Homework 5 (Group 5)

Count Regression

Maria A Ginorio

r Sys.Date()

Contents

Overview . . . 2
Dataset . . . 2
1. Data Exploration . . . 3
- Objective . . . 3
- Data Overview . . . 3
- Distributions . . . 4
- Outliers . . . 5
- Relationships . . . 6
- Skeweness . . . 7
- Correlation . . . 16

2. Data Preparation . . . 18
- Missing Data . . . 18
- Correlation . . . 19
- Preprocess . . . 20
- **Partition** . . . 20
- Imputation . . . 22
- **Normalization** . . . 23
- Features . . . 24

3. Building Models . . . 26
- Poisson 1 . . . 26
- Poisson 2 . . . 28
- Quassipoisson . . . 29
- Negative Binomial 1 . . . 35
- Negative Binomial with BIC recomendation . . . 36
- Zero Inflated Model . . . 37

4. Select Models . . . 39
- Predicted Probabilities . . . 43

Apendix . . . 43

Overview

In this homework assignment, you will 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

Dataset

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

1. Data Exploration

Objective

- Build a count regression model
- Predict the number of cases of wine that will be sold. (properties)

Data Overview

Lets first look at the raw data values by using the skim package

Table 1: Data summary

Name	wine_train
Number of rows	12795
Number of columns	16
Column type frequency:	
numeric	16
Group variables	None

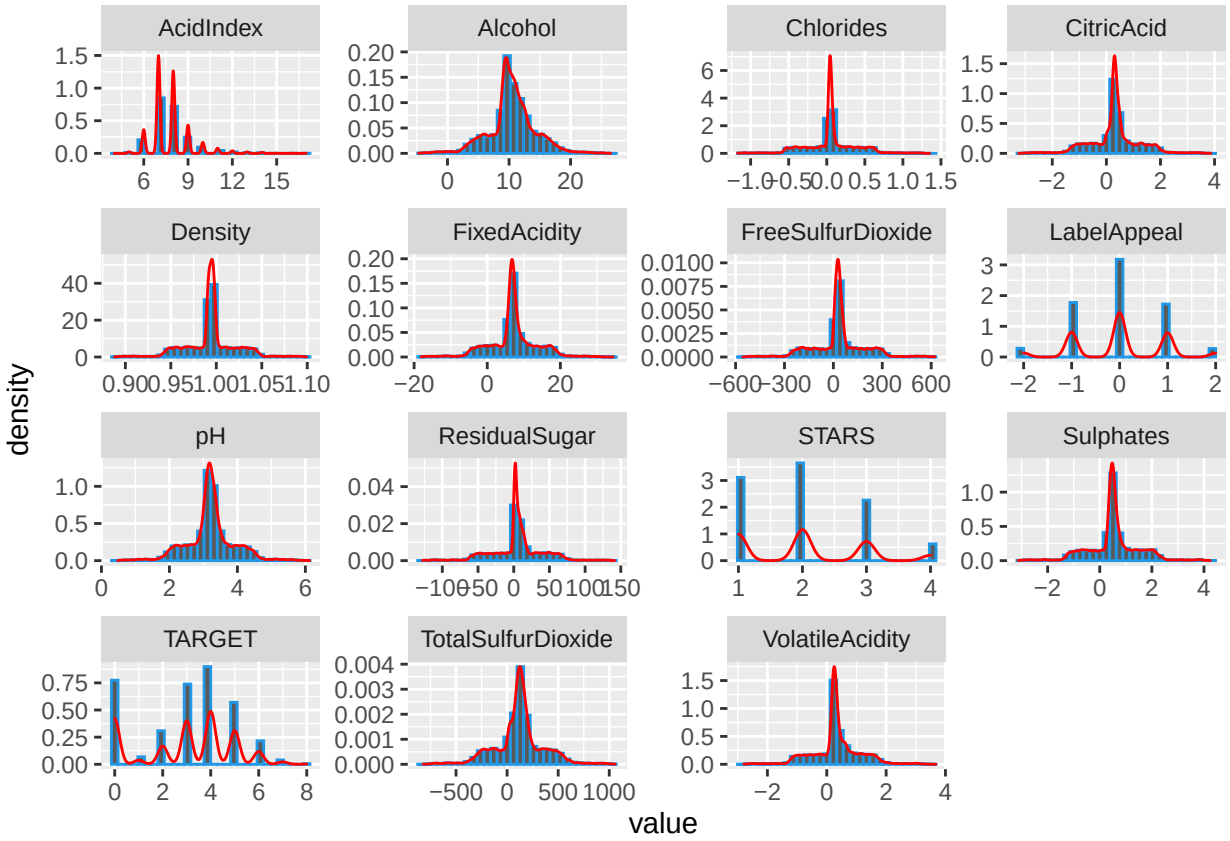
Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
INDEX	0	1.00	8069.98	4656.91	1.00	4037.50	8110.00	12106.50	16129.00
TARGET	0	1.00	3.03	1.93	0.00	2.00	3.00	4.00	8.00
FixedAcidity	0	1.00	7.08	6.32	-18.10	5.20	6.90	9.50	34.40
VolatileAcidity	0	1.00	0.32	0.78	-2.79	0.13	0.28	0.64	3.68
CitricAcid	0	1.00	0.31	0.86	-3.24	0.03	0.31	0.58	3.86
ResidualSugar	616	0.95	5.42	33.75	-127.80	-2.00	3.90	15.90	141.15
Chlorides	638	0.95	0.05	0.32	-1.17	-0.03	0.05	0.15	1.35
FreeSulfurDioxide	647	0.95	30.85	148.71	-555.00	0.00	30.00	70.00	623.00
TotalSulfurDioxide	682	0.95	120.71	231.91	-823.00	27.00	123.00	208.00	1057.00
Density	0	1.00	0.99	0.03	0.89	0.99	0.99	1.00	1.10
pH	395	0.97	3.21	0.68	0.48	2.96	3.20	3.47	6.13
Sulphates	1210	0.91	0.53	0.93	-3.13	0.28	0.50	0.86	4.24
Alcohol	653	0.95	10.49	3.73	-4.70	9.00	10.40	12.40	26.50
LabelAppeal	0	1.00	-0.01	0.89	-2.00	-1.00	0.00	1.00	2.00
AcidIndex	0	1.00	7.77	1.32	4.00	7.00	8.00	8.00	17.00
STARS	3359	0.74	2.04	0.90	1.00	1.00	2.00	3.00	4.00

From the description seen by the skim package we can observe all of our variables are type numeric. There are some variables like `ResidualSugar`, `Chlorides`, `FreeSulfurDioxide`, `TotalSulfurDioxide`, `Sulphates`, `Alcohol` that have missing values.

Distributions

We will first explore the data looking for issues or challenges (i.e. missing data, outliers, possible coding errors, multicollinearlity, etc). Once we have a handle on the data, we will apply any necessary cleaning steps. Once we have a reasonable dataset to work with, we will build and evaluate three different Logistic models that predict seasonal wins.



The distribution of our variables can also alert us of of unusual patterns, in this case we have observed most of our variables appear to have normal distributions. We will explore with more detail each variable to understand their individual behavior.

After creating independent histograms for each variable we have found 3 variables that appear to be bi-modal. We notice that the graphs of this variables have two or more distinct humps or peaks with a valley separating them. We could attribute this observations to possibly different groups. We find that `AcidIndex`, `LabelAppeal` and `STARS` and `Target` are bi-modal.

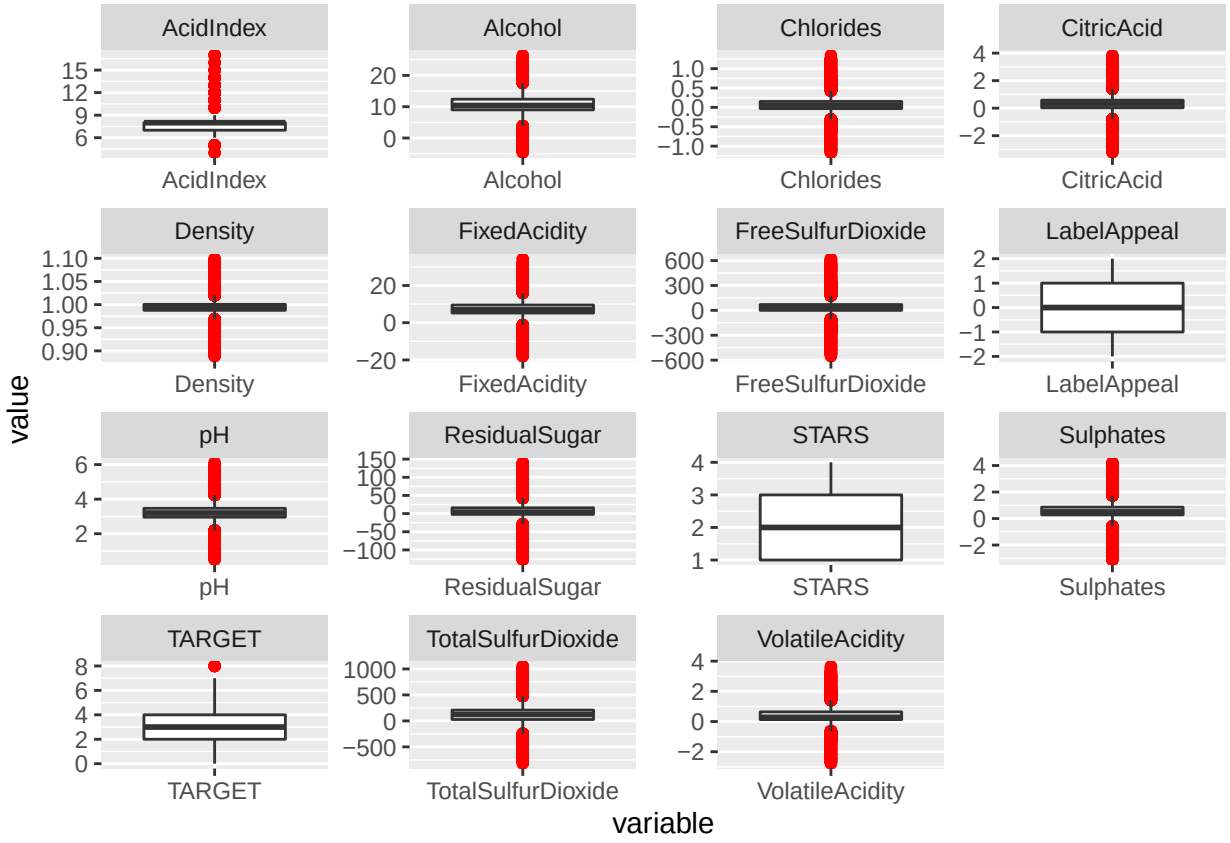
Outliers

In addition to histogram graph of our variable we thought it was pertinent to take a look at our variables using a boxplot. It will help us quickly visualize the distribution of the values in the dataset and see where the five number summary values are located.

In addition, we will be able to create a clear picture of the median values and the spreads across all the distributions. One of the most important observation we will obtain from this graph however, is outlier detection.

Find outliers in red below:

```
## Warning: Removed 8200 rows containing non-finite values (stat_boxplot).
```



Indication of outliers is present in variables almost all variables.

A key is whether an outlier represents a contaminated observation or a rare case.

Are these data points unusual or different in some way from the rest of the data? We will have to consider removing this and refit the data if we consider they could be affecting our results.

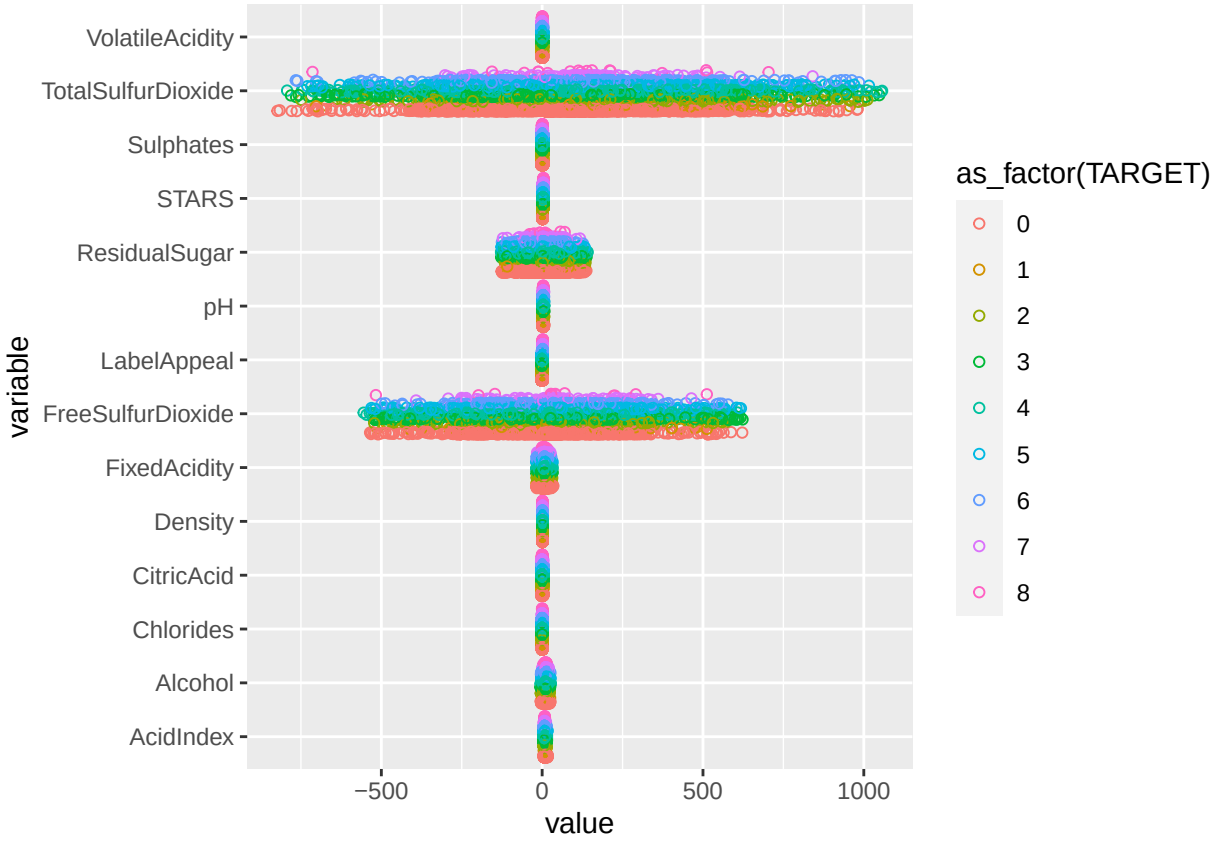
One of the first steps in any type of analysis is to take a closer look at the observations that have high leverage since they could have a large impact on the results of a given model.

Relationships

We want use scatter plots in each variable versus the target variable to get an idea of the relationship between them.

The plots indicate interesting relationship between the `target` variable however some of them start showing signs of relationship and groups.

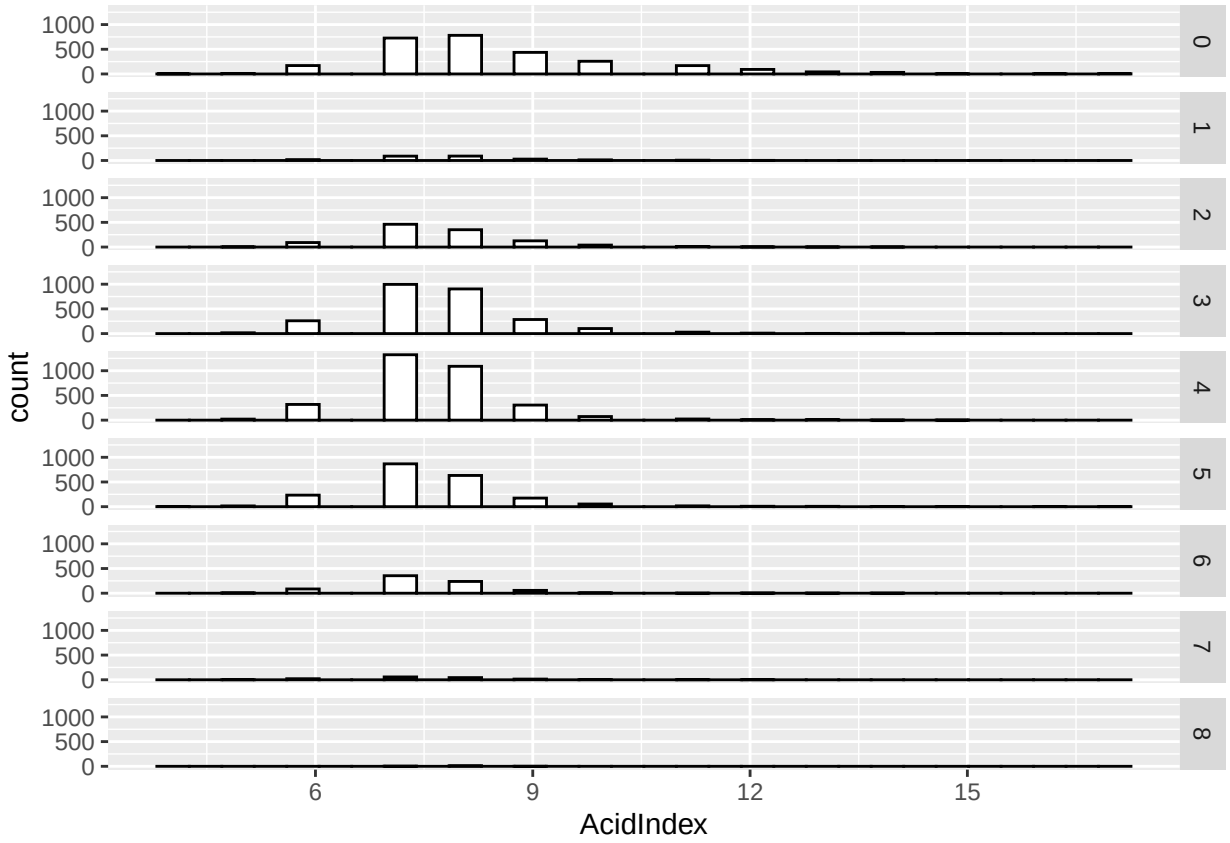
Some of the predictors variables are skewed and not normally distributed, in addition we have outliers and bimodality.

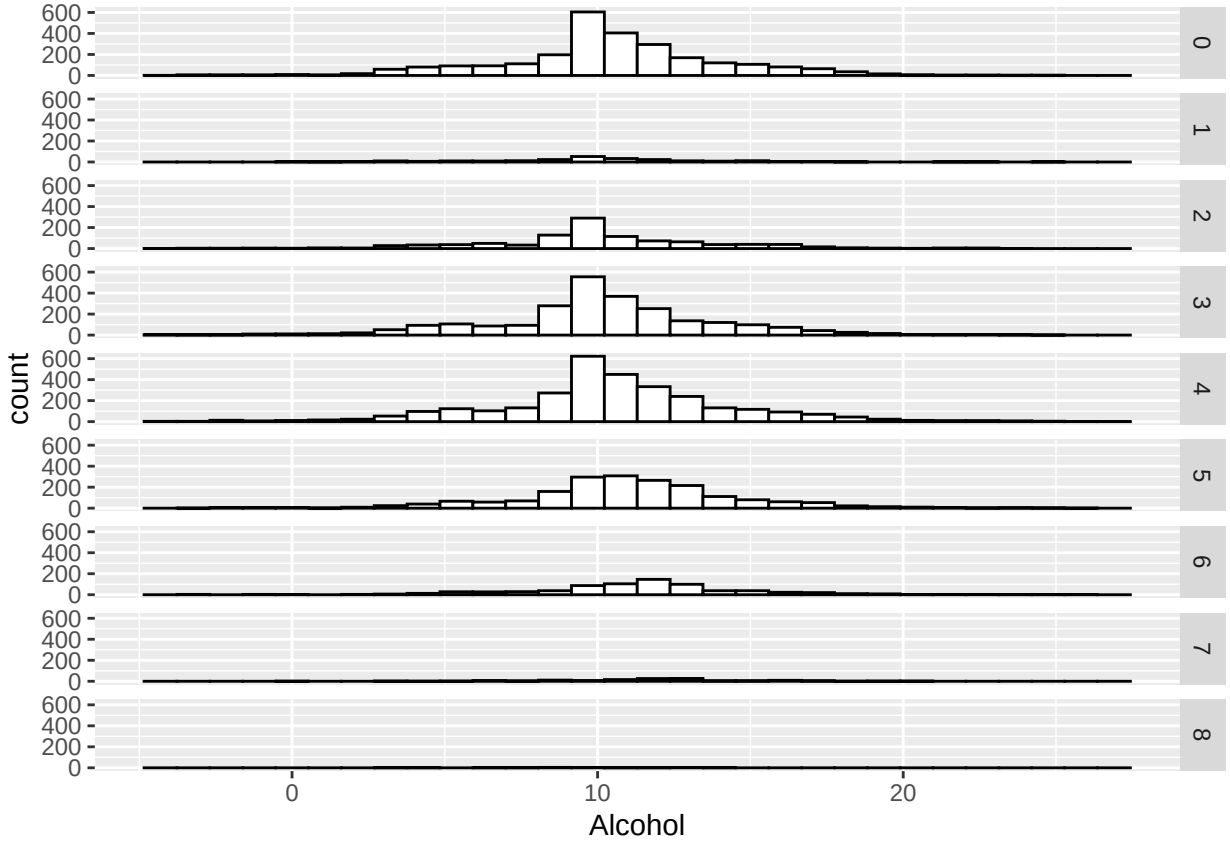


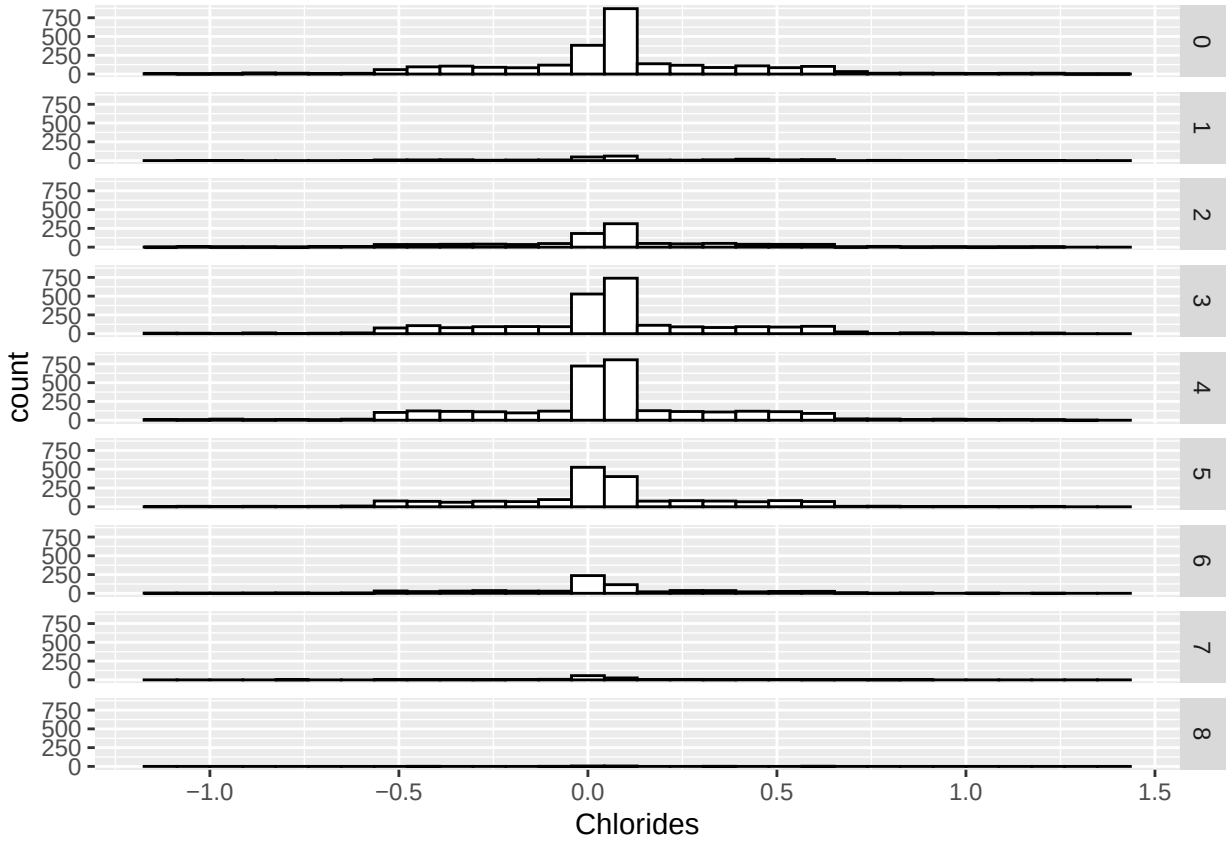
We take some of the variables to be analyzed separately against the target

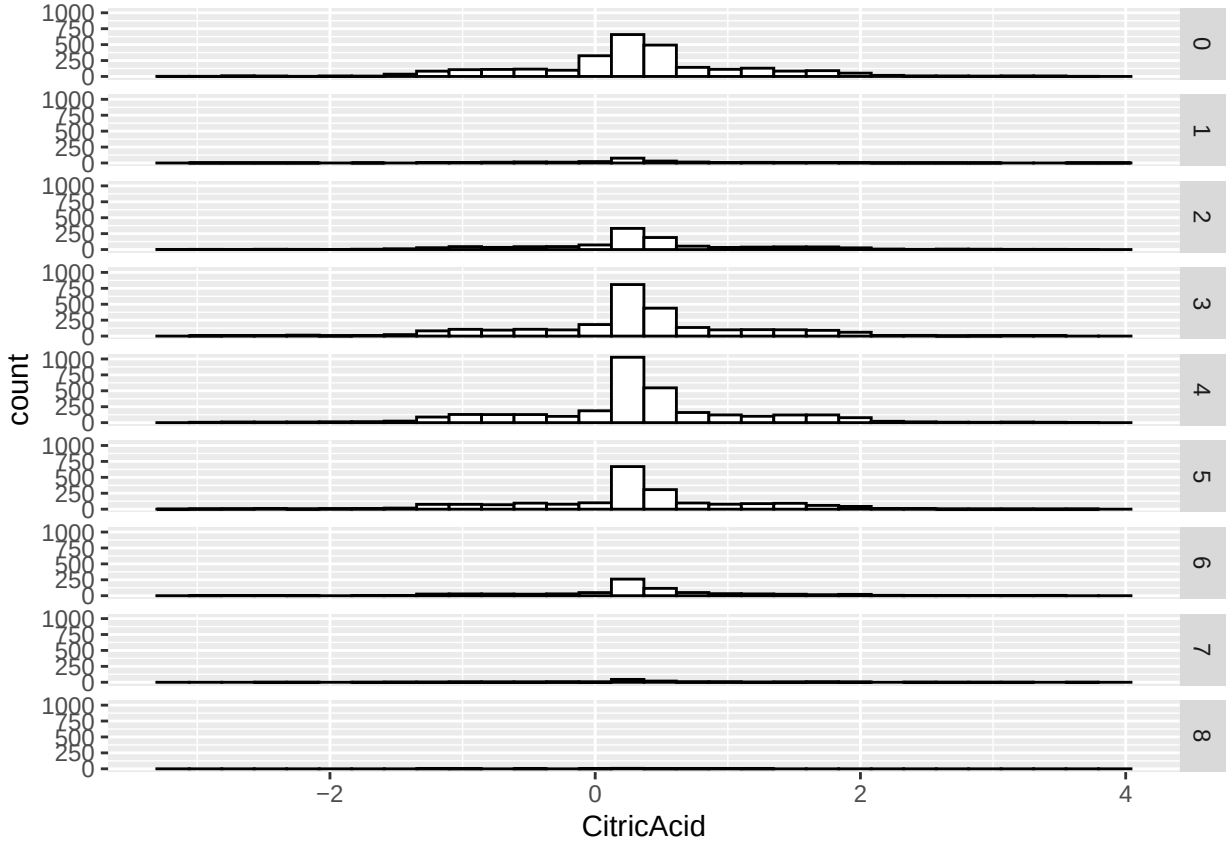
Skeweness

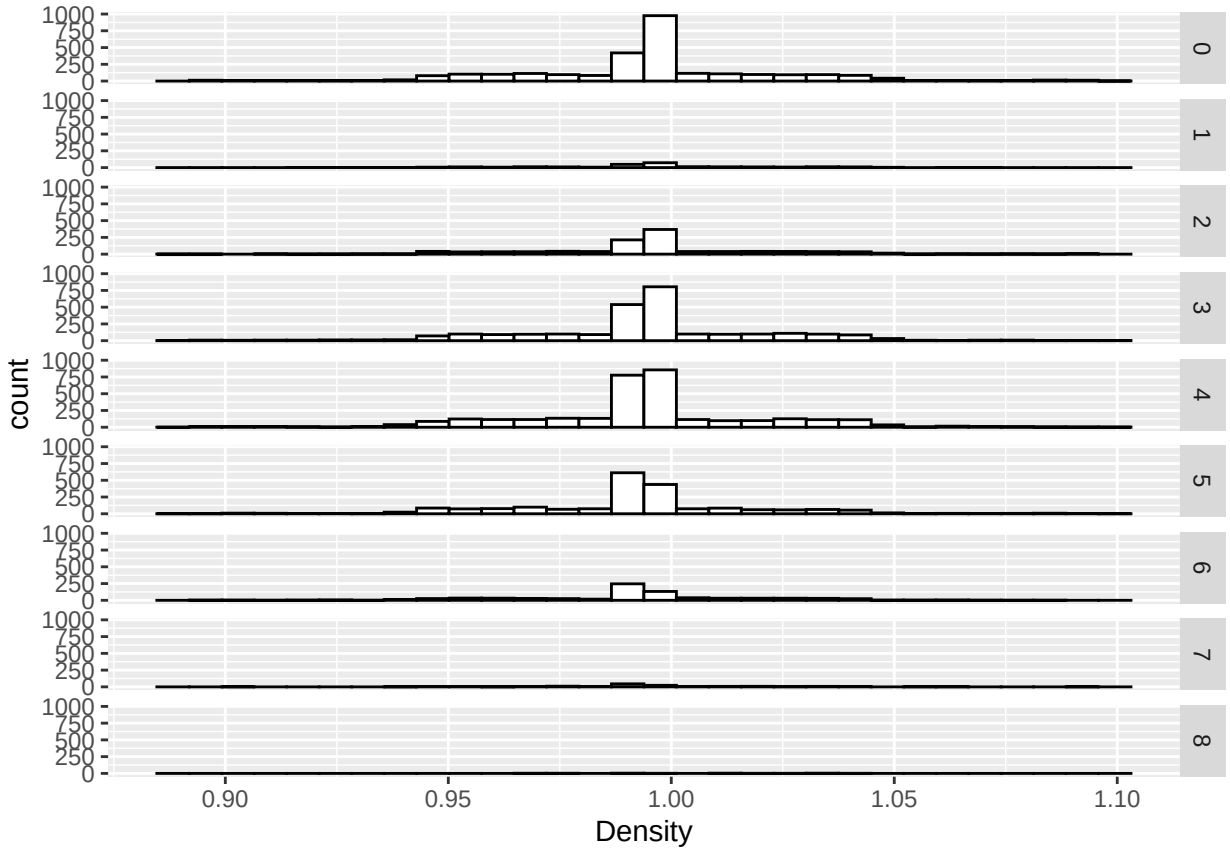
Histogram of predictors by factors of target

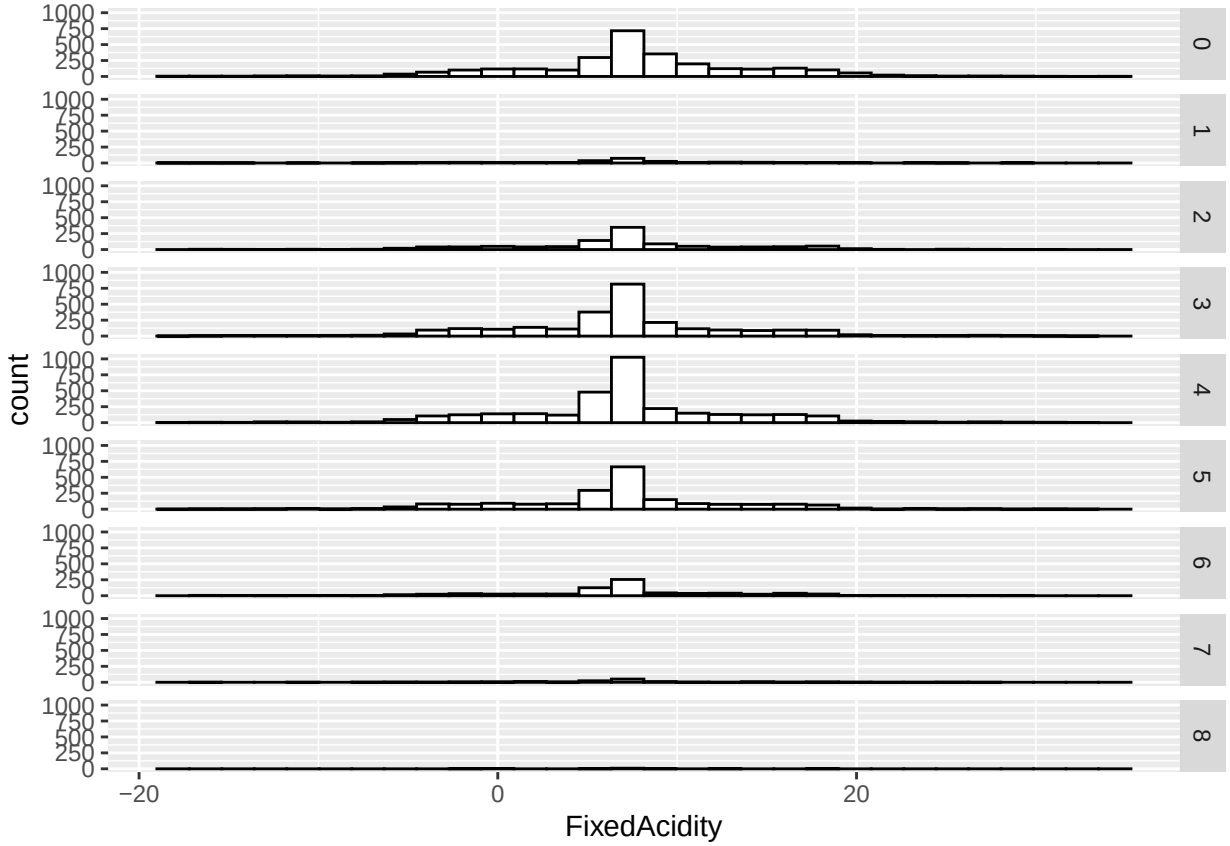


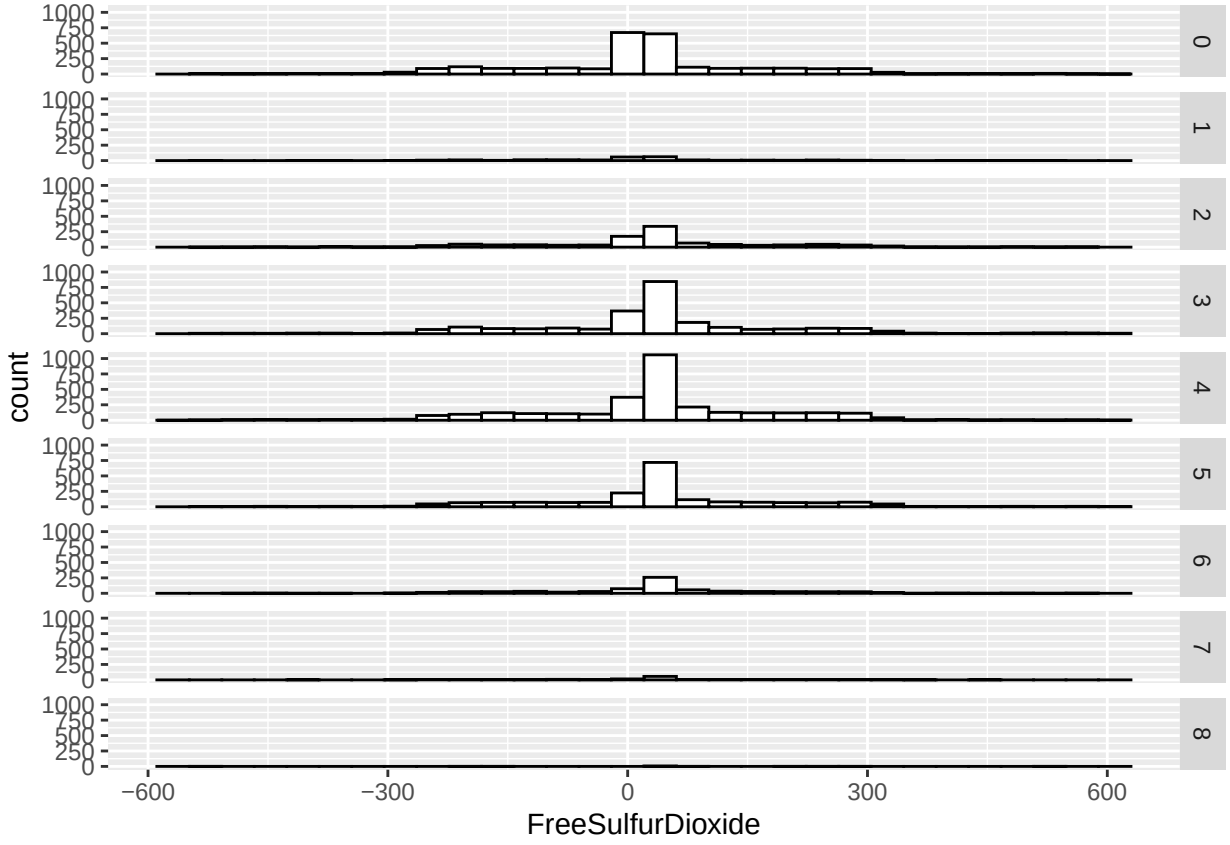

count
600
400
200
0
0
1
2
3
4
5
6
7
8
0
10
20
Alcohol

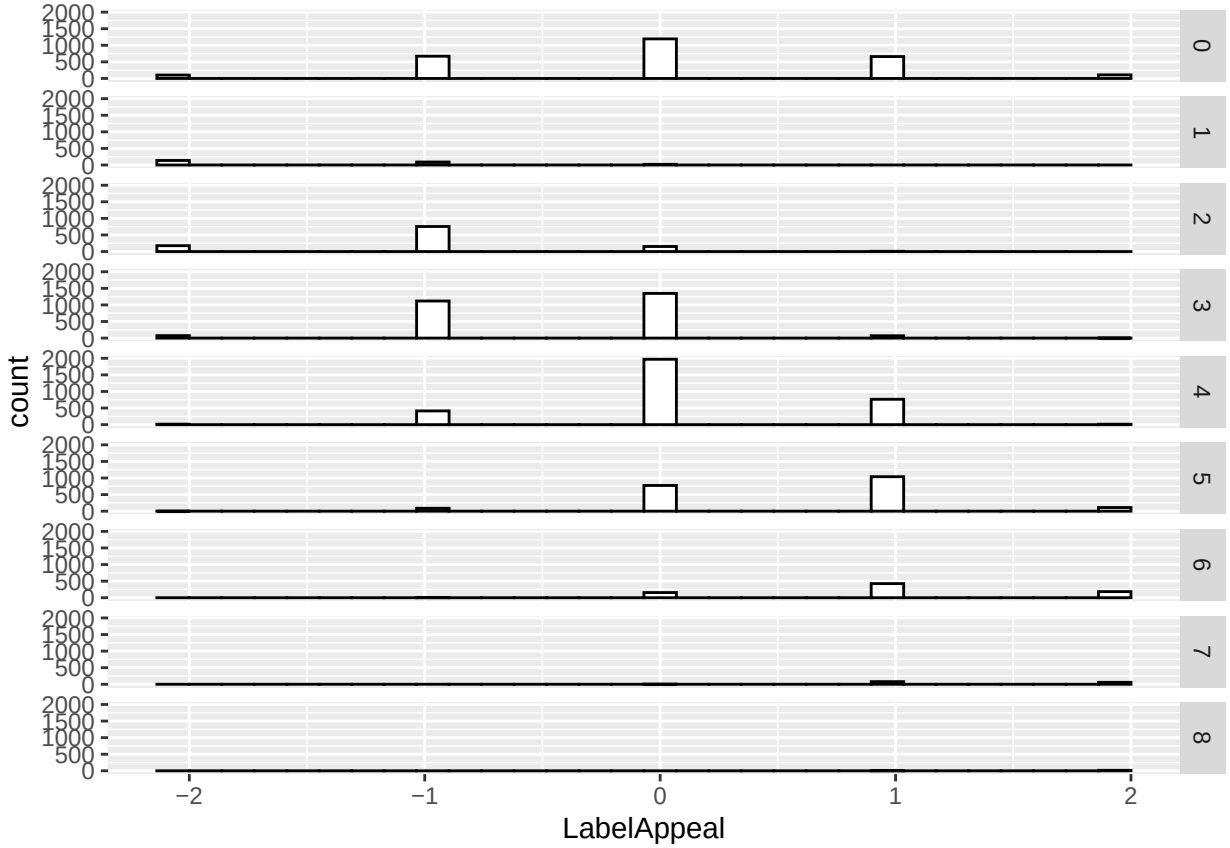

count
750
500
250
0
0
1
2
3
4
5
6
7
8
−1.0
−0.5
0.0
0.5
1.0
1.5
Chlorides

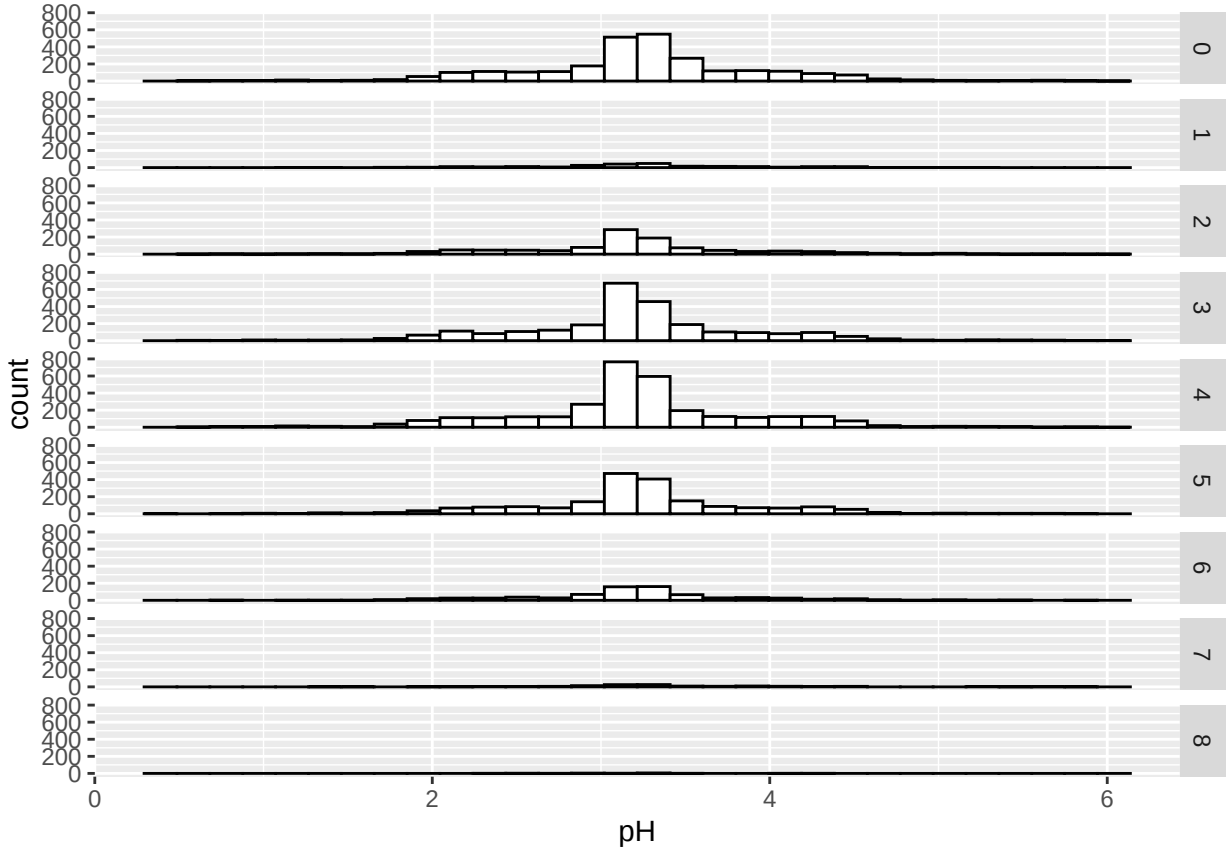

count
1000
750
500
250
0
0
1
2
3
4
5
6
7
8
−2
0
2
4
CitricAcid

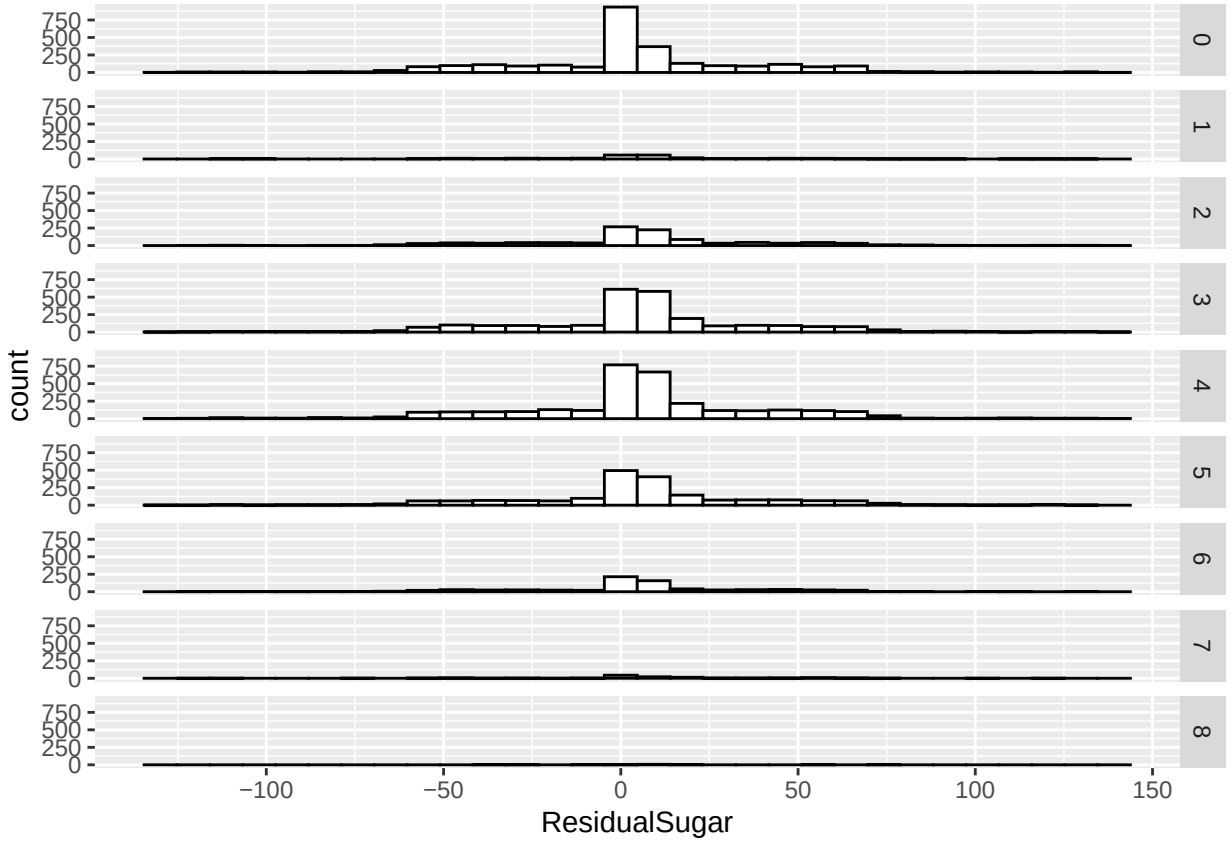

count
1000
750
500
250
0
0
1
2
3
4
5
6
7
8
0.90
0.95
1.00
1.05
1.10
Density

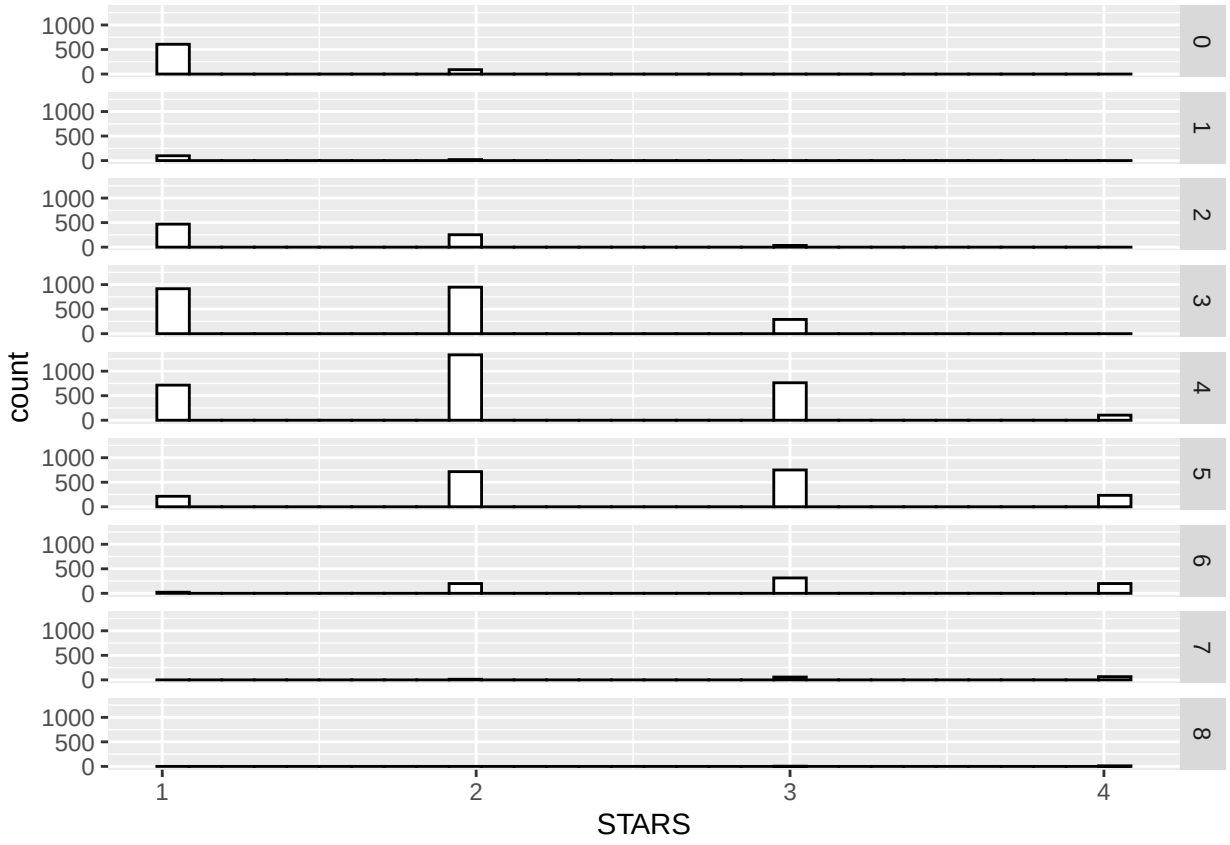

count
1000
750
500
250
0
0
1
2
3
4
5
6
7
8
−20
0
20
FixedAcidity

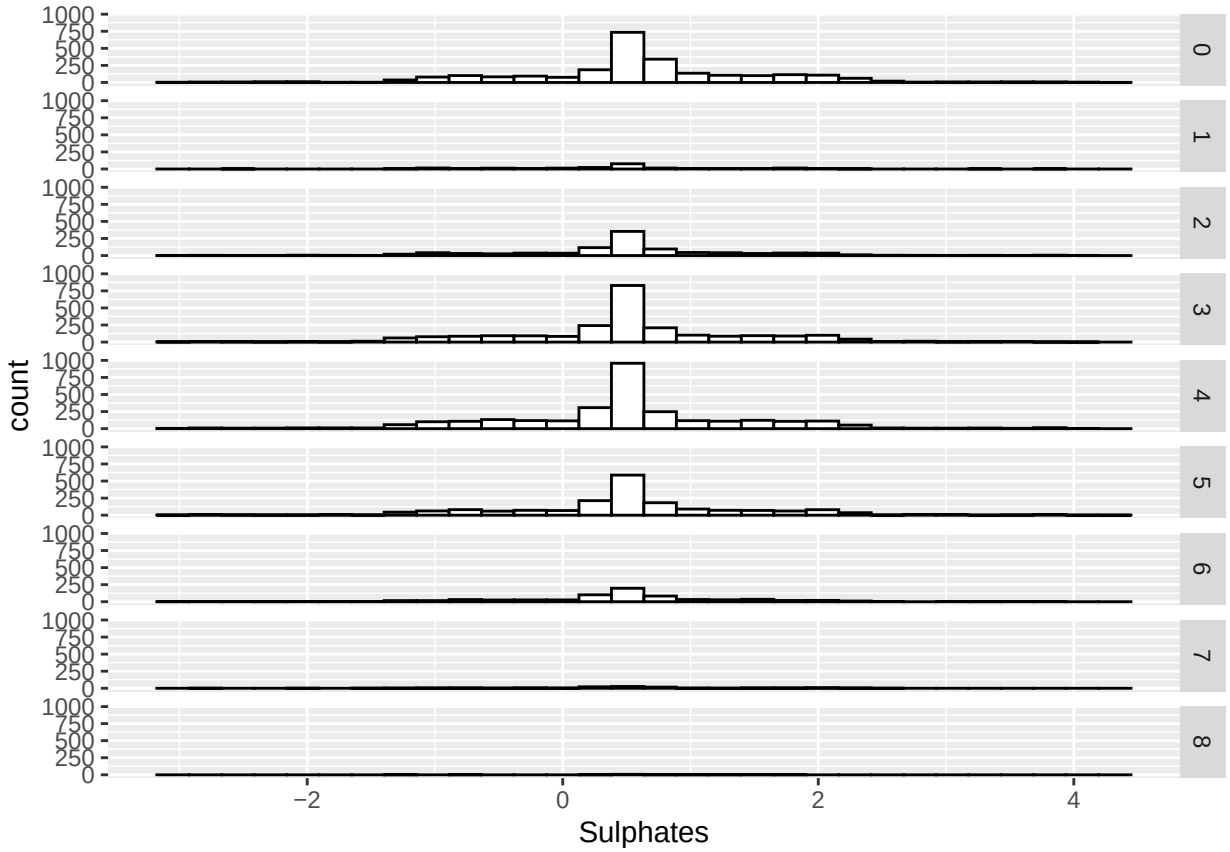

count
1000
750
500
250
0
0
1
2
3
4
5
6
7
8
−600
−300
0
300
600
FreeSulfurDioxide

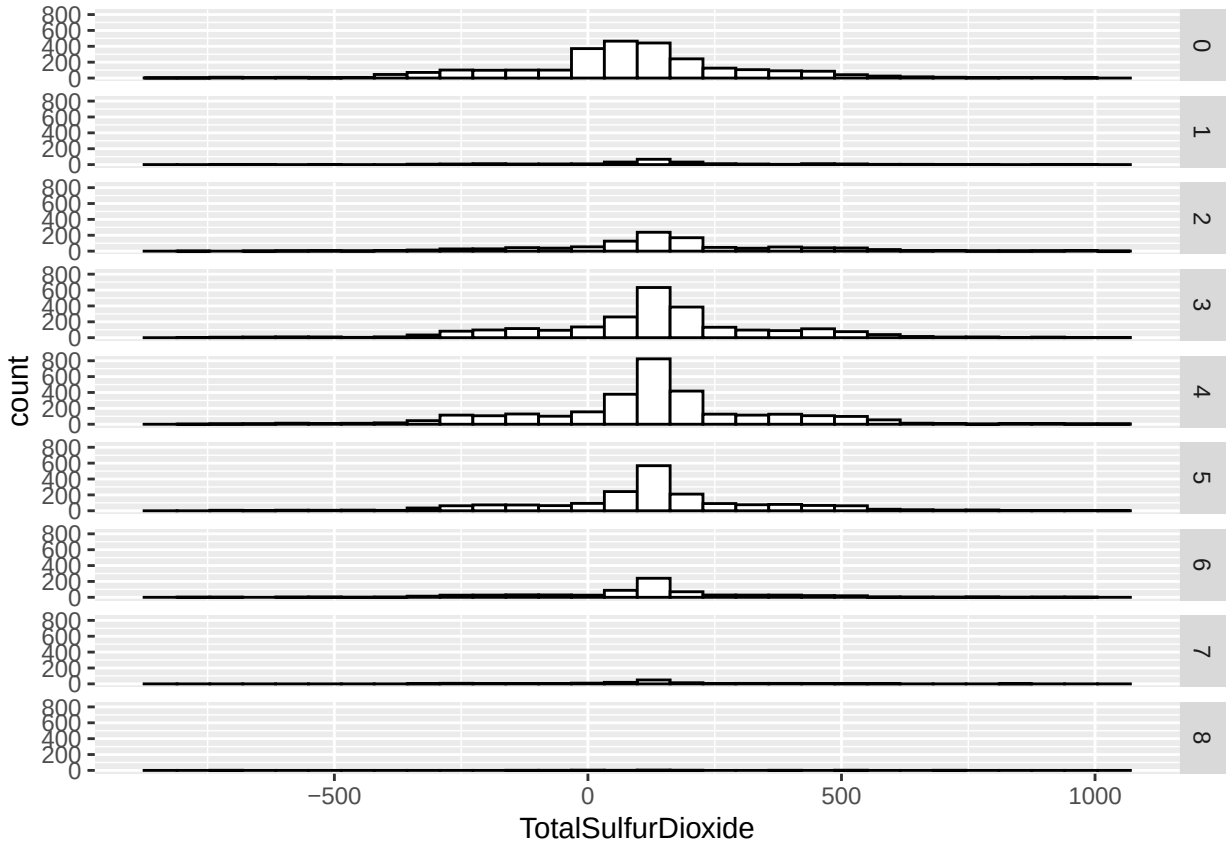

count
LabelAppeal
0
1
2
3
4
5
6
7
8
−2
−1
0
1
2

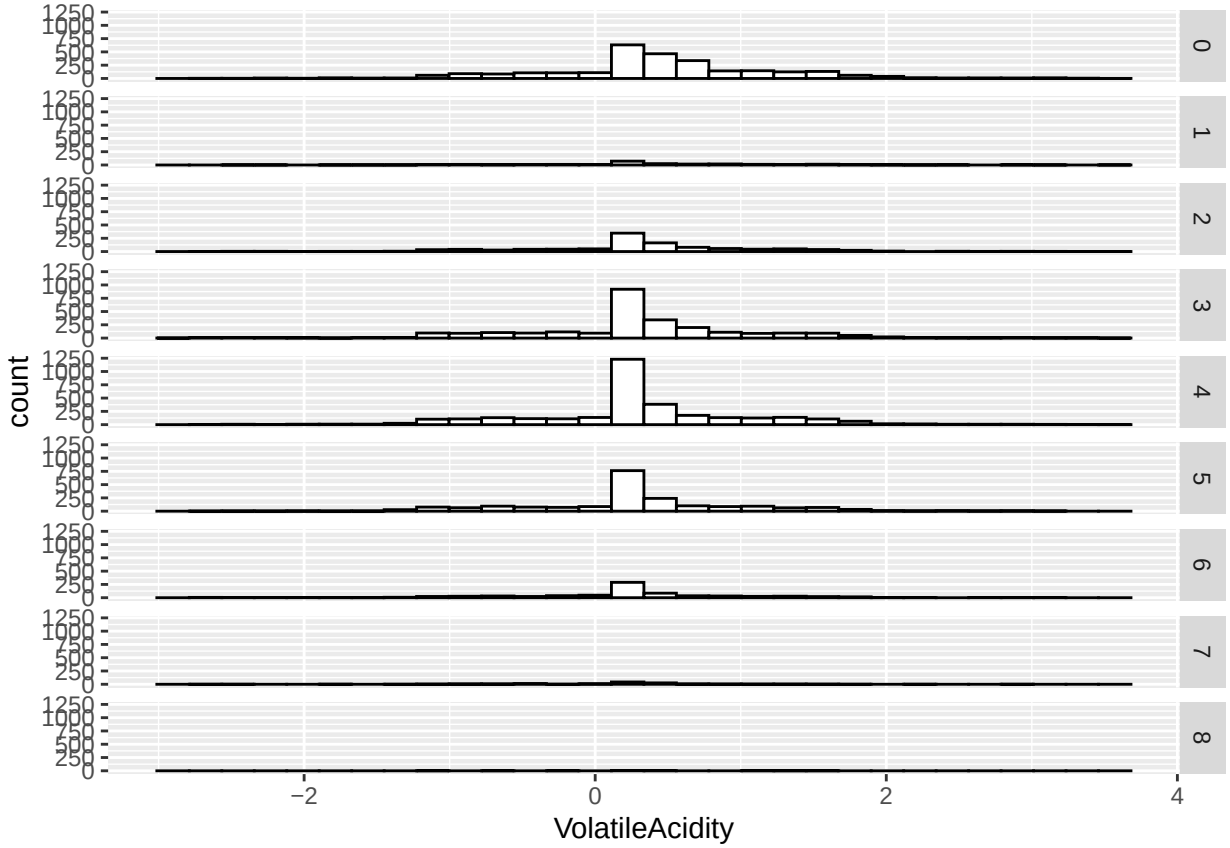

count
pH
0
1
2
3
4
5
6
7
8
0
2
4
6


count
750
500
250
0
0
1
2
3
4
5
6
7
8
−100
−50
0
50
100
150
ResidualSugar


count
1000
500
0
0
1
2
3
4
5
6
7
8
1
2
3
4
STARS

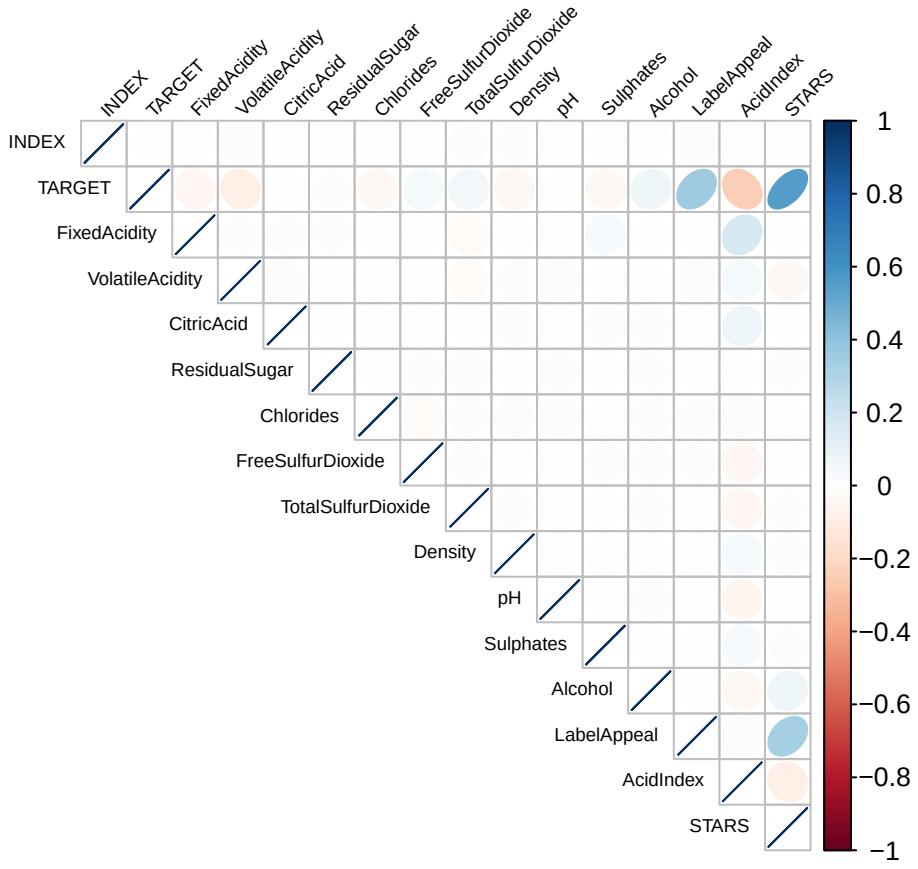

count
Sulphates
0
1
2
3
4
5
6
7
8
−2
0
2
4


count
TotalSulfurDioxide
0
1
2
3
4
5
6
7
8
−500
0
500
1000



Variable	Target : Number of cases purchased
Acid Index	
Alcohol	
Chlorides	
Citric Acid	
Density	
Fixed Acidity	
Free Sulfur Dioxide	
Label Appeal	
Residual Sugar	
STARS	
Sulphites	
Total Sulphur Dioxide	
Volatile Acidy	
pH	

Multicolinearity



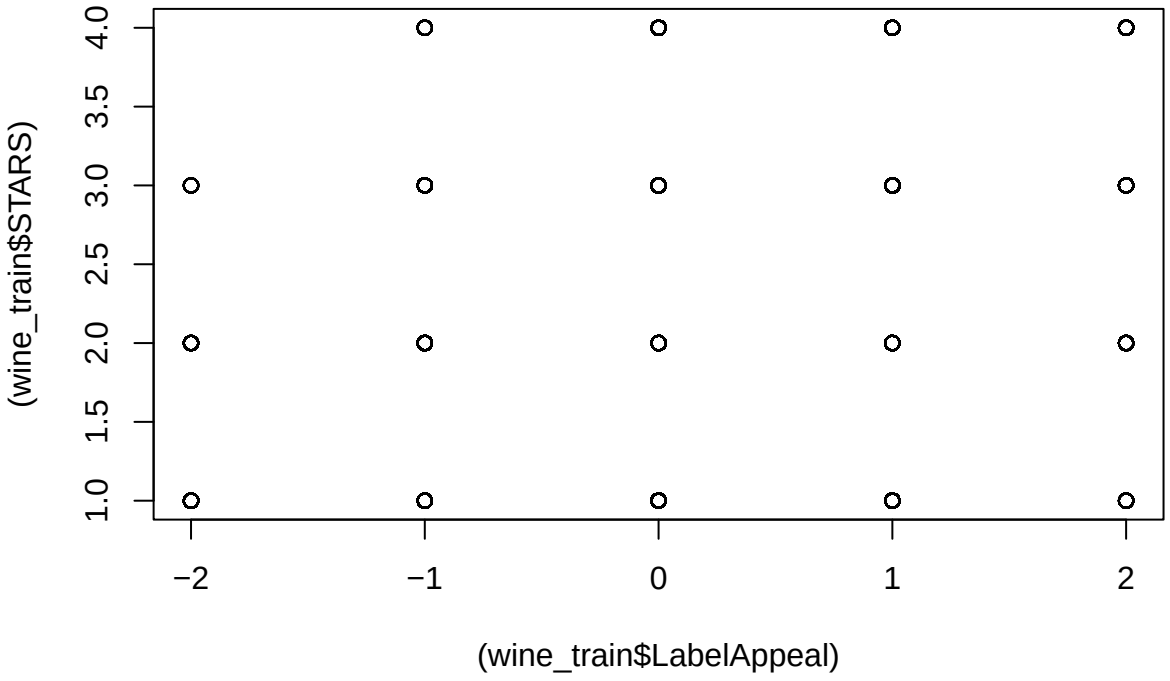
We can observe that there are some variables that have correlation between each other.

`LabelAppeal` and `STARTS` seem to have positive correlation. In addition, we can observe that `LabelAppeal` also has positive correlation with Alcohol.

There is slight negative correlation between `AcidIndex, STARS, TARGET`.

Correlation

Earlier we discovered the correlation between `LabelAppeal` and `STARS`. We want to understand this relationship better by plotting them.



The plot of correlation between rad and tax shoes 90% of the relationship is made by the influential points. This predictors are not really correlated.

```
##
##  Pearson's product-moment correlation
##
## data:  wine_train$LabelAppeal and wine_train$STARS
## t = 34.509, df = 9434, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3167501 0.3525834
## sample estimates:
##       cor
## 0.3347877
```

Regarding the strength of the relationship: The **more extreme** the correlation coefficient (the closer to -1 or 1), the **stronger the relationship**. This also means that a **correlation close to 0** indicates that the two variables are **independent**, that is, as one variable increases, there is no tendency in the other variable to either decrease or increase.

The p-value of the correlation test between these 2 variables is 2.2e-16. At the 3% significance level, we do not reject the null hypothesis of no correlation. We therefore conclude that we do not reject the hypothesis that there is no linear relationship between the 2 variables.

This test proves that even if the correlation coefficient is different from 0 (the correlation is 0.3 in the sample), it is actually not significantly different from 0 in the population.

The larger the sample size and the more extreme the correlation (closer to -1 or 1), the more likely the null hypothesis of no correlation will be rejected. With a small sample size, it is thus possible to obtain a *relatively* large correlation in the sample (based on the correlation coefficient), but still find a correlation not significantly different from 0 in the population (based on the correlation test). For this reason, it is recommended to always perform a correlation test before interpreting a correlation coefficient to avoid flawed conclusions.

2. Data Preparation

Missing Data

To prepare our data we have determined that there is some data that has missing values.

See below:

variable	total	isna	num.isna	pct
AcidIndex	12795	FALSE	12795	100.000000
Alcohol	12795	FALSE	12142	94.896444
Alcohol	12795	TRUE	653	5.103556
Chlorides	12795	FALSE	12157	95.013677
Chlorides	12795	TRUE	638	4.986323
CitricAcid	12795	FALSE	12795	100.000000
Density	12795	FALSE	12795	100.000000
FixedAcidity	12795	FALSE	12795	100.000000
FreeSulfurDioxide	12795	FALSE	12148	94.943337
FreeSulfurDioxide	12795	TRUE	647	5.056663
LabelAppeal	12795	FALSE	12795	100.000000
pH	12795	FALSE	12400	96.912857
pH	12795	TRUE	395	3.087143
ResidualSugar	12795	FALSE	12179	95.185619
ResidualSugar	12795	TRUE	616	4.814381
STARS	12795	FALSE	9436	73.747558
STARS	12795	TRUE	3359	26.252442
Sulphates	12795	FALSE	11585	90.543181
Sulphates	12795	TRUE	1210	9.456819
TARGET	12795	FALSE	12795	100.000000
TotalSulfurDioxide	12795	FALSE	12113	94.669793
TotalSulfurDioxide	12795	TRUE	682	5.330207
VolatileAcidity	12795	FALSE	12795	100.000000

Correlation

In order to determine the best predictor for our model we need to detect which are the predictor variables with low correlation value. We use the corrr package to determine all the variables with values <0.10. This will allow us to only manipulate the variables that have significance to our model.

```
##                  term TARGET
## 1               INDEX    .00
## 2        FixedAcidity   -.06
## 3     VolatileAcidity   -.11
## 4          CitricAcid    .02
## 5       ResidualSugar    .02
## 6           Chlorides   -.09
## 7   FreeSulfurDioxide    .08
## 8  TotalSulfurDioxide    .06
## 9             Density   -.08
## 10                 pH   -.01
## 11          Sulphates   -.05
## 12            Alcohol    .09
## 13        LabelAppeal    .43
## 14          AcidIndex   -.20
## 15              STARS    .56
```

Preprocess

```
## # A tibble: 9 x 3
##   TARGET count    prop
##    <dbl> <int>   <dbl>
## 1      0  2734 0.214
## 2      1   244 0.0191
## 3      2  1091 0.0853
## 4      3  2611 0.204
## 5      4  3177 0.248
## 6      5  2014 0.157
## 7      6   765 0.0598
## 8      7   142 0.0111
## 9      8    17 0.00133
```

Number of Cases Purchased	Target var codes	Percent Frequency
0	0	21%
1	1	2%
2	2	9%
3	3	20%
4	4	25%
5	5	16%
6	6	6%
7	7	1%
8	8	0.1%

Partition

The first thing we will do is to divide our training data into two parts: train set and test set. Our partition will be 70%, 30%

Caret provides us the `CreateDataPartition()` function for this, which will allow us to partition based on the proportion from the response variable.

Table 6: Data summary

Name	trainSet
Number of rows	8958
Number of columns	15
Column type frequency:	
numeric	15
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
TARGET	0	1.00	3.03	1.93	0.00	2.00	3.00	4.00	8.00
FixedAcidity	0	1.00	7.03	6.32	-18.00	5.10	6.90	9.40	34.40
VolatileAcidity	0	1.00	0.33	0.78	-2.79	0.13	0.28	0.63	3.68

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
CitricAcid	0	1.00	0.31	0.86	-3.24	0.03	0.31	0.59	3.77
ResidualSugar	437	0.95	5.26	33.96	-127.10	-2.90	3.90	15.90	141.15
Chlorides	434	0.95	0.05	0.32	-1.17	-0.03	0.05	0.15	1.35
FreeSulfurDioxide	461	0.95	30.91	148.15	-555.00	0.00	30.00	70.00	623.00
TotalSulfurDioxide	480	0.95	119.91	230.70	-793.00	27.00	122.00	208.00	1057.00
Density	0	1.00	0.99	0.03	0.89	0.99	0.99	1.00	1.10
pH	281	0.97	3.21	0.69	0.48	2.96	3.20	3.48	6.13
Sulphates	869	0.90	0.53	0.93	-3.12	0.28	0.50	0.86	4.24
Alcohol	458	0.95	10.52	3.73	-4.70	9.00	10.40	12.40	26.50
LabelAppeal	0	1.00	0.00	0.89	-2.00	-1.00	0.00	1.00	2.00
AcidIndex	0	1.00	7.77	1.31	4.00	7.00	8.00	8.00	17.00
STARS	2342	0.74	2.04	0.91	1.00	1.00	2.00	3.00	4.00

Imputation

We've seen that the dataset has few missing values across all columns, we may to do well to impute it.

A popular algorithm to do imputation is the k-Nearest Neighbors. This can be quickly and easily be done using caret. Because, caret offers a nice convenient `preProcess` function that can predict missing values besides other preprocessing. To predict the missing values with k-Nearest Neighbors using `preProcess()`:

```
## Created from 4497 samples and 14 variables
##
## Pre-processing:
##   - bagged tree imputation (14)
##   - ignored (0)
```

Table 8: Data summary

Name	trainSet
Number of rows	8958
Number of columns	15
Column type frequency:	
numeric	15
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
TARGET	0	1	3.03	1.93	0.00	2.00	3.00	4.00	8.00
FixedAcidity	0	1	7.03	6.32	-18.00	5.10	6.90	9.40	34.40
VolatileAcidity	0	1	0.33	0.78	-2.79	0.13	0.28	0.63	3.68
CitricAcid	0	1	0.31	0.86	-3.24	0.03	0.31	0.59	3.77
ResidualSugar	0	1	5.24	33.12	-127.10	0.66	4.90	14.90	141.15
Chlorides	0	1	0.05	0.31	-1.17	0.00	0.05	0.13	1.35
FreeSulfurDioxide	0	1	31.08	144.29	-555.00	5.00	33.00	64.00	623.00
TotalSulfurDioxide	0	1	120.13	224.43	-793.00	34.00	124.05	198.00	1057.00
Density	0	1	0.99	0.03	0.89	0.99	0.99	1.00	1.10
pH	0	1	3.21	0.68	0.48	2.98	3.21	3.47	6.13
Sulphates	0	1	0.53	0.88	-3.12	0.34	0.51	0.77	4.24
Alcohol	0	1	10.53	3.63	-4.70	9.10	10.50	12.20	26.50
LabelAppeal	0	1	0.00	0.89	-2.00	-1.00	0.00	1.00	2.00
AcidIndex	0	1	7.77	1.31	4.00	7.00	8.00	8.00	17.00
STARS	0	1	2.02	0.79	1.00	1.64	2.00	2.38	4.00

Normalization

Typically we **normalize** data when performing some type of analysis in which we have multiple variables that are measured on different scales and we want each of the variables to have the same range.

Table 10: Data summary

Name	trainSet
Number of rows	8958
Number of columns	15
Column type frequency:	
numeric	15
Group variables	None

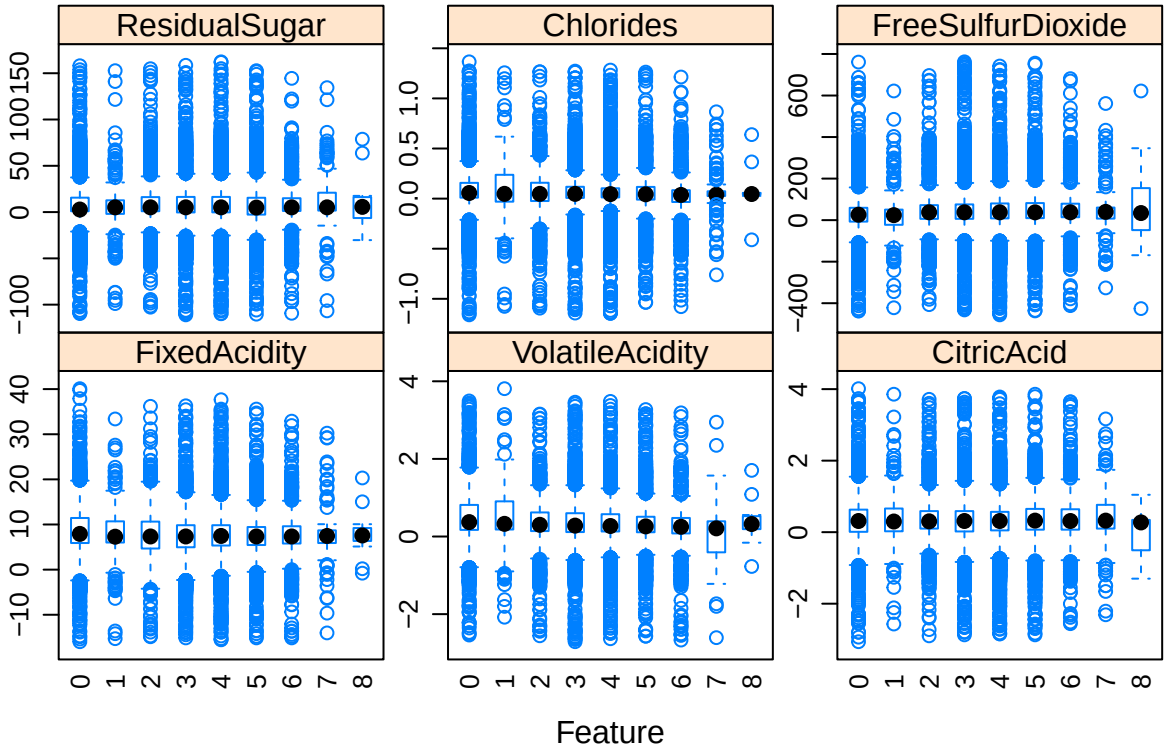
Variable type: numeric

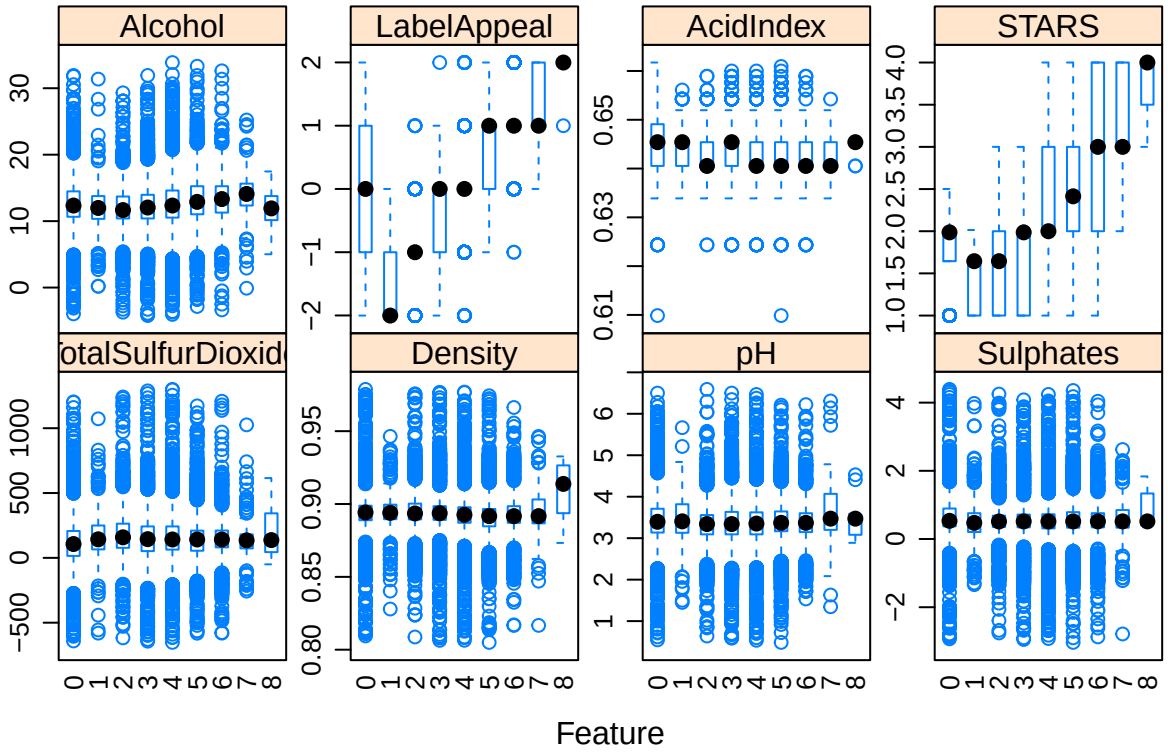
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
TARGET	0	1	3.03	1.93	0.00	2.00	3.00	4.00	8.00
FixedAcidity	0	1	7.82	6.92	-15.98	5.45	7.46	10.30	40.09
VolatileAcidity	0	1	0.34	0.79	-2.71	0.13	0.28	0.64	3.81
CitricAcid	0	1	0.33	0.87	-3.06	0.03	0.31	0.60	4.01
ResidualSugar	0	1	7.48	33.86	-111.02	0.67	5.10	15.95	162.32
Chlorides	0	1	0.05	0.31	-1.16	0.00	0.05	0.13	1.37
FreeSulfurDioxide	0	1	47.15	151.32	-456.11	5.22	36.38	72.17	762.25
TotalSulfurDioxide	0	1	153.46	244.50	-652.22	37.28	141.81	229.91	1299.32
Density	0	1	0.89	0.02	0.80	0.89	0.89	0.90	0.98
pH	0	1	3.38	0.74	0.49	3.12	3.37	3.66	6.59
Sulphates	0	1	0.54	0.89	-3.03	0.34	0.52	0.78	4.39
Alcohol	0	1	12.54	4.59	-4.21	10.65	12.43	14.61	33.87
LabelAppeal	0	1	0.00	0.89	-2.00	-1.00	0.00	1.00	2.00
AcidIndex	0	1	0.64	0.01	0.61	0.64	0.65	0.65	0.66
STARS	0	1	2.02	0.79	1.00	1.64	2.00	2.38	4.00

Features

Importance of Features

Now that the preprocessing is complete, let's visually examine how the predictors influence the Y (Target - Number of Cases Purchased). In this problem, the X variables are numeric whereas the Y is categorical. So how to gauge if a given X is an important predictor of Y? A simple common sense approach is, if you group the X variable by the categories of Y, a significant mean shift among the X's groups is a strong indicator (if not the only indicator) that X will have a significant role to help predict Y. It is possible to watch this shift visually using box plots and density plots. In fact, caret's `featurePlot()` function makes it so convenient. Simply set the X and Y parameters and set `plot='box'`. You can additionally adjust the label font size (using `strip`) and the scales to be free as I have done in the below plot.





Let me quickly refresh how to interpret a boxplot. Each subplot in the above figure has eight (8) boxplots (in blue) inside it, one each for each of the Y categories, 1-8 purchased boxex. The top of the box represents the 25th %ile and the bottom of the box represents the 75th %ile.

The black dot inside the box is the mean. The blue box represents the region where most of the regular data point lie. The subplots also show many blue dots lying outside the top and bottom dashed lines called whiskers. These dots are formally considered as extreme values. What other predictors do you notice have significant mean differences?`LabelAppleal` is the only one with significant mean differences.

3. Building Models

- Dependent Variable: Whether or not the crime rate is above median crime rate
- Independent variables: all predictors described earlier.

Poisson 1

We will begin our first model using all the predictors without being transformed to see the level of significance of each one of them. This model will include all original values without any transformation.

Poisson with Raw Data

```
##
## Call:
## glm(formula = TARGET ~ ., family = "poisson", data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2158  -0.2734   0.0616   0.3732   1.6830
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          1.593e+00  2.506e-01   6.359 2.03e-10 ***
## FixedAcidity         3.293e-04  1.053e-03   0.313  0.75447
## VolatileAcidity     -2.560e-02  8.353e-03  -3.065  0.00218 **
## CitricAcid          -7.259e-04  7.575e-03  -0.096  0.92365
## ResidualSugar       -6.141e-05  1.941e-04  -0.316  0.75165
## Chlorides           -3.007e-02  2.056e-02  -1.463  0.14346
## FreeSulfurDioxide    6.734e-05  4.404e-05   1.529  0.12620
## TotalSulfurDioxide   2.081e-05  2.855e-05   0.729  0.46618
## Density             -3.725e-01  2.462e-01  -1.513  0.13026
## pH                  -4.661e-03  9.598e-03  -0.486  0.62722
## Sulphates           -5.164e-03  7.051e-03  -0.732  0.46398
## Alcohol              3.948e-03  1.771e-03   2.229  0.02579 *
## LabelAppeal          1.771e-01  7.954e-03  22.271  < 2e-16 ***
## AcidIndex           -4.870e-02  5.903e-03  -8.251  < 2e-16 ***
## STARS                1.871e-01  7.487e-03  24.993  < 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: 5844.1  on 6435  degrees of freedom
## Residual deviance: 4009.1  on 6421  degrees of freedom
##   (6359 observations deleted due to missingness)
## AIC: 23172
##
## Number of Fisher Scoring iterations: 5
```

- The deviance can be used as a goodness-of-fit.
- We test H0: ' the model is appropiate' vs. h1: The model is not appropiate'

p-value of Residual Deviance goodness-of-fit test

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

Pearson Goodness-of-fit

```
## [1] 0
```

Poisson 2

Poisson with Transformed Data

```
##
## Call:
## glm(formula = TARGET ~ ., family = "poisson", data = trainSet)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.7038  -0.5049   0.2254   0.6436   2.1808
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          1.651e+01  7.805e-01  21.151  < 2e-16 ***
## FixedAcidity        -1.567e-03  8.967e-04  -1.748 0.080543 .
## VolatileAcidity     -5.477e-02  7.718e-03  -7.097 1.28e-12 ***
## CitricAcid           1.125e-02  6.967e-03   1.615 0.106274
## ResidualSugar       -1.082e-04  1.798e-04  -0.602 0.547116
## Chlorides           -5.788e-02  1.986e-02  -2.914 0.003571 **
## FreeSulfurDioxide    1.501e-04  3.986e-05   3.766 0.000166 ***
## TotalSulfurDioxide   1.097e-04  2.479e-05   4.426 9.60e-06 ***
## Density             -6.305e-01  2.816e-01  -2.239 0.025147 *
## pH                  -1.830e-02  8.333e-03  -2.196 0.028127 *
## Sulphates           -1.487e-02  6.813e-03  -2.183 0.029041 *
## Alcohol              3.364e-03  1.329e-03   2.531 0.011386 *
## LabelAppeal          1.857e-01  7.437e-03  24.964  < 2e-16 ***
## AcidIndex           -2.375e+01  1.164e+00 -20.399  < 2e-16 ***
## STARS                2.114e-01  7.870e-03  26.860  < 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: 16108  on 8957  degrees of freedom
## Residual deviance: 13257  on 8943  degrees of freedom
## AIC: 35619
##
## Number of Fisher Scoring iterations: 5
```

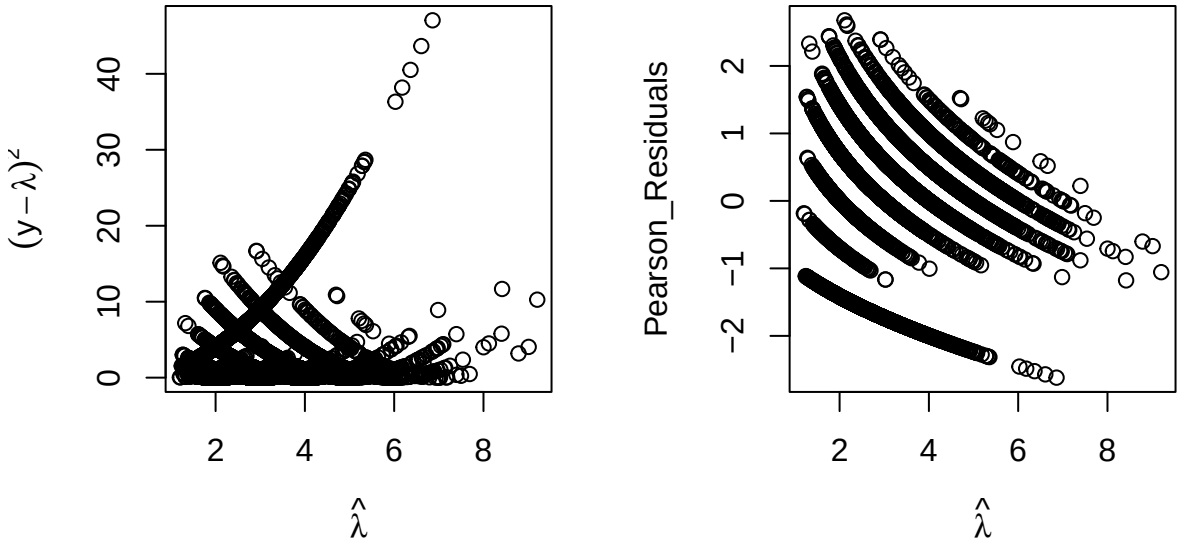
- The deviance can be used as a goodness-of-fit.
- We test H0: ' the model is appropiate' vs. h1: The model is not appropiate'

p-value of Residual Deviance goodness-of-fit test

```
## [1] 0
```

```
## [1] 0.9680968
```

Mean = Variance Assumption



From the first graph we can see that the range of the variance differs from the range of the mean. Moreover, from the second graph, we see that the residuals show some kind of pattern thus mean = var seems not to hold. We will examine the dispersion of the data and try a Quasipoisson in case of overdispersion.

Assesing Overdispersion

The variance of Y must be somewhat proportional to its mean. We can write

$$var(Y) = E[Y] = \theta\lambda$$

if $\theta > 1$, the data are overdisperesed and if $\theta < 1$, the data is underdisperesed. If a Poisson model is fitted under overdispersion of the response, then the standard errors of the estimated coefficients are underestimated.

```
## [1] 0.9724635
```

Quassipoisson

```
##
## Call:
## glm(formula = TARGET ~ ., family = "quasipoisson", data = trainSet)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.7038  -0.5049   0.2254   0.6436   2.1808
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.651e+01  7.697e-01  21.448  < 2e-16 ***
```

```
## FixedAcidity       -1.567e-03  8.843e-04  -1.772 0.076410 .
## VolatileAcidity    -5.477e-02  7.611e-03  -7.197 6.67e-13 ***
## CitricAcid          1.125e-02  6.870e-03   1.638 0.101482
## ResidualSugar      -1.082e-04  1.773e-04  -0.611 0.541512
## Chlorides          -5.788e-02  1.959e-02  -2.955 0.003138 **
## FreeSulfurDioxide   1.501e-04  3.931e-05   3.819 0.000135 ***
## TotalSulfurDioxide  1.097e-04  2.444e-05   4.488 7.27e-06 ***
## Density            -6.305e-01  2.777e-01  -2.271 0.023194 *
## pH                 -1.830e-02  8.218e-03  -2.226 0.026014 *
## Sulphates          -1.487e-02  6.718e-03  -2.214 0.026880 *
## Alcohol             3.364e-03  1.311e-03   2.566 0.010298 *
## LabelAppeal         1.857e-01  7.334e-03  25.315  < 2e-16 ***
## AcidIndex          -2.375e+01  1.148e+00 -20.686  < 2e-16 ***
## STARS               2.114e-01  7.761e-03  27.238  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 0.9724643)
##
##     Null deviance: 16108  on 8957  degrees of freedom
## Residual deviance: 13257  on 8943  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

p-value of Residual Deviance goodness-of-fit test

```
## [1] 0
```

Variable Selection using BIC

Some variables may not be relevant to the model or have low explanatory power. Stepwise model selection provides one possible solution to select our covariates based on AIC or BIC reduction (not available for Quassi)

```
##
## Attaching package: 'MASS'

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

## Start:  AIC=35725.79
## TARGET ~ FixedAcidity + VolatileAcidity + CitricAcid + ResidualSugar +
##     Chlorides + FreeSulfurDioxide + TotalSulfurDioxide + Density +
##     pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS
##
##                       Df Deviance   AIC
## - ResidualSugar        1    13257 35717
## - CitricAcid           1    13260 35719
## - FixedAcidity         1    13260 35720
## - Sulphates            1    13262 35721
## - pH                   1    13262 35722
## - Density              1    13262 35722
## - Alcohol              1    13263 35723
## - Chlorides            1    13266 35725
```

```
## <none>                     13257 35726
## - FreeSulfurDioxide   1    13271 35731
## - TotalSulfurDioxide  1    13276 35736
## - VolatileAcidity     1    13307 35767
## - AcidIndex           1    13669 36129
## - LabelAppeal         1    13880 36340
## - STARS               1    13967 36427
##
## Step:  AIC=35717.05
## TARGET ~ FixedAcidity + VolatileAcidity + CitricAcid + Chlorides +
##     FreeSulfurDioxide + TotalSulfurDioxide + Density + pH + Sulphates +
##     Alcohol + LabelAppeal + AcidIndex + STARS
##
##                       Df Deviance   AIC
## - CitricAcid           1    13260 35711
## - FixedAcidity         1    13260 35711
## - Sulphates            1    13262 35713
## - pH                   1    13262 35713
## - Density              1    13262 35713
## - Alcohol              1    13264 35714
## - Chlorides            1    13266 35716
## <none>                      13257 35717
## - FreeSulfurDioxide    1    13271 35722
## + ResidualSugar        1    13257 35726
## - TotalSulfurDioxide   1    13277 35727
## - VolatileAcidity      1    13308 35758
## - AcidIndex            1    13669 36120
## - LabelAppeal          1    13880 36331
## - STARS                1    13967 36418
##
## Step:  AIC=35710.58
## TARGET ~ FixedAcidity + VolatileAcidity + Chlorides + FreeSulfurDioxide +
##     TotalSulfurDioxide + Density + pH + Sulphates + Alcohol +
##     LabelAppeal + AcidIndex + STARS
##
##                       Df Deviance   AIC
## - FixedAcidity         1    13263 35704
## - pH                   1    13265 35706
## - Sulphates            1    13265 35706
## - Density              1    13265 35707
## - Alcohol              1    13267 35708
## - Chlorides            1    13269 35710
## <none>                      13260 35711
## - FreeSulfurDioxide    1    13274 35716
## + CitricAcid           1    13257 35717
## + ResidualSugar        1    13260 35719
## - TotalSulfurDioxide   1    13280 35721
## - VolatileAcidity      1    13310 35752
## - AcidIndex            1    13669 36111
## - LabelAppeal          1    13884 36325
## - STARS                1    13970 36411
##
## Step:  AIC=35704.41
## TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide + TotalSulfurDioxide +
##     Density + pH + Sulphates + Alcohol + LabelAppeal + AcidIndex +
##     STARS
```

```
##
##                       Df Deviance   AIC
## - pH                   1    13268 35700
## - Sulphates            1    13268 35700
## - Density              1    13268 35700
## - Alcohol              1    13270 35702
## - Chlorides            1    13272 35704
## <none>                      13263 35704
## - FreeSulfurDioxide    1    13277 35709
## + FixedAcidity         1    13260 35711
## + CitricAcid           1    13260 35711
## + ResidualSugar        1    13262 35713
## - TotalSulfurDioxide   1    13283 35715
## - VolatileAcidity      1    13314 35746
## - AcidIndex            1    13694 36126
## - LabelAppeal          1    13886 36319
## - STARS                1    13972 36404
##
## Step:  AIC=35700.29
## TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide + TotalSulfurDioxide +
##     Density + Sulphates + Alcohol + LabelAppeal + AcidIndex +
##     STARS
##
##                       Df Deviance   AIC
## - Sulphates            1    13273 35696
## - Density              1    13273 35696
## - Alcohol              1    13275 35698
## - Chlorides            1    13276 35699
## <none>                      13268 35700
## + pH                   1    13263 35704
## - FreeSulfurDioxide    1    13282 35705
## + FixedAcidity         1    13265 35706
## + CitricAcid           1    13265 35707
## + ResidualSugar        1    13268 35709
## - TotalSulfurDioxide   1    13288 35711
## - VolatileAcidity      1    13319 35742
## - AcidIndex            1    13695 36118
## - LabelAppeal          1    13890 36313
## - STARS                1    13978 36401
##
## Step:  AIC=35696.38
## TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide + TotalSulfurDioxide +
##     Density + Alcohol + LabelAppeal + AcidIndex + STARS
##
##                       Df Deviance   AIC
## - Density              1    13278 35692
## - Alcohol              1    13280 35694
## - Chlorides            1    13281 35696
## <none>                      13273 35696
## + Sulphates            1    13268 35700
## + pH                   1    13268 35700
## - FreeSulfurDioxide    1    13287 35701
## + FixedAcidity         1    13270 35702
## + CitricAcid           1    13270 35703
## + ResidualSugar        1    13273 35705
## - TotalSulfurDioxide   1    13293 35707
```

```
## - VolatileAcidity     1    13324 35738
## - AcidIndex           1    13702 36117
## - LabelAppeal         1    13894 36308
## - STARS               1    13984 36398
##
## Step:  AIC=35692.48
## TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide + TotalSulfurDioxide +
##     Alcohol + LabelAppeal + AcidIndex + STARS
##
##                      Df Deviance   AIC
## - Alcohol             1    13285 35690
## - Chlorides           1    13287 35692
## <none>                     13278 35692
## + pH                  1    13273 35696
## + Density             1    13273 35696
## + Sulphates           1    13273 35696
## - FreeSulfurDioxide   1    13292 35697
## + FixedAcidity        1    13275 35698
## + CitricAcid          1    13276 35699
## + ResidualSugar       1    13278 35701
## - TotalSulfurDioxide  1    13298 35703
## - VolatileAcidity     1    13329 35734
## - AcidIndex           1    13714 36119
## - LabelAppeal         1    13899 36304
## - STARS               1    13991 36396
##
## Step:  AIC=35690.22
## TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide + TotalSulfurDioxide +
##     LabelAppeal + AcidIndex + STARS
##
##                      Df Deviance   AIC
## - Chlorides           1    13294 35690
## <none>                     13285 35690
## + Alcohol             1    13278 35692
## + pH                  1    13280 35694
## + Density             1    13280 35694
## + Sulphates           1    13280 35694
## - FreeSulfurDioxide   1    13299 35695
## + FixedAcidity        1    13282 35696
## + CitricAcid          1    13282 35696
## + ResidualSugar       1    13285 35699
## - TotalSulfurDioxide  1    13304 35700
## - VolatileAcidity     1    13336 35732
## - AcidIndex           1    13727 36123
## - LabelAppeal         1    13904 36300
## - STARS               1    14011 36407
##
## Step:  AIC=35689.77
## TARGET ~ VolatileAcidity + FreeSulfurDioxide + TotalSulfurDioxide +
##     LabelAppeal + AcidIndex + STARS
##
##                      Df Deviance   AIC
## <none>                     13294 35690
## + Chlorides           1    13285 35690
## + Alcohol             1    13287 35692
## + Density             1    13288 35693
```

```
## + pH                 1    13289 35694
## + Sulphates          1    13289 35694
## - FreeSulfurDioxide  1    13308 35695
## + FixedAcidity       1    13290 35696
## + CitricAcid         1    13291 35696
## + ResidualSugar      1    13293 35698
## - TotalSulfurDioxide 1    13313 35700
## - VolatileAcidity    1    13345 35732
## - AcidIndex          1    13739 36126
## - LabelAppeal        1    13911 36298
## - STARS              1    14020 36407

##
## Call:  glm(formula = TARGET ~ VolatileAcidity + FreeSulfurDioxide +
##     TotalSulfurDioxide + LabelAppeal + AcidIndex + STARS, family = "poisson",
##     data = trainSet)
##
## Coefficients:
##        (Intercept)     VolatileAcidity   FreeSulfurDioxide  TotalSulfurDioxide
##          1.618e+01          -5.514e-02           1.501e-04           1.083e-04
##        LabelAppeal           AcidIndex               STARS
##          1.847e-01          -2.418e+01           2.134e-01
##
## Degrees of Freedom: 8957 Total (i.e. Null);  8951 Residual
## Null Deviance:       16110
## Residual Deviance: 13290     AIC: 35640
```

StepAIC	Variables
35470	VolatileAcidity FreeSulfurDioxide TotalSulfurDioxide LabelAppeal AcidIndex STARS

Negative Binomial 1

For this negative Binomial Model we will use all the predictor variables.

```
##
## Call:
## glm.nb(formula = TARGET ~ ., data = trainSet, init.theta = 35439.95334,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.7036  -0.5049   0.2254   0.6436   2.1807
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          1.651e+01  7.805e-01  21.150  < 2e-16 ***
## FixedAcidity        -1.567e-03  8.968e-04  -1.747 0.080553 .
## VolatileAcidity     -5.477e-02  7.718e-03  -7.097 1.28e-12 ***
## CitricAcid           1.125e-02  6.967e-03   1.615 0.106292
## ResidualSugar       -1.082e-04  1.798e-04  -0.602 0.547154
## Chlorides           -5.788e-02  1.986e-02  -2.914 0.003573 **
## FreeSulfurDioxide    1.501e-04  3.987e-05   3.766 0.000166 ***
## TotalSulfurDioxide   1.097e-04  2.479e-05   4.426 9.60e-06 ***
## Density             -6.305e-01  2.816e-01  -2.239 0.025151 *
## pH                  -1.830e-02  8.334e-03  -2.195 0.028128 *
## Sulphates           -1.487e-02  6.813e-03  -2.183 0.029045 *
## Alcohol              3.364e-03  1.329e-03   2.530 0.011390 *
## LabelAppeal          1.857e-01  7.437e-03  24.963  < 2e-16 ***
## AcidIndex           -2.375e+01  1.164e+00 -20.399  < 2e-16 ***
## STARS                2.114e-01  7.870e-03  26.858  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(35439.95) family taken to be 1)
##
##     Null deviance: 16107  on 8957  degrees of freedom
## Residual deviance: 13256  on 8943  degrees of freedom
## AIC: 35621
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  35440
##           Std. Err.:  73789
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -35589.35
```

Negative Binomial with BIC recomendation

StepAIC	Variables
35470	VolatileAcidity FreeSulfurDioxide TotalSulfurDioxide LabelAppeal AcidIndex STARS

```
##
## Call:
## glm.nb(formula = TARGET ~ VolatileAcidity + FreeSulfurDioxide +
##     TotalSulfurDioxide + LabelAppeal + AcidIndex + STARS, data = trainSet,
##     init.theta = 34952.71908, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.7297  -0.5095   0.2171   0.6427   2.1961
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          1.618e+01  7.348e-01  22.019  < 2e-16 ***
## VolatileAcidity     -5.514e-02  7.722e-03  -7.141 9.29e-13 ***
## FreeSulfurDioxide    1.501e-04  3.986e-05   3.767 0.000165 ***
## TotalSulfurDioxide   1.083e-04  2.474e-05   4.377 1.20e-05 ***
## LabelAppeal          1.847e-01  7.435e-03  24.848  < 2e-16 ***
## AcidIndex           -2.418e+01  1.141e+00 -21.195  < 2e-16 ***
## STARS                2.134e-01  7.854e-03  27.173  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(34952.72) family taken to be 1)
##
##     Null deviance: 16107  on 8957  degrees of freedom
## Residual deviance: 13293  on 8951  degrees of freedom
## AIC: 35642
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  34953
##           Std. Err.:  73650
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -35626.14
```

Zero Inflated Model

```
## 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 ~ ., data = trainSet, dist = "negbin", link = "logit")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -2.2888 -0.3719  0.1714  0.5103  2.0716
##
## Count model coefficients (negbin with log link):
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         3.374e+00  8.843e-01   3.816 0.000136 ***
## FixedAcidity        4.172e-06  9.363e-04   0.004 0.996444
## VolatileAcidity    -1.004e-02  8.210e-03  -1.223 0.221223
## CitricAcid         -7.428e-04  7.251e-03  -0.102 0.918410
## ResidualSugar      -1.105e-04  1.884e-04  -0.587 0.557483
## Chlorides          -1.232e-02  2.097e-02  -0.588 0.556794
## FreeSulfurDioxide   3.619e-05  4.084e-05   0.886 0.375580
## TotalSulfurDioxide -2.875e-05  2.506e-05  -1.147 0.251194
## Density            -5.612e-01  2.979e-01  -1.884 0.059610 .
## pH                  6.665e-03  8.742e-03   0.762 0.445802
## Sulphates           1.098e-03  7.208e-03   0.152 0.878958
## Alcohol             5.359e-03  1.382e-03   3.877 0.000106 ***
## LabelAppeal         2.420e-01  7.670e-03  31.549  < 2e-16 ***
## AcidIndex          -2.958e+00  1.330e+00  -2.224 0.026175 *
## STARS               1.106e-01  7.715e-03  14.340  < 2e-16 ***
## Log(theta)          1.669e+01        NaN     NaN      NaN
##
## Zero-inflation model coefficients (binomial with logit link):
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -8.538e+01  4.624e+00 -18.465  < 2e-16 ***
## FixedAcidity        6.286e-03  4.603e-03   1.365 0.172122
## VolatileAcidity     2.696e-01  4.068e-02   6.628 3.40e-11 ***
## CitricAcid         -7.748e-02  3.666e-02  -2.113 0.034565 *
## ResidualSugar       1.268e-04  9.361e-04   0.135 0.892275
## Chlorides           2.340e-01  1.038e-01   2.254 0.024222 *
## FreeSulfurDioxide  -7.443e-04  2.144e-04  -3.471 0.000519 ***
## TotalSulfurDioxide -8.625e-04  1.365e-04  -6.318 2.65e-10 ***
## Density             1.038e+00  1.488e+00   0.697 0.485648
## pH                  1.599e-01  4.344e-02   3.680 0.000233 ***
## Sulphates           1.009e-01  3.591e-02   2.811 0.004944 **
## Alcohol             1.069e-02  6.982e-03   1.532 0.125591
## LabelAppeal         3.368e-01  3.920e-02   8.592  < 2e-16 ***
## AcidIndex           1.293e+02  6.938e+00  18.637  < 2e-16 ***
## STARS              -5.752e-01  4.429e-02 -12.986  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 17709409.4791
```

```
## Number of iterations in BFGS optimization: 43
## Log-likelihood: -1.582e+04 on 31 Df
```

4. Select Models

Assessing the fit of a count regression model is not necessarily straightforward; often we just look at residuals, which invariably contain patterns of some form due to the discrete nature of the observations, or we plot observed versus fitted values as a scatter plot. Kleiber and Zeileis (2016) https://arxiv.org/abs/1605.01311 proposes rootogram as an improved approach to the assessment of fit of a count regression model. The paper is illustrated using R and the authors' countreg package.

Rootograms are calculated using the rootogram() function. You can provide the observed and expected (given the model) counts as arguments to rootogram() or, most usefully for our purposes, a fitted count model object from which the relevant values will be extracted. rootogram() knows about glm, gam, gamlss, hurdle, and zeroinfl objects at the time of writing.

Three different kinds of rootograms are discussed in the paper • Standing, • Hanging, and • Suspended.

Kleiber and Zeileis (2016) recommend hanging or suspended rootograms. Which type of rootogram is produced is controlled via argument style. We will look at six different models, two Poisson models, two negative-binomial models and an ols regression thrown in for good measure. Both the Poisson-Logit Hurdle Regression and the zero-inflated Poisson are very close in log likelihoods and BIC's. The Poisson-Logit Hurdle Regression provides a closer fit to the observed than does the other models.

The hurdle model is a modified count model in which there are two processes, one generating the zeros and one generating the positive values.

```
##                     df      AIC
## poisson_trans       15 35619.29
## quasipoisson_model 15       NA
## negbin_m2            8 35642.14
## zeroinf_model       31 31692.81
```

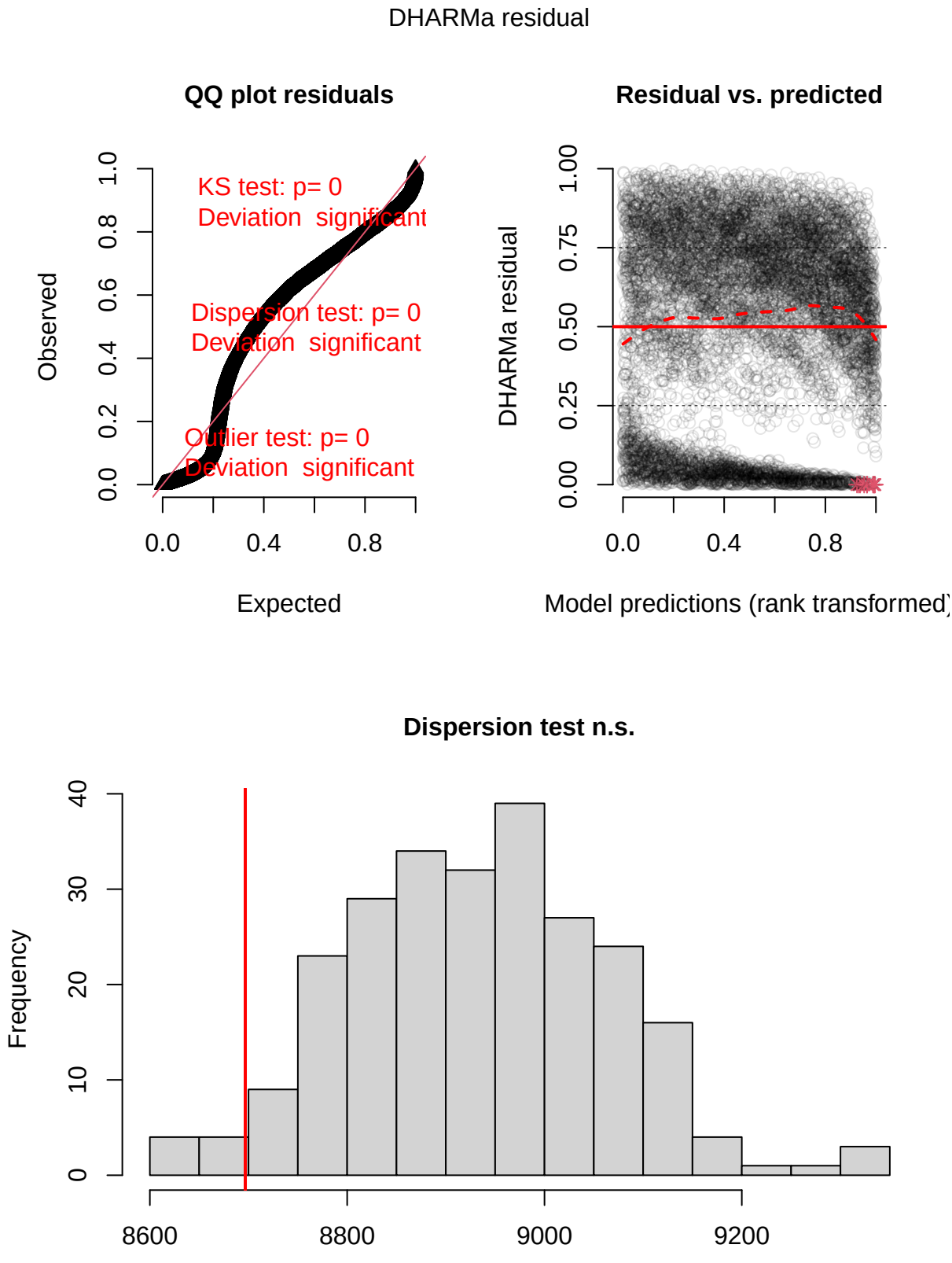
Model	AIC
poisson_trans	35455.44
quasipoisson_trans	-
negbin_m2	35475.08
zeroinf_model	31642.62

Poisson FIT

The Q-Q plot using quantile residuals with a Poisson distribution indicates that the counts of TARGET given certain properties of wine are not well approximated by a Poisson distribution – there are too many observed values near the ends of the expected tails, indicating the expected values are not spread out enough. This pattern emerges because the observed counts are underdespersed compared to a Poisson distribution.

```
## This is DHARMa 0.4.5. For overview type '?DHARMa'. For recent changes, type news(package = 'DHARMa')
```

```
## DHARMa:testOutliers with type = binomial may have inflated Type I error rates for integer-valued distri
```

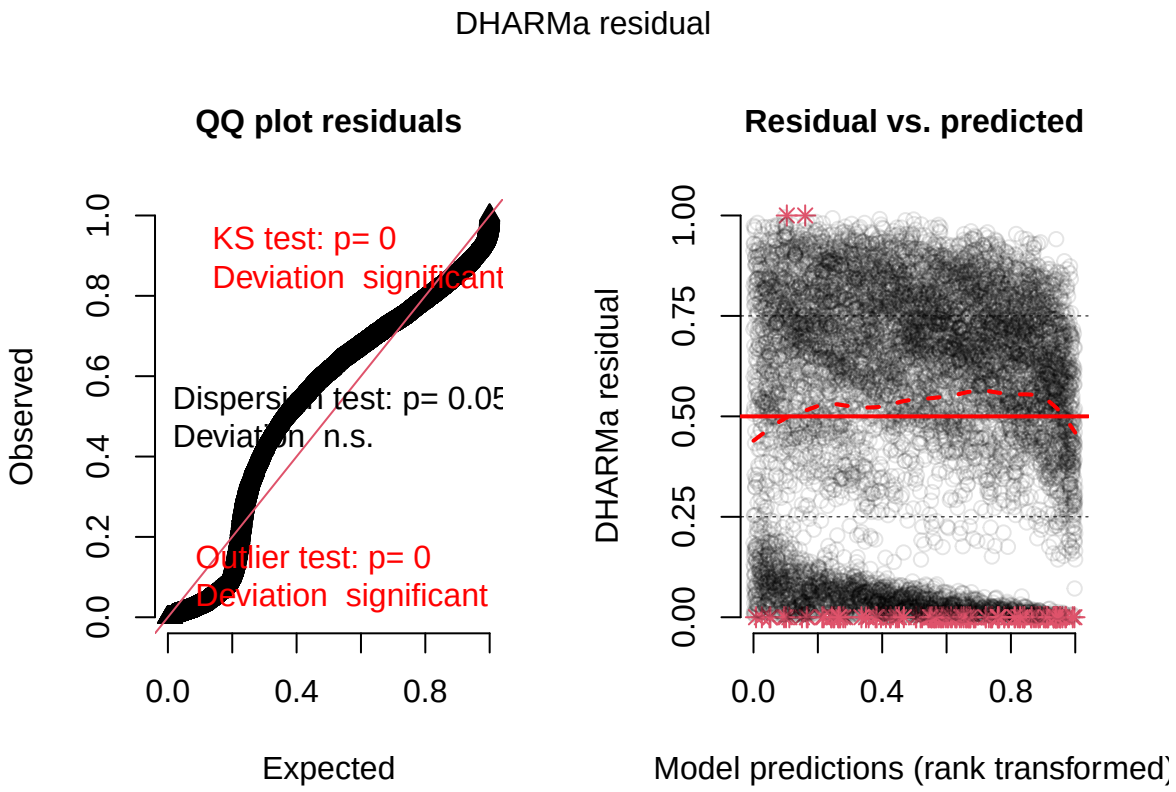

DHARMa residual
QQ plot residuals
KS test: p= 0
Deviation significant
Dispersion test: p= 0
Deviation significant
Outlier test: p= 0
Deviation significant
Observed
Expected
0.0
0.2
0.4
0.6
0.8
1.0
Residual vs. predicted
DHARMa residual
Model predictions (rank transformed)
0.00
0.25
0.50
0.75
1.00
Dispersion test n.s.
Frequency
0
10
20
30
40
8600
8800
9000
9200
Simulated values, red line = fitted model. p−value (two.sided) = 0.056

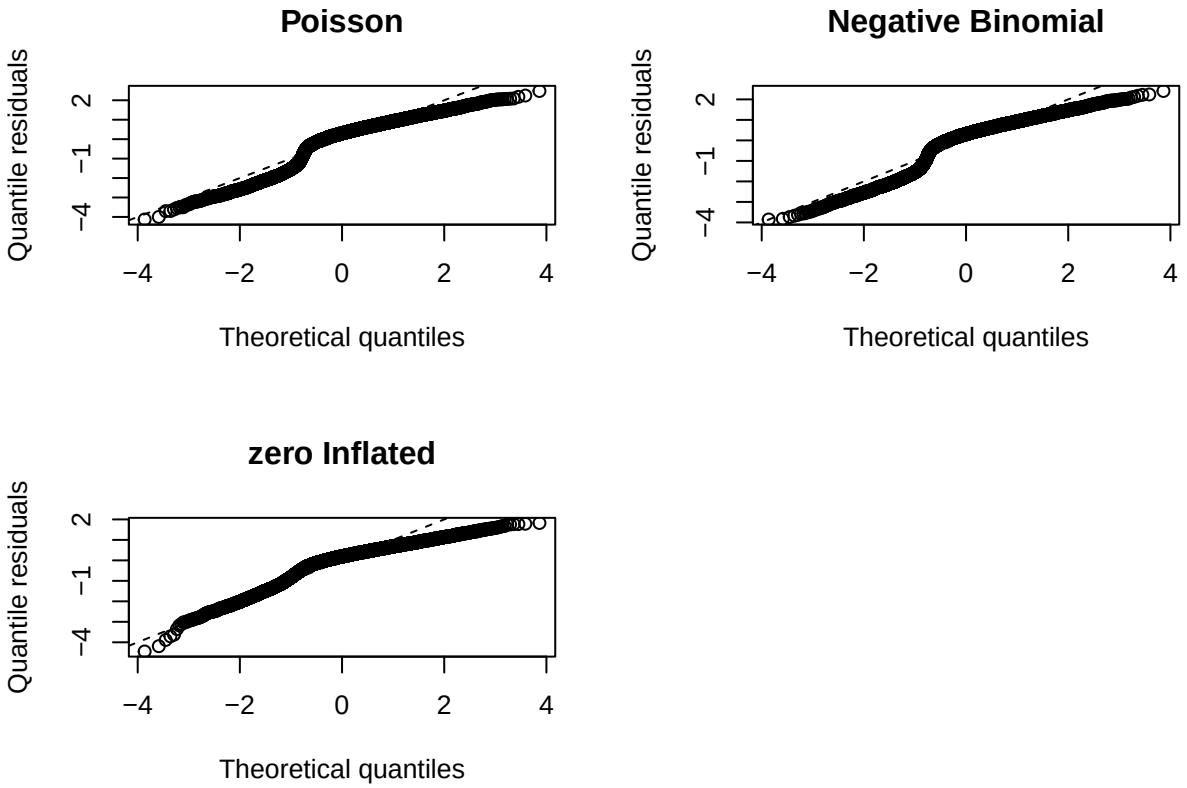
##

```
##  DHARMa nonparametric dispersion test via mean deviance residual fitted
##  vs. simulated-refitted
##
## data:  simulationOutput
## dispersion = 0.97334, p-value = 0.056
## alternative hypothesis: two.sided
```

NEGATIVE BINOMIAL FIT

```
## DHARMa:testOutliers with type = binomial may have inflated Type I error rates for integer-valued distri
```





The Zero inflated model are very close in the log likelihoods and BIC's. It provides a closer fit to the observed than the other models.

Predicted Probabilities

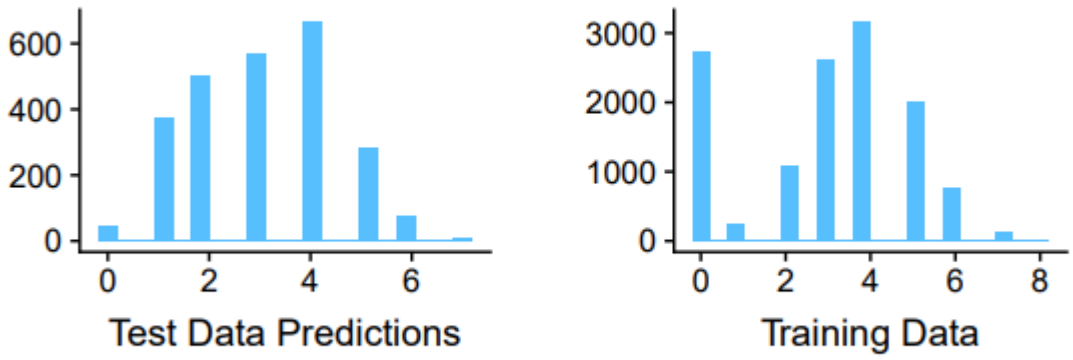


Figure 17: Predictions vs. training data

Apendix

```
knitr::opts_chunk$set(echo = FALSE)

library(tidyverse)
library(skimr)
library(tinytex)
library(e1071)
library(ggthemes)
library(caret)
wine_train <- read_csv("D:/PORTFOLIO_DS/Portfolio_R/Business_Analytics/Business_Analytics/Projects/PROJECT

# wine_eval <- read_csv("https://raw.githubusercontent.com/mgino11/Business_Analytics/main/Projects/PROJEC

skim_without_charts(wine_train)


wine_dist <- wine_train %>%
  select(-INDEX) %>%
  pivot_longer(
    everything(),
    names_to = c("variable"),
    values_to = "value"
  )

ggplot(wine_dist, aes(value)) +
  geom_histogram(aes(x=value, y = ..density..),
                 colour = 4, bins = 30) +
  geom_density(aes(x=value), color = "red") +
  facet_wrap(~variable, scales = "free")

ggsave("wine_dist.pdf")

ggplot(wine_dist, aes(value, variable)) +
```

```
  geom_boxplot(outlier.color = "red") +
  facet_wrap(~variable, scales = "free", drop = FALSE)+
  coord_flip()
wine_scatter <- wine_train %>%
  select(-INDEX) %>%
  pivot_longer(
    cols = -TARGET,
    names_to = c("variable"),
    values_to = "value")
ggplot(wine_scatter, aes(x = value,
                         y = variable,
                         color = as_factor(TARGET))) +
  geom_jitter(position = position_jitterdodge(dodge.width = 0.8,
                                              jitter.width = 0.3),
              shape=21  )
wine_skew <- wine_train %>%
  select(-INDEX)
  #convert target to factor and new names
  # wine_skew$TARGET <- recode_factor(
  #   wine_skew$TARGET, '0' = '','1' = 'high crime' )

ggplot(wine_skew, aes(x=AcidIndex)) +
  geom_histogram(fill = 'white', colour = 'black') +
  facet_grid(TARGET ~ .)
ggplot(wine_skew, aes(x=Alcohol)) +
  geom_histogram(fill = 'white', colour = 'black') +
  facet_grid(TARGET ~ .)
ggplot(wine_skew, aes(x=Chlorides)) +
  geom_histogram(fill = 'white', colour = 'black') +
  facet_grid(TARGET ~ .)
ggplot(wine_skew, aes(x=CitricAcid)) +
  geom_histogram(fill = 'white', colour = 'black') +
  facet_grid(TARGET ~ .)

ggplot(wine_skew, aes(x=Density)) +
  geom_histogram(fill = 'white', colour = 'black') +
  facet_grid(TARGET ~ .)

ggplot(wine_skew, aes(x=FixedAcidity)) +
  geom_histogram(fill = 'white', colour = 'black') +
  facet_grid(TARGET ~ .)

ggplot(wine_skew, aes(x=FreeSulfurDioxide)) +
  geom_histogram(fill = 'white', colour = 'black') +
  facet_grid(TARGET ~ .)

ggplot(wine_skew, aes(x=LabelAppeal)) +
  geom_histogram(fill = 'white', colour = 'black') +
  facet_grid(TARGET ~ .)
ggplot(wine_skew, aes(x=pH)) +
  geom_histogram(fill = 'white', colour = 'black') +
  facet_grid(TARGET ~ .)

ggplot(wine_skew, aes(x=ResidualSugar)) +
  geom_histogram(fill = 'white', colour = 'black') +
  facet_grid(TARGET ~ .)
```

```
ggplot(wine_skew, aes(x=STARS)) +
  geom_histogram(fill = 'white', colour = 'black') +
  facet_grid(TARGET ~ .)

ggplot(wine_skew, aes(x=Sulphates)) +
  geom_histogram(fill = 'white', colour = 'black') +
  facet_grid(TARGET ~ .)

ggplot(wine_skew, aes(x=TotalSulfurDioxide)) +
  geom_histogram(fill = 'white', colour = 'black') +
  facet_grid(TARGET ~ .)

ggplot(wine_skew, aes(x=VolatileAcidity)) +
  geom_histogram(fill = 'white', colour = 'black') +
  facet_grid(TARGET ~ .)


library(corrplot)

corrplot(corr = cor(wine_train,
                    use = 'pairwise.complete.obs'),
         method = "ellipse",
         type = "upper",
         order = "original",
         tl.col = "black",
         tl.srt = 45,
         tl.cex = 0.55)
plot((wine_train$LabelAppeal),(wine_train$STARS))
cor.test(wine_train$LabelAppeal, wine_train$STARS, method = "pearson")
wine_na <- wine_train %>%
  select(-INDEX) %>%
  pivot_longer(
    everything(),
    names_to = c("variable"),
    values_to = "value" ) %>%
  mutate(isna = is.na(value)) %>%
  group_by(variable) %>%
  mutate(total = n()) %>%
  group_by(variable,total,isna) %>%
  summarise(num.isna = n()) %>%
  mutate(pct = num.isna / total * 100)
knitr::kable(wine_na)

library(corrr)
wine_corr <- correlate(wine_train,
                       use = "pairwise.complete.obs",
                       method = "spearman")


wine_corr %>%
  select(-INDEX) %>%
  focus(TARGET) %>%
  fashion()
wine_train %>%
  group_by(TARGET) %>%
```

```
  summarise(count = n() ) %>%
  mutate( prop = count / sum(count) )
df <- wine_train %>%
  select(-INDEX)
set.seed(1188)
trainIndex <- createDataPartition(df$TARGET, p = 0.7, list = F)
trainSet <- df[trainIndex,]
testSet <- df[-trainIndex,]
skim_without_charts(trainSet)
preProcess_na_model <- preProcess(trainSet[,-1], method='bagImpute')
preProcess_na_model

trainSet <- predict(preProcess_na_model, newdata = trainSet)
skim_without_charts(trainSet)
yeoJohnsonModel <- preProcess(trainSet[,2:14], method = "YeoJohnson")
trainSet <- predict(yeoJohnsonModel, newdata = trainSet)
skim_without_charts(trainSet)

featurePlot(x = trainSet[, 2:7],
            y = as.factor(trainSet$TARGET),
            plot = "box",
            ## Pass in options to bwplot()
            scales = list(y = list(relation="free"),
                          x = list(rot = 90)),
            auto.key = list(columns = 2))
featurePlot(x = trainSet[, 8:15],
            y = as.factor(trainSet$TARGET),
            plot = "box",
            ## Pass in options to bwplot()
            scales = list(y = list(relation="free"),
                          x = list(rot = 90)),
            auto.key = list(columns = 2))
poisson_model <- glm(TARGET ~ .,
                     family = "poisson",
                     data = df)
summary(poisson_model)

1-pchisq(deviance(poisson_model), df = poisson_model$df.residual)
Pearson_raw <- sum((df$TARGET - poisson_model$fitted.values)^2
                   / poisson_model$fitted.values)

1 - pchisq(Pearson_raw, df = poisson_model$df.residual)
poisson_trans <- glm(TARGET ~ .,
                     family = "poisson",
                     data = trainSet)
summary(poisson_trans)
1-pchisq(deviance(poisson_trans), df = poisson_trans$df.residual)
Pearson_trans <- sum((trainSet$TARGET - poisson_trans$fitted.values)^2
                   / poisson_trans$fitted.values)

1 - pchisq(Pearson_trans, df = poisson_trans$df.residual)
lambdahat <- fitted(poisson_trans)
par(mfrow=c(1,2), pty="s")

plot(lambdahat, (trainSet$TARGET-lambdahat)^2,
      xlab=expression(hat(lambda)), ylab=expression((y-hat(lambda))^2 ))
```

```
plot(lambdahat, resid(poisson_trans, type = "pearson"),
     xlab=expression(hat(lambda)), ylab="Pearson_Residuals")
# Estimated dispersion parameter
Pearson_trans / poisson_trans$df.residual
quasipoisson_model <- glm(TARGET ~ .,
                      family = "quasipoisson",
                      data = trainSet)
summary(quasipoisson_model)
1-pchisq(deviance(quasipoisson_model), df = quasipoisson_model$df.residual)
library(MASS)

stepAIC(poisson_trans, direction = 'both', k = log(dim(trainSet)[1]))

negbin_m1 <- glm.nb(TARGET ~ .,  data = trainSet)
summary(negbin_m1)

negbin_m2 <- glm.nb(TARGET ~ VolatileAcidity +
                            FreeSulfurDioxide +
                            TotalSulfurDioxide +
                            LabelAppeal +
                            AcidIndex +
                            STARS,  data = trainSet)

summary(negbin_m2)
library(pscl)

zeroinf_model <- zeroinfl(TARGET ~ ., link = "logit",
                          dist = "negbin", data = trainSet)

summary(zeroinf_model)
AIC(poisson_trans, quasipoisson_model, negbin_m2, zeroinf_model)
library(DHARMa)

n_sim <-250
simulationOutput <- simulateResiduals(fittedModel = poisson_trans,
                                      n = n_sim)
plot(simulationOutput, asFactor = F)
n_sim <-250
simulationOutput <- simulateResiduals(fittedModel = poisson_trans,
                                      n = n_sim, refit = T)
testDispersion(simulationOutput)
simulationOutput <- simulateResiduals(fittedModel = negbin_m2,
                                      n = n_sim, refit = T)
plot(simulationOutput, asFactor = F)

library(countreg)
par(mfrow = c(2, 2))
qqrplot(poisson_trans, main = "Poisson")
qqrplot(negbin_m2, main = "Negative Binomial")
qqrplot(zeroinf_model, main = "zero Inflated")
par(mfrow = c(1, 1))
```