



ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΣΧΟΛΗ ΘΕΤΙΚΩΝ
ΕΠΙΣΤΗΜΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ

Μεταπτυχιακή Εξειδίκευση στα Πληροφοριακά Συστήματα

Διπλωματική Εργασία

Δημιουργία εκπαιδευτικού υλικού για την εκμάθηση της γλώσσας
προγραμματισμού R

Κυριακή Κατσάνη

Επιβλέπων καθηγητής: Γεώργιος Παπαμιχαήλ

Πάτρα, Ιούνιος 2021

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή (Κυριακή Κατσάνη) που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης ο συγγραφέας/δημιουργός εκχωρεί στο ΕΑΠ, μη αποκλειστική άδεια χρήσης του δικαιώματος αναπαραγωγής, προσαρμογής, δημόσιου δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσής τους διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος και για όλο το χρόνο διάρκειας των δικαιωμάτων πνευματικής ιδιοκτησίας. Η ανοικτή πρόσβαση στο πλήρες κείμενο για μελέτη και ανάγνωση δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, αποθήκευση, πώληση, εμπορική χρήση, μετάδοση, διανομή, έκδοση, εκτέλεση, «μεταφόρτωση» (downloading), «ανάρτηση» (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού. Ο συγγραφέας/δημιουργός διατηρεί το σύνολο των ηθικών και περιουσιακών του δικαιωμάτων.



Δημιουργία εκπαιδευτικού υλικού για την εκμάθηση της γλώσσας προγραμματισμού R

Κυριακή Κατσάνη

Επιτροπή Επίβλεψης Διπλωματικής Εργασίας

Επιβλέπων Καθηγητής:

Γεώργιος Παπαμιχαήλ

ΣΕΠ, Ελληνικό Ανοικτό Πανεπιστήμιο

Σχολή Θετικών Επιστημών και

Τεχνολογίας

Συν-Επιβλέπων Καθηγητής:

Δημήτρης Καλλές

Καθηγητής, Ελληνικό Ανοικτό

Πανεπιστήμιο

Σχολή Θετικών Επιστημών και

Τεχνολογίας

Πάτρα, Ιούνιος 2021

*Αφιερωμένη στους γονείς μου Παναγιώτη και Αλεξάνδρα που μου συμπαραστέκονται
και με εμπνέουν σε κάθε μου βήμα.*

Περίληψη

Η παρούσα διπλωματική εργασία στοχεύει στην δημιουργία εκπαιδευτικού υλικού για την εκμάθηση της γλώσσας προγραμματισμού R. Γι' αυτό το λόγο έχουν σχεδιαστεί 10 μαθήματα τα οποία παρέχουν όλα όσα χρειάζεται κάποιος για να ξεκινήσει την εκμάθηση της γλώσσας προγραμματισμού R, με επέκταση σε βασικές εργασίες της επιστήμης των δεδομένων. Στα μαθήματα περιλαμβάνονται εργαλεία και βιβλιοθήκες για Data Science μέσω της R και συνοδεύονται επίσης από υλικό ασκήσεων και βιντεοσκοπημένα μαθήματα. Τα μαθήματα απευθύνονται σε κάθε ενδιαφερόμενο που θέλει να αποκτήσει ένα γνωστικό υπόβαθρο σχετικό με την Επιστήμη των Δεδομένων και τη γλώσσα προγραμματισμού R.

Η ΔΕ καθώς και το αρχείο μαθημάτων έχουν υλοποιηθεί σε R - Markdown στο περιβάλλον του R- Studio και στην συνέχεια έγιναν convert σε word και HTML. Στο HTML αρχείο (υλοποιημένο εξ ολοκλήρου με R- Markdown) υπάρχουν διαθέσιμα 10 αναλυτικά μαθήματα με παραδείγματα, που συνοδεύονται από 18 βιντεοσκοπημένα μαθήματα και quiz πολλαπλών επιλογών. Είναι αναρτημένα και διαθέσιμα στο <https://github.com/KatsaniK/DIPLOMATIKI> .

Λέξεις – Κλειδιά

R, R προγραμματισμός, RStudio, Rmarkdown, ggplot2, R μαθήματα

Creation educational material for learning the R programming language

KYRIAKI KATSANI

Abstract

This thesis aims to create educational material for learning the R programming language. For this reason, 10 courses have been designed that provide everything someone needs to start learning the R programming language, extending to basic science work of the data. The courses include tools and libraries for Data Science via R and are also accompanied by exercise material and video courses. The courses are aimed at anyone interested in acquiring a knowledge background related to Data Science and the R programming language.

The thesis aims as well and the course file have been implemented in R - Markdown in the R-Studio environment and then converted to word and HTML. In the HTML file (implemented entirely with R-Markdown) there are 10 detailed tutorials with examples, 18 video lessons and multiple choice quizzes. They are posted and available at <https://github.com/KatsaniK/DIPLOMATIKI>.

Keywords

R, R programming, RStudio, Rmarkdown, ggplot2, R lessons

Περιεχόμενα

Περίληψη.....	2
Abstract	4
1 Ιστορική Αναδρομή και επισκόπηση της R	11
1.1 Σύντομη ιστορική Αναδρομή	11
1.2 Γιατί να χρησιμοποιήσω την R για στατιστικούς υπολογισμούς και γραφικά.....	11
1.3 Τι είναι η R	12
1.4 Εφαρμογές	13
1.5 Η γλώσσα προγραμματισμού R.....	13
2 Μαθήματα	16
2.1 Μάθημα 1ο	16
2.1.1 Εγκατάσταση της R στα Windows.....	16
2.1.2 Εγκατάσταση του R Studio	17
2.1.3 R Studio – Cloud.....	19
2.1.4 Το πρώτο μας πρόγραμμα σε R.....	19
2.2 Μάθημα 2ο	22
2.2.1 Μαθηματικές Πράξεις.....	22
2.2.2 Μεταβλητές και κανόνες.....	23
2.2.3 Βασικές κατηγορίες αντικειμένων «Atomic» Classes.....	24
2.3 Μάθημα 3ο	26
2.3.1 Δομές επιλογής.....	26
2.3.2 Δομές επανάληψης.....	29
2.4 Μάθημα 4ο	33
2.4.1 Διάνυσμα - Vector.....	33
2.4.2 Λίστα List.....	34
2.4.3 Πίνακας Matrix	36

2.4.4	Παράγοντες Factors	42
2.4.5	Πολυδιάστατοι πίνακες Arrays	44
2.4.6	Πλαίσια δεδομένων Data frames.....	46
2.5	Μάθημα 5ο	51
2.5.1	R - Πακέτα	51
2.5.2	Function στην R	52
2.6	Μάθημα 6ο	55
2.6.1	import R - csv.....	55
2.6.2	import R - xls	58
2.6.3	import R - mySQL	67
2.6.4	import R json.....	69
2.7	Μάθημα 7ο	71
2.7.1	Διάγραμμα πίτας	71
2.7.2	Γραφήματα ράβδων.....	73
2.7.3	Boxplots	75
2.7.4	Ιστογράμμο - Histograms.....	77
2.8	Μάθημα 8ο	79
2.8.1	Οπτικοποίηση δεδομένων	79
2.9	Μάθημα 9ο	98
2.9.1	R - Μέσος όρος, διάμεσος.....	98
2.9.2	R - Γραμμική παλινδρόμηση - R Linear Regression	100
2.9.3	R - Κανονική κατανομή	103
2.9.4	R - Δέντρο απόφασης.....	108
2.10	Μάθημα 10ο.....	113
2.10.1	Markdown	113
2.10.2	Βασικά στοιχεία του Markdown	113

2.10.3	Βιβλιογραφίες και παραπομπές.....	114
2.10.4	Εγκατάσταση R Markdown	115
3	Quiz.....	117
3.1.1	Quiz 1	117
3.1.2	Quiz 2	118
3.1.3	Quiz 3	119
4	Δημιουργία HTML αρχείου	121
4.1	Πίνακας περιεχομένων	121
4.1.1	Αρίθμηση τμημάτων	123
4.1.2	Ενότητες σε καρτέλες.....	123
4.1.3	Εμφάνιση και στυλ.....	124
4.1.4	Το πακέτο rmdformats	124
4.1.5	Εικόνες ΔΕ από R studio.....	125
5	Επίλογος.....	128
6	Βιβλιογραφικές Αναφορές	129

Κατάλογος Εικόνων / Σχημάτων

Εικόνα 1 Κορυφαίες γλώσσες προγραμματισμού το 2020 σύμφωνα με το [IEEE](https://www.ieee.org)

Εικόνα 2 Επιλογή mirror από το www.R-project.org εγκατάστασης

Εικόνα 3 Επιλογή Εγκατάστασης σε Windows από www.R-project.org

Εικόνα 4 Κύριο menu R Studio

Εικόνα 5 Κλασική προβολή R studio

Εικόνα 6 Παραγόμενο γράφημα ράβδων

Εικόνα 7 Παραγόμενο γράφημα ράβδων

Εικόνα 8 Διάγραμμα ροής If .. Else

Εικόνα 9 Διάγραμμα ροής While

Εικόνα 10 Διάγραμμα πίτας «Πόλεις»

Εικόνα 11 Διάγραμμα πίτας «Πόλεις» με λεζάντα

Εικόνα 12 Διάγραμμα ράβδων «Revenue chart»

Εικόνα 13 Διάγραμμα ράβδων «Total Revenue chart»

Εικόνα 14 Διάγραμμα Boxplot «Mileage Data»

Εικόνα 15 Γράφημα Ιστογράμματος

Εικόνα 16 Τα χαρακτηριστικά της ggplot2

Εικόνα 17 Διάγραμμα point με ggplot «Σχέση μεγέθους κινητήρα και απόδοσης καυσίμου»

Εικόνα 18 Διάγραμμα point με ggplot «Σχέση μεγέθους κινητήρα, απόδοσης καυσίμου και κατηγορίας αυτοκινήτου»

Εικόνα 19 Διάγραμμα point με ggplot «Σχέση μεγέθους κινητήρα, απόδοσης καυσίμου και κατηγορίας αυτοκινήτου» Απεικόνιση κατηγορίας αυτοκινήτου με μέγεθος point

Εικόνα 20 Διάγραμμα point με ggplot «Σχέση μεγέθους κινητήρα, απόδοσης καυσίμου και κατηγορίας αυτοκινήτου» Απεικόνιση κατηγορίας αυτοκινήτου με όρισμα alpha

Εικόνα 21 Διάγραμμα point με ggplot «Σχέση μεγέθους κινητήρα, απόδοσης καυσίμου και κατηγορίας αυτοκινήτου» Απεικόνιση κατηγορίας αυτοκινήτου με σχήμα

Εικόνα 22 Διάγραμμα point με ggplot, διερεύνηση συντακτικών προβλημάτων

Εικόνα 23 Διάγραμμα point με ggplot κατηγορηματικές και συνεχείς μεταβλητές

Εικόνα 24 Διάγραμμα point με ggplot πολλαπλή αισθητική

Εικόνα 25 Διάγραμμα point με ggplot με stroke

Εικόνα 26 Διάγραμμα point με ggplot χαρτογράφηση με κριτήρια

Εικόνα 27 Όψεις 2 μεταβλητές

Εικόνα 28 Όψεις 1 μεταβλητή

Εικόνα 29 Όψεις 1 μεταβλητή

Εικόνα 30 Διάγραμμα point με ggplot. «Σχέση μεγέθους κινητήρα, απόδοσης καυσίμου»

Εικόνα 31 Διάγραμμα point με ggplot. «Σχέση μεγέθους κινητήρα, απόδοσης καυσίμου»
smooth

Εικόνα 32 Διάγραμμα point με ggplot. «Σχέση μεγέθους κινητήρα, απόδοσης καυσίμου» με
linetype

Εικόνα 33 Εισαγωγή πολλαπλών geoms

Εικόνα 34 Εισαγωγή πολλαπλών geoms

Εικόνα 35 Γραφική παράσταση Πρόβλεψης Βάρους – Ύψους

Εικόνα 36 Γραφική παράσταση Πρόβλεψης Βάρους – Ύψους με ggplot2

Εικόνα 37 Κανονική κατανομή dnorm

Εικόνα 38 Ύψος κανονικής κατανομής rnorm

Εικόνα 39 Γραφική παράσταση qnorm

Εικόνα 40 Γραφική παράσταση rnorm

Εικόνα 41 Κατανομή με ggplot

Εικόνα 42 Δένδρο απόφασης

Εικόνα 43 Δένδρο απόφασης με ggplot

Εικόνα 44 Ξεχωριστά tabsets σε HTML

Συντομογραφίες & Ακρωνύμια

Ακολουθούν κάποια παραδείγματα:

ΔΕ Διπλωματική Εργασία ΕΑΠ Ελληνικό Ανοικτό Πανεπιστήμιο

ΘΕ Θεματική Ενότητα

ΠΕ Πτυχιακή Εργασία

ΠΣ Πρόγραμμα Σπουδών

ΣΥΝ Συντονιστής

1 Ιστορική Αναδρομή και επισκόπηση της R

Υπάρχουν δύο είδη γλωσσών προγραμματισμού, αυτές που ο κόσμος παραπονιέται και αυτές που δεν τις χρησιμοποιεί κανείς. – Bjarne Stroustrup

1.1 Σύντομη ιστορική Αναδρομή

Η 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. Αρχικά το λογισμικό ονομαζόταν S και αναπτύχθηκε στα εργαστήρια της AT&T στο New Jersey. Η γλώσσα προγραμματισμού S μετατράπηκε σε S- Plus και έγινε εμπορικό πακέτο που ουσιαστικά είχε διαμορφωθεί κατά το πρότυπο της γλώσσας S για στατιστικούς υπολογισμούς. Σχεδιάστηκε από τον John Chambers, Rick Becker, Trevor Hastie, Allan Wilks και άλλους στα εργαστήρια της Bell στα μέσα της δεκαετίας του 1970 και έγινε δημόσια διαθέσιμη στις αρχές της δεκαετίας του 1980. Το 1994, ο Ross Ihaka και ο Robert Gentleman στο Πανεπιστήμιο Όκλαντ έγραψαν την πρώτη έκδοση της S σαν πακέτο λογισμικού και την ονόμασαν “R”, συνεχίζοντας την παράδοση της επιστήμης των υπολογιστών. Ο Robert και ο Ross καθιέρωσαν την R ως γλώσσα ανοικτού κώδικα το 1995. Το 1997 συντάσσετε η ομάδα ανάπτυξής της R και το Φεβρουάριου του 2000 κυκλοφορεί η πρώτη έκδοση R 1.0.0.

Έκαναν το λογισμικό αυτό ελεύθερα διαθέσιμο και με αυτή η κίνηση τους αποδείχθηκε πως είχαν πιάσει το πνεύμα και άλλων προγραμματιστών (Luke Tierney είχε αναπτύξει την Lisp-Stat, ο Martin Maechler είχε γράψει Emacs) με τους οποίους 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 και είναι ελεύθερα διαθέσιμο. (R programming language, χ.χ)

1.2 Γιατί να χρησιμοποιήσω την R για στατιστικούς υπολογισμούς και γραφικά

1. **Η R είναι ανοιχτού κώδικα και δωρεάν.** Η R είναι δωρεάν και διαθέτει άδεια σύμφωνα με τους όρους της GNU General Public License. Τα περισσότερα πακέτα

R είναι διαθέσιμα με την ίδια άδεια, ώστε να μπορούν να χρησιμοποιηθούν απόφια, ακόμη και σε εμπορικές εφαρμογές.

2. **Η R είναι δημοφιλής**. Το [IEEE](#) δημοσιεύει κάθε χρόνο μια λίστα με τις πιο δημοφιλείς γλώσσες προγραμματισμού. Η R κατέχει μία από τις 10 πρώτες θέσεις της λίστας. Είναι πολύ σημαντικό για μια γλώσσα που χρησιμοποιείτε μόνο για συγκεκριμένο σκοπό να είναι τόσο δημοφιλής. Αυτό δεν δείχνει μόνο το αυξανόμενο ενδιαφέρον για την R ως γλώσσα προγραμματισμού, αλλά και για τα πεδία της Επιστήμης των Δεδομένων και της Μηχανικής Μάθηση όπου η R χρησιμοποιείται συνήθως.

Κορυφαίες γλώσσες προγραμματισμού το 2020 σύμφωνα με το [IEEE](#)

Rank	Language	Type	Score
1	Python	🔧 📄 📊	100.0
2	Java	🔧 📄 📊	95.3
3	C	📄 📊 📈	94.6
4	C++	📄 📊 📈	87.0
5	JavaScript	🔧	79.5
6	R	📄	78.6
7	Arduino	🔧	73.2
8	Go	🔧 📄	73.1
9	Swift	📄 📊	70.5
10	Matlab	📄	68.4

Εικόνα 1

3. **Η R τρέχει σε όλες τις πλατφόρμες**. Η R υποστηρίζει όλες τις δημοφιλείς πλατφόρμες - Windows, Linux και Mac. Ο κώδικας R που γράφετε σε μια πλατφόρμα μπορεί εύκολα να μεταφερθεί σε άλλη χωρίς προβλήματα. Η διαλειτουργικότητα μεταξύ πλατφορμών είναι ένα σημαντικό χαρακτηριστικό που υπάρχει στον σημερινό κόσμο των υπολογιστών. (Roger D. Peng, 2015)

1.3 Τι είναι η R

Η R είναι ουσιαστικά μια γλώσσα προγραμματισμού και ένα περιβάλλον. Προσφέρει ένα ολοκληρωμένο σύνολο υπηρεσιών λογισμικού για ανάλυση δεδομένων, υπολογισμών και γραφημάτων. Αν και χρησιμοποιείται κυρίως στη στατιστική οι δημιουργοί της προτιμούν να το αποκαλούν εργαλείο για ανάλυση δεδομένων τονίζοντας ότι περιλαμβάνει

και μοντέρνες και παλιές στατιστικές μεθοδολογίες (Φωκιανός & Χαραλάμπους, 2010 Ιανουάριος) Η Ιστοσελίδα της R είναι η <http://www.r-project.org> και αποτελεί την κύρια πηγή πληροφόρησης.

1.4 Εφαρμογές

Η R χρησιμοποιείται για την υλοποίηση στατιστικών μεθόδων, κατασκευή γραφημάτων προσομοίωσης, ανάλυσης διασποράς και παλινδρόμησης. Όλα μπορούμε να τα εντάξουμε στο πλαίσιο της ανάλυσης δεδομένων. Η R όμως μπορεί να χρησιμοποιηθεί επιπλέον τόσο στην εξόρυξη με χρήση πακέτων όπως όσο και στην συγγραφή συγγραμμάτων.

Οι εφαρμογές της R αφορούν πλέον ένα ευρύ φάσμα πεδίων όπως ιατρική, οικονομία, τηλεπικοινωνίες. (Βερύκιος Σ Βασίλειος, Καγκλής Βασίλειος, Σταυρόπουλος Κ. Ηλίας, (2015)).

1.5 Η γλώσσα προγραμματισμού R

Η παρούσα διπλωματική εργασία φιλοδοξεί να είναι ένα εγχειρίδιο εκμάθησης της γλώσσας προγραμματισμού R. Για το σκοπό αυτό έχουν υλοποιηθεί 10 μαθήματα τα οποία ξεκινούν τον αναγνώστη από το σημείο μηδέν ως την πλήρη εικόνα προγραμματισμού σε R.

Σε κάθε μάθημα έχουν συμπεριληφθεί στοιχεία θεωρίας αλλά και ασκήσεις που δίνονται με σκοπό την πρακτική εφαρμογή κώδικα σε R, καθώς και σχετικό βιντεοσκοπημένο μάθημα. Στο τέλος των μαθημάτων διατίθενται quiz πολλαπλών επιλογών.

Το R studio με την χρήση του R Markdown έχει την δυνατότητα εξαγωγής σε διάφορα formats όπως word, power point, HTML. Τόσο τα κείμενα της εργασίας όσο και το συνοδευτικό HTML file με τα μαθήματα έχουν δημιουργηθεί στο περιβάλλον του R studio με την χρήση του πακέτου R markdown.

Τα βιντεοσκοπημένα μαθήματα είναι ανεβασμένα στο vimeo και τα αντίστοιχα link τους είναι:

Μάθημα 1ο

Εγκατάσταση της R <https://vimeo.com/manage/videos/544518680>

Εγκατάσταση R studio <https://vimeo.com/manage/videos/544502208>

Το πρώτο μας πρόγραμμα σε R <https://vimeo.com/manage/videos/544508633>

Μάθημα 2ο

Μαθηματικές πράξεις, μεταβλητές κανόνες <https://vimeo.com/manage/videos/544516293>

Μάθημα 3°

Δομές επιλογής <https://vimeo.com/manage/videos/570171874>

Δομές επανάληψης <https://vimeo.com/manage/videos/570544672>

Μάθημα 4°

Διάνυσμα <https://vimeo.com/manage/videos/575550907>

Λίστα <https://vimeo.com/manage/videos/575552170>

Πίνακας <https://vimeo.com/manage/videos/576106316>

Παράγοντες <https://vimeo.com/manage/videos/576112564>

Πολυδιάστατοι πίνακες <https://vimeo.com/manage/videos/576137400>

Πλαίσια δεδομένων <https://vimeo.com/manage/videos/576142356>

Μάθημα 5°

Packages και function <https://vimeo.com/manage/videos/576151445>

Μάθημα 6°

Εισαγωγή και εξαγωγή αρχείων csv <https://vimeo.com/manage/videos/585225011>

Εισαγωγή και εξαγωγή αρχείων excel <https://vimeo.com/manage/videos/585270041>

Εργασίες στην MySql με το Rmysql <https://vimeo.com/manage/videos/585274047>

Μάθημα 7°

Διαγράμματα και Γραφήματα <https://vimeo.com/manage/videos/586206303>

Μάθημα 10°

R Markdown <https://vimeo.com/manage/videos/586665962>

Για την βιντεοσκόπηση των μαθημάτων χρησιμοποιήθηκε το δωρεάν λογισμικό OBS Studio (<https://obsproject.com/download>). Για την επεξεργασία των videos όπου κρίθηκε απαραίτητο χρησιμοποιήθηκε το λογισμικό Filmora. Το λογισμικό Filmora διαθέτει άδειες χρήσεις οι οποίες πρέπει να αγοραστούν από τον site της εφαρμογής πριν την χρήση της. (<https://filmora.wondershare.com/>).

Η δημιουργία των quiz πολλαπλών επιλογών έγινε με την χρήση του δωρεάν λογισμικού HotPotatoes (<https://hotpot.uvic.ca/>). Η σουίτα HotPotatoes είναι ένα σύνολο έξι εργαλείων συγγραφής, που δημιουργήθηκαν από την ομάδα Έρευνας και Ανάπτυξης στο Πανεπιστήμιο της Βικτόριας [Ανθρωπιστικών Επιστημών και Κέντρου Μέσων](#). Δίνουν τη δυνατότητα δημιουργίας διαδραστικών ασκήσεων. Οι ασκήσεις είναι τυπικές ιστοσελίδες που χρησιμοποιούν κώδικα XHTML 5 για προβολή και JavaScript (ECMAScript) για

διαδραστικότητα. Αυτά τα βασικά πρότυπα W3C υποστηρίζονται από όλα τα καλά σύγχρονα προγράμματα περιήγησης.

2 Μαθήματα

2.1 Μάθημα 1ο

Επισκόπηση

- Εγκατάσταση της R στα Windows
- Εγκατάσταση του R Studio
- R Studio – Cloud
- Το πρώτο μας πρόγραμμα σε R

2.1.1 Εγκατάσταση της R στα Windows

Βιντεοσκοπημένο Μάθημα :

Εγκατάσταση της R <https://vimeo.com/manage/videos/544518680>

Οδηγίες Εγκατάστασης της R

1. Μεταβείτε στον [επίσημο ιστότοπο του προγραμματισμού R, www.R-project.org](http://www.R-project.org)
2. Κάντε κλικ στον σύνδεσμο CRAN
3. Επιλέξτε mirror

Greece

<https://ftp.cc.uoc.gr/mirrors/CRAN/>

University of Crete

--

Εικόνα 2

4. Κάντε κλικ στο «Download R for Windows»
5. Κάντε κλικ στον σύνδεσμο που πραγματοποιεί λήψη της βασικής διανομής *Base*
6. Κάντε κλικ στο Download R 4.0.4 for Windows

[Download R 4.0.4 for Windows](#) (85 megabytes, 32/64 bit)

[Installation and other instructions](#)

[New features in this version](#)

Εικόνα 3

7. Εκτελέστε το αρχείο και ακολουθήστε τα βήματα για να εγκαταστήσετε την R.

2.1.2 Εγκατάσταση του R Studio

Βιντεοσκοπημένο Μάθημα :

Εγκατάσταση R studio <https://vimeo.com/manage/videos/544502208>

Το **RStudio** είναι το πιο δημοφιλές IDE για την εκτέλεση προγραμμάτων R και διαθέτει δωρεάν άδεια χρήσης. Η διαδικασία εγκατάστασης είναι απλή στα Windows.

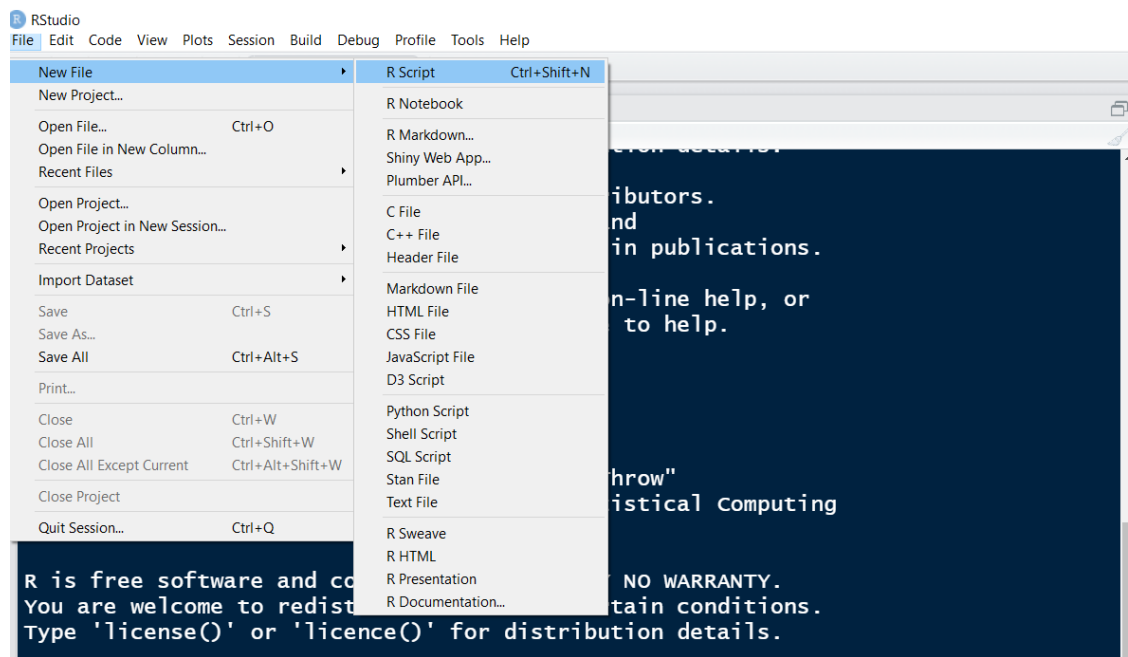
- Πηγαίνετε στην σελίδα <http://www.rstudio.com/products/rstudio/download/>
- Επιλέξτε τον **installer για Windows** και κατεβάστε το.
- Δεξί κλικ στο πακέτο -> run as administrator.
- Ακολουθήστε τις οδηγίες και κάντε την εγκατάσταση

Σημείωση η R πρέπει να είναι εγκατεστημένη στο σύστημά σας πριν εκτελέσετε το RStudio.

Αφού εγκαταστήσετε το RStudio και το ανοίξετε για πρώτη φορά, θα σας ζητήσει να επιλέξετε ποια έκδοση του R θα χρησιμοποιήσετε.

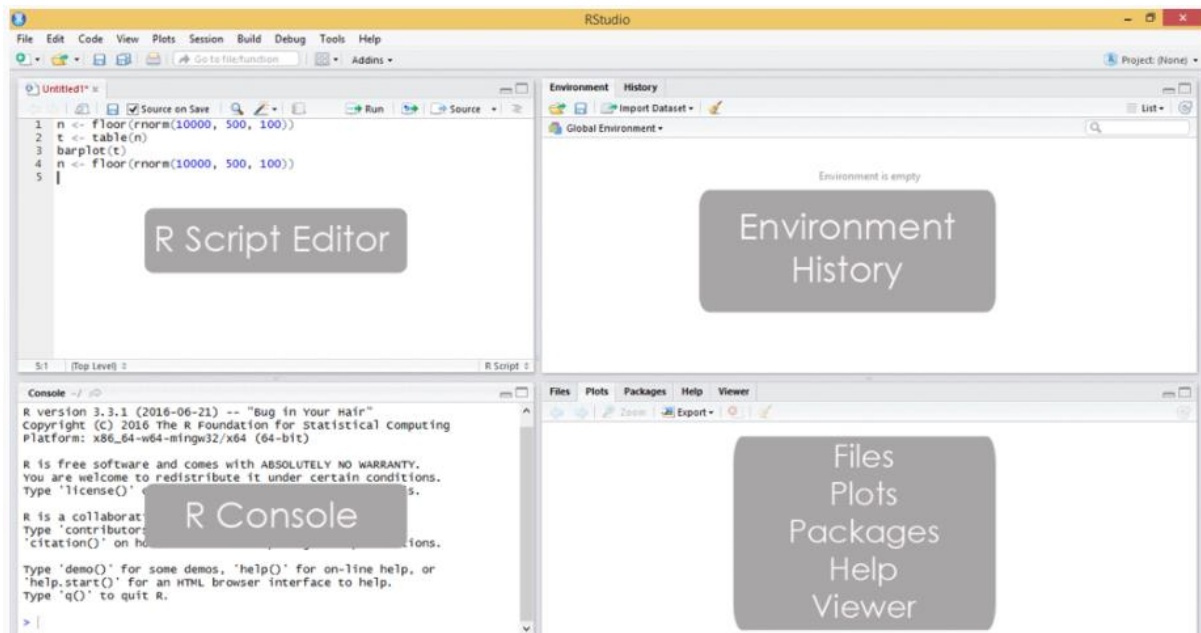
Εάν το RStudio εντοπίσει ότι η R δεν έχει εγκατασταθεί στο σύστημά σας, θα σας δείξει μια προειδοποίηση.

Εάν η R έχει εγκατασταθεί, θα δείτε τη διεπαφή του RStudio. Στην αρχή, μπορείτε να δείτε μόνο την κονσόλα της R όπου μπορείτε να γράψετε δηλώσεις μιας γραμμής και να τις εκτελέσετε. Ωστόσο, ακόμη και για μια ασήμαντη εργασία, θα πρέπει να εκτελέσετε μια ακολουθία βημάτων οπότε είναι καλύτερα να δημιουργήσετε ένα σενάριο R. Μεταβείτε στο menu **File > New File > R Script** όπως φαίνεται στην παρακάτω εικόνα για να δημιουργήσετε ένα νέο **R Script**.



Εικόνα 4

Τώρα μπορείτε να δείτε το R Script Editor όπου μπορείτε να πληκτρολογήσετε και να αποθηκεύσετε προγράμματα της R που εκτείνονται σε πολλές γραμμές. Το RStudio δεν είναι απλώς ένα πρόγραμμα επεξεργασίας κειμένου, αλλά ένα IDE που σας βοηθά να εκτελείτε και να εντοπίζετε σφάλματα στην R Script με ευκολία. Το RStudio GUI χωρίζεται σε 4 μεγάλες ενότητες όπως φαίνεται στο παρακάτω στιγμιότυπο οθόνης:



Εικόνα 5

Η κλασική προβολή του RStudio περιλαμβάνει 4 πλαίσια:

1. Το πλαίσιο πάνω αριστερά χρησιμοποιείται για την εκτέλεση Script . Μπορείτε να γράψετε κώδικα ανοίγοντας ένα νέο Script.

2. Το πλαίσιο κάτω αριστερά χρησιμοποιείται σαν Console όπου εμφανίζει τα αποτελέσματα του κώδικα.

3. Το πλαίσιο πάνω δεξιά εμφανίζει το ιστορικό του περιβάλλοντος , μεταβλητές , τιμές κλπ

4. Το πλαίσιο κάτω δεξιά είναι ένα ακόμα βοηθητικό πλαίσιο που δείχνει files και Plots μέσω των Viewer, Help και Packages

2.1.3 R Studio – Cloud

1. Πηγαίνετε σε αυτήν την σελίδα <http://rstudio.cloud>
2. Επιλέξτε Sing Up.
3. Το R Studio Cloud έχει δημιουργήσει ένα δικό σας χώρο στο cloud.
4. Ξεκινήστε δημιουργώντας ένα νέο Project.
5. Στο menu files επιλέξτε new file => R script.

2.1.4 Το πρώτο μας πρόγραμμα σε R

Βιντεοσκοπημένο Μάθημα :

Το πρώτο μας πρόγραμμα σε R <https://vimeo.com/manage/videos/544508633>

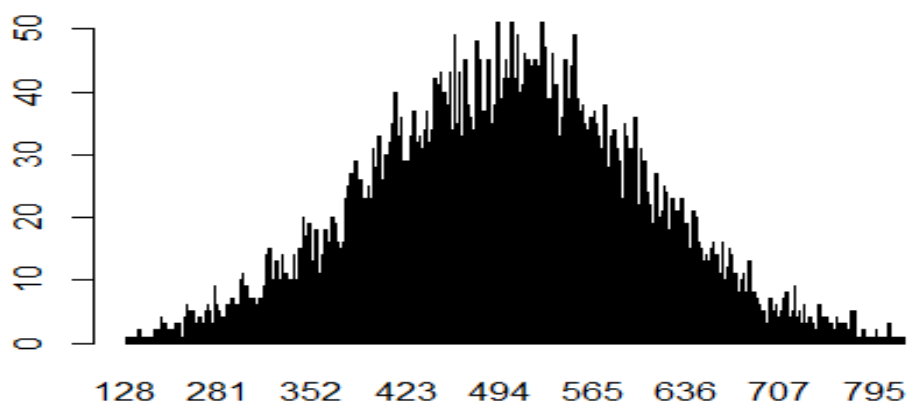
Στο πρώτο μας πρόγραμμα στην R θα παράγουμε 10.000 αριθμούς με τυχαία κατανομή, θα τους οργανώσουμε με βάση τη συχνότητα εμφάνισης και θα δημιουργήσουμε το πρώτο μας γράφημα ράβδων.

Στην πρώτη γραμμή η συνάρτηση `rnorm()` δημιουργεί μια λίστα με 10000 τυχαίους αριθμούς με κανονική κατανομή έτσι ώστε ο μέσος όρος αυτών των αριθμών να είναι 500 και η τυπική απόκλιση 100. Η συνάρτηση `floor()` παίρνει μόνο το ακέραιο μέρος από κάθε

αριθμό της λίστας και αφαιρεί το δεκαδικό ψηφίο. Η συνάρτηση `table()` παίρνει αυτούς τους 10000 αριθμούς και μετρά τη συχνότητα του καθενός.

```
n <- floor(rnorm(10000, 500, 100))
t <- table(n)
```

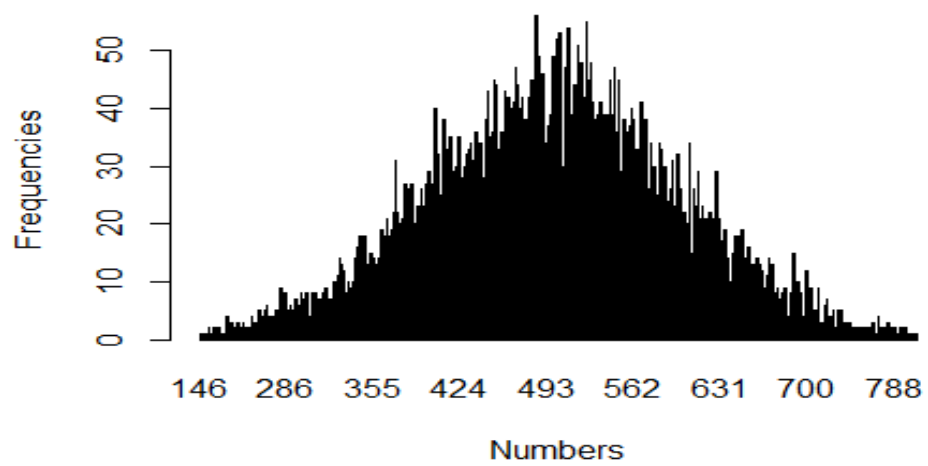
Η συνάρτηση `barplot()` παίρνει τον πίνακα με τις συχνότητες και δημιουργεί ένα γράφημα ράβδων από τα δεδομένα.



Εικόνα 6

Στην πραγματικότητα δεν χρειαζόμαστε 3 γραμμές. Σε μία μόνο γραμμή, θα μπορούσαμε να κάνουμε το ίδιο, προσθέτοντας επιπλέον και ετικέτες στους άξονες x και y :

```
barplot(table(floor(rnorm(10000, 500, 100))), xlab="Numbers", ylab="Frequencies")
```

Εικόνα 7

2.2 Μάθημα 2ο

Επισκόπηση

- Μαθηματικές πράξεις
- Μεταβλητές – Variable
- Κανόνες χρήσης μεταβλητών
- Βασικές κατηγορίες αντικειμένων - «Atomic» Classes
- Λογικοί Τελεστές - Logical Operators

Βιντεοσκοπημένο Μάθημα :

Μαθηματικές πράξεις, μεταβλητές κανόνες <https://vimeo.com/manage/videos/544516293>

2.2.1 Μαθηματικές Πράξεις

Ο πίνακας περιγράφει για την κάθε μαθηματική πράξη τον τρόπο απεικόνισης στην R :

Μαθηματική Πράξη	Απεικόνιση σε R
Πρόσθεση	8+3
Αφαίρεση	8-3
Πολλαπλασιασμός	8*3
Διαίρεση	8/3
Δύναμη	8^3
Modulus	8%%3
Απόλυτη Τιμή	abs(-8)
Τετραγωνική ρίζα	sqrt(4)
Εκθετικό	exp(4)

Φυσ λογάριθμος	log(8)
Ημίτονο	sin(pi)
Συνημίτονο	cos(pi)

2.2.2 Μεταβλητές και κανόνες

Η Μεταβλητή συμβολίζει ένα τυχαίο στοιχείο ενός συνόλου. Σε μια μεταβλητή μπορούμε να αποθηκεύσουμε μια αξία (value). Αποθηκεύει δηλαδή μια πληροφορία. Μπορούμε να την καλέσουμε αν γράψουμε στην κονσόλα το όνομά της. Δεν απαιτείται εκ των προτέρων ορισμός του τύπου της κάθε μεταβλητής. Για την ονομασία των μεταβλητών μπορούμε να χρησιμοποιήσουμε γράμματα κεφαλαία και μικρά του λατινικού αλφαβήτου, δηλαδή “**A-Z**”, “**a-z**”, αριθμούς “**0-9**”, τελεία “.” και κάτω παύλα “_”.

- Οι μεταβλητές δεν πρέπει να περιέχουν κενά.
- Οι μεταβλητές θα πρέπει να ξεκινούν με γράμμα ή τελεία.
- Δεν μπορούμε να χρησιμοποιήσουμε σαν μεταβλητές λέξεις κλειδιά τις οποίες χρησιμοποιεί η R. Αυτές είναι <code>if</code>, <code>else</code>, <code>repeat</code>, <code>while</code>, <code>break</code>, <code>in</code>, <code>function</code>, <code>for</code>, <code>next</code>, <code>NULL</code>, <code>FALSE</code>, <code>TRUE</code>, <code>NA</code>, <code>NaN</code>, <code>Inf</code>, <code>NA_integer</code>, <code>NA_real_</code>, <code>NA_complex_</code>, <code>NA_character</code>».

Αυτή η λίστα είναι διαθέσιμη πληκτρολογώντας `help(reserved)` ή `?reserved` στη γραμμή εντολών του R Studio.

Χρήσιμες πληροφορίες :

- `TRUE` και `FALSE` είναι οι λογικές σταθερές στην R.
- `NULL` αντιπροσωπεύει την απουσία μιας τιμής.
- `Inf` είναι για το “Infinity”, για παράδειγμα όταν το 1 διαιρείται με το 0 ενώ `NaN` για το “Not a Number”, για παράδειγμα όταν το 0 διαιρείται με το 0.
- `NA` σημαίνει “Μη Διαθέσιμο” και χρησιμοποιείται για την αναπαράσταση των τιμών που λείπουν.
- Η R είναι case sensitive, που σημαίνει ότι `TRUE` και `True` δεν είναι το ίδιο. Το `TRUE` είναι μια δεσμευμένη λέξη που δηλώνει μια λογική σταθερά στην R και `True` μπορεί να χρησιμοποιηθεί σαν όνομα μεταβλητής.

2.2.3 Βασικές κατηγορίες αντικειμένων «Atomic» Classes

Αριθμοί : Μπορεί να είναι τύπου integer, double ή complex. Η συνάρτηση typeof() εμφανίζει τον τύπο του αριθμού. Οι αριθμητικές σταθερές που ακολουθούνται από L θεωρούνται integer και αυτές που ακολουθούνται i θεωρούνται complex.

```
typeof(5)
[1] "double"
typeof(5L)
[1] "integer"
typeof(5i)
[1] "complex"
```

Οι αριθμοί που προηγούνται 0x ή 0X ερμηνεύονται ως δεκαεξαδικοί αριθμοί.

```
0xff
[1] 255
0XF + 1
[1] 16
```

Χαρακτήρες : Οι χαρακτήρες μπορούν να αναπαρασταθούν χρησιμοποιώντας είτε μεμονωμένα εισαγωγικά (') είτε με διπλά εισαγωγικά (").

```
'example'
[1] "example"
typeof("1")
[1] "character"
```

Λογικοί : Οι λογικοί τελεστές μπορούν να αναπαραχθούν με τις δεσμευμένες λέξεις TRUE ή T και FALSE ή F.

Λογικοί Τελεστές Οι λογικοί τελεστές έχουν δύο πιθανές τιμές , TRUE ή FALSE.

Τελεστής	Απεικόνιση στην R
Ίσον ==	equal
Όχι ίσο !=	not equal
Μικρότερο Από <	less than
Μεγαλύτερο από >	grander than
Μικρότερο ή ίσο από <=	less or equal
Μεγαλύτερο ή ίσο >=	grander or equal

Όχι !	not
Ή	or
Και &	and

Ενσωματωμένες μεταβλητές (built in variables)

Η R έχει έναν αριθμό ενσωματωμένων μεταβλητών. Συγκεκριμένα, έχει διαθέσιμες τις παρακάτω (Becker, Chambers & Wilks, 1988) :

- LETTERS: Τα 26 κεφαλαία λατινικά γράμματα.
- letters: Τα 26 μικρά λατινικά γράμματα.
- month.abb: Συντομογραφία των μηνών. • month.name: Ονόματα μηνών.
- pi: τιμή του π.

2.3 Μάθημα 3ο

Επισκόπηση

- Δομές επιλογής
- Δομές επανάληψης

2.3.1 Δομές επιλογής

Βιντεοσκοπημένο Μάθημα :

Δομές επιλογής <https://vimeo.com/manage/videos/570171874>

Η λήψη αποφάσεων είναι ένα σημαντικό μέρος του προγραμματισμού. Αυτό μπορεί να επιτευχθεί στον προγραμματισμό σε R χρησιμοποιώντας την if...else.

IF

Η σύνταξη του if είναι:

```
if (δήλωση αληθείς) {  
  κώδικας  
}
```

Εάν η δήλωση είναι TRUE, τότε εκτελείται ο κώδικας που έχουμε συμπεριλάβει στις αγκύλες {}. Αλλά αν είναι FALSE, τότε προσπερνάτε η εκτέλεση του κώδικα. Η δήλωση μπορεί να είναι ένα λογικό ή αριθμητικό διάνυσμα, αλλά σε αυτή την περίπτωση ελέγχετε μόνο το πρώτο στοιχείο του διανύσματος.

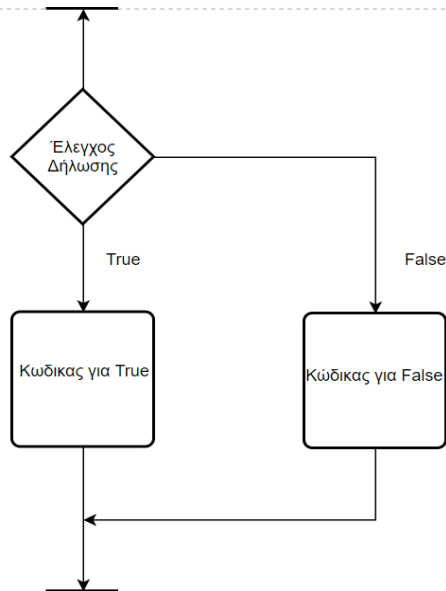
If ...else

Σύνταξη της δήλωσης if... else:

```
if (Δήλωση αληθείς) {  
  κώδικας 1  
} else {  
  Κώδικας 2  
}
```

Ο κώδικας 2 τρέχει αν η δήλωση είναι false. Επιπλέον είναι σημαντικό να σημειωθεί ότι στη δήλωση του else πρέπει να είναι στην ίδια γραμμή οι αγκύλες } else { .

Διάγραμμα ροής της δήλωσης if... else



Εικόνα 8

Παράδειγμα if ...else

```
x <- -5
if(x > 0) {
  print("Θετικός")
} else {
  print("Αρνητικός")
}
[1] "Αρνητικός"
```

Η ίδια σύνταξη μπορεί να γίνει σε μία γραμμή, σε αυτή την περίπτωση δεν χρειαζόμαστε καθόλου αγκύλες.

```
x <- 5
if(x > 0) print("Θετικός") else print("Αρνητικός")
[1] "Θετικός"
```

If ...else if

Η δήλωση `if ... else if` μας επιτρέπει να εκτελέσουμε έναν μπλοκ κώδικα για περισσότερες από δύο εναλλακτικές.

Η σύνταξη είναι :

```
if ( δήλωση αληθείς ) {  
  κώδικας 1  
} else if ( δήλωση αληθείς 2 ) {  
  Κώδικας 2  
} else if ( δήλωση αληθείς 3 ) {  
  Κώδικας 3  
} else {  
  Κώδικας 4  
}
```

Φυσικά και πάλι μόνο μια δήλωση θα εκτελεστεί, η πρώτη που είναι αληθής.

Παράδειγμα ένθετου `if ...else`

```
x <- 0  
if (x < 0) {  
  print("Αρνητικός")  
} else if (x > 0) {  
  print("Θετικός")  
} else  
  print("Μηδέν")  
[1] "Μηδέν"
```

Ifelse()

Υπάρχει όμως στην R και ένας ευκολότερος τρόπος να χρησιμοποιηθεί η `if ...else`, με την συνάρτηση `ifelse()` και την χρήση κάποιου διάνυσματος. Τα διανύσματα αποτελούν βασικό δομικό στοιχείο στο προγραμματισμό σε R. Οι περισσότερες συναρτήσεις λαμβάνουν ως είσοδο ένα διάνυσμα ή έχουν σαν έξοδο κάποιο διάνυσμα.

Σύνταξη της συνάρτησης `ifelse ()`

ifelse(δήλωση αληθείς, true, false)

Εδώ, η αληθείς δήλωση πρέπει να είναι κάποιο διάνυσμα. Η τιμή επιστροφής είναι ένα διάνυσμα με το ίδιο μήκος με αυτό της δήλωσης.

Παράδειγμα: συνάρτησης ifelse ()

```
a = c(5,7,2,9)
ifelse(a %% 2 == 0,"άρτιος","περιττός")
[1] "περιττός" "περιττός" "άρτιος" "περιττός"
```

2.3.2 Δομές επανάληψης

Βιντεοσκοπημένο Μάθημα :

Δομές επανάληψης <https://vimeo.com/manage/videos/570544672>

Οι δομές επανάληψης, loops χρησιμοποιούνται για την επανάληψη ενός συγκεκριμένου κώδικα.

While

Στη R , το «while» επαναλαμβάνει ένα κομμάτι κώδικα για όσες φορές ικανοποιείται μια συγκεκριμένη συνθήκη.

Σύνταξη `While

while (Δήλωση)

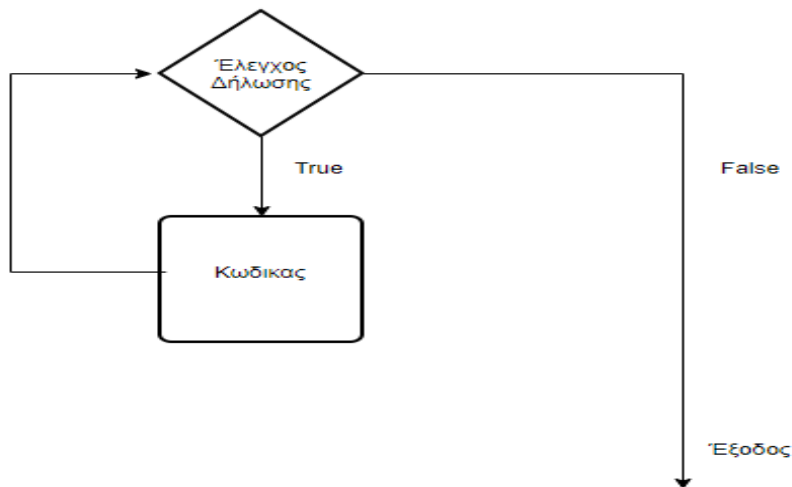
{

Κώδικας

}

Η δοθείσα δήλωση αξιολογείται και εκτελείται το σώμα για όσο είναι αληθής. Οι δηλώσεις εντός βρόγχου εκτελούνται και η ροή επιστρέφει στην αρχή του βρόγχου και επαναξιολογεί την «δήλωση», αν είναι αληθείς επαναλαμβάνεται ο βρόγχος μέχρις ότου η «δήλωση» είναι ψευδής οπότε βγαίνει από το βρόγχο.

Διάγραμμα ροής While



Εικόνα 9

Παράδειγμα While

```
i <- 1
while (i < 6) {
  print(i)
  i = i+1
}
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

Το «i» αρχικοποιείται σε 1. Η αρχική «δήλωση» $i < 6$ το αξιολογεί ως true και εκτελείτε το σώμα του βρόγχου που αυξάνει το «i» κατά ένα σε κάθε επανάληψη. Όταν το «i» πάρει την τιμή 6 η δήλωση δίνει false και το while τελειώνει.

For

Το For χρησιμοποιείτε για την επανάληψη ενός συγκεκριμένου μπλοκ κώδικα.

Σύνταξη for

```
for ( i in επαναλήψεις)
```

```
{
```

```
  Κώδικας
```

```
}
```

Οι επαναλήψεις είναι ένα διάνυσμα, και το *i* παίρνει τιμή κατά την διάρκεια εκτελέσεις του βρόγχου σε κάθε επανάληψη.

Παράδειγμα for

```
x <- c(2,3,4,5,6,7)
count <- 0
for (i in x) {
  if(i %% 2 == 0) count = count+1
}
print(count)
[1] 3
```

Ο παραπάνω βρόγχος εκτελείτε 6 φορές όσα και τα στοιχεία του δοθέντος διανύσματος. Σε κάθε επανάληψη το «i» παίρνει την τιμή του αντίστοιχου στοιχείου του διανύσματος. Αρχικά ελέγχετε αν ο αριθμός είναι μονός ή ζυγός και αν είναι ζυγός αυξάνετε ο μετρητής count κατά ένα.

Repeat

Το repeat, χρησιμοποιείται για επανάληψη ενός μπλοκ κώδικα πολλές φορές. Δεν υπάρχει κάποια συνθήκη εξόδου από τον βρόγχο. Πρέπει να βάλουμε εμείς μια κατάσταση εξόδου μέσα στον κώδικα του βρόγχου.

Σύνταξη Repeat

Κώδικας

Break

```
}
```

Δηλώνουμε το break για να βγούμε από τον βρόγχο.

Παράδειγμα repeat

```
x <- 1
repeat {
  print(x)
  x = x+1
  if (x == 6){
```

```
break
```

```
}
```

```
}
```

```
[1] 1
```

```
[1] 2
```

```
[1] 3
```

```
[1] 4
```

```
[1] 5
```

Χρησιμοποιήσαμε την συνθήκη για τον έλεγχο εξόδου . Όταν το x πάρει την τιμή 6 τερματίζεται ο βρόγχος και τυπώνονται οι τιμές του x .

2.4 Μάθημα 4ο

Δομές Δεδομένων της R

Επισκόπηση

- Διάνυσμα
- Λίστα
- Πίνακας
- Παράγοντας
- Πολυδιάστατος πίνακας
- Πλαίσια δεδομένων

2.4.1 Διάνυσμα - Vector

Βιντεοσκοπημένο Μάθημα :

Διάνυσμα <https://vimeo.com/manage/videos/575550907>

Το διάνυσμα είναι μια βασική δομή δεδομένων στην R. Περιέχει στοιχεία του ίδιου τύπου. Οι τύποι δεδομένων μπορεί να είναι λογικοί, ακέραιοι, double, χαρακτήρες ή μικτοί αριθμοί . Μπορείτε να ελέγξετε τον τύπο των δεδομένων με τη `typeof()` συνάρτηση. Μια άλλη σημαντική ιδιότητα ενός διανύσματος είναι το μήκος του. Ο αριθμός των στοιχείων του διανύσματος μπορεί να ελεγχθεί με τη συνάρτηση `length()`. Για την δημιουργία ενός διανύσματος χρησιμοποιείται η συνάρτηση `c()`. Δεδομένου ότι ένα διάνυσμα έχει στοιχεία μόνο του ίδιου τύπου η συνάρτηση αυτή θα “εξαναγκάσει” τα δοθέντα στοιχεία στον ίδιο τύπο.

```
x <- c(1, 5, 4, 9, 0)
typeof(x)
[1] "double"
```

Εύρεση αριθμού στοιχείων διανύσματος:

```
length(x)
[1] 5
```

Εξαναγκασμός σε character:

```
x <- c(1, 5.4, TRUE, "hello")
typeof(x)
[1] "character"
```

Αν θέλουμε να δημιουργήσουμε έναν διάνυσμα διαδοχικών αριθμών τότε χρησιμοποιούμε το «:» ως εξής :

```
x <- 1:7
x
[1] 1 2 3 4 5 6 7
y <- 2:-2
y
[1] 2 1 0 -1 -2
```

2.4.2 Λίστα List

Βιντεοσκοπημένο Μάθημα :

Λίστα <https://vimeo.com/manage/videos/575552170>

Η λίστα είναι μια δομή δεδομένων που έχει στοιχεία μεικτών τύπων δεδομένων. Ένα διάνυσμα vector έχει στοιχεία του ίδιου τύπου μόνο , μια λίστα μπορεί να έχει στοιχεία διαφορετικού τύπου. Η συνάρτηση `typeof()` μας δίνει σαν αποτέλεσμα μια λίστα, και η συνάρτηση `length()` τον αριθμό των διακριτών στοιχείων της. Μία λίστα μπορούμε να τη δημιουργήσουμε με τη συνάρτηση `list()`. Ακολουθεί ένα παράδειγμα μιας λίστας που έχει τρία στοιχεία διαφορετικού τύπου δεδομένων.

Παράδειγμα:

```
x <- list("a" = 2.5, "b" = TRUE, "c" = 1:3)
x
$a
[1] 2.5

$b
[1] TRUE

$c
[1] 1 2 3
```

```
typeof(x)
[1] "list"
length(x)
[1] 3
```

Δημιουργήσαμε μια λίστα με τρία στοιχεία με τύπους δεδομένων double, logical και integer αντίστοιχα. Με την συνάρτηση str() μπορούμε να πάρουμε μια πιο αναλυτική κατάσταση των στοιχείων της λίστας μας.

```
str(x)
List of 3
 $ a: num 2.5
 $ b: logi TRUE
 $ c: int [1:3] 1 2 3
```

Στο παράδειγμα μας, τα a, b και c ονομάζονται ετικέτες και καθιστούν ευκολότερη την αναφορά στα στοιχεία μιας λίστας. Ωστόσο, οι ετικέτες είναι προαιρετικές, από άποψη καλής προγραμματιστικής πρακτική συνιστάτε η χρήση τους. Μπορούμε να δημιουργήσουμε την ίδια λίστα χωρίς τις ετικέτες ως εξής :

```
list(2.5,TRUE,1:3)
[[1]]
[1] 2.5

[[2]]
[1] TRUE

[[3]]
[1] 1 2 3
```

Πρόσβαση σε στοιχεία μιας λίστας

Μπορείτε να έχετε πρόσβαση σε λίστες με παρόμοιο τρόπο με τα διανύσματα. Ας εξετάσουμε μια λίστα ως εξής:

```
x<-list(name="Dimitris",plays=c("Football","Running"),age=20)
x["age"] # Επιστρέφει την τιμή
$age
[1] 20
typeof(x["age"]) # Μονό Bracket επιστρέφει τον τύπο της λίστας
[1] "list"
x[["age"]] # Διπλό bracket επιστρέφει την τιμή
[1] 20
typeof(x[["age"]]) # Διπλό bracket και παρένθεση επιστρέφει τον τύπο του αντικειμένου
[1] "double"
```

Εναλλακτικά αντί για `[]` μπορούμε να χρησιμοποιήσουμε το `$` ως εξής:

```
x$name  
[1] "Dimitris"  
x$plays[1]  
[1] "Football"
```

Τροποποίηση μιας λίστας

Μπορούμε να αλλάξουμε τα στοιχεία μιας λίστας μέσω εκ νέου ανάθεσης. Μπορούμε δηλαδή να επιλέξουμε οποιαδήποτε από τις τεχνικές πρόσβασης στα στοιχεία που αναφέραμε παραπάνω για να τα τροποποιήσουμε.

Παράδειγμα

```
x$name<-"Elena"  
x  
$name  
[1] "Elena"  
  
$plays  
[1] "Football" "Running"  
  
$age  
[1] 20
```

Διαγραφή στοιχείων από μια λίστα

Μπορούμε να διαγράψουμε ένα στοιχείο εκχωρώντας τιμή NULL σε αυτό.

```
x$age<-NULL  
x  
$name  
[1] "Elena"  
  
$plays  
[1] "Football" "Running"
```

2.4.3 Πίνακας Matrix

Βιντεοσκοπημένο Μάθημα :

Πίνακας <https://vimeo.com/manage/videos/576106316>

Matrix, πίνακας είναι μια δισδιάστατη δομή δεδομένων στον προγραμματισμό σε R. Είναι παρόμοιο με το διάνυσμα, περιλαμβάνει στοιχεία του **ίδιου είδους** μόνο, αλλά περιέχει επιπλέον το χαρακτηριστικό διάστασης. Όλα τα χαρακτηριστικά ενός αντικειμένου μπορούν να ελεγχθούν με τη συνάρτηση `attributes()`. Η διάσταση του `matrix` ελέγχεται με τη συνάρτηση `dim()`.

Μπορούμε να ελέγξουμε εάν μια μεταβλητή είναι ένας πίνακας ή όχι με τη συνάρτηση `class()`.

Δημιουργούμε ένα Matrix χρησιμοποιώντας τη συνάρτηση `matrix()`. Η διάσταση του πίνακα ορίζετε περνώντας τιμές στα ορίσματα `nrow` και `ncol`.

Παράδειγμα:

```
x <- matrix(1:9, nrow = 3, ncol = 3)
class(x)
[1] "matrix" "array"
attributes(x)
$dim
[1] 3 3
dim(x)
[1] 3 3
```

Βλέπουμε ότι ο πίνακας γεμίζει με τα στοιχεία 1:9 με βάση τη στήλη. Αυτό μπορεί να αντιστραφεί με το όρισμα `byrow=TRUE` που γεμίζει τον πίνακα κατά σειρά.

```
x <- matrix(1:9, nrow=3, byrow=TRUE)
```

Ενα επιπλέον χρήσιμο όρισμα των πινάκων είναι το **dimnames** όπου μπορείτε να ονομάσετε τις σειρές και τις στήλες του πίνακα κατά τη δημιουργία, περνώντας μια λίστα 2 στοιχείων στο όρισμα.

```
x <- matrix(1:9, nrow = 3, dimnames = list(c("X", "Y", "Z"), c("A", "B", "C")))
```

Μπορείτε να αποκτήσετε πρόσβαση ή να τροποποιήσετε τα ονόματα με δύο χρήσιμες συναρτήσεις `colnames()` και `rownames()`.

```
colnames(x)
[1] "A" "B" "C"
rownames(x)
[1] "X" "Y" "Z"
colnames(x) <- c("a1", "a2", "a3")
rownames(x) <- c("b1", "b2", "b3")
```

Ένας άλλος τρόπος δημιουργίας ενός πίνακα είναι με τη χρήση των συναρτήσεων `cbind()` και `rbind()`.

Παράδειγμα:

#Συμπλήρωση by column

```
cbind(c(1,2,3),c(4,5,6),c(7,8,9))
```

```
  [,1] [,2] [,3]  
[1,]  1  4  7  
[2,]  2  5  8  
[3,]  3  6  9
```

#Συμπλήρωση by row

```
rbind(c(1:3),c(4:6),c(7:9))
```

```
  [,1] [,2] [,3]  
[1,]  1  2  3  
[2,]  4  5  6  
[3,]  7  8  9
```

Τέλος, μπορείτε επίσης να δημιουργήσετε έναν νέο πίνακα από ένα διάνυσμα ρυθμίζοντας τη διάστασή του, χρησιμοποιώντας την συνάρτηση `dim()`.

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

```
x
```

```
[1] 1 2 3 4 5 6 7 8 9
```

```
class(x)
```

```
[1] "numeric"
```

```
dim(x) <- c(3,3) # Μπορούμε να βάλουμε διαστάσεις σε ένα vector
```

```
x
```

```
  [,1] [,2] [,3]  
[1,]  1  4  7  
[2,]  2  5  8  
[3,]  3  6  9
```

```
class(x)
```

```
[1] "matrix" "array"
```

Πρόσβαση σε στοιχεία ενός πίνακα

Μπορούμε να έχουμε πρόσβαση σε στοιχεία ενός πίνακα χρησιμοποιώντας της `[]`. Δεν ξεχνάμε όμως ότι στους πίνακες οι γραμμές (rows) και οι στήλες (columns) είναι διανύσματα.

Με χρήση ακέραιου διανύσματος

Ορίζουμε τα στοιχεία των γραμμών και τα στοιχεία των στηλών που θα χρησιμοποιήσουμε.

Μπορούμε να χρησιμοποιήσουμε αρνητικούς ακέραιους αριθμούς για να καθορίσουμε τις γραμμές ή τις στήλες που θα εξαιρεθούν.

Παράδειγμα:

```
x <- matrix(1:9, nrow = 3, dimnames = list(c("X", "Y", "Z"), c("A", "B", "C")))
```

Επιλέγονται οι γραμμές 1 & 2 και οι στήλες 2 & 3

```
x[c(1,2),c(2,3)]  
  B C  
X 4 7  
Y 5 8
```

Επιλέγονται οι γραμμές 2,1 και όλες οι στήλες:

```
x[c(2,1),]  
  A B C  
Y 2 5 8  
X 1 4 7
```

Επιλογή όλων:

```
x[,]  
  A B C  
X 1 4 7  
Y 2 5 8  
Z 3 6 9
```

Επιλογή όλων των γραμμών εκτός της 1ης:

```
x[-1,]  
  A B C  
Y 2 5 8  
Z 3 6 9
```

Επιλογή μόνο μιας γραμμής:

```
x[1,]  
A B C  
1 4 7
```

Επιλογή μόνο μιας στήλης:

```
x[,1]  
X Y Z  
1 2 3
```

Αυτό που πρέπει να προσέξουμε εδώ είναι ότι κατα την επιλογή μιας γραμμής ή στήλης το αποτέλεσμα δίνεται ως διάνυσμα και όχι σαν πίνακας.

```
class(x[1,])  
[1] "integer"
```

Αυτό μπορεί να αποφευχθεί χρησιμοποιώντας το όρισμα `drop = FALSE` ως εξής:

```
x[1,,drop=FALSE]
```

```
A B C
X 1 4 7
```

Τώρα το αποτέλεσμά μας είναι ένας πίνακας 1x3 και όχι ένα διάνυσμα. Η συνάρτηση `class()` επιστρέφει `matrix`.

```
class(x[,drop=FALSE])
[1] "matrix" "array"
```

Με χρήση λογικού διανύσματος

Μπορούμε να δηλώσουμε για την κάθε γραμμή και στήλη αν θα επιστραφούν οι τιμές τους ή όχι, με `TRUE` επιστρέφονται και με `FALSE` όχι.

Παράδειγμα:

```
x
  A B C
X 1 4 7
Y 2 5 8
Z 3 6 9
#Επιστρέφει τις γραμμές 1 & 3 και τις στήλες 2 & 3
x[c(TRUE,FALSE,TRUE),c(FALSE,TRUE,TRUE)]
  B C
X 4 7
Z 6 9
```

Μπορούμε να αναμείξουμε λογικές τιμές με ακεραίους:

```
x[c(TRUE,FALSE,TRUE),c(2,3)]
  B C
X 4 7
Z 6 9
```

Επιλογή στοιχείων με συνθήκη

Επιλέγονται τα στοιχεία >2 :

```
x[x>2]
[1] 3 4 5 6 7 8 9
```

Επιλογή μονών αριθμών:

```
x[x%%2 != 0]
[1] 1 3 5 7 9
```

Με διάνυσμα χαρακτήρων

Μπορούμε να προσπελάσουμε έναν πίνακα χρησιμοποιώντας τα ονόματα των γραμμών και στηλών. Μπορούμε επίσης να αναμερίζουμε λογικούς, ακέραιους και χαρακτήρες.

Παράδειγμα:

```
x[, "A"]
X Y Z
1 2 3
x[TRUE,c("A","C")]
  A C
X 1 7
Y 2 8
Z 3 9
x[2:3,c("A","C")]
  A C
Y 2 8
Z 3 9
```

Τροποποίηση των στοιχείων ενός πίνακα :

Χρησιμοποιώντας τον τελεστή ανάθεσης τιμής με κάποιον από τους παραπάνω τρόπους προσπέλασης μπορούμε να τροποποιήσουμε τα στοιχεία ενός πίνακα.

Παράδειγμα:

```
x
  A B C
X 1 4 7
Y 2 5 8
Z 3 6 9
x[2,2] <- 10; x  # Αλλαγή τιμής στο στοιχείο της θέσης 2,2
  A B C
X 1 4 7
Y 2 10 8
Z 3 6 9
x[x<3] <- 0; x  # Αλλαγή τιμής σε όλα τα στοιχεία μικρότερα του 3
  A B C
X 0 4 7
Y 0 10 8
Z 3 6 9
```

Συνάρτηση t() transpose

Η συνάρτηση `transpose` αλλάζει τις γραμμές με τις στήλες:

```
t(x)
  X Y Z
A 0 0 3
B 4 10 6
C 7 8 9
```

Προσθήκη νέας γραμμής ή στήλης μπορούμε να κάνουμε με τις συναρτήσεις `rbind()` και `cbind()`.

```
y<-rbind(x,c(20,21,22))  # Προσθήκη γραμμής
y<-cbind(y, c(10,11,12,13)) # Προσθήκη στήλης
y
  A B C
X 0 4 7 10
Y 0 10 8 11
Z 3 6 9 12
 20 21 22 13
```

Διαγραφή γραμμής 4.

```
y <- y[1:3,];
y
  A B C
X 0 4 7 10
Y 0 10 8 11
Z 3 6 9 12
```

Η διάσταση του πίνακα μπορεί να αλλάξει χρησιμοποιώντας τη `dim()` συνάρτηση, `dimension` όπου δηλώνουμε νέα διάταξη στηλών και γραμμών, πρέπει βέβαια να απεικονίζονται στην νέα διάταξη όλα τα στοιχεία. Δηλαδή αν έχουμε έναν πίνακα $3 \times 4 = 12$ στοιχεία η νέα διάταξη θα πρέπει να περιλαμβάνει 12 στοιχεία (12×1 , $6 \times 2 \dots$).

```
dim(y) <- c(2,6);
y
  [,1] [,2] [,3] [,4] [,5] [,6]
[1,]  0  3  10  7  9  11
[2,]  0  4  6  8  10  12
```

2.4.4 Παράγοντες Factors

Βιντεοσκοπημένο Μάθημα :

Παράγοντες <https://vimeo.com/manage/videos/576112564>

Factors είναι μια δομή δεδομένων που χρησιμοποιείται για πεδία που λαμβάνουν μόνο προκαθορισμένο, πεπερασμένο αριθμό τιμών (κατηγορηματικά δεδομένα). Για παράδειγμα: ένα πεδίο δεδομένων όπως η οικογενειακή κατάσταση μπορεί να περιέχει μόνο τις τιμές άγαμος, έγγαμος, διεξευγμένος και χήρος. Οι διακριτές αυτές τιμές ονομάζονται επίπεδα.

Παράδειγμα: Δημιουργούμε ένα νέο factor με τη συνάρτηση factor().

```
f <- factor(c("single", "married", "married", "single"))
class(f)
[1] "factor"
levels(f)
[1] "married" "single"
```

Ο factor f έχει τέσσερα στοιχεία και δύο επίπεδα. Μπορούμε να ελέγξουμε αν μια μεταβλητή είναι factor με την συνάρτηση class() και τα επίπεδα με τη συνάρτηση levels().

Οι factors σχετίζονται στενά με τα διανύσματα. Στην πραγματικότητα, οι factors αποθηκεύονται ως ακέραια διανύσματα. Αυτό φαίνεται καθαρά από τη δομή τους.

```
str(f)
Factor w/ 2 levels "married","single": 2 1 1 2
```

Τα επίπεδα τους αποθηκεύονται σε ένα διάνυσμα χαρακτήρων και τα μεμονωμένα στοιχεία αποθηκεύονται στην πραγματικότητα ως δείκτες.

Οι factors δημιουργούνται επίσης όταν διαβάζουμε μη αριθμητικές στήλες σε ένα data frame.

Πρόσβαση στα στοιχεία ενός factor

```
f
[1] single married married single
Levels: married single
f[3]      # Πρόσβαση στο 3ο στοιχείο
[1] married
Levels: married single
f[c(3, 4)] # Πρόσβαση στο 3ο και στο 4ο στοιχείο
[1] married single
Levels: married single
f[-1]     # Πρόσβαση σε όλα εκτός του 1ου στοιχείου
[1] married married single
Levels: married single
f[c(TRUE, FALSE, TRUE, TRUE)] # Πρόσβαση με χρήση λογικών μεταβλητών
```

```
[1] single married single  
Levels: married single
```

Τροποποίηση στοιχείων ενός factor

Τα στοιχεία ενός factor μπορούν να τροποποιηθούν χρησιμοποιώντας απλές αναθέσεις. Δεν μπορούμε όμως να επιλέξουμε τιμές εκτός των προκαθορισμένων levels.

```
f[2] <- "single" # Αναθεση τιμής στη 2η θέση  
f  
[1] single single married single  
Levels: married single  
f[3] <- "widowed" # Τιμή εκτός levels δεν μπορεί να γίνει ανάθεση  
Warning in `[<-factor`(*tmp*, 3, value = "widowed")': invalid factor level, NA  
generated
```

Προσοχή η θέση που κάνουμε ανάθεση γίνεται NA μετά την αποτυχημένη ανάθεση τιμής.

Η λύση είναι να προσθέσετε πρώτα την νέα τιμή στο factor.

```
levels(f) <- c(levels(f), "widowed") # προσθήκη επιπέδου  
f[3] <- "widowed" #Ανάθεση τιμής  
f  
[1] single single widowed single  
Levels: married single widowed
```

2.4.5 Πολυδιάστατοι πίνακες Arrays

Βιντεοσκοπημένο Μάθημα :

Πολυδιάστατοι πίνακες <https://vimeo.com/manage/videos/576137400>

Τα Arrays είναι τα αντικείμενα δεδομένων R που μπορούν να αποθηκεύσουν δεδομένα σε περισσότερες από δύο διαστάσεις. Για παράδειγμα αν δημιουργήσουμε έναν πίνακα διαστάσεων (2, 3, 4) τότε δημιουργεί 4 ορθογώνιους πίνακες ο καθένας με 2 σειρές και 3 στήλες. Τα Arrays μπορούν να αποθηκεύσουν τον ίδιο τύπο δεδομένων σε όλα τα δεδομένα τους.

Δημιουργείται χρησιμοποιώντας τη συνάρτηση `array()` .


```

# Δημιουργούμε τα παρακάτω διανύσματα
vector1 <- c(5,9,3)
vector2 <- c(10,11,12,13,14,15)
# Δημιουργούμε ένα νέο array με διαστάσεις 3Χ3 η τελευταία επιλογή στην
# διάσταση είναι 2 οπότε έχουμε δισδιάστατο Array
my_Array <- array(c(vector1,vector2),dim = c(3,3,2))
my_Array
,, 1

      [,1] [,2] [,3]
[1,]   5  10  13
[2,]   9  11  14
[3,]   3  12  15

,, 2

      [,1] [,2] [,3]
[1,]   5  10  13
[2,]   9  11  14
[3,]   3  12  15

```

Ονομασία στηλών και σειρών

Μπορούμε να δώσουμε ονόματα στις σειρές, τις στήλες και τους πίνακες στον πίνακα χρησιμοποιώντας την παράμετρο `dimnames`.

```

# Δημιουργούμε τα παρακάτω διανύσματα
column.names <- c("COL1", "COL2", "COL3") # Ονόματα στηλών
row.names <- c("ROW1", "ROW2", "ROW3") # Ονόματα γραμμών
matrix.names <- c("Matrix1", "Matrix2") # Ονόματα διαστάσεων
my_Array <- array(c(vector1,vector2),dim = c(3,3,2),
                  dimnames = list(row.names,column.names,matrix.names))
print(my_Array)
,, Matrix1

      COL1 COL2 COL3
ROW1   5  10  13
ROW2   9  11  14
ROW3   3  12  15

,, Matrix2

      COL1 COL2 COL3
ROW1   5  10  13
ROW2   9  11  14
ROW3   3  12  15

```

****Πρόσβαση στα στοιχεία του Array:**

```
# Εκτύπωσε την 3η γραμμή του 2ου πίνακα
print(my_Array[3,,2])
COL1 COL2 COL3
3 12 15
# Εκτύπωσε το 1ο στοιχείο της 3ης στήλης του 1ου πίνακα
print(my_Array[1,3,1])
[1] 13
# Εκτύπωσε τον 2ο πίνακα
print(my_Array[,2])
COL1 COL2 COL3
ROW1 5 10 13
ROW2 9 11 14
ROW3 3 12 15
```

2.4.6 Πλαίσια δεδομένων Data frames

Βιντεοσκοπημένο Μάθημα :

Πλαίσια δεδομένων <https://vimeo.com/manage/videos/576142356>

Το πλαίσιο δεδομένων είναι μια δισδιάστατη δομή δεδομένων στην R με μορφή πίνακα. Κάθε στοιχείο καταλαμβάνει μια στήλη και οι τιμές τους εμφανίζονται στις σειρές. Μπορούμε να ελέγξουμε αν μια μεταβλητή είναι ένα πλαίσιο δεδομένων με την συνάρτηση `class()`. Δημιουργούμε ένα πλαίσιο δεδομένων χρησιμοποιώντας τη συνάρτηση `data.frame()`.

Βασικές εντολές data frame

```
x <- data.frame("SN" = 1:2, "Age" = c(20,10), "Name" = c("John","Elen"))
str(x) # structure of x
'data.frame': 2 obs. of 3 variables:
 $ SN : int 1 2
 $ Age : num 20 10
 $ Name: chr "John" "Elen"
```

Περιεχόμενα

x

```
SN Age Name
1 1 20 John
2 2 10 Elen
```

Τύπος

```
typeof(x) # Είναι μία ειδική περίπτωση λίστας
[1] "list"
```

κλάση

```
class(x)
[1] "data.frame"
```

Ονομασίες Στηλών

```
names(x)
[1] "SN" "Age" "Name"
```

Αριθμός στηλών

```
ncol(x)
[1] 3
```

Αριθμός γραμμών

```
nrow(x)
[1] 2
```

Μέγεθος

```
length(x)
[1] 3
```

Επιπλέον υπάρχουν πολλές λειτουργίες εισαγωγής δεδομένων όπως `read.table()`, `read.csv()`, `read.delim()`, `read.fwf()` θα δούμε κάποιες στην συνέχεια.

Πρόσβαση σε στοιχεία ενός πλαισίου δεδομένων Μπορείτε να έχετε πρόσβαση σε στοιχεία του πλαισίου δεδομένων όπως μια λίστα ή σαν ένα `matrix`.

Πρόσβαση σαν λίστα Μπορούμε να χρησιμοποιήσουμε είτε `[`, `[[` ή `$` φορέα για την πρόσβαση στα δεδομένα του `data frame`.

```
x["Name"]
Name
1 John
2 Elen
x$Name
```

```
[1] "John" "Elen"  
x[["Name"]]  
[1] "John" "Elen"
```

Η πρόσβαση με `[[` και `$` είναι παρόμοια. Ωστόσο, διαφέρει στο `[`. Η ανεύρεση με `[` θα μας επιστρέψει ένα πλαίσιο δεδομένων, ενώ τα άλλα δύο ένα διάνυσμα.

Πρόσβαση σαν matrix

Τα πλαίσια δεδομένων είναι προσβάσιμα σαν Matrix, παρέχοντας την δυνατότητα ανεύρεσης σε σειρές ή στήλες. Για τα παρακάτω παραδείγματα θα χρησιμοποιήσουμε κάποια σύνολα δεδομένων που είναι διαθέσιμα στην R. Μπορείτε να ανατρέξετε στα διαθέσιμα σύνολα δεδομένων με την εντολή `library(help = "datasets")`.

Θα χρησιμοποιήσουμε το σύνολο δεδομένων `trees`. Αρχικά θα διερευνήσουμε το σύνολο δεδομένων με την εντολή `str(trees)`. Περιλαμβάνει διάμετρο, ύψος και όγκο για τις κερασιές. Το `trees` επιστέφει όλες τις γραμμές ενός data frame.

Ένα πλαίσιο δεδομένων μπορεί να εξεταστεί χρησιμοποιώντας λειτουργίες όπως `str()` και `head()`.

```
str(trees)  
'data.frame': 31 obs. of 3 variables:  
 $ Girth : num 8.3 8.6 8.8 10.5 10.7 10.8 11 11 11.1 11.2 ...  
 $ Height: num 70 65 63 72 81 83 66 75 80 75 ...  
 $ Volume: num 10.3 10.3 10.2 16.4 18.8 19.7 15.6 18.2 22.6 19.9 ...  
head(trees,n=3) # φέρνει τις 3 πρώτες γραμμές  
 Girth Height Volume  
1 8.3 70 10.3  
2 8.6 65 10.3  
3 8.8 63 10.2
```

Τώρα προχωρούμε στην πρόσβαση στο πλαίσιο δεδομένων όπως ένα Matrix.

```
trees[2:3,] # Επιλογή 2η και 3η γραμμή  
 Girth Height Volume  
2 8.6 65 10.3  
3 8.8 63 10.2  
trees[trees$Height > 82,] # φέρνει τις γραμμές με ύψος >82  
 Girth Height Volume  
6 10.8 83 19.7  
17 12.9 85 33.8  
18 13.3 86 27.4  
31 20.6 87 77.0  
trees[10:12,2]  
[1] 75 79 76
```

Μπορούμε να δούμε στην τελευταία περίπτωση, επιστρέφεται διάνυσμα. Για να το αποφύγουμε μπορούμε να χρησιμοποιήσουμε και εδώ το όρισμα `drop=FALSE`.

```
trees[10:12,2, drop = FALSE]
  Height
10    75
11    79
12    76
```

Τροποποίηση στοιχείων πλαισίου δεδομένων

Τα πλαίσια δεδομένων μπορούν να τροποποιηθούν όπως τροποποιήσαμε πίνακες, μέσω εκ νέου ανάθεσης.

```
x[1,"Age"] <- 20
x
  SN Age Name
1 1  20 John
2 2  10 Elen
```

Προσθήκη στοιχείων Οι σειρές μπορούν να προστεθούν σε ένα πλαίσιο δεδομένων χρησιμοποιώντας τη συνάρτηση `rbind()`.

```
rbind(x,list(1,16,"Paul"))
  SN Age Name
1 1  20 John
2 2  10 Elen
3 1  16 Paul
```

Ομοίως, μπορούμε να προσθέσουμε στήλες χρησιμοποιώντας `cbind()`.

```
cbind(x,State=c("NY","FL"))
  SN Age Name State
1 1  20 John   NY
2 2  10 Elen   FL
```

Μπορούμε να προσθέσουμε στήλες μέσω απλών αναθέσεων όπως:

```
x$State <- c("NY","FL")
```

Διαγραφή στοιχείου

Οι στήλες πλαισίου δεδομένων μπορούν να διαγραφούν εκχωρώντας `NULL` σε μία στήλη.

```
x$State <- NULL
```

Ομοίως, οι σειρές μπορούν να διαγραφούν μέσω αλλαγών.

```
x <- x[-1,]
```

2.5 Μάθημα 5ο

Επισκόπηση
R – Πακέτα
R – Βιβλιοθήκη
R - Functions

Βιντεοσκοπημένο Μάθημα :

Packages και function <https://vimeo.com/manage/videos/576151445>

2.5.1 R - Πακέτα

Τα πακέτα στην R είναι μια συλλογή συναρτήσεων R, διαμορφωμένου κώδικα και δειγμάτων δεδομένων. Αποθηκεύονται κάτω από έναν κατάλογο που ονομάζεται “library”. Από προεπιλογή, η R εγκαθιστά ένα σύνολο πακέτων κατά την εγκατάσταση. Όταν ξεκινάμε την κονσόλα R, μόνο τα προεπιλεγμένα πακέτα είναι διαθέσιμα από προεπιλογή. Άλλα πακέτα που δεν είναι ήδη εγκατεστημένα πρέπει να φορτωθούν για να χρησιμοποιηθούν.

Όλα τα πακέτα που είναι διαθέσιμα στη γλώσσα R παρατίθενται στα [πακέτα R](#).

Ακολουθεί μια λίστα εντολών που πρέπει να χρησιμοποιούνται για έλεγχο, επαλήθευση και χρήση των πακέτων R.

Έλεγχος διαθέσιμων πακέτων R

Για να λάβετε τις τοποθεσίες library του υπολογιστή σας εκτελέστε την εντολή:

```
.libPaths()
```

Για να αποκτήσετε τη λίστα όλων των πακέτων που έχουν εγκατασταθεί:

```
library()
```

Για να δείτε τα πακέτα που χρησιμοποιείτε αυτή τη στιγμή:

```
search()
```

Για να εγκαταστήσετε ένα νέο πακέτο:

Υπάρχουν δύο τρόποι προσθήκης νέων πακέτων στην R. Ο ένας εγκαθιστά απευθείας από τον κατάλογο CRAN και ο άλλος κατεβάζει το πακέτο στο τοπικό σας σύστημα και το εγκαθιστά χειροκίνητα.

Εγκατάσταση απευθείας από το CRAN

Η ακόλουθη εντολή λαμβάνει τα πακέτα απευθείας από την ιστοσελίδα CRAN και εγκαθιστά το πακέτο στο περιβάλλον R.

```
install.packages("Package Name")
```

Εγκαταστήστε το πακέτο με μη αυτόματο τρόπο

Μεταβείτε στον σύνδεσμο R Πακέτα για να κατεβάσετε το απαραίτητο πακέτο. Αποθηκεύστε το πακέτο ως αρχείο .zip σε κατάλληλη τοποθεσία στο τοπικό σύστημα.

Τώρα μπορείτε να εκτελέσετε την ακόλουθη εντολή για να εγκαταστήσετε αυτό το πακέτο στο περιβάλλον R.

```
install.packages(file_name_with_path, repos = NULL, type = "source")
```

Φόρτωση πακέτου στη βιβλιοθήκη

Πρώτου χρησιμοποιηθεί ένα πακέτο στον κώδικα, πρέπει να φορτωθεί στο τρέχον περιβάλλον R. Πρέπει επίσης να φορτώσετε ένα πακέτο που έχει ήδη εγκατασταθεί στο παρελθόν αλλά δεν είναι διαθέσιμο στο τρέχον περιβάλλον.

Ένα πακέτο φορτώνεται χρησιμοποιώντας την ακόλουθη εντολή:

```
library("package Name", lib.loc = "path to library")
```

2.5.2 Function στην R

Οι συναρτήσεις χρησιμοποιούνται για τη λογική διάσπαση του κώδικα σε απλούστερα μέρη ώστε να γίνονται εύκολα στη συντήρηση και την κατανόησή τους.

Είναι πολύ απλό να δημιουργήσετε δικές σας function στην R.

Σύνταξη


```
func_name <- function (επιχείρημα) {  
  δήλωση  
}
```

- Η δεσμευμένη λέξη function χρησιμοποιείται για να δηλώσει μια συνάρτηση στην R.
- Οι δηλώσεις εντός των αγκυλών {} αποτελούν το σώμα της συνάρτησης. Οι αγκύλες είναι προαιρετικές εάν το σώμα περιέχει μόνο μία έκφραση.
- Τέλος, σε μία function δίνετε ένα όνομα με την ανάθεση σε μια μεταβλητή, func_name.

Παράδειγμα συνάρτησης

Συνάρτηση για την ύψωση του x σε δύναμη του y

```
pow <- function(x, y) {  
  result <- x^y  
  print(paste(x, "Υψωμένο στη δύναμη ", y, " είναι ", result))  
}
```

Εδώ, δημιουργήσαμε μια συνάρτηση που ονομάζεται pow(). Παίρνει δύο μεταβλητές x,y, και κάνει την πράξη ύψωση σε δύναμη. Επιπλέον χρησιμοποιούμε την ενσωματωμένη συνάρτηση paste() που συνενώνει συμβολοσειρές.

Πώς να καλέσετε μια συνάρτηση;

Μπορούμε να καλέσουμε την παραπάνω function ως εξής:

```
pow(8, 2)  
[1] "8 Υψωμένο στη δύναμη 2 είναι 64"  
pow(2, 8)  
[1] "2 Υψωμένο στη δύναμη 8 είναι 256"
```

Εδώ, τα ορίσματα που χρησιμοποιούνται στη δήλωση συνάρτησης (x και y) ονομάζονται formal arguments και αυτά που χρησιμοποιούνται κατά την κλήση της συνάρτησης ονομάζονται actual arguments.

Προεπιλεγμένες τιμές μεταβλητών Μπορούμε να εκχωρήσουμε προεπιλεγμένες τιμές σε μεταβλητές μιας συνάρτησης στο R. Ένα παράδειγμα για την παραπάνω συνάρτηση είναι:

```
pow <- function(x, y = 2) {  
  result <- x^y  
  print(paste(x, "Υψωμένο στη δύναμη ", y, " είναι ", result))  
}
```

Η χρήση της προεπιλεγμένης τιμής σε μία μεταβλητή την καθιστά προαιρετική κατά την κλήση της συνάρτησης.

Εδώ, το y είναι προαιρετικό και θα πάρει την τιμή 2.

```
pow(3)  
[1] "3 Υψωμένο στη δύναμη 2 είναι 9"  
pow(3,1)  
[1] "3 Υψωμένο στη δύναμη 1 είναι 3"
```

2.6 Μάθημα 6ο

Επισκόπηση

Import csv

Import xlsx

Import sql

Import Json

Στην R, μπορούμε να διαβάσουμε δεδομένα από αρχεία που είναι αποθηκευμένα εκτός του περιβάλλοντος R. Μπορούμε επίσης να γράψουμε δεδομένα σε αρχεία που θα αποθηκευτούν και θα έχουν πρόσβαση στο λειτουργικό σύστημα. Η R μπορεί να διαβάσει και να γράψει σε διάφορες μορφές αρχείων όπως csv, excel, xml κ.λπ.

2.6.1 import R - csv

Βιντεοσκοπημένο Μάθημα :

Εισαγωγή και εξαγωγή αρχείων csv <https://vimeo.com/manage/videos/585225011>

Σε αυτό το κεφάλαιο θα μάθουμε να διαβάζουμε δεδομένα από ένα αρχείο csv και μετά να γράφουμε δεδομένα σε ένα αρχείο csv. Το αρχείο πρέπει να υπάρχει στον τρέχοντα κατάλογο εργασίας, ώστε η R να μπορεί να το διαβάσει. Για την ρύθμιση του καταλόγου εργασίας χρησιμοποιήστε την συνάρτηση `getwd()` που μας δίνει ποιος είναι ο τρέχον κατάλογος εργασίας. Μπορούμε να ορίσουμε νέο κατάλογο εργασίας με την συνάρτηση `setwd()`.

```
print(getwd())  
[1] "C:/Users/Korina/OneDrive/ΔΙΠΛΩΜΑΤΙΚΗ"  
setwd("D:/R_SIE/repositories/R/")
```

Διαβάζοντας ένα αρχείο CSV Ακολουθεί ένα απλό παράδειγμα της λειτουργίας `read.csv()` για την ανάγνωση ενός αρχείου CSV που είναι διαθέσιμο στον τρέχοντα κατάλογο εργασίας σας. (χρησιμοποιήστε το αρχείο `input.csv` που θα βρείτε στον φάκελο Common)

```
data <- read.csv("input.csv")  
print(data)  
  id  name salary start_date  dept  
1  1  Rick 623.30 2012-01-01    IT
```

```

2 2   Dan 515.20 2013-09-23 Operations
3 3  Michelle 611.00 2014-11-15   IT
4 4    Ryan 729.00 2014-05-11    HR
5 5    Gary 843.25 2015-03-27  Finance
6 6    Nina 578.00 2013-05-21    IT
7 7   Simon 632.80 2013-07-30 Operations
8 8    Guru 722.50 2014-06-17  Finance

```

Ανάλυση του αρχείου CSV

Από προεπιλογή, η λειτουργία `read.csv()` δίνει την έξοδο ως `dataframe`. Αυτό μπορεί εύκολα να ελεγχθεί ως εξής:

```

#Είναι data.frame ?
print(is.data.frame(data))
[1] TRUE

```

Επίσης, μπορούμε να ελέγξουμε τον αριθμό στηλών και σειρών:

```

#Τύπωσε αριθμό στηλών
print(ncol(data))
[1] 5
#Τύπωσε αριθμό γραμμών
print(nrow(data))
[1] 8

```

Μόλις διαβάσουμε δεδομένα σε ένα πλαίσιο δεδομένων, μπορούμε να εφαρμόσουμε όλες τις λειτουργίες που ισχύουν για τα πλαίσια δεδομένων:

```

# Μέγιστος μισθός
sal <- max(data$salary)
print(sal)
[1] 843.25

#Άτομο με μέγιστο μισθό
Person_max_sal <- subset(data, salary == max(salary))
print(Person_max_sal)
  id name salary start_date dept
5  5 Gary 843.25 2015-03-27 Finance

# Εργάζονται στο τμήμα πληροφορικής
Employee_Dept <- subset( data, dept == "IT")
print(Employee_Dept)
  id  name salary start_date dept
1  1  Rick 623.3 2012-01-01   IT

```

```
3 3 Michelle 611.0 2014-11-15 IT
6 6 Nina 578.0 2013-05-21 IT
```

Σύνθετα κριτήρια: άτομα στο τμήμα πληροφορικής των οποίων ο μισθός είναι μεγαλύτερος από 600

```
IT_employees_sal <- subset(data, salary > 600 & dept == "IT")
print(IT_employees_sal)
  id  name salary start_date dept
1 1  Rick 623.3 2012-01-01  IT
3 3 Michelle 611.0 2014-11-15  IT
```

#Προσλήψεις >= 2014

```
hire_a2014 <- subset(data, as.Date(start_date) > as.Date("2014-01-01"))
print(hire_a2014)
  id  name salary start_date dept
3 3 Michelle 611.00 2014-11-15  IT
4 4  Ryan 729.00 2014-05-11  HR
5 5  Gary 843.25 2015-03-27 Finance
8 8  Guru 722.50 2014-06-17 Finance
```

Αποθήκευση σε αρχείο CSV

Η R μπορεί να δημιουργήσει αρχείο csv με τη συνάρτηση write.csv (). Αυτό το αρχείο δημιουργείται στον κατάλογο εργασίας.

```
write.csv(hire_a2014,"new_input.csv")
newdata <- read.csv("new_input.csv")
print(newdata)
  X id  name salary start_date dept
1 3 3 Michelle 611.00 2014-11-15  IT
2 4 4  Ryan 729.00 2014-05-11  HR
3 5 5  Gary 843.25 2015-03-27 Finance
4 8 8  Guru 722.50 2014-06-17 Finance
```

Παρατηρήστε ότι έχει δημιουργηθεί μια νέα στήλη x που περιλαμβάνει τα δεδομένα της στήλης id. Αν ανοίξετε το αρχείο θα δείτε ότι ουσιαστικά η στήλη δεν έχει τίτλο. Μπορούμε να αποφύγουμε τη δημιουργία αυτής της στήλης επιλέγοντας κατά την εγγραφή row.names=FALSE :

```
write.csv(hire_a2014,"new_input.csv", row.names = FALSE)
newdata <- read.csv("new_input.csv")
print(newdata)
  id  name salary start_date dept
1 3 Michelle 611.00 2014-11-15  IT
2 4  Ryan 729.00 2014-05-11  HR
```

```
3 5 Gary 843.25 2015-03-27 Finance
4 8 Guru 722.50 2014-06-17 Finance
```

2.6.2 import R - xls

Βιντεοσκοπημένο Μάθημα :

Εισαγωγή και εξαγωγή αρχείων excel <https://vimeo.com/manage/videos/585270041>

Η R μπορεί να διαβάσει απευθείας από αυτά τα αρχεία χρησιμοποιώντας ορισμένα ειδικά πακέτα excel. Θα χρησιμοποιήσουμε το πακέτο xlsx. Η R μπορεί επίσης να γράψει σε αρχείο excel χρησιμοποιώντας αυτό το πακέτο.

Εγκατάσταση πακέτου xlsx

Φορτώστε το πακέτο γράφοντας την παρακάτω εντολή στην Console :

```
install.packages("xlsx")
```

Επαληθεύστε ότι το πακέτο φορτώθηκε γράφοντας την παρακάτω εντολή στην κονσόλα:

```
any(grepl("xlsx",installed.packages()))
[1] TRUE
```

Τώρα θα πρέπει να φορτώσουμε την library του πακέτου με την παρακάτω εντολή :

```
library("xlsx")
```

Ανάγνωση του αρχείου Excel

Αντιγράψτε το input.xlsx που θα βρείτε το directory Common στο working directory.

Οι συναρτήσεις read.xlsx () και read.xlsx2 () μπορούν να χρησιμοποιηθούν για την ανάγνωση του περιεχομένου ενός φύλλου εργασίας του Excel. Η διαφορά μεταξύ αυτών των δύο λειτουργιών είναι ότι η read.xlsx διατηρεί τον τύπο δεδομένων των περιεχομένων του xlsx. Προσπαθεί να μαντέψει τον τύπο κλάσης της μεταβλητής που αντιστοιχεί σε κάθε

στήλη στο φύλλο εργασίας. Λάβετε υπόψη ότι η λειτουργία `read.xlsx` είναι αργή για μεγάλα σύνολα δεδομένων (φύλλο εργασίας με περισσότερα από 100.000 κελιά). Η `read.xlsx2` είναι ταχύτερη σε μεγάλα αρχεία σε σύγκριση με τη λειτουργία `read.xlsx`. Οι απλουστευμένες μορφές αυτών των δύο λειτουργιών είναι:

```
read.xlsx(file, sheetIndex, header=TRUE, colClasses=NA) read.xlsx2(file, sheetIndex,  
header=TRUE, colClasses="character")
```

- `file` : η διαδρομή του αρχείου για ανάγνωση
- `sheetIndex` : ένας αριθμός που δείχνει το ευρετήριο του φύλλου για ανάγνωση. π.χ.: χρησιμοποιήστε το 1 για να διαβάσετε το πρώτο φύλλο.
- `header` : μια λογική τιμή. Εάν `TRUE`, η πρώτη σειρά χρησιμοποιείται για τα ονόματα των μεταβλητών
- `colClasses` : ένα διάνυσμα χαρακτήρων που αντιπροσωπεύει την κλάση κάθε στήλης

```
data <- read.xlsx("input.xlsx",1)  
print(data)  
  id  name salary start_date  dept name.1  city  
1 1  Rick 623.30 2012-01-01    IT  Rick Seattle  
2 2   Dan 515.20 2013-09-23 Operations  Dan  Tampa  
3 3 Michelle 611.00 2014-11-15    IT Michelle Chicago  
4 4   Ryan 729.00 2014-05-11    HR   Ryan Seattle  
5 5   Gary 843.25 2015-03-27 Finance  Gary Houston  
6 6   Nina 578.00 2013-05-21    IT   Nina Boston  
7 7  Simon 632.80 2013-07-30 Operations Simon Mumbai  
8 8   Guru 722.50 2014-06-17 Finance  Guru Dallas
```

Γράψτε δεδομένα σε ένα αρχείο Excel

Οι συναρτήσεις `write.xlsx()` και `write.xlsx2()` μπορούν να χρησιμοποιηθούν για την εξαγωγή δεδομένων από την R σε ένα βιβλίο εργασίας του Excel. Σημειώστε ότι το `write.xlsx2` επιτυγχάνει καλύτερη απόδοση σε σύγκριση με το `write.xlsx` για πολύ μεγάλο `data.frame` (με περισσότερα από 100.000 κελιά).

Οι απλουστευμένες μορφές αυτών των δύο λειτουργιών είναι:

```
write.xlsx(x, file, sheetName="Sheet1",col.names=TRUE, row.names=TRUE)
write.xlsx2(x, file, sheetName="Sheet1",col.names=TRUE, row.names=TRUE)
```

- x : το data.frame που θα γραφτεί στο βιβλίο εργασίας
- file : η διαδρομή του αρχείου εξόδου
- sheetName : μια συμβολοσειρά χαρακτήρων που θα χρησιμοποιηθεί για το όνομα του φύλλου.
- col.names, row.names : μια λογική τιμή που καθορίζει εάν τα ονόματα στηλών / ονόματα σειρών του x πρόκειται να γραφτούν στο αρχείο

Παραδείγματα:

Για να προσθέσετε πολλά σύνολα δεδομένων στο ίδιο βιβλίο εργασίας του Excel, πρέπει να χρησιμοποιήσετε το όρισμα `append = TRUE`. Αυτό απεικονίζεται στον ακόλουθο κώδικα R:

```
write.xlsx(data, file="new_input.xlsx",sheetName="Employees")
write.xlsx(data, file="new_input.xlsx",sheetName="Employees2", append=TRUE)
# Νέο worksheet
write.xlsx(mtcars, file="new_input.xlsx",sheetName="mtcars2", append=TRUE)
```

Δημιουργήστε και διαμορφώστε ένα ωραίο βιβλίο εργασίας του Excel

Η συνάρτηση `write.xlsx()` είναι χρήσιμη όταν θέλετε απλά να γράψετε ένα data.frame σε ένα αρχείο `xlsx`. Ο στόχος αυτής της ενότητας είναι να σας δείξει πώς να δημιουργήσετε μια ωραία αναφορά Excel που περιέχει έναν μορφοποιημένο πίνακα δεδομένων και γραφικές παραστάσεις.

Τα βήματα που θα ακολουθήσουμε είναι :

```
install.packages("devtools")
devtools::install_github("kassambara/r2excel")
library(r2excel)
```

Βήμα 1/4 Δημιουργήστε ένα βιβλίο εργασίας


```
wb<-createWorkbook(type="xlsx")
```

Βήμα 2/4 Ορίστε ορισμένα στυλ κελιών :

Χρώμα και μέγεθος γραμματοσειράς , ευθυγράμμιση κειμένου , περίγραμμα και μορφή δεδομένων.

Η συνάρτηση CellStyle () μπορεί να χρησιμοποιηθεί για τη δημιουργία στυλ κελιού. Μια απλοποιημένη μορφή της συνάρτησης είναι:

```
CellStyle(wb, dataFormat=NULL, alignment=NULL, border=NULL, fill=NULL, font=NULL)
```

Τίτλος και Υπότιτλοι

```
TITLE_STYLE <- CellStyle(wb)+ Font(wb, heightInPoints=16,color="blue", isBold=TRUE, underline=1)
```

```
SUB_TITLE_STYLE <- CellStyle(wb) + Font(wb, heightInPoints=14, isItalic=TRUE, isBold=FALSE)
```

Στυλ για δεδομένα , γραμμές και στήλες

```
TABLE_ROWNames_STYLE <- CellStyle(wb) + Font(wb, isBold=TRUE)  
TABLE_COLNames_STYLE <- CellStyle(wb) + Font(wb, isBold=TRUE) +  
Alignment(wrapText=TRUE, horizontal="ALIGN_CENTER") +  
Border(color="black", position=c("TOP", "BOTTOM"),  
pen=c("BORDER_THIN", "BORDER_THICK"))
```

wb : ένα αντικείμενο βιβλίου εργασίας όπως επιστρέφεται από το createWorkbook ή το loadWorkbook.

dataFormat : ένα αντικείμενο DataFormat

alignment : ένα αντικείμενο ευθυγράμμισης

Border : ένα αντικείμενο περιγράμματος

font : ένα αντικείμενο γραμματοσειράς

Πιο αναλυτικά :

1. **wb** : ένα αντικείμενο βιβλίου εργασίας όπως επιστράφηκε από createWorkbook ή loadWorkbook.

2. Τα κύρια ορίσματα για τη συνάρτηση Font () :

color : χρώμα γραμματοσειράς

heightInPoints : μέγεθος γραμματοσειράς . Οι συνήθεις τιμές είναι 10, 12, 14 κ.λπ.

isBold, isItalic : μια λογική που δείχνει εάν η γραμματοσειρά πρέπει να είναι έντονη ή πλάγια υπογράμμιση : ένας ακέραιος αριθμός που καθορίζει το πάχος της υπογράμμισης . Οι πιθανές τιμές είναι 0, 1, 2.

name : η γραμματοσειρά που θα χρησιμοποιηθεί. π.χ.: “Courier New”.

3. Τα κύρια ορίσματα για τη συνάρτηση Alignment () :

wrapText : μια λογική τιμή που δείχνει εάν το κείμενο είναι τυλιγμένο.

horizontal : η οριζόντια ευθυγράμμιση . Πιθανές τιμές είναι: “ALIGN_CENTER”, “ALIGN_JUSTIFY”, “ALIGN_LEFT”, “ALIGN_RIGHT”.

vertical : η κάθετη στοίχιση. Πιθανές τιμές είναι: “VERTICAL_BOTTOM”, “VERTICAL_CENTER”, “VERTICAL_JUSTIFY”, “VERTICAL_TOP”.

rotation : Περιστροφή, μια αριθμητική τιμή που καθορίζει τους βαθμούς που θέλετε να περιστρέψετε το κείμενο στο κελί. Η προεπιλεγμένη τιμή είναι 0.

4. Τα κύρια ορίσματα για τη συνάρτηση Border () :

color : το χρώμα περιγράμματος . π.χ.: color = “red” ή color = “# FF0000”

position : η συνοριακή θέση . Οι επιτρεπόμενες τιμές είναι: “BOTTOM”, “LEFT”, “TOP”, “RIGHT”

pen : το στυλ της πέννας. Επιτρεπόμενες τιμές είναι: «BORDER_DASH_DOT», «BORDER_DASH_DOT_DOT», «BORDER_DASHED», «BORDER_DOTTED», «BORDER_DOUBLE», «BORDER_HAIR», «BORDER_MEDIUM», «BORDER_MEDIUM_DASH_DOT», «BORDER_MEDIUM_DASH_DOT_DOT», «BORDER_MEDIUM_DASHED», «BORDER_NONE», «BORDER_SLANTED_DASH_DOT “,” BORDER_THICK “,” BORDER_THIN ”.

Βήμα 3/4 Γράψτε δεδομένα και γραφικές παραστάσεις στο βιβλίο εργασίας

Για να προσθέσετε δεδομένα, το πρώτο βήμα είναι να δημιουργήσετε ένα φύλλο στο βιβλίο εργασίας για να περιέχει τα δεδομένα. Αυτό μπορεί να γίνει χρησιμοποιώντας τη συνάρτηση `createSheet()` :

```
sheet <- createSheet(wb, sheetName = "US State Facts")
```

Προσθέστε έναν τίτλο σε ένα φύλλο εργασίας

Για να προσθέσετε έναν τίτλο, η διαδικασία είναι:

1. Δημιουργήστε μια νέα σειρά
2. Δημιουργήστε ένα κελί σε αυτήν τη σειρά για να περιέχει τον τίτλο.
3. ορίστε την τιμή κελιού.

Για να απλοποιηθεί ο κώδικας στην R μπορούμε να μαζέψουμε όλα τα δεδομένα στην παρακάτω συνάρτηση :

```
#+++++  
# Βοηθητική function εισαγωγής τίτλων  
#+++++  
# - sheet : Το αντικείμενο που περιλαμβάνει τον τίτλο  
# - rowIndex : Αριθμητική τιμή που δηλώνετε η γραμμή του τίτλου  
# - title : Το κείμενο του Τίτλου  
# - titleStyle : Το στυλ του τίτλου  
  
xlsx.addTitle<-function(sheet, rowIndex, title, titleStyle){  
  rows <-createRow(sheet,rowIndex=rowIndex)  
  sheetTitle <-createCell(rows, colIndex=1)  
  setCellValue(sheetTitle[[1,1]], title)  
  setCellStyle(sheetTitle[[1,1]], titleStyle)  
}
```

Τώρα μπορούμε να χρησιμοποιήσουμε την `addTitle` για να εισάγουμε τίτλο και υπότιτλο στο xls

```
# Add title  
xlsx.addTitle(sheet, rowIndex=1, title="US State Facts",  
  titleStyle = TITLE_STYLE)  
# Add sub title  
xlsx.addTitle(sheet, rowIndex=2,
```

```
title="Data sets related to the 50 states of USA.",  
titleStyle = SUB_TITLE_STYLE)
```

Προσθέστε έναν πίνακα σε ένα φύλλο εργασίας

Η συνάρτηση `addDataframe ()` μπορεί να χρησιμοποιηθεί για την προσθήκη του πίνακα στο νέο φύλλο.

Δείτε τις πρώτες γραμμές του πίνακα με την συνάρτηση `head()`

```
head(state.x77)  
# προσθήκη πίνακα  
addDataFrame(state.x77, sheet, startRow=3, startColumn=1,  
              colnamesStyle = TABLE_COLNAMES_STYLE,  
              rownamesStyle = TABLE_ROWNAME_STYLE)  
# Αλλαγή μεγέθους στήλης  
setColumnWidth(sheet, colIndex=c(1:ncol(state.x77)), colWidth=11)
```

Ορίσματα της συνάρτησης `addDataFrame ()` :

startRow , startColumn : μια αριθμητική τιμή που δείχνει την αρχική σειρά και στήλη

colnameStyle , rownameStyle : Ένα αντικείμενο `CellStyle` για να προσαρμόσετε τα ονόματα κεφαλίδας και γραμμής πίνακα

Ορίσματα της συνάρτησης `setColumnWidth ()` :

colIndex : ένα αριθμητικό διάνυσμα που δείχνει τις στήλες που θέλετε να αλλάξετε το μέγεθος.

colWidth : Πλάτος της στήλης

Προσθέστε ένα γράφημα σε ένα φύλλο εργασίας του Excel

```
# Δημιουργία εικόνας γραφήματος  
png("boxplot.png", height=800, width=800, res=250, pointsize=8)  
boxplot(count ~ spray, data = InsectSprays,  
        col = "blue")  
dev.off()  
png  
2  
  
# Δημιουργία νέου φύλλου εργασίας  
sheet <-createSheet(wb, sheetName = "boxplot")  
# Add a title to the sheet
```

```

xlsx.addTitle(sheet, rowIndex=1,
              title="Box plot using InsectSprays data",
              titleStyle = TITLE_STYLE)

# Προσθήκη της εικόνας στο νέο φύλλο
addPicture("boxplot.png", sheet, scale = 1, startRow = 4,
          startColumn = 1)

# Διαγραφή εικόνας από το δίσκο
res<-file.remove("boxplot.png")

```

Βήμα 4/4 Αποθήκευση του αρχείου xlsx

```
saveWorkbook(wb, "r-xlsx-report-example.xlsx")
```

Το πλήρες παράδειγμα :

```

library(xlsx)
# Δημιουργήσει ένα νέο αρχείο

wb<-createWorkbook()
# Φτιάξε κάποια στυλ

# Στυλ Τίτλων και υποτίτλων

TITLE_STYLE <- CellStyle(wb)+ Font(wb, heightInPoints=16, color="blue", isBold=TRUE, underline=1)
SUB_TITLE_STYLE <- CellStyle(wb) + Font(wb, heightInPoints=14,
    isItalic=TRUE, isBold=FALSE)

# Στυλ για πίνακα δεδομένων (στήλες και γραμμές)
TABLE_ROWNames_STYLE <- CellStyle(wb) + Font(wb, isBold=TRUE)
TABLE_COLNames_STYLE <- CellStyle(wb) + Font(wb, isBold=TRUE) +
    Alignment(wrapText=TRUE, horizontal="ALIGN_CENTER") +
    Border(color="black", position=c("TOP", "BOTTOM"),
        pen=c("BORDER_THIN", "BORDER_THICK"))

# Δημιουργήσει ένα νέο φύλλο εργασίας

sheet <- createSheet(wb, sheetName = "US State Facts")

# Βοηθητική function για στυλ
xlsx.addTitle<-function(sheet, rowIndex, title, titleStyle){
  rows <-createRow(sheet,rowIndex=rowIndex)
  sheetTitle <-createCell(rows, colIndex=1)
  setCellValue(sheetTitle[[1,1]], title)
  setCellStyle(sheetTitle[[1,1]], titleStyle)
}

```

```
}
```

Εισαγωγή τίτλου και υπότιτλου

```
xlsx.addTitle(sheet, rowIndex=1, title="US State Facts",  
  titleStyle = TITLE_STYLE)
```

Add sub title

```
xlsx.addTitle(sheet, rowIndex=2,  
  title="Data sets related to the 50 states of USA.",  
  titleStyle = SUB_TITLE_STYLE)
```

Εισαγωγή πίνακα σε ένα νέο φύλλο εργασίας

```
addDataFrame(state.x77, sheet, startRow=3, startColumn=1,  
  colnamesStyle = TABLE_COLNAMES_STYLE,  
  rownamesStyle = TABLE_ROWNAME_STYLE)
```

Change column width

```
setColumnWidth(sheet, colIndex=c(1:ncol(state.x77)), colWidth=11)
```

Δημιουργία εικόνας γραφήματος

```
png("boxplot.png", height=800, width=800, res=250, pointsize=8)  
boxplot(count ~ spray, data = InsectSprays,  
  col = "blue")  
dev.off()
```

Δημιουργία νέου φύλλου

```
sheet <-createSheet(wb, sheetName = "boxplot")
```

Προσθήκη τίτλου

```
xlsx.addTitle(sheet, rowIndex=1, title="Box plot using InsectSprays data",  
  titleStyle = TITLE_STYLE)
```

Προσθήκη εικόνας

```
addPicture("boxplot.png", sheet, scale = 1, startRow = 4,  
  startColumn = 1)
```

Διαγραφή εικόνας από το δίσκο

```
res<-file.remove("boxplot.png")
```

Αποθήκευση του αρχείου

```
saveWorkbook(wb, "r-xlsx-report-example.xlsx")
```

Εμπνευσμένο από το site (<http://www.sthda.com/english/wiki/r-xlsx-package-a-quick-start-guide-to-manipulate-excel-files-in-r> , 2019)

2.6.3 import R - mySQL

Βιντεοσκοπημένο Μάθημα :

Εργασίες στην MySql με το Rmysql <https://vimeo.com/manage/videos/585274047>

Η R διαθέτει ένα ενσωματωμένο πακέτο με το όνομα “RMySQL” το οποίο παρέχει εγγενή σύνδεση με τη βάση δεδομένων MySql. Μπορείτε να εγκαταστήσετε αυτό το πακέτο στο περιβάλλον R χρησιμοποιώντας την ακόλουθη εντολή.

```
install.packages("RMySQL")  
library(RMySQL)  
install.packages("odbc")  
library(odbc)
```

Σύνδεση R με MySql

Μόλις εγκατασταθεί το πακέτο δημιουργούμε ένα αντικείμενο σύνδεσης στην R για σύνδεση με τη βάση δεδομένων. Παίρνει ως είσοδο το όνομα χρήστη, τον κωδικό πρόσβασης, το όνομα βάσης δεδομένων και το όνομα κεντρικού υπολογιστή.

```
# Δημιουργία σύνδεσης στο MySQL database.
```

```
# Σύνδεση με την demo βάση "sakila"
```

```
mysqlconnection <- dbConnect(MySQL(), user='root',password='root', dbname='sakila',  
host='localhost', port=8111)
```

Για να δούμε τους διαθέσιμους πίνακες της βάσης δεδομένων:

```
dbListTables(mysqlconnection)
```

Query σε πινάκες

Μπορούμε να εκτελέσουμε queries τους πίνακες της βάσης δεδομένων χρησιμοποιώντας τη συνάρτηση `dbSendQuery ()` . Το ερώτημα εκτελείται στο MySql και το σύνολο αποτελεσμάτων επιστρέφεται χρησιμοποιώντας τη συνάρτηση R `fetch ()` . Τέλος αποθηκεύεται ως πλαίσιο δεδομένων στην R.

```
# Επίλογή όλων των γραμμών από τον πίνακα actor"
```

```
result <- dbSendQuery(mysqlconnection, "select * from actor")
```

```
# Φόρτωση τα αποτελέσματα στο data frame MysqlDataFrame με την εντολή fetch και  
φέρτε τις 5 πρώτες γραμμές
```

```
MysqlDataFrame <- fetch(result, n = 5)
```

```
print(MysqlDataFrame)
```

Ερώτημα με φίλτρο

Μπορούμε να περάσουμε οποιοδήποτε έγκυρο επιλεγμένο ερώτημα για να λάβουμε το αποτέλεσμα.

```
result = dbSendQuery(mysqlconnection, "select * from actor where last_name = 'TORN'")
```

```
# Fetch all the records(with n = -1) and store it as a data frame.
```

```
MysqlDataFrame = fetch(result, n = -1)
```

```
print(MysqlDataFrame)
```

Update σε πίνακα

Μπορούμε να κάνουμε update στα δεδομένα της βάσης μέσω R με τη συνάρτηση dbSendQuery ().

```
dbSendQuery(mysqlconnection, "update payment set amount = 1.5 where payment_id =  
1 and customer_id= '1'")
```

```
# Επίλογή όλων των γραμμών από τον πίνακα payment"
```

```
result <- dbSendQuery(mysqlconnection, "select * from payment where payment_id = 1  
and customer_id= '1' ")
```

```
# Φόρτωση τα αποτελέσματα στο data frame MysqlDataFrame με την εντολή fetch και  
φέρτε τις 5 πρώτες γραμμές
```

```
Mypayment <- fetch(result, n = -1)
```

```
print(Mypayment)
```

Insert σε πίνακα

```
dbSendQuery(mysqlconnection,  
  "insert into actor(actor_id,first_name,last_name,last_update)  
  values('201', 'PENELOPE', 'GUINNESS', '2006-02-15 04:34:33')"  
)
```

Create πίνακα

Μπορούμε να δημιουργήσουμε πίνακες στη MySQL χρησιμοποιώντας τη συνάρτηση `dbWriteTable()`. Αντικαθιστά τον πίνακα εάν υπάρχει ήδη και παίρνει ένα πλαίσιο δεδομένων ως είσοδο.

```
#δημιουργούμε μια νέα σύνδεση τη βάση που θέλουμε να δημιουργήσουμε τον πίνακα
db <- dbConnect(MySQL(), user='root',password='root', dbname='sakila',
host='localhost', port=8111)
dbSendQuery(con, "SET GLOBAL local_infile = true;")
#Θα εισάγουμε το dataframe mtcars στην βάση δεδομένων "sakila"
dbWriteTable(db, "mtcars", mtcars[, ], overwrite = TRUE)
```

Drop πίνακα

Μπορούμε να διαγράψουμε πίνακα στη βάση δεδομένων MySQL περνώντας τη δήλωση `drop table` στη συνάρτηση `dbSendQuery()` με τον ίδιο τρόπο που χρησιμοποιήσαμε για την αναζήτηση δεδομένων από πίνακες.

```
dbSendQuery(mysqlconnection, 'drop table if exists mtcars')
```

2.6.4 import R json

Το αρχείο JSON αποθηκεύει δεδομένα ως κείμενο σε μορφή αναγνώσιμη. Το `Json` αποθηκεύει απλές δομές δεδομένων και αντικείμενα με μορφή `JavaScript`. Η `R` μπορεί να διαβάσει αρχεία `JSON` χρησιμοποιώντας το πακέτο `rjson`.

Εγκαταστήστε το πακέτο `rjson`

```
install.packages("rjson")
```

Στην συνέχεια φορτώστε την `library`

```
library("rjson")
```

Διαβάστε το αρχείο `JSON` Το αρχείο `JSON` διαβάζεται από την `R` χρησιμοποιώντας τη συνάρτηση `JSON()` και αποθηκεύεται στην `R` ως λίστα.

```
MyJson <- fromJSON(file = "input.json")
print(MyJson)
```

Μπορούμε να μετατρέψουμε τα δεδομένα της λίστας σε ένα πλαίσιο δεδομένων `R` για περαιτέρω ανάλυση χρησιμοποιώντας τη συνάρτηση `as.data.frame()`.

```
json_data_frame <- as.data.frame(MyJson)  
print(json_data_frame)
```

2.7 Μάθημα 7ο

Διαγράμματα και Γραφήματα

Επισκόπηση

Διάγραμμα πίτας

Διάγραμμα ράβδων

Boxplots

Ιστογράμμο

Βιντεοσκοπημένο Μάθημα :

Διαγράμματα και Γραφήματα <https://vimeo.com/manage/videos/586206303>

2.7.1 Διάγραμμα πίτας

Η γλώσσα προγραμματισμού R έχει πολλές βιβλιοθήκες για τη δημιουργία διαγραμμάτων και γραφημάτων. Ένα γράφημα πίτας είναι μια αναπαράσταση των τιμών ως φέτες ενός κύκλου με διαφορετικά χρώματα.

Στην R το γράφημα πίτας δημιουργείται χρησιμοποιώντας τη συνάρτηση `pie()` σαν διάνυσμα θετικών αριθμών. Οι πρόσθετες παράμετροι χρησιμοποιούνται για τον ορισμό ετικετών, χρώματος, τίτλου κ.λπ.

Σύνταξη Η βασική σύνταξη για τη δημιουργία ενός γραφήματος πίτας χρησιμοποιώντας το R είναι:

```
pie(x, labels, radius, main, col, clockwise)
```

όπου: `x` είναι ένα διάνυσμα που περιέχει τις αριθμητικές τιμές που χρησιμοποιούνται στο γράφημα πίτας.

labels χρησιμοποιούνται για να δώσουν περιγραφή στις φέτες.

radius είναι η ακτίνα του κύκλου του γραφήματος πίτας. (τιμή μεταξύ -1 και $+1$).

main είναι ο τίτλο του γραφήματος.

col είναι η παλέτα χρωμάτων.

clockwise είναι μια λογική τιμή που δείχνει εάν οι φέτες σχεδιάζονται δεξιόστροφα ή αριστερόστροφα.

Παράδειγμα

Δημιουργούμε δεδομένα για την πίτα με ένα απλό διάνυσμα θετικών αριθμών.

```
x <- c(21, 62, 10, 53)
```

Δημιουργούμε ετικέτες για την πίτα.

```
labels <- c("London", "New York", "Singapore", "Mumbai")
```

Σχεδιασμός πίτας.

```
pie(x, labels, main = "Πόλεις", col = rainbow(length(x)))
```



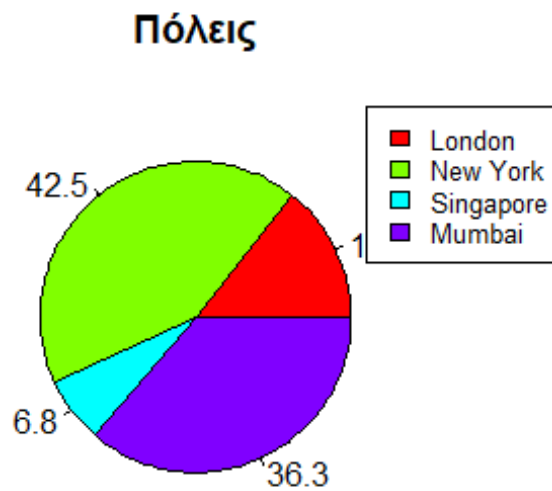
Εικόνα 10

Μπορούμε να προσθέσουμε κάποιες επιπλέον παραμέτρους, όπως να συμπεριλάβουμε το ποσοστό επί του συνόλου σαν ετικέτα, υπολογίζοντας την μεταβλητή `piepercent` και να ορίσουμε το μέγεθος του επεξηγηματικού πλαισίου `legend` με τη μεταβλητή `cx`.

```
piepercent<- round(100*x/sum(x), 1)
```

```
pie(x, labels = piepercent, main = "Πόλεις",col = rainbow(length(x)))
```

```
legend("topright", c("London", "New York", "Singapore", "Mumbai"), cex = 0.8,
      fill = rainbow(length(x)))
```



Εικόνα 11

3D γράφημα πίτας

Ένα γράφημα πίτας με 3 διαστάσεις μπορεί να σχεδιαστεί χρησιμοποιώντας πρόσθετα πακέτα. Το πακέτο `plotrix` έχει μια συνάρτηση που ονομάζεται `pie3D()` που χρησιμοποιείται για αυτό.

```
install.packages("plotrix")
library(plotrix)
pie3D(x, labels = labels, labelcex=0.8, explode = 0.1, main = "Χώρες", shade=0.5)
```

2.7.2 Γραφήματα ράβδων

Ένα γράφημα ράβδων αντιπροσωπεύει δεδομένα σε ορθογώνιες ράβδους με μήκος της ράβδου ανάλογο με την τιμή της μεταβλητής. Το R χρησιμοποιεί τη συνάρτηση `barplot()` για τη δημιουργία γραφημάτων. Η R μπορεί να σχεδιάσει τόσο κάθετες όσο και οριζόντιες ράβδους στο γράφημα ράβδων.

Η βασική σύνταξη για τη δημιουργία ενός ραβδόγραμμα στο R είναι

`barplot(H,xlab,ylab,main, names.arg,col)` Όπου :

H είναι ένα διάνυσμα ή *matrix* που περιέχει αριθμητικές τιμές που χρησιμοποιούνται στο γράφημα ράβδων.

xlab είναι η ετικέτα για τον άξονα x.

ylab είναι η ετικέτα για τον άξονα y.

main είναι ο τίτλος του ραβδογράμματος.

names.arg είναι ένα διάνυσμα ονομάτων που εμφανίζονται κάτω από κάθε γραμμή. **col** χρησιμοποιείται για να δώσει χρώματα στις γραμμές του γραφήματος.

Παράδειγμα

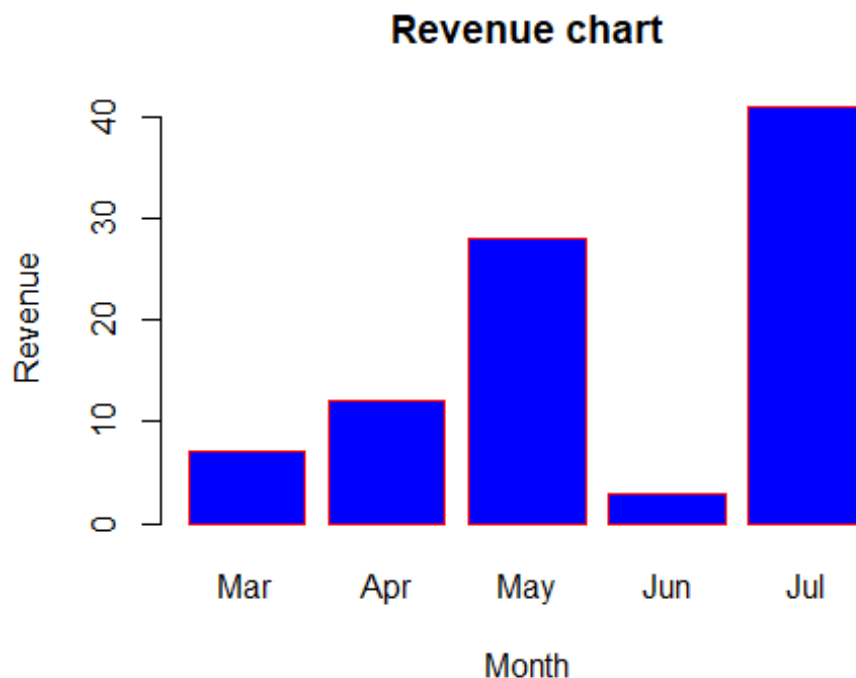
Δημιουργούμε ένα απλό γράφημα ράβδων χρησιμοποιώντας μόνο το διάνυσμα εισαγωγής και το όνομα κάθε ράβδου.

```
H <- c(7,12,28,3,41)
```

```
M <- c("Mar","Apr","May","Jun","Jul")
```

Σχεδιασμός Γραφήματος

```
barplot(H,names.arg=M,xlab="Month",ylab="Revenue",col="blue",  
main="Revenue chart",border="red")
```



Εικόνα 12

Μπορούμε να δημιουργήσουμε γράφημα ράβδων με ομάδες ράβδων και στοίβες σε κάθε γραμμή χρησιμοποιώντας έναν πίνακα ως τιμές εισόδου.

```
# Διανύσματα
```

```
colors = c("green","orange","brown")
```

```
months <- c("Mar","Apr","May","Jun","Jul")
```

```
regions <- c("East","West","North")
```

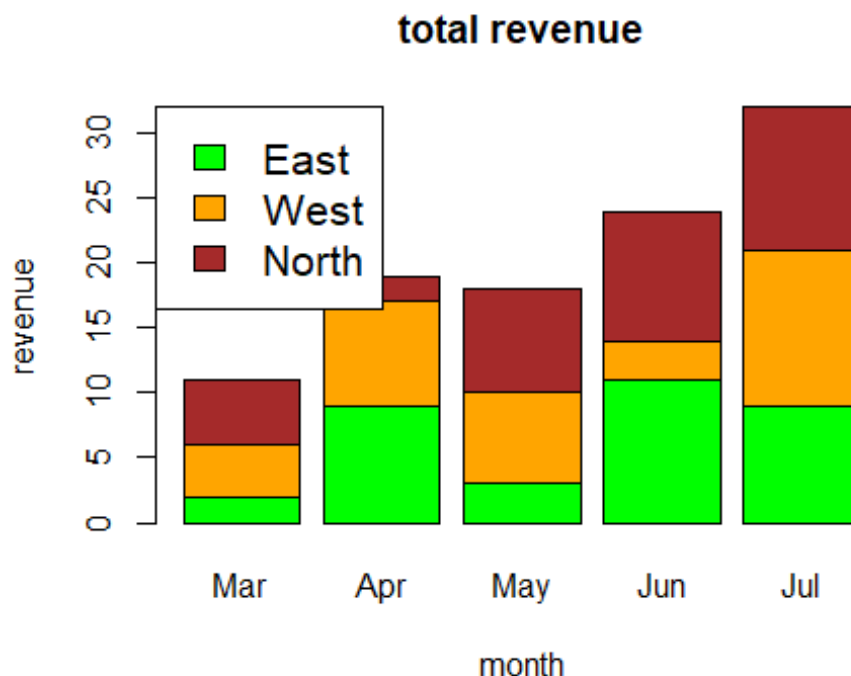
```
#Τιμές
```

```
Values <- matrix(c(2,9,3,11,9,4,8,7,3,12,5,2,8,10,11), nrow = 3, ncol = 5, byrow = TRUE)
```

```
#Σχεδιασμός Ραβδογράμματος
```

```
barplot(Values, main = "total revenue", names.arg = months, xlab = "month", ylab = "revenue", col = colors)
```

```
legend("topleft", regions, cex = 1.3, fill = colors)
```



Εικόνα 13

2.7.3 Boxplots

Στα περιγραφικά στατιστικά στοιχεία, η γραφική παράσταση κουτιού είναι μια μέθοδος για την γραφική απεικόνιση ομάδων αριθμητικών δεδομένων μέσω των τεταρτημωρίων τους. Τα γραφήματα κουτιού μπορεί επίσης να έχουν γραμμές που εκτείνονται από τα κουτιά που δείχνουν μεταβλητότητα έξω από τα άνω και κάτω τεταρτημώρια, εξ ου και οι όροι διάγραμμα box-and-whisker plot και box-and-whisker diagram.

Τα boxplots δημιουργούνται στην R χρησιμοποιώντας τη συνάρτηση `boxplot()`.

Η βασική σύνταξη για τη δημιουργία ενός boxplot είναι:

```
boxplot(x, data, notch, varwidth, names, main)
```

Όπου :

x είναι ένα διάνυσμα ή ένας τύπος.

data είναι το πλαίσιο δεδομένων.

notch είναι μια λογική τιμή. Ορίστε ως *Αληθινή* για να σχεδιάσετε μια εγκοπή.

varwidth είναι μια λογική τιμή. Ορίστε ως *αληθές* για να σχεδιάσετε το πλάτος του κουτιού ανάλογα με το μέγεθος του δείγματος.

names είναι οι ομαδικές ετικέτες που θα εκτυπωθούν κάτω από κάθε πλαίσιο.

main χρησιμοποιείται για να δώσει έναν τίτλο στο γράφημα.

Παράδειγμα

Χρησιμοποιούμε το σύνολο δεδομένων `mtcars` που είναι διαθέσιμο στο περιβάλλον R για να δημιουργήσουμε ένα βασικό boxplot και επιλέγουμε να δούμε τις στήλες `mpg` και `cyl`.

```
input <- mtcars[,c('mpg','cyl')]
print(head(input))
      mpg cyl
Mazda RX4    21.0  6
Mazda RX4 Wag 21.0  6
Datsun 710   22.8  4
Hornet 4 Drive 21.4  6
Hornet Sportabout 18.7  8
Valiant      18.1  6
```

Δημιουργία του Boxplot

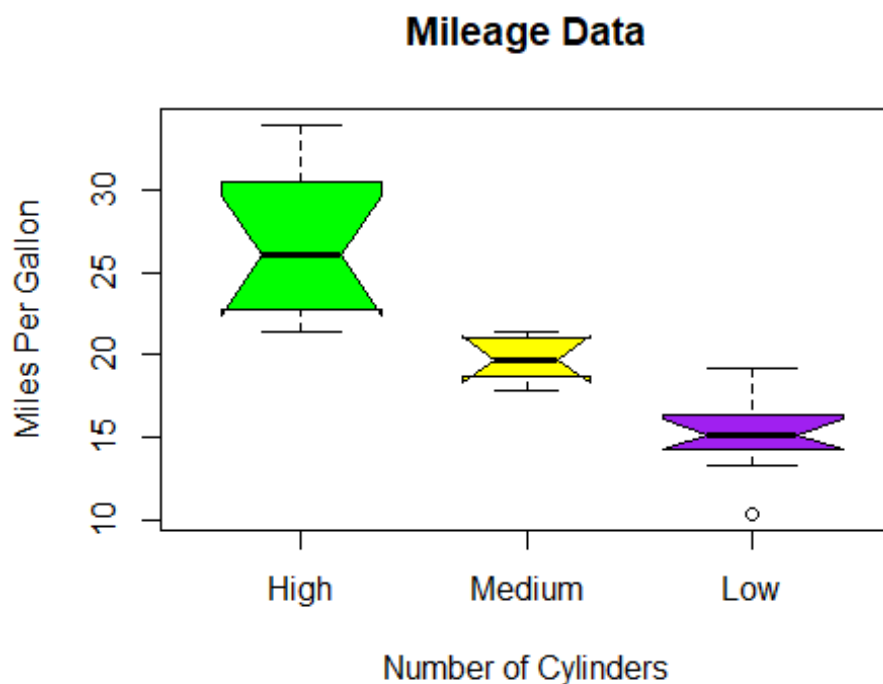
Το παρακάτω σενάριο θα δημιουργήσει ένα γράφημα boxplot για τη σχέση μεταξύ `mpg` (μίλια ανά γαλόνι) και `cyl` (αριθμός κυλίνδρων).

```
# Σχεδιασμός διαγράμματος.
boxplot(mpg ~ cyl, data = mtcars,
        xlab = "Number of Cylinders",
```



```
ylab = "Miles Per Gallon",
main = "Mileage Data",
notch = TRUE,
varwidth = TRUE,
col = c("green", "yellow", "purple"),
names = c("High", "Medium", "Low")
)
```

Warning in bxp(list(stats = structure(c(21.4, 22.8, 26, 30.4, 33.9, 17.8, : some notches went outside hinges ('box'): maybe set notch=FALSE



Εικόνα 14

2.7.4 Ιστογράμμα - Histograms

Ένα ιστόγραμμα αντιπροσωπεύει τις συχνότητες των τιμών μιας μεταβλητής που έχει ενσωματωθεί σε περιοχές. Το ιστόγραμμα είναι παρόμοιο με το ραβδόγραμμα, αλλά η διαφορά είναι ότι ομαδοποιεί τις τιμές σε συνεχή εύρη. Κάθε γραμμή στο ιστόγραμμα αντιπροσωπεύει το ύψος του αριθμού των τιμών που υπάρχουν σε αυτό το εύρος.

Στην R η δημιουργία ιστογραμμάτων γίνεται με τη συνάρτηση `hist()`. Η βασική σύνταξη για τη δημιουργία ιστογράμματος με χρήση του R είναι:

`hist(v,main,xlab,xlim,ylim,breaks,col,border)` όπου :

V είναι ένα διάνυσμα που περιέχει αριθμητικές τιμές που χρησιμοποιούνται στο ιστόγραμμα.

main δείχνει τον τίτλο του γραφήματος.

col χρησιμοποιείται για τον καθορισμό του χρώματος των ράβδων.

border χρησιμοποιείται για να ορίσετε το χρώμα περιγράμματος κάθε ράβδου.

xlab χρησιμοποιείται για να δώσει περιγραφή του άξονα x.

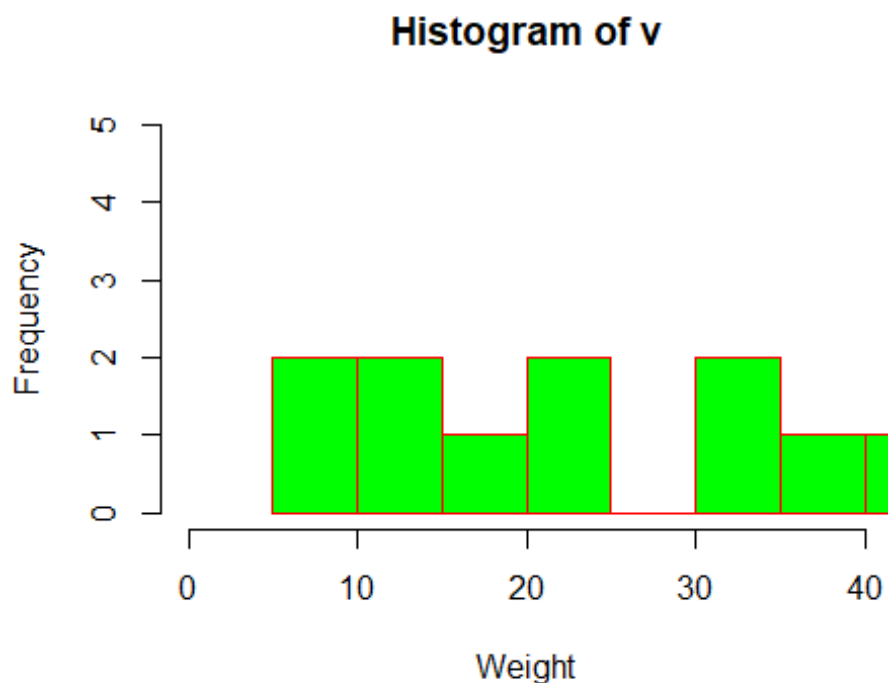
xlim χρησιμοποιείται για τον καθορισμό του εύρους τιμών στον άξονα x.

ylim χρησιμοποιείται για τον καθορισμό του εύρους τιμών στον άξονα y.

breaks χρησιμοποιείται για να δημιουργήσει κενά μεταξύ.

Παράδειγμα

```
#δημιουργία διανύσματος  
v <- c(9,13,21,8,36,22,12,41,31,33,19)  
#Δημιουργία Ιστογράμματος  
hist(v,xlab = "Weight",col = "green",border = "red", xlim = c(0,40), ylim = c(0,5),  
breaks = 5)
```



Εικόνα 15

2.8 Μάθημα 8ο

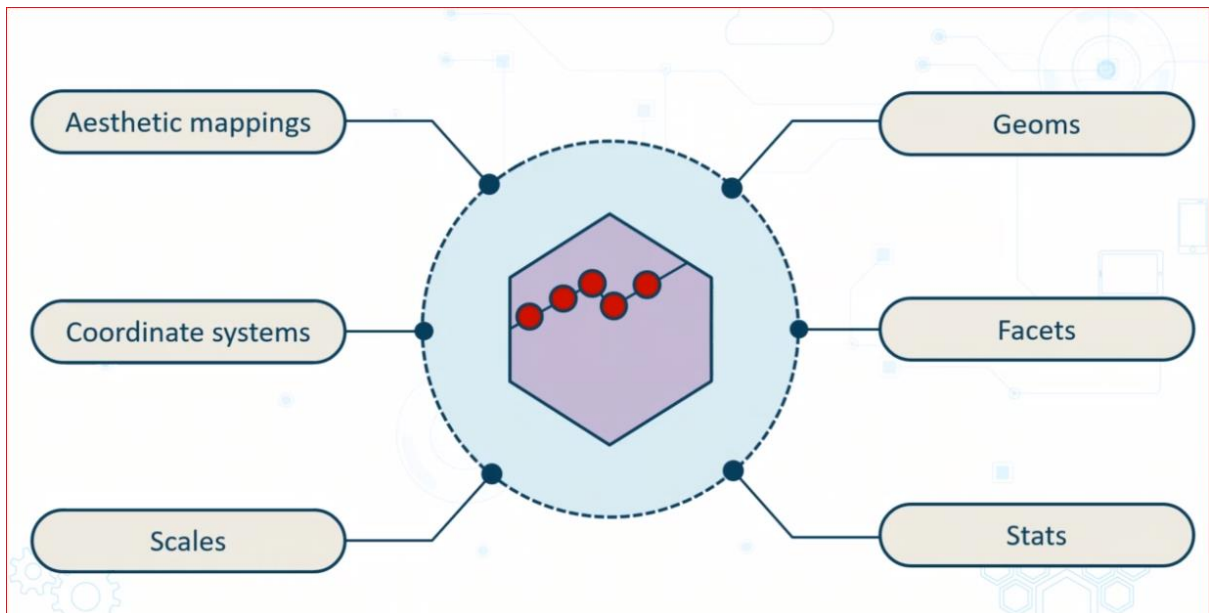
Επισκόπησή:

ggplot2

2.8.1 Οπτικοποίηση δεδομένων

Η R έχει πολλά συστήματα για τη δημιουργία γραφημάτων, αλλά το ggplot2 είναι ένα από τα πιο κομψά και πιο ευέλικτα. Σε αυτό το μάθημα θα επικεντρωθούμε στην παρουσίαση της ggplot γιατί ο τρόπος με τον οποίο δημιουργείτε γραφικές παραστάσεις στο ggplot2 είναι πολύ διαφορετικός από τα βασικά γραφικά της R.

Παρακάτω παρουσιάζονται τα χαρακτηριστικά της ggplot2.



Εικόνα 16

Aesthetic mappings: Συντεταγμένες που χρησιμοποιείται για να δείξει τις μεταβλητές x και y. Μπορεί επίσης να χρησιμοποιηθεί για τον έλεγχο του χρώματος, του μεγέθους ή του σχήματος των σημείων.

Geoms : Γεωμετρικά στοιχεία, τα οποία καθορίζουν τους τύπους γραφικών, είτε είναι ιστόγραμμα ή γραφική παράσταση.

Facet : Οι όψεις χρησιμοποιούνται σε περίπτωση που θέλετε να χωρίσετε τα δεδομένα σας με μία ή περισσότερες μεταβλητές και στη συνέχεια να σχεδιάσετε μαζί το υποσύνολο των δεδομένων.

Stats : Στατιστικά στοιχεία που είναι ένα από τα στοιχεία του ggplot2. Χρησιμοποιείται σε περίπτωση που κάποια plots απεικονίζουν έναν μετασχηματισμό του αρχικού συνόλου δεδομένων.

Scales : Οι κλίμακες ελέγχουν τον τρόπο με τον οποίο ένα γράφημα χαρτογραφεί τιμές δεδομένων.

Coordinates systems : Συστήματα συντεταγμένων αποτελούν μέρος όλων των διαγραμμάτων. Υπάρχουν πολλές πολύ χρήσιμες επιλογές που μπορούν να χρησιμοποιηθούν για την προσαρμογή του συστήματος συντεταγμένων πλοκής (Grolemund, G. & Wickham, H, 2018).

Το ggplot2 είναι μια βιβλιοθήκη του πακέτου tidyverse. Για να αποκτήσετε πρόσβαση στα σύνολα δεδομένων, τις σελίδες βοήθειας και τις λειτουργίες που θα χρησιμοποιήσουμε, φορτώστε το tidyverse εκτελώντας τον παρακάτω κώδικα:

```
install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

Θα χρησιμοποιήσουμε δεδομένα του dataset mpg. Για να γίνει πιο κατανοητή η ανάλυση των γραφημάτων θα εφαρμόσουμε κάποια ερωτήματα στα δεδομένα μας και θα αναζητήσουμε απαντήσεις στην εμφάνιση των γραφημάτων :

1. Τα αυτοκίνητα με μεγάλους κινητήρες χρησιμοποιούν περισσότερα καύσιμα από τα αυτοκίνητα με μικρούς κινητήρες;

Τώρα θα προσπαθήσουμε να απαντήσουμε χρησιμοποιώντας το dataset mpg, που περιλαμβάνει παρατηρήσεις που συλλέγονται από την Αμερικανική Υπηρεσία Προστασίας Περιβάλλοντος για 38 μοντέλα αυτοκινήτων. Θα διερευνήσουμε πώς φαίνεται η σχέση μεταξύ του μεγέθους του κινητήρα και της απόδοσης καυσίμου. Είναι θετική; Αρνητική; Γραμμική; Μη γραμμική;

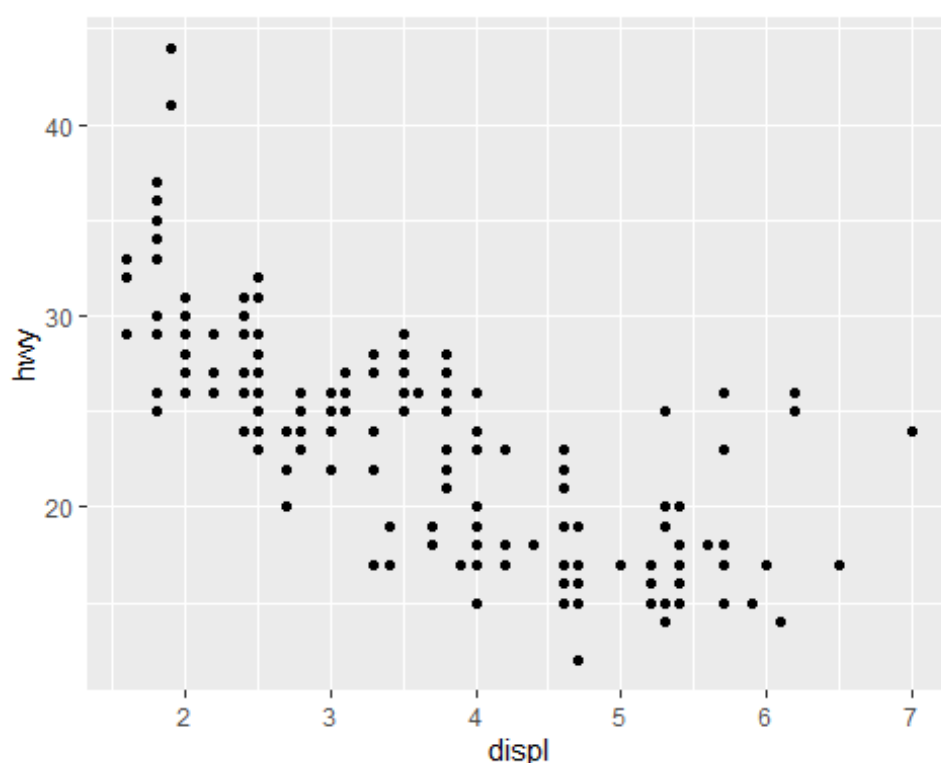
mpg

Θα χρησιμοποιήσουμε την μεταβλητή `displ`, που αφορά το μέγεθος του κινητήρα ενός αυτοκινήτου, σε λίτρα και την μεταβλητή `hwy`, που αφορά την απόδοση καυσίμου ενός αυτοκινήτου στον αυτοκινητόδρομο, σε μίλια ανά γαλόνι (`mpg`). Ένα αυτοκίνητο με χαμηλή απόδοση καυσίμου καταναλώνει περισσότερο καύσιμο από ένα αυτοκίνητο με υψηλή απόδοση καυσίμου όταν ταξιδεύουν στην ίδια απόσταση.

Για να μάθετε περισσότερα για το dataset `mpg`, ανοίξτε τη σελίδα βοήθειας εκτελώντας `?mpg`.

Δημιουργία `ggplot`

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy))
```



Εικόνα 17

Το διάγραμμα δείχνει μια αρνητική σχέση μεταξύ του μεγέθους του κινητήρα (`displ`) και της απόδοσης καυσίμου (`hwy`). Με άλλα λόγια, τα αυτοκίνητα με μεγάλους κινητήρες χρησιμοποιούν περισσότερα καύσιμα.

Η συνάρτηση `geom_point()` προσθέτει **επίπεδα** στην πλοκή του διαγράμματος. Το `ggplot2` συνοδεύεται από πολλές συναρτήσεις `geom` που η κάθε μία προσθέτει διαφορετικό τύπο επιπέδου σε μια πλοκή. Το `aes()` αφορά τα `x` και `y` ορίσματα, τις συντεταγμένες του διαγράμματος.

Ένας γενικός τύπος σύνταξης της ggplot είναι :

ggplot(data = <DATA>) + <GEOM_FUNCTION> (mapping = aes(<MAPPINGS>))

2. Στο παρακάτω διάγραμμα, μια ομάδα σημείων (επισημαίνονται με κόκκινο χρώμα) και φαίνονται να πέφτουν εκτός της γραμμικής τάσης. Αυτά τα αυτοκίνητα έχουν μεγαλύτερη απόσταση σε μίλια από ό,τι περιμέναμε. Πώς μπορείτε να εξηγήσετε αυτά τα αυτοκίνητα;

Ας υποθέσουμε ότι τα αυτοκίνητα είναι υβριδικά. Ένας τρόπος για να δοκιμάσουμε αυτήν την υπόθεση είναι να κοιτάζουμε την τιμή της μεταβλητής class για κάθε αυτοκίνητο. Η class μεταβλητή του mpg συνόλου δεδομένων ταξινομεί τα αυτοκίνητα σε ομάδες όπως compact, μεσαίο μέγεθος και SUV. Εάν τα απομακρυσμένα σημεία είναι υβριδικά, θα πρέπει να ταξινομηθούν ως compact αυτοκίνητα ή, ίσως, αυτοκίνητα subcompact.

Προσθέτουμε μια τρίτη μεταβλητή, την class, σε ένα διδιάστατο scatterplot, αλλάζοντας τα ορίσματα της aes() . Η aes() αισθητική περιλαμβάνει δηλώσεις όπως το μέγεθος, το σχήμα ή το χρώμα των σημείων.

Παράδειγμα

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy, colour = class))
```

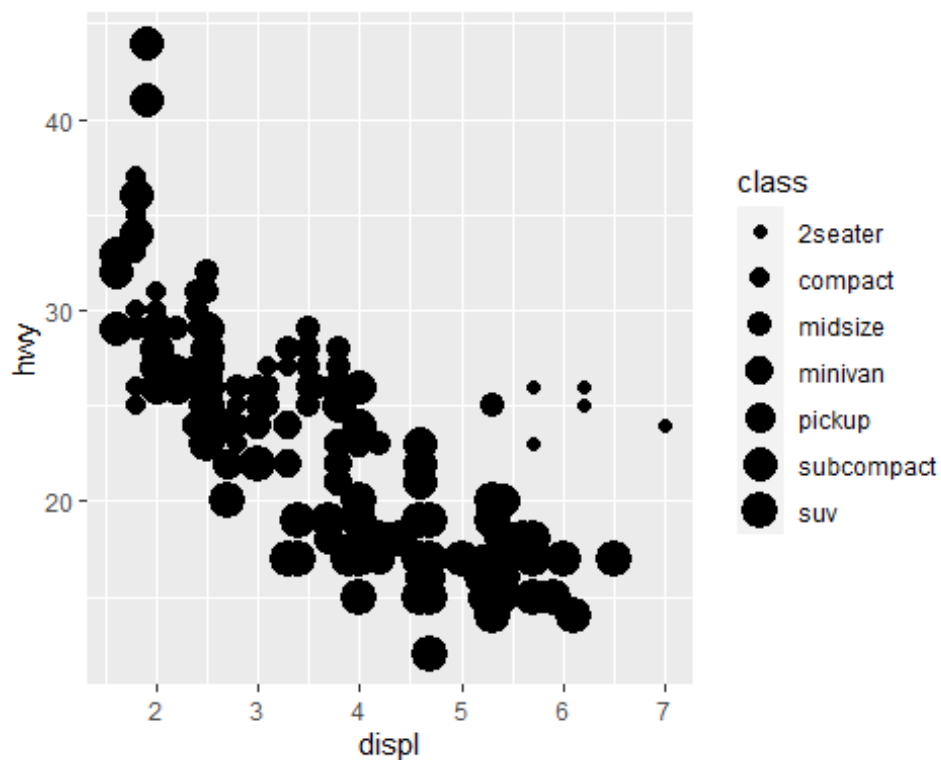


Εικόνα 18

Για να χαρτογραφήσουμε μια αισθητική σε μια μεταβλητή, συσχετίζουμε το όνομα της αισθητικής με το όνομα της μεταβλητής στην `aes()`. Η `ggplot2` θα εκχωρήσει αυτόματα ένα μοναδικό επίπεδο αισθητικής (ένα μοναδικό χρώμα) σε κάθε μοναδική τιμή της μεταβλητής, μια διαδικασία γνωστή ως κλιμάκωση (*scale*). Η `ggplot2` θα προσθέσει επίσης λεζάντα που εξηγεί ποια επίπεδα αντιστοιχούν, σε ποιες τιμές.

Στο παραπάνω παράδειγμα, αναλύθηκε την μεταβλητή *class* με χρώματα, αλλά θα μπορούσε να αντιστοιχιστεί και με αλλαγή μεγέθους. Σε αυτήν την περίπτωση ελέγχουμε το μέγεθος κάθε σημείου. Λαμβάνουμε μια προειδοποίηση εδώ, γιατί η αντιστοίχιση μιας μη ταξινομημένης μεταβλητής (*class*) σε μια ταξινομημένη αισθητική (*size*) δεν συνιστάται από την R.

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy, size = class))  
Warning: Using size for a discrete variable is not advised.
```

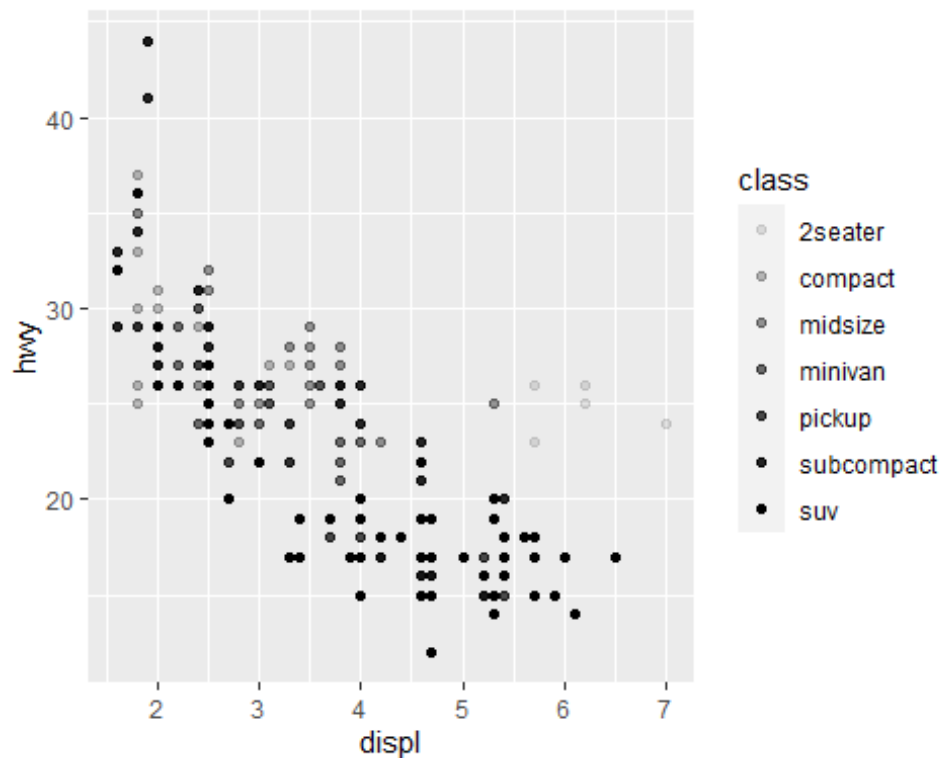


Εικόνα 19

Θα μπορούσε επίσης να αντιστοιχιστεί η *class* με το όρισμα *alpha*, το οποίο ελέγχει τη διαφάνεια των σημείων,

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy, alpha = class))
```

Warning: Using alpha for a discrete variable is not advised.



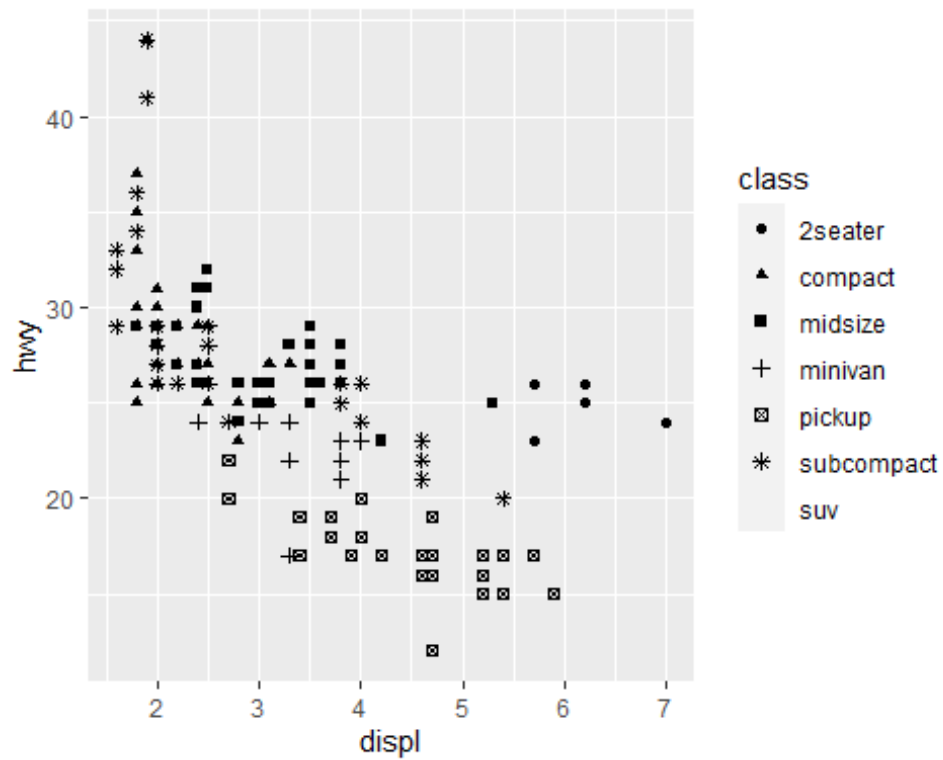
Εικόνα 20

ή με το όρισμα `shape` , το οποίο ελέγχει το σχήμα των σημείων.

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy, shape = class))
```

Warning: The shape palette can deal with a maximum of 6 discrete values because more than 6 becomes difficult to discriminate; you have 7. Consider specifying shapes manually if you must have them.

Warning: Removed 62 rows containing missing values (geom_point).



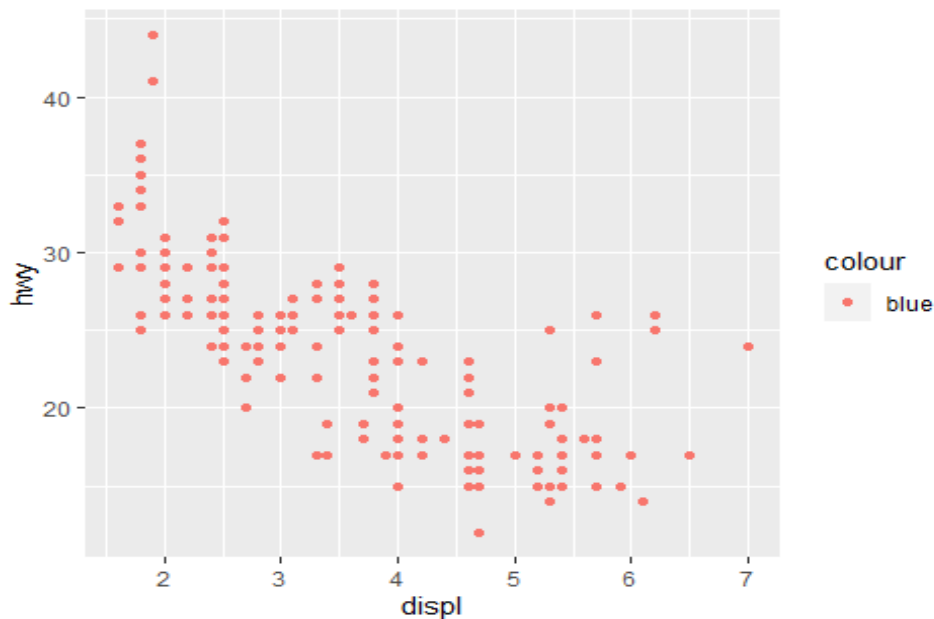
Εικόνα 21

Η ggplot2 χρησιμοποιεί μόνο έξι σχήματα κάθε φορά. Από προεπιλογή, οι πρόσθετες ομάδες απορρίπτονται.

Διερεύνηση συντακτικών προβλημάτων

1. Τι πήγε στραβά με αυτόν τον κώδικα; Γιατί τα σημεία δεν είναι μπλε;

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy, color = "blue"))
```



Εικόνα 22

Το `colour = "blue"` περιλαμβάνεται στο `mapping`, στο όρισμα `aes()`. Συνεπώς η έκφραση, `colour = "blue"`, “blue” ερμηνεύεται ως κατηγορική μεταβλητή η οποία λαμβάνει μόνο μία απλή τιμή “blue”. Η λύση είναι να βγεί από το όρισμα `aes()`.

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy), color = "blue")
```

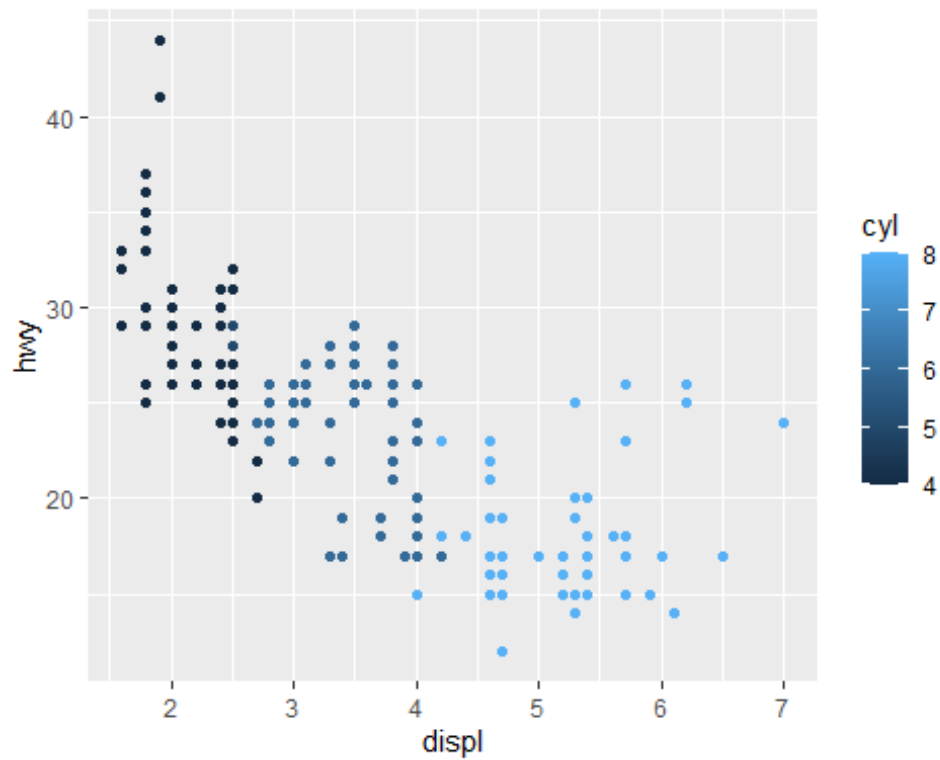
2. Ποιες μεταβλητές `mpg` είναι κατηγορηματικές και ποιες είναι συνεχείς; Πώς μπορείτε να δείτε αυτές τις πληροφορίες όταν εκτελείτε `mpg`;
(Συμβουλή: πληκτρολογήστε `??mpg` για να διαβάσετε την τεκμηρίωση για το σύνολο δεδομένων).

Όσες μεταβλητές έχουν `<chr>` πάνω από τις στήλες τους είναι κατηγορηματικές, ενώ με `<dbl>` ή `<int>` είναι συνεχείς. Η `glimpse()` είναι μια άλλη συνάρτηση που εμφανίζει συνοπτικά τον τύπο κάθε στήλης στο πλαίσιο δεδομένων:

```
glimpse(mpg)
```

Συνεχής μεταβλητή με `color`, `size` και `shape`. Πώς συμπεριφέρονται οι κατηγορηματικές έναντι συνεχών μεταβλητών;

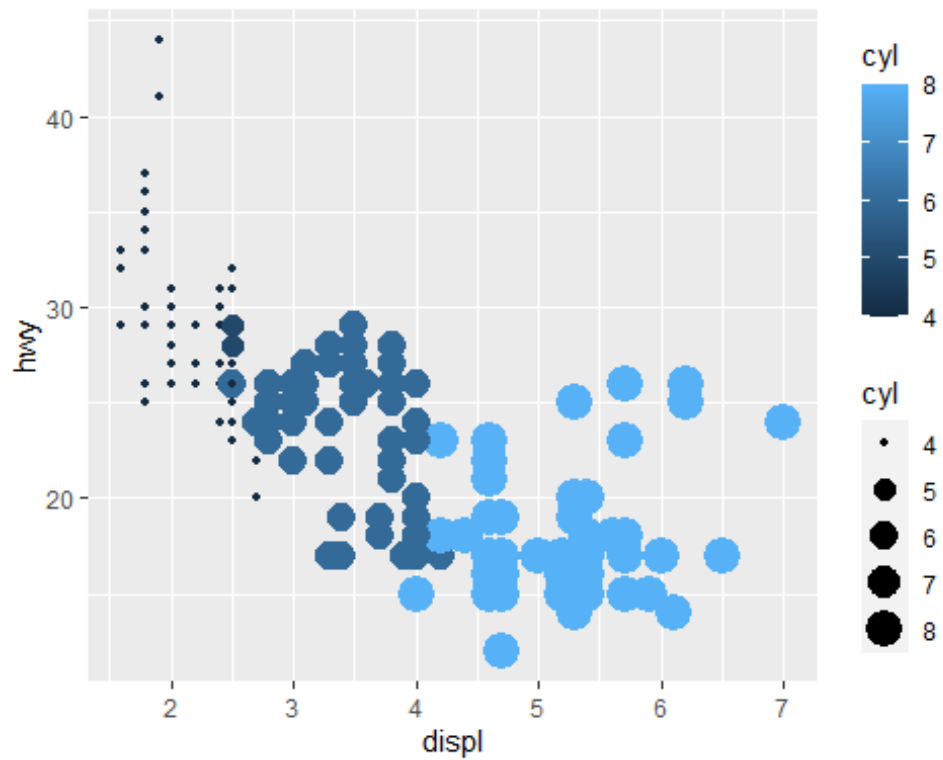
```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy, color = cyl ))
```



Εικόνα 23

3. Τι συμβαίνει εάν αντιστοιχίσετε την ίδια μεταβλητή σε πολλαπλή αισθητική;

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy, color = cyl,size=cyl ))
```



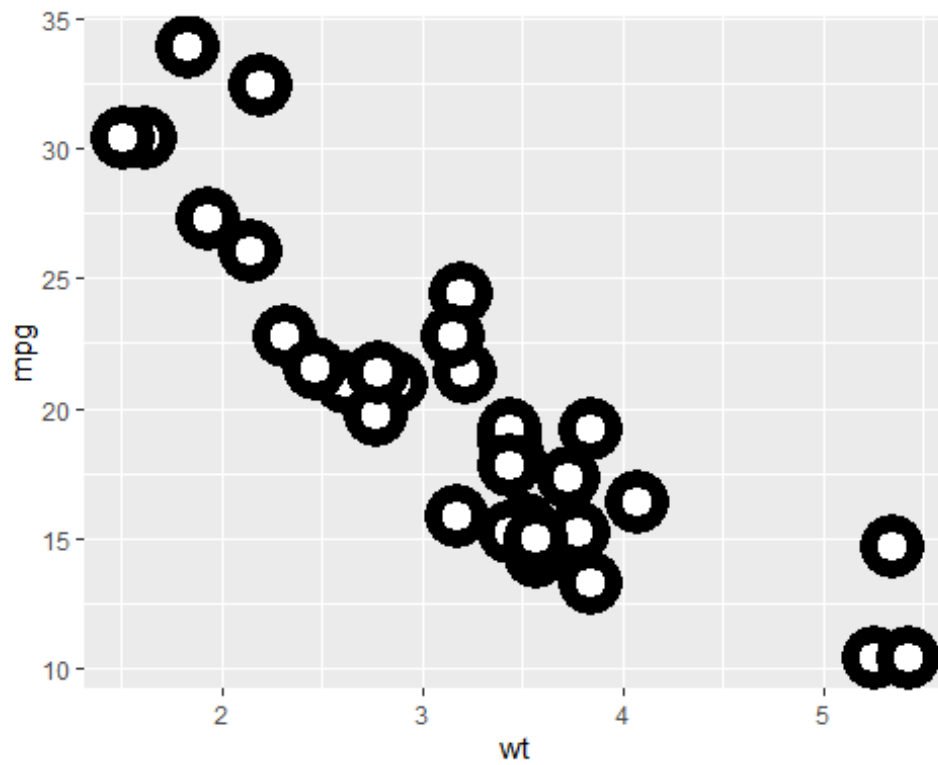
Εικόνα 24

4. Τι κάνει η stroke ;

(Συμβουλή: χρήση `?geom_point`)

#αλλάζει το μέγεθος του περιγράμματος

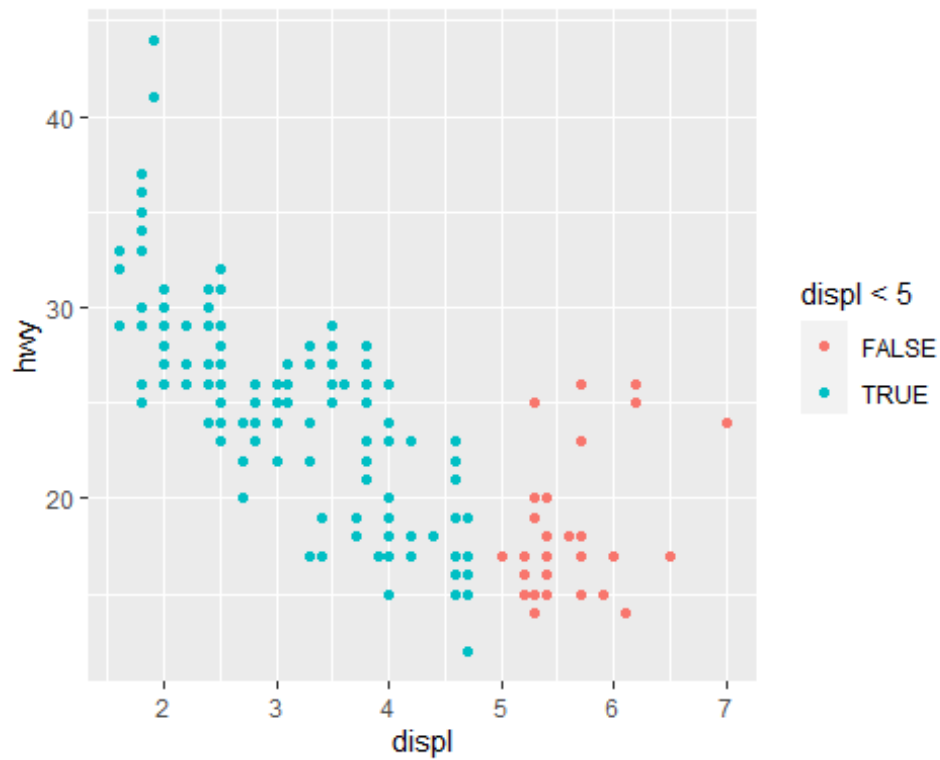
```
ggplot(mtcars, aes(wt, mpg)) +  
  geom_point(shape = 21, colour = "black", fill = "white", size = 5, stroke = 5)
```



Εικόνα 25

5. Τι συμβαίνει αν χαρτογραφήσετε μια αισθητική σε κάτι διαφορετικό από ένα μεταβλητό όνομα, όπως `aes(colour = displ < 5)`;

```
ggplot(mpg, aes(x = displ, y = hwy, colour = displ < 5)) +  
  geom_point()
```



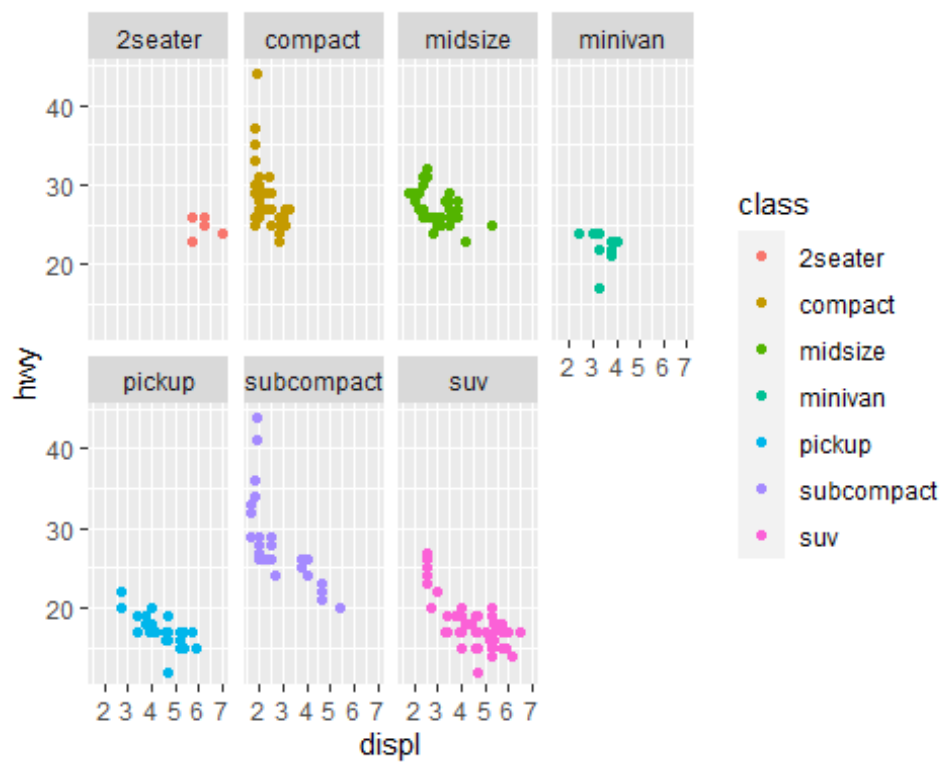
Εικόνα 26

Όψεις

Οι όψεις (facets) χωρίζουν το διάγραμμα σε υπό-περιοχές που η κάθε μία εμφανίζει ένα υποσύνολο των δεδομένων.

Για να τοποθετηθεί η κάθε μία μεταβλητή ξεχωριστά χρησιμοποιούμε το `facet_wrap()`.

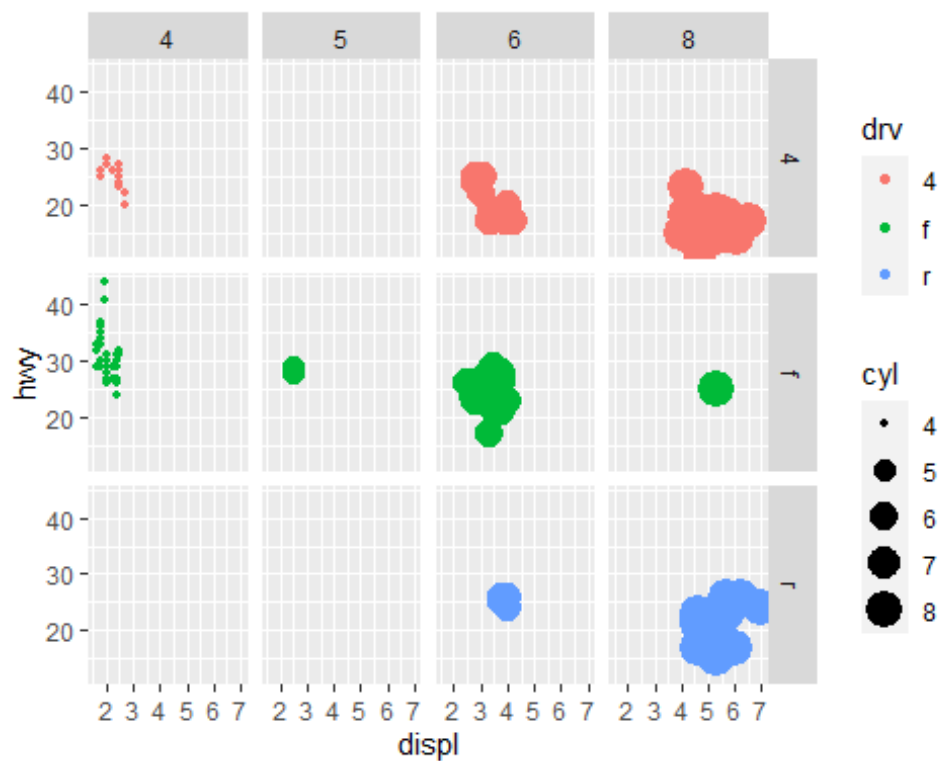
```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy, color = class)) +  
  facet_wrap(~ class, nrow = 2)
```



Εικόνα 27

Για να ενσωματώσουμε στην πλοκή συνδυασμό δύο μεταβλητών, επιλέγουμε `facet_grid()` στην κλήση της πλοκής.

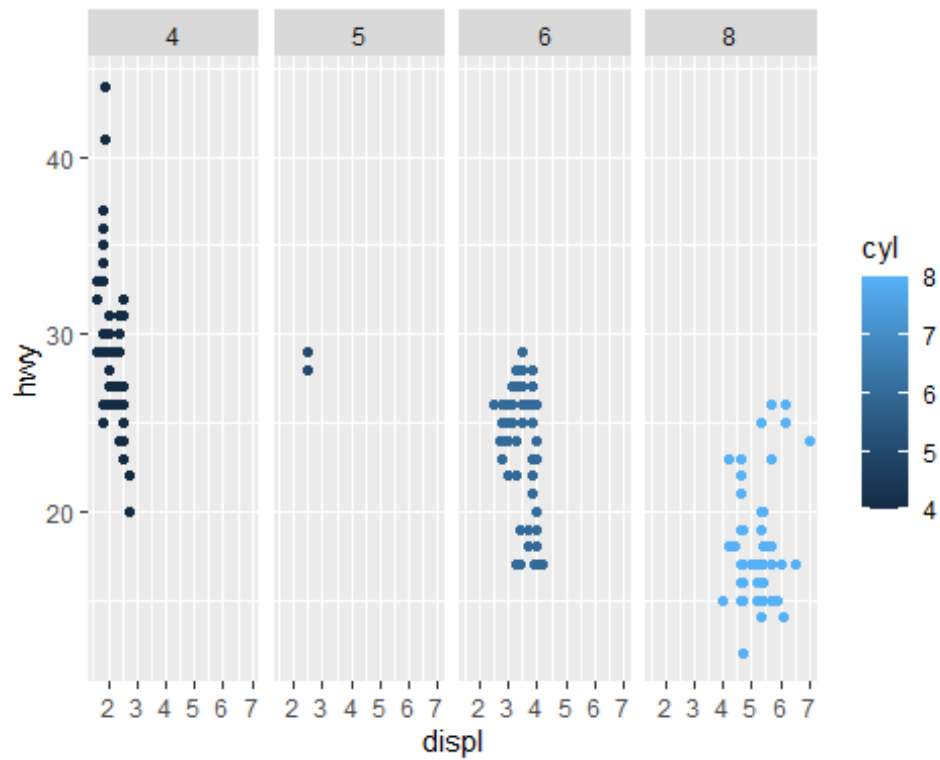
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color=drv, size=cyl)) +
  facet_grid(drv ~ cyl)
```



Εικόνα 28

Αν προτιμάτε να μην διαχωρίζονται οι γραμμές και οι στήλες στη διασταύρωσή τους , χρησιμοποιήστε μία “.” αντί για ένα όνομα μεταβλητής, π.χ. + facet_grid(. ~ cyl).

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color=cyl)) +
  facet_grid(.~ cyl)
```

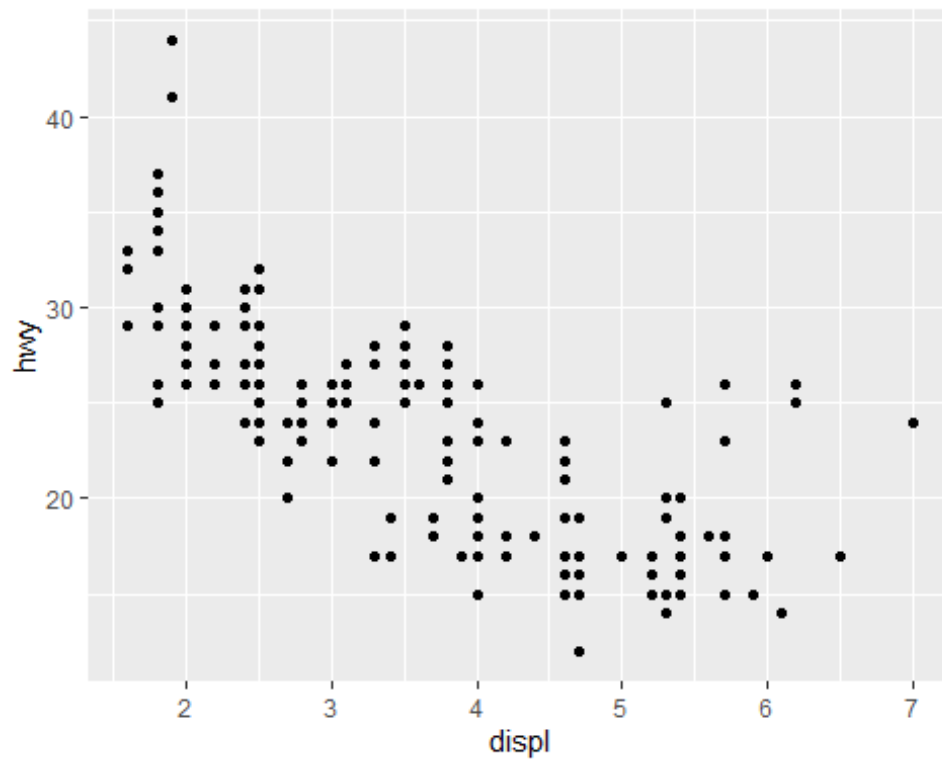



Εικόνα 29

Γεωμετρικά αντικείμενα

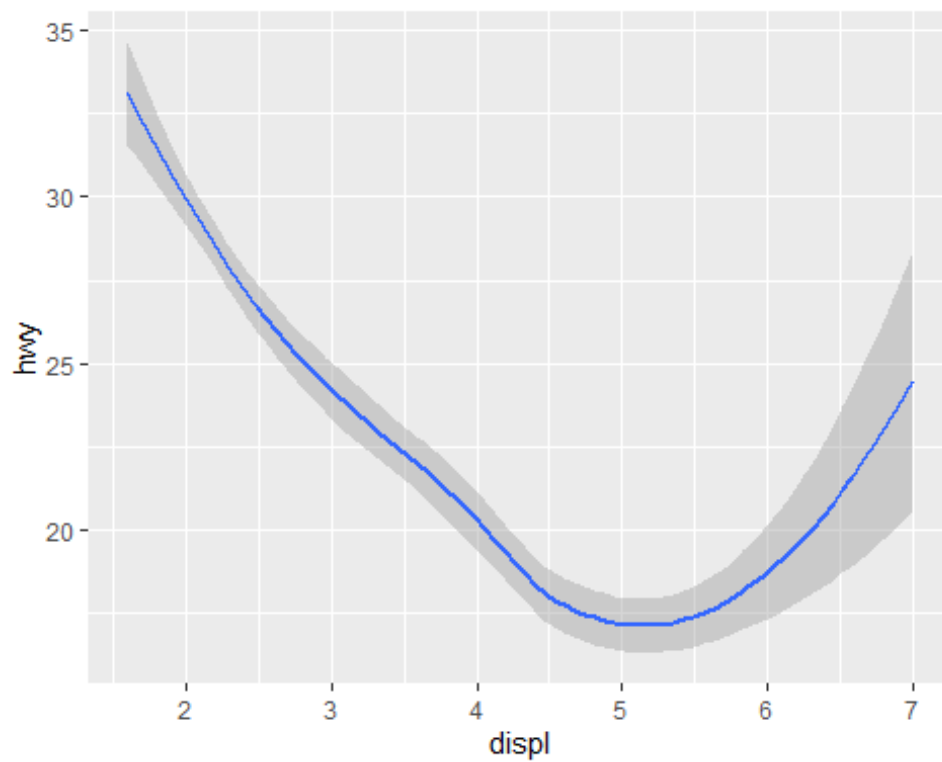
Κάθε πλοκή μπορεί να χρησιμοποιεί ένα διαφορετικό οπτικό αντικείμενο για την αναπαράσταση των δεδομένων (geoms).

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy))
```



Εικόνα 30

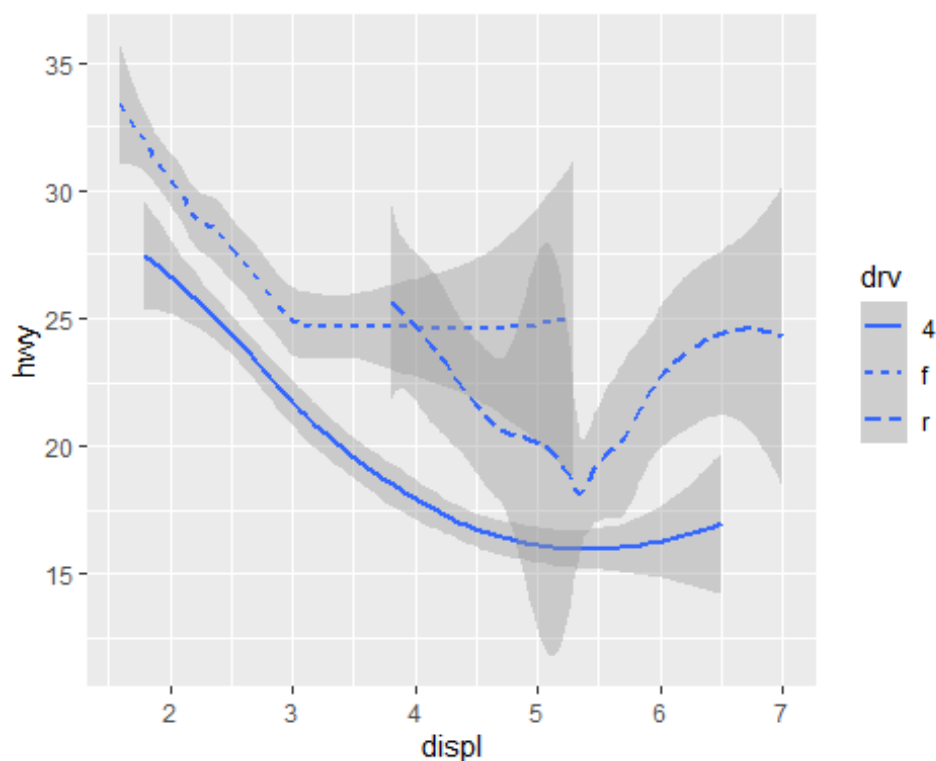
```
ggplot(data = mpg) +  
  geom_smooth(mapping = aes(x = displ, y = hwy))  
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Εικόνα 31

Μπορείτε να ορίσετε το `linetype` μιας γραμμής στο `geom_smooth()` και θα σχεδιάσει μια διαφορετική γραμμή, με διαφορετικό τύπο `linetype`, για κάθε μοναδική τιμή της μεταβλητής που αντιστοιχίζετε στο `linetype` όπως παρακάτω :

```
ggplot(data = mpg) +  
  geom_smooth(mapping = aes(x = displ, y = hwy, linetype = drv))  
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



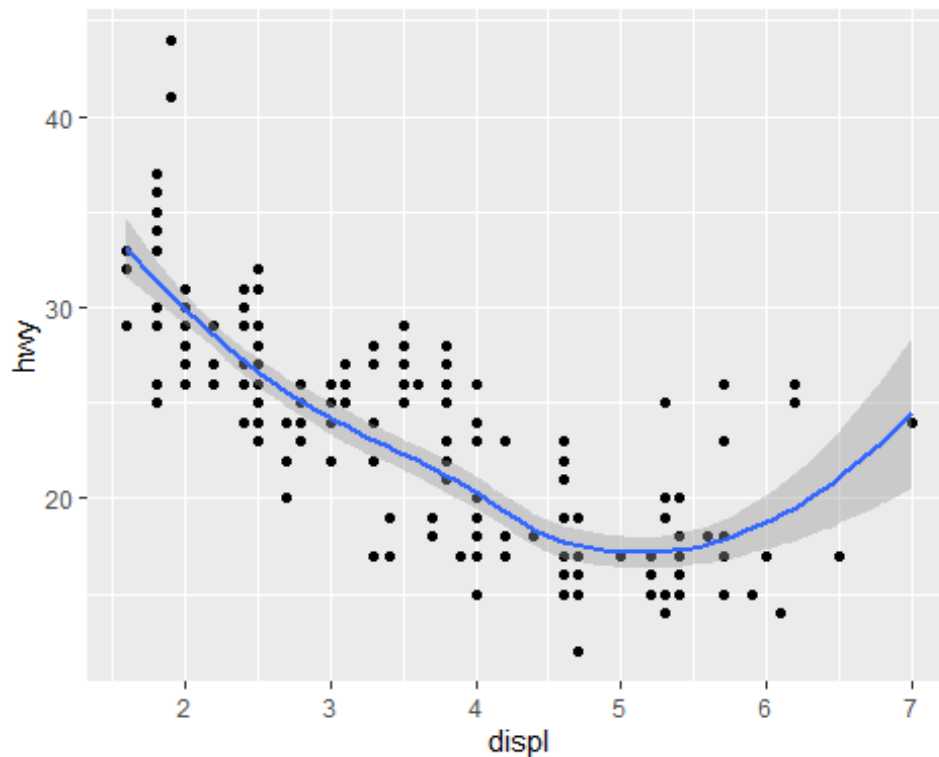
Εικόνα 32

Εδώ `geom_smooth()` χωρίζει τα αυτοκίνητα σε τρεις γραμμές με βάση την `drv` μεταβλητή, η οποία περιγράφει το σύστημα μετάδοσης κίνησης ενός αυτοκινήτου. Το 4 σημαίνει τετράτροχη κίνηση, f για κίνηση μπροστά και r για κίνηση πίσω.

Το `ggplot2` παρέχει πάνω από 40 `geom` και τα πακέτα επέκτασης παρέχουν ακόμη περισσότερα (δείτε <https://exts.ggplot2.tidyverse.org/gallery/>). Ο καλύτερος τρόπος για να λάβετε μια ολοκληρωμένη επισκόπηση είναι το cheatsheet `ggplot2`, το οποίο μπορείτε να βρείτε στη διεύθυνση <http://rstudio.com/resources/cheatsheets>.

Για να εμφανίσετε πολλά `geom` στην ίδια γραφική παράσταση, προσθέστε πολλές συναρτήσεις `geom` στο `ggplot()` όπως παρακάτω:

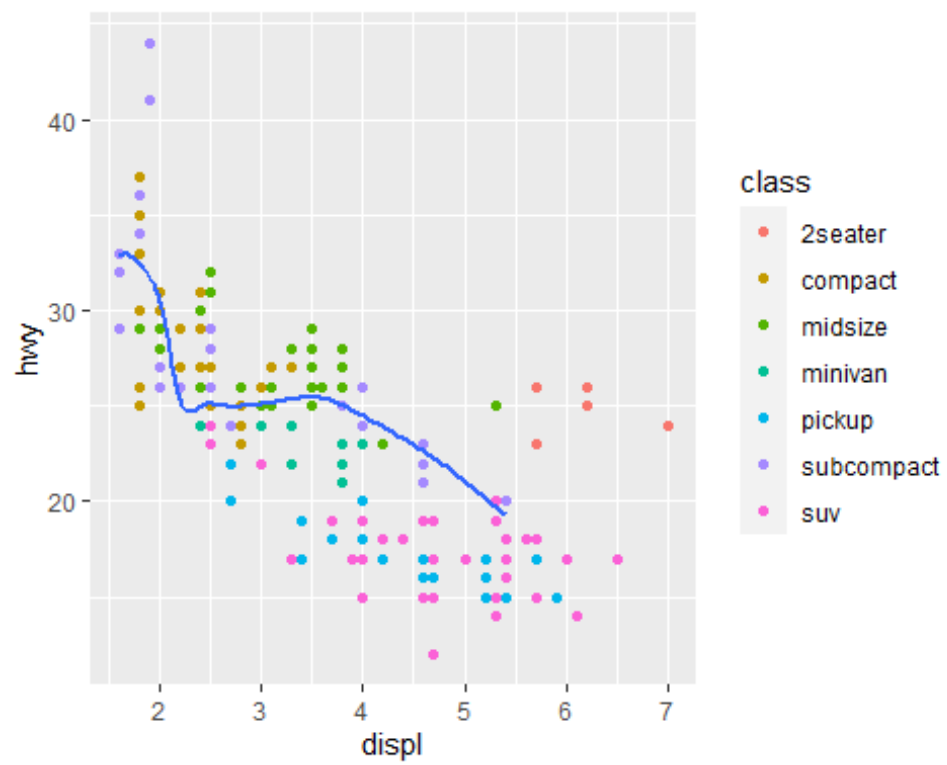
```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point() +
  geom_smooth()
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Εικόνα 33

Μπορείτε να χρησιμοποιήσετε την ίδια ιδέα για να καθορίσετε διαφορετικά data για κάθε επίπεδο. Εδώ, η ομαλή γραμμή μας εμφανίζει μόνο ένα υποσύνολο του mpg, τα αυτοκίνητα subcompact.

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = class)) +
  geom_smooth(data = filter(mpg, class == "subcompact"), se = FALSE)
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Εικόνα 34

(<https://r4ds.had.co.nz/data-visualisation.html>,2018)

2.9 Μάθημα 9ο

Επισκόπηση

Μέσος όρος, διάμεσος

Γραμμική παλινδρόμηση

Κανονική κατανομή

Δέντρο απόφασης

Παραδείγματα στατιστικών στοιχείων R

Σκοπός αυτού του κεφαλαίου είναι να δούμε την απεικόνισή βασικών στατιστικών υπολογισμών στην R

2.9.1 R - Μέσος όρος, διάμεσος

Η στατιστική ανάλυση στην R πραγματοποιείται χρησιμοποιώντας πολλές ενσωματωμένες συναρτήσεις. Οι περισσότερες από αυτές τις λειτουργίες αποτελούν μέρος του βασικού πακέτου R.

Μέσος Όρος

Υπολογίζεται λαμβάνοντας το άθροισμα των τιμών και διαιρώντας με τον αριθμό τιμών σε μια σειρά δεδομένων. Ο υπολογισμός στην R γίνεται με την function `mean()` και έχει την παρακάτω σύνταξη :

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

Όπου :

Το **x** είναι το διάνυσμα εισαγωγής.

Το **trim** χρησιμοποιείται για να ρίξει μερικές παρατηρήσεις και από τα δύο άκρα του ταξινομημένου διανύσματος.

Το **na.rm** χρησιμοποιείται για την αφαίρεση των τιμών που λείπουν από το διάνυσμα εισόδου.

Παράδειγμα

```
# Δημιουργία Διανύσματος
x <- c(12,7,3,4.2,18,2,54,-21,8,-5)
# Εύρεση Μέσου όρου
MO <- mean(x)
print(MO)
[1] 8.22
```

TRIM

Όταν παρέχεται η παράμετρος trim, οι τιμές στο διάνυσμα ταξινομούνται και, στη συνέχεια, οι απαιτούμενοι αριθμοί παρατηρήσεων μειώνονται από τον υπολογισμό του μέσου όρου.

Όταν το trim = 0,3, 3 τιμές από κάθε άκρο θα αφαιρεθούν από τους υπολογισμούς πριν την εύρεση του μέσου όρου.

Σε αυτήν την περίπτωση αν το διάνυσμα είναι (-21, -5, 2, 3, 4.2, 7, 8, 12, 18, 54) οι τιμές που αφαιρούνται από το διάνυσμα για τον υπολογισμό του μέσου όρου είναι (-21, -5, 2) από αριστερά και (12, 18, 54) από δεξιά.

```
x <- c(12,7,3,4.2,18,2,54,-21,8,-5)
MOT <- mean(x,trim = 0.3)
print(MOT)
[1] 5.55
```

Εφαρμογή NA

Εάν λείπουν τιμές, τότε η συνάρτηση mean() επιστρέφει NA.

Για να απορρίψετε τις τιμές που λείπουν από τον υπολογισμό χρησιμοποιήστε na.rm = TRUE. που σημαίνει αφαίρεση των τιμών NA.

```
x <- c(12,7,3,4.2,18,2,54,-21,8,-5,NA)
MONA <- mean(x)
print(MONA)
[1] NA
x <- c(12,7,3,4.2,18,2,54,-21,8,-5,NA)
MONA <- mean(x,na.rm = TRUE)
print(MONA)
[1] 8.22
```

Διάμεσος - Median

Ο διάμεσος είναι ο μεσαίος αριθμός σε μια ταξινομημένη, αύξουσα ή φθίνουσα, λίστα αριθμών και μπορεί να είναι πιο περιγραφικός του συνόλου δεδομένων από τον μέσο όρο. Ο διάμεσος χρησιμοποιείται μερικές φορές σε αντίθεση από τον μέσο όρο όταν υπάρχουν ακραίες τιμές στην ακολουθία που μπορεί να παραμορφώσουν το μέσο όρο των τιμών.

Για να προσδιοριστεί η διάμεση τιμή σε μια ακολουθία αριθμών, οι αριθμοί πρέπει πρώτα να ταξινομηθούν με αύξουσα ή φθίνουσα ταξινόμηση. Ο διάμεσος μπορεί να χρησιμοποιηθεί για τον προσδιορισμό ενός κατά προσέγγιση μέσου όρου, αλλά δεν πρέπει να συγχέεται με τον πραγματικό μέσο όρο. Εάν υπάρχει ζυγός αριθμός αριθμών στη λίστα, το μεσαίο ζεύγος πρέπει να προσδιοριστεί, αν προστεθεί και διαιρεθεί με δύο για να βρεθεί η μέση τιμή. (<https://www.investopedia.com/terms/m/median.asp> , 2019)

Στην R για τον υπολογισμό του διαμέσου χρησιμοποιούμε την συνάρτηση `median()`. Η βασική σύνταξη είναι **`median(x, na.rm = TRUE)`** όπου:

Το **`x`** είναι το διάνυσμα εισαγωγής.

Το **`na.rm`** χρησιμοποιείται για την αφαίρεση των τιμών που λείπουν από το διάνυσμα εισόδου.

Παράδειγμα :

```
x <- c(12,7,3,4.2,18,2,54,-21,8,-5,NA)
Mymed <- median(x,na.rm=TRUE)
print(Mymed)
[1] 5.6
```

2.9.2 R - Γραμμική παλινδρόμηση - R Linear Regression

Η ανάλυση παλινδρόμησης είναι ένα ευρέως χρησιμοποιούμενο στατιστικό εργαλείο για τη δημιουργία ενός μοντέλου σχέσης μεταξύ δύο μεταβλητών. Μία από αυτές τις μεταβλητές ονομάζεται μεταβλητή πρόβλεψης, της οποίας η τιμή συλλέγεται μέσω πειραμάτων. Η άλλη μεταβλητή ονομάζεται μεταβλητή απόκρισης της οποίας η τιμή

προέρχεται από τη μεταβλητή πρόβλεψης.(<https://www.r-bloggers.com/2009/11/r-tutorial-series-simple-linear-regression/,2019>)

Η γενική μαθηματική εξίσωση για μια γραμμική παλινδρόμηση είναι $y = ax + b$ όπου :

y είναι η μεταβλητή απόκρισης.

x είναι η μεταβλητή πρόβλεψης.

a και b είναι σταθερές που ονομάζονται συντελεστές.

Παράδειγμα

Ένα απλό παράδειγμα παλινδρόμησης είναι η πρόβλεψη του βάρους ενός ατόμου όταν το ύψος του είναι γνωστό. Για να γίνει αυτό πρέπει να έχουμε τη σχέση μεταξύ ύψους και βάρους ενός ατόμου.

```
# Δεδομένα ύψους
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

# Δεδομένα Βάρους
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

# Δημιουργία συσχέτισης μέσω της συνάρτησης lm()
sxesi <- lm(y~x)
```

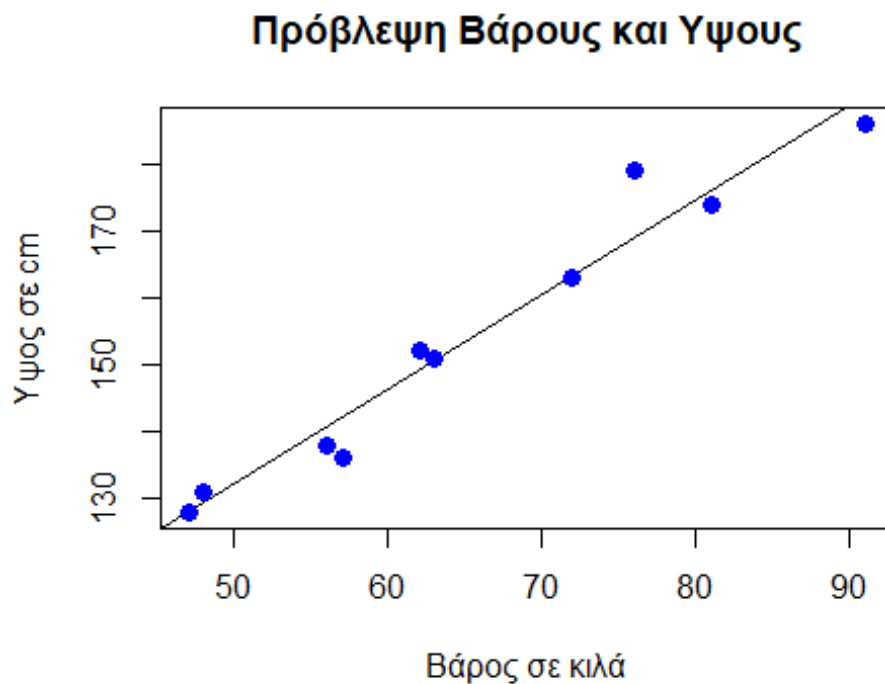
Η συνάρτηση lm() Linear Regration δημιουργεί ένα απλό μοντέλο παλινδρόμησης της μορφής $y = ax + b$

Στη συνέχεια μπορούμε να προβλέψουμε το βάρος y δίνοντας οποιοδήποτε ύψος x. Για την πρόβλεψη χρησιμοποιούμε την συνάρτηση predict() όπως παρακάτω :

```
a <- data.frame(x = 170)
result <- predict(sxesi,a)
print(result)
1
76.22869
```

Γραφική παράσταση

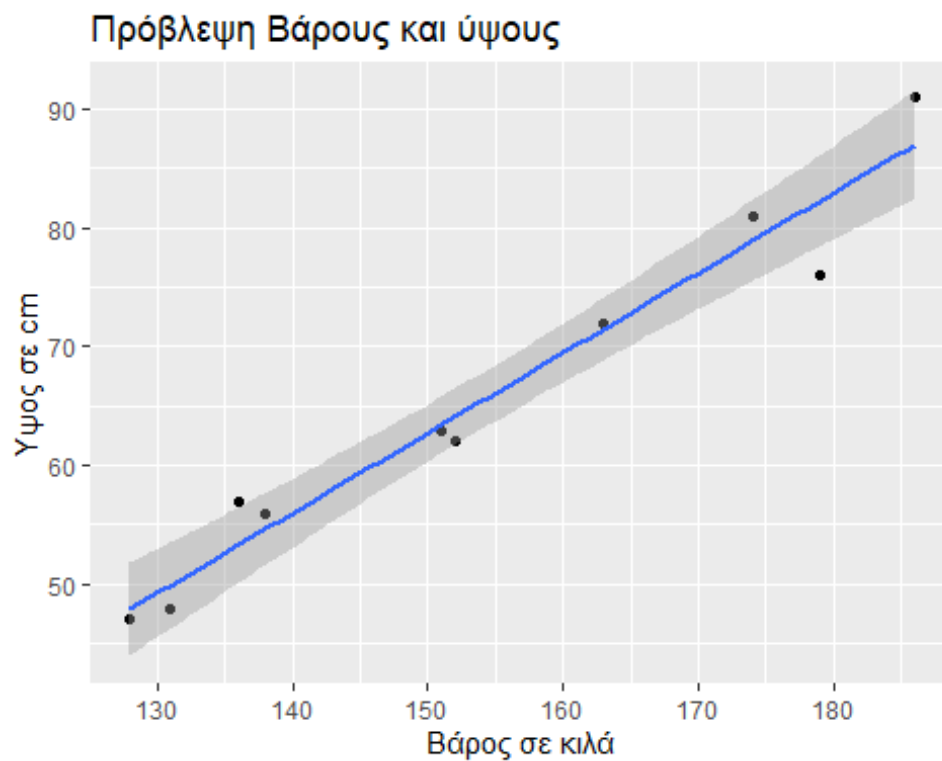
```
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
relation <- lm(y~x)
plot(y,x,col = "blue",main = "Πρόβλεψη Βάρους και Υψους",
abline(lm(x~y)),cex = 1.3,pch = 16,xlab = "Βάρος σε κιλά",ylab = "Υψος σε cm")
```



Εικόνα 35

Με τη ggplot στο ίδιο παράδειγμα αρκεί να δημιουργήσουμε ένα data.frame με τα δεδομένα μας.

```
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
mydata <- data.frame("x" = x, "y" = y, "Name" = c("x", "y"))
ggplot(mydata,aes(x,y))+
  geom_point()+
  geom_smooth(method = "lm")+
  ggtitle("Πρόβλεψη Βάρους και ύψους") + ylab("Υψος σε cm")+ xlab("Βάρος σε κιλά")
`geom_smooth()` using formula 'y ~ x'
```



Εικόνα 36

2.9.3 R - Κανονική κατανομή

Η R έχει τέσσερις ενσωματωμένες λειτουργίες για τη δημιουργία κανονικής κατανομής.

`dnorm(x, mean, sd)` `pnorm(x, mean, sd)` `qnorm(p, mean, sd)` `rnorm(n, mean, sd)`

Ακολουθεί η περιγραφή των παραμέτρων που χρησιμοποιούνται στις παραπάνω συναρτήσεις :

Το **x** είναι ένα διάνυσμα αριθμών.

Το **p** είναι ένα διάνυσμα πιθανοτήτων.

n είναι ο αριθμός των παρατηρήσεων (μέγεθος δείγματος).

mean είναι η μέση τιμή του δείγματος δεδομένων, με default τιμή το μηδέν.

sd είναι η τυπική απόκλιση, με default τιμή το 1.

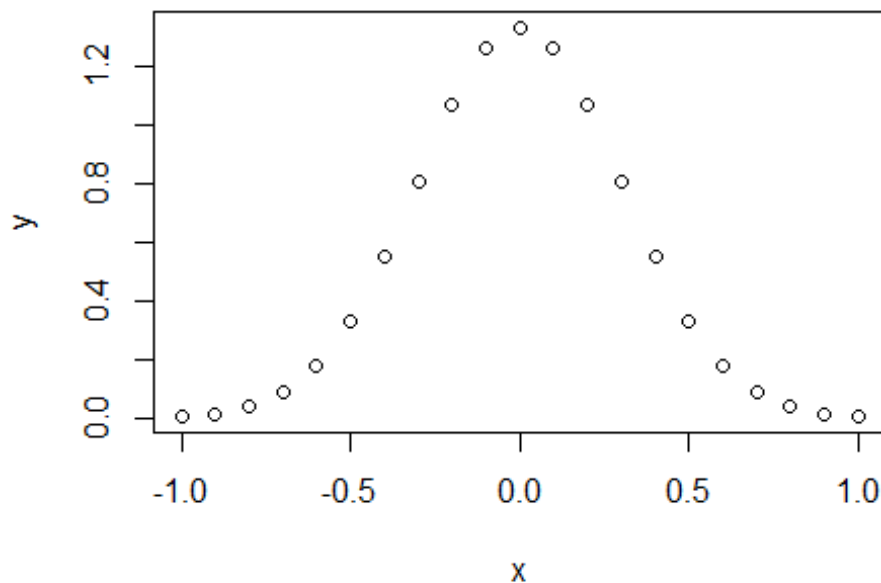
dnorm() Αυτή η συνάρτηση δίνει το ύψος της κατανομής της πιθανότητας σε κάθε σημείο για μια δεδομένη μέση και τυπική απόκλιση.

Παράδειγμα

```
# Δημιουργούμε μια ακολουθία αριθμών από το -1 ως το 1 με απόσταση μεταξύ τους 0.1  
x <- seq(-1, 1, by = .1)
```

```
# Επιλέγουμε την μέση τιμή στο 0 και τυπική απόκλιση 0,3.  
y <- dnorm(x, mean = 0, sd = 0.3)
```

```
# Σχεδιάζουμε το ύψος της κατανομής  
plot(x,y)
```



Εικόνα 37

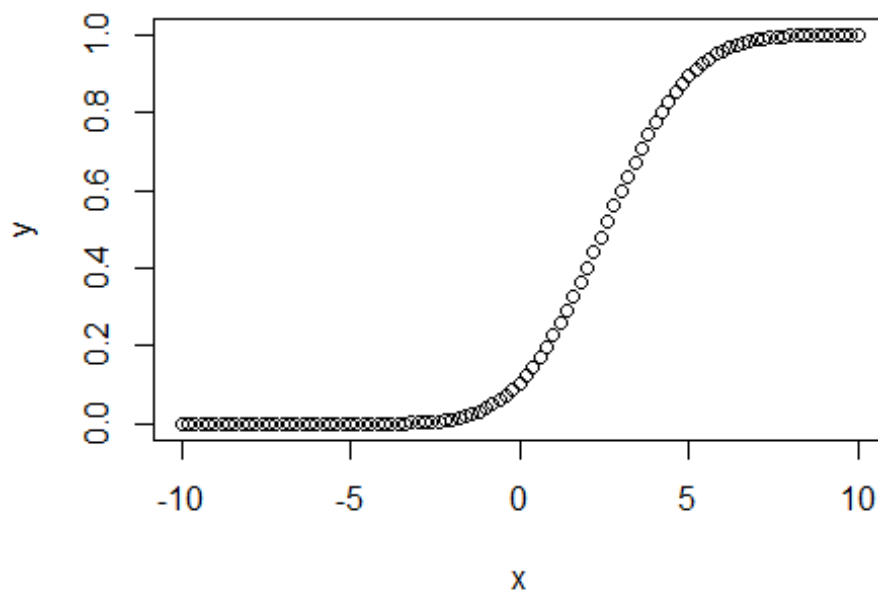
pnorm() Αυτή η συνάρτηση δείχνει την πιθανότητα ενός κανονικά κατανεμημένου τυχαίου αριθμού να είναι μικρότερη από την τιμή ενός δεδομένου αριθμού. Ονομάζεται επίσης “Λειτουργία αθροιστικής διανομής”.

```
# Δημιουργούμε μια ακολουθία αριθμών από το -10 ως το 10 με απόσταση μεταξύ τους 0.2  
x <- seq(-10,10,by = .2)
```

```
# Επιλέγουμε την μέση τιμή στο 2.5 και τυπική απόκλιση 2.
```

```
y <- pnorm(x, mean = 2.5, sd = 2)
```

```
# Σχεδιάζουμε το ύψος της κατανομής  
plot(x,y)
```



Εικόνα 38

qnorm () Αυτή η συνάρτηση λαμβάνει την τιμή πιθανότητας και δίνει έναν αριθμό του οποίου η αθροιστική τιμή ταιριάζει με την τιμή πιθανότητας.

```
# Δημιουργούμε μια ακολουθία αριθμών από το 0 ως 1 με απόσταση μεταξύ τους 0.02
```

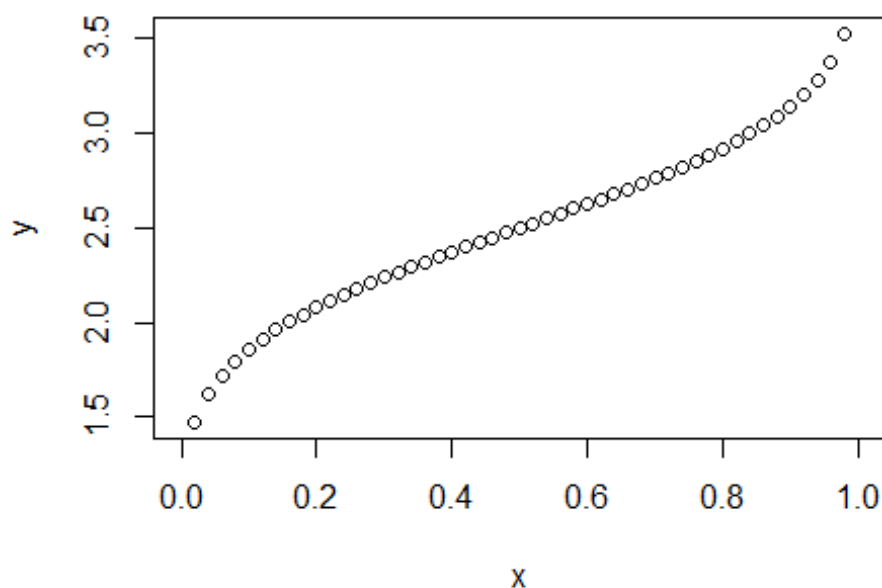
```
x <- seq(0, 1, by = 0.02)
```

```
# Επιλέγουμε μέση τιμή 2.5 και τυπική απόκλιση 0.5.
```

```
y <- qnorm(x, mean = 2.5, sd = 0.5)
```

```
# Σχεδιάζουμε το ύψος της κατανομής
```

```
plot(x,y)
```



Εικόνα 39

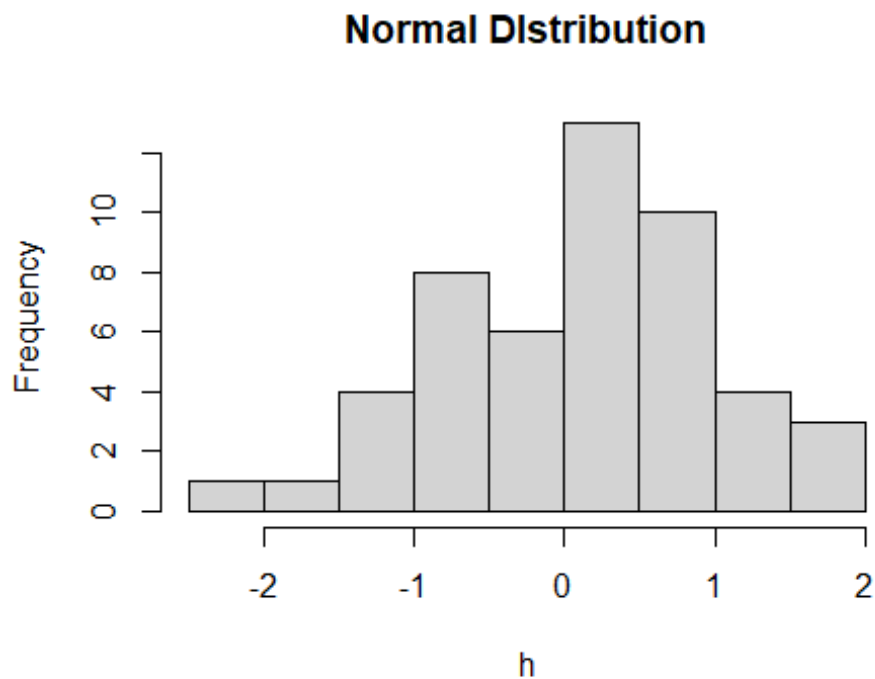
rnorm () Αυτή η συνάρτηση χρησιμοποιείται για τη δημιουργία τυχαίων αριθμών των οποίων η κατανομή είναι φυσιολογική. Παίρνει το μέγεθος του δείγματος ως είσοδο και δημιουργεί πολλούς τυχαίους αριθμούς. Σχεδιάζουμε ένα ιστόγραμμα για να δείξουμε την κατανομή των παραγόμενων αριθμών. (http://rstudio-pubs-static.s3.amazonaws.com/374857_5a23bad9783a43c1b102aa80aa5c1a7c.html,2020)

Create a sample of 50 numbers which are normally distributed.

```
h <- rnorm(50)
```

Plot the histogram for this sample.

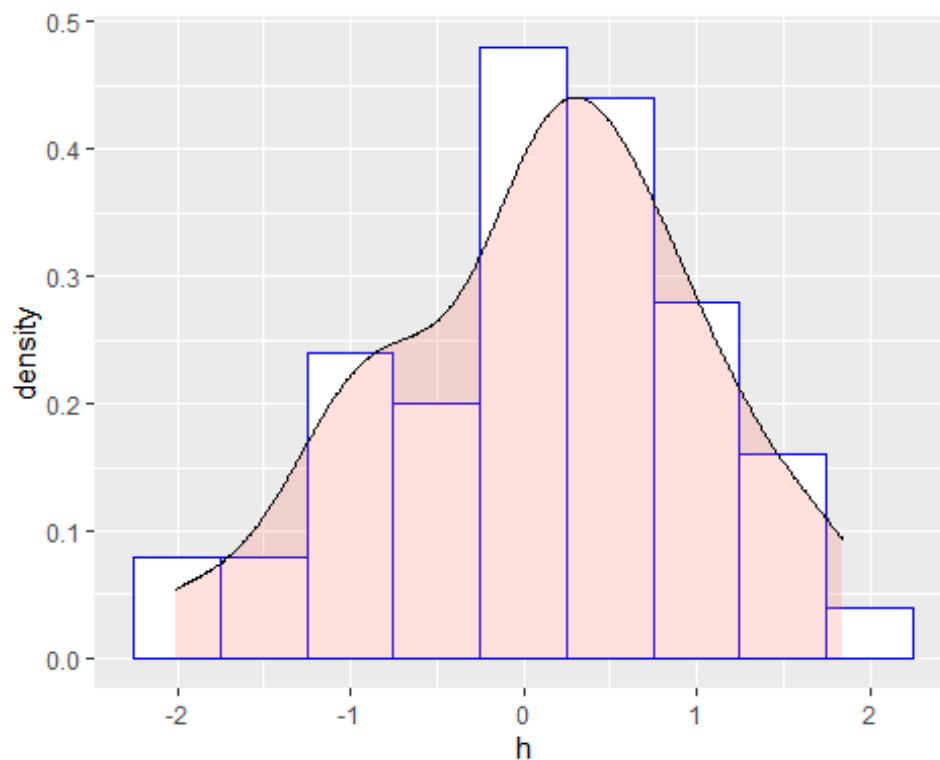
```
hist(h, main = "Normal Distribution")
```



Εικόνα 40

Ας δούμε το αντίστοιχο παράδειγμα με τη χρήση της ggplot :

```
data<-data.frame("x"=h)
ggplot(data, aes(x = h)) +
  geom_histogram(aes(y = ..density..), # Το Ιστόγραμμα θα εμφανίζει την πυκνότητα στον
y-axis
  binwidth = .5, colour = "blue", fill = "white") +
  geom_density(alpha = .2, fill="#FF6655") # Επικάλυψη με διαφάνεια της πυκνότητας
```



Εικόνα 41

2.9.4 R - Δέντρο απόφασης

Το δέντρο αποφάσεων είναι ένα γράφημα που απεικονίζει τις επιλογές και τα αποτελέσματά τους σε μορφή δέντρου. Οι κόμβοι στο γράφημα αντιπροσωπεύουν ένα συμβάν ή επιλογή και τα άκρα του γραφήματος αντιπροσωπεύουν τους κανόνες ή τις προϋποθέσεις απόφασης. Χρησιμοποιείται κυρίως σε εφαρμογές μηχανικής μάθησης και εξόρυξης δεδομένων.

Παραδείγματα χρήσης του τρόπου λήψης αποφάσεων είναι η πρόβλεψη ενός μηνύματος ηλεκτρονικού ταχυδρομείου ως ανεπιθύμητου, η πρόβλεψη ενός δανείου ως καλό ή κακό με βάση τον πιστωτικό κίνδυνο και γενικά, δημιουργείται ένα μοντέλο με παρατηρούμενα δεδομένα που ονομάζονται επίσης δεδομένα εκπαίδευσης. Στη συνέχεια χρησιμοποιείται ένα σύνολο δεδομένων επικύρωσης, για την επαλήθευση και τη βελτίωση του μοντέλου. Η R έχει πακέτα που χρησιμοποιούνται για τη δημιουργία και την απεικόνιση δέντρων αποφάσεων. Χρησιμοποιούμε αυτό το μοντέλο για να λάβουμε μια απόφαση σχετικά με την κατηγορία (Ναι / Όχι) των δεδομένων.

Το πακέτο **party** χρησιμοποιείται για τη δημιουργία δέντρων αποφάσεων. Αρχικά θα πρέπει να εγκαταστήσουμε το πακέτο.


```
library(party)
```

```
Loading required package: grid
```

```
Loading required package: mvtnorm
```

```
Loading required package: modeltools
```

```
Loading required package: stats4
```

```
Loading required package: strucchange
```

```
Loading required package: zoo
```

```
Attaching package: 'zoo'
```

```
The following objects are masked from 'package:base':
```

```
as.Date, as.Date.numeric
```

```
Loading required package: sandwich
```

```
Attaching package: 'strucchange'
```

```
The following object is masked from 'package:stringr':
```

```
boundary
```

Το πακέτο “**party**” περιλαμβάνει τη συνάρτηση **ctree ()** που χρησιμοποιείται για τη δημιουργία και την ανάλυση του δέντρου.

Η βασική σύνταξη για τη δημιουργία ενός δέντρου αποφάσεων στην R είναι **ctree(formula, data)**.

Παράδειγμα

Χρησιμοποιούμε δεδομένα του dataset readingSkills

```
print(head(readingSkills))
```

	native Speaker	age	shoeSize	score
1	yes	5	24.83189	32.29385
2	yes	6	25.95238	36.63105
3	no	11	30.42170	49.60593
4	yes	7	28.66450	40.28456
5	yes	11	31.88207	55.46085
6	yes	10	30.07843	52.83124

Θα δημιουργήσουμε ένα δέντρο απόφασης που θα περιγράφει το σκορ της ικανότητας ανάγνωσης κάποιου αν γνωρίζουμε τις μεταβλητές “ηλικία”, “αριθμό παπουτσιού”, “σκορ” και αν το άτομο είναι εγγενής ομιλητής ή όχι.

```
# Μεταφέρουμε δεδομένα σε ένα data.frame
```

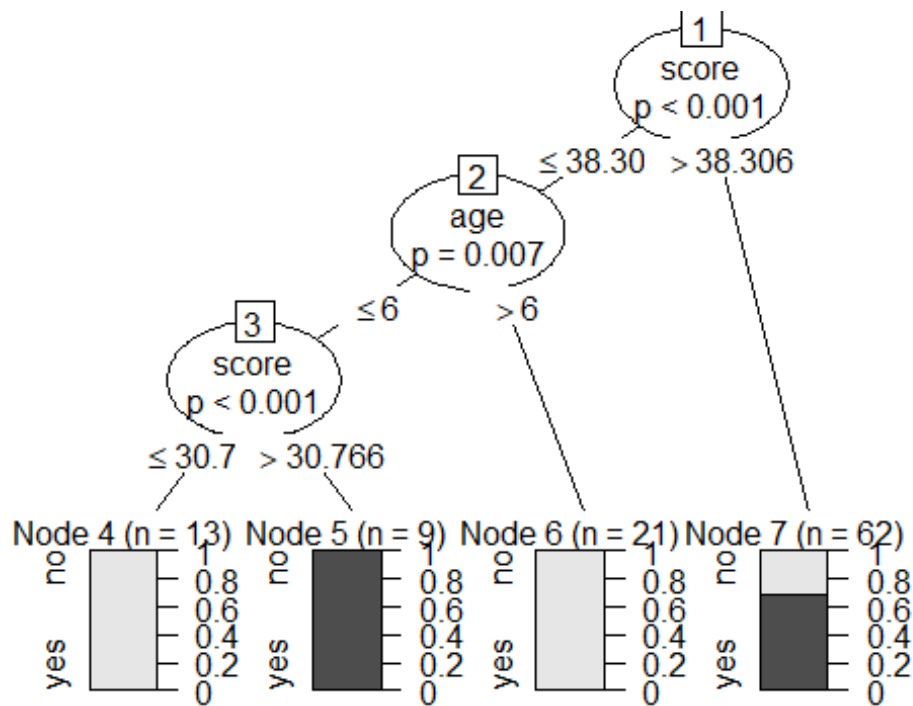
```
Newdata <- readingSkills[c(1:105),]
```

```
# Δημιουργούμε ένα δένδρο
```

```
mytree <- ctree(nativeSpeaker ~ age + shoeSize + score,  
data = Newdata)
```

```
# Plot the tree.
```

```
plot(mytree)
```



Εικόνα 42

Από το δέντρο αποφάσεων που φαίνεται παραπάνω μπορούμε να συμπεράνουμε ότι όποιος έχει σκορ μικρότερο από 38,3 και η ηλικία είναι άνω των 6 δεν είναι εγγενής ομιλητής.

Το παραπάνω παράδειγμα με χρήση της ggplot:

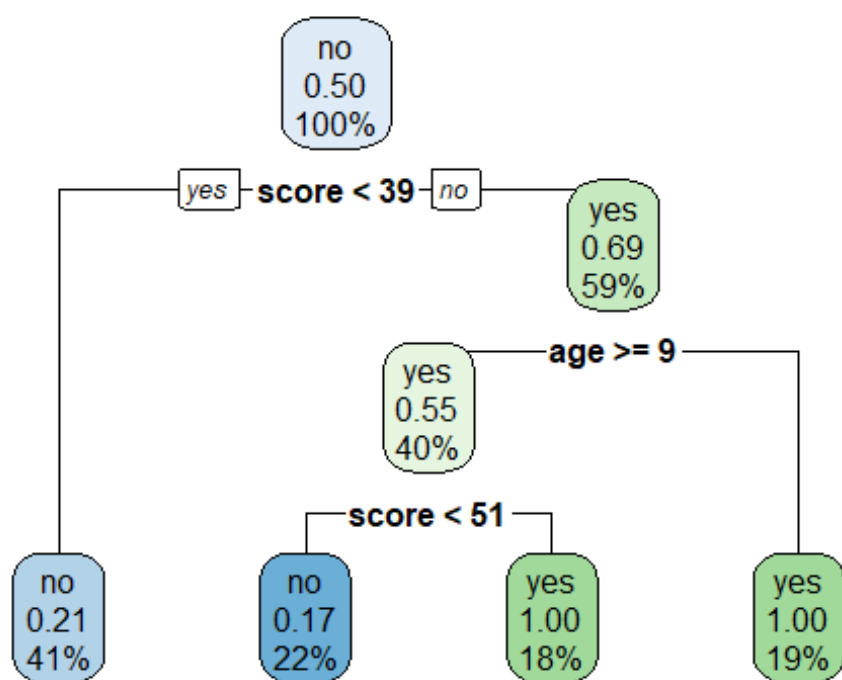
```
# Modeling packages
```

```
library(rpart) # Αυτόματη μηχανή για δημιουργία δέντρων απόφασης
```

```
library(rpart.plot) # Σχεδιασμός δέντρων αποφάσεων
```

```
fit <- rpart(nativeSpeaker ~ age + shoeSize + score, data = Newdata)
```

```
rpart.plot(fit)
```



Εικόνα 43

2.10 Μάθημα 10ο

Επισκόπηση R Markdown

Βιντεοσκοπημένο Μάθημα :

R Markdown <https://vimeo.com/manage/videos/586665962>

2.10.1 Markdown

Το Markdown είναι μια μορφή απλού κειμένου για τη σύνταξη δομημένων εγγράφων, βάσει συμβάσεων μορφοποίησης από email και usenet. Αναπτύχθηκε το 2004 από τον John Gruber σε συνεργασία με τον Aaron Swartz . Ο Gruber έγραψε τον πρώτο μετατροπέα markdown-to-html σε Perl και σύντομα χρησιμοποιήθηκε ευρέως σε ιστότοπους.

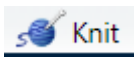
Το **R Markdown** παρέχει ένα ενοποιημένο πλαίσιο σύνταξης για την επιστήμη των δεδομένων, συνδυάζοντας στον κώδικά σας, τα αποτελέσματά του και τον σχολιασμό. Τα έγγραφα R Markdown είναι αναπαραγώγιμα και υποστηρίζουν δεκάδες μορφές εξόδου, όπως PDF, αρχεία Word, προβολές διαφανειών και άλλα.

2.10.2 Βασικά στοιχεία του Markdown

Ένα αρχείο R Markdown, απλού κειμένου που έχει την επέκταση .Rmd, περιέχει τρεις σημαντικούς τύπους περιεχομένου:

- Μια (προαιρετική) κεφαλίδα YAML που περιβάλλεται από παύλες «—».
- Κομμάτια του κώδικα R που περιβάλλεται από άνω κόμμα ```.
- Κείμενο που αναμειγνύεται με απλή μορφοποίηση κειμένου όπως # heading και *italics*.

Όταν ανοίγετε ένα .Rmd αρχείο, λαμβάνετε μια διεπαφή σημειωματάριου όπου ο κώδικας και η έξοδος αλληλοεμπλέκονται. Μπορείτε να εκτελέσετε κάθε κομμάτι κώδικα κάνοντας κλικ στο εικονίδιο «Εκτέλεση» (μοιάζει με κουμπί αναπαραγωγής στο πάνω μέρος του κώδικα) ή πατώντας Ctrl + Shift + Enter. Το RStudio εκτελεί τον κώδικα και εμφανίζει τα αποτελέσματα.

Για να δημιουργήσετε μια πλήρη αναφορά που περιέχει όλο το κείμενο, τον κώδικα και τα αποτελέσματα, κάντε κλικ στο κουμπί «Knit»  ή πατήστε Ctrl + Shift + K. Το ίδιο μπορεί να γίνει με κώδικα, μέσω προγραμματισμού `rmarkdown::render("1-example.Rmd")`. Το RStudio θα εμφανίσει την αναφορά στο παράθυρο προβολής και θα δημιουργήσει ένα αυτόνομο αρχείο που μπορείτε να μοιραστείτε με άλλους.

Όταν δημιουργείται το έγγραφο, το R Markdown στέλνει το αρχείο .Rmd στο knitr , <http://yihui.name/knitr/> , το οποίο εκτελεί όλα τα κομμάτια κώδικα και δημιουργεί ένα νέο έγγραφο markdown (.md) που περιλαμβάνει τον κώδικα και το κείμενο. Στη συνέχεια, το αρχείο markdown που δημιουργείται από το knitr υποβάλλεται σε επεξεργασία από το pandoc , <http://pandoc.org/> , το οποίο είναι υπεύθυνο για τη δημιουργία του τελικού αρχείου. Το πλεονέκτημα αυτής της ροής εργασιών είναι ότι μπορείτε να δημιουργήσετε ένα πολύ ευρύ φάσμα μορφών εξόδου.

Για να ξεκινήσετε με το δικό σας .Rmd αρχείο, επιλέξτε Αρχείο=> Νέο αρχείο R Markdonw στο μενού. Το RStudio θα ξεκινήσει έναν οδηγό που μπορείτε να χρησιμοποιήσετε για να συμπληρώσετε το αρχείο σας με περιεχόμενο που σας υπενθυμίζει πως λειτουργούν τα βασικά χαρακτηριστικά του R Markdown.

Για την συγγραφή ολοκληρωμένων συγγραμμάτων με R Markdown είναι χρήσιμο να χρησιμοποιείται cheat Sheet όπως το:

<http://datascience.maths.unitn.it/events/reaction2017/day1/Cheatsheets/rmarkdown-cheatsheet-2.0.pdf>.

2.10.3 Βιβλιογραφίες και παραπομπές

Στο Pandoc μπορείτε να δημιουργήσετε αυτόματα παραπομπές και βιβλιογραφία σε διάφορα στυλ. Για να χρησιμοποιήσετε αυτήν τη δυνατότητα, καθορίστε ένα αρχείο βιβλιογραφίας χρησιμοποιώντας το **bibliography** πεδίο στην κεφαλίδα του αρχείου σας. Το πεδίο αυτό πρέπει να περιέχει μια διαδρομή από τον κατάλογο του αρχείου που περιέχει τη βιβλιογραφία πχ (bibliography: rmarkdown.bib)

Για να δημιουργήσετε μια αναφορά μέσα στο αρχείο .Rmd, χρησιμοποιήστε ένα κλειδί που αποτελείται από το «@» + το αναγνωριστικό παραπομπής από το αρχείο βιβλιογραφίας. Στη συνέχεια, τοποθετήστε την αναφορά σε αγκύλες.

Παράδειγμα: [[@αναφορά1](#)]

Διαχωρίστε με ερωτηματικό `;` διπλές αναφορές όπως : [[@αναφορά1](#); [@αναφορά2](#)].

Μπορείτε να προσθέσετε σχόλια εντός των αγκίστρων όπως :

[βλέπε [@αναφορά1](#); επίσης [@αναφορά2](#)].

Μπορείτε να συμπεριλάβετε μια αναφορά εντός του κειμένου σας χωρίς άγκιστρο απ' ευθείας με [@αναφορά1](#).

Το R Markdown, θα προσθέσει μια βιβλιογραφία στο τέλος του εγγράφου σας. Η βιβλιογραφία θα περιέχει κάθε μία από τις αναφερόμενες αναφορές από το αρχείο βιβλιογραφίας, αλλά δεν θα περιέχει επικεφαλίδα ενότητας. Ως αποτέλεσμα, είναι κοινή πρακτική να τερματίζετε το αρχείο σας με μια κεφαλίδα ενότητας για τη βιβλιογραφία, όπως **#References ή #Βιβλιογραφία.**

2.10.4 Εγκατάσταση R Markdown

Όπως και όλα στην R, το R Markdown είναι δωρεάν και ανοιχτού κώδικα. Μπορείτε να εγκαταστήσετε το πακέτο R Markdown από το CRAN με:

```
install.packages("rmarkdown")
```

Επιλέξτε File => New File => RMarkdown για την δημιουργία ενός νέου αρχείου σε rMarkdown. Στο παράθυρο επιλογής θα πρέπει να διαλέξετε από τα διαθέσιμα formats :

Document με επιλογή δημιουργίας αρχείου σε HTML, PDF και WORD

Presentation με επιλογή δημιουργίας αρχείου σε

- HTML (ioSlide) με δυνατότητα εμφάνισης io Slide σε PDF και Chrome
- HTML με δυνατότητα εμφάνισης Slide σε PDF και Chrome
- PDF που προϋποθέτει την ύπαρξη TeX
- PowerPoint

Shiny με δυνατότητα δημιουργίας διαδραστικών εγγράφων σε HTML και ioSlides

Και με χρήση διαφόρων Template(πρότυπα εγγράφων) όπου σε αυτό το σημείο υπάρχουν διαθέσιμα από την αρχική εγκατάσταση βασικά templates αλλά υπάρχει η δυνατότητα να φορτωθούν τα έτοιμα templates. Το πακέτο rtticles έχει σχεδιαστεί για να απλοποιήσει τη δημιουργία εγγράφων που συμμορφώνονται με τα πρότυπα υποβολής και είναι ένα καλό παράδειγμα προσαρμοσμένων προτύπων για μια σειρά περιοδικών (Yihui Xie, JJ Allaire, Garrett Grolmund, (2021)).

Εφόσον επιλέξετε από τα διαθέσιμα format στο περιβάλλον του RStudio δημιουργείται ένα νέο αρχείο με κατάληξη .Rmd που υποδηλώνει ότι πρόκειται για ένα αρχείο αυτής της μορφής.

Στο πάνω μέρος του αρχείου υπάρχουν τα στοιχεία ετικέτας YAML

```
output:  
  html_document:  
    toc: true  
    toc_depth: 2  
    dev: 'svg'
```

που υποδηλώνουν τον τύπο του αρχείου κατά την εξαγωγή του όταν περάσει από knitr αναλυτική τεκμηρίωση μπορείτε να βρείτε στο <https://yihui.org/knitr/options/> που ειδικεύεται στις επιλογές του knitr. Επιπλέον παρακολουθήστε το βιντεοσκοπημένο μάθημα της ενότητας που περιγράφει αναλυτικά την διαδικασία παραγωγής αρχείων.

3 Quiz

3.1.1 Quiz 1

1. Ποια από τα παρακάτω είναι αντικείμενα δεδομένων στην R;

- A. Vector
- B. Factor
- C. List
- D. ΟΛΑ ΤΑ ΠΑΡΑΠΑΝΩ

2. Ποια από τα παρακάτω είναι valid όνομα για μεταβλητή;

- A. TRUE
- B. _a3
- C. .a1
- D. 1a

3. Ποια η διαφορά μεταξύ Array και Matrix;

- A. Matrix είναι πίνακας ενώ το Array είναι διάνυσμα
- B. Matrix μπορεί να αποθηκεύσει πολλαπλούς τύπους δεδομένων ενώ τα Arrays μόνο έναν κάθε φορά
- C. Matrix είναι πίνακας δύο διαστάσεων ενώ τα Arrays είναι πίνακες με δύο ή περισσότερες διαστάσεις.

4. Ποιο αντικείμενο στην R χρησιμοποιείται για την αποθήκευση και επεξεργασία κατηγορηματικών δεδομένων ;

- A. Factors
- B. Vectors
- C. Matrix

5. Πως μπορείτε να δείτε τον κατάλογο εργασίας στην R;

A. getwd()

B. setwd()

3.1.2 Quiz 2

6. Ποιο από τα παρακάτω είναι λάθος;

A. c(1,2,3)<-x

B. assing("x",c(1,2,3))

C. x<-c(1,2,3)

D. Κανένα από τα παραπάνω

7. Ποιό θα είναι το αποτέλεσμα του παρακάτω κώδικα;
x<-vector("numeric",length=5)

A. 00000

B. NaN

C. 1

D. 10

8. Ποιό θα είναι το αποτέλεσμα του παρακάτω κώδικα;
x <- 1:6
x

A. 0 1 2 3 4 5 6

B. 1 2 3 4 5 6

C. Κανένα από τα παραπάνω

9. Ποιό θα είναι το αποτέλεσμα του παρακάτω κώδικα;
x<-c("a","b","c")
as.numeric(x)

A. ? | Error

B. :-) | Warning

C. ? | Εκτυπώνει το x

10. Ποιό θα είναι το αποτέλεσμα του παρακάτω κώδικα;
x<-2
class(x)

A. ? | Όλα τα παρακάτω

B. ? | "real"

C. :-) | "numeric"

D. ? | "integer"

3.1.3 Quiz 3

Αν θέλεις έναν integer θα πρέπει να βάλεις το παρακάτω πρόθεμα δίπλα στον αριθμό.

A. ? | D

B. ? | R

C. ✓ | L

D. ? | Κανένα από τα παραπάνω

Ποιο είναι το αποτέλεσμα του παρακάτω κώδικα
x<-1:4
y<-1:4
x*y

- A. ☒ 1 4 9 16
- B. ☐ 1 1 1 1
- C. ☐ 1 2 3 4
- D. ☐ Κανένα από τα παραπάνω

Ποιο είναι το αποτέλεσμα του παρακάτω κώδικα :
f <-function() {
##This is an empty function
}
f()

- A. ☐ TRUE
- B. ☒ NULL
- C. ☐ Κανένα από τα παραπάνω

Ποιο από τα παρακάτω χρησιμοποιείται για την ανάγνωση από csv;

- A. ☐ read.data
- B. ☐ read.table
- C. ☐ load.csv
- D. ☒ read.csv

4 Δημιουργία HTML αρχείου

Η πρώτη έκδοση του Markdown εφευρέθηκε κυρίως για συγγραφή πιο εύκολου περιεχομένου σε HTML. Για παράδειγμα, μπορείτε να γράψετε εύκολα ένα κείμενο απευθείας αντί για κώδικα HTML ` text`.

Η σύνταξη του Markdown έχει επεκταθεί σε μεγάλο βαθμό από τον Pandoc. Επιπλέον, το Pandoc καθιστά δυνατή τη μετατροπή ενός εγγράφου Markdown σε μια μεγάλη ποικιλία μορφών εξόδου. Σε αυτό το κεφάλαιο, εστιάζουμε στη δημιουργία του HTML αρχείου των μαθημάτων της ΔΕ.

Για να δημιουργήσετε ένα έγγραφο HTML από το R Markdown, καθορίζετε τη `html_document` μορφή εξόδου στα μέτα-δεδομένα YAML του εγγράφου σας:

```
---
title: Habits
author: John Doe
date: March 22, 2005
output: html_document
---
```

4.1 Πίνακας περιεχομένων

Μπορείτε να προσθέσετε έναν πίνακα περιεχομένων χρησιμοποιώντας την `toc` επιλογή και να καθορίσετε το βάθος των κεφαλίδων που ισχύει για τη χρήση της `toc_depth` επιλογής. Για παράδειγμα:

```
---
title: "Habits"
output:
  html_document:
    toc: true
    toc_depth: 2
---
```

Εάν το βάθος του πίνακα περιεχομένων δεν είναι ρητά καθορισμένο, έχει προεπιλογή 3 (που σημαίνει ότι όλες οι κεφαλίδες επιπέδου 1, 2 και 3 θα περιλαμβάνονται στον πίνακα περιεχομένων).

Μπορείτε να ορίσετε την `toc_float` επιλογή να μετακινηθεί ο πίνακας περιεχομένων στα αριστερά του περιεχομένου του κύριου εγγράφου. Ο κυμαινόμενος πίνακας περιεχομένων θα είναι πάντα ορατός ακόμη και όταν το έγγραφο κάνει κύλιση. Για παράδειγμα:

```
---  
title: "Habits"  
output:  
  html_document:  
    toc: true  
    toc_float: true  
---
```

Μπορείτε προαιρετικά να ορίσετε μια λίστα επιλογών για την `toc_float` παράμετρο που ελέγχει τη συμπεριφορά της. Αυτές οι επιλογές περιλαμβάνουν:

- `Collapsed` (από προεπιλογή σε `TRUE`) ελέγχει εάν τα περιεχόμενα εμφανίζονται μόνο με τις κεφαλίδες ανώτατου επιπέδου (π.χ., H2).
- `smooth_scroll` (από προεπιλογή σε `TRUE`) ελέγχει εάν κινούνται οι κυλίσεις σελίδων όταν προηγούνται τα στοιχεία του πίνακα περιεχομένων (TOC) μέσω των κλικ του ποντικιού.

Για παράδειγμα:

```
---  
title: "Habits"  
output:  
  html_document:  
    toc: true  
    toc_float:  
      collapsed: false  
      smooth_scroll: false  
---
```

4.1.1 Αρίθμηση τμημάτων

Μπορείτε να προσθέσετε αρίθμηση ενότητας στις κεφαλίδες χρησιμοποιώντας την `number_sections` επιλογή:

```
title: "Habits"

output:

  html_document:

    toc: true

    number_sections: true

---
```

Συμβουλή , αν επιλέξετε να χρησιμοποιήσετε την `number_sections` επιλογή, θα πρέπει να χρησιμοποιήσετε `#(H1)` επικεφαλίδες στο έγγραφό σας πριν τις `##(H2)` κεφαλίδες αλλιώς η αρίθμηση θα γίνετε με δεκαδικά ψηφία, χωρίς H1 κεφαλίδες.

4.1.2 Ενότητες σε καρτέλες

Μπορείτε να οργανώσετε περιεχόμενο χρησιμοποιώντας καρτέλες εφαρμόζοντας το `.tabset` χαρακτηριστικό κλάσης σε κεφαλίδες μέσα σε ένα έγγραφο. Αυτό θα έχει ως αποτέλεσμα όλες οι υποκεφαλίδες της κεφαλίδας με το `.tabset` χαρακτηριστικό να εμφανίζονται ως υποκαρτέλες και όχι ως αυτόνομες ενότητες. Για παράδειγμα:

```
## Quarterly Results {.tabset}

### By Product

(tab content)

### By Region

(tab content)
```

Εικόνα 44 Ξεχωριστά tabsets σε HTML

4.1.3 Εμφάνιση και στυλ

Υπάρχουν αρκετές επιλογές που ελέγχουν την εμφάνιση των εγγράφων HTML:

- `Theme` καθορίζει το θέμα που θα χρησιμοποιηθεί για τη σελίδα (τα θέματα αντλούνται από τη βιβλιοθήκη θεμάτων [Bootstrap](#)). Τα έγκυρα θέματα περιλαμβάνουν προεπιλεγμένα, ορισμένα, `journal`, `flatly`, `darkly`, `readable`, `spacelab`, `united`, `cosmo`, `lumen`, `paper`, `sandstone`, `simplex`, και `yeti`. Περάστε `null` χωρίς θέμα (σε αυτήν την περίπτωση μπορείτε να χρησιμοποιήσετε την `css` παράμετρο για να προσθέσετε τα δικά σας στυλ).
- `Highlight` καθορίζει το στυλ επισήμανσης σύνταξης. Υποστηριζόμενες μορφές περιλαμβάνουν `default`, `tango`, `pygments`, `kate`, `monochrome`, `espresso` και το `null` αποτρέπει την επισήμανση σύνταξης.

```
---  
title: "Habits"  
output:  
  html_document:  
    theme: united  
    highlight: tango  
---
```

4.1.4 Το πακέτο `rmdformats`

Το πακέτο `rmdformats` (Barnier 2021) παρέχει διάφορες μορφές εξόδου HTML μορφοποιημένων στυλ, όπως:

`material`: Με αυτήν τη μορφή, κάθε ενότητα πρώτου επιπέδου γίνεται ξεχωριστή σελίδα.

readthedown: Διαθέτει διάταξη πλευρικής γραμμής. Ο πίνακας περιεχομένων εμφανίζεται στην πλαϊνή γραμμή στα αριστερά. Καθώς κάνετε κύλιση στη σελίδα, η τρέχουσα κεφαλίδα της ενότητας θα επισημανθεί αυτόματα στην πλαϊνή γραμμή.

- `html_clean`: Ένα απλό και καθαρό πρότυπο HTML, με έναν δυναμικό πίνακα περιεχομένων στην επάνω δεξιά γωνία της σελίδας.

```
output: rmdformats::material
```

4.1.5 Ευκόνες ΔΕ από R studio

Ετικέτα YAML

```
1. ---
2 title: "Δημιουργία εκπαιδευτικού υλικού για την εκμάθηση της γλώσσας προγραμματισμού R"
3
4 output:
5   rmdformats::material:
6     highlight: kate
7. ---
```

Παράμετροι Knitr

```
10. {r setup, include=FALSE}
11 library(knitr)
12 library(rmdformats)
13
14 ## Global options
15 options(max.print="75")
16 knitr::opts_chunk$set(
17   echo = TRUE,
18   message = FALSE,
19   warning = FALSE,
20   cache = TRUE,
21   comment = NA,
22   prompt = FALSE,
23   tidy = TRUE
24 )
25 opts_knit$set(width=75)
26. }
```

Δημιουργία tabset

```
30. # Μοθήματα
31
32. # Μάθημα 1ο - Εγκατάσταση λογισμικού {.tabset}
33
34. ## Εγκατάσταση της R στα windows
35
```

Λίστα κεφαλαίων εντός το R studio

Δημιουργία εκπαιδευτικού υλικού για την εκμάθηση της γλώσσας προγραμματισμού R
Chunk 1: setup
Μαθήματα
Μάθημα 1ο - Εγκατάσταση Λογισμικού
Εγκατάσταση της R στα Windows
Εγκατάσταση του R Studio
R Studio – Cloud
Το πρώτο μας πρόγραμμα σε R
Chunk 2: first_programm
Chunk 3: pressure
Chunk 4

Σχεδιασμός πίνακα στο R Studio

___Λογικοί___ : Οι λογικοί μπορούν να αναπαράχθουν με τις δεσμευμένες λέξεις 'TRUE' ή 'T' και 'FALSE' ή 'F'.

___Λογικοί Τελεστές___

Οι λογικοί τελεστές έχουν δύο πιθανές τιμές, 'TRUE' ή 'FALSE'.

```
{r}
```

Τελεστής	Απεικόνιση στην R
Ίσον	'=='
Όχι ίσο	'!='
Μικρότερο Από	'<'
Μεγαλύτερο από	'>'
Μικρότερο ή ίσο από	'<='
Μεγαλύτερο ή ίσο	'>='
Όχι	'!'
Η	' '
Και	'&'

Παράδειγμα κώδικα και αποτελέσματος

```
##παράδειγμα ένθετου if ...else ##
{r}
x <- 0
if (x < 0) {
  print("Αρνητικός")
} else if (x > 0) {
  print("Θετικός")
} else
  print("Μηδέν")

[1] "Μηδέν"
```

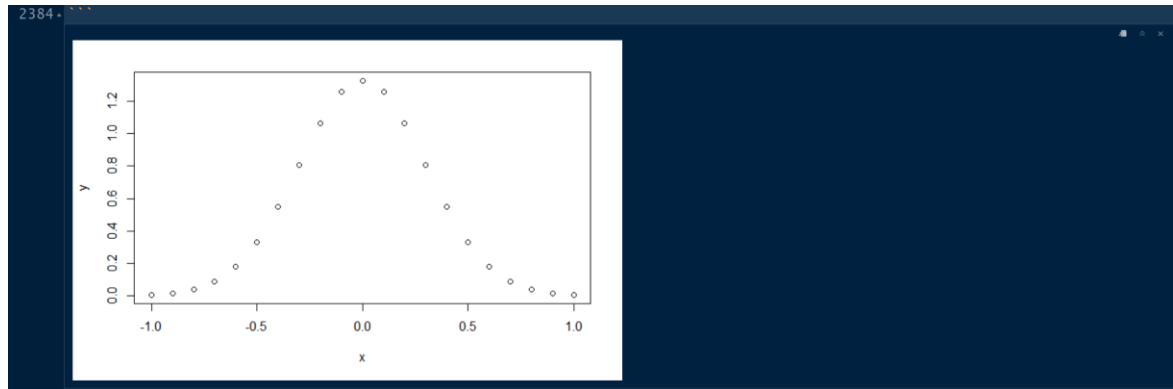
Διαβάζοντας ένα αρχείο csv

```
##Διαβάζοντας ένα αρχείο csv##
Ακολουθεί ένα απλό παράδειγμα της λειτουργίας 'read.csv()' για την ανάγνωση ενός αρχείου csv που είναι διαθέσιμο στον
τρέχοντα κατάλογο εργασίας σας. (χρησιμοποιήστε το αρχείο 'input.csv' που θα βρείτε στον φάκελο 'common')
{r echo=TRUE}
data <- read.csv("C:\\Users\\Korina\\OneDrive\\ΔΙΠΛΩΜΑΤΙΚΗ\\input.csv")
{r}
print(data)
```

Εισαγωγή iframe Βιντεοσκοπημένα μαθήματα

```
## import R - xls  
  
<iframe title="vimeo-player" src="https://player.vimeo.com/video/585270041?h=6738142162" width="640" height="480"  
frameborder="0" allowfullscreen></iframe>
```

Γραφήματα στο R studio



5 Επίλογος

Η R είναι πολύ διαφορετική από τις άλλες γλώσσες προγραμματισμού. Θα μπορούσε να πει κάποιος ότι η R δεν μαθαίνεται, απλά χρησιμοποιείται. Αυτό που πρέπει να μάθει κάποιος είναι η θεωρία της εξόρυξης δεδομένων, μηχανικής μάθησης, μαθηματικών και στατιστικής, των τομέων δηλαδή της επιστήμης των δεδομένων στα οποία δραστηριοποιείται η R. Εφόσον γνωρίζει κάποιος την θεωρία μπορεί εύκολα εφαρμόσει τα πάντα στην R. Με περισσότερα από 18000 διαθέσιμα πακέτα ανοικτού κώδικα αναρτημένα στο επίσημο site της <https://ftp.cc.uoc.gr/mirrors/CRAN/> αποτελεί ένα από τα δυνατότερα εργαλεία της επιστήμης των δεδομένων.

Η R έχει δεχτεί δριμύ ανταγωνισμό τα τελευταία χρόνια από διάφορα στατιστικά πακέτα και γλώσσες προγραμματισμού όπως SPSS, SAS, STATA με κυριότερη την PYTHON. Η Python είναι μια ευέλικτη γλώσσα προγραμματισμού και κατά γενική ομολογία πιο εύκολη στο χειρισμό της από την R. Τα τελευταία χρόνια κερδίζει θέσεις στην κατάταξη IEEE Spectrum. Η Python μπορεί να κάνει σχεδόν τις ίδιες εργασίες με την R στην ανάλυση δεδομένων, στην μηχανική μάθηση, στην άντληση στοιχείων από το internet (web scrapping) κλπ. Η συγγραφή κώδικα σε Python είναι εύκολοσυντήρητη και το μεγαλύτερο μέρος της ανάλυσης δεδομένων μπορεί να γίνει με 5 βιβλιοθήκες (Numpy, Pandas, Scipy, Scikit-learn και Seaborn).

Οι βασικές διαφορές μεταξύ R και Python είναι οι παρακάτω:

- Η R χρησιμοποιείται κυρίως για στατιστική ανάλυση, ενώ η Python παρέχει πιο ευρύτερη προσέγγιση στην επιστήμη των δεδομένων.
- Ο πρωταρχικός στόχος της R είναι η ανάλυση δεδομένων και οι στατιστική, ενώ ο πρωταρχικός στόχος της Python είναι η ανάπτυξη και η παραγωγή λογισμικού.
- Οι χρήστες R αποτελούνται κυρίως από μελετητές και επαγγελματίες R&D, ενώ οι χρήστες Python είναι κυρίως προγραμματιστές.
- Η R παρέχει ευελιξία στη χρήση διαθέσιμων βιβλιοθηκών ενώ η Python παρέχει ευελιξία στην κατασκευή νέου λογισμικού.
- Την R είναι δύσκολο να την μάθεις στην αρχή ενώ η Python είναι εύκολη στη μάθηση της.
- Η R είναι αυτόνομη, ενώ η Python είναι ενσωματωμένη με εφαρμογές.
- Τόσο η R όσο και η Python μπορούν να χειριστούν τεράστιο μέγεθος βάσης δεδομένων. (Daniel Johnson,2021)

Συμπέρασμα

Τελικά, η επιλογή μεταξύ R ή Python εξαρτάται από το τι θέλετε να κάνετε, *Στατιστική ανάλυση ή Ανάπτυξη Λογισμικού* και το χρονικό διάστημα που μπορείτε να επενδύσετε. Σε κάθε περίπτωση η R είναι μια γλώσσα προγραμματισμού δοκιμασμένη με μεγάλη κοινότητα χρηστών και τεράστια συμβολή στην επιστήμη των δεδομένων, που σίγουρα αξίζει να την μάθεις.

6 Βιβλιογραφικές Αναφορές

Ελληνόγλωσση βιβλιογραφία

Φωκιανός & Χαραλάμπους, (2010). Εισαγωγή στην R Πρόχειρες Σημειώσεις Τμήμα Μαθηματικών και Στατιστικής Πανεπιστήμιου Κύπρου 2η έκδοση

Βερύκιος Σ Βασίλειος, Καγκλής Βασίλειος, Σταυρόπουλος Κ. Ηλίας, (2015). Η επιστήμη των δεδομένων μέσα από τη γλώσσα R

Ξενόγλωσση Βιβλιογραφία

Roger D. Peng, (2015). R Programming for Data Science

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988), The New S Language.

Maechler Martin Seminar fur statistic <https://stat.ethz.ch/R-manual/>

Yihui Xie, JJ Allaire, Garrett Grolemond, (2021) R Markdown: The Definitive Guide (<https://bookdown.org/yihui/rmarkdown/>)

Barnier, Julien. 2021.Rmdformats: HTML Output Formats and Templates for Rmarkdown Documents.<https://github.com/juba/rmdformats>.

Διαδίκτυο

OBS Studio <https://obsproject.com/download>

[https://en.wikipedia.org/wiki/R_\(programming_language\)](https://en.wikipedia.org/wiki/R_(programming_language)) R programming language, (2021)

<http://rstudio.cloud>

<https://www.tutorialspoint.com/r/index.htm>

<http://www.sthda.com/english/wiki/r-xlsx-package-a-quick-start-guide-to-manipulate-excel-files-in-r>,2019

<https://r4ds.had.co.nz/data-visualisation.html>,2018

<https://r4ds.had.co.nz/transform.html>

https://www.tutorialspoint.com/r/r_mean_median_mode.htm

http://rstudio-pubs-static.s3.amazonaws.com/374857_5a23bad9783a43c1b102aa80aa5c1a7c.html,2020

<https://www.investopedia.com/terms/m/median.asp> , 2019)

Daniel Johnson, 2021, Άρθρο R Vs Python: What's the Difference?

ΠΑΡΑΡΤΗΜΑ

Στο παρόν παράρτημα θα γίνει παρουσίαση του διδακτικού υλικού των ηλεκτρονικών μαθημάτων που έχει υλοποιηθεί σε HTML με R Markdown το οποίο είναι το βασικό εργαλείο συγγραφής του διαδραστικού εκπαιδευτικού υλικού που σχεδιάσαμε και δημιουργήσαμε στα πλαίσια της εργασίας αυτής. Θα ακολουθήσουν βήμα προς βήμα οι οθόνες των μαθημάτων.

Μάθημα 1^ο Εγκατάσταση Λογισμικού

Το μάθημα αποτελείται από 4 ενότητες διαχωρισμένες σε tab επιπλέον διατίθενται

- 1) Κώδικας του μαθήματος σε.Rmd
- 2) Παρουσίαση του μαθήματος σε Power Point
- 3) 3 βιντεοσκοπημένα μαθήματα

Δημιουργία εκπαιδευτικού υλικού για την εκμάθηση της γλώσσας προγραμματισμού R

Μαθήματα

- Μάθημα 1ο - Εγκατάσταση Λογισμικού
- Μάθημα 2ο - Βασικοί κανόνες
- Μάθημα 3ο - Δομές επιλογής και επανάληψης
- Μάθημα 4ο - Δομές Δεδομένων
- Μάθημα 5ο - Packages & Functions

Μάθημα 1ο - Εγκατάσταση Λογισμικού

Εγκατάσταση της R στα Windows Εγκατάσταση του R Studio R Studio – Cloud

Το πρώτο μας πρόγραμμα σε R

Οδηγίες Εγκατάστασης της R

Δημιουργία εκπαιδευτικού υλικού για την εκμάθηση της γλώσσας προγραμματισμού R

Μαθήματα

Μάθημα 1ο

Μάθημα 2ο

Μάθημα 3ο

Μάθημα 4ο

Μάθημα 5ο

1. Μεταβείτε στον επίσημο ιστότοπο του προγραμματισμού R, <http://www.R-project.org>
2. Κάντε κλικ στον σύνδεσμο CRAN
3. Επιλέξτε mirror

Μάθημα 2^ο – Βασικοί κανόνες

Το μάθημα αποτελείτε από 4 ενότητες διαχωρισμένες σε tab επιπλέον διατίθενται

- 1) Κώδικας του μαθήματος σε 3 ξεχωριστά files σε R
- 2) Παρουσίαση του μαθήματος σε Power Point
- 3) 1 βιντεοσκοπημένο μάθημα
- 4) Quiz ερωτήσεων σε ξεχωριστό file

Δημιουργία εκπαιδευτικού υλικού για την εκμάθηση της γλώσσας προγραμματισμού R

Μαθήματα

Μάθημα 1ο - Εγκατάσταση Λογισμικού


Μάθημα 2ο - Βασικοί κανόνες

Μάθημα 3ο - Δομές επιλογής και επανάληψης

Μάθημα 4ο - Δομές Δεδομένων

Μάθημα 5ο - Packages & Functions

Μάθημα 2ο - Βασικοί κανόνες



Μαθηματικές Πράξεις

Μεταβλητές και κανόνες

Βασικές κατηγορίες αντικειμένων «Atomic» Classes

Ενσωματωμένες μεταβλητές (built in variables)

Ο πίνακας περιγράφει για την κάθε μαθηματική πράξη τον τρόπο απεικόνισης στην R

Μαθηματική Πράξη	Απεικόνιση σε R
Πρόσθεση	8+3
Αφαίρεση	8-3
Πολλαπλασιασμός	8*3
Διαίρεση	8/3

Μάθημα 3^ο – Δομές Επιλογής και επανάληψης

Το μάθημα αποτελείτε από 2 ενότητες διαχωρισμένες σε tab επιπλέον διατίθενται

- 1) Κώδικας του μαθήματος σε 2 ξεχωριστά files σε R
- 2) Παρουσίαση του μαθήματος σε Power Point
- 3) 2 βιντεοσκοπημένα μαθήματα

Δημιουργία εκπαιδευτικού υλικού για την εκμάθηση της γλώσσας προγραμματισμού R

- Μαθήματα
- Μάθημα 1ο - Εγκατάσταση Λογισμικού
- Μάθημα 2ο - Βασικοί κανόνες
- Μάθημα 3ο - Δομές επιλογής και επανάληψης
- Μάθημα 4ο - Δομές Δεδομένων
- Μάθημα 5ο - Packages & Functions

Μάθημα 3ο - Δομές επιλογής και επανάληψης

Δομές επιλογής

Δομές επανάληψης

Η λήψη αποφάσεων είναι ένα σημαντικό μέρος του προγραμματισμού. Αυτό μπορεί να επιτευχθεί στον προγραμματισμό σε R χρησιμοποιώντας τη `if...else`.

IF

Η σύνταξη του `if` είναι:

```
if (δηλώση αληθείς) {
  κώδικας
}
```

Μάθημα 4ο – Δομές Δεδομένων

Το μάθημα αποτελείται από 6 ενότητες διαχωρισμένες σε tab επιπλέον διατίθενται

- 1) Κώδικας του μαθήματος σε 5 ξεχωριστά files σε R
- 2) 6 βιντεοσκοπημένα μαθήματα

Δημιουργία εκπαιδευτικού υλικού για την εκμάθηση της γλώσσας προγραμματισμού R

- Μαθήματα
- Μάθημα 1ο - Εγκατάσταση Λογισμικού
- Μάθημα 2ο - Βασικοί κανόνες
- Μάθημα 3ο - Δομές επιλογής και επανάληψης
- Μάθημα 4ο - Δομές Δεδομένων
- Μάθημα 5ο - Packages & Functions

Μάθημα 4ο - Δομές Δεδομένων

Δομές Δεδομένων της R

Διάγραμμα Vector

Λίστα List

Πίνακας Matrix

Παράγοντες Factors

Πολυδιάστατοι πίνακες Arrays

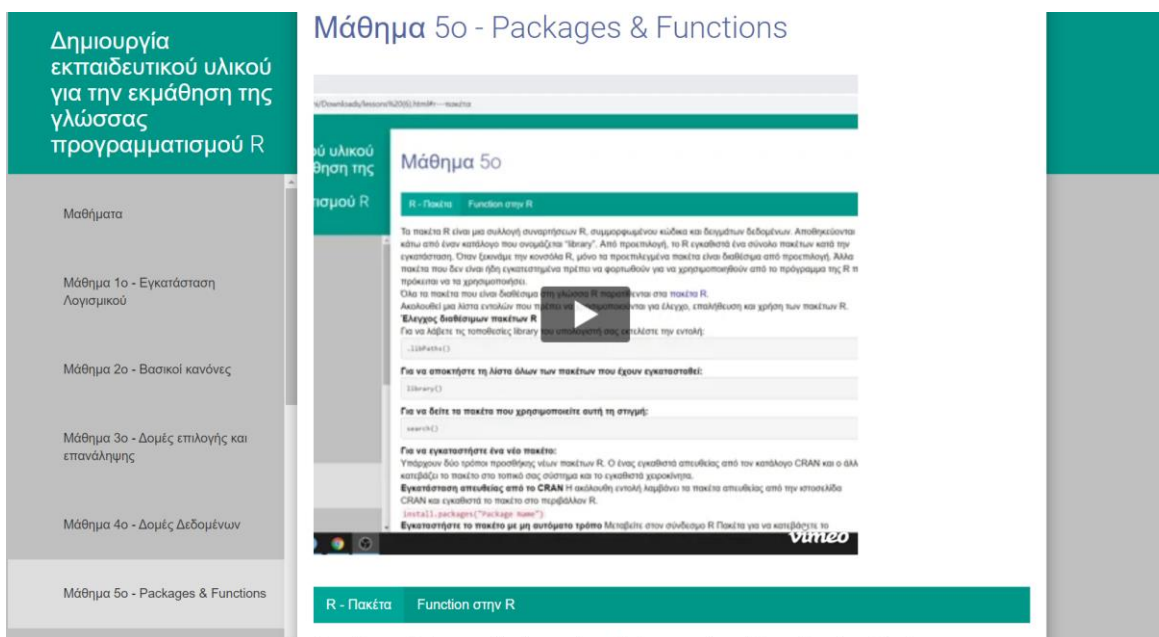
Πλαίσια δεδομένων Data frames

Το διάγραμμα είναι μια βασική δομή δεδομένων στην R. Περιέχει στοιχεία του ίδιου τύπου. Οι τύποι δεδομένων μπορεί να είναι λογικοί, ακέραιοι, `double`, `character`, `factor`, `list`, `matrix`, `array`. Μπορείτε να ελέγξετε τον τύπο των δεδομένων με τη

Μάθημα 5ο – Packages & Function

Το μάθημα αποτελείται από 2 ενότητες διαχωρισμένες σε tab επιπλέον διατίθενται

- 1) Κώδικας του μαθήματος σε 1 ξεχωριστό files σε R
- 2) 2 βιντεοσκοπημένα μαθήματα



Μάθημα 6ο – Import & Export

Το μάθημα αποτελείται από 4 ενότητες διαχωρισμένες σε tab επιπλέον διατίθενται

- 1) Κώδικας του μαθήματος σε 4 ξεχωριστά files σε R
- 2) Αρχείο δεδομένων τύπου .csv
- 3) Αρχείο δεδομένων τύπου .xlsx
- 4) Αρχείο δεδομένων τύπου .json
- 5) 3 βιντεοσκοπημένα μαθήματα

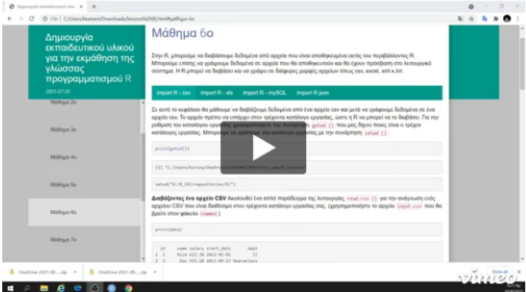
Δημιουργία εκπαιδευτικού υλικού για την εκμάθηση της γλώσσας προγραμματισμού R

- Μάθημα 5ο - Packages & Functions
- Μάθημα 6ο - Import & Export**
- Μάθημα 7ο - Διάγράμματα & Γραφήματα
- Μάθημα 8ο - Οπτικοποίηση δεδομένων
- Μάθημα 9ο - Στατιστικά στοιχεία στην R
- Μάθημα 10ο - R Markdown

Μάθημα 6ο - Import & Export

Στην R, μπορούμε να διαβάσουμε δεδομένα από αρχεία που είναι αποθηκευμένα εκτός του περιβάλλοντος R. Μπορούμε επίσης να γράψουμε δεδομένα σε αρχεία που θα αποθηκευτούν και θα έχουν πρόσβαση στο λειτουργικό σύστημα. Η R μπορεί να διαβάσει και να γράψει σε διάφορες μορφές αρχείων όπως csv, excel, xml κ.λπ.

import R - csv
import R - xls
import R - mySQL
import R json



Σε αυτό το κεφάλαιο θα μάθουμε να διαβάζουμε δεδομένα από ένα αρχείο csv και μετά να γράφουμε δεδομένα σε ένα αρχείο csv. Το αρχείο πρέπει να υπάρχει στον τρέχοντα κατάλογο εργασίας, ώστε η R να μπορεί να το διαβάσει. Για την

Μάθημα 7ο – Διαγράμματα & Γραφήματα

Το μάθημα αποτελείται από 4 ενότητες διαχωρισμένες σε tab επιπλέον διατίθενται

- 1) Κώδικας του μαθήματος σε 1 ξεχωριστό files σε R
- 2) 1 βιντεοσκοπημένο μάθημα

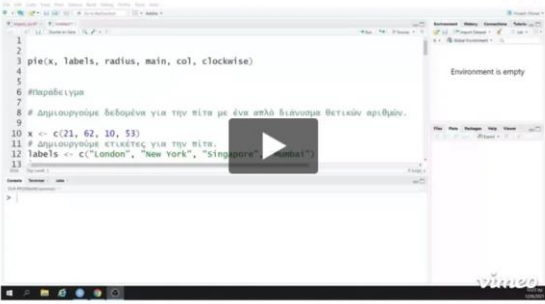
Δημιουργία εκπαιδευτικού υλικού για την εκμάθηση της γλώσσας προγραμματισμού R

- Μάθημα 5ο - Packages & Functions
- Μάθημα 6ο - Import & Export
- Μάθημα 7ο - Διάγράμματα & Γραφήματα**
- Μάθημα 8ο - Οπτικοποίηση δεδομένων
- Μάθημα 9ο - Στατιστικά στοιχεία στην R
- Μάθημα 10ο - R Markdown

Μάθημα 7ο - Διάγράμματα & Γραφήματα

Διάγράμματα και Γραφήματα

Εισαγωγή στα γραφήματα



Διαγράμματα πίτας
Γραφήματα ράβδων
Boxplots
Ιστογράμμο - Histograms

Η γλώσσα προγραμματισμού R έχει πολλές βιβλιοθήκες για τη δημιουργία διαγραμμάτων και γραφημάτων. Ένα γράφημα πίτας είναι μια αναπαράσταση των τιμών ως φέτες ενός κύκλου με διαφορετικά χρώματα. Στην R το γράφημα πίτας δημιουργείται χρησιμοποιώντας τη συνάρτηση `pie()` σαν διάγραμμα θετικών αριθμών. Οι

Μάθημα 8^ο – Οπτικοποίηση Δεδομένων

Το μάθημα αποτελείται από 1 ενότητες επιπλέον διατίθενται :

- 1) Κώδικας του μαθήματος σε 1 ξεχωριστό files σε R

The screenshot shows a presentation slide with a teal header and a grey sidebar. The header contains the text 'Δημιουργία εκπαιδευτικού υλικού για την εκμάθηση της γλώσσας προγραμματισμού R'. The sidebar lists several topics: 'Μάθημα 5ο - Packages & Functions', 'Μάθημα 6ο - Import & Export', 'Μάθημα 7ο - Διάγράμματα & Γραφήματα', 'Μάθημα 8ο - Οπτικοποίηση δεδομένων' (which is highlighted), 'Μάθημα 9ο - Στατιστικά στοιχεία στην R', and 'Μάθημα 10ο - R Markdown'. The main content area is titled 'Μάθημα 8ο - Οπτικοποίηση δεδομένων' and has a subtitle 'Οπτικοποίηση δεδομένων'. It includes a URL '(https://r4ds.had.co.nz/data-visualisation.html)', an introduction to ggplot2, and R code for installing and loading the tidyverse package. The code is:

```
install.packages("tidyverse", repos = "http://cran.us.r-project.org")  
  
package 'tidyverse' successfully unpacked and MD5 sums checked  
  
The downloaded binary packages are in  
C:\Users\Korina\AppData\Local\Temp\RtmpwL77ND\downloaded_packages  
  
library(tidyverse)
```

 Below the code, there is a paragraph explaining the use of the mpg dataset and a list of tasks: '1. Τα αυτοκίνητα με μεγάλους κινητήρες χρησιμοποιούν περισσότερα καύσιμα από τα αυτοκίνητα με μικρούς κινητήρες;'. The slide concludes with the sentence: 'Τώρα θα προσπαθήσουμε να απαντήσουμε χρησιμοποιώντας το dataset mpg, που περιλαμβάνει παρατηρήσεις που'.

Μάθημα 9^ο - Στατιστικά στοιχεία στην R

Το μάθημα αποτελείται από 4 ενότητες επιπλέον διατίθενται :

- 1) Κώδικας του μαθήματος σε 1 ξεχωριστό files σε R

Δημιουργία εκπαιδευτικού υλικού για την εκμάθηση της γλώσσας προγραμματισμού R

- Μάθημα 5ο - Packages & Functions
- Μάθημα 6ο - Import & Export
- Μάθημα 7ο - Διάγράμματα & Γραφήματα
- Μάθημα 8ο - Οπτικοποίηση δεδομένων
- Μάθημα 9ο - Στατιστικά στοιχεία στην R**
- Μάθημα 10ο - R Markdown

Μάθημα 9ο - Στατιστικά στοιχεία στην R

Παραδείγματα στατιστικών στοιχείων R
Σκοπός αυτού του κεφαλαίου είναι να δούμε την απικόνιση βασικών στατιστικών υπολογισμών στην R

R - Μέσος όρος, διάμεσος
R - Γραμμική παλινδρόμηση - R Linear Regression
R - Κανονική κατανομή
R - Δέντρο απόφασης

Η στατιστική ανάλυση στην R πραγματοποιείται χρησιμοποιώντας πολλές ενσωματωμένες συναρτήσεις. Οι περισσότερες από αυτές τις λειτουργίες αποτελούν μέρος του βασικού πακέτου R.

Μέσος Όρος
Υπολογίζεται λαμβάνοντας το άθροισμα των τιμών και διαιρώντας με τον αριθμό τιμών σε μια σειρά δεδομένων. Ο υπολογισμός στην R γίνεται με την function `mean()` και έχει την παρακάτω σύνταξη :

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

Όπου :

Το x είναι το διάνυσμα εισαγωγής.
Το trim χρησιμοποιείται για να ρίξει μερικές παρατηρήσεις και από τα δύο άκρα του ταξινομημένου διανύσματος.
Το na.rm χρησιμοποιείται για την αφαίρεση των τιμών που λείπουν από το διάνυσμα εισόδου.

Παράδειγμα

```
# Δημιουργία διανύσματος
x <- c(12,7,3,4.2,18,2,54,-21,8,-5)

# Εύρεση Μέσου όρου
MO <- mean(x)
print(MO)
```

Μάθημα 10ο – R Markdown

Το μάθημα αποτελείται από 1 ενότητα επιπλέον διατίθενται :

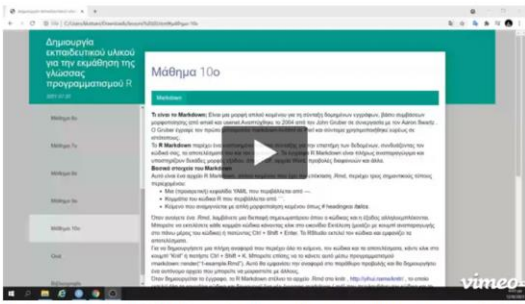
- 1) Κώδικας του μαθήματος σε 1 ξεχωριστό files σε R
- 2) CheatSheet του R Markdown από το site

Δημιουργία εκπαιδευτικού υλικού για την εκμάθηση της γλώσσας προγραμματισμού R

- Μάθημα 5ο - Packages & Functions
- Μάθημα 6ο - Import & Export
- Μάθημα 7ο - Διάγράμματα & Γραφήματα
- Μάθημα 8ο - Οπτικοποίηση δεδομένων
- Μάθημα 9ο - Στατιστικά στοιχεία στην R
- Μάθημα 10ο - R Markdown**

Μάθημα 10ο - R Markdown

Markdown



Τι είναι το Markdown; Το Markdown είναι μια μορφή απλού κειμένου για τη σύνταξη δομημένων εγγράφων, βάσει συμβάσεων μορφοποίησης από email και usenet. Αναπτύχθηκε το 2004 από τον John Gruber σε συνεργασία με τον Aaron Swartz. Ο Gruber έγραψε τον πρώτο μετατροπέα markdown-to-html σε Perl και σύντομα χρησιμοποιήθηκε ευρέως σε ιστότοπους.

Το R Markdown παρέχει ένα ενοποιημένο πλαίσιο σύνταξης για την επιστήμη των δεδομένων, συνδυάζοντας στον κώδικά σας, τα αποτελέσματά του και τον σχολιασμό. Τα έγγραφα R Markdown είναι αναπαραγώγιμα και υποστηρίζουν δεκάδες μορφές εξόδου, όπως PDF, αρχεία Word, παραβλέψεις διαφανειών και άλλα.

Quiz

Quiz 10 ερωτήσεων πολλαπλής επιλογής με βαθμολόγηση

Δημιουργία εκπαιδευτικού υλικού για την εκμάθηση της γλώσσας προγραμματισμού R

Μάθημα 7ο - Διάγράμματα & Γραφήματα

Μάθημα 8ο - Οπτικοποίηση δεδομένων

Μάθημα 9ο - Στατιστικά στοιχεία στην R

Μάθημα 10ο - R Markdown

Quiz

Βιβλιογραφία

Quiz

Quiz 1

<

Show all questions

1 / 10

=>

Ποια από τα παρακάτω είναι αντικείμενα δεδομένων στην R;

A. ?

Vector

B. ?

Factor