Week 03 Worksheet - part 1

A first gentle introduction

INSERT YOUR NAME HERE

Welcome to RStudio!

This worksheet will introduce you to the RStudio interface and the concept of literate programming using R.

The RStudio Interface (Lab Tutor Tour)

When you open RStudio, you'll see four main panes:

1. **Source Editor** (top-left): This is where you write and edit your code and text.
2. **Console** (bottom-left): This is where R commands are executed and where you see the output.
3. **Environment/History** (top-right): This shows the objects in your workspace and your command history.
4. **Files/Plots/Packages/Help** (bottom-right): This multi-purpose pane shows your files, plots, installed packages, and help documentation.

This is a .qmd file

It is markdown - a text only language that can be edited anywhere, even on your phone, because it uses commonly occurring symbols to do all the formatting. If you happen to know html, it's pretty similar. It's simple (once you get the idea), very small in terms of file-size, easily accessed, and versatile.

Understanding Literate Programming

Literate programming is a paradigm that combines explanatory text with executable code in a single document. This approach, pioneered by Donald Knuth in 1984, aims to make programming more accessible, understandable, and maintainable. When you learn about Open Science practices next week with Stacey, she will probably talk about sharing of data, replicability and other big issues. One way in which we can make Science more accessible and open is by using techniques like this.

Key Concepts

1. **Integration of Code and Documentation**: In literate programming, the code is interspersed with narrative text that explains the purpose and functionality of the code.
2. **Human-Oriented**: The primary focus is on making the program understandable to humans, rather than just computers.
3. **Executable Documents**: The resulting document can be both read as a coherent explanation and executed as a functional program.

Benefits in Data Science and Research

Literate programming is particularly valuable in data science and research for several reasons:

1. **Reproducibility**: By combining code, results, and explanations, others can easily reproduce and verify your work.
2. **Clear Communication**: It allows you to explain your thought process, methodology, and interpretation of results alongside the code that generates them.
3. **Error Checking**: The close proximity of code and explanation makes it easier to spot inconsistencies or errors.
4. **version Control**: Changes in both code and narrative can be tracked together, providing a comprehensive history of the project's evolution.

Example in R

Here's a simple example of literate programming in R using a Quarto document:

In this analysis, we'll explore the relationship between a car's horsepower and its fuel efficiency using the `mtcars` dataset (this is a commonly used dataset that comes installed in R).

"The Results of My Amazing Experiment"

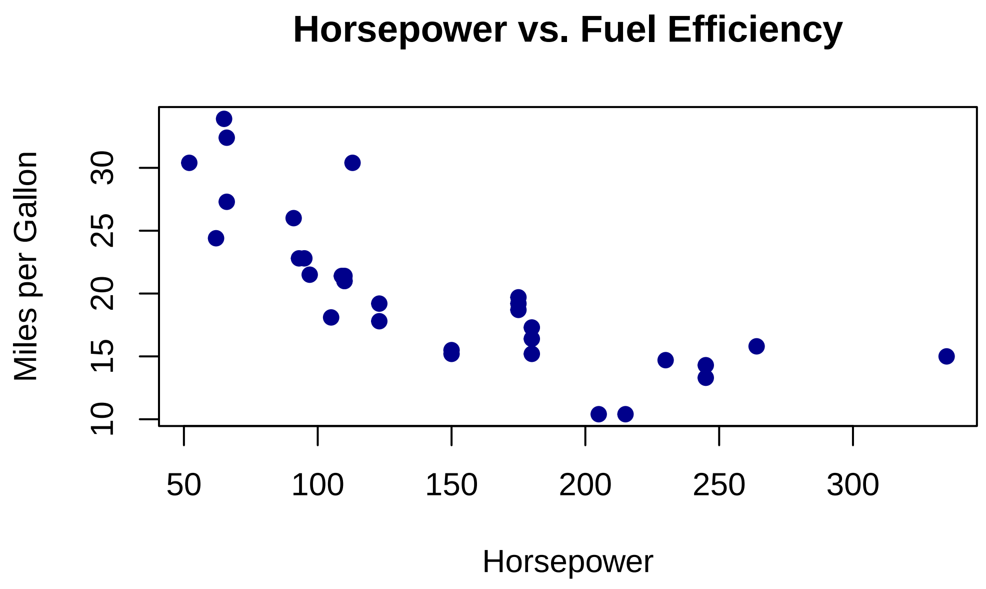
Dear Reader, be amazed at my wonderfulness! Below, I am going to walk you through the analysis. If you are reading this as a .qmd file, you will see all the code. But after I render it to html or pdf, you'll only see what I want you to see, such as just the pretty pictures, without the code! Or I can fold up the code here too!

Another aspect of literate programming, other than narrating the larger document, you can also add comments in the code, to draw an interested reader to specialist aspects of what's happening there, or just to signal what the code is doing. When you try to work with someone else's code, comments like the ones below (but better) are really helpful!

If you are reading along in the .qmd file, press the green arrow to the right of the code cell to execute it. And bingo! R does it's work and performs the calculations.

```r
# Load the mtcars dataset
data(mtcars)

# Create a scatter plot
plot(mtcars$hp, mtcars$mpg,
     main="Horsepower vs. Fuel Efficiency", # Clearly these are the most important things!
     xlab="Horsepower",
     ylab="Miles per Gallon", # Can we change this to Kilometers per Litre for other countri
     pch=19,
     col="darkblue") # The best colour for car-based statistics
```



The Visual Editor

RStudio's visual editor provides a user-friendly interface for creating Quarto documents. It allows you to write text and code in a "What You See Is What You Get" (WYSIWYG) environment.

To switch to the visual editor, click the "Visual" button at the top-left of the Source Editor.

Writing Narrative Text

In the visual editor, you can simply type your text as you would in any word processor. You can use the formatting toolbar at the top to:

- Apply **bold** or *italic* formatting
- Create bullet or numbered lists
- Add headings
- Insert links or images

Try writing a short paragraph about why you're studying Psychology below:

[Your text here]

Adding Code Cells

To add a code cell:

1. Click the "+C" button in the toolbar or use the keyboard shortcut Cmd+Option+I (Mac) or Ctrl+Alt+I (Windows/Linux)
2. You'll see a new code cell appear
3. Type your R code inside this cell

Let's try a simple calculation. Add a code cell and type the following:

```r
2 + 2
```

```
[1] 4
```

Running Code

To run the code in a cell:

Click the "Run" button (green play icon) at the top-right of the cell, or Use the keyboard shortcut: Cmd+Enter (Mac) or Ctrl+Enter (Windows/Linux)

The output will appear directly below the code cell. Rendering Your Document To create the final document:

Click the "Render" button at the top of the editor Choose your desired output format (HTML or PDF) RStudio will process your document and display the result

Exercise

Load the data Gordon has been using today. It is on the VLE here https://learn.gold.ac.uk/pluginfile.php/243093

```r
# Welcome to R! This is a comment. R doesn't run anything on a line after a '#'.

# Before we can read our data, we need a special tool called 'readr'.
# Think of this like getting a special pair of glasses to read a book.
# We only need to do this once, just like you only need to buy glasses once.
install.packages("readr")
```

```
The following package(s) will be installed:
- readr [2.1.5]
These packages will be installed into "~/Week0203/renv/library/macos/R-4.4/x86_64-apple-darw

# Installing packages --------------------------------------------------------
- Installing readr ...                          OK [linked from cache]
Successfully installed 1 package in 7.1 milliseconds.
```

```
# push the green play button
```

```
# Now that we've got our special glasses (installed the package), we need to put them on to
# In R, we do this by 'loading' the library. We'll need to do this each time we start a new
# It's like putting on your glasses each time you want to read.
library(readr)  # This is like putting on the 'readr' glasses to use its features

# push the green play button
```

```
# Now let's read our uploaded data file. This is like opening a book that's right in front o
# Make sure the CSV file is in your current working directory.
# data <- "Y1W3_data.csv" # Please DELETE the # just before `data` at the beginning of this

# Look to see `data` pop up in the Environment panel
# push the green play button
```

```
# Let's take a quick look at the first few rows of our data.
# This is like reading the first page of the book.
# print(head(data)) # Please DELETE the # at the beginning of this line

# ouch. Lots of info about the data - don't stress it now.
# push the green play button
```

```
# Now, let's get a summary of our data.
# This is like reading the table of contents and index to get an overview.
# print(summary(data)) # Please DELETE the # just before `data` at the beginning of this lin

# Some Summary Stats! Yay!
# push the green play button
```

```
# We can check how big our dataset is.
# This is like counting the pages and chapters in our book.
# cat("Our dataset has", dim(data)[1], "rows (like pages) and", dim(data)[2], "columns (like
# push the green play button
```

```
# Finally, let's see what information (columns) we have in our dataset.
# This is like looking at the headings in our book.
# cat("Our dataset contains information about:\n") # Please DELETE the # just before `data` a
# print(colnames(data)) # Please DELETE the # just before `data` at the beginning of this li

# You did it!
# Congratulations! You've just used R to read a data file and take a first look at it.
# It's like you've opened a book, skimmed through its contents, and got an idea of what it's
```

Normally...

That code chunk would only be like this:

```
install.packages("readr")
library(readr)
data <- read_csv("Y1W3_data.csv") # obviously this could be another file!
summary(data) # Even this isn't necessary if you know what the data includes already!
dim(data) # or this
colnames(data) # or this.
```

Data Viewer

If you look in the Environment panel to the right, you will see `data` and if you click on the little spreadsheet icon to the right, you can look at it like a spreadsheet!

My First Plot

```
# COPY SOME CODE IN HERE from the slides from this morning - choose one of the early, simple
```

A super easy example is `plot(Y1W3_data$LoginCount)` which is just asking to plot the Login-Count column from the Y1W3_data dataset!

To be honest, that would be enough for this first visit! Well done.

Remember to Run the code cell

Bingo!

Now Press the Render Button at the top of the page and choose either html (for a website) or pdf (for a nice tidy document).

Render your document to html (website) or pdf (document) and view the results!

Conclusion

You've now learned the basics of:

> **Note**
>
> - Navigating the RStudio interface Using the visual editor for Quarto documents
> - Writing narrative text and adding formatted elements
> - Inserting and running code cells to do sums and create graphs
> - Rendering your document

Keep practicing these skills as you continue your journey with R and data analysis!