HW5

Zachary Palmore

5/13/2021

Overview

```
library(tidyverse)
library(psych)
library(kableExtra)
library(caret)
library(reshape2)
library(ggpubr)
theme_set(theme_minimal())
```

In this homework assignment, you will explore, analyze and model a data set containing information on approximately 12,000 commercially available wines. The variables are mostly related to the chemical properties of the wine being sold. The response variable is the number of sample cases of wine that were purchased by wine distribution companies after sampling a wine. These cases would be used to provide tasting samples to restaurants and wine stores around the United States. The more sample cases purchased, the more likely is a wine to be sold at a high end restaurant. A large wine manufacturer is studying the data in order to predict the number of wine cases ordered based upon the wine characteristics. If the wine manufacturer can predict the number of cases, then that manufacturer will be able to adjust their wine offering to maximize sales.

Your objective is to build a count regression model to predict the number of cases of wine that will be sold given certain properties of the wine. HINT: Sometimes, the fact that a variable is missing is actually predictive of the target. You can only use the variables given to you (or variables that you derive from the variables provided).

Introduction

```
tdata <- read.csv(
  "https://raw.githubusercontent.com/palmorezm/msds/main/621/HW5/wine-training-data.csv")
edata <- read.csv(
  "https://raw.githubusercontent.com/palmorezm/msds/main/621/HW5/wine-evaluation-data.csv")
```

```
tdata[1:5,] %>%
  t() %>%
  kbl(booktabs = T, caption = "Raw Data") %>%
  kable_styling(latex_options = c("striped", "HOLD_position"), full_width = F) %>%
  footnote(c("Includes the initial observations of all variables in the data"))
```

Table 1: Raw Data

	1	2	3	4	5
ï..INDEX	1.0000	2.00000	4.00000	5.0000	6.00000
TARGET	3.0000	3.00000	5.00000	3.0000	4.00000
FixedAcidity	3.2000	4.50000	7.10000	5.7000	8.00000
VolatileAcidity	1.1600	0.16000	2.64000	0.3850	0.33000
CitricAcid	-0.9800	-0.81000	-0.88000	0.0400	-1.26000
ResidualSugar	54.2000	26.10000	14.80000	18.8000	9.40000
Chlorides	-0.5670	-0.42500	0.03700	-0.4250	NA
FreeSulfurDioxide	NA	15.00000	214.00000	22.0000	-167.00000
TotalSulfurDioxide	268.0000	-327.00000	142.00000	115.0000	108.00000
Density	0.9928	1.02792	0.99518	0.9964	0.99457
pH	3.3300	3.38000	3.12000	2.2400	3.12000
Sulphates	-0.5900	0.70000	0.48000	1.8300	1.77000
Alcohol	9.9000	NA	22.00000	6.2000	13.70000
LabelAppeal	0.0000	-1.00000	-1.00000	-1.0000	0.00000
AcidIndex	8.0000	7.00000	8.00000	6.0000	9.00000
STARS	2.0000	3.00000	3.00000	1.0000	2.00000

Note:
Includes the initial observations of all variables in the data

Exploration

```
tdata %>%
  describe() %>%
  round(digits = 1) %>%
  mutate(missing = 12975 - n) %>%
  select(n, missing, median, mean, sd, min, max, range, skew, se) %>%
  kbl(booktabs = T, caption = "Raw Summary") %>%
  kable_styling(latex_options = c("striped", "HOLD_position", "scale_down"), full_width = F) %>%
  column_spec(1, width = "8em") %>%
  footnote(c("Missing variables calculated based on the assumption of 12795 observations for each"))
```

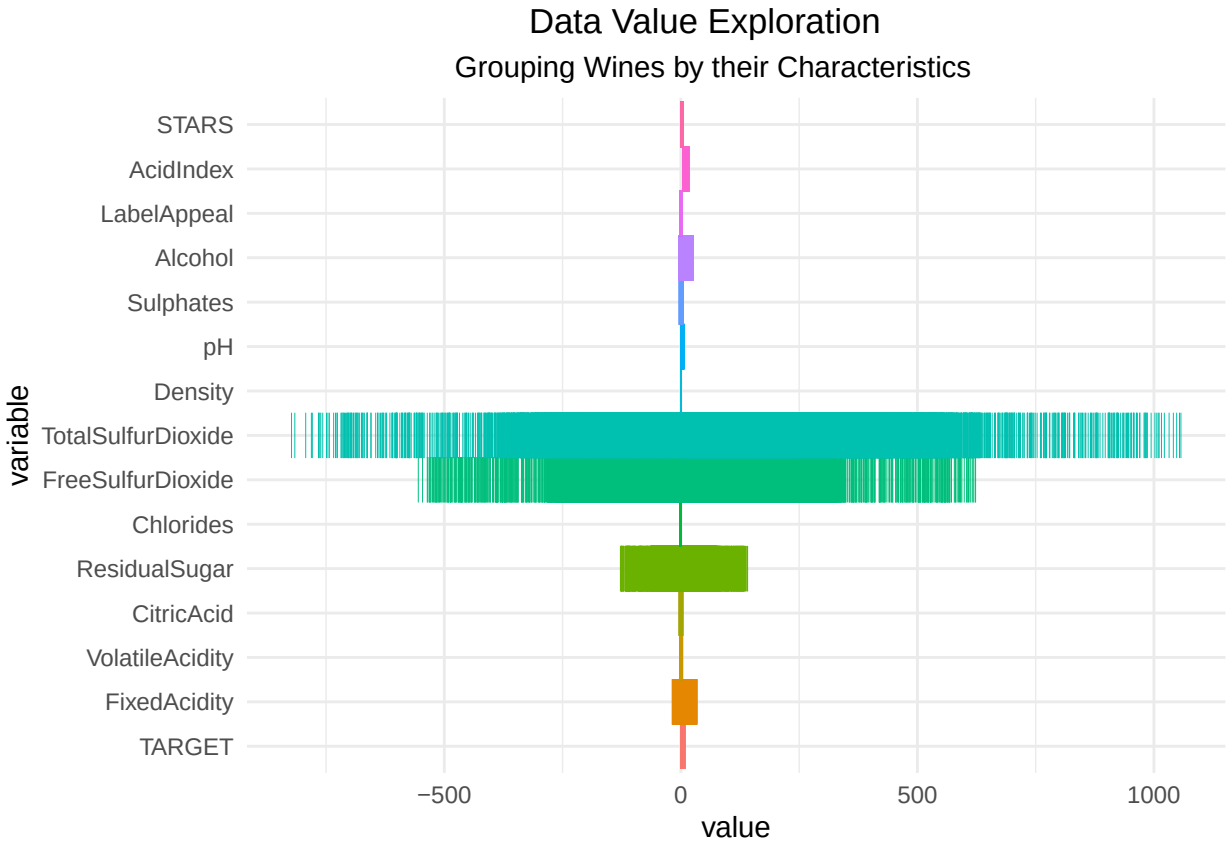
Table 2: Raw Summary

	n	missing	median	mean	sd	min	max	range	skew	se
ï..INDEX	12795	180	8110.0	8070.0	4656.9	1.0	16129.0	16128.0	0.0	41.2
TARGET	12795	180	3.0	3.0	1.9	0.0	8.0	8.0	-0.3	0.0
FixedAcidity	12795	180	6.9	7.1	6.3	-18.1	34.4	52.5	0.0	0.1
VolatileAcidity	12795	180	0.3	0.3	0.8	-2.8	3.7	6.5	0.0	0.0
CitricAcid	12795	180	0.3	0.3	0.9	-3.2	3.9	7.1	-0.1	0.0
ResidualSugar	12179	796	3.9	5.4	33.7	-127.8	141.2	269.0	-0.1	0.3
Chlorides	12157	818	0.0	0.1	0.3	-1.2	1.4	2.5	0.0	0.0
FreeSulfurDioxide	12148	827	30.0	30.8	148.7	-555.0	623.0	1178.0	0.0	1.3
TotalSulfurDioxide	12113	862	123.0	120.7	231.9	-823.0	1057.0	1880.0	0.0	2.1
Density	12795	180	1.0	1.0	0.0	0.9	1.1	0.2	0.0	0.0
pH	12400	575	3.2	3.2	0.7	0.5	6.1	5.7	0.0	0.0
Sulphates	11585	1390	0.5	0.5	0.9	-3.1	4.2	7.4	0.0	0.0
Alcohol	12142	833	10.4	10.5	3.7	-4.7	26.5	31.2	0.0	0.0
LabelAppeal	12795	180	0.0	0.0	0.9	-2.0	2.0	4.0	0.0	0.0
AcidIndex	12795	180	8.0	7.8	1.3	4.0	17.0	13.0	1.6	0.0
STARS	9436	3539	2.0	2.0	0.9	1.0	4.0	3.0	0.4	0.0

Note:
Missing variables calculated based on the assumption of 12795 observations for each

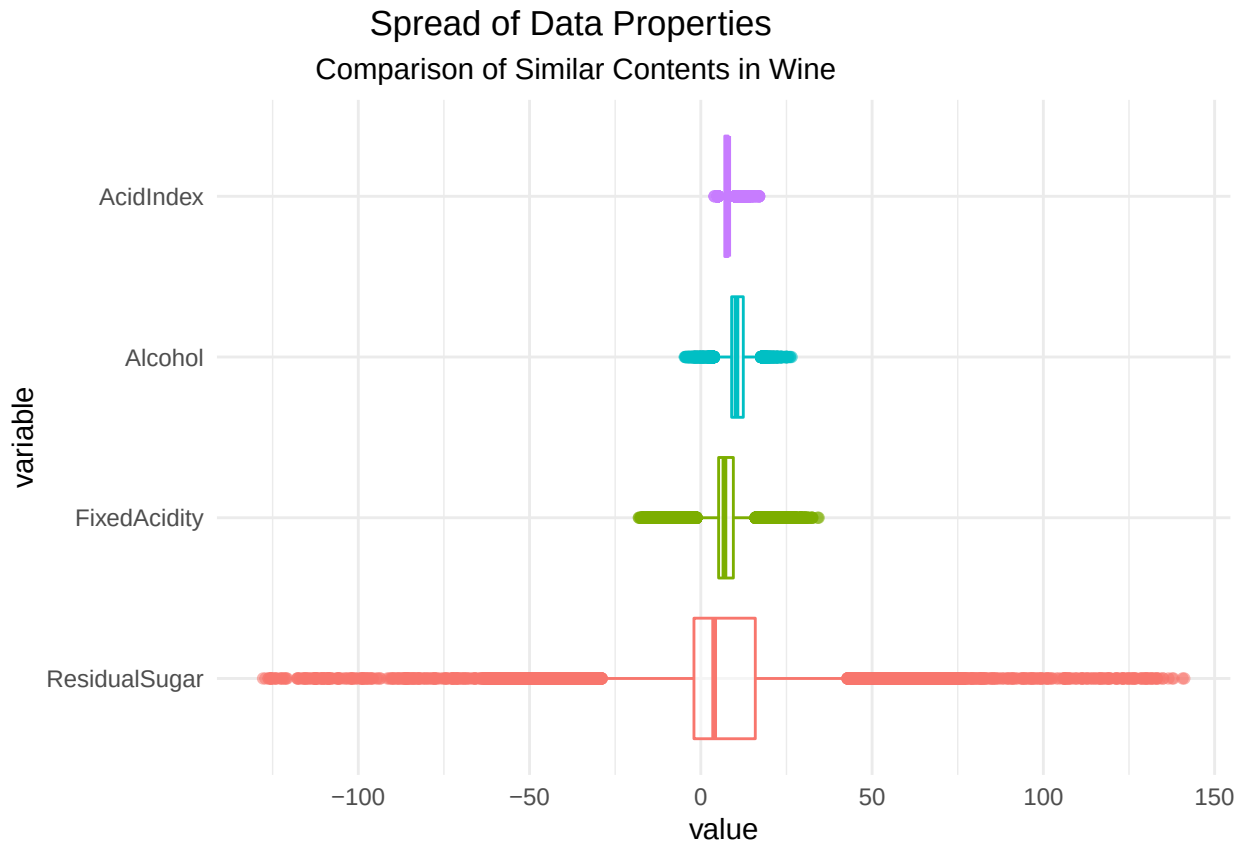
```
# Remove index variable
tdata <- tdata[-1]
```

```
tdata %>%
  melt() %>%
  ggplot(aes(variable, value, color = variable)) +
  geom_tile(aes()) + coord_flip() + ggtitle("Data Value Exploration", subtitle = "Grouping Wines by the
  theme(legend.position = "none", plot.title = element_text(hjust = 0.45), plot.subtitle = element_text
```

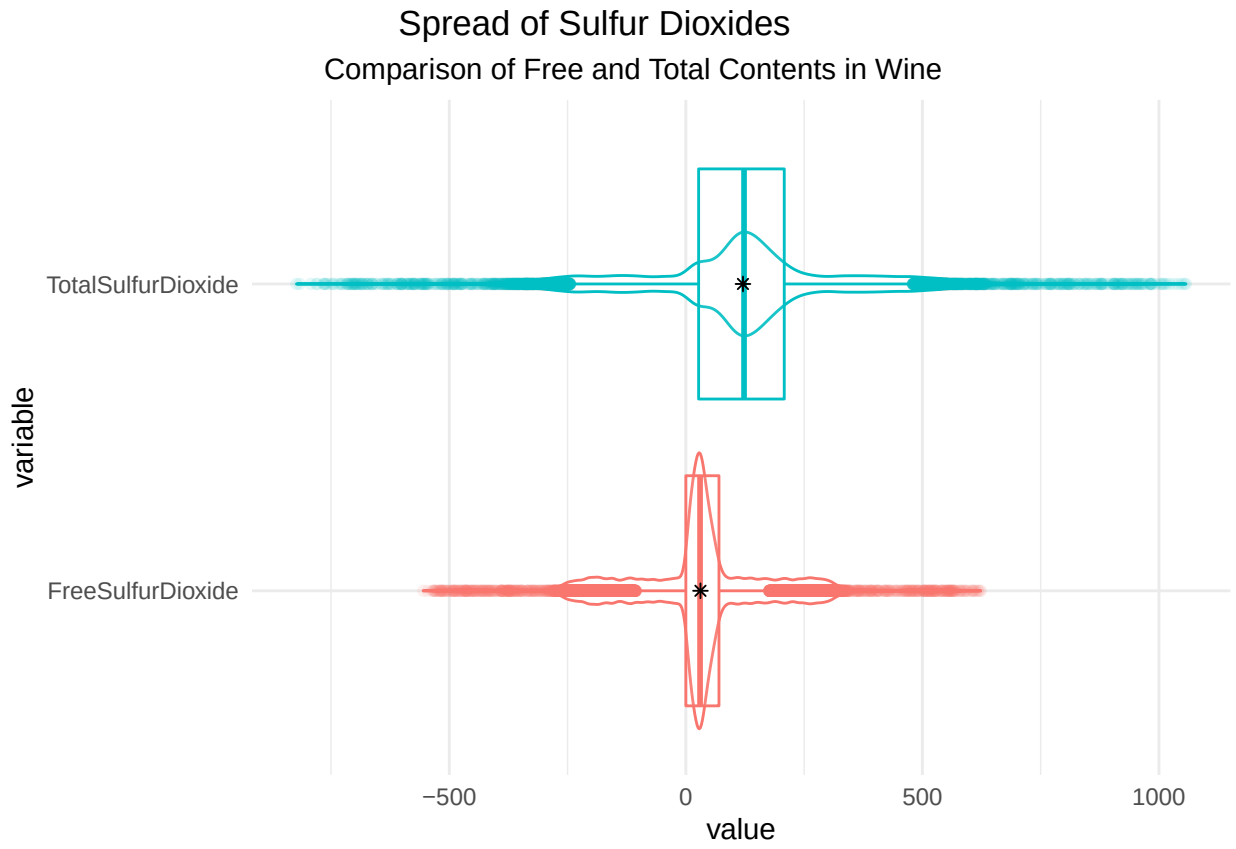



```
spread.misc <- tdata %>%
  select(-TotalSulfurDioxide, -FreeSulfurDioxide, -ResidualSugar, -FixedAcidity, -Alcohol, -AcidIndex) %
  melt() %>%
  ggplot() +
  geom_boxplot(aes(variable, value, alpha = .15, color=variable)) +
  coord_flip() +
  ggtitle("Spread of Data Properties", subtitle = "Comparison of Similar Contents in Wine") +
  theme(legend.position = "none", plot.title = element_text(hjust = 0.25), plot.subtitle = element_text
spread.misc
```

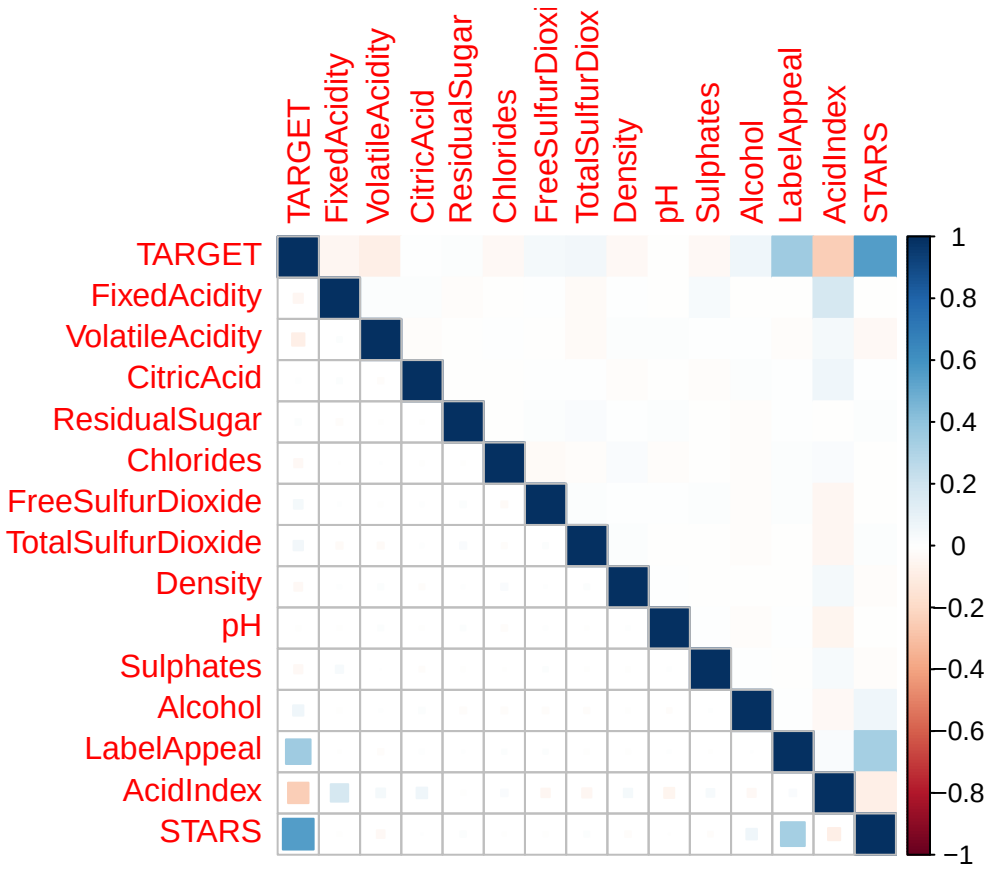

```
spread.ferments <- tdata %>%
  select(ResidualSugar, FixedAcidity, Alcohol, AcidIndex) %>%
  melt() %>%
  ggplot() +
  geom_boxplot(aes(variable, value, alpha = .15, color=variable)) +
  coord_flip() +
  ggtitle("Spread of Data Properties", subtitle = "Comparison of Similar Contents in Wine") +
  theme(legend.position = "none", plot.title = element_text(hjust = 0.25), plot.subtitle = element_text
spread.ferments
```




```
spread.dixoides <- tdata %>%
  select(FreeSulfurDioxide,
         TotalSulfurDioxide) %>%
  melt() %>%
  ggplot(aes(variable, value)) +
  geom_violin(aes(variable, value,
                  color = variable, alpha = 1)) +
  geom_boxplot(aes(alpha = .15,
                   color = variable, notch = TRUE)) +
  stat_summary(fun.y = mean, geom = "point",
               shape = 8, size = 1.5, color = "#000000") +
  coord_flip() +
  ggtitle("Spread of Sulfur Dioxides", subtitle = "Comparison of Free and Total Contents in Wine") +
  theme(legend.position = "none", plot.title = element_text(hjust = 0.25), plot.subtitle = element_text
spread.dixoides
```




```
corrplot::corrplot.mixed(cor(tdata, method = "pearson", use="pairwise.complete.obs"),
    bg = "light blue",
    addgrid.col = "black",
    diag = c("l"),
    lower.col = NULL,
    mar = c(0, 1, 0, 1),
    tl.pos = c("lt"),
    lower = "square",
    upper = "color",
    plotCI = "n")
```



Preparation

```
# Consider realistic values and adjust accordingly
tdata <- abs(tdata)
tdata %>%
  describe() %>%
  round(digits = 1) %>%
  mutate(missing = 12975 - n) %>%
  select(n, missing, median, mean, sd, min, max, range, skew, se) %>%
  kbl(booktabs = T, caption = "Updated Summary") %>%
  kable_styling(latex_options = c("striped", "HOLD_position", "scale_down"), full_width = F) %>%
  column_spec(1, width = "8em") %>%
  footnote(c("Minimum values were adjusted where applicable to describe the data realisticly"))
```

Table 3: Updated Summary

	n	missing	median	mean	sd	min	max	range	skew	se
TARGET	12795	180	3.0	3.0	1.9	0.0	8.0	8.0	-0.3	0.0
FixedAcidity	12795	180	7.0	8.1	5.0	0.0	34.4	34.4	1.2	0.0
VolatileAcidity	12795	180	0.4	0.6	0.6	0.0	3.7	3.7	1.7	0.0
CitricAcid	12795	180	0.4	0.7	0.6	0.0	3.9	3.9	1.6	0.0
ResidualSugar	12179	796	12.9	23.4	24.9	0.0	141.2	141.2	1.5	0.2
Chlorides	12157	818	0.1	0.2	0.2	0.0	1.4	1.4	1.5	0.0
FreeSulfurDioxide	12148	827	56.0	106.7	108.1	0.0	623.0	623.0	1.5	1.0
TotalSulfurDioxide	12113	862	154.0	204.3	163.1	0.0	1057.0	1057.0	1.6	1.5
Density	12795	180	1.0	1.0	0.0	0.9	1.1	0.2	0.0	0.0
pH	12400	575	3.2	3.2	0.7	0.5	6.1	5.7	0.0	0.0
Sulphates	11585	1390	0.6	0.8	0.7	0.0	4.2	4.2	1.7	0.0
Alcohol	12142	833	10.4	10.5	3.6	0.0	26.5	26.5	0.2	0.0
LabelAppeal	12795	180	1.0	0.6	0.6	0.0	2.0	2.0	0.4	0.0
AcidIndex	12795	180	8.0	7.8	1.3	4.0	17.0	13.0	1.6	0.0
STARS	9436	3539	2.0	2.0	0.9	1.0	4.0	3.0	0.4	0.0

Note:

Minimum values were adjusted where applicable to describe the data realisticly

```
# Impute missing values
tdata <- tdata %>%
  mutate(
  ResidualSugar = ifelse(is.na(ResidualSugar), median(ResidualSugar, na.rm = T), ResidualSugar),
  Chlorides = ifelse(is.na(Chlorides), median(Chlorides, na.rm = T), Chlorides),
  FreeSulfurDioxide = ifelse(is.na(FreeSulfurDioxide), median(FreeSulfurDioxide, na.rm = T), FreeSulfurl
  TotalSulfurDioxide = ifelse(is.na(TotalSulfurDioxide), median(TotalSulfurDioxide, na.rm = T), TotalSul
  pH = ifelse(is.na(pH), median(pH, na.rm = T), pH),
  Sulphates = ifelse(is.na(Sulphates), median(Sulphates, na.rm = T), Sulphates),
  Alcohol = ifelse(is.na(Alcohol), median(Alcohol, na.rm = T), Alcohol),
  STARS_imputed = ifelse(is.na(STARS), 1, 0),
  STARS = ifelse(is.na(STARS), 1, STARS))
```

```
set.seed(1225)
train_index <- createDataPartition(tdata$TARGET, p = .7, list = FALSE, times = 1)
train <- tdata[train_index,]
eval <- tdata[-train_index,]
```

```
train[1:5,] %>%
  t() %>%
  kbl(booktabs = T, caption = "Training Data") %>%
  kable_styling(latex_options = c("striped", "HOLD_position"), full_width = F) %>%
  footnote(c("Includes the initial observations of all variables in the data"))
```

Table 4: Training Data

	4	5	6	8	10
TARGET	3.0000	4.00000	0.0000	4.00000	6.00000
FixedAcidity	5.7000	8.00000	11.3000	6.50000	5.50000
VolatileAcidity	0.3850	0.33000	0.3200	1.22000	0.22000
CitricAcid	0.0400	1.26000	0.5900	0.34000	0.39000
ResidualSugar	18.8000	9.40000	2.2000	1.40000	1.80000
Chlorides	0.4250	0.09800	0.5560	0.04000	0.27700
FreeSulfurDioxide	22.0000	167.00000	37.0000	523.00000	62.00000
TotalSulfurDioxide	115.0000	108.00000	15.0000	551.00000	180.00000
Density	0.9964	0.99457	0.9994	1.03236	0.94724
pH	2.2400	3.12000	3.2000	3.20000	3.09000
Sulphates	1.8300	1.77000	1.2900	0.59000	0.75000
Alcohol	6.2000	13.70000	15.4000	11.60000	12.60000
LabelAppeal	1.0000	0.00000	0.0000	1.00000	0.00000
AcidIndex	6.0000	9.00000	11.0000	7.00000	8.00000
STARS	1.0000	2.00000	1.0000	3.00000	4.00000
STARS_imputed	0.0000	0.00000	1.0000	0.00000	0.00000

Note:

Includes the initial observations of all variables in the data

```
eval[1:5,] %>%
  t() %>%
  kbl(booktabs = T, caption = "Evaluation Data") %>%
  kable_styling(latex_options = c("striped", "HOLD_position"), full_width = F) %>%
  footnote(c("Includes the initial observations of all variables in the data"))
```

Table 5: Evaluation Data

	1	2	3	7	9
TARGET	3.0000	3.00000	5.00000	0.00000	3.0000
FixedAcidity	3.2000	4.50000	7.10000	7.70000	14.8000
VolatileAcidity	1.1600	0.16000	2.64000	0.29000	0.2700
CitricAcid	0.9800	0.81000	0.88000	0.40000	1.0500
ResidualSugar	54.2000	26.10000	14.80000	21.50000	11.2500
Chlorides	0.5670	0.42500	0.03700	0.06000	0.0070
FreeSulfurDioxide	56.0000	15.00000	214.00000	287.00000	213.0000
TotalSulfurDioxide	268.0000	327.00000	142.00000	156.00000	154.0000
Density	0.9928	1.02792	0.99518	0.99572	0.9962
pH	3.3300	3.38000	3.12000	3.49000	4.9300
Sulphates	0.5900	0.70000	0.48000	1.21000	0.2600
Alcohol	9.9000	10.40000	22.00000	10.30000	15.0000
LabelAppeal	0.0000	1.00000	1.00000	0.00000	0.0000
AcidIndex	8.0000	7.00000	8.00000	8.00000	6.0000
STARS	2.0000	3.00000	3.00000	1.00000	1.0000
STARS_imputed	0.0000	0.00000	0.00000	1.00000	1.0000

Note:
Includes the initial observations of all variables in the data

```
train %>%
  describe() %>%
  round(digits = 1) %>%
  mutate(missing = 8958 - n) %>%
  select(n, missing, median, mean, sd, min, max, range, skew, se) %>%
  kbl(booktabs = T, caption = "Raw Summary") %>%
  kable_styling(latex_options = c("striped", "HOLD_position", "scale_down"), full_width = F) %>%
  column_spec(1, width = "8em") %>%
  footnote(c("Missing variables calculated based on the assumption of 8958 observations for each"))
```

Table 6: Raw Summary

	n	missing	median	mean	sd	min	max	range	skew	se
TARGET	8958	0	3.0	3.0	1.9	0.0	8.0	8.0	-0.3	0.0
FixedAcidity	8958	0	7.0	8.1	5.0	0.0	34.4	34.4	1.2	0.1
VolatileAcidity	8958	0	0.4	0.6	0.6	0.0	3.7	3.7	1.7	0.0
CitricAcid	8958	0	0.4	0.7	0.6	0.0	3.9	3.9	1.6	0.0
ResidualSugar	8958	0	12.9	23.0	24.4	0.0	140.7	140.7	1.5	0.3
Chlorides	8958	0	0.1	0.2	0.2	0.0	1.4	1.4	1.5	0.0
FreeSulfurDioxide	8958	0	56.0	104.3	106.1	0.0	623.0	623.0	1.6	1.1
TotalSulfurDioxide	8958	0	154.0	203.0	160.4	0.0	1057.0	1057.0	1.7	1.7
Density	8958	0	1.0	1.0	0.0	0.9	1.1	0.2	0.0	0.0
pH	8958	0	3.2	3.2	0.7	0.5	6.0	5.6	0.0	0.0
Sulphates	8958	0	0.6	0.8	0.6	0.0	4.2	4.2	1.8	0.0
Alcohol	8958	0	10.4	10.5	3.5	0.0	26.5	26.5	0.2	0.0
LabelAppeal	8958	0	1.0	0.6	0.6	0.0	2.0	2.0	0.4	0.0
AcidIndex	8958	0	8.0	7.8	1.3	4.0	17.0	13.0	1.7	0.0
STARS	8958	0	1.0	1.8	0.9	1.0	4.0	3.0	0.9	0.0
STARS_imputed	8958	0	0.0	0.3	0.4	0.0	1.0	1.0	1.1	0.0

Note:

Missing variables calculated based on the assumption of 8958 observations for each

```
eval %>%
  describe() %>%
  round(digits = 1) %>%
  mutate(missing = 3837 - n) %>%
  select(n, missing, median, mean, sd, min, max, range, skew, se) %>%
  kbl(booktabs = T, caption = "Raw Summary") %>%
  kable_styling(latex_options = c("striped", "HOLD_position", "scale_down"), full_width = F) %>%
  column_spec(1, width = "8em") %>%
  footnote(c("Missing variables calculated based on the assumption of 3837 observations for each"))
```

Table 7: Raw Summary

	n	missing	median	mean	sd	min	max	range	skew	se
TARGET	3837	0	3.0	3.0	1.9	0.0	8.0	8.0	-0.3	0.0
FixedAcidity	3837	0	7.0	8.1	4.9	0.0	34.1	34.1	1.1	0.1
VolatileAcidity	3837	0	0.4	0.6	0.6	0.0	3.5	3.5	1.6	0.0
CitricAcid	3837	0	0.4	0.7	0.6	0.0	3.8	3.8	1.6	0.0
ResidualSugar	3837	0	12.9	22.5	24.6	0.0	141.2	141.2	1.6	0.4
Chlorides	3837	0	0.1	0.2	0.2	0.0	1.3	1.3	1.6	0.0
FreeSulfurDioxide	3837	0	56.0	103.6	105.5	0.0	618.0	618.0	1.6	1.7
TotalSulfurDioxide	3837	0	154.0	198.4	156.0	0.0	1054.0	1054.0	1.7	2.5
Density	3837	0	1.0	1.0	0.0	0.9	1.1	0.2	0.0	0.0
pH	3837	0	3.2	3.2	0.7	0.6	6.1	5.5	0.1	0.0
Sulphates	3837	0	0.6	0.8	0.6	0.0	4.2	4.2	1.9	0.0
Alcohol	3837	0	10.4	10.5	3.5	0.1	26.0	25.9	0.2	0.1
LabelAppeal	3837	0	1.0	0.7	0.6	0.0	2.0	2.0	0.4	0.0
AcidIndex	3837	0	8.0	7.8	1.3	5.0	17.0	12.0	1.6	0.0
STARS	3837	0	2.0	1.8	0.9	1.0	4.0	3.0	0.9	0.0
STARS_imputed	3837	0	0.0	0.3	0.4	0.0	1.0	1.0	1.1	0.0

Note:
Missing variables calculated based on the assumption of 3837 observations for each

Model Building

```
model1 <- glm(TARGET ~ FixedAcidity + VolatileAcidity + Alcohol, family = quasipoisson, train)
summary(model1)
```

```
##
## Call:
## glm(formula = TARGET ~ FixedAcidity + VolatileAcidity + Alcohol,
##     family = quasipoisson, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7215  -0.7003   0.1346   0.7012   2.4663
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.099366   0.024787  44.352  < 2e-16 ***
## FixedAcidity     -0.006599   0.001353  -4.877 1.10e-06 ***
## VolatileAcidity -0.083292   0.012524  -6.651 3.09e-11 ***
## Alcohol           0.010835   0.001881   5.760 8.67e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1.207516)
##
##     Null deviance: 15871  on 8957  degrees of freedom
```

```
## Residual deviance: 15747  on 8954  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```
train1 <- train %>%
  select(-LabelAppeal, -AcidIndex, -STARS, -STARS_imputed)
model2 <- glm(TARGET ~ ., family = quasipoisson, train1)
summary(model2)
```

```
##
## Call:
## glm(formula = TARGET ~ ., family = quasipoisson, data = train1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8123  -0.7224   0.1528   0.7229   2.5676
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          2.181e+00  2.537e-01   8.598  < 2e-16 ***
## FixedAcidity        -6.430e-03  1.355e-03  -4.744 2.12e-06 ***
## VolatileAcidity     -8.070e-02  1.254e-02  -6.437 1.28e-10 ***
## CitricAcid           7.202e-03  1.096e-02   0.657  0.51104
## ResidualSugar        6.899e-05  2.732e-04   0.253  0.80065
## Chlorides           -8.595e-02  2.945e-02  -2.919  0.00352 **
## FreeSulfurDioxide    1.581e-04  6.203e-05   2.549  0.01081 *
## TotalSulfurDioxide   1.136e-04  4.101e-05   2.770  0.00561 **
## Density             -1.073e+00  2.522e-01  -4.254 2.12e-05 ***
## pH                  -7.518e-03  1.003e-02  -0.749  0.45371
## Sulphates           -3.056e-02  1.084e-02  -2.820  0.00481 **
## Alcohol              1.105e-02  1.882e-03   5.872 4.47e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1.207704)
##
##     Null deviance: 15871  on 8957  degrees of freedom
## Residual deviance: 15685  on 8946  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```
model3 <- glm(TARGET ~ LabelAppeal + STARS + AcidIndex + STARS_imputed, family = quasipoisson, train)
summary(model3)
```

```
##
## Call:
## glm(formula = TARGET ~ LabelAppeal + STARS + AcidIndex + STARS_imputed,
##     family = quasipoisson, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -2.7256  -0.8539   0.0154   0.5614   4.0784
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.336135   0.043935  30.412   <2e-16 ***
## LabelAppeal    0.001062   0.009607   0.111    0.912
## STARS          0.244117   0.006723  36.311   <2e-16 ***
## AcidIndex     -0.073467   0.005248 -14.000   <2e-16 ***
## STARS_imputed -0.802386   0.021243 -37.771   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 0.9603535)
##
##     Null deviance: 15871  on 8957  degrees of freedom
## Residual deviance: 10102  on 8953  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 6
```

Model Selection

```
modstats <- function(model, df, yhat = FALSE){
  y <- data.frame(yhat=c(0:8), TARGET = c(0:8), n=c(0))
  if(yhat){
    df$yhat <- yhat
  } else {
    df$yhat <- round(predict.glm(model, newdata=df, type="response"), 0)
  }
  df <- df %>%
    group_by(yhat, TARGET) %>%
    tally() %>%
    mutate(accuracy = ifelse(yhat > TARGET, "Over", ifelse(yhat < TARGET, "Under", "Accurate"))) %>%
    mutate(cases_sold = ifelse(yhat > TARGET, TARGET, yhat) * n,
           glut = ifelse(yhat > TARGET, yhat - TARGET, 0) * n,
           missed_opportunity = ifelse(yhat < TARGET, TARGET - yhat, 0) * n) %>%
    mutate(net_cases_sold = cases_sold - glut,
           adj_net_cases_sold = cases_sold - glut - missed_opportunity)
  results <- df %>%
    group_by(accuracy) %>%
    summarise(n = sum(n)) %>%
    spread(accuracy, n)
  Ac <- results$Accurate
  over <- results$Over
  under <- results$Under
  cases_sold <- sum(df$cases_sold)
  net_cases_sold <- sum(df$net_cases_sold)
  adj_net_cases_sold <- sum(df$adj_net_cases_sold)
  missed_opportunity <- sum(df$missed_opportunity)
  glut <- sum(df$glut)
  cm <- df %>%
    bind_rows(y) %>%
```

```
    group_by(yhat, TARGET) %>%
    summarise(n = sum(n)) %>%
    spread(TARGET, n, fill = 0)
  return(
    list("confusion_matrix" = cm,
         "results" = results,
         "df" = df,
         "accuracy" = Ac,
         "over" = over,
         "under" = under,
         "cases_sold" = cases_sold,
         "net_cases_sold" = net_cases_sold,
         "adj_net_cases_sold" = adj_net_cases_sold,
         "glut" = glut,
         "missed_opportunity" = missed_opportunity))
}
```

```
modstats(model1, eval)
```

```
## $confusion_matrix
## # A tibble: 9 x 10
## # Groups:   yhat [9]
##    yhat   `0`   `1`   `2`   `3`   `4`   `5`   `6`   `7`   `8`
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     0     0     0     0     0     0     0     0     0     0
## 2     1     0     0     0     0     0     0     0     0     0
## 3     2    14     1     4     7     9     6     1     0     0
## 4     3   824    75   297   774   931   595   217    44    10
## 5     4     2     0     3     2    13     6     1     1     0
## 6     5     0     0     0     0     0     0     0     0     0
## 7     6     0     0     0     0     0     0     0     0     0
## 8     7     0     0     0     0     0     0     0     0     0
## 9     8     0     0     0     0     0     0     0     0     0
##
## $results
## # A tibble: 1 x 3
##   Accurate  Over Under
##      <int> <int> <int>
## 1      791  1218  1828
##
## $df
## # A tibble: 23 x 9
## # Groups:   yhat [3]
##     yhat TARGET     n accuracy cases_sold  glut missed_opportun~ net_cases_sold
##    <dbl>  <int> <int> <chr>         <dbl> <dbl>           <dbl>          <dbl>
##  1     2      0    14 Over              0    28               0            -28
##  2     2      1     1 Over              1     1               0              0
##  3     2      2     4 Accurate          8     0               0              8
##  4     2      3     7 Under            14     0               7             14
##  5     2      4     9 Under            18     0              18             18
##  6     2      5     6 Under            12     0              18             12
##  7     2      6     1 Under             2     0               4              2
##  8     3      0   824 Over              0  2472               0          -2472
```

```
##  9     3      1    75 Over             75   150                0           -75
## 10     3      2   297 Over            594   297                0           297
## # ... with 13 more rows, and 1 more variable: adj_net_cases_sold <dbl>
##
## $accuracy
## [1] 791
##
## $over
## [1] 1218
##
## $under
## [1] 1828
##
## $cases_sold
## [1] 8533
##
## $net_cases_sold
## [1] 5569
##
## $adj_net_cases_sold
## [1] 2513
##
## $glut
## [1] 2964
##
## $missed_opportunity
## [1] 3056
```

```
modstats(model2, eval)
```

```
## $confusion_matrix
## # A tibble: 9 x 10
## # Groups:   yhat [9]
##    yhat   `0`   `1`   `2`   `3`   `4`   `5`   `6`   `7`   `8`
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     0     0     0     0     0     0     0     0     0     0
## 2     1     0     0     0     0     0     0     0     0     0
## 3     2    26     3     7    17    18     5     3     1     0
## 4     3   800    73   291   746   897   580   212    43    10
## 5     4    14     0     6    20    38    22     4     1     0
## 6     5     0     0     0     0     0     0     0     0     0
## 7     6     0     0     0     0     0     0     0     0     0
## 8     7     0     0     0     0     0     0     0     0     0
## 9     8     0     0     0     0     0     0     0     0     0
##
## $results
## # A tibble: 1 x 3
##   Accurate  Over Under
##      <int> <int> <int>
## 1      791  1233  1813
##
## $df
## # A tibble: 24 x 9
## # Groups:   yhat [3]
```

```
##     yhat TARGET     n accuracy cases_sold  glut missed_opportun~ net_cases_sold
##    <dbl>  <int> <int> <chr>         <dbl> <dbl>            <dbl>          <dbl>
##  1     2      0    26 Over              0    52                0            -52
##  2     2      1     3 Over              3     3                0              0
##  3     2      2     7 Accurate         14     0                0             14
##  4     2      3    17 Under            34     0               17             34
##  5     2      4    18 Under            36     0               36             36
##  6     2      5     5 Under            10     0               15             10
##  7     2      6     3 Under             6     0               12              6
##  8     2      7     1 Under             2     0                5              2
##  9     3      0   800 Over              0  2400                0          -2400
## 10     3      1    73 Over             73   146                0            -73
## # ... with 14 more rows, and 1 more variable: adj_net_cases_sold <dbl>
##
## $accuracy
## [1] 791
##
## $over
## [1] 1233
##
## $under
## [1] 1813
##
## $cases_sold
## [1] 8556
##
## $net_cases_sold
## [1] 5576
##
## $adj_net_cases_sold
## [1] 2543
##
## $glut
## [1] 2980
##
## $missed_opportunity
## [1] 3033
```

```
modstats(model3, eval)
```

```
## $confusion_matrix
## # A tibble: 9 x 10
## # Groups:   yhat [9]
##    yhat   `0`   `1`   `2`   `3`   `4`   `5`   `6`   `7`   `8`
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     0     0     0     0     0     0     0     0     0     0
## 2     1   618    42    83   135    81    25     6     2     1
## 3     2    48     2    10    10     8     5     1     0     0
## 4     3   166    30   160   410   385   179    45     1     0
## 5     4     8     2    40   184   330   201    62    10     1
## 6     5     0     0    11    44   123   135    57    14     1
## 7     6     0     0     0     0    25    56    44    15     7
## 8     7     0     0     0     0     1     6     4     3     0
## 9     8     0     0     0     0     0     0     0     0     0
```

```
##
## $results
## # A tibble: 1 x 3
##   Accurate  Over Under
##      <int> <int> <int>
## 1      974  1528  1335
##
## $df
## # A tibble: 49 x 9
## # Groups:   yhat [7]
##     yhat TARGET     n accuracy cases_sold  glut missed_opportun~ net_cases_sold
##    <dbl>  <int> <int> <chr>         <dbl> <dbl>            <dbl>          <dbl>
##  1     1      0   618 Over              0   618                0           -618
##  2     1      1    42 Accurate         42     0                0             42
##  3     1      2    83 Under            83     0               83             83
##  4     1      3   135 Under           135     0              270            135
##  5     1      4    81 Under            81     0              243             81
##  6     1      5    25 Under            25     0              100             25
##  7     1      6     6 Under             6     0               30              6
##  8     1      7     2 Under             2     0               12              2
##  9     1      8     1 Under             1     0                7              1
## 10     2      0    48 Over              0    96                0            -96
## # ... with 39 more rows, and 1 more variable: adj_net_cases_sold <dbl>
##
## $accuracy
## [1] 974
##
## $over
## [1] 1528
##
## $under
## [1] 1335
##
## $cases_sold
## [1] 9441
##
## $net_cases_sold
## [1] 7336
##
## $adj_net_cases_sold
## [1] 5188
##
## $glut
## [1] 2105
##
## $missed_opportunity
## [1] 2148
```

Stuff to delete

Conclusion

When basing the assumption soley on how many cases are sold, it looks like model 3 is best. This model also has the greatest accuracy and was the best estimate of total cases for the business. If choosing a model based only the number of cases sold, this model should take priority.