

# Μια (πολύ) σύντομη εισαγωγή στην R

Paul Torfs & Claudia Brauer

Ομάδα Υδρολογίας & Ποσοτικής Διαχείρισης Υδάτων  
Πανεπιστήμιο Wageningen, Ολλανδία

4 Νοεμβρίου 2014

<http://www.r-project.org/>

και εκτελέστε τα ακόλουθα (υποθέτοντας ότι δουλεύετε σε έναν υπολογιστή με Windows):

- κάντε κλικ στο download CRAN στην αριστερή στήλη
- επιλέξτε έναν ιστότοπο απ' όπου θα γίνει η λήψη
- επιλέξτε Windows ως λειτουργικό σύστημα
- κάντε κλικ στο base
- επιλέξτε Download R 3.0.3 for Windows<sup>2</sup> και αφήστε τις προεπιλεγμένες απαντήσεις σε όλες τις ερωτήσεις

## 1 Εισαγωγή

Η R είναι μια ισχυρή γλώσσα και ένα ισχυρό περιβάλλον ανάπτυξης για στατιστικούς υπολογισμούς και γραφικά. Ως έργο αποτελεί κοινό κτήμα (ή αλλιώς, είναι GNU project), και είναι παρόμοια με την εμπορική γλώσσα και περιβάλλον S, που είχε αναπτυχθεί στα Bell Laboratories (πρώην AT&T, πλέον Lucent Technologies) από τον John Chambers και τους συνεργάτες του. Η R μπορεί να θεωρηθεί πως είναι μια διαφορετική υλοποίηση της S, και χρησιμοποιείται ευρέως ως εκπαιδευτική γλώσσα και ως ερευνητικό εργαλείο.

Τα κύρια πλεονεκτήματα της R είναι το γεγονός ότι η R είναι ελεύθερο λογισμικό και ότι υπάρχει πολύ βοήθεια διαθέσιμη στο διαδίκτυο. Είναι αρκετά παρόμοια με άλλα προγραμματιστικά πακέτα όπως η MATLAB (που δεν είναι ελεύθερο λογισμικό), αλλά πιο φιλική προς τον χρήστη από γλώσσες προγραμματισμού όπως η C++ και η Fortran. Μπορείτε να χρησιμοποιήσετε την R όπως είναι, αλλά για εκπαιδευτικούς λόγους εμείς προτιμούμε τη χρήση της R σε συνδυασμό με την διεπαφή του RStudio (που είναι κι αυτό ελεύθερο λογισμικό), το οποίο έχει μια οργανωμένη διάταξη και διάφορες πρόσθετες επιλογές.

Το παρόν έγγραφο περιλαμβάνει επεξηγήσεις, παραδείγματα και ασκήσεις, τα οποία (ελπίζουμε ότι) μπορούν να γίνουν κατανοητά από ανθρώπους που δεν έχουν κάποια προγραμματιστική εμπειρία. Το διάβασμα όλου του κειμένου και των ασκήσεων απαιτεί περίπου 1 με 2 ώρες. Παραδείγματα από εντολές, που χρησιμοποιούνται συχνά, και από μηνύματα λάθους παρατίθενται στις τελευταίες δύο σελίδες αυτού του εγγράφου και μπορούν να χρησιμοποιηθούν ως αναφορά κατά τον προγραμματισμό.

## 2 Ξεκινώντας

### 2.1 Εγκατάσταση της R

Για να εγκαταστήσετε την R στον υπολογιστή σας (δωρεάν και με νόμιμο τρόπο!), πηγαίνετε στην αρχική σελίδα του ιστοτόπου της R<sup>1</sup>:

<sup>1</sup>Στον ιστότοπο της R μπορείτε επίσης να βρείτε και το παρόν έγγραφο: <http://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf>

Είναι επίσης δυνατόν να εκτελέσετε την R και το RStudio μέσω ενός USB αντί να τα εγκαταστήσετε. Αυτό θα μπορούσε να φανεί χρήσιμο όταν δεν έχετε δικαιώματα διαχειριστή στον υπολογιστή σας. Δείτε και την ξεχωριστή σημείωσή μας "How to use portable versions of R and RStudio" για βοήθεια στο συγκεκριμένο θέμα.

### 2.2 Εγκατάσταση του RStudio

Μετά την ολοκλήρωση της εγκατάστασης, θα πρέπει να βλέπετε ένα εικονίδιο "R" στην επιφάνεια εργασίας σας. Κάνοντας κλικ σε αυτό θα εκκινήσετε την τυπική διεπαφή. Εμείς θα σας συνιστούσαμε, ωστόσο, να χρησιμοποιήσετε την διεπαφή του RStudio.<sup>3</sup> Για να εγκαταστήσετε το RStudio, πηγαίνετε στον ιστότοπο:

<http://www.rstudio.org/>

και εκτελέστε τα ακόλουθα βήματα (υποθέτοντας ότι δουλεύετε σε έναν υπολογιστή με Windows):

- κάντε κλικ στο Download RStudio
- κάντε κλικ στο Download RStudio Desktop
- κάντε κλικ στο Recommended For Your System
- κατεβάστε το εκτελέσιμο .exe αρχείο και τρέξτε το (αφήστε τις προεπιλεγμένες απαντήσεις για όλες τις ερωτήσεις)

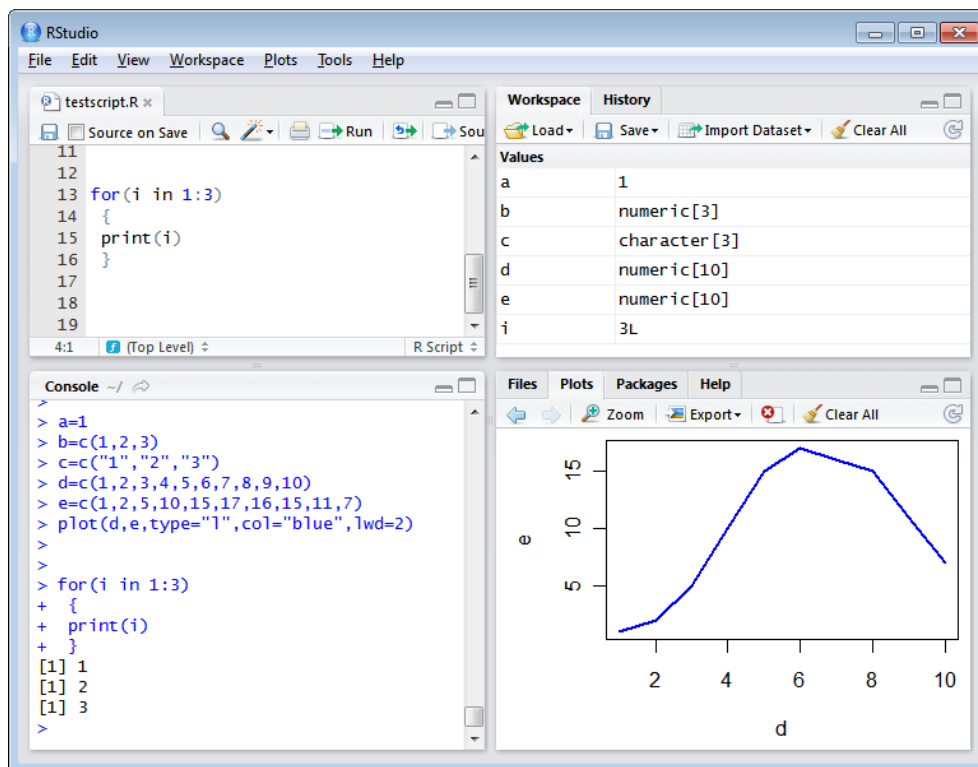
### 2.3 Η διάταξη του RStudio

Η διεπαφή του RStudio αποτελείται από διάφορα παράθυρα (βλέπε το Σχήμα 1).

- Κάτω αριστερά: **παράθυρο κονσόλας** (καλείται επίσης και **παράθυρο εντολών**). Εδώ μπορείτε να εισάγετε απλές εντολές μετά το σύμβολο υποβολής ">" και η R στη συνέχεια θα εκτελέσει την εντολή σας. Αυτό είναι το πιο σημαντικό παράθυρο, επειδή στην πραγματικότητα εκεί τρέχει η R.
- Πάνω αριστερά: **παράθυρο επεξεργαστή κειμένου** (καλείται επίσης και **παράθυρο σεναρίων**). Εδώ μπορούν να υποστούν επεξεργασία και να

<sup>2</sup>Τη στιγμή της συγγραφής του κειμένου αυτού η τελευταία έκδοση ήταν η 3.0.3. Επιλέξτε την πιο πρόσφατη.

<sup>3</sup>Υπάρχουν πολλές άλλες διεπαφές (ελεύθερου λογισμικού), όπως το Tinn-R.



**Σχήμα 1:** Τα παράθυρα του επεξεργαστή κειμένου (editor), του χώρου εργασίας (workspace), της κονσόλας (console) και των γραφικών παραστάσεων (plots) στο RStudio.

σωθούν σύνολα από εντολές (σενάρια). Όταν δεν υπάρχει αυτό το παράθυρο, μπορείτε να το ανοίξετε μέσω της διαδρομής `File → New → R script`. Η απλή πληκτρολόγηση μιας εντολής στο παράθυρο του επεξεργαστή δεν είναι αρκετή, πρέπει επίσης να πάει και στο παράθυρο εντολών πριν η R μπορέσει να εκτελέσει την εντολή αυτή. Εάν θέλετε να τρέξετε μία γραμμή από το παράθυρο σεναρίων (ή και ολόκληρο το σενάριο), μπορείτε να κάνετε κλικ στο `Run` ή να πατήσετε τα πλήκτρα `CTRL+ENTER`, ώστε να τη στείλετε στο παράθυρο εντολών.

- Πάνω δεξιά: **χώρος εργασίας / ιστορικό**. Στο παράθυρο του χώρου εργασίας μπορείτε να δείτε ποια δεδομένα και ποιες τιμές έχει η R στη μνήμη της. Μπορείτε να δείτε και να επεξεργαστείτε τις τιμές κάνοντας κλικ πάνω τους. Το παράθυρο του ιστορικού δείχνει το τι έχει πληκτρολογηθεί παλιότερα.
- Κάτω δεξιά: **αρχεία / γραφικές παραστάσεις / πακέτα / βοήθεια**. Από εδώ μπορείτε να ανοίξετε αρχεία, να δείτε γραφικές παραστάσεις (και προηγούμενες γραφικές παραστάσεις, επίσης), να εγκαταστήσετε και να φορτώσετε πακέτα ή να χρησιμοποιήσετε τη λειτουργία της βοήθειας.

Μπορείτε να αλλάξετε το μέγεθος των παραθύρων σέρνοντας τα γκριζα διαχωριστικά μεταξύ των παραθύρων.

## 2.4 Κατάλογος εργασίας

Ο *κατάλογος εργασίας* σας (working directory) είναι ο φάκελος του υπολογιστή σας μέσα στον οποίο εργάζεστε. Όταν ζητήσετε από την R να ανοίξει ένα συγκεκριμένο αρχείο, αυτή θα κοιτάξει πρώτα στον κατάλογο εργασίας γι' αυτό το αρχείο, και όταν πείτε στην R να αποθηκεύσει ένα αρχείο δεδομένων ή ένα γράφημα, αυτή θα το αποθηκεύσει στον κατάλογο εργασίας.

Πριν ξεκινήσετε να εργάζεστε, παρακαλούμε θέστε τον κατάλογο εργασίας σας εκεί που έχετε ή εκεί που πρέπει να αποθηκεύονται όλα τα δεδομένα σας και τα αρχεία σεναρίων.

Εισάγετε στο παράθυρο εντολών το εξής: `setwd("directoryname")`. Για παράδειγμα:

```
> setwd("M:/Hydrology/R/")
```

Σιγουρευτείτε ότι οι μπάρες είναι πλάγιες μπάρες (/) και ότι δεν έχετε ξεχάσει τις αποστρόφους (για την ανάγκη των αποστρόφων, δείτε στην ενότητα 10.1). Η R διακρίνει τα πεζά από τα κεφαλαία γράμματα, οπότε βεβαιωθείτε ότι γράφετε με κεφαλαία εκεί που απαιτείται.

Μέσα στο RStudio μπορείτε επίσης να πάτε στο `Tools → Set working directory`.

## 2.5 Βιβλιοθήκες

Η R μπορεί να κάνει πολλές στατιστικές αναλύσεις και αναλύσεις δεδομένων. Αυτές είναι οργανωμένες

στα λεγόμενα *πακέτα* ή *βιβλιοθήκες*. Με την τυπική εγκατάσταση, εγκαθίστανται και τα περισσότερα συνήθη πακέτα.

Για να δείτε μια λίστα με όλα τα εγκατεστημένα πακέτα, πηγαίνετε στο παράθυρο πακέτων ή πληκτρολογήστε `library()` στο παράθυρο κονσόλας. Εάν το τετραγωνάκι μπροστά από το όνομα του πακέτου είναι σημειωμένο, το πακέτο φορτώνεται (ενεργοποιείται) και τότε μπορεί να χρησιμοποιηθεί.

Υπάρχουν πολλά επιπλέον διαθέσιμα πακέτα στον ιστότοπο της R. Εάν θέλετε να εγκαταστήσετε και να χρησιμοποιήσετε ένα πακέτο (για παράδειγμα, το πακέτο που λέγεται “geometry”), τότε πρέπει να :

- Εγκαταστήσετε το πακέτο: κάντε κλικ στο `install packages` στο παράθυρο πακέτων και πληκτρολογήστε `geometry` ή εισάγετε την εντολή `install.packages("geometry")` στο παράθυρο εντολών.
- Φορτώστε το πακέτο: σημειώστε το κουτάκι μπροστά από το `geometry` ή εισάγετε την εντολή `library("geometry")` στο παράθυρο εντολών.

## 3 Μερικά πρώτα παραδείγματα εντολών της R

### 3.1 Αριθμομηχανή

Η R μπορεί να χρησιμοποιηθεί ως αριθμομηχανή. Μπορείτε απλά να εισάγετε την εξίσωση που θέλετε στο παράθυρο εντολών μετά από το “>”:

```
> 10^2 + 36
```

και η R θα σας δώσει την απάντηση

```
[1] 136
```

#### ToDo

Υπολογίστε τη διαφορά μεταξύ του 2014 και του έτους στο οποίο ξεκινήσατε να σπουδάζετε στο πανεπιστήμιο, και διαιρέστε το με τη διαφορά ανάμεσα στο 2014 και στο έτος το οποίο γεννηθήκατε. Πολλαπλασιάστε το επί 100 για να πάρετε ως αποτέλεσμα το ποσοστό της ζωής σας που έχετε περάσει σε αυτό το πανεπιστήμιο. Χρησιμοποιήστε παρενθέσεις, εάν χρειαστεί.

Εάν χρησιμοποιήσετε παρενθέσεις και ξεχάσετε να προσθέσετε μια στο τέλος, τότε το σύμβολο “>” στη γραμμή εντολών αλλάζει και γίνεται “+”. Το “+” μπορεί επίσης να σημαίνει ότι η R είναι ακόμα απασχολημένη με κάποιον βαρύ υπολογισμό. Εάν θέλετε η R να σταματήσει αυτό που κάνει και να επανέλθει στο σύμβολο “>”, τότε πιάστε ESC (δείτε τη λίστα με τις αναφορές στην τελευταία σελίδα).

## 3.2 Χώρος εργασίας

Μπορείτε επίσης να δώσετε σε αριθμούς ένα όνομα. Κάνοντάς το, αυτοί μετατρέπονται στις λεγόμενες μεταβλητές, οι οποίες μπορούν να χρησιμοποιηθούν αργότερα. Για παράδειγμα, μπορείτε να πληκτρολογήσετε στο παράθυρο εντολών το εξής:

```
> a = 4
```

Μπορείτε να δείτε ότι το `a` εμφανίζεται στο παράθυρο του χώρου εργασίας, κάτι το οποίο σημαίνει ότι η R πλέον θυμάται τι είναι το `a`.<sup>4</sup> Μπορείτε επίσης να ζητήσετε από την R να σας πει τι είναι το `a` (απλά πατήστε `a` ENTER στο παράθυρο εντολών):

```
> a
[1] 4
```

ή μπορείτε να κάνετε υπολογισμούς με το `a`:

```
> a * 5
[1] 20
```

Εάν προσδιορίσετε ξανά το `a`, η R θα ξεχάσει την τιμή που είχε αυτό πριν. Μπορείτε επίσης να αναθέσετε μια νέα τιμή στο `a` χρησιμοποιώντας την παλιά.

```
> a = a + 10
> a
[1] 14
```

Για να απομακρύνετε όλες τις μεταβλητές από τη μνήμη της R, πληκτρολογήστε

```
> rm(list=ls())
```

ή κάντε κλικ στο “clear all” στο παράθυρο του χώρου εργασίας. Μπορείτε να δείτε ότι τότε το RStudio αδειάζει το παράθυρο του χώρου εργασίας. Εάν θέλετε να απομακρύνετε μόνο τη μεταβλητή `a`, μπορείτε να πληκτρολογήσετε `rm(a)`.

#### ToDo

Επαναλάβετε το προηγούμενο ToDo, αλλά με διάφορα βήματα ενδιάμεσα. Μπορείτε να δώσετε στις μεταβλητές ό,τι όνομα επιθυμείτε, αλλά κάθε όνομα πρέπει να ξεκινάει με ένα γράμμα.

<sup>4</sup>Μερικοί προτιμούν τη χρήση του `<-` αντί του `=` (κάνουν το ίδιο πράγμα). Το `<-` αποτελείται από δύο χαρακτήρες, τον `<` και τον `-`, και συμβολίζει ένα βέλος που δείχνει στο αντικείμενο στο οποίο εκχωρείται η τιμή της έκφρασης.

### 3.3 Βαθμωτοί, διανύσματα και μητρώα

Όπως και πολλά άλλα προγράμματα, έτσι κι η R οργανώνει τους αριθμούς σε *βαθμωτούς* (απλοί αριθμοί – μηδενικής διάστασης), σε *διανύσματα* (μια ακολουθία αριθμών – μονοδιάστατα) και σε *μητρώα* (όπως ένας πίνακας – δισδιάστατα).

Το `a` που ορίσατε πριν ήταν ένας βαθμωτός. Για να ορίσετε ένα διάνυσμα με τους αριθμούς 3, 4 και 5, χρειάζεστε τη συνάρτηση<sup>5</sup> `c`, το όνομα της οποίας είναι το αρχικό του ρήματος “concatenate” (στα ελληνικά σημαίνει «συνενώνω»).

```
b=c(3,4,5)
```

Τα μητρώα και οι υπόλοιπες δισδιάστατες δομές θα παρουσιαστούν στην ενότητα 6.

### 3.4 Συναρτήσεις

Εάν θέλετε να υπολογίσετε το μέσο όρο όλων των στοιχείων του διανύσματος `b` του παραπάνω παραδείγματος, θα μπορούσατε να πληκτρολογήσετε

```
> (3+4+5)/3
```

Αλλά όταν το διάνυσμα είναι πολύ μεγάλο, αυτή η διαδικασία είναι πολύ βαρετή και χρονοβόρα. Γι' αυτό το λόγο πράγματα τα οποία κάνετε συχνά αυτοματοποιούνται στις λεγόμενες *συναρτήσεις*. Μερικές συναρτήσεις είναι εξ αρχής στην R ή βρίσκονται σε κάποιο από τα πακέτα. Μπορείτε επίσης να προγραμματίσετε τις δικές σας συναρτήσεις (ενότητα 11.3). Όταν θέλετε να χρησιμοποιήσετε μια συνάρτηση για να υπολογίσετε έναν μέσο όρο, τότε θα πληκτρολογήσετε:

```
> mean(x=b)
```

Μέσα στις παρενθέσεις προσδιορίζετε τα *ορίσματα*. Τα ορίσματα παρέχουν επιπλέον πληροφορίες στη συνάρτηση. Στη συγκεκριμένη περίπτωση, το όρισμα `x` δηλώνει από ποιο σύνολο αριθμών (δηλαδή από ποιο διάνυσμα) πρέπει να υπολογιστεί ο μέσος όρος (εδώ είναι από το διάνυσμα `b`). Μερικές φορές, το όνομα του ορίσματος δεν είναι απαραίτητο: η εντολή `mean(b)` δουλεύει εξίσου καλά.

#### ToDo

Υπολογίστε το άθροισμα των 4, 5, 8 και 11 αρχικά συνδυάζοντάς τα σε ένα διάνυσμα, και έπειτα χρησιμοποιώντας τη συνάρτηση `sum`.

Πάλι για παράδειγμα, η συνάρτηση `rnorm` είναι μια πρότυπη συνάρτηση της R, η οποία δημιουργεί τυχαία δείγματα από μία κανονική κατανομή. Πιέστε το πλήκτρο ENTER και θα δείτε 10 τυχαίους αριθμούς όπως οι ακόλουθοι:

<sup>5</sup>Δείτε στην επόμενη ενότητα για την επεξήγηση των συναρτήσεων.

```
1 > rnorm(10)
2 [1] -0.949 1.342 -0.474 0.403
3 [5] -0.091 -0.379 1.015 0.740
4 [9] -0.639 0.950
```

• Η γραμμή 1 περιέχει την εντολή: η `rnorm` είναι η συνάρτηση και το 10 είναι το όρισμα που ορίζει πόσους τυχαίους αριθμούς επιθυμείτε — σε αυτήν την περίπτωση, 10 αριθμούς (πληκτρολογώντας `n=10` αντί μόνο του 10 θα δούλευε επίσης).

• Οι γραμμές 2-4 περιέχουν τα αποτελέσματα: 10 τυχαίους αριθμούς τοποθετημένους σε ένα διάνυσμα μήκους 10.

Η εισαγωγή της ίδιας εντολής ξανά παράγει 10 νέους τυχαίους αριθμούς. Αντί να πληκτρολογήσετε το ίδιο κείμενο πάλι, μπορείτε επίσης να πιέσετε το πλήκτρο με το άνω βελάκι (↑) για να προσπελάσετε παλιότερες εντολές. Εάν θέλετε 10 τυχαίους αριθμούς από μία κανονική κατανομή με μέσο όρο 1.2 και τυπική απόκλιση 3.4, μπορείτε να πληκτρολογήσετε

```
> rnorm(10, mean=1.2, sd=3.4)
```

κάτι που δείχνει ότι η ίδια συνάρτηση (`rnorm`) μπορεί να έχει διαφορετικές διεπαφές, και ότι η R έχει τα λεγόμενα *επώνυμα ορίσματα* (σε αυτήν την περίπτωση τα `mean` και `sd`). Παρεμπιπτόντως, τα κενά γύρω από το “,” και το “=” δεν μετράνε.

Συγκρίνοντας αυτό το παράδειγμα με το προηγούμενο, βλέπει κανείς επίσης ότι για τη συνάρτηση `rnorm` μόνο το πρώτο όρισμα (ο αριθμός 10) είναι υποχρεωτικό, και ότι η R δίνει προκαθορισμένες τιμές στα υπόλοιπα προαιρετικά (όπως λέγονται) ορίσματα.<sup>6</sup>

Το RStudio έχει ένα ωραίο χαρακτηριστικό: εάν πληκτρολογήσετε `rnorm` (στο παράθυρο εντολών και πιέσετε TAB, τότε το RStudio θα εμφανίσει τα πιθανά ορίσματα (Σχήμα 2).

### 3.5 Γραφικές παραστάσεις

Η R μπορεί να φτιάξει γραφικές παραστάσεις. Το ακόλουθο είναι ένα πολύ απλό<sup>7</sup> παράδειγμα:

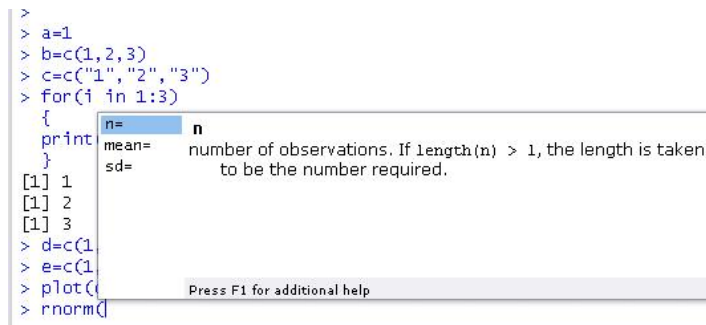
```
1 > x = rnorm(100)
2 > plot(x)
```

• Στην πρώτη γραμμή, εκχωρούνται 100 τυχαίοι αριθμοί στη μεταβλητή `x`, η οποία γίνεται διάνυσμα μέσω αυτής της διαδικασίας.

• Στη δεύτερη γραμμή, όλες αυτές οι τιμές σχεδιάζονται στο παράθυρο γραφικών παραστάσεων.

<sup>6</sup>Χρησιμοποιήστε τη συνάρτηση `help` (ενότητα 4) για να δείτε τις τιμές που χρησιμοποιούνται ως προκαθορισμένες.

<sup>7</sup>Δείτε την ενότητα 7 για λίγο λιγότερο «ασήμαντα» παραδείγματα.



**Σχήμα 2:** Το RStudio προτείνει πιθανά ορίσματα, εάν πατήσετε το TAB μετά από το όνομα της συνάρτησης και την παρένθεση.

### ToDo

Κάντε τη γραφική παράσταση 100 τυχαίων αριθμών (κανονικής κατανομής).

## 4 Βοήθεια και τεκμηρίωση

Υπάρχει διαθέσιμο πολύ υλικό (δωρεάν) τεκμηρίωσης και βοήθειας. Ένα μέρος της βοήθειας εγκαθίσταται αυτόματα. Η πληκτρολόγηση στο παράθυρο της κονσόλας της εντολής

```
> help(rnorm)
```

θα σας προσφέρει βοήθεια σχετικά με την συνάρτηση `rnorm`. Σας δίνει μια περιγραφή της συνάρτησης, πιθανά ορίσματα και τις τιμές που χρησιμοποιούνται ως προεπιλογή για τα προαιρετικά ορίσματα. Εάν πληκτρολογήσετε

```
> example(rnorm)
```

θα σας επιστρέψει μερικά παραδείγματα του πώς μπορεί να χρησιμοποιηθεί αυτή η συνάρτηση.

Καθολική βοήθεια βασισμένη σε HTML μπορεί να κληθεί μέσω της εντολής:

```
> help.start()
```

ή με μετάβαση στο παράθυρο βοήθειας.

Οι ακόλουθοι σύνδεσμοι μπορεί επίσης να φανούν πολύ χρήσιμοι:

- <http://cran.r-project.org/doc/manuals/R-intro.pdf> Ένα πλήρες εγχειρίδιο.
- <http://cran.r-project.org/doc/contrib/Short-refcard.pdf> Μια σύντομη καρτέλα αναφοράς.
- [http://zoonek2.free.fr/UNIX/48\\_R/all.html](http://zoonek2.free.fr/UNIX/48_R/all.html) Μια πολύ πλούσια πηγή παραδειγμάτων.
- <http://rwiki.sciviews.org/doku.php> Ένα τυπικό wiki για χρήστες.
- <http://www.statmethods.net/> Καλείται επίσης και Quick-R. Παρέχει πολύ παραγωγική και άμεση βοήθεια. Επίσης, κατάλληλη για

χρήστες που έρχονται από άλλες γλώσσες προγραμματισμού.

- <http://mathesaurus.sourceforge.net/> Λεξικό για άλλες γλώσσες προγραμματισμού (π.χ. R για χρήστες Matlab).

- Και μόνο η χρήση της μηχανής Google (πληκτρολογήστε π.χ. "R rnorm" στο πεδίο αναζήτησης) μπορεί να είναι πολύ παραγωγική.

### ToDo

Βρείτε βοήθεια για τη συνάρτηση `sqrt`.

## 5 Σενάρια

Η R είναι ένας διερμηνέας που χρησιμοποιεί ένα περιβάλλον βασισμένο στη γραμμή εντολών. Αυτό σημαίνει ότι θα χρειαστεί να πληκτρολογείτε εντολές, αντί να χρησιμοποιείτε απλά το ποντίκι και τα μενού. Αυτό έχει το πλεονέκτημα του ότι δε χρειάζεται εσείς να πληκτρολογείτε κάθε φορά ξανά όλες τις εντολές, και έτσι έχετε λιγότερες πιθανότητες να εμφανίσετε πόνους στα χέρια, το λαιμό και τους ώμους σας.

Μπορείτε να αποθηκεύετε τις εντολές σας σε αρχεία, τα λεγόμενα *σενάρια*. Αυτά τα σενάρια έχουν τυπικά ονόματα αρχείων με την κατάληξη `.R`, π.χ. `foo.R`. Μπορείτε να ανοίξετε τον επεξεργαστή κειμένου σε ένα παράθυρο και να επεξεργαστείτε αυτά τα αρχεία κάνοντας κλικ στο File και New ή στο Open file...<sup>8</sup>.

Μπορείτε να τρέξετε (να στείλετε δηλαδή στο παράθυρο κονσόλας) ένα μέρος του κώδικα, επιλέγοντας γραμμές του και πατώντας CTRL+ENTER ή κάνοντας κλικ στο Run στο παράθυρο του επεξεργαστή κειμένου. Εάν δεν επιλέξετε κάτι, η R θα τρέξει τη γραμμή στην οποία βρίσκεται ο κέρσορας. Μπορείτε πάντα να τρέξετε όλο το σενάριο με την εντολή κονσόλας `source`, κι έτσι π.χ. για το σενάριο στο αρχείο `foo.R` θα πρέπει να πληκτρολογήσετε:

```
> source("foo.R")
```

<sup>8</sup>Όπου είναι διαθέσιμες επίσης και οι επιλογές Save και Save as.

Μπορείτε επίσης να κάνετε κλικ στο `Run all` στο παράθυρο του επεξεργαστή κειμένου ή να πατήσετε τα πλήκτρα `CTRL+SHIFT+S` ώστε να τρέξετε ολόκληρο το σενάριο με τη μία.

#### ToDo

Δημιουργήστε ένα αρχείο με όνομα `firstscript.R`, το οποίο θα περιέχει κώδικα R που θα δημιουργεί 100 τυχαίους αριθμούς και θα κάνει τη γραφική τους παράσταση, και τρέξτε αυτό το σενάριο αρκετές φορές.

## 6 Δομές δεδομένων

Εάν δεν είστε αρκετά εξοικειωμένοι με την R, τότε είναι λογικό να επαναπληκτρολογείτε απλά τις εντολές που παρατίθενται σε αυτήν την ενότητα. Ίσως να μην χρειαστείτε όλες αυτές τις δομές στην αρχή, αλλά πάντα είναι καλό να έχετε τουλάχιστον μια πρώτη εικόνα της ορολογίας και των πιθανών εφαρμογών.

### 6.1 Διανύσματα

Τα *διανύσματα* τα έχουμε γνωρίσει ήδη, όμως μπορούν να κάνουν περισσότερα:

```
1 > vec1 = c(1,4,6,8,10)
2 > vec1
3 [1] 1 4 6 8 10
4 > vec1[5]
5 [1] 10
6 > vec1[3] = 12
7 > vec1
8 [1] 1 4 12 8 10
9 > vec2 = seq(from=0, to=1, by=0.25)
10 > vec2
11 [1] 0.00 0.25 0.50 0.75 1.00
12 > sum(vec1)
13 [1] 35
14 > vec1 + vec2
15 [1] 1.00 4.25 12.50 8.75 11.00
```

- Στη γραμμή 1, ένα διάνυσμα `vec1` δημιουργείται ρητά από τη συνάρτηση συνένωσης `c()`, την οποία έχουμε δει νωρίτερα. Τα στοιχεία των διανυσμάτων μπορούν να προσπελαστούν μέσω της πρότυπης ευρετηρίασης `[i]`, όπως φαίνεται στις γραμμές 4-5.
- Στη γραμμή 6, ένα από τα στοιχεία αντικαθίσταται με ένα νέο αριθμό. Το αποτέλεσμα εμφανίζεται στη γραμμή 8.
- Στη γραμμή 9 βλέπουμε ακόμα ένα χρήσιμο τρόπο δημιουργίας ενός διανύσματος: τη συνάρτηση `seq()` (sequence ή ελληνιστί ακολουθία).
- Στις γραμμές 10-15 γίνονται μερικοί τυπικοί υπολογισμοί που αφορούν διανύσματα. Εάν προσθέσετε δύο διανύσματα ίδιου μήκους, το πρώτο στοιχείο κάθε διανύσματος αθροίζεται με το άλλο, το ίδιο και το δεύτερο,

κ.ο.κ., έχοντας ως αποτέλεσμα ένα νέο διάνυσμα μήκους 5 (ακριβώς όπως και στους κανονικούς υπολογισμούς με διανύσματα). Προσέξτε ότι η συνάρτηση `sum` αθροίζει όλα τα στοιχεία ενός διανύσματος, έχοντας ως αποτέλεσμα έναν αριθμό (ένα βαθμωτό).

### 6.2 Μητρώα

Τα *μητρώα* δεν είναι τίποτε άλλο παρά δισδιάστατα διανύσματα. Για να ορίσετε ένα μητρώο, χρησιμοποιήστε τη συνάρτηση `matrix`:

```
1 mat=matrix(data=c(9,2,3,4,5,6),ncol=3)
2 > mat
3      [,1] [,2] [,3]
4 [1,]  9   3   5
5 [2,]  2   4   6
```

Το όρισμα `data` καθορίζει ποια νούμερα πρέπει να εισαχθούν στο μητρώο. Χρησιμοποιείτε είτε το `ncol` για να προσδιορίσετε τον αριθμό των στηλών, είτε το `nrow` για να προσδιορίσετε τον αριθμό των γραμμών.

#### ToDo

Εισάγετε τους αριθμούς από το 31 έως το 60 σε ένα διάνυσμα με όνομα `P` και σε ένα μητρώο με 6 γραμμές και 5 στήλες με όνομα `Q`. Υπόδειξη: χρησιμοποιείτε τη συνάρτηση `seq`. Δείτε τους διάφορους τρόπους με τους οποίους συμβολίζονται οι βαθμωτοί, τα διανύσματα και τα μητρώα στο παράθυρο του χώρου εργασίας.

Οι πράξεις με μητρώα είναι παρόμοιες με τις πράξεις σε διανύσματα:

```
1 > mat[1,2]
2 [1] 3
3 > mat[2,]
4 [1] 2 4 6
5 > mean(mat)
6 [1] 4.8333
```

- Τα στοιχεία ενός μητρώου μπορούν να προσπελαστούν με το συνήθη τρόπο: `[row,column]` (γραμμή 1).
- Γραμμή 3: όταν θελήσετε να επιλέξετε μια ολόκληρη γραμμή, αφήστε τη θέση για το όρισμα του αριθμού των στηλών κενή (και φυσικά το ανάποδο όταν θελήσετε στήλες).
- Στη γραμμή 5 βλέπουμε ότι πολλές συναρτήσεις εξακολουθούν να δουλεύουν όταν έχουν μητρώα ως όρισμα.

## 6.3 Πλαίσια δεδομένων

Οι χρονοσειρές (time series) συχνά κατατάσσονται στα *πλαίσια δεδομένων* (data frames). Ένα πλαίσιο δεδομένων είναι ένα μητρώο με ονόματα πάνω από τις στήλες του. Αυτό είναι καλό, γιατί έτσι μπορείτε να καλέσετε και να χρησιμοποιήσετε όποια από τις στήλες θέλετε χωρίς να γνωρίζετε σε ποια θέση είναι αυτή.

```
1 > t = data.frame(x = c(11,12,14),
2   y = c(19,20,21), z = c(10,9,7))
3 > t
4   x y z
5 1 11 19 10
6 2 12 20 9
7 3 14 21 7
8 > mean(t$z)
9 [1] 8.666667
10 > mean(t[["z"]])
11 [1] 8.666667
```

- Στις γραμμές 1-2 κατασκευάζεται ένα τυπικό πλαίσιο δεδομένων με όνομα `t`. Οι στήλες έχουν τα ονόματα `x`, `y` and `z`.
- Στις γραμμές 8-11 βλέπουμε δύο τρόπους με τους οποίους μπορείτε να επιλέξετε τη στήλη με όνομα `z` από το πλαίσιο δεδομένων με όνομα `t`.

### ToDo

Δημιουργήστε ένα σενάριο το οποίο θα κατασκευάζει τρία τυχαία (κανονικά) διανύσματα μήκους 100. Ονομάστε αυτά τα διανύσματα `x1`, `x2` και `x3`. Φτιάξτε ένα πλαίσιο δεδομένων με όνομα `t` με τρεις στήλες (που θα λέγονται `a`, `b` και `c`) που θα περιέχει αντίστοιχα τα `x1`, `x1+x2` και `x1+x2+x3`. Καλέστε τις ακόλουθες συναρτήσεις για αυτό το πλαίσιο δεδομένων: `plot(t)` και `sd(t$x1)`. Μπορείτε να κατανοήσετε τα αποτελέσματα; Τρέξτε το σενάριο ξανά μερικές φορές.

## 6.4 Λίστες

Μια άλλη βασική δομή στην R είναι η *λίστα*. Το κύριο πλεονέκτημα των λιστών είναι ότι οι «στήλες» (δεν είναι πλέον διατεταγμένες σε στήλες, αλλά μοιάζουν πιο πολύ με μια συλλογή διανυσμάτων) δεν είναι υποχρεωτικό να έχουν το ίδιο μήκος, αντίθετα με τις περιπτώσεις των μητρώων και των πλαισίων δεδομένων.

```
1 > L = list(one=1, two=c(1,2),
2   five=seq(0, 1, length=5))
3 > L
4 $one
5 [1] 1
6 $two
7 [1] 1 2
```

```
8 $five
9 [1] 0.00 0.25 0.50 0.75 1.00
10 > names(L)
11 [1] "one" "two" "five"
12 > L$five + 10
13 [1] 10.00 10.25 10.50 10.75 11.00
```

- Στις γραμμές 1-2 δημιουργείται μια λίστα μέσω της εισόδου ονομάτων και τιμών. Η λίστα εμφανίζεται επίσης και στο παράθυρο του χώρου εργασίας.
- Στις γραμμές 3-9 φαίνεται μια τυπική εκτύπωση (μετά από πάτημα των `L` και `ENTER`).
- Η γραμμή 10 δείχνει πώς μπορούμε να δούμε τι υπάρχει μέσα στη λίστα.
- Η γραμμή 12 παρουσιάζει έναν τρόπο χρήσης των αριθμών.

## 7 Γραφικά

Η σχεδίαση γραφικών παραστάσεων είναι μια σημαντική στατιστική δραστηριότητα. Οπότε δεν πρέπει να σας εκπλήσσει το γεγονός ότι η R έχει πολλές δυνατότητες σχεδιασμού γραφικών παραστάσεων. Οι ακόλουθες γραμμές εμφανίζουν ένα απλό γράφημα:

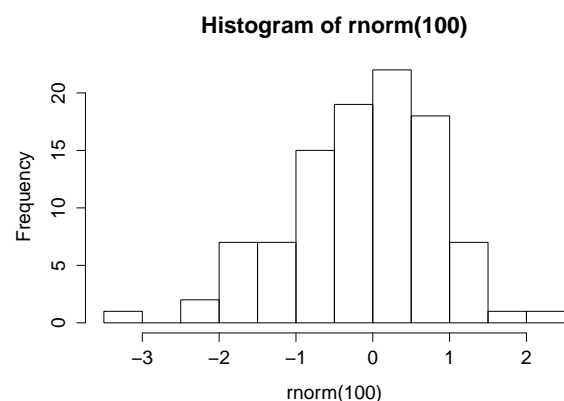
```
> plot(rnorm(100), type="l", col="gold")
```

Εκατοντάδες τυχαίοι αριθμοί αναπαρίστανται γραφικά μέσω της σύνδεσης των σημείων με γραμμές (το σύμβολο μέσα σε εισαγωγικά μετά το `type=` είναι το γράμμα `l`, όχι ο αριθμός 1) σε χρυσό χρώμα.

Ένα άλλο πολύ απλό παράδειγμα είναι το κλασικό στατιστικό γράφημα του ιστογράμματος, που δημιουργείται από την απλή εντολή

```
> hist(rnorm(100))
```

η οποία παράγει το γράφημα στο Σχήμα 3.



**Σχήμα 3:** Ένα απλό ιστόγραμμα.

Οι ακόλουθες γραμμές δημιουργούν ένα γράφημα χρησιμοποιώντας το πλαίσιο δεδομένων `t` που φτιάξαμε στο προηγούμενο `ToDo`:

```
1 plot(t$a, type="l", ylim=range(t),  
2     lwd=3, col=rgb(1,0,0,0.3))  
3 lines(t$b, type="s", lwd=2,  
4     col=rgb(0.3,0.4,0.3,0.9))  
5 points(t$c, pch=20, cex=4,  
6     col=rgb(0,0,1,0.3))
```

### ToDo

Προσθέστε αυτές τις γραμμές στο αρχείο σεναρίου της προηγούμενης ενότητας. Δοκιμάστε να ανακαλύψετε, είτε μέσω πειραματισμών, είτε με τη χρήση της βοήθειας, ποιο είναι το νόημα της `rgb`, του τελευταίου ορίσματος της `rgb`, του `lwd`, του `pch`, και του `cex`.

Για να μάθετε περισσότερα σχετικά με τη μορφοποίηση των γραφημάτων, ψάξτε το `par` στη βοήθεια της R. Κάντε αναζήτηση στη μηχανή της Google τον όρο “R color chart” για να βρείτε ένα αρχείο pdf που περιέχει πληθώρα επιλογών χρωμάτων.

Για να αντιγράψετε τη γραφική σας παράσταση σε ένα έγγραφο, πηγαίνετε στο παράθυρο των γραφικών παραστάσεων, κάντε κλικ στο κουμπί “Export”, επιλέξτε το μήκος και πλάτος που σας αρέσει περισσότερο και κάντε κλικ στο `Copy` ή στο `Save`.

## 8 Διαβάζοντας από και γράφοντας σε αρχεία δεδομένων

Υπάρχουν πολλοί τρόποι για να καταγράψει κανείς δεδομένα σε αρχεία μέσω του περιβάλλοντος της R, και για να διαβάσει δεδομένα από αρχεία. Εδώ θα παρουσιάσουμε έναν τέτοιο τρόπο. Οι ακόλουθες γραμμές παρουσιάζουν τα στοιχειώδη:

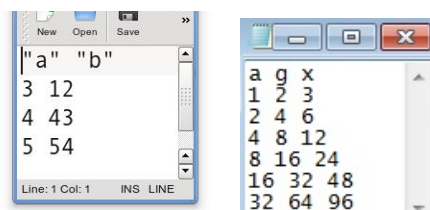
```
1 > d = data.frame(a = c(3,4,5),  
2   b = c(12,43,54))  
3 > d  
4   a  b  
5 1 3 12  
6 2 4 43  
7 3 5 54  
8 > write.table(d, file="tst0.txt",  
9   row.names=FALSE)  
10 > d2 = read.table(file="tst0.txt",  
11   header=TRUE)  
12 > d2  
13   a  b  
14 1 3 12  
15 2 4 43  
16 3 5 54
```

- Στις γραμμές 1-2, δημιουργείται ένα απλό πλαίσιο δεδομένων ως παράδειγμα και αποθηκεύεται στη μεταβλητή `d`.

- Στις γραμμές 3-7 φαίνεται το περιεχόμενο αυτού του πλαισίου δεδομένων: δύο στήλες (με όνομα `a` και `b`), κάθε μία από τις οποίες περιέχει τρεις αριθμούς.

- Στη γραμμή 8 καταγράφεται αυτό το πλαίσιο δεδομένων σε ένα αρχείο κειμένου, με όνομα `tst0.txt`. Το όρισμα `row.names=FALSE` αποτρέπει τα ονόματα των γραμμών να καταγραφούν στο αρχείο. Επειδή δεν καθορίζεται κάτι για τα `col.names` (ονόματα των στηλών), επιλέγεται η προκαθορισμένη επιλογή `col.names=TRUE` και τα ονόματα των στηλών καταγράφονται στο αρχείο. Στο Σχήμα 4 φαίνεται το τελικό αρχείο (ανοιγμένο σε έναν επεξεργαστή κειμένου, όπως το Σημειωματάριο), με τα ονόματα των στηλών (`a` και `b`) στην πρώτη γραμμή.

- Οι γραμμές 10-11 δείχνουν πώς μπορούμε να εισάγουμε ένα αρχείο μέσα σε ένα πλαίσιο δεδομένων. Σημειώστε ότι εισάγονται και τα ονόματα των στηλών. Το πλαίσιο δεδομένων εμφανίζεται επίσης στο παράθυρο του χώρου εργασίας.



**Σχήμα 4:** Τα αρχεία `tst0.txt` της ενότητας 8 (αριστερά) και `tst1.txt` από το `ToDo` παρακάτω (δεξιά), ανοιγμένα σε δύο επεξεργαστές κειμένου.

### ToDo

Φτιάξτε ένα αρχείο με όνομα `tst1.txt` στο Σημειωματάριο από το παράδειγμα του Σχήματος 4 και αποθηκεύστε το στον κατάλογο εργασίας σας. Γράψτε ένα σενάριο που να το διαβάζει, που να πολλαπλασιάζει τη στήλη με όνομα `g` επί 5 και να την αποθηκεύει με όνομα `tst2.txt`.

## 9 Μη-διαθέσιμα δεδομένα

### ToDo

Υπολογίστε το μέσο όρο της τετραγωνικής ρίζας ενός διανύσματος 100 τυχαίων αριθμών. Τι θα συμβεί;

Όταν δουλεύετε με πραγματικά δεδομένα, θα βρεθείτε αντιμέτωποι με τιμές που λείπουν επειδή υπήρξαν αστοχίες στα όργανα μέτρησης ή επειδή δεν θέλατε να κάνετε μετρήσεις το Σαββατοκύριακο. Όταν



ένα δεδομένο δεν είναι διαθέσιμο, τότε πρέπει να γράψετε NA (τα αρχικά του “Not Available”) αντί κάποιου αριθμού.

```
> j = c(1,2,NA)
```

Ο υπολογισμός στατιστικών από ημιτελή σύνολα δεδομένων είναι αδύνατος, αυστηρά μιλώντας. Μπορεί η μέγιστη τιμή να εμφανίστηκε κατά τη διάρκεια του Σαββατοκύριακου, όταν δεν κάνατε μετρήσεις. Κατά συνέπεια, η R θα σας πει ότι δε γνωρίζει ποια είναι η μέγιστη τιμή του j:

```
> max(j)
[1] NA
```

Εάν δεν έχετε πρόβλημα με τα δεδομένα που λείπουν και θέλετε να υπολογίσετε τα στατιστικά όπως και να 'χει, μπορείτε να προσθέσετε το όρισμα `na.rm=TRUE` (να απομακρύνω/remove/rm τις τιμές NA; Ναι!).

```
> max(j, na.rm=TRUE)
[1] 2
```

## 10 Κλάσεις

Οι ασκήσεις που κάνατε πριν ήταν σχεδόν όλες με αριθμούς. Μερικές φορές θα χρειαστεί να προσδιορίσετε κάτι το οποίο δεν είναι αριθμός, για παράδειγμα το όνομα ενός σταθμού μετρήσεων ή ενός αρχείου δεδομένων. Σε αυτήν την περίπτωση θέλετε η μεταβλητή να είναι μια ακολουθία χαρακτήρων αντί να είναι αριθμός.

Ένα αντικείμενο στην R μπορεί να έχει διάφορες από τις λεγόμενες *κλάσεις*. Οι τρεις πιο σημαντικές είναι η *numeric*, η *character* και η *POSIX* (συνδυασμοί ημερομηνίας-ώρας). Μπορείτε να ρωτήσετε την R ποια είναι η κλάση μιας συγκεκριμένης μεταβλητής πληκτρολογώντας `class(...)`.

### 10.1 Χαρακτήρες

Για να δηλώσετε στην R ότι κάτι είναι ακολουθία χαρακτήρων, πρέπει να πληκτρολογήσετε το κείμενο ανάμεσα σε αποστρόφους, αλλιώς η R θα αρχίσει να ψάχνει για μια καθορισμένη μεταβλητή με το ίδιο όνομα:

```
> m = "apples"
> m
[1] "apples"
> n = pears
Error: object 'pears' not found
```

Φυσικά, δεν μπορείτε να κάνετε υπολογισμούς με ακολουθίες χαρακτήρων:

```
> m + 2
Error in m + 2 : non-numeric argument to
binary operator
```

## 10.2 Ημερομηνίες

Οι ημερομηνίες και οι ώρες είναι πολύπλοκες περιπτώσεις. Η R πρέπει να γνωρίζει ότι η ώρα 3 ακριβώς είναι μετά από τις 2:59 και ότι ο Φεβρουάριος έχει 29 ημέρες σε μερικά έτη. Ο ευκολότερος τρόπος για να πείτε στην R ότι κάτι αποτελεί συνδυασμό ημερομηνίας-ώρας είναι μέσω της συνάρτησης `strptime`:

```
1 > date1=strptime( c("20100225230000",
2   "20100226000000", "20100226010000"),
3   format="%Y%m%d%H%M%S")
4 > date1
5 [1] "2010-02-25 23:00:00"
6 [2] "2010-02-26 00:00:00"
7 [3] "2010-02-26 01:00:00"
```

- Στις γραμμές 1-2 δημιουργείται ένα διάνυσμα με την `c(...)`. Οι αριθμοί στα διανύσματα είναι ανάμεσα σε αποστρόφους, επειδή η συνάρτηση `strptime` απαιτεί ακολουθίες χαρακτήρων ως είσοδο.

- Στη γραμμή 3 το όρισμα `format` προσδιορίζει πώς θα πρέπει να διαβαστεί η ακολουθία χαρακτήρων. Σε αυτή την περίπτωση το έτος αναπαρίσταται στην αρχή (%Y), έπειτα ο μήνας (%m), η ημέρα (%d), η ώρα (%H), τα λεπτά (%M) και τα δευτερόλεπτα (%S). Δεν χρειάζεται να τα προσδιορίσετε όλα αυτά, εφόσον η μορφή αυτή είναι αντίστοιχη με την ακολουθία εισόδου.

### ToDo

Δημιουργήστε ένα γράφημα, το οποίο θα έχει στον άξονα των x ημερομηνίες από τις 6 Δεκεμβρίου 2014 έως τα επόμενα γενέθλιά σας, και στον άξονα των y τον αριθμό των δώρων που περιμένετε σε κάθε μία από αυτές τις μέρες. Υπόδειξη: φτιάξτε δύο διανύσματα πρώτα.

## 11 Προγραμματιστικά εργαλεία

Όταν φτιάχνετε ένα μεγαλύτερο πρόγραμμα από τα προηγούμενα παραδείγματα ή όταν χρησιμοποιείτε τα σενάρια κάποιου άλλου, μπορεί να συναντήσετε κάποιες προγραμματιστικές δομές. Στην ενότητα αυτή θα περιγράψουμε μερικές σχετικές συμβουλές και κόλπα.

### 11.1 Δομή ελέγχου if

Η δομή ελέγχου *if* χρησιμοποιείται όταν συγκεκριμένοι υπολογισμοί πρέπει να γίνουν μόνο όταν ικανοποιείται μια συγκεκριμένη συνθήκη (και πιθανόν να

πρέπει να γίνει κάτι άλλο, όταν η συνθήκη δεν ικανοποιείται). Ένα παράδειγμα:

```
1 > w = 3
2 > if( w < 5 )
3   {
4     d=2
5   }else{
6     d=10
7   }
8 > d
9 2
```

- Στη γραμμή 2 καθορίζεται μια συνθήκη: το `w` πρέπει να είναι μικρότερο του 5.
- Εάν η συνθήκη ικανοποιηθεί, τότε η R θα εκτελέσει ό,τι βρίσκεται ανάμεσα στις πρώτες αγκύλες στη γραμμή 4.
- Εάν η συνθήκη δεν ικανοποιηθεί, τότε η R θα εκτελέσει ό,τι βρίσκεται ανάμεσα στις δεύτερες αγκύλες, μετά το `else` στην γραμμή 6. Μπορείτε να παραλείψετε το κομμάτι του `else{...}` εάν δεν το χρειάζεστε.
- Σε αυτή την περίπτωση, η συνθήκη ικανοποιείται και στη μεταβλητή `d` εκχωρείται η τιμή 2 (γραμμές 8-9).

Για να πάρετε ένα υποσύνολο στοιχείων ενός διανύσματος για τα οποία ισχύει μια συγκεκριμένη συνθήκη, μπορείτε να χρησιμοποιήσετε μια πιο σύντομη μέθοδο:

```
1 > a = c(1,2,3,4)
2 > b = c(5,6,7,8)
3 > f = a[b==5 | b==8]
4 > f
5 [1] 1 4
```

- Στη γραμμή 1 και 2 δημιουργούνται δύο διανύσματα.
- Στη γραμμή 3 δηλώνετε ότι η `f` συντίθεται από εκείνα τα στοιχεία του διανύσματος `a` για τα οποία τα αντίστοιχα στοιχεία του `b` ισούται με 5 ή με 8.

Προσέξτε το διπλό `=` στη συνθήκη. Άλλες συνθήκες (που ονομάζονται επίσης λογικοί ή Boolean τελεστές) είναι οι `<`, `>`, `!=` ( $\neq$ ), `<=` ( $\leq$ ) και `>=` ( $\geq$ ). Για να ελέγξετε περισσότερες από μία συνθήκες σε μία δομή `if`, χρησιμοποιήστε το `&` εάν και οι δύο συνθήκες πρέπει να ικανοποιούνται (λογικό «και» - “and”) και το `|` εάν μία από τις συνθήκες πρέπει να ικανοποιείται (λογικό «ή» - “or”).

## 11.2 Βρόχος επανάληψης for

Όταν θέλετε να μοντελοποιήσετε μια χρονοσειρά, συνήθως κάνετε τους υπολογισμούς για ένα χρονικό βήμα και έπειτα για το επόμενο και για το μεθεπόμενο, κ.ο.κ. Επειδή κανένας δε θέλει να πληκτρολογεί τις ίδιες εντολές ξανά και ξανά, αυτοί οι υπολογισμοί μπορούν να αυτοματοποιηθούν με βρόχους επανάληψης `for`.

Σε έναν βρόχο `for` προσδιορίζετε τι πρέπει να γίνει και πόσες φορές πρέπει να γίνει. Για να δηλώσετε το «πόσες φορές», καθορίζετε το λεγόμενο μετρητή. Ένα παράδειγμα:

```
1 > h = seq(from=1, to=8)
2 > s = c()
3 > for(i in 2:10)
4   {
5     s[i] = h[i] * 10
6   }
7 > s
8 [1] NA 20 30 40 50 60 70 80 NA NA
```

- Αρχικά δημιουργείται το διάνυσμα `h`.
- Στη γραμμή 2 δημιουργείται ένα κενό διάνυσμα (`s`). Αυτό είναι απαραίτητο γιατί όταν δημιουργείτε μια μεταβλητή μέσα σε βρόχο `for`, η R δε θα είναι σε θέση να τη θυμάται όταν βγει έξω από αυτόν.
- Στη γραμμή 3 ξεκινάει ο βρόχος `for`. Σε αυτή την περίπτωση, το `i` είναι ο μετρητής και τρέχει από το 2 έως το 10.
- Οποδήποτε βρίσκεται ανάμεσα στις αγκύλες (γραμμή 5) θα υποστεί επεξεργασία 9 φορές. Την πρώτη φορά για `i=2`, το δεύτερο στοιχείο του `h` πολλαπλασιάζεται με 10 και τοποθετείται στη δεύτερη θέση του διανύσματος `s`. Την δεύτερη φορά για `i=3`, κτλ. Στις τελευταίες δύο εκτελέσεις, ζητούνται το 9<sup>ο</sup> και 10<sup>ο</sup> στοιχείο του `h`, τα οποία δεν υπάρχουν. Σημειώστε ότι αυτές οι δηλώσεις ελέγχονται και υπολογίζονται χωρίς να υπάρχει κάποιο ρητό μήνυμα λάθους.

### ToDo

Δημιουργήστε ένα διάνυσμα από το 1 έως το 100. Φτιάξτε ένα βρόχο `for` ο οποίος θα διατρέχει ολόκληρο το διάνυσμα. Πολλαπλασιάστε τα στοιχεία τα οποία είναι μικρότερα του 5 και μεγαλύτερα του 90 επί 10, και τα υπόλοιπα στοιχεία επί 0.1.

## 11.3 Γράφοντας τις δικές σας συναρτήσεις

Οι συναρτήσεις που προγραμματίζετε εσείς οι ίδιοι λειτουργούν με τον ίδιο τρόπο που λειτουργούν οι συναρτήσεις που είναι προ-προγραμματισμένες μέσα στην R.

```
1 > fun1 = function(arg1, arg2 )
2   {
3     w = arg1 ^ 2
4     return(arg2 + w)
5   }
6 > fun1(arg1 = 3, arg2 = 5)
7 [1] 14
8
```

- Στη γραμμή 1 καθορίζεται το όνομα της συνάρτησης (`fun1`) και τα ορίσματά της (`arg1` και `arg2`).
- Στις γραμμές 2-5 καθορίζεται το τι θα πρέπει να κάνει η συνάρτηση όταν καλεστεί. Η τιμή που θα επιστρέφει (`arg2+w`) εμφανίζεται στην οθόνη.
- Στη γραμμή 6 καλείται η συνάρτηση με ορίσματα 3 και 5.

### ToDo

Γράψτε μια συνάρτηση για το προηγούμενο `ToDo`, ώστε να μπορείτε να την τροφοδοτείτε με οποιοδήποτε διάνυσμα επιθυμείτε εσείς (ως όρισμα). Χρησιμοποιήστε ένα βρόχο `for` μέσα στη συνάρτηση ώστε να κάνετε τον υπολογισμό κάθε στοιχείου. Χρησιμοποιείτε την πρότυπη συνάρτηση της R `length` κατά τον προσδιορισμό του μετρητή.<sup>α)</sup>

<sup>α)</sup> Στην πραγματικότητα, οι χρήστες χρησιμοποιούν πιο συχνά βρόχους `for` απ' ό,τι είναι ουσιαστικά απαραίτητο. Το παρ'απάνω `ToDo` μπορεί να γίνει πιο εύκολα και γρήγορα χωρίς τη χρήση βρόχου `for`, μα με τυπικούς υπολογισμούς διανυσμάτων.

- `x[["n"]]`: ό.π.
- `[i,j]`: το στοιχείο στην *i*-στη γραμμή και *j*-στη στήλη
- `[i,]`: η γραμμή *i* σε ένα μητρώο

### Πληροφορίες για μεταβλητές

- `length`: το μήκος ενός διανύσματος
- `ncol` ή `nrow`: ο αριθμός των στηλών ή των γραμμών ενός μητρώου
- `class`: η κλάση μιας μεταβλητής
- `names`: τα ονόματα των αντικειμένων σε μια λίστα
- `print`: εμφάνισε τη μεταβλητή ή την ακολουθία χαρακτήρων στην οθόνη (χρησιμοποιείται σε σενάρια ή σε βρόχους `for`)
- `return`: εμφάνισε τη μεταβλητή στην οθόνη (χρησιμοποιείται σε συναρτήσεις)
- `is.na`: έλεγξε εάν η μεταβλητή ισούται με `NA`
- `as.numeric` ή `as.character`: άλλαξε την κλάση σε αριθμό ή σε ακολουθία γραμμάτων
- `strptime`: άλλαξε την κλάση από χαρακτήρα σε ημερομηνία-ώρα (`POSIX`)

### Στατιστικά

- `sum`: το άθροισμα ενός διανύσματος (ή ενός μητρώου)
- `mean`: ο μέσος όρος ενός διανύσματος
- `sd`: η τυπική απόκλιση ενός διανύσματος
- `max` ή `min`: μέγιστο ή ελάχιστο στοιχείο
- `rowSums` (ή `rowMeans`, `colSums` και `colMeans`): αθροίσματα (ή μέσοι όροι) όλων των αριθμών σε κάθε γραμμή (ή στήλη) ενός μητρώου. Το αποτέλεσμα είναι ένα διάνυσμα.
- `quantile(x,c(0.1,0.5))`: δειγματοληψία του 0.1-στου και του 0.5-στου ποσοστημόριου του διανύσματος `x`

### Επεξεργασία δεδομένων

- `seq`: δημιούργησε ένα διάνυσμα με ίσες αποστάσεις μεταξύ των αριθμών
- `rnorm`: δημιούργησε ένα διάνυσμα με τυχαίους αριθμούς που ακολουθούν την κανονική κατανομή (και άλλες κατανομές είναι επίσης διαθέσιμες)
- `sort`: ταξινόμησε τα στοιχεία σε αύξουσα διάταξη
- `t`: αντιστρέψε ένα μητρώο
- `aggregate(x,by=ls(y),FUN="mean")`: διαχώρισε το σύνολο δεδομένων `x` σε υποσύνολα (που καθορίζονται από το `y`) και υπολόγισε τους μέσους όρους των υποσυνόλων. Αποτέλεσμα: μια νέα λίστα.
- `na.approx`: παρεμβολή (στο πακέτο `zoo`). Όρισμα: διάνυσμα με στοιχεία `NA`. Αποτέλεσμα: διάνυσμα χωρίς `NA`.
- `cumsum`: σωρευτικό άθροισμα. Το αποτέλεσμα είναι ένα διάνυσμα.
- `rollmean`: μετακίνηση μέσου όρου (στο πακέτο `zoo`)
- `paste`: συνένωση ακολουθιών χαρακτήρων μεταξύ τους
- `substr`: εξαγωγή μέρους μιας ακολουθίας χαρακτή-

## 12 Μερικές χρήσιμες αναφορές

### 12.1 Συναρτήσεις

Το παρακάτω είναι ένα υποσύνολο από συναρτήσεις που επεξηγούνται στην καρτέλα αναφορών (`reference card`) της R.

#### Δημιουργία δεδομένων (data creation)

- `read.table`: διάβασε έναν πίνακα από ένα αρχείο. Ορίσματα: `header=TRUE`: διάβασε την πρώτη γραμμή ως τίτλους των στηλών `sep=","`: οι αριθμοί διαχωρίζονται με κόμμα `skip=n`: μη διαβάσεις τις πρώτες *n* γραμμές.
- `write.table`: αποθήκευσε έναν πίνακα σε ένα αρχείο
- `c`: συνένωσε αριθμούς μεταξύ τους ώστε να δημιουργήσεις ένα διάνυσμα
- `array`: δημιούργησε ένα διάνυσμα, Ορίσματα: `dim`: μήκος
- `matrix`: δημιούργησε ένα μητρώο, Ορίσματα: `ncol` και/ή `nrow`: αριθμός γραμμών/ στηλών
- `data.frame`: δημιούργησε ένα πλαίσιο δεδομένων
- `list`: δημιούργησε μια λίστα
- `rbind` και `cbind`: συνένωσε διανύσματα σε ένα μητρώο κατά γραμμή ή κατά στήλη

#### Εξαγωγή δεδομένων (extracting data)

- `x[n]`: το *n*-στο στοιχείο ενός διανύσματος
- `x[m:n]`: από το *m*-στο έως το *n*-στο στοιχείο
- `x[c(k,m,n)]`: συγκεκριμένα στοιχεία
- `x[x>m & x<n]`: τα στοιχεία ανάμεσα στο *m* και *n*
- `x$n`: το στοιχείο της λίστας ή του πλαισίου δεδομένων με όνομα *n*

ρων

### Παρεμβολή (fitting)

- `lm(v1~v2)`: γραμμική παρεμβολή (γραμμή παλινδρόμησης) μεταξύ του διανύσματος `v1` στον άξονα των `y` και του `v2` στον άξονα των `x`
- `nls(v1~a+b*v2, start=ls(a=1,b=0))`: μη-γραμμική παρεμβολή. Πρέπει να περιλαμβάνει μια εξίσωση με μεταβλητές (εδώ είναι τα `v1` και `v2`) και παραμέτρους (εδώ είναι τα `a` και `b`) με αρχικές τιμές
- `coef`: επιστρέφει τους συντελεστές μιας παρεμβολής
- `summary`: επιστρέφει όλα τα αποτελέσματα από μια παρεμβολή

### Σχεδίαση γραφικών παραστάσεων

- `plot(x)`: σχεδίαση του `x` (άξονας `y`) προς τον αριθμό ευρετηρίου (άξονας `x`) σε ένα νέο παράθυρο
- `plot(x,y)`: σχεδίαση του `y` (άξονας `y`) έναντι του `x` (άξονας `x`) σε ένα νέο παράθυρο
- `image(x,y,z)`: σχεδίαση του `z` (χρωματική κλίμακα) έναντι του `x` (άξονας `x`) και του `y` (άξονας `y`) σε ένα νέο παράθυρο
- `lines` ή `points`: πρόσθεσε γραμμές ή σημεία σε μια προηγούμενη γραφική παράσταση
- `hist`: σχεδίαση του ιστογράμματος των αριθμών ενός διανύσματος
- `barplot`: γράφημα με μπάρες ενός διανύσματος ή ενός πλαισίου δεδομένων
- `contour(x,y,z)`: σχεδίαση γραφήματος με περιγράμματα
- `abline`: σχεδίαση γραμμής (τμήματος). Ορίσματα: `a`, `b` με σημείο τομής `a` και κλίση `b`; ή `h=y` για οριζόντια γραμμή στο `y`; ή `v=x` για κάθετη γραμμή στο `x`.
- `curve`: εισάγετε συνάρτηση για σχεδίαση. Χρειάζεται ένα `x` στην έκφραση. Παράδειγμα: `curve(x^2)`
- `legend`: προσθήκη υπομνήματος με δεδομένα σύμβολα (`lty` ή `pch` και `col`) και κείμενο (`legend`) στο σημείο (`x="topright"`)
- `axis`: προσθήκη άξονα. Ορίσματα: `side - 1=κάτω`, `2=αριστερά`, `3=πάνω`, `4=δεξιά`
- `mtext`: προσθήκη κειμένου στον άξονα. Ορίσματα: `text` (ακολουθία χαρακτήρων) και `side`
- `grid`: προσθήκη πλέγματος
- `par`: παράμετροι σχεδιασμού που πρέπει να προσδιοριστούν πριν τις γραφικές παραστάσεις. Ορίσματα: π.χ. `mfrow=c(1,3)`: αριθμός των γραφημάτων ανά σελίδα (1 σειρά, 3 στήλες); `new=TRUE`: σχεδίαση γραφικής παράστασης πάνω από ήδη υπάρχουσα.

### Παράμετροι γραφικών παραστάσεων

- Αυτές μπορούν να προστεθούν ως ορίσματα στις `plot`, `lines`, `image`, κτλ. Για βοήθεια δείτε την `par`.
- `type`: "l"=γραμμές (`lines`), "p"=σημεία (`points`), κτλ.
  - `col`: χρώμα – "blue", "red", κτλ.
  - `lty`: τύπος γραμμής – 1=ενιαία, 2=διακεκομμένη, κτλ.
  - `pch`: τύπος σημείου – 1=κύκλος, 2=τρίγωνο, κτλ.

- `main`: τίτλος - ακολουθία χαρακτήρων
- `xlab` και `ylab`: ετικέτες αξόνων – ακολουθίες χαρακτήρων
- `xlim` και `ylim`: εύρος των αξόνων – π.χ. `c(1,10)`
- `log`: λογαριθμικός άξονας – "x", "y" ή "xy"

### Προγραμματισμός

- `function(arglist){expr}`: ορισμός συνάρτησης: εκτέλεσε την έκφραση `expr` με αυτή τη λίστα ορισμάτων `arglist`
- `if(cond){expr1}else{expr2}`: δομή ελέγχου `if`: εάν η συνθήκη `cond` αληθεύει, τότε `expr1`, αλλιώς `expr2`
- `for(var in vec) {expr}`: βρόχος επανάληψης `for`: ο μετρητής `var` διατρέχει το διάνυσμα `vec` και εκτελεί την έκφραση `expr` σε κάθε επανάληψη
- `while(cond){expr}`: βρόχος επανάληψης `while`: όσο η συνθήκη `cond` αληθεύει, εκτέλεσε την έκφραση `expr` σε κάθε επανάληψη

## 12.2 Συντομεύσεις πληκτρολογίου

Υπάρχουν αρκετές χρήσιμες συντομεύσεις πληκτρολογίου για το RStudio (βλ. `Help` → `Keyboard Shortcuts`):

- `CRL+ENTER`: στείλε τις εντολές από το παράθυρο σεναρίου στο παράθυρο εντολών
- `↑` ή `↓` στο παράθυρο εντολών: προηγούμενη ή επόμενη εντολή
- `CTRL+1`, `CTRL+2`, κτλ.: εναλλαγή μεταξύ των παραθύρων

Αν και δεν είναι συγκεκριμένες για την R, αποτελούν πολύ χρήσιμες συντομεύσεις:

- `CTRL+C`, `CTRL+X` και `CTRL+V`: αντιγραφή, αποκοπή και επικόλληση
- `ALT+TAB`: μετάβαση σε παράθυρο άλλου προγράμματος
- `↑`, `↓`, `←` ή `→`: μετακίνηση του κέρσορα
- `HOME` ή `END`: μετακίνηση του κέρσορα στην αρχή ή στο τέλος της γραμμής
- `Page Up` ή `Page Down`: μετακίνηση του κέρσορα μια σελίδα πιο πάνω ή πιο κάτω
- `SHIFT+↑/↓/←/→/HOME/END/PgUp/PgDn`: επιλογή

## 12.3 Μηνύματα σφάλματος

- `No such file or directory` ή `Cannot change working directory`  
Σιγουρευτείτε ότι ο κατάλογος εργασίας και τα ονόματα των αρχείων είναι σωστά.
- `Object 'x' not found`  
Η μεταβλητή `x` δεν έχει οριστεί ακόμα. Ορίστε την `x` ή προσθέστε αποστροφους εάν η `x` πρέπει να είναι ακολουθία χαρακτήρων.
- `Argument 'x' is missing without default`  
Δεν έχετε προσδιορίσει το υποχρεωτικό όρισμα `x`.

- +

Η R είναι ακόμα απασχολημένη με κάτι ή έχετε ξεχάσει να κλείσετε κάποια παρένθεση ή αγκύλη. Περιμένετε, πληκτρολογήστε } ή ) ή πιέστε το πλήκτρο ESC.

- Unexpected ')' in ")" ή Unexpected '}' in "}"  
Το αντίθετο από το προηγούμενο. Προσπαθείτε να κλείσετε κάτι που δεν έχει ανοιχθεί ακόμα. Ανοίξτε παρενθέσεις ή αγκύλες.

- Unexpected `else' in "else"

Βάλτε το else μιας δομής if στην ίδια γραμμή με την τελευταία αγκύλη του κομματιού "then": }else{.

- Missing value where TRUE/FALSE needed

Κάτι πάει στραβά στο κομμάτι της συνθήκης (if(x==1)) μιας δομής if. Μήπως το x είναι NA;

- The condition has length > 1 and only the first element will be used

Στο μέρος της συνθήκης (if(x==1)) μιας δομής if, ένα διάνυσμα συγκρίνεται με ένα βαθμωτό. Μήπως το x είναι διάνυσμα; Μήπως εννοούσατε x[i];

- Non-numeric argument to binary operator

Προσπαθείτε να κάνετε υπολογισμούς με κάτι το οποίο δεν είναι αριθμός. Χρησιμοποιείστε την class(...) ώστε να βρείτε τι είναι αυτό που πήγε στραβά ή χρησιμοποιείστε την as.numeric(...) για να μετατρέψετε τη μεταβλητή σε αριθμό.

- Argument is of length zero ή Replacement is of length zero

Η εν λόγω μεταβλητή είναι ίση με NULL, το οποίο σημαίνει ότι είναι κενή, για παράδειγμα δημιουργήθηκε από την c(). Ελέγξτε τον ορισμό της μεταβλητής.