Data 621 - HW5

Chi Pong, Euclid Zhang, Jie Zou, Joseph Connolly, LeTicia Cancel

5/3/2022

Loading of Libraries

```
train_df <- read.csv("wine-training-data.csv",fileEncoding="UTF-8-BOM")
test_df <- read.csv("wine-evaluation-data.csv",fileEncoding="UTF-8-BOM")

train_df$INDEX <- NULL
test_df$IN <- NULL
```

DATA EXPLORATION

Data Summary

```
summary(train_df)
```

```
##      TARGET       FixedAcidity      VolatileAcidity     CitricAcid
##  Min.   :0.000   Min.   :-18.100   Min.   :-2.7900   Min.   :-3.2400
##  1st Qu.:2.000   1st Qu.:  5.200   1st Qu.: 0.1300   1st Qu.: 0.0300
##  Median :3.000   Median :  6.900   Median : 0.2800   Median : 0.3100
##  Mean   :3.029   Mean   :  7.076   Mean   : 0.3241   Mean   : 0.3084
##  3rd Qu.:4.000   3rd Qu.:  9.500   3rd Qu.: 0.6400   3rd Qu.: 0.5800
##  Max.   :8.000   Max.   : 34.400   Max.   : 3.6800   Max.   : 3.8600
##
##  ResidualSugar        Chlorides       FreeSulfurDioxide TotalSulfurDioxide
##  Min.   :-127.800   Min.   :-1.1710   Min.   :-555.00   Min.   :-823.0
##  1st Qu.:  -2.000   1st Qu.:-0.0310   1st Qu.:   0.00   1st Qu.:  27.0
##  Median :   3.900   Median : 0.0460   Median :  30.00   Median : 123.0
##  Mean   :   5.419   Mean   : 0.0548   Mean   :  30.85   Mean   : 120.7
##  3rd Qu.:  15.900   3rd Qu.: 0.1530   3rd Qu.:  70.00   3rd Qu.: 208.0
##  Max.   : 141.150   Max.   : 1.3510   Max.   : 623.00   Max.   :1057.0
##  NA's   :616        NA's   :638       NA's   :647       NA's   :682
##     Density             pH          Sulphates          Alcohol
##  Min.   :0.8881   Min.   :0.480   Min.   :-3.1300   Min.   :-4.70
##  1st Qu.:0.9877   1st Qu.:2.960   1st Qu.: 0.2800   1st Qu.: 9.00
##  Median :0.9945   Median :3.200   Median : 0.5000   Median :10.40
##  Mean   :0.9942   Mean   :3.208   Mean   : 0.5271   Mean   :10.49
##  3rd Qu.:1.0005   3rd Qu.:3.470   3rd Qu.: 0.8600   3rd Qu.:12.40
```

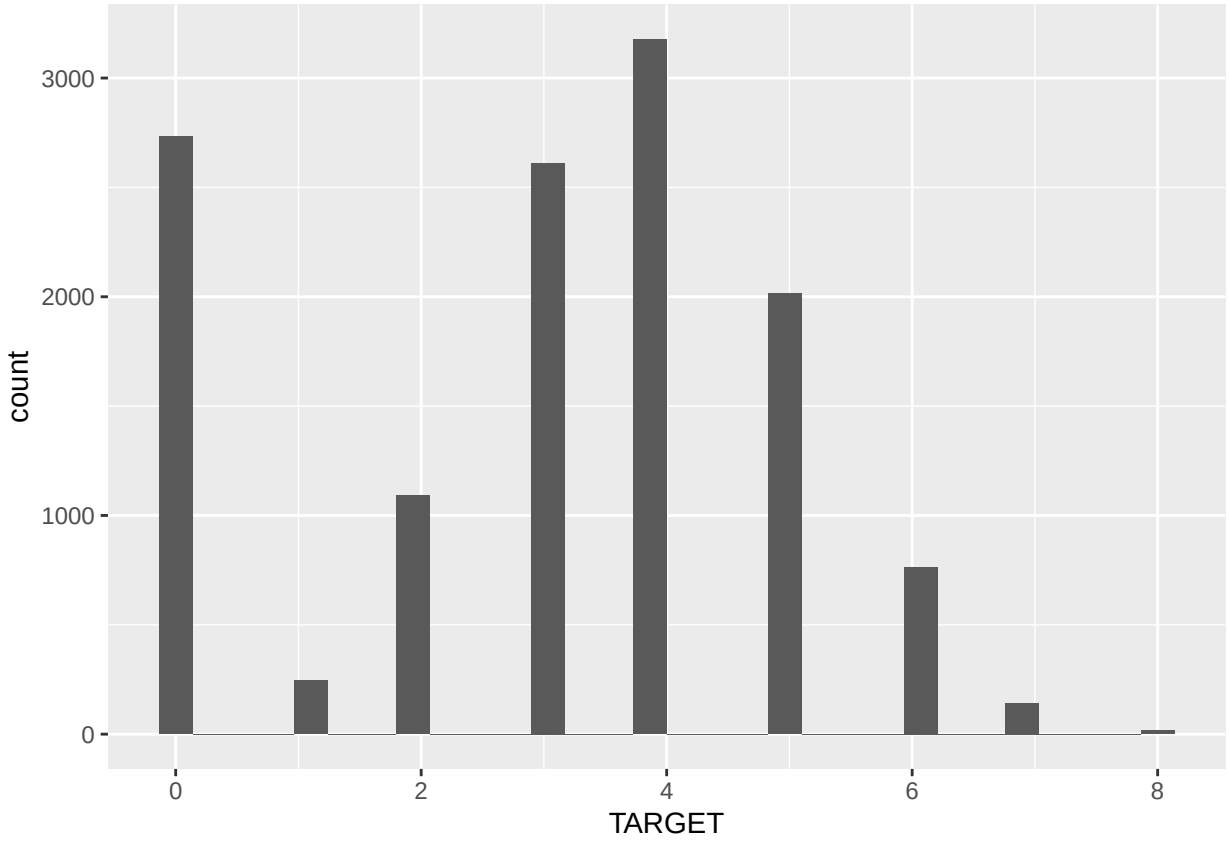
```
##  Max.   :1.0992   Max.   :6.130   Max.   : 4.2400   Max.   :26.50
##                   NA's   :395     NA's   :1210      NA's   :653
##   LabelAppeal          AcidIndex          STARS
##  Min.   :-2.000000   Min.   : 4.000   Min.   :1.000
##  1st Qu.:-1.000000   1st Qu.: 7.000   1st Qu.:1.000
##  Median : 0.000000   Median : 8.000   Median :2.000
##  Mean   :-0.009066   Mean   : 7.773   Mean   :2.042
##  3rd Qu.: 1.000000   3rd Qu.: 8.000   3rd Qu.:3.000
##  Max.   : 2.000000   Max.   :17.000   Max.   :4.000
##                                       NA's   :3359
```

From the summary:

- We can see that most of the chemical properties range from a negative value to a positive value of similar magnitude. These predictor variables seem to be already scaled / standardized. Hence, there is no extreme outliers.
- **ResidualSugar**, **Chlorides**, **FreeSulfurDioxide**, **TotalSulfurDioxide**, **pH**, **Sulphates**, **Alcohol** have numerous missing values. We will impute the missing values using mice (multivariate imputation by chained equations).
- **STARS** also has missing values. However, the values are missing simply because they don't have a rating, not because of data collecting problems. We may consider imputing this variable differently.

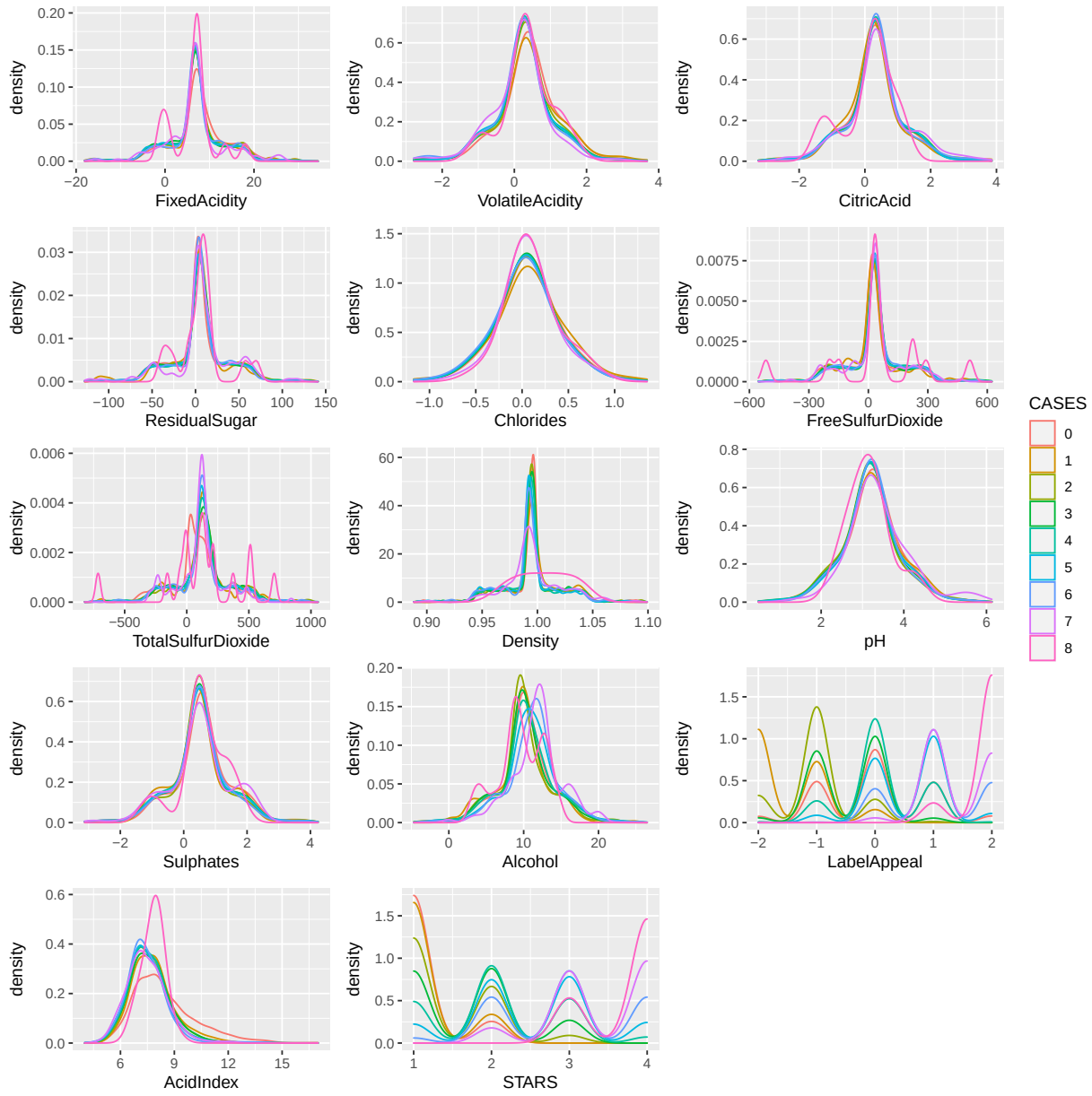
Distribution plots

```
ggplot(train_df, aes(x=TARGET)) + geom_histogram(na.rm =TRUE, bins=30)
```



From above plot, we see that the target value is **zero-inflated**, not a regular poisson distribution nor any distribution of the exponential family.
Hence, practically it is not suggested to fit the data to a poisson, negative binomial or linear model. Since we're tasked with fitting those models, we'll show the steps to demonstrate how everything works.

```
CASES <- as.factor(train_df$TARGET)

plot_FixedAcidity <- ggplot(train_df, aes(x=FixedAcidity, color=CASES)) + geom_density(na.rm =TRUE, bw=
plot_VolatileAcidity <- ggplot(train_df, aes(x=VolatileAcidity, color=CASES)) + geom_density(na.rm =TRU
plot_CitricAcid <- ggplot(train_df, aes(x=CitricAcid, color=CASES)) + geom_density(na.rm =TRUE, bw=0.3)
plot_ResidualSugar <- ggplot(train_df, aes(x=ResidualSugar, color=CASES)) + geom_density(na.rm =TRUE, bw
plot_Chlorides <- ggplot(train_df, aes(x=Chlorides, color=CASES)) + geom_density(na.rm =TRUE, bw=0.2)
plot_FreeSulfurDioxide <- ggplot(train_df, aes(x=FreeSulfurDioxide, color=CASES)) + geom_density(na.rm =
plot_TotalSulfurDioxide <- ggplot(train_df, aes(x=TotalSulfurDioxide, color=CASES)) + geom_density(na.rm
plot_Density <- ggplot(train_df, aes(x=Density, color=CASES)) + geom_density(na.rm =TRUE)
plot_pH <- ggplot(train_df, aes(x=pH, color=CASES)) + geom_density(na.rm =TRUE, bw=0.3)
plot_Sulphates <- ggplot(train_df, aes(x=Sulphates, color=CASES)) + geom_density(na.rm =TRUE, bw=0.3)
plot_Alcohol <- ggplot(train_df, aes(x=Alcohol, color=CASES)) + geom_density(na.rm =TRUE, bw=0.8)
plots_LabelAppeal <- ggplot(train_df, aes(x=LabelAppeal, color=CASES)) + geom_density(na.rm =TRUE, bw=0
plots_AcidIndex <- ggplot(train_df, aes(x=AcidIndex, color=CASES)) + geom_density(na.rm =TRUE, bw=0.5)
plots_STARS <- ggplot(train_df, aes(x=STARS, color=CASES)) + geom_density(na.rm =TRUE, bw=0.2)

plot_FixedAcidity+plot_VolatileAcidity+plot_CitricAcid+plot_ResidualSugar+plot_Chlorides+
  plot_FreeSulfurDioxide+plot_TotalSulfurDioxide+plot_Density+plot_pH+plot_Sulphates+
  plot_Alcohol+plots_LabelAppeal+plots_AcidIndex+plots_STARS+
  plot_layout(ncol = 3, guides = "collect")
```

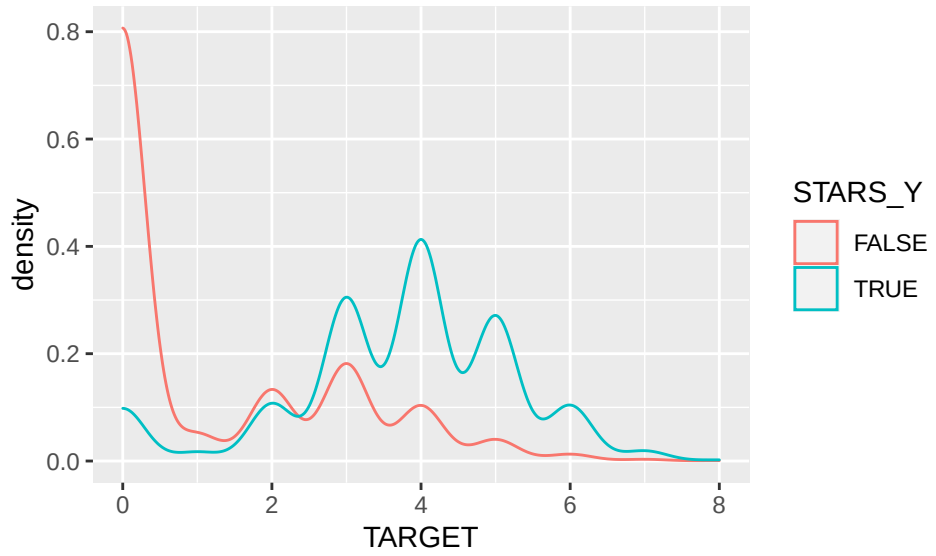


The distributions of the predictor variables show that **LabelAppeal** and **STARS** are good candidates of predicting the target variable. The distributions of other variables do not vary a lot based on the different values of the target variable.

```
ResidualSugar_Y <- !is.na(train_df$ResidualSugar)
Chlorides_Y <- !is.na(train_df$Chlorides)
FreeSulfurDioxide_Y <- !is.na(train_df$FreeSulfurDioxide)
TotalSulfurDioxide_Y <- !is.na(train_df$TotalSulfurDioxide)
pH_Y <- !is.na(train_df$pH)
Sulphates_Y <- !is.na(train_df$Sulphates)
Alcohol_Y <- !is.na(train_df$Alcohol)
STARS_Y <- !is.na(train_df$STARS)
```

Now, let's check whether **STARS** is missing or not have an effect to the cases of wine purchased.

```
ggplot(train_df, aes(x=TARGET, color=STARS_Y)) + geom_density(na.rm =TRUE, bw=0.3)
```

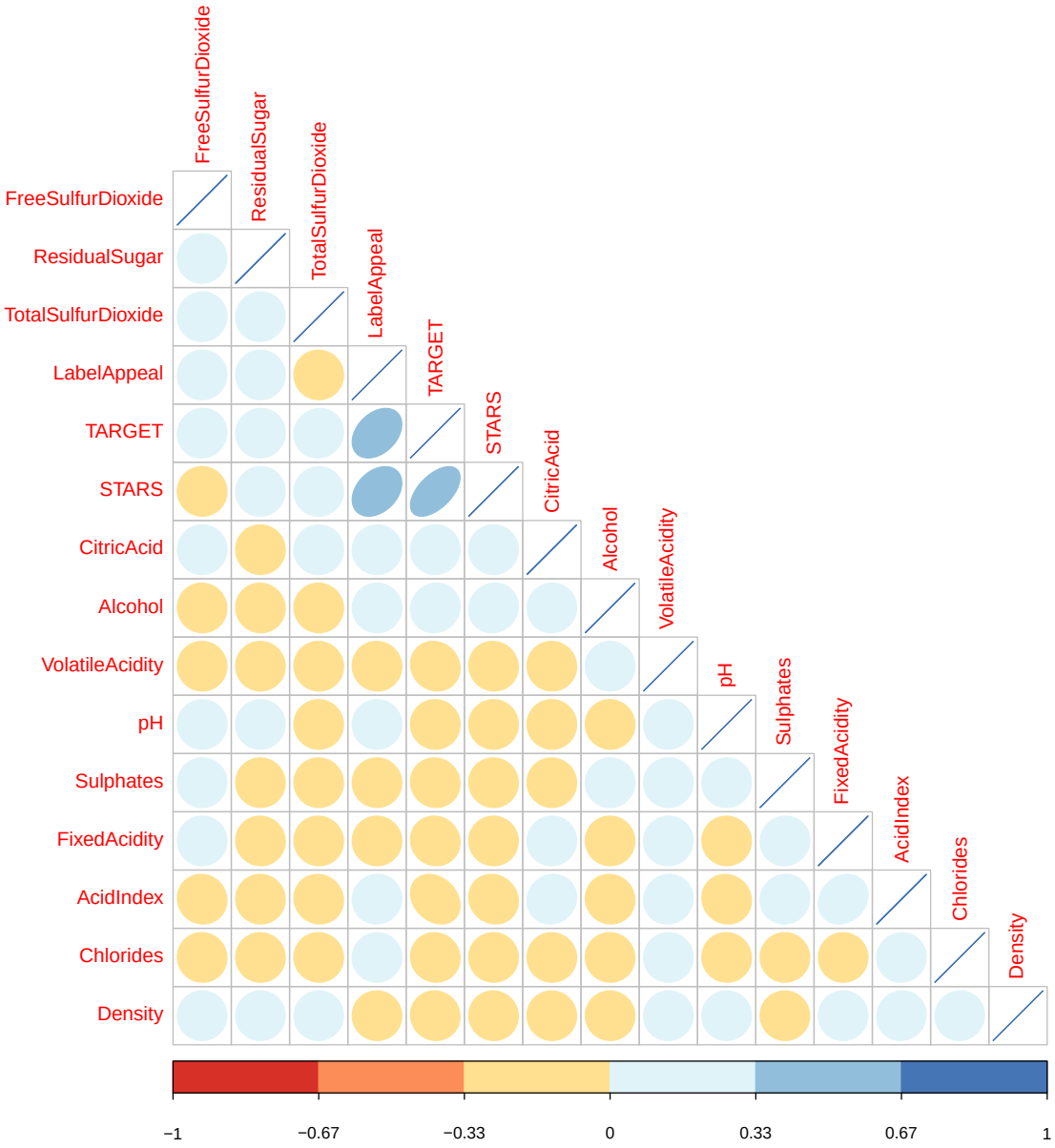


The distributions plot above indicates that most people are willing to buy wines with **STARS** provided and not willing to buy wines with **STARS** unavailable. We may add a dummy variable, or transform the **STARS** variable to indicate **STARS** is available or not.

Multi-collinearity

The best way to check for multi-collinearity is to use correlation coefficients among variables, or predictors.

```
# corrplot::corrplot(cor(train_df, use = "na.or.complete"),
#                    method = 'number', type = 'lower', diag = FALSE, tl.srt = 0.1)
correlation = cor(train_df, use = 'pairwise.complete.obs')
corrplot::corrplot(correlation, method = 'ellipse', type = 'lower',  order = 'hclust', col=brewer.pal(n=
```



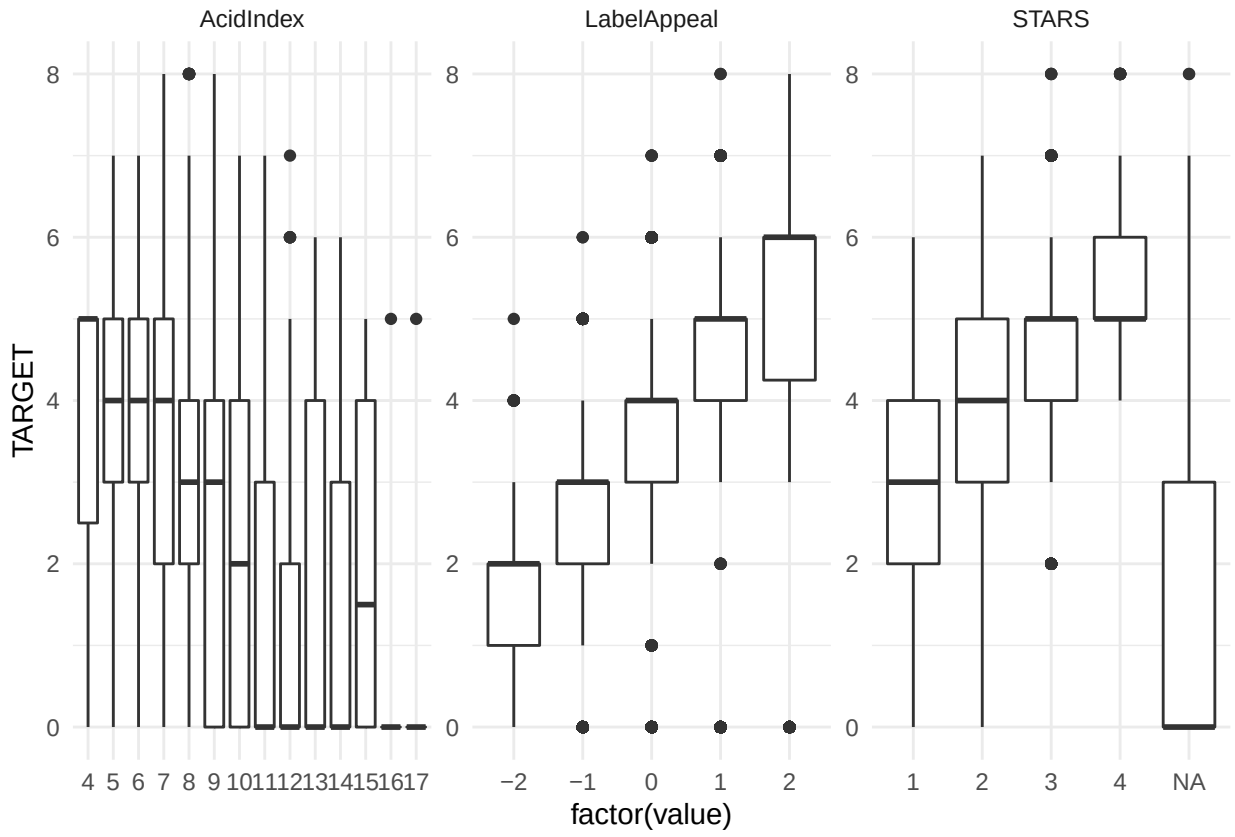
The correlation coefficients among predictors are quite low. With that said, we checked all the assumptions for linear regressions.

DATA PREPARATION

Boxplots

```
df_pivot_wide <- train_df %>%
  dplyr::select(STARS, LabelAppeal, AcidIndex, TARGET ) %>%
  pivot_longer(cols = -TARGET, names_to="variable", values_to="value") %>%
  arrange(variable, value)
```

```
df_pivot_wide %>%
  ggplot(mapping = aes(x = factor(value), y = TARGET)) +
    geom_boxplot() +
    facet_wrap(.~variable, scales="free") +
    theme_minimal()
```



Commentaries:

There aren't too many outliners for AcidIndex. You can tell there are a lot of zeros for AcidIndex 12, 16, and 17. There is no clear pattern in relation to TARGET. As for LabelAppeal, I do see there is positive correlation with TARGET. The higher the LabelAppeal, the higher volume of TARGET you get. As for STARS, there is an obvious positive correlation with TARGET. TARGET = NA seems to be distribute across all spectrum of STARS. In order to satisfy some of the requirements for the model, I'd impute NA with 0. The overall trend with the existing values is still the same where the higher the value of STARS will naturally net a higher volume in TARGET, which is cases of wine sold.

Data Imputation

For imputing the missing values of the chemical properties, the following variables are not included as predictors:

- **TARGET**: the target variable should not be used to predict the missing values of the predictors, as the objective of the models is to predict the target variables using the predictors.

- **LabelAppeal**: the label appeal of the bottle should not have anything to do with the chemical properties of the wines.
- **STARS**: More than 25% of the wines have missing STARS. Whether it is missing or not should not have anything to do with the chemical properties of the wines.

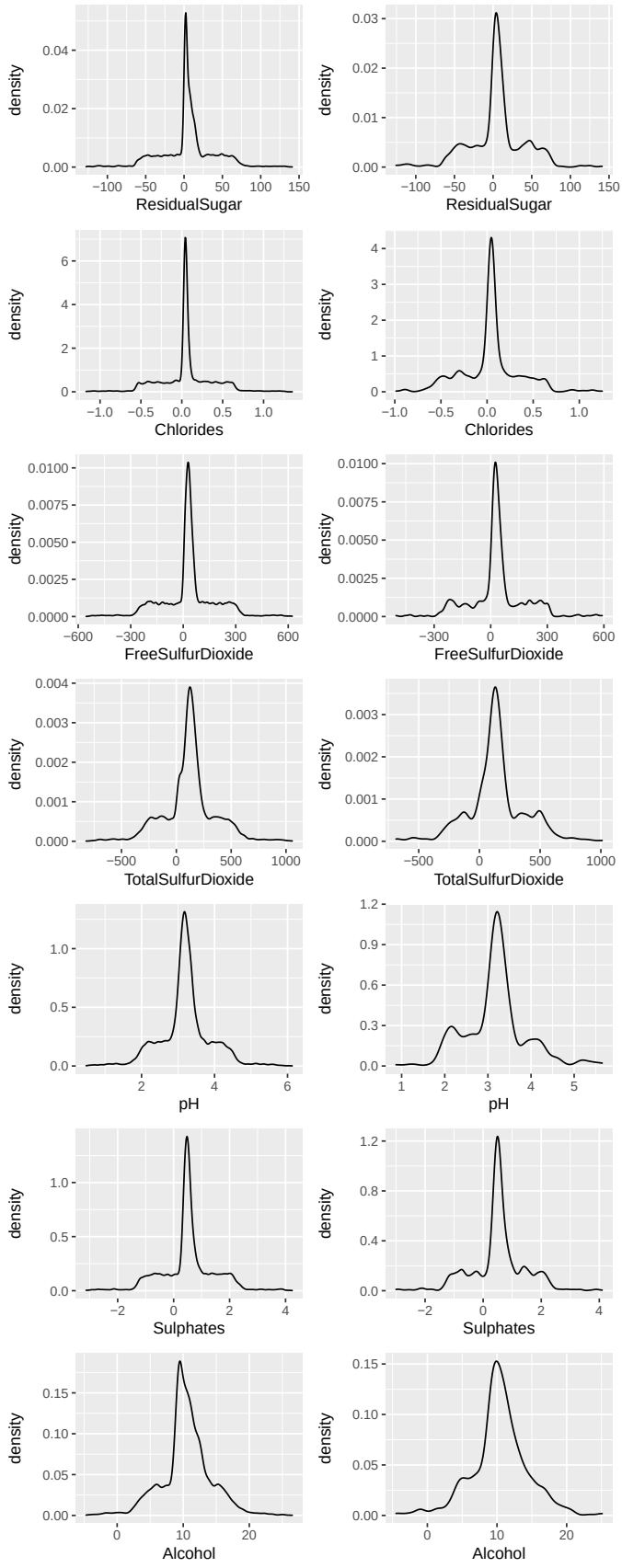
Multivariate Imputation by Chained Equations (MICE) is used to impute the missing values

```
#temporary exclude TARGET, LabelAppeal, and STARS in our imputation
TARGET <- train_df$TARGET
LabelAppeal <- train_df$LabelAppeal
STARS <- train_df$STARS

train_df$TARGET <- NULL
train_df$LabelAppeal <- NULL
train_df$STARS <- NULL

#save the imputation models to impute the test data set later
mickey <- parlmice(train_df, maxit = 5, m = 1, printFlag = FALSE, seed = 2022,
                   cluster.seed = 2022)

#save the imputation result
train_df <- complete(mickey,1)

#Add TARGET, LabelAppeal, and STARS back to our dataframe
train_df$TARGET <- TARGET
train_df$LabelAppeal <- LabelAppeal
train_df$STARS <- STARS

TARGET <- NULL
LabelAppeal <- NULL
STARS <- NULL
```

We can compare the imputed data values and the original data values.
The plots on the left below show the distributions of the values from the original data.
The plots on the right below show the distributions of the imputed values.

```
plot_ResidualSugar <- ggplot(train_df[ResidualSugar_Y,], aes(x=ResidualSugar)) +
  geom_density(na.rm =TRUE)
plot_Chlorides <- ggplot(train_df[Chlorides_Y,], aes(x=Chlorides)) +
  geom_density(na.rm =TRUE)
plot_FreeSulfurDioxide <- ggplot(train_df[FreeSulfurDioxide_Y,], aes(x=FreeSulfurDioxide)) +
  geom_density(na.rm =TRUE)
plot_TotalSulfurDioxide <- ggplot(train_df[TotalSulfurDioxide_Y,], aes(x=TotalSulfurDioxide)) +
  geom_density(na.rm =TRUE)
plot_pH <- ggplot(train_df[pH_Y,], aes(x=pH)) +
  geom_density(na.rm =TRUE)
plot_Sulphates <- ggplot(train_df[Sulphates_Y,], aes(x=Sulphates)) +
  geom_density(na.rm =TRUE)
plot_Alcohol <- ggplot(train_df[Alcohol_Y,], aes(x=Alcohol)) +
  geom_density(na.rm =TRUE)

plot_ResidualSugar2 <- ggplot(train_df[!ResidualSugar_Y,], aes(x=ResidualSugar)) +
  geom_density(na.rm =TRUE)
```

```r
plot_Chlorides2 <- ggplot(train_df[!Chlorides_Y,], aes(x=Chlorides)) +
  geom_density(na.rm =TRUE)
plot_FreeSulfurDioxide2 <- ggplot(train_df[!FreeSulfurDioxide_Y,], aes(x=FreeSulfurDioxide)) +
  geom_density(na.rm =TRUE)
plot_TotalSulfurDioxide2 <- ggplot(train_df[!TotalSulfurDioxide_Y,], aes(x=TotalSulfurDioxide)) +
  geom_density(na.rm =TRUE)
plot_pH2 <- ggplot(train_df[!pH_Y,], aes(x=pH)) +
  geom_density(na.rm =TRUE)
plot_Sulphates2 <- ggplot(train_df[!Sulphates_Y,], aes(x=Sulphates)) +
  geom_density(na.rm =TRUE)
plot_Alcohol2 <- ggplot(train_df[!Alcohol_Y,], aes(x=Alcohol)) +
  geom_density(na.rm =TRUE)

plot_ResidualSugar+plot_ResidualSugar2+
  plot_Chlorides+plot_Chlorides2+
  plot_FreeSulfurDioxide+plot_FreeSulfurDioxide2+
  plot_TotalSulfurDioxide+plot_TotalSulfurDioxide2+
  plot_pH+plot_pH2+
  plot_Sulphates+plot_Sulphates2+
  plot_Alcohol+plot_Alcohol2+
  plot_layout(ncol = 2, guides = "collect")
```


density
ResidualSugar
density
ResidualSugar
density
Chlorides
density
Chlorides
density
FreeSulfurDioxide
density
FreeSulfurDioxide
density
TotalSulfurDioxide
density
TotalSulfurDioxide
density
pH
density
pH
density
Sulphates
density
Sulphates
density
Alcohol
density
Alcohol

The distributions look similar and so the imputed values are plausible.

Data Transformation

As discussed above, whether **STARS** is available or not is predictive of the target. Moreover, the marginal effect of increasing 1 star may not be equal. For example, the effect from 1 star to 2 star may not be the same as the effect from 4 star to 5 star. Hence, we will impute the missing values of **STARS** by 0 and convert **STARS** to a factor variable. The variable will then be converted to 4 dummies variables in the models.

Similarly, we will also convert **LabelAppeal** to a factor variable as the marginal effects may change.

```
train_df$STARS[!STARS_Y] <- 0
train_df$STARS <- as.factor(train_df$STARS)
train_df$LabelAppeal <- as.factor(train_df$LabelAppeal)
```

BUILD MODELS

Poisson models

We start building our Poisson model with all predictors.

```
poisson_full <- glm(TARGET ~ ., data=train_df, family=poisson)
summary(poisson_full)
```

```
##
## Call:
## glm(formula = TARGET ~ ., family = poisson, data = train_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2287  -0.6546  -0.0045   0.4509   3.7781
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          6.837e-01  1.990e-01   3.435 0.000591 ***
## FixedAcidity         5.313e-05  8.200e-04   0.065 0.948339
## VolatileAcidity     -3.044e-02  6.530e-03  -4.662 3.13e-06 ***
## CitricAcid           5.086e-03  5.899e-03   0.862 0.388548
## ResidualSugar        8.016e-05  1.504e-04   0.533 0.594151
## Chlorides           -3.442e-02  1.606e-02  -2.143 0.032146 *
## FreeSulfurDioxide    8.272e-05  3.423e-05   2.417 0.015661 *
## TotalSulfurDioxide   7.082e-05  2.207e-05   3.209 0.001330 **
## Density             -2.563e-01  1.918e-01  -1.336 0.181389
## pH                  -1.144e-02  7.525e-03  -1.520 0.128477
## Sulphates           -1.037e-02  5.506e-03  -1.883 0.059666 .
## Alcohol              3.720e-03  1.368e-03   2.720 0.006535 **
## AcidIndex           -7.971e-02  4.573e-03 -17.431  < 2e-16 ***
## LabelAppeal-1        2.355e-01  3.799e-02   6.199 5.67e-10 ***
## LabelAppeal0         4.262e-01  3.705e-02  11.502  < 2e-16 ***
## LabelAppeal1         5.584e-01  3.769e-02  14.814  < 2e-16 ***
```

```
## LabelAppeal2        6.965e-01  4.243e-02  16.413  < 2e-16 ***
## STARS1              7.664e-01  1.954e-02  39.219  < 2e-16 ***
## STARS2              1.086e+00  1.824e-02  59.525  < 2e-16 ***
## STARS3              1.205e+00  1.920e-02  62.786  < 2e-16 ***
## STARS4              1.325e+00  2.431e-02  54.486  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 22861  on 12794  degrees of freedom
## Residual deviance: 13641  on 12774  degrees of freedom
## AIC: 45625
##
## Number of Fisher Scoring iterations: 6
```

Backward Elimination by AIC

Starting with our full model, perform backward elimination by comparing the **AIC** of the models. Note that K is the multiple of the number of degrees of freedom used for the penalty. K = 2 achieves the same outcome as k not being passed any value.

```
poisson_AIC <- step(poisson_full,trace=0)
summary(poisson_AIC)
```

```
##
## Call:
## glm(formula = TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide +
##     TotalSulfurDioxide + pH + Sulphates + Alcohol + AcidIndex +
##     LabelAppeal + STARS, family = poisson, data = train_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2316  -0.6554  -0.0043   0.4486   3.7693
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         4.302e-01  6.109e-02   7.042 1.89e-12 ***
## VolatileAcidity    -3.066e-02  6.529e-03  -4.696 2.65e-06 ***
## Chlorides          -3.516e-02  1.606e-02  -2.190  0.02853 *
## FreeSulfurDioxide   8.288e-05  3.422e-05   2.422  0.01542 *
## TotalSulfurDioxide  7.036e-05  2.205e-05   3.191  0.00142 **
## pH                 -1.135e-02  7.524e-03  -1.508  0.13146
## Sulphates          -1.040e-02  5.504e-03  -1.889  0.05885 .
## Alcohol             3.742e-03  1.368e-03   2.736  0.00622 **
## AcidIndex          -7.966e-02  4.515e-03 -17.645  < 2e-16 ***
## LabelAppeal-1       2.354e-01  3.799e-02   6.197 5.76e-10 ***
## LabelAppeal0        4.262e-01  3.705e-02  11.502  < 2e-16 ***
## LabelAppeal1        5.585e-01  3.769e-02  14.819  < 2e-16 ***
## LabelAppeal2        6.961e-01  4.243e-02  16.406  < 2e-16 ***
## STARS1              7.666e-01  1.954e-02  39.233  < 2e-16 ***
## STARS2              1.086e+00  1.823e-02  59.580  < 2e-16 ***
## STARS3              1.206e+00  1.920e-02  62.824  < 2e-16 ***
```

```
## STARS4              1.325e+00  2.431e-02  54.526  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 22861  on 12794  degrees of freedom
## Residual deviance: 13644  on 12778  degrees of freedom
## AIC: 45620
##
## Number of Fisher Scoring iterations: 6
```

Backward Elimination by BIC

Starting with our full model, perform backward elimination by comparing the **BIC** of the models. Note that k = 2 gives the genuine AIC and k = log(n) gives you BIC.

```
poisson_BIC <- step(poisson_full,trace=0, k=log(nrow(train_df)))
summary(poisson_BIC)
```

```
##
## Call:
## glm(formula = TARGET ~ VolatileAcidity + TotalSulfurDioxide +
##     AcidIndex + LabelAppeal + STARS, family = poisson, data = train_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2431  -0.6537  -0.0059   0.4551   3.8098
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         4.342e-01  5.235e-02   8.293  < 2e-16 ***
## VolatileAcidity    -3.082e-02  6.529e-03  -4.720 2.35e-06 ***
## TotalSulfurDioxide  7.018e-05  2.204e-05   3.184  0.00145 **
## AcidIndex          -8.048e-02  4.496e-03 -17.898  < 2e-16 ***
## LabelAppeal-1       2.345e-01  3.798e-02   6.174 6.64e-10 ***
## LabelAppeal0        4.249e-01  3.705e-02  11.469  < 2e-16 ***
## LabelAppeal1        5.566e-01  3.768e-02  14.770  < 2e-16 ***
## LabelAppeal2        6.950e-01  4.243e-02  16.380  < 2e-16 ***
## STARS1              7.687e-01  1.953e-02  39.352  < 2e-16 ***
## STARS2              1.088e+00  1.823e-02  59.718  < 2e-16 ***
## STARS3              1.210e+00  1.917e-02  63.126  < 2e-16 ***
## STARS4              1.330e+00  2.427e-02  54.815  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 22861  on 12794  degrees of freedom
## Residual deviance: 13667  on 12783  degrees of freedom
## AIC: 45633
##
## Number of Fisher Scoring iterations: 6
```

Negative Binomial models

We start building our Negative Binomial model with all predictors.

Because the data is zero inflated. the glm.nb function is not able to find the optimal value for the additional parameter r. Since the density is highest at target = 0, we will build our model using r = 1.

```
nb_full <- glm(TARGET ~ ., data=train_df,negative.binomial(1))
summary(nb_full)
```

```
##
## Call:
## glm(formula = TARGET ~ ., family = negative.binomial(1), data = train_df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.90412  -0.34155  -0.01171   0.21572   2.02710
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.057e+00  2.383e-01   4.436 9.25e-06 ***
## FixedAcidity        -1.551e-04  9.885e-04  -0.157  0.87529
## VolatileAcidity     -4.178e-02  7.867e-03  -5.311 1.11e-07 ***
## CitricAcid           6.534e-03  7.140e-03   0.915  0.36017
## ResidualSugar        1.835e-04  1.815e-04   1.011  0.31201
## Chlorides           -4.725e-02  1.937e-02  -2.439  0.01473 *
## FreeSulfurDioxide    1.275e-04  4.142e-05   3.077  0.00209 **
## TotalSulfurDioxide   1.218e-04  2.656e-05   4.587 4.55e-06 ***
## Density             -2.833e-01  2.317e-01  -1.223  0.22147
## pH                  -2.419e-02  9.058e-03  -2.670  0.00759 **
## Sulphates           -1.679e-02  6.633e-03  -2.531  0.01140 *
## Alcohol              2.556e-03  1.646e-03   1.553  0.12050
## AcidIndex           -1.132e-01  5.147e-03 -21.994  < 2e-16 ***
## LabelAppeal-1        2.218e-01  3.658e-02   6.063 1.37e-09 ***
## LabelAppeal0         3.905e-01  3.567e-02  10.946  < 2e-16 ***
## LabelAppeal1         4.910e-01  3.691e-02  13.303  < 2e-16 ***
## LabelAppeal2         6.318e-01  4.617e-02  13.685  < 2e-16 ***
## STARS1               7.580e-01  1.878e-02  40.352  < 2e-16 ***
## STARS2               1.088e+00  1.794e-02  60.652  < 2e-16 ***
## STARS3               1.217e+00  2.009e-02  60.566  < 2e-16 ***
## STARS4               1.349e+00  3.017e-02  44.694  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1) family taken to be 0.3426981)
##
##     Null deviance: 9042.5  on 12794  degrees of freedom
## Residual deviance: 6475.9  on 12774  degrees of freedom
## AIC: 55249
##
## Number of Fisher Scoring iterations: 5
```

Backward Elimination by AIC

Starting with our full model, perform backward elimination by comparing the **AIC** of the models.

```
nb_AIC <- step(nb_full,trace=0)
summary(nb_AIC)
```

```
##
## Call:
## glm(formula = TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide +
##     TotalSulfurDioxide + pH + Sulphates + Alcohol + AcidIndex +
##     LabelAppeal + STARS, family = negative.binomial(1), data = train_df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.90641  -0.33963  -0.01073   0.21652   2.02222
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          7.769e-01  6.547e-02  11.868  < 2e-16 ***
## VolatileAcidity     -4.217e-02  7.865e-03  -5.362 8.37e-08 ***
## Chlorides           -4.833e-02  1.936e-02  -2.496   0.0126 *
## FreeSulfurDioxide    1.280e-04  4.141e-05   3.091   0.0020 **
## TotalSulfurDioxide   1.220e-04  2.655e-05   4.597 4.34e-06 ***
## pH                  -2.398e-02  9.056e-03  -2.648   0.0081 **
## Sulphates           -1.682e-02  6.629e-03  -2.537   0.0112 *
## Alcohol              2.556e-03  1.646e-03   1.553   0.1205
## AcidIndex           -1.133e-01  5.068e-03 -22.353  < 2e-16 ***
## LabelAppeal-1        2.220e-01  3.658e-02   6.069 1.33e-09 ***
## LabelAppeal0         3.906e-01  3.567e-02  10.952  < 2e-16 ***
## LabelAppeal1         4.914e-01  3.690e-02  13.315  < 2e-16 ***
## LabelAppeal2         6.317e-01  4.616e-02  13.684  < 2e-16 ***
## STARS1               7.581e-01  1.878e-02  40.365  < 2e-16 ***
## STARS2               1.089e+00  1.793e-02  60.727  < 2e-16 ***
## STARS3               1.217e+00  2.009e-02  60.604  < 2e-16 ***
## STARS4               1.349e+00  3.016e-02  44.731  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1) family taken to be 0.3426476)
##
##     Null deviance: 9042.5  on 12794  degrees of freedom
## Residual deviance: 6477.1  on 12778  degrees of freedom
## AIC: 55242
##
## Number of Fisher Scoring iterations: 5
```

Backward Elimination by BIC

Starting with our full model, perform backward elimination by comparing the **BIC** of the models.

```
nb_BIC <- step(nb_full,trace=0, k=log(nrow(train_df)))
summary(nb_BIC)
```

```
##
## Call:
## glm(formula = TARGET ~ VolatileAcidity + FreeSulfurDioxide +
##     TotalSulfurDioxide + AcidIndex + LabelAppeal + STARS, family = negative.binomial(1),
##     data = train_df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.91942  -0.34091  -0.01183   0.21507   2.06096
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          7.158e-01  5.378e-02  13.311  < 2e-16 ***
## VolatileAcidity     -4.215e-02  7.866e-03  -5.358 8.55e-08 ***
## FreeSulfurDioxide    1.276e-04  4.140e-05   3.082  0.00206 **
## TotalSulfurDioxide   1.217e-04  2.655e-05   4.584 4.60e-06 ***
## AcidIndex           -1.133e-01  5.052e-03 -22.421  < 2e-16 ***
## LabelAppeal-1        2.212e-01  3.658e-02   6.046 1.53e-09 ***
## LabelAppeal0         3.887e-01  3.567e-02  10.898  < 2e-16 ***
## LabelAppeal1         4.888e-01  3.690e-02  13.247  < 2e-16 ***
## LabelAppeal2         6.278e-01  4.617e-02  13.598  < 2e-16 ***
## STARS1               7.599e-01  1.878e-02  40.466  < 2e-16 ***
## STARS2               1.090e+00  1.793e-02  60.807  < 2e-16 ***
## STARS3               1.221e+00  2.007e-02  60.856  < 2e-16 ***
## STARS4               1.353e+00  3.013e-02  44.905  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1) family taken to be 0.3429317)
##
##     Null deviance: 9042.5  on 12794  degrees of freedom
## Residual deviance: 6484.6  on 12782  degrees of freedom
## AIC: 55242
##
## Number of Fisher Scoring iterations: 5
```

Multiple Linear Regression Models

We start building our Multiple Linear Regression model with all predictors.

```
lm_full <- lm(TARGET ~ ., data=train_df)
summary(lm_full)
```

```
##
## Call:
## lm(formula = TARGET ~ ., data = train_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -4.9661 -0.8616  0.0247  0.8423  6.1850
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         2.831e+00  4.467e-01   6.337 2.42e-10 ***
## FixedAcidity        5.976e-04  1.859e-03   0.321  0.74791
## VolatileAcidity    -9.449e-02  1.478e-02  -6.392 1.69e-10 ***
## CitricAcid          1.724e-02  1.344e-02   1.282  0.19974
## ResidualSugar       2.465e-04  3.417e-04   0.721  0.47071
## Chlorides          -1.123e-01  3.637e-02  -3.089  0.00201 **
## FreeSulfurDioxide   2.433e-04  7.807e-05   3.117  0.00183 **
## TotalSulfurDioxide  1.966e-04  4.990e-05   3.940 8.18e-05 ***
## Density            -7.923e-01  4.359e-01  -1.817  0.06917 .
## pH                 -2.877e-02  1.701e-02  -1.691  0.09079 .
## Sulphates          -2.690e-02  1.244e-02  -2.161  0.03069 *
## Alcohol             1.231e-02  3.097e-03   3.976 7.04e-05 ***
## AcidIndex          -2.001e-01  9.102e-03 -21.983  < 2e-16 ***
## LabelAppeal-1       3.607e-01  6.287e-02   5.736 9.91e-09 ***
## LabelAppeal0        8.285e-01  6.131e-02  13.513  < 2e-16 ***
## LabelAppeal1        1.292e+00  6.404e-02  20.177  < 2e-16 ***
## LabelAppeal2        1.882e+00  8.437e-02  22.306  < 2e-16 ***
## STARS1              1.364e+00  3.293e-02  41.407  < 2e-16 ***
## STARS2              2.399e+00  3.202e-02  74.935  < 2e-16 ***
## STARS3              2.966e+00  3.706e-02  80.036  < 2e-16 ***
## STARS4              3.650e+00  5.926e-02  61.583  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.306 on 12774 degrees of freedom
## Multiple R-squared:  0.5409, Adjusted R-squared:  0.5402
## F-statistic: 752.6 on 20 and 12774 DF,  p-value: < 2.2e-16
```

Backward Elimination by AIC

Starting with our full model, perform backward elimination by comparing the **AIC** of the models.

```
lm_AIC <- step(lm_full,trace=0)
summary(lm_AIC)
```

```
##
## Call:
## lm(formula = TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide +
##     TotalSulfurDioxide + Density + pH + Sulphates + Alcohol +
##     AcidIndex + LabelAppeal + STARS, data = train_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9667 -0.8621  0.0247  0.8432  6.1794
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         2.839e+00  4.467e-01   6.355 2.15e-10 ***
## VolatileAcidity    -9.488e-02  1.478e-02  -6.420 1.41e-10 ***
```

```
## Chlorides          -1.131e-01  3.636e-02  -3.110  0.00187 **
## FreeSulfurDioxide   2.452e-04  7.805e-05   3.141  0.00169 **
## TotalSulfurDioxide  1.974e-04  4.989e-05   3.957 7.64e-05 ***
## Density            -7.998e-01  4.359e-01  -1.835  0.06653 .
## pH                 -2.872e-02  1.701e-02  -1.688  0.09135 .
## Sulphates          -2.712e-02  1.244e-02  -2.180  0.02927 *
## Alcohol             1.233e-02  3.096e-03   3.982 6.86e-05 ***
## AcidIndex          -1.988e-01  8.945e-03 -22.226  < 2e-16 ***
## LabelAppeal-1       3.604e-01  6.287e-02   5.732 1.02e-08 ***
## LabelAppeal0        8.280e-01  6.131e-02  13.507  < 2e-16 ***
## LabelAppeal1        1.292e+00  6.403e-02  20.172  < 2e-16 ***
## LabelAppeal2        1.882e+00  8.436e-02  22.309  < 2e-16 ***
## STARS1              1.364e+00  3.292e-02  41.421  < 2e-16 ***
## STARS2              2.401e+00  3.200e-02  75.008  < 2e-16 ***
## STARS3              2.967e+00  3.706e-02  80.062  < 2e-16 ***
## STARS4              3.651e+00  5.925e-02  61.617  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.306 on 12777 degrees of freedom
## Multiple R-squared:  0.5408, Adjusted R-squared:  0.5402
## F-statistic: 885.3 on 17 and 12777 DF,  p-value: < 2.2e-16
```

Backward Elimination by BIC

Starting with our full model, perform backward elimination by comparing the **BIC** of the models.

```
lm_BIC <- step(lm_full,trace=0, k=log(nrow(train_df)))
summary(lm_BIC)
```

```
##
## Call:
## lm(formula = TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide +
##     TotalSulfurDioxide + Alcohol + AcidIndex + LabelAppeal +
##     STARS, data = train_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0281 -0.8632  0.0247  0.8391  6.2010
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.936e+00  1.003e-01  19.307  < 2e-16 ***
## VolatileAcidity    -9.546e-02  1.478e-02  -6.458 1.10e-10 ***
## Chlorides          -1.128e-01  3.636e-02  -3.103  0.00192 **
## FreeSulfurDioxide   2.412e-04  7.806e-05   3.090  0.00201 **
## TotalSulfurDioxide  1.973e-04  4.990e-05   3.954 7.73e-05 ***
## Alcohol             1.233e-02  3.096e-03   3.981 6.90e-05 ***
## AcidIndex          -1.991e-01  8.919e-03 -22.321  < 2e-16 ***
## LabelAppeal-1       3.614e-01  6.289e-02   5.746 9.34e-09 ***
## LabelAppeal0        8.293e-01  6.132e-02  13.524  < 2e-16 ***
## LabelAppeal1        1.293e+00  6.405e-02  20.180  < 2e-16 ***
## LabelAppeal2        1.880e+00  8.438e-02  22.283  < 2e-16 ***
```

```
## STARS1              1.366e+00  3.293e-02  41.485  < 2e-16 ***
## STARS2              2.403e+00  3.200e-02  75.089  < 2e-16 ***
## STARS3              2.971e+00  3.704e-02  80.212  < 2e-16 ***
## STARS4              3.654e+00  5.926e-02  61.665  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.307 on 12780 degrees of freedom
## Multiple R-squared:  0.5405, Adjusted R-squared:  0.5399
## F-statistic:  1074 on 14 and 12780 DF,  p-value: < 2.2e-16
```

Model Coefficients Comparison

Now, let's compare the results of our Poisson, Negative Binomial, and Linear models

```
poisson_full_coef <- data.frame(poisson_full=poisson_full$coefficients)
poisson_AIC_coef <- data.frame(poisson_AIC=round(poisson_AIC$coefficients,4))
poisson_BIC_coef <- data.frame(poisson_BIC=round(poisson_BIC$coefficients,4))
nb_AIC_coef <- data.frame(nb_AIC=round(nb_AIC$coefficients,4))
nb_BIC_coef <- data.frame(nb_BIC=round(nb_BIC$coefficients,4))
lm_AIC_coef <- data.frame(lm_AIC=round(lm_AIC$coefficients,4))
lm_BIC_coef <- data.frame(lm_BIC=round(lm_BIC$coefficients,4))
```

```
summary_table <- merge(x=poisson_full_coef, y=poisson_AIC_coef,
                       by="row.names", all=TRUE)
summary_table <- merge(x=summary_table, y=poisson_BIC_coef,
                       by.x="Row.names", by.y = "row.names", all=TRUE)
summary_table <- merge(x=summary_table, y=nb_AIC_coef,
                       by.x="Row.names", by.y="row.names", all=TRUE)
summary_table <- merge(x=summary_table, y=nb_BIC_coef,
                       by.x="Row.names", by.y="row.names", all=TRUE)
summary_table <- merge(x=summary_table, y=lm_AIC_coef,
                       by.x="Row.names", by.y="row.names", all=TRUE)
summary_table <- merge(x=summary_table, y=lm_BIC_coef,
                       by.x="Row.names", by.y="row.names", all=TRUE)
summary_table$poisson_full <- NULL
summary_table
```

```
##             Row.names poisson_AIC poisson_BIC  nb_AIC  nb_BIC  lm_AIC  lm_BIC
## 1         (Intercept)      0.4302      0.4342  0.7769  0.7158  2.8386  1.9365
## 2           AcidIndex     -0.0797     -0.0805 -0.1133 -0.1133 -0.1988 -0.1991
## 3             Alcohol      0.0037          NA  0.0026      NA  0.0123  0.0123
## 4           Chlorides     -0.0352          NA -0.0483      NA -0.1131 -0.1128
## 5         CitricAcid          NA          NA      NA      NA      NA      NA
## 6             Density          NA          NA      NA      NA -0.7998      NA
## 7       FixedAcidity          NA          NA      NA      NA      NA      NA
## 8   FreeSulfurDioxide      0.0001          NA  0.0001  0.0001  0.0002  0.0002
## 9       LabelAppeal-1      0.2354      0.2345  0.2220  0.2212  0.3604  0.3614
## 10       LabelAppeal0      0.4262      0.4249  0.3906  0.3887  0.8280  0.8293
## 11       LabelAppeal1      0.5585      0.5566  0.4914  0.4888  1.2917  1.2926
## 12       LabelAppeal2      0.6961      0.6950  0.6317  0.6278  1.8820  1.8803
## 13                 pH     -0.0113          NA -0.0240      NA -0.0287      NA
```

```
## 14      ResidualSugar          NA          NA      NA      NA      NA      NA
## 15              STARS1      0.7666      0.7687  0.7581  0.7599  1.3638  1.3660
## 16              STARS2      1.0863      1.0885  1.0891  1.0905  2.4006  2.4031
## 17              STARS3      1.2060      1.2103  1.2175  1.2211  2.9670  2.9714
## 18              STARS4      1.3254      1.3302  1.3493  1.3530  3.6510  3.6543
## 19          Sulphates     -0.0104          NA -0.0168      NA -0.0271      NA
## 20 TotalSulfurDioxide      0.0001      0.0001  0.0001  0.0001  0.0002  0.0002
## 21    VolatileAcidity     -0.0307     -0.0308 -0.0422 -0.0421 -0.0949 -0.0955
```

- Both **STARS** and **LabelAppeal** have positive effect in all models.
- The coefficients of **STARS** and **LabelAppeal** are very close in the poisson and negative binomial models.
- **TotalSulfurDioxide** has positive effect in all models. The coefficients seem small but the scale of **TotalSulfurDioxide** is more than 100 times larger than the scales of most other variables.
- **CitricAcid**, **FixedAcidity**, **ResidualSugar** are not significant in all models.
- **AcidIndex** and **VolatileAcidity** have negative effect in all models.
- **Alcohol** and **FreeSulfurDioxide** have positive or no effect in all models.
- **Chlorides**, **Density**, **pH**, and **Sulphates** have negative or no effect in all models.

As discussed above, the target variable is zero inflated. It should be better to fit the data in a Hurdle model or a zero-inflated model.

Hurdle Model

```
model_hurdle <- hurdle(TARGET~.-FixedAcidity-Density-CitricAcid-ResidualSugar-Chlorides, data=train_df)
summary(model_hurdle)
```

```
##
## Call:
## hurdle(formula = TARGET ~ . - FixedAcidity - Density - CitricAcid - ResidualSugar -
##     Chlorides, data = train_df)
##
## Pearson residuals:
##       Min        1Q    Median        3Q       Max
## -2.093543 -0.443575 -0.003091  0.395108  4.561259
##
## Count model coefficients (truncated poisson with log link):
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          3.605e-01  7.049e-02   5.113 3.16e-07 ***
## VolatileAcidity     -1.070e-02  6.914e-03  -1.547 0.121773
## FreeSulfurDioxide    2.520e-05  3.556e-05   0.708 0.478671
## TotalSulfurDioxide -2.494e-05  2.259e-05  -1.104 0.269603
## pH                   7.987e-03  7.944e-03   1.005 0.314681
## Sulphates            5.227e-04  5.811e-03   0.090 0.928339
## Alcohol              7.207e-03  1.437e-03   5.015 5.30e-07 ***
## AcidIndex           -1.648e-02  4.934e-03  -3.340 0.000838 ***
## LabelAppeal-1        5.397e-01  4.973e-02  10.853  < 2e-16 ***
## LabelAppeal0         8.433e-01  4.880e-02  17.279  < 2e-16 ***
## LabelAppeal1         1.040e+00  4.937e-02  21.073  < 2e-16 ***
## LabelAppeal2         1.201e+00  5.318e-02  22.582  < 2e-16 ***
## STARS1               4.937e-02  2.142e-02   2.305 0.021153 *
```

```
## STARS2              1.643e-01  1.997e-02   8.228  < 2e-16 ***
## STARS3              2.554e-01  2.091e-02  12.213  < 2e-16 ***
## STARS4              3.578e-01  2.588e-02  13.826  < 2e-16 ***
## Zero hurdle model coefficients (binomial with logit link):
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         4.298e+00  2.754e-01  15.606  < 2e-16 ***
## VolatileAcidity    -1.839e-01  3.646e-02  -5.044 4.56e-07 ***
## FreeSulfurDioxide   5.552e-04  1.955e-04   2.840  0.00450 **
## TotalSulfurDioxide  7.866e-04  1.238e-04   6.354 2.10e-10 ***
## pH                 -1.783e-01  4.179e-02  -4.266 1.99e-05 ***
## Sulphates          -8.433e-02  3.049e-02  -2.766  0.00568 **
## Alcohol            -1.900e-02  7.642e-03  -2.486  0.01292 *
## AcidIndex          -3.893e-01  2.141e-02 -18.187  < 2e-16 ***
## LabelAppeal-1      -4.862e-01  1.371e-01  -3.547  0.00039 ***
## LabelAppeal0       -9.055e-01  1.338e-01  -6.767 1.32e-11 ***
## LabelAppeal1       -1.449e+00  1.433e-01 -10.109  < 2e-16 ***
## LabelAppeal2       -1.816e+00  2.214e-01  -8.202 2.37e-16 ***
## STARS1              1.825e+00  6.133e-02  29.757  < 2e-16 ***
## STARS2              4.260e+00  1.170e-01  36.401  < 2e-16 ***
## STARS3              2.024e+01  3.635e+02   0.056  0.95561
## STARS4              2.039e+01  6.944e+02   0.029  0.97657
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 23
## Log-likelihood: -2.03e+04 on 32 Df
```

Zero Inflation Model 1 (default: poisson distribution)

```
# Adding dist = "poisson" is the same as without providing such argument
# model_zeroinfl <- zeroinfl(TARGET~.-FixedAcidity-Density-CitricAcid-ResidualSugar-Chlorides, data=tra
model_zeroinfl1 <- zeroinfl(TARGET~.-FixedAcidity-Density-CitricAcid-ResidualSugar-Chlorides, data=trai

summary(model_zeroinfl1)
```

```
##
## Call:
## zeroinfl(formula = TARGET ~ . - FixedAcidity - Density - CitricAcid -
##     ResidualSugar - Chlorides, data = train_df)
##
## Pearson residuals:
##      Min       1Q   Median       3Q      Max
## -2.26457 -0.42897  0.00214  0.38107  5.36112
##
## Count model coefficients (poisson with log link):
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         4.960e-01  6.406e-02   7.742 9.76e-15 ***
## VolatileAcidity    -1.243e-02  6.707e-03  -1.853  0.06395 .
## FreeSulfurDioxide   2.123e-05  3.455e-05   0.614  0.53901
## TotalSulfurDioxide -1.393e-05  2.194e-05  -0.635  0.52532
## pH                  5.535e-03  7.711e-03   0.718  0.47284
## Sulphates           7.949e-04  5.645e-03   0.141  0.88802
```

```
## Alcohol             6.799e-03  1.394e-03   4.876 1.08e-06 ***
## AcidIndex          -1.909e-02  4.833e-03  -3.949 7.84e-05 ***
## LabelAppeal-1       4.403e-01  4.133e-02  10.652  < 2e-16 ***
## LabelAppeal0        7.288e-01  4.041e-02  18.036  < 2e-16 ***
## LabelAppeal1        9.184e-01  4.107e-02  22.360  < 2e-16 ***
## LabelAppeal2        1.076e+00  4.558e-02  23.605  < 2e-16 ***
## STARS1              6.115e-02  2.113e-02   2.894  0.00381 **
## STARS2              1.831e-01  1.975e-02   9.270  < 2e-16 ***
## STARS3              2.812e-01  2.067e-02  13.603  < 2e-16 ***
## STARS4              3.787e-01  2.562e-02  14.784  < 2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -6.179e+00  4.475e-01 -13.807  < 2e-16 ***
## VolatileAcidity     1.867e-01  4.353e-02   4.289 1.80e-05 ***
## FreeSulfurDioxide  -7.330e-04  2.358e-04  -3.109  0.00188 **
## TotalSulfurDioxide -8.782e-04  1.480e-04  -5.934 2.96e-09 ***
## pH                  2.087e-01  4.998e-02   4.175 2.97e-05 ***
## Sulphates           1.137e-01  3.645e-02   3.120  0.00181 **
## Alcohol             2.601e-02  9.189e-03   2.830  0.00465 **
## AcidIndex           4.317e-01  2.569e-02  16.801  < 2e-16 ***
## LabelAppeal-1       1.506e+00  3.320e-01   4.536 5.73e-06 ***
## LabelAppeal0        2.266e+00  3.296e-01   6.874 6.23e-12 ***
## LabelAppeal1        2.973e+00  3.350e-01   8.876  < 2e-16 ***
## LabelAppeal2        3.416e+00  3.858e-01   8.857  < 2e-16 ***
## STARS1             -2.085e+00  7.616e-02 -27.370  < 2e-16 ***
## STARS2             -5.736e+00  3.276e-01 -17.509  < 2e-16 ***
## STARS3             -2.024e+01  3.404e+02  -0.059  0.95260
## STARS4             -2.039e+01  6.400e+02  -0.032  0.97458
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 39
## Log-likelihood: -2.035e+04 on 32 Df
```

Zero Inflation Model 2 (negative binomial distribution)

```
model_zeroinfl2 <- zeroinfl(TARGET~.-FixedAcidity-Density-CitricAcid-ResidualSugar-Chlorides, data=trai
```

```
## Warning in sqrt(diag(vc)[np]): NaNs produced
```

```
summary(model_zeroinfl2)
```

```
##
## Call:
## zeroinfl(formula = TARGET ~ . - FixedAcidity - Density - CitricAcid -
##     ResidualSugar - Chlorides, data = train_df, dist = "negbin")
##
## Pearson residuals:
##       Min        1Q    Median        3Q       Max
## -2.264590 -0.428975  0.002136  0.381078  5.361372
```

```
##
## Count model coefficients (negbin with log link):
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          4.960e-01  6.406e-02   7.743 9.71e-15 ***
## VolatileAcidity     -1.242e-02  6.707e-03  -1.853  0.06395 .
## FreeSulfurDioxide    2.123e-05  3.455e-05   0.614  0.53902
## TotalSulfurDioxide -1.393e-05  2.194e-05  -0.635  0.52541
## pH                   5.533e-03  7.711e-03   0.718  0.47302
## Sulphates            7.950e-04  5.645e-03   0.141  0.88801
## Alcohol              6.799e-03  1.394e-03   4.876 1.08e-06 ***
## AcidIndex           -1.909e-02  4.833e-03  -3.949 7.84e-05 ***
## LabelAppeal-1        4.403e-01  4.133e-02  10.652  < 2e-16 ***
## LabelAppeal0         7.287e-01  4.041e-02  18.035  < 2e-16 ***
## LabelAppeal1         9.184e-01  4.107e-02  22.360  < 2e-16 ***
## LabelAppeal2         1.076e+00  4.558e-02  23.605  < 2e-16 ***
## STARS1               6.115e-02  2.113e-02   2.894  0.00381 **
## STARS2               1.831e-01  1.975e-02   9.270  < 2e-16 ***
## STARS3               2.812e-01  2.067e-02  13.603  < 2e-16 ***
## STARS4               3.787e-01  2.562e-02  14.784  < 2e-16 ***
## Log(theta)           1.733e+01        NaN     NaN      NaN
##
## Zero-inflation model coefficients (binomial with logit link):
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -6.179e+00  4.475e-01 -13.808  < 2e-16 ***
## VolatileAcidity      1.867e-01  4.353e-02   4.289 1.79e-05 ***
## FreeSulfurDioxide   -7.330e-04  2.358e-04  -3.109  0.00188 **
## TotalSulfurDioxide -8.781e-04  1.480e-04  -5.933 2.97e-09 ***
## pH                   2.087e-01  4.998e-02   4.176 2.97e-05 ***
## Sulphates            1.137e-01  3.645e-02   3.120  0.00181 **
## Alcohol              2.601e-02  9.189e-03   2.831  0.00464 **
## AcidIndex            4.317e-01  2.569e-02  16.801  < 2e-16 ***
## LabelAppeal-1        1.506e+00  3.320e-01   4.536 5.72e-06 ***
## LabelAppeal0         2.266e+00  3.296e-01   6.874 6.24e-12 ***
## LabelAppeal1         2.973e+00  3.350e-01   8.876  < 2e-16 ***
## LabelAppeal2         3.416e+00  3.858e-01   8.856  < 2e-16 ***
## STARS1              -2.085e+00  7.616e-02 -27.370  < 2e-16 ***
## STARS2              -5.736e+00  3.277e-01 -17.506  < 2e-16 ***
## STARS3              -2.024e+01  3.406e+02  -0.059  0.95261
## STARS4              -2.039e+01  6.401e+02  -0.032  0.97459
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 33573273.5323
## Number of iterations in BFGS optimization: 47
## Log-likelihood: -2.035e+04 on 33 Df
```

SELECT MODELS

Root Mean Squared Error

As we've a regression model, the best metric to evaluate the performance is Root Mean Squared Error (RMSE).

poisson_AIC	poisson_BIC	nb_AIC	nb_BIC	lm_AIC	lm_BIC	model_hurdle	model_zeroinfl1	model_z
1.300907	1.302221	1.325879	1.325339	1.305269	1.305832	**1.262711**	1.264324	1

```
model.summary <- data.frame(poisson_AIC=sqrt(mean(residuals(poisson_AIC, type="response")^2)),
           poisson_BIC=sqrt(mean(residuals(poisson_BIC, type="response")^2)),
           nb_AIC=sqrt(mean(residuals(nb_AIC, type="response")^2)),
           nb_BIC=sqrt(mean(residuals(nb_BIC, type="response")^2)),
           lm_AIC=sqrt(mean(residuals(lm_AIC, type="response")^2)),
           lm_BIC=sqrt(mean(residuals(lm_BIC, type="response")^2)),
           model_hurdle=sqrt(mean(residuals(model_hurdle, type="response")^2)),
           model_zeroinfl1=sqrt(mean(residuals(model_zeroinfl1, type="response")^2)),
           model_zeroinfl2=sqrt(mean(residuals(model_zeroinfl2, type="response")^2))
          )
kable(model.summary) %>%
  kable_paper(full_width = F) %>%
  column_spec(7:7, bold = T, color = "white", background = "purple")
```

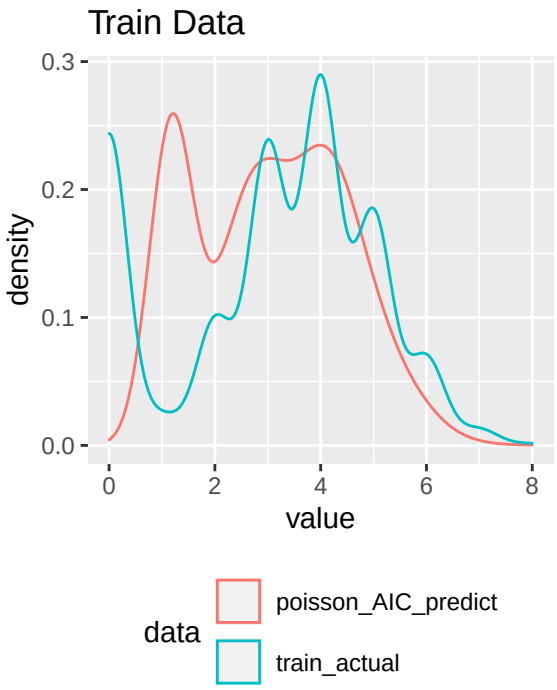
From the RMSE of all models, the hurdle model has the best performance. This is expected since hurdle model is designed for zero-inflated data.

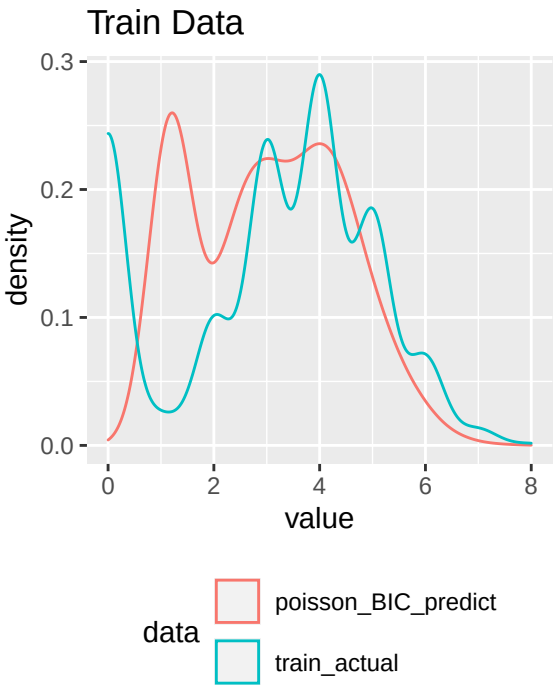
Distribution of Predicted Values (train data)

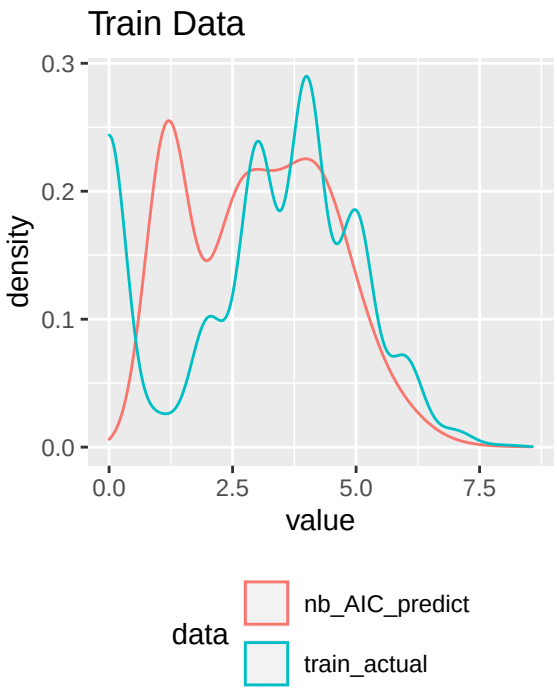
We can also look at the distributions of the model predictions of the training data.

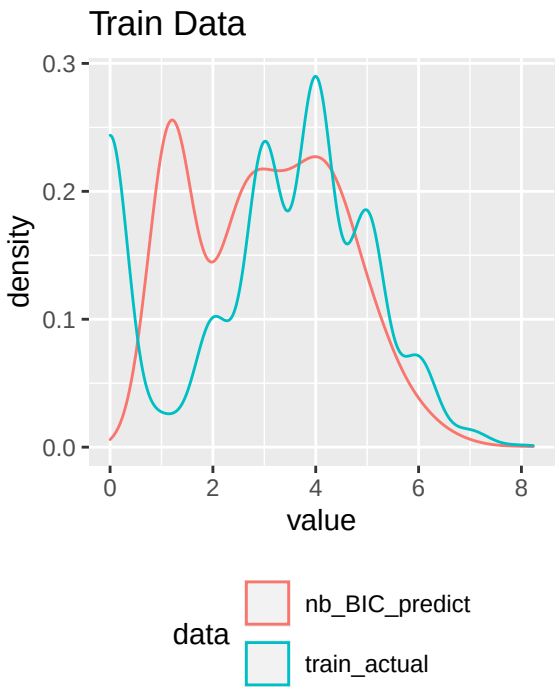
```
train_actual <- train_df$TARGET
poisson_AIC_predict <- predict(poisson_AIC,type="response")
poisson_BIC_predict <- predict(poisson_BIC,type="response")
nb_AIC_predict <- predict(nb_AIC,type="response")
nb_BIC_predict <- predict(nb_BIC,type="response")
lm_AIC_predict <- predict(lm_AIC,type="response")
lm_BIC_predict <- predict(lm_BIC,type="response")
model_hurdle_predict <- predict(model_hurdle,type="response")
model_zeroinfl1_predict <- predict(model_zeroinfl1,type="response")
model_zeroinfl2_predict <- predict(model_zeroinfl2,type="response")


dist_df <- data.frame(rbind(
      cbind(train_actual,"train_actual"),
      cbind(poisson_AIC_predict,"poisson_AIC_predict"),
      cbind(poisson_BIC_predict,"poisson_BIC_predict"),
      cbind(nb_AIC_predict,"nb_AIC_predict"),
      cbind(nb_BIC_predict,"nb_BIC_predict"),
      cbind(lm_AIC_predict,"lm_AIC_predict"),
      cbind(lm_BIC_predict,"lm_BIC_predict"),
      cbind(model_hurdle_predict,"model_hurdle_predict"),
      cbind(model_zeroinfl1_predict,"model_zeroinfl1_predict"),
      cbind(model_zeroinfl2_predict, "model_zeroinfl2_predict")
      ),stringsAsFactors=FALSE)
colnames(dist_df) <- c("value","data")
dist_df$value <- as.numeric(dist_df$value)
```

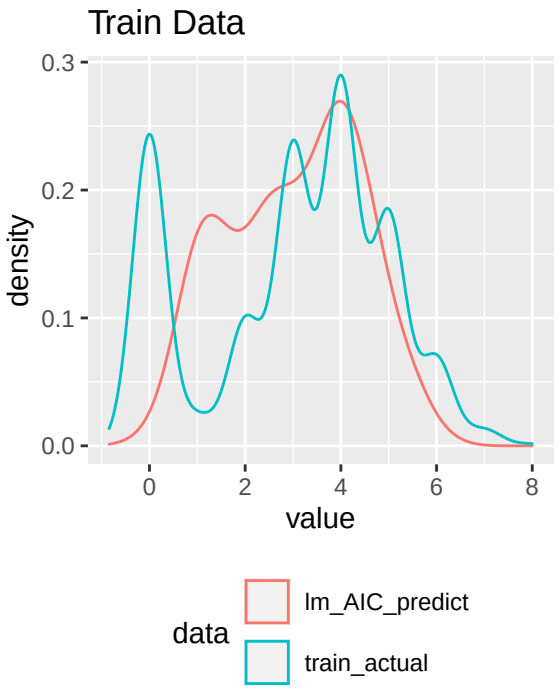
```
models <- unique(dist_df$data)[-1]
for (model in models) {
    plot<-ggplot(dist_df[dist_df$data=="train_actual" | dist_df$data==model,],
           aes(x=value, color=data))+ggtitle("Train Data")+geom_density(bw=0.35)+
           theme(legend.position="bottom")+
           guides(color=guide_legend(nrow=2, byrow=TRUE))
    print(plot)
}
```

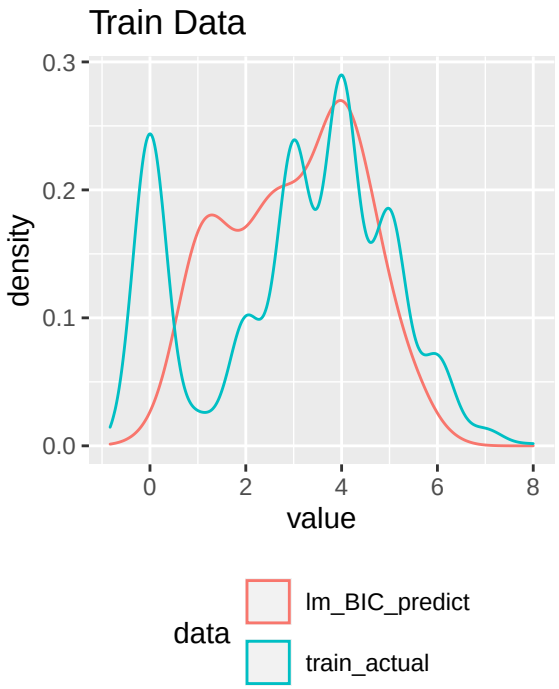


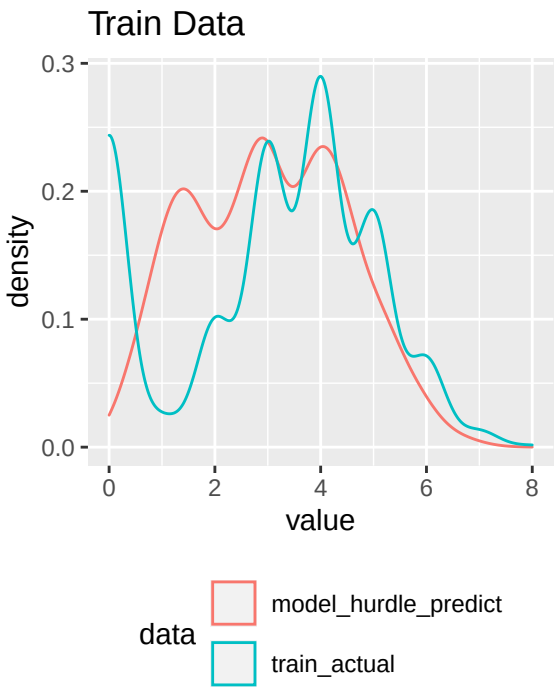


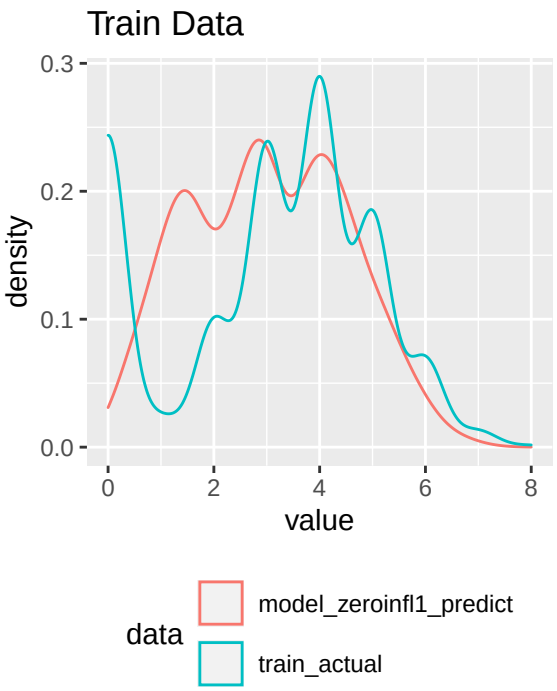

Train Data
density
value
0.0
0.1
0.2
0.3
0.0
2.5
5.0
7.5
data
nb_AIC_predict
train_actual

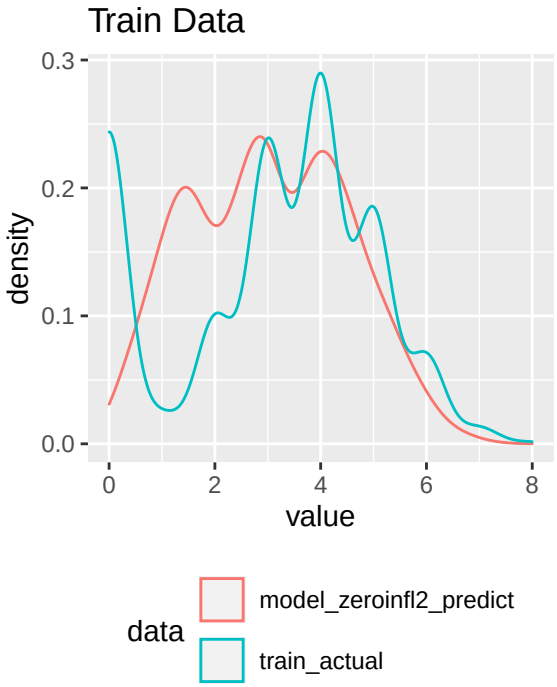

Train Data
density
value
0.0
0.1
0.2
0.3
0
2
4
6
8
data
nb_BIC_predict
train_actual


Train Data
density
value
0.0
0.1
0.2
0.3
0
2
4
6
8
data
lm_AIC_predict
train_actual


Train Data
density
value
0.0
0.1
0.2
0.3
0
2
4
6
8
data
lm_BIC_predict
train_actual







The predictions of Poisson models and Negative Binomial models have similar distribution. They do well in modeling the peak near 0. However, the peak is at 1, there is nearly no prediction of target = 0.

The linear models are the worst, they do even predict some negative values since it is not bounded.

The hurdle model and the zero-inflated models do not model the peak near 0 as well as the Poisson models or Negative Binomial models do. However, they successfully predict some cases with target = 0. Moreover, the models are fitting the data better at target greater or equal to 3.

This confirms our findings above that the hurdle model and the zero-inflated models suit our data better.

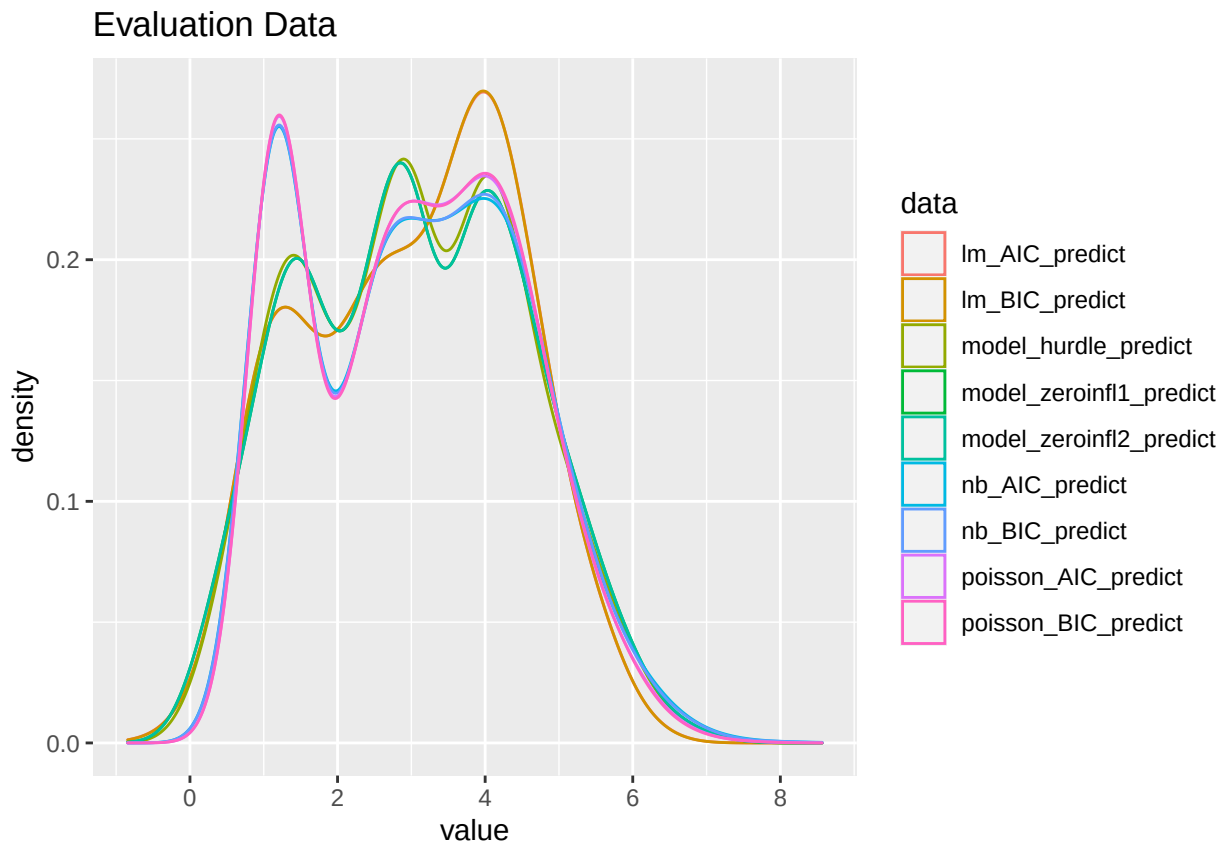
Distribution of Predicted Values (test data or evaluation data)

```
#temporary exclude LabelAppeal and STARS in our imputation
LabelAppeal <- test_df$LabelAppeal
STARS <- test_df$STARS

test_df$TARGET <- NULL
test_df$LabelAppeal <- NULL
test_df$STARS <- NULL

#save the imputation result
test_df <- mice.reuse(mickey, test_df, maxit = 5, printFlag = FALSE, seed = 2022)[[1]]

#Add TARGET, LabelAppeal, and STARS back to our dataframe
test_df$LabelAppeal <- LabelAppeal
test_df$STARS <- STARS

LabelAppeal <- NULL
STARS <- NULL

#data transformation
STARS_Y <- !is.na(test_df$STARS)
test_df$STARS[!STARS_Y] <- 0
test_df$STARS <- as.factor(test_df$STARS)
test_df$LabelAppeal <- as.factor(test_df$LabelAppeal)
```

The following are the distributions of the predicted values of our models using the evaluation data.

```
poisson_AIC_predict <- predict(poisson_AIC,type="response",data=test_df)
poisson_BIC_predict <- predict(poisson_BIC,type="response",data=test_df)
nb_AIC_predict <- predict(nb_AIC,type="response",data=test_df)
nb_BIC_predict <- predict(nb_BIC,type="response",data=test_df)
lm_AIC_predict <- predict(lm_AIC,type="response",data=test_df)
lm_BIC_predict <- predict(lm_BIC,type="response",data=test_df)
model_hurdle_predict <- predict(model_hurdle,type="response",data=test_df)
model_zeroinfl1_predict <- predict(model_zeroinfl1,type="response",data=test_df)
model_zeroinfl2_predict <- predict(model_zeroinfl2,type="response",data=test_df)


dist_df <- data.frame(rbind(
      cbind(poisson_AIC_predict,"poisson_AIC_predict"),
      cbind(poisson_BIC_predict,"poisson_BIC_predict"),
      cbind(nb_AIC_predict,"nb_AIC_predict"),
      cbind(nb_BIC_predict,"nb_BIC_predict"),
      cbind(lm_AIC_predict,"lm_AIC_predict"),
      cbind(lm_BIC_predict,"lm_BIC_predict"),
      cbind(model_hurdle_predict,"model_hurdle_predict"),
      cbind(model_zeroinfl1_predict,"model_zeroinfl1_predict"),
      cbind(model_zeroinfl2_predict,"model_zeroinfl2_predict")
      ),stringsAsFactors=FALSE)
colnames(dist_df) <- c("value","data")
dist_df$value <- as.numeric(dist_df$value)
```

```
ggplot(dist_df, aes(x=value, color=data))+
  ggtitle("Evaluation Data")+geom_density(bw=0.35)
```



The distributions are very close to our predictions using the training data. The predictions produce plausible and acceptable results.