29/11/2022

Αναλυτικές Μέθοδοι στη Γεωπληροφορική

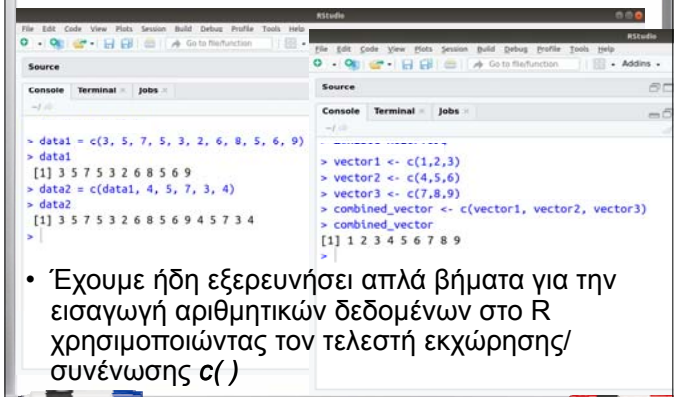
Εισαγωγή κοινών αρχείων δεδομένων στο R

Μορφοποίηση δεδομένων στο R

- Τυπικά, η αρχική μορφοποίηση των δεδομένων ενδιαφέροντος για μια ανάλυση θα πρέπει να έχει ολοκληρωθεί **πριν από την εισαγωγή τους στο R : *Γενικά επιδιώκεται η διάθεση καλά τακτοποιημένων (οργανωμένων) δεδομένων → Ευκολότερη εισαγωγή στο R***
- Εισαγωγή δεδομένων από εξωτερικά αρχεία και οργάνωση/αποθήκευσή τους στον Η/Υ σας
- Βέλτιστές/ορθές πρακτικές για τη μορφοποίηση δεδομένων προς επεξεργασία στο R
- Εισαγωγή δεδομένων σε συνήθεις μορφότυπους

- Η εισαγωγή/εξαγωγή δεδομένων στο R, γενικά (αλλά όχι πάντα), είναι αρκετά απλή
- Τυπικά μπορεί να γίνει είτε με τη χρήση ενσωματωμένων συναρτήσεων ανάγνωσης αρχείων, είτε με τη χρήση εξωτερικά διαθέσιμων εξειδικευμένων πακέτων του R

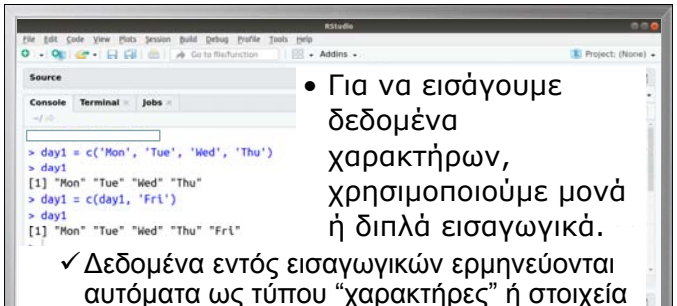
Σύνδεση με τα προηγούμενα …



```
> data1 = c(3, 5, 7, 5, 3, 2, 6, 8, 5, 6, 9)
> data1
 [1] 3 5 7 5 3 2 6 8 5 6 9
> data2 = c(data1, 4, 5, 7, 3, 4)
> data2
 [1] 3 5 7 5 3 2 6 8 5 6 9 4 5 7 3 4
>
```

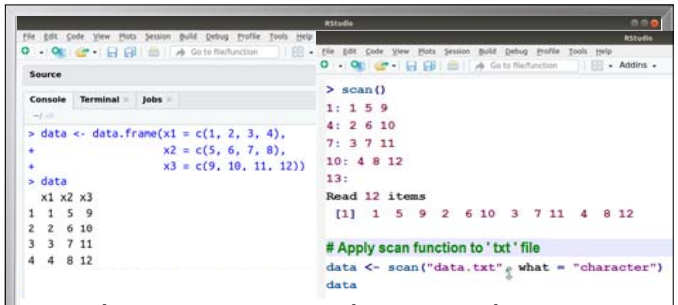
```
> vector1 <- c(1,2,3)
> vector2 <- c(4,5,6)
> vector3 <- c(7,8,9)
> combined_vector <- c(vector1, vector2, vector3)
> combined_vector
[1] 1 2 3 4 5 6 7 8 9
>
```

- Έχουμε ήδη εξερευνήσει απλά βήματα για την εισαγωγή αριθμητικών δεδομένων στο R χρησιμοποιώντας τον τελεστή εκχώρησης/ συνένωσης *c( )*



```
> day1 = c('Mon', 'Tue', 'Wed', 'Thu')
> day1
[1] "Mon" "Tue" "Wed" "Thu"
> day1 = c(day1, 'Fri')
> day1
[1] "Mon" "Tue" "Wed" "Thu" "Fri"
```

- Για να εισάγουμε δεδομένα χαρακτήρων, χρησιμοποιούμε μονά ή διπλά εισαγωγικά.
  - ✓Δεδομένα εντός εισαγωγικών ερμηνεύονται αυτόματα ως τύπου "χαρακτήρες" ή στοιχεία κειμένου.
- Εάν συνδυάζονται αριθμητικές τιμές και στοιχεία κειμένου, το R εφαρμόζει *κανόνες 'εξαναγκασμού'* και μετατρέπει αριθμούς σε χαρακτήρες.



```
> data <- data.frame(x1 = c(1, 2, 3, 4),
+                    x2 = c(5, 6, 7, 8),
+                    x3 = c(9, 10, 11, 12))
> data
  x1 x2 x3
1  1  5  9
2  2  6 10
3  3  7 11
4  4  8 12
```

```
> scan()
1: 1 5 9
4: 2 6 10
7: 3 7 11
10: 4 8 12
13:
Read 12 items
 [1]  1  5  9  2  6 10  3  7 11  4  8 12
```

```
# Apply scan function to ' txt ' file
data <- scan("data.txt", what = "character")
data
```

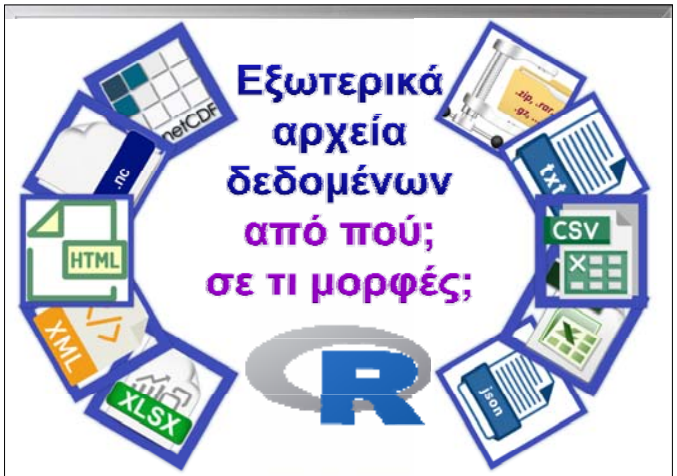
- ✓Αντί να πληκτρολογούμε δεδομένα εισόδου στην κονσόλα του R, μπορούμε να χρησιμοποιήσουμε την εντολή *scan()*,
  - ✓π.χ., με *copy / paste*
  - ✓ή για τη λήψη δεδομένων από αρχεία


```
CONSIDER THE TXT FILE data.txt, WITH CONTENTS AS FOLLOWS:
"x1" "x2" "x3"
4 1 5
4 8 3
1 4 5
9 0 6
```

Μια κοινή χρήση της συνάρτησης *scan()* είναι για τη γρήγορη ανάγνωση πινακοποιημένων δεδομένων

```
data1 <- scan("data.txt", what = "character")  # Apply scan function to txt file
data1                                          # Print scan output to console
 [1] "x1" "x2" "x3" "4"  "1"  "5"  "4"  "8"  "3"  "1"  "4"  "5"  "9"  "0"  "6"

data3 <- scan("data.txt", skip = 1)   # Skip first line of txt file
data3                                 # Print scan output to console
 [1] 4 1 5 4 8 3 1 4 5 9 0 6

x <- matrix(data3,nrow=4,ncol=3,byrow=TRUE)
x
     [,1] [,2] [,3]
[1,]    4    1    5
[2,]    4    8    3
[3,]    1    4    5
[4,]    9    0    6
```

Τυπικά, τα δεδομένα ενός πίνακα 200x2000, μέσω της *scan()* διαβάζονται σε 1-3 s ή αντίστοιχα σε 7-10 s, μέσω της *read.table()*



Τα δεδομένα για χρήση στο R μπορούν …

- Να βρίσκονται τοπικά σε έναν Η/Υ, στο διαδίκτυο *('http:///', 'ftp:///' , 'file:///', …)*, ή να μπορούν να ληφθούν από άλλες πηγές (π.χ. μια βάση δεδομένων)
- Να είναι σε διάφορους κοινούς μορφότυπους κειμένου *(ASCII, txt, csv,…)*, εικόνων (*bmp, jpeg, png, tiff, …*), binary *(NetCDF, HDF5, dBase, …)* ή τύπους αρχείων *(xls, xlsx, shp, …)*
- Να προέρχονται από άλλα γνωστά λογισμικά
  - Άλλα στατιστικά λογισμικά (*SAS, SPSS, Minitab, Strata, Systat*, …)
- Η κατάλληλη διαδικασία εισαγωγής των δεδομένων είναι το κλειδί για να ξεκινήσει η ανάλυση τους

- Για τη σωστή εισαγωγή των δεδομένων στο R είναι χρήσιμο να ακολουθούνται μερικές 'καλές' πρακτικές προετοιμασίας τους
  - Ονοματολογίες και κωδικοποιήσεις των αρχείων δεδομένων συμβατές με τους κανόνες της R
  - Αντικατάσταση κενών τιμών στα δεδομένα
  - Διαχείριση ημερομηνιών ως ετικέτες
  - Μετατροπές δεδομένων σε "βολικούς" μορφότυπους ως όχημα ανταλλαγής δεδομένων
  - …

Για αναλύσεις μικρού όγκου δεδομένων, η ευκολότερη μορφή τους για εισαγωγή στο R είναι απλά αρχεία excel ή πινακοποιημένα δεδομένα κειμένου

- Συνήθως η πρώτη σειρά χρησιμοποιείται ως **ονόματα στηλών**. Γενικά, οι στήλες αντιπροσωπεύουν **μεταβλητές**.
- Η πρώτη στήλη, συχνά χρησιμοποιείται ως **ονόματα γραμμών** . Γενικά οι σειρές αντιπροσωπεύουν **παρατηρήσεις**.
- Κάθε όνομα σειράς πρέπει να είναι μοναδικό → όχι διπλότυπα ονόματα.

Ονοματολογία των δεδομένων

πρώτη σειρά ως **κεφαλίδες** στηλών

πρώτη στήλη ως **ονόματα γραμμών**

	name	100m	Long jump	Shot_put	High jump	400m
2	SEBRLE	11.04	7.58	14.8	2.07	49.81
3	CLAY	10.76	7.4	14.26	1.86	49.37
4	KARPOV			14.77		
5	BERNARD			14.25		
6	YURKOV	11.34	7.09			
7	WARNERS	11.11	7.6		1.98	48.68
8	ZSIVOCZKY	11.13	7.3	13.48	2.01	48.62
9	McMULLEN	10.83	7.31	13.76	2.13	49.91
10	MARTINEAU		6.81	14.57	1.95	50.14
11	HERNU	11.37	7.56	14.41		51.1
12	BARRAS		6.97	14.09		49.48
13	NOOL	11.33			1.98	49.2
14	BOURGUIGNO	11.36			1.86	51.16

Bad names

Good name

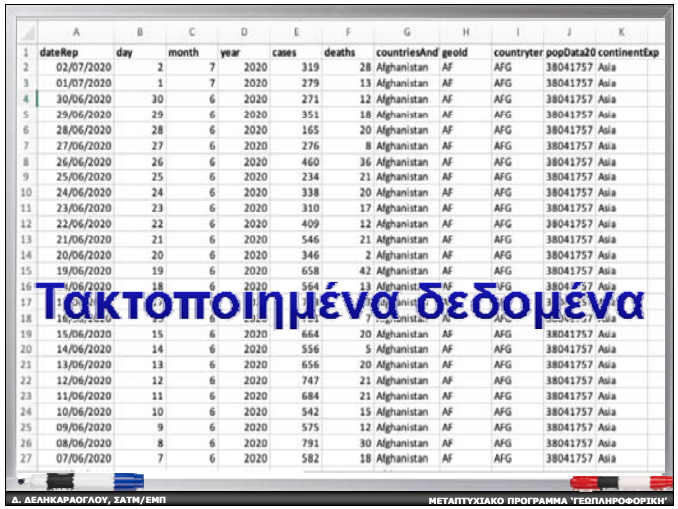
Missing data

decathlon.txt

Mode Normal Prêt

	A	B	C	D	E	F	G	H	I	J
1	**Lead (Pb) Concentrations in 1999 Spokane River Fish**									
2	Source: WA State Dept. of Ecology report									
3	concentrations in parts per million (ppm)									
4										
5	**filets**	trout	whitefish	sucker		**whole**	trout	whitefish	sucker	
6		0.480	0.020	0.088			0.730	0.650	4.340	
7		0.071	0.020	0.210			1.140	0.560	1.980	
8		0.110	0.020	0.280			0.600		3.120	
9		0.320	0.020	0.030			1.590		1.800	
10		0.120	0.020	0.036						
11		0.220	0.065	0.047						
12		0.055	0.020	0.077						
13		0.320	0.037	0.069						
14		0.077	0.020	0.160						
15		0.081	0.036	0.088						
16		0.170		0.120						
17		0.130		0.054						
18		0.110		0.080						
19		0.081		0.059						
20		0.098		0.094						
21		0.180		0.059						
22		0.230		0.068						
23		0.082		0.020						
24		0.210		0.090						
25		0.200		0.046						
26		0.025								
27		0.038								
28										

Τυπικοί κανόνες για 'τακτοποιημένα' δεδομένα στο R περιγράφονται λεπτομερέστερα στην ιστοσελίδα: https://cran.r-project.org/web/packages/tidyr/vignettes/tidy-data.html



Τακτοποιημένα δεδομένα

Τυπική ροή εργασιών εισαγωγής/χρήσης δεδομένων στο R;

1) Καθαρισμός δεδομένων
- α) αντικατάσταση των κενών κελιών με NA (ή άλλους χαρακτήρες).
- β) διόρθωση των ονομάτων των στηλών (ότι δεν υπάρχουν κενά, δεν υπάρχουν ειδικοί χαρακτήρες. '.' και '_' είναι ΟΚ).
- γ) αποθήκευση ως αρχείο *.csv* (ή *.txt*).

2) Προσπαθήστε να εισαγάγετε τα δεδομένα με *read.csv (...)*.
- α) Εάν προκύψουν σφάλματα, ανοίξτε το αρχείο *.csv* σε ένα πρόγραμμα επεξεργασίας κειμένου και, βρείτε και διορθώστε τυχόν προβλήματα και επαναλάβετε τα βήματα 1γ και 2).

Τυπική ροή εργασιών εισαγωγής/χρήσης δεδομένων στο R;

2) Για το βήμα #2 είναι ιδιαίτερα χρήσιμο ο επεξεργαστής κειμένου να έχει τη δυνατότητα να βλέπει "αόρατους" χαρακτήρες

3) Ελέγξτε τα δεδομένα, χρησιμοποιώντας τις διαγνωστικές/διαχειριστικές εντολές του R – όπως: *names()*, *dim()* , *summary()* είναι τα συνήθη εργαλεία, όπως και η εντολή *str()* λειτουργεί επίσης.

Είναι επίσης σημαντικό, το R να γνωρίζει επακριβώς που θα αναζητάει τα αρχεία των δεδομένων (π.χ., *setwd("file/path/here")*

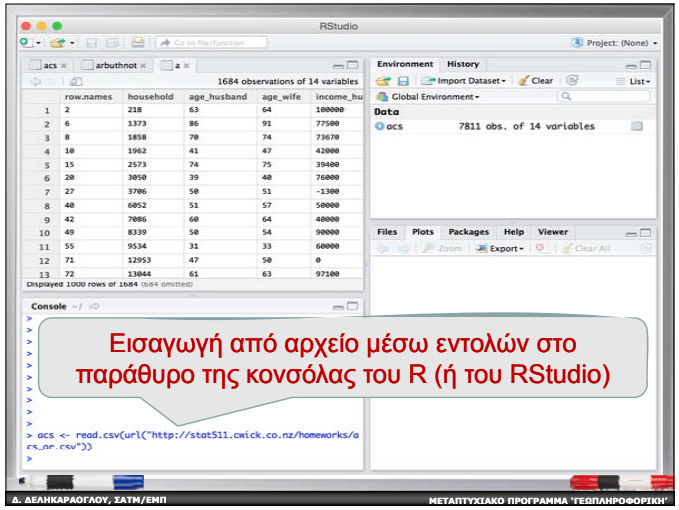
Εισαγωγή διαθέσιμων R δεδομένων

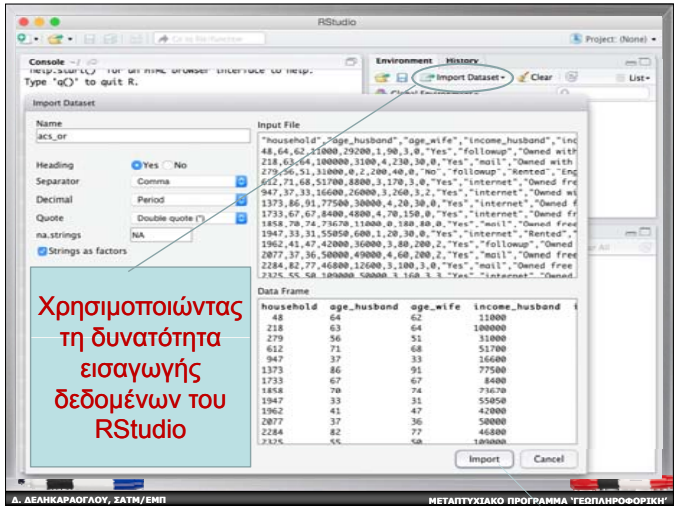
https://vincentarelbundock.github.io/Rdatasets/datasets.html

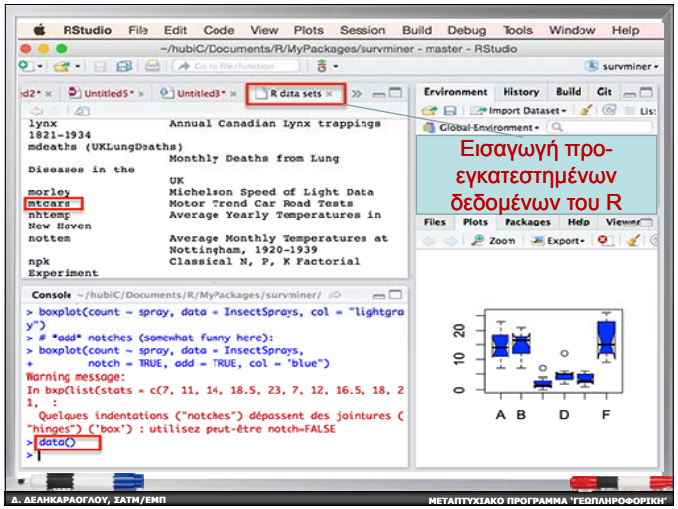
- Συχνά, για περιπτώσεις που απλά επιδιώκεται να δοκιμαστεί πως φορτώνουν ή πως λειτουργούν βασικές συναρτήσεις του R, συνήθως μπορούν να χρησιμοποιηθούν ενσωματωμένα σύνολα δεδομένων που διατίθενται με την εγκατάσταση του R

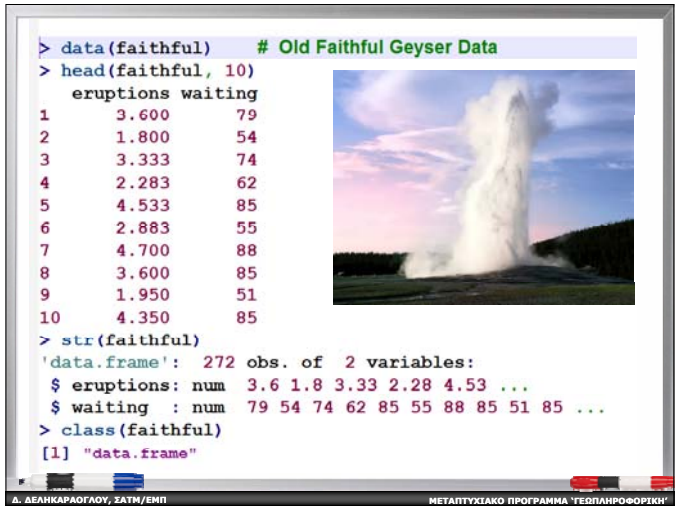
```
data()                  # inspect the available build-in datasets
data(dataset_name)      # load a specific dataset
head(dataset_name, 6)   # print e.g. the first 6 rows
? dataset_name          # get help on the specific dataset
```

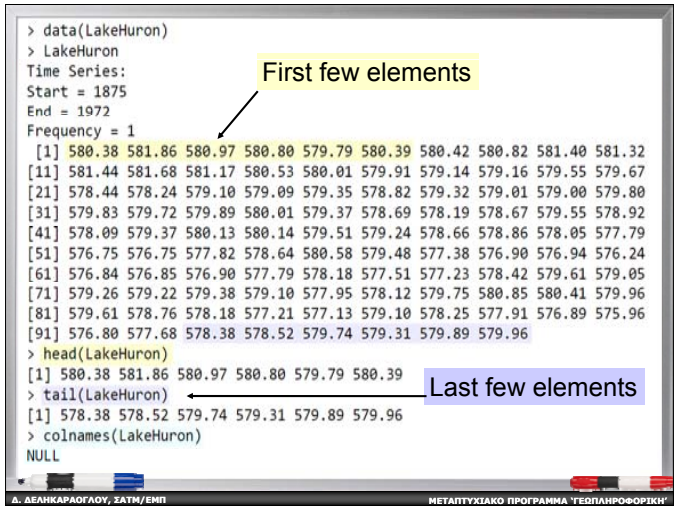
Package	Item	Title	Rows	Cols	n_binary	n_character	n_factor	n_logical	n_numeric	CSV	Doc
boot	acme	Monthly Excess Returns	60	3	0	1	0	0	2	CSV	DOC
boot	aids	Delay in AIDS Reporting in England and Wales	570	6	1	0	0	0	6	CSV	DOC
boot	aircondit	Failures of Air-conditioning Equipment	12	1	0	0	0	0	1	CSV	DOC
boot	aircondit7	Failures of Air-conditioning Equipment	24	1	0	0	0	0	1	CSV	DOC
boot	amis	Car Speeding and Warning Signs	8437	4	1	0	0	0	4	CSV	DOC
boot	aml	Remission Times for Acute Myelogenous Leukaemia	23	3	2	0	0	0	3	CSV	DOC
boot	beaver	Beaver Body Temperature Data	100	4	2	0	0	0	4	CSV	DOC
boot	bigcity	Population of U.S. Cities	49	2	0	0	0	0	2	CSV	DOC
boot	brambles	Spatial Location of Bramble Canes	823	3	0	0	0	0	3	CSV	DOC
boot	breslow	Smoking Deaths Among Doctors	10	5	1	0	1	0	4	CSV	DOC
boot	calcium	Calcium Uptake Data	27	2	0	0	0	0	2	CSV	DOC
boot	cane	Sugar-cane Disease Data	180	5	0	0	2	0	3	CSV	DOC
boot	capability	Simulated Manufacturing Process Data	75	1	0	0	0	0	1	CSV	DOC







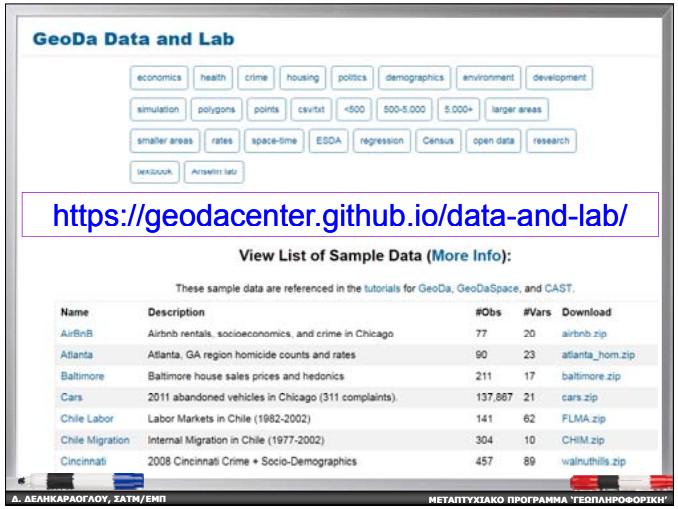




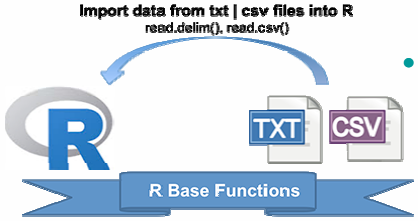

```
# define this function
> getdata <- function(...)
+ {
+     e <- new.env()
+     name <- data(..., envir = e)[1]
+     e[[name]]
+ }
>
# now load your data calling getdata()
> x <- getdata("faithful")
> str(x)
'data.frame':  272 obs. of  2 variables:
 $ eruptions: num  3.6 1.8 3.33 2.28 4.53 ...
 $ waiting  : num  79 54 74 62 85 55 88 85 51 85 ...
>
> xx <- list(x=getdata(faithful),z=getdata(sunspots))
> str(xx)
List of 2
 $ x:'data.frame':  272 obs. of  2 variables:
  ..$ eruptions: num [1:272] 3.6 1.8 3.33 2.28 4.53 ...
  ..$ waiting  : num [1:272] 79 54 74 62 85 55 88 85 51 85 ...
 $ z: Time-Series [1:2820] from 1749 to 1984: 58 62.6 70 55.7 85 83.5 94.8 ...
```



```
library(MASS)     #  load the package MASS
data()            #  list the datasets in it

Data sets in package 'datasets':
AirPassengers          Monthly Airline Passenger Numbers 1949-1960
BJsales                Sales Data with Leading Indicator
BJsales.lead (BJsales) Sales Data with Leading Indicator
BOD                    Biochemical Oxygen Demand
CO2                    Carbon Dioxide uptake in grass plants
ChickWeight            Weight versus age of chicks on different diets
DNase                  Elisa assay of DNase
--------

Data sets in package 'MASS':
Aids2                  Australian AIDS Survival Data
Animals                Brain and Body Weights for 28 Species
...
oats                   Data from an Oats Field Trial
painters               The Painters Data of de Piles
---
phones                 Belgium Phone Calls 1950-1973
---
data(phones) ; phones
$year
 [1] 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73
$calls
 [1]  4.4  4.7  4.7  5.9  6.6  7.3  8.1  8.8 10.6 12.0 13.5 14.9
[13] 16.1 21.2 119.0 124.0 142.0 159.0 182.0 212.0  43.0  24.0  27.0  29.0
```

Ανάγνωση δεδομένων από επίπεδα (ASCII) αρχεία δεδομένων



- Απλά αρχεία δεδομένων με μορφή πίνακα (π.χ. αρχεία excel, πινακοποιημένα δεδομένα σε ιστοσελίδες, ...)

5λεπτο βίντεο:
https://www.youtube.com/watch?v=YWvU9GTyGkg&feature=youtu.be


```
     2.11           Observation        G (GPS)             RINEX VERSION / TYPE
FreeFlyer           a.i. solutions     10/15/2009 13:51:08 PGM / RUN BY / DATE
                                                           MARKER NAME
FFUser              FFUser                                  OBSERVER / AGENCY
                                                           REC # / TYPE / VERS
                                                           ANT # / TYPE
        0.0000         0.0000         0.0000               APPROX POSITION XYZ
        0.0000         0.0000         0.0000               ANTENNA: DELTA H/E/N
     1     1                                               WAVELENGTH FACT L1/2
     4    C1    C2    P1    P2                            # / TYPES OF OBSERV
  2008     8     1     0     0   14.0000000     GPS         TIME OF FIRST OBS
                                                           END OF HEADER

 08  8  1  0  0 14.0000000  0 12G02G05G09G10G12G14G15G17G18G21G22G24
  27631235.834    27631243.638    27631240.407    27631238.556
  21554346.831    21554345.592    21554349.678    21554343.778
  19705383.109    19705378.674    19705378.907    19705383.851
  23810273.723    23810272.598    23810274.460    23810272.339
  20832107.402    20832108.345    20832108.947    20832109.194
  26706262.088    26706267.362    26706265.857    26706264.032
  21706635.676    21706633.615    21706635.452    21706638.599
  26927865.582    21332007.346    21332011.420    21332013.064
  21945748.253    21945747.413    21945747.187    21945750.275
  24989459.233    24989463.784    24989465.153    24989464.876
  22466862.899    22466865.674    22466864.390    22466869.601
```

RINEX αρχεία δεδομένων GNSS

Εισαγωγή: ASCII δεδομένα

- Ανάγνωση αρχείων *.txt, .csv*
  - *Typically, values in text files are often separated, or delimited, by tabs or spaces*

prgtype	gender	id	ses	schtyp	level
general	0	70	4	1	1
vocati	1	121	4	2	1
general	0	86	4	3	1
vocati	0	141	4	3	1
academic	0	172	4	2	1
academic	0	113	4	2	1
general	0	50	3	2	1
academic	0	11	1	2	1

```
prgtype,  gender,  id,  ses,  schtyp,  level
general,  0,  70,  4,  1,  1
vocati,  1,  121,  4,  ,  1
general,  0,  86,  ,  ,  1
vocati,  0,  141,  4,  3,  1
academic,  0,  172,  4,  2,  1
academic,  0,  113,  4,  2,  1
general,  0,  50,  3,  2,  1
academic,  0,  11,  1,  2,  1
```


Εισαγωγή: ASCII δεδομένα

- Ανάγνωση αρχείων *.txt, .csv*
  - *Typically, values in text files are often separated, or delimited, by tabs or spaces*

```
Sepal length   Sepal width   Petal length   Petal width   Species
5.1    3.5    1.4    0.2    I. setosa
4.9    3.0    1.4    0.2    I. setosa
4.7    3.2    1.3    0.2    I. setosa
4.6    3.1    1.5    0.2    I. setosa
5.0    3.6    1.4    0.2    I. setosa
```

Αρχείο τιμών διαχωρισμένων με διαστήματα (tabs)

Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	I. setosa
4.9	3.0	1.4	0.2	I. setosa
4.7	3.2	1.3	0.2	I. setosa
4.6	3.1	1.5	0.2	I. setosa
5.0	3.6	1.4	0.2	I. setosa

Αντιστοιχεί στα δεδομένα πίνακα

Εισαγωγή: ASCII δεδομένα

- Χρειάζεται προσοχή σε περιπτώσεις που οι οριοθέτες των στοιχείων επαναλαμβάνονται συνεχώς πολλές φορές μεταξύ τιμών → *repeated separators*

(πολλαπλά κενά, κόμματα, άλλοι χαρακτήρες) → διασύνδεση της εντολής ανάγνωσης με το όρισμα εισόδου 'sep'

```
prgtype,  gender,  id,  ses,  schtyp,  level
general,  0,  70,  4,  1,  1
vocati,  1,  121,  4,  ,  1
general,  0,  86,  ,  ,  1
vocati,  0,  141,  4,  3,  1
academic,  0,  172,  4,  2,  1
academic,  0,  113,  4,  2,  1
general,  0,  50,  3,  2,  1
academic,  0,  11,  1,  2,  1
```


- Η συνάρτηση *read.table* είναι ο πιο βολικός τρόπος ανάγνωσης δεδομένων στην μορφή ενός ορθογώνιου πίνακα τιμών
  - Υπάρχουν παρόμοιες συναρτήσεις *(read.csv, read.csv2, read.delim, read.delim2)* που την καλούν αλλά αλλάζουν τις παραμέτρους εισόδου:
- Κατά τη χρήση τους, μερικά διαχειριστικά θέματα που χρήζουν προσοχής είναι:

- Header line
- Separator
- Quoting (" ", ' ', ...)
- Missing values
- Unfilled lines
- White space in character fields
- Blank lines
- Classes for the variables
- Comments


```
# The read.table() function reads a file into data frame in table format
read.table(file, header = FALSE, sep = "", quote = "\"'",
           dec = ".", row.names, col.names,
           as.is = !stringsAsFactors,
           na.strings = "NA", colClasses = NA, nrows = -1,
           skip = 0, check.names = TRUE, fill = !blank.lines.skip,
           strip.white = FALSE, blank.lines.skip = TRUE,
           comment.char = "#",
           allowEscapes = FALSE, flush = FALSE,
           stringsAsFactors = default.stringsAsFactors(),
           fileEncoding = "", encoding = "unknown", text)
```

- file: file name ← **can be .txt or .csv files**
- header: 1st line as header or not, logical
- sep: field separator
- quote: quoting characters
- ...

Σημειώστε ότι το πρώτο όρισμα της εντολής 'read.table()' δεν είναι πάντα όνομα αρχείου, αλλά θα μπορούσε ενδεχομένως να είναι επίσης μια ιστοσελίδα που περιέχει δεδομένα.

```
#    The header argument specifies whether or not you have
#    specified column names in your data file
#    Τυπικά στην κλήση της συνάρτησης 'read.table()'
#    θα πρέπει να δίνεται το όνομα του αρχείου και η επέκταση.

df <- read.table("https://somewebsite.com/data_sets/test.txt",
                 header = FALSE)
df
```

	V1	V2	V3
1	1	6	a
2	2	7	b
3	3	8	c
4	4	9	d
5	5	10	e

Εισαγωγή δεδομένων σε αρχεία κειμένου από τον τρέχοντα χώρο εργασίας του R ή διευθύνσεις URL → αποθήκευση των δεδομένων σε ένα πλαίσιο δεδομένων


```
# read.csv and read.csv2 are identical to read.table except for the defaults
read.csv(file, header = TRUE, sep = ",", quote="\"", dec=".",
         fill = TRUE, comment.char="", ...)

read.csv2(file, header = TRUE, sep = ";", quote="\"", dec=",",
          fill = TRUE, comment.char="", ...)
# read.csv2 is used in countries that use a comma as decimal point
#     and a semicolon as field separator

# Similarly, read.delim and read.delim2 are for reading delimited files,
#     defaulting to the TAB character for the delimiter
read.delim(file, header = TRUE, sep = "\t", quote="\"", dec=".",
           fill = TRUE, comment.char="", ...)

read.delim2(file, header = TRUE, sep = "\t", quote="\"", dec=",",
            fill = TRUE, comment.char="", ...)
```

> *mydata <- read.csv("filename.txt")*

- Τα δεδομένα του αρχείου αποθηκεύονται σε ένα εξαιρετικά εύχρηστο τύπο δεδομένων για το R → *Ένα **πλαίσιο δεδομένων** οργανωμένο με σειρές και στήλες, παρόμοιο με ένα υπολογιστικό φύλλο ή πίνακα βάσης δεδομένων*
- Η προηγούμενη κλίση της συνάρτησης ***read.csv*** υποθέτει ότι το αρχείο έχει μια γραμμή κεφαλίδας, δηλ. η σειρά 1 περιέχει το όνομα κάθε στήλης, αλλιώς πρέπει να επεκταθεί η εντολή με την παράμετρο εισόδου *header*

> *mydata <- read.csv("filename.txt", header=FALSE)*

ή γενικότερα για αρχεία όπου

```
# η 1η σειρά περιέχει ονόματα μεταβλητών (στηλών), ο οριοθέτης τιμών είναι το ‘,’, και τα ονόματα των γραμμών καθορίζονται από μια μεταβλητή ‘id’
# note the / instead of \ on MSWindows systems
> mydata <- read.table("c:/myfile.csv", header=TRUE, sep=",", row.names="id")
```

- Εάν τα δεδομένα χρησιμοποιούν, όχι ένα κόμμα, αλλά έναν άλλο χαρακτήρα για να διαχωρίσουν τα πεδία τιμών, υπάρχει επίσης η γενικότερη συνάρτηση ***read.table()***, π.χ.

```
> mydata <- read.table("filename.txt", sep="\t", header=TRUE)
```



```
> x <- read.table("tp.txt",header=T,sep="\t");
> is.data.frame(x)
[1] TRUE
> x
      X t1 t2 t3 t4 t5 t6 t7 t8
1    r1  1  0  1  0  0  1  0  2
2    r2  1  2  2  1  2  1  2  1
3    r3  0  0  0  2  1  1  0  1
. . . . . . . . . . . . . . . .
18 r18  1  1  0  0  1  0  1  2
19 r19  0  1  1  1  1  0  0  1
20 r20  0  0  2  1  1  0  0  1

> ncol(x)
[1] 9
> nrow(x)
[1] 20
```



```
# complete data, space delimited, variable names in first row
test <- read.table("https://ntua.gr.edu/data/test.txt", header = TRUE))
   prgtype gender  id ses schtyp level
1  general      0  70   4      1     1
2  cont.edu     1 121   4      2     1
3  general      0  86   4      3     1
4  cont.edu     0 141   4      3     1
5  academic     0 172   4      2     1
6  academic     0 113   4      2     1
# showing the file with missing values, space delimited (test_missing.txt data file)
prgtype  gender  id ses schtyp  level
  general    0  70   4   1       1
  cont.edu   1 121   4           1
  general    0  86               1
  cont.edu   0 141   4   3       1
  academic   0 172   4   2       1
  academic   0 113   4   2       1
test.missing <- read.table("https://ntua.gr.edu/data/test_missing.txt",
    header = TRUE)
```

αρχεία μπορεί να φορτωθούν και από το διαδίκτυο

δεδομένα με κενά δημιουργούν προβλήματα ανάγνωσης


```
R- Import (Reading) Data from the Web
read.table("http://.../data/some.data")
Error in scan(file = file, what = what, sep = sep,
  quote = quote, dec = dec, : line 1 did not have 13 elements

read.table("http://.../data/some.data",
skip=1, nrow=70, na.strings="-99.90")
V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13
1 1948 -99.90 -99.90 -99.90 -99.90 -99.90 -99.90 -99.90 -99.90 -99.90 -99.90 -99.90 -99.90
2 1949 -99.90 -99.90 -99.90 -99.90 -99.90 -99.90 -99.90 -99.90 -99.90 -99.90 -99.90 -99.90
3 1950 0.56 0.01 -0.78 0.65 -0.50 0.25 -1.23 -0.19 0.39 1.43 -1.46 -1.03
4 1951 -0.42 0.35 -1.47 -0.38 -0.50 -1.35 1.39 -0.41 -1.18 2.54 -0.54 1.13
5 1952 0.57 -1.38 -1.97 0.95 -0.99 -0.10 -0.06 -0.49 -0.38 -0.28 -1.32 -0.49
6 1953 -0.12 -1.00 -0.45 -1.96 -0.56 1.41 0.43 -1.04 -0.19 1.95 0.96 -0.52
7 1954 -0.08 0.40 -1.27 1.31 -0.03 0.06 -0.57 -2.57 -0.28 1.16 0.29 0.55
8 1955 -2.65 -1.71 -0.96 -0.60 -0.26 -0.80 1.78 1.25 0.46 -1.09 -1.49 0.07
9 1956 -0.76 -1.71 -0.46 -1.30 2.10 0.41 -0.72 -1.89 0.38 1.47 0.40 0.00
10 1957 0.71 -0.32 -1.73 0.39 -0.68 -0.42 -1.16 -0.83 -1.47 1.95 0.63 0.02
```

Χωρίς τη χρήση συγκεκριμένων παραμέτρων εισόδου, η *read.table()* διαβάζει όλες τις στήλες ως διανύσματα χαρακτήρων και στη συνέχεια προσπαθεί να επιλέξει μια κατάλληλη κλάση για κάθε μεταβλητή στο πλαίσιο δεδομένων (λογικές, ακέραιες, αριθμητικές ... τιμές). Εάν όλα αυτά αποτύχουν, η μεταβλητή μετατρέπεται σε παράγοντα (factor).


```
# showing the file with missing data, comma delimited (test_missing_comma.txt data file)
prgtype,  gender,  id,  ses,  schtyp,  level
   general,    0,  70,    4,       1,      1
  cont.edu,    1, 121,    4,       ,      1
   general,    0,  86,    ,       ,      1
  cont.edu,    0, 141,    4,       3,      1
  academic,    0, 172,    4,       2,      1
  academic,    0, 113,    4,       2,      1
# the easiest way to fix the missing values problem is to change the type of delimiter
  test.missing <- read.table("https://.../data/test_missing_comma.txt",
    header = TRUE, sep = ",")
    prgtype gender  id ses schtyp level
1  general      0  70   4      1     1
2  cont.edu     1 121   4     NA     1
3  general      0  86  NA     NA     1
4  cont.edu     0 141   4      3     1
5  academic     0 172   4      2     1
6  academic     0 113   4      2     1
```

Σε αρχείαν δεδομένα οροθετημένα με κόμματα, τα κενά αντικαθίστανται με NA


```
# Assume that file "doublesep.txt" contains the following data
A::B::C
23::34::56
12::56::87
90::43::74
lines <- readLines("doublesep.txt")   # read the lines one by one
> lines
[1] "A::B::C"     "23::34::56" "12::56::87" "90::43::74"

lines <- gsub("::", ",", lines)
> lines
[1] "A,B,C"     "23,34,56" "12,56,87" "90,43,74"
# you can either convert to a data.frame object
> read.table(text=lines, sep=",", header=T)
   A  B  C
1 23 34 56
2 12 56 87
3 90 43 74
# or you can write to data file
> writeLines(lines, "doubletosingle.csv")
```

δεδομένα που διαχωρίζονται με ::

Αντικατάσταση ‘ :: ’ με ‘ , ’

ή δεδομένα που διαχωρίζονται με πολλαπλούς κενούς χαρακτήρες

```
# Suppose in the data lines, two variables are separated
#      with two consecutive white spaces

# Trying to use the read.table function will not work,
#      since only one character is allowed for separator.
txt<-"2013-08-13 19:26:58 Method for modifying a piece of 3D geometry
2013-08-13 19:26:57 Method of interactively modifying a feature"

# The gsub() function replaces all matches of a string
#      gsub(pattern, replacement, x, ignore.case = FALSE, perl = FALSE,
#           fixed = FALSE, useBytes = FALSE)
read.table(sep="|",text=gsub("  ","|",txt), header=F)

                   V1                                  V2
1 2013-08-13 19:26:58 Method for modifying a piece of 3D geometry
2 2013-08-13 19:26:57 Method of interactively modifying a feature
```


- Η συνάρτηση *read.table* δεν είναι το σωστό εργαλείο για την ανάγνωση μεγάλων πινάκων, ειδικά εκείνων με πολλές στήλες: έχει σχεδιαστεί για να διαβάζει *πλαίσια δεδομένων τα* οποία μπορεί να έχουν στήλες πολύ διαφορετικών κλάσεων.
- Αντί της *read.table* συχνά χρησιμοποιείται η συνάρτηση *scan()*

```
# Read data into a vector or list from the console or file
scan(file = "", what = double(0), nmax = -1, n = -1, sep = "",
     quote = if(identical(sep, "\n")) "" else "'\"", dec = ".",
     skip = 0, nlines = 0, na.strings = "NA",
     flush = FALSE, fill = FALSE, strip.white = FALSE,
     quiet = FALSE, blank.lines.skip = TRUE, multi.line = TRUE,
     comment.char = "", allowEscapes = FALSE, encoding = "unknown")
```

```
> x <- scan()    # Reading in numeric data
1: 3 5 6
4: 3 5 78 29
8: 34 5 1 78
12:
Read 11 items
> x  ;  mode(x)
[1]  3  5  6  3  5 78 29 34  5  1 78
[1] "numeric"
# Reading in string data, empty quotes indicates character input
> y <- scan(what=" ")
1: red blue
3: green red
5: blue yellow
7:
Read 6 items
> y ; mode(y)
[1] "red"    "blue"   "green"  "red"    "blue"   "yellow"
[1] "character"
```



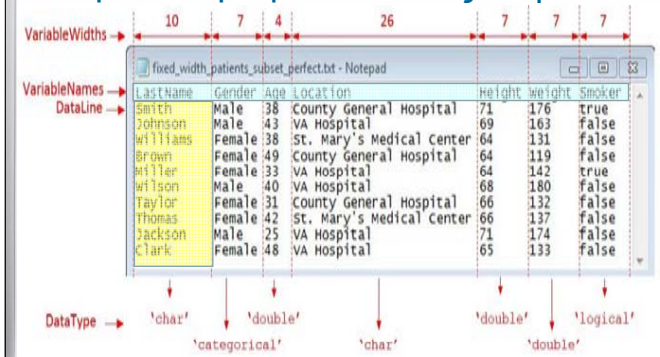
```
# Read data from the console or file (e.g., ordermatrix.csv) containing
,t1,t2,t3,t4,t5,t6,t7,t8
r1,1,0,1,0,0,1,0,2
r2,1,2,5,1,2,1,2,1
r3,0,0,9,2,1,1,0,1
r4,0,0,2,1,2,0,0,0
r5,0,2,15,1,1,0,0,0
r6,2,2,3,1,1,1,0,0
r7,2,2,3,1,1,1,0,1
#  Πολλά δεδομένα μπορούν να σαρωθούν με απλή αντιγραφή "ctrl + C" και
#  π.χ. από ένα αρχείο κειμένου ή το excel
> x <- scan("ordermatrix.csv",what="character",skip=1,quiet=TRUE);
# Στη συνέχεια, με τη χρήση "ctrl + V" τα δεδομένα επικολλούνται, και
#  ο τύπος των δεδομένων καθορίζεται αυτόματα (ως vector, list, ...), ανάλογα
#  με τον τύπο των δεδομένων από την παράμετρο what ():  type of data
> x
[1] "r1,1,0,1,0,0,1,0,2"  "r2,1,2,5,1,2,1,2,1"  "r3,0,0,9,2,1,1,0,1"
[4] "r4,0,0,2,1,2,0,0,0"  "r5,0,2,15,1,1,0,0,0" "r6,2,2,3,1,1,1,0,0"
[7] "r7,2,2,3,1,1,1,0,1"
```



```
# inputting a text file and outputting a list
(x <- scan("https://survey.ntua.gr.edu/stat/data/scan.txt",
    what = list(age = 0,name = "")))
$age
[1] 12 24 35 20

$name
[1] "nicholas"   "katerina"   "ioanna"   "michael"

# using the same text file and saving only the names as a vector
x <- scan("https://survey.ntua.gr.edu/stat/data/scan.txt",
what = list(NULL, name = character()))

x <- x[sapply(x, length) > 0]
$name
[1] "nicholas"   "katerina"   "ioanna"   "michael"
is.vector(x)
[1] TRUE
```

Η συνάρτηση scan επιστρέφει τα δεδομένα ως list ή vector

Ανάγνωση σε δεδομένων με προκαθορισμένο πλάτος στηλών



- Μερικές φορές, τα αρχεία δεδομένων δεν έχουν χαρακτήρες οριοθέτησης των πεδίων τιμών αλλά τα δεδομένα είναι σε στήλες καθεμία με προκαθορισμένο πλάτος (*Fixed-width-format files*).
- Η συνάρτηση *read.fwf* παρέχει ένα απλό τρόπο ανάγνωσης τέτοιων αρχείων, προσδιορίζοντας ένα διάνυσμα τιμών του πλάτους των στηλών

read.fwf(file, widths, header = FALSE, sep = "\t", skip = 0, row.names, col.names, n = -1, buffersize = 2000, fileEncoding = "", ...)


```
Column1 Column2   Column3            Column4Column5
1647    pi        'important'        3.141596.28318
1731    euler     'quite important'  2.718285.43656
1979    answer    'The Answer.'      42     42
```

fwf : αρχεία όπου κάθε στήλη έχει ακριβώς το ίδιο πλάτος

```
df <- read.fwf('constants.txt', widths = c(8,10,18,7,8),
  +            header = FALSE, skip = 1)
df
#>      V1     V2                 V3         V4         V5
#> 1 1647     pi          'important'    3.14159    6.28318
#> 2 1731  euler    'quite important'    2.71828    5.43656
#> 3 1979 answer        'The Answer.'    42         42.0000
```



```
#  Fixed-width-format files
> ff <- tempfile()
> cat(file=ff, "123456", "987654", sep="\n")
> read.fwf(ff, width=c(1,2,3))
V1 V2 V3
1  1 23 456
2  9 87 654
> unlink(ff)
> cat(file=ff, "123", "987654", sep="\n")
> read.fwf(ff, width=c(1,0,2,3))
V1 V2 V3 V4
1  1 NA 23 NA
2  9 NA 87 654
```



```
Weekly SST data starts week centered on 3Jan1990

                Nino1+2      Nino3       Nino34      Nino4
Week          SST SSTA    SST SSTA    SST SSTA    SST SSTA
03JAN1990     23.4-0.4    25.1-0.3    26.6 0.0    28.6 0.3
10JAN1990     23.4-0.8    25.2-0.3    26.6 0.1    28.6 0.3
17JAN1990     24.2-0.3    25.3-0.3    26.5-0.1    28.6 0.3
24JAN1990     24.4-0.5    25.5-0.4    26.5-0.1    28.4 0.2
31JAN1990     25.1-0.2    25.8-0.2    26.7 0.1    28.4 0.2
07FEB1990     25.8 0.2    26.1-0.1    26.8 0.1    28.4 0.3
14FEB1990     25.9-0.1    26.4 0.0    26.9 0.2    28.5 0.4
21FEB1990     26.1-0.1    26.7 0.2    27.1 0.3    28.9 0.8
28FEB1990     26.1-0.2    26.7-0.1    27.2 0.3    29.0 0.8
07MAR1990     26.7 0.3    26.7-0.2    27.3 0.2    28.9 0.7
14MAR1990     26.1-0.4    26.9-0.2    27.3 0.1    28.6 0.4
21MAR1990     26.1-0.2    27.2 0.0    27.6 0.3    28.7 0.5
28MAR1990     25.7-0.4    27.5 0.2    27.8 0.3    28.8 0.5
04APR1990     25.6-0.3    27.6 0.3    27.9 0.4    28.8 0.4
```



```
x <- read.fwf(
  file=url("http://www.someweb.com/.../wksst8110.txt"),
  skip=4,
  widths=c(12, 7, 4, 9, 4, 9, 4, 9, 4))
head(x)
          V1   V2   V3   V4   V5   V6   V7   V8  V9
1 03JAN1990 23.4 -0.4 25.1 -0.3 26.6  0.0 28.6 0.3
2 10JAN1990 23.4 -0.8 25.2 -0.3 26.6  0.1 28.6 0.3
3 17JAN1990 24.2 -0.3 25.3 -0.3 26.5 -0.1 28.6 0.3
4 24JAN1990 24.4 -0.5 25.5 -0.4 26.5 -0.1 28.4 0.2
5 31JAN1990 25.1 -0.2 25.8 -0.2 26.7  0.1 28.4 0.2
6 07FEB1990 25.8  0.2 26.1 -0.1 26.8  0.1 28.4 0.3
```

URL to read file

ignore 4 first lines

πλάτος στηλών

- Σε κάποια αρχεία στηλών καθορισμένου πλάτους ενδέχεται να μην έχουμε τα ονόματα των μεταβλητών στην πρώτη γραμμή και επομένως πρέπει να προστεθούν μετά την ανάγνωση στα δεδομένα συχνά σε ξεχωριστό αρχείο
  - χρησιμοποιώντας τη συνάρτηση **dimnames** και τη συμβολοσειρά με τα ονόματα που προσδίδουμε στις μεταβλητές (στήλες) του αρχείου δεδομένων
    - Αυτό είναι ιδιαίτερα βολικό όταν το αρχείο σταθερής μορφής είναι πολύ μεγάλο και έχει πολλές μεταβλητές


```
# Εισαγωγή από ASCII αρχεία – Header & Data σε ξεχωριστά αρχεία

names <- scan("https://.../data/names.txt", what = character())
[1] "prgtyp" "gender" "id"     "ses"    "schtyp" "level"

test.fixed <- read.fwf("https://.../test_fixed.txt",
    col.names = names, width = c(8, 1, 3, 1, 1, 1)))

# By using the col.names option in the read.fwf function,
#     the object names will supply the variables names.

    prgtype gender  id ses schtyp level
1  general      0  70   4      1     1
2  cont.edu     1 121   4      2     1
3  general      0  86   4      3     1
4  cont.edu     0 141   4      3     1
5  academic     0 172   4      2     1
6  academic     0 113   4      2     1
```


Table A

Parcel-ID	Acres
2	2
5	1.5
6	6
1	3
8	1.6

+

Table B

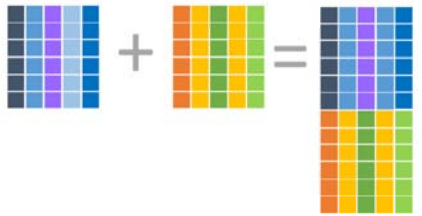
Parcel-ID	Owner
2	John Smith
5	Bruce Martin
6	Anne Davis
1	Steve Arnold
8	Rick James

=

Parcel-ID	Acres	Owner
2	2	John Smith
5	1.5	Bruce Martin
6	6	Anne Davis
1	3	Steve Arnold
8	1.6	Rick James

Συνένωση αρχείων δεδομένων

- Συγχώνευση οριζόντια και κάθετα
- Αριστερή-, εσωτερική-ένωση, πλήρης ένταξη



Συνένωση 'παρόμοιων' αρχείων

```
Syntax
merge(x, y, ...)
# For data frames:
merge(x, y, # Data frames or objects to be coerced
      by = intersect(names(x), names(y)), # Columns used for merging
      by.x = by, by.y = by, # Columns used for merging
      all = FALSE, # If TRUE, all.x = TRUE and all.y = TRUE
      all.x = all, all.y = all, # If TRUE, adds rows for each row in x (y) that not match a row in y (x).
      sort = TRUE, # Whether to sort the output by the 'by' columns
      suffixes = c(".x",".y"), # Suffixes for creating unique column names
      no.dups = TRUE, # Whether to avoid duplicated column names appending more suffixes or not
      incomparables = NULL, # How to deal with values that can not be matched
      ...) # Additional arguments
#   However, merge is a generic function that can be also used with other objects
#   (like vectors or matrices), but they will be coerced to data.frame class

# Merge two data files, dataset1 and dataset2, into a single data set
merged.data <- merge(dataset1, dataset2, by="uniqueID")

# Merge two data files, dataset1 and dataset2, by more than one id variable
merged.data <- merge(dataset1, dataset2, by=c("regionID", "uniqueID"))

# Merge the two files if the unique id variable has a different name in each data set
merged.data <- merge(dataset1, dataset2, by.x="countryID", by.y="stateID")
```

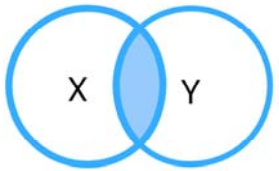


```
#  Merge data frames
> student_id <- 1:10
> student_name <- c("Andrew", "Susan", "John", "Joe", "Jack",
+                   "James", "Mary", "Kate", "Jacqueline", "Nicholas")
> student_loan <- round(rnorm(10, mean = 800, sd = 100))
> student_age <- round(rnorm(10, mean = 20, sd = 3))
> student_position <- c(rep("Engineer", 2), "Admin", rep("Technician", 7))
>
>
> df_1 <- data.frame(id = student_id[1:8], name = student_name[1:8],
+                    month_loan = student_loan[1:8])
> df_2 <- data.frame(id = student_id[-5], name = student_name[-5],
+                    age = student_age[-5], position = student_position[-5])
>
> df_1
  id   name month_loan
1  1 Andrew        633
2  2  Susan        935
3  3   John        714
4  4    Joe        784
5  5   Jack        933
6  6  James        953
7  7   Mary        714
8  8   Kate        746
>
> df_2
  id       name age   position
1  1     Andrew  14   Engineer
2  2      Susan  15   Engineer
3  3       John  19      Admin
4  4        Joe  18 Technician
5  6      James  24 Technician
6  7       Mary  21 Technician
7  8       Kate  17 Technician
8  9 Jacqueline  20 Technician
9 10    Nicholas  27 Technician
```



```
> merge(x = df_1, y = df_2)
  id   name month_loan age   position
1  1 Andrew        633  14   Engineer
2  2  Susan        935  15   Engineer
3  3   John        714  19      Admin
4  4    Joe        784  18 Technician
5  6  James        953  24 Technician
6  7   Mary        714  21 Technician
7  8   Kate        746  17 Technician
> merge(x = df_1, y = df_2,
+ by = c("id", "name")) # Equivalent
  id   name month_loan age   position
1  1 Andrew        633  14   Engineer
2  2  Susan        935  15   Engineer
3  3   John        714  19      Admin
4  4    Joe        784  18 Technician
5  6  James        953  24 Technician
6  7   Mary        714  21 Technician
7  8   Kate        746  17 Technician
>
```

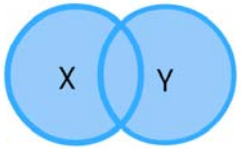
INNER JOIN



Σε αυτό τον τύπο ένωσης δεδομένων, στην προκύπτουσα έξοδο *λείπουν τα μη κοινά στοιχεία* των δύο συνόλων δεδομένων


```
>
> #Full (outer) join
> merge(x = df_1, y = df_2, all = TRUE)
   id       name month_loan age   position
1   1     Andrew        633  14   Engineer
2   2      Susan        935  15   Engineer
3   3       John        714  19      Admin
4   4        Joe        784  18 Technician
5   5       Jack        933  NA       <NA>
6   6      James        953  24 Technician
7   7       Mary        714  21 Technician
8   8       Kate        746  17 Technician
9   9 Jacqueline         NA  20 Technician
10 10    Nicholas         NA  27 Technician
>
```

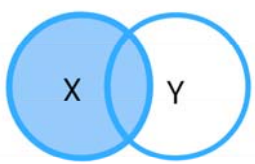
OUTER JOIN



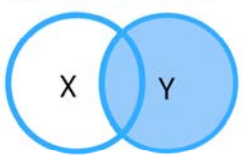
Πλήρης εξωτερική ένωση ή πλήρης σύνδεση, συγχωνεύει όλες τις στήλες και των δύο συνόλων δεδομένων σε μία για όλα τα στοιχεία


```
#  Left (outer) join
> merge(x = df_1, y = df_2, all.x = TRUE)
  id   name month_loan age   position
1  1 Andrew        633  14   Engineer
2  2  Susan        935  15   Engineer
3  3   John        714  19      Admin
4  4    Joe        784  18 Technician
5  5   Jack        933  NA       <NA>
6  6  James        953  24 Technician
7  7   Mary        714  21 Technician
8  8   Kate        746  17 Technician
>
#  Right (outer) join
> merge(x = df_1, y = df_2, all.y = TRUE)
  id       name month_loan age   position
1  1     Andrew        633  14   Engineer
2  2      Susan        935  15   Engineer
3  3       John        714  19      Admin
4  4        Joe        784  18 Technician
5  6      James        953  24 Technician
6  7       Mary        714  21 Technician
7  8       Kate        746  17 Technician
8  9 Jacqueline         NA  20 Technician
9 10    Nicholas         NA  27 Technician
```

LEFT JOIN



RIGHT JOIN




```
> #Merge data frames by row names
> df1 <- data.frame(var = c("one", "two",
+ "three", "four", "five"),
+                   data = c(1, 5, 1, 6, 8))
> rownames(df1) <- c("A", "B", "C", "D", "E")
> df1
    var data
A   one    1
B   two    5
C three    1
D  four    6
E  five    8
>
> df2 <- data.frame(var = c("three", "one",
+ "eight", "two", "nine"),
+                   data = c(1, 5, 1, 6, 8))
> rownames(df2) <- c("E", "A", "B", "D", "C")
> df2
    var data
E three    1
A   one    5
B eight    1
D   two    6
C  nine    8
```

Συνένωση πλαισίων δεδομένων ανά ονόματα σειρών

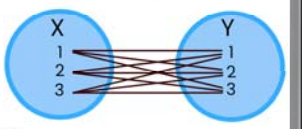
Τυπικά είναι ως μια πλήρης εξωτερική συνένωση …

… που ισοδυναμεί με αριστερή και δεξιά συνένωση

```
> merge(df1, df2, by = 0, all = TRUE)
  Row.names var.x data.x var.y data.y
1         A   one      1   one      5
2         B   two      5 eight      1
3         C three      1  nine      8
4         D  four      6   two      6
5         E  five      8 three      1
> merge(df1, df2, by = "row.names",
+ all = TRUE) # Equivalent
  Row.names var.x data.x var.y data.y
1         A   one      1   one      5
2         B   two      5 eight      1
3         C three      1  nine      8
4         D  four      6   two      6
5         E  five      8 three      1
```

```
> Merged <- merge(x = df_1, y = df_2, by = NULL)
> head(Merged)
  id.x name.x month_loan id.y name.y age position
1    1 Andrew        633    1 Andrew  14 Engineer
2    2  Susan        935    1 Andrew  14 Engineer
3    3   John        714    1 Andrew  14 Engineer
4    4    Joe        784    1 Andrew  14 Engineer
5    5   Jack        933    1 Andrew  14 Engineer
6    6  James        953    1 Andrew  14 Engineer
>
>
> tail(Merged)
   id.x name.x month_loan id.y   name.y age   position
67    3   John        714   10 Nicholas  27 Technician
68    4    Joe        784   10 Nicholas  27 Technician
69    5   Jack        933   10 Nicholas  27 Technician
70    6  James        953   10 Nicholas  27 Technician
71    7   Mary        714   10 Nicholas  27 Technician
72    8   Kate        746   10 Nicholas  27 Technician
>
>
```

CROSS JOIN




```
> x <- data.frame(id = 1:4, year = 1995:1998)
>
> y <- data.frame(id = c(4, 1, 3, 2),
+                 year = c(1998, 1995, 1997, 1996), age = c(22, 25, 23, 24))
>
> z <- data.frame(id = c(1, 2, 3), year = 1995:1997, loan = c(1000, 1200, 1599))
> x
  id year
1  1 1995
2  2 1996
3  3 1997
4  4 1998
> y
  id year age
1  4 1998  22
2  1 1995  25
3  3 1997  23
4  2 1996  24
> z
  id year loan
1  1 1995 1000
2  2 1996 1200
3  3 1997 1599
```

```
> merge(x, merge(y, z))
  id year age loan
1  1 1995  25 1000
2  2 1996  24 1200
3  3 1997  23 1599
>
>
> merge(x, merge(y, z, all = TRUE), all = TRUE)
  id year age loan
1  1 1995  25 1000
2  2 1996  24 1200
3  3 1997  23 1599
4  4 1998  22   NA
```

Συνενώνοντας τη ***merge*** συνάρτηση →

→ Συγχωνεύστε περισσότερα από δύο πλαίσια δεδομένων

Merging two data files

```
pg <- read.csv("data/big5/countries/PG.csv")
str(pg)
'data.frame':   2 obs. of  57 variables:
$ race    : int  3 13
$ age     : int  30 47
$ engnat  : int  1 2
$ gender  : int  1 1
$ hand    : int  1 1
$ source  : int  1 1
$ country : Factor w/ 1 level "PG": 1 1
$ E1      : int  4 1
$ E2      : int  1 2
...
$ O8      : int  5 4
$ O9      : int  5 2
$ O10     : int  4 5
```

Read also a data frame which has 57 columns, and both data sets 'PG.csv' and 'GT.csv' are two-dimensional

```
gt <- read.csv("data/big5/countries/GT.csv")
```


Read also a data frame which has 57 columns, and both data sets 'PG.csv' and 'GT.csv' are two-dimensional

```
gt <- read.csv("data/big5/countries/GT.csv")
```

Before we merge data frames, we need to be sure that the columns mean the same thing in both datasets (i.e., same column names in both datafiles)

Στόχος: *διατηρώντας τις στήλες* του συνόλου δεδομένων, *να προστεθούν περισσότερες σειρές* σε αυτό

```
names(pg) == names(gt)
[1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[15] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[29] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[43] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[57] TRUE
```

To merge these data sets, we simply need to bind them by row

```
pg_and_gt <- rbind(pg, gt)
```



```
inner_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"),
  ...)

left_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"), ...)

right_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"),
  ...)

full_join(x, y, by = NULL, copy = FALSE, suffix = c(".x", ".y"), ...)

semi_join(x, y, by = NULL, copy = FALSE, ...)

anti_join(x, y, by = NULL, copy = FALSE, ...)
```

- Από το πακέτο ***dplyr v0.7.8*** Χρήση γενικών συναρτήσεων συνένωσης σετ δεδομένων x, y προερχόμενων συνήθως από τις ίδιες πηγές

superheroes

name	alignment	gender	publisher
Magneto	bad	male	Marvel
Storm	good	female	Marvel
Mystique	bad	female	Marvel
Batman	good	male	DC
Joker	bad	male	DC
Catwoman	bad	female	DC
Hellboy	good	male	Dark Horse Comics

publishers

publisher	yr_founded
DC	1934
Marvel	1939
Image	1992

Εσωτερική ένωση

inner_join(x = superheroes, y = publishers)

name	alignment	gender	publisher	yr_founded
Magneto	bad	male	Marvel	1939
Storm	good	female	Marvel	1939
Mystique	bad	female	Marvel	1939
Batman	good	male	DC	1934
Joker	bad	male	DC	1934
Catwoman	bad	female	DC	1934

Οι πίνακες A και B συνδέονται με βάση τη στήλη-κλειδί στην οποία αποφασίζουμε να συνδέσουμε τα δεδομένα τους.

Table A

Parcel-ID	Acres
2	2
5	1.5
6	6
1	3
8	1.6

+

Table B

Parcel-ID	Owner
2	John Smith
5	Bruce Martin
6	Anne Davis
1	Steve Arnold
8	Rick James

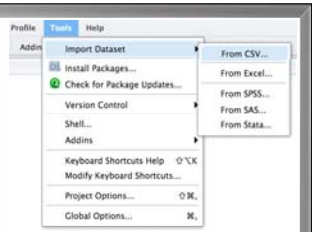
=

Parcel-ID	Acres	Owner
2	2	John Smith
5	1.5	Bruce Martin
6	6	Anne Davis
1	3	Steve Arnold
8	1.6	Rick James

- Σετ δεδομένων που αντιπροσωπεύουν το ίδιο σύνολο παρατηρήσεων, μπορούν να συνδυαστούν οριζόντια → *με χρήση της συνάρτησης cbind()*
- Πρέπει να ελέγχεται ότι η σειρά των παρατηρήσεων είναι η ίδια
- ΔΕΝ ΕΧΕΙ ΝΟΗΜΑ - εάν τα σετ δεδομένων έχουν διαφορετικό αριθμό σειρών ή εάν έχουν τον ίδιο αριθμό σειρών ανόμοια διατεταγμένων


```
# Merging more than two data files, using a user-defined function

batch_read <- function(path, extension) {
  file_names <- list.files(path, pattern = extension)
  data_list <- lapply(paste0(path, file_names), read.csv)
  data_frame <- bind_rows(data_list)
  data_frame
  }

# Now can read as many 'similar' datafiles in file_path
d <- batch_read(file_path, ".csv")   # e.g.: data/big5/countries/

# or write the combined dataframe in a new file
write_csv(d, path = "data/big5/master-data.csv")

dim(d)
[1] 19324    57
```


Ανάγνωση δεδομένων από αρχεία .txt και .csv, στο *RStudio*



- Το RStudio περιλαμβάνει δυνατότητες για την εισαγωγή δεδομένων από τα αρχεία *csv, xls, xlsx, sav, dta, por, sas* και *stata*
  - από το παράθυρο περιβάλλοντος ή
  - από το μενού εργαλείων
- Import from the file system or a URL

Εισαγωγή δεδομένων στο R με τη χρήση πακέτων

Ανάγνωση δεδομένων από αρχεία .txt και .csv, με το πακέτο ***readr***

- Διαθέσιμες συναρτήσεις: *read_delim(), read_tsv(), read_csv(), read_csv2()*
- Ανάγνωση αρχείων τοπικά ή από το διαδίκτυο
  - Διαγνωστικά στην περίπτωση προβλημάτων
- Δυνατότητες καθορισμού του τύπου των δεδομένων σε κάθε στήλη (numeric, character, logical, …)
- Ανάγνωση γραμμών από ένα αρχείο: *read_lines ()*
- Ανάγνωση ολόκληρου του αρχείου: *read_file ()*

Μια πιο γρήγορη (x10) και φιλική λύση σε σχέση με τις παρόμοιες συναρτήσεις του R


```
install.packages("readr")          # Installing
library("readr")                   # Loading
# Read tab separated values
read_tsv(file.choose())
# Read comma (",") separated values
read_csv(file.choose())
# Read semicolon (";") separated values
read_csv2(file.choose())
```



```
# General read function in readr for reading a delimited file
read_delim(file, delim, quote = "\"", escape_backslash = FALSE,
  escape_double = TRUE, col_names = TRUE, col_types = NULL,
  locale = default_locale(), na = c("", "NA"), quoted_na = TRUE,
  comment = "", trim_ws = FALSE, skip = 0, n_max = Inf,
  guess_max = min(1000, n_max), progress = show_progress())
# read_csv() and read_tsv() are special cases of the general read_delim()
  read_csv(file, col_names = TRUE, col_types = NULL,
  locale = default_locale(), na = c("", "NA"), quoted_na = TRUE,
  quote = "\"", comment = "", trim_ws = TRUE, skip = 0, n_max = Inf,
  guess_max = min(1000, n_max), progress = show_progress())
read_csv2(file, col_names = TRUE, col_types = NULL,
  locale = default_locale(), na = c("", "NA"), quoted_na = TRUE,
  quote = "\"", comment = "", trim_ws = TRUE, skip = 0, n_max = Inf,
  guess_max = min(1000, n_max), progress = show_progress())
read_tsv(file, col_names = TRUE, col_types = NULL,
  locale = default_locale(), na = c("", "NA"), quoted_na = TRUE,
  quote = "\"", comment = "", trim_ws = TRUE, skip = 0, n_max = Inf,
  guess_max = min(1000, n_max), progress = show_progress())
```



```
read_delim(file, delim, quote = "\"", escape_backslash = FALSE,
  escape_double = TRUE, col_names = TRUE, col_types = NULL,
  locale = default_locale(), na = c("", "NA"), quoted_na = TRUE,
  comment = "", trim_ws = FALSE, skip = 0, n_max = Inf,
  guess_max = min(1000, n_max), progress = show_progress())
```

- **file** : διαδρομή αρχείου ενδιαφέροντος
  - Τα αρχεία με επέκταση σε .gz, .bz2, .xz ή .zip θα **αποσυμπιέζονται αυτόματα**.
  - Αρχεία που ξεκινούν με " http: // ", " https: // ", " ftp: // " ή "ftps: //" θα ληφθούν αυτόματα. Απομακρυσμένα αρχεία gz, στο διαδίκτυο, μπορούν επίσης να μεταφορτωθούν και να αποσυμπιεστούν αυτόματα.


```
# Reading a text file using file.choose()
myFile = read.delim(file.choose(), header = FALSE)
    # Using the code in RStudio
    # will be asked to choose a file
print(myFile)

# Read text files using the ' readr ' package
    # Import the readr library
library(readr)

# Use read_tsv() to read a tab-separated text file
myData = read_tsv("mydatafile.txt", col_names = FALSE)
print(myData)

# Use read_lines() to read one line at a time
myData = read_lines("mydatafile.txt", n_max = 1)
print(myData)

# Use read_lines() to read two lines at a time
myData = read_lines("mydatafile.txt", n_max = 2)
print(myData)
```



```
read_delim(file, delim, quote = "\"", escape_backslash = FALSE,
  escape_double = TRUE, col_names = TRUE, col_types = NULL,
  locale = default_locale(), na = c("", "NA"), quoted_na = TRUE,
  comment = "", trim_ws = FALSE, skip = 0, n_max = Inf,
  guess_max = min(1000, n_max), progress = show_progress())
```

- **delim** : ο χαρακτήρας που χωρίζει τις τιμές στο αρχείο δεδομένων.
- **col_names** : TRUE, FALSE ή διάνυσμα χαρακτήρων που καθορίζει τα ονόματα στηλών και δεν θα συμπεριληφθεί στο πλαίσιο δεδομένων
  - Εάν είναι TRUE, η πρώτη σειρά των στοιχείων εισόδου θα χρησιμοποιηθεί ως ονόματα στηλών
- **col_type** () : καθορίσει τον τύπο δεδομένων κάθε στήλης ("i", "d", "l", "c", "f", "-" ή "_")

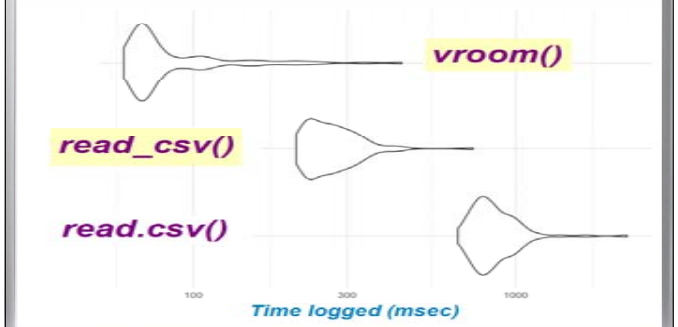

```
read_delim(file, delim, quote = "\"", escape_backslash = FALSE,
  escape_double = TRUE, col_names = TRUE, col_types = NULL,
  locale = default_locale(), na = c("", "NA"), quoted_na = TRUE,
  comment = "", trim_ws = FALSE, skip = 0, n_max = Inf,
  guess_max = min(1000, n_max), progress = show_progress())
```

- **skip**: Αριθμός γραμμών που θα παραληφθούν κατά την ανάγνωση των δεδομένων
- **n_max** : Αριθμοί γραμμών για ανάγνωση.
  - Αν n = -1, θα διαβαστούν όλες οι γραμμές στο αρχείο.
- **progress** : Εμφάνιση μιας μπάρας προόδου της ανάγνωσης του αρχείου που ανανεώνεται κάθε 50000 τιμές και θα εμφανίζεται μόνο εάν ο εκτιμώμενος χρόνος ανάγνωσης είναι 5 δευτερόλεπτα ή περισσότερο.

Καθώς το μέγεθος ενός αρχείου δεδομένων αυξάνεται σε μέγεθος, η εξοικονόμηση απόδοσης στην εισαγωγή δεδομένων είναι πολύ μεγαλύτερη με τη χρήση αντίστοιχων συναρτήσεων από ειδικά πακέτα του R

```
# vroom is a package designed specifically for speed to import plain-text data files
results_vroom <- microbenchmark(
  read.csv = read.csv(file = here("static", "data", "sim-data-large.csv")),
  read_csv = read_csv(file = here("static", "data", "sim-data-large.csv")),
  vroom = vroom::vroom(file = here("static", "data", "sim-data-large.csv"))
)
autoplot(object = results_vroom) +
  scale_y_log10() +
  labs(y = "Time [milliseconds], logged")
```

Καθώς το μέγεθος ενός αρχείου δεδομένων αυξάνεται σε μέγεθος, η εξοικονόμηση απόδοσης στην εισαγωγή δεδομένων είναι πολύ μεγαλύτερη με τη χρήση αντίστοιχων συναρτήσεων από ειδικά πακέτα του R



Καθώς το μέγεθος ενός αρχείου δεδομένων αυξάνεται σε μέγεθος, η εξοικονόμηση απόδοσης στην εισαγωγή δεδομένων είναι πολύ μεγαλύτερη με τη χρήση αντίστοιχων συναρτήσεων από πακέτα του R

```
##Unit: seconds
##        expr   min    lq      mean median    uq   max neval
##     readCSV 200.0 200.0 211.187125  210.0 220.0 240.0    10
##    readrCSV  27.0  28.0  29.770890   29.0  32.0  33.0    10
##       fread  15.0  16.0  17.250016   17.0  17.0  22.0    10
##   loadRdata   4.4   4.7   5.018918    4.8   5.5   5.9    10
##    readRds   4.6   4.7   5.053674    5.1   5.3   5.6    10
## readFeather   1.5   1.8   2.988021    3.4   3.6   4.1    10
```


Εισαγωγή δεδομένων: using the functions in various packages

```
> library(gdata) # load gdata package
> help(read.xls) # documentation
> mydata = read.xls("mydata.xls") # read from first sheet

> library(XLConnect) # load XLConnect package
> wk = loadWorkbook("mydata.xls")
> df = readWorksheet(wk, sheet="Sheet1")

> library(foreign) # load the foreign package
> help(read.spss)   # documentation
> mydata = read.spss("myfile", to.data.frame=TRUE)

> library(foreign) # load the foreign package
> help(read.mtp)    # documentation
> mydata = read.mtp("mydata.mtp") # read from .mtp file
> mydata = read.csv("mydata.csv") # read csv file
```


Εισαγωγή δεδομένων από αρχεία διαφορετικών εκδόσεων excel


```
# Reading data From Excel (.xls, .xlsx) Files into R

install.packages("readxl")     # Install readxl package, then load it
library("readxl")
my_data <- read_excel("my_file.xlsx")

install.packages("xlsx")       # Install xlsx package, then load it
library("xlsx")
my_data <- read.xlsx("my_file.xlsx")
```


Εισαγωγή δεδομένων: **xlsx** package

```
require(xlsx)
read.xlsx("myfile.xlsx", sheetName = "Sheet1")
read.xlsx2("myfile.xlsx", sheetName = "Sheet1")

coln = function(x) { # A function to see column numbers
  y = rbind(seq(1, ncol(x)))
  colnames(y) = colnames(x)
  rownames(y) = "col.number"
  return(y)
}
data = read.xlsx2("myfile.xlsx", 1) # open the file
coln(data) # check the column numbers you want to have
as factors
x = 3 # Say you want columns 1-3 as factors, the rest
numeric
data = read.xlsx2("myfile.xlsx", 1,
  colClasses = c(rep("character", x), rep("numeric",
ncol(data)-x+1))
)
```


Εισαγωγή δεδομένων: Input data into R

```
    # first row contains variable names, comma is
      separator
    # assign the variable id to row names
    # note the / instead of \ on MS Windows systems

mydata <- read.table("c:/mydata.csv", header=TRUE,
    sep=",", row.names="id")
```

Από αρχεία Excel – **Comma Delimited Text File**

```
# read in the first worksheet from the
  workbook excelfile.xlsx
# first row contains variable names

library(xlsx)
mydata <- read.xlsx("c:/excelfile.xlsx", 1)

    # read in the worksheet named mysheet

mydata <- read.xlsx("c:/excelfile.xlsx", sheetName =
"mysheet")
```

& using the **xlsx** package

Δεν απαιτεί εγκατάσταση του MS Excel

```
# Install from CRAN
install.packages("XLConnect")

# or from github repository using the 'devtools' package
require(devtools)

# Installs the master branch of XLConnect (= current development version)
install_github("xlconnect", username = "miraisolutions",
               ref = "master")

# Installs XLConnect 0.2-13
install_github("xlconnect", username = "miraisolutions",
               ref = "0.2-13")
```

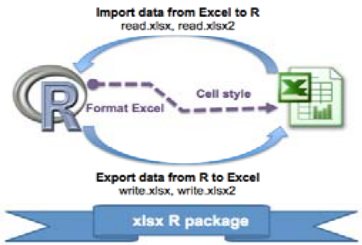
… μόνη απαίτηση είναι μια πρόσφατη έκδοση JRE


```
# Read in existing Excel files into R
df <- readWorksheetFromFile("<file name and extension>",
                            sheet=1,
                            startRow = 4,
                            endCol = 2)

# Alternatively, load in a whole workbook
wb <- loadWorkbook("<name and extension of your file>")
# Load in Worksheet
df <- readWorksheet(wb, sheet=1)

# Reading Data From Workbook (numeric or character), full usage
readWorksheet(object,sheet,startRow,startCol,endRow,endCol,
autofitRow,autofitCol,region,header,rownames,colTypes,forceConversion,
dateTimeFormat,check.names,useCachedValues,keep,drop,simplify,
readStrategy)
```



```
# Using the readxl package ----------------------------------------------
# It's also possible to choose a file interactively
#      using the function file.choose()
my_data <- read_excel(file.choose())

# Assuming that the files "my_file.xls" and "my_file.xlsx"
#      are in the current working directory
# Specify sheet by its name
my_data <- read_excel("my_file.xlsx", sheet = "data")

# Specify sheet by its index
my_data <- read_excel("my_file.xlsx", sheet = 2)

# In case of missing values: NA (not available), other than blank cells,
#      set the na argument
my_data <- read_excel("my_file.xlsx", na = "---")
```

Πακέτο xlsx: **read**, **write** και **format Excel** αρχεία




```
# Using the xlsx package (two main functions: read.xlsx and read.xlsx2)
read.xlsx(file, sheetIndex, header=TRUE)
read.xlsx2(file, sheetIndex, header=TRUE)

# file: file path
# sheetIndex: the index of the sheet to be read
# header: a logical value. If TRUE, the first row is used as column names.
```


Το πακέτο ***xlsx***

- Λειτουργεί τόσο για αρχεία excel 97 /2000 /XP / 2003 / 2007

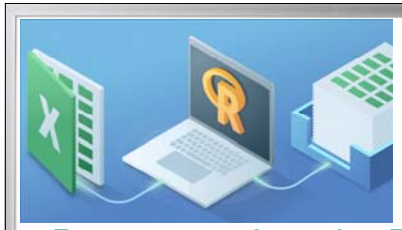
Επιτρέπει

- **Ανάγνωση** και **εξαγωγή** αρχείων του Excel
- **Πρόσθεση συνόλων δεδομένων**, **εικόνες** και **γραφήματα** σε ένα φύλλο εργασίας του Excel
- **Διαμόρφωση της εμφάνισης** του φύλλου εργασίας του Excel ορίζοντας τις μορφές των δεδομένων, γραμματοσειρές, χρώματα και περιθώρια
- Εξαρτάται από τα πακέτα *rJava* και *xlsxjars*


```
library(xlsx)
# full usage statement
read.xlsx(xlsxFile, sheet = 1, startRow = 1, colNames = TRUE,
  rowNames = FALSE, detectDates = FALSE, skipEmptyRows = TRUE,
  skipEmptyCols = TRUE, rows = NULL, cols = NULL, check.names = FALSE,
  namedRegion = NULL, na.strings = "NA", fillMergedCells = FALSE)

file <- system.file("tests", "test_import.xlsx", package = "xlsx")
res <- read.xlsx(file, 1)  # read first sheet
head(res[, 1:6])
```

	NA.	Population	Income	Illiteracy	Life.Exp
1	Alabama	3615	3624	2.1	69.05
2	Alaska	365	6315	1.5	69.31
3	Arizona	2212	4530	1.8	70.55
4	Arkansas	2110	3378	1.9	70.66
5	California	21198	5114	1.1	71.71
6	Colorado	2541	4884	0.7	72.06

xlsx είναι πολύ αργή για μεγάλα σύνολα δεδομένων



πακέτο XLConnect

(Windows, Unix/Linux and Mac)

- Για τον χειρισμό αρχείων Excel από το R
  - Επιτρέπει τη δημιουργία βιβλίων εργασίας (workbooks) του Excel, με πολλαπλά φύλλα και την εισαγωγή δεδομένων σε αυτά.
  - Ανάγνωση υπαρχόντων αρχείων του Excel στο R
  - Δεν απαιτεί εγκατάσταση του Excel ή άλλους 'οδηγούς' για την ανάγνωσή και δημιουργία αρχείων Excel


```
# Load the XLConnect package
require(XLConnect)

# Download sample data
download.file("http://files.someweb.com/example.xlsx",
destfile = "example.xlsx")

# Load the workbook
wb = loadWorkbook("example.xlsx")

# Read the data on the sheet named "data"
data = readWorksheet(wb, sheet = "data")
print(head(data))

# Let's assume know the data is located within some
# given boundaries (exact location would be startRow = 7,
# startCol = 7, endRow = 46, endCol = 23)
data = readWorksheet(wb, sheet = "data", startRow = 7,
startCol = 7, endRow = 46, endCol = 23)
print(head(data))
```



```
# Load an  Excel workbook (create if not existing)
wb = loadWorkbook("../data/xlconnect1.xlsx",create=T)
createSheet(wb,"Rivers")
createSheet(wb,"Quakes location Fiji")

writeWorksheet(wb,rivers,"Rivers")
writeWorksheet(wb,quakes,"Quakes location Fiji")

saveWorkbook(wb)
# Alternatively you can pass a filename
# saveWorkbook(wb,'path_to_file')

wb1 = loadWorkbook("../data/example1.xlsx", create =
TRUE)
createSheet(wb1, name = "chickSheet")
writeWorksheet(wb1, ChickWeight, sheet = "chickSheet",
startRow = 3, startCol = 4)
saveWorkbook(wb1)
```

Εξαιρετικά παραδείγματα:
https://miraisolutions.wordpress.com/


```
install.packages(pkgs="gdata") ; library(gdata)
dat <- read.xls("smalltest.xlsx") ; head(dat)
```

	Person	Potato	Carrot	Tomato
1	1	4	10	1
2	2	10	5	1
3	3	6	10	8
4	4	9	2	4
5	5	1	3	9
6	6	10	9	7

```
# Can access multiple sheets in a workbook, by name or number
dat <- read.xls("http://db.tt/zOCGC4ve", sheet="bread")
head(dat)
```

	Person	White	Brown	Wholewheat
1	1	4	6	9
2	2	2	9	10
3	3	4	9	8
4	4	9	7	8
5	5	1	1	8
6	6	4	1	3

gdata package

... διάφορα εργαλεία προγραμματισμού R για τη διαχείριση δεδομένων

Εισαγωγή δεδομένων: using the functions in the **foreign** package

```
data.restore {foreign}    Read an S3 Binary or data.dump File
lookup.xport {foreign}  Lookup Information on a SAS XPORT Format
                          Library
read.arff {foreign}   Read Data from ARFF Files
read.dbf {foreign}    Read a DBF File
read.dta {foreign}    Read Stata Binary Files
read.epiinfo {foreign}    Read Epi Info Data Files
read.mtp {foreign}    Read a Minitab Portable Worksheet
read.octave {foreign}Read Octave Text Data Files
read.S {foreign}      Read an S3 Binary or data.dump File
read.spss {foreign}   Read an SPSS Data File
read.ssd {foreign}    Obtain a Data Frame from a SAS Permanent
                          Dataset, via read.xport
read.systat {foreign}Obtain a Data Frame from a Systat File
read.xport {foreign}  Read a SAS XPORT Format Library
write.arff {foreign}  Write Data into ARFF Files
write.dbf {foreign}   Write a DBF File
write.dta {foreign}   Write Files in Stata Binary Format
write.foreign {foreign}   Write Text Files and Code to Read Them
```


Εισαγωγή δεδομένων: using the functions in the **foreign** package

```
> library(foreign)
>library(help=foreign)
>?read.dta   # To view the help file for a specific function

# reads data saved in .sav files for SPSS
> food <-
read.spss("http://www.methodsconsultants.com/data/food.
sav",
+ to.data.frame=TRUE)

# open up a dialog box, to navigate to the folder where
the desired file resides
> food<-read.spss(file.choose(), to.data.frame=FALSE)

# input Stata file
mydata <- read.dta("c:/mydata.dta")
# input Systat file
mydata <- read.systat("c:/mydata.dta")
```

Πίνακες δεδομένων από ιστοσελίδες

```
# Use 'htmltab' package: Assemble Data Frames from HTML Tables
#     htmltab() recognizes spans and expands tables automatically
htmltab(doc, which = NULL, header = NULL,
  headerFun = function(node) XML::xmlValue(node), headerSep = " >> ",
  body = NULL, bodyFun = function(node) XML::xmlValue(node),
  complementary = TRUE, fillNA = NA, rm_superscript = TRUE,
  rm_escape = " ", rm_footnotes = TRUE, rm_nodata_cols = TRUE,
  rm_nodata_rows = TRUE, rm_invisible = TRUE, rm_whitespace = TRUE,
  colNames = NULL, ...)

library(htmltab)
url <- "http://www.someweb.com/.../some_Wikipedia.html"
ukLang <- htmltab(doc = url, which = "//th[text() = 'Ability']/ancestor:
head(ukLang)
```


Ανάγνωση δεδομένων από βάσεις SQL

- Database interface (DBI) που διαχωρίζει τη συνδεσιμότητα με το σύστημα διαχείρισης DBMS σε "front-end" και "back-end"
  - R πακέτα *DBI*, *ODBC* (Open Database Connectivity) με βάσεις SQL Server, Oracle, MySQL, PostgreSQL, SQLite, …
- Πακέτο *dplyr* κυρίως για τη χρήση όγκου δεδομένων που δεν χωρούν όλα στη μνήμη ταυτόχρονα
  - Ο ρόλος του dplyr είναι να δημιουργεί εντολές SQL και να ζητά τα αποτελέσματα από τη βάση δεδομένων
- Πακέτο *pool* για την αποδοτική διαχείριση συνδέσεων με μια βάση δεδομένων, όχι απευθείας αλλά μέσω αντικειμένων (pools) με αναφορά σε αυτήν.

Σε επόμενη ενότητα του μαθήματος …

Εξαγωγή δεδομένων από το R, και
Διαχείριση πολυσύνθετων γεωδεδομένων