HW1

Yoonseo Mok

```
library(tidyverse)
library(knitr)
library(kableExtra)
```

Problem 1

A

```
# Importing wine dataset
wine <- read.csv("~/Desktop/Fall 2024/STATS 506/HW1/wine/wine.data",
    header = FALSE)
names(wine) <- c("Class", "Alcohol", "Malic acid", "Ash", "Alcalinity of ash",
    "Magnesium", "Total phenols", "Flavanoids", "Nonflavanoid phenols",
    "Proanthocyanins", "Color intensity", "Hue", "OD280/OD315 of diluted wines",
    "Proline")
```

B

```
# Checking how many wines within each class
table(wine$Class)
```

```
##
##  1  2  3
## 59 71 48
```

Class 1: 59, Class 2:71, Class 3: 48
It is the same number as reported in “wine.names”

C-1

```
# Looking at correlation
cor(wine$Alcohol, wine$`Color intensity`)
```

```
## [1] 0.5463642
```

The correlation is 0.55

C-2

```
# Looking at correlation by each class between alcohol
# content and color intensity
wine %>%
    group_by(Class) %>%
    summarise(cor = cor(Alcohol, `Color intensity`))
```

```
## # A tibble: 3 x 2
##   Class   cor
##   <int> <dbl>
## 1     1 0.408
## 2     2 0.270
## 3     3 0.350
```

Class 1 has the highest correlation and class 2 has the lowest correlation between alcohol content and color intensity

C-3

```
# Alcohol content that has the highest color intensity
wine$Alcohol[which.max(wine$`Color intensity`)]
```

```
## [1] 14.34
```

The alcohol content of the wine with the highest color intensity is 14.34

C-4

```
# Comparing proanthocyanins to ash
tmp = wine[wine$Proanthocyanins > wine$Ash, ]
nrow(tmp)/nrow(wine) * 100
```

```
## [1] 8.426966
```

8.43% of wines had a higher content of proanthocyanins compare to ash

D

```r
tmp2 = matrix(NA, 4 * (ncol(wine) - 1), 2)
colnames(tmp2) = c("Variable", "Average")
for (i in 2:ncol(wine)) {
    tmp2[4 * i - 7, 1] = paste0(names(wine)[i], "-Overall")
    tmp2[4 * i - 6, 1] = paste0(names(wine)[i], "-Class 1")
    tmp2[4 * i - 5, 1] = paste0(names(wine)[i], "-Class 2")
    tmp2[4 * i - 4, 1] = paste0(names(wine)[i], "-Class 3")

    tmp2[4 * i - 7, 2] = round(mean(wine[, i], na.rm = T), 2)
    tmp2[4 * i - 6, 2] = round(mean(wine[wine$Class %in% 1, i],
        na.rm = T), 2)
    tmp2[4 * i - 5, 2] = round(mean(wine[wine$Class %in% 2, i],
        na.rm = T), 2)
    tmp2[4 * i - 4, 2] = round(mean(wine[wine$Class %in% 3, i],
        na.rm = T), 2)
}

kable(tmp2) %>%
    kable_styling("striped", full_width = F)
```

Variable	Average
Alcohol-Overall	13
Alcohol-Class 1	13.74
Alcohol-Class 2	12.28
Alcohol-Class 3	13.15
Malic acid-Overall	2.34
Malic acid-Class 1	2.01
Malic acid-Class 2	1.93
Malic acid-Class 3	3.33
Ash-Overall	2.37
Ash-Class 1	2.46
Ash-Class 2	2.24
Ash-Class 3	2.44
Alcalinity of ash-Overall	19.49
Alcalinity of ash-Class 1	17.04
Alcalinity of ash-Class 2	20.24
Alcalinity of ash-Class 3	21.42
Magnesium-Overall	99.74
Magnesium-Class 1	106.34
Magnesium-Class 2	94.55
Magnesium-Class 3	99.31
Total phenols-Overall	2.3
Total phenols-Class 1	2.84
Total phenols-Class 2	2.26
Total phenols-Class 3	1.68
Flavanoids-Overall	2.03
Flavanoids-Class 1	2.98
Flavanoids-Class 2	2.08
Flavanoids-Class 3	0.78
Nonflavanoid phenols-Overall	0.36

Nonflavanoid phenols-Class 1	0.29
Nonflavanoid phenols-Class 2	0.36
Nonflavanoid phenols-Class 3	0.45
Proanthocyanins-Overall	1.59
Proanthocyanins-Class 1	1.9
Proanthocyanins-Class 2	1.63
Proanthocyanins-Class 3	1.15
Color intensity-Overall	5.06
Color intensity-Class 1	5.53
Color intensity-Class 2	3.09
Color intensity-Class 3	7.4
Hue-Overall	0.96
Hue-Class 1	1.06
Hue-Class 2	1.06
Hue-Class 3	0.68
OD280/OD315 of diluted wines-Overall	2.61
OD280/OD315 of diluted wines-Class 1	3.16
OD280/OD315 of diluted wines-Class 2	2.79
OD280/OD315 of diluted wines-Class 3	1.68
Proline-Overall	746.89
Proline-Class 1	1115.71
Proline-Class 2	519.51
Proline-Class 3	629.9

E

```r
# Class 1 vs Class 2
class12 = subset(wine, wine$Class %in% c(1, 2))
t.test(class12$`Total phenols` ~ class12$Class)
```

```
##
##  Welch Two Sample t-test
##
## data:  class12$`Total phenols` by class12$Class
## t = 7.4206, df = 119.14, p-value = 1.889e-11
## alternative hypothesis: true difference in means between group 1 and group 2 is not equal to 0
## 95 percent confidence interval:
##  0.4261870 0.7364055
## sample estimates:
## mean in group 1 mean in group 2
##        2.840169        2.258873
```

```r
# Class 1 vs Class 3
class13 = subset(wine, wine$Class %in% c(1, 3))
t.test(class13$`Total phenols` ~ class13$Class)
```

```
##
##  Welch Two Sample t-test
```

```
##
## data:  class13$`Total phenols` by class13$Class
## t = 17.12, df = 98.356, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group 1 and group 3 is not equal to 0
## 95 percent confidence interval:
##  1.026801 1.296038
## sample estimates:
## mean in group 1 mean in group 3
##        2.840169        1.678750
```

```r
# Class 2 vs Class 3
class23 = subset(wine, wine$Class %in% c(2, 3))
t.test(class23$`Total phenols` ~ class23$Class)
```

```
##
##  Welch Two Sample t-test
##
## data:  class23$`Total phenols` by class23$Class
## t = 7.0125, df = 116.91, p-value = 1.622e-10
## alternative hypothesis: true difference in means between group 2 and group 3 is not equal to 0
## 95 percent confidence interval:
##  0.4162855 0.7439610
## sample estimates:
## mean in group 2 mean in group 3
##        2.258873        1.678750
```

The p-values of all three t-test are significant and show that the level of phenols differs across the three classes. The mean of phenols in group 1 vs group 2 vs group 3 are significantly different.

Problem 2

A

```r
# Importing dataset
AskAManager <- read.csv("~/Desktop/Fall 2024/STATS 506/HW1/AskAManager.csv")
```

B

```r
# Cleaning column name
names(AskAManager) = c("ID", "Timestamp", "Age", "Industry",
    "Jobtitle", "Jobtitle_add", "Salary", "Salary_add", "Currency",
    "Currency_Other", "Context", "Country", "State", "City",
    "Num years_overall", "Num years_field", "Education", "Gender",
    "Race")
```

C

```
# Only USD
USDonly = AskAManager[AskAManager$Currency %in% "USD", ]
nrow(AskAManager)
```

```
## [1] 28062
```

```
nrow(USDonly)
```

```
## [1] 23374
```

Comparing the number of observations, before it was 20862 and after restricting we now have 23374

D

```
# No one starts working before age 18
no18 = USDonly[!(USDonly$Age %in% "under 18"), ]

# Taking the midpoint of age
no18$Age_rev = NA
no18$Age_rev[no18$Age %in% "18-24"] = 21
no18$Age_rev[no18$Age %in% "25-34"] = 30
no18$Age_rev[no18$Age %in% "35-44"] = 40
no18$Age_rev[no18$Age %in% "45-54"] = 50
no18$Age_rev[no18$Age %in% "55-64"] = 60
no18$Age_rev[no18$Age %in% "65 or over"] = 65

# Taking the midpoint of overall experience of working
no18$`Num years_overall_rev` = NA
no18$`Num years_overall_rev`[no18$`Num years_overall` %in% "1 year or less"] = 0.5
no18$`Num years_overall_rev`[no18$`Num years_overall` %in% "2 - 4 years"] = 3
no18$`Num years_overall_rev`[no18$`Num years_overall` %in% "5-7 years"] = 6
no18$`Num years_overall_rev`[no18$`Num years_overall` %in% "8 - 10 years"] = 9
no18$`Num years_overall_rev`[no18$`Num years_overall` %in% "11 - 20 years"] = 15
no18$`Num years_overall_rev`[no18$`Num years_overall` %in% "21 - 30 years"] = 25
no18$`Num years_overall_rev`[no18$`Num years_overall` %in% "31 - 40 years"] = 35
no18$`Num years_overall_rev`[no18$`Num years_overall` %in% "41 years or more"] = 45

# Taking the midpoint of field experience of working
no18$`Num years_field_rev` = NA
no18$`Num years_field_rev`[no18$`Num years_field` %in% "1 year or less"] = 0.5
no18$`Num years_field_rev`[no18$`Num years_field` %in% "2 - 4 years"] = 3
no18$`Num years_field_rev`[no18$`Num years_field` %in% "5-7 years"] = 6
no18$`Num years_field_rev`[no18$`Num years_field` %in% "8 - 10 years"] = 9
no18$`Num years_field_rev`[no18$`Num years_field` %in% "11 - 20 years"] = 15
no18$`Num years_field_rev`[no18$`Num years_field` %in% "21 - 30 years"] = 25
no18$`Num years_field_rev`[no18$`Num years_field` %in% "31 - 40 years"] = 35
```

```r
no18$`Num years_field_rev`[no18$`Num years_field` %in% "41 years or more"] = 45

# Now subtract age from overall experience of working
no18$startwork = NA
no18$startwork = no18$Age_rev - no18$`Num years_overall_rev`

# Exclude those who have negative start work value
no18 = no18[no18$startwork >= 0, ]

# Now subtract age from field experience of working
no18$startwork_field = NA
no18$startwork_field = no18$Age_rev - no18$`Num years_field_rev`

# Exclude those who have negative start work in field value
no18 = no18[no18$startwork_field >= 0, ]

nrow(no18)
```

```
## [1] 23340
```

Eliminating any rows for which their age, years of experience in their field, and years of experience total are impossible, I have 23340 rows.

E

```r
# First adding annual salary and bonus
no18$finalsalary = rowSums(no18[, c("Salary", "Salary_add")],
    na.rm = TRUE)

# Find the outlier value based on IQR
low = mean(no18$finalsalary) - 1.5 * IQR(no18$finalsalary)
up = mean(no18$finalsalary) + 1.5 * IQR(no18$finalsalary)

# Eliminating any outlier(extreme) income
incomeokay = no18[no18$finalsalary <= up & no18$finalsalary >
    low, ]
nrow(incomeokay)
```

```
## [1] 21643
```

I considered the extreme value of income using 1.5*IQR. Any income less than $13000 and higher than 199010 is considered as outlier based on our data. After excluding those people I have 21643 number of observation in my data

Problem 3

A

```r
#' Identify if it is a palindrome and shows the reverse
#' @param x A number.
#' @returns A list if it is palindromic and the reverse of x
#' @examples isPalindromic(199)

isPalindromic = function(x) {
    x = as.character(x)

    if (x < 0) {
        stop("It is a negative integer")
    }

    if (substr(x, nchar(x), nchar(x)) == 0) {
        stop("The integer ends with 0")
    }

    reversed = stringi::stri_reverse(x)

    if (nchar(x)%%2 == 0) {
        # If x is even number
        i = nchar(x)

        firsthalfchar = substr(x, 1, i/2)
        lasthalfchar = substr(x, (i/2) + 1, i)

        if (stringi::stri_reverse(firsthalfchar) == lasthalfchar) {
            isPalindromic = TRUE
        } else {
            isPalindromic = FALSE
        }

    } else {
        # If x is odd number
        i = nchar(x)

        firsthalfchar = substr(x, 1, floor(i/2))
        lasthalfchar = substr(x, ceiling(i/2) + 1, i)

        if (stringi::stri_reverse(firsthalfchar) == lasthalfchar) {
            isPalindromic = TRUE
        } else {
            isPalindromic = FALSE
        }
    }
    return(as.list(data.frame(isPalindromic, reversed)))
}

isPalindromic(728827)
```

```
## $isPalindromic
## [1] TRUE
##
## $reversed
## [1] "728827"
```

```
isPalindromic(39951)
```

```
## $isPalindromic
## [1] FALSE
##
## $reversed
## [1] "15993"
```

B

```
#' Finds the next palindromic number strictly greater than the input
#' @param x A number.
#' @returns A numeric vector
#' @examples nextPalindrome(199)

nextPalindrome = function(x) {
    x = as.character(x)

    if (x < 0) {
        stop("It is a negative integer")
    }

    if (nchar(x)%%2 == 0) {
        # If x is even number
        i = nchar(x)

        firsthalfchar = substr(x, 1, i/2)

        if (as.numeric(paste0(firsthalfchar, stringi::stri_reverse(firsthalfchar))) >
            x) {
            # If the first half numbers and the reverse of
            # that is greater than x then,
            nextvalue = as.numeric(paste0(firsthalfchar, stringi::stri_reverse(firsthalfchar)))
        } else {
            # Else, take the number that is right after the
            # half (first number of the second half) and
            # replace the last number in the first half
            # then, reverse that
            takelastvalue = substr(x, (i/2) + 1, (i/2) + 1)
            takelastvalueadd = paste0(substr(firsthalfchar, 1,
                (i/2) - 1), takelastvalue)
            nextvalue = as.numeric(paste0(takelastvalueadd, stringi::stri_reverse(takelastvalueadd)))
        }

    } else {
```

```
        # if x is odd number
        i = nchar(x)

        firsthalfchar = substr(x, 1, floor(i/2))
        if (as.numeric(paste0(firsthalfchar, substr(x, floor(i/2) +
            1, floor(i/2) + 1), stringi::stri_reverse(firsthalfchar))) >
            x) {
            # If the first half numbers with the middle
            # number and the reverse of that is greater
            # than x then,
            nextvalue = as.numeric(paste0(firsthalfchar, substr(x,
                floor(i/2) + 1, floor(i/2) + 1), stringi::stri_reverse(firsthalfchar)))
        } else {
            # similar logic with even number +1 with the
            # middle number instead we also need to take
            # account when the middle number is 9 because
            # 9+1=10
            takelastvalue = substr(x, floor(i/2) + 1, floor(i/2) +
                1)
            if (takelastvalue < 9) {
                takelastvalueadd = paste0(substr(firsthalfchar,
                  1, floor(i/2) + 1), (as.numeric(takelastvalue) +
                  1))
                nextvalue = as.numeric(paste0(takelastvalueadd,
                  stringi::stri_reverse(firsthalfchar)))
            } else {
                takelastvalueadd = paste0(as.numeric(substr(x,
                  1, ceiling(i/2))) + 1)
                nextvalue = as.numeric(paste0(takelastvalueadd,
                  stringi::stri_reverse(takelastvalueadd)))
            }
        }

    }
    return(nextvalue)
}

nextPalindrome(391)
```

```
## [1] 393
```

```
nextPalindrome(9928)
```

```
## [1] 9999
```

```
nextPalindrome(19272719)
```

```
## [1] 19277291
```

```
nextPalindrome(109)
```

```
## [1] 111
```

```r
nextPalindrome(2)
```

```
## [1] 3
```