Basic R and Exploratory Analysis cribsheet

In the book Introductory R, read through the chapters entitled “What is R?”, “A first R session”, “Basics” and “Using the help files”. If you aren’t familiar with command line interfaces or the use of vectors and matrices, work through the exercises in the ‘Basics” chapter. If you’re happy that you’re on top of this, try the following more advanced exercises.

The effect of sample size on frequency distributions

Use the function `rnorm()` to do the following (type `?rnorm()` to get the help file):

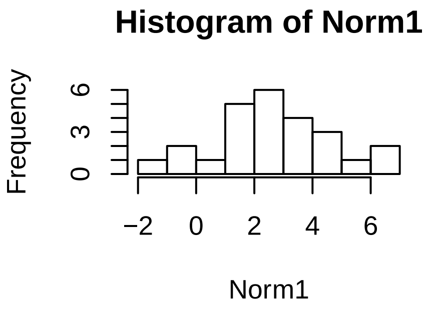
1. Create an object called “Norm1” with 25 normally distributed random numbers with mean 2 and standard deviation 2.
2. Create an object called “Norm2” with 50 normally distributed random numbers with mean 2 and standard deviation 2.
3. Create an object called “Norm3” with 100 normally distributed random numbers with mean 2 and standard deviation 2.
4. Create an object called “Norm4” with 500 normally distributed random numbers with mean 2 and standard deviation 2.
5. Set the parameter “mfrow” so that R will draw a 2x2 grid of graphs in its graphics window. You can do this by inputting the following code:

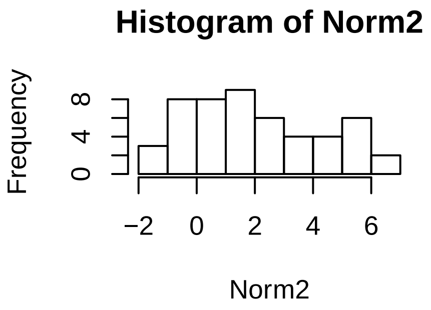
```
par(mfrow=c(2,2))
```

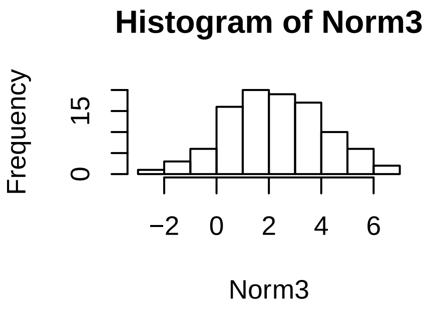
6. Use the `hist()` function to draw frequency histograms of Norm1, Norm2, Norm3 and Norm4 in the same window.
7. What do you see as the sample size increases?

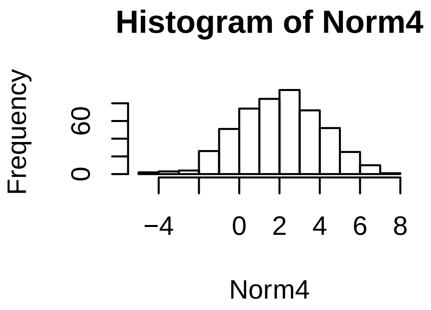
The shape of the distribution becomes closer to the shape of the underlying distribution

```
Norm1<-rnorm(25, mean=2, sd=2)

Norm2<-rnorm(50, mean=2, sd=2)

Norm3<-rnorm(100, mean=2, sd=2)

Norm4<-rnorm(500, mean=2, sd=2)

par(mfrow=c(2,2))

hist(Norm1)
hist(Norm2)
hist(Norm3)
hist(Norm4)
```


Histogram of Norm1
Frequency
0
3
6
−2
0
2
4
6
Norm1


Histogram of Norm2
Frequency
0
4
8
−2
0
2
4
6
Norm2


Histogram of Norm3
Frequency
0
15
−2
0
2
4
6
Norm3


Histogram of Norm4
Frequency
0
60
−4
0
2
4
6
8
Norm4

The effect of increasing the mean on the shape and variance of Poisson distributed data

Use the function `rpois()` to do the following:

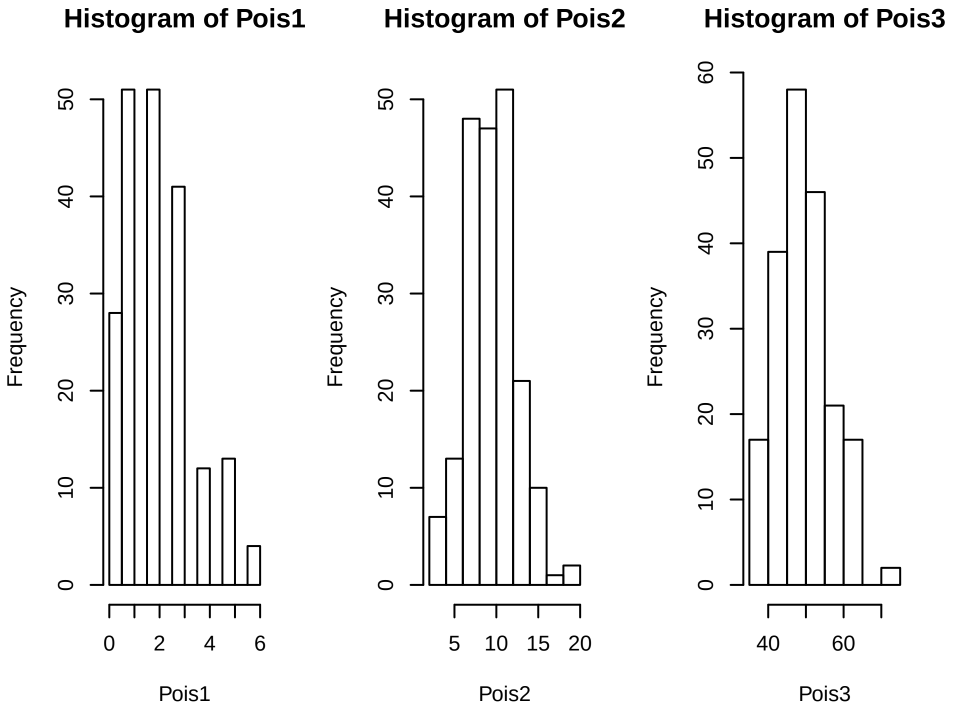
NB rpois is a little confusing in terms of how it deals with the mean, the easiest thing to do is just add it as a second argument like this: `rpois(186, 22)` which will give 186 numbers drawn from a Poisson distribution with mean 22.

1. Create an object called "Pois1" with 200 Poisson distributed numbers with a mean of 2
2. Create an object called "Pois2" with 200 Poisson distributed numbers with a mean of 10
3. Create an object called "Pois3" with 200 Poisson distributed numbers with a mean of 50
4. Set the parameter "`mfrow`" so that R will draw a 3x1 grid of graphs in its graphics window. You can do this by inputting the following code:

```
par(mfrow=c(3,1))
```

5. Use the hist() function to draw frequency histograms of Pois1, Pois2 and Pois3 in the same window.

```
Pois1<-rpois(200, 2)

Pois2<-rpois(200, 10)

Pois3<-rpois(200, 50)

par(mfrow=c(1,3))

hist(Pois1)
hist(Pois2)
hist(Pois3)
```



6. How does the shape of the Poisson distribution change as the mean increases?

It becomes less skewed and approximates more closely to a normal distribution

7. Calculate the variance for Pois1, Pois2 and Pois3. How does the variance change as the mean increases?

It increases and is approximately equal to the mean

```
var(Pois1)
```

```
## [1] 2.171633
```

```
var(Pois2)
```

```
## [1] 8.689849
```

```
var(Pois3)
```

```
## [1] 49.95337
```

8. Try to set the graphics window so that it will only show one plot at a time again.

```
par(mfrow=c(1,1))
```

Exploratory Analysis Problems

1. Download the spreadsheet entitled “Francis Black 1966 Measles Data.xlsx” from the module QM+ page. This is a dataset from a famous study which demonstrated the very strong link between population size and infectious disease epidemiology. For 19 islands from the Atlantic and Pacific Oceans we have data for the population size in 1956 and the percentage of months during the study period when measles was reported as being present on the islands.

2. Check through the spreadsheet and save it as a text file that you can read into R
3. Read the text file into R

```
measles<-read.csv("~/Dropbox/Current_teaching/MSc stats 2016/Intro exercise/Francis Black 1966 measles data.csv")
```

4. To check whether the data have been read in properly and are in the format we want, you can use the `head()` function to give you the top 6 lines of the data frame, or you can use the `str()` function to give you details about each variable in the data frame. In particular, you will want to check whether the variables that should be factors have been recognised as such.

```
head(measles)
```

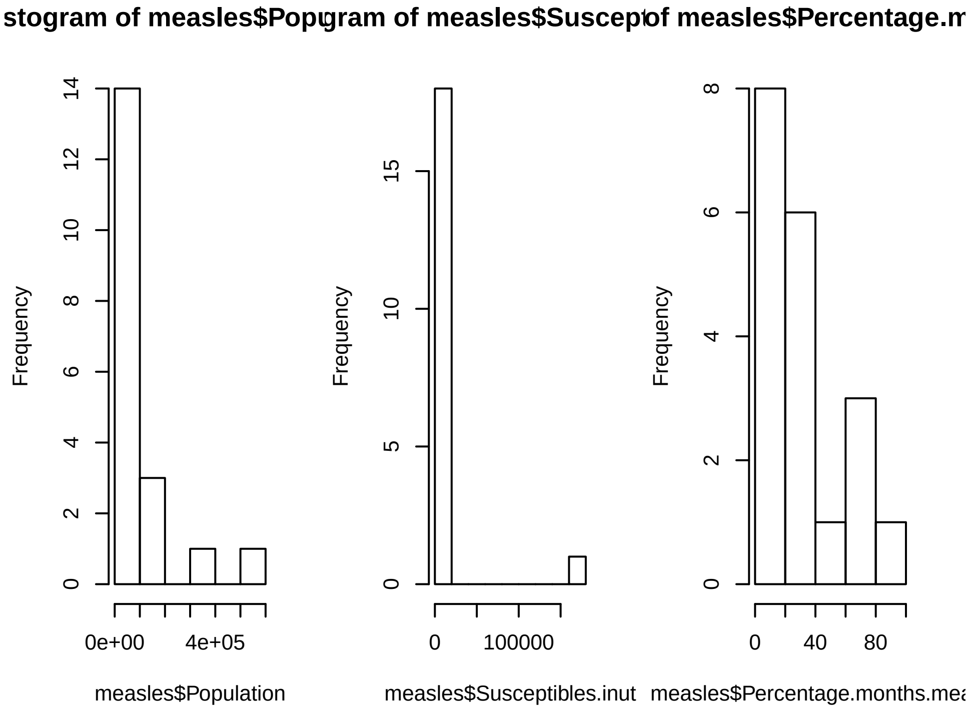
```
##     Region    Island Population Susceptibles.inut Measles.reporting.rate
## 1 Atlantic   Iceland     160000              4490                     45
## 2 Atlantic Greenland      28000              1190                    111
## 3 Atlantic   Bermuda      41000              1130                     10
## 4 Atlantic    Faroes      34000               744                     24
## 5 Atlantic St Helena       5000               116                     54
## 6 Atlantic  Falkland       2500                43                     NA
##   Percentage.months.measles
## 1                        61
## 2                        24
## 3                        51
## 4                        32
## 5                         4
## 6                         0
```

```
str(measles)
```

```
## 'data.frame':    19 obs. of  6 variables:
##  $ Region                   : Factor w/ 2 levels "Atlantic","Pacific": 1 1 1 1 1 1 2 2 2 2 ...
##  $ Island                   : Factor w/ 19 levels "Bermuda","Cooks",..: 11 8 1 4 18 3 10 5 16 17 ...
##  $ Population               : int  160000 28000 41000 34000 5000 2500 550000 345000 118000 110000 ...
##  $ Susceptibles.inut        : int  4490 1190 1130 744 116 43 167000 13400 4440 4060 ...
##  $ Measles.reporting.rate   : int  45 111 10 24 54 NA 24 8 9 6 ...
##  $ Percentage.months.measles: int  61 24 51 32 4 0 100 64 28 32 ...
```

5. If all is well, we can start with some preliminary looks at our data. Using the `hist()` function, have a look at the frequency distribution of the population sizes and, annual susceptibles input and the percent of months with measles reported. If you don’t like the number of bins used in a particular histogram, you can change it using the breaks= argument.

```
par(mfrow=c(1,3))  #I'm putting them all on one line for the sake of brevity

hist(measles$Population)

hist(measles$Susceptibles.inut)

hist(measles$Percentage.months.measles)
```

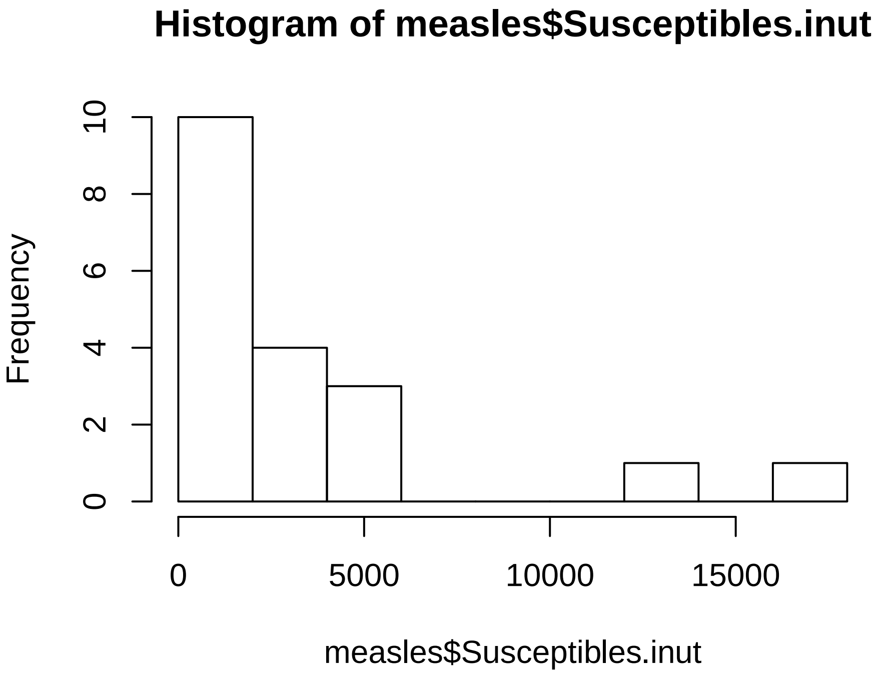


```
par(mfrow=c(1,1))
```

6. What do you notice about these variables? Are there any issues with these variables that might affect your analysis, and if so how might you deal with them?

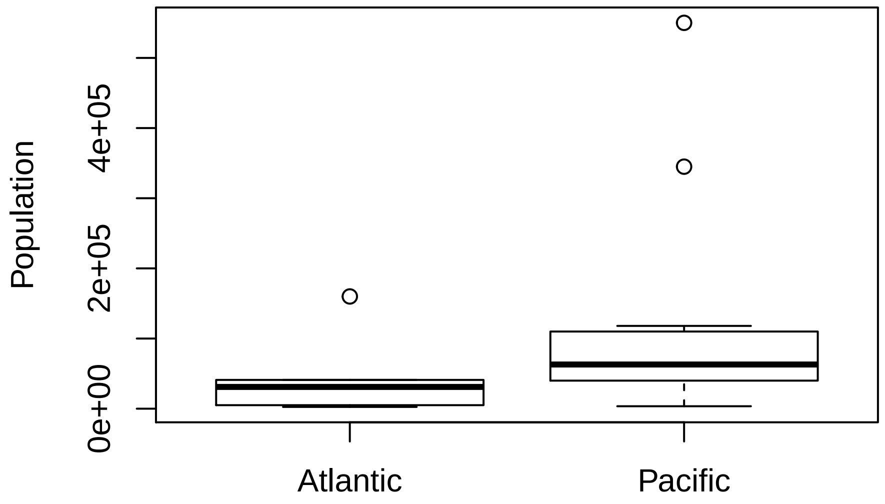
All datasets are very positively skewed. Big outlier in the susceptibles input data (has acquired an extra zero). Deal with it either by correcting in the Excel sheet (easy) and re-saving as .csv, or by changing the number using a subscript

```
measles$Susceptibles.inut[7]<-16700

hist(measles$Susceptibles.inut)
```

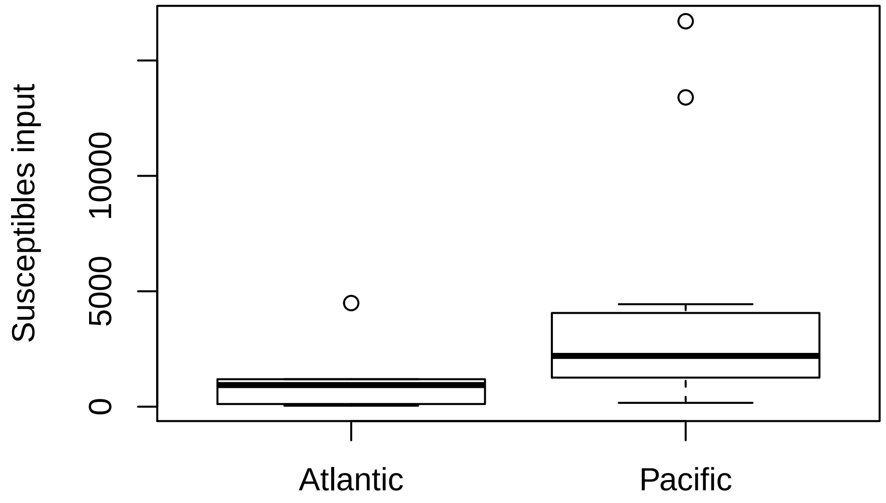


7. It's possible that the region might be an important predictor of the patterns in our data. Using the `plot()` function, generate some boxplots showing how population size, susceptibles input and the percent of months with measles reported are related to region.

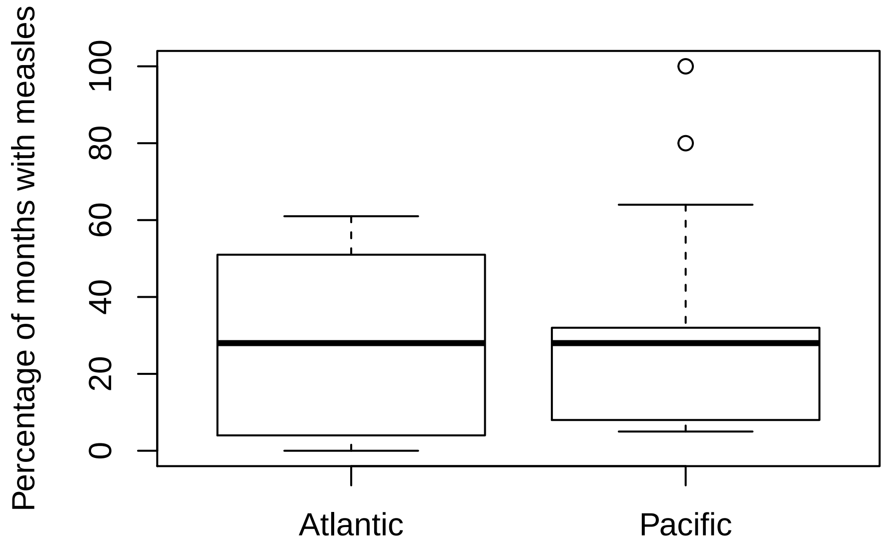
```
plot(measles$Region, measles$Population, ylab="Population")
```



```
plot(measles$Region, measles$Susceptibles.inut, ylab="Susceptibles input")
```

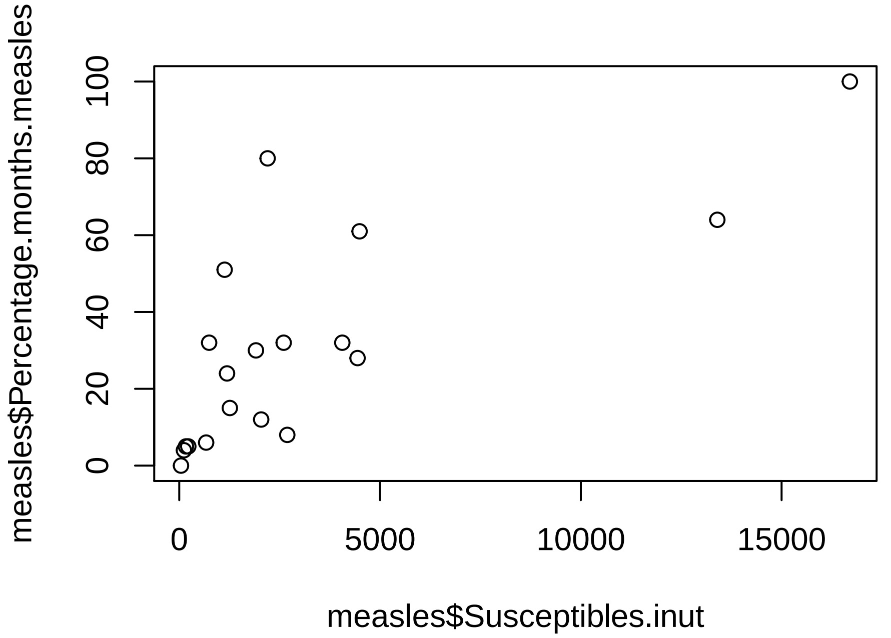


```
plot(measles$Region, measles$Percentage.months.measles, ylab="Percentage of months with measles")
```

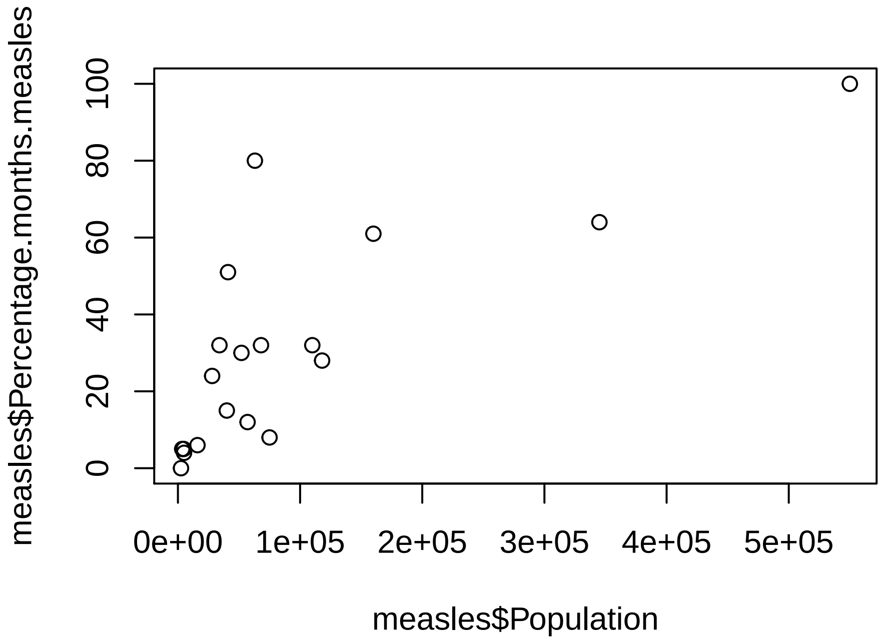


9. Draw scatter plots of the percent of months with measles reported against susceptibles input and population size (with any adjustments that you might have made earlier). What do you conclude?

```
plot(measles$Susceptibles.inut, measles$Percentage.months.measles)
```

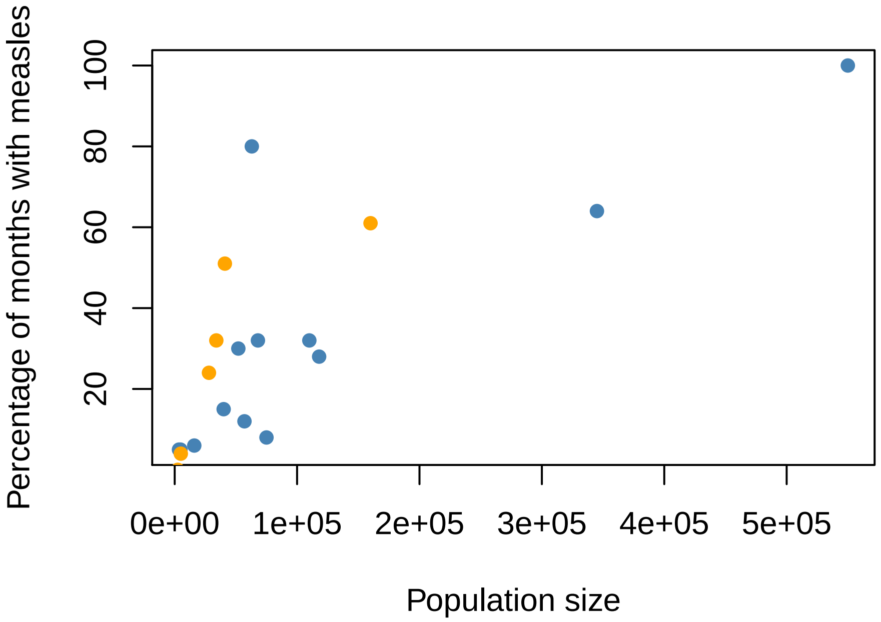


```
plot(measles$Population, measles$Percentage.months.measles)
```



10. You can change your plot parameters to change the look of the plot in the arguments for the `plot()` function. There's a good brief guide here http://www.statmethods.net/advgraphs/parameters.html. Try changing the plot symbol (`pch=16` for a filled circle for example), the colour (`col="red"` or `col="steelblue"`: see http://bc.bojanorama.pl/wp-content/uploads/2013/04/rcolorsheet-0.png for the full set). `xlab="Some text"` will set the x-axis label and `ylab="Some more text"` the label for the y-axis, and `main="Title"` will give our graph an overall title.
11. You can use the `points()` function to overwrite the points for a particular region with a different plot symbol or a different colour. You need to specify the points using a subscript (e.g. `[Region=="Pacific"]`) for both the variables used in the plot function.

```
plot(measles$Population[measles$Region=="Pacific"],
     measles$Percentage.months.measles[measles$Region=="Pacific"], pch=16, col="steelblue",
     xlab="Population size", ylab="Percentage of months with measles")

points(measles$Population[measles$Region=="Atlantic"],
       measles$Percentage.months.measles[measles$Region=="Atlantic"], pch=16, col="orange")
```


Percentage of months with measles
100
80
60
40
20
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
Population size