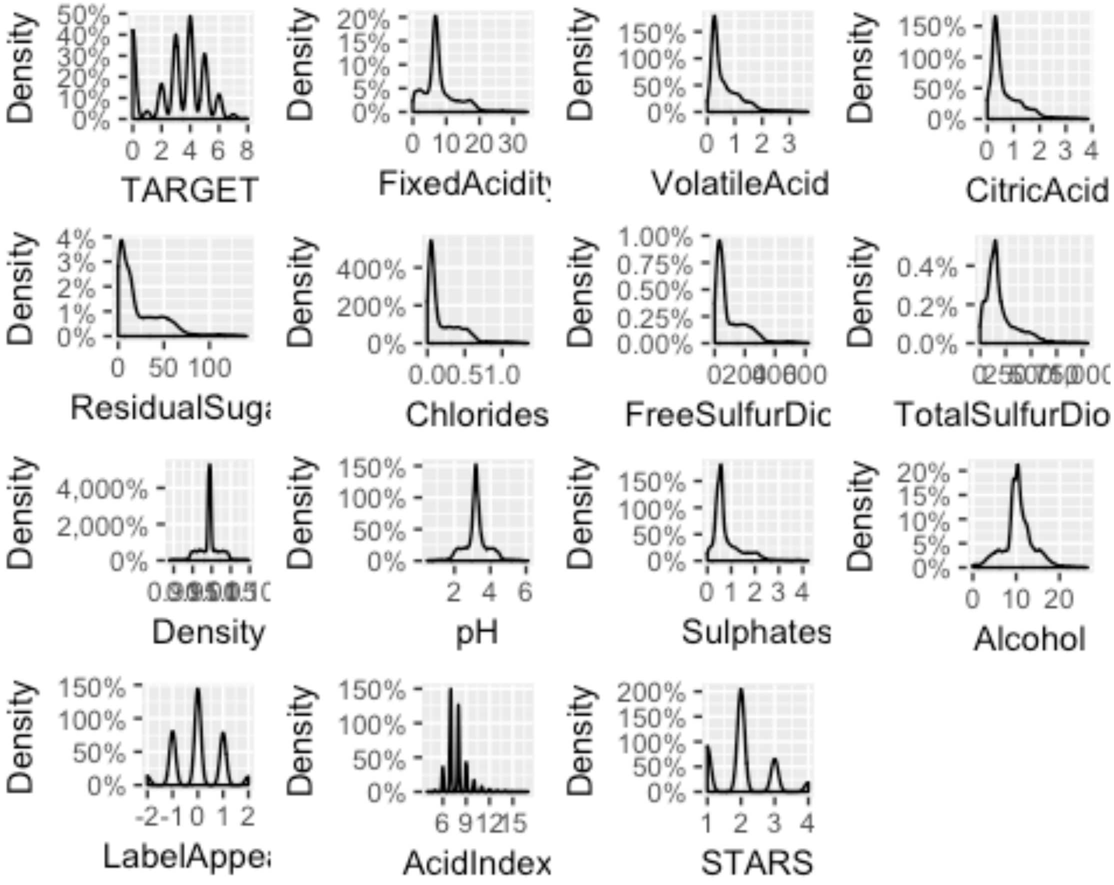
If positve plus imputed values has a deterimental effect on the model, then we will use the unaltered data wine_training2 and consider some alternate transformation.

```
## 
##  616 values imputed to 12.9
## 
## 
##  638 values imputed to 0.098
## 
## 
##  647 values imputed to 56
## 
## 
##  682 values imputed to 154
## 
## 
##  395 values imputed to 3.2
## 
## 
```

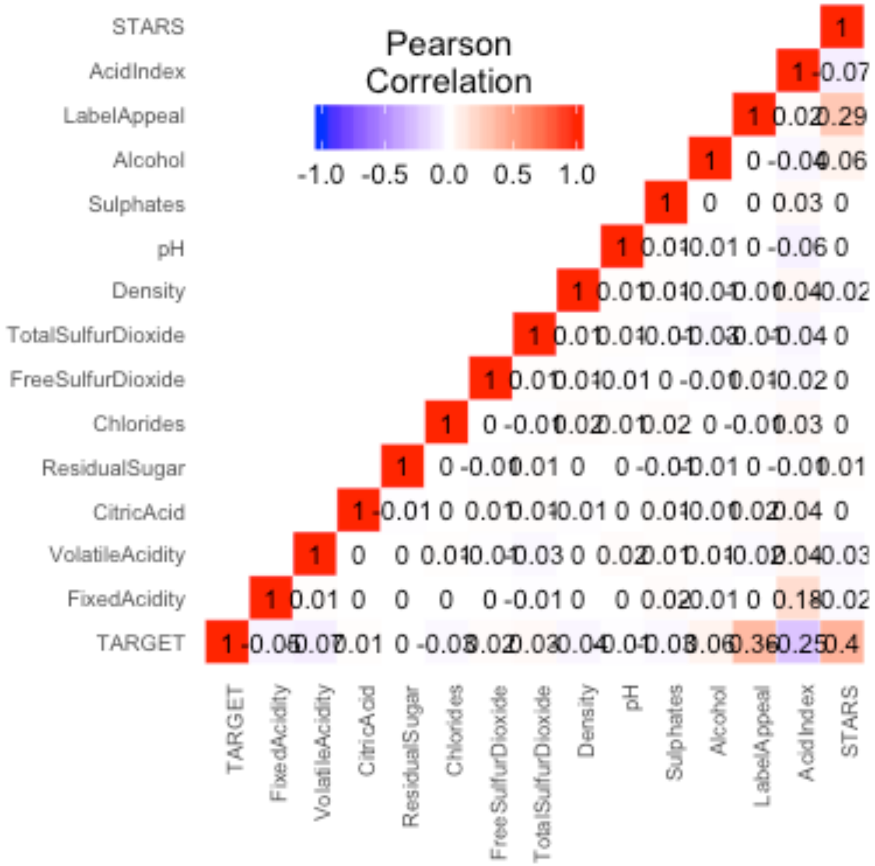
```
##  1210 values imputed to 0.59
##
##
##  653 values imputed to 10.4
##
##
##  3359 values imputed to 2
```

```
##      TARGET       FixedAcidity    VolatileAcidity    CitricAcid
##  Min.   :0.000   Min.   : 0.000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:2.000   1st Qu.: 5.600   1st Qu.:0.2500   1st Qu.:0.2800
##  Median :3.000   Median : 7.000   Median :0.4100   Median :0.4400
##  Mean   :3.029   Mean   : 8.063   Mean   :0.6411   Mean   :0.6863
##  3rd Qu.:4.000   3rd Qu.: 9.800   3rd Qu.:0.9100   3rd Qu.:0.9700
##  Max.   :8.000   Max.   :34.400   Max.   :3.6800   Max.   :3.8600
##  ResidualSugar      Chlorides      FreeSulfurDioxide
TotalSulfurDioxide
##  Min.   :  0.00   Min.   :0.0000   Min.   :  0.0     Min.   :   0.0
##  1st Qu.:  4.00   1st Qu.:0.0460   1st Qu.: 29.0     1st Qu.: 102.0
##  Median : 12.90   Median :0.0980   Median : 56.0     Median : 154.0
##  Mean   : 22.86   Mean   :0.2163   Mean   :104.1     Mean   : 201.6
##  3rd Qu.: 37.20   3rd Qu.:0.3530   3rd Qu.:164.0     3rd Qu.: 251.0
##  Max.   :141.15   Max.   :1.3510   Max.   :623.0     Max.   :1057.0
##     Density             pH          Sulphates        Alcohol
##  Min.   :0.8881   Min.   :0.480   Min.   :0.0000   Min.   : 0.00
##  1st Qu.:0.9877   1st Qu.:2.970   1st Qu.:0.4500   1st Qu.: 9.10
##  Median :0.9945   Median :3.200   Median :0.5900   Median :10.40
##  Mean   :0.9942   Mean   :3.207   Mean   :0.8224   Mean   :10.52
##  3rd Qu.:1.0005   3rd Qu.:3.450   3rd Qu.:1.0000   3rd Qu.:12.20
##  Max.   :1.0992   Max.   :6.130   Max.   :4.2400   Max.   :26.50
##   LabelAppeal          AcidIndex          STARS
##  Min.   :-2.000000   Min.   : 4.000   Min.   :1.000
##  1st Qu.:-1.000000   1st Qu.: 7.000   1st Qu.:2.000
##  Median : 0.000000   Median : 8.000   Median :2.000
##  Mean   :-0.009066   Mean   : 7.773   Mean   :2.031
##  3rd Qu.: 1.000000   3rd Qu.: 8.000   3rd Qu.:2.000
##  Max.   : 2.000000   Max.   :17.000   Max.   :4.000
```

Howdid the distributions change?



How Does correlation change? (wine_training3)



We start to see evidence of more relationships within the data after we have transformed the data. Once we do modeling, we will be able to determine if the data has strong integrity. To reiterate, the entires in the variables relating to chemical properties were made positive because we discovered that they pertain to measurments in units such as mg and l. It does not make sense for measurments to be negative in the physical world. I assume the negative values were a result of a data collection error.

III) Build Models

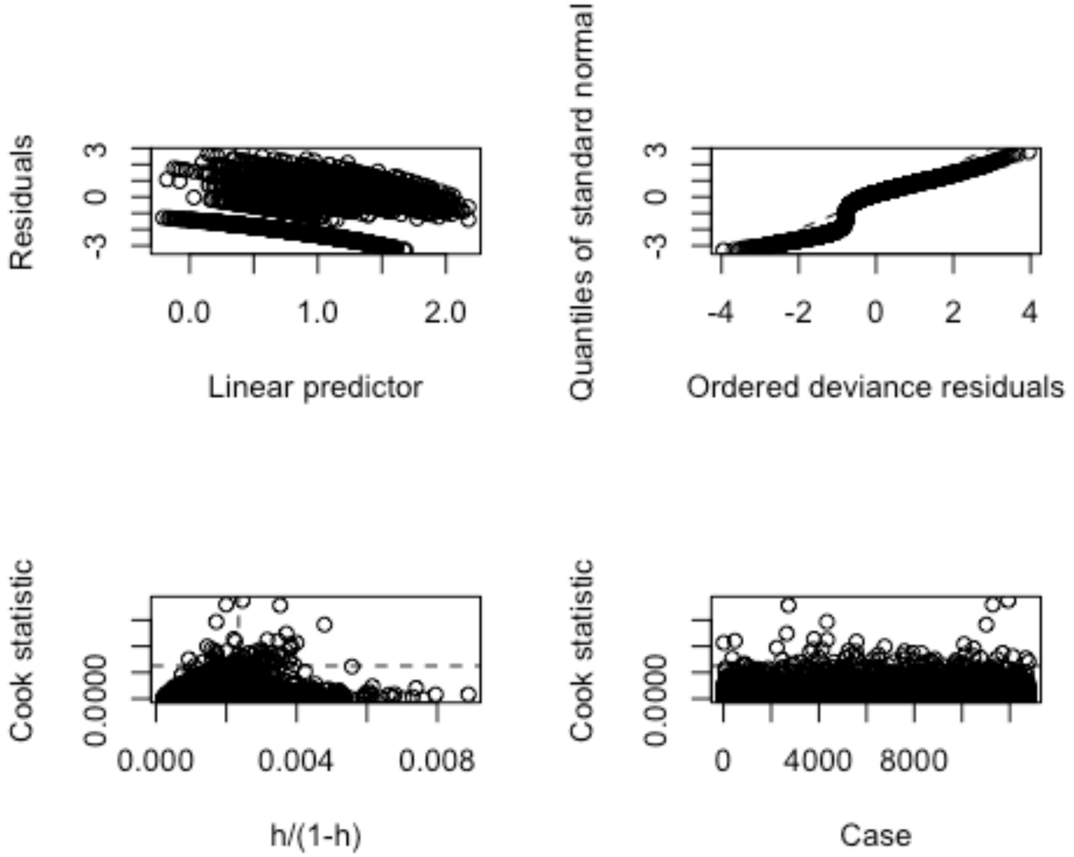
Objective: Using the training data set, build at least two different poisson regression models, at least two different negative binomial regression models, and at least two multiple linear regression models, using different variables (or the same variables with different transformations)

Full Poisson Regression Model on Transformed Data

```
## Loading required package: grid

##
## Attaching package: 'faraway'
## The following object is masked from 'package:survival':
##
##     rats

## The following object is masked from 'package:lattice':
##
##     melanoma

## Loading required package: car

## Warning: package 'car' was built under R version 3.4.4

## Loading required package: carData

## Warning: package 'carData' was built under R version 3.4.4

##
## Attaching package: 'car'

## The following objects are masked from 'package:faraway':
##
##     logit, vif

## Loading required package: lmtest

## Warning: package 'lmtest' was built under R version 3.4.4

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

##
## Attaching package: 'boot'

## The following object is masked from 'package:car':
##
##     logit

## The following objects are masked from 'package:faraway':
##
##     logit, melanoma

## The following object is masked from 'package:survival':
##
##     aml
## The following object is masked from 'package:lattice':
##
##     melanoma
```

```
##
## Call:
## glm(formula = TARGET ~ ., family = "poisson", data = wine_training3)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2792  -0.5084   0.1987   0.6366   2.7592
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         2.053e+00  1.962e-01  10.461  < 2e-16 ***
## FixedAcidity       -7.866e-04  1.046e-03  -0.752 0.452212
## VolatileAcidity    -5.881e-02  9.416e-03  -6.246 4.21e-10 ***
## CitricAcid          1.734e-02  8.292e-03   2.091 0.036540 *
## ResidualSugar       1.821e-05  2.078e-04   0.088 0.930182
## Chlorides          -4.456e-02  2.230e-02  -1.998 0.045703 *
## FreeSulfurDioxide   9.112e-05  4.782e-05   1.906 0.056709 .
## TotalSulfurDioxide  1.167e-04  3.165e-05   3.688 0.000226 ***
## Density            -4.520e-01  1.922e-01  -2.352 0.018658 *
## pH                 -2.349e-02  7.635e-03  -3.077 0.002093 **
## Sulphates          -2.297e-02  8.229e-03  -2.791 0.005247 **
## Alcohol             5.905e-03  1.445e-03   4.087 4.37e-05 ***
## LabelAppeal         1.963e-01  6.020e-03  32.606  < 2e-16 ***
## AcidIndex          -1.233e-01  4.453e-03 -27.681  < 2e-16 ***
## STARS               2.211e-01  6.466e-03  34.202  < 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: 18475  on 12780  degrees of freedom
## AIC: 50447
##
## Number of Fisher Scoring iterations: 5
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: TARGET
##
## Terms added sequentially (first to last)
##
##
##                    Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                 12794      22861
## FixedAcidity        1    44.59      12793      22816 2.428e-11 ***
## VolatileAcidity     1    77.89      12792      22738 < 2.2e-16 ***
## CitricAcid          1     2.84      12791      22736 0.0916712 .
## ResidualSugar       1     0.15      12790      22735 0.6973282
## Chlorides           1    11.55      12789      22724 0.0006771 ***
## FreeSulfurDioxide   1     8.03      12788      22716 0.0045888 **
## TotalSulfurDioxide  1    13.89      12787      22702 0.0001938 ***
## Density             1    20.14      12786      22682 7.199e-06 ***
## pH                  1     1.03      12785      22681 0.3099732
## Sulphates           1    13.56      12784      22667 0.0002316 ***
## Alcohol             1    62.20      12783      22605 3.110e-15 ***
## LabelAppeal         1  1975.12      12782      20630 < 2.2e-16 ***
## AcidIndex           1  1006.22      12781      19624 < 2.2e-16 ***
## STARS               1  1148.95      12780      18475 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



```
##      res.deviance    df             p
## [1,]     18474.73 12780 1.430684e-216

## [1] 0.1918551

## [1] 1.430684e-216
```

Our first model tells us that unit decrases in attributes such as volatile acidity cause the expected number of sales to increase. Residual sugars and fixed acidity have high p values. Residal Sugars have almost no correlation with any of the variables even after transformation. Based on the p-value from the chi square goodness of fit test, we are unable to conclude that the full model is a good fit. The G-statistic is our substitue for r square and in this case, it comes out to .19.

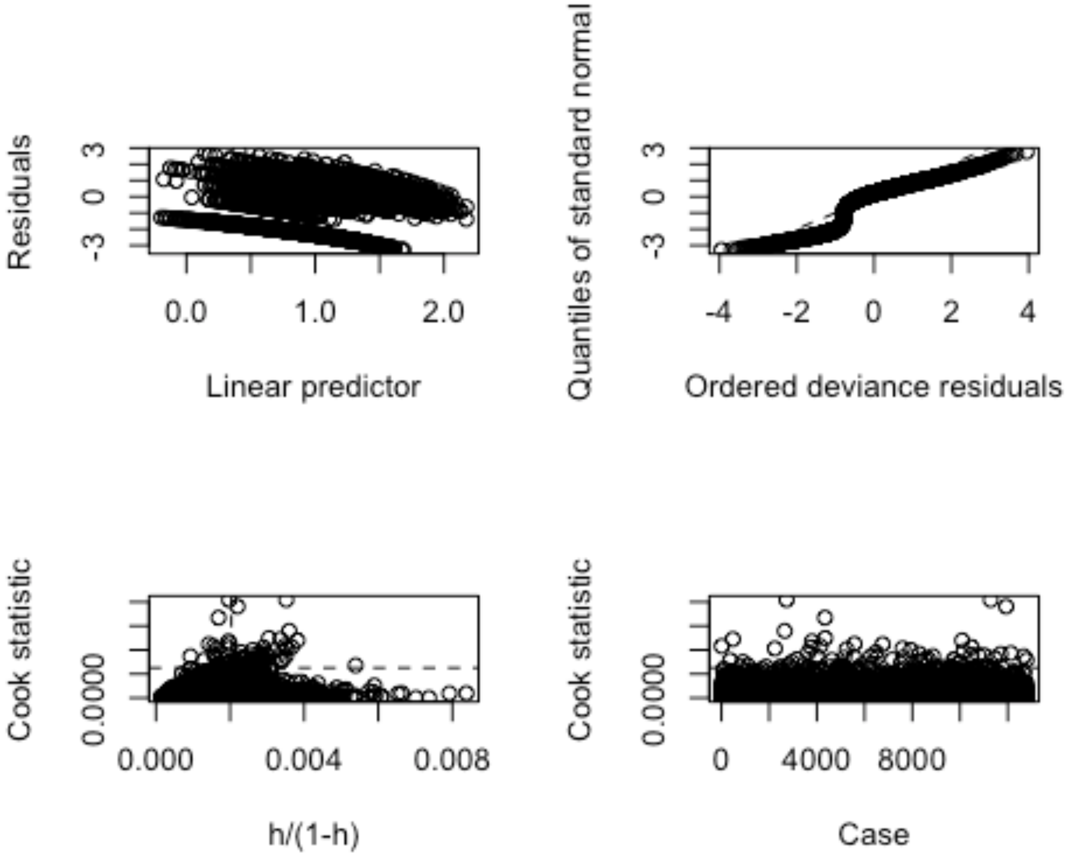
Lets consider some variable selection using Aic

```
## Stepwise Model Path 
## Analysis of Deviance Table
## 
## Initial Model:
## TARGET ~ FixedAcidity + VolatileAcidity + CitricAcid + ResidualSugar 
+ 
##     Chlorides + FreeSulfurDioxide + TotalSulfurDioxide + Density + 
##     pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS
## 
## Final Model:
## TARGET ~ VolatileAcidity + CitricAcid + Chlorides + 
FreeSulfurDioxide + 
##     TotalSulfurDioxide + Density + pH + Sulphates + Alcohol + 
##     LabelAppeal + AcidIndex + STARS
## 
## 
##              Step Df   Deviance Resid. Df Resid. Dev      AIC
## 1                                   12780   18474.73 50446.75
## 2 - ResidualSugar  1 0.00767479     12781   18474.74 50444.76
## 3  - FixedAcidity  1 0.56501594     12782   18475.31 50443.33
```

AIC suggested a model with Residual Sugar and Fixed Acidity removed. Lets formulate the generated model.

```
## 
## Call:
## glm(formula = TARGET ~ VolatileAcidity + CitricAcid + Chlorides + 
##     FreeSulfurDioxide + TotalSulfurDioxide + Density + pH + 
Sulphates + 
##     Alcohol + LabelAppeal + AcidIndex + STARS, family = "poisson", 
##     data = wine_training3)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -3.2775  -0.5077   0.1990   0.6363   2.7576  
## 
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)    
## (Intercept)         2.050e+00  1.961e-01  10.453  < 2e-16 ***
## VolatileAcidity    -5.887e-02  9.416e-03  -6.252 4.06e-10 ***
## CitricAcid          1.742e-02  8.290e-03   2.101 0.035644 *  
## Chlorides          -4.451e-02  2.230e-02  -1.996 0.045905 *  
## FreeSulfurDioxide   9.127e-05  4.782e-05   1.909 0.056285 .  
## TotalSulfurDioxide  1.167e-04  3.164e-05   3.689 0.000225 ***
## Density            -4.509e-01  1.922e-01  -2.347 0.018942 *  
## pH                 -2.356e-02  7.635e-03  -3.086 0.002028 ** 
## Sulphates          -2.303e-02  8.228e-03  -2.799 0.005118 ** 
## Alcohol             5.908e-03  1.445e-03   4.089 4.33e-05 ***
## LabelAppeal         1.963e-01  6.020e-03  32.612  < 2e-16 ***
## AcidIndex          -1.238e-01  4.400e-03 -28.135  < 2e-16 ***
## STARS               2.211e-01  6.466e-03  34.203  < 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: 18475  on 12782  degrees of freedom
## AIC: 50443
## 
## Number of Fisher Scoring iterations: 5
```

```
## Analysis of Deviance Table
## 
## Model: poisson, link: log
## 
## Response: TARGET
## 
## Terms added sequentially (first to last)
## 
## 
##                    Df Deviance Resid. Df Resid. Dev  Pr(>Chi)    
## NULL                               12794      22861              
## VolatileAcidity     1    79.20     12793      22782 < 2.2e-16 ***
## CitricAcid          1     2.95     12792      22779 0.0858538 .  
## Chlorides           1    11.59     12791      22767 0.0006616 ***
## FreeSulfurDioxide   1     8.15     12790      22759 0.0043023 ** 
## TotalSulfurDioxide  1    14.46     12789      22744 0.0001430 ***
## Density             1    20.14     12788      22724 7.185e-06 ***
## pH                  1     1.00     12787      22723 0.3183278    
## Sulphates           1    14.43     12786      22709 0.0001455 ***
## Alcohol             1    63.20     12785      22646 1.869e-15 ***
## LabelAppeal         1  1976.00     12784      20670 < 2.2e-16 ***
## AcidIndex           1  1045.43     12783      19624 < 2.2e-16 ***
## STARS               1  1149.04     12782      18475 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

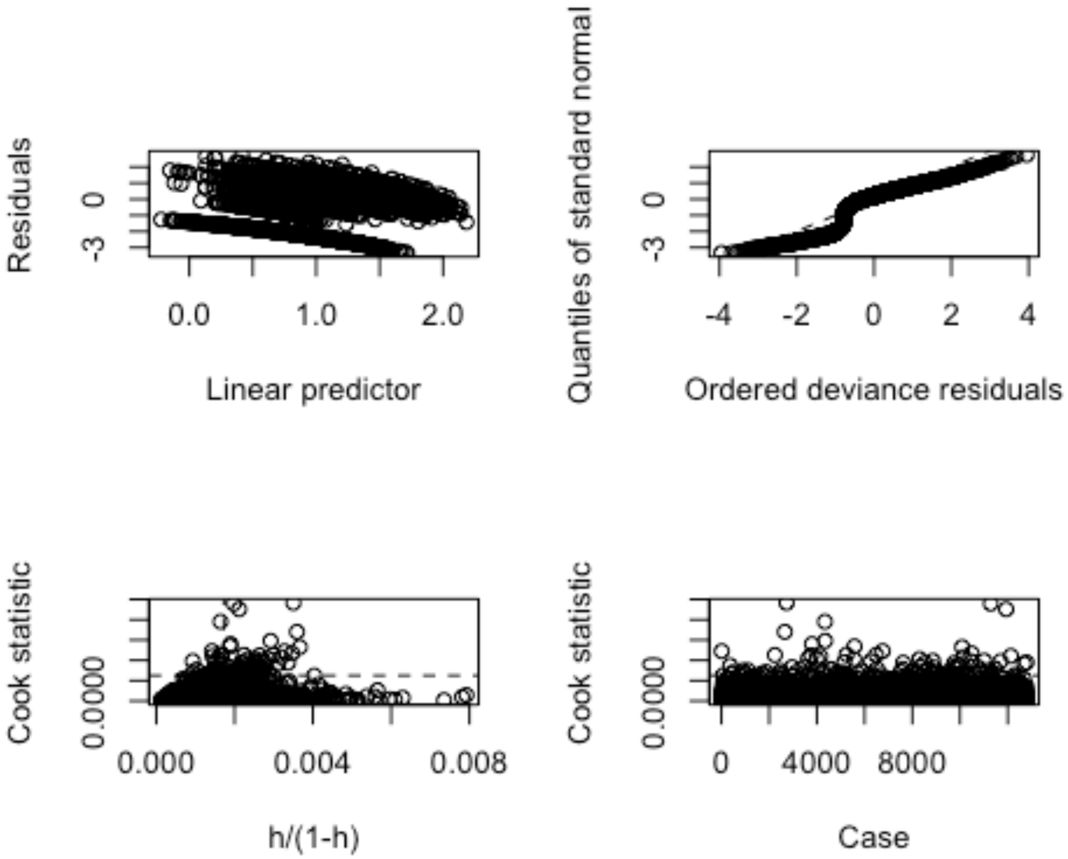


```
##      res.deviance    df             p
## [1,]     18475.31 12782 1.893996e-216

## [1] 0.1918551

## [1] 1.893996e-216
```

According to this second iteration, volitile acidity and pH are features that do not show evidence of a gooffit according to chi square test. We can build a third model with these additional variables removed.

```
## 
## Call:
## glm(formula = TARGET ~ CitricAcid + Chlorides + FreeSulfurDioxide + 
##     TotalSulfurDioxide + Density + Sulphates + Alcohol + LabelAppeal 
+ 
##     AcidIndex + STARS, family = "poisson", data = wine_training3)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -3.3320  -0.4995   0.2007   0.6325   2.7482  
## 
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         1.926e+00  1.942e-01   9.915  < 2e-16 ***
## CitricAcid          1.766e-02  8.284e-03   2.132 0.032993 *
## Chlorides          -4.576e-02  2.230e-02  -2.052 0.040160 *
## FreeSulfurDioxide   9.448e-05  4.780e-05   1.977 0.048080 *
## TotalSulfurDioxide  1.227e-04  3.161e-05   3.882 0.000103 ***
## Density            -4.434e-01  1.920e-01  -2.309 0.020946 *
## Sulphates          -2.370e-02  8.232e-03  -2.879 0.003992 **
## Alcohol             5.783e-03  1.444e-03   4.004 6.24e-05 ***
## LabelAppeal         1.964e-01  6.017e-03  32.645  < 2e-16 ***
## AcidIndex          -1.235e-01  4.384e-03 -28.164  < 2e-16 ***
## STARS               2.222e-01  6.462e-03  34.393  < 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: 18525  on 12784  degrees of freedom
## AIC: 50489
##
## Number of Fisher Scoring iterations: 5
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: TARGET
##
## Terms added sequentially (first to last)
##
##
##                    Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                               12794      22861
## CitricAcid          1     3.04     12793      22858 0.0813918 .
## Chlorides           1    12.19     12792      22846 0.0004797 ***
## FreeSulfurDioxide   1     8.63     12791      22837 0.0033041 **
## TotalSulfurDioxide  1    16.85     12790      22820 4.036e-05 ***
## Density             1    19.79     12789      22800 8.642e-06 ***
## Sulphates           1    14.96     12788      22785 0.0001098 ***
## Alcohol             1    61.60     12787      22724 4.213e-15 ***
## LabelAppeal         1  1986.11     12786      20738 < 2.2e-16 ***
## AcidIndex           1  1050.67     12785      19687 < 2.2e-16 ***
## STARS               1  1161.79     12784      18525 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



```
##      res.deviance    df            p
## [1,]     18525.26 12784 1.20718e-219

## [1] 0.1918551

## [1] 1.20718e-219
```

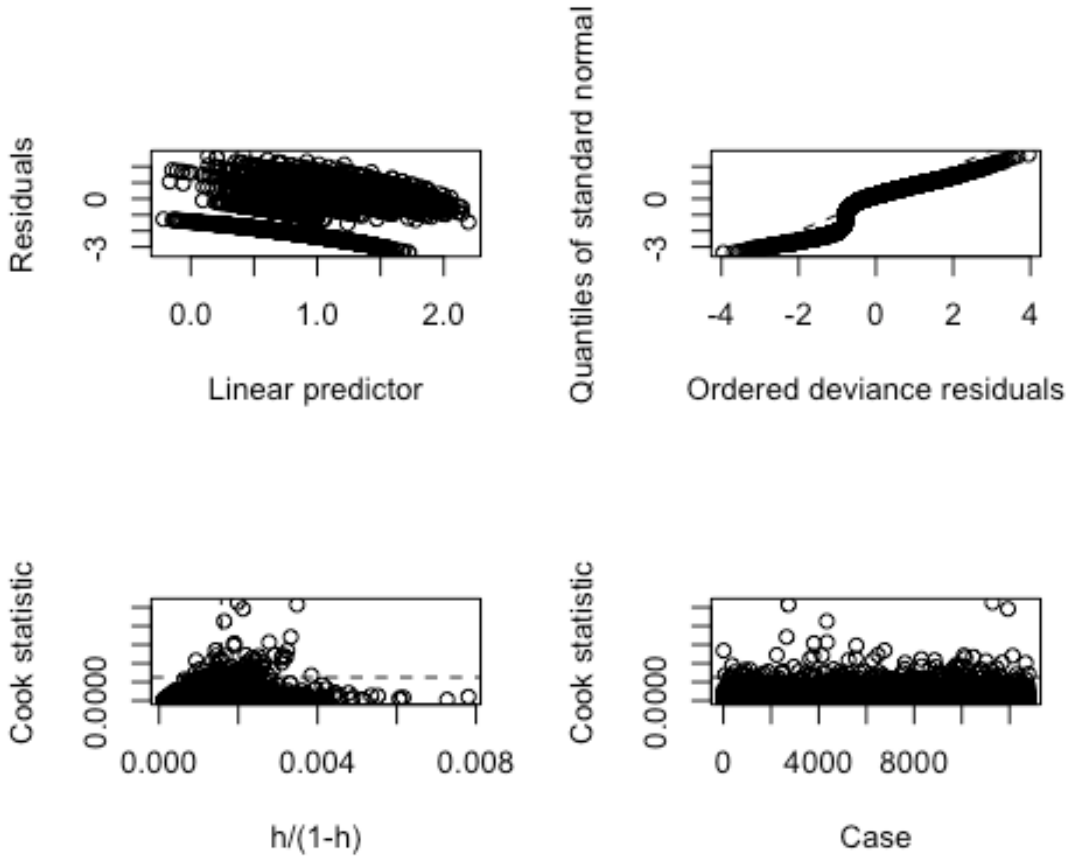
Citric Acid does not appear to have any evidence of being a good fit for the model. Lets build an additional model with citric acid removed.

We will also take the diagnostic plots one step furthur. We want to estimate the variance for the target given the mean.The variance appears to be much smaller than the mean

```
##
## Call:
## glm(formula = TARGET ~ Chlorides + FreeSulfurDioxide +
TotalSulfurDioxide +
##     Density + Sulphates + Alcohol + LabelAppeal + AcidIndex +
##     STARS, family = "poisson", data = wine_training3)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3519  -0.4936   0.2012   0.6351   2.7432
```

```
## 
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)    
## (Intercept)         1.938e+00  1.941e-01   9.982  < 2e-16 ***
## Chlorides          -4.603e-02  2.230e-02  -2.064   0.0390 *  
## FreeSulfurDioxide   9.474e-05  4.780e-05   1.982   0.0475 *  
## TotalSulfurDioxide  1.234e-04  3.161e-05   3.902 9.52e-05 ***
## Density            -4.457e-01  1.920e-01  -2.322   0.0203 *  
## Sulphates          -2.344e-02  8.231e-03  -2.848   0.0044 ** 
## Alcohol             5.768e-03  1.445e-03   3.993 6.53e-05 ***
## LabelAppeal         1.966e-01  6.017e-03  32.677  < 2e-16 ***
## AcidIndex          -1.232e-01  4.381e-03 -28.111  < 2e-16 ***
## STARS               2.223e-01  6.462e-03  34.394  < 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: 18530  on 12785  degrees of freedom
## AIC: 50492
## 
## Number of Fisher Scoring iterations: 5
```

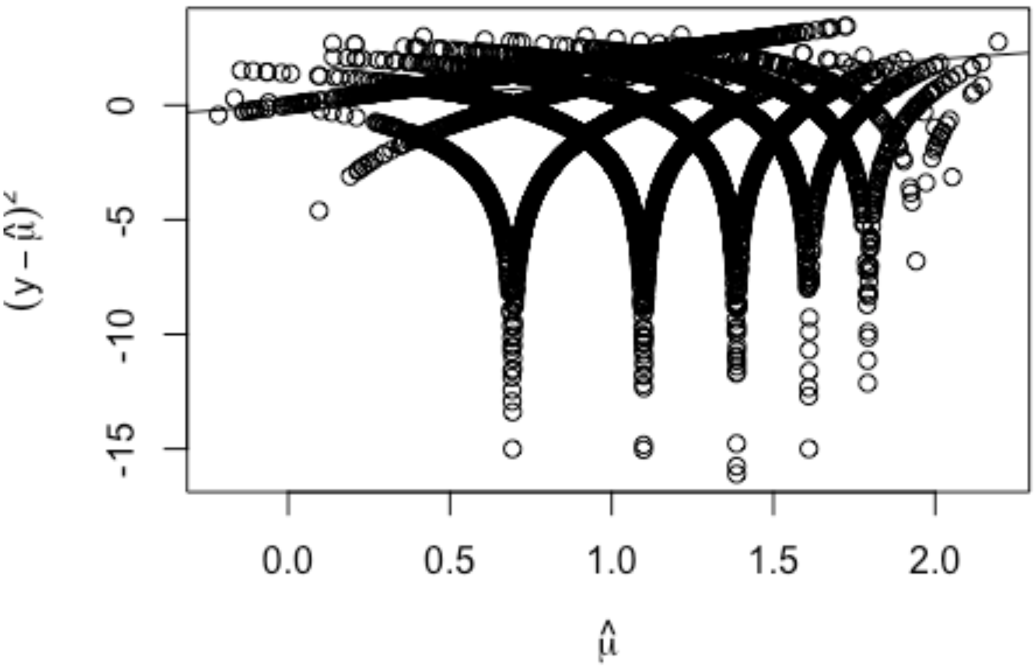
```
## Analysis of Deviance Table
## 
## Model: poisson, link: log
## 
## Response: TARGET
## 
## Terms added sequentially (first to last)
## 
## 
##                    Df Deviance Resid. Df Resid. Dev  Pr(>Chi)    
## NULL                               12794      22861              
## Chlorides           1    12.25     12793      22849 0.0004649 ***
## FreeSulfurDioxide   1     8.69     12792      22840 0.0032076 ** 
## TotalSulfurDioxide  1    16.98     12791      22823 3.783e-05 ***
## Density             1    19.95     12790      22803 7.969e-06 ***
## Sulphates           1    14.77     12789      22788 0.0001217 ***
## Alcohol             1    61.45     12788      22727 4.532e-15 ***
## LabelAppeal         1  1988.15     12787      20739 < 2.2e-16 ***
## AcidIndex           1  1046.99     12786      19692 < 2.2e-16 ***
## STARS               1  1161.90     12785      18530 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



```
##      res.deviance    df             p
## [1,]     18529.78 12785 7.209302e-220
```

```
##                     Odds ratio     2.5 %     97.5 %
## (Intercept)          6.9420251 4.7453397 10.1555874
## Chlorides            0.9550140 0.9141708  0.9976819
## FreeSulfurDioxide    1.0000947 1.0000011  1.0001884
## TotalSulfurDioxide   1.0001234 1.0000614  1.0001853
## Density              0.6403568 0.4395335  0.9329366
## Sulphates            0.9768312 0.9611996  0.9927171
## Alcohol              1.0057845 1.0029409  1.0086362
## LabelAppeal          1.2172588 1.2029888  1.2316981
## AcidIndex            0.8841256 0.8765664  0.8917499
## STARS                1.2488944 1.2331763  1.2648128
```

```
## [1] 0.1918551
```



```
## [1] 7.209302e-220
```

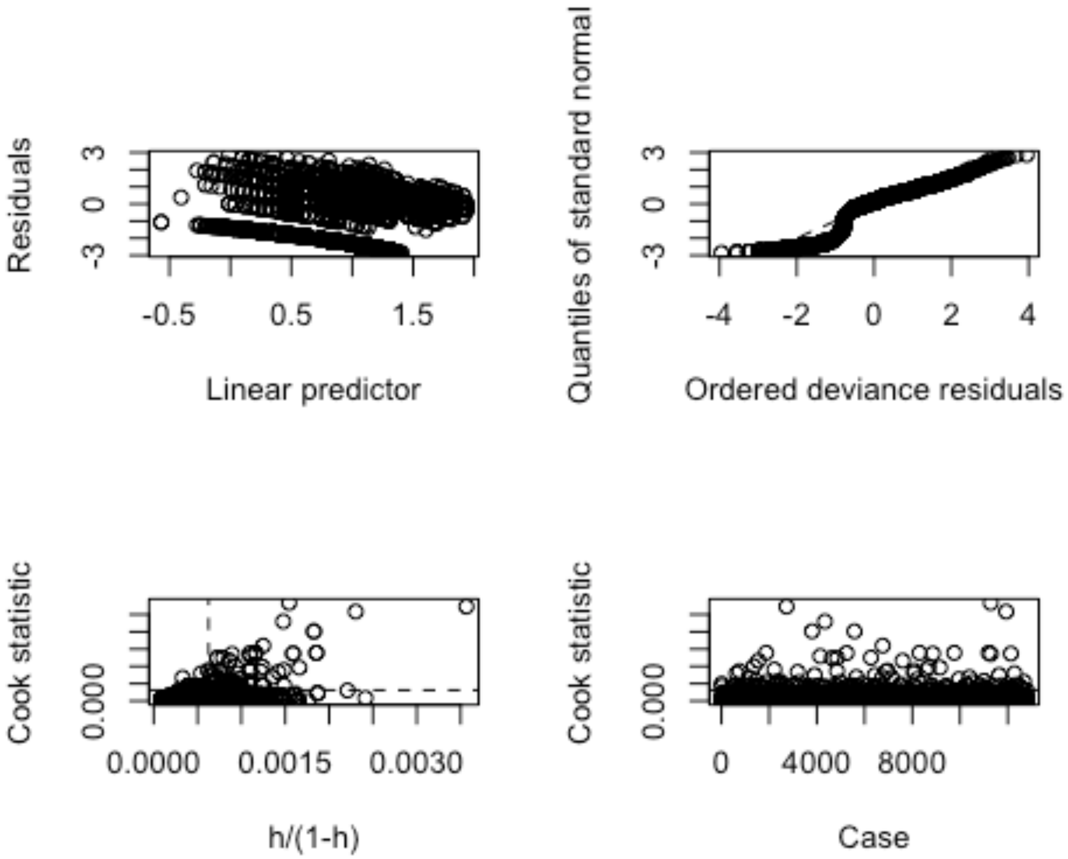
This model tells us that average number of wine sales increases when wines have a lower concentration of chlorides. This makes sense considering that high chloride makes the wine taste salty and not as good according to certain documentation. A higher alcohol conentration is a good indicator of a better quality wine so it makes sense to increase as number of wine units sold increases. The same story is reflected when we convert exponents to odds ratios. With the odds ratio, we can see that as wine sales are more likely to increase when wine rating increases. We managed to make a simpler model that still has the same G statistic, meaning only .20 of the proportion of deviance is explained by this model. http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-20612015000100095

Lets take model five and build a poisson regression model using smoothing splines. We will also use predictors that had a higher correlation with the response variable. These predictors are labelappeal, STARS, and acidindex.

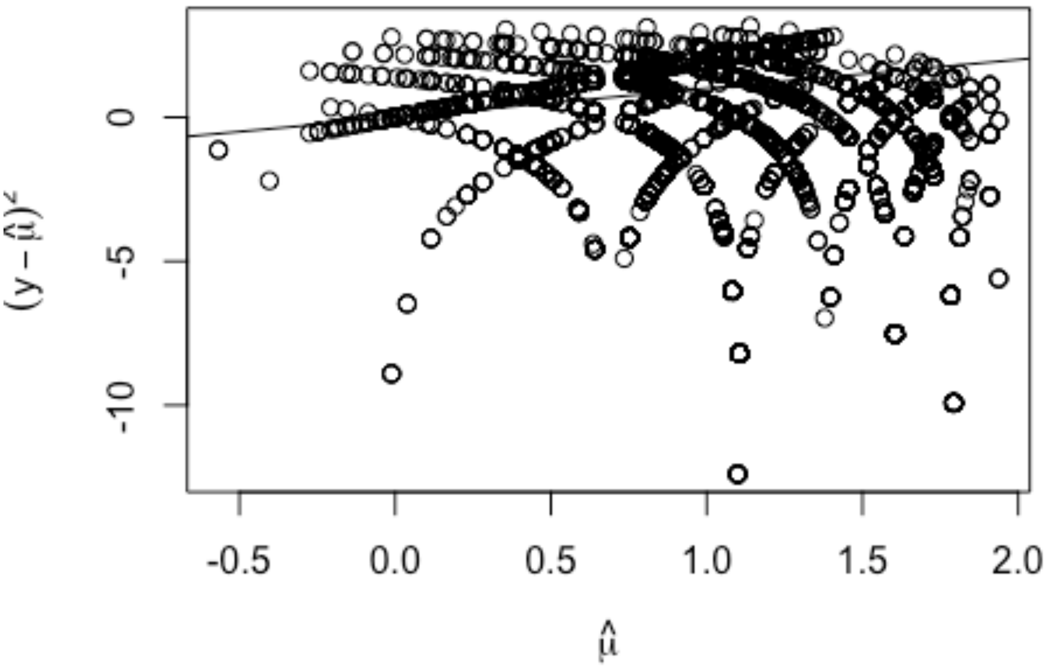
```
## Loading required package: splines

## Loading required package: foreach

## Loaded gam 1.15
```

```
##
## Call: gam(formula = TARGET ~ s(LabelAppeal) + s(AcidIndex) +
s(STARS),
##     family = poisson(link = log), data = wine_training3)
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8576 -0.5427  0.1516  0.6300  2.8615
##
## (Dispersion Parameter for poisson family taken to be 1)
##
##     Null Deviance: 22860.89 on 12794 degrees of freedom
## Residual Deviance: 17735.95 on 12783 degrees of freedom
## AIC: 49701.97
##
## Number of Local Scoring Iterations: 6
##
## Anova for Parametric Effects
##                   Df  Sum Sq Mean Sq F value    Pr(>F)
## s(LabelAppeal)     1  1954.2 1954.19 2049.64 < 2.2e-16 ***
## s(AcidIndex)       1   812.3  812.31  851.98 < 2.2e-16 ***
## s(STARS)           1  1305.1 1305.12 1368.87 < 2.2e-16 ***
## Residuals      12783 12187.7    0.95
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                Npar Df Npar Chisq    P(Chi)
## (Intercept)
## s(LabelAppeal)       3      58.53 1.212e-12 ***
## s(AcidIndex)         3     184.12 < 2.2e-16 ***
## s(STARS)             2     578.91 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Anova for Nonparametric Effects
##                Npar Df Npar Chisq    P(Chi)
## (Intercept)
## s(LabelAppeal)       3      58.53 1.212e-12 ***
## s(AcidIndex)         3     184.12 < 2.2e-16 ***
## s(STARS)             2     578.91 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



```
##      res.deviance    df             p
## [1,]     17735.95 12783 3.975734e-169

##                 Odds ratio     2.5 %    97.5 %
## (Intercept)      4.5606391 4.2257272 4.9220947
## s(LabelAppeal)   1.2076470 1.1931274 1.2223432
## s(AcidIndex)     0.8902623 0.8822326 0.8983651
## s(STARS)         1.2511907 1.2360710 1.2664953

## [1] 0.2241794
```

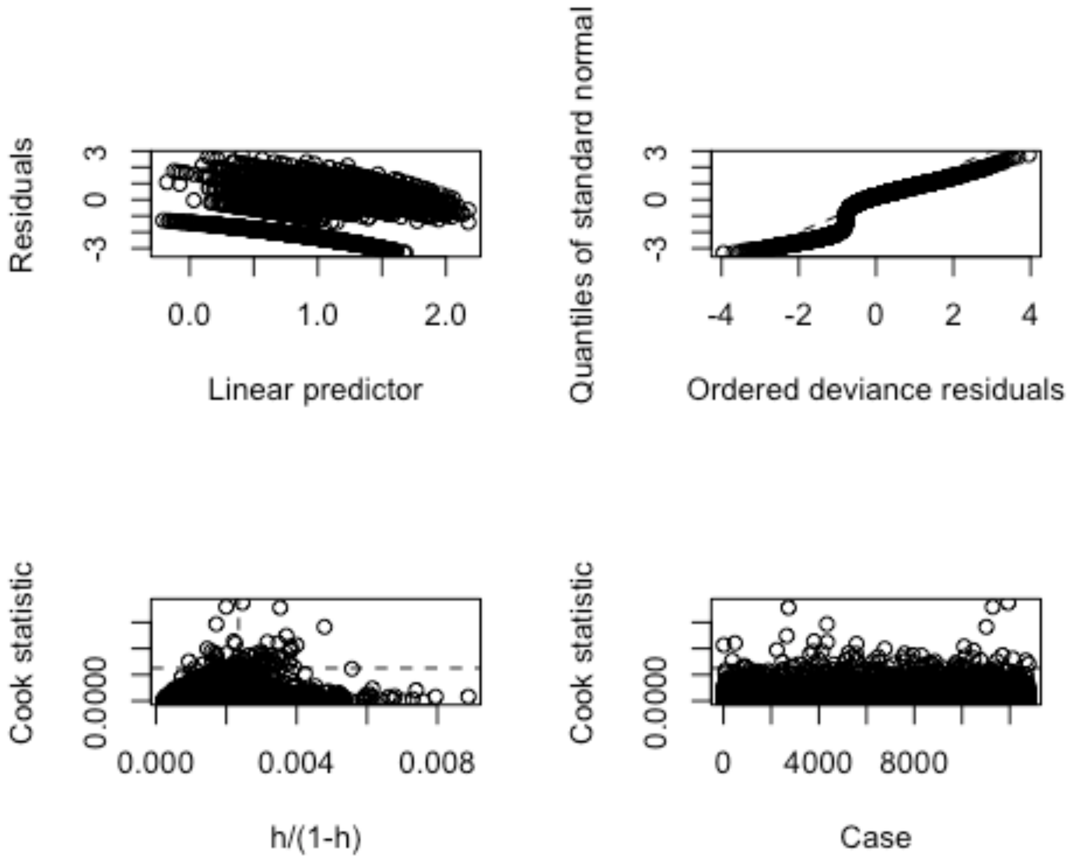


```
## [1] 3.975734e-169
```

The poisson model built with smoothing splines yielded the best psuedo r squared value. The predictors are significant with low p values and the smoothing parameters all yield low p values as well. This indicates that there is evidence that the selected predictors form a good overall fit. Using splines yields a marginally better psuedo r square but has better proportion. The odds ratios in this simple model are also the most interpretable. We see that wine sales are 6 times more likley to increase with a unit increase in label appeal and stars. This makes sense since label appeal measures how desirable a wine looks to a customer. Stars is a quality rating. Both stars and label appeal lead to increased sales.

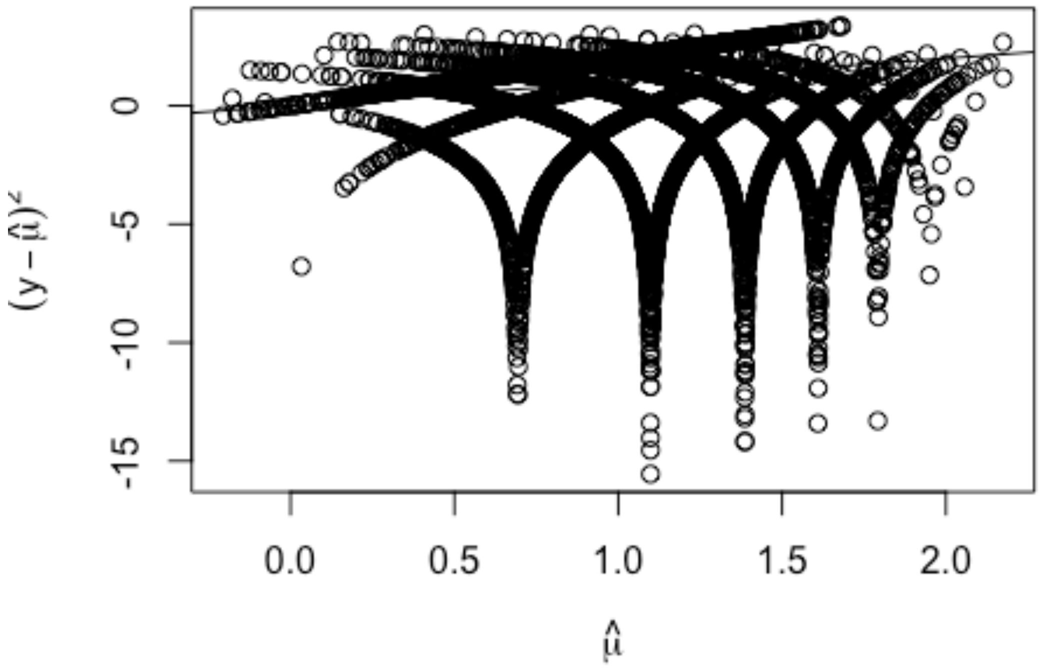
We proceed to building negative binomial models and optimizing said models.

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached

## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached

##
## Call:
## glm.nb(formula = TARGET ~ FixedAcidity + VolatileAcidity +
CitricAcid + 
##     ResidualSugar + Chlorides + FreeSulfurDioxide + 
TotalSulfurDioxide + 
##     Density + pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + 
##     STARS, data = wine_training3, init.theta = 39167.5272, link = 
log)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -3.2791  -0.5084   0.1987   0.6366   2.7592  
## 
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)    
## (Intercept)         2.053e+00  1.963e-01  10.460  < 2e-16 ***
## FixedAcidity       -7.867e-04  1.046e-03  -0.752 0.452214    
## VolatileAcidity    -5.881e-02  9.416e-03  -6.246 4.22e-10 ***
## CitricAcid          1.734e-02  8.292e-03   2.091 0.036549 *  
## ResidualSugar       1.821e-05  2.078e-04   0.088 0.930181    
## Chlorides          -4.456e-02  2.230e-02  -1.998 0.045711 *  
## FreeSulfurDioxide   9.112e-05  4.782e-05   1.905 0.056717 .  
## TotalSulfurDioxide  1.167e-04  3.165e-05   3.688 0.000226 ***
## Density            -4.520e-01  1.922e-01  -2.352 0.018661 *  
## pH                 -2.349e-02  7.636e-03  -3.077 0.002094 ** 
## Sulphates          -2.297e-02  8.229e-03  -2.791 0.005248 ** 
## Alcohol             5.905e-03  1.445e-03   4.087 4.37e-05 ***
## LabelAppeal         1.963e-01  6.021e-03  32.605  < 2e-16 ***
## AcidIndex          -1.233e-01  4.454e-03 -27.681  < 2e-16 ***
## STARS               2.211e-01  6.466e-03  34.200  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for Negative Binomial(39167.53) family taken 
to be 1)
## 
##     Null deviance: 22860  on 12794  degrees of freedom
## Residual deviance: 18474  on 12780  degrees of freedom
## AIC: 50449
## 
## Number of Fisher Scoring iterations: 1
## 
## 
##               Theta:  39168 
##           Std. Err.:  59671 
## Warning while fitting theta: iteration limit reached 
## 
##  2 x log-likelihood:  -50416.88

## Warning in anova.negbin(nmod1, test = "Chisq"): tests made without 
re-
## estimating 'theta'
## Analysis of Deviance Table
##
## Model: Negative Binomial(39167.53), link: log
##
## Response: TARGET
##
## Terms added sequentially (first to last)
##
##
##                     Df Deviance Resid. Df Resid. Dev  Pr(>Chi)    
## NULL                                12794      22860              
## FixedAcidity         1    44.59     12793      22815 2.432e-11 ***
## VolatileAcidity      1    77.88     12792      22737 < 2.2e-16 ***
## CitricAcid           1     2.84     12791      22734 0.0916850 .  
## ResidualSugar        1     0.15     12790      22734 0.6973393    
## Chlorides            1    11.55     12789      22723 0.0006774 ***
## FreeSulfurDioxide    1     8.03     12788      22715 0.0045905 ** 
## TotalSulfurDioxide   1    13.89     12787      22701 0.0001939 ***
## Density              1    20.14     12786      22681 7.205e-06 ***
## pH                   1     1.03     12785      22680 0.3099861    
## Sulphates            1    13.55     12784      22666 0.0002317 ***
## Alcohol              1    62.19     12783      22604 3.117e-15 ***
## LabelAppeal          1  1974.97     12782      20629 < 2.2e-16 ***
## AcidIndex            1  1006.15     12781      19623 < 2.2e-16 ***
## STARS                1  1148.84     12780      18474 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



```
##      res.deviance    df            p
## [1,]     18473.87 12780 1.634151e-216
```

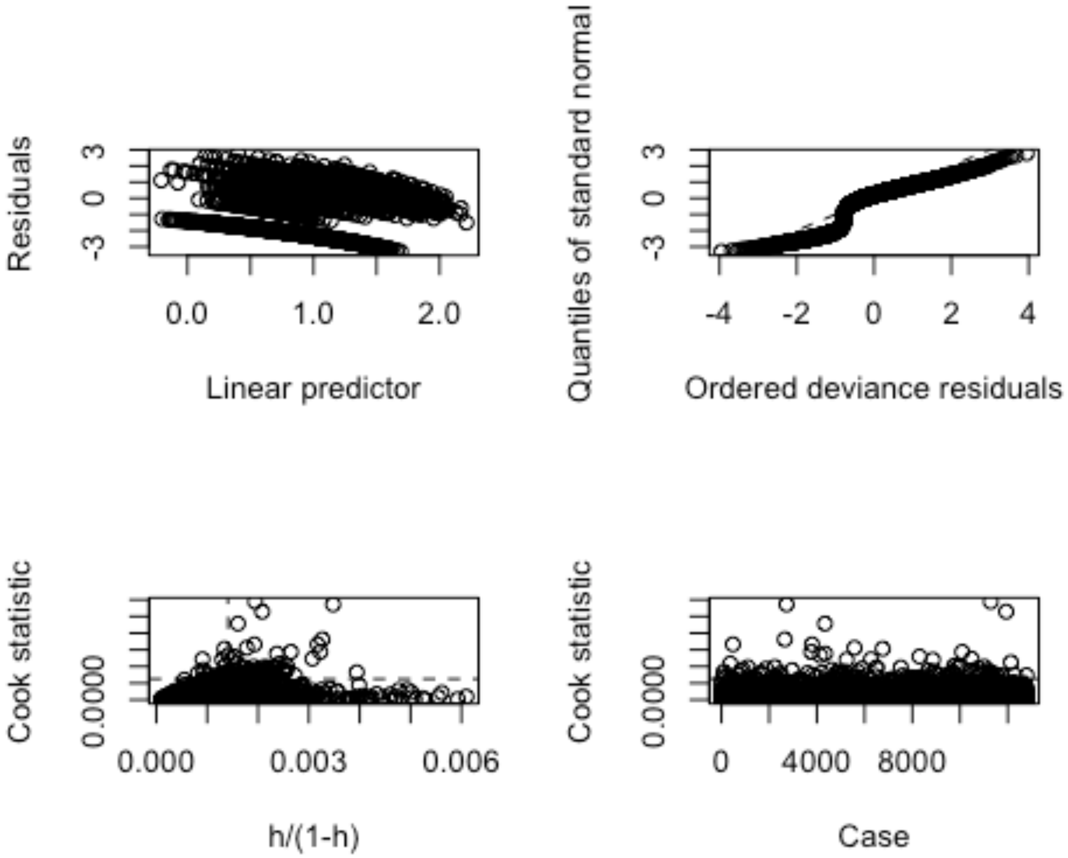
```
##                    Odds ratio     2.5 %     97.5 %
## (Intercept)         7.7905888 5.3029584 11.4451727
## FixedAcidity        0.9992136 0.9971663  1.0012652
## VolatileAcidity     0.9428829 0.9256408  0.9604462
## CitricAcid          1.0174879 1.0010852  1.0341593
## ResidualSugar       1.0000182 0.9996110  1.0004256
## Chlorides           0.9564211 0.9155186  0.9991509
## FreeSulfurDioxide   1.0000911 0.9999974  1.0001849
## TotalSulfurDioxide  1.0001167 1.0000547  1.0001788
## Density             0.6363314 0.4366222  0.9273867
## pH                  0.9767818 0.9622725  0.9915099
## Sulphates           0.9772904 0.9616539  0.9931812
## Alcohol             1.0059225 1.0030779  1.0087752
## LabelAppeal         1.2168961 1.2026209  1.2313408
## AcidIndex           0.8840196 0.8763368  0.8917697
## STARS               1.2475000 1.2317897  1.2634108
```



```
## [1] 1.634151e-216
```

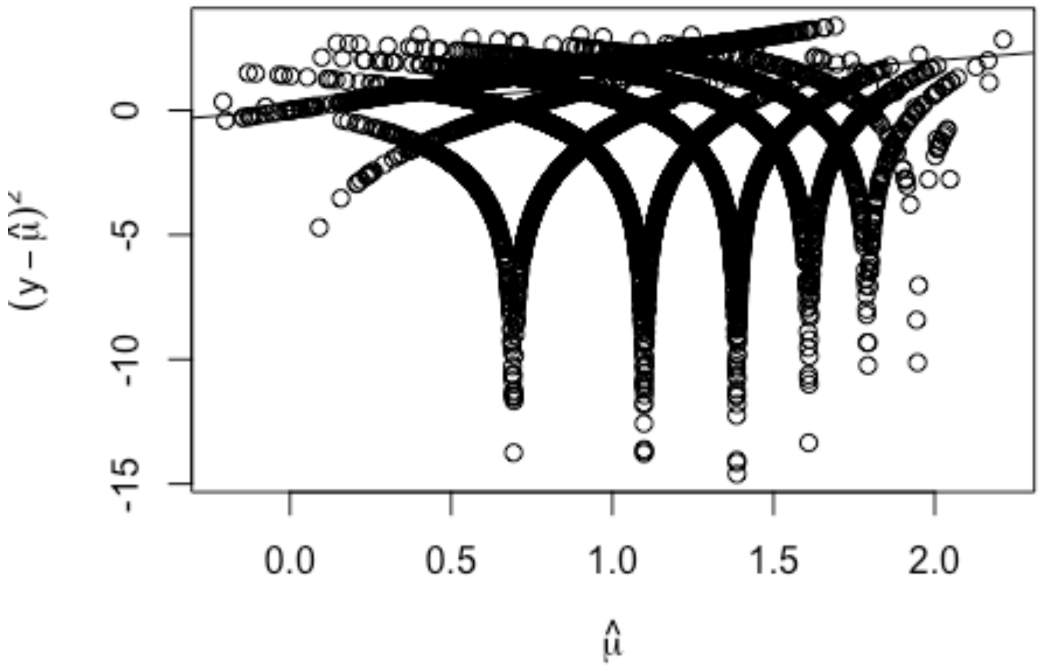
According to this model, fixed acidity, residual sugar and free sulfur dioxide are not significant. Citric acid and ph also do not show evidence of being a good model fit. Lets remove those variables and refit the model. This model has the same fit as the poisson model.

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached

## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached

##
## Call:
## glm.nb(formula = TARGET ~ VolatileAcidity + Chlorides +
TotalSulfurDioxide +
##     Density + Alcohol + LabelAppeal + AcidIndex + STARS, data =
wine_training3,
##     init.theta = 39133.88486, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2950  -0.4980   0.2044   0.6381   2.7688
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         1.980e+00  1.942e-01  10.192  < 2e-16 ***
## VolatileAcidity    -5.985e-02  9.417e-03  -6.356 2.07e-10 ***
## Chlorides          -4.641e-02  2.230e-02  -2.081 0.037428 *
## TotalSulfurDioxide  1.184e-04  3.163e-05   3.744 0.000181 ***
## Density            -4.590e-01  1.921e-01  -2.389 0.016889 *
## Alcohol             5.916e-03  1.445e-03   4.095 4.23e-05 ***
## LabelAppeal         1.964e-01  6.018e-03  32.638  < 2e-16 ***
## AcidIndex          -1.230e-01  4.383e-03 -28.069  < 2e-16 ***
## STARS               2.213e-01  6.466e-03  34.223  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(39133.88) family taken
to be 1)
##
##     Null deviance: 22860  on 12794  degrees of freedom
## Residual deviance: 18500  on 12786  degrees of freedom
## AIC: 50463
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  39134
##           Std. Err.:  59912
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -50442.87

## Warning in anova.negbin(nmod2, test = "Chisq"): tests made without
re-
## estimating 'theta'

## Analysis of Deviance Table
##
## Model: Negative Binomial(39133.88), link: log
##
## Response: TARGET
##
## Terms added sequentially (first to last)
##
##
##                    Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                               12794      22860
## VolatileAcidity     1    79.19     12793      22780 < 2.2e-16 ***
## Chlorides           1    11.64     12792      22769 0.0006439 ***
## TotalSulfurDioxide  1    14.82     12791      22754 0.0001184 ***
## Density             1    20.17     12790      22734 7.093e-06 ***
## Alcohol             1    62.67     12789      22671 2.449e-15 ***
## LabelAppeal         1  1980.48     12788      20691 < 2.2e-16 ***
## AcidIndex           1  1040.51     12787      19650 < 2.2e-16 ***
## STARS               1  1150.36     12786      18500 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



```
##      res.deviance    df             p
## [1,]     18499.85 12786 8.948867e-218
```

```
##                     Odds ratio     2.5 %     97.5 %
## (Intercept)          7.2409441 4.9482358 10.5959526
## VolatileAcidity      0.9419023 0.9246764  0.9594490
## Chlorides            0.9546526 0.9138260  0.9973032
## TotalSulfurDioxide   1.0001184 1.0000564  1.0001804
## Density              0.6319319 0.4336569  0.9208613
## Alcohol              1.0059340 1.0030893  1.0087868
## LabelAppeal          1.2170376 1.2027667  1.2314778
## AcidIndex            0.8842408 0.8766772  0.8918697
## STARS                1.2476934 1.2319803  1.2636069
```



```
## [1] 8.948867e-218
```

Our negative binomial models so far indicate there is not much difference between the earlier iterations of our posisson negative binomial models.

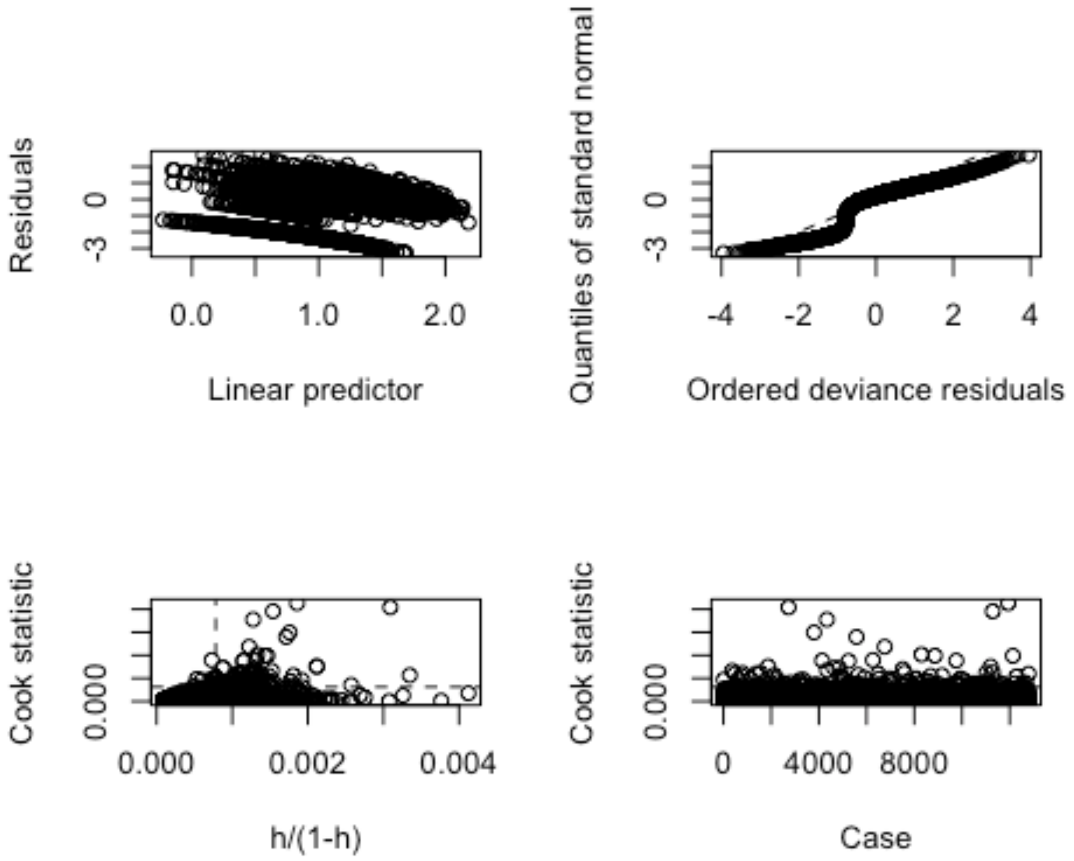
Lets build a standard negative binomial with the three variables from the last iteration of the poisson model.

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached

## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached

## 
## Call:
## glm.nb(formula = TARGET ~ Alcohol + LabelAppeal + AcidIndex + 
##     STARS, data = wine_training3, init.theta = 38811.50732, link = 
log)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -3.2844  -0.4895   0.2118   0.6315   2.7140  
## 
```

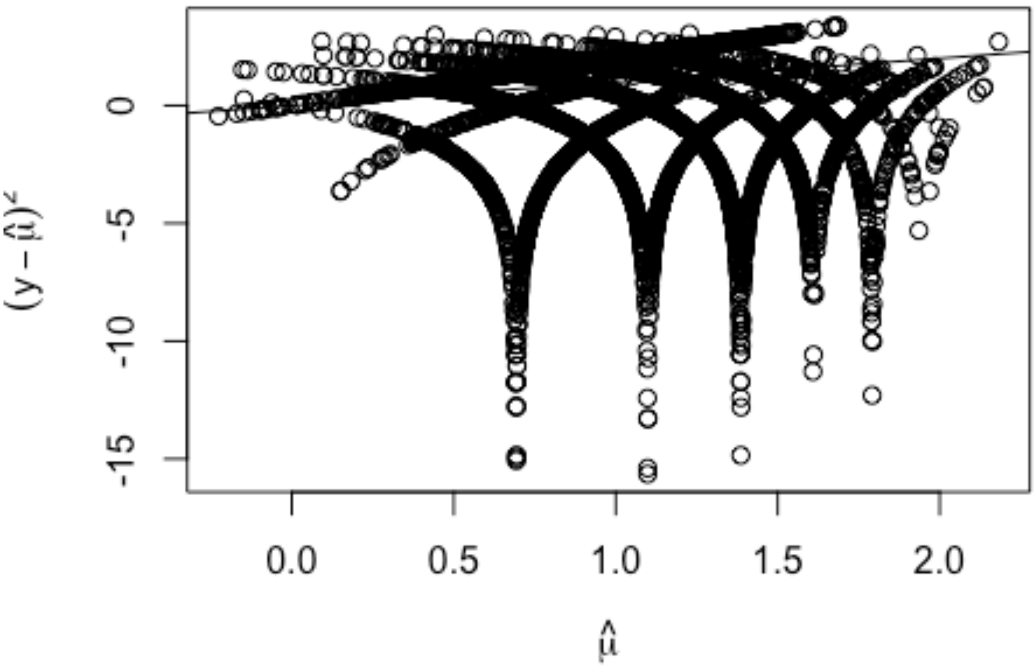
```
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.513600   0.040623  37.260  < 2e-16 ***
## Alcohol      0.005629   0.001444   3.898 9.72e-05 ***
## LabelAppeal  0.196719   0.006015  32.704  < 2e-16 ***
## AcidIndex   -0.124695   0.004369 -28.539  < 2e-16 ***
## STARS        0.222389   0.006461  34.421  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(38811.51) family taken
to be 1)
##
##     Null deviance: 22860  on 12794  degrees of freedom
## Residual deviance: 18566  on 12790  degrees of freedom
## AIC: 50521
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  38812
##           Std. Err.:  60163
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -50509.49
```

```
## Warning in anova.negbin(nmod4, test = "Chisq"): tests made without
re-
## estimating 'theta'
```

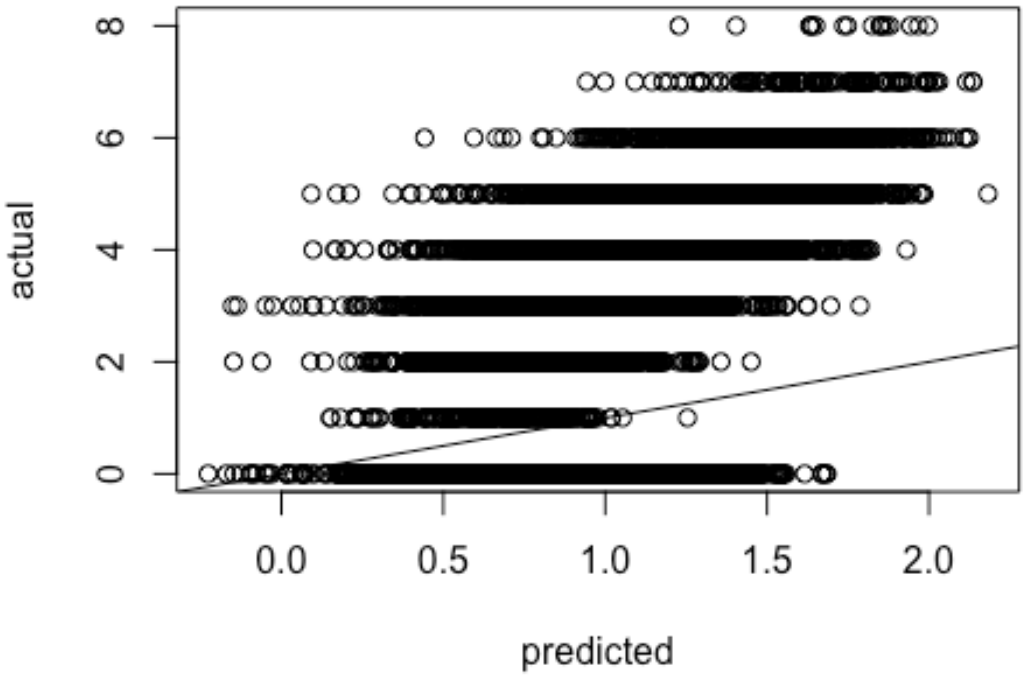
```
## Analysis of Deviance Table
##
## Model: Negative Binomial(38811.51), link: log
##
## Response: TARGET
##
## Terms added sequentially (first to last)
##
##
##             Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                        12794      22860
## Alcohol      1    59.54     12793      22800 1.201e-14 ***
## LabelAppeal  1  1990.44     12792      20810 < 2.2e-16 ***
## AcidIndex    1  1079.60     12791      19730 < 2.2e-16 ***
## STARS        1  1163.62     12790      18566 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



```
##      res.deviance    df             p
## [1,]     18566.47 12790 6.094005e-222

##              Odds ratio     2.5 %    97.5 %
## (Intercept)  4.5430578 4.1953678 4.9195625
## Alcohol      1.0056448 1.0028022 1.0084955
## LabelAppeal  1.2174018 1.2031337 1.2318391
## AcidIndex    0.8827663 0.8752389 0.8903585
## STARS        1.2490567 1.2333395 1.2649741
```



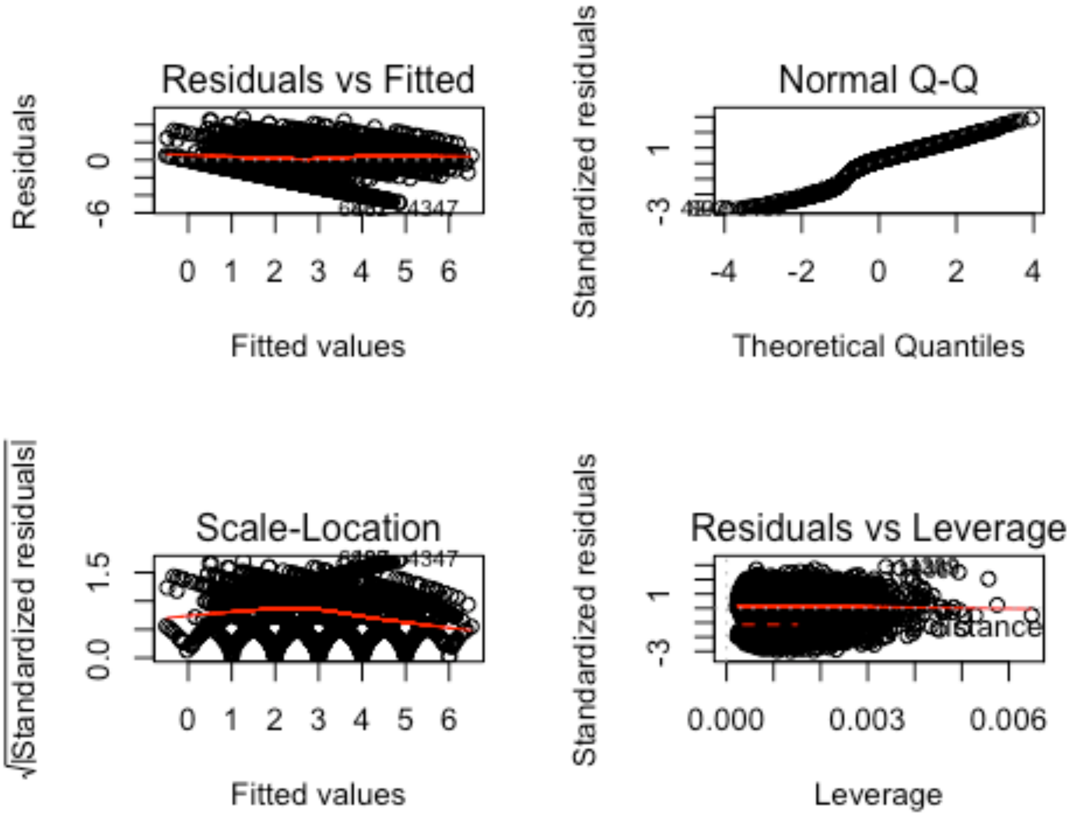
$(y - \hat{\mu})^2$
0
-5
-10
-15
0.0
0.5
1.0
1.5
2.0
$\hat{\mu}$



```
## [1] 6.094005e-222
```

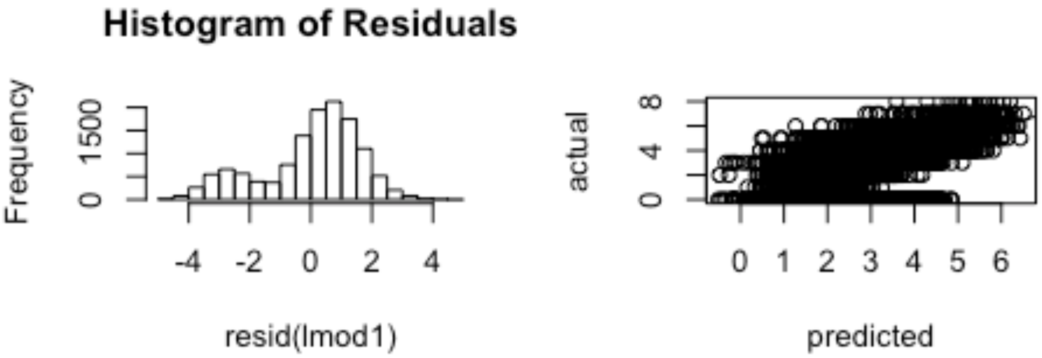
Finall, we can attempt to build linear regression models. There are some challenges to building linear regression models. They include the fact that the response variable is ero inflated. We can see that from our histogram.

```
##
## Attaching package: 'olsrr'

## The following object is masked from 'package:MASS':
##
##     cement

## The following object is masked from 'package:faraway':
##
##     hsb

## The following object is masked from 'package:datasets':
##
##     rivers

##
## Call:
## lm(formula = TARGET ~ FixedAcidity + VolatileAcidity + CitricAcid +
##     ResidualSugar + Chlorides + FreeSulfurDioxide + 
TotalSulfurDioxide + 
##     Density + pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + 
##     STARS, data = wine_training3)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -4.8571 -0.7435  0.3683  1.1240  4.7336 
## 
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)    
## (Intercept)         5.467e+00  5.544e-01   9.861  < 2e-16 ***
## FixedAcidity       -1.900e-03  2.935e-03  -0.647 0.517368    
## VolatileAcidity    -1.686e-01  2.600e-02  -6.483 9.29e-11 ***
## CitricAcid          5.345e-02  2.382e-02   2.244 0.024870 *  
## ResidualSugar      -1.419e-04  5.901e-04  -0.240 0.809961    
## Chlorides          -1.435e-01  6.275e-02  -2.287 0.022220 *  
## FreeSulfurDioxide   2.681e-04  1.362e-04   1.969 0.049008 *  
## TotalSulfurDioxide  3.515e-04  9.083e-05   3.870 0.000109 ***
## Density            -1.340e+00  5.441e-01  -2.463 0.013808 *  
## pH                 -6.306e-02  2.160e-02  -2.920 0.003505 ** 
## Sulphates          -6.802e-02  2.297e-02  -2.961 0.003072 ** 
## Alcohol             2.039e-02  4.090e-03   4.985 6.29e-07 ***
## LabelAppeal         5.938e-01  1.691e-02  35.116  < 2e-16 ***
## AcidIndex          -3.292e-01  1.118e-02 -29.454  < 2e-16 ***
## STARS               7.507e-01  1.950e-02  38.488  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.631 on 12780 degrees of freedom
## Multiple R-squared:  0.2841, Adjusted R-squared:  0.2833 
## F-statistic: 362.2 on 14 and 12780 DF,  p-value: < 2.2e-16
```



```
##
##  Breusch Pagan Test for Heteroskedasticity
##  -----------------------------------------
##  Ho: the variance is constant
##  Ha: the variance is not constant
##
##                Data
##  ----------------------------------
##  Response : TARGET
##  Variables: fitted values of TARGET
##
##          Test Summary
##  ------------------------------
##  DF            =    1
##  Chi2          =    9.602864
##  Prob > Chi2   =    0.001942741
```

```
##      FixedAcidity    VolatileAcidity         CitricAcid
##          1.034051           1.003856           1.002462
##     ResidualSugar          Chlorides  FreeSulfurDioxide
##          1.000737           1.001894           1.001163
## TotalSulfurDioxide           Density                 pH
##          1.004620           1.002760           1.004413
##          Sulphates            Alcohol        LabelAppeal
##           1.002218           1.005986           1.092379
##          AcidIndex              STARS
##           1.053469           1.099862
```



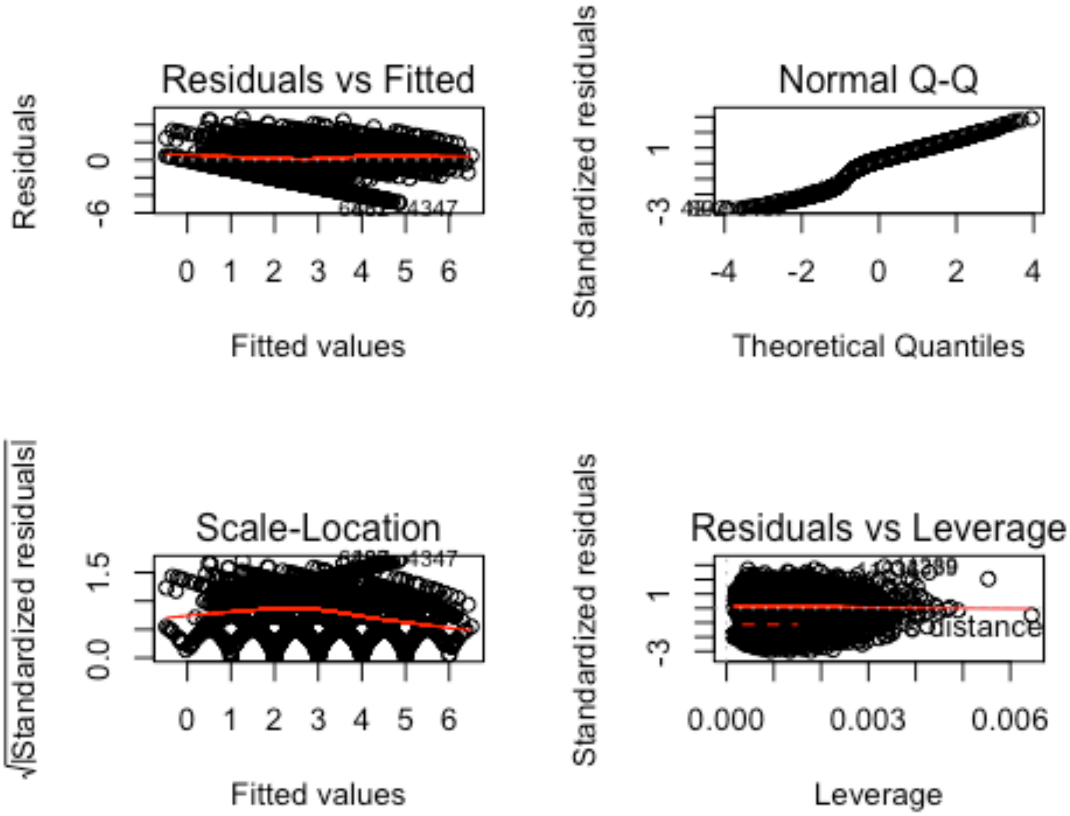
Lets see what step AIC produces

```
## Stepwise Model Path 
## Analysis of Deviance Table
## 
## Initial Model:
## TARGET ~ FixedAcidity + VolatileAcidity + CitricAcid + ResidualSugar 
+ 
##     Chlorides + FreeSulfurDioxide + TotalSulfurDioxide + Density + 
##     pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS
## 
## Final Model:
## TARGET ~ VolatileAcidity + CitricAcid + Chlorides + 
FreeSulfurDioxide + 
##     TotalSulfurDioxide + Density + pH + Sulphates + Alcohol + 
##     LabelAppeal + AcidIndex + STARS
## 
## 
```

```
##              Step Df  Deviance Resid. Df Resid. Dev      AIC
## 1                                  12780   33990.22 12530.95
## 2 - ResidualSugar  1 0.1538105     12781   33990.37 12529.01
## 3  - FixedAcidity  1 1.1201094     12782   33991.49 12527.43
```

Formulate the model generated by step

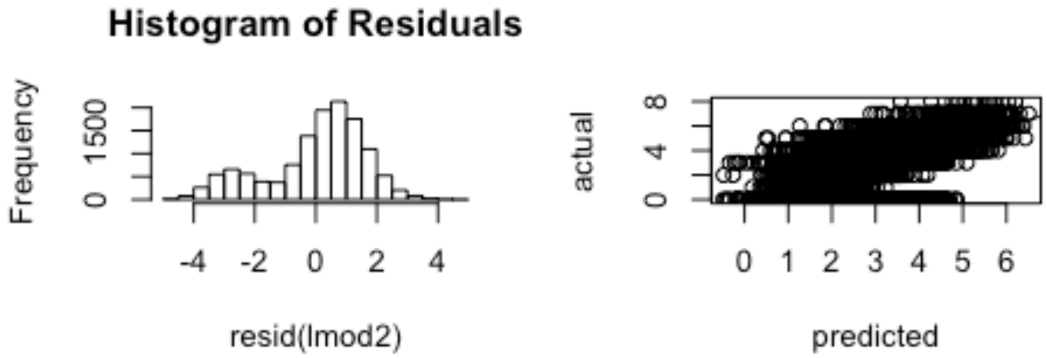
```
lmod2 <- lm(TARGET ~ VolatileAcidity + CitricAcid +
    Chlorides + FreeSulfurDioxide + TotalSulfurDioxide + Density +
    pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS
 ,  data=wine_training3);

summary(lmod2);

## 
## Call:
## lm(formula = TARGET ~ VolatileAcidity + CitricAcid + Chlorides + 
##     FreeSulfurDioxide + TotalSulfurDioxide + Density + pH + 
Sulphates + 
##     Alcohol + LabelAppeal + AcidIndex + STARS, data = 
wine_training3)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -4.8568 -0.7428  0.3693  1.1235  4.7350 
## 
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)    
## (Intercept)         5.455e+00  5.540e-01   9.847  < 2e-16 ***
## VolatileAcidity    -1.686e-01  2.600e-02  -6.487 9.08e-11 ***
## CitricAcid          5.369e-02  2.382e-02   2.254 0.024182 *  
## Chlorides          -1.433e-01  6.275e-02  -2.283 0.022426 *  
## FreeSulfurDioxide   2.685e-04  1.362e-04   1.972 0.048683 *  
## TotalSulfurDioxide  3.515e-04  9.081e-05   3.870 0.000109 ***
## Density            -1.337e+00  5.440e-01  -2.457 0.014022 *  
## pH                 -6.317e-02  2.159e-02  -2.926 0.003444 ** 
## Sulphates          -6.818e-02  2.297e-02  -2.969 0.002996 ** 
## Alcohol             2.040e-02  4.090e-03   4.989 6.16e-07 ***
## LabelAppeal         5.939e-01  1.691e-02  35.122  < 2e-16 ***
## AcidIndex          -3.305e-01  1.100e-02 -30.055  < 2e-16 ***
## STARS               7.507e-01  1.950e-02  38.491  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.631 on 12782 degrees of freedom
## Multiple R-squared:  0.284,  Adjusted R-squared:  0.2834 
## F-statistic: 422.6 on 12 and 12782 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2))
plot(lmod2)
```



```
hist(resid(lmod2), main="Histogram of Residuals");
ols_test_breusch_pagan(lmod2);

##
##   Breusch Pagan Test for Heteroskedasticity
##   -----------------------------------------
##   Ho: the variance is constant
##   Ha: the variance is not constant
##
##                 Data
##   ----------------------------------
##   Response : TARGET
##   Variables: fitted values of TARGET
##
##           Test Summary
##   ------------------------------
##   DF            =    1
##   Chi2          =    9.600033
##   Prob > Chi2   =    0.001945739
```

```
vif(lmod2);

##     VolatileAcidity         CitricAcid          Chlorides
##            1.003832           1.002177           1.001865
##  FreeSulfurDioxide TotalSulfurDioxide            Density 
##           1.001064           1.004416           1.002690 
##                 pH          Sulphates            Alcohol 
##           1.004345           1.002012           1.005931 
##        LabelAppeal          AcidIndex              STARS 
##           1.092339           1.019707           1.099804
```

```
plot(predict(lmod2),wine_training3$TARGET,
      xlab="predicted",ylab="actual")
 abline(a=0,b=1)
```

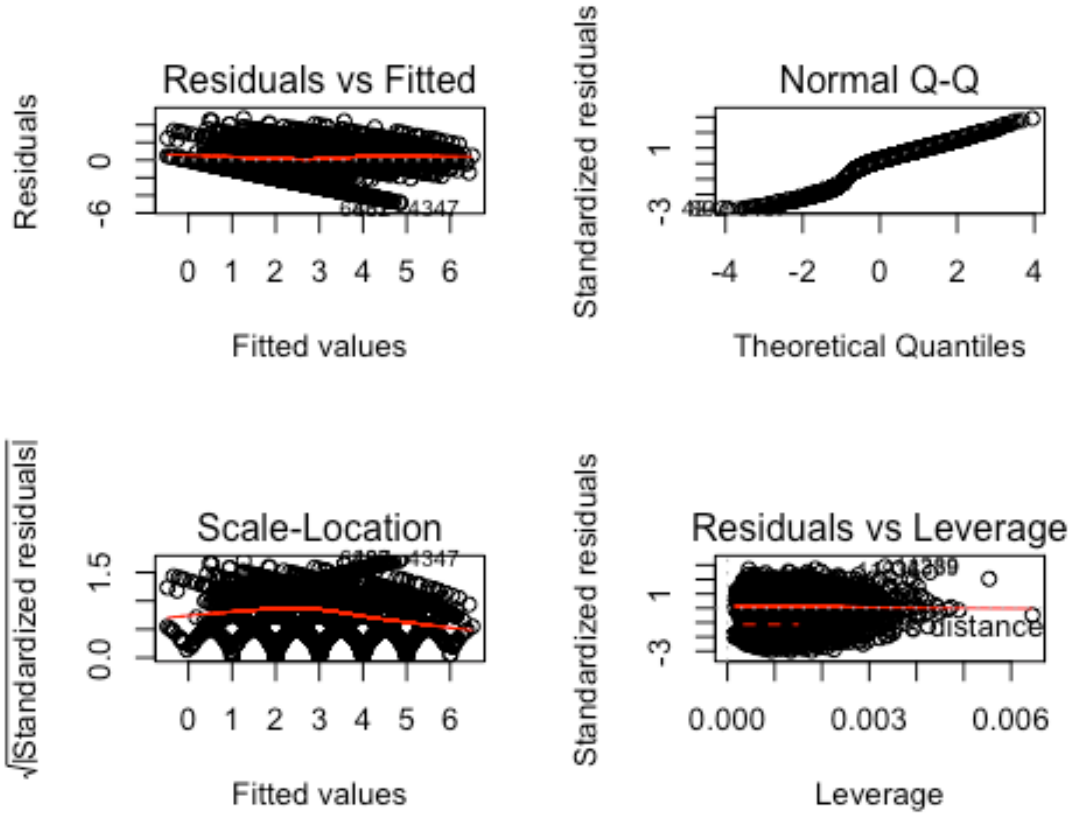


The results of the constant variance test indicate that there is non constant variance,however residuals are closely normal in the qq plot. The VIF numbers are mostly around one, meaning there is not indication of strong multi-colinearity.

We conclude the model building by constructing an additive linear model

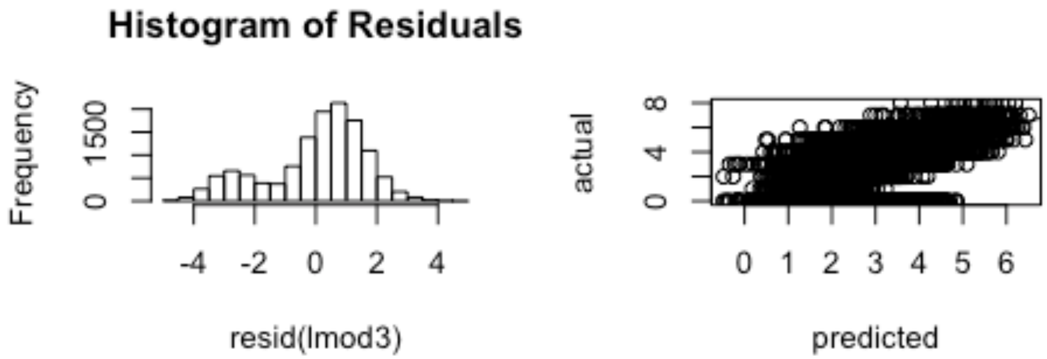
```
## 
## Attaching package: 'ISLR'

## The following object is masked from 'package:vcd':
## 
##     Hitters
```

```
##
## Call:
## lm(formula = TARGET ~ VolatileAcidity + CitricAcid + Chlorides +
##     FreeSulfurDioxide + TotalSulfurDioxide + Density + pH +
Sulphates +
##     Alcohol + s(LabelAppeal) + AcidIndex + s(STARS), data =
wine_training3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8568 -0.7428  0.3693  1.1235  4.7350
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         5.455e+00  5.540e-01   9.847  < 2e-16 ***
## VolatileAcidity    -1.686e-01  2.600e-02  -6.487 9.08e-11 ***
## CitricAcid          5.369e-02  2.382e-02   2.254 0.024182 *
## Chlorides          -1.433e-01  6.275e-02  -2.283 0.022426 *
## FreeSulfurDioxide   2.685e-04  1.362e-04   1.972 0.048683 *
## TotalSulfurDioxide  3.515e-04  9.081e-05   3.870 0.000109 ***
## Density            -1.337e+00  5.440e-01  -2.457 0.014022 *
## pH                 -6.317e-02  2.159e-02  -2.926 0.003444 **
## Sulphates          -6.818e-02  2.297e-02  -2.969 0.002996 **
## Alcohol             2.040e-02  4.090e-03   4.989 6.16e-07 ***
## s(LabelAppeal)      5.939e-01  1.691e-02  35.122  < 2e-16 ***
## AcidIndex          -3.305e-01  1.100e-02 -30.055  < 2e-16 ***
## s(STARS)            7.507e-01  1.950e-02  38.491  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.631 on 12782 degrees of freedom
## Multiple R-squared:  0.284,  Adjusted R-squared:  0.2834
## F-statistic: 422.6 on 12 and 12782 DF,  p-value: < 2.2e-16
```



```
##
##  Breusch Pagan Test for Heteroskedasticity
##  -----------------------------------------
##  Ho: the variance is constant
##  Ha: the variance is not constant
##
##                Data
##  ----------------------------------
##  Response : TARGET
##  Variables: fitted values of TARGET
##
##          Test Summary
##  ------------------------------
##  DF            =    1
##  Chi2          =    9.600033
##  Prob > Chi2   =    0.001945739
```

```
##    VolatileAcidity         CitricAcid          Chlorides
##           1.003832           1.002177           1.001865
##  FreeSulfurDioxide TotalSulfurDioxide            Density
##           1.001064           1.004416           1.002690
##                 pH          Sulphates            Alcohol
##           1.004345           1.002012           1.005931
##     s(LabelAppeal)          AcidIndex           s(STARS) 
##           1.092339           1.019707           1.099804
```



Building an additive linear model does not seem to improve what we already know from the existing linear models.

In order have a large enough pool of models to pick from, we should consider the case of zero inflation models. This model type can be addapted for poisson regression or negative binomial regression, which are two model types we have considered till this point. Right off the bat, we can disregard the linear models due to the nature of the response variable.

The provided documentation states "Zero-inflated poisson regression is used to model count data that has an excess of zero counts. Further, theory suggests that the excess zeros are generated by a separate process from the count values and that the excess zeros can be modeled independently" https://stats.idre.ucla.edu/r/dae/zip/

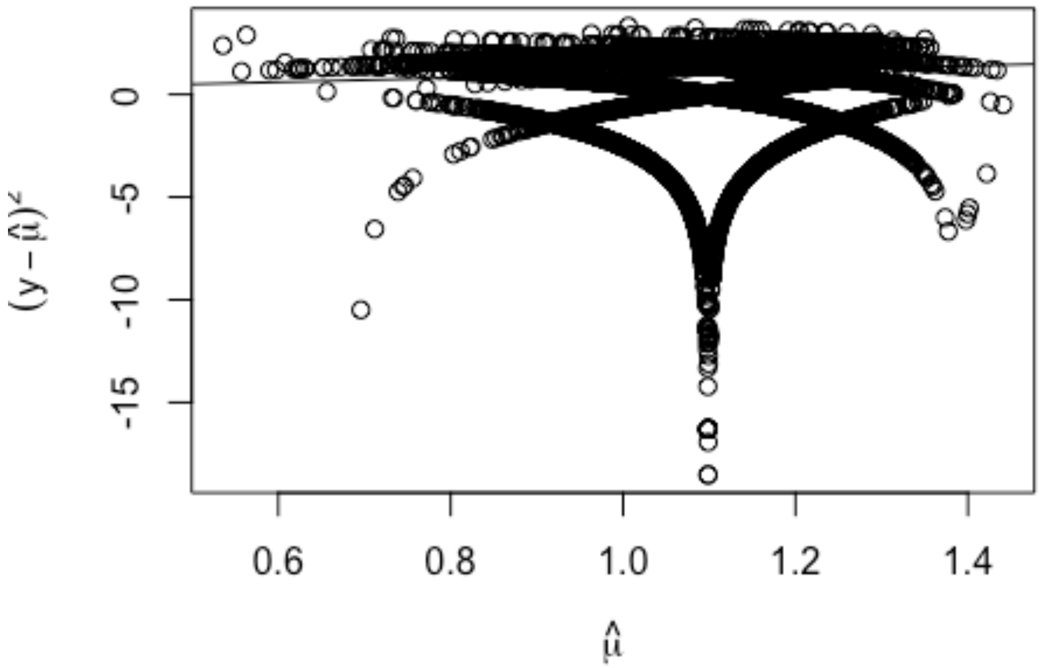
Just how many zeroes are present in our dataset?

```
##             TARGET      FixedAcidity    VolatileAcidity 
##               2734                39                 18 
##         CitricAcid     ResidualSugar          Chlorides 
##                115                 6                  5
##  FreeSulfurDioxide TotalSulfurDioxide            Density
##                 11                  7                  0
##                 pH          Sulphates            Alcohol
##                  0                 22                  2
##        LabelAppeal          AcidIndex              STARS
##               5617                  0                  0
```

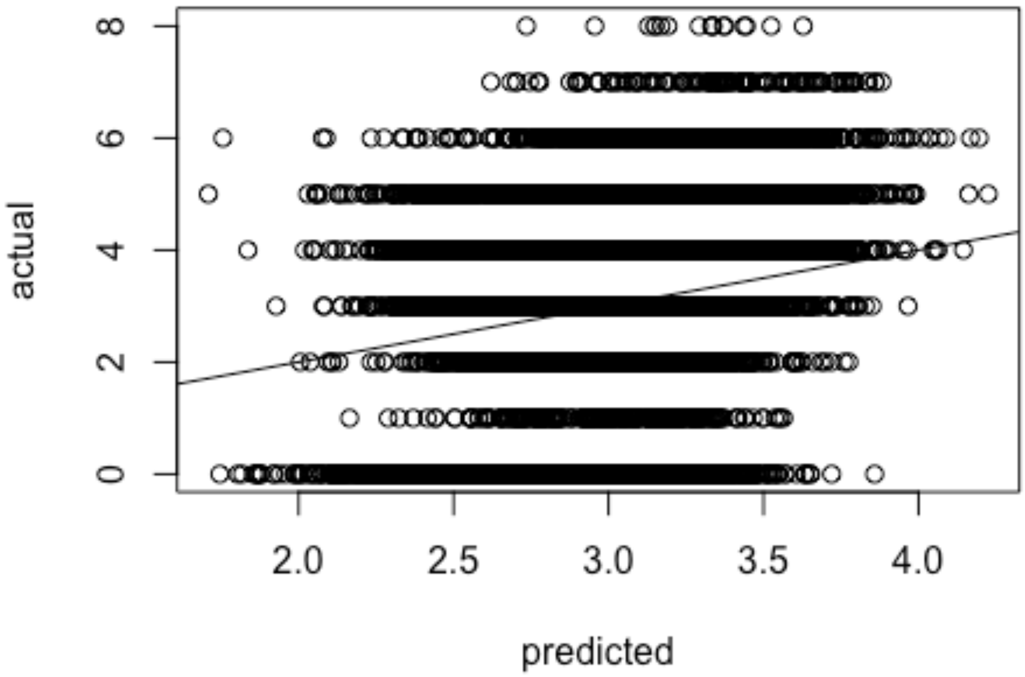
There is a substantial amount of data that contains zero values, hence we are more than justified to use zero inflation model types. We also have some logical arguments to consider. First, lets understand our data here. We have a count of the number of wine cases sold based on marketing and chemical attributes associated with that wine. A use case could be that a stakeholder also wants to predict the probability of a wine having a zero label appeal or a zero quality rating. This could be telling of how wine sales are impacted. We also have a goodness of fit motivation to try something different. The goodness of fit tests suggest our models are not good fits.

Poisson Zero Inflated Model-We will use only the variables from the last poisson model with our lofit predictors being STARS and LabelAppeal

```
## Loading required package: pscl

## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis

## 
## Call:
## zeroinfl(formula = TARGET ~ Alcohol + AcidIndex | STARS + 
LabelAppeal, 
##     data = wine_training3)
## 
## Pearson residuals:
##     Min      1Q  Median      3Q     Max 
## -1.5470 -0.5324  0.1649  0.6139  2.3798 
## 
## Count model coefficients (poisson with log link):
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  1.535087   0.044167  34.756  < 2e-16 ***
## Alcohol      0.010018   0.001480   6.767 1.31e-11 ***
## AcidIndex   -0.041976   0.005369  -7.818 5.37e-15 ***
## 
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -0.16377    0.06729  -2.434   0.0149 *  
## STARS       -0.65848    0.03445 -19.114  < 2e-16 ***
## LabelAppeal  0.18011    0.02862   6.294  3.1e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 12
## Log-likelihood: -2.424e+04 on 6 Df
```



```
## Vuong Non-Nested Hypothesis Test-Statistic:
## (test-statistic is asymptotically distributed N(0,1) under the
##  null that the models are indistinguishible)
## -------------------------------------------------------------
##               Vuong z-statistic             H_A    p-value
## Raw                    12.26657 model1 > model2 < 2.22e-16
## AIC-corrected          12.31571 model1 > model2 < 2.22e-16
## BIC-corrected          12.49891 model1 > model2 < 2.22e-16
```



The vuong test indicates that zero inflated poisson model is better than the regular poisson model due to the small p value. Our predictors in the count and inflation portions of the model are significant.

Lets see how this model compares to the null model

```
## 'log Lik.' 3.404314e-112 (df=6)
```

We can conclude that our model is staistically significant based on this hypothesis test.

IV) Model Selection

We need to parition a test and control data set from our larger training subset in order to predict model accuracy before we deploy on the evaluation data.

Lets show why the zero inflation poisson regression model is our best bet.

```
## structure(c(1.5350871583595, 0.0100184890625652, -0.0419759809109108
## ), .Names = c("(Intercept)", "Alcohol", "AcidIndex"))

## structure(c(-0.163773933205731, -0.658480007306658, 
0.180109965808006
## ), .Names = c("(Intercept)", "STARS", "LabelAppeal"))
```

We extract the logit portion of linear portion of our model.

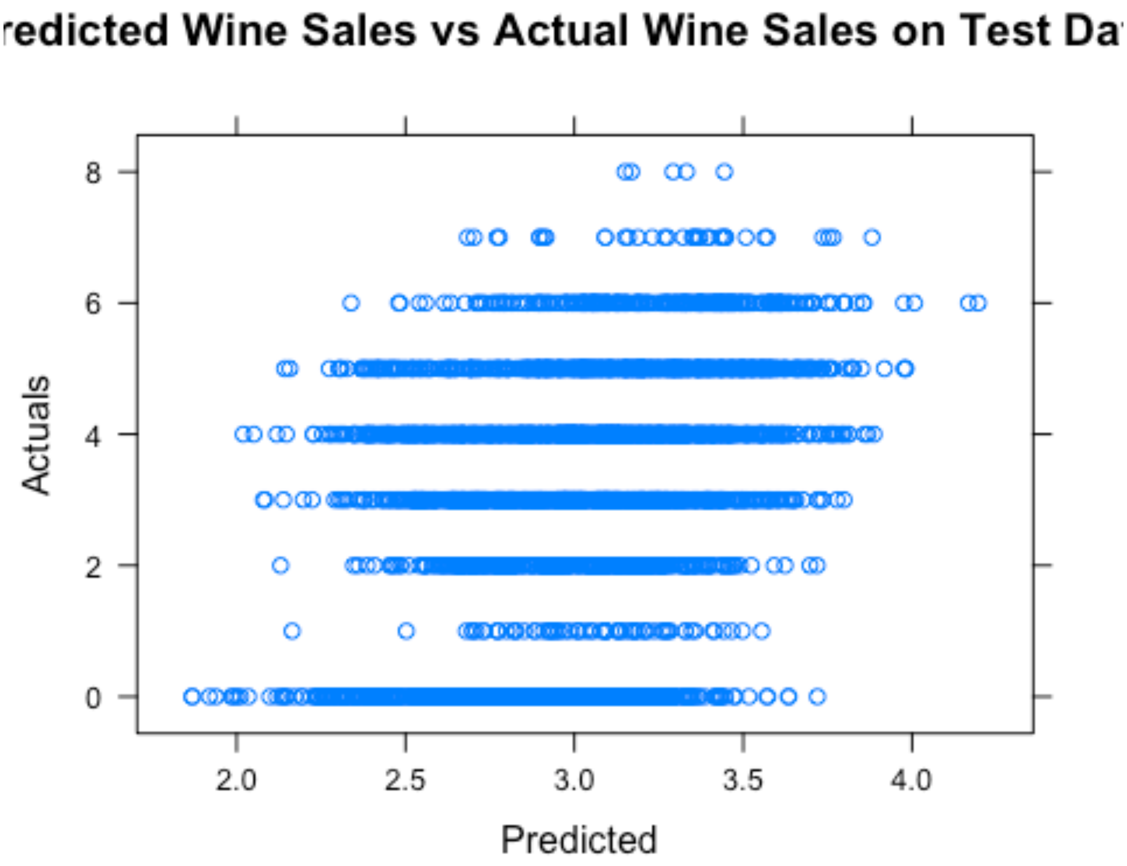
```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = wine_training3, statistic = f, R = 100, parallel =
"snow",
##     ncpus = 4)
##
##
## Bootstrap Statistics :
##          original        bias     std. error
## t1*   1.535087158 -2.650632e-03 3.572330e-02
## t2*   0.044167429 -7.028079e-06 6.540012e-04
## t3*   0.010018488 -9.778092e-05 1.007721e-03
## t4*   0.001480428 -3.339340e-06 1.416369e-05
## t5*  -0.041975982  5.950387e-04 4.414965e-03
## t6*   0.005369237  1.801399e-07 9.323353e-05
## t7*  -0.163773933  1.171076e-02 5.037472e-02
## t8*   0.067289071  3.238981e-06 5.425711e-04
## t9*  -0.658480007 -6.222192e-03 2.352005e-02
## t10*  0.034451020  3.138112e-05 3.216014e-04
## t11*  0.180109967  3.838638e-03 2.974772e-02
## t12*  0.028617846  1.927628e-05 3.328356e-04
```

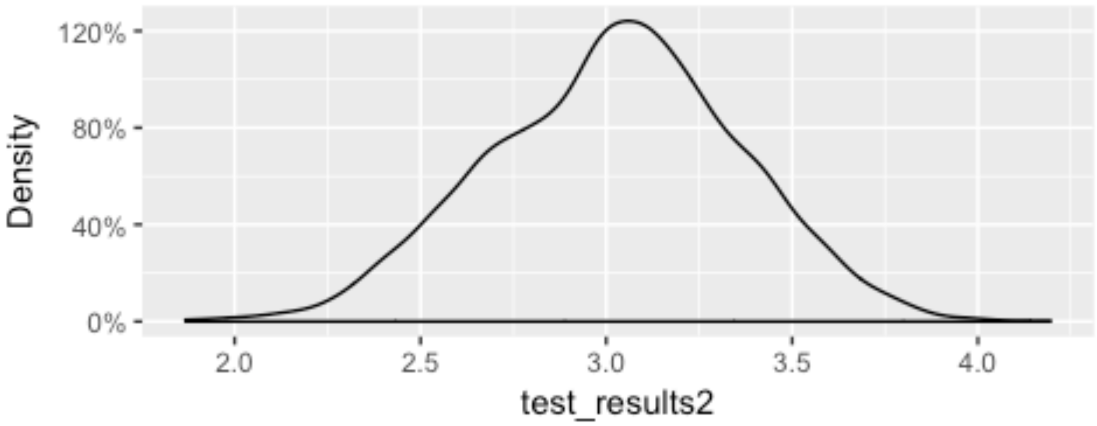
The output here are alternating parameter estimates. tw pertains to parameter estimates,tw has the standard error, and t3 contains the bootstrap standard errors.

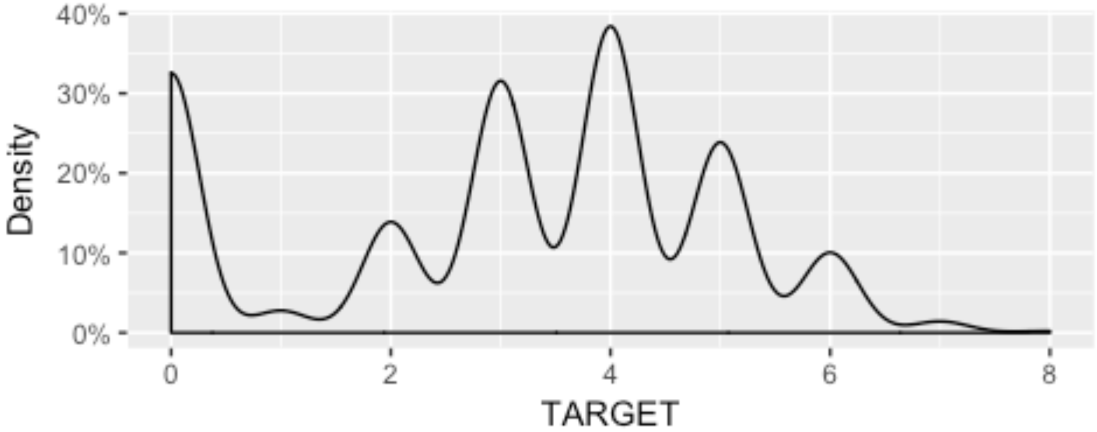
Confidence intervals

```
##                           2.5 %      97.5 %
## count_(Intercept)  1.448520589  1.62165373
## count_Alcohol      0.007116903  0.01292008
## count_AcidIndex   -0.052499493 -0.03145247
## zero_(Intercept)  -0.295658088 -0.03188978
## zero_STARS        -0.726002765 -0.59095725
## zero_LabelAppeal   0.124020018  0.23619991
```

How well does it predict values in our test data? Lets deploy our model on the evaluation data and look at some descriptives to compare to the training data.Before that, We can also partition our training data into a smaller subset and see actuals vs predicted


redicted Wine Sales vs Actual Wine Sales on Test Da
Actuals
0
2
4
6
8
2.0
2.5
3.0
3.5
4.0
Predicted





```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   2.000   3.000   3.038   4.000   8.000

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   1.868   2.789   3.033   3.024   3.252   4.196
```

Deploy to production on evaluation data and compare

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   1.838   2.720   3.030   3.014   3.292   4.175     841

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   2.000   3.000   3.029   4.000   8.000
```

It looks like the distribution of our predicted values are roughly the same as the distirbution of the actuals. We can conclude that the zero inflated poisson model is our best model to predict the number of wine sales.

```
## 
## Call:
## zeroinfl(formula = TARGET ~ Alcohol + AcidIndex | STARS + 
LabelAppeal, 
##     data = wine_training3)
## 
```

```
## Pearson residuals:
##     Min      1Q  Median      3Q     Max 
## -1.5470 -0.5324  0.1649  0.6139  2.3798 
## 
## Count model coefficients (poisson with log link):
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  1.535087   0.044167  34.756  < 2e-16 ***
## Alcohol      0.010018   0.001480   6.767 1.31e-11 ***
## AcidIndex   -0.041976   0.005369  -7.818 5.37e-15 ***
## 
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -0.16377    0.06729  -2.434   0.0149 *  
## STARS       -0.65848    0.03445 -19.114  < 2e-16 ***
## LabelAppeal  0.18011    0.02862   6.294  3.1e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Number of iterations in BFGS optimization: 12 
## Log-likelihood: -2.424e+04 on 6 Df
```

Appendix)

```
url <- 'https://raw.githubusercontent.com/vindication09/DATA-621-Week-5/master/wine-training-data.csv'
url2<-'https://raw.githubusercontent.com/vindication09/DATA-621-Week-5/master/wine-evaluation-data.csv'

wine_training <- read.csv(url, header = TRUE)
wine_evaluation <- read.csv(url2, header = TRUE)



head(wine_training,10)
wine_training2<-subset(wine_training, select=-c(INDEX))
#wine_training2<-subset(wine_training, select=-c(INDEX))
wine_evaluation2<-subset(wine_evaluation, select=-c(IN))
names(wine_training2)
str(wine_training2)
#install.packages('DataExplorer)
library(DataExplorer)
plot_str(wine_training2)
plot_missing(wine_training2)
plot_histogram(wine_training2);plot_density(wine_training2)
barplot(table(wine_training2$TARGET), ylim=c(0, 5000), xlab="Result", ylab="N", col="black",
        main="Distribution of Target(Response)")
summary(wine_training2)
summary(wine_training2$TARGET);var(wine_training2$TARGET)
#12,795
colSums(wine_training2 < 0)
```

```
#has.neg <- apply(wine_training2, 1, function(row) any(row < 0))
#which(has.neg)
apply(wine_training2,2,  function(col)cor(col, wine_training2$TARGET))
#correlation matrix and visualization
correlation_matrix <- round(cor(wine_training2),2)

# Get lower triangle of the correlation matrix
  get_lower_tri<-function(correlation_matrix){
    correlation_matrix[upper.tri(correlation_matrix)] <- NA
    return(correlation_matrix)
  }
  # Get upper triangle of the correlation matrix
  get_upper_tri <- function(correlation_matrix){
    correlation_matrix[lower.tri(correlation_matrix)]<- NA
    return(correlation_matrix)
  }

  upper_tri <- get_upper_tri(correlation_matrix)



library(reshape2)

# Melt the correlation matrix
melted_correlation_matrix <- melt(upper_tri, na.rm = TRUE)

# Heatmap
library(ggplot2)

ggheatmap <- ggplot(data = melted_correlation_matrix, aes(Var2, Var1,
fill = value))+
 geom_tile(color = "white")+
 scale_fill_gradient2(low = "blue", high = "red", mid = "white",
   midpoint = 0, limit = c(-1,1), space = "Lab",
   name="Pearson\nCorrelation") +
  theme_minimal()+
 theme(axis.text.x = element_text(angle = 45, vjust = 1,
    size = 15, hjust = 1))+
 coord_fixed()


#add nice labels
ggheatmap +
geom_text(aes(Var2, Var1, label = value), color = "black", size = 3) +
theme(
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),
  axis.text.x=element_text(size=rel(0.8), angle=90),
  axis.text.y=element_text(size=rel(0.8)),
  panel.grid.major = element_blank(),
  panel.border = element_blank(),
  panel.background = element_blank(),
  axis.ticks = element_blank(),
  legend.justification = c(1, 0),
  legend.position = c(0.6, 0.7),
  legend.direction = "horizontal")+
  guides(fill = guide_colorbar(barwiwine_training3h = 7, barheight = 1,
                title.position = "top", title.hjust = 0.5))
outlierKD<-function(wine_training2, var) {
     var_name <- eval(substitute(var),eval(wine_training2))
     na1 <- sum(is.na(var_name))
     m1 <- mean(var_name, na.rm = T)
     par(mfrow=c(2, 2), oma=c(0,0,3,0))
     boxplot(var_name, main="With outliers")
     hist(var_name, main="With outliers", xlab=NA, ylab=NA)
     outlier <- boxplot.stats(var_name)$out
     mo <- mean(outlier)
     var_name <- ifelse(var_name %in% outlier, NA, var_name)
     boxplot(var_name, main="Without outliers")
     hist(var_name, main="Without outliers", xlab=NA, ylab=NA)
     title("Outlier Check", outer=TRUE)
     na2 <- sum(is.na(var_name))
     cat("Outliers identified:", na2 - na1, "n")
     cat("Propotion (%) of outliers:", round((na2 - na1) /
sum(!is.na(var_name))*100, 1), "n")
     cat("Mean of the outliers:", round(mo, 2), "n")
     m2 <- mean(var_name, na.rm = T)
     cat("Mean without removing outliers:", round(m1, 2), "n")
     cat("Mean if we remove outliers:", round(m2, 2), "n")
     response <- readline(prompt="Do you want to remove outliers and to
replace with NA? [yes/no]: ")
     if(response == "y" | response == "yes"){
          wine_training3[as.character(substitute(var))] <-
invisible(var_name)
          assign(as.character(as.list(match.call())$wine_training2),
wine_training2, envir = .GlobalEnv)
          cat("Outliers successfully removed", "n")
          return(invisible(wine_training2))
     } else{
          cat("Nothing changed", "n")
          return(invisible(var_name))
     }
}
outlierKD(wine_training2, TARGET)
outlierKD(wine_training2, Chlorides)
outlierKD(wine_training2, Alcohol)
outlierKD(wine_training2, FixedAcidity)
outlierKD(wine_training2, FreeSulfurDioxide)
outlierKD(wine_training2, LabelAppeal)
outlierKD(wine_training2, VolatileAcidity)
outlierKD(wine_training2, TotalSulfurDioxide)
outlierKD(wine_training2, AcidIndex)
outlierKD(wine_training2, CitricAcid)
outlierKD(wine_training2, Density)
outlierKD(wine_training2, ResidualSugar)
outlierKD(wine_training2, pH)
outlierKD(wine_training2, STARS)
outlierKD(wine_training2, Sulphates)
colSums(is.na(wine_training2))
library(Hmisc)

wine_training3<-wine_training2

wine_training3$STARS<-impute(wine_training3$STARS, median)

#make an additional subset that retains the same values but simply
removes negative values (not possible)
wine_training_redux <- wine_training2[wine_training2$Alcohol >= 0 &&
wine_training2$Sulphates >= 0
                                      && wine_training2$Sulphates >= 0
                                      &&
wine_training2$TotalSulfurDioxide >= 0
                                      &&
wine_training2$FreeSulfurDioxide >= 0
                                      && wine_training2$Chlorides >= 0
                                      && wine_training2$ResidualSugar
                                      && wine_training2$CitricAcid
                                      && wine_training2$VolatileAcidity
>= 0
                                      && wine_training2$FixedAcidity >=
0,]

#wine_training_redux <- wine_training2[wine_training2$Sulphates >= 0,]
#wine_training_redux <-
wine_training2[wine_training2$TotalSulfurDioxide >= 0,]
#wine_training_redux <- wine_training2[wine_training2$FreeSulfurDioxide
>= 0, ]
#wine_training_redux <- wine_training2[wine_training2$Chlorides >= 0, ]
#wine_training_redux <- wine_training2[wine_training2$ResidualSugar >=
0,]
#wine_training_redux <- wine_training2[wine_training2$CitricAcid >= 0,]
#wine_training_redux <- wine_training2[wine_training2$VolatileAcidity
>= 0,]
#wine_training_redux <- wine_training2[wine_training2$FixedAcidity >=
0,]
summary(wine_training3$STARS);summary(wine_training2$STARS)
barplot(table(wine_training3$STARS), ylim=c(0, 7000), xlab="Rating
(post impute)", ylab="N", col="black");
```

```r
barplot(table(wine_training2$STARS), ylim=c(0, 7000), xlab="Rating (pre
impute)", ylab="N", col="black")

colSums(wine_training3<0);colSums(is.na(wine_training3))
wine_training3$Sulphates<-abs(wine_training3$Sulphates)
wine_training3$pH<-abs(wine_training3$pH)
wine_training3$ResidualSugar<-abs(wine_training3$ResidualSugar)
wine_training3$Chlorides<-abs(wine_training3$Chlorides)
wine_training3$FreeSulfurDioxide<-abs(wine_training3$FreeSulfurDioxide)
wine_training3$TotalSulfurDioxide<-
abs(wine_training3$TotalSulfurDioxide)
wine_training3$VolatileAcidity<-abs(wine_training3$VolatileAcidity)
wine_training3$Alcohol<-abs(wine_training3$ Alcohol)
wine_training3$CitricAcid<-abs(wine_training3$CitricAcid)
wine_training3$FixedAcidity<-abs(wine_training3$FixedAcidity)

wine_evaluation3<-wine_evaluation

wine_evaluation3$Sulphates<-abs(wine_evaluation3$Sulphates)
wine_evaluation3$pH<-abs(wine_evaluation3$pH)
wine_evaluation3$ResidualSugar<-abs(wine_evaluation3$ResidualSugar)
wine_evaluation3$Chlorides<-abs(wine_evaluation3$Chlorides)
wine_evaluation3$FreeSulfurDioxide<-
abs(wine_evaluation3$FreeSulfurDioxide)
wine_evaluation3$TotalSulfurDioxide<-
abs(wine_evaluation3$TotalSulfurDioxide)
wine_evaluation3$VolatileAcidity<-abs(wine_evaluation3$VolatileAcidity)
wine_evaluation3$Alcohol<-abs(wine_evaluation3$ Alcohol)
wine_evaluation3$CitricAcid<-abs(wine_evaluation3$CitricAcid)
wine_evaluation3$FixedAcidity<-abs(wine_evaluation3$FixedAcidity)
wine_training3$Sulphates<-impute(wine_training3$Sulphates, median)
wine_training3$pH<-impute(wine_training3$pH, median)
wine_training3$ResidualSugar<-impute(wine_training3$ResidualSugar,
median)
wine_training3$Chlorides<-impute(wine_training3$Chlorides, median)
wine_training3$FreeSulfurDioxide<-
impute(wine_training3$FreeSulfurDioxide, median)
wine_training3$TotalSulfurDioxide<-
impute(wine_training3$TotalSulfurDioxide, median)
wine_training3$Alcohol<-impute(wine_training3$Alcohol, median)

wine_evaluation3$Sulphates<-impute(wine_evaluation3$Sulphates, median)
wine_evaluation3$pH<-impute(wine_evaluation3$pH, median)
wine_evaluation3$ResidualSugar<-impute(wine_evaluation3$ResidualSugar,
median)
wine_evaluation3$Chlorides<-impute(wine_evaluation3$Chlorides, median)
wine_evaluation3$FreeSulfurDioxide<-
impute(wine_evaluation3$FreeSulfurDioxide, median)
wine_evaluation3$TotalSulfurDioxide<-
```

```r
impute(wine_evaluation3$TotalSulfurDioxide, median)
wine_evaluation3$Alcohol<-impute(wine_evaluation3$Alcohol, median)
#wine_evaluation3$TARGET<-impute(wine_evaluation3$Alcohol, median)



summary(wine_training3)
plot_density(wine_training3)
#testing
wine_training4<-wine_training3

wine_training4$Sulphates<-log(wine_training4$Sulphates+1)
wine_training4$pH<-log(wine_training4$pH+1)
wine_training4$ResidualSugar<-log(wine_training4$ResidualSugar+1)
wine_training4$Chlorides<-log(wine_training4$Chlorides+1)
wine_training4$FreeSulfurDioxide<-
log(wine_training4$FreeSulfurDioxide+1)
wine_training4$TotalSulfurDioxide<-
log(wine_training4$TotalSulfurDioxide+1)
wine_training4$Alcohol<-log(wine_training4$Alcohol+1)


plot_density(wine_training4);plot_density(wine_training2)

#correlation matrix and visualization
correlation_matrix <- round(cor(wine_training3),2)

# Get lower triangle of the correlation matrix
  get_lower_tri<-function(correlation_matrix){
    correlation_matrix[upper.tri(correlation_matrix)] <- NA
    return(correlation_matrix)
  }
  # Get upper triangle of the correlation matrix
  get_upper_tri <- function(correlation_matrix){
    correlation_matrix[lower.tri(correlation_matrix)]<- NA
    return(correlation_matrix)
  }

  upper_tri <- get_upper_tri(correlation_matrix)



library(reshape2)

# Melt the correlation matrix
melted_correlation_matrix <- melt(upper_tri, na.rm = TRUE)

# Heatmap
library(ggplot2)
```

```
ggheatmap <- ggplot(data = melted_correlation_matrix, aes(Var2, Var1,
fill = value))+
 geom_tile(color = "white")+
 scale_fill_gradient2(low = "blue", high = "red", mid = "white",
   midpoint = 0, limit = c(-1,1), space = "Lab",
   name="Pearson\nCorrelation") +
  theme_minimal()+
 theme(axis.text.x = element_text(angle = 45, vjust = 1,
    size = 15, hjust = 1))+
 coord_fixed()


#add nice labels
ggheatmap +
geom_text(aes(Var2, Var1, label = value), color = "black", size = 3) +
theme(
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),
  axis.text.x=element_text(size=rel(0.8), angle=90),
  axis.text.y=element_text(size=rel(0.8)),
  panel.grid.major = element_blank(),
  panel.border = element_blank(),
  panel.background = element_blank(),
  axis.ticks = element_blank(),
  legend.justification = c(1, 0),
  legend.position = c(0.6, 0.7),
  legend.direction = "horizontal")+
  guides(fill = guide_colorbar(barwiwine_training3h = 7, barheight = 1,
                title.position = "top", title.hjust = 0.5))
#correlation matrix and visualization
correlation_matrix <- round(cor(wine_training4),2)

# Get lower triangle of the correlation matrix
  get_lower_tri<-function(correlation_matrix){
    correlation_matrix[upper.tri(correlation_matrix)] <- NA
    return(correlation_matrix)
  }
  # Get upper triangle of the correlation matrix
  get_upper_tri <- function(correlation_matrix){
    correlation_matrix[lower.tri(correlation_matrix)]<- NA
    return(correlation_matrix)
  }

  upper_tri <- get_upper_tri(correlation_matrix)



library(reshape2)
```

```
# Melt the correlation matrix
melted_correlation_matrix <- melt(upper_tri, na.rm = TRUE)

# Heatmap
library(ggplot2)

ggheatmap <- ggplot(data = melted_correlation_matrix, aes(Var2, Var1,
fill = value))+
 geom_tile(color = "white")+
 scale_fill_gradient2(low = "blue", high = "red", mid = "white",
   midpoint = 0, limit = c(-1,1), space = "Lab",
   name="Pearson\nCorrelation") +
  theme_minimal()+
 theme(axis.text.x = element_text(angle = 45, vjust = 1,
    size = 15, hjust = 1))+
 coord_fixed()


#add nice labels
ggheatmap +
geom_text(aes(Var2, Var1, label = value), color = "black", size = 3) +
theme(
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),
  axis.text.x=element_text(size=rel(0.8), angle=90),
  axis.text.y=element_text(size=rel(0.8)),
  panel.grid.major = element_blank(),
  panel.border = element_blank(),
  panel.background = element_blank(),
  axis.ticks = element_blank(),
  legend.justification = c(1, 0),
  legend.position = c(0.6, 0.7),
  legend.direction = "horizontal")+
  guides(fill = guide_colorbar(barwiwine_training3h = 7, barheight = 1,
                title.position = "top", title.hjust = 0.5))
library(vcd)
library(faraway)
library(AER)
library(boot)

pmod <- glm(TARGET~., family="poisson", data=wine_training3)
summary(pmod);



#goodness of fit
anova(pmod, test="Chisq");
```

```
glm.diag.plots(pmod, glmdiag = glm.diag(pmod), subset = NULL,
               iden = FALSE, labels = NULL, ret = FALSE)

#cov.pmod <- vcovHC(pmod, type="HC0")
#std.err <- sqrt(diag(cov.pmod))
#r.est <- cbind(Estimate= coef(pmod), "Robust SE" = std.err,
#"Pr(>|z|)" = 2 * pnorm(abs(coef(pmod)/std.err), lower.tail=FALSE),
#LL = coef(pmod) - 1.96 * std.err,
#UL = coef(pmod) + 1.96 * std.err);

#r.est;

with(pmod, cbind(res.deviance = deviance, df = df.residual,p =
pchisq(deviance, df.residual, lower.tail=FALSE)));

p1<-1-(18475/22861)
p1;

pchisq(pmod$deviance, df=pmod$df.residual, lower.tail=FALSE)

#dispersion test

#deviance(pmod)/pmod$df.residual
#dispersiontest(pmod);

#halfnorm(residuals(pmod))
library(MASS)
step<-stepAIC(pmod, trace=FALSE)
step$anova

pmod2<-glm(TARGET ~ VolatileAcidity + CitricAcid + Chlorides +
FreeSulfurDioxide +
    TotalSulfurDioxide + Density + pH + Sulphates + Alcohol +
    LabelAppeal + AcidIndex + STARS, family="poisson",
data=wine_training3)

summary(pmod2);

#goodness of fit
anova(pmod2, test="Chisq");

glm.diag.plots(pmod2, glmdiag = glm.diag(pmod2), subset = NULL,
               iden = FALSE, labels = NULL, ret = FALSE)

#cov.pmod <- vcovHC(pmod, type="HC0")
#std.err <- sqrt(diag(cov.pmod))
#r.est <- cbind(Estimate= coef(pmod), "Robust SE" = std.err,
#"Pr(>|z|)" = 2 * pnorm(abs(coef(pmod)/std.err), lower.tail=FALSE),
#LL = coef(pmod) - 1.96 * std.err,
```

```r
#UL = coef(pmod) + 1.96 * std.err);

#r.est;

with(pmod2, cbind(res.deviance = deviance, df = df.residual,p =
pchisq(deviance, df.residual, lower.tail=FALSE)));

p2<-1-(18475/22861)
p2;

pchisq(pmod2$deviance, df=pmod2$df.residual, lower.tail=FALSE)

#dispersion test

#deviance(pmod)/pmod$df.residual
#dispersiontest(pmod);

#halfnorm(residuals(pmod))
pmod3<-glm(TARGET ~ CitricAcid + Chlorides + FreeSulfurDioxide +
    TotalSulfurDioxide + Density + Sulphates + Alcohol +
    LabelAppeal + AcidIndex + STARS, family="poisson",
data=wine_training3)

summary(pmod3);

#goodness of fit
anova(pmod3, test="Chisq");

glm.diag.plots(pmod3, glmdiag = glm.diag(pmod3), subset = NULL,
               iden = FALSE, labels = NULL, ret = FALSE)

#cov.pmod <- vcovHC(pmod, type="HC0")
#std.err <- sqrt(diag(cov.pmod))
#r.est <- cbind(Estimate= coef(pmod), "Robust SE" = std.err,
#"Pr(>|z|)" = 2 * pnorm(abs(coef(pmod)/std.err), lower.tail=FALSE),
#LL = coef(pmod) - 1.96 * std.err,
#UL = coef(pmod) + 1.96 * std.err);

#r.est;

with(pmod3, cbind(res.deviance = deviance, df = df.residual,p =
pchisq(deviance, df.residual, lower.tail=FALSE)));

p3<-1-(18475/22861)
p3

pchisq(pmod3$deviance, df=pmod3$df.residual, lower.tail=FALSE)
#dispersion test
```

```r
#deviance(pmod)/pmod$df.residual
#dispersiontest(pmod);

#halfnorm(residuals(pmod))
pmod4<-glm(TARGET ~ Chlorides + FreeSulfurDioxide +
    TotalSulfurDioxide + Density + Sulphates + Alcohol +
    LabelAppeal + AcidIndex + STARS, family="poisson",
data=wine_training3)

summary(pmod4);

#goodness of fit
anova(pmod4, test="Chisq");

glm.diag.plots(pmod4, glmdiag = glm.diag(pmod4), subset = NULL,
               iden = FALSE, labels = NULL, ret = FALSE)

#cov.pmod <- vcovHC(pmod, type="HC0")
#std.err <- sqrt(diag(cov.pmod))
#r.est <- cbind(Estimate= coef(pmod), "Robust SE" = std.err,
#"Pr(>|z|)" = 2 * pnorm(abs(coef(pmod)/std.err), lower.tail=FALSE),
#LL = coef(pmod) - 1.96 * std.err,
#UL = coef(pmod) + 1.96 * std.err);

#r.est;

with(pmod4, cbind(res.deviance = deviance, df = df.residual,p =
pchisq(deviance, df.residual, lower.tail=FALSE)));

exp(cbind("Odds ratio" = coef(pmod4), confint.default(pmod4, level =
0.95)));

p4<-1-(18475/22861)
p4;

plot(log(fitted(pmod4)), log((wine_training3$TARGET-fitted(pmod4))^2),
xlab=expression(hat(mu)), ylab=expression((y-hat(mu))^2))
abline(0, 1);

#goodness of fit
pchisq(pmod4$deviance, df=pmod4$df.residual, lower.tail=FALSE)

#dispersion test

#deviance(pmod)/pmod$df.residual
#dispersiontest(pmod);
```

```
#halfnorm(residuals(pmod))
library(gam)

pmod_smooth<-gam(TARGET ~ s(LabelAppeal)+s(AcidIndex)+s(STARS) ,
family=poisson(link=log),  data=wine_training3)

#pmod_smooth<-gam(TARGET ~ LabelAppeal+AcidIndex+STARS ,
family=poisson(link=log),  data=wine_training3)

summary(pmod_smooth);

#goodness of fit
anova(pmod_smooth, test="Chisq");

glm.diag.plots(pmod_smooth, glmdiag = glm.diag(pmod_smooth), subset =
NULL,
               iden = FALSE, labels = NULL, ret = FALSE)

#cov.pmod <- vcovHC(pmod, type="HC0")
#std.err <- sqrt(diag(cov.pmod))
#r.est <- cbind(Estimate= coef(pmod), "Robust SE" = std.err,
#"Pr(>|z|)" = 2 * pnorm(abs(coef(pmod)/std.err), lower.tail=FALSE),
#LL = coef(pmod) - 1.96 * std.err,
#UL = coef(pmod) + 1.96 * std.err);

#r.est;

with(pmod_smooth, cbind(res.deviance = deviance, df = df.residual,p =
pchisq(deviance, df.residual, lower.tail=FALSE)));

exp(cbind("Odds ratio" = coef(pmod_smooth),
confint.default(pmod_smooth, level = 0.95)));

p_smooth<-1-(17735.95/22860.89)
p_smooth;

plot(log(fitted(pmod_smooth)), log((wine_training3$TARGET-
fitted(pmod_smooth))^2), xlab=expression(hat(mu)), ylab=expression((y-
hat(mu))^2))
abline(0, 1);

#goodness of fit
# 1-pchisq(summary(pmod_smooth)$deviance,
summary(pmod_smooth)$df.residual)
pchisq(pmod_smooth$deviance, df=pmod_smooth$df.residual,
lower.tail=FALSE)

#dispersion test
```

```
#deviance(pmod)/pmod$df.residual
#dispersiontest(pmod);

#halfnorm(residuals(pmod))
#plot(p_smooth, pages = 1, scheme = 1, all.terms = TRUE, seWithMean =
TRUE)
library(MASS)
nmod1 <- glm.nb(TARGET ~ FixedAcidity + VolatileAcidity + CitricAcid +
ResidualSugar +
    Chlorides + FreeSulfurDioxide + TotalSulfurDioxide + Density +
    pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS
 ,  data=wine_training3)

summary(nmod1);

#goodness of fit
anova(nmod1, test="Chisq");

glm.diag.plots(nmod1, glmdiag = glm.diag(nmod1), subset = NULL,
               iden = FALSE, labels = NULL, ret = FALSE)

#cov.pmod <- vcovHC(pmod, type="HC0")
#std.err <- sqrt(diag(cov.pmod))
#r.est <- cbind(Estimate= coef(pmod), "Robust SE" = std.err,
#"Pr(>|z|)" = 2 * pnorm(abs(coef(pmod)/std.err), lower.tail=FALSE),
#LL = coef(pmod) - 1.96 * std.err,
#UL = coef(pmod) + 1.96 * std.err);

#r.est;

with(nmod1, cbind(res.deviance = deviance, df = df.residual,p =
pchisq(deviance, df.residual, lower.tail=FALSE)));

exp(cbind("Odds ratio" = coef(nmod1), confint.default(nmod1, level =
0.95)));

#n<-1-(18474/22860)
#n;

plot(log(fitted(nmod1)), log((wine_training3$TARGET-fitted(nmod1))^2),
xlab=expression(hat(mu)), ylab=expression((y-hat(mu))^2))
abline(0, 1);

#goodness of fit
pchisq(nmod1$deviance, df=nmod1$df.residual, lower.tail=FALSE)

nmod2 <- glm.nb(TARGET ~ VolatileAcidity  +
    Chlorides + TotalSulfurDioxide + Density + Alcohol + LabelAppeal +
AcidIndex + STARS
```

```
 ,  data=wine_training3)

summary(nmod2);

#goodness of fit
anova(nmod2, test="Chisq");

glm.diag.plots(nmod2, glmdiag = glm.diag(nmod2), subset = NULL,
               iden = FALSE, labels = NULL, ret = FALSE)

#cov.pmod <- vcovHC(pmod, type="HC0")
#std.err <- sqrt(diag(cov.pmod))
#r.est <- cbind(Estimate= coef(pmod), "Robust SE" = std.err,
#"Pr(>|z|)" = 2 * pnorm(abs(coef(pmod)/std.err), lower.tail=FALSE),
#LL = coef(pmod) - 1.96 * std.err,
#UL = coef(pmod) + 1.96 * std.err);

#r.est;

with(nmod2, cbind(res.deviance = deviance, df = df.residual,p =
pchisq(deviance, df.residual, lower.tail=FALSE)));

exp(cbind("Odds ratio" = coef(nmod2), confint.default(nmod2, level =
0.95)));

#n2<-1-(18500/22860)
#n2;

plot(log(fitted(nmod2)), log((wine_training3$TARGET-fitted(nmod2))^2),
xlab=expression(hat(mu)), ylab=expression((y-hat(mu))^2))
abline(0, 1);

#goodness of fit
pchisq(nmod2$deviance, df=nmod2$df.residual, lower.tail=FALSE)
nmod4 <- glm.nb(TARGET ~ Alcohol+LabelAppeal + AcidIndex + STARS
 ,  data=wine_training3)

summary(nmod4);

#goodness of fit
anova(nmod4, test="Chisq");

glm.diag.plots(nmod4, glmdiag = glm.diag(nmod4), subset = NULL,
               iden = FALSE, labels = NULL, ret = FALSE)

#cov.pmod <- vcovHC(pmod, type="HC0")
#std.err <- sqrt(diag(cov.pmod))
#r.est <- cbind(Estimate= coef(pmod), "Robust SE" = std.err,
#"Pr(>|z|)" = 2 * pnorm(abs(coef(pmod)/std.err), lower.tail=FALSE),
```

```r
#LL = coef(pmod) - 1.96 * std.err,
#UL = coef(pmod) + 1.96 * std.err);

#r.est;

with(nmod4, cbind(res.deviance = deviance, df = df.residual,p =
pchisq(deviance, df.residual, lower.tail=FALSE)));

exp(cbind("Odds ratio" = coef(nmod4), confint.default(nmod4, level =
0.95)));

#n2<-1-(18566/22860)
#n2;

plot(log(fitted(nmod4)), log((wine_training3$TARGET-fitted(nmod4))^2),
xlab=expression(hat(mu)), ylab=expression((y-hat(mu))^2))
abline(0, 1);

plot(predict(nmod4),wine_training3$TARGET,
      xlab="predicted",ylab="actual")
 abline(a=0,b=1);

 #goodness of fit
pchisq(nmod4$deviance, df=nmod4$df.residual, lower.tail=FALSE)
library(olsrr)

lmod1 <- lm(TARGET ~ FixedAcidity + VolatileAcidity + CitricAcid +
ResidualSugar +
    Chlorides + FreeSulfurDioxide + TotalSulfurDioxide + Density +
    pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS
 ,  data=wine_training3);

summary(lmod1);

par(mfrow=c(2,2))
plot(lmod1)
hist(resid(lmod1), main="Histogram of Residuals");
ols_test_breusch_pagan(lmod1);
vif(lmod1);

plot(predict(lmod1),wine_training3$TARGET,
      xlab="predicted",ylab="actual")
 abline(a=0,b=1)
lstep<-stepAIC(lmod1, trace=FALSE)
lstep$anova
lmod2 <- lm(TARGET ~ VolatileAcidity + CitricAcid +
    Chlorides + FreeSulfurDioxide + TotalSulfurDioxide + Density +
    pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS
 ,  data=wine_training3);
```

```
summary(lmod2);

par(mfrow=c(2,2))
plot(lmod2)
hist(resid(lmod2), main="Histogram of Residuals");
ols_test_breusch_pagan(lmod2);
vif(lmod2);

plot(predict(lmod2),wine_training3$TARGET,
      xlab="predicted",ylab="actual")
 abline(a=0,b=1)
library(ISLR)

lmod3<-lm(TARGET ~ VolatileAcidity + CitricAcid +
    Chlorides + FreeSulfurDioxide + TotalSulfurDioxide + Density +
    pH + Sulphates + Alcohol + s(LabelAppeal) + AcidIndex + s(STARS)
 ,  data=wine_training3)

summary(lmod3);

par(mfrow=c(2,2))
plot(lmod3)
hist(resid(lmod3), main="Histogram of Residuals");
ols_test_breusch_pagan(lmod2);
vif(lmod3);

plot(predict(lmod3),wine_training3$TARGET,
      xlab="predicted",ylab="actual")
 abline(a=0,b=1);

 #goodness of fit
 #pchisq(summary(pmod7)$deviance,
  #         summary(pmod7)$df.residual
   #        )

colSums(wine_training3==0)
require(ggplot2)
require(pscl)
require(MASS)
require(boot)

pmod7 <- zeroinfl(TARGET ~   Alcohol  + AcidIndex | STARS+LabelAppeal,
  data = wine_training3)
summary(pmod7);


#glm.diag.plots(nmod3, glmdiag = glm.diag(nmod3), subset = NULL,
 #              iden = FALSE, labels = NULL, ret = FALSE)
```

```
#cov.pmod <- vcovHC(pmod, type="HC0")
#std.err <- sqrt(diag(cov.pmod))
#r.est <- cbind(Estimate= coef(pmod), "Robust SE" = std.err,
#"Pr(>|z|)" = 2 * pnorm(abs(coef(pmod)/std.err), lower.tail=FALSE),
#LL = coef(pmod) - 1.96 * std.err,
#UL = coef(pmod) + 1.96 * std.err);

#r.est;

#with(nmod3, cbind(res.deviance = deviance, df = df.residual,p =
pchisq(deviance, df.residual, lower.tail=FALSE)));

#exp(cbind("Odds ratio" = coef(pmod7), confint.default(pmod7, level =
0.95)));

#n2<-1-(18500/22860)
#n2;

plot(log(fitted(pmod7)), log((wine_training3$TARGET-fitted(pmod7))^2),
xlab=expression(hat(mu)), ylab=expression((y-hat(mu))^2))
abline(0, 1)

vuong(pmod7, pmod4);

plot(predict(pmod7),wine_training3$TARGET,
      xlab="predicted",ylab="actual")
 abline(a=0,b=1);

#goodness of fit
# pchisq(summary(pmod7)$deviance,
 #          summary(pmod7)$df.residual
  #         )
pnull <- update(pmod7, . ~ 1)

pchisq(2 * (logLik(pmod7) - logLik(pnull)), df = 3, lower.tail = FALSE)
library(caret)
Train <- createDataPartition(wine_training3$TARGET, p=0.7, list=FALSE)
train <- wine_training3[Train, ]
test <- wine_training3[-Train, ]
dput(coef(pmod7, "count"));dput(coef(pmod7, "zero"))
f <- function(data, i)
  {
  require(pscl)
  m <- zeroinfl(TARGET ~   Alcohol  + AcidIndex | STARS+LabelAppeal,
data = data[i, ],
    start = list(count = c(1.5350871583595, 0.0100184890625652, -
0.0419759809109108
), zero = c(-0.163773933205731, -0.658480007306658, 0.180109965808006
)))
  as.vector(t(do.call(rbind, coef(summary(m)))[, 1:2]))
 }

set.seed(10)
res <- boot(wine_training3, f, R = 100, parallel = "snow", ncpus = 4)

## print results
res
confint(pmod7)
#gather predicted
test_results2<-predict(pmod7, newdata=test, type = "response")
target_pred<-data.frame(test_results2)

actuals<-subset(test,select=c(TARGET))

#plot
results<-data.frame(target_pred, actuals)

xyplot(TARGET ~ test_results2, data = results,
  xlab = "Predicted ",
  ylab = "Actuals",
  main = "Predicted Wine Sales vs Actual Wine Sales on Test Data");

plot_density(results);

summary(results$TARGET);summary(results$test_results2)
test_results<-predict(pmod7, newdata=wine_evaluation3, type =
"response")
test.df<-data.frame(test_results)
summary(test_results);summary(wine_training3$TARGET)

summary(pmod7)
#read in data
url <- 'https://raw.githubusercontent.com/vindication09/DATA-621-Week-
5/master/wine-training-data.csv'
url2<-'https://raw.githubusercontent.com/vindication09/DATA-621-Week-
5/master/wine-evaluation-data.csv'

wine_training <- read.csv(url, header = TRUE)
wine_evaluation <- read.csv(url2, header = TRUE)


head(wine_training,10)

wine_training2<-subset(wine_training, select=-c(INDEX))
#wine_training2<-subset(wine_training, select=-c(INDEX))
wine_evaluation2<-subset(wine_evaluation, select=-c(IN))
names(wine_training2)
```

```
str(wine_training2)

#eda
#install.packages('DataExplorer)
library(DataExplorer)
plot_str(wine_training2)
plot_missing(wine_training2)
plot_histogram(wine_training2);plot_density(wine_training2)


barplot(table(wine_training2$TARGET), ylim=c(0, 5000), xlab="Result",
ylab="N", col="black",
        main="Distribution of Target(Response)")

summary(wine_training2)

summary(wine_training2$TARGET);var(wine_training2$TARGET)

#12,795
colSums(wine_training2 < 0)
#has.neg <- apply(wine_training2, 1, function(row) any(row < 0))
#which(has.neg)

apply(wine_training2,2,  function(col)cor(col, wine_training2$TARGET))


#correlation matrix and visualization
correlation_matrix <- round(cor(wine_training2),2)

# Get lower triangle of the correlation matrix
  get_lower_tri<-function(correlation_matrix){
    correlation_matrix[upper.tri(correlation_matrix)] <- NA
    return(correlation_matrix)
  }
  # Get upper triangle of the correlation matrix
  get_upper_tri <- function(correlation_matrix){
    correlation_matrix[lower.tri(correlation_matrix)]<- NA
    return(correlation_matrix)
  }

  upper_tri <- get_upper_tri(correlation_matrix)



library(reshape2)

# Melt the correlation matrix
melted_correlation_matrix <- melt(upper_tri, na.rm = TRUE)
```

```
# Heatmap
library(ggplot2)

ggheatmap <- ggplot(data = melted_correlation_matrix, aes(Var2, Var1,
fill = value))+
 geom_tile(color = "white")+
 scale_fill_gradient2(low = "blue", high = "red", mid = "white",
   midpoint = 0, limit = c(-1,1), space = "Lab",
   name="Pearson\nCorrelation") +
  theme_minimal()+
 theme(axis.text.x = element_text(angle = 45, vjust = 1,
    size = 15, hjust = 1))+
 coord_fixed()


#add nice labels
ggheatmap +
geom_text(aes(Var2, Var1, label = value), color = "black", size = 3) +
theme(
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),
  axis.text.x=element_text(size=rel(0.8), angle=90),
  axis.text.y=element_text(size=rel(0.8)),
  panel.grid.major = element_blank(),
  panel.border = element_blank(),
  panel.background = element_blank(),
  axis.ticks = element_blank(),
  legend.justification = c(1, 0),
  legend.position = c(0.6, 0.7),
  legend.direction = "horizontal")+
  guides(fill = guide_colorbar(barwiwine_training3h = 7, barheight = 1,
                title.position = "top", title.hjust = 0.5))




  outlierKD<-function(wine_training2, var) {
     var_name <- eval(substitute(var),eval(wine_training3))
     na1 <- sum(is.na(var_name))
     m1 <- mean(var_name, na.rm = T)
     par(mfrow=c(2, 2), oma=c(0,0,3,0))
     boxplot(var_name, main="With outliers")
     hist(var_name, main="With outliers", xlab=NA, ylab=NA)
     outlier <- boxplot.stats(var_name)$out
     mo <- mean(outlier)
     var_name <- ifelse(var_name %in% outlier, NA, var_name)
     boxplot(var_name, main="Without outliers")
     hist(var_name, main="Without outliers", xlab=NA, ylab=NA)
     title("Outlier Check", outer=TRUE)
     na2 <- sum(is.na(var_name))
     cat("Outliers identified:", na2 - na1, "n")
     cat("Propotion (%) of outliers:", round((na2 - na1) / 
sum(!is.na(var_name))*100, 1), "n")
     cat("Mean of the outliers:", round(mo, 2), "n")
     m2 <- mean(var_name, na.rm = T)
     cat("Mean without removing outliers:", round(m1, 2), "n")
     cat("Mean if we remove outliers:", round(m2, 2), "n")
     response <- readline(prompt="Do you want to remove outliers and to 
replace with NA? [yes/no]: ")
     if(response == "y" | response == "yes"){
          wine_training3[as.character(substitute(var))] <- 
invisible(var_name)
          assign(as.character(as.list(match.call())$wine_training2), 
wine_training2, envir = .GlobalEnv)
          cat("Outliers successfully removed", "n")
          return(invisible(wine_training2))
     } else{
          cat("Nothing changed", "n")
          return(invisible(var_name))
     }
}
```

```
outlierKD(wine_training2, TARGET)
```

```
outlierKD(wine_training2, Chlorides)
```

```
outlierKD(wine_training2, Alcohol)
```

```
outlierKD(wine_training2, FixedAcidity)
```

```
outlierKD(wine_training2, FreeSulfurDioxide)
outlierKD(wine_training2, LabelAppeal)

outlierKD(wine_training2, VolatileAcidity)

outlierKD(wine_training2, TotalSulfurDioxide)


outlierKD(wine_training2, AcidIndex)

outlierKD(wine_training2, CitricAcid)


outlierKD(wine_training2, Density)

outlierKD(wine_training2, ResidualSugar)

outlierKD(wine_training2, pH)


outlierKD(wine_training2, STARS)

outlierKD(wine_training2, Sulphates)
```

```
#Data prep

library(Hmisc)

wine_training3<-wine_training2

wine_training3$STARS<-impute(wine_training3$STARS, median)

#make an additional subset that retains the same values but simply
removes negative values (not possible)
wine_training_redux <- wine_training2[wine_training2$Alcohol >= 0 &&
wine_training2$Sulphates >= 0
                                      && wine_training2$Sulphates >= 0
                                      &&
wine_training2$TotalSulfurDioxide >= 0
                                      &&
wine_training2$FreeSulfurDioxide >= 0
                                      && wine_training2$Chlorides >= 0
                                      && wine_training2$ResidualSugar
                                      && wine_training2$CitricAcid
                                      && wine_training2$VolatileAcidity
>= 0
                                      && wine_training2$FixedAcidity >=
0,]

#wine_training_redux <- wine_training2[wine_training2$Sulphates >= 0,]
#wine_training_redux <-
wine_training2[wine_training2$TotalSulfurDioxide >= 0,]
#wine_training_redux <- wine_training2[wine_training2$FreeSulfurDioxide
>= 0, ]
#wine_training_redux <- wine_training2[wine_training2$Chlorides >= 0, ]
#wine_training_redux <- wine_training2[wine_training2$ResidualSugar >=
0,]
#wine_training_redux <- wine_training2[wine_training2$CitricAcid >= 0,]
#wine_training_redux <- wine_training2[wine_training2$VolatileAcidity
>= 0,]
#wine_training_redux <- wine_training2[wine_training2$FixedAcidity >=
0,]


barplot(table(wine_training3$STARS), ylim=c(0, 7000), xlab="Rating
(post impute)", ylab="N", col="black");
barplot(table(wine_training2$STARS), ylim=c(0, 7000), xlab="Rating (pre
impute)", ylab="N", col="black")




colSums(wine_training3<0);colSums(is.na(wine_training3))
```

```
wine_training3$Sulphates<-abs(wine_training3$Sulphates)
wine_training3$pH<-abs(wine_training3$pH)
wine_training3$ResidualSugar<-abs(wine_training3$ResidualSugar)
wine_training3$Chlorides<-abs(wine_training3$Chlorides)
wine_training3$FreeSulfurDioxide<-abs(wine_training3$FreeSulfurDioxide)
wine_training3$TotalSulfurDioxide<-
abs(wine_training3$TotalSulfurDioxide)
wine_training3$VolatileAcidity<-abs(wine_training3$VolatileAcidity)
wine_training3$Alcohol<-abs(wine_training3$ Alcohol)
wine_training3$CitricAcid<-abs(wine_training3$CitricAcid)
wine_training3$FixedAcidity<-abs(wine_training3$FixedAcidity)

wine_evaluation3<-wine_evaluation

wine_evaluation3$Sulphates<-abs(wine_evaluation3$Sulphates)
wine_evaluation3$pH<-abs(wine_evaluation3$pH)
wine_evaluation3$ResidualSugar<-abs(wine_evaluation3$ResidualSugar)
wine_evaluation3$Chlorides<-abs(wine_evaluation3$Chlorides)
wine_evaluation3$FreeSulfurDioxide<-
abs(wine_evaluation3$FreeSulfurDioxide)
wine_evaluation3$TotalSulfurDioxide<-
abs(wine_evaluation3$TotalSulfurDioxide)
wine_evaluation3$VolatileAcidity<-abs(wine_evaluation3$VolatileAcidity)
wine_evaluation3$Alcohol<-abs(wine_evaluation3$ Alcohol)
wine_evaluation3$CitricAcid<-abs(wine_evaluation3$CitricAcid)
wine_evaluation3$FixedAcidity<-abs(wine_evaluation3$FixedAcidity)



wine_training3$Sulphates<-impute(wine_training3$Sulphates, median)
wine_training3$pH<-impute(wine_training3$pH, median)
wine_training3$ResidualSugar<-impute(wine_training3$ResidualSugar,
median)
wine_training3$Chlorides<-impute(wine_training3$Chlorides, median)
wine_training3$FreeSulfurDioxide<-
impute(wine_training3$FreeSulfurDioxide, median)
wine_training3$TotalSulfurDioxide<-
impute(wine_training3$TotalSulfurDioxide, median)
wine_training3$Alcohol<-impute(wine_training3$Alcohol, median)

wine_evaluation3$Sulphates<-impute(wine_evaluation3$Sulphates, median)
wine_evaluation3$pH<-impute(wine_evaluation3$pH, median)
wine_evaluation3$ResidualSugar<-impute(wine_evaluation3$ResidualSugar,
median)
wine_evaluation3$Chlorides<-impute(wine_evaluation3$Chlorides, median)
wine_evaluation3$FreeSulfurDioxide<-
```

```r
impute(wine_evaluation3$FreeSulfurDioxide, median)
wine_evaluation3$TotalSulfurDioxide<-
impute(wine_evaluation3$TotalSulfurDioxide, median)
wine_evaluation3$Alcohol<-impute(wine_evaluation3$Alcohol, median)
#wine_evaluation3$TARGET<-impute(wine_evaluation3$Alcohol, median)


wine_training_redux$Sulphates<-impute(wine_training_redux$Sulphates,
median)
wine_training_redux$pH<-impute(wine_training_redux$pH, median)
wine_training_redux$ResidualSugar<-
impute(wine_training_redux$ResidualSugar, median)
wine_training_redux$Chlorides<-impute(wine_training_redux$Chlorides,
median)
wine_training_redux$FreeSulfurDioxide<-
impute(wine_training_redux$FreeSulfurDioxide, median)
wine_training_redux$TotalSulfurDioxide<-
impute(wine_training_redux$TotalSulfurDioxide, median)
wine_training_redux$Alcohol<-impute(wine_training_redux$Alcohol,
median)

summary(wine_training3)



#Modeling
library(vcd)
library(faraway)
library(AER)
library(boot)

pmod <- glm(TARGET~., family="poisson", data=wine_training3)
summary(pmod);



#goodness of fit
anova(pmod, test="Chisq");

glm.diag.plots(pmod, glmdiag = glm.diag(pmod), subset = NULL,
               iden = FALSE, labels = NULL, ret = FALSE)

#cov.pmod <- vcovHC(pmod, type="HC0")
#std.err <- sqrt(diag(cov.pmod))
#r.est <- cbind(Estimate= coef(pmod), "Robust SE" = std.err,
#"Pr(>|z|)" = 2 * pnorm(abs(coef(pmod)/std.err), lower.tail=FALSE),
#LL = coef(pmod) - 1.96 * std.err,
#UL = coef(pmod) + 1.96 * std.err);
```

```
#r.est;

with(pmod, cbind(res.deviance = deviance, df = df.residual,p =
pchisq(deviance, df.residual, lower.tail=FALSE)));

p1<-1-(18475/22861)
p1;

pchisq(pmod$deviance, df=pmod$df.residual, lower.tail=FALSE)

#dispersion test

#deviance(pmod)/pmod$df.residual
#dispersiontest(pmod);

#halfnorm(residuals(pmod))


library(MASS)
step<-stepAIC(pmod, trace=FALSE)
step$anova


pmod2<-glm(TARGET ~ VolatileAcidity + CitricAcid + Chlorides +
FreeSulfurDioxide +
    TotalSulfurDioxide + Density + pH + Sulphates + Alcohol +
    LabelAppeal + AcidIndex + STARS, family="poisson",
data=wine_training3)

summary(pmod2);

#goodness of fit
anova(pmod2, test="Chisq");

glm.diag.plots(pmod2, glmdiag = glm.diag(pmod2), subset = NULL,
               iden = FALSE, labels = NULL, ret = FALSE)

#cov.pmod <- vcovHC(pmod, type="HC0")
#std.err <- sqrt(diag(cov.pmod))
#r.est <- cbind(Estimate= coef(pmod), "Robust SE" = std.err,
#"Pr(>|z|)" = 2 * pnorm(abs(coef(pmod)/std.err), lower.tail=FALSE),
#LL = coef(pmod) - 1.96 * std.err,
#UL = coef(pmod) + 1.96 * std.err);

#r.est;

with(pmod2, cbind(res.deviance = deviance, df = df.residual,p =
pchisq(deviance, df.residual, lower.tail=FALSE)));
```

```
p2<-1-(18475/22861)
p2;

pchisq(pmod2$deviance, df=pmod2$df.residual, lower.tail=FALSE)

#dispersion test

#deviance(pmod)/pmod$df.residual
#dispersiontest(pmod);

#halfnorm(residuals(pmod))


pmod3<-glm(TARGET ~ CitricAcid + Chlorides + FreeSulfurDioxide +
    TotalSulfurDioxide + Density + Sulphates + Alcohol +
    LabelAppeal + AcidIndex + STARS, family="poisson",
data=wine_training3)

summary(pmod3);

#goodness of fit
anova(pmod3, test="Chisq");

glm.diag.plots(pmod3, glmdiag = glm.diag(pmod3), subset = NULL,
               iden = FALSE, labels = NULL, ret = FALSE)

#cov.pmod <- vcovHC(pmod, type="HC0")
#std.err <- sqrt(diag(cov.pmod))
#r.est <- cbind(Estimate= coef(pmod), "Robust SE" = std.err,
#"Pr(>|z|)" = 2 * pnorm(abs(coef(pmod)/std.err), lower.tail=FALSE),
#LL = coef(pmod) - 1.96 * std.err,
#UL = coef(pmod) + 1.96 * std.err);

#r.est;

with(pmod3, cbind(res.deviance = deviance, df = df.residual,p =
pchisq(deviance, df.residual, lower.tail=FALSE)));

p3<-1-(18475/22861)
p3

pchisq(pmod3$deviance, df=pmod3$df.residual, lower.tail=FALSE)
#dispersion test



pmod4<-glm(TARGET ~ Chlorides + FreeSulfurDioxide +
    TotalSulfurDioxide + Density + Sulphates + Alcohol + 
    LabelAppeal + AcidIndex + STARS, family="poisson", 
data=wine_training4)

summary(pmod4);

#goodness of fit
anova(pmod4, test="Chisq");

glm.diag.plots(pmod4, glmdiag = glm.diag(pmod4), subset = NULL,
               iden = FALSE, labels = NULL, ret = FALSE)

#cov.pmod <- vcovHC(pmod, type="HC0")
#std.err <- sqrt(diag(cov.pmod))
#r.est <- cbind(Estimate= coef(pmod), "Robust SE" = std.err,
#"Pr(>|z|)" = 2 * pnorm(abs(coef(pmod)/std.err), lower.tail=FALSE),
#LL = coef(pmod) - 1.96 * std.err,
#UL = coef(pmod) + 1.96 * std.err);

#r.est;

with(pmod4, cbind(res.deviance = deviance, df = df.residual,p = 
pchisq(deviance, df.residual, lower.tail=FALSE)));

exp(cbind("Odds ratio" = coef(pmod4), confint.default(pmod4, level = 
0.95)));

p4<-1-(18475/22861)
p4;

plot(log(fitted(pmod4)), log((wine_training4$TARGET-fitted(pmod4))^2), 
xlab=expression(hat(mu)), ylab=expression((y-hat(mu))^2))
abline(0, 1);

#goodness of fit
pchisq(pmod4$deviance, df=pmod4$df.residual, lower.tail=FALSE)

#dispersion test

#deviance(pmod)/pmod$df.residual
#dispersiontest(pmod);

#halfnorm(residuals(pmod))


library(gam)

pmod_smooth<-gam(TARGET ~ s(LabelAppeal)+s(AcidIndex)+s(STARS) ,
family=poisson(link=log),  data=wine_training3)

#pmod_smooth<-gam(TARGET ~ LabelAppeal+AcidIndex+STARS ,
family=poisson(link=log),  data=wine_training3)

summary(pmod_smooth);

#goodness of fit
anova(pmod_smooth, test="Chisq");

glm.diag.plots(pmod_smooth, glmdiag = glm.diag(pmod_smooth), subset =
NULL,
               iden = FALSE, labels = NULL, ret = FALSE)

#cov.pmod <- vcovHC(pmod, type="HC0")
#std.err <- sqrt(diag(cov.pmod))
#r.est <- cbind(Estimate= coef(pmod), "Robust SE" = std.err,
#"Pr(>|z|)" = 2 * pnorm(abs(coef(pmod)/std.err), lower.tail=FALSE),
#LL = coef(pmod) - 1.96 * std.err,
#UL = coef(pmod) + 1.96 * std.err);

#r.est;

with(pmod_smooth, cbind(res.deviance = deviance, df = df.residual,p =
pchisq(deviance, df.residual, lower.tail=FALSE)));

exp(cbind("Odds ratio" = coef(pmod_smooth),
confint.default(pmod_smooth, level = 0.95)));

p_smooth<-1-(17735.95/22860.89)
p_smooth;

plot(log(fitted(pmod_smooth)), log((wine_training3$TARGET-
fitted(pmod_smooth))^2), xlab=expression(hat(mu)), ylab=expression((y-
hat(mu))^2))
abline(0, 1);

#goodness of fit
# 1-pchisq(summary(pmod_smooth)$deviance,
summary(pmod_smooth)$df.residual)
pchisq(pmod_smooth$deviance, df=pmod_smooth$df.residual,
lower.tail=FALSE)

#dispersion test

#deviance(pmod)/pmod$df.residual
#dispersiontest(pmod);

#halfnorm(residuals(pmod))
```

```r
#plot(p_smooth, pages = 1, scheme = 1, all.terms = TRUE, seWithMean =
TRUE)


library(MASS)
nmod1 <- glm.nb(TARGET ~ FixedAcidity + VolatileAcidity + CitricAcid +
ResidualSugar +
    Chlorides + FreeSulfurDioxide + TotalSulfurDioxide + Density +
    pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS
 ,  data=wine_training3)

summary(nmod1);

#goodness of fit
anova(nmod1, test="Chisq");

glm.diag.plots(nmod1, glmdiag = glm.diag(nmod1), subset = NULL,
               iden = FALSE, labels = NULL, ret = FALSE)

#cov.pmod <- vcovHC(pmod, type="HC0")
#std.err <- sqrt(diag(cov.pmod))
#r.est <- cbind(Estimate= coef(pmod), "Robust SE" = std.err,
#"Pr(>|z|)" = 2 * pnorm(abs(coef(pmod)/std.err), lower.tail=FALSE),
#LL = coef(pmod) - 1.96 * std.err,
#UL = coef(pmod) + 1.96 * std.err);

#r.est;

with(nmod1, cbind(res.deviance = deviance, df = df.residual,p =
pchisq(deviance, df.residual, lower.tail=FALSE)));

exp(cbind("Odds ratio" = coef(nmod1), confint.default(nmod1, level =
0.95)));

#n<-1-(18474/22860)
#n;

plot(log(fitted(nmod1)), log((wine_training3$TARGET-fitted(nmod1))^2),
xlab=expression(hat(mu)), ylab=expression((y-hat(mu))^2))
abline(0, 1);

#goodness of fit
pchisq(nmod1$deviance, df=nmod1$df.residual, lower.tail=FALSE)


nmod2 <- glm.nb(TARGET ~ VolatileAcidity  +
    Chlorides + TotalSulfurDioxide + Density + Alcohol + LabelAppeal +
AcidIndex + STARS
 ,  data=wine_training3)
```

```
summary(nmod2);

#goodness of fit
anova(nmod2, test="Chisq");

glm.diag.plots(nmod2, glmdiag = glm.diag(nmod2), subset = NULL,
               iden = FALSE, labels = NULL, ret = FALSE)

#cov.pmod <- vcovHC(pmod, type="HC0")
#std.err <- sqrt(diag(cov.pmod))
#r.est <- cbind(Estimate= coef(pmod), "Robust SE" = std.err,
#"Pr(>|z|)" = 2 * pnorm(abs(coef(pmod)/std.err), lower.tail=FALSE),
#LL = coef(pmod) - 1.96 * std.err,
#UL = coef(pmod) + 1.96 * std.err);

#r.est;

with(nmod2, cbind(res.deviance = deviance, df = df.residual,p =
pchisq(deviance, df.residual, lower.tail=FALSE)));

exp(cbind("Odds ratio" = coef(nmod2), confint.default(nmod2, level =
0.95)));

#n2<-1-(18500/22860)
#n2;

plot(log(fitted(nmod2)), log((wine_training3$TARGET-fitted(nmod2))^2),
xlab=expression(hat(mu)), ylab=expression((y-hat(mu))^2))
abline(0, 1);

#goodness of fit
pchisq(nmod2$deviance, df=nmod2$df.residual, lower.tail=FALSE)





nmod4 <- glm.nb(TARGET ~ Alcohol+LabelAppeal + AcidIndex + STARS
 ,  data=wine_training3)

summary(nmod4);

#goodness of fit
anova(nmod4, test="Chisq");

glm.diag.plots(nmod4, glmdiag = glm.diag(nmod4), subset = NULL,
               iden = FALSE, labels = NULL, ret = FALSE)
```

```
#cov.pmod <- vcovHC(pmod, type="HC0")
#std.err <- sqrt(diag(cov.pmod))
#r.est <- cbind(Estimate= coef(pmod), "Robust SE" = std.err,
#"Pr(>|z|)" = 2 * pnorm(abs(coef(pmod)/std.err), lower.tail=FALSE),
#LL = coef(pmod) - 1.96 * std.err,
#UL = coef(pmod) + 1.96 * std.err);

#r.est;

with(nmod4, cbind(res.deviance = deviance, df = df.residual,p =
pchisq(deviance, df.residual, lower.tail=FALSE)));

exp(cbind("Odds ratio" = coef(nmod4), confint.default(nmod4, level =
0.95)));

#n2<-1-(18566/22860)
#n2;

plot(log(fitted(nmod4)), log((wine_training3$TARGET-fitted(nmod4))^2),
xlab=expression(hat(mu)), ylab=expression((y-hat(mu))^2))
abline(0, 1);

plot(predict(nmod4),wine_training3$TARGET,
      xlab="predicted",ylab="actual")
 abline(a=0,b=1);

 #goodness of fit
pchisq(nmod4$deviance, df=nmod4$df.residual, lower.tail=FALSE)



library(olsrr)

lmod1 <- lm(TARGET ~ FixedAcidity + VolatileAcidity + CitricAcid +
ResidualSugar +
    Chlorides + FreeSulfurDioxide + TotalSulfurDioxide + Density +
    pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS
 ,  data=wine_training3);

summary(lmod1);

par(mfrow=c(2,2))
plot(lmod1)
hist(resid(lmod1), main="Histogram of Residuals");
ols_test_breusch_pagan(lmod1);
vif(lmod1);
```

```r
plot(predict(lmod1),wine_training3$TARGET,
      xlab="predicted",ylab="actual")
 abline(a=0,b=1)


 lstep<-stepAIC(lmod1, trace=FALSE)
lstep$anova


lmod2 <- lm(TARGET ~ VolatileAcidity + CitricAcid +
    Chlorides + FreeSulfurDioxide + TotalSulfurDioxide + Density +
    pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS
 ,  data=wine_training3);

summary(lmod2);

par(mfrow=c(2,2))
plot(lmod2)
hist(resid(lmod2), main="Histogram of Residuals");
ols_test_breusch_pagan(lmod2);
vif(lmod2);

plot(predict(lmod2),wine_training3$TARGET,
      xlab="predicted",ylab="actual")
 abline(a=0,b=1)


 library(ISLR)

lmod3<-lm(TARGET ~ VolatileAcidity + CitricAcid +
    Chlorides + FreeSulfurDioxide + TotalSulfurDioxide + Density +
    pH + Sulphates + Alcohol + s(LabelAppeal) + AcidIndex + s(STARS)
 ,  data=wine_training3)

summary(lmod3);

par(mfrow=c(2,2))
plot(lmod3)
hist(resid(lmod3), main="Histogram of Residuals");
ols_test_breusch_pagan(lmod2);
vif(lmod3);

plot(predict(lmod3),wine_training3$TARGET,
      xlab="predicted",ylab="actual")
 abline(a=0,b=1);

 #goodness of fit
 #pchisq(summary(pmod7)$deviance,
  #         summary(pmod7)$df.residual
```

```
   #        )



require(ggplot2)
require(pscl)
require(MASS)
require(boot)

pmod7 <- zeroinfl(TARGET ~   Alcohol  + AcidIndex | STARS+LabelAppeal,
  data = wine_training3)
summary(pmod7);


#glm.diag.plots(nmod3, glmdiag = glm.diag(nmod3), subset = NULL,
 #              iden = FALSE, labels = NULL, ret = FALSE)

#cov.pmod <- vcovHC(pmod, type="HC0")
#std.err <- sqrt(diag(cov.pmod))
#r.est <- cbind(Estimate= coef(pmod), "Robust SE" = std.err,
#"Pr(>|z|)" = 2 * pnorm(abs(coef(pmod)/std.err), lower.tail=FALSE),
#LL = coef(pmod) - 1.96 * std.err,
#UL = coef(pmod) + 1.96 * std.err);

#r.est;

#with(nmod3, cbind(res.deviance = deviance, df = df.residual,p =
pchisq(deviance, df.residual, lower.tail=FALSE)));

#exp(cbind("Odds ratio" = coef(pmod7), confint.default(pmod7, level =
0.95)));

#n2<-1-(18500/22860)
#n2;

plot(log(fitted(pmod7)), log((wine_training3$TARGET-fitted(pmod7))^2),
xlab=expression(hat(mu)), ylab=expression((y-hat(mu))^2))
abline(0, 1)

vuong(pmod7, pmod4);

plot(predict(pmod7),wine_training3$TARGET,
      xlab="predicted",ylab="actual")
 abline(a=0,b=1);

#goodness of fit
# pchisq(summary(pmod7)$deviance,
 #          summary(pmod7)$df.residual
```

```
  #         )



#model selection

  pnull <- update(pmod7, . ~ 1)

pchisq(2 * (logLik(pmod7) - logLik(pnull)), df = 3, lower.tail = FALSE)



library(caret)
Train <- createDataPartition(wine_training3$TARGET, p=0.7, list=FALSE)
train <- wine_training3[Train, ]
test <- wine_training3[-Train, ]


dput(coef(pmod7, "count"));dput(coef(pmod7, "zero"))


f <- function(data, i)
  {
  require(pscl)
  m <- zeroinfl(TARGET ~   Alcohol  + AcidIndex | STARS+LabelAppeal,
data = data[i, ],
    start = list(count = c(1.5350871583595, 0.0100184890625652, -
0.0419759809109108
), zero = c(-0.163773933205731, -0.658480007306658, 0.180109965808006
)))
  as.vector(t(do.call(rbind, coef(summary(m)))[, 1:2]))
 }

set.seed(10)
res <- boot(wine_training3, f, R = 100, parallel = "snow", ncpus = 4)

## print results
res



#conclusion
#gather predicted
test_results2<-predict(pmod7, newdata=test, type = "response")
target_pred<-data.frame(test_results2)

actuals<-subset(test,select=c(TARGET))
```

```r
#plot
results<-data.frame(target_pred, actuals)

xyplot(TARGET ~ test_results2, data = results,
  xlab = "Predicted ",
  ylab = "Actuals",
  main = "Predicted Wine Sales vs Actual Wine Sales on Test Data");

plot_density(results);

summary(results$TARGET);summary(results$test_results2)
```