Biological resources - lect 4 - lab 1

Francesco Penasa

March 4, 2020

1 Matrix

rectangular table of data of the same type

```
# matrix (vector, nrow, ncol) [row, colmn] dimnames == names of row and column
matrix(1:9, nrow = 3, ncol =3, dimnames = list(c(‘‘X’’, ‘‘Y’’, ‘‘Z’’), c(‘‘A’’, ‘‘B’’

# update names
colnames(x) = c(‘‘a’’,‘‘b’’,‘‘c’’)
rownames(x) = c(‘‘a’’,‘‘b’’,‘‘c’’)

# update rows or columns
cbind(c(1,2,3),c(4,5,6))
rbind(c(1,2,3),c(4,5,6))

# change the dim of a vector of a matrix
dim(x) <- c(2,3)
```

```
m <- cbind(c(1,2,3), c(1,2,3), c(1,2,3))
v <- c(2,4,6)

m*2    # scalar product
m*v    # scalar product
m*m    # scalar product
v*m    # scalar product
m%*%v # dot product
m%*%m # dot product
```

2 Lists

ordered collection of data or arbitrary types

```
doe <- list("john", 28, F)
doe

doe <- list(name="john", age=28, married=F)
```

```
doe$name

unlist(doe)
doe
```

3 Data Frames

data frame, excel table. Rectangular table with rows and columns BUT data within each column has the same type. but different columns may have different types.

```
d <- data.frame(n1 = c(3,2,1), n2=c("a","b", "c"), n3=c(T,T,F))
d
class(d)
typeof(d)

# to access use parenthesis, using index or the name
d[3,2]
d["1", "n2"] # first row and second column
d[,"n2"] # just the second column

# to keep the structure, be careful about the DROP!
d[,2, drop=FALSE]
d["1" , , drop=FALSE]

# more complex subsetting
m[1:2, c(1,3)]
m[which(m[,1] > 1), ]

# update values through subsetting
d [1,1] <- 5
d [1,] <- c(5, "S", FALSE)

# access list elem
doe[1] = "you"
doe$name = "yourmother"
```

4 Levels

Careful about levels!

1. *character* is your basic string data structure.

2. *factor* is something important for statistics.

5 Program time

5.1 Branching

```
x <- c(1,2,3)
if (any(x>2)){
        x <- x+1
} else {
        x <- x+2
}
```

5.2 Loops

```
for(i in 1:10){
        print(i*i)
}

i <- 1
while(i<=5){
        print(i*i)
        i = i+ sqrt(i)
}
```

```
# apply, sapply, lapply
x <- matrix(1:9, nrow = 3, ncol =3, dimnames = list(c(”X”, ”Y”, ”Z”), c(”A”, ”B”, ”C”

# apply(arr, margin, fct)

# sum all rows
apply(x, 1, sum)

# sum all columns
apply(x, 2, sum)

# lapply: generization for lists
lapply


# sapply: create a data structure that it is the simplest as possible
li <- list(”t”, ”s”, ”q”)
sapply(li, toupper)
li

fct <- function(x){ return(c(x, x*x, x*x*x))}
sapply(1:5, fct)
```

6 Import data

```
x <− read.delim(”filename.txt”)
write.table(x, file=”s”)

# read csv
csv <− read.csv(file)
```

String manipulation

```
lines <− readLines(”Table.txt”)
strsplit(lines[[1]], ”\t”)

writeLines(x, ”~/Test.txt”, sep=”\t”)
```

7 Storing data

```
save(x, file=”x.RData”)
```

8 Plot

plot()

```
x <− seq(−pi, pi, −.1)
plot(x, sin(x))

x <− c(1, 3, 5, 7, 10, 4, 1)
barplot(x)

x <− c(1, 3, 5, 7, 10, 4, 1)
y <− c(1, 3, 5, 7, 10, 4, 1)+3
boxplot(x, y)

# scatter plot
plot(x,y)

# ggplot to see more
```

9 Basic statistic in R

Qualitative data

```
# import package
library(MASS)
# watch dataset head
head(painters)
```

```
# project the distribution of elements
table(painters$School)

pie(table(painters$School))


# complicate this, count the occurencies of the compose the data
table(painters$School, painters$Composition)
```

Quantitative

```
library(MASS)
head(faithful, n=3)

duration = faithful$eruption
stuff...


plot(faithful$eruptions, faithful$waiting)
```

Statistic tools

1. mean
2. median
3. quartile
4. ...
5. ...

```
# perform correlation coefficient
cor.test(faithful$eruptions, faithful$waiting)
```

Probability distributions

```
# normal distribution
vals <- rnorm(1200, mean=72, sd=15.2)
hist(vals)
```

help help(function) ? function