Κεφάλαιο 1

Εισαγωγή στην R

Ο κύριος σκοπός αυτών των σημειώσεων είναι η εισαγωγή στην στατιστική γλώσσα προγραμματισμού R. Η γλώσσα R είναι ελεύθερα διαθέσιμη από το διαδίκτυο και η υποστήριξή της γίνεται μέσω της εθελοντικής συνεισφοράς πολλών ανθρώπων ανά τον κόσμο, οι οποίοι είναι και υπεύθυνοι για την ανάπτυξή της. Η ιστοσελίδα `http://www.r-project.org/` περιέχει περαιτέρω πληροφορίες καθώς και συνδέσμους για τα σχετικά προγράμματα που αφορούν την αποθήκευση και εκτέλεση του προγράμματος σε διάφορα λειτουργικά συστήματα. Σημειωτέον, ότι η R μπορεί να τρέξει σε περιβάλλον Linux, Mac OS και Windows.

Όπως θα δούμε, η R είναι μία γλώσσα προγραμματισμού που χρησιμεύει κατεξοχήν στην επεξηγηματική ανάλυση δεδομένων καθώς και στην εφαρμογή διαφόρων στατιστικών μοντέλων. Μπορεί να χρησιμοποιηθεί είτε με κατευθείαν εντολές είτε με προγράμματα τα οποία μπορούν να αναπτυχθούν και να δοθούν για εκτέλεση. Σε αυτές τις σημειώσεις θα μάθουμε πώς να προγραμματίζουμε στην R καθώς και το πώς κατασκευάζονται ειδικές *συναρτήσεις* (functions) οι οποίες χρησιμεύουν για ανάπτυξη ιδίων προγραμμάτων.

Περιληπτικά, θα δούμε τα παρακάτω

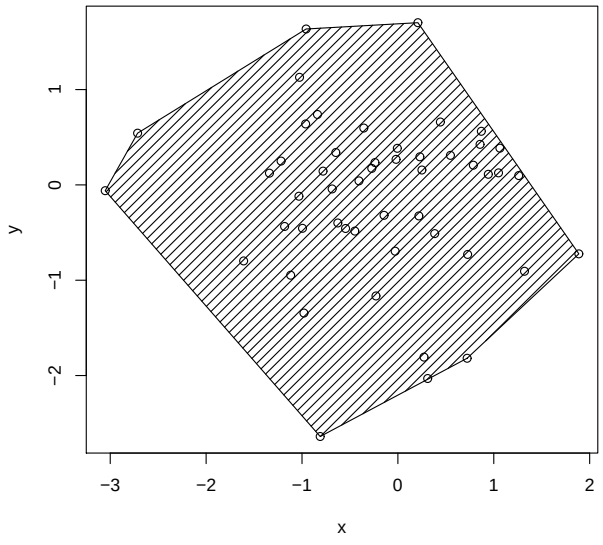
- Γενικές έννοιες που αφορούν την R.
- Πώς χρησιμοποιείται η R στην ανάλυση δεδομένων.
- Προγραμματισμός και ανάπτυξη στην R.

1.1 Μία Εισαγωγική Περίοδος

Οι παρακάτω εντολές θα δώσουν μία πρώτη γεύση από το τι μπορεί να κάνει η R. Καταρχάς μπορεί να μην γίνονται κατανοητές οι εντολές αυτές, αλλά τυχόν σύγχυση θα φύγει όταν προχωρήσουμε στα παρακάτω κεφάλαια.

Πρώτο Παράδειγμα

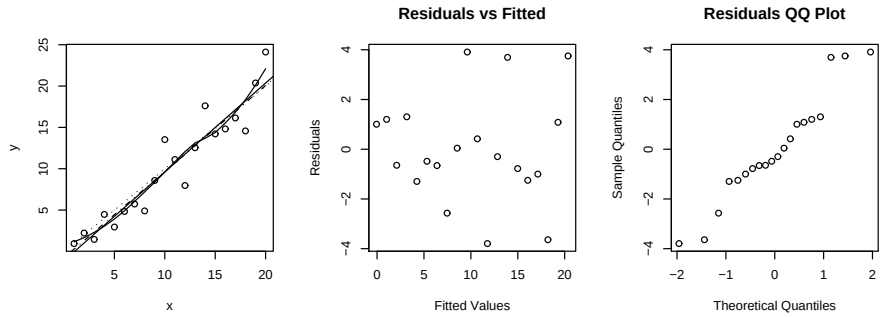
`x <- rnorm(50)`	Προσομοίωση δύο τυχαίων τυπικών κανονικών
`y <- rnorm(x)`	διανυσμάτων x και y.
`hull <- chull(x,y)`	Υπολογισμός κυρτού περιβλήματος των δεδομένων
`plot(x,y)`	Κατασκευάζει τη γραφική παράσταση των σημείων στο επίπεδο
`polygon(x[hull], y[hull],dens=15)`	και σημειώνει το κυρτό τους περίβλημα.
`objects()`	Βλέπει ποια αντικείμενα της R υπάρχουν μέσα στο αρχείο Data.
`rm(x,y)`	Αφαιρεί τα αντικείμενα x και y.



Σχήμα 1.1: Πρώτο παράδειγμα.

Δεύτερο Παράδειγμα

`x <- 1:20`	Δημιουργεί το διάνυσμα $x = (1, 2, ..., 20)$.
`w <- 1+sqrt(x)/2`	Δημιουργεί το διάνυσμα των βαρών των τυπικών αποκλίσεων.
`dummy <- data.frame(x=x,`	Κατασκευάζει ένα *πλαίσιο δεδομένων* με 2 στήλες
`y=x+rnorm(x)*w)`	x και y και το παρουσιάζει.
`dummy`	
`objects()`	Βλέπει ποια αντικείμενα της R
	υπάρχουν μέσα στο αρχείο Data.
`fm <- lm(y~x, data=dummy)`	Εφαρμόζει απλή γραμμική παλινδρόμιση της y ως προς x
`summary(fm)`	και παρουσιάζει τα αποτελέσματα
`fm1 <- lm(y~x, data=dummy,`	Εφαρμόζει σταθμισμένη παλινδρόμιση.
`weight=1/w^2)`	
`lrf <- loess(y~x, data=dummy)`	Κάνει απαραμετρική παλινδρόμιση.
`attach(dummy)`	Άμεσα προσβάσιμες στήλες πλαισίου δεδομένων.
`plot(x,y)`	Κάνει την γραφική παράσταση του x συναρτήσει του y.
`lines(x, fitted(lrf))`	Προσθέτει στο γράφημα το μοντέλο από
	την απαραμετρική παλινδρόμηση.
`abline(0,1,lty=3)`	Προσθέτει στο γράφημα την πραγματική
	γραμμή παλινδρόμισης.
`abline(coef(fm))`	Η γραμμή από την απλή γραμμική παλινδρόμηση.
`abline(coef(fm1), lty=4)`	Η γραμμή από την σταθμική παλινδρόμηση.
	Οποιαδήποτε στιγμή μπορείτε να τυπώσετε αντίγραφο
	της γραφικής παράστασης πατώντας στο παράθυρο
	`Graph` και επιλέγοντας το `Print`.
`detach()`	Αφαιρεί τις στηλες του πλαισίου δεδομένων
	απο τη λίστα αντικειμένων.
`plot(fitted(fm), resid(fm),`	Γραφική παράσταση των υπολοίπων
`xlab="Fitted Values",`	για έλεγχο της ετεροσκεδαστικότητας.
`ylab="Residuals", main=`	
`"Residuals vs Fitted")`	
`qqnorm(resid(fm), main=`	`QQ plot` των υπολοίπων.
`"Residuals QQ Plot")`	
`rm(fm,fm1,lrf,x,dummy)`	

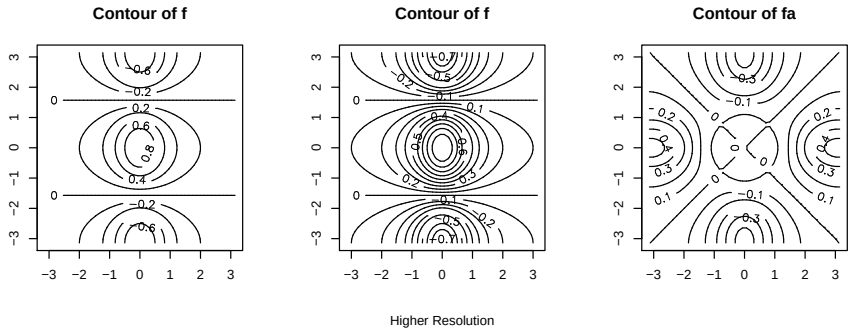


Σχήμα 1.2: Δεύτερο παράδειγμα.

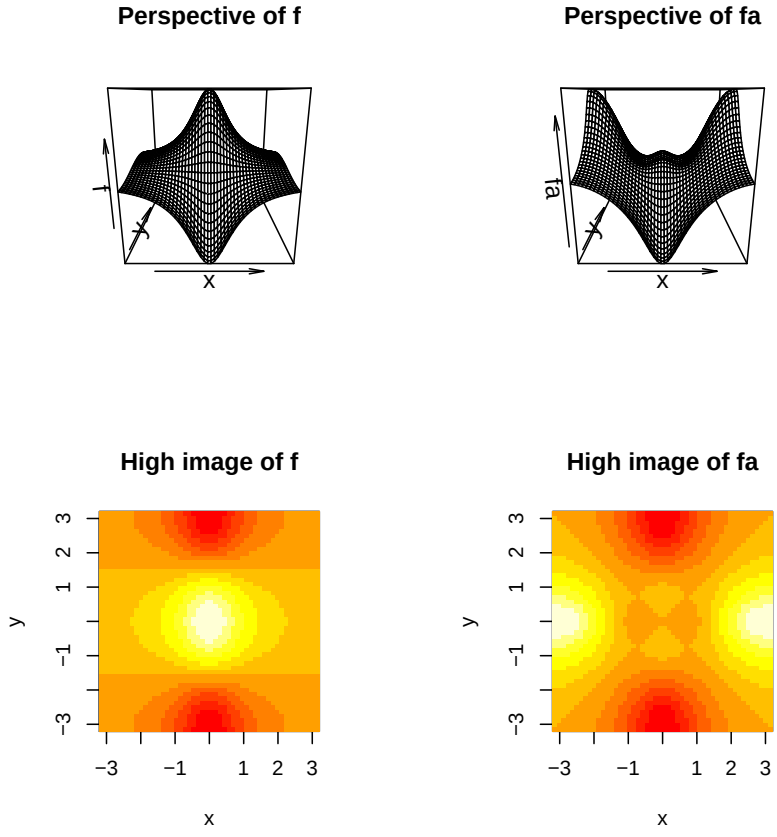
Τρίτο Παράδειγμα

Γραφικές δυνατότητες της R: διάγραμμα ισουψών και 3-διάστατες γραφικές παραστάσεις.

`x <- seq(-pi,pi,length=50)`	x είναι διάνυσμα με 50 ισαπέχοντες τιμές στο $(-\pi, \pi)$.
`y <- x`	Το ίδιο με το x.
`f <- outer(x,y,`	Ορίζουμε ένα πίνακα f του οποίου οι γραμμές
`function(x,y)`	και οι στήλες έχουν δείκτες x και y αντίστοιχα,
`cos(y)/(1+x^2))`	και ικανοποιούν τη εξίσωση $\cos(y)/(1+x^2)$.
`oldpar <- par()`	Φυλάει τις εξ ορισμού γραφικές παραμέτρους.
`par(pty="s")`	Καθορίζει την περιοχή του γραφήματος σε *τετράγωνο*.
`contour(x,y,f)`	Κάνει το διάγραμμα ισουψών της f.
`contour(x,y,f,`	Προσθέτει στο διάγραμμα πιο ψηλή ευκρίνεια.
`nlevels=15, add=T)`	
`fa <- (f-t(f))/2`	fa είναι το ασσύμετρο κομμάτι της f.
`contour(x,y, fa, nlevels=15)`	Δημιουργεί το διάγραμμα ισουψών της fa.
`par(oldpar)`	Επαναφέρει τις εξ ορισμού γραφικές παραμέτρους.
`persp(x,y,f)`	Δημιουργεί προοπτική απεικόνιση και υψηλού
`persp(x,y,fa)`	επιπέδου γραφική παράσταση.
`image(x,y, f)`	
`image(x,y,fa)`	
`objects(); rm(x,y,f,fa)`	Αφαιρεί τα υπάρχοντα αντικείμενα.
`q()`	Έξοδος από R.



Σχήμα 1.3: Τρίτο παράδειγμα (I).



Σχήμα 1.4: Τρίτο παράδειγμα (II).

1.2 Βασικές έννοιες

Η R εφαρμόζει μια διάλεκτο της γλώσσας S η οποία ειναι μια διερμηνέας γλώσσα προγραμματισμού. Αυτό σημαίνει ότι οι εντολές διαβάζονται και μετά εκτελούνται αμέσως. Αντίθετα, η C και η Fortran είναι μεταγλωτίστριες γλώσσες προγραμματισμού στις οποίες ολοκληρωμένα προγράμματα μεταφράζονται με τη βοήθεια ενός μεταγλωτιστή στην κατάλληλη γλώσσα μηχανής. Το μεγάλο πλεονέκτημα των διερμηνέων γλωσσών προγραμματισμού είναι ότι επιτρέπουν σταδιακή ανάπτυξη. Με άλλα λόγια, μια συνάρτηση μπορεί να δημιουργηθεί, να εκτελεσθεί και μετά να δημιουργηθεί μια καινούργια συνάρτηση η οποία καλεί την πρηγούμενη κ.ο.κ. Σημειώστε όμως ότι μεταγλωτισμένος κώδικας τρέχει πιο γρήγορα και χρειάζεται λιγότερη μνήμη από το διερμηνευμένο κώδικα.

Η αλληλεπίδραση με την R επιτυγχάνεται πληκτρολογώντας εκφράσεις, τις οποίες ο διερμηνέας αξιολογεί και μετά τις εκτελεί. Για παράδειγμα

```
> sqrt
function(x)
x^0.5
> sqrt(2)
[1] 1.414214
```

ή

```
log
function(x, base = 2.71828182845905)
{
    y <- .Internal(log(x), "do_math", T, 106)
    if(missing(base))
        y
    else y/.Internal(log(base), "do_math", T, 106)
}
> log(10)
[1] 2.302585
```

Αξίζει να σημειωθεί ότι η R είναι ευαίσθητη στα κεφαλαία γράμματα. Αυτό σημαίνει οτι το `x` και το `X` είναι διαφορετικά αντικείμενα. Μια συνάρτηση καλείται συνήθως γράφοντας το όνομα της ακολουθούμενο από μια λίστα ορισμάτων. Για παράδειγμα

```
> plot(fdeaths)
> mean(fdeaths)
[1] 560.6806
```

Οι μαθηματικές πράξεις είναι συναρτήσεις με δύο ορίσματα τα οποία έχουν ειδικό κάλεσμα. Π.χ.

```
> 2+5
[1] 7
> 3*6.8
[1] 20.4
> 12.6/6
[1] 2.1
```

Ένα από τα σύμβολα που χρησιμοποιείται πιο συχνά είναι το σύμβολο εγχώρησης <-, το οποίο καταχωρεί στις μεταβλητές συγκεκριμένες τιμές (π.χ. αριθμό, διάνυσμα, πίνακα, πλαίσιο δεδομένων κ.α.) ή αποτελέσματα πράξεων.

```
test <- 4
> test
[1] 4
```

Ακόμη ένα πολύ συνηθισμένο σύμβολο στην R είναι το σύμβολο δείκτη [, το οποίο χρησιμοποιείται για να εξάγει υποσύνολα από ένα αντικείμενο, π.χ.

```
> letters
 [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o"
[16] "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"
> letters[3]
 [1] "c"
> letters[-3]
 [1] "a" "b" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p"
[16] "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"
```

Επίσης μπορεί να υπολογιστεί η λογική τιμή μιας πρότασης, όπως

```
> j <- 1:26
> j<5
 [1]  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
[11] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[21] FALSE FALSE FALSE FALSE FALSE FALSE
```

```
> letters[j<5]
[1] "a" "b" "c" "d"
```

Η τοποθέτηση δεικτών είναι πολύ σημαντική στην αποτελεσματική χρήση της R γιατί δίνει έμφαση στο να επεξεργάζεται αντικείμενα δεδομένων σαν ολοκληρωμένες οντότητες, παρά σαν μια συλλογή από ξεχωριστές παρατηρήσεις.

Σαν τελευταία εισαγωγική σημείωση, τονίζεται ότι κάθε έκφραση της R ερμηνεύεται από τον αξιολογητή και επιστρέφει ένα *αντικείμενο δεδομένων*. Τα αντικείμενα δεδομένων έχουν τις παρακάτω μορφές :

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

Οι μορφές είναι γραμμένες από αυτήν που παρέχει την λιγότερη πληροφορία εώς εκείνη που παρέχει την περισσότερη πληροφορία. Όταν είναι ανάγκη να συνδυάσεις διαφορετικές μορφές, τότε η R χρησιμοποιεί εκείνη με την περισσότερη πληροφορία. Το επόμενο παράδειγμα επεξηγεί αυτό το σκεπτικό:

```
> -3.6
[1] -3.6
> "Munich"
[1] "Munich"
> c(T, F, T)
[1] T F T
> c(-2, pi, 2)
[1] -2.000000  3.141593  2.000000
> c(T, pi, F)
[1] 1.000000 3.141593 0.000000
> c(T, pi, "Munich")
[1] "TRUE"             "3.14159265358979" "Munich"
> mode(c(T, pi, "Munich"))
[1] "character"
```