```
knitr::opts_chunk$set(echo = TRUE)
library(MASS)
library(car)
library(caret)
library(vcd)
library(ggplot2)
library(rms)
library(Hmisc)
library(kernlab)
library(car)
library(caret)
library(corrplot)
library(data.table)
library(dplyr)
library(geoR)
library(ggplot2)
library(grid)
library(gridExtra)
library(knitr)
library(MASS)
library(naniar)
library(nortest)
library(psych)
library(randomForest)
library(testthat)
library(kknn)
```

```
a <- ggplot(white, aes(x = fixed.acidity))
a + geom_density(aes(fill = factor(quality)), alpha=0.4)
a <- ggplot(white, aes(x = volatile.acidity))
a + geom_density(aes(fill = factor(quality)), alpha=0.4)
a <- ggplot(white, aes(x = citric.acid))
a + geom_density(aes(fill = factor(quality)), alpha=0.4)
a <- ggplot(white, aes(x = residual.sugar))
a + geom_density(aes(fill = factor(quality)), alpha=0.4)
a <- ggplot(white, aes(x = chlorides))
a + geom_density(aes(fill = factor(quality)), alpha=0.4)
a <- ggplot(white, aes(x = free.sulfur.dioxide))
a + geom_density(aes(fill = factor(quality)), alpha=0.4)
a <- ggplot(white, aes(x = total.sulfur.dioxide))
a + geom_density(aes(fill = factor(quality)), alpha=0.4)
a <- ggplot(white, aes(x = density))
a + geom_density(aes(fill = factor(quality)), alpha=0.4)
a <- ggplot(white, aes(x = pH))
a + geom_density(aes(fill = factor(quality)), alpha=0.4)
a <- ggplot(white, aes(x = sulphates))
a + geom_density(aes(fill = factor(quality)), alpha=0.4)
a <- ggplot(white, aes(x = alcohol))
a + geom_density(aes(fill = factor(quality)), alpha=0.4)
ggplot(white, aes(pH, quality)) +
geom_jitter(aes(color = quality), size = 0.5)
ggplot(white, aes(density, quality)) +
geom_jitter(aes(color = quality), size = 0.5)
ggplot(white, aes(alcohol, quality)) +
geom_jitter(aes(color = quality), size = 0.5)
```

```
a <- ggplot(white, aes(x= factor(quality), y  = citric.acid, col=factor(quality)))

a  + geom_jitter(position = position_jitter(0.2)) + geom_violin(trim = FALSE)

white_all <-
data.frame(read.csv("https://raw.githubusercontent.com/jjohn81/DATA621_Final_Project/master/wine
quality-white.csv", sep = ";"))

white_all$quality <- factor(white_all$quality)

levels(white_all$quality)

smp_size <- floor(0.85 * nrow(white))

## set the seed to make your partition reproducible

set.seed(123)

train_ind <- sample(seq_len(nrow(white_all)), size = smp_size)

white <- white_all[train_ind, ]

test <- white_all[-train_ind, ]

table(factor(white$quality))

mod.fit.ord <- polr(formula = quality ~ ., data=white, method= "logistic" )

summary(mod.fit.ord)

Anova(mod.fit.ord)

ctable <- coef(summary(mod.fit.ord))

## calculate and store p values

p <- pnorm(abs(ctable[, "t value"]), lower.tail = FALSE) * 2

ctable <- round(cbind(ctable, "p value" = p),4)

ctable

mod.fit.ord2 <- polr(formula = quality ~ .-citric.acid-chlorides-total.sulfur.dioxide, data=white, method=
"logistic" )

summary(mod.fit.ord2)

Anova(mod.fit.ord2)

cor(white2[,1:8])

plot(residual.sugar, alcohol)

df3<- NULL

a <- NULL
```

```
for (j in 1:length(lvl)){
   a[j] <- mean(white2$fixed.acidity[white2$quality == lvl[j]])}
b <- NULL
for (j in 1:length(lvl)){
   b[j] <- mean(white2$volatile.acidity[white2$quality == lvl[j]])}

c <- NULL
for (j in 1:length(lvl)){
   c[j] <- mean(white2$residual.sugar[white2$quality == lvl[j]])}

d <- NULL
for (j in 1:length(lvl)){
   d[j] <- mean(white2$free.sulfur.dioxide[white2$quality == lvl[j]])}

e <- NULL
for (j in 1:length(lvl)){
   e[j] <- mean(white2$density[white2$quality == lvl[j]])}

f <- NULL
for (j in 1:length(lvl)){
   f[j] <- mean(white2$pH[white2$quality == lvl[j]])}

g <- NULL
for (j in 1:length(lvl)){
   g[j] <- mean(white2$sulphates[white2$quality == lvl[j]])}

h <- NULL
for (j in 1:length(lvl)){
   h[j] <- mean(white2$alcohol[white2$quality == lvl[j]])}
```

```
df3 <- t(data.frame(rbind(a,b,c,d,e,f,g,h)))

colnames(df3) <- names(white2[1:8])

row.names(df3) <- lvl

df3



mod.fit.ord3 <- polr(formula = quality ~ .+fixed.acidity*pH+residual.sugar*alcohol+
residual.sugar*density+ density*alcohol-citric.acid-chlorides-total.sulfur.dioxide, data=white, method=
"logistic" )

summary(mod.fit.ord3)

Anova(mod.fit.ord3)

attach(white)

pred <- predict(object=mod.fit.ord3, type="class")

cmatrix.t <- t(table(quality,pred))

Caret_cmat <- confusionMatrix(cmatrix.t)

Caret_cmat

mosaic(cmatrix.t , shade=TRUE, legend=TRUE)

plot.xmean.ordinaly(mod.fit.ord3, white2)

white3 <- white2

white3$quality[white3$quality == "3"] <- "4"

white3$quality[white3$quality == "9"] <- "8"

table(white3$quality)

white3$quality <- factor(white3$quality)

levels(white3$quality)

mod.fit.ord5 <- polr(formula = quality ~ .-citric.acid-chlorides-total.sulfur.dioxide, data=white3, method=
"logistic" )

summary(mod.fit.ord5)

pred <- predict(object=mod.fit.ord5, type="class")
```

```
cmatrix.t2 <- t(table(white3$quality,pred))

Caret_cmat2 <- confusionMatrix(cmatrix.t2)

Caret_cmat2

white4 <- white3

white4$quality[white4$quality == "4"] <- "5"

white4$quality[white4$quality == "8"] <- "7"

table(white4$quality)

white4$quality <- factor(white4$quality)

levels(white4$quality)

mod.fit.ord6 <- polr(formula = quality ~ .-citric.acid-chlorides-total.sulfur.dioxide, data=white4, method= "logistic" )

summary(mod.fit.ord6)

pred <- predict(object=mod.fit.ord6, type="class")

cmatrix.t3 <- t(table(white4$quality,pred))

Caret_cmat3 <- confusionMatrix(cmatrix.t3)

Caret_cmat3


red <- data.frame(read.csv("https://raw.githubusercontent.com/jjohn81/DATA621_Final_Project/master/winequality-red.csv", sep = ";"))

white <- data.frame(read.csv("https://raw.githubusercontent.com/jjohn81/DATA621_Final_Project/master/winequality-white.csv", sep = ";"))


summary(red)


#quality of whites from 3 to 9 (mean = 5.878)

summary(white)


plot(red$quality)
```

```
#quality of whites from 3 to 9 (mean = 5.878)
plot(white$quality)
red$type <- as.factor("R")
white$type <- as.factor("W")
wines <- rbind(red,white)

# 6497 obs. of  13 variables
str(wines)



# Data Visualizations (to fill in) - boxplots, etc.

# Scatterplot matrix

panel.cor <- function(x, y, digits=2, prefix="", cex.cor, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y))
  txt <- format(c(r, 0.123456789), digits=digits)[1]
  txt <- paste(prefix, txt, sep="")
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}

pairs(~quality + fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
      chlorides + free.sulfur.dioxide + total.sulfur.dioxide + density +pH +
      sulphates + alcohol + type, data = wines,
```

```
    lower.panel=panel.smooth, upper.panel=panel.cor, pch=20, main="Wines Scatterplot Matrix")


corrplot(cor(red[1:12]))

corrplot(cor(white[1:12]))

white <- white[,1:12]

white$quality <- as.factor(white$quality )

set.seed(100)

trainingRows <- sample(1:nrow(white), 0.7 * nrow(white))

trainingData <- white[trainingRows, ]

testData <- white[-trainingRows, ]


fullModel <- polr(quality ~. , data = trainingData, Hess=TRUE)

summary(fullModel)

p <- predict(fullModel, testData)

confusionMatrix(p, testData$quality)

stepModel <- step(m)

p <- predict(stepModel, testData)

confusionMatrix(p, testData$quality)

#loading white wine data from github (raw)

white <-
data.frame(read.csv("https://raw.githubusercontent.com/jjohn81/DATA621_Final_Project/master/wine
quality-white.csv", sep = ";"))


names(white)

#quality of whites from 3 to 9 (mean = 5.878)

summary(white)

vis_miss(white)

panel.cor <- function(x, y, digits=2, prefix="", cex.cor, ...)

{
```

```
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y))
  txt <- format(c(r, 0.123456789), digits=digits)[1]
  txt <- paste(prefix, txt, sep="")
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}

pairs(~quality + fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
       chlorides + free.sulfur.dioxide + total.sulfur.dioxide + density +pH +
       sulphates + alcohol, data = white,
     lower.panel=panel.smooth, upper.panel=panel.cor, pch=20, main="White Wines Scatterplot Matrix")

white$quality <- as.factor(white$quality)
inTrain <- createDataPartition(white$quality, p = 2/3, list = F)
train.white <- white[inTrain,]
test.white <- white[-inTrain,]
t.ctrl <- trainControl(method = "repeatedcv", number = 5, repeats = 5)
kknn.grid <- expand.grid(kmax = c(3, 5, 7 ,9, 11), distance = c(1, 2),
                         kernel = c("rectangular", "gaussian", "cos"))
kknn.train <- train(quality ~ ., data = train.white, method = "kknn",
                    trControl = t.ctrl, tuneGrid = kknn.grid,
                    preProcess = c("center", "scale"))
plot(kknn.train)
kknn.train$bestTune
kknn.predict <- predict(kknn.train, test.white)
confusionMatrix(kknn.predict, test.white$quality)
```