

Exercise 3B: R Markdown and pipe

Niels Richard Hansen

December 8, 2016

The *R Markdown Cheat Sheet* may be helpful.

This is a continuation of Exercise 3A. It's a good idea to have a script with the code from Exercise 3A that works when you source it. Try hitting the Compile Report button before proceeding.

R Markdown

- Create a new R Markdown document. Choose the default output format HTML. Save the file and knit the document. When you have checked it works, erase the default template content except the YAML header.
- Create your first code chunk (the Insert button or Code menu, what is the keyboard shortcut?). Write code in the chunk to read in the supermarket data and print the resulting tibble. Knit the document. Note that you will need to explicitly load all required packages somewhere as the document's code chunks are evaluated in a different R session.
- Add the `cache = TRUE` option to the chunk. Knit the document twice and note the difference.

Convert your script

- Add the remaining parts of your script to the document as code chunks. Give the chunks names.
- Introduce sensible section headers and other structure to the document, add relevant comments etc.
- Introduce a global `cache = TRUE` option. Specify, for each chunk, which chunks it depends upon. (Manually specifying dependencies sounds like a hassle, but is really not a big issue in practice. Dependency specifications control which cached chunks need to be updated when you make changes. You can try to rely on automatic detection of dependencies by setting `autodep = TRUE` – either globally or for individual chunks.)
- Experiment with converting nested function calls into usage of the pipe `%>%` operator.

Customization

It can require some customization to get figures and tables displayed well.

- Experiment with the global YAML options `fig_width` and `fig_height` and the chunk options `out.width` and `out.height`.
- Use the function `kable` from the `knitr` package to format tables.