It was at the EdinbR talk this week by the RStudio community lead – Curtis Kephart. It was really interesting, but I disagree with his suggestion to point and click different parameters when you want to generate multiple reports from the same RMarkdown file. This might be acceptable if you have one or two, but any more and the chance for error and tedium is greatly increased. This blog post shows you how to loop (yes – an actual for loop!) through a variable to generate different reports for each of its unique values.

First, we need an RMarkdown file (.Rmd). This is largely the same as your usual .Rmd file, and I strongly encourage you to develop it like one. i.e. write your single .Rmd file and convert it into a special use case to be a template. Working like this makes debugging a whole lot easier. Here’s an example of a “normal” .Rmd:

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo=F, message=F, results='hide', warning=F, fig.height=9)

source("read.R")

library(tidyverse)

library(RColorBrewer)

library(knitr)

```

## Introduction

This file provides a summary of the digital potential in the rural UK survey.

In particular it shows the answers for females.

The summary had `r nrow(df)` responses, of these `r sum(is.na(df$Eastings))` either had no postcode or the postcode supplied did not match the current list from Ordnance Survey or the UK data service (Northern Irish postcodes).

Those postcodes which were matchable could be related to country and their rural-urban classification.

You can see it’s not far off what you get when you opt to start a new RMarkdown file in RStudio. I’ve abstracted the data reading to a separate file (it has some lengthy factor cleaning and is used in a few different situations), and I’m loading the knitr library so I can make tables with kable().

The next code chunk shows how the file is adapted to be used as a template for many outputs:

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo=F, message=F, results='hide', warning=F, fig.height=9)

library(tidyverse)

library(RColorBrewer)

library(knitr)

df1 = df %>% filter(gender==v)

df1 = droplevels(df1)

```

## Introduction

This file provides a summary of the digital potential in the rural UK survey.

In particular it shows the answers of `r v`.

The summary had `r nrow(df1)` responses, of these `r sum(is.na(df1$Eastings))` either had no postcode or the postcode supplied did not match the current list from Ordnance Survey or the UK data service (Northern Irish postcodes).

Those postcodes which were matchable could be related to country and their rural-urban classification.

Pretty similar, but there are some subtle differences. We’re now passing a title parameter to our .Rmd, our data are already loaded and we subset them to df1. In reality, this second step could happen in the next file – choose your preference for readability (and whether you want to change all the df variables in your .Rmd to df1).

Finally, we need a separate script to loop through our variable and make some reports!

library("rmarkdown")

source("~/repo/read.R")

slices = unique(df$gender)

for(v in slices){

render("~/repo/exploratory\_template.Rmd",

output\_file=paste0("~/results/exploratory\_", v, ".html"),

params=list(new\_title=paste("Exploratory analysis -", v)))

}

Note we’re explicitly loading the rmarkdown library here so we can use the render function. We’re also loading our data before our loop, to speed our code up. The object v is passed to the .Rmd file, which is what we use to subset our data.

If you’ve made it this far you should now have the tools to make multiple reports with a lot less effort! Beware that it becomes very easy to make more outputs than anyone could possibly read – with great power comes etc, etc..