ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Γλώσσες Προγραμματισμού

Εισαγωγικές έννοιες - R

Χαρά Πρασσά, chprassa@gmail.com

Acknowledgments

- Οι διαφάνειες αντιστοιχούν στα 1ο και 2ο κεφάλαια του βιβλίου ‘Ανάλυση δεδομένων με την R. Νικολάου, Χ., Εκδόσεις Δίσιγμα, 2019’.

Χαρακτηριστικά της γλώσσας R

- Απλότητα
- Γρήγορη καμπύλη εκμάθησης
- Ευελιξία και πληρότητα

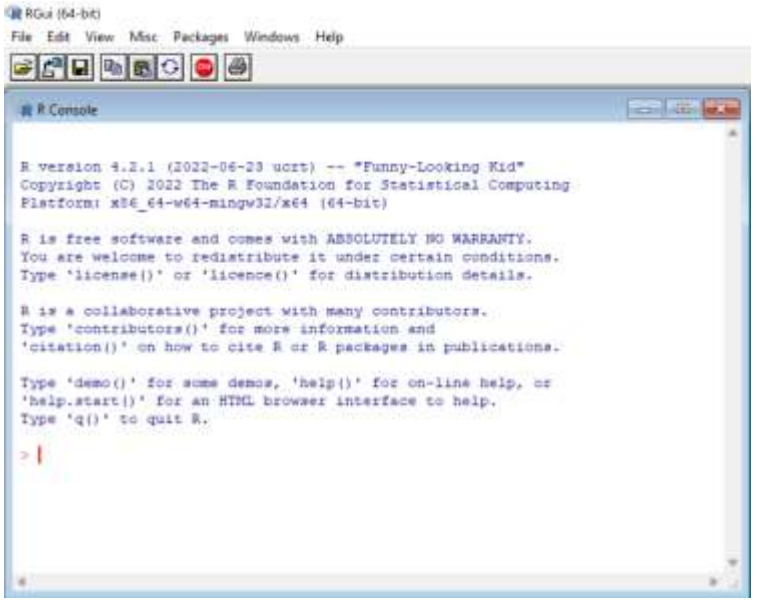
- Η R είναι ένα *υπολογιστικό πακέτο* που προσφέρει δυνατότητες διαχείρισης και *στατιστικής ανάλυσης δεδομένων* καθώς και δυνατότητες κατασκευής γραφημάτων.

- Βασίζεται στην *γλώσσα προγραμματισμού* S (που χρησιμοποιεί και το στατιστικό πακέτο S plus) και πρόκειται για *λογισμικό ανοικτού κώδικα* (open source) που διατίθεται ελεύθερα.

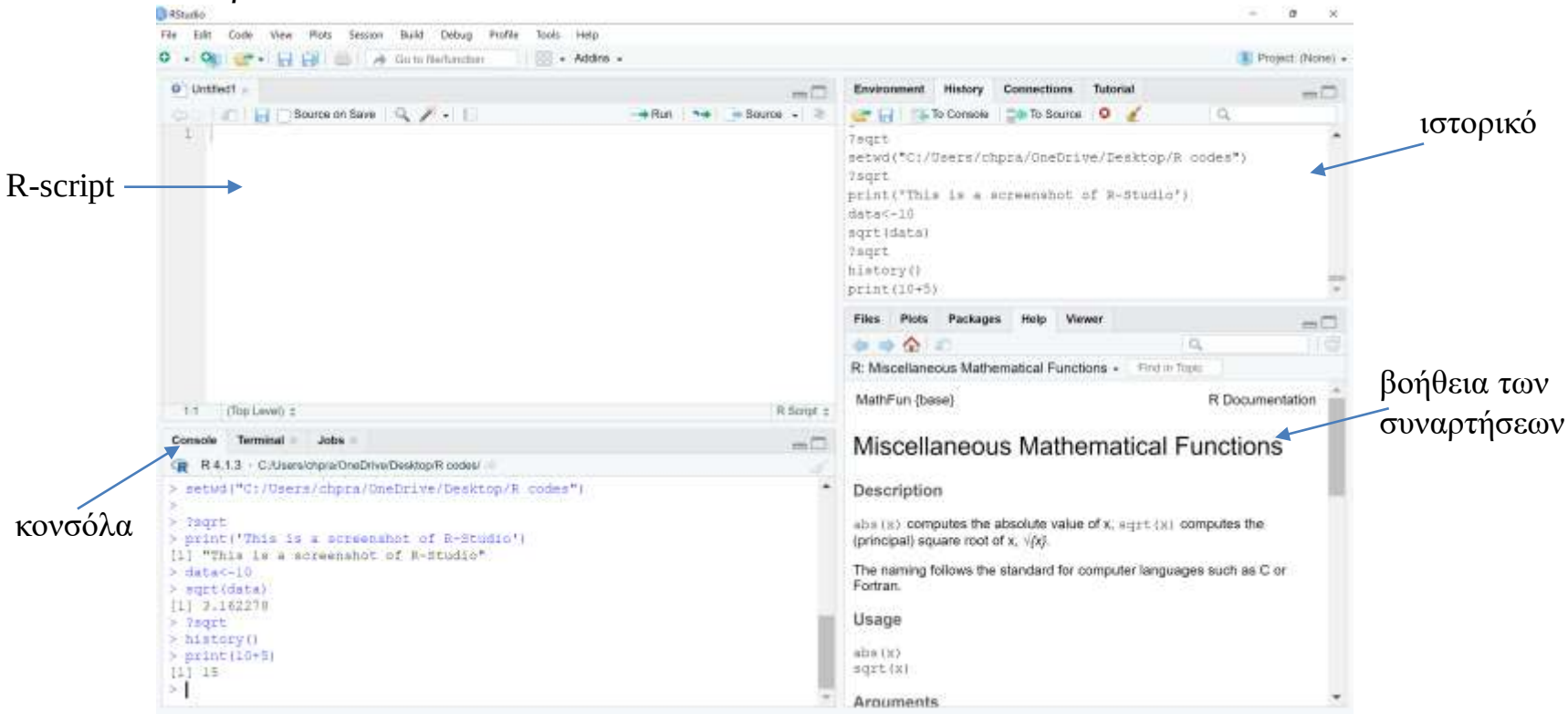
- Από πρακτικής άποψης η R διαφέρει από την S, καθώς είναι ταυτόχρονα μια γλώσσα και ένα περιβάλλον προγραμματισμού. Αυτό σημαίνει ότι κανείς μπορεί μεν να γράψει προγράμματα σε έναν επεξεργαστή κειμένου και να τα εκτελέσει αργότερα, ωστόσο υπάρχει και η δυνατότητα ο χρήστης να εργαστεί απευθείας στο περιβάλλον ή αλλιώς στην ‘κονσόλα’ η οποία δίνει τη δυνατότητα άμεσης αλληλεπίδρασης.

- Η R διατίθεται δωρεάν ενώ η εγκατάσταση της είναι αρκετά απλή και γίνεται μέσα από την ιστοσελίδα του R project (https://www.r-project.org/), με πρόσβαση σε κάποια από τα διαπιστευμένα αποθετήρια (mirrors) του Comprehensive R Archive Network (CRAN) μέσω της επιλογής 'Download R'. Στην Ελλάδα τα mirrors υποστηρίζονται από το Πανεπιστήμιο Κρήτης.
- Στη συνέχεια και ανάλογα με το λειτουργικό σύστημα ο χρήστης επιλέγει την κατάλληλη έκδοση και προχωράει σε εγκατάσταση.
- Η εκκίνηση της R γίνεται με διπλό κλικ στο αντίστοιχο εικονίδιο 

- Η βασική μορφή του interface (GUI) είναι 'παραθυρική' με ένα βασικό μενού στην κορυφή. Οι εντολές δίνονται μέσω της R Console : μετά το σύμβολο > γράφουμε την εντολή (ή το σύνολο των εντολών) που θέλουμε να εκτελεστούν και πατώντας ENTER λαμβάνουμε το αποτέλεσμα στην επόμενη γραμμή.



- Το πιο ευρέως χρησιμοποιούμενο περιβάλλον ανάπτυξης εφαρμογών (Integrated Development Environments, IDE) της R, είναι το R-Studio (https://www.rstudio.com/) το οποίο επιτρέπει στο χρήστη να συντάσσει κώδικα, να κρατάει έλεγχο των μεταβλητών και των συναρτήσεων που δημιουργεί, να αποθηκεύει γραφικά και να ανατρέχει στο ιστορικό εντολών με έναν εποπτικό τρόπο.



- Για παράδειγμα, μπορεί κανείς απλά να τυπώσει έναν αριθμό:

```
> 12
[1] 12
```

για να τον δει να εμφανίζεται ως αποτέλεσμα. Αντίστοιχα μπορεί να τυπώσει μια σειρά χαρακτήρων μέσα σε διπλά ή μονά εισαγωγικά:

```
> 'my first R string'
[1] "my first R string"
```

Επίσης, μπορούμε να κάνουμε μια αριθμητική πράξη στην R, χρησιμοποιώντας τα γνωστά σύμβολα πατώντας enter μετά το τέλος της παράστασης:

```
> 4+12
[1] 16
```

- Η R βασίζεται σε μια πολύ μεγάλη γκάμα συναρτήσεων που μπορούν να εκτελέσουν από τις πιο απλές έως τις πιο σύνθετες διαδικασίες. Για να λάβουμε βοήθεια για τον τρόπο σύνταξης και λειτουργίας μιας συνάρτησης, μπορούμε να γράψουμε ένα λατινικό ερωτηματικό πριν από το όνομα της. Για παράδειγμα για να λάβουμε πληροφορίες για τη συνάρτηση sqrt η οποία υπολογίζει την τετραγωνική ρίζα ενός αριθμού γράφουμε ?sqrt

```
> ?sqrt
```

Η παραπάνω εντολή θα ανοίξει ένα νέο παράθυρο με τα παρακάτω στοιχεία:

MathFun {base} R Documentation

Miscellaneous Mathematical Functions

Description

`abs(x)` computes the absolute value of x, `sqrt(x)` computes the (principal) square root of x, $\sqrt{x}$.

The naming follows the standard for computer languages such as C or Fortran.

Usage

```
abs(x)
sqrt(x)
```

Arguments

x
a numeric or `complex` vector or array.

Details

These are internal generic primitive functions: methods can be defined for them individually or via the `Math` group generic. For complex arguments (and the default method), z, `abs(z) == Mod(z)` and `sqrt(z) == z^0.5`.

`abs(x)` returns an `integer` vector when x is `integer` or `logical`.

- Σημαντικό στοιχείο ώστε να εργαστεί κανείς απρόσκοπτα είναι η πλοήγηση στα αρχεία. Η εκκίνηση της R γίνεται πάντα στον φάκελο που είναι εγκατεστημένο το ίδιο το πρόγραμμα της R. Για να δούμε το φάκελο εργασίας πληκτρολογούμε:

```
> getwd()
[1] "C:/Users/chpra/OneDrive/Έγγραφα"
```

το οποίο θα επιστρέψει το πλήρες μονοπάτι (full path) του φακέλου. Αν θέλουμε να αλλάξουμε τον φάκελο εργασίας πληκτρολογούμε:

```
> setwd("C:/Users/chpra/OneDrive/Desktop/R codes")
```

η οποία θα αλλάξει τον φάκελο εργασίας σε αυτόν που αντιστοιχεί στο πλήρες μονοπάτι που δίνουμε μέσα σε εισαγωγικά (εννοείται ότι ο συγκεκριμένος φάκελος θα πρέπει να υπάρχει).

- Κάθε φορά που εκκινούμε την R ξεκινάμε και μια νέα ‘συνεδρία’ (session). Το σύνολο των εντολών του session αποθηκεύεται σε ένα αρχείο ιστορικού (history) το οποίο μπορούμε να ανακαλέσουμε με την εντολή:

```
> history()
```

Το ιστορικό ανακαλείται επίσης βήμα-βήμα με την χρήση του πλήκτρου για το πάνω βέλος που επιτρέπει στο χρήστη να ανακαλέσει προηγούμενες εντολές από την πιο πρόσφατη στην παλαιότερη.

- Για την έξοδο από το πρόγραμμα αναγράφουμε την εντολή

```
> q()
```

- Οι μεταβλητές εισάγονται με το όνομα τους χωρίς να χρειάζεται να εξειδικευτεί κάποια άλλη παράμετρος. Για παράδειγμα:

> x<-5

Θα δημιουργήσουμε μια μεταβλητή που το όνομα της είναι x και θα έχει την τιμή 5. Για να δούμε την τιμή μιας μεταβλητής αρκεί να δώσουμε το όνομα της στην κονσόλα ακολουθούμενο από enter.

> x

[1] 5

Ο αριθμός 1 μέσα στην αγκύλη δεν είναι μέρος της μεταβλητής αλλά είναι απλά δηλωτικός του μεγέθους της μεταβλητής που διαβάζουμε.

- **Εκχώρηση τιμής σε μεταβλητή:** Ο τελεστής απόδοσης τιμής -> δίνει στην R την εντολή να δώσει την τιμή που βρίσκεται από την πλευρά της παύλας στη μεταβλητή που βρίσκεται στην κορυφή του βέλους. Έτσι οι δύο παρακάτω εντολές

> x <- 5

> 5 ->x

είναι ισοδύναμες, ενώ η πράξη:

> x -> 5

οδηγεί σε σφάλμα καθώς δεν είναι δυνατόν να αποδοθεί τιμή σε μια σταθερά.

- Η R μπορεί να εκτελέσει ένα σύνολο εντολών, διαδοχικά, τη μία μετά την άλλη. Οι εντολές μπορεί να χωρίζονται με ";"

```
> x<- 5; print(x)
[1] 5
```

- Αν μια εντολή δεν είναι πλήρης, τότε στην επόμενη γραμμή η R περιμένει (εμφανίζοντας ένα "+") την συνέχεια (αυτό γίνεται στην R-console)

```
> print(x
+
+ )
[1] 5
```

- Μπορούμε να ομαδοποιήσουμε εντολές γράφοντάς τις μεταξύ άγκιστρων

```
> {x<-5;
+ print(x);
+ x^2}
[1] 5
[1] 25
```

- Όσα ακολουθούν τον χαρακτήρα "#" σε μία γραμμή θεωρούνται σχόλια και δεν λαμβάνονται υπόψη

```
> x<-5 #εκχώρηση τιμής 5 στο x
> #τυπώνουμε το αποτέλεσμα
> print(x)
[1] 5
```

- Για να δούμε ποια αντικείμενα (π.χ. μεταβλητές) έχουν οριστεί στο session που εργαζόμαστε εκτελούμε

```
objects()
[1] "x"
```

- Για να διαγράψουμε κάποια από αυτά εκτελούμε:

```
rm(x)
```

- Για να διαγράψουμε όλες τις μεταβλητές εκτελούμε:

```
rm(list = ls())
```

- Για να γίνει εκκαθάριση της R-Console πληκτρολογούμε Ctrl+L

- *Κανόνες ονομασίας μεταβλητών*

1. Ο πρώτος χαρακτήρας να είναι πάντοτε γράμμα (πεζό ή κεφαλαίο). Κάτω από προϋποθέσεις μπορεί να είναι και η τελεία αλλά καλό είναι να αποφεύγεται.
2. Να μην περιέχουν σύμβολα άλλα από την τελεία και το underscore.
3. Να μην ταυτίζονται με ονόματα αντικειμένων της R όπως συναρτήσεις ή ειδικές εντολές.

- Παραδείγματα αποδεκτών ονομάτων είναι:

Total, my_variable0, my.value, a1_number

- Παραδείγματα μη αποδεκτών ονομάτων είναι:

22val, _sum, TRUE

❑ *Βασικά είδη μεταβλητών:*

1. Αριθμοί (ακέραιοι, πραγματικοί δηλ. με δεκαδικό μέρος, μιγαδικοί)
2. Σειρές χαρακτήρων
3. Λογικές τιμές

- Μπορούμε να δούμε το είδος της μεταβλητής με χρήση της συνάρτησης class().

```
> an integer<-12
> a_real<-3.141
> a_complex<--5+2i
> a_character<-'100'
> a_logical <- TRUE
```

δίνονται σε μονά ή διπλά εισαγωγικά

TRUE, FALSE ή συντομογραφίες T, F

```
> class(an integer)
[1] "numeric"
> class(a_real)
[1] "numeric"
> class(a_complex)
[1] "complex"
> class(a_character)
[1] "character "
> class(a_logical)
[1] "logical"
```

- *Αριθμητικές πράξεις*

Σύμβολο	Πράξη
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Διαίρεση
**	Ύψωση σε δύναμη
%%	Υπόλοιπο διαίρεσης

Για τη σειρά εκτέλεσης των αριθμητικών πράξεων ισχύουν οι γνωστοί κανόνες προτεραιότητας, με τις δυνάμεις να προηγούνται του πολλαπλασιασμού/διαίρεσης και αυτών της πρόσθεσης/αφαίρεσης. Η R υποστηρίζει την χρήση παρενθέσεων, οι πράξεις εντός των οποίων προηγούνται.

Παράδειγμα υπολογισμού του εμβαδού κυλίνδρου διαμέτρου d και ύψους h:

```
> d<-2
> h<-2
> surface<- (3.14*d*h)+(2*3.14*(d/2)**2)
```

- *Λογικές πράξεις*

Σύμβολο	Πράξη
==	'Ισο/'Ομοιο
!=	Διάφορο
>	Μεγαλύτερο
<	Μικρότερο
>=	Μεγαλύτερο ή ίσο
<=	Μικρότερο ή ίσο

Προσοχή χρειάζεται το διπλό == για την ισότητα. Οι λογικές πράξεις έχουν αποτέλεσμα λογικές τιμές TRUE/FALSE.

```
> x<-6
> y<-8
> x>y
[1] FALSE

> x<-'one'
> y<-'one'
>x==y
[1] TRUE
```

οι πράξεις μεγαλύτερου/μικρότερου δεν έχουν νόημα σε σειρές χαρακτήρων

- Εκτός των έτοιμων συναρτήσεων που διαθέτει η R όπως για παράδειγμα η class() που έχουμε ήδη χρησιμοποιήσει, μας δίνεται η δυνατότητα να δημιουργήσουμε τη δική μας συνάρτηση ακολουθώντας την σύνταξη:

```
> testfunc<-function(x,n) {
+ ...
+ ...
+ }
```

- Το όνομα της συνάρτησης είναι προσωπική επιλογή του κάθε χρήστη. Μετά το όνομα ακολουθεί ο χαρακτήρας εκχώρησης <- και η εντολή function. Ακολουθούν μέσα σε παρένθεση οι μεταβλητές που θα χρησιμοποιηθούν μέσα στη συνάρτηση, δηλαδή οι μεταβλητές εισόδου. Ακολουθούν μέσα σε άγκιστρα οι εντολές που αποτελούν τη συνάρτηση.
- Παράδειγμα δημιουργίας συνάρτησης που υπολογίζει τη μηνιαία δόση δανείου

```
loan<- function(amount,rate,years){
  #η συνάρτηση loan υπολογίζει τη μηνιαία δόση του δανείου
  #μεταβλητές εισόδου:
  #amount: το κεφάλαιο του δανείου σε ευρώ
  #rate: το ετήσιο επιτόκιο σε μονάδες %
  #years: αριθμός των ετών του δανείου
  #μεταβλητή εξόδου:
  #mpay: μηνιαία δόση του δανείου
  ratem<- rate*(0.01/12)
  a=1+ratem
  mpay=amount*(ratem/(1-(a)**(-12*years)))
  return(mpay)
}
```

- Μπορούμε να εκτελέσουμε τη συνάρτηση loan καλώντας την με τρείς αριθμούς ως όρισμα, χωριζόμενους με κόμμα:

```
> loan(100000,8,20)
[1] 836.4401
```

Όπως και στη Matlab, οι μεταβλητές διακρίνονται σε τοπικές και καθολικές.

1. Οι τοπικές μεταβλητές ορίζονται μέσα στη συνάρτηση και ισχύουν μόνο μέσα στη συνάρτηση, ενώ οι τιμές τους παύουν να ισχύουν μετά.
2. Οι καθολικές μεταβλητές είναι μεταβλητές κάθε είδους και ισχύουν γενικά και όχι μόνο μέσα στη συνάρτηση που τις χρησιμοποιεί.

- ***Για παράδειγμα:***

```
> ratem
Error: object 'ratem' not found
```

Η μεταβλητή ratem είναι τοπική μεταβλητή καθώς έχει οριστεί μέσα στη συνάρτηση loan και παύει να έχει ισχύ εκτός αυτής, οπότε και δεν αναγνωρίζεται όταν την καλούμε.

- Η εντολή if επιτρέπει να εκτελέσουμε κάποια ή κάποιες εντολές, μόνο στην περίπτωση που μια συνθήκη ισχύει. Η γενική της σύνταξη έχει τη μορφή:

```
if (συνθήκη) εντολή
```

ή

```
if (συνθήκη) { εντολή #1; εντολή #2; . . . εντολή #n }
```

ή

```
if (συνθήκη) {
      εντολή #1
      εντολή #2
      ⋮
      εντολή #n
      }
```

Η λειτουργία της είναι πως πρώτα ελέγχεται η συνθήκη και αν είναι αληθής τότε εκτελείται η εντολή ή οι εντολές που ακολουθούν. Αν οι εντολές είναι περισσότερες από μια, τότε πρέπει να τις συμπεριλάβουμε μέσα σε άγκιστρα.

- Ο ακόλουθος κώδικας υπολογίζει το συνολικό κόστος μιας παραγγελίας βάσει των μονάδων πώλησης και της τιμής πώλησης. Εάν η παραγγελία αναφέρεται σε περισσότερες από 40 μονάδες τότε στην αρχική τιμή πώλησης παρέχεται έκπτωση 15%.

```
# Υπολογισμός του συνολικού κόστους παραγγελίας
# Εάν οι μονάδες πώλησης είναι πάνω από 40 η τιμή πώλησης
# είναι μειωμένη κατά 15%
# taking input using readline()
# this command will prompt you
# to input a desired value
q = readline('Πληκτρολογείστε τις μονάδες πώλησης: ')
p = readline('Πληκτρολογείστε την τιμή πώλησης: ')
q=as.numeric(q)
p=as.numeric(p)
c=q*p;
if (q>40){
c=c*0.85
}
print(c)
```

- Στις περιπτώσεις που δεν ισχύει η συνθήκη, αλλά θα θέλαμε να εκτελεστεί κάποια άλλη εντολή, τότε πρέπει να χρησιμοποιήσουμε την σύνταξη:

```
if (συνθήκη) { εντολές #1 } else { εντολές #2 }
```

- Η εντολή αυτή ελέγχει αν ισχύει μια συνθήκη. Αν ισχύει, εκτελείται η πρώτη ομάδα εντολών { εντολές #1}, αλλιώς εκτελείται η δεύτερη ομάδα εντολών {εντολές #2}.

Και πάλι, αν οι { εντολές #1} ή οι {εντολές #2} είναι περισσότερες από μια, πρέπει να τις εσωκλείσουμε μέσα σε άγκιστρα. Επίσης, η εντολή else πρέπει να είναι στην ίδια γραμμή που τελειώνει το κομμάτι της if, είτε αυτή είναι μια εντολή, είτε το άγκιστρο που δηλώνει το τέλος των εντολών της ομάδας { εντολές #1}, γιατί αλλιώς η R θα καταλάβει ότι πρόκειται για απλή if εντολή.

- Η δομή **if - else if - else** χρησιμοποιείται στην περίπτωση που υπάρχουν περισσότερες από δύο αμοιβαία αποκλειόμενες περιπτώσεις που καθορίζονται από την ισχύ μιας συνθήκης.

.

- Ο ακόλουθος κώδικας υπολογίζει το ποσό του φόρου βάσει της ακόλουθης κλίμακας φορολόγησης: 15% επί των πρώτων 10.000 ευρώ, 25% από 10.000-40.000 ευρώ και 40% για ποσά μεγαλύτερα από 40.000 ευρώ.

```
salary=readline('Πληκτρολογείστε το ετήσιο εισόδημα: ')
salary=as.numeric(salary)
taxhigh=0.4;
taxmid=0.25;
taxlow=0.15;

if (salary<=10000){
payment=taxlow*salary
} else if (salary<=40000){
payment=taxlow*10000+taxmid*(salary-10000)
}else{
  payment=taxlow*10000+taxmid*30000+taxhigh*(salary-40000)
}

print(payment)
```

- Η εντολή for όπως και η εντολή while, μας επιτρέπει να επαναλάβουμε κάποιες εντολές περισσότερες από μια φορές. Η βασική διαφορά τους είναι πως με την εντολή for ξέρουμε εκ των προτέρων και καθορίζουμε εμείς πόσες φορές θα εκτελεστεί, ενώ με την εντολή while πρέπει να συμπεριλάβουμε μια συνθήκη ελέγχου, η οποία όσο είναι αληθής, συνεχίζονται οι επαναλήψεις και αν δεν είναι αληθής, σταματάνε οι επαναλήψεις.

Η γενική σύνταξη της for είναι η ακόλουθη:

for (name in values) εντολές

Η μεταβλητή name θα πάρει όλες τις τιμές που έχουμε ορίσει στο σύνολο values. Συνήθως, με την εντολή for ενδιαφερόμαστε για διαδοχικές ακέραιες τιμές, οπότε το σύνολο αυτό έχει για παράδειγμα την τιμή 1:100.

```
for (i in 1:5){
  i<- i+10
  print(i)
}
```

```
[1] 11
[1] 12
[1] 13
[1] 14
[1] 15
```

- Η εντολή while είναι και αυτή μια επαναληπτική εντολή αλλά ο αριθμός των επαναλήψεων δεν είναι προκαθορισμένος και εξαρτάται από κάποια συνθήκη ελέγχου. Η γενική σύνταξη είναι η ακόλουθη:

while (συνθήκη) εντολές

Παράδειγμα:

```
i<-1
while (i<10){
  print(i)
  i<-i+1
}
```

αν δεν υπήρχε, ο βρόγχος δεν θα σταματήσει ποτέ να εκτελείται (ατέρμονας)

(Esc για να σταματήσει η εκτέλεση της while)

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

- Οι επαναληπτικές εντολές είναι ιδιαίτερα αργές όταν υλοποιούνται στην R. Ένας τρόπος για να τις κάνουμε πιο γρήγορες, είναι με τη χρήση της εντολής apply. Η εντολή αυτή έχει τη γενική σύνταξη:

apply (X, MARGIN, FUN)

όπου X είναι κάποιο αντικείμενο, συνήθως πίνακας ή array, MARGIN είναι η μεταβλητή που μας δείχνει ως προς ποια διάσταση του αντικειμένου θα επαναλάβουμε τη διαδικασία. Αν το X είναι πίνακας (matrix), τότε το 1 αντιστοιχεί στις γραμμές και το 2 στις στήλες. Δηλαδή, αν η παράμετρος MARGIN πάρει την τιμή 1, λέμε στην R να επαναλάβει τη συνάρτηση ως προς όλες τις γραμμές (π.χ. να υπολογίσει το μέσο κάθε γραμμής ξεχωριστά).

Η παράμετρος FUN είναι η συνάρτηση την οποία θα επαναλάβουμε και η οποία μπορεί να είναι, είτε μια από τις συναρτήσεις της R (π.χ. mean, max κλπ.), είτε κάποια συνάρτηση που έχει ορίσει ο χρήστης.

```
> testmatrix <- matrix(1:12,3,4)
> testmatrix
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
> apply(testmatrix, 1, mean)
[1] 5.5 6.5 7.5
> apply(testmatrix, 2, mean)
[1]  2  5  8 11
```

Η χρήση της εντολής apply προσφέρει πολύ μεγάλη ταχύτητα και όταν είναι εφικτό θα πρέπει να την προτιμάμε από την εντολή for, ή όποια άλλη επαναληπτική εντολή

- Η χρήση απλών βαθμωτών μεταβλητών στην R, δεν παρουσιάζει ιδιαίτερο ενδιαφέρον. Οι δυνατότητες που μας προσφέρει η R φαίνονται στην χρήση πιο πολύπλοκων δομών δεδομένων.

Έτοιμα (built-in) δεδομένα

Για πρακτικούς λόγους η R περιέχει προεγκατεστημένα σύνολα δεδομένων. Κάποια από αυτά είναι μέρος της βασικής έκδοσης της R και κάποια άλλα ανήκουν σε ειδικές βιβλιοθήκες συναρτήσεων. Προκειμένου να δούμε ένα σετ δεδομένων στην κονσόλα δεν έχουμε παρά να τυπώσουμε το όνομα της μεταβλητής στην οποία περιέχεται:

```
> mtcars
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

Η mtcars είναι ένα σετ δεδομένων (dataframe) το οποίο περιέχει τεχνικές προδιαγραφές και χαρακτηριστικά για μια σειρά από μοντέλα αυτοκινήτων.

- Η dim() δίνει τις διαστάσεις μιας μεταβλητής που στην προκειμένη περίπτωση είναι m x n (γραμμές x στήλες)

```
> dim(mtcars)
[1] 32 11
```

- Οι επιμέρους διαστάσεις δίνονται από τις συναρτήσεις nrow() και ncol()

```
> nrow(mtcars)
[1] 32
> ncol(mtcars)
[1] 11
```

- Η head() επιστρέφει τις πρώτες γραμμές της μεταβλητής (με προεπιλεγμένη τιμή τις 6 γραμμές). Αντίστοιχα η tail() επιστρέφει τις τελευταίες γραμμές. Αν κανείς θέλει να δει ένα συγκεκριμένο αριθμό γραμμών θα πρέπει να προσθέσει μια παράμετρο ακόμα στην εκτέλεση της συνάρτησης

```
> head(mtcars)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

```
> tail(mtcars,n=2)
                mpg cyl disp  hp drat   wt qsec vs am gear carb
Maserati Bora 15.0   8  301 335 3.54 3.57 14.6  0  1    5    8
Volvo 142E    21.4   4  121 109 4.11 2.78 18.6  1  1    4    2
```

- Η str() δίνει τη δομή της μεταβλητής με πολύ πιο αναλυτικό τρόπο από την dim(). Συγκεκριμένα δίνει αρχικά τον τύπο της μεταβλητής, ο οποίος είναι dataframe δηλαδή πίνακας και οι διαστάσεις του είναι 32 x 11 (32 παρατηρήσεις και 11 μεταβλητές). Εδώ όλες οι μεταβλητές είναι numeric.

```
> str(mtcars)
'data.frame':   32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
 $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
 $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

- Μπορούμε να πάρουμε τα ονόματα των συνιστωσών μεταβλητών του αντικειμένου mtcars με τη συνάρτηση names()

```
> names(mtcars)
```

```
 [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
[11] "carb"
```

Ένα παράδειγμα της str() σε ένα άλλο προεγκατεστημένο σετ δεδομένων φαίνεται εδώ για το dataset iris:

```
> str(iris)
```

```
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1
 1 1 ...
```

Το οποίο βλέπουμε ότι είναι ένας 150 x 5 πίνακας με 4 αριθμητικές μεταβλητές (num) και μια μεταβλητή που ανήκει στο είδος factor (που θα δούμε στη συνέχεια). Και από τα δύο παραδείγματα καταλαβαίνουμε ότι μια σύνθετη μεταβλητή μπορεί να αποτελείται από επιμέρους μεταβλητές με ξεχωριστά ονόματα.

- Η R υποστηρίζει τη διαδικασία ανάγνωσης αρχείων και οι κυριότερες συναρτήσεις ανάγνωσης είναι οι ακόλουθες:

Συνάρτηση	Λειτουργία
read.csv()	Ανάγνωση πίνακα με διαχωριστή κόμμα
read.delim()	Ανάγνωση πίνακα με δήλωση διαχωριστή
readLines()	Ανάγνωση κειμένου ανά γραμμή

- Οι δύο πρώτες από τις παραπάνω συναρτήσεις δημιουργούν πίνακες του τύπου dataframe, ενώ η τελευταία δημιουργεί μια απλούστερη λίστα διαβάζοντας μια-μια τις γραμμές του αρχείου κειμένου.
- Η read.csv() διαβάζει δεδομένα που είναι διαχωρισμένα σε στήλες με κόμμα μεταξύ τους (csv) και η read.delim() επιτρέπει στο χρήστη να επιλέξει ο ίδιος τον διαχωριστή αρχείου που έχουν τα δεδομένα του. Ως διαχωριστές χρησιμοποιούνται συνήθως το κενό " ", το κόμμα "," ή το tab " \t ". Η γενική σύνταξη των παραπάνω συναρτήσεων είναι η εξής:

readfunction("nameoffile", header=T/F, sep= "character")

- Στο nameoffile δίνεται το όνομα του αρχείου μαζί με την κατάληξη του μέσα σε εισαγωγικά. Ανάλογα με τον φάκελο που δουλεύουμε ενδέχεται να πρέπει να δηλώσουμε όχι μόνο το όνομα του αρχείου αλλά και το πλήρες μονοπάτι.
- Στο header=LOGICAL δίνεται μια λογική τιμή TRUE/FALSE ή πιο σύντομα T/F η οποία καθορίζει αν η πρώτη γραμμή του αρχείου θα χρησιμοποιηθεί ως επικεφαλίδα ή όχι.
- Στο sep= "character" δίνεται ως τιμή ένας χαρακτήρας που θα χρησιμοποιηθεί ως διαχωριστής.

- Με την παρακάτω εντολή διαβάζουμε τα δεδομένα του αρχείου eurodollar1.txt, το οποίο είναι tab separated, σε μια μεταβλητή που ονομάζουμε data1. Για να ελέγξουμε τη δομή δεδομένων και να επιβεβαιώσουμε ότι διαβάστηκαν σωστά, μπορούμε να καλέσουμε την str():

```
> data1 <- read.delim('eurodollar1.txt',header=F,sep="\t")
> str(data1)
'data.frame':   27 obs. of  9 variables:
 $ V1: num  4.5 5 5.25 5.5 6 0 0 0 0 0 ...
 $ V2: num  0.88 0.43 0.235 0.095 0.005 0 0 0 0 0 ...
 $ V3: num  5.37 5.37 5.37 5.37 5.37 5.37 5.37 5.37 5.37 5.37 ...
 $ V4: num  4.25 4.5 4.75 5 5.5 5.75 6 6.25 6.5 6.75 ...
 $ V5: num  0.005 0.02 0.035 0.065 0.225 0.4 0.63 0.88 1.13 1.38 ...
 $ V6: int  6 6 6 6 6 6 6 6 6 6 ...
 $ V7: int  14 14 14 14 14 14 14 14 14 14 ...
 $ V8: num  0.288 0.288 0.288 0.288 0.288 ...
 $ V9: num  4.78 4.78 4.78 4.78 4.78 4.78 4.78 4.78 4.78 4.78 ...
```

- Αντίστοιχα, μπορούμε να χρησιμοποιήσουμε την ίδια εντολή για να διαβάσουμε τα δεδομένα του αρχείου eurodollar2.txt τα οποία είναι διαχωρισμένα με κόμμα.

```
> data2 <- read.delim('eurodollar2.txt',header=F,sep=",")
> str(data2)
'data.frame':   10 obs. of  9 variables:
 $ V1: num  4.5 5 5.25 5.5 5.75 6 0 0 0 0
 $ V2: num  0.88 0.43 0.235 0.095 0.025 0.005 0 0 0 0
 $ V3: num  5.37 5.37 5.37 5.37 5.37 5.37 5.37 5.37 5.37 5.37
 $ V4: num  4.25 4.5 4.75 5 5.25 5.5 5.75 6 6.25 6.5
 $ V5: num  0.005 0.02 0.035 0.065 0.115 0.225 0.4 0.63 0.88 1.13
 $ V6: int  6 6 6 6 6 6 6 6 6 6
 $ V7: int  14 14 14 14 14 14 14 14 14 14
 $ V8: num  0.288 0.288 0.288 0.288 0.288 ...
 $ V9: num  4.78 4.78 4.78 4.78 4.78 4.78 4.78 4.78 4.78 4.78
```

- Ισοδύναμος τρόπος ανάγνωσης δεδομένων με διαχωριστή κόμμα από το αρχείο eurodollar2.txt μέσω της read.csv()

```
> data2 <- read.csv('eurodollar2.txt',header=F)
> str(data2)
'data.frame':   10 obs. of  9 variables:
 $ V1: num  4.5 5 5.25 5.5 5.75 6 0 0 0 0
 $ V2: num  0.88 0.43 0.235 0.095 0.025 0.005 0 0 0 0
 $ V3: num  5.37 5.37 5.37 5.37 5.37 5.37 5.37 5.37 5.37 5.37
 $ V4: num  4.25 4.5 4.75 5 5.25 5.5 5.75 6 6.25 6.5
 $ V5: num  0.005 0.02 0.035 0.065 0.115 0.225 0.4 0.63 0.88 1.13
 $ V6: int  6 6 6 6 6 6 6 6 6 6
 $ V7: int  14 14 14 14 14 14 14 14 14 14
 $ V8: num  0.288 0.288 0.288 0.288 0.288 ...
 $ V9: num  4.78 4.78 4.78 4.78 4.78 4.78 4.78 4.78 4.78 4.78
```

- Στην περίπτωση που τα δεδομένα που θέλουμε να διαβάσουμε δεν είναι μορφής πίνακα αλλά πρόκειται για αρχεία κειμένου μπορούμε να χρησιμοποιήσουμε τη συνάρτηση readLines().

```
> datatext<- readLines('eurodollar3.txt')
> str(datatext)
 chr [1:3] "This is the first line" "This is the second line" ...
```

Η μεταβλητή datatext είναι μια λίστα από σειρές χαρακτήρων που αντιστοιχούν στις γραμμές του αρχείου. Το αρχείο eurodollar3.txt έχει 3 γραμμές, με το περιεχόμενο της καθεμιάς να αποθηκεύεται στην αντίστοιχη θέση της λίστας. Η datatext είναι μια λίστα 3 στοιχείων όπως φαίνεται από την αγκύλη [1:3]. Το είδος της μεταβλητής είναι σειρά χαρακτήρων όπως φαίνεται από τη συντομογραφία chr.

- Το πακέτο **readxl** το οποίο αναπτύχθηκε από τον Hadley Wickham, μπορεί να χρησιμοποιηθεί για την εισαγωγή Excel αρχείων (xls ή xlsx) στην R.

➢ Εγκατάσταση πακέτου readxl

```
install.packages("readxl")
```

➢ Εισαγωγή βιβλιοθήκης readxl

```
library("readxl")
```

```
> my_data <- read_excel("EquityIndices.xlsx",sheet=1)
> str(my_data)
tibble [61 x 15] (S3: tbl_df/tbl/data.frame)
 $ Date          : POSIXct[1:61], format: "2022-08-08" "2022-08-09" ...
 $ FTSE100       : num [1:61] 6045 6527 6469 6544 6280 ...
 $ FTSE100_PO    : num [1:61] 6233 6490 6471 6518 6404 ...
 $ FTSE100_PH    : num [1:61] 6238 6536 6491 6552 6425 ...
 $ FTSE100_PL    : num [1:61] 6039 6490 6441 6511 6260 ...
 $ FRCAC40       : num [1:61] 6196 6555 6564 6594 6405 ...
 $ FRCAC40_PO    : num [1:61] 6425 6570 6499 6640 6570 ...
 $ FRCAC40_PH    : num [1:61] 6428 6592 6567 6650 6570 ...
 $ FRCAC40_PL    : num [1:61] 6187 6536 6484 6588 6396 ...
 $ DAXINDX       : num [1:61] 6989 7227 7480 7232 6765 ...
 $ DAXINDX_PO    : num [1:61] 7182 7208 7365 7286 6953 ...
 $ DAXINDX_PH    : num [1:61] 7182 7270 7492 7321 6973 ...
 $ DAXINDX_PL    : num [1:61] 6959 7202 7304 7217 6726 ...
 $ Return FTSE100: num [1:61] NA 0.07664 -0.00891 0.01147 -0.04115 ...
 $ Return FRCAC40: num [1:61] NA 0.0563 0.00135 0.00465 -0.02907 ...
```

Το όρισμα sheet είναι προαιρετικό. Μπορούμε να δηλώσουμε είτε τον αριθμό είτε το όνομα του φύλλου εργασίας μέσα σε " ".

Ορισμός φακέλου εργασίας και πλήρους μονοπατιού

- Στα παραδείγματα που είδαμε τα αρχεία που διαβάσαμε βρίσκονται στον φάκελο εργασίας. Για να δούμε ποιος είναι ο φάκελος εργασίας εκτελούμε την:

```
> getwd()
[1] "C:/Users/chpra/OneDrive/Desktop"
```

Στην περίπτωση που τα αρχεία που θέλουμε να διαβάσουμε δεν βρίσκονται σε αυτό το φάκελο οι επιλογές που έχουμε είναι δύο:

1. Να αλλάξουμε το φάκελο εργασίας, με τη συνάρτηση setwd():

```
> setwd("C:/Users/chpra/OneDrive/Desktop/R_codes")
> getwd()
[1] "C:/Users/chpra/OneDrive/Desktop/R_codes"
```

2. Να δώσουμε το πλήρες μονοπάτι του αρχείου ακολουθώντας την ιεραρχία των φακέλων:

```
> data1 <- read.delim("C:/Users/chpra/OneDrive/Desktop/R_codes/eurodollar1.txt",header=F,sep="\t")
```

Εγγραφή αποτελεσμάτων σε αρχεία

- Με συμμετρικό τρόπο, η R επιτρέπει εκτός από την εισαγωγή και την εξαγωγή δεδομένων σε αρχεία στον υπολογιστή, με μια σειρά από συναρτήσεις τύπου write πχ. write(), writeLines(), write.csv(), ωστόσο σε ό,τι αφορά δεδομένα η καλύτερη επιλογή είναι η write.table(). Μια ενδεικτική σύνταξη της είναι:

```
> setwd("C:/Users/chpra/OneDrive/Desktop/R_codes")

> write.table(data1, file="out.txt",sep=",",row.names=F,col.names=T,quote=F)
```

η οποία χρησιμοποιείται σε ένα πίνακα που έχει ήδη τιμές που κατανέμονται σε σειρές και στήλες. Η παράμετρος row.names=F λέει στην R να παραλείψει να απαριθμήσει τις σειρές, καθώς συνήθως προσθέτει και επιπλέον στήλη στο αρχείο εξόδου, ενώ η quote=F ορίζει ότι οι σειρές χαρακτήρων θα τυπωθούν "γυμνές" στο αρχείο δηλαδή χωρίς να εσωκλείονται σε εισαγωγικά.

- Η συνάρτηση write_xlsx του πακέτου writexl, μπορεί να χρησιμοποιηθεί για την εξαγωγή δεδομένων σε αρχεία Excel (xls ή xlsx) στην R.

```
>  install.packages("writexl")  #εγκατάσταση πακέτου writexl
>  library("writexl")              #εισαγωγή βιβλιοθήκης writexl
>  write_xlsx(data1, "data1.xlsx",col_names = TRUE,format_headers = TRUE)
```

- col_names = TRUE (εμφάνιση ονομάτων των στηλών)
- format_headers=TRUE (τα ονόματα των στηλών στο κέντρο και με έντονη γραφή)

- Η R επιτρέπει την εισαγωγή και εκτέλεση εντολών από αρχεία που είναι αποθηκευμένα στον υπολογιστή μας με τη χρήση της συνάρτησης source(). Το όρισμα της source() είναι ένα αρχείο που περιέχει αποκλειστικά εντολές στην R οι οποίες εισάγονται στην κονσόλα ως εξής:

```
> source("Rcommands.R")
```

Στο συγκεκριμένο παράδειγμα το αρχείο Rcommands.R περιέχει μια σειρά από εντολές οι οποίες με την κλήση της source θα εκτελεστούν αυτόματα ως μέρος της συνεδρίας.