Writing an R Script
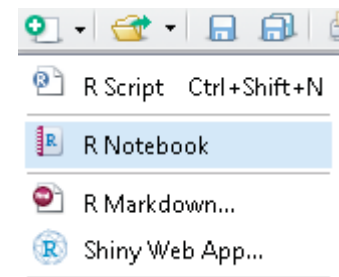
1) OPEN a new R Notebook

2) CREATE the outline

Create the outline of your notebook by writing some text describing each of the tasks in step 3 and then creating a code chunk for each one that you can fill in.

To create a code chunk, use the syntax shown on the right or select "R" from the "Insert" drop down. If you want to save this script, when doing "Save As," the ".Rmd" extension must be manually added.

3) WRITE a script that:

- Loads the necessary libraries - in this case ggplot2 (use the function `library()`, use `?library` for help)

- Reads the gapMinderDataFiveYear data set into an object called `data.full` (use the function `read.csv()`, use `?read.csv` for help), Note that if the data set is not in your working directory, the full path must be included. You can change the working directory in the Session menu.

- Creates a summary of the data (use the function `summary()`, use `?summary` for help)

- Creates a simple plot of population vs life expectancy (use the function `plot()`, use `?plot` for help - additionally you may want to do a Google search for how to plot columns from a data frame)

- Creates a plot of the distribution of population using the `geom_histogram` function from the ggplot2 package (see `?geom_histogram` - the examples can be quite helpful)

4) RUN the script

1. Run all chunks in the script (Ctrl + Alt + R)

2. Run individual lines in the script (highlight then Ctrl + Enter)

5) VIEW the output

1. View the data in the environment tab (click on `data.full` to open a spreadsheet view)

2. Check the output under the code chunks for the graphical output. Note if you are using a plain R Script the graphical output will appear in the Plots tab.

3. View the console and the output under the code chunks for the non-graphical output from the script

6) EDIT the script

1. Add a title and axis labels to your plots (see `?plot` and `?ggtitle` for help)

2. Save the script and run just the section that modifies the plot (highlight then Ctrl + Enter, or just run that chunk)

3. Check the output

7) REMOVE items

1. Remove `data.full` from the environment (use the environment tab or see ?rm)

   ➤ This is useful when data is so large it uses up too much memory.

Tip: R is a programming language that works purely "in memory", which means that all of the objects you see in your environment are right now sitting in the computer's RAM (Random Access Memory). If you read in data, R goes to the hard drive to find it, then makes a copy that sits in RAM. If you have very large data sets in your environment this can usually take up a significant portion of your computer's RAM and you may be limited with what more you can do in R. Therefore it is always good practice to be careful with what you leave in your environment, if you don't need it anymore, remove it. You can always save the object as an .RData file and reload it later if you need to.

8) SAVE the session

1. Go to Session > Save Workspace As..., save to a file with a ".RData" extension (e.g. "fileName.RData").

2. You can then reload your session later including the environment and history by going to Session > Load Workspace... (or by double clicking on the ".RData" file).