There are four basic data formats in R. Knowing which type you are working with is important as there are different ways of interacting with them:

1. Vectors - a one dimensional list of data items
2. Dataframes - your standard spreadsheet format in 2 dimensions (rows and columns)
3. Others - see: https://www.programcreek.com/2014/01/vector-array-list-and-data-frame-in-r/

Searching and manipulating vectors (subsetting)

```
vector1[5]           # return the 5th value of vector1
vector1[1:3]         # returns values 1 to 3 of vector1
vector1[vector1 > 1]  # returns values of vector 1 that are >1
```

Searching and manipulating dataframes (subsetting)

```
dataframe$column_name    # return this column from the df
dataframe[1,3]           # return the value in row 1, col 3
dataframe[ ,3]               # return all values in col 3
dataframe[ ,"column_name"]   # return all values of
column_name
dataframe[1, ]           # return all values in row 1
```

The below will be useful for the practical session on Thursday...

Data Import / Set Up

```
getwd()              # finds the current working directory
setwd("file_path")   # sets the working directory for the
session
read.csv('file.csv') # read in a csv file
?function_name       # bring up the R help information for a
function and examples. Very useful!
```

Exploring the dataset

```
View(df)             # view the dataframe in your R environment
                     (like an excel spreadsheet)
class()              # determine the class of an object
str(df)              # a summary of the structure of an object
(e.g. data types)
```

```
head(df),tail(df)   # prints the first, or last, six rows of a
                    dataframe
dim(df)             # the dimensions (rows and columns) of a
                    dataframe
nrow(df),ncol(df)   # the number of rows or columns in a
                    dataframe
summary(df)         # statistical summary of a dataframe
table(df$column1)   # tabulate a categorical variable in a
                dataframe e.g. how many males and females?
prop.table(table(df$column1)
                    # as above but with proportions rather
                    than absolute numbers
table(df$breed, df$coat_colour)
                    # generate a two way table of e.g. breed
                     and coat colour
```

Further Investigations

```
mean(df$column_name)
          # what is the mean of  this column in the dataframe?
mean(df$column_name, na.rm = TRUE)
          # as above but ignore NA values
median(df$column_name, na.rm = TRUE)
          # as above but for median
aggregate(numeric_variable1 ~ splitting_factor, dataframe1,
median)
          # find the median of variable1 for each level of
          splitting factor
which()   # returns the position of the elements (e.g. row of
          dataframe) for which the logical expression is true.
which(df$column > 30)
          # returns the row numbers which have values >30 in
           this column
which.max() and which.min()
     # returns the position of the element with the maximal,
     or minimal, value in an array (vector, column or row of
     data)
```

Investigate data for the Americas

```
subset(dataframe, Column_name == "column_value")
          # filter the dataframe for values of Column_name
          equal to "column_value" (must be in "" if a factor)
```

```
subset(dataframe, select = c(column1, column2))
          # subset only these columns

subset(dataframe, select = -c(column1, column2))
          # subset everything BUT these column

ifelse(dataframe$col >= 100, “high”, “low”)
ifelse(dataframe$col >= 100, 1, 0)
          # if a value is greater than or equal to 100, code
          as this, otherwise code as that
          # useful for making categorical variables
```