IS621_hw6

Charley Ferrari

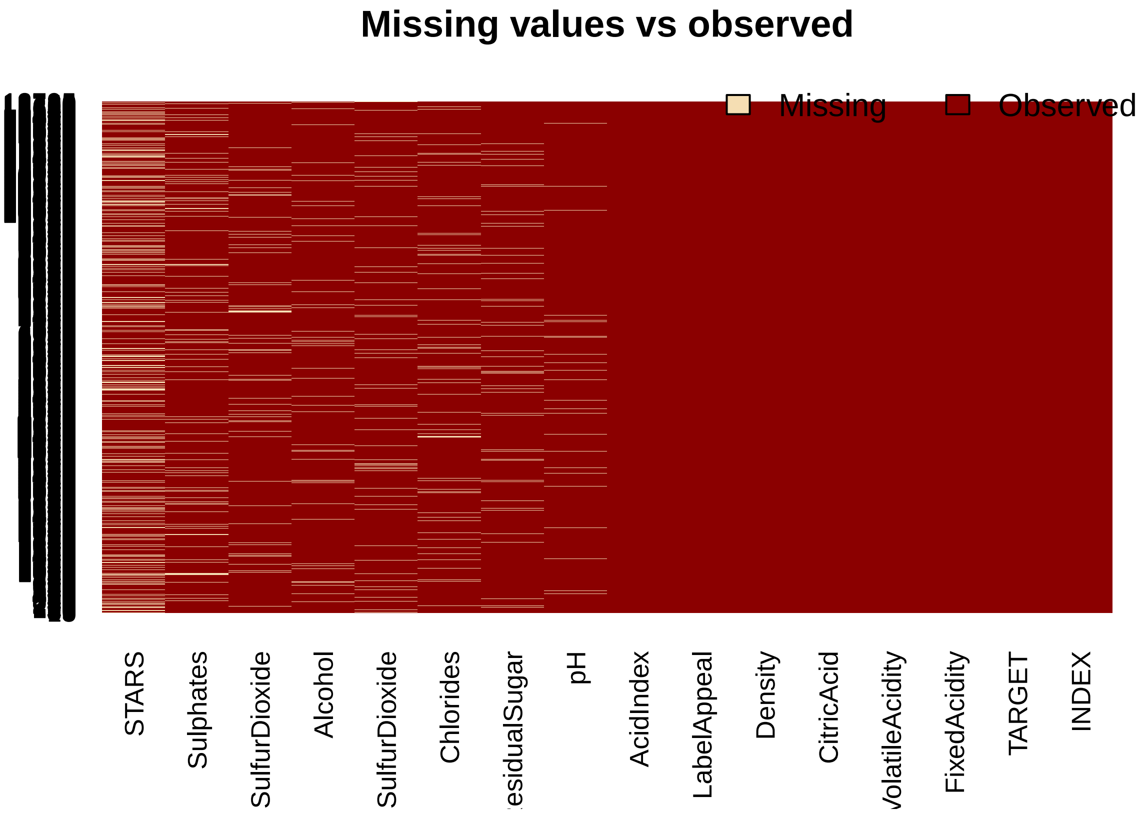
April 21, 2016

Data Exploration

The wine dataset includes 12795 observations of 16 variables (one of which is the INDEX.) There are a substantial number of NA values, as seen in the table below:

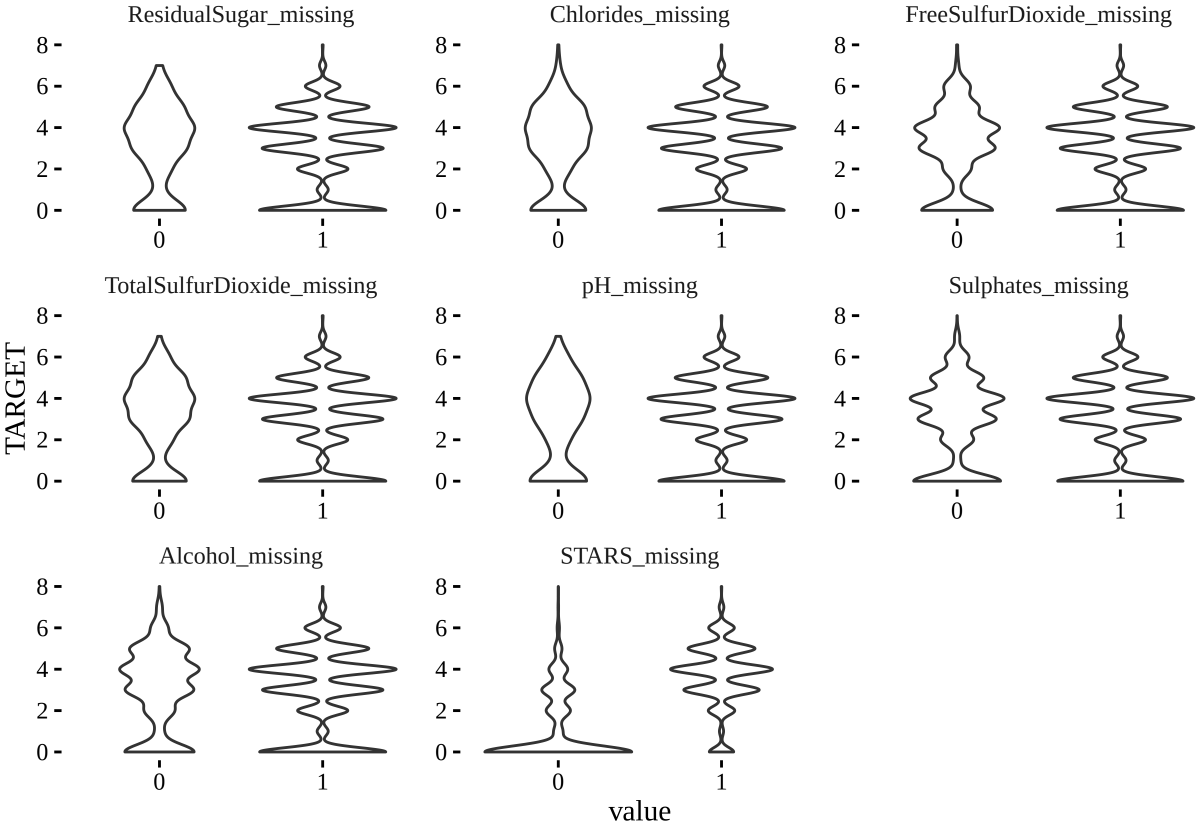
variable	NAs
TARGET	0
FixedAcidity	0
VolatileAcidity	0
CitricAcid	0
ResidualSugar	616
Chlorides	638
FreeSulfurDioxide	647
TotalSulfurDioxide	682
Density	0
pH	395
Sulphates	1210
Alcohol	653
LabelAppeal	0
AcidIndex	0
STARS	3359

The high number of NA values deserve further investigation. First, lets check out a matrix of the NAs to visualize any patterns in the missing values:

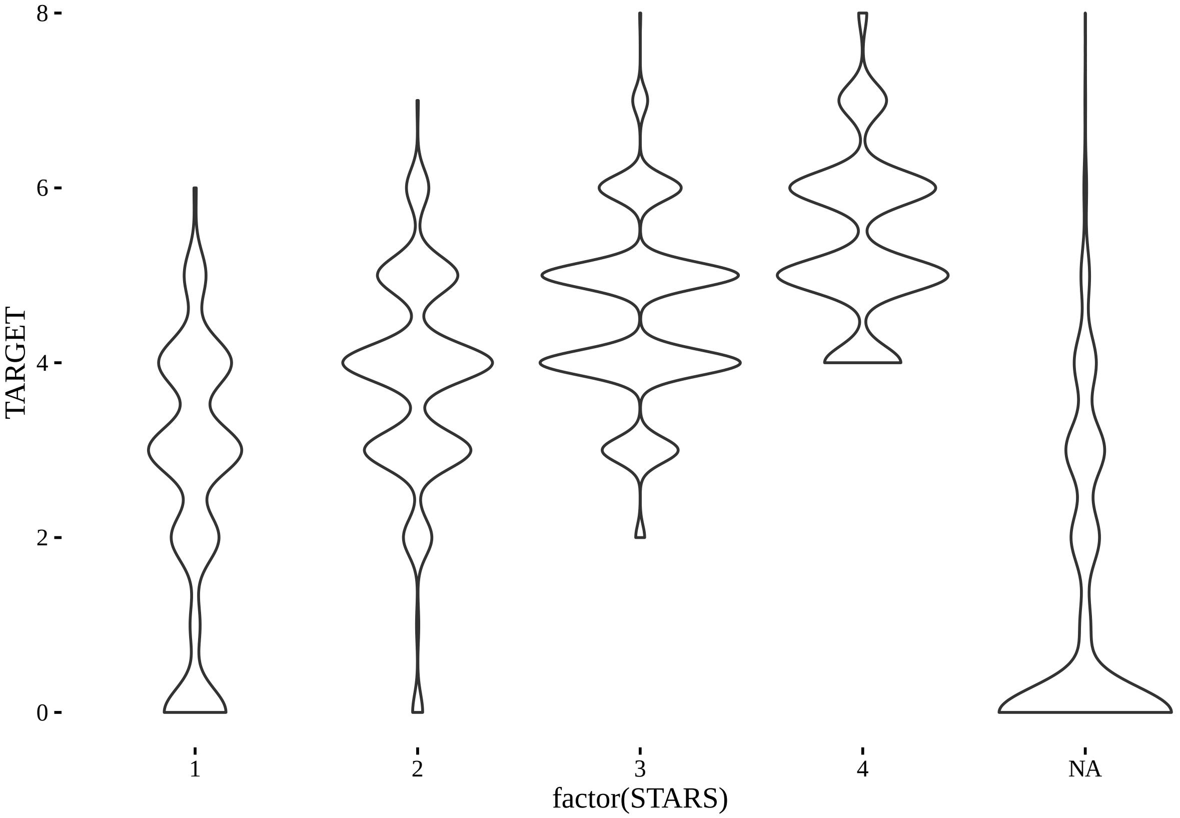


There seem to be no overarching patterns with the NA values. The STARS variable has the highest number of missing values. It looks like there might be some shared NA values between FreeSulfurDioxide and TotalSulfurDioxide, so I'll look out for multicollinearity between these two variables.

As a last analysis of missing values, I'll see how NAs affect the TARGET variable of number of cases sold. I'll divide the dataset between NA and present for each variable, and plot a violin plot for each case:

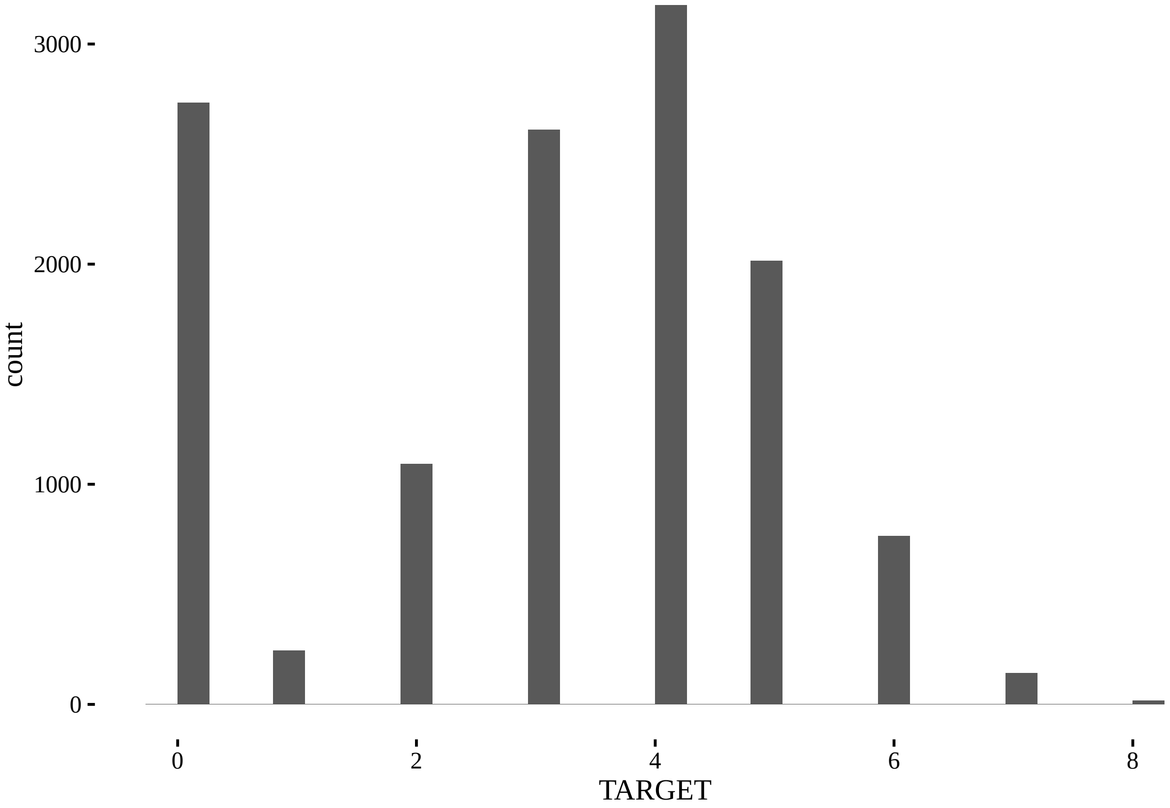


These plots will give us an idea if the NAs change the distribution of our TARGET count or if the presence of NA values can be predictive of our count. In most cases, the distribution looks very similar whether or not the selected variable contains NAs (it looks spikier in the non-NA cases due to higher overall counts.) It would appear however that an NA for STARS leads to a different distribution. Looking at violin plots of each number of stars further proves this:



We will deal with this issue in the data transformation phase.

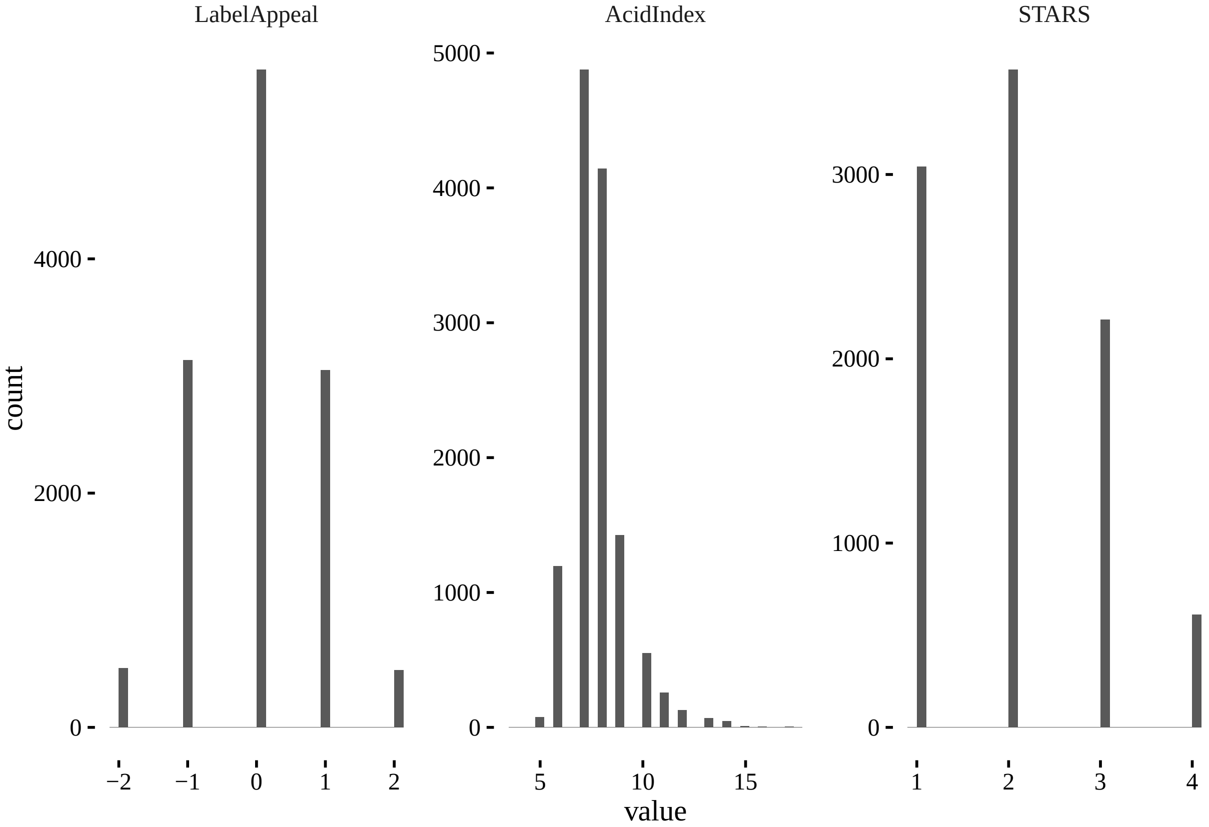
To continue our data exploration, lets examine the types of variables we have. Our target variable, number of cases, is a count variable, with a distribution described in the histogram below:



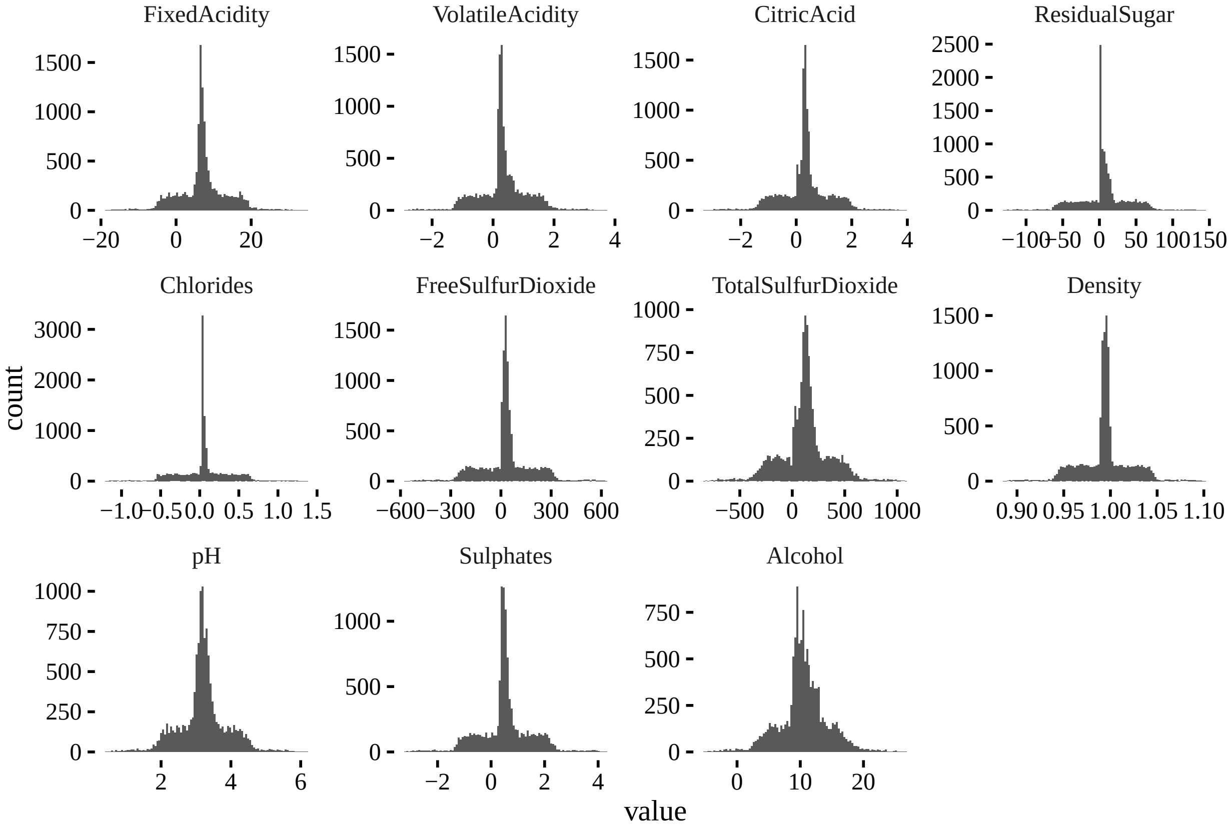
This distribution looks close to a Poisson distribution, but with a large number of zero values. One would think that this could imply hidden NA values. However, I can think of real world reasons for large number of zero values. One could imagine a threshold under which wines won't get stocked in stores, and thus zero cases will be sold.

Next, I'd like to look at the mean and variance of the TARGET variable. If this is indeed a Poisson distribution, I would expect them to be equal. My mean turns out to be 3.029074, while my variance is 3.710895. This would suggest overdispersion. The fact that we have a zero inflation complicates the matter, it would tend to both bring down the mean and the variance, but it's not clear whether it would ultimately lead to over or under dispersion.

Along with STARS, LabelAppeal and AcidIndex are ordinal variables. For purposes of prediction, I will treat them as categorical variables.



AcidIndex seems more classically shaped like a poisson distribution. LabelAppeal and STARS seem a bit more categorical with no particular distribution. Next, lets analyze the rest of the variables, which are numeric:



These variables all appear to be normally distributed, in some cases with high kurtosis. If I chose to transform any of these variables, it would be best to normalize them in some way to prevent additional skews being added. For the time being however, I will leave these variables as is.

Data Transformation

The first data transformation I'll need to perform involves the STARS data. As we mentioned above, this data will be treated as categorical, and the NA values seem to result in a substantially different distribution of TARGET number of cases sold. This suggests that we can treat NA values as a separate category. For the purposes of this model, I will call NA values of stars 0.

After dealing with the STARS variable's NAs, if we omit observations with NAs, we'll end up with 8675 observations out of 12795. Because we have high kurtosis in many of our numeric variables, and because we have enough variables excluding NAs to build a sensible model, I will choose to omit my NAs.

Build Models

I will consider a few different model types for this assignment: Poisson, Negative Binomial, and Linear models. This data is obviously count data, but seems to be overdispersed. This suggests that a negative binomial model might lead to a best fit.

In testing out a few models, I noticed a mixture of p-values when using AcidIndex as a categorical variable. It does seem to be shaped as a Poisson distribution, but that results in a large number of dummy variables. Because of this, I will build models with AcidIndex as a categorical variable and with AcidIndex as a numerical variable, and see how they compare.

I'm more interested in comparing different types of models using the same variables for this exercize. For this reason, for each classification of the AcidIndex variable, I will perform a backwards stepwise algorithm from a full model using the Akaike Information Criterion.

I will end up with a total of six models: Three with AcidIndex treated as numeric and three with AcidIndex treated as categorical. Each of these groups will have the same variable selection for comparability.

The estimates for my coefficients are described below:

Variable	Linear Model Numeric	Negative Binomial Model Numeric	Poisson Model Numeric
(Intercept)	3.8448366	1.1864392	1.1864140
AcidIndex	-0.1964924	-0.0793470	-0.0793446
Alcohol	0.0112043	0.0030019	0.0030020
Chlorides	-0.1386831	-0.0453040	-0.0453027
Density	-1.0593718	-0.3703344	-0.3703289
FreeSulfurDioxide	0.0002842	0.0000951	0.0000951
LabelAppeal	0.4638881	0.1581892	0.1581899
STARS1	1.3384665	0.7485854	0.7485864
STARS2	2.3686379	1.0724564	1.0724572
STARS3	2.9288460	1.1894101	1.1894103
STARS4	3.6502235	1.3095355	1.3095349
Sulphates	-0.0316577	-0.0132385	-0.0132379
TotalSulfurDioxide	0.0002693	0.0000934	0.0000934
VolatileAcidity	-0.0945622	-0.0307698	-0.0307691

Variable	Linear Model Categorical	Negative Binomial Model Categorical	Poisson Model Categorical
(Intercept)	3.6519085	0.9522887	0.9522441
Alcohol	0.0118687	0.0033771	0.0033772
Chlorides	-0.1350581	-0.0447674	-0.0447662
Density	-1.0897445	-0.3873198	-0.3873157
FreeSulfurDioxide	0.0002688	0.0000893	0.0000893
LabelAppeal	0.4665369	0.1586974	0.1586980
STARS1	1.3235676	0.7376002	0.7376008
STARS2	2.3523227	1.0601581	1.0601583
STARS3	2.9071546	1.1755868	1.1755865
STARS4	3.6274829	1.2952943	1.2952931
Sulphates	-0.0271592	-0.0111025	-0.0111020
TotalSulfurDioxide	0.0002581	0.0000865	0.0000865
VolatileAcidity	-0.0946332	-0.0296755	-0.0296748
AcidIndex10	-1.8438450	-0.5906577	-0.5906132
AcidIndex11	-2.4150011	-1.0042186	-1.0041695
AcidIndex12	-2.4420925	-1.0344369	-1.0343864
AcidIndex13	-2.3940593	-0.8187574	-0.8187072
AcidIndex14	-2.0418117	-0.8175601	-0.8175145
AcidIndex15	-1.4669859	-0.4492920	-0.4492487
AcidIndex16	-2.5799743	-0.9883979	-0.9883437
AcidIndex17	-2.5643313	-1.1199846	-1.1199252
AcidIndex5	-1.2314592	-0.3317036	-0.3316653
AcidIndex6	-1.0277317	-0.2615304	-0.2614925
AcidIndex7	-1.1675872	-0.3073736	-0.3073349
AcidIndex8	-1.2446811	-0.3302843	-0.3302454
AcidIndex9	-1.5314220	-0.4366375	-0.4365949

The coefficients of the linear models cannot be compared directly compared to the Poisson and Negative Binomial models. The latter two models have log-link functions. While linear models are in the form:

$$Y = \beta_0 + \beta_1 \times x_1 + \beta_2 \times x_2 + ...$$

Negative Binomial and Poisson models are in the form:

$$log(Y) = \beta_0 + \beta_1 \times x_1 + \beta_2 \times x_2 + ...$$

This is another way of saying entire right side of the equation is exponentiated.

This has ramifications for the interpretations of the coefficients. For linear regression, a unit change in a variable x_n results in a $\beta_n \times x_n$ change in the response variable. For Poisson and Negative Binomial, the original Y would be multiplied by β_n.

Select Models

Lets take a look at the p-scores of our variables for each of the models:

Variable	Linear Model Numeric	Negative Binomial Model Numeric	Poisson Model Numeric
(Intercept)	0.0000000	0.0000005	0.0000005
AcidIndex	0.0000000	0.0000000	0.0000000
Alcohol	0.0033814	0.0734285	0.0734055
Chlorides	0.0016518	0.0194850	0.0194831
Density	0.0461108	0.1122050	0.1121943
FreeSulfurDioxide	0.0027908	0.0223392	0.0223373
LabelAppeal	0.0000000	0.0000000	0.0000000
STARS1	0.0000000	0.0000000	0.0000000
STARS2	0.0000000	0.0000000	0.0000000
STARS3	0.0000000	0.0000000	0.0000000
STARS4	0.0000000	0.0000000	0.0000000
Sulphates	0.0366569	0.0467545	0.0467542
TotalSulfurDioxide	0.0000115	0.0005339	0.0005340
VolatileAcidity	0.0000001	0.0000981	0.0000980

Variable	Linear Model Categorical	Negative Binomial Model Categorical	Poisson Model Categorical
(Intercept)	0.0005945	0.0148262	0.0148241
Alcohol	0.0018651	0.0442252	0.0442097
Chlorides	0.0021226	0.0210621	0.0210598
Density	0.0397038	0.0967078	0.0966965
FreeSulfurDioxide	0.0045656	0.0318846	0.0318820
LabelAppeal	0.0000000	0.0000000	0.0000000
STARS1	0.0000000	0.0000000	0.0000000
STARS2	0.0000000	0.0000000	0.0000000
STARS3	0.0000000	0.0000000	0.0000000
STARS4	0.0000000	0.0000000	0.0000000
Sulphates	0.0724051	0.0957286	0.0957286
TotalSulfurDioxide	0.0000253	0.0013546	0.0013548
VolatileAcidity	0.0000001	0.0001744	0.0001744
AcidIndex10	0.0469890	0.0641327	0.0641317

Variable	Linear Model Categorical	Negative Binomial Model Categorical	Poisson Model Categorical
AcidIndex11	0.0094837	0.0019565	0.0019559
AcidIndex12	0.0090422	0.0018248	0.0018243
AcidIndex13	0.0114291	0.0153335	0.0153323
AcidIndex14	0.0318450	0.0196127	0.0196111
AcidIndex15	0.1698887	0.2947339	0.2947527
AcidIndex16	0.0228747	0.0714352	0.0714396
AcidIndex17	0.0192704	0.0411682	0.0411711
AcidIndex5	0.1923021	0.3091874	0.3092055
AcidIndex6	0.2675154	0.4099465	0.4099770
AcidIndex7	0.2073701	0.3322105	0.3322328
AcidIndex8	0.1789604	0.2975248	0.2975431
AcidIndex9	0.0984376	0.1690476	0.1690552

There are a few p-values above a 0.05 threshold, which make me question using the AIC as my selection criterion. Once again, my results for the Poisson and Negative Binomial models are very similar. One other problem this brings up is the treatment of categorical variables. For a variable like AcidIndex, with a high number of categories, there's more of a chance that some of them are not significant. Because of this it might be better to use the AcidIndex as a numeric variable.

Because we're comparing different model types, it's tough to find a common measure for them. The Mean squared error, however, should still be comparable:

ModelName	MSE	SE
Linear Model Numeric	1.704721	0.0302421
Linear Model Categorical	1.705351	0.0286477
Negative Binomial Model Numeric	1.719217	0.0286477
Negative Binomial Model Categorical	1.704719	0.0302421
Poisson Model Numeric	1.717888	0.0299897
Poisson Model Categorical	1.717890	0.0299897

These results deviate from the pattern we were seeing before. It seems like the Negative Binomial model is less comparable to the Poisson. In fact, depending on our choice of treatment for the AcidIndex variable, the MSE is either the highest of the six or the lowest.

Based on the raw results, this would suggest that we treat AcidIndex as a categorical variable, and run a Negative Binomial Model. The overdispersion supports this choice, since Poisson models imply that the mean and variance are equal. Although, the variability in my MSE suggests that the results will be highly dependent on your variable choices. If you don't optimize your model, you might be better off with a Linear model for example.

For further research, I would look into performing a zero-inflated model for this data. It was obvious that there was a high number of zero counts, and imagineable that these results are natural (and not hidden NAs). One thing I noticed was that there is a pretty high corellation between NA values in STARS and 0 counts, which might suggest a two step model combining Logistic and Poisson or Negative Binomial models.

Appendix

NA Table Creation:

```r
winetest <- select(wine, -INDEX)

natable <- data.frame(variable = colnames(winetest), NAs =
                        c(0,0,0,0,616,638,647,682,0,395,1210,653,0,0,3359))

kable(natable)

missmap(wine, main = "Missing values vs observed")
```

Violin Plots comparing NAs:

```r
winemissing <- wine

winemissing$ResidualSugar_missing <- factor(ifelse(is.na(wine$ResidualSugar),0,1))
winemissing$Chlorides_missing <- factor(ifelse(is.na(wine$Chlorides), 0, 1))
winemissing$FreeSulfurDioxide_missing <-
  factor(ifelse(is.na(wine$FreeSulfurDioxide), 0, 1))
winemissing$TotalSulfurDioxide_missing <-
  factor(ifelse(is.na(wine$TotalSulfurDioxide), 0, 1))
winemissing$pH_missing <- factor(ifelse(is.na(wine$pH), 0, 1))
winemissing$Sulphates_missing <- factor(ifelse(is.na(wine$Sulphates), 0, 1))
winemissing$Alcohol_missing <- factor(ifelse(is.na(wine$Alcohol), 0, 1))
winemissing$STARS_missing <- factor(ifelse(is.na(wine$STARS),0,1))

winemelt <- melt(winemissing, id.vars=colnames(winemissing)[1:16],
                 variable.name = 'Measure')

winemelt$value <- factor(winemelt$value)

ggplot(winemelt, aes(x=value, y=TARGET)) + geom_violin() +
  facet_wrap( ~ Measure, scales = 'free') + theme_tufte()

ggplot(wine, aes(x=factor(STARS), y=TARGET)) + geom_violin() + theme_tufte()
```

Distribution of Variables:

```r
ggplot(wine, aes(x=TARGET)) + geom_histogram() + theme_tufte()

winecatmelt <- melt(wine, id.vars=colnames(wine)[1:13])

ggplot(winecatmelt, aes(x=value)) + geom_histogram() +
  facet_wrap( ~ variable, scales='free') + theme_tufte()

winenummelt <- melt(wine, id.vars=colnames(wine)[c(1,2,14:16)])

ggplot(winenummelt, aes(x=value)) + geom_histogram(bins=100) +
  facet_wrap( ~ variable, scales='free') + theme_tufte()
```

Data Transformation:

```r
wine$STARS <- as.character(wine$STARS)
```

```r
wine[is.na(wine$STARS),'STARS'] <- 0

wine$STARS <- factor(wine$STARS)
```

Diagnostic Table Creation:

```r
wineAINum <- na.omit(wine) %>% select(-INDEX)
wineAICat <- na.omit(wine) %>% select(-INDEX)
wineAICat$AcidIndex <- factor(wineAICat$AcidIndex)

fullpoismod1 <- glm(TARGET ~ ., data=wineAINum, family=poisson)
backpoismod1 <- step(fullpoismod1, trace=0)

fullpoismod2 <- glm(TARGET ~ ., data=wineAICat, family=poisson)
backpoismod2 <- step(fullpoismod2, trace=0)

#with(backNBmod1, cbind(res.deviance = deviance, df = df.residual,
#  p = pchisq(deviance, df.residual, lower.tail=FALSE)))

backNBmod1 <- glm.nb(formula(backpoismod1), data=wineAINum)
backNBmod2 <- glm.nb(formula(backpoismod2), data=wineAICat)

backlinmod1 <- lm(formula(backpoismod1), data=wineAINum)
backlinmod2 <- lm(formula(backpoismod2), data=wineAICat)

modelList <- c('backpoismod1', 'backpoismod2', 'backNBmod1', 'backNBmod2',
               'backlinmod1', 'backlinmod2')

coefdiag <- data.frame(Variable = character(0), Estimate = numeric(0),
                       StdError = numeric(0), t.value = numeric(0),
                       Pr.t = numeric(0), Model = character(0))

for(mod in modelList){
  moddf <- data.frame(summary(get(mod))$coefficients)
  moddf <- data.frame(row.names(moddf), moddf, row.names=NULL)
  moddf$model <- mod
  colnames(moddf) <- c('Variable', 'Estimate', 'StdError', 't.value', 'Pr.t', 'Model')
  coefdiag <- rbind(coefdiag, moddf)
}

coefdiag <- melt(coefdiag, id.vars = c('Variable', 'Model'),
                 variable.name = 'Measure', value.name = 'Value')

coefdiag$Model <- factor(coefdiag$Model)

modelDic <- data.frame(Model = modelList,
                       ModelName = c('Poisson Model Numeric','Poisson Model Categorical',
                                     'Negative Binomial Model Numeric',
                                     'Negative Binomial Model Categorical',
                                     'Linear Model Numeric', 'Linear Model Categorical'))

coefdiag <- merge(coefdiag, modelDic, by='Model')
```

```r
estimates1 <- filter(coefdiag, Measure == 'Estimate') %>%
  filter(Model %in% c('backlinmod1', 'backNBmod1', 'backpoismod1')) %>%
  select(-c(Measure, Model))

estimates1 <- dcast(estimates1, Variable ~ ModelName, value.var = 'Value')

estimates2 <- filter(coefdiag, Measure == 'Estimate') %>%
  filter(Model %in% c('backlinmod2', 'backNBmod2', 'backpoismod2')) %>%
  select(-c(Measure, Model))

estimates2 <- dcast(estimates2, Variable ~ ModelName, value.var = 'Value')

MSEList <- c(
  mean((wineAINum$TARGET - exp(predict(backpoismod1)))^2),
  mean((wineAINum$TARGET - exp(predict(backNBmod1)))^2),
  mean((wineAINum$TARGET - predict(backlinmod1))^2),
  mean((wineAICat$TARGET - exp(predict(backpoismod2)))^2),
  mean((wineAICat$TARGET - exp(predict(backNBmod2)))^2),
  mean((wineAICat$TARGET - predict(backlinmod2))^2)
)

SEList <- c(
  sd((wineAINum$TARGET - exp(predict(backpoismod1)))^2)/sqrt(length(wineAINum$TARGET)),
  sd((wineAINum$TARGET - exp(predict(backNBmod1)))^2)/sqrt(length(wineAINum$TARGET)),
  sd((wineAINum$TARGET - predict(backlinmod1))^2)/sqrt(length(wineAINum$TARGET)),
  sd((wineAICat$TARGET - exp(predict(backpoismod2)))^2)/sqrt(length(wineAINum$TARGET)),
  sd((wineAICat$TARGET - exp(predict(backNBmod2)))^2)/sqrt(length(wineAINum$TARGET)),
  sd((wineAICat$TARGET - predict(backlinmod1))^2)/sqrt(length(wineAICat$TARGET))
)

modelDiag <- data.frame(Model = modelList, MSE = MSEList, SE = SEList)
modelDiag <- merge(modelDiag, modelDic, by='Model')
modelDiag <- select(modelDiag, -Model)
modelDiag <- select(modelDiag, ModelName, MSE, SE)

pvalues1 <- filter(coefdiag, Measure == 'Pr.t') %>%
  filter(Model %in% c('backlinmod1', 'backNBmod1', 'backpoismod1')) %>%
  select(-c(Measure, Model))

pvalues1 <- dcast(pvalues1, Variable ~ ModelName, value.var = 'Value')

pvalues2 <- filter(coefdiag, Measure == 'Pr.t') %>%
  filter(Model %in% c('backlinmod2', 'backNBmod2', 'backpoismod2')) %>%
  select(-c(Measure, Model))

pvalues2 <- dcast(pvalues2, Variable ~ ModelName, value.var = 'Value')
```