Practical Exercise 2

Niels Richard Hansen

31 Aug 2016

This exercise is a continuation of Practical Exercise 1. It is assumed that you have a working `myBreaks` function.

Problem 2.1

Write a function called `myHist`, which takes a single argument `h` and plots a histogram of `log(F12)` using your function to compute breakpoints. Extend the implementation so that any additional argument specified when calling `myHist` is passed on to `hist`.

Investigate and explain what happens when executing the following function calls.

```
myHist()
myHist(h = 5, freq = TRUE)
myHist(h = 0)
```

Finally, what happens if you remove that data from the global environment and call `myHist` subsequently?

Problem 2.2

What is the environment of `myHist`? Change it to a new environment, and assign (using the function `assign`) the data to a variable with an appropriate name in that environment. Once this is done, check what now happens when calling `myHist` after the data is removed from the global environment.

Problem 2.3

Write a *function factory* that takes an argument `x` (the data) and returns a function, where the returned function takes an argument `h` (just as `myHist`) and plots a histogram (just as `myHist`).

What is the environment of the function created by the function factory? What is in the environment? Does it have any effect when calling the function whether the data is altered or removed from the global environment?

Problem 2.4

Execute the following function call:

```
tmp <- myHist(10, plot = FALSE)
```

What is the type and class of `tmp`? What happens when `plot(tmp, col = "red")` is executed? How can you find help on what `plot` does with an object of this class? Specifically, how do you find the documentation for the argument `col`, which is not an argument of `plot`?