Εισαγωγή στην R
Πρόχειρες Σημειώσεις

Κωνσταντίνος Φωκιανός & Χαράλαμπος Χαραλάμπους
Τμήμα Μαθηματικών & Στατιστικής
Πανεπιστήμιο Κύπρου

1η Έκδοση: Φεβρουάριος 2008
2η Έκδοση: Ιανουάριος 2010

Περιεχόμενα

I Στατιστικές Μέθοδοι στην R –I **11**

1 Εισαγωγή στην R **13**

1.1 Μία Εισαγωγική Περίοδος 14

1.2 Βασικές έννοιες 18

2 Αντικείμενα Δεδομένων **21**

2.1 Διανύσματα 21

2.2 Πίνακες 23

2.3 Πίνακες μεγαλύτερης διάστασης (Arrays) 27

2.4 Λίστες 28

2.5 Παράγοντες 29

2.6 Πλαίσια Δεδομένων (Data Frames) 31

3 Μαθηματικοί Υπολογισμοί στην R **37**

3.1 Αριθμητικές πράξεις και απλές συναρτήσεις 37

3.2 Πράξεις Διανυσμάτων και Πινάκων 41

3.3 Γραμμικό Σύστημα Εξισώσεων 43

3.4 Τυχαίοι Αριθμοί 43

3.5 Άλλες Χρήσιμες Συναρτήσεις 45

4 Γραφήματα **47**

4.1 Απλά Γραφήματα 47

4.2 Γραφικές Δυνατότητες 49

4.3 Είδη και Γραμμές Γραφικής Παράστασης 51

4.4 Προσθήκη Πληροφοριών σε Γράφημα 52

4.5 Γραφήματα Σε Μεγαλύτερες Διαστάσεις 57

5 Απλός Προγραμματισμός στην R **59**
5.1 Λογικοί Τελεστές και Τελεστές Σύγκρισης 59
5.2 Χρησιμοποιώντας Υποσύνολα των Δεδομένων 60
5.3 Κατασκευή Συναρτήσεων 63

6 Προσομοίωση **67**
6.1 Ο Ασθενής Νόμος των Μεγάλων Αριθμών 67
6.2 Κεντρικό Οριακό Θεώρημα 70
6.3 Προσεγγίσεις της Διωνυμικής Κατανομής 72
6.4 Monte Carlo Ολοκλήρωση 75
6.5 Βελόνα του Buffon 77
6.6 Εμπειρική Σύγκριση Εκτιμητριών 78

7 Στατιστική Συμπερασματολογία **83**
7.1 Περιγραφική Στατιστική 83
7.2 Συμπερασματολογία για Ένα Δείγμα 86
7.3 Συμερασματολογία για Δυο Δείγματα 89
7.4 Συμπερασματολογία για Δείγματα Κατά Ζεύγη 91
7.5 Έλεγχος Καλής Προσαρμογής 93
7.6 Έλεγχος Υποθέσεων για Ποσοστά 95
7.7 Πίνακες Συνάφειας 98
7.8 Παράδειγμα 100

8 Γραμμική Παλινδρόμηση **109**
8.1 Γραμμικά Μοντέλα στην R 109
8.2 Πολλαπλή Γραμμική Παλινδρόμηση 110

9 Ανάλυση της Διακύμανσης **125**
9.1 Ανάλυση Διακύμανσης κατά ένα Παράγοντα 125
9.2 Πολλαπλές Συγκρίσεις 132

10 Λογιστική Παλινδρόμηση **135**
10.1 Περιγραφή των Δεδομένων 135
10.2 Λογιστική Παλινδρόμηση 136
10.3 Ανάλυση στην R 136
10.4 Μοντέλο Probit 140

11 Τεχνικές Αναδειγματοληψίας **147**
11.1 Μέθοδος Jackknife 147
11.2 Μέθοδος Bootstrap 151
11.3 Εκτίμηση Συντελεστή Συσχέτισης 153
11.4 Συντελεστές Παλινδρόμησης 155

12 Ασκήσεις Μέρους I **159**

II Στατιστικές Μέθοδοι στην R –II **163**

13 Ειδικά Γραφήματα **165**
13.1 Γραφήματα Trellis 165

14 Μέθοδος Newton-Raphson **173**
14.1 Παράδειγμα 173

15 Ανάλυση της Συνδιακύμανσης **181**
15.1 Ανάλυση της Συνδιακύμανσης 181

16 Εκτίμηση Μη-Γραμμικών Μοντέλων **185**
16.1 Περιγραφή των Δεδομένων 185
16.2 Ανάλυση με Μη-Γραμμικό Μοντέλο 186

17 Poisson Παλινδρόμηση και Λογαριθμικά Γραμμικά Μοντέλα **191**
17.1 Poisson Παλινδρόμηση 191
17.2 Παράδειγμα 193

18 Μη Παραμετρική Παλινδρόμηση **199**
18.1 Τοπική Πολυωνυμική Παλινδρόμηση 200
18.2 Εξομαλυντές Splines 202
18.3 Αθροιστική Απαραμετρική Παλινδρόμηση 203

19 Ανάλυση Επιβίωσης **207**
19.1 Συνάρτηση Επιβίωσης 207
19.2 Συνάρτηση Κινδύνου 208
19.3 Μοντέλο αναλόγων συναρτήσεων κινδύνου 208
19.4 Παράδειγμα 209

20 Ανάλυση σε Κύριες Συνιστώσες και Διαχωριστική Ανάλυση 219
20.1 Ανάλυση σε Κύριες Συνιστώσες 219
20.2 Διαχωριστική Ανάλυση 227

21 Ανάλυση Κατά Συστάδες στην R 233
21.1 Εισαγωγή 233
21.2 Ιεραρχική Ανάλυση κατά Συστάδες 237
21.3 Μεθοδολογία K-means (MacQueen) 246
21.4 Partitioning Around Medoids (PAM) 248
21.5 Self Organizing Maps (SOM) 250
21.6 Fuzzy Analysis Clustering (Fanny) 253
21.7 Παράδειγμα ανάλυσης δεδομένων 257
Βιβλιογραφία 264

22 Ανάλυση Χρονοσειρών 265
22.1 Ανάλυση Χρονοσειρών 265
22.2 Παράδειγμα 266

23 Παραδείγματα Μεθόδων Ε-Μ Αλγόριθμου 273
23.1 Πολυωνυμική Κατανομή 274
23.2 Έλεγχος Επιβίωσης 277
23.3 Μοντέλο Πεπερασμένης Μίξης Κανονικών 280

Πρόλογος 2ης Έκδοσης

Στην παρούσα έκδοση των Πρόχειρων Σημειώσεων στην R συμπεριλαμβάνονται θέματα τα οποία ανήκουν σε πιο εξειδικευμένα θέματα Στατιστικής και η διδασκαλία τους γίνεται σε επίπεδο μεταπτυχιακού μαθήματος. Πιο εδικά, έχουμε συμπεριλάβει μεθόδους ανάλυσης συνδιακύμανσης, Poisson λογαριθμικά μοντέλα, μη παραμετρική παλινδρόμηση, ανάλυση επιβίωσης, μέθοδοι ανάλυσης χρονοσειρών καθώς και μεθόδους ανάλυσης πολυδιάστατων δεδομένων. Επίσης συμπεριλαμβάνονται ειδικές γραφικές παραστάσεις που μπορεί να κατασκευάσει η R καθώς και κάποια παραδείγματα προγραμματισμού με εφαρμογές την μέθοδο Newton–Raphson καθώς και τον αλγόριθμο EM. Ευελπιστούμε ότι η παρούσα έκδοση των σημειώσεων θα είναι χρήσιμη για την εκμάθηση ποικίλων στατιστικών μεθόδων καθώς και για την καλύτερη κατανόηση του λογισμικού R.

Λευκωσία, Ιανουάριος 2010

Κωνσταντίνος Φωκιανός Χαράλαμπος Χαραλάμπους

Πρόλογος 1ης Έκδοσης

Οι σημειώσεις αυτές αποτελούν μία εισαγωγή στην στατιστική γλώσσα προγραμματισμού R η οποία αναπτύσσεται ραγδαία τα τελευταία χρόνια. Η R είναι ελεύθερα διαθέσιμη στην ιστοσελίδα `http://www.r-project.org/` και στηρίζεται στην ανάπτυξη προγραμμάτων μέσω πακέτων (packages) τα οποία διατίθενται πάλι ελεύθερα από χρήστες ανά τον κόσμο. Συνεπώς, είναι λογικό να γραφεί και ένα σύγγραμμα στα ελληνικά έτσι ώστε η R να γίνει ευρύτερα γνωστή με ελεύθερο τρόπο. Το βιβλίο απευθύνεται πρωτίστως σε φοιτητές Μαθηματικών τμημάτων με κατεύθυνση Στατιστική αλλά μπορεί να χρησιμεύσει και σε φοιτητές άλλων σχολών που το αντικείμενό τους συνάδει με την Στατιστική. Απαραίτητη προϋπόθεση για να διαβάσει ο αναγνώστης τις σημειώσεις είναι η επιτυχής ολοκλήρωση εισαγωγικών μαθημάτων Πιθανοτήτων και Στατιστικής.

Η δομή των σημειώσεων έχει ως εξής. Τα κεφάλαια 1–5 εισάγουν το φοιτητή στις βασικές έννοιες της R, το κεφάλαιο 6 συζητά εφαρμογές στατιστικής θεωρίας μέσω προσομοιώσεων ενώ τα κεφάλαια 7–10 είναι αφιερωμένα σε βασικές στατιστικές μεθόδους ανάλυσης δεδομένων. Το κεφάλαιο 11 συζητά μεθόδους αναδειγματοληψίας ενώ η παρουσίαση κλείνει με παραδείγματα και ασκήσεις. Τα μαθήματα αυτά έχουν διδαχτεί σε προπτυχιακούς και μεταπτυχιακούς φοιτητές του τμήματος Μαθηματικών και Στατιστικής του Πανεπιστημίου Κύπρου και η εμπειρία του πρώτου συγγραφέα είναι ότι οι όλα τα κεφάλαια μπορούν να γίνουν σε δεκατρία δίωρα εργαστήρια έχοντας τους φοιτητές να αλληλεπιδρούν με τον ηλεκτρονικό υπολογιστή. Σε αυτό το στάδιο δεν γίνεται προσπάθεια να γίνουν κατανοητές οι μαθηματικές έννοιες αλλά να μπορεί ο αναγνώστης να εφαρμόζει τις πιο απλές στατιστικές μεθόδους.

Φυσικά σε κάθε τέτοιο επιχείρημα υπάρχουν ελλείψεις και λάθη. Τα προγράμματα που παρουσιάζονται εδώ δεν έχουν κανένα πρόβλημα αφού έχουν δοκιμαστεί σε διάφορες πλατφόρμες. Οι ελλείψεις, κατά την ταπεινή μας γνώμη, είναι η απουσία μη παραμετρικής εκτιμήτριας συνάρτησης πυκνότητας πιθανότη-

τας, μη παραμετρικής παλινδρόμηση και ανάλυση της διακύμανσης και επιλογή και εκτίμηση μοντέλων. Ευελπιστώντας στην κατανόηση του αναγνωστικού κοινού, πιστεύουμε στην εποικοδομητική κριτική η οποία θα βελτιστοποιήσει τις ανά χείρας σημειώσεις και θα τις επιτρέψει να έχουν 2η, 3η έκδοση κ.ο.κ. Για τον αναγνώστη που ενδιαφέρεται να μάθει παραπάνω και να εξερευνήσει την R εις βάθος του προτείνουμε τα παρακάτω εγχειρίδια:

1. Richard A. Becker, John M. Chambers, and Allan R. Wilks. The New S Language. Chapman & Hall, London, 1988.
2. John M. Chambers and Trevor J. Hastie. Statistical Models in S. Chapman & Hall, London, 1992.
3. William N. Venables and Brian D. Ripley. Modern Applied Statistics with S. Fourth Edition. Springer, New York, 2002.
4. William N. Venables and Brian D. Ripley. S Programming. Springer, New York, 2000.
5. Frank E. Harrell. Regression Modeling Strategies, with Applications to Linear Models, Survival Analysis and Logistic Regression.
6. John Fox. An R and S-Plus Companion to Applied Regression. Sage Publications, Thousand Oaks, CA, USA, 2002.
7. Julian J. Faraway. Linear Models with R. Chapman & Hall/CRC, Boca Raton, FL, 2004.

Φυσικά υπάρχουν πλήθος άλλα συγγράμματα και ο ενδιαφερόμενος αναγνώστης μπορεί να βρει περαιτέρω πληροφορίες στην ιστοσελίδα της R.

Συμπερασματικά, κανείς δεν μπορεί, την γνώμη του πρώτου συγγραφέα, να ισχυριστεί ότι με το διάβασμα μερικών σημειώσεων καθώς και με λίγη εξάσκηση έμαθε να προγραμματίζει, σε οποιαδήποτε γλώσσα προγραμματισμού. Για να γίνει αυτό χρειάζεται συνεχής ενασχόληση, κριτική σκέψη και διαρκής εξερεύνηση.

Λευκωσία, Ιανουάριος 2008

Κωνσταντίνος Φωκιανός Χαράλαμπος Χαραλάμπους

Μέρος Ι

Στατιστικές Μέθοδοι στην R

–Ι

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

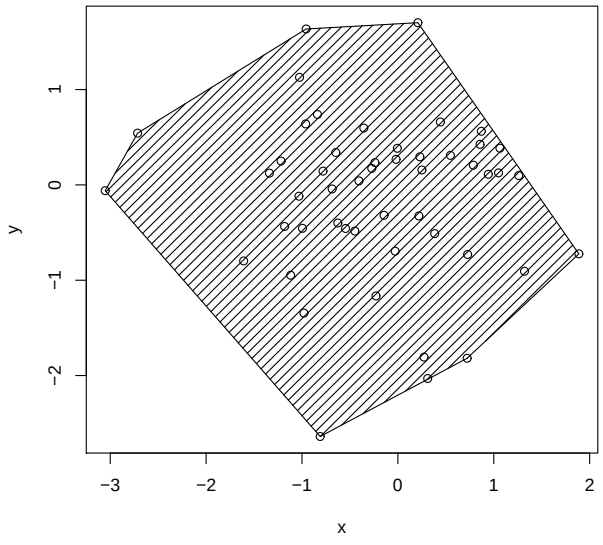
- Γενικές έννοιες που αφορούν την R.
- Πώς χρησιμοποιείται η R στην ανάλυση δεδομένων.
- Προγραμματισμός και ανάπτυξη στην R.

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

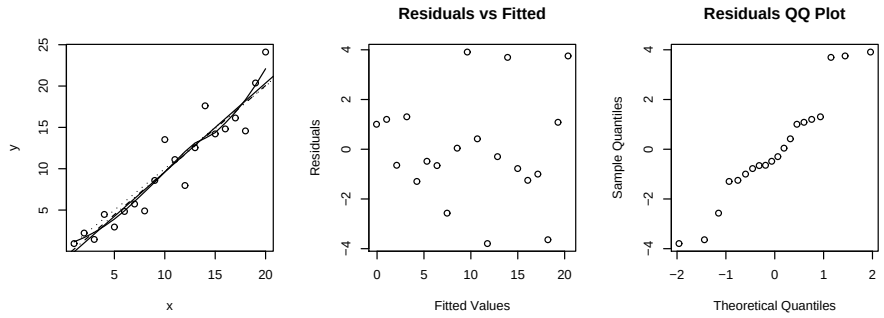
`x <- rnorm(50)`	Προσομοίωση δύο τυχαίων τυπικών κανονικών
`y <- rnorm(x)`	διανυσμάτων x και y.
`hull <- chull(x,y)`	Υπολογισμός κυρτού περιβλήματος των δεδομένων
`plot(x,y)`	Κατασκευάζει τη γραφική παράσταση των σημείων στο επίπεδο
`polygon(x[hull], y[hull],dens=15)`	και σημειώνει το κυρτό τους περίβλημα.
`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)`	

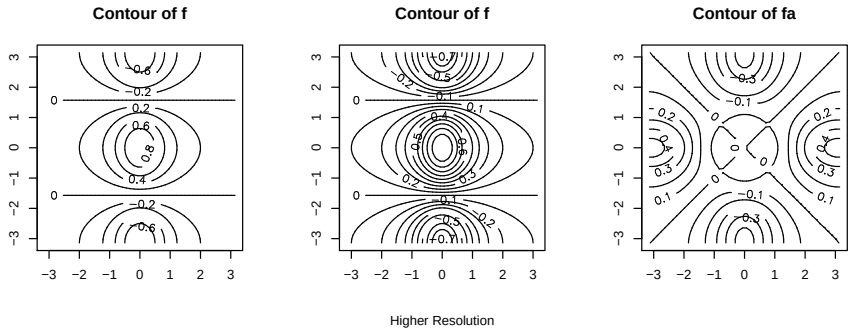


Σχήμα 1.2: Δεύτερο παράδειγμα.

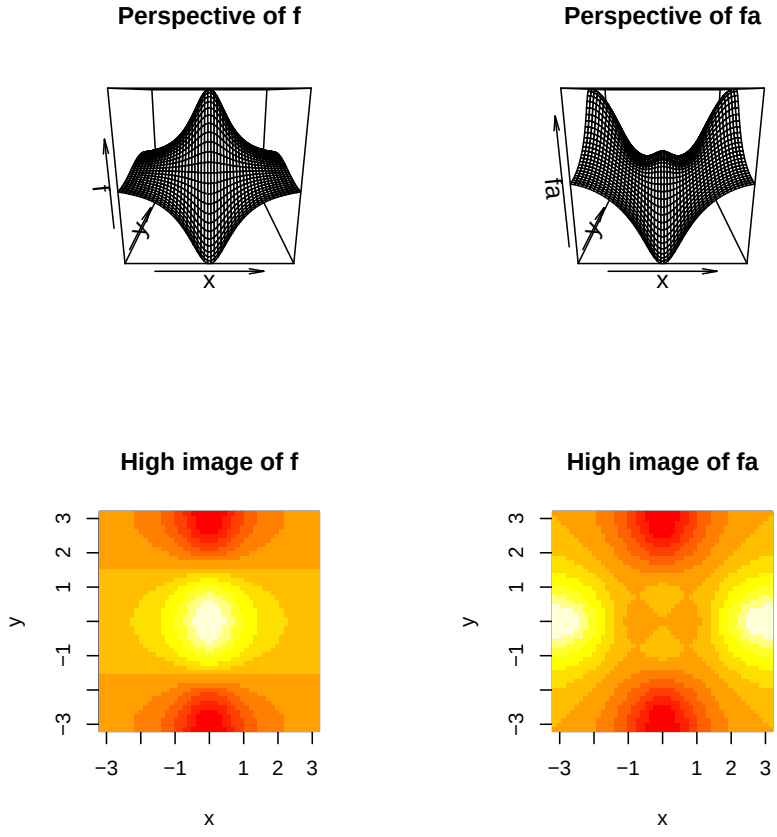
Τρίτο Παράδειγμα

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"
```

Κεφάλαιο 2

Αντικείμενα Δεδομένων

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

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

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

2.1 Διανύσματα

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

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

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

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

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

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

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

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

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

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

2.2 Πίνακες

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

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

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

```
[2,]    1    2    3    4
[3,]    1    2    3    4
> matr2 <- seq(-2,2,length=25)
> matr2
 [1] -2.0000000 -1.8333333 -1.6666667 -1.5000000 -1.3333333 -1.1666667 -1.0000000
 [8] -0.8333333 -0.6666667 -0.5000000 -0.3333333 -0.1666667  0.0000000  0.1666667
[15]  0.3333333  0.5000000  0.6666667  0.8333333  1.0000000  1.1666667  1.3333333
[22]  1.5000000  1.6666667  1.8333333  2.0000000
> dim(matr2) <- c(5,5)
> matr2
          [,1]       [,2]       [,3]      [,4]     [,5]
[1,] -2.000000 -1.1666667 -0.3333333 0.5000000 1.333333
[2,] -1.833333 -1.0000000 -0.1666667 0.6666667 1.500000
[3,] -1.666667 -0.8333333  0.0000000 0.8333333 1.666667
[4,] -1.500000 -0.6666667  0.1666667 1.0000000 1.833333
[5,] -1.333333 -0.5000000  0.3333333 1.1666667 2.000000
```

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

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

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

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

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

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

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

```
[2,]   -2   -1    0
[3,]    1    2    3
[4,]    4    5    6
> matrix(1:23, nrow=7, ncol=4)
     [,1] [,2] [,3] [,4]
[1,]    1    8   15   22
[2,]    2    9   16   23
[3,]    3   10   17    1
[4,]    4   11   18    2
[5,]    5   12   19    3
[6,]    6   13   20    4
[7,]    7   14   21    5
Warning messages:
  Replacement length not a multiple of number of
  elements to replace in: data[1:ll] <-old
>  matrix(1:23, nrow=7)
     [,1] [,2] [,3] [,4]
[1,]    1    8   15   22
[2,]    2    9   16   23
[3,]    3   10   17    1
[4,]    4   11   18    2
[5,]    5   12   19    3
[6,]    6   13   20    4
[7,]    7   14   21    5
Warning messages:
Replacement length not a multiple of number of
elements to replace in: data[1:ll] <-old
```

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

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

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

2.3 Πίνακες μεγαλύτερης διάστασης (Arrays)

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

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

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

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

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

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

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

2.4 Λίστες

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

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

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

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

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

2.5 Παράγοντες

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

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

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

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

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

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

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

```
> fact <- rnorm(10)
> fact1 <- cut(fact, breaks=c(-5,-1,1,2,4))
> fact1
 [1] (-1,1]  (-5,-1] (-5,-1] (-1,1]  (-1,1]  (-5,-1] (-1,1]  (-1,1]  (-5,-1]
[10] (-1,1]
Levels: (-5,-1] (-1,1] (1,2] (2,4]
> fact2 <- cut(fact, breaks=5)
```

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

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

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

2.6 Πλαίσια Δεδομένων (Data Frames)

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

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

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

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

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

Σε αυτό το σημείο θα εξεταστεί μόνο ο δεύτερος τρόπος.

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

```
 [1]  0.4452555  0.4751909  0.9357701  1.5757764 -1.0958323  2.1620200
 [7] -1.1306062 -0.4240480  0.2262562 -1.4675688 -0.5541264  1.0983463
[13]  1.3681229  0.2005744  0.5910446  0.8732281  0.3314982  0.8242927
[19]  0.7790229  0.5909648
> my.matrix<-matrix(rnorm(40),ncol=2)
> my.matrix
              [,1]        [,2]
 [1,]  0.650290036  1.52145790
 [2,] -1.059263140 -0.09996308
 [3,]  0.216042514  1.14707512
 [4,] -0.114622892  0.59689871
 [5,] -0.004433541  1.21214093
 [6,] -0.978986416 -0.60250469
 [7,] -0.609778169  0.68110679
 [8,]  0.138456517 -0.65849203
 [9,]  1.271366406 -2.23159156
[10,] -0.016984227  1.06334080
[11,] -0.135241342  0.05793721
[12,]  1.781934098 -0.22806050
[13,]  1.268863189 -2.14581499
[14,] -0.963995714 -1.38571628
[15,]  0.160374068 -0.18793847
[16,]  0.511570707  0.09455187
[17,] -1.126031052 -0.07339069
[18,]  0.394865156 -0.23565899
[19,] -0.238627823 -0.92214415
[20,] -0.755950206  0.86695967
> my.dataframe<-data.frame(my.logic,my.complex,my.numeric,my.matrix)
> my.dataframe
   my.logic            my.complex my.numeric           X1          X2
1      TRUE  0.1782105+0.9851267i  0.4452555  0.650290036  1.52145790
2      TRUE -2.6140989+0.3541577i  0.4751909 -1.059263140 -0.09996308
3     FALSE -0.0767501+0.2550082i  0.9357701  0.216042514  1.14707512
4     FALSE -0.7275827+0.3569999i  1.5757764 -0.114622892  0.59689871
5     FALSE -0.3280398+0.5163078i -1.0958323 -0.004433541  1.21214093
6     FALSE -0.7451964+0.8069675i  2.1620200 -0.978986416 -0.60250469
7      TRUE -0.6853192+0.5144991i -1.1306062 -0.609778169  0.68110679
```

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

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

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

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

Κάποιες άλλες εντολές οι οποίες είναι χρήσιμες είναι οι ακόλουθες :

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

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

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

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

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

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

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

Κεφάλαιο 3

Μαθηματικοί Υπολογισμοί στην R

Ένα μεγάλο μέρος της ανάλυσης δεδομένων απαιτεί διάφορους μαθηματικούς υπολογισμούς. Αυτό το κεφάλαιο εισαγάγει τον αναγνώστη στις διάφορες δυνατότητες που έχει η R για να γίνουν τέτοιοι υπολογισμοί. Οι υπολογιστικές δυνατότητες της R αρχίζουν από απλές πράξεις μέχρι και πολύπλοκους μαθηματικούς υπολογισμούς, όπως π.χ. η μεγιστοποίηση συναρτήσεων.

3.1 Αριθμητικές πράξεις και απλές συναρτήσεις

Οι βασικές αριθμητικές πράξεις γίνονται με τη βοήθεια των συμβόλων που βρίσκονται στον ακόλουθο πίνακα.

Σύμβολο	**Πράξη**
+	Πρόσθεση
−	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση
^	Ύψωση σε δύναμη
%/%	Ακέραια διαίρεση
%%	Υπόλοιπο διαίρεσης

Πίνακας 3.1: Βασικά αριθμητικά σύμβολα.

Ακολουθούν μερικά παραδείγματα τα οποία καταδεικνύουν πώς χρησιμοποιούνται τα σύμβολα των βασικών αριθμητικών πράξεων.

```
> 7+3
[1] 10
> 15-19
[1] -4
> 4*67
[1] 268
> 56/9
[1] 6.222222
> 2^6
[1] 64
> 27%/%3.4
[1] 7
> 27%%3.4
[1] 3.2
> 7*3.4+3.2
[1] 27
```

Το σύμβολο ^ είναι χρήσιμο όχι μόνο για ύψωση σε δύναμη αλλά και υπολογισμό ριζών.

```
> 16^(1/2)
[1] 4
> 2^(1/3)
[1] 1.259921
```

Αυτές οι εντολές χρησιμοποιούνται όχι μόνο με αριθμούς αλλά και με διανύσματα και πίνακες. Το επόμενο παράδειγμα δείχνει πως λειτουργούν σε αυτές τις περιπτώσεις.

```
> x <- c(1,4,7)
> y <- c(2,4,6,4,6,10)
> A <- matrix(c(2,3,4,5,6,7,1,2,3), nrow=3)
> A
     [,1] [,2] [,3]
[1,]    2    5    1
[2,]    3    6    2
```

```
[3,]    4    7    3
> B <- rbind(c(0,0,1), c(2,4,5), c(1,4,2))
> B
     [,1] [,2] [,3]
[1,]    0    0    1
[2,]    2    4    5
[3,]    1    4    2
> A*B
     [,1] [,2] [,3]
[1,]    0    0    1
[2,]    6   24   10
[3,]    4   28    6
> x+y
[1]  3  8 13  5 10 17
> A/y
          [,1] [,2] [,3]
[1,] 1.0000000 1.25  0.5
[2,] 0.7500000 1.00  0.5
[3,] 0.6666667 0.70  0.5
Warning messages:
  Length of longer object is not a multiple
  of the length of the shorter object in: A/y
```

Οι περισσότεροι υπολογισμοί με διανύσματα και πίνακες γίνονται *κατά στοιχείο*, υποθέτωντας ότι οι πίνακες έχουν τις ίδιες διαστάσεις. Στις πράξεις με διανύσματα, αν το ένα διάνυσμα είναι μικρότερης διάστασης από το άλλο, τότε τα στοιχεία του μικρότερου διανύσματος επαναλαμβάνονται κυκλικά έτσι ώστε τα δύο διανύσματα να έχουν στο τέλος ίσες διαστάσεις. Μαθηματικοί υπολογισμοί μεταξύ διανυσμάτων και πινάκων δεν έχουν συνήθως τα αναμενόμενα αποτελέσματα και θα πρέπει να χρησιμοποιούνται με μεγάλη προσοχή.

Ακολουθούν μερίκες γνωστές συναρτήσεις οι οποίες είναι εγκατεστημένες μέσα στην R και μπορούν να κληθούν ανά πάσα στιγμή από τον χρήστη. Οι συναρτήσεις αυτές υπολογίζονται κατά στοιχείο.

Συνάρτηση	Πράξη
sqrt()	Τετραγωνική ρίζα
abs()	Απόλυτη τιμή
floor()	Προηγούμενος ακέραιος
ceiling()	Επόμενος ακέραιος
sin()	Ημίτονο
cos()	Συνημίτονο
tan()	Εφαπτωμένη
asin()	Τόξο ημιτόνου
acos()	Τόξο συνημιτόνου
atan()	Τόξο εφαπτωμένης
exp()	Εκθετική συνάρτηση
log()	Λογάριθμος
log10()	Λογάριθμος με βάση το 10
gamma()	Συνάρτηση Γάμμα
lgamma()	Φυσικός λογάριθμος της απόλυτης τιμής της συνάρτηση Γάμμα

Πίνακας 3.2: Αριθμητικές συναρτήσεις.

Τα επόμενα παραδείγματα χρησιμοποιούν μερικές από αυτές τις συναρτήσεις.

```
> abs(-10.56)
[1] 10.56
> floor(5.6)
[1] 5
> ceiling(5.6)
[1] 6
> log(x)
[1] 0.000000 1.386294 1.945910
> log(x, base=2)   #logaritm to base 2
[1] 0.000000 2.000000 2.807355
> cos(A)
           [,1]      [,2]       [,3]
[1,] -0.4161468 0.2836622  0.5403023
[2,] -0.9899925 0.9601703 -0.4161468
[3,] -0.6536436 0.7539023 -0.9899925
> atan(A)
```

```
         [,1]     [,2]      [,3]
[1,] 1.107149 1.373401 0.7853982
[2,] 1.249046 1.405648 1.1071487
[3,] 1.325818 1.428899 1.2490458
> exp(y)
[1]     7.389056    54.598150   403.428793    54.598150   403.428793 22026.465795
```

3.2 Πράξεις Διανυσμάτων και Πινάκων

Όπως αναφέρθηκε πιο πάνω, στις περιπτώσεις των διανυσμάτων οι διάφορες αριθμητικές πράξεις εφαρμόζονται σε κάθε στοιχείο τους. Σε αυτό το σημείο θα γίνει αναφορά στο πώς εκτελούνται διάφοροι υπολογισμοί με διανύσματα ή πίνακες. Ο επόμενος πίνακας δίνει σύμβολα και συναρτήσεις για αυτές τις πράξεις.

Σύμβολα - Συνάρτηση	**Πράξη**
`%*%`	Εσωτερικό γινόμενο διανυσμάτων ή πολλαπλασιασμός πινάκων
`t()`	Ανάστροφος πίνακα
`solve()`	Αντίστροφος πίνακα (αν υπάρχει)
`diag()`	Εξαγωγή της διαγωνίου αλλά και κατασκευή διαγώνιου πίνακα
`eigen()`	Ιδιοτιμές και ιδιοδιανύσματα πίνακα

Πίνακας 3.3: Πράξεις διανυσμάτων και πινάκων.

Ακολουθούν μερικά παραδείγματα.

```
> A%*%B   #matrix multiplication
     [,1] [,2] [,3]
[1,]   11   24   29
[2,]   14   32   37
[3,]   17   40   45
> z <- c(2,3,1)
> z%*%x   #vector dot product
     [,1]
[1,]   21
> t(A)    # transpose of a matrix
     [,1] [,2] [,3]
```

```
[1,]    2    3    4
[2,]    5    6    7
[3,]    1    2    3
> diag(A)   # extract the diagonal
[1] 2 6 3
> sum(diag(A))   # trace of a matrix
[1] 11
> X <- diag(c(1,2,3,4)) # create a diagonal matrix
> X
     [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]    0    2    0    0
[3,]    0    0    3    0
[4,]    0    0    0    4
> I <-  diag(4) # create an identity matrix
> I
     [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]    0    1    0    0
[3,]    0    0    1    0
[4,]    0    0    0    1
> solve(B)
      [,1]  [,2] [,3]
[1,] -3.00  1.00 -1.0
[2,]  0.25 -0.25  0.5
[3,]  1.00  0.00  0.0
> eigen(A)    # compute eigenvalues and eigenvectors of a matrix
$values:
[1]  1.072015e+001  2.798467e-001 -1.887379e-015
$vectors:
           [,1]      [,2]       [,3]
[1,] -0.4902022 -2.332769 -0.7817656
[2,] -0.6806916  0.239993  0.1954414
[3,] -0.8711809  2.812755  0.5863242
> prod(eigen(A)$values)   # determinant
[1] -5.662137e-015
```

Επίσης μπορούν να χρησιμοποιηθούν οι συναρτήσεις `kronecker` (για το Kronecker γινόμενο δύο πινάκων), `qr` (για ανάλυση QR), `svd` (για ανάλυση ιδιάζουσας τιμής) και `chol` (για ανάλυση Choleski).

3.3 Γραμμικό Σύστημα Εξισώσεων

Η εντολή `solve` δε χρησιμεύει μόνο στον υπολογισμό του αντίστροφου ενός πίνακα, αλλά και στην επίλυση ενός γραμμικού συστήματος εξισώσεων της μορφής $Ax = y$, με την προϋπόθεση ότι υπάρχει λύση. Για παράδειγμα, έστω το γραμμικό σύστημα 2 εξισώσεων και 2 αγνώστους,

$$\begin{aligned} 2x + 3y &= 13 \\ x - 2y &= -4 \end{aligned}$$

Για να λυθεί αυτό το σύστημα, πρώτα κατασκευάζεται ο πίνακας A με τους συντελεστές των αγνώστων και μετά υπολογίζεται η λύση, θέτοντας σαν δεύτερο όρισμα το διάνυσμα των σταθερών όρων, όπως φαίνεται πιο κάτω.

```
> A <- rbind( c(2,3), c(1,-2))
> A
     [,1] [,2]
[1,]    2    3
[2,]    1   -2
> solve(A, c(13,-4))
[1] 2 3
> solve(A)  # getting the inverse
          [,1]       [,2]
[1,] 0.2857143  0.4285714
[2,] 0.1428571 -0.2857143
> solve(rbind(c(1,2), c(2,4))) # getting the inverse of a singular matrix
Error in solve.qr(a): apparently singular matrix
```

Περισσότερες συναρτήσεις σε σχέση με πράξεις πινάκων βρίσκονται μέσα στη βιβλιοθήκη της R, `Matrix`, η οποία καλείται με την εντολή `library(Matrix)`.

3.4 Τυχαίοι Αριθμοί

Στην R υπάρχουν πολλές συναρτήσεις για τη γέννηση τυχαίων αριθμών και υπολογισμών πιθανοτήτων σε σχέση με τις πιο γνωστές κατανομές πιθανοτήτων. Κάθε

μια από αυτές τις συναρτήσεις έχει όνομα το οποίο αρχίζει με ένα από τα ακόλουθα τέσσερα γράμματα, τα οποία καθορίζουν το είδος της συνάρτησης.

r: Γεννήτρια τυχαίων αριθμών.

p: Συνάρτηση κατανομής ($F(x) = P[X \leq x]$).

d: Συνάρτηση πιθανότητας ($f(x)$).

q: Αντίστροφη συνάρτηση κατανομής ($F^{-1}(x)$).

Ο ακόλουθος πίνακας παρουσιάζει τις πιο σημαντικές συναρτήσεις κατανομών στην R και τα επόμενα παραδείγματα εξηγούν πως χρησιμοποιούνται αυτές οι συναρτήσεις.

`beta`	Κατανομή Βήτα
`binom`	Διωνυμική Κατανομή
`chisq`	X^2 Κατανομή
`gamma`	Κατανομή Γάμμα
`lnorm`	Κατανομή Lognormal
`norm`	Κανονική Κατανομή
`pois`	Κατανομή Poisson
`t`	Κατανομή t
`unif`	Ομοιόμορφη Κατανομή

Πίνακας 3.4: Κατανομές τυχαίων μεταβλητών.

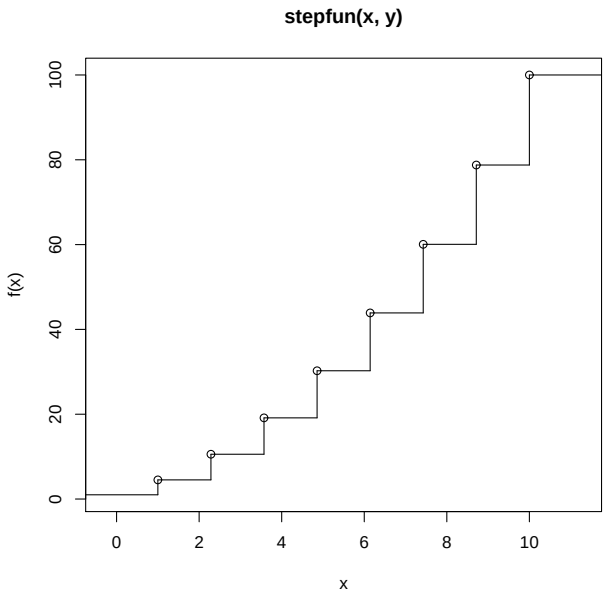
```
> x
[1] 1 4 7
> pnorm(x)
[1] 0.8413447 0.9999683 1.0000000
> pnorm(x, mean=2, sd=2)
[1] 0.3085375 0.8413447 0.9937903
> dnorm(x)
[1] 2.419707e-001 1.338302e-004 9.134720e-012
> qchisq(c(0.90,0.95,0.99), 2)
[1] 4.605170 5.991465 9.210340
> runif(30, -10, 10)
 [1]  9.213183280  8.749200171 -9.117961340  5.292370273  4.117153846
```

```
 [6]  0.071010422  8.572964445  6.805462409  0.942033436 -0.243897894
[11] -2.020305358 -4.729607552  8.518492579 -1.429708684  9.201227576
[16] -4.033246860  1.544312881 -0.227093771 -6.805263087 -6.349458788
[21] -5.736339493 -4.680284206  4.654475767  2.877361933  7.949797967
[26] -0.003705453  1.538872020  8.116273358 -9.711499065  4.931013034
```

3.5 Άλλες Χρήσιμες Συναρτήσεις

Στην R υπάρχουν και άλλες πολλές συναρτήσεις οι οποίες μπορούν να χρησιμοποιηθούν για υπολογισμούς αλλά δεν μπορούν όλες να επεξηγηθούν λεπτομερώς σε αυτό το κεφάλαιο. Αξίζει να αναφερθεί η συνάρτηση `integrate`, η οποία υπολογίζει το ολοκλήρωμα μιας πραγματικής συνάρτησης σε ένα διάστημα τιμών, η συνάρτηση `diff` η οποία επιστρέφει την n-οστή διαφορά με βήμα k για ένα σύνολο τιμών και η συνάρτηση `fft` η οποία δίνει τον γρήγορο μετασχηματισμό Fourier ενός συνόλου τιμών. Ακολουθεί ένα παράδειγμα με τη συνάρτηση `stepfun` η οποία υπολογίζει την αριστερή συνεχή συνάρτηση βήματος από σημεία (x, y).

```
> x <- seq(1,10, length=8)
> y <- seq(1,10,length=9)^2
> stepfun(x,y)
Step function
Call: stepfun(x, y)
 x[1:8] =      1, 2.2857, 3.5714,  ..., 8.7143,     10
9 plateau levels =      1, 4.5156, 10.562,  ..., 78.766,    100
> plot.stepfun(stepfun(x,y))
```



Σχήμα 3.1: Συνάρτηση βήματος

Κεφάλαιο 4

Γραφήματα

Τα *γραφήματα* είναι πολύ χρήσιμα για την οπτική αναπαράσταση των δεδομένων και καθοδηγούν τον στατιστικό στην διαδικασία της μοντελοποίησης και αξιολόγησης της ανάλυσης. Το κεφάλαιο αυτό περιγράφει μερικές χρήσιμες συναρτήσεις γραφημάτων που υπάρχουν στην R και κάνει εισαγωγή στις διάφορες γραφικές παραμέτρους όπως την εισαγωγή πληροφοριών στο γράφημα αλλά και την εποπτική συσχέτιση. Όπως θα δούμε, η R δίνει ένα πολύ ισχυρό περιβάλλον για τη δημιουργία γραφημάτων.

Εκτός από τα γραφήματα και τα χαρακτηριστικά τους τα οποία θα δούμε πιο κάτω, η R περιλαμβάνει και τη βιβλιοθήκη Trellis Graphics. Τα γραφήματα Trellis έχουν περισσότερη ευελιξία και μπορούν να χρησιμοποιηθούν για πολλαπλά γραφήματα και βελτιωμένες τρισδιάστατες αναπαραστάσεις.

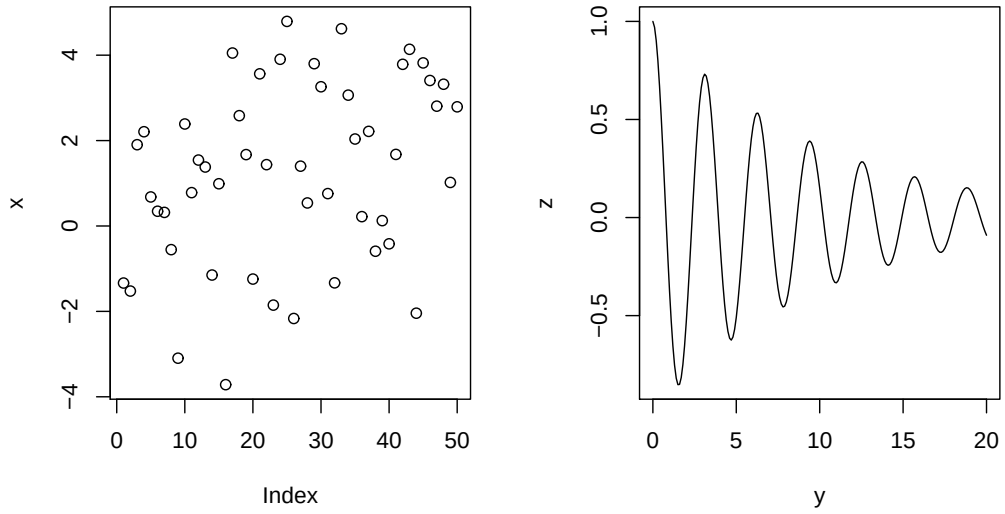
4.1 Απλά Γραφήματα

Τα πιο απλά γραφήματα είναι οι γραφικές παραστάσεις που συσχετίζονται με μονοδιάστατη τυχαία μεταβλητή, οι γραφικές παραστάσεις συναρτήσεων και οι γραφικές παραστάσεις χρονοσειρών. Η βασική εντολή για γραφική παράσταση είναι η εντολή `plot`, η οποία έχει πολλές δυνατότητες και μπορεί να πάρει διάφορες γραφικές παραμέτρους για ορίσματα. Ακολουθούν μερικά απλά παραδείγματα.

```
> x <- rnorm(50, mean=1, sd=2)
> plot(x)
> y <- seq(0,20, .1)
> z <- exp(-y/10)*cos(2*y)
```

```
> plot(y,z, type="l")
```

Η τέταρτη εντολή δίνει το γράφημα της $f(y) = e^{-\frac{y}{10}} \cos 2y$. Με αυτόν τον τρόπο δουλεύουμε συνήθως όταν θέλουμε να δημιουργήσουμε γραφικές παραστάσεις συναρτήσεων. Τα αντίστοιχα γραφήματα παρουσιάζονται στο Σχήμα 4.1.



Σχήμα 4.1: Απλά γραφήματα

Όπως φαίνεται στο δεύτερο παράδειγμα, ένα διάγραμμα διασποράς (scatter plot) μπορεί να κατασκευαστεί στην R εφαρμόζοντας την εντολή `plot` σε ένα ζεύγος διανυσμάτων της ίδιας διάστασης, ή σε μια λίστα με συνιστώσες `x` και `y`.

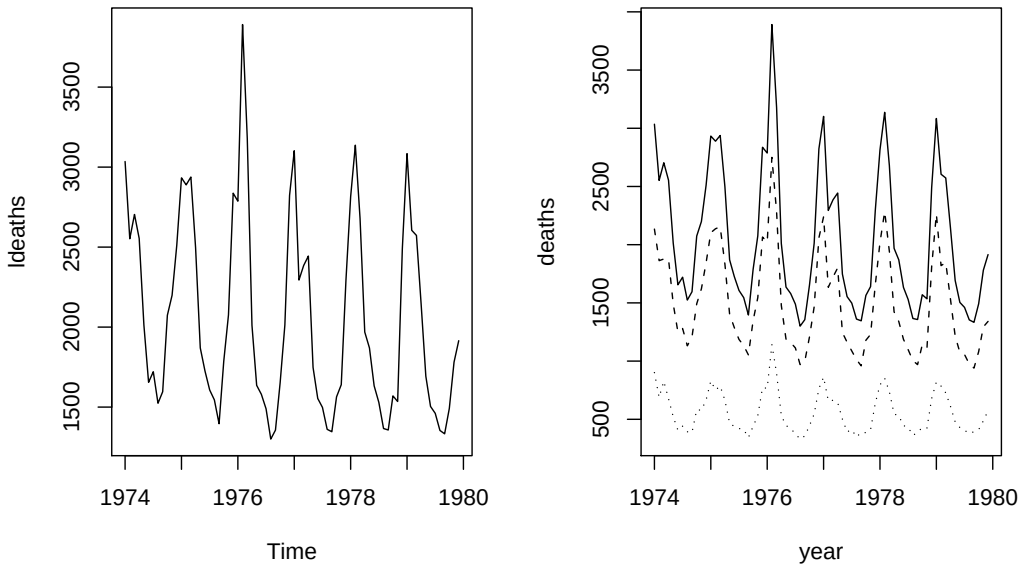
```
> x1 <- rnorm(50)
> plot(x1,x)
> plot(cbind(x1,x))
```

Όταν τα δεδομένα παρατηρούνται διαδοχικά στον χρόνο, είναι φυσικό να γίνει η γραφική παράσταση των δεδομένων σε συνάρτηση με τον χρόνο (χρονοσειρές). Στην R αυτό γίνεται χρησιμοποιώντας την εντολή `ts.plot`. Για παράδειγμα, έστω τα πλαίσια δεδομένων ldeaths, mdeaths και fdeaths που ανήκουν στην στην R και αναφέρονται στους μηνιαίους θανάτους από καρκίνο του πνεύμονα στο Ηνωμένο

Βασίλειο κατά την περίοδο από το 1974 ως το 1979 συνολικά, στους άντρες και στις γυναίκες, αντίστοιχα.

```
> ts.plot(ldeaths)
> ts.plot(ldeaths,mdeaths,fdeaths,gpars=list(xlab="year",ylab="deaths",lty=1:3))
```

Οι γραφικές παραστάσεις φαίνονται στο Σχήμα 4.2 και παρατηρείται ότι στο δεύτερο γράφημα κάθε χρονοσειρά παρουσιάζεται με διαφορετικό είδος γραμμής.



Σχήμα 4.2: Γραφήματα χρονοσειρών.

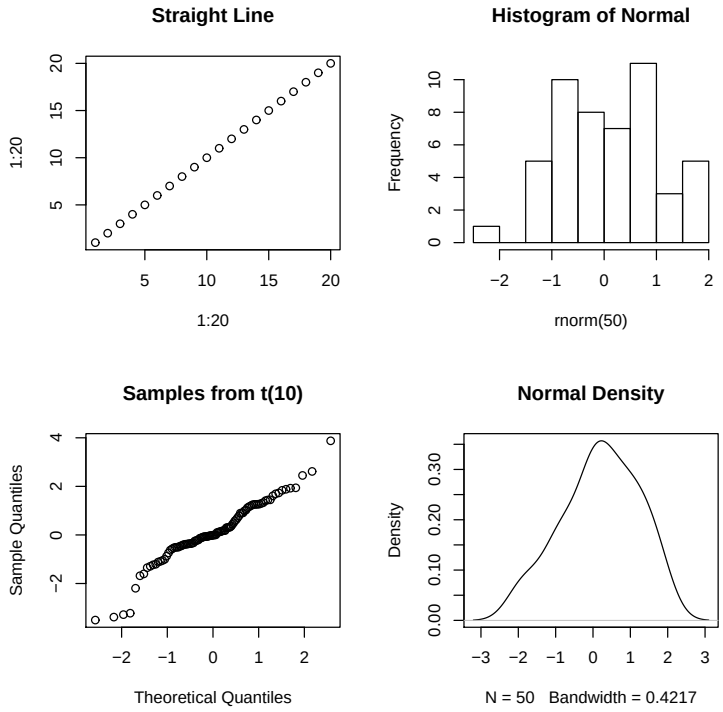
4.2 Γραφικές Δυνατότητες

Υπάρχουν πολλές γραφικές δυνατότητες αλλά η παρουσίαση θα περιοριστεί μόνο σε μερικές οι οποίες στο τέλος είναι πιο χρήσιμες από τις υπόλοιπες. Η διαρρύθμιση του γραφήματος μπορεί να τακτοποιηθεί έτσι ώστε να παρουσιάζει περισσότερες από μια γραφική παράσταση (βλέπε Σχήμα 4.3).

```
> par(mfrow=c(2,2)) #a 2x2 plot
```

```
> plot(1:20,1:20,main="Straight Line")
> hist(rnorm(50),main="Histogram of Normal")
> qqnorm(rt(100,10),main="Samples from t(10)")
> plot(density(rnorm(50)),main="Normal Density")
> par(mfrow=c(1,1))
```

Το πρώτο γράφημα (Σχήμα 4.3, πάνω αριστερά) δίνει τη γραφική παράσταση της $f(x) = x$, $x \in [1, 20]$. Το δεύτερο γράφημα (Σχήμα 4.3, πάνω δεξιά) μας δίνει το ιστόγραμμα 50 τυχαίων παρατηρήσεων από την τυπική κανονική. Παρόμοια, το τρίτο γράφημα (Σχήμα 4.3, κάτω αριστερά) μας δίνει την γραφική παράσταση των δειγματικών ποσοστημορίων από την t κατανομή με 10 βαθμούς ελευθερίας ως προς τα θεωρητικά ποσοστημόρια της τυπικής κανονικής. Το τελευταίο γράφημα απεικονίζει τη μη παραμετρική εκτιμήτρια συνάρτησης πυκνότητας πιθανότητας από 50 παρατηρήσεις της τυπικής κανονικής. Παρόμοια διάταξη γραφημάτων μπορεί να επιτευχθεί με την εντολή `split.screen`.



Σχήμα 4.3: $2x2$ διαρύθμιση γραφημάτων

Έχοντας σαν βάση το προηγούμενο παράδειγμα, είναι αρκετά εύκολο να εισαχθεί κύριος τίτλος ή υπότιτλος σε μια γραφική παράσταση.

```
> plot(x,main="Sample From Normal")
> plot(x,sub="Mean 1 and variance 4")
> plot(x, main="Sample from Normal", sub="Mean 1 and Variance 4")
> plot(x)
> title(main="Sample from Normal", sub="Mean 1 and Variance 4")
```

Επίσης, μπορούν να δοθούν ονόματα στους άξονες χρησιμοποιώντας τα ορίσματα `xlab` και `ylab`, όπως το επόμενο παράδειγμα.

```
> plot(x, xlab="Index", ylab="Sample from Normal")
> plot(x, xlab="", ylab="")  #no axis labels
> title(xlab="Index", ylab="Sample from Normal")
```

Η δεύτερη εντολή δε δίνει ονόματα στους άξονες. Οι εντολές `xlim` και `ylim` χρησιμεύουν στο να αλλάξουν τα όρια των αξόνων, θέτοντας αυτά που κάνουν το γράφημα πιο εύκολο για κατανόηση.

4.3 Είδη και Γραμμές Γραφικής Παράστασης

Στην R τα δεδομένα μπορούν να απεικονιστούν σε γράφημα με διάφορους τρόπους. Αυτό επιτυγχάνεται με το όρισμα `type` στην εντολή `plot`. Αυτοί οι τρόποι φαίνονται στον πιο κάτω πίνακα.

Σύμβολο	**Είδος (Type)**
`"p"`	Σημεία
`"l"`	Γραμμή
`"b"`	Γραμμή και Σημεία
`"c"`	Γραμμή με κενό στα σημεία
`"o"`	Γραμμή και Σημεία ενωμένα
`"h"`	Κάθετες γραμμές για κάθε σημείο
`"s"`	Με Βήμα
`"n"`	Τίποτα

Πίνακας 4.1: Είδη Γραφικής Παράστασης

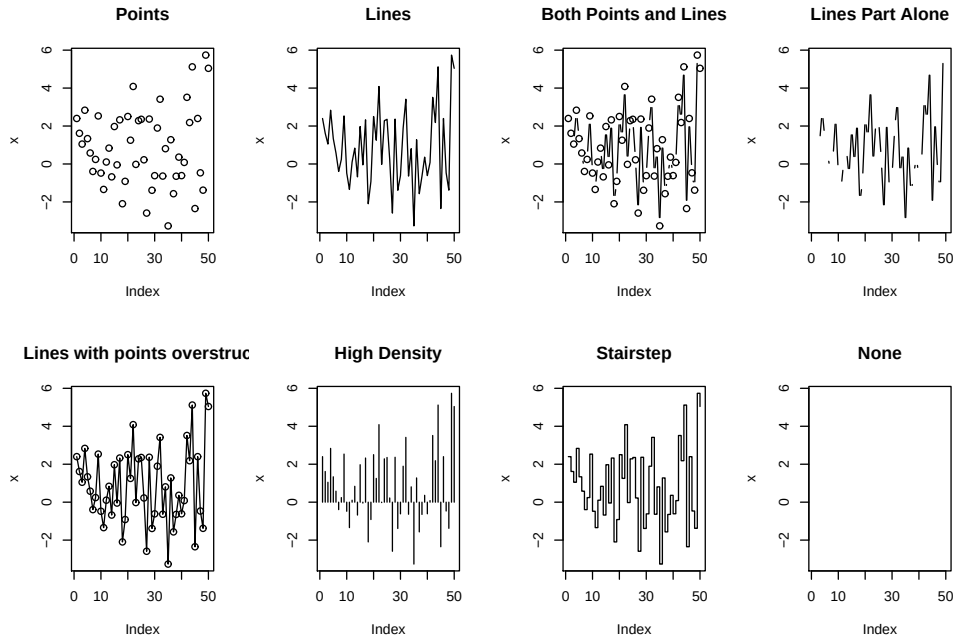
Ακολουθεί ένα παράδειγμα για το πως χρησιμοποιούνται τα πιο πάνω. Τα γραφήματα φαίνονται στο Σχήμα 4.4. Δημιουργούνται από τα αριστερά προς δεξιά ανά γραμμή.

```
> par(mfrow=c(2,4))
> plot(x, type="p")
> title(main="Points")
> plot(x, type="l")
> title(main="Lines")
> plot(x, type="b")
> title(main="Both Points and Lines")
> plot(x, type="c")
> title(main="Lines Part Alone")
> plot(x, type="o")
> title(main="Lines with points overstruck")
> plot(x, type="h")
> title(main="High Density")
> plot(x, type="s")
> title(main="Stairstep")
> plot(x, type="n")
> title(main="None")
> par(mfrow=c(1,1))
```

Όταν το είδος της γραφικής παράστασης περιλαμβάνει γραμμές, τότε μπορεί να επιλεγεί διαφορετικό είδος γραμμής δίνοντας διάφορους αριθμούς στο όρισμα `lty`. Για παράδειγμα, η διακεκομμένη γραμμή με παύλες συμβολίζεται με `lty=2`. Το εξ ορισμού είδος γραμμής είναι η συνεχής γραμμή. Υπάρχουν οκτώ διαφορετικά είδη γραμμής. Επιπρόσθετα, μπορούμε να δώσουμε χρώμα στο είδος γραφικής παράστασης δίνοντας αριθμούς ή σε εισαγωγικά τα αγγλικά ονόματα των χρωμάτων στο όρισμα `col` της εντολής `plot` (π.χ. `col="green"` για πράσινο χρώμα).

4.4 Προσθήκη Πληροφοριών σε Γράφημα

Σε μερικές περιπτώσεις, είναι αναγκαίο να υποδειχθούν οι απομακρυσμένες τιμές, να προστεθεί ένα κείμενο ή άλλες πληροφορίες σε ένα γράφημα με διαδραστικό τρόπο. Υπάρχουν διάφοροι τρόποι με τους οποίους η R μπορεί να το κάνει αυτό αλληλεπιδρώντας *με το χρήστη*. Ακολουθεί ένα παράδειγμα για το πώς μπορεί να υποδειχθεί μια απομακρυσμένη τιμή (βλ. Σχήμα 4.5).

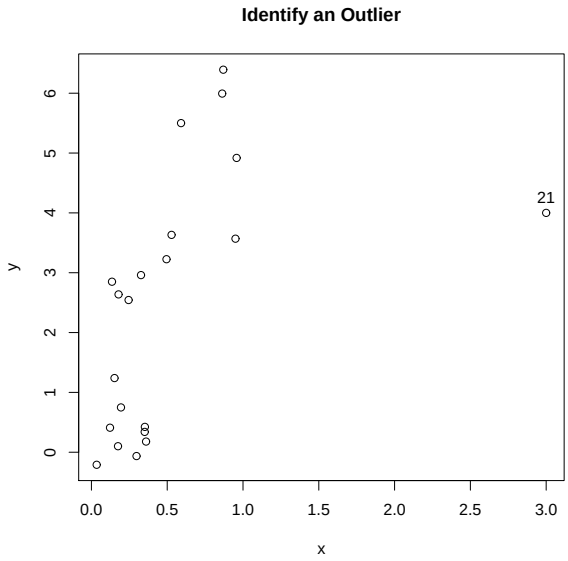


Σχήμα 4.4: Είδη γραφικών παραστάσεων.

```
> x <- runif(20)
> y <- 6*x+rnorm(20)
> x <- c(x,3)
> y <- c(y,4)
> plot(x,y)
> identify(x,y, n=1) #R waits until you click the mouse on the selected point
[1] 21
```

Γενικά, μπορούν να υποδειχθούν όσα σημεία επιθυμεί ο χρήστης με την εντολή `identify(x,y, n=k)`, όπου k είναι ο αριθμός των σημείων που θα υποδειχθούν. Τα ακόλουθα βοηθούν στο να γίνει κατανοητός ο τρόπος που μπορεί να προστεθεί η ευθεία ελαχίστων τετραγώνων, αλλά και καινούργια σημεία ή ευθείες σε μια γραφική παράσταση (βλ. Σχήμα 4.6).

```
> plot(x,y)
> abline(lm(y~x), lty=2)
> plot(y^2, type="l", xlab="", ylab="Square of Y")
> lines(y, lty=2)
```

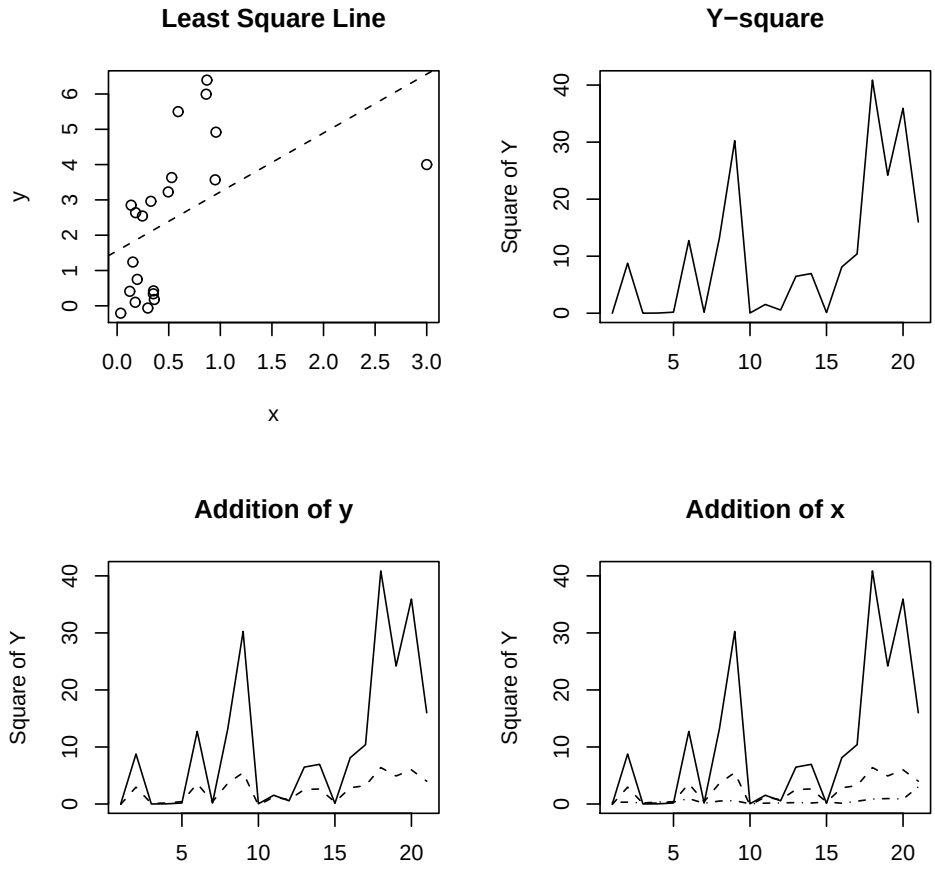


Σχήμα 4.5: Υπόδειξη απομακρυσμένης τιμής.

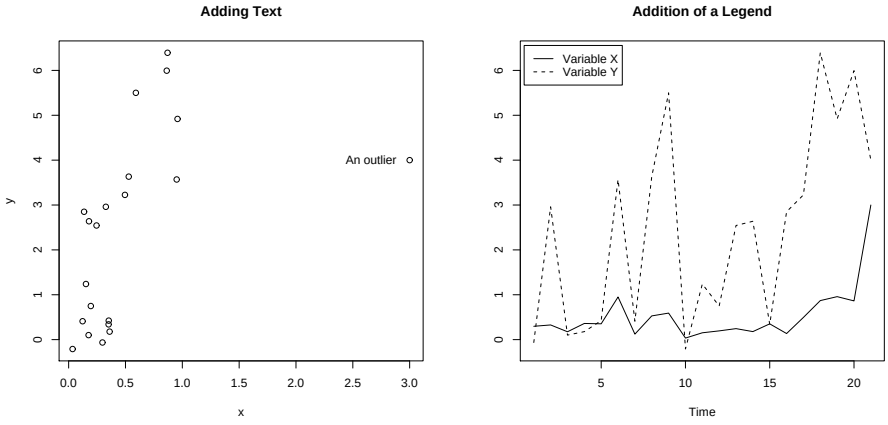
```
> lines(x, lty=4)
```

Είναι δυνατόν επίσης να προστεθεί κείμενο και υπόμνημα στη γραφική παράσταση (βλ. Σχήμα 4.7).

```
> plot(x, y, main="Adding Text")
> text(locator(1), "An outlier")  #click the mouse to place the text
> ts.plot(ts(x),ts(y),gpars=list(lty=1:2)) #time series plot of both x and y
> leg.names <- c("Variable X","Variable Y")
> legend(locator(1), leg.names, lty=1:2)  #click the mouse to place the legend
```



Σχήμα 4.6: Εισαγωγή γραμμών.

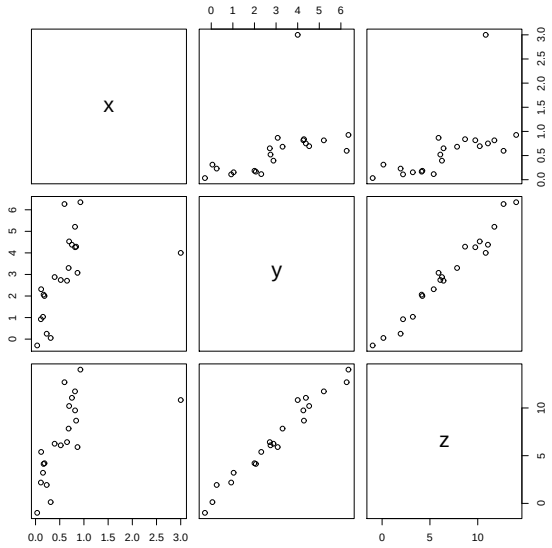


Σχήμα 4.7: Προσθήκη κειμένου και υπομνήματος.

4.5 Γραφήματα Σε Μεγαλύτερες Διαστάσεις

Για να κατασκευαστούν γραφικές παραστάσεις πολυδιάστατων τυχαίων μεταβλητών, ένας τρόπος είναι να κατασκευαστούν διαγράμματα διασπορών για κάθε ζευγάρι μεταβλητών, ξεχωριστά. Για παράδειγμα, έστω ότι ορίζεται η μεταβλητή Z από τις προυπάρχουσες μεταβλητές X και Y και κατασκευάζεται πίνακας με τρεις στήλες. Για να ερευνηθεί η συσχέτιση μεταξύ των τριών μεταβλητών χρησιμοποιείται η εντολή pairs(), δηλαδή

```
> z <- x+2*y+rnorm(21)
> pairs(cbind(x,y,z))
```



Σχήμα 4.8: Ζευγάρια διαγραμμάτων διασποράς.

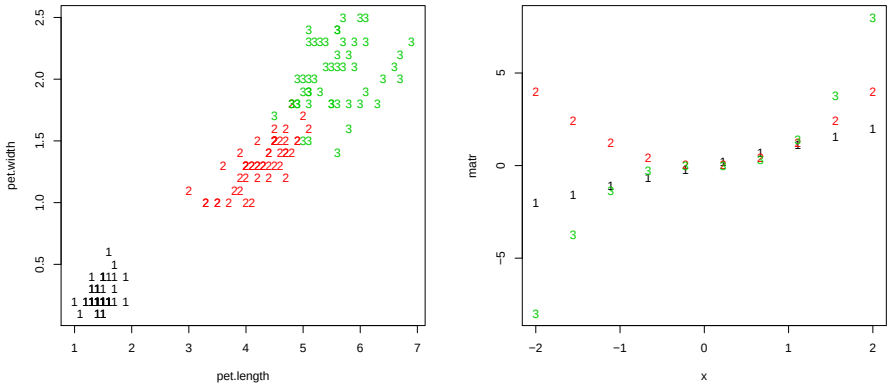
Όταν πρέπει να παρουσιαστούν διάφορα διανύσματα δεδομένων ή πολυδιάστατα δεδομένα στο ίδιο γράφημα, τότε μπορεί να εφαρμοστεί η εντολή matplot η οποία κατασκευάζει γραφική παράσταση των στηλών ενός πίνακα συναρτήσει των στηλών κάποιου άλλου. Για σκοπό παραδείγματος, έστω το πλαίσιο δεδομένων iris3 από το οποίο εξάγουμε το pet.length και το pet.width. Ο πίνακας pet.length περιλαμβάνει 50 παρατηρήσεις (γραμμές) του μήκους του πετάλου τριών ειδών ίριδων (στήλες): Setosa, Versicolor και Virginica. Ο πίνακας pet.width περιλαμβάνει 50 παρατηρήσεις (γραμμές) του πλάτους του πετάλου για καθένα από

τα τρία είδη ίριδων. Για να ερευνηθεί γραφικά η συσχέτιση μεταξύ του μήκους και του πλάτους των πετάλων, χρησιμοποιείται η εντολή `matplot` για να παρουσιάσει το μήκος συναρτήσει του πλάτους και για τα τρία είδη σε ένα γράφημα.

```
> pet.length <- iris3[,3,]
> pet.width <- iris3[,4,]
> matplot(pet.length,pet.width)
```

Ακόμη ένα παράδειγμα για τη χρήση της εντολής `matplot` είναι το ακόλουθο:

```
x<-seq(-2,2,length=10)
y<-x^2
z<-x^3
matr<-cbind(x,y,z)
matplot(x,matr)
```



Σχήμα 4.9: Γράφημα matplot.

Πολλά είδη δεδομένων μπορούν να παρουσιαστούν σε μορφή επιφάνειας που παράγεται από συναρτήσεις δύο μεταβλητών. Η R παρέχει τρείς εντολές για παρουσίαση τέτοιων δεδομένων. Η πιο απλή, `contour`, παρουσιάζει την επιφάνεια σε διάγραμμα ισουψών. Η προοπτική απεικόνιση της επιφάνειας γίνεται με την εντολή `persp`. Τέλος η εντολή `image` παρουσιάζει την επιφάνεια με βοήθεια χρωμάτων ή αποχρώσεις του γκρίζου. Και οι τρεις εντολές έχουν τα ίδια ορίσματα: το διάνυσμα των συντεταγμένων του x, το διάνυσμα των συντεταγμένων του y, και ένα πίνακα με τις τιμές του z με διαστάσεις το μήκος του x και το μήκος του y. Τέτοια παραδείγματα μπορείτε να δείτε με την εντολή `demo("persp")`.

Κεφάλαιο 5

Απλός Προγραμματισμός στην R

Η έννοια του προγραμματισμού στην R βασίζεται στη δημιουργία καινούργιων *συναρτήσεων* οι οποίες θα χρησιμοποιηθούν για περαιτέρω ανάπτυξη της γλώσσας. Το κύριο δομικό υλικό είναι οι υπάρχουσες συναρτήσεις (functions) της R, μερικές από τις οποίες ήδη έχουμε εξετάσει σε προηγούμενα κεφάλαια.

5.1 Λογικοί Τελεστές και Τελεστές Σύγκρισης

Οι κύριοι λογικοί τελεστές και τελεστές σύγκρισης αναφέρονται στον πίνακα που ακολουθεί.

Οι τελεστές `&` και `|` αξιολογούν τις ανάλογες εκφράσεις στοιχείο με στοιχείο και επιστρέφουν ένα διάνυσμα με τις λογικές τιμές `TRUE` και `FALSE`.

```
> x <- seq(-1,1,length=12)
> x
 [1] -1.00000000 -0.81818182 -0.63636364 -0.45454545 -0.27272727 -0.09090909
 [7]  0.09090909  0.27272727  0.45454545  0.63636364  0.81818182  1.00000000
> x < 0 | x > 0.8
 [1] TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE TRUE TRUE
> x < 0 & x > 0.8
 [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

Τελεστής	Ερμηνεία
==	ίσο με
>	μεγαλύτερο από
!=	άνισο από
<	μικρότερο από
>=	μεγαλύτερο ή ίσο από
<=	μικρότερο ή ίσο από
&	και
&&	*και* ελέγχου
|	ή
||	*ή* ελέγχου
!	όχι

Πίνακας 5.1: Λογικοί τελεστές και τελεστές σύγκρισης.

Οι τελεστές ελέγχου χρησιμοποιούνται για να κατασκευάζονται υποθετικές προτάσεις.

5.2 Χρησιμοποιώντας Υποσύνολα των Δεδομένων

Αρκετές φορές υπάρχει η ανάγκη να γίνουν διάφοροι υπολογισμοί χρησιμοποιώντας ένα συγκεκριμένο κομμάτι των δεδομένων. Η μέθοδος αυτή ονομάζεται *υποστίξη*. Η R έχει πολύ καλές και εύκολες δυνατότητες στο να πετυχαίνει την υποστίξη. Στο επόμενο παράδειγμα αυτή εφαρμόζεται αρχικά σε διανύσματα.

```
> x
 [1] -1.00000000 -0.81818182 -0.63636364 -0.45454545 -0.27272727 -0.09090909
 [7]  0.09090909  0.27272727  0.45454545  0.63636364  0.81818182  1.00000000
> x[3]    # extract  the third element
[1] -0.6363636
> x[c(1,2,5)]   # extract the first, second and fifth elements.
[1] -1.0000000 -0.8181818 -0.2727273
> x[-(3:10)]    # extract all the elements except those in positions 3 to 10.
[1] -1.0000000 -0.8181818  0.8181818  1.0000000
> x[ x > 0]     # extract the elements that satisfy the condition.
 [1] 0.09090909 0.27272727 0.45454545 0.63636364 0.81818182 1.00000000
> x[ x > 0 & x < 0.5]
```

```
[1] 0.09090909 0.27272727 0.45454545
```

Δηλαδή είναι εφικτό να πάρουμε υπόσυνολο δεδομένων είτε βάση της θέσης των στοιχείων του είτε βάση μιας συνθήκης. Η υπόστιξη μπορεί να γενικευθεί και στην περίπτωση των πινάκων.

```
>A <- cbind(c(1,2,-1), c(12,15,18), c(-1,-4,-9))
> A
     [,1] [,2] [,3]
[1,]    1   12   -1
[2,]    2   15   -4
[3,]   -1   18   -9
> A[1,1]  #extracts the (1,1) element
[1] 1
> A[1,3]  # extracts the (1,3) element
[1] -1
> A[1:2,3]  #extracts the elements (1,3), (2,3)
[1] -1 -4
> A[1:2,2:3]   #extracts a two by two matrix
     [,1] [,2]
[1,]   12   -1
[2,]   15   -4
> A[,2:3]      # omission of a dimension gives the corresponding columns
     [,1] [,2]
[1,]   12   -1
[2,]   15   -4
[3,]   18   -9
> A[-1,2:3]   # use of negative indices
     [,1] [,2]
[1,]   15   -4
[2,]   18   -9
```

Γενικεύεται επίσης και στα αντικείμενα λίστας,

```
> mylist <- list(x,A)
> mylist
[[1]]:
 [1] -1.00000000 -0.81818182 -0.63636364 -0.45454545 -0.27272727 -0.09090909
```

```
 [7]  0.09090909  0.27272727  0.45454545  0.63636364  0.81818182  1.00000000
[[2]]:
     [,1] [,2] [,3]
[1,]    1   12   -1
[2,]    2   15   -4
[3,]   -1   18   -9
> mylist[[1]]
 [1] -1.00000000 -0.81818182 -0.63636364 -0.45454545 -0.27272727 -0.09090909
 [7]  0.09090909  0.27272727  0.45454545  0.63636364  0.81818182  1.00000000
> mylist[[2]]
     [,1] [,2] [,3]
[1,]    1   12   -1
[2,]    2   15   -4
[3,]   -1   18   -9
```

καθώς και σε πλαίσια δεδομένων με τη χρήση των συμβόλων `[[]]` και `$`, αντίστιχα.

```
> library(MASS)
> is.data.frame(survey)
[1] TRUE
> names(survey)
 [1] "Sex"    "Wr.Hnd" "NW.Hnd" "W.Hnd"  "Fold"   "Pulse"  "Clap"   "Exer"
 [9] "Smoke"  "Height" "M.I"    "Age"
> survey$Age[1:100]
  [1] 18.250 17.583 16.917 20.333 23.667 21.000 18.833 35.833 19.000 22.333
 [11] 28.500 18.250 18.750 17.500 17.167 17.167 19.333 18.333 19.750 17.917
 [21] 17.917 18.167 17.833 18.250 19.167 17.583 17.500 18.083 21.917 19.250
 [31] 41.583 17.500 39.750 17.167 17.750 18.000 19.000 17.917 35.500 19.917
 [41] 17.500 17.083 28.583 17.500 17.417 18.500 18.917 19.417 18.417 30.750
 [51] 18.500 17.500 18.333 17.417 20.000 18.333 17.167 17.417 17.667 18.417
 [61] 20.333 17.333 17.500 19.833 18.583 18.000 30.667 16.917 19.917 18.333
 [71] 17.583 17.833 17.667 17.417 17.750 20.667 23.583 17.167 17.083 18.750
 [81] 16.750 20.167 17.667 17.167 17.167 17.250 18.000 18.750 21.583 17.583
 [91] 19.667 18.000 19.667 17.083 22.833 17.083 19.417 23.250 18.083 19.083
```

5.3 Κατασκευή Συναρτήσεων

Για να γίνει κατανοητή η έννοια της κατασκευής νέων συναρτήσεων στην R, θα εξεταστεί το ακόλουθο παράδειγμα το οποίο δίνει σαν αποτέλεσμα την τυπική απόκλιση ενός διανύσματος x:

```
>standard.deviation <- function(x)
{
sqrt(var(x))
}
> x <- rnorm(100, mean=0, sd=2) #100 observations from normal
                                #with mean 0 and variance 4
> var(x)
[1] 3.879332
> standard.deviation(x)
[1] 1.969602
```

Συνεπώς, για να υπολογιστεί η τυπική απόκλιση (`standard.deviation`) αξίζει να σημειωθεί ότι χρησιμοποιήθηκαν δύο από τις προυπάρχουσες συναρτήσεις, η `sqrt()` και η `var()`. Αυτή είναι η θεμελιώδης ιδέα όταν υπάρχει η ανάγκη ορισμού νέας συνάρτησης στην R. Φυσικά υπάρχει συγκεκριμένη συνάρτηση στην R για υπολογισμό της τυπικής απόκλισης και αυτή είναι η `sd`. Παρατίθενται μερικές βασικές δηλώσεις που είναι χρήσιμες στον ορισμό καινούργιων συναρτήσεων:
Μερικές επιπρόσθετες δηλώσεις είναι οι η `switch()` και η `stop()`. Τα επόμενα παραδείγματα αναλύουν τα πιο πάνω. Το πρώτο παράδειγμα επεξηγεί το πώς χρησιμοποιείται η εντολή `if` για να γεννηθούν δείγματα από διάφορες κατανομές.

```
random.gener <- function(n, distribution, shape)
{
# a function to generate n random numbers
if(distribution=="gamma") rgamma(n, shape) else
if(distribution=="exp")   rexp(n) else
if(distribution=="norm") rnorm(n) else
stop("Invalid Distribution")
}
> random.gener(10, "gamma", 2)
 [1] 0.3461286 2.0791867 3.2288429 4.3973702 1.7676279 2.7317868
 [7] 0.4084932 2.4203665 0.7430161 5.1688287
```

Εντολή	**Ερμηνεία**
`if` (A) B	ελέγχει αν ισχύει το A, αν ναι τότε εκτελεί το B
`if` (A) B1 `else` B2	ελέγχει αν ισχύει το A, αν ναι τότε εκτελεί το B1, διαφορετικά εκτελεί το B2
`ifelse`(A,B1,B2)	πιο απλός τρόπος γραφής του προηγούμενου
`break`	τερματίζει τον τρέχων βρόγχο
`next`	τερματίζει τον τρέχων βρόγχο και αρχίζει την επόμενη επανάληψη
`return`(A)	τερματίζει την τρέχων συνάρτηση και επιστρέφει το A
`while` (A) B	ελέγχει κατ΄ επανάληψη αν ισχύει το A, αν ναι τότε εκτελεί το B
`repeat` A	απλούστερη συνάρτηση για το `while`
`for` (`index in` A) B	βρόγχος, αλλά απαιτεί αρκετόν υπολογιστικό χρόνο

Πίνακας 5.2: Βασικές δηλώσεις.

```
> random.gener(10, "unif", 2)
Error in random.gener(10, "unif", 2): Invalid Distribution
```

Η παραπάνω συνάρτηση δημιουργεί δείγμα μεγέθους n από τις κατανομές Γάμμα, Εκθετική και Τυπική Κανονική. Διαφορετικά, αν του δώσουμε μια οποιαδήποτε άλλη κατανομή θα επιστρέψει ότι έγινε σφάλμα.

Το επόμενο παράδειγμα παρουσιάζει τον τρόπο που μπορούν να χρησιμοποιηθούν οι εντολές `for` και `if` για να βρεθεί το πρόσημο ενός πραγματικού αριθμού.

```
new.sign <-  function(x)
{
   for (i in 1:length(x))
   {
         if(x[i] > 0)
            x[i] <- 1
    else if(x[i] < 0)
            x[i] <- -1
   }
   x
}
> new.sign(-10:5)
 [1] -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  0  1  1  1  1  1
```

Ωστόσο, υπάρχει καλύτερος τρόπος για να επιτευχθεί αυτό, αποφεύγοντας τις επαναλήψεις, οι οποίες απαιτούν περισσότερο υπολογιστικό χρόνο. Εδώ, φανερώνεται ακόμη μία φορά η χρησιμότητα της υπόστιξης, η οποία βοηθάει στο να κερδίζεται πολύτιμος υπολογιστικός χρόνος.

```
sgnfunction <- function(x)
{
ifelse(x > 0, 1, ifelse(x<0, -1, 0))
}
> sgnfunction(-10:10)
[1] -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  0  1  1  1  1  1  1  1  1  1  1
```

Παράδειγματα συναρτήσεων καθώς και περαιτέρω εφαρμογές τους θα δούμε στα κεφάλαια που ακολουθούν.

Κεφάλαιο 6

Προσομοίωση

Αυτό το κεφάλαιο συμπληρώνει το προηγούμενο κεφάλαιο το οποίο αναφερόταν στο πώς μπορεί να γίνει απλός προγραμματισμός στην R. Θα χρησιμοποιηθούν οι έννοιες του προγραμματισμού για να εξαχθούν αποτελέσματα από απλές προσομοιώσεις γνωστών πιθανοθεωρητικών αποτελεσμάτων.

6.1 Ο Ασθενής Νόμος των Μεγάλων Αριθμών

Σύμφωνα με τον ασθενή νόμο των μεγάλων αριθμών, αν $X_1, \ldots, X_n$ είναι ανεξάρτητες και ισόνομες τυχαίες μεταβλητές με πεπερασμένη μέση τιμή μ, τότε

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i \to \mu,$$

κατά πιθανότητα, όταν το $n \to \infty$. Συγκεκριμένα, αν $X_1, \ldots, X_n$ είναι δίτιμες τυχαίες μεταβλητές με $P(X_i = 1) = p$, τότε

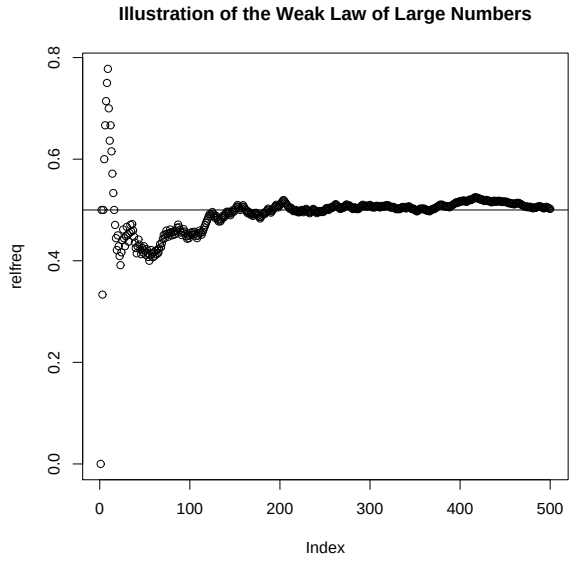
$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i \to p,$$

κατά πιθανότητα, όταν το $n \to \infty$. Το αποτέλεσμα αυτό μπορεί να παρουσιαστεί εμπειρικά στην R με τις ακόλουθες συναρτήσεις :

```
uniforms <- runif(500)
tosses   <- as.numeric(I(uniforms > 0.5))
relfreq  <- cumsum(tosses)/(1:500)
plot(relfreq)
```

```
abline(0.5,0)
title(main="Illustration of the Weak Law of Large Numbers")
```

Η πρώτη εντολή δίνει τυχαίο δείγμα $U_1, \ldots, U_{500}$ από την ομοιόμορφη στο (0,1). Στη συνέχεια ορίζουμε τη δίτιμη τυχαία μεταβλητή $X_i = I(U_i > \frac{1}{2})$, όπου I δείκτρια, $i = 1, \ldots, 500$. Μετά θεωρούμε τη συνάρτηση $\overline{X}$ σαν ακολουθία, δηλαδή το ακολουθιακό ποσοστό επιτυχιών (γιατί ;). Το γράφημα (Σχήμα 6.1) μας δίνει την σύγκλιση της ακολουθίας στο 0.5, όταν $n \to \infty$ σύμφωνα με το νόμο των μεγάλων αριθμών.

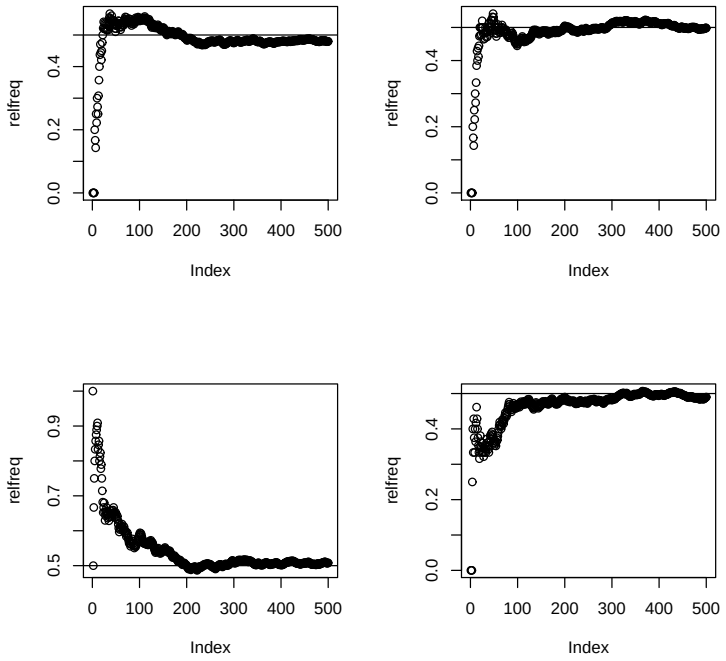


Σχήμα 6.1: Ο ασθενής νόμος των μεγάλων αριθμών για δυαδικές τυχαίες μεταβλητές.

Εφαρμόζοντας τα πιο πάνω τέσσερις φορές και απεικονίζοντας τις γραφικές παραστάσεις σε ένα 2×2 γράφημα (Σχήμα 6.2), έχουμε ότι,

```
par(mfrow=c(2,2))
for(rep in 1:4)
{
    uniforms <- runif(500)
    tosses   <- as.numeric(uniforms > 0.5)
    relfreq  <- cumsum(tosses)/(1:500)
```

```
    plot(relfreq)
    abline(0.5,0)
}
```



Σχήμα 6.2: Ο ασθενής νόμος των μεγάλων αριθμών για δυαδικές τυχαίες μεταβλητές.

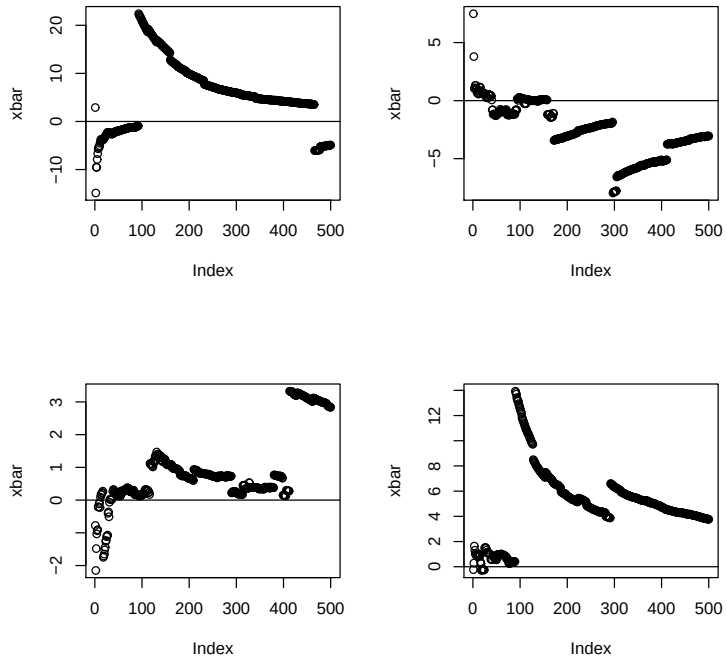
Δηλαδή το γράφημα αυτό δείχνει καθαρά την τυχαιότητα αλλά και τη σύγκλιση. Τι συμβαίνει όμως όταν η αναμενόμενη τιμή της τυχαίας μεταβλητής δεν υπάρχει; Ένα πολύ γνωστό παράδειγμα μιας τέτοιας τυχαίας μεταβλητής είναι η κατανομή Cauchy

$$f(x) = \frac{1}{\pi(1+x^2)}, \quad x \in R,$$

της οποίας η αναμενόμενη τιμή δεν υπάρχει (γιατί;). Οι ακόλουθες συναρτήσεις μαζί με το Σχήμα 6.3 καταδεικνύουν ότι η ακολουθία των μέσων τιμών δεν συγκλίνει.

```
par(mfrow=c(2,2))
```

```
for(rep in  1:4)
{
    cauchys <- rcauchy(500)
    xbar    <- cumsum(cauchys)/(1:500)
    plot(xbar)
    abline(0,0)
}
```



Σχήμα 6.3: Ακολουθία μέσων τιμών από την κατανομή Cauchy.

6.2 Κεντρικό Οριακό Θεώρημα

Έστω $X_1, \ldots, X_n$ τυχαίες μεταβλητές με πεπερασμένη μέση τιμή μ και διασπορά σ^2. Τότε, σύμφωνα με το κεντρικό οριακό θεώρημα

$$\sqrt{n}(\bar{X} - \mu) \Rightarrow \mathcal{N}(0, \sigma^2),$$

κατά κατανομή, όταν το $n \to \infty$. Στο πιο πάνω, το $\mathcal{N}$ συμβολίζει την κανονική κατανομή. Η προσομοίωση μπορεί να βοηθήσει διαισθητικά στην κατανόηση του θεωρήματος.

Έστω $X_1, \ldots, X_{100}$ ανεξάρτητες και ισόνομες τυχαίες κατανομές από την Poisson με παράμετρο $\lambda = 1$. Τότε $\mu = \sigma^2 = 1$ και συνεπώς, από το κεντρικό οριακό θεώρημα

$$\mathrm{E}(\bar{X}) = 1$$

και

$$\mathrm{Var}(\bar{X}) = 1/100 = 0.01.$$

Στην R, η πιο κάτω συνάρτηση παράγει δείγματα από την ασυμπτωτική κατανομή της μέσης τιμής

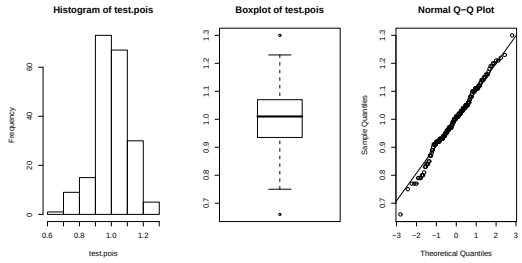
```
poisson.clt    <- function(k,n, parameter)
{
  samples.mean   <- rep(NA, k)
  for (i in 1:k)
  {
    samples.mean[i]   <- mean(rpois(n, lambda=parameter))
  }
  return(samples.mean)
}
```

Τρέχοντας αυτή την συνάρτηση παράγονται τα ακόλουθα αποτελέσματα:

```
> test.pois <- poisson.clt(200,100,1)
> summary(test.pois)
 Min. 1st Qu. Median  Mean 3rd Qu. Max.
 0.69    0.93      1 1.001   1.072 1.31
> var(test.pois)
[1] 0.009584601
> par(mfrow=c(1,3))
> hist(test.pois)
> boxplot(test.pois)
> qqnorm(test.pois)
> qqline(test.pois)
```

Δηλαδή, δημιουργούμε k δείγματα από την Poisson, και για καθένα από αυτά υπολογίζουμε το μέσο όρο τους, έστω, $\overline{X_1}, \ldots, \overline{X_k}$. Στο παράδειγμα $k = 200$ και

το δείγμα που παίρνουμε κάθε φορά έχει μέγεθος 100. Να εξηγήσετε λεπτομερώς κάθε βήμα του προγράμματος. Τα αριθμητικά αποτελέσματα μαζί με το Σχήμα 6.4 παρουσιάζουν εμπειρικά το κεντρικό οριακό θεώρημα.



Σχήμα 6.4: Κεντρικό οριακό θεώρημα για την κατανομή Poisson.

Πρέπει να σημειωθεί ότι η συνάρτηση `poisson.clt` δεν είναι και ο πιο αποτελεσματικός τρόπος προγραμματισμού, αλλά παρουσιάζει την γενική ιδέα πίσω από τους υπολογισμούς. Για πιο αποτελεσματική χρήση των βρόγχων σε σχέση με την μνήμη και τον υπολογιστικό χρόνο του υπολογιστή, η συνάρτηση `lapply` είναι καταλληλότερη.

6.3 Προσέγγιση της Διωνυμικής Κατανομής από την Κανονική και την Poisson

Σε αυτό το σημείο θα εξερευνηθεί το πως προσεγγίζεται η διωνυμική κατανομή με τη βοήθεια του κεντρικού οριακού θεωρήματος αλλά και από την κατανομή Poisson. Έστω η διωνυμική κατανομή $Bin(n, p)$. Από τη θεωρεία είναι γνωστό ότι αυτή προσεγγίζεται από

- την κατανομή Poisson όταν το n είναι μεγάλο και το p είναι μικρό και
- την κανονική κατανομή όταν το n είναι μεγάλο.

Στην προσομοίωση που ακολουθεί θα δούμε πόσο καλή είναι η προσέγγιση για διάφορες τιμές του n και του p. Στην αρχή επιλέγονται οι τιμές για το n και το p να είναι:

```
p<-c(0.01,0.1,0.3,0.5)
n<-c(10,100,1000,10000)
```

Για κάθε συνδυασμό (n, p) θα συγκριθούν τρείς κατανομές, η διωνυμική κατανομή, η προσέγγιση από την Poisson, και η προσέγγιση από την κανονική. Για καλύτερη εποπτική ανάλυση θα παρασταθεί γραφικά η συνάρτηση πυκνότητας πιθανότητάς τους ταυτόχρονα στο ίδιο γράφημα. Θα κατασκευαστεί ένα γράφημα για κάθε συνδυασμό (n, p).

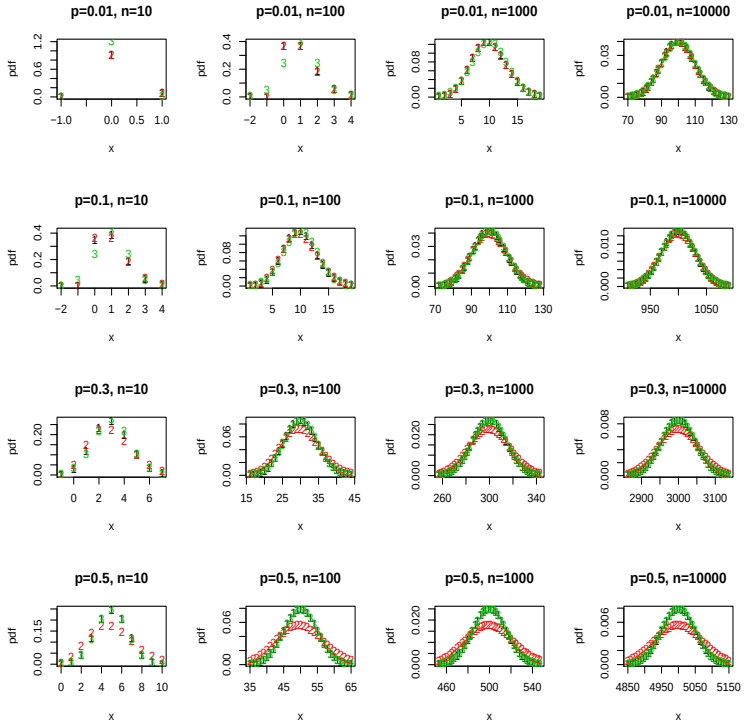
```
par (mfrow=c(4,4))
for (i in 1:4)
{
    for (j in 1:4)
    {
        mu <- n[j]*p[i]
        sd <- sqrt(mu * (1-p[i]))
        lo <- round(mu-3*sd)
        hi <- round(mu+3*sd)
        if (hi-lo<40)
        x <- seq(lo,hi,by=1) else
        x <- round(seq(lo,hi,len=40))
        pdf <- cbind(dbinom(x,n[j],p[i]),dpois(x,mu),dnorm(x,mu,sd))
        pdf[x<0,1:2]<- 0
        matplot(x,pdf,main=paste("p=", p[i],", n=",n[j],sep=""))
    }
}
```

Στο πιο πάνω πρόγραμμα, `pdf` είναι ένας πίνακας με τρεις στήλες οι οποίες περιέχουν τις συναρτήσεις πυκνότητας πιθανότητας πιθανότητας της διωνυμικής, της Poisson και της κανονικής, αντίστοιχα. Η εντολή `matplot` κατασκευάζει γράφημα πίνακα. Παρουσιάζει τη γραφική παράσταση της κάθε στήλης συναρτήσει του x. Η πρώτη στήλη αναπαρίσταται με το σύμβολο ¨1¨, η δεύτερη με το σύμβολο ¨2¨ και η τρίτη με το σύμβολο ¨3¨ (Σχήμα 6.5).

Πρώτα θα εξεταστεί στο γράφημα η προσέγγιση από την Poisson. Σε κάθε στήλη, η προσέγγιση από την Poisson είναι μεγαλύτερη στο πάνω μέρος της στήλης όπου το p είναι μικρό, και σταδιακά γίνεται ασθενέστερη όσο το p μεγαλώνει (πηγαίνοντας προς τα κάτω). Ο λόγος είναι ότι η μέση τιμή της διωνυμικής κατανομής είναι np και η διακύμανση $np(1-p)$. Παρόλο που η μέση τιμή και η διακύμανση δεν είναι ίσες, προσεγγιστικά γίνονται ίσα όταν το p είναι πολύ μικρό. Για την Poisson με παράμετρο λ, η μέση τιμή και η διακύμανση είναι ίση με λ. Συνεπώς, η κατανομή Poisson δεν μπορεί να είναι καλή προσέγγιση μιας κατανομής της

οποίας η μέση τιμή και η διακύμανση είναι πολύ διαφορετικές μεταξύ τους και έτσι η προσέγγιση είναι καλή για τη διωνυμική μόνο όταν το p είναι μικρό.

Στην περίπτωση της προσέγγισης από την κανονική κατανομή, η προσέγγιση δεν είναι καλή στο πάνω αριστερό κομμάτι του γραφήματος. Συγκεκριμένα, όταν $(p = .01, n = 100)$ ή $(p = .1, n = 10)$, η προσέγγιση από την κανονική επιτρέπει στο x να είναι αρνητικό ενώ η διωνυμική (και η Poisson) κατανομή απαιτεί το x να μην παίρνει αρνητικές τιμές. Σε κάθε γραμμή, η προσέγγιση από την κανονική γίνεται καλύτερη όσο προχωρούμε διαμέσου της γραμμής. Αυτό είναι το αποτέλεσμα από το κεντρικό οριακό θεώρημα. Επίσης, σε κάθε στήλη, η προσέγγιση από την κανονική γίνεται καλύτερη όσο προχωρούμε προς τα κάτω. Αυτό συμβαίνει επειδή η διωνυμική κατανομή είναι συμμετρική όταν $p = 0.5$, δηλαδή έχει μορφή παρόμοια με την κανονική, και συνεπώς δεν χρειάζεται μεγάλο n για να είναι είναι καλή η προσέγγιση από την κανονική. Σε αντίθεση, η διωνυμική με $p = 0.01$ είναι πολύ λοξή, δηλαδή δεν μοιάζει με την κανονική, και συνεπώς χρειάζεται μεγάλο n για να γίνει καλή η προσέγγιση από την κανονική.



Σχήμα 6.5: Προσέγγιση της διωνυμικής από την Poisson και την κανονική.

6.4 Monte Carlo Ολοκλήρωση

Έστω ότι είναι αναγκαίο να εκτιμηθεί η αναμενόμενη τιμή της τυχαίας μεταβλητής X με συνάρτηση πυκνότητας πιθανότητας $f(x)$ δεδομένου ότι αυτή υπάρχει. Συγκεκριμένα, έστω ότι το δ μπορεί να οριστεί από

$$\delta = \int c(x)f(x)dx = \mathrm{E}_f\left[c(X)\right],$$

και υποθέστε ότι υπάρχει και είναι πεπερασμένο. Υπάρχουν διάφοροι μέθοδοι για υπολογισμό του δ και ίσως η πιο γνωστές ανάμεσα στους στατιστικούς είναι αυτές που βασίζονται στη Monte Carlo ολοκλήρωση. Ανάλογα, αν $X_1, \ldots, X_n$ τυχαίο δείγμα από την $f(x)$, τότε σύμφωνα με τον ασθενή νόμο των μεγάλων αριθμών, η εκτιμήτρια

$$\hat{\delta} = \frac{1}{n}\sum_{i=1}^{n} c(X_i)$$

προσεγγίζει το δ με μεγάλη πιθανότητα όταν το n τείνει στο άπειρο.

Παρατίθεται πως μπορεί η R να χρησιμοποιηθεί για να υπολογίσει τέτοια ολοκληρώματα. Έστω ότι η X είναι τυχαία μεταβλητή που ακολουθεί τη κατανομή Beta με συνάρτηση πυκνότητας πιθανότητας

$$f(x) = \frac{1}{B(\alpha,\beta)}x^{\alpha-1}(1-x)^{\beta-1}$$

για $x \in (0,1)$ και έστω ότι είναι αναγκαίο να υπολογιστούν τα ακόλουθα δύο ολοκληρώματα:

$$\delta_1 = \int_{0.2}^{0.4} f(x)dx = \int I_{[0.2,0.4]}f(x)dx,$$

όπου I είναι η δείκτρια συνάρτηση, και

$$\delta_2 = \int \sin(x)e^{-x}f(x)dx$$

όπου υποθέτουμε ότι η f είναι η συνάρτηση πυκνότητας πιθανότητας της κατανομής Βήτα με παράμετρους 2.5 και 5. Τότε η ακόλουθη συνάρτηση δίνει τα επιθυμητά αποτελέσματα σε μορφή πίνακα.

```
estim.beta  <- function(k)
{
delta1      <- rep(NA, k)
delta2      <- rep(NA, k)
for (i in 1:k)
```

```
{
x   <- rbeta(500, shape1=2.5, shape2=5)
delta1[i] <- mean(as.numeric(I(0.2 <x & x<0.4)))
delta2[i] <- mean(sin(x)*exp(-x))
}
return(cbind(delta1,delta2))
}
```

Να εξηγήσετε λεπτομερώς τι κάνει κάθε βήμα στην πιο πάνω συνάρτηση. Τρέχοντας τη συνάρτηση παίρνονται τα ακόλουθα

```
> delta.estim <- estim.beta(500)
> apply(delta.estim,2,mean)
 delta1    delta2
 0.23072 0.2172346
> apply(delta.estim,2,var)
    delta1        delta2
 0.0003627351 9.412981e-006
> sqrt(apply(delta.estim,2,var))
   delta1      delta2
 0.01904561 0.003068058
```

Συνεπώς, το δ_1 εκτιμάται να είναι ίσο με 0.03086395 με τυπικό σφάλμα 0.0027, ενώ το δ_2 εκτιμάται να είναι ίσο με 0.08337394 με τυπικό σφάλμα 0.0025.

Παρόλο που εδώ υπάρχουν διάφορες συνηθισμένες μέθοδοι για να παραχθούν ψευδο-τυχαία αποτελέσματα για αρκετές κατανομές, συνήθως ένα τέτοιο εγχείρημα μπορεί να είναι αρκετά απαιτητικό εξαιτίας της μορφής της συνάρτησης πυκνότητας πιθανότητας. Για αυτό το λόγο κάποιες άλλες τεχνικές μπορούν να χρησιμοποιηθούν εναλλακτικά και μια από τις πιο δημοφιλείς μεθόδους προσομοίωσης είναι η *importance sampling* η εφαρμογή της οποίας παρουσιάζεται πιο κάτω:

- Γέννηση $Z_1, \ldots, Z_n$ ανεξάρτητων και ισόνομων τυχαίων μεταβλητών με συνάρτηση πυκνότητα πιθανότητας $g(z)$ των οποίων ο φορέας, έστω A, περιέχει το φορέα της $f(.)$.

- Αφού παρατηρηθεί ότι

$$\delta \quad = \quad \int c(z) \frac{f(z)}{g(z)} g(z) dz$$

$$= \int c(z)w(z)g(z)dz = \mathrm{E}_g\left[c(Z)w(Z)\right],$$

με $w = f/g$, κατασκευάζεται η ακόλουθη εκτιμήτρια

$$\tilde{\delta} = \frac{1}{n}\sum_{i=1}^{n} c(Z_i)w(Z_i).$$

Προφανώς, η εκτιμήτρια $\tilde{\delta}$ μιμείται την εκτιμήτρια $\hat{\delta}$ στην έννοια ότι η αναμενόμενη τιμή σε σχέση με το X ακολουθώντας την f, αντικαθίσταται από την αντίστοιχη αναμενόμενη τιμή σε σχέση με το Z, ακολουθώντας την g. Θα ήταν εκπαιδευτικό σκόπιμο να προσπαθήσει ο αναγνώστης να χρησιμοποιήσει την R για να υπολογίσει το $\tilde{\delta}$ για το πιο πάνω παράδειγμα όταν η g είναι η συνάρτηση πυκνότητας πιθανότητας από την ομοιόμορφη κατανομή.

6.5 Βελόνα του Buffon

Ένα τραπέζι χωρίζεται σε παράλληλες ευθείες οι οποίες απέχουν d μονάδες μεταξύ τους. Ρίχνουμε μία βελόνα μήκους l στο τραπέζι (με $l \leq d$) n φορές και μετράμε R τον αριθμό των φορών που η βελόνα τέμνει μία ευθεία. Έστω X η απόσταση από το κέντρο της βελόνας στην πιο κοντινή παράλληλη ευθεία και θ η γωνία που σχηματίζει η κάθετη ευθεία από το κέντρο της βελόνας στην πιο κοντινή παράλληλη ευθεία. Τότε η βελόνα θα τέμνει μία από τις παράλληλες ευθείες αν και μόνο αν

$$\frac{x}{\cos\theta} \leq \frac{l}{2}.$$

Όμως, αφού η X μεταβάλλεται μεταξύ 0 και $d/2$ και η θ είναι μεταξύ 0 και $\pi/2$, μπορούμε να υποθέσουμε ότι είναι ανεξάρτητες τυχαίες μεταβλητές από την ομοιόμορφη. Συνεπώς, έχουμε

$$\begin{aligned} P(X \leq \frac{l\cos\theta}{2}) &= \frac{4}{\pi d}\int_0^{\pi/2}\int_0^{l\cos y/2} dxdy \\ &= \frac{4}{\pi d}\int_0^{\pi/2}\frac{l\cos y}{2}dy \\ &= \frac{2l}{\pi d}. \end{aligned}$$

Συνεπώς, αν $\rho = l/d$, και $\phi = 1/\pi$, μία εκτιμήτρια του π δίνεται από

$$\hat{\pi}_0 = \frac{1}{\hat{\phi}_0} = \frac{2\rho}{\hat{p}}$$

όπου $\hat{p} = R/n$.

Μία βασική ερώτηση είναι πώς να βελτιστοποιήσουμε τις τιμές των l και d για να ελαχιστοποιήσουμε την διακύμανση της εκτιμήτριας $\hat{\phi}_0$. Επειδή η R είναι διωνυμική τυχαία μεταβλητή έχουμε ότι $\text{Var}(\hat{p}) = p(1-p)/n$. Συνεπώς, $\text{Var}(\hat{\phi}_0) = 2\rho\phi(1-2\rho\phi)/4\rho^2 n$ και αυτή η ποσότητα ελαχιστοποιείται αν $\rho = 1$ ή $l = d$.

Το παρακάτω πρόγραμμα δίνει ακριβώς την παραπάνω μεθοδολογία.

```
buf<-function(n, d,l)  # n is the number of simulations,
                       #d is the distance between the lines
{                      # and l is the needle's length (l =< d).

       R             <- rep(NA, n)
       x             <- runif(n, 0, d/2)
       theta         <- runif(n, 0, pi/2)
       y             <- (l/2)*cos(theta)
       R             <- ifelse(y > x,1,0)
       pi            <- cumsum(R)/(1:n)
       rho           <- l/d
       phi           <- pi/(2*rho)
       pi.hat        <- 1/phi
       pi.hat
       plot(1:n, pi.hat, type="l", xlab="Number of Simulations",
            ylab="Proportion of Hits")
}
```

6.6 Εμπειρική Σύγκριση Εκτιμητριών

Η προσομοίωση μπορεί να χρησιμοποιηθεί για εμπειρική σύγκριση διαφόρων εκτιμητρίων οι οποίες χρησιμοποιούνται για την εκτίμηση συγκεκριμένης παραμέτρου. Ας υποθέσουμε ότι στο Πανεπιστήμιο φοιτούν 3000 φοιτητές εκ των οποίων το 30% είναι μέλη συνδικαλιστικών οργανώσεων ενώ το άλλο 70% είναι ανεξάρτητοι. Υπάρχει μία μελλοντική εκλογή προέδρου των φοιτητών και ας υποθέσουμε ότι δύο ανεξάρτητοι υποψήφιοι, ο Α και Β, διεκδικούν την εκλογή. Έστω

$$\begin{aligned} \theta_U &= \text{ποσοστό μη ανεξάρτητων φοιτητών που υποστηρίζουν τον Α} \\ \theta_I &= \text{ποσοστό ανεξάρτητων φοιτητών που υποστηρίζουν τον Α} \end{aligned}$$

θ = ποσοστό φοιτητών που υποστηρίζουν τον Α

Γίνεται δειγματοληπτική έρευνα σε 100 φοιτητές για την εκτίμηση της παραμέτρου θ και προτείνονται 3 διαφορετικές μεθόδοι εκτίμησης.

1. Τυχαία επιλογή 100 φοιτητών και εκτίμηση μέσω της στατιστικής συνάρτησης

$$\hat{\theta}_1 = \text{ποσοστό φοιτητών που υποστηρίζουν τον Α.}$$

2. Τυχαία επιλογή 100 φοιτητών και εκτίμηση μέσω της στατιστικών συνάρτησεων

$\hat{\theta}_U$ = ποσοστό μη ανεξάρτητων φοιτητών που υποστηρίζουν τον Α στο δείγμα
$\hat{\theta}_I$ = ποσοστό ανεξάρτητων φοιτητών που υποστηρίζουν τον Α στο δείγμα
$\hat{\theta}_2 = 0.30\hat{\theta}_U + 0.70\hat{\theta}_I$

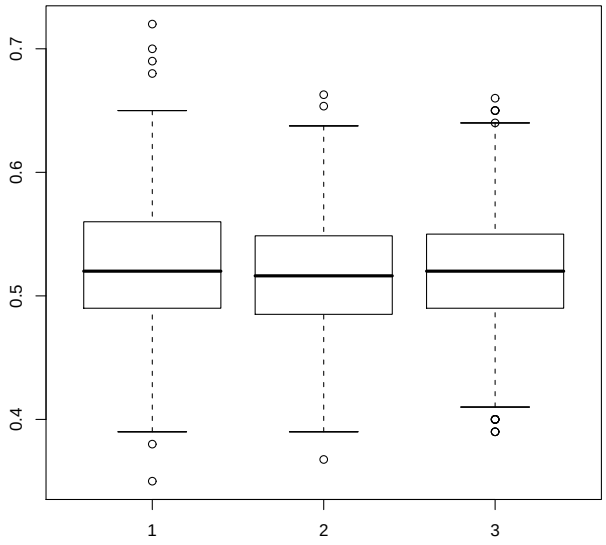
3. Επιλογή 30 μη ανεξάρτητων και 70 ανέξαρτητων. Τότε

$\hat{\theta}_U$ = ποσοστό από μη ανεξάρτητους φοιτητές που υποστηρίζουν τον Α
$\hat{\theta}_I$ = ποσοστό από ανεξάρτητους φοιτητές που υποστηρίζουν τον Α
$\hat{\theta}_3 = 0.30\hat{\theta}_U + 0.70\hat{\theta}_I$

Ποια διαδικασία είναι η καλύτερη; Αν και μπορούμε να απαντήσουμε θεωρητικά στην ερώτηση, εδώ θα δούμε πώς μπορεί να μας βοηθήσει η προσομοίωση. Αρκεί να επαναλάβουμε κάθε διαδικασία αρκετές φορές και μετά να εξετάσουμε πόσο ακριβή είναι τα αποτελέσματα. Πρέπει να επιλέξουμε τις αληθινές τιμές των παραμέτρων φυσικά και η όλη θεωρία προσομοιώνεται με τον παρακάτω κώδικα. Τα αποτελέσματα δίνουν την γραφική παράσταση 6.6.

```
# choose "true" theta.u and theta.i
theta.u <- .8
theta.i <- .4
prop.u <- .3
prop.i <- 1 - prop.u
theta <- prop.u * theta.u + prop.i * theta.i

sim.1 <- function() {
x <- rbinom(1,sampsize,theta)
```

```
return ( x / sampsize )
}

sim.2 <- function() {
n.u <- rbinom ( 1, sampsize, prop.u )
n.i <- sampsize - n.u
x.u <- rbinom ( 1, n.u, theta.u )
x.i <- rbinom ( 1, n.i, theta.i )
t.hat.u <- x.u / n.u
t.hat.i <- x.i / n.i
return ( prop.u * t.hat.u + (1-prop.u) * t.hat.i )
}

sim.3 <- function() {
n.u <- sampsize * prop.u
n.i <- sampsize * prop.i
x.u <- rbinom ( 1, n.u, theta.u )
x.i <- rbinom ( 1, n.i, theta.i )
t.hat.u <- x.u / n.u
t.hat.i <- x.i / n.i
return ( prop.u * t.hat.u + (1-prop.u) * t.hat.i )
}

sampsize <- 100
n.times <- 1000 # should be enough
theta.hat <- matrix ( NA, n.times, 3 )
for ( i in 1:n.times ) {
theta.hat[i,1] <- sim.1()
theta.hat[i,2] <- sim.2()
theta.hat[i,3] <- sim.3()
}

print ( apply(theta.hat,2,mean) )

[1] 0.5228600 0.5178319 0.5189100

boxplot ( theta.hat ~ col(theta.hat) )
```



Σχήμα 6.6: 1000 προσομοιώσεις της $\hat{\theta}$ για τρεις διαφορετικές μεθόδους εκτίμησης.

Κεφάλαιο 7

Στατιστική Συμπερασματολογία

Στο κεφάλαιο αυτό γίνεται μια εισαγωγή σε μερικές απλές στατιστικές μεθόδους σε προβλήματα συμπερασματολογίας σε ένα και δύο δείγματα. Το πρώτο μέρος αναφέρεται στην Περιγραφική Στατιστική η οποία είναι μια συλλογή από γραφικές μεθόδους για εξερεύνηση και κατανόηση των δεδομένων. Στη συνέχεια θα γίνει αναφορά σε στατιστικούς ελέγχους που χρησιμοποιούνται κατά γενικό κανόνα, όπως τους ελέγχους t και Wilcoxon για να εξαχθούν στατιστικά συμπεράσματα για τις υπό διερεύνηση παραμέτρους. Προχωρώντας, παρουσιάζονται περαιτέρω έλεγχοι υποθέσεων όπως ο έλεγχος καλής προσαρμογής και ο έλεγχος για ποσοστά. Στο τέλος, εξετάζεται ο τρόπος χειρισμού πινάκων συνάφειας για εξαγωγή συμπεράσματος για τη συσχέτιση μεταξύ δυο μεταβλητών όταν ταξινομηθούν σε σχέση μιας τρίτης.

7.1 Περιγραφική Στατιστική

Η Περιγραφική Στατιστική χρησιμοποιεί γραφήματα τα οποία βοηθούν στο να εξερευνηθεί αν ισχύουν οι υποθέσεις των στατιστικών μοντέλων. Μερικές ερωτήσεις που ενδιαφέρουν σε αυτές τις περιπτώσεις είναι:

- Τα δεδομένα ακολουθούν την κανονική κατανομή;
- Υπάρχουν απομακρυσμένες τιμές στα δεδομένα;

- Αν τα δεδομένα συγκεντρώθηκαν διαδοχικά στον χρόνο (χρονοσειρές), υπάρχει ένδειξη σειριακή συσχέτισης;

Τα κύρια γραφήματα τα οποία βοηθούν στην εποπτική εξερεύνηση των δεδομένων είναι τα ακόλουθα:

- ιστογράμματα (histograms),
- κυτιογραφήματα (boxplots),
- γράφημα πυκνότητας (density plots),
- QQ plots (quantile-quantile plots).

Ακολουθεί μια απλή συνάρτηση στην R η οποία κατασκευάζει αυτά τα τέσσερα γραφήματα:

```
eda.shape <- function(x)
 {
   par(mfrow = c(2, 2))
   hist(x)
   boxplot(x)
   iqd <- summary(x)[5] - summary(x)[2]
   plot(density(x,width=2*iqd), xlab = "x",ylab = "", type = "l")
   qqnorm(x)
   qqline(x)
 }
```

Με την εντολή `hist()` κατασκευάζεται το ιστόγραμμα και με την εντολή `boxplot` το κυτιόγραμμα. Η εντολή `summary()` δίνει την περίληψη πέντε αριθμών και άρα το `iqd` στη συνάρτηση υπολογίζει το ενδοτεταρτημοριακό εύρος (interquartile range). Η εντολή `plot(density())` κατασκευάζει το γράφημα μη παραμετρικής εκτιμήτριας συνάρτησης πυκνότητας πιθανότητας με την μέθοδο των πυρήνων και οι εντολές `qqnorn` και `qqline` δημιουργούν το QQ plot μαζί με την γραμμή για απόκλιση από την κανονική κατανομή.

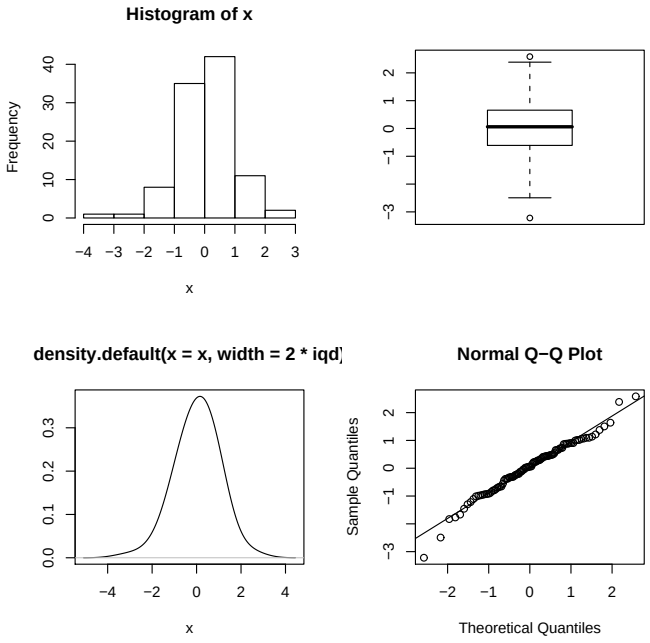
Για να εξερευνηθεί εποπτικά αν υπάρχει σειριακή συσχέτιση είναι χρήσιμο να γίνει στην αρχή η γραφική παράσταση χρονοσειρών των δεδομένων. Η ακόλουθη συνάρτηση χειρίζεται αυτήν την περίπτωση:

```
eda.ts <- function(x)
```

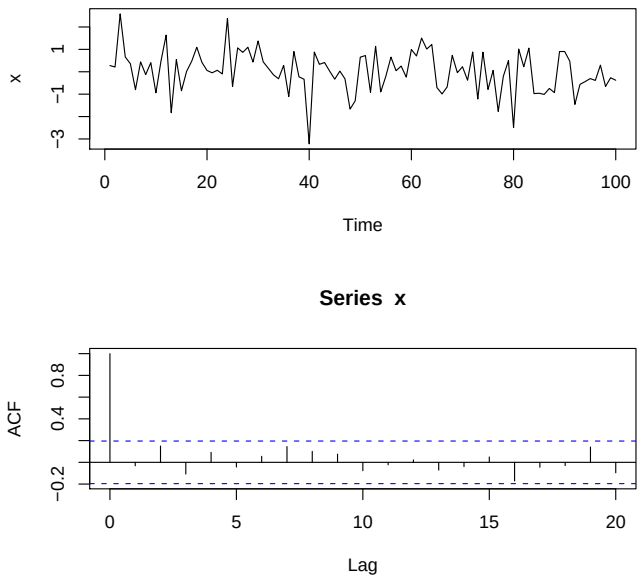
```
{
   par(mfrow=c(2,1))
   ts.plot(x)
   acf(x)
   invisible()
}
```

Η εντολή `ts.plot` παράγει γράφημα των διαδοχικών παρατηρήσεων ενώ η εντολή `acf` δίνει την δειγματική συνάρτηση αυτοσυσχέτισης. Οι δυο πιο πάνω συναρτήσεις εφαρμόζονται σε 100 τιμές από την κανονική και τα αποτελέσματα παρουσιάζονται στα σχήματα 7.1 και 7.2, αντίστοιχα.

```
x <- rnorm(100)
eda.shape(x)
eda.ts(x)
```



Σχήμα 7.1: Γραφήματα για τη μορφή των δεδομένων.

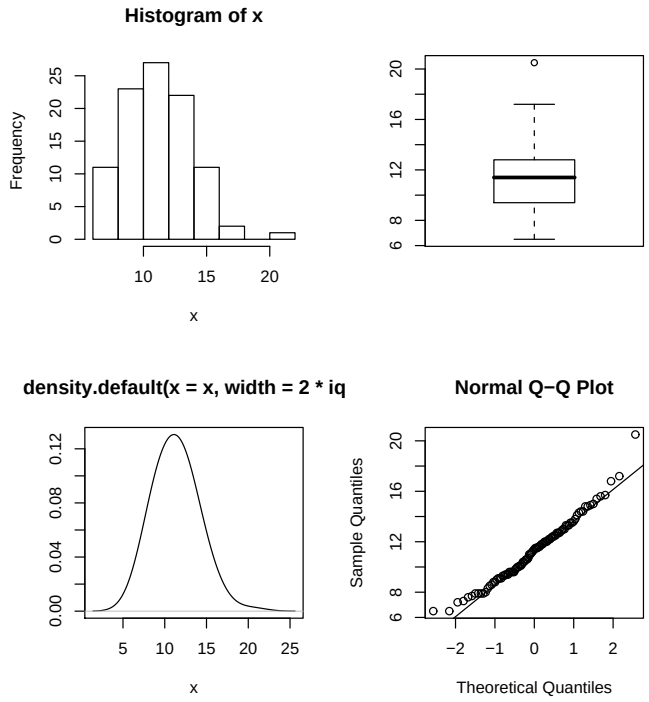


Σχήμα 7.2: Γραφήματα για σειριακή συσχέτιση.

7.2 Συμπερασματολογία για Ένα Δείγμα

Για το κομμάτι αυτό του κεφαλαίου θα χρησιμοποιηθεί το πλαίσιο δεδομένων `cats`, το οποίο αποτελείται από παρατηρήσεις του βάρους του σώματος και της καρδιάς 140 αρσενικών και θηλυκών γάτων οι οποίες χρησιμοποιήθηκαν για ένα πείραμα. Το συγκεκριμένο πλαίσιο βρίσκεται στη βιβλιοθήκη της R `MASS` (κατασκευάστηκε από τους Venables και Ripley)

```
> library(MASS)
> cats
> male   <-  cats[cats$Sex=="M",]$Hwt
> female <-  cats[cats$Sex=="F",]$Hwt
> eda.shape(male)
```



Σχήμα 7.3: Γραφήματα για τη μορφή των δεδομένων για το βάρος της καρδιάς των αρσενικών.

```
> t.test(male, mu=10)

        One Sample t-test

data:  male
t = 5.1241, df = 96, p-value = 1.544e-06
alternative hypothesis: true mean is not equal to 10
95 percent confidence interval:
 10.81030 11.83506
sample estimates:
mean of x
 11.32268

> t.test(male, mu=10, conf.level=0.99)
```

```
        One Sample t-test

data:  male
t = 5.1241, df = 96, p-value = 1.544e-06
alternative hypothesis: true mean is not equal to 10
99 percent confidence interval:
 10.64431 12.00105
sample estimates:
mean of x
 11.32268
```

Οι εντολές `cats[cats$Sex=="M",]$Hwt` και `cats[cats$Sex=="F",]$Hwt` χρησιμοποιήθηκαν για να εξάγουν το βάρος της καρδιάς των αρσενικών και θηλυκών γάτων, αντίστοιχα. Από τα γραφήματα για τη μορφή των παρατηρήσεων για το βάρος της καρδιάς των αρσενικών γάτων συμπεραίνεται ότι είναι λογικό να γίνει η υπόθεση ότι ακολουθούν την κανονική κατανομή με μέση τιμή περίπου ίση με 11 (βλ. Σχήμα 7.3). Η συνάρτηση `t.test` μπορεί να χρησιμοποιηθεί για να γίνει ο έλεγχος t με μηδενική υπόθεση $\mu = \mu_0$ και στατιστική συνάρτηση

$$t = \frac{\overline{x} - \mu_0}{s/\sqrt{n}}.$$

Το αποτέλεσμα δείχνει ότι για $\mu_0 = 10$, η μηδενική υπόθεση απορρίπτεται (το `p-value` είναι πολύ κοντά στο 0 και άρα μικρότερο από 0.05). Πρέπει να σημειωθεί ότι εξ ορισμού το αποτέλεσμα της συνάρτησης δίνει το 95% διαστήματα εμπιστοσύνης. Είναι δυνατόν όμως να δοθεί το διάστημα εμπιστοσύνης για οποιοδήποτε επίπεδο εμπιστοσύνης $(1 - \alpha)\%$ δίνοντας τιμή στο όρισμα `conf.level`. Για να γίνει ο απαραμετρικός έλεγχος Wilcoxon για τη μέση τιμή χρησιμοποιείται η συνάρτηση `wilcox.test` όπως πιο κάτω:

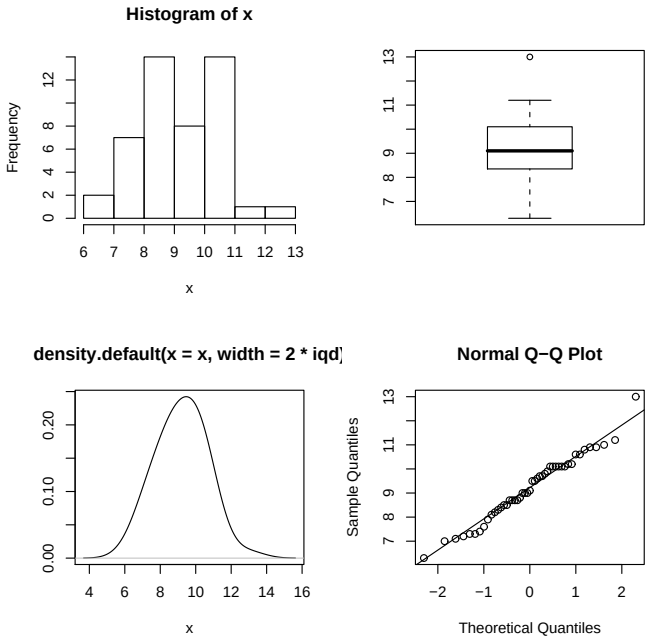
```
> wilcox.test(male, mu=10)

        Wilcoxon signed rank test with continuity correction

data:  male
V = 3491.5, p-value = 6.934e-06
alternative hypothesis: true location is not equal to 10
```

Η μηδενική υπόθεση απορρίπτεται και σε αυτήν την περίπτωση και συνεπώς η μέση τιμή μ διαφέρει στατιστικώς σημαντικά από την τιμή 10.

7.3 Συμερασματολογία για Δυο Δείγματα

Ας εξεταστεί τώρα ο έλεγχος της διαφοράς δύο μέσων τιμών βασισμένος σε πρόβλημα δύο δειγμάτων. Για παράδειγμα, έστω ότι είναι ανάγκη να εξεταστεί η διαφορά μεταξύ της μέσης τιμής του βάρους της καρδιάς των αρσενικών και των θηλυκών γάτων από το πλαίσιο δεδομένων που χρησιμοποιήθηκε πιο πάνω.

```
> eda.shape(female)
```



Σχήμα 7.4: Γραφήματα για τη μορφή των δεδομένων για το βάρος της καρδιάς των θηλυκών.

```
> var.test(male,female)

        F test to compare two variances

data:  male and female
F = 3.5064, num df = 96, denom df = 46, p-value = 8.159e-06
alternative hypothesis: true ratio of variances is not equal to 1
```

```
95 percent confidence interval:
 2.075412 5.664645
sample estimates:
ratio of variances
           3.50642


> t.test(male,female,var.equal=T)

        Two Sample t-test

data:  male and female
t = 5.3539, df = 142, p-value = 3.38e-07
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.337588 2.903517
sample estimates:
mean of x mean of y
11.322680  9.202128

> t.test(male,female,var.equal=F)

        Welch Two Sample t-test

data:  male and female
t = 6.5179, df = 140.608, p-value = 1.186e-09
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.477352 2.763753
sample estimates:
mean of x mean of y
11.322680  9.202128

> wilcox.test(male,female)

        Wilcoxon rank sum test with continuity correction
```

```
data:  male and female
W = 3460.5, p-value = 4.882e-07
alternative hypothesis: true location shift is not equal to 0
```

Από το γράφημα στο Σχήμα 7.3 συμπεραίνεται ότι τα δεδομένα για το βάρος της καρδιάς των θηλυκών δεν εφαρμόζουν στην κανονική κατανομή. Ο πρώτος έλεγχος που γίνεται στο πιο πάνω ελέγχει την ισότητα των διακυμάνσεων των δύο δειγμάτων (μηδενική υπόθεση) με την στατιστική συνάρτηση

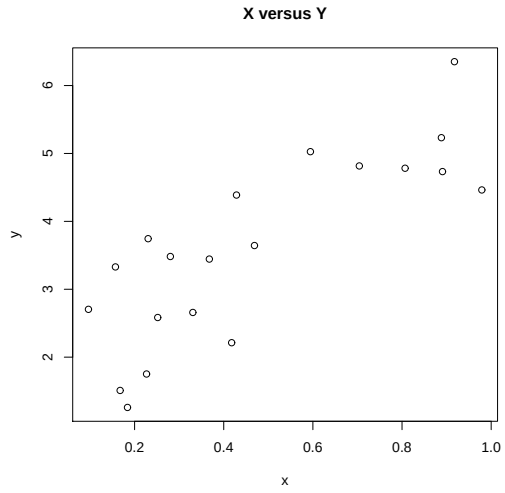
$$F = \frac{S_x^2}{S_y^2}$$

και δείχνει ότι η ισότητα απορρίπτεται. Συνεπώς, ο συνηθισμένος έλεγχος t για τη διαφορά των μέσων τιμών δεν μπορεί να εφαρμοστεί σε αυτήν την περίπτωση. Ωστόσο, στη συνάρτηση `t.test` μπορεί να προστεθεί το όρισμα `var.equal` στο οποίο δηλώνεται αν ισχύει η ισότητα των διακυμάνσεων με τις τιμές ¨TRUE¨ και ¨FALSE¨. Με αυτόν το τρόπο είναι δυνατό να γίνει ο έλεγχος t και στο πιο πάνω παράδειγμα συμπεραίνεται ότι η υπόθεση ισότητας των μέσων τιμών των δυο δειγμάτων απορρίπτεται. Το ίδιο συμπέρασμα βγαίνει και από τον απαραμετρικό έλεγχο Wilcoxon.

7.4 Συμπερασματολογία για Δείγματα Κατά Ζεύγη

Πολλές φορές τα δεδομένα παρουσιάζονται σε ζεύγη της μορφής (X_i, Y_i). Για να γίνει ο έλεγχος t σε αυτήν την περίπτωση, είναι χρήσιμο να οριστεί μια καινούργια μεταβλητή Z με τιμές $Z_i = X_i - Y_i$ και μετά να εργαστούμε όπως πιο πάνω. Για να μελετηθεί η συσχέτιση μεταξύ των δυο μεταβλητών είναι πάντα χρήσιμο να γίνει το γράφημα διασποράς τους για να εξεταστεί γραφικά η σχέση τους. Στη συνέχεια γίνεται ο έλεγχος συσχέτισης με την εντολή `cor.test()` για να ελεγχθεί αν ο συντελεστής συσχέτισης είναι μεγαλύτερος, μικρότερος ή διαφορετικός από 0. Ακολουθεί ένα απλό παράδειγμα.

```
> x <- runif(20)
> y <- 2+3*x+rnorm(20)
> plot(x,y,main="X versus Y")


> cor.test(x,y)
```



Σχήμα 7.5: Διάγραμμα διασποράς του y συναρτήσει του x.

```
        Pearson's product-moment correlation

data:  x and y
t = 5.9907, df = 18, p-value = 1.149e-05
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5847064 0.9246687
sample estimates:
      cor
0.8160727

> cor.test(x,y, alt="l")

        Pearson's product-moment correlation

data:  x and y
t = 5.9907, df = 18, p-value = 1
alternative hypothesis: true correlation is less than 0
95 percent confidence interval:
 -1.0000000  0.9127702
```

```
sample estimates:
      cor
0.8160727
```

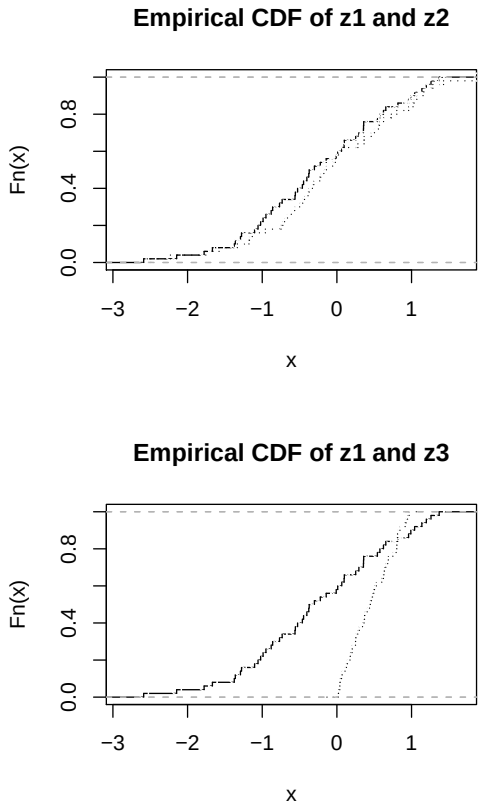
Όπως φαίνεται από το Σχήμα 7.5, υπάρχει μεγάλη υψηλή συσχέτιση μεταξύ του x και του y και άρα αναμένεται ο συντελεστής συσχέτισης να είναι κοντά στο 1. Για τον λόγο αυτό, ο πρώτος έλεγχος, με εναλλακτική υπόθεση η συσχέτιση να είναι διαφορετική από 0, δεν απορρίπτεται ενώ ο δεύτερος έλεγχος, με εναλλακτική υπόθεση η συσχέτιση να είναι αρνητική, απορρίπτεται.

7.5 Έλεγχος Καλής Προσαρμογής

Ο έλεγχος καλής προσαρμογής (goodness of fit test) θεωρείται ως ακόμη ένας τρόπος για να εξακριβωθούν οι υποθέσεις της κατανομής των δεδομένων. Η R αξιολογεί την ¨καλή προσαρμογή¨ των δεδομένων με τον έλεγχο Kolmogorov-Smirnov. Ο έλεγχος Kolmogorov-Smirnov χρησιμοποιείται στην περίπτωση σύγκρισης ενός δείγματος με μια κατανομή, αλλά και στην περίπτωση σύγκρισης της κατανομής δυο δειγμάτων μεταξύ τους. Στην πρώτη περίπτωση, η μηδενική υπόθεση H_0 δηλώνει ότι τα δεδομένα ακολουθούν την συγκεκριμένη κατανομή, ενώ στη δεύτερη δηλώνει ότι τα δυο δείγματα ακολουθούν την ίδια κατανομή. Η σύγκριση των κατανομών δυο δειγμάτων μπορεί να γίνει και γραφικά με τη βοήθεια της γραφικής παράστασης της εμπειρικής κατανομής τους. Πιο κάτω παρουσιάζεται ένα παράδειγμα στο οποίο συγκρίνεται γραφικά η κατανομή δύο δειγμάτων από την κανονική και ένα από την ομοιόμορφη κατανομή (Σχήμα 7.6). Στο πάνω γράφημα συγκρίνονται τα δύο δείγματα από την ίδια κατανομή (κανονική), ενώ στο δεύτερο δυο δείγματα από διαφορετική κατανομή (κανονική και ομοιόμορφη) και η διαφορά τους είναι εμφανής.

```
> z1<-rnorm(50)
> z2<-rnorm(50)
> z3<-runif(50)
> par(mfrow=c(2,1))
> plot.ecdf(z1,verticals=TRUE, col.p="white",col.v="black",
+ main="Empirical CDF of z1 and z2")
> plot(ecdf(z2),add=TRUE,verticals=TRUE,col.p="white",col.v="black",
+ lty="dotted")
> plot.ecdf(z1,verticals=TRUE, col.p="white",col.v="black",
```

```
+ main="Empirical CDF of z1 and z3")
> plot(ecdf(z3),add=TRUE,verticals=TRUE,col.p="white",col.v="black",
+ lty="dotted")
```



Σχήμα 7.6: Σύγκριση γραφήματος εμπειρικής συνάρτησης κατανομής δύο δειγμάτων.

Ακολουθούν τα αποτελέσματα από τον έλεγχο Kolmogorov–Smirnov σε ένα και δύο δείγματα για το πιο πάνω παράδειγμα. Ο πρώτος έλεγχος συγκρίνει ένα δείγμα με την κανονική κατανομή, ο δεύτερος δυο όμοια δείγματα μεταξύ τους και ο τρίτος δυο διαφορετικά δείγματα μεταξύ τους. Στον πρώτο και δεύτερο έλεγχο η μηδενική υπόθεση δεν απορρίπτεται ενώ στον τρίτο απορρίπτεται, όπως αναμένεται.

```
> ks.test(z1,"pnorm")
```

```
        One-sample Kolmogorov-Smirnov test

data:  z1
D = 0.0973, p-value = 0.6946
alternative hypothesis: two-sided

> ks.test(z1,z2)

        Two-sample Kolmogorov-Smirnov test

data:  z1 and z2
D = 0.16, p-value = 0.5487
alternative hypothesis: two-sided

> ks.test(z1,z3)

        Two-sample Kolmogorov-Smirnov test

data:  z1 and z3
D = 0.46, p-value = 3.801e-05
alternative hypothesis: two-sided
```

Μια παραλλαγή του ελέγχου Kolmogorov–Smirnov είναι ο έλεγχος Anderson-Darling ο οποίος βρίσκεται στη βιβλιοθήκη `nortest` της R.

7.6 Έλεγχος Υποθέσεων για Ποσοστά

Η R παρέχει την εντολή `binom.test` για έλεγχο υποθέσεων για ποσοστά. Για παράδειγμα, έστω το πείραμα ρίψεως ενός νομίσματος 500 φορές με αποτέλεσμα εμφάνισης *κεφαλή* 226 φορές. Θα ελεγχθεί η υπόθεση αν η πιθανότητα να έρθει *κεφαλή* είναι ίση με $p = 0.5$.

```
> binom.test(226,500,p=0.5)

        Exact binomial test
```

```
data:  226 and 500
number of successes = 226, number of trials = 500, p-value = 0.03546
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.4077723 0.4967984
sample estimates:
probability of success
                0.452
```

Όπως φαίνεται, ο έλεγχος απορρίπτεται σε επίπεδο σημαντικότητας $\alpha = 0.05$, αλλά δεν απορρίπτεται για $\alpha = 0.01$. Επίσης, το διάστημα στο διάστημα εμπιστοσύνης δεν περιέχει την τιμή 0.5. Συνεπώς, η πιθανότητα να έρθει *κεφαλή* είναι στατιστικά μικρότερη από 0.5. Παρόμοια αποτελέσματα εξάγονται και για $p = 0.4$, συμπεραίνοντας ότι το p είναι μεγαλύτερο.

```
> binom.test(226,500,p=0.4)

        Exact binomial test

data:  226 and 500
number of successes = 226, number of trials = 500, p-value = 0.01983
alternative hypothesis: true probability of success is not equal to 0.4
95 percent confidence interval:
 0.4077723 0.4967984
sample estimates:
probability of success
                0.452
```

Με την εντολή `prop.test` γίνεται ο έλεγχος για ποσοστό $p = 0.5$ λαμβάνοντας υπόψη τη διόρθωση συνέχειας. Στο επόμενο παράδειγμα υποθέστε ότι εμφανίστηκε *κεφαλή* 266 φορές. Η μηδενική υπόθεση δεν απορρίπτεται με αποτέλεσμα η πιθανότητα να έρθει *κεφαλή* δεν διαφέρει στατιστικώς σημαντικά από 0.5.

```
> prop.test(266,500)

        1-sample proportions test with continuity correction

data:  266 out of 500, null probability 0.5
```

```
X-squared = 1.922, df = 1, p-value = 0.1656
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.4871883 0.5763127
sample estimates:
    p
0.532
```

Στη συνέχεια παρουσιάζεται πως μπορεί να γίνει ο προσημικός έλεγχος για ένα ή δυο δείγματα με τη βοήθεια της διαμέσου. Για ένα δείγμα ελέγχεται η υπόθεση αν η διάμεσος παίρνει μια συγκεκριμένη τιμή, ενώ για δυο δείγματα αν η διάμεσος της διαφοράς τους είναι διαφορετική από 0.

```
> x<-rnorm(100)
> y<-sum(x>0)
> binom.test(y,100)

        Exact binomial test

data:  y and 100
number of successes = 54, number of trials = 100, p-value = 0.4841
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.4374116 0.6401566
sample estimates:
probability of success
                  0.54

> z<-rnorm(100)
> d<-x-z
> binom.test(sum(d>0),length(d))

        Exact binomial test

data:  sum(d > 0) and length(d)
number of successes = 47, number of trials = 100, p-value = 0.6173
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
```

```
 0.3694052 0.5724185
sample estimates:
probability of success
                  0.47
```

Έστω, τώρα, ο έλεγχος σύγκρισης δυο κερμάτων και έστω ότι σε 200 ρίψεις του πρώτου κέρματος εμφανίστηκε *κεφαλή* 80 φορές, ενώ σε 150 ρίψεις του δεύτερου κέρματος εμφανίστηκε 100 φορές *κεφαλή*. Κάνοντας το έλεγχο στην R όπως πιο κάτω, συμπεραίνεται ότι υπάρχει στατιστική διαφορά μεταξύ των δυο κερμάτων.

```
> x<-c(80,100)
> n<-c(200,150)
> prop.test(x,n)

        2-sample test for equality of proportions with continuity correction

data:  x out of n
X-squared = 23.345, df = 1, p-value = 1.354e-06
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.3739929 -0.1593405
sample estimates:
   prop 1    prop 2
0.4000000 0.6666667
```

7.7 Πίνακες Συνάφειας

Για τα πιο κάτω θα χρησιμοποιηθεί το πλαίσιο δεδομένων `solder` το οποίο περιέχεται στη βιβλιοθήκη `faraway`. Οι μεταβλητές προς εξέταση είναι η `Solder` (παράγοντας με 5 επίπεδα) και η `Mask` (με 2 επίπεδα). Για να κατασκευαστεί ο $2x5$ πίνακας συνάφειας (contigency table) χρησιμοποιείται η εντολή `table`. Έπειτα, με την εντολή `summary` δίνεται ο Pearson X^2 έλεγχος ανεξαρτησίας. Επιπρόσθετα, η σταυρωτή ταξινόμηση (cross classification) είναι δυνατή με την εντολή `xtabs`. Το όρισμα στ' αριστερά δίνει την μεταβλητή η οποία θα ταξινομηθεί, ενώ τα ορίσματα στα δεξιά δίνουν τις αντίστοιχες κατηγορίες. Με την εντολή `CrossTable`, η οποία βρίσκεται στη βιβλιοθήκη `gmodels` παρουσιάζεται η ταξινόμηση σε πίνακα μαζί με την συνεισφορά στον X^2 έλεγχο κάθε συνδυασμού. Η εντολή `summary`

χρησιμοποιείται και σ' αυτήν την περίπτωση για τον X^2 έλεγχο ανεξαρτησίας αφού γίνει η ταξινόμηση.

```
> library("gmodels")
> library("faraway")
> attach(solder)
> names(solder)
[1] "Opening" "Solder"  "Mask"    "PadType" "Panel"   "skips"
> X<-table(Solder,Mask)
> X
       Mask
Solder  A1.5  A3  A6  B3  B6
  Thick   90 150  30  90  90
  Thin    90 120  60  90  90
> summary(X)
Number of cases in table: 900
Number of factors: 2
Test for independence of all factors:
        Chisq = 13.333, df = 4, p-value = 0.009757
> cross<-xtabs(.~Solder+Mask)
Error in eval(expr, envir, enclos) : object "." not found
> cross<-xtabs(~Solder+Mask)
> cross
       Mask
Solder  A1.5  A3  A6  B3  B6
  Thick   90 150  30  90  90
  Thin    90 120  60  90  90
```

```
>  CrossTable(cross)


   Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  900
```

	Mask					
Solder	A1.5	A3	A6	B3	B6	Row Total
Thick	90	150	30	90	90	450
	0.000	1.667	5.000	0.000	0.000	
	0.200	0.333	0.067	0.200	0.200	0.500
	0.500	0.556	0.333	0.500	0.500	
	0.100	0.167	0.033	0.100	0.100	
Thin	90	120	60	90	90	450
	0.000	1.667	5.000	0.000	0.000	
	0.200	0.267	0.133	0.200	0.200	0.500
	0.500	0.444	0.667	0.500	0.500	
	0.100	0.133	0.067	0.100	0.100	
Column Total	180	270	90	180	180	900
	0.200	0.300	0.100	0.200	0.200	

7.8 Παράδειγμα

Τα δεδομένα αυτά αποτελούνται από το δείκτη νοημοσύνης (IQ) παιδιών ηλικίας 5 χρονών, τα οποία τα ξεχωρίζουμε βάση του κριτηρίου ότι οι μητέρες τους έχουν υποφέρει από επεισόδιο επιλόχειας κατάθλιψης. Επικεντρωνόμαστε στο να απαντήσουμε το ερώτημα αν τα παιδιά από τις δύο κατηγορίες έχουν διαφορετικό δείκτη νοημοσύνης. Τα δεδομένα βρίσκονται στο παράρτημα αυτού του

κεφαλαίου.

Συνήθως, χρησιμοποιείται ο έλεγχος t για τον έλεγχο με μηδενική υπόθεση ότι οι δύο κατηγορίες έχουν ίσες πληθυσμιακές μέσες τιμές και εναλλακτική ότι δεν έχουν. Ο έλεγχος υποθέτει ότι οι παρατηρήσεις :

1. είναι ανεξάρτητες μεταξύ τους,
2. προέρχονται από πληθυσμό από την κανονική κατανομή,
3. προέρχονται από πληθυσμούς με ίσες διασπορές.

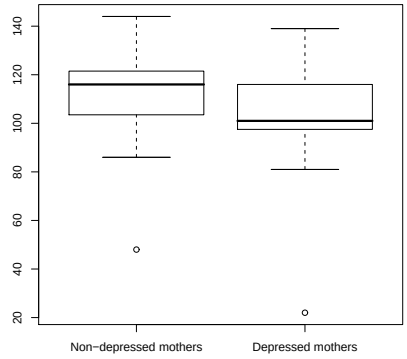
Εκτός από το p-value, το οποίο λαμβάνεται από τον έλεγχο, συνήθως είναι χρήσιμο ένα διάστημα εμπιστοσύνης για τη διαφορά των μέσων τιμών.

Πριν την εφαρμογή του ελέγχου υποθέσεων, πρέπει να εξεταστεί αν τα δεδομένα ικανοποιούν τις υποθέσεις στις οποίες βασίζεται ο έλεγχος. Η αρχική εξέταση των δεδομένων γίνεται εποπτικά με τη βοήθεια γραφημάτων, όπως τα ιστογράμματα, τα κυτιογραφήματα και τα QQ-γραφήματα. Με τα γραφήματα αυτά μπορεί να αναγνωριστεί η απόκλιση από την κανονική κατανομή, η παρουσία απομακρυσμένων τιμών κ.τ.λ. Τα γραφήματα αυτά και για τις δύο κατηγορίες παρουσιάζονται στα Σχήματα 7.7-7.9.

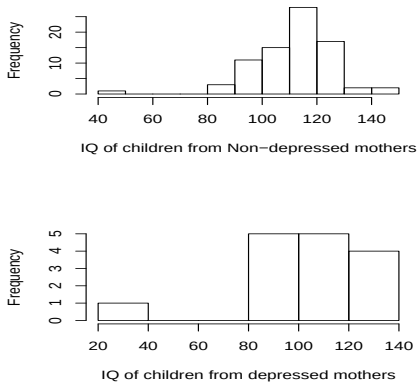
Για να κατασκευαστούν τα γραφήματα και η ανάλυση πρέπει πρώτα να διαχωριστούν οι δύο κατηγορίες των δεδομένων, `scorend` για τα παιδιά με μη καταθλιπτικές μητέρες και `scored` για τα παιδιά με καταθλιπτικές μητέρες.

```
> iqdata<-read.table("iqdata.txt")
> attach(iqdata)
> scorend <- iqdata$V2[V1=="nd"]
> scored <- iqdata$V2[V1=="d"]
> boxplot(scorend, scored, names=c("Non-depressed mothers",
+ "Depressed mothers"))
> par(mfrow=c(2,1))
> hist(scorend, xlab="IQ of children from Non-depressed mothers")
> hist(scored, xlab="IQ of children from depressed mothers")
> qqnorm(scorend,main="")
> qqline(scorend)
> title(main="Normal Probability plot for IQ of children
+ from Non-depressed mothers")
> qqnorm(scored,main="")
> qqline(scored)
```

```
> title(main="Normal Probability plot for IQ of children
+ from depressed mothers")
```

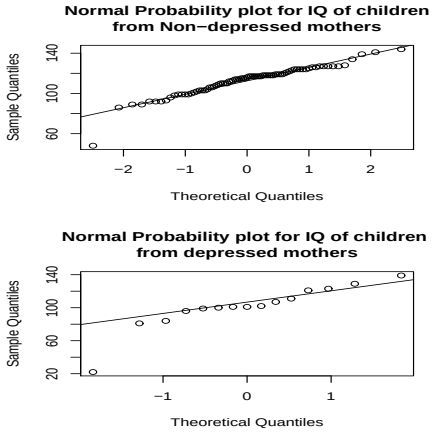


Σχήμα 7.7: Κυτιογραφήματα των δεικτών νοημοσύνης των παιδιών.



Σχήμα 7.8: Ιστογράμματα των δεικτών νοημοσύνης των παιδιών.

Το πιο σημαντικό στοιχείο από όλα τα γραφήματα είναι ότι και στις δύο κατηγορίες παρουσιάζεται από μια απομακρυσμένη παρατήρηση για το δείκτη νοημοσύνης, η οποία και στις δυο περιπτώσεις ανήκει σε παιδί με πολύ χαμηλό δείκτη νοημοσύνης. Τέτοιες παρατηρήσεις μπορούν να επηρεάσουν σημαντικά περιγραφικά μέτρα όπως τη μέση τιμή και τη διακύμανση, αλλά μπορούν να επηρεάσουν



Σχήμα 7.9: QQ-γράφημα των δεικτών νοημοσύνης των παιδιών.

επίσης τους στατιστικούς ελέγχους υποθέσεων οι οποίοι βασίζονται στην κανονικότητα. Προς το παρών, έστω ότι δε λαμβάνεται κανένα διορθωτικό μέτρο για αυτές τις παρατηρήσεις, και εφαρμόζεται ο έλεγχος t για τον οποίο θεωρείται ότι τα δεδομένα των δυο κατηγοριών έχουν την ίδια διακύμανση. Τα αποτελέσματα δίνονται πιο κάτω:

```
> t.test(scorend,scored,var.equal=T)

        Two Sample t-test

data:  scorend and scored
t = 2.4637, df = 92, p-value = 0.01561
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  2.27152 21.16477
sample estimates:
mean of x mean of y
 112.7848  101.0667
```

Η διαφορά στη μέση τιμή του δείκτη νοημοσύνης των δύο κατηγοριών είναι στατιστικώς σημαντική, γεγονός που φαίνεται από τη μικρή τιμή για το p-value, αλλά και από το ότι το μηδέν δεν ανήκει στο διάστημα εμπιστοσύνης. Ωστόσο, η παρουσία των απομακρυσμένων τιμών μπορεί να επηρεάζει αυτό το αποτέλεσμα. Η

διακύμανση σε κάθε κατηγορία, για παράδειγμα, είναι 205.50 (παιδιά από μη καταθλιπτικές μητέρες) και 729.21 (παιδιά από καταθλιπτικές μητέρες). Συνεπώς, υπάρχει μια σημαντική διαφορά στις διακυμάνσεις των δύο κατηγοριών, με αποτέλεσμα να παραβιάζεται μία από τις υποθέσεις του ελέγχου πιο πάνω. Αυτό μπορεί να αποδειχθεί και στατιστικά με τον έλεγχο F για την ισότητα των διακυμάνσεων.

```
> var.test(scorend,scored)

        F test to compare two variances

data:  scorend and scored
F = 0.2818, num df = 78, denom df = 14, p-value = 0.0003276
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.1089903 0.5746361
sample estimates:
ratio of variances
          0.281818
```

Η υπόθεση ισότητας των διακυμάνσεων απορρίπτεται ξεκάθαρα, αφού η σχετική p–value είναι πολύ μικρή. Τι συμβαίνει όμως όταν ο έλεγχος επαναληφθεί αφού αφαιρεθούν οι απομακρυσμένες τιμές από τα δεδομένα; Εφαρμόζεται ξανά ο έλεγχος ισότητας των διακυμάνσεων των δύο κατηγοριών αφού αφαιρεθούν οι τιμές των δύο παιδιών με χαμηλό δείκτη νοημοσύνης, δηλαδή δείκτη νοημοσύνης μικρότερο ή ίσο του 50.

```
> var.test(scorend[scorend>50],scored[scored>50])

        F test to compare two variances

data:  scorend[scorend > 50] and scored[scored > 50]
F = 0.5664, num df = 77, denom df = 13, p-value = 0.1283
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.2102836 1.1765891
sample estimates:
ratio of variances
```

```
          0.5664062
```

Ο έλεγχος τώρα δίνει ότι ο λόγος των διακυμάνσεων είναι ίσος με 0.57, και επισημαίνει ότι η διαφορά ανάμεσα στις διακυμάνσεις δεν είναι στατιστικά σημαντική αφού το p–value είναι μεγαλύτερο από 0.05 και το 1 ανήκει στο σχετικό 95% διάστημα εμπιστοσύνης. Άρα, τα δεδομένα με την αφαίρεση των απομακρυσμένων τιμών έγιναν καταλληλότερα για την εφαρμογή του έλεγχου t. Επίσης, βάση της μελέτης των δεδομένων αυτών είναι πιο λογικό να αφαιρεθούν αφού η μία παρατήρηση ανήκει σε αυτιστικό παιδί ενώ η άλλη σε παιδί το οποίο έπαθε ζημιά στον εγκέφαλο κατά τον τοκετό. Επαναλαμβάνεται, λοιπόν, ο έλεγχος t αφού έχουν αφαιρεθεί οι απομακρυσμένες τιμές :

```
> t.test(scorend[scorend>50],scored[scored>50],var.equal=T)

        Two Sample t-test

data:  scorend[scorend > 50] and scored[scored > 50]
t = 1.8242, df = 90, p-value = 0.07145
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.6148208 14.4170186
sample estimates:
mean of x mean of y
 113.6154  106.7143
```

Είναι φανερό ότι αφού αφαιρεθούν οι δύο απομακρυσμένες τιμές, δεν υπάρχει πλέον η ένδειξη της διαφοράς των μέσων τιμών των δεικτών νοημοσύνης των παιδιών από μη καταθλιπτικές και καταθλιπτικές μητέρες.

Παράρτημα

Τα δεδομένα που χρησιμοποιούνται σε αυτό το κεφάλαιο για απλή στατιστική συμπερασματολογία.

```
> iqdata
   V1  V2
1  nd 103
2  nd 124
```

```
3  nd 124
4  nd 104
5   d  96
6  nd  92
7  nd 124
8  nd  99
9  nd  92
10 nd 116
11 nd  99
12  d  22
13  d  81
14 nd 117
15  d 100
16 nd  89
17 nd 125
18 nd 127
19 nd 112
20 nd  48
21 nd 139
22 nd 118
23  d 107
24 nd 106
25  d 129
26 nd 117
27 nd 123
28 nd 118
29  d  84
30 nd 117
31  d 101
32 nd 141
33 nd 124
34 nd 110
35 nd  98
36 nd 109
37 nd 120
38 nd 127
39 nd 103
```

```
40 nd 118
41 nd 117
42 nd 115
43 nd 119
44 nd 117
45 nd  92
46 nd 101
47 nd 119
48 nd 144
49 nd 119
50 nd 127
51 nd 113
52 nd 127
53 nd 103
54 nd 128
55 nd  86
56 nd 112
57 nd 115
58 nd 117
59 nd  99
60 nd 110
61  d 139
62 nd 117
63 nd  96
64  d 111
65 nd 118
66 nd 126
67 nd 126
68 nd  89
69 nd 102
70 nd 134
71 nd  93
72 nd 115
73  d  99
74 nd  99
75 nd 122
76 nd 106
```

```
77 nd 124
78 nd 100
79 nd 114
80 nd 121
81 nd 119
82 nd 108
83 nd 110
84 nd 127
85 nd 118
86 nd 107
87  d 123
88  d 102
89 nd 110
90 nd 114
91 nd 118
92  d 101
93  d 121
94 nd 114
```

Κεφάλαιο 8

Γραμμική Παλινδρόμηση

Η γραμμική παλινδρόμηση είναι ένα από τα πιο σημαντικά θέματα της Στατιστική θεωρείας. Στη συνέχεια αυτή η πολύ γνωστή μεθοδολογία θα αναπτυχθεί στην R μέσω των τύπων για τα μοντέλα.

8.1 Γραμμικά Μοντέλα στην R

Ο τύπος είναι μια έκφραση της R η οποία καθορίζει τη μορφή του μοντέλου με τις ανάλογες μεταβλητές. Για παράδειγμα, για να καθοριστεί ότι η Y είναι γραμμικός συνδυασμός δύο επεξηγηματικών μεταβλητών $X1$ και $X2$, χρησιμοποιείται ο ακόλουθος τύπος :

```
Y ~ X1 + X2.
```

Η περισπωμένη διαχωρίζει την εξαρτημένη μεταβλητή από τις επεξηγηματικές μεταβλητές. Με άλλα λόγια, εφαρμόζεται το μοντέλο

$$Y = \beta_0 + \beta_1 X1 + \beta_2 X2.$$

Ο τύπος *πάντα* υπονοεί την ύπαρξη του σταθερού όρου στο μοντέλο (β_0 στον πιο πάνω τύπο). Ωστόσο, είναι δυνατόν να αφαιρεθεί ο σταθερός όρος από το μοντέλο, προσθέτοντας στον τύπο του μοντέλου τον όρο -1 σαν επεξηγηματική μεταβλητή:

```
Y ~ -1 + X1 + X2.
```

Όταν ορίζονται κατηγορικές μεταβλητές,δηλαδή παράγοντες, σαν επεξηγηματικές μεταβλητές στα μοντέλα, η συνάρτηση μοντελοποίησης εφαρμόζει έναν σταθερό όρο για κάθε επίπεδο της μεταβλητής. Για παράδειγμα, για να κατασκευαστεί το γραμμικό μοντέλο με εξαρτημένη μεταβλητή το μισθό (`salary`) και επεξηγηματικές μεταβλητές την ηλικία (`age`), η οποία είναι συνεχής, και το φύλο (`gender`), η οποία είναι παράγοντας, ορίζεται όπως πιο κάτω:

```
salary ~ age + gender
```

Ωστόσο, διαφορετική παράμετρος εφαρμόζεται για κάθε ένα από τα δύο επίπεδα για το φύλο. Αυτό είναι ισοδύναμο με το να κατασκευαστεί το μοντέλο με δυο ψευδο-μεταβλητές, μία για *άρρεν* και μία για *θήλυ*. Συνεπώς δε χρειάζεται να οριστούν οι ψευδο-μεταβλητές στο μοντέλο.

Οι κύριες εκφράσεις για ορισμό γραμμικού μοντέλου είναι οι ακόλουθες

- `Y~X`: Γραμμικό μοντέλο του Y συναρτήσει του X
- `X1+X2`: Συμπεριλαμβάνει το $X1$ και το $X2$ στο μοντέλο
- `X1-X2`: Συμπεριλαμβάνει όλα από το $X1$ εκτός από αυτά που βρίσκονται στο $X2$ στο μοντέλο
- `X1:X2`: Συμπεριλαμβάνει την αλληλεπίδραση μεταξύ $X1$ και $X2$, $X1 : X2$, στο μοντέλο
- `X1*X2`: Όλο το μοντέλο $X1 + X2 + X1 : X2$

Οι επόμενες ενότητες εφαρμόζουν αυτές τις έννοιες σε μοντέλο πολλαπλής γραμμικής παλινδρόμισης.

8.2 Πολλαπλή Γραμμική Παλινδρόμηση

Το πλαίσιο δεδομένων `Trees` είναι ένα δείγμα δέντρων μαύρης κερασιάς. Στον ακόλουθο πίνακα παρουσιάζονται μερίκες από τις μετρήσεις για τη διάμετρο (σε ίντσες), το ύψος (σε πόδια) και τον όγκο (σε κυβικά πόδια). Η πλήρης συλλογή δεδομένων βρίσκεται στο παράρτημα αυτού του κεφαλαίου.

Ο σκοπός της συλλογής αυτών των δεδομένων ήταν για να βρεθεί ένας τρόπος πρόβλεψης του όγκου της ξυλείας των δέντρων από τις μετρήσεις για το ύψος και τη διάμετρό τους, χρησιμοποιώντας γραμμικό μοντέλο. Σε αυτήν την περίπτωση

Diameter	Height	Volume
8.3	70	10.3
8.6	65	10.3
8.8	63	10.2
10.5	72	16.4
10.7	81	18.8
10.8	83	19.7

Πίνακας 8.1: Οι πρώτες έξι παρατηρήσεις από το σχετικό πλαίσιο δεδομένων.

η εξαρτημένη μεταβλητή είναι συνεχής και το αρχικό μοντέλο που θα εξεταστεί είναι το συνηθισμένο γραμμικό μοντέλο, με γενική μορφή

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \epsilon$$

όπου Y είναι η εξαρτημένη μεταβλητή, $X_1, \ldots, X_p$ το σύνολο των επεξηγηματικών μεταβλητών, και ϵ το υπόλοιπο. Οι συντελεστές παλινδρόμησεις β_i εκτιμούνται με τη μέθοδο ελαχίστων τετραγώνων υποθέτωντας ότι το ϵ ακολουθεί την κανονική κατανομή με μέση τιμή 0 και σταθερή διακύμανση σ^2. Για n παρατηρήσεις για την εξαρτημένη και τις επεξηγηματικές μεταβλητές, το μοντέλο μπορεί να γραφεί συνοπτικά ως

$$E(\mathbf{Y}) = \mathbf{X}\beta.$$

Η ανάλυση των γραμμικών μοντέλων στην R γίνεται με την εντολή `lm()` όπως παρουσιάζεται πιο κάτω:

```
> trees<-read.table("trees.txt")
> names(trees)<-c("Diameter","Height","Volume")
> trees.fit <- lm(Volume~ Diameter+Height, trees)
> trees.fit

Call:
lm(formula = Volume ~ Diameter + Height, data = trees)

Coefficients:
(Intercept)     Diameter       Height
   -57.9877       4.7082       0.3393
> summary(trees.fit)
```

```
Call:
lm(formula = Volume ~ Diameter + Height, data = trees)

Residuals:
    Min      1Q  Median      3Q     Max
-6.4065 -2.6493 -0.2876  2.2003  8.4847

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -57.9877     8.6382  -6.713 2.75e-07 ***
Diameter      4.7082     0.2643  17.816  < 2e-16 ***
Height        0.3393     0.1302   2.607   0.0145 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.882 on 28 degrees of freedom
Multiple R-Squared: 0.948,      Adjusted R-squared: 0.9442
F-statistic:   255 on 2 and 28 DF,  p-value: < 2.2e-16
> anova(trees.fit)
Analysis of Variance Table

Response: Volume
          Df Sum Sq Mean Sq  F value  Pr(>F)
Diameter   1 7581.8  7581.8 503.1503 < 2e-16 ***
Height     1  102.4   102.4   6.7943 0.01449 *
Residuals 28  421.9    15.1
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> trees.res <- residuals(trees.fit)
> trees.prd <- predict(trees.fit)
```

Η εντολή summary χρησιμοποιείται για τον t έλεγχο για τους συντελεστές της παλινδρόμησης, με μηδενική υπόθεση $\beta_i = 0$. Τα αποτελέσματα μας οδηγούν στο συμπέρασμα ότι οι συντελεστές παλινδρόμησης για τη διάμετρο και το ύψος είναι σημαντικά διαφορετικοί από 0. Επίσης, δίνεται εκτίμηση για τη διακύμανση των υπολοίπων, όπως και ο συντελεστής μεταβλητότητας R^2 από τον οποίο συμπε-

ραίνεται ότι περίπου το 95% της διακύμανσης του όγκου των δέντρων εξηγείται από τις δυο επεξηγηματικές μεταβλητές. Τέλος, δίνεται η τιμή της στατιστικής συνάρτησης F για τον στατιστικό έλεγχο ο οποίος ελέγχει αν όλοι οι συντελεστές παλινδρόμησης είναι ταυτόχρονα ίσοι με 0, το οποίο και απορρίπτεται (γιατί;). Με την εντολή `anova` παρουσιάζεται ο πίνακας ανάλυσης διακύμανσης (ΑΝΑΔΙΑ), ενώ οι εντολές `residuals` και `predict` δίνουν τα υπόλοιπα και τις εκτιμήσεις του μοντέλου, αντίστοιχα.

Το επόμενο στάδιο στην ανάλυση είναι η ανάλυση των υπολοίπων του μοντέλου, δηλαδή της διαφοράς μεταξύ των αρχικών παρατηρήσεων και των εκτιμώμενων από το μοντέλο τιμών. Αυτή γίνεται κυρίως γραφικά, και οι πιο χρήσιμες γραφικές παραστάσεις είναι οι πιο κάτω:

1. Γραφική παράσταση των υπολοίπων συναρτήσει των επεξηγηματικών μεταβλητών του μοντέλου. Η παρουσία καμπυλόγραμμης σχέσης, για παράδειγμα, εισηγείται την πρόσθεση ενός όρου μεγαλύτερου βαθμού, ίσως τετραγωνικού, στο μοντέλο (Σχήμα 8.1).

2. Γραφική παράσταση των υπολοίπων συναρτήσει των εκτιμώμενων τιμών της εξαρτημένης μεταβλητής. Αν η διακύμανση της εξαρτημένης μεταβλητής φαίνεται να μεγαλώνει μαζί με την εκτιμώμενη τιμή, είναι δυνατό να χρειαστεί να γίνει μετασχηματισμός της εξαρτημένης μεταβλητής (Σχήμα 8.2).

3. QQ-γράφημα των υπολοίπων. Μετά την αφαίρεση όλης της συστηματικής διασποράς από τα δεδομένα, τα υπόλοιπα πρέπει να μοιάζουν με ένα δείγμα από την κανονική κατανομή. Είναι το γράφημα των ποσοστημορίων των υπολοίπων συναρτήσει των αναμενόμενων ποσοστημορίων από την κανονική κατανομή (Σχήμα 8.3).

Θα εργαστούμε με τα τυποποιημένα υπόλοιπα τα οποία ορίζονται από

$$r_i = \frac{y_i - \hat{y}_i}{s\sqrt{1 - h_{ii}}}$$

όπου h_{ii} τα διαγώνια στοιχεία του πίνακα

$$\mathbf{H} = \mathbf{X}(\mathbf{X}^{'}\mathbf{X})^{-1}\mathbf{X}^{'}.$$

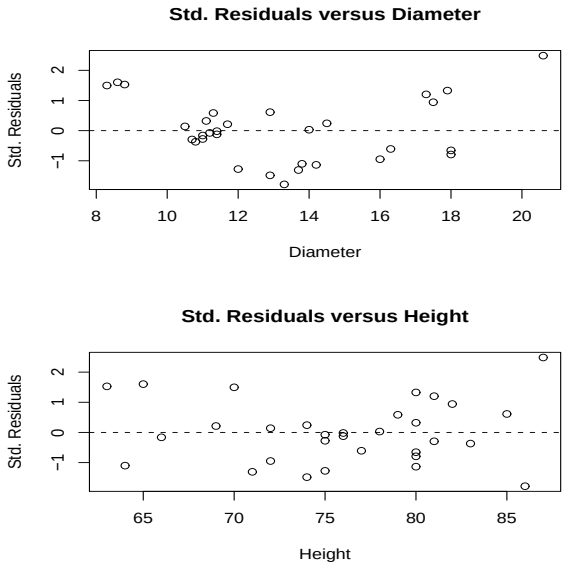
Στην R έχουμε ότι:

```
> s <- summary(trees.fit)$sigma
> h <- lm.influence(trees.fit)$hat
> trees.res <- trees.res/(s*sqrt(1-h))
```

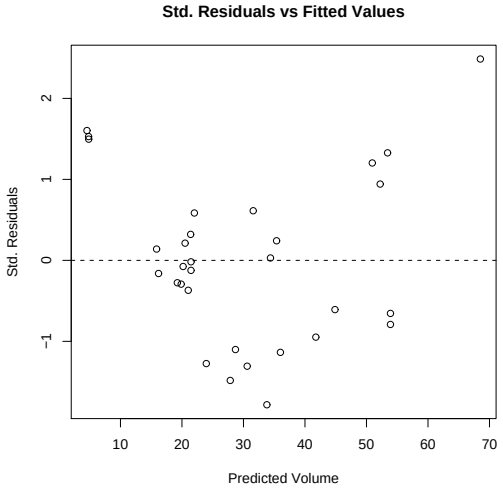
Η πρώτη εντολή εξάγει την εκτιμήτρια για το σ, η δεύτερη δίνει τα διαγώνια στοιχεία του πίνακα $\mathbf{H}$ και η τελευταία υπολογίζει τα τυποποιημένα υπόλοιπα. Ακολουθούν τα γραφήματα των υπολοίπων (Σχήματα 8.1 - 8.3) τα οποία κατασκευάζονται με τις πιο κάτω εντολές :

```
> par(mfrow=c(2,1))
> plot(trees[,"Diameter"], trees.res, xlab="Diameter",
+ ylab="Std. Residuals")
> abline(h=0, lty=2)
> title("Std. Residuals versus Diameter")
> plot(trees[,"Height"], trees.res, xlab="Height",
+ ylab="Std. Residuals")
> abline(h=0, lty=2)
> title("Std. Residuals versus Height")
> par(mfrow=c(1,1))
> plot(trees.prd, trees.res, xlab="Predicted Volume",
+ ylab="Std. Residuals")
> abline(h=0,lty=2)
> title("Std. Residuals vs Fitted Values")
> qqnorm(trees.res,ylab="Std. Residuals",
+ main="Normal Plot of Std. Residuals")
> qqline(trees.res)
```

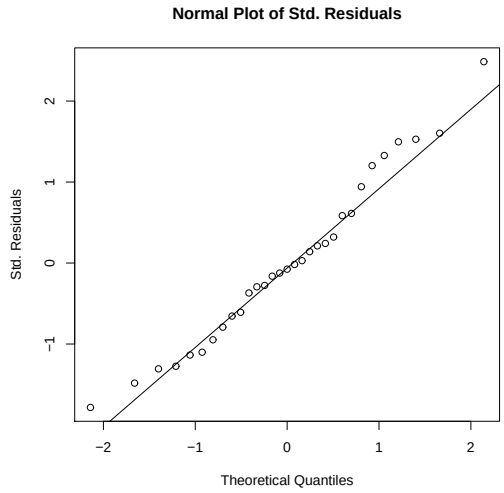
Τα γραφήματα των υπολοίπων συναρτήσει τη διαμέτρου των δέντρων, αλλά και τις εκτιμώμενες τιμές, δείχνουν ότι ένας τετραγωνικός όρος θα μπορούσε να προστεθεί στο μοντέλο. Η ερμηνεία του QQ-γραφήματος συχνά δεν μπορεί να είναι ξεκάθαρη, ειδικά στις περιπτώσεις με μικρά δείγματα. Ωστόσο, εξετάζοντας το φαίνεται ότι τα υπόλοιπα έχουν μικρή απόκλιση από την κανονική.



Σχήμα 8.1: Τυποποιημένα υπόλοιπα συναρτήσει των επεξηγηματικών μεταβλητών.



Σχήμα 8.2: Τυποποιημένα υπόλοιπα συναρτήσει των εκτιμώμενων τιμών της εξαρτημένης μεταβλητης.

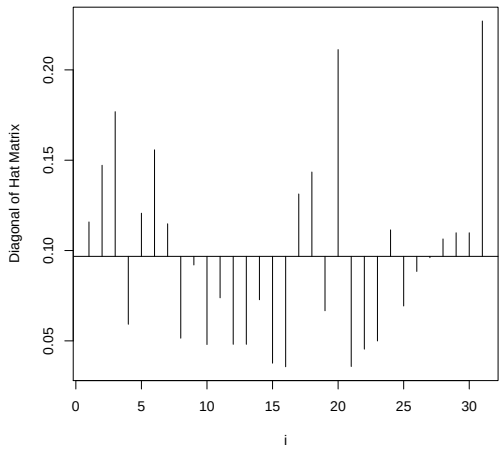


Σχήμα 8.3: QQ-γράφημα των τυποποιημένων υπολοίπων.

Ο πίνακας $\mathbf{H}$ είναι επίσης πολύ βοηθητικός στην αναγνώριση παράξενων ή ασυνήθιστων σημείων των δεδομένων, τα οποία συχνά έχουν μεγάλη επίδραση στην γραμμική παλινδρόμηση. Τέτοια σημεία αναγνωρίζονται από τις σχετικά μεγάλες τιμές στην αντίστοιχη θέση στη διαγώνιο του $\mathbf{H}$. Η μεγαλύτερη τιμή σε οποιοδήποτε στοιχείο της διαγωνίου είναι το 1. Τεχνικά αυτά τα σημεία φαίνονται να έχουν μεγάλη επιρροή (leverage) (Σχήμα 8.4).

```
> h <- lm.influence(trees.fit)$hat
> h
         1          2          3          4          5          6
0.11582883 0.14720958 0.17686186 0.05919131 0.12066468 0.15575111
         7          8          9         10         11         12
0.11480262 0.05148096 0.09200658 0.04797237 0.07382512 0.04809206
        13         14         15         16         17         18
0.04809206 0.07275901 0.03764563 0.03566543 0.13130916 0.14346152
        19         20         21         22         23         24
0.06665975 0.21123665 0.03580935 0.04541796 0.04994875 0.11142518
        25         26         27         28         29         30
0.06930648 0.08841762 0.09603041 0.10641665 0.10982638 0.10982638
        31
 0.22705852
```

```
> plot(1:31,h, type="n", xlab="i", ylab="Diagonal of Hat Matrix")
> abline(h=mean(h))
> segments(1:31, h, 1:31, mean(h))
```



Σχήμα 8.4: Γράφημα επιρροής.

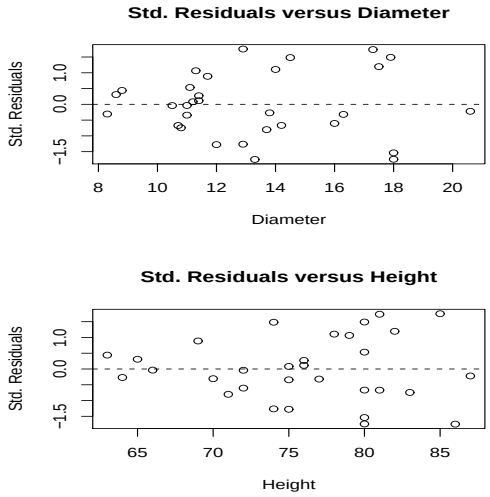
Εδώ δε φαίνεται να υπάρχουν οποιαδήποτε προβληματικά σημεία τα οποία είναι δυνατό να επηρεάσουν υπερβολικά τη διαδικασία εκτίμησης. Οι τιμές της επιρροής είναι σχετικά μικρές.

Επιστρέφοντας τώρα στην ένδειξη από τα γραφήματα των υπολοίπων, θα μελετηθεί ένα νέο μοντέλο το οποίο περιέχει τον τετραγωνικό όρο για τη διάμετρο.

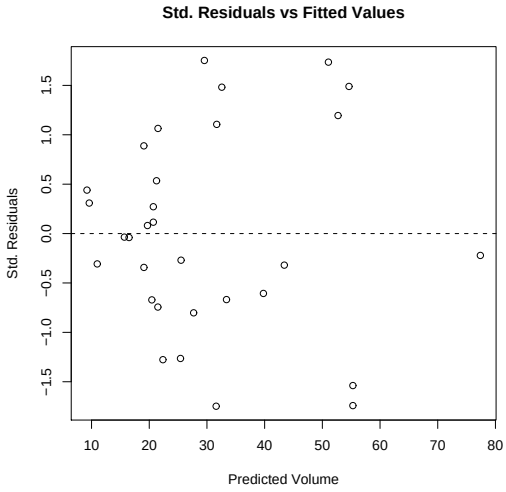
```
> trees1.fit <- lm(Volume~Diameter+I(Diameter*Diameter)+Height,
+ trees)
> trees1.fit

Call:
lm(formula=Volume~Diameter+I(Diameter*Diameter)+ Height,data=trees)

Coefficients:
          (Intercept)              Diameter  I(Diameter * Diameter)
              -9.9204               -2.8851                   0.2686
               Height
```

```
               0.3764

> summary(trees1.fit)

Call:
lm(formula = Volume ~ Diameter + I(Diameter * Diameter) + Height,
    data = trees)

Residuals:
    Min      1Q  Median      3Q     Max
-4.2928 -1.6693 -0.1018  1.7851  4.3489

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)           -9.92041   10.07911  -0.984 0.333729
Diameter              -2.88508    1.30985  -2.203 0.036343 *
I(Diameter * Diameter)  0.26862    0.04590   5.852 3.13e-06 ***
Height                 0.37639    0.08823   4.266 0.000218 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.625 on 27 degrees of freedom
Multiple R-Squared: 0.9771,     Adjusted R-squared: 0.9745
F-statistic: 383.2 on 3 and 27 DF,  p-value: < 2.2e-16
> trees1.res<-residuals(trees1.fit)
> trees1.prd<-predict(trees1.fit)
```

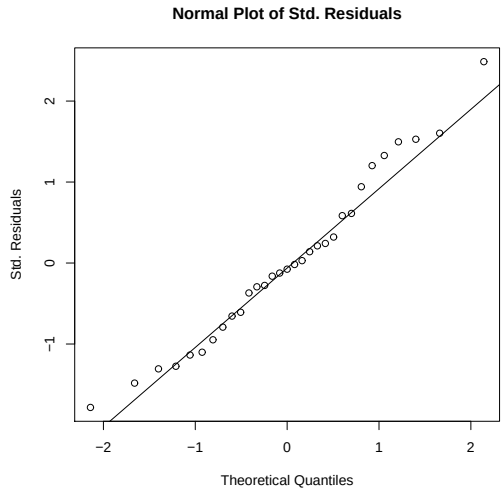
Τα γραφήματα των υπολοίπων κατασκευάζονται όπως προηγουμένως και παρουσιάζονται πιο κάτω (Σχήματα 8.5-8.7). Φαίνεται ότι στο γράφημα των νέων τυποποιημένων υπολοίπων συναρτήσει της διαμέτρου δεν υπάρχει πλέον η συστηματικότητα που υπήρχε πριν. Τα υπόλοιπα γραφήματα κρίνονται ικανοποιητικά.



Σχήμα 8.5: Τυποποιημένα υπόλοιπα δευτεροβάθμιου μοντέλου συναρτήσει των επεξηγηματικών μεταβλητών.



Σχήμα 8.6: Τυποποιημένα υπόλοιπα δευτεροβάθμιου μοντέλου συναρτήσει των εκτιμώμενων τιμών της εξαρτημένης μεταβλητης.



Σχήμα 8.7: QQ-γράφημα των τυποποιημένων υπολοίπων δευτεροβάθμιου μοντέλου.

Παρόλο που τα αποτελέσματα της πολλαπλής γραμμικής παλινδρόμησης υποδεικνύουν ότι οι συντελεστές παλινδρόμησης και για τη διάμετρο και για το ύψος είναι σημαντικά διαφορετικοί από μηδέν, πολύ συχνά είναι χρήσιμο να ερευνηθεί ένας αριθμός μοντέλων σε μια προσπάθεια να βρεθεί το πιο απλό μοντέλο που εφαρμόζει καλύτερα τα δεδομένα. Ουσιαστικά, η διαδικασία επιλογής μοντέλου περιλαμβάνει την πρόσθεση ή την αφαίρεση όρων από ένα προυπάρχων μοντέλο και τον υπολογισμό της επίδρασης της αλλαγής. Υπολογισμός αυτός γίνεται με τη βοήθεια του κριτηρίου πληροφορίας του Akaike (AIC),το οποίο είναι ένα μέτρο της καλής εφαρμογής των δεδομένων από το μοντέλο. Όσο πιο μικρό το AIC τόσο καλύτερο είναι το μοντέλο.

Αρχικά, από το μοντέλο το οποίο περιέχει τη διάμετρο και το ύψος αφαιρείται μια μια μεταβλητή και υπολογίζεται η αλλαγή με το AIC. Όπως φαίνεται, αν αφαιρεθεί οποιαδήποτε από τις μεταβλητές το AIC μεγαλώνει και άρα το αρχικό μοντέλο εφαρμόζει καλύτερα τα δεδομένα. Επίσης διαφαίνεται και η σημαντικότητα της μεταβλητής της διαμέτρου στο μοντέλου αφού αν αφαιρεθεί μεγαλώνει σημαντικά το AIC.

```
> attach(trees)
> trees.drop1 <- drop1(trees.fit)
> trees.drop1
```

```
Single term deletions

Model:
Volume ~ Diameter + Height
         Df Sum of Sq    RSS    AIC
<none>                 421.9   86.9
Diameter  1    4783.0 5204.9  162.8
Height    1     102.4  524.3   91.7
```

Αντίθετα τώρα, ξεκινώντας από το μοντέλο που περιέχει μόνο τον σταθερό όρο, προσθέτουμε μια μια τις μεταβλητές.

```
> trees0.fit <- lm(Volume~1)
> trees.add1 <- add1(trees0.fit,~ Height+Diameter)
> trees.add1
Single term additions

Model:
Volume ~ 1
         Df Sum of Sq    RSS    AIC
<none>                8106.1  174.6
Height    1    2901.2 5204.9  162.8
Diameter  1    7581.8  524.3   91.7
```

Από τα αποτελέσματα συμπεραίνεται και πάλι η σημαντικότητα της ύπαρξης και των δυο μεταβλητών στο μοντέλο αφού το AIC γίνεται μικρότερο με την πρόσθεση τους. Εφαρμόζοντας τώρα τις εντολές `drop1` και `add1` στο δευτεροβάθμιο μοντέλο παρατηρείται ότι όλοι οι παράγοντες είναι σημαντικοί για το μοντέλο.

```
> trees1.drop1 <- drop1(trees1.fit)
> trees1.drop1
Single term deletions

Model:
Volume ~ Diameter + I(Diameter * Diameter) + Height
                       Df Sum of Sq    RSS    AIC
<none>                                186.01  63.55
Diameter                1     33.42 219.44  66.67
I(Diameter * Diameter)  1    235.91 421.92  86.94
```

```
Height                   1    125.37 311.38  77.52
>
> trees10.fit <- lm(Volume~1)
> trees1.add1 <- add1(trees10.fit,~ Height+Diameter+I(Diameter*Diameter))
> trees1.add1
Single term additions

Model:
Volume ~ 1
                       Df Sum of Sq    RSS    AIC
<none>                              8106.1  174.6
Height                  1    2901.2 5204.9  162.8
Diameter                1    7581.8  524.3   91.7
I(Diameter * Diameter)  1    7776.8  329.3   77.3
```

Παράρτημα

Τα δεδομένα που χρησιμοποιούνται σε αυτό το κεφάλαιο για εφαρμογή της πολλαπλής παλινδρόμησης.

```
> trees
   Diameter Height Volume
1       8.3     70   10.3
2       8.6     65   10.3
3       8.8     63   10.2
4      10.5     72   16.4
5      10.7     81   18.8
6      10.8     83   19.7
7      11.0     66   15.6
8      11.0     75   18.2
9      11.1     80   22.6
10     11.2     75   19.9
11     11.3     79   24.2
12     11.4     76   21.0
13     11.4     76   21.4
14     11.7     69   21.3
15     12.0     75   19.1
16     12.9     74   22.2
17     12.9     85   33.8
18     13.3     86   27.4
19     13.7     71   25.7
20     13.8     64   24.9
21     14.0     78   34.5
22     14.2     80   31.7
23     14.5     74   36.3
24     16.0     72   38.3
25     16.3     77   42.6
26     17.3     81   55.4
27     17.5     82   55.7
28     17.9     80   58.3
29     18.0     80   51.5
30     18.0     80   51.0
```

```
31      20.6      87   77.0
```

Κεφάλαιο 9

Ανάλυση της Διακύμανσης

Η ανάλυση της διακύμανσης είναι μια από τις πλέον σημαντικές μεθόδους για ανάλυση δεδομένων. Η μέθοδος αυτή αναφέρετε στη διαμέριση του συνολικού αθροίσματος τετραγώνων σε αθροίσματα τετραγώνων λόγω των επιδράσεων των παραγόντων.

9.1 Ανάλυση Διακύμανσης κατά ένα Παράγοντα

Το πιο απλό είδος πειραμάτων είναι αυτά στα οποία μια απλή συνεχής εξαρτημένη μεταβλητή μετρείται έναν αριθμό φορών για κάθε ένα από τα διάφορα επίπεδα ενός πειραματικού παράγοντα. Για παράδειγμα, έστω τα δεδομένα στον Πίνακα 9.1, ο οποίος περιλαμβάνει τις τιμές του χρόνου πήξης του αίματος για τέσσερις διαφορετικές δίαιτες.

Ο χρόνος πήξης είναι η συνεχής εξαρτημένη μεταβλητή, ενώ η δίαιτα είναι ποιοτική μεταβλητή, ή παράγοντας, με τέσσερα επίπεδα: A, B, C, D. Ο κύριος στόχος είναι να εξεταστεί αν αν ο παράγοντας δίαιτα έχει οποιαδήποτε επίδραση στο μέσο χρόνο πήξης του αίματος. Για να γίνει η ανάλυση δεδομένων, πρέπει να γραφούν στην R με τέτοιο τρόπο έτσι ώστε να μπορούν να χρησιμοποιηθούν για ανάλυση της διακύμανσης. Αυτό επιτυγχάνεται με το σχεδιασμό ενός πλαισίου δεδομένων όπως πιο κάτω:

A	B	C	D
62	63	68	56
60	67	66	62
63	71	71	60
59	64	67	61
	65	68	63
	66	68	64
			63
			59

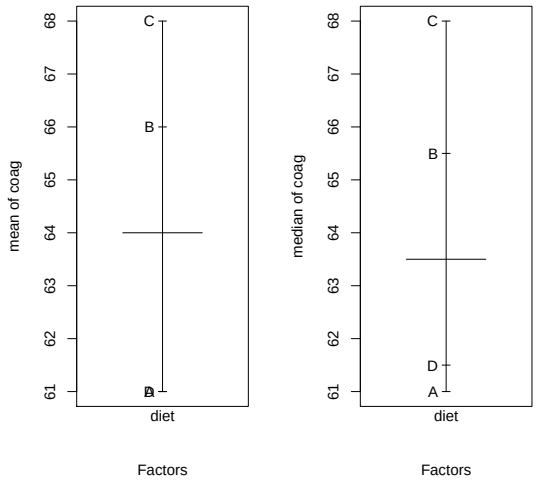
Πίνακας 9.1: Χρόνος πήξης του αίματος για τέσσερις δίαιτες.

```
> coag <- scan()
1: 62 60 63 59
5: 63 67 71 64 65 66
11: 68 66 71 67 68 68
17: 56 62 60 61 63 64 63 59
25:
> coag
 [1] 62 60 63 59 63 67 71 64 65 66 68 66 71 67 68 68 56 62 60 61
[21] 63 64 63 59
> diet <- factor(rep(LETTERS[1:4],c(4,6,6,8)))  #create a factor
> diet
 [1] A A A A B B B B B B C C C C C C D D D D D D D D
> coag.df <- data.frame(diet,coag)          #create a data frame
> coag.df
   diet coag
 1    A   62
 2    A   60
 3    A   63
 4    A   59
 5    B   63
 6    B   67
 7    B   71
 8    B   64
 9    B   65
```

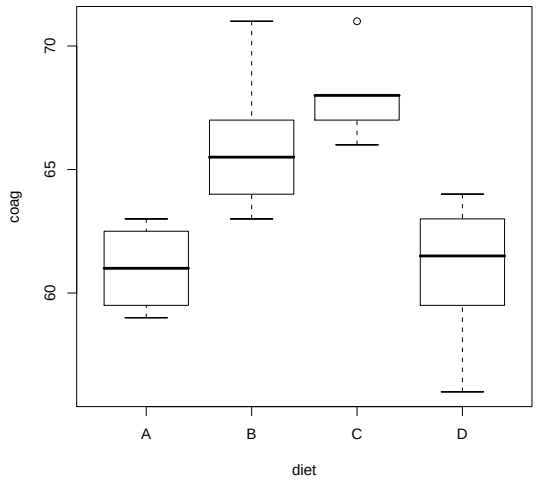
```
10    B   66
11    C   68
12    C   66
13    C   71
14    C   67
15    C   68
16    C   68
17    D   56
18    D   62
19    D   60
20    D   61
21    D   63
22    D   64
23    D   63
24    D   59
```

Το πρώτο βήμα στην ανάλυση δεδομένων είναι να ερευνηθεί γραφικά αν υπάρχουν ή όχι διαφορές ανάμεσα στα επίπεδα του παράγοντα. Τα Σχήματα 9.1 και 9.2 παρουσιάζουν τις μέσες τιμές και τις διαμέσους για κάθε επίπεδο του παράγοντα και το αντίστοιχο κυτιογράφημα. Η οριζόντια ευθεία στο αριστερό (δεξιό) γράφημα δίνει τη μέση τιμή (διάμεσο) όλων των δεδομένων. Είναι φανερό πως τα επίπεδα A και D σχηματίζουν μια κατηγορία, ενώ τα επίπεδα B και C μιαν άλλη κατηγορία.

```
> par(mfrow=c(1,2))
> plot.design(coag.df)
> plot.design(coag.df, fun= median)
> par(mfrow=c(1,1))
> plot(coag.df)
```



Σχήμα 9.1: Μέσες τιμές και διάμεσοι των επιπέδων του παράγοντα.



Σχήμα 9.2: Κυτιογράφημα των επιπέδων του παράγοντα.

Για να εφαρμοστεί η ανάλυση της διακύμανσης στην R χρησιμοποιείται η εντολή aov ως ακολούθως

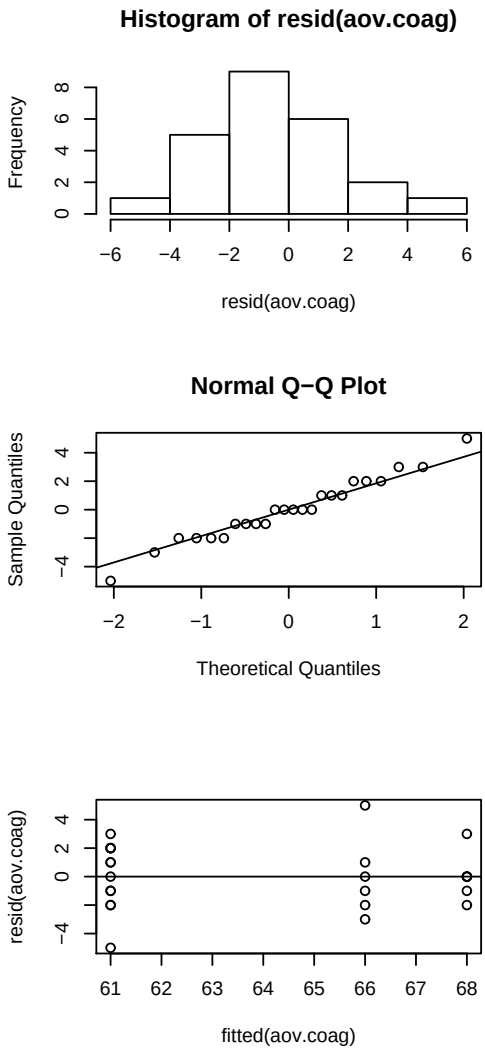
```
> aov.coag <- aov(coag ~ diet, coag.df)
> aov.coag
Call:
   aov(formula = coag ~ diet, data = coag.df)

Terms:
                diet Residuals
Sum of Squares   228       112
Deg. of Freedom    3        20

Residual standard error: 2.366432
Estimated effects may be unbalanced
> summary(aov.coag)
            Df Sum Sq Mean Sq F value    Pr(>F)
diet         3  228.0    76.0  13.571 4.658e-05 ***
Residuals   20  112.0     5.6
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Σημειώνεται ότι η εντολή aov χρησιμοποιείται με ανάλογο τρόπο όπως και η εντολή lm για τη γραμμική παλινδρόμηση. Ο τύπος coag ~ diet στο πρώτο όρισμα δίνει συμβολικά το μοντέλο της ανάλυσης της διακύμανσης κατά ένα παράγοντα, ενώ το δεύτερο όρισμα, coag.df, καθορίζει το πλαίσιο δεδομένων. Με την εντολή summary δίνεται ο πίνακας ΑΝΑΔΙΑ. Το αποτέλεσμα υποδεικνύει τη σημαντικότητα του παράγοντα οδηγώντας στο συμπέρασμα ότι υπάρχουν διαφορές ανάμεσα στις τέσσερις δίαιτες.

Όπως και στη γραμμική παλινδρόμηση, είναι χρήσιμη η γραφική ανάλυση των υπολοίπων για τον έλεγχο των υποθέσεων που απαιτούνται από την ανάλυση της διακύμανσης, δηλαδή να είναι ασυσχέτιστα, να έχουν σταθερή διακύμανση και να είναι κανονικά. Από τα γραφήματα (Σχήμα 9.3) φαίνεται ότι οι υποθέσεις αυτές ικανοποιούνται σε μεγάλο βαθμό.

```
> fitted.values(aov.coag)
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
61 61 61 61 66 66 66 66 66 66 68 68 68 68 68 68 61 61 61 61 61 61
```

```
23 24
61 61
> par(mfrow=c(3.1))
> hist(resid(aov.coag))
> qqnorm(resid(aov.coag))
> qqline(resid(aov.coag))
> plot(fitted(aov.coag), resid(aov.coag))
> abline(h=0)
```



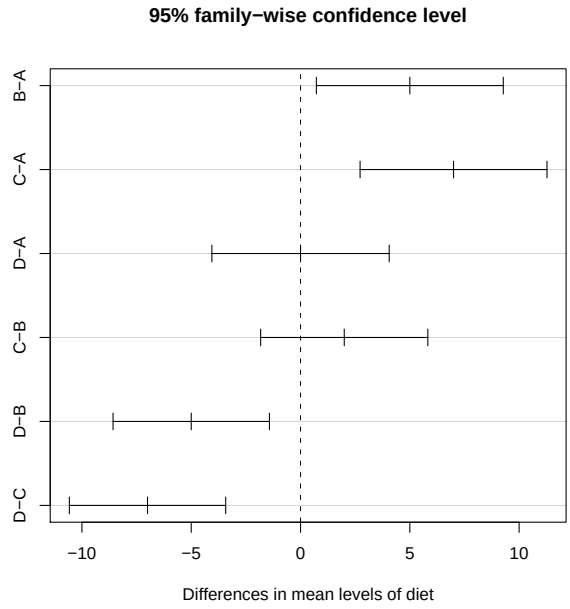
Σχήμα 9.3: Ανάλυση υπολοίπων.

9.2 Πολλαπλές Συγκρίσεις

Από την προηγούμενη ανάλυση επισημάνθηκε η διαφορά μεταξύ των επιπέδων του παράγοντα δίαιτα. Συνεπώς, είναι ενδιαφέρον να αναγνωριστούν αυτές οι διαφορές. Η κύρια μέθοδος πολλαπλών συγκρίσεων που χρησιμοποιείται στην R είναι η μέθοδος Tukey, η οποία εφαρμόζεται με την εντολή TukeyHSD. Η εντολή αυτή υπολογίζει τα 95% διαστήματα εμπιστοσύνης για όλα τα ζεύγη διαφορών ανάμεσα των μέσων τιμών των ειδών δίαιτας. Τα διαστήματα αυτά μπορούν να παρουσιαστούν και γραφικά για εποπτική σύγκριση, θέτοντας σαν όρισμα στην εντολή plot το αντικείμενο που παράγεται από την εντολή TukeyHSD. Όπως αναφέρθηκε και προηγουμένως, παρατηρείται ότι οι δίαιτες A και D σχηματίζουν μια κατηγορία, ενώ τα επίπεδα B και C μιαν άλλη κατηγορία, αφού το μηδέν περιέχεται στο διάστημα εμπιστοσύνης της διαφοράς τους.

```
> mca.coag <- TukeyHSD(aov.coag,"diet")
> mca.coag
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = coag ~ diet, data = coag.df)

$diet
             diff         lwr       upr     p adj
B-A  5.000000e+00   0.7245544  9.275446 0.0183283
C-A  7.000000e+00   2.7245544 11.275446 0.0009577
D-A -1.421085e-14  -4.0560438  4.056044 1.0000000
C-B  2.000000e+00  -1.8240748  5.824075 0.4766005
D-B -5.000000e+00  -8.5770944 -1.422906 0.0044114
D-C -7.000000e+00 -10.5770944 -3.422906 0.0001268
> plot(mca.coag)
```

Γενικά, μπορούν να εφαρμοστούν και οι υπόλοιπες γνωστές μέθοδοι πολλαπλών συγκρίσεων (Dunnett, Sidak, Bonferroni και Scheffe) χρησιμοποιώντας τη βιβλιοθήκη της R multcomp.



Σχήμα 9.4: 95 % ταυτόχρονα διαστήματα εμπιστοσύνης των διαφορών των μέσων των επιπέδων του παράγοντα με τη μέθοδο Tukey.

Κεφάλαιο 10

Λογιστική Παλινδρόμηση

Στο κεφάλαιο αυτό θα δούμε την μέθοδο της λογιστικής παλινδρόμησης η οποία χρησιμεύει στο να αναπτύξουμε σχέση μίας δίτιμης ανεξάρτητης τυχαίας μεταβλητής και συνεχών η διακριτών ανεξάρτητων μεταβλητών. Ουσιαστικά η μέθοδος αυτή γενικεύει τα γραμμικά μοντέλα, έτσι ώστε η εξαρτημένη μεταβλητή να ακολουθεί την εκθετική οικογένεια κατανομών.

10.1 Περιγραφή των Δεδομένων

Έρευνα με εργάτες της αμερικάνικης βιομηχανίας βαμβακιού θέλει να εξετάσει αν κάποιος εργάτης πάσχει από κάποια συγκεκριμένη ασθένεια του πνεύμονα. Επίσης, συγκεντρώθηκαν οι τιμές για τις ακόλουθες πέντε μεταβλητές :

- φυλή (race) (1=λευκός, 2=άλλο)
- φύλο (sex) (1=άρρεν, 2=θήλυ)
- κάπνισμα (1=καπνιστής, 2=μη καπνιστής)
- διάρκεια εργασίας (1= λιγότερο από 10 χρόνια, 2=10–22 χρόνια, 3= περισσότερο από 20 χρόνια)
- σκόνη: ποσοστό σκόνης στον εργασιακό χώρο (1=ψηλό, 2=μέτριο 3=χαμηλό)

Τα δεδομένα βρίσκονται στο παράρτημα αυτού του κεφαλαίου.

Το πρόβλημα για αυτά τα δεδομένα είναι το να εξακριβωθεί κατά πόσο οι επεξηγηματικές μεταβλητές είναι σημαντικές στην εμφάνιση αυτής της ασθένειας.

Με άλλα λόγια, ποιες από αυτές τις μεταβλητές μπορούν να χρησιμοποιηθούν για να προβλέψουν κατά πόσο ένας εργάτης πάσχει από ασθένεια του πνεύμονα. Επειδή η ανεξάρτητη μεταβλητή είναι δυαδική, θα χρησιμοποιηθεί η λογιστική παλινδρόμηση για την ανάλυση.

10.2 Λογιστική Παλινδρόμηση

Αντί να χρησιμοποιηθεί ένα γραμμικό μοντέλο για να εξεταστεί η εξάρτηση της πιθανότητας εμφάνισης της ασθένειας του πνεύμονα από τις επεξηγηματικές μεταβλητές, χρησιμοποιείται ο λογιστικός μετασχηματισμός, ο οποίος ορίζεται ως

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 x_2 + \ldots + \beta_k X_k. \tag{10.1}$$

Στο παράδειγμα, p είναι η πιθανότητα ένας εργάτης να πάσχει από ασθένεια του πνεύμονα. Στο μοντέλο υπάρχουν k (στο παράδειγμα 5) επεξηγηματικές μεταβλητές. Οι συντελεστές παλινδρόμησης εκτιμούνται με τη μέθοδο της μέγιστης πιθανοφάνειας με την υπόθεση ότι η εξαρτημένη μεταβλητή ακολουθεί τη διωνυμική κατανομή. Από την εξίσωση (10.1), το p μπορεί να υπολογιστεί από

$$p = \frac{\exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k)}{1 + \exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k)} \tag{10.2}$$

10.3 Ανάλυση στην R

Οι πρώτες δύο στήλες των δεδομένων καταγράφουν τη συχνότητα των εργατών με ή χωρίς την ασθένεια για τις αντίστοιχες τιμές (κατηγορίες) των επεξηγηματικών μεταβλητών. Η ανάλυση της λογιστικής παλινδρόμησης γίνεται με την εντολή `glm` με ανάλογο τρόπο με τη εντολή `lm` για τη γραμμική παλινδρόμησης δίνοντας και την συνάρτηση σύνδεσης (link function) με το όρισμα `family`.

```
> logreg<-read.table("logistic1.txt",header=T)
> attach(logreg)
> out1<-glm( cbind(Yes, No)~dust+race+sex+smoking+Empleng,
+ family=binomial)
> out1

Call:  glm(formula = cbind(Yes, No) ~ dust + race + sex + smoking
       +Empleng, family = binomial)
```

```
Coefficients:
(Intercept)       dust       race        sex    smoking    Empleng
    -0.4852    -1.3751     0.2463    -0.2590    -0.6292     0.3856

Degrees of Freedom: 64 Total (i.e. Null);  59 Residual
Null Deviance:      322.5
Residual Deviance: 69.51        AIC: 188.2
```

Το αποτέλεσμα δίνει τις εκτιμήσεις των συντελεστών των παραμέτρων, την απόκλιση (deviance) του μηδενικού μοντέλου και των υπολοίπων μαζί με τους βαθμούς ελευθερίας τους αλλά και την τιμή του κριτηρίου AIC. Πιο λεπτομερή ανάλυση των συντελεστών των παραμέτρων δίνεται με την εντολή summary, ενώ η εντολή anova παρουσιάζει τον πίνακα ανάλυσης της απόκλισης.

```
> summary(out1)

Call:
glm(formula = cbind(Yes, No) ~ dust + race + sex + smoking +
    Empleng, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.4126  -0.7573  -0.2421   0.3688   1.9804

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.4852     0.6060  -0.801 0.423312
dust         -1.3751     0.1155 -11.901  < 2e-16 ***
race          0.2463     0.2061   1.195 0.232026
sex          -0.2590     0.2116  -1.224 0.220949
smoking      -0.6292     0.1931  -3.259 0.001119 **
Empleng       0.3856     0.1069   3.607 0.000310 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 322.527  on 64  degrees of freedom
Residual deviance:  69.509  on 59  degrees of freedom
AIC: 188.19

Number of Fisher Scoring iterations: 5
> anova(out1)
Analysis of Deviance Table

Model: binomial, link: logit

Response: cbind(Yes, No)

Terms added sequentially (first to last)


        Df Deviance Resid. Df Resid. Dev
NULL                    64     322.53
dust     1   221.96        63     100.56
race     1     1.05        62      99.51
sex      1     5.97        61      93.54
smoking  1    10.73        60      82.82
Empleng  1    13.31        59      69.51
```

Από τα πιο πάνω συμπεραίνεται ότι οι μεταβλητές dust, smoking και Empleng είναι οι πιο σημαντικές για την πρόβλεψη ασθένειας του πνεύμονα, ενώ φαίνεται ότι οι άλλες δύο μεταβλητές δεν είναι τόσο σημαντικές. Στο συμπέρασμα αυτό καταλήγουμε από το p-value τους για τον t έλεγχο, αλλά και από την συνεισφορά της κάθε μεταβλητής στην απόκλιση όταν αυτή προστεθεί στο μοντέλο, η οποία παρουσιάζεται στο πίνακα ανάλυσης της απόκλισης. Συνεπώς, εφαρμόζεται ένα νέο μοντέλο λογιστικής παλινδρόμησης με τις τρεις σημαντικές μεταβλητές και το συγκρίνεται με το προηγούμενο.

```
> out2<-glm( cbind(Yes, No)~dust+smoking+Empleng, family=binomial)
> anova(out2,out1)
Analysis of Deviance Table

Model 1: cbind(Yes, No) ~ dust + smoking + Empleng
Model 2: cbind(Yes, No) ~ dust + race + sex + smoking + Empleng
```

```
  Resid. Df Resid. Dev Df Deviance
1        61     72.562
2        59     69.509  2    3.053
> 1-pchisq(3.053,2)
[1] 0.2172949
```

Ο έλεγχος σύγκρισης μοντέλου έχει για μηδενική υπόθεση H_0 ότι το νέο μοντέλο εφαρμόζει καλύτερα τα δεδομένα. Ο έλεγχος είναι X^2 και αφού το p-value (`1-pchisq(3.053,2)`) είναι μεγαλύτερο από 0.05, δεν απορρίπτεται η μηδενική υπόθεση. Ποιο κάτω παρουσιάζεται η ανάλυση για του συντελεστές του μικρότερου μοντέλου.

```
> summary(out2)

Call:
glm(formula = cbind(Yes, No) ~ dust + smoking + Empleng, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.3421  -0.7700  -0.2518   0.4001   2.0523

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.14177    0.34120  -0.415 0.677783
dust        -1.46572    0.10578 -13.856  < 2e-16 ***
smoking     -0.67781    0.18871  -3.592 0.000328 ***
Empleng      0.33313    0.08861   3.760 0.000170 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 322.527  on 64  degrees of freedom
Residual deviance:  72.562  on 61  degrees of freedom
AIC: 187.24

Number of Fisher Scoring iterations: 5
```

Θεωρώντας τις τιμές των συντελεστών από πιο πάνω, το μοντέλο λογιστικής παλινδρόμησης που εφαρμόζει καλύτερα τα δεδομένα δίνεται από

$$\log\left(\frac{\hat{p}}{1-\hat{p}}\right) = -0.1418 - 1.4657 \times \text{dust} - 0.6778 \times \text{smoking} + 0.3331 \times \text{Empleng}$$

και είναι δυνατόν να υπολογιστεί η εκτιμώμενη τιμή της πιθανότητας κάποιος εργάτης να πάσχει από ασθένεια του πνεύμονα για κάθε συνδυασμό τιμών από τις τρεις επεξηγηματικές μεταβλητές. Για παράδειγμα, αν ένας εργάτης δουλεύει σε εργασιακό χώρο με ψηλό ποσοστό σκόνης (`dust=1`), καπνίζει (`smoking=1`) και δουλεύει για περισσότερο από 20 χρόνια (`Empleng=3`), η εξίσωση δίνει το αποτέλεσμα $\log\left(\hat{p}/(1-\hat{p})\right) = -1.286$, και συνεπώς $\hat{p} = 0.2165$.

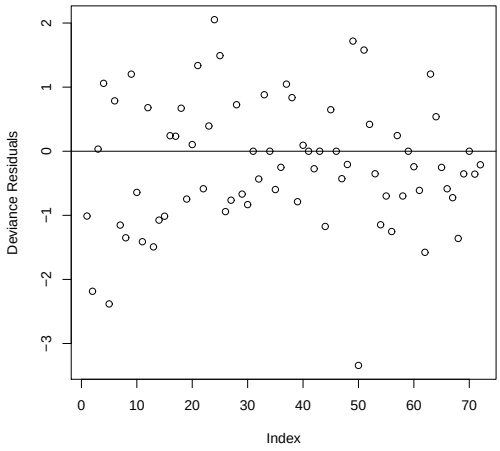
Στη συνέχεια υπολογίζονται δύο είδη υπολοίπων της λογιστικής παλινδρόμησης, τα υπόλοιπα απόκλισης και τα υπόλοιπα Pearson, και κατασκευάζεται το γράφημά τους (Σχήματα 10.1 και 10.2). Η μεθοδολογία της ανάλυσης υπολοίπων είναι παρόμοια με εκείνης της πολλαπλής γραμμικής παλινδρόμησης. Και τα δύο γραφήματα δείχνουν ότι η 50η παρατήρηση είναι λίγο προβληματική. Η αρνητική τιμή του υπολοίπου υποδεικνύει ότι η εκτιμώμενη τιμή είναι μεγαλύτερη από την παρατηρούμενη τιμή. Εξετάζοντας τα δεδομένα, παρατηρείται ότι η 50η παρατήρηση αναφέρεται στους εργάτες με μέτριο ποσοστό σκόνης στον εργασιακό τους χώρο (`dust=2`), καπνίζουν (`smoking=1`) και εργάζονται για περισσότερα από 20 χρόνια (`Empleng=3`) και άρα $\hat{p} = 0.059$. Η εκτιμώμενη πιθανότητα είναι $1/142 = 0.007$.

```
> residuals(out2, type="d")
> residuals(out2, type="pear")
> plot(residuals(out2, type="d"), xlab="Index",
+ ylab="Deviance Residuals")
> abline(h=0)
> plot(residuals(out2, type="pear"), xlab="Index",
+ ylab="Pearson Residuals")
> abline(h=0)
```

10.4 Μοντέλο Probit

Όμοια ανάλυση μπορεί να γίνει χρησιμοποιώντας το μοντέλο probit

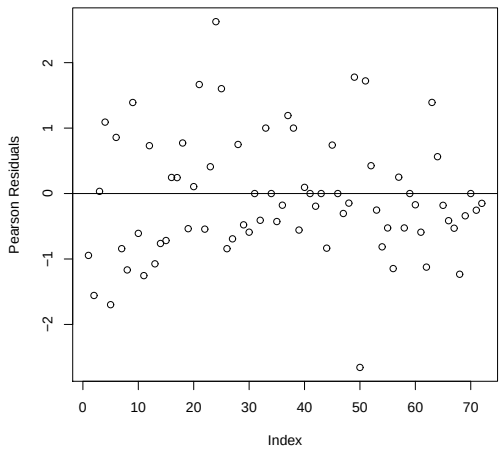
$$p = \Phi(\beta_0 + \beta_1 X_1 + \ldots + \beta_p X_K),$$



Σχήμα 10.1: Υπόλοιπα απόκλισης.

όπου Φ η συνάρτηση πυκνότητας πιθανότητας της τυπικής κανονικής κατανομής.

```
> out3<-glm( cbind(Yes, No)~dust+smoking+Empleng,
+ family=binomial(link=probit))
> out3

Call:  glm(formula = cbind(Yes, No) ~ dust + smoking
+ Empleng, family = binomial(link = probit))

Coefficients:
(Intercept)         dust      smoking      Empleng
    -0.4044      -0.6268      -0.2840       0.1406

Degrees of Freedom: 64 Total (i.e. Null);  61 Residual
Null Deviance:      322.5
Residual Deviance: 84.59        AIC: 199.3
> summary(out3)

Call: glm(formula = cbind(Yes, No) ~ dust + smoking + Empleng,
family = binomial(link = probit))
```



Σχήμα 10.2: Υπόλοιπα Pearson.

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.5085  -0.7912  -0.2626   0.2894   2.5515

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.40438    0.15877  -2.547 0.010867 *
dust        -0.62685    0.04632 -13.532  < 2e-16 ***
smoking     -0.28397    0.08214  -3.457 0.000546 ***
Empleng      0.14065    0.04056   3.468 0.000525 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 322.527  on 64  degrees of freedom
Residual deviance:  84.587  on 61  degrees of freedom
AIC: 199.26

Number of Fisher Scoring iterations: 5
```

Δεδομένα Κεφαλαίου 10

	Yes	No	dust	race	sex	smoking	Empleng
1	3	37	1	1	1	1	1
2	0	74	2	1	1	1	1
3	2	258	3	1	1	1	1
4	25	139	1	2	1	1	1
5	0	88	2	2	1	1	1
6	3	242	3	2	1	1	1
7	0	5	1	1	2	1	1
8	1	93	2	1	2	1	1
9	3	180	3	1	2	1	1
10	2	22	1	2	2	1	1
11	2	145	2	2	2	1	1
12	3	260	3	2	2	1	1
13	0	16	1	1	1	2	1
14	0	35	2	1	1	2	1
15	0	134	3	1	1	2	1
16	6	75	1	2	1	2	1
17	1	47	2	2	1	2	1
18	1	122	3	2	1	2	1
19	0	4	1	1	2	2	1
20	1	54	2	1	2	2	1
21	2	169	3	1	2	2	1
22	1	24	1	2	2	2	1
23	3	142	2	2	2	2	1
24	4	301	3	2	2	2	1
25	8	21	1	1	1	1	2
26	1	50	2	1	1	1	2
27	1	187	3	1	1	1	2
28	8	30	1	2	1	1	2
29	0	5	2	2	1	1	2
30	0	33	3	2	1	1	2
31	0	0	1	1	2	1	2
32	1	33	2	1	2	1	2
33	2	94	3	1	2	1	2
34	0	0	1	2	2	1	2

35	0	4	2	2	2	1	2
36	0	3	3	2	2	1	2
37	2	8	1	1	1	2	2
38	1	16	2	1	1	2	2
39	0	58	3	1	1	2	2
40	1	9	1	2	1	2	2
41	0	0	2	2	1	2	2
42	0	7	3	2	1	2	2
43	0	0	1	1	2	2	2
44	0	30	2	1	2	2	2
45	1	90	3	1	2	2	2
46	0	0	1	2	2	2	2
47	0	4	2	2	2	2	2
48	0	4	3	2	2	2	2
49	31	77	1	1	1	1	3
50	1	141	2	1	1	1	3
51	12	495	3	1	1	1	3
52	10	31	1	2	1	1	3
53	0	1	2	2	1	1	3
54	0	45	3	2	1	1	3
55	0	1	1	1	2	1	3
56	3	91	2	1	2	1	3
57	3	176	3	1	2	1	3
58	0	1	1	2	2	1	3
59	0	0	2	2	2	1	3
60	0	2	3	2	2	1	3
61	5	47	1	1	1	2	3
62	0	39	2	1	1	2	3
63	3	182	3	1	1	2	3
64	3	15	1	2	1	2	3
65	0	1	2	2	1	2	3
66	0	23	3	2	1	2	3
67	0	2	1	1	2	2	3
68	3	187	2	1	2	2	3
69	2	340	3	1	2	2	3
70	0	0	1	2	2	2	3
71	0	2	2	2	2	2	3

72	0	3	3	2	2	2	3

Κεφάλαιο 11

Τεχνικές Αναδειγματοληψίας

Ο στατιστικός πολύ συχνά ενδιαφέρεται να υπολογίσει μια εκτιμήτρια μαζί με το τυπικό της σφάλμα με σκοπό να κατασκευάσει διαστήματα εμπιστοσύνης για την πραγματική τιμή της παραμέτρου. Ωστόσο, αρκετές φορές είναι δύσκολο να βρεθεί μια ακριβής έκφραση για τη διακύμανση διαφόρων εκτιμητριών, και συνεπώς, είναι αδύνατο να υπολογιστεί το τυπικό τους σφάλμα. Βασικές μέθοδοι που οι στατιστικοί έχουν χρησιμοποιήσει είναι οι προσεγγίσεις ή οι μετασχηματισμοί για να πετύχουν κανονικότητα. Αυτό, όμως, μπορεί να είναι απαγορευτικό για ένα μεγάλο αριθμό προβλημάτων.

Σήμερα, η υπολογιστική δύναμη οδήγησε στις τεχνικές αναδειγματοληψίας, όπως είναι οι μέθοδοι jackknife και bootstrap. Σκοπός αυτού του κεφαλαίου είναι να παρουσιάσει τον τρόπο που μπορούν να εφαρμοστούν αυτές οι δυο μέθοδοι στην R, είτε ξεκινώντας από τις βασικές έννοιες, είτε χρησιμοποιώντας έτοιμες συναρτήσεις που υπάρχουν στις βιβλιοθήκες της.

11.1 Μέθοδος Jackknife

Η μέθοδος jackknife αποτελείται από δυο βήματα. Πρώτα, παράγονται τα jackknife δείγματα αφαιρώντας την x_i τιμή από το αρχικό δείγμα. Έπειτα, υπολογίζεται η προς εξέταση εκτιμήτρια για κάθε ένα από τα δείγματα jackknife, δηλαδή η

$$\hat{\theta}_i(x_1, \ldots, x_{i-1}, \ldots, x_n).$$

Στη συνέχεια ορίζεται η ψευδοτιμή

$$\hat{\theta}_i^* = n\hat{\theta} - (n-1)\hat{\theta}_i,$$

όπου $\hat{\theta}$ η εκτιμήτρια από το αρχικό δείγμα. Τέλος, η jackknife εκτιμήτρια είναι ίση με

$$J(\hat{\theta}) = \frac{1}{n}\sum \hat{\theta}_i^*$$

με τυπικό σφάλμα

$$\hat{\sigma}_{jack}(\hat{\theta}) = \left[\frac{1}{n(n-1)}\sum_{i=1}^{n}\left(\hat{\theta}_i^* - \hat{\theta^*}(.)\right)^2\right]^{1/2},$$

όπου

$$\hat{\theta^*}(.) = \frac{1}{n}\sum_{i=1}^{n}\hat{\theta}_i^*.$$

Το προσεγγιστικό 95% διάστημα εμπιστοσύνης για την προς εκτίμηση παράμετρο δίνεται από

$$J(\hat{\theta}) \pm t_{0.975,n-1}\cdot\hat{\sigma}_{jack}(\hat{\theta})$$

Θα εξεταστεί τώρα πως μπορεί να προγραμματιστούν τα πιο πάνω στην R για να γίνει η εκτίμηση του συντελεστή μεταβλητότητας $CV = \sqrt{Var(x)}/\bar{x}$ μαζί με το διάστημα εμπιστοσύνης του για ένα δείγμα με 25 παρατηρήσεις :

```
8.26    6.33    10.4    5.27    5.35    5.61    6.12    6.19
5.2     7.01    8.74    7.78    7.02    6       6.5     5.8
5.12    7.41    6.52    6.21    12.28   5.6     5.38    6.6
8.74
```

Αρχικά, εισάγονται τα δεδομένα στην R στη μορφή διανύσματος και μετά ορίζεται η συνάρτηση για τον υπολογισμό του συντελεστή μεταβλητότητας

```
> x <-c(8.26, 6.33, 10.4, 5.27, 5.35, 5.61, 6.12, 6.19, 5.2,
+ 7.01, 8.74, 7.78, 7.02, 6, 6.5, 5.8, 5.12, 7.41, 6.52, 6.21,
+ 12.28, 5.6, 5.38, 6.6, 8.74)
> CV<-function(x) {sqrt(var(x))/mean(x)}
> CV(x)
[1] 0.2524712
```

Στη συνέχεια, προχωρούμε με τον κώδικα υπολογισμού της jackknife εκτιμήτρια μαζί με το τυπικό της σφάλμα.

```
> jack <- numeric(length(x)-1)
> pseudo <- numeric(length(x))
> for (i in 1:length(x))
```

```
+ {
+ jack<-x[-i]
+ pseudo[i]<-length(x)*CV(x)-(length(x)-1)*CV(jack)
+ }
> jack.estim<-mean(pseudo)
> jack.estim
[1] 0.2617376
> jack.se<-sqrt(var(pseudo)/length(x))
> jack.se
[1] 0.05389943
```

Η πρώτη εντολή καθορίζει στην R ότι θα δημιουργηθούν τα δείγματα jackknife, `jack`, τα οποία περιέχουν $n-1$ παρατηρήσεις. Το δεύτερο διάνυσμα `pseudo` είναι αυτό που θα περιέχει τις n ψευδοτιμές. Με την εντολή `for` δημιουργείται ο βρόγχος με τον οποίο θα κατασκευαστούν οι ψευδοτιμές. Για κάθε i δημιουργείται το jackknife δείγμα αφαιρώντας την x_i παρατήρηση από το αρχικό δείγμα, και στη συνέχεια υπολογίζεται η i ψευδοτιμή. Με την εντολή `mean(pseudo)` υπολογίζεται η jackknife εκτιμήτρια με τυπικό σφάλμα το `jack.se`.

Το άνω φράγμα του προσεγγιστικού 95% διαστήματος εμπιστοσύνης για το συντελεστή μεταβλητότητας υπολογίζεται στην R από

```
> jack.estim+qt(0.975,length(x)-1)*jack.se
[1] 0.3729806
```

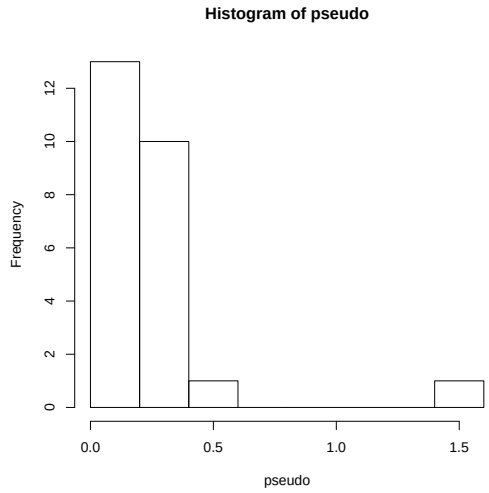
ενώ το κάτω φράγμα από

```
> jack.estim-qt(0.975,length(x)-1)*jack.se
[1] 0.1504947
```

Η μορφή των ψευδοτιμών από την μέθοδο jackknife φαίνεται από το ιστόγραμμα στο Σχήμα 11.1.

```
>hist(pseudo)
```

Ως τώρα παρουσιάστηκε η μέθοδος jackknife ξεκινώντας από τις βασικές της έννοιες. Το επόμενο παράδειγμα παρουσιάζει πως μπορεί να εφαρμοστεί η μέθοδος χρησιμοποιώντας την εντολή `jackknife`, η οποία βρίσκεται στη βιβλιοθήκη `bootstrap` της R, για τον υπολογισμό ενός διαστήματος εμπιστοσύνης για τη μέγιστη τιμή από έξι τιμές από την ομοιόμορφη κατανομή.



Σχήμα 11.1: Ιστόγραμμα ψευδοτιμών jackknife.

```
> library(bootstrap)
> x1<-runif(6)
> x1
[1] 0.3180501 0.6395107 0.2261756 0.2970479 0.4609984 0.8353474
> jack1<-jackknife(x1,max)
> jack1
$jack.se
[1] 0.1631973

$jack.bias
[1] -0.1631973

$jack.values
[1] 0.8353474 0.8353474 0.8353474 0.8353474 0.8353474 0.6395107

$call
jackknife(x = x1, theta = max)

> estim.jack<-mean(jack1$jack.values)
> estim.jack
```

```
[1] 0.802708
> bias<-jack1$jack.bias
> quantile(jack1$jack.values,c(0.025,0.05,0.95,0.975))
     2.5%        5%       95%     97.5%
0.6639903 0.6884699 0.8353474 0.8353474
> estim.jack+qt(0.975,length(x)-1)*jack1$jack.se
[1] 1.139531
> estim.jack-qt(0.975,length(x)-1)*jack1$jack.se
[1] 0.4658853
```

Οι ψευδοτιμές παίρνονται με την εντολή `jack1$jack.values` και για να υπολογιστεί η jackknife εκτιμήτρια παίρνουμε τη μέση τους τιμή. Το τυπικό σφάλμα της εκτιμήτριας παίρνεται με την εντολή `jack1$jack.se` ενώ η εντολή `jack1$jack.bias` δίνει την μεροληψία της εκτιμήτριας. Με την εντολή `quantile` λαμβάνονται τα εμπειρικά ποσοστημόρια των ψευδοτιμών και με τις τελευταίες δυο εντολές υπολογίζεται το προσεγγιστικό 95% διάστημα εμπιστοσύνης.

11.2 Μέθοδος Bootstrap

Η μέθοδος bootstrap βασίζεται στη δημιουργία B νέων δειγμάτων με ίδιο μέγεθος με το αρχικό δείγμα. Τα δείγματα αυτά δημιουργούνται με δειγματοληψία με επανάθεση από το αρχικό δείγμα. Η εκτιμήτρια της παραμέτρου που μας ενδιαφέρει υπολογίζεται για το κάθε ένα από τα B δείγματα bootstrap και παράγουν την κατανομή bootstrap της εκτιμήτριας. Βασική προϋπόθεση είναι ότι οι παρατηρήσεις του αρχικού δείγματος απεικονίζουν όλον τον πληθυσμό.

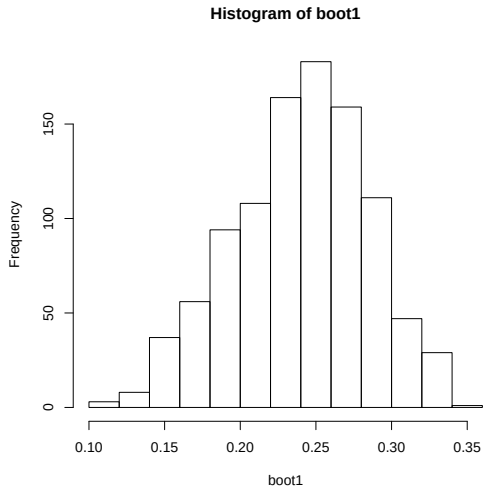
Στην R μπορούν να χρησιμοποιηθούν διάφορες εντολές για τον υπολογισμό των bootstrap εκτιμητριών όπως και το διάστημα εμπιστοσύνης για τη παράμετρο. Ο επόμενος κώδικας παρουσιάζει τον τρόπο εκτίμησης του συντελεστή μεταβλητότητας και την κατασκευή του διαστήματος εμπιστοσύνης με τη μέθοδο bootstrap χρησιμοποιώντας τα προηγούμενα δεδομένα.

```
> boot1 <-numeric(1000)
> for (i in 1:1000)
+ {
+ boot1[i] <- CV(sample(x,replace=T))
+ }
> boot.estim<-mean(boot1)
```

```
> boot.se<-sqrt(var(boot1))
> hist(boot1)
> quantile(boot1,0.975)
   97.5%
0.315552
> quantile(boot1,0.025)
     2.5%
0.1485921
> bias <- mean(boot1) - CV(x)
> CV(x) - bias
[1] 0.2646007
> CV(x) - bias - 1.96*boot.se
[1] 0.1772671
> CV(x) - bias + 1.96*boot.se
[1] 0.3519343
```

Η πρώτη εντολή καθορίζει στην R το διάνυσμα στο οποίο θα φυλαχθούν οι εκτιμήτριες για το συντελεστή μεταβλητότητας για κάθε bootstrap δείγμα. Η δημιουργία κάθε bootstrap δείγματος γίνεται με την εντολή `sample(x,replace=T)`. Με το βρόγχο `for` υπολογίζεται η εκτιμήτρια του συντελεστή μεταβλητότητας για 1000 bootstrap δείγματα. Η bootstrap εκτιμήτρια δίνεται παίρνοντας τη μέση τιμή όλων των εκτιμητριών από τα bootstrap δείγματα, `mean(boot1)`, ενώ το τυπικό της σφάλμα υπολογίζεται από `sqrt(var(boot1))`. Στη συνέχεια δίνεται η εντολή για κατασκευή του ιστογράμματος των εκτιμητριών από τα bootstrap δείγματα για να παρατηρηθεί η κατανομή τους, η οποία δε φαίνεται να διαφέρει πολύ από την κανονική (βλέπε Σχήμα 11.2). Επίσης, δίνονται τα 2.5% και 97.5% εμπειρικά ποσοστημόριά τους, τα οποία ορίζουν και το εμπειρικό 95% διάστημα εμπιστοσύνης. Τέλος, υπολογίζεται η μεροληψία του συντελεστή μεταβλητότητας του αρχικού δείγματος πριν την εφαρμογή της μεθόδου bootstrap για να κατασκευαστεί στη συνέχεια το 95% προσεγγιστικό διάστημα εμπιστοσύνης, υποθέτοντας κανονικότητα.

Πιο κάτω θα παρουσιαστούν δυο παραδείγματα της μεθόδου bootstrap χρησιμοποιώντας τη βιβλιοθήκη `boot` της R. Το πρώτο παράδειγμα αναφέρεται στην εκτίμηση του συντελεστή συσχέτισης, ενώ το δεύτερο στην εκτίμηση των συντελεστών παλινδρόμησης.



Σχήμα 11.2: Ιστόγραμμα εκτιμητριών από τα bootstrap δείγματα.

11.3 Εκτίμηση Συντελεστή Συσχέτισης

Έστω το παράδειγμα από τους Efron και Tibshirani (1993) στο οποίο 82 σχολές νομικής συμμετείχαν σε μια μελέτη για την πρακτική εισδοχής των φοιτητών. Για κάθε μια από αυτές τις σχολές, 15 σχολεία επιλέγηκαν τυχαία και εξετάστηκε η συσχέτιση μεταξύ των αποτελεσμάτων της εξέτασης LSAT και του μέσου όρου (GPA) βάση της τάξης του 1973. Η bootstrap ανάλυση στην R έγινε με την εντολή `boot`, η οποία βρίσκεται στην ομώνυμη βιβλιοθήκη, όπως πιο κάτω

```
> library("boot")
> school<-1:15
> lsat<-c(576,635,558,578,666,580,555,661,651,605,653,575,545,572,594)
> gpa<-c(3.39,3.30,2.81,3.03,3.44,3.07,3.00,3.43,3.36,3.13,3.12,2.74,
+ 2.76,2.88,2.96)
> law.data <- data.frame(School=school, LSAT=lsat, GPA=gpa)
> correl<-function(data,indices)
+ {
+  data<-law.data[indices,]
+ cor(data[,2],data[,3])
+ }
> boot.obj1 <- boot(law.data, correl, 1000)
```

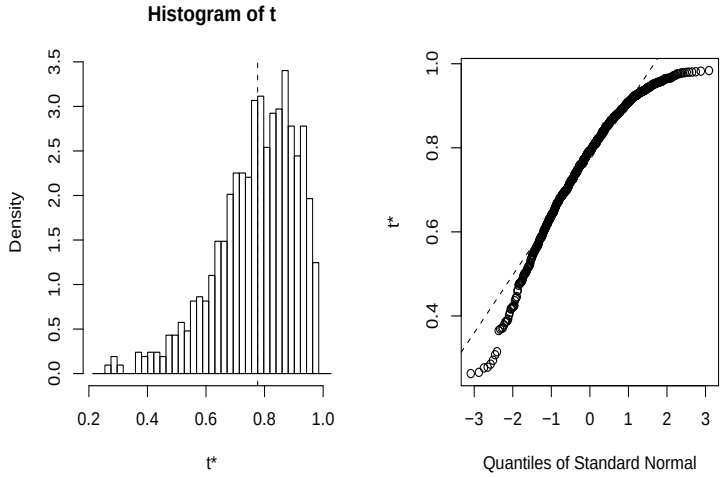
```
> boot.obj1

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = law.data, statistic = correl, R = 1000)


Bootstrap Statistics :
     original       bias    std. error
t1* 0.7763745 -0.005066455   0.1371331
> boot.ci(boot.obj1,type=c("norm","perc","bca"),conf=c(0.90,0.95))
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = boot.obj1, conf = c(0.9, 0.95), type = c("norm",
    "perc", "bca"))

Intervals :
Level      Normal             Percentile            BCa
90%   ( 0.5559,  1.0070 )   ( 0.5071,  0.9510 )   ( 0.3852,  0.9265 )
95%   ( 0.5127,  1.0502 )   ( 0.4245,  0.9644 )   ( 0.2788,  0.9407 )
Calculations and Intervals on Original Scale
Some BCa intervals may be unstable
> plot(boot.obj1)
```

Στην αρχή κατασκευάζεται η στατιστική συνάρτηση (συντελεστής συσχέτισης) με τέτοιο τρόπο έτσι ώστε να μπορεί να χρησιμοποιηθεί στην εντολή `boot`. Η παράμετρος `data` της συνάρτησης καθορίζει το πλαίσιο δεδομένων (δείγμα), ενώ η παράμετρος `indices` θα επιτρέψει στην εντολή `boot` να διαλέξει το δείγμα bootstrap από το αρχικό δείγμα με δειγματοληψία με επανάθεση. Η εντολή `boot` δημιουργεί 1000 συντελεστές συσχέτισης για τα δεδομένα `law.data`. Τα αποτελέσματα της εντολής `boot` δίνουν την αρχική εκτιμήτρια για τον συντελεστή συσχέτισης (πριν την εφαρμογή της μεθόδου) μαζί με την μεροληψία και το τυπικό της σφάλμα. Σημειώνεται εδώ ότι η bootstrap εκτιμήτρια υπολογίζεται αφαιρώντας

τη μεροληψία από την αρχική εκτιμήτρια. Η εντολή `boot.ci` δίνει τα διαστήματα εμπιστοσύνης για το συντελεστή συσχέτισης. Στο παράδειγμα επιλέγηκαν τα ακόλουθα διαστήματα εμιστοσύνης με επίπεδα εμπιστοσύνης 90% και 95%:

1. το προσεγγιστικό διάστημα εμπιστοσύνης με την κανονική (Normal),
2. το εμπειρικό διάστημα εμπιστοσύνης χρησιμοποιώντας ποσοστημόρια (Percentile),
3. το διάστημα εμπιστοσύνης χρησιμοποιώντας τα προσαρμοσμένα ποσοστημόρια λαμβάνοντας υπόψην τη διόρθωση της μεροληψίας (BCa).

Παρατηρώντας το ιστόγραμμα και το QQ-γράφημα (Σχήμα 11.3), τα οποία κατασκευάζονται με την εντολή `plot` και όρισμα το αντικείμενο `boot`, φαίνεται ότι οι bootstrap εκτιμήτριες δεν ακολουθούν την κανονική κατανομή. Συνεπώς, είναι καλύτερο να χρησιμοποιηθούν τα εμπειρικά διαστήματα εμπιστοσύνης, παρά το προσεγγιστικό με τη βοήθεια της κανονικής.



Σχήμα 11.3: Ιστόγραμμα και QQ γράφημα των 1000 εκτιμητριών bootstrap για τον συντελεστή συσχέτισης.

11.4 Συντελεστές Παλινδρόμησης

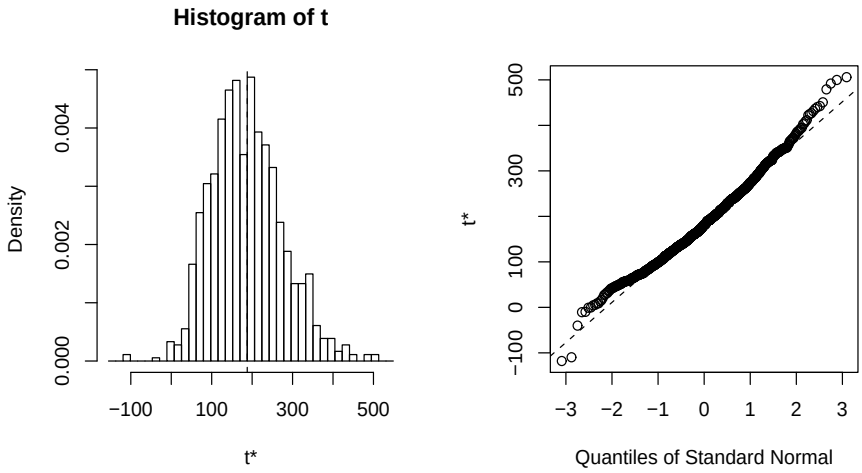
Το ακόλουθο παράδειγμα προβάλλει τον τρόπο εκτίμησης των συντελεστών παλινδρόμησης με την μέθοδο bootstrap χρησιμοποιώντας τα δεδομένα από την

προηγούμενη ενότητα.

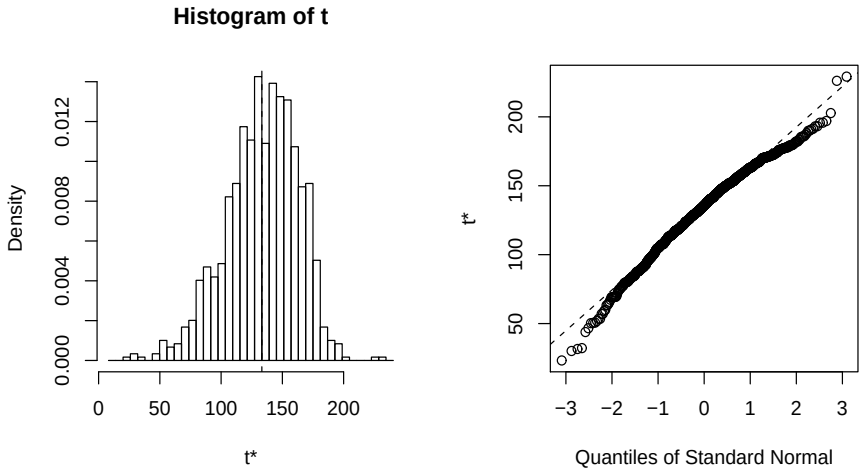
```
> regcoef<-function(data,indices)
+ {
+  data<-law.data[indices,]
+ mod<-lm(LSAT~GPA,data)
+ coef(mod)
+ }
> boot.obj2 <- boot(law.data,regcoef,1000)
> boot.obj2

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = law.data, statistic = regcoef, R = 1000)


Bootstrap Statistics :
    original     bias    std. error
t1* 187.8996 -0.5759099    88.19863
t2* 133.2509  0.2066162    29.52671
> boot.ci(boot.obj2,index=1,type=c("norm","perc","bca"),
+ conf=c(0.90,0.95))
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = boot.obj2, conf = c(0.9, 0.95), type = c("norm",
    "perc", "bca"), index = 1)

Intervals :
Level      Normal             Percentile            BCa
90%   ( 43.4, 333.5 )   ( 58.3, 343.6 )   ( 74.8, 387.7 )
95%   ( 15.6, 361.3 )   ( 43.8, 376.4 )   ( 62.3, 441.7 )
Calculations and Intervals on Original Scale
Some BCa intervals may be unstable
```

```
> boot.ci(boot.obj2,index=2,type=c("norm","perc","bca"),
+ conf=c(0.90,0.95))
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = boot.obj2, conf = c(0.9, 0.95), type = c("norm",
    "perc", "bca"), index = 2)

Intervals :
Level      Normal             Percentile            BCa
90%   ( 84.5, 181.6 )   ( 81.0, 176.0 )   ( 63.5, 170.3 )
95%   ( 75.2, 190.9 )   ( 69.3, 180.9 )   ( 45.0, 173.8 )
Calculations and Intervals on Original Scale
Some BCa intervals may be unstable
> plot(boot.obj2,index=1)
> plot(boot.obj2,index=2)
```

Τα πιο πάνω αποτελέσματα δείχνουν ότι η bootstrap εκτιμήτριες για το σταθερό όρο και την κλίση να είναι ίσες με 133.3 και 187.9, αντίστοιχα. Συγκεκριμένα, η κλίση είναι θετική όπως αναμενόταν από τα προηγούμενα αποτελέσματα (γιατι;). Για τα διαστήματα εμπιστοσύνης για τον καθένα συντελεστή παλινδρόμησης, τίθεται στην εντολή `boot.ci` το όρισμα `index` να είναι ίσο με 1 και 2, αντίστοιχα. Το ίδιο όρισμα δίνεται και για την κατασκευή των γραφημάτων της μεθόδου (Σχήματα 11.4 και 11.5).



Σχήμα 11.4: Ιστόγραμμα και QQ γράφημα των 1000 εκτιμητριών bootstrap για τον σταθερό όρο.



Σχήμα 11.5: Ιστόγραμμα και QQ γράφημα των 1000 εκτιμητριών bootstrap για την κλίση.

Κεφάλαιο 12

Ασκήσεις Μέρους Ι

1. Οι ακόλουθες 10 παρατηρήσεις δίνουν την κάλυψη από χιόνια στην Ευρασία κατά το μήνα Οκτώβριο για τα χρόνια 1970-79. (Κάλυψη από χιόνια σε εκατομμύρια τετραγωνικά χιλιόμετρα):

year	snow.cover
1970	6.5
1971	12.0
1972	14.9
1973	10.0
1974	10.7
1975	7.9
1976	21.9
1977	12.5
1978	14.5
1979	9.2

   (α΄) Εισάγετε τα δεδομένα στην R.

   (β΄) Κατασκευάστε το γράφημα της μεταβλητής `snow.cover` συναρτήσει της μεταβλητής `year`.

   (γ΄) Χρησιμοποιείστε την εντολή `hist()` για να κατασκευάσετε το ιστόγραμμα για τη μεταβλητή `snow.cover`.

   (δ΄) Επαναλάβετε τα πιο πάνω αφού πάρετε το λογαριθμικό μετασχηματισμό για την μεταβλητή `snow cover`.

2. Για κάθε ένα από τους ακόλουθους κώδικες να προβλέψετε το αποτέλεσμα. Στη συνέχεια να κάνετε τους υπολογισμούς :

   (α΄)
   ```
   answer <- 0
   for (j in 3:5){ answer <- j+answer }
   ```

   (β΄)
   ```
   answer<- 10
   for (j in 3:5){ answer <- j+answer }
   ```

   (γ΄)
   ```
   answer <- 10
   for (j in 3:5){ answer <- j*answer }
   ```

3. Χρησιμοποιήστε την εντολή `prod()` για να κάνετε το 2γ΄ πιο πάνω. Πώς αναμένετε να δουλέψει η εντολή; Δοκιμάστε το! (Για βοήθεια γράψτε `?prod`).

4. Προσθέστε όλους τους αριθμούς από το 1 μέχρι και το 100 με 2 τρόπους, χρησιμοποιώντας πρώτα το `for` και μετά το `sum`. Τώρα εφαρμόστε την εντολή στην ακολουθία 1:100. Ποια τα αποτελέσματα;

5. Πολλαπλασιάστε όλους τους αριθμούς από το 1 μέχρι και το 50 με 2 τρόπους, χρησιμοποιώντας πρώτα το `for` και το μετά `prod`.

6. Ο όγκος σφαίρας με ακτίνα r δίνεται από $\frac{4}{3}\pi r^3$. Για σφαίρες με ακτίνα 3, 4, 5, ..., 20, να βρείτε τον αντίστοιχο όγκο και κατασκευάστε πλαίσιο δεδομένων με στήλες *ακτίνα* (radius) και *όγκος* (volume).

7. Χρησιμοποιείστε την εντολή `sapply()` για να εφαρμόσετε την εντολή `is.factor` σε κάθε στήλη του πλαισίου δεδομένων `tinting` που μπορείτε να βρείτε στο πακέτο `DAAG`. Για τις στήλες οι οποίες αναγνωρίστηκαν ως παράγοντες, προσδιορίστε τα επίπεδα. Ποιες στήλες είναι διατακτικοί παράγοντες; (Χρησιμοποιείστε `is.ordered()`).

8. Κατασκευάστε τη γραφική παράσταση του βάρους εγκεφάλου (`brain`) συναρτήσει του βάρους σώματος (`body`) για το πλαίσιο δεδομένων `Animals` από τη βιβλιοθήκη `MASS`. Ονομάστε τους άξονες ανάλογα. Ονομάστε επίσης το σημείο για το ζώο με το μεγαλύτερο σωματικό βάρος.

9. Επαναλάβετε το γράφημα 8, αλλά αυτή τη φορά της μεταβλητής `log(brain)` συναρτήσει της μεταβλητής `log(body)` και ονομάστε τους άξονες ανάλογα

10. Επαναλάβετε τα γραφήματα 8 και 9, αλλά αυτή τη φορά τοποθετείστε τα γραφήματα σε μια σελίδα, το ένα δίπλα από το άλλο.

11. Ελέγξτε την κατανομή του μήκους κεφαλής (`hdlngth`) από το πλαίσιο δεδομένων `possum` που μπορείτε να βρείτε στο πακέτο `DAAG`. Κατασκευάστε και συγκρίνεται τα ακόλουθα γραφήματα:

    (α') ιστόγραμμα,

    (β') δενδροδιάγραμμα (stem and leaf),

    (γ') QQ-γράφημα και

    (δ') γράφημα πυκνότητας πιθανότητας.

    Ποια τα πλεονεκτήματα και ποια τα μειονεκτήματα του κάθε γραφήματος.

12. Δοκιμάστε `x <- rnorm(10)`. Τυπώστε τους αριθμούς που παίρνετε. Δείτε τη βοήθεια για την εντολή `rnorm`. Να παράγετε ένα δείγμα μεγέθους 10 από την κανονική κατανομή με μέση τιμή 170 και τυπική απόκλιση 4.

13. Χρησιμοποιείστε την εντολή `mfrow()` για να δημιουργήσετε μια 3 επί 4 διάταξη γραφημάτων. Στην πρώτη γραμμή να παρουσιάσετε τα QQ-γραφήματα τεσσάρων τυχαίων δειγμάτων μεγέθους 10 από την τυπική κανονική. Στη δεύτερη και τρίτη γραμμή να κάνετε το ίδιο για τυχαία δείγματα από την κανονική κατανομή μεγέθους 100 και 1000, αντίστοιχα. Σχολιάστε τις αλλαγές στο γράφημα όσο αλλάζει το μέγεθος του δείγματος.

14. Η εντολή `runif()` μπορεί να χρησιμοποιηθεί για τη δημιουργία δείγματος από την ομοιόμορφη κατανομή, εξ ορισμού για το διάστημα από το 0 ως το 1. Δοκιμάστε `x<-runif(10)`, και τυπώστε το αποτέλεσμα. Στη συνέχεια επαναλάβετε την άσκηση 13 παίρνοντας δείγματα από την ομοιόμορφη κατανομή. Τι σχήμα έχουν τα σημεία;

15. Δοκιμάστε την προηγούμενη άσκηση για τη X^2 κατανομή με 1 βαθμό ελευθερίας και τη t κατανομή με 2 βαθμούς ελευθερίας με τις εντολές `rchisq()` και `rt()`, αντίστοιχα. Οι βαθμοί ελευθερίας δίνονται σαν δεύτερο όρισμα στις εντολές αυτές.

16. Εξετάστε την κατανομή των 2 πρώτων στηλών του πλαισίου δεδομένων `hills` χρησιμοποιώντας

    (α') ιστογράμματα,

    (β') γράφημα πυκνότητας πιθανότητας,

    (γ') QQ-γραφήματα.

    Επαναλάβετε τα πιο πάνω παίρνοντας του λογάριθμους των 2 στηλών.

Μέρος ΙΙ

Στατιστικές Μέθοδοι στην R –ΙΙ

Κεφάλαιο 13

Ειδικά Γραφήματα

13.1 Γραφήματα Trellis

Τα γραφήματα Trellis βρίσκονται στη βιβλιοθήκη *lattice* της R. Ο κύριος σκοπός τους είναι να δημιουργήσουν πολλαπλά γραφήματα ανά σελίδα στα οποία παρουσιάζεται η σχέση μεταξύ μεταβλητών, εξαρτημένων με μία ή περισσότερες μεταβλητές. Τα γραφήματα παράγονται το ένα δίπλα στο άλλο, συνήθως για κάθε επίπεδο μίας κατηγορικής μεταβλητής (παράγοντας). Υπάρχουν γραφήματα Trellis για διάφορα είδη γραφικών παραστάσεων, τα οποία παρουσιάζονται στον Πίνακα 13.1.

Το όρισμα των συναρτήσεων καθορίζει τη μεταβλητή ή τη σχέση των μεταβλητών που θα παρουσιαστούν ανά επίπεδο των εξαρτημένων παραγόντων. Για παράδειγμα, το όρισμα `~x|A` σημαίνει να κατασκευαστεί το γράφημα της `x` για κάθε επίπεδο του παράγοντα `A`, ενώ, το όρισμα `y~x|A*B` σημαίνει να κατασκευαστεί το γράφημα της `y` συναρτήσει της `x` ξεχωριστά για κάθε συνδυασμό των επιπέδων των παραγόντων `A` και `B`.

Στη συνέχεια παρατίθενται μερικά παραδείγματα. Τα δεδομένα, `mtcars`, είναι παρμένα από το αμερικάνικο περιοδικό Motor Trend που κυκλοφόρησε το 1974 και παρουσιάζουν την κατανάλωση καυσίμων και δέκα τεχνικά χαρακτηριστικά για 32 μοντέλα αυτοκινήτων.

```
> library(lattice)
> attach(mtcars)
> names(mtcars)
 [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
```

Συνάρτηση	Περιγραφή
barchart()	Ραβδόγραμμα
bwplot()	Κυτιογράφημα
cloud()	Τρισδιάστατο διάγραμμα διασποράς
contourplot()	Γράφημα ισοϋψών
densityplot()	Γράφημα συνάρτησης πυκνότητας πιθανότητας
dotplot()	Dot plot
histogram()	Ιστόγραμμα
levelplot()	Γράφημα Επιπέδων
parallel()	Γράφημα παράλληλων συντεταγμένων
splom()	Πίνακας διαγραμμάτων διασποράς
stripplot()	Strip plot
xyplot()	Διάγραμμα διασποράς
wireframe()	Τρισδιάστατες επιφάνειες
qqmath()	QQ-γράφημα

Πίνακας 13.1: Είδη γραφημάτων Trellis

```
[11] "carb"
```

Όπως αναφέρθηκε πιο πάνω, τα γραφήματα Trellis κατασκευάζονται σε σχέση με κατηγορικές μεταβλητές και γι' αυτό στην αρχή καθορίζονται οι μεταβλητές gear (αριθμός ταχυτήτων) και gear (αριθμός κυλίνδρων) ως παράγοντες.

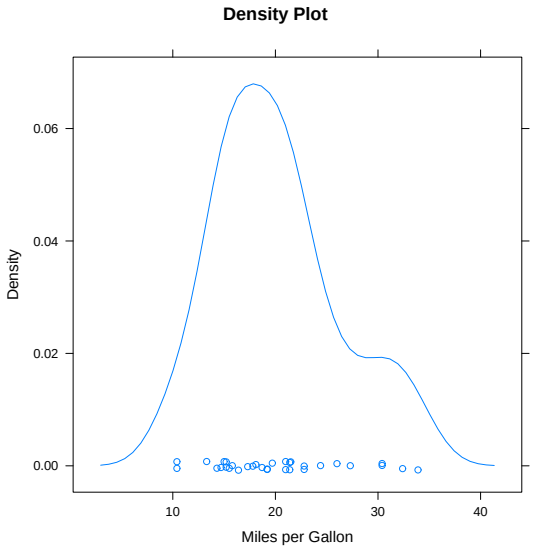
```
> gear.f<-factor(gear,levels=c(3,4,5),labels=c("3gears","4gears","5gears"))
> cyl.f <-factor(cyl,levels=c(4,6,8),labels=c("4cyl","6cyl","8cyl"))
```

Το πρώτο γράφημα που κατασκευάζεται είναι το γράφημα συνάρτησης πυκνότητας πιθανότητας της κατανάλωσης καυσίμων, δηλαδή τα μίλια που διανύουν τα αυτοκίνητα ανά γαλόνι (Σχήμα 13.1).

```
> densityplot(~mpg,main="Density Plot",xlab="Miles per Gallon")
```

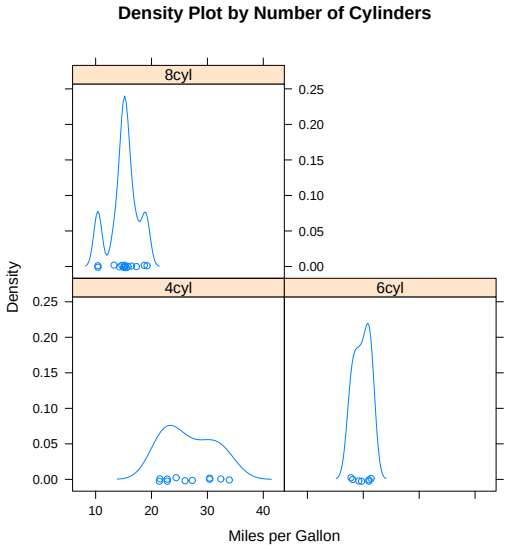
Στη συνέχεια κατασκευάζεται το ίδιο γράφημα αλλά για κάθε επίπεδο του παράγοντα cyl.f (Σχήμα 13.2).

```
> densityplot(~mpg|cyl.f,main="Density Plot by Number of Cylinders",
+ xlab="Miles per Gallon")
```



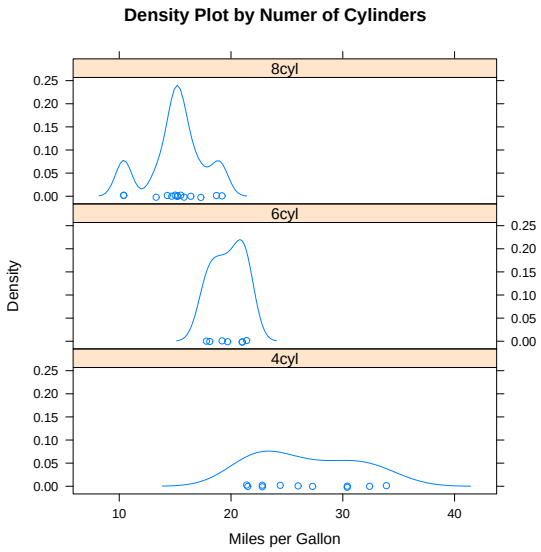
Σχήμα 13.1: Γράφημα συνάρτησης πυκνότητας πιθανότητας της κατανάλωσης καυσίμων.

Για να αλλαχθεί ο τρόπος παρουσίασης των γραφημάτων χρησιμοποιούμε το όρισμα `layout`. Στο επόμενο παράδειγμα επιλέχθηκε η διάταξη μίας στήλης και τριών γραμμών (Σχήμα 13.3).



Σχήμα 13.2: Γράφημα συνάρτησης πυκνότητας πιθανότητας της κατανάλωσης καυσίμων ανά αριθμό κυλίνδρων.

```
> densityplot(~mpg|cyl.f,main="Density Plot by Numer of Cylinders",
+ xlab="Miles per Gallon",layout=c(1,3))
```



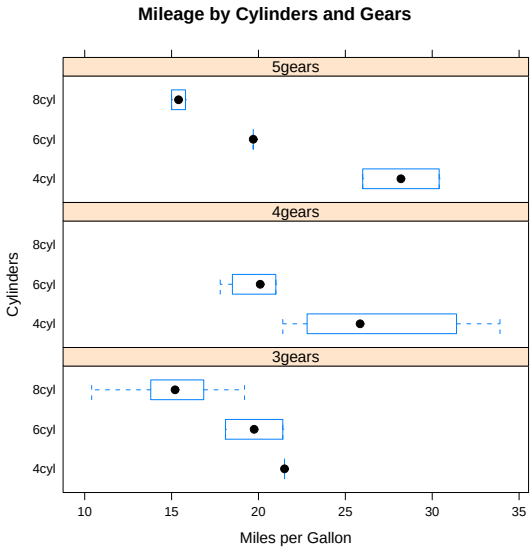
Σχήμα 13.3: Γράφημα συνάρτησης πυκνότητας πιθανότητας της κατανάλωσης καυσίμων ανά αριθμό κυλίνδρων.

Προχωρώντας, κατασκευάζεται το κυτιογράφημα της κατανάλωσης καυσίμων για κάθε συνδυασμό των επιπέδων των παραγόντων `cyl.f` και `gear.f` (Σχήμα 13.4).

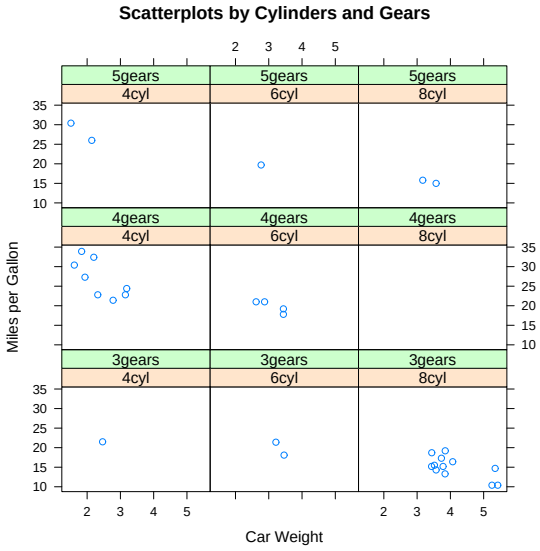
```
> bwplot(cyl.f~mpg|gear.f,ylab="Cylinders", xlab="Miles per Gallon",
+ main="Mileage by Cylinders and Gears",layout=(c(1,3)))
```

Χρησιμοποιώντας τη συνάρτηση `xyplot` κατασκευάζεται και το διάγραμμα διασποράς της κατανάλωσης καυσίμων συναρτήσει του βάρους του αυτοκινήτου, για κάθε συνδυασμό αριθμού κυλίνδρων και ταχυτήτων (Σχήμα 13.5.

```
> xyplot(mpg~wt|cyl.f*gear.f,main="Scatterplots by Cylinders and Gears",
+ ylab="Miles per Gallon", xlab="Car Weight")
```



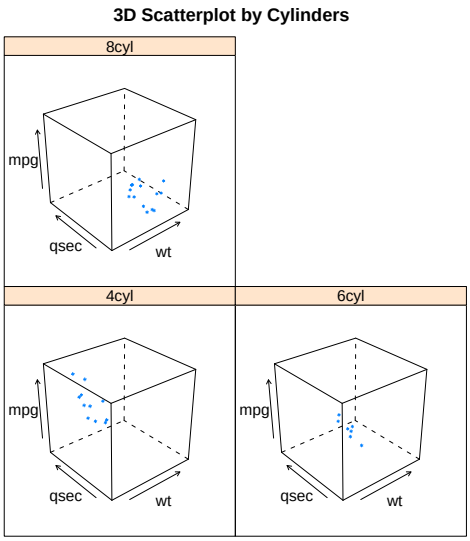
Σχήμα 13.4: Κυτιογράφημα της κατανάλωσης καυσίμων ανά αριθμό κυλίνδρων και αριθμό ταχυτήτων.



Σχήμα 13.5: Διάγραμμα διασποράς της κατανάλωσης καυσίμων συναρτήσει του βάρους του αυτοκινήτου ανά αριθμό κυλίνδρων και αριθμό ταχυτήτων.

Με τη συνάρτηση cloud επεκτείνεται το προηγούμενο γράφημα στις τρεις διαστάσεις. Ως τρίτη διάσταση ορίζεται η μεταβλητή qsec, η οποία παρουσιάζει το χρόνο που χρειάζεται ένα αυτοκίνητο για να καλύψει απόσταση ίση με το $1/4$ του μίλι. Το γράφημα παρουσιάζεται στο Σχήμα 13.6.

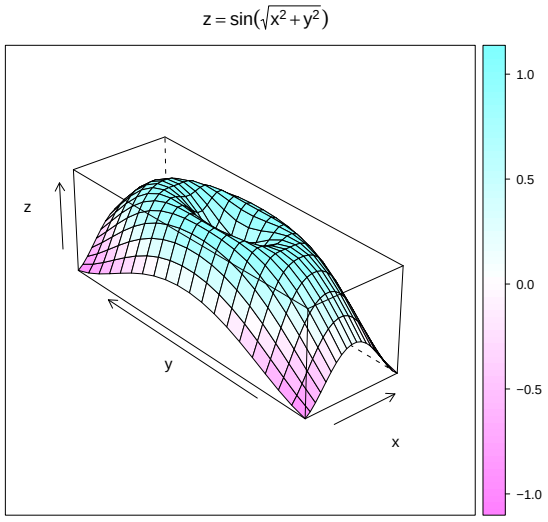
```
> # 3d scatterplot by factor level
> cloud(mpg~wt*qsec|cyl.f,main="3D Scatterplot by Cylinders")
```



Σχήμα 13.6: Τρισδιάστατο διάγραμμα διασποράς της κατανάλωσης καυσίμων συναρτήσει του βάρους του αυτοκινήτου και του χρόνου κάλυψης του $1/4$ του μίλι, ανά αριθμό κυλίνδρων και αριθμό ταχυτήτων.

Τέλος, βλέπουμε πώς μπορεί να κατασκευαστεί το τρισδιάστατο γράφημα μιας επιφάνειας που ορίζεται από μια διδιάστατη μαθηματική συνάρτηση. Στο παράδειγμά παρουσιάζεται το γράφημα της επιφάνειας $z = sin\left(\sqrt{x^2 + y^2}\right)$ (Σχήμα 13.7).

```
> x <- seq(-pi, pi, len = 20)
> y <- seq(-pi, pi, len = 20)
> g <- expand.grid(x = x, y = y)
> g$z <- sin(sqrt(g$x^2 + g$y^2))
> wireframe(z ~ x * y, g, drape = TRUE, aspect = c(3,1), colorkey = TRUE,
+ main=expression(paste(z==sin(sqrt(x^2+y^2)))))
```



Σχήμα 13.7: Η τρισδιάστατη επιφάνεια $z = sin\left(\sqrt{x^2 + y^2}\right)$.

Κεφάλαιο 14

Μέθοδος Newton-Raphson

Θα συζητήσουμε υπολογισμό της εκτιμήτριας μεγίστης πιθανοφάνειας με τη μέθοδο Newton-Raphson. Αν και υπάρχουν περιπτώσεις για τις οποίες η λύση μπορεί να υπολογιστεί ακριβώς, στα περισσότερα παραδείγματα η Ε.Μ.Π. πρέπει να βρεθεί με αναδρομικές αριθμητικές μεθόδους, όπως η Newton-Raphson. Θα συζητήσουμε τις βασικές αρχές με ένα παράδειγμα.

14.1 Παράδειγμα

Τα δεδομένα αναφέρονται σε ώρες επιβίωσης που μετρούν την αντοχή συγκεκριμένων πλοίων σε συνθήκες πίεσης (Τα δεδομένα παρατίθενται στο Παράρτημα). Ένα συγκεκριμένο μοντέλο που χρησιμοποιείται για ανάλυση δεδομένων επιβίωσης είναι η κατανομή Weibull.

$$f(y;\lambda;\theta) = \frac{\lambda y^{\lambda-1}}{\theta^{\lambda}} e^{-(\frac{y}{\theta})^{\lambda}}, y > 0,$$

όπου,

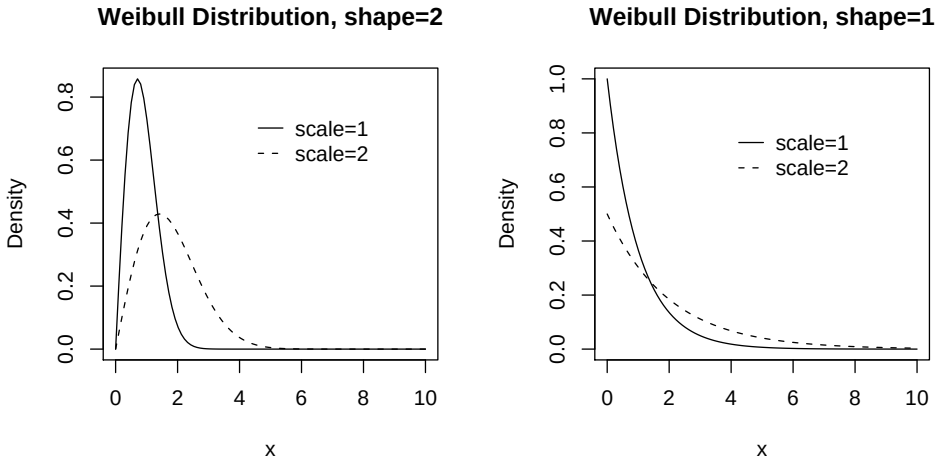
λ: παράμετρος που καθορίζει το σχήμα της κατανομής, και

ϑ: παράμετρος που καθορίζει την κλίμακα

Δίνονται τα γραφήματα της κατανομής Weibull για $\lambda = 1, 2$ και $\theta = 1, 2$ (Σχήμα 14.1) για κατανόηση του σχήματος της συνάρτησης πυκνότητας πιθανότητας.

```
> x=seq(0,10, length=100)
> par(mfrow=c(1,2))
> plot(x, dweibull(x, shape=2, scale=1), type="l", ylab="Density")
> lines(x, dweibull(x, shape=2, scale=2), lty=2)
```

```
> title(main="Weibull Distribution, shape=2")
> leg.names<-c("scale=1","scale=2")
> legend(locator(1),leg.names,lty=1:2,bty="n")

> plot(x, dweibull(x, shape=1, scale=1), type="l", ylab="Density")
> lines(x, dweibull(x, shape=1, scale=2), lty=2)
> title(main="Weibull Distribution, shape=1")
> leg.names<-c("scale=1","scale=2")
> legend(locator(1),leg.names,lty=1:2,bty="n")
```

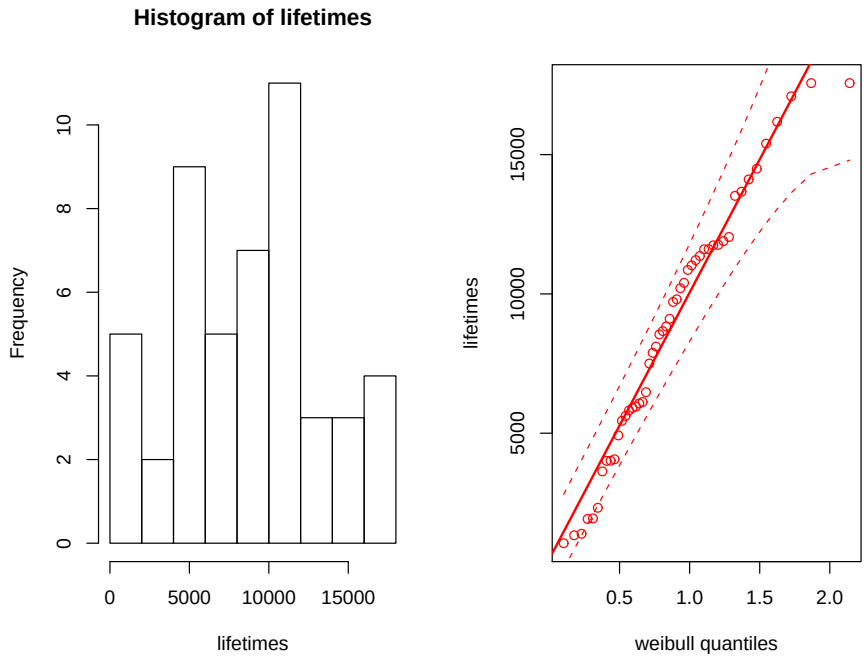


Σχήμα 14.1: Κατανομή Weibull για $\lambda = 1, 2$ και $\theta = 1, 2$.

Επίσης στο Σχήμα 14.2 δίνεται το ιστόγραμμα των σχετικών δεδομένων και το QQ plot με βάση την κατανομή Weibull με $\lambda = 2$.

```
> par(mfrow=c(1,2))
> hist(lifetimes)
> library("car")
> qq.plot(lifetimes, dist="weibull", shape=2)
> qqline(lifetimes, rweibull(49, shape=2))
```

Όπως βλέπουμε από το Σχήμα 14.2, μπορούμε να υποθέσουμε ότι τα δεδομένα ακολουθούν την κατανομή Weibull με $\lambda = 2$ (γαιτί·) Αφού έχουμε αποδεχτεί ότι



Σχήμα 14.2: Ιστόγραμμα και QQ plot των δεδομένων.

η παράμετρος λ είναι γνωστή, έστω τώρα $y_1, \ldots, y_n$ δεδομένα με λ γνωστό (στην περίπτωσή μας $n = 49$). Τότε η από κοινού συνάρτηση πυκνότητας πιθανότητας δίνεται από

$$f(y_1, \ldots, y_n; \theta) = \prod_{i=1}^{n} \frac{\lambda {y_i}^{\lambda-1}}{\theta^{\lambda}} e^{-(\frac{y_i}{\theta})^{\lambda}},$$

οπότε η πιθανοφάνεια δίνεται από

$$L(\theta) = \log f(y_1, \ldots, y_n; \theta) = \sum_{i=1}^{n} \left\{ (\lambda - 1) \log y_i + \log \lambda - \lambda \log \theta - (\frac{y_i}{\theta})^{\lambda} \right\}.$$

Στην R η συνάρτηση λογαριθμικής πιθανοφάνειας $L(\theta)$ μπορεί να οριστεί ως ακολούθως :

```
> loglikelihood <- function(data, theta, lambda=2)
+ { +
```

```
logl<-sum((lambda-1)*log(data)+log(lambda)-lambda*log(theta)-
+    (data/theta)^{lambda})
+ return(logl)
+ }
> theta1 <- seq(7000, 15000, by=100)
> loglik=rep(NA, length(theta1))
> for (i in 1:length(theta1))
+ {
+ loglik[i]=loglikelihood(lifetimes, theta1[i])
+ }
```

Το γράφημα της παρουσιάζεται στο Σχήμα 14.3 και παρατηρούμε ότι υπάρχει τιμή της θ η οποία μεγιστοποιεί την $L(\theta)$. Η συνάρτηση score δίνεται από

$$\frac{dl}{d\theta} = U = \sum_{i=1}^{n} \left\{ -\frac{\lambda}{\theta} + \frac{\lambda {y_i}^{\lambda}}{\theta^{\lambda+1}} \right\} = -\frac{\lambda n}{\theta} + \frac{\lambda \sum_{i=1}^{n} y_i^{\lambda}}{\theta^{\lambda+1}},$$

και στην R ορίζεται ως

```
> get.score <- function(data, theta, lambda=2)
+ { + score<--(lambda*length(data)/theta)+
+ lambda*(sum(data^{lambda}))/(theta^{3})
+ return(score)
+ }
```

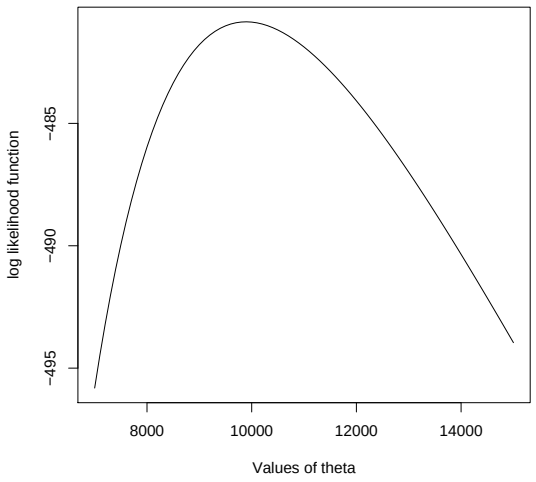
Παρατηρούμε ότι για $\lambda = 2$,

$$U(\theta) = 0 \Rightarrow -\frac{2n}{\theta} + \frac{2\sum_{i=1}^{n} {y_i}^2}{\theta^3} = 0 \Rightarrow \frac{n}{\theta} = \frac{\sum_{i=1}^{n} {y_i}^2}{\theta^3} \Rightarrow \theta^2 = \frac{\sum_{i=1}^{n} {y_i}^2}{n}$$

$$\Rightarrow \theta = \sqrt{\frac{\sum_{i=1}^{n} {y_i}^2}{n}},$$

δηλαδή η Ε.Μ.Π. μπορεί να υπολογιστεί ακριβώς. Θα συγκρίνουμε το ακριβές αποτέλεσμα με εκείνο το οποίο δίνουν οι αναδρομικές μέθοδοι. Πρώτα όμως εξηγούμε τη μέθοδο Newton-Raphson. Γενικά θέλουμε να υπολογίσουμε την τιμή της x, για την οποία $f(x) = 0$. Η εφαπτομένη της $f(x)$ στο σημείο $x^{(m-1)}$ δίνεται από

$$\left[\frac{df}{dx}\right]_{x=x^{(m-1)}} = f'(x^{(m-1)}) = \frac{f(x^{(m)}) - f(x^{(m-1)})}{x^{(m)} - x^{(m-1)}},$$



Σχήμα 14.3: Γράφημα της συνάρτησης λογαριθμικής πιθανοφάνειας.

όπου η απόσταση $x^{(m)} - x^{(m-1)}$ είναι μικρή. Αν το $x^{(m)}$ είναι η λύση της $f(x) = 0$, δηλαδή $f(x^{(m)}) = 0$, έχουμε ότι

$$x^{(m)} = x^{(m-1)} - \frac{f(x^{(m-1)})}{f'(x^{(m-1)})}.$$

Για $m = 1, 2, \ldots,$ και με αρχική τιμή $x^{(1)}$, βρίσκουμε διαδοχικές προσεγγίσεις έτσι ώστε $|x^{(m)} - x^{(m-1)}| < \varepsilon$. Ειδικά, για την εκτιμήτρια μεγίστης πιθανοφάνειας (Ε.Μ.Π),

$$\theta^{(m)} = \theta^{(m-1)} - \frac{U(\theta^{(m-1)})}{U'(\theta^{(m-1)})}.$$

Έχουμε ότι,

$$U(\theta) = -\frac{2n}{\theta} + \frac{2\sum_{i=1}^{n} {y_i}^2}{\theta^3} = 0,$$

και

$$\frac{dU(\theta)}{d\theta} = U'(\theta) = \sum_{i=1}^{n} \left\{ \frac{\lambda}{\theta^2} - \frac{\lambda(\lambda+1){y_i}^{\lambda}}{\theta^{\lambda+2}} \right\} = \frac{2n}{\theta^2} - \frac{2 \cdot 3 \cdot \sum {y_i}^2}{\theta^4},$$

όπου στην τελευταία ισότητα θέτουμε $\lambda = 2$. Αντί να χρησιμοποιήσουμε την $U'(\theta)$, θέτουμε στην παραπάνω αναδρομική σχέση την $E(U'(\theta)) = -J(\theta)$. Μπορεί να

αποδειχτεί ότι

$$J(\theta) = -E(U'(\theta)) = \frac{\lambda^2 n}{\theta^2}.$$

Η συνάρτηση $J(\theta)$ ονομάζεται πληροφορία Fischer και γραφεταί στην R ως

```
> get.information <- function(data, theta, lambda=2)
+ {
+ information <- (lambda^{2}*length(data))/(theta^{2})
+ return(information)
+ }
```

Συνεπώς, καταλήγουμε σε μια τροποποίηση του αλγορίθμου Newton-Raphson, ο οποίος ονομάζεται Fisher scoring.Δηλαδή,

$$\theta^{(m)} = \theta(m-1) + \frac{U(\theta^{(m-1)})}{J(\theta^{(m-1)})},$$

Χρησιμοποιώντας την πιο πάνω αναδρομική σχέση μπορούμε να βρούμε την εκτιμήτρια μέγιστης πιθανοφάνειας για το θ. Το τυπικό σφάλμα για το θ δίνεται από τον τύπο

$$s(\hat{\theta}) = \sqrt{\frac{1}{J}},$$

και ένα 95% διάστημα εμπιστοσύνης δίνεται από

$$\hat{\theta} \pm 1.96 \cdot s(\hat{\theta}).$$

Στη συνέχεια παρουσιάζεται πώς εφαρμόζεται η μέθοδος αυτή στην R για να υπολογιστεί η Ε.Μ.Π. για το θ, το τυπικό του σφάλμα και ένα 95% διάστημα εμπιστοσύνης.

```
> ybar <- mean(lifetimes)
> ybar
[1] 8805.694
> theta <- ybar
> it  <- 0   ####iterative count
> del <- 1   ####iterative adjustment
> while(abs(del) > 0.00001 && (it <- it+1) < 10)
+ {
+ del<-get.score(lifetimes,theta)/get.information(lifetimes,theta)
+ theta <- theta+del
+ cat(it,theta,get.score(lifetimes,theta),
```

```
+ get.information(lifetimes,theta),"\n")
+ }
1 9959.204 -0.0001320064 1.976090e-06
2 9892.402 -4.517492e-07 2.002869e-06
3 9892.177 -5.150392e-12 2.002960e-06
4 9892.177 1.734723e-18  2.002960e-06
> sqrt(mean(lifetimes^{2})) ###exact value
[1] 9892.177
> ##########Confidence Interval
> sderror <- sqrt(1/2.002960e-06)
> sderror
[1] 706.5841
> 9892.177-1.96*sderror;    9892.177+1.96*sderror
[1] 8507.272 [1] 11277.08
```

Από τα παραπάνω, παρατηρούμε ότι οι αναδρομικές σχέσεις που ορίζονται από τον αλγόριθμο Fisher-scoring συγκλίνουν στην ακριβή τιμή της $\hat{\theta}$.

Παράρτημα

Τα δεδομένα που χρησιμοποιούνται σε αυτό το κεφάλαιο για την εύρεση της Ε.Μ.Π. με την μέθοδο Newton-Raphson.

```
   lifetimes
1       1051
2       1337
3       1389
4       1921
5       1942
6       2322
7       3629
8       4006
9       4012
10      4063
11      4921
12      5445
13      5620
```

14	5817
15	5905
16	5956
17	6068
18	6121
19	6473
20	7501
21	7886
22	8108
23	8546
24	8666
25	8831
26	9106
27	9711
28	9806
29	10205
30	10396
31	10861
32	11026
33	11214
34	11362
35	11604
36	11608
37	11745
38	11762
39	11895
40	12044
41	13520
42	13670
43	14110
44	14496
45	15395
46	16179
47	17092
48	17568
49	17568

Κεφάλαιο 15

Ανάλυση της Συνδιακύμανσης

15.1 Ανάλυση της Συνδιακύμανσης

Ανάλυση της Συνδιακύμασης (ANCOVA) είναι ο όρος που χρησιμοποιείται για την ανάλυση ενός γραμμικού μοντέλου όταν κάποιες από τις ανεξάρτητες μεταβλητές είναι παράγοντες και κάποιες συνεχείς. Όπως και με την ανάλυση της διακύμανσης ενδιαφερόμαστε στη σύγκριση μέσων όρων ανάμεσα στα επίπεδα του παράγοντα, αλλά αναγνωρίζουμε το γεγονός ότι η συνεχής μεταβλητή έχει επίδραση στην εξαρτημένη. Το παρακάτω παράδειγμα δείχνει πως εφαρμόζεται η μέθοδος.

Έστω τα ότι έχουμε τα ακόλουθα δεδομένα:

- Y_{jk}: η βαθμολογία για τρεις διαφορετικές μεθόδους, A, B, C, και
- x: η ικανότητα μάθησης πριν την διδασκαλία για 7 μαθητές.

Για να εξετάσουμε κατά πόσον υπάρχουν διαφορές μεταξύ μεθόδων δοθέντος της x, θεωρούμε το πλήρες μοντέλο

$$E(Y_{jk}) = \mu_j + \gamma x_{jk}, \quad j = 1, 2, 3, \quad k = 1, \ldots, 7$$

και το μοντέλο

$$E(Y_{jk}) = \mu + \gamma x_{jk}$$

Παρατηρούμε ότι $j = 1$ αντιστοιχεί στη μέθοδο A, $j = 2$ στη μέθοδο B και $j = 3$ στη μέθοδο C.

Έστω,

$$\mathbf{Y_j} = \begin{bmatrix} Y_{j1} \\ \vdots \\ Y_{j7} \end{bmatrix}$$

και

$$\mathbf{x_j} = \begin{bmatrix} x_{j1} \\ \vdots \\ x_{j7} \end{bmatrix}$$

Το πλήρες μοντέλο δίνεται από

$$E(\mathbf{Y}) = X\boldsymbol{\beta},$$

με

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y_1} \\ \mathbf{Y_2} \\ \mathbf{Y_3} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \gamma \end{bmatrix},$$

και

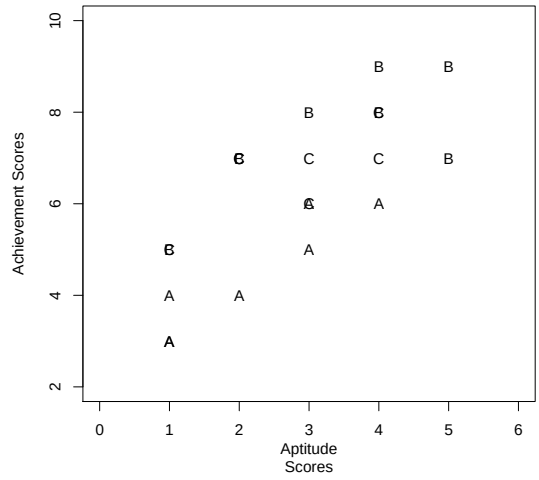
$$X = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{x_1} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{x_2} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{x_3} \end{bmatrix}$$

Η ίδια ανάλυση μπορεί να γίνει και για το μικρότερο μοντέλο.

Στη συνέχεια παρουσιάζεται ο τρόπος που επιτυγχάνεται η ανάλυση της συνδιακύμανσης στην R. Στην αρχή παρουσιάζεται το διάγραμμα διασπορών της ικανότητας μάθησης πριν τη διδασκαλία συναρτήσει της βαθμολογίας για τις τρεις διαφορετικές μεθόδους (Σχήμα 15.1). Από το διάγραμμα φαίνεται ότι υπάρχουν διαφορές ανάμεσα στις τρεις μεθόδους όταν ληφθεί υπόψη η ικανότητα μάθησης πριν την διδασκαλία, και αυτό είναι που θα εξεταστεί με την ανάλυση συνδιακύμασης. Αυτή εφαρμόζεται με τη εντολή που χρησιμοποιείται και για τη γραμμική παλινδρόμηση, δηλαδή την `lm()`. Για να υπολογιστεί ο πίνακας συνδιακύμανσης χρησιμοποιείται η εντολή `anova()` και όρισμα το αντικείμενο της γραμμικής παλινδρόμησης με μερικές ανεξάρτητες μεταβλητές παράγοντες και μερικές συνεχείς. Από τον πίνακα συνδιακύμανσης συμπεραίνεται ότι η ικανότητα μάθησης πρίν τη διδασκαλία (x) επηρεάζει τη βαθμολογία (Y), και ότι υπάρχουν διαφορές ανάμεσα στις τρεις όταν πάρουμε υπόψη την ικανότητα μάθησης.

```
> y <- c(6,4,5,3,4,3,6, 8,9,7,9,8,5,7, 6,7,7,7,8,5,7)
> x <- c(3,1,3,1,2,1,4, 4,5,5,4,3,1,2, 3,2,2,3,4,1,4)
```

```
> m <- gl(3,7)
> plot(x[m==1], y[m==1], pch="A", xlim=c(0,6), ylim=c(2,10),
+ xlab="Aptitude Scores", ylab="Achievement Scores")
> points(x[m==2], y[m==2], pch="B")
> points(x[m==3], y[m==3], pch="C")
> anova(z <- lm(y~x+m))
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value    Pr(>F)
x          1 36.575  36.575  60.355 5.428e-07 ***
m          2 16.932   8.466  13.970 0.0002579 ***
Residuals 17 10.302   0.606
---
Signif. codes:  0 ‘‘***’’ 0.001 ‘‘**’’ 0.01 ‘‘*’’ 0.05 ‘‘.’’ 0.1 ‘‘ ’’ 1
```



Σχήμα 15.1: Διάγραμμα διασπορών της ικανότητας μάθησης συναρτήσει της βαθμολογίας για τις 3 μεθόδους.

Κεφάλαιο 16

Εκτίμηση Μη-Γραμμικών Μοντέλων

16.1 Περιγραφή των Δεδομένων

Τα δεδομένα που θα χρησιμοποιηθούν στο κεφάλαιο αυτό λήφθηκαν από μια δοκιμή με δέκτη-ορμονών σχετικά με τον όγκο στο στήθος στους ανθρώπους. Σε μια τέτοια δοκιμή η συγκέντρωση στον δέκτη καθορίζεται τεχνικά από την έκθεση ενός συγκεκριμένου κυττάρου ή ιστού σε διάφορες συγκεντρώσεις ραδιενεργά ιχνηθετημένου συναρμοτή μέχρι να φτάσει ο δέκτης κοντά στον κορεσμό. Η συγκέντρωση του δεσμευμένου (B) και του ελεύθερου (F) συναρμοτή στην κατάσταση ισορροπίας μετρείται για κάθε επανάληψη. Τα δεδομένα δίνονται πιο κάτω.

```
> F<-c(84.6,83.9,148.2,147.8,463.9,463.8,964.1,967.6,1926.0,1900.0)
> B<-c(12.1,12.5,17.2,16.7,28.3,26.9,37.6,35.8,38.5,39.9)
> hormone.dat<-data.frame(F=F,B=B)
> hormone.dat
        F    B
1    84.6 12.1
2    83.9 12.5
3   148.2 17.2
4   147.8 16.7
5   463.9 28.3
6   463.8 26.9
7   964.1 37.6
```

```
8   967.6 35.8
9  1926.0 38.5
10 1900.0 39.9
```

16.2 Ανάλυση με Μη-Γραμμικό Μοντέλο

Η σχέση μεταξύ της συγκέντρωσης του φραγμένου και του ελεύθερου συναρμοτή στη δοκιμή του δέκτη ορμονλων περιγράφεται από την εξίσωση Michaelis-Menten

$$B_i = \frac{B_{max} F_i}{K_D + F_i} + \varepsilon_i$$

όπου ε_i είναι τυχαίο σφάλμα με μέση τιμή μηδέν, και B_{max} και K_D είναι οι παράμετροι, γνωστές ως ικανότητα και συνάφεια, οι οποίες θα προσδιοριστούν από n ζευγάρια παρατηρήσεων (F_i, B_i). Τα σφάλματα για τα διάφορα ζευγάρια παρατηρήσεων θεωρούνται ασυσχέτιστα, αλλά δεν είναι αναγκαίο να έχουν την ίδια διακύμανση.

Η εκτίμηση των δύο παραμέτρων στο πιο πάνω μη-γραμμικό μοντέλο μπορεί να γίνει με διάφορους τρόπους. Εδώ θα παρουσιαστεί μια απευθείας μέθοδος βασισμένη στην ελαχιστοποίηση της συνάρτησης αθροίσματος τετραγώνων, S, όπου η S δίνεται από:

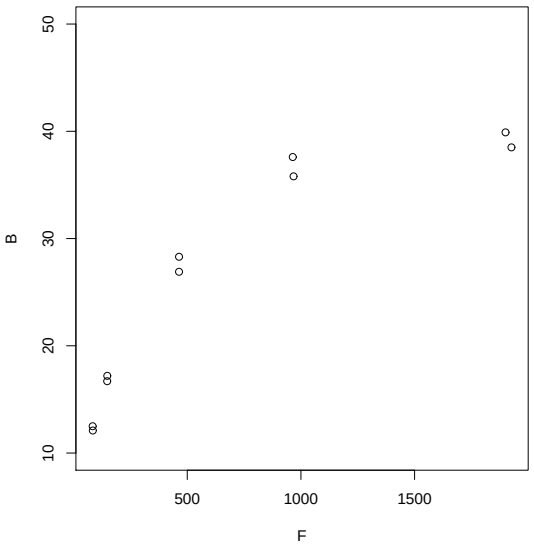
$$S = \sum_{i=1}^{n} \left(B_i - \frac{B_{max} F_i}{K_D + F_i} \right)^2 .$$

Στην R η μέθοδος αυτή εφαρμόζεται με τη βοήθεια της εντολής `nls`. Η διαδικασία της εκτίμησης απαιτεί τον ορισμό αρχικών τιμών για τις παραμέτρους B_{max} και K_D. Τέτοιες τιμές μπορούν να βρεθούν σχετικά εύκολα με τη βοήθεια του γραφήματος των δεδομένων που παρουσιάζεται στο Σχήμα 16.1. Η μεγαλύτερη τιμή του B είναι περίπου 40 και αυτή μπορεί να χρησιμοποιηθεί για αρχική τιμή του B_{max}. Η τιμή του K_D είναι η συγκέντρωση για την οποία η B είναι ίση με $B_{max}/2$. Από το γράφημα αυτή η τιμή είναι περίπου ίση με 250. Η εκτίμηση των παραμέτρων του μη-γραμμικού μοντέλου στην R παρουσιάζεται πιο κάτω:

```
> hormone.fit<-nls(B~Bmax*F/(KD+F),hormone.dat,start=list(Bmax=40,KD=250))
> summary(hormone.fit)

Formula: B ~ Bmax * F/(KD + F)

Parameters:
```

```
     Estimate Std. Error t value Pr(>|t|)
Bmax   44.378      1.129   39.31 1.93e-10 ***
KD    241.688     20.946   11.54 2.89e-06 ***
---
Signif. codes:  0 ‘‘***’’ 0.001 ‘‘**’’ 0.01 ‘‘*’’ 0.05 ‘‘.’’ 0.1 ‘‘ ’’ 1

Residual standard error: 1.288 on 8 degrees of freedom

Number of iterations to convergence: 4
Achieved convergence tolerance: 7.268e-07
```

Από τα αποτελέσματα παρατηρείται ότι το B_{max} εκτιμήθηκε να είναι ίσο με 44.378 με τυπικό σφάλμα 1.129, ενώ το K_D εκτιμήθηκε να είναι ίσο με 241.688 με τυπικό σφάλμα 20.946. Το t-test για τον έλεγχο υποθέσεων $B_{max} = 0$ και $K_D = 0$, δείχνει ότι και οι δύο εκτιμήσεις είναι διαφορετικές του μηδενός.



Σχήμα 16.1: Γράφημα των Δεδομένων.

Στις περισσότερες περιπτώσεις ανάλυσης με μη-γραμμικό μοντέλο είναι σημαντικό να συγκριθούν οι εκτιμήσεις εφαρμόζοντας τη μέθοδο με διάφορες αρχικές τιμές. Χρησιμοποιώντας για αρχικές τιμές $B_{max} = 30$ και $K_D = 300$ παίρνουμε

περίπου τα ίδια αποτελέσματα με πριν.

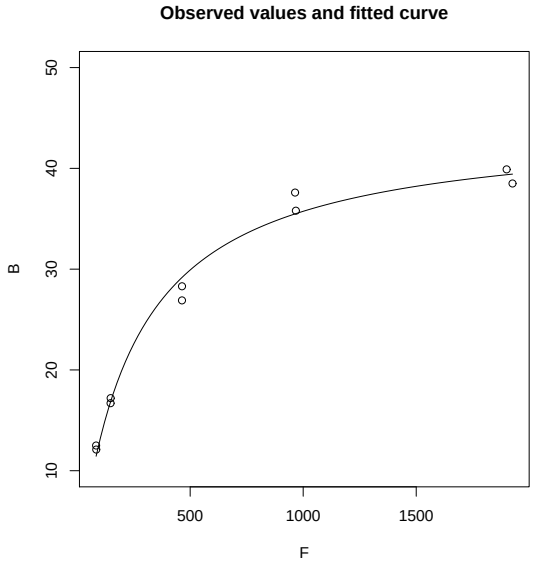
```
> hormone.fit2<-nls(B~Bmax*F/(KD+F),hormone.dat,start=list(Bmax=30,KD=300))
> summary(hormone.fit2)

Formula: B ~ Bmax * F/(KD + F)

Parameters:
     Estimate Std. Error t value Pr(>|t|)
Bmax   44.378      1.129   39.31 1.93e-10 ***
KD    241.687     20.946   11.54 2.89e-06 ***
---
Signif. codes:  0 ‘‘***’’ 0.001 ‘‘**’’ 0.01 ‘‘*’’ 0.05 ‘‘.’’ 0.1 ‘‘ ’’ 1

Residual standard error: 1.288 on 8 degrees of freedom

Number of iterations to convergence: 5
Achieved convergence tolerance: 7.389e-06
```

Για να εξετασθεί πόσο καλά εφαρμόζει η εξίσωση Michaelis-Menten στα δεδομένα, προσθέτουμε στο γράφημα των δεδομένων την εκτιμώμενη καμπύλη της εξίσωσης (Σχήμα 16.2). Οι εκτιμήσεις των παραμέτρων μπορούν να εξαχθούν χρησιμοποιώντας την εντολή `coef` και μετά να χρησιμοποιηθούν για να υπολογιστεί η εκτιμώμενη καμπύλη. Από το γράφημα είναι φανερό ότι η εκτιμώμενη καμπύλη εφαρμόζει πάρα πολύ καλά στα δεδομένα.

```
> Bmax.hat<-coef(hormone.fit)[1]
> KD.hat<-coef(hormone.fit)[2]
> F.seq<-seq(range(F)[1],range(F)[2],length=100)
> B.seq<-Bmax.hat*F.seq/(KD.hat+F.seq)

> plot(F,B,ylim=c(10,50))
> lines(F.seq,B.seq)
> title(main="Observed values and fitted curve")
```

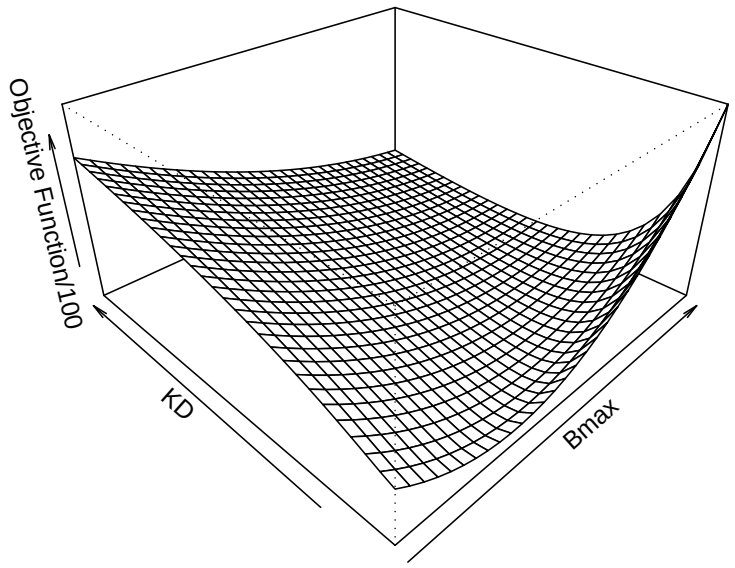
Στην περίπτωση του μη-γραμμικού μοντέλου με την πιο πάνω εξίσωση μόνο δύο παράμετροι έπρεπε να εκτιμηθούν. Συνεπώς, με τη βοήθεια της συνάρτησης `persp` μπορεί να κατασκευαστεί μια τρισδιάστατη προοπτική απεικόνιση της εξίσωσης, η οποία μπορεί να βοηθήσει στο να ελεγχθεί η ορθότητα των εκτιμήσεων

από το μοντέλο αλλά και η μοναδικότητά τους. Αυτό γίνεται στην R όπως πιο κάτω και το γράφημα παρουσιάζεται στο Σχήμα 16.3. Η επιφάνεια παρουσιάζεται επίπεδη σε διάφορα τμήματά της υποδεικνύοντας ότι και άλλα ζευγάρια τιμών των δύο παραμέτρων εκτός από τις εκτιμώμενες θα οδηγούσαν σε εξίσου καλή εφαρμογή των δεδομένων.



Σχήμα 16.2: Γράφημα των δεδομένων με την εκτιμώμενη καμπύλη από την εξίσωση Michaelis-Menten.

```
> Bmax.val<-seq(20,60,length=30)
> KD.val<-seq(50,1000,length=30)
> S<-matrix(0,30,30)
> for(i in 1:30)
+ {
+  for(j in 1:30)
+  {
+   S[i,j]<-sum((B-Bmax.val[i]*F/(KD.val[j]+F))^2)
+  }
+ }

> persp(Bmax.val,KD.val,S/100,xlab="Bmax",ylab="KD",zlab="Objective Function/100",
```



Σχήμα 16.3: Προοπτική απεικόνιση της επιφάνειας της εξίσωσης Michaelis-Menten.

Κεφάλαιο 17

Poisson Παλινδρόμηση και Λογαριθμικά Γραμμικά Μοντέλα

Πολλές φορές σε εφαρμογές παρατηρούνται δεδομένα συχνοτήτων, π.χ. ο αριθμός των περιπτώσεων στα κελιά ενός πίνακα συνάφειας, ο αριθμός τροχαίων αυτοκινητιστικών δυστυχημάτων, ο αριθμός πελατών στην τράπεζα κ.ο.κ. Η κατανομή Poisson χρησιμεύει πιο πολύ στην ανάλυση αυτών των δεδομένων και είναι γνωστό ότι δίνεται από τον τύπο,

$$P(Y = y) = \frac{\mu^y e^{-\mu}}{y!}.$$

Η παράμετρος μ, η μέση τιμή της Poisson, είναι σημαντική και συνήθως δίνεται σαν "ρυθμός", όπως π.χ. ο αριθμός των πελατών που αγοράζουν το προϊόν Α ανά 100 που πελάτες του ιδίου καταστήματος, ο αριθμός τροχαίων ανά 1000 άτομα, κ.ο.κ.

17.1 Poisson Παλινδρόμηση

Έστω $Y_1, Y_2, \ldots, Y_n$ ανεξάρτητες τ.μ. από την $Poisson(\mu_i)$. Υποθέτουμε ότι $E(Y_i) = \mu_i = n_i\theta_i$. Αν για παράδειγμα Y_i είναι ο αριθμός απαιτήσεων σε ασφαλιστική εταιρεία για ένα μοντέλο αυτοκινήτου Α, τότε, n_i είναι ο αριθμός των μοντέλων Α που έχουν ασφαλιστεί, και θ_i μπορεί να είναι η ηλικία, η χρήση, η περιοχή κ.ο.κ. Για την ανάλυση τέτοιου είδους δεδομένων συνήθως χρησιμοποιείται

το μοντέλο

$$\theta_i = e^{x_i^T \beta},$$

οπότε το αντίστοιχο γενικευμένο γραμμικό μοντέλο δίνεται από

$$E(Y_i) = \mu_i = n_i e^{x_i^T \beta}.$$

Για παράδειγμα αν $\mathbf{x_i = 0, 1}$, τότε

$$E(Y_i|X_i = 0) = n_i, E(Y_i|X_i = 0) = n_i e^{\beta}$$

και συνεπώς το ποσοστιαίο πηλίκο δίνεται από

$$RR = \frac{E(Y_i|X_i = 1)}{E(Y_i|X_i = 0)} = e^{\beta}$$

και δείχνει την αλλαγή στην αναμενόμενη τιμή. Η εκτίμηση της παραμέτρου $\boldsymbol{\beta}$ γίνεται μέσω της θεωρίας πιθανοφάνειας για γενικευμένα γραμμικά μοντέλα.

Αν $\hat{\beta}$ είναι η Ε.Μ.Π. τότε μπορούμε να ελέγξουμε τις υποθέσεις $H_0 : \beta = \beta_0$ με score test, Wald test και έλεγχο πηλίκου πιθανοφάνειας.

Επίσης,

$$\hat{Y}_i = \hat{\mu}_i = n_i e^{\mathbf{x_i}\hat{\boldsymbol{\beta}}}.$$

Τα υπόλοιπα Pearson δίνονται από

$$r_i = \frac{O_i - E_i}{\sqrt{E_i}},$$

με $O_i = Y_i$ και $E_i = \hat{Y}_i$. Τότε,

$$X^2 = \sum r_i^2 = \sum (\frac{O_i - E_i}{\sqrt{E_i}})^2.$$

Η συνάρτηση deviance δίνεται από

$$D = 2 \sum \left\{ O_i \log \left(\frac{O_i}{E_i} \right) - (O_i - E_i) \right\},$$

και τα υπόλοιπα deviance

$$d_i = sign(O_i - E_i) \sqrt{2 \left[O_i \log \left(\frac{O_i}{E_i} \right) - (O_i - E_i) \right]}$$

Οπότε, $D = \sum_{i=1}^{n} d_i^2$, και απορρίπτω το μοντέλο αν σε επίπεδο σημαντικότητας α, είτε το D είτε το X^2 είναι μεγαλύτερο του X^2_{N-p}.

17.2 Παράδειγμα

Τα παρακάτω δεδομένα αναφέρονται σε μία μελέτη όπου όλοι οι Βρετανοί γιατροί απάντησαν σε ένα ερωτηματολόγιο σχετικά με το αν καπνίζουν ή όχι. Ο παρακάτω πίνακας δείχνει τον αριθμό θανάτων από στεφανιαία νόσο μετά από 10 χρόνια. Παρουσιάζει επίσης και τον ολικό πληθυσμό.

	Smokers		Non-Smokers	
Age group	Deaths	Population	Deaths	Population
35 - 44	32	52407	2	18790
45 - 54	104	43248	12	10673
55 - 64	206	28612	28	5710
65 - 74	186	12663	28	2585
75 - 84	102	5317	31	1462

Στην R το πλαίσιο των δεδομένων κατασκευάζεται ως εξης :

```
> deaths <- c(32,2,104,12,206,28,186,28,102,31)
> population <- c(52407,18790,43248,10673,28612,5710,12663,2585,5317,1462)
> smoke <-gl(2,1,10,labels=c("Yes","No"))
> age <- gl(5,2,10,labels=c("35--44","45--54","55--64", "65--74", "75--84"))
> chddata=data.frame(deaths,population,smoke,age)
> chddata
   deaths population smoke    age
1      32      52407   Yes 35--44
2       2      18790    No 35--44
3     104      43248   Yes 45--54
4      12      10673    No 45--54
5     206      28612   Yes 55--64
6      28       5710    No 55--64
7     186      12663   Yes 65--74
8      28       2585    No 65--74
9     102       5317   Yes 75--84
10     31       1462    No 75--84
```

Θα εξεταστούν τρία ερωτήματα:

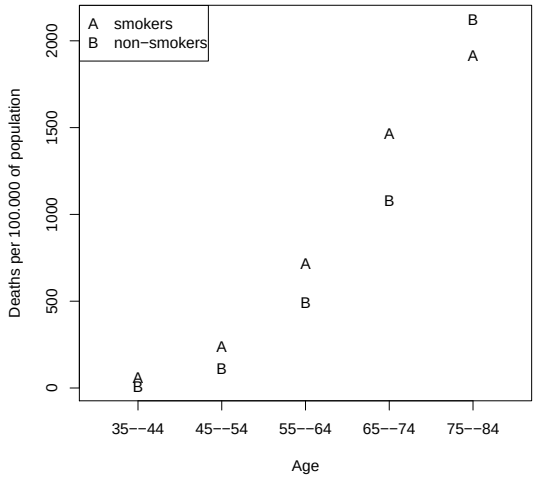
1. Είναι τα ποσοστά θανάτου πιο ψηλά στους καπνιστές;

2. Αν ναι, κατα πόσο;

3. Υπάρχει διαφοροποίηση λόγω ηλικίας;

Μια πρώτη περιγραφή του προβλήματος δίνεται μέσω του γραφήματος στο Σχήμα 17.1, το οποίο παρουσιάζει τους θανάτους ανά 100 χιλιάδες άτομα του πληθυσμού συναρτήσει της ηλικιακής ομάδας για τους καπνιστές (A) και μη καπνιστές (B), αντίστοιχα. Υπάρχει ένδειξη ότι,με εξαίρεση την ηλικιακή ομάδα 75 εως 84 χρονών, τα ποσοστά θανάτου στους καπνιστές είναι μεγαλύτερα από τα αντίστοιχα στους μη καπνιστές, αλλά και η διαφορά των ποσοστών αυξάνεται όσο μεγαλώνει η ηλικία των ατόμων. Το γράφημα αυτό κατασκευάζεται χρησιμοποιώντας τις εντολές :

```
> rate= deaths*100000/population
> plot(age[smoke=="No"], rate[smoke=="No"], xlab="Age",
+ ylab="Deaths per 100.000 of population",lty=0,ylab=c(0,2500))

> points(age[smoke=="Yes"], rate[smoke=="Yes"], pch="A")
> points(age[smoke=="No"], rate[smoke=="No"], pch="B")
> legend("topleft",c("smokers","non-smokers"),pch=c("A","B"))
```



Σχήμα 17.1: Θανάτοι ανά 100 χιλιάδες άτομα συναρτήσει της ηλικιακής ομάδας.

Το μοντέλο που θα χρησιμοποιηθεί για την ανάλυση είναι το ακόλουθο:

$$\log(deaths_i) = \log(population_i) + \beta_1 + \beta_2 smoke_i + \beta_3 agecat_i + \beta_4 agesq_i + \beta_5 smkage_i$$

όπου $i = 1, 2, \ldots, 5$ για τους καπνιστές και $i = 6, 7, \ldots, 10$ για τους μη καπνιστές. Επίσης,

$$smoke_i = \begin{cases} 1 & \text{για} \quad \text{ΝΑΙ} \\ 0 & \text{για} \quad \text{ΟΧΙ} \end{cases}, \quad agecat_i = \begin{cases} 1 & \text{για} \quad \text{35-44} \\ 2 & \text{για} \quad \text{45-54} \\ 3 & \text{για} \quad \text{55-64} \\ 4 & \text{για} \quad \text{65-74} \\ 5 & \text{για} \quad \text{75-84}, \end{cases},$$

και

$$agesq_i = (agecat_i)^2, \quad smkage_i = \begin{cases} agecat_i & \text{για} \quad \text{καπνιστές} \\ 0 & \text{για} \quad \text{μη καπνιστές} \end{cases},$$

Για την εφαρμογή της ανάλυσης στην R χρησιμοποιήθηκαν οι ακόλουθες εντολές :

```
> age <- as.numeric(age)
> age
 [1] 1 1 2 2 3 3 4 4 5 5
> smoke <- ifelse(smoke=="Yes",1,0)
> smoke
 [1] 1 0 1 0 1 0 1 0 1 0
> agesq <- age^{2}
> agesq
 [1]  1  1  4  4  9  9 16 16 25 25
> agesm <-ifelse(smoke==0, age, 0)
> agesm
 [1] 0 1 0 2 0 3 0 4 0 5
> populationl <- log(population)

> fit1 <- glm(deaths~offset(populationl)+smoke+age+agesq+agesm, family=poisson)
> summary(fit1)

Call:
glm(formula = deaths ~ offset(populationl) + smoke + age + agesq +
    agesm, family = poisson)

Deviance Residuals:
       1         2         3         4         5         6         7         8
```

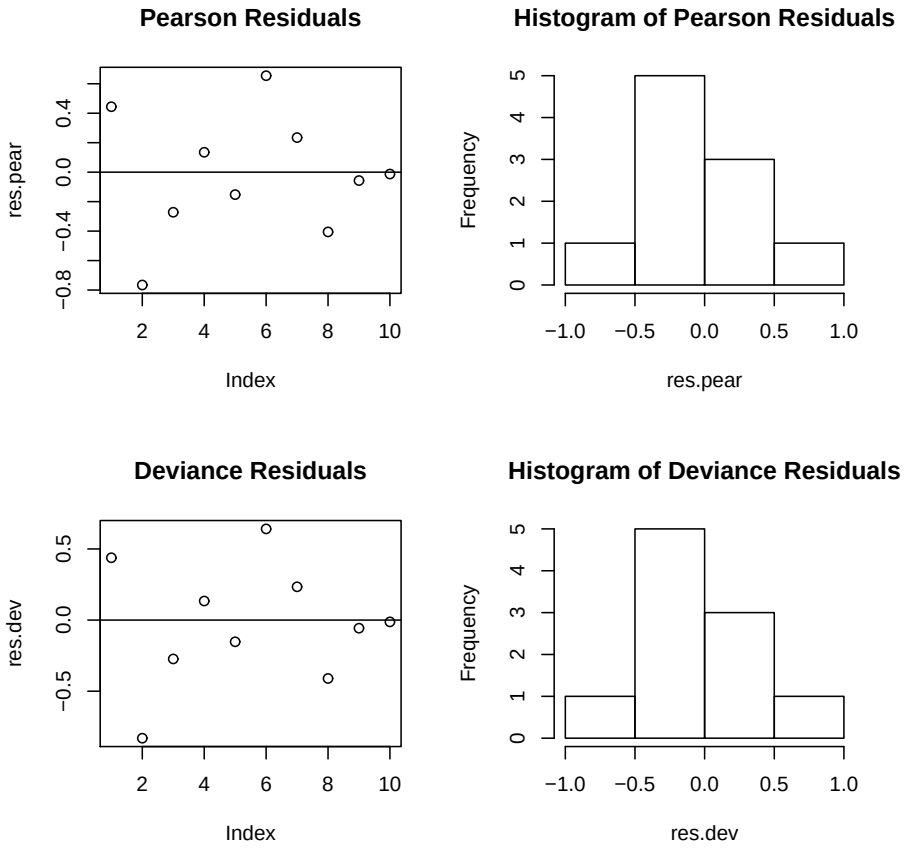
```
 0.43820  -0.83049  -0.27329   0.13404  -0.15265   0.64107   0.23393  -0.41058
       9        10
-0.05700  -0.01275

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -10.79176    0.45008 -23.978  < 2e-16 ***
smoke         1.44097    0.37220   3.872 0.000108 ***
age           2.06893    0.18170  11.386  < 2e-16 ***
agesq        -0.19768    0.02737  -7.223 5.08e-13 ***
agesm         0.30755    0.09704   3.169 0.001528 **
---
Signif. codes:  0 ‘‘***’’ 0.001 ‘‘**’’ 0.01 ‘‘*’’ 0.05 ‘‘.’’ 0.1 ‘‘ ’’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 935.0673  on 9  degrees of freedom
Residual deviance:   1.6354  on 5  degrees of freedom
AIC: 66.703

Number of Fisher Scoring iterations: 4
> rate.ratio <- exp(fit1$coef[-1])
> rate.ratio
    smoke       age     agesq     agesm
4.2247998 7.9163500 0.8206353 1.3600862
```

Από τον πίνακα συντελεστών συμπεραίνεται ότι και οι 4 επεξηγηματικές μεταβλητές είναι σημαντικές για το μοντέλο. Συνεπώς, η πιθανότητα θανάτου επηρεάζεται από το αν κάποιος είναι καπνιστής αλλά και από την ηλικία του. Το `rate.ratio` που υπολογίστηκε στο τέλος, δείχνει ότι για αυτούς που καπνίζουν, το ρίσκο θανάτου είναι 4 φορές μεγαλύτερο.

Τέλος, θα εξεταστεί η καταλληλότητα του μοντέλου με τη βοήθεια της ανάλυσης των υπολοίπων. Για το λόγο αυτό υπολογίζονται τα υπόλοιπα `pearson` και `deviance`, όπως και οι εκτιμώμενες τιμές της εξαρτημένης μεταβλητής, η οποία στο πιο πάνω παράδειγμα είναι ο αριθμός των θανάτων. Επίσης, κατασκευάζεται ένας πίνακας ο οποίος παρουσιάζει τα αποτελέσματα αυτά για κάθε συνδυασμό των παραγόντων `age` και `smoke`. Στη συνέχεια δίνεται ο έλεγχος καλής προσαρ-

μογής και για τα δύο είδη υπολοίπων, υποδεικνύοντας την καταλληλότητα του μοντέλου. Από το γράφημα και το ιστόγραμμα των υπολοίπων είναι φανερή η τυχαιότητα και η κανονικότητά τους. Φυσικά δεν μπορούν να εξαχθούν ακριβή συμπεράσματα λόγω του μικρού αριθμού των υπολοίπων.

```
> res.pear <- residuals(fit1, type="pearson")
> res.dev  <- residuals(fit1, type="deviance")
> predict.fit <- predict(fit1, type="response")
> cbind(age, smoke, deaths, predict.fit, res.pear, res.dev)
   age smoke deaths predict.fit    res.pear     res.dev
1    1     1     32   29.584734  0.44404929  0.43820403
2    1     2      2    3.414801 -0.76561908 -0.83049031
3    2     1    104  106.811960 -0.27208163 -0.27328873
4    2     2     12   11.541629  0.13492231  0.13404370
5    3     1    206  208.198646 -0.15237591 -0.15264528
6    3     2     28   24.743377  0.65469354  0.64106682
7    4     1    186  182.827893  0.23459923  0.23392570
8    4     2     28   30.229155 -0.40544060 -0.41058325
9    5     1    102  102.576767 -0.05694769 -0.05700118
10   5     2     31   31.071038 -0.01274427 -0.01274913

> devian.fit <- sum(res.dev^{2})
> 1-pchisq(devian.fit, df=10-5)
[1] 0.8969393
> pear.fit <- sum(res.pear^{2})
> 1-pchisq(pear.fit, df=10-5)
[1] 0.907199

> par(mfrow=c(2,2))
> plot(res.pear,main="Pearson Residuals")
> abline(h=0)
> hist(res.pear,main="Histogram of Pearson Residuals")
> plot(res.dev,main="Deviance Residuals")
> abline(h=0)
> hist(res.dev,main="Histogram of Deviance Residuals")
```



Σχήμα 17.2: Γραφήματα υπολοίπων.

Κεφάλαιο 18

Μη Παραμετρική Παλινδρόμηση

Το παραδοσιακό παραμετρικό μοντέλο δίνεται από την εξίσωση

$$y_i = f(\boldsymbol{\beta}, \mathbf{x_i'}) + \varepsilon_i,$$

όπου $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$ το διάνυσμα των παραμέτρων που θα εκτιμηθούν, και $\mathbf{x_i'} = (x_1, \ldots, x_k)$ το διάνυσμα των επεξηγηματικών μεταβλητών για την i παρατήρηση. Για τα σφάλματα ε_i υποθέτουμε ότι είναι ανεξάρτητα και ακολουθούν την κανονική κατανομή με μέση τιμή 0 και σταθερή διακύμανση σ^2. Η συνάρτηση $f(.)$ ορίζει την σχέση μεταξύ της μέσης τιμής της εξαρτημένης μεταβλητής και των επεξηγηματικών μεταβλητών. Το γενικό μη παραμετρικό μοντέλο παλινδρόμησης γράφεται με παρόμοιο τρόπο χωρίς όμως να ορίζεται η f:

$$y_i = f(\mathbf{x_i'}) + \varepsilon_i = f(x_{i1}, \ldots, x_{ik}) + \varepsilon_i.$$

Επιπρόσθετα, ο σκοπός της μη παραμετρικής παλινδρόμησης είναι να εκτιμήσει την συνάρτηση παλινδρόμησης f απ' ευθείας παρά να εκτιμήσει παραμέτρους. Οι περισσότερες μέθοδοι μη παραμετρικής παλινδρόμησης υποθέτουν ότι η f είναι ομαλή και συνεχής.

Ειδική περίπτωση του γενικού μοντέλου είναι η μη παραμετρική απλή παλινδρόμηση, στην οποία υπάρχει μόνο μία επεξηγηματική μεταβλητή:

$$y_i = f(x_i) + \varepsilon_i$$

Το μοντέλο αυτό ονομάζεται επίσης και «εξομάλυνση διαγράμματος διασποράς»

αφού κατασκευάζει μια ομαλή καμπύλη για το διάγραμμα διασποράς του y συναρτήσει του x.

Λόγω της δυσκολία εφαρμογής και απεικόνισης ενός γενικού μη παραμετρικού μοντέλου παλινδρόμησης με πολλές επεξηγηματικές μεταβλητές, έχουν αναπτυχθεί πιο περιοριστικά μοντέλα, όπως π.χ. το αθροιστικό μοντέλο παλινδρόμησης (additive regression model)

$$y_i = \alpha + f_1(x_{i1}) + f_2(x_{i2}) + \ldots + f_k(x_{ik}) + \varepsilon_i.$$

Για αυτό το μοντέλο υποθέτουμε ότι οι μερικές συναρτήσεις παλινδρόμησης $f_j(.)$ είναι ομαλές και μπορούν να εκτιμηθούν από τα δεδομένα. Το μοντέλο αυτό είναι πιο περιοριστικό από το γενικό μη παραμετρικό μοντέλο, αλλά λιγότερο περιοριστικό από το μοντέλο γραμμικής παλινδρόμησης, στο οποίο όλες οι μερικές συναρτήσεις παλινδρόμησης θεωρούνται γραμμικές.

18.1 Τοπική Πολυωνυμική Παλινδρόμηση

Απλή Παλινδρόμηση

Το μοντέλο απλής παλινδρόμησης το οποίο θεωρείται δίνεται από

$$y_i = f(x_i) + \varepsilon_i,$$

όπου $f(.)$ η άγνωστη παράμετρος. Η συνάρτηση παλινδρόμησης f θα εκτιμηθεί σε ένα συγκεκριμένο σημείο x_0. Αυτό είναι δυνατόν χρησιμοποιώντας τη πολυωνυμική παλινδρόμηση p-τάξης της y πάνω στη x των τοπικά σταθμισμένων ελαχίστων τετραγώνων (weighted least squares),

$$y_i = \alpha + b_1(x_i - x_0) + b_2(x_i - x_0)^2 + \ldots + +b_p(x_i - x_0)^p + e_i$$

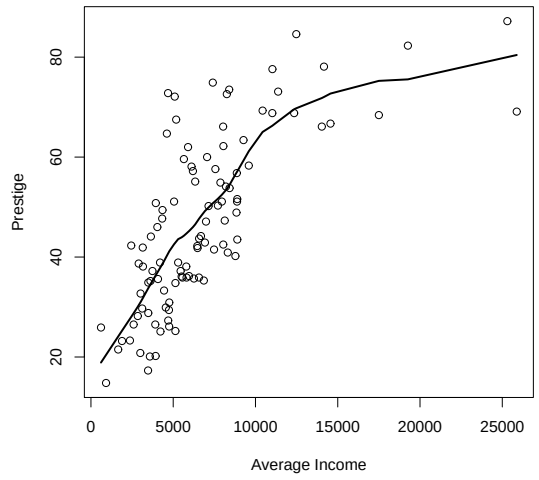
με την οποία οι παρατηρήσεις σταθμίζονται σε σχέση με το πόσο κοντά είναι στο σημείο x_0. Η εκτίμηση γίνεται όχι μόνο στο σημείο x_0, αλλά και σε όλες τις n παρατηρήσεις, x_i. Μία γνωστή συνάρτηση στάθμισης, η οποία χρησιμοποιείται συχνά, είναι η τρικυβική συνάρτηση:

$$W(z) = \begin{cases} (1 - |z|^3)^3 & \text{για} \quad |z| < 1 \\ 0 & \text{για} \quad |z| \geq 1 \end{cases}.$$

Στο παρακάτω παράδειγμα χρησιμοποιείται το πλαίσιο δεδομένων `Prestige`, το οποίο βρίσκεται στη βιβλιοθήκη `car` και δίνει το βαθμό γοήτρου για τα διάφορα

επαγγέλματα στον Καναδά. Θα εξεταστεί η σχέση του βαθμού γοήτρου (prestige) με το εισόδημα (income). Για τη εφαρμογή της πολυωνυμικής παλινδρόμησης των τοπικά σταθμισμένων ελαχίστων τετραγώνων στην R χρησιμοποιείται η συνάρτηση lowess (local weighted scatter plot smoothing). Το όρισμα f δίνει το περίβλημα του εξομαλυντή, δηλαδή την αναλογία των σημείων στο γράφημα τα οποία επηρεάζουν την εξομάλυνση σε κάθε σημείο, ενώ το όρισμα iter δίνει τον αριθμό των επαναλήψεων που θα εκτελεστούν για τη διαδικασία εκτίμησης με σκοπό τη μείωση της βαρύτητας στα τελικά αποτελέσματα των απομακρυσμένων παρατηρήσεων. Το διάγραμμα διασπορών των δεδομένων μαζί με τη γραμμή εξομάλυνσης lowess παρουσιάζεται στο Σχήμα 18.1.

```
> library(car)
> attach(Prestige)
> plot(income,prestige,xlab="Average Income",ylab="Prestige")
> lines(lowess(income,prestige,f=0.5,iter=0),lwd=2)
```

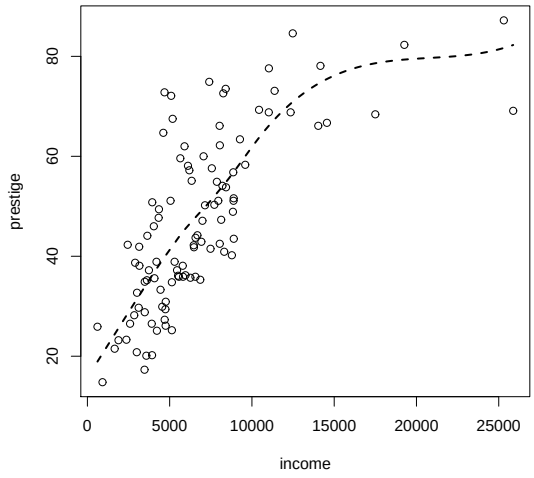


Σχήμα 18.1: Διάγραμμα διασποράς των δεδομένων μαζί με τη γραμμή εξομάλυνσης lowess.

Στην R υπάρχει επίσης η συνάρτηση loess η οποία χρησιμοποιείται για την πιο πάνω μέθοδο, η οποία έχει και περισσότερες δυνατότητες. Ακολουθεί ένα παράδειγμα για το πως χρησιμοποιείται για τα πιο πάνω δεδομένα. Ορίζοντας

degree=1 εφαρμόζεται η τοπικά γραμμική παλινδρόμηση. Το Σχήμα 18.2 δίνει το διάγραμμα διασπορών των δεδομένων με τη γραμμή εξομάλυνσης loess.

```
> mod.lo.inc<-loess(prestige~income,span=0.7,degree=1)
> mod.lo.inc
Call: loess(formula = prestige ~ income, span = 0.7, degree = 1)

Number of Observations: 102 Equivalent Number of Parameters: 3.85
Residual Standard Error: 11.13
> inc.100<-seq(min(income),max(income),len=100)
> pres<-predict(mod.lo.inc,data.frame(income=inc.100))
> plot(income,prestige)
> lines(inc.100,pres,lty=2,lwd=2)
```



Σχήμα 18.2: Διάγραμμα διασποράς των δεδομένων μαζί με τη γραμμή εξομάλυνσης lowess.

18.2 Εξομαλυντές Splines

Οι εξομαλυντές Splines είναι η λύση του προβλήματος απλής παλινδρόμησης, το οποίο επιζητά την εύρεση των συναρτήσεων $\hat{f}(x)$ με δύο συνεχείς παραγώγους, οι

οποίες ελαχιστοποιούν το άθροισμα τετραγώνων ποινής (penalized sum of squares),

$$SS^*(h) = \sum_{i=1}^{n}[y_i - f(x_i)]^2 + h\int_{x_{min}}^{x_{max}}[f''(x)]^2 dx,$$

όπου h είναι η παράμετρος της εξομάλυνσης, η οποία θεωρείται ανάλογη του πλάτους της γειτονιάς των τοπικά πολυωνυμικών εκτιμητών. Ο πρώτος όρος της εξίσωσης πιο πάνω είναι το άθροισμα τετραγώνων των υπολοίπων, ενώ ο δεύτερος όρος είναι η ποινή τραχύτητας (roughness penalty). Η ποινή αυτή είναι μεγάλη όταν η ολοκληρωτική δεύτερη παράγωγος της συνάρτησης παλινδρόμησης $f''(x)$ είναι μεγάλη, δηλαδή όταν η $f(x)$ αλλάζει γρήγορα κλίση. Όταν η σταθερά εξομάλυνσης h είναι ίση με 0, τότε η $\hat{f}(x)$ απλά παρεμβάλει τα δεδομένα. Αυτό είναι παρόμοιο με την εκτίμηση με τοπική παλινδρόμηση με περίβλημα ίσο με $1/n$. Αν το h όμως είναι αρκετά μεγάλο, τότε η $\hat{f}(x)$ θα επιλεχθεί έτσι ώστε η $\hat{f}''(x)$ είναι παντού 0, η οποίο ουσιαστικά είναι ισοδύναμη με μια γενική γραμμική εφαρμογή ελαχίστων τετραγώνων στα δεδομένα.

Το Σχήμα 18.3 παρουσιάζει στο ίδιο γράφημα την εξομάλυνση τοπικής πολυωνυμικής παλινδρόμησης `loess`, και την εξομάλυνση splines, η οποία γίνεται με την εντολή `smooth.spline` που βρίσκεται στη βιβλιοθήκη `splines`. Το γράφημα αυτό κατασκευάζεται όπως το γράφημα στο Σχήμα 18.2 προστέθοντας στο τέλος την εξομάλυνση splines με τη βοήθεια της εντολής `lines` όπως πιο κάτω:

```
> library(splines)
> lines(smooth.spline(income,prestige,df=3.85),lwd=2)
> legend("bottomright",c("loess","smoothing splines"),lty=c(2,1),lwd=c(2,2))
```

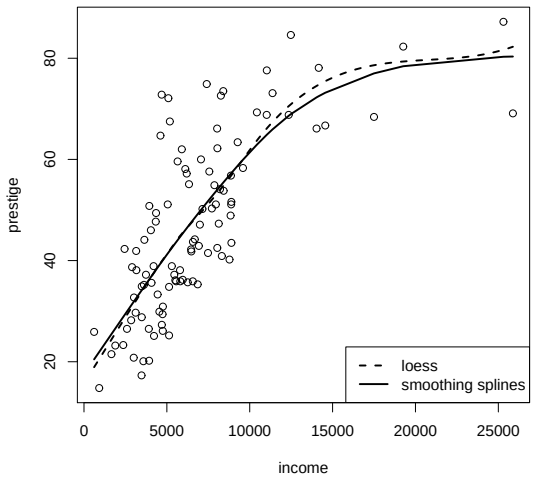
Οι βαθμοί ελευθερίας `df` τέθηκαν ίσοι με 3.85 για να συμφωνούν με τους βαθμούς ελευθερίας της εξομάλυνσης τοπικής πολυωνυμικής παλινδρόμησης.

18.3 Αθροιστική Απαραμετρική Παλινδρόμηση

Το μοντέλο της αθροιστικής μη παραμετρικής παλινδρόμησης δίνεται από

$$y_i = \alpha + f_1(x_{i1}) + f_2(x_{i2}) + \ldots + +f_k(x_{ik}) + \varepsilon_i,$$

όπου οι μερικές συναρτήσεις παλινδρόμησης f_j εφαρμόζονται χρησιμοποιώντας ένα εξομαλυντή απλής παλινδρόμησης, όπως για παράδειγμα η τοπικά πολυωνυμική παλινδρόμηση ή ο εξομαλυντής splines. Για την εφαρμογή της μεθόδου για την παλινδρόμηση του βαθμού γοήτρου συναρτήσει του εισοδήματος και της



Σχήμα 18.3: Διάγραμμα διασποράς των δεδομένων μαζί με τη γραμμή εξομάλυνσης lowess και smoothing splines.

εκπαίδευσης, χρησιμοποιείται η εντολή gam που βρίσκεται στη βιβλιοθήκη mgcv, όπως πιο κάτω:

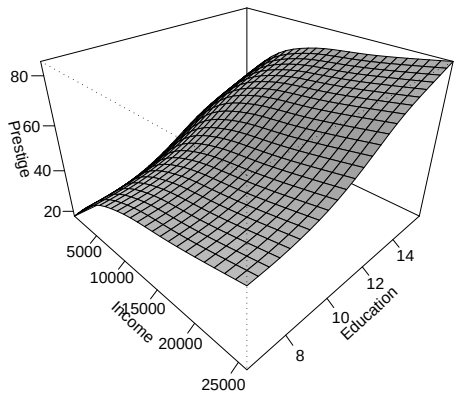
```
> library(mgcv)
> mod.gam<-gam(prestige~s(income)+s(education))
> mod.gam

Family: gaussian Link function: identity

Formula: prestige ~ s(income) + s(education)

Estimated degrees of freedom:
 3.117833 3.177297   total =  7.29513

GCV score:  52.1428
```

Η συνάρτηση s στην εντολή gam υποδεικνύει ότι κάθε όρος θα αναλυθεί χρησιμοποιώντας εξομαλυντή spline. Οι βαθμοί ελευθερίας υπολογίζονται με γενικευμένη σταυρωτή επαλήθευση. Σε αυτήν την περίπτωση, 3.1178 παράμετροι έχουν

χρησιμοποιηθεί για τον όρο `income`, και 3.1773 για τον όρο `education`. Οι βαθμοί ελευθερίας του μοντέλου είναι ίσοι με το άθροισμα τους συν 1, τη σταθερά της παλινδρόμησης.

Η επιφάνεια της αθροιστικής παλινδρόμησης δίνεται στο Σχήμα 18.4 και κατασκευάζεται όπως πιο κάτω:

```
> inc<-seq(min(income),max(income),len=25)
> ed<-seq(min(education),max(education),len=25)
> newdata<-expand.grid(income=inc,education=ed)
> fit.prestige<-matrix(predict(mod.gam,newdata),25,25)
> persp(inc,ed,fit.prestige,theta=45,phi=30,ticktype="detailed",
+ xlab="Income",ylab="Education",zlab="Prestige",expand=2/3,
+ shade=0.5)
```

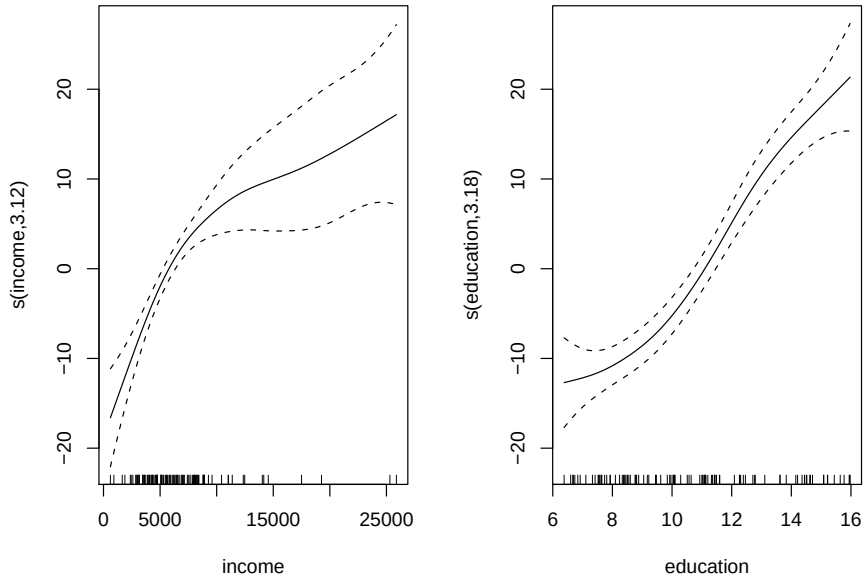


Σχήμα 18.4: Εκτιμώμενη επιφάνεια της αθροιστική απαραμετρικής παλινδρόμηση.

Για το λόγο ότι κομμάτια της επιφάνειας στην κατεύθυνση της μίας επεξηγηματικής μεταβλητής είναι παράλληλα, είναι αρκετό να γίνει το γράφημα κάθε μερικής παλινδρόμησης ξεχωριστά. Αυτή είναι και μια χρησιμότητα του μοντέλου αθροιστικής παλινδρόμησης, μειώνει το πολυδιάστατο πρόβλημα παλινδρόμησης σε μια σειρά από διδιάστατα γραφήματα των μερικών παλινδρομήσεων. Η σειρά

των γραφημάτων αυτών κατασκευάζεται στην R χρησιμοποιώντας το αντικείμενο gam ως όρισμα στην εντολή plot (Σχήμα 18.5).

```
> par(mfrow=c(1,2))
> plot(mod.gam)
```



Σχήμα 18.5: Γραφήματα των μερικών παλινδρομήσεων του μοντέλου αθροιστικής απαραμετρικής παλινδρόμηση.

Κεφάλαιο 19

Ανάλυση Επιβίωσης

Η ανάλυση επιβίωσης (survival analysis) αναφέρεται στην ανάλυση δεδομένων που αφορούν στο χρόνο που μεσολαβεί μέχρι κάποιο συγκεκριμένο συμβάν. Αρχικά η ανάλυση αναφερόταν στο χρόνο μεταξύ της θεραπείας μέχρι τον θάνατο και για αυτό το λόγο πήρε και το συγκεκριμένο όνομα. Η ανάλυση επιβίωσης όμως μπορεί να εφαρμοστεί σε αρκετές περιπτώσεις, όπως για παράδειγμα στη μηχανολογία, για την ανάλυση του χρόνου μέχρι την εμπλοκή ενός μηχανήματος ή τη γεωργία, για την ανάλυση του χρόνου μέχρι την στιγμή να βγάλει καρπό ένα δέντρο. Στην περίπτωση της μηχανολογίας η ανάλυση αναφέρεται και ως θεωρία αξιοπιστίας (reliability theory). Ο χρόνος επιβίωσης χρίζει ειδικής μεταχείρισης για το λόγο ότι είναι περιορισμένος στο να είναι πάντα θετικός, και γιατί τα δεδομένα περιέχουν λογοκριμένες (cencored) παρατηρήσεις. Τα λογοκριμένα δεδομένα είναι αυτά για τα οποία δεν είναι γνωστός ο χρόνος που συμβαίνει το γεγονός. Το μόνο που μπορεί να λεχθεί είναι ότι ο χρόνος επιβίωσής τους είναι μεγαλύτερος από την τιμή που έχει καταγραφεί.

Στην ανάλυση επιβίωσης είναι πολύ σημαντικές δύο συναρτήσεις οι οποίες περιγράφουν την κατανομή του χρόνου επιβίωσης: η συνάρτηση επιβίωσης και η συνάρτηση κινδύνου.

19.1 Συνάρτηση Επιβίωσης

Συμβολίζοντας το χρόνο επιβίωσης με T, η συνάρτηση επιβίωσης (survival function) $S(t)$ ορίζεται ως η πιθανότητα επιβίωσης ενός ατόμου πέραν τη χρονική

στιγμή t και δίνεται από τη σχέση:

$$S(t) = P(T > t) = 1 - F(t)$$

Η συνάρτηση επιβίωσης είναι μη αρνητική και μη αύξουσα συνάρτηση του t με $S(0) = 1$ και $S(\infty) = 0$. Η γραφική παράσταση της $S(t)$ συναρτήσει του t είναι γνωστή ως καμπύλη επιβίωσης και είναι πολύ σημαντική στην ανάλυση δεδομένων χρόνου επιβίωσης.

19.2 Συνάρτηση Κινδύνου

Η συνάρτηση κινδύνου, $h(t)$, ορίζεται ως η πιθανότητα αποβίωσης (ή πραγμάτωσης του γεγονότος που εξετάζεται) τη χρονική στιγμή t, δεδομένου ότι το άτομο έχει επιβιώσει μέχρι τη χρονική στιγμή t. Δηλαδή,

$$h(t) = \lim_{s \to 0} \frac{P(t \leq T \leq t + s \mid T \geq t)}{s}$$

Η συνάρτηση κινδύνου δίνει ένα μέτρο του πόσο πιθανό είναι ένα άτομο να αποβιώσει ως συνάρτηση της ηλικίας του ατόμου, για παράδειγμα ο κίνδυνος θανάτου ανάμεσα σε αυτούς που είναι ζωντανοί τη συγκεκριμένη στιγμή.

19.3 Μοντέλο αναλόγων συναρτήσεων κινδύνου

Στην ανάλυση επιβίωση παίζει μεγάλο ρόλο η εξεύρεση παραγόντων οι οποίοι να σχετίζονται με το χρόνο επιβίωσης. Αυτοί οι παράγοντες θα πρέπει να συμπεριληφθούν στο μοντέλο που θα χρησιμοποιηθεί για τη σχετική ανάλυση των δεδομένων. Αφού η συνάρτηση κινδύνου είναι μη αρνητική, ο λογάριθμός της μπορεί να εκφραστεί ως γραμμική συνάρτηση επεξηγηματικών μεταβλητών:

$$\ln h(t) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

Το μοντέλο αυτό όμως είναι πολύ περιοριστικό αφού υποθέτει ότι η συνάρτηση κινδύνου δεν εξαρτάται από το χρόνο. Υπάρχουν διάφορες μέθοδοι με τις οποίες το μοντέλο θα μπορούσε να υιοθετήσει την εξάρτηση του χρόνου, με την πιο γνωστή να είναι το μοντέλο αναλόγων συναρτήσεων κινδύνου (Cox 1972). Το μοντελό αυτό δίνεται από

$$\ln h(t) = \alpha(t) + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

όπου $\alpha(t)$ είναι οποιαδήποτε συνάρτηση του χρόνου. Ο όρος ¨αναλόγων συναρτήσεων κινδύνου¨ είναι λόγω του γεγονότος ότι για οποιαδήποτε άτομα για οποιδήποτε σημείο του χρόνου, ο λόγος των συναρτήσεων κινδύνου είναι σταθερός. Εξαιτίας του ότι η συνάρτηση κινδύνου $\alpha(t)$ δεν είναι ανάγκη να οριστεί εξ ολοκλήρου, το μοντέλο αναλόγων συναρτήσεων κινδύνου θεωρείται ως ημιπαραμετρικό.

Ο Cox εισηγήθηκε μια μέθοδο δεσμευμένης πιθανοφάνειας για εκτίμηση των παραμέτρων. Το σημαντικό στοιχείο αυτής της μεθόδου είναι οι εκτιμήσεις εξαρτώνται από τη σειρά με την οποία συμβαίνει το γεγονός και όχι τον ακριβή χρόνο.

19.4 Παράδειγμα

Τα δεδομένα του παραδείγματος, τα οποία δίνονται στο παράρτημα στο τέλος του κεφαλαίου, αναφέρονται σε 51 ασθενείς οι οποίοι πάσχουν από οξεία μυελοβλαστική λευχαιμία και που δεν έχουν μέχρι τώρα δεχθεί οποιαδήποτε θεραπεία. Οι ασθενείς αυτοί υποβάλλονται σε μια σειρά θεραπείας, στο τέλος της οποίας έχουν εξετασθεί αν έχουν ανταποκριθεί ή όχι. Έχουν καταγραφεί πριν τη θεραπεία έξι μεταβλητές :

1. η ηλικία διάγνωσης, `Age`,
2. το ποσοστό επίστρωσης των βλαστοκυττάρων, `Smear`,
3. το ποσοστό των κυττάρων από τη λευχαιμία που εισήλθαν στο μυελό των οστών, `Infil`,
4. το ποσοστό των κυττάρων που προήλθαν από το μυελό των οστών, `Index`,
5. τα απόλυτα βλαστοκύτταρα, `Blasts`, και
6. η ψηλότερη θερμοκρασία σώματος πριν τη θεραπεία, `Temp`.

Επίσης, καταγράφεται ο χρόνος επιβίωσης του ατόμου, `Time`, και η ανταπόκρισή του στη θεραπεία, `Resp`. Τέλος, η μεταβλητή `Status` δείχνει αν οι παρατηρήσεις ενός ατόμου είναι λογοκριμένες ή όχι. Η προς εξέταση ερώτηση είναι το κατά πόσο σημαντικές είναι στη πρόβλεψη του χρόνου επιβίωσης οι έξι μεταβλητές που καταγράφηκαν πριν από τη θεραπεία. Για να απαντηθεί η ερώτηση αυτή θα χρησιμοποιηθεί το μοντέλο αναλόγων συναρτήσεων κινδύνου.

Στην R οι αναγκαίες συναρτήσεις για την ανάλυση επιβίωσης βρίσκονται στη βιβλιοθήκη `survival`. Το μοντέλο αναλόγων συναρτήσεων κινδύνου εφαρμόζεται χρησιμοποιώντας την εντολή `coxph`. Η περιγραφή του μοντέλου στην `coxph`

γίνεται με παρόμοιο τρόπος όπως και στην περίπτωση των γραμμικών μοντέλων με την lm, με τη διαφορά ότι το αριστερό μέρος της είναι αντικείμενο επιβίωσης που δημιουργείται από τη συνάρτηση Surv. Στην περίπτωση που τα δεδομένα είναι δεξιά-λογοκριμένα, η συνάρτηση Surv έχει τη μορφή Surv(time,event), με time να είναι είτε ο χρόνος μέχρι το γεγονός ή ο χρόνος λογοκρισίας, και event να είναι μια δείκτρια μεταβλητή με τιμή ίση με 1 αν το γεγονός παρατηρείται ή ίση με 0 αν αν η παρατήρηση είναι λογοκριμένη. Εφαρμόζοντας το μοντέλο αναλόγων συναρτήσεων κινδύνου με τις 6 επεξηγηματικές μεταβλητές παίρνονται τα ακόλουθα αποτελέσματα:

```
> library(survival)
> cancer.dat <- read.table("cancer.dat", col.names=c("Age","Smear",
+ "Infil","Index","Blasts","Temp","Resp","Time","Status"))
> time<-cancer.dat[,"Time"]
> status<-1-cancer.dat[,"Status"]
> attach(cancer.dat)
> cancer.cox<-coxph(Surv(time,status)~Age+Smear+Infil+Index+Blasts)
> summary(cancer.cox)
Call:
coxph(formula = Surv(time, status) ~ Age + Smear + Infil + Index +
    Blasts)

  n= 51
          coef exp(coef) se(coef)      z       p
Age     0.03536     1.036  0.01018  3.473 0.00052
Smear   0.00915     1.009  0.01451  0.631 0.53000
Infil  -0.01835     0.982  0.01247 -1.472 0.14000
Index  -0.08955     0.914  0.04482 -1.998 0.04600
Blasts  0.00285     1.003  0.00973  0.293 0.77000

       exp(coef) exp(-coef) lower .95 upper .95
Age        1.036      0.965     1.016     1.057
Smear      1.009      0.991     0.981     1.038
Infil      0.982      1.019     0.958     1.006
Index      0.914      1.094     0.837     0.998
Blasts     1.003      0.997     0.984     1.022
```

```
Rsquare= 0.312   (max possible= 0.996 )
Likelihood ratio test= 19.1  on 5 df,   p=0.00188
Wald test            = 17.6  on 5 df,   p=0.00351
Score (logrank) test = 18.9  on 5 df,   p=0.00197
```

Είναι φανερό ότι η ηλικία διάγνωσης, Age, είναι η πιο σημαντική μεταβλητή για την πρόβλεψη του χρόνου επιβίωσης, αφού έχει p-value πολύ κοντά στο 0. Για το λόγο αυτό εφαρμόζεται ένα νέο μοντέλο με μόνο αυτή τη μεταβλητή ως επεξηγηματική και δίνει τα ακόλουθα αποτελέσματα:

```
> cancer.cox1<-coxph(Surv(time,status)~Age)
> cancer.cox1
Call:
coxph(formula = Surv(time, status) ~ Age)


      coef exp(coef) se(coef)   z       p
Age 0.0324      1.03  0.00952 3.4 0.00067

Likelihood ratio test=11.8  on 1 df, p=0.000577  n= 51
```

Η ερμηνεία του εκτιμημένου συντελεστή του μοντέλου είναι ότι κάθε επιπρόσθετος χρόνος ζωής αυξάνει το λογάριθμο της συνάρτησης κινδύνου κατά 0.0324. Μια πιο σωστή προσέγγιση της ερμηνείας μπορεί να γίνει αφού πρώτα βρεθεί η εκθετική συνάρτηση του συντελεστή. Έπειτα για κάθε αύξηση κατά μία μονάδα της επεξηγηματική μεταβλητής, η συνάρτηση κινδύνου πολλαπλασιάζεται με τον εκθετικό συντελεστή. Η τιμή της

$$100(\exp(coefficient) - 1)$$

δίνει την ποσοστιαία αλλαγή στη συνάρτηση κινδύνου για κάθε μοναδιαία αύξηση στην επεξηγηματική μεταβλητή. Συνεπώς, αύξηση ενό χρόνου στην ηλικία διάγνωσης οδηγεί σε αύξηση 3% της συνάρτησης κινδύνου.

Το επόμενο στάδιο της ανάλυσης η ανάλυση των υπολοίπων από το μοντέλο με τη βοήθεια διαγνωστικών γραφημάτων για την εξεύρεση απομακρυσμένων τιμών, παρατηρήσεων επιρροής κ.α. Θα χρησιμοποιηθούν τρία διαφορετικά είδη υπολοίπων:

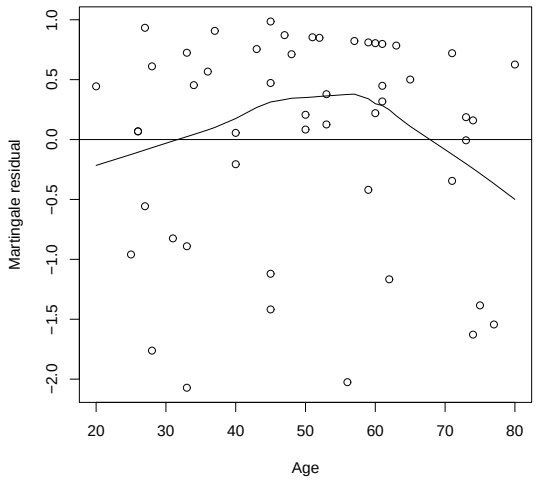
- martingale: χρήσιμα στο να αποκαλύπτουν τη συναρτησιακή μορφή των επεξηγηματικών μεταβλητών,

- deviance: χρήσιμα στην αναγνώριση των λιγότερο καλών εκτιμώμενων παρατηρήσεων,
- Schoenfeld: χρήσιμα στο να υποδεικνύουν αν το μοντέλο αναλόγων συναρτήσεων κινδύνου είναι κατάλληλο ή όχι.

Στην αρχή κατασκευάζεται το γράφημα των υπολοίπων martingale συναρτήσει της ηλικίας διάγνωσης χρησιμοποιώντας τις πιο κάτω εντολές,

```
> cancer.cox1.mart<-residuals(cancer.cox1,type="martingale")
> plot(Age,cancer.cox1.mart,ylab="Martingale residual")
> abline(h=0)
> lines(lowess(Age,cancer.cox1.mart))
```

Το γράφημα παρουσιάζεται στο Σχήμα 19.1. Η τελευταία εντολή δίνει τη γραμμή εξομάλυνσης lowess, η οποία υποδεικνύει ότι ίσως να πρέπει να θεωρηθεί ο τετραγωνικός παράγοντας της ηλικίας διάγνωσης στο μοντέλο, ή εναλλακτικά, να θεωρηθούν ξεχωριστά οι ηλικίες μικρότερες και μεγαλύτερες του 45.

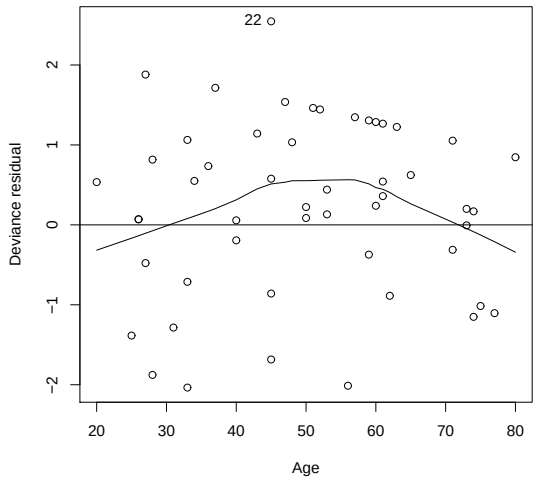


Σχήμα 19.1: Ηλικία διάγνωσης συναρτήσει των υπολοίπων martingale μαζί με τη γραμμή εξομάλυνσης lowess.

Στη συνέχεια, κατασκευάζεται με παρόμοιο τρόπο το γράφημα των υπολοίπων deviance συναρτήσει της ηλικίας διάγνωσης και παρουσιάζεται στο Σχήμα 19.2.

Μόνο το υπόλοιπο το οποίο αντιστοιχεί στην 22η παρατήρηση παρουσιάζεται ως απομακρυσμένη τιμή. Αυτό αντιστοιχεί σε άτομο με μηδενικό χρόνο επιβίωσης.

```
> cancer.cox1.dev<-residuals(cancer.cox1,type="deviance")
> plot(Age,cancer.cox1.dev,ylab="Deviance residual")
> abline(h=0)
> lines(lowess(Age,cancer.cox1.dev))
> identify(Age,cancer.cox1.dev,n=1)
[1] 22
```

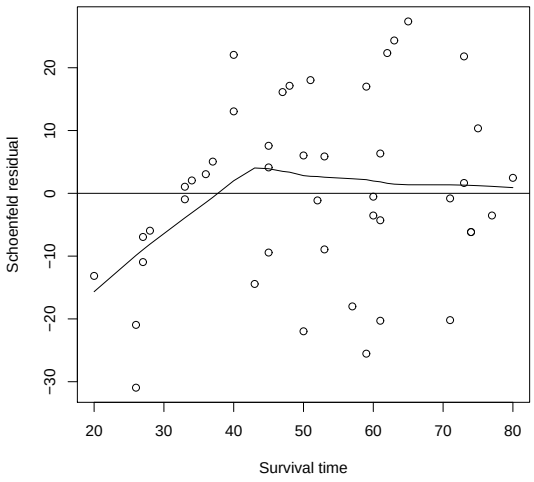


Σχήμα 19.2: Ηλικία διάγνωσης συναρτήσει των υπολοίπων deviance μαζί με τη γραμμή εξομάλυνσης lowess.

Τέλος, δίνεται το γράφημα των υπολοίπων Schoenfeld συναρτήσει του χρόνου επιβίωσης με την προσθήκη της γραμμής εξομάλυνσης lowess (Σχήμα 19.3).

```
> cancer.cox1.shoen<-residuals(cancer.cox1,type="schoenfeld")
> plot(Age[status==1],cancer.cox1.shoen,xlab="Survival time",
+ ylab="Schoenfeld residual")
> abline(h=0)
> lines(lowess(Age[status==1],cancer.cox1.shoen))
```

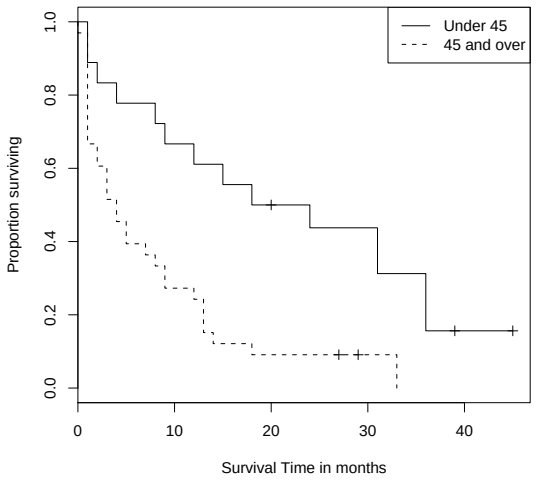
Το πιο σημαντικό συμπέρασμα είναι ότι όσο μικραίνει ο χρόνος επιβίωσης οι τιμές των υπολοίπων τείνουν να είναι αρνητικές. Αυτό αντιστοιχεί στους νεαρότερους ασθενείς και συνεπώς, υπάρχει και εδώ ένδειξη ότι ίσως να είναι καλύτερα να γίνει διαχωρισμός των ασθενών κατά ηλικία.



Σχήμα 19.3: Χρόνος επιβίωσης συναρτήσει των υπολοίπων Schoenfeld μαζί με τη γραμμή εξομάλυνσης lowess.

Λαμβάνοντας υπόψη την ένδειξη από την ανάλυση υπολοίπων, θα γίνει προσπάθεια ανάλυσης των δεδομένων εξετάζοντας τους ασθενείς βάσει της ηλικίας τους, διαχωρίζοντάς τους σε δύο κατηγορίες, μικρότερους και μεγαλύτερους από 45 ετών. Με τη βοήθεια των ακόλουθων εντολών κατασκευάζεται το γράφημα των καμπυλών διαβίωσης των δύο ηλικιακών κατηγοριών στο ίδιο γράφημα (Σχήμα 19.4).

```
> agroup<-cancer.dat[,"Age"]-45
> agroup[agroup>=0]<-1
> agroup[agroup<0]<-0
> plot(survfit(Surv(time,status)~agroup),xlab="Survival Time in months",
+ ylab="Proportion surviving",lty=1:2)
> legend("topright",c("Under 45","45 and over"),lty=1:2)
```



Σχήμα 19.4: Καμπύλη επιβίωσης για τα άτομα με ηλικία διάγνωσης κάτω και πάνω από 45 χρονών.

Όπως διαφαίνεται από το γράφημα, υπάρχει σημαντική διαφορά στις καμπύλες διαβίωσης των δύο κατηγοριών.Αυτό εξετάζεται και με τη βοήθεια του ελέγχου log-rank, ό οποίος είναι έλεγχος X^2 και εφαρμόζεται στην R με την εντολή `survdiff`.

```
> survdiff(Surv(time,status)~agroup)
Call:
survdiff(formula = Surv(time, status) ~ agroup)

          N Observed Expected (O-E)^2/E (O-E)^2/V
agroup=0 18       14     23.8      4.05      11.0
agroup=1 33       31     21.2      4.55      11.0

 Chisq= 11  on 1 degrees of freedom, p= 0.000926
```

Ο έλεγχος δίνει τιμή για την ελεγχοσυνάρτηση ίση με 11 με 1 βαθμό ελευθερίας. Το p-value του ελέγχου είναι πολύ κοντά στο 0 με αποτέλεσμα να απορρίπτεται η υπόθεση ότι οι δύο ηλικιακές κατηγορίες έχουν την ίδια συνάρτηση διαβίωσης.

Παράρτημα

Τα δεδομένα που χρησιμοποιούνται σε αυτό το κεφάλαιο για εφαρμογή της ανάλυσης διαβίωσης.

```
> cancer.dat
   Age Smear Infil Index Blasts Temp Resp Time Status
1   20    78    39     7    0.6  990    1   18      0
2   25    64    61    16   35.0 1030    1   31      1
3   26    61    55    12    7.5  982    1   31      0
4   26    64    64    16   21.0 1000    1   31      0
5   27    95    95     6    7.5  980    1   36      0
6   27    80    64     8    0.6 1010    0    1      0
7   28    88    88    10    4.8  986    1    9      0
8   28    70    70    14   10.0 1010    1   39      1
9   31    72    72     5    2.3  988    1   20      1
10  33    58    58     7    5.7  986    0    4      0
11  33    92    92     5    2.6  980    1   45      1
12  33    42    38    12    2.5  984    1   36      0
13  34    26    26     7    7.0  982    0   12      0
14  36    55    55    14    4.5  986    1    8      0
15  37    71    71    15    4.4 1020    0    1      0
16  40    91    91     9   35.0  986    1   15      0
17  40    52    49    12    2.1  988    1   24      0
18  43    74    63     4    0.1  986    0    2      0
19  45    78    47    14    4.2  980    1   33      0
20  45    60    36    10    0.6  992    1   29      1
21  45    82    32    10   28.1 1016    0    7      0
22  45    79    79     4    1.1 1030    0    0      0
23  47    56    28     2    0.9  990    0    1      0
24  48    60    54    10    2.2 1002    0    2      0
25  50    83    66    19   11.6  996    1   12      0
26  50    36    32    14    4.5  992    1    9      0
27  51    88    70     8    0.5  982    0    1      0
28  52    87    87     7   10.3  986    0    1      0
29  53    75    68    13    2.3  980    1    9      0
30  53    65    65     6    2.3  982    0    5      0
```

31	56	97	92	10	16.0	992	1	27	1
32	57	87	83	19	21.6	1020	0	1	0
33	59	45	45	8	1.1	999	0	13	0
34	59	36	34	5	0.0	1038	0	1	0
35	60	39	33	7	0.9	988	0	5	0
36	60	76	53	12	0.4	982	0	1	0
37	61	46	37	4	1.4	1006	0	3	0
38	61	39	8	8	0.3	990	0	4	0
39	61	90	90	1	9.9	990	0	1	0
40	62	84	84	19	115.0	1020	1	18	0
41	63	42	27	5	0.3	1014	0	1	0
42	65	75	75	10	20.0	1004	0	2	0
43	71	44	22	6	0.3	990	0	1	0
44	71	63	63	11	10.0	986	1	8	0
45	73	33	33	4	0.5	1010	0	3	0
46	73	93	84	6	38.0	1020	0	4	0
47	74	58	58	10	2.4	1002	1	14	0
48	74	32	30	16	6.7	988	0	3	0
49	75	60	60	17	8.2	990	1	13	0
50	77	69	69	9	1.5	986	1	13	0
51	80	73	73	7	1.5	986	0	1	0

Κεφάλαιο 20

Ανάλυση σε Κύριες Συνιστώσες και Διαχωριστική Ανάλυση

20.1 Ανάλυση σε Κύριες Συνιστώσες

Η μέθοδος των Κυρίων Συνιστωσών είναι μία τεχνική ανάλυσης δεδομένων με σκοπό τη δημιουργία καινούργιων μεταβλητών, οι οποίες είναι γραμμικοί συνδυασμοί των αρχικών μεταβλητών, έτσι ώστε να είναι ασυσχέτιστες μεταξύ τους και να περιέχουν όσο το δυνατόν μεγαλύτερο μέρος της διακύμανσης των αρχικών μεταβλητών. Οι νέες μεταβλητές που παράγονται ονομάζονται *Κύριες Συνιστώσες*. Το τι επιτυγχάνεται από τη μέθοδο αυτή είναι ότι από ένα σύνολο συσχετισμένων μεταβλητών καταλήγουμε σε ένα σύνολο ασυσχέτιστων μεταβλητών, το οποίο είναι χρήσιμο για αρκετές στατιστικές μεθόδους. Επίσης, οι κύριες συνιστώσες που προκύπτουν μπορούν να ερμηνεύσουν το μεγαλύτερο ποσοστό της διακύμανσης, που σημαίνει πως καταλήγουμε σε ένα πιο μικρό αριθμό μεταβλητών από ότι είχαμε αρχικά, με κόστος ότι χάνουμε ένα μικρό ποσοστό της συνολικής μεταβλητότητας. Αυτό είναι πολύ σημαντικό ιδιαίτερα στις περιπτώσεις που έχουμε λίγες παρατηρήσεις και πολλές μεταβλητές. Συνεπώς, αν σε μια τέτοια περίπτωση θέλαμε να εφαρμόσουμε ένα (γενικευμένο) γραμμικό μοντέλο, η υπερπαραμετροποίηση του μοντέλου μπόρει να ξεπεραστεί χρησιμοποιώντας την παραπάνω μέθοδο.

Για να δούμε πως εφαρμόζεται η μέθοδος αυτή στην R, θα χρησιμοποιήσουμε

το πλαίσιο δεδομένων possum το οποίο βρίσκεται στη βιβλιοθήκη DAAG και αναφέρεται σε εννέα μορφομετρικές μετρήσεις για 104 οπόσσουμ (δενδρόβιο μαρσιποφόρο ζώο). Στην R υπάρχουν δύο εντολές οι οποίες μπορούν να εφαρμόσουν την ανάλυση σε κύριες συνιστώσες. Προτιμητέα είναι η εντολή `prcomp()` με την οποία ο υπολογισμός γίνεται με την διάσπαση ιδιάζουσας τιμής του πίνακα δεδομένων (πιθανώς κανονικοποιημένος) και όχι υπολογίζοντας τις ιδιοτιμές με φασματική διάσπαση του πίνακα συνδιακύμανσης (ή συσχετίσεων) όπως γίνεται εφαρμόζοντας την εντολή `princomp()`. Το πρώτο βήμα είναι να ελέγξουμε τη μεταβλητότητα των μορφομετρικών μεταβλητών με τη βοήθεια του πίνακα συνδιακύμανσης, ο οποίος υπολογίζεται με την εντολή `cov()`.

```
> library(DAAG)
> possum.dat<-na.omit(possum[,6:14])
> cov(possum.dat)
           hdlngth      skullw  totlngth      taill    footlgth     earconch
hdlngth  12.719151 7.990574910 10.579248  2.0111222  6.13856368  1.475288407
skullw    7.990575 9.788319056  7.138732  1.5772178  3.78241767  0.002080716
totlngth 10.579248 7.138732153 18.597533  4.8039977  8.43164953  2.446375405
taill     2.011122 1.577217780  4.803998  3.8747858 -1.09253760 -3.171178374
footlgth  6.138564 3.782417666  8.431650 -1.0925376 19.31871312 13.957338664
earconch  1.475288 0.002080716  2.446375 -3.1711784 13.95733866 16.445541595
eye       1.351369 1.062334856  1.167453  0.4161098  0.02410813 -0.653397106
chest     4.661275 4.050000000  5.145588  0.7083333  4.07107843  1.691176471
belly     5.520779 3.925904245  6.160689  1.5989197  3.68086808  0.600737674
                eye     chest     belly
hdlngth   1.35136874 4.6612745 5.5207786
skullw    1.06233486 4.0500000 3.9259042
totlngth  1.16745288 5.1455882 6.1606891
taill     0.41610984 0.7083333 1.5989197
footlgth  0.02410813 4.0710784 3.6808681
earconch -0.65339711 1.6911765 0.6007377
eye       1.10688559 0.3240196 0.7081715
chest     0.32401961 4.2254902 3.4583333
belly     0.70817152 3.4583333 7.6600514
> diag(cov(possum.dat))
  hdlngth    skullw  totlngth     taill  footlgth  earconch       eye     chest
12.719151  9.788319 18.597533  3.874786 19.318713 16.445542  1.106886  4.225490
```

```
   belly
7.66005
```

Η διαγώνιος του πίνακα συνδιακύμανσης παρουσιάζει τη διασπορά και όπως είναι φανερό, οι μεταβλητές έχουν διαφορετική μεταβλητότητα. Ο λόγος είναι ότι το κάθε μέρος του μαρσιποφόρου ζώου το οποίο μετρήθηκε έχει διαφορετικό μέγεθος. Συνεπώς, για να δοθεί η ίδια βαρύτητα στις μεταβλητές, αυτές πρέπει να κανονικοποιηθούν. Αυτό γίνεται χρησιμοποιώντας τον πίνακα συσχετίσεων στην ανάλυση κυρίων συνιστωσών, αφού στη διαγώνιό του υπάρχουν μόνο μονάδες. Στην R αυτό γίνεται προσθέτοντας το όρισμα `scale=T` στην `prcomp()`.

```
> cor(possum.dat)
           hdlngth       skullw  totlngth      taill     footlgth      earconch
hdlngth  1.0000000 0.7161350133 0.6878569  0.2864742  0.391605342  0.1020055445
skullw   0.7161350 1.0000000000 0.5291019  0.2561025  0.275059297  0.0001639965
totlngth 0.6878569 0.5291019496 1.0000000  0.5659151  0.444831755  0.1398850635
taill    0.2864742 0.2561024623 0.5659151  1.0000000 -0.126276764 -0.3972580707
footlgth 0.3916053 0.2750592967 0.4448318 -0.1262768  1.000000000  0.7830498840
earconch 0.1020055 0.0001639965 0.1398851 -0.3972581  0.783049884  1.0000000000
eye      0.3601583 0.3227423866 0.2573122  0.2009244  0.005213424 -0.1531446219
chest    0.6358242 0.6297416468 0.5804553  0.1750553  0.450590285  0.2028739514
belly    0.5593137 0.4533877714 0.5161617  0.2934857  0.302583550  0.0535235532
                 eye     chest      belly
hdlngth   0.360158311 0.6358242 0.55931366
skullw    0.322742387 0.6297416 0.45338777
totlngth  0.257312171 0.5804553 0.51616171
taill     0.200924441 0.1750553 0.29348570
footlgth  0.005213424 0.4505903 0.30258355
earconch -0.153144622 0.2028740 0.05352355
eye       1.000000000 0.1498240 0.24320430
chest     0.149823996 1.0000000 0.60787245
belly     0.243204300 0.6078724 1.00000000

> possum.prc<-prcomp(possum.dat,scale=T)
> summary(possum.prc)
Importance of components:
                          PC1   PC2   PC3   PC4    PC5    PC6    PC7    PC8   PC9
Standard deviation     1.986 1.407 0.944 0.859 0.7508 0.5563 0.5165 0.4048 0.380
```

```
Proportion of Variance 0.438 0.220 0.099 0.082 0.0626 0.0344 0.0296 0.0182 0.016
Cumulative Proportion  0.438 0.658 0.757 0.839 0.9018 0.9362 0.9658 0.9840 1.000
```

Η εντολή `summary()`, με όρισμα το αντικείμενο της ανάλυσης σε κύριες συνιστώσες, δίνει τις ιδιοτιμές για κάθε κύρια συνιστώσα οι οποίες αντιστοιχούν στις τυπικές αποκλίσεις τους. Αυτές δίνονται σε αύξουσα σειρά. Συνεπώς, στην πρώτη κύρια συνιστώσα (PC1) αντιστοιχεί η μεγαλύτερη ιδιοτιμή, στην δεύτερη (PC2) η δεύτερη μεγαλύτερη κ.ο.κ. Επίσης, στη δεύτερη γραμμή παρουσιάζεται το ποσοστό της μεταβλητότητας που εξηγείται από κάθε κύρια συνιστώσα, ενώ στην τρίτη γραμμή παρουσιάζεται το αθροιστικό ποσοστό μεταβλητότητας.

Στη συνέχεια, επιλέγουμε τις k πρώτες κύριες συνιστώσες που εξηγούν ένα σημαντικό ποσοστό της ολικής μεταβλητότητας. Στη βιβλιογραφία υπάρχουν πολλά κριτήρια τα οποία χρησιμοποιούνται για τον σκοπό αυτό. Πιο κάτω αναφέρονται τα τρία πιο δημοφιλή:

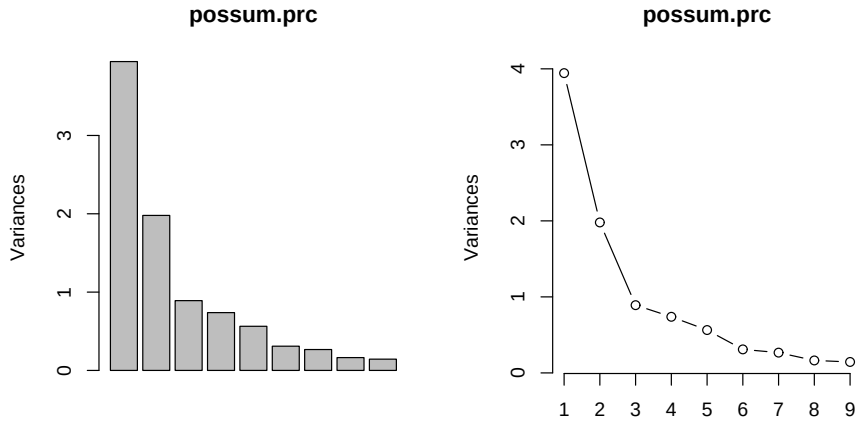
1. Ποσοστό συνολικής διακύμανσης που εξηγούν οι κύριες συνιστώσες
2. Κριτήριο του Kaiser
3. Τεχνική του αγκώνα - scree plot.

Σύμφωνα με το πρώτο κριτήριο, διαλέγουμε τόσες συνιστώσες ώστε αθροιστικά να εξηγούν μεγαλύτερο ποσοστό μεταβλητότητας από τον στόχο που θέσαμε στην αρχή (συνήθως μεγαλύτερο του 80%). Άρα, στο παραπάνω παράδειγμα επιλέγονται οι πρώτες τέσσερις συνιστώσες που εξηγούν αθροιστικά το 83.9% της συνολικής μεταβλητότητας των δεδομένων. Το κριτήριο του Kaiser λέει, αν χρησιμοποιείται ο πίνακας συνδιακύμανσης, να θεωρήσουμε τόσες συνιστώσες όσες η ιδιοτιμή τους είναι μεγαλύτερη από από τη μέση τους τιμή, ενώ αν χρησιμοποιείται ο πίνακας συσχετίσεων, τόσες όσες έχουν ιδιοτιμή μεγαλύτερη του 1 (γιατί;). Συνεπώς, στο παράδειγμα επιλέγονται οι δύο πρώτες κύριες συνιστώσες. Τέλος, η τεχνική του αγκώνα είναι μια γραφική μέθοδος για την επιλογή του αριθμού των κυρίων συνιστωσών. Το γράφημα αυτό έχει στον οριζόντιο άξονα των x τη σειρά και στον κάθετο άξονα των y την τιμή κάθε ιδιοτιμής. Προτείνεται να επιλεγούν τόσες συνιστώσες μέχρι το γράφημα αρχίζει να δημιουργεί «αγκώνα», δηλαδή να αλλάζει κλίση. Το scree plot του παραδείγματος παρουσιάζεται στο Σχήμα 20.1. Το πρώτο παρουσιάζεται ως ραβδόγραμμα, ενώ το δεύτερο ως πολύγωνο, αφού προσθέσαμε το όρισμα type="line". Είναι φανερό ότι η κλίση αλλάζει από την τρίτη ιδιοτιμή και μετά, και άρα επιλέγονται οι δύο πρώτες συνιστώσες. Συμπερασματικά, για

το παραπάνω παράδειγμα είναι κατάλληλη η επιλογή των δύο πρώτων κυρίων συνιστωσών PC1 και PC2.

Το πιο δύσκολο κομμάτι της ανάλυσης σε κύριες συνιστώσες είναι ίσως η ερμηνεία τους. Αυτή βασίζεται στους συντελεστές των κυρίων συνιστωσών (principal component loadings) που συνιστούν το γραμμικό συνδυασμό των αρχικών μεταβλητών. Οι συντελεστές αυτοί αντιστοιχούν στα στοιχεία των ιδιοδιανυσμάτων των ιδιοτιμών των συνιστωσών. Αυτοί δίνονται στην R μαζί με τις αντίστοιχες ιδιοτιμές καλώντας το αντικείμενο της ανάλυσης σε κύριες συνιστώσες που δημιουργήθηκε χρησιμοποιώντας την εντολή `prcomp()`. Θα δοθεί η ερμηνεία για τις πρώτες δύο συνιστώσες αφού αυτές επιλέγηκαν με βάση τα πιο πάνω κριτήρια. Η πρώτη κύρια συνιστώσα μπορεί να ερμηνευθεί ως ένας σταθμισμένος μέσος όρος των μεταβλητών, αφού όλοι οι συντελεστές τους έχουν το ίδιο πρόσημο. Η δεύτερη κύρια συνιστώσα κάνει σύγκριση του μήκους της ουράς (`taill`) και του μεγέθους του ματιού (`eye`) με το μήκος του ποδιού (`footlgth`) και της κόγχης του αυτιού (`earconch`) των οπόσσουμ.

```
> par(mfrow=c(1,2))
> screeplot(possum.prc)
> screeplot(possum.prc,type="line")
```



Σχήμα 20.1: Scree plot.

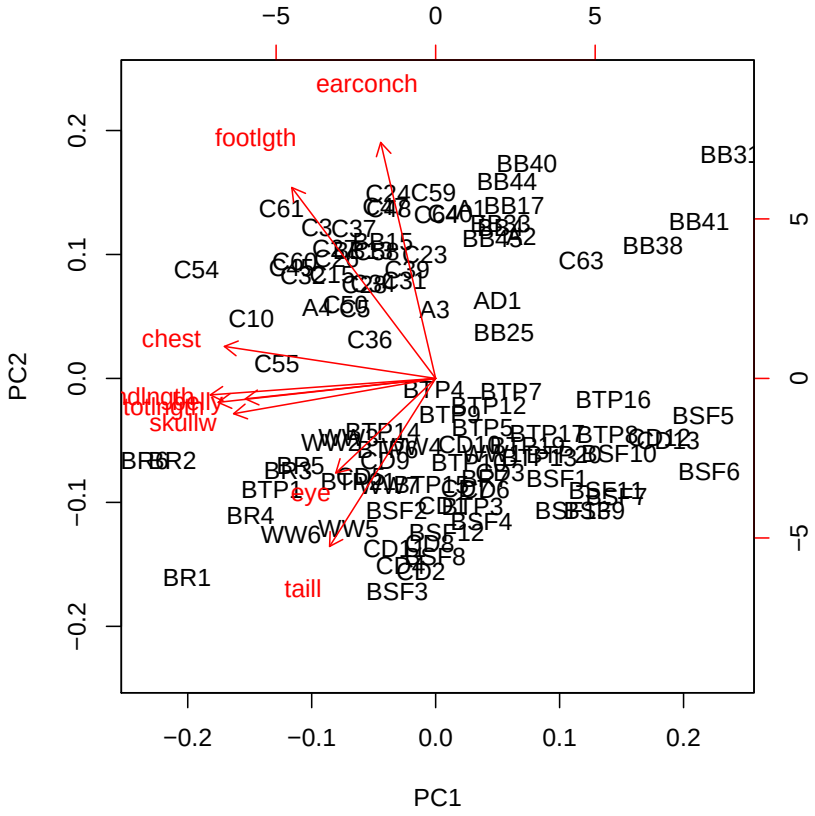
```
> possum.prc
Standard deviations:
[1] 1.9858729 1.4068546 0.9439907 0.8591514 0.7508320 0.5562584 0.5164833
[8] 0.4047894 0.3795100

Rotation:
                PC1         PC2          PC3         PC4          PC5
hdlngth  -0.4380078 -0.04519844  0.138623007 -0.07495574  0.238115223
skullw   -0.3927120 -0.09678900  0.223585881 -0.32364347  0.501176778
totlngth -0.4227082 -0.06564460 -0.312750646  0.33593765  0.120510988
taill    -0.2061992 -0.46100652 -0.514351991  0.38374763  0.026090130
footlgth -0.2796247  0.52384020 -0.062416712  0.27857386  0.002538413
earconch -0.1067739  0.64749223 -0.004813129  0.26143915 -0.017773149
eye      -0.1940729 -0.25932151  0.748023649  0.49876897 -0.192020543
chest    -0.4106589  0.08780572 -0.037517222 -0.41050426 -0.058210391
belly    -0.3699061 -0.05604043 -0.059633425 -0.25745316 -0.797707163
```

```
                PC6          PC7          PC8         PC9
hdlngth  -0.567980976  0.447545311 -0.439147036  0.09393289
skullw   -0.028120923 -0.617005313  0.216587933  0.04581025
totlngth -0.023209818  0.329000800  0.693643612 -0.04355347
taill     0.149454731 -0.307513951 -0.429684531  0.17585120
footlgth  0.026742546 -0.203737843 -0.229883084 -0.68616711
earconch  0.004502918 -0.142151245 -0.018410561  0.69289065
eye       0.221844208  0.006576962 -0.007914748  0.02419833
chest     0.720163882  0.313950090 -0.170795256  0.06299215
belly    -0.291796339 -0.236383757  0.109774529  0.02794910
```

Όπως έχει αναφερθεί στην αρχή του κεφαλαίου, αφού έγινε η επιλογή των κυρίων συνιστωσών με τις οποίες χάνεται κάποια πληροφορία, αυτές μπορούν να χρησιμοποιηθούν στη συνέχεια σε ένα (γενικευμένο) γραμμικό μοντέλο. Σε αυτό το μοντέλο θα χρησιμοποιηθούν οι τιμές των κυρίων συνιστωσών, οι οποίες υπολογίζονται για κάθε γραμμή των δεδομένων από τον γραμμικό συνδυασμό των τιμών των αρχικών μεταβλητών, με συντελεστές τους συντελεστές από τις κύριες συνιστώσες (principal components loadings). Αυτές οι τιμές ονομάζονται principal components scores και στην R υπολογίζονται από την εντολή `predict()` με όρισμα το αντικείμενο της ανάλυσης σε κύριες συνιστώσες. Έτσι, οι τιμές των πρώτων δύο κυρίων συνιστωσών στο πιο πάνω παράδειγμα δίνονται από:

```
> predict(possum.prc)[,1:2]
           PC1     PC2
C3    -1.93077  1.7406
C5    -1.31261  0.8120
C10   -2.99606  0.6969
C15   -1.68351  1.2040
C23   -0.17734  1.4358
.          .       .
.          .       .
.          .       .
BTP16  2.88827 -0.2431
BTP17  1.81839 -0.6301
BTP19  1.49279 -0.7675
BTP20  2.08842 -0.8750
BTP21 -1.25482 -1.1833
```

Τέλος, ένα χρήσιμο γράφημα που χρησιμοποιείται στην ανάλυση σε κύριες συνιστώσες είναι το biplot. Είναι η γραφική παράσταση των τιμών των κυρίων συνιστωσών (principal components scores) και των συντελεστών (principal components loadings) ταυτόχρονα στο ίδιο γράφημα. Οι μεταβλητές παριστάνονται με βέλη που ξεκινούν από την αρχή των αξόνων. Αν τα βέλη είναι κάθετα μεταξύ τους τότε οι αντίστοιχες μεταβλητές δε συσχετίζονται μεταξύ τους, ενώ αντίθετα, αν τα βέλη δείχνουν προς την ίδια (ή αντίθετη) κατεύθυνση τότε οι μεταβλητές συσχετίζονται ισχυρά θετικά (ή αρνητικά). Η αντίστοιχη εντολή στην R είναι η `biplot()` και στο Σχήμα 20.2 παρουσιάζεται το γράφημα για το πιο πάνω παράδειγμα.



Σχήμα 20.2: Biplot.

20.2 Διαχωριστική Ανάλυση

Η βασική ιδέα της διαχωριστικής ανάλυσης (discriminant analysis) είναι να κατατάξει παρατηρήσεις (συνήθως πολυδιάστατες) σε γνωστούς πληθυσμούς με γνωστές κατανομές για κάθε πληθυσμό. Η διαχωριστική ανάλυση αποτελεί μια μέθοδο με πλήθος εφαρμογών σε πολλές επιστήμες.

Έστω ότι υπάρχουν k υπο-πληθυσμοί (ομάδες), $\Pi_1, \Pi_2, \ldots, \Pi_k$ με $k \geq 2$. Για τον κάθε υπο-πλυθυσμό Π_k υπάρχει και μία κατανομή, f_k. Η διαχωριστική ανάλυση έχει 2 στόχους :

1. Τη διαχώριση ενός πληθυσμού σε ευδιάκριτα σύνολα (υπο-πληθυσμούς) και
2. Την ταξινόμηση παρατηρήσεων στους προηγούμενους γνωστούς πληθυσμούς με γνωστές κατανομές για κάθε πληθυσμό, με τη βοήθεια ενός κανόνα.

Αυτό που θα εξεταστεί εδώ είναι το πως οι επεξηγηματικές μεταβλητές συνεισφέρουν στην σωστή ταξινόμηση των ατόμων, των οποίων η ιδιότητα είναι ήδη γνωστή (supervised classification). Για τη διαχώριση σε k ομάδες χρειάζονται $k-1$ διαχωριστές (descriminators).

Οι συναρτήσεις που χρειάζονται για τη διαχωριστική ανάλυση βρίσκονται στη βιβλιοθήκη `MASS`. Ως παράδειγμα θα αναπτυχθεί η διαχωριστική ανάλυση στο πλαίσιο δεδομένων `possum`, το οποίο χρησιμοποιήθηκε και στην πιο πάνω ανάλυση σε κύριες συνιστώσες. Αυτό που θα εξεταστεί είναι το αν είναι δυνατόν, βάσει των μορφομετρικών μετρήσεων, να διακριθούν τα ζώα από διάφορες περιοχές (`sites`). Αυτό γίνεται χρησιμοποιώντας την εντολή `lda` όπως πιο κάτω.

```
> library(MASS)
> here <- !is.na(possum$footlgth)
> possum.lda <- lda(site ~ hdlngth+skullw+totlngth+ taill+footlgth+
+   earconch+eye+chest+belly, data=possum, subset=here)
> possum.lda
Call:
lda(site ~ hdlngth + skullw + totlngth + taill + footlgth + earconch +
    eye + chest + belly, data = possum, subset = here)

Prior probabilities of groups:
      1       2       3       4       5       6       7
0.32039 0.11650 0.06796 0.06796 0.12621 0.12621 0.17476
```

```
Group means:
  hdlngth skullw totlngth taill footlgth earconch   eye chest belly
1   93.72  57.20    89.71 36.38    73.00    52.58 15.02 27.88 33.27
2   89.85  55.13    81.67 34.67    70.75    52.11 14.37 26.29 31.17
3   94.57  58.90    88.07 37.21    66.60    45.26 16.07 27.57 34.86
4   97.61  61.69    92.24 39.71    68.93    45.84 15.47 29.64 34.57
5   92.18  56.23    86.92 37.65    64.73    43.87 15.38 26.65 32.04
6   89.25  54.19    84.54 37.65    63.07    43.97 15.34 25.23 31.50
7   92.63  57.23    85.69 37.69    65.74    45.86 14.47 26.14 31.92

Coefficients of linear discriminants:
               LD1      LD2      LD3      LD4      LD5      LD6
hdlngth  -0.150534  0.05833 -0.25569 -0.01238 -0.08195 -0.18713
skullw   -0.026530  0.03985 -0.24977  0.12448 -0.13648  0.14378
totlngth  0.106957  0.27924  0.30519 -0.18493 -0.13904 -0.08918
taill    -0.450631 -0.08960 -0.44579 -0.17304  0.32416  0.49507
footlgth  0.301903 -0.03601 -0.03909  0.07558  0.11910 -0.12606
earconch  0.586270 -0.04357 -0.07310 -0.08819 -0.06376  0.28015
eye      -0.056141  0.08921  0.78453  0.46439  0.28481  0.29723
chest     0.090620  0.10418  0.04985  0.12749  0.64746 -0.07871
belly     0.009966 -0.05166  0.09358  0.16715 -0.29033  0.19391

Proportion of trace:
   LD1    LD2    LD3    LD4    LD5    LD6
0.8927 0.0557 0.0365 0.0082 0.0047 0.0022

> options(digits=4)
> possum.lda$svd   # Examine the singular values
[1] 15.7578  3.9372  3.1860  1.5078  1.1420  0.7772
```

Το αντικείμενο της διαχωριστικής ανάλυσης (στο παράδειγμα, possum.lda) περιέχει τις αρχικές αναλογίες (prior probabilities) των επιπέδων της εξαρτημένης μεταβλητής, τις μέσες τιμές κάθε επεξηγηματικής μεταβλητής για κάθε επίπεδο της εξαρτημένης μεταβλητής, τους συντελεστές των διαχωριστών και την αναλογία του ίχνους κάθε διαχωριστή. Οι διαχωριστές ταξινομούνται σε σειρά, από αυτόν με το μεγαλύτερο ίχνος σε αυτόν με το μικρότερο ίχνος. Από την αναλογία του ίχνους κάθε διαχωριστή παρατηρείται ότι οι τελευταίοι τρεις έχουν πολύ μικρή

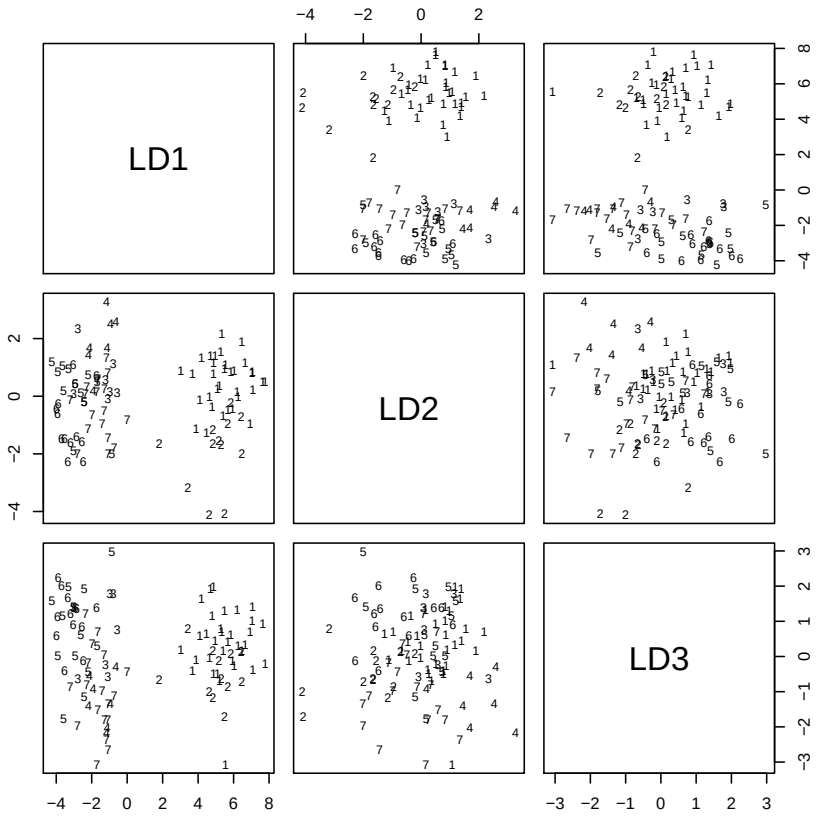
διαχωριστική ικανότητα. Οι μεταβλητές που φαίνονται πιο σημαντικές είναι οι hdlngth και tail, συγκρινόμενες με τις totlngth και footlngth. Το συμπέρασμα αυτό βγαίνει από τις τιμές των συντελεστών των διαχωριστών. Η τελευταία εντολή δίνει τις ιδιάζουσες τιμές, οι οποίες είναι ο λόγος του μεταξύ (between) προς του ανάμεσα (within) αθροίσματος τετραγώνων, για όλους τις κανονικές μεταβλητές (canonical variates) σε σειρά, από αυτήν με την μεγαλύτερη σε αυτήν με τη μικρότερη ιδιάζουσα τιμή. Είναι φανερό και πάλι ότι οι κανονικές μεταβλητές μετά την τρίτη θα έχουν λίγη ή σχεδόν καθόλου διαχωριστική δύναμη. Στη συνέχεια δίνεται ο πίνακας διαγραμμάτων διασποράς για τις τιμές (scores) των τριών πρώτων κανονικών μετβλητών (Σχήμα 20.3).

```
> plot(possum.lda, dimen=3)
```

Με την εντολή predict και όρισμα το αντικείμενο της διαχωριστικής ανάλυσης παρουσιάζεται η εκτίμησης της ταξινόμησης της εξαρτημένης μεταβλητής (class), και οι τιμές (scores) των κανονικών μεταβλητών. Πιο κάτω παρουσιάζεται μόνο η εκτίμηση της ταξινόμησης.

```
> predict(possum.lda)$class
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2
 [38] 2 2 2 2 2 2 2 2 3 3 3 3 3 4 3 4 4 4 4 4 4 7 7 5 7 5 5 6 5 5 5 7 5 6 6 6 3
 [75] 6 6 7 6 6 6 6 5 6 5 6 7 5 7 7 7 7 7 7 5 7 7 7 7 7 7 7 7 7
Levels: 1 2 3 4 5 6 7
```

Τέλος, για να ελεγχθεί η ακρίβεια της πρόβλεψης της ταξινόμησης από τη διαχωριστική ανάλυση, κατασκευάζεται ο πίνακας συνάφειας με την αρχική ταξινόμηση των δεδομένων και στη συνέχεια υπολογίζεται το ποσοστό της σωστής ως προς τη λανθασμένη ταξινόμηση. Στο παράδειγμα, η ακρίβεια της εκτίμησης υπολογίστηκε ίση με 83.5%.



Σχήμα 20.3: Διάγραμμα διασπορών των τιμών των τριών πρώτων κανονικών μεταβλητών.

```
> tab<-table(possum[here,]$site,predict(possum.lda)$class)
> tab

    1  2  3  4  5  6  7
  1 33  0  0  0  0  0  0
  2  3  9  0  0  0  0  0
  3  0  0  6  1  0  0  0
  4  0  0  0  6  0  0  1
  5  0  0  0  0  7  3  3
  6  0  0  1  0  2  9  1
  7  0  0  0  0  2  0 16
> sum(diag(tab))/sum(tab)
[1] 0.835
```

Ένας τρόπος για να γίνει σταυρωτή-επικύρωση (cross-validation) είναι να αφαιρούνται παρατηρήσεις, μία κάθε φορά, και μετά χρησιμοποιώντας τα εναπομείναντα δεδομένα να γίνει εκτίμηση για την παρατήρηση που αφαιρέθηκε. Για να εκτελεστεί αυτή η μέθοδος, είναι αρκετό να προστεθεί το όρισμα `CV=T` στην εντολή της διαχωριστικής ανάλυσης. Πρέπει να αναφερθεί ότι τώρα η πρόβλεψη της ταξινόμησης δίνεται απ' ευθείας από το αντικείμενο της διαχωριστικής ανάλυσης.

```
> possum.lda.cv <- lda(site ~ hdlngth+skullw+totlngth+ taill+footlgth+
+   earconch+eye+chest+belly, data=possum, subset=here, CV=T)
> tab.cv<-table(possum[here,]$site,possum.lda.cv$class)
> tab.cv

    1  2  3  4  5  6  7
  1 31  2  0  0  0  0  0
  2  4  7  0  0  0  0  1
  3  0  0  3  1  1  0  2
  4  0  0  1  4  0  0  2
  5  0  0  1  0  4  4  4
  6  0  0  1  0  4  7  1
  7  0  0  1  2  3  0 12
> sum(diag(tab.cv))/sum(tab.cv)
[1] 0.6602
```

Κεφάλαιο 21

Ανάλυση Κατά Συστάδες στην R [1]

21.1 Εισαγωγή

Συστάδα θεωρούμε μια συλλογή από στοιχεία τα οποία είναι όμοια μεταξύ τους (ή βρίσκονται κοντά) και έχουν διαφορές (ή βρίσκονται μακριά) από στοιχεία που ανήκουν σε άλλες συστάδες.

Η ανάλυση κατά συστάδες αποσκοπεί στο διαχωρισμό μιας συλλογής από στοιχεία σε υποσύνολα έτσι ώστε να υπάρχει ομοιογένεια μέσα σε ένα υποσύνολο και ανομοιογένεια μεταξύ των στοιχείων που ανήκουν σε διαφορετικά υποσύνολα. Επιπρόσθετα μπορεί να αποσκοπεί στην ιεραχική οργάνωση των συστάδων με την διαδοχική ομαδοποίηση αυτών, έτσι ώστε σε κάθε στάδιο της ιεραρχίας, οι συστάδες που ανήκουν στην ίδια ομάδα να είναι πιο όμοιες μεταξύ τους από αυτές που ανήκουν σε άλλη ομάδα [1, 2].

Σημαντική έννοια στην ανάλυση κατά συστάδες είναι η απόσταση (ή ομοιότητα), δηλαδή το μέτρο βάση του οποίου δημιουργούνται οι συστάδες. Παραδείγματα μετρικών που μπορούν να χρησιμοποιηθούν ως απόσταση μεταξύ δύο διανυσμάτων $x = (x_1, \ldots, x_p)$ και $y = (y_1, \ldots, y_p)$ είναι:

- Η μετρική Minkowski $d(x, y) = \left[\sum_{i=1}^{p} |x_i - y_i|^m\right]^{\frac{1}{m}}$.
- Για $m = 2$ στην μετρική Minkowski παίρνουμε την Ευκλείδια απόσταση

[1] Το κεφάλαιο στηρίζεται σε ανεξάρτητη εργασία του Α. Ιωάννου

$d(x,y) = \sqrt{(x_1 - y_1)^2 + \ldots + (x_p - y_p)^2}$.

- Η μέγιστη απόσταση $d(x,y) = \max\{(x_1 - y_1), \ldots, (x_p - y_p)\}$.
- Η μετρική Canberra $d(x,y) = \sum_{i=1}^{p} \frac{|x_i - y_i|}{(x_i - y_i)}$.

Η απόσταση μπορεί να χρησιμοποιηθεί για να κατασκευαστεί πίνακας αποστάσεων σε κάθε στάδιο της ανάλυσης. Ο πίνακας αυτός θα έχει μηδενικά στοιχεία στοιχεία στη διαγώνιο και την απόσταση μεταξύ του i στοιχείου (ή συστάδας) και του j στοιχείου (ή συστάδας) στην θέση (i,j). Ο πίνακας μπορεί να υπολογιστεί στην R με την εντολή `dist(dataset, method)`, όπου `method` κάποια από τις διαθέσιμες μετρικές όπως τις πιο πάνω και `dataset` ο πίνακας των δεδομένων. Όταν δεν προσδιοριστεί κάποια συγκεκριμένη μετρική, χρησιμοποιείται η Ευκλείδια απόσταση.

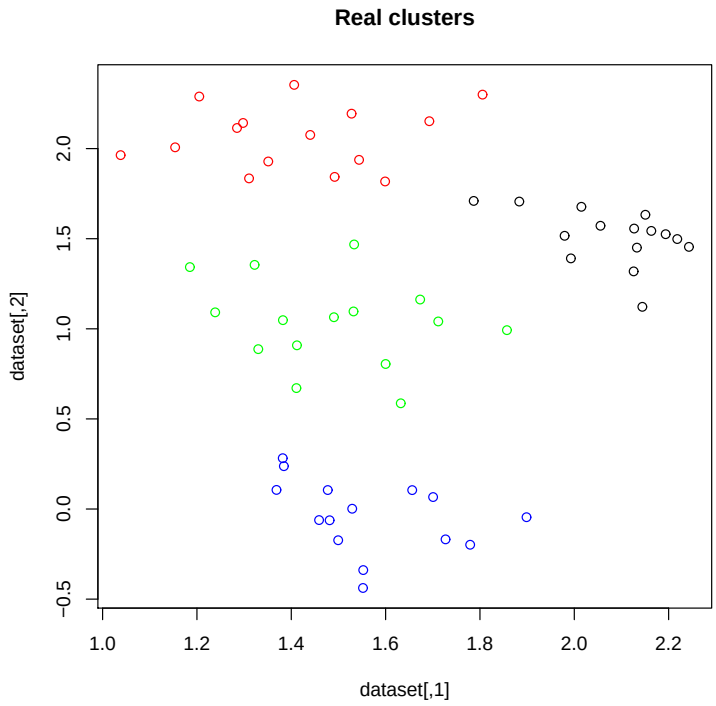
Παρουσιάζουμε παρακάτω ορισμένες μεθόδους ανάλυσης κατά συστάδων. Αρχικά δημιουργούμε ένα σύνολο δεδομένων από 4 γνωστές συστάδες για να ελέγξουμε την αποτελεσματικότητα των μεθόδων ανάλυσης.

```
library(MASS)
library(lattice)
library(cluster)
Sigma<- matrix(c(0.04,0,0,0.04),c(2,2))
x<- mvrnorm(15,c(1.5,2), Sigma)
y<- mvrnorm(15,c(2,1.5), Sigma)
z<- mvrnorm(15,c(1.5,1), Sigma)
w<- mvrnorm(15,c(1.5,0), Sigma)
```

Οι συστάδες θα έχουν 15 στοιχεία η κάθε μια. Τα στοιχεία θα προέρχονται από διδιάστατες κανονικές κατανομές με μέσες τιμές $\mu_1 = (1.5, 2), \mu_2 = (2, 1.5), \mu_3 = (1.5, 1)$ και $\mu_4 = (1.5, 0)$ αντίστοιχα και κοινό πίνακα συνδιακυμάνσεων $\Sigma = \begin{pmatrix} 0.04 & 0 \\ 0 & 0.04 \end{pmatrix}$.

```
x<- cbind(x,1)
y<- cbind(y,2)
z<- cbind(z,3)
w<- cbind(w,4)
data1 <- data.frame(rbind(x,y,z,w))
dataset<- cbind(data1$X1,data1$X2)
distmatrix<- dist(dataset)
mycol <- c("red", "black", "green", "blue")
plot(dataset,col=mycol[data1$X3],main="Real clusters")
```

Συσχετίζουμε κάθε στοιχείο με τον αριθμό της συστάδας από την οποία προέρχεται. Ακολούθως υπολογίζουμε τον πίνακα με τις Ευκλείδιες αποστάσεις για τα δεδομένα για να χρησιμοποιηθεί στη συνέχεια για την ανάλυση σε συστάδες. Τέλος δημιουργούμε το γράφημα των δεδομένων χρησιμοποιώντας διαφορετικό χρώμα για κάθε συστάδα.



Σχήμα 21.1: Πραγματικές συστάδες

21.2 Ιεραρχική Ανάλυση κατά Συστάδες

Προσθετική μέθοδος (Agglomerative Hierarchical Clustering)

Περιγραφή

1. Αρχίζουμε με N συστάδες, με την κάθε μία να περιέχει μόνο ένα στοιχείο και ενα $N \times N$ πίνακα με αποστάσεις.

2. Βρίσκουμε στον πίνακα το ζεύγος U και V συστάδων με την μικρότερη απόσταση μεταξύ τους.

3. Ενώνουμε τις συστάδες U και V σε μια συστάδα, έστω UV. Ανανεώνουμε τον πίνακα αποστάσεων διαγράφοντας τις γραμμές και στήλες που αντιστοιχούν στις U και V και προσθέτοντας μια γραμμή και μια στήλη με τις αποστάσεις της UV από τις υπόλοιπες συστάδες.

4. Επαναλαμβάνουμε τα βήματα 2 και 3 $(N-1)$ φορές μέχρι να υπάρχει μόνο μια συστάδα. Καταγράφουμε τις συστάδες που δημιουργήθηκαν κατά τη διάρκεια της διαδικασίας και το επίπεδο (απόσταση) στο οποίο δημιουργήθηκε η κάθε μία.

Επιλογές για απόσταση μεταξύ συστάδων

(α) Single Linkage: hclust (distmatrix , method="single")
Ως απόσταση μεταξύ δύο συστάδων U και V θεωρούμε την απόσταση με την μικρότερη τιμή από όλες τις πιθανές αποστάσεις μεταξύ ενός στοιχείου (ή συστάδας) του U και ενός στοιχείου (ή συστάδας) του V.

(β) Complete linkage: hclust(distmatrix,method="complete")
Ως απόσταση μεταξύ δύο συστάδων U και V θεωρούμε την απόσταση με την μεγαλύτερη τιμή από όλες τις πιθανές αποστάσεις μεταξύ ενός στοιχείου (ή συστάδας) του U και ενός στοιχείου (ή συστάδας) του V.

(γ) Average linkage: hclust(distmatrix,method="average")
Ως απόσταση μεταξύ δύο συστάδων U και V θεωρούμε την μέση απόσταση μεταξύ των δύο συστάδων (το άθροισμα όλων των πιθανών αποστάσεων μεταξύ ενός στοιχείου του U και ενός στοιχείου του V διά του γινομένου του πλήθους των στοιχείων της U επί του πλήθους των στοιχείων της V).

(δ) Ward's Hierarchical Clustering: hclust(distmatrix,method="ward")
Για κάθε συστάδα k θεωρούμε ως ESS_k το άθροισμα των τετραγώνων των

αποστάσεων κάθε στοιχείου της συστάδας από τον μέσο της συστάδας και ESS το άθροισμα των ESS_k. Ως απόσταση μεταξύ δύο συστάδων U και V θεωρούμε την αύξηση που θα προκύψει στο ESS από την ένωση των δύο συστάδων.

Εφαρμογή

```
hrs<-hclust(distmatrix,method="single")
hrc<-hclust(distmatrix,method="complete")
hra<-hclust(distmatrix,method="average")
hrw<-hclust(distmatrix,method="ward")
```

Εφαρμόζουμε την προσθετική μέθοδο της ιεραρχικής ανάλυσης κατά συστάδες για τις 4 διαφορετικές επιλογές αποστάσεων.

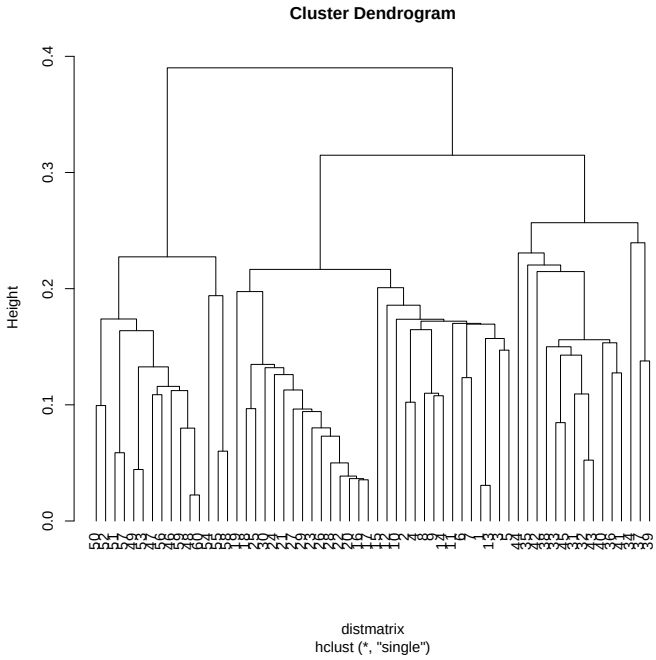
```
membs<- cutree(hrs,k=4)
membc<- cutree(hrc,k=4)
memba<- cutree(hra,k=4)
membw<- cutree(hrw,k=4)
```

Χρησιμοποιούμε την εντολή cutree για να χωρίσουμε τα δεδομένα σε 4 συστάδες. Η εντολή επιστρέφει ένα διάνυσμα μήκους όσο και το πλήθος των δεδομένων,το οποίο έχει τιμές που υποδηλώνουν σε ποια συστάδα ανήκει το αντίστοιχο στοιχείο.

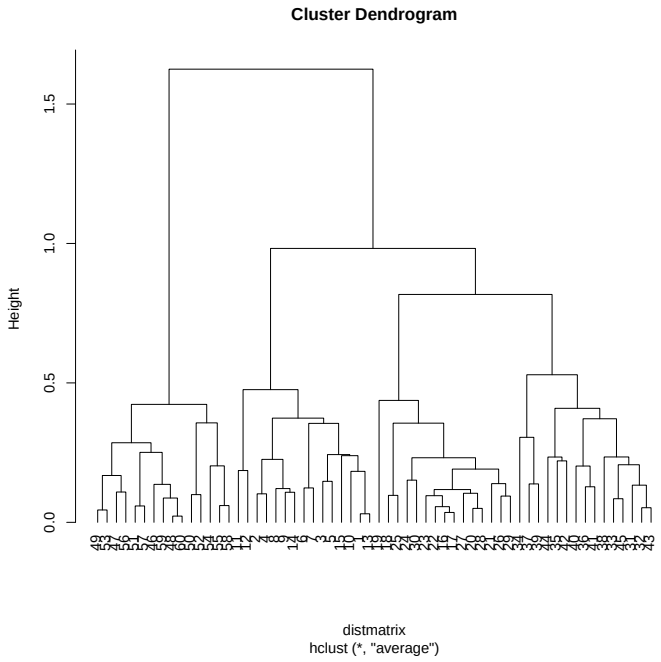
```
> c(sum(membs==1),sum(membs==2),sum(membs==3),sum(membs==4))
[1] 30 12  3 15
> c(sum(membc==1),sum(membc==2),sum(membc==3),sum(membc==4))
[1] 15 15 15 15
> c(sum(memba==1),sum(memba==2),sum(memba==3),sum(memba==4))
[1] 15 15 15 15
> c(sum(membw==1),sum(membw==2),sum(membw==3),sum(membw==4))
[1] 15 15 15 15
> hrs$order
  50 52 51 57 49 53 47 56 46 59 48 60 54 55 58 19 18 25 30 24 21 27 29 23 26
  28 22 20 16 17 15 12 10  2  4  8  9 14 11  6  7  1 13  3  5 44 35 42 38 33
  45 31 32 43 40 36 41 34 37 39
> hrc$order
  49 53 47 56 50 52 54 55 58 51 57 46 59 48 60 11 12  2  4  8  9 14  6  7  3
  5 15 10  1 13 19 30 18 25 21 23 22 26 20 28 27 16 17 24 29 44 35 42 40 32
  43 36 41 33 45 31 38 34 37 39
> hra$order
  49 53 47 56 51 57 46 59 48 60 50 52 54 55 58 11 12  2  4  8  9 14  6  7  3
  5 15 10  1 13 19 18 25 24 30 23 22 16 17 27 20 28 21 26 29 34 37 39 44 35
  42 40 36 41 38 33 45 31 32 43
```

```
> hrw$order
  49 53 47 56 51 57 54 55 58 50 52 46 59 48 60  6  7  3  5  1 13 10 15 11 12
  2  4  8  9 14 18 25 21 26 29 23 22 16 17 27 20 28 19 24 30 34 37 39 44 35
  42 40 36 41 32 43 33 45 31 38
 plot(hrs,hang=-1); plot(hra,hang=-1); plot(hrc,hang=-1); plot(hrw,hang=-1)
```

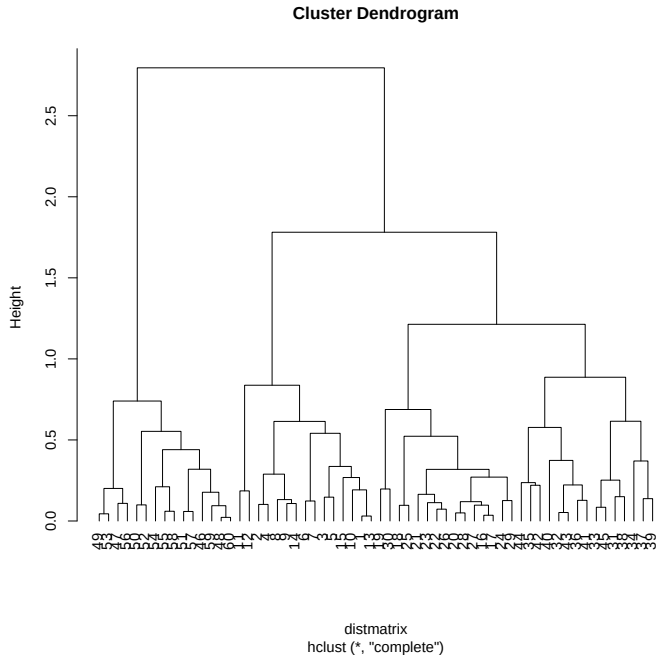
Το αντικείμενο `order` περιέχει σε σειρά την ταξινόμηση των στοιχείων στις συστάδες. Η επιλογή απόστασης `single` δημιουργεί 4 συστάδες οι οποίες δεν αντιπροσωπεύουν τα πραγματικά δεδομένα, ενώ οι υπόλοιπες τεχνικές ομαδοποιούν ορθά τα δεδομένα, όπως φαίνεται και από τα ακόλουθα γραφήματα.



Σχήμα 21.2: Αποτελέσματα ομαδοποίησης από την μέθοδο “Single Linkage”



Σχήμα 21.3: Αποτελέσματα ομαδοποίησης από την μέθοδο “Average Linkage”



Σχήμα 21.4: Αποτελέσματα ομαδοποίησης από την μέθοδο “Complete Linkage”

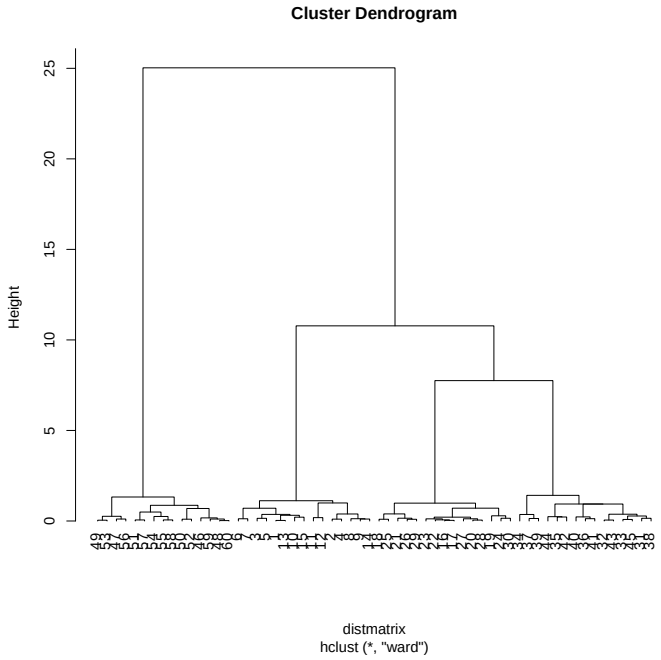
Μέθοδος Διαιρετότητας (Divisive Analysis Clustering-DIANA)

Περιγραφή

diana(distmatrix)

1. Αρχικά επιλέγουμε μια συστάδα.
2. Επιλέγουμε μετά το στοιχείο με τη μεγαλύτερη μέση απόσταση από τα υπόλοιπα στοιχεία της συστάδας, το οποίο γίνεται μια νέα συστάδα.
3. Κατανέμουμε τα στοιχεία της συστάδας είτε στην παλιά συστάδα είτε στην νέα, βάση της απόστασης του κάθε στοιχείου απο τις συστάδες.
4. Επιλέγουμε τη συστάδα με τη μεγαλύτερη διάμετρο (μεγαλύτερη απόσταση μεταξύ δυο στοιχείων της συστάδας) και επιστρέφουμε στο βήμα 2 μέχρι να έχουμε τόσες συστάδες όσα τα στοιχεία μας.

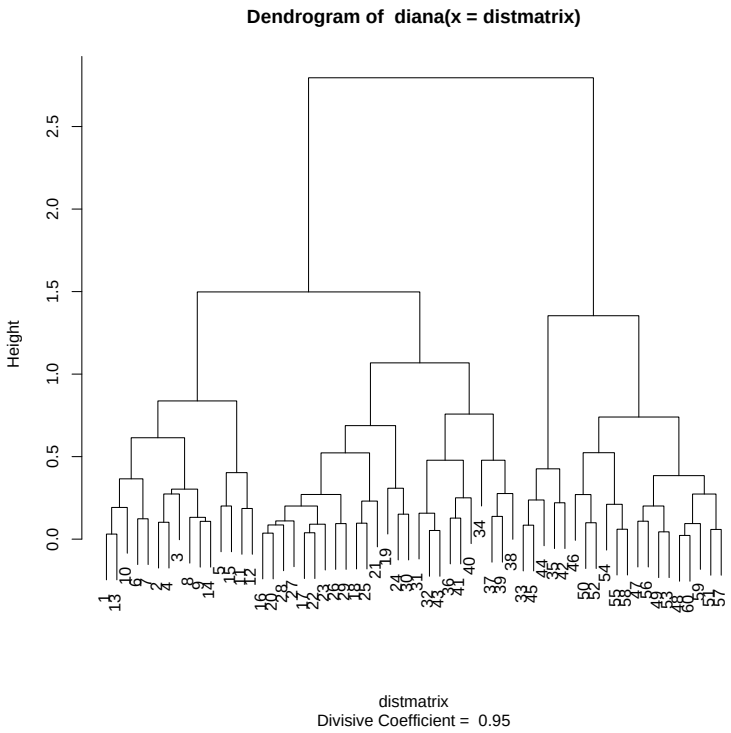
Εφαρμογή



Σχήμα 21.5: Αποτελέσματα ομαδοποίησης από την μέθοδο "Ward's Hierarchical Clustering"

```
> dv<-diana(distmatrix)
> plot(dv)
> dv1 <- cutree(as.hclust(dv), k = 4)
> dv1
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 2 2 2
[39] 2 2 2 3 2 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
> c(sum(dv1==1),sum(dv1==2),sum(dv1==3),sum(dv1==4))
[1] 15 25  5 15
```

Παρατηρούμε ότι η μέθοδος αυτή δεν ομαδοποιεί ορθά τα δεδομένα αφού 10 στοιχεία που θα έπρεπε να είχαν ταξινομηθεί στην τρίτη συστάδα, έχουν τοποθετηθεί στη δεύτερη συστάδα.



Σχήμα 21.6: Αποτελέσματα ομαδοποίησης από την μέθοδο "DIANA"

21.3 Μεθοδολογία K-means (MacQueen)

Περιγραφή

kmeans(dataset, nclusters, algorithm="MacQueen")

1. Επιλέγουμε τυχαία K στοιχεία τα οποία θα αποτελέσουν τους αρχικούς πυρήνες των συστάδων.

2. Για κάθε στοιχείο στα δεδομένα, κατανέμουμε το στοιχείο στην συστάδα της οποίας ο πυρήνας είναι πιο κοντά στο στοιχείο. Οι νέοι πυρήνες (centroids) για τις συστάδες υπολογίζονται ως ο μέσος όρος των στοιχείων της κάθε συστάδας.

3. Επαναλαμβάνουμε το βήμα 2 μέχρι να μην γίνουν αλλαγές στις συστάδες (ή μέχρι ενός ορισμένου αριθμού επαναλήψεων)

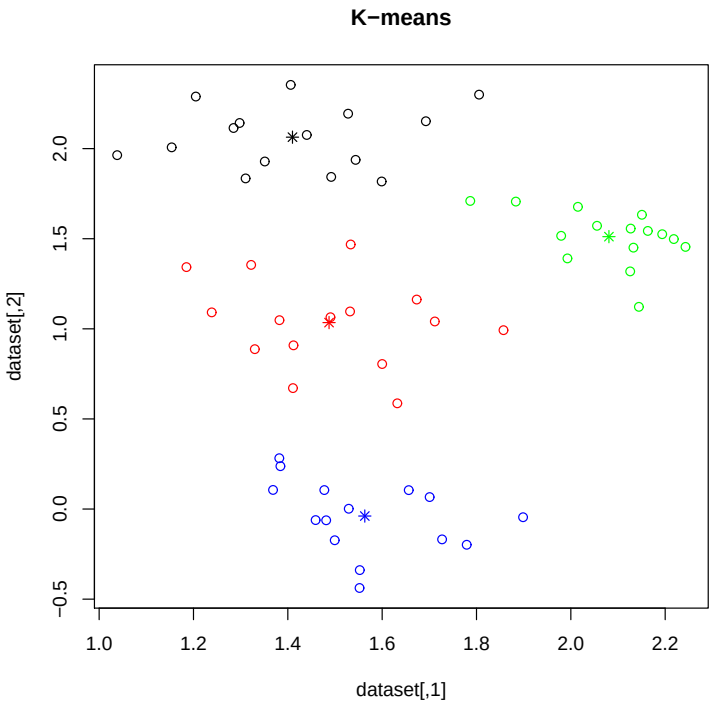
Εφαρμογή

Εφαρμόζουμε την μεθοδολογία K-means επιλέγοντας τον αλγόριθμο `MacQueen`, ο οποίος είναι αυτός που χρησιμοποιείται πιο συχνά για υλοποίηση αυτής της τεχνικής και περιγράφεται πιο πάνω. Χωρίζουμε τα δεδομένα σε 4 συστάδες.

```
>(cl <- kmeans(dataset, 4, algorithm="MacQueen"))
K-means clustering with 4 clusters of sizes 15, 15, 15, 15

Cluster means:
      [,1]        [,2]
1 1.487378  1.03460243
2 1.409862  2.06348506
3 2.080743  1.51151186
4 1.563167 -0.03878101

Clustering vector:
 [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1
[39] 1 1 1 1 1 1 1 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4

Within cluster sum of squares by cluster:

[1] 1.3218168 1.0320610 0.5656294 0.9120689
```

```
Available components: [1] "cluster"  "centers"  "withinss" "size"

> plot(dataset, col = mycol[cl$cluster], main="K-means")
> points(cl$centers, col = mycol, pch = 8)
```

Παρατηρούμε ότι η μέθοδος ομαδοποιεί ορθά τα δεδομένα σε συστάδες μεγέθους 15 στοιχείων η κάθε μία. Το `Clustering vector` δίνει την συστάδα που ανήκει το κάθε στοιχείο. Τα στοιχεία δημιουργήθηκαν με τέτοιο τρόπο έτσι ώστε τα στοιχεία της κάθε συστάδας να είναι διαδοχικά. Επίσης οι πυρήνες της κάθε συστάδας (`Cluster means`) είναι πολύ κοντά στους πυρήνες που χρησιμοποιήθηκαν για να δημιουργήσουμε στοιχεία για τις 4 συστάδες.



Σχήμα 21.7: Αποτελέσματα ομαδοποίησης από την μέθοδο "K-Means"

21.4 Partitioning Around Medoids (PAM)

Περιγραφή

pam(distmatrix, nclusters)

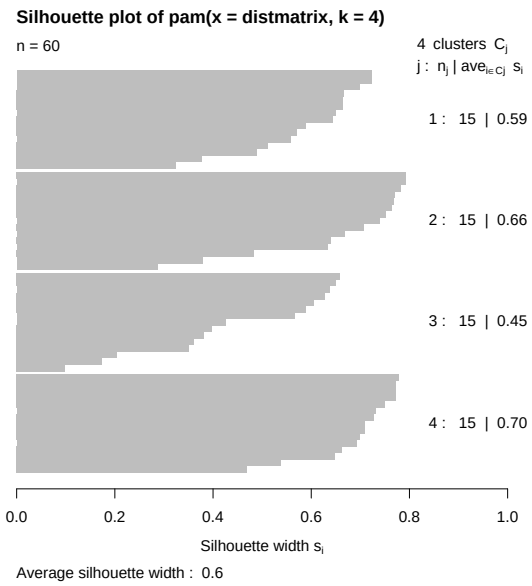
Η μέθοδος Partitioning Around Medoids διαφέρει από την μέθοδο kmeans στο σημείο ότι ως πυρήνας μιας συστάδας είναι πάντα ένα στοιχείο της συστάδας (medoid) και επιδιώκεται η ελαχιστοποίηση της απόστασης των υπόλοιπων στοιχείων από τον πυρήνα. Αρχικά επιλέγεται ένα καλό αρχικό σύνολο από medoids (build phase). Ακολούθως ελέγχεται κατά πόσο η εναλλαγή ενός στοιχείου με ένα medoid θα ελαχιστοποιήσει την απόσταση μεταξύ του πυρήνα και των άλλων στοιχείων και αν ναι, πραγματοποιείται (swap phase).

Εφαρμογή

```
> pamx<-pam(distmatrix,4)
> pamx
Medoids:
     ID
[1,]  3  3
[2,] 22 22
[3,] 32 32
[4,] 59 59
Clustering vector:
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3
[39] 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
Objective function:
    build      swap
0.2418506 0.2237345


Available components:
[1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
[6] "clusinfo"   "silinfo"    "diss" "call"
> plot(pamx)
```

Παρατηρούμε ότι τα στοιχεία που υποδεικνύει η μέθοδος ως πυρήνες (Medoids) είναι τα στοιχεία 3, 22, 32 και 59 που πραγματικά ανήκουν σε διαφορετικές συστάδες και όλα τα υπόλοιπα στοιχεία ταξινομούνται στην πραγματική τους συστάδα. Επίσης από το Silhouette plot παρατηρούμε ότι δεν υπάρχουν στοιχεία με

αρνητική τιμή, η οποία θα φανέρωνε ότι το αντίστοιχο στοιχείο έχει τοποθετηθεί σε λάθος συστάδα. Τιμές κοντά στο 1 φανερώνουν πολυ καλή ομαδοποίηση ενώ τιμές κοντά στο 0 ότι το στοιχείο βρίσκεται μεταξύ συστάδων.



Σχήμα 21.8: Αποτελέσματα ομαδοποίησης από την μέθοδο "Partitioning Around Medoids"

21.5 Self Organizing Maps (SOM)

Περιγραφή

Η μέθοδος SOM μπορεί να θεωρηθεί ως μια παραλλαγή της μεθόδου kmeans, η οποία περιορίζει τοπολογικά τους πυρήνες των συστάδων. Κάθε μονάδα αντιστοιχεί σε μια συστάδα και ο αριθμός των συστάδων καθορίζεται από το μέγεθος του πλέγματος (ορθογώνιου ή εξαγωνικού σχήματος) πάνω στο οποίο βρίσκονται οι συστάδες.

Αρχικά αναθέτουμε ένα διάνυσμα (codebook vector) σε κάθε μονάδα, το οποίο θα έχει το ρόλο ενός τυπικού μοτίβου συσχετισμένου με τη συγκεκριμένη μονάδα. Συνήθως ένα υποσύνολο των στοιχείων (training set) κατανέμονται τυχαία στις μονάδες. Κατά τη διάρκεια της εκπαίδευσης του αλγορίθμου, τα στοιχεία αυτά παρουσιάζονται επανηλειμμένα, σε τυχαία σειρά, στον τοπολογικό χάρτη. Η μονάδα, η οποία είναι πιο όμοια (winning unit) με το στοιχείο που χρησιμοποιούμε σε κάποιο στάδιο της διαδικασίας, τροποποείται έτσι ώστε η απόσταση της να μειωθεί περαιτέρω από το συγκεκριμένο στοιχείο. Αυτό πραγματοποιείται

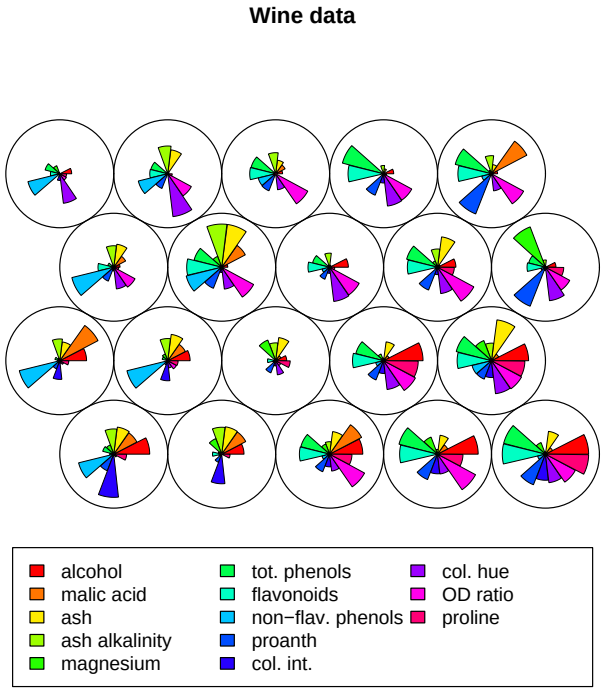
χρησιμοποιώντας σταθμισμένο μέσο όρο, με την βαρύτητα του στοιχείου (learning rate) να είναι μια από τις παραμέτρους της μεθόδου SOM. Συνήθως έχει μικρή τιμή (κοντά στο 0.5). Κατά τη διάρκεια της διαδικασίας, η τιμή αυτή μειώνεται έτσι ώστε ο τοπολογικός χάρτης να συγκλίνει.
Ο περιορισμός τοπολογικά προκύπτει από την απαίτηση του αλγορίθμου γειτονικές μονάδες να έχουν όμοια codebook vectors. Αυτό επιτυγχάνεται τροποποιώντας και τις μονάδες που γειτνυάζουν με την winning unit με τον ίδιο τρόπο. Ο αριθμός των μονάδων που θεωρούνται γειτονικές ως προς την μονάδα αυτή, μειώνεται κατά την εκπαίδευση, έτσι ώστε μετά από ορισμένες επαναλήψεις να τροποποιείται μόνο η συγκεκριμένη μονάδα.

Εφαρμογή

Θα χρησιμοποιήσουμε τα δεδομένα `wines` από το πακέτο `kohonen`, το οποίο παρουσιάζεται στο [3]. Τα δεδομένα αυτά περιέχουν τα αποτελέσματα χημικής ανάλυσης για 177 κρασιά που παράγονται σε μια περιοχή στην Ιταλία και αφορούν 13 χαρακτηριστικά των κρασιών.

```
> library("kohonen")
Loading required package: class
> data("wines")
> wines.sc <- scale(wines)
> set.seed(7)
> wine.som <- som(data = wines.sc,grid=somgrid(5,4,"hexagonal"))
> plot(wine.som, main= "Wine data")
```

Τα αποτελέσματα του σχήματος 9 δείχνουν ότι υψηλά επίπεδα οινοπνεύματος βρίσκονται στα δείγματα κρασιού στην κάτω δεξιά πλευρά του σχήματος, ενώ υψηλή χρωματική συχνότητα βρίσκεται στην κάτω αριστερά πλευρά του γραφήματος.



Σχήμα 21.9: Αποτελέσματα ομαδοποίησης από την μέθοδο “Self Organizing Maps”

21.6 Fuzzy Analysis Clustering (Fanny)

Περιγραφή

fanny(distmatrix, nclusters)

Η μέθοδος ομαδοποίησης Fuzzy επιτρέπει σε κάθε στοιχείο να ανήκει σε περισσότερες απο μια συστάδες. Αυτό επιτυχάνεται υπολογίζοντας κάποια ποσοστά (memberships) για κάποιο στοιχείο για κάθε συστάδα τέτοια ώστε το άθροισμά τους να είναι ίσο με 1.

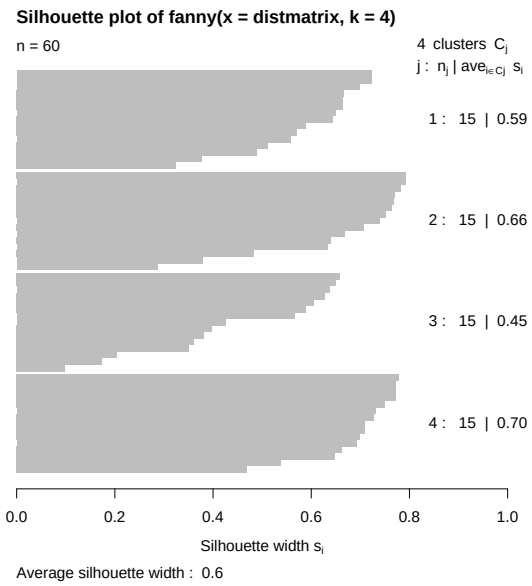
Εφαρμογή

```
> fuzzyc <- fanny(distmatrix,4)
> fuzzyc
Fuzzy Clustering object of class 'fanny' :
m.ship.expon.        2
objective     5.834809
tolerance        1e-15
iterations          13
converged            1
maxit              500
n                   60
Membership coefficients (in %, rounded):
      [,1] [,2] [,3] [,4]
 [1,]   79    9    8    4
 [2,]   66   14   15    6
 [3,]   80    9    7    3
 [4,]   76   10   10    4
 [5,]   71   13   10    5
 [6,]   59   15   18    8
 [7,]   69   12   13    6
 [8,]   54   23   17    6
 [9,]   64   16   14    5
[10,]   66   14   13    7
[11,]   59   21   14    6
[12,]   50   25   16    8
[13,]   79    9    8    4
[14,]   69   15   12    5
```

```
[15,]   66   15   13    6
[16,]    7   83    7    3
.
.
.
[43,]    8   11   74    7
[44,]   11   14   52   23
[45,]   10   11   67   12
[46,]    4    5    9   82
[47,]    6    8   14   72
[48,]    3    4    7   86
[49,]    8   10   21   61
[50,]    8   10   15   66
[51,]    5    7   12   77
[52,]    7    8   13   72
[53,]    9   11   23   57
[54,]    8   11   16   65
[55,]    6    8   13   73
[56,]    5    6   11   79
[57,]    5    7   11   77
[58,]    5    7   11   77
[59,]    3    4    7   87
[60,]    4    4    7   85
Fuzzyness coefficients:
dunn_coeff normalized
 0.5181044  0.3574725
Closest hard clustering:
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3
[39] 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4

Available components:
 [1] "membership"  "coeff"       "memb.exp"    "clustering"  "k.crisp"
 [6] "objective"   "convergence" "diss"        "call"        "silinfo"
> plot(fuzzyc)
```

Παρατηρούμε ότι τα `Membership coefficients` είναι αρκετά μεγάλα για την συστάδα στην οποία πραγματικά ανήκουν τα στοιχεία και χαμηλά για τις υπόλοιπες.

Όσο πιο κοντά βρίσκεται ο συντελεστής Dunn(`dunn_coeff`) στο 1 τόσο πιο ξεκάθαρη είναι η ομαδοποίηση των στοιχείων. Από το Silhouette plot παρατηρούμε ότι δεν υπάρχουν στοιχεία με αρνητική τιμή και όλα τα στοιχεία έχουν ταξινομηθεί ορθά.

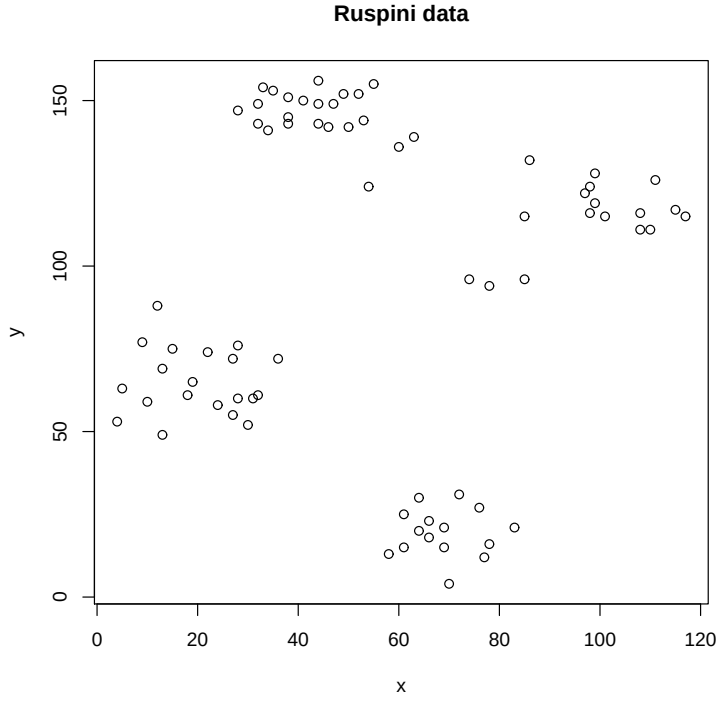


Σχήμα 21.10: Αποτελέσματα ομαδοποίησης από την μέθοδο “Fuzzy Analysis”

21.7 Παράδειγμα ανάλυσης δεδομένων

Θα χρησιμοποιήσουμε το σύνολο δεδομένων `Ruspini` από το πακέτο `cluster`. Τα δεδομένα αυτά είναι χρήσιμα για δοκιμή μεθόδων ανάλυσης κατά συστάδες και περιλαμβάνουν 75 σημεία στον $\mathbb{R}^2$, τα οποία χωρίζονται σε 4 ομάδες.

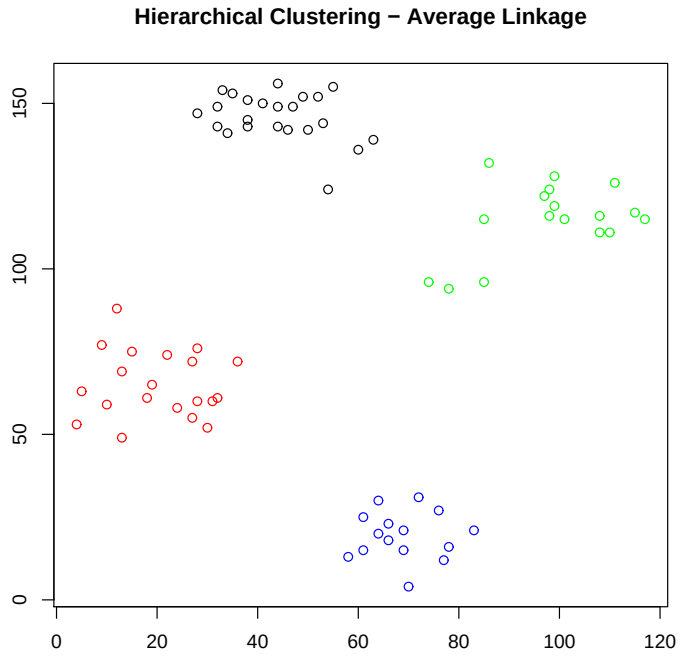
```
> library(cluster)
> data(ruspini)
> plot(ruspini,main="Ruspini data")
> distmatrix <- dist(ruspini)
```



Σχήμα 21.11: Σύνολο δεδομένων Ruspini

```
> mycol <- c("red", "black", "green", "blue")
> hca<-hclust(distmatrix,method="average")
> memba<- cutree(hca,k=4)
> c(sum(memba==1),sum(memba==2),sum(memba==3),sum(memba==4))
[1] 20 23 17 15
> dataset<- cbind(ruspini$x,ruspini$y,memba)
> plot(dataset,col=mycol[memba],main="Hierarchical Clustering - Average Linkage")
```

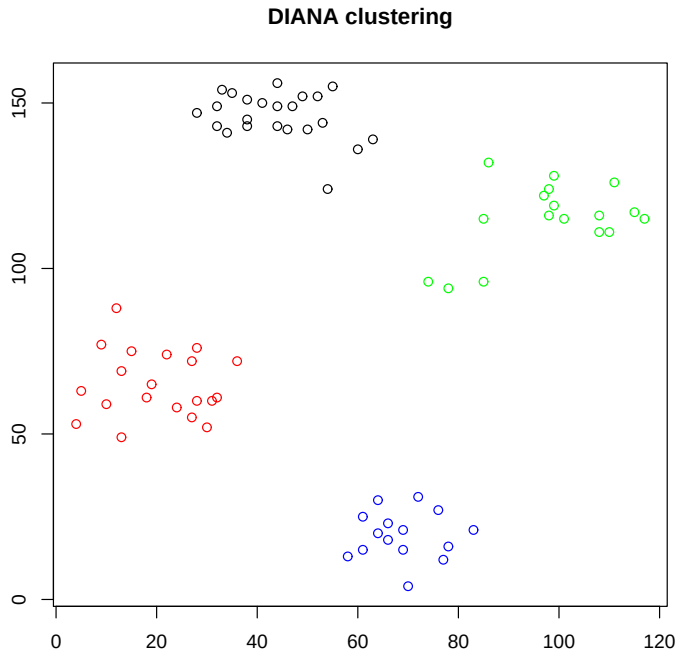
Αρχικά εφαρμόζουμε ιεραρχική ανάλυση κατά συστάδες με average linkage και δημιουργούμε το γράφημα των δεδομένων χρησιμοποιώντας διαφορετικό χρώμα για κάθε συστάδα. Παρατηρούμε ότι η τεχνική εντοπίζει ορθά τις 4 συστάδες.



Σχήμα 21.12: Αποτελέσματα ομαδοποίησης από την μέθοδο “Average Linkage”

```
> dv<-diana(distmatrix)
> dc <- cutree(as.hclust(dv), k = 4)
> dataset<- cbind(ruspini$x,ruspini$y,dc)
> plot(dataset,col=mycol[dc],main="DIANA clustering")
```

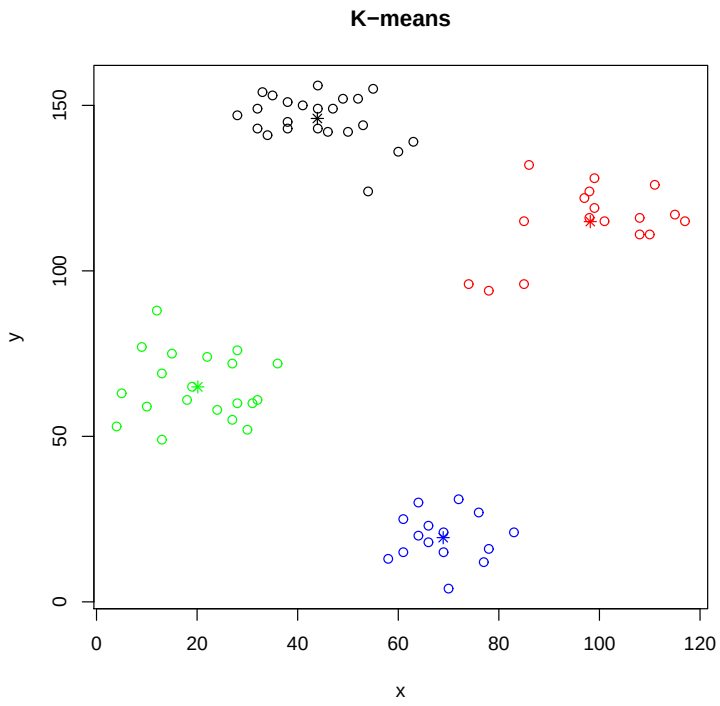
Στη συνέχεια εφαρμόζουμε τη μέθοδο DIANA. Δημιουργούμε το γράφημα των δεδομένων και παρατηρούμε ότι η ομαδοποίηση γίνεται ορθά.



Σχήμα 21.13: Αποτελέσματα ομαδοποίησης από την μέθοδο "DIANA"

```
> cl <- kmeans(ruspini, 4, algorithm="MacQueen")
> plot(ruspini, col = mycol[cl$cluster], main="K-means")
> points(cl$centers, col = mycol, pch = 8)
```

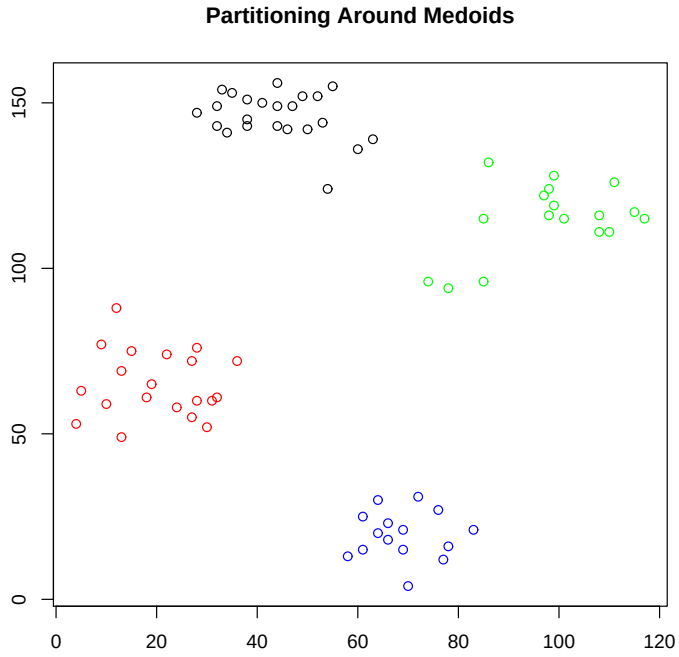
Ακολούθως εφαρμόζουμε τη μέθοδο K-means. Στο γράφημα των δεδομένων βλέπουμε ότι οι συστάδες έχουν επιλεχθεί σωστά, όπως και οι πυρήνες των συστάδων.



Σχήμα 21.14: Αποτελέσματα ομαδοποίησης από την μέθοδο "K-Means"

```
> pamx<-pam(distmatrix,4)
> clusters<-pamx$clustering
> dataset<- cbind(ruspini$x,ruspini$y,clusters)
> plot(dataset,col=mycol[clusters],main="Partitioning Around Medoids")
```

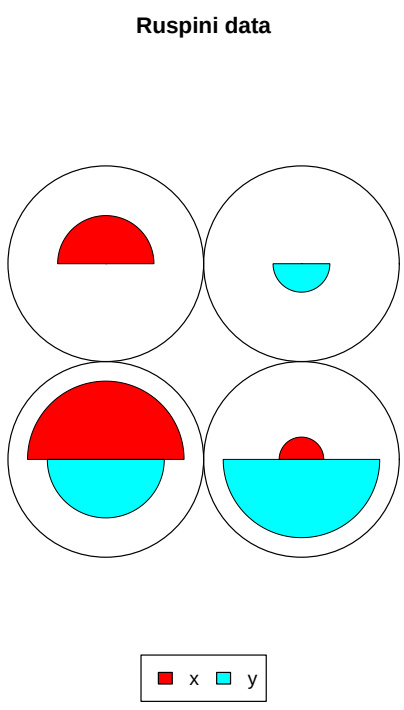
Συνεχίζουμε χρησιμοποιώντας τη μέθοδο Partitioning Around Medoids. Παρατηρούμε ότι και αυτή η τεχνική εντοπίζει ορθά τις συστάδες.



Σχήμα 21.15: Αποτελέσματα ομαδοποίησης από την μέθοδο "Partitioning Around Medoids"

```
> library("kohonen")
> ruspini.sc <- scale(ruspini)
> set.seed(7)
> ruspini.som <- som(data = ruspini.sc,grid=somgrid(2,2,"rectangular"))
> plot(ruspini.som, main= "Ruspini data")
```

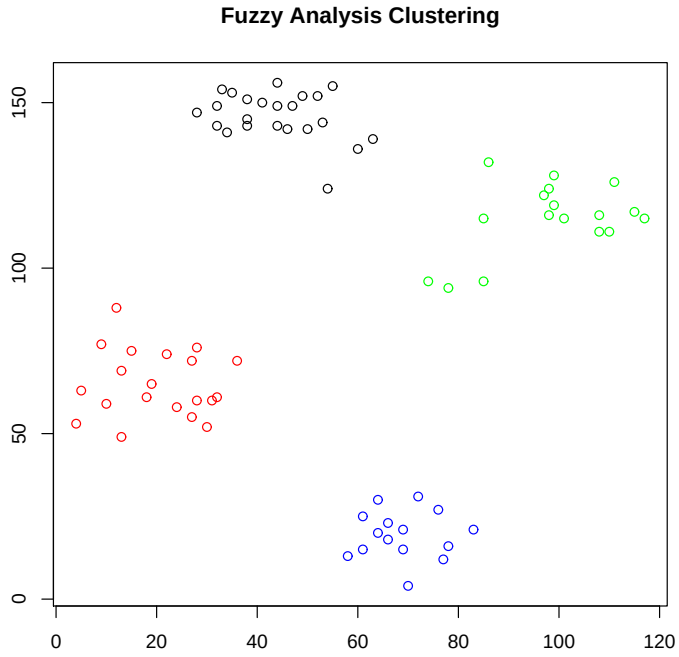
Εφαρμόζουμε επίσης τη μέθοδο "Self Organizing Maps". Τυποποιούμε τα δεδομένα και χρησιμοποιούμε ένα πλέγμα 2×2. Παρατηρούμε ότι η μέθοδος εντοπίζει τα χαρακτηριστικά των 4 συστάδων, δηλαδή ότι η πρώτη συστάδα έχει x κοντά στο 0 και μικρό y, η δεύτερη συστάδα έχει y κοντά στο 0 και μικρό x, η τρίτη συστάδα έχει μεγάλο x και μέτριο y ενώ η τέταρτη συστάδα έχει μικρό x και μεγάλο y.



Σχήμα 21.16: Αποτελέσματα ομαδοποίησης από την μέθοδο "Self Organizing Maps"

```
> fuzzyc <- fanny(distmatrix,4)
> clusters<-fuzzyc$clustering
> dataset<- cbind(ruspini$x,ruspini$y,clusters)
> plot(dataset,col=mycol[clusters],main="Fuzzy Analysis Clustering")
```

Χρησιμοποιούμε τέλος τη μέθοδο ομαδοποίησης “Fuzzy”. Βλέπουμε ότι η επιλογή των συστάδων είναι ορθή.



Σχήμα 21.17: Αποτελέσματα ομαδοποίησης από την μέθοδο “Fuzzy Analysis”

Βιβλιογραφία

1. Trevor Hastie, Robert Tibshirani, Jerome Friedman (2001), The Elements of Statistical Learning, Data Mining,Inference and Prediction, Springer Series in Statistics.

2. Richard A. Johnson, Dean W. Wichern (1998), Applied Multivariate Statistical Analysis, Prentice Hall.

3. Ron Wehrens, Lutgarde M. C. Buydens (October 2007), Self- and Super-organizing Maps in R: The kohonen Package,Journal of Statistical Software, Volume 21, Issue 5.

Κεφάλαιο 22

Ανάλυση Χρονοσειρών

22.1 Ανάλυση Χρονοσειρών

Με τον όρο Χρονοσειρά εννοούμε μια σειρά από παρατηρήσεις που παίρνονται σε ορισμένες χρονικές στιγμές ή περιόδους που ισαπέχουν μεταξύ τους. Υπάρχουν ένα μεγάλο εύρος στατιστικών μεθόδων για την ανάλυση χρονοσειρών. Γενικά οι μέθοδοι αυτοί ανήκουν σε δύο κατηγορίες: αυτές που βασίζονται στη μελέτη συναρτήσεων που εξαρτώνται από τον χρόνο, και σε αυτές οι οποίες εξαρτώνται από τις συχνότητες και οι οποίες ερευνούν τις περιοδικές ιδιότητες που μπορεί να έχει η σειρά.

Τα τρία κυριότερα στοιχεία της ανάλυσης χρονοσειρών είναι η περιγραφή, η επεξήγηση και η πρόβλεψη των εξαρτημένων δεδομένων. Η περιγραφή επιτυγχάνεται με τη βοήθεια διαφόρων γραφημάτων, η επεξήγηση χρησιμοποιώντας κάποιας μορφής μοντέλα για να εξερευνηθούν οι μηχανισμοί δημιουργίας της χρονοσειράς, και η πρόβλεψη περιλαμβάνει τη χρησιμοποίηση ενός μοντέλου για να προβλεφθούν μελλοντικές τιμές της σειράς. Στο παράδειγμα το οποίο θα χρησιμοποιηθεί στο κεφάλαιο αυτό η ανάλυση θα επικεντρωθεί στο πεδίο του χρόνου.

Η **συνάρτηση αυτοδιακύμανσης** (autocovariance function) είναι το βασικό εργαλείο για την περιγραφή της σειριακής εξάρτησης μιας μονομεταβλητής, στάσιμης (χωρίς περιοδικότητα) χρονοσειράς και ορίζεται από

$$\gamma(k) = E(X_t - \mu)(X_{t+k} - \mu), k = 0, \pm 1, \pm 2, \ldots$$

όπου X_t, $t = 0, \pm 1, \pm 2, \ldots$ οι τιμές της σειράς, μ η μέση τιμή της και k η υστέρηση (lag) για την οποία υπολογίζεται η αυτοδιακύμανση. Η **συνάρτηση αυτοσυσχέτισης** (autocorrelation function) είναι η κανονικοποιημένη μορφή της συνάρτηση

αυτοδιακύμανσης και ορίζεται ως

$$\varrho(k) = \frac{\gamma(k)}{\gamma(0)} = \frac{\gamma(k)}{\sigma_x^2}$$

όπου σ_x^2 η διακύμανση της χρονοσειράς.
Η συνάρτηση αυτοδιακύμανσης μπορεί να εκτιμηθεί από

$$\widehat{\gamma(k)} = \frac{1}{n}\sum_{i=1}^{n-k}(x_i - \overline{x})(x_{i+k} - \overline{x}).$$

Συνεπώς, η συνάρτηση αυτοσυσχέτισης μπορεί να εκτιμηθεί από

$$\widehat{\varrho(k)} = \frac{\widehat{\gamma(k)}}{\widehat{\gamma(0)}}$$

Για να διερευνηθούν τυχών μηχανισμοί δημιουργίας των δεδομένων, αλλά και να υποδειχθούν κατάλληλα μοντέλα, χρησοποιούνται οι γραφικές παραστάσεις της αυτοδιακύμανσης ή της αυτοσυσχέτισης συναρτήσει της υστέρησης k. Τέτοια μοντέλα ανάλυσης χρονοσειρών είναι της μορφής

$$X_t = \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \ldots + \alpha_p X_{t-p} + \epsilon_t$$

όπου ϵ_t το σφάλμα. Το πιο πάνω μοντέλο είναι γνωστό ως **διαδικασία αυτοπαλινδρόμισης βαθμού** p.

22.2 Παράδειγμα

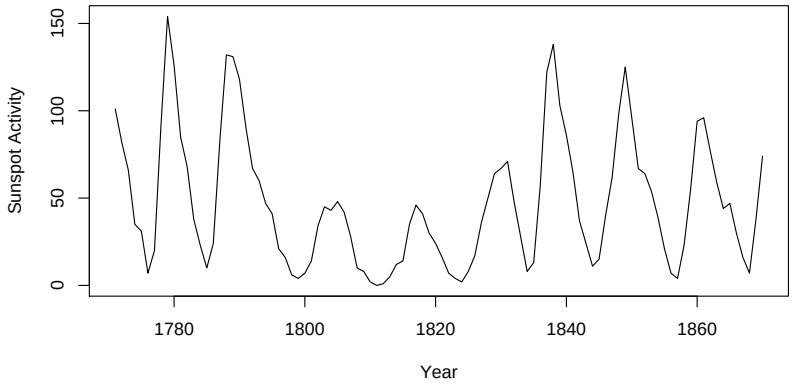
Τα δεδομένα που θα χρησιμοποιηθούν παρουσιάζουν τον αριθμό των ηλιακών κηλίδων από το 1771 ως το 1870. Η εντολή `ts` χρησιμοποιείται για να δημιουργήσει ένα αντικείμενο χρονοσειράς.

```
> sun.ts<-ts(round(sunspot.year[71:171]),start=1771,end=1870)
> sun.ts
Time Series:
Start = 1771
End = 1870
Frequency = 1
  [1] 101  82  66  35  31   7  20  92 154 126  85  68  38  23  10  24  83 132
 [19] 131 118  90  67  60  47  41  21  16   6   4   7  14  34  45  43  48  42
 [37]  28  10   8   2   0   1   5  12  14  35  46  41  30  24  16   7   4   2
```

```
 [55]   8  17  36  50  64  67  71  48  28   8  13  57 122 138 103  86  65  37
 [73]  24  11  15  40  62  98 125  96  67  64  54  39  21   7   4  23  55  94
 [91]  96  77  59  44  47  30  16   7  38  74
```

Το πρώτο βήμα στην ανάλυση χρονοσειρών είναι να γίνει η περιγραφή της χρονοσειράς χρησιμοποιώντας το γράφημα των τιμών της συναρτήσει του χρόνου. Το γράφημα αυτό αποκαλύπτει τα κυριότερα στοιχεία της σειράς όπως είναι η τάση και η περιοδικότητα. Το γράφημα κατασκευάζεται με την εντολή `ts.plot` και παρουσιάζεται στο Σχήμα 22.1.

```
> ts.plot(sun.ts,xlab="Year",ylab="Sunspot Activity")
```



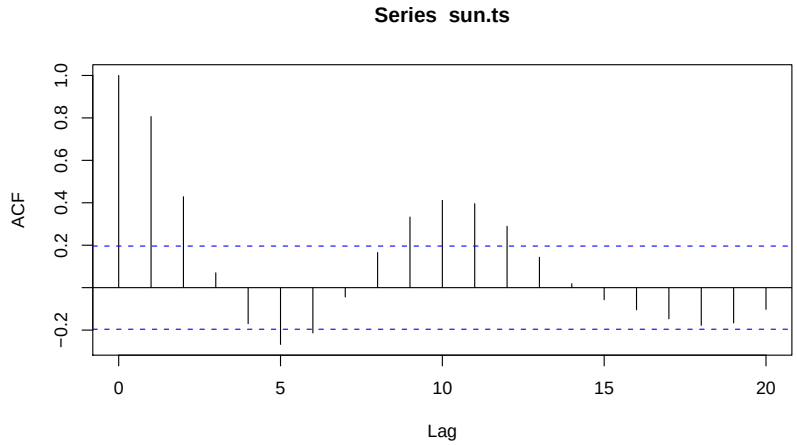
Σχήμα 22.1: Γράφημα της χρονοσειράς του αριθμού των ηλιακών κηλίδων από το 1771 ως το 1870.

Το πιο πάνω γράφημα δείχνει ότι υπάρχει μια περιοδικότητα στα δεδομένα αλλά δεν φαίνεται να υπάρχει κάποια τάση. Στη συνέχεια υπολογίζονται οι αυτοσυσχετίσεις και κατασκευάζεται το γράφημά τους συναρτήσει της υστέρησης (Σχήμα 22.2). Η συνάρτηση στην R για κατασκευή του γραφήματος της συνάρτησης αυτοσυσχέτισης είναι η `acf`.

```
> acf(sun.ts,lag.max=20,type="correlation")
```

Το όρισμα `lag.max` ορίζει τον μέγιστο αριθμό της υστέρησης για την οποία θα γίνει η γραφική παράσταση της συνάρτησης αυτοσυσχέτισης, ενώ το όρισμα `type`

αν θα χρησιμοποιηθεί η αυτοδιακύμανση ή η αυτοσυσχέτιση. Εξ ορισμού το type είναι η αυτοσυσχέτιση και για αυτό στο παράδειγμά μας δεν ήταν αναγκαίο να οριστεί. Στο γράφημα οι δύο οριζόντιες διακεκομμένες γραμμές δείχνουν το 95% διάστημα εμπιστοσύνης για τον έλεγχο $H_o : \varrho = 0$. Είναι φανερό ότι η αυτοσυσχέτιση έχει μια ημιτονοειδή μορφή και αυτό υποδεικνύει ότι την περιοδικότητα που παρουσιάστηκε και στο γράφημα του Σχήματος 22.1. Επίσης, φαίνεται ότι αυτή η περιοδικότητα είναι ίση με 10 με 11 χρόνια, ένα στοιχείο που είναι γενικά γνωστό για την δραστηριότητα μιας ηλιακής κηλίδας.



Σχήμα 22.2: Συνάρτηση Αυτοσυσχέτισης του αριθμού των ηλιακών κηλίδων από το 1771 ως το 1870.

Πριν να γίνει προσπάθεια μοντελλοποίησης της χρονοσειράς, πρέπει να γίνει προσπάθεια να εξαληφθεί η περιοδικότητα. Ένας απλός τρόπος να επιτευχθεί αυτό είναι να θεωρηθούν οι κατάλληλες διαφορές της σειράς. Συνεπώς, υποθέτωντας περιοδικότητα 11 χρόνων, μπορούν να θεωρηθούν οι διαφορές μεταξύ των σημείων ανά 11 χρόνια. Δηλαδή, ορίζουμε μια νέα χρονοσειρά

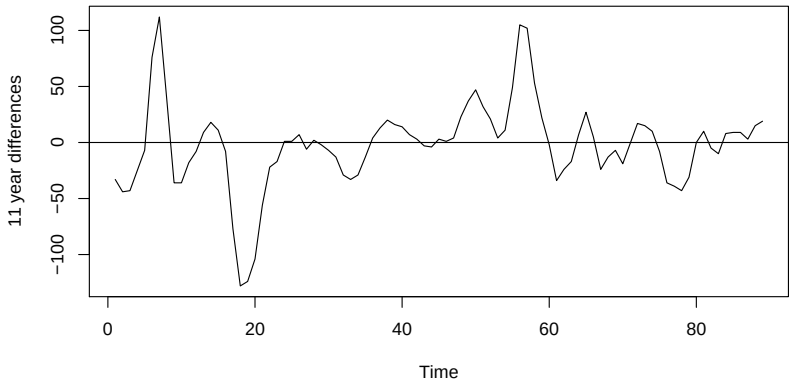
$$y_t = x_t - x_{t-11}.$$

Στην R αυτό γίνεται όπως πιο κάτω:

```
> newsun<-sun.ts[12:100]-sun.ts[1:89]
> ts.plot(newsun,xlab="Time",ylab="11 year differences")
> abline(h=0)
> acf(newsun,lag.max=20)
```

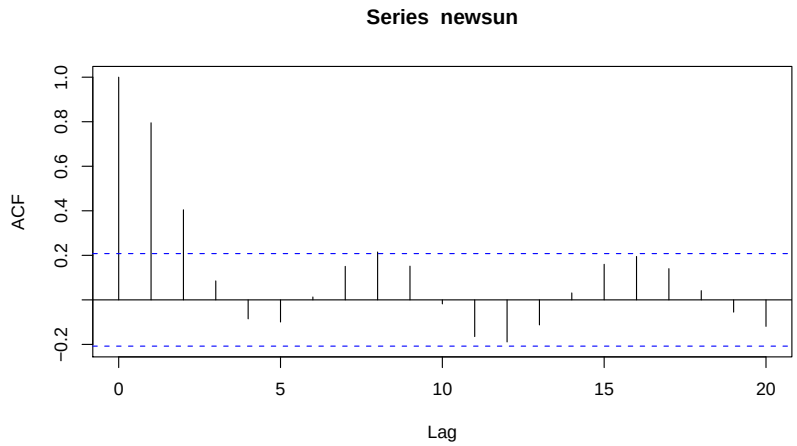
Το γράφημα της νέας χρονοσειράς (Σχήμα 22.3), από την 11-χρονη διαφορά των τιμών της αρχικής, ελάχιστο στοιχείο περιοδικότητας. Επίσης, η συνάρτηση αυτοσυσχέτισης στο Σχήμα 22.4 δείχνει ότι η ημιτονοειδής μορφή δεν είναι σε τόσο μεγάλο βαθμό όπως πριν, αλλά η αυτοσυσχέτιση περιορίστηκε σε υστέρηση ίση με 2.

Συνεχίζοντας, μπορεί να εφαρμοστεί στη νέα χρονοσειρά των 11-χρονων διαφορών ένα μοντέλο αυτοπαλινδρόμισης και αυτό στην R επιτυγχάνεται με τη συνάρτηση `ar.yw`. Σημαντικό στοιχείο της συνάρτησης αυτής είναι ότι μπορεί να εκτιμηθεί η τάξη του μοντέλου που χρειάζεται η σειρά με την επιλογή της τιμής της υστέρησης k η οποία ελαχιστοποιεί το κριτήριο AIC.



Σχήμα 22.3: Γράφημα της 11-χρονης διαφοράς του αριθμού των ηλιακών κηλίδων από το 1771 ως το 1870.

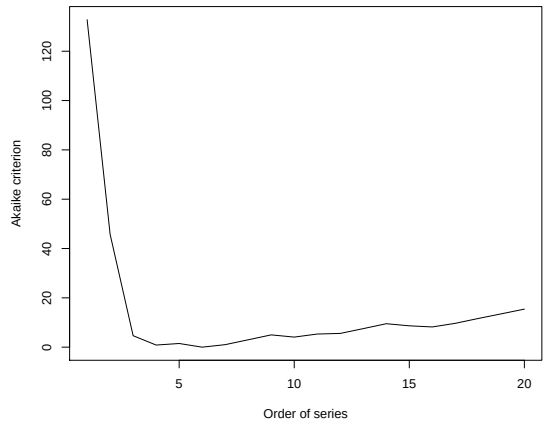
```
> sun.ar<-ar.yw(newsun)
> sun.ar

Call: ar.yw.default(x = newsun)

Coefficients:
      1        2        3        4        5
 1.4985  -1.1434   0.6369  -0.4126   0.1961

Order selected 5  sigma^2 estimated as  313.4
```



Σχήμα 22.4: Συνάρτηση Αυτοσυσχέτισης της 11-χρονης διαφοράς του αριθμού των ηλιακών κηλίδων από το 1771 ως το 1870.

Το αποτέλεσμα του μοντέλου παρουσιάζει τους εκτιμώμενους συντελεστές του όπως και την τάξη του. Επίσης δίνεται και η εκτίμηση της διασποράς του σφάλματος του μοντέλου. Συνεπώς, η τάξη του μοντέλου αυτοπολινδρόμισης που εφαρμόζει καλύτερα τα δεδομένα εκτιμήθηκε να είναι ίση με 5. Μπορεί να κατασκευαστεί το γράφημα του AIC όπως πιο κάτω και παρουσιάζεται στο Σχήμα 22.5. Στο γράφημα φαίνεται ότι το κριτήριο AIC ελαχιστοποιείται στην τιμή 6, αλλά αυτό εξηγείται από το γεγονός ότι το γράφημα ξεκινά από το σημείο 1, το οποίο βασικά αντιστοιχεί στην τιμή του AIC για το μοντέλο τάξης 0.

```
> ts.plot(sun.ar$aic,xlab="Order of series",ylab="Akaike criterion")
```

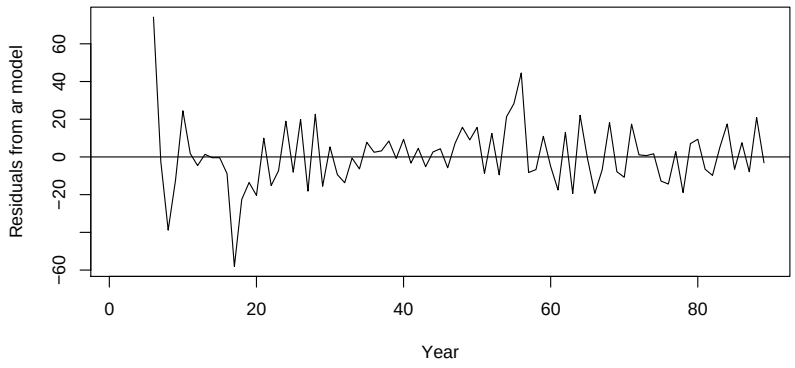


Σχήμα 22.5: Γράφημα του κριτηρίου AIC από την εφαρμογή μοντέλων αυτοπαλινδρόμισης στα δεδομένα.

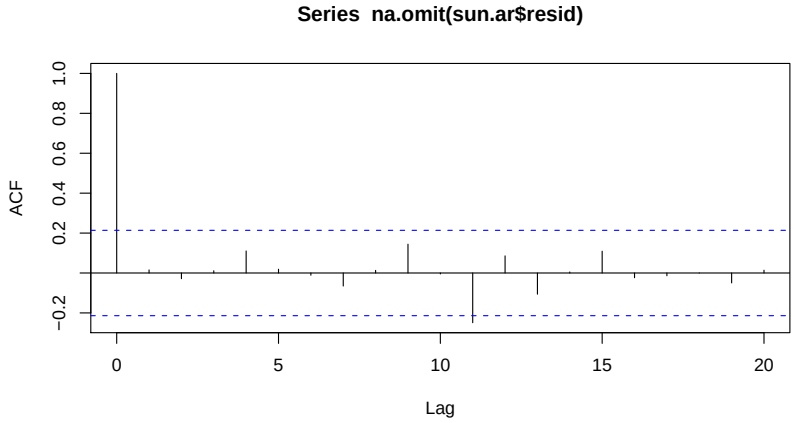
Τα γράφημα των υπολοίπων κατασκευάζονται με τις εντολές

```
> ts.plot(sun.ar$resid,xlab="Year",ylab="Residuals from ar model")
> abline(h=0)
> acf(na.omit(sun.ar$resid),lag.max=20)
```

Στην περίπτωση που το μοντέλο εφαρμόζει καλά στα δεδομένα, τότε τα υπόλοιπα θα είχαν τη μορφή τυχαίας σειράς στο Σχήμα 22.6, δηλαδή θα είχαν τη μορφή μιας χρονοσειράς λευκού θορύβου. Είναι φανερό πως τα υπόλοιπα δεν παρουσιάζουν μια προφανή δομή. Το κορελλόγραμμα των υπολοίπων βοηθάει επίσης στο να ελεγχθεί η τυχαιότητά τους. Σε μια σειρά λευκού θορύβου δεν παρουσιάζεται καμία ημιτονοειδής μορφή αλλά και η αυτοσυσχέτιση για όλες τις υστερήσεις είναι μεταξύ των ορίων του 95% διαστήματος εμπιστοσύνης. Αυτό δείχνει και το γράφημα της συνάρτησης αυτοσυσχέτισης στο Σχήμα 22.7.



Σχήμα 22.6: Γράφημα των υπολοίπων του μοντέλου αυτοπαλινδρόμισης.



Σχήμα 22.7: Συνάρτηση Αυτοσυσχέτισης των υπολοίπων του μοντέλου αυτοπαλινδρόμισης.

Κεφάλαιο 23

Παραδείγματα Μεθόδων Ε-Μ Αλγόριθμου

Οι μέθοδοι Ε-Μ αλγόριθμου μπορούν να επεξηγηθούν πιο εύκολα στην περίπτωση ενός τυχαίου δείγματος το οποίο αποτελείται από παρατηρηθείσες και μη παρατηρηθείσες ή εκλειπούσες τιμές.

Ένα απλό παράδειγμα δείγματος με εκλειπούσες τιμές προκύπτει στην περίπτωση ελέγχου του χρόνου επιβίωσης. Για παράδειγμα, ένας αριθμός ηλεκτρικών λαμπτήρων ανάβει συνεχώς και καταμετρείται ο χρόνος που χρειάζεται μέχρι να πάψουν να λειτουργούν. Σε ένα τέτοιο παράδειγμα, είναι συνήθες φαινόμενο το πείραμα να διακοπεί πριν να πάψουν να λειτουργούν όλοι οι λαμπτήρες. Ο χρόνος επιβίωσης των λαμπτήρων οι οποίοι συνεχίζουν να δουλεύουν δεν έχει παρατηρηθεί. Όμως, προφανώς ο αριθμός των λογοκριμένων παρατηρήσεων και ο χρόνος της λογοκρισίας περιέχουν πληροφορία για την κατανομή του χρόνου επιβίωσης.

Ακόμη ένα γνωστό παράδειγμα στο οποίο μπορεί να χρησιμοποιηθεί ο Ε-Μ αλγόριθμος είναι το πεπερασμένο μοντέλο μίξης κατανομών. Κάθε παρατήρηση προέρχεται από μία άγνωστη παρατήρηση ενός υποτιθέμενου συνόλου κατανομών. Οι εκλείπουσες τιμές προσδιορίζουν την κατανομή. Οι παράμετροι των κατανομών πρόκειται να εκτιμηθούν. Ένα παράπλευρο κέρδος της μεθόδου είναι ότι εκτιμάται σε ποια κατηγορία ανήκουν τα δεδομένα.

Τα ελλειπή δεδομένα μπορούν να είναι εκλείπουσες παρατηρήσεις της ίδιας τυχαίας μεταβλητής η οποία παράγει το δείγμα που παρατηρήθηκε, όπως στην περίπτωση του παραδείγματος λογοκρισίας, ή μπορούν να προέρχονται από μία διαφορετική τυχαία μεταβλητή η οποία σχετίζεται με κάποιο τρόπο με την τυχαία

μεταβλητή που έχει παρατηρηθεί.

Πολλές εφαρμογές της μεθόδου του Ε-Μ αλγόριθμου περιλαμβάνουν προβλήματα με ελλειπή δεδομένα, αλλά αυτό δεν είναι αναγκαίο. Συχνά, ο Ε-Μ αλγόριθμος μπορεί να εφαρμοστεί βασισμένος σε μία τεχνητή ¨ελλειπής¨ τυχαία μεταβλητή για να συμπληρώσει το δεδομένα που παρατηρήθηκαν.

23.1 Πρώτο Παράδειγμα: Πολυωνυμική Κατανομή

Ένα από τα πιο απλά παραδείγματα της μεθόδου Ε-Μ αλγορίθμου δόθηκε από τους Dempster, Laird και Rubin (1977). Έστω η πολυωνυμική κατανομή με τέσσερα πιθανά αποτελέσματα, η οποία έχει συνάρτηση πυκνότητας πιθανότητας,

$$p(x_1, x_2, x_3, x_4) = \frac{n!}{x_1!x_2!x_3!x_4!}{\pi_1}^{x_1}{\pi_2}^{x_2}{\pi_3}^{x_3}{\pi_4}^{x_4}$$

με $n = x_1 + x_2 + x_3 + x_4$ και $\pi_1 + \pi_2 + \pi_3 + \pi_4 = 1$. Έστω ότι όλες οι πιθανότητες συσχετίζονται με μία παράμετρο θ, ως εξής:

$$\pi_1 = \frac{1}{2} + \frac{1}{4}\theta$$

$$\pi_2 = \frac{1}{4} - \frac{1}{4}\theta$$

$$\pi_3 = \frac{1}{4} - \frac{1}{4}\theta$$

$$\pi_4 = \frac{1}{4}\theta$$

όπου $0 \leq \theta \leq 1$.

Δεδομένου μιας παρατήρησης (x_1, x_2, x_3, x_4), η λογαριθμική συνάρτηση πιθανοφάνειας δίνεται από

$$l(\theta) = x_1 \log(2 + \theta) + (x_2 + x_3) \log(1 - \theta) + x_4 \log(\theta).$$

Σκοπός είναι να εκτιμηθεί η παράμετρος θ. Η παράγωγος δίνεται από

$$\frac{d}{d\theta} l(\theta) = \frac{x_1}{2 + \theta} - \frac{x_2 + x_3}{1 - \theta} + \frac{x_4}{\theta}.$$

και σε αυτό το απλό παράδειγμα, η εκτιμήτρια μεγίστης πιθανοφάνειας για το θ μπορεί να υπολογιστεί επιλύοντας μία απλή πολυωνυμική εξίσωση.

Για να χρησιμοποιηθεί ο Ε-Μ αλγόριθμος για αυτό το παράδειγμα,μπορεί κάποιος να υποθέσει την πολυωνυμική πέμπτης τάξης, η οποία παράγεται χωρίζοντας την πρώτη τάξη της αρχικής πολυωνυμικής σε δύο με αντίστοιχες πιθανότητες $1/2$ και $\theta/4$. Η αρχική μεταβλητή x_1 είναι τώρα το άθροισμα της u_1 και της u_2.

Κάτω από αυτόν το μετασχηματισμό, η Ε.Μ.Π. του θ θεωρώντας το άθροισμα u_2+x4 (ή x_2+x_3) να είναι μία πραγμάτωση της διωνυμικής με $n = u_2+x_4+x_2+x_3$ και $\pi = \theta$ (ή $1-\theta$). Ωστόσο, το u_2 (ή το u_1) δεν είναι γνωστό. Συνεχίζοντας σαν να είχαμε παρατήρηση από μία πολυωνυμική με πέντε πιθανά αποτελέσματα, με δύο ελλείποντα στοιχεία, η λογαριθμική πιθανοφάνεια δίνεται από,

$$l_c(\theta) = (u_2 + x_4)\log(\theta) + (x_2 + x_3)\log(1-\theta),$$

και η εκτιμήτρια μεγίστης πιθανοφάνειας για το θ είναι ίση με

$$\frac{u_2 + x_4}{u_2 + x_2 + x_3 + x_4}.$$

Το Ε-βήμα του επαναληπτικού Ε-Μ αλγόριθμου συμπληρώνει την ελλείπουσα τιμή με την αναμενόμενη τιμή δεδομένου της τρέχουσας τιμής της παραμέτρου, $\theta^{(k)}$ και της τιμής που έχει παρατηρηθεί. Αυτή είναι μία διωνυμική τυχαία μεταβλητή σαν μέρος της x_1. Έτσι, με $\theta = \theta^{(k)}$,

$$E_{\theta^{(k)}}(u_2) = \frac{\frac{1}{4}x_1\theta^{(k)}}{\frac{1}{2} + \frac{1}{4}\theta^{(k)}} = u_2^{(k)}.$$

Πρέπει τώρα να μεγιστοποιηθεί η $E_{\theta^{(k)}}(l_c(\theta))$. Επειδή η $l_c(\theta)$ είναι γραμμική έχουμε,

$$E(l_c(\theta)) = E(u_2 + x_4)\log(\theta) + E(x_2 + x_3)\log(1-\theta).$$

Το μέγιστο επιτυγχάνεται όταν

$$\theta^{(k+1)} = \frac{u_2^{(k)} + x_4}{u_2^{(k)} + x_2 + x_3 + x_4}.$$

Πιο κάτω βλέπουμε πως εφαρμόζεται ο Ε-Μ αλγόριθμος για το πιο πάνω παράδειγμα στην R, θέτοντας $\theta^{(0)} = 0.10$:

```
> theta=0.1
> p1=1/2+theta/4
> p2=1/4-theta/4
> p3=1/4-theta/4
> p4=theta/4
> x<-rmultinom(1,size=100,prob=c(p1,p2,p3,p4))
```

```
>
> thetainitial=0.60  ####initial value, that is \theta_{0}
> it  <- 0  ####iterative count
> del <- 1  ####iterative adjustment
> thetaold=thetainitial  ###assign the initial value to theta_{1}
> while(abs(del) > 0.000001 && (it <- it+1) < 20) ##Loop for 20 iterations and
+                                                  ##prespecified presicion
+ {
+ u2=(x[1]*thetaold)/(2+thetaold)        ##Expectation step
+ thetanew=(u2+x[4])/(sum(x)-x[1]+u2)    ##Maximization step
+ del=thetanew-thetaold                  ##Calculate the difference
+                                        ##between two iterations
+ thetaold=thetanew                      ##assign thetanew to thetaold for
+                                        ##the recursions
+ cat(it, thetanew, "\n")                ##List the iterations
+ }
1 0.2333333
2 0.1391061
3 0.1047372
4 0.09068726
5 0.08467897
6 0.08206028
7 0.08090947
8 0.08040192
9 0.0801777
10 0.08007859
11 0.08003476
12 0.08001537
13 0.0800068
14 0.080003
15 0.08000133
16 0.08000059
```

Όπως παρατηρούμε, για να πάρουμε ακρίβεια στο έκτο δεκαδικό στοιχείο στην εκτίμησή μας χρειάστηκαν δεκαέξι επαναλήψεις, και η εκτιμήτρια μεγίστης πιθανοφάνειας για το θ υπολογίστηκε να είναι ίση με `0.08000059`, ενώ η αληθινή τιμή του θ είναι ίση με 0.10. Ξαντρέχωντας τον αλγόριθμο για $\theta^{(0)} = 0.2$ παίρνουμε τα

πιο κάτω αποτελέσματα:

```
1 0.3402985
2 0.2634429
3 0.2330699
4 0.2197439
5 0.2136309
6 0.2107696
7 0.2094176
8 0.2087760
9 0.2084709
10 0.2083256
11 0.2082564
12 0.2082235
13 0.2082078
14 0.2082003
15 0.2081967
16 0.208195
17 0.2081942
```

Είναι φανερό ότι σε αυτήν την περίπτωση για να γίνει η σύγκλιση σε ακρίβεια στο έκτο δεκαδικό στοιχείο χρειάστηκαν δεκαεπτά επαναλήψεις, καταλήγοντας στην τιμή `0.2081942` ως εκτίμηση του θ.

23.2 Δεύτερο Παράδειγμα: Παραλλαγή του Πειράματος Ελέγχου Επιβίωσης Χρησιμοποιώντας Εκθετικό Μοντέλο

Έστω ότι ο χρόνος επιβίωσης ενός λαμπτήρα ακολουθεί την εκθετική κατανομή με μέση τιμή θ. Για να εκτιμηθεί το θ, καταμετρήθηκε ο χρόνος n λαμπτήρων από την ώρα που ανάβουν για πρώτη φορά μέχρι να πάψουν να λειτουργούν, $x_1, \ldots, x_n$. Σε ένα άλλο πείραμα, ελέγχθηκαν m λαμπτήρες, αλλά αυτήν την φορά δεν καταμετρήθηκαν ξεχωριστά ο χρόνος επιβίωσης του κάθε λαμπτήρα, αλλά ο αριθμός των λαμπτήρων, r, οι οποίοι έπαψαν να λειτουργούν σε μία χρονική στιγμή t.

Τα ελλειπή δεδομένα είναι οι χρόνοι επιβίωσης των λαμπτήρων στο δεύτερο

πείραμα, $u_1, \ldots, u_m$. Τότε,

$$l_c(\theta; x; u) = -n(\log(\theta) + \bar{x}/\theta) - \sum_{i=1}^{m} (\log(\theta) + u_i/\theta).$$

Η αναμενόμενη τιμή του χρόνου επιβίωσης ενός λαμπτήρα που δεν έχει ακόμη πάψει να λειτουργεί είναι ίση με

$$t + \theta,$$

ενώ, κάποιου που έχει πάψει να λειτουργεί είναι ίση με

$$\theta - \frac{te^{-t/\theta^{(k)}}}{1 - e^{-t/\theta^{(k)}}}.$$

Συνεπώς, χρησιμοποιώντας μία προσωρινή τιμή $\theta^{(k)}$, και το γεγονός ότι οι r από τους m λαμπτήρες δεν λειτουργούν, έχουμε την $E_{U|x,\theta^{(k)}}(l_c)$ να δίνεται στη μορφή

$$q^{(k)}(x, \theta) = -(n + m)\log(\theta) - \frac{1}{\theta}(n\bar{x} + (m - r)(t + \theta^{(k)}) + r(\theta^{(k)} - th^{(k)})),$$

όπου,

$$h^{(k)} = \frac{e^{-t/\theta^{(k)}}}{1 - e^{-t/\theta^{(k)}}}.$$

Το k-οστό Μ βήμα ορίζει τη μέγιστη τιμή συναρτήσει της μεταβλητής θ, η οποία, δεδομένου του $\theta^{(k)}$, παρατηρείται στο σημείο

$$\theta^{(k+1)} = \frac{1}{n + m}(n\bar{x} + (m - r)(t + \theta^{(k)}) + r(\theta^{(k)} - th^{(k)})).$$

Ξεκινώντας με μία θετική τιμή $\theta^{(0)}$, η πιο πάνω εξίσωση επαναλαμβάνεται μέχρι να επέλθει η σύγκλιση. Η αναμενόμενη τιμή $q^{(k)}$ δε χρειάζεται να υπολογίζεται κάθε φορά. Για να δούμε πως δουλεύει ο αλγόριθμος, παράγονται αρχικά μερικά τεχνητά δεδομένα με τη βοήθεια της R:

```
> # Generate data from an exponential with theta=2, and with the second
> # experiment truncated at t=3. Note that R uses a form of the
> # exponential in which the parameter is a multiplier; i.e., the R
> # parameter is 1/theta. Set the seed, so computations are reproducible.
>
> set.seed(4)
> n<-100
> m<-500
> theta<-2
```

```
> t<-3
> x<-rexp(n,1/theta)
> r<-min(which(sort(rexp(m,1/theta))>=3))-1
```

Συνεχίζοντας, εφαρμόζεται ο Ε-Μ αλγόριθμος στην R χρησιμοποιώντας ως αρχική τιμή το $\theta^{(0)} = 1$

```
> # We begin with theta=1.
> # (Note that theta.k is set to theta.kp1 at the beginning of the loop.)
> theta.k<-.01
> theta.kp1<-1
> # Do some preliminary computations
> n.xbar<-sum(x)
> # Then loop and test for convergence
> it  <- 0  ####iterative count
> del <- 1  ####iterative adjustment
> while(abs(del) > 0.000001 && (it <- it+1) < 20) ###Loop for 20 iterations
+                                                 ###and prespecified presicion
+ {
+ theta.kp1<-(n.xbar+
+ (m-r)*(t+theta.k)+
+ r*(theta.k-
+ t*exp(-t/theta.k)/(1-exp(-t/theta.k))
+   )
+ )/(n+m)
+ del<- theta.kp1-theta.k
+ theta.k<-theta.kp1
+ cat(it, theta.kp1, "\n")
+ }
1 0.938414
2 1.631730
3 1.933344
4 2.034420
5 2.065981
6 2.075641
7 2.078580
8 2.079472
9 2.079743
```

```
10 2.079826
11 2.079851
12 2.079858
13 2.079860
14 2.079861
```

Παρατηρείται λοιπόν ότι η σύγκλιση στην εκτίμηση του θ επέρχεται μετά από δεκατέσσερις επαναλήψεις, δίνοντας την τιμή `2.079861` με ακρίβεια στο έκτο δεκαδικό στοιχείο.

23.3 Τρίτο Παράδειγμα: Εκτίμηση Κανονικού Μοντέλου Πεπερασμένης Μίξης

Ένα κανονικό μοντέλο μίξης μπορεί να ορισθεί από δύο κανονικές κατανομές, $N(\mu_1, \sigma_1^2)$ και $N(\mu_2, \sigma_2^2)$. Η πιθανότητα μία τυχαία μεταβλητή (αυτή που μπορεί να παρατηρηθεί) να ακολουθεί την πρώτη κατανομή είναι w. Η παράμετρος σε αυτό το μοντέλο είναι το διάνυσμα $\theta = (w, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2)$. Η συνάρτηση πυκνότητας πιθανότητας της μίξης δίνεται από:

$$p(y;\theta) = wp_1(y;\mu_1,\sigma_1^2) + (1-w)p_2(y;\mu_2,\sigma_2^2),$$

όπου $p_j(y;\mu_j,\sigma_j^2)$ είναι η συνάρτηση πυκνότητας πιθανότητας της κανονικής με παραμέτρους μ_j και σ_j^2.

Στον τυπικό μετασχηματισμό $C = (X, U)$, το X συμβολίζει τα δεδομένα που έχουν παρατηρηθεί, και το U δίνει την κατηγορία των δεδομένων που δεν έχουν παρατηρηθεί. Έστω $U = 1$ αν η παρατήρηση είναι από την πρώτη κατανομή, και $U = 0$ αν η παρατήρηση είναι από την δεύτερη κατανομή. Η μη δεσμευμένη αναμενόμενη τιμή $E(U)$ δίνει την πιθανότητα μία παρατήρηση να προέρχεται από την πρώτη κατανομή, η οποία είναι ίση με w.

Έστω n παρατηρήσεις του X, $x_1, \ldots, x_n$. Δεδομένης μιας αρχικής τιμής του θ, μπορεί να εκτιμηθεί η δεσμευμένη αναμενόμενη τιμή $E(U/x)$ για οποιαδήποτε πραγμάτωση του X:

$$E(U/x, \theta^{(k)}) = \frac{w^{(k)} p_1(x;\mu_1^{(k)}, \sigma_1^{2(k)})}{p(x; w^{(k)}, \mu_1^{(k)}, \sigma_1^{2(k)}, \mu_2^{(k)}, \sigma_2^{2(k)})}.$$

Το βήμα Μ του Ε-Μ αλγόριθμου είναι οι γνωστές Ε.Μ.Π. των παραμέτρων:

$$w^{(k+1)} = \frac{1}{n} \sum E(U|x_i, \theta^{(k)})$$

$$\mu_1^{(k+1)} = \frac{1}{nw^{(k+1)}} \sum q^{(k)}(x_i, \theta^{(k)}) x_i$$

$$\sigma_1^{2(k+1)} = \frac{1}{nw^{(k+1)}} \sum q^{(k)}(x_i, \theta^{(k)})(x_i - \mu_1^{(k+1)})^2$$

$$\mu_2^{(k+1)} = \frac{1}{n(1-w^{(k+1)})} \sum q^{(k)}(x_i, \theta^{(k)}) x_i$$

$$\sigma_2^{2(k+1)} = \frac{1}{n(1-w^{(k+1)})} \sum q^{(k)}(x_i, \theta^{(k)})(x_i - \mu_2^{(k+1)})^2$$

Για να δούμε πως δουλεύει ο αλγόριθμος για την εκτίμηση της w, παράγουμε μερικά τεχνητά δεδομένα στην R:

```
> # Normal mixture. Generate data from normal mixture with w=0.7,
> # mu_1=0, sigma^2_1=1, mu_2=1, sigma^2_2=2.
> # Note that R uses sigma, rather than sigma^2 in rnorm.
> # Set the seed, so computations are reproducible.
>
> set.seed(4)
> n<-300
> w<-0.7
> mu1<-0
> sigma21<-1
> mu2<-5
> sigma22<-2
> x<-ifelse(runif(n)<w, rnorm(n,mu1,sqrt(sigma21)),rnorm(n,mu2,sqrt(sigma22)))

> # Initialize
> theta.k<-.1
> theta.kp1<-.5
>
> it  <- 0  ####iterative count
> del <- 1  ####iterative adjustment
> while(abs(del) > 0.000001 && (it <- it+1) < 20) ###Loop for 20 iterations
+                                                 ###and prespecified presicion
+ {
+ tmp<-theta.k*dnorm(x,mu1,sqrt(sigma21))
+ ehat.k<-tmp/(tmp+(1-theta.k)*dnorm(x,mu2,sqrt(sigma22)))
```

```
+ theta.kp1<-mean(ehat.k)
+ del<- theta.kp1-theta.k
+ theta.k<-theta.kp1
+ cat(it, theta.kp1, "\n")
+ }
1 0.6130451
2 0.6686083
3 0.6715901
4 0.671751
5 0.6717596
6 0.6717601
```

Όπως φαίνεται από τα αποτελέσματα, ο αλγόριθμος συγκλίνει σε έξι επαναλήψεις, με ακρίβεια στο έκτο δεκαδικό στοιχείο, στην εκτίμηση `0.6717601`.