Quantitative Text Analysis
Exercise 1: Using RStudio and Importing Texts

21st July 2014, Essex Summer School

Kenneth Benoit and Paul Nulty

In today's lab we will work with RStudio and look at ways of loading texts into R for analysis.

RStudio is an environment for using R that combines panes for running interactive commands, running scripts, viewing variables interactively and viewing graphs and plots.

One of the first steps in the text analysis process is to load the texts from disk into a datastructure in memory where we can analyse and process them. The `quanteda` package has several functions for creating a corpus of texts which we will use today.

Instructions

1. **Working with RStudio and Packages**

   (a) Open RStudio and explore the four panes and the tabs available in each. Execute some commands at the prompt in the lower pane and also using the script pane with the `source` and `run` buttons. To see an example in the graphics pane, try the `plot` function (use `example(plot)`).

   (b) Download and install the `quanteda` module, following the instructions at `http://github.com/kbenoit/quanteda`. You may see some red 'warning' messages, but there should be no 'error' messages.

   (c) Explore the documentation with the `?` and `??` commands. The command `help (package=quanteda)` should display a list of links to documentation of `quanteda` functions. When entering commands in RStudio, use the TAB key to complete commands automatically or see suggestions, and use the arrow keys to navigate command history at the prompt.

2. **Making a corpus and corpus structure**

   (a) The simplest way to create a corpus is to use a vector of texts already present in R's memory. Some text and corpus objects are built into `quanteda`, and can be loaded with the data command. Type `data(amicusTexts)` to load a character vector containing 100 Supreme Court briefs that we will use in this example. Note how the variable appears in the top-right RStudio pane under the environment tab. Now type

   ```
   amiCorp <- corpusCreate(amicusTexts)
   ```

   to make a new corpus from these texts.

   (b) Use the `summary` command to inspect the corpus. The attributes of this kind of R object can be accessed with the $ symbol, and the vectors indexed with brackets — `amiCorp$attribs$texts[[1]]` will return the first text in the corpus. What metadata is associated with this corpus already?

3. **Other ways of creating a corpus**

   (a) In practice, you will rarely begin with a set of texts already neatly loaded into R character vectors. `quanteda` provides functions to create a corpus from a set of files on disk. To see an example of this, look at the example code on the `quanteda` page at https://github.com/kbenoit/quanteda#more-documentation. Copy this code into RStudio, and execute it line by line.

   (b) How are the petitioner (P) and respondent (R) attributes set using the text names? Look at the documentation for the `names` and `grep` commands.

   (c) What does the `iconv` command do? Take note of the platform-specific implementation details in the documentation.

   (d) The `grep` command is a powerful function in its own right, and is available at the terminal on all Unix-based systems. Examine the functions documented under `?grep` in RStudio.

   (e) Download the Irish budget speeches texts and save them to a local directory. The attributes of these files are represented explicitly in the filenames, separated by underscores: for our corpus of Irish budget speeches, the filename `2010_BUDGET_03_Joan_Burton_LAB.txt` tells us the year of the speech (2010), the type ("BUDGET"), a serial number (03), the first and last name of the speaker, and a party label ("LAB" for Labour).

   (f) Load these texts into a corpus object using the `corpusFromFilenames` function, supplying a vector of attribute labels that correspond with the elements of the filename, and the path to the folder with the texts. On Windows systems you might need to use double backslashes in the filepath. Examine the resulting corpus object with the `summary` command.