Nathan Groom

621 Homework 6

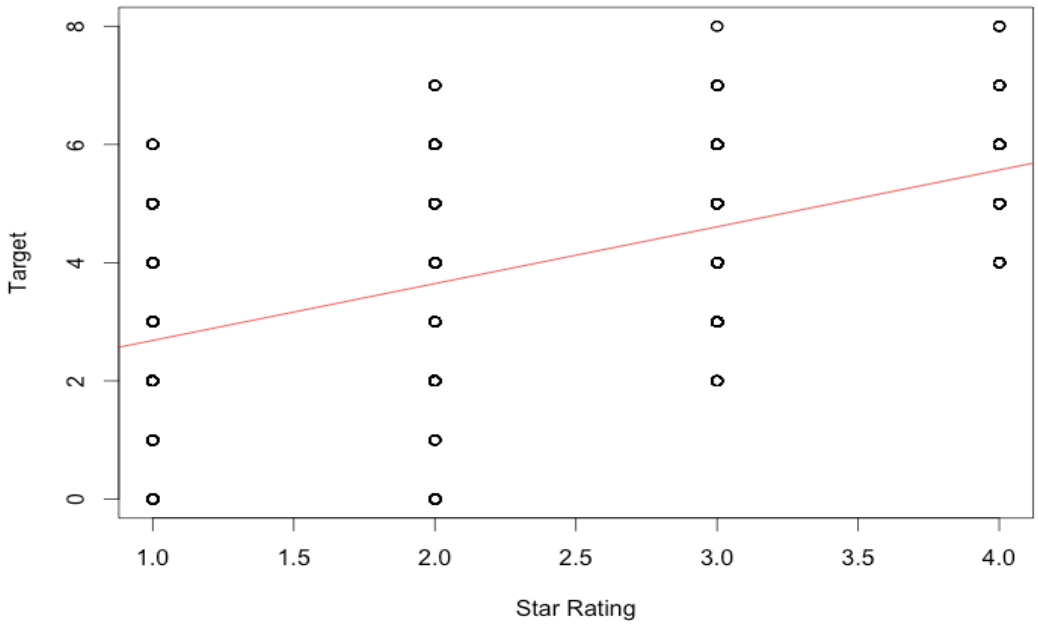
The purpose of this analysis is to explore, transform, and model a dataset of approximately 12,000 commercially available wines, with the intent to determine which features of a particular wine most directly influence its purchase by wine distributors. Our target variable, which represents cases of a certain wine sold, ranges from 0 through 8 and includes all integers in between those values. The predictor variables mostly represent the chemical content of the wines and are as follows: acid index, alcohol content, chloride content, citric acid content, fixed acidity, density, free sulfur dioxide content, label appeal, residual sugar content, sulfate content, total sulfur dioxide content, volatile acidity, pH, and a star rating (from 1 to 4 stars).

Data Exploration:

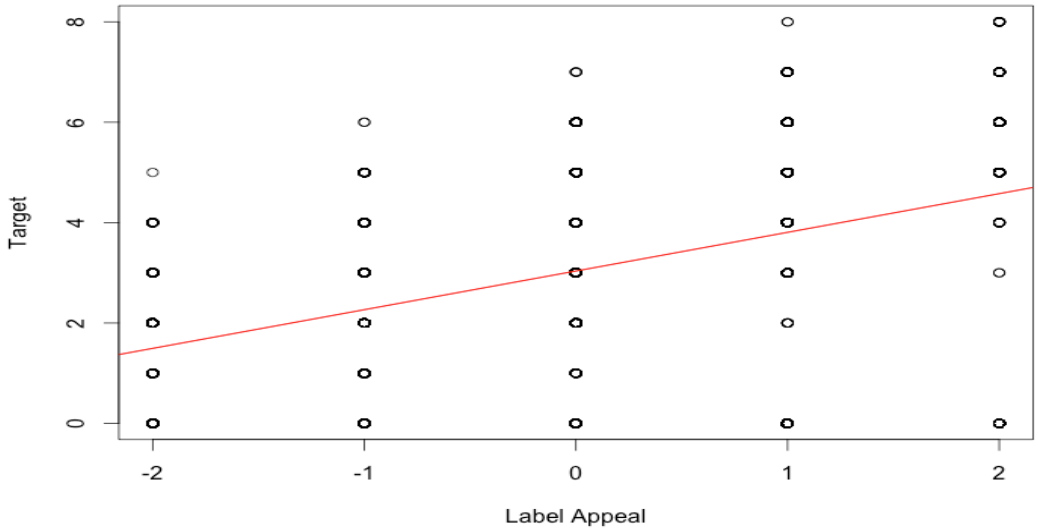
Since all variables are numeric, we can easy see the summary statistics of the whole dataset by using the summary() command. There do not seem to be too many instances in which the median and mean are drastically different. An exception is sugar content, which shows a median of 3.9 and a mean of 5.4, indicating that there may be some outliers with far more sugar content than most wines. However, we also see that, strangely, the minimum sugar content among wines is -127.8. This is strange as it is not possible for a wine to contain a negative amount of sugar. We also see nonsensical negative values for several other features, including sulfur content, acid content, and chloride content. This will have to be addressed in our data transformation section. In addition, there are also several variables with high numbers of NA values. Variables representing sulfur dioxide levels, alcohol content, chloride levels, sugar levels, and star ratings each have hundreds or thousands of NA values. This will also be addressed in the data transformation section. However, it should be noted that if a value of a variable like “sulphates” for example is missing for a wine, this might actually help that wine be purchased if its sulphate level is higher than normal (the distributor making the purchase is unaware of an undesirable quality) or conversely, it might lead to the potential distributor being less likely to buy the wine because the person making the decision to purchase might not trust a wine that does not disclose such information.

As with any data exploration, it helps to plot the target variable against the predictor variables to see if it is possible to visualize any sort of relationship. In this case, however there are only a few obvious visible relationships between the target and the predictors.

The first is with star ratings

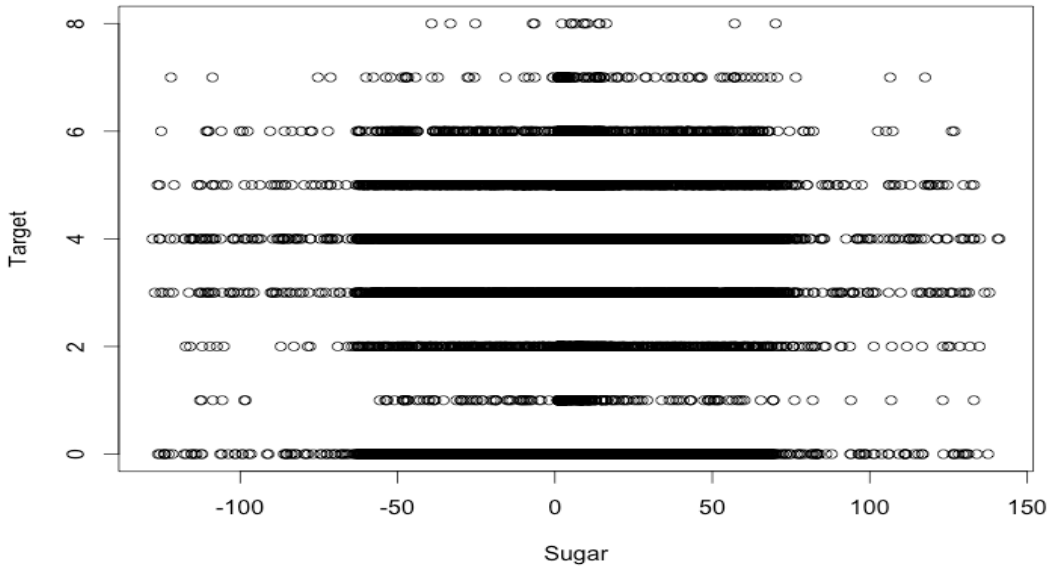


And the second is label appeal:



Both of these predictors are also discrete countable variables that consist of subjective ratings given to the wine by some kind of judge or judges. Both of these are not surprises—star ratings are meant to rate overall quality of wine and this is likely the easiest single statistic for a distributor to observe and use to decide whether or not to buy. Label appeal is similar, though likely less influential. However, the sight of an appealing label would understandably have a stronger influence than the amount of sulfur, sugars, acidity etc. to an average customer (even if not as grounded in reason).

Every other variable shows no clear relationship with the target variable. For example, here we see sugar:



Of course, it's obvious from this plot that half the values are below zero. We will decide what to do with these in the next section.

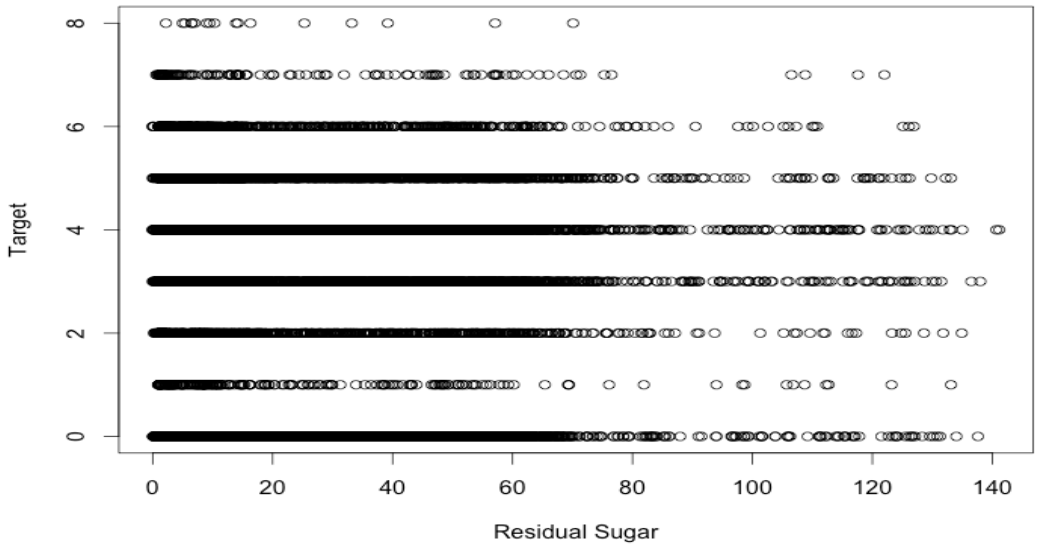
Data Transformation:

Two major issues exist with this data. The first is a high frequency of negative values in variables that should only have non-negative values. The second is a high frequency of NA or missing values.

In the first case, the decision is either to change the variable's negative value to zero, or to remove the negative sign and replace each negative value with its absolute value. One clue in terms of what would be best comes in the above plot of sugar

levels. Approximately half of the values for residual sugar are negative. It is highly unlikely that this number of wines has absolutely no sugar. Far more likely is that these wines do have some sugar and there was an error in the data collection or measurement techniques. Therefore, for the variables with illogical negative values I will replace those negative values with their absolute values. This is the only solution that I believe makes sense.

Replacing negative values with absolute values leads to the following plot:



Still no clear relationship but at least the data is more realistic.

The following variables had values changed from negative to absolute values: Citric acid content, fixed acidity, volatile acidity, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, sulphates, and alcohol.

The next issue is the fact that there are a high number of missing values for certain predictor variables. Sometimes the fact that a variable is missing is actually predictive of its target but, as mentioned above, there exists more than one way this prediction could work. We can assume that missing values are simply missing values and not representative of zero or another value. I think the best way to deal with these are to replace NAs with the column's 40th percentile value, and this is why: some distributors (assuming they care at all about the variable with the missing value) might look at missing values as missing because the winemaker declined to report a less than desirable rating. However another distributor might look at that same wine and decide that the value is missing not because it was withheld by the winemaker but because it simply was not available for another

reason. Therefore, even if the true value of the missing variable were to be considered below average desirability by the potential distributor, the wine would get the benefit of the doubt. We can split the difference in our models. We will assume some of the missing values were withheld by a winemaker who does not want the true content of sulfates, sugar, acidity to be known (and assume the true value is around the 25th percentile of the most desirable outcome of any wine in that column), and half of the missing values are missing for an unknown reason (and assume that the true value is simply the median value of the column. I would suspect that the variables missing for an "unknown reason" is the case more often, so therefore I will replace all NA values with the 40th percentile of the most desirable value of that variable in its respective column in the dataset (with some exceptions to be noted below).

The NA values will be replaced as follows: Fixed Acidity: 7.5, Volatile Acidity: 0.57, Citric Acid: 0.57, Sugar: 19.6, Chlorides: .198, Free Sulfur Dioxide: 89, Total Sulfur Dioxide: 182, Sulfates: .71, Alcohol: 10.4, Stars: 2, pH: 3.2.

Since all the items above are considered undesirable, except for alcohol and stars, I'm actually taking the 60th percentile of these variables because a lower value is more desirable, and a higher value less desirable. Alcohol can be considered undesirable at too high or too low levels so I am taking the 50th percentile with alcohol, as well as the 50th percentile with star rating since this is not naturally occurring and its direct valuation is not in the hands of the winemaker. Sugar is a bit of a tossup since too little or too much could also be considered to be undesirable, but ultimately I decided to stick with the 60th percentile level because I believe too sweet would be less desirable than not sweet enough when it comes to wine (though I understand that some people might have the opposite opinion). Finally, I similarly do not believe that the pH value is considered either desirable or undesirable; it only measures a chemical property of the wine and is not exactly an "ingredient" in the way sugar or alcohol would be. Therefore, I am replacing the pH column's NA values with the median pH value.

Build Models:

When it comes to building the models for our analysis, we will end up building six total models: two Poisson regression models, two negative binomial models, and two multiple linear regression models. Each of the three aforementioned categories will consist of one stepwise model and one model with manually selected variables.

The first model, built in the Poisson family and created with stepwise selection, returns most of the variables in the model formula. In fact the only variable that is fully removed by R is Fixed Acidity. There is one other variable, Free Sulfur Dioxide, that is not significant at the 0.05 level. As there are probably too many variables anyway, we should remove this one as well. That leaves us with 11 of the 13 original predictor variables, and all are highly significant, with an AIC of 50444. This is

clearly a well-fitting model, but considering it has almost all of the predictor variables in, it is likely overfitting the data to some degree as well. And this is not to mention that there are several variables measuring the acidity of the wine to some degree: Volitile Acidiy, Citric Acid, Acid Index, and pH. Therefore, for the next model, I am going to take three of these acid variables out and stick with Acid Index, which in the dataset is described as a "proprietary method of testing total acidity of wine by using a weighted average." Also, I think we should remove the star rating because this is essentially a combination of all of the other predictors and not really an independent predictor of how popular a wine is going to be. So to recap, this is the manually selected model, using the Poisson family. We will be including the following variables: Chlorides, Total Sulfur Dioxide, Density, Sulphates, Alcohol, Label Appeal, and Acid Index.

Building the above model shows every variable significant at p=0.05 except for Chlorides. For the sake of having fewer variables, we will once again remove this insignificant variable and rebuild the model. Now all six variables in this model are significant, with an AIC of 51654.

Next we will try an overdispersion test for both of these Poisson models. The stepwise model gives us a 0.9962 p-value for overdispersion, but the manual model gives us a 0.08876 p-value for overdispersion—not low enough to disregard the model but perhaps enough to consider using a negative binomial instead.

Now a word about the coefficients, which for the manual model are as follows: 0.0001289 for total sulfur dioxide, -0.5211 for density, -0.02511 for sulphates 0.008855 for alcohol, 0.2589 for label appeal, and -0.1357 for acid index.

For the stepwise model they are: -0.05902 for volatile acid, 0.0174 for citric acid, -0.04618 for chlorides, 0.00011 for total sulfur, -0.447987 for density, -0.02363 for pH, -0.02402 for sulphates, 0.00588 for alcohol, 0.19650 for label appeal, -0.12389 for acid index, and 0.22107 for stars.

There is not too much we can tell about these coefficients simply by looking at them. They are part of a larger exponential equation that can help us figure out that variable's effect on the target. One thing we can immediately tell is whether the coefficients are positive or negative. The coefficients make sense, in this way, for the most part. I would assume only someone with a somewhat extensive knowledge of chemistry or wine, or both, would immediately know if higher or lower pH values, sulfur, sulphates, or density make a wine "better" or more appealing, but label appeal and stars both have positive ratings, which does make sense. Alcohol does have a (slight) positive rating as well, which is what I would assume.

The next step is to build two models with the negative binomial family. This can be done easily with the glm.nb() function in the MASS package.

After the first model, with stepwise variable selection, is created, we see that the model is equivalent to the Poisson model. This happens relatively often and is not a surprising occurrence. With the Poisson stepwise model, we proceeded to remove

Free Sulfur Dioxide, but for the sake of variety, we can leave that variable in (even though I generally believe the model with less variables to be better). So we can leave this negative binomial model as it is. All variables are significant except for Free Sulfur Dioxide which has a p-value of 0.0578. The most significant variables are Acid Index, Label Appeal, and Stars.

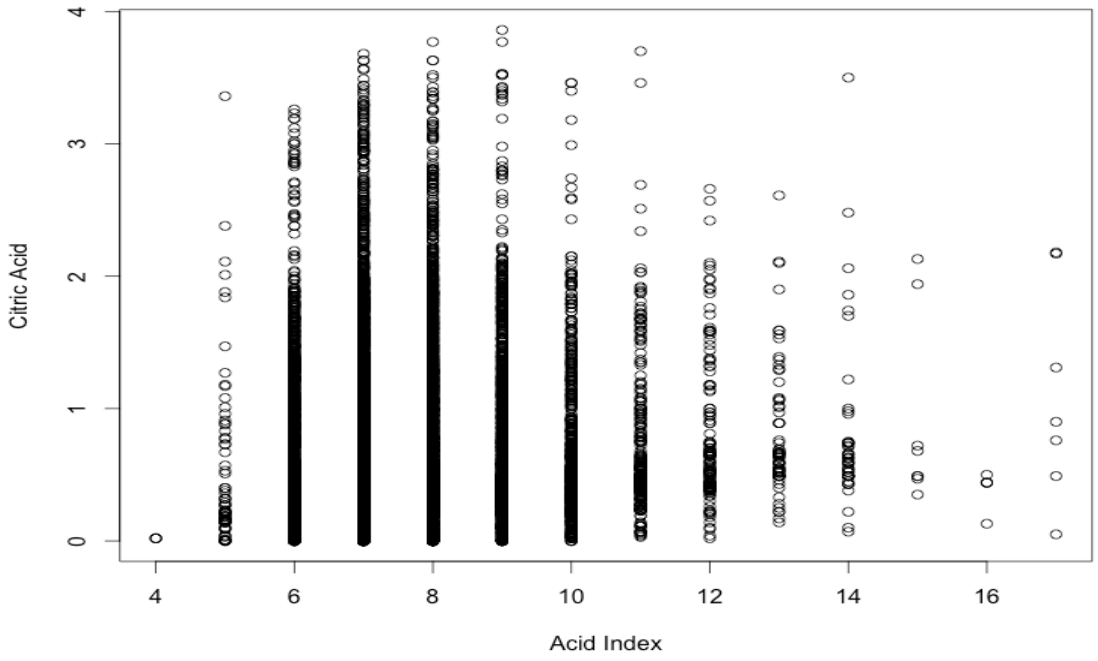
Next we can build a manually selected model using the negative binomial family. First, we can use the same variables from our Poisson manual model to see if there are any differences. Again, these are the Total Sulfur, Density, Sulphates, Alcohol, Label Appeal, and Acid Index variables. It turns out this model is again very similar to its Poisson equivalent. It has an AIC of 51656 whereas the Poisson model had an AIC of 51654. They have the same log likelihood of -25820 but the Poisson model has 7 degrees of freedom whereas the Negative Binomial model has 8. All in all, these are very similar models. However, with the very low p-value resulting from the dispersion test for the Poisson model, I would pick the negative binomial manual model over the Poisson. More on this in the next section.

Lastly, we will build two multiple linear regression models and observe how similar these are to the Poisson and negative binomial models. Like the models from those families, one will have variables chosen through a stepwise selection and one will be manually chosen.

Amazingly, the variables chosen with the stepwise selection are exactly the same as both the Poisson and negative binomial stepwise models. Again, all variables are significant other than Free Sulfur Dioxide, which is nearly significant with a p-value of 0.05033. Still, for the sake of having fewer variables, I am going to once again remove Free Sulfur Dioxide, leaving us with 11 variables in our linear model. The difference between this model, however, and the previous models, is that the coefficients are easier to understand. There are almost definitely some collinearity issues here but we can address those in our next (manual) model. First, we should comment on the coefficients. Because it is a linear model, we can say that as one of the variables goes up by one unit, the target is affected by the amount of that predictor's coefficient. Those coefficients are as follows. Volitile Acidity: -0.169, Citric Acid: -0.054, Chlorides: -0.1467, Total Sulfur Dioxide: 0.00035, density, -1.133, pH: -0.0634, Sulphates: -0.0709, Alcohol: 0.0203, Label Apeal: 0.5943, Acid Index: -0.3309, and Stars: 0.7505. We can see that Label Appeal and Stars both have a strong influence on the target. Density has the largest coefficient, but since its values range only from 0.8881 to 1.0990, it does not have a huge effect on the target compared to Stars (1 to 4) or Label Appeal (-2 to 2). Similarly to coefficients in earlier models, none of the other variables have very intuitive coefficient values here (for someone who is not a wine expert), but at least it is easier to understand the effects of each variable. In terms of positive/negative influence, every coefficient in the linear model has the same sign as the previous stepwise models.

The manually chosen multiple linear model, like the previous manually selected models, will have some variables that are clearly related removed. As mentioned

previously, there are four different variables that measure acidity in some respect. Here is a plot of Acid Index and Citric Acid:



It almost looks like a normal distribution. In any case, it clearly is not random. There are multiple other similar relationships in this data. So again, we will remove variables with such conflicts and revert back to the previous combination of Total Sulfur Dioxide, Density, Sulfates, Alcohol, Label Appeal and Acid Index.

Building this model we once again get all significant variables with the coefficients as follows. Total Sulfur Dioxide: 0.00038, Density: -1.5809, Sulphates: -0.07153, Alcohol: -0.02859, Label Appeal: 0.784331, Acid Index: -0.36414. Nothing is very controversial here, but it appears that this model just is not a very good fit. The R-Squared value is 0.1968 and the adjusted R-Squared is 0.1964. The F-statistic is 522.2 on 6 and 12788 degrees of freedom. I would not be very inclined to use this model. Which brings us to our final section.

Select Models:

My choice for the best model of the six presented above is the negative binomial model with manually selected variables. The reasons I have chosen this model are simple. Firstly, I believe the models chosen with stepwise selection have too many variables, several of which measure similar or related features of the wine. The

variables are indeed shown to be significant in the model summary, but I am afraid that these models could be overfitting the data. I could remove some variables, but then it is essentially a different model. This is what I ended up doing to create the models with manually selected variables. I tried to choose variables that each represented a different, unique aspect of the wine. I did throw away the "Stars" rating because I thought, like the target, it too would be a culmination of all other variables in the data (not to mention a high number of missing values in the original dataset). The reason I chose the negative binomial over the Poisson family was because when I ran the dispersion test, the Poisson model seemed a bit too close to overdispersion for my liking. Poisson model did not fail the hypothesis test so in theory it would be OK to use, but it was on the cusp of failing. Therefore, just to be safe, I feel it is more appropriate to use the negative binomial model. This is essentially the only reason to choose the negative binomial over the Poisson model because all the summary statistics and coefficients are the same between those two models. There is a very slight, insignificant difference in AIC and log likelihood (8 degrees of freedom for negative binomial, 7 for Poisson), but for all practical purposes these are the same. As for the multiple linear regression models, they simply are not the most appropriate choice for this data. Multiple linear regression is designed for continuous values, but even more importantly, it can predict negative values. A distributor can not order negative cases of wine, so these models are not the right choice when we can build count regression models instead.

Unsurprisingly, making predictions using both the negative binomial and the Poisson give strikingly similar results. A summary of the predictions (using training data) are as follows:

Negative Binomial Model:

Min	1st Qu	Median	Mean	3rd Qu	Max
0.7479	2.4250	2.9190	3.0290	3.5730	7.7700

Poisson Model:

Min	1st Qu	Median	Mean	3rd Qu	Max
0.7479	2.4250	2.9190	3.0290	3.5730	7.7690

Actual Target Variable Summary:

Min	1st Qu	Median	Mean	3rd Qu	Max
0.000	2.000	3.000	3.0290	4.000	8.000

We can see that both these models are very good and almost identical. The most surprising thing is that the mean for the training data is exactly the same for the models as for the actual target variable, down to the thousandth!

Now we can predict the target variable using the evaluation dataset and my "choice" model, the negative binomial model (even though using the Poisson model would be just as good).

Predicted Target Variable—Evaluation Data Set:

Min	1st Qu	Median	Mean	3rd Qu	Max	NAs
0.6929	2.4330	2.9530	3.0520	3.5730	7.5270	615

We can see that there are 615 NA values because this dataset was not cleaned like the training data. Other than that however, the predictions look similar to what we saw in the training data. The median and mean values are close, which means the data is relatively normal. This is a good model.

Appendix: R Code

```
library(MASS)

library(AER)

library(pscl)

wt=read.csv("/users/nathangroom/desktop/621csvs/wine-training-data.csv")

we=read.csv("/users/nathangroom/desktop/621csvs/wine-evaluation-data.csv")

names(wt)<-
c('index','target','f.acid','v.acid','c.acid','sugar','chlorides','f.sulfer','t.sulfer','density','ph','sulphates','alcohol','l.appeal','acidindex','stars')

names(we)<-
c('index','target','f.acid','v.acid','c.acid','sugar','chlorides','f.sulfer','t.sulfer','density','ph','sulphates','alcohol','l.appeal','acidindex','stars')

## data transformation ##

#replacing missing values with 0

wt.1=wt

wt.2=wt

names(wt.1)<-
c('index','target','f.acid','v.acid','c.acid','sugar','chlorides','f.sulfer','t.sulfer','density','ph','sulphates','alcohol','l.appeal','acidindex','stars')

names(wt.2)<-
c('index','target','f.acid','v.acid','c.acid','sugar','chlorides','f.sulfer','t.sulfer','density','ph','sulphates','alcohol','l.appeal','acidindex','stars')

wt.1$f.acid[is.na(wt.1$f.acid)]<-0

wt.1$v.acid[is.na(wt.1$v.acid)]<-0

wt.1$c.acid[is.na(wt.1$c.acid)]<-0

#no such thing as negative acid content

wt.1$v.acid[wt.1$v.acid<0]<-0

wt.1$c.acid[wt.1$c.acid<0]<-0

wt.1$f.acid[wt.1$f.acid<0]<-0

#no such thing as negative sugar
```

```
wt.1$sugar[is.na(wt.1$sugar)]<-0
wt.1$sugar[wt.1$sugar<0] <- 0
wt.1$chlorides[is.na(wt.1$chlorides)]<-0
#no such thing as negative chlorides
wt.1$chlorides[wt.1$chlorides<0]<-0
wt.1$f.sulfer[is.na(wt.1$f.sulfer)]<-0
#no such thing as negative sulfur
wt.1$f.sulfer[wt.1$f.sulfer<0]<-0
wt.1$t.sulfer[is.na(wt.1$t.sulfer)]<-0
wt.1$t.sulfer[wt.1$t.sulfer<0]<-0
wt.1$density[is.na(wt.1$density)]<-0
#replace ph NAs with 3 (approx mean) because 0 is unrealistically acidic
wt.1$ph[is.na(wt.1$ph)]<-3
wt.1$sulphates[is.na(wt.1$sulphates)]<-0
#no such thing as negative sulphates
wt.1$sulphates[wt.1$sulphates<0]<-0
#no such thing as negative alcohol
wt.1$alcohol[is.na(wt.1$alcohol)]<-0
wt.1$alcohol[wt.1$alcohol<0]<-0
wt.1$l.appeal[is.na(wt.1$l.appeal)]<-0
wt.1$acidindex[is.na(wt.1$acidindex)]<-0
wt.1$acidindex[wt.1$acidindex<0]<-0
#shouldn't assume NA for stars is '0', just not rated
#wt.1$stars[is.na(wt.1$stars)]<-0

#plots#
plot(x=wt.1$f.acid,y=wt.1$target)
```

```
plot(x=wt.1$v.acid,y=wt.1$target)
plot(x=wt.1$c.acid,y=wt.1$target)
plot(x=wt.1$sugar,y=wt.1$target)
plot(x=wt.1$chlorides,y=wt.1$target)
plot(x=wt.1$f.sulfer,y=wt.1$target)
plot(x=wt.1$t.sulfer,y=wt.1$target)
plot(x=wt.1$density,y=wt.1$target)
plot(x=wt.1$ph,y=wt.1$target)
plot(x=wt.1$sulphates,y=wt.1$target)
plot(x=wt.1$alcohol,y=wt.1$target)
plot(x=wt.1$l.appeal,y=wt.1$target)
plot(x=wt.1$acidindex,y=wt.1$target)
plot(x=wt.1$stars,y=wt.1$target)

#there are very few visible relationships with the target variable
# "stars' is clearly a positive relationship
# 'acidindex' looks like a slight negative relationship
# 'label appeal' definitely has a positive relationship
# nothing else is obvious in terms of a relationship.

#explore mean/median
summary(wt.1)
#nothing where median is very different than mean
# except "sugar" where median is 2.9 and mean is 5.15; so there must be
# several very sugary wines.
# with 'stars' the median is 1.0 and the mean is 1.5
wt.2<-wt.2[-1]
```

```
#trying more transformation: instead of replacing negative w/ zero
# now replace negative with absolute values.
wt.2$f.acid<-abs(wt.2$f.acid)
wt.2$c.acid<-abs(wt.2$c.acid)
wt.2$v.acid<-abs(wt.2$v.acid)
wt.2$sugar<-abs(wt.2$sugar)
wt.2$chlorides<-abs(wt.2$chlorides)
wt.2$f.sulfer<-abs(wt.2$f.sulfer)
wt.2$t.sulfer<-abs(wt.2$t.sulfer)
wt.2$sulphates<-abs(wt.2$sulphates)
wt.2$alcohol<-abs(wt.2$alcohol)

#finding out what to replace NA values with
quantile(wt.2$c.acid,.60)
quantile(wt.2$f.acid,.60)
quantile(wt.2$v.acid,.60)
quantile(na.omit(wt.2$sugar),.60)
quantile(na.omit(wt.2$chlorides),.60)
quantile(na.omit(wt.2$f.sulfer),.60)
quantile(na.omit(wt.2$t.sulfer),.60)
quantile(na.omit(wt.2$sulphates),.60)
quantile(na.omit(wt.2$alcohol),.50)
quantile(na.omit(wt.2$stars),.50)

wt.2$f.acid[is.na(wt.2$f.acid)]<-7.5
wt.2$v.acid[is.na(wt.2$v.acid)]<-0.57
wt.2$c.acid[is.na(wt.2$c.acid)]<-0.57
```

```
wt.2$sugar[is.na(wt.2$sugar)]<-19.6

wt.2$chlorides[is.na(wt.2$chlorides)]<-.198

wt.2$f.sulfer[is.na(wt.2$f.sulfer)]<-89

wt.2$t.sulfer[is.na(wt.2$t.sulfer)]<-182

wt.2$sulphates[is.na(wt.2$sulphates)]<-.71

wt.2$alcohol[is.na(wt.2$alcohol)]<-10.4

wt.2$stars[is.na(wt.2$stars)]<-2

wt.2$ph[is.na(wt.2$ph)]<-3.2



##build models##

p1<-glm(target~.,data=wt.2,family=poisson)

p1back=step(p1)

formula(p1back)

p1back2<-glm(target ~ v.acid + c.acid + chlorides +  t.sulfer +

        density + ph + sulphates + alcohol + l.appeal + acidindex +

        stars, data=wt.2,family='poisson')


# manual

p3<-
glm(target~chlorides+t.sulfer+density+sulphates+alcohol+l.appeal+acidindex,data=
wt.2, family='poisson')

#manual v.2 (without chlorides)

p4<-glm(target~t.sulfer+density+sulphates+alcohol+l.appeal+acidindex,data=wt.2,
family='poisson')


dispersiontest(p1back2)

dispersiontest(p4)
```

```
#negative binomial

#build it similar to how we built previous model


nb1<-glm.nb(target~.,data=wt.2)

nb1back<-step(nb1)

formula(nb1back)

#formula is the same as stepwise model from poisson family

nb2<-
glm.nb(target~t.sulfer+density+sulphates+alcohol+l.appeal+acidindex,data=wt.2)

# log liklihoods

logLik(nb2)

logLik(p4)

nb2$coefficients

p4$coefficients


#multiple linear regression

mlr1<-lm(target~.,data=wt.2)

mlr1.back=step(mlr1)

formula(mlr1.back)

#remove f.sulfer

mlr1.back2<-lm(target ~ v.acid + c.acid + chlorides + t.sulfer +

        density + ph + sulphates + alcohol + l.appeal + acidindex +

        stars,data=wt.2)

mlr2<-
lm(target~t.sulfer+density+sulphates+alcohol+l.appeal+acidindex,data=wt.2)


#ANOVA test
```

```
anova(mlr2,nb2,p4,test="Chisq")
summary(wt.3)

#my preferred model is nb2
wt.3$predictions<-predict(nb2,newdata=wt.3,type='response')
summary(wt.3$predictions)
summary(wt.3$target)

we.1=we
we.1$predictions<-predict(nb2,newdata=we.1,type='response')
summary(we.1$predictions)
```