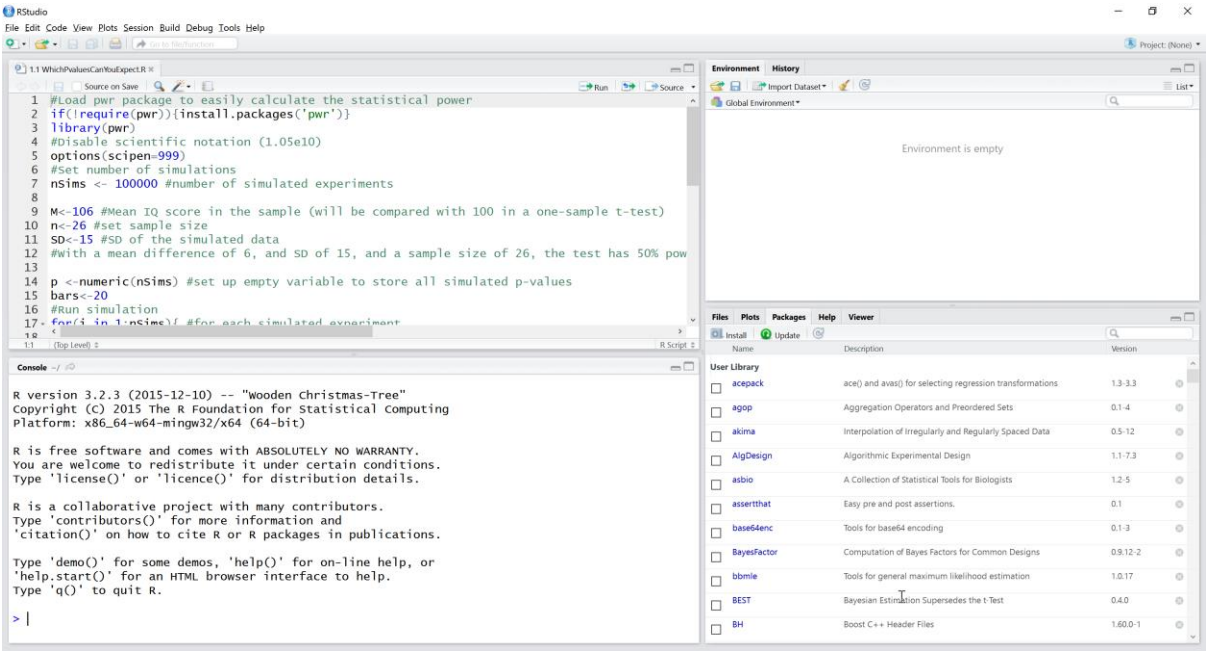
Running R code

In this course we will examine many statistical concepts through simulations or calculations in the free software R. You don't need to know how to program in R – you just need to know how to run code. Here's a detailed explanation for people who have never used R.

First install R. Then, **install R studio**.

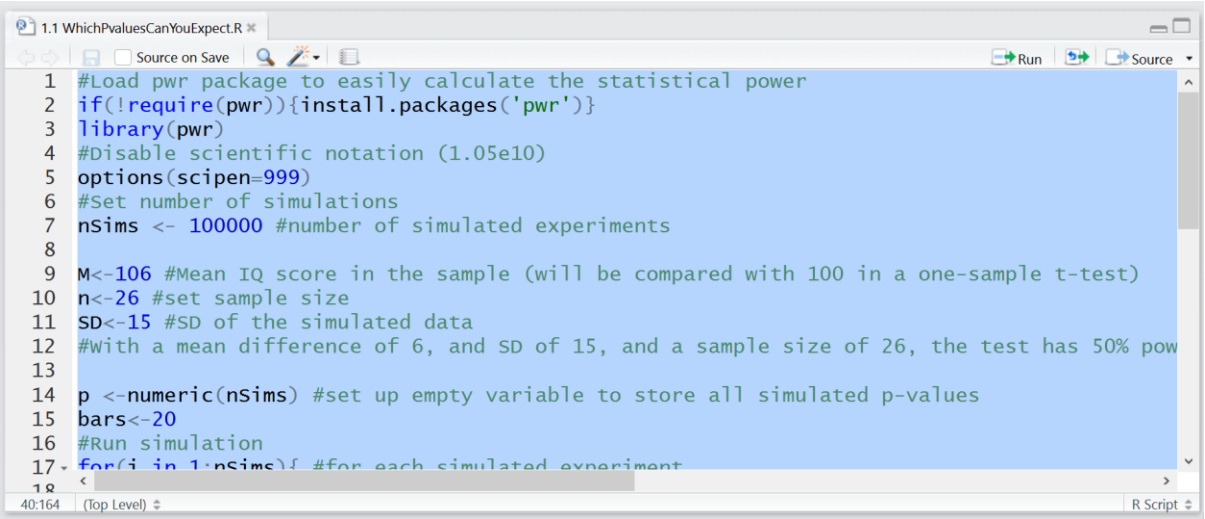
Then, double-click the R script 1.1 WhichPvaluesCanYouExpect.R. It should open in R Studio. If it doesn't, open Rstudio, go to File>Open File, and open the script. You should see a screen like the screenshot below.



You see four sections in the four corners. Top left, you see the R script. In this window, you type new code, or change existing code. In the bottom left you see the console window, which gives output related to the code you are running, and where the results from calculations will appear. In the top right there is a window which will inform you about variables that have been defined – it's useful for coding, but you won't need it just to run code. In the bottom right, there is a window with 5 tabs. The only tab we need is the Plots tab, where plots that are created by R scripts will appear.

To interact with the R script, you need to click inside the top-left window. A blinking cursor should be visible.

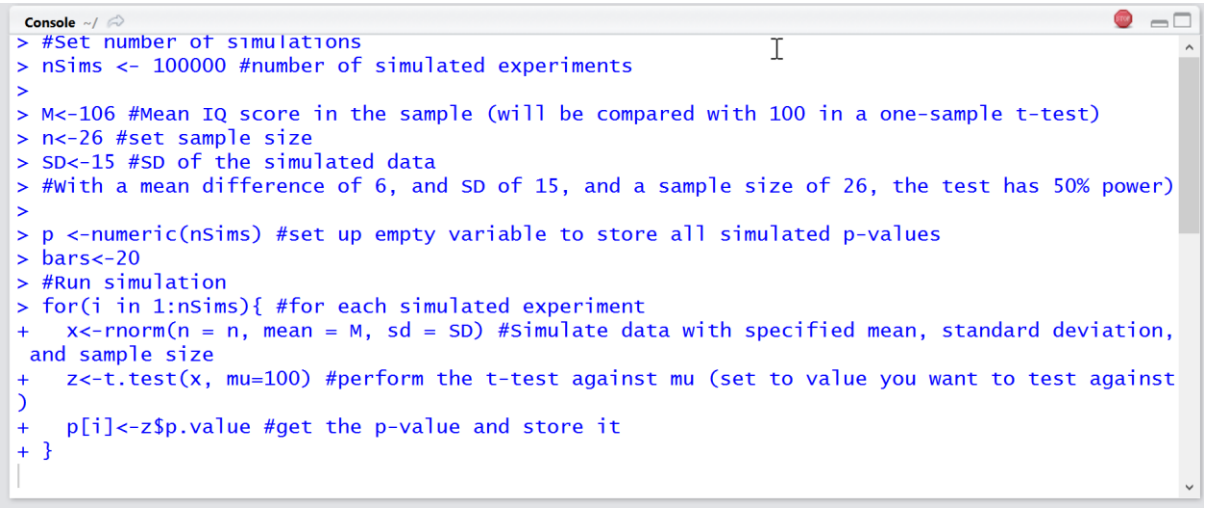
If you want to run all the code in a script, you have to select all lines of the script, and run the code. The easiest way to do this is through keyboard shortcuts. Press **CTRL+A** to select all the code. All lines in the script window should be light blue:



Then, you can use the keyboard shortcut **CTRL+ENTER** to run the selected code. If you prefer to use the mouse, you can click on the 

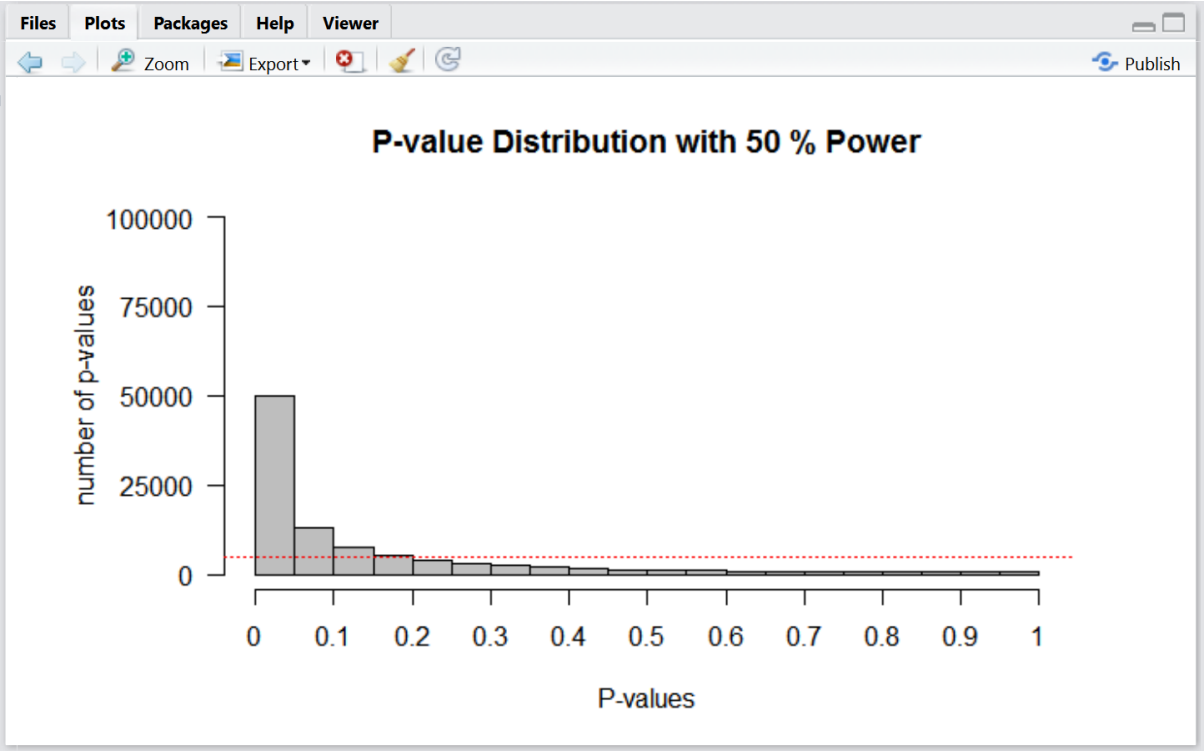
icon in the upper bar of the script window.

Look at the console window:



You'll see that the code we ran is visible in a blue font – if any output from calculations was visible, it would be in a black font. We are running 100.000 simulations, which takes some time. When R is busy running code, you can see a small red stop sign in the upper bar of the console window:  . If the code takes too long, you can stop R running the code by clicking the stop sign. For now, wait patiently.

After a few seconds, the code has run, and a plot has appeared in the bottom right window:



Let's go back to the script window. Sometimes, you might want to select only a few lines of code. Notice how each line in the script window has a number on the left of the code, starting at 1. This makes it easy to refer to lines in the code. Scroll down to line 24 of the code. Put your cursor at the beginning of line 24, and select only this line. I like to use the keyboard shortcut **SHIFT+DOWNARROW**. Now, a single line is selected.

```
22
23 #Check power by summing significant p-values and dividing by number of simulations
24 (sum(p < 0.05)/nSims) #power
25 #Calculate power formally by power analysis
26 power<-pwr.t.test(d=(M-100)/SD, n=n,sig.level=0.05,type="one.sample",alternative="t
```

Run this single line by CTRL+ENTER. In this single line, the statistical power of our test is calculated, by taking the sum of all simulated p-values smaller than 0.05, and dividing this number by the number of simulations.

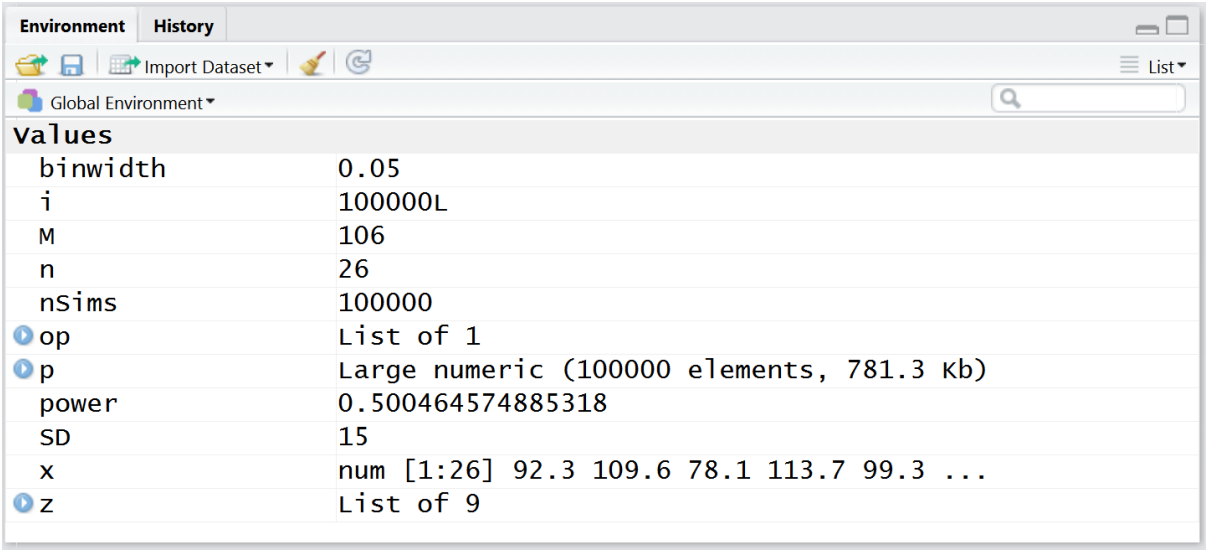
Look at the Console window in the bottom left. We see the code we ran (in blue), and the result (in black) – but remember that we are randomly simulating data! You will see the same blue text, but the outcome of your simulation will be slightly different from the black number!

```
> (sum(p < 0.05)/nSims) #power
[1] 0.50086
>
```

Repeat the previous step, re-run the entire simulation, and run the code at line 24 once more – you will get a slightly different number. To get stable results, you need to increase the number of simulations, but you also have to wait longer. 100000 simulations gives accurate enough results for current purposes.

Note that this appeared the previous time we ran the code, but because there was a lot of output in the Console window, we had to scroll up to see this code. Remember that when the console window produces a lot of output, you can simply scroll up to see all output.

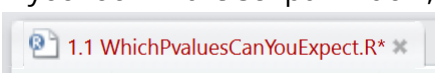
It's easy to change a small part of the code by just typing and replacing. In line 10, you will see n <- 26. You can read the "<-" combination as 'is assigned the value' or 'becomes'. So the variable n (a variable name consisting of just a single letter) is assigned the numerical value 26. You can also see this in the top right window where n has the value 26:



Change the n<-26 to n<-50.

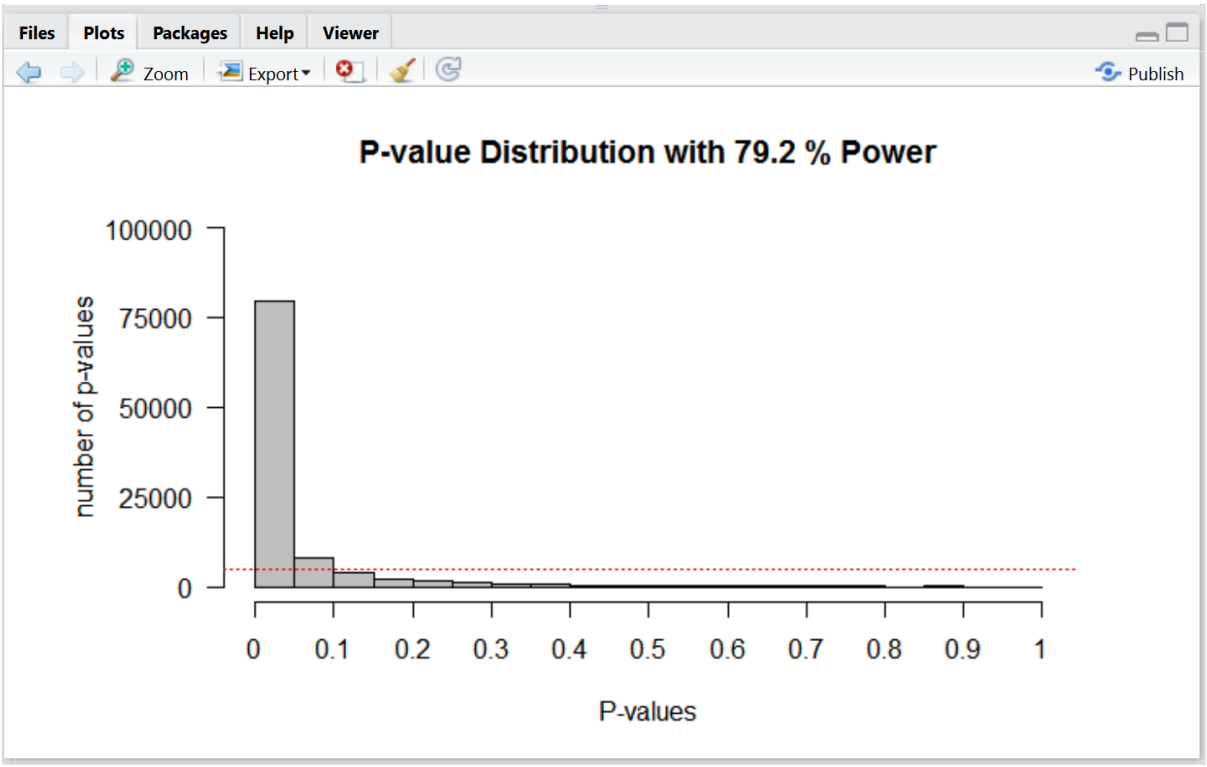
```
10  n<-50 #set sample size
```

If you look in the script window, the name of the script in the top has colored red:



to indicate the code has changed. If you save the code, you will need to decide if you want to save or don't save these changes.

Run the entire code again. You will see a new plot:



If you want to see the old plot, in the upper bar of the plot tab, you can click on back and forward arrows () to scroll through the plots you have generated.

That's all you need to know to run the code for all assignments in this course. You might still get stuck now and then. It happens when programming. Remember my favorite quote from the book 'Zen and the Art of Motorcycle Maintenance': *Stuckness shouldn't be avoided. It's the psychic predecessor of all real understanding.*

Good luck!

© Daniel Lakens, 2016. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 License.