

# STATS 506 HW 1

ZHAO Shengchun

**Github URL:**

<https://github.com/ZHAOShengchun67734538/STAT-506-HW-1>

## Question 1

(a)

```
data = read.table("C:/Users/z1883/Desktop/wine/wine.data", sep = ",",
                  ,header = FALSE)

wine.data = data.frame(Type=data[,1],Alcohol=data[,2],Malic.acid=data[,3],
                        Ash=data[,4],Alcalinity.of.ash=data[,5],
                        Magnesium=data[,6],Total.phenols=data[,7],
                        Flavanoids=data[,8],Nonflavanoid.phenols=data[,9],
                        Proanthocyanins=data[,10],Color.intensity=data[,11],
                        Hue=data[,12],OD280.or.OD315=data[,13],
                        Proline=data[,14])

head(wine.data)
```

	Type	Alcohol	Malic.acid	Ash	Alcalinity.of.ash	Magnesium	Total.phenols
1	1	14.23	1.71	2.43	15.6	127	2.80
2	1	13.20	1.78	2.14	11.2	100	2.65
3	1	13.16	2.36	2.67	18.6	101	2.80
4	1	14.37	1.95	2.50	16.8	113	3.85
5	1	13.24	2.59	2.87	21.0	118	2.80
6	1	14.20	1.76	2.45	15.2	112	3.27

	Flavanoids	Nonflavanoid.phenols	Proanthocyanins	Color.intensity	Hue
1	3.06		0.28	2.29	5.64 1.04
2	2.76		0.26	1.28	4.38 1.05

3	3.24	0.30	2.81	5.68	1.03
4	3.49	0.24	2.18	7.80	0.86
5	2.69	0.39	1.82	4.32	1.04
6	3.39	0.34	1.97	6.75	1.05

	OD280.or.OD315	Proline
1	3.92	1065
2	3.40	1050
3	3.17	1185
4	3.45	1480
5	2.93	735
6	2.85	1450

```
nrow(wine.data);ncol(wine.data)
```

```
[1] 178
```

```
[1] 14
```

(b)

```
table(wine.data$Type)
```

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

From the result, we can find that class 1 has 59 instances, class 2 has 71 instances, and class 3 has 48 instances. The output result is the same as the outcome reported in the “wine.names” file.

(c)

(1)

```
cor(wine.data$Alcohol, wine.data$Color.intensity)
```

```
[1] 0.5463642
```

(2)

```
wine.class1 = wine.data[which(wine.data[,1] == 1),]
wine.class2 = wine.data[which(wine.data[,1] == 2),]
wine.class3 = wine.data[which(wine.data[,1] == 3),]
cor1 = cor(wine.class1$Alcohol, wine.class1$Color.intensity)
cor2 = cor(wine.class2$Alcohol, wine.class2$Color.intensity)
cor3 = cor(wine.class3$Alcohol, wine.class3$Color.intensity)
cor1;cor2;cor3
```

```
[1] 0.4082913
```

```
[1] 0.2697891
```

```
[1] 0.3503777
```

From the result, we can find that in three class, the alcohol and color intensity have positive correlation, and the class 1 has the highest correlation and class 2 has the lowest correlation.

(3)

```
wine.data[which(wine.data[,11] == max(wine.data[,11])),2]
```

```
[1] 14.34
```

The 11-th column is the Color Intensity. From the result, we can find the alcohol content of wine with the highest color intensity is 14.34.

(4)

```
rownumber = nrow(wine.data[which(wine.data[,10]>wine.data[,4]),])
(rownumber/nrow(wine.data))*100
```

```
[1] 8.426966
```

The 10-th column is Proanthocyanins and the 4-th column is Ash. From the result, we can find it has 8.426966% of wines had a higher content of Proanthocyanins compare to Ash.

(d)

```

overall = apply(wine.data[,2:14],2,mean)
c1 = apply(wine.data[which(wine.data[,1] == 1),2:14],2,mean)
c2 = apply(wine.data[which(wine.data[,1] == 2),2:14],2,mean)
c3 = apply(wine.data[which(wine.data[,1] == 3),2:14],2,mean)
rbind(overall, c1, c2, c3)

```

	Alcohol	Malic.acid	Ash	Alcalinity.of.ash	Magnesium	Total.phenols
overall	13.00062	2.336348	2.366517	19.49494	99.74157	2.295112
c1	13.74475	2.010678	2.455593	17.03729	106.33898	2.840169
c2	12.27873	1.932676	2.244789	20.23803	94.54930	2.258873
c3	13.15375	3.333750	2.437083	21.41667	99.31250	1.678750

	Flavanoids	Nonflavanoid.phenols	Proanthocyanins	Color.intensity
overall	2.0292697	0.3618539	1.590899	5.058090
c1	2.9823729	0.2900000	1.899322	5.528305
c2	2.0808451	0.3636620	1.630282	3.086620
c3	0.7814583	0.4475000	1.153542	7.396250

	Hue	OD280.or.OD315	Proline
overall	0.9574494	2.611685	746.8933
c1	1.0620339	3.157797	1115.7119
c2	1.0562817	2.785352	519.5070
c3	0.6827083	1.683542	629.8958

(e)

```

# The 7-th column is phenols
class1 = wine.data[which(wine.data[,1] == 1),7]
class2 = wine.data[which(wine.data[,1] == 2),7]
class3 = wine.data[which(wine.data[,1] == 3),7]
n1 = length(class1)
n2 = length(class2)
n3 = length(class3)
mean1 = mean(class1)
mean2 = mean(class2)
mean3 = mean(class3)
var1 = var(class1)
var2 = var(class2)
var3 = var(class3)

# In the t-test, we use two sample t-test
# Assumen they have equal variance, set alpha = 0.05

```

```

# For class 1 and class 2
pooled.var12 = (((n1 - 1)*var1) + ((n2 - 1)*var2))/(n1 + n2 - 2)
t.stat12 = (mean1 - mean2)/sqrt(pooled.var12*(1/n1 + 1/n2))
pvalue12 = 2*pt(-abs(t.stat12), (n1+n2-2))

# For class 2 and class 3
pooled.var23 = (((n2 - 1)*var2) + ((n3 - 1)*var3))/(n2 + n3 - 2)
t.stat23 = (mean2 - mean3)/sqrt(pooled.var23*(1/n2 + 1/n3))
pvalue23 = 2*pt(-abs(t.stat23), (n2+n3-2))

# For class 1 and class 3
pooled.var13 = (((n1 - 1)*var1) + ((n3 - 1)*var3))/(n1 + n3 - 2)
t.stat13 = (mean1 - mean3)/sqrt(pooled.var13*(1/n1 + 1/n3))
pvalue13 = 2*pt(-abs(t.stat13), (n1+n3-2))

# We also need to adjust the p-value for comparison
pvalues = c(pvalue12, pvalue23, pvalue13)
pvalues

```

```
[1] 6.865291e-11 2.203376e-09 2.437614e-32
```

```

adjust.pvalues = p.adjust(pvalues, method = "holm")
adjust.pvalues

```

```
[1] 1.373058e-10 2.203376e-09 7.312843e-32
```

Class 1 & Class 2: After Holm correction, the adjusted p-value is 1.373058e-10, which is lower than 0.05, there is a statistically significant.

Class 2 & Class 3: After Holm correction, the adjusted p-value is 2.203376e-09, which is lower than 0.05, there is a statistically significant.

Class 1 & Class 3: After Holm correction, the adjusted p-value is 7.312843e-32, which is lower than 0.05, there is a statistically significant.

Summary: The two-sample t-tests show that there are statistically significant differences in phenol levels between any of the class pairs.

## Question 2

(a)

```
# After download the data to my own computer,
# import the data from my own computer
data = read.csv("C:/Users/z1883/Desktop/AskAManager.csv",header = TRUE)
head(data)
```

```
X          Timestamp How.old.are.you. What.industry.do.you.work.in.
1 1 4/27/2021 11:02:10          25-34 Education (Higher Education)
2 2 4/27/2021 11:02:22          25-34      Computing or Tech
3 3 4/27/2021 11:02:38          25-34 Accounting, Banking & Finance
4 4 4/27/2021 11:02:41          25-34      Nonprofits
5 5 4/27/2021 11:02:42          25-34 Accounting, Banking & Finance
6 6 4/27/2021 11:02:46          25-34 Education (Higher Education)
```

```
Job.title
1      Research and Instruction Librarian
2 Change & Internal Communications Manager
3      Marketing Specialist
4      Program Manager
5      Accounting Manager
6      Scholarly Publishing Librarian
```

If.your.job.title.needs.additional.context..please.clarify.here.

```
1
2
3
4
5
6
```

What.is.your.annual.salary...You.ll.indicate.the.currency.in.a.later.question..If.you.are..

```
1
2
3
4
5
6
```

How.much.additional.monetary.compensation.do.you.get..if.any..for.example..bonuses.or.over

```
1
2
3
4
5
6
```

Please.indicate.the.currency If..Other...please.indicate.the.currency.here..

```
1          USD
```

2	GBP
3	USD
4	USD
5	USD
6	USD

If.your.income.needs.additional.context..please.provide.it.here.

1  
2  
3  
4  
5  
6

What.country.do.you.work.in.

1	United States
2	United Kingdom
3	US
4	USA
5	US
6	USA

If.you.re.in.the.U.S...what.state.do.you.work.in. What.city.do.you.work.in.

1	Massachusetts	Boston
2		Cambridge
3	Tennessee	Chattanooga
4	Wisconsin	Milwaukee
5	South Carolina	Greenville
6	New Hampshire	Hanover

How.many.years.of.professional.work.experience.do.you.have.overall.

1	5-7 years
2	8 - 10 years
3	2 - 4 years
4	8 - 10 years
5	8 - 10 years
6	8 - 10 years

How.many.years.of.professional.work.experience.do.you.have.in.your.field.

1	5-7 years
2	5-7 years
3	2 - 4 years
4	5-7 years
5	5-7 years
6	2 - 4 years

What.is.your.highest.level.of.education.completed. What.is.your.gender.

1	Master's degree	Woman
2	College degree	Non-binary

```

3           College degree           Woman
4           College degree           Woman
5           College degree           Woman
6           Master's degree          Man
What.is.your.race...Choose.all.that.apply..
1           White
2           White
3           White
4           White
5           White
6           White

```

(b)

```

colnames(data) = c("No.,""Timestamp","Age","Industry","Job Title",
  "Job Title Supplement","Salary","Monetary Compensation",
  "Currency of Salary","Other Currency",
  "Additional Context of Income",
  "Working Country","Working State(if in the US)",
  "Working City","Overall Working Years",
  "Working Years in your field",
  "Highest education level","Gender","Race(indicate all)")

head(data)

```

```

No.      Timestamp  Age      Industry
1  1 4/27/2021 11:02:10 25-34 Education (Higher Education)
2  2 4/27/2021 11:02:22 25-34 Computing or Tech
3  3 4/27/2021 11:02:38 25-34 Accounting, Banking & Finance
4  4 4/27/2021 11:02:41 25-34 Nonprofits
5  5 4/27/2021 11:02:42 25-34 Accounting, Banking & Finance
6  6 4/27/2021 11:02:46 25-34 Education (Higher Education)
      Job Title Job Title Supplement Salary
1  Research and Instruction Librarian 55000
2 Change & Internal Communications Manager 54600
3           Marketing Specialist 34000
4           Program Manager 62000
5           Accounting Manager 60000
6  Scholarly Publishing Librarian 62000
Monetary Compensation Currency of Salary Other Currency
1           0 USD
2        4000 GBP
3           NA USD

```



4	3000	USD	
5	7000	USD	
6	NA	USD	
Additional Context of Income Working Country Working State(if in the US)			
1		United States	Massachusetts
2		United Kingdom	
3		US	Tennessee
4		USA	Wisconsin
5		US	South Carolina
6		USA	New Hampshire
Working City Overall Working Years Working Years in your field			
1	Boston	5-7 years	5-7 years
2	Cambridge	8 - 10 years	5-7 years
3	Chattanooga	2 - 4 years	2 - 4 years
4	Milwaukee	8 - 10 years	5-7 years
5	Greenville	8 - 10 years	5-7 years
6	Hanover	8 - 10 years	2 - 4 years
Highest education level Gender Race(indicate all)			
1	Master's degree	Woman	White
2	College degree	Non-binary	White
3	College degree	Woman	White
4	College degree	Woman	White
5	College degree	Woman	White
6	Master's degree	Man	White

(c)

```
us.data = data[which(data[,9] == "USD"),]
head(us.data)
```

No.	Timestamp	Age	Industry
1	1 4/27/2021 11:02:10	25-34	Education (Higher Education)
3	3 4/27/2021 11:02:38	25-34	Accounting, Banking & Finance
4	4 4/27/2021 11:02:41	25-34	Nonprofits
5	5 4/27/2021 11:02:42	25-34	Accounting, Banking & Finance
6	6 4/27/2021 11:02:46	25-34	Education (Higher Education)
7	7 4/27/2021 11:02:51	25-34	Publishing
	Job Title	Job Title	Supplement Salary
1	Research and Instruction Librarian		55000
3	Marketing Specialist		34000
4	Program Manager		62000
5	Accounting Manager		60000

6	Scholarly Publishing Librarian	62000
7	Publishing Assistant	33000

	Monetary Compensation	Currency of Salary	Other Currency
1	0	USD	
3	NA	USD	
4	3000	USD	
5	7000	USD	
6	NA	USD	
7	2000	USD	

Additional Context of Income Working Country Working State(if in the US)		
1	United States	Massachusetts
3	US	Tennessee
4	USA	Wisconsin
5	US	South Carolina
6	USA	New Hampshire
7	USA	South Carolina

Working City Overall Working Years Working Years in your field		
1	Boston	5-7 years
3	Chattanooga	2 - 4 years
4	Milwaukee	8 - 10 years
5	Greenville	8 - 10 years
6	Hanover	8 - 10 years
7	Columbia	2 - 4 years

Highest education level Gender Race(indicate all)		
1	Master's degree	Woman
3	College degree	Woman
4	College degree	Woman
5	College degree	Woman
6	Master's degree	Man
7	College degree	Woman

```
# the number of rows before restrict
nrow(data)
```

```
[1] 28062
```

```
# the number of rows after restrict
nrow(us.data)
```

```
[1] 23374
```

We can see from the result, after restrict, there is 23374 data, clearly, it worked.

(d)

There are three types of mistakes that we need to delete.

- 1, The age is under 18, which is contradictory with our assumption.
- 2, The overall-working-years is smaller than working-years-in-your-field. because it is nonsense, we need to delete the working-years-in-your-field is larger than overall-working-years.
- 3, Since in “Age” and “Overall Working Years” column is a range, if the max age minus 18 is less than the min working-overall-years, which means they must start to work under 18, which is contradictory with our assumption, so, we also should delete these rows.

```
# solve the mistake 1
unique(us.data$Age)
```

```
[1] "25-34"      "45-54"      "35-44"      "18-24"      "65 or over"
[6] "55-64"      "under 18"
```

```
us.data1 = us.data[-which(us.data[,3] == "under 18"),]
unique(us.data1$Age)
```

```
[1] "25-34"      "45-54"      "35-44"      "18-24"      "65 or over"
[6] "55-64"
```

```
# solve the mistake 2
# First we need to check how many categories they have
unique(us.data1$`Overall Working Years`)
```

```
[1] "5-7 years"      "2 - 4 years"      "8 - 10 years"      "21 - 30 years"
[5] "11 - 20 years"  "41 years or more" "31 - 40 years"      "1 year or less"
```

```
unique(us.data1$`Working Years in your field`)
```

```
[1] "5-7 years"      "2 - 4 years"      "21 - 30 years"      "11 - 20 years"
[5] "8 - 10 years"   "1 year or less"   "31 - 40 years"      "41 years or more"
```

From the result, we can find “overall working years” and “Working Years in your field” have the same categories, which are: 1 year or less, 2-4 years, 5-7 years, 8-10 years, 11-20 years, 21-30 years, 31-40 years, 41 years or more. So, we could use integer number 1, 2, 3,..., 8 to represent each category respectively, since it is easy to compare.

```
# do transformation to "overall working years"
n1 = nrow(us.data1)
overall.label = c(1:n1)*0
for(i in 1:n1)
{
  if(us.data1[i,15] == "1 year or less"){overall.label[i] = 1}
  if(us.data1[i,15] == "2 - 4 years"){overall.label[i] = 2}
  if(us.data1[i,15] == "5-7 years"){overall.label[i] = 3}
  if(us.data1[i,15] == "8 - 10 years"){overall.label[i] = 4}
  if(us.data1[i,15] == "11 - 20 years"){overall.label[i] = 5}
  if(us.data1[i,15] == "21 - 30 years"){overall.label[i] = 6}
  if(us.data1[i,15] == "31 - 40 years"){overall.label[i] = 7}
  if(us.data1[i,15] == "41 years or more"){overall.label[i] = 8}
}
# do the same transformation to "Working Years in your field"
field.label = c(1:n1)*0
for(i in 1:n1)
{
  if(us.data1[i,16] == "1 year or less"){field.label[i] = 1}
  if(us.data1[i,16] == "2 - 4 years"){field.label[i] = 2}
  if(us.data1[i,16] == "5-7 years"){field.label[i] = 3}
  if(us.data1[i,16] == "8 - 10 years"){field.label[i] = 4}
  if(us.data1[i,16] == "11 - 20 years"){field.label[i] = 5}
  if(us.data1[i,16] == "21 - 30 years"){field.label[i] = 6}
  if(us.data1[i,16] == "31 - 40 years"){field.label[i] = 7}
  if(us.data1[i,16] == "41 years or more"){field.label[i] = 8}
}
us.data1 = cbind(us.data1, overall.label, field.label)
us.data2 = us.data1[-which(us.data1[,20]<us.data1[,21]),1:19]
nrow(us.data2)
```

[1] 23145

```
# solve the mistake 3
unique(us.data2$Age)
```

```
[1] "25-34"      "45-54"      "35-44"      "18-24"      "65 or over"
[6] "55-64"
```

```
unique(us.data2$`Overall Working Years`)
```

```
[1] "5-7 years"      "2 - 4 years"      "8 - 10 years"      "21 - 30 years"
[5] "11 - 20 years"  "41 years or more" "31 - 40 years"      "1 year or less"
```

Transformation to the Age column:

From the above result, we find the age column has 6 categories, so we only take the max value of each category and minus 18, for example, if the Age = “45-54”, we only take 54 and do the calculation  $54 - 18 = 36$ , then assign 36 to the new column “max.age”. And for category “65 or over”, we take the max value as 84 (because in general case, there are very little people working over 84) and do  $84 - 18 = 66$ , then assign 66 to the new column “max.age”.

Transformation to the Overall working years column:

From the above result, we find this column has 8 categories, so, we only take the min value of each category. For example, if “Overall working years” = “1 year or less”, then assign 0 to the new column “min.overall”, if “Overall working years” = “41 years or more”, then assign 41 to the new column “min.overall”.

```
n2 = nrow(us.data2)
max.age = c(1:n2)*0
for(i in 1:n2)
{
  if(us.data2[i,3] == "18-24"){max.age[i] = 6}
  if(us.data2[i,3] == "25-34"){max.age[i] = 16}
  if(us.data2[i,3] == "35-44"){max.age[i] = 26}
  if(us.data2[i,3] == "45-54"){max.age[i] = 36}
  if(us.data2[i,3] == "55-64"){max.age[i] = 46}
  if(us.data2[i,3] == "65 or over"){max.age[i] = 66}
}
min.overall = c(1:n2)*0
for(i in 1:n2)
{
  if(us.data2[i,15] == "1 year or less"){min.overall[i] = 0}
  if(us.data2[i,15] == "2 - 4 years"){min.overall[i] = 2}
  if(us.data2[i,15] == "5-7 years"){min.overall[i] = 5}
  if(us.data2[i,15] == "8 - 10 years"){min.overall[i] = 8}
  if(us.data2[i,15] == "11 - 20 years"){min.overall[i] = 11}
```

```

    if(us.data2[i,15] == "21 - 30 years"){min.overall[i] = 21}
    if(us.data2[i,15] == "31 - 40 years"){min.overall[i] = 31}
    if(us.data2[i,15] == "41 years or more"){min.overall[i] = 41}
  }

  us.data2 = cbind(us.data2, max.age, min.overall)
  us.data3 = us.data2[-which(us.data2[,20]<us.data2[,21]),1:19]
  nrow(us.data3)

```

[1] 23116

From, we have eliminate all the rows for which their age, years of experience in their field, and years of experience total are impossible, the final total number of observation is 23116.

(e)

```

# First, let find whether they have NA value
sum(is.na(us.data3[,7]))

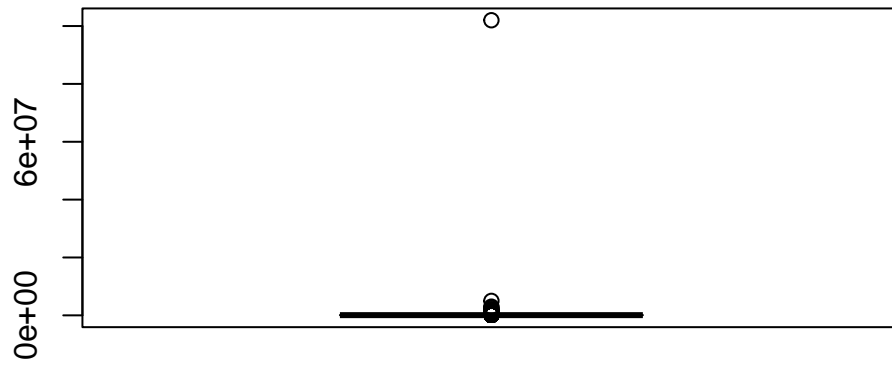
```

[1] 0

```

# From the result, we can find there is no NA value,
# So, we just need to eliminate the outliers of the salary
# In this section, we choose to use IQR Method to delete the outliers.
boxplot(us.data3[,7])

```



```
Q1 = quantile(us.data3[,7],0.25)
Q3 = quantile(us.data3[,7],0.75)
IQR = Q3 - Q1
IQR
```

```
75%
55644
```

```
low = Q1-1.5*IQR
up = Q3+1.5*IQR
low;up
```

```
25%
-27110
```

```
75%
195466
```

```
# since low is negative, but salary cannot be negative
# So, we just need to consider the data which is larger than Q3+1.5*IQR
us.data4 = us.data3[-which(us.data3[,7]>up),]
head(us.data4)
```

No.	Timestamp	Age	Industry
1	1 4/27/2021 11:02:10	25-34	Education (Higher Education)
3	3 4/27/2021 11:02:38	25-34	Accounting, Banking & Finance
4	4 4/27/2021 11:02:41	25-34	Nonprofits
5	5 4/27/2021 11:02:42	25-34	Accounting, Banking & Finance
6	6 4/27/2021 11:02:46	25-34	Education (Higher Education)
7	7 4/27/2021 11:02:51	25-34	Publishing

	Job Title	Job Title Supplement	Salary
1	Research and Instruction Librarian		55000
3	Marketing Specialist		34000
4	Program Manager		62000
5	Accounting Manager		60000
6	Scholarly Publishing Librarian		62000
7	Publishing Assistant		33000

	Monetary Compensation	Currency of Salary	Other Currency
1	0	USD	
3	NA	USD	
4	3000	USD	
5	7000	USD	
6	NA	USD	
7	2000	USD	

	Additional Context of Income Working Country	Working State(if in the US)
1	United States	Massachusetts
3	US	Tennessee
4	USA	Wisconsin
5	US	South Carolina
6	USA	New Hampshire
7	USA	South Carolina

	Working City	Overall Working Years	Working Years in your field
1	Boston	5-7 years	5-7 years
3	Chattanooga	2 - 4 years	2 - 4 years
4	Milwaukee	8 - 10 years	5-7 years
5	Greenville	8 - 10 years	5-7 years
6	Hanover	8 - 10 years	2 - 4 years
7	Columbia	2 - 4 years	2 - 4 years

	Highest education level	Gender	Race(indicate all)
1	Master's degree	Woman	White

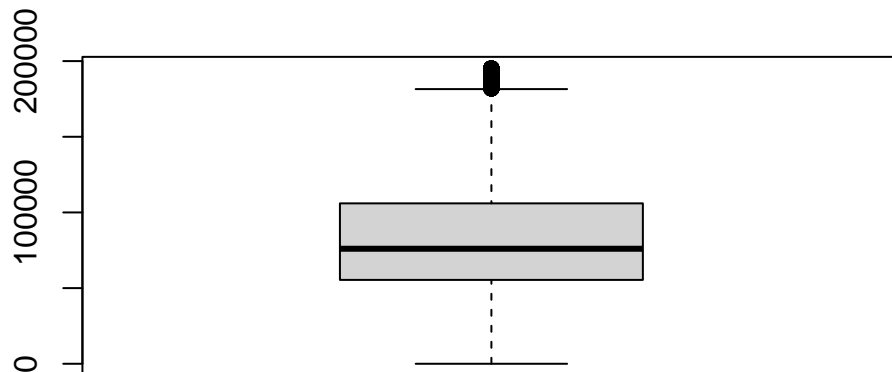


3	College degree	Woman	White
4	College degree	Woman	White
5	College degree	Woman	White
6	Master's degree	Man	White
7	College degree	Woman	White

```
nrow(us.data4)
```

```
[1] 22207
```

```
boxplot(us.data4[,7])
```



```
# from the new plot, we can find the new data is much better than before.
# and the new data set has 22207 observations
```

### Question 3

(a)

```

#' check palindromic
#' @param positive integer
#' @return a list which contain
#' whether it is a palindromic
#' and give us the reverse of the input
#' @examples if you input not a positive integer
#' it will stop run and give you a warning, if you input
#' 728827, it will return $isPalindromic TRUE, reversed 728827
isPalindromic = function(value)
{
  # we must assure that the input is a postive integer.
  if(!is.numeric(value))
  {
    warning("Input must be a positive integer.")
    stop("This input is not a numeric, please try again.")
  }
  if(abs(value) != value)
  {
    warning("Input must be a positive integer.")
    stop("This input is not a positive number, please try again.")
  }
  if(value != as.integer(value))
  {
    warning("Input must be a positive integer.")
    stop("This input is not an integer, please try again.")
  }
  # it is more easy to do operation on character rather than integer.
  value = as.integer(value)
  input = as.character(value)
  # this is to split the whole string into one character
  # and it will be easy to reverse, after reverse,
  # let the reversed single characters be pasted without blank.
  rev.input = paste(rev(unlist(strsplit(input, NULL))), collapse="")
  rev.value = as.integer(rev.input)
  # check whether it is a palindromic
  result <- (rev.value == value)
  list.data = list(result, rev.value)
  names(list.data) = c("isPalindromic", "reversed")
  return(list.data)
}

```

```
# you can verified the below three comments from the R code I uploaded in
# Github, we cannot render this quarto file if we execute the below
# three comments.
#isPalindromic("dasdas")
#isPalindromic(-12.88)
#isPalindromic(3456.897)
isPalindromic(7556)
```

```
$isPalindromic
[1] FALSE
```

```
$reversed
[1] 6557
```

```
isPalindromic(6789)
```

```
$isPalindromic
[1] FALSE
```

```
$reversed
[1] 9876
```

```
isPalindromic(8228)
```

```
$isPalindromic
[1] TRUE
```

```
$reversed
[1] 8228
```

## (b)

For palindromic, there exist 3 special cases.

1, Integer 0-8: for this case, we consider these number are palindromic, the next palindromic is number + 1, for example, if input is 8, then, output will be 9.

2, 9, 99, 999, ... , for this kind of number, the next palindromic is number + 2, for example, if you input is 99999, then, the output is 100001.

3 10, 100, 1000, ..., for this kind of number, the next palindromic is number + 1, for example, if you input is 100000, then, the output is 100001.

```

#' Find next palindromic
#' @param positive integer
#' @return the next palindromic which is
#' strictly larger than the input.
#' @examples if you input not a positive integer
#' it will stop run and give you a warning
#' if you input 7668, the output is 7777.
nextPalindrome = function(value)
{
  next.palindromic = 0
  if(!is.numeric(value))
  {
    warning("Input must be a positive integer.")
    stop("This input is not a numeric, please try again.")
  }
  if(abs(value) != value)
  {
    warning("Input must be a positive integer.")
    stop("This input is not a positive number, please try again.")
  }
  if(value != as.integer(value))
  {
    warning("Input must be a positive integer.")
    stop("This input is not an integer, please try again.")
  }

  #####
  # Special Case

  value = as.integer(value)
  if(value <= 8 & value >= 0 )
  {
    next.palindromic = value + 1
    return(next.palindromic)
  }
  if(log10(value) == as.integer(log10(value)))
  {
    next.palindromic = value + 1
    return(next.palindromic)
  }
}

```

```

if(log10(value+1) == as.integer(log10(value+1)))
{
  next.palindromic = value + 2
  return(next.palindromic)
}

#####
# Genral Case

# First, we need to find how many digits does this number have.
# it's more easy to do operations after converting the number to a string.
input = as.character(value)
n = nchar(input)
if(n%%2 == 0)
{
  # For the number has even digits:
  # Splitting the number into two halves.
  # Mirroring the left half.
  # If the mirrored left half number is larger than the right half number,
  # just paste the original left half with the mirrored left half,
  # and then it will be the next palindromic.

  # If the mirrored left half number is less or equal than
  # the right half number,
  # then, constantly increase the original left half number by 1,
  # then mirrored this new left half number,
  # then paste the new left half number with its mirrored one to form a
  # palindromic, then use this palindromic to compare with the input,
  # until this palindromic is larger than the input value
  # we can stop increasing the left half number
  # and output the palindromic.
  # This will be the next palindromic.
  leftinput = substr(input,1,n/2)
  rightinput = substr(input,((n/2)+1),n)
  rev.leftinput = paste(rev(unlist(strsplit(leftinput,NULL))),
                        collapse="")

  rev.leftvalue = as.integer(rev.leftinput)
  leftvalue = as.integer(leftinput)
  rightvalue = as.integer(rightinput)
  if(rev.leftvalue > rightvalue)

```

```

{
    char = paste(leftinput,rev.leftinput,sep = "")
    next.palindromic = as.integer(char)
    return(next.palindromic)
}
if(rev.leftvalue <= rightvalue)
{
    sign = TRUE
    while(sign)
    {
        leftvalue = leftvalue + 1
        leftinput = as.character(leftvalue)
        rev.leftinput = paste(rev(unlist(strsplit(leftinput,NULL))),
                             collapse="")
        char = paste(leftinput,rev.leftinput,sep = "")
        palindromic = as.integer(char)
        if(palindromic > value)
        {
            next.palindromic = palindromic
            sign = FALSE
            return(next.palindromic)
        }
    }
}
}else
{
    # For the number has odd digits
    # the odd digits number has the middle digit
    # splitting the number into left half number (without the middle digit),
    # left-middle half number (means left half + middle digit)
    # and the right half number.
    # if the left half number is larger than the right half number,
    # just paste the original left half number, middle digits,
    # and with the mirrored left half number,
    # and then it will be the next palindromic.

    # if the left half number is less or equal the right half number,
    # constantly increase the left-middle half number by 1,
    # mirror this left-middle half number without mirroring the middle
    # digits (which means after increaseing, we only mirror the left half,
    # but donot mirror the middle digit), then paste the increased

```

```

# left-middle half number with the mirrored left half number together,
# this will be a palindromic, then, compare this palindromic with
# the input value,
# until this palindromic is larger than the input value
# we can stop increasing the left-middle half number,
# and output the palindromic.
# This will be the next palindromic.
leftinput = substr(input,1,(n-1)/2)
left.middle.input = substr(input,1,(n+1)/2)
rightinput = substr(input,(((n+1)/2)+1),n)
rev.leftinput = paste(rev(unlist(strsplit(leftinput,NULL))),
                      collapse="")

rev.leftvalue = as.integer(rev.leftinput)
leftvalue = as.integer(leftinput)
left.middle.value = as.integer(left.middle.input)
rightvalue = as.integer(rightinput)
if(rev.leftvalue > rightvalue)
{
  char = paste(left.middle.input,rev.leftinput,sep = "")
  next.palindromic = as.integer(char)
  return(next.palindromic)
}
if(rev.leftvalue <= rightvalue)
{
  sign = TRUE
  while(sign)
  {
    left.middle.value = left.middle.value + 1
    left.middle.input = as.character(left.middle.value)
    num = nchar(left.middle.input)
    leftinput = substr(left.middle.input,1,(num-1))
    rev.leftinput = paste(rev(unlist(strsplit(leftinput,NULL))),
                        collapse="")

    char = paste(left.middle.input,rev.leftinput,sep = "")
    palindromic = as.integer(char)
    if(palindromic > value)
    {
      next.palindromic = palindromic
      sign = FALSE
      return(next.palindromic)
    }
  }
}

```

```

    }
  }
}

}
# Test Case:
nextPalindrome(728827)

```

[1] 729927

```
nextPalindrome(39951)
```

[1] 39993

```
nextPalindrome(7152)
```

[1] 7227

```
nextPalindrome(765431537)
```

[1] 765434567

```
nextPalindrome(999)
```

[1] 1001

```
nextPalindrome(10000)
```

[1] 10001

```
nextPalindrome(1800)
```

[1] 1881

(c)



```
nextPalindrome(391)
```

```
[1] 393
```

```
nextPalindrome(9928)
```

```
[1] 9999
```

```
nextPalindrome(19272719)
```

```
[1] 19277291
```

```
nextPalindrome(109)
```

```
[1] 111
```

```
nextPalindrome(2)
```

```
[1] 3
```