

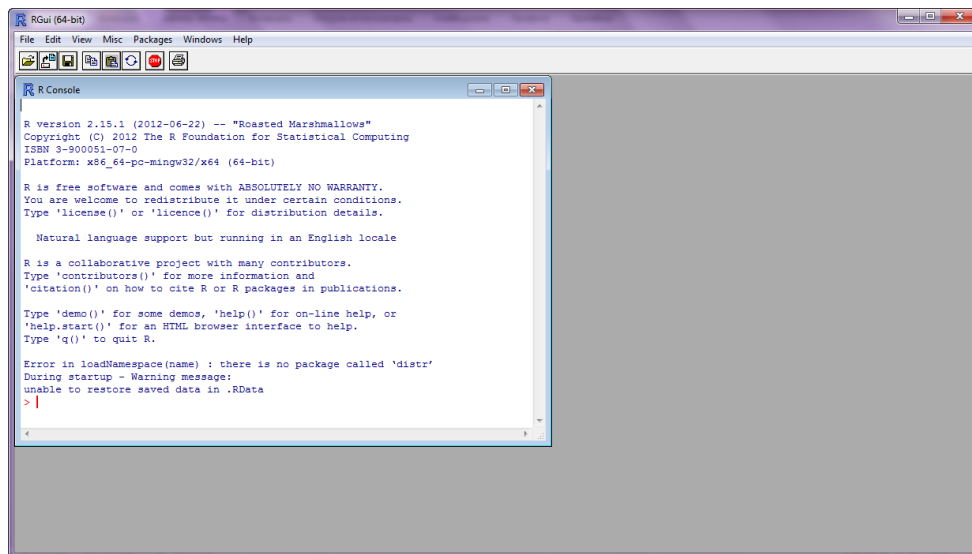
3 Εισαγωγή στην R

Στο κεφάλαιο αυτό θα παρουσιάσουμε τις βασικές έννοιες, την εισαγωγή και επεξεργασία δεδομένων, την εξαγωγή αποτελεσμάτων και τέλος τις γραφικές παραστάσεις που υλοποιούνται με τη γλώσσα προγραμματισμού R. Οι δυνατότητες της R παρουσιάζονται σε δύο διαφορετικά περιβάλλοντα εργασίας, στη βασική έκδοση της R και στο RStudio. Η διεπαφή γραμμής εντολών είναι το κοινό χαρακτηριστικό των δύο ψηφιακών εφαρμογών, όμως το RStudio παρέχει επιπλέον πλεονεκτήματα για την αποδοτικότερη αξιοποίηση της R.

3.1 Η βασική έκδοση της R

Αρχικά, θα παρουσιάσουμε τις οδηγίες εγκατάστασης της βασικής έκδοσης της R, η οποία διανέμεται ελεύθερα από το διαδικτυακό τόπο με διεύθυνση <http://www.r-project.org>. Για να αποθηκεύσουμε το αρχείο εγκατάστασης στον υπολογιστή μας, επιλέγουμε CRAN από το αριστερό τμήμα της ιστοσελίδας και ακολούθως κάποιο από τα προσφερόμενα mirror sites, ενδεχομένως του Πανεπιστημίου της Κρήτης. Κατόπιν, επιλέγουμε [Download R for Windows](#) ή τον αντίστοιχο σύνδεσμο αν διαθέτουμε κάποιο άλλο λειτουργικό σύστημα (Linux και MacOS X) και στο επόμενο παράθυρο [install R for the first time](#). Τέλος, επιλέγουμε [Download R 2.15.1 for Windows](#) και αποθήκευση στον υπολογιστή μας, του αρχείου R-2.15.1-win.exe. Η διαδικασία εγκατάστασης ξεκινά με διπλό αριστερό κλικ πάνω στο όνομα του παραπάνω αρχείου και ολοκληρώνεται ακολουθώντας τις αναγραφόμενες σε κάθε βήμα οδηγίες. Επισημαίνεται ότι, αν το λειτουργικό σύστημα του υπολογιστή μας είναι Windows – 64bit, τότε μπορούμε να εγκαταστήσουμε και τις δύο εκδόσεις της R (32bit και 64bit). Αν το λειτουργικό σύστημα του υπολογιστή μας είναι Windows – 32bit πρέπει να εγκαταστήσουμε μόνο την 32bit έκδοση της R. Για να ελέγξουμε ποια έκδοση των Windows χρησιμοποιεί ο υπολογιστής μας επιλέγουμε **Έναρξη/Πίνακας Ελέγχου/Σύστημα** ή πατάμε δεξί κλικ πάνω στο εικονίδιο **Υπολογιστής** (συνήθως βρίσκεται στην επιφάνεια εργασίας) και επιλέγουμε **Ιδιότητες**.

Για να εκκινήσουμε την R, επιλέγουμε **Έναρξη/Όλα τα Προγράμματα/R/R 2.15.1**. Η αρχική οθόνη της R έχει την παρακάτω μορφή:



Η βασική έκδοση της R διαθέτει τα μενού **File**, **Edit**, **View**, **Misc**, **Packages**, **Windows** και **Help**. Όλα τα μενού χρησιμοποιούνται για να γίνουν ρυθμίσεις του περιβάλλοντος της βασικής έκδοσης της R και δεν περιέχουν εντολές για τη στατιστική ανάλυση και επεξεργασία δεδομένων. Στη συνέχεια θα αναφερθούμε σε ορισμένες από τις δυνατότητες που παρέχουν αυτά τα μενού. Τα εικονίδια στην αμέσως επόμενη από τα μενού γραμμή, αντιστοιχούν σε κάποιες από τις πιο συνηθισμένες εντολές που εμφανίζονται σε επεξεργαστές κειμένου (όπως αντιγραφή, επικόλληση, εκτύπωση) και δεν κρίνεται σκόπιμο να αναλυθούν περεταίρω.

3.2 Πληροφορικός αλφαριθμητισμός για την R

Η R είναι μια γλώσσα προγραμματισμού στην οποία οι εντολές διαβάζονται και κατόπιν εκτελούνται αμέσως. Η επικοινωνία του χρήστη με την R διεξάγεται μέσω της κονσόλας της R (R Console). Οι εντολές δίνονται σε κάθε γραμμή μετά την ένδειξη **>** και τα αποτελέσματά τους εμφανίζονται στην επόμενη γραμμή μετά το πάτημα του πλήκτρου Enter. Τα συστατικά στοιχεία κάθε εντολής της R είναι συνδυασμός των παρακάτω εκφραστικών μορφών:

- αριθμητική (numeric)
- κειμένου (character)
- λογική (logical)

Στις επόμενες υποενότητες θα παρουσιαστούν και οι τρεις παραπάνω μορφές.

3.2.1 Οι αριθμητικοί υπολογισμοί

Οι αριθμητικές πράξεις στην R γίνονται με τη βοήθεια των συμβόλων που βρίσκονται στον παρακάτω πίνακα:

Σύμβολο	Αριθμητική πράξη
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση
^	Ύψωση σε δύναμη
%/%	Ακέραιο πηλίκo
%%	Ακέραιο υπόλοιπο

Αν για παράδειγμα θέλουμε το αποτέλεσμα της πρόσθεσης του 3 με το 5, πληκτρολογούμε 3+5 και πατάμε το πλήκτρο Enter, δηλαδή

```
> 3+5 [Enter]
```

Στην επόμενη γραμμή εμφανίζεται το αποτέλεσμα της πράξης που είναι 8.

```
[1] 8
```

Η ένδειξη [1] δηλώνει το πλήθος των εξαγομένων αποτελεσμάτων, που στην περίπτωση μας είναι μοναδικό και ίσο με 8. Ομοίως (στα υπόλοιπα παραδείγματα η εντολή Enter θα παραλείπεται, όπου αυτό δε δημιουργεί σύγχυση):

```
> 2-7
```

```
[1] -5
```

```
> 3*8
```

```
[1] 24
```

```
> 24/3
```

```
[1] 8
```

Η δύναμη 2^3 ισούται με $2 \cdot 2 \cdot 2 = 8$. Στην R γράφεται

```
> 2^3
```

```
[1] 8
```

και για επιβεβαίωση

```
> 2*2*2
```

```
[1] 8
```

Αν στην ακέραια διαίρεση, διαιρέσουμε το 23 με το 7 προκύπτει πηλίκο 3 και υπόλοιπο 2. Πράγματι,

```
> 23%/%7
```

```
[1] 3
```

```
> 23%%7
```

```
[1] 2
```

Οι αριθμητικοί υπολογισμοί εμπλουτίζονται στην R με τη βοήθεια ορισμένων γνωστών μαθηματικών συναρτήσεων, οι οποίες είναι εγκατεστημένες και μπορούν να κληθούν οποιαδήποτε στιγμή από το χρήστη. Οι πιο απλές από αυτές είναι η απόλυτη τιμή ενός αριθμού, που συμβολίζεται με `abs`, π.χ.

```
> abs(3)
```

```
[1] 3
```

```
> abs(-3)
```

```
[1] 3
```

και η τετραγωνική ρίζα ενός αριθμού, που συμβολίζεται με `sqrt`, π.χ.

```
> sqrt(25)
```

```
[1] 5
```

```
> 2+sqrt(2)
```

```
[1] 3.414214
```

Η R διαθέτει επίσης πληθώρα άλλων συναρτήσεων που διευκολύνουν την επεξεργασία δεδομένων, όπως η εντολή `round` για τη στρογγυλοποίηση δεκαδικών αριθμών. Για παράδειγμα η εντολή

```
> round(2+sqrt(2), digits=3)
```

```
[1] 3.414
```

στρογγυλοποιεί το αποτέλεσμα της αριθμητική πράξης $2+\sqrt{2}$ σε τρία δεκαδικά ψηφία. Αν ο χρήστης θέλει να μάθει περισσότερα στοιχεία για κάποια εντολή (π.χ. `round`) αρκεί να πληκτρολογήσει

```
> ?round
```

ή

```
> help(round)
```

ή

από το μενού **Help** να επιλέξει **R functions (text)...**, στο εμφανιζόμενο πλαίσιο της οποίας να πληκτρολογήσει `round` και θα ανοίξει μια σελίδα στο πρόγραμμα περιήγησης του Internet (όπως ο Internet Explorer και ο Mozilla Firefox) με όλες τις σχετικές πληροφορίες στην Αγγλική γλώσσα.

3.2.2 Οι εκφράσεις κειμένου

Σε ορισμένες περιπτώσεις επεξεργασίας δεδομένων είναι απαραίτητες οι εκφράσεις με περιεχόμενο κάποιους χαρακτήρες κειμένου, όπως ένα γράμμα της αλφαβήτου, μία ή περισσότερες λέξεις. Στην R οι εκφράσεις κειμένου περιέχονται μεταξύ εισαγωγικών. Συνήθη παραδείγματα αποτελούν η δήλωση του φύλου ή του επαγγέλματος κάποιου ατόμου. Έτσι, το φύλο δηλώνεται με `"male"` ή `"female"` ενώ το επάγγελμα π.χ. με `"teacher"`, `"policeman"` κλπ.

Η R διαθέτει εντολές κειμένου που περιέχουν τα γράμματα του λατινικού αλφαβήτου (μικρά ή κεφαλαία) και τους μήνες του χρόνου στην αγγλική γλώσσα. Έτσι

```
> letters
```

```
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p"  
[17] "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"
```

ή

```
> LETTERS
```

```
[1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P"  
[17] "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"
```

και

```
> month.name
```

```
[1] "January" "February" "March" "April" "May" "June"  
[7] "July" "August" "September" "October" "November" "December"
```

Το αποτέλεσμα της εντολής `month.name` περιέχει 12 στοιχεία, όσοι και οι μήνες του χρόνου. Η πρώτη γραμμή του αποτελέσματος ξεκινά από το μήνα `"January"` που βρίσκεται μετά από το `[1]` και τελειώνει στο μήνα `"June"`, που είναι ο έκτος μήνας του έτους. Το `[7]` μπροστά από τον `"July"` δηλώνει ότι η αναγραφή των μηνών συνεχίζεται από τον έβδομο μήνα και συνεχίζεται μέχρι τέλους.

3.2.3 Οι λογικές εκφράσεις

Λογική έκφραση στην R είναι μια οποιαδήποτε έκφραση που μπορεί να χαρακτηριστεί αληθής (`TRUE`) ή ψευδής (`FALSE`). Οι λογικές εκφράσεις συντάσσονται με τη βοήθεια των συμβόλων που βρίσκονται στον πίνακα που ακολουθεί.

Σύμβολο	Αριθμητική πράξη
---------	------------------

<	Μικρότερο από
<=	Μικρότερο ή ίσο από
>	Μεγαλύτερο από
>=	Μεγαλύτερο ή ίσο από
==	Ακριβώς ίσο με
!=	Όχι ίσο με

Για παράδειγμα, η λογική έκφραση «ο αριθμός 3 είναι μεγαλύτερος από τον αριθμό 1», κωδικοποιείται στην R με την έκφραση `3>1`, η οποία είναι προφανώς αληθής.

```
> 3>1
[1] TRUE
```

Αντίθετα,

```
> 2+sqrt(2)==3
[1] FALSE
```

3.3 Αντικείμενα δεδομένων

Είναι ενδεδειγμένο η ανάλυση και επεξεργασία δεδομένων στην R να διενεργείται μέσω μεταβλητών, οι οποίες στη γλώσσα της R ονομάζονται αντικείμενα δεδομένων (data objects). Έτσι, το αποτέλεσμα του αριθμητικού υπολογισμού $2 + \sqrt{2}$ μπορεί να καταχωρηθεί στο αντικείμενο δεδομένων `x`, μέσω του συμβόλου ανάθεσης `<-`, ως εξής:

```
x <- 2+sqrt(2) [Enter]
```

Για επιβεβαίωση πληκτρολογούμε `x` για να δούμε την τιμή του.

```
> x
[1] 3.414214
```

Τα αντικείμενα δεδομένων αποθηκεύονται στο χώρο εργασίας (Workspace) της R, ο οποίος δεν είναι ορατός στη βασική έκδοση της R. Για να δούμε το περιεχόμενό του αρκεί να πληκτρολογήσουμε

```
> objects()
```

ή

```
> ls()
```

ή να επιλέξουμε από το μενού **Misc** την εντολή **List objects**, για να έχουμε έναν κατάλογο με όλα τα αντικείμενα δεδομένων. Στην περίπτωση που δεν έχουμε

καταχωρημένο κανένα αντικείμενο τότε η απόκριση της R είναι:

```
character(0)
```

Οι σημαντικότερες μορφές αντικειμένων που χρησιμοποιεί η R είναι το διάνυσμα, ο πίνακας, ο παράγοντας και το πλαίσιο δεδομένων.

3.3.1 Διανύσματα

Το πιο απλό είδος αντικειμένου δεδομένων είναι το διάνυσμα (vector). Το διάνυσμα είναι ένα διατεταγμένο σύνολο τιμών, δηλαδή μία σειρά αριθμών ή χαρακτήρων κειμένου ή λογικών εκφράσεων ή συνδυασμός όλων αυτών των μορφών. Αν για παράδειγμα θέλουμε να εισάγουμε τα αριθμητικά δεδομένα 15, 12, 14, 11, -2 στην R, ο πιο εύκολος τρόπος είναι με τη συνάρτηση `c` (το αρχικό γράμμα από τη λέξη concatenation, που σημαίνει μια σειρά από γεγονότα, ιδέες ή αντικείμενα τα οποία είναι συνδεδεμένα). Έτσι,

```
> y <- c(15, 12, 14, 11, -2)
> y
[1] 15 12 14 11 -2
```

Επισημαίνεται ότι τα κενά ανάμεσα στα κόμματα δεν έχουν καμία επίδραση στον ορισμό του `y`. Το `y` είναι ένα διάνυσμα που αποτελείται από **αριθμητικά δεδομένα**, επομένως

```
> mode(y)
[1] "numeric"
```

Επίσης, το `y` είναι ένα διάνυσμα που διαθέτει πέντε διατεταγμένα στοιχεία, ή το **μήκος** (length) του διανύσματος είναι 5. Έτσι,

```
> length(y)
[1] 5
```

Το αντικείμενο δεδομένων της μεταβλητής `x`, που ορίστηκε παραπάνω, είναι ένα διάνυσμα με μήκος 1. Πράγματι,

```
> length(x)
[1] 1
```

Για το διάνυσμα `y` δεν έχουμε ορίσει κάποια **ετικέτα**, επομένως

```
> names(y)
NULL
```

Το `NULL` παριστάνει στην R το αντικείμενο δεδομένων που δεν περιέχει τίποτα.

Για να εμφανίσουμε κάποια μόνο από τα στοιχεία ενός αντικειμένου δεδομένων, πληκτρολογούμε το όνομα του αντικειμένου και μέσα στις αγκύλες `[]`

κάποιον αριθμό που αντιστοιχεί στη σειρά του στοιχείου. Για παράδειγμα, για να εμφανίσουμε το δεύτερο στοιχείο του `y` και τα τρία τελευταία στοιχεία του `y` πληκτρολογούμε αντίστοιχα

```
> y[2]
[1] 12
> y[3:5]
[1] 14 11 -2
```

Το όρισμα `3:5` προκαλεί τη σειρά των αριθμών από το 3 έως το 5, με αύξηση κατά 1. Πράγματι,

```
> 3:5
[1] 3 4 5
> -5:30
[1] -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
[21] 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
```

Για να προσθέσουμε π.χ. το 2 ή να υψώσουμε στο τετράγωνο κάθε στοιχείο του `y`, απλώς πληκτρολογούμε

```
> y+2
[1] 17 14 16 13 0
```

ή

```
> y^2
[1] 225 144 196 121 4
```

Μια άλλη δυνατότητα για να κατασκευάζουμε διανύσματα, ειδικά στην περίπτωση που γίνεται επανάληψη κάποιων τιμών, δίνεται με τη βοήθεια της συνάρτησης `rep`. Η συνάρτηση `rep()` καθορίζει πόσες φορές θα γίνει η επανάληψη κάποιων τιμών, με το όρισμα `times`.

```
> rep(3, times=5)
[1] 3 3 3 3 3
> rep( c(-1,2), times=3 )
[1] -1 2 -1 2 -1 2
```

Αν κατά τη διαδικασία πληκτρολόγησης έγινε κάποιο λάθος, π.χ. το πέμπτο στοιχείο του `y` θέλαμε να είναι 2, ενώ πληκτρολογήσαμε -2, τότε μπορούμε να επαναλάβουμε τον ορισμό με τα σωστά στοιχεία, π.χ.

```
> y <- c(15, 12, 14, 11, 2)
> y
[1] 15 12 14 11 2
```

ή απλώς να αντικαταστήσουμε μόνο το πέμπτο στοιχείο του `y` που είναι λανθασμένο:

```
> y[5]=2
```

```
> y
[1] 15 12 14 11 2
```

Επιπλέον, να επισημάνουμε ότι μπορούμε να χρησιμοποιούμε τελείες στην ονοματοδοσία των αντικειμένων δεδομένων. Για παράδειγμα, αν οι τιμές του `y` αντιστοιχούν στους βαθμούς πέντε μαθητών στα μαθηματικά, ένα κατάλληλο όνομα θα ήταν

```
> maths.grades <- c(15, 12, 14, 11, 2)
> maths.grades
[1] 15 12 14 11 2
```

Εκτός, από τα αριθμητικά διανύσματα μπορούμε να ορίσουμε διανύσματα **χαρακτήρων κειμένου ή λογικών εκφράσεων**. Για παράδειγμα,

```
> gender <- c( "male", "female", "female", "male", "male" )
> gender
[1] "male"  "female" "female" "male"  "male"
> mode(gender)
[1] "character"
> length(gender)
[1] 5
```

και

```
> y.men <- gender=="male"
```

Η λογική έκφραση `gender=="male"` είναι αληθής για όσα στοιχεία του διανύσματος `gender` έχουν τιμή ίση με `"male"`. Επομένως,

```
> y.men
[1] TRUE FALSE FALSE TRUE TRUE
> mode(y.men)
[1] "logical"
```

Αν θέλουμε να επεξεργαστούμε στην R ένα αντικείμενο δεδομένων σε περιβάλλον που έχει τη μορφή λογιστικού φύλλου, πρέπει καταρχήν να δηλώσουμε τι είδους αντικείμενο θέλουμε να κατασκευάσουμε, π.χ. ένα διάνυσμα `x`

```
> x <- vector()
```

το οποίο αρχικά δεν περιέχει κανένα στοιχείο. Εισάγουμε τα στοιχεία του διανύσματος μετά την εντολή

```
> data.entry(x)
```

στο παρακάτω λογιστικό φύλλο

	x	var2	var3
1	5		
2	-6		
3	80		
4	6		
5	-8		
6			
7			
8			
9			
10			
11			
12			

στη στήλη με τίτλο **x**. Η μετάβαση από κελί σε κελί γίνεται με Enter ή με το ποντίκι. Όταν ολοκληρώσουμε την εισαγωγή των στοιχείων κλείνουμε το παράθυρο. Αν θέλουμε να δούμε το νέο διάνυσμα **x** σε μορφή λογιστικού φύλλου πληκτρολογούμε `> View(x)`

και θα προκύψει ένα νέο παράθυρο της μορφής:

	x
1	5
2	-6
3	80
4	6
5	-8

3.3.2 Πίνακες

Οι πίνακες (matrices) χρησιμοποιούνται για να διατάξουμε τις τιμές ενός αντικειμένου δεδομένων κατά γραμμές και στήλες. Υπάρχουν διάφοροι τρόποι για να ορίσουμε έναν πίνακα. Ο πιο συνηθισμένος είναι μέσω της εντολής `matrix`.

```
> z <- matrix( c(1,2,3, 4,5,6), nrow = 2, ncol=3, byrow=TRUE )
> z
```

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

Το πρώτο όρισμα `c(1,2,3, 4,5,6)` περιλαμβάνει όλα τα στοιχεία που θέλουμε να τοποθετήσουμε στον πίνακα **z**. Τα δύο επόμενα ορίσματα `nrow = 2, ncol=3`

καθορίζουν τις διαστάσεις του πίνακα, δηλαδή τον αριθμό των γραμμών (στην περίπτωση μας 2) και τον αριθμό των στηλών (στην περίπτωση μας 3). Η παράμετρος `byrow` καθορίζει αν τα περιεχόμενα του πίνακα τοποθετηθούν κατά γραμμές. Αν το όρισμα αυτό ισούται με `TRUE`, όπως συμβαίνει στο παραπάνω παράδειγμα τότε τα στοιχεία `1,2,3` θα τοποθετηθούν στην πρώτη γραμμή και τα στοιχεία `4,5,6` θα τοποθετηθούν στη δεύτερη γραμμή. Ένας άλλος τρόπος σχηματισμού πίνακα γίνεται με τη βοήθεια της εντολής `rbind`. Συγκεκριμένα, δίνουμε ως ορίσματα στην εντολή αυτή κάθε γραμμή του πίνακα, που θέλουμε να σχηματίσουμε, με τη μορφή διανύσματος. Για το προηγούμενο παράδειγμα,

```
> z <- rbind( c(1,2,3), c(4,5,6) )  
> z
```

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

Ο πίνακας `z` περιέχει **αριθμητικά δεδομένα** οπότε

```
> mode(z)  
[1] "numeric"
```

Με τη γνωστή εντολή `length` δίνεται το σύνολο όλων των στοιχείων του πίνακα, ενώ με την εντολή `dim` οι **διαστάσεις** του πίνακα, δηλαδή το πλήθος των γραμμών και των στηλών του.

```
> length(z)  
[1] 6  
> dim(z)  
[1] 2 3
```

Αν θέλουμε να εμφανιστεί κάποιο συγκεκριμένο στοιχείο του πίνακα `z` πρέπει να καθοριστεί η γραμμή και η στήλη στην οποία ανήκει. Έτσι, με

```
> z[2,3]  
[1] 6
```

δηλώνουμε το στοιχείο του πίνακα `z` που βρίσκεται στη δεύτερη γραμμή (πρώτο όρισμα) και στην τρίτη στήλη (δεύτερο όρισμα). Επιπλέον, η ένδειξη `[,2]` αντιστοιχεί σε όλα τα στοιχεία της δεύτερης στήλης και η ένδειξη `[1,]` σε όλα τα στοιχεία της πρώτης γραμμής. Δηλαδή,

```
> z[,2]  
[1] 2 5
```

και

```
> z[1,]
```

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

Για να **ονομάσουμε** τις γραμμές και τις στήλες ενός πίνακα χρησιμοποιούμε τις εντολές `rownames` και `colnames`. Έτσι, για τον πίνακα `z`, δίνουμε τις ονομασίες Γ1, Γ2 για τις γραμμές και Σ1, Σ2, Σ3 για τις στήλες, ως εξής:

```
> rownames(z) <- c("Γ1", "Γ2")
> colnames(z) <- c("Σ1", "Σ2", "Σ3")
> z
      Σ1 Σ2 Σ3
Γ1    1  2  3
Γ2    4  5  6
```

Με παρόμοιο με τα διανύσματα τρόπο μπορούμε να δημιουργήσουμε πίνακες με **χαρακτήρες κειμένου** ή **λογικές εκφράσεις**, π.χ. με την εντολή `z>2` δηλώνουμε ότι τα στοιχεία του πίνακα `z` είναι μεγαλύτερα του 2, κάτι που είναι αληθές μόνο για ορισμένα από αυτά.

```
> z.log <- z>2
> z.log
      [,1] [,2] [,3]
[1,] FALSE FALSE TRUE
[2,]  TRUE  TRUE TRUE
> mode(z.log)
[1] "logical"
```

3.3.3 Παράγοντες

Σε αρκετές περιπτώσεις στην ανάλυση και επεξεργασία δεδομένων μας ενδιαφέρει η ομαδοποίηση των τιμών ενός αντικειμένου δεδομένων. Σε αυτή την περίπτωση οι ομάδες τιμών του αντικειμένου ονομάζονται **κατηγορίες** (levels) και το αντικείμενο δεδομένων **παράγοντας** (factor). Έτσι, το διάνυσμα `gender` μπορεί να γίνει παράγοντας με κατηγορίες `"male"` και `"female"`, μέσω της εντολής

```
> factor(gender)
[1] male  female female
Levels: female male
```

Σε αριθμητικά δεδομένα ένας παράγοντας μπορεί να κατασκευαστεί και με την εφαρμογή της εντολής `cut`, όπως για το διάνυσμα

```
> x <- c(9, 11, 9, 14, 16, 19, 17, 11, 10, 18)
> x
[1]  9 11  9 14 16 19 17 11 10 18
```

Η εντολή `cut` καθορίζει τα όρια των κατηγοριών του αντικειμένου `x`, μέσω του ορίσματος `breaks`.

```
> x.factor <- cut( x, breaks=c(8, 12, 16, 20) )
> x.factor
[1] (8,12] (8,12] (8,12] (12,16] (12,16] (16,20] (16,20] (8,12]
[9] (8,12] (16,20]
Levels: (8,12] (12,16] (16,20]
```

Το αντικείμενο δεδομένων `x.factor` είναι πλέον ένας παράγοντας με κατηγορίες `(8,12]`, `(12,16]` και `(16,20]`. Αυτό σημαίνει ότι όποιο στοιχείο του διανύσματος `x` είναι μεγαλύτερο από το 8 και μικρότερο ή ίσο από το 12 εντάσσεται στην πρώτη ομάδα τιμών `(8,12]`, μεγαλύτερο από το 12 και μικρότερο ή ίσο από το 16 εντάσσεται στη δεύτερη ομάδα τιμών `(12,16]` και τέλος μεγαλύτερο από το 16 και μικρότερο ή ίσο από το 20 εντάσσεται στην τρίτη ομάδα τιμών `(16,20]`. Έτσι η ακολουθία

```
[1] (8,12] (8,12] (8,12] (12,16] (12,16] (16,20] (16,20] (8,12]
[9] (8,12] (16,20]
```

σημαίνει ότι τα τρία πρώτα στοιχεία του `x`, δηλαδή τα 9, 11 και 9 ανήκουν στο `(8,12]`, τα επόμενα δύο, δηλαδή τα 14 και 16 στο `(12,16]` και ούτω καθεξής.

3.3.4 Πλαίσια δεδομένων

Τα πλαίσια δεδομένων (data frames) είναι αντικείμενα στα οποία κάθε στήλη είναι ένα διάνυσμα του ιδίου μήκους, αλλά ενδεχομένως διαφορετικού τύπου (αριθμητικό, κείμενο, λογικό). Για να κατασκευάσουμε ένα πλαίσιο δεδομένων καλούμε την εντολή `data.frame`. Για παράδειγμα, το πλαίσιο δεδομένων `students` περιέχει 5 παρατηρήσεις (μαθητές) για καθεμία από τις οποίες καταγράφουμε το φύλο και την επίδοση στα μαθηματικά. Έτσι, η πρώτη παρατήρηση αντιστοιχεί σε ένα μαθητή με επίδοση 15, η δεύτερη σε μια μαθήτρια με επίδοση 12 και ούτω καθεξής.

```
> students <- data.frame( gender, maths.grades )
> students
  gender maths.grades
1  male           15
2 female           12
3 female           14
```

```
4 male 11
5 male 2
```

Παρατηρούμε ότι η πρώτη στήλη είναι ένα διάνυσμα χαρακτήρων κειμένου, ενώ η δεύτερη ένα αριθμητικό διάνυσμα. Τα δύο διανύσματα έχουν ίδιο μήκος, που ισούται με 5. Κάποιες χρήσιμες εντολές για τα πλαίσια δεδομένων είναι οι ακόλουθες:

```
> length(students)
[1] 2
> dim(students)
[1] 5 2
> names(students)
[1] "gender" "maths.grades"
```

Για να επεξεργαστούμε κάποια από τις στήλες του `students` χρησιμοποιούμε το σύμβολο `$` και το όνομα της στήλης, ως εξής:

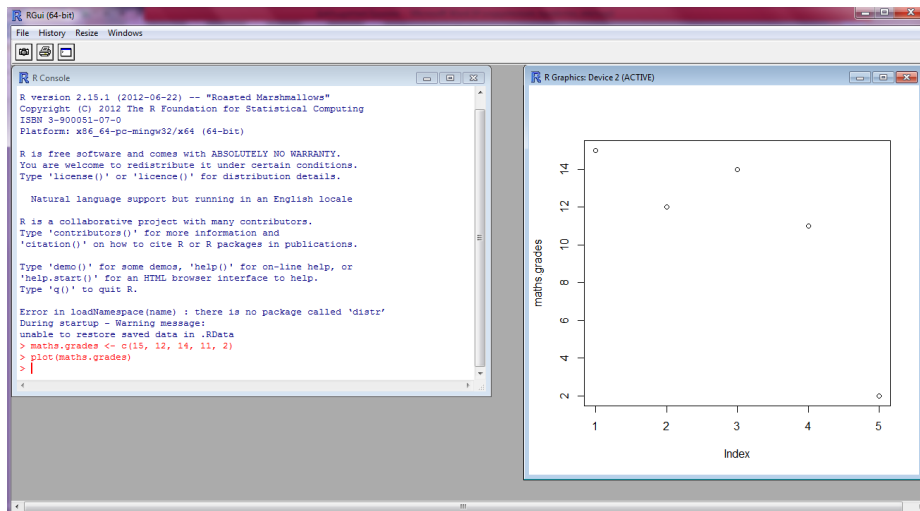
```
> students$maths.grades
[1] 15 12 14 11 2
```

3.4 Γραφικές παραστάσεις

Μια άλλη πολύ σημαντική δυνατότητα της R, που είναι θεμελιώδης για τη στατιστική, είναι η δημιουργία γραφικών παραστάσεων. Μία από τις βασικότερες εντολές για τη δημιουργία γραφημάτων είναι η `plot`. Έτσι, αν θέλουμε να κάνουμε τη γραφική παράσταση των τιμών του διανύσματος `maths.grades`, δίνουμε την εντολή

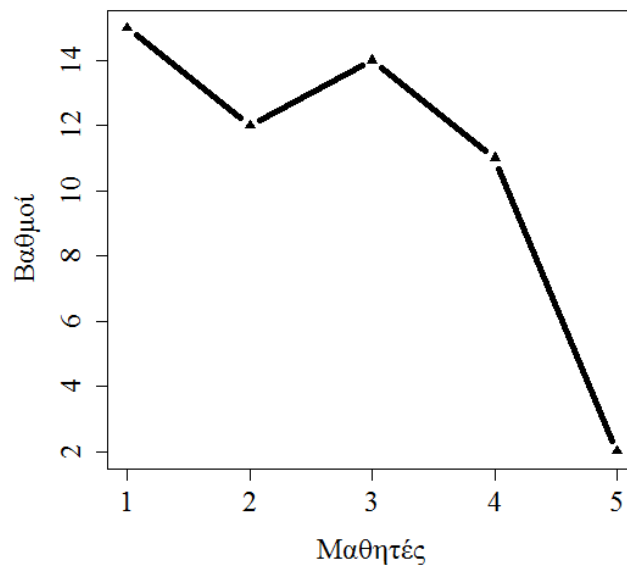
```
> plot(maths.grades)
```

με αποτέλεσμα η R να δημιουργήσει αυτόματα ένα παράθυρο με το παραγόμενο γράφημα (Σχήμα 5).



Σχήμα 5. Περιβάλλον εργασίας της R μετά τη δημιουργία γραφήματος.

Στον οριζόντιο άξονα με τον τίτλο `Index` έχουμε τις τιμές που εκφράζουν τη διάταξη κάθε τιμής του `maths.grades` (από 1 έως 5), ενώ στον κατακόρυφο άξονα έχουμε τις τιμές του διανύσματος, στην προκειμένη περίπτωση τους βαθμούς. Έτσι, τα σημεία του γραφήματος αντιστοιχούν στα ζεύγη (1,15), (2,12), (3,14), (4,11) και (5,2). Ο χαρακτήρας με τον οποίο συμβολίζουμε το κάθε σημείο μπορεί να αλλάξει με την παράμετρο `pch`, η οποία μπορεί να πάρει οποιαδήποτε ακέραια τιμή από το μηδέν μέχρι και το 18 (προκαθορισμένη τιμή το 1). Σε κάθε τιμή αντιστοιχεί ένα διαφορετικό σύμβολο. Αν επιπλέον θέλουμε τα σημεία του γραφήματος να συνδέονται με ευθύγραμμα τμήματα, τότε πρέπει να χρησιμοποιήσουμε το όρισμα `type="b"`. Η δυνατότητα αυτή θα μας ενδιέφερε αν το διάνυσμα `maths.grades` περιείχε τους πέντε βαθμούς ενός μαθητή στη διάρκεια μιας χρονικής περιόδου. Η γραμμή σε αυτή την περίπτωση αναπαριστά τη χρονική εξέλιξη της επίδοσης. Το όρισμα `lwd=5` καθορίζει το πλάτος της γραμμής σε 5, από 1 που είναι η προκαθορισμένη τιμή της R. Για να τροποποιήσουμε τις ετικέτες των αξόνων εμπλουτίζουμε τα ορίσματα της εντολής `plot` με `xlab="Μαθητές"` και `ylab="Βαθμοί"`.



Σχήμα 6. Το γράφημα του διανύσματος `maths.grades` μετά από τροποποιήσεις στις παραμέτρους της εντολής `plot`.

Αν θέλουμε να τροποποιήσουμε το μέγεθος της γραμματοσειράς, τότε προσθέτουμε `cex.lab=1.5` (για τις ετικέτες των αξόνων), `cex.axis=1.5` (για το μέγεθος των αριθμών στους άξονες). Η γραμματοσειρά που θα χρησιμοποιήσουμε δηλώνεται με το όρισμα `family`. Τελικά, η εντολή

```
> plot(maths.grades, type="b", lwd=5, xlab="Μαθητές", ylab="Βαθμοί",
cex.lab=1.5, cex.axis=1.5, family="serif")
```

μας δίνει το παρακάτω αποτέλεσμα του Σχήματος 6.

Οι πολλαπλές δυνατότητες της εντολής `plot` θα παρουσιαστούν εκτενέστερα στο 5^ο κεφάλαιο. Επίσης, η R διαθέτει μια τεράστια ποικιλία γραφικών παραστάσεων, εκτός από την `plot`, στις οποίες θα αναφερθούμε σε επόμενα κεφάλαια.

3.5 Στοιχειώδης προγραμματισμός με την R

Οι εντολές που δώσαμε μέχρι τώρα στην R αποκαλούνται και συναρτήσεις (functions), επειδή δέχονται ορίσματα που διαφοροποιούν το αποτέλεσμα της εντολής. Για παράδειγμα, η εντολή `plot` δέχεται διάφορα ορίσματα για το είδος της γραμμής, τις ετικέτες των αξόνων, τη γραμματοσειρά που παράγουν διαφορετικό κάθε φορά αποτέλεσμα. Οι συναρτήσεις που χρησιμοποιούνται στην R χωρίζονται σε δύο κατηγορίες. Τις ενσωματωμένες συναρτήσεις (built-in functions) που διαθέτει η γλώσσα προγραμματισμού και αυτές που μπορεί να δημιουργήσει ο χρήστης. Παρόλο

που η R διαθέτει μια ευρύτατη συλλογή στατιστικών κυρίως συναρτήσεων εντούτοις πολλές φορές είναι πολύ χρήσιμη η δυνατότητα, ο ίδιος ο χρήστης να δημιουργεί νέες συναρτήσεις που είναι προσαρμοσμένες στις δικές του ανάγκες και προτιμήσεις.

Για να κατανοήσουμε τον τρόπο δημιουργίας και εφαρμογής των συναρτήσεων, θα ασχοληθούμε με την έννοια του μέσου όρου κάποιων τιμών, που είναι ίσως η πιο συνηθισμένη στατιστική συνάρτηση. Ας υποθέσουμε ότι θέλουμε να υπολογίσουμε το μέσο όρο των τιμών του διανύσματος `maths.grades` που ορίσαμε παραπάνω. Η μέση τιμή αυτών των τιμών υπολογίζεται από το πηλίκο του αθροίσματος των τιμών με το πλήθος τους, δηλαδή το πέντε. Επομένως, δίνεται από τη σχέση

$$\frac{15+12+14+11+2}{5} = \frac{54}{5} = 10.8.$$

Ο μέσος όρος υπολογίζεται στην R με τη συνάρτηση `mean`.

```
> mean (maths.grades)
[1] 10.8
```

Το ίδιο αποτέλεσμα μπορούμε να επιτύχουμε με μια συνάρτηση που θα κατασκευάσουμε εμείς και θα έχει το όνομα `mean.1`. Ο κώδικας εντολών της συνάρτησης είναι:

```
> mean.1 <- function(x) [Enter]
+ { [Enter]
+ (x[1]+x[2]+x[3]+x[4]+x[5]) / 5 [Enter]
+ } [Enter]
```

Με την πρώτη γραμμή κατασκευάζουμε το αντικείμενο της συνάρτησης με όνομα `mean.1`, ως μια συνάρτηση της μεταβλητής `x` και πατάμε **Enter**. Η R εντοπίζει ότι δεν έχει ολοκληρωθεί η συνάρτηση, διότι οι εντολές οποιασδήποτε συνάρτησης πρέπει να περιέχονται μεταξύ δύο αγκίστρων `{ }`. Για αυτό και εμφανίζει στην επόμενη γραμμή το `+` που υποδηλώνει ότι αναμένει από το χρήστη να συμπληρωθούν οι υπόλοιπες εντολές της συνάρτησης. Ακολουθώντας πληκτρολογούμε `{` και πατάμε και πάλι **Enter**. Μετά από το `+` της τρίτης γραμμής δίνουμε τον τύπο του μέσου όρου των πέντε τιμών της μεταβλητής `x`. Το άθροισμα μέσα στην παρένθεση αποτελεί τον αριθμητή του κλάσματος και το 5 τον παρονομαστή. Τέλος, η συνάρτηση ολοκληρώνεται με το άγκιστρο `}`. Αν για οποιοδήποτε λόγο κάνουμε λάθος κατά την πληκτρολόγηση ο καλύτερος τρόπος διόρθωσης είναι η επανάληψη της πληκτρολόγησης. Επίσης, η πληκτρολόγηση θα μπορούσε να γίνει σε μία μόνο γραμμή, ως εξής:

```
> mean.1 <- function(x) { (x[1]+x[2]+x[3]+x[4]+x[5]) / 5 }
```

Με την εντολή `objects` διαπιστώνουμε ότι το αντικείμενο `mean.1` έχει προστεθεί στο Workspace. Για να καλέσουμε τη νέα συνάρτηση να υπολογίσει τη μέση τιμή του διανύσματος `maths.grades` πληκτρολογούμε

```
> mean.1(maths.grades)
[1] 10.8
```

Η συνάρτηση `mean.1` είναι κατάλληλη για τον υπολογισμό της μέσης τιμής του `maths.grades`, όμως έχει ένα βασικό μειονέκτημα. Υπολογίζει το άθροισμα των πρώτων πέντε στοιχείων της μεταβλητής `x`, ακόμα και αν αυτή έχει περισσότερα στοιχεία. Επιπλέον, ο παρονομαστής του κλάσματος παραμένει πέντε, ανεξάρτητα από το μήκος του διανύσματος `x`. Για παράδειγμα το διάνυσμα `maths.grades.6` έχει τα πέντε στοιχεία του `maths.grades` και επιπλέον τον αριθμό 18. Δηλαδή

```
> maths.grades.6 <- c(15, 12, 14, 11, 2, 18)
```

με νέο μέσο όρο το 12. Πράγματι,

```
> mean(maths.grades.6)
[1] 12
```

Παρόλα αυτά η συνάρτηση `mean.1` δίνει λανθασμένο αποτέλεσμα που προκύπτει από τις πρώτες πέντε τιμές του `maths.grades.6`.

```
> mean.1(maths.grades.6)
[1] 10.8
```

Για να διορθώσουμε το λάθος κατασκευάζουμε τη συνάρτηση `mean.2`, ως εξής:

```
> mean.2 <- function(x)
+ {
+   sum(x) / length(x)
+ }
```

Στη συνάρτηση αυτή έχουμε αντικαταστήσει τον αριθμητή του κλάσματος με το άθροισμα όλων των στοιχείων της μεταβλητής `x` (με την εντολή `sum(x)`), ανεξάρτητα από το μήκος της και τον παρονομαστή με το πλήθος των στοιχείων του `x` (με την εντολή `length(x)`). Το σκεπτικό της συνάρτησης `mean.2` είναι ότι αντικαταστήσαμε τις «στατικές» εντολές της συνάρτησης `mean.1` με τις «δυναμικές» εντολές της συνάρτησης `mean.2`. Έτσι,

```
> mean.2(maths.grades.6)
[1] 12
```

Το μεγάλο πλεονέκτημα των γλωσσών προγραμματισμού, όπως η R, είναι η σταδιακή ανάπτυξη (incremental development). Δηλαδή, μια συνάρτηση (όπως η `mean.2`) μπορεί να δημιουργηθεί από τις ενσωματωμένες συναρτήσεις της R (όπως οι

`sum` και `length`), να εκτελεστεί και μετά να δημιουργηθεί μία άλλη συνάρτηση που να στηρίζεται σε αυτήν και ούτω καθ' εξής. Με τον τρόπο αυτόν απλοποιείται ο κώδικας εντολών των νέων συναρτήσεων και έτσι μπορούμε να επεξεργαστούμε με σχετική ευκολία πιο πολύπλοκες συναρτήσεις.

Ολοκληρώνοντας τις κατασκευές των συναρτήσεων για τον υπολογισμό της μέσης τιμής των συναρτήσεων θέλουμε να διαγράψουμε τις συναρτήσεις `mean.1` και `mean.2` από το Workspace, ώστε να χρησιμοποιούμε μόνο την εντολή `mean`. Αυτό γίνεται με την εντολή

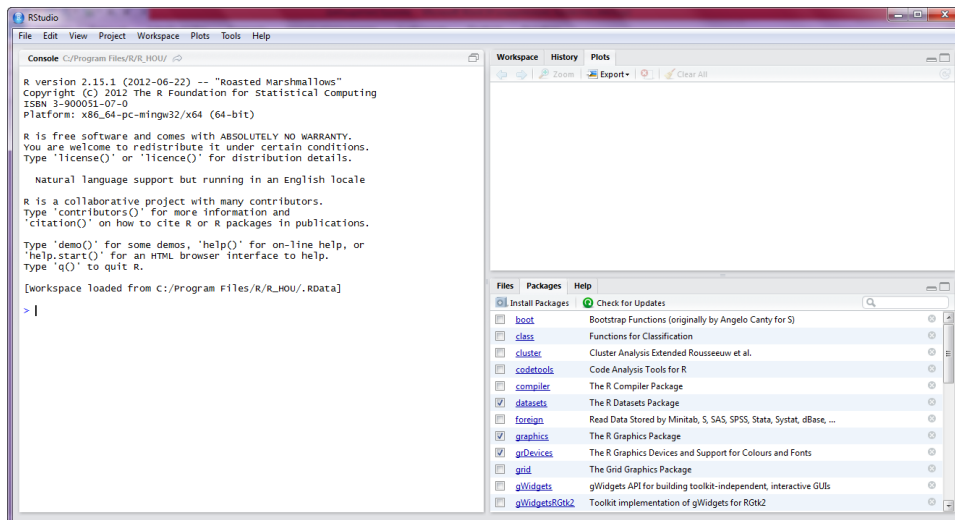
```
> rm(mean.1, mean.2)
```

3.6 RStudio

Το RStudio είναι μια εφαρμογή ανοικτού κώδικα που διανέμεται δωρεάν. Το RStudio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Enviroment - IDE) της R (Verzani, 2011). Διανέμεται ελεύθερα από την ιστοσελίδα του, στη διεύθυνση <http://www.rstudio.org/>.

Αρχικά, θα δώσουμε τις οδηγίες εγκατάστασης και λειτουργίας του RStudio. Επισημαίνουμε ότι, για την κανονική λειτουργία του RStudio είναι απαραίτητη η εγκατάσταση της βασικής έκδοσης της R (2.11.1 ή οποιαδήποτε νεώτερη από αυτή). Για να αποθηκεύσουμε το αρχείο εγκατάστασης στον υπολογιστή, επιλέγουμε Download RStudio (στο άνω δεξί τμήμα της ιστοσελίδας) και ακολούθως Download RStudio Desktop. Κατόπιν, αποθηκεύουμε το αρχείο Recommended For Your System στον υπολογιστή μας (RStudio-0.96.331.exe – Windows XP/Vista/7). Η διαδικασία εγκατάστασης ξεκινά με διπλό αριστερό κλικ πάνω στο όνομα του παραπάνω αρχείου και ολοκληρώνεται ακολουθώντας τις αναγραφόμενες σε κάθε βήμα οδηγίες. Η διαδικασία εγκατάστασης περιγράφεται αναλυτικά με κείμενο και δύο βιντεοδιαλέξεις στη διεύθυνση <http://users.sch.gr/gkaravasilis/installation.html>.

Για να εκκινήσουμε το RStudio στο λειτουργικό σύστημα των Windows, επιλέγουμε **Έναρξη/Όλα τα Προγράμματα/RStudio/RStudio**. Η αρχική οθόνη της έκδοσης 0.96.316, έχει περίπου την παρακάτω μορφή:



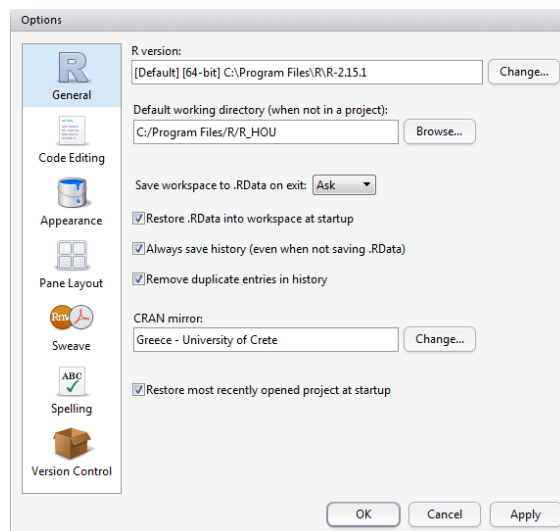
Το RStudio διαθέτει τα μενού **File**, **Edit**, **View**, **Project**, **Workspace**, **Plots**, **Tools** και **Help**. Όλα τα μενού χρησιμοποιούνται για να γίνουν ρυθμίσεις του περιβάλλοντος εργασίας του RStudio και δεν περιέχουν εντολές για τη στατιστική ανάλυση και επεξεργασία δεδομένων. Στη συνέχεια θα αναφερθούμε σε ορισμένες από τις δυνατότητες που παρέχουν αυτά τα μενού.

Όλες οι εντολές της R που παρουσιάστηκαν στις προηγούμενες ενότητες μπορούν να χρησιμοποιηθούν στην κονσόλα του RStudio, στο αριστερό ορθογώνιο πλαίσιο της οθόνης (Console), όπως ακριβώς χρησιμοποιούνται και στην κονσόλα της βασικής έκδοσης της R. Το δεξί τμήμα της οθόνης είναι χωρισμένο σε δύο ομάδες παραθύρων. Στο πλαίσιο που βρίσκεται άνω δεξιά έχουμε τον κατάλογο αντικειμένων της R, δηλαδή το Workspace, έναν κατάλογο με τις τελευταίες εντολές τις οποίες πληκτρολογήσαμε στην κονσόλα (History) και ένα πλαίσιο για την εμφάνιση των γραφημάτων (Plots). Στο πλαίσιο που βρίσκεται κάτω δεξιά παρατηρούμε το παράθυρο με τα αρχεία που δημιουργεί ο χρήστης (Files), τον κατάλογο των εγκατεστημένων πακέτων (Packages) που θα αναλύσουμε στην επόμενη ενότητα και τη βοήθεια της εφαρμογής (Help). Η παρουσίαση των δυνατοτήτων και λειτουργιών του RStudio που ακολουθεί θα χρησιμοποιεί την παραπάνω διάταξη. Η διάταξη των παραθύρων αυτών μπορεί να τροποποιηθεί από την επιλογή **Options** στο μενού **Tools**. Στη συνέχεια επιλέγοντας **Pane Layout** μπορούμε να διαμορφώσουμε το περιβάλλον που μας ταιριάζει.

Το RStudio, όπως και η βασική έκδοση της R, δεν έχει κάποιο μενού εντολών που θα διευκόλυνε την ανάλυση και επεξεργασία δεδομένων, διαθέτει όμως ένα περιβάλλον με όλες τις βασικές λειτουργίες της R που επιτρέπουν τον εύκολο

χειρισμό των δεδομένων, των συναρτήσεων και των αποτελεσμάτων τους. Ένα από τα πλεονεκτήματα του RStudio σε σχέση με τη βασική έκδοση της R είναι η δημιουργία και η επεξεργασία του κώδικα εντολών των συναρτήσεων. Έχουμε ήδη αναφέρει ότι η δημιουργία μιας συνάρτησης υλοποιείται στην κονσόλα της βασικής έκδοσης της R, ενώ η επεξεργασία της μπορεί να γίνει με τον ενσωματωμένο editor που διαθέτει. Παρόλα αυτά, η δημιουργία και κυρίως η επεξεργασία συναρτήσεων με το RStudio είναι πολύ πιο εύχρηστη και λειτουργική.

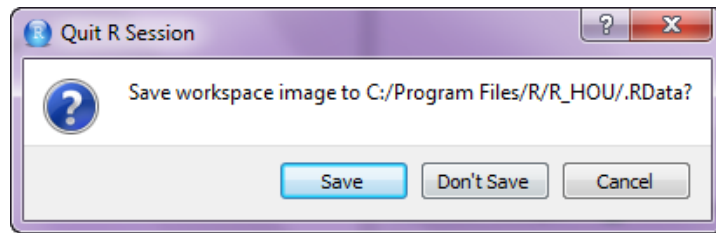
Κατ' αρχήν, επιλέγουμε τον κατάλογο εργασίας (working directory), από το μενού **Tools**, την επιλογή **Options** και την ενότητα **R General**, π.χ. C:/Program Files/R/R_HOU και πατάμε το **OK**, όπως φαίνεται στο παρακάτω σχήμα:



Για να επιβεβαιώσουμε ότι αποθηκεύτηκε ο επιθυμητός κατάλογος εργασίας, πληκτρολογούμε στην κονσόλα:

```
> getwd()  
[1] "C:/Program Files/R/R_HOU"
```

Ο κατάλογος εργασίας είναι ένας φάκελος στον οποίο αποθηκεύεται όλες οι εργασίες του χρήστη. Το Workspace, με όλα τα αντικείμενα δεδομένων, το ιστορικό των εντολών που δόθηκαν στην κονσόλα και όλα τα αρχεία που περιέχουν συναρτήσεις ή γραφήματα. Για αυτό και όταν κλείνουμε το RStudio αποθηκεύουμε τις εργασίες μας απαντώντας καταφατικά στο σχετικό μήνυμα:



Για να δημιουργήσουμε την πρώτη μας συνάρτηση με το RStudio, επιλέγουμε **File/New/R Script** και εμφανίζεται μία κενή σελίδα στο αριστερό τμήμα της οθόνης στην οποία καταχωρούμε τον κώδικα εντολών της συνάρτησης. Για παράδειγμα, η συνάρτηση `example.sum`, που υπολογίζει το άθροισμα των στοιχείων του διάνυσματος `x` αφού πρώτα αφαιρεθεί από αυτά η μέση τους τιμή `mean(x)`.

```
example.sum <- function(x)
{
  sum( x-mean(x) )
}
```

Έτσι, για το διάνυσμα `maths.grades` η εντολή `x-mean(x)` υπολογίζει τις διαφορές $15-10.8=4.2$, $12-10.8=1.2$, $14-10.8=3.2$, $11-10.8=0.2$, $2-10.8=-8.8$, που έχουν άθροισμα μηδέν. Πραγματικά,

$$4.2+1.2+3.2+0.2+(-8.8)=0.$$

Η χρησιμότητα της συνάρτησης `example.sum` έγκειται στη δυνατότητα να εφαρμόσουμε αυτή τη διαδικασία όχι μόνο για το διάνυσμα `maths.grades`, αλλά για οποιοδήποτε διάνυσμα τεθεί στη θέση του `x`, χωρίς να επαναλαμβάνουμε κάθε φορά τον κώδικα εντολών της συνάρτησης. Προφανώς η χρησιμότητα αυτή για οποιαδήποτε συνάρτηση αυξάνεται καθώς αυξάνεται η έκταση και κυρίως η πολυπλοκότητα του κώδικα εντολών.

Για να αποθηκεύσουμε και κατόπιν να εκτελέσουμε την παραπάνω συνάρτηση επιλέγουμε το πλαίσιο Source on Save και μετά τη δισκέτα που βρίσκεται λίγο πιο αριστερά. Στο παράθυρο που ανοίγει εμφανίζονται τα περιεχόμενα του working directory, στο οποίο το RStudio προτείνει να αποθηκεύσουμε το αρχείο μας. Πληκτρολογούμε το όνομα του αρχείου που επιθυμούμε στο σχετικό πλαίσιο και μετά πατάμε το κουμπί Αποθήκευση, ώστε να ολοκληρωθεί η αποθήκευση του αρχείου στο working directory, π.χ. `example_sum` (τονίζεται ότι στο όνομα του αρχείου δεν χρησιμοποιούμε τελείες). Αυτό προκαλεί την εμφάνιση στην κονσόλα της εντολής

```
> source('C:/Program Files/R/R_HOU/example_sum.R')
```

και του ονόματος της συνάρτησης στο Workspace. Επίσης, στο παράθυρο με τίτλο Files εμφανίζεται το όνομα του αρχείου με την προσθήκη της επέκτασης .R, δηλαδή `example_sum.R`. Η δημιουργία της συνάρτησης έχει ολοκληρωθεί και απομένει η εκτέλεση της, που μπορεί να γίνει με την εντολή (στην κονσόλα)

```
> example.sum(x=maths.grades)
[1] -3.552714e-15
```

Με την εντολή αυτή ζητάμε την εκτέλεση της συνάρτησης `example.sum`, αν στη θέση του `x` αντικαταστήσουμε το διάνυσμα `maths.grades`. Το αποτέλεσμα είναι

-0.0000000000000003552714

και δίνεται λόγω της έκτασης του στη λεγόμενη εκθετική μορφή αριθμών, δηλαδή $-3.552714 \cdot 10^{-15}$. Το αποτέλεσμα είναι αναπάντεχο, διότι γνωρίζουμε εκ των προτέρων ότι η σωστή τιμή είναι το μηδέν. Παρόλα αυτά, επειδή το αποτέλεσμα παρουσιάζει ελάχιστη απόκλιση από το μηδέν, δεν επηρεάζει την εξαγωγή συμπερασμάτων. Η εξήγηση αυτής της απόκλισης οφείλεται στον τρόπο αποθήκευσης των αριθμών από τους υπολογιστές και δεν είναι ασύνηθες φαινόμενο (Braun & Murdoch 2007, σελ. 9-11). Για παράδειγμα

```
> 0.1+0.2-0.3
[1] 5.551115e-17
```

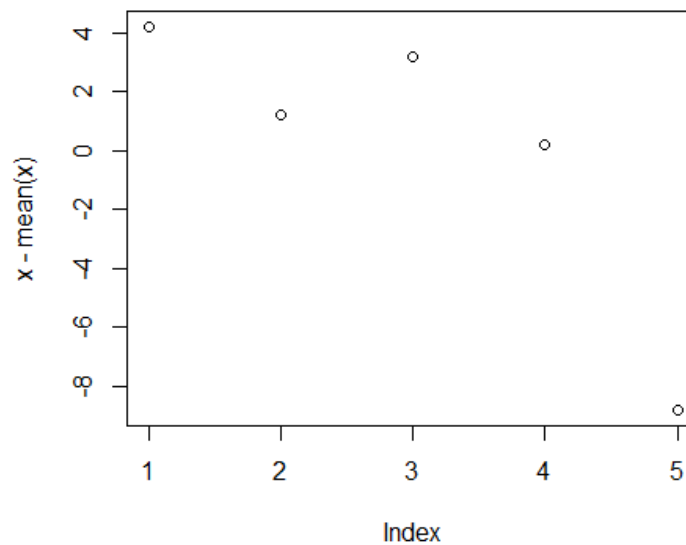
Αν θελήσουμε να τροποποιήσουμε κάποιες από τις εντολές της συνάρτησης `example.sum`, π.χ.

```
example.sum <- function(x)
{
  plot ( x-mean(x) )
}
```

επεμβαίνουμε στο ανοικτό αρχείο και επαναλαμβάνουμε τη διαδικασία. Αποθηκεύουμε το αρχείο με το πάτημα της δισκέτας και πληκτρολογούμε στην κονσόλα:

```
> example.sum(maths.grades)
```

Το αποτέλεσμα πλέον είναι ένα γράφημα των διαφορών `x-mean(x)`, που εμφανίζεται στην ενότητα **Plots** (Σχήμα 7).



Σχήμα 7. Γράφημα των διαφορών `x - mean(x)`, που δημιουργήθηκε με την εκτέλεση της συνάρτησης `example.sum`.

Αν θέλουμε να αποθηκεύσουμε το γράφημα για μελλοντική χρήση ή για να το ενσωματώσουμε σε ένα αρχείου κειμένου, επιλέγουμε Export και Save Plot as Image ή Save Plot as PDF. Σε κάθε περίπτωση εμφανίζεται ένα παράθυρο με τα περιεχόμενα του working directory για να επιλέξουμε τις διαστάσεις του γραφήματος, τον τύπο της εικόνας (Image format) και το όνομα του αρχείου. Μετά την ολοκλήρωση της αποθήκευσης το όνομα του αρχείου εμφανίζεται στο παράθυρο **Files**. Τέλος, αν θέλουμε να επεξεργαστούμε τη συνάρτηση σε μια επόμενη είσοδο στο RStudio, αρκεί η επιλογή του αντίστοιχου αρχείου από το παράθυρο **Files** (με αριστερό κλικ πάνω στο όνομά του αρχείου) για να εμφανιστεί ο κώδικας εντολών στο άνω αριστερά πλαίσιο του RStudio.

3.7 Πακέτα επέκτασης: εγκατάσταση και ενεργοποίηση

Όταν εγκαθίσταται η R σε έναν υπολογιστή ενεργοποιούνται αυτόματα ορισμένα πακέτα (packages), που περιέχουν τις βασικότερες εντολές για την ανάλυση και επεξεργασία δεδομένων. Τα πακέτα είναι συλλογές συναρτήσεων και δεδομένων τα οποία έχουν συνήθως κάποιο κοινό σημείο αναφοράς, όπως για παράδειγμα το πακέτο `graphics` που προσανατολίζεται στη δημιουργία γραφικών παραστάσεων.

Μπορούμε να δούμε τα πακέτα αυτά αν πληκτρολογήσουμε

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

Υπάρχουν όμως συναρτήσεις της R οι οποίες παρόλο που είναι πολύ χρήσιμες ανήκουν σε κάποιο πακέτο που δεν ενεργοποιείται αρχικά και απαιτείται μια επιπλέον διαδικασία για την ενεργοποίησή του. Θα δώσουμε τώρα τις απαραίτητες οδηγίες για την εγκατάσταση και ενεργοποίηση των πακέτων επέκτασης δυνατοτήτων (π.χ. TeachingDemos, pmg, Rcmdr και άλλα) στη βασική έκδοση της R και στο RStudio, σε υπολογιστές που διαθέτουν το λειτουργικό σύστημα των Windows. Επισημαίνεται ότι οι οδηγίες αυτές μπορεί να χρειάζονται, κατά την εφαρμογή τους σε κάποιο υπολογιστή, μικρές και συνήθως προφανείς τροποποιήσεις λόγω των διαφορετικών εκδόσεων των Windows που κυκλοφορούν. Απαραίτητες προϋποθέσεις σε κάθε περίπτωση αποτελούν:

- η σύνδεση του υπολογιστή με το Internet και
- οι ρυθμίσεις ελέγχου λογαριασμού χρήστη. Η ρύθμιση γίνεται από την ακολουθία επιλογών: **Έναρξη\Πίνακας Ελέγχου\Όλα τα στοιχεία του Πίνακα Ελέγχου\Λογαριασμοί χρηστών** και **Αλλαγή ρυθμίσεων ελέγχου λογαριασμού χρήστη**, στο οποίο επιλέγουμε **Να μην ειδοποιούμαι ποτέ** (όταν επιχειρούνται αλλαγές στον υπολογιστή μας από πιθανώς επιβλαβή προγράμματα), εννοώντας ότι δεν θα έχουμε μήνυμα προειδοποίησης στην οθόνη μας, όταν επιχειρούμε να εγκαταστήσουμε ένα λογισμικό ή μια εφαρμογή στον υπολογιστή μας, συνήθως από το Internet. Κατόπιν κάνουμε επανεκκίνηση του υπολογιστή μας και ανοίγουμε τη βασική έκδοση της R ή το RStudio.

Η λειτουργία των πακέτων μπορεί να γίνει είτε με τη βασική έκδοση της R είτε με το RStudio. Παρακάτω θα δοθούν όλα τα βήματα που απαιτούνται για τη λειτουργία τους και στις δύο περιπτώσεις.

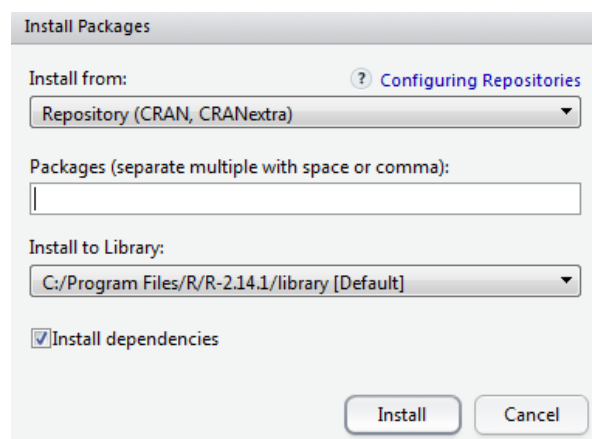
- **Βασική έκδοση της R**. Μετά την έναρξη της R, από το μενού **Packages** επιλέγουμε **Install package(s)...** και από το εμφανιζόμενο παράθυρο με τίτλο **CRAN mirror**, επιλέγουμε κάποιο από τα mirror sites της R σε διάφορες χώρες του κόσμου, για παράδειγμα της Ελλάδας (**Greece**) και πατάμε το κουμπί **OK**. Στο επόμενο παράθυρο με τίτλο **Packages** αναζητούμε οποιοδήποτε πακέτο επιθυμούμε (π.χ. το TeachingDemos), το επιλέγουμε με αριστερό κλικ και

πατάμε το κουμπί **OK**. Αυτόματα, θα εκκινήσει η εγκατάσταση του πακέτου στον υπολογιστή μας και όλων των άλλων πακέτων που ενδεχομένως υποστηρίζουν τη λειτουργία του. Στο τέλος της διαδικασίας, για να δούμε όλα τα πακέτα που είναι εγκατεστημένα στον υπολογιστή μας, πληκτρολογούμε

```
> library()
```

και εμφανίζεται ένα παράθυρο με τα διαθέσιμα πακέτα, ανάμεσα στα οποία πρέπει να υπάρχει και το TeachingDemos. Όλα αυτά τα πακέτα είναι εγκατεστημένα αλλά όχι ενεργοποιημένα. Η ενεργοποίηση γίνεται αν επιλέξουμε ξανά το μενού **Packages** και μετά **Load package**. Από το παράθυρο **Select one** επιλέγουμε με αριστερό κλικ το πακέτο που θέλουμε (π.χ. το TeachingDemos) και μετά πατάμε **OK**.

- **RStudio**. Μετά την έναρξη του RStudio, στο κάτω δεξιά τμήμα της οθόνης του RStudio επιλέγουμε, με απλό αριστερό κλικ, την ταμπέλα **Packages**. Κατόπιν, στο παράθυρο που ανοίγει επιλέγουμε την εντολή **Install Packages** που βρίσκεται ακριβώς κάτω από την ταμπέλα **Packages**. Στην οθόνη του υπολογιστή εμφανίζεται ένα σχετικά μικρό παράθυρο, όπως αυτό που φαίνεται παρακάτω:



Στο παράθυρο αυτό πρέπει να συμπληρώσουμε μόνο τη δεύτερη γραμμή με το όνομα του πακέτου που θέλουμε να εγκαταστήσουμε (π.χ. TeachingDemos). Πατάμε ένα αριστερό κλικ στη δεύτερη γραμμή, ώστε να ενεργοποιηθεί ο cursor (η κατακόρυφη κάθετη γραμμή που αναβοσβήνει), πληκτρολογούμε TeachingDemos και πατάμε το κουμπί **Install**. Το RStudio ξεκινά τη διαδικασία εγκατάστασης του πακέτου, η οποία όταν ολοκληρωθεί θα έχει προστεθεί, στη λίστα πακέτων που είναι

προεγκατεστημένα, το TeachingDemos. Τα εγκατεστημένα πακέτα εμφανίζονται στο κάτω δεξί μέρος του RStudio, στην ταμπέλα **Packages**. Για την ενεργοποίηση τους χρειάζεται απλώς να τσεκάρουμε το αντίστοιχο κουτί μπροστά από το όνομα του πακέτου.