

# Σύντομη Εισαγωγή στο R

Φώτης Σιάνης

<http://users.uoa.gr/~fsiannis/teaching.html>

Φεβρουάριος 2008

## 1. Αρχικά

Το πακέτο R θεωρείται μια εφαρμογή της γλώσσας προγραμματισμού S, που δημιουργήθηκε στα Bell Laboratories και αποτελεί επίσης την βάση του (εμπορικού) πακέτου S-Plus.

### a. Βοήθεια:

> help()	Ανοίγει τον help browser (γενικά)
> help(...)	Συγκεκριμένα για ότι είναι στην
> ?...	παρένθεση ή μετα το ?
> help.start()	Μέγα-βοήθεια (νέο παράθυρο)

### b. Μπορούμε να κάνουμε save/load Workspace από το drop down menu, ακολουθώντας το path:

"File -> Save/Load Workspace".

### c.

<- ή =	Συμβολισμός ανάθεσης
ls()	Λίστα με περιεχόμενα στο χώρο εργασίας
q()	Τερματισμός
;	Πάνω από μία 'εντολές' μπορούν να γραφούν στην ίδια γραμμή με το ';' ως διαχωρητικό.
#	Στην αρχή του σχολιασμού (ότι ακολουθεί μετά το # δεν λαμβάνεται υπ' οψιν στους υπολογισμούς)
search()	Λίστα με τα directories που υπάρχουν στον χώρο εργασίας

```
> ls()
[1] "a"      "b"      "c"      "geo"    "long"   "vec"    "vec.c"  "vec.m"  "vec.n"
[10] "w"      "x"      "y"      "Y"      "z"
```

```
> search()
[1] ".GlobalEnv"      "faithful"        "esoph"
[4] "package:stats"   "package:graphics" "package:grDevices"
[7] "package:utils"   "package:datasets" "package:methods"
[10] "Autoloads"       "package:base"
```

## 2. Βασικές Πράξεις

+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση
^ ή **	Ύψωση σε δύναμη
%/%	Ακέραια διαίρεση
%%	Υπόλοιπο διαίρεσης

```
> 6+2  
[1] 8
```

```
> 9-3  
[1] 6
```

```
> 3*5  
[1] 15
```

```
> 6/4  
[1] 1.5
```

```
> 6%/4  
[1] 1
```

```
> 6%%4  
[1] 2
```

## 3. Βασικές Συναρτήσεις

log/log2/log10(...)	Λογάριθμος: νεπέριος/βάση 2/βάση 10
exp(...)	Εκθετική συνάρτηση
sin/cos/tan(...)	Ημίτονο/συνημίτονο/εφαπτομένη
sqrt(...)	Τετραγωνική ρίζα
seq(i, j, k)	Δίνει την σειρά αριθμών από το i έως το j ανά k.
seq(i, j) ή i:j	Αν k=1, τότε μπορεί να παραληφθεί.
rep(i, k)	Επαναλαμβάνει το i, k φορές.
rep(i, times=k)	

```
> log(10)  
[1] 2.302585
```

```
> log(exp(1))  
[1] 1
```

```
> log2(2)  
[1] 1
```

```
> log10(10)  
[1] 1
```

```

> sqrt(12)
[1] 3.464102

> seq(1,10)
[1] 1 2 3 4 5 6 7 8 9 10

> seq(1,10,2)
[1] 1 3 5 7 9

> 1:10
[1] 1 2 3 4 5 6 7 8 9 10

> rep(3,5)
[1] 3 3 3 3 3

> rep(3,times=5)
[1] 3 3 3 3 3

> x=1:3; rep(x,each=5)
[1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3

```

#### 4. Διανύσματα - Vectors

Τα διανύσματα μπορούν να έχουν για στοιχεία αριθμούς (numeric) ή χαρακτήρες/λέξεις (character). Ένα διάνυσμα ορίζεται με την συνάτηση `c(...)`. Οι λέξεις στα character vectors μπαίνουν σε `"..."` ή `'...'`.

```

> vec.n=c(2.1,3.2,7.3,8.9,5.1,6.9,1.7); vec.n
[1] 2.1 3.2 7.3 8.9 5.1 6.9 1.7

> vec.c=c("Athens","London","NY","Paris"); vec.c
[1] "Athens" "London" "NY" "Paris"

```

Έστω  $X$  ένα numeric vector.

- <code>X[i]</code> - <code>X[c(...)]</code> <code>X[1:3]</code>	- Δίνει το $i$ -οστό στοιχείο του $X$ - Δίνει τα στοιχεία του $X$ που ορίζονται εντός την αγγίλης.
<code>min/max(X)</code>	Μέγιστο/Ελάχιστο στοιχείο του $X$
<code>range(X)</code>	Εύρος του $X$ (min, max)
<code>length(X)</code>	Αριθμός στοιχείων του $X$
<code>sum/prod(X)</code>	Άθροισμα/Γινόμενο των στοιχείων του $X$
<code>sort(X)</code>	Διάνυσμα με τα στοιχεία του $X$ σε σειρά μεγέθους
<code>order(X)</code>	Διάνυσμα με τα σειρά μεγέθους του κάθε στοιχείου του διανύσματος
<code>rep(X,each=w)</code>	Επαναλαμβάνει κάθε στοιχείο του $X$ $w$ -φορές.
<code>mode(...)</code>	Δίνει τον τύπο του στοιχείου

```

> x=c(8,11,23,2,9,6,21,17,31); x
[1] 8 11 23 2 9 6 21 17 31

```

```

> x[5]
[1] 9

> x[1:5]
[1] 8 11 23 2 9

> x[c(2,3,8)]
[1] 11 23 17

> min(x); max(x)
[1] 2
[1] 31

> range(x)
[1] 2 31

> length(x)
[1] 9

> sort(x)
[1] 2 6 8 9 11 17 21 23 31

> order(x)
[1] 4 6 1 5 2 8 7 3 9

> x[order(x)] # ίδιο με το sort(x)
[1] 2 6 8 9 11 17 21 23 31

> mode(x)
[1] "numeric"

```

Έστω character vector.

<ul style="list-style-type: none"> <li>- character</li> <li>- as.character</li> <li>- is.character</li> </ul>	Μεταβλητή με χαρακτήρες
paste(...)	Παίρνει διάφορα στοιχεία και τα βάζει μαζί για να κάνει μεγαλύτερες σειρές από χαρακτήρες

```

> Y=paste("Today is", date()); Y
[1] "Today is Wed Feb 06 17:52:18 2008"

> Y=paste(vec.c,1:4); Y
[1] "Athens 1" "London 2" "NY 3" "Paris 4"

> is.character(vec.c)
[1] TRUE

```

## Λογικές Πράξεις

<, <=	Μικρότερο, μικρότερο ή ίσο
>, >=	Μεγαλύτερο, μεγαλύτερο ή ίσο
==	Ίσο
!=	Όχι ίσο
&	'and'
	'or'
!	Άρνηση

Τα αποτελέσματα λογικών πράξεων είναι TRUE, FALSE και NA ('not available').

```
> x=seq(1:10); x
[1] 1 2 3 4 5 6 7 8 9 10

> x>5
[1] FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE

> x>5 & x<=8
[1] FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE FALSE FALSE

> x>5 | x<=2
[1] TRUE TRUE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE

> x==5
[1] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE

> x!=5
[1] TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE

> a = (x==5); a
[1] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE

> !a
[1] TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE
```

## 5. Πίνακες – Arrays & Matrices

Αυτό που χαρακτηρίζει ένα αντικείμενο (object) αυτής της μορφής είναι οι διαστάσεις του [dim(...)]. Ένα διάνυσμα, ορισμένο όπως στην ενότητα 4 δεν έχει διάσταση.

```
> w=seq(1:24); w
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
[21] 21 22 23 24

> dim(w)
NULL
```

Ένας array 'ορίζετε' ως η συλλογή δεδομένων σε έναν χώρο τριών διαστάσεων. Συνεπώς ο ορισμός της διάστασης του w σε 2X3X2

```

> dim(w)=c(3,4,2); w

, , 1

      [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12

, , 2

      [,1] [,2] [,3] [,4]
[1,]   13   16   19   22
[2,]   14   17   20   23
[3,]   15   18   21   24

```

δημιουργεί array που αποτελείται από δύο πίνακες 3X4. Η εντολή είναι [?array]

```
array(data = NA, dim = length(data), dimnames = NULL)
```

όπου, γενικά, μπορούμε να δημιουργήσουμε ένα 3-διάστατο, 2-διάστατο (πίνακα) ή και 1-διάστατο (διάνυσμα) array, ανάλογα με την επιλογή στο dim(...).

Έστω ο 2-διάστατος πίνακας

```

> z.a=array(1:20,dim=c(4,5)); z.a

      [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20

```

Αυτός μπορεί να γραφεί και με την χρήση της εντολής matrix(...)

```

> z.m=matrix(1:20,nrow=4,ncol=5); z.m

      [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20

> dim(z.m)
[1] 4 5

```

όπου, μιάς και είμαστε σε δύο διαστάσεις, αρκεί να ορίσουμε τον αριθμό των γραμμών (nrow) και στηλών (ncol). [?matrix]

Έστω πίνακες  $X(n, k)$ ,  $Y(n, w)$ .

<code>%**</code>	Πολλαπλασιασμός πινάκων
<code>+, -, *, /</code>	Πράξεις στοιχείο με στοιχείο
<code>t(X)</code>	Γραμμές σε στήλες (transpose)
<code>crossprod(X, Y)</code> <code>t(X) %** Y</code>	Γινόμενο πινάκων αφού πρώτα $t(X)$
<code>nrow(X)</code>	Αριθμός γραμμών του $X$
<code>ncol(X)</code>	Αριθμός στηλών του $X$
<code>diag(2, 3)</code> <code>diag(2, 3, 3)</code>	Διαγώνιος $3 \times 3$ πίνακας με 2-άρια στην διαγώνιο
<code>diag(r)</code>	Διαγώνιος $r \times r$ πίνακας με 1 στη διαγώνιο
<code>solve(X)</code>	Αντίστροφος του $X$
<code>cbind(X, Y)</code>	Ενώνει τους $X$ και $Y$ κατά στήλες
<code>rbind(X, Y)</code>	Ενώνει τους $X$ και $Y$ κατά γραμμές
<code>table(...)</code>	Πίνακες συχνοτήτων
<code>as.matrix(...)</code>	Μετατρέπει το στοιχείο σε πίνακα
<code>is.matrix(...)</code>	Έλεγχος αν το στοιχείο είναι <code>matrix</code>

```
> x=matrix(1:6,2,3); x
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6

> x*x
      [,1] [,2] [,3]
[1,]    1    9   25
[2,]    4   16   36

> t(x)
      [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6

> nrow(x)
[1] 2

> ncol(x)
[1] 3

> crossprod(x,x)
      [,1] [,2] [,3]
[1,]    5   11   17
[2,]   11   25   39
[3,]   17   39   61

> t(x)%**x
      [,1] [,2] [,3]
[1,]    5   11   17
[2,]   11   25   39
[3,]   17   39   61
```

```

> diag(2,3)
      [,1] [,2] [,3]
[1,]    2    0    0
[2,]    0    2    0
[3,]    0    0    2

> diag(3)
      [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1

> solve(diag(2,3))
      [,1] [,2] [,3]
[1,] 0.5 0.0 0.0
[2,] 0.0 0.5 0.0
[3,] 0.0 0.0 0.5

> cbind(diag(2,3),diag(3))
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    2    0    0    1    0    0
[2,]    0    2    0    0    1    0
[3,]    0    0    2    0    0    1

> rbind(diag(2,3),diag(3))
      [,1] [,2] [,3]
[1,]    2    0    0
[2,]    0    2    0
[3,]    0    0    2
[4,]    1    0    0
[5,]    0    1    0
[6,]    0    0    1

> data() # όλα τα διαθέσιμα δεδομένα στο R

# Επιλογή esoph με δεδομένα από κατανάλωση ποτού και τσιγάρων σε
# σχέση με καρκίνο στον οισοφάγο.

> table(esoph[,1])

25-34 35-44 45-54 55-64 65-74 75+
  15   15   16   16   15   11

> table(esoph[,1],esoph[,2]) # Ηλικία και κατανάλωση ποτού

      0-39g/day 40-79 80-119 120+
25-34         4     4     3     4
35-44         4     4     4     3
45-54         4     4     4     4
55-64         4     4     4     4
65-74         4     3     4     4
75+           3     4     2     2

> vec=c(2.1 3.2 7.3 8.9); vec
[1] 2.1 3.2 7.3 8.9

> dim(vec)
NULL

```



```

> vec.m=as.matrix(vec); vec.m
  [,1]
[1,] 2.1
[2,] 3.2
[3,] 7.3
[4,] 8.9

> dim(vec.m)
[1] 4 1

> is.matrix(vec.m)
[1] TRUE

```

## 6. Data frames

Ένα `data.frame` αποτελεί την γενίκευση του πίνακα `matrix`. Σε έναν πίνακα όλα τα στοιχεία πρέπει να είναι της ίδιας μορφής (αριθμοί), ενώ σε μια βάση δεδομένων αυτό δεν αποτελεί περιορισμό. Κάθε στήλη αποτελεί και ένα διάνυσμα με τα δικά του χαρακτηριστικά και με δικό του όνομα.

```

> a=c("Athens","London","NY","Paris"); a
[1] "Athens" "London" "NY"      "Paris"

> b=c("Greece","UK","USA","France"); b
[1] "Greece" "UK"      "USA"     "France"

> c=c(5,8,19,10); c
[1] 5 8 19 10

> geo=data.frame(country=b,city=a,population=c); geo
  country  city population
1 Greece Athens         5
2   UK London          8
3   USA   NY           19
4 France Paris          10

```

Γενικότερα:

<code>read.table(...)</code>	Διαβάζει δεδομένα από εξωτερικό αρχείο
<code>as.data.frame(...)</code>	Μετατρέπει το στοιχείο σε <code>data.frame</code>
<code>is.data.frame</code>	Έλεγχος αν ένα στοιχείο είναι <code>data.frame</code> (TRUE/FALSE)
<code>names(...)</code>	Τα ονόματα των στηλών
<code>...\$...</code>	Αναφορά στις στήλες με το όνομα τους
<code>attach(...)</code>	Προσθέτει στον χώρο εργασίας το <code>data.frame</code> (θέση 2)
<code>detach(...)</code>	Αφαιρεί το <code>data.frame</code> από τον χώρο εργασίας

```

> as.data.frame(c)
  c
1 5
2 8
3 19
4 10

```

```

> is.data.frame(c)
[1] FALSE

> esoph$agegp
 [1] 25-34 25-34 25-34 25-34 25-34 25-34 25-34 25-34 25-34 25-34 25-34 25-34
[13] 25-34 25-34 25-34 35-44 35-44 35-44 35-44 35-44 35-44 35-44 35-44 35-44
[25] 35-44 35-44 35-44 35-44 35-44 35-44 45-54 45-54 45-54 45-54 45-54 45-54
[37] 45-54 45-54 45-54 45-54 45-54 45-54 45-54 45-54 45-54 45-54 55-64 55-64
[49] 55-64 55-64 55-64 55-64 55-64 55-64 55-64 55-64 55-64 55-64 55-64 55-64
[61] 55-64 55-64 65-74 65-74 65-74 65-74 65-74 65-74 65-74 65-74 65-74 65-74
[73] 65-74 65-74 65-74 65-74 65-74 75+ 75+ 75+ 75+ 75+ 75+ 75+
[85] 75+ 75+ 75+ 75+
Levels: 25-34 < 35-44 < 45-54 < 55-64 < 65-74 < 75+

> attach(esoph)

> search()
 [1] ".GlobalEnv"      "esoph"             "package:stats"
 [4] "package:graphics" "package:grDevices" "package:utils"
 [7] "package:datasets" "package:methods"   "Autoloads"
[10] "package:base"

> names(esoph)
 [1] "agegp"      "alcgp"      "tobgp"      "ncases"     "ncontrols"

> agegp
 [1] 25-34 25-34 25-34 25-34 25-34 25-34 25-34 25-34 25-34 25-34 25-34 25-34
[13] 25-34 25-34 25-34 35-44 35-44 35-44 35-44 35-44 35-44 35-44 35-44 35-44
[25] 35-44 35-44 35-44 35-44 35-44 35-44 45-54 45-54 45-54 45-54 45-54 45-54
[37] 45-54 45-54 45-54 45-54 45-54 45-54 45-54 45-54 45-54 45-54 55-64 55-64
[49] 55-64 55-64 55-64 55-64 55-64 55-64 55-64 55-64 55-64 55-64 55-64 55-64
[61] 55-64 55-64 65-74 65-74 65-74 65-74 65-74 65-74 65-74 65-74 65-74 65-74
[73] 65-74 65-74 65-74 65-74 65-74 75+ 75+ 75+ 75+ 75+ 75+ 75+
[85] 75+ 75+ 75+ 75+
Levels: 25-34 < 35-44 < 45-54 < 55-64 < 65-74 < 75+

> ncases
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 3 1 0 0 0
[26] 0 0 2 0 2 1 0 0 0 6 4 5 5 3 6 1 2 4 3 2 4 2 3 3 4
[51] 9 6 4 3 9 8 3 4 5 6 2 5 5 4 2 0 17 3 5 6 4 2 1 3 1
[76] 1 1 1 2 1 2 1 0 1 1 1 2 1

```

## 7. Loops

if ... else ...	
ifelse(cond, a, b)	Συντόμευση της πιο πάνω εντολής
for (... in ...) {...}	Loop
repeat <i>εκφραση</i>	Loop
while (...) <i>εκφραση</i>	Loop
&&	and
	or
break	
next	
tapply/apply/lapply	

```

> c
[1] 5 8 19 10

> for (i in 1:4){if (c[i]==5) print(1) else print(0)}
[1] 1
[1] 0
[1] 0
[1] 0

> for (i in 1:4){print(ifelse(c[i]==5,1,0))}
[1] 1
[1] 0
[1] 0
[1] 0

> for (i in 1:4){if (c[i]>0 && c[i]<10) print(1) else print(0)}
[1] 1
[1] 1
[1] 0
[1] 0

> for (i in 1:4){if (c[i]>10 || c[i]<5) print(1) else print(0)}
[1] 0
[1] 0
[1] 1
[1] 0

> n <- 17; fac <- factor(rep(1:3, length = n), levels = 1:5)

> table(fac)
fac
1 2 3 4 5
6 6 5 0 0

> tapply(1:n, fac, sum)
 1  2  3  4  5
51 57 45 NA NA

> tapply(1:n, fac, range)
$`1`
[1] 1 16

$`2`
[1] 2 17

$`3`
[1] 3 15

$`4`
NULL

$`5`
NULL

> ma <- matrix(c(1:4, 1, 6:8), nrow = 2)

> apply(ma, 1, table) #--> a list of length 2
[[1]]

1 3 7

```

```
2 1 1
```

```
[[2]]
```

```
2 4 6 8
```

```
1 1 1 1
```

## 8. Κατανομές

Distribution	R name	additional arguments
beta	beta	shape1, shape2, ncp
binomial	binom	size, prob
Cauchy	cauchy	location, scale
chi-squared	chisq	df, ncp
exponential	exp	rate
F	f	df1, df2, ncp
gamma	gamma	shape, scale
geometric	geom	prob
hypergeometric	hyper	m, n, k
log-normal	lnorm	meanlog, sdlog
logistic	logis	location, scale
negative binomial	nbinom	size, prob
normal	norm	mean, sd
Poisson	pois	lambda
Student's t	t	df, ncp
uniform	unif	min, max
Weibull	weibull	shape, scale
Wilcoxon	wilcox	m, n

### Πρόθεμα:

d	Density – Σ.Π. ή Σ.Π.Π.
p	CDF – Συνάρτηση κατανομής
q	Ποσοστημόριο
r	Simulation – Προσομοίωση

```
> pt(-2.43,df=13)          # p-value από t-κατανομή με df=13  
[1] 0.01516545
```

```
> runif(5,0,10)  
[1] 3.2341961 5.3954878 0.9960781 0.6891724 8.1298156
```

```
> qnorm(0.50,0,1)  
[1] 0
```

```
> pnorm(0,0,1)
[1] 0.5

> qf(0.99,2,7)
[1] 9.546578
```

density(...)	Υπολογίζει την density function
hist(...)	Ιστόγραμμα
ecdf(...)	Empirical CDF
qqnorm	QQ plot
qqline	Προσθέτει στο QQ-plot μια γραμμή που περνά από το πρώτο και τρίτο τεταρτημόριο.
qqplot	QQ-plot για δύο datasets

```
> attach(faithful)      # data του R

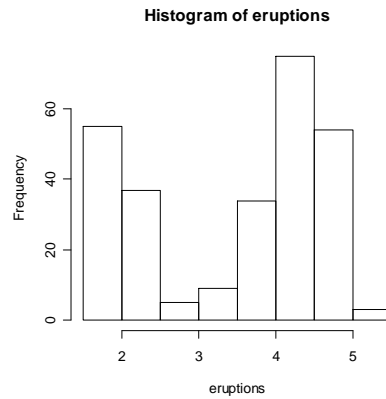
> summary(eruptions)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 1.600  2.163   4.000   3.488   4.454   5.100

> stem(eruptions)

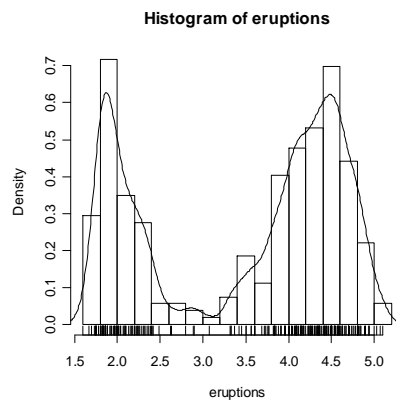
The decimal point is 1 digit(s) to the left of the |

16 | 0703555558
18 | 0000222333333557777777888822335777888
20 | 00002223378800035778
22 | 0002335578023578
24 | 00228
26 | 23
28 | 080
30 | 7
32 | 2337
34 | 250077
36 | 0000823577
38 | 2333335582225577
40 | 0000003357788888002233555577778
42 | 03335555778800233333555577778
44 | 0222233555778000000023333357778888
46 | 0000233357700000023578
48 | 00000022335800333
50 | 0370

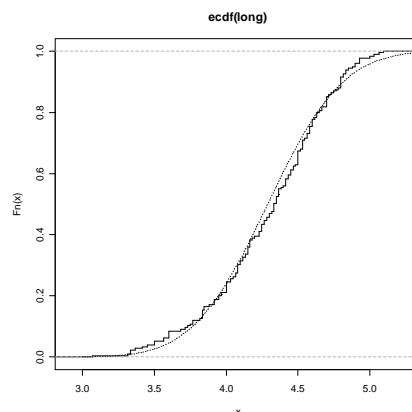
> hist(eruptions)
```



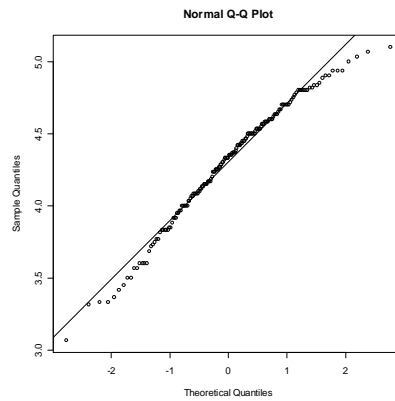
```
> hist(eruptions,seq(1.6,5.2,0.2),prob=TRUE)
> lines(density(eruptions,bw=0.1))
> rug(eruptions)
```



```
> long=eruptions[eruptions>3]
> plot(ecdf(long),do.points=FALSE,verticals=TRUE)
> x=seq(3,5.4,0.01)
> lines(x,pnorm(x,mean=mean(long),sd=sqrt(var(long))),lty=3)
```



```
> qqnorm(long); qqline(long)
```

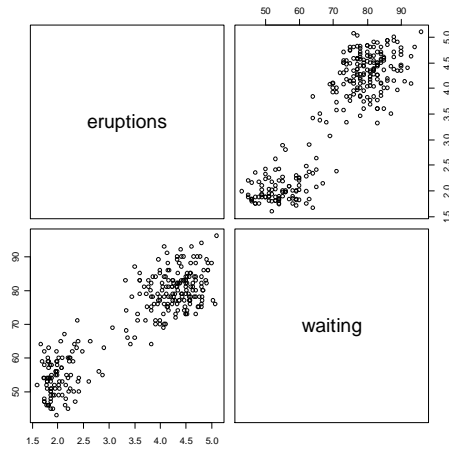


## 9. Γραφήματα

<code>plot</code>	Generic function για να δημιουργεί διάφορα γραφήματα
<code>lines</code>	Generic function για να προσθέτει γραμμές σε ήδη υπάρχοντα γραφήματα
<code>points</code>	Generic function για να προσθέτει σημεία σε ήδη υπάρχοντα γραφήματα
<code>pairs</code>	Δημιουργεί scatter plot
<code>hist</code>	Ιστόγραμμα
<code>boxplot</code>	Box Plot
<code>abline</code>	Προσθέτει ευθείες γραμμές σε υπάρχοντα γραφήματα
<code>legend</code>	Προσθέτει υπόμνημα σε γράφημα
<code>title</code>	Προσθέτει τίτλο
<code>axis</code>	Επιτρέπει αλλαγές στην παρουσίαση των αξόνων στο γράφημα
<code>par()</code>	Δίνει την δυνατότητα ορισμού/αλλαγών σε μεγάλο αριθμό παραμέτρων σχετικών με το γράφημα

```
> summary(faithful)
```

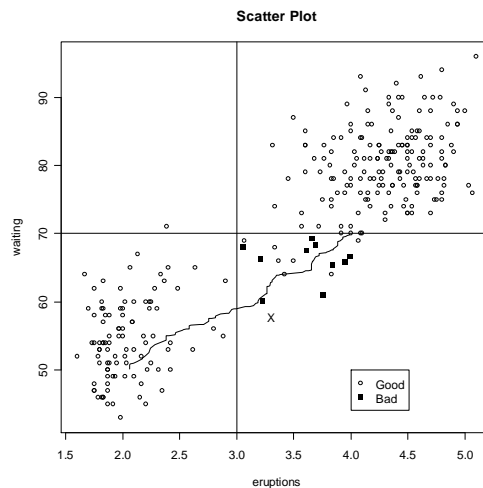
```
> pairs(faithful)
```



```

> plot(eruptions,waiting)
> points(runif(10,3,4),runif(10,60,70),pch=15)
> abline(h=70)
> abline(v=3)
> lines(sort(runif(100,2,4)),sort(runif(100,50,70)))
> legend(4,50,legend=c("Good","Bad"),pch=c(1,15))
> title("Scatter Plot")
> text(3.3,58,"X")

```



## 10. Γράφοντας functions

Όλα τα functions που ήδη υπάρχουν στο R είναι γραμμένα με τον ίδιο ακριβώς τρόπο που γράφουμε και τις δικές μας functions. Αν γράψουμε το όνομα του function χωρίς παραμέτρους, τότε μπορούμε να δούμε τον κώδικα.

Ένα function ορίζετε ως

```

όνομα <- function(παρα_1, παρα_2, ...) { ...έκφραση της συνάρτησης... }

```

Για παράδειγμα, η απλή συνάρτηση se

```

se <- function(vec){ sqrt( var(vec) ) }

```



υπολογίζει την τυπική απόκλιση για το διάνυσμα τιμών `vec` .

**Notes:**

- Οι μεταβλητές και τα functions που ορίζονται εντός ενός function είναι τοπικά (local) που σημαίνει ότι είναι προσωρινά και δεν δημιουργούνται στον χώρο εργασίας. Γίνονται στα πλαίσια του function και μετά χάνονται.
- Μόνιμες (global) αναθέσεις γίνονται με την χρήση του συμβόλου `<<-` ή της εντολής `assign( )`.
- Είναι σημαντική η σειρά που μπαίνουν οι παράμετροι κατα τον ορισμό του function, έτσι ώστε να δέχονται τις σωστές τιμές. ΠΧ, την πύ πάνω συνάρτηση μπορούμε να την καλεσουμε με δύο τρόπους για ένα διάνυσμα τιμών X
  - `se(X)`
  - `se(vec=X)`

Συνεπώς, ή δίνονται οι παράμετροι με την σειρά που ορίζονται ή απαραίτητως πρέπει να βάζουμε το όνομα της μεταβλητής και μετά την τιμή.

- Αυτό που θέλουμε να είναι το αποτέλεσμα το γράφουμε τελευταίο στο function.
- Μπορούμε να πέρνουμε ‘ενδιάμεσα’ αποτελέσματα κατα την εφαρμογή του function με την χρήση εντολών όπως `print(...)`.
- Κατά τον ορισμό των παραμέτρων ενός function μπορούμε να χρησιμοποιήσουμε το ‘...’ ως στοιχείο.

...	Επιτρέπει την ‘μεταφορά’ παραμέτρων από άλλα functions.
-----	---

**11. Διάφορες Χρήσιμες Εντολές/Συναρτήσεις**

<code>summary</code>	Generic function για ‘λεπτομέρειες’
- <code>factor</code> - <code>as.factor</code> - <code>is.factor</code>	- Ποιοτική Μεταβλητή - Δημιουργεί ποιοτική μεταβλητή από άλλη, συνηθως ποσοτική - Έλεγχος ποιοτικής μεταβλητής
<code>is.na</code>	Έλεγχος NA
- <code>integer</code> - <code>as.integer</code> - <code>is.integer</code>	- Ακέραιος - Δημιουργεί ακέραια μεταβλητή από άλλη - Έλεγχος ποιοτικής μεταβλητής
<code>round</code>	Στρογγυλοποίηση
<code>int</code>	Ολοκλήρωμα
- <code>numeric</code> - <code>as.numeric</code> - <code>is.numeric</code>	Αριθμητική μεταβλητή

## Προβλήματα

- Φτιάξτε τα πιο κάτω διανύσματα
  - (9,9,9,9,9,9,9,9,9,9,9,1)
  - (1.0,1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8,1.9,2.0)
  - (1.0,1.3,1.4,1.6,1.7,1.8,1.9)
  - (10,9,8,7,6,5,4)
  - (5,5,5,5,5,6,7,8,9,10)
- Φτιάξτε ένα τυχαίο διάνυσμα με length 100. Υπολογίστε
  - Μέση τιμή
  - Διασπορά
  - Τυπική απόκλιση
  - Τεταρτημόρια
  - Range
- Δημιουργήστε ένα πίνακα/data.frame της αρεσκείας σας.
  - Κάντε sort τις γραμμές του πίνακα/data.frame με βάση τη σειρά μεγέθους μιάς εκ των numeric στήλης/variables.
  - Δημιουργήστε νέα data.frame από διαφορετικές στήλες του αρχικού.
- Πάρτε ένα character vector με τα 3 πρώτα γράμματα ελληνικών πόλεων. Ελέγξτε αν η ΑΘΗ(ΝΑ) είναι στο διάνυσμα.
- Κατασκευάστε ένα 3x4x5 array.
  - Πάρτε το γινόμενο του πρώτου από τους πίνακες με τον τέταρτο.
  - Θέστε τον δεύτερο ίσο με τον πέμπτο
  - Αλλάξτε το (2,2) στοιχείο του 1<sup>ου</sup> πίνακα με το (3,2) του 3<sup>ου</sup>.
  - Φτιάξτε loop ώστε να πολλαπλασιάσετε το  
(3,3) στοιχείο του 1<sup>ου</sup> πίνακα με 10  
(2,3) στοιχείο του 2<sup>ου</sup> πίνακα με 9  
(2,3) στοιχείο του 3<sup>ου</sup> πίνακα με 8  
(2,2) στοιχείο του 4<sup>ου</sup> πίνακα με 7  
(1,3) στοιχείο του 5<sup>ου</sup> πίνακα με 6
  - Φτιάξτε loop ώστε να πολλαπλασιάζετε τα στοιχεία της πρώτης γραμμής κάθε 3x4 πίνακα με 7, 9, 13, 21,37 αντίστοιχα.
- Δημιουργήστε 4 τυχαία διανύσματα του ίδιου length, έστω 20, τα οποία να εκφράζουν ηλικία, βάρος, ύψος και φύλλο (σε numeric μορφή).
  - Κατασκευάστε τον 20x4 πίνακα από τα διανύσματα.
  - Μετατρέψτε τον matrix σε data.frame, δίνοντας στις στήλες τα κατάλληλα ονόματα.
  - Μετατρέψτε την μεταβλητή 'φύλλο' από numeric σε character.
  - Βρείτε το ύψος του δεύτερου άνδρα και αλλάξτε το με νέα τιμή.
  - Κάντε plot το scatter plot μεταξύ ηλικίας και βάρους. Προσθέστε στο γράφημα την γραμμή παλινδρόμησης.
- Πάρτε 20 ακέραιους τυχαίους αριθμούς στο [1,1000]. Από αυτούς τους 20:
  - Πάρτε 10 τυχαίους αριθμούς με επανάθεση.
  - Πάρτε 10 τυχαίους αριθμούς χωρίς επανάθεση. Βεβαιωθείτε ότι έχετε σώσει κάπου και τους υπόλοιπους 10 που δεν επιλέγησαν.
  - Πάρτε μία μετάθεση των 20 αριθμών.
- Από το data.frame στο 2 (ή από όποιο data.frame θέλετε), γράψτε μία δική σας function που να υπολογίζει:
  - Την τυπική απόκλιση της επιθυμητής στήλης/variable.

- b. Την συνδιασπορά και τον συντελεστή συσχέτισης μεταξύ των επιθυμητών στηλών/variables.
  - c. Όσα μέτρα θέσης και μεταβλητότητας θέλετε, τα οποία και θα δίνει συνολικά ως ένα και μοναδικό αποτέλεσμα (υπό όποια μορφή σας βολεύει – vector/matrix/data.frame ή list).
9. Θεωρήστε το δεδομένα `esoph` που υπάρχουν στο R. Κατασκευάστε
- a. `boxplot` των `ncontrols` ανα `agegr`
  - b. Ιστόγραμμα των `ncases`
  - c. 2x2 πίνακες συχνοτήτων `agegr` και `alcgr` ανά `tobgr`.
10. Έστω  $X_1, X_2, \dots, X_n$  τ.δ. από εκθετική με παράμετρο  $\theta$ ,  $\text{Exp}(\theta)$ . Γράψτε μία function που να υπολογίζει τον  $\log$  της συνάρτησης μεγίστης πιθανοφάνειας,  $\log L$ , με δύο τρόπους
- a. Γράφοντας τις δικές σας functions για τον υπολογισμό της συνάρτησης μεγίστης πιθανοφάνειας.
  - b. Με την χρήση των ήδη υπάρχοντων functions του R.
- Για ένα τυχαίο δείγμα από  $\text{Exp}(0.2)$  και  $n=1000$ , βρείται το  $\hat{\theta}$ .