MM916 Week 3 Question Sheet

Question 1

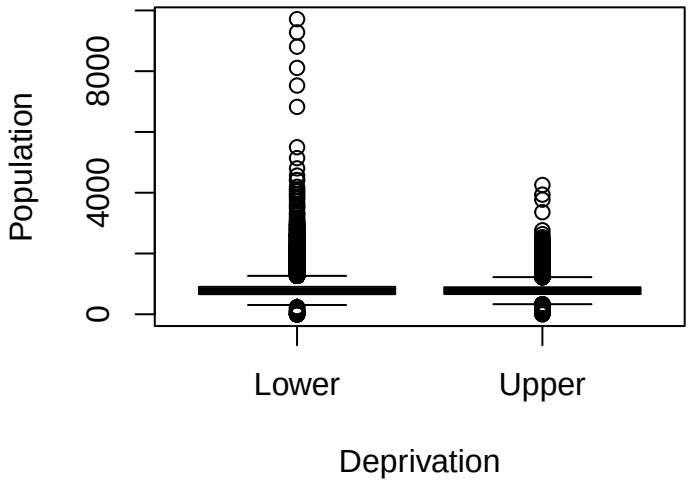
The data set `population_deprivation.txt` contains the population in each data zone in Scotland between 2001 and 2013. A data zone is a small region created for data analysis purposes. It also contains information on whether the data zone is in one of the 50% most or least deprived regions.

a) Download the data `population_deprivation.txt` and read this into R, tidying it as in last week's question sheet.

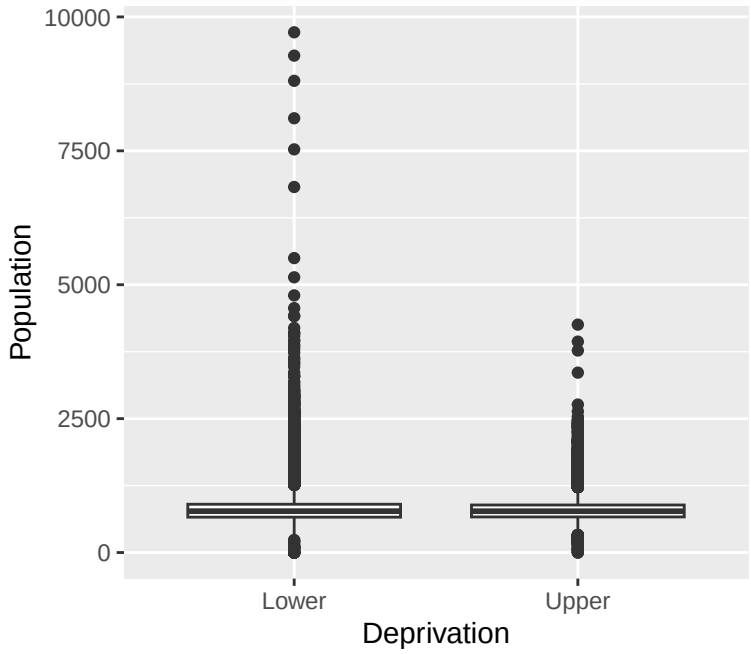
```
population <- read.table("population_deprivation.txt", header=TRUE)

# need to tidy this
population %>%
  pivot_longer(starts_with("Y"), names_to="Year",
               values_to="Population", names_prefix="Y") ->
  population
```

b) What differences can you observe in the populations between the 50% most and least deprived regions? Use boxplots. (Transforming to a log scale would probably be helpful: see if a bit of experimenting or reading of cheatsheets shows you how to do this, but don't get hung up on it.)

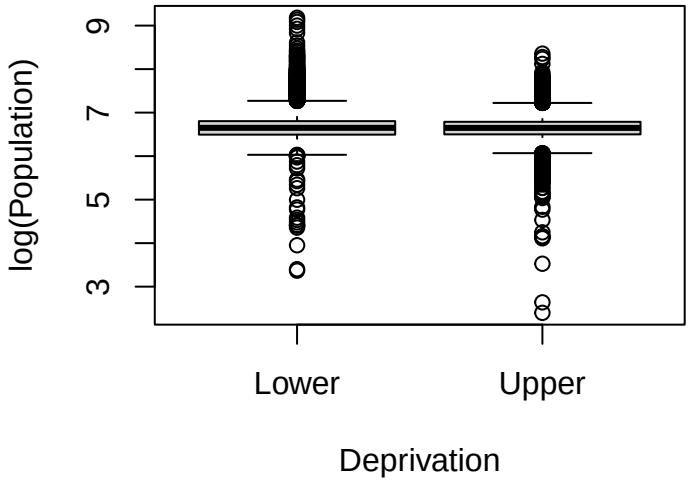
```
# Goal is to explore differences that depend on Deprivation. The only variable that would
# be sensible to plot as a function of Deprivation is Population. So the question becomes:
# Make a boxplot of Population, divided into categories using Deprivation.

# You can use base R
boxplot(Population~Deprivation, data=population)
```



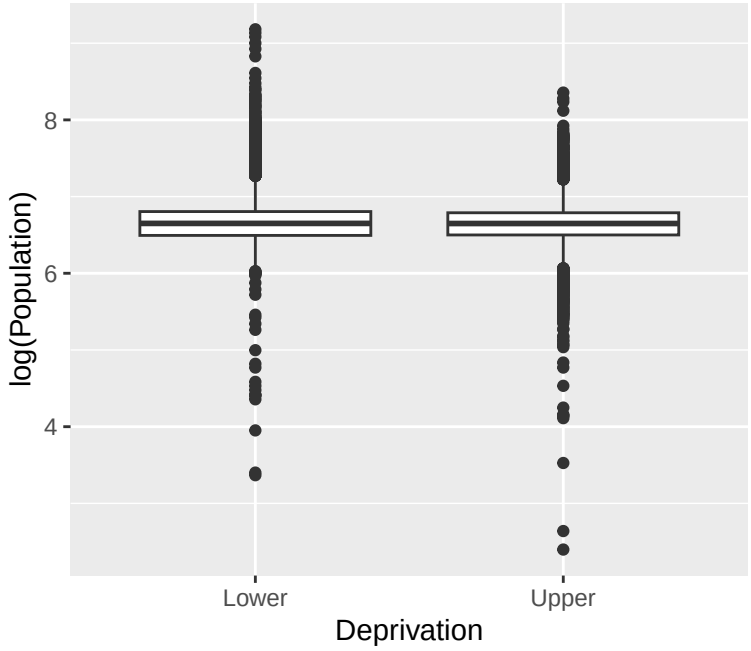

```
# Or you can use ggplot:

ggplot(population) +
  geom_boxplot(aes(x=Deprivation, y=Population))
```



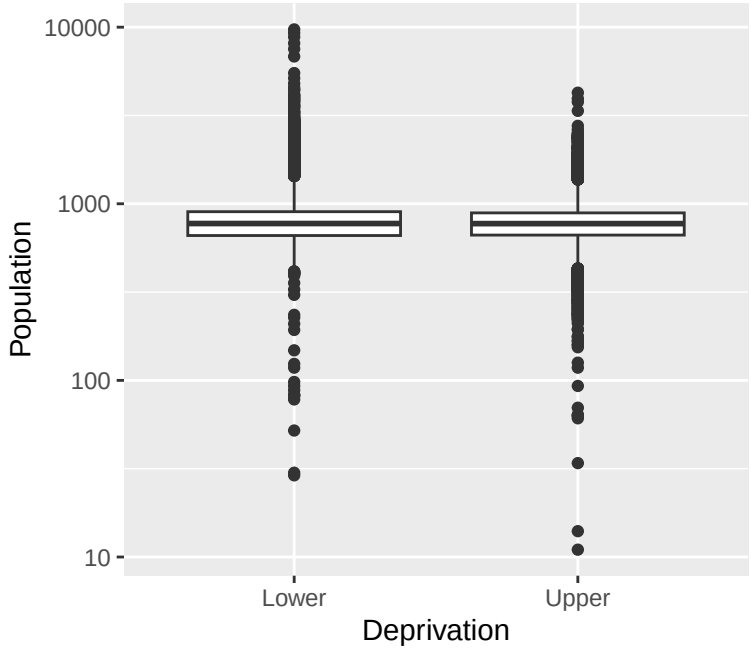

```
# In this case it might be appropriate to take a log due to the data being
# skewed. You can do this without changing the data:

# You can use base R
boxplot(log(Population)~Deprivation, data=population)
```




```
# Or you can use ggplot:
ggplot(population) +
  geom_boxplot(aes(x=Deprivation, y=log(Population)))
```

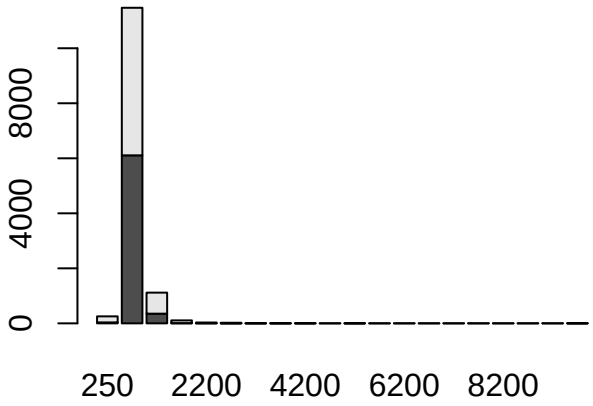



```
# another way to do this, which gives nicer labelling, is
ggplot(population) +
  geom_boxplot(aes(x=Deprivation, y=Population)) +
  scale_y_log10()
```

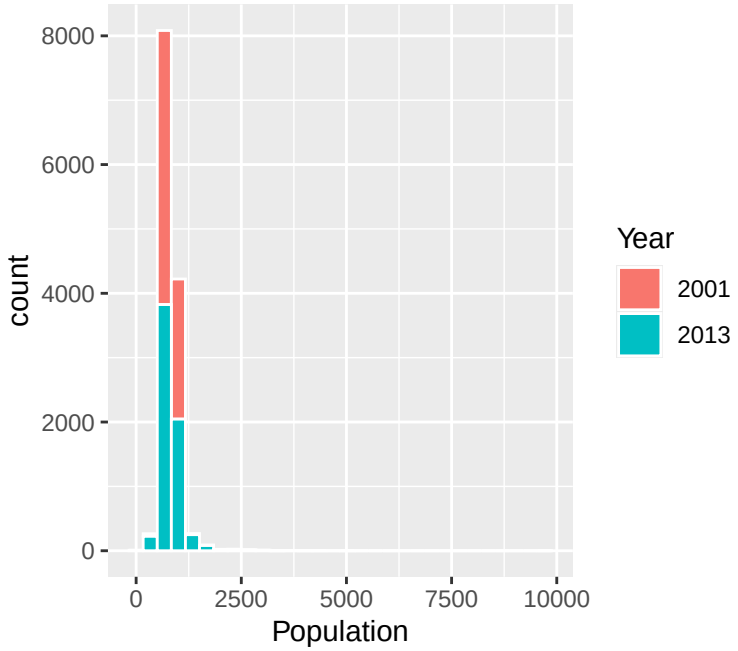


c) Compare the distribution of the populations in 2001 and 2013 using histograms.

```
# You can use base R and the plotrix package
require(plotrix)
multhist(split(population$Population[population$Year%in%c(2001, 2013)],
               population$Year[population$Year%in%c(2001, 2013)]), beside=FALSE)
```

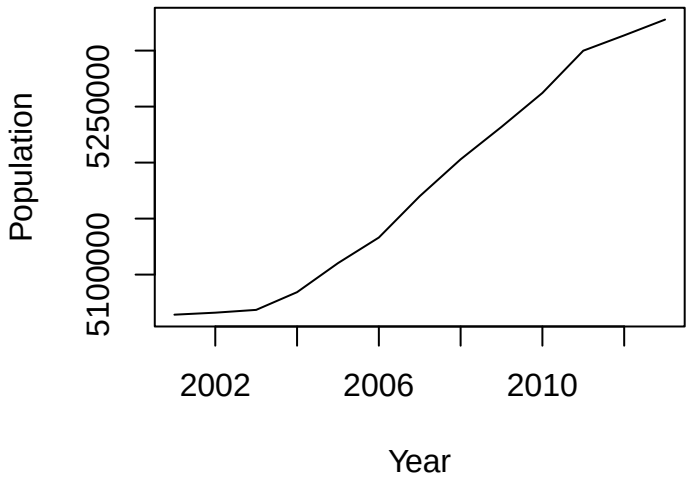



```
# Or you can use ggplot (simpler in this case):
population %>%
  filter(Year%in%c(2001, 2013)) %>%
  ggplot(aes(Population, fill=Year)) +
  geom_histogram(colour="white")
```

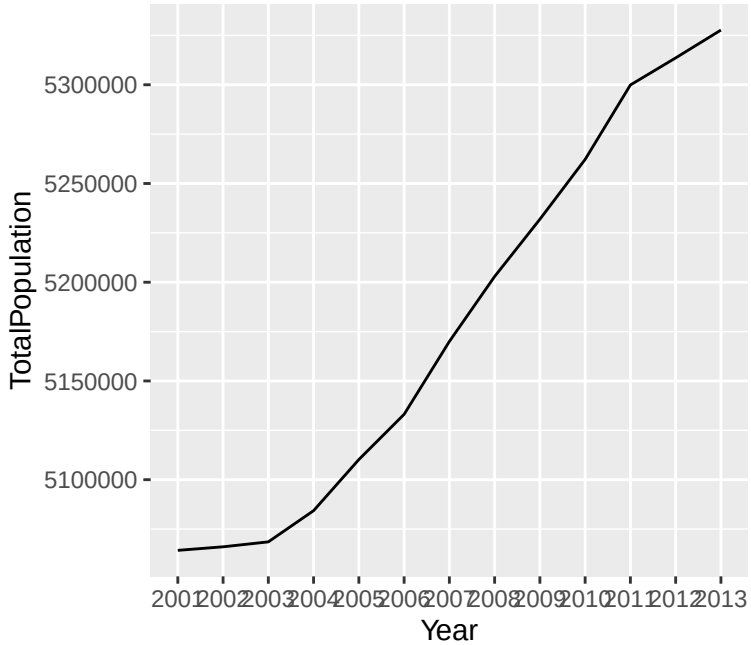


d) Produce a line or scatter plot to visualise the overall change in the population through time.

```
# You can use base R

to_plot <- aggregate(population$Population, list(Year=population$Year), sum)
plot(x~Year, data=to_plot, type="l", ylab="Population")
```



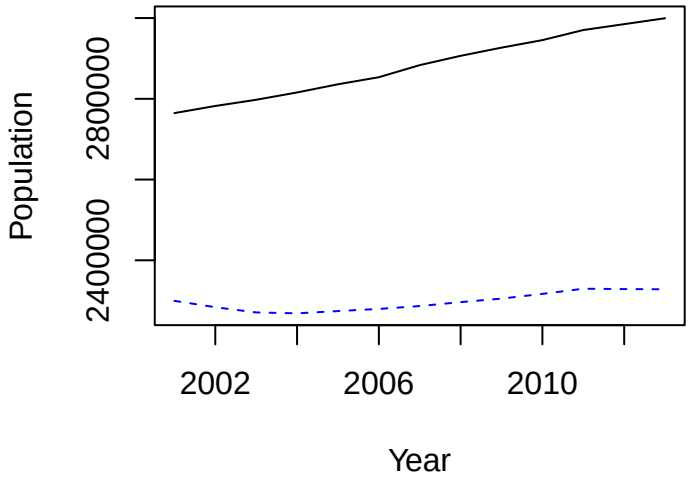

```
# Or you can use tidyverse+ggplot.
# use summarise() and group_by() to get the total population by year
totPop = population %>%
  group_by(Year) %>%
  summarise(TotalPopulation = sum(Population)) %>% ungroup()
# now make the line plot
ggplot(totPop) +
  geom_line(aes(Year, TotalPopulation, group=1))
```



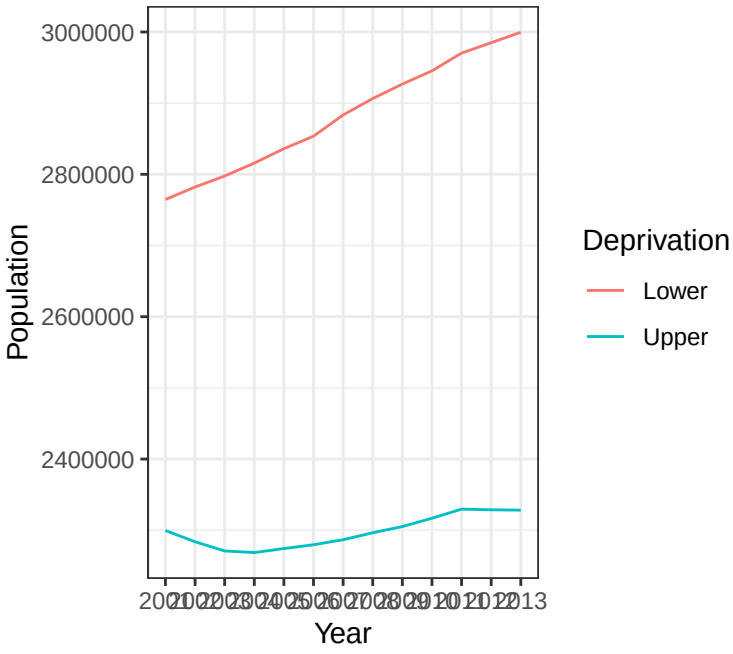

```
# if you try to leave off the group=1, you either get the error
#    geom_path: Each group consists of only one observation. Do you need to adjust
#    the group aesthetic?
# or a blank plot. The reference to "group aesthetic" is a sign that you need to
# specify group=[something] in geom_line. group=1 means "connect all the values
# with a single line". Compare with # part e, where group= does more useful work.
```

e) Produce an appropriate plot to find out whether the pattern through time is the same for those in the 50% most deprived areas compared to the 50% least deprived areas.

```r
# You can use base R

to_plot <- aggregate(population$Population,
                     list(Year=population$Year, Deprivation=population$Deprivation),
                     sum)

# Create a plot using one of the series (use subset to specify which)
# Need to adjust the y axis to allow for the other series to be plotted
plot(x~Year, data=to_plot, type="l", ylab="Population", subset=Deprivation=="Lower",
     ylim=c(min(to_plot$x), max(to_plot$x)))
# Add a line
lines(x~Year, data=to_plot, col="blue", lty=2, subset=Deprivation=="Upper")
```




```r
# Or you can use ggplot:

population %>%
  group_by(Year, Deprivation) %>%
  summarise(Population=sum(Population)) %>%
  ungroup() %>%
  ggplot(aes(Year, Population, group=Deprivation, colour=Deprivation)) +
  geom_line() +
  theme_bw()
```



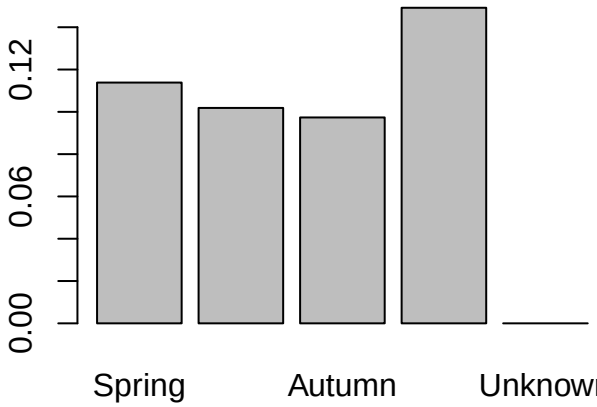
Question 2

`expeditions.RData` contains data on mountaineering expeditions in the Himalayas. Information on the contents of these tables can be found at https://github.com/rfordatascience/tidytuesday/blob/master/data/2020/2020-09-22/readme.md.

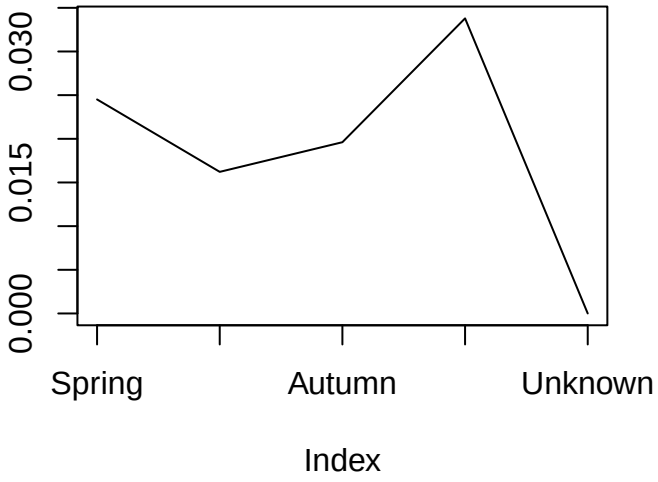
```
load("expeditions.RData")
```

a) Make a plot to visualise the seasonal variation in the number of deaths and the number of injuries. (It's up to you to figure out what plot type makes the most sense.)

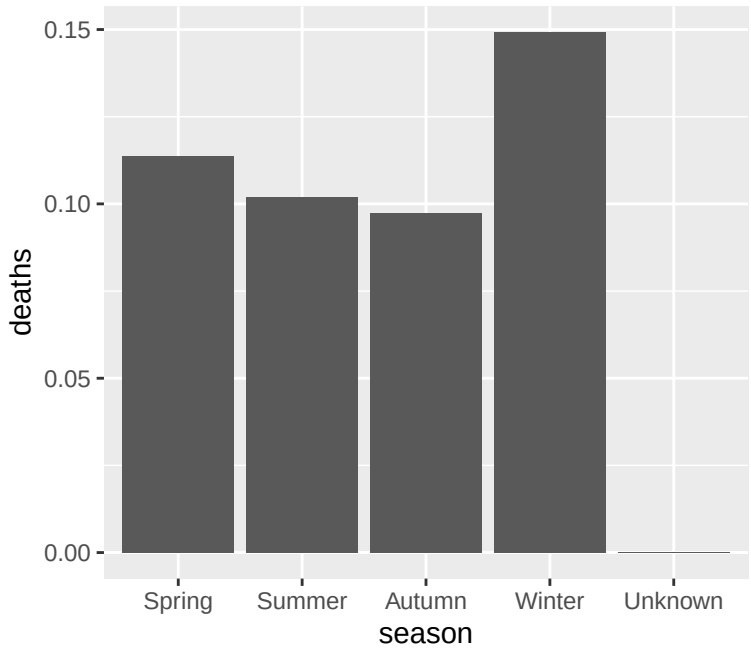
```
## Either a bar chart or a line graph would work here

# option 1: barplot in Base R (deaths data as an example)

# reorder seasons. All the examples below include a line like this; the plots will work
# without it, but you'll end up with alphabetical order and "unknown" mixed into the actual
# seasons.
expeditions$season <-
  factor(expeditions$season, levels=c("Spring","Summer","Autumn","Winter","Unknown"))

barplot(tapply(expeditions$member_deaths+expeditions$hired_staff_deaths,
               list(season=expeditions$season), mean))
```




```r
# option 2: line plot in base R (injuries data as an example)

# get injuries from the climbers data set
climbers$season <-
  factor(climbers$season, levels=c("Spring","Summer","Autumn","Winter","Unknown"))
plot(tapply(climbers$injured, list(season=climbers$season), mean),
     type='l', xaxt="n", ylab="")
axis(side=1, labels=levels(climbers$season), at=1:5)
```



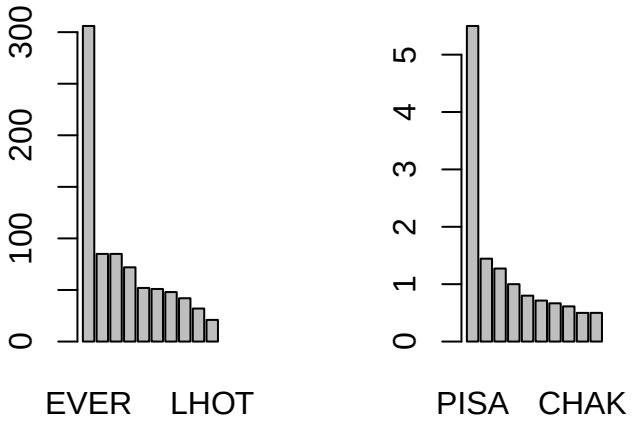

```r
## option 3: bar plot in ggplot (deaths data as an example)

expeditions %>%
  mutate(season=factor(season, levels=c("Spring","Summer","Autumn","Winter","Unknown")),
         deaths = member_deaths+hired_staff_deaths) %>%
  group_by(season) %>%
  summarise(deaths=mean(deaths)) %>%
  ggplot() +
  geom_col(aes(season, deaths))
```




```
# option 4: line or scatter plot in ggplot. As in option 3, but replace
# geom_col with geom_line or geom_point. For unknown reasons, you also need to specify
# group=1 in aes (as in Q2d). geom_col doesn't seem to require this.
```

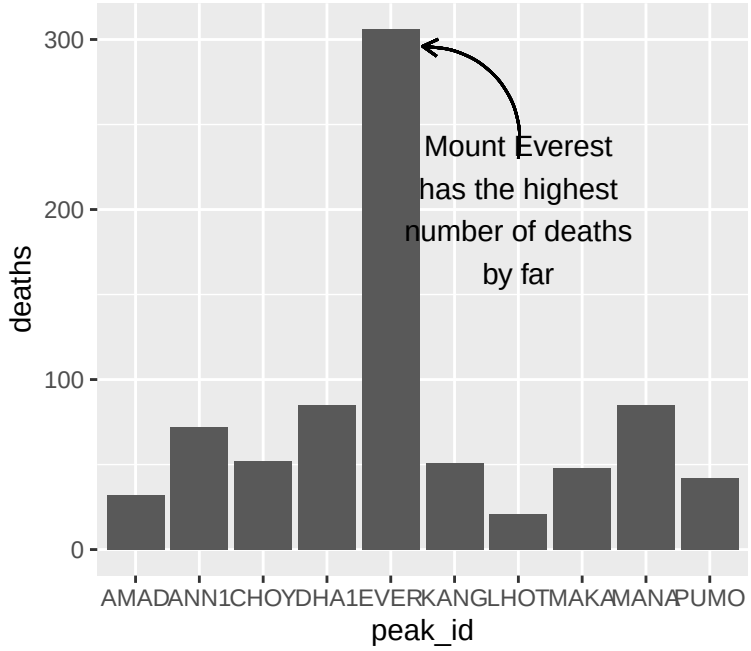
b) Which peaks have had the most deaths?

```
par(mfrow=c(1,2))
barplot(sort(tapply(expeditions$member_deaths+expeditions$hired_staff_deaths,
                    list(peak_id=expeditions$peak_id), sum), decreasing=TRUE)[1:10])

barplot(sort(tapply(expeditions$member_deaths+expeditions$hired_staff_deaths,
                    list(peak_id=expeditions$peak_id), mean), decreasing=TRUE)[1:10])
```

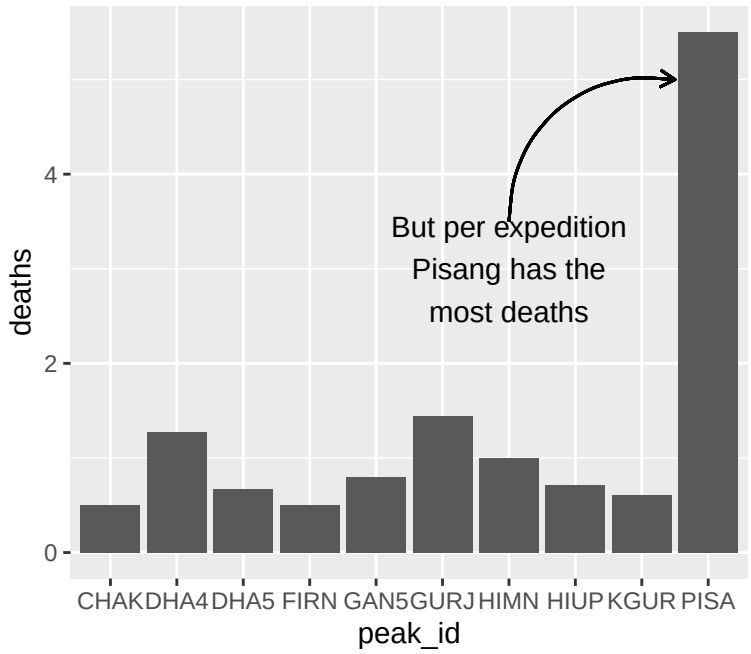



```
## ggplot
expeditions %>%
  group_by(peak_id) %>%
  summarise(deaths=sum(member_deaths+hired_staff_deaths)) %>%
  arrange(desc(deaths)) %>%
  slice_head(n=10) %>%
  ggplot() +
  geom_col(aes(peak_id, deaths)) +
```

```
  annotate("text", x="LHOT", y=200,
           label="Mount Everest\nhas the highest\nnumber of deaths\nby far") +
  geom_curve(
  aes(x ="LHOT", y = 230, xend = 5.5, yend = max(deaths)-10),
  arrow = arrow(length = unit(0.03, "npc"))
)  ->p1

expeditions %>%
  group_by(peak_id) %>%
  summarise(deaths=mean(member_deaths+hired_staff_deaths)) %>%
  arrange(desc(deaths)) %>%
  slice_head(n=10) %>%
  ggplot() +
  geom_col(aes(peak_id, deaths))+
  annotate("text", x="HIMN", y=3, label="But per expedition\nPisang has the\nmost deaths")+
  geom_curve(
  aes(x ="HIMN", y = 3.5, xend = 9.5, yend =5),
  curvature=-0.5,
  arrow = arrow(length = unit(0.03, "npc"))
) ->p2

p1
```

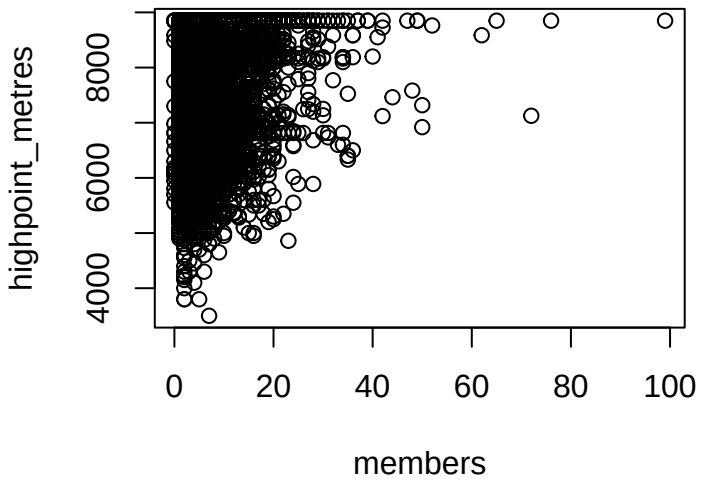



```
p2
```

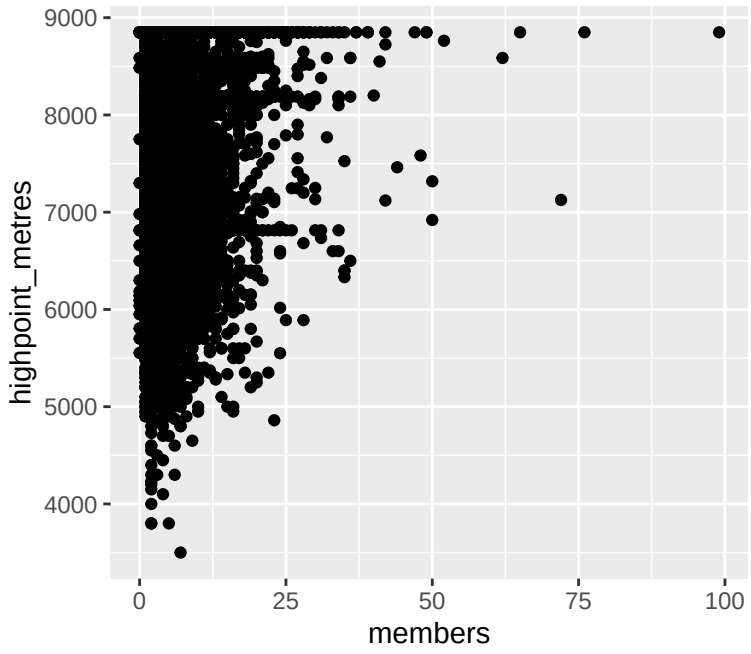


c) Is there a relationship between the number of members in a team and the highest point reached? Make a plot that lets you see every data point.

```
## base R

plot(highpoint_metres~members, data=expeditions)
```



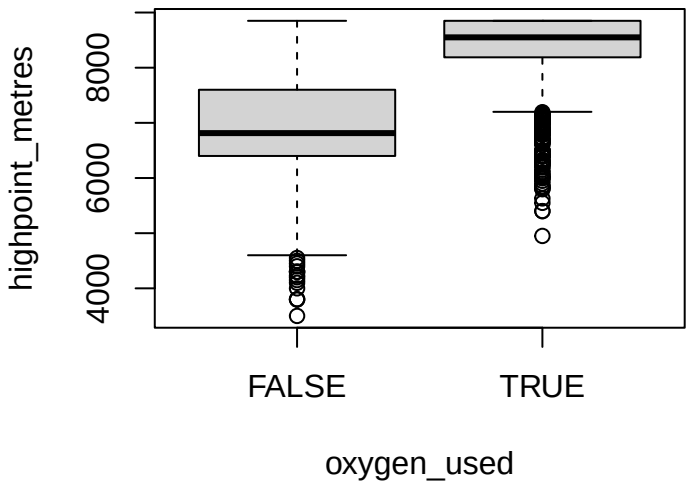

```
## ggplot
ggplot(expeditions) +
  geom_point(aes(members, highpoint_metres))
```



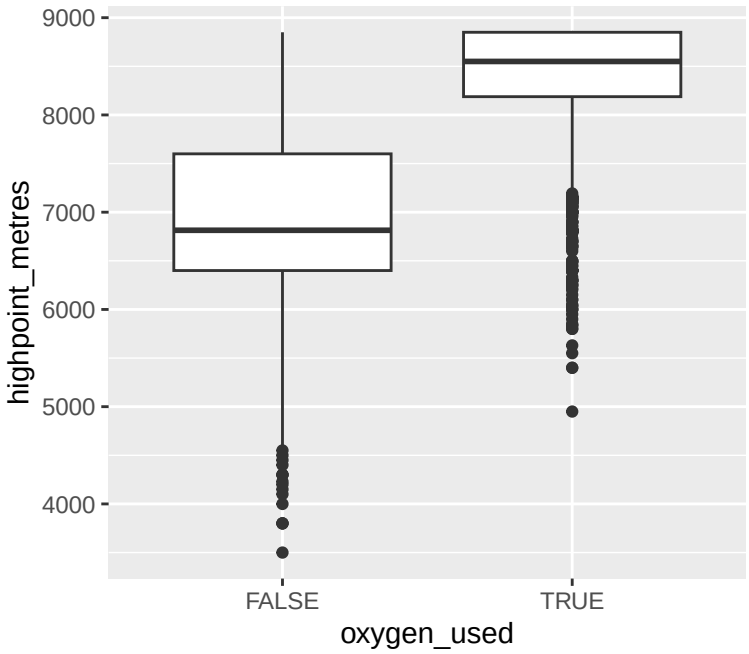

```
# Certainly seem to be more higher climbs for larger teams, but if you truncate
# the values to smaller teams there doesn't appear to be a relationship between
# members and highest point reached.`
```

d) Is oxygen use a sign of a good expedition or a bad expedition?

```
# If we use the high point reached to measure a good versus bad expedition:

# base R
boxplot(highpoint_metres ~ oxygen_used, data=expeditions)
```




```
# ggplot
ggplot(expeditions) +
  geom_boxplot(aes(oxygen_used, highpoint_metres))
```




```
# It does look like more successful climbs made use of oxygen.`
```

Extra practice

Question 3

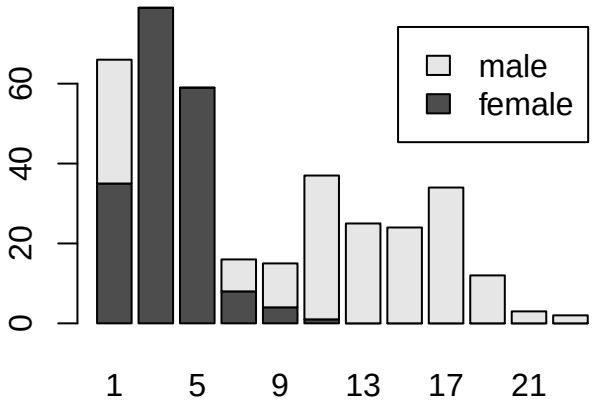
The data contained in `suicides.csv` contains the number of suicides and the population for males vs females at different age groups.

a) Download the file `suicides.csv` and read this into R

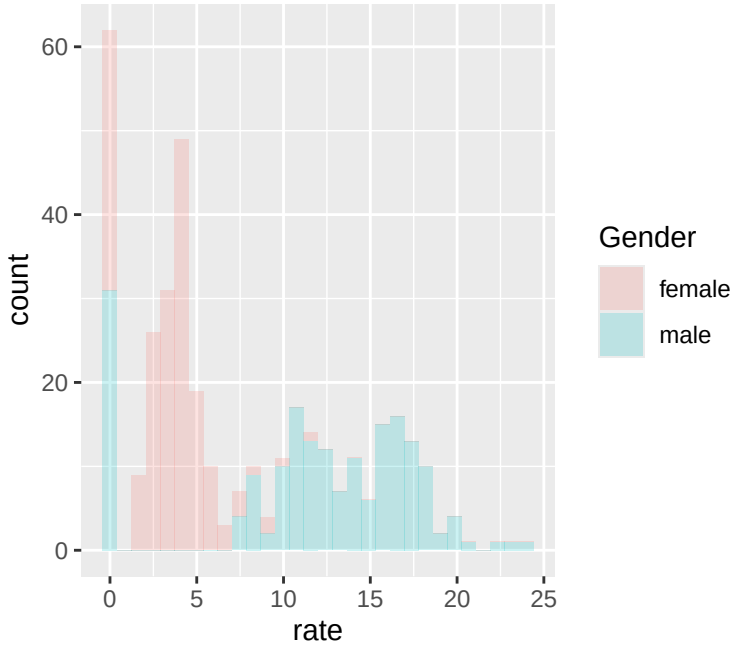
```
suicides <- read_csv("suicides.csv")

# Tidy
suicides %>%
  pivot_longer(4:ncol(suicides), names_to=c("Gender","Age"), names_sep=":",
                          values_to="Count") %>%
  pivot_wider(names_from=Type, values_from=Count) ->
  suicides
```

b) Produce a plot to compare the distribution of the rate of suicide per 100,000 for males and females.

```
# Using base R

suicides$rate <- suicides$suicides_no*100000/suicides$population

multhist(split(suicides$rate, suicides$Gender), beside=FALSE, legend.text=TRUE)
```

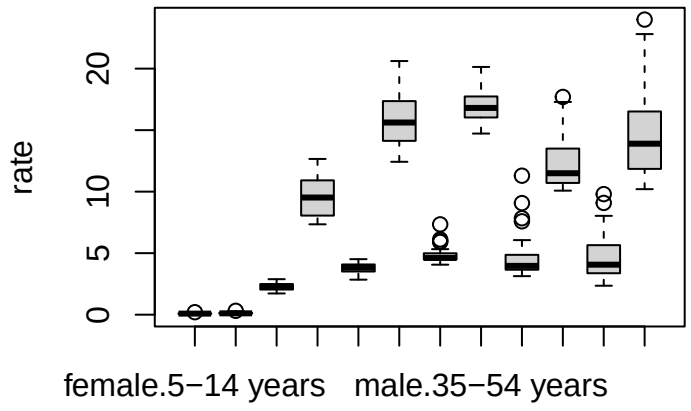



```
# Using ggplot

suicides %>%
  mutate(rate=suicides_no*100000/population) %>%
  ggplot() +
  geom_histogram(aes(rate, fill=Gender), alpha=0.2)
```

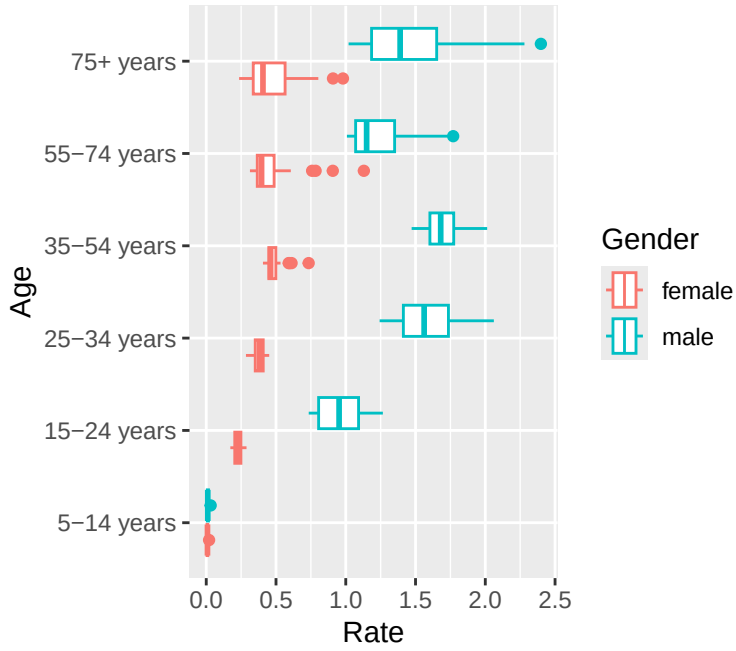


c) Produce a single plot that will compare the differences for each combination of gender and age. Make sure that the age groups are appropriately labelled.

```
# base R

suicides$Age <- factor(suicides$Age,
                       levels=paste(c("5-14","15-24","25-34","35-54","55-74","75+"),
                                    "years"))

boxplot(rate~Gender + Age, data=suicides)
```




```
# ggplot
suicides %>%
  # you would probably save this step
  mutate(Rate=suicides_no*10000/population,
         Age=factor(Age,
                    levels=paste(c("5-14","15-24","25-34","35-54","55-74","75+"),
                                 "years"))) %>%
  ggplot(aes(Rate, Age, colour=Gender)) +
  geom_boxplot()
```

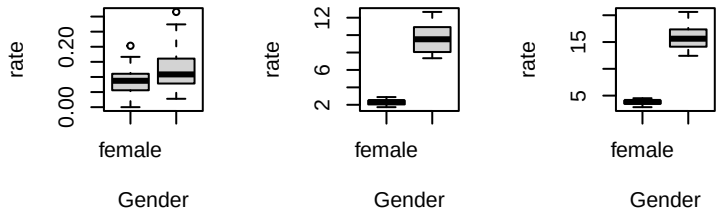


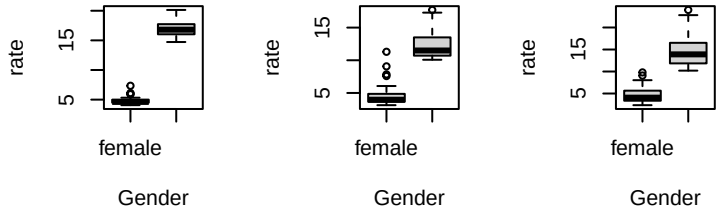

```
# You could put the gender/age labelling the other way around, but this is easier to read.
# You can see that the difference between males and females is largest in the 25-34 age
# group and then decreases slightly in the older age groups`
```

d) Produce separate plots comparing gender for each age group. Put these on one figure.

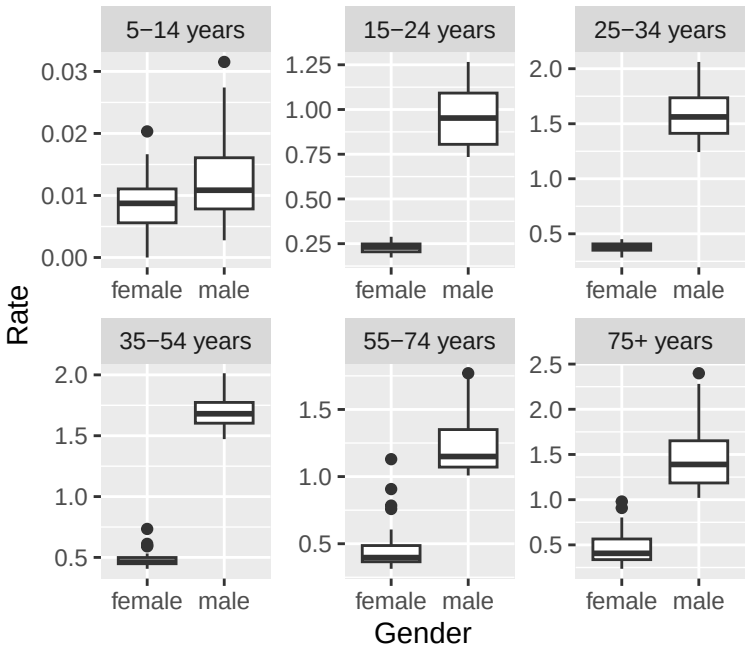
```
# base R

suicides$Age <- factor(suicides$Age,
```

```
                    levels=paste(c("5-14","15-24","25-34","35-54","55-74","75+"),
                                 "years"))
par(mfrow=c(2,3))
boxplot(rate~Gender, data=suicides, subset=Age=="5-14 years")
boxplot(rate~Gender, data=suicides, subset=Age=="15-24 years")
boxplot(rate~Gender, data=suicides, subset=Age=="25-34 years")
boxplot(rate~Gender, data=suicides, subset=Age=="35-54 years")
boxplot(rate~Gender, data=suicides, subset=Age=="55-74 years")
boxplot(rate~Gender, data=suicides, subset=Age=="75+ years")
```






```
# ggplot
suicides %>%
  # you would probably save this step
  mutate(Rate=suicides_no*10000/population,
         Age=factor(Age,
                    levels=paste(c("5-14","15-24","25-34","35-54","55-74","75+"),
                                 "years"))) %>%
  ggplot(aes(Gender, Rate)) +
  geom_boxplot() +
  # set scale="free" to allow the y scale to vary
  # this is more equivalent to the base R implementation
  facet_wrap("Age", scale="free")
```



Question 4

Load the data stored in `friends.RData`. This contains information on the TV series Friends, more information about the data can be found at https://github.com/rfordatascience/tidytuesday/blob/master/data/2020/2020-09-08/readme.md. From this data set you should produce visualisations which help to determine:

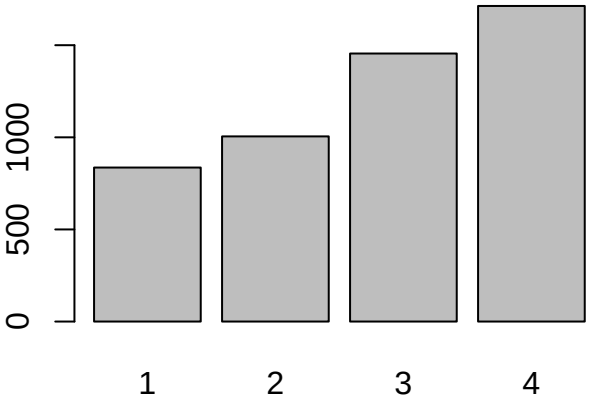
```
load("friends.RData")
```

a) Which season was the most positive?

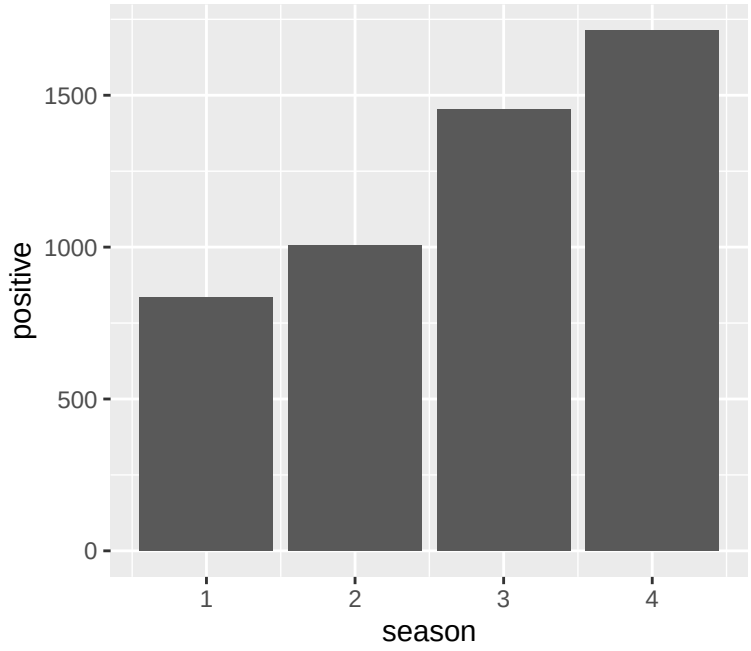
```
# look at the unique emotions
unique(friends_emotions$emotion)
```

```
## [1] "Mad"      "Neutral"  "Joyful"   "Scared"   "Sad"      "Powerful" "Peaceful"
```

```
## Base R
friends_emotions$positive <-
  ifelse(friends_emotions$emotion%in%c("Peaceful","Joyful","Powerful"), 1, 0)

barplot(tapply(friends_emotions$positive, list(friends_emotions$season), sum))
```

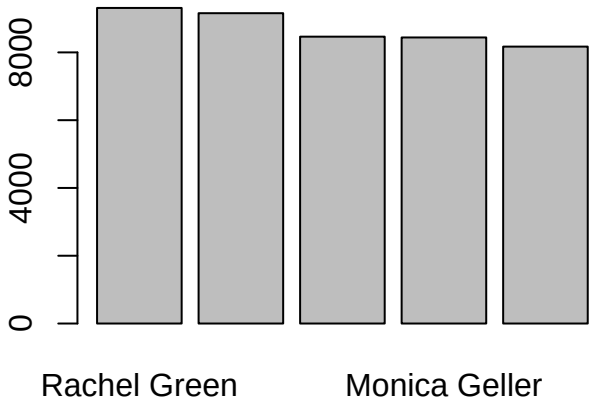


```r
## ggplot
friends_emotions %>%
  mutate(positive=ifelse(emotion%in%c("Peaceful","Joyful","Powerful"), 1, 0)) %>%
  group_by(season) %>%
  summarise(positive=sum(positive)) %>%
  ggplot() +
  geom_col(aes(season, positive))
```

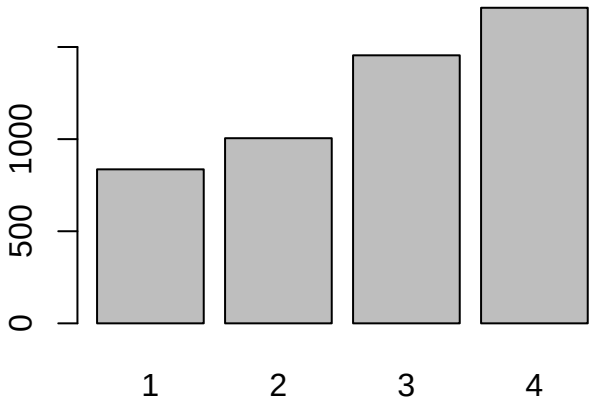


c) Which character spoke most overall?

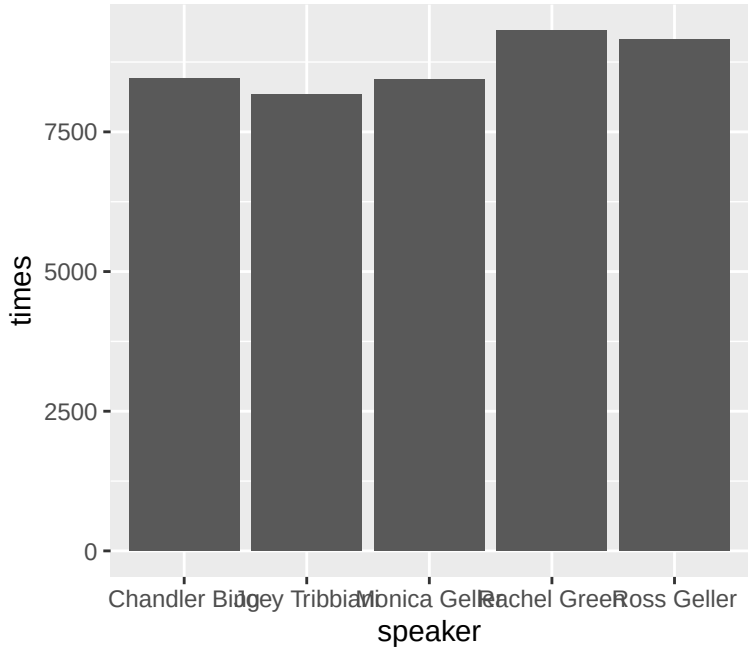
```r
# base R
barplot(sort(tapply(rep(1, nrow(friends)), list(friends$speaker), sum),
             decreasing=TRUE)[1:5])
```




```r
barplot(tapply(friends_emotions$positive, list(friends_emotions$season), sum))
```



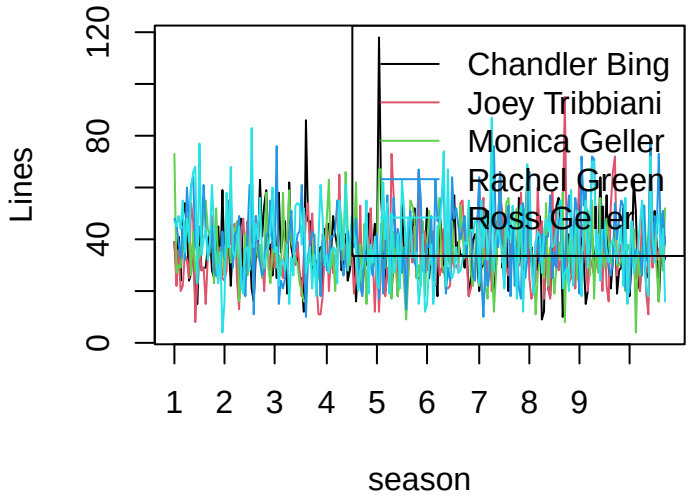

```
# ggplot
friends %>%
  group_by(speaker) %>%
  summarise(times=n()) %>%
  arrange(desc(times)) %>%
  # this extracts the top 5 speakers
  slice_head(n=5) %>%
  ggplot() +
  geom_col(aes(speaker, times))
```



d) Did the amount of speech change for each character as time went on?

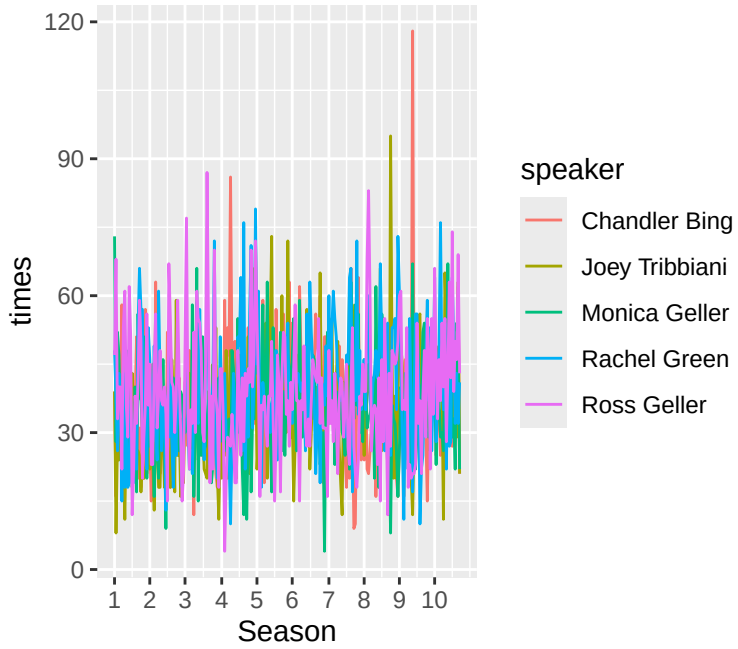
```
## base R

to_plot <- friends[friends$speaker%in%c("Chandler Bing","Joey Tribbiani",
                      "Monica Geller","Rachel Green","Ross Geller"),]

to_plot <- aggregate(rep(1, nrow(to_plot)),
                     list(speaker=to_plot$speaker,
                          season=to_plot$season,
                          episode=to_plot$episode), sum)
```

```
# set up blank plot using type='n'
plot(to_plot$x, type='n',
     # set the x limits - number of episodes in total
     xlim=c(1, nrow(unique(to_plot[,c("season","episode")]))),
     # remove the x-axis
     xaxt="n", xlab="season", ylab="Lines")
axis(side=1, at=cumsum(c(1,aggregate(episode~season, data=to_plot, max)$episode[-10])),
     labels=1:10)
col <- 1

for(i in unique(to_plot$speaker)){
  lines(to_plot$x[to_plot$speaker==i],col=col)
  col <- col+1
}

# add a legend

legend("topright", col=1:length(unique(to_plot$speaker)),
       legend=unique(to_plot$speaker), lty=1)
```

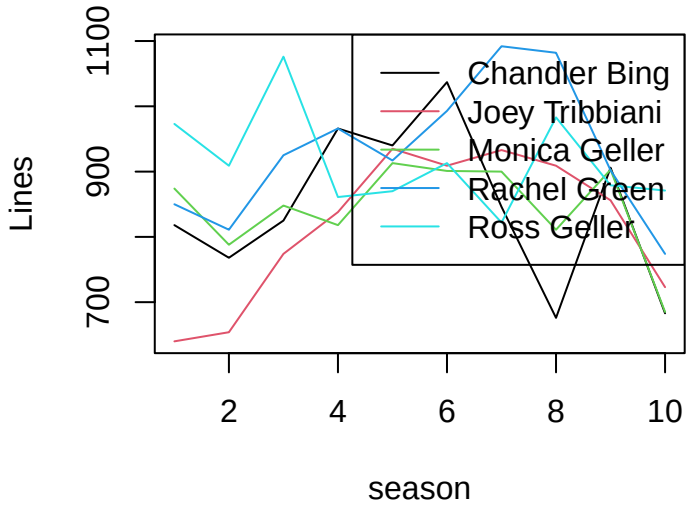



```
## ggplot

friends %>%
  filter(speaker%in%c("Chandler Bing","Joey Tribbiani",
                      "Monica Geller","Rachel Green","Ross Geller")) %>%
  group_by(speaker, season, episode) %>%
  summarise(times=n()) %>%
  arrange(season, episode) %>%
  ungroup() %>%
  group_by(speaker) %>%
  mutate(id=row_number()) -> to_plot

  # now have a continuous variable _id_ that runs 1..236 for episode number
  # across seasons. For plotting, we want to know where each season begins
  ep_count = friends %>% group_by(season) %>% summarise(num_ep = max(episode))
  season_breaks = cumsum(c(1, ep_count$num_ep[-10]))
```

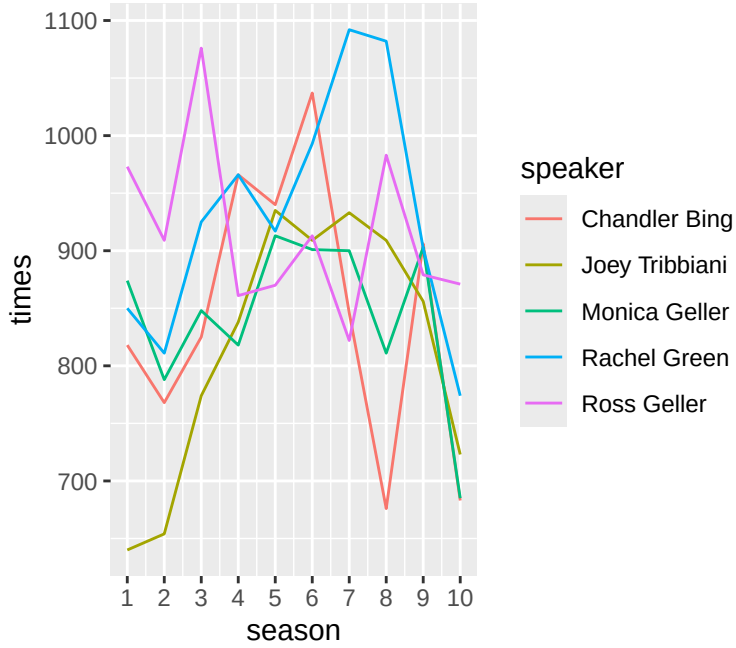
```
  ggplot(to_plot) +
  geom_line(aes(id, times, group=speaker, colour=speaker))+
  scale_x_continuous(
    breaks=season_breaks, labels=1:10, name="Season")
```




```
# The plot above is difficult to read - try looking at it by season:`

# base R
to_plot <- friends[friends$speaker%in%c("Chandler Bing","Joey Tribbiani",
                      "Monica Geller","Rachel Green","Ross Geller"),]

to_plot <- aggregate(rep(1, nrow(to_plot)),
                     list(speaker=to_plot$speaker,
                          season=to_plot$season), sum)

# set up blank plot using type='n'
plot(x~season, data=to_plot, type='n', ylab="Lines")

col <- 1

for(i in unique(to_plot$speaker)){
  lines(to_plot$x[to_plot$speaker==i],col=col)
  col <- col+1
}

legend("topright", col=1:length(unique(to_plot$speaker)),
       legend=unique(to_plot$speaker), lty=1)
```




```
## ggplot
friends %>%
  filter(speaker%in%c("Chandler Bing","Joey Tribbiani",
                      "Monica Geller","Rachel Green","Ross Geller")) %>%
  group_by(speaker, season) %>%
  summarise(times=n()) %>%
  arrange(season) %>%
  ggplot() +
  geom_line(aes(season, times, group=speaker, colour=speaker)) +
  # make the x-axis more appropriate
  scale_x_continuous(breaks=1:10)
```




```
# Joey started off with relatively few lines but seems to have increased
# over the series up until series 7, then lines decreased again. Rachel
# had a big increase in lines in seasons 7 and 8 and then decreased again.
# everyone else remained roughly constant.`
```

Question 5

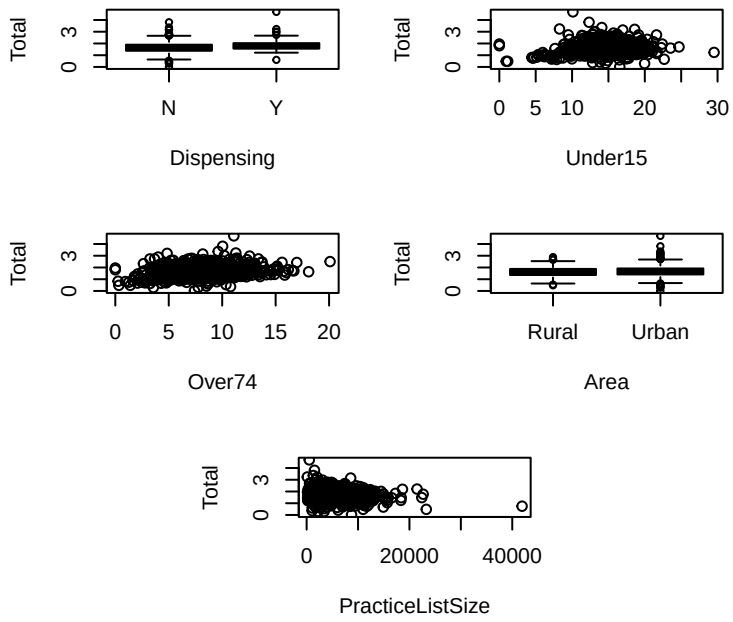
Load the data stored in `prescribing.csv`. The data description is shown below:

Column	Class	Description
PracticeCode	Factor	Unique GP practice code identifier
Total	Numeric	Total antibiotic items prescribed per 1000 registered patients
PracticeName	Factor	GP Practice name
PracticeListSize	Numeric	Number of registered patients at the GP practice
Postcode	Factor	GP Practice postcode
Dispensing	Factor	Can the GP dispense medication withint he practice
Under15	Numeric	% of GP practice patients under the age of 15
Over74	Numeric	% of GP practice patients under the age of 74
HBCode	Factor	Health Board Code
Area	Factor	Whether the Practice is in an urban or rural area

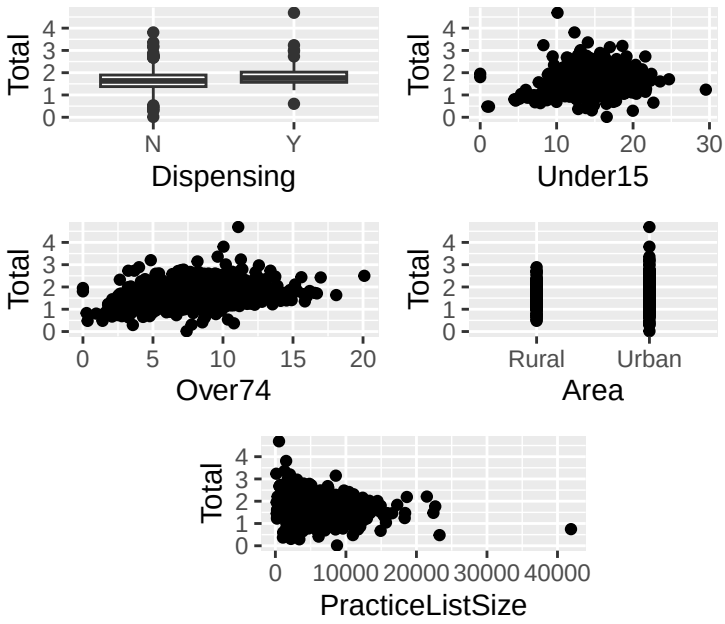
We are interested in:

a) The relationship between the prescribing rate and each of the demographic variables (put these on one plot)

```
prescribing <- read.csv("prescribing.csv")
# Demographic variables are: Dispensing, Under15, Over74, Area, PracticeListSize`

# base R

# set better margins...
par(mar=c(4.5,4,2,2))
# tip for getting nicely laid out plots in base R
# Use layout
layout(matrix(c(1,1,2,2,3,3,4,4,0,5,5,0), nrow=3, byrow=TRUE))
boxplot(Total~Dispensing, data=prescribing)
plot(Total~Under15, data=prescribing)
plot(Total~Over74, data=prescribing)
boxplot(Total~Area, data=prescribing)
plot(Total~PracticeListSize, data=prescribing)
```

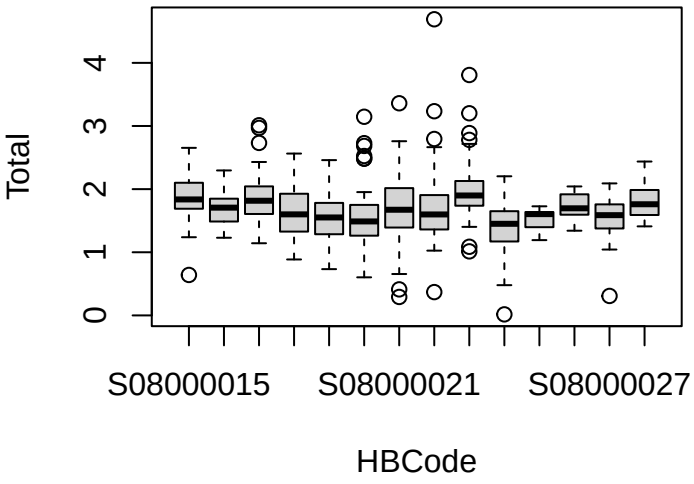



```
# ggplot
p1 <- ggplot(prescribing) +
        geom_boxplot(aes(Dispensing, Total))
p2 <- ggplot(prescribing) +
        geom_point(aes(Under15,Total))
p3 <- ggplot(prescribing) +
        geom_point(aes(Over74,Total))
p4 <- ggplot(prescribing) +
        geom_point(aes(Area, Total))
p5 <- ggplot(prescribing) +
        geom_point(aes(PracticeListSize,Total))

# Use patchwork to do the same thing
library(patchwork)
layout <- "
AABB
CCDD
#EE#
"
p1+p2+p3+p4+p5+plot_layout(design=layout)
```

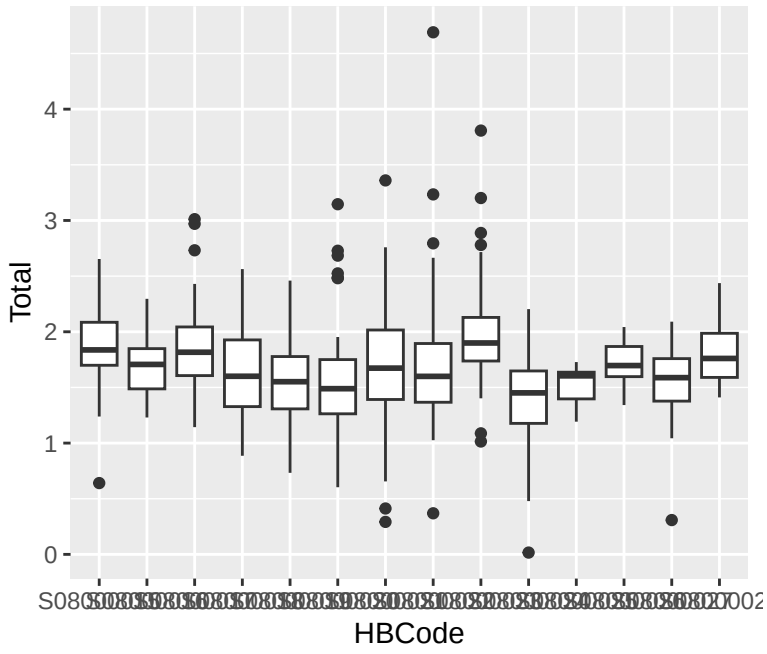


b) Are there differences in prescribing rate by health board?

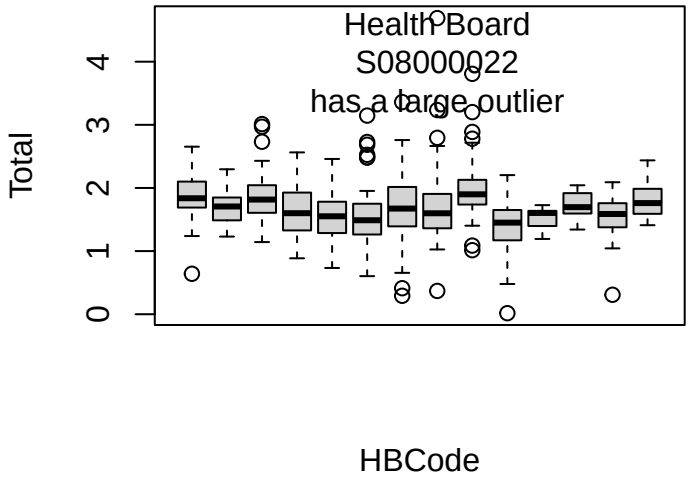
```
## base R
boxplot(Total~HBCode, data=prescribing)
```



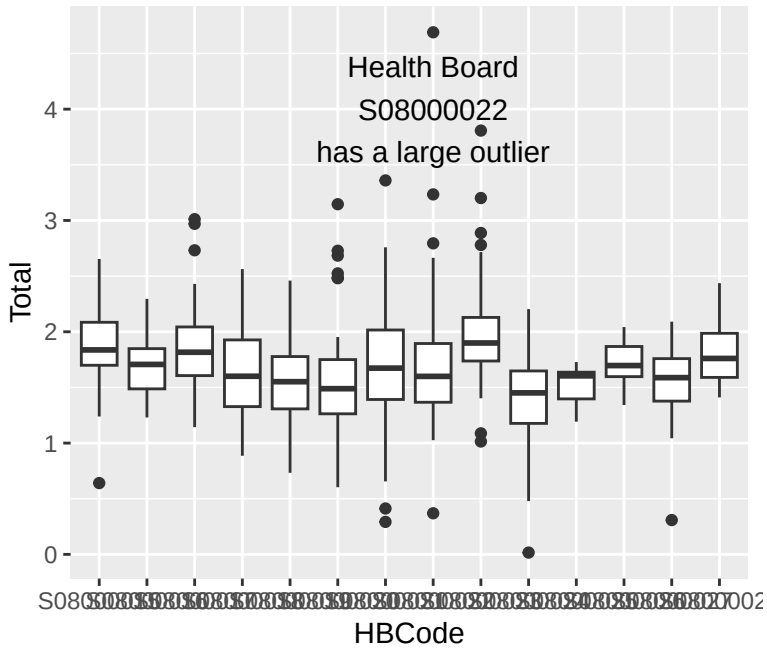

```
## ggplot
ggplot(prescribing) +
  geom_boxplot(aes(HBCode, Total))
```



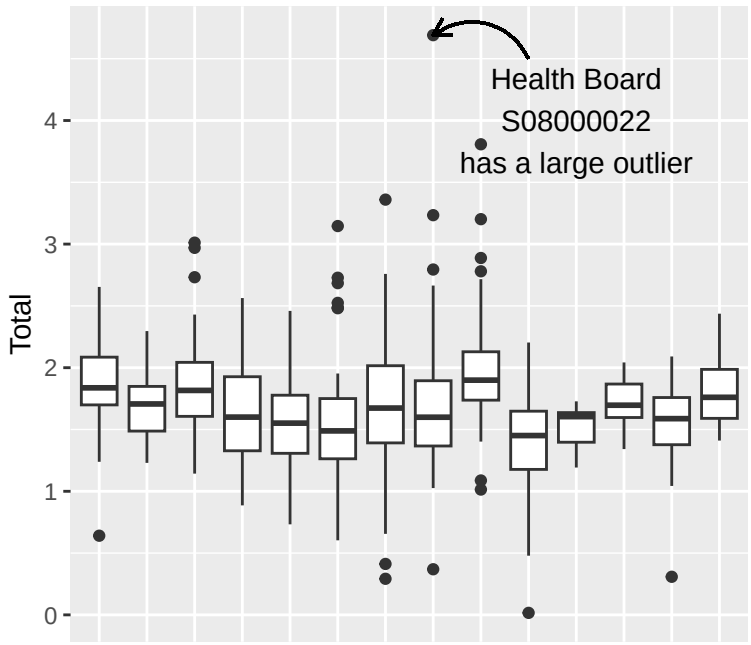

```
# You might consider removing the x-axis here and just labelling individual boxes of interest`

# Find the values of interest (we've gone with the largest outlier)

label <- prescribing$HBCode[which.max(prescribing$Total)]

## base R
boxplot(Total~HBCode, data=prescribing, xaxt="n")
## Getting this label in the right place takes a bit of trial and error!
text(x= which(sort(unique(prescribing$HBCode))==label), y=4,
     # \n means new line
     labels=paste("Health Board\n", label, "\nhas a large outlier"), adj=0.5)
```




```
## ggplot
ggplot(prescribing) +
  geom_boxplot(aes(HBCode, Total)) +
  annotate("text", x=label, y= 4, label=paste("Health Board\n", label, "\nhas a large outlier"))
```




```
## Or move the text away and add a nice curved arrow
ggplot(prescribing) +
  geom_boxplot(aes(HBCode, Total)) +
  annotate("text", x=11, y= 4, label=paste("Health Board\n", label, "\nhas a large outlier")) +
  geom_curve(
  aes(x = 10, y = 4.5, xend = label, yend = max(Total)),
  data = prescribing,
  arrow = arrow(length = unit(0.03, "npc"))
)+
  # remove x-axis labels
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```


Health Board
S08000022
has a large outlier
Total
4
3
2
1
0