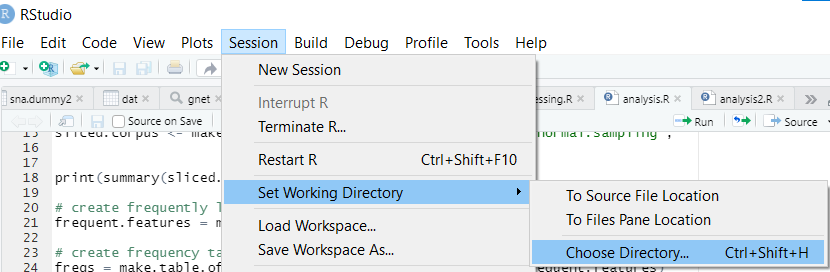
**Stylo demonstration**

## Download the Nahda dataset from GitHub

1. Download the Nahda dataset from GitHub: `git pull` in Desktop/DH23 folder
2. Show what is in the nahda dataset (in DH23/session\_8 folder): subfolder “corpus” containing 65 19th-C novels. Question: can the computer detect which novels were written by which author?
   1. Plain texts, no page numbers, etc.
   2. File names: authorID underscore title: first element before underscore is class of the text

## Prepare stylo for use:

1. Open Rstudio
2. Set working directory to the DH23/session\_8/stylo\_nahda\_novels:   
   use setwd() or menu: Session > Set Working Directory > Choose Directory…



1. Make sure stylo is installed: > library(stylo)  
   If it is not yet installed: install it using install.packages(stylo)
2. Show the help function > help(stylo)

## First run: use nahda dataset.

1. > stylo()
2. Popup will show up
   1. NB: popup may not be visible if Rstudio is full screen; look for the TK feather icon
3. Keep INPUT settings: to plain text
4. Set LANGUAGE settings: to Other (that is, UTF-8): defines how texts will be tokenized
   1. NB: if it is still set to other, reset it by removing the config file!
5. Click OK button to run the program with its default parameters:
6. RStudio console shows some preliminary output:
   1. First, the 65 texts in the corpus are sliced into tokens
   2. A frequency table is calculated
   3. Distances are calculated
7. Once the analysis is done, the plot will be shown in the Plots tab in Rstudio’s Files pane. Click the Zoom button to enlarge it
   1. This output is called a dendrogram, or a tree graph, which is used to visualize clusters of similarity.
   2. What can you see on first sight, even if you don’t know how this type of graph works? Clusters of colours are all very nicely clustered.  
      Where do these colours come from? They are defined by the first element of the file names.   
      The algorithm does not use the information from the filenames in its analysis. Stylo uses the file names only to colour the results, so you can clearly see if the categories you defined cluster together.
   3. You can see that the standard settings of the stylo function did a great job at clustering the books written by the same author.   
      To show that this is the case, let’s re-run the analysis but remove the author information from one text
   4. Some more information on how this graph works: you have to start reading the graph on the right. Vertical lines connect books together; the further to the right the vertical line is, the smaller the difference between those books is (we’ll talk in a minute about how it calculated this distance). The vertical line does not only connect these two nearest neighbours, but it also shows their similarity score. Together, they form our first cluster. The dendrogram then connects to the item (or cluster) that is most closely related to our first cluster, and so on.

## Second run: how did it do this?

1. Show parameters:
   1. FEATURES: define the features we want stylo to take into account:
      1. n-grams: sequences of a fixed number of words or characters (see presentation)
      2. MFW = most frequent words: by default, stylo uses 100 most frequent words in the corpus. Stylo creates a frequency list from all the texts, and saves it: “table\_with\_frequencies.txt”; you can open this in a spreadsheet (space-separated!)
         1. Exercise: take a look at the first 100 words; how would you describe these words? Function words: not very specific words, related to a specific author!
      3. VARIOUS: existing frequencies: reuse the frequency table instead of recalculating it
         1. Exercise: check the “Existing frequencies” parameter, and then experiment with different values for the MFW settings (make sure to set both Minimum and Maximum to the same number!). What does changing these parameters do to the graph
   2. STATISTICS tab: here you can set the different statistical methods that can be used; most of these (all except PCA) represent similarity between texts as a distance measure, and there are different ways to calculate these distances.
   3. SAMPLING: rather than using the entire text as a unit, you can take only smaller samples of it; we’ll use the default: not cutting up the text
   4. OUTPUT: here you can choose whether you want to:
      1. show the output graph on screen or save it as a file
      2. some settings on what the graph should look like
      3. what parts of the analysis you want stylo to save. Let’s check “Save frequencies”
2. What is going on behind the scenes? What did stylo do to recognize these authors?
   1. Let’s go into our stylo\_nahda\_novels folder. You’ll see stylo created a number of files that were not there before:
      1. Stylo\_config.txt: saves all the configurations you used for running stylo; this is very useful for making experiments repeatable
      2. Wordlist.txt: a text document containing all features extracted from the corpus, in order of frequency
      3. Table\_with\_frequencies.txt: remember that we checked the “Save frequencies” checkbox in the OUTPUT tab in the stylo GUI? This is its output: table with the frequency of every word in the wordlist per 100 words in each text
      4. A csv file that contains a network representation of the similarity data
   2. Move to slides for some explanation
3. Alternative outputs: PCA scatter plot: Principal components analysis takes a different approach: it does not calculate a combined distance value for each separate word in some way, but rather it looks at all separate word distributions and looks which two result in the biggest variance across texts, and uses these two to plot them in a scatter plot.   
   hard to explain what PCA really is; basically, it’s like trying to take a picture of a group of people by finding the vantage point from which you can see them best, without one person’s arm blocking someone else’s face.
4. Changing the settings: how many MFW to use? can easily result in cherry-picking. One solution is to let the machine cycle through settings, and combine the information from the different analyses into one output. Let’s try:
   1. FEATURES:
      1. MFW: set minimum to 50, maxim to 3000, increment to 150
      2. CULLING: set minimum to 0, maximum to 100, increment to 20
   2. STATISTICS: set to Consensus Tree
   3. This will take a while…
      1. First it’s tokenizing the corpus
      2. Then it’s running through all possible combinations of variables
   4. Finally it outputs a consensus tree in which the most closely related books are on the same branch.