Practical 1

Jumping Rivers

Getting Started

Start a new script file

If a new empty R script file does not open automatically when you launch RStudio you can open a new one[1]

```
File -> New File -> Rscript
```

Make sure that you choose the correct type of file. The RStudio IDE is clever in the way that it treats file extensions. Choosing a R script file will make sure that when you come to save your script it is saved as the correct type. It is also necessary to allow us to easily send R code from the text editor to be evaluated in the R terminal.

Sending your first code

Let's start with sending our first piece of code to be evaluated. In the script editor (likely the top left window) write a simple piece of code, say

```
x = 5
```

To send this code from the editor to be run by R you have a few options:

1. If the cursor is on the same line as the code you can either[2]
   - Click `Run` at the top left of the editor window
   - Press `Ctrl + enter`
2. Highlight the code that you would like to run and follow the options in step 1 above.
3. `Ctrl + Shift + s` sends all code from the editor to be evaluated.

I tend to prefer using 'Ctrl + Enter' as I like using keyboard shortcuts. If you want to see all available keyboard shortcuts either go to Help and choose keyboard shortcuts, alternatively 'Alt + Shift + k' is the keyboard shortcut for the keyboard shortcut menu.

Course R package

Make sure that you have the course package loaded into the current session. Instructions on how to install the package are contained in the appendix to the course notes. You should only need to install a package once using `install.packages()` and then we need to use `library()` in every session or script that we want to use the functionality of that package.

```
library("jrIntroduction")
```

[1] Alternatively use `Ctrl + shift + n` to launch a new R script

[2] Both of these will only run the one line of code.

Vectors

Write the following code in the editor and run it

```
x1 = GetNumericVector()
```

This code generates a large vector of random numbers such that everyone has the same.[3]

1. What is the length of `x1`?
2. What is the 55th element of `x1`?
3. What is the final value in `x1`?
4. What is the 50th smallest value of `x1`?
5. How many unique values are there in `x1`?
6. What is the total of all elements?

Solutions

Solutions to the practical questions are contained within the package

```
vignette("solutions1", package = "jrIntroduction")
```

[3] This function is part of the course package **jrIntroduction**, you can view it's help page if you like `r ?GetNumericVector()`