---

title: "Final Project DDS 6306"

author: "Payden Bullis"

date: "2024-12-15"

output: html\_document

---

initial data view

```{r}

library(tidyverse)

library(e1071)

Winedatatrain <- read.csv("C:\\Users\\payde\\Desktop\\Homework and data sets\\winedata\\Wine Train.csv")

winelocal <- read.csv("C:\\Users\\payde\\Desktop\\Homework and data sets\\winedata\\Wine Types And Locations.csv")

#pairs(Winedatatrain) I ran this but made it a note because it slowed everything down.

```

prep data for review

```{r}

#inner join

Winedatamerged <- inner\_join(Winedatatrain, winelocal, by = "ID")

print(Winedatamerged)

```

```{r}

# Convert "type" to binary (1 for "white", 0 for "red")

Winedatamerged <- Winedatamerged %>% mutate(typenumber = ifelse(type == "white", 1, 0))

Winedatamerged <- na.omit(Winedatamerged)

print(Winedatamerged)

```

```{r}

# Split data frame by 'Type'

bins <- c(-Inf, 6, 8, Inf)

labels <- c("1", "2", "3")

# Create a new column 'quality\_group' with the binned quality values

Winedatamerged$quality\_group <- cut(Winedatamerged$quality, breaks = bins, labels = labels, right = FALSE)

print(Winedatamerged)

df\_white <- Winedatamerged %>% filter(type == "white")

df\_red <- Winedatamerged %>% filter(type == "red")

print(df\_white)

print(df\_red)

df\_red <- subset(df\_red, select = -c(type, location))

df\_white<- subset(df\_white, select = -c(type, location))

```

```{r}

#split by local

# Convert "type" to binary (1 for "Cali", 0 for "Texas")

Winedatamerged2 <- Winedatamerged %>% mutate(localnum = ifelse(location == "California", 1, 0))

#collapse category of quality. into mid high low

#8 and 9 high

#5 6 7 mid

#3 and 4 low

# Define the bins and labels

bins <- c(-Inf, 6, 8, Inf)

labels <- c("1", "2", "3")

# Create a new column 'quality\_group' with the binned quality values

Winedatamerged2$quality\_group <- cut(Winedatamerged2$quality, breaks = bins, labels = labels, right = FALSE)

df\_cali <- Winedatamerged2 %>% filter(location == "California")

df\_texas <- Winedatamerged2 %>% filter(location == "Texas")

# drop cate for comparision

df\_cali <- subset(df\_cali, select = -c(type, location))

df\_texas<- subset(df\_texas, select = -c(type, location))

print(df\_cali)

print(df\_texas)

```

Graphs and data

```{r}

#create scatter charts of just the independant variables on dependant variable.

# List of independent variables

independent\_vars <- c("fixed.acidity", "volatile.acidity", "citric.acid", "residual.sugar",

"chlorides", "free.sulfur.dioxide", "total.sulfur.dioxide", "density",

"pH", "sulphates", "alcohol")

# Loop through each independent variable and create a scatter plot

for (var in independent\_vars) {

p <- ggplot(Winedatamerged2, aes\_string(x = var, y = "quality")) +

geom\_jitter() +

labs(title = paste("Scatter plot of", var, "vs quality"), x = var, y = "quality")

print(p)

}

```

chucking for legibility

```{r}

#red wine and white wine seperated

for (var in independent\_vars) {

W <- ggplot(df\_white, aes\_string(x = var, y = "quality\_group")) +

geom\_jitter() +

labs(title = paste("White Wine Scatter plot of", var, "vs quality"), x = var, y = "quality\_group")

print(W)

}

```

```{r}

for (var in independent\_vars) {

R <- ggplot(df\_red, aes\_string(x = var, y = "quality\_group")) +

geom\_jitter() +

labs(title = paste("Red Wine Scatter plot of", var, "vs quality"), x = var, y = "quality\_group")

print(R)

}

```

Based on the charts and data I'm not convinced that there are any significant transformations to be used but grouping categories to flatten variables was doen with quality broken into three groups this helps solve the issue that in the data set there were only 4 9 quality wines.

based on the data the quality of wine was dependent on location with California wines exhibiting a higher overall quality with an grouping at 6 quality while Texas exhibited a grouping around 5.

Alcohol content shows the most visually strong relation

Log data to better meet assumptions of test

```{r}

# Apply logarithmic transformation to all variables

Winedatamerged3 <- inner\_join(Winedatatrain, winelocal, by = "ID")

print(Winedatamerged3)

# Apply logarithmic transformation to selected variables

log\_wine <- Winedatamerged3 %>%

mutate(across(c("fixed.acidity", "volatile.acidity", "citric.acid", "residual.sugar",

"chlorides", "free.sulfur.dioxide", "total.sulfur.dioxide", "density",

"pH", "sulphates", "alcohol"), log))

#charts

for (var in independent\_vars) {

l <- ggplot(log\_wine, aes\_string(x = var, y = "quality")) +

geom\_jitter() +

labs(title = paste("Scatter plot of", var, "vs quality"), x = var, y = "quality")

print(l)

}

```

Shiny app coding

```{r}

# Load necessary libraries

library(shiny)

library(ggplot2)

# Define UI for the application

ui <- fluidPage(

titlePanel("Wine Quality Analysis"),

sidebarLayout(

sidebarPanel(

selectInput("variable", "Select Variable:",

choices = names(Winedatamerged)[!names(Winedatamerged) %in% c("quality", "location")]),

selectInput("location", "Select Location:",

choices = c("All", unique(Winedatamerged$location)))

),

mainPanel(

plotOutput("qualityPlot")

)

)

)

# Define server logic

server <- function(input, output) {

filteredData <- reactive({

if (input$location == "All") {

Winedatamerged

} else {

subset(Winedatamerged, location == input$location)

}

})

output$qualityPlot <- renderPlot({

ggplot(filteredData(), aes\_string(x = input$variable, y = "quality")) +

geom\_point(color = "blue") +

theme\_minimal() +

labs(title = "Quality vs Selected Variable",

x = input$variable,

y = "Quality")

})

}

# Run the application

shinyApp(ui = ui, server = server)

```

Model building

```{r}

library(MASS)

# Function to perform stepwise regression

perform\_stepwise <- function(data) {

# Define the full model with all predictors

full\_model <- lm(quality ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +

chlorides + free.sulfur.dioxide + total.sulfur.dioxide + density +

pH + sulphates + alcohol + typenumber + localnum, data = data)

# Perform stepwise regression

stepwise\_model <- stepAIC(full\_model, direction = "both", trace = FALSE)

return(stepwise\_model)

}

stepwise\_Wine <- perform\_stepwise(Winedatamerged2)

# Print summaries of the models

summary(stepwise\_Wine)

```

```{r}

#try with log

perform\_stepwise <- function(data) {

# Define the full model with all predictors

full\_model\_log <- lm(quality ~ fixed.acidity + volatile.acidity + residual.sugar +

chlorides + free.sulfur.dioxide + total.sulfur.dioxide + density +

pH + sulphates + alcohol, data = data)

# Perform stepwise regression

stepwise\_model <- stepAIC(full\_model\_log, direction = "both", trace = FALSE)

return(stepwise\_model)

}

stepwise\_Wine\_log <- perform\_stepwise(log\_wine)

# Print summaries of the models

summary(stepwise\_Wine\_log)

```

Based on analysis sulphate free sulpher and total sulpher should be logged.

check variable inflation factor of model for multicollinearity

```{r}

install.packages("car")

library(car)

# Calculate VIF

vif\_values1 <- vif(stepwise\_Wine\_log)

# Print VIF values

print(vif\_values1)

# Calculate VIF

vif\_values2 <- vif(stepwise\_Wine)

# Print VIF values

print(vif\_values2)

```

Prep test and train set to match with same columns and details.

```{r}

#clean data

Winedatatrain <- read.csv("C:\\Users\\payde\\Desktop\\Homework and data sets\\winedata\\Wine Train.csv")

winelocal <- read.csv("C:\\Users\\payde\\Desktop\\Homework and data sets\\winedata\\Wine Types And Locations.csv")

Winedatatest <- read.csv("C:\\Users\\payde\\Desktop\\Homework and data sets\\winedata\\Wine Test Set.csv")

Winedatatest$quality <- 0

# Combine train and test data

combined\_data <- rbind(Winedatatrain, Winedatatest)

#inner join

combined\_data <- inner\_join(combined\_data, winelocal, by = "ID")

print(combined\_data)

# Convert "type" to binary (1 for "white", 0 for "red")

combined\_data <- combined\_data %>% mutate(typenumber = ifelse(type == "white", 1, 0))

combined\_data <- na.omit(combined\_data)

print(combined\_data)

# Convert "type" to binary (1 for "Cali", 0 for "Texas")

combined\_data <- combined\_data %>% mutate(localnum = ifelse(location == "California", 1, 0))

#log variables in final data set. free.sulfur.dioxide total.sulfur.dioxide sulphates

Final\_wine <- combined\_data %>%

mutate(across(c("free.sulfur.dioxide", "total.sulfur.dioxide", "sulphates"), log))

#final model

final\_model <- lm(quality ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +

chlorides + free.sulfur.dioxide + total.sulfur.dioxide +

pH + sulphates + alcohol + typenumber + localnum, data = Final\_wine)

summary(final\_model)

#export

write.csv(Final\_wine, "C:\\Users\\payde\\Desktop\\Homework and data sets\\winedata\\Finalwine.csv", row.names = TRUE)

# Calculate VIF

vif\_values\_final <- vif(final\_model)

# Print VIF values

print(vif\_values\_final)

```

prediction in SAS code below

PROC IMPORT DATAFILE=REFFILE

DBMS=CSV

OUT=finalwine;

GETNAMES=YES;

RUN;

PROC CONTENTS DATA=finalwine; RUN;

%web\_open\_table(finalwine);

proc print data=finalwine;

run;

proc glm data = finalwine;

class typenumber localnum;

model quality = fixedacidity volatileacidity citricacid residualsugar chlorides freesulfurdioxide totalsulfurdioxide pH sulphates alcohol typenumber localnum;

output out = results p = Predict;

run;

data predictfinal;

set results;

if Predict > 0 then quality = round(Predict);

if Predict < 0 then quality = 3;

keep id quality;

where id >5464;

;

proc print data=predictfinal;

run;