

Tutorial 1

STAT5002 - Introduction to statistics

March 8, 2018

Summary

In the first lecture, the most major objectives I want to achieve was to get you all familiar are: 1. Creating RProject and place all of your data in the same folder. 2. Being able to read in data into R via functions like `read.csv`. 3. Creating your own RMarkdown file and calculate simple graphical and numerical summaries.

There were some R functions in this tutorial which we didn't cover in the lecture. We will cover them in Week 02. But before that, you should try Google these functions and try them out before approaching your tutors about it.

This week we also discussed the ways that data can be collected and potential sources of bias. We also examined various approaches for exploring and summarising data numerically and graphically and their limitations. Remember, when choosing a graphical or numerical summary think...

- Why am I using this summary?
- What properties of the data will this summary highlight?
- Is this summary appropriate for communicating what I want to communicate?

When using a summary think...

- What is this summary showing me?
- What is this summary not showing me?

And remember,

- Always think critically about how your data was collected, when exploring raw or summarised data, and, when someone offers an interpretation of data.

Question 1: Simpson's Paradox

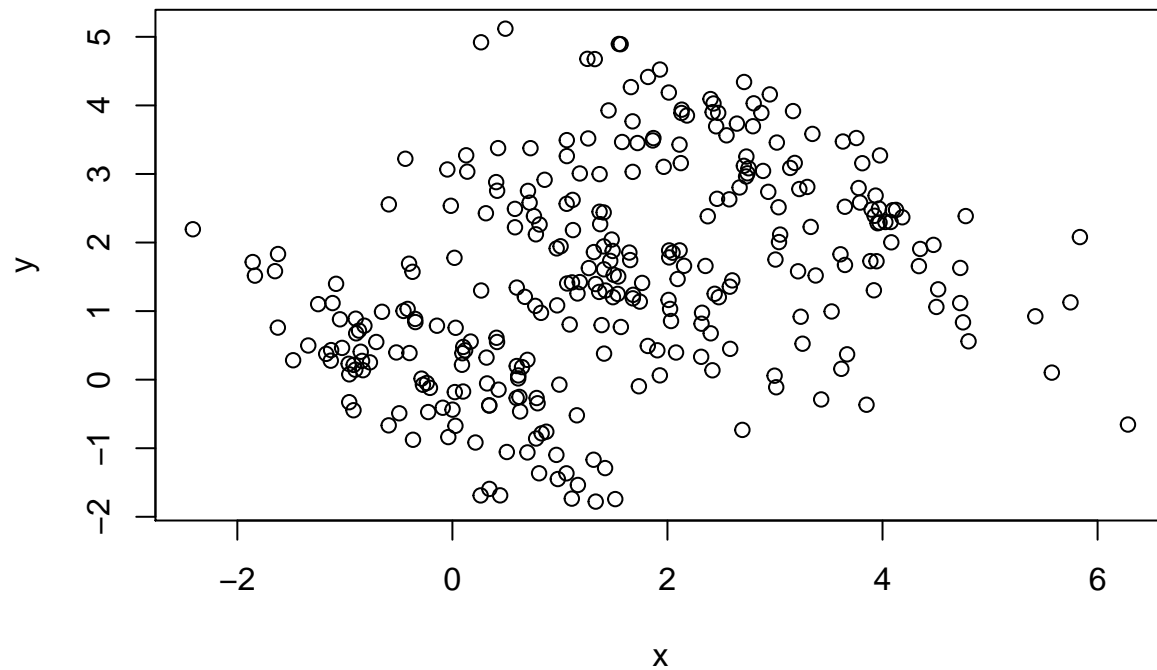
Three groups of students were asked to sit two tests. The (standardised) test scores were recorded as `x` and `y` in the data file `Simpsons.csv`. The group information was recorded as `group`, which is an indication of the high school years completed by the students.

(a). Read the data `Simpsons.csv` into R. What does `header = TRUE` do? (Hint: change it to `FALSE` to see what happens.) Also run the `summary()` function on the data.

```
      x              y      group
Min. :-2.4137 Min. :-1.7786 Min. :1
1st Qu.: 0.3374 1st Qu.: 0.3746 1st Qu.:1
Median : 1.4073 Median : 1.4144 Median :2
Mean : 1.5110 Mean : 1.4912 Mean :2
3rd Qu.: 2.6773 3rd Qu.: 2.5812 3rd Qu.:3
Max. : 6.2826 Max. : 5.1187 Max. :3
```

(b). Extract `x = dat$x` and `y = dat$y`, and then create a scatter plot of these two variables. What can you conclude from this data? Is the trend between `x` and `y` positive or negative? In other words, if a student scored well in the first test (`x`), is the student more likely to achieve a good score or a poor score in the second test (`y`)? (If you know about correlation, then you can also try to run `cor(x,y)`).

```
x = dat$x
y = dat$y
plot(x, y)
```



```
cor(x, y)
```

```
[1] 0.3129037
```

(c). Calculate the mean, median, standard deviation, and IQR of x and y. (Hint: use in-built functions for these.)

```
mean(x)
```

```
## [1] 1.511021
```

```
mean(y)
```

```
## [1] 1.491241
```

```
median(x)
```

```
## [1] 1.407315
```

```
median(y)
```

```
## [1] 1.414423
```

```
sd(x)
```

```
## [1] 1.66287
```

```
sd(y)
```

```
## [1] 1.531393
```

```
IQR(x)
```

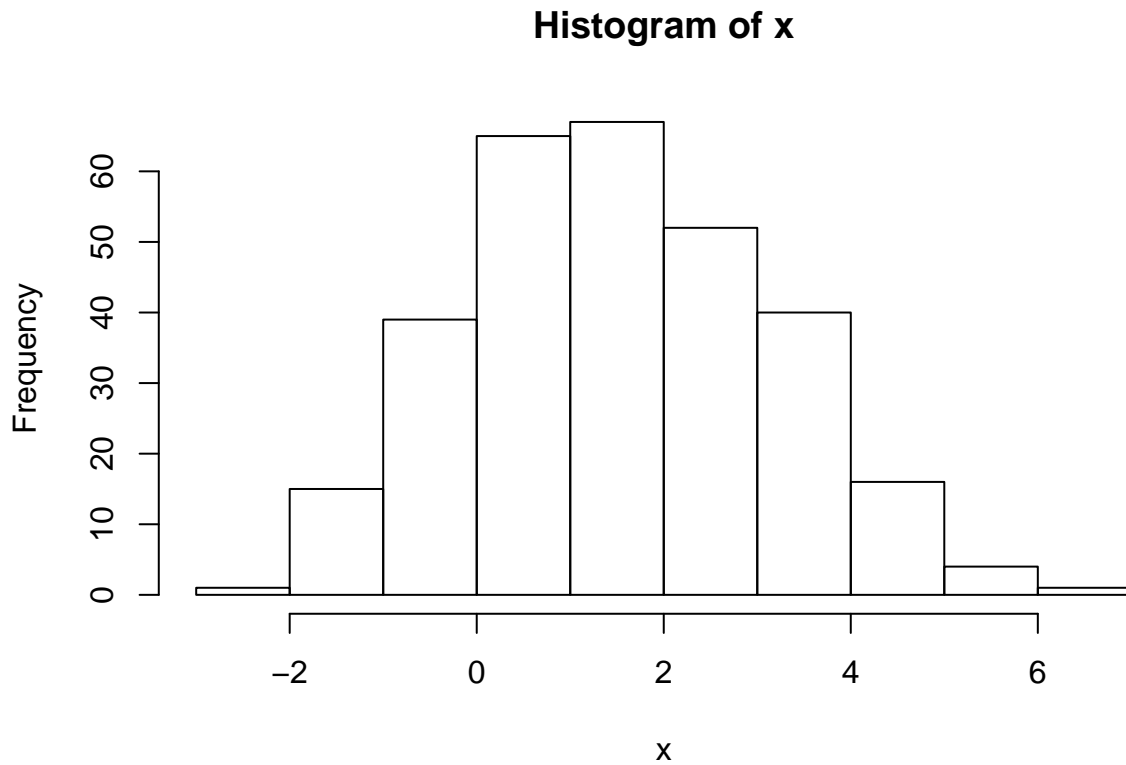
```
## [1] 2.339899
```

```
IQR(y)
```

```
## [1] 2.206554
```

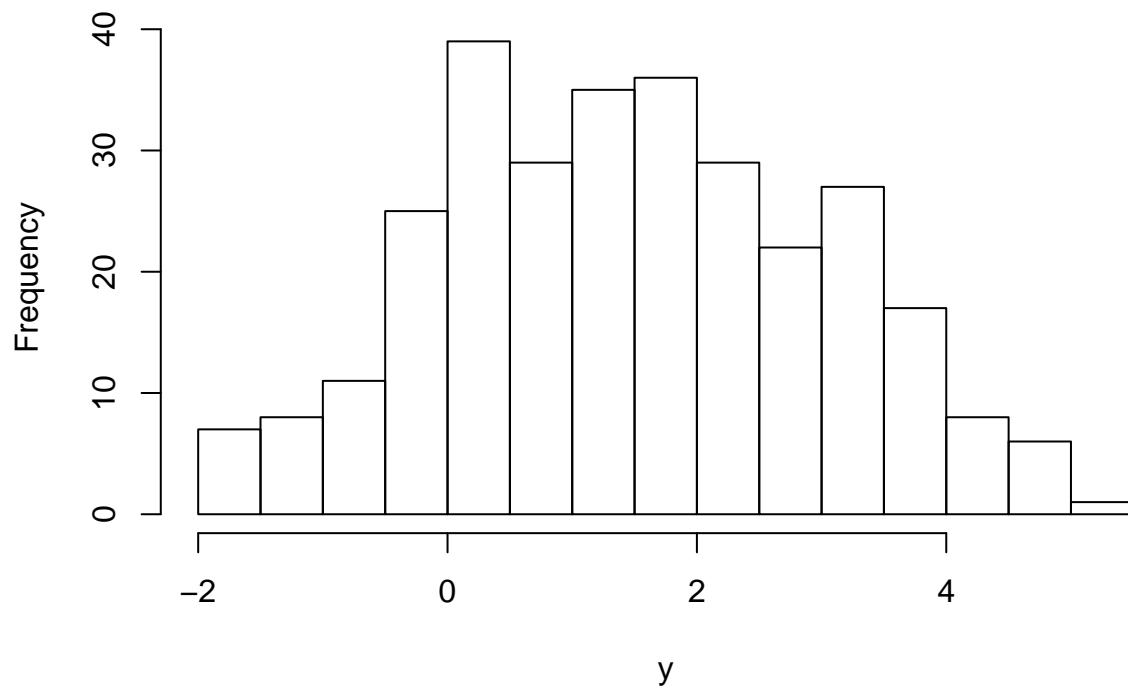
(d). Create histograms and boxplots for x and y separately. Are there any outliers that could potentially inflate/deflate the values of above statistics?

```
hist(x)
```

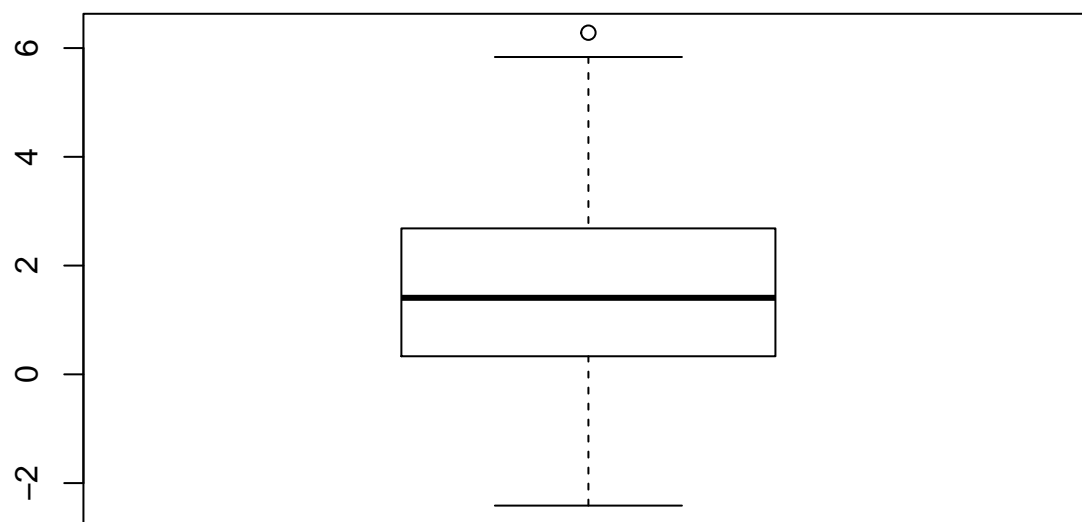


```
hist(y)
```

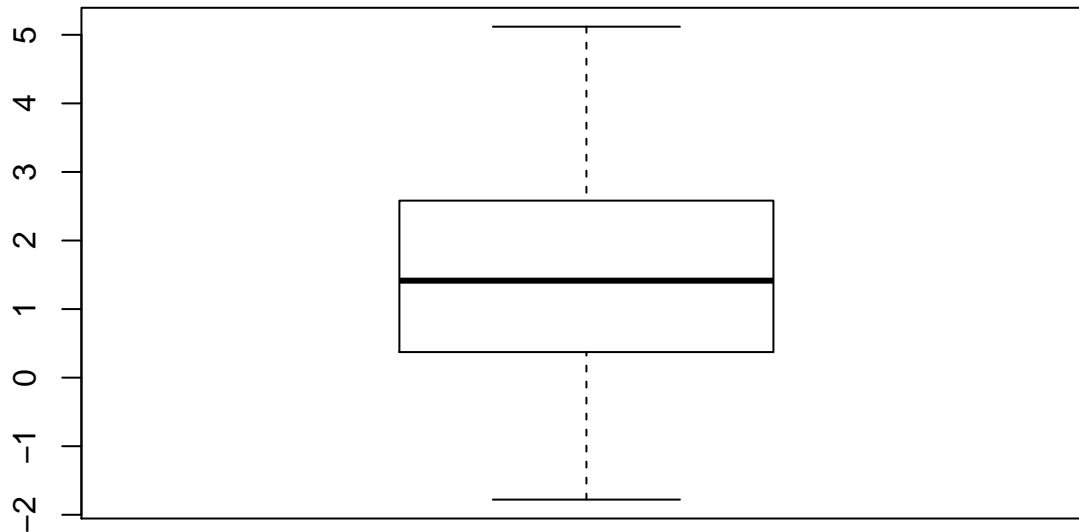
Histogram of y



```
boxplot(x)
```



```
boxplot(y)
```



(e). Check the documentations on the `quantile` function by typing `?quantile` into R. Then, calculate the median of `x` and `y` using `type = 1` and `type = 7`. Which one is “more” valid?

```
quantile(x, 0.5, type = 7)
```

```
##      50%
## 1.407315
```

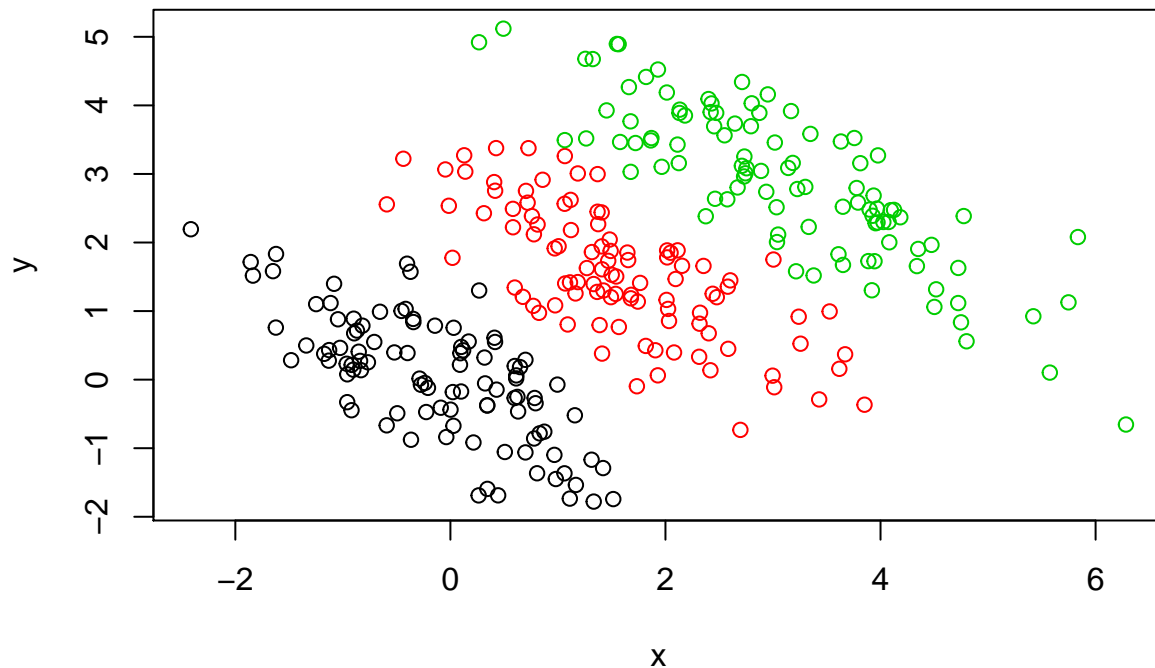
```
quantile(x, 0.5, type = 1)
```

```
##      50%
## 1.406826
```

These different definitions equally valid.

(f). Create `group = dat$group` and then make another scatter plot for `x` and `y` again. But this time, use `plot(x, y, col=group)`. What is the trend between `x` and `y` for each `group` now? Should your conclusion in (b) be altered?

```
group = dat$group
plot(x,y,col = group)
```



(g). Repeat part (c), but now, calculate these statistics within each group. You can either use `mean(x[group == 1])` etc, or you could look up `?tapply` and perform these calculations faster. What can you conclude from these statistics when split by groups?

```
tapply(x, group, mean)
```

```
##          1          2          3
## -0.1377196  1.5542109  3.1165706
```

```
tapply(y, group, mean)
```

```
##          1          2          3
## 0.03418449  1.54757076  2.89196683
```

```
tapply(x, group, median)
```

```
##          1          2          3
## -0.06389448  1.44551022  3.03528976
```

```
tapply(y, group, median)
```

```
##          1          2          3
## 0.1416216  1.4579536  3.0200473
```

```
tapply(x, group, sd)
```

```
##          1          2          3
## 0.8545093  0.9235797  1.1901965
```

```
tapply(y, group, sd)
```

```
##          1          2          3
## 0.8997503  0.9372806  1.1248618
```

```
tapply(x, group, IQR)
```

```
##          1          2          3
## 1.454476  1.110448  1.620276
```

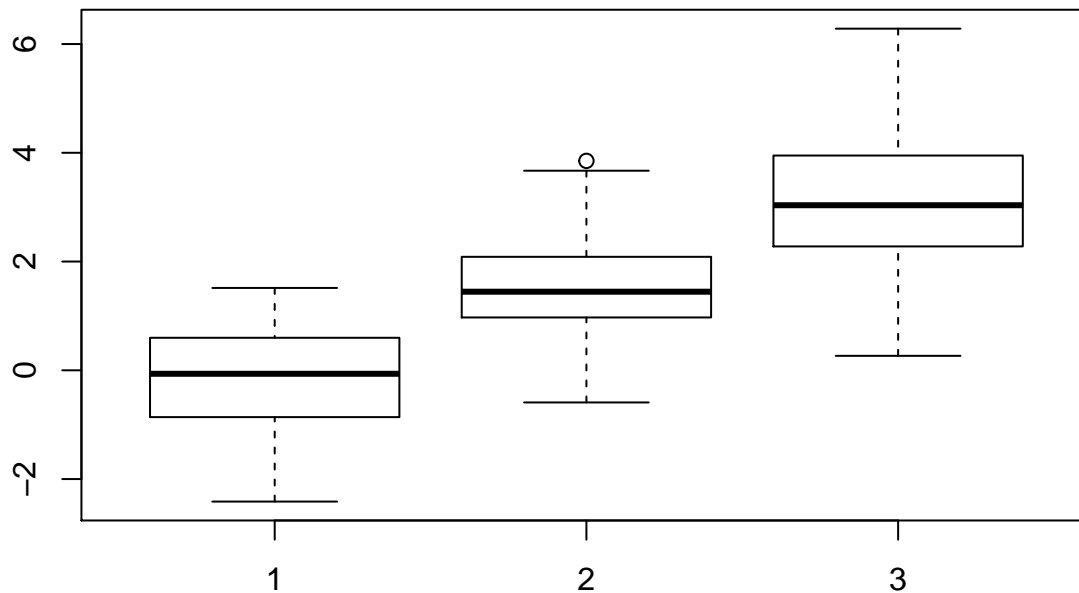
```
tapply(y, group, IQR)
```

```
##          1          2          3  
## 1.046477 1.257155 1.437663
```

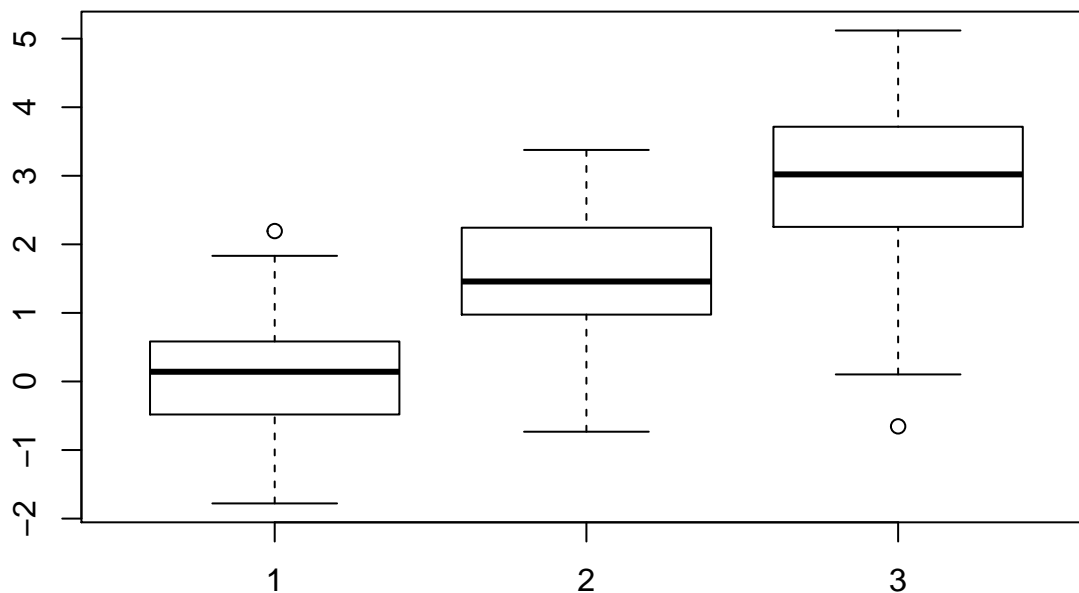
It should be clear that as the group number increase, the mean/median also increased. Other statistics more or less had the same value across the groups.

(h). Create a boxplot for x and y , but now, try to split the boxplot by the `group` variable.

```
boxplot(x ~ group)
```



```
boxplot(y ~ group)
```



(i). What did you learn from this exercise? Why was the conclusion reversed when the extra `group` variable was taken into account? Is pooling data from different sources always a good idea? How should we analyse data when a confounding variable is present?

Some possible ideas:

- This is a case of analysing the data without realising there is a potentially important confounder in the data.
- When confounders are taken into account, (depending on its strength) it is possible that our previous conclusion might not hold anymore.
- In this data, the mean/median of x and y both increase as **group** increases. Thus, if **group** is not taken into account, the comparison between the pooled x and pooled y showed that there is a potentially increasing trend.
- By splitting the data into groups, we can compare the data **within** a group, thus the comparison between x and y is more fair.
- This should be an important lesson in experimental design and confounding variables. Insights into your data can often arise when controlling important variables.

(j). Reading exercise: read the Wikipedia article on “Simpson’s paradox”.

Question 2

A government agency wanted to know on average, how many glasses of water does a person drink a day. To do this they randomly called the home phones of 20 people. As part of their survey they asked if people fell into one of 3 age groups that represent the lower (L), middle (M) and upper (U) third of the population.

<i>Glasses</i>	8	7	3	4	3	3	2	4	5	8	9	2	4	5	6	3	2	4	9	100
<i>Age</i>	L	M	U	U	U	M	U	M	M	L	L	M	M	U	U	U	U	U	M	L

- Characterise and explore the data using numerical and graphical summaries. What do you notice?
- Calculate the mean and median number of glasses of water that were drunk by the survey participants.
- Calculate the mean and median number of glasses of water that were drunk by survey participants in each age group.
 - Which of these are more appropriate estimates of the usual number of glasses drunk by a person from a particular age group in the general population?
- Calculate an appropriate estimate of the number of glasses of water drunk by a person from the general population.
 - Why is this an appropriate estimate?
 - Use one or two plots to help communicate your answer.

a)

From these summaries we might see...

- There are more older people than younger people.
- The large outlier (100) in glasses. Why is this an outlier (i.e. different to the bulk of the dataset).
 - Maybe it was suppose to be 10 but got entered in wrong,
 - Maybe someone gave a silly answer over the phone,
 - Maybe someone actually drinks 100 glasses of water a day (ultra marathon runner???)
 - Maybe someone drinks out of tiny paper shot glasses at work.
- We also see that maybe older people drink less water than younger people.

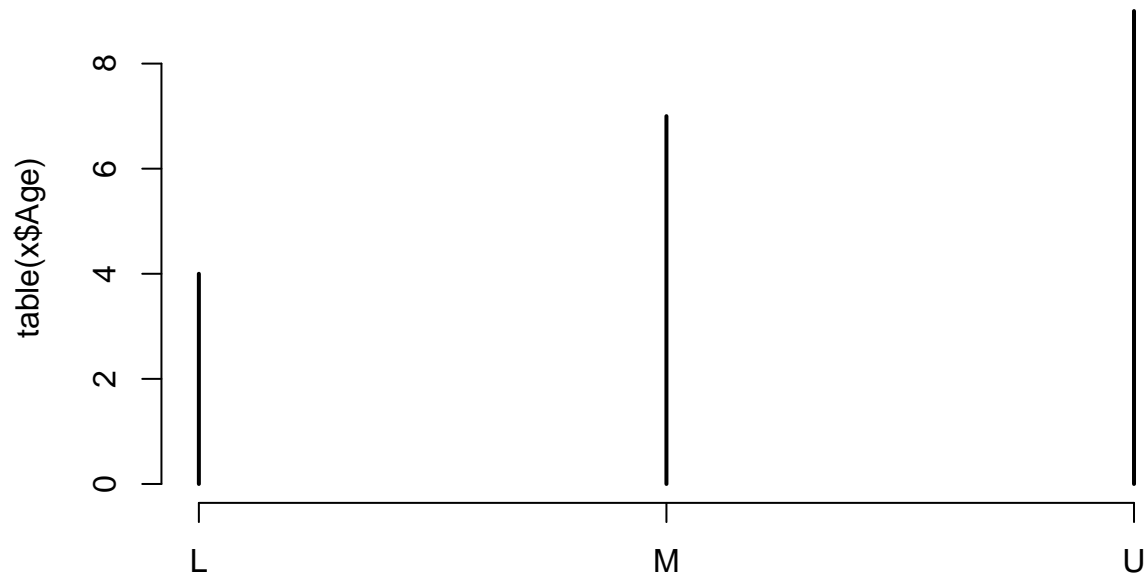
```
x = data.frame(Glasses = c(8,7,3,4,3,3,2,4,5,8,9,2,4,5,6,3,2,4,9,100),
Age = c('L','M','U','U','U','M','U','M','M','L','L','M','M','U','U','U','U','U','M','L'))

#Numerical summary
summary(x)
```



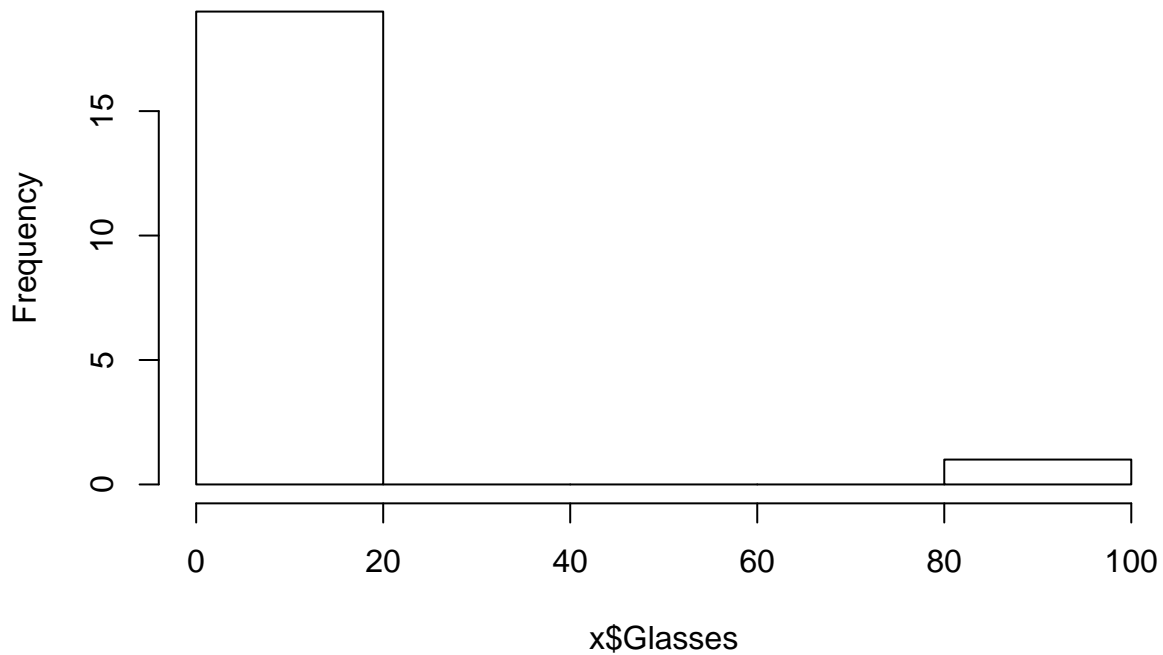
```
##      Glasses      Age
## Min.   : 2.00    L:4
## 1st Qu.: 3.00    M:7
## Median : 4.00    U:9
## Mean   : 9.55
## 3rd Qu.: 7.25
## Max.   :100.00
```

```
#Some ideas for graphical summaries
plot(table(x$Age))
```



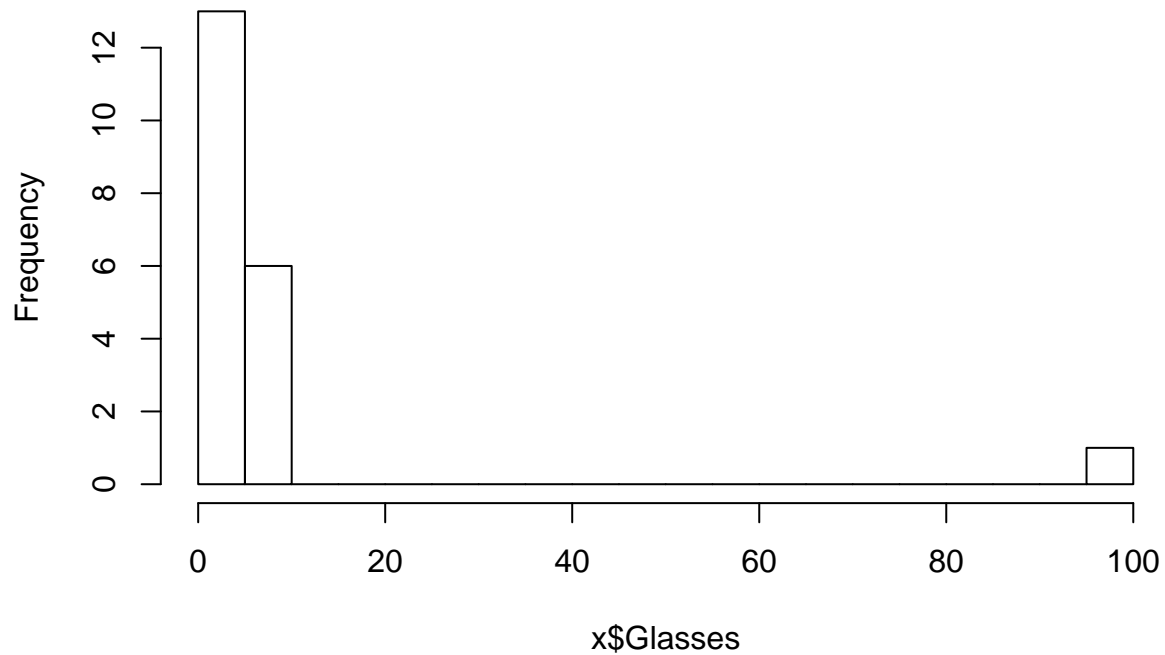
```
hist(x$Glasses)
```

Histogram of x\$Glasses

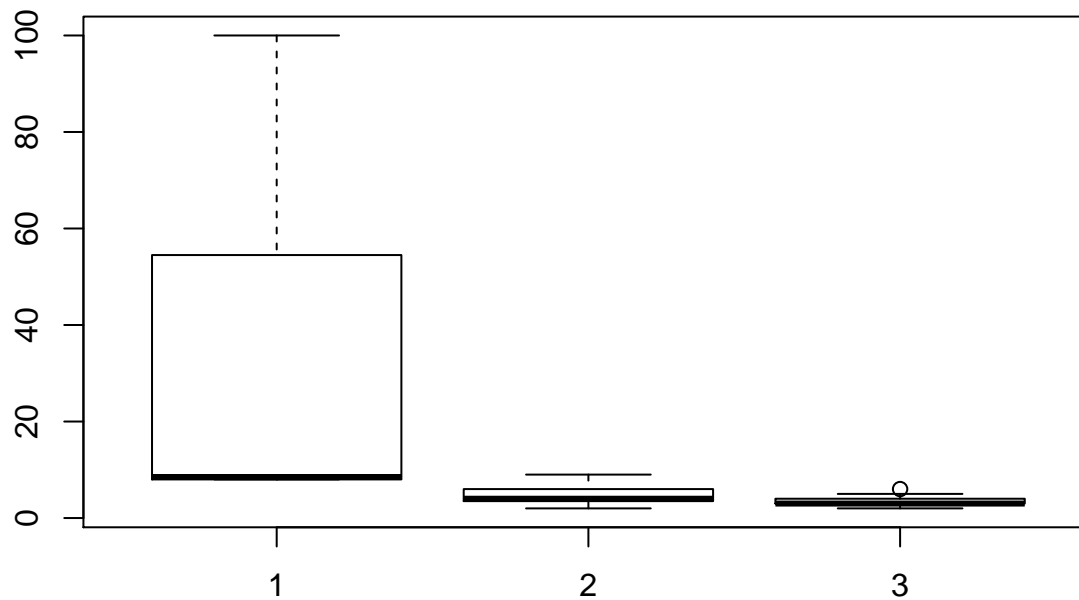


```
hist(x$Glasses,breaks = 20)
```

Histogram of x\$Glasses



```
boxplot(x$Glasses[x$Age=='L'],x$Glasses[x$Age=='M'],x$Glasses[x$Age=='U'])
```



```
#boxplot(x$Glasses~x$Age)
```

```
### Let's get rid of the extreme value
```

```

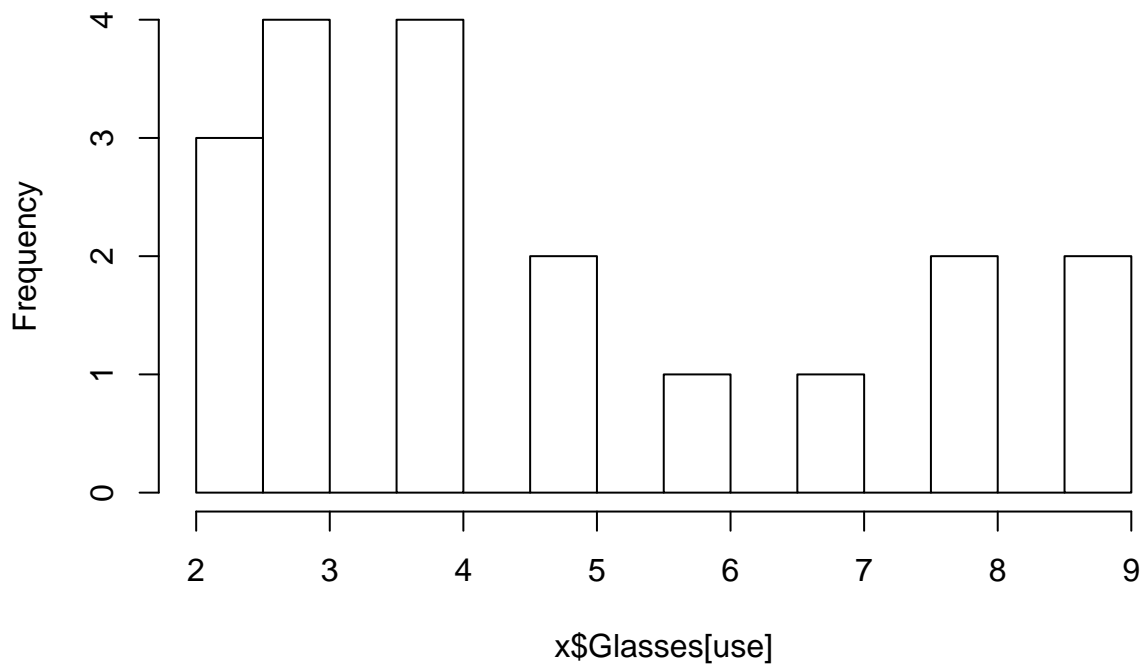
iqr = quantile(x$Glasses,.75) - quantile(x$Glasses,.25)

#Use an arbitrary rule to filter out extreme values (outliers)
use = x$Glasses > median(x$Glasses) - 1.5*iqr & x$Glasses < median(x$Glasses) + 1.5*iqr

hist(x$Glasses[use],breaks = 20)

```

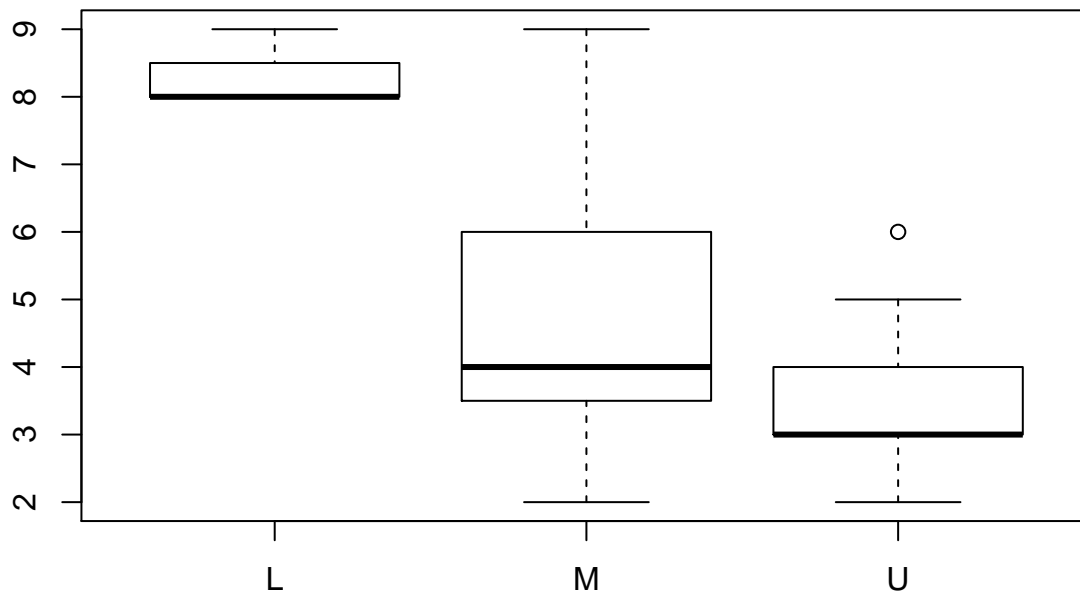
Histogram of x\$Glasses[use]



```

boxplot(x$Glasses[use]~x$Age[use])

```



```

#boxplot(x$Glasses[x$Age=='L'&use],x$Glasses[x$Age=='M'&use],x$Glasses[x$Age=='U'&use])

```

b)

The mean of the number of glasses drunk is dramatically larger than the median. This is pulled up by the 100 value.

```
mean(x$Glasses)
```

```
## [1] 9.55
```

```
median(x$Glasses)
```

```
## [1] 4
```

c)

- We see that the mean and median are similar for the older and middle age groups but there is a large difference in the younger.
- We also see the evidence that young people might drink more water.

```
mean(x$Glasses[x$Age=='L'])
```

```
## [1] 31.25
```

```
mean(x$Glasses[x$Age=='M'])
```

```
## [1] 4.857143
```

```
mean(x$Glasses[x$Age=='U'])
```

```
## [1] 3.555556
```

```
median(x$Glasses[x$Age=='L'])
```

```
## [1] 8.5
```

```
median(x$Glasses[x$Age=='M'])
```

```
## [1] 4
```

```
median(x$Glasses[x$Age=='U'])
```

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

```
### Or if you are lazy/comfortable
```

```
lapply(split(x$Glasses,x$Age),mean)
```

```
## $L
```

```
## [1] 31.25
```

```
##
```

```
## $M
```

```
## [1] 4.857143
```

```
##
```

```
## $U
```

```
## [1] 3.555556
```

```
lapply(split(x$Glasses,x$Age),median)
```

```
## $L
```

```
## [1] 8.5
```

```
##
```

```
## $M
```

```
## [1] 4
##
## $U
## [1] 3
```

d)

To get an estimate of the population's center (mode) we have many options, none are "right". One might simply take the mean of all samples (or the median to remove the influence of the outlier).

By definition, all the age groups should be seen equally in our survey, however we do not observe this. This may be due to response bias. We might be able to work around this bias by weighting the estimates of center from each of the groups equally. I like taking the mean of the medians... Taking the medians within each age group reduces the influence of the outlier. I am more comfortable taking the mean of the three age groups than the median as I "feel" this is better (I "feel" that a mean of 3 numbers is probably better than the median of 3 numbers).

```
#Naive
mean(x$glasses)

## Warning in mean.default(x$glasses): argument is not numeric or logical:
## returning NA
## [1] NA
median(x$glasses)

## Warning in is.na(x): is.na() applied to non-(list or vector) of type 'NULL'
## NULL

#Weighting all age groups equally.
mean(mean(x$Glasses[x$Age=='L']), mean(x$Glasses[x$Age=='M']), mean(x$Glasses[x$Age=='U']))

## [1] 31.25
median(c(median(x$Glasses[x$Age=='L']), median(x$Glasses[x$Age=='M']), median(x$Glasses[x$Age=='U'])))

## [1] 4
median(c(mean(x$Glasses[x$Age=='L']), mean(x$Glasses[x$Age=='M']), mean(x$Glasses[x$Age=='U'])))

## [1] 4.857143
mean(median(x$Glasses[x$Age=='L']), median(x$Glasses[x$Age=='M']), median(x$Glasses[x$Age=='U']))

## [1] 8.5
#####
#####

median(unlist(lapply(split(x$Glasses,x$Age),median)))

## [1] 4
mean(unlist(lapply(split(x$Glasses,x$Age),median)))

## [1] 5.166667
mean(unlist(lapply(split(x$Glasses,x$Age),mean)))

## [1] 13.2209
```

Question 3

Download one or more datasets from <http://www.maths.usyd.edu.au/u/UG/JM/StatsData.html> or elsewhere.

- a. Use numerical and graphical summaries to explore the datasets.
 - How many variables in the dataset?
 - How many observations?
 - What types of variables are measured?
 - Do any variables have interesting behaviours?
 - Outliers?
 - Skewed?
 - Multimodal?
 - Missing values?
 - Is there any possible interpretation for what you find?
- b. Try the following graphical summaries on your dataset. What are the advantages and disadvantages of the different plots?

```
# If needed explore the documentation for each function i.e.
?stripchart
# or
help('stripchart')

install.packages('vioplot') # Install the vioplot library
library('vioplot') # Load the vioplot library

plot(table(x))
barplot(table(x))
hist(x)
plot(density(x))
boxplot(x)
stripchart(x)
vioplot(x)
```

- c. Many of the plotting functions allow you to put two plots overlayed in the one window. Can you combine these with any of the plots above? Do these change any of your previous interpretations or your assessment of advantages and disadvantages?

```
points(density(x))
stripchart(x,add = TRUE, vertical = TRUE)
```

- d. Many functions have parameters that alter their behaviour. Do these alter your interpretations?

```
stripchart(list(x,x), vertical = TRUE, method = 'jitter',jitter = 1)
points(density(x),type = 'line')
hist(x, breaks = 100)
```

- e. Can you use any of these functions to compare two variables? Does this alter your findings?

Q2

The answers to these questions will depend greatly on the datasets you look at. A general, but non-exhaustive list of observations may include.

- `plot(table(x))` and `barplot(table(x))` both show the same information but one has thicker bars. Thicker bars can be more striking when there are a few categories. If you have a lot of categories, especially

with ordinal or discrete data, the thicker bars may make it tougher to read off specific categories. I generally just use `barplot`.

- `hist(x)` and `boxplot(x)` show very similar information. The histogram allows you to see how many data points you have whereas you don't see this in the boxplot. The boxplot makes it very easy to read off the median, IQR and spot potential outliers. It is also much nicer to put two boxplots side by side, while you can do this with histograms it is often harder to compare. This difficulty gets exaggerated when comparing lots and lots of plots. You also cannot see bimodal data with a boxplot
- A stripchart has a similar layout to the boxplot but you trade-off not being able to immediately visualise the summary statistics for being able to see how much data you have. A stripchart is useless once you have lots of observations (using jitter can help this but this is for your own interest only). Overlaying a stripchart with a boxplot can often give you the best of both worlds
- `plot(density(x))` displays very similar information to a probability histogram. This is essentially a histogram with a smoothing applied. As it plots lines instead of boxes it can allow you to overlay lots of densities (smoothed boxplots for lots of variables) and make them comparable. Unfortunately, choosing the way that you smooth is very arbitrary and you lose the ability to see how many samples you have.
- A violin plot is an extension of the boxplot that combines with density information. This allows you to see multi-modal behaviour with a boxplot. You still can't see how many data points are in the plot though... This could however be combined with a stripchart to get the best of everything!