Programming (ERIM)

Lecture 3: Subroutines and scoping

Tommi Tervonen

Econometric Institute, Erasmus School of Economics

Code style and documentation

- Use descriptive variable names
- Single character / short names appropriate for loop counters and for implementing mathematical equations (add documentation where to read the actual equations)
- Document longer / unclear parts of code in more detail
- Stick to “standard” layout

```
if x < 5
  if y > 4
    dosomething();
  else
    dosomethingelse();
  end
end
```

Rule #1: Never duplicate code

```r
res <- array(dim=2)
elems <- c(0, 4, 5, 3)
elems2 <- c(1, 2, 7, 8, 10, 3)

sum1 <- 0
for (x in elems) {
  sum1 <- sum1 + x
}
res[1] <- sum1 / length(elems)

sum2 <- 0
for (x in elems2) {
  sum2 <- sum2 + x
}
res[2] <- sum2 / length(elems2)
```

```r
compute.avg <- function(x) {
  temp <- 0
  for (elem in x) {
    temp <- temp + elem
  }
  temp / length(x) # or return(temp / length(x))
}

elems <- c(0, 4, 5, 3)
elems2 <- c(1, 2, 7, 8, 10, 3)

res <- array(data=c(compute.avg(elems),
                    compute.avg(elems2)))
compute.avg <- function(x) {
  res <- array(0, dim=nrow(x))

  for (i in 1:nrow(x)) {
    for (j in 1:ncol(x)) {
      res[i] <- res[i] + x[i,j]
    }
  }

  res / ncol(x) # divide all elements
}

elems <- matrix(c(0, 4, 5, 3, 0, 0,
                  1, 2, 7, 8, 10, 3), ncol=6,
                  byrow=TRUE)

res <- compute.avg(elems)
```

Subroutines (methods)

- Re-usable functionality should be **abstracted** into methods
- Variability in the functionality defined with method parameters
- Do not try to make too general methods
- Methods are divided into functions and procedures (but R and Matlab have mostly functions!) - more on the following lecture
- Use methods also to split code into pieces of each max. 1 screen

Method signature

```
function [r1, r2] = mynewfunction(p1, p2, p3)
  ...
end

[a, b] = mynewfunction(0.3, 0.2, 0.0)
```

```
mynewfunction <- function(p1, p2, p3=0.0) {
  ...
  list(r1=...,r2=...)
}

res <- mynewfunction(0.3, p2=0.2)
print(res$r1)
print(res$r2)
```

File organization

- Matlab: files in the current working directory are in the current namespace
  - Script files can be executed with their name (no extension)
  - Function files have the first defined function visible (!)
- R: include functions into the current namespace by executing them, e.g. `source('myfuncs.R')`

Variable scoping

- In general, variables are visible within the block where they are introduced after the introduction
- Variable visibility and lifetime = scope
- In Matlab and R, variable scope is on method level

```
mystuff <- function(x) {
  y <- x + 2
  x <- y * 2
  x
}
x <- 3
x <- mystuff(x)
print(y) # error
```

Pre-knowledge?

- Design of computational tests
- Seeding?
- Reproducibility?
- Availability of code?
- DOI: 10.1038/nature10836