DATA 621: BUSINESS ANALYTICS AND DATA MINING
HOMEWORK#5 Assignment Requirements

Group 2 - Gabriel Campos, Melissa Bowman, Alexander Khaykin, & Jennifer Abinette

Last edited December 16, 2023

Contents

Overview

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

Your objective is to build a count regression model to predict the number of cases of wine that will be sold given certain properties of the wine. HINT: Sometimes, the fact that a variable is missing is actually predictive of the target. You can only use the variables given to you (or variables that you derive from the variables provided). Below is a short description of the variables of interest in the data set:

Deliverables

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

Write Up:

1. DATA EXPLORATION (25 Points)

Describe the size and the variables in the wine training data set. Consider that too much detail will cause a manager to lose interest while too little detail will make the manager consider that you aren't doing your job. Some suggestions are given below. Please do NOT treat this as a check list of things to do to complete the assignment. You should have your own thoughts on what to tell the boss. These are just ideas.

a. Mean / Standard Deviation / Median
b. Bar Chart or Box Plot of the data
c. Is the data correlated to the target variable (or to other variables?)
d. Are any of the variables missing and need to be imputed "fixed"?

2. DATA PREPARATION (25 Points)

Describe how you have transformed the data by changing the original variables or creating new variables. If you did transform the data or create new variables, discuss why you did this. Here are some possible transformations.

a. Fix missing values (maybe with a Mean or Median value)
b. Create flags to suggest if a variable was missing
c. Transform data by putting it into buckets
d. Mathematical transforms such as log or square root (or use Box-Cox)
e. Combine variables (such as ratios or adding or multiplying) to create new variables

3. BUILD MODELS (25 Points)

Using the training data set, build at least two different poisson regression models, at least two different negative binomial regression models, and at least two multiple linear regression models, using different variables (or the same variables with different transformations). Sometimes poisson and negative binomial regression models give the same results. If that is the case, comment on that. Consider changing the input variables if that occurs so that you get different models. Although not covered in class, you may also want to consider building zero-inflated poisson and negative binomial regression models. You may select the variables manually, use an approach such as Forward or Stepwise, use a different approach such as trees, or use a combination of techniques. Describe the techniques you used. If you manually selected a variable for inclusion into the model or exclusion into the model, indicate why this was done

Discuss the coefficients in the models, do they make sense? In this case, about the only thing you can comment on is the number of stars and the wine label appeal. However, you might comment on the coefficient and magnitude of variables and how they are similar or different from model to model. For example, you might say "pH seems to have a major positive impact in my poisson regression model, but a negative effect in my multiple linear regression model". Are you keeping the model even though it is counter intuitive? Why? The boss needs to know.

4. SELECT MODELS (25 Points)

Decide on the criteria for selecting the best count regression model. Will you select models with slightly worse performance if it makes more sense or is more parsimonious? Discuss why you selected your models.

For the count regression model, will you use a metric such as AIC, average squared error, etc.? Be sure to explain how you can make inferences from the model, and discuss other relevant model output. If you like the multiple linear regression model the best, please say why. However, you must select a count regression model for model deployment. Using the training data set, evaluate the performance of the count regression model. Make predictions using the evaluation data set.

1 DATA EXPLORATION

Import Data

```
df_wine_eval <-
  read.csv(paste0(url_git,"wine-evaluation-data.csv"))

head(df_wine_eval)
```

```
##   IN TARGET FixedAcidity VolatileAcidity CitricAcid ResidualSugar Chlorides
## 1  3     NA          5.4          -0.860       0.27         -10.7     0.092
## 2  9     NA         12.4           0.385      -0.76         -19.7     1.169
## 3 10     NA          7.2           1.750       0.17         -33.0     0.065
## 4 18     NA          6.2           0.100       1.80           1.0    -0.179
## 5 21     NA         11.4           0.210       0.28           1.2     0.038
## 6 30     NA         17.6           0.040      -1.15           1.4     0.535
##   FreeSulfurDioxide TotalSulfurDioxide Density   pH Sulphates Alcohol
## 1                23                398 0.98527 5.02      0.64   12.30
## 2               -37                 68 0.99048 3.37      1.09   16.00
## 3                 9                 76 1.04641 4.61      0.68    8.55
## 4               104                 89 0.98877 3.20      2.11   12.30
## 5                70                 53 1.02899 2.54     -0.07    4.80
## 6              -250                140 0.95028 3.06     -0.02   11.40
##   LabelAppeal AcidIndex STARS
## 1          -1         6    NA
## 2           0         6     2
## 3           0         8     1
## 4          -1         8     1
## 5           0        10    NA
## 6           1         8     4
```

```
df_wine_train <-
  read.csv(paste0(url_git,"wine-training-data.csv"))

head(df_wine_train)
```

```
##   INDEX TARGET FixedAcidity VolatileAcidity CitricAcid ResidualSugar Chlorides
## 1     1      3          3.2           1.160      -0.98          54.2    -0.567
## 2     2      3          4.5           0.160      -0.81          26.1    -0.425
## 3     4      5          7.1           2.640      -0.88          14.8     0.037
## 4     5      3          5.7           0.385       0.04          18.8    -0.425
## 5     6      4          8.0           0.330      -1.26           9.4        NA
## 6     7      0         11.3           0.320       0.59           2.2     0.556
##   FreeSulfurDioxide TotalSulfurDioxide Density   pH Sulphates Alcohol
## 1                NA                268 0.99280 3.33     -0.59     9.9
## 2                15               -327 1.02792 3.38      0.70      NA
## 3               214                142 0.99518 3.12      0.48    22.0
## 4                22                115 0.99640 2.24      1.83     6.2
## 5              -167                108 0.99457 3.12      1.77    13.7
## 6               -37                 15 0.99940 3.20      1.29    15.4
##   LabelAppeal AcidIndex STARS
## 1           0         8     2
## 2          -1         7     3
```

```
## 3          -1         8     3
## 4          -1         6     1
## 5           0         9     2
## 6           0        11    NA
```

1.1 Evaluation Data set

The evaluation data set contains 3,335 observations and 16 variables, although the Target variable is currently missing all values as we will predict those later once we choose a model.

1.1.1 Summary Statistics

```
dim(df_wine_eval)
```

```
## [1] 3335   16
```

```
describe(df_wine_eval)
```

```
##                    vars    n    mean      sd  median trimmed     mad     min
## IN                    1 3335 8048.31 4655.48 7906.00 8044.28 5960.05    3.00
## TARGET                2    0     NaN      NA      NA     NaN      NA     Inf
## FixedAcidity          3 3335    6.86    6.32    6.90    6.91    2.82  -18.20
## VolatileAcidity       4 3335    0.31    0.81    0.28    0.31    0.46   -2.83
## CitricAcid            5 3335    0.31    0.87    0.31    0.31    0.44   -3.12
## ResidualSugar         6 3167    5.32   34.37    3.60    5.46   16.90 -128.30
## Chlorides             7 3197    0.06    0.31    0.05    0.06    0.12   -1.15
## FreeSulfurDioxide     8 3183   34.95  149.63   30.00   34.26   57.82 -563.00
## TotalSulfurDioxide    9 3178  123.41  225.80  124.00  124.00  137.88 -769.00
## Density              10 3335    0.99    0.03    0.99    0.99    0.01    0.89
## pH                   11 3231    3.24    0.68    3.21    3.23    0.37    0.60
## Sulphates            12 3025    0.53    0.91    0.50    0.53    0.39   -3.07
## Alcohol              13 3150   10.58    3.76   10.40   10.58    2.52   -4.20
## LabelAppeal          14 3335    0.01    0.89    0.00    0.01    1.48   -2.00
## AcidIndex            15 3335    7.75    1.32    8.00    7.62    1.48    5.00
## STARS                16 2494    2.04    0.91    2.00    1.97    1.48    1.00
##                       max    range  skew kurtosis    se
## IN                16130.00 16127.00  0.01    -1.20 80.62
## TARGET                -Inf     -Inf    NA       NA    NA
## FixedAcidity         33.50    51.70 -0.12     2.04  0.11
## VolatileAcidity       3.61     6.44 -0.04     1.62  0.01
## CitricAcid            3.76     6.88 -0.03     1.66  0.02
## ResidualSugar       145.40   273.70 -0.06     1.97  0.61
## Chlorides             1.26     2.41 -0.04     1.74  0.01
## FreeSulfurDioxide   617.00  1180.00  0.07     1.88  2.65
## TotalSulfurDioxide 1004.00  1773.00 -0.05     1.50  4.01
## Density               1.10     0.21 -0.03     1.94  0.00
## pH                    6.21     5.61  0.12     1.69  0.01
## Sulphates             4.18     7.25  0.01     1.83  0.02
## Alcohol              25.60    29.80  0.05     1.54  0.07
## LabelAppeal           2.00     4.00  0.05    -0.26  0.02
```

```
## AcidIndex             17.00    12.00  1.51     4.28  0.02
## STARS                  4.00     3.00  0.44    -0.75  0.02
```

```
summary(df_wine_eval)
```

```
##        IN         TARGET         FixedAcidity     VolatileAcidity
##  Min.   :    3   Mode:logical   Min.   :-18.200   Min.   :-2.8300
##  1st Qu.: 4018   NA's:3335      1st Qu.:  5.200   1st Qu.: 0.0800
##  Median : 7906                  Median :  6.900   Median : 0.2800
##  Mean   : 8048                  Mean   :  6.864   Mean   : 0.3103
##  3rd Qu.:12061                  3rd Qu.:  9.000   3rd Qu.: 0.6300
##  Max.   :16130                  Max.   : 33.500   Max.   : 3.6100
##
##    CitricAcid      ResidualSugar        Chlorides        FreeSulfurDioxide
##  Min.   :-3.1200   Min.   :-128.300   Min.   :-1.15000   Min.   :-563.00
##  1st Qu.: 0.0000   1st Qu.:  -2.600   1st Qu.: 0.01600   1st Qu.:   3.00
##  Median : 0.3100   Median :   3.600   Median : 0.04700   Median :  30.00
##  Mean   : 0.3124   Mean   :   5.319   Mean   : 0.06143   Mean   :  34.95
##  3rd Qu.: 0.6050   3rd Qu.:  17.200   3rd Qu.: 0.17100   3rd Qu.:  79.25
##  Max.   : 3.7600   Max.   : 145.400   Max.   : 1.26300   Max.   : 617.00
##                    NA's   :168        NA's   :138        NA's   :152
##  TotalSulfurDioxide    Density             pH          Sulphates
##  Min.   :-769.00    Min.   :0.8898   Min.   :0.600   Min.   :-3.0700
##  1st Qu.:  27.25    1st Qu.:0.9883   1st Qu.:2.980   1st Qu.: 0.3300
##  Median : 124.00    Median :0.9946   Median :3.210   Median : 0.5000
##  Mean   : 123.41    Mean   :0.9947   Mean   :3.237   Mean   : 0.5346
##  3rd Qu.: 210.00    3rd Qu.:1.0005   3rd Qu.:3.490   3rd Qu.: 0.8200
##  Max.   :1004.00    Max.   :1.0998   Max.   :6.210   Max.   : 4.1800
##  NA's   :157                         NA's   :104     NA's   :310
##     Alcohol       LabelAppeal         AcidIndex          STARS
##  Min.   :-4.20   Min.   :-2.00000   Min.   : 5.000   Min.   :1.00
##  1st Qu.: 9.00   1st Qu.:-1.00000   1st Qu.: 7.000   1st Qu.:1.00
##  Median :10.40   Median : 0.00000   Median : 8.000   Median :2.00
##  Mean   :10.58   Mean   : 0.01349   Mean   : 7.748   Mean   :2.04
##  3rd Qu.:12.50   3rd Qu.: 1.00000   3rd Qu.: 8.000   3rd Qu.:3.00
##  Max.   :25.60   Max.   : 2.00000   Max.   :17.000   Max.   :4.00
##  NA's   :185                                         NA's   :841
```

```
str(df_wine_eval)
```

```
## 'data.frame':    3335 obs. of  16 variables:
##  $ IN                : int  3 9 10 18 21 30 31 37 39 47 ...
##  $ TARGET            : logi  NA NA NA NA NA NA ...
##  $ FixedAcidity      : num  5.4 12.4 7.2 6.2 11.4 17.6 15.5 15.9 11.6 3.8 ...
##  $ VolatileAcidity   : num  -0.86 0.385 1.75 0.1 0.21 0.04 0.53 1.19 0.32 0.22 ...
##  $ CitricAcid        : num  0.27 -0.76 0.17 1.8 0.28 -1.15 -0.53 1.14 0.55 0.31 ...
##  $ ResidualSugar     : num  -10.7 -19.7 -33 1 1.2 1.4 4.6 31.9 -50.9 -7.7 ...
##  $ Chlorides         : num  0.092 1.169 0.065 -0.179 0.038 ...
##  $ FreeSulfurDioxide : num  23 -37 9 104 70 -250 10 115 35 40 ...
##  $ TotalSulfurDioxide: num  398 68 76 89 53 140 17 381 83 129 ...
##  $ Density           : num  0.985 0.99 1.046 0.989 1.029 ...
##  $ pH                : num  5.02 3.37 4.61 3.2 2.54 3.06 3.07 2.99 3.32 4.72 ...
##  $ Sulphates         : num  0.64 1.09 0.68 2.11 -0.07 -0.02 0.75 0.31 2.18 -0.64 ...
```

```
##  $ Alcohol           : num  12.3 16 8.55 12.3 4.8 11.4 8.5 11.4 -0.5 10.9 ...
##  $ LabelAppeal       : int  -1 0 0 -1 0 1 0 1 0 0 ...
##  $ AcidIndex         : int  6 6 8 8 10 8 12 7 12 7 ...
##  $ STARS             : int  NA 2 1 1 NA 4 3 NA NA NA ...
```

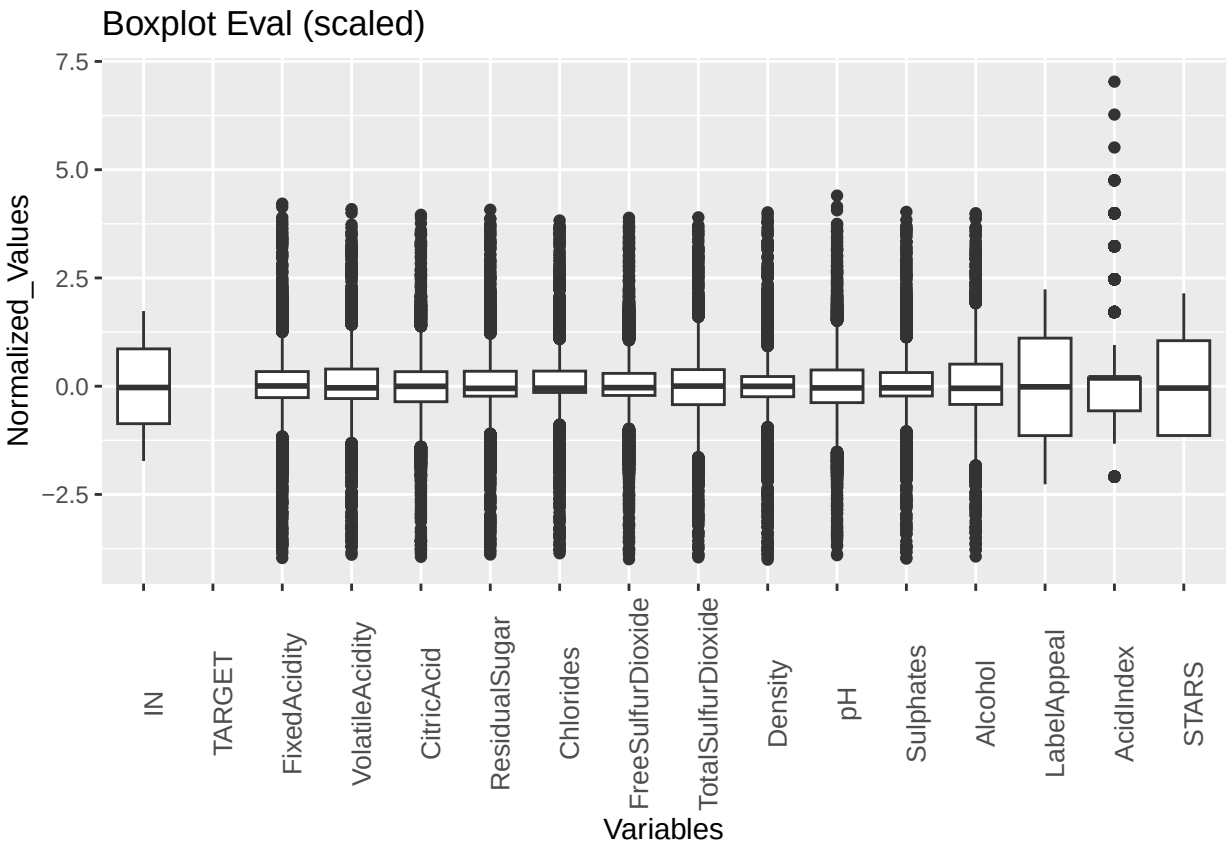
1.1.2 Missing Data

```
for (i in colnames(df_wine_eval)){
  print(paste(i,"  ", sum(is.na(df_wine_eval[,i])),sep = ""))
}
```

```
## [1] "IN  0"
## [1] "TARGET  3335"
## [1] "FixedAcidity  0"
## [1] "VolatileAcidity  0"
## [1] "CitricAcid  0"
## [1] "ResidualSugar  168"
## [1] "Chlorides  138"
## [1] "FreeSulfurDioxide  152"
## [1] "TotalSulfurDioxide  157"
## [1] "Density  0"
## [1] "pH  104"
## [1] "Sulphates  310"
## [1] "Alcohol  185"
## [1] "LabelAppeal  0"
## [1] "AcidIndex  0"
## [1] "STARS  841"
```

1.1.3 Outliers

```
df_wine_eval %>%
  scale() %>%
  as.data.frame() %>%
  stack() %>%
  ggplot(aes(x = ind, y = values)) +
  geom_boxplot() +
  labs(title = 'Boxplot Eval (scaled)',
       x = 'Variables',
       y = 'Normalized_Values')+
  theme(axis.text.x=element_text(size=10, angle=90))
```



1.2 Training Data set

The training data set contains 12,795 observations and 16 variables with our response variable (TARGET) indicating the number of wine cases purchased which ranges from 0 to 8. Our variables include information on the content of each wine such as alcohol, citric acid, sulfur dioxide, etc. as well as a wine rating by a team of experts (STARS variable). Our intent is to use this training data set to create the best fitted regression model so that we can predict the number of cases sold for the wines in the evaluation data set.

1.2.1 Summary Statistics

```
describe(df_wine_train)
```

```
##                       vars     n    mean      sd  median trimmed     mad     min
## INDEX                    1 12795 8069.98 4656.91 8110.00 8071.03 5977.84    1.00
## TARGET                   2 12795    3.03    1.93    3.00    3.05    1.48    0.00
## FixedAcidity             3 12795    7.08    6.32    6.90    7.07    3.26  -18.10
## VolatileAcidity          4 12795    0.32    0.78    0.28    0.32    0.43   -2.79
## CitricAcid               5 12795    0.31    0.86    0.31    0.31    0.42   -3.24
## ResidualSugar            6 12179    5.42   33.75    3.90    5.58   15.72 -127.80
## Chlorides                7 12157    0.05    0.32    0.05    0.05    0.13   -1.17
## FreeSulfurDioxide        8 12148   30.85  148.71   30.00   30.93   56.34 -555.00
## TotalSulfurDioxide       9 12113  120.71  231.91  123.00  120.89  134.92 -823.00
## Density                 10 12795    0.99    0.03    0.99    0.99    0.01    0.89
```

```
## pH                    11 12400    3.21    0.68    3.20    3.21    0.39    0.48
## Sulphates             12 11585    0.53    0.93    0.50    0.53    0.44   -3.13
## Alcohol               13 12142   10.49    3.73   10.40   10.50    2.37   -4.70
## LabelAppeal           14 12795   -0.01    0.89    0.00   -0.01    1.48   -2.00
## AcidIndex             15 12795    7.77    1.32    8.00    7.64    1.48    4.00
## STARS                 16  9436    2.04    0.90    2.00    1.97    1.48    1.00
##                         max    range  skew kurtosis    se
## INDEX               16129.00 16128.00  0.00    -1.20 41.17
## TARGET                  8.00     8.00 -0.33    -0.88  0.02
## FixedAcidity           34.40    52.50 -0.02     1.67  0.06
## VolatileAcidity         3.68     6.47  0.02     1.83  0.01
## CitricAcid              3.86     7.10 -0.05     1.84  0.01
## ResidualSugar         141.15   268.95 -0.05     1.88  0.31
## Chlorides               1.35     2.52  0.03     1.79  0.00
## FreeSulfurDioxide     623.00  1178.00  0.01     1.84  1.35
## TotalSulfurDioxide   1057.00  1880.00 -0.01     1.67  2.11
## Density                 1.10     0.21 -0.02     1.90  0.00
## pH                      6.13     5.65  0.04     1.65  0.01
## Sulphates               4.24     7.37  0.01     1.75  0.01
## Alcohol                26.50    31.20 -0.03     1.54  0.03
## LabelAppeal             2.00     4.00  0.01    -0.26  0.01
## AcidIndex              17.00    13.00  1.65     5.19  0.01
## STARS                   4.00     3.00  0.45    -0.69  0.01
```

```
summary(df_wine_train)
```

```
##      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
```

```
str(df_wine_train)
```

```
## 'data.frame':    12795 obs. of  16 variables:
##  $ 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 ...
```

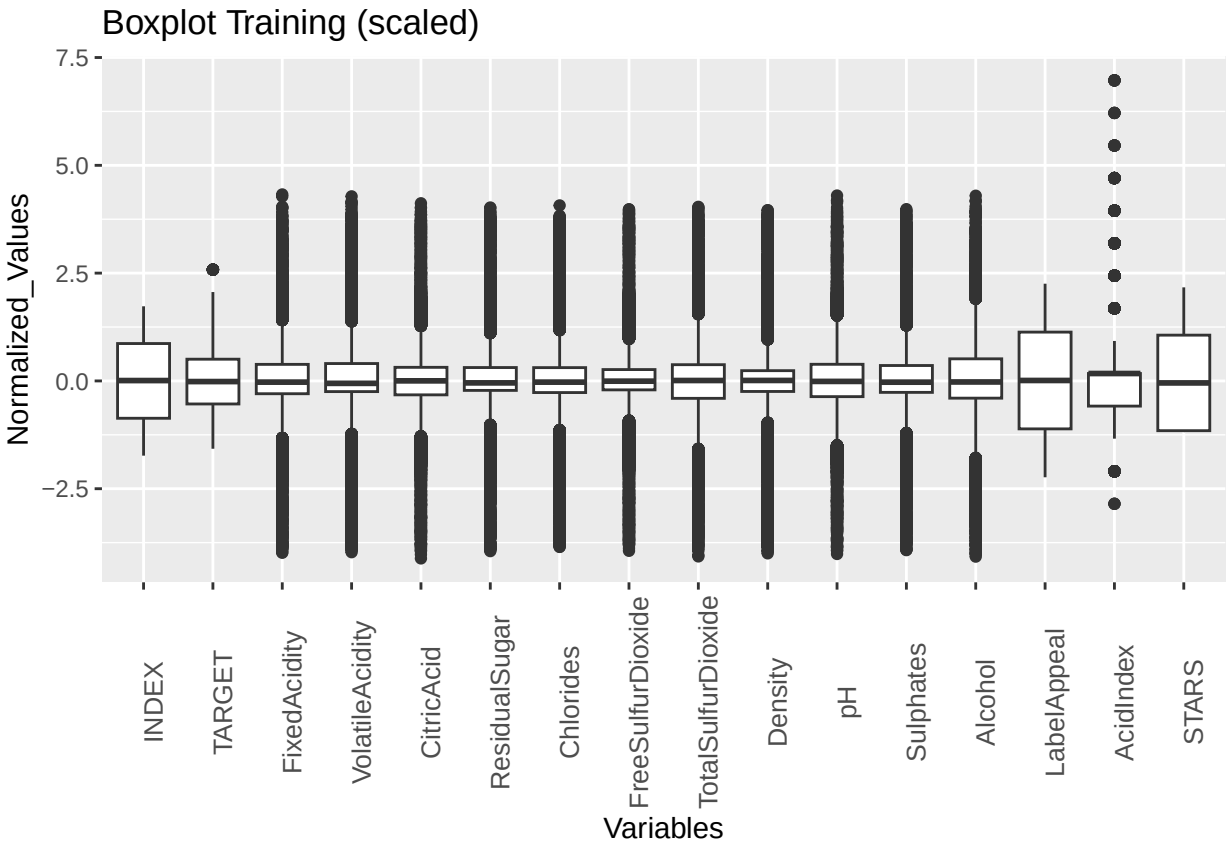
1.2.2 Missing Data

```
for (i in colnames(df_wine_train)){
  print(paste(i,"  ", sum(is.na(df_wine_train[,i])),sep = ""))
}
```

```
## [1] "INDEX  0"
## [1] "TARGET  0"
## [1] "FixedAcidity  0"
## [1] "VolatileAcidity  0"
## [1] "CitricAcid  0"
## [1] "ResidualSugar  616"
## [1] "Chlorides  638"
## [1] "FreeSulfurDioxide  647"
## [1] "TotalSulfurDioxide  682"
## [1] "Density  0"
## [1] "pH  395"
## [1] "Sulphates  1210"
## [1] "Alcohol  653"
## [1] "LabelAppeal  0"
## [1] "AcidIndex  0"
## [1] "STARS  3359"
```

1.2.3 Outliers

```
df_wine_train %>%
  scale() %>%
  as.data.frame() %>%
  stack() %>%
  ggplot(aes(x = ind, y = values)) +
  geom_boxplot() +
  labs(title = 'Boxplot Training (scaled)',
       x = 'Variables',
       y = 'Normalized_Values')+
  theme(axis.text.x=element_text(size=10, angle=90))
```



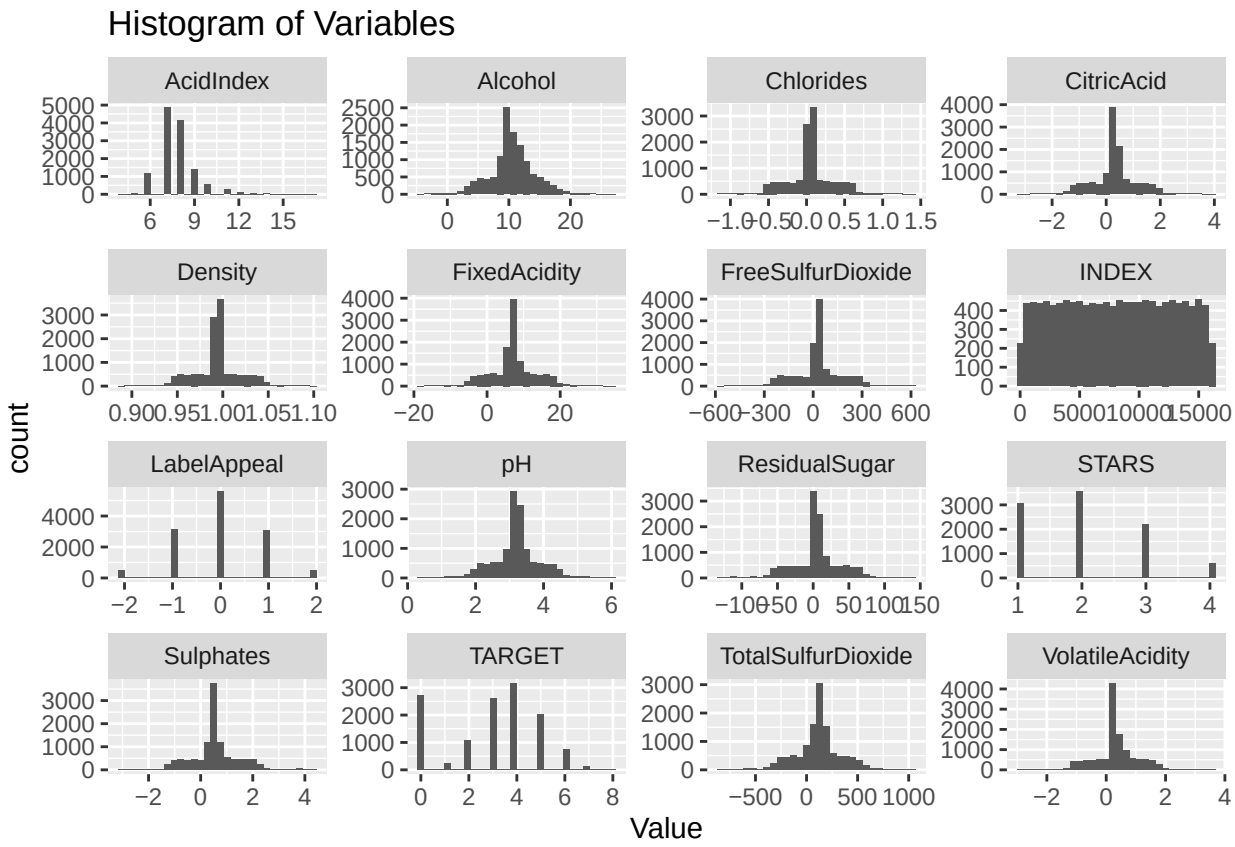
1.2.4 Variable Distributions

```
# Gather the data into a long format
data_long <- gather(df_wine_train, key = "Variable", value = "Value")

ggplot(data_long, aes(x = Value)) +
  geom_histogram() +
  facet_wrap(~Variable, scales = "free") +
  labs(title = "Histogram of Variables")
```

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

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



As shown above, the distribution of data is relatively normal so we do not have to transform any variables to improve their distribution.

1.2.5 Correlation between Variables

```
# Create a correlation matrix for all variables
(cor_matrix <- cor(df_wine_train, use='complete.obs'))
```

```
##                         INDEX        TARGET FixedAcidity VolatileAcidity
## INDEX             1.0000000000  0.0236764338 -0.002831415   -0.0008743296
## TARGET            0.0236764338  1.0000000000 -0.012538100   -0.0759978765
## FixedAcidity     -0.0028314152 -0.0125380998  1.000000000    0.0190109733
## VolatileAcidity  -0.0008743296 -0.0759978765  0.019010973    1.0000000000
## CitricAcid        0.0278869710  0.0023450490  0.014000376   -0.0234315631
## ResidualSugar     0.0208952098  0.0035195999 -0.015429391    0.0015279517
## Chlorides         0.0026827829 -0.0304301331 -0.006104447    0.0148489225
## FreeSulfurDioxide 0.0046416504  0.0226398054  0.015438463   -0.0114408079
## TotalSulfurDioxide 0.0064949038 0.0216020726 -0.023323485   -0.0007434083
## Density          -0.0034840089 -0.0475989086  0.011574241    0.0130977690
## pH               -0.0274556333  0.0002198557 -0.004553886    0.0072030364
## Sulphates        -0.0053946247 -0.0212203783  0.042229181    0.0015161001
## Alcohol          -0.0024453460  0.0737771084 -0.013085026    0.0002603082
```

```
## LabelAppeal          0.0314911460  0.4979464796  0.011375965   -0.0202419713
## AcidIndex            0.0055244862 -0.1676430648  0.154167846    0.0250529742
## STARS               -0.0057807296  0.5546857223 -0.004937345   -0.0402432388
##                        CitricAcid ResidualSugar     Chlorides FreeSulfurDioxide
## INDEX                0.0278869710   0.020895210  0.0026827829       0.004641650
## TARGET               0.0023450490   0.003519600 -0.0304301331       0.022639805
## FixedAcidity         0.0140003760  -0.015429391 -0.0061044471       0.015438463
## VolatileAcidity     -0.0234315631   0.001527952  0.0148489225      -0.011440808
## CitricAcid           1.0000000000  -0.009843146 -0.0335608661       0.012113248
## ResidualSugar       -0.0098431456   1.000000000  0.0041215692       0.021959113
## Chlorides           -0.0335608661   0.004121569  1.0000000000      -0.020492488
## FreeSulfurDioxide    0.0121132485   0.021959113 -0.0204924876       1.000000000
## TotalSulfurDioxide -0.0099174506   0.017030939  0.0004188605       0.013461673
## Density             -0.0169919691  -0.007120841  0.0206724860      -0.008663509
## pH                  -0.0007581304   0.017563769 -0.0179702278      -0.002008516
## Sulphates           -0.0144237270  -0.002705775  0.0026187777       0.026829029
## Alcohol              0.0169864284  -0.018943324 -0.0228849573      -0.023867458
## LabelAppeal          0.0153315666  -0.004579308 -0.0063870237       0.014960087
## AcidIndex            0.0545838104  -0.020301890 -0.0017134096      -0.014733717
## STARS                0.0071401699   0.019665541 -0.0063242568      -0.015390398
##                    TotalSulfurDioxide      Density            pH    Sulphates
## INDEX                  0.0064949038 -0.003484009 -0.0274556333 -0.005394625
## TARGET                 0.0216020726 -0.047598909  0.0002198557 -0.021220378
## FixedAcidity          -0.0233234848  0.011574241 -0.0045538857  0.042229181
## VolatileAcidity       -0.0007434083  0.013097769  0.0072030364  0.001516100
## CitricAcid            -0.0099174506 -0.016991969 -0.0007581304 -0.014423727
## ResidualSugar          0.0170309394 -0.007120841  0.0175637691 -0.002705775
## Chlorides              0.0004188605  0.020672486 -0.0179702278  0.002618778
## FreeSulfurDioxide      0.0134616726 -0.008663509 -0.0020085157  0.026829029
## TotalSulfurDioxide     1.0000000000  0.023167955 -0.0034227601  0.002504051
## Density                0.0231679548  1.000000000 -0.0020192285 -0.010609294
## pH                    -0.0034227601 -0.002019229  1.0000000000  0.010449255
## Sulphates              0.0025040509 -0.010609294  0.0104492547  1.000000000
## Alcohol               -0.0168515467 -0.006128355 -0.0122034469  0.010844330
## LabelAppeal           -0.0027237419 -0.018094403  0.0002181758  0.003768700
## AcidIndex             -0.0221292631  0.047778830 -0.0537128921  0.031071782
## STARS                  0.0220949002 -0.028492455 -0.0044002985 -0.023135130
##                         Alcohol   LabelAppeal    AcidIndex        STARS
## INDEX               -0.0024453460  0.0314911460  0.005524486 -0.005780730
## TARGET               0.0737771084  0.4979464796 -0.167643065  0.554685722
## FixedAcidity        -0.0130850260  0.0113759650  0.154167846 -0.004937345
## VolatileAcidity      0.0002603082 -0.0202419713  0.025052974 -0.040243239
## CitricAcid           0.0169864284  0.0153315666  0.054583810  0.007140170
## ResidualSugar       -0.0189433242 -0.0045793083 -0.020301890  0.019665541
## Chlorides           -0.0228849573 -0.0063870237 -0.001713410 -0.006324257
## FreeSulfurDioxide   -0.0238674577  0.0149600871 -0.014733717 -0.015390398
## TotalSulfurDioxide -0.0168515467 -0.0027237419 -0.022129263  0.022094900
## Density             -0.0061283546 -0.0180944026  0.047778830 -0.028492455
## pH                  -0.0122034469  0.0002181758 -0.053712892 -0.004400299
## Sulphates            0.0108443299  0.0037686996  0.031071782 -0.023135130
## Alcohol              1.0000000000 -0.0006449123 -0.055891906  0.064854486
## LabelAppeal         -0.0006449123  1.0000000000  0.010300984  0.318897022
## AcidIndex           -0.0558919056  0.0103009840  1.000000000 -0.095482582
## STARS                0.0648544864  0.3188970216 -0.095482582  1.000000000
```

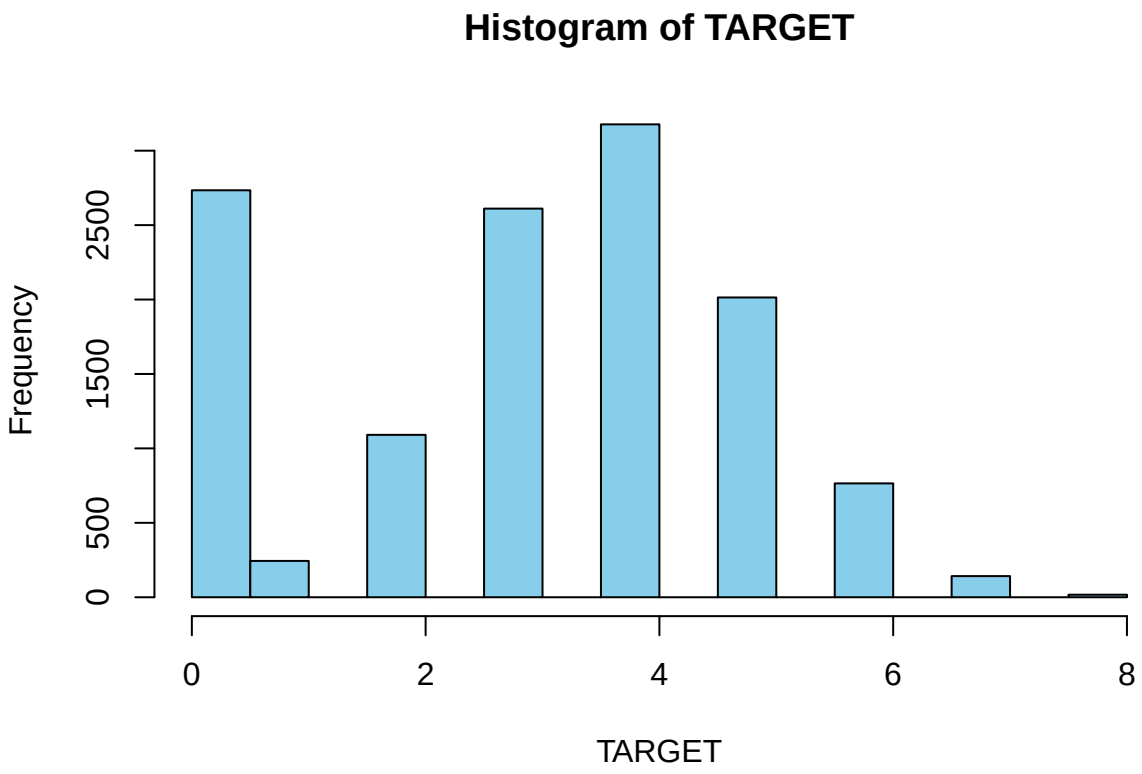
```r
cor(df_wine_train, y=df_wine_train$TARGET)
```

```
##                             [,1]
## INDEX                0.001255550
## TARGET               1.000000000
## FixedAcidity        -0.049010939
## VolatileAcidity     -0.088793212
## CitricAcid           0.008684633
## ResidualSugar                NA
## Chlorides                    NA
## FreeSulfurDioxide            NA
## TotalSulfurDioxide           NA
## Density             -0.035517502
## pH                           NA
## Sulphates                    NA
## Alcohol                      NA
## LabelAppeal          0.356500469
## AcidIndex           -0.246049449
## STARS                        NA
```

At first glance, Only two variables Label Appeal, and Acid Index have a relationship with the Target variable, number of wine cases purchased. We do see that there are several variables with missing values that we will explore transforming and then reevaluate the correlation with TARGET. Of particular interest to us is the wine rating variable (STARS) given we would predict this to be strongly correlated to the number of cases purchased. However, the STARS variable contains 3,359 missing values out of 12,795, which is about 26% of our sample so we will need to address how to best handle this.

1.2.6 Target Variable

```r
hist(df_wine_train$TARGET,main="Histogram of TARGET",xlab="TARGET",col="skyblue",border="black")
```



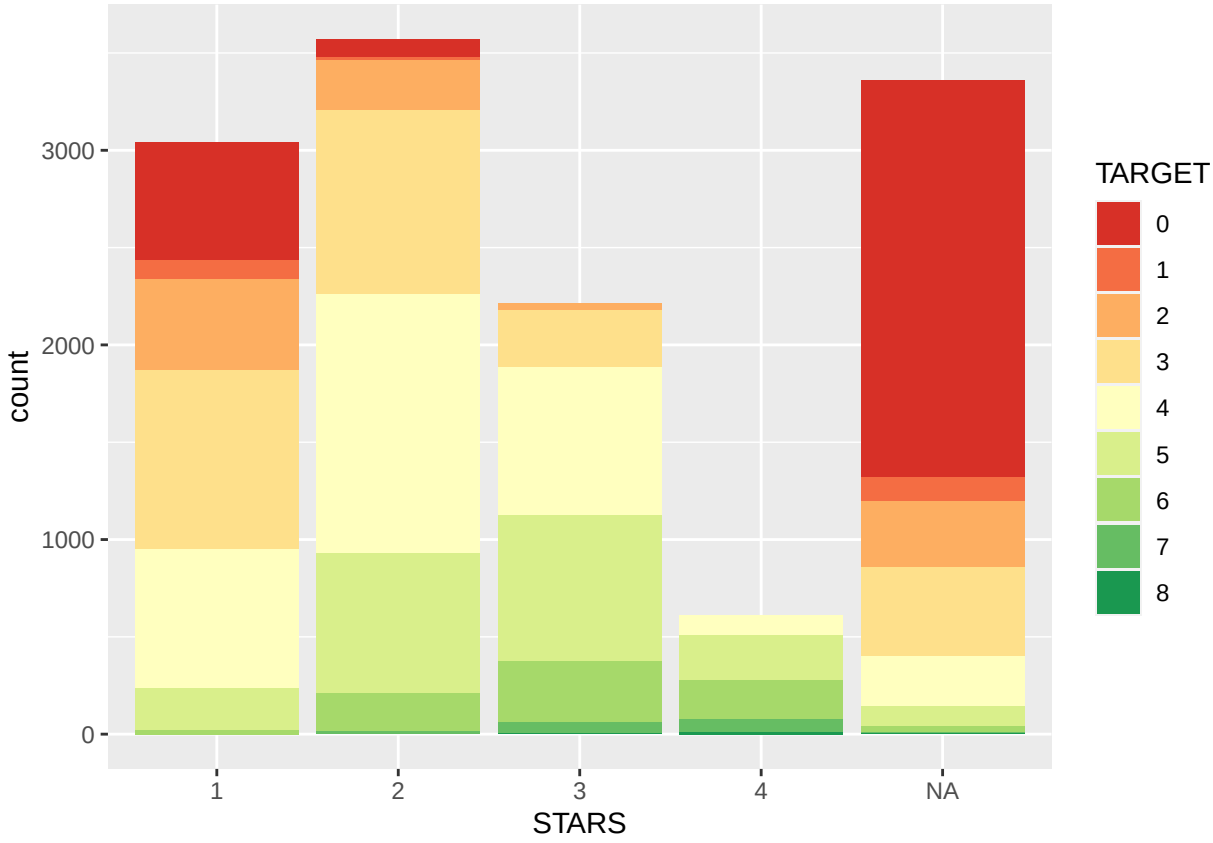

```
# Calculate the percentage of unique values in the TARGET variable
target_table <- table(df_wine_train$TARGET)
target_percentage <- prop.table(target_table) * 100


rounded_percentage <- round(target_percentage, 2)


print(rounded_percentage)
```

```
##
##     0     1     2     3     4     5     6     7     8
## 21.37  1.91  8.53 20.41 24.83 15.74  5.98  1.11  0.13
```

```
df_wine_train %>%
  mutate(STARS = as.factor(STARS),
         TARGET = as.factor(TARGET)) %>%
  ggplot(aes(STARS)) +
  geom_bar(aes(fill = TARGET)) +
  scale_fill_brewer(palette = "RdYlGn")
```



2 DATA PREPARATION

2.1 Dealing with Missing Values

As demonstrated above, STARS has a lot of NA values that relate to a TARGET value of 0 so removing all cases that have missing values in STARS would negatively impact our model. Alternatively, we can impute zero for every missing value in the STARS variable instead of eliminating 26% of our cases by dropping NA values

2.1.1 Zero Imputation for All

```
df_wine_train_zero <- df_wine_train %>%
  mutate(STARS = replace(STARS, is.na(STARS) , 0)) %>%
  mutate(ResidualSugar = replace(ResidualSugar, is.na(ResidualSugar) , 0)) %>%
  mutate(Chlorides = replace(Chlorides, is.na(Chlorides) , 0)) %>%
  mutate(FreeSulfurDioxide = replace(FreeSulfurDioxide, is.na(FreeSulfurDioxide) , 0)) %>%
  mutate(TotalSulfurDioxide = replace(TotalSulfurDioxide, is.na(TotalSulfurDioxide) , 0)) %>%
  mutate(Density = replace(Density, is.na(Density) , 0)) %>%
  mutate(pH = replace(pH, is.na(pH) , 0)) %>%
  mutate(Sulphates = replace(Sulphates, is.na(Sulphates) , 0)) %>%
  mutate(Alcohol = replace(Alcohol, is.na(Alcohol) , 0))
```

```
df_na_transformations1 <- subset(df_wine_train_zero, select=c("TARGET","STARS","ResidualSugar","Chloride
                                                              "TotalSulfurDioxide","Density","pH
cor(df_na_transformations1, y=df_na_transformations1$TARGET)
```

```
##                              [,1]
## TARGET               1.0000000000
## STARS                0.6853814727
## ResidualSugar        0.0156641597
## Chlorides           -0.0373181796
## FreeSulfurDioxide    0.0426478121
## TotalSulfurDioxide   0.0490028854
## Density             -0.0355175015
## pH                  -0.0007828372
## Sulphates           -0.0342186132
## Alcohol              0.0502073911
```

As predicted, STARS has a strong positive relationship with the number of cases purchased. Meanwhile, the other variables we used zero imputation on have weak relationships. Let's consider removing the cases with missing values for the other variables besides STARS.

2.1.2 Removing Cases with NA values from all other variables

```
df_wine_train_zero_removed <- df_wine_train %>%
  mutate(STARS = replace(STARS, is.na(STARS) , 0)) %>%
  na.omit()
```

```
df_na_transformations2 <- subset(df_wine_train_zero_removed, select=c("TARGET","STARS","ResidualSugar",
                                                              "TotalSulfurDioxide","Density","pH
cor(df_na_transformations2, y=df_na_transformations2$TARGET)
```

```
##                              [,1]
## TARGET               1.000000000
## STARS                0.678404394
## ResidualSugar        0.008329561
## Chlorides           -0.043495088
## FreeSulfurDioxide    0.038258292
## TotalSulfurDioxide   0.060656907
## Density             -0.049755657
## pH                  -0.015163491
## Sulphates           -0.038382312
## Alcohol              0.062578857
```

If we choose to remove cases with missing values (except in STARS variable) then we cut our sample down from 12,795 to 8,675. Given that the removal of cases with NA does not appear to affect the correlations with the response variable, we will instead keep all cases & use mean imputation.

2.1.3 Zero for STARS & Mean Imputation for all other variables

```r
# Get the Means of columns in Data
train_means<-sapply(df_wine_train, function(x) round(mean(x, na.rm = TRUE)))

df_wine_train_mean <- df_wine_train %>%
  mutate(STARS = replace(STARS, is.na(STARS) , 0)) %>%
# Replace other NA values in 'column_name' with 'mean'
  mutate(ResidualSugar = replace(ResidualSugar, is.na(ResidualSugar) , train_means[6])) %>%
  mutate(Chlorides = replace(Chlorides, is.na(Chlorides) , train_means[7])) %>%
  mutate(FreeSulfurDioxide = replace(FreeSulfurDioxide, is.na(FreeSulfurDioxide) , train_means[8])) %>%
  mutate(TotalSulfurDioxide = replace(TotalSulfurDioxide, is.na(TotalSulfurDioxide) , train_means[9])) %
  mutate(Density = replace(Density, is.na(Density) , train_means[10])) %>%
  mutate(pH = replace(pH, is.na(pH) , train_means[11])) %>%
  mutate(Sulphates = replace(Sulphates, is.na(Sulphates) , train_means[12])) %>%
  mutate(Alcohol = replace(Alcohol, is.na(Alcohol) , train_means[13]))
```

```r
df_na_transformations3 <- subset(df_wine_train_mean, select=c("TARGET","STARS","ResidualSugar","Chloride
                                                              "TotalSulfurDioxide","Density","pH"
cor(df_na_transformations3, y=df_na_transformations3$TARGET)
```

```
##                             [,1]
## TARGET                1.000000000
## STARS                 0.685381473
## ResidualSugar         0.016037894
## Chlorides            -0.037318180
## FreeSulfurDioxide     0.042687140
## TotalSulfurDioxide    0.050098970
## Density              -0.035517502
## pH                   -0.008733047
## Sulphates            -0.038393081
## Alcohol               0.060362045
```

Besides STARS, the different imputations saw little to no improvement for the relationship with TARGET. As it stands, cases of wine purchased appears to have meaningful relationships to STARS, Label Appeal, and Acid Index. Taking a further look at the other variables, we see many variables that have negative values where we would not expect, such as alcohol content. Let's consider transforming these variables given negative values could negatively impact our models and it does not make sense to have negative content. Additionally, we will create indicators for alcohol content and another for strong acidity (pH less than 3) as could be better predictors than the exact numeric value.

2.2 Transformations and New Variables

```r
df_wine_train_transformed <- subset(df_wine_train, select =-INDEX) %>%
  mutate(STARS = replace(STARS, is.na(STARS) , 0)) %>%
# Replace missing or negative values with zero
  mutate(FixedAcidity = replace(FixedAcidity, is.na(FixedAcidity)|FixedAcidity <0  , 0)) %>%
  mutate(VolatileAcidity = replace(VolatileAcidity, is.na(VolatileAcidity)|VolatileAcidity <0  , 0)) %>
  mutate(CitricAcid = replace(CitricAcid, is.na(CitricAcid)|CitricAcid <0  , 0)) %>%
  mutate(ResidualSugar = replace(ResidualSugar, is.na(ResidualSugar)|ResidualSugar <0 , 0)) %>%
  mutate(Chlorides = replace(Chlorides, is.na(Chlorides)|Chlorides <0  , 0)) %>%
```

```
  mutate(FreeSulfurDioxide = replace(FreeSulfurDioxide, is.na(FreeSulfurDioxide)|FreeSulfurDioxide <0
  mutate(TotalSulfurDioxide = replace(TotalSulfurDioxide, is.na(TotalSulfurDioxide)|TotalSulfurDioxide
  mutate(Density = replace(Density, is.na(Density)|Density <0  , 0)) %>%
  mutate(Sulphates = replace(Sulphates, is.na(Sulphates)|Sulphates <0 , 0)) %>%
  mutate(Alcohol = replace(Alcohol, is.na(Alcohol)|Alcohol <0 , 0)) %>%
# pH values can be negative so we will only impute zero for missing values
  mutate(pH = replace(pH, is.na(pH), 0)) %>%

# Create new variables
  mutate(Alcohol_ind = ifelse(Alcohol == 0, 0 , 1)) %>%
  mutate(pH_acidic = ifelse(pH > 0 & pH < 3, 1, 0))
```

```
cor(df_wine_train_transformed, y=df_wine_train_transformed$TARGET)
```

```
##                              [,1]
## TARGET                1.0000000000
## FixedAcidity         -0.0532985698
## VolatileAcidity      -0.0969570000
## CitricAcid            0.0133345878
## ResidualSugar         0.0124832074
## Chlorides            -0.0431178732
## FreeSulfurDioxide     0.0443925369
## TotalSulfurDioxide    0.0480410677
## Density              -0.0355175015
## pH                   -0.0007828372
## Sulphates            -0.0352841819
## Alcohol               0.0505708279
## LabelAppeal           0.3565004690
## AcidIndex            -0.2460494491
## STARS                 0.6853814727
## Alcohol_ind          -0.0009410295
## pH_acidic            -0.0024435600
```

Since there are an excess of zero values in the data set, the Poisson and Negative Binomial Regression may not be able to give the best model outcome. Therefore, we will also test Hurdle Poisson and Zero-Inflated Poisson Regression models to see if these models work best. To compare these models, we will be using the The Root Mean Squared Error (RMSE). The lowest number will tell us which model works best.

2.3 Train-test split

```
set.seed(100)
n <- nrow(df_wine_train_transformed)
train_index <- sample(1:n, 0.8 * n)  # 80% for training, 20% for testing
df_train <- df_wine_train_transformed[train_index, ]
df_test <- df_wine_train_transformed[-train_index, ]
```

3 BUILD MODELS

We will be exploring various types of regression models including Multiple Linear, Poisson, and Negative Binomial.

3.1 Multiple Linear Regression

3.1.1 Model 1 - Backward Elimination

```
MLR_model_all <- lm(TARGET ~ ., data = df_train)
MLR_step <- step(MLR_model_all, direction = "backward", test = "F")
```

```
## Start:  AIC=5669.42
## TARGET ~ FixedAcidity + VolatileAcidity + CitricAcid + ResidualSugar +
##     Chlorides + FreeSulfurDioxide + TotalSulfurDioxide + Density +
##     pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS +
##     Alcohol_ind + pH_acidic
##
##                       Df Sum of Sq   RSS     AIC   F value    Pr(>F)
## - pH_acidic            1       0.1 17752  5667.5    0.0753 0.7838174
## - FixedAcidity         1       0.2 17752  5667.5    0.1228 0.7259752
## - ResidualSugar        1       0.3 17752  5667.6    0.1541 0.6946831
## - pH                   1       0.7 17752  5667.8    0.3949 0.5297731
## <none>                             17751  5669.4
## - CitricAcid           1       5.3 17757  5670.5    3.0433 0.0811010 .
## - Sulphates            1       8.9 17760  5672.6    5.1410 0.0233876 *
## - Density              1      10.4 17762  5673.4    5.9921 0.0143870 *
## - Chlorides            1      15.0 17766  5676.1    8.6609 0.0032584 **
## - Alcohol_ind          1      17.1 17768  5677.3    9.8273 0.0017243 **
## - TotalSulfurDioxide   1      17.7 17769  5677.6   10.1685 0.0014329 **
## - Alcohol              1      22.8 17774  5680.6   13.1505 0.0002888 ***
## - FreeSulfurDioxide    1      27.3 17779  5683.1   15.6973 7.484e-05 ***
## - VolatileAcidity      1      65.2 17817  5704.9   37.5157 9.402e-10 ***
## - AcidIndex            1     693.7 18445  6059.8  399.3617 < 2.2e-16 ***
## - LabelAppeal          1    1294.2 19046  6387.8  745.0648 < 2.2e-16 ***
## - STARS                1   12354.9 30106 11074.8 7112.3870 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=5667.5
## TARGET ~ FixedAcidity + VolatileAcidity + CitricAcid + ResidualSugar +
##     Chlorides + FreeSulfurDioxide + TotalSulfurDioxide + Density +
##     pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS +
##     Alcohol_ind
##
##                       Df Sum of Sq   RSS     AIC   F value    Pr(>F)
## - FixedAcidity         1       0.2 17752  5665.6    0.1238 0.7249450
## - ResidualSugar        1       0.3 17752  5665.7    0.1564 0.6924952
## - pH                   1       1.3 17753  5666.2    0.7427 0.3888164
## <none>                             17752  5667.5
## - CitricAcid           1       5.3 17757  5668.5    3.0460 0.0809670 .
```

```
## - Sulphates           1       8.9 17760  5670.6    5.1358 0.0234579 *
## - Density             1      10.4 17762  5671.5    5.9964 0.0143521 *
## - Chlorides           1      15.1 17767  5674.2    8.6665 0.0032485 **
## - Alcohol_ind         1      17.1 17769  5675.3    9.8183 0.0017327 **
## - TotalSulfurDioxide  1      17.6 17769  5675.7   10.1562 0.0014425 **
## - Alcohol             1      22.8 17774  5678.7   13.1455 0.0002896 ***
## - FreeSulfurDioxide   1      27.2 17779  5681.2   15.6773 7.563e-05 ***
## - VolatileAcidity     1      65.2 17817  5703.0   37.5105 9.427e-10 ***
## - AcidIndex           1     693.8 18445  6058.0  399.4638 < 2.2e-16 ***
## - LabelAppeal         1    1294.2 19046  6385.8  745.0969 < 2.2e-16 ***
## - STARS               1   12356.8 30108 11073.5 7114.1413 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=5665.62
## TARGET ~ VolatileAcidity + CitricAcid + ResidualSugar + Chlorides +
##     FreeSulfurDioxide + TotalSulfurDioxide + Density + pH + Sulphates +
##     Alcohol + LabelAppeal + AcidIndex + STARS + Alcohol_ind
##
##                       Df Sum of Sq   RSS     AIC   F value    Pr(>F)
## - ResidualSugar        1       0.3 17752  5663.8    0.1533 0.6954564
## - pH                   1       1.3 17753  5664.4    0.7389 0.3900216
## <none>                              17752  5665.6
## - CitricAcid           1       5.3 17757  5666.7    3.0508 0.0807266 .
## - Sulphates            1       9.0 17761  5668.8    5.1775 0.0229012 *
## - Density              1      10.4 17762  5669.6    5.9868 0.0144300 *
## - Chlorides            1      15.0 17767  5672.3    8.6470 0.0032833 **
## - Alcohol_ind          1      17.0 17769  5673.4    9.8079 0.0017425 **
## - TotalSulfurDioxide   1      17.7 17769  5673.8   10.1776 0.0014259 **
## - Alcohol              1      22.8 17775  5676.8   13.1445 0.0002898 ***
## - FreeSulfurDioxide    1      27.2 17779  5679.3   15.6731 7.580e-05 ***
## - VolatileAcidity      1      65.2 17817  5701.1   37.5380 9.295e-10 ***
## - AcidIndex            1     720.5 18472  6070.9  414.8473 < 2.2e-16 ***
## - LabelAppeal          1    1294.9 19047  6384.3  745.5548 < 2.2e-16 ***
## - STARS                1   12357.9 30110 11071.9 7115.3874 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=5663.77
## TARGET ~ VolatileAcidity + CitricAcid + Chlorides + FreeSulfurDioxide +
##     TotalSulfurDioxide + Density + pH + Sulphates + Alcohol +
##     LabelAppeal + AcidIndex + STARS + Alcohol_ind
##
##                       Df Sum of Sq   RSS     AIC   F value    Pr(>F)
## - pH                   1       1.3 17753  5662.5    0.7386 0.3901377
## <none>                              17752  5663.8
## - CitricAcid           1       5.3 17757  5664.9    3.0744 0.0795626 .
## - Sulphates            1       9.0 17761  5666.9    5.1682 0.0230247 *
## - Density              1      10.4 17762  5667.7    5.9698 0.0145696 *
## - Chlorides            1      15.0 17767  5670.4    8.6314 0.0033116 **
## - Alcohol_ind          1      17.1 17769  5671.6    9.8523 0.0017010 **
## - TotalSulfurDioxide   1      17.6 17770  5671.9   10.1151 0.0014750 **
## - Alcohol              1      23.0 17775  5675.0   13.2203 0.0002783 ***
## - FreeSulfurDioxide    1      27.2 17779  5677.5   15.6696 7.594e-05 ***
```

```
## - VolatileAcidity     1      65.2 17817  5699.3   37.5537 9.221e-10 ***
## - AcidIndex           1     720.3 18472  6068.9  414.7488 < 2.2e-16 ***
## - LabelAppeal         1    1295.0 19047  6382.5  745.7042 < 2.2e-16 ***
## - STARS               1   12358.0 30110 11070.1 7116.0341 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=5662.51
## TARGET ~ VolatileAcidity + CitricAcid + Chlorides + FreeSulfurDioxide +
##     TotalSulfurDioxide + Density + Sulphates + Alcohol + LabelAppeal +
##     AcidIndex + STARS + Alcohol_ind
##
##                       Df Sum of Sq   RSS     AIC   F value    Pr(>F)
## <none>                             17753  5662.5
## - CitricAcid          1       5.3 17759  5663.6    3.0665 0.0799493 .
## - Sulphates           1       9.1 17762  5665.8    5.2483 0.0219881 *
## - Density             1      10.4 17764  5666.5    6.0105 0.0142374 *
## - Chlorides           1      14.9 17768  5669.1    8.6038 0.0033620 **
## - Alcohol_ind         1      17.1 17770  5670.4    9.8554 0.0016981 **
## - TotalSulfurDioxide  1      17.6 17771  5670.7   10.1344 0.0014596 **
## - Alcohol             1      23.0 17776  5673.8   13.2458 0.0002745 ***
## - FreeSulfurDioxide   1      27.3 17781  5676.2   15.6952 7.492e-05 ***
## - VolatileAcidity     1      65.6 17819  5698.3   37.7663 8.272e-10 ***
## - AcidIndex           1     719.1 18472  6066.9  414.0637 < 2.2e-16 ***
## - LabelAppeal         1    1294.6 19048  6381.0  745.4742 < 2.2e-16 ***
## - STARS               1   12360.4 30114 11069.3 7117.5762 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

After backwards elimination, our model contains 12 variables. However, we will additionally remove CitricAcid for having a p-value greater than .05 and the Alcohol Indicator as the Alcohol variable is also in the model with a lower p-value.

```
MLR_model_back <- lm(TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide +
    TotalSulfurDioxide + Density + Sulphates + Alcohol + LabelAppeal +
    AcidIndex + STARS, data = df_train)
summary(MLR_model_back)
```

```
##
## Call:
## lm(formula = TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide +
##     TotalSulfurDioxide + Density + Sulphates + Alcohol + LabelAppeal +
##     AcidIndex + STARS, data = df_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5726 -0.9456  0.0684  0.9015  6.0208
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         4.351e+00  4.937e-01   8.815  < 2e-16 ***
## VolatileAcidity    -1.438e-01  2.340e-02  -6.145 8.31e-10 ***
## Chlorides          -1.864e-01  6.271e-02  -2.973  0.00296 **
```

```
## FreeSulfurDioxide   5.253e-04  1.338e-04   3.926 8.71e-05 ***
## TotalSulfurDioxide  2.422e-04  7.728e-05   3.134  0.00173 **
## Density            -1.232e+00  4.908e-01  -2.510  0.01210 *
## Sulphates          -4.299e-02  1.894e-02  -2.270  0.02325 *
## Alcohol             6.817e-03  3.066e-03   2.223  0.02621 *
## LabelAppeal         4.155e-01  1.523e-02  27.274  < 2e-16 ***
## AcidIndex          -2.045e-01  1.008e-02 -20.292  < 2e-16 ***
## STARS               9.869e-01  1.168e-02  84.490  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.319 on 10225 degrees of freedom
## Multiple R-squared:  0.5275, Adjusted R-squared:  0.5271
## F-statistic:  1142 on 10 and 10225 DF,  p-value: < 2.2e-16
```

3.1.2 Prediction of test-split data

```
MLR_back_preds <- round( predict(MLR_model_back, newdata = df_test, type = "response"))
```

3.1.3 RMSE

```
MLR_back_rmse <- sqrt(mean((MLR_back_preds - df_test$TARGET)^2))
```

3.1.4 Model 2 - Manual Variable Selection

We noted that the backwards elimination steps demonstrated that many of the variables, even if the coefficients are significant, do not have much impact on model fit evidenced by the AIC and Adjusted R-Square values changing very little. Also, many of these variables do not intuitively have a direct relationship with the number of cases of wine purchased. In this manner, we want to explore simpler models and focus on variables with stronger correlations with our response variable.

```
MLR_model_manual <- lm(TARGET ~ Alcohol + LabelAppeal + AcidIndex + STARS, data = df_train)
summary(MLR_model_manual)
```

```
##
## Call:
## lm(formula = TARGET ~ Alcohol + LabelAppeal + AcidIndex + STARS,
##     data = df_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5327 -0.9233  0.0884  0.9097  6.0717
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.128510   0.090040  34.746   <2e-16 ***
## Alcohol      0.006630   0.003077   2.155   0.0312 *
## LabelAppeal  0.414334   0.015289  27.101   <2e-16 ***
```

```
## AcidIndex   -0.212175   0.010079 -21.052   <2e-16 ***
## STARS        0.993912   0.011697  84.970   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.324 on 10231 degrees of freedom
## Multiple R-squared:  0.5235, Adjusted R-squared:  0.5234
## F-statistic:  2811 on 4 and 10231 DF,  p-value: < 2.2e-16
```

3.1.5 Prediction of test-split data

```
MLR_manual_preds <- round( predict(MLR_model_manual, newdata = df_test, type = "response"))
```

3.1.6 RMSE

```
MLR_manual_rmse <- sqrt(mean((MLR_manual_preds - df_test$TARGET)^2))
```

3.2 Poisson Regression

3.2.1 Model 1 - Manual Selection

As noted above, many of the variables have weak relationships with our TARGET variable so we will choose to focus on those with a clear relationship including Label Appeal, Acid Index, and STARS.

```
poisson_model <- glm(TARGET ~ LabelAppeal + AcidIndex + STARS, data = df_train, family = poisson)
summary(poisson_model)
```

```
##
## Call:
## glm(formula = TARGET ~ LabelAppeal + AcidIndex + STARS, family = poisson,
##     data = df_train)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.214316   0.040775   29.78   <2e-16 ***
## LabelAppeal  0.126620   0.006776   18.69   <2e-16 ***
## AcidIndex   -0.087650   0.004987  -17.58   <2e-16 ***
## STARS        0.316755   0.005063   62.56   <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: 18089  on 10235  degrees of freedom
## Residual deviance: 11718  on 10232  degrees of freedom
## AIC: 37354
##
## Number of Fisher Scoring iterations: 5
```

```
poisson_preds <- round( predict(poisson_model, newdata = df_test, type = "response"))
```

3.2.1.1 Prediction of test-split data

```
poisson_rmse <- sqrt(mean((poisson_preds - df_test$TARGET)^2))
```

3.2.1.2 RMSE

3.2.2 Model 2 - Hurdle Poisson Regression

```
hurdle_poisson_model <- hurdle(TARGET ~ LabelAppeal + AcidIndex + STARS, data = df_train, dist = "poisso
summary(hurdle_poisson_model)
```

```
##
## Call:
## hurdle(formula = TARGET ~ LabelAppeal + AcidIndex + STARS, data = df_train,
##     dist = "poisson")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -2.1167 -0.4289 -0.0295  0.3900  5.4879
##
## Count model coefficients (truncated poisson with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.242282   0.043991  28.240  < 2e-16 ***
## LabelAppeal  0.240808   0.007319  32.902  < 2e-16 ***
## AcidIndex   -0.018532   0.005485  -3.378 0.000729 ***
## STARS        0.099243   0.005924  16.752  < 2e-16 ***
## Zero hurdle model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.58070    0.19029   13.56   <2e-16 ***
## LabelAppeal -0.49811    0.03739  -13.32   <2e-16 ***
## AcidIndex   -0.39648    0.02367  -16.75   <2e-16 ***
## STARS        2.08167    0.04860   42.83   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 10
## Log-likelihood: -1.634e+04 on 8 Df
```

```
hurdle_preds <- round( predict(hurdle_poisson_model, newdata = df_test, type = "response") )
```

3.2.2.1 Prediction of test-split data

```
hurdle_rmse <- sqrt(mean((hurdle_preds - df_test$TARGET)^2))
```

3.2.2.2 RMSE

3.2.3 Model 3 - Zero-Inflated Poisson Regression

```
zip_model <- zeroinfl(TARGET ~ LabelAppeal + AcidIndex + STARS | 1, data = df_train, dist = "poisson")
summary(zip_model)
```

```
##
## Call:
## zeroinfl(formula = TARGET ~ LabelAppeal + AcidIndex + STARS | 1, data = df_train,
##     dist = "poisson")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.6328 -0.3246  0.1745  0.4957  2.8957
##
## Count model coefficients (poisson with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.378339   0.045278   30.44   <2e-16 ***
## LabelAppeal  0.193934   0.007571   25.62   <2e-16 ***
## AcidIndex   -0.061714   0.005828  -10.59   <2e-16 ***
## STARS        0.182323   0.007230   25.22   <2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.81723    0.04322  -42.05   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 11
## Log-likelihood: -1.834e+04 on 5 Df
```

```
zip_preds <- round( predict(zip_model, newdata = df_test, type = "response") )
```

3.2.3.1 Prediction of test-split data

```
zip_rmse <- sqrt(mean((zip_preds - df_test$TARGET)^2))
```

3.2.3.2 RMSE

3.3 Negative Binomial Regression

3.3.1 Model 1 - Strongly Correlated Variables

```
neg_binom_model <- glm.nb(TARGET ~ LabelAppeal + AcidIndex + STARS, data = df_train)
```

```
## 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
```

```
summary(neg_binom_model)
```

```
##
## Call:
## glm.nb(formula = TARGET ~ LabelAppeal + AcidIndex + STARS, data = df_train,
##     init.theta = 49642.85517, link = log)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.214324   0.040777   29.78   <2e-16 ***
## LabelAppeal  0.126619   0.006776   18.69   <2e-16 ***
## AcidIndex   -0.087652   0.004987  -17.58   <2e-16 ***
## STARS        0.316759   0.005063   62.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(49642.86) family taken to be 1)
##
##     Null deviance: 18088  on 10235  degrees of freedom
## Residual deviance: 11717  on 10232  degrees of freedom
## AIC: 37356
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  49643
##           Std. Err.:  57483
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -37346.41
```

3.3.2 Prediction of test-split data

```
neg_binom_preds <- round( predict(neg_binom_model, newdata = df_test, type = "response") )
```

3.3.3 RMSE

```
neg_binom_rmse <- sqrt(mean((neg_binom_preds - df_test$TARGET)^2))
```

3.3.4 Model 2 - Include Alcohol content

The alcohol variable in previous models has increased effect size and customers may be attracted to a higher alcohol content so we will see if the addition will create a better model fit.

```
neg_binom_model_alc <- glm.nb(TARGET ~ Alcohol + LabelAppeal + AcidIndex + STARS, data = df_train)
```

```
## 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
```

```
summary(neg_binom_model_alc)
```

```
##
## Call:
## glm.nb(formula = TARGET ~ Alcohol + LabelAppeal + AcidIndex +
##     STARS, data = df_train, init.theta = 49691.46044, link = log)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.197662   0.043271  27.678   <2e-16 ***
## Alcohol      0.001537   0.001336   1.151     0.25
## LabelAppeal  0.126687   0.006776  18.697   <2e-16 ***
## AcidIndex   -0.087427   0.004990 -17.521   <2e-16 ***
## STARS        0.316463   0.005070  62.423   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(49691.46) family taken to be 1)
##
##     Null deviance: 18088  on 10235  degrees of freedom
## Residual deviance: 11716  on 10231  degrees of freedom
## AIC: 37357
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  49691
##           Std. Err.:  57546
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -37345.08
```

3.3.5 Prediction of test-split data

```
neg_binom_alc_preds <- round( predict(neg_binom_model_alc, newdata = df_test, type = "response") )
```

3.3.6 RMSE

```
neg_binom_alc_rmse <- sqrt(mean((neg_binom_alc_preds - df_test$TARGET)^2))
```

4 SELECT MODELS

4.1 Compare RMSE

```
comparison <- data.frame(
  Model = c("MLR Backward Elimination", "MLR Manual", "Poisson", "Hurdle Poisson",
            "Zero-Inflated Poisson","Negative Binomial 1", "Negative Binomial 2"),
  RMSE = c(MLR_back_rmse, MLR_manual_rmse, poisson_rmse, hurdle_rmse,
           zip_rmse, neg_binom_rmse, neg_binom_alc_rmse)
)

print(comparison)
```

```
##                       Model     RMSE
## 1 MLR Backward Elimination 1.380515
## 2               MLR Manual 1.388419
## 3                  Poisson 1.462038
## 4           Hurdle Poisson 1.355232
## 5    Zero-Inflated Poisson 1.477459
## 6      Negative Binomial 1 1.462038
## 7      Negative Binomial 2 1.466974
```

Hurdle Poisson Regression gave us the lowest RMSE, meaning it outperformed the other models in accurately predicting the response variable, although the Multiple Linear Regression models came close in second. We are selecting the Hurdle Poisson model given the RMSE, the least amount of predictors, and the intuitive sense the model makes.

4.2 Evaluation Data Set

As our selected model only contains the predictors LabelAppeal, AcidIndex, and STARS, then the only transformation needed for the evaluation data set is to replace missing values in STARS with zero.

4.2.1 Zero Imputation for Variable STARS

```
df_wine_eval_transformed  <- df_wine_eval %>%
    mutate(STARS = replace(STARS, is.na(STARS) , 0))
```

4.2.2 Predictions using the hurdle_poisson_model

```
df_wine_eval_transformed$TARGET <- round( predict(hurdle_poisson_model,
                                                  newdata = df_wine_eval_transformed, type = "response")

table(df_wine_eval_transformed$TARGET)
```

```
##
##   0   1   2   3   4   5   6   7   8
##  67 623 383 934 799 364 116  48   1
```