

Κεφάλαιο 2

Αντικείμενα Δεδομένων

Στο κεφάλαιο αυτό γίνεται εισαγωγή στην ιδέα των αντικειμένων δεδομένων. Τα αντικείμενα δεδομένων είναι οι διάφορες μορφές στις οποίες μπορούν να φυλαχθούν δεδομένα στην R. Οι κύριες μορφές αντικειμένων δεδομένων που υπάρχουν στην R είναι τα ακόλουθα:

- διάνυσμα (vector)
- πίνακας (matrix)
- πίνακας μεγαλύτερης διάστασης (array)
- λίστα (list)
- παράγοντας (factor)
- χρονοσειρές (time series)
- πλαίσιο δεδομένων (data frame).

Σε αυτό το κεφάλαιο θα αναπτυχθούν όλες οι πιο πάνω μορφές αντικειμένων, εκτός από τις χρονοσειρές οι οποίες θα αναλυθούν σε επόμενο κεφάλαιο.

2.1 Διανύσματα

Το πιο απλό είδος αντικειμένου είναι το διάνυσμα. Το διάνυσμα είναι απλά ένα διατεταγμένο σύνολο τιμών σε σειρά. Η εσωτερική διάταξη του διανύσματος υποδεικνύει ότι υπάρχει ένας κατάλληλος τρόπος με τον οποίο μπορούν να εξαχθούν

μερικά ή όλα από τα στοιχεία του. Ο πιο εύκολος τρόπος για να προσδιοριστεί ένα διάνυσμα είναι μέσω της εντολής `c`. Για παράδειγμα,

```
> x <- c(1,3,4,5)
> x
[1] 1 3 4 5
> length(x)
[1] 4
> mode(x)
[1] "numeric"
> names(x)
NULL
> y <- c( c(2,3), c(1,-6))
> y
[1] 2 3 1 -6
```

Ένας άλλος τρόπος, ο οποίος μπορεί να χρησιμοποιηθεί για την κατασκευή διανύσματος, ειδικά στην περίπτωση που είναι αναγκαίο να γίνει επανάληψη κάποιων τιμών, δίνεται με τη βοήθεια της συνάρτησης `rep`. Η συνάρτηση `rep()` καθορίζει είτε το πόσες φορές θα γίνει η επανάληψη με το όρισμα `times`, είτε το μέγεθος του διανύσματος με το όρισμα `length`.

```
> rep(NA,6)
[1] NA NA NA NA NA NA
> rep(x, 3)
[1] 1 3 4 5 1 3 4 5 1 3 4 5
> rep(x, c(1,2,2,3))
[1] 1 3 3 4 4 5 5 5
```

Όπως παρατηρούμε στο τελευταίο παράδειγμα όταν το όρισμα `times` είναι ένα διάνυσμα με το ίδιο μέγεθος με το διάνυσμα των τιμών οι οποίες θα επαναληφθούν, τότε κάθε τιμή επαναλαμβάνεται τις αντίστοιχες φορές. Επιπλέον, ο τελεστής ακολουθίας : παράγει μία ακολουθία τιμών οι οποίες απέχουν μεταξύ τους μία μονάδα.

```
> 1:13
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13
> -3:6
[1] -3 -2 -1 0 1 2 3 4 5 6
```

```
> 1.1:5
[1] 1.1 2.1 3.1 4.1
> 4:-5
[1] 4 3 2 1 0 -1 -2 -3 -4 -5
```

Γενικότερα, με τη βοήθεια της συνάρτησης `seq` μπορούμε να κατασκευάσουμε μία ακολουθία αριθμών με οποιαδήποτε διαφορά μεταξύ των τιμών. Το επόμενο παράδειγμα επεξηγεί πως χρησιμοποιείται:

```
> seq(-1,2, 0.5)
[1] -1.0 -0.5 0.0 0.5 1.0 1.5 2.0
> seq(-1,2, length=12)
[1] -1.00000000 -0.72727273 -0.45454545 -0.18181818
[5] 0.09090909 0.36363636 0.63636364 0.90909091
[9] 1.18181818 1.45454545 1.72727273 2.00000000
> seq(1, by=0.5, length=12)
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5
```

2.2 Πίνακες

Οι πίνακες χρησιμοποιούνται για να τακτοποιήσουν τιμές κατά γραμμές και στήλες σε ένα ορθογώνιο πίνακα. Στην ανάλυση δεδομένων, οι διάφορες μεταβλητές συνήθως παρουσιάζονται σε διαφορετικές στήλες και οι διάφορες περιπτώσεις ή τιμές παρουσιάζονται σε διαφορετικές γραμμές. Οι πίνακες διαφέρουν από τα διάνυσματα γιατί έχουν διαστάσεις και σε αυτούς μπορεί να εφαρμοστεί η συνάρτηση διάστασης `dim`.

Για να δημιουργηθεί ένας πίνακας από ένα διάνυσμα, χρησιμοποιείται η συνάρτηση διάστασης `dim` εκχωρώντας ένα διάνυσμα με 2 ακέραιους αριθμούς οι οποίοι αντιστοιχούν στον αριθμό των γραμμών και των στηλών του πίνακα, αντίστοιχα.

```
> matr <- rep(1:4, rep(3,4))
> matr
[1] 1 1 1 2 2 2 3 3 3 4 4 4
> dim(matr) <- c(3,4)
> matr
      [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
```

```

[2,] 1 2 3 4
[3,] 1 2 3 4
> matr2 <- seq(-2,2,length=25)
> matr2
 [1] -2.000000 -1.8333333 -1.6666667 -1.5000000 -1.3333333 -1.1666667 -1.0000000
 [8] -0.8333333 -0.6666667 -0.5000000 -0.3333333 -0.1666667 0.0000000 0.1666667
[15] 0.3333333 0.5000000 0.6666667 0.8333333 1.0000000 1.1666667 1.3333333
[22] 1.5000000 1.6666667 1.8333333 2.0000000
> dim(matr2) <- c(5,5)
> matr2
      [,1]      [,2]      [,3]      [,4]      [,5]
[1,] -2.000000 -1.1666667 -0.3333333 0.5000000 1.3333333
[2,] -1.8333333 -1.0000000 -0.1666667 0.6666667 1.5000000
[3,] -1.6666667 -0.8333333 0.0000000 0.8333333 1.6666667
[4,] -1.5000000 -0.6666667 0.1666667 1.0000000 1.8333333
[5,] -1.3333333 -0.5000000 0.3333333 1.1666667 2.0000000

```

Συχνά χρειάζεται να συνδεθούν μεταξύ τους διάφορα διανύσματα ή πίνακες για να δημιουργηθεί ένας καινούργιος πίνακας. Αυτό γίνεται εφικτό με τη βοήθεια των συναρτήσεων `rbind` και `cbind`.

```

> matr3 <- rbind(c(1,2,-1), c(-3,1,5))
> matr3
      [,1] [,2] [,3]
[1,] 1 2 -1
[2,] -3 1 5
> matr4 <- cbind(c(1,2,-1), c(-3,1,5))
> matr4
      [,1] [,2]
[1,] 1 -3
[2,] 2 1
[3,] -1 5
> matr5 <- cbind(c(1,2,-1), c(-3,3,2,0))
Warning messages:
  Number of rows of result is not a multiple of
  vector length (arg 1) in: cbind(c(1, 2,-1), c(-3, 3, 2, 0))
> matr5
      [,1] [,2]

```

```
[1,]  1  -3
[2,]  2   3
[3,] -1   2
[4,]  1   0
```

Στην περίπτωση σύνδεσης διανυσμάτων με διαφορετικά μεγέθη, η χρήση των συναρτήσεων `cbind` ή `rbind`, δίνει σαν αποτέλεσμα τις τιμές των μικρότερων από αυτά να επαναλαμβάνονται κυκλικά έτσι ώστε ο πίνακας να συμπληρωθεί εντελώς.

```
matr6 <- cbind(matr, matr4)
> matr6
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    2    3    4    1   -3
[2,]    1    2    3    4    2    1
[3,]    1    2    3    4   -1    5
> matr6 <- cbind(matr, matr3)
Error in cbind(matr, matr3): Number of rows of matrices and
lengths of names vectors must match (see arg 2)
```

Ένας εναλλακτικός τρόπος για να δημιουργηθεί ένας πίνακας είναι με τη συνάρτηση `matrix`, η οποία παίρνει ως ορίσματα τον αριθμό των γραμμών (`nrow`) και των στηλών (`ncol`).

```
> matr7 <- matrix(1:28, nrow=7, ncol=4)
> matr7
      [,1] [,2] [,3] [,4]
[1,]    1    8   15   22
[2,]    2    9   16   23
[3,]    3   10   17   24
[4,]    4   11   18   25
[5,]    5   12   19   26
[6,]    6   13   20   27
[7,]    7   14   21   28
> matr8 <- matrix(-5:6, ncol=3, byrow=T)
> matr8
      [,1] [,2] [,3]
[1,]   -5   -4   -3
```

```
[2,] -2 -1 0
[3,] 1 2 3
[4,] 4 5 6
> matrix(1:23, nrow=7, ncol=4)
```

```
      [,1] [,2] [,3] [,4]
[1,] 1 8 15 22
[2,] 2 9 16 23
[3,] 3 10 17 1
[4,] 4 11 18 2
[5,] 5 12 19 3
[6,] 6 13 20 4
[7,] 7 14 21 5
```

Warning messages:

```
Replacement length not a multiple of number of
elements to replace in: data[1:11] <-old
```

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

Warning messages:

```
Replacement length not a multiple of number of
elements to replace in: data[1:11] <-old
```

Το όρισμα `byrow` είναι πολύ χρήσιμο όταν γίνεται η ανάγνωση των δεδομένων από ένα αρχείο κειμένου (*text file*). Τέλος δίνονται μερικές εντολές οι οποίες χρησιμοποιούνται στην αναγνώριση του μεγέθους, των διαστάσεων και τη μορφή των τιμών του πίνακα, αλλά και πώς μπορούν να δοθούν ονόματα στις γραμμές και τις στήλες του.

```
> matr8
      [,1] [,2] [,3]
[1,] -5 -4 -3
[2,] -2 -1 0
```

```

[3,]  1  2  3
[4,]  4  5  6
> length(matr8)
[1] 12
> dim(matr8)
[1] 4 3
> mode(matr8)
[1] "numeric"
> dimnames(matr8)
NULL
> dimnames(matr8) <- list(c("A","B","C","D"), c("K1","K2","K3"))
> matr8
  K1 K2 K3
A -5 -4 -3
B -2 -1  0
C  1  2  3
D  4  5  6

```

2.3 Πίνακες μεγαλύτερης διάστασης (Arrays)

Τα arrays γενικεύουν τους πίνακες επεκτείνοντας την έννοια της διάστασής τους σε παραπάνω από δύο. Κατά συνέπεια, μεγαλώνει και η διάσταση της συνάρτησης `dim`. Για παράδειγμα, αν οι γραμμές και οι στήλες ενός πίνακα (`matrix`) είναι το μήκος και το πλάτος μιας ορθογώνιας διευθέτησης τιμών ίσων διαστάσεων κύβου, τότε το μήκος, το πλάτος και το ύψος εκπροσωπούν τις διαστάσεις ενός πίνακα τριών διαστάσεων (*three way array*). Δεν υπάρχει κανένας περιορισμός στον αριθμό των διαστάσεων ενός πίνακα μεγαλύτερης διάστασης.

```

> arr1 <- array( c(2:9,12:19,112:119), dim=c(2,4,3))
> arr1
, , 1
  [,1] [,2] [,3] [,4]
[1,]  2   4   6   8
[2,]  3   5   7   9

, , 2
  [,1] [,2] [,3] [,4]

```

```
[1,] 12 14 16 18
[2,] 13 15 17 19

, , 3
      [,1] [,2] [,3] [,4]
[1,] 112 114 116 118
[2,] 113 115 117 119
```

Η πρώτη διάσταση (γραμμές) συμπληρώνεται πρώτη. Αυτό είναι το ίδιο με το να τοποθετούνται οι τιμές στήλη με στήλη. Η δεύτερη διάσταση συμπληρώνεται δεύτερη. Η τρίτη διάσταση συμπληρώνεται με τη δημιουργία ενός πίνακα για κάθε επίπεδο της τρίτης διάστασης. Στους πίνακες μεγαλύτερης διάστασης εφαρμόζονται οι ίδιες εντολές για την αναγνώριση του μεγέθους, των διαστάσεων και τη μορφή των τιμών τους όπως και στην περίπτωση των πινάκων, αλλά και με τον ίδιο τρόπο δίνονται ονόματα στις διαστάσεις τους.

```
> length(arr1)
[1] 24
> mode(arr1)
[1] "numeric"
> dim(arr1)
[1] 2 4 3
> dimnames(arr1)
NULL
```

2.4 Λίστες

Ως αυτό το σημείο, όλα τα αντικείμενα δεδομένων τα οποία έχουν περιγραφεί είναι *ατομικά*. Αυτό σημαίνει ότι περιέχουν μόνο μιας μορφής δεδομένα. Όμως, είναι αρκετές εκείνες οι περιπτώσεις στις οποίες υπάρχει η ανάγκη να δημιουργηθούν αντικείμενα δεδομένων τα οποία περιέχουν διάφορες μορφές τιμών. Η λύση προσφέρεται μέσω των αντικειμένων *λίστας (list)* τα οποία αποτελούνται από διάφορες συνιστώσες, η κάθε μια από τις οποίες περιέχει διαφορετική μορφή δεδομένων.

```
> group1 <- c(rep(1,11), rep(2,17))
```

```

> group2 <- c(23,45,67,76,-8,3.5,2.19,4)
> groups <- list(case=group1, control=group2, descrip="An example")
> groups
$case:
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
$control:
 [1] 23.00 45.00 67.00 76.00 -8.00 3.50 2.19 4.00
$descrip:
 [1] "An example"

```

Για την εξαγωγή μιας συνιστώσας της λίστας χρησιμοποιούμε το σύμβολο \$ ή [[]].

```

> groups$case
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
> groups$control
 [1] 23.00 45.00 67.00 76.00 -8.00 3.50 2.19 4.00
> groups[[1]]
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
> groups[[2]]
 [1] 23.00 45.00 67.00 76.00 -8.00 3.50 2.19 4.00
> groups[[2]][1:2]
 [1] 23 45
> length(groups)
 [1] 3
> mode(groups)
 [1] "list"
> names(groups)
 [1] "case" "control" "descrip"

```

2.5 Παράγοντες

Για σκοπούς ανάλυσης δεδομένων, μερικές από τις μεταβλητές μπορεί να μην είναι ποσοτικές αλλά ποιοτικές ή κατηγορικές. Μερικά παραδείγματα τέτοιων μεταβλητών είναι

- το *φύλο* με τιμές άντρας ή γυναίκα,
- η *οικογενειακή κατάσταση* με τιμές ελεύθερος, παντρεμένος ή χωρισμένος.

Οι κατηγορικές μεταβλητές παρουσιάζονται στην R με το αντικείμενο δεδομένων που λέγεται παράγοντας (factor). Για να κατασκευαστεί ένας παράγοντας εφαρμόζεται η συνάρτηση factor. Παρατίθενται μερικά παραδείγματα:

```
> gender <- c("male", "female", "male", "male", "female", "female", "male")
> gender
[1] "male" "female" "male" "male" "female" "female" "male"
> factor(gender)
[1] male female male male female female male
> intensity <- factor(c("Hi", "Med", "Lo", "Hi", "Hi", "Lo"),
+ levels=c("Hi","Lo"))
> intensity
[1] Hi NA Lo Hi Hi Lo
> levels(intensity)
[1] "Hi" "Lo"
> intensity <- factor(c("Hi", "Med", "Lo", "Hi", "Hi", "Lo"),
+ levels=c("Hi","Lo"), labels=c("HighDose", "LowDose"))
> intensity
[1] HighDose NA LowDose HighDose HighDose LowDose
```

Αν η σειρά των κατηγοριών του παράγοντα είναι σημαντική, τότε χρησιμοποιείται η συνάρτηση ordered.

```
> intensity <- ordered(c("Hi", "Med", "Lo", "Hi", "Hi", "Lo"),
+ levels=c("Lo", "Med", "Hi"))
> intensity
[1] Hi Med Lo Hi Hi Lo
Lo < Med < Hi
```

Ένας παράγοντας μπορεί να κατασκευαστεί επίσης και από μια συνεχή μεταβλητή με τη βοήθεια της συνάρτησης cut.

```
> fact <- rnorm(10)
> fact1 <- cut(fact, breaks=c(-5,-1,1,2,4))
> fact1
[1] (-1,1] (-5,-1] (-5,-1] (-1,1] (-1,1] (-5,-1] (-1,1] (-1,1] (-5,-1]
[10] (-1,1]
Levels: (-5,-1] (-1,1] (1,2] (2,4]
```

```

> fact2
[1] (-0.429,-0.0166] (-1.67,-1.25] (-1.25,-0.841] (-0.429,-0.0166]
[5] (-0.841,-0.429] (-1.25,-0.841] (-0.0166,0.396] (-0.0166,0.396]
[9] (-1.67,-1.25] (-0.429,-0.0166]
5 Levels: (-1.67,-1.25] (-1.25,-0.841] (-0.841,-0.429] ... (-0.0166,0.396]

```

Κάποιες άλλες χρήσιμες εντολές στην περίπτωση των παραγόντων είναι οι ακόλουθες :

```

> length(intensity)
[1] 6
> mode(intensity)
[1] "numeric"
> names(intensity)
NULL
> levels(intensity)
[1] "Lo" "Med" "Hi"
> class(intensity)
[1] "ordered" "factor"

```

2.6 Πλαίσια Δεδομένων (Data Frames)

Το κύριο πλεονέκτημα του πλαισίου δεδομένων είναι ότι επιτρέπει τον συνδυασμό δεδομένων διαφορετικών μορφών μέσα σε ένα αντικείμενο για να χρησιμοποιηθεί για ανάλυση και μοντελοποίηση. Η ιδέα του πλαισίου δεδομένων είναι η ταξινόμηση των τιμών κατά μεταβλητή (στήλη) ανεξάρτητα της μορφής τους. Έπειτα, όλες οι παρατηρήσεις ενός συγκεκριμένου συνόλου μεταβλητών ταξινομούνται σε πλαίσιο δεδομένων. Για παράδειγμα, παρατίθενται 13 τυχαίες παρατηρήσεις (γραμμές) του πλαισίου δεδομένων `solder` το οποίο υπάρχει μέσα στο πακέτο της R `"faraway"`. Η επιλογή τυχαίου δείγματος γίνεται μέσω της συνάρτησης `sample`.

```

> library("faraway")
> test <- sample(1:900, 13)
> solder[test,]
  Opening Solder Mask PadType Panel skips
713      S  Thin   B3     L8     2    28
652      L  Thin   B3     L8     1     1
793      S Thick   B6     L6     1     7

```

372	L	Thick	A3	D6	3	0
200	L	Thick	A3	L7	2	0
725	L	Thick	B6	D4	2	0
495	M	Thin	A6	L6	3	6
364	L	Thick	A3	D4	1	0
499	M	Thin	A6	L7	1	4
782	S	Thick	B6	W4	2	10
29	L	Thick	A1.5	L9	2	0
196	L	Thick	A3	D7	1	0
724	L	Thick	B6	D4	1	1

Η μεταβλητή `skips` είναι συνεχής ενώ οι υπόλοιπες είναι διάφοροι παράγοντες `factors`. Υπάρχουν διάφοροι τρόποι για να κατασκευαστεί ένα πλαίσιο δεδομένων:

- `read.table` διαβάζει δεδομένα από ένα εξωτερικό αρχείο (δοκιμάστε το με ένα δικό σας αρχείο),
- `data.frame` τοποθετεί μαζί αντικείμενα διαφόρων μορφών.
- `as.data.frame` μετατρέπει αντικείμενα συγκεκριμένης μορφής σε αντικείμενο της τάξης `data.frame`.

Σε αυτό το σημείο θα εξεταστεί μόνο ο δεύτερος τρόπος.

```
> my.logic<-sample(c(T,F),size=20,replace=T)
> my.logic
[1] TRUE TRUE FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE TRUE
[13] TRUE TRUE FALSE FALSE FALSE TRUE TRUE TRUE
> my.complex<-rnorm(20)+runif(20)*1i
> my.complex
[1] 0.1782105+0.9851267i -2.6140989+0.3541577i -0.0767501+0.2550082i
[4] -0.7275827+0.3569999i -0.3280398+0.5163078i -0.7451964+0.8069675i
[7] -0.6853192+0.5144991i -0.4147151+0.5529729i 2.1608968+0.5833807i
[10] 0.5956258+0.2673282i 1.5650520+0.3968731i 1.4445524+0.3118638i
[13] 1.6063870+0.3436187i -1.4068157+0.2399498i 2.2837193+0.3538850i
[16] 1.4940472+0.9729339i -1.2086423+0.8382830i 0.4870967+0.9581304i
[19] 0.6865523+0.6611055i 0.9580948+0.1250858i
> my.numeric<-rnorm(20)
> my.numeric
```

```
[1] 0.4452555 0.4751909 0.9357701 1.5757764 -1.0958323 2.1620200
[7] -1.1306062 -0.4240480 0.2262562 -1.4675688 -0.5541264 1.0983463
[13] 1.3681229 0.2005744 0.5910446 0.8732281 0.3314982 0.8242927
[19] 0.7790229 0.5909648
```

```
> my.matrix<-matrix(rnorm(40),ncol=2)
```

```
> my.matrix
```

```
      [,1]      [,2]
[1,] 0.650290036 1.52145790
[2,] -1.059263140 -0.09996308
[3,] 0.216042514 1.14707512
[4,] -0.114622892 0.59689871
[5,] -0.004433541 1.21214093
[6,] -0.978986416 -0.60250469
[7,] -0.609778169 0.68110679
[8,] 0.138456517 -0.65849203
[9,] 1.271366406 -2.23159156
[10,] -0.016984227 1.06334080
[11,] -0.135241342 0.05793721
[12,] 1.781934098 -0.22806050
[13,] 1.268863189 -2.14581499
[14,] -0.963995714 -1.38571628
[15,] 0.160374068 -0.18793847
[16,] 0.511570707 0.09455187
[17,] -1.126031052 -0.07339069
[18,] 0.394865156 -0.23565899
[19,] -0.238627823 -0.92214415
[20,] -0.755950206 0.86695967
```

```
> my.dataframe<-data.frame(my.logic,my.complex,my.numeric,my.matrix)
```

```
> my.dataframe
```

```
  my.logic      my.complex my.numeric      X1      X2
1    TRUE 0.1782105+0.9851267i 0.4452555 0.650290036 1.52145790
2    TRUE -2.6140989+0.3541577i 0.4751909 -1.059263140 -0.09996308
3   FALSE -0.0767501+0.2550082i 0.9357701 0.216042514 1.14707512
4   FALSE -0.7275827+0.3569999i 1.5757764 -0.114622892 0.59689871
5   FALSE -0.3280398+0.5163078i -1.0958323 -0.004433541 1.21214093
6   FALSE -0.7451964+0.8069675i 2.1620200 -0.978986416 -0.60250469
7    TRUE -0.6853192+0.5144991i -1.1306062 -0.609778169 0.68110679
```

```

8 FALSE -0.4147151+0.5529729i -0.4240480 0.138456517 -0.65849203
9 TRUE 2.1608968+0.5833807i 0.2262562 1.271366406 -2.23159156
10 FALSE 0.5956258+0.2673282i -1.4675688 -0.016984227 1.06334080
11 FALSE 1.5650520+0.3968731i -0.5541264 -0.135241342 0.05793721
12 TRUE 1.4445524+0.3118638i 1.0983463 1.781934098 -0.22806050
13 TRUE 1.6063870+0.3436187i 1.3681229 1.268863189 -2.14581499
14 TRUE -1.4068157+0.2399498i 0.2005744 -0.963995714 -1.38571628
15 FALSE 2.2837193+0.3538850i 0.5910446 0.160374068 -0.18793847
16 FALSE 1.4940472+0.9729339i 0.8732281 0.511570707 0.09455187
17 FALSE -1.2086423+0.8382830i 0.3314982 -1.126031052 -0.07339069
18 TRUE 0.4870967+0.9581304i 0.8242927 0.394865156 -0.23565899
19 TRUE 0.6865523+0.6611055i 0.7790229 -0.238627823 -0.92214415
20 TRUE 0.9580948+0.1250858i 0.5909648 -0.755950206 0.86695967

```

Μπορούν επίσης να χρησιμοποιηθούν οι εντολές `cbind` και `rbind` για να δημιουργηθεί ένα πλαίσιο δεδομένων μαζί με άλλες επιλογές.

```

> my.dataframe2 <- cbind(1, my.dataframe)
> my.dataframe2
  1 my.logic      my.complex my.numeric      X1      X2
1 1 TRUE 0.1782105+0.9851267i 0.4452555 0.650290036 1.52145790
2 1 TRUE -2.6140989+0.3541577i 0.4751909 -1.059263140 -0.09996308
3 1 FALSE -0.0767501+0.2550082i 0.9357701 0.216042514 1.14707512
4 1 FALSE -0.7275827+0.3569999i 1.5757764 -0.114622892 0.59689871
5 1 FALSE -0.3280398+0.5163078i -1.0958323 -0.004433541 1.21214093
6 1 FALSE -0.7451964+0.8069675i 2.1620200 -0.978986416 -0.60250469
7 1 TRUE -0.6853192+0.5144991i -1.1306062 -0.609778169 0.68110679
8 1 FALSE -0.4147151+0.5529729i -0.4240480 0.138456517 -0.65849203
9 1 TRUE 2.1608968+0.5833807i 0.2262562 1.271366406 -2.23159156
10 1 FALSE 0.5956258+0.2673282i -1.4675688 -0.016984227 1.06334080
11 1 FALSE 1.5650520+0.3968731i -0.5541264 -0.135241342 0.05793721
12 1 TRUE 1.4445524+0.3118638i 1.0983463 1.781934098 -0.22806050
13 1 TRUE 1.6063870+0.3436187i 1.3681229 1.268863189 -2.14581499
14 1 TRUE -1.4068157+0.2399498i 0.2005744 -0.963995714 -1.38571628
15 1 FALSE 2.2837193+0.3538850i 0.5910446 0.160374068 -0.18793847
16 1 FALSE 1.4940472+0.9729339i 0.8732281 0.511570707 0.09455187
17 1 FALSE -1.2086423+0.8382830i 0.3314982 -1.126031052 -0.07339069
18 1 TRUE 0.4870967+0.9581304i 0.8242927 0.394865156 -0.23565899

```

```
19 1 TRUE 0.6865523+0.6611055i 0.7790229 -0.238627823 -0.92214415
20 1 TRUE 0.9580948+0.1250858i 0.5909648 -0.755950206 0.86695967
```

Κάποιες άλλες εντολές οι οποίες είναι χρήσιμες είναι οι ακόλουθες :

```
> length(my.dataframe)
[1] 5
> dim(my.dataframe)
[1] 20 5
> is.data.frame(my.dataframe)
[1] TRUE
> is.list(my.dataframe)
[1] TRUE
> is.matrix(my.dataframe)
[1] TRUE
> is.vector(my.dataframe)
[1] FALSE
> names(my.dataframe)
[1] "my.logic" "my.complex" "my.numeric" "X1" "X2"
```

Τι κάνει η κάθε μία από τις παραπάνω εντολές; Οι συναρτήσεις `attach` και `detach` είναι πολύ χρήσιμες όταν αναλύεται ένα συγκεκριμένο πλαίσιο δεδομένων.

Η εντολή

```
attach(my.dataframe)
```

τοποθετεί το πλαίσιο δεδομένων στο περιβάλλον εργασίας πρώτο και έτσι οι μεταβλητές του πλαισίου μπορούν να επεξεργαστούν ή να χρησιμοποιηθούν απ' ευθείας.

```
> my.logic
[1] TRUE TRUE FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE TRUE
[13] TRUE TRUE FALSE FALSE FALSE TRUE TRUE TRUE
> my.complex
[1] 0.1782105+0.9851267i -2.6140989+0.3541577i -0.0767501+0.2550082i
[4] -0.7275827+0.3569999i -0.3280398+0.5163078i -0.7451964+0.8069675i
[7] -0.6853192+0.5144991i -0.4147151+0.5529729i 2.1608968+0.5833807i
[10] 0.5956258+0.2673282i 1.5650520+0.3968731i 1.4445524+0.3118638i
[13] 1.6063870+0.3436187i -1.4068157+0.2399498i 2.2837193+0.3538850i
```

[16] $1.4940472+0.9729339i$ $-1.2086423+0.8382830i$ $0.4870967+0.9581304i$
[19] $0.6865523+0.6611055i$ $0.9580948+0.1250858i$

Για να φύγει το πλαίσιο δεδομένων από το περιβάλλον εργασίας χρησιμοποιείται η συνάρτηση detach.