Tutorial 1 | Introduction to R

R objects and attributes

R Objects

There are five basic(atomic) classes of objects:

- character
- numeric("1" is just a numeric object, but typing "1L" makes it specifically integer/"Inf" - infinity/"NaN" - not a number or a missing number)
- integer
- complex
- logical

The most basic object is a vector. The vector could only contain items of one class. However, there are **lists** which are just like vectors, but can contain multiple objects of different classes

Attributes

Attributes of an object could be accessed using attributes()

- names, dimnames
- dimensions
- class
- length
- other user-defined attributes

```
x <- c(2,3,4,5)
y <- vector("numeric", length=10)
print(x)
```

```
## [1] 2 3 4 5
```

```
print(y)
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0
```

Explicit coercion

Objects can be explicitly coerced from one type to another using as.* functions.

```r
a <- as.character(x)
class(a)
```

```
## [1] "character"
```

```r
print(a)
```

```
## [1] "2" "3" "4" "5"
```

Lists

```r
student <- list(first_name = "John", last_name = "Holmes",age = 19, enrolled = T)
print(student)
```

```
## $first_name
## [1] "John"
##
## $last_name
## [1] "Holmes"
##
## $age
## [1] 19
##
## $enrolled
## [1] TRUE
```

Matrices

Matrices are basically vectors with the dimension attribute. Matrices are conctructed column-wise, meaning entries are starting from the upper left corner of the matrix and running down/"filling in" the columns

```r
m <- matrix(1:6,nrow = 3, ncol = 2)
print(m)
```

```
##      [,1] [,2]
## [1,]    1    4
## [2,]    2    5
## [3,]    3    6
```

```r
dim(m)
```

```
## [1] 3 2
```

Matrices can also be directly created from vectors by assigning dim attributes. For example the vector x we created earlier could be turned into matrix:

```r
dim(x) <- c(2,2)
print(x)
```

```
##      [,1] [,2]
## [1,]    2    4
## [2,]    3    5
```

Matrices could also be created using **column-binding** or **row-binding**:

```r
c1 <- 1:4
c2 <- 5:8
cf <- cbind(c1,c2)
cf
```

```
##      c1 c2
## [1,]  1  5
## [2,]  2  6
## [3,]  3  7
## [4,]  4  8
```

```r
rf <- rbind(c1,c2)
rf
```

```
##    [,1] [,2] [,3] [,4]
## c1    1    2    3    4
## c2    5    6    7    8
```

Factors

Factor is a type of data used to represent categorical data. Factors can be unordered or ordered. **Ex. "Male"/"Female" (unordered)** You can of think of the factor as an integer vector, where each vector has a label.

```r
t <- factor(c("yes","yes","no","yes"),
levels = c("yes","no"))
## this is done in order to determine the baseline level of the factor(in basic configuration "no" woul
print(t)
```

```
## [1] yes yes no  yes
## Levels: yes no
```

```r
table(t) ##prints out the table describing factors
```

```
## t
## yes  no
##   3   1
```

```r
unclass(t) ##prints out details of the factor vector
```

```
## [1] 1 1 2 1
## attr(,"levels")
## [1] "yes" "no"
```

Missing values

NaN - undefined mathematical operations, NA - everything that is missing. To test this we use functions is.na(), is.nan()

```r
e <- c(1,4,7,NA,0)
is.na(e)
```

```
## [1] FALSE FALSE FALSE  TRUE FALSE
```

```r
e <- c(1,4,NaN,NA,0)
is.na(e)
```

```
## [1] FALSE FALSE  TRUE  TRUE FALSE
```

```r
is.nan(e)
```

```
## [1] FALSE FALSE  TRUE FALSE FALSE
```

Data frames

Data frames are used to store tabular data. Unlike matrices, data frames can store different classes of objects in each column(just like lists). Data frames also have an attribute called row.names() Data frames are ususally created by calling read.csv(), read.table() or data.frame() They also can be converted into matrix using data.matrix().

```r
b <- data.frame(ID=1:3, Names =c("John","Marry","Elena"),Passed=c(T,F,T))
b
```

```
##   ID Names Passed
## 1  1  John   TRUE
## 2  2 Marry  FALSE
## 3  3 Elena   TRUE
```

```r
nrow(b)
```

```
## [1] 3
```

```r
ncol(b)
```

```
## [1] 3
```

Names attribute

In order to increase the redability of the code and make objects self-describing the names attribute could be added.

```
names(c1) <- c("col1","col2","col3","col4")
c1
```

```
## col1 col2 col3 col4
##    1    2    3    4
```

```
dimnames(x) <- list(c("row1","row2"),c("col1","col2")) ##names for matrices
x
```

```
##      col1 col2
## row1    2    4
## row2    3    5
```