Data 621 Homework 5: Wine

Tommy Jenkins, Violeta Stoyanova, Todd Weigel, Peter Kowalchuk, Eleanor R-Secoquian, Anthony Pagan

12/05/2019

## OVERVIEW

In this homework assignment, we 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.

## Objective:

Our 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:

Variable Name

Definition

Theoretical Effect

INDEX

Identification Variable (do not use)

None

TARGET

Number of Cases Purchased

None

AcidIndex

Proprietary method of testing total acidity of wine by using a weighted average

Alcohol

Alcohol Content

Chlorides

Chloride content of wine

CitricAcid

Citric Acid Content

Density

Density of Wine

FixedAcidity

Fixed Acidity of Wine

FreeSulfurDioxide

Sulfur Dioxide content of wine

LabelAppeal

Marketing Score indicating the appeal of label design for consumers. High numbers suggest customers like the label design. Negative numbers suggest customes don’t like the design.

Many consumers purchase based on the visual appeal of the wine label design. Higher numbers suggest better sales

ResidualSugar

Residual Sugar of wine

STARS

Wine rating by a team of experts. 4 Stars = Excellent, 1 Star = Poor

A high number of stars suggests high sales

Sulphates

Sulfate conten of wine

TotalSulfurDioxide

Total Sulfur Dioxide of Wine

VolatileAcidity

Volatile Acid content of wine

pH

pH of wine

# DATA EXPLORATION

## Data Summary

With over 12,000 observations in our sample, we must look into the data and explore key summary statistics. We also calculate the counts for NA’s, 0, negative, and unique values.

## 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

vars

n

mean

sd

median

trimmed

mad

min

max

range

skew

kurtosis

se

na\_count

neg\_count

zero\_count

unique\_count

TARGET

1

12795

3.0290739

1.9263682

3.00000

3.0538244

1.4826000

0.00000

8.00000

8.00000

-0.3263010

-0.8772457

0.0170302

0

0

2734

9

FixedAcidity

2

12795

7.0757171

6.3176435

6.90000

7.0736739

3.2617200

-18.10000

34.40000

52.50000

-0.0225860

1.6749987

0.0558515

0

1621

548

470

VolatileAcidity

3

12795

0.3241039

0.7840142

0.28000

0.3243890

0.4299540

-2.79000

3.68000

6.47000

0.0203800

1.8322106

0.0069311

0

2827

9982

815

CitricAcid

4

12795

0.3084127

0.8620798

0.31000

0.3102520

0.4151280

-3.24000

3.86000

7.10000

-0.0503070

1.8379401

0.0076213

0

2966

9686

602

ResidualSugar

5

12179

5.4187331

33.7493790

3.90000

5.5800410

15.7155600

-127.80000

141.15000

268.95000

-0.0531229

1.8846917

0.3058158

616

NA

NA

2078

Chlorides

6

12157

0.0548225

0.3184673

0.04600

0.0540159

0.1349166

-1.17100

1.35100

2.52200

0.0304272

1.7886044

0.0028884

638

NA

NA

1664

FreeSulfurDioxide

7

12148

30.8455713

148.7145577

30.00000

30.9334877

56.3388000

-555.00000

623.00000

1178.00000

0.0063930

1.8364966

1.3492769

647

NA

NA

1000

TotalSulfurDioxide

8

12113

120.7142326

231.9132105

123.00000

120.8895367

134.9166000

-823.00000

1057.00000

1880.00000

-0.0071794

1.6746665

2.1071703

682

NA

NA

1371

Density

9

12795

0.9942027

0.0265376

0.99449

0.9942130

0.0093552

0.88809

1.09924

0.21115

-0.0186938

1.8999592

0.0002346

0

0

9492

5933

pH

10

12400

3.2076282

0.6796871

3.20000

3.2055706

0.3854760

0.48000

6.13000

5.65000

0.0442880

1.6462681

0.0061038

395

NA

NA

498

Sulphates

11

11585

0.5271118

0.9321293

0.50000

0.5271453

0.4447800

-3.13000

4.24000

7.37000

0.0059119

1.7525655

0.0086602

1210

NA

NA

631

Alcohol

12

12142

10.4892363

3.7278190

10.40000

10.5018255

2.3721600

-4.70000

26.50000

31.20000

-0.0307158

1.5394949

0.0338306

653

NA

NA

402

LabelAppeal

13

12795

-0.0090660

0.8910892

0.00000

-0.0099639

1.4826000

-2.00000

2.00000

4.00000

0.0084295

-0.2622916

0.0078777

0

3640

5617

5

AcidIndex

14

12795

7.7727237

1.3239264

8.00000

7.6431572

1.4826000

4.00000

17.00000

13.00000

1.6484959

5.1900925

0.0117043

0

0

0

14

STARS

15

9436

2.0417550

0.9025400

2.00000

1.9711258

1.4826000

1.00000

4.00000

3.00000

0.4472353

-0.6925343

0.0092912

3359

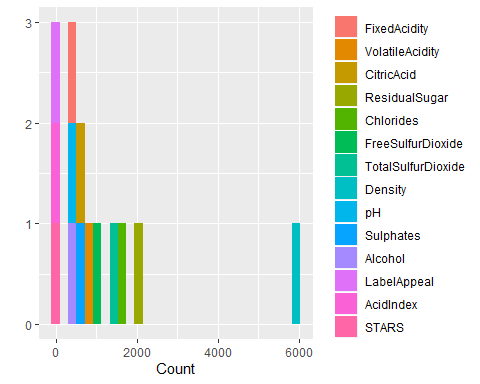
NA

NA

5

We visualize these counts per variable and then explore solutions further below.

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



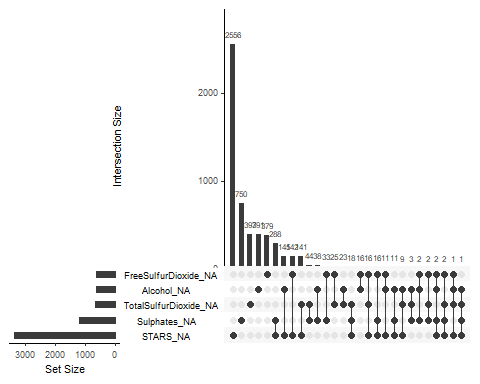
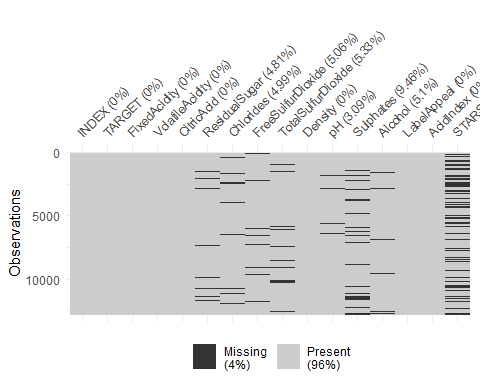
The dataset consists of two data files: training and evaluation. The training dataset contains 16 columns, and the evaluation dataset also contains 16 columns.

## Missing Data

An important aspect of any dataset is to determine how much, if any, data is missing. We look at all the variables to see which if any have missing data. We look at the basic descriptive statistics as well as the missing data and percentages.

We start by looking at the dataset as a whole and determine how many complete rows, that is rows with data for all predictors we have.

## Mode FALSE TRUE   
## logical 6359 6436



With these results, if we remove all rows with incomplete rows, there will be a total of 6436 rows out of 12795 .If we eliminate all non-complete rows and keep only rows with data for all the predictors in the dataset, our new dataset will results in 50% of the total dataset. We create a subset of data with complete cases if needed later in our analysis.

## Observations: 6,436  
## Variables: 16  
## $ INDEX <int> 4, 5, 13, 14, 16, 23, 24, 25, 26, 27, 28, 32, 34...  
## $ TARGET <int> 5, 3, 6, 0, 3, 4, 5, 4, 3, 2, 3, 4, 4, 3, 4, 3, ...  
## $ FixedAcidity <dbl> 7.1, 5.7, 5.5, -17.2, 6.0, -1.3, 10.0, 6.8, 5.8,...  
## $ VolatileAcidity <dbl> 2.640, 0.385, -0.220, 0.520, 0.330, 0.220, 0.230...  
## $ CitricAcid <dbl> -0.88, 0.04, 0.39, 0.15, -1.06, 2.95, 0.27, -0.2...  
## $ ResidualSugar <dbl> 14.80, 18.80, 1.80, -33.80, 3.00, -53.00, 14.10,...  
## $ Chlorides <dbl> 0.037, -0.425, -0.277, -0.022, 0.518, 0.541, 0.0...  
## $ FreeSulfurDioxide <dbl> 214, 22, 62, 551, 5, -85, -188, -88, 87, 15, 32,...  
## $ TotalSulfurDioxide <dbl> 142, 115, 180, 65, 378, -266, 229, 508, -283, 60...  
## $ Density <dbl> 0.99518, 0.99640, 0.94724, 0.99340, 0.96643, 0.9...  
## $ pH <dbl> 3.12, 2.24, 3.09, 4.31, 3.55, 3.61, 3.14, 3.23, ...  
## $ Sulphates <dbl> 0.48, 1.83, 0.75, 0.56, -0.86, 0.82, 0.88, 0.35,...  
## $ Alcohol <dbl> 22.0, 6.2, 12.6, 13.1, 3.9, 10.0, 11.0, 18.3, 11...  
## $ LabelAppeal <int> -1, -1, 0, 1, 1, 0, 1, -1, -1, -1, 0, 0, 1, -1, ...  
## $ AcidIndex <int> 8, 6, 8, 5, 7, 8, 11, 8, 6, 7, 8, 7, 7, 8, 6, 9,...  
## $ STARS <int> 3, 1, 4, 1, 2, 3, 2, 2, 1, 1, 1, 2, 2, 1, 3, 1, ...

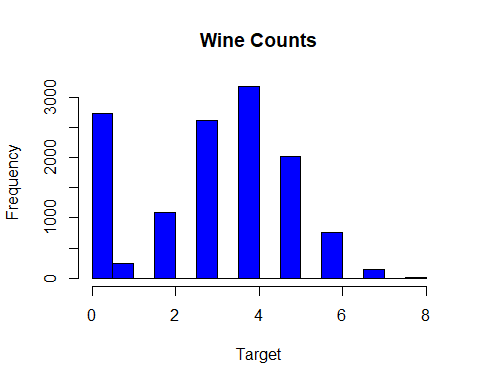
## Visualization

We consider each variable

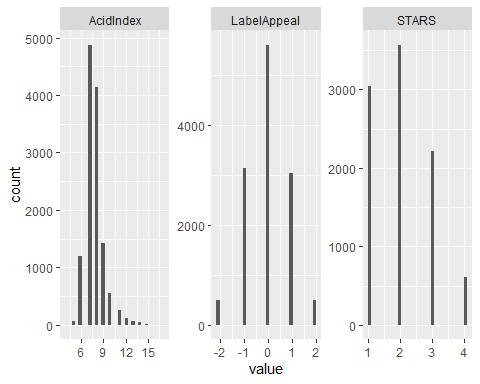
### Target Variable

The distribution of our target variable is normal with the exception of many 0 Wine count entries. At such a high percentage, the zero scores likely reflect lack of popularity rather than error, especially if they get low human ratings.

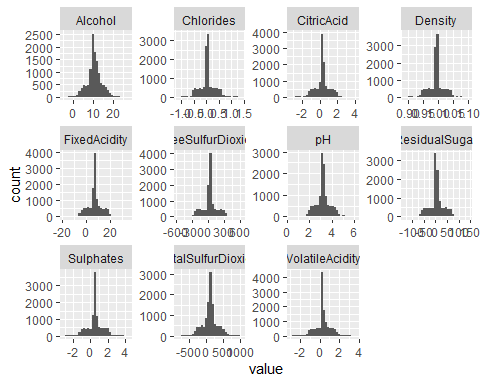
#### Histogram



#### Integers

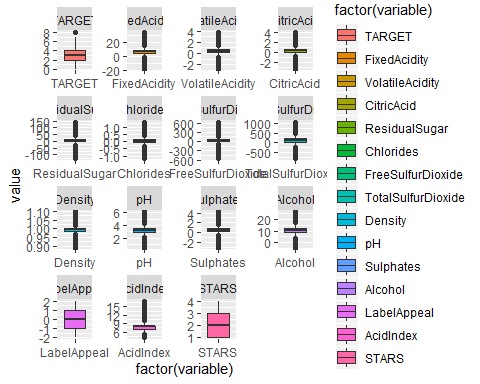
The integer variables have a small range and look normal, similar to TARGET. Stars has the least number of values and has many 0 entries. We will treat these as meaningful due to the percentage of NA’s. Decision makers who buy wine are similar to the population who creates the integer variables and the range of values is small, so we choose not to impute these. 

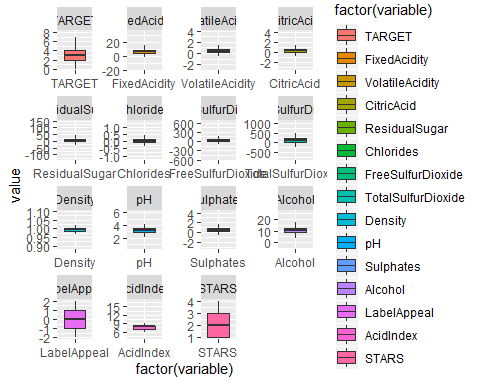
#### Doubles

The Double variable types look very similar to one another, and look somewhat normal. These look okay to impute after we’ve run our diagnostic plots. 

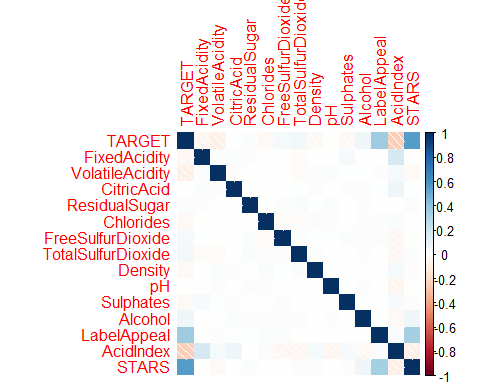
### Outliers

#### Boxplot

 #### Boxplot Without outliers



#### Correlation

We note that the human ratings all have high correlations than do our chemical features. 

# DATA PREPARATION

## na\_count neg\_count zero\_count unique\_count  
## TARGET 0 0 2734 9  
## FixedAcidity 0 1621 548 470  
## VolatileAcidity 0 2827 9982 815  
## CitricAcid 0 2966 9686 602  
## ResidualSugar 616 NA NA 2078  
## Chlorides 638 NA NA 1664  
## FreeSulfurDioxide 647 NA NA 1000  
## TotalSulfurDioxide 682 NA NA 1371  
## Density 0 0 9492 5933  
## pH 395 NA NA 498  
## Sulphates 1210 NA NA 631  
## Alcohol 653 NA NA 402  
## LabelAppeal 0 3640 5617 5  
## AcidIndex 0 0 0 14  
## STARS 3359 NA NA 5

We recall that STARS has a high correlation with TARGET and we see that it has 26.2524424% NA’s and no zero’s. We change NA to 0.

## na\_count neg\_count zero\_count unique\_count  
## STARS 3359 NA NA 5  
## Sulphates 1210 NA NA 631  
## TotalSulfurDioxide 682 NA NA 1371  
## Alcohol 653 NA NA 402  
## FreeSulfurDioxide 647 NA NA 1000  
## Chlorides 638 NA NA 1664  
## ResidualSugar 616 NA NA 2078  
## pH 395 NA NA 498  
## TARGET 0 0 2734 9  
## FixedAcidity 0 1621 548 470  
## VolatileAcidity 0 2827 9982 815  
## CitricAcid 0 2966 9686 602  
## Density 0 0 9492 5933  
## LabelAppeal 0 3640 5617 5  
## AcidIndex 0 0 0 14

The remaining NA counts include continuous variables which we will impute.

We can normalize the negative counts, through BoxCox, since one of them has some correlation and is based on human ratings. While the negative ratings make the data irregular to work with, it is unlikely that so many people (NA% )did not mean to give a negative rating. We can do this to the continuous variables as well rather than delete them, since they also have high counts.

## na\_count neg\_count zero\_count unique\_count  
## LabelAppeal 0 3640 5617 5  
## CitricAcid 0 2966 9686 602  
## VolatileAcidity 0 2827 9982 815  
## FixedAcidity 0 1621 548 470  
## TARGET 0 0 2734 9  
## Density 0 0 9492 5933  
## AcidIndex 0 0 0 14  
## ResidualSugar 616 NA NA 2078  
## Chlorides 638 NA NA 1664  
## FreeSulfurDioxide 647 NA NA 1000  
## TotalSulfurDioxide 682 NA NA 1371  
## pH 395 NA NA 498  
## Sulphates 1210 NA NA 631  
## Alcohol 653 NA NA 402  
## STARS 3359 NA NA 5

By the same logic we will leave the zero counts alone until normalization. We can exclude the TARGET variable unless we will be normalizing it specifically for our later work.

## na\_count neg\_count zero\_count unique\_count  
## VolatileAcidity 0 2827 9982 815  
## CitricAcid 0 2966 9686 602  
## Density 0 0 9492 5933  
## LabelAppeal 0 3640 5617 5  
## TARGET 0 0 2734 9  
## FixedAcidity 0 1621 548 470  
## AcidIndex 0 0 0 14  
## ResidualSugar 616 NA NA 2078  
## Chlorides 638 NA NA 1664  
## FreeSulfurDioxide 647 NA NA 1000  
## TotalSulfurDioxide 682 NA NA 1371  
## pH 395 NA NA 498  
## Sulphates 1210 NA NA 631  
## Alcohol 653 NA NA 402  
## STARS 3359 NA NA 5

We want to take a look at the least unique counts next, and by a large margin LabelAppeal, STARS, and AcidIndex show low unique counts. We can exclude TARGET until we analyze our feature transformation decisions. We see that AcidIndex is a proprietary weighted method for measuring Acid. We do not need to impute this, given its complexity and relative influence, and the risk or changing our results our very small ranges.

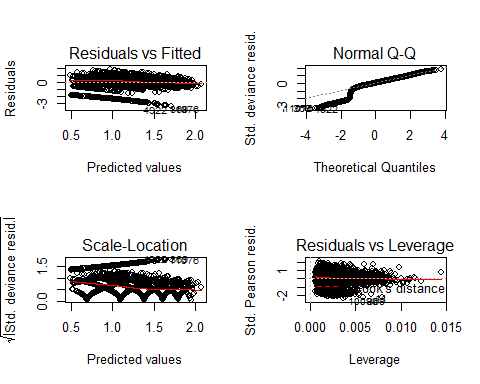
## na\_count neg\_count zero\_count unique\_count  
## LabelAppeal 0 3640 5617 5  
## STARS 3359 NA NA 5  
## TARGET 0 0 2734 9  
## AcidIndex 0 0 0 14  
## Alcohol 653 NA NA 402  
## FixedAcidity 0 1621 548 470  
## pH 395 NA NA 498  
## CitricAcid 0 2966 9686 602  
## Sulphates 1210 NA NA 631  
## VolatileAcidity 0 2827 9982 815  
## FreeSulfurDioxide 647 NA NA 1000  
## TotalSulfurDioxide 682 NA NA 1371  
## Chlorides 638 NA NA 1664  
## ResidualSugar 616 NA NA 2078  
## Density 0 0 9492 5933

# BUILD MODEL

## Model 1: Poisson Regression (all predictors)

For the first model, we used the Poisson regression and all of the predictors.

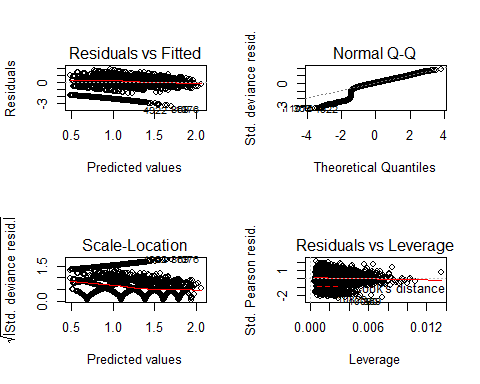
##   
## Call:  
## glm(formula = TARGET ~ . - INDEX, family = poisson, data = WineTrain)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.3301 -0.2799 0.0536 0.3861 1.7744   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.688e+00 2.503e-01 6.742 1.56e-11 \*\*\*  
## FixedAcidity 4.868e-04 1.053e-03 0.462 0.6438   
## VolatileAcidity -2.481e-02 8.362e-03 -2.968 0.0030 \*\*   
## CitricAcid -1.614e-03 7.578e-03 -0.213 0.8313   
## ResidualSugar -6.875e-05 1.939e-04 -0.355 0.7229   
## Chlorides -3.327e-02 2.053e-02 -1.621 0.1050   
## FreeSulfurDioxide 6.032e-05 4.399e-05 1.371 0.1703   
## TotalSulfurDioxide 2.141e-05 2.854e-05 0.750 0.4531   
## Density -3.695e-01 2.462e-01 -1.501 0.1334   
## pH -2.547e-03 9.608e-03 -0.265 0.7909   
## Sulphates -6.672e-03 7.051e-03 -0.946 0.3440   
## Alcohol 4.443e-03 1.772e-03 2.507 0.0122 \*   
## LabelAppeal 1.783e-01 7.958e-03 22.403 < 2e-16 \*\*\*  
## AcidIndex -4.762e-02 5.911e-03 -8.056 7.88e-16 \*\*\*  
## STARS2 3.232e-01 1.740e-02 18.574 < 2e-16 \*\*\*  
## STARS3 4.387e-01 1.896e-02 23.134 < 2e-16 \*\*\*  
## STARS4 5.509e-01 2.688e-02 20.491 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 5844.1 on 6435 degrees of freedom  
## Residual deviance: 3928.8 on 6419 degrees of freedom  
## (6359 observations deleted due to missingness)  
## AIC: 23095  
##   
## Number of Fisher Scoring iterations: 5



## Model 2: Poisson Regression (reduced predictors)

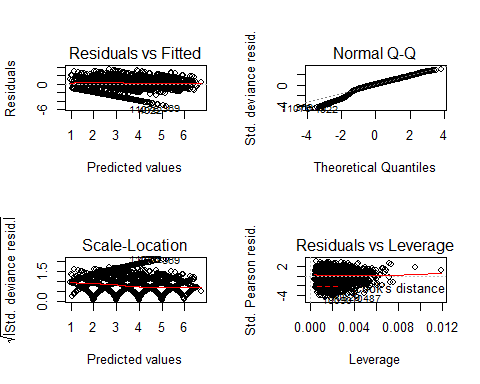
For the second model, based on model 1, we reduced the number of predictors.

##   
## Call:  
## glm(formula = TARGET ~ VolatileAcidity + CitricAcid + Chlorides +   
## FreeSulfurDioxide + TotalSulfurDioxide + Density + pH + Sulphates +   
## Alcohol + LabelAppeal + AcidIndex + STARS, family = poisson,   
## data = WineTrain)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.3170 -0.2829 0.0543 0.3858 1.7731   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.669e+00 2.442e-01 6.832 8.37e-12 \*\*\*  
## VolatileAcidity -2.319e-02 8.142e-03 -2.849 0.00439 \*\*   
## CitricAcid -4.347e-04 7.395e-03 -0.059 0.95313   
## Chlorides -2.918e-02 2.000e-02 -1.459 0.14447   
## FreeSulfurDioxide 7.263e-05 4.282e-05 1.696 0.08987 .   
## TotalSulfurDioxide 2.206e-05 2.785e-05 0.792 0.42828   
## Density -3.417e-01 2.402e-01 -1.422 0.15491   
## pH -3.949e-03 9.411e-03 -0.420 0.67478   
## Sulphates -6.911e-03 6.869e-03 -1.006 0.31437   
## Alcohol 4.113e-03 1.730e-03 2.377 0.01743 \*   
## LabelAppeal 1.766e-01 7.744e-03 22.801 < 2e-16 \*\*\*  
## AcidIndex -4.784e-02 5.720e-03 -8.363 < 2e-16 \*\*\*  
## STARS2 3.283e-01 1.700e-02 19.307 < 2e-16 \*\*\*  
## STARS3 4.443e-01 1.852e-02 23.996 < 2e-16 \*\*\*  
## STARS4 5.556e-01 2.611e-02 21.280 < 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: 6145.7 on 6746 degrees of freedom  
## Residual deviance: 4123.6 on 6732 degrees of freedom  
## (6048 observations deleted due to missingness)  
## AIC: 24216  
##   
## Number of Fisher Scoring iterations: 5



## Model 3: Gaussian Regression (significant predictors)

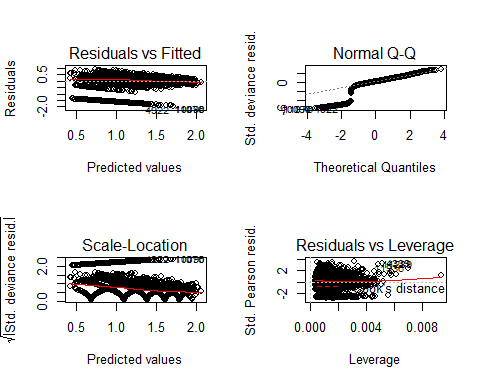
##   
## Call:  
## glm(formula = TARGET ~ VolatileAcidity + FreeSulfurDioxide +   
## TotalSulfurDioxide + Chlorides + Density + pH + Sulphates +   
## LabelAppeal + AcidIndex + STARS, family = gaussian, data = WineTrain)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -5.1949 -0.5359 0.0996 0.7405 3.3160   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.331e+00 5.229e-01 10.194 < 2e-16 \*\*\*  
## VolatileAcidity -8.491e-02 1.740e-02 -4.881 1.08e-06 \*\*\*  
## FreeSulfurDioxide 2.752e-04 9.243e-05 2.978 0.00291 \*\*   
## TotalSulfurDioxide 7.090e-05 5.952e-05 1.191 0.23358   
## Chlorides -1.158e-01 4.293e-02 -2.697 0.00701 \*\*   
## Density -1.300e+00 5.169e-01 -2.516 0.01191 \*   
## pH -1.875e-03 2.020e-02 -0.093 0.92607   
## Sulphates -2.413e-02 1.471e-02 -1.641 0.10092   
## LabelAppeal 6.411e-01 1.649e-02 38.878 < 2e-16 \*\*\*  
## AcidIndex -1.611e-01 1.153e-02 -13.966 < 2e-16 \*\*\*  
## STARS2 1.005e+00 3.313e-02 30.326 < 2e-16 \*\*\*  
## STARS3 1.513e+00 3.839e-02 39.411 < 2e-16 \*\*\*  
## STARS4 2.151e+00 6.199e-02 34.701 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 1.319364)  
##   
## Null deviance: 17036.4 on 7102 degrees of freedom  
## Residual deviance: 9354.3 on 7090 degrees of freedom  
## (5692 observations deleted due to missingness)  
## AIC: 22141  
##   
## Number of Fisher Scoring iterations: 2



Model 3 shows a better Q-Q plot than the previous two models.

## Model 4: Negative Binomial Regression

##   
## Call:  
## glm(formula = TARGET ~ VolatileAcidity + TotalSulfurDioxide +   
## pH + Sulphates + LabelAppeal + AcidIndex + STARS, family = negative.binomial(1),   
## data = WineTrain)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.93848 -0.13164 0.02425 0.17731 0.87362   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.443e+00 3.642e-02 39.613 < 2e-16 \*\*\*  
## VolatileAcidity -2.766e-02 5.397e-03 -5.125 3.04e-07 \*\*\*  
## TotalSulfurDioxide 2.430e-05 1.823e-05 1.333 0.1827   
## pH -3.613e-03 6.190e-03 -0.584 0.5595   
## Sulphates -8.920e-03 4.527e-03 -1.970 0.0488 \*   
## LabelAppeal 1.862e-01 5.081e-03 36.649 < 2e-16 \*\*\*  
## AcidIndex -5.607e-02 3.622e-03 -15.480 < 2e-16 \*\*\*  
## STARS2 3.232e-01 1.037e-02 31.160 < 2e-16 \*\*\*  
## STARS3 4.375e-01 1.185e-02 36.914 < 2e-16 \*\*\*  
## STARS4 5.464e-01 1.839e-02 29.716 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for Negative Binomial(1) family taken to be 0.1073286)  
##   
## Null deviance: 2393.6 on 7878 degrees of freedom  
## Residual deviance: 1870.2 on 7869 degrees of freedom  
## (4916 observations deleted due to missingness)  
## AIC: 37774  
##   
## Number of Fisher Scoring iterations: 5



# SELECT MODEL

## Pick the best regression model

Model 1

Model 2

Model 3

Model 4

AIC

23095.3698167386

24216.2765273715

22141.0249335269

37774.0980469856

BIC

23210.4540793765

24318.5293259809

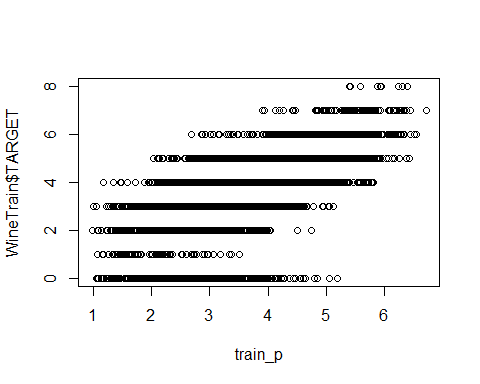
22237.1807486529

37843.8176096981

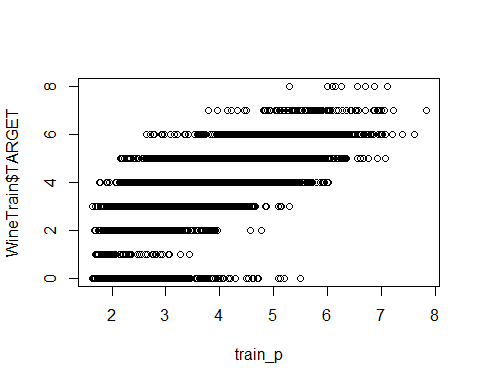
With 4 models computed, we select the model with the lowest combination of AIC and BIC. From the table, we can see the model to pick is model 3

#CONCLUSION

Model 3 showed the best result. We can observe its performance by plotting the datasets TARGET values agaisnt the predicted values. One thing we observe is that the model doesn’t predict a TARGET of 8.



Other models, although of worse performace according to our selection metric, do show results of TARGET 8, but as can be seen in the graph below, they do not corresponde to real TARGET 8 classifications.



# APPENDIX

**Code used in analysis**

knitr::opts\_chunk$set(echo = FALSE, warning = FALSE)  
require(knitr)  
library(MASS)  
library(psych)  
library(kableExtra)  
library(tidyverse)  
library(faraway)  
library(gridExtra)  
library(reshape2)  
library(leaps)  
library(caret)  
library(naniar)  
library(pander)  
library(pROC)  
library(corrplot)  
#library(jtools)  
WineTrain <- read.csv("https://raw.githubusercontent.com/pkowalchuk/CUNY621-HW5/master/wine-training-data.csv",fileEncoding="UTF-8-BOM",na.strings="",header=TRUE)  
WineTrain1 <- WineTrain  
WineEval <- read.csv("https://raw.githubusercontent.com/pkowalchuk/CUNY621-HW5/master/wine-evaluation-data.csv",fileEncoding="UTF-8-BOM",na.strings="",header=TRUE)  
vr <- c("INDEX", "TARGET", "AcidIndex", "Alcohol", "Chlorides", "CitricAcid", "Density", "FixedAcidity", "FreeSulfurDioxide", "LabelAppeal", "ResidualSugar", "STARS", "Sulphates", "TotalSulfurDioxide", "VolatileAcidity", "pH")  
def <- c("Identification Variable (do not use)", "Number of Cases Purchased", "Proprietary method of testing total acidity of wine by using a weighted average", "Alcohol Content", "Chloride content of wine", "Citric Acid Content", "Density of Wine", "Fixed Acidity of Wine", "Sulfur Dioxide content of wine", "Marketing Score indicating the appeal of label design for consumers. High numbers suggest customers like the label design. Negative numbers suggest customes don't like the design.", "Residual Sugar of wine", "Wine rating by a team of experts. 4 Stars = Excellent, 1 Star = Poor", "Sulfate conten of wine", "Total Sulfur Dioxide of Wine", "Volatile Acid content of wine", "pH of wine")  
te <- c("None", "None", "", "", "", "", "", "", "", "Many consumers purchase based on the visual appeal of the wine label design. Higher numbers suggest better sales", "", "A high number of stars suggests high sales", "", "", "", "")  
kable(cbind(vr, def, te), col.names = c("Variable Name", "Definition", "Theoretical Effect")) %>%   
 kable\_styling()  
#glimpse(WineTrain)  
#colnames(WineTrain[-1])<-"INDEX"  
summary(WineTrain)  
  
var\_stats<- function(WineTrain){  
 wt <- WineTrain[-1]  
 wine1 <- describe(wt)  
 #wine1$na\_count <- sapply(WineTrain[-1], function(y) sum(length(which(is.na(y)))))  
 wine1$na\_count <- sapply(wt, function(y) sum(is.na(y)))  
 wine1$neg\_count <- sapply(wt, function(y) sum(y<0))  
 wine1$zero\_count <- sapply(wt, function(y) sum(as.integer(y)==0))  
 wine1$unique\_count <- sapply(wt, function(y) sum(n\_distinct(y)))  
   
 return(wine1)  
}  
wine1 <- var\_stats(WineTrain)  
  
kable(wine1, "html", escape = F) %>%  
 kable\_styling("striped", full\_width = T) %>%  
 column\_spec(1, bold = T) %>%  
 scroll\_box(width = "100%", height = "700px")  
  
# geom\_point(data = wine1, color = "red", shape = 15, size = 5)  
  
qplot(wine1[c(-1),]$unique\_count,   
 geom="auto", xlab = "Count",  
 fill=as\_factor(rownames(wine1[c(-1),])))+  
 theme(legend.title=element\_blank())  
  
colsTrain<-ncol(WineTrain)  
colsEval<-ncol(WineEval)  
missingCol<-colnames(WineTrain)[!(colnames(WineTrain) %in% colnames(WineEval))]  
cc<-summary(complete.cases(WineTrain))  
cWineTrain<-subset(WineTrain, complete.cases(WineTrain))  
cc  
vis\_miss(WineTrain)  
gg\_miss\_upset(WineTrain)  
glimpse(cWineTrain)  
WineTrain1$INDEX <- NULL  
hist(WineTrain1$TARGET, col = "blue", xlab = " Target ", main = "Wine Counts")  
WineTrain1[-1] %>%  
 keep(is.integer) %>%   
 gather() %>%   
 ggplot(aes(value), main="") +  
 facet\_wrap(~ key, scales = "free") +  
 geom\_histogram()  
WineTrain1 %>%  
 keep(is.double) %>%   
 gather() %>%   
 ggplot(aes(value)) +  
 facet\_wrap(~ key, scales = "free") +  
 geom\_histogram()  
ggplot(melt(WineTrain[-1]), aes(x=factor(variable), y=value, fill=factor(variable))) + facet\_wrap(~variable, scale="free") + geom\_boxplot()  
ggplot(melt(WineTrain[-1]), aes(x=factor(variable), y=value, fill=factor(variable))) + facet\_wrap(~variable, scale="free") + geom\_boxplot(outlier.shape=NA)  
corrplot(as.matrix(cor(WineTrain[-1], use = "pairwise.complete")),method = "shade")  
print(wine1[,14:17])  
print(wine1[order(-wine1$na\_count),14:17])   
#WineTrain$STARS <- sapply(WineTrain$STARS,function(x) ifelse(is.na(x),0,x))  
#WineTrain<-as.factor(WineTrain)  
#wine1 <- var\_stats(WineTrain)  
#mice imputation  
#print(wine1[order(-wine1$na\_count),14:17])  
print(wine1[order(-wine1$neg\_count),14:17])   
   
   
   
print(wine1[order(-wine1$zero\_count),14:17])  
print(wine1[order(wine1$unique\_count),14:17])   
   
WineTrain$STARS<-as.factor(WineTrain$STARS)  
WineEval$STARS<-as.factor(WineEval$STARS)  
#WineTest   
m1 <- glm(TARGET ~ .-INDEX , family = poisson, data = WineTrain)  
#m1 <- glm(TARGET ~ ., family = poisson, data = WineTrain)  
summary(m1)  
par(mfrow = c(2,2))  
plot(m1)  
m2 <- glm(TARGET ~ VolatileAcidity + CitricAcid + Chlorides + FreeSulfurDioxide  
 + TotalSulfurDioxide + Density + pH + Sulphates + Alcohol + LabelAppeal  
 + AcidIndex + STARS, family = poisson, data = WineTrain)  
summary(m2)  
par(mfrow = c(2,2))  
plot(m2)  
m3 <- glm(TARGET ~ VolatileAcidity + FreeSulfurDioxide + TotalSulfurDioxide + Chlorides + Density + pH + Sulphates + LabelAppeal + AcidIndex + STARS, family=gaussian, data = WineTrain)  
summary(m3)  
par(mfrow = c(2,2))  
plot(m3)  
m4 <- glm(TARGET ~ VolatileAcidity + TotalSulfurDioxide +   
 pH + Sulphates + LabelAppeal + AcidIndex + STARS, family = negative.binomial(1),   
 data = WineTrain)  
summary(m4)  
par(mfrow = c(2,2))  
plot(m4)  
m1AIC <- AIC(m1)  
m1BIC <- BIC(m1)  
m2AIC <- AIC(m2)  
m2BIC <- BIC(m2)  
m3AIC <- AIC(m3)  
m3BIC <- BIC(m3)  
m4AIC <- AIC(m4)  
m4BIC <- BIC(m4)  
  
AIC <- list(m1AIC, m2AIC, m3AIC, m4AIC)  
BIC <- list(m1BIC, m2BIC, m3BIC, m4BIC)  
kable(rbind(AIC, BIC), col.names = c("Model 1", "Model 2", "Model 3", "Model 4")) %>%   
 kable\_styling(full\_width = T)  
  
eval\_p<-predict(m3,WineEval, type = "response")  
write.csv(eval\_p,"predicted\_eval\_values.csv")  
train\_p<-predict(m3,WineTrain, type = "response")  
plot(train\_p,WineTrain$TARGET)  
train\_p<-predict(m2,WineTrain, type = "response")  
plot(train\_p,WineTrain$TARGET)