Data 621 Hw #5

Group #3- Coco Donovan, Matthew Roland, Marjete Vucinaj, Jean Jimenez

2024-04-18

Wine Data

Goal: to Build a count regression model to predict the number of cases that will be sold given certain properties of wine.

Packages

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(ggplot2)
library(summarytools)
```

```
## Warning: package 'summarytools' was built under R version 4.3.3
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.3.3
```

```
## corrplot 0.92 loaded
```

```
library(gt)
```

```
## Warning: package 'gt' was built under R version 4.3.3
```

```
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.3.2
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
```

```
##     cov, smooth, var
```

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.3.2
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(glue)
```

Importing Data

```
wine_train_raw=read.csv(url("https://raw.githubusercontent.com/sleepysloth12/data621_hw5/main/wine-trai

wine_test_raw=read.csv(url("https://raw.githubusercontent.com/sleepysloth12/data621_hw5/main/wine-evalua
```

Part I

The wine training dataset contains 12,795 observations and 16 elements. The variable types are numeric values represented as integers. There is missing data in the following columns: ResidualSugar, Chlorides, FreeSulfurDioxide, TotalSulfurDioxide, pH, Sulphates, Alcohol, STARS.

```
dim(wine_train_raw)
```

```
## [1] 12795    16
```

```
colSums(is.na(wine_train_raw))
```

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

```
summary_stats <- wine_train_raw %>%
  summarise(
    Mean_Target = mean(TARGET, na.rm = TRUE),
    SD_Target = sd(TARGET, na.rm = TRUE),
    Median_Target = median(TARGET, na.rm = TRUE),
    Mean_AcidIndex = mean(AcidIndex, na.rm = TRUE),
    SD_AcidIndex = sd(AcidIndex, na.rm = TRUE),
    Median_AcidIndex = median(AcidIndex, na.rm = TRUE),
    Mean_Alcohol = mean(Alcohol, na.rm = TRUE),
    SD_Alcohol = sd(Alcohol, na.rm = TRUE),
    Median_Alcohol = median(Alcohol, na.rm = TRUE),
    Mean_Chlorides = mean(Chlorides, na.rm = TRUE),
    SD_Chlorides = sd(Chlorides, na.rm = TRUE),
    Median_Chlorides = median(Chlorides, na.rm = TRUE),
    Mean_CitricAcid = mean(CitricAcid, na.rm = TRUE),
    SD_CitricAcid = sd(CitricAcid, na.rm = TRUE),
```

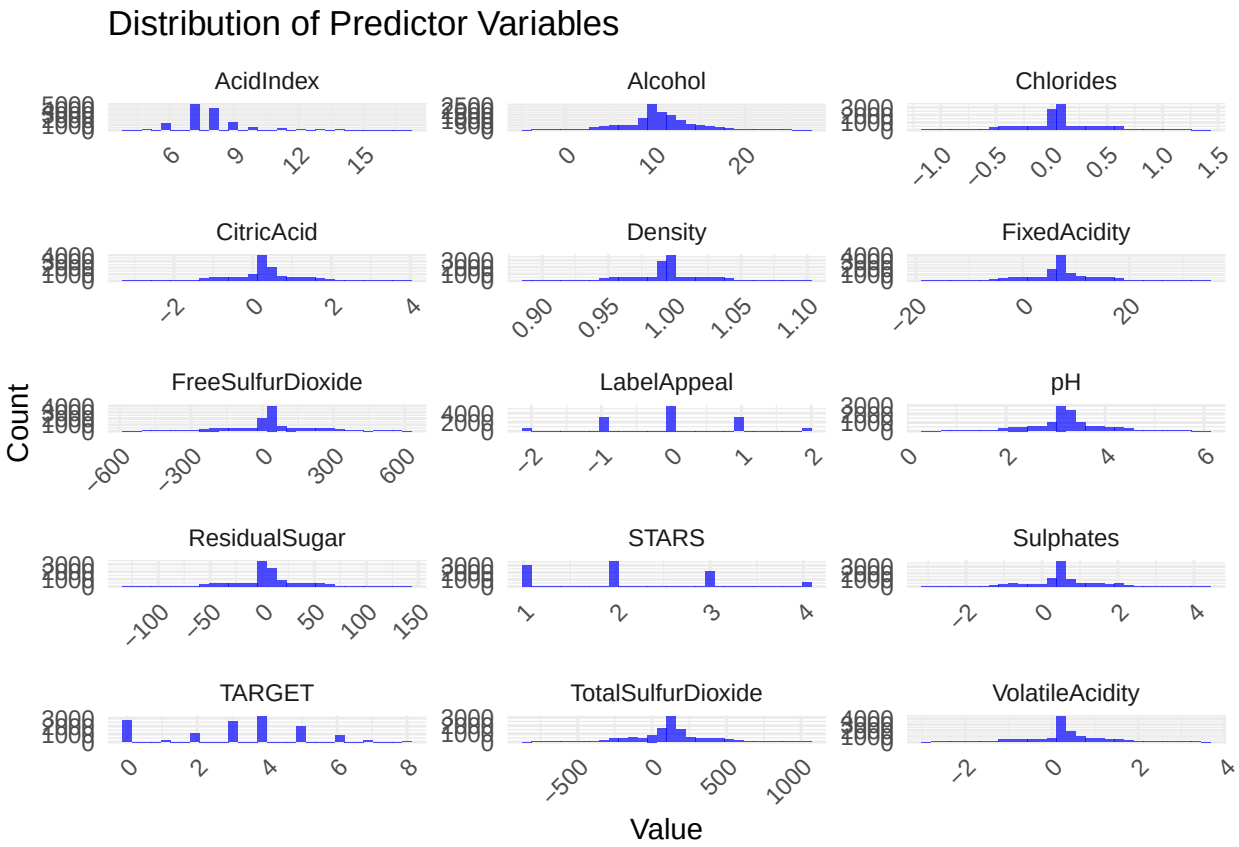
```
  Median_CitricAcid = median(CitricAcid, na.rm = TRUE),
  Mean_Density = mean(Density, na.rm = TRUE),
  SD_Density = sd(Density, na.rm = TRUE),
  Median_Density = median(Density, na.rm = TRUE),
  Mean_FixedAcidity = mean(FixedAcidity, na.rm = TRUE),
  SD_FixedAcidity = sd(FixedAcidity, na.rm = TRUE),
  Median_FixedAcidity = median(FixedAcidity, na.rm = TRUE),
  Mean_FreeSulfurDioxide = mean(FreeSulfurDioxide, na.rm = TRUE),
  SD_FreeSulfurDioxide = sd(FreeSulfurDioxide, na.rm = TRUE),
  Median_FreeSulfurDioxide = median(FreeSulfurDioxide, na.rm = TRUE),
  Mean_LabelAppeal = mean(LabelAppeal, na.rm = TRUE),
  SD_LabelAppeal = sd(LabelAppeal, na.rm = TRUE),
  Median_LabelAppeal = median(LabelAppeal, na.rm = TRUE),
  Mean_ResidualSugar = mean(ResidualSugar, na.rm = TRUE),
  SD_ResidualSugar = sd(ResidualSugar, na.rm = TRUE),
  Median_ResidualSugar = median(ResidualSugar, na.rm = TRUE),
  Mean_STARS = mean(STARS, na.rm = TRUE),
  SD_STARS = sd(STARS, na.rm = TRUE),
  Median_STARS = median(STARS, na.rm = TRUE),
  Mean_Sulphates = mean(Sulphates, na.rm = TRUE),
  SD_Sulphates = sd(Sulphates, na.rm = TRUE),
  Median_Sulphates = median(Sulphates, na.rm = TRUE),
  Mean_TotalSulfurDioxide = mean(TotalSulfurDioxide, na.rm = TRUE),
  SD_TotalSulfurDioxide = sd(TotalSulfurDioxide, na.rm = TRUE),
  Median_TotalSulfurDioxide = median(TotalSulfurDioxide, na.rm = TRUE),
  Mean_VolatileAcidity = mean(VolatileAcidity, na.rm = TRUE),
  SD_VolatileAcidity = sd(VolatileAcidity, na.rm = TRUE),
  Median_VolatileAcidity = median(VolatileAcidity, na.rm = TRUE),
  Mean_pH = mean(pH, na.rm = TRUE),
  SD_pH = sd(pH, na.rm = TRUE),
  Median_pH = median(pH, na.rm = TRUE)
) %>%
pivot_longer(everything(), names_to = "Statistic", values_to = "Value") %>%
separate(Statistic, into = c("Measure", "Variable"), sep = "_") %>%
pivot_wider(names_from = Measure, values_from = Value) %>%
dplyr::select(Variable, Mean, SD, Median) %>%
mutate(Variable = case_when(
  Variable == "TARGET" ~ "Target",
  Variable == "AcidIndex" ~ "Acid Index",
  Variable == "Alcohol" ~ "Alcohol Content",
  Variable == "Chloride" ~ "Chloride Concentration",
  Variable == "CitricAcid" ~ "Citric Acid Content",
  Variable == "Density" ~ "Density",
  Variable == "FixedAcidity" ~ "Fixed Acidity",
  Variable == "FreeSulfurDioxide" ~ "Free Sulfur Dioxide",
  Variable == "LabelAppeal" ~ "Label Appeal",
  Variable == "ResidualSugar" ~ "Residual Sugar",
  Variable == "STARS" ~ "STARS",
  Variable == "Sulphates" ~ "Sulphates Content",
  Variable == "TotalSulfurDioxide" ~ "Total Sulfur Dioxide",
  Variable == "VolatileAcidity" ~ "Volatile Acidity",
  Variable == "pH" ~ "pH",
  TRUE ~ Variable
```

```
  ))

summary_stats %>%
  gt() %>%
  tab_header(
    title = "Summary Statistics of Predictor Variables"
  ) %>%
  cols_label(
    Variable = "Variable",
    Mean = "Mean",
    SD = "Standard Deviation",
    Median = "Median"
  )
```

Summary Statistics of Predictor Variables

Variable	Mean	Standard Deviation	Median
Target	3.029073857	1.92636822	3.00000
Acid Index	7.772723720	1.32392637	8.00000
Alcohol Content	10.489236260	3.72781904	10.40000
Chlorides	0.054822489	0.31846729	0.04600
Citric Acid Content	0.308412661	0.86207979	0.31000
Density	0.994202718	0.02653765	0.99449
Fixed Acidity	7.075717077	6.31764346	6.90000
Free Sulfur Dioxide	30.845571287	148.71455765	30.00000
Label Appeal	-0.009066041	0.89108925	0.00000
Residual Sugar	5.418733065	33.74937899	3.90000
STARS	2.041754981	0.90254005	2.00000
Sulphates Content	0.527111782	0.93212926	0.50000
Total Sulfur Dioxide	120.714232643	231.91321051	123.00000
Volatile Acidity	0.324103947	0.78401424	0.28000
pH	3.207628226	0.67968708	3.20000

```
train_dat_long = wine_train_raw %>%
  dplyr::select(-INDEX) %>%
  pivot_longer(cols = everything(), names_to = "Variable", values_to = "Value")

train_dat_long <- train_dat_long[complete.cases(train_dat_long$Value), ]

options(repr.plot.width=20, repr.plot.height=20)

ggplot(train_dat_long, aes(x = Value)) +
  geom_histogram(bins = 30, fill = "blue", alpha = 0.7) +
  facet_wrap(~ Variable, scales = "free", ncol = 3) +
  theme_minimal() +
  labs(title = "Distribution of Predictor Variables", x = "Value", y = "Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



- The mean and median are close in value for all variables above suggesting the data is likely symmetrically distributed. Additionally, most variables have low variability in standard deviation.
- The following variables have higher variability in Standard Deviation: 'Free Sulfur Dioxide', 'Total Sulfur Dioxide', 'Residual Sugar'
- As seen in the histogram above the the distribution for 'TARGET' is slightly right skewed likely due to the amount of 0 case purchased. This pattern of slight right skew is also present in the variable 'STARS' and 'AcidIndex'.
- The histograms also reflect that all other variables are unimodal and seem to have a normal distribution.
- 'LabelAppeal', Marketing Score indicating the appeal of label design for consumers, 'STARS', wine rating 1-4, and 'AcidIndex', Proprietary method of testing total acidity of wine by using a weighted average, are actually categorical variables.
- Some variables have negative values which does not make sense in context such as alcohol content.

Summary Statistics and Distribution

```
summary1 = dfSummary(wine_train_raw[, -which(names(wine_train_raw) == "INDEX")])


print(summary1, method = "render")
```

```
view(summary1)
```

```
## Switching method to 'browser'
```

```
## Output file written: C:\Users\bleac\AppData\Local\Temp\Rtmp4iWVfO\file44403d1276f9.html
```

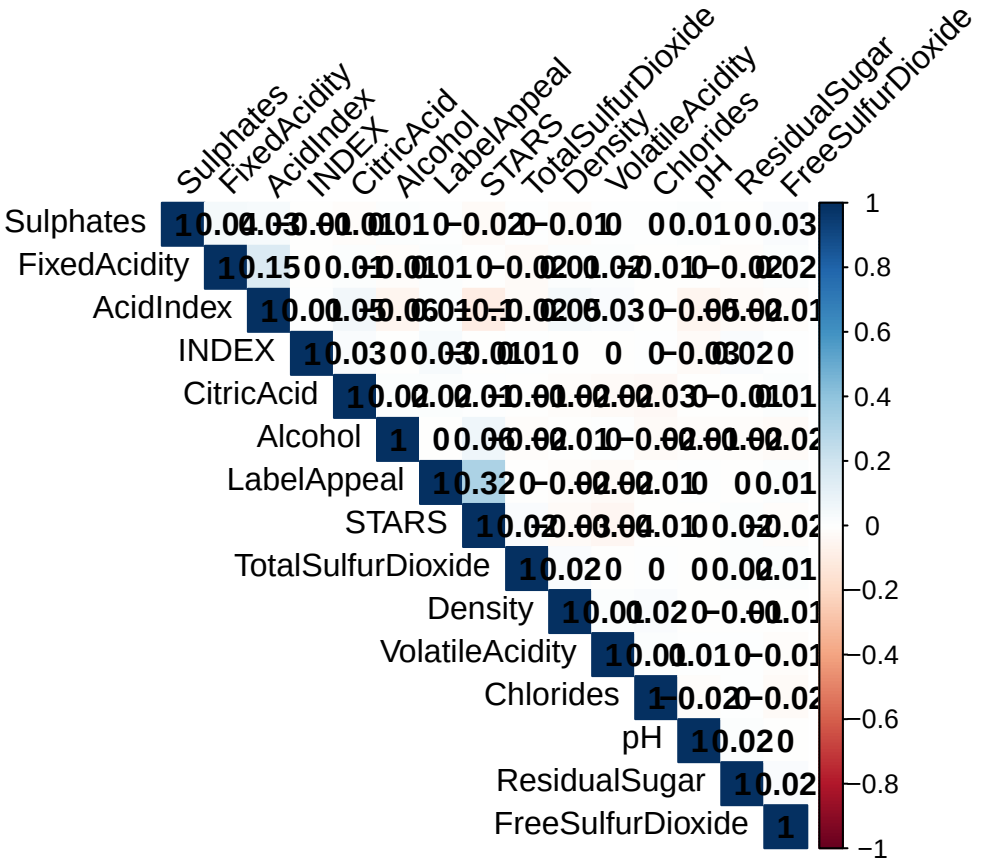
Correlation Matrix

```
numeric_data = wine_train_raw %>%
  dplyr::select(-TARGET) %>%
  select_if(is.numeric)

correlation_matrix = cor(numeric_data, use = "complete.obs")

print(correlation_matrix)
```

```
##                            INDEX FixedAcidity VolatileAcidity    CitricAcid
## INDEX                1.0000000000 -0.002831415   -0.0008743296  0.0278869710
## FixedAcidity        -0.0028314152  1.000000000    0.0190109733  0.0140003760
## VolatileAcidity     -0.0008743296  0.019010973    1.0000000000 -0.0234315631
## CitricAcid           0.0278869710  0.014000376   -0.0234315631  1.0000000000
## ResidualSugar        0.0208952098 -0.015429391    0.0015279517 -0.0098431456
## Chlorides            0.0026827829 -0.006104447    0.0148489225 -0.0335608661
## FreeSulfurDioxide    0.0046416504  0.015438463   -0.0114408079  0.0121132485
## TotalSulfurDioxide   0.0064949038 -0.023323485   -0.0007434083 -0.0099174506
## Density             -0.0034840089  0.011574241    0.0130977690 -0.0169919691
## pH                  -0.0274556333 -0.004553886    0.0072030364 -0.0007581304
## Sulphates           -0.0053946247  0.042229181    0.0015161001 -0.0144237270
## Alcohol             -0.0024453460 -0.013085026    0.0002603082  0.0169864284
## LabelAppeal          0.0314911460  0.011375965   -0.0202419713  0.0153315666
## AcidIndex            0.0055244862  0.154167846    0.0250529742  0.0545838104
## STARS               -0.0057807296 -0.004937345   -0.0402432388  0.0071401699
##                     ResidualSugar     Chlorides FreeSulfurDioxide
## INDEX                 0.020895210  0.0026827829       0.004641650
## FixedAcidity         -0.015429391 -0.0061044471       0.015438463
## VolatileAcidity       0.001527952  0.0148489225      -0.011440808
## CitricAcid           -0.009843146 -0.0335608661       0.012113248
## ResidualSugar         1.000000000  0.0041215692       0.021959113
## Chlorides             0.004121569  1.0000000000      -0.020492488
## FreeSulfurDioxide     0.021959113 -0.0204924876       1.000000000
## TotalSulfurDioxide    0.017030939  0.0004188605       0.013461673
## Density              -0.007120841  0.0206724860      -0.008663509
## pH                    0.017563769 -0.0179702278      -0.002008516
## Sulphates            -0.002705775  0.0026187777       0.026829029
## Alcohol              -0.018943324 -0.0228849573      -0.023867458
## LabelAppeal          -0.004579308 -0.0063870237       0.014960087
## AcidIndex            -0.020301890 -0.0017134096      -0.014733717
## STARS                 0.019665541 -0.0063242568      -0.015390398
##                     TotalSulfurDioxide      Density            pH    Sulphates
## INDEX                       0.0064949038 -0.003484009 -0.0274556333 -0.005394625
## FixedAcidity               -0.0233234848  0.011574241 -0.0045538857  0.042229181
## VolatileAcidity            -0.0007434083  0.013097769  0.0072030364  0.001516100
## CitricAcid                 -0.0099174506 -0.016991969 -0.0007581304 -0.014423727
## ResidualSugar               0.0170309394 -0.007120841  0.0175637691 -0.002705775
## Chlorides                   0.0004188605  0.020672486 -0.0179702278  0.002618778
## FreeSulfurDioxide           0.0134616726 -0.008663509 -0.0020085157  0.026829029
## TotalSulfurDioxide          1.0000000000  0.023167955 -0.0034227601  0.002504051
## Density                     0.0231679548  1.000000000 -0.0020192285 -0.010609294
```

```
## pH                    -0.0034227601 -0.002019229  1.0000000000  0.010449255
## Sulphates              0.0025040509 -0.010609294  0.0104492547  1.000000000
## Alcohol               -0.0168515467 -0.006128355 -0.0122034469  0.010844330
## LabelAppeal           -0.0027237419 -0.018094403  0.0002181758  0.003768700
## AcidIndex             -0.0221292631  0.047778830 -0.0537128921  0.031071782
## STARS                  0.0220949002 -0.028492455 -0.0044002985 -0.023135130
##                         Alcohol   LabelAppeal    AcidIndex        STARS
## INDEX              -0.0024453460  0.0314911460  0.005524486 -0.005780730
## FixedAcidity       -0.0130850260  0.0113759650  0.154167846 -0.004937345
## VolatileAcidity     0.0002603082 -0.0202419713  0.025052974 -0.040243239
## CitricAcid          0.0169864284  0.0153315666  0.054583810  0.007140170
## ResidualSugar      -0.0189433242 -0.0045793083 -0.020301890  0.019665541
## Chlorides          -0.0228849573 -0.0063870237 -0.001713410 -0.006324257
## FreeSulfurDioxide  -0.0238674577  0.0149600871 -0.014733717 -0.015390398
## TotalSulfurDioxide -0.0168515467 -0.0027237419 -0.022129263  0.022094900
## Density            -0.0061283546 -0.0180944026  0.047778830 -0.028492455
## pH                 -0.0122034469  0.0002181758 -0.053712892 -0.004400299
## Sulphates           0.0108443299  0.0037686996  0.031071782 -0.023135130
## Alcohol             1.0000000000 -0.0006449123 -0.055891906  0.064854486
## LabelAppeal        -0.0006449123  1.0000000000  0.010300984  0.318897022
## AcidIndex          -0.0558919056  0.0103009840  1.000000000 -0.095482582
## STARS               0.0648544864  0.3188970216 -0.095482582  1.000000000
```

```
corrplot(correlation_matrix, method = "color", type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45, addCoef.col = "black")
```



Weak Correlations Mostly:

Most variables exhibit very weak correlations with each other, which are close to 0. This suggests no strong linear relationships between most pairs of features.

LabelAppeal and STARS: The strongest correlation observed is between LabelAppeal and STARS (0.3189). This suggests a moderate positive relationship, where wines with more appealing labels tend to be rated higher by experts.

FixedAcidity and AcidIndex: There's a noticeable positive correlation between FixedAcidity and AcidIndex (0.1542). Although not very strong, it indicates that as fixed acidity increases, the overall acidity index tends to increase as well.

Alcohol and STARS: Alcohol content shows a modest correlation with STARS (0.0649), suggesting that higher alcohol content could be slightly associated with higher ratings by experts.

Negative Correlations: There are several mild negative correlations, such as AcidIndex and STARS (-0.0955), indicating that higher acid index values may slightly correlate with lower expert ratings.

Part II

Looking For Missing Values and Fixing it

The following are missing:

616 Records of `ResidualSugar` (4.8%) 638 Records of `Chlorides` (5%) 647 Records of `FreeSulfurDioxide` (5.1%) 682 Records of `TotalSulfurDioxide` (5.3%) 395 Records of `pH` (3.1%) 1210 Records of `Sulphates` (9.5%) 653 Records of `Alcohol` (5.1%) 3359 Records of `STARS` (26.3%)

All of these columns (with the exception of `STARS`) are normally distributed.

To fill in the missing values, I will use mean and standard deviation by using the `rnorm()` function and creating a function that will process these columns to fill in the missing data.

```
fill_missing_with_normal = function(data, column_name) {

  column_data = data[[column_name]]

  mean_val = mean(column_data, na.rm = TRUE)
  sd_val = sd(column_data, na.rm = TRUE)

  num_missing = sum(is.na(column_data))

  new_values = rnorm(num_missing, mean = mean_val, sd = sd_val)

  data[[column_name]][is.na(column_data)] = new_values

  return(data)
}

columns_with_missing = c("ResidualSugar", "Chlorides", "FreeSulfurDioxide",
                         "TotalSulfurDioxide", "pH", "Sulphates", "Alcohol")

for (column in columns_with_missing) {
  wine_train_raw = fill_missing_with_normal(wine_train_raw, column)
}

for (column in columns_with_missing) {
  wine_test_raw = fill_missing_with_normal(wine_test_raw, column)
}
```

```
summary2 = dfSummary(wine_train_raw[, -which(names(wine_train_raw) == "INDEX")])


print(summary2, method = "render")
```

```
view(summary2)
```

```
## Switching method to 'browser'
```

```
## Output file written: C:\Users\bleac\AppData\Local\Temp\Rtmp4iWVfO\file44404605af1.html
```

Now, there are no more missing values for the normally distributed columns.

Only 3359 (26.3%) of records now are missing `STARS` label.

I will use the existing probabilities of the distribution of `STARS` data to do bootstraping to generate the missing values.

```
fill_missing_categorical = function(data, column_name, levels, probabilities) {

  column_data = data[[column_name]]


  num_missing = sum(is.na(column_data))


  new_values = sample(levels, num_missing, replace = TRUE, prob = probabilities)


  data[[column_name]][is.na(column_data)] <- new_values

  return(data)
}


stars_levels = c(1, 2, 3, 4)


stars_probabilities = c(32.2, 37.8, 23.4, 6.5) / 100


wine_train_raw = fill_missing_categorical(wine_train_raw, "STARS", stars_levels, stars_probabilities)
wine_test_raw = fill_missing_categorical(wine_test_raw, "STARS", stars_levels, stars_probabilities)


summary3 = dfSummary(wine_train_raw[, -which(names(wine_train_raw) == "INDEX")])


print(summary3, method = "render")
```

```
view(summary3)
```

```
## Switching method to 'browser'
```

```
## Output file written: C:\Users\bleac\AppData\Local\Temp\Rtmp4iWVfO\file4440ec21bf.html
```

No missing values, and we keep the distribution patterns that originally existed. Nice!

Transforming the Data

Adding interaction terms can help to capture the combined effects of variables.

Acidity Interactions: We will explore interactions between `FixedAcidity`, `VolatileAcidity`, and `CitricAcid`.

`Sulfur Dioxide` and `Free Sulfur Dioxide`: These could interact with other chemical properties like `pH` and `Alcohol` content, affecting wine stability and taste.

Sulfur Ratio: The ratio of `FreeSulfurDioxide` to `TotalSulfurDioxide` might give insights into how bound versus free sulfur dioxide impacts wine quality.

Sugar to Acid Ratio: We will use `ResidualSugar` to `TotalAcidity` (`TotalAcidity` as a sum of `FixedAcidity` and other acid measures).

```
wine_train_clean = wine_train_raw %>%
  mutate(

    FixedAcid_VolatileAcid = FixedAcidity * VolatileAcidity,
    Alcohol_CitricAcid = Alcohol * CitricAcid,
    Alcohol_pH = Alcohol * pH,


    SugarToAcidRatio = ResidualSugar / (FixedAcidity + CitricAcid),
    FreeToTotalSulfur = FreeSulfurDioxide / TotalSulfurDioxide,


    Acid_Alcohol_Index = (FixedAcidity * VolatileAcidity) / Alcohol
  )


wine_train_clean = wine_train_clean %>%
  dplyr::select(TARGET,
         FixedAcid_VolatileAcid,
         Alcohol_CitricAcid,
         Alcohol_pH,
         SugarToAcidRatio,
         FreeToTotalSulfur,
         Acid_Alcohol_Index,
         AcidIndex, Alcohol, Chlorides, CitricAcid, Density,
         FixedAcidity, FreeSulfurDioxide, LabelAppeal, pH,
         ResidualSugar, STARS, Sulphates, TotalSulfurDioxide, VolatileAcidity
        )

wine_test_clean = wine_test_raw %>%
  mutate(

    FixedAcid_VolatileAcid = FixedAcidity * VolatileAcidity,
    Alcohol_CitricAcid = Alcohol * CitricAcid,
    Alcohol_pH = Alcohol * pH,


    SugarToAcidRatio = ResidualSugar / (FixedAcidity + CitricAcid),
    FreeToTotalSulfur = FreeSulfurDioxide / TotalSulfurDioxide,
```

```
    Acid_Alcohol_Index = (FixedAcidity * VolatileAcidity) / Alcohol
  )


wine_test_clean = wine_test_clean %>%
  dplyr::select(TARGET,
         FixedAcid_VolatileAcid,
         Alcohol_CitricAcid,
         Alcohol_pH,
         SugarToAcidRatio,
         FreeToTotalSulfur,
         Acid_Alcohol_Index,
         AcidIndex, Alcohol, Chlorides, CitricAcid, Density,
         FixedAcidity, FreeSulfurDioxide, LabelAppeal, pH,
         ResidualSugar, STARS, Sulphates, TotalSulfurDioxide, VolatileAcidity
        )
```

New Correlation Matrix

```
cor_matrix2 = cor(wine_train_clean %>% dplyr::select(-TARGET), use = "complete.obs")
```

Given the high correlations between the interaction terms and their original variables, I will remove original variables.

```
wine_train_clean = wine_train_clean %>%
  dplyr::select(TARGET,
         FixedAcid_VolatileAcid,
         Alcohol_CitricAcid,
         Alcohol_pH,
         SugarToAcidRatio,
         FreeToTotalSulfur,
         Acid_Alcohol_Index,
         AcidIndex,  Chlorides, Density,
          FreeSulfurDioxide, LabelAppeal,
         ResidualSugar, STARS
        )
```

Part III

Poisson Regression Model One

```
poisson_1 = glm(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + Alcohol_pH + AcidIndex + Chloride

summary(poisson_1)
```

```
##
## Call:
## glm(formula = TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid +
##     Alcohol_pH + AcidIndex + Chlorides + Density + FreeSulfurDioxide +
##     LabelAppeal + ResidualSugar + STARS, family = poisson, data = wine_train_clean)
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)             2.147e+00  1.933e-01  11.108  < 2e-16 ***
## FixedAcid_VolatileAcid -4.195e-03  6.840e-04  -6.133 8.63e-10 ***
## Alcohol_CitricAcid      1.236e-03  5.243e-04   2.358 0.018388 *
## Alcohol_pH              6.487e-04  3.637e-04   1.783 0.074506 .
## AcidIndex              -1.278e-01  4.395e-03 -29.073  < 2e-16 ***
## Chlorides              -5.657e-02  1.602e-02  -3.531 0.000414 ***
## Density                -4.467e-01  1.918e-01  -2.329 0.019864 *
## FreeSulfurDioxide       1.202e-04  3.417e-05   3.516 0.000438 ***
## LabelAppeal             2.154e-01  5.910e-03  36.440  < 2e-16 ***
## ResidualSugar           2.045e-04  1.503e-04   1.361 0.173614
## STARS                   1.617e-01  5.654e-03  28.593  < 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: 18863  on 12784  degrees of freedom
## AIC: 50827
##
## Number of Fisher Scoring iterations: 5
```

For the first Poisson Regression Model, I selected all variables that would work in a lm() function. The reason why I omitted certain variables was that the lm() would not work with "Inf" values present in the created ratios and indexes.

Poisson Regression Model Two

```
poisson_2 = glm(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + AcidIndex + Chlorides + Density +

summary(poisson_2)
```

```
##
## Call:
## glm(formula = TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid +
##     AcidIndex + Chlorides + Density + FreeSulfurDioxide + LabelAppeal +
##     STARS, family = poisson, data = wine_train_clean)
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             2.175e+00  1.928e-01  11.280  < 2e-16 ***
## FixedAcid_VolatileAcid -4.211e-03  6.840e-04  -6.157 7.42e-10 ***
## Alcohol_CitricAcid      1.349e-03  5.212e-04   2.588 0.009642 **
## AcidIndex              -1.284e-01  4.381e-03 -29.313  < 2e-16 ***
## Chlorides              -5.703e-02  1.602e-02  -3.561 0.000370 ***
## Density                -4.468e-01  1.918e-01  -2.330 0.019823 *
## FreeSulfurDioxide       1.198e-04  3.417e-05   3.507 0.000454 ***
## LabelAppeal             2.155e-01  5.910e-03  36.457  < 2e-16 ***
## STARS                   1.621e-01  5.651e-03  28.685  < 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: 18868  on 12786  degrees of freedom
## AIC: 50828
##
## Number of Fisher Scoring iterations: 5
```

For the second poisson regression model, I selected all the predictors from the first model which were labeled as significant predictors.

Negative Binomial Regression Model One

```
nb_1 = MASS::glm.nb(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + Alcohol_pH + AcidIndex + Chl
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached

## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
summary(nb_1)
```

```
##
## Call:
## MASS::glm.nb(formula = TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid +
##     Alcohol_pH + AcidIndex + Chlorides + Density + FreeSulfurDioxide +
##     LabelAppeal + ResidualSugar + STARS, data = wine_train_clean,
##     init.theta = 34248.82388, link = log)
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             2.147e+00  1.933e-01  11.108  < 2e-16 ***
## FixedAcid_VolatileAcid -4.195e-03  6.840e-04  -6.133 8.64e-10 ***
## Alcohol_CitricAcid      1.236e-03  5.243e-04   2.358 0.018395 *
## Alcohol_pH              6.487e-04  3.637e-04   1.783 0.074532 .
## AcidIndex              -1.278e-01  4.395e-03 -29.072  < 2e-16 ***
## Chlorides              -5.658e-02  1.602e-02  -3.531 0.000414 ***
## Density                -4.467e-01  1.918e-01  -2.329 0.019869 *
## FreeSulfurDioxide       1.202e-04  3.417e-05   3.516 0.000438 ***
## LabelAppeal             2.154e-01  5.910e-03  36.439  < 2e-16 ***
## ResidualSugar           2.045e-04  1.503e-04   1.361 0.173616
## STARS                   1.617e-01  5.655e-03  28.592  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(34248.82) family taken to be 1)
##
##     Null deviance: 22860  on 12794  degrees of freedom
## Residual deviance: 18862  on 12784  degrees of freedom
## AIC: 50829
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  34249
##           Std. Err.:  59615
```

```
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -50805.07
```

For the first negative binomial regression

Negative Binomial Regression Model Two

```
nb_2 = MASS::glm.nb(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + AcidIndex + Chlorides + Dens
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached

## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
summary(nb_2)
```

```
##
## Call:
## MASS::glm.nb(formula = TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid +
##     AcidIndex + Chlorides + Density + FreeSulfurDioxide + LabelAppeal +
##     STARS, data = wine_train_clean, init.theta = 34136.02722,
##     link = log)
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)              2.175e+00  1.928e-01  11.280  < 2e-16 ***
## FixedAcid_VolatileAcid -4.211e-03  6.841e-04  -6.157 7.44e-10 ***
## Alcohol_CitricAcid      1.349e-03  5.212e-04   2.588 0.009646 **
## AcidIndex              -1.284e-01  4.382e-03 -29.312  < 2e-16 ***
## Chlorides              -5.703e-02  1.602e-02  -3.560 0.000370 ***
## Density                -4.468e-01  1.918e-01  -2.330 0.019827 *
## FreeSulfurDioxide       1.198e-04  3.417e-05   3.507 0.000454 ***
## LabelAppeal             2.155e-01  5.910e-03  36.455  < 2e-16 ***
## STARS                   1.621e-01  5.651e-03  28.683  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(34136.03) family taken to be 1)
##
##     Null deviance: 22860  on 12794  degrees of freedom
## Residual deviance: 18867  on 12786  degrees of freedom
## AIC: 50830
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  34136
##           Std. Err.:  59500
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -50810.07
```

Similarly, I selected the variables that were labeled as significant predictors.

Multiple Linear Regression Model One

```
ml_1 = lm(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + Alcohol_pH + AcidIndex + Chlorides + De

summary(ml_1)
```

```
##
## Call:
## lm(formula = TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid +
##     Alcohol_pH + AcidIndex + Chlorides + Density + FreeSulfurDioxide +
##     LabelAppeal + ResidualSugar + STARS, data = wine_train_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3309 -0.7155  0.4032  1.1286  4.3831
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              5.8776197  0.5586424  10.521  < 2e-16 ***
## FixedAcid_VolatileAcid -0.0120245  0.0019280  -6.237 4.61e-10 ***
## Alcohol_CitricAcid      0.0037334  0.0015378   2.428 0.015205 *
## Alcohol_pH              0.0028128  0.0010530   2.671 0.007567 **
## AcidIndex              -0.3413471  0.0112529 -30.334  < 2e-16 ***
## Chlorides              -0.1808617  0.0462980  -3.906 9.41e-05 ***
## Density                -1.3429652  0.5553637  -2.418 0.015613 *
## FreeSulfurDioxide       0.0003650  0.0000989   3.691 0.000224 ***
## LabelAppeal             0.6548607  0.0170357  38.441  < 2e-16 ***
## ResidualSugar           0.0006845  0.0004358   1.571 0.116303
## STARS                   0.5200576  0.0168784  30.812  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.665 on 12784 degrees of freedom
## Multiple R-squared:  0.2538, Adjusted R-squared:  0.2532
## F-statistic: 434.8 on 10 and 12784 DF,  p-value: < 2.2e-16
```

From the wine_train_clean, I selected all variables that would work in a lm() function. The reason why I omitted certain variables was that the lm() would not work with "Inf" values present in the created ratios and indexes.

Multiple Linear Regression Model Two

```
ml_2 = step(ml_1)
```

```
## Start:  AIC=13053.18
## TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + Alcohol_pH +
##     AcidIndex + Chlorides + Density + FreeSulfurDioxide + LabelAppeal +
##     ResidualSugar + STARS
##
##                          Df Sum of Sq   RSS   AIC
## <none>                                 35428 13053
## - ResidualSugar           1       6.8 35435 13054
## - Density                 1      16.2 35445 13057
## - Alcohol_CitricAcid      1      16.3 35445 13057
## - Alcohol_pH              1      19.8 35448 13058
```

```
## - FreeSulfurDioxide       1      37.8 35466 13065
## - Chlorides               1      42.3 35471 13066
## - FixedAcid_VolatileAcid  1     107.8 35536 13090
## - AcidIndex               1    2550.0 37978 13940
## - STARS                   1    2631.0 38059 13968
## - LabelAppeal             1    4095.1 39523 14451
```

```
summary(ml_2)
```

```
##
## Call:
## lm(formula = TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid +
##     Alcohol_pH + AcidIndex + Chlorides + Density + FreeSulfurDioxide +
##     LabelAppeal + ResidualSugar + STARS, data = wine_train_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3309 -0.7155  0.4032  1.1286  4.3831
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              5.8776197  0.5586424  10.521  < 2e-16 ***
## FixedAcid_VolatileAcid -0.0120245  0.0019280  -6.237 4.61e-10 ***
## Alcohol_CitricAcid       0.0037334  0.0015378   2.428 0.015205 *
## Alcohol_pH               0.0028128  0.0010530   2.671 0.007567 **
## AcidIndex               -0.3413471  0.0112529 -30.334  < 2e-16 ***
## Chlorides               -0.1808617  0.0462980  -3.906 9.41e-05 ***
## Density                 -1.3429652  0.5553637  -2.418 0.015613 *
## FreeSulfurDioxide        0.0003650  0.0000989   3.691 0.000224 ***
## LabelAppeal              0.6548607  0.0170357  38.441  < 2e-16 ***
## ResidualSugar            0.0006845  0.0004358   1.571 0.116303
## STARS                    0.5200576  0.0168784  30.812  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.665 on 12784 degrees of freedom
## Multiple R-squared:  0.2538, Adjusted R-squared:  0.2532
## F-statistic: 434.8 on 10 and 12784 DF,  p-value: < 2.2e-16
```

Using the initial multiple linear regression, I applied the step() function to create a more filtered down multiple linear regression model.

Selecting the Most Appropriate Count Regression Model

Model Validation

MSE Calculations

```
predicted_vals_p1 <- predict(poisson_1, type = 'response')
observed_vals_p1 <- wine_train_clean$TARGET
res_p1 <- observed_vals_p1 - predicted_vals_p1
mse_p1 <- mean(res_p1**2)

predicted_vals_p2 <- predict(poisson_2, type = 'response')
observed_vals_p2 <- wine_train_clean$TARGET
```

```r
res_p2 <- observed_vals_p2 - predicted_vals_p2
mse_p2 <- mean(res_p2**2)

predicted_vals_nb_1 <- predict(nb_1, type = 'response')
observed_vals_nb_1 <- wine_train_clean$TARGET
res_nb_1 <- observed_vals_nb_1 - predicted_vals_nb_1
mse_nb_1 <- mean(res_nb_1**2)

predicted_vals_nb_2 <- predict(nb_2, type = 'response')
observed_vals_nb_2 <- wine_train_clean$TARGET
res_nb_2 <- observed_vals_nb_2 - predicted_vals_nb_2
mse_nb_2 <- mean(res_nb_2**2)

predicted_vals_ml_1 <- predict(ml_1, type = 'response')
observed_vals_ml_1 <- wine_train_clean$TARGET
rse_ml_1 <- observed_vals_ml_1 - predicted_vals_ml_1
mse_ml_1 <- mean(rse_ml_1**2)

predicted_vals_ml_2 <- predict(ml_2, type = 'response')
observed_vals_ml_2 <- wine_train_clean$TARGET
rse_ml_2 <- observed_vals_ml_2 - predicted_vals_ml_2
mse_ml_2 <- mean(rse_ml_2**2)
```

```r
cat("Poisson 1 MSE:", mse_p1, "\n")
```

```
## Poisson 1 MSE: 2.767449
```

```r
cat("Poisson 2 MSE:", mse_p2, "\n")
```

```
## Poisson 2 MSE: 2.769021
```

```r
cat("Negative Binomial 1 MSE:", mse_nb_1, "\n")
```

```
## Negative Binomial 1 MSE: 2.76745
```

```r
cat("Negative Binomial 2 MSE:", mse_nb_2, "\n")
```

```
## Negative Binomial 2 MSE: 2.769021
```

```r
cat("Linear Regression 1 MSE:", mse_ml_1, "\n")
```

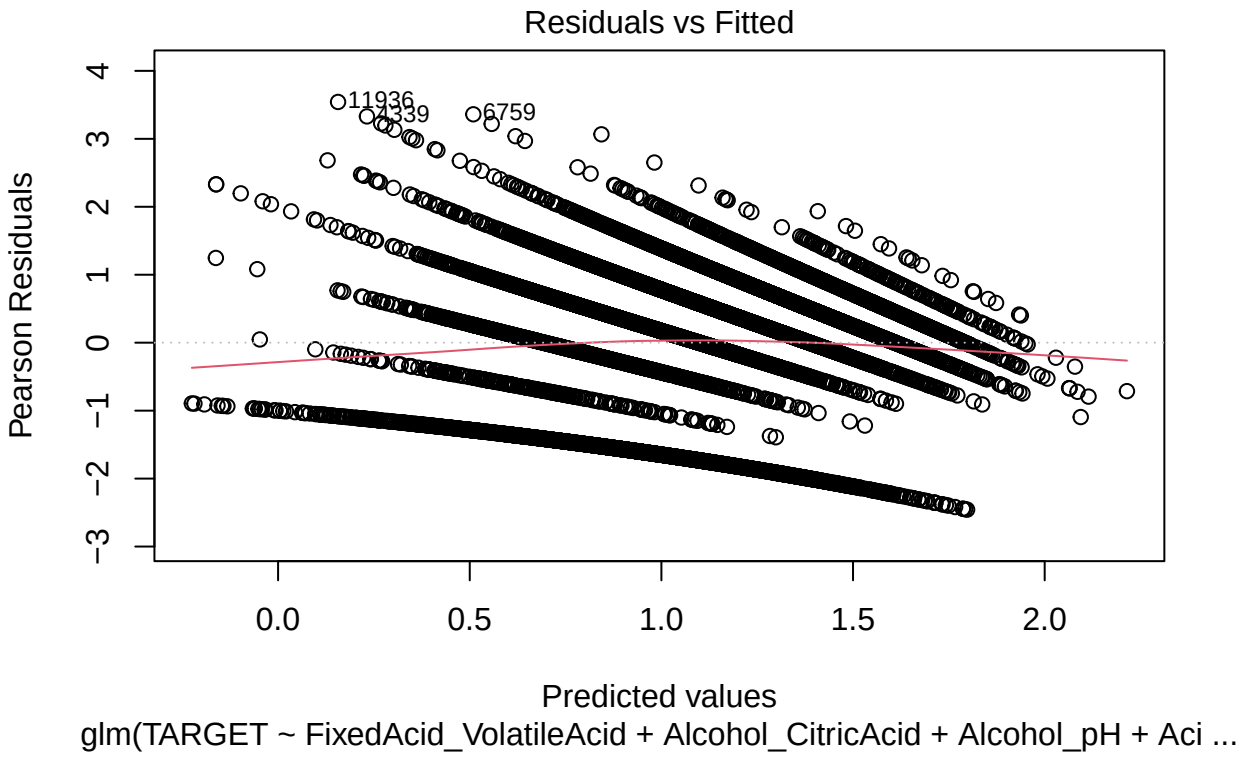
```
## Linear Regression 1 MSE: 2.768924
```

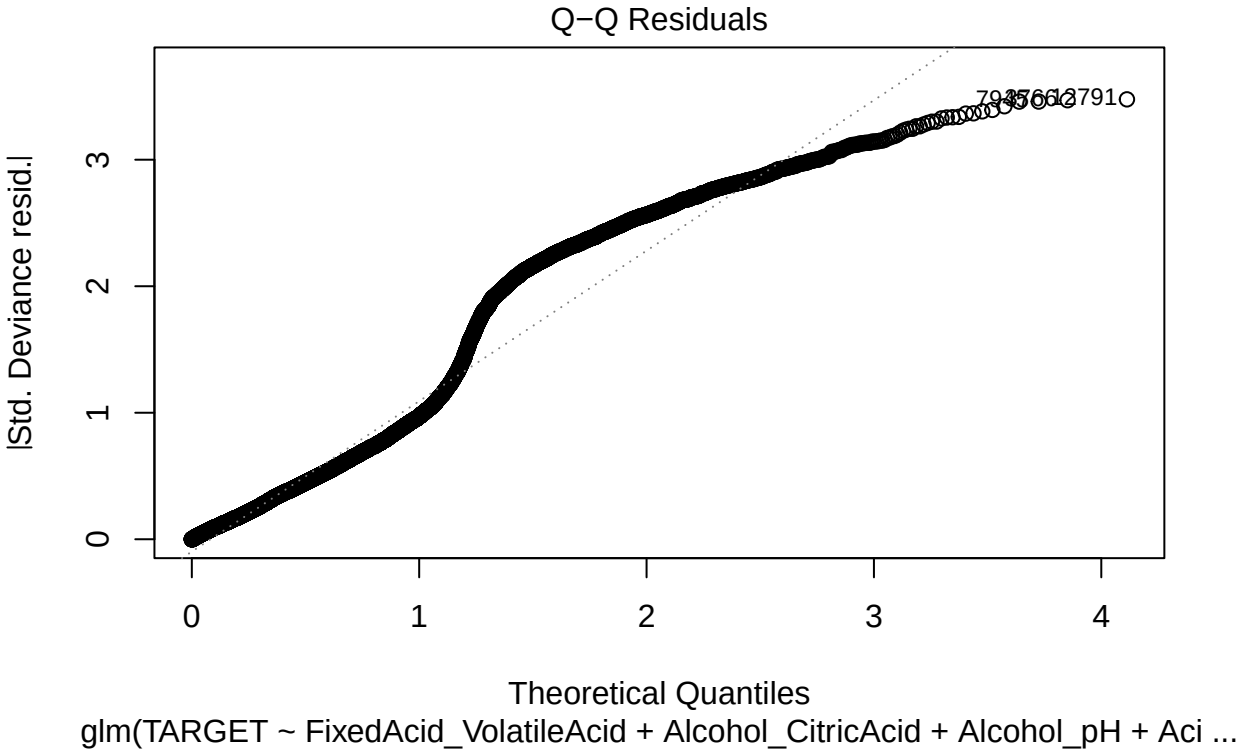
```r
cat("Linear Regression 2 MSE:", mse_ml_2, "\n")
```

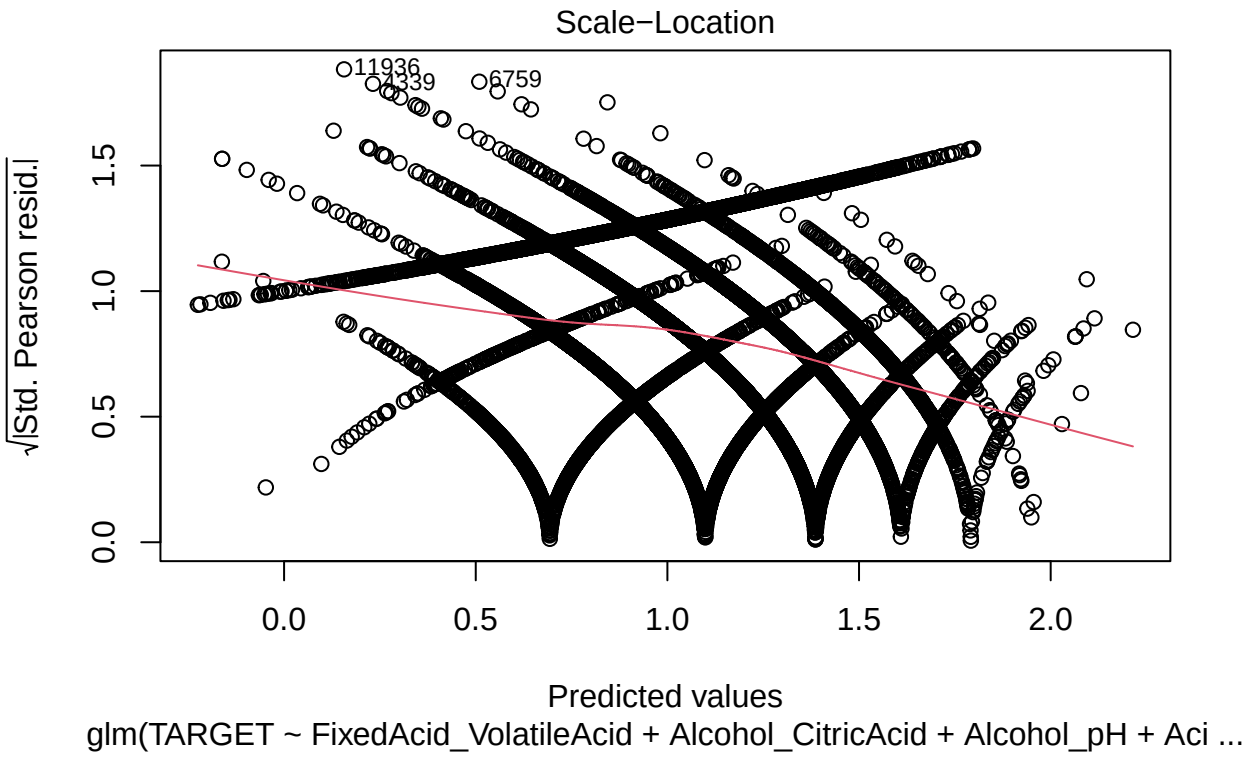
```
## Linear Regression 2 MSE: 2.768924
```

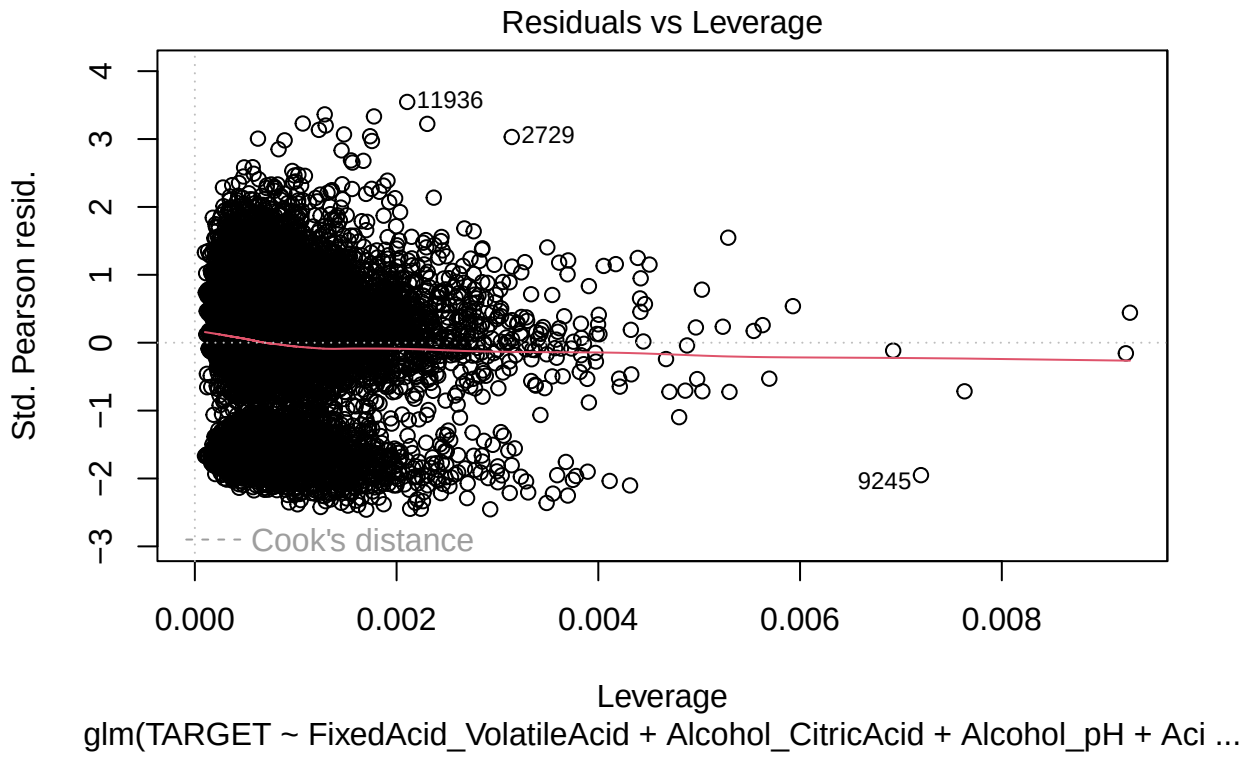
Plotted residuals

```r
plot(poisson_1)
```

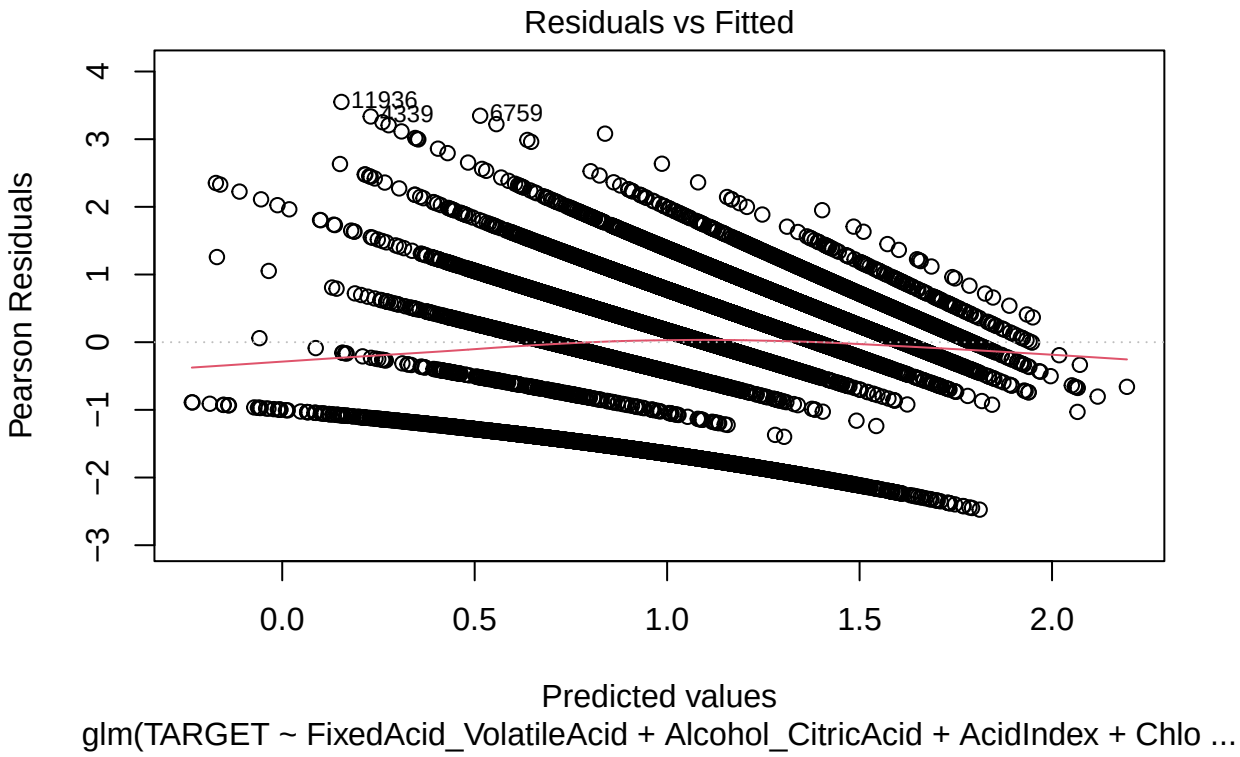

Residuals vs Fitted
Pearson Residuals
Predicted values
glm(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + Alcohol_pH + Aci ...
11936
4339
6759
4
3
2
1
0
−1
−2
−3
0.0
0.5
1.0
1.5
2.0

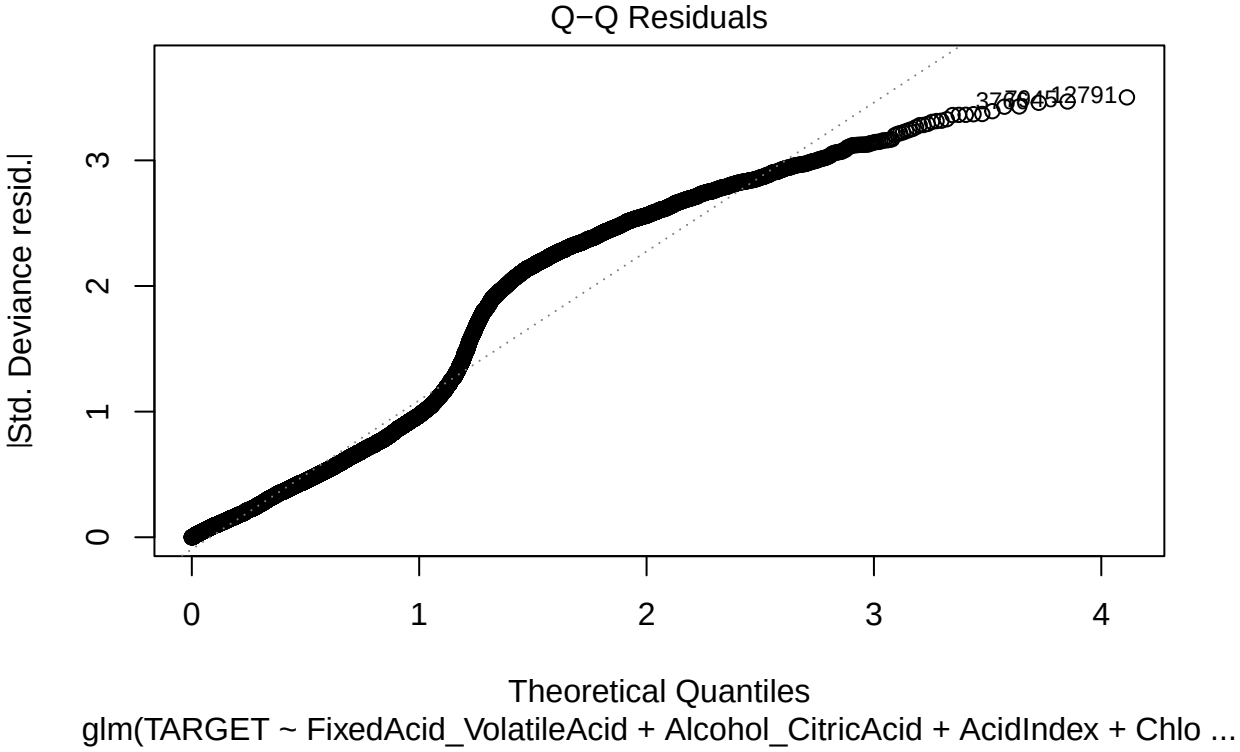

Q−Q Residuals
|Std. Deviance resid.|
0
1
2
3
0
1
2
3
4
Theoretical Quantiles
glm(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + Alcohol_pH + Aci ...

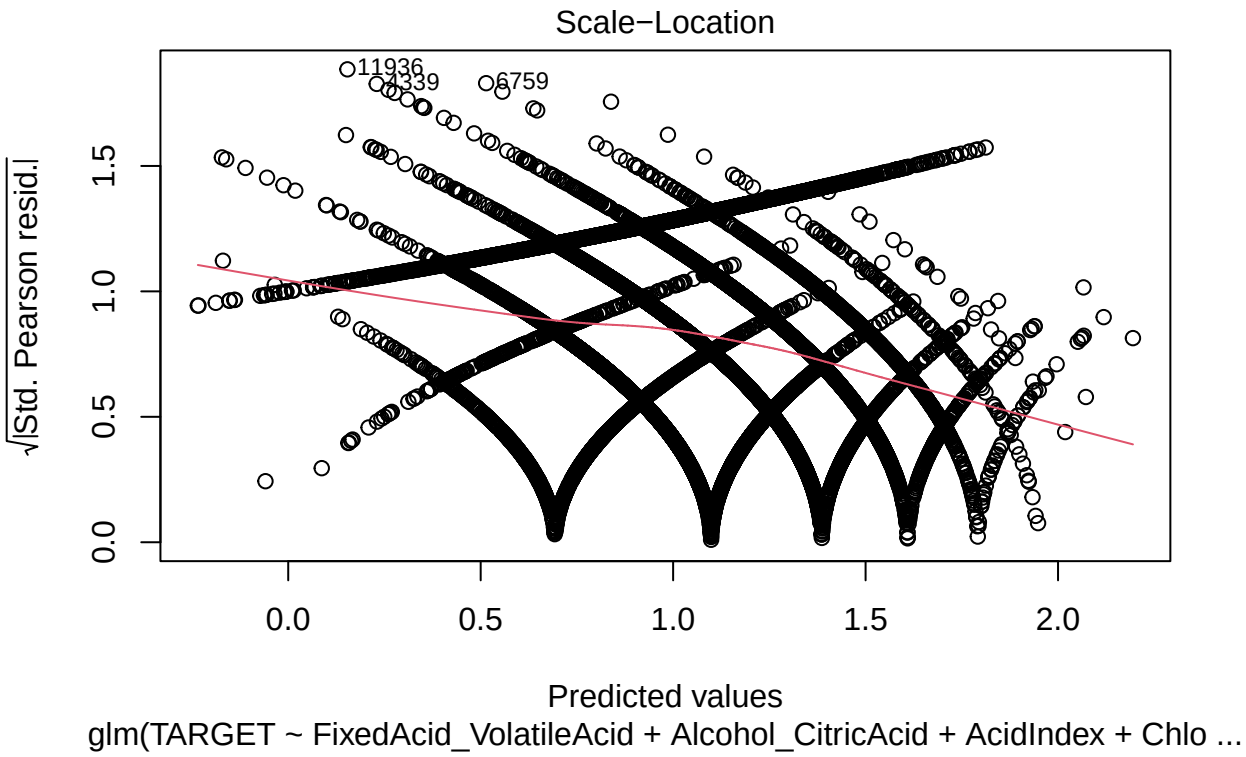

Scale−Location
11936
4339
6759
√|Std. Pearson resid.|
0.0
0.5
1.0
1.5
0.0
0.5
1.0
1.5
2.0
Predicted values
glm(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + Alcohol_pH + Aci ...

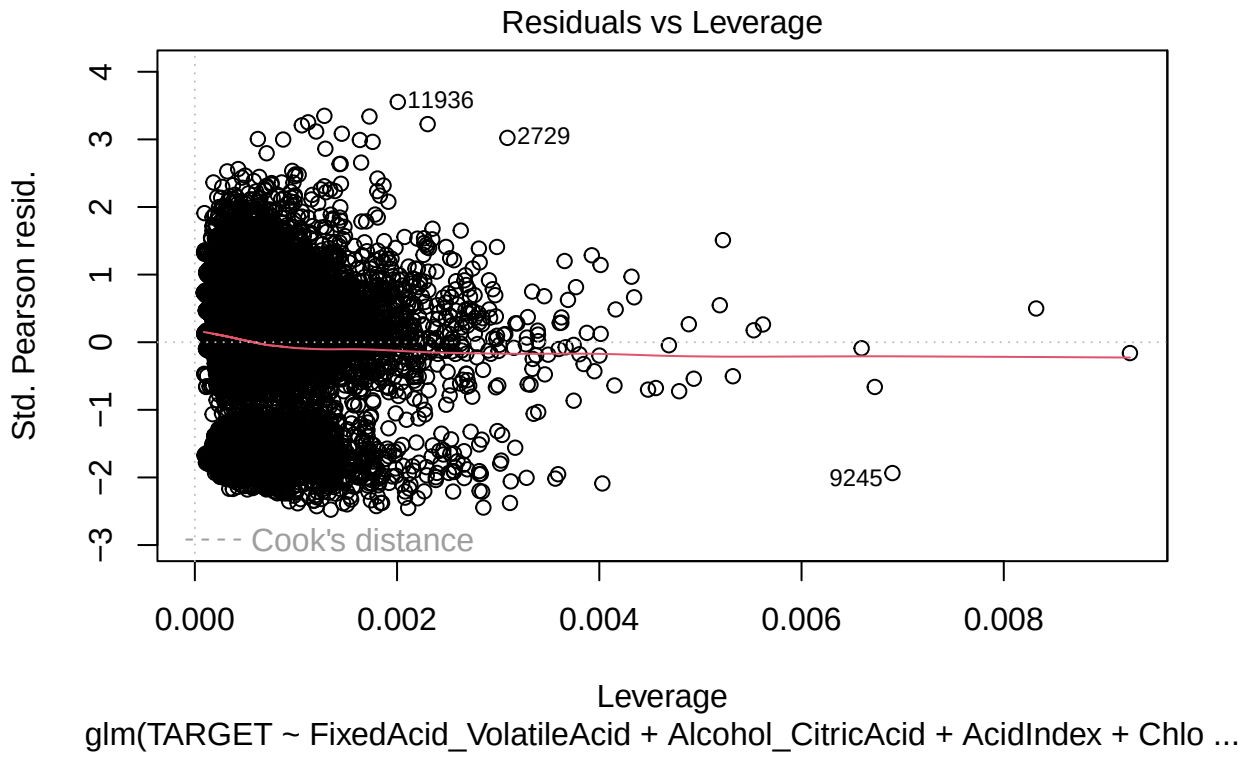



```
plot(poisson_2)
```

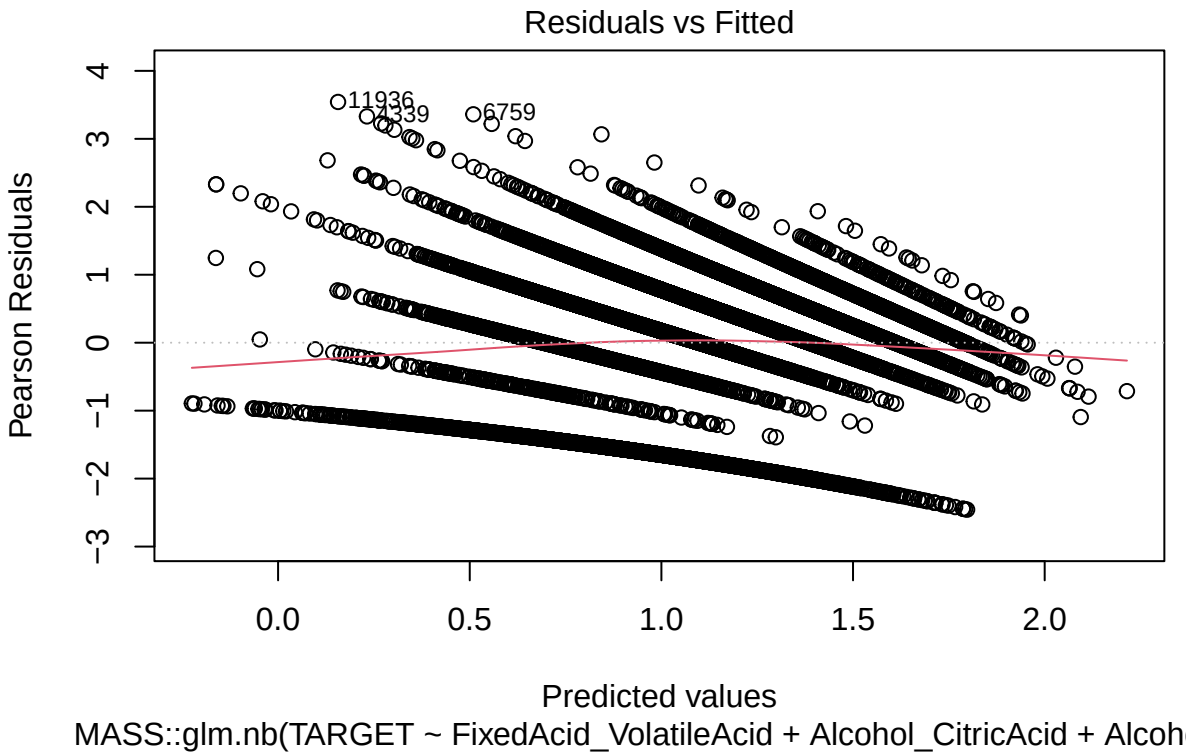

Residuals vs Fitted
Pearson Residuals
Predicted values
glm(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + AcidIndex + Chlo ...
11936
14339
6759
4
3
2
1
0
−1
−2
−3
0.0
0.5
1.0
1.5
2.0

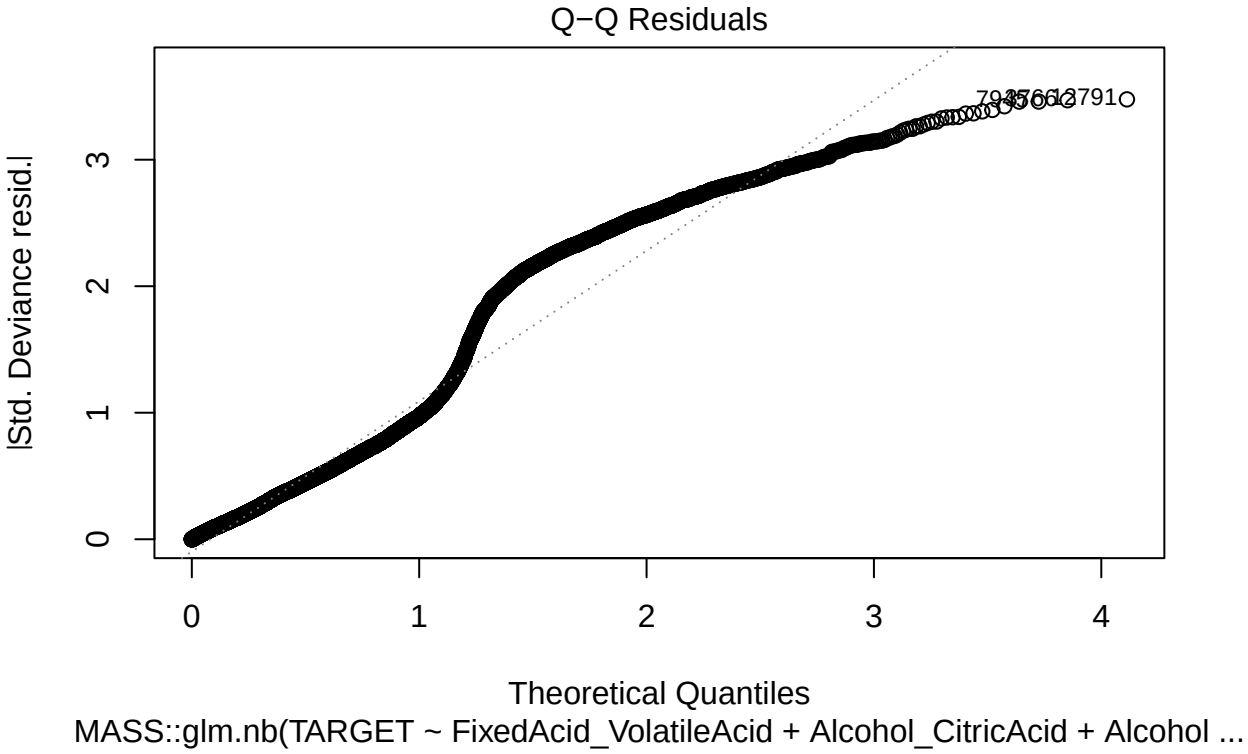

Q−Q Residuals
|Std. Deviance resid.|
0
1
2
3
0
1
2
3
4
Theoretical Quantiles
glm(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + AcidIndex + Chlo ...

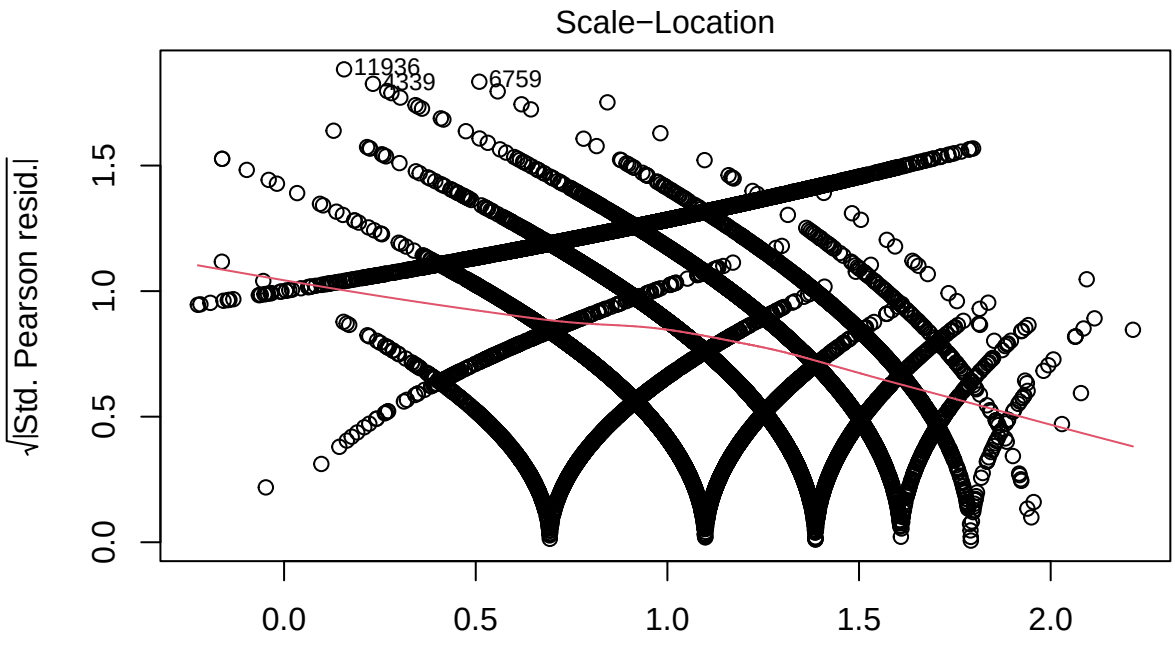

Scale−Location
11936
4339
6759
√|Std. Pearson resid.|
0.0
0.5
1.0
1.5
0.0
0.5
1.0
1.5
2.0
Predicted values
glm(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + AcidIndex + Chlo ...

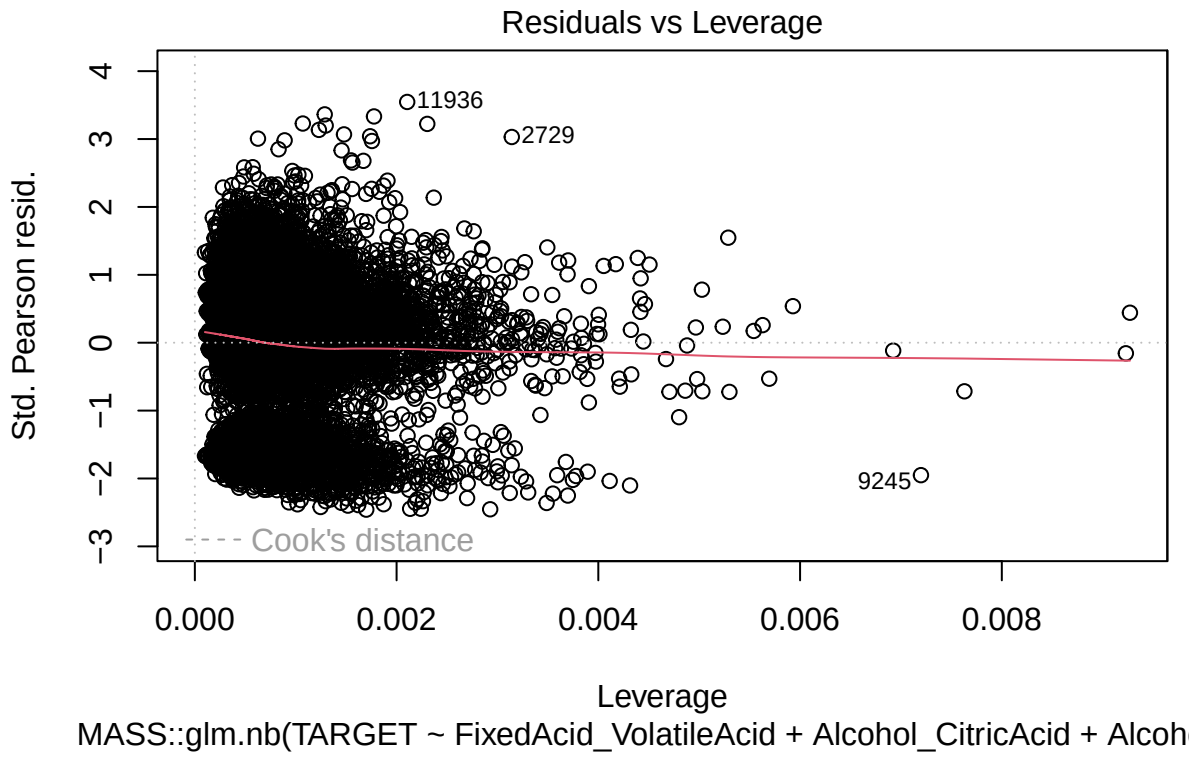

Residuals vs Leverage
Std. Pearson resid.
4
3
2
1
0
−1
−2
−3
11936
2729
9245
Cook's distance
0.000
0.002
0.004
0.006
0.008
Leverage
glm(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + AcidIndex + Chlo ...

```
plot(nb_1)
```

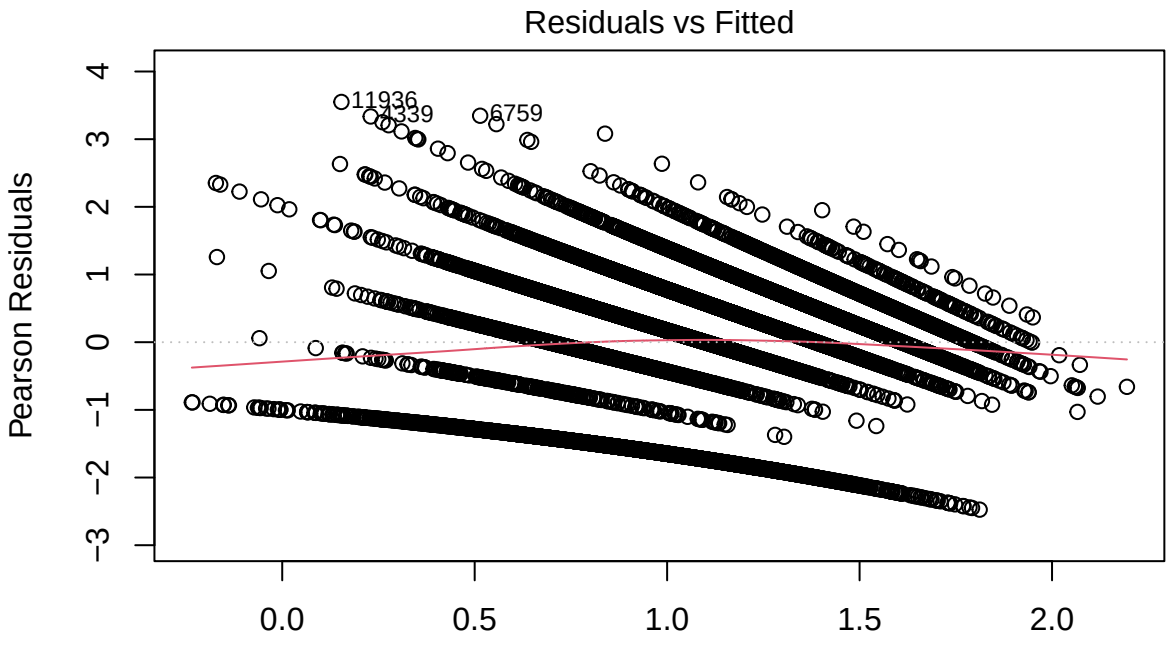

Residuals vs Fitted
Pearson Residuals
Predicted values
MASS::glm.nb(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + Alcohol ...
11936
4339
6759
4
3
2
1
0
−1
−2
−3
0.0
0.5
1.0
1.5
2.0


Q−Q Residuals
|Std. Deviance resid.|
0
1
2
3
0
1
2
3
4
Theoretical Quantiles
MASS::glm.nb(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + Alcohol ...

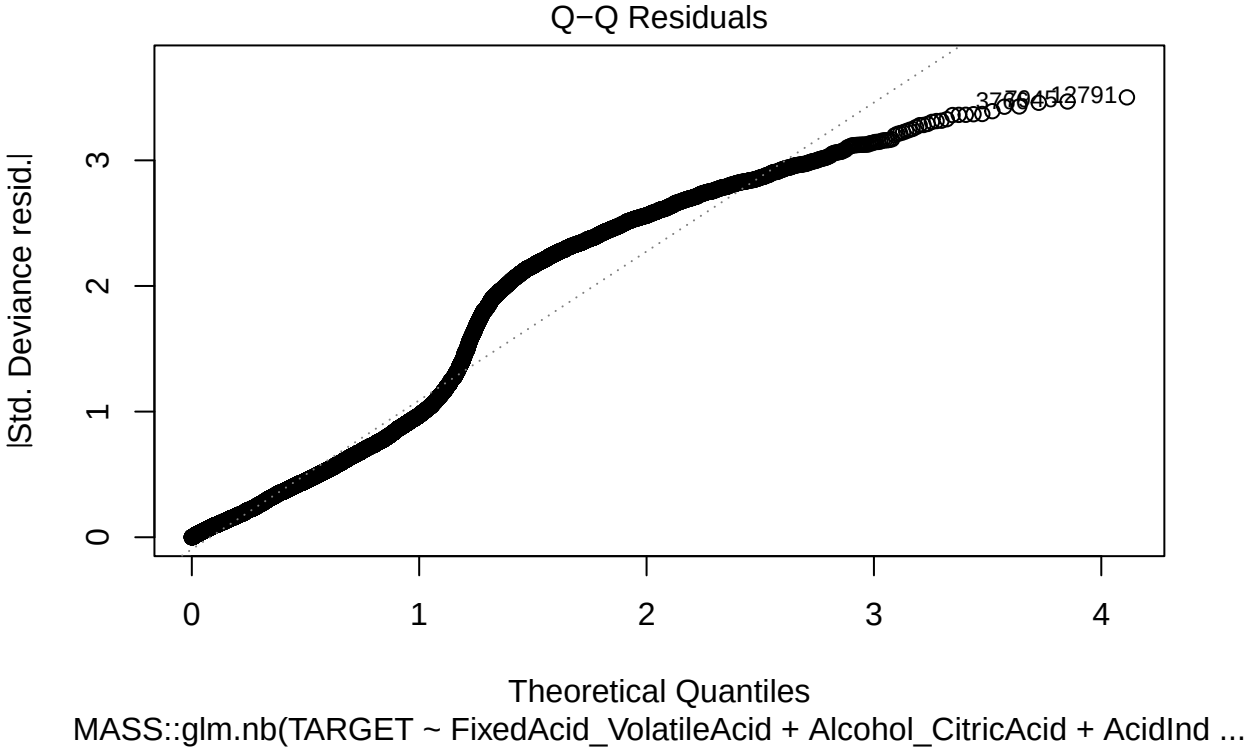

Scale−Location
11936
4339
6759
√|Std. Pearson resid.|
0.0
0.5
1.0
1.5
0.0
0.5
1.0
1.5
2.0
Predicted values
MASS::glm.nb(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + Alcohol ...

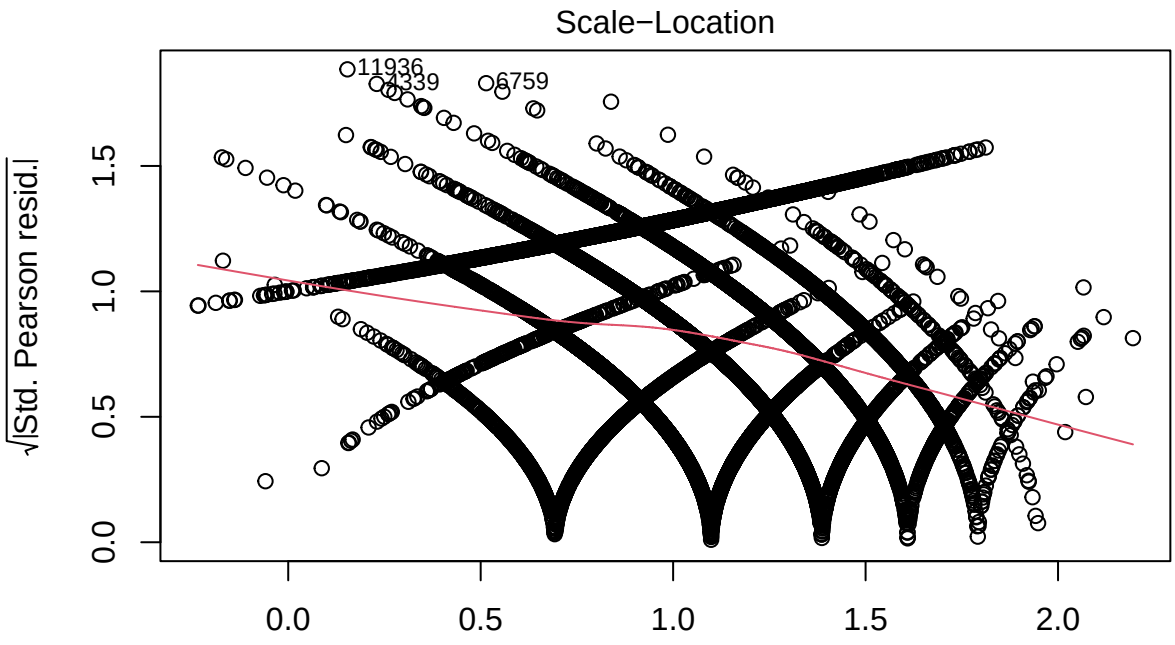

Residuals vs Leverage
Std. Pearson resid.
Leverage
MASS::glm.nb(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + Alcohol ...
11936
2729
9245
Cook's distance
4
3
2
1
0
−1
−2
−3
0.000
0.002
0.004
0.006
0.008

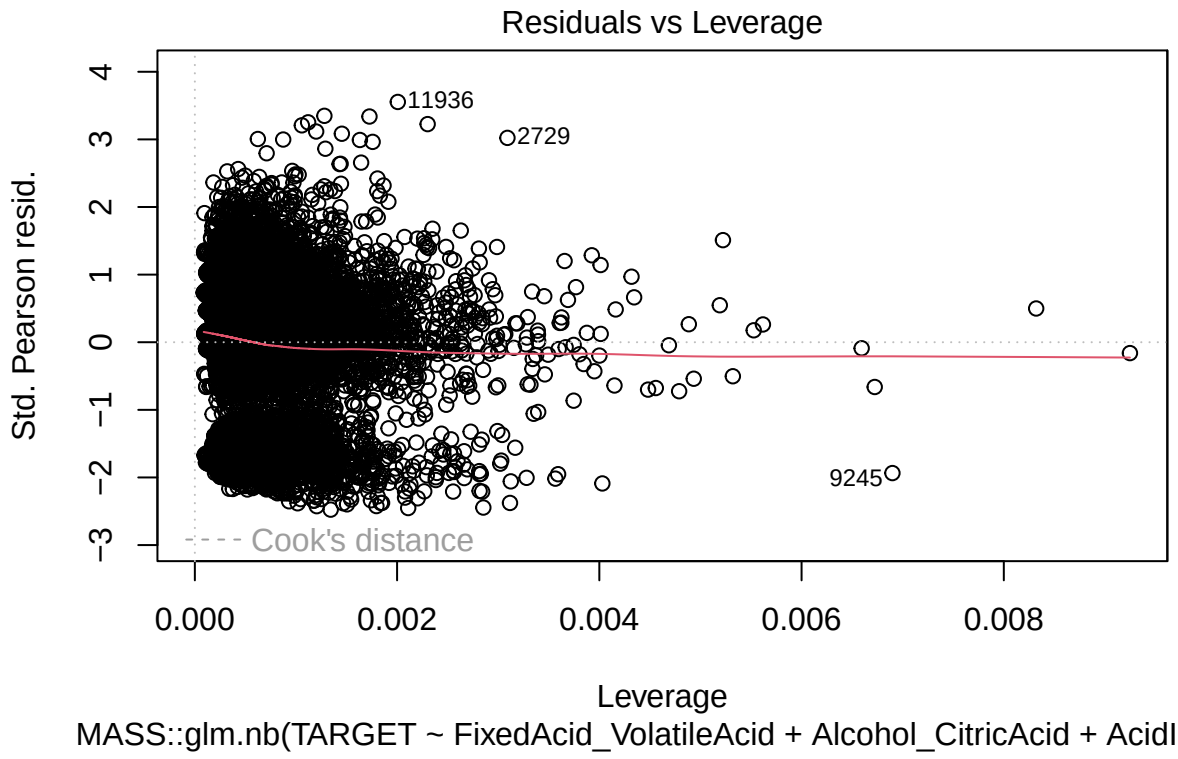
```
plot(nb_2)
```



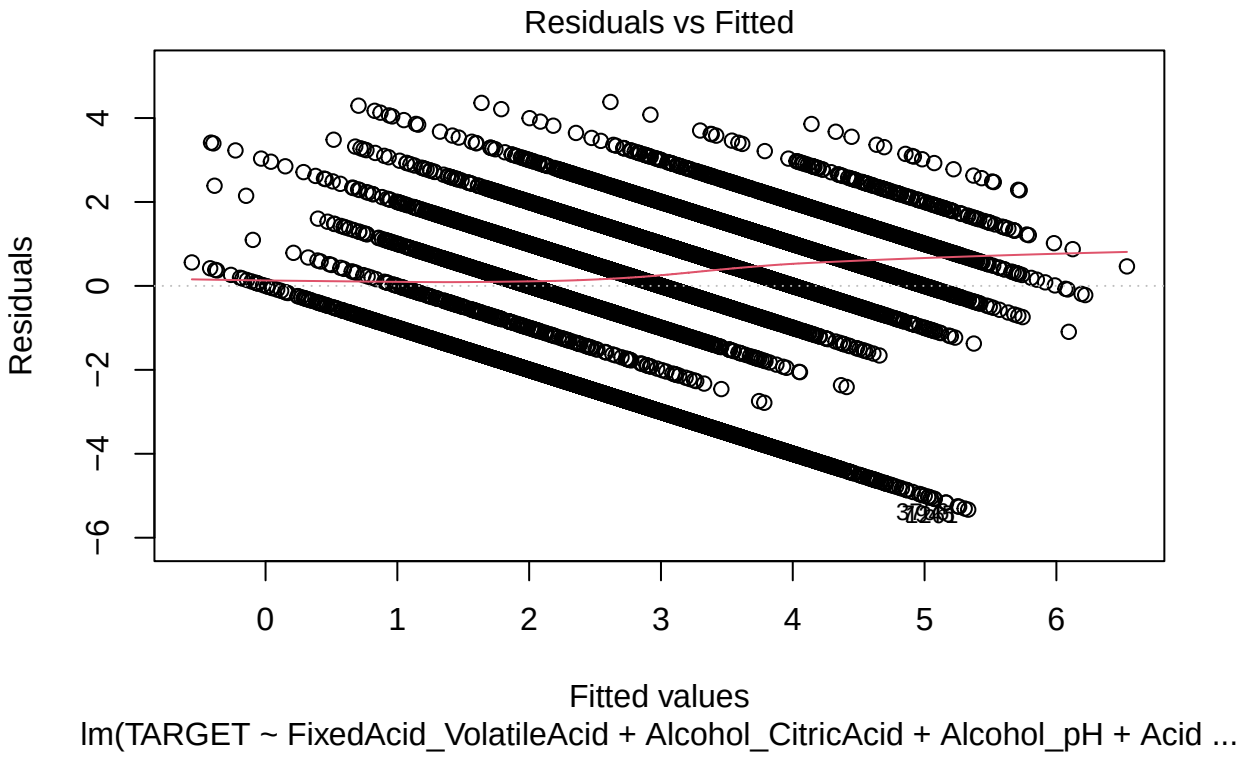
MASS::glm.nb(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + AcidInd ...

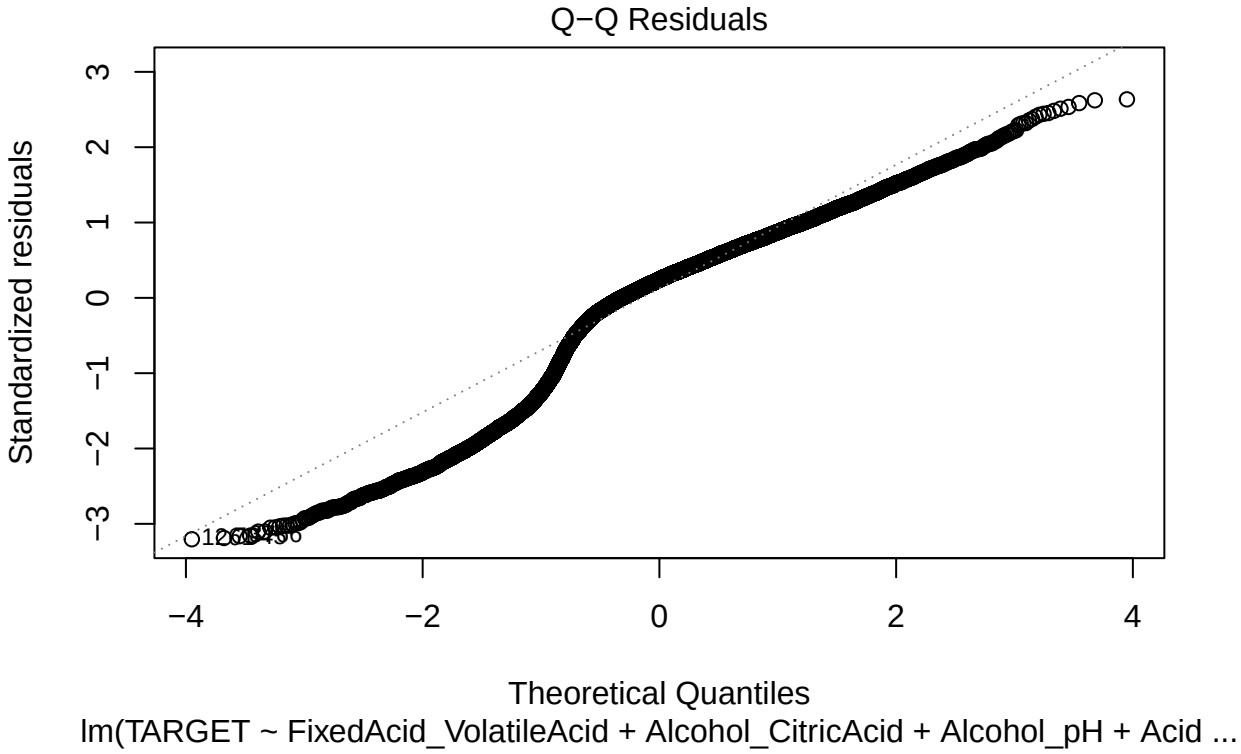

Q−Q Residuals
|Std. Deviance resid.|
0
1
2
3
4
Theoretical Quantiles
MASS::glm.nb(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + AcidInd ...

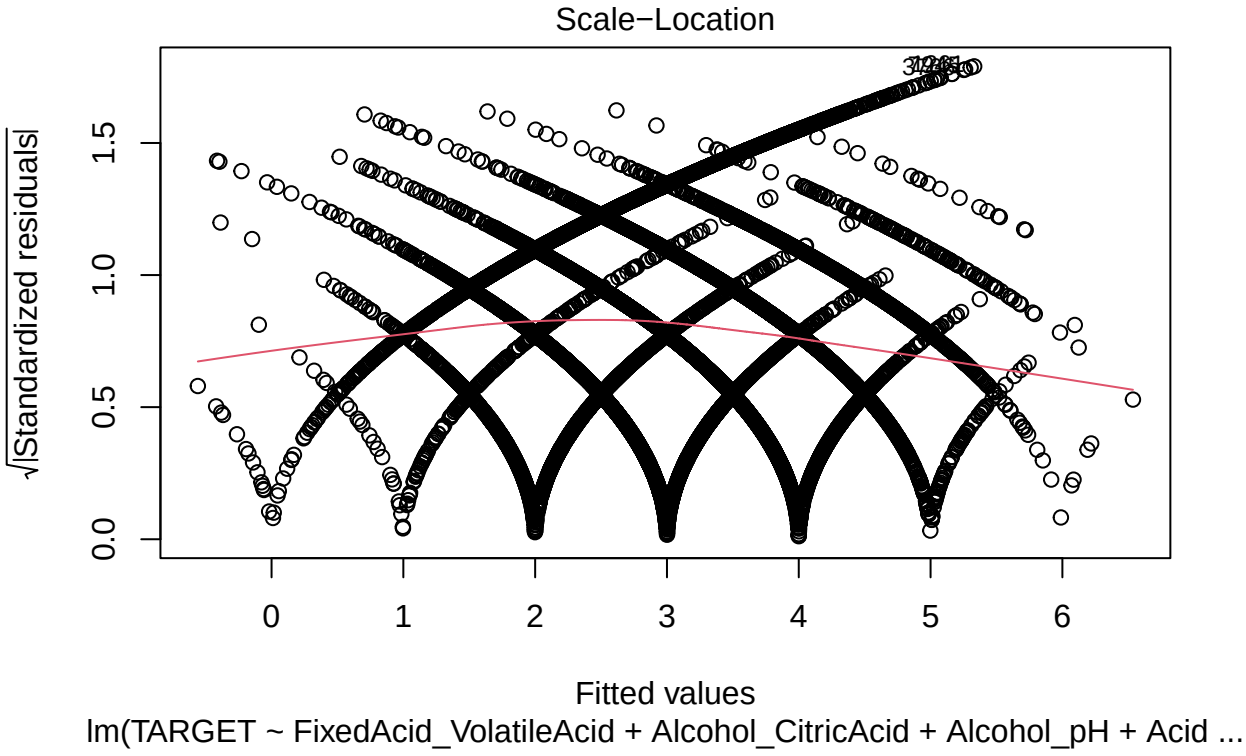

Scale−Location
11936
4339
6759
√|Std. Pearson resid.|
0.0
0.5
1.0
1.5
0.0
0.5
1.0
1.5
2.0
Predicted values
MASS::glm.nb(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + AcidInd ...

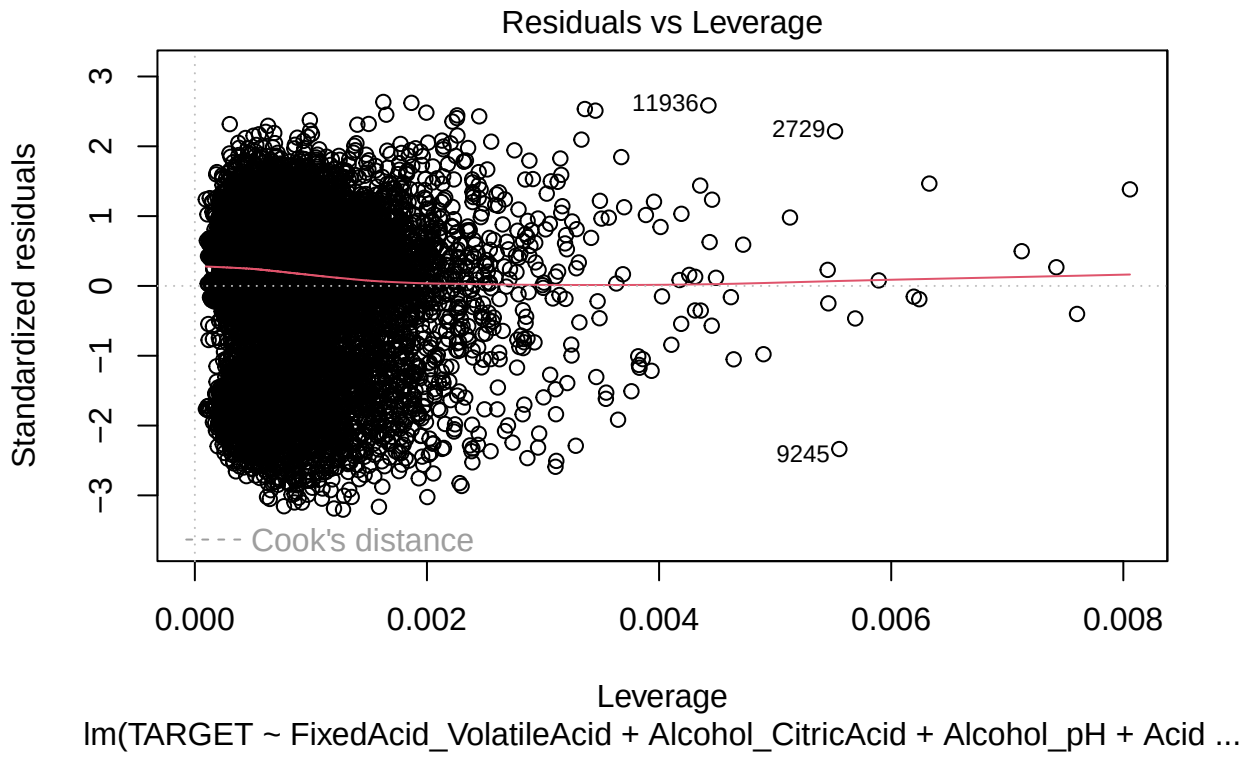

Residuals vs Leverage
Std. Pearson resid.
4
3
2
1
0
−1
−2
−3
11936
2729
9245
Cook's distance
0.000
0.002
0.004
0.006
0.008
Leverage
MASS::glm.nb(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + AcidInd ...

```
plot(ml_1)
```

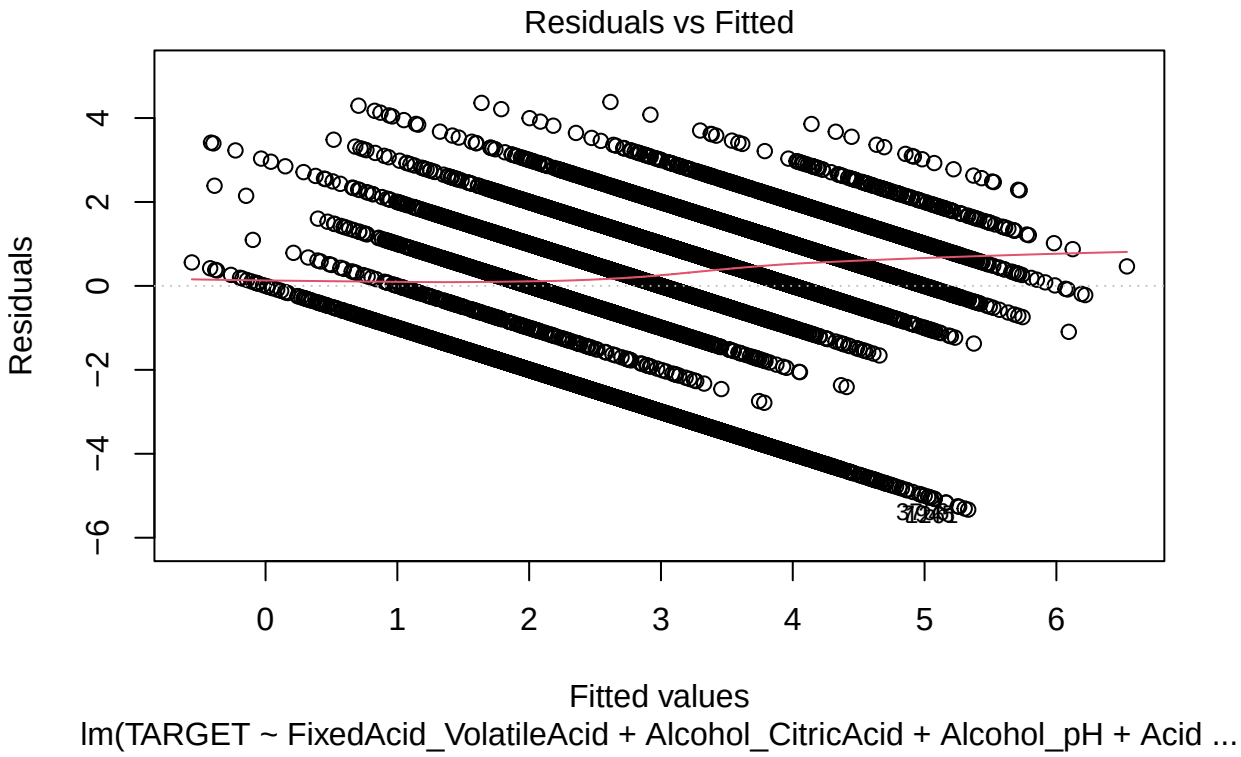

Residuals vs Fitted
Residuals
4
2
0
−2
−4
−6
0
1
2
3
4
5
6
Fitted values
lm(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + Alcohol_pH + Acid ...

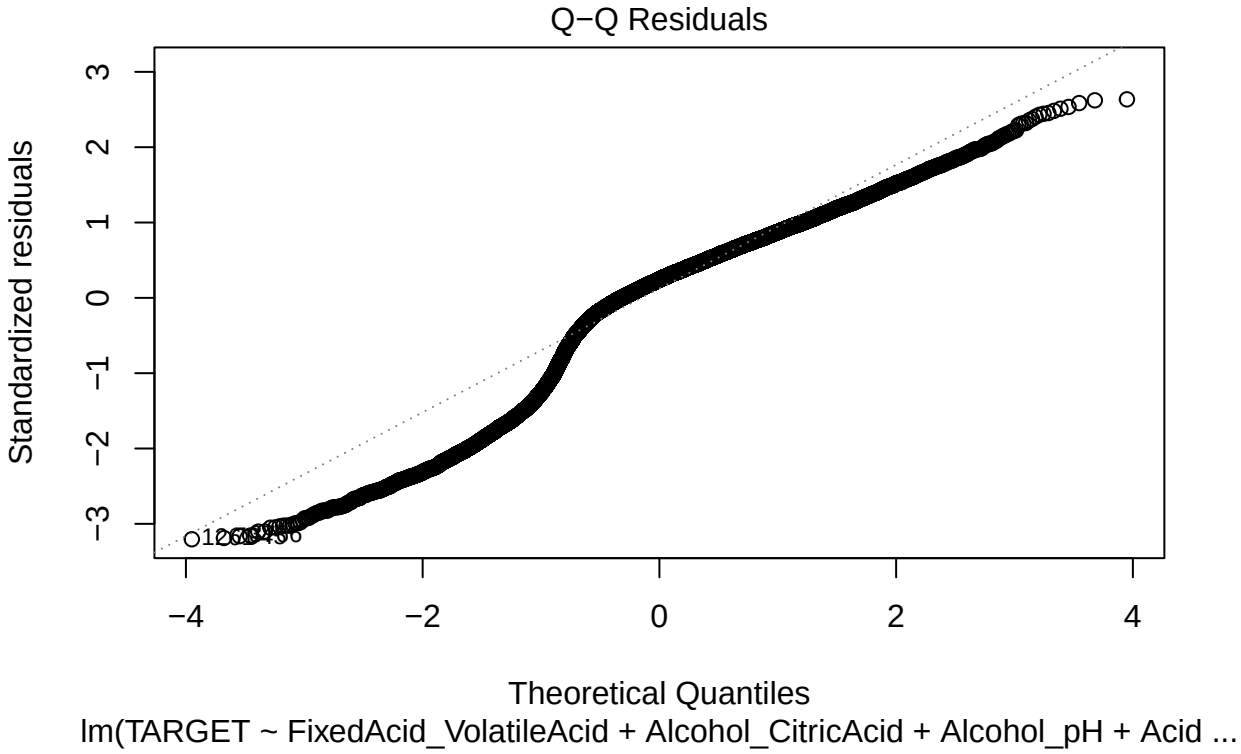

Q−Q Residuals
Standardized residuals
−3
−2
−1
0
1
2
3
−4
−2
0
2
4
Theoretical Quantiles
lm(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + Alcohol_pH + Acid ...

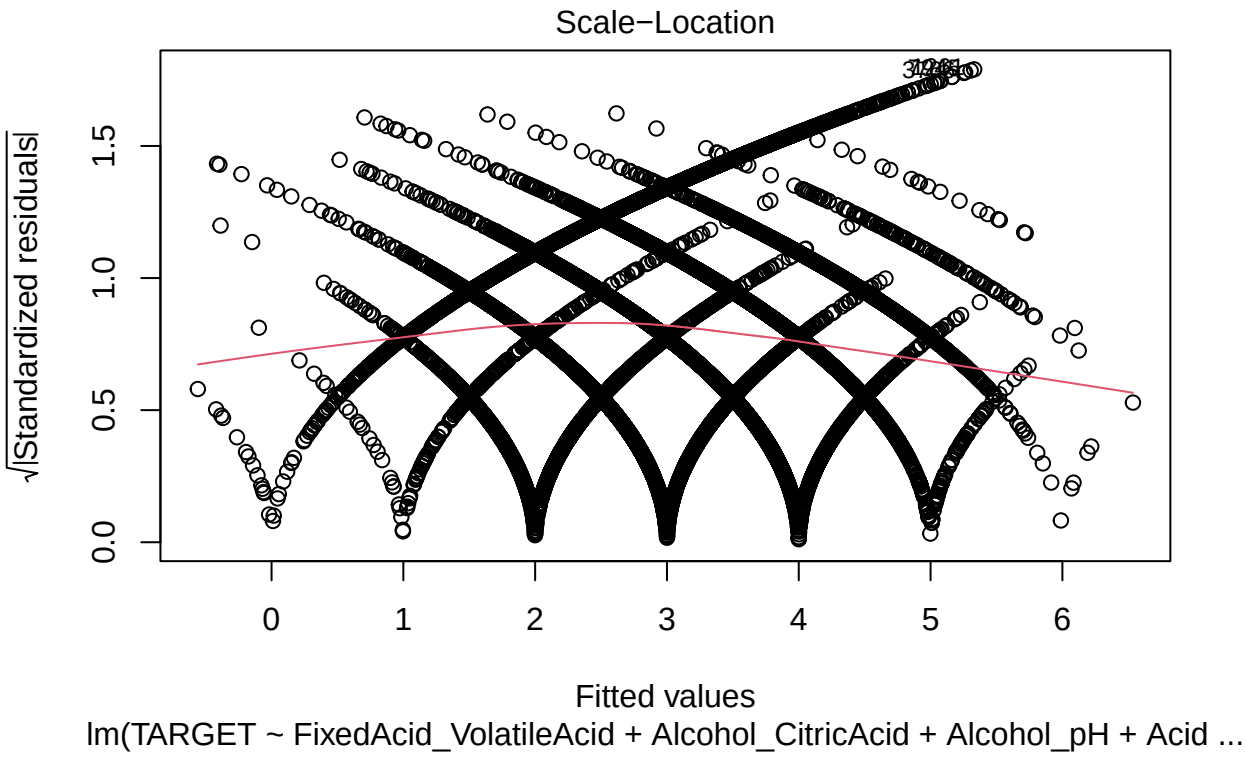

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
0
1
2
3
4
5
6
Fitted values
lm(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + Alcohol_pH + Acid ...

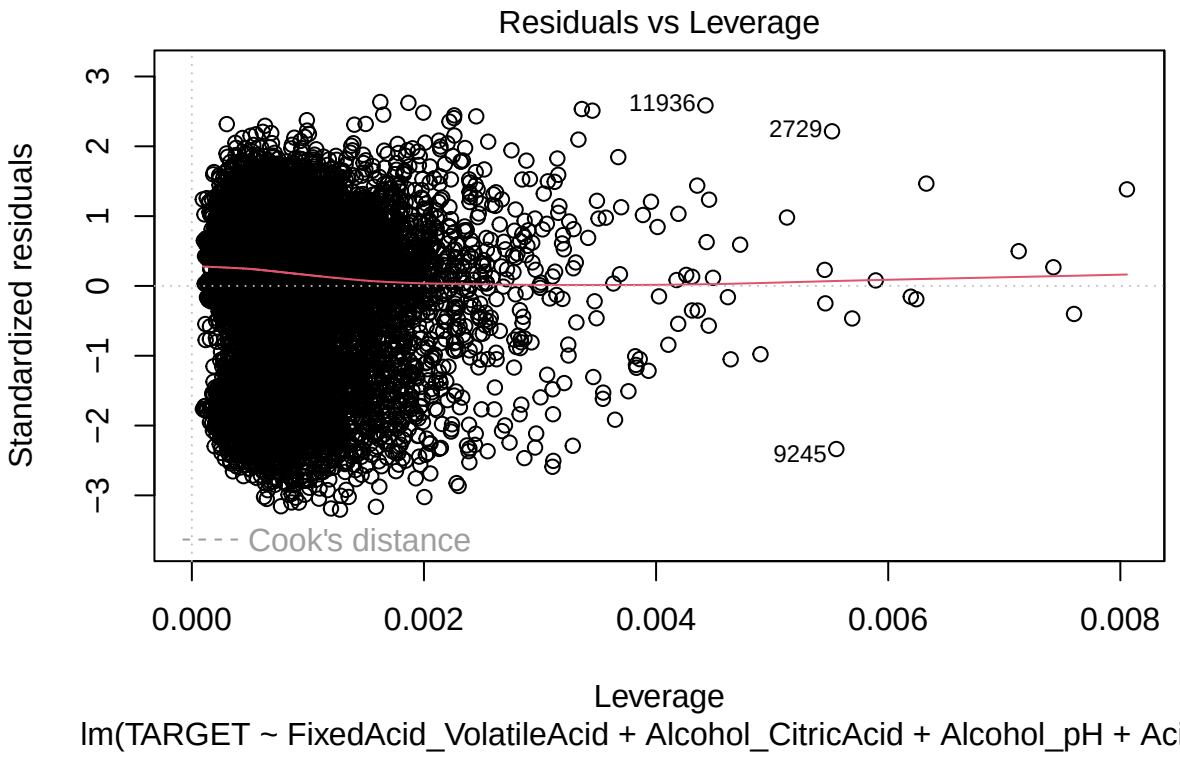

Residuals vs Leverage
Standardized residuals
3
2
1
0
−1
−2
−3
11936
2729
9245
Cook's distance
0.000
0.002
0.004
0.006
0.008
Leverage
lm(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + Alcohol_pH + Acid ...

```
plot(ml_2)
```


Residuals vs Fitted
Residuals
4
2
0
−2
−4
−6
0
1
2
3
4
5
6
Fitted values
lm(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + Alcohol_pH + Acid ...


Q−Q Residuals
Standardized residuals
−3
−2
−1
0
1
2
3
−4
−2
0
2
4
Theoretical Quantiles
lm(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + Alcohol_pH + Acid ...


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
0
1
2
3
4
5
6
Fitted values
lm(TARGET ~ FixedAcid_VolatileAcid + Alcohol_CitricAcid + Alcohol_pH + Acid ...



Based on the analyses performed, it appears that all count models generated exhibit similar performance at face value. Specifically, all models generated yielded similar deviance values, AIC values, and MSE values. That said, the negative binomial models tended to produce slightly better metrics, overall compared to the Poisson models. However, I do not believe a negative binomial model is particularly necessary for our data, as our target variable is likely more aligned with a Poisson or normal distribution. Furthermore, our model is not overdispersed, as evident by the fact that the residual and null deviance are smaller than their respective degrees of freedom. Finally, the maximum number of iterations were reached for both negative binomial models, suggesting difficulties in convergence. For these reasons, I will choose to use a Poisson model for our final evaluation, despite the fact that the negative binomial models present slightly better metrics. Specifically, I will use second poisson model, as the metrics for that model are slightly superior.

I will also mention that I do believe the linear regression models are good fits for our data, as well. As previously mentioned, I believe our target variable follows a rather normal distribution, as a result of the large sample size. In addition, our residual plots do show some violations in normality assumptions (i.e., some non-linearity, and the presence of outliers impacting skewness), but we can see that the assumptions are not overly violated and could likely be rectified via transformations or outlier removal. That said, the current linear models do appear to be poor fits, however, with R^2 values around .25.

Running Predictions on the Test Dataset

```
TARGET <- predict(poisson_2, newdata = wine_test_clean, type = 'response')

wine_test_clean <- wine_test_clean %>% dplyr::select(-TARGET)

wine_test_clean_scored <- cbind(wine_test_clean, TARGET)
```

```
hist(wine_test_clean_scored$TARGET)
```