University of Glasgow
College of Arts
SGSAH/SGSSS: Spring into Methods

Wednesday 18 April
0930–1100
Professor Marc Alexander

Stylometry with R

Today, we will use the *Stylometry with R* package, written by Maciej Eder, Jan Rybicki and Mike Kestemont. Their site is at https://sites.google.com/site/computationalstylistics/.

The *Stylometry with R (SwR)* scripts run inside R, which is a programming language used mainly by statisticians, but which has become popular with a wider range of people lately (including a large number of linguists). It is installed on most universities' machines by default; it is also freely available should you want to use it yourself at home (http://www.r-project.org/ – if you are interested in R, you might want to look up Baayen's 2008 *Analyzing Linguistic Data: A Practical Introduction to Statistics Using R* (Cambridge University Press) or Gries' 2009 *Quantitative Corpus Linguistics with R: A practical introduction* (Routledge), as while very useful, R is often not for the faint of heart!). Many people prefer using RStudio (https://www.rstudio.com/) as an interface to R.

The main difference that many humanists are unused to with R is that it runs from a command line – to use it, you type in commands by hand and press return. If you have correctly entered a valid command, it works; if you haven't, it will refuse to do anything. You must accurately and precisely type any command you wish to use, with no errors, even minor ones.

However, the *Stylometry with R* package includes an embedded clickable user interface, so we can use this. To get to the stage where we can start this interface, we do need to briefly use the command line first, though…

What you need

- A working folder
- Inside this folder, another folder called 'corpus' (must be lower case), which contains all your corpus files for analysis (preferably in plain text format)
  - Note: When adding in your own files, eg from Gutenberg or somewhere else, you are usually better opening up the Notepad program in Windows and copying and pasting the full text you want to use into a blank Notepad document (you can hold down Ctrl and A at the same time to select all the text in a window). Then you can save the text file where you want within Notepad.
  - The files should be named with the author's name, then an underscore, and then the text's title, so: 'authorname_title.txt' (eg 'Austen_PrideAndPrejudice.txt' or 'Christie_MysteriousAffairAtStyles.txt' – or, 'Unknown_Unknown.txt'). *SwR* colour codes the visualisations based on what comes before the underscore.
  - For the *SwR* script, when saving text you are best to choose the ANSI text encoding option in the Save dialog box. Not all text formats are the same, unfortunately, and *SwR* likes ANSI best.
- You also need R to have the *Stylometry with R* package installed.

How to start the script

Open **R x64** (the icon is a large R, shown to the right). This will open up what looks like a text document – this is actually a command line. Typing things into this window (after the > prompt, which tells you R is ready for you to type) and pressing return will tell R to do whatever you have just typed, providing it can understand you. Above the > prompt is some help and about text, including a

copyright notice.

- You can play with R at this stage, and give it some basic commands to try out how a command line operates. Try entering `1+2` and pressing return. Now try `12/3`, which is how R represents division, or `sqrt(6)`, which is a command which finds the square root of whatever's in the brackets.

Now, firstly tell R how to find your corpus folder. To do this, go to the Misc menu item and choose Change Working Directory. Then select the folder which contains your 'corpus' folder. (For us, *stylo1* and *stylo2* in the folders you downloaded from Dropbox.)

Then install Stylometry with R. Go to the Packages & Data menu item and then Package Installer. In the box which comes up, click Get List (which means it will get a list of packages available) and scroll down to install the `stylo` package. Depending on the machine, you might need to try a few options here depending on where it will allow you to install the package.

Now load and run the package:

- `library(stylo)`
  - This tells R to load the *Stylo* package of scripts from its library.
- `stylo()`
  - And, finally, this command tells R to now start the *Stylometry with R* script.

At this point, R will think for a second and then start up the *Stylometry with R* interface.

We won't use the command line any further, other than running the *SwR* script again and again. For this, all we need to do is press the up key on the keyboard, meaning R will repeat the last command entered, and then hit return again. Remember if you want to look at another folder other than Stylo, you'll need to redo setting your working directory.

I've called the folders I've already given you "Stylo", for simplicity. When working on your own texts, you may want to give the folder which contains your corpus directory a meaningful name (like HenryJames or CurriculumReports), since the name of the folder will appear as the title of the graphs that *Stylometry with R* will generate for you.

Stylometry with R - the interface

There are five tabs in the *Stylometry with R* interface. You tell it in these tabs what you want it to do, click OK, and then R will run its analysis on whatever you have in your 'corpus' folder, which is inside the folder you chose when you changed your working directory. You don't get to pick the files – it runs on everything which it finds in this folder, and this is why we have multiple folders to analyse.

For now, just to see what happens, click OK. The interface will disappear, you will see text in the R command line as the script runs, and after a few moments a graph will pop up which will tell you things about the files in the corpus directory.

This is what the *SwR* script does; you set options and it does an analysis for you, showing you the results afterwards. Close the graph and re-run the *SwR* script by hitting the up key and the return key in R.

This time, you can pick your options. The tabs are:

- *Input & Language*. Here, you can choose the types of files in your corpus folder (plain text is recommended) and tell it what language your files are in. The difference between the three English options is fairly technical, and can be ignored.
  - If you really *must* know, 'English' treats contractions and compound words as separate words ('it's topsy-turvy' becomes 'it s topsy turvy'), 'English (contr.)' treats contractions together but still counts compounds as

separate ('it's topsy-turvy' becomes 'it's topsy turvy'), and 'English (ALL)' treats both contractions and compound words together ('it's topsy-turvy' stays as 'it's topsy-turvy'). You normally want to stick with 'English', which is normal in corpus linguistics (AntConc does this, as you've probably noticed in recent weeks).

- *Features*. Probably the most important in terms of the analysis. Here, you choose what it is the script will be looking at in the texts: words or characters, and how many of each it will count (ngram size, which we've looked at in previous workshops; this should be at 1). As well as this, you can choose your MFW settings. Here, you can choose which Most Frequent Words to look at, and some more advanced settings we're not going to use here (check the *SwR* manual for more!).
  - **Important:** You can either set *SwR* to run its analysis once, or to run its analysis over and over for different MFW values. You do this by setting a minimum MFW value, a maximum MFW value, and an increment value. Then the system will run the analysis once for the minimum value, then add the increment value to that number, run it again, and so on over and over until you hit the maximum. So if your minimum is 100, your maximum 500 and your increment 50, the system will run an analysis for the first 100 MFWs, and then also for the first 150, the first 200, 250, 300, 350, 400, 450, and 500 MFWs.
  - Needless to say, you should use this with caution! It takes a while to run each of these. In most cases, you'll be running one analysis and want the Minimum and Maximum MFW values to be the same.
  - You can also choose to ignore the first part of the frequency list by changing the *Start at freq. rank* option.
- *Statistics*. Here you choose what you want to get out of the script and how it should do it. The options we will use are:
  - Cluster analysis (the tree-like graph dividing up the corpus)
  - PCA (corr.) (the scatterplot graph with points representing each text in the corpus)
  - Consensus Tree (the analysis which runs lots of cluster analyses and shows a radial graph with clusters emerging from the centre of the screen)
    - **Warning!** This takes a fair time to run, and it requires a range of MFW values (see above) with an increment; you should not set this to be too large (4 or 5 steps is fine).
  - The other options we generally ignore (MDS and PCA (cov.) are other variations on a PCA graph), and we don't go into using other distance statistics, preferring in this course to stick with Classic Delta.
    - You can feel free to experiment with others in your own time, if you like – there's lots to be said for the improvements to Delta made by Eder and Argamon.
- *Sampling*. We ignore this in our present exercise.
  - For future reference, the *SwR* manual says: 'When the analyzed texts are significantly unequal in length, it is not a bad idea to prepare samples as randomly chosen "bags of words". If this option is switched on, the desired size of the sample should be indicated.' You can also read more about this issue at http://dh2010.cch.kcl.ac.uk/academic-programme/abstracts/papers/html/ab-744.html.
- *Output*. You choose here how you want to see your graph. You can choose multiple options; if you pick Onscreen you will see a pop-up of your graph. You probably also want to save it as a file, so you can open it later, use it in essays, or just compare it to others (the file will be automatically saved to your Stylo folder with a default name saying what it is). I'd recommend the PNG format, but you can use JPEG if you prefer.
  - Other options: *Colours* (this groups the files in the corpus folder based on the author part of the filename, and gives each group a different colour in the PCA and Bootstrap graphs),

Horizontal CA tree (some people prefer to have their cluster analysis horizontal, some vertical; you choose), and *Symbols* (replaces the filenames on the PCA graph with symbols grouped by author name).

- At any stage, clicking OK runs the analysis and closes the interface. Check the R window to see if the script is running or if it has finished (you can check if you are at the > prompt or not).

What to do?

1. Run a cluster analysis and a PCA analysis of the corpus in your corpus folder. Get used to how they work.
2. In the Stylo2 folder you will find some files which are novels of unknown authorship, but are by authors in the corpus you've already been given. Pick a sample of no more than three and add them to your corpus folder. Who do you think they are by? Remove these three and try a different sample, and another.
3. Run a bootstrap Consensus Tree analysis of the corpus including all the unknown files. Be careful not to have more than two or three iterations set (see above).
4. In the Stylo3 folder you will find a corpus of Henry James novels, some of which have had their dates removed. Can you estimate the missing dates? You may have to run the analysis a few times to get good MFW settings to group the known novels.

Note: if you run into errors at any point (eg you've accidentally changed a setting and can't get it to work properly again), then delete the file in your Stylo folder called 'config.txt'.