About this workshop

The aim of this workshop is to give a foundational understanding of the statistics needed to conduct maternal and fetal health research, and how to programme these using R. It is aimed at those who have no prior programming experience or knowledge or R, and only limited knowledge of basic statistics.

Learning objectives include:

- To understand how the R programming language works (understanding the 'syntax')
- To be able to confidently work through a data analysis lifecycle using R:
  - Loading in data
  - Preparing your data for analysis ('cleaning')
  - Visualising and summarising your data
  - Running and interpreting basic hypothesis tests and regression-based models

This workshop is intended to be hands-on: We will apply learnings to a dataset relevant to maternal and fetal health, and there will be an optional post-workshop assessment to further your understanding. The resources you will use in this workshop, as well as in the post-workshop assessment, should serve as a useful reference point for your future research.

To ensure you are able to participate fully in the workshop, **it is essential that you complete the pre-requisite activity (detailed below).**

Course materials

We will be sharing the course materials through GitHub, as well as via email. The GitHub link is here: https://github.com/UoM-R-Workshop

About R and RStudio

R is a statistical computing software and a programming language commonly used for analysing data. It's free to use and is open-source, meaning a large community of users can contribute to its maintenance. This has resulted in a software which is well equipped for analysing data, and a popularity within the academic community. RStudio is a separate software that works together with R. RStudio is what's known as an 'Integrated Development Environment', which is basically a front-end application that makes R more user-friendly. You need to have R installed in order for RStudio to work, but in you'll likely only directly work with the RStudio application.

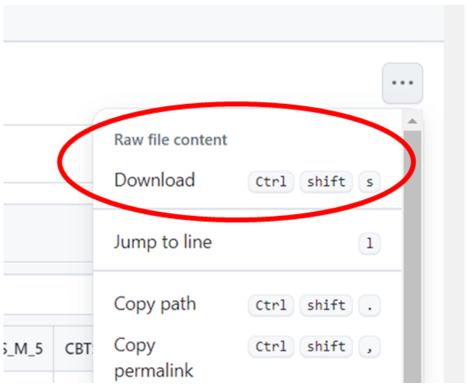
Data

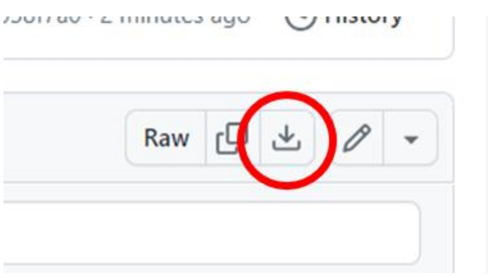
We will be using data collected from an online study of 410 mother-infant pairs. This data was collected as part of a 2022 study by Sandoz et al., who aimed to better understand the link between maternal mental health and infant sleep:

> Sandoz V, Lacroix A, Stuijfzand S, Bickle Graz M, Horsch A. Maternal Mental Health Symptom Profiles and Infant Sleep: A Cross-Sectional Survey. *Diagnostics*. 2022; 12(7):1625.

The study can be read here: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9319039/

The data can be downloaded from the following GitHub link: https://github.com/UoM-R-Workshop/Data/blob/main/data_input.csv. To download the file, please visit the link and either (1) press the three dots and 'Download', or (2) press the save icon, or (3) Ctrl + shift + S.





The original file is available on Zenodo: https://zenodo.org/records/5070945#.Y8Oq4NJBwUE (Dataset_maternal_mental_health_infant_sleep.csv). However, we recommend not using this file as there are some minor formatting errors, as well as variables which are not of interest for this workshop.

Acknowledgements

This course has been developed by Harriet Cant and Dr Vicki Palin. We would like to acknowledge Dr Hannah Lennon, whose teaching materials provided the basis of this workshop, and Chantelle Cornett, for her feedback on the course content and assistance in delivering the workshop on the day.

Pre-requisite activity

We ask that those with no experience of R please complete the below mandatory activity prior to the workshop. This should take a **maximum of 1 hour**. This activity aims to teach the basics of operating R. If you have used R before, please read through the activity before deciding whether to do it or not.

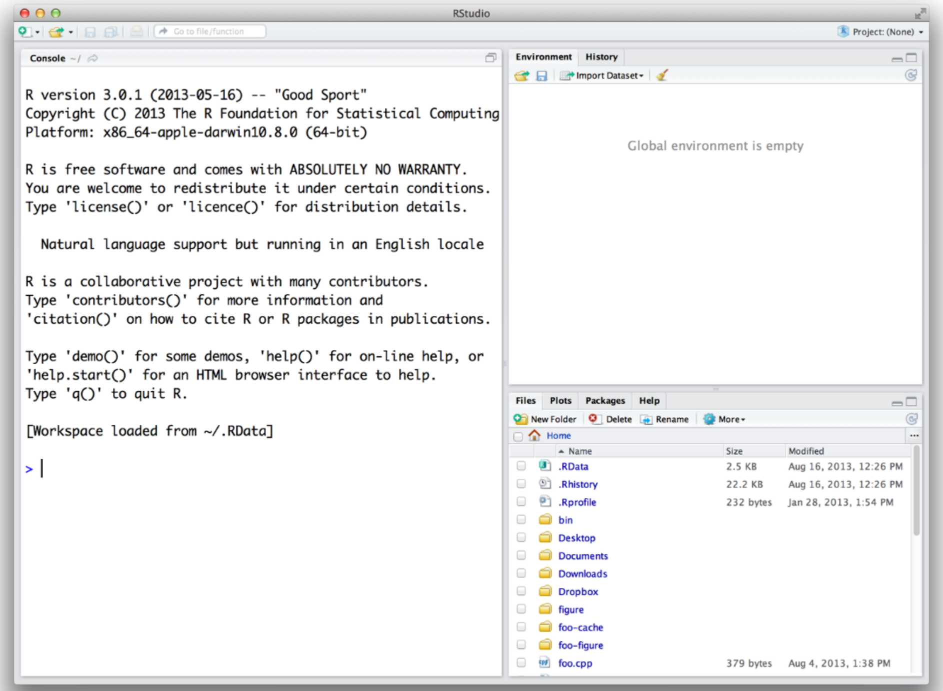
If you have any issues, please contact harriet.cant@manchester.ac.uk. We advise to complete the activity at least 48 hours prior to the workshop to allow for sufficient time to resolve any problems.

1. Download the data on maternal mental health and infant sleep from the link above. Save this in a folder where you can store all resources from this workshop, e.g. into a folder called “R workshop pregnancy”.

2. Read Sections 2.1 and 2.2 of the publication from which the data was taken (this tells you which data were collected, and how)

3. Download and install the latest version of R (Version 4.3.3 at the time of writing) and RStudio, available from:
   a. Windows:
      i. R: https://cran.r-project.org/bin/windows/base/
      ii. RStudio: https://posit.co/download/rstudio-desktop/
   b. Mac:
      i. R: https://cran.r-project.org/bin/macosx/
      ii. RStudio: https://posit.co/download/rstudio-desktop/

4. After both are installed, you should open up RStudio to find something that looks like this:



```
R version 3.0.1 (2013-05-16) -- "Good Sport"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin10.8.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

> |
```

Name	Size	Modified
.RData	2.5 KB	Aug 16, 2013, 12:26 PM
.Rhistory	22.2 KB	Aug 16, 2013, 12:26 PM
.Rprofile	232 bytes	Jan 28, 2013, 1:54 PM
bin		
Desktop		
Documents		
Downloads		
Dropbox		
figure		
foo-cache		
foo-figure		
foo.cpp	379 bytes	Aug 4, 2013, 1:38 PM

RStudio is formed of different panels. The panel on the left is your **console**, the panel on your top right shows your **environment** and **history**, and the bottom right panel shows your **file tree**, **plots**, **packages** and **help viewer**.

5. You are now using R! The **console** is where you interact with R – you write a line of code (i.e. an instruction), press enter, and R will return the output. Try writing the following things into the console, and hit 'enter' after you type them:
   a. `2+2`
   b. `2*3`
   c. `(2+2)*3`
   d. `2/3`
   e. `2^3`
   f. `a<-2`
   g. `a`
   h. `a*3`

   We use `<-` to *assign* a value to an **object**. Anything that stores data in R is called an object. In this case, we tell R that we want to store '2' as an object called 'a'. You can overwrite the value of an object by simply reassigning using the `<-` command. E.g. `a <- 3` will change the value of the object a from 2 to 3.

6. R can handle different data **types**. The ones you are likely to encounter are *numeric* (which includes *integer*), *character* and *logical* (TRUE/FALSE). Type the following into the console:
   a. `myobject <- 3`
   b. `class(myobject)`

   c. `myobject <- “hello world”`
   d. `class(myobject)`

   e. `myobject <- “3”`
   f. `class(myobject)`

   g. `myobject <- TRUE`
   h. `class(myobject)`

   We call '`class`' a **function**. Functions are commands that perform a specific task. They take inputs and return outputs. Inputs are often called **arguments**. Here, the class function takes the object name as its argument and returns the corresponding data type.

   Top tip: press the up/down arrows when in the console to search through (and easily re-run) previous commands

7. Now try the commands below:
   a. `1:10`
   b. `1:11`
   c. `c(1,3,5,7,9)`
   d. `b <- c(1,3,5,7,9)`
   e. `b`
   f. `b+1`
   g. `b/2`

Instead of single values, these commands return a **vector** of values. Vectors are one of the basic **data structures** in R. They are simply a list of items that are all of the same type (e.g. all numeric). The function 'c' takes vector values as arguments and returns a vector with these inside (c='combine'). The items are called **elements**. If you try `c(1, 2, “hello world”)` you will notice that R will convert 1 and 2 into character forms (“1” and “2”) so that they are of the same *type* as `“hello world”`.

8. R has stored our vector object b in its memory. We can see this in the **environment** panel. If you type 'B' (upper case) into the console, R tells you that it could not find the object. This is because R is **case-sensitive**.

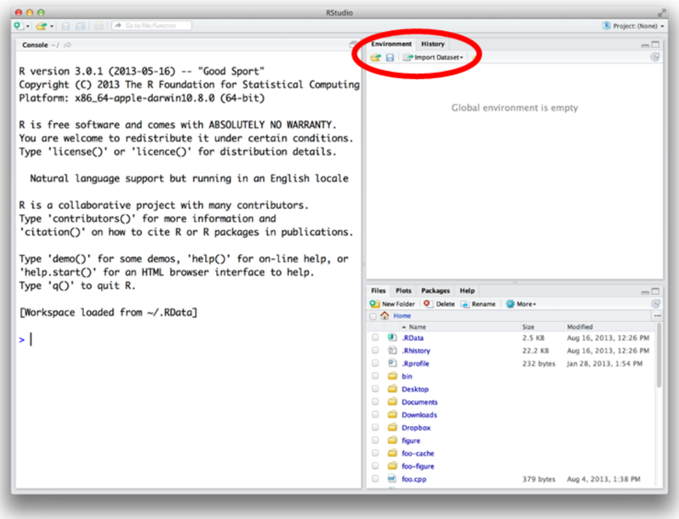
Top tip: Many, many errors in R can be attributed to using the wrong syntax (e.g. accidentally putting '-' instead of '<-'), or recalling your objects incorrectly (e.g. 'HbA1C' instead of 'HbA1c')

9. Vectors are one-dimensional. The command `len(b)` will tell you how many elements are in vector b, which is referred to as the vector length (have a go at this with one of the earlier vectors). **Data frames** are two-dimensional data structures. They have a number of **rows** and a number of **columns**. Columns are **variables**, and rows are **observations**. Data frames are analogous to an Excel data table. Try the command below to create a data frame:

```
data <- data.frame(ID = c("Person 1", "Person 2", "Person 3"),
    Age = c(23, 25, 31),
    FirstChild = c(TRUE, TRUE, FALSE))

data
```

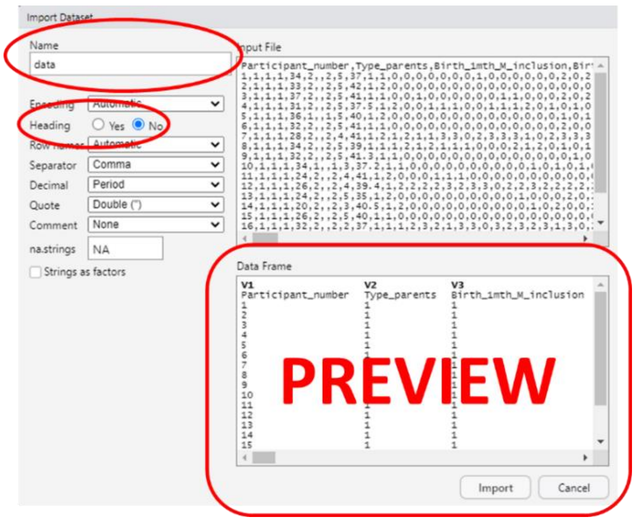
In the above, you used the `data.frame()` command to create a data frame, and specified vectors to include as columns.

10. You can also upload files into R as a data frame. To upload the maternity dataset you downloaded earlier:
    a. Go to your environment tab and select 'import dataset' > 'from text'
    b. Select the file from your local folder
    c. Edit the 'Name' field to 'data'
    d. As the file already has variable names, make sure 'Heading' is set to 'Yes'
    e. R will show you a preview of your data in the bottom panel. Click 'import'.



```
R version 3.0.1 (2013-05-16) -- "Good Sport"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin10.8.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

> |
```

Name	Size	Modified
.RData	2.5 KB	Aug 16, 2013, 12:26 PM
.Rhistory	22.2 KB	Aug 16, 2013, 12:26 PM
.Rprofile	232 bytes	Jan 28, 2013, 1:54 PM
bin		
Desktop		
Documents		
Downloads		
Dropbox		
figure		
foo-cache		
foo-figure		
foo.cpp	379 bytes	Aug 4, 2013, 1:38 PM



Name: data
Encoding: Automatic
Heading: Yes [✓ No]
Row names: Automatic
Separator: Comma
Decimal: Period
Quote: Double (")
Comment: None
na.strings: NA
Strings as factors

Input File

```
Participant_number,Type_parents,Birth_1mth_M_inclusion,Bir
1,1,1,1,34,2,,2,5,37,1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,2,0,2
2,1,1,1,33,2,,2,5,42,1,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3,1,1,1,37,2,,2,5,41,1,1,0,0,1,0,0,0,0,0,0,1,1,0,0,0,2,0,2
4,1,1,1,31,2,,2,5,37.5,1,2,0,0,1,1,1,0,0,1,1,1,2,0,1,0,1,0
5,1,1,1,36,1,,1,5,40,1,2,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1
6,1,1,1,32,2,,2,5,41,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0
7,1,1,1,28,2,,2,4,41,1,2,1,2,1,1,3,3,0,2,3,3,3,1,0,2,3,3,3
8,1,1,1,34,2,,2,5,39,1,1,1,2,1,2,1,1,1,0,0,0,2,1,2,0,1,0,1
9,1,1,1,32,2,,2,5,41.3,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0
10,1,1,1,34,1,,1,3,37.2,1,1,0,0,0,0,0,0,0,0,0,1,0,1,0,1,
11,1,1,1,24,2,,2,4,41,1,2,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,
12,1,1,1,26,2,,2,4,39.4,1,2,2,2,2,3,2,3,3,0,2,2,3,2,2,2,2,
13,1,1,1,24,2,,2,5,35,1,2,0,0,0,0,0,0,0,0,0,1,0,0,0,2,0,
14,1,1,1,20,2,,2,3,40.5,1,2,0,0,0,0,0,0,0,0,0,1,0,2,0,0,
15,1,1,1,26,2,,2,5,40,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
16,1,1,1,32,2,,2,2,37,1,1,1,2,3,2,1,3,3,0,3,2,3,2,3,1,3,0,
```

Data Frame

```
V1                  V2            V3
Participant_number  Type_parents  Birth_1mth_M_inclusion
1                   1             1
2                   1             1
3                   1             1
4                   1             1
5                   1             1
6                   1             1
7                   1             1
8                   1             1
9                   1             1
10                  1             1
11                  1             1
12                  1             1
13                  1             1
14                  1             1
15                  1             1
```

After you click import, a line of code will appear in your console. This is the line of code you *could* have used to import your data instead of doing this manually:

```
data <- read.csv("filepath")
```

The file path will be individual to you – this is the location of your file on your local machine.

Top tip: R requires forward slashes when specifying file paths (e.g. "C:**/**Users**/**me**/**R workshop pregnancy **/**data_input.csv"). In Windows, file paths are back slashes by default, so make sure to change these first.

11. Often you will have all your data, code, and outputs saved into the same folder, e.g. "R workshop pregnancy". You can tell R to use this folder as your **working directory**. Working directories take advantage of file paths being *relative* – i.e. the file "C:/Users/me/R workshop pregnancy /data_input.csv" is relative to the folder "C:/Users/me/R workshop pregnancy".

A working directory is what R will use as the relative path. This means that if we want to import or export data, we only need to specify the file name and R will automatically use our working directory as the 'starting point'. To set your working directory to the folder where you have your data saved, use the following command (replacing the file path with that on your local computer):

```
setwd("filepath/R workshop pregnancy")
```

E.g. `setwd("C:/Users/p34264hc/Teaching/R workshop pregnancy")`

Then, instead of having to specify the full path of the file you want to import, you can simply name the file and R will already know the folder your file is contained in:

```
data <- read.csv("data_input.csv")
```

12. Check the format of your data frame using the commands '`summary(data)`' and '`str(data)`'. The summary command summarises the spread of each variable within a data frame, and the str command shows you the structure (data types etc.)

13. It's very likely that you'll want to save the code you use. For example, you may wish to re-run your commands in future. The console allows for quick and simple actions, but is less useful if you want to re-run or share your code, or use code that spans multiple lines. In this case, we use a **script**. A script is simply a document of code that can be saved and used later. Follow the steps below to create, run and save a script:
    a. Go to File > New File > R Script
    b. Copy and paste some of the commands you've used earlier in your script to import your data:

```
setwd("filepath/R workshop pregnancy")

data <- read.csv("data_input.csv") # Change to your file name

summary(data)
```

    c. Press the 'run' button in the top right of the script panel
    d. To save your script, use File > Save As…

Top tip: You can also highlight the lines of code you wish to run in your script and press ctrl+enter

14. **Comments** are denoted by a hashtag, and tell R not to interpret the following text as a command. Comments provide extra information about what your code does and why, which can be helpful when retracing your footsteps or sharing code with others. Comments can also be used to turn lines of code 'on' or 'off'. In your script, try putting a hashtag at the start of the summary command and re-running the script. You should notice that R no longer gives you a summary of the data, because you have 'turned this line off'.

15. Finally, R consists of packages. These are free libraries of tools designed for a specific purpose, for example to perform a specific type of analysis. R packages can be developed by anyone who has written code and wishes to share this with the wider community. During the workshop we will work with several packages. Please install these before the workshop by using the following commands:

    ```
    install.packages("ggplot2")
    install.packages("table1")
    ```

That's it! Thank you for completing the pre-requisite activity. If there are any questions, please contact using the email addresses above. Otherwise, we look forward to seeing you at the workshop.