data621hw5

jim lung

May 1, 2018

Description

Explore, analyze and model a data set containing information on approximately 12,000 commercially available wines. The variables are mostly related to the chemical properties of the wine being sold. The response variable is the number of sample cases of wine that were purchased by wine distribution companies after sampling a wine. These cases would be used to provide tasting samples to restaurants and wine stores around the United States. The more sample cases purchased, the more likely is a wine to be sold at a high end restaurant. A large wine manufacturer is studying the data in order to predict the number of wine cases ordered based upon the wine characteristics. If the wine manufacturer can predict the number of cases, then that manufacturer will be able to adjust their wine offering to maximize sales.

The objective is to build a count regression model to predict the number of cases of wine that will be sold given certain properties of the wine. This variable is called TARGET.

Data Exploration

There are numerous NAs in certain variables, and variables with negative values. Variables with negative values have nearly normal distributions so it is possible some previous data adjustments have been made. The variable data with negative values in stable, normal distributions will be used as-is. Below is a summary of variables by type, followed by their basic statistical summaries:

```
## Loading required package: ggplot2

## Loading required package: ROCR

## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess

## Loading required package: RCurl

## Loading required package: bitops

## Loading required package: knitr
## Loading required package: Hmisc

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units

## Loading required package: caret

##
## Attaching package: 'caret'

## The following object is masked from 'package:survival':
##
##     cluster

## Loading required package: stringr

## Loading required package: mice

##
## Attaching package: 'mice'

## The following object is masked from 'package:RCurl':
##
##     complete

## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:Hmisc':
##
##     src, summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: reshape2
## Loading required package: MASS

## 
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
## 
##     select

## 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

## Loading required package: broom

## Loading required package: lmtest

## Loading required package: zoo

## 
## Attaching package: 'zoo'

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

## 
## Attaching package: 'lmtest'

## The following object is masked from 'package:RCurl':
## 
##     reset
```

```r
# Read in the dataset from github
wine.raw <-
read.csv(text=getURL("https://raw.githubusercontent.com/fung1091/data621/mast
er/HW5/wine-training-data.csv"),header=TRUE,na.strings=c(""," "),
stringsAsFactors = FALSE)
wine_eval <-
read.csv(text=getURL("https://raw.githubusercontent.com/fung1091/data621/mast
er/HW5/wine-evaluation-data.csv"),header=TRUE,na.strings=c(""," "),
stringsAsFactors = FALSE)

# remove INDEX column since it's not used
wine <- wine.raw[2:length(wine.raw)]

# Let's start by exploring the type of each variable
```

```r
types <- sapply(1:length(wine),function(x) typeof(wine[,x]))
types.df <- data.frame(VAR=names(wine),TYPE=types)
kable(types.df)
```

VAR	TYPE
TARGET	integer
FixedAcidity	double
VolatileAcidity	double
CitricAcid	double
ResidualSugar	double
Chlorides	double
FreeSulfurDioxide	double
TotalSulfurDioxide	double
Density	double
pH	double
Sulphates	double
Alcohol	double
LabelAppeal	integer
AcidIndex	integer
STARS	integer

```r
# Now generate some summary statistics
kable(summary(wine[1:6]))
```

TARGET	FixedAcidity	VolatileAcidity	CitricAcid	ResidualSugar	Chlorides
Min. :0.000	Min. :-18.100	Min. :-2.7900	Min. :-3.2400	Min. :-127.800	Min. :-1.1710
1st Qu.:2.000	1st Qu.: 5.200	1st Qu.: 0.1300	1st Qu.: 0.0300	1st Qu.: -2.000	1st Qu.:-0.0310
Median :3.000	Median : 6.900	Median : 0.2800	Median : 0.3100	Median : 3.900	Median : 0.0460
Mean :3.029	Mean : 7.076	Mean : 0.3241	Mean : 0.3084	Mean : 5.419	Mean : 0.0548
3rd Qu.:4.000	3rd Qu.: 9.500	3rd Qu.: 0.6400	3rd Qu.: 0.5800	3rd Qu.: 15.900	3rd Qu.: 0.1530
Max. :8.000	Max. : 34.400	Max. : 3.6800	Max. : 3.8600	Max. : 141.150	Max. : 1.3510
NA	NA	NA	NA	NA's :616	NA's :638

```r
kable(summary(wine[7:12]))
```

FreeSulfurDioxide	TotalSulfurDioxide	Density	pH	Sulphates	Alcohol

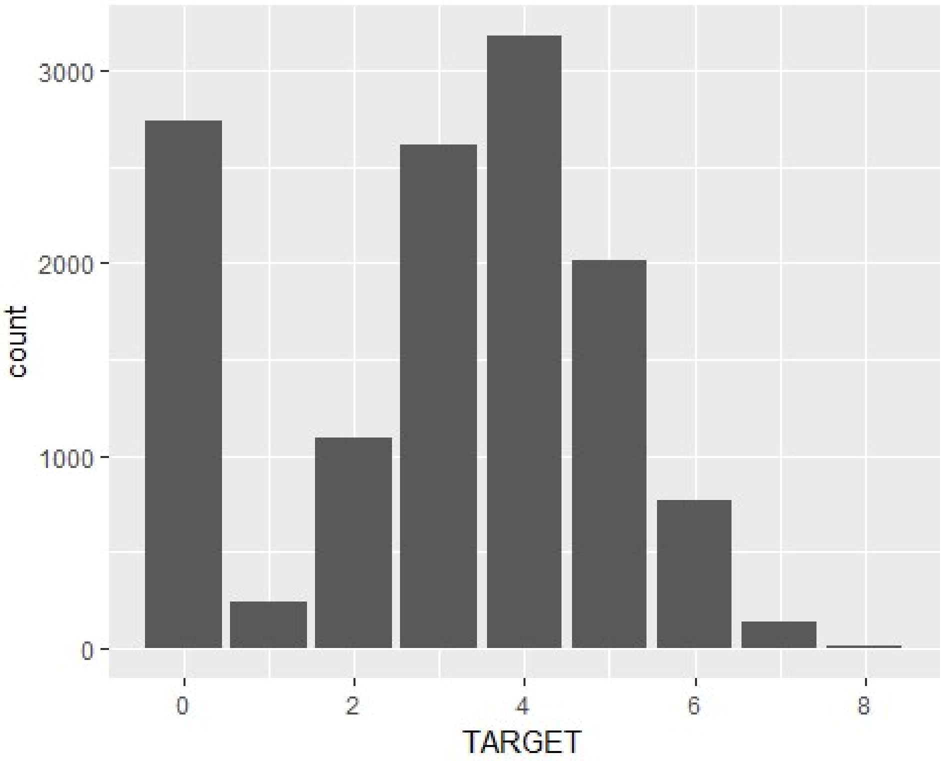
Min. :-555.00	Min. :-823.0	Min. :0.8881	Min. :0.480	Min. :-3.1300	Min. :-4.70
1st Qu.: 0.00	1st Qu.: 27.0	1st Qu.:0.9877	1st Qu.:2.960	1st Qu.: 0.2800	1st Qu.: 9.00
Median : 30.00	Median : 123.0	Median :0.9945	Median :3.200	Median : 0.5000	Median :10.40
Mean : 30.85	Mean : 120.7	Mean :0.9942	Mean :3.208	Mean : 0.5271	Mean :10.49
3rd Qu.: 70.00	3rd Qu.: 208.0	3rd Qu.:1.0005	3rd Qu.:3.470	3rd Qu.: 0.8600	3rd Qu.:12.40
Max. : 623.00	Max. :1057.0	Max. :1.0992	Max. :6.130	Max. : 4.2400	Max. :26.50
NA's :647	NA's :682	NA	NA's :395	NA's :1210	NA's :653

```
kable(summary(wine[13:length(wine)]))
```

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	NA	NA's :3359

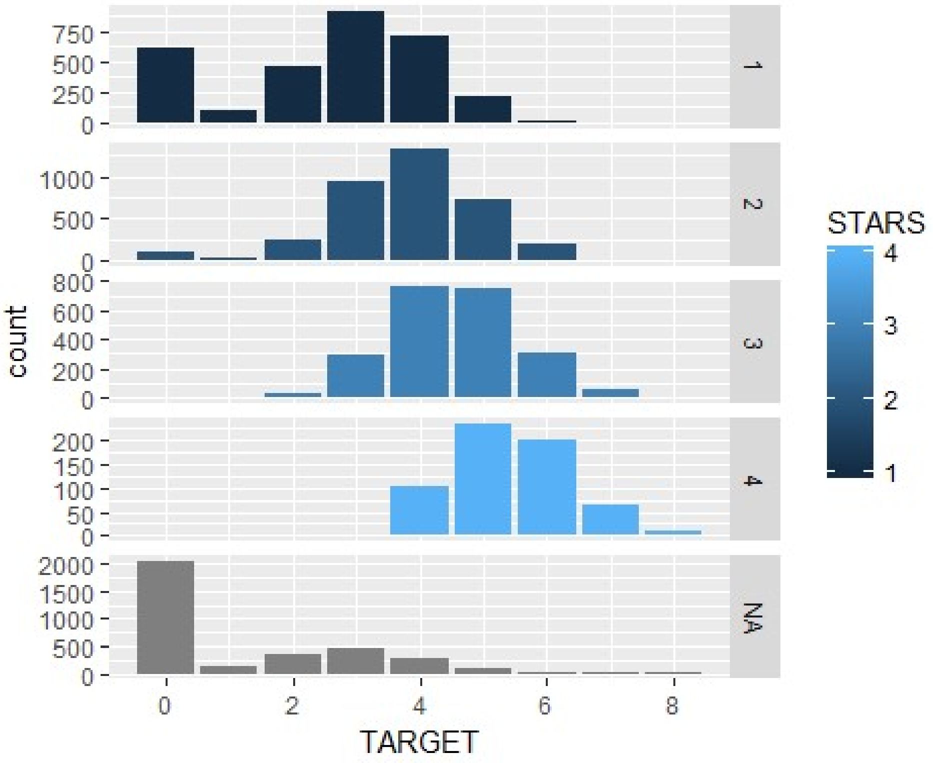
There are numerous NAs in certain variables, and variables with negative values. Variables with negative values have apparently normal distributions so it's possible some previous data adjustments have been made. The variable data with negative values in stable, normal distributions will be used as-is.

Below is a plot of the distribution of counts for the TARGET variable.



Here is another look at the TARGET variable, stratified by the number of Stars rating given for each wine.

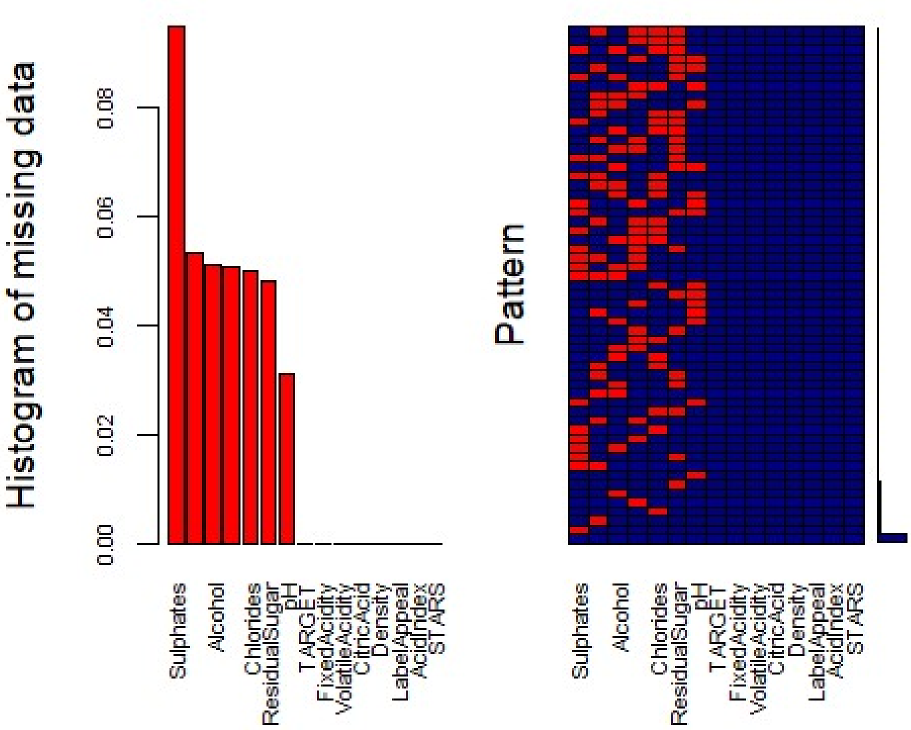
```
ggplot(wine, aes(TARGET, fill = STARS)) + geom_bar(stat = "count") +
facet_grid(STARS ~
    ., margins = FALSE, scales = "free")
```



Data Preparation

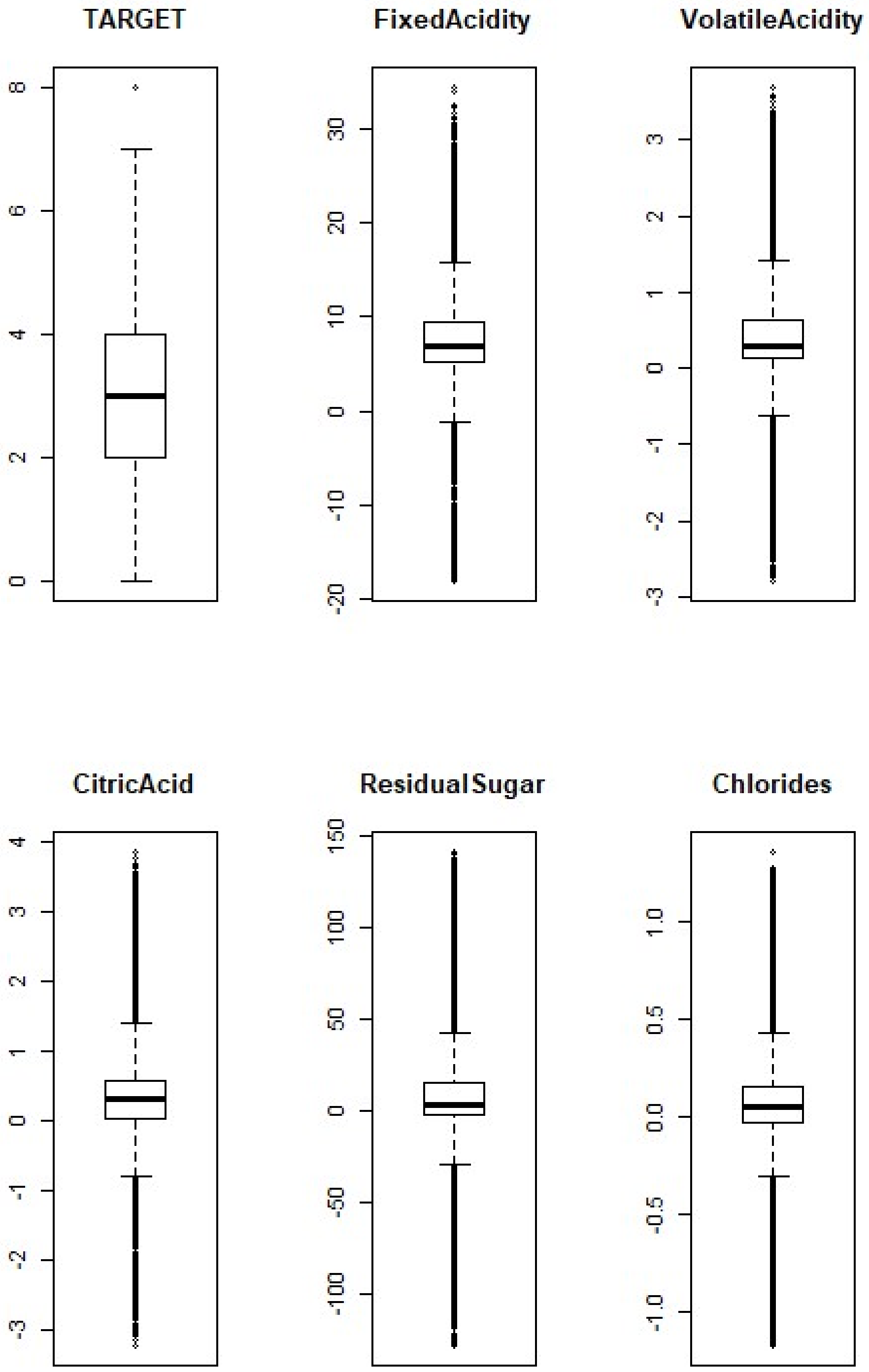
We will cleanse the data by removing the index column, using the MICE package to replace NA's with meaningful values, and setting the unrated wines (no stars) to zero stars, so they can be analyzed quantitatively.

```
## Loading required package: colorspace

## Loading required package: grid

## Loading required package: data.table

## 
## Attaching package: 'data.table'

## The following objects are masked from 'package:reshape2':
## 
##     dcast, melt

## The following objects are masked from 'package:dplyr':
## 
##     between, first, last

## VIM is ready to use. 
##  Since version 4.0.0 the GUI is in its own package VIMGUI.
## 
##           Please use the package to use the new (and old) GUI.
## Suggestions and bug-reports can be submitted at:
https://github.com/alexkowa/VIM/issues

##
## Attaching package: 'VIM'

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

## Warning in plot.aggr(res, ...): not enough vertical space to display
## frequencies (too many combinations)
```

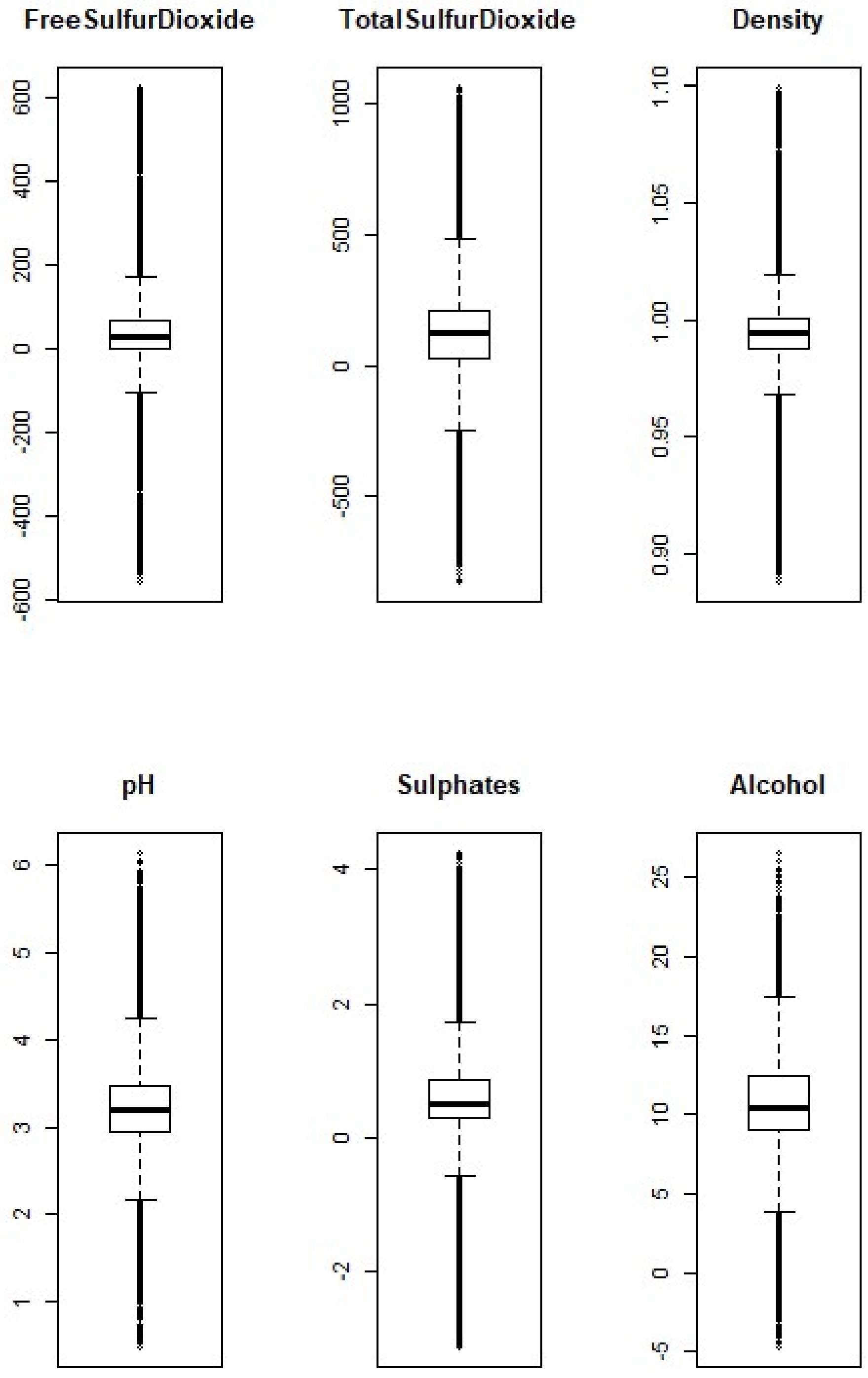


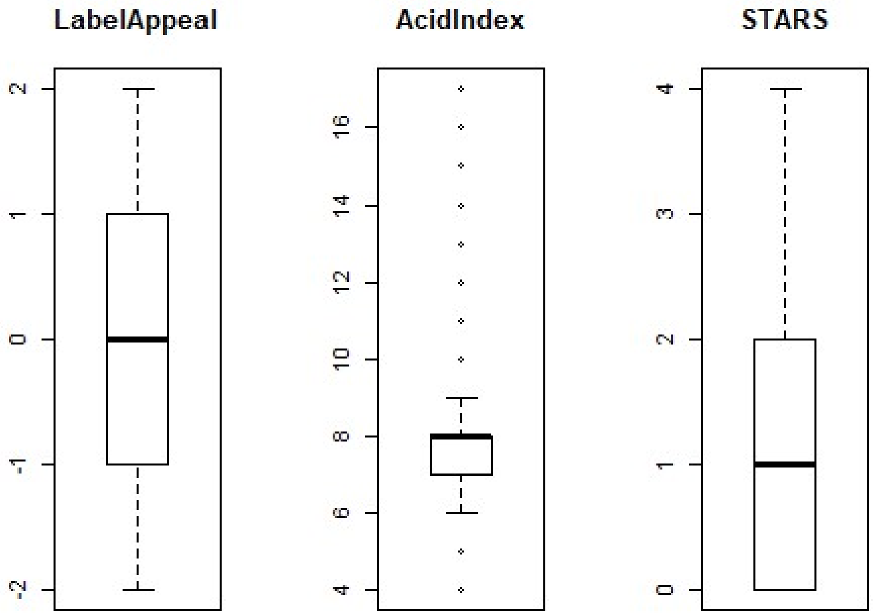
```
##
##  Variables sorted by number of missings:
##            Variable      Count
##           Sulphates 0.09456819
##  TotalSulfurDioxide 0.05330207
##             Alcohol 0.05103556
##   FreeSulfurDioxide 0.05056663
##           Chlorides 0.04986323
##       ResidualSugar 0.04814381
##                  pH 0.03087143
##              TARGET 0.00000000
##        FixedAcidity 0.00000000
##     VolatileAcidity 0.00000000
##          CitricAcid 0.00000000
##             Density 0.00000000
##         LabelAppeal 0.00000000
##           AcidIndex 0.00000000
##               STARS 0.00000000
```

Below are boxplots of the independent variables, which illustrate the normality of the data.



TARGET
0
2
4
6
8
FixedAcidity
-20
-10
0
10
20
30
VolatileAcidity
-3
-2
-1
0
1
2
3
CitricAcid
-3
-2
-1
0
1
2
3
4
ResidualSugar
-100
-50
0
50
100
150
Chlorides
-1.0
-0.5
0.0
0.5
1.0

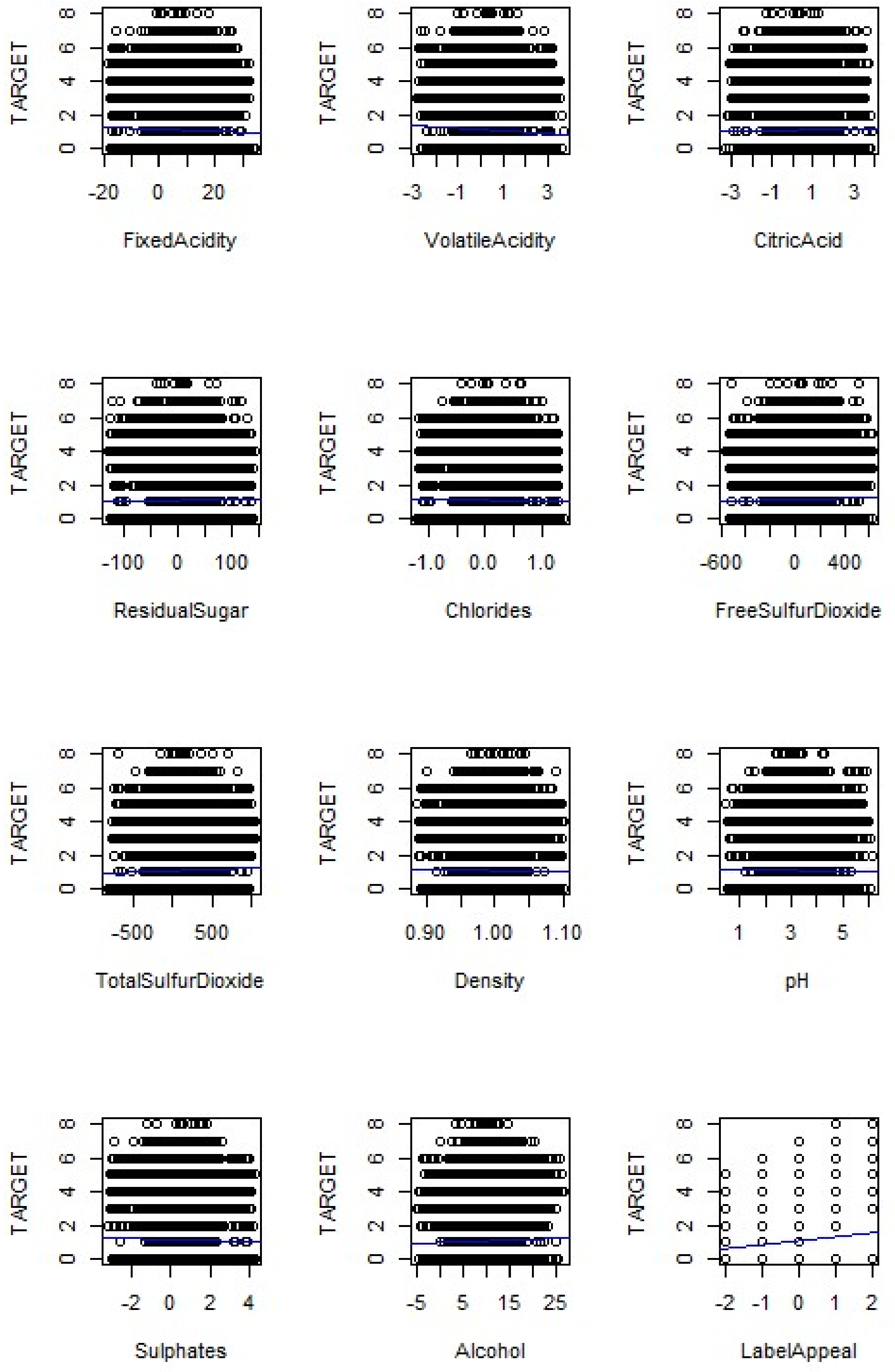

FreeSulfurDioxide
600
400
200
0
-200
-400
-600
TotalSulfurDioxide
1000
500
0
-500
Density
1.10
1.05
1.00
0.95
0.90
pH
6
5
4
3
2
1
Sulphates
4
2
0
-2
Alcohol
25
20
15
10
5
0
-5

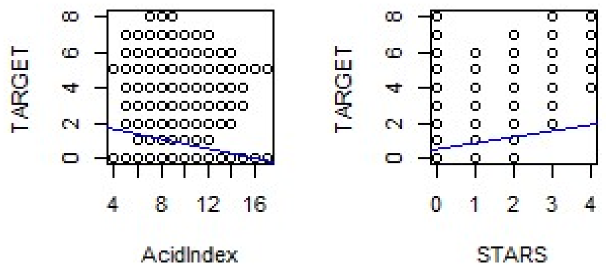


Build Models

Regular Poisson

To take a deeper look at the data, first we create a model for each variable individually - to get a sense of how each variable interacts with the outcome on its own, as a means to inform us how we might use groups of variables to build the best models.

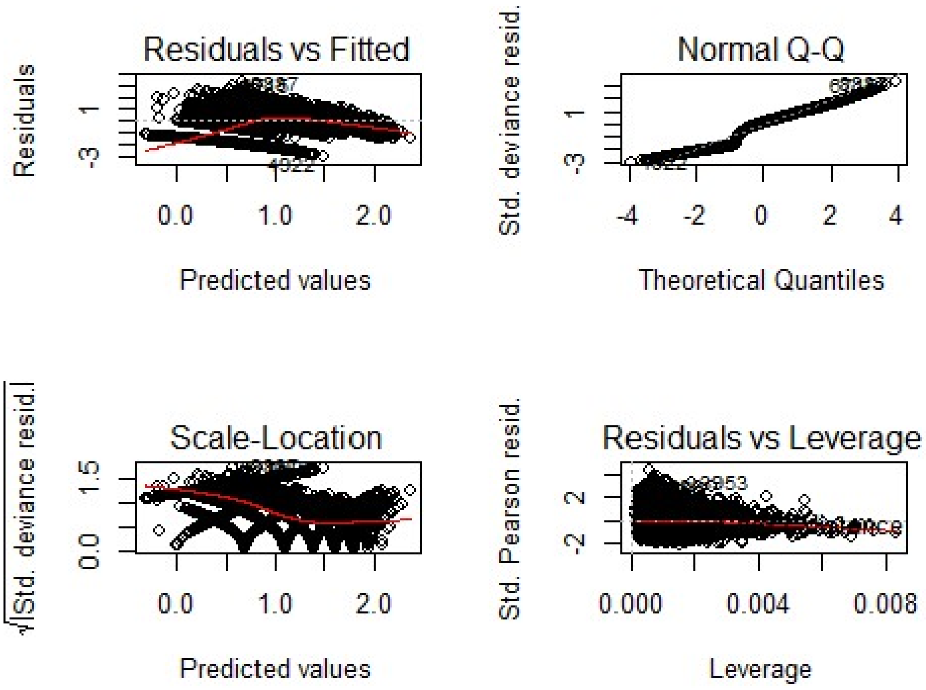

TARGET
0
2
4
6
8
-20
0
20
FixedAcidity
TARGET
-3
-1
1
3
VolatileAcidity
TARGET
-3
-1
1
3
CitricAcid
TARGET
-100
0
100
ResidualSugar
TARGET
-1.0
0.0
1.0
Chlorides
TARGET
-600
0
400
FreeSulfurDioxide
TARGET
-500
500
TotalSulfurDioxide
TARGET
0.90
1.00
1.10
Density
TARGET
1
3
5
pH
TARGET
-2
0
2
4
Sulphates
TARGET
-5
5
15
25
Alcohol
TARGET
-2
-1
0
1
2
LabelAppeal



By looking at these models we suspect there may be two forces at work. The first we will call Perception. The two Perception variables by theoretical effect are Stars and Label Appeal. Based on the high coefficients and high significance, Perception seems to impact the outcome much more than anything else. All the other variables could belong to this group. The pattern we see here is that the best outcome (highest number of cases purchased) tends to occur which variables are close to the mean.

Next we will create a generalized linear model, Poisson family, that combines all the variables:

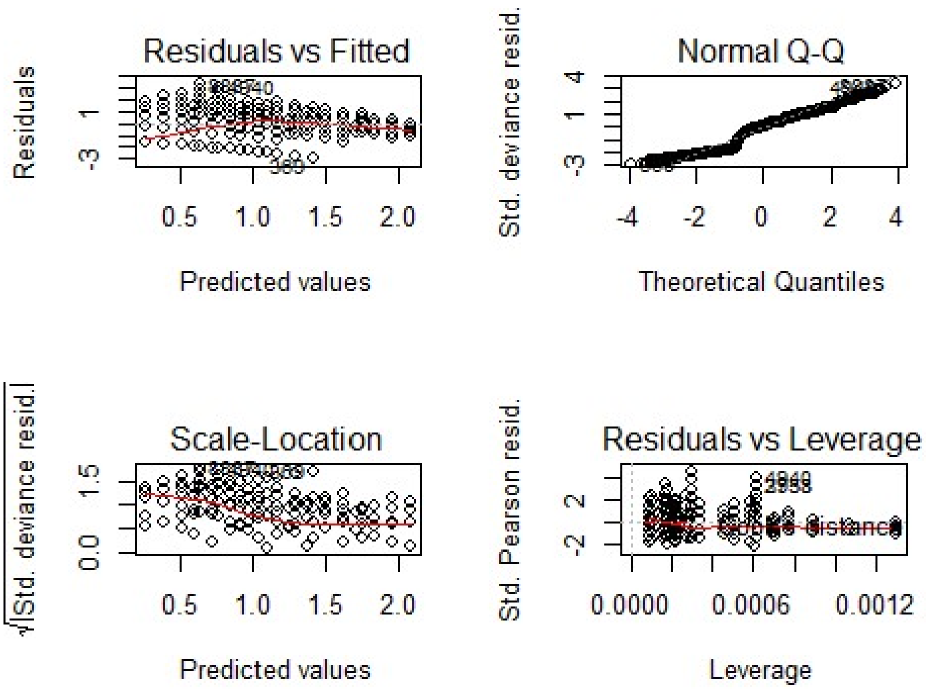
```
##
## Call:
## glm(formula = as.formula(paste(colnames(wine)[1], "~",
paste(colnames(wine)[-1],
##     collapse = "+"), sep = "")), family = poisson(), data = wine)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9665  -0.7241   0.0683   0.5785   3.2292
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         1.540e+00  1.953e-01   7.885 3.15e-15 ***
## FixedAcidity       -3.144e-04  8.206e-04  -0.383 0.701618
## VolatileAcidity    -3.335e-02  6.517e-03  -5.118 3.08e-07 ***
## CitricAcid          7.738e-03  5.894e-03   1.313 0.189191
## ResidualSugar       1.234e-04  1.509e-04   0.818 0.413367    
## Chlorides          -4.122e-02  1.598e-02  -2.579 0.009907 ** 
## FreeSulfurDioxide   1.205e-04  3.418e-05   3.527 0.000421 ***
## TotalSulfurDioxide  8.393e-05  2.205e-05   3.807 0.000141 ***
## Density            -2.845e-01  1.919e-01  -1.482 0.138278    
## pH                 -1.747e-02  7.514e-03  -2.324 0.020109 *  
## Sulphates          -1.413e-02  5.480e-03  -2.579 0.009919 ** 
## Alcohol             1.688e-03  1.373e-03   1.230 0.218822    
## LabelAppeal         1.334e-01  6.063e-03  21.995  < 2e-16 ***
## AcidIndex          -8.705e-02  4.551e-03 -19.128  < 2e-16 ***
## STARS               3.112e-01  4.533e-03  68.647  < 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: 14724  on 12780  degrees of freedom
## AIC: 46696
## 
## Number of Fisher Scoring iterations: 5

## [1] "Chi-Square Test =  0"
```



Here we see that the Perception variables have an outsize impact on the outcome.

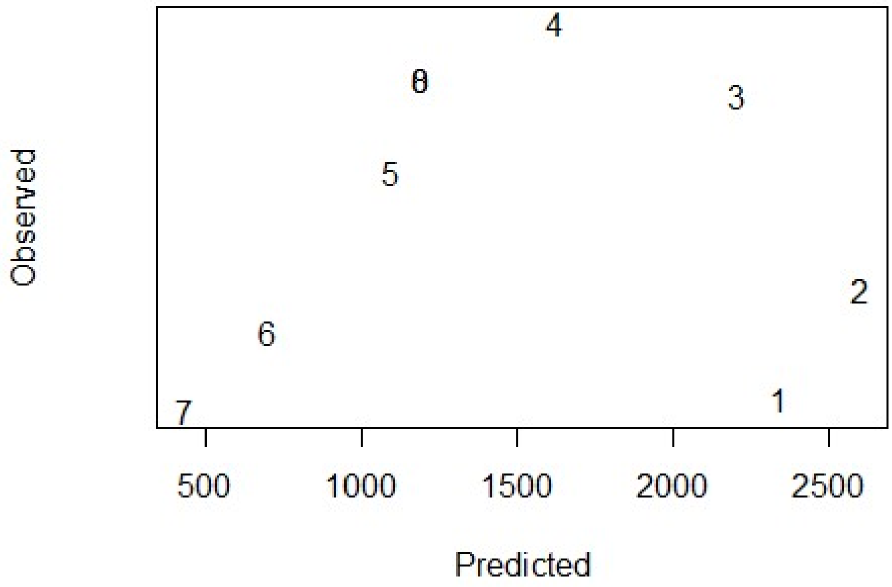
Let’s create a Poisson model using only the two Perception variables:

```
##
## Call:
## glm(formula = TARGET ~ STARS + LabelAppeal, family = poisson(),
##     data = wine)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8852  -0.7533   0.0842   0.6161   3.2856
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.516912   0.010057   51.40   <2e-16 ***
## STARS       0.329083   0.004437   74.16   <2e-16 ***
## LabelAppeal 0.125476   0.006042   20.77   <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: 15221  on 12792  degrees of freedom
## AIC: 47169
##
## Number of Fisher Scoring iterations: 5

## [1] 0
```



Zero-inflated Poisson Model

We next explore the seemingly high number of zero cases in the TARGET count as seen in the previous histogram. We can easily see if the number of zeros observed is in line with the number of zeros predicted by the Poisson model alone.



The number of observed zero cases and the predicted zero cases do not match up well so we'll move to look at the influence of the zero counts on the model by separating out the modeling of zero counts and the modeling of the non-zero counts.

Staying with our concepts of Perception with other variables, we will look treating the high number of zero counts using the Perception variables of STARS and LabelAppeal, and the non-zero counts will use all other variables.

```
##
## Call:
## zeroinfl(formula = TARGET ~ . - (STARS + LabelAppeal) | STARS +
##     LabelAppeal, data = wine, dist = "poisson")
##
## Pearson residuals:
##      Min       1Q   Median       3Q      Max
## -1.95838 -0.49259  0.04251  0.52324  4.77815
##
## Count model coefficients (poisson with log link):
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         1.856e+00  2.012e-01   9.226  < 2e-16 ***
## FixedAcidity        9.075e-06  8.347e-04   0.011 0.991325
## VolatileAcidity    -2.338e-02  6.679e-03  -3.501 0.000464 ***
## CitricAcid          4.397e-03  6.025e-03   0.730 0.465532    
## ResidualSugar       4.993e-05  1.538e-04   0.325 0.745454    
## Chlorides          -2.263e-02  1.632e-02  -1.387 0.165362    
## FreeSulfurDioxide   3.561e-05  3.441e-05   1.035 0.300773    
## TotalSulfurDioxide -1.900e-05  2.183e-05  -0.870 0.384190    
## Density            -4.142e-01  1.977e-01  -2.095 0.036189 *  
## pH                  8.575e-03  7.698e-03   1.114 0.265326    
## Sulphates          -1.482e-03  5.634e-03  -0.263 0.792521    
## Alcohol             8.955e-03  1.386e-03   6.463 1.03e-10 ***
## AcidIndex          -2.986e-02  5.042e-03  -5.923 3.17e-09 ***
## 
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  0.57368    0.03747   15.31   <2e-16 ***
## STARS       -2.28579    0.05256  -43.49   <2e-16 ***
## LabelAppeal  0.55886    0.03628   15.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Number of iterations in BFGS optimization: 21 
## Log-likelihood: -2.196e+04 on 16 Df
```

```
# p-value = 1 - pchisq(deviance, degrees of freedom)
# Chi-Square Test
lrt <- function (obj1, obj2) {
    L0 <- logLik(obj1)
    L1 <- logLik(obj2)
    L01 <- as.vector(- 2 * (L0 - L1))
    df <- attr(L1, "df") - attr(L0, "df")
    list(L01 = L01, df = df,
        "p-value" = pchisq(L01, df, lower.tail = FALSE))
}

lrt(zp,zp2)
```

```
## $L01
## [1] 3117.713
## 
## $df
## [1] 14
## 
## $`p-value`
## [1] 0
```

This yields a high significant p-value; thus, our overall model is statistically significant.

After analyzing the p-values for the Chemistry portion of the zero-inflated model, there are only 4 statistically significant variables: VolatileAcidity, Density, Alcohol, and AcidIndex.

We'll re-reun the zero-inflated Poisson model with just these variables in the Poisson portion.

```
##
## Call:
## zeroinfl(formula = TARGET ~ (VolatileAcidity + Density + Alcohol +
##     AcidIndex) - (STARS + LabelAppeal) | STARS + LabelAppeal, data = wine,
##     dist = "poisson")
##
## Pearson residuals:
##      Min       1Q   Median       3Q      Max
## -1.95788 -0.49203  0.04298  0.52756  4.79642
##
## Count model coefficients (poisson with log link):
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      1.892609   0.199159   9.503  < 2e-16 ***
## VolatileAcidity -0.023508   0.006676  -3.521  0.00043 ***
## Density         -0.422350   0.197584  -2.138  0.03255 *
## Alcohol          0.008980   0.001385   6.485 8.88e-11 ***
## AcidIndex       -0.030189   0.004977  -6.066 1.31e-09 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.57372    0.03747   15.31   <2e-16 ***
## STARS       -2.28597    0.05255  -43.50   <2e-16 ***
## LabelAppeal  0.55911    0.03628   15.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 13
## Log-likelihood: -2.196e+04 on 8 Df
```

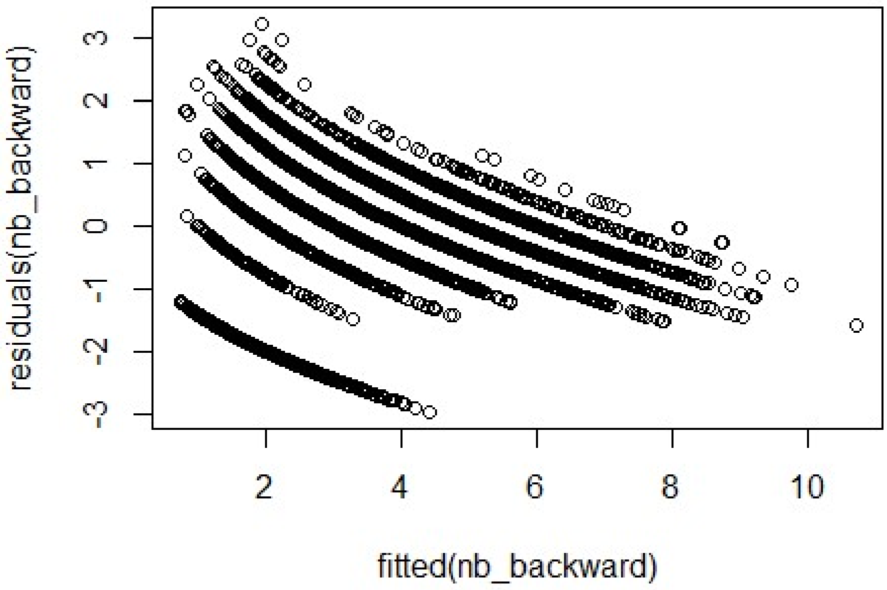
We have reduced the degrees-of-freedom from 16 down to 8 which is as far as we'll go with the zero-inflated Poisson model.

Regular Negative Binomial Model

For regular negative binomial model, we start with all the dependent variables, and perform a backward stepwise algorithm. Initially, we have 14 dependent variables; using this process we reduce to 10 variables. The AIC is 46692

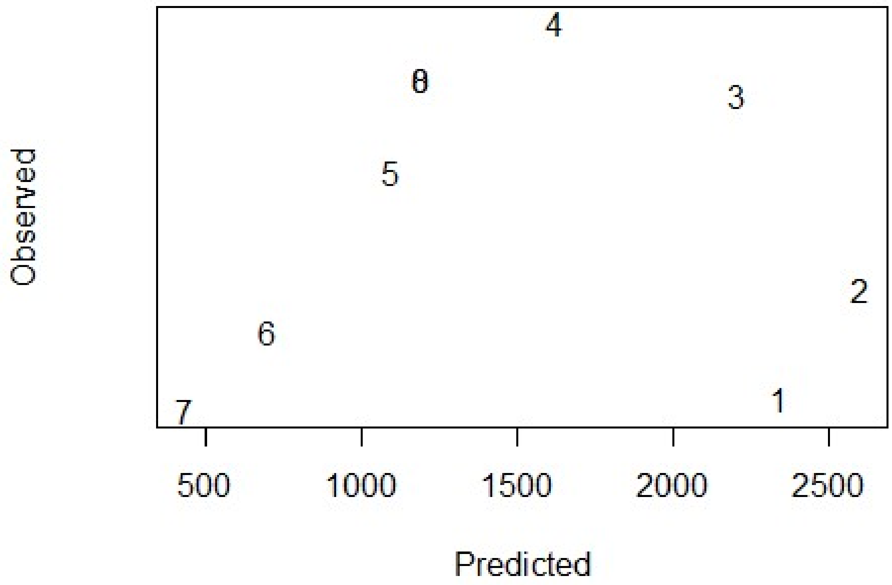
```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached

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

## 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

## 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

## 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

## 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 + FreeSulfurDioxide
+
##     TotalSulfurDioxide + Density + pH + Sulphates + LabelAppeal +
##     AcidIndex + STARS, data = wine, init.theta = 48910.02946,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9743  -0.7281   0.0674   0.5791   3.2408
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         1.563e+00  1.945e-01   8.034 9.43e-16 ***
## VolatileAcidity    -3.355e-02  6.516e-03  -5.148 2.63e-07 ***
## Chlorides          -4.179e-02  1.598e-02  -2.615 0.008912 **
## FreeSulfurDioxide   1.203e-04  3.416e-05   3.522 0.000428 ***
## TotalSulfurDioxide  8.397e-05  2.204e-05   3.810 0.000139 ***
## Density            -2.889e-01  1.919e-01  -1.505 0.132272
## pH                 -1.749e-02  7.512e-03  -2.328 0.019912 *
## Sulphates          -1.427e-02  5.478e-03  -2.604 0.009206 **
## LabelAppeal         1.333e-01  6.064e-03  21.990  < 2e-16 ***
## AcidIndex          -8.717e-02  4.491e-03 -19.410  < 2e-16 ***
## STARS               3.117e-01  4.521e-03  68.938  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for Negative Binomial(48910.03) family taken to be
1)
##
##     Null deviance: 22860  on 12794  degrees of freedom
## Residual deviance: 14728  on 12784  degrees of freedom
## AIC: 46694
##
## Number of Fisher Scoring iterations: 1
##
##
##                Theta:  48910
##            Std. Err.:  50654
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -46670.38
```



Zero-inflated Negative Binomial Regession Model

We'll continue our exploration of the seemingly high number of zero cases in the TARGET count as seen in the previous histogram. In this case, we'll see if the number of zeros observed is in line with the number of zeros predicted by the negative binomial model alone.



The number of observed zero cases and the predicted zero cases do not match up well so we'll move to look at the influence of the zero counts on the model by separating out the modeling of zero counts and the modeling of the non-zero counts.

Staying with our concepts of Perception with other variables, we will look at treating the high number of zero counts using the Perception variables of STARS and LabelAppeal, and the non-zero counts will use all other variables.

```
##
## Call:
## zeroinfl(formula = TARGET ~ . - (STARS + LabelAppeal) | (STARS +
##     LabelAppeal), data = wine, dist = "negbin")
##
## Pearson residuals:
##      Min       1Q   Median       3Q      Max
## -1.95836 -0.49257  0.04248  0.52322  4.77837
##
## Count model coefficients (negbin with log link):
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         1.856e+00  2.012e-01   9.225  < 2e-16 ***
## FixedAcidity        8.135e-06  8.347e-04   0.010 0.992224
## VolatileAcidity    -2.338e-02  6.679e-03  -3.501 0.000463 ***
## CitricAcid          4.400e-03  6.025e-03   0.730 0.465256
## ResidualSugar       4.954e-05  1.538e-04   0.322 0.747385
## Chlorides          -2.262e-02  1.632e-02  -1.386 0.165663
## FreeSulfurDioxide   3.559e-05  3.441e-05   1.034 0.300952
## TotalSulfurDioxide -1.895e-05  2.183e-05  -0.868 0.385495
## Density            -4.144e-01  1.977e-01  -2.096 0.036104 *  
## pH                  8.617e-03  7.698e-03   1.119 0.263007    
## Sulphates          -1.490e-03  5.634e-03  -0.265 0.791381    
## Alcohol             8.960e-03  1.386e-03   6.466 1.00e-10 ***
## AcidIndex          -2.985e-02  5.042e-03  -5.920 3.22e-09 ***
## Log(theta)          1.132e+01  2.789e+00   4.059 4.92e-05 ***
## 
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  0.57372    0.03747   15.31   <2e-16 ***
## STARS       -2.28592    0.05257  -43.48   <2e-16 ***
## LabelAppeal  0.55890    0.03628   15.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Theta = 82456.8215 
## Number of iterations in BFGS optimization: 37 
## Log-likelihood: -2.196e+04 on 17 Df
```

```
# p-value = 1 - pchisq(deviance, degrees of freedom)
# Chi-Square Test
lrt <- function (obj1, obj2) {
    L0 <- logLik(obj1)
    L1 <- logLik(obj2)
    L01 <- as.vector(- 2 * (L0 - L1))
    df <- attr(L1, "df") - attr(L0, "df")
    list(L01 = L01, df = df,
        "p-value" = pchisq(L01, df, lower.tail = FALSE))
}

lrt(zn,zn2)
```

```
## $L01
## [1] 3117.969
## 
## $df
## [1] 14
## 
## $`p-value`
## [1] 0
```

This yields a high significant p-value; thus, our overall model is statistically significant.

After analyzing the p-values for the other variables portion of the zero-inflated model, there are only four statistically significant variables: VolatileAcidity, Density, Alcohol, and AcidIndex. We'll re-reun the zero-inflated Poisson model with just these variables in the negative binomial portion.

```
## Warning in sqrt(diag(vc)[np]): NaNs produced
```

```
##
## Call:
## zeroinfl(formula = TARGET ~ (VolatileAcidity + Density + Alcohol +
##     AcidIndex) - (STARS + LabelAppeal) | STARS + LabelAppeal, data = wine,
##     dist = "negbin")
##
## Pearson residuals:
##      Min       1Q   Median       3Q      Max
## -1.95789 -0.49203  0.04296  0.52756  4.79639
##
## Count model coefficients (negbin with log link):
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      1.892966   0.199159   9.505  < 2e-16 ***
## VolatileAcidity -0.023509   0.006676  -3.521 0.000429 ***
## Density         -0.422703   0.197584  -2.139 0.032406 *
## Alcohol          0.008979   0.001385   6.485 8.89e-11 ***
## AcidIndex       -0.030190   0.004977  -6.066 1.31e-09 ***
## Log(theta)      15.438232         NA      NA       NA
##
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.57375    0.03747   15.31   <2e-16 ***
## STARS       -2.28599    0.05255  -43.50   <2e-16 ***
## LabelAppeal  0.55910    0.03628   15.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 5066861.9845
## Number of iterations in BFGS optimization: 39
## Log-likelihood: -2.196e+04 on 9 Df
```

We have reduced the degrees-of-freedom from 17 down to 9 which is as far as we'll go with the zero-inflated negative binomial model.

Linear Regression Models:

Lastly we are going to look at a regular linear model, as a comparison to the analysis shown above. Again we are going to compare Perception and other variables.

```
lin.mod.perc <- lm(TARGET ~ . - STARS - LabelAppeal,data = wine)
summary(lin.mod.perc)
```

```
##
## Call:
## lm(formula = TARGET ~ . - STARS - LabelAppeal, data = wine)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6184 -1.3123  0.2873  1.3200  5.5565
##
```

```
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)    
## (Intercept)           7.461e+00  6.253e-01  11.932  < 2e-16 ***
## FixedAcidity         -1.263e-03  2.637e-03  -0.479 0.632018    
## VolatileAcidity      -1.873e-01  2.093e-02  -8.947  < 2e-16 ***
## CitricAcid            4.552e-02  1.906e-02   2.388 0.016944 *  
## ResidualSugar         9.120e-04  4.859e-04   1.877 0.060570 .  
## Chlorides            -1.834e-01  5.133e-02  -3.572 0.000356 ***
## FreeSulfurDioxide     4.336e-04  1.102e-04   3.934 8.40e-05 ***
## TotalSulfurDioxide    3.151e-04  7.066e-05   4.460 8.27e-06 ***
## Density              -1.776e+00  6.182e-01  -2.873 0.004066 ** 
## pH                   -6.986e-02  2.413e-02  -2.896 0.003790 ** 
## Sulphates            -6.520e-02  1.765e-02  -3.694 0.000222 ***
## Alcohol               2.616e-02  4.401e-03   5.945 2.84e-09 ***
## AcidIndex            -3.440e-01  1.269e-02 -27.096  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.853 on 12782 degrees of freedom
## Multiple R-squared:  0.07571,    Adjusted R-squared:  0.07484 
## F-statistic: 87.24 on 12 and 12782 DF,  p-value: < 2.2e-16
```

We can see that not all the chemistry variables are significant, using backward stepwise elimination, so we eliminate the insignificant independent variables.

```
lin.mod.back <- step(lin.mod.perc, direction="backward")
```

```
## Start:  AIC=15795.45
## TARGET ~ (FixedAcidity + VolatileAcidity + CitricAcid + ResidualSugar + 
##     Chlorides + FreeSulfurDioxide + TotalSulfurDioxide + Density + 
##     pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS) - 
##     STARS - LabelAppeal
## 
##                      Df Sum of Sq   RSS   AIC
## - FixedAcidity        1      0.79 43884 15794
## <none>                            43883 15795
## - ResidualSugar       1     12.09 43895 15797
## - CitricAcid          1     19.58 43902 15799
## - Density             1     28.35 43911 15802
## - pH                  1     28.79 43912 15802
## - Chlorides           1     43.80 43927 15806
## - Sulphates           1     46.84 43930 15807
## - FreeSulfurDioxide   1     53.13 43936 15809
## - TotalSulfurDioxide  1     68.29 43951 15813
## - Alcohol             1    121.33 44004 15829
## - VolatileAcidity     1    274.84 44158 15873
## - AcidIndex           1   2520.64 46404 16508
## 
## Step:  AIC=15793.68
## TARGET ~ VolatileAcidity + CitricAcid + ResidualSugar + Chlorides + 
##     FreeSulfurDioxide + TotalSulfurDioxide + Density + pH + Sulphates +
##     Alcohol + AcidIndex
##
##                       Df Sum of Sq   RSS   AIC
## <none>                            43884 15794
## - ResidualSugar       1     12.20 43896 15795
## - CitricAcid          1     19.56 43903 15797
## - Density             1     28.34 43912 15800
## - pH                  1     28.82 43912 15800
## - Chlorides           1     43.76 43927 15804
## - Sulphates           1     47.14 43931 15805
## - FreeSulfurDioxide   1     52.92 43937 15807
## - TotalSulfurDioxide  1     68.48 43952 15812
## - Alcohol             1    121.33 44005 15827
## - VolatileAcidity     1    274.97 44159 15872
## - AcidIndex           1   2617.56 46501 16533
```

```
# backward elimiation for CitricAcid and ResidualSugar
lin.mod.back <- lm(TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide +
    TotalSulfurDioxide + Density + pH + Sulphates + Alcohol +
    AcidIndex, data = wine)
summary(lin.mod.back)
```

```
##
## Call:
## lm(formula = TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide +
##     TotalSulfurDioxide + Density + pH + Sulphates + Alcohol +
##     AcidIndex, data = wine)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7148 -1.3075  0.2865  1.3245  5.6254
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          7.482e+00  6.254e-01  11.963  < 2e-16 ***
## VolatileAcidity     -1.886e-01  2.093e-02  -9.009  < 2e-16 ***
## Chlorides           -1.852e-01  5.134e-02  -3.607 0.000311 ***
## FreeSulfurDioxide    4.389e-04  1.102e-04   3.983 6.84e-05 ***
## TotalSulfurDioxide   3.203e-04  7.065e-05   4.534 5.85e-06 ***
## Density             -1.794e+00  6.183e-01  -2.902 0.003712 **
## pH                  -6.986e-02  2.413e-02  -2.895 0.003799 **
## Sulphates           -6.638e-02  1.765e-02  -3.761 0.000170 ***
## Alcohol              2.625e-02  4.400e-03   5.965 2.51e-09 ***
## AcidIndex           -3.431e-01  1.247e-02 -27.519  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.853 on 12785 degrees of freedom
## Multiple R-squared:  0.07503,    Adjusted R-squared:  0.07437
## F-statistic: 115.2 on 9 and 12785 DF,  p-value: < 2.2e-16
```

Comparing this to just the perception data we can see the following:

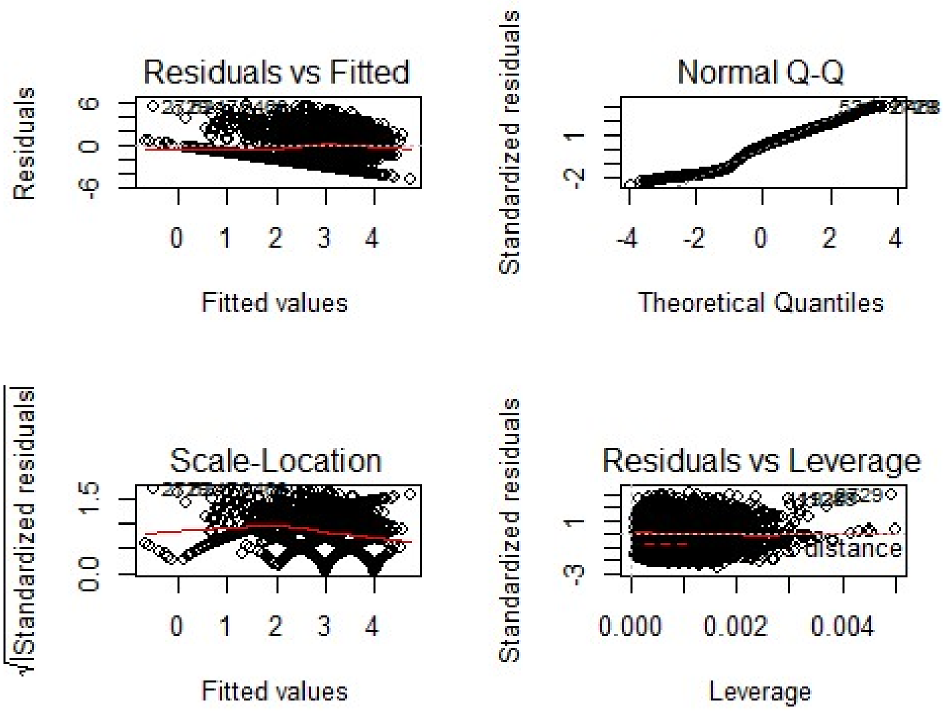
```
lin.mod.app <- lm(TARGET ~ STARS + LabelAppeal,data = wine)
summary(lin.mod.app)

##
## Call:
## lm(formula = TARGET ~ STARS + LabelAppeal, data = wine)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3568 -1.0721  0.0184  0.9279  6.1139
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.47910    0.01988   74.41   <2e-16 ***
## STARS        1.03182    0.01050   98.30   <2e-16 ***
## LabelAppeal  0.40701    0.01398   29.12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.359 on 12792 degrees of freedom
## Multiple R-squared:  0.5027, Adjusted R-squared:  0.5026
## F-statistic:  6466 on 2 and 12792 DF,  p-value: < 2.2e-16
```

From the the models, we can clearly see that the perception data was a much more appropriate fit. This can be seen through the R-Squared value, which shows that the perception model explains roughly 50% of the variance in the model. This is a pretty "good-fit." Using the chemistry data, we also see a significant model, however, the fit is much worse, with practically none of the variance explained.

Checking the residuals we can see the following:

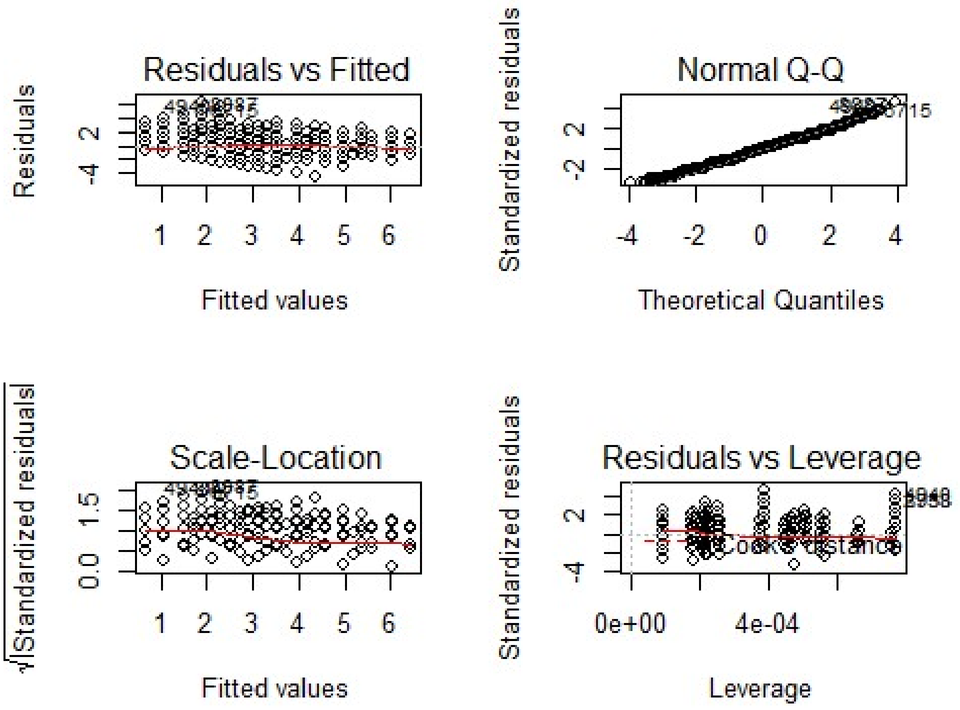
Other variables model:

```
par(mfrow=c(2,2))
plot(lin.mod.back)
```



Perception Model:

```
par(mfrow=c(2,2))
plot(lin.mod.app)
```



From the plots above, we can clearly see that the other variables model shows clear patterns in the residuals, which indicates that linear modeling is not at all a good choice for these particular variables. However, just using the perception variables we see a much better picture, with a more random residual distribution with no clear patterns that would suggest another choice in models.

Select Models

To compare all our regular models first, we build a dataframe which contains all the performance parameters of the models. Out of the four regular models, it is clear that the regular linear model with focus on perception is the best model. It has the lowest AIC and BIC. The Log-Likelihood is also the highest among the four.

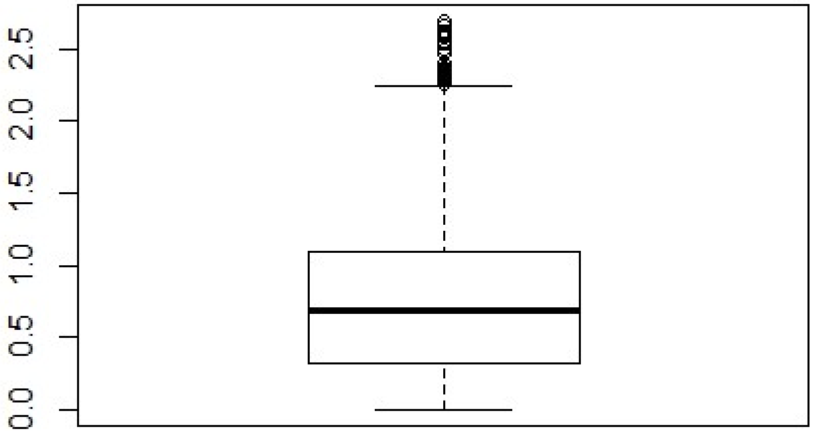
```
##                       Models      AIC      BIC    LogLik Deviance
## 1            Regular Poisson 47168.98 47191.35 -23581.49 15220.96
## 2 Regular Negative Binomial 46694.38 46783.86 -23335.19 14727.57
## 3    Regular Linear Science 52111.51 52193.53 -26044.75 43915.19
## 4 Regular Linear Perception 44157.01 44186.84 -22074.51 23609.92
##   df.residual
## 1       12792
## 2       12784
## 3       12785
## 4       12792
```

When we compare the two zero-inflated models against each other, the following code tells us that the performance differences between two zero inflated models (in terms of LogLik) is not statistically significant since the p value is 0.9569, which is much higher than the significance level 0.05. Their Log Likelihood are both -21960, we have to compare some other performance parameters such as AIC. zero inflated poisson model has slightly smaller AIC (43936.68) compare to the other one(43938.69)

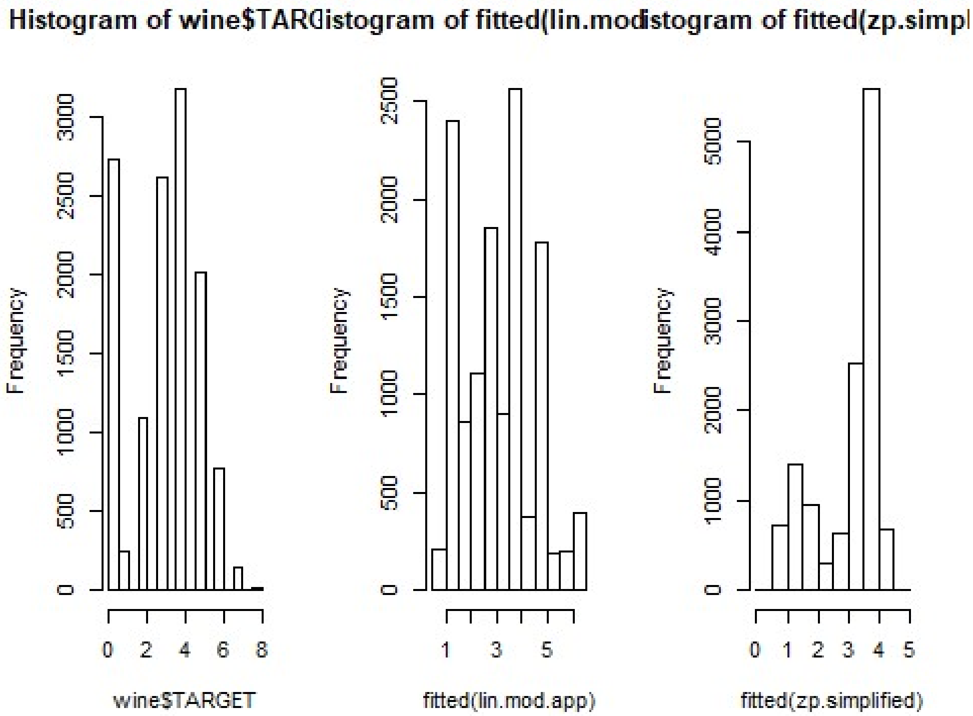
```
#Compare two zero-inflated models
lmtest::lrtest(zn.simplified, zp.simplified)

## Likelihood ratio test
##
## Model 1: TARGET ~ (VolatileAcidity + Density + Alcohol + AcidIndex) -
##     (STARS + LabelAppeal) | STARS + LabelAppeal
## Model 2: TARGET ~ (VolatileAcidity + Density + Alcohol + AcidIndex) -
##     (STARS + LabelAppeal) | STARS + LabelAppeal
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1   9 -21963
## 2   8 -21963 -1 0.0042     0.9485

AIC(zn.simplified)

## [1] 43943.27

AIC(zp.simplified)

## [1] 43941.26
```

By comparing the regular linear model with focus on perception to the zero inflated poisson model, the histogram shows that zero inflated model takes good care of those structural zeros, which are not really zero but more like out of scope. Both models generate predictions that peak at 4 cases of wine, which correspond to the actual observation. Another thing we notice is that the predictions made by two models differ quite significantly. It is recognized according to the boxplot of the absolute differences between two models' residuals. However, based on AIC and Log Likelihood, zero inflated poisson model is still the winner here.

```
boxplot(abs(resid(lin.mod.app) - resid(zp.simplified)))
```



```r
par(mfcol=c(1,3))
hist(wine$TARGET)
hist(fitted(lin.mod.app))
hist(fitted(zp.simplified))
```



```
AIC(lin.mod.app)

## [1] 44157.01

AIC(zp.simplified)

## [1] 43941.26

logLik(lin.mod.app)

## 'log Lik.' -22074.51 (df=4)

logLik(zp.simplified)

## 'log Lik.' -21962.63 (df=8)

#The following code is just the data preparation step for the evaluation
dataset, before we apply our model.
wine_eval <- wine_eval[2:length(wine_eval)]
wine_eval$STARS[is.na(wine_eval$STARS)] <- 0
wine_eval <- mice(wine_eval, m = 3, print=F)
wine_eval <- complete(wine_eval,1)
```

Our final predicted results.

```
wine_eval$TARGET <- predict(zp.simplified, newdata = wine_eval, type =
"response")
head(wine_eval, 20)
```

```
##       TARGET FixedAcidity VolatileAcidity CitricAcid ResidualSugar
## 1  2.0658283          5.4          -0.860       0.27         -10.7
## 2  4.0942548         12.4           0.385      -0.76         -19.7
## 3  2.9413699          7.2           1.750       0.17         -33.0
## 4  3.4673678          6.2           0.100       1.80           1.0
## 5  1.1896843         11.4           0.210       0.28           1.2
## 6  3.8606619         17.6           0.040      -1.15           1.4
## 7  3.2216867         15.5           0.530      -0.53           4.6
## 8  0.9110424         15.9           1.190       1.14          31.9
## 9  1.0781975         11.6           0.320       0.55         -50.9
## 10 1.4487476          3.8           0.220       0.31          -7.7
## 11 3.1048689          6.8           1.680       0.44         -13.3
## 12 0.9005782          9.0          -0.210       0.04          51.4
## 13 3.5640982         24.6           0.030      -1.20           1.3
## 14 1.2945504         13.0           0.210       0.32          -3.2
## 15 1.9084674         17.9          -0.420      -0.91           7.1
## 16 3.4885673         10.0           0.200       1.27          30.9
## 17 3.4104288          7.4           0.290       0.50           8.5
## 18 0.7296401         11.7           1.180      -0.94         -62.0
## 19 3.7437066          9.7           0.410      -1.00          10.2
## 20 4.1808932         -5.2          -0.980      -0.08           6.4
##    Chlorides FreeSulfurDioxide TotalSulfurDioxide Density   pH Sulphates
## 1      0.092                23                398 0.98527 5.02      0.64
## 2      1.169               -37                 68 0.99048 3.37      1.09
## 3      0.065                 9                 76 1.04641 4.61      0.68
## 4     -0.179               104                 89 0.98877 3.20      2.11
## 5      0.038                70                 53 1.02899 2.54     -0.07
## 6      0.535              -250                140 0.95028 3.06     -0.02
## 7      1.263                10                 17 1.00020 3.07      0.75
## 8     -0.299               115                381 1.03416 2.99      0.31
## 9      0.076                35                 83 1.00020 3.32      2.18
## 10     0.039                40                129 0.90610 4.72     -0.64
## 11     0.046               -18                583 1.00833 3.12      1.64
## 12     0.237              -213               -527 0.99516 3.16      0.70
## 13     0.035               241                297 0.99232 2.22      0.50
## 14    -0.263               111                141 0.95918 3.20      0.33
## 15     0.045              -177                169 0.95307 3.17     -1.12
## 16     0.050                19                152 0.99400 3.03      0.42
## 17    -0.480               178                647 0.97275 3.45      0.50
## 18     0.675                 7               -393 0.99974 3.96      0.69
## 19    -0.235                24                113 0.99772 3.44      0.53
## 20     0.046               180                166 0.99400 3.30      2.18
##    Alcohol LabelAppeal AcidIndex STARS
## 1    12.30          -1         6     0
## 2    16.00           0         6     2
## 3     8.55           0         8     1
## 4    12.30          -1         8     1
## 5     4.80           0        10     0
## 6    11.40           1         8     4
## 7     8.50           0        12     3
## 8    11.40           1         7     0
## 9    -0.50           0        12     0
## 10   10.90           0         7     0
## 11   12.60           0         8     1
## 12   14.70           1        10     0
## 13    9.80           0         9     2
## 14    4.20           0         8     0
## 15   13.20          -1         9     0
## 16   13.80          -1        11     2
## 17   10.20          -1         8     1
## 18    5.20           1        13     0
## 19    9.80           0         7     2
## 20    9.90           1         5     3
```