Data Handling: Import, Cleaning and Visualisation

Lecture 5: Programming with Data

Prof. Dr. Ulrich Matter

18/10/2018

Programming: Murrell chapter 9.11 - Go through key concepts (maybe extend a bit, take some from cs50) - Go through the case study (maybe update needed here)

0.1 Working with Data in R: Data Structures and Indeces

(Practical part is important here, they should get an overview, should recall data structures in objects etc.)

0.1.1 Vectors and Lists

```
# A vector containing numeric (or integer) values
numeric_vector <- 10:20
numeric_vector[2]
```

```
## [1] 11
```

```
numeric_vector[2:5]
```

```
## [1] 11 12 13 14
```

```
# A string vector ('a vector containing text')
string_vector <- c("a", "b", "c")
string_vector[-3]
```

```
## [1] "a" "b"
```

```
# Lists
# A list can contain different types of elements, for example a numeric vector and a string_vector
mylist <- list(numbers = numeric_vector, letters = string_vector)
mylist
```

```
## $numbers
##  [1] 10 11 12 13 14 15 16 17 18 19 20
##
## $letters
## [1] "a" "b" "c"
```

```
# We can access the elements of a list in various ways
# with the element's name
mylist$numbers
```

```
##  [1] 10 11 12 13 14 15 16 17 18 19 20
```

```
mylist["numbers"]
```

```
## $numbers
##  [1] 10 11 12 13 14 15 16 17 18 19 20
```

```r
# via the index
mylist[1]
```

```
## $numbers
##  [1] 10 11 12 13 14 15 16 17 18 19 20
```

```r
# with [[]] we can access directly the content of the element
mylist[[1]]
```

```
##  [1] 10 11 12 13 14 15 16 17 18 19 20
```

```r
# lists can also be nested (list of lists of lists....)
mynestedlist <- list(a = mylist, b = 1:5)
```

0.1.2 Matrices and Data Frames

```r
# matrices
mymatrix <- matrix(numeric_vector, nrow = 4)
```

```
## Warning in matrix(numeric_vector, nrow = 4): data length [11] is not a sub-multiple or multiple of
## the number of rows [4]
```

```r
# get the second row
mymatrix[2,]
```

```
## [1] 11 15 19
```

```r
# get the first two columns
mymatrix[, 1:2]
```

```
##      [,1] [,2]
## [1,]   10   14
## [2,]   11   15
## [3,]   12   16
## [4,]   13   17
```

```r
# data frames ("lists as columns")
mydf <- data.frame(Name = c("Alice", "Betty", "Claire"), Age = c(20, 30, 45))
mydf
```

```
##     Name Age
## 1  Alice  20
## 2  Betty  30
## 3 Claire  45
```

```r
# select the age column
mydf$Age
```

```
## [1] 20 30 45
```

```r
mydf[, "Age"]
```

```
## [1] 20 30 45
```

```r
mydf[, 2]
```

```
## [1] 20 30 45
```

```r
# select the second row
mydf[2,]
```

```
##    Name Age
## 2 Betty  30
```

0.1.3 Classes and Data Structure

```r
# have a look at what kind of object you are dealing with
class(mydf)
```

```
## [1] "data.frame"
```

```r
class(mymatrix)
```

```
## [1] "matrix"
```

```r
# have a closer look at the data structure
str(mydf)
```

```
## 'data.frame':    3 obs. of  2 variables:
##  $ Name: Factor w/ 3 levels "Alice","Betty",..: 1 2 3
##  $ Age : num  20 30 45
```

1 References