# RMarkdown - Exercises

Before you start:

* You will need to install the ‘arsenal’ package. To do this type:
  + install.packages(arsenal)
* RMarkdown ‘cheatsheets’ are available directly in RStudio
  + Help -> Cheatsheets

1. **Default documents and output styles**
   1. Generate a default Rmarkdown document with html output
   2. Change the output options to PDF and Word.

Notes:

Your computer may not have Latex installed, if so see the solutions for the PDF output

If you want to update the Rmarkdown document and see the changes you will need to close the Pdf or Word outputs, otherwise they won’t load

* + 1. What changes happen to the YAML?
    2. Which output format do you think looks best?

1. **Lists and formatting**
   1. Replace the contents of the Rmarkdown document (from ‘## R Markdown’ on line 12 onwards) with a numbered list of your 5 New Year’s resolutions (you can make these up…!)
      1. Give your document a title, change the author and check the date is correct
      2. Make bold any/all resolutions you still haven’t broken
      3. Strikethrough resolutions you have broken and add the date as an indented bullet point (note: 4 blank spaces to indent)
      4. ‘Knit’ the document to your preferred output (check it looks as expected)
      5. Optional: Explore other text formatting options described on the RMarkdown cheatsheet
2. Embedding plots, tables and inline maths
   1. Start a new Rmarkdown document, select your preferred output style
   2. Delete the default text (from line 12)
   3. Plots
      1. Insert a blank chunk or r code
      2. Add the code for a plot to this chunk (use plots from previous sessions or start with the ‘mpg’ dataset in the ‘ggplot2’ package)
   4. Tables
      1. Insert another blank chunk of r code, include the option ‘results=”asis”’ to the chunk options
      2. Add the following code to create a table

library(arsenal)

table\_one <- tableby(drv ~ hwy + class,

data = mpg,

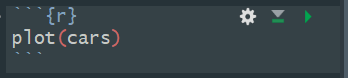
test=TRUE,

total=TRUE)

summary(table\_one)

* + 1. Optional: Compare outputs from the ‘arsenal’ package with tables generated by the ‘table’ or ‘ftable’ commands. Explore other tables which can be generated by the ‘arsenal’ package or compare
  1. Inline R code
     1. Add a new heading ‘Inline R code’ after the latest chunk
     2. Combine text and inline R code to describe the number of rows in the ggplot2::mpg dataset (nrow(ggplot2::mpg))
     3. Describe the mean fuel consumption (‘hwy’) and its associated standard deviation.
     4. Optional: format the mean and standard deviation to show only 3 decimal places

1. Chunk options
   1. Using the previous Rmarkdown document use chunk option ‘echo=TRUE/FALSE’ to change whether the R code is printed in the output
   2. Use the chunk option ‘eval=TRUE/FALSE’ to control whether the code in the chunk is run
   3. Change the size of the first plot by varying the ‘output.width’ or ‘output.height’ chunk options
   4. Add a caption to the boxplot using the ‘fig.cap = “”’ option
      1. Vary the output formats, does the caption still look right?
2. Optional: RNotebook
   1. Load a new RNotebook
      1. File -> New File -> R Notebook
   2. Click the ‘Preview’ button
      1. Does the output include a plot?
   3. In the RNotebook run the chunk of R code containing the plot by clicking on the green ‘play’ in the top right corner of the chunk



* 1. Add in code chunks from previous exercises and explore whether chunks have to be run in order or can be run out of order
  2. Explore toggling the visibility of code in the output document