DATA 621 - HW5

Andrew Bowen, Glen Davis, Shoshana Farber, Joshua Forster, Charles Ugiagbe

2023-11-27

Homework 5 - Count Regression

Data Exploration

The dataset to be used in this analysis involves sales of over 12,000 different types of commercially available wine. There are 12,795 records across the training set with a response variable `TARGET` that indicates the number of sample cases purchased by wine distribution companies. It is generally more appropriate to use poisson or negative binomial regression methods to predict a discrete dependent variable and different variations will be explored later in the analysis.

Below is a short description of all the variables of interest in the data set, including these response variables:

VARIABLE NAME	DEFINITION
`INDEX`	Identification Variable
`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 customers don't like the design.
`RESIDUAL SUGAR`	Residual Sugar of Wine
`STARS`	Win rating by a team of experts
`SULPHATES`	Sulfate content of Wine
`TOTLASULFURDIOXIDE`	Total Sulfur Dioxide of Wine
`VOLATILEACIDITY`	Volatile Acide content of Wine
`PH`	pH Level of Wine

Let's review the basic numeric distributions from the summary of the dataframe:

```
##      INDEX           TARGET       FixedAcidity     VolatileAcidity
##  Min.   :    1   Min.   :0.000   Min.   :-18.100   Min.   :-2.7900
##  1st Qu.: 4038   1st Qu.:2.000   1st Qu.:  5.200   1st Qu.: 0.1300
##  Median : 8110   Median :3.000   Median :  6.900   Median : 0.2800
```

```
##  Mean   : 8070   Mean   :3.029   Mean   :  7.076   Mean   : 0.3241
##  3rd Qu.:12106   3rd Qu.:4.000   3rd Qu.:  9.500   3rd Qu.: 0.6400
##  Max.   :16129   Max.   :8.000   Max.   : 34.400   Max.   : 3.6800
##
##    CitricAcid      ResidualSugar        Chlorides       FreeSulfurDioxide
##  Min.   :-3.2400   Min.   :-127.800   Min.   :-1.1710   Min.   :-555.00
##  1st Qu.: 0.0300   1st Qu.:  -2.000   1st Qu.:-0.0310   1st Qu.:   0.00
##  Median : 0.3100   Median :   3.900   Median : 0.0460   Median :  30.00
##  Mean   : 0.3084   Mean   :   5.419   Mean   : 0.0548   Mean   :  30.85
##  3rd Qu.: 0.5800   3rd Qu.:  15.900   3rd Qu.: 0.1530   3rd Qu.:  70.00
##  Max.   : 3.8600   Max.   : 141.150   Max.   : 1.3510   Max.   : 623.00
##                    NA's   :616        NA's   :638       NA's   :647
##  TotalSulfurDioxide    Density             pH          Sulphates
##  Min.   :-823.0     Min.   :0.8881   Min.   :0.480   Min.   :-3.1300
##  1st Qu.:  27.0     1st Qu.:0.9877   1st Qu.:2.960   1st Qu.: 0.2800
##  Median : 123.0     Median :0.9945   Median :3.200   Median : 0.5000
##  Mean   : 120.7     Mean   :0.9942   Mean   :3.208   Mean   : 0.5271
##  3rd Qu.: 208.0     3rd Qu.:1.0005   3rd Qu.:3.470   3rd Qu.: 0.8600
##  Max.   :1057.0     Max.   :1.0992   Max.   :6.130   Max.   : 4.2400
##  NA's   :682                         NA's   :395     NA's   :1210
##     Alcohol       LabelAppeal          AcidIndex          STARS
##  Min.   :-4.70   Min.   :-2.000000   Min.   : 4.000   Min.   :1.000
##  1st Qu.: 9.00   1st Qu.:-1.000000   1st Qu.: 7.000   1st Qu.:1.000
##  Median :10.40   Median : 0.000000   Median : 8.000   Median :2.000
##  Mean   :10.49   Mean   :-0.009066   Mean   : 7.773   Mean   :2.042
##  3rd Qu.:12.40   3rd Qu.: 1.000000   3rd Qu.: 8.000   3rd Qu.:3.000
##  Max.   :26.50   Max.   : 2.000000   Max.   :17.000   Max.   :4.000
##  NA's   :653                                          NA's   :3359
```

All of the variables are numeric included `STARS` although that independent predictor could potentially be treated as a factor given that it's values correspond to a rating scale. Perhaps that will be explored with transformations in the modeling section to determine if that is a better means of prediction. It also has by far the most `NA` values. From reviewing the mean values of all of the variables it appears `CitricAcid` and `FreeSulfurDioxide` have much larger average values than the other potential predictor variables and it'll be interesting to see if that impact coefficients in the models. `ResidualSugar` appears to have some large outlier values given the interquartile range between -2 and 15.9. It is not immediately obvious what a negative value would represent for this column. Further review may be necessary to determine if these are valid data points or distort the model.

How many observations have negative values (excluding NA values)?

	x
FixedAcidity	1621
VolatileAcidity	2827
CitricAcid	2966
ResidualSugar	3136
Chlorides	3197
FreeSulfurDioxide	3036
TotalSulfurDioxide	2504
Density	0
Sulphates	2361
Alcohol	118
LabelAppeal	3640

	x
AcidIndex	0

Given the proclivity of negative values across multiple columns it would appear to be too widespread an occurrence unless many of the measurements were incorrectly assessed or recorded. It is also not appropriate to change the sign of said observations when it is unclear if that would be a more accurate representation of the data. This may make it a bit harder to explain certain relationships that are encountered in the regression models. The only column that makes sense to have negative values is `LabelAppeal` given that a customer could have a negative rating on a specific type of label.

Are there also negative values in the evaluation data?

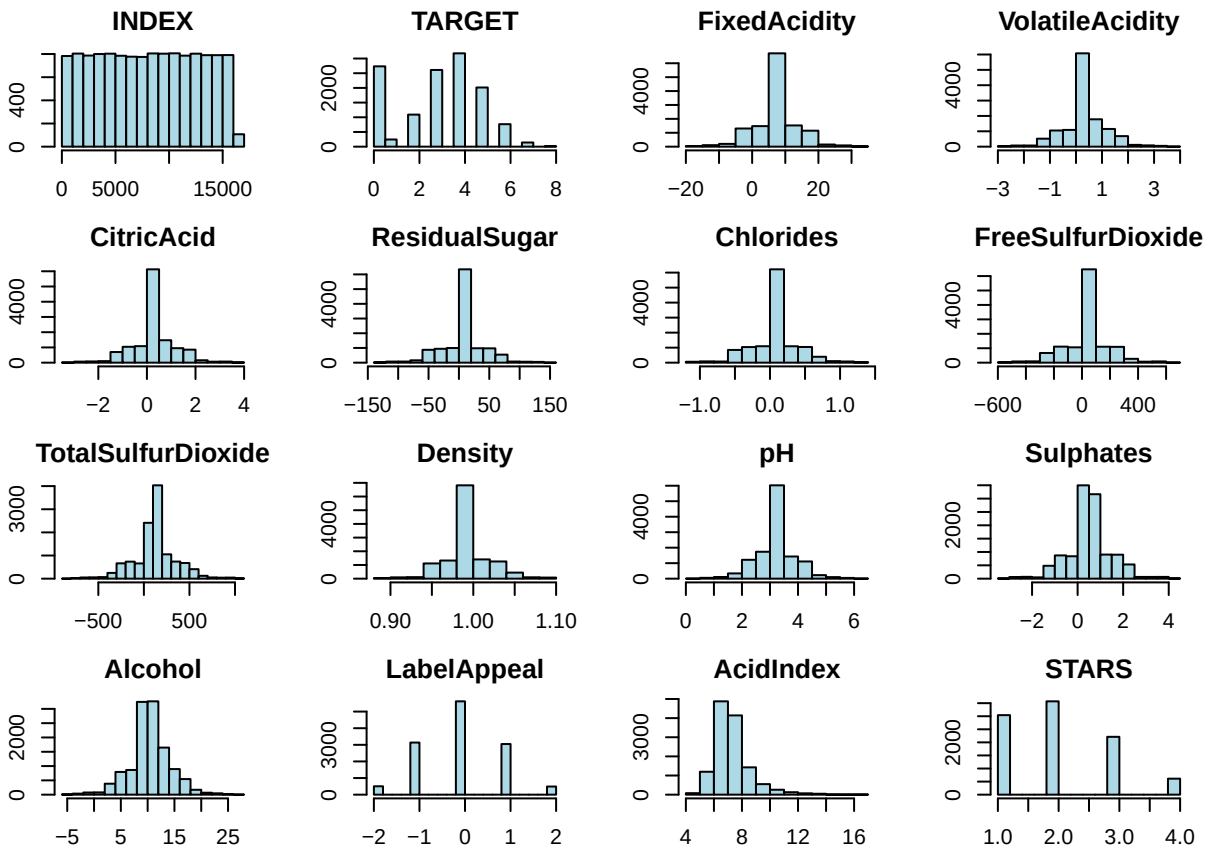
	x
FixedAcidity	439
VolatileAcidity	788
CitricAcid	804
ResidualSugar	828
Chlorides	776
FreeSulfurDioxide	774
TotalSulfurDioxide	639
Density	0
pH	0
Sulphates	594
Alcohol	25
LabelAppeal	924
AcidIndex	0
STARS	0

There are also negative values in all of the same columns as in the training data, which begs the question of how to appropriately handle these occurrences. Excluding that number of observations in training data would distort the model and make it less likely to extrapolate well to the evaluation. Therefore, we are forced to exclude any of these predictors from the model given that negative measurements of specific chemical properties do not make sense and are so common that there are limited alternatives available to us.

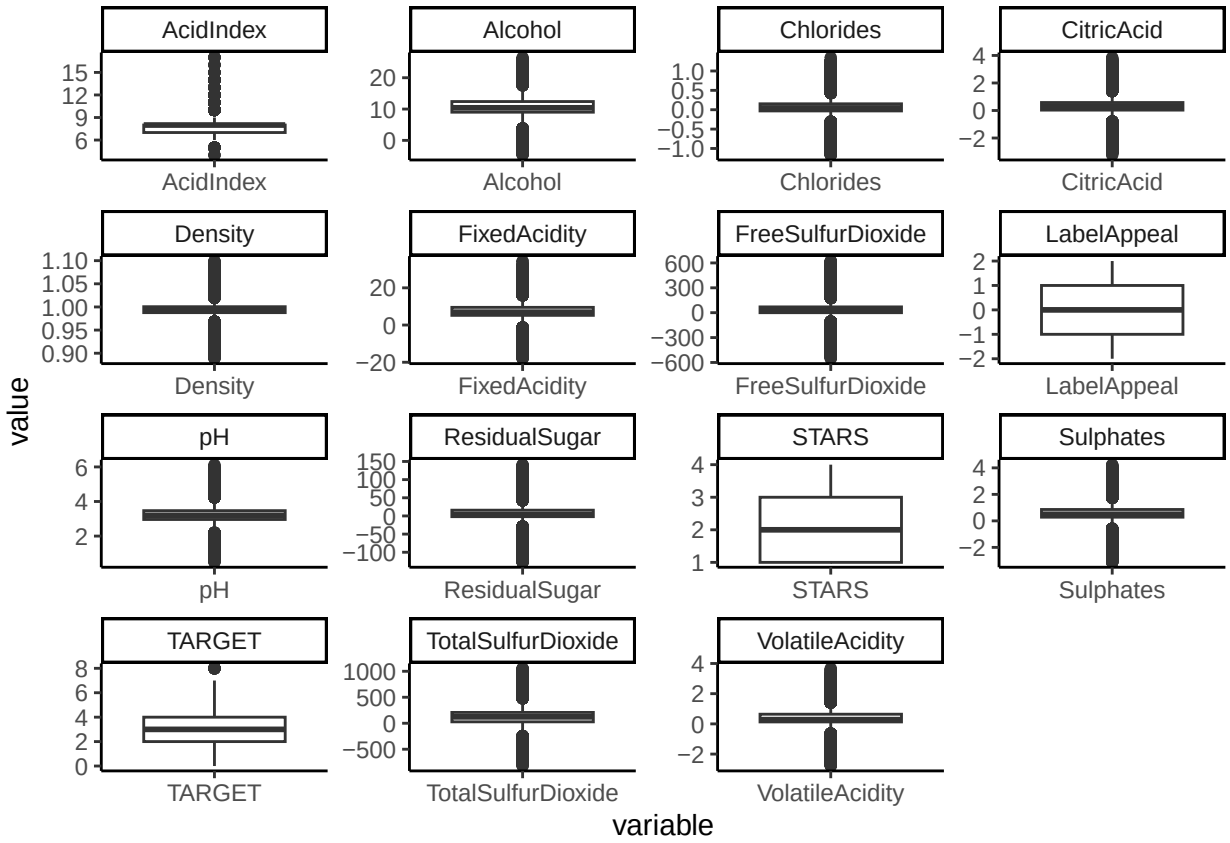
Let's review the frequency of missing values overall across the training dataset:

As mentioned from the summary of the dataframe `STARS` is missing about a quarter of all of the observations which will require some form of transformation or modification to be included in a model. `Sulphates` has more null values as well but at 10% may not be as problematic and will likely require some imputation to represent the missing rows. For the other columns that have missing values it is hard to say that samples were taken for these wines for these attributes or maybe something tainted the reading for them.

Review of all variable distributions:

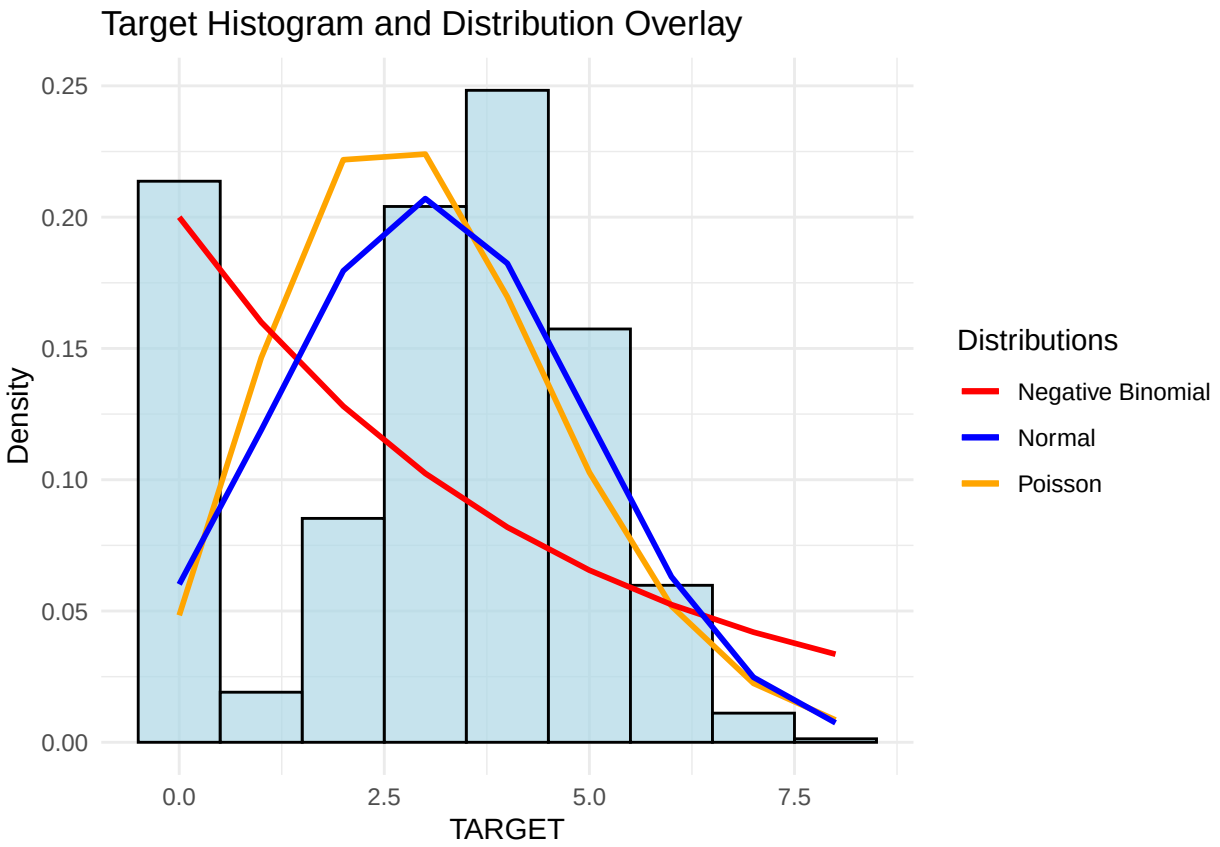


The vast majority of predictor variables appear to have a fairly normal approximation although `ResidualSugar`, `Chlorides`,`FreeSulfurDioxide`,`CitricAcid` have some minor skew in their distributions. `AcidIndex` appears to more closely resemble a poisson or maybe exponential distribution and the skew may require further transformations. As discussed previously, `STARS` given it's more discrete values in nature is skewed, but likely should not be treated as a numeric input. `LabelAppeal` does seem to be discrete as well which makes sense given its perhaps a standardized marketing score based on consumer feedback.



Most of the variables appear to be highly concentrated in the center, in line with the distributions shown in the histograms, which might cause some of the outlier points per a Boxplot to skew the model. It is not immediately obvious if one dimension will impact the model or if perhaps some of these predictors will not be significant within the regression.

The response variable appears to have a large amount of zero values, which requires some further review:



One key assumption when using poisson regression models is that the response is expected to mirror a poisson distribution which has been plotted on the graph above. An attempt will be made to use a standard poisson model despite the fact that there are many zero values from this distribution. It may be necessary to incorporate zero-inflated modeling techniques to occur for this pattern in the data to improve the accuracy/performance of our regression models. The negative binomial function seems to be able to account for the frequency of zero values in the `TARGET` response, but it is not capturing the distribution of the remaining predicted cases very well. The normal distribution was also included to compare against the poisson given that at higher λ values these two distributions tend to converge and in this case are not that substantially different from one another when evaluating the training dataset.

Reviewing λ for the poisson distribution:

```
## The response mean: 3.02907385697538 and variance: 3.71089452283923
```

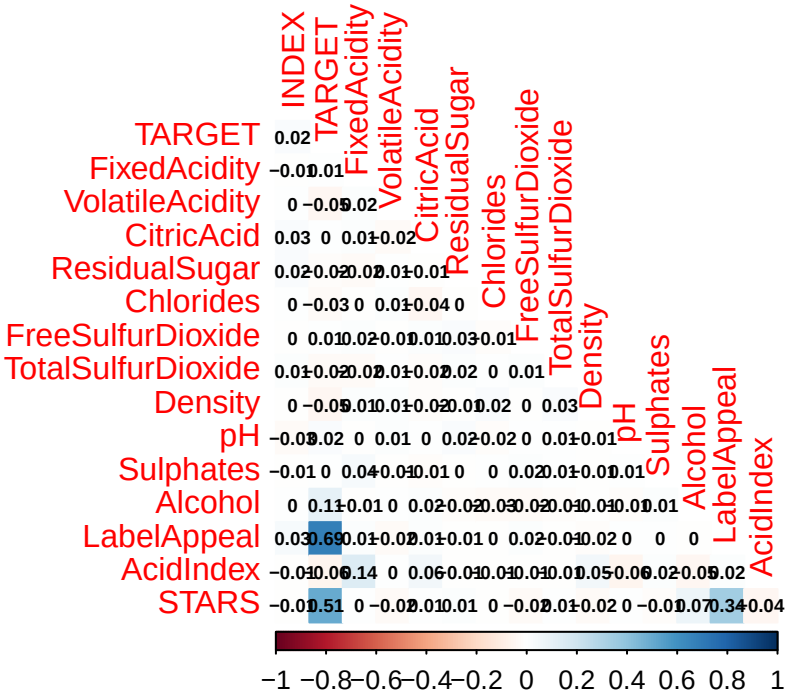
Another important assumption for poisson models relates to the fact that the λ parameter which is a critical input for the poisson distribution expects that the mean and variance must be equal. In practice this is nearly impossible and although there is some overdispersion for the response it is not bad enough to disqualify using this method.

Let's analyze the correlation relationship among predictors and the response:

After excluding the NA values, there are weak correlation values across the board except for `TARGET`,`STARS` and `LabelAppeal`. This is not the most encouraging evidence that the predictor variables provided in the dataset are going to be that successful in modeling the response.

One hypothesis for the apparent lack of linear relationships between variables and the response may be driven off the substantial number of zero values in the response. A potential guess for missing `TARGET` values might be the fact that many more brands of wine exist that are minimally distributed on a commercial scale.

What is the relationship among the variables when excluding the zero response values?



There aren't many changes across the correlations after excluding the zero response rows. `LabelAppeal` appears to have a stronger linear relationship with the `TARGET` and `STARS`. Two different acidity (`AcidIndex` and `FixedAcidity`) inputs also have more correlation than before although it is still a fairly weak relationship.

Data Preparation

As discussed in the exploratory section, the `STARS` variable appears to be something that can be treated as a factor. The number of NA values would likely indicate that it is not a rated wine by the experts and much less likely to have sample cases purchased by distributors. The missing values will be updated to zero to mirror the scale of this discrete predictor.

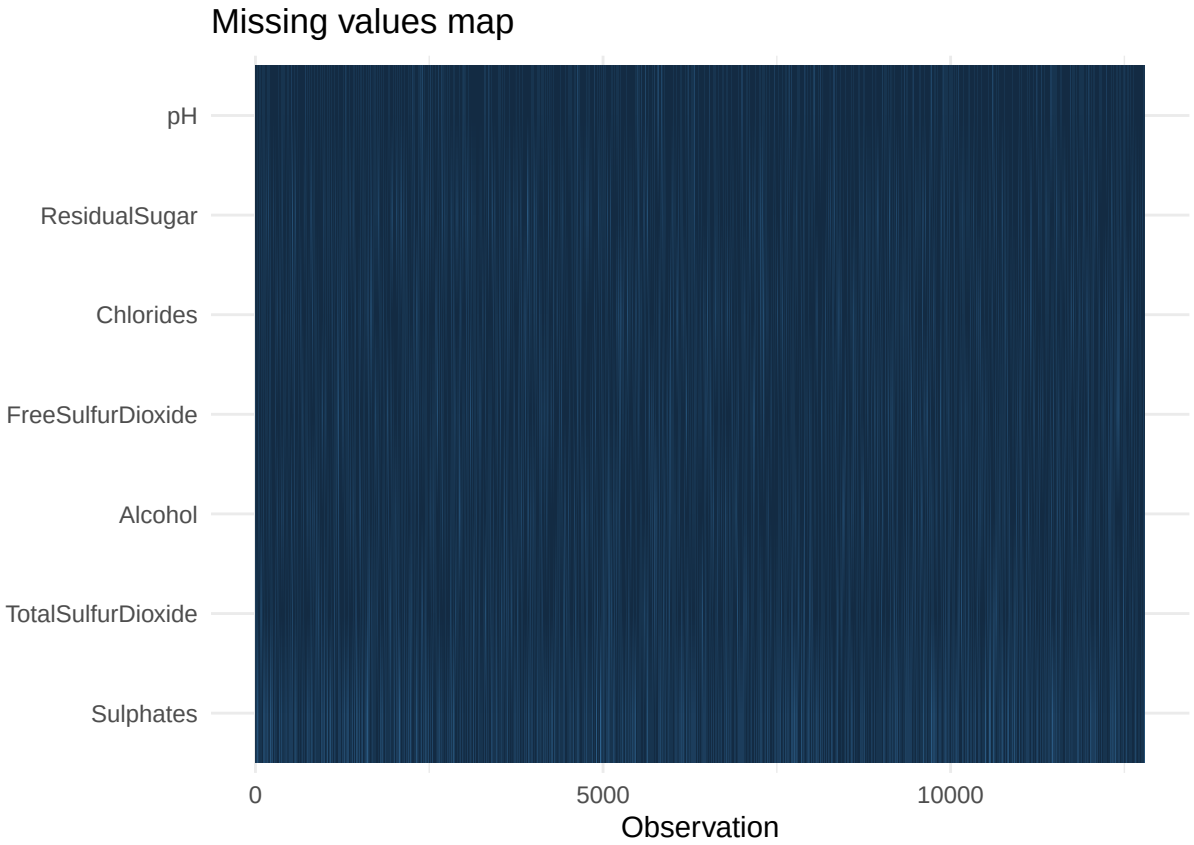
The remaining columns with missing values are far less clear to evaluate and will need a more standardized imputation method:

One more comprehensive method to evaluate if the data is Missing Completely at random (MCAR) is running Little's test although it is unlikely to generate a statistically significant difference for most real world datasets.

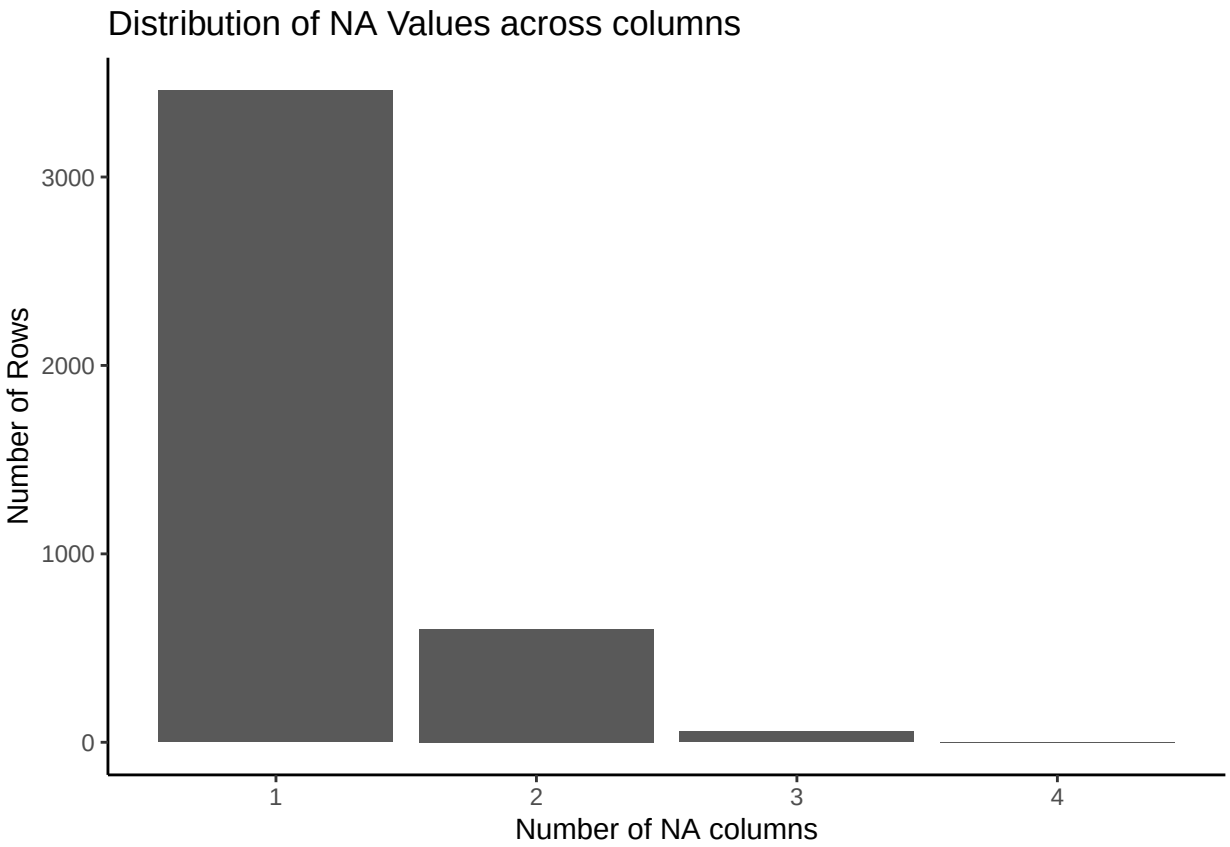
statistic	df	p.value	missing.patterns
5566.644	1235	0	94

The zero in the p-value indicates that the missing values across these columns are not in fact MCAR.

Let's review if the NA values appear in similar observations:



The plot above highlights via a heat map all of the individuals rows where the independent predictors have NA values (shaded in light blue). It's a bit challenging to discern where observations have more than one null value, but there is definitely some overlap across records.




```
## Only 660 rows have more than 1 column with NA, which is only 16.02% of the observations that have NAs
```

It does not appear that many observations have more than 1 NA value and the heatmap was a little bit less clear in helping us identify overlapping columns.

Let's split the data into train and test sets to provide some observations to evaluate the selected model as a holdout set:

We are going to compare two methods of imputation to see how the results impact the regression models. Given the central concentration of many of the predictors it may be enough to apply the averages to the missing values to estimate them for inclusion in the modeling. The first of which will be replacing the missing values with averages except for `STARS` which we will add a zero value:

Let's confirm that all the missing values were populated after applying the averages:

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

As anticipated this imputation method brings in all of the observations for the models that we select in the next section.

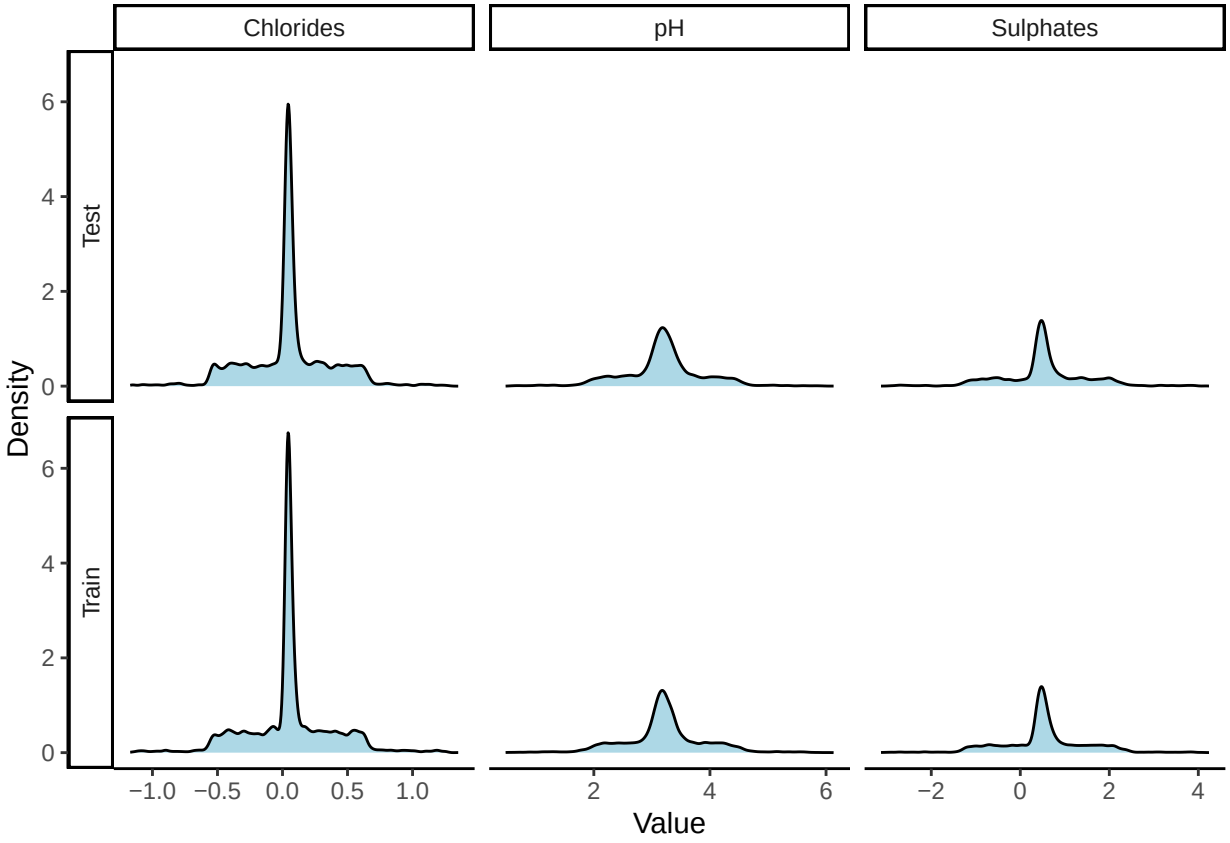
Alternatively, we will use the MICE package to fill the NA values:

Let's confirm that the imputation filled all the null values for MICE:

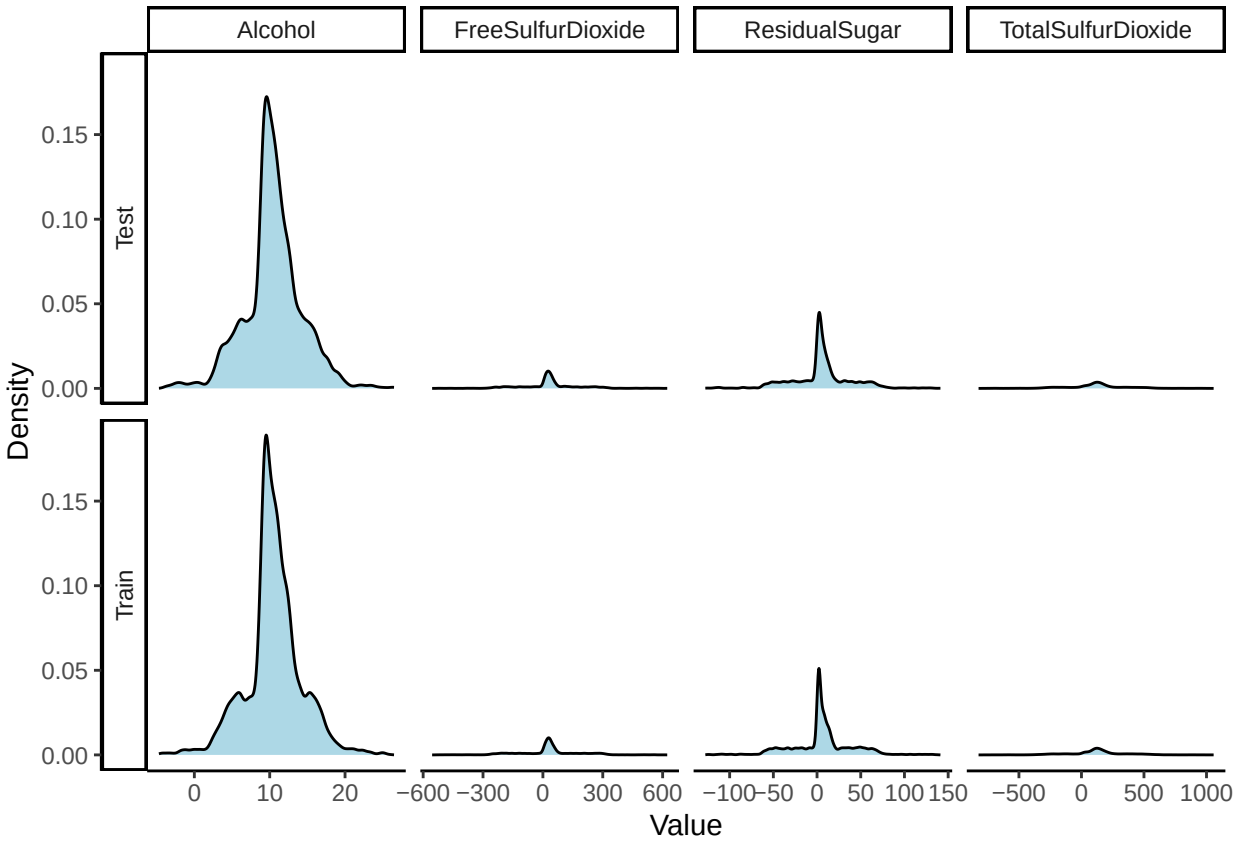
```
## [1] TRUE
```

As expected the MICE imputed estimate values for all the missing fields and we can now include all observations within the model.

Let's review the distributions of the train and test imputed datasets to confirm they remain similar to the initial unmodified dataset:

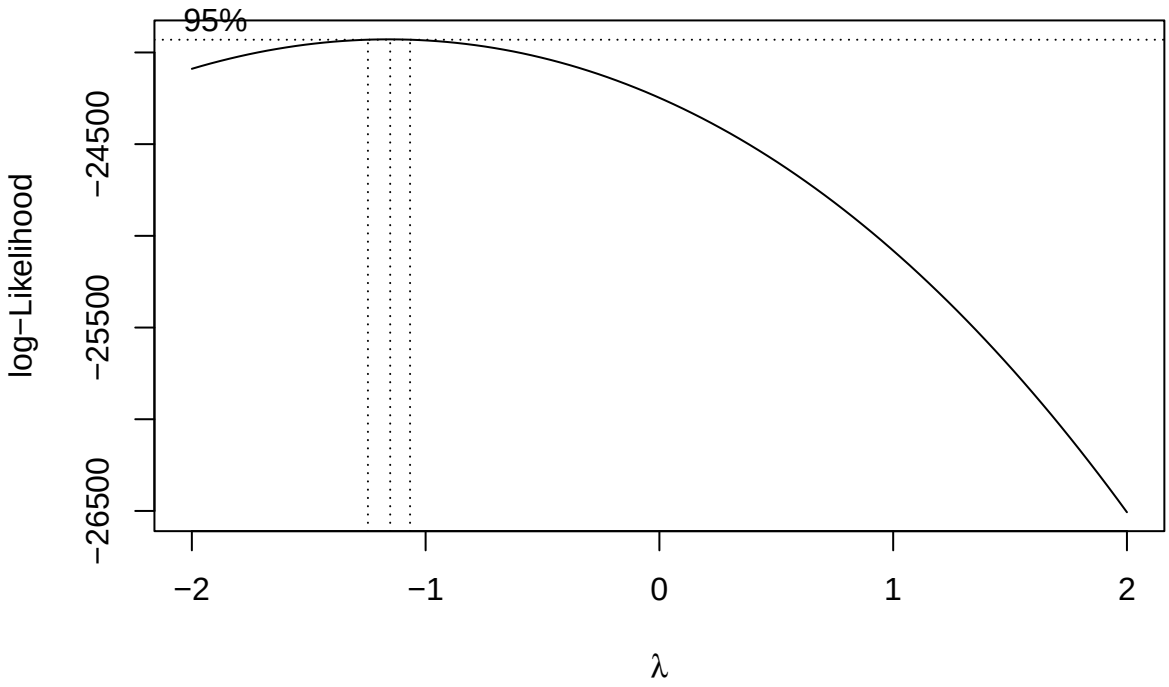


These columns were separated as they had higher density points distorting the remainder of predictors that had NA values in the density plots. Across train and test sets it still appears to be similarly distributed and not substantially different from the initial input data.



The remaining columns that had imputations also mirror one another in the train and test sets as well as the initial distributions. We can proceed with additional review of other variables that need transformations.

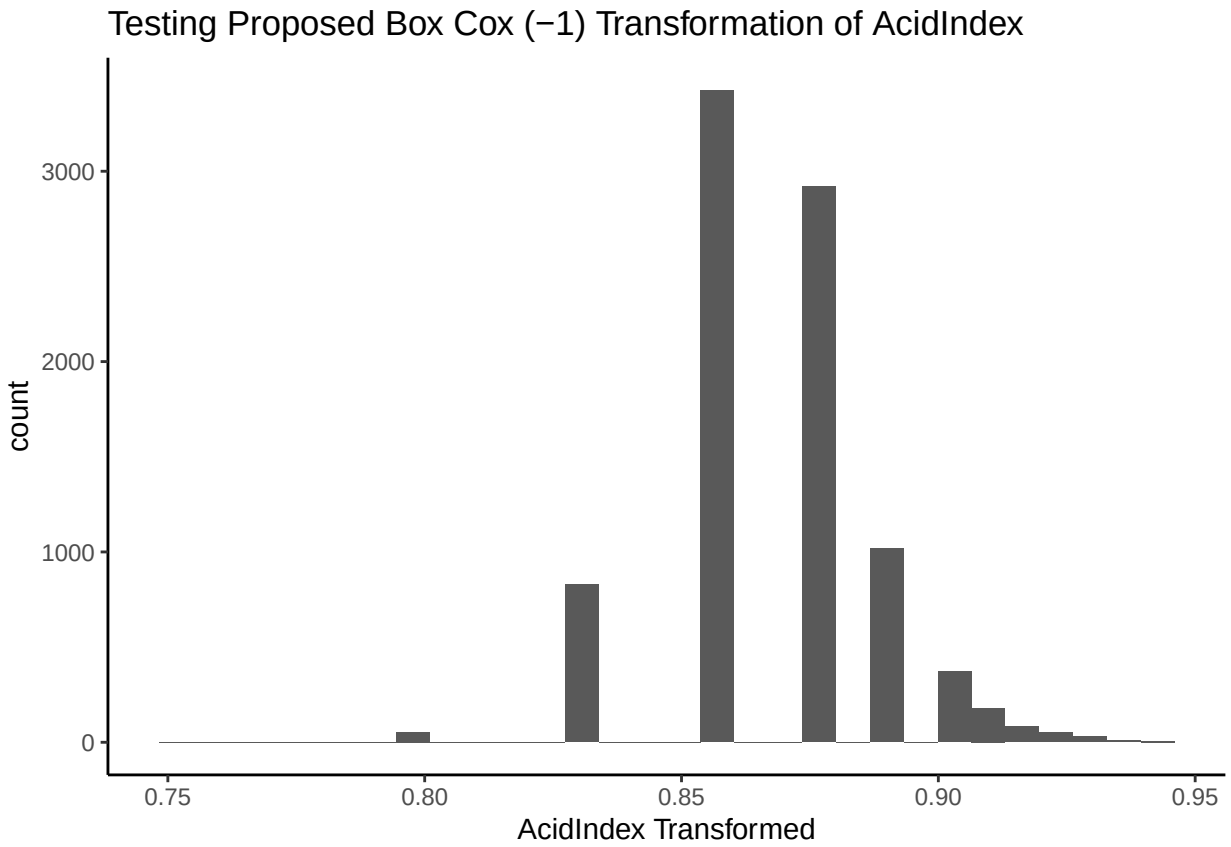
For the most part the predictors themselves seemed approximately normal, but there is one additional variable, `AcidIndex` that remains skewed that may require further transformation.



The maximum likelihood estimation for this variable was -1.15 although we are bit skeptical that this transformation is necessary or will approximately normalize the distribution of this column.

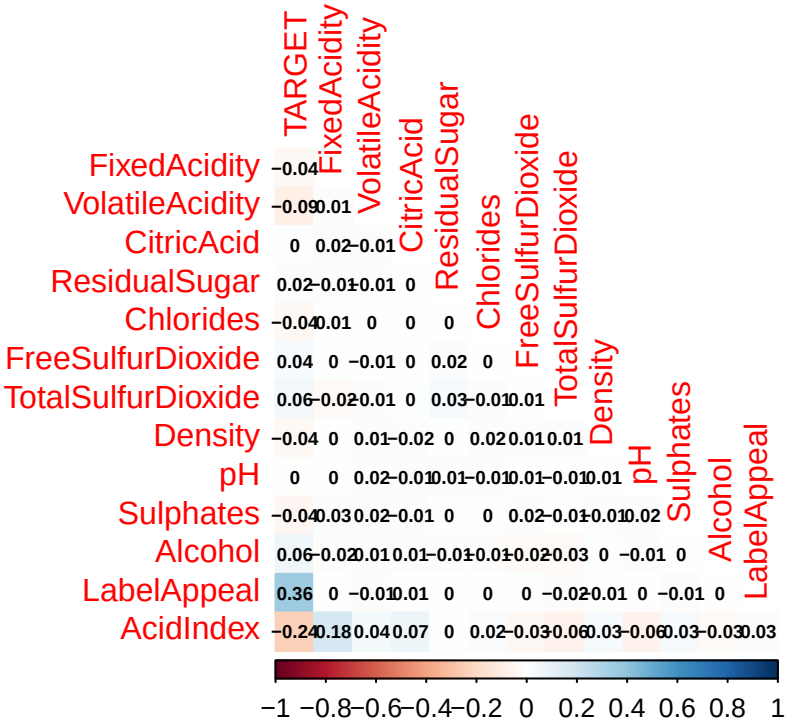
Let's see what the distribution of `AcidIndex` looks like with the reciprocal transformation $\frac{1}{x}$:

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



As expected it does not seem warranted to transform this variable after reviewing the histogram of the transformed predictor given that the data is still fairly sparse although it does have a more bell shape than before it doesn't seem to improve the distribution enough to warrant implementation.

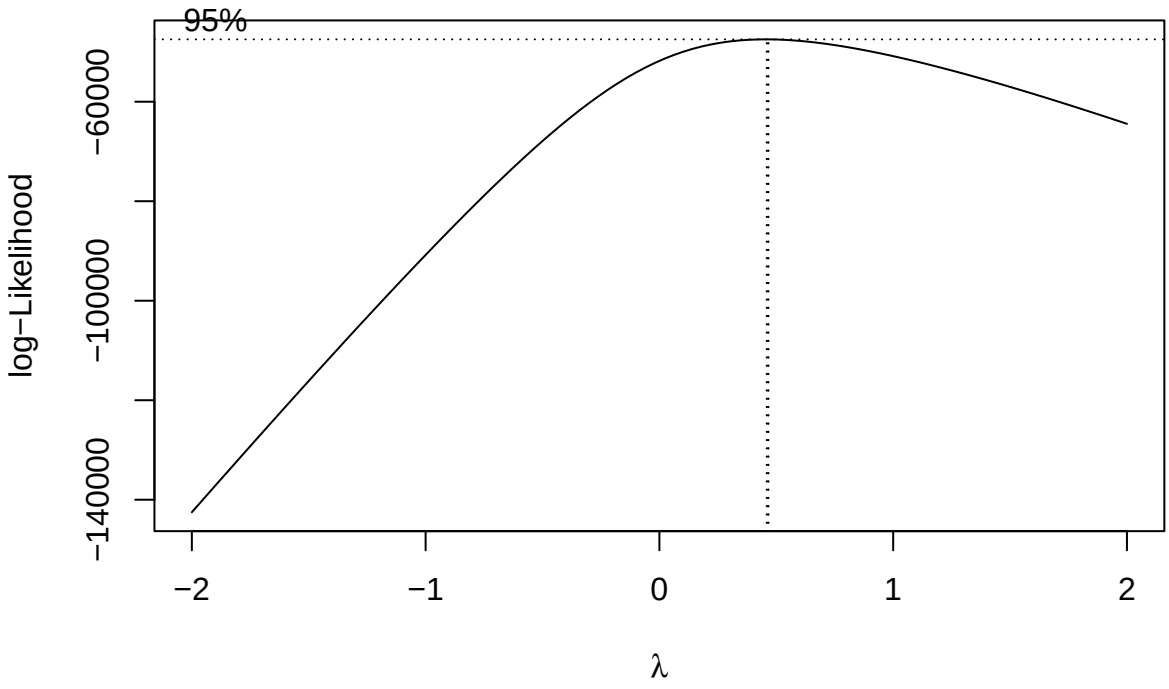
Before proceeding further let's double check the correlations of the target and predictor variables to see how imputation impacted the linear relationships across the columns:



By modifying the `STARS` into a factor we have excluded it from the correlation plot but would expect it remains one of the stronger correlated predictors. The `LabelAppeal` value seems to have decreased quite a bit although it's relationship is far stronger than most of the other potential predictors. Perhaps it's worth considering modifying this variable into a factor as well to see if that impacts the performance. Depending on model performance this might be something worth exploring. `AcidIndex` has the strongest negative relationship among the predictors based on the imputed observations and has a somewhat strengthened relationship with `FixedAcidity`. Semantically the two names of these variables appear as though they should have some relationship, but it is not that strong of a correlation.

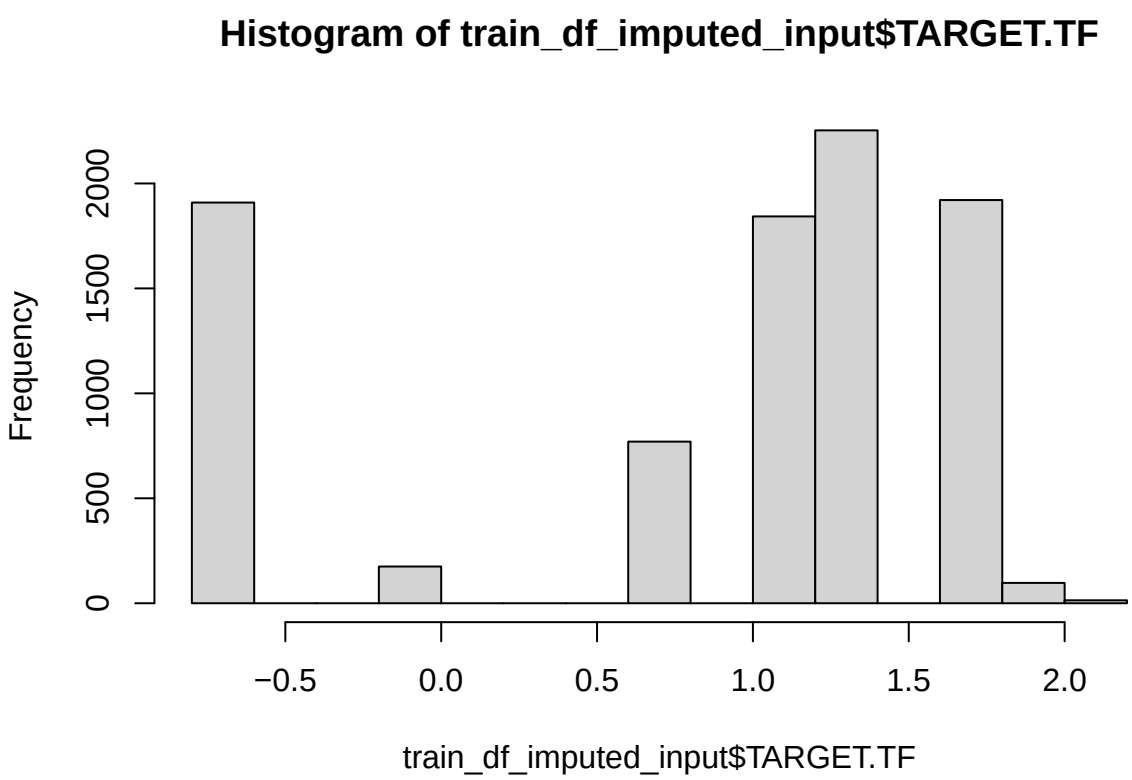
One means of prediction using OLS for count data is taking the $ln y$ as a means of replicating the Poisson linear transformation by applying it on the response. This transformation is based on a UC Davis (publication) [https://cameron.econ.ucdavis.edu/racd/simplepoisson.pdf] on count regression. Therefore we will prepare an alternative response variable although some minor modifications are needed since there are many zero values.

Let's review what the Box-Cox method proposes for the transformation:




```
## [1] 0.4646465
```

The Box-Cox preferred transformation appears to show the square root as the optimal transformation, but given the primer on count data from UC-Davis we will apply the *lny* instead.



Modeling

Model POIS:1 - Full Poisson Count Model with All Missing Values Imputed using Averages - Reduced via Stepwise AIC Model Selection As indicated in the exploratory section of this analysis the negative variables are fairly commonplace and present a unique challenge given the limitations around exclusion. Therefore, we will begin the modeling utilizing the remaining predictors available that did not have illogical negative values.

We create Model 1: Poisson with all missing values imputed and we perform stepwise model selection to select the model with the smallest AIC value using the `stepAIC()` function from the `MASS` package.

A summary of Model 1: Poisson below:

```
##
## Call:
## glm(formula = TARGET ~ STARS + LabelAppeal + AcidIndex, family = "poisson",
##     data = avg_impute_train_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2741  -0.6551   0.0066   0.4567   3.8006
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.820471   0.046595   17.61   <2e-16 ***
## STARS1       0.763381   0.023376   32.66   <2e-16 ***
```

```
## STARS2       1.086723   0.021737   49.99   <2e-16 ***
## STARS3       1.209465   0.022865   52.90   <2e-16 ***
## STARS4       1.332710   0.028836   46.22   <2e-16 ***
## LabelAppeal  0.159226   0.007326   21.74   <2e-16 ***
## AcidIndex   -0.078097   0.005341  -14.62   <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: 15951.4  on 8981  degrees of freedom
## Residual deviance:  9649.9  on 8975  degrees of freedom
## AIC: 32101
##
## Number of Fisher Scoring iterations: 6
```

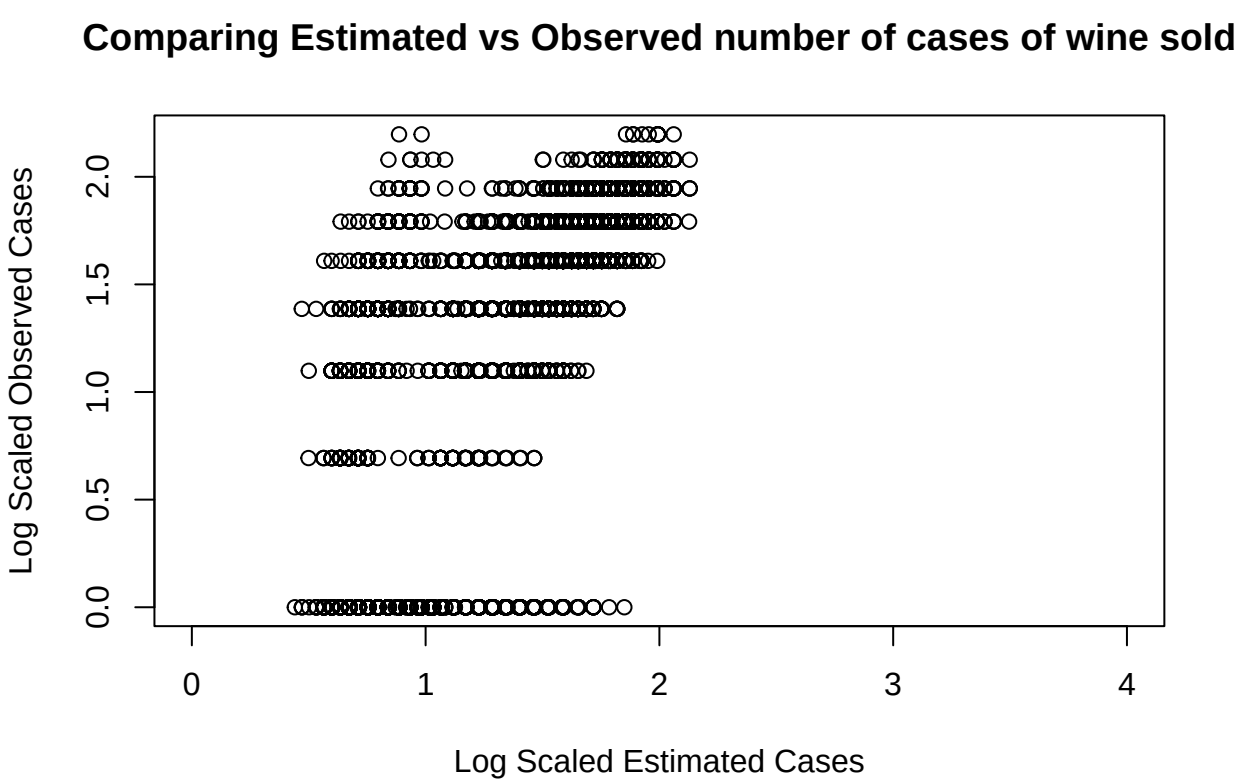
The AIC of Model 1 is 32101 which appears to be very high and may indicate other problems that have been discussed during the explanatory section. We will assess the predictors more formally after building a zero inflated equivalent as a comparison. When conducting backward stepwise regression it appears that `pH` and `Density` were not significant in the poisson regression model.

Residual Deviance: Chi-Squared Test:

```
##      res.deviance   df            p
## [1,]     9649.931 8975 4.292594e-07
```

This test measures the goodness of fit of the poisson model and an ideal case is if when the test returns a non-significant p-value. It is designed to compare the expected counts (response) returned from the model to the observed values and if the model did as expected there would not be a statistically significant difference between the two values. Therefore, this would indicate that the data does not fit the model well and given that all the available predictor variables were initially included that we violated the linear distribution of the Poisson and there was more dispersion than allowed for the Poisson.

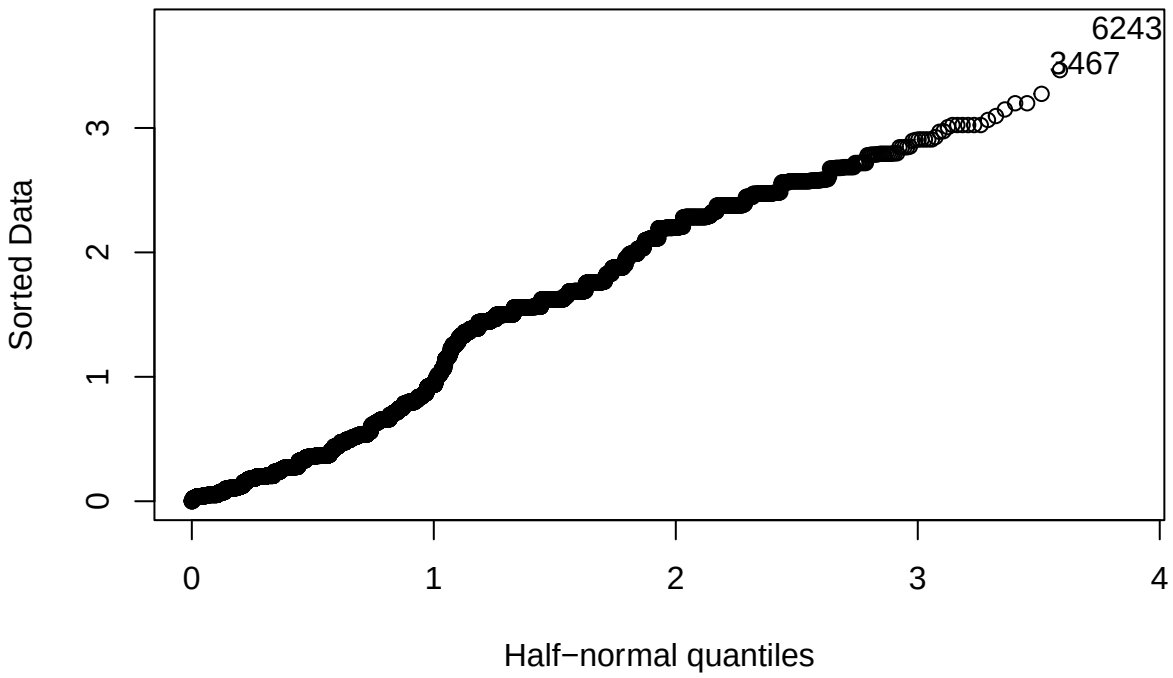
Let's plot the log scaled version of the response and the predicted number of cases:



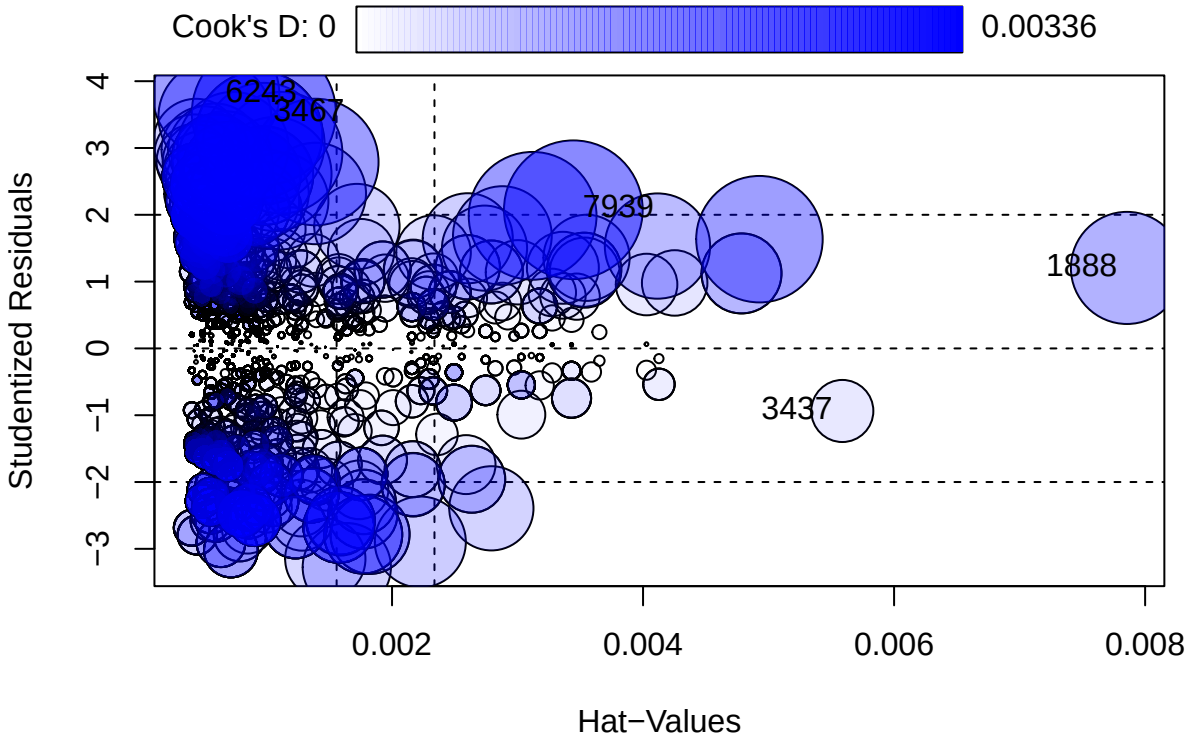
Consistent with the Chi-Squared Test it appears there are too many observed values of zero to fit into the Poisson distribution. This plot is designed to identify where the estimated number of cases on the x axis diverges from the actual values particularly emphasizing the concentration of zero values that a poisson distribution is unable to accurately estimate.

The half-norm plot of the residuals and a plot of Cook's D statistic are two different ways of identifying outliers:

The half norm plot assesses the distribution using quantiles similar to the qqplot

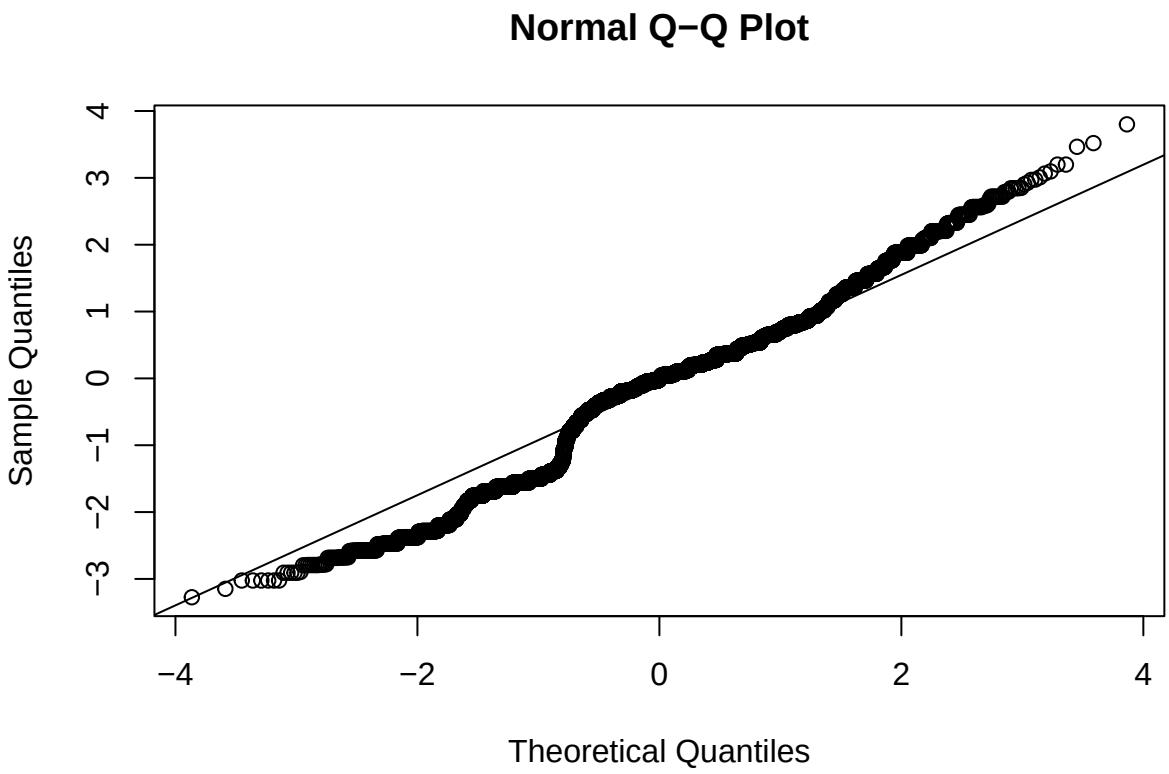


The half-norm plot highlights two points at the top right of the graph, but does not appear to show substantial divergence to warrant classifying observations 6243 and 3467 as outliers.




```
##          StudRes          Hat        CookD
## 1888  1.2048239 0.0078552749 0.0020331844
## 3437 -0.9364402 0.0055875335 0.0006235831
## 3467  3.5231534 0.0009790835 0.0033557191
## 6243  3.8029717 0.0005978377 0.0025912833
## 7939  2.0752195 0.0034445782 0.0031103476
```

The influence plot appears to show that point 1888 is a leverage point; however, it doesn't cross the -2 boundary that would classify it as an outlier for the model. There are other points listed below the graph as well that have much larger standardized residual values but don't have divergent case predictions and therefore do not appear to be outliers as well.



The QQplot would seem to indicate that the deviance residuals are diverging from the expected distribution.

We check for over/underdispersion of the model to see if there are any additional violations of assumptions.

Let's calculate the dispersion from the model:

```
## [1] 0.8875544
```

It appears there is a decent amount of under-dispersion in this model as the ideal case would return a value of 1, which may limit our abilities to reliably use Poisson Regression for this count data response consistent with other diagnostic methods conducted. Generally, the concern to evaluate Poisson regression is assessing overdispersion violations, but in this case the zero values are driving this issue. It does appear that the frequency of zero values does not follow a standard Poisson distribution

Model POIS:2 - Zero Inflated Poisson Count Model (ZIP) with All Missing Values Imputed with MICE

```
##
## Call:
## pscl::zeroinfl(formula = TARGET ~ STARS + LabelAppeal + AcidIndex | STARS +
##     LabelAppeal, data = train_df_imputed_input, dist = "poisson")
##
## Pearson residuals:
##       Min        1Q    Median        3Q       Max
## -2.314869 -0.475163  0.007133  0.399826  4.172198
##
```

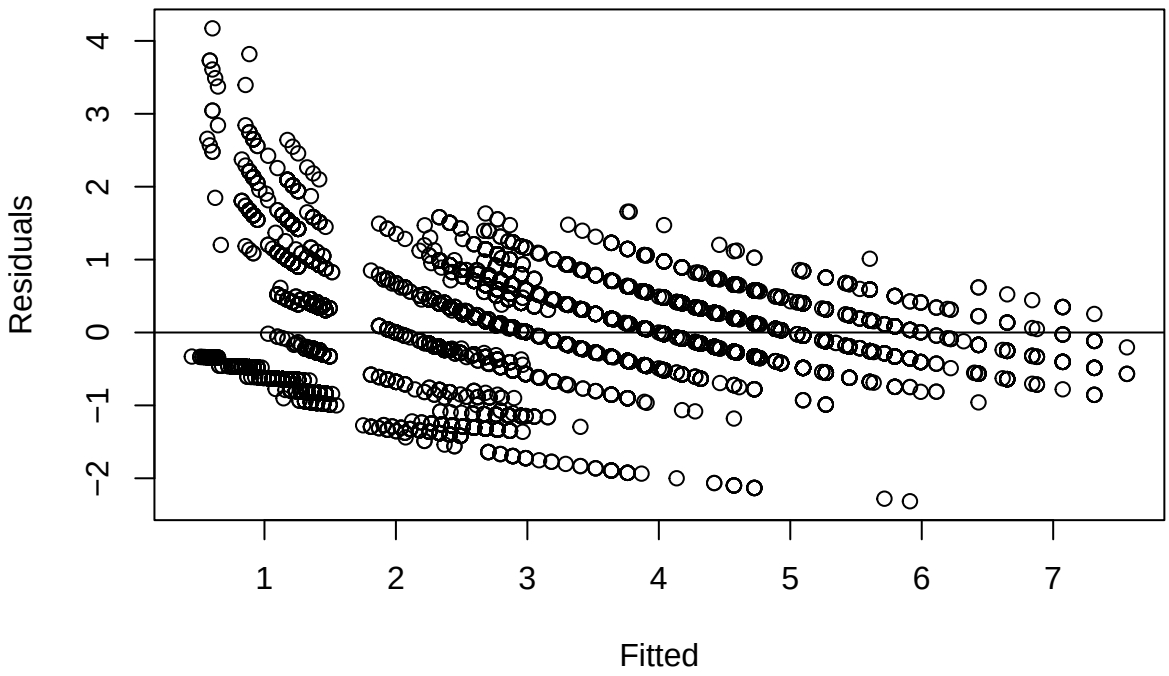
```
## Count model coefficients (poisson with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.371153   0.048434  28.309  < 2e-16 ***
## STARS1       0.065831   0.025434   2.588  0.00965 **
## STARS2       0.191492   0.023626   8.105 5.27e-16 ***
## STARS3       0.292150   0.024705  11.825  < 2e-16 ***
## STARS4       0.387123   0.030486  12.698  < 2e-16 ***
## LabelAppeal  0.232548   0.007558  30.770  < 2e-16 ***
## AcidIndex   -0.033392   0.005808  -5.750 8.94e-09 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.44952    0.04841   9.285   <2e-16 ***
## STARS1       -2.05286    0.08759 -23.436   <2e-16 ***
## STARS2       -5.98793    0.46372 -12.913   <2e-16 ***
## STARS3      -20.30062  418.30173  -0.049    0.961
## STARS4      -20.49269  757.77762  -0.027    0.978
## LabelAppeal   0.74423    0.04864  15.302   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 19
## Log-likelihood: -1.45e+04 on 13 Df
```

This predictors included in this model were based on the items that were significant to the model rather than including a full one with all of the variables. There is some discussion in statistical publications around using robust methods to estimate the standard error when using zero inflation techniques, but we will review other diagnostic methods to evaluate this model although it does not typically impact coefficients. The logit model used the strongest predictors (`STARS` and `LabelAppeal`) to identify the instances when no cases were expected to be sold based on the expectation that negative customer evaluation and missing expert reviews would prompt distributors to not purchases cases of these wines. Somewhat logically the higher `STARS` ratings did not closely align with zero cases and these values are not significant in the logistic piece of the model. When reviewing the coefficients in the binomial count model it makes sense that the negative impact of additional `STARS` increases with each additional improvement in the rating.

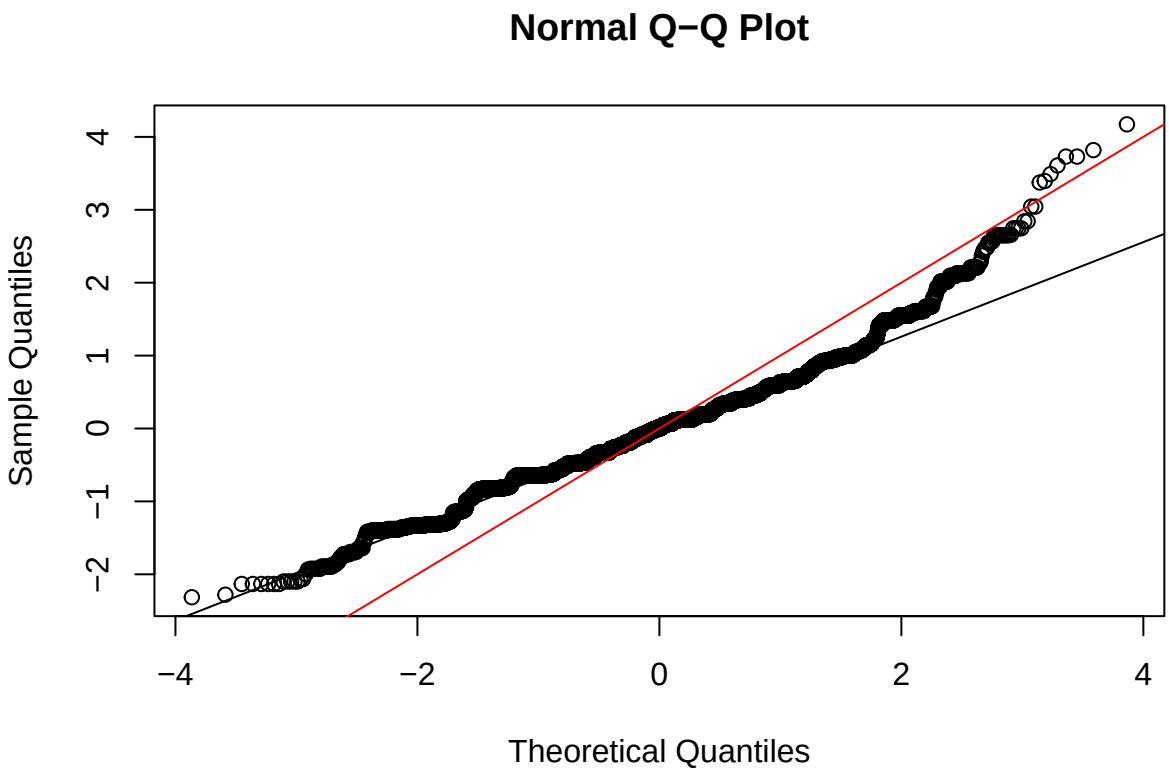
Let's run a Chi-Squared Test to evaluate the goodness of fit:

```
## 'log Lik.' 0 (df=13)
```

A statistically significant result in the Chi-Squared Test would seem to indicate that there is a goodness of fit for the ZIP model. The degrees of freedom is based on the number of predictors in the model excluding the intercept (the null model)



When assessing the fitted vs residuals plot there is some pattern to the residuals and the largest ones still exist for the zero cases examples it would appear. They are still centered around zero.



The residuals primarily follow the QQline for most of the quantiles, but appear to diverge on the higher end and follow a line tracking from zero to one.

```
## [1] 0.3266363
```

There is more substantial underdispersion captured in this model than the other poisson regression one and we would like to avoid any under/over dispersion that exists.

Let's compare the Akaike Information Criterion for both poisson based models:

```
##           df      AIC
## mod1_pois  7 32101.01
## mod2_zip  13 29019.26
```

While the AIC values are not vastly different it is apparent that the zero inflated Poisson regression model is preferable to the first poisson regression model. This likely has to do with the strong concentration of zero values that exist in the response variable that is accounted for as a separate logistic model in the zero inflated method.

Let's review both of the models to compare the similarities and differences from the summaries and model coefficients despite some obvious limitations identified with the the first model using standard poisson regression. In terms of coefficients we will begin with the most influential which alternate between `STARS` and `LabelAppeal` although both have appear to have the same p-value between both models. (note that some resources online suggest applying a robust method of convergence to determine the standard error which would influence the significance value). It makes sense that these two predictors are in some ways the biggest drivers for distributors because on paper it would seem the reviews of customers implies sales potential for

a given wine type. Irrespective of the chemical makeup of the wines the customers will drive companies to meet that demand. `AcidIndex` was the only other significant predictors in the zero inflated poisson (ZIP) model. The predictor has the same direction for it's coefficient and it appears the `AcidIndex` will decrease the number of sold wine cases.

Model NB:1 - Negative Binomial Count Model with All Imputed Variables based on Averages

```
## 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 ~ STARS + LabelAppeal + AcidIndex, data = avg_impute_train_df,
##     init.theta = 40671.55888, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2740  -0.6551   0.0066   0.4566   3.8004
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.820490   0.046597   17.61   <2e-16 ***
## STARS1       0.763381   0.023377   32.66   <2e-16 ***
## STARS2       1.086722   0.021737   49.99   <2e-16 ***
## STARS3       1.209464   0.022866   52.89   <2e-16 ***
## STARS4       1.332711   0.028837   46.22   <2e-16 ***
## LabelAppeal  0.159225   0.007326   21.73   <2e-16 ***
## AcidIndex   -0.078099   0.005341  -14.62   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(40671.56) family taken to be 1)
##
##     Null deviance: 15950.6  on 8981  degrees of freedom
## Residual deviance:  9649.6  on 8975  degrees of freedom
## AIC: 32103
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  40672
##           Std. Err.:  41019
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -32087.3
```

The negative binomial distribution can sometimes better approximate dispersion when the mean and variance are not equal to one another which would violate the Poisson distribution and related assumptions for Poisson Count regression. Therefore, we will use the same variables as selected in the backward selection to try to

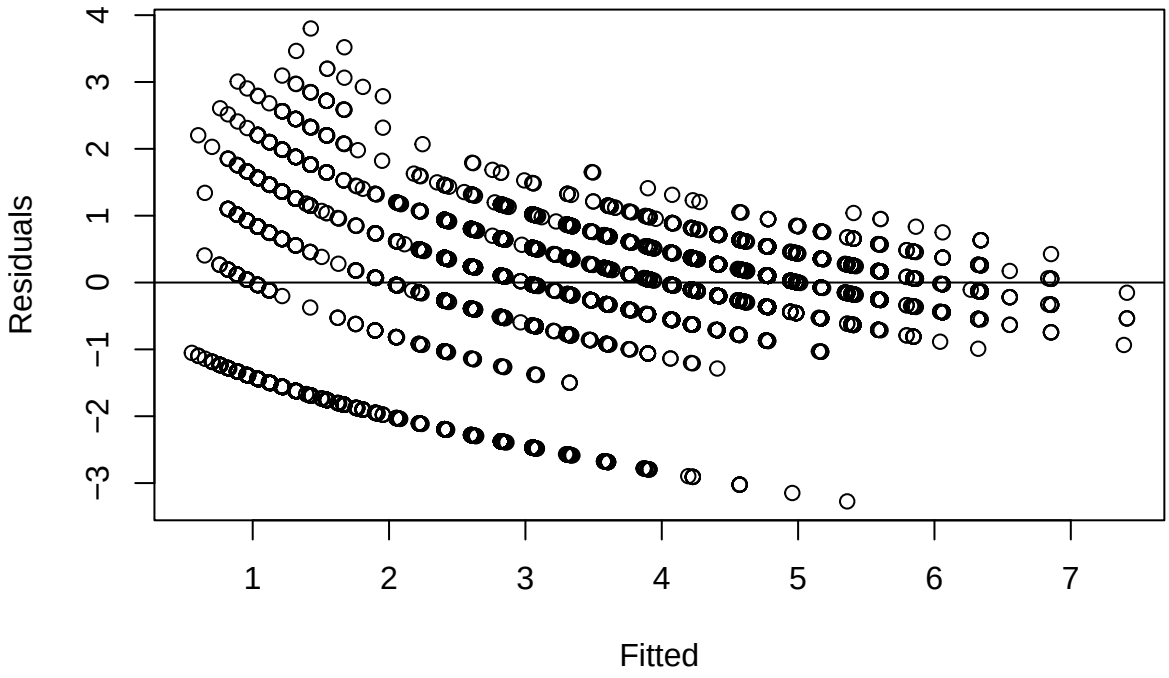
create a similar comparison. It appears that the algorithm is unable to converge with a negative binomial distribution.

In terms of the strength of the coefficients, the `STARS`,`LabelAppeal`, and `AcidIndex` are the largest predictors in both models and all have the same sign. It makes sense that these are likely the most important independent variables.

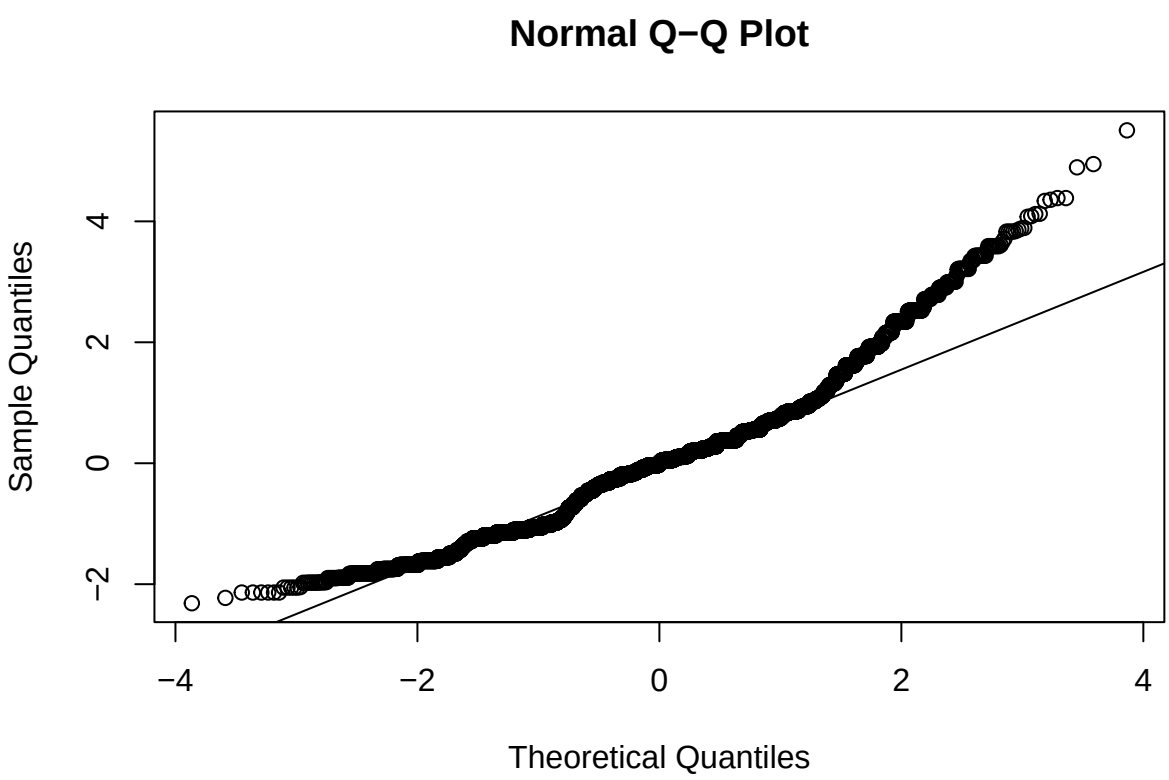
One way to assess the negative binomial model is to run a likelihood ratio test to compare Model 3 using the negative binomial family against Model 1 using the Poisson family.

```
## 'log Lik.' 1 (df=8)
```

Based on this test it would appear that the negative binomial model is also not appropriate as the result is above the generally accepted alpha level at 5%.



When reviewing the fitted versus residual plot there is some pattern in the residuals which appears to track closely with the `TARGET` shape where even the negative binomial has the largest residuals for many zero cases. The spread overall doesn't make it particularly if the variance is changing although there is clear separation in the plot due to the discrete nature of the response variable.



When using the pearson residuals we can see that for the interquartile range the negative binomial model tracks well with the expected distribution, but ultimately diverges on both tails somewhat similarly to the other ZIP model. On the higher end there is substantial skew which would indicate that the residuals are not normally distributed.

Let's calculate the dispersion for this model:

```
## [1] 0.6229317
```

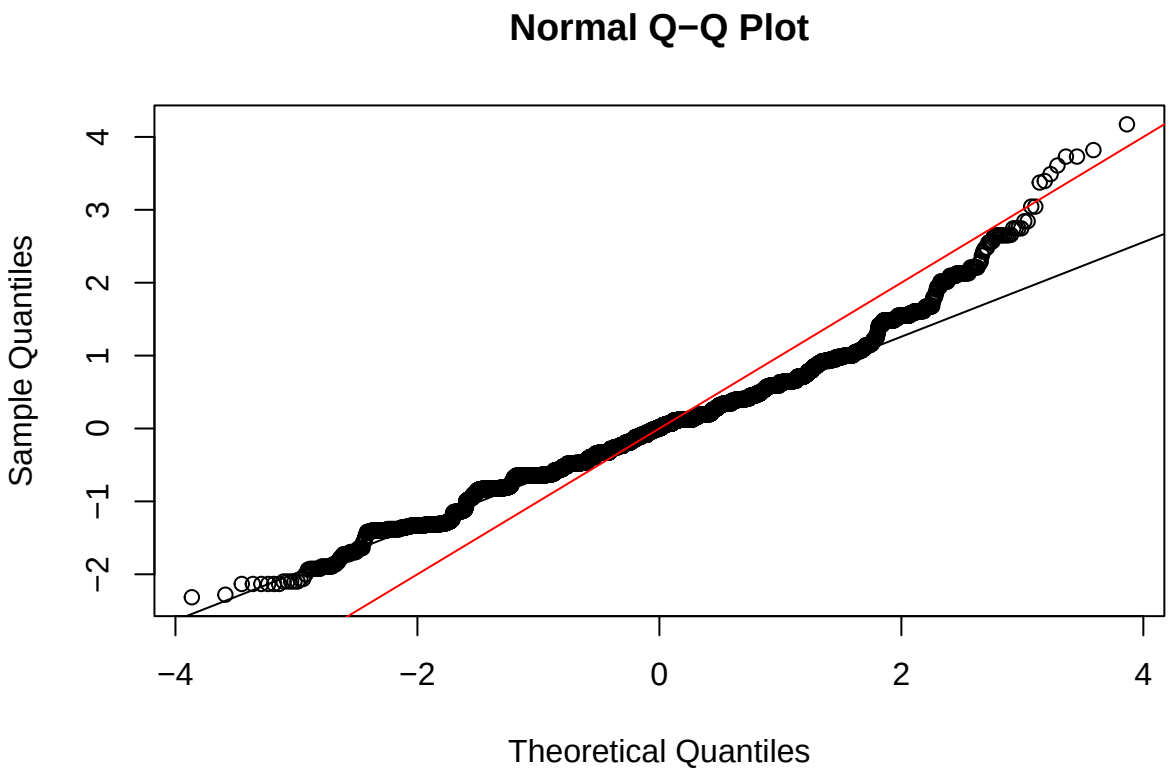
It appears there is even more underdispersion than identified in the poisson model although it is not as low as the zero inflated model.

Model NB:2 - Zero Inflated Negative Binomial (ZINB) using Imputed values with MICE Let's try to evaluate the data based on a zero inflated negative binomial:

```
##
## Call:
## pscl::zeroinfl(formula = TARGET ~ STARS + LabelAppeal + AcidIndex | STARS +
##     LabelAppeal, data = train_df_imputed_input, dist = "negbin")
##
## Pearson residuals:
##      Min       1Q   Median       3Q      Max
## -2.31491 -0.47515  0.00713  0.39983  4.17247
##
## Count model coefficients (negbin with log link):
```

```
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.371161   0.048434  28.310  < 2e-16 ***
## STARS1       0.065829   0.025434   2.588  0.00965 **
## STARS2       0.191500   0.023627   8.105 5.26e-16 ***
## STARS3       0.292159   0.024706  11.826  < 2e-16 ***
## STARS4       0.387132   0.030486  12.699  < 2e-16 ***
## LabelAppeal  0.232549   0.007558  30.770  < 2e-16 ***
## AcidIndex   -0.033395   0.005808  -5.750 8.92e-09 ***
## Log(theta)  17.374824   1.322322  13.140  < 2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.44951    0.04841   9.285   <2e-16 ***
## STARS1       -2.05302    0.08760 -23.436   <2e-16 ***
## STARS2       -5.98839    0.46391 -12.909   <2e-16 ***
## STARS3      -20.30435  419.07491  -0.048    0.961
## STARS4      -20.49383  758.18493  -0.027    0.978
## LabelAppeal   0.74430    0.04864  15.303   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 35139056.8825
## Number of iterations in BFGS optimization: 26
## Log-likelihood: -1.45e+04 on 14 Df
```

We tried to be consistent with the zero inflated model for the negative binomial and used the same independent variables as what was identified as significant in zero inflated poisson model. From a review of the summary, the residuals appear to be distributed around zero although there is a slight positive skew as compared to the negative. All of the predictors are also significant which is consistent with the Poisson ZIP Model. The 1-Star ratings appear to be a bit less significant to the model and in the logistic component the higher `STARS` are also insignificant although this makes some sense given higher rated wines would likely have cases ordered.



The QQplot aligns more closely with the distribution line; however, similar to the other distributions there are a lot of points on the right tail that are skewed.

Let's calculate the dispersion from the model:

```
## [1] 0.3266696
```

The dispersion statistic is in line with the other zero inflated model, which appears to be a knock on both models.

	df	AIC
mod1_pois	7	32101.01
mod2_zip	13	29019.26
mod3_nb	8	32103.30
mod4_zinb	14	29021.26

When comparing the performance of the models using the AIC statistic, both zero inflated models have a very similar criterion score and the negative binomial model more closely aligns with the poisson model.

Model MLR 5: Multiple Linear Regression on All Imputed Variables based on Averages

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

```
##     data = avg_impute_train_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8111 -0.8804  0.0701  0.8487  6.2530
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.77087    0.52886   7.130 1.08e-12 ***
## STARS1       1.36064    0.03962  34.341  < 2e-16 ***
## STARS2       2.38699    0.03828  62.354  < 2e-16 ***
## STARS3       2.96445    0.04428  66.950  < 2e-16 ***
## STARS4       3.70767    0.07055  52.552  < 2e-16 ***
## LabelAppeal  0.46437    0.01637  28.364  < 2e-16 ***
## Density     -0.89085    0.52579  -1.694   0.0902 .
## AcidIndex   -0.19838    0.01069 -18.562  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.314 on 8974 degrees of freedom
## Multiple R-squared:  0.5316, Adjusted R-squared:  0.5312
## F-statistic:  1455 on 7 and 8974 DF,  p-value: < 2.2e-16
```

When running backward selection on the variables in multiple linear regression we see that `Density` variable remain despite being classified as insignificant. The signs for each coefficient match Model 3 (NB) and the most significant predictors remain `STARS`, `LabelAppeal`, and `AcidIndex`. The F-statistic indicates that the model is better than a pure intercept approach when running MLR.

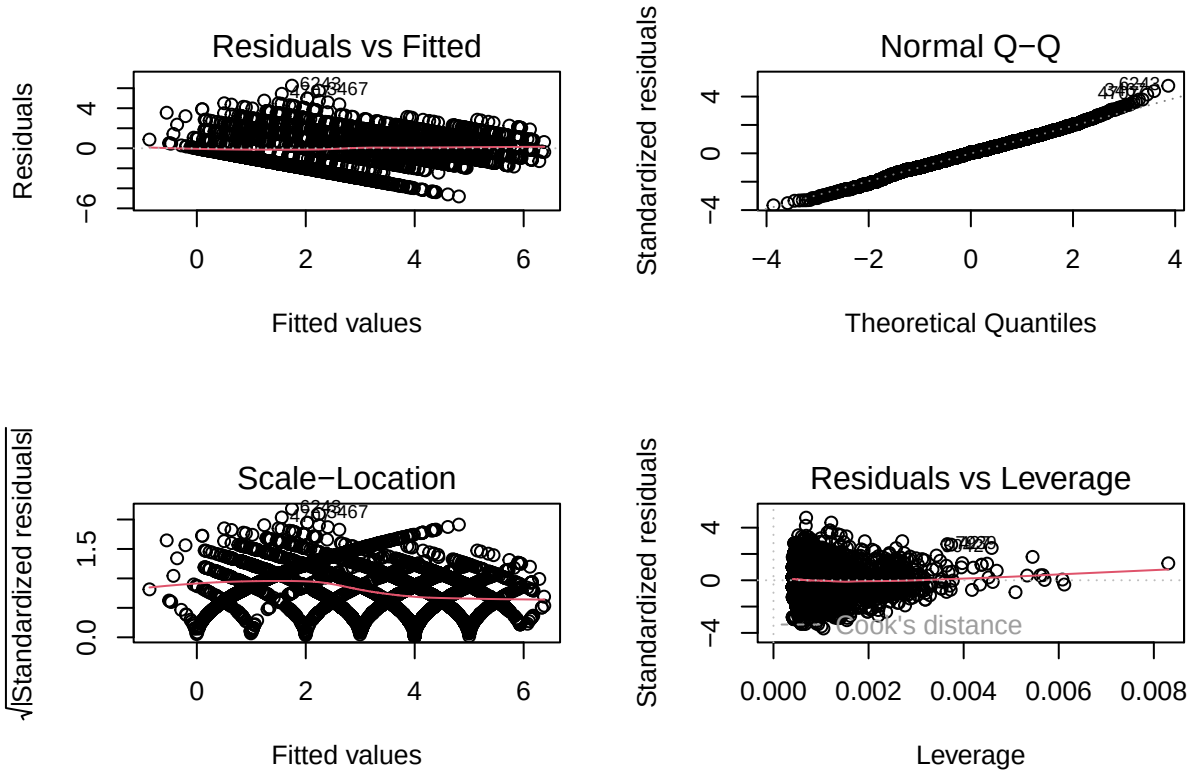
The residuals appears to be somewhat evenly spread around zero although there is a partial right skew. The R^2_{adj} value shows about 53.12% of the variability can be explained by the variation in the predictors.

Are there any multicollinearity concerns in this model?

	x
STARS1	1.475430
STARS2	1.537176
STARS3	1.469217
STARS4	1.198153
LabelAppeal	1.106512
Density	1.001713
AcidIndex	1.036559

No issues between multicollinearity which is to be expected given the low correlation values across many of the predictors in the correlation plot from the explanatory section.

Let's review the diagnostic plots to assess if any assumptions are violated:



The Fitted vs Residuals appears to show a slight pattern with the majority of the residuals above zero although this kind of makes sense given the heavy concentration of zeros and the fact that predicted number of cases will never be negative. The QQ plot seems to follow the assumption that the residuals follow a normal distribution and the right skew does not diverge that substantially from the expected case. The Standardized residuals show a clear pattern at each discrete value which is generally one reason why OLS is not encouraged for count data as it does not appear to as if a linear model adequately captures the relationship between the predictors and the response variables. Not a ton of high leverage points that exist although 7699 appears to be close to the Cook's distance statistic.

Model MLR 6: Multiple Linear Regression on All Imputed Variables based on MICE and ln y response

```
##
## Call:
## lm(formula = TARGET.TF ~ STARS + LabelAppeal + pH + AcidIndex,
##     data = train_df_imputed_input)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.18961 -0.31889  0.05444  0.34242  1.97842
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.861489   0.055183  15.612   <2e-16 ***
## STARS1       0.743279   0.018915  39.296   <2e-16 ***
## STARS2       1.210938   0.018278  66.251   <2e-16 ***
```

```
## STARS3       1.390170   0.021140  65.760   <2e-16 ***
## STARS4       1.549924   0.033676  46.024   <2e-16 ***
## LabelAppeal  0.096461   0.007816  12.341   <2e-16 ***
## pH          -0.017942   0.009705  -1.849   0.0645 .
## AcidIndex   -0.099280   0.005110 -19.429   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6274 on 8974 degrees of freedom
## Multiple R-squared:  0.4849, Adjusted R-squared:  0.4845
## F-statistic:  1207 on 7 and 8974 DF,  p-value: < 2.2e-16
```

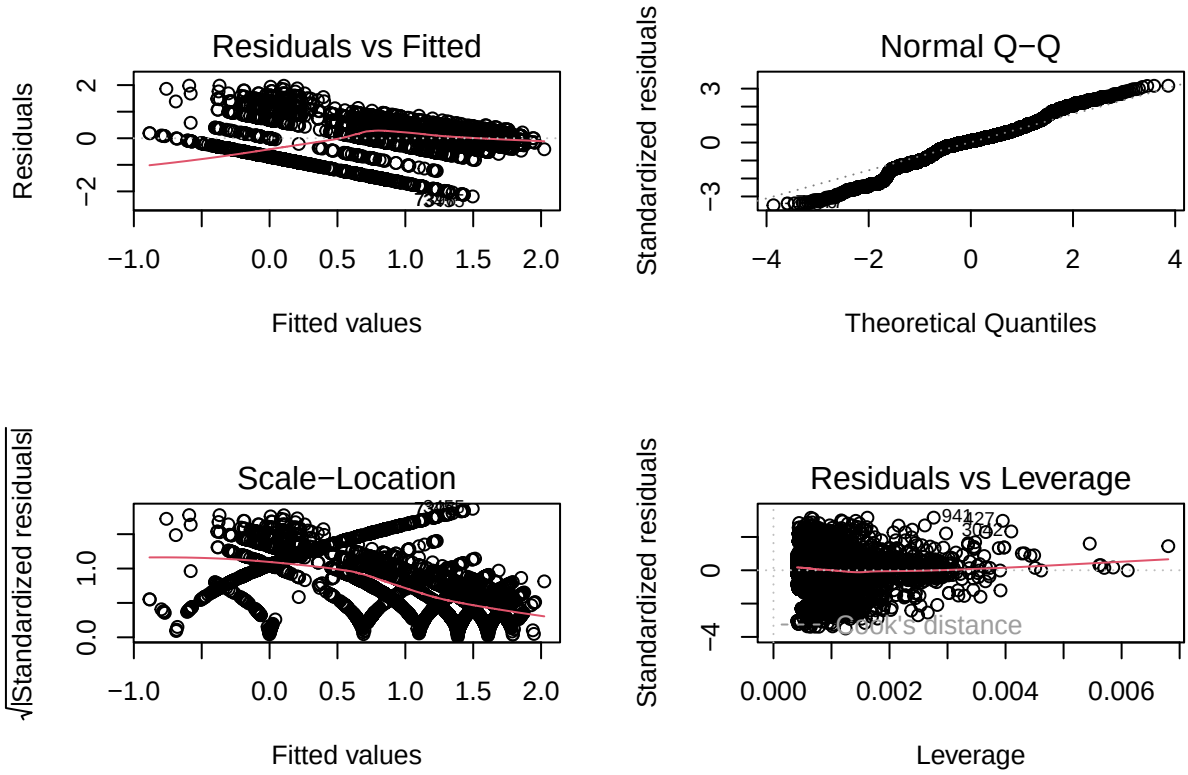
Both this variation of the model and model 5 which both use OLS have very similar coefficients for all of the significant variables after running backward selection. They appear to emphasize the importance of `STARS`, `LabelAppeal`, and `AcidIndex` much like the other models. `pH` instead of `Density` is kept in this model although neither is significant. The R^2_{adj} is slightly lesser than Model 5 at 48.45% and the residual standard error is about half the equivalent in Model 5. The interpretability of the model is certainly reduced when considering the lny because it is a percentage change in cases of wine given a one unit change in a given predictor.

Is there any multicollinearity concerns in this version of the model?

	x
STARS1	1.475415
STARS2	1.537557
STARS3	1.469373
STARS4	1.197765
LabelAppeal	1.106512
pH	1.004365
AcidIndex	1.039764

No issues between multicollinearity which is to be expected given the low correlation values across many of the predictors in the correlation plot from the explanatory section.

Let's review the diagnostic plots to assess if any assumptions are violated:



The fitted residuals are not evenly distributed around zero although the variance does not appear to change across the range of the predicted values. The QQ plot for the most part conforms with the expected normal distribution of errors except for a left skew which is likely driven off of all the transformed zero values that were in the `TARGET` variable. The scaled version of the fitted versus residuals shows a clear pattern that would violate the assumption that there is a linear relationship between the response and the predictors. It's not clear that it is acceptable to use this model for any inference going forward due to the violations of key assumptions.

```
##           df      AIC
## mod5_mlr   9 30408.24
## mod6_mlr   9 17125.77
```

Despite the fact that each MLR model has a lower AIC value it would be unwise to use these models further given the violation of assumptions as previously mentioned when reviewing the diagnostic plots in prior sections.

Select Models Based on the fact that multiple models that were tested in the prior section had somewhat serious flaws, we are going to evaluate the zero inflated models against the hold out set in order to make a final selection of the "best" model. It still appears as though there are some challenges in accurately identifying all of the zero cases scenarios.

Let's review the AIC for each model again:

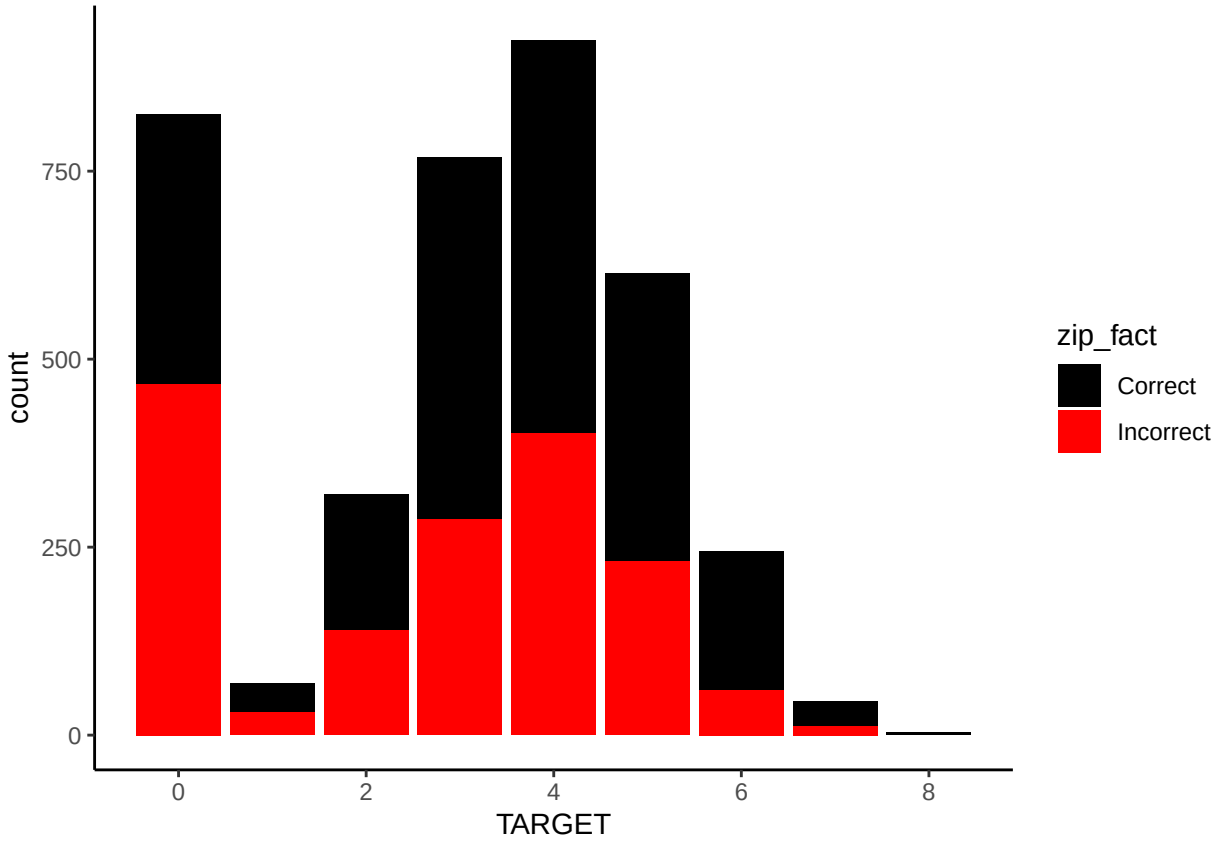
```
##           df      AIC
## mod2_zip  13 29019.26
## mod4_zinb 14 29021.26
```

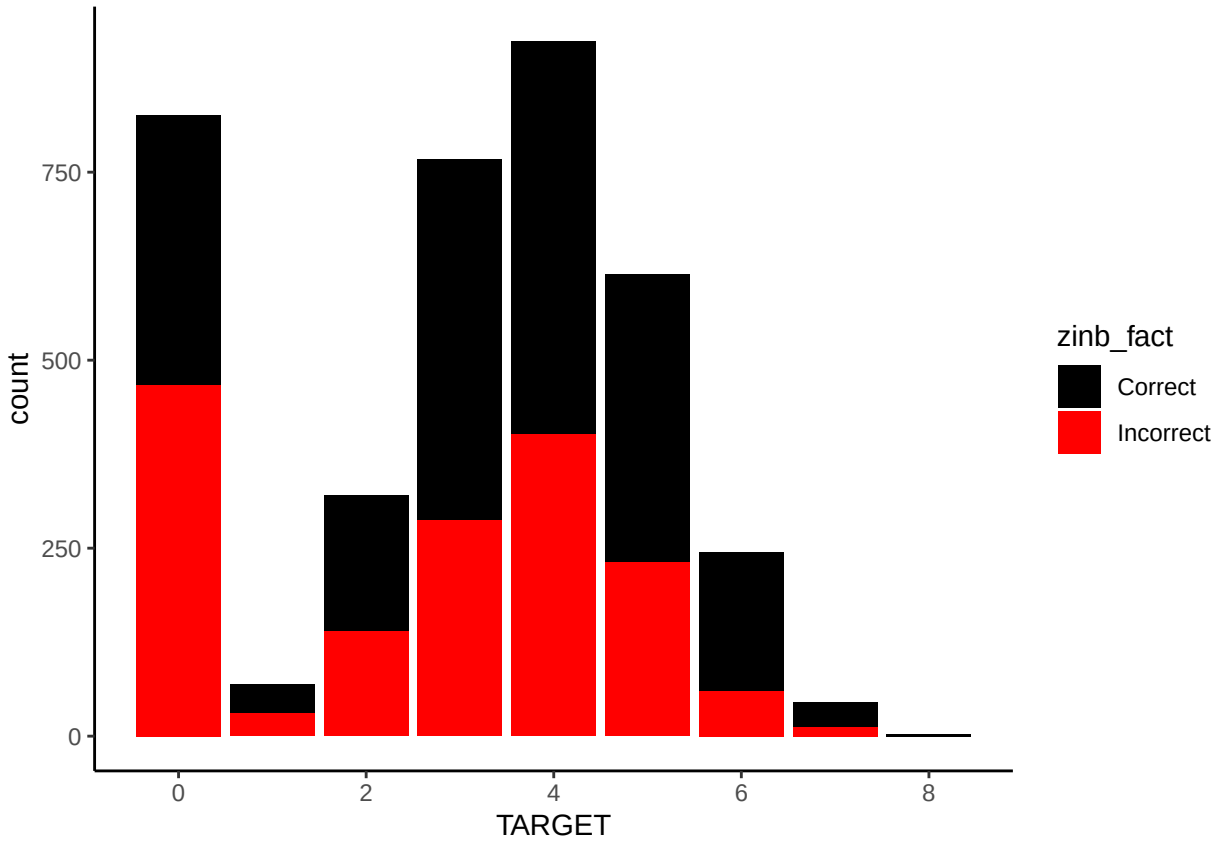
The zero inflated poisson model slightly edges the zero inflated negative binomial model. From a closer review these models also appear to be highly similar to one with nearly identical estimates for given predictors.

The process for determining the correct prediction is to first determine if the logistic model classified zero values. After rounding the predicted values to the closest discrete value for both variations of the model it appears that the the ZIP and ZINB models are returning identical predictions for the holdout set. While the expected values are slightly different to the decimal place that has not impacted the end discrete prediction.

Each model's accuracy is not that strong at $1627/3813 \sim 42.67\%$

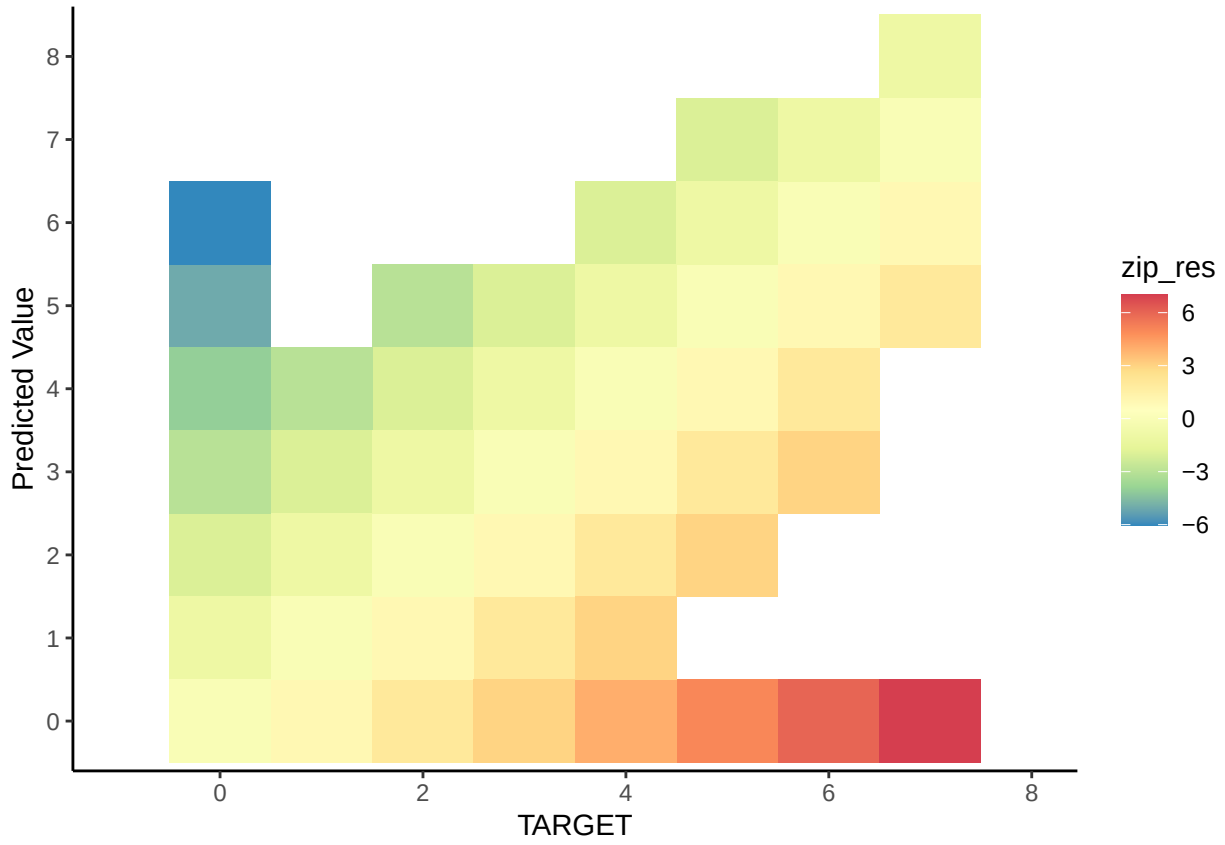
zip_match	cnt	zinb_match	cnt
0	2186	0	2186
1	1627	1	1627



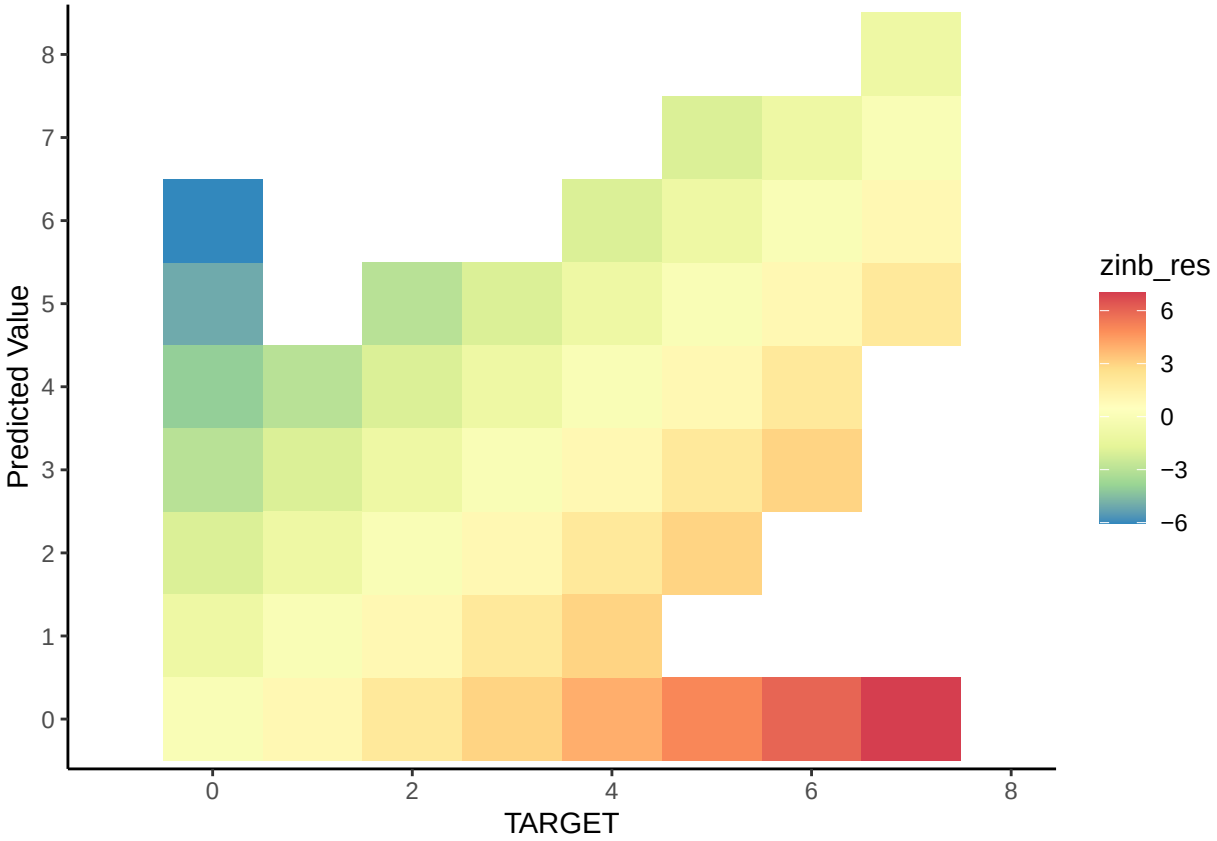


Let's review a comparison of the Target to the predicted value.

```
## Warning: Removed 3 rows containing missing values (`geom_tile()`).
```




```
## Warning: Removed 3 rows containing missing values (`geom_tile()`).
```



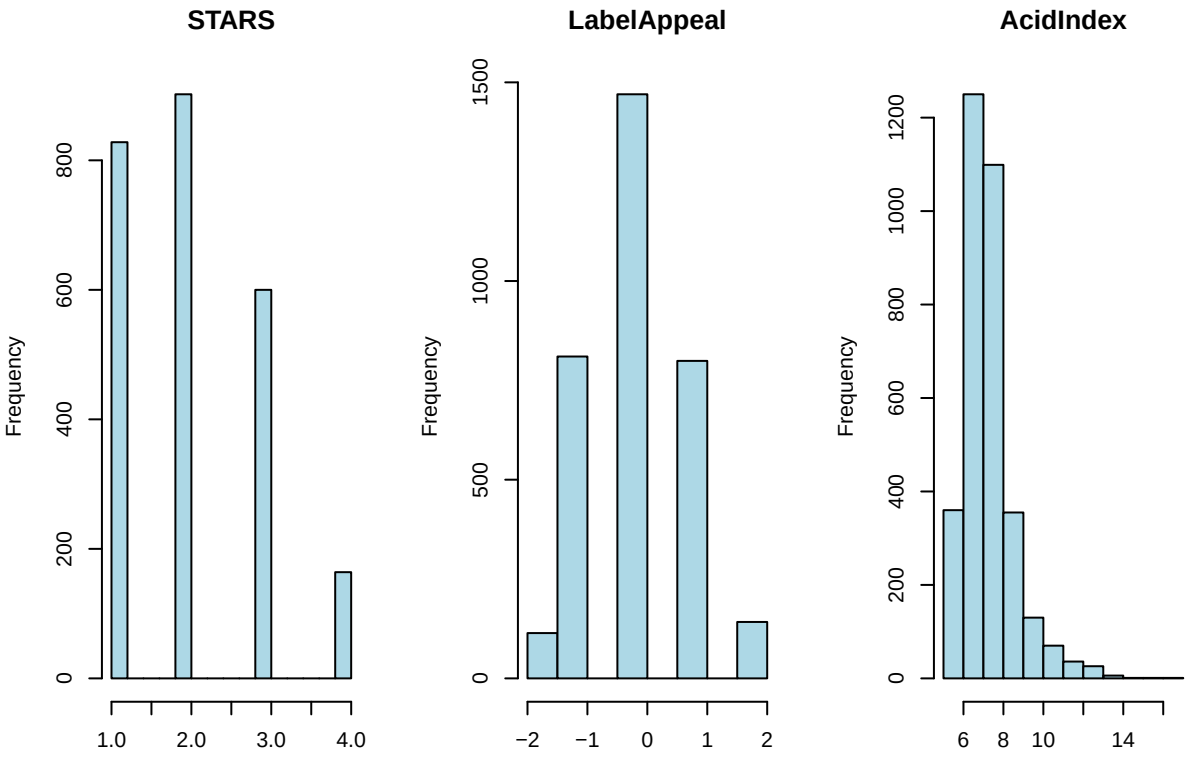
Both models struggled to correctly identify the skewed ends of the target and were most divergent for zero values and the high end number of cases sold.

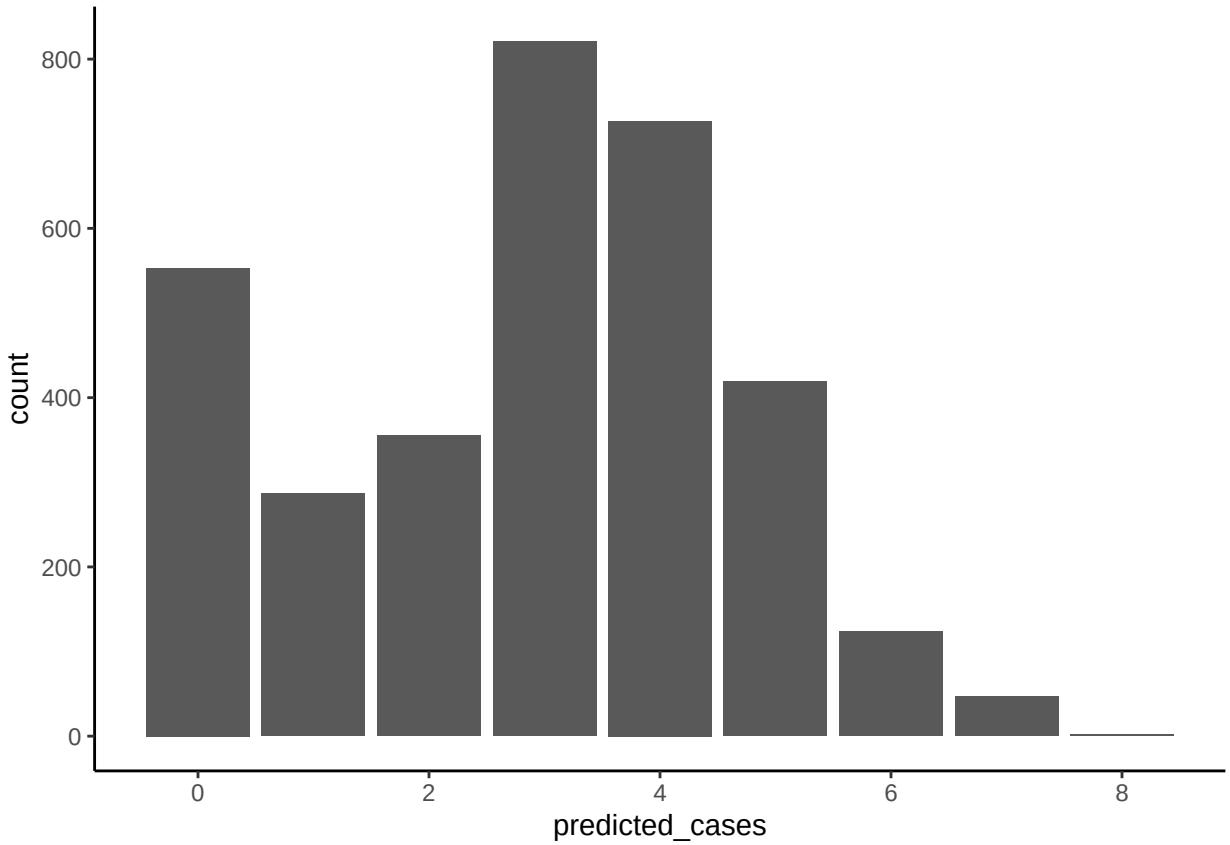
Based on all of the above criteria, it would appear that the Zero Inflated Poisson just edges at the Zero Inflated Negative Binomial given it's slightly lower AIC metric. All the other criteria stacks up fairly evently between these two models.

Evaluation

Let's check for missing values in the evaluation set:

Since most of the missing values are in predictors with negative values that have been excluded from the model, we will need to impute for `STARS` only as the `pH` did not make it as a significant predictor in the final model.


STARS
Frequency
0
200
400
600
800
1.0
2.0
3.0
4.0
LabelAppeal
Frequency
0
500
1000
1500
−2
−1
0
1
2
AcidIndex
Frequency
0
200
400
600
800
1000
1200
6
8
10
14



The final predicted distribution appears to expect less zero values than what was in the training dataset; however, it is in line what prior data and appears to follow the general shape of a Poisson distribution.

Appendix

```
knitr::opts_chunk$set(echo = FALSE)
library(tidyverse)
library(DataExplorer)
library(knitr)
library(mice)
library(cowplot)
library(scales)
library(MASS)
library(glue)
library(corrplot)
library(naniar)
library(car)
library(finalfit)
library(pscl)
library(faraway)
library(caret)
cur_theme <- theme_set(theme_classic())

git_link = 'https://raw.githubusercontent.com/andrewbowen19/businessAnalyticsDataMiningDATA621/main/data
```

```
eval_link = 'https://raw.githubusercontent.com/andrewbowen19/businessAnalyticsDataMiningDATA621/main/da
input_df <- read_csv(git_link,show_col_types = FALSE)
eval_df <- read_csv(eval_link,show_col_types=FALSE)
summary(input_df)
excl_neg <- c('INDEX','TARGET','pH','STARS')

neg_cnts <- sapply(input_df |> dplyr::select(-all_of(excl_neg)), function(x) sum(!is.na(x) & x<0))
knitr::kable(neg_cnts, format = "simple")
excl_eval_neg <- c('IN','TARGET')
neg_eval_cnts <- sapply(eval_df |> dplyr::select(-all_of(excl_eval_neg)), function(x) sum(!is.na(x) & x<
knitr::kable(neg_eval_cnts, format = "simple")
p1 <- plot_missing(input_df, missing_only = TRUE,
                   ggtheme = theme_classic(), title = "Missing Values")


#input_df <- input_df |> dplyr::select(-INDEX)

numeric_train <- input_df[,sapply(input_df, is.numeric)]
par(mfrow=c(4,4))
par(mai=c(.3,.3,.3,.3))
variables <- names(numeric_train)
for (i in 1:(length(variables))) {
  hist(numeric_train[[variables[i]]], main = variables[i], col = "lightblue")
}

gather_df <- input_df %>% dplyr::select(-INDEX) %>%
  gather(key = 'variable', value = 'value')
ggplot(gather_df, aes(variable, value)) +
  geom_boxplot() +
  facet_wrap(. ~variable, scales='free', ncol=4)

ggplot(input_df, aes(x = TARGET, y = after_stat(density))) +
  geom_histogram(binwidth = 1, fill = 'lightblue', color = 'black', alpha = 0.7,
                 aes(color = "Histogram")) +
  geom_line(aes(y = dpois(TARGET, mean(TARGET), log = FALSE), color = "Poisson"),
            linewidth = 1) +
  geom_line(aes(y = dnbinom(x = TARGET, size = 1, prob = 0.2), color = "Negative Binomial"),
            linewidth = 1) +
  geom_line(aes(y = dnorm(x = TARGET, mean = mean(TARGET), sd = sd(TARGET), log = FALSE), color = "Norma
  labs(title = 'Target Histogram and Distribution Overlay',
       x = 'TARGET',
       y = 'Density') +
  scale_color_manual(values = c( "red", "blue", "orange"),
                     labels = c("Negative Binomial","Normal","Poisson")) +
  guides(color = guide_legend(title = "Distributions")) +
  theme_minimal()


print(glue("The response mean: {mean(input_df$TARGET)} and variance: {var(input_df$TARGET)}"))

corrplot(cor(na.omit(input_df)),method="color",diag=FALSE,type="lower",addCoef.col = "black",number.cex=

nonzero_df <- na.omit(input_df) |> filter(TARGET!=0)
```

```
corrplot(cor(nonzero_df),method="color",diag=FALSE,type="lower",addCoef.col = "black",number.cex=0.60)
mod_df <- input_df |> dplyr::select(-INDEX) |> mutate(STARS=as.factor(ifelse(is.na(STARS),0,STARS)))
littles_test <- input_df |>
    mcar_test()
knitr::kable(littles_test, format = "simple")

na_cols <- c("pH", "ResidualSugar", "Chlorides", "FreeSulfurDioxide",'Alcohol','TotalSulfurDioxide','Su
na_col_review <- mod_df |>
    dplyr::select(all_of(na_cols)) |>
    missing_plot()
na_col_review


na_freq <- mod_df |> dplyr::select(all_of(na_cols)) |> mutate(row_index=1:nrow(mod_df)) |> pivot_longer

ggplot(na_freq,aes(x=num_na)) +
    geom_bar() +
    labs(x='Number of NA columns',y='Number of Rows',title='Distribution of NA Values across columns')

print(glue("Only {pull(na_freq |> filter(num_na>1)|>summarise(total=n()))} rows have more than 1 column

set.seed(19)
rows <- sample(nrow(mod_df))
sample <- sample(c(TRUE, FALSE), nrow(mod_df), replace=TRUE,
                 prob=c(0.7,0.3))
train_df <- mod_df[sample, ]
test_df <- mod_df[!sample, ]

avg_impute_train_df <- train_df
avg_impute_test_df <- test_df

avg_impute_train_df$STARS[is.na(avg_impute_train_df$STARS)] <- 0
avg_impute_train_df$STARS <- as.factor(avg_impute_train_df$STARS)
avg_impute_train_df$Sulphates[is.na(avg_impute_train_df$Sulphates)] <- mean(avg_impute_train_df$Sulphat
avg_impute_train_df$TotalSulfurDioxide[is.na(avg_impute_train_df$TotalSulfurDioxide)] <- mean(avg_imput
avg_impute_train_df$FreeSulfurDioxide[is.na(avg_impute_train_df$FreeSulfurDioxide)] <- mean(avg_impute_
avg_impute_train_df$Alcohol[is.na(avg_impute_train_df$Alcohol)] <- mean(avg_impute_train_df$Alcohol, na
avg_impute_train_df$Chlorides[is.na(avg_impute_train_df$Chlorides)] <- mean(avg_impute_train_df$Chlorid
avg_impute_train_df$ResidualSugar[is.na(avg_impute_train_df$ResidualSugar)] <- mean(avg_impute_train_df
avg_impute_train_df$pH[is.na(avg_impute_train_df$pH)] <- mean(avg_impute_train_df$pH, na.rm = TRUE)
avg_impute_train_df$FixedAcidity[is.na(avg_impute_train_df$FixedAcidity)] <- mean(avg_impute_train_df$F

avg_impute_test_df$STARS[is.na(avg_impute_test_df$STARS)] <- 0
avg_impute_test_df$STARS <- as.factor(avg_impute_test_df$STARS)
avg_impute_test_df$Sulphates[is.na(avg_impute_test_df$Sulphates)] <- mean(avg_impute_test_df$Sulphates,
avg_impute_test_df$TotalSulfurDioxide[is.na(avg_impute_test_df$TotalSulfurDioxide)] <- mean(avg_impute_
avg_impute_test_df$FreeSulfurDioxide[is.na(avg_impute_test_df$FreeSulfurDioxide)] <- mean(avg_impute_te
avg_impute_test_df$Alcohol[is.na(avg_impute_test_df$Alcohol)] <- mean(avg_impute_test_df$Alcohol, na.rm
avg_impute_test_df$Chlorides[is.na(avg_impute_test_df$Chlorides)] <- mean(avg_impute_test_df$Chlorides,
avg_impute_test_df$ResidualSugar[is.na(avg_impute_test_df$ResidualSugar)] <- mean(avg_impute_test_df$Re
avg_impute_test_df$pH[is.na(avg_impute_test_df$pH)] <- mean(avg_impute_test_df$pH, na.rm = TRUE)
avg_impute_test_df$FixedAcidity[is.na(avg_impute_test_df$FixedAcidity)] <- mean(avg_impute_test_df$Fixe
```

```r
x <- sapply(avg_impute_test_df, function(x) sum(is.na(x)))
y <- sapply(avg_impute_train_df, function(x) sum(is.na(x)))
sum(x, y) == 0

df_names <- colnames(train_df)
non_na_cols <- df_names[!df_names %in% na_cols]

if (file.exists("imputed_wine_test_df.csv") & file.exists("imputed_wine_train_df.csv")){
    train_df_imputed_input <- read.csv("imputed_wine_train_df.csv", na.strings = "")
    test_df_imputed_input <- read.csv("imputed_wine_test_df.csv", na.strings = "")
    train_df_imputed_input$STARS <- as.factor(train_df_imputed_input$STARS)
    test_df_imputed_input$STARS <- as.factor(test_df_imputed_input$STARS)
}else{
        #Train Data Imputation First
    init = mice(train_df, maxit=0)
    meth = init$method
    predM = init$predictorMatrix


    meth[non_na_cols] = ""

    meth[c("pH")] = "pmm"
    meth[c("ResidualSugar")] = "pmm"
    meth[c("Chlorides")] = "pmm"
    meth[c("FreeSulfurDioxide")] = "pmm"
    meth[c("Alcohol")] = "pmm"
    meth[c("TotalSulfurDioxide")] = "pmm"
    meth[c("Sulphates")] = "pmm"

    imputed_train = mice(train_df, method=meth, predictorMatrix=predM, m=5,
                  printFlag = FALSE)
    train_df_imputed <- complete(imputed_train)
    write.csv(train_df_imputed, "imputed_wine_train_df.csv", row.names = FALSE,
              fileEncoding = "UTF-8")

    #Repeat for test_df
    init = mice(test_df, maxit=0)
    meth = init$method
    predM = init$predictorMatrix
    meth[non_na_cols] = ""
    meth[c("pH")] = "pmm"
    meth[c("ResidualSugar")] = "pmm"
    meth[c("Chlorides")] = "pmm"
    meth[c("FreeSulfurDioxide")] = "pmm"
    meth[c("Alcohol")] = "pmm"
    meth[c("TotalSulfurDioxide")] = "pmm"
    meth[c("Sulphates")] = "pmm"
    imputed_test = mice(test_df, method=meth, predictorMatrix=predM, m=5,
                  printFlag = FALSE)
    imputed_test_df <- complete(imputed_test)
    write.csv(imputed_test_df, "imputed_wine_test_df.csv", row.names = FALSE,
              fileEncoding = "UTF-8")
```

```
}

x <- sapply(train_df_imputed_input, function(x) sum(is.na(x)))
y <- sapply(test_df_imputed_input, function(x) sum(is.na(x)))
sum(x, y) == 0
#need to fix for spacing
sub_vars <- c('Chlorides','pH','Sulphates')
other_nas <- c('ResidualSugar','FreeSulfurDioxide','Alcohol','TotalSulfurDioxide')
impute_train_input <- train_df_imputed_input |>
    dplyr::select(all_of(sub_vars)) |>
    mutate(Set = "Train")
impute_test_input <- test_df_imputed_input |>
    dplyr::select(all_of(sub_vars)) |>
    mutate(Set = "Test")
impute_both <- impute_train_input |>
    bind_rows(impute_test_input)
impute_pivot <- impute_both |>
    pivot_longer(!Set, names_to = "Variable", values_to = "Value")
impute_plot <- impute_pivot |>
    ggplot(aes(x = Value)) +
    geom_density(fill = "lightblue", color = "black") +
    labs(y = "Density") +
    facet_grid(rows = vars(Set), cols = vars(Variable),
               switch = "y", scales = "free_x")
impute_plot



impute_train_input2 <- train_df_imputed_input |>
    dplyr::select(all_of(other_nas)) |>
    mutate(Set = "Train")
impute_test_input2 <- test_df_imputed_input |>
    dplyr::select(all_of(other_nas)) |>
    mutate(Set = "Test")
impute_both2 <- impute_train_input2 |>
    bind_rows(impute_test_input2)
impute_pivot2 <- impute_both2 |>
    pivot_longer(!Set, names_to = "Variable", values_to = "Value")
impute_plot2 <- impute_pivot2 |>
    ggplot(aes(x = Value)) +
    geom_density(fill = "lightblue", color = "black") +
    labs(y = "Density") +
    facet_grid(rows = vars(Set), cols = vars(Variable),
               switch = "y", scales = "free_x")
impute_plot2

acid_bc <- boxcox(lm(train_df_imputed_input$AcidIndex ~ 1))
acid_lambda <- acid_bc$x[which.max(acid_bc$y)]

train_df_imputed_input |> mutate(bc_acidindex=(AcidIndex ^ round(acid_lambda) - 1)/round(acid_lambda)) %
    ggplot(aes(x=bc_acidindex)) +
    geom_histogram() +
    labs(title='Testing Proposed Box Cox (-1) Transformation of AcidIndex',x='AcidIndex Transformed')
```

```
corrplot(cor(train_df_imputed_input |> dplyr::select(-STARS)),method="color",diag=FALSE,type="lower",ad
train_df_imputed_input <- train_df_imputed_input |> mutate(TARGET.TF=ifelse(TARGET==0,TARGET+0.001,TARG
target_bc <- boxcox(lm(train_df_imputed_input$TARGET.TF  ~ 1))
target_lambda <- target_bc$x[which.max(target_bc$y)]
target_lambda
train_df_imputed_input <- train_df_imputed_input |> mutate(TARGET.TF=log(ifelse(TARGET==0,0.5,TARGET)))
hist(train_df_imputed_input$TARGET.TF)
mod1_pois <- glm(TARGET ~ STARS + LabelAppeal + pH + Density + AcidIndex, family = 'poisson',
                 data = avg_impute_train_df)
mod1_pois <- stepAIC(mod1_pois, trace = 0)
summary(mod1_pois)

with(mod1_pois, cbind(res.deviance = deviance, df = df.residual,
  p = pchisq(deviance, df.residual, lower.tail=FALSE)))
e_y <- predict(mod1_pois,type="response")
plot(log(e_y+1),log(avg_impute_train_df$TARGET+1),xlim=c(0,4),xlab='Log Scaled Estimated Cases',ylab='L
title(main='Comparing Estimated vs Observed number of cases of wine sold')
faraway::halfnorm(residuals(mod1_pois))

car::influencePlot(mod1_pois)
res <- residuals(mod1_pois, type="deviance")
#abline(h=0, lty=2)
qqnorm(res)
qqline(res)
## Check for over/underdispersion in the model
E2 <- resid(mod1_pois, type = "pearson")
N  <- nrow(train_df_imputed_input)
p  <- length(coef(mod1_pois))
sum(E2^2) / (N - p)
mod2_zip <- pscl::zeroinfl(formula=TARGET ~ STARS + LabelAppeal + AcidIndex |STARS + LabelAppeal,data=t


summary(mod2_zip)
m2null <- update(mod2_zip, . ~ 1)

pchisq(2 * (logLik(mod2_zip) - logLik(m2null)), df = 7, lower.tail = FALSE)
plot(residuals(mod2_zip) ~
      fitted(mod2_zip),xlab="Fitted",ylab="Residuals")
abline(h=0)
res_mod2 <- residuals(mod2_zip,'pearson')
qqnorm(res_mod2)
qqline(res_mod2)

abline(0, 1, col = "red")
E2_zip2 <- resid(mod2_zip, type = "pearson")
N_zip2 <- nrow(input_df)
p_zip2 <- length(coef(mod2_zip)) # '+1' is for variance parameter in NB
sum(E2_zip2^2) / (N_zip2 - p_zip2)
AIC(mod1_pois,mod2_zip)

mod3_nb <- glm.nb(TARGET ~ STARS + LabelAppeal + AcidIndex,data = avg_impute_train_df)

# summary of results
```

```
summary(mod3_nb)

pchisq(2 * (logLik(mod3_nb) - logLik(mod1_pois)), df = 1, lower.tail = FALSE)
plot(residuals(mod3_nb) ~
      fitted(mod3_nb),xlab="Fitted",ylab="Residuals")
abline(h=0)
res_mod_nb <- residuals(mod3_nb,'pearson')
qqnorm(res_mod_nb)
qqline(res_mod_nb)

E2_mod3 <- resid(mod3_nb, type = "pearson")
N  <- nrow(input_df)
p_mod3  <- length(coef(mod3_nb)) + 1 # added for variance parameter
sum(E2_mod3^2) / (N - p_mod3)
mod4_zinb <- pscl::zeroinfl(formula=TARGET ~ STARS + LabelAppeal + AcidIndex| STARS + LabelAppeal,data=

summary(mod4_zinb)
res_mod4 <- residuals(mod4_zinb,'pearson')
qqnorm(res_mod4)
qqline(res_mod4)

abline(0, 1, col = "red")
E2_mod4 <- resid(mod4_zinb, type = "pearson")
N  <- nrow(input_df)
p_mod4  <- length(coef(mod4_zinb)) + 1 # added for variance parameter
sum(E2_mod4^2) / (N - p_mod4)
aic_compare2<- AIC(mod1_pois,mod2_zip,mod3_nb,mod4_zinb)
knitr::kable(aic_compare2, format = "simple")
mod5_mlr <- lm(TARGET ~ STARS + LabelAppeal + pH + Density + AcidIndex,data=avg_impute_train_df)
mod5_mlr <- stepAIC(mod5_mlr, trace = 0)
summary(mod5_mlr)
knitr::kable(vif(mod5_mlr), format = "simple")
par(mfrow=c(2,2))
plot(mod5_mlr)
mod6_mlr <- lm(TARGET.TF ~ STARS + LabelAppeal + pH + Density + AcidIndex - TARGET,data=train_df_imputed
mod6_mlr <- stepAIC(mod6_mlr, trace = 0)
summary(mod6_mlr)
knitr::kable(vif(mod6_mlr), format = "simple")
par(mfrow=c(2,2))
plot(mod6_mlr)
AIC(mod5_mlr,mod6_mlr)
AIC(mod2_zip,mod4_zinb)
zip_pred <- predict(mod2_zip,test_df_imputed_input,type='response')
zip_zero <- predict(mod2_zip,test_df_imputed_input,type='zero')
zinb_pred <- predict(mod4_zinb,test_df_imputed_input,type='response')
zinb_zero <- predict(mod4_zinb,test_df_imputed_input,type='zero')
holdout_zi <- as.data.frame(cbind(test_df_imputed_input,zip_pred,zinb_pred,zip_zero,zinb_zero)) |> dply

knitr::kable(cbind(holdout_zi |> group_by(zip_match) |> summarize(cnt=n()),holdout_zi |> group_by(zinb_
ggplot(data=holdout_zi,aes(x=TARGET,fill=zip_fact)) +
    geom_bar() +
    scale_fill_manual(values=c('black', 'red'),labels = c("Correct","Incorrect")) +
    guides(color = guide_legend(title = "Discrete Predictions"))
```

```
ggplot(data=holdout_zi,aes(x=TARGET,fill=zinb_fact)) +
    geom_bar() +
    scale_fill_manual(values=c('black', 'red'),labels = c("Correct","Incorrect")) +
    guides(color = guide_legend(title = "Discrete Predictions"))
ggplot(holdout_zi,aes(x=TARGET,y=is_zip_zero,fill=zip_res)) +
    xlim(-1,8) +
    geom_tile() +
    scale_fill_distiller(palette = "Spectral") +
    labs(y='Predicted Value') +
    guides(color = guide_legend(title = "Target - Prediction"))
ggplot(holdout_zi,aes(x=TARGET,y=is_zinb_zero,fill=zinb_res)) +
    xlim(-1,8) +
    geom_tile() +
    scale_fill_distiller(palette = "Spectral") +
    labs(y='Predicted Value')+
    guides(color = guide_legend(title = "Target - Prediction"))
p1 <- plot_missing(eval_df, missing_only = TRUE,
                   ggtheme = theme_classic(), title = "Missing Values")

in_model_vars <- c('STARS','LabelAppeal','AcidIndex')
dist_eval_df <- eval_df |> dplyr::select(all_of(in_model_vars))

par(mfrow=c(1,3))
variables_eval <- names(dist_eval_df)
for (i in 1:(length(variables_eval))) {
  hist(dist_eval_df[[variables_eval[i]]], main = variables_eval[i], col = "lightblue",xlab='')
}

mod_eval_df <- dist_eval_df |> mutate(STARS=as.factor(ifelse(is.na(STARS),0,STARS)))
eval_pred <- predict(mod2_zip,mod_eval_df,type='response')
eval_pred_zero <- predict(mod2_zip,mod_eval_df,type='zero')
final_pred_df <- as.data.frame(cbind(mod_eval_df,eval_pred,eval_pred_zero)) |> mutate(predicted_cases=i

ggplot(final_pred_df,aes(x=predicted_cases)) +
    geom_bar()


write.csv(final_pred_df,'HW5_Eval_Predictions.csv')
```