

MATH40005 Coursework Spring 2023

CID: 12345678

Part A (1 mark)

```
# by inspection of the file can see is semi-colon separated
df <- read.table("salaries.txt", sep=';', header=TRUE)
```

- 1 mark for reading the data in correctly. Check the sep and header parameters.

Part B (1 mark)

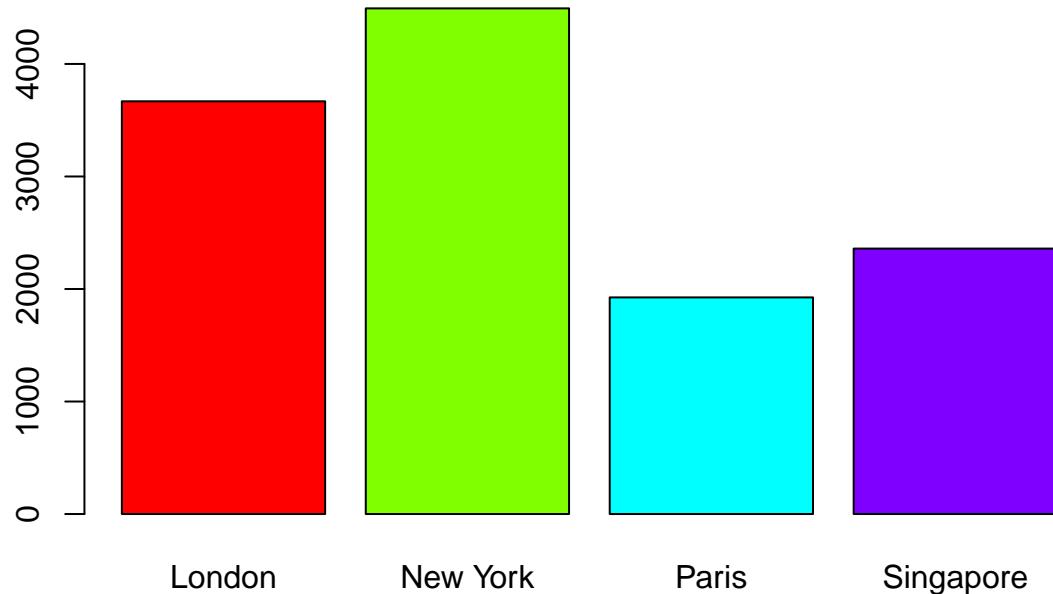
```
cat("Median salary is: ", median(df$salary), "\n", sep="")
#> Median salary is: 47125
```

- 1 mark for computing the median and printing to screen correctly.

Part C (2 marks)

```
# compute the number of respondents from each city  
t <- table(df$city)  
# create a bar chart; colours of bars optional  
barplot(t, col=rainbow(length(t)), main="Number of respondents per city")
```

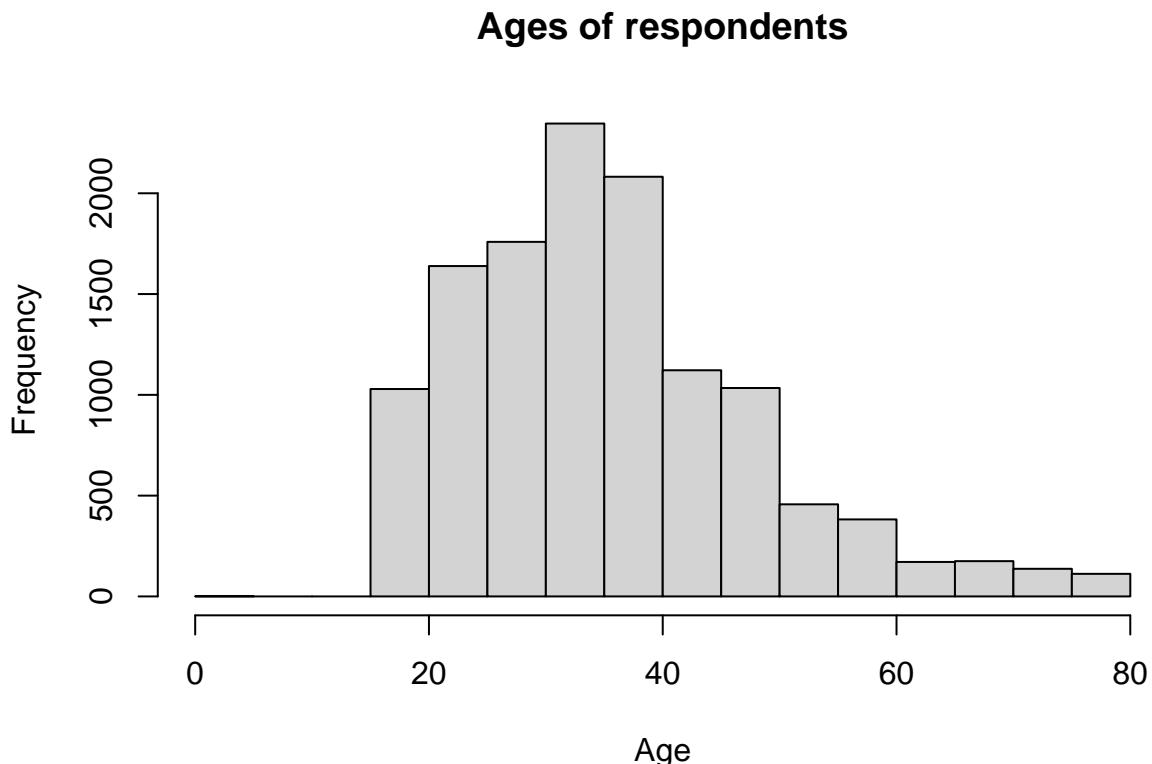
Number of respondents per city



- 1 mark for counting the number of respondents per city correctly
- 1 mark for bar chart. Pie chart would show proportions, so not preferred here.

Part D (1 mark)

```
# histogram of the number of respondents from each city  
hist(df$age, main="Ages of respondents", xlab="Age")
```



- 1 mark for histogram. A bar chart splitting up the ages into categories, e.g. 20-29, 30-39, etc, would also be fine. But a bar chart for each age would not be acceptable. A box plot would be acceptable, but a histogram is better here.

Part E (2 marks)

There are multiple possible ways to solve this part.

Part E: first method

The first method is to subset each city manually.

```
# subsetting each city manually
londonsalaries <- df$salary[df$city=="London"]
cat("Mean salary for London is", round(mean(londonsalaries), 2), "\n")
#> Mean salary for London is 47430.99

newyorksalaries <- df$salary[df$city=="New York"]
cat("Mean salary for New York is", round(mean(newyorksalaries), 2), "\n")
#> Mean salary for New York is 58584.41

parissalaries <- df$salary[df$city=="Paris"]
cat("Mean salary for Paris is", round(mean(parissalaries), 2), "\n")
#> Mean salary for Paris is 46152.79

singaporesalaries <- df$salary[df$city=="Singapore"]
cat("Mean salary for Singapore is", round(mean(singaporesalaries), 2), "\n")
#> Mean salary for Singapore is 58520.08
```

Part E: second method

The second method is similar to the first method, but using a for loop, since we are repeating the same procedure multiple times.

```
# compute the mean salary per city, using a for loop
# the 'unique' function gets unique values in a vector
cities <- unique(df$city)
for (city in cities){
  citysalaries <- df$salary[df$city==city]
  cat("Mean salary for", city, "is", round(mean(citysalaries), 2), "\n")
}
#> Mean salary for Singapore is 58520.08
#> Mean salary for Paris is 46152.79
#> Mean salary for London is 47430.99
#> Mean salary for New York is 58584.41
```

Part E: third method

The third method is to use the built-in aggregate function:

```
# compute the mean salary per city
res <- aggregate(df$salary, list(df$city), mean)
#print(res)
# create string result
s <- paste0("Mean salary for ", res$Group.1, " is ", round(res$x, 2), "\n")
#print string to screen
cat(s)
#> Mean salary for London is 47430.99
#> Mean salary for New York is 58584.41
```

```
#> Mean salary for Paris is 46152.79  
#> Mean salary for Singapore is 58520.08
```

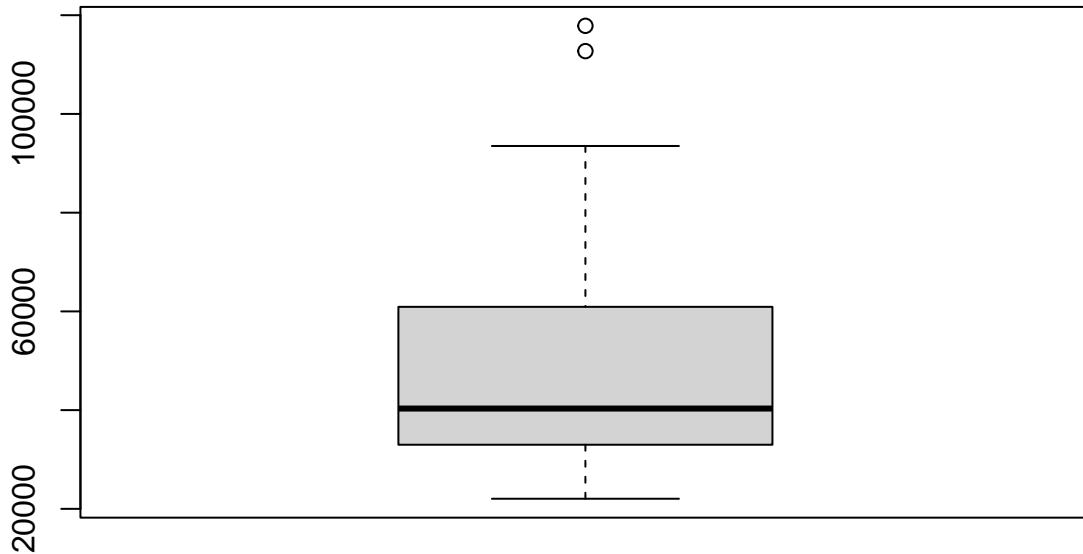
In fact, there are also other methods, some of which use additional R packages, but the recommended approach is the second or third method.

- 1 mark for computing means correctly; any method above is fine.
- 1 mark for outputting to screen AND the numbers are rounded to 2 decimal places.

Part F (3 marks)

```
df_L <- df[df$city=="London", ]
cat("Outlier salaries:", boxplot.stats(df_L$salary)$out, "\n")
#> Outlier salaries: 112725 117849
boxplot(df_L$salary, main="Boxplot of respondents from London")
```

Boxplot of respondents from London



The rule used in the function `boxplot.stats` and the boxplot to find or display outliers is as follows: first compute the lower and upper quartiles of the data as $q_{0.25}$ and $q_{0.75}$, respectively. Then, compute the interquartile range as $IQR = q_{0.75} - q_{0.25}$. If, for a value x , either of the two conditions hold

$$\begin{aligned} x &> q_{0.75} + 1.5IQR, \\ x &< q_{0.25} - 1.5IQR, \end{aligned}$$

then x is considered to be an outlier. Using this criterion, two values were found to be outliers in among the respondents from London, namely 112725 and 117849.

- 1 mark for correctly finding the outliers
- 1 mark for correctly explaining the criterion
- 1 mark for the box plot

Part G (3 marks)

The key idea is to create a boolean index vector for each sub-dataframe which ensures the conditions hold.

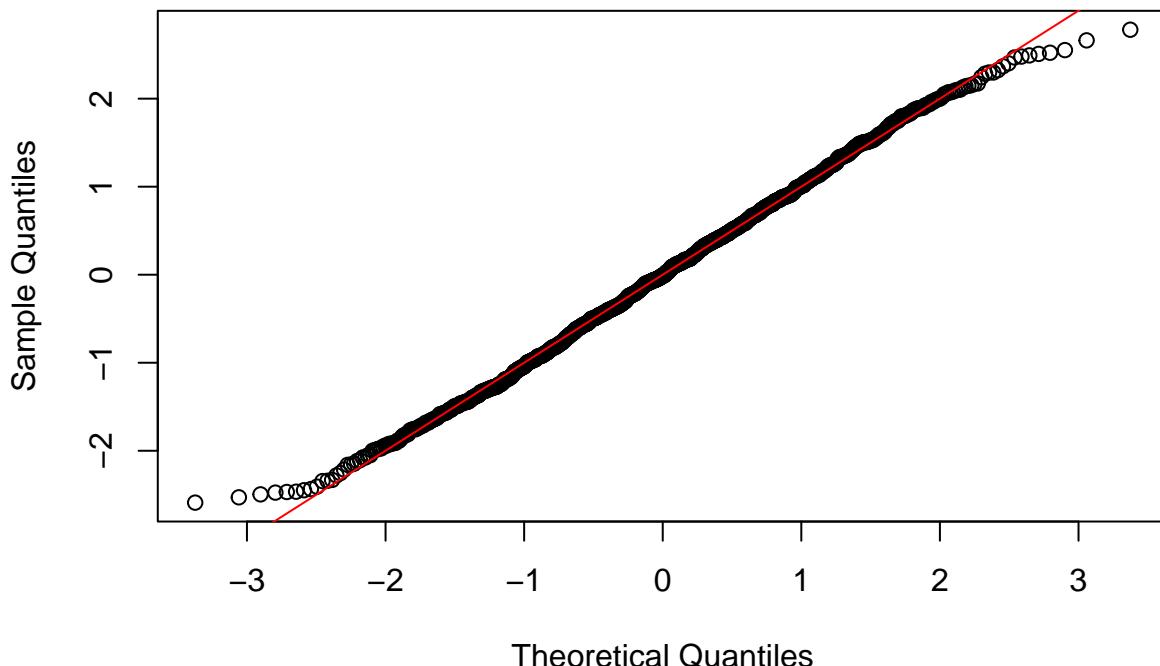
```
ageindex <- (df$age >= 20) & (df$age <= 29)
nyindex <- df$city=="New York" & ageindex
singindex <- df$city=="Singapore" & ageindex
df_NY <- df[nyindex, ]
df_S <- df[singindex, ]
s <- "Mean salary for respondents aged 20-29 in"
cat(s, "New York:", round(mean(df_NY$salary), 2), "\n")
#> Mean salary for respondents aged 20-29 in New York: 38915.8
cat(s, "Singapore:", round(mean(df_S$salary), 2), "\n")
#> Mean salary for respondents aged 20-29 in Singapore: 39217.59
```

- 1 mark for correctly subsetting for the New York data.
- 1 mark for correctly subsetting for the Singapore data.
- 1 mark for correctly computing the means and printing to screen.

Part H (3 marks)

```
z_NY <- (df_NY$salary - mean(df_NY$salary)) / sd(df_NY$salary)
qqnorm(z_NY, main="Normal Q-Q plot for respondents 20-29 from New York")
abline(0, 1, col="red")
```

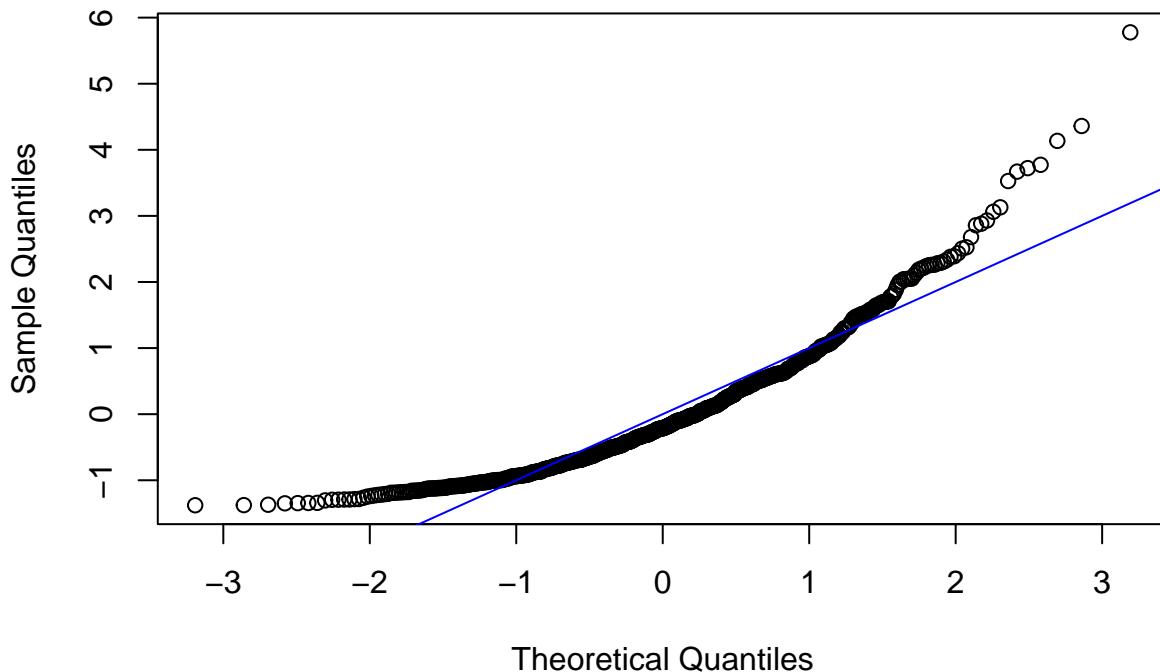
Normal Q-Q plot for respondents 20–29 from New York



The Q-Q plot above for the salaries of respondents from New York aged 20-29 suggests that the data is normally distributed, because most values lie on the line $y = x$.

```
z_S <- (df_S$salary - mean(df_S$salary)) / sd(df_S$salary)
qqnorm(z_S, main="Normal Q-Q plot for respondents 20-29 from Singapore")
abline(0, 1, col="blue")
```

Normal Q–Q plot for respondents 20–29 from Singapore



The Q–Q plot above for the salaries of respondents from New York aged 20–29 suggests that the data is NOT normally distributed, because many of the values do not lie on the line $y = x$.

- 1 mark for creating a Q–Q plot for the New York and Singapore data sets.
- 1 mark for correctly interpreting that the Q–Q plot for the New York data suggests that the data follows a normal distribution.
- 1 mark for correctly interpreting that the Q–Q plot for the Singapore data suggests that the data does NOT follow a normal distribution.

Part I (3 marks)

```
mytest <- function(x, y, alpha){  
  # compute pooled standard deviation  
  n <- length(x)  
  m <- length(y)  
  s_p_sq <- ( (n-1) * var(x) + (m-1) * var(y) ) / (n+m - 2)  
  s_p <- sqrt(s_p_sq)  
  
  # compute the t-statistic  
  t_stat <- ( mean(x) - mean(y) ) / ( s_p * sqrt(1/n + 1/m) )  
  
  # compute the threshold for the t-statistic, based on alpha  
  t_thresh <- qt(1 - alpha/2, df=(n+m-2))  
  
  # output result of test  
  cat("t-statistic:", t_stat, "\n")  
  cat("alpha:", alpha, "\n")  
  cat("threshold:", t_thresh, "\n")  
  cat("decision: ")  
  if (abs(t_stat) > t_thresh){  
    cat("reject\n")  
  } else {  
    cat("fail to reject\n")  
  }  
}  
mytest(df_NY$salary, df_S$salary, 0.05)  
#> t-statistic: -2.196794  
#> alpha: 0.05  
#> threshold: 1.96112  
#> decision: reject
```

- 1 mark for writing a function that produces output in the correct format, as per the question. Note that the built-in `t.test` may not be used.
- 1 mark for correctly computing the t-statistic (and the pooled variance) correctly.
- 1 mark for computing the threshold correctly.

Presentation (1 mark)

- 1 mark if report is presented well, CID at top of report, code commented, each question starts on a new page.