Homework 05

CUNY MSDS DATA 621

Duubar Villalobos Jimenez mydvtech@gmail.com

December 09, 2018

Contents

1 HOMEWORK

1.1 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.

1.2 Objective

Your 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. HINT: Sometimes, the fact that a variable is missing is actually predictive of the target. You can only use the variables given to you (or variables that you derive from the variables provided).

1.3 Dataset description

1.3.1 Variable definitions

The below list represent the definitions for each given variable.

VARIABLE_NAME	DEFINITION
INDEX	Identification Variable (do not use)
TARGET	Number of Cases Purchased
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.
ResidualSugar	Residual Sugar of wine
STARS	Wine rating by a team of experts. 4 Stars = Excellent, 1 Star = Poor
Sulphates	Sulfate conten of wine
TotalSulfurDioxide	Total Sulfur Dioxide of Wine
VolatileAcidity	Volatile Acid content of wine
pH	pH of wine

1.3.2 Theoretical effect of variables

The below list represent the theoretical effects for each given variable.

VARIABLE_NAME	THEORETICAL_EFFECT
INDEX	None
TARGET	None
AcidIndex	
Alcohol	
Chlorides	
CitricAcid	
Density	
FixedAcidity	
FreeSulfurDioxide	
LabelAppeal	Many consumers purchase based on the visual appeal of the wine label design. Higher numbers suggest better sales.
ResidualSugar	
STARS	A high number of stars suggests high sales
Sulphates	
TotalSulfurDioxide	
VolatileAcidity	
pH	

1.4 Deliverables

- A write-up submitted in PDF format. Your write-up should have four sections. Each one is described below. You may assume you are addressing me as a fellow data scientist, so do not need to shy away from technical details.
- Assigned predictions (number of cases of wine sold) for the evaluation data set.
- Include your R statistical programming code in an Appendix.

2 DATA EXPLORATION

2.1 Data acquisition

For reproducibility purposes, I have included the original data sets in my Git Hub account, I will read it as a data frame from that location.

```
data.train <- get_data(git_user, git_dir, 'wine-training-data.csv')
data.eval <- get_data(git_user, git_dir, 'wine-evaluation-data.csv')
```

2.2 General exploration

The below process will help us obtain insights from our given data.

2.2.1 Dimensions

Let's see the dimensions of our training data set.

Records	Variables
12795	16

From the above table, we can see how the training data set has a total of 12795 different records and 16 variables including **INDEX** and **TARGET**. These variables do not represent much of the initial insights since they correspond to our response variables and do not offer a theoretical effect.

For simplicity reasons, I will discard the **INDEX** column.

2.2.2 Structure

The below structure is currently present in the data, for simplicity reasons, I have previously loaded and treated this data set as a data frame in which all the variables with decimals are numeric.

variable	class	levels
TARGET	integer	NA
FixedAcidity	numeric	NA
VolatileAcidity	numeric	NA
CitricAcid	numeric	NA
ResidualSugar	numeric	NA
Chlorides	numeric	NA
FreeSulfurDioxide	numeric	NA
TotalSulfurDioxide	numeric	NA
Density	numeric	NA
pH	numeric	NA
Sulphates	numeric	NA
Alcohol	numeric	NA
LabelAppeal	integer	NA
AcidIndex	integer	NA
STARS	integer	NA

From the above table, we can notice how we need to take care of certain values that could be treated as factors. This will be addressed in more detail as we advance in case of this becoming a need.

2.3 Summaries

Let's find some summary statistics about our given data, for that; I will get a little bit more insights for all the columns including the **TARGET** variable.

2.3.1 Combined Summary

In this section, we will explore the combined results as introductory insights.

	Min	1st Qu	Median	Mean	3rd Qu	Max	Other
TARGET	0.0000	2.0000	3.0000	3.029000	4.0000	8.0000	
FixedAcidity	-18.1000	5.2000	6.9000	7.076000	9.5000	34.4000	
VolatileAcidity	-2.7900	0.1300	0.2800	0.324100	0.6400	3.6800	
CitricAcid	-3.2400	0.0300	0.3100	0.308400	0.5800	3.8600	
ResidualSugar	-127.8000	-2.0000	3.9000	5.419000	15.9000	141.1500	NA's :616
Chlorides	-1.1710	-0.0310	0.0460	0.054800	0.1530	1.3510	NA's :638
FreeSulfurDioxide	-555.0000	0.0000	30.0000	30.850000	70.0000	623.0000	NA's :647
TotalSulfurDioxide	-823.0000	27.0000	123.0000	120.700000	208.0000	1057.0000	NA's :682
Density	0.8881	0.9877	0.9945	0.994200	1.0005	1.0992	
pH	0.4800	2.9600	3.2000	3.208000	3.4700	6.1300	NA's :395
Sulphates	-3.1300	0.2800	0.5000	0.527100	0.8600	4.2400	NA's :1210
Alcohol	-4.7000	9.0000	10.4000	10.490000	12.4000	26.5000	NA's :653
LabelAppeal	-2.0000	-1.0000	0.0000	-0.009066	1.0000	2.0000	
AcidIndex	4.0000	7.0000	8.0000	7.773000	8.0000	17.0000	
STARS	1.0000	1.0000	2.0000	2.042000	3.0000	4.0000	NA's :3359

Please note that this is for introductory insights and should not be considered as complete results.

2.4 Findings

From the above table, is interesting to note as follows:

- The training data-set shows the presence of missing values or **NAs** in some columns; that can be seeing in the **Other** column. This will be addressed as we prepare our data down the road.
- Interesting to see a lot of negative values.
- Some variables could suggest non continuity of values such as `LabelAppeal`, `AcidIndex` and `STARS`; these variables could be interpreted as to be categorical variables.

3 DATA PREPARATION

In this section, I will prepare our given data-set. For that I will need to address a few things, like categorical variables and missing data.

3.1 Data conversion

In this section, I will describe the conversion of the data that is required in order to have a more manageable understanding of it.

3.1.1 New Variable: `LabelAppealDISLIKE`

In this section, I will create a new variable named `LabelAppealDISLIKE` in which I will assign the value of 1 if the value `LabelAppeal` < 0; also, I will assign a value of 0 otherwise.

```
data.train <- create_LabelAppealDISLIKE(data.train)
```

3.1.2 New Variable: `LabelAppealBETTER_SALES`

In this section, I will create a new variable named `LabelAppealBETTER_SALES` in which I will assign the value of 1 if the value `LabelAppeal` $= 2$; also, I will assign a value of 0 otherwise.

```
data.train <- create_LabelAppealBETTER_SALES(data.train)
```

3.1.3 New Variable: `STARS_BETTER_SALES`

In this section, I will create a new variable named `STARS_BETTER_SALES` in which I will assign the value of 1 if the value `STARS` $= 4$; also, I will assign a value of 0 otherwise.

```
data.train <- create_STARS_BETTER_SALES(data.train)
```

3.2 NA findings

Let's calculate the proportion of missing values in order to determine the best approach for these variables.

The below list display the combined missing percentage values for each variable.

	NAs %
ResidualSugar	4.81
Chlorides	4.99
FreeSulfurDioxide	5.06
TotalSulfurDioxide	5.33
pH	3.09
Sulphates	9.46
Alcohol	5.10
STARS	26.25

From the above results we can identify a lot of problematic missing data. It is easy to identify that the only one that could be considered as a discrete variable will be STARS.

3.2.1 Complete Cases & NA findings

Let's compare the complete cases vs all given records.

Complete Cases	All Cases	Difference %
6436	12795	49.7

Now, let's compare the means of the complete cases vs the means of all the given original records.

	Mean Complete Cases	Mean All Cases	Difference %
TARGET	3.66900	3.029000	-21.13
FixedAcidity	6.87500	7.076000	2.84
VolatileAcidity	0.30180	0.324100	6.88
CitricAcid	0.31640	0.308400	-2.59
ResidualSugar	5.54000	5.419000	-2.23
Chlorides	0.04797	0.054800	12.46
FreeSulfurDioxide	32.45000	30.850000	-5.19
TotalSulfurDioxide	124.90000	120.700000	-3.48
Density	0.99370	0.994200	0.05
pH	3.19500	3.208000	0.41
Sulphates	0.50550	0.527100	4.10
Alcohol	10.56000	10.490000	-0.67
LabelAppeal	0.04475	-0.009066	593.60
AcidIndex	7.65000	7.773000	1.58
STARS	2.03900	2.042000	0.15
LabelAppealDISLIKE	0.25480	0.284500	10.44
LabelAppealBETTER_SALES	0.04055	0.038300	-5.87
STARS_BETTER_SALES	0.06122	0.047830	-27.99

From the above table, we notice how by sub-setting the data, we might run the risk of over-performing our lineal model by over estimating `TARGET` by about 21.13 % higher.

Also, we can easily observe how if the `LabelAppeal` variable is considered in the model as an estimator, this could mislead in about 593.6 % for that single estimator value in the remaining 50% of the data set.

From the above, it is evident that we need to be careful on how to approach these missing values due to high variability of results on both cases.

3.2.2 New Variable: `STARS_NA`

In this section, I will create a new variable named `STARS_NA` in which I will assign the value of 1 if the value `STARS` $= NA$; also, I will assign a value of 0 otherwise; on top of that I will assign a value of ZERO to all NAs present in the `STARS` column. The idea is to increase the number of records that could be used. The idea is to see how we could correct those problematic differences evidenced above.

```
data.train <- create_STARS_NA(data.train)
```

3.2.3 New NA findings

Let's calculate the proportion of missing values after the above work was performed in order to determine the best approach from now on.

The below list display the combined missing percentage values for each variable.

Complete Cases	All Cases	Difference %
8675	12795	32.2

We can note an increase of records and the difference now is much less than before.

Now, let's compare the means of the complete cases vs all the given original records.

	Mean Complete Cases	Mean All Cases	Difference %
TARGET	3.033000	3.029000	-0.13
FixedAcidity	7.030000	7.076000	0.65
VolatileAcidity	0.327100	0.324100	-0.93
CitricAcid	0.309500	0.308400	-0.36
ResidualSugar	5.473000	5.419000	-1.00
Chlorides	0.053810	0.054800	1.81
FreeSulfurDioxide	30.420000	30.850000	1.39
TotalSulfurDioxide	120.300000	120.700000	0.33
Density	0.994200	0.994200	0.00
pH	3.202000	3.208000	0.19
Sulphates	0.517500	0.527100	1.82
Alcohol	10.470000	10.490000	0.19
LabelAppeal	-0.003112	-0.009066	65.67
AcidIndex	7.787000	7.773000	-0.18
STARS	1.513000	1.506000	-0.46
LabelAppealDISLIKE	0.278300	0.284500	2.18
LabelAppealBETTER_SALES	0.038270	0.038300	0.08
STARS_BETTER_SALES	0.045420	0.047830	5.04
STARS_NA	0.258100	0.262500	1.68

Now, we can easily identify a more approachable data set since the percentage value of the difference of the means are not that far apart. Even the value corresponding to `LabelAppeal` is much more manageable.

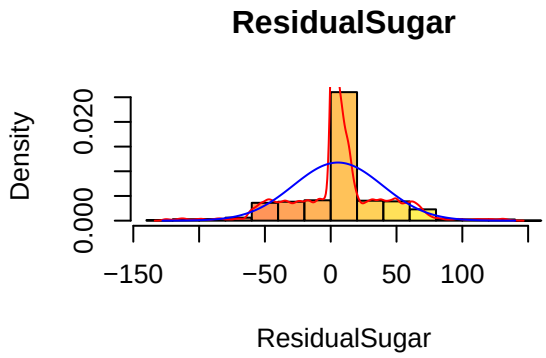
3.3 Fill in remaining NAs

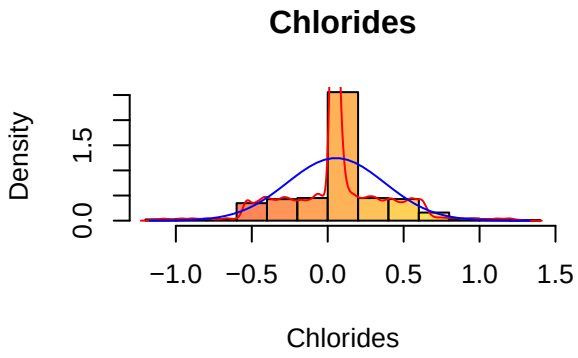
From the previous table, it was observed some percentages of missing values. Let's recap them.

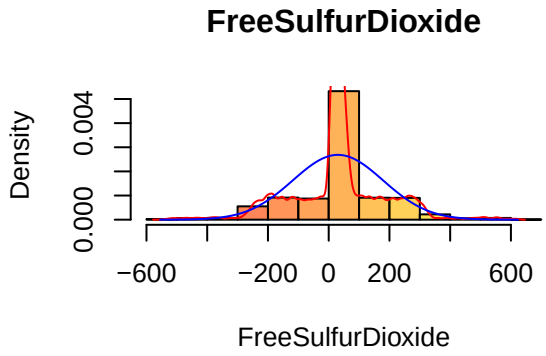
	NAs %
ResidualSugar	4.81
Chlorides	4.99
FreeSulfurDioxide	5.06
TotalSulfurDioxide	5.33
pH	3.09
Sulphates	9.46
Alcohol	5.10

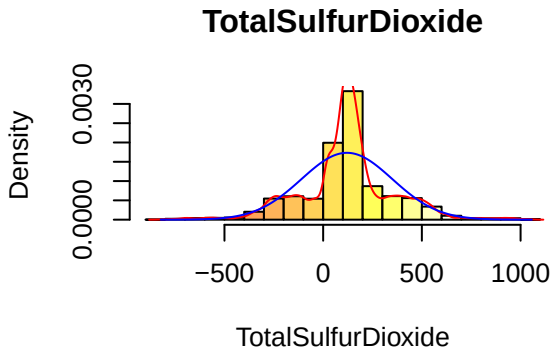
Please note that the `STARS` variable missing values were previously addressed by assigning ZERO values and by creating a new variable `STARS_NA` indicating that it was missing.

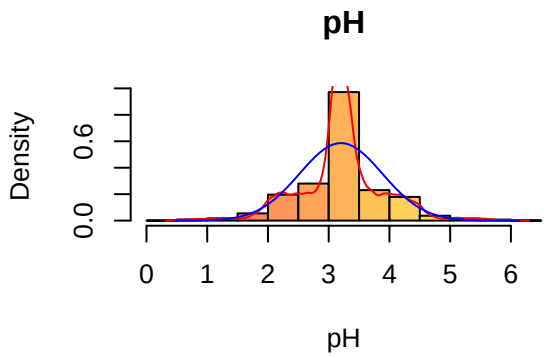
In this section, I will replace all the remaining missing values with randomly generated values from the minimum value to the maximum value for each variable. That is considered since the distributions seems to follow a normal curve as seeing in the below graphs.

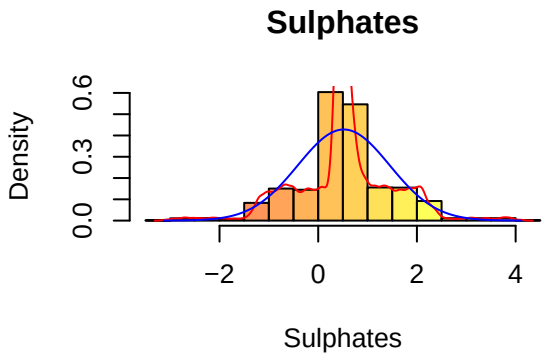


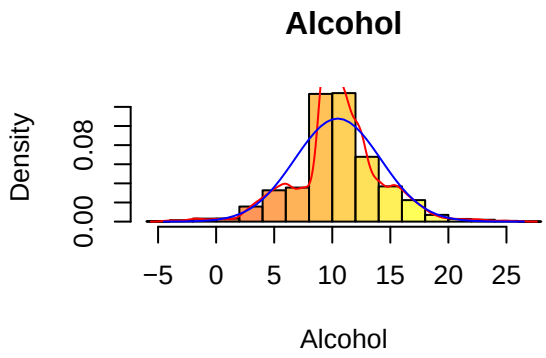


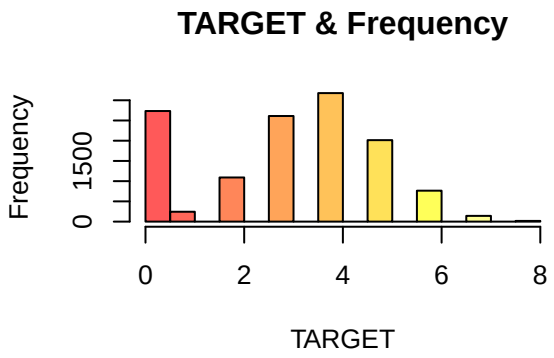













```
data.train <- fill_missing_na(data.train)
```

3.4 Tranformations

In order to make our data more "linear", I will transform the values of some variables in the following form.

$log\_var = log(|min(var)| + var + 1)$

That is, the absolute value of the minimum value for that variable plus the same variable plus one, thus in order to satisfy logarithmic properties.

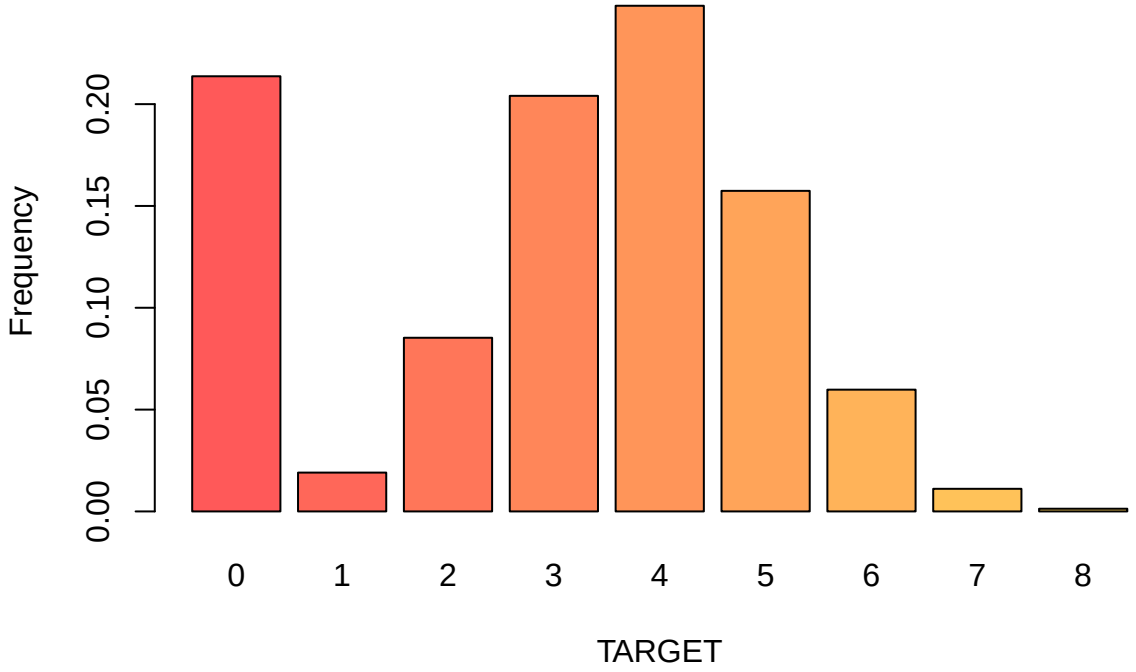
```
data.train <- transform_vars(data.train)
data.train <- data.train[c(1,13:29)]
```

3.5 Visualizations

Let's create a few visualizations in order to get a better understanding.

3.5.1 `TARGET` density

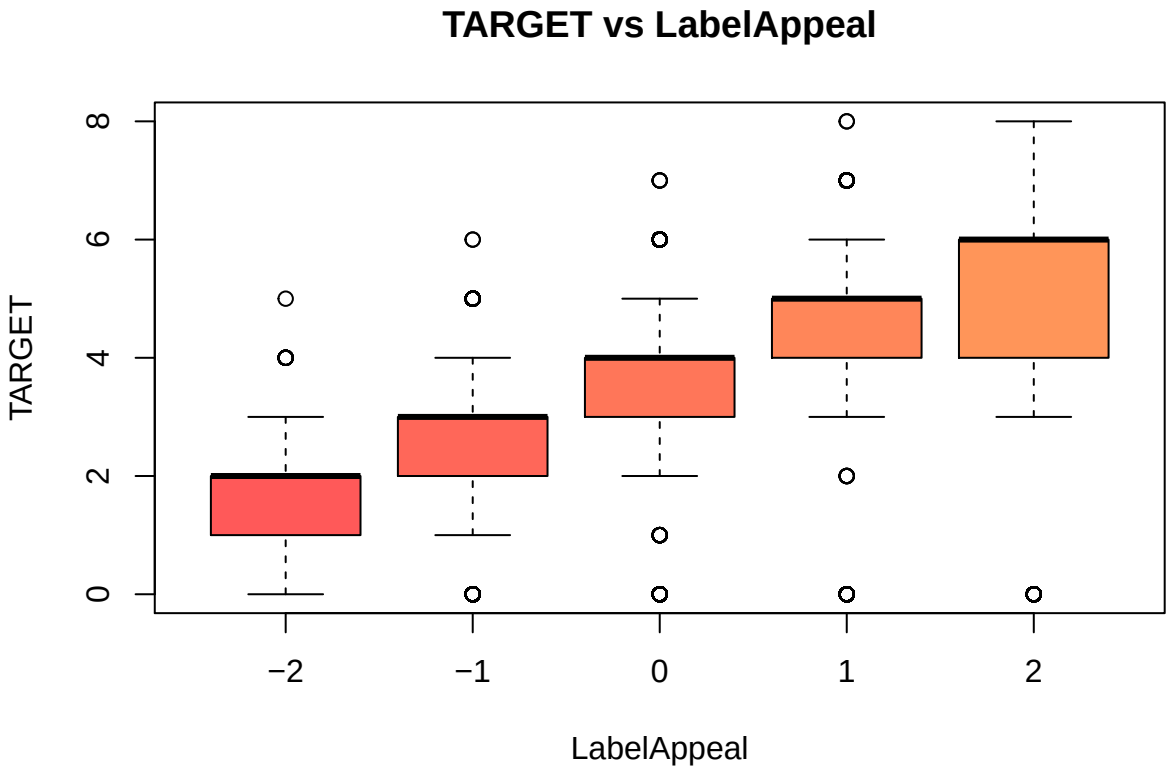
Let's visualize the frequency of `TARGET`.



The above visualization, shows that `TARGET` seems to follow a normal frequency distribution pattern for all `TARGET` values higher than ZERO. Interesting to see a lot of ZERO cases in the `TARGET` variable, I was not expecting to see those many records displaying that frequency.

3.5.2 `TARGET` vs `LabelAppeal`

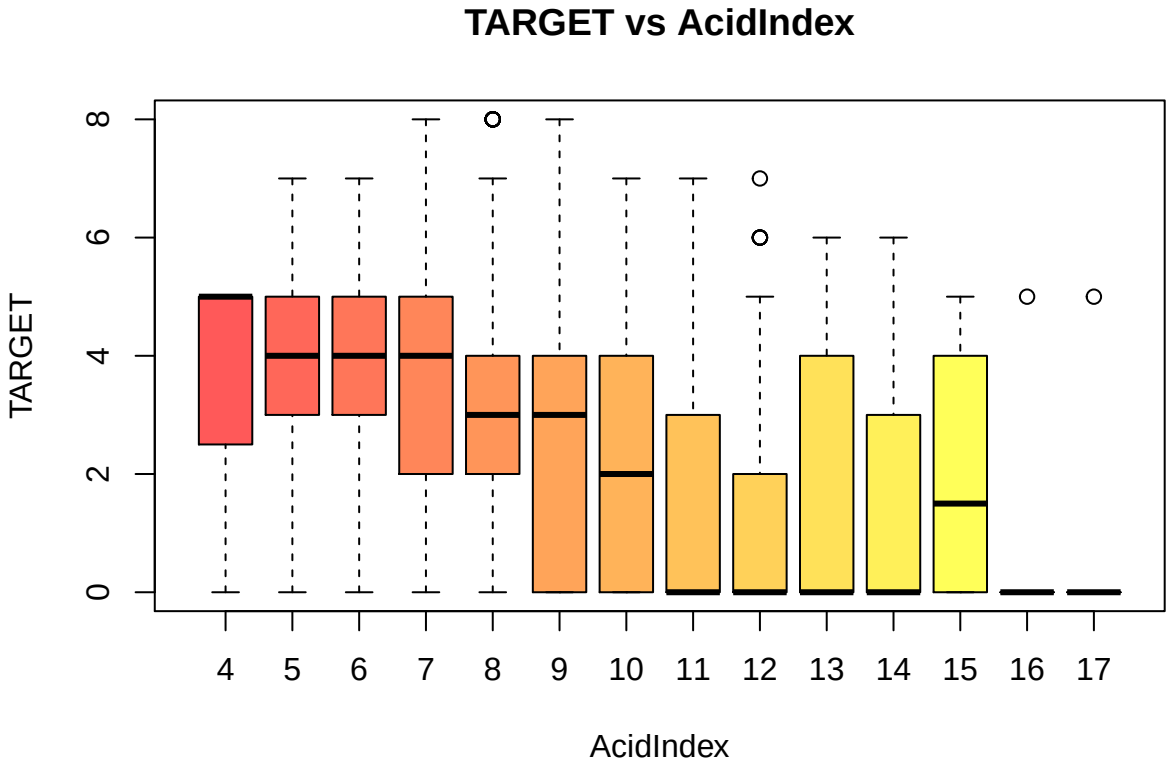
Let's visualize the label appeal.



From the above graph, we can easily identify a linear trend in which the higher the appeal the higher the `TARGET`.

3.5.3 `TARGET` vs `AcidIndex`

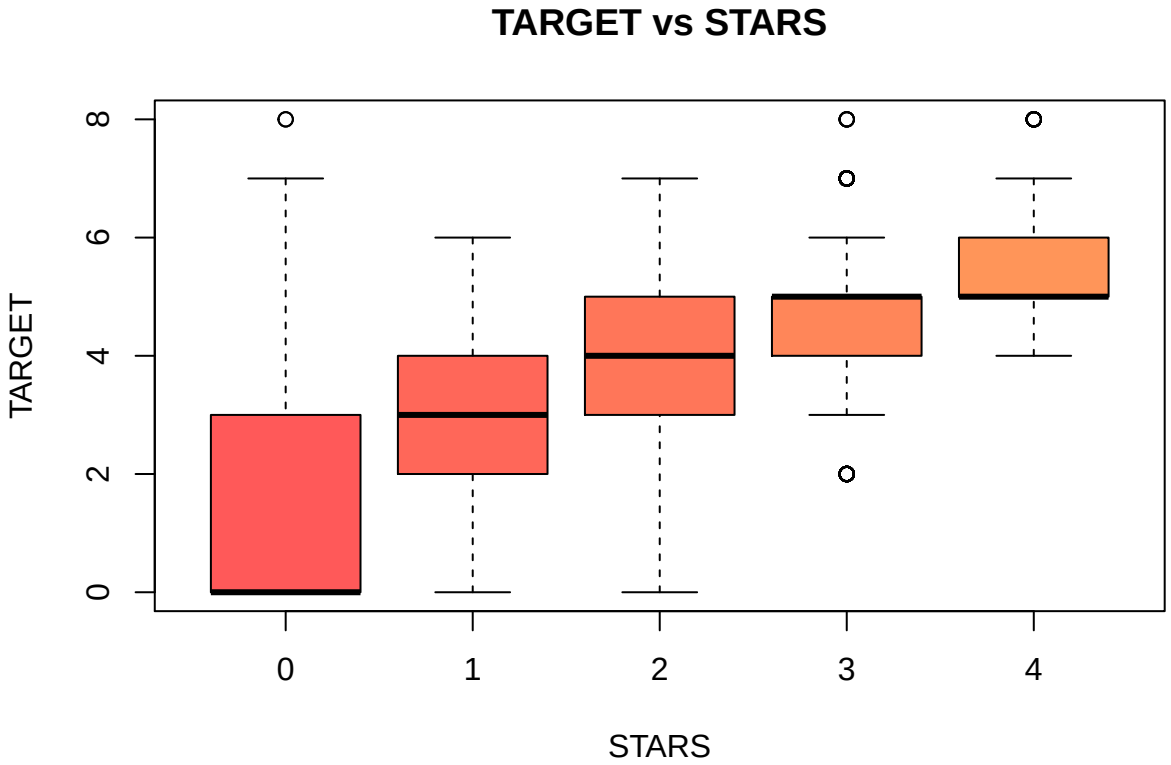
Let's see if we could identify some sort of pattern in the below graph.



The above results seem to be interesting and they suggest some sort of downward relationship; that is, the higher the `AcidIndex` the lower the `TARGET`. However, there seems to be the presence of outliers as well.

3.5.4 `TARGET` vs `STARS`

In this case, we must keep in mind that NAs were assigned a ZERO value.



From the above graph, we can easily start some analysis to confirm the theoretical effect in which a high number of `STARS` could suggest a high number of sales.

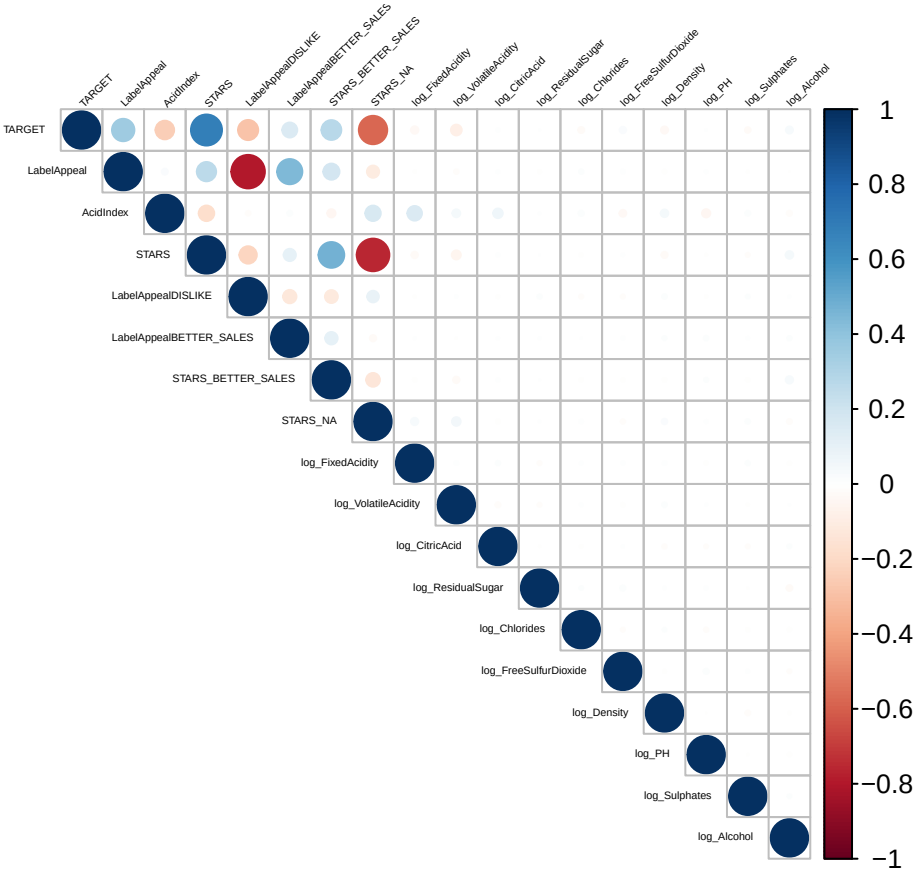
3.6 Correlations

Let's create some visualizations for the correlation matrix.

Let's start with a combined correlation in between `TARGET` and the other variables.

3.6.1 Graphical correlations

First, let's create a visual representation of correlations with a heat-map as a guide.



In the correlation matrix, We notice some very strong correlations in the above plot. We also can note some strong negative correlations in between `LabelAppeal` and `LabelAppealDISLIKE`; this makes sense since I have purposely created this relationship in negative terms. Also, the same applies for `STARS` and `STARS_NA`.

3.6.2 Numerical correlation

From the above graph, we can easily identify some sort of correlations in between the response variables `TARGET` and other variables.

Let's read our correlations table to gain extra insights.

	TARGET
TARGET	1.0000000
LabelAppeal	0.3565005
AcidIndex	-0.2460494
STARS	0.6853815
LabelAppealDISLIKE	-0.2873964
LabelAppealBETTER_SALES	0.1595674
STARS_BETTER_SALES	0.2783731
STARS_NA	-0.5715792
log_FixedAcidity	-0.0388490
log_VolatileAcidity	-0.0821966
log_CitricAcid	0.0073295
log_ResidualSugar	0.0009639
log_Chlorides	-0.0251238
log_FreeSulfurDioxide	0.0286649
log_Density	-0.0354320
log_PH	0.0013567
log_Sulphates	-0.0218038
log_Alcohol	0.0349371

Interesting to note how `STARS` now show to have the strongest correlation. Let's see if we could take some variables in which the absolute value could be higher or equal than let's say 0.25.

	TARGET
TARGET	1.0000000
LabelAppeal	0.3565005
STARS	0.6853815
LabelAppealDISLIKE	-0.2873964
STARS_BETTER_SALES	0.2783731
STARS_NA	-0.5715792

From the above table, we can now narrow it down to the "strongest" correlations related to `TARGET`; we can easily identify some positive and some negative correlations which I consider interesting. I will keep these variables in mind for later on in case I need to refine some models.

4 BUILD MODELS

At this point, we are getting ready to start building models, however I would like to point out that in this case is a little bit difficult to determine what data transformation could be used in order to refine our models.

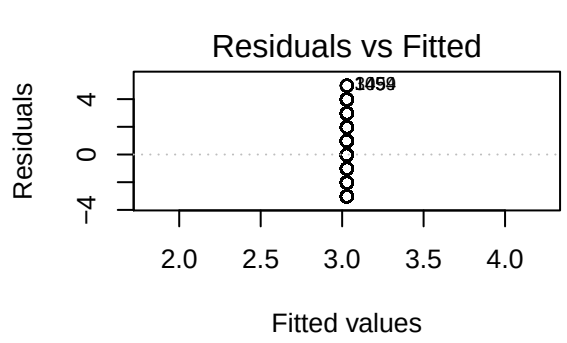
4.1 Multiple Linear Regression Models

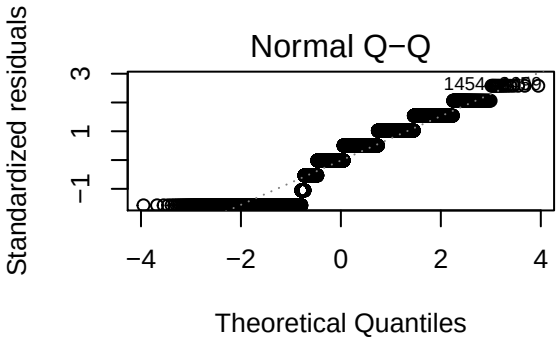
In this section I will present multiple models, build with the linear model.

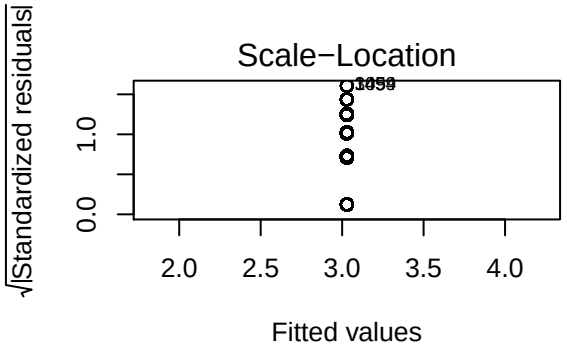
4.1.1 NULL Model

Let's start with a null model in order to start having a better understanding. This model will be considered to be valid and will be considered as we advance.

```
##
## Call:
## lm(formula = TARGET ~ 1, data = data.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0291 -1.0291 -0.0291  0.9709  4.9709
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.02907    0.01703   177.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.926 on 12794 degrees of freedom
```







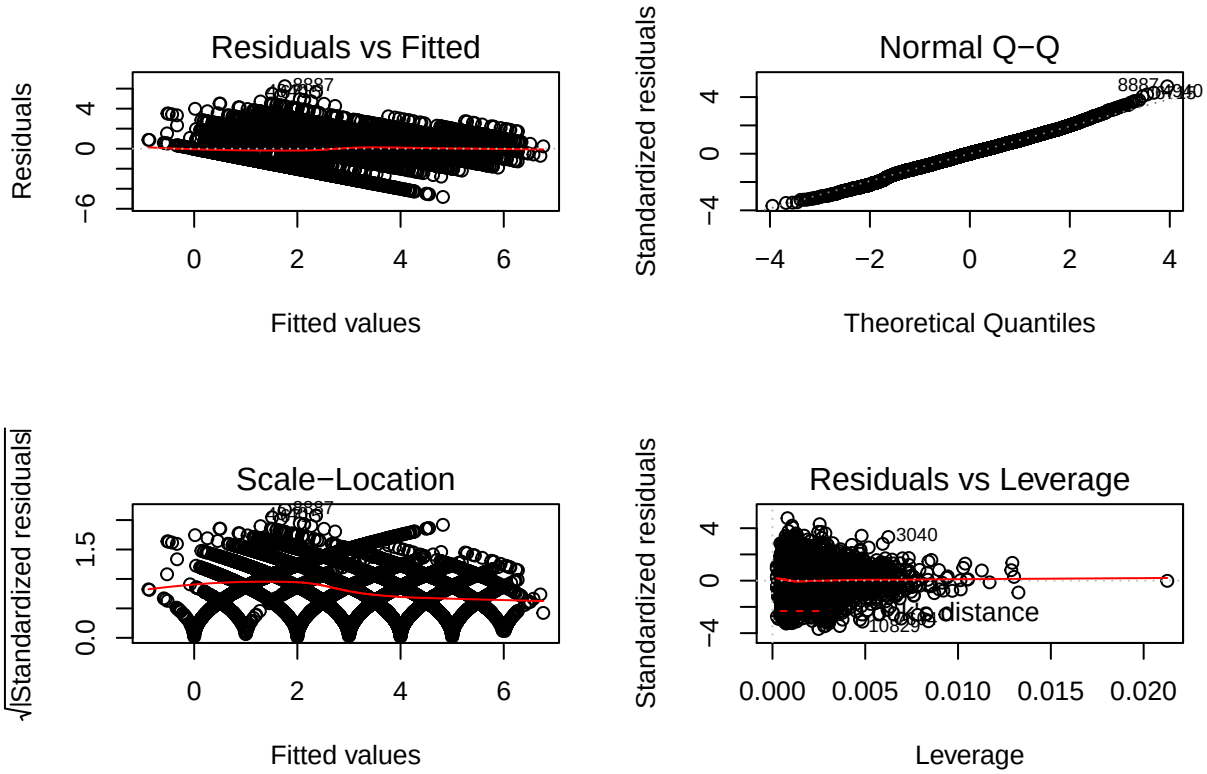
In this model, we can see a constant value of 3 cases for all diverse options; in somehow it agrees with the data since the majority of cases sold were from 3 to 4 cases but since it's constant, we should take it as a baseline.

4.1.2 FULL Model

In this section, I will build a FULL model, thus in order to keep having a better understanding of the model. This model will be considered to be valid and will be considered as we advance.

```
##
## Call:
## lm(formula = TARGET ~ ., data = data.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8214 -0.8535  0.0230  0.8539  6.2433
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               5.863025   1.375656   4.262 2.04e-05 ***
## LabelAppeal               0.445227   0.027118  16.418  < 2e-16 ***
## AcidIndex                -0.203697   0.009074 -22.448  < 2e-16 ***
## STARS                     0.821676   0.018805  43.693  < 2e-16 ***
## LabelAppealDISLIKE       -0.016959   0.047564  -0.357 0.721440
## LabelAppealBETTER_SALES   0.141252   0.075203   1.878 0.060368 .
## STARS_BETTER_SALES       -0.247876   0.066942  -3.703 0.000214 ***
## STARS_NA                 -0.626019   0.044216 -14.158  < 2e-16 ***
## log_FixedAcidity          0.021335   0.039332   0.542 0.587529
## log_VolatileAcidity      -0.336176   0.055504  -6.057 1.43e-09 ***
```

```
## log_CitricAcid           0.079108   0.055414   1.428 0.153441
## log_ResidualSugar        0.012540   0.030022   0.418 0.676187
## log_Chlorides           -0.215609   0.063589  -3.391 0.000699 ***
## log_FreeSulfurDioxide    0.081968   0.029558   2.773 0.005561 **
## log_Density             -2.276959   1.260986  -1.806 0.070989 .
## log_PH                  -0.049066   0.068781  -0.713 0.475636
## log_Sulphates           -0.075970   0.038984  -1.949 0.051348 .
## log_Alcohol              0.058993   0.034831   1.694 0.090351 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.311 on 12777 degrees of freedom
## Multiple R-squared:  0.5371, Adjusted R-squared:  0.5365
## F-statistic: 872.2 on 17 and 12777 DF,  p-value: < 2.2e-16
```



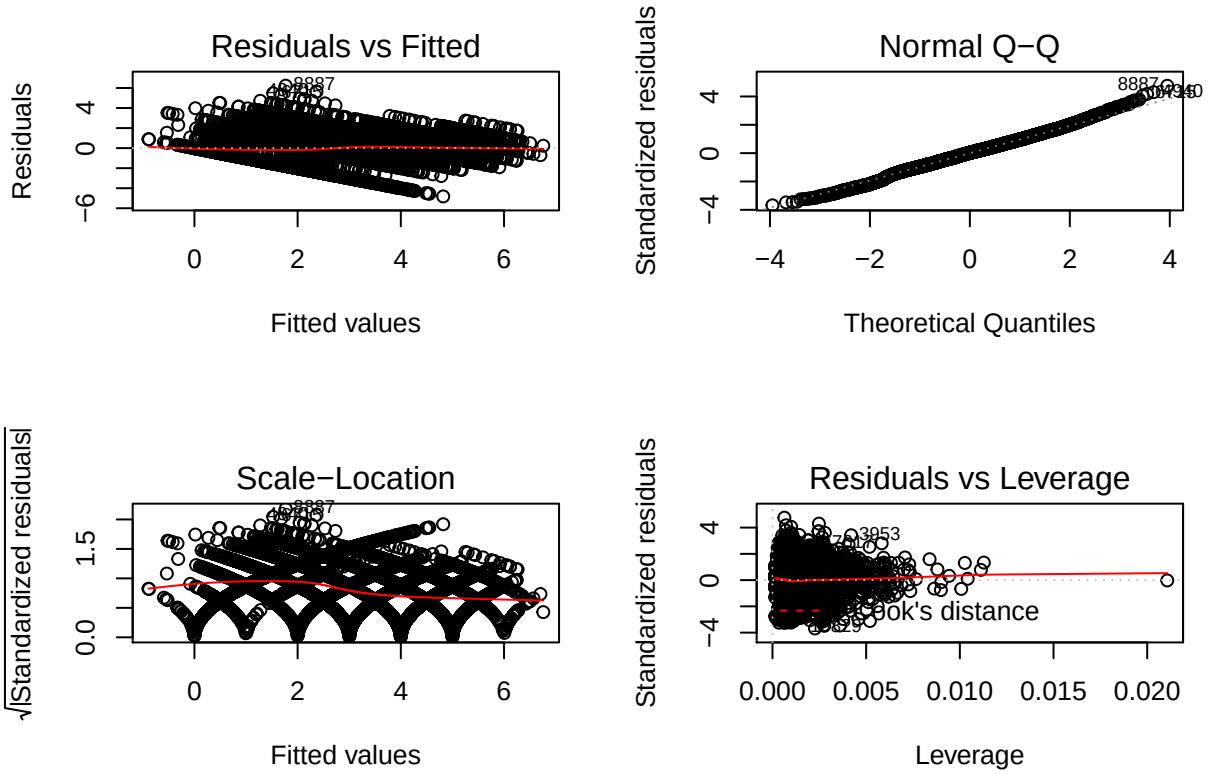
In the above results, we can notice how some predictors are statistically significant while others are not. Also, we noticed how the R^2 is presenting a "moderate" rate in which just about 53% of the variability is fully explained by this model. Also, we can notice how the residual values vs the fitted values do not seem to be homoescedastic.

4.1.3 STEP Model

In this section, I will build a model by employing the STEP function from R, thus in order to keep having a better understanding of the model. This model will be considered to be valid and will be considered as we advance.

```
##
## Call:
## lm(formula = TARGET ~ STARS + LabelAppeal + AcidIndex + STARS_NA +
```

```
##     log_VolatileAcidity + STARS_BETTER_SALES + log_Chlorides +
##     log_FreeSulfurDioxide + log_Sulphates + LabelAppealBETTER_SALES +
##     log_Density + log_Alcohol + log_CitricAcid, data = data.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8217 -0.8535  0.0250  0.8514  6.2319
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               5.901102   1.360131   4.339 1.44e-05 ***
## STARS                     0.822046   0.018800  43.727  < 2e-16 ***
## LabelAppeal               0.453049   0.015181  29.843  < 2e-16 ***
## AcidIndex                -0.202660   0.008964 -22.609  < 2e-16 ***
## STARS_NA                 -0.625614   0.044204 -14.153  < 2e-16 ***
## log_VolatileAcidity      -0.336247   0.055487  -6.060 1.40e-09 ***
## STARS_BETTER_SALES       -0.249709   0.066872  -3.734 0.000189 ***
## log_Chlorides            -0.214656   0.063572  -3.377 0.000736 ***
## log_FreeSulfurDioxide     0.082164   0.029541   2.781 0.005421 **
## log_Sulphates            -0.075892   0.038974  -1.947 0.051527 .
## LabelAppealBETTER_SALES   0.129326   0.067729   1.909 0.056224 .
## log_Density              -2.275653   1.260801  -1.805 0.071110 .
## log_Alcohol               0.058803   0.034815   1.689 0.091242 .
## log_CitricAcid            0.079571   0.055406   1.436 0.150983
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.311 on 12781 degrees of freedom
## Multiple R-squared:  0.5371, Adjusted R-squared:  0.5366
## F-statistic:  1141 on 13 and 12781 DF,  p-value: < 2.2e-16
```



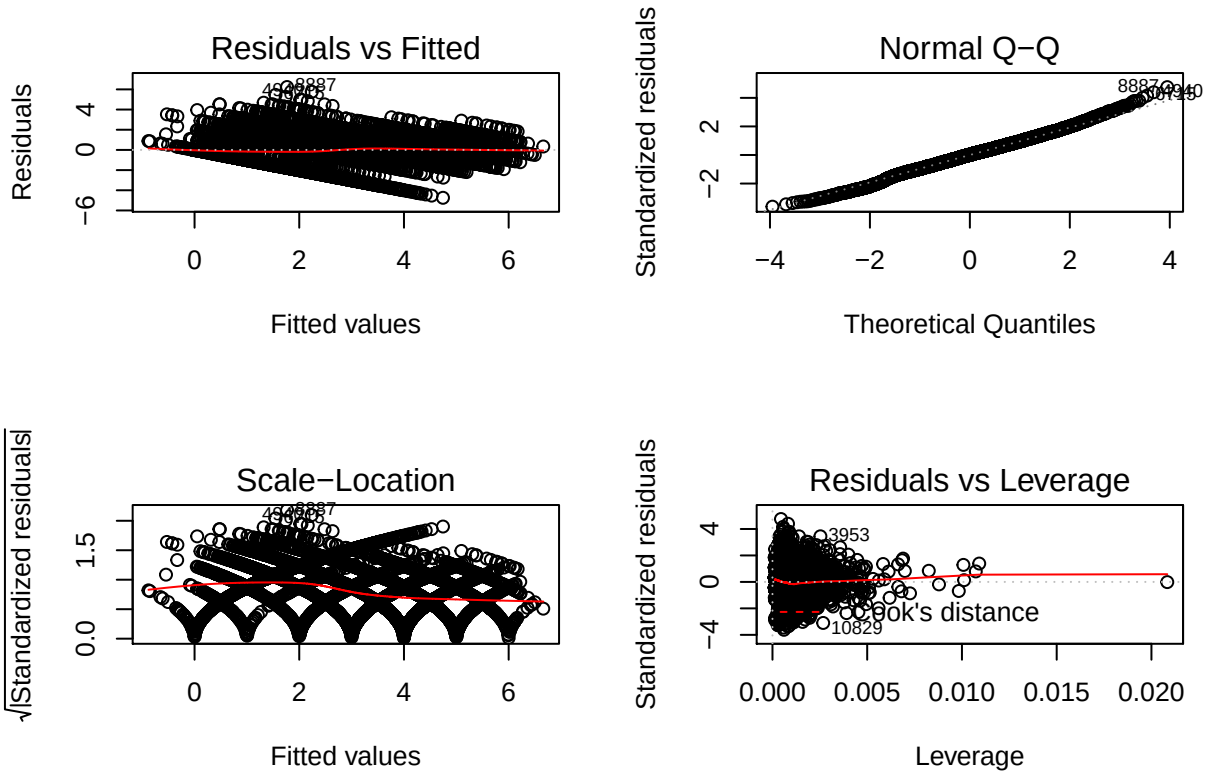
Interesting to note that not much was gained from the above model generated from the STEP Model. From the above plot, we can notice how the fitted values vs the fitted values do not seems to be homoscedastic; the normal quantile to quantile line seems to be followed with an exception to the top values.

4.1.4 STEP Model Modified

In this section, I will build a model by making a modification of the STEP model given above. I will remove `log_CitricAcid log_Alcohol log_Density LabelAppealBETTER_SALES log_Sulphates`.

```
##
## Call:
## lm(formula = TARGET ~ STARS + LabelAppeal + AcidIndex + STARS_NA +
##     log_VolatileAcidity + STARS_BETTER_SALES + log_Chlorides +
##     log_FreeSulfurDioxide, data = data.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7456 -0.8493  0.0269  0.8481  6.2331
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              3.669213   0.225334  16.283  < 2e-16 ***
## STARS                    0.822427   0.018796  43.756  < 2e-16 ***
## LabelAppeal              0.465493   0.013686  34.012  < 2e-16 ***
## AcidIndex               -0.202790   0.008939 -22.687  < 2e-16 ***
## STARS_NA                -0.626053   0.044211 -14.161  < 2e-16 ***
## log_VolatileAcidity     -0.337539   0.055492  -6.083 1.22e-09 ***
## STARS_BETTER_SALES      -0.241776   0.066840  -3.617 0.000299 ***
## log_Chlorides           -0.216930   0.063588  -3.411 0.000648 ***
```

```
## log_FreeSulfurDioxide  0.081880   0.029549   2.771 0.005598 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.312 on 12786 degrees of freedom
## Multiple R-squared:  0.5365, Adjusted R-squared:  0.5363
## F-statistic:  1850 on 8 and 12786 DF,  p-value: < 2.2e-16
```



Very similar plots and results to the previous one, not much gain.

4.2 Poisson Regression Models

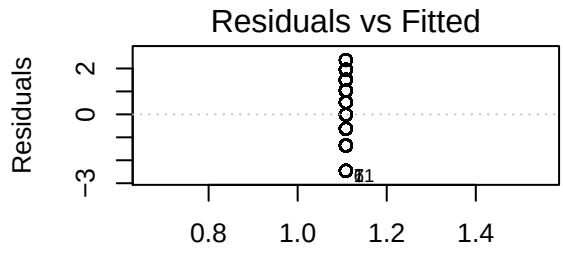
In this section I will present multiple models, build using the generalized linear model. Poisson regression is often used for modeling count data. Poisson regression has a number of extensions useful for count models.

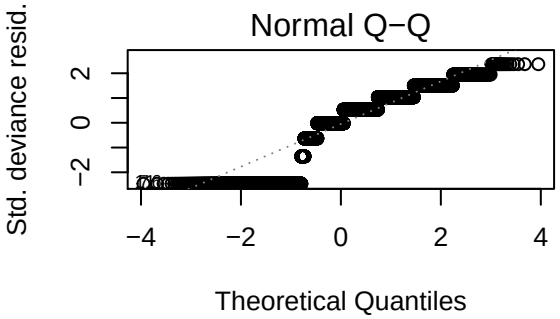
4.2.1 NULL Model

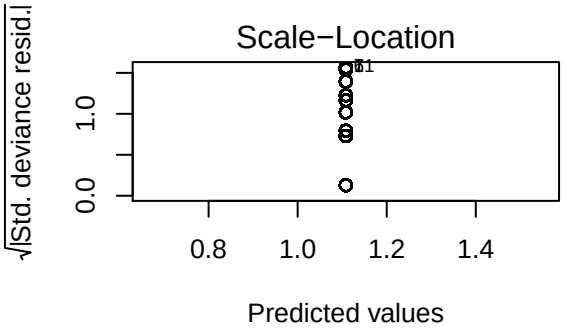
Let's start with a null model in order to have better understanding. This model will be considered to be valid and will be considered as we advance.

```
##
## Call:
## glm(formula = TARGET ~ 1, family = "poisson", data = data.train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.46133  -0.63064  -0.01673   0.53146   2.36582
##
## Coefficients:
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.10826    0.00508   218.2   <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: 22861  on 12794  degrees of freedom
## AIC: 54805
##
## Number of Fisher Scoring iterations: 5
```







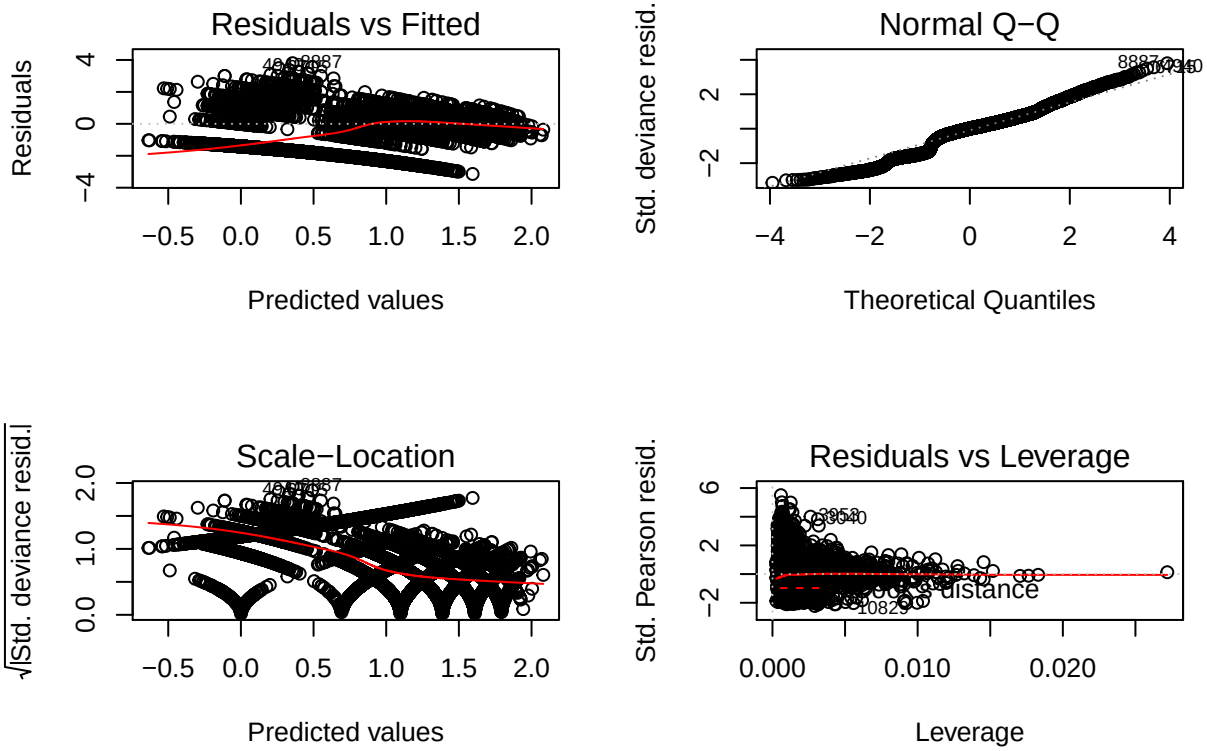
In this particular case, we notice how the number of cases has decreased to about 1 for all the complete cases.

4.2.2 FULL Model

Let's build a full model in order to keep having a better understanding. This model will be considered to be valid and will be considered as we advance.

```
##
## Call:
## glm(formula = TARGET ~ ., family = "poisson", data = data.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1388  -0.6441   0.0038   0.4551   3.7992
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)              2.229897   0.602591   3.701 0.000215 ***
## LabelAppeal              0.142194   0.011632  12.225  < 2e-16 ***
## AcidIndex               -0.081014   0.004546 -17.820  < 2e-16 ***
## STARS                    0.216347   0.007628  28.361  < 2e-16 ***
## LabelAppealDISLIKE      -0.054541   0.021525  -2.534 0.011282 *
## LabelAppealBETTER_SALES -0.014582   0.028519  -0.511 0.609121
## STARS_BETTER_SALES      -0.137151   0.023184  -5.916 3.30e-09 ***
## STARS_NA                -0.599643   0.022875 -26.214  < 2e-16 ***
## log_FixedAcidity         0.004590   0.016995   0.270 0.787115
## log_VolatileAcidity     -0.105543   0.023793  -4.436 9.17e-06 ***
## log_CitricAcid           0.024298   0.024261   1.002 0.316565
## log_ResidualSugar        0.003132   0.013058   0.240 0.810411
## log_Chlorides           -0.069784   0.027817  -2.509 0.012120 *
## log_FreeSulfurDioxide    0.026622   0.013166   2.022 0.043180 *
## log_Density             -0.727234   0.552371  -1.317 0.187983
## log_PH                  -0.024959   0.030075  -0.830 0.406597
## log_Sulphates           -0.026456   0.016989  -1.557 0.119422
## log_Alcohol              0.015167   0.015444   0.982 0.326072
## ---
## 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: 13748  on 12777  degrees of freedom
## AIC: 45726
##
## Number of Fisher Scoring iterations: 6
```



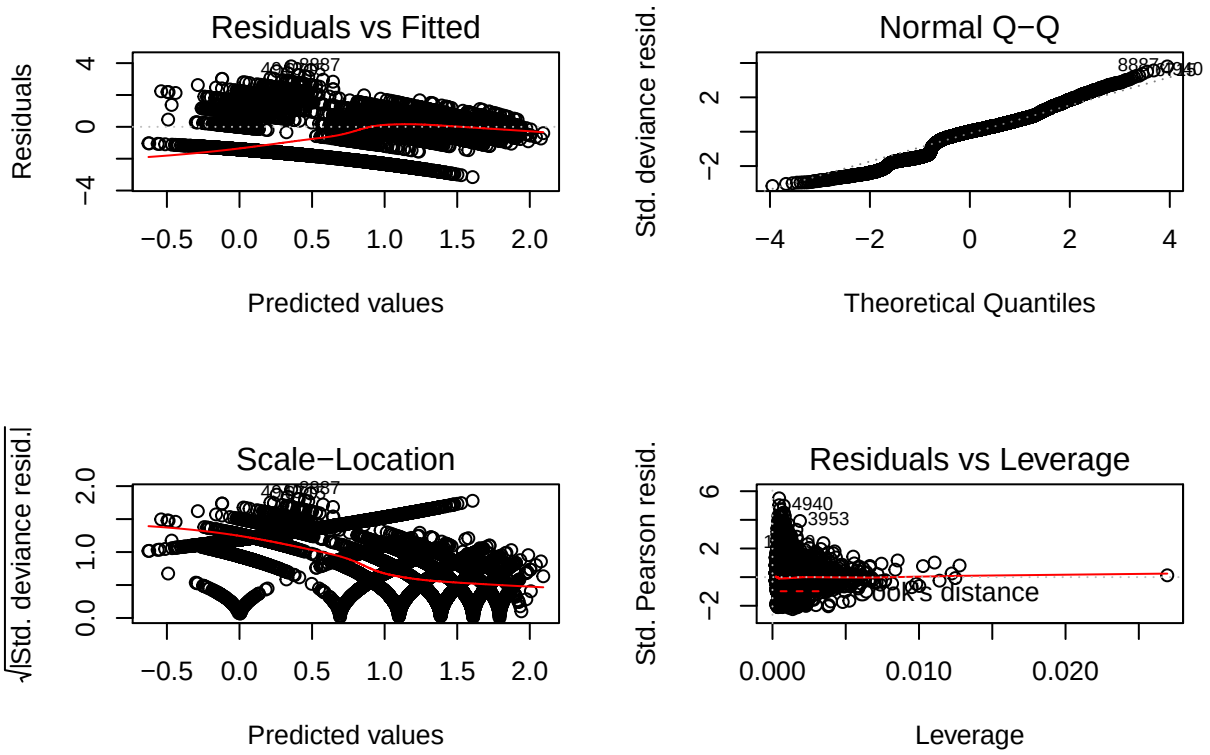
From above, we notice that the residuals vs fitted seems no to be homoescedastic, also the normal Q-Q line is

followed in certain areas, also we can notice how the p-values for some predictors, make them not statistically significant.

4.2.3 STEP Model

Let's build an automated STEP model in order to keep having a better understanding.

```
##
## Call:
## glm(formula = TARGET ~ STARS + STARS_NA + LabelAppeal + AcidIndex +
##     STARS_BETTER_SALES + log_VolatileAcidity + LabelAppealDISLIKE +
##     log_Chlorides + log_FreeSulfurDioxide + log_Sulphates, family = "poisson",
##     data = data.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1567  -0.6472   0.0047   0.4548   3.7985
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)             1.528791   0.103819  14.726  < 2e-16 ***
## STARS                   0.216849   0.007621  28.456  < 2e-16 ***
## STARS_NA               -0.599492   0.022867 -26.217  < 2e-16 ***
## LabelAppeal             0.138239   0.009026  15.316  < 2e-16 ***
## AcidIndex              -0.080762   0.004492 -17.978  < 2e-16 ***
## STARS_BETTER_SALES     -0.137094   0.023175  -5.916 3.31e-09 ***
## log_VolatileAcidity    -0.106567   0.023784  -4.481 7.44e-06 ***
## LabelAppealDISLIKE     -0.059791   0.019146  -3.123  0.00179 **
## log_Chlorides          -0.070014   0.027800  -2.518  0.01179 *
## log_FreeSulfurDioxide   0.026632   0.013160   2.024  0.04300 *
## log_Sulphates          -0.025961   0.016982  -1.529  0.12632
## ---
## 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: 13752  on 12784  degrees of freedom
## AIC: 45716
##
## Number of Fisher Scoring iterations: 6
```



From above, we can notice how the residuals vs the fitted values do not seems to be homoescedastic, also the normal q-q seems to follow in some how the given line.

4.2.3.1 ANOVA

Let's see the generated ANOVA table based on the above testing results.

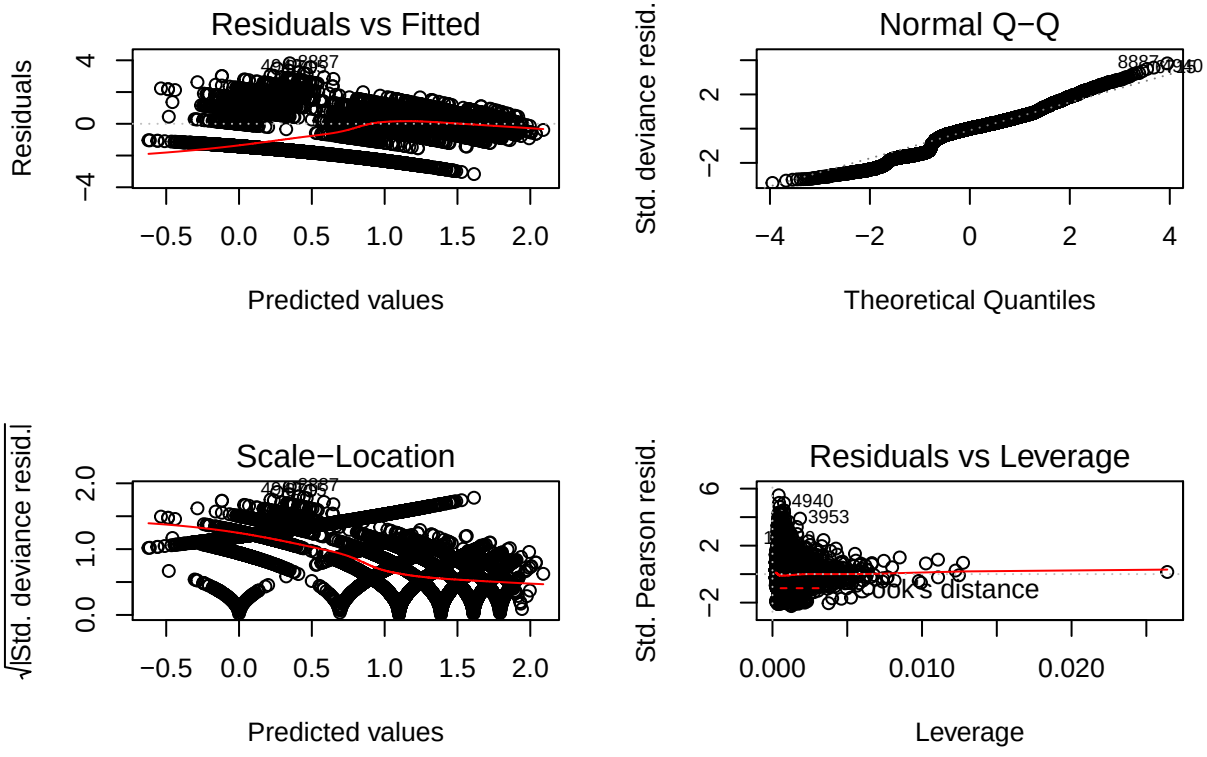
Step	Df	Deviance	Resid. Df	Resid. Dev	AIC
	NA	NA	12794	22860.89	54804.92
+ STARS	-1	7208.940067	12793	15651.95	47597.98
+ STARS_NA	-1	841.948279	12792	14810.01	46758.03
+ LabelAppeal	-1	625.569009	12791	14184.44	46134.46
+ AcidIndex	-1	352.691174	12790	13831.75	45783.77
+ STARS_BETTER_SALES	-1	36.890869	12789	13794.85	45748.88
+ log_VolatileAcidity	-1	19.737541	12788	13775.12	45731.14
+ LabelAppealDISLIKE	-1	9.883632	12787	13765.23	45723.26
+ log_Chlorides	-1	6.350533	12786	13758.88	45718.90
+ log_FreeSulfurDioxide	-1	4.102686	12785	13754.78	45716.80
+ log_Sulphates	-1	2.324815	12784	13752.46	45716.48

4.2.4 STEP Model Modified

In this section, I will build a modified model from the above STEP model. I will remove `log_Sulphates`.

```
##
## Call:
## glm(formula = TARGET ~ STARS + STARS_NA + LabelAppeal + AcidIndex +
##     STARS_BETTER_SALES + log_VolatileAcidity + LabelAppealDISLIKE +
##     log_Chlorides + log_FreeSulfurDioxide, family = "poisson",
##     data = data.train)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1687  -0.6477   0.0054   0.4562   3.8118
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)             1.491063   0.100843  14.786  < 2e-16 ***
## STARS                   0.216769   0.007621  28.445  < 2e-16 ***
## STARS_NA               -0.599850   0.022866 -26.233  < 2e-16 ***
## LabelAppeal             0.138148   0.009025  15.307  < 2e-16 ***
## AcidIndex              -0.080813   0.004492 -17.989  < 2e-16 ***
## STARS_BETTER_SALES     -0.136626   0.023173  -5.896 3.73e-09 ***
## log_VolatileAcidity    -0.106366   0.023784  -4.472 7.74e-06 ***
## LabelAppealDISLIKE     -0.060051   0.019146  -3.136  0.00171 **
## log_Chlorides          -0.069735   0.027796  -2.509  0.01211 *
## log_FreeSulfurDioxide   0.026479   0.013159   2.012  0.04420 *
## ---
## 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: 13755  on 12785  degrees of freedom
## AIC: 45717
##
## Number of Fisher Scoring iterations: 6
```



Once again, it seems that we are still getting a downward direction of the above residuals vs fitted; the normal q-q seems to follow the line with some values under captured in some sections of the line.

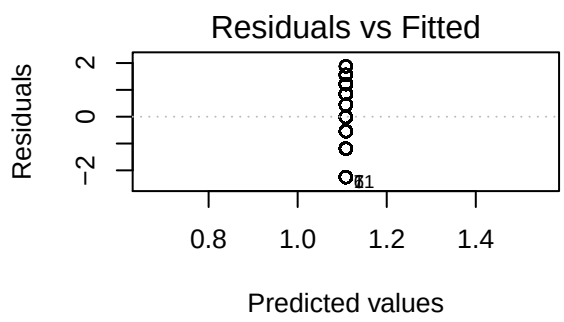
4.3 Negative Binomial Regression Models

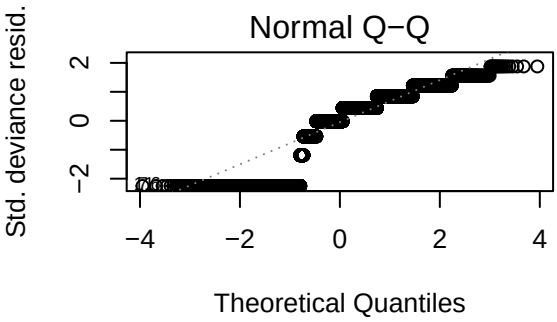
In this section I will present multiple models, build using the generalized linear model. Negative binomial regression can be used for over-dispersed count data, that is when the conditional variance exceeds the conditional mean. It can be considered as a generalization of Poisson regression since it has the same mean structure as Poisson regression and it has an extra parameter to model the over-dispersion. If the conditional distribution of the outcome variable is over-dispersed, the confidence intervals for the Negative binomial regression are likely to be narrower as compared to those from a Poisson regression model [1].

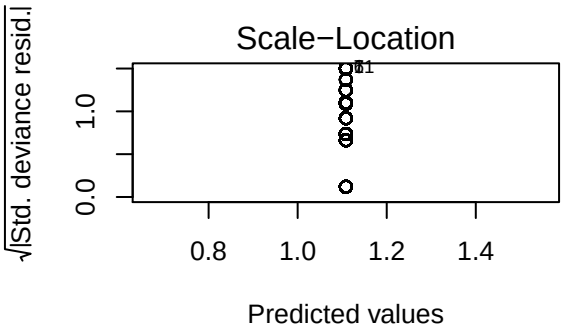
4.3.1 NULL Model

Let's start with a null model in order to have better understanding. This model will be considered to be valid and will be considered as we advance.

```
##
## Call:
## glm.nb(formula = TARGET ~ 1, data = data.train, init.theta = 7.339392284,
##     link = log)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.25205  -0.54038  -0.01408   0.44073   1.87709
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 1.108257   0.006037   183.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(7.3394) family taken to be 1)
##
##     Null deviance: 18129  on 12794  degrees of freedom
## Residual deviance: 18129  on 12794  degrees of freedom
## AIC: 54334
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  7.339
##           Std. Err.:  0.416
##
##  2 x log-likelihood:  -54329.582
```







In this particular case, we notice how the number of cases has decreased to about 1 for all the complete cases.

4.3.2 FULL Model

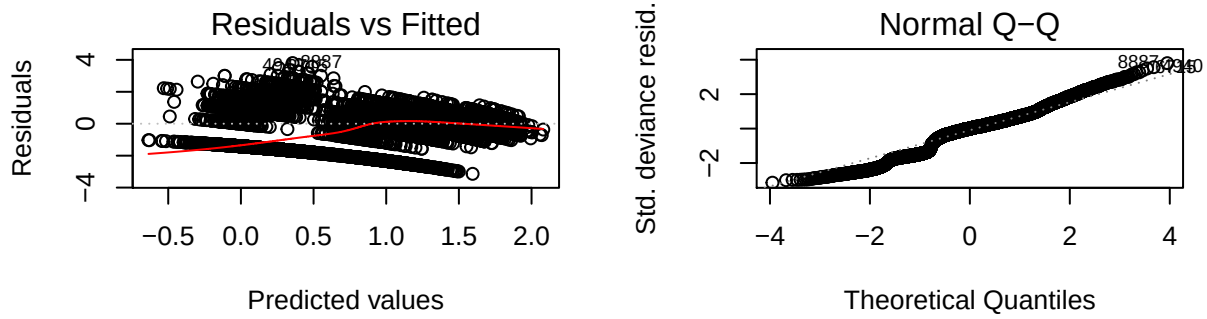
Let's build a full model in order to keep having a better understanding. This model will be considered to be valid and will be considered as we advance.

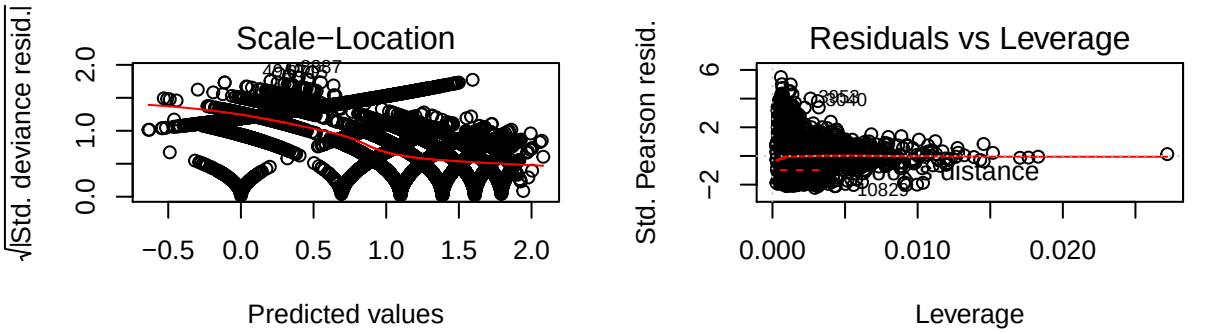
```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached

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

##
## Call:
## glm.nb(formula = TARGET ~ ., data = data.train, init.theta = 40599.31787,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1387  -0.6441   0.0038   0.4550   3.7991
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)               2.229937   0.602618   3.700 0.000215 ***
## LabelAppeal               0.142193   0.011632  12.224  < 2e-16 ***
## AcidIndex                -0.081016   0.004546 -17.820  < 2e-16 ***
## STARS                     0.216349   0.007629  28.360  < 2e-16 ***
## LabelAppealDISLIKE       -0.054541   0.021526  -2.534 0.011286 *
## LabelAppealBETTER_SALES -0.014582   0.028520  -0.511 0.609148
```

```
## STARS_BETTER_SALES      -0.137153   0.023185  -5.916 3.31e-09 ***
## STARS_NA                -0.599639   0.022876 -26.213  < 2e-16 ***
## log_FixedAcidity         0.004590   0.016996   0.270 0.787136
## log_VolatileAcidity     -0.105546   0.023794  -4.436 9.17e-06 ***
## log_CitricAcid           0.024299   0.024262   1.002 0.316575
## log_ResidualSugar        0.003133   0.013058   0.240 0.810393
## log_Chlorides           -0.069787   0.027819  -2.509 0.012120 *
## log_FreeSulfurDioxide    0.026622   0.013167   2.022 0.043184 *
## log_Density             -0.727249   0.552396  -1.317 0.187994
## log_PH                  -0.024961   0.030076  -0.830 0.406570
## log_Sulphates           -0.026457   0.016990  -1.557 0.119423
## log_Alcohol              0.015166   0.015445   0.982 0.326128
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(40599.32) family taken to be 1)
##
##     Null deviance: 22860  on 12794  degrees of freedom
## Residual deviance: 13747  on 12777  degrees of freedom
## AIC: 45728
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  40599
##           Std. Err.:  34382
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -45690.04
```



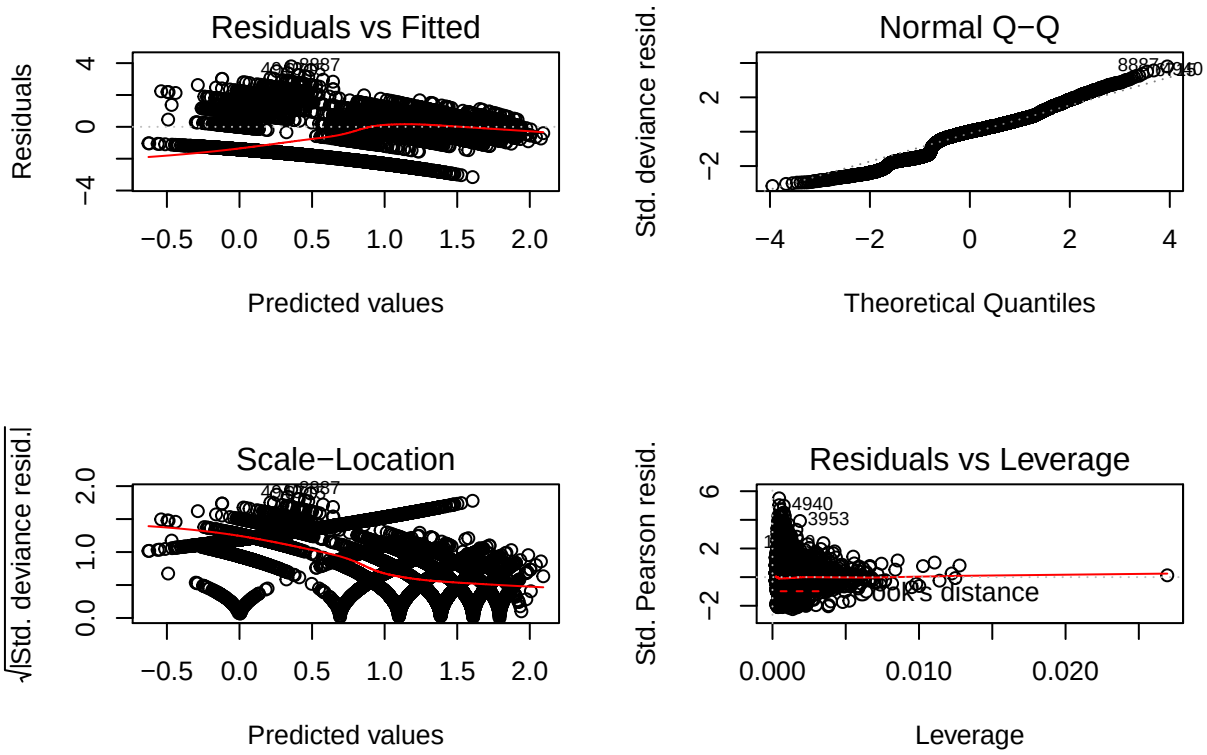


From above, we notice that the residuals vs fitted seems no to be homoescedastic, also the normal Q-Q line is followed in certain areas, also we can notice how the p-values for some predictors, make them not statistically significant.

4.3.3 STEP Model

Let's build an automated STEP model in order to keep having a better understanding.

```
##
## Call:
## glm.nb(formula = TARGET ~ STARS + STARS_NA + LabelAppeal + AcidIndex +
##     STARS_BETTER_SALES + log_VolatileAcidity + LabelAppealDISLIKE +
##     log_Chlorides + log_FreeSulfurDioxide + log_Sulphates, data = data.train,
##     init.theta = 40568.84259, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1566  -0.6472   0.0047   0.4548   3.7984
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)            1.528810   0.103823  14.725  < 2e-16 ***
## STARS                  0.216851   0.007621  28.454  < 2e-16 ***
## STARS_NA              -0.599488   0.022868 -26.216  < 2e-16 ***
## LabelAppeal            0.138238   0.009026  15.315  < 2e-16 ***
## AcidIndex             -0.080764   0.004492 -17.978  < 2e-16 ***
## STARS_BETTER_SALES    -0.137096   0.023176  -5.915 3.31e-09 ***
## log_VolatileAcidity   -0.106570   0.023785  -4.481 7.44e-06 ***
## LabelAppealDISLIKE    -0.059791   0.019147  -3.123  0.00179 **
## log_Chlorides         -0.070017   0.027802  -2.518  0.01179 *
## log_FreeSulfurDioxide  0.026633   0.013161   2.024  0.04301 *
## log_Sulphates         -0.025963   0.016982  -1.529  0.12632
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(40568.84) family taken to be 1)
##
##     Null deviance: 22860  on 12794  degrees of freedom
## Residual deviance: 13752  on 12784  degrees of freedom
## AIC: 45719
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  40569
##           Std. Err.:  34357
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -45694.89
```



From above, we can notice how the residuals vs the fitted values do not seems to be homoscedastic, also the normal q-q seems to follow in some how the given line.

4.3.3.1 ANOVA

Let's see the generated ANOVA table based on the above testing results.

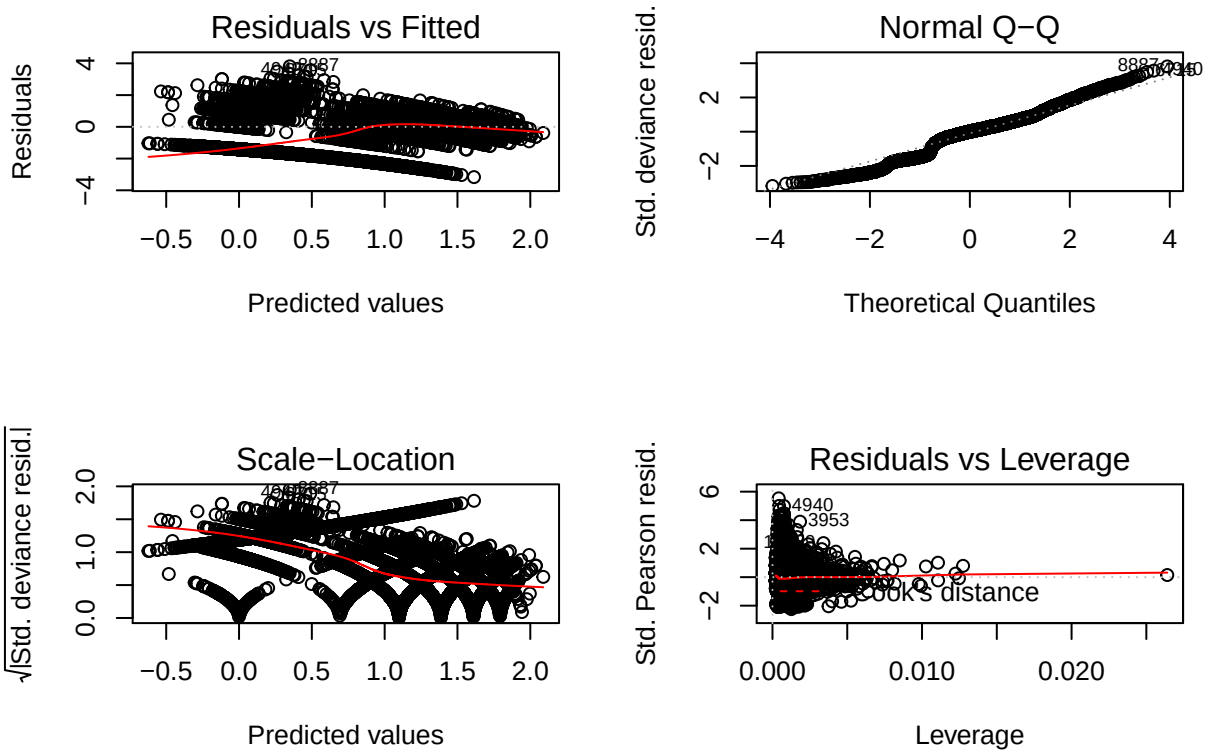
Step	Df	Deviance	Resid. Df	Resid. Dev	AIC
	NA	NA	12794	18128.90	54331.58
+ STARS	-1	2477.592210	12793	15651.30	47598.24
+ STARS_NA	-1	842.018333	12792	14809.29	46758.42
+ LabelAppeal	-1	625.425727	12791	14183.86	46134.87
+ AcidIndex	-1	352.664391	12790	13831.20	45784.18
+ STARS_BETTER_SALES	-1	36.886219	12789	13794.31	45749.29
+ log_VolatileAcidity	-1	19.735927	12788	13774.57	45731.55
+ LabelAppealDISLIKE	-1	9.882530	12787	13764.69	45723.67
+ log_Chlorides	-1	6.349861	12786	13758.34	45719.32
+ log_FreeSulfurDioxide	-1	4.102220	12785	13754.24	45717.22
+ log_Sulphates	-1	2.324681	12784	13751.91	45716.89

4.3.4 STEP Model Modified

In this section, I will build a modified model from the above STEP model. I will remove `log_Sulphates`.

```
##
## Call:
## glm.nb(formula = TARGET ~ STARS + STARS_NA + LabelAppeal + AcidIndex +
##     STARS_BETTER_SALES + log_VolatileAcidity + LabelAppealDISLIKE +
##     log_Chlorides + log_FreeSulfurDioxide, data = data.train,
##     init.theta = 40558.5662, link = log)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1686  -0.6477   0.0054   0.4562   3.8116
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)             1.491081   0.100848  14.785  < 2e-16 ***
## STARS                   0.216771   0.007621  28.444  < 2e-16 ***
## STARS_NA               -0.599846   0.022867 -26.233  < 2e-16 ***
## LabelAppeal             0.138147   0.009026  15.306  < 2e-16 ***
## AcidIndex              -0.080815   0.004492 -17.989  < 2e-16 ***
## STARS_BETTER_SALES     -0.136628   0.023174  -5.896 3.73e-09 ***
## log_VolatileAcidity    -0.106369   0.023785  -4.472 7.74e-06 ***
## LabelAppealDISLIKE     -0.060050   0.019147  -3.136  0.00171 **
## log_Chlorides          -0.069738   0.027797  -2.509  0.01211 *
## log_FreeSulfurDioxide   0.026480   0.013160   2.012  0.04421 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(40558.57) family taken to be 1)
##
##     Null deviance: 22860  on 12794  degrees of freedom
## Residual deviance: 13754  on 12785  degrees of freedom
## AIC: 45719
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  40559
##           Std. Err.:  34344
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -45697.22
```



Once again, it seems that we are still getting a downward direction of the above residuals vs fitted; the normal q-q seems to follow the line with some values under captured in some sections of the line.

4.4 Comparing Poisson vs Negative Binomial

It is interesting to note that the Poisson and the Negative Binomial models return the same results up to 4 decimals. We can be compare as follows:

4.4.1 Comparing FULL Models

Let's compare both FULL models; that is the coefficients generated from the Poisson FULL model vs the coefficients returned by the Negative Binomial FULL Model.

	Poisson	Negative Binomial	Similar Coefficients
(Intercept)	-2.2299	-2.2299	TRUE
LabelAppeal	-0.1422	-0.1422	TRUE
AcidIndex	0.0810	0.0810	TRUE
STARS	-0.2163	-0.2163	TRUE
LabelAppealDISLIKE	0.0545	0.0545	TRUE
LabelAppealBETTER_SALES	0.0146	0.0146	TRUE
STARS_BETTER_SALES	0.1372	0.1372	TRUE
STARS_NA	0.5996	0.5996	TRUE
log_FixedAcidity	-0.0046	-0.0046	TRUE
log_VolatileAcidity	0.1055	0.1055	TRUE
log_CitricAcid	-0.0243	-0.0243	TRUE
log_ResidualSugar	-0.0031	-0.0031	TRUE
log_Chlorides	0.0698	0.0698	TRUE
log_FreeSulfurDioxide	-0.0266	-0.0266	TRUE
log_Density	0.7272	0.7272	TRUE
log_PH	0.0250	0.0250	TRUE
log_Sulphates	0.0265	0.0265	TRUE
log_Alcohol	-0.0152	-0.0152	TRUE

4.4.2 Comparing STEP Models

Let's compare both STEP models; that is the coefficients generated from the Poisson STEP model vs the coefficients returned by the Negative Binomial STEP Model.

	Poisson	Negative Binomial	Similar Coefficients
(Intercept)	-1.5288	-1.5288	TRUE
STARS	-0.2168	-0.2169	Rounding Aproximation
STARS_NA	0.5995	0.5995	TRUE
LabelAppeal	-0.1382	-0.1382	TRUE
AcidIndex	0.0808	0.0808	TRUE
STARS_BETTER_SALES	0.1371	0.1371	TRUE
log_VolatileAcidity	0.1066	0.1066	TRUE
LabelAppealDISLIKE	0.0598	0.0598	TRUE
log_Chlorides	0.0700	0.0700	TRUE
log_FreeSulfurDioxide	-0.0266	-0.0266	TRUE
log_Sulphates	0.0260	0.0260	TRUE

4.4.3 Comparing STEP Models Modified

Let's compare both STEP models Modified; that is the coefficients generated from the Poisson STEP model Modified vs the coefficients returned by the Negative Binomial STEP Model Modified.

	Poisson	Negative Binomial	Similar Coefficients
(Intercept)	-1.4911	-1.4911	TRUE
STARS	-0.2168	-0.2168	TRUE
STARS_NA	0.5998	0.5998	TRUE
LabelAppeal	-0.1381	-0.1381	TRUE
AcidIndex	0.0808	0.0808	TRUE
STARS_BETTER_SALES	0.1366	0.1366	TRUE
log_VolatileAcidity	0.1064	0.1064	TRUE
LabelAppealDISLIKE	0.0601	0.0601	TRUE
log_Chlorides	0.0697	0.0697	TRUE
log_FreeSulfurDioxide	-0.0265	-0.0265	TRUE

4.5 Zero-Inflated Poisson Regression Models

In this section I will present multiple models, build using the `pscl` package in order to build a Zero-inflated Poisson regression linear model. Zero-inflated Poisson regression is used to model count data that has an excess of zero counts. Further, theory suggests that the excess zeros are generated by a separate process from the count values and that the excess zeros can be modeled independently. Thus, the generated model has two parts, a Poisson count model and the logit model for predicting excess zeros [2].

4.5.1 NULL Model

Let's start with a null model in order to have better understanding. This model will be considered to be valid and will be considered as we advance.

```
##
## Call:
## zeroinfl(formula = TARGET ~ 1, data = data.train, dist = "poisson")
##
## Pearson residuals:
##      Min       1Q   Median       3Q      Max
## -1.32182 -0.44906 -0.01268  0.42370  2.16921
##
## Count model coefficients (poisson with log link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 1.325142   0.005323   248.9   <2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.41797    0.02376  -59.68   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 6
## Log-likelihood: -2.45e+04 on 2 Df
```

In this case is interesting to note how the Zero-Inflated model actually lowered the intercept from 1 case to about -1 case, making this model not realistic based on our assumptions that we need at least ZERO cases for the TARGET variable.

4.5.2 FULL Model

Let's build a full model in order to keep having a better understanding. This model will be considered to be valid and will be considered as we advance.

```
##
## Call:
## zeroinfl(formula = TARGET ~ ., data = data.train, dist = "poisson")
##
## Pearson residuals:
##       Min        1Q    Median        3Q       Max
## -2.280994 -0.428763  0.004488  0.387370  6.121471
##
## Count model coefficients (poisson with log link):
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               1.9279276  0.6189709   3.115 0.001841 **
## LabelAppeal               0.2103146  0.0118591  17.735  < 2e-16 ***
## AcidIndex                -0.0196744  0.0048710  -4.039 5.37e-05 ***
## STARS                     0.1098063  0.0080132  13.703  < 2e-16 ***
## LabelAppealDISLIKE       -0.0923529  0.0222150  -4.157 3.22e-05 ***
## LabelAppealBETTER_SALES -0.0643562  0.0287777  -2.236 0.025331 *
## STARS_BETTER_SALES       -0.0198456  0.0234867  -0.845 0.398125
## STARS_NA                  0.0388791  0.0245474   1.584 0.113231
## log_FixedAcidity          0.0048219  0.0176521   0.273 0.784727
## log_VolatileAcidity      -0.0426647  0.0245854  -1.735 0.082676 .
## log_CitricAcid            0.0069503  0.0247135   0.281 0.778530
## log_ResidualSugar        -0.0040702  0.0132890  -0.306 0.759391
## log_Chlorides            -0.0326467  0.0285149  -1.145 0.252250
## log_FreeSulfurDioxide     0.0059070  0.0132194   0.447 0.654988
## log_Density              -0.7665704  0.5691258  -1.347 0.178004
## log_PH                    0.0295745  0.0312337   0.947 0.343701
## log_Sulphates            -0.0008668  0.0174768  -0.050 0.960443
## log_Alcohol               0.0523539  0.0158924   3.294 0.000987 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -4.23014    4.07983  -1.037 0.299809
## LabelAppeal                 0.74282    0.08150   9.114  < 2e-16 ***
## AcidIndex                   0.43581    0.02604  16.736  < 2e-16 ***
## STARS                      -3.82091    0.34039 -11.225  < 2e-16 ***
## LabelAppealDISLIKE         -0.04061    0.13800  -0.294 0.768575
## LabelAppealBETTER_SALES    -0.32440    0.24933  -1.301 0.193226
## STARS_BETTER_SALES         -8.90046 1792.60192  -0.005 0.996038
## STARS_NA                   -1.75930    0.35670  -4.932 8.13e-07 ***
## log_FixedAcidity           -0.04208    0.11670  -0.361 0.718410
## log_VolatileAcidity         0.71565    0.16783   4.264 2.01e-05 ***
## log_CitricAcid             -0.11256    0.16119  -0.698 0.484962
## log_ResidualSugar          -0.05852    0.08606  -0.680 0.496521
## log_Chlorides               0.31493    0.18600   1.693 0.090410 .
## log_FreeSulfurDioxide      -0.20619    0.08748  -2.357 0.018421 *
## log_Density                 1.28436    3.72476   0.345 0.730232
## log_PH                      0.69331    0.20669   3.354 0.000796 ***
## log_Sulphates               0.24209    0.11408   2.122 0.033839 *
## log_Alcohol                 0.30403    0.10500   2.896 0.003785 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 46
## Log-likelihood: -2.04e+04 on 36 Df
```

In this particular case is interesting to notice how the two models return different coefficients and different statistical significance for many variables.

4.6 Zero-Inflated Negative Binomial Regression Models

In this section I will present multiple models, build using the `pscl` package in order to build a Zero-inflated Negative Binomial regression linear model. Zero-inflated negative binomial regression is for modeling count variables with excessive zeros and it is usually for over-dispersed count outcome variables. Furthermore, theory suggests that the excess zeros are generated by a separate process from the count values and that the excess zeros can be modeled independently [3].

A zero-inflated model assumes that zero outcome is due to two different processes. The two parts of the a zero-inflated model are a binary model, usually a logit model to model which of the two processes the zero outcome is associated with and a count model, in this case, a negative binomial model, to model the count process. The expected count is expressed as a combination of the two processes.

4.6.1 NULL Model

Let's start with a null model in order to have better understanding. This model will be considered to be valid and will be considered as we advance.

```
##
## Call:
## zeroinfl(formula = TARGET ~ 1, data = data.train, dist = "negbin")
##
## Pearson residuals:
##      Min       1Q   Median       3Q      Max
## -1.32181 -0.44907 -0.01269  0.42368  2.16918
##
## Count model coefficients (negbin with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.325156   0.005323 248.940   <2e-16 ***
## Log(theta)  13.390509   8.252281   1.623    0.105
##
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.41793    0.02376  -59.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 653768.5341
## Number of iterations in BFGS optimization: 22
## Log-likelihood: -2.45e+04 on 3 Df
```

In this case is interesting to note how the Zero-Inflated model actually lowered the intercept from 1 case to about -1 case, making this model not realistic based on our assumptions that we need at least ZERO cases for the TARGET variable.

4.6.2 FULL Model

Let's build a full model in order to keep having a better understanding. This model will be considered to be valid and will be considered as we advance.

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

```
##
## Call:
## zeroinfl(formula = TARGET ~ ., data = data.train, dist = "negbin")
##
## Pearson residuals:
##       Min        1Q    Median        3Q       Max
## -2.280999 -0.428747  0.004457  0.387304  6.122829
##
## Count model coefficients (negbin with log link):
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)              1.9278712  0.6189716   3.115 0.001842 **
## LabelAppeal              0.2103203  0.0118591  17.735  < 2e-16 ***
## AcidIndex               -0.0196751  0.0048710  -4.039 5.36e-05 ***
## STARS                    0.1098011  0.0080133  13.702  < 2e-16 ***
## LabelAppealDISLIKE      -0.0923447  0.0222151  -4.157 3.23e-05 ***
## LabelAppealBETTER_SALES -0.0643743  0.0287778  -2.237 0.025290 *
## STARS_BETTER_SALES      -0.0198395  0.0234867  -0.845 0.398272
## STARS_NA                 0.0388549  0.0245477   1.583 0.113459
## log_FixedAcidity         0.0048217  0.0176522   0.273 0.784736
## log_VolatileAcidity     -0.0426963  0.0245853  -1.737 0.082447 .
## log_CitricAcid           0.0069572  0.0247136   0.282 0.778318
## log_ResidualSugar       -0.0040714  0.0132890  -0.306 0.759320
## log_Chlorides           -0.0326518  0.0285149  -1.145 0.252176
## log_FreeSulfurDioxide    0.0059020  0.0132193   0.446 0.655261
## log_Density             -0.7664342  0.5691263  -1.347 0.178081
## log_PH                   0.0295668  0.0312337   0.947 0.343826
## log_Sulphates           -0.0008797  0.0174768  -0.050 0.959856
## log_Alcohol              0.0523659  0.0158925   3.295 0.000984 ***
## Log(theta)              17.9477081         NA      NA       NA
##
## Zero-inflation model coefficients (binomial with logit link):
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -4.23979    4.08015  -1.039 0.298746
## LabelAppeal                0.74291    0.08151   9.114  < 2e-16 ***
## AcidIndex                  0.43586    0.02604  16.737  < 2e-16 ***
## STARS                     -3.82120    0.34052 -11.222  < 2e-16 ***
## LabelAppealDISLIKE        -0.04076    0.13801  -0.295 0.767715
## LabelAppealBETTER_SALES   -0.32418    0.24934  -1.300 0.193548
## STARS_BETTER_SALES        -8.90088 1793.85016  -0.005 0.996041
## STARS_NA                  -1.75928    0.35683  -4.930 8.21e-07 ***
## log_FixedAcidity          -0.04210    0.11670  -0.361 0.718273
## log_VolatileAcidity        0.71563    0.16784   4.264 2.01e-05 ***
## log_CitricAcid            -0.11249    0.16120  -0.698 0.485291
## log_ResidualSugar         -0.05854    0.08606  -0.680 0.496368
## log_Chlorides              0.31507    0.18601   1.694 0.090295 .
## log_FreeSulfurDioxide     -0.20625    0.08749  -2.357 0.018400 *
## log_Density                1.29365    3.72505   0.347 0.728377
## log_PH                     0.69321    0.20671   3.354 0.000798 ***
```

```
## log_Sulphates              0.24211    0.11409   2.122 0.033836 *
## log_Alcohol                0.30398    0.10501   2.895 0.003793 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 62314710.7528
## Number of iterations in BFGS optimization: 68
## Log-likelihood: -2.04e+04 on 37 Df
```

In this particular case is interesting to notice how the two models return different coefficients and different statistical significance for many variables.

Also, something considerable to mention is the similitude of the coefficients on both models the Zero-Inflated Poison model and the Zero-Inflated Negative Binomial.

4.7 Comparing Zero-Inflated Model Coefficients

In this section I will proceed to make a small comparison of coefficients for both Zero inflated models that is The Zero-Inflated Poisson FULL Model and the Zero-Inflated Negative-Binomial FULL Model. In this case, we can see how the differences keep up to 3, 4 or even five or mode decimals.

	Poisson	Neg Binomial	Difference
(Intercept)	-4.2301417	-4.2397876	0.0096
LabelAppeal	0.7428173	0.7429074	-0.0001
AcidIndex	0.4358056	0.4358597	-0.0001
STARS	-3.8209103	-3.8211956	0.0003
LabelAppealDISLIKE	-0.0406051	-0.0407637	0.0002
LabelAppealBETTER_SALES	-0.3243972	-0.3241850	-0.0002
STARS_BETTER_SALES	-8.9004591	-8.9008810	0.0004
STARS_NA	-1.7593041	-1.7592758	0.0000
log_FixedAcidity	-0.0420786	-0.0421034	0.0000
log_VolatileAcidity	0.7156499	0.7156291	0.0000
log_CitricAcid	-0.1125642	-0.1124876	-0.0001
log_ResidualSugar	-0.0585178	-0.0585426	0.0000
log_Chlorides	0.3149339	0.3150698	-0.0001
log_FreeSulfurDioxide	-0.2061945	-0.2062462	0.0001
log_Density	1.2843630	1.2936530	-0.0093
log_PH	0.6933116	0.6932087	0.0001
log_Sulphates	0.2420856	0.2421065	0.0000
log_Alcohol	0.3040346	0.3039806	0.0001

What is interesting once again are the similar values returned on both models. Also, by looking at the p-values and the level of significance, we notice close similarities.

5 MODEL SELECTION

In this section, I will describe the process in order to select the final model.

In particular case, I would like to select the `Linear Model STEP Modified` as my final model. The reasons are as follows:

- The `Linear Model STEP Modified` follows most of the normal quantile to quantile plot.
- The quantile to quantile plot deviated just a little bit towards the top.

- The Median is considered to be near zero.
- The R^2 is some how significant in terms that this model describe about 54% of the variation of the data.
- Even though there's a lot of problems in terms of correlation with the data, I believe this to be the "best" model from the ones that I have previously build.
- However, the Poisson model is a great choice for counts, it seems to follow the trends as well.
- The Median in the Poisson model shows to be near zero.
- Is very difficult to decide at this point; let's run both models and see what might come up.

6 PREDICTIONS

In this section, I will proceed to predict values from the evaluation data set.

6.1 Evaluation data transformations

In this section I will transform our evaluation data same as our original data has.

```
# Let's create a backup in order to transform our data
df <- data.eval

# Create New Variable: LabelAppealDISLIKE
df <- create_LabelAppealDISLIKE(df)

# Create New Variable: LabelAppealBETTER_SALES
df <- create_LabelAppealBETTER_SALES(df)

# Create New Variable: STARS_BETTER_SALES
df <- create_STARS_BETTER_SALES(df)

# Create New Variable: STARS_NA
df <- create_STARS_NA(df)

# Fill in missing NAs
df <- fill_missing_na(df)

# Transform variables into "log_var = log(|min(var)| + var + 1)"
df <- transform_vars(df)

# Extracting new and transformed columns
df <- df[c(1:2,14:30)]
```

6.2 Predict TARGET

In this section, I will predict the number of cases based on the final linear model selected. In order to accomplish this goal, I need to do as follows:

First, I need to predict the number of cases and the I need to round the final results to zero decimals since the goal is to predict whole cases.

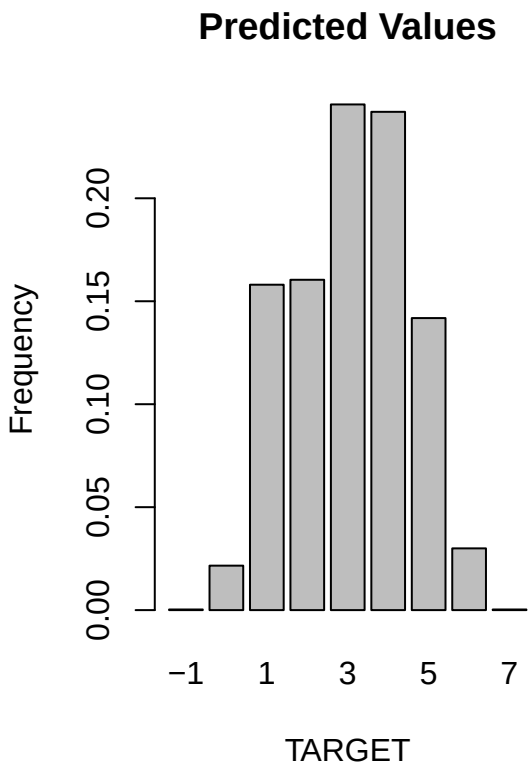
6.2.1 Predicted table: Linear STEP Modified Model

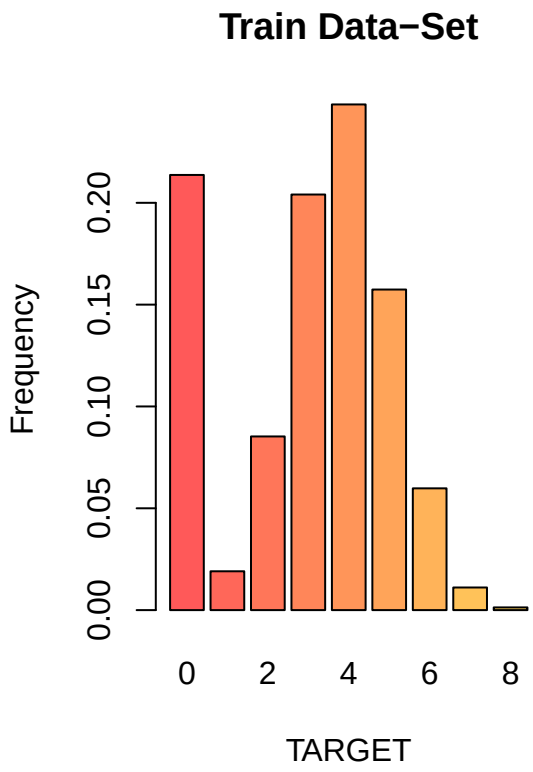
The below is a table in which the records show the generated predicted number of cases for the first 20 records employing the linear STEP Modified model.

IN	TARGET	FixedAcidity	VolatileAcidity	CitricAcid	ResidualSugar	Chlorides	FreeSulfurDioxide
3	1	5.4	-0.860	0.27	-10.7	0.092	23
9	4	12.4	0.385	-0.76	-19.7	1.169	-37
10	3	7.2	1.750	0.17	-33.0	0.065	9
18	2	6.2	0.100	1.80	1.0	-0.179	104
21	1	11.4	0.210	0.28	1.2	0.038	70
30	5	17.6	0.040	-1.15	1.4	0.535	-250
31	3	15.5	0.530	-0.53	4.6	1.263	10
37	2	15.9	1.190	1.14	31.9	-0.299	115
39	0	11.6	0.320	0.55	-50.9	0.076	35
47	2	3.8	0.220	0.31	-7.7	0.039	40
60	3	6.8	1.680	0.44	-13.3	0.046	NA
62	1	9.0	-0.210	0.04	51.4	0.237	-213
63	3	24.6	0.030	-1.20	1.3	0.035	241
64	1	13.0	0.210	0.32	-3.2	-0.263	111
68	1	17.9	-0.420	-0.91	7.1	0.045	-177
75	2	10.0	0.200	1.27	30.9	0.050	19
76	2	7.4	0.290	0.50	8.5	-0.480	178
83	1	11.7	1.180	-0.94	-62.0	0.675	7
87	4	9.7	0.410	-1.00	NA	-0.235	24
92	6	-5.2	-0.980	-0.08	6.4	0.046	180

6.2.1.1 Visuals

In this section, I am presenting the compared graphs for the predicted values vs the values from the training data-set.





6.2.1.2 Summaries

In this section I am presenting some count summaries by predicted `TARGET` which represents the number of cases predicted from the evaluation data-set.

TARGET	count
-1	1
0	72
1	527
2	535
3	819
4	807
5	473
6	100
7	1

It is interesting to note the presence of 1 case in which shows the predicted value of -1; this is considered not to be a true value since we can not order -1 cases unless the product is so bad that we need to return the one case of wine.

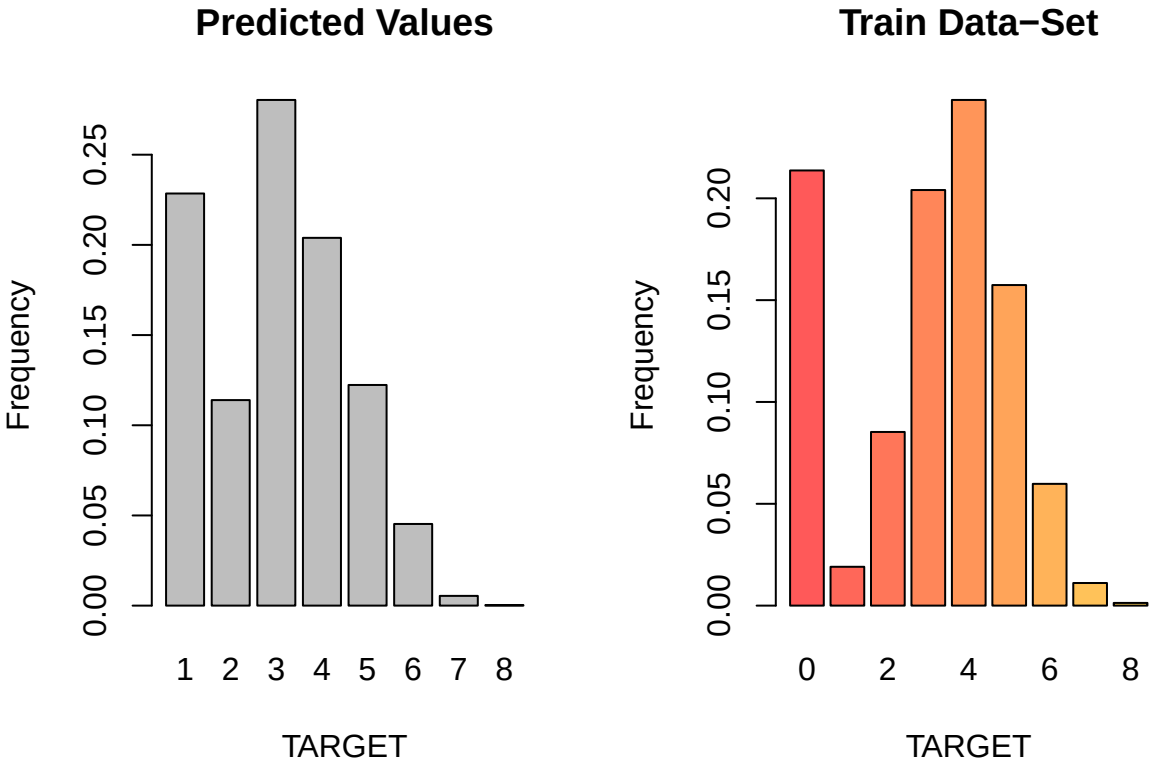
6.2.2 Predicted table: Poisson Modified Model

The below is a table in which the records show the generated predicted number of cases for the first 20 records employing the Poisson Modified model.

IN	TARGET	FixedAcidity	VolatileAcidity	CitricAcid	ResidualSugar	Chlorides	FreeSulfurDioxide
3	1	5.4	-0.860	0.27	-10.7	0.092	23
9	4	12.4	0.385	-0.76	-19.7	1.169	-37
10	3	7.2	1.750	0.17	-33.0	0.065	9
18	2	6.2	0.100	1.80	1.0	-0.179	104
21	1	11.4	0.210	0.28	1.2	0.038	70
30	5	17.6	0.040	-1.15	1.4	0.535	-250
31	3	15.5	0.530	-0.53	4.6	1.263	10
37	2	15.9	1.190	1.14	31.9	-0.299	115
39	1	11.6	0.320	0.55	-50.9	0.076	35
47	1	3.8	0.220	0.31	-7.7	0.039	40
60	3	6.8	1.680	0.44	-13.3	0.046	NA
62	1	9.0	-0.210	0.04	51.4	0.237	-213
63	3	24.6	0.030	-1.20	1.3	0.035	241
64	1	13.0	0.210	0.32	-3.2	-0.263	111
68	1	17.9	-0.420	-0.91	7.1	0.045	-177
75	2	10.0	0.200	1.27	30.9	0.050	19
76	2	7.4	0.290	0.50	8.5	-0.480	178
83	1	11.7	1.180	-0.94	-62.0	0.675	7
87	4	9.7	0.410	-1.00	NA	-0.235	24
92	7	-5.2	-0.980	-0.08	6.4	0.046	180

6.2.2.1 Visuals

In this section, I am presenting the compared graphs for the predicted values vs the values from the training data-set.



6.2.2.2 Summaries

In this section I am presenting some count summaries by predicted `TARGET` which represents the number of cases predicted from the evaluation data-set.

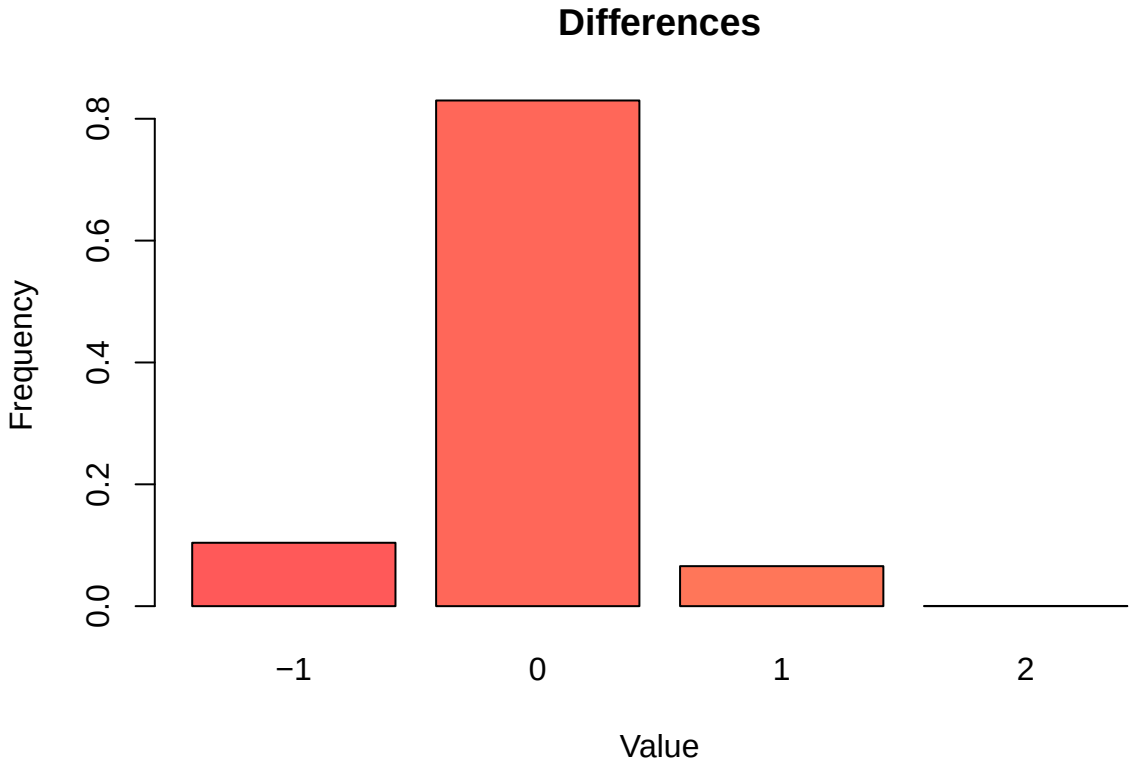
TARGET	count
1	762
2	380
3	935
4	680
5	408
6	151
7	18
8	1

It is interesting to note that in this case, the Poisson STEP Modified Model shows a better fit for the values that we are looking, that is in between ZERO and EIGHT. It is also notable that in this case the number of cases with the value of 1 have increased considerable compared to our linear model just in a tiny fraction while the other values seems to follow accordingly.

6.3 Comparing predictions

In this section, I will compare the predictions from the Linear model and the Poisson model; the theoretical effect should be of a small difference that is, maybe one digit in difference in between cases.

Let's see the frequency of differences.



Let's take a look at the counts.

Difference	count
-1	347
0	2768
1	219
2	1

In effect, those results display that even though there's only one value with two units difference, is actually a good approximation; we notice how the majority of the values fit very well and the maximum difference in cases is only one, that is one short or one over; which could be perceived as a very good approximation.

7 FINAL MODEL

From above, I will pick the `Poisson STEP Modified Model` as my final model; as previously compared in the ANOVA table, that model shows a low AIC and follows the normal quantile to quantile line very accurately, also, from the above summaries, we noticed that it provides values for all desired counts. Perhaps it could be refined even further. Also, is remarkable to note that similitude with the Negative Binomial STEP Modified Model, which could be picked as well but for consistency reasons I will stick with the Poisson STEP Modified Model.

```
selected_FINAL_MODEL <- glm_P_Model_STEP_Modified
```

7.1 Export file

In order to provide a csv output for the predictions table.

```
write.csv(data.eval, file = "wine-my-evaluated-data.csv",row.names=FALSE)
```

8 REFERENCES

[1] https://stats.idre.ucla.edu/r/dae/negative-binomial-regression/

[2] https://stats.idre.ucla.edu/r/dae/zip/

[3] https://stats.idre.ucla.edu/r/dae/zinb/