Τμήμα Μηχανικών Πληροφορικής τ.ε.
Τεχνολογικό Εκπαιδευτικό Ίδρυμα Δυτικής Ελλάδας

Τμήμα Μηχανικών Πληροφορικής Τ.Ε.Ι
Δυτικής Ελλάδας

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

"Προγραμματισμός και ανάλυση δεδομένων με το στατιστικό πακέτο R"

ΠΟΛΙΤΗΣ ΜΑΡΙΝΟΣ

Α.Μ:1794

ΕΠΙΒΛΕΠΩΝ: Κούγιας Ιωάννης

Ευχαριστίες

Η ολοκλήρωση αυτής της έρευνας υλοποιήθηκε με την υποστήριξη ενός αριθμού ανθρώπων, που χωρίς αυτούς θα ήταν δύσκολο να πραγματοποιηθεί. Πρώτα από όλα θα ήθελα να ευχαριστήσω τον επιβλέποντα της πτυχιακής εργασίας, κ. Ιωάννη Κούγια, για την πολύτιμη βοήθειά του αλλά και καθοδήγησή του καθ' όλη την διάρκεια της δουλειάς μου. Επίσης, είμαι ευγνώμων στην μητέρα μου και την αδερφή μου, που ήταν δίπλα μου και με υποστήριζαν από την αρχή των σχολικών μου χρόνων μέχρι και τώρα που φτάνω στο τέλος.

Περιεχόμενα

Περίληψη

Σκοπός αυτής της εργασίας είναι η γνωριμία, η κατανόηση και εξοικείωση με το πρόγραμμα R σε ότι αφορά θέματα στατιστικής φύσης. Η έρευνα αυτή θα αναφερθεί ως επί των πλείστων σε βασικούς ορισμούς της στατιστικής, σε εισαγωγικά θέματα που αφορούν το πρόγραμμα, αλλά και σε στατιστικές μελέτες χρησιμοποιώντας το R. Μέσω των ασκήσεων θα δούμε, με τον πιο απλό τρόπο την επίλυση προβλημάτων χρησιμοποιώντας εντολές και στην συνέχεια θα εξετάσουμε τα αποτελέσματα τα οποία προέκυψαν. Τέλος, ακόμα ένας σκοπός αυτής της εργασίας είναι η εξαγωγή χρήσιμων συμπερασμάτων σχετικά με τις στατιστικές μελέτες μέσω του προγράμματος R, η κριτική του προγράμματος αυτού κάθε αυτού, η σύγκριση του με αλλά στατιστικά προγράμματα , αλλά και η διαμόρφωση προτάσεων σχετικά με το πρόγραμμα.

Εισαγωγή

Η παρούσα εργασία αποτελείται από 3 κεφάλαια. Στο πρώτο κεφάλαιο γίνεται η γνωριμία με το περιβάλλον καθώς και μια ιστορική αναδρομή του προγράμματος. Στο δεύτερο και τρίτο κεφάλαιο γίνεται η παρουσίαση της R μέσω παραδειγμάτων και ασκήσεων. Γενικές πληροφορίες για το πρόγραμμα Το R είναι μια πλατφόρμα την οποία την χρησιμοποιούν για την επεξεργασία υπολογισμών, γραφημάτων και την εφαρμογή στατιστικών τεχνικών. Οι δυνατότητες του είναι τεράστιες μίας και ο χρήστης έχει την δυνατότητα να χρησιμοποιεί έτοιμα προγράμματα τα οποία είναι ενσωματωμένα μέσα σε πακέτα ή μπορεί να προγραμματίσει και ο ίδιος για την επίλυση πολύπλοκων προβλημάτων. Η γλώσσα πάνω στην οποία ο χρήστης μπορεί να προγραμματίσει είναι μια διάλεκτος της S. Αν και το R με την S δεν είναι απόλυτα συμβατά μεταξύ τους, μπορεί και τρέχει στο R χωρίς καμία αλλαγή. Στην S οι εντολές αφού διαβαστούν εκτελούνται αμέσως κάτι το οποίο δεν συμβαίνει στη γλώσσα Fortran. Ένα χαρακτηριστικό των διερμηνέων γλωσσών ,όπως είναι η S,είναι ότι επιτρέπουν την σταδιακή ανάπτυξη. Πιο αναλυτικά, ο χρήστης δημιουργεί μια συνάρτηση, την εκτελεί και μετά έχει την δυνατότητα να δημιουργήσει μια καινούρια η οποία χρησιμοποίει και την πρώτη. Τέλος, ένα από τα πλεονεκτήματα του R είναι ότι μπορεί να αποκτηθεί δωρεάν, μέσα από τις ιστοσελίδες http:/www.r-project.org και http:/www.cran.r-project.org .

Ιστορική αναδρομή της R

Εφαρμόστηκε για πρώτη φορά από τον Robert Gentleman και Ross Ihaka, και οι δύο μέλη ΔΕΠ στο Πανεπιστήμιο του Όκλαντ, στις αρχές της δεκαετίας του 1990. Robert και ο Ross καθιέρωσε R ως ένα έργο ανοικτού κώδικα το 1995. Από το 1997 το έργο R έχει αντιμετωπιστεί από την Ομάδα Πυρήνα R. R 1.0.0 κυκλοφόρησε το Φεβρουάριο του 2000.Στα μέσα της δεκαετίας του 1980,το στατιστικό λογισμικό που ονομάστηκε S αναπτύχθηκε στα εργαστήρια AT&T στο New Jersey χρησιμοποιώντας την ερμηνευτική γλώσσα υπολογιστή Συστήματος. Γράφτηκε για να χειριστεί στατιστική μοντελοποίηση και σχεδίαση και να παραταθεί χωρίς τροποποιήσεις. Αν και έχει επεκταθεί θα πολλαπλασιαστεί μακροπρόθεσμα με επιπλέον λειτουργίες και δυνατότητες, ουσιαστικά όμως η βασική του μορφή θα είναι ίδια μετά από 2 δεκαετίες. Η γλώσσα προγραμματισμού S μετατράπηκε σε S-PLUS και έγινε ένα εμπορικό πακέτο, ουσιαστικά. είχε διαμορφωθεί κατά το πρότυπο της γλώσσας S για στατιστικούς υπολογισμούς σχεδιάστηκε από τον John Chambers, Rick Becker, Trevor Hastie, Allan Wilks και άλλοι στα Bell Labs στα μέσα της δεκαετίας του 1970 και έκανε δημόσια διαθέσιμες στις αρχές της δεκαετίας του 1980.Το 1994, ο Ross Ihaka και Robert Gentleman στο Πανεπιστήμιο Όκλαντ έγραψε την πρώτη έκδοση έκδοσης της S σαν πακέτο λογισμικού και την ονόμασε “R”, συνεχίζοντας την παράδοση της επιστήμης των υπολογιστών π. χ. C, S. Έκαναν το λογισμικό αυτό να διατίθεται ελεύθερα και αυτή η κίνηση τους αποδείχθηκε πως είχαν πιάσει το πνεύμα των άλλων προγραμματιστών λογισμικού (Luke Tierney είχε αναπτύξει την Lisp-Stat , ο Martin Maechler είχε γράψει Emacs) και ένωσαν τις δυνάμεις τους. Η R συνεχίζει να αναπτύσσεται και τώρα υποστηρίζεται από κορυφαίους στατιστικολόγους και επιστήμονες γνώστες των υπολογιστών με παγκόσμια φήμη. Είναι λογισμικό ανοικτού κώδικα (όπου στα επόμενα κεφάλαια θα αναλυθεί εκτενέστερα) και είναι διαθέσιμο ελεύθερα. Η R εμφανίζεται παρόμοια χαρακτηριστικά με S ή S-PLUS, αλλά ουσιαστικά είναι διαφορετικές και σε αυτό το σημείο δεν θα αναπτυχθούν οι διαφορές μεταξύ τους. Έχοντας αναφέρει για τον προγραμματισμό και την ιστορία των γλωσσών ωφέλιμο θα ήταν σε αυτό το εισαγωγικό σημείο να τονίσουμε πως η γλώσσα προγραμματισμού R που θα εξετάσουμε υπάγεται στον αντικειμενοστραφής προγραμματισμό.

Κεφάλαιο 1ο

1. Γνωριμία με το περιβάλλον

Τα τελευταία περίπου δέκα χρόνια το R έχει γίνει ένα από τα πιο σημαντικά στατιστικά εργαλεία. Υπολογίζεται μάλιστα ότι πάνω από τρία εκατομμύρια χρηστές το χρησιμοποιούν τόσο στην ακαδημαϊκή κοινότητα όσο και στον επαγγελματικό τομέα. Το περιβάλλον του είναι απλό. Ανοίγοντας το πρόγραμμα εμφανίζεται η βασική οθόνη στην οποία βρίσκεται το παράθυρο των εντολών και η γραμμή εργαλείων. Πατώντας το κουμπί “file” να μπορούμε να κάνουμε μια σειρά από βασικές ενέργειες. Αρχικά, μπορούμε να εισάγουμε κώδικα και γενικότερα εντολές από προηγούμενες αναλύσεις και εφαρμογές μας. Αυτό επιτυγχάνεται με το source R

code. Μια πολύ σημαντική επιλογή που μας δίνει το πρόγραμμα είναι το "new script". Εδώ μπορούμε να γράψουμε τις εντολές που θέλουμε να εκτελέσουμε.

Μαυρίζοντας αυτές που θέλουμε να τρέξουμε και πατώντας με δεξί κλικ πάνω στον συντάκτη επιλεγούμε το run line ή selection. Μπορούμε να ανοίξουμε έναν παλιό συντάκτη με το "openscript" και να δούμε τα αρχεία R που μπορούμε να χρησιμοποιήσουμε του φακέλου που βρισκόμαστε με το "display file(s)". Μπορούμε να φορτώσουμε και να αποθηκεύσουμε χώρο εργασίας (load/save workspace) όπως και να φορτώσουμε ή να αποθηκεύσουμε εντολές που έχουμε χρησιμοποιήσει στο παρελθόν (load/save history). Με την επιλογή "change dir" μπορούμε να αλλάξουμε τον φάκελο εργασίας μας. Τέλος, μπορούμε να εκτυπώσουμε (print) να αποθηκεύσουμε τη δουλειά μας σε μορφή txt (save to file) και να τερματίσουμε το πρόγραμμα (exit).

Στο menu Edit μας παρέχετε η δυνατότητα της αντιγραφής(copy),επικόλλησης (paste), της επιλογής όλων όσων έχουμε πληκτρολογήσει (select all) πχ εντολές, όπως επίσης και το να καθαρίσομε το παράθυρο των εντολών. Τέλος, μπορούμε κάνοντας κλικ πάνω στο "data editor" και "GUI preferences" να ανοίξουμε έναν συντάκτη δεδομένων για τα δεδομένα που είναι υπό τη μορφή πλαισίου δεδομένων και να τα επεξεργαστούμε και να αλλάξουμε το πώς φαίνεται το περιβάλλον στο όποιο δουλεύουμε αντίστοιχα.

Στο "View" μπορούμε να την εμφανίσουμε ή όχι το την μπάρα τα βασικά εργαλεία δουλειάς(toolbar) από το περιβάλλον εργασίας όπως επίσης και τις πληροφορίες για την έκδοση του προγράμματος που χρησιμοποιείτε (statusbar). Πατώντας το κουμπί "misc"μπορούμε να σταματήσουμε το τρέχον πρόγραμμα (stop current computations) ή όλα τα προγράμματα που εκτελούνται(stop all computations). Επίσης, έχουμε την δυνατότητα να σταματήσουμε την εκτύπωση των αποτελεσμάτων στην οθόνη (buffered output), να δούμε όλα τα αντικείμενα και τις αναλύσεις που έχουμε κάνει (list objects) και να τα διαγράψουμε (remove all objects). Τέλος , να δούμε τις βιβλιοθήκες (libraries) και τα πλαίσια (data frames) που υπάρχουν στο περιβάλλον εργασίας μας.

Από το μενού packages ο χρήστης μπορεί να φορτώσει βιβλιοθήκες που ειδή υπάρχουν (load packages), να κατεβάσει και να εγκαταστήσει βιβλιοθήκες από πρότυπα CRAN(install package(s)),να εγκαταστήσει από zip αρχεία μέσα από τον σκληρό του δίσκο (install package(s) from local zip files) και να τις ενημερώσει με πιο πρόσφατες εκδοχές τους. Τέλος ο χρήστης μπορεί να επιλέξει από πιο μέρος του κόσμου θα κατεβάσει μέσω των προτύπων CRAN τις βιβλιοθήκες (set CRAN mirror) και να επιλέξει, πέρα από το CRAN, από ποιόν διανομέα θέλει να τις κατεβάσει (set repositories).

Με το μενού windows μπορεί κάποιος να μετακινηθεί μεταξύ των παραθύρων των οποίων χρησιμοποίει εκείνη την στιγμή. Επίσης μπορεί να τα τοποθετήσει όπως επιθυμεί είτε κάθετα(Tile Vertically) είτε οριζόντια(Tile Horizontally).

Από το μενού Help ο χρήστης μπορεί να βοηθήσει για όλες τις ιδιότητες του πακέτου. Πιο αναλυτικά: Στο Console υπάρχουν πληροφορίες για την βασική οθόνη του προγράμματος R. Στα FAQ on R,FAQ on R for Windows υπάρχουν απαντήσεις σε ερωτήσεις που γίνονται συχνά για την R. Στο Manuals (in PDF) έχουμε το βασικό εγχειρίδιο χρήσης της R σε PDF. Στο R functions(text) έχουμε πληροφορίες για τις ήδη υπάρχουσες εντολές της R. Με το Html help μεταφερόμαστε σε έναν διαδικτυακό τόπο όπου μας παρέχει πληροφορίες για το πρόγραμμα μας. Από το Search help μπορούμε να ψάξουμε όποιο αρχείο επιθυμούμε να βρούμε . Στο Search.r-project.org μπορούμε να αναζητήσουμε όποιον σύνδεσμο στο διαδίκτυο θέλουμε. Από το Apropos μπορούμε να αναζητήσουμε εντολές που είναι ήδη φορτωμένες στην R. Από το R project home page μεταφερόμαστε στην ιστοσελίδα της R. Από το CRAN home page μεταφερόμαστε στην ιστοσελίδα της CRAN. About μας παρέχει πληροφορίες για τα δικαιώματα και την τρέχον έκδοση του πακέτου μας.

Το R όντας ένα πρόγραμμα με πολλές δυνατότητες σου παρέχει την δυνατότητα να του φορτώσεις δεδομένα από πολλούς διαφορετικούς τύπους αρχείων. Με την χρήση διαφορετικών βασικών εντολών μπορούμε να εισάγουμε δεδομένα από τους εξής τύπους αρχείων: Excel,Minitab,SPSS,Table,CSV, Stata, systat. Πιο αναλυτικά, αρκετά συχνά τα δεδομένα μας είναι σε μορφή Excel. Για να τα εισάγου στο R χρησιμοποιούμε την εντολή data<-read.xls(“data.xls”), όπου με το “data<-” εισάγουμε την τιμή μας στο αντικείμενο data. Επίσης πολύ σημαντικό είναι το ότι άμα δεν το αρχείο μας δεν βρίσκεται στον ίδιο φάκελο με το πρόγραμμα μας τότε μέσα στην παρένθεση θα πρέπει να γράψουμε το μονοπάτι της ακριβής τοποθεσίας του αρχείου μας. Για παράδειγμα αν τα 18 δεδομένα μας είναι στο σκληρό δίσκο C στον φάκελο παράδειγμα τότε η εντολή που θα πρέπει να γράψουμε θα είναι η εξής data<-read.xls ("C:παράδειγμαdata.xlsx"). Εάν τα δεδομένα μας είναι σε μορφή Minitab ο τρόπος διαβάσματος του αρχείου θα είναι ο ίδιος με μια μικρή διαφορά. Θα γράψουμε data<- read.mtp(“data.mtp”) και σε περίπτωση που το αρχείο μας είναι σε διαφορετικό φάκελο θα πράξουμε με τον ίδιο τρόπο Με τον ίδιο τρόπο περνάμε τα δεδομένα στο πρόγραμμα μας με την μόνη διαφορά τα τελειώματα τα όποια θα αντιστοιχούν στον τύπο του αρχείου που είναι αποθηκευμένα. Αν είναι αποθηκευμένα σε αρχείο SPSS τότε θα έχω data<- read.spss(“data.spss”),αν είναι σε αρχείο table ή csv ή Stata ή systat θα έχω data<-read.table(“data.txt”) ,data<-read.csv(“data.csv”), data<- read.dta(“data.dta”) και data<-read.systat(“data.dta”)αντίστοιχα.

2. Πλεονεκτήματα

Ένα από τα μεγαλύτερα θετικά στοιχεία του προγράμματος είναι τα σχετικά λίγα προβλήματα που μπορεί να συναντήσει ο χρήστης καθώς και η εύκολη επίλυση τους. Η ανοιχτή κοινότητα R είναι πάντοτε πρόθυμη να βοηθήσει αρχάριους αλλά και προχωρημένους χρήστες, η εκμάθηση και η κατανόηση του R γίνεται με σχετική άνεση. Ακόμα και με το ίδιο πρόγραμμα να δίνει λύσεις, όπου αυτό γίνεται δυνατό, ο

χρήστης μαθαίνει να βρίσκει και να επιλύει προβλήματα παντός φύσης. Ένα ακόμα σημαντικό πλεονέκτημα είναι πως ρο πρόγραμμα λειτουργεί με βιβλιοθήκες(packages). Με την χρήση βιβλιοθηκών ο χρήστης μπορεί να χρησιμοποιήσει πληθώρα παραδειγμάτων προς επίλυση όπως επίσης και εντολές οι οποίες βοηθάνε στην λύση των ασκήσεων. Επίσης ακόμα και αν δεν υπάρχει κάποια βιβλιοθήκη έτσι ώστε να βοηθήσει το χρήστη, ο ίδιος μπορεί να τη δημιουργήσει. Το R πέρα από πολυεργαλείο είναι μια γλώσσα προγραμματισμού η οποία μας δίνει την δυνατότητα να κατασκευάσουμε την εντολή που χρειαζόμαστε. Ένα από τα με μεγαλύτερα πλεονεκτήματά της θα μπορούσαμε ακόμα να αναφέρουμε την λυτή αλλά άκρως κατανοητή παρουσίαση δεδομένων στο περιβάλλον του προγράμματος. Το R εμφανίζει στον χρήστη το απαιτούμενο αποτέλεσμα με όλες τις πληροφορίες που μπορεί να ζητήσει ο χρήστης χωρίς να χρειάζεται να ανατρέξει αλλού ή να δώσει παραπάνω εντολές από τις απαιτούμενες. Επιπλέον το R δίνει ένα ευρύ φάσμα γραφημάτων στα οποία γίνεται αντιληπτή και η πιο μικρή και υπάρχει η δυνατότητα να το προσαρμόσει ο χρήστης στις δικές του ανάγκες και απαιτήσεις. Ένα ακόμα σημαντικό πλεονέκτημα είναι πως είναι δωρεάν το πρόγραμμα και μπορεί να χρησιμοποιηθεί από οποιονδήποτε εκτός από το κομμάτι στην πληθώρα εργασιών μπορεί να χρησιμοποιηθεί στον προγραμματισμό και στην δημιουργία βάσεων δεδομένων.

3. Μειονεκτήματα

Πέρα από τα θετικά στοιχεία παρατηρούμε και μια σειρά από κάποια μειονεκτήματα τα οποία δεν ξεπερνούν σε αριθμό τα πλεονεκτήματα. Επειδή δεν διδάσκεται σε αρκετά εκπαιδευτικά ιδρύματα αντιμετωπίζει ένα μεγάλο πρόβλημα αναγνωσιμότητας. Επιπλέον μπορεί να προκαλέσει άσχημη πρώτη εντύπωση στους νέους χρήστες επειδή το περιβάλλον δεν θυμίζει σε τίποτα από τα προγράμματα όπως είναι το Excel και η εικόνα αυτή μπορεί να θεωρηθεί ασυνήθιστη και να δημιουργήσει άσχημη εντύπωση. Αυτό έχει ως αποτέλεσμα ο χρήστης να κάνει αρκετά λάθη και να αφιερώσει αρκετές ώρες μέχρι να φτάσει στο επιθυμητό αποτέλεσμα.

4. Προτάσεις

Ως μια πρόταση, ύστερα από την εξαγωγή των συμπερασμάτων μας, θα μπορούσαμε να πούμε την αναγκαιότητα εκμάθησης του προγράμματος σε όλα τα εκπαιδευτικά ιδρύματα της χώρας. Το R, με την μεγάλη γκάμα δυνατοτήτων είτε στη στατιστική, είτε στα μαθηματικά, είτε στο προγραμματισμό θα ήταν τέλειο εργαλείο δουλειάς για τους φοιτητές γιατί θα τους βοηθούσε να καταλάβουν πολύ πιο εύκολα, πιο εξειδικευμένα προγράμματα ή άλλες γλώσσες προγραμματισμού. Συγκεκριμένα για τις σχολές με κύριο αντικείμενο την στατιστική, την οικονομία, τα μαθηματικά και τους ηλεκτρονικούς υπολογιστές θα πρέπει να θεωρείται απαραίτητο. Το R είναι ένα πρόγραμμα με μηδενικό κόστος για την εκπαίδευση. Τέλος θα ήταν ένας τρόπος προώθησης του R στην Ελληνική αγορά, με τους πιο πιθανούς χρήστες να βρίσκουν άμεσα λύση στα προβλήματα τους ξεπερνώντας έτσι το εμπόδιο μιας ξένης γλώσσας.

5. Σύγκριση με άλλα στατιστικά προγράμματα

Σε σύγκριση με τα υπόλοιπα στατιστικά προγράμματα, το R έχει κάνει τεράστια πρόοδο από την μέρα δημιουργίας του μέχρι σήμερα. Η δωρεάν διανομή του σε σύγκριση με πολλά άλλα στατιστικά πακέτα του δίνει ένα δυναμικό πλεονέκτημα. Επίσης σε σύγκριση με άλλα στατιστικά πακέτα SPSS,STATA έχει μεγάλη ποικιλία γραφημάτων αλλά σου παρέχει μεγάλη ευκολία στην δημιουργία τους, στην διαχείρισή τους και στην μετατροπή τους. Σχετικά με το προγραμματισμό, το R φαίνεται να προτιμάται σε σχέση με το STATA καθώς είναι πολύ πιο εύκολο να το προγραμματίσεις. Στα περισσότερα προβλήματα στατιστικής ακόμα και αν τα περισσότερα πακέτα μας δίνουν τα στοιχεία και τις εξισώσεις που χρειαζόμαστε, πολλές φορές ο προγραμματισμός είναι απαραίτητος και έτσι το R έχει συγκριτικό πλεονέκτημα. Συγκριτικά με το SPSS, το συγκεκριμένο πρόγραμμα, με το πλεονέκτημα του προγραμματισμού επιτρέπει στον χρήστη να κάνει πιο εξειδικευμένες μελέτες, που είναι πιθανό να μην υπάρχουν, να είναι δύσκολο στον χειρισμό ή να απαιτεί περισσότερο χρόνο και περισσότερα βήματα στην ίδια μελέτη στο SPSS. Επίσης, σε διάφορες ιστοσελίδες νε κύριο θέμα την στατιστική, το R δείχνει να έχει μεγάλη απήχηση, με τους νέους ηλικιακά να το υποστηρίζουν σε σχέση με κάποια αλλά στατιστικά πακέτα, ισχυριζόμενοι μάλιστα ότι το πρόγραμμα R είναι για εξειδικευμένες στατιστικές μελέτες σε αντίθεση με το SPSS ή το STATA. Η αλήθεια είναι ότι και ακόμα και σήμερα τα πακέτα αυτά θεωρούνται κορυφαία στατιστικά πακέτα αλλά με το πέρασμα του χρόνου η απήχησή τους μειώνεται. Το R αυξάνει την δυναμική του και επακόλουθου αυτού είναι η συνεχής αύξηση των χρηστών του προγράμματος. Συμβολικό αυτής της απήχησης για το R είναι η δημοσίευση από μία από τις μεγαλύτερες εφημερίδες των Ηνωμένων Πολιτειών της Αμερικής, την New York Times το 2009 το οποίο δηλώνει πως το πακέτο αυτό κερδίζει την εμπιστοσύνη των στατιστικών αναλυτών ανά τον κόσμο καθώς επίσης και ότι μπορεί στο μέλλον να αποτελέσει μεγάλο ανταγωνιστή των μέχρι τότε μεγάλων στατιστικών προγραμμάτων.

6. Συμπεράσματα

Αυτό που μπορεί να συμπεράνει κανείς είναι ότι το πρόγραμμα έχει αρκετά πλεονεκτήματα έναντι άλλων στατιστικών πακέτων, αλλά και προβλήματα που με τον καιρό αντιμετωπίζονται. Το πρόγραμμα R δείχνει να έχει σύμμαχο τον χρόνο καθώς κερδίζει με τον καιρό την εμπιστοσύνη χιλιάδων χρηστών ανά τον κόσμο, με αποτέλεσμα να γίνεται ένα από τα πιο διαδεδομένα και αξιόπιστα προγράμματα όχι μόνο για στατικούς αλλά και για άλλους σκοπούς. Η R είναι μια ισχυρή γλώσσα και ένα ισχυρό περιβάλλον ανάπτυξης για στατιστικούς υπολογισμούς και γραφικά. Ως έργο αποτελεί κοινό κτήμα (ή αλλιώς είναι GNU project), και είναι παρόμοια με την εμπορική γλώσσα και περιβάλλον S, που είχε αναπτυχθεί στα Bell Laboratories (πρώην AT&T, πλέον Lucent Technologies) από τον John Chambers και τους

συνεργάτες του. Η R μπορεί να θεωρηθεί πως είναι μια διαφορετική υλοποίηση της S, και χρησιμοποιείται ευρέως ως εκπαιδευτική γλώσσα και ως ερευνητικό εργαλείο. Τα κύρια πλεονεκτήματα της R είναι το γεγονός ότι η R είναι ελεύθερο λογισμικό και ότι υπάρχει πολύ βοήθεια διαθέσιμη στο διαδίκτυο. Είναι αρκετά παρόμοια με άλλα προγραμματιστικά πακέτα όπως η MATLAB (που δεν είναι ελεύθερο λογισμικό), αλλά πιο φιλική προς τον χρήστη από γλώσσες προγραμματισμού όπως η C++ και η Fortran. Μπορείτε να χρησιμοποιήσετε την R όπως είναι, αλλά για εκπαιδευτικούς λόγους εμείς προτιμούμε τη χρήση της R σε συνδυασμό με την διεπαφή του RStudio (που είναι κι αυτό ελεύθερο λογισμικό), το οποίο έχει μια οργανωμένη διάταξη και διάφορες πρόσθετες επιλογές. Παρ' όλα αυτά παρατηρούμε και αδυναμίες όπως είναι η αργή διάδοση του και πιο συγκεκριμένα στην χώρα μας. Ακόμα και σε αυτή την περίπτωση όμως τα προβλήματα φαίνονται να είναι πολύ λίγα σε σχέση με την δυναμική του και την αξιοπιστία του. Κάτι που μας κάνει να πιστεύουμε πως πολύ σύντομα θα είναι ίσως το πιο επιτυχημένο στατιστικό πακέτο.

Κεφάλαιο 2ο

1. Εισαγωγή Δεδομένων στο Πρόγραμμα

Το R όντας ένα πρόγραμμα με πολλές δυνατότητες σου παρέχει τη δυνατότητα να του φορτώσεις δεδομένα από πολλούς διαφορετικούς τύπους αρχείων. Με την χρήση διαφορετικών βασικών εντολών μπορούμε να εισάγουμε δεδομένα από τους εξής τύπους αρχείων: Excel,Minitab,SPSS,Table,CSV, Stata, systat . Πιο αναλυτικά, αρκετά συχνά τα δεδομένα μας είναι σε μορφή Excel. Για να τα εισάγου στο R χρησιμοποιούμε την εντολή data<-read.xls(“data.xls”), όπου με το “data<-” εισάγουμε την τιμή μας στο αντικείμενο data. Επίσης πολύ σημαντικό είναι το ότι άμα δεν το αρχείο μας δεν βρίσκεται στον ίδιο φάκελο με το πρόγραμμα μας τότε μέσα στην παρένθεση θα πρέπει να γράψουμε το μονοπάτι της ακριβής τοποθεσίας του αρχείου μας . Για παράδειγμα αν τα δεδομένα μας είναι στο σκληρό δίσκο C στον φάκελο παράδειγμα τότε η εντολή που θα πρέπει να γράψουμε θα είναι η εξής data<-read.xls ("C:παράδειγμαdata.xlsx"). Εάν τα δεδομένα μας είναι σε μορφή Minitab ο τρόπος διαβάσματος του αρχείου θα είναι ο ίδιος με μια μικρή διαφορά. Θα γράψουμε data<- read.mtp(“data.mtp”) και σε περίπτωση πουτο αρχείο μας είναι σε διαφορετικό φάκελο θα πράξουμε με τον ίδιο τρόπο. Χρησιμοποιώντας το προηγούμενο παράδειγμα θα έχω: data<-read.mtp("C:παράδειγμαdata.mtp") Με τον ίδιο τρόπο περνάμε τα δεδομένα στο πρόγραμμα μας με την μόνη διαφορά τα τελειώματα τα όποια θα αντιστοιχούν στον τύπο του αρχείου που είναι αποθηκευμένα. Αν είναι αποθηκευμένα σε αρχείο SPSS τότε θα έχω data<-read.spss(“data.spss”), αν είναι σε αρχείο table ή csv ή Stata ή systat θα έχω data<-read.table(“data.txt”),data<- read.csv(“data.csv”), data<- read.dta(“data.dta”) και data<- read.systat(“data.dta”) αντίστοιχα. Τέλος, στην συγκεκριμένη εργασία θα σας

δείξουμε πώς εισάγουμε τα δεδομένα μας χωρίς να τα διαβάσουμε από κάποιο άλλο αρχείο (1.2.4 Αποθήκευση και επανάκτηση δεδομένων Μια άλλη δυνατότητα που μας προσφέρει το R είναι η αποθήκευση των αντικειμένων. Για την αποθήκευση τους χρησιμοποιούμε την εντολή save(data, file="data.Rdata", ascii=TRUE) όπου το data είναι το όνομα του αρχείου μας και όπου το data. Rdata είναι το όνομα του φάκελου που θα αποθηκευτεί. Η παράμετρος "ascii=TRUE" είναι προαιρετική στην περίπτωση που θέλουμε να χρησιμοποιήσουμε το αποθηκευμένο αντικείμενο και σε αλλά στατιστικά πακέτα. Συχνά προβλήματα και αντιμετώπιση τους Τα προβλήματα τα οποία μπορεί να αντιμετωπίσει κάποιος στο πρόγραμμα R δεν είναι πολλά. Οι λύσεις αυτών των προβλημάτων βρίσκονται σχετικά εύκολα, κάτι που κάνει το πρόγραμμα ακόμα πιο αξιόπιστο και λειτουργικό. Τα πιο συνήθη λάθη-προβλήματα που μπορεί να αντιμετωπίσει κάποιος είναι αυτά της ορθογραφίας. Το πρόγραμμαR είναι ευαίσθητο σε κεφαλαία και μικρά γράμματα και όπως γίνεται αντιληπτό καμία εντολή δεν θα πραγματοποιηθεί αν δεν έχει διατυπωθεί με τον σωστό τρόπο. Ακόμα πιθανό είναι να έχει δοθεί στο πρόγραμμα κάποια εντολή αλλά με κάποιο λάθος γράμμα η συμβολισμό. Στις δύο αυτές περιπτώσεις το πρόγραμμα βγάζει ένα μήνυμα λάθους (error) βοηθώντας έτσι τον χρήστη να καταλάβει ποίο ακριβώς είναι το πρόβλημα. Παρ' όλα αυτά, τα λάθη λογικής είναι αυτά τα οποία δυσκολεύουν περισσότερο από αυτά της ορθογραφίας . Συχνά στο πρόγραμμα γίνεται χρήση εντολών οι οποίες μπορεί να είναι σωστές αλλά να μας δίνουν διαφορετικό αποτέλεσμα από αυτό που θέλουμε ή από αυτό που περιμέναμε να δούμε. Τα λάθη λογικής είναι συχνό φαινόμενο στις γλώσσες προγραμματισμού και το R δεν αποτελεί εξαίρεση. Ο χρήστης θα πρέπει να είναι ιδιαίτερα προσεκτικός ώστε να έχει το επιθυμητό αποτέλεσμα. Τέλος, υπάρχουν και τα ανθρώπινα λάθη όπως η λάθος καταχώριση αρχείων, η ονομασία ενός αρχείου με το ίδιο όνομα με ενός άλλου ή η χρησιμοποίηση λάθος βιβλιοθήκης (package).

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.

2. Τελεστές

Τελεστές Εκχώρησης και Σύγκρισης

Με τους Τελεστές εκχώρησης όπως μας προϊδεάζει και η λέξη έχουμε την δυνατότητα να δώσουμε τιμές σε αντικείμενα και μεταβλητές .Οι Τελεστές σύγκρισης μας βοηθούν στο να συγκρίνουμε δυο τιμές. Αυτοί οι Τελεστές είναι οι πιο κάτω.

Τελεστής Ιδιότητα <- Το αριστερό μέρος της σχέσης μας παίρνει την τιμή

-> Το δεξί μέρος της σχέσης μας παίρνει την τιμή

< Μεγαλύτερο

> Μικρότερο

<= Μικρότερο ή ίσο

>= Μεγαλύτερο ή ίσο

!= Όχι ίσο

== Ίσο Αριθμητικοί Τελεστές

Με αυτούς τους Τελεστές μπορούμε να εκτελέσουμε τις βασικές αριθμητικές πράξεις- λειτουργιές, δηλαδή, πρόσθεση, αφαίρεση, πολλαπλασιασμός όπως και να υψώσουμε έναν αριθμό σε δύναμη. Πιο αναλυτικά:

+ Πρόσθεση

- Αφαίρεση

* Πολλαπλασιασμός

/ Διαίρεση

^ Ύψωση σε δύναμη

%/% Ακέραια Διαίρεση

%% Υπόλοιπο Διαίρεσης

Βασικές Αριθμητικές Συναρτήσεις της R

Συνάρτηση Πράξη sqrt() Τετραγωνική ρίζα abs()

Απόλυτη τιμή log()

Λογάριθμος cos()

Συνημίτονο sin()

Ημίτονο tan()

Εφαπτομένη acos()

Τόξο συνημίτονου asin()

Τόξο ημιτόνου atan()

Τόξο εφαπτομένης gamma()

Λογάριθμος συνδυασμών exp() Εκθετική Συνάρτηση

3. Σύνταξη εντολών στην R

Για την εισαγωγή δεδομένων και, γενικότερα, για την απόδοση τιμών σε μεταβλητές χρησιμοποιούμε συνήθως το συνδυασμό "<-" (assignment symbol), που "φαίνεται" σαν βέλος από δεξιά προς τα αριστερά. Η λειτουργία του συμβόλου αυτού είναι να υπολογίσει την έκφραση που δίνεται δεξιά του και να την αποδώσει στη μεταβλητή που βρίσκεται αριστερά του, χωρίς να τυπωθεί το αποτέλεσμα. Έτσι η εντολή > x <- 3.1 που διαβάζεται "x παίρνει (gets) την τιμή 3.1", αποδίδει για επόμενη χρήση την τιμή 3.1 στη μεταβλητή x. Άλλος τρόπος για την αντιστοίχιση τιμών σε μεταβλητές είναι το ίσον "=", ή το αντίστροφο βέλος.

Οι παρακάτω εντολές είναι ισοδύναμες με την προηγούμενη:

> x = 3.1

> 3.1 -> x

Αν το x είχε άλλη τιμή προηγούμενα, αυτή αντικαθίσταται με το 3.1. Δίνοντας στη συνέχεια το όνομα της μεταβλητής ή κάποια επιτρεπτή πράξη με αυτήν παίρνουμε ως εξαγόμενο την αντίστοιχη τιμή. Π.χ.

> x

[1] 3.1

> 3 * x

[1] 9.3

> x <- 3 * x

> x

[1] 9.3

Το σύμβολο [1] στο εξαγόμενο σημαίνει ότι η καταγραφή ξεκινά από το πρώτο στοιχείο του διανύσματος. Σημειώνεται ότι στην S δεν υπάρχουν απλοί αριθμοί, αλλά μόνον διανύσματα, συναρτήσεις λίστες και άλλα αντικείμενα (objects) Οι απλοί αριθμοί θεωρούνται ως διανύσματα διαστάσεως 1. Έτσι το x=3.1 του παραδείγματος, είναι διάνυσμα με μοναδικό στοιχείο.

4. Εγκατάσταση της R

Για να εγκαταστήσετε την R στον υπολογιστή σας (δωρεάν και με νόμιμο τρόπο), πηγαίνετε στην αρχική σελίδα του ιστοτόπου της R1 http://www.r-project.org/ και εκτελέστε τα ακόλουθα (υποθέτοντας ότι δουλεύετε σε έναν υπολογιστή με Windows):

• κάντε κλικ στο download CRAN στην αριστερή στήλη

• επιλέξτε έναν ιστότοπο απ' όπου θα γίνει η λήψη

• επιλέξτε Windows ως λειτουργικό σύστημα

• κάντε κλικ στο base

• επιλέξτε Download R 3.0.3 for Windows 2 και αφήστε τις προεπιλεγμένες απαντήσεις σε όλες τις ερωτήσεις.

Είναι επίσης δυνατόν να εκτελέσετε την R και το RStudio μέσω ενός USB αντί να τα εγκαταστήσετε. Αυτό θα μπορούσε να φανεί χρήσιμο όταν δεν έχετε δικαιώματα διαχειριστή στον υπολογιστή σας. Δείτε και την ξεχωριστή σημείωσή μας "How to use portable versions of R and RStudio"3 για βοήθεια στο συγκεκριμένο θέμα.

Εγκατάσταση του RStudio

Μετά την ολοκλήρωση της εγκατάστασης, θα πρέπει να βλέπετε ένα εικονίδιο "R" στην επιφάνεια εργασίας σας. Κάνοντας κλικ σε αυτό θα εκκινήσετε την τυπική διεπαφή. Εμείς θα σας συνιστούσαμε, ωστόσο, να χρησιμοποιήσετε την διεπαφή του RStudio. 4 Για να εγκαταστήσετε το RStudio, πηγαίνετε στον ιστότοπο:

http://www.rstudio.org/ και εκτελέστε τα ακόλουθα βήματα (υποθέτοντας ότι

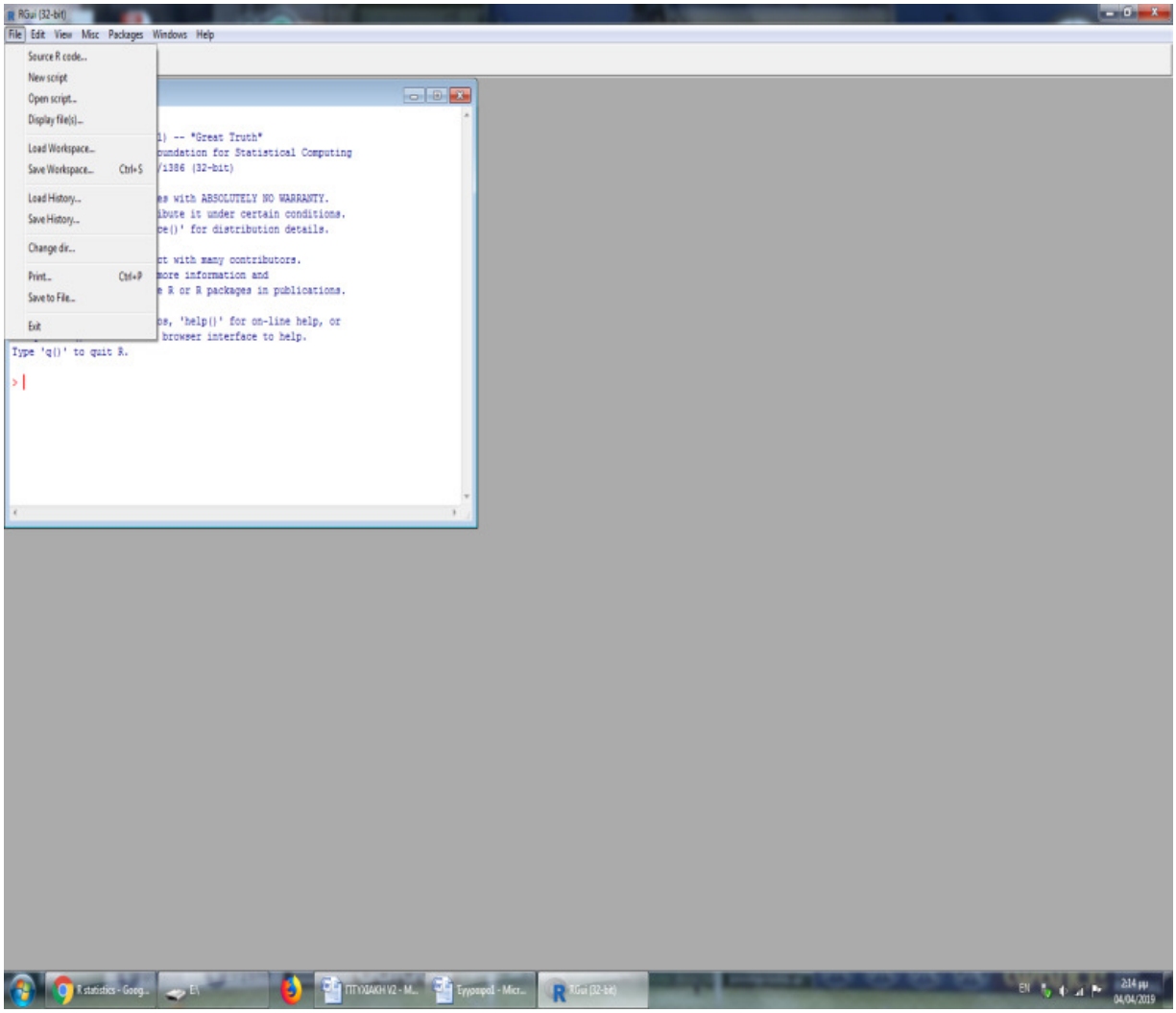
δουλεύετε σε έναν υπολογιστή με Windows):

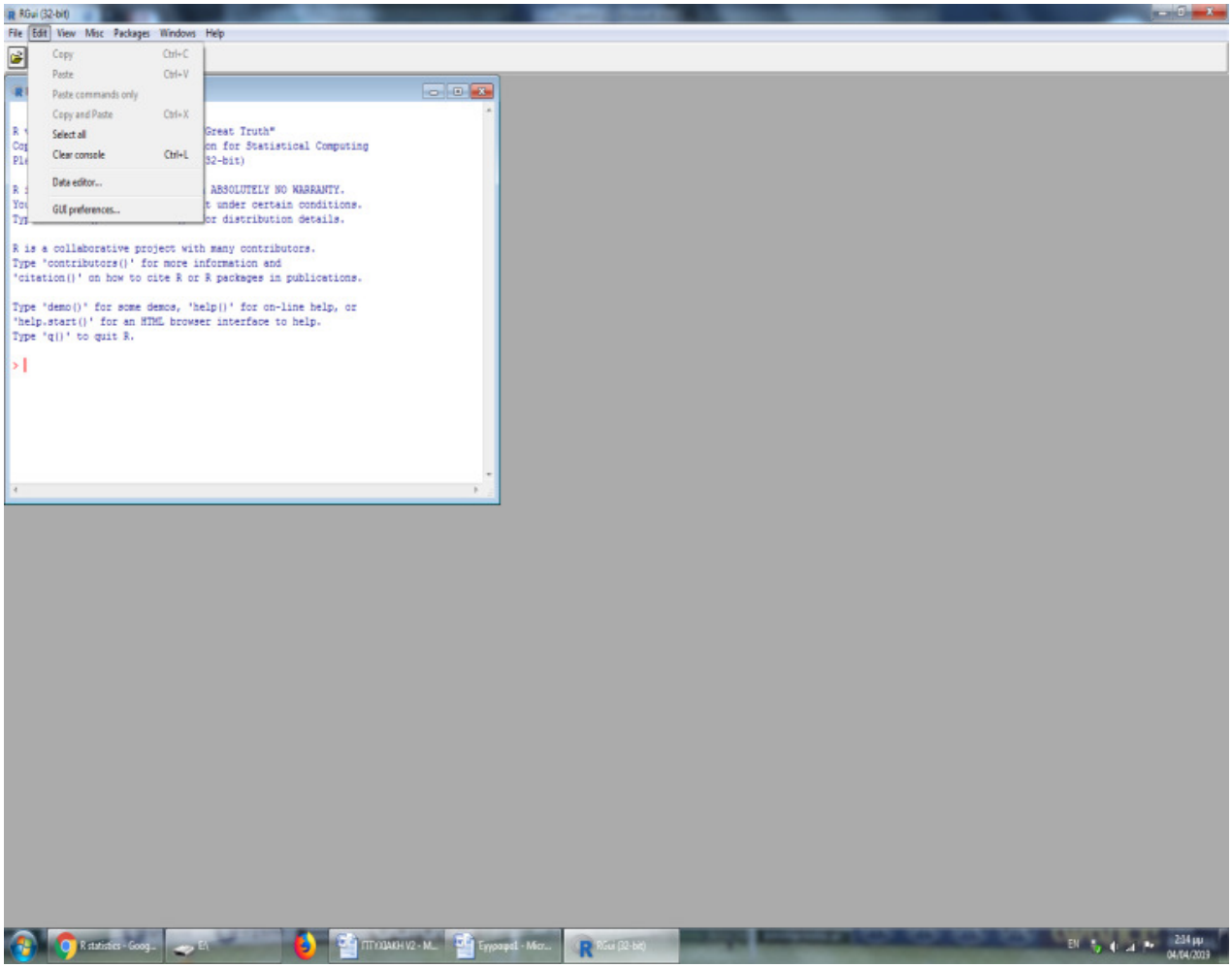
• κάντε κλικ στο Download RStudio

• κάντε κλικ στο Download RStudio Desktop

• κάντε κλικ στο Recommended For Your System

• κατεβάστε το εκτελέσιμο .exe αρχείο και τρέξτε το προεπιλεγμένο πρόγραμμα


RGui (32-bit)
File Edit View Misc Packages Windows Help
Source R code...
New script
Open script...
Display file(s)...
Load Workspace...
Save Workspace... Ctrl+S
Load History...
Save History...
Change dir...
Print... Ctrl+P
Save to File...
Exit
Type 'q()' to quit R.
>
2:14 μμ
04/04/2019



H R μπορεί να χρησιμοποιηθεί ως αριθμομηχανή**.**

Μπορείτε απλά να εισάγετε την εξίσωση που θέλετε στο παράθυρο εντολών μετά από το “>”:

> 10^2 + 36

και η R θα σας δώσει την απάντηση

[1] 136

5. Παραδείγματα - Ασκήσεις

Υπολογίστε τη διαφορά μεταξύ του 2014 και του έτους στο οποίο ξεκινήσατε να σπουδάζετε στο πανεπιστήμιο, και διαιρέστε το με τη διαφορά ανάμεσα στο 2014 και στο έτος το οποίο γεννηθήκατε. Πολλαπλασιάστε το επί 100 για να πάρετε ως αποτέλεσμα το ποσοστό της ζωής σας που έχετε περάσει σε αυτό το πανεπιστήμιο. Χρησιμοποιήστε παρενθέσεις, εάν χρειαστεί. Εάν χρησιμοποιήσετε παρενθέσεις και ξεχάσετε να προσθέσετε μια στο τέλος, τότε το σύμβολο “>” στη γραμμή εντολών αλλάζει και γίνεται “+”. Το “+” μπορεί επίσης να σημαίνει ότι η R είναι ακόμα

απασχολημένη με κάποιον βαρύ υπολογισμό. Εάν θέλετε η R να σταματήσει αυτό που κάνει και να επανέλθει στο σύμβολο “>”, τότε πιέστε ESC.

Χώρος εργασίας

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

> a = 4

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

> a

[1] 4

ή μπορείτε να κάνετε υπολογισμούς με το a:

> a * 5

[1] 20

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

> a = a + 10

> a

[1] 14

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

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

6. Συναρτήσεις

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

> (3+4+5)/3

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

> mean(x=b)

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

7. Γραφικές παραστάσεις

Η R μπορεί να φτιάξει γραφικές παραστάσεις. Το ακόλουθο είναι ένα πολύ απλό παράδειγμα:

> x = rnorm(100)

2 > plot(x)

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

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

Βοήθεια και τεκμηρίωση

Υπάρχει διαθέσιμο πολύ υλικό (δωρεάν) τεκμηρίωσης και βοήθειας. Ένα μέρος της βοήθειας εγκαθίσταται αυτόματα. Η πληκτρολόγηση στο παράθυρο της κονσόλας της εντολής> help(rnorm) θα σας προσφέρει βοήθεια σχετικά με την συνάρτηση rnorm. Σας δίνει μια περιγραφή της συνάρτησης, πιθανά ορίσματα και τις τιμές που χρησιμοποιούνται ως προεπιλογή για τα προαιρετικά ορίσματα. Εάν πληκτρολογήσετε > example(rnorm) θα σας επιστρέψει μερικά παραδείγματα του πώς μπορεί να χρησιμοποιηθεί αυτή η συνάρτηση. Καθολική βοήθεια βασισμένη σε HTML μπορεί να κληθεί μέσω της εντολής: > help.start() ή με μετάβαση στο παράθυρο βοήθειας. Καλείται επίσης και Quick-R. Παρέχει πολύ παραγωγική και άμεση βοήθεια. Επίσης, κατάλληλη για χρήστες που έρχονται από άλλες γλώσσες προγραμματισμού. Και μόνο η χρήση της μηχανής Google (πληκτρολογήστε π.χ. "R rnorm" στο πεδίο αναζήτησης) μπορεί να είναι πολύ παραγωγική.

Σενάρια

Η R είναι ένας διερμηνέας που χρησιμοποιεί ένα περιβάλλον βασισμένο στη γραμμή εντολών. Αυτό σημαίνει ότι θα χρειαστεί να πληκτρολογείτε εντολές, αντί να χρησιμοποιείτε απλά το ποντίκι και τα μενού. Αυτό έχει το πλεονέκτημα του ότι δε χρειάζεται εσείς να πληκτρολογείτε κάθε φορά ξανά όλες τις εντολές, και έτσι έχετε λιγότερες πιθανότητες να εμφανίσετε πόνους στα χέρια, το λαιμό και τους ώμους σας. Μπορείτε να αποθηκεύετε τις εντολές σας σε αρχεία, τα λεγόμενα σενάρια. Αυτά τα σενάρια έχουν τυπικά ονόματα αρχείων με την κατάληξη .R, π.χ. foo.R. Μπορείτε να ανοίξετε τον επεξεργαστή κειμένου σε ένα παράθυρο και να επεξεργαστείτε αυτά τα αρχεία κάνοντας κλικ στο File και New ή στο Open File όπου είναι διαθέσιμες επίσης και οι επιλογές Save και Save as. Μπορείτε να τρέξετε (να στείλετε δηλαδή στο παράθυρο κονσόλας) ένα μέρος του κώδικα, επιλέγοντας γραμμές του και πατώντας CTRL+ENTER ή κάνοντας κλικ στο Run στο παράθυρο του επεξεργαστή κειμένου. Εάν δεν επιλέξετε κάτι, η R θα τρέξει τη γραμμή στην οποία βρίσκεται ο κέρσορας. Μπορείτε πάντα να τρέξετε όλο το σενάριο με την εντολή κονσόλας source και έτσι π.χ. για το σενάριο στο αρχείο foo.R θα πρέπει να πληκτρολογήσετε:

>source("foo.R").

Κεφάλαιο 3ο

1. Δομές δεδομένων

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

Διανύσματα

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

1 > vec1 = c(1,4,6,8,10)

2 > vec1

3 [1] 1 4 6 8 10

4 > vec1[5]

5 [1] 10

6 > vec1[3] = 12

7 > vec1

8 [1] 1 4 12 8 10

9 > vec2 = seq(from=0, to=1, by=0.25)

10 > vec2

11 [1] 0.00 0.25 0.50 0.75 1.00

12 > sum(vec1)

13 [1] 35

14 > vec1 + vec2

15 [1] 1.00 4.25 12.50 8.75 11.00

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

Μητρώα

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

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

2 > mat

3 [,1] [,2] [,3]

4 [1,] 9 3 5

5 [2,] 2 4 6

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

Παραδείγματα

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

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

1 > mat[1,2]

2 [1] 3

3 > mat[2,]

4 [1] 2 4 6

5 > mean(mat)

6 [1] 4.8333

Τα στοιχεία ενός μητρώου μπορούν να προσπελαστούν με το συνήθη τρόπο: [row,column] (γραμμή 1).

Γραμμή 3: όταν θελήσετε να επιλέξετε μια ολόκληρη γραμμή, αφήστε τη θέση για το όρισμα του αριθμού των στηλών κενή (και φυσικά το ανάποδο όταν θελήσετε στήλες).

Στη γραμμή 5 βλέπουμε ότι πολλές συναρτήσεις εξακολουθούν να δουλεύουν όταν έχουν μητρώα ως όρισμα.

Πλαίσια δεδομένων

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

1 > t = data.frame(x = c(11,12,14),

2 y = c(19,20,21), z = c(10,9,7))

3 > t

4 x y z

5 1 11 19 10

6 2 12 20 9

7 3 14 21 7

8 > mean(t$z)

9 [1] 8.666667

10 > mean(t[["z"]])

11 [1] 8.666667

Στις γραμμές 1-2 κατασκευάζεται ένα τυπικό πλαίσιο δεδομένων με όνομα t. Οι στήλες έχουν τα ονόματα x, y and z.

Στις γραμμές 8-11 βλέπουμε δύο τρόπους με τους οποίους μπορείτε να επιλέξετε τη στήλη με όνομα z από το πλαίσιο δεδομένων με όνομα t.

Παράδειγμα

Δημιουργήστε ένα σενάριο το οποίο θα κατασκευάζει τρία τυχαία (κανονικά) διανύσματα μήκους 100. Ονομάστε αυτά τα διανύσματα x1,

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

2. Λίστες - Γραφικά

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

```
1 > L = list(one=1, two=c(1,2),
2 five=seq(0, 1, length=5))
3 > L
4 $one
5 [1] 1
6 $two
7 [1] 1 2
8 $five
9 [1] 0.00 0.25 0.50 0.75 1.00
10 > names(L)
11 [1] "one" "two" "five"
12 > L$five + 10
13 [1] 10.00 10.25 10.50 10.75 11.00
```

Στις γραμμές 1-2 δημιουργείται μια λίστα μέσω της εισόδου ονομάτων και τιμών. Η λίστα εμφανίζεται επίσης και στο παράθυρο του χώρου εργασίας.

Στις γραμμές 3-9 φαίνεται μια τυπική εκτύπωση (μετά από πάτημα των L και ENTER).

Η γραμμή 10 δείχνει πώς μπορούμε να δούμε τι υπάρχει μέσα στη λίστα.

Η γραμμή 12 παρουσιάζει έναν τρόπο χρήσης των αριθμών.

Γραφικά

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

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

Εκατοντάδες τυχαίοι αριθμοί αναπαρίστανται γραφικά μέσω της σύνδεσης των σημείων με γραμμές (το σύμβολο μέσα σε εισαγωγικά μετά το type= είναι το γράμμα l, όχι ο αριθμός 1) σε χρυσό χρώμα. Ένα άλλο πολύ απλό παράδειγμα είναι το κλασικό στατιστικό γράφημα του ιστογράμματος, που δημιουργείται από την απλή εντολή > hist(rnorm(100))η οποία παράγει το γράφημα.

Οι ακόλουθες γραμμές δημιουργούν ένα γράφημα

χρησιμοποιώντας το πλαίσιο δεδομένων t που φτιάξαμε στο προηγούμενο ToDo:

1 plot(t$a, type="l", ylim=range(t),

2 lwd=3, col=rgb(1,0,0,0.3))

3 lines(t$b, type="s", lwd=2,

7

Histogram of rnorm(100)

rnorm(100)

Frequency

−3 −2 −1 0 1 2

0

5 10 15 20

Υπάρχουν πολλοί τρόποι για να καταγράψει κανείς δεδομένα σε αρχεία μέσω του περιβάλλοντος της R και για να διαβάσει δεδομένα από αρχεία. Εδώ θα

παρουσιάσουμε έναν τέτοιο τρόπο. Οι ακόλουθες γραμμές παρουσιάζουν τα στοιχειώδη:

```
1 > d = data.frame(a = c(3,4,5),

2 b = c(12,43,54))

3 > d

4 a b

5 1 3 12

6 2 4 43

7 3 5 54

8 > write.table(d, file="tst0.txt",

9 row.names=FALSE)

10 > d2 = read.table(file="tst0.txt",

11 header=TRUE)

12 > d2

13 a b

14 1 3 12

15 2 4 43

16 3 5 54
```

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

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

Στη γραμμή 8 καταγράφεται αυτό το πλαίσιο δεδομένων σε ένα αρχείο κειμένου, με όνομα tst0.txt

Το όρισμα row.names=FALSE αποτρέπει τα ονόματα των γραμμών να καταγραφούν στο αρχείο. Επειδή δεν καθορίζεται κάτι για τα col.names (ονόματα των στηλών), επιλέγεται η προκαθορισμένη επιλογή col.names=TRUE και τα ονόματα των στηλών καταγράφονται στο αρχείο.

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

Μη-διαθέσιμα δεδομένα

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

Όταν δουλεύετε με πραγματικά δεδομένα, θα βρεθείτε αντιμέτωποι με τιμές που λείπουν επειδή υπήρξαν αστοχίες στα όργανα μέτρησης ή επειδή δεν θέλατε να κάνετε μετρήσεις το Σαββατοκύριακο. Όταν ένα δεδομένο δεν είναι διαθέσιμο, τότε πρέπει να γράψετε NA (τα αρχικά του "Not Available") αντί κάποιου αριθμού. > j = c(1,2,NA) Ο υπολογισμός στατιστικών από ημιτελή σύνολα δεδομένων είναι αδύνατος, αυστηρά μιλώντας. Μπορεί η μέγιστη τιμή να εμφανίστηκε κατά τη διάρκεια του Σαββατοκύριακου, όταν δεν κάνατε μετρήσεις. Κατά συνέπεια, η R θα σας πει ότι δε γνωρίζει ποια είναι η μέγιστη τιμή του j:

> max(j)

[1] NA

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

> max(j, na.rm=TRUE)

[1] 2

3. Κλάσεις

Οι ασκήσεις που κάνατε πριν ήταν σχεδόν όλες με αριθμούς. Μερικές φορές θα χρειαστεί να προσδιορίσετε κάτι το οποίο δεν είναι αριθμός, για παράδειγμα το όνομα ενός σταθμού μετρήσεων ή ενός αρχείου δεδομένων. Σε αυτήν την περίπτωση θέλετε η μεταβλητή να είναι μια ακολουθία χαρακτήρων αντί να είναι αριθμός. Ένα αντικείμενο στην R μπορεί να έχει διάφορες από τις λεγόμενες κλάσεις. Οι τρεις πιο σημαντικές είναι η numeric, η character και η POSIX (συνδυασμοί ημερομηνίας-ώρας). Μπορείτε να ρωτήσετε την R ποια είναι η κλάση μιας συγκεκριμένης μεταβλητής πληκτρολογώντας class(...).

Χαρακτήρες

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

> m = "apples"

> m

[1] "apples"

> n = pears

Error: object `pears' not found

Φυσικά, δεν μπορείτε να κάνετε υπολογισμούς με ακολουθίες χαρακτήρων:

> m + 2

Error in m + 2 : non-numeric argument to

binary operator

Ημερομηνίες

Οι ημερομηνίες και οι ώρες είναι πολύπλοκες περιπτώσεις. Η R πρέπει να γνωρίζει ότι η ώρα 3

ακριβώς είναι μετά από τις 2:59 και ότι ο Φεβρουάριος έχει 29 ημέρες σε μερικά έτη. Ο ευκολότερος τρόπος για να πείτε στην R ότι κάτι αποτελεί συνδυασμό ημερομηνίας-ώρας είναι μέσω της συνάρτησης

strptime:

1 > date1=strptime(c("20100225230000",

2 "20100226000000", "20100226010000"),

3 format="%Y%m%d%H%M%S")

4 > date1

5 [1] "2010-02-25 23:00:00"

6 [2] "2010-02-26 00:00:00"

7 [3] "2010-02-26 01:00:00"

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

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

Προγραμματιστικά εργαλεία

Όταν φτιάχνετε ένα μεγαλύτερο πρόγραμμα από τα προηγούμενα παραδείγματα ή όταν χρησιμοποιείτε τα σενάρια κάποιου άλλου, μπορεί να συναντήσετε κάποιες προγραμματιστικές δομές.

Δομή ελέγχου if

Η δομή ελέγχου if χρησιμοποιείται όταν συγκεκριμένοι υπολογισμοί πρέπει να γίνουν μόνο όταν ικανοποιείται μια συγκεκριμένη συνθήκη (και πιθανόν να πρέπει να γίνει κάτι άλλο, όταν η συνθήκη δεν ικανοποιείται). Ένα παράδειγμα:

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

Στη γραμμή 2 καθορίζεται μια συνθήκη: το w πρέπει

να είναι μικρότερο του 5.

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

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

1 > a = c(1,2,3,4)

2 > b = c(5,6,7,8)

3 > f = a[b==5 | b==8]

4 > f

5 [1] 1 4

Στη γραμμή 1 και 2 δημιουργούνται δύο διανύσματα.

Στη γραμμή 3 δηλώνετε ότι η f συντίθεται από εκείνα τα στοιχεία του διανύσματος a για τα οποία τα αντίστοιχα στοιχεία του b ισούνται με 5 ή με 8. Προσέξτε το διπλό = στη συνθήκη. Άλλες συνθήκες (που ονομάζονται επίσης λογικοί ή Boolean τελεστές)

είναι οι <, >, != (≠), <= (≤) και >= (≥). Για να ελέγξετε περισσότερες από μία συνθήκες σε μία δομή if, χρησιμοποιήστε το & εάν και οι δύο συνθήκες πρέπει να ικανοποιούνται (λογικό «και» - "and") και το | εάν μία από τις συνθήκες πρέπει να ικανοποιείται (λογικό «ή» - "or").

Βρόχος επανάληψης for

Όταν θέλετε να μοντελοποιήσετε μια χρονοσειρά, συνήθως κάνετε τους υπολογισμούς για ένα χρονικό βήμα και έπειτα για το επόμενο και για το μεθεπόμενο. Επειδή κανένας δε θέλει να πληκτρολογεί τις ίδιες εντολές ξανά και ξανά, αυτοί οι υπολογισμοί μπορούν να αυτοματοποιηθούν με βρόχους επανάληψης for. Σε έναν βρόχο for προσδιορίζετε τι πρέπει να γίνει και πόσες φορές πρέπει να γίνει. Για να δηλώσετε το «πόσες φορές», καθορίζετε το λεγόμενο μετρητή. Ένα παράδειγμα:

1 > h = seq(from=1, to=8)

2 > s = c()

3 > for(i in 2:10)

4 {

5 s[i] = h[i] * 10

6 }

7 > s

8 [1] NA 20 30 40 50 60 70 80 NA NA

Αρχικά δημιουργείται το διάνυσμα h.

Στη γραμμή 2 δημιουργείται ένα κενό διάνυσμα (s). Αυτό είναι απαραίτητο γιατί όταν δημιουργείτε μια μεταβλητή μέσα σε βρόχο for, η R δε θα είναι σε θέση να τη θυμάται όταν βγει έξω από αυτόν.

Στη γραμμή 3 ξεκινάει ο βρόχος for. Σε αυτή την περίπτωση, το i είναι ο μετρητής και τρέχει από το 2 έως το 10.

Οτιδήποτε βρίσκεται ανάμεσα στις αγκύλες (γραμμή 5) θα υποστεί επεξεργασία 9 φορές. Την πρώτη φορά για i=2, το δεύτερο στοιχείο του h πολλαπλασιάζεται με 10 και τοποθετείται στη δεύτερη θέση του διανύσματος s. Την δεύτερη φορά για i=3, κτλ. Στις τελευταίες δύο εκτελέσεις, ζητούνται το 9ο και 10ο στοιχείο του h, τα οποία δεν υπάρχουν. Σημειώστε ότι αυτές οι δηλώσεις ελέγχονται και υπολογίζονται χωρίς να υπάρχει κάποιο ρητό μήνυμα λάθους.

Γράφοντας τις δικές σας συναρτήσεις

Οι συναρτήσεις που προγραμματίζετε εσείς οι ίδιοι λειτουργούν με τον ίδιο τρόπο που λειτουργούν οι 10 συναρτήσεις που είναι προ-προγραμματισμένες μέσα στην R.

```
1 > fun1 = function(arg1, arg2 )

2 {

3 w = arg1 ^ 2

4 return(arg2 + w)

5 }

6 > fun1(arg1 = 3, arg2 = 5)

7 [1] 14

8
```

Στη γραμμή 1 καθορίζεται το όνομα της συνάρτησης (fun1) και τα ορίσματά της (arg1 και arg2).

Στις γραμμές 2-5 καθορίζεται το τι θα πρέπει να κάνει η συνάρτηση όταν καλεστεί. Η τιμή που θα επιστρέφει (arg2+w) εμφανίζεται στην οθόνη.

Στη γραμμή 6 καλείται η συνάρτηση με ορίσματα 3 και 5.

4. Συναρτήσεις

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

Δημιουργία δεδομένων (data creation)

read.table: διάβασε έναν πίνακα από ένα αρχείο.

Ορίσματα: header=TRUE: διάβασε την πρώτη γραμμή

ως τίτλους των στηλών·sep=",": οι αριθμοί διαχωρίζονται με κόμμα·skip=n: μη διαβάσεις τις πρώτες n γραμμές.

write.table: αποθήκευσε έναν πίνακα σε ένα αρχείο

c: συνένωσε αριθμούς μεταξύ τους ώστε να δημιουργήσεις ένα διάνυσμα

array: δημιούργησε ένα διάνυσμα, Ορίσματα: dim: μήκος

matrix: δημιούργησε ένα μητρώο, Ορίσματα: ncol και/ή nrow: αριθμός γραμμών/ στηλών

data.frame: δημιούργησε ένα πλαίσιο δεδομένων

list: δημιούργησε μια λίστα

rbind και cbind: συνένωσε διανύσματα σε ένα μητρώο κατά γραμμή ή κατά στήλη

Εξαγωγή δεδομένων (extracting data)

• x[n]: το n-στο στοιχείο ενός διανύσματος

• x[m:n]: από το m-στο έως το n-στο στοιχείο

• x[c(k,m,n)]: συγκεκριμένα στοιχεία

• x[x>m & x<n]: τα στοιχεία ανάμεσα στο m και n

• x$n: το στοιχείο της λίστας ή του πλαισίου δεδομένων με όνομα n

• x[["n"]]: ό.π.

• [i,j]: το στοιχείο στην i-στη γραμμή και j-στη στήλη

• [i,]: η γραμμή i σε ένα μητρώο

Πληροφορίες για μεταβλητές

• length: το μήκος ενός διανύσματος

• ncol ή nrow: ο αριθμός των στηλών ή των γραμμών

ενός μητρώου

• class: η κλάση μιας μεταβλητής

• names: τα ονόματα των αντικειμένων σε μια λίστα

• print: εμφάνισε τη μεταβλητή ή την ακολουθία

χαρακτήρων στην οθόνη (χρησιμοποιείται σε σενάρια

ή σε βρόχους for)

• return: εμφάνισε τη μεταβλητή στην οθόνη (χρησιμοποιείται σε συναρτήσεις)

• is.na: έλεγξε εάν η μεταβλητή ισούται με NA

• as.numeric ή as.character: άλλαξε την κλάση σε

αριθμό ή σε ακολουθία γραμμάτων

• strptime: άλλαξε την κλάση από χαρακτήρα σε ημερομηνία-ώρα (POSIX)

Στατιστικά

sum: το άθροισμα ενός διανύσματος (ή ενός μητρώου)

mean: ο μέσος όρος ενός διανύσματος

sd: η τυπική απόκλιση ενός διανύσματος

max ή min: μέγιστο ή ελάχιστο στοιχείο

rowSums (ή rowMeans, colSums και colMeans): αθροίσματα (ή μέσοι όροι) όλων των αριθμών σε κάθε γραμμή (ή στήλη) ενός μητρώου. Το αποτέλεσμα είναι ένα διάνυσμα.

quantile(x,c(0.1,0.5)): δειγματοληψία του 0.1-στου και του 0.5-στου ποσοστημόριου του διανύσματος x

Επεξεργασία δεδομένων

seq: δημιούργησε ένα διάνυσμα με ίσες αποστάσεις

μεταξύ των αριθμών

rnorm: δημιούργησε ένα διάνυσμα με τυχαίους

αριθμούς που ακολουθούν την κανονική κατανομή

(και άλλες κατανομές είναι επίσης διαθέσιμες)

sort: ταξινόμησε τα στοιχεία σε αύξουσα διάταξη

t: ανάστρεψε ένα μητρώο

aggregate(x,by=ls(y),FUN="mean"): διαχώρισε το σύνολο δεδομένων x σε υποσύνολα (που καθορίζονται από το y) και υπολόγισε τους μέσους όρους των υποσυνόλων. Αποτέλεσμα: μια νέα λίστα.

na.approx: παρεμβολή (στο πακέτο zoo). Όρισμα: διάνυσμα με στοιχεία NA. Αποτέλεσμα: διάνυσμα χωρίς NA.

cumsum: σωρευτικό άθροισμα. Το αποτέλεσμα είναι ένα διάνυσμα.

rollmean: κινητός μέσος όρος (στο πακέτο zoo)

paste: συνένωση ακολουθιών χαρακτήρων μεταξύ τους

substr: εξαγωγή μέρους μιας ακολουθίας χαρακτήρων

Προσαρμογή (fitting)

lm(v1~v2): γραμμική προσαρμογή (γραμμή παλινδρόμησης) μεταξύ του διανύσματος v1 στον άξονα των y και του v2 στον άξονα των x

nls(v1~a+b*v2, start=ls(a=1,b=0)): μη γραμμική παρεμβολή. Πρέπει να περιλαμβάνει μια εξίσωση με μεταβλητές (εδώ είναι τα v1 και v2) και παραμέτρους (εδώ είναι τα a και b)με αρχικές τιμές.

coef: επιστρέφει τους συντελεστές μιας παρεμβολής

summary: επιστρέφει όλα τα αποτελέσματα από μια παρεμβολή

Σχεδίαση γραφικών παραστάσεων

plot(x): σχεδίαση του x (άξονας y) προς τον αριθμό

ευρετηρίου (άξονας x) σε ένα νέο παράθυρο

plot(x,y): σχεδίαση του y (άξονας y) έναντι του x (άξονας x) σε ένα νέο παράθυρο image(x,y,z): σχεδίαση του z (χρωματική κλίμακα) έναντι του x (άξονας x) και του y (άξονας y) σε ένα νέο παράθυρο

lines ή points: πρόσθεσε γραμμές ή σημεία σε μια προηγούμενη γραφική παράσταση

hist: σχεδίαση του ιστογράμματος των αριθμών ενός διανύσματος

barplot: ραβδόγραμμα ενός διανύσματος ή ενός πλαισίου δεδομένων

contour(x,y,z): σχεδίαση διαγράμματος ισοϋψών καμπυλών

abline: σχεδίαση γραμμής (τμήματος). Ορίσματα: a,b με σημείο τομής (σταθερός όρος) a και κλίση b; ή h=y για οριζόντια γραμμή στο y; ή v=x για κάθετη γραμμή στο x.

curve: εισάγετε συνάρτηση για σχεδίαση. Χρειάζεται ένα x στην έκφραση.

Παράδειγμα: curve(x^2)

legend: προσθήκη υπομνήματος με δεδομένα

σύμβολα (lty ή pch και col) και κείμενο (legend) στο σημείο (x="topright")

axis: προσθήκη άξονα. Ορίσματα: side – 1=κάτω, 2=αριστερά, 3=πάνω, 4=δεξιά

mtext: προσθήκη κειμένου στον άξονα. Ορίσματα: text (ακολουθία χαρακτήρων) και side

grid: προσθήκη πλέγματος

par: παράμετροι σχεδιασμού που πρέπει να προσδιοριστούν πριν τις γραφικές παραστάσεις. Ορίσματα: π.χ. mfrow=c(1,3)): αριθμός των γραφημάτων ανά σελιδα

Παράμετροι γραφικών παραστάσεων

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

κτλ.

pch: τύπος σημείου – 1=κύκλος, 2=τρίγωνο, κτλ.

main: τίτλος - ακολουθία χαρακτήρων

xlab και ylab: ετικέτες αξόνων – ακολουθίες χαρακτήρων

xlim και ylim: εύρος των αξόνων – π.χ. c(1,10)

log: λογαριθμικός άξονας – "x", "y" ή "xy"

5. Προγραμματισμός

function(arglist){expr}: ορισμός συνάρτησης:

εκτέλεσε την έκφραση expr με αυτή τη λίστα ορισμάτων arglist

if(cond){expr1}else{expr2}: δομή ελέγχου if: εάν η συνθήκη cond αληθεύει, τότε expr1, αλλιώς expr2

for(var in vec) {expr}: βρόχος επανάληψης for:

Ο μετρητής var διατρέχει το διάνυσμα vec και εκτελει την έκφραση expr σε κάθε επανάληψη while(cond){expr}: βρόχος επανάληψης while: όσο η συνθήκη cond αληθεύει, εκτέλεσε την έκφραση expr σε κάθε επανάληψη

6. Συντομεύσεις πληκτρολογίου

Υπάρχουν αρκετές χρήσιμες συντομεύσεις πληκτρολογίου για το RStudio (βλ. Help → Keyboard Shortcuts): CRL+ENTER: στείλε τις εντολές από το παράθυρο σεναρίου στο παράθυρο εντολών ↑ ή ↓ στο παράθυρο εντολών: προηγούμενη ή επόμενη εντολή CTRL+1, CTRL+2, κτλ.: εναλλαγή μεταξύ των παραθύρων. Αν και δεν είναι συγκεκριμένες για την R, αποτελούν πολύ χρήσιμες συντομεύσεις: CTRL+C, CTRL+X και CTRL+V: αντιγραφή, αποκοπή και επικόλληση ALT+TAB: μετάβαση σε παράθυρο άλλου προγράμματος ↑, ↓, ← ή →: μετακίνηση του κέρσορα HOME ή END: μετακίνηση του κέρσορα στην αρχή ή στο τέλος της γραμμής Page Up ή Page Down: μετακίνηση του κέρσορα μια σελίδα πιο πάνω ή πιο κάτω SHIFT+↑/↓/←/→/HOME/END/PgUp/PgDn: επιλογή

Μηνύματα σφάλματος

No such file or directory ή Cannot change working directory

Σιγουρευτείτε ότι ο κατάλογος εργασίας και τα ονόματα των αρχείων είναι σωστά.

Object 'x' not found Η μεταβλητή x δεν έχει οριστεί ακόμα. Ορίστε την x ή προσθέστε αποστρόφους εάν η x πρέπει να είναι ακολουθία χαρακτήρων.

Argument 'x' is missing without default Δεν έχετε προσδιορίσει το υποχρεωτικό όρισμα x. Η R είναι ακόμα απασχολημένη με κάτι ή έχετε ξεχάσει να κλείσετε κάποια παρένθεση ή αγκύλη. Περιμένετε, πληκτρολογήστε } ή) ή πιέστε το πλήκτρο ESC.

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

Unexpected 'else' in "else" Βάλτε το else μιας δομής if στην ίδια γραμμή με την τελευταία αγκύλη του κομματιού "then": }else{. Missing value where TRUE/FALSE needed Κάτι πάει στραβά στο κομμάτι της συνθήκης (if(x==1)) μιας δομής if. Μήπως το x είναι NA; The condition has length > 1 and only the first element will be used Στο μέρος της συνθήκης (if(x==1)) μιας δομής if, ένα διάνυσμα συγκρίνεται με ένα βαθμωτό. Μήπως το x είναι διάνυσμα; Μήπως εννοούσατε x[i];

Non-numeric argument to binary operator. Προσπαθείτε να κάνετε υπολογισμούς με κάτι το οποίο δεν είναι αριθμός. Χρησιμοποιείστε την class(...) ώστε να βρείτε τι είναι αυτό που πήγε στραβά ή χρησιμοποιείστε την as.numeric(...) για να μετατρέψετε τη μεταβλητή σε αριθμό.Argument is of length zero ή Replacement is of length zero. Η εν λόγω μεταβλητή είναι ίση με NULL, το οποίο σημαίνει ότι είναι κενή, για παράδειγμα δημιουργήθηκε από την c(). Ελέγξτε τον ορισμό της μεταβλητής.

x <- 1:2 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 παλινδρόμηση της y ως προς x

summary(fm) και παρουσιάζει τα αποτελέσματα

fm1 <- lm(y~x, data=dummy, εφαρμόζει σταθμισμένη παλινδρόμηση.

weight=1/w^2)

lrf <- loess(y~x, data=dummy) Κάνει απαραμετρική παλινδρόμηση.

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)

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.

7. Παράδειγμα - Ασκήσεις

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 (−π, π).

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 f a είναι το ασσύμετρο κομμάτι της f.

contour(x,y, fa, nlevels=15) Δημιουργεί το διάγραμμα ισουψών της f a.

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)

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
```

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

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

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
```

Πράξεις Διανυσμάτων και Πινάκων

΄Οπως 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

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

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,

2x + 3y = 13

x − 2y = −4

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

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.

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 −Y 10 cos 2y. Με αυτόν τον τρόπο δουλεύουμε συνήθως όταν θέλουμε να δημιουργήσουμε γραφικές παραστάσεις συναρτήσεων. Τα αντίστοιχα γραφήματα παρουσιάζονται.

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

Ακολουθεί ένα παράδειγμα για το πως χρησιμοποιούνται τα πιο πάνω. 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))
```

΄Οταν 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).

8. 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

Λογικοί τελεστές και τελεστές σύγκρισης.

Οι τελεστές ελέγχου χρησιμοποιούνται για να κατασκευάζονται υποθετικές προτάσεις.

Χρησιμοποιώντας Υποσύνολα των Δεδομένων

Αρκετές φορές υπάρχει η ανάγκη να γίνουν διάφοροι υπολογισμοί χρησιμοποιώντας ένα συγκεκριμένο κομμάτι των δεδομένων. Η μέθοδος αυτή ονομάζεται υπόστιξη. 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

[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

Επιλέγοντας υποσύνολα δεδομένων στην R

Έστω πως έχουμε ένα μεγάλο σύνολο δεδομένων. πως μπορούμε να επιλέξουμε ένα υποσύνολο από αυτά;

Για παράδειγμα, έστω ο παρακάτω πίνακας δεδομένων:

x1 <- sample(1:10, 50, replace=T)

x2 <- rnorm(50)

x <- data.frame(x1, x2)

Μερικές ερωτήσεις-απαντήσεις:

Να βρεθεί η στήλη x1 με τιμές μεγαλύτερες του 5

subset(x, select=x1, x1>5)

Να βρεθεί η στήλη x2 εκεί όπου η στήλη x1 έχει τιμές μεγαλύτερες του 5

subset(x, select=x2, x1>5)

Να βρεθούν οι στήλες x1,x2 εκεί όπου η στήλη x1 έχει τιμές μεγαλύτερες του 5

subset(x, select=c(x1,x2), x1>5)

Να βρεθεί η στήλη x2 εκεί όπου η στήλη x1 έχει τιμή 3

subset(x, select=x2, x1==3)

Να βρεθεί η στήλη x2 εκεί όπου η στήλη x1 έχει τιμή 3 ή 6

subset(x, select=x2, x1==3 | x1==6)

Να βρεθεί οι στήλες x1,x2 εκεί όπου η στήλη x2 έχει τιμές (45,50]

subset(x, select=c(x1,x2), x2>45 & x2<=50)

9. Άθροισμα συνδυασμών με την R

Μπορούμε να υπολογίσουμε το πλήθος των συνδυασμών με τη χρήση του παραγοντικού**. Για παράδειγμα**:

> n <- 3

> k <- 2

> factorial(n) / (factorial(k) * factorial (n-k))

[1] 3

Εναλλακτικά μπορούμε να χρησιμοποιήσουμε τη συνάρτηση choose:

> choose(3,2)

[1] 3

άθροισμα συνδυασμών

Δείτε και ένα άλλο ερώτημα. Αν και , πόσους αθροιστικά συνδυασμούς μπορούμε να κάνουμε;

Αν :

> choose(4,1)

[1] 4

Αν :

```
> choose(4,2)

[1] 6
```

Αν :

```
> choose(4,3)

[1] 4
```

Αν :

```
> choose(4,4)

[1] 1
```

Οπότε, αθροιστικά:

```
> choose(4,1) + choose(4,2) + choose(4,3) + choose(4,4)

[1] 15
```

Δηλαδή, επιλέγοντας από 1 ως 4 στοιχεία από ένα σύνολο 4 στοιχείων μπορούμε να κάνουμε 15 συνδυασμούς.

Επίσης, μια περισσότερο γενική -και προγραμματιστική- λύση είναι αυτή:

```
n <- 4

s <- c()

for (k in 1:n)

{

   s[k] <- choose(n,k)}

}

sum(s)
```

Έλεγχος του μέσου με το t.test με την R

Έστω ένα δείγμα από το βάρος (Kg) και το ύψος (m) 8 αντρών:

```
weight <- c (65, 72, 81, 79, 67, 78, 76, 70)

height <- c (1.65, 1.70, 1.76, 1.62, 1.81, 1.82, 1.77, 1.82)
```

Ορίζουμε το σωματομετρικό δείκτη bmi (body mass index) ως:

```
bmi <- weight/height^2

bmi

[1] 23.87511 24.91349 26.14928 30.10212 20.45115 23.54788 24.25867 21.13271
```

Το ερώτημα είναι αν ο μέσος του bmi είναι 22.5, όπως προβλέπει σχετική έρευνα για το σωματότυπο υγιών ανδρών με κανονικό βάρος και ύψος.

Ο μέσος μπορεί να υπολογιστεί ως :

```
> mean(bmi)

[1] 24.3038
```

Η τιμή 24.3 είναι βέβαια αλγεβρικά διαφορετική από την τιμή 22.5, αλλά το ερώτημα είναι αν είναι στατιστικά σημαντική η διαφορά 24.3-22.5 από το μηδέν.

Για τον έλεγχο του μέσου θα κάνουμε το t-test ως εξής:

: Ο μέσος είναι 22.5

: Ο μέσος είναι διαφορετικός από το 22.5

```
t.test(bmi, mu=22.5)

    One Sample t-test

data:  bmi

t = 1.6999, df = 7, p-value = 0.1329

alternative hypothesis: true mean is not equal to 22.5

95 percent confidence interval:

 21.79467 26.81294

sample estimates:

mean of x

  24.3038
```

Η ερμηνεία μπορεί να γίνει εύκολα με βάση την τιμή του p-value: Η τιμή 0.1329 μας λέει πως δεν απορρίπτεται η μηδενική υπόθεση σε επίπεδο σημαντικότητας μικρότερο του 13.29%. Για παράδειγμα, δεν απορρίπτουμε τη μηδενική υπόθεση σε επίπεδο σημαντικότητας 5% επειδή 0.1329 > 0.05. Το διάστημα εμπιστοσύνης του μέσου, σε επίπεδο σημαντικότητας 5%, είναι 21.79467-26.81294. Αν θέλουμε, μπορούμε να αλλάξουμε το επίπεδο σημαντικότητας του υπολογισμού, ως:

```
t.test(bmi, mu=22.5, conf.level=0.99)
```

Βιβλιογραφία

Anany Levitin, Εισαγωγή στην Ανάλυση και Σχεδίαση Αλγορίθμων, 2η έκδοση.

Αθανάσιος Παπούλης, Πιθανότητες τυχαίες μεταβλητές και στοχαστικές διαδικασίες 4η έκδοση, Εκδόσεις Τζιόλα, 2007.

S. Sahli, Δομές δεδομένων Αλγόριθμοι και Εφαρμογές στη C++, Εκδόσεις Τζιόλα, 2004.

C.L. Liu, Στοιχεία Διακριτών Μαθηματικών, Πανεπιστημιακές εκδόσεις Κρήτης, 2012.

Πηγές (Χρήσιμες ιστοσελίδες)

https://cran.r-project.org/doc/contrib/mainfokianoscharalambous.pdf

http://www.math.ntua.gr/~fouskakis/Data_Analysis/02.pdf

http://www.r-tutor.com/

https://www.statmethods.net/

http://www.cyclismo.org/tutorial/R/index.htmlhttp://stavrakoudis.econ.uoi.gr/stavrakoudis/?iid=214