Advanced R Worksheet

Answer any of the following questions in whatever order you want – you probably won't have time to do them all in the lab, so focus on those you think will be of most interest/use to you.

Data Simulation

1. Simulate an independent samples experiment where you have two groups of participants (15 in each group) responding on a reaction time task – assume the data distribution shape is normal and that the parameters of the RT data for Group 1 are a mean of 500 and a SD of 25, and for Group 2 the mean is 750 and the SD is 25. Plot the data using `geom_point()` for one simulation.

2. Write a loop to simulate 1000 samples of the above experiment. What is the estimated effect size, and what power do we have to detect that effect? Hint – in addition to setting a loop to simulate the data from 1000 samples, you'll also need to work out the t-value and associated p-value for each sample. You could either do this within the same loop you're using the create the 1000 samples, or you could write another loop just to do the t-tests after you've created your 1000 samples. Have a look at both the slides and the R script this morning if you're not sure where to start. When you're learning to code, much of what you'll find useful is modifying code written by some else…

Animated Data Visualisations

3. Using either a built in dataset (e.g., the mgp or starwars dataset) or an external one (e.g., the NHANES dataset which is accessible in the package called "NHANES") build some data viz animations to communicate some interesting aspect of the data.

4. Install the BBC style package using:

```
devtools::install_github('bbc/bbplot')
```

and apply the `bbc_style()` function to the above animations.

Text Analysis

5. Use the tidytext package to plot a wordcloud and run a sentiment analysis on qualitative data you might have already collected. You'll need to have a look at this morning's script for the code needed to get the text in the right format for (e.g.) plotting a wordcloud. Have a look at the code from line 658 – the `unnest_tokens()` function is used to split a text file into individual words (or tokens) and `get_sentiments()` function (see line 520) is used to get the sentiment for each of those words. A good starting point would simply be to map your textfile onto the variable called `text` and then modify my code.