OneZoo R course pre-worksheet:

Introduction to R (Workstream 1)

Prerequisites:

Users able to install and run programs on their operating system.

Users understanding file management on their operating system.

Worksheet introduction:

This worksheet should be completed before the commencement of session one on day one (please aim to do it in advance so we can assist if there are any problems). It should not take you long, if you have used RStudio to any standard before this should be straightforward. Completing this worksheet is crucial to minimize setup time and maximize your learning experience for complete R beginners. If you encounter difficulties with the installation steps, it is essential you reach out to the research software engineer (RSE) team. Please give this worksheet your full attention and follow the instructions carefully.

Why are we learning R?

Upon completing this course, R will emerge as an additional toolset to enhance your research capabilities. You will gain the ability to analyse data you have collected from experiments. You will be encouraged to do so in a repeatable, scalable, and sophisticated manner. R will support you to streamline your workflow, generate insightful visualizations and improve the reproducibility of your research. By learning R you will also become part of a growing online scientific community where you can seek support and find possible collaborations.

Section 1- Installing R and RStudio (Windows, Linux & Mac):

Note 1: If you've previously installed RStudio before this course, you can bypass this section. However, please ensure you have the latest version installed (you should receive a notification if an update is required when you open RStudio).

Note 2: We will be using RStudio for this course. While there are alternative options available, RStudio is widely regarded as an excellent choice due to its extensive community support and user-friendly interface. If you are already using an alternative, you are welcome to continue using it; however, we would recommend using RStudio as we cannot provide support for alternatives during this course.

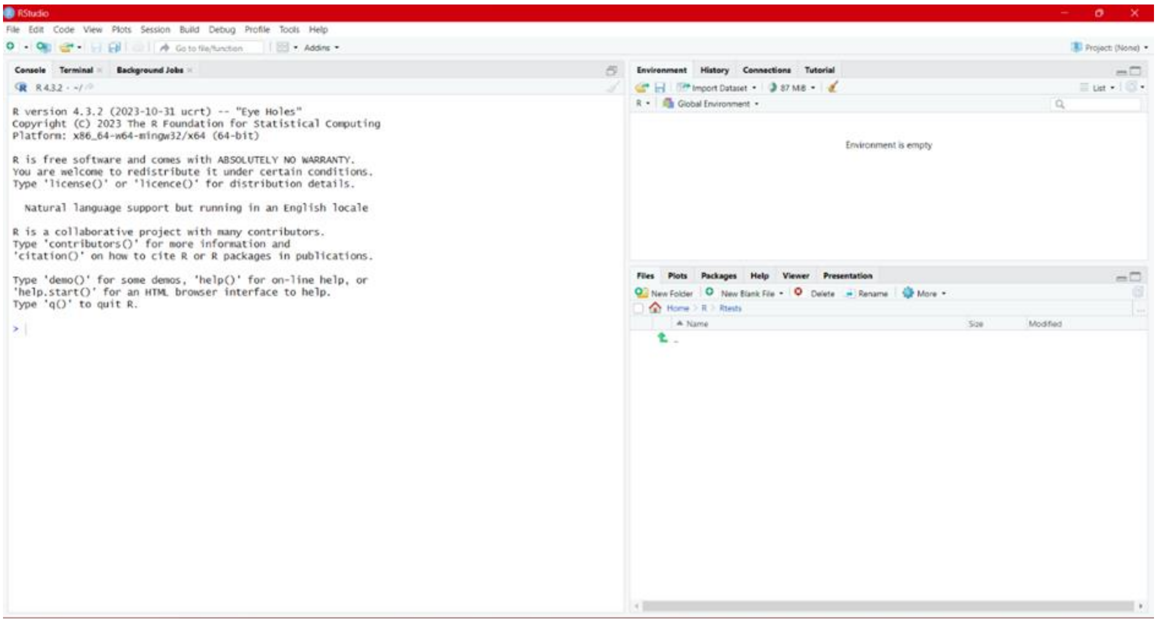
Firstly, we need to ensure that your personal system is set up correctly. Please click on the link below or copy and paste it into your browser. Follow the instructions on the page to install R and RStudio. Note that there are separate instructions for each operating system (OS). Choose the instructions that correspond to your machine's OS.

https://rstudio-education.github.io/hopr/starting.html

If the distinction wasn't clear during installation, it's important to note that R and RStudio are separate entities. R serves as the programming language itself, while RStudio functions as an integrated development environment (IDE). RStudio is specifically designed to provide an interface and toolset that simplifies the process of writing efficient, readable, and maintainable R code.

Note: If this is the first-time installing R and RStudio you may be asked some set up questions on your first initialisation. Choose the default options rather than any custom set-up.

If everything has gone smoothly you should now be able to open RStudio and be presented with a screen that looks something similar (dependant on your OS) to the image you see below. If that checks out, you can move on to section 2.



```
R version 4.3.2 (2023-10-31 ucrt) -- "Eye Holes"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (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.

> 
```

Section 2 – First steps:

Familiarise yourself with the RStudio layout:

- The interactive R console/Terminal (entire left)
- Environment/History/Connections (tabbed in upper right)
- Files/Plots/Packages/Help/Viewer (tabbed in lower right)

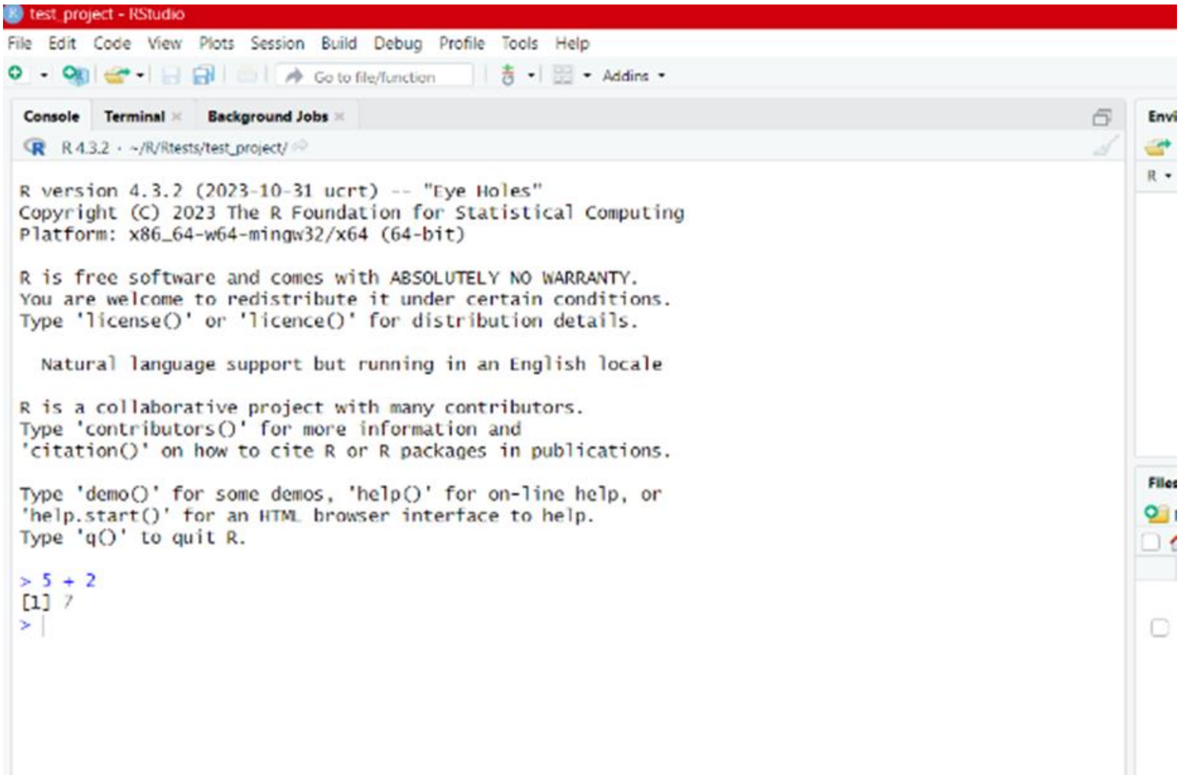
We will explore this in more detail during our introduction session.

Reminder: This worksheet is not exploring concepts in detail; our aim is to ensure you are ready for the course.

Task 1: Ensure your cursor is flashing in the console screen and enter:

```
5 + 2
```

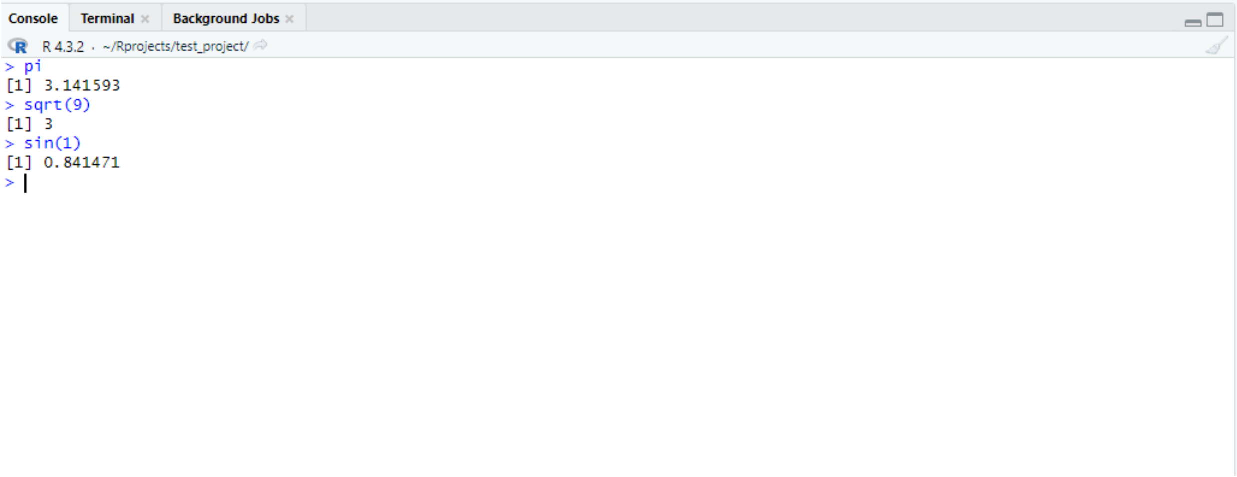
Followed by the return key (enter key). You should now have something that looks like this:


test_project - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Console
Terminal
Background Jobs
R 4.3.2 · ~/R/Rtests/test_project/
Envi
R
Files

```
R version 4.3.2 (2023-10-31 ucrt) -- "Eye Holes"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (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.

> 5 + 2
[1] 7
> 
```

Task 2 – Now try these three commands, type them separately, complete each line with enter:

```
pi
sqrt(9)
sin(1)
```



```
> pi
[1] 3.141593
> sqrt(9)
[1] 3
> sin(1)
[1] 0.841471
> 
```

This is an extremely brief introduction to inbuilt functions and variables. Hopefully these inputs are decipherable to you and the results are not surprising. You should see an output for all inputs (see screenshot).

Task 3 – Set up a project

Project management is important to ensure maintainability and collaboration. Minimise RStudio and create a folder (call it something explanatory like "r_projects") in a location that is convenient for you. Now go back to RStudio and from the main menu select 'File' > 'new project' > 'new directory' > 'new project'. You will be prompted for a directory name for your project and a file path where you want to create the project. Name your project something that you can easily identify like "OneZooRCourseWorksheet1" and choose the folder you just created as your file path.

You should now have created an R project. You can explore this project from your file browser, and you may notice you have a file that is of type 'R Project'. You can run this file to load a project into RStudio (not so useful now but possibly advantageous in the future if you have many projects). Go back to RStudio and type:

```
getwd()
```

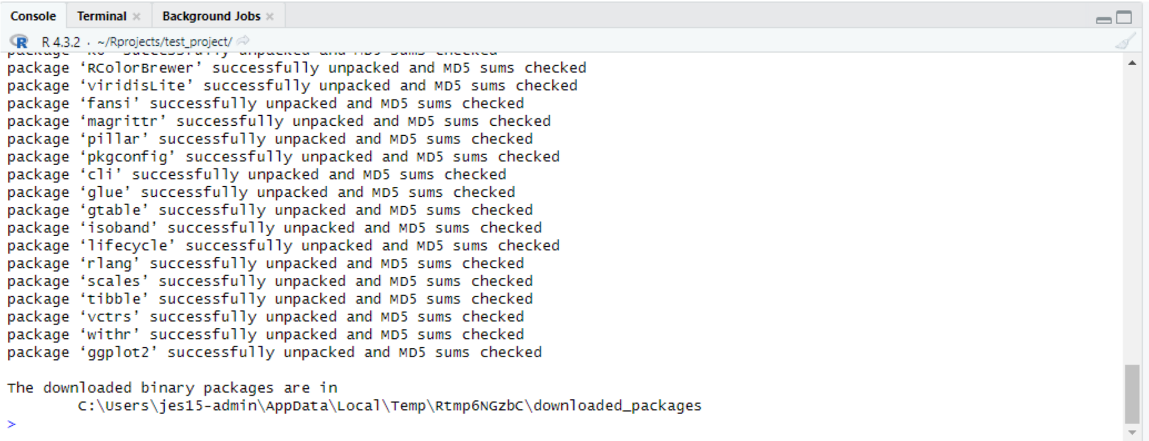
This should now show your current working directory (wd) is the path and project you specified. This is important for project organisation and data import and export.

Task 4- Packages

We need to ensure you have a series of packages installed for this course. We already have a very limited time, so we do not want to waste it installing packages. In the console in RStudio type:

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

You may get a warning message related to Rtools. This is normal and should not obstruct you from installing other packages or completing the worksheet.



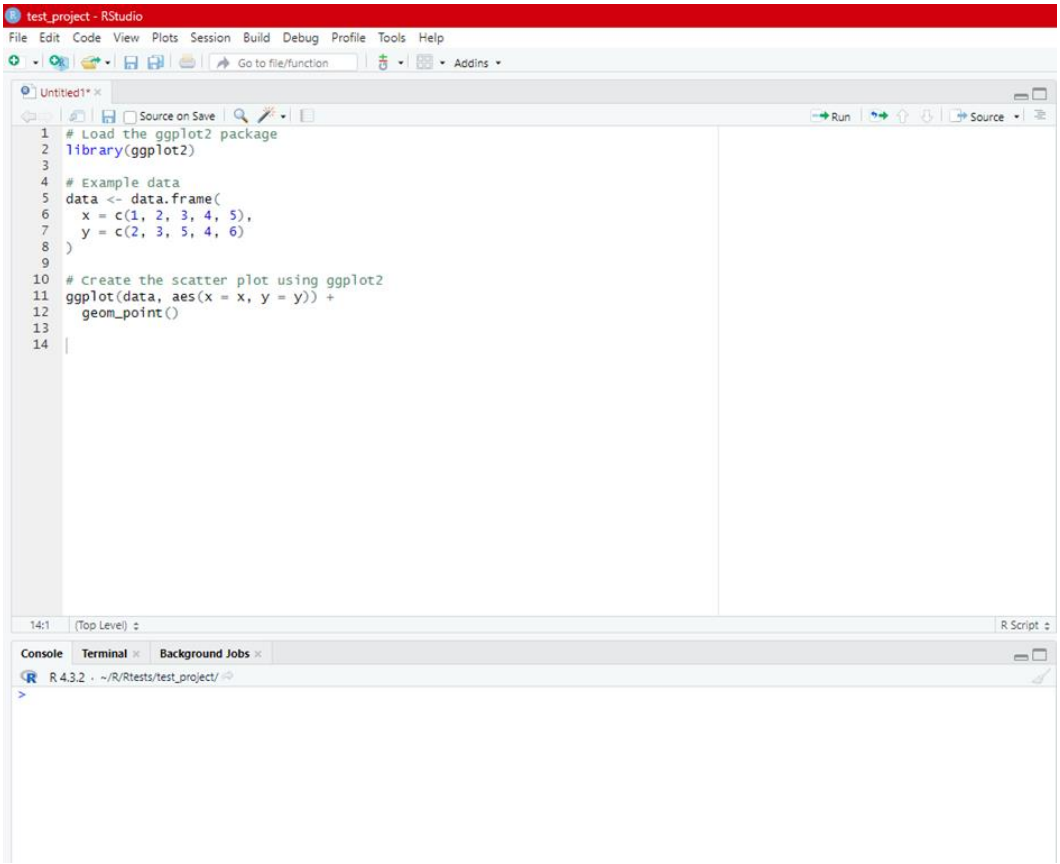
```
package ‘R6’ successfully unpacked and MD5 sums checked
package ‘RColorBrewer’ successfully unpacked and MD5 sums checked
package ‘viridisLite’ successfully unpacked and MD5 sums checked
package ‘fansi’ successfully unpacked and MD5 sums checked
package ‘magrittr’ successfully unpacked and MD5 sums checked
package ‘pillar’ successfully unpacked and MD5 sums checked
package ‘pkgconfig’ successfully unpacked and MD5 sums checked
package ‘cli’ successfully unpacked and MD5 sums checked
package ‘glue’ successfully unpacked and MD5 sums checked
package ‘gtable’ successfully unpacked and MD5 sums checked
package ‘isoband’ successfully unpacked and MD5 sums checked
package ‘lifecycle’ successfully unpacked and MD5 sums checked
package ‘rlang’ successfully unpacked and MD5 sums checked
package ‘scales’ successfully unpacked and MD5 sums checked
package ‘tibble’ successfully unpacked and MD5 sums checked
package ‘vctrs’ successfully unpacked and MD5 sums checked
package ‘withr’ successfully unpacked and MD5 sums checked
package ‘ggplot2’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\jes15-admin\AppData\Local\Temp\Rtmp6NGzbC\downloaded_packages
>
```

Note: if this is not your first time using RStudio and ggplot2 you do not need to run this line. If you do you will get a relatively short message informing you the packaged has been successfully unpacked.

You should see a set of messages informing you of all package dependencies that are being downloaded from various locations. Now open a new R script by pressing the green button with a white plus in it. This will open a new tab on the top left of RStudio. Scripts allow you to assemble commands and run them in sequence rather entering lines manually into the console like we have done so far.

Copy and paste the following into your new script (use the screenshot below for context):

```
# This loads the ggplot2 package you installed
library(ggplot2)
# This generates some simple example data
data <- data.frame(
  x = c(1, 2, 3, 4, 5),
  y = c(2, 3, 5, 4, 6)
)
# This uses the ggplot package to create a scatter plot displaying the data we created
ggplot(data, aes(x = x, y = y)) +
  geom_point()
```

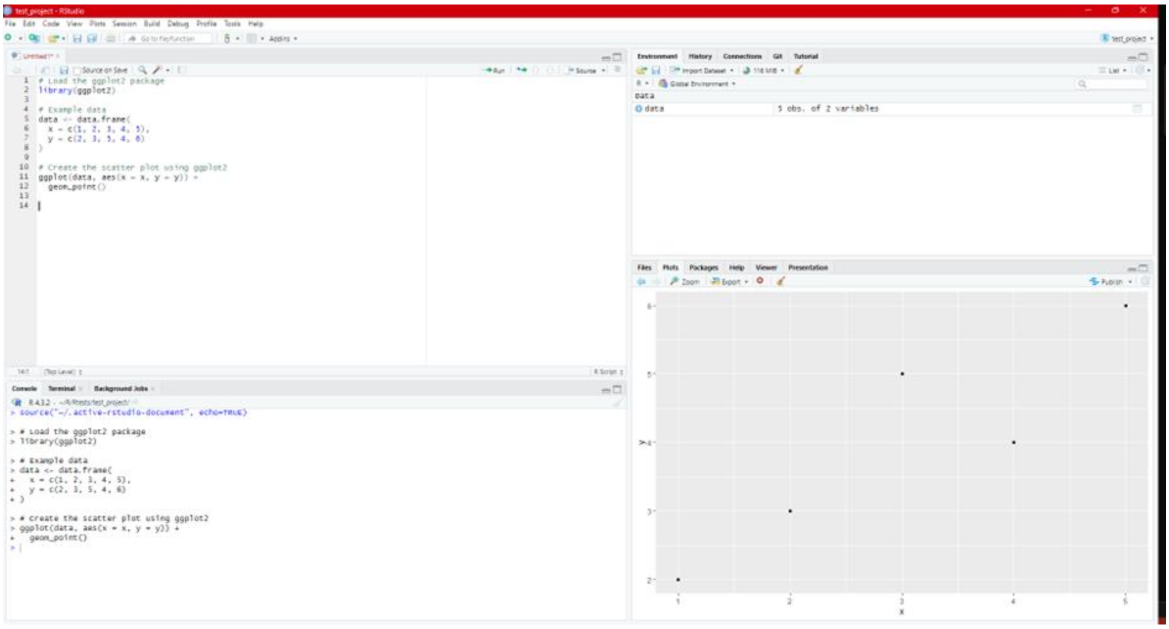


```
# Load the ggplot2 package
library(ggplot2)

# Example data
data <- data.frame(
  x = c(1, 2, 3, 4, 5),
  y = c(2, 3, 5, 4, 6)
)

# Create the scatter plot using ggplot2
ggplot(data, aes(x = x, y = y)) +
  geom_point()

```

```
R 4.3.2 · ~/R/Rtests/test_project/
>
```

To give some context to those who are using R for the first time. The # signifies a note, it is a line that will not be treated as R language and is used to give explanation (to yourself and to others who might read your code) about what the following lines of code are doing. For the moment we are not going to consider any of the lines of code we are just going to trust the process and run the script by pressing Ctrl + Enter, or Ctrl + Alt + r (Linux or Mac OS) and Ctrl + Shift + Enter (on Windows OS).

If everything is correct you should notice two things. Firstly, you should be provided with a scatter plot in the plotting window (bottom right). Secondly, you should see that the lines in the script window have been processed in the console (bottom left). Your output should match the image provided.


test_project - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
test_project
Untitled1*
Source on Save
Run
Source
Environment History Connections Git Tutorial
Import Dataset
116 MiB
List
Global Environment
Console Terminal Background Jobs
R 4.3.2 · ~/R/Rtests/test_project/
Files Plots Packages Help Viewer Presentation
Zoom
Export
Publish
(Top Level)
R Script
2
3
4
5
6
y
1
2
3
4
5
x

```
# Load the ggplot2 package
library(ggplot2)

# Example data
data <- data.frame(
  x = c(1, 2, 3, 4, 5),
  y = c(2, 3, 5, 4, 6)
)

# Create the scatter plot using ggplot2
ggplot(data, aes(x = x, y = y)) +
  geom_point()

```

```
> source("~/.active-rstudio-document", echo=TRUE)

> # Load the ggplot2 package
> library(ggplot2)

> # Example data
> data <- data.frame(
+   x = c(1, 2, 3, 4, 5),
+   y = c(2, 3, 5, 4, 6)
+ )

> # create the scatter plot using ggplot2
> ggplot(data, aes(x = x, y = y)) +
+   geom_point()
> 
```

```
Data
data          5 obs. of 2 variables
```

You can now save you script by pressing the save symbol on the script pane. Name it something sensible like ‘simplePlotExample’.

Now we will install the remaining packages.

install.packages("cluster")

install.packages("ggbiplot")

install.packages("tsne")

install.packages("caTools")

install.packages("class")

install.packages("neuralnet")

Congratulations, that is the end of the technical sections of this worksheet. If you have no experience of R or programming, you may feel like you have more questions than answers at this point. This is understandable, as previously stated this worksheet is a preparation exercise to ensure you are ready for the course. If you were unable to complete any of the exercises, please contact jas140@aber.ac.uk before the course commencement data and we will assist.

Section 3- Course expectations

The final section of this worksheet is about you and your research. We are offering you a foundational understanding of R and RStudio, along with some of the essential tools and techniques you'll require. However, please note that this course cannot be customized to everyone’s specific research and skill set. When considering the following questions (some may be relevant to you and others may not) keep in mind you are applying this course to your own research. You are the most equipped person to identify the skills you need to acquire and the challenges you need to overcome. You may want to take some time to jot down notes and thoughts. We will revisit this during our first session to help the RSE team gain a better understanding of how to support you.

Consider:

What are 3 main outcomes you are hoping to achieve from this course?

What does success look like for you in upon completion of this course?

Do you know the specific skills do you want to attain?

How do expect this course is going to support your research?

What barriers (related to the material in this course) have stopped you achieving results in your research to date?

What is your expectation for how significant mastering the content of this course will be to your ongoing research?

Do you have a data set, what questions do you want to ask of it?