Intro to Data Visualization and Statistics in R

Session #3 extra

Genome Institute of Singapore

10th October 2019

Data types | Overview

Let's breeze through some common basic data types in R.

- Scalar
- Vector
- Factor
- Matrix
- **Data Frames**
- Lists

Data types | Vectors

Definition: Collection of data elements of the same type

Creation

```
x <- c("e", "h", "l", "o", "w", "r", "d")
y <- c(2, 1, 3, 3, 4)
```

Data types | Vectors

Definition: Collection of data elements of the same type

Creation

```r
x <- c("e", "h", "l", "o", "w", "r", "d")
y <- c(2, 1, 3, 3, 4)
```

Extraction

```r
x[1]        # single element
```

```
## [1] "e"
```

```r
x[c(1,4)]   # multiple element
```

```
## [1] "e" "o"
```

```r
x[y]        # multiple element using another vector
```

Data types | Vectors | Some useful things

```r
x <- c(9, 1, 5, 7, 3, 2, 8, 6, 10, 4)
length(x)
```

```
## [1] 10
```

```r
sort(x)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```r
mean(x)
```

```
## [1] 5.5
```

```r
x < mean(x)
```

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

```r
table( x < mean(x) )
```

```
##
```

Data types | Vectors | Set operators

Set operators can be useful when finding elements that are common/unique to each set. Example:

```
left  <- c("A", "A", "B", "C")
right <- c("B", "C", "C", "C", "E", "D")
setdiff(left, right)
```

```
## [1] "A"
```

```
setdiff(right, left)
```

```
## [1] "E" "D"
```

```
intersect(left, right) # same as intersect(right, left)
```

```
## [1] "B" "C"
```

```
union(left, right)     # same as union(right, left)
```

Data types | Vectors | For loops

```r
num <- 1:10

for(x in num){
  cat("Input:", x, "Square:", x^2, "\n")
}
```

```
## Input: 1 Square: 1
## Input: 2 Square: 4
## Input: 3 Square: 9
## Input: 4 Square: 16
## Input: 5 Square: 25
## Input: 6 Square: 36
## Input: 7 Square: 49
## Input: 8 Square: 64
## Input: 9 Square: 81
## Input: 10 Square: 100
```

The ":" is range operator with increments of 1. Also see `seq()` for

Data types | Factors

```
expr <- c("Low", "Low", "High", "Medium", "Low")
table(expr)
```

```
## expr
##   High    Low Medium
##      1      3      1
```

The table lists the alphabetically which is messy. We can override this by converting it to a factor using levels.

Data types | Factors

```
expr <- c("Low", "Low", "High", "Medium", "Low")
table(expr)
```

```
## expr
##   High    Low Medium
##      1      3      1
```

The table lists the alphabetically which is messy. We can override this by converting it to a factor using levels.

```
levs <- c("Absent", "Low", "Medium", "High")
expr <- factor(expr, levels=levs)
table(expr)
```

```
## expr
## Absent    Low Medium   High
##      0      3      1      1
```

Note: We also added a level for "Absent" for generalization.

Data types | Matrix

A 2-dimensional object with rows and columns.

Each column is a vector.

	$Sample_1$	$Sample_2$	$Sample_3$	$Sample_4$	$Sample_5$
$Gene_1$	12.5	8.1	11.7	8.1	11.5
$Gene_2$	2.9	3.1	3.7	3.2	2.8
...	...	...	...	...	...
$Gene_N$	7.6	7.8	7.7	7.7	7.9

All entries must be numeric ***or*** character, but not a mix of both.

Data types | Data Frames

Same 2-dimensional structure as a matrix but it can be a mix of character columns and numeric columns.

	Age	Sex	Status
Subject_1	55	Male	Healthy
Subject_2	49	Female	Cancer
Subject_3	80	Female	Health
...	...	...	...

Data import functions store in a data frame e.g. `read_excel()`, `read_csv()`, `read_delim()`

Data types | List

A list contain elements of other data types. Here is how to create one:

```
experiment <- list(

  Requester = "Meng How, Tan",

  Library_Kit = "Illumina TruSeq",

  Read = c(Length="150", type="paired-end"),

  out = data.frame(ID      = c("MUX1", "MUX2", "MUX3"),
                   Machine = c("HS08", "HS08", "HS06"),
                   Date    = c("1Jun", "2Jun", "1Jun"))
)
```

Data types | List

```
experiment
```

```
## $Requester
## [1] "Meng How, Tan"
##
## $Library_Kit
## [1] "Illumina TruSeq"
##
## $Read
##       Length          type
##        "150" "paired-end"
##
## $out
##     ID Machine Date
## 1 MUX1    HS08 1Jun
## 2 MUX2    HS08 2Jun
## 3 MUX3    HS06 1Jun
```