Using Windows Command Prompt to Send Script To R Engine Directly

This method is useful when you have an R script that needs to be run automatically using task schedulers on your PC or server. It is NOT recommended while developing an R script.

1. Open windows command prompt (or the equivalent for your system)
2. Find the directory in which R is installed as well as the directory in which you saved your R script and tell R to run the script using the **Rscript.exe** application:

```
C:\> "C:\[My R directory]\bin\x64\RScript.exe" "C:\[My Script Dir
ectory]\helloworld.R"
```

Optionally you can tell it to put the output in a text file for you:

```
C:\> "C:\[My R directory]\bin\x64\RScript.exe" "C:\[My Script Dir
ectory]\helloworld.R" > "C:\[My Script Directory]\output.txt"
```

3. Make a change to your R script: Generate 1 million random numbers (`rand <- runif(1000000)`) and print the average (`text <- mean(rand)`).
4. Re-run the script from command prompt

   Tip: You can use the up arrow key to scroll through the most recent commands that you ran
5. In your R script create another variable `rand10` which generates 1 million uniformly distributed random numbers between 0 and 10 (do a Google search if you are unsure of the syntax) and print out the averages of `rand` and `rand10`

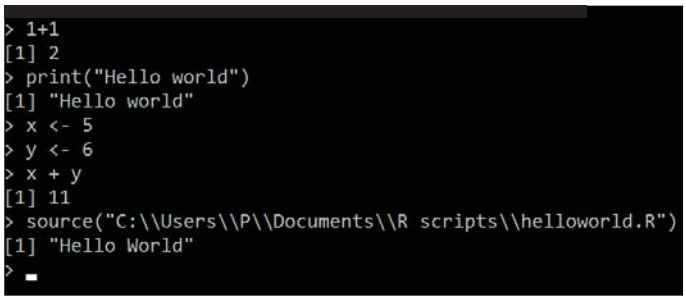
For more options you can enter the following command:

```
C:\> "C:\[My R directory]\bin\x64\RScript.exe" --help
```

If you want more detailed information about using R from the command line then you may find this documentation useful: https://cran.r-project.org/doc/manuals/R-intro.html#Invoking-R-from-the-command-line

Using the R Front End

1. Navigate to the directory where R is installed and open the application **R.exe**. This opens the *R console*, a direct window into the R engine.
2. Try running the following simple commands directly in the console:
   - `1+1`
   - `print("Hello world")`
   - `x <- 5`
   - `y <- 6`
   - `x + y`



3. Run your R script using the following command:

```
source("C:/[My Script Directory]/helloworld.R") # Note the use of forward slashes
```

4. Because we are using the R front end now, an *environment* has been created and is still active. To see the objects in your environment use the following command:

- `ls()` - type `?ls` for online help on the command

5. Find the maximum (`max()`) and minimum (`min()`) values of the `rand10` array by typing the commands directly into the R window
6. Store the maximum and minimum values into variables (e.g. `mn <- min(rand10)`)
7. Modify your script to make `rand` be 1 million Normal(0,1) random variables and `rand10` be 1 million Normal(0,10) random variables and to print the difference between the maximums of these two arrays (for help on generating normally distributed random variables, run the command `?rnorm`)