# HW3

Johnny Lydon

2024-02-24

## Homework 3

```
library(readr)
library(tidyr)
library(dplyr)
```

```
##
## 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(purrr)
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:purrr':
##
##     some

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

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 4.1-8
```

## Question 1

```
#1.1

url1 <- "https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv"

url2 <- "https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv"

df1 <- read.csv(url1, sep = ';')
df2 <- read.csv(url2, sep = ';')
```

```
#1.2

df1$type <- 'white'
df2$type <- 'red'
df <- rbind(df1, df2)

names(df) <- gsub(" ", "_", names(df))

df <- df[, !(names(df) %in% c("fixed.acidity", "free.sulfur.dioxide"))]

df$type <- as.factor(df$type)

df <- na.omit(df)

dim(df)
```

```
## [1] 6497   11
```

```
#1.3

mean_red <- mean(df[df$type == 'red', 'quality'])
mean_white <- mean(df[df$type == 'white', 'quality'])
diff_mean <- mean_red - mean_white

num_red <- sum(df$type == 'red')
num_white <- sum(df$type == 'white')
var_red <- var(df[df$type == 'red', 'quality'])
var_white <- var(df[df$type == 'white', 'quality'])

sp_squared <- ((num_red - 1) * var_red + (num_white - 1) * var_white) / (num_red + num_white - 2)

t1 <- diff_mean / sqrt(sp_squared * (1/num_red + 1/num_white))
```

```r
#1.4

t_test <- t.test(quality ~ type, data = df, var.equal = TRUE)
t2 <- t_test$statistic
```

```r
#1.5

linreg <- lm(quality ~ type, data = df)

sumlinreg <- summary(linreg)

t3 <- sumlinreg$coefficients["typewhite", "t value"]
```

```r
#1.6

t_vector <- c(t1, t2, t3)

print(t_vector)
```

```
##                 t
## -9.68565 -9.68565  9.68565
```

```r
#Simply put, the t-values are showing that there's a very big difference in the quality between white a
```

## Question 2

```r
#2.1

library(broom)

model <- lm(quality ~ ., data = df)

model_sum <- tidy(model)

print(model_sum)
```

```
## # A tibble: 11 x 5
##    term                  estimate std.error statistic  p.value
##    <chr>                    <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)           57.5       9.33         6.17  7.44e-10
##  2 volatile.acidity      -1.61      0.0806     -20.0   4.07e-86
##  3 citric.acid            0.0272    0.0783       0.347 7.28e- 1
##  4 residual.sugar         0.0451    0.00416     10.8   3.64e-27
##  5 chlorides             -0.964     0.333       -2.90  3.78e- 3
##  6 total.sulfur.dioxide  -0.000329  0.000262    -1.25  2.10e- 1
##  7 density              -55.2       9.32        -5.92  3.34e- 9
##  8 pH                     0.188     0.0661       2.85  4.38e- 3
##  9 sulphates              0.662     0.0758       8.73  3.21e-18
## 10 alcohol                0.277     0.0142      19.5   1.87e-82
## 11 typewhite             -0.386     0.0549      -7.02  2.39e-12
```

```r
# Basically every indicator of wine quality does in fact have an effect on wine quality. We're not usin
```

```r
#2.2a

model_citric <- lm(quality ~ citric.acid, data = df)

summary(model_citric)
```

```
##
## Call:
## lm(formula = quality ~ citric.acid, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9938 -0.7831  0.1552  0.2426  3.1963
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.65461    0.02602 217.343   <2e-16 ***
## citric.acid  0.51398    0.07429   6.918    5e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8701 on 6495 degrees of freedom
## Multiple R-squared:  0.007316,   Adjusted R-squared:  0.007163
## F-statistic: 47.87 on 1 and 6495 DF,  p-value: 5.002e-12
```

```r
#I'm splitting this question for the summaries.
```

```r
#2.2b

model_sulfur <- lm(quality ~ total.sulfur.dioxide, data = df)

summary(model_sulfur)
```

```
##
## Call:
## lm(formula = quality ~ total.sulfur.dioxide, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8866 -0.7971  0.1658  0.2227  3.1965
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.8923848  0.0246717 238.831  < 2e-16 ***
## total.sulfur.dioxide -0.0006394  0.0001915  -3.338 0.000848 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8726 on 6495 degrees of freedom
## Multiple R-squared:  0.001713,   Adjusted R-squared:  0.001559
## F-statistic: 11.14 on 1 and 6495 DF,  p-value: 0.000848
```

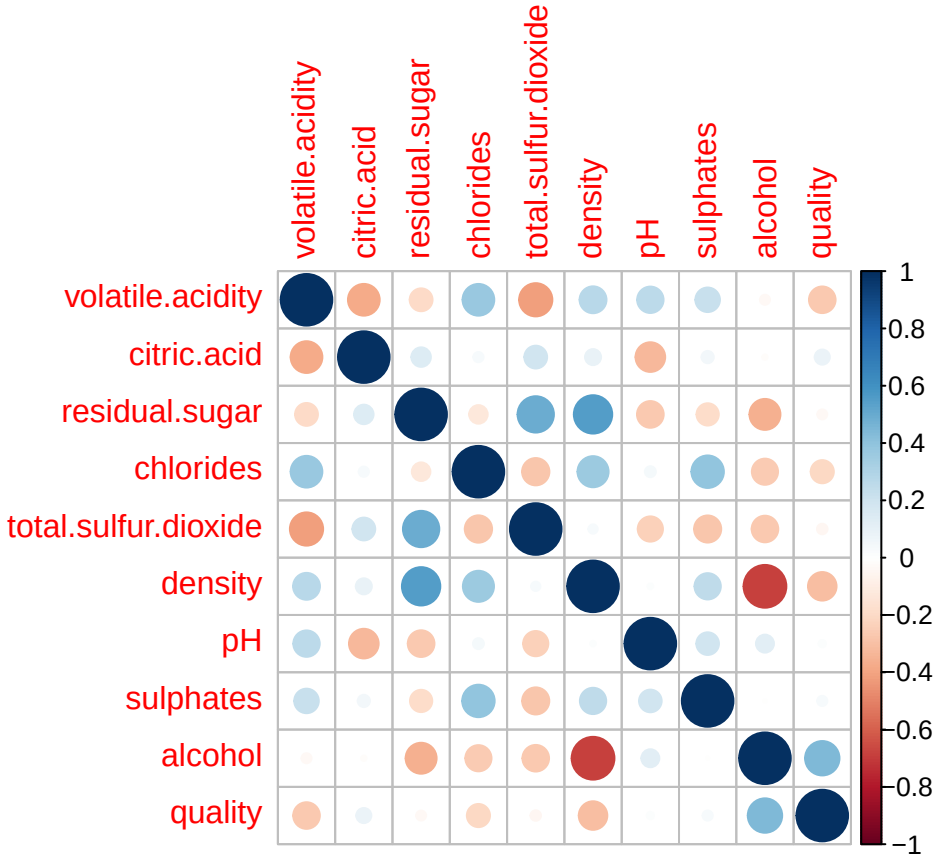
```r
#2.3

library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
wineplot <- cor(df[, sapply(df, is.numeric)])

corrplot(wineplot, method = "circle")
```




```r
#2.4

full_model <- lm(quality ~ ., data = df)

vif_values <- vif(full_model)

print(vif_values)
```

```
##     volatile.acidity          citric.acid       residual.sugar
##             2.103853             1.549248             4.680035
##           chlorides total.sulfur.dioxide              density
##             1.625065             2.628534             9.339357
##                  pH            sulphates              alcohol
##             1.352005             1.522809             3.419849
```

```
##                 type
##             6.694679
```

```
# It looks like the indicators used have pretty low multicollinearities. Of all of them, density has th
```

## Question 3

```
#3.1

backwardreg <- step(full_model, direction = "backward")
```

```
## Start:  AIC=-3953.43
## quality ~ volatile.acidity + citric.acid + residual.sugar + chlorides +
##     total.sulfur.dioxide + density + pH + sulphates + alcohol +
##     type
##
##                        Df Sum of Sq    RSS     AIC
## - citric.acid           1     0.066 3523.6 -3955.3
## - total.sulfur.dioxide  1     0.854 3524.4 -3953.9
## <none>                              3523.5 -3953.4
## - pH                    1     4.413 3527.9 -3947.3
## - chlorides             1     4.559 3528.1 -3947.0
## - density               1    19.054 3542.6 -3920.4
## - type                  1    26.794 3550.3 -3906.2
## - sulphates             1    41.399 3564.9 -3879.5
## - residual.sugar        1    63.881 3587.4 -3838.7
## - alcohol               1   206.860 3730.4 -3584.8
## - volatile.acidity      1   216.549 3740.0 -3567.9
##
## Step:  AIC=-3955.3
## quality ~ volatile.acidity + residual.sugar + chlorides + total.sulfur.dioxide +
##     density + pH + sulphates + alcohol + type
##
##                        Df Sum of Sq    RSS     AIC
## - total.sulfur.dioxide  1     0.818 3524.4 -3955.8
## <none>                              3523.6 -3955.3
## - chlorides             1     4.495 3528.1 -3949.0
## - pH                    1     4.536 3528.1 -3948.9
## - density               1    20.794 3544.4 -3919.1
## - type                  1    26.943 3550.5 -3907.8
## - sulphates             1    41.491 3565.1 -3881.2
## - residual.sugar        1    67.371 3590.9 -3834.3
## - alcohol               1   235.151 3758.7 -3537.6
## - volatile.acidity      1   252.565 3776.1 -3507.5
##
## Step:  AIC=-3955.8
## quality ~ volatile.acidity + residual.sugar + chlorides + density +
##     pH + sulphates + alcohol + type
##
##                    Df Sum of Sq    RSS     AIC
## <none>                          3524.4 -3955.8
```

```
## - pH                1     4.295 3528.7 -3949.9
## - chlorides         1     4.523 3528.9 -3949.5
## - density           1    21.540 3545.9 -3918.2
## - sulphates         1    40.711 3565.1 -3883.2
## - type              1    43.664 3568.0 -3877.8
## - residual.sugar    1    66.572 3591.0 -3836.2
## - alcohol           1   244.545 3768.9 -3521.9
## - volatile.acidity  1   256.695 3781.1 -3501.0
```

```
backward_formula <- formula(backwardreg)

print(backward_formula)
```

```
## quality ~ volatile.acidity + residual.sugar + chlorides + density +
##     pH + sulphates + alcohol + type
```

```
#3.2

library(MASS)
```

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

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

```
null_model <- lm(quality ~ 1, data=df)

forwardreg <- stepAIC(null_model, direction="forward", scope= ~ volatile.acidity + citric.acid + residua
```

```
## Start:  AIC=-1760.04
## quality ~ 1
##
##                       Df Sum of Sq    RSS     AIC
## + alcohol              1    977.95 3975.7 -3186.9
## + density              1    463.41 4490.3 -2396.2
## + volatile.acidity     1    349.71 4604.0 -2233.7
## + chlorides            1    199.47 4754.2 -2025.1
## + type                 1     70.53 4883.2 -1851.2
## + citric.acid          1     36.24 4917.4 -1805.7
## + total.sulfur.dioxide 1      8.48 4945.2 -1769.2
## + sulphates            1      7.34 4946.3 -1767.7
## + residual.sugar       1      6.77 4946.9 -1766.9
## + pH                   1      1.88 4951.8 -1760.5
## <none>                             4953.7 -1760.0
##
## Step:  AIC=-3186.88
## quality ~ alcohol
##
##                       Df Sum of Sq    RSS     AIC
```

```
## + volatile.acidity      1   307.508 3668.2 -3707.9
## + residual.sugar        1    85.662 3890.1 -3326.4
## + type                  1    54.335 3921.4 -3274.3
## + citric.acid           1    40.303 3935.4 -3251.1
## + chlorides             1    39.696 3936.0 -3250.1
## + total.sulfur.dioxide  1    31.346 3944.4 -3236.3
## + sulphates             1     7.859 3967.9 -3197.7
## + pH                    1     5.938 3969.8 -3194.6
## <none>                              3975.7 -3186.9
## + density               1     0.005 3975.7 -3184.9
##
## Step:  AIC=-3707.89
## quality ~ alcohol + volatile.acidity
##
##                         Df Sum of Sq    RSS     AIC
## + sulphates              1    48.259 3620.0 -3791.9
## + density                1    38.704 3629.5 -3774.8
## + residual.sugar         1    29.751 3638.5 -3758.8
## + type                   1    28.895 3639.3 -3757.3
## + total.sulfur.dioxide   1     5.619 3662.6 -3715.9
## + pH                     1     5.533 3662.7 -3715.7
## <none>                               3668.2 -3707.9
## + chlorides              1     0.162 3668.1 -3706.2
## + citric.acid            1     0.099 3668.1 -3706.1
##
## Step:  AIC=-3791.94
## quality ~ alcohol + volatile.acidity + sulphates
##
##                         Df Sum of Sq    RSS     AIC
## + residual.sugar         1    43.989 3576.0 -3869.4
## + density                1    18.661 3601.3 -3823.5
## + type                   1     6.012 3614.0 -3800.7
## + chlorides              1     4.988 3615.0 -3798.9
## + citric.acid            1     2.031 3617.9 -3793.6
## + pH                     1     1.903 3618.1 -3793.4
## <none>                               3620.0 -3791.9
## + total.sulfur.dioxide   1     0.817 3619.2 -3791.4
##
## Step:  AIC=-3869.37
## quality ~ alcohol + volatile.acidity + sulphates + residual.sugar
##
##                         Df Sum of Sq    RSS     AIC
## + type                   1   20.7581 3555.2 -3905.2
## + total.sulfur.dioxide   1   13.3542 3562.6 -3891.7
## + pH                     1    6.6430 3569.3 -3879.5
## + citric.acid            1    4.3384 3571.6 -3875.3
## + chlorides              1    1.8907 3574.1 -3870.8
## <none>                               3576.0 -3869.4
## + density                1    0.0071 3576.0 -3867.4
##
## Step:  AIC=-3905.19
## quality ~ alcohol + volatile.acidity + sulphates + residual.sugar +
##     type
##
```

```
##                         Df Sum of Sq    RSS     AIC
## + density                1   20.4623 3534.8 -3940.7
## + chlorides              1    6.6602 3548.6 -3915.4
## + citric.acid            1    5.2242 3550.0 -3912.7
## + pH                     1    3.9477 3551.3 -3910.4
## + total.sulfur.dioxide   1    1.2539 3554.0 -3905.5
## <none>                               3555.2 -3905.2
##
## Step:  AIC=-3940.7
## quality ~ alcohol + volatile.acidity + sulphates + residual.sugar +
##     type + density
##
##                         Df Sum of Sq    RSS     AIC
## + chlorides              1    6.0826 3528.7 -3949.9
## + pH                     1    5.8541 3528.9 -3949.5
## <none>                               3534.8 -3940.7
## + citric.acid            1    0.8471 3533.9 -3940.3
## + total.sulfur.dioxide   1    0.5646 3534.2 -3939.7
##
## Step:  AIC=-3949.89
## quality ~ alcohol + volatile.acidity + sulphates + residual.sugar +
##     type + density + chlorides
##
##                         Df Sum of Sq    RSS     AIC
## + pH                     1    4.2945 3524.4 -3955.8
## <none>                               3528.7 -3949.9
## + total.sulfur.dioxide   1    0.5765 3528.1 -3948.9
## + citric.acid            1    0.2338 3528.4 -3948.3
##
## Step:  AIC=-3955.8
## quality ~ alcohol + volatile.acidity + sulphates + residual.sugar +
##     type + density + chlorides + pH
##
##                         Df Sum of Sq    RSS     AIC
## <none>                               3524.4 -3955.8
## + total.sulfur.dioxide   1   0.81762 3523.6 -3955.3
## + citric.acid            1   0.02919 3524.4 -3953.9
```

```
forward_formula <- formula(forwardreg)

print(forward_formula)
```

```
## quality ~ alcohol + volatile.acidity + sulphates + residual.sugar +
##     type + density + chlorides + pH
```

```
#3.3a

library(glmnet)

make_model_matrix <- function(formula){
  X <- model.matrix(formula, df)[, -1]
  cnames <- colnames(X)
  for(i in 1:ncol(X)){
```

```
    if(!cnames[i] == "typewhite"){
      X[, i] <- scale(X[, i])
    } else {
      colnames(X)[i] <- "type"}
    }
  return(X)}

y <- df$quality

model_columns <- c('volatile.acidity','citric.acid','residual.sugar','chlorides','total.sulfur.dioxide'
model_formula <- make_formula(model_columns)

X <- make_model_matrix(model_formula)

cv_lasso <- cv.glmnet(X, y, alpha = 1)

cv_ridge <- cv.glmnet(X, y, alpha = 0)
best_lambda_lasso <- cv_lasso$lambda.min
best_lambda_ridge <- cv_ridge$lambda.min

# Not sure what I'm doing wrong, got some help and they couldn't figure out what was happening either.
# Just going to split this one into two as well.
```

```
#3.3b
# Gotta use my stuff from the last part so it doesn't work.
plot(cv_lasso)
title("LASSO Regression (alpha = 1)")

plot(cv_ridge)
title("Ridge Regression (alpha = 0)")

# Can't really say anything about this because it's not working for me.
```

```
#3.4

lasso_coef <- coef(cv_lasso, s = "lambda.1se")
lasso_matrix <- as.matrix(lasso_coef)
lasso_df <- as.data.frame(lasso_matrix)
lasso_vars <- rownames(lasso_df)[lasso_df[, 1] != 0]
lasso_formula <- make_formula(lasso_vars)
print(lasso_vars)
print(lasso_formula)

# Ugh.
```

```
#3.5

ridge_coef <- coef(cv_ridge, s = "lambda.1se")
ridge_matrix <- as.matrix(ridge_coef)
ridge_df <- as.data.frame(ridge_matrix)
ridge_vars <- rownames(ridge_df)[ridge_df[, 1] != 0]
ridge_formula <- make_formula(ridge_vars[-1])
# Print variable names and the formula
```

```r
print(ridge_vars)
print(ridge_formula)
# :(
```

```r
# I wish I could tell you, but something is going wrong for me.
```

## Question 4

```r
#4.1

# 2^10 = 1024
```

```r
#4.2
library(purrr)

all_columns <- colnames(df)
x_vars <- all_columns[all_columns != "quality"]

formulas <- map(
  1:length(x_vars),
  function(k) {
    combn(x_vars, k, function(vars) {
      make_formula(c(vars))}, simplify = FALSE)}
  ) %>%
  unlist()

sample(formulas, 4) %>% as.character()

# I'm confused because I keep being told my code is wrong, but I copied and pasted that code chunk from
```

```r
#4.3

models <- map(formulas, ~lm(.x, data = df))
summaries <- map(models, glance)

tibble_summaries <- bind_rows(summaries)
```

```r
#4.4

max_adj_rsq <- which.max(summaries$adj.r.squared)
rsq_formula <- formulas[max_adj_rsq]
```

```r
#4.5

min_aic <- which.min(tibble_summaries$AIC)

aic_formula <- formulas[min_aic]
```

```r
#4.6

null_formula <- formula(null_model)
full_formula <- formula(full_model)

final_formulas <- c(
  null_formula,
  full_formula,
  backward_formula,
  forward_formula,
  lasso_formula,
  ridge_formula,
  rsq_formula,
  aic_formula
)

# aic_formula and rsq_formula can sometimes be the same, but also may not be the same. The reason is th

# AIC is more reliable since it's reduces the lack of information and makes adjustments to reduce compl

# I'll be honest, I'm not sure. I would think that AIC would struggle to handle such a large number of
```

```r
#4.7

summary_table <- map(
  final_formulas,
  \(x) {
    model <- lm(x, data=df)
    broom::glance(model) %>%
      select(sigma, adj.r.squared, AIC, df, p.value)
    }
) %>% bind_rows()

summary_table %>% knitr::kable()

# Nothing's been working for questions 3 & 4.
```