Subsetting

Gerardo Ferrara

Master in Economics and Complexity, Collegio Carlo Alberto

Fall 2013

Subsetting

There are a number of operators that can be used to extract subsets of R objects:

- **[ ]** always returns an object of the same class as the original; can be used to select more than one element.
- **[[ ]]** is used to extract elements of a list or a data frame; it can only be used to extract a single element and the class of the returned object will not necessarily be a list or data frame.
- $ is used to extract elements of a list or data frame by name.

Subsetting a Vector

We use the same indexing rules for character vectors that you use for numeric vectors (or for vectors of any type):

```
> x <-c("a","b","e","f","a")
> x <-x[1]
 [1] "a"
> x <-x[x > "a"]
 [1] "b" "e" "f"
> p <-x > "a"
> p
 [1] "FALSE" "TRUE" "TRUE" "TRUE" "FALSE" > x[p]
 [1] "b" "e" "f"
```

Subsetting a Matrix

Matrices can be subsetted in the usual way with *(i,j)* type indices:

```
> m <-matrix(1:6, 2, 3)
> m [1, 2]
  [1]  3
```

Indices can also be missing:

```
> m [1, ]
  [1] 1  3  5
> m [ , 2]
  [1] 3  4
```

Subsetting a Matrix (cont.)

By default, subsetting a single column or a single row will give you a vector, not a matrix. This behavior can be turned off by setting **drop = FALSE** as follow:

```
> m <-matrix(1:6, 2, 3)
> m [1, ]
  [1] 1 3 5
> m [1, , drop = FALSE]
     [,1] [,2] [,3]
[1,]  1    2    5
```

Subsetting a List

```
> y <- list(foot = 1:4, height = 0.45)
> y[1]
  $foot
  [1] 1 2 3 4        #The result is a vector
> y[[1]]
  [1] 1 2 3 4
> y$height
  [1] 0.45
> y["height"]
  [1] 0.45
> y[["height"]]      #Extracting one element of a list
  $height
  [1] 0.45
> y[c(1,2)]          #Extracting multiple elements of a list
  $foot
  [1] 1 2 3 4
  $height
  [1] 0.45
```