Wine Sales

Biguzzi, Connin, Greenlee, Moscoe, Sooklall, Telab, and Wright

12/03/2021

Introduction

Using the information about sample wine orders by restaurants and wine stores after a tasting, how can we predict wine sales by various wine characteristics? Using the `wine` dataset with 12,000 entries and variables mostly related to the chemical properties of each wine, we will build a count regression model to predict the number of cases of wine that will be sold. In practice, if a wine manufacturer can predict which wines will lead to greater sales, they can choose to offer those at more tastings in restaurants and wine stores.

Using the training data set we will build:

- four different poisson regression models
- three different negative binomial regression models
- two different multiple linear regression models

In this report we will:

- explore the data
- transform the data to meet conditions of count modeling
- compare models
- select an optimal model
- generate predictions for the evaluation data set

Data Exploration

As part of our initial data exploration below, we find the following issues that will be handled in the Data Preparation section:

- large amounts of negative values for 8 of the chemical measures, which should be adjusted to 0
- 26.25% of cases have a missing `STARS` value
- `LabelAppeal` and `STARS` have imported as integers, but as could be interpreted as categorical variables due to lack of continuity
- many of the chemical variables could benefit from log transformations, after seeing the normality plots
- very few variables are correlated with the `TARGET` beyond `STARS` and `LabelAppeal`, consider dropping some

First Look

Taking a look at the structure of the dataset we have 12,795 cases and 14 potential predictor variables. All variables are numeric. We'll remove the `INDEX` variable as it isn't needed in model building.

```
## 'data.frame':    12795 obs. of  16 variables:
##  $ ï..INDEX          : int  1 2 4 5 6 7 8 11 12 13 ...
##  $ TARGET            : int  3 3 5 3 4 0 0 4 3 6 ...
##  $ FixedAcidity      : num  3.2 4.5 7.1 5.7 8 11.3 7.7 6.5 14.8 5.5 ...
##  $ VolatileAcidity   : num  1.16 0.16 2.64 0.385 0.33 0.32 0.29 -1.22 0.27 -0.22 ...
##  $ CitricAcid        : num  -0.98 -0.81 -0.88 0.04 -1.26 0.59 -0.4 0.34 1.05 0.39 ...
##  $ ResidualSugar     : num  54.2 26.1 14.8 18.8 9.4 ...
##  $ Chlorides         : num  -0.567 -0.425 0.037 -0.425 NA 0.556 0.06 0.04 -0.007 -0.277 ...
##  $ FreeSulfurDioxide : num  NA 15 214 22 -167 -37 287 523 -213 62 ...
##  $ TotalSulfurDioxide: num  268 -327 142 115 108 15 156 551 NA 180 ...
##  $ Density           : num  0.993 1.028 0.995 0.996 0.995 ...
##  $ pH                : num  3.33 3.38 3.12 2.24 3.12 3.2 3.49 3.2 4.93 3.09 ...
##  $ Sulphates         : num  -0.59 0.7 0.48 1.83 1.77 1.29 1.21 NA 0.26 0.75 ...
##  $ Alcohol           : num  9.9 NA 22 6.2 13.7 15.4 10.3 11.6 15 12.6 ...
##  $ LabelAppeal       : int  0 -1 -1 -1 0 0 0 1 0 0 ...
##  $ AcidIndex         : int  8 7 8 6 9 11 8 7 6 8 ...
##  $ STARS             : int  2 3 3 1 2 NA NA 3 NA 4 ...
```

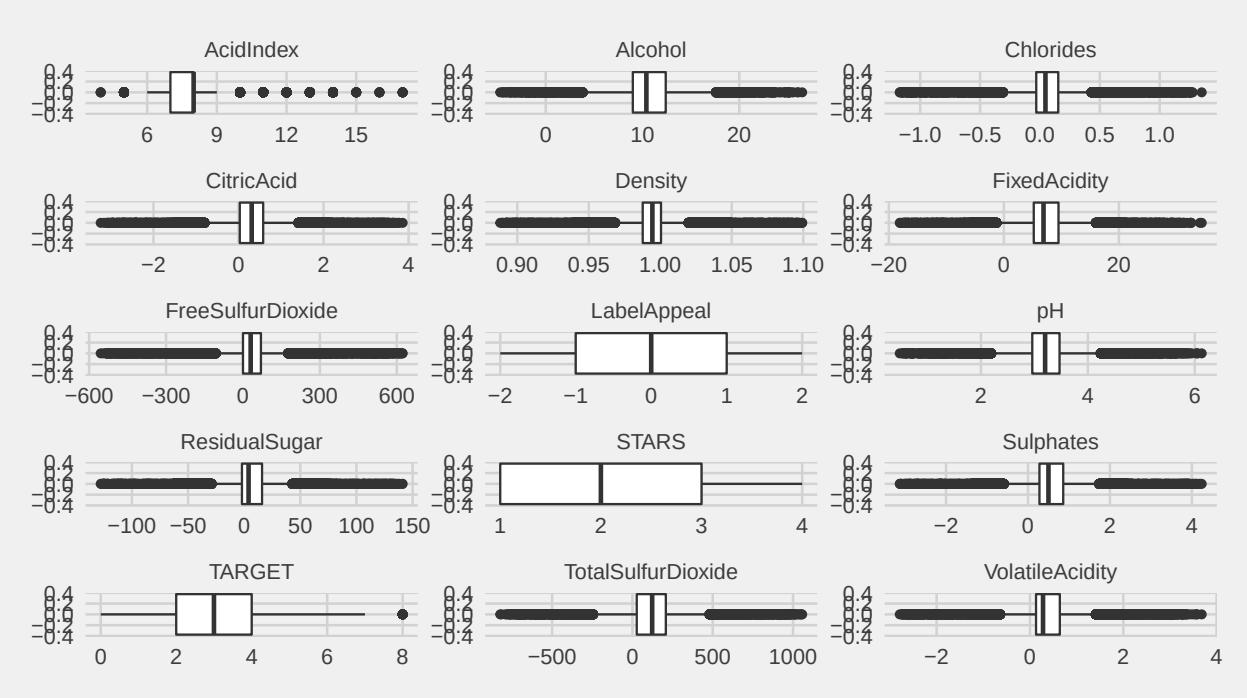
The table below shows the range of the TARGET (# cases purchased) ranges from 0 - 8. We also see a large amount of negative values across some of the chemical variables. We will need to deal with these values and/or cases later.

variables	min	mean	median	max	zero	minus
TARGET	0.00	3.03	3.00	8.00	2,734	0
FixedAcidity	-18.10	7.08	6.90	34.40	39	1,621
VolatileAcidity	-2.79	0.32	0.28	3.68	18	2,827
CitricAcid	-3.24	0.31	0.31	3.86	115	2,966
ResidualSugar	-127.80	5.42	3.90	141.15	6	3,136
Chlorides	-1.17	0.05	0.05	1.35	5	3,197
FreeSulfurDioxide	-555.00	30.85	30.00	623.00	11	3,036

variables	min	mean	median	max	zero	minus
TotalSulfurDioxide	-823.00	120.71	123.00	1,057.00	7	2,504
Density	0.89	0.99	0.99	1.10	0	0
pH	0.48	3.21	3.20	6.13	0	0
Sulphates	-3.13	0.53	0.50	4.24	22	2,361
Alcohol	-4.70	10.49	10.40	26.50	2	118
LabelAppeal	-2.00	-0.01	0.00	2.00	5,617	3,640
AcidIndex	4.00	7.77	8.00	17.00	0	0
STARS	1.00	2.04	2.00	4.00	0	0

Checking for Normality

We check the distribution of all of the predictor variables with boxplots. All variables except for `STARS` and `LabelAppeal` appear to have a small IQRs with large ranges of outliers. This could limit the type of modeling that is appropriate to those don't have strict normality assumptions.

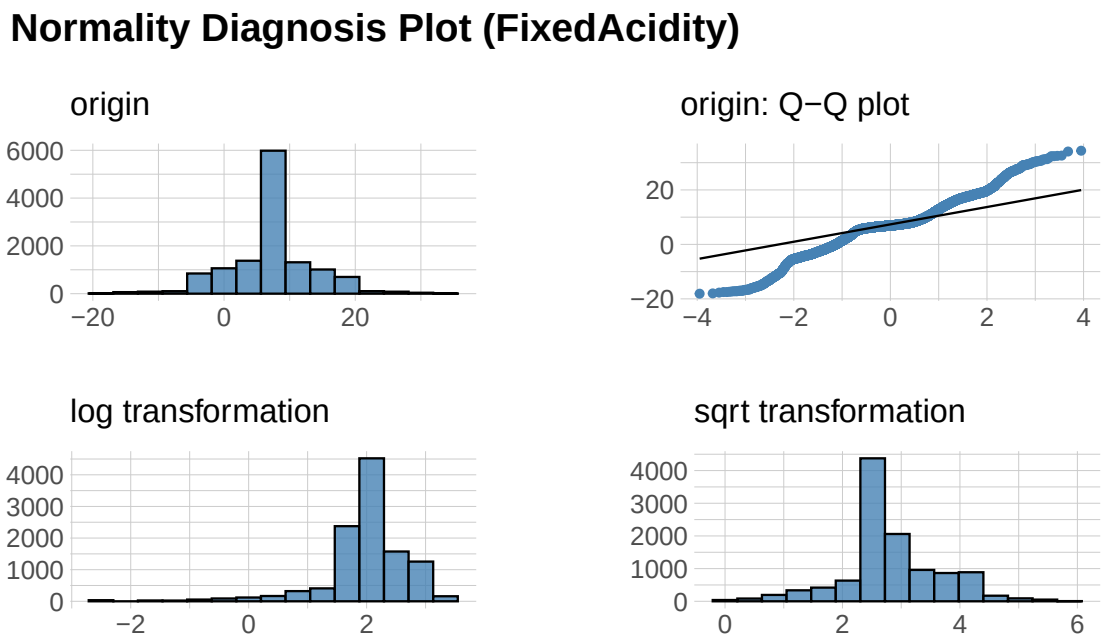


Using the Shapiro-Wilk normality test on each of the variables, we see all of the p-values are less than 0.05 which means none of our variables are normally distributed in their raw form.

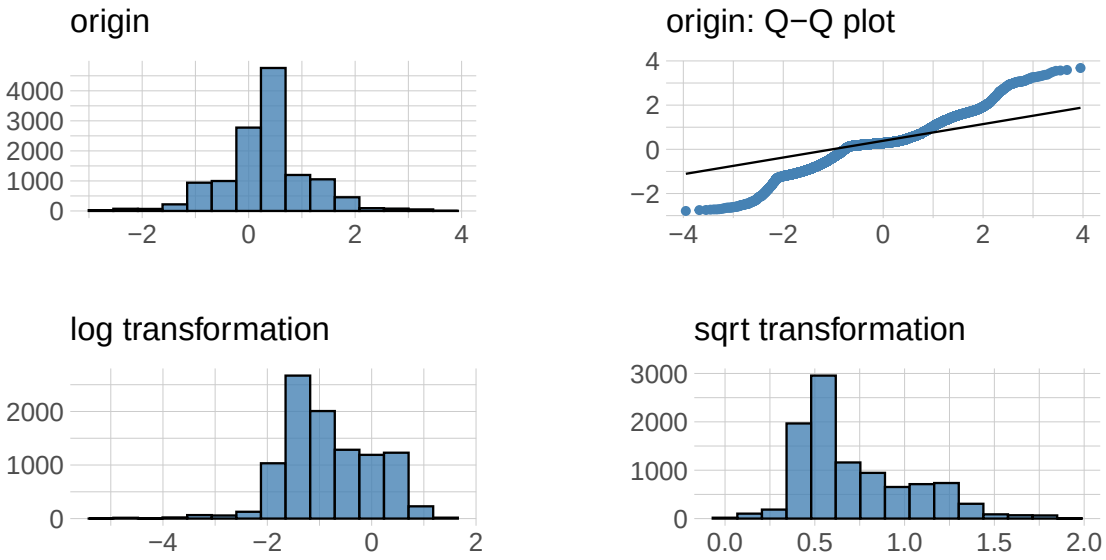
vars	statistic	p_value	sample
Alcohol	0.973	0.000	5,000.000
TotalSulfurDioxide	0.961	0.000	5,000.000
pH	0.960	0.000	5,000.000

vars	statistic	p_value	sample
FixedAcidity	0.951	0.000	5,000.000
Sulphates	0.945	0.000	5,000.000
VolatileAcidity	0.947	0.000	5,000.000
CitricAcid	0.947	0.000	5,000.000
ResidualSugar	0.941	0.000	5,000.000
FreeSulfurDioxide	0.938	0.000	5,000.000
Chlorides	0.929	0.000	5,000.000
Density	0.930	0.000	5,000.000
TARGET	0.902	0.000	5,000.000
LabelAppeal	0.895	0.000	5,000.000
STARS	0.852	0.000	5,000.000
AcidIndex	0.843	0.000	5,000.000

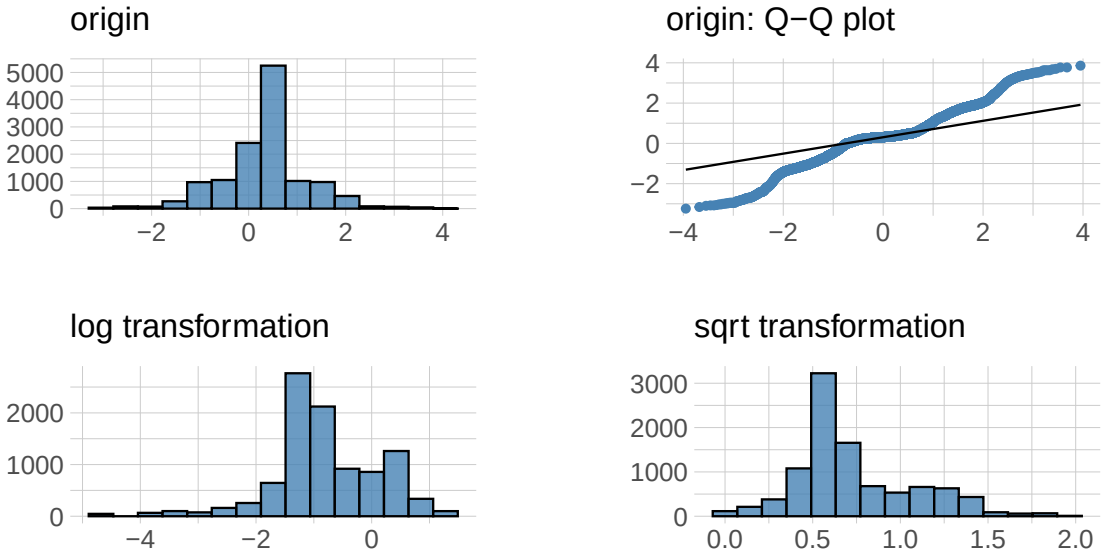
Below we visualize with a histogram and Q-Q plot for each variable for which we want to check for normality, excluding the `TARGET` variable and variables we will convert to factors later. It appears that all 11 of the variables would benefit from a log transformation, which will likely become even more necessary once we correct for the un-interpretable negative values many of these variables have in the original dataset.



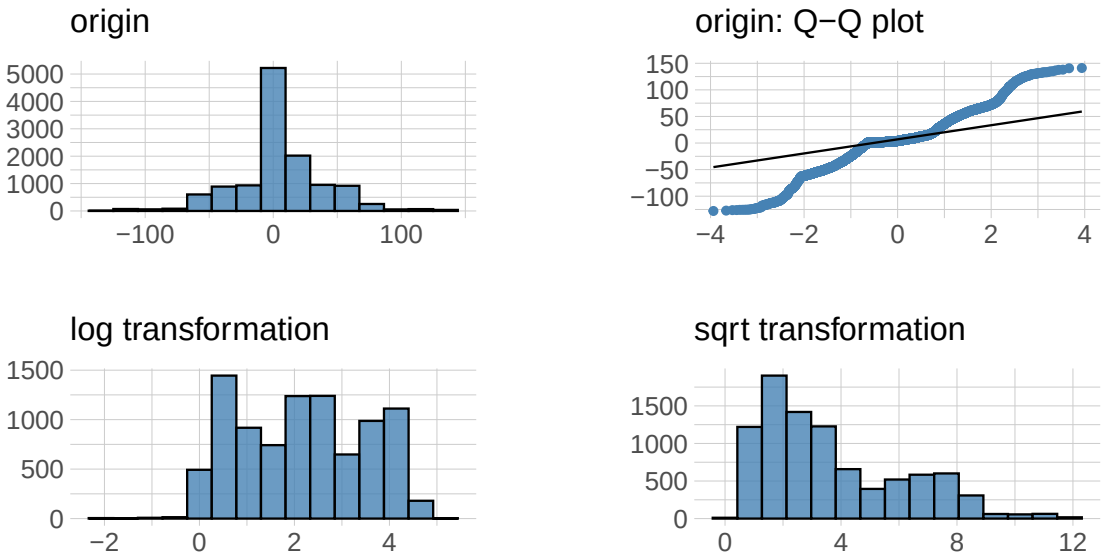
Normality Diagnosis Plot (VolatileAcidity)



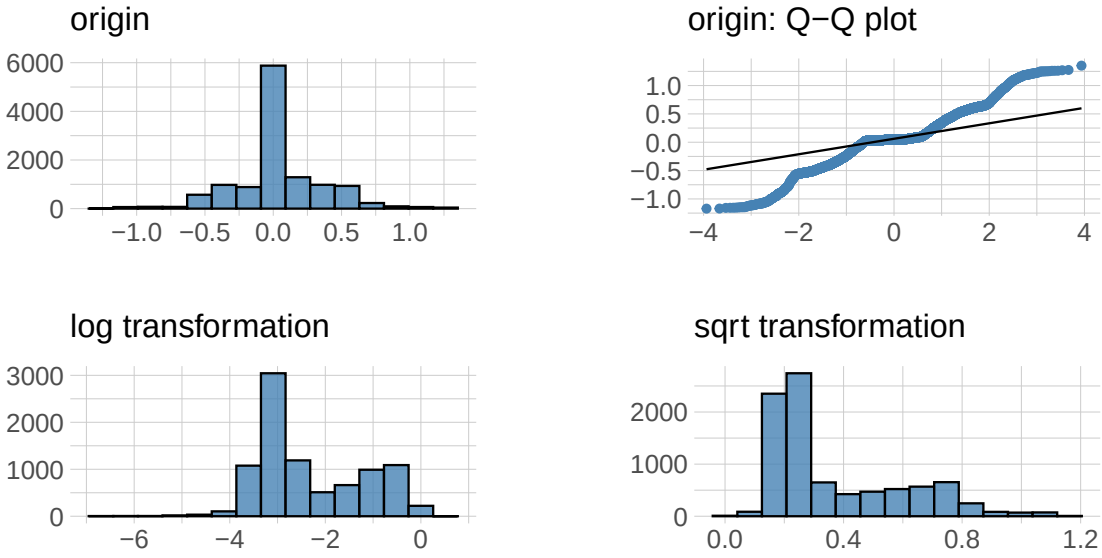
Normality Diagnosis Plot (CitricAcid)



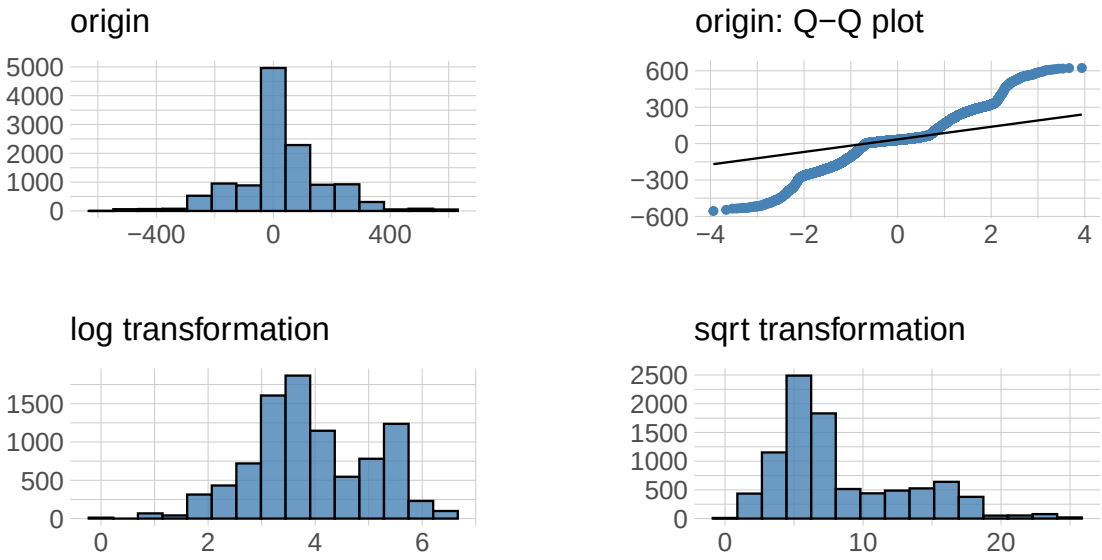
Normality Diagnosis Plot (ResidualSugar)



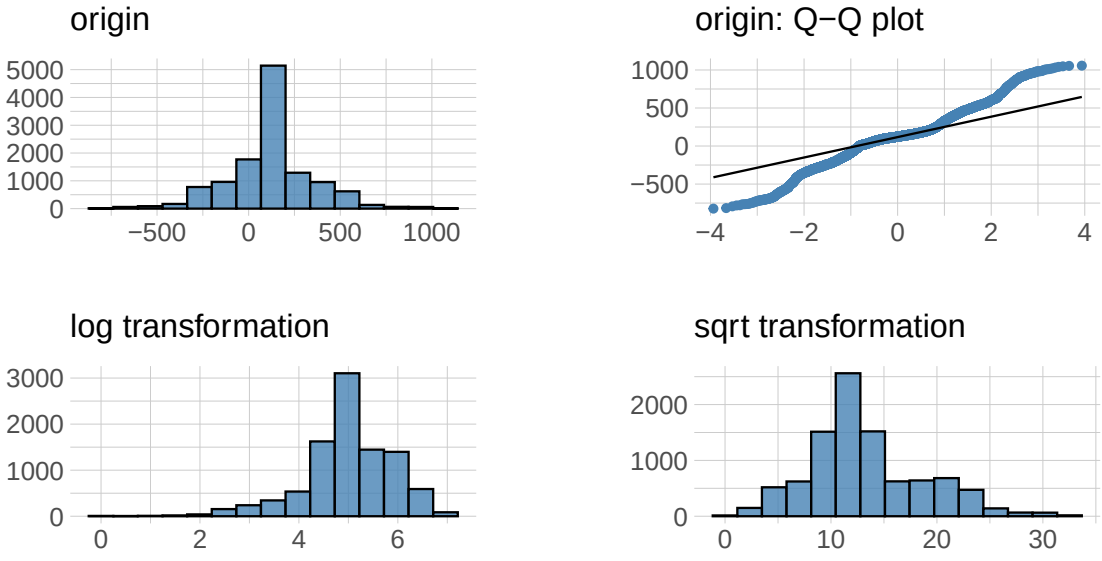
Normality Diagnosis Plot (Chlorides)



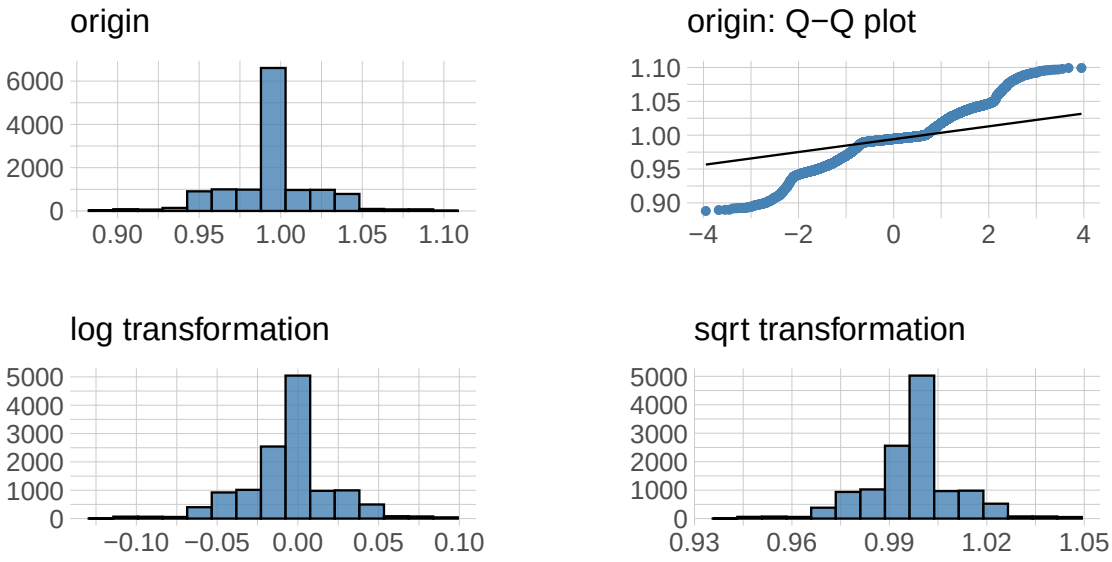
Normality Diagnosis Plot (FreeSulfurDioxide)



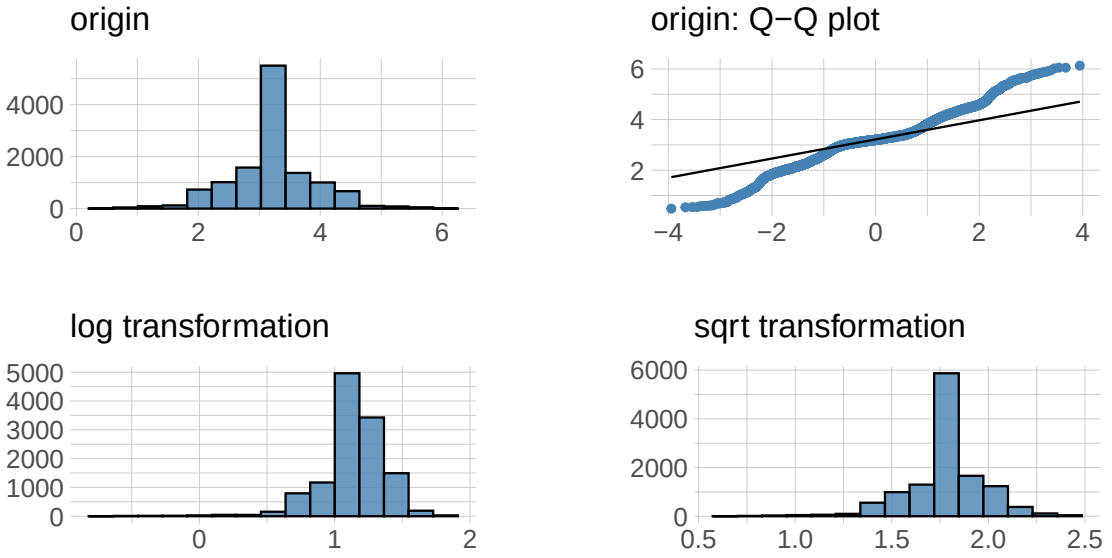
Normality Diagnosis Plot (TotalSulfurDioxide)



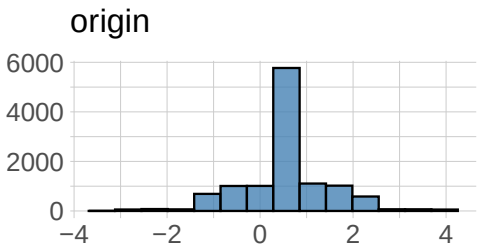
Normality Diagnosis Plot (Density)

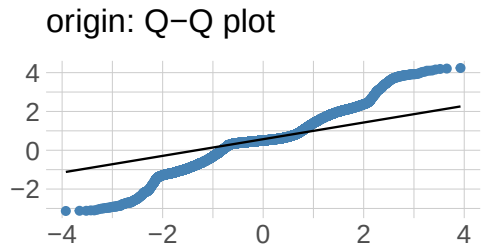


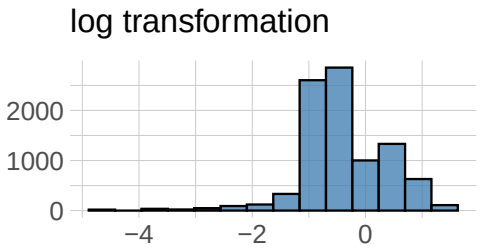
Normality Diagnosis Plot (pH)

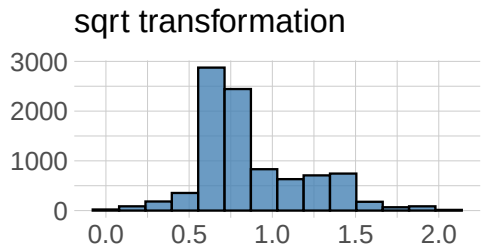


Normality Diagnosis Plot (Sulphates)

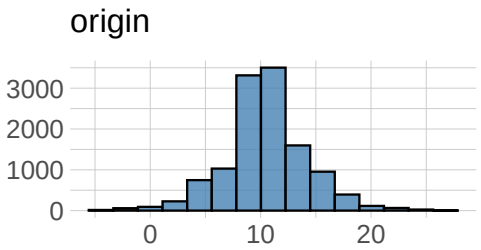


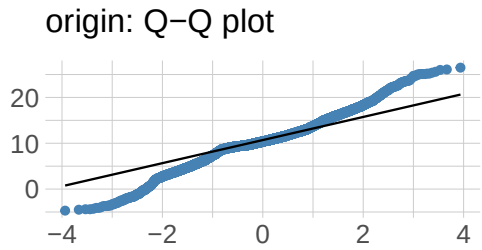


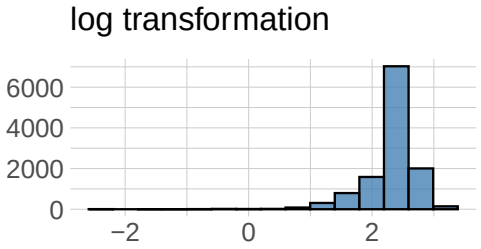


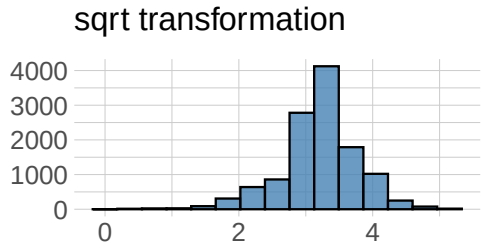


Normality Diagnosis Plot (Alcohol)









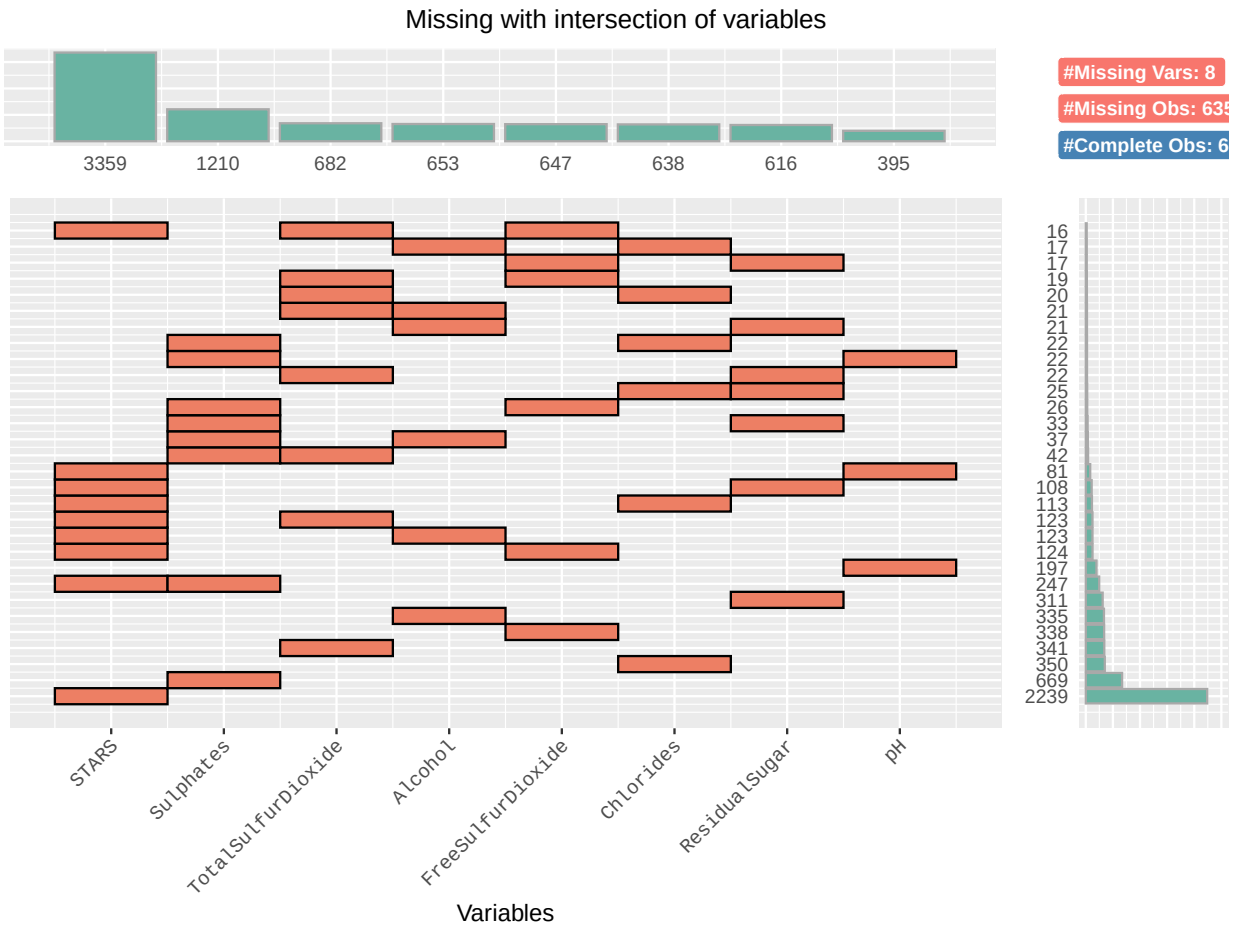
Missingness

With regards to missingness, we have 6 variables with data for every case. This leaves 8 variables that have some degree of missing data, with the worst being the `STARS` variable with 26.25% of cases missing a value for `STARS` (the wine rating by experts). None of the remaining 7 variables have more than 10% missing, so imputing values will be considered.

variables	types	missing_count	missing_percent
STARS	integer	3,359	26.25
Sulphates	numeric	1,210	9.46
TotalSulfurDioxide	numeric	682	5.33
Alcohol	numeric	653	5.10
FreeSulfurDioxide	numeric	647	5.06
Chlorides	numeric	638	4.99
ResidualSugar	numeric	616	4.81
pH	numeric	395	3.09

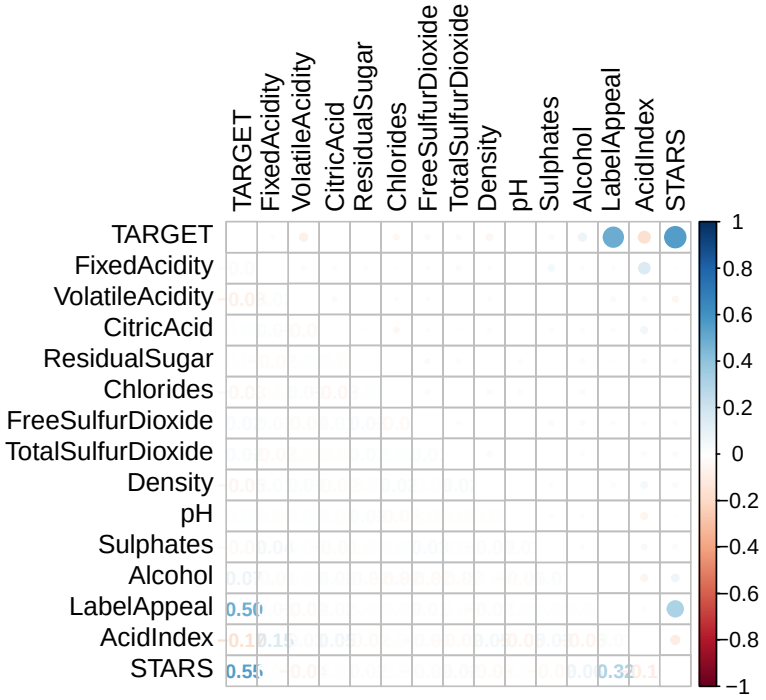
Looking for patterns in missingness can be telling. Below the 8 variables with at least some missingness are plotted for the top 30 most frequent combinations of missing values.

We don't see any major relationships of missingness that affect a large number of cases in the dataset.


Missing with intersection of variables
3359
1210
682
653
647
638
616
395
#Missing Vars: 8
#Missing Obs: 635
#Complete Obs: 6
16
17
17
19
20
21
21
22
22
22
25
26
33
37
42
81
108
113
123
123
124
197
247
311
335
338
341
350
669
2239
STARS
Sulphates
TotalSulfurDioxide
Alcohol
FreeSulfurDioxide
Chlorides
ResidualSugar
pH
Variables

Correlation

We see a very sparse correlation plot below. `LabelAppeal` and `STARS` jump out as having the largest positive correlation with `TARGET`, with a faint negative correlation between `AcidIndex` and the `TARGET`. We also see a correlation between `LabelAppeal` and `STARS`.



Check for Outliers

The table below shows the count and percent of outliers within each variable arranged with the highest outlier count starting at the top. The outliers_ratio shows the percent of outliers identified withing all cases, which is greater than 5% for the first 12 variables. We will recheck that table after transformations in the Data Preparation section.

variables	outliers_cnt	outliers_ratio	outliers_mean	with_mean	without_mean
Density	3,823	29.879	0.994	0.994	0.994
FreeSulfurDioxide	3,712	29.011	24.169	30.846	33.783
ResidualSugar	3,298	25.776	3.499	5.419	6.132
Chlorides	3,021	23.611	0.048	0.055	0.057
CitricAcid	2,688	21.008	0.288	0.308	0.314
Sulphates	2,606	20.367	0.463	0.527	0.546
VolatileAcidity	2,599	20.313	0.208	0.324	0.354
FixedAcidity	2,455	19.187	6.761	7.076	7.150
pH	1,864	14.568	3.203	3.208	3.208
TotalSulfurDioxide	1,590	12.427	127.610	120.714	119.672
AcidIndex	1,151	8.996	10.552	7.773	7.498
Alcohol	928	7.253	9.470	10.489	10.574
TARGET	17	0.133	8.000	3.029	3.022
LabelAppeal	0	0.000		-0.009	-0.009
STARS	0	0.000		2.042	2.042

Data Preparation

To address the issues in the data we discovered in the data exploration section, we will:

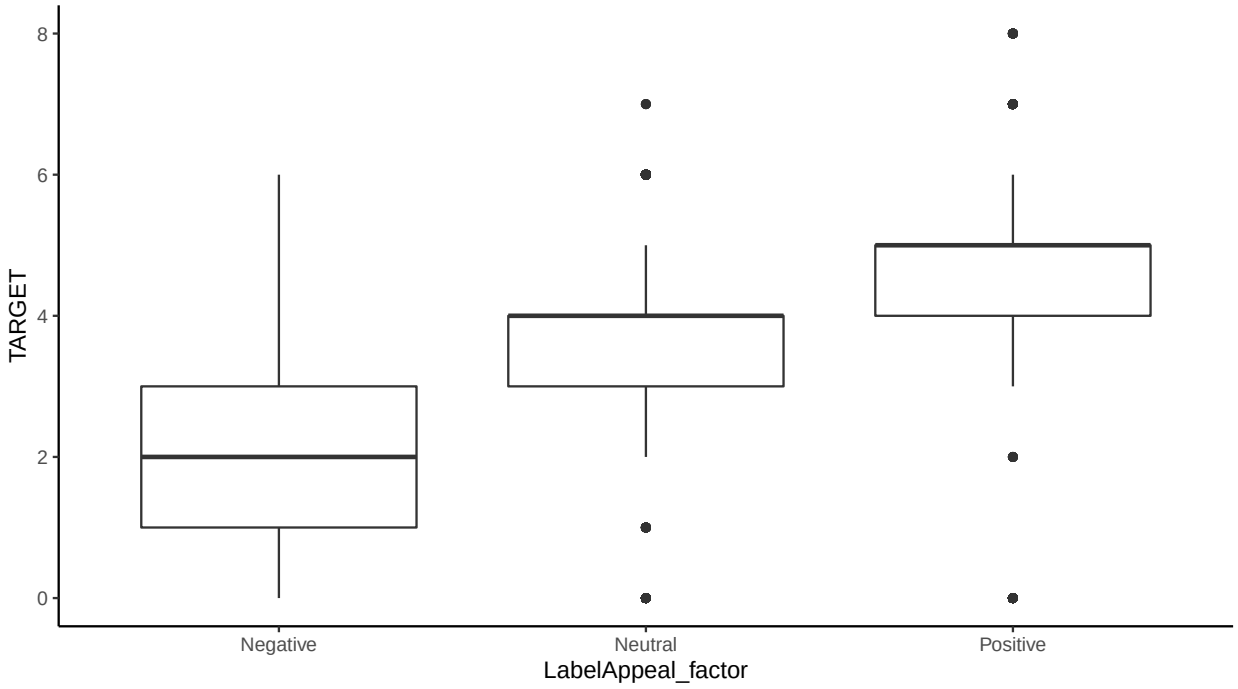
- **CHANGE DATA TYPE** to factors for `STARS`, interpreting as categorical variables due to lack of continuity
- **Bucket `LabelAppeal` Ratings** into Negative, Neutral, and Positive scores
- **ASSIGN ZERO to all negative values** for 8 of the chemical measures
- **IMPUTE missing values** for variables (except `STARS`) using the median
- **Create missing `STARS` variable** for 26.25% of cases that have a missing `STARS` value
- **Drop variables** that have no correlation to `TARGET` even after above transformations
- **LOG transformation** for all of the numeric chemical variables that were not normally distributed
- **Recheck** normality, outliers, and correlation

Change Data Type of `STARS`

We set the `STARS` variable as a factor as it's a rating and not a continuous scale.

Bucket `LabelAppeal` Ratings

Switch `LabelAppeal` to a factor after coding the ratings into buckets of Negative, Neutral, and Positive. This will be easier to interpret as well. The boxplots after this change show how increased `LabelAppeal` seems to be positively associated with the `TARGET`.



Fix Negative Values

With 21,766 negative values across many of the chemical variables, for which it doesn't make sense to have a negative value, we convert these to positive zero so we retain some value (as opposed to omitting these

measures or cases). Variables include: `FixedAcidity`, `VolatileAcidity`, `CitricAcid`, `ResidualSugar`, `Chlorides`, `FreeSulfurDioxide`, `TotalSulfurDioxide`, `Sulphates`, `Alcohol`.

It's very common in chemical analyses that negative values reflect sample concentrations below the instrumental detection limit, thus we feel justified in treating these measures as zero.

Impute Missing Values

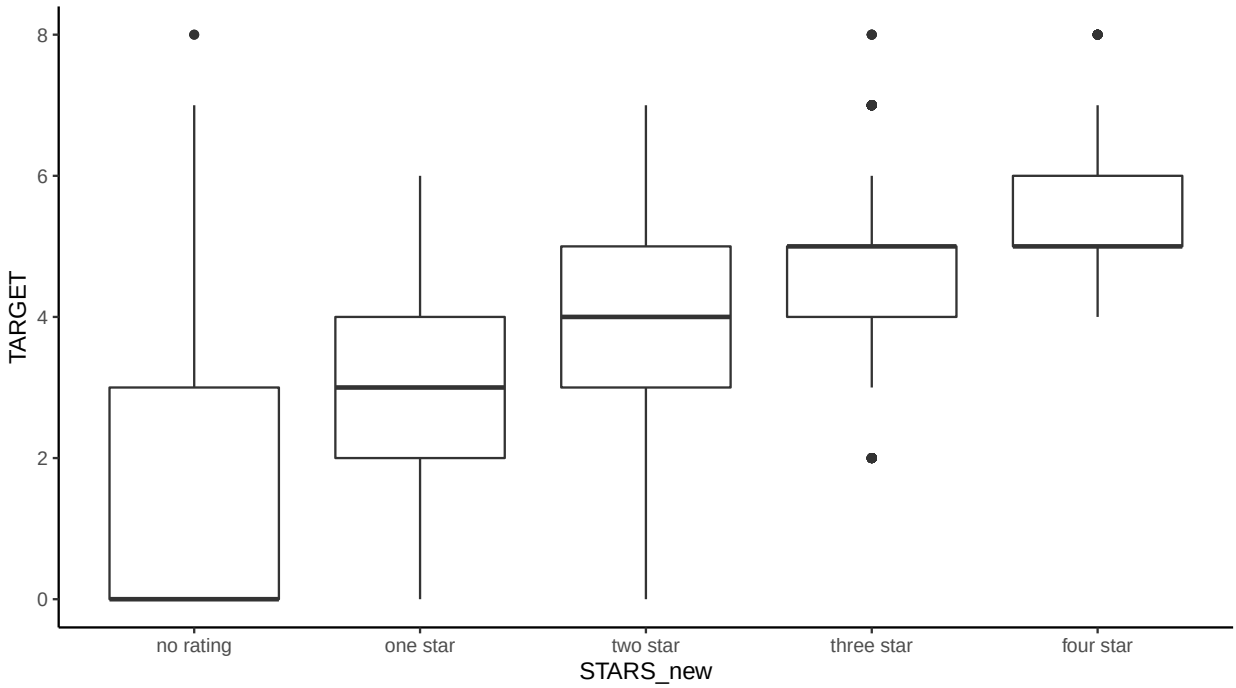
For the 7 variables that had <10% missing values, we will impute the missing values with the median for that variable. Variables include: `ResidualSugar`, `Chlorides`, `FreeSulfurDioxide`, `TotalSulfurDioxide`, `pH`, `Sulphates`, `Alcohol`.

Flag `STARS` Missing Values

With 26.25% of cases missing a `STARS` rating, we need to consider if we should drop the variable from modeling, impute the missing values, or flag these cases. There are some guidelines in data science that if there are beyond 30% missing values, the variable may be best dropped. Theoretically, the `STARS` variable should represent something close to our `TARGET` variable as we'd expect a relationship between high expert reviews and high case sales.

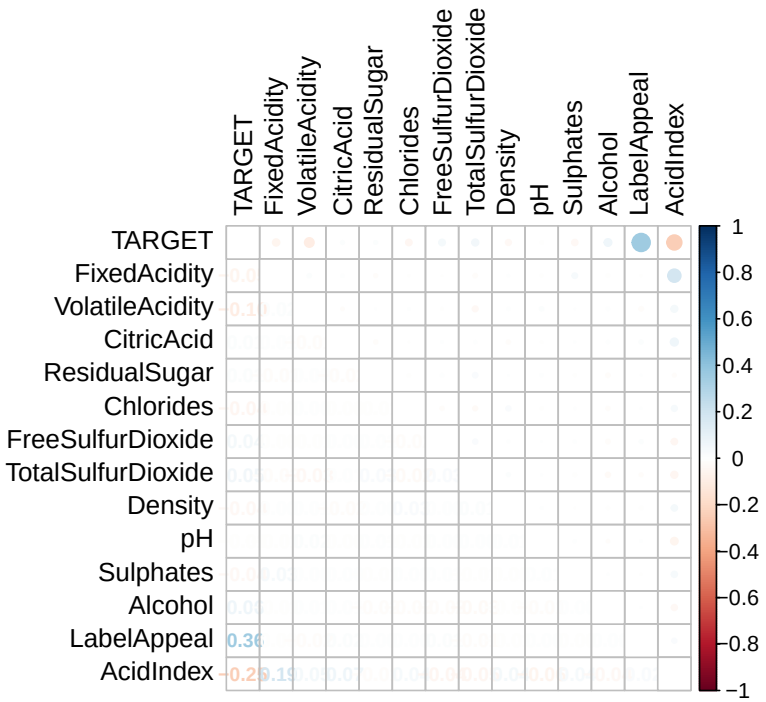
Since this seems like too valuable of a variable to drop, and considering that depending on how the `STARS` data was obtained, a missing value could indicate a lesser quality or less popular wine (such that it hasn't been rated by experts), we choose to create a new variable where a '1' indicates a missing `STARS` value. We leave the NAs in the original `STARS` variable. We set the `STARS_new` variable as a factor with 5 levels.

The box plots below show increasing sales as the star rating increasing, which is intuitive.



Drop Variables

The correlation plot below show that, even after some of the above transformations, the majority of the numeric variables are not correlated with the `TARGET`. We choose to drop `FixedAcidity`, `VolatileAcidity`, `CitricAcid` as theoretically they should be represented in the `AcidIndex`.



Preform Log Transformations

The normality plots earlier identified the following variables could benefit from log transformations, which is even more true after having taken adjusted the negative values to zero. Variables include: `ResidualSugar`, `Chlorides`, `FreeSulfurDioxide`, `TotalSulfurDioxide`, `Density`, `PH`, `Sulphates`, and `Alcohol`.

Re-Run Normality, Outlier, & Correlation Plots

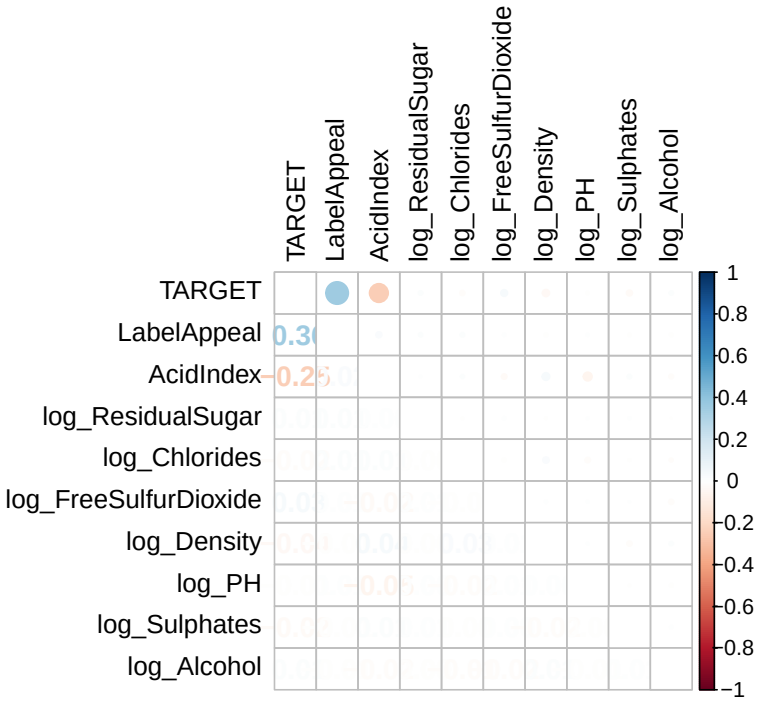
After all of the above changes, we check the dataset below. We've dropped the pre-logging variables for simplicity. In seeing the differences in mean and median within variables we still detect significantly skewed datapoints for the logged chemical variables.

variables	min	mean	median	max	zero	minus
TARGET	0.00	3.03	3.00	8.00	2,734	0
LabelAppeal	-2.00	-0.01	0.00	2.00	5,617	3,640
AcidIndex	4.00	7.77	8.00	17.00	0	0
log_ResidualSugar	-18.42	-2.93	1.36	4.95	0	3,245
log_Chlorides	-18.42	-6.38	-3.08	0.30	0	12,690
log_FreeSulfurDioxide	-18.42	-1.42	3.40	6.43	0	3,047
log_Density	-0.12	-0.01	-0.01	0.09	0	9,492
log_PH	-0.73	1.14	1.16	1.81	0	53
log_Sulphates	-18.42	-3.79	-0.69	1.44	0	10,240
log_Alcohol	-18.42	2.10	2.34	3.28	0	162

Next we check the outlier table to see if we still have great than 5% of data points identified as outliers in a large proportion of the chemical variables. It appears we have normalized some of the outliers with the above transformations, but many still have more than 5% of values identified as outliers.

variables	outliers_cnt	outliers_ratio	outliers_mean	with_mean	without_mean
log_Density	3,817	29.832	-0.007	-0.006	-0.006
log_ResidualSugar	3,142	24.556	-18.421	-2.928	2.115
log_FreeSulfurDioxide	3,047	23.814	-18.421	-1.425	3.888
log_Sulphates	2,734	21.368	-16.166	-3.790	-0.427
log_PH	2,121	16.577	0.963	1.141	1.176
log_Alcohol	1,485	11.606	-0.023	2.099	2.378
AcidIndex	1,151	8.996	10.552	7.773	7.498
TARGET	17	0.133	8.000	3.029	3.022
LabelAppeal	0	0.000		-0.009	-0.009
log_Chlorides	0	0.000		-6.382	-6.382

Creating a new correlation plot with our transformed numeric variables it appears there is still very little correlation between our `TARGET` variable and any of these chemical measures with the exception of `AcidIndex`. At this point the `LabelAppeal`, `AcidIndex`, and earlier we saw the `STARS` variable have the strongest correlations with the `TARGET`.



Build Models

Our prompt asks us to build at least two of three different models: *poisson*, *negative binomial*, and *multiple linear regression*. As there appears to be no correlation to our `TARGET` from the logged chemical variables beyond `AcidIndex`, we won't include those in any of the following models.

The Poissons

Poisson distributions are used for count data and relies on the distribution generally have a mean = variance. Poisson is particularly useful when the counts are small, if they are larger we may be able to get use out of a linear regression (which we'll try later).

First we model based on the `AcidIndex`, `STAR_new`, and `LabelAppeal_factor` variables. It appears all have significant p-values. We see a residual deviance of 13,776 which is greater than the 12,787 degrees of freedom - this suggests over-dispersion could exist so we need to check that more carefully. Finally, comparing the null deviance (a model with only the intercept) we see that we have a greatly lower residual deviance for the 8 degrees of freedom we lost in adding our predictive variables. This means the model we built fits better than the null.

```
##
## Call:
## glm(formula = TARGET ~ AcidIndex + STARS_new + LabelAppeal_factor,
##     family = "poisson", data = raw)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0499  -0.6778   0.0020   0.4831   3.7943
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  0.638253   0.039938   15.98   <2e-16 ***
## AcidIndex                   -0.080665   0.004488  -17.97   <2e-16 ***
## STARS_newone star            0.770272   0.019528   39.44   <2e-16 ***
## STARS_newtwo star            1.097896   0.018193   60.35   <2e-16 ***
## STARS_newthree star          1.222536   0.019141   63.87   <2e-16 ***
## STARS_newfour star           1.348169   0.024200   55.71   <2e-16 ***
## LabelAppeal_factorNeutral    0.213817   0.013850   15.44   <2e-16 ***
## LabelAppeal_factorPositive   0.365598   0.014892   24.55   <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: 13776  on 12787  degrees of freedom
## AIC: 45734
##
## Number of Fisher Scoring iterations: 6
```

We calculate the dispersion parameter that's based on Pearson's Chi-squared statistic and the degrees of freedom with a function from the AER library. A value over 1 indicates over-dispersion, which in this case we do not find as we have a value of 0.897. Generally a value greater than 1.10 is considering dispersed, so we extend this rational to say a value of nearly 0.9 is close enough to 1.0 to not be concerned about dispersion issues.

```
##
##  Overdispersion test
##
## data:  m_pois_1
## z = -7.8039, p-value = 1
## alternative hypothesis: true dispersion is greater than 1
## sample estimates:
## dispersion
##  0.8971707
```

When calculating the dispersion mathematically and adding it to the summary, we can see that because the dispersion is below 1 the p-values and coefficients stay the same, while the Std. Error are slightly smaller and the z value slightly larger. This further proves that our data and model is not suffering from an overdispersion problem.

```
## [1] 0.8956689
```

```
##
## Call:
## glm(formula = TARGET ~ AcidIndex + STARS_new + LabelAppeal_factor,
##     family = "poisson", data = raw)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0499  -0.6778   0.0020   0.4831   3.7943
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 0.638253   0.037797   16.89   <2e-16 ***
## AcidIndex                  -0.080665   0.004248  -18.99   <2e-16 ***
## STARS_newone star           0.770272   0.018481   41.68   <2e-16 ***
## STARS_newtwo star           1.097896   0.017218   63.76   <2e-16 ***
## STARS_newthree star         1.222536   0.018115   67.49   <2e-16 ***
## STARS_newfour star          1.348169   0.022902   58.87   <2e-16 ***
## LabelAppeal_factorNeutral   0.213817   0.013108   16.31   <2e-16 ***
## LabelAppeal_factorPositive  0.365598   0.014093   25.94   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 0.8956689)
##
##     Null deviance: 22861  on 12794  degrees of freedom
## Residual deviance: 13776  on 12787  degrees of freedom
## AIC: 45734
##
## Number of Fisher Scoring iterations: 6
```

For our second poisson model we include all variables, except for the original `STARS` rating. A dispersion check yields a 0.897 value, which is not cause for concern.

The model below shows that `log_ResidualSugar`, `log_Density`, `log_PH`, and `log_Sulphates` do not have significant p-values. Despite low values in the correlation plots earlier, it does appear that `log_Chlorides`, `log_FreeSulfurDioxide`, and `log_Alcohol` are significant at this stage.

```
##
## Call:
## glm(formula = TARGET ~ ., family = "poisson", data = raw[, c(1,
##     3, 5:13)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0788  -0.6836  -0.0079   0.4696   3.7828
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 0.6597981  0.0489485  13.479   <2e-16 ***
## AcidIndex                  -0.0806490  0.0045024 -17.912   <2e-16 ***
## LabelAppeal_factorNeutral   0.2136923  0.0138544  15.424   <2e-16 ***
## LabelAppeal_factorPositive  0.3656207  0.0148945  24.547   <2e-16 ***
## STARS_newone star           0.7692263  0.0195319  39.383   <2e-16 ***
## STARS_newtwo star           1.0963681  0.0182010  60.237   <2e-16 ***
## STARS_newthree star         1.2204417  0.0191541  63.717   <2e-16 ***
## STARS_newfour star          1.3473753  0.0242058  55.663   <2e-16 ***
## log_ResidualSugar           0.0002018  0.0005721   0.353   0.7243
## log_Chlorides              -0.0006717  0.0007230  -0.929   0.3529
## log_FreeSulfurDioxide       0.0009764  0.0005365   1.820   0.0688 .
## log_Density                -0.2346560  0.1900026  -1.235   0.2168
## log_PH                     -0.0266390  0.0216584  -1.230   0.2187
## log_Sulphates              -0.0009687  0.0007159  -1.353   0.1760
## log_Alcohol                 0.0010885  0.0025310   0.430   0.6672
## ---
## 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: 13766  on 12780  degrees of freedom
## AIC: 45738
##
## Number of Fisher Scoring iterations: 6


##
##  Overdispersion test
##
## data:  m_pois_2
## z = -7.8582, p-value = 1
## alternative hypothesis: true dispersion is greater than 1
## sample estimates:
## dispersion
##  0.8966059
```

For our third poisson model we use backwards selection on the above model to see if we can simplify the model.

```
##
## Call:
## glm(formula = TARGET ~ AcidIndex + LabelAppeal_factor + STARS_new +
```

```
##     log_FreeSulfurDioxide, family = "poisson", data = raw[, c(1,
##     3, 5:13)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0597  -0.6866  -0.0068   0.4727   3.7861
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 0.6394644  0.0399420  16.010   <2e-16 ***
## AcidIndex                  -0.0805972  0.0044885 -17.956   <2e-16 ***
## LabelAppeal_factorNeutral   0.2138056  0.0138497  15.438   <2e-16 ***
## LabelAppeal_factorPositive  0.3654140  0.0148916  24.538   <2e-16 ***
## STARS_newone star           0.7697945  0.0195298  39.416   <2e-16 ***
## STARS_newtwo star           1.0973022  0.0181961  60.304   <2e-16 ***
## STARS_newthree star         1.2220261  0.0191428  63.837   <2e-16 ***
## STARS_newfour star          1.3483797  0.0241993  55.720   <2e-16 ***
## log_FreeSulfurDioxide       0.0009672  0.0005364   1.803   0.0714 .
## ---
## 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: 13772  on 12786  degrees of freedom
## AIC: 45732
##
## Number of Fisher Scoring iterations: 6
```

For our fourth and final Poisson Model, we see if using a Zero-Inflated Poisson Model is valuable, as this can account for having a large amount of count data as zero.

```
##
## Call:
## zeroinfl(formula = TARGET ~ AcidIndex + STARS_new + LabelAppeal_factor,
##     data = raw)
##
## Pearson residuals:
##      Min       1Q   Median       3Q      Max
## -2.15290 -0.44873  0.01295  0.42117  5.87427
##
## Count model coefficients (poisson with log link):
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 0.969010   0.041609  23.289  < 2e-16 ***
## AcidIndex                  -0.020126   0.004823  -4.173 3.01e-05 ***
## STARS_newone star           0.065939   0.021188   3.112  0.00186 **
## STARS_newtwo star           0.200231   0.019764  10.131  < 2e-16 ***
## STARS_newthree star         0.303629   0.020674  14.687  < 2e-16 ***
## STARS_newfour star          0.411019   0.025532  16.098  < 2e-16 ***
## LabelAppeal_factorNeutral   0.333354   0.014552  22.907  < 2e-16 ***
## LabelAppeal_factorPositive  0.543792   0.015685  34.670  < 2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##                              Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)                 -3.90337    0.21610 -18.063   <2e-16 ***
## AcidIndex                    0.43626    0.02538  17.189   <2e-16 ***
## STARS_newone star           -2.06672    0.07556 -27.352   <2e-16 ***
## STARS_newtwo star           -5.80213    0.35751 -16.229   <2e-16 ***
## STARS_newthree star        -20.20356  349.00106  -0.058    0.954
## STARS_newfour star         -20.33076  656.00428  -0.031    0.975
## LabelAppeal_factorNeutral    0.89779    0.08254  10.877   <2e-16 ***
## LabelAppeal_factorPositive   1.65363    0.09762  16.939   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 21
## Log-likelihood: -2.051e+04 on 16 Df
```

The AIC value is used when one wants to balance goodness of fit and a penalty for model complexity. This measure is generally considered better than others when prediction is the aim of the project, which is true in our case.

Poisson Model #1 had an AIC of 4.5733713×10^4.
Poisson Model #2 had an AIC of 4.5738×10^4.
Poisson Model #3 had an AIC of 4.5732×10^4.

These are all very similar, but Poisson Model #3 that used backwards elimination on the fullest model has the best AIC score. However, Poisson Model #4 that used a zero-inflated Poisson model accounts for the large number of zero-count data in our dataset.

In order to see how Poisson Model #3 compares to Poisson Model #4 using a zero-inflated model we use the Vuong Non-Nested Hypothesis Test-Statistic. Our test statistic is significant, which means the zero-inflated model is preferable to the standard Poisson model using backwards elimination.

```
## Vuong Non-Nested Hypothesis Test-Statistic:
## (test-statistic is asymptotically distributed N(0,1) under the
##  null that the models are indistinguishible)
## -------------------------------------------------------------
##                Vuong z-statistic             H_A    p-value
## Raw                   -43.41274 model2 > model1 < 2.22e-16
## AIC-corrected         -43.28341 model2 > model1 < 2.22e-16
## BIC-corrected         -42.80124 model2 > model1 < 2.22e-16
```

The Negative Binomials

Generally the Negative Binomial is used in favor of the Poisson when the response variable is a count but the mean does not equal the variance. Further if there is overdispersion a Negative Binomial should be used instead of a Poisson. That doesn't appear to be the case with our data, but we will take a look.

Our first Negative Binomial model uses the same 3 variables as the first Poisson model, `Acid Index`, `STARS_new`, and `LabelAppeal_factor`.

The AIC of 4.5736×10^4 is extremely close to the parallel Poisson model which had an AIC of 4.5734×10^4.

```
##
## Call:
## glm.nb(formula = TARGET ~ AcidIndex + STARS_new + LabelAppeal_factor,
##     data = raw, init.theta = 39984.53511, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0498  -0.6777   0.0020   0.4831   3.7942
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 0.638275   0.039939   15.98   <2e-16 ***
## AcidIndex                  -0.080668   0.004488  -17.97   <2e-16 ***
## STARS_newone star           0.770271   0.019529   39.44   <2e-16 ***
## STARS_newtwo star           1.097895   0.018194   60.34   <2e-16 ***
## STARS_newthree star         1.222536   0.019142   63.87   <2e-16 ***
## STARS_newfour star          1.348170   0.024201   55.71   <2e-16 ***
## LabelAppeal_factorNeutral   0.213816   0.013851   15.44   <2e-16 ***
## LabelAppeal_factorPositive  0.365595   0.014892   24.55   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(39984.54) family taken to be 1)
##
##     Null deviance: 22860  on 12794  degrees of freedom
## Residual deviance: 13775  on 12787  degrees of freedom
## AIC: 45736
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  39985
##           Std. Err.:  33617
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -45718.13
```

For our second Negative Binomal model, we use all variables except the original `STARS` variable (before it was re-coded into `STARS_new`).

This is the lowest AIC we've see so far, at 4.5741×10^4‘.

```
##
## Call:
```

```
## glm.nb(formula = TARGET ~ ., data = raw[, c(1, 3, 5:13)], init.theta = 40014.64178,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0787  -0.6836  -0.0079   0.4696   3.7827
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 0.6598227  0.0489504  13.479   <2e-16 ***
## AcidIndex                  -0.0806516  0.0045026 -17.912   <2e-16 ***
## LabelAppeal_factorNeutral   0.2136912  0.0138549  15.424   <2e-16 ***
## LabelAppeal_factorPositive  0.3656175  0.0148951  24.546   <2e-16 ***
## STARS_newone star           0.7692252  0.0195323  39.382   <2e-16 ***
## STARS_newtwo star           1.0963672  0.0182014  60.235   <2e-16 ***
## STARS_newthree star         1.2204415  0.0191547  63.715   <2e-16 ***
## STARS_newfour star          1.3473762  0.0242069  55.661   <2e-16 ***
## log_ResidualSugar           0.0002018  0.0005721   0.353   0.7243
## log_Chlorides              -0.0006717  0.0007230  -0.929   0.3529
## log_FreeSulfurDioxide       0.0009764  0.0005365   1.820   0.0688 .
## log_Density                -0.2346618  0.1900114  -1.235   0.2168
## log_PH                     -0.0266416  0.0216594  -1.230   0.2187
## log_Sulphates              -0.0009687  0.0007159  -1.353   0.1760
## log_Alcohol                 0.0010884  0.0025311   0.430   0.6672
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(40014.64) family taken to be 1)
##
##     Null deviance: 22860  on 12794  degrees of freedom
## Residual deviance: 13766  on 12780  degrees of freedom
## AIC: 45741
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  40015
##           Std. Err.:  33649
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -45708.86
```

For the third Negative Binomial we perform backwards selection on the above binomial model. This results in the lowest AIC yet at 4.5735×10^4.

```
##
## Call:
## glm.nb(formula = TARGET ~ AcidIndex + LabelAppeal_factor + STARS_new +
##     log_FreeSulfurDioxide, data = raw[, c(1, 3, 5:13)], init.theta = 39992.08643,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0596  -0.6866  -0.0068   0.4727   3.7860
```

```
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  0.6394859  0.0399434  16.010   <2e-16 ***
## AcidIndex                   -0.0805997  0.0044887 -17.956   <2e-16 ***
## LabelAppeal_factorNeutral    0.2138045  0.0138501  15.437   <2e-16 ***
## LabelAppeal_factorPositive   0.3654108  0.0148922  24.537   <2e-16 ***
## STARS_newone star            0.7697934  0.0195302  39.416   <2e-16 ***
## STARS_newtwo star            1.0973013  0.0181966  60.303   <2e-16 ***
## STARS_newthree star          1.2220260  0.0191434  63.836   <2e-16 ***
## STARS_newfour star           1.3483805  0.0242004  55.717   <2e-16 ***
## log_FreeSulfurDioxide        0.0009672  0.0005364   1.803   0.0714 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(39992.09) family taken to be 1)
##
##     Null deviance: 22860  on 12794  degrees of freedom
## Residual deviance: 13772  on 12786  degrees of freedom
## AIC: 45735
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  39992
##           Std. Err.:  33621
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -45714.87
```

Negative Binomial Model #1 had an AIC of 4.5736×10^4.
Negative Binomial Model #2 had an AIC of 4.5741×10^4. Negative Binomial Model #3 had an AIC of 4.5735×10^4.

Once again, the 3rd model has the best AIC score.

The Multiple Linears

Theoretically, a linear regression is not a good choice for count data and further not a good choice when we have had some troubles with this dataset with respect to normality.

First, we try a model with the 3 variables we started with in the Poisson and Negative Binomial sections, `AcidIndex`, `STARS_new`, and `LabelAppeal_factor`. The Adjusted R-squared of 0.533 means that the model can explain 53.3% of the variance in the data.

```
##
## Call:
## lm(formula = TARGET ~ AcidIndex + STARS_new + LabelAppeal_factor,
##     data = raw)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4474 -0.9080  0.0106  0.8610  6.1847
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  2.389258   0.077577   30.80   <2e-16 ***
## AcidIndex                   -0.203199   0.008961  -22.68   <2e-16 ***
## STARS_newone star            1.375290   0.033162   41.47   <2e-16 ***
## STARS_newtwo star            2.428917   0.032141   75.57   <2e-16 ***
## STARS_newthree star          3.012819   0.037207   80.97   <2e-16 ***
## STARS_newfour star           3.722427   0.059498   62.56   <2e-16 ***
## LabelAppeal_factorNeutral    0.512209   0.028410   18.03   <2e-16 ***
## LabelAppeal_factorPositive   1.051598   0.032602   32.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.317 on 12787 degrees of freedom
## Multiple R-squared:  0.5329, Adjusted R-squared:  0.5326
## F-statistic:  2084 on 7 and 12787 DF,  p-value: < 2.2e-16
```

Second, we run a full model using all variables (except the original `STARS`) and then perform backwards selection. We obtain a very similar Adjusted R-Squared that means the model can explain 53.4% of the variance in the data. For the sake of simplicity we would consider the first Multiple Linear Regression model 'best' as it would be easier to explain the relationship between the predictive variables and the target.

```
##
## Call:
## lm(formula = TARGET ~ AcidIndex + LabelAppeal_factor + STARS_new +
##     log_Chlorides + log_FreeSulfurDioxide + log_Density + log_Sulphates,
##     data = raw[, c(1, 3, 5:13)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4988 -0.8875  0.0109  0.8616  6.1690
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  2.357352   0.078740  29.939   <2e-16 ***
## AcidIndex                   -0.202059   0.008966 -22.537   <2e-16 ***
```

```
## LabelAppeal_factorNeutral   0.510732   0.028405  17.981   <2e-16 ***
## LabelAppeal_factorPositive  1.051216   0.032597  32.249   <2e-16 ***
## STARS_newone star           1.372693   0.033160  41.396   <2e-16 ***
## STARS_newtwo star           2.425634   0.032144  75.461   <2e-16 ***
## STARS_newthree star         3.008240   0.037218  80.828   <2e-16 ***
## STARS_newfour star          3.720723   0.059483  62.551   <2e-16 ***
## log_Chlorides              -0.002362   0.001659  -1.423   0.1547
## log_FreeSulfurDioxide       0.002722   0.001220   2.232   0.0256 *
## log_Density                -0.766903   0.435995  -1.759   0.0786 .
## log_Sulphates              -0.002712   0.001657  -1.637   0.1016
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.317 on 12783 degrees of freedom
## Multiple R-squared:  0.5333, Adjusted R-squared:  0.5329
## F-statistic:  1328 on 11 and 12783 DF,  p-value: < 2.2e-16
```

Model Selection

While we cannot directly compare the Adjusted R-squared values of the Multiple Linear Regression models to the AIC values in the Poisson and Negative Binomial models, we know that linear regression models are based on assumptions of normality that are not present in our data - so we don't consider these as options for our final model. Another assumption for linear regression is for there to be no correlation between the fitted and residual values. When we looked at the fitted vs residual plots in the linear regression models, there seems to be a negative correlation between the two, strengthening our decision to not use the linear models.

We saw the Negative Binomial and Poisson models perform very similarly with regard to the AIC values, in both the 3rd model that used backwards elimination from the full value were best. However, the first model of each, that contained `STARS_new`, `AcidIndex`, and `LabelAppeal_factor` are the easiest to explain with interpretable coefficients. It makes theoretical sense that higher expert star ratings and attractive labels would increase sales, and that higher acidity in wine may be less palatable to most and decrease sales.

In looking at models 1 & 3 from the Poisson and Negative Binomials, we see very similar accuracy ratings when tested against our original dataset, all around 29%.

Further, Poisson Model #4 made use of a zero-inflated Poisson model to account for the large number of zeros we had in our count data. This model was also based on the 3 most interpretable factors (same as Poisson #1). The Vuong Non-Nested Hypothesis Test-Statistic revealed this performed better than any of the other Poisson models as well.

A quick check of the accuracy of the considered models (Poisson #1, #3, and #4 in order below) also confirms the accuracy, at least on the training dataset, is higher on the zero-inflated model at 33.5% as opposed to around 29% for the other models.

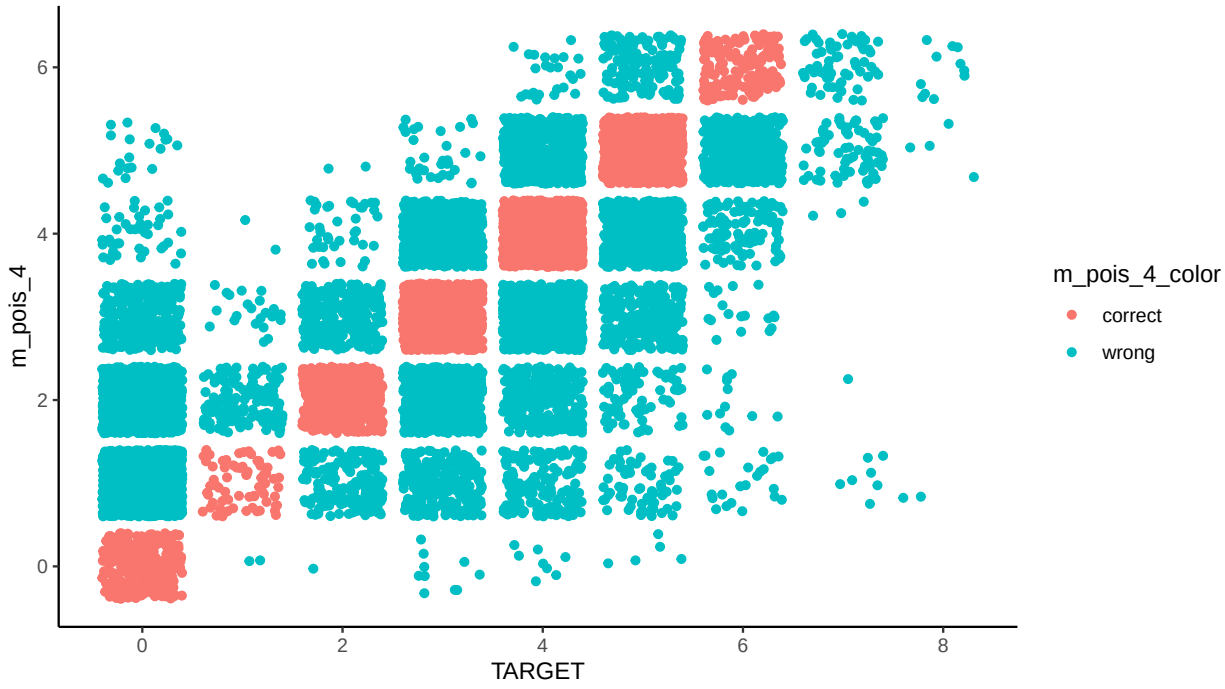
```
##  Accuracy
## 0.2900352
```

```
##  Accuracy
## 0.2898789
```

```
##  Accuracy
## 0.3349746
```

We choose to move forward with the fourth zero-inflated Poisson model, as theoretically a Poisson model is more appropriate for our dataset than a Negative Binomial and because it is easiest to interpret than the backward selection models. This model is based on the `AcidIndex`, `STAR_new`, and `LabelAppeal_factor` variables, where having a higher star rating, more positive label appeal, and less acidity increase the predicted number of cases sold after sampling. We predict that a wine distributor should experience higher case sales after choosing to sample wines that balance these qualities appropriately.

Zero-Inflated Poinsson Models prediction accuracy on the training set.



Resources Used

https://www.dataquest.io/blog/tutorial-poisson-regression-in-r/
https://cran.r-project.org/web/packages/GlmSimulatoR/vignettes/count_data_and_overdispersion.html
https://towardsdatascience.com/adjust-for-overdispersion-in-poisson-regression-4b1f52baa2f1
https://stats.idre.ucla.edu/r/dae/zip/