Βιομετρία / Βιοστατιστική

Εισαγωγή στη R

Μάρτιος, 2018

Εισαγωγή στη R

Τι είναι η R

Η R είναι ένα λογισμικό ανοιχτού κώδικα (GNU) για ανάλυση και γραφική απεικόνιση δεδομένων που θεμελιώθηκε από τους Ross Ihaka (Univ. Auckland) και Robert Gentleman (Harvard Biostat). Πρόκειται για μία υλοποίηση της ολοκληρωμένης σουίτας προγραμμάτων S και προσφέρει μεταξύ άλλων,

- μία εκτενή συλλογή εργαλείων ανάλυσης και απεικόνισης δεδομένων για Unix / Linux, Windows και Macintosh, καθώς και
- μια αποτελεσματική γλώσσα προγραμματισμού που μπορεί εύκολα να επεκταθεί από την κοινότητα των χρηστών.

Εγκατάσταση της R

Windows

1. Με ένα πρόγραμμα περιήγησης κατευθύνεστε στη διεύθυνση `http://www.r-project.org/`, όπου και κάνετε κλικ στην επιλογή ”CRAN”.
2. Στη λίστα με διευθύνσεις ανά χώρα που εμφανίζεται, επιλέγετε την πλησιέστερη.
3. Κάνετε κλικ ”Windows”, στο πλαίσιο ”Download and Install R”.
4. Κάνετε κλικ στο ”base”.
5. Κάνετε κλικ στο σύνδεσμο για να κατεβάσετε την τελευταία έκδοση της R (αρχείο *.exe*), την οποία εγκαθιστάτε, εκτελώντας το αρχείο που λάβατε, απαντώντας τις συνήθεις ερωτήσεις.

Έναρξη / Διακοπή / Τερματισμός της R

Windows

- Για έναρξη, ακολουθείτε τη διαδρομή Έναρξη → Όλα τα προγράμματα → R ή κάνετε διπλό κλικ στο εικονίδιο της R στην επιφάνεια εργασίας,
- ή διπλό κλικ σε ένα .RData αρχείο.
- Για διακοπή, το πλήκτρο Esc ή το εικονίδιο τερματισμού θα τερματίσει την τρέχουσα λειτουργία.
- Για τερματισμό, πληκτρολογώντας `q()` θα βγείτε από τη R,
- ή Exit από την επιλογή File, του κεντρικού μενού,
- ή απλά, κλείνετε το παράθυρο.

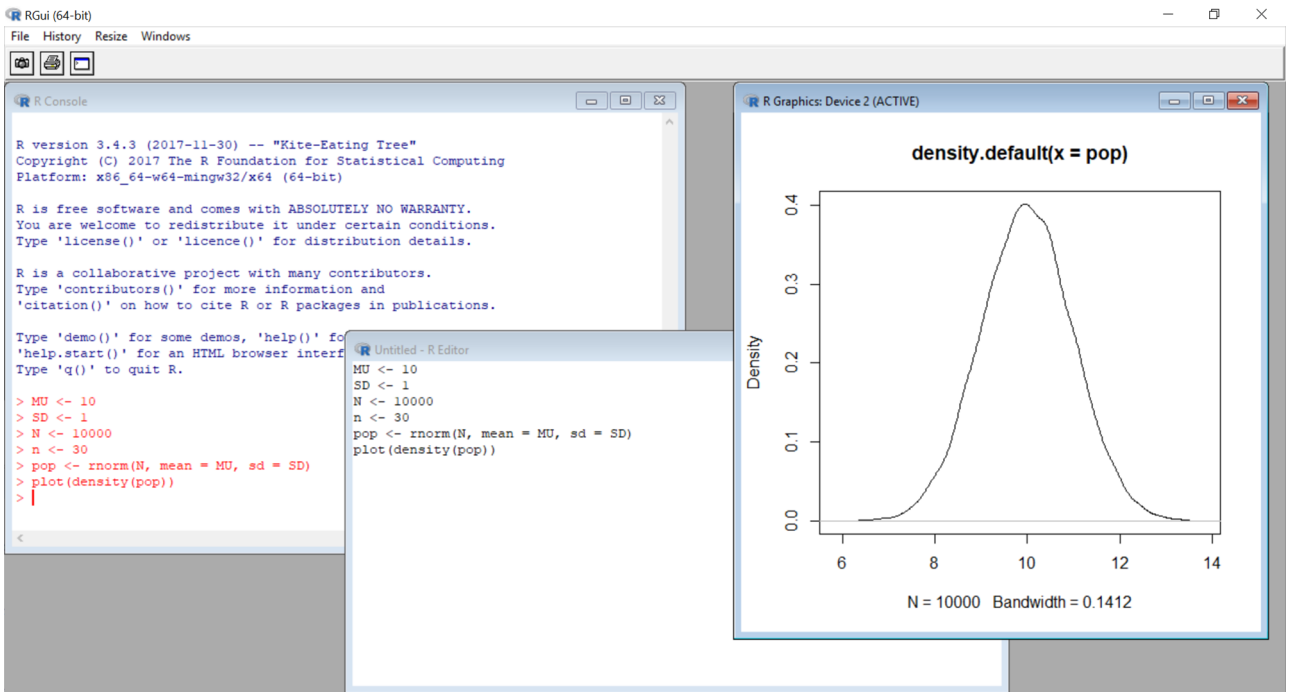
R GUI

Το γραφικό περιβάλλον χρήστη (*Graphical User Interface*; για την έκδοση 3.4.3 των Windows) περιλαμβάνει τα παρακάτω παράθυρα,

(i) R *Console*

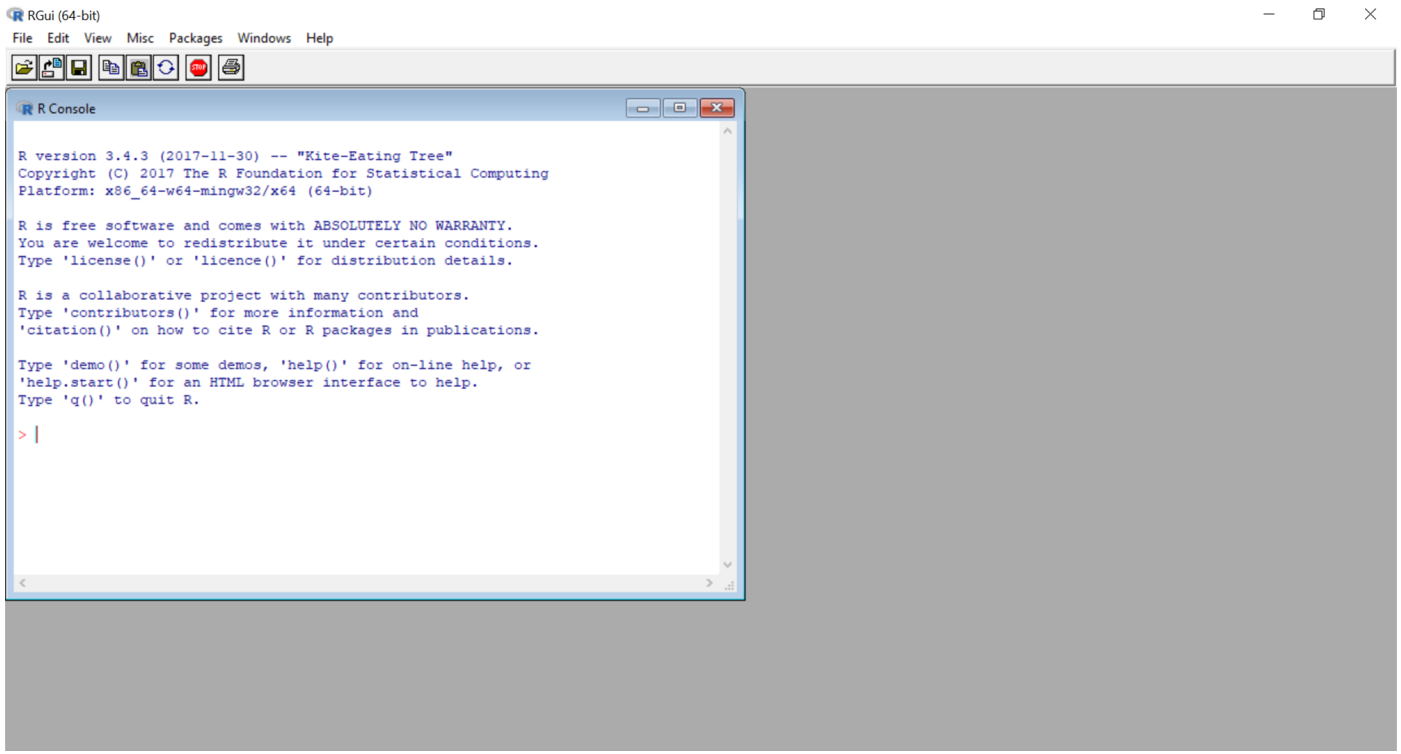
(ii) R *Graphics*

(iii) R *Editor*



R *Console*

Το βασικότερο παράθυρο, το οποίο εμφανίζεται με το άνοιγμα του λογισμικού είναι η κονσόλα (*console*),



1. Η σημαντικότερη λεπτομέρεια, πέρα από τις λοιπές πληροφορίες, είναι το σύμβολο **>** που αποτελεί τη γραμμή εντολών (*command prompt*).
2. Η R είναι ένα διαδραστικό σύστημα. Ο χρήστης πληκτρολογεί και η R αποκρίνεται αναλόγως.
3. Μετά την πληκτρολόγηση κάποιας έκφρασης (*expression*), η R εκτελεί τους υπολογισμούς και εμφανίζει (κατά κανόνα) στην οθόνη την απάντηση. Η απάντηση δεν έχει αποθηκευτεί.
4. Μία απάντηση αποθηκεύεται εκχωρώντας την σε ένα *αντικείμενο* (βάσει της R ορολογίας). Η εκχώρηση εκτελείται συνήθως σιωπηλά. Υπάρχουν διάφοροι τρόποι εκχώρησης, συμπεριλαμβανομένου του "=", συνίσταται, ωστόσο, η χρήση του "<−".

R *Editor*

Ο R *Editor* διευκολύνει την εργασία στη R.

Για να ανοίξετε ένα νέο αρχείο *script* στη R

Από το κύριο μενού, επιλέξτε *File* → *New script*.

Για να εκτελέσετε μια γραμμή του *script*

Τοποθετήστε το *cursor* στη γραμμή και πατήστε Ctrl-R.

Για να εκτελέσετε πολλές γραμμές του *script*

Επιλέξτε τις γραμμές και πατήστε Ctrl-R.

Για να εκτελέσετε ολόκληρο το περιεχόμενο του *script*

Επιλέγετε το σύνολο του περιεχομένου (Ctrl-A) και πατήστε Ctrl-R.

RStudio

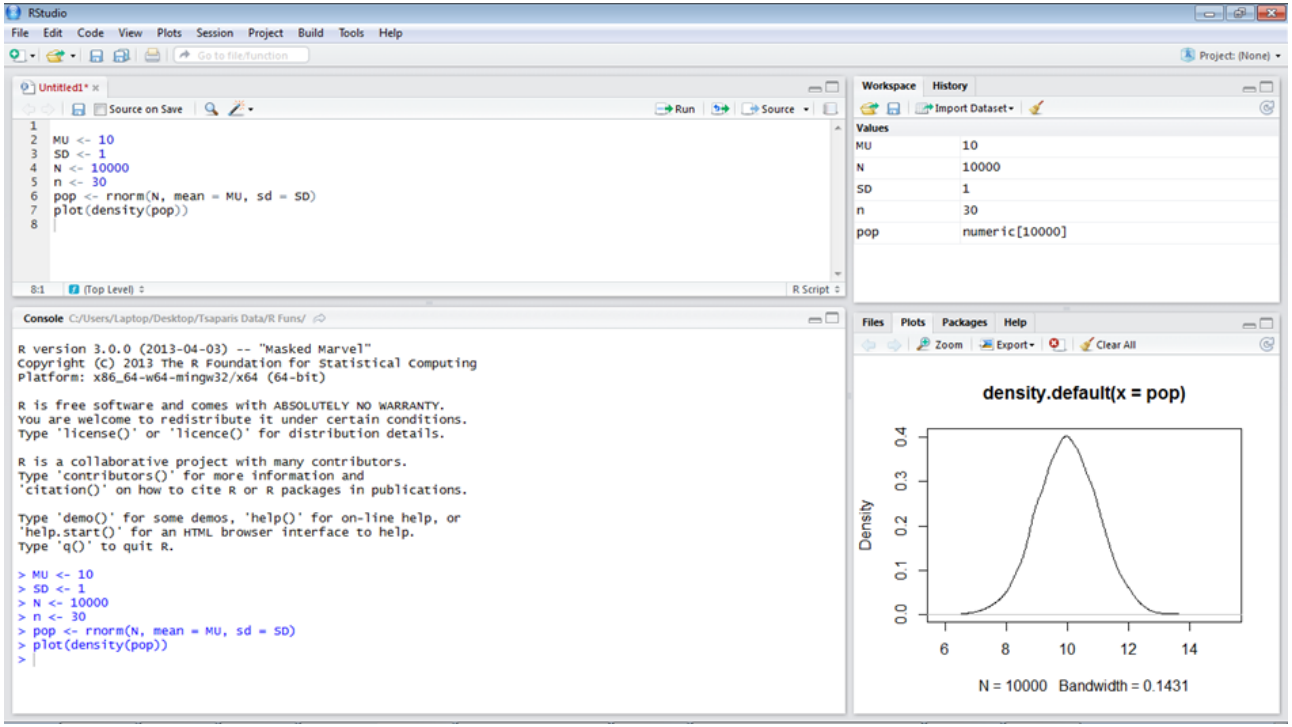
Το RStudio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (*Integrated Development Environment*, IDE), που διευκολύνει την εργασία με R. Ως IDE διαδραστικής γλώσσας προγραμματισμού περιλαμβάνει,

- μια κονσόλα για την αλληλεπίδραση του χρήστη με το πρόγραμμα,
- έναν επεξεργαστή (*editor*) που περιλαμβάνει ένα εκτενές σύνολο από βοηθήματα, όπως συντομεύσεις πληκτρολογίου, αυτόματη μορφοποίηση κώδικα, βοήθεια για το περιβάλλον και πολλά άλλα,
- ένα σύστημα περιήγησης σε αρχεία και βοηθητικά έγγραφα,
- καθώς επίσης και αρκετά εργαλεία διαχείρισης γραφημάτων.

Η προεπιλεγμένη διάταξη του RStudio χωρίζει το περιβάλλον εργασίας σε τέσσερα κύρια τμήματα,

- τον *editor*, στην επάνω αριστερή πλευρά, για προβολή ή / και επεξεργασία, με τη δυνατότητα διαχείρισης πολλαπλών αρχείων,
- τη *console*, κάτω αριστερά, για την αλληλεπίδραση του χρήστη με το πρόγραμμα,
- διάφορες καρτέλες (τρέχον περιβάλλον, ιστορικό κ.τ.λ.), επάνω δεξιά, σχετικές με το *session* και
- διάφορες καρτέλες για τα αρχεία, τα γραφήματα, τα πακέτα και τη βοήθεια κάτω δεξιά.

Εισαγωγή στη R



Το RStudio είναι διαθέσιμο στη διεύθυνση `http://www.rstudio.com/download` και η εγκατάσταση του είναι απλή. Για τη λειτουργία του απαιτείται μία σύγχρονη έκδοση της R (Ver. 3.0.1 ή νεότερη).

Βασικές Λειτουργίες

Εφαρμογή II

```
> 2+3
[1] 5

> sum(rivers)
[1] 83357

> mean(Rivers)
Error in mean(Rivers) : object 'Rivers' not found

> m <- mean(rivers); v <- var(rivers)
> sqrt(v)/m                                  # Coefficient of variation
[1] 0.8353922
```

Σχόλια.

1. Ο δείκτης `[1]` καθορίζει τη θέση του στοιχείου που έπεται, στο αντίστοιχο αντικείμενο (π.χ., στην πρώτη θέση του διανύσματος).

2. Η R διακρίνει πεζούς και κεφαλαίους χαρακτήρες (*case-sensitive*; τα r και R είναι διαφορετικά). Πρακτικά, οποιοσδήποτε συνδυασμός γραμμάτων, αριθμών και συμβόλων μπορεί να χρησιμοποιηθεί για ονοματοδοσία, εκτός από κάποιες επιλογές (π.χ. `for`, `if`, `TRUE`, `FALSE`).

3. Τα αντικείμενα που είναι αποθηκευμένα (στο *session*) εμφανίζονται, πληκτρολογώντας `objects()` ή `ls()`.

4. Οι διαδοχικές εκφράσεις (εντολές) διακρίνονται είτε με το σύμβολο ";" (*semi-colon*), ή με μία νέα γραμμή.

5. Το ιστορικό των εντολών είναι διαθέσιμο χρησιμοποιώντας τα κατάλληλα κουμπιά πάνω/κάτω (*arrow keys*) ή πληκτρολογώντας `history()`.

6. Το σύμβολο δίεσης (*hash*), #, χρησιμοποιείται για σχολιασμό. Οτιδήποτε ακολουθεί μετά από αυτό αγνοείται από τη R.

7. Εάν μία έκφραση δεν είναι ολοκληρωμένη και εκτελεστεί, η R αλλάζει το σύμβολο της γραμμής εντολών, από > σε +, στην επόμενη γραμμή και συνεχίζει να διαβάζει μέχρι να ολοκληρωθεί η έκφραση.

Βασικοί Τελεστές

Μεταξύ των απλών εντολών της R είναι αριθμητικές πράξεις όπως πρόσθεση και πολλαπλασιασμός. Με τη χρήση κατάλληλων τελεστών μπορούμε να κάνουμε λογικές συγκρίσεις, ενώ η R παρέχει ενσωματωμένες συναρτήσεις και σταθερές έτοιμες για χρήση.

1. Οι κλασικοί τελεστές `+`, `-`, `*`, `/`, `^` είναι διαθέσιμοι, όπου το `^` χρησιμοποιείται για εκθέτες.
2. Επιπλέον, χρήσιμοι τελεστές είναι `%/%` για το ακέραιο μέρος της διαίρεσης και `%%` για το υπόλοιπο (*modulo operation*).
3. Κοινές συναρτήσεις είναι επίσης διαθέσιμες, συμπεριλαμβανομένου των `abs`, `sign`, `log`, `sqrt`, `exp`, `sin`, `cos`, `tan` με την αντί-

στοιχη συνήθη ερμηνεία.

4. Οι σταθερές π, e και i είναι διαθέσιμες ως `pi`, `exp(1)` και `1i`.
5. Οι λογικοί τελεστές <, <=, >, >= είναι επίσης διαθέσιμοι, ενώ το σύμβολο == χρησιμοποιείται για έλεγχο ισότητας και το `!=` για άρνηση ισότητας. Επιπλέον, το `&` συμβολίζει την τομή, το `|` την ένωση και το `!` την άρνηση, εν γένει.

Σχόλια.

1. Οι βασικές πράξεις επιστρέφουν έναν αριθμό. Οι λογικές πράξεις `TRUE` ή `FALSE`.
2. Η `R` χρησιμοποιεί τον κοινό κανόνα διαδοχής πράξεων (BODMAS).
3. Οι λογικές εκφράσεις χρησιμοποιούνται και σε αριθμητικές πράξεις, με το `FALSE` να αντιστοιχεί στο `0` και το `TRUE` στο `1`.

Ειδικές Τιμές

Υπάρχουν ορισμένοι τύποι δεδομένων που πρέπει να αντιμετωπίζονται καταλλήλως στους υπολογισμούς.

- **NA** (*not available*). Η τιμή **NA** χρησιμοποιείται από το R για ελλιπή στοιχεία (*missing values*) των δεδομένων. Η τιμή αυτή δεν είναι ένα *string* χαρακτήρων, οπότε το "**NA**" θα αντιμετωπίζεται διαφορετικά από την τιμή **NA**.
- **Inf** (*infinity*). Το αποτέλεσμα, π.χ. της διαίρεσης κάθε μη μηδενικού αριθμού με το μηδέν.
- **NaN** (*not a number*). Το αποτέλεσμα, π.χ. της διαίρεσης του μηδέν με το μηδέν ή του λογαρίθμου ενός αρνητικού αριθμού.

Βοήθεια στη R

Η R διαθέτει ένα εκτεταμένο σύστημα βοήθειας,

- πληκτρολογώντας `?help` λαμβάνετε πληροφορίες σχετικά με τη λειτουργία της βοήθειας,
- πληκτρολογώντας `help.search("subject")` λαμβάνετε συνολικές πληροφορίες πάνω στο θέμα,
- πληκτρολογώντας `help(command)` ή `?command` στην περίπτωση που γνωρίζετε το όνομα της εντολής,
- πληκτρολογώντας `?"&&"` για τους τελεστές ή λέξεις όπως `if`,

- πληκτρολογώντας `help.start()` για διαδικτυακή βοήθεια στη R,
- πληκτρολογώντας `library(help = MASS)` για πληροφορίες σχετικά ένα πακέτο, π.χ. MASS.
- Το *Help* από το κεντρικό μενού παρέχει ακόμα περισσότερες επιλογές.

Χρήση της R

Η χρησιμοποίηση της R έγκειται στην εκτέλεση εντολών. Για να γίνει χρήση μίας εντολής, μία σειρά από ορίσματα θα πρέπει να πάρουν τιμές. Συνήθως δεν χρειάζεται να καθοριστούν όλα τα ορίσματα, καθώς πολλά έχουν προεπιλεγμένες τιμές. Ορίσματα και προεπιλεγμένες τιμές μίας εντολής μπορεί κανείς να τα δει με την εντολή `args`,

```
> args(log)
function (x, base = exp(1))
```

Ορισμένες εντολές στη R συμπεριφέρονται διαφορετικά ανάλογα με την *κλάση* του αντικειμένου στο οποίο εφαρμόζονται. Οι εντολές αυτές καλούνται *generic*. Το συνολικό πλαίσιο που περιλαμβάνει *generic* εντολές ονομάζεται *object-oriented* (αντικειμενοστρεφής) προγραμματισμός.

Αντικείμενα στη R

Η R είναι ένα λογισμικό με βασικό δομικό στοιχείο το *αντικείμενο*. Τα δεδομένα, οι εντολές και τα αποτελέσματα θεωρούνται όλα αντικείμενα για τα οποία ισχύουν οι ίδιοι κανόνες.

Αποθήκευση, Ανάκτηση και Διαγραφή

- Όλα τα αντικείμενα διατηρούνται στη μνήμη και δεν είναι αποθηκευμένα κάπου εκτός αν εκτελέσετε αναλόγως.
- Για να αποθηκεύσετε τα αντικείμενα, `x` και `y`, σε ένα αρχείο που ονομάζεται `xy.RData`

```
> save(x, y, file = "xy.RData")
```

- Για να αποθηκεύσετε ολόκληρο το *session*, πληκτρολογείτε `save.image()` ή χρησιμοποιείτε την επιλογή από το κεντρικό μενού *File / Save Workspace*. Η R θα σας ζητήσει ένα όνομα αρχείου.
- Για να φορτώσετε ένα αρχείο που αποθηκεύσατε νωρίτερα `xy.RData`, πληκτρολογείτε `load("xy.RData")`.
- Όταν κλείνετε τη R, θα ρωτηθείτε αν θέλετε να αποθηκεύσετε την εργασία σας. Δεν είναι (σχεδόν) ποτέ καλή ιδέα να απαντήσετε ναι στην παραπάνω ερώτηση!
- Η εντολή `rm` μπορεί να χρησιμοποιηθεί για τη διαγραφή ενός αντικειμένου. Η σύνταξη της είναι,

```
> rm(object1, object2, ...)
```

- Υπάρχει επίσης και η εντολή `remove`,

```
> remove(list = ls(pattern = "^f"))
```

- Το περιεχόμενο της μνήνης διαγράφεται συνολικά ως εξής,

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

- Σημειώνεται ότι δεν υπάρχει αναίρεση διαγραφής. Όταν ένα αντικείμενο έχει διαγραφεί, έχει διαγραφεί!

Αντικείμενα Δεδομένων

Τα κύρια αντικείμενα δεδομένων στη R είναι,

(i) διάνυσμα (*vector*)
(ii) πίνακας (*matrix*)
(iii) πίνακας μεγαλύτερης διάστασης (*array*)
(iv) λίστα (*list*)
(v) ορθογώνια λίστα (*data frame*)

Ορισμένα αντικείμενα μπορούν να περιέχουν μόνο ένα τύπο δεδομένων (όπως τα διανύσματα και οι πίνακες γενικότερα), ενώ τα άλλα υποστηρίζουν διάφορους τύπους. Περισσότερες λεπτομέρειες για τους διαφορετικούς τύπους αντικειμένων δίνονται ακολούθως, ξεκινώντας με τα διανύσματα.

Διανύσματα

Τα διανύσματα είναι βασικά αντικείμενα της R. Ένα διάνυσμα είναι μία συλλογή στοιχείων του ιδίου τύπου, ή, κατά τη R ορολογία, του ιδίου *mode*. Ένα διάνυσμα, δηλαδή, μπορεί να περιέχει είτε αριθμούς (mode = `numeric`), χαρακτήρες (`character`), ή λογικές τιμές (`logical`) αλλά όχι κάποιο συνδυασμό των παραπάνω. Η εντολή που χρησιμοποιείται κυρίως για τη δημιουργία διανυσμάτων είναι η `c()`.

```
> c(0,1,1,2,3,5,8,13,21)
[1] 0 1 1 2 3 5 8 13 21
> c("Department", "of", "Public", "Health")
[1] "Department" "of" "Public" "Health"
> c(TRUE,TRUE,FALSE,TRUE)
[1] TRUE TRUE FALSE TRUE
```

Τα διανύσματα (όπως και οι λίστες που αναφέρονται παρακάτω) μπορούν να έχουν ονόματα - κάθε στοιχείο (ή συνιστώσα μίας λίστας) έχει ένα όνομα. Οι ακόλουθες εντολές είναι ισοδύναμες, διαφέροντας μόνο στην ονοματοθεσία των στοιχείων, όπου στην πρώτη γίνεται ταυτόχρονα με τη δημιουργία του διανύσματος ενώ στη δεύτερη σε επόμενο στάδιο.

```
> bmidata <- c(John=12.61, Mary=17.63, Bob=9.77, Ann=13.93)
> bmidata <- c(12.61, 17.63, 9.77, 13.93)
> names(bmidata) <- c("John", "Mary", "Bob", "Ann")
> bmidata
 John  Mary   Bob   Ann
12.61 17.63  9.77 13.93
```

Παρατηρείστε ότι δεν υπάρχει πλέον δείκτης που καθορίζει τη θέση του στοιχείου στο αντικείμενο.

Αριθμητικές πράξεις Διανυσμάτων

Οι πράξεις με διανύσματα είναι ένα από τα βασικά προτερήματα της R. Οι συνήθεις αριθμητικές / λογικές πράξεις μπορούν να εφαρμοστούν σε διανύσματα κατά στοιχείο (*element-wise*). Οι περισσότερες εντολές λειτουργούν κατά αυτόν τον τρόπο για τα διανύσματα και παράγουν ένα διάνυσμα ως τελικό αποτέλεσμα.

```
> fib <- c(0, 1, 1, 2, 3, 5, 8, 13, 21)
> fib + 2
[1]  2  3  3  4  5  7 10 15 23
> fib > 4
[1] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
> 2^fib
[1]       1       2       2       4       8      32     256    8192
[9] 2097152
```

```
> log(fib)
[1]      -Inf 0.0000000 0.0000000 0.6931472 1.0986123 1.6094379
[7] 2.0794415 2.5649494 3.0445224
```

Η λειτουργία αυτή (*vectorization*) έχει δύο σημαντικά πλεονεκτήματα. Η πρώτη και η πιο προφανής είναι η ευκολία. Η δεύτερη είναι η ταχύτητα. Οι περισσότερες *vectorized* εντολές εκτελούνται σε κώδικα C και είναι ουσιαστικά γρηγορότερες από τις αντίστοιχες που είναι γραμμένες εξ ολοκλήρου στη R.

Μερικές χρήσιμες εντολές για διανύσματα,

`min`, `max`, `range`, `length`, `sum`, `prod`, `sort` και `order`.

Κανόνας Ανακύκλωσης

Οι αριθμητικές πράξεις μεταξύ διανυσμάτων είναι προβλέψιμες για διανύσματα ίσου μήκους. Διαφορετικά, η R εφαρμόζει τον κανόνα ανακύκλωσης (*Recycling Rule*), όπου το αποτέλεσμα είναι ένα διάνυσμα με μήκος ίσο του μεγαλύτερου διανύσματος. Τα μικρότερα διανύσματα ανακυκλώνονται όσο χρειαστεί. Η R προειδοποιεί μόνο όταν το μήκος του μεγαλύτερου δεν είναι ακέραιο πολλαπλάσιο του μικρότερου.

```
> x <- c(1,2,3,4,5,6) ; y <- c(1,2,3) ; z <- c(1,2,3,4)
> x + y
[1] 2 4 6 5 7 9
> x + z          # Oops! Length of x not a multiple of 4
[1] 2 4 6 8 6 8
Warning message:
In x + z : longer object length is not a multiple of shorter object
```

Επιλογή στοιχείων Διανύσματος

Για την επιλογή στοιχείων ενός διανύσματος χρησιμοποιούνται οι αγκύλες (*square brackets*; π.χ. `x[5]`). Ο δείκτης μέσα στις `[]` καθορίζει τα επιλεγόμενα στοιχεία, μπορεί να είναι μία από τις ακόλουθες επιλογές.

1. Ένα διάνυσμα θετικών ακεραίων, οποιουδήποτε μήκους και σε οποιαδήποτε σειρά, που καθορίζει τα προς επιλογή στοιχεία,

```
> fib[c(2, 5)]                    # Select elements 2 and 5
[1] 1 3
```

2. Ένα διάνυσμα αρνητικών ακεραίων, για όσα αποκλείονται,

```
> fib[-c(2, 5)]                   # Exclude elements 2 and 5
[1]  0  1  2  5  8 13 21
```

Μίξη θετικών και αρνητικών δεικτών δεν μπορεί να χρησιμοποιηθεί, ενώ το 0 επιστρέφει ένα διάνυσμα μηδενικού μήκους.

3. Ένα λογικό διάνυσμα. Το λογικό διάνυσμα θα πρέπει να είναι ιδίου μήκους με το αρχικό, διαφορετικά η R εφαρμόζει το κανόνα ανακύκλωσης. Τα στοιχεία που αντιστοιχούν στο TRUE του διανύσματος δείκτη θα επιλεγούν,

```
> fib %% 2 == 0
[1]  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE
> fib[fib %% 2]                    # Select the even elements
[1] 0 2 8
> (1:length(fib))[fib %% 2 == 0]
[1] 1 4 7
```

4. Ένα διάνυσμα *string* χαρακτήρων. Η επιλογή αυτή έχει αξία όταν το αρχικό διάνυσμα έχει ονόματα. Τα ονόματα πρέπει να χρησιμοποιηθούν όπως οι θετικοί ακέραιοι στο 1.

```
> bmidata[c("John", "Bob")]
 John   Bob
12.61  9.77
```

Επιπλέον εντολές Διανυσμάτων

Η R έχει περισσότερους τρόπους να δημιουργεί διανύσματα.

(i) Η άνω και κάτω τελεία (`:`) είναι χρήσιμη για τη δημιουργία ακολουθιών ακεραίων. Το `:` έχει προτεραιότητα μέσα σε μία έκφραση.

(ii) Η `seq` είναι μία περισσότερο γενικευμένη εντολή για τη δημιουργία ακολουθιών.

(iii) Η εντολή `rep` μπορεί να χρησιμοποιηθεί για την επανάληψη δομών με διαφορετικούς τρόπους.

(iv) Η εντολή `paste` δημιουργεί διανύσματα χαρακτήρων ως αποτέλεσμα συνένωσης δύο ή περισσοτέρων διανυσμάτων αντίστοιχων στοιχείων.

Factors

- Οι *factor*s είναι μεταβλητές που παίρνουν τιμές μέσα από ένα πεπερασμένο σύνολο διακριτών τιμών. Ορίζονται ως διανύσματα και μπορεί να είναι,
  - αριθμητικά π.χ. δόσεις φαρμάκων με τιμές 1mg, 2mg, 5mg, ή
  - χαρακτήρων π.χ. ομάδα αίματος με τιμές A, B, AB, 0

  Η εντολή `factor` κωδικοποιεί ένα διάνυσμα διακριτών τιμών σε ένα *factor* :

```
> f <- factor(vector, levels = vector)
```

- Παρ' ότι οι *factor*s αποθηκεύονται ως αριθμοί, μαζί με τα αντίστοιχα ονόματα (*labels*), δεν μπορούν να χρησιμοποιηθούν ως αριθμητικά διανύσματα, π.χ.

```
> mean(f)             # Meaningless
```

- Μία περισσότερο χρήσιμη εντολή για *factor*s είναι η `table` με την οποία δίνονται οι συχνότητες των διακριτών τιμών του διανύσματος.

Εφαρμογή III. Gladstone Survey

Το αρχείο `gladstone.III.RData` περιέχει μέρος των δεδομένων για το σύνολο των παρατηρήσεων της μελέτης του Gladstone. Ανοίξτε το αρχείο στη R και απαντήστε τα παρακάτω,

1. Πόσα αντικείμενα είναι αποθηκευμένα.
2. Πόσες παρατηρήσεις έχει το αντικείμενο `head`.
3. Επιλέξτε τις 8 πρώτες παρατηρήσεις από το αντικείμενο `head` και ελέγξτε πόσα από αυτά είναι μεγαλύτερα του 3924.25.
4. Επαναλάβετε για το αντικείμενο `brain` και την τιμή 1373.63.
5. Διαγράψτε το αντικείμενο `head` και αποθηκεύστε το *session* στην επιφάνεια εργασίας.

Πίνακες

Στη R, ένας πίνακας είναι απλά ένα διάνυσμα με διαστάσεις. Όλα τα στοιχεία ενός πίνακα, συνεπώς, θα πρέπει να είναι το ίδιου *mode* (π.χ. αριθμοί, λογικές τιμές ή χαρακτήρες).

Η δημιουργία ενός πίνακα γίνεται, κυρίως, με την εντολή `matrix`,

```
matrix(data = vector, nrow = value, ncol = value, byrow = FALSE)
> m <- matrix(c(1.1,1.2,2.1,2.2,3.1,3.2), nrow = 2, ncol = 3)
     [,1] [,2] [,3]
[1,]  1.1  2.1  3.1
[2,]  1.2  2.2  3.2
```

Τα στοιχεία καταχωρούνται κατά στήλες, εκτός αν καθοριστεί το όρισμα `byrow=TRUE`. Εάν τα δεδομένα δεν φτάνουν να συμπληρώσουν τον πίνακα, η R θα εφαρμόσει το γνωστό *Recycling* κανόνα.

Επιλογή στοιχείων Πίνακα

Σε αντιστοιχία με το διάνυσμα, για την επιλογή στοιχείων ενός πίνακα χρησιμοποιούνται δύο δεικτών, χωριζόμενοι με κόμμα. Εάν ένας δείκτης παραληφθεί, ολόκληρη η γραμμή ή η στήλη δίνεται,

```
> m[2, 3]
[1] 3.2
> row1 <- m[1,]
[1] 1.1 2.1 3.1
> col3 <- m[,3]
[1] 3.1 3.2
```

Όπως με τα διανύσματα, αρνητικοί ακέραιοι, λογικά διανύσματα ή *string*s χαρακτήρων μπορούν να χρησιμοποιηθούν για την επιλογή συγκεκριμένων γραμμών ή στηλών,

```
> m[-2, -1]
[1] 2.1 3.1
> m[, m[1,]>2]
     [,1] [,2]
[1,]  2.1  3.1
[2,]  2.2  3.2
```

Οι συνήθεις αριθμητικοί τελεστές εκτελούν πράξεις ανά στοιχείο (*element-wise*), όπως επίσης και οι εντολές `log`, `exp`, `sqrt` κ.τ.λ.. Για πράξεις πινάκων χρησιμοποιούνται το σύμβολο `%*%` για πολλαπλασιασμό και οι εντολές `t`, `solve` και `diag` για ανάστροφο, αντίστροφο και διαγώνιο πίνακα, αντίστοιχα.

Πίνακας δημιουργείται, επιπλέον, ενώνοντας διανύσματα κατά γραμμές (στήλες) με την εντολή `rbind` (`cbind`), ή και με την εντολή `dim`.

Εφαρμογή IV. Gladstone Survey

Το αρχείο `gladstone.III.RData` περιέχει μέρος των δεδομένων για το σύνολο των παρατηρήσεων της μελέτης του Gladstone. Ανοίξτε το αρχείο στη R και απαντήστε τα παρακάτω,

1. Δημιουργήστε ένα πίνακα `gladstone`, όπου η πρώτη στήλη αποτελείται από τα στοιχεία του διανύσματος `head` και η δεύτερη από τα αντίστοιχα του `brain`.
2. Δημιουργήστε, εκ νέου ένα πίνακα `t.gladstone`, που στην πρώτη γραμμή θα βρίσκεται το διάνυσμα `head` και στη δεύτερη το `brain`. Διαγράψτε τα αντικείμενα `brain` και `head`.
3. Υπολογίστε τον συνολικό κρανιακό όγκο (σε cm^3) και το συνολικό εγκεφαλικό βάρος (σε kg).

Λίστες

Μια λίστα είναι μια δομή που δύναται να περιέχει στοιχεία διαφορετικών τύπων και κατά συνέπεια αποτελεί ένα τρόπο αποθήκευσης αντικειμένων διαφορετικών τύπων. Μία λίστα μπορεί να δημιουργηθεί χρησιμοποιώντας την εντολή `list`, η οποία επιτρέπει την ονοματοδοσία των στοιχεία της με την κατάλληλη σύνταξη, `name = value`,

```
> lst <- list(a = 1:5, b = matrix(10:15,2,3), "converged")
$a
[1] 1 2 3 4 5
$b
     [,1] [,2] [,3]
[1,]   10   12   14
[2,]   11   13   15
[[3]]
[1] "converged"
```

Επιλογή στοιχείων Λίστας

Υπάρχουν περισσότεροι, του ενός, τρόποι επιλογής στοιχείων μίας λίστας,

1. χρησιμοποιώντας διπλές αγκύλες (*double-square-bracket*, `[[]]`) και τη θέση ενός στοιχείου,

```
> lst[[1]]
[1] 1 2 3 4 5
```

2. ή, το όνομα του στοιχείου. Το ίδιο ισχύει με το σύμβολο `$` και το όνομα του στοιχείου, δηλαδή,

```
> lst$a                 # or lst[["a"]]
[1] 1 2 3 4 5
```

3. Διαφορετικά, χρησιμοποιώντας μονή αγκύλη (*single-square-bracket*, `[]`) και τη θέση του στοιχείου

```
> lst[1]
$a
      [1] 1 2 3 4 5
```

4. ή το όνομα του αντικειμένου,

```
> lst["a"]
$a
[1] 1 2 3 4 5
```

Χρησιμοποιώντας τους δύο τελευταίους τρόπους παράγεται μία λίστα με ένα στοιχείο, σε αντίθεση με τους δύο πρώτους που επιστρέφουν το στοιχείο με τη μορφή που βρίσκεται στη λίστα.

Επεξεργασία Λίστας

Για να αφαιρεθεί ένα στοιχείο από μία λίστα, εκχωρείται το κενό αντικείμενο, NULL, στο στοιχείο αυτό. Για να προστεθεί ένα νέο στοιχείο, αυτό εκχωρείται στη λίστα με κάποιο όνομα (ή αριθμό θέσης),

```
> lst[[3]] <- NULL            # Delete the third element
$a
[1] 1 2 3 4 5
$b
     [,1] [,2] [,3]
[1,]   10   12   14
[2,]   11   13   15
> lst$d <- letters[1:3]       # Add a new element named "d"
$a
[1] 1 2 3 4 5
$b
```

```
     [,1] [,2] [,3]
[1,]   10   12   14
[2,]   11   13   15
$d
[1] "a" "b" "c"
```

Η εντολή `c()` ενώνει δύο ή περισσότερες λίστες και κατά συνέπεια είναι ένας επιπλέον τρόπος προσθήκης νέου στοιχείου σε μία υπάρχουσα λίστα,

```
> lst <- c(lst, e = letters[1:5])
$a
[1] 1 2 3 4 5
$b
     [,1] [,2] [,3]
[1,]   10   12   14
[2,]   11   13   15
```

```
[[3]]
[1] "converged"
$e1
[1] "A"
$e2
[1] "B"
$e3
[1] "C"
$e4
[1] "D"
$e5
[1] "E"
```

Data Frames

Ένα *data frame* είναι μια ορθογώνια δομή δεδομένων, ένα αντικείμενο, δηλαδή που έχει γραμμές και στήλες. Δεν είναι, ωστόσο, πίνακας καθώς διαφορετικές στήλες μπορούν να έχουν διαφορετικούς τύπους δεδομένων (*modes*). Αντίθετα, ένα *data frame* είναι μία λίστα. Για παράδειγμα, εξετάστε το ενσωματωμένο *data frame* `ToothGrowth`,

```
> ToothGrowth
     len supp dose
1    4.2   VC  0.5
2   11.5   VC  0.5
3    7.3   VC  0.5
4    5.8   VC  0.5
...
59  29.4   OJ  2.0
60  23.0   OJ  2.0
```

Τα *data frames* είναι πολύ χρήσιμα επειδή τα πραγματικά δεδομένα έχουν, συχνά, αυτή τη μορφή, είναι ορθογώνια, με κάθε σειρά να αντιπροσωπεύει μία παρατήρηση και κάθε στήλη μία μεταβλητή.

Ένα *data frame* μπορεί να δημιουργηθεί από διανύσματα και πίνακες με την εντολή `data.frame`,

```
> age <- c(23,43,34,26,56)
> sex <- c('F','M','F','F','M')
> smoke <- c('Yes','No','No','Yes','Yes')
> dfrm <- data.frame(age,sex,smoke)
```

```
  age sex smoke
1  23   F  Yes
2  43   M   No
3  34   F   No
4  26   F  Yes
5  56   M  Yes
```

Επιλογή στοιχείων *Data Frame*

Επειδή ένα *data frame* είναι μία λίστα με ορθογώνια δομή, η R έχει δύο διαφορετικούς τρόπους πρόσβασης στο περιεχόμενό του,

- χρησιμοποιώντας τους τρόπους επιλογής στοιχείων λίστας, όπως `dfrm[i]`, `dfrm[[i]]`, ή `dfrm$name`,
- ή, διαφορετικά, με τους αντίστοιχους τρόπους ενός πίνακα, όπως `dfrm[i,j]`, `dfrm[i,]`, ή `dfrm[,j]`. Για παράδειγμα, οι ακόλουθες εντολές είναι ισοδύναμες,

```
> dfrm[,1]                    # (Matrix type) Column 1
> dfrm[[1]]                   # (List type) Item 1
> dfrm[,"age"]                # (Matrix type) Column named "age"
> dfrm$age                    # (List type) Item named "age"
[1] 23 43 34 26 56
```

Εφαρμογή V. Gladstone Survey

Το αρχείο `gladstone.V.RData` περιέχει μέρος των δεδομένων για το σύνολο των παρατηρήσεων της μελέτης του Gladstone. Ανοίξτε το αρχείο στη R και απαντήστε τα παρακάτω,

1. Δημιουργήστε ένα *data frame* με τις μεταβλητές `gender`, `head` και `brain`.
2. Ποιο είναι το φύλο, το κρανιακό μέγεθος και το εγκεφαλικό βάρος της παρατήρησης 58.
3. Αντίστοιχα για τις παρατηρήσεις 133 έως 136; Ποιο το συνολικό βάρος για αυτές τις παρατηρήσεις.
4. Πόσοι είναι οι άνδρες και πόσες οι γυναίκες της έρευνας.

Εισαγωγή Δεδομένων

Όλες οι στατιστικές επεξεργασίες βασίζονται σε δεδομένα, τα οποία βρίσκονται, συνήθως, μέσα σε αρχεία (βάσεις δεδομένων). Η R διαθέτει μια σειρά από βασικές λειτουργίες που διευκολύνουν την ανάγνωση δεδομένων από αρχεία, μερικές εκ των οποίων ακολουθούν.

Για περισσότερο σύνθετες λειτουργίες, κανείς μπορεί να ανατρέξει στον οδηγό "R Data Import/Export", διαθέσιμος στη διεύθυνση `http://cran.r-project.org/doc/manuals/R-data.pdf`.

Εισάγωντας δεδομένα μέσω πληκτρολογίου

Για μικρά σύνολα δεδομένων, χρησιμοποιούνται κυρίως οι παρακάτω δύο εντολές,

```
> x <- c(2.9, 3.1, 3.4, 3.4, 3.7, 3.7, 2.8, 2.5)
> x <- scan()

1: 2.9 3.1 3.4 3.4 3.7 3.7 2.8 2.5
9:
```

όπου η δεύτερη μέθοδος εισαγωγής τερματίζεται με μία κενή εισαγωγή.

Εισάγωντας δεδομένα σε μορφή πινάκων

Συχνά, τα δεδομένα βρίσκονται μορφοποιημένα σε κάποιο αρχείο. Σε αυτή την περίπτωση,

1. χρησιμοποιώντας την εντολή `scan`

```
scan(file = "filename", what = list, ...)
```

όπου,

`file` το όνομα του αρχείου, με την πλήρη διεύθυνση του (τοπικά ή διαδυκτιακά) σε εισαγωγικά. Στα Windows χρησιμοποιείται το *slash* (/) ή το διπλό *blackslash* (\\ ) αλλά όχι το μονό (\ ) για τον καθορισμό της διεύθυνσης.

`what` λίστα με τα *modes* των μεταβλητών. Τα *modes* μπορεί να είναι είτε αριθμητικά ή χαρακτήρες που αντιστοιχούν στο `0` ή το `""`. Η `scan` θα επαναλάβει τις τιμες των *modes* μέχρι να χαρακτηριστούν όλες οι στήλες. Εάν η λίστα έχει ονόματα, τα ονόματα αυτά θα χρησιμοποιηθούν στο παραγώμενο αποτέλεσμα.

```
> cat("25.09 26.73 18.74 11.01 27.85", file="ex1.dat", sep="\n")
> simples <- scan("ex1.dat", what = 0)
Read 5 items
> simples
[1] 25.09 26.73 18.74 11.01 27.85
> cat("2 3 5","11 13 17", file = "ex2.dat", sep = "\n")
> triples <- scan("ex2.dat", what = list(x = 0, y = "", z = 0))
Read 2 records
> triples
$x
[1]  2 11

$y
[1] "3"  "13"

$z
[1]  5 17
```

2. Χρησιμοποιώντας την εντολή `read.table`. Η `read.table` είναι ο πιο κοινός τρόπος εισαγωγής δεδομένων σε ένα *data frame*,

```
read.table(file = "filename", ...)
```

Έστω ένα αρχείο "statisticians.txt" το οποίο περιέχει τα δεδομένα,

```
Fisher   1890   1962
Pearson  1857   1936
Cox      1900   1978
Yates    1902   1994
```

Το περιεχόμενο του αρχείου μπορεί να διαβαστεί στη R ως εξής,

```
> dfrm <- read.table("C:/.../statisticians.txt")
       V1   V2   V3
1  Fisher 1890 1962
2 Pearson 1857 1936
3     Cox 1900 1978
4   Yates 1902 1994
```

Επιλογές για τη *read.table*

Οι μορφοποιήσεις των αρχείων ποικίλουν αρκετά και για αυτό η εντολή `read.table` έχει πολλές επιλογές για τον έλεγχο των διαφοροποιήσεων αυτών. Μερικές εκ των πλέον χρήσιμων είναι,

`header = TRUE`
Όρισμα που καθορίζει εάν η πρώτη γραμμή περιέχει ονόματα μεταβλητών (προεπιλεγμένη τιμή `FALSE`)

`skip = number`
Όρισμα που καθορίζει τον αριθμό των (αρχικών) γραμμών που αγνοούνται κατά την ανάγνωση του αρχείου (προεπιλεγμένη τιμή `0`)

sep = character
Όρισμα που καθορίζει το χαρακτήρα που διακρίνει τις τιμές του αρχείου (προεπιλεγμένη τιμή οποιοσδήποτε κενός χαρακτήρας, *white space*)

na.strings = string
Όρισμα που καθορίζει τα *string* χαρακτήρων για τις ελλειπούσες τιμές (προεπιλεγμένη τιμή `NA`)

Για περισσότερες πληροφορίες, δείτε τη σχετική σελίδα βοήθειας (`?read.table`).

Παραλλαγές της *read.table*

Ορισμένες πολύ συνηθισμένες μορφοποιήσεις αρχείων δεδομένων μπορούν να διαβαστούν από παραλλαγές της `read.table`,

- η εντολή **`read.csv`** καλεί τη **`read.table`** με κατάλληλες προεπιλογές για την ανάγνωση αρχείων που τα δεδομένα χωρίζονται με κόμμα, όπου για υποδιαστολή χρησιμοποιείται η ”.”,
- η εντολή **`read.delim`** καλεί τη **`read.table`** με κατάλληλες προεπιλογές για την ανάγνωση αρχείων που τα δεδομένα διακρίνονται με το χαρακτήρα *tab* (στηλοθέτη), όπου για υποδιαστολή χρησιμοποιείται η ”.”.
- Άλλες παραλλαγές, **`read.csv2`**, **`read.delim2`**, **`read.fwf`**.

Εφαρμογή VI. Gladstone Survey

Το αρχείο ”gladstone.txt” περιέχει το σύνολο των δεδομένων της μελέτης του Gladstone. Ανοίξτε τη R και απαντήστε τα παρακάτω,

1. Εισάγετε τα δεδομένα σε ένα *data frame*.
2. Δημιουργείστε ένα *factor* με τιμές ’Male/Female’ στη βάση της μεταβλητής `Gender` του *data frame*.
3. Αφαιρέστε από το *data frame* τη μεταβλητή `Gender`. Προσθέστε το *factor* που δημιουργήσατε στο *data frame* με όνομα `Sex`.
4. Ποιο είναι το φύλο, το κρανιακό μέγεθος και το εγκεφαλικό βάρος της παρατήρησης 58. Των παρατηρήσεων 133 έως 136.
5. Πόσοι είναι οι άνδρες και πόσες οι γυναίκες της έρευνας.

Αναφορές

Adler, J. (2010). *R in a nutshell*. O' Reilly Media, Inc.

Gladstone, R. J. (1905). A study of the relations of the brain to the size of the head. *Biometrika*, 4(1/2):105–123.

Teetor, P. (2011). *R Cookbook: Proven recipes for data analysis, statistics, and graphics*. O' Reilly Media, Inc.

Venables, W. N. and Ripley, B. D. (2013). *Modern applied statistics with S-PLUS*. Springer Science & Business Media.

Venables, W. N. and Smith, D. M. (2009). *An introduction to R*. Network Theory Ltd. URL `www.r-project.org/manuals.html`.