Programming (ERIM)

Lecture 8: Anonymous functions, function references, and more

Tommi Tervonen

Econometric Institute, Erasmus School of Economics

Anonymous functions: definition

- Anonymous functions are functions that are not bound to any identifier, i.e. name

- R:

```
(function(x) { x * x })(5)
```

- Matlab:

```
(@(x) x * x)(5)
```

Function references

- When calling a named function, the names represents reference to the actual execution code
- Similarly, anonymous functions define references that are often immediately passed somewhere
- Matlab's `@` is similar to R's `function`:

```
f = @(x, y) x + y;
f(2, 3)
```

- In R, you can see the source code of functions written in R by executing the function without arguments in command-line
- Same in Matlab for functions constructed with `@`

Using function references

- Function references allows to parameterize methods with functionality that is left free for definition
- This is similar to implementing interfaces in strongly-typed OOP languages

```
apply(m, 1, function(x) {
  ## do something with each row ##
})
```

Closures

- Functions can return a function (!)
- Closures are functions including an execution frame (some variables bound)

```
addSome <- function(x) { function(y){x + y} }
addTwo <- addSome(2)
addTwo(3) # returns 5
```

- Example: spline smoothing basis functions

Tools of the trade

- Console (execution environment)
- Good text editor
- Unit testing framework
- Source control (git)
- Source repository (github)
- Build system (GNU Make, Maven)
- Continuous integration server (Hudson)

Additional topics

- Alternative (better?) plotting system: `ggplot2`
- Data conversion: regular expressions (`gsub`)
- Optimizing for speed: `.Call` (hitandrun example - see `https://github.com/gertvv/hitandrun/blob/master/hitandrun/src/har.c` and `https://github.com/gertvv/hitandrun/blob/master/hitandrun/R/sample.R`)