ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ

ΣΧΟΛΗ ΕΠΑΓΓΕΛΜΑΤΩΝ ΥΓΕΙΑΣ ΚΑΙ ΠΡΟΝΟΙΑΣ

ΤΜΗΜΑ ΟΠΤΙΚΗΣ & ΟΠΤΟΜΕΤΡΙΑΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Χρήση λογισμικού R για στατιστική ανάλυση βιοιατρικών δεδομένων

Σπουδάστρια:

Γάγαλη Ανδριανή

Επιβλέπων Καθηγητής:

κ. Δροσόπουλος Αναστάσιος

Αίγιο- 2015

ΠΡΟΛΟΓΟΣ- ΕΥΧΑΡΙΣΤΙΕΣ

Ευχαριστώ το Ανώτατο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Δυτικής Ελλάδος, Τμήμα Οπτικής και Οπτομετρίας, για της πολύτιμες και εξειδικευμένες γνώσεις που μου έδωσε καθώς και την ώθηση για ανάπτυξη προσωπικών ικανοτήτων, την αναγκαιότητα της κριτικής σκέψης και πρωτοβουλιών μου.

Η παρούσα πτυχιακή εργασία διεύρυνε σε μεγάλο βαθμό τους γνωστικούς μου ορίζοντες στον χώρο των ηλεκτρονικών υπολογιστών, ενώ στον χώρο της επιστήμης παραθέτει ένα ολοκληρωμένο ενημερωτικό υλικό στον κλάδο της υγείας και πρόνοιας.

Η εργασία περιλαμβάνει τη σωστή χρήση του λογισμικού R και τις δυνατότητές της. Στόχος είναι οι περαιτέρω γνώσεις στα ηλεκτρονικά προγράμματα που μας βοηθούν σε απαντήσεις ερωτημάτων, πειράματα, μετρήσεις και στατιστικές αναλύσεις.

Θα ήθελα να ευχαριστήσω ιδιαίτερα τον επιβλέπων Καθηγητή της πτυχιακής μου, κ. Δροσόπουλο Αναστάσιο, για τη καθοδήγηση και την παροχή πληροφοριών καθ' όλη τη διάρκεια της διπλωματικής εργασίας.

ΠΕΡΙΛΗΨΗ

Η πτυχιακή εργασία αφορά ένα πρόγραμμα ηλεκτρονικού υπολογιστή και συγκεκριμένα το λογισμικό R. Βασίζεται σε μία γλώσσα προγραμματισμού όπου μπορεί κανείς να κάνει στατιστικούς υπολογισμούς, γραφήματα καθώς και ανάλυση δεδομένων.

Το πρώτο κεφάλαιο παρουσιάζει τη σωστή λήψη από το διαδίκτυο και εγκατάσταση των προγραμμάτων R, rstudio και git. Παρουσιάζονται αναλυτικά οι ηλεκτρονικοί ιστότοποι για το κάθε ένα και η διαδικασία εγκατάστασης.

Το δεύτερο κεφάλαιο αναφέρεται στις δυνατότητες του προγράμματος και πως μπορούμε να το χρησιμοποιήσουμε. Περιγράφονται οι βασικές δομές της γλώσσας και παρουσιάζονται διάφορα παραδείγματα χρήσης.

Στο τρίτο κεφάλαιο γίνεται ανάλυση πραγματικών δεδομένων από τον δικτυακό τόπο medicare.gov. Τα δεδομένα αναφέρονται σε πολλούς δείκτες ποιότητος από 4706 νοσοκομεία των ΗΠΑ . Η ανάλυσή μας εστιάζει στην θνησιμότητα 30 ημερών από καρδιακή προσβολή ή πνευμονία, δηλαδή ποια είναι η θνησιμότητα ασθενών που εισήχθησαν σε νοσοκομείο μέσα σε 30 ημέρες από την ημέρα εισαγωγής. Προφανώς εάν η τιμή είναι μηδενική ο ασθενής επέζησε. Τα νοσοκομεία που έχουν μικρότερες τιμές αξιολογούνται ως καλύτερης ποιότητας στην παροχή φροντίδας.

Το λογισμικό R χρησιμοποιείται σήμερα σε πολλούς κλάδους επιστημών όπως υγείας, τεχνολογίας ακόμα και οικονομικών για την εις βάθος αυτοματοποιημένη ανάλυση πραγματικών δεδομένων και εξαγωγή σωστών συμπερασμάτων για περαιτέρω αποφάσεις και ενέργειες.

SUMMARY

The thesis deals with a computer program/application, specifically the R software. This is based on a programming language where one can make statistical calculations, graphs and data analysis.

The first chapter outlines the correct download from the Internet and installation of the programs R, rstudio and git. The corresponding electronic sites for each program/application are presented as well as the installation process. The second chapter refers to the potential of the program and how we can use it. The basic structures of the language are described and a number of usage examples are presented.

The third chapter describes an analysis scenario of real data from the website medicare.gov. The data refer to many quality indices from 4706 US hospitals. Our analysis focuses on the 30-day mortality rate from heart attack or pneumonia, i.e. what is the mortality of patients admitted to hospital within 30 days from the date of admission. Obviously if the index value is zero, the patient survived. The hospitals that have smaller mortality rate values are evaluated as being of better quality in the care provided.

The R software is currently used in many sectors such as health sciences, technology and even finance for an in-depth automated analysis of real data and the generation of correct conclusions for further decisions and actions.

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ

Η R είναι μια γλώσσα προγραμματισμού και συγχρόνως περιβάλλον ανάπτυξης λογισμικού με έμφαση στατιστικούς υπολογισμούς και γραφικά. Χρησιμοποιείται ευρέως στην ανάπτυξη στατιστικού λογισμικού και ανάλυση δεδομένων. Δημιουργήθηκε το 1991 στο Πανεπιστήμιο του Όκλαντ στη Νέα Ζηλανδία από τους Ross Ihaka και Robert Gentleman και τώρα αναπτύσσεται από το R Development Core Team. Βασίστηκε στην προγενέστερη γλώσσα S με σημασιολογικές βελτιώσεις από τη γλώσσα Scheme. Είναι γλώσσα ανοικτού λογισμικού και ο πηγαίος κώδικας είναι δωρεάν διαθέσιμος στο χρήστη (GNU General Public License).

Η αλληλεπίδραση με τη γλώσσα R γίνεται μέσω διεπαφής εντολών (command line interface) όπου δίδονται διάφορες εντολές στην κονσόλα της R, το πρόγραμμα επεξεργάζεται άμεσα τις εντολές και δίνει απάντηση στην κάθε μια διαδοχικά. Υπάρχουν και γραφικά front-end που προσθέτουν επιπλέον δυνατότητες.

Το λογισμικό εφαρμόζει μια ευρεία ποικιλία στατιστικών και γραφικών μεθόδων, όπου συμπεριλαμβάνει γραμμικά και μη γραμμικά μοντέλα, ανάλυση χρονοσειρών, κλασικούς στατιστικούς ελέγχους, ομαδοποίηση και άλλα. Η βασική γλώσσα R είναι επεκτάσιμη και εύκολα μπορούν να αναπτυχθούν εξειδικευμένα πακέτα (στην ουσία βιβλιοθήκες λογισμικού με εξειδικευμένες συναρτήσεις και δεδομένα). Στον βασικό ιστότοπο http://www.r-project.org/ υπάρχουν αυτή τη στιγμή πάνω από 4000 τέτοια διαθέσιμα πακέτα τα οποία μπορεί να χρησιμοποιήσει κανείς στις δικές του αναλύσεις. Το front-end Rstudio που θα χρησιμοποιήσουμε στα παρακάτω προσθέτει επιπλέον δυνατότητες στην ανάπτυξη λογισμικού με R καθώς και τεκμηρίωσης με δημιουργία αρχείων που συνδυάζουν κείμενο, κώδικα και αποτελέσματα με γραφικές. Τα αρχεία αυτά είναι γραμμένα σε R markdown language με άμεση μετατροπή σε html ή άλλες μορφές και βοηθούν στην αποτελεσματική διάδοση τεκμηριωμένων αποτελεσμάτων (μέσω π.χ. του Github).

Κεφάλαιο 1: The Data Scientist's Toolbox

1.1 Τι είναι Data Science

Data Science - Επιστήμη Δεδομένων. Είναι η επιστήμη με στόχο την σωστή απάντηση ερωτημάτων βάσει ανάλυσης δεδομένων. Η διαδικασία εύρεσης της σωστής απάντησης είναι:

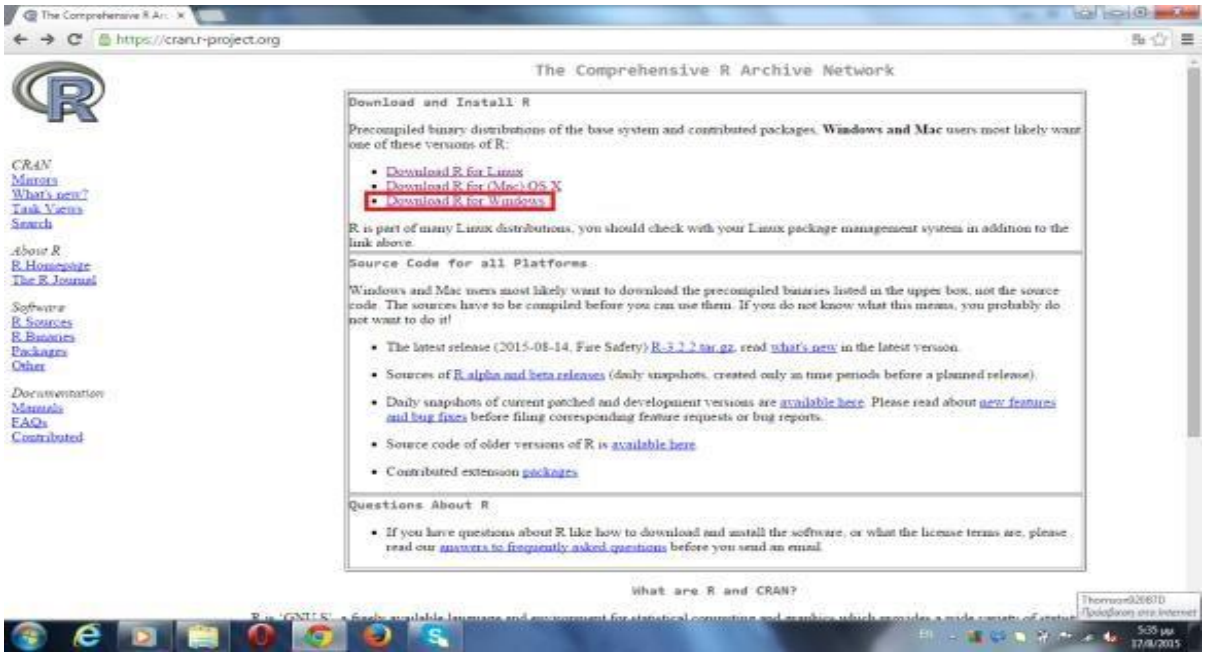
1. Προσδιορισμός ερωτήματος ή ερωτημάτων.
2. Προσδιορισμός των δεδομένων που είναι αναγκαία για να απαντηθούν τα ερωτήματα.
3. Απόκτηση των δεδομένων. Πειράματα. Μετρήσεις. Άλλες πηγές. Καθάρισμα δεδομένων. Εξερεύνηση.
4. Στατιστική ανάλυση - μοντελοποίηση. Κριτική επεξήγηση και αιτιολόγηση αποτελεσμάτων.
5. Γράψιμο αναφοράς όλων των προηγουμένων προσβάσιμη σε άλλους.

1.2 R και rstudio - Εγκατάσταση σε λειτουργικό Windows

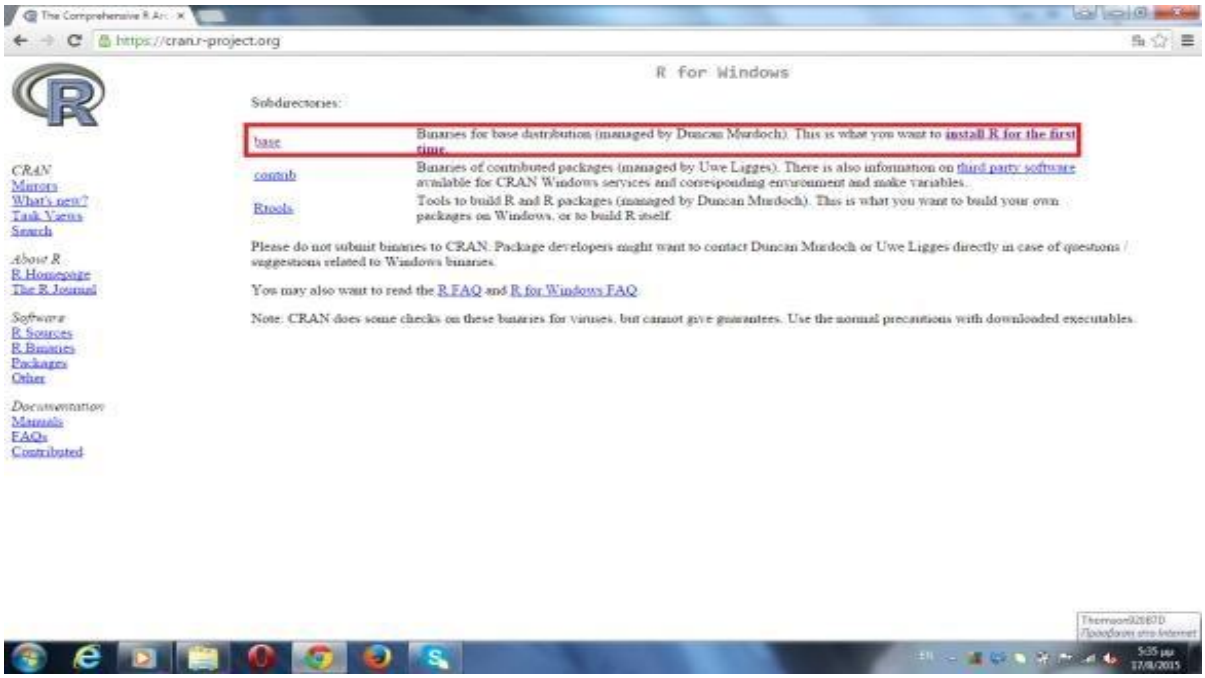
1.2.1 Εγκατάσταση R

-Πηγαίνουμε στη σελίδα cran.r-project.org

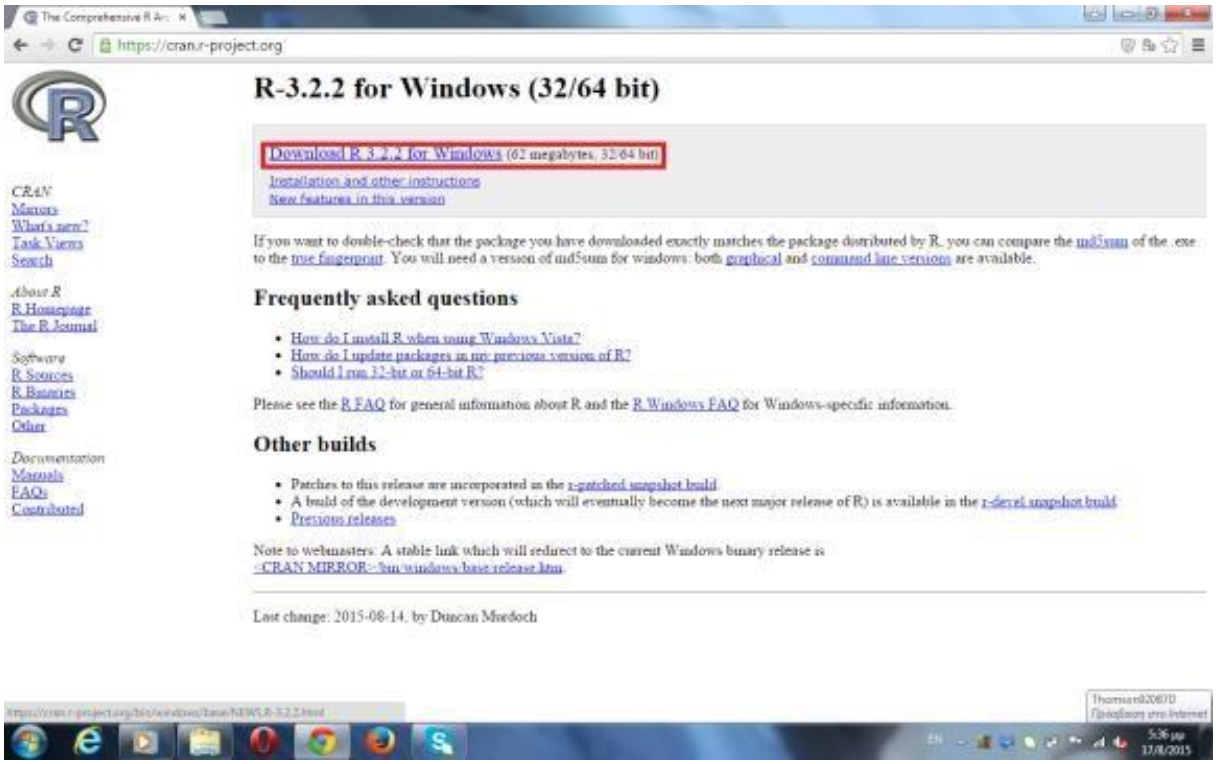
-κάνουμε κλικ στο Download R for Windows



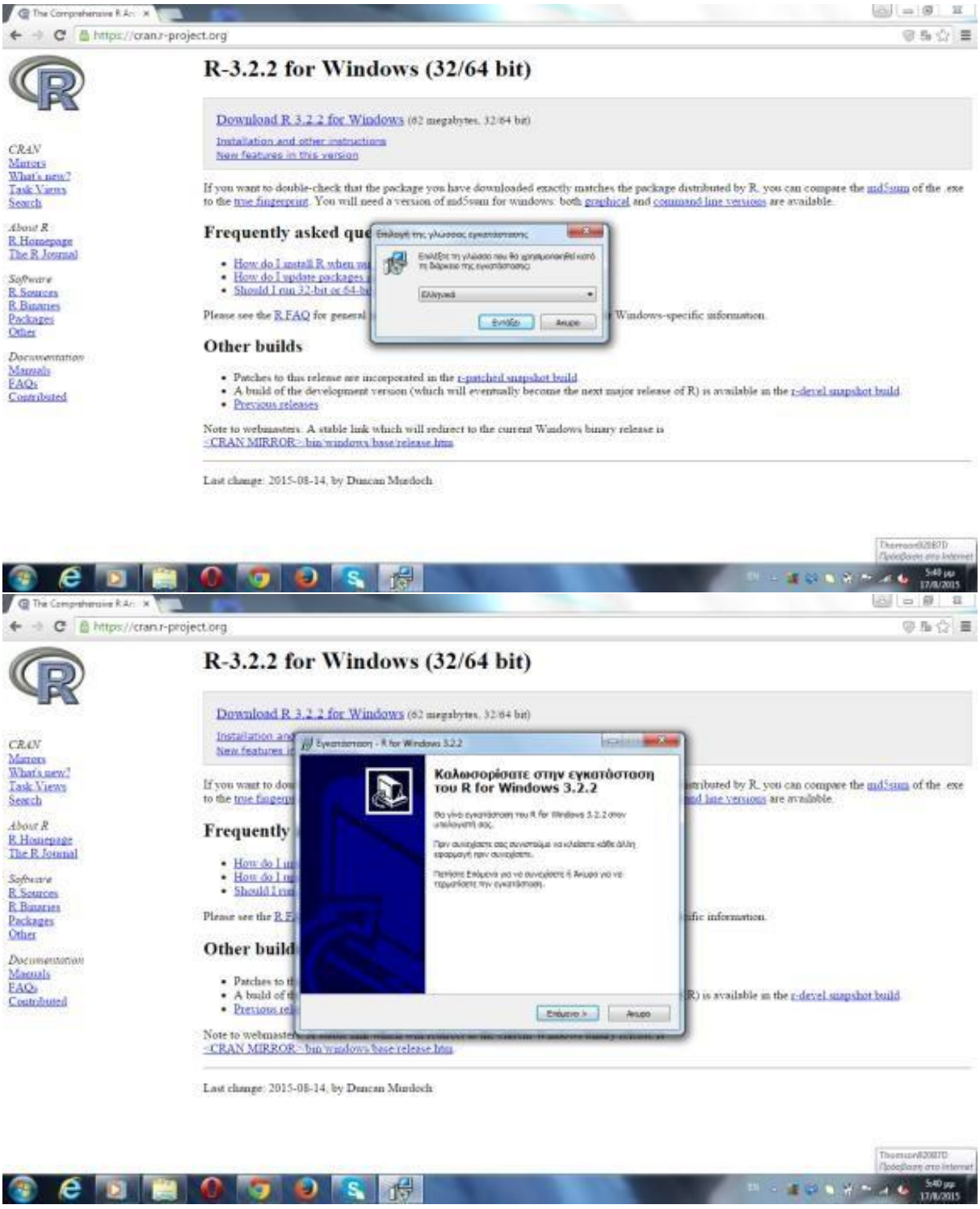
-κάνουμε κλικ στο base



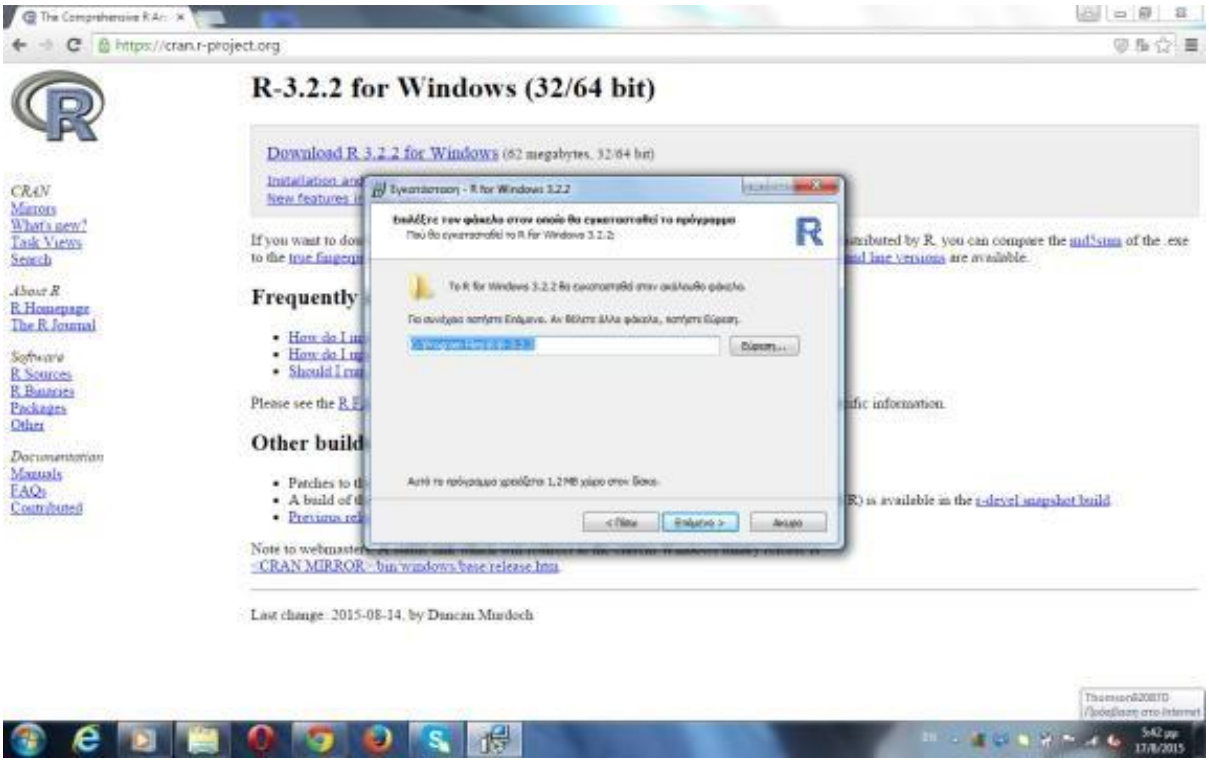
-κάνουμε κλικ στο Download R 3.2.2. for Windows

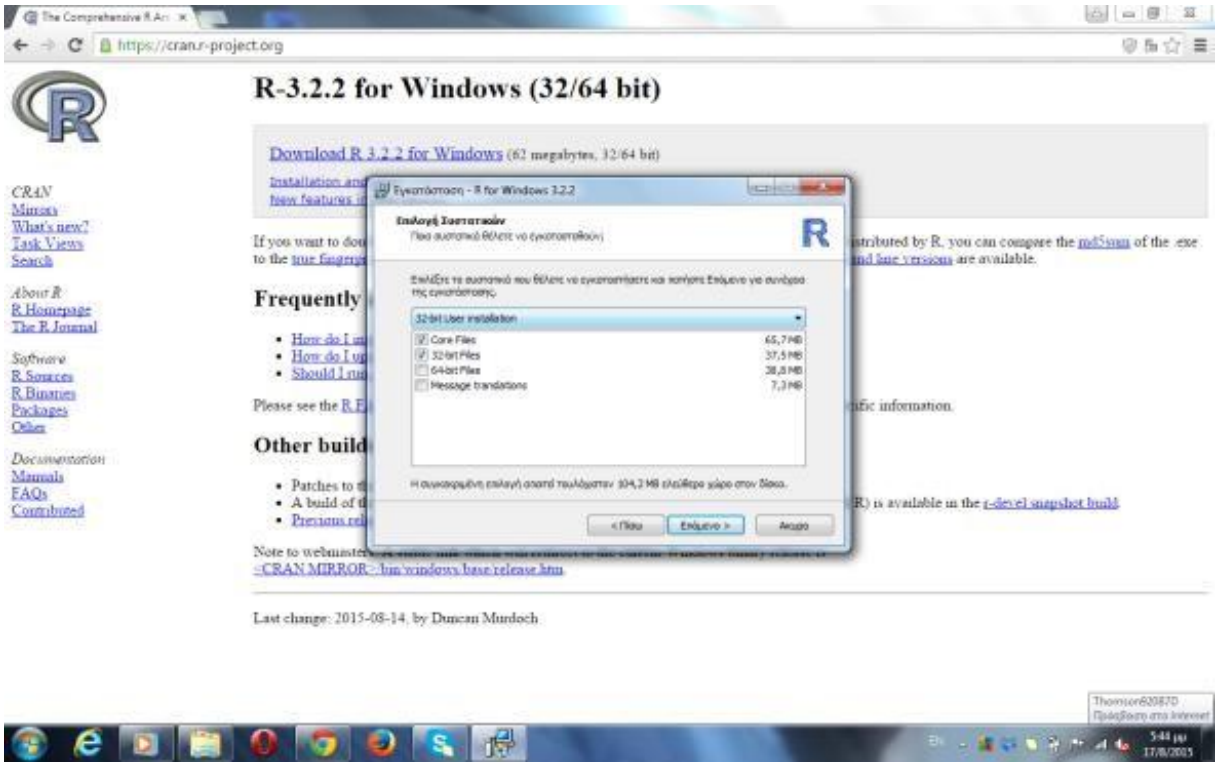


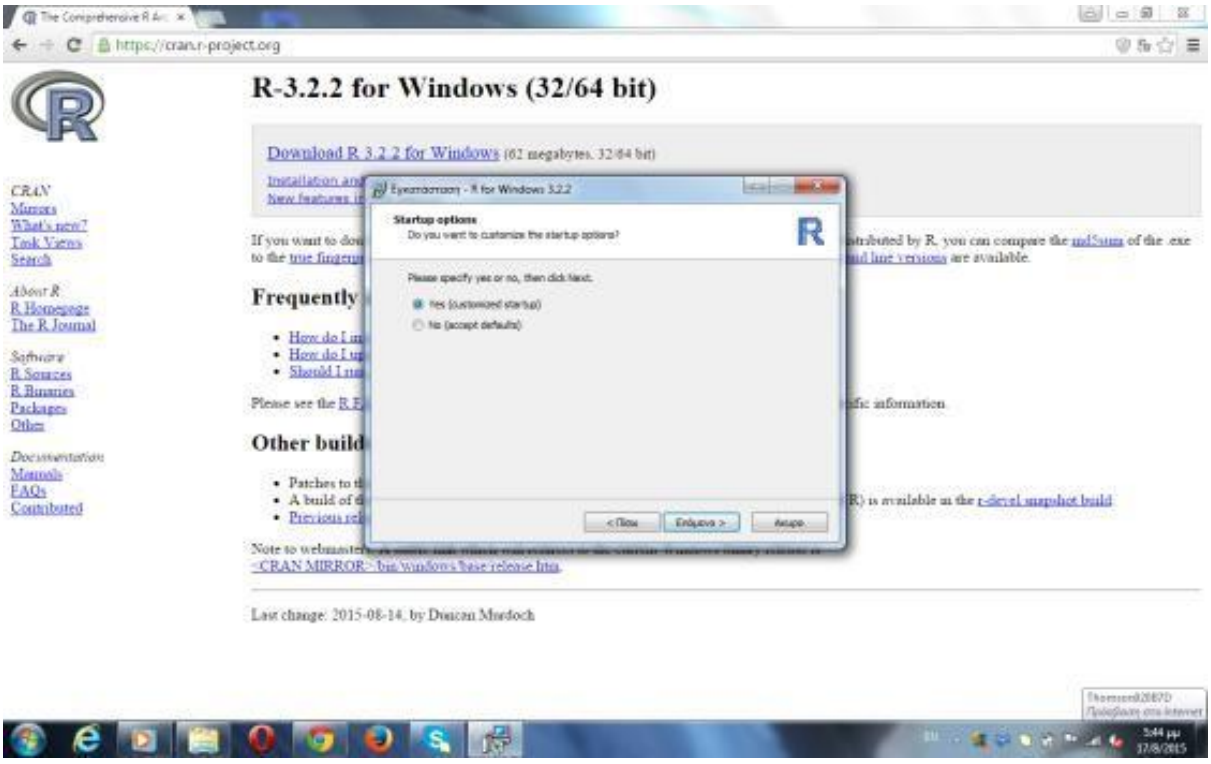
-ακολουθούμε τις προεπιλεγμένες επιλογές μέχρι ώσπου ολοκληρωθεί η εγκατάσταση

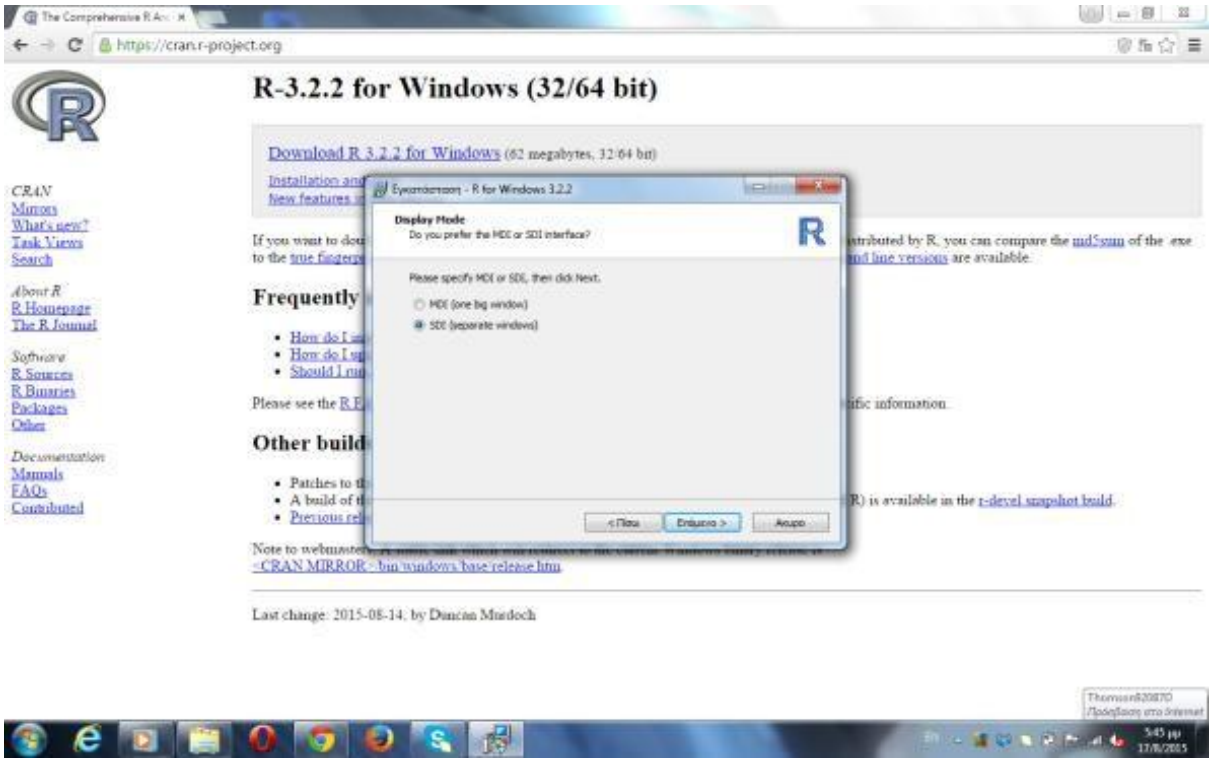

https://cran.r-project.org
R-3.2.2 for Windows (32/64 bit)
Download R 3.2.2 for Windows (62 megabytes, 32/64 bit)
Installation and other instructions
New features in this version
If you want to double-check that the package you have downloaded exactly matches the package distributed by R, you can compare the md5sum of the .exe to the true fingerprint. You will need a version of md5sum for windows: both graphical and command line versions are available.
Frequently asked que
How do I install R when
How do I update packages
Should I run 32-bit or 64-bit
Please see the R FAQ for general
Windows-specific information.
Other builds
Patches to this release are incorporated in the r-patched snapshot build.
A build of the development version (which will eventually become the next major release of R) is available in the r-devel snapshot build.
Previous releases
Note to webmasters: A stable link which will redirect to the current Windows binary release is
<CRAN MIRROR>/bin/windows/base/release.htm
Last change: 2015-08-14, by Duncan Murdoch
CRAN
Mirrors
What's new?
Task Views
Search
About R
R Homepage
The R Journal
Software
R Sources
R Binaries
Packages
Other
Documentation
Manuals
FAQs
Contributed
Επιλογή της γλώσσας εγκατάστασης
Επιλέξτε τη γλώσσα που θα χρησιμοποιηθεί κατά τη διάρκεια της εγκατάστασης.
Ελληνικά
Εντάξει
Άκυρο
Εγκατάσταση - R for Windows 3.2.2
Καλωσορίσατε στην εγκατάσταση του R for Windows 3.2.2
Θα γίνει εγκατάσταση του R for Windows 3.2.2 στον υπολογιστή σας.
Πριν συνεχίσετε σας συνιστούμε να κλείσετε κάθε άλλη εφαρμογή που εκτελείται.
Πατήστε Επόμενο για να συνεχίσετε ή Άκυρο για να τερματίσετε την εγκατάσταση.
Επόμενο >
Άκυρο

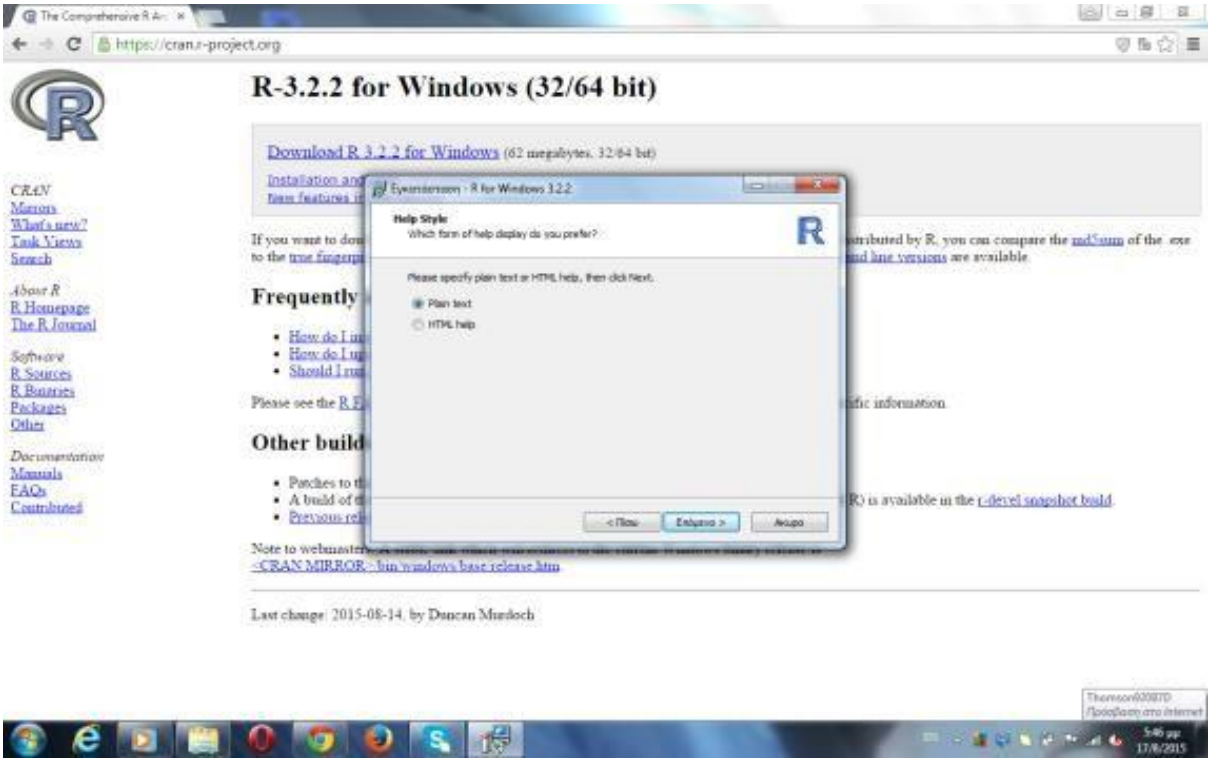

https://cran.r-project.org
R-3.2.2 for Windows (32/64 bit)
Download R 3.2.2 for Windows (62 megabytes, 32/64 bit)
CRAN
Mirrors
What's new?
Task Views
Search
About R
R Homepage
The R Journal
Software
R Sources
R Binaries
Packages
Other
Documentation
Manuals
FAQs
Contributed
Frequently
Other build
Last change: 2015-08-14, by Duncan Murdoch

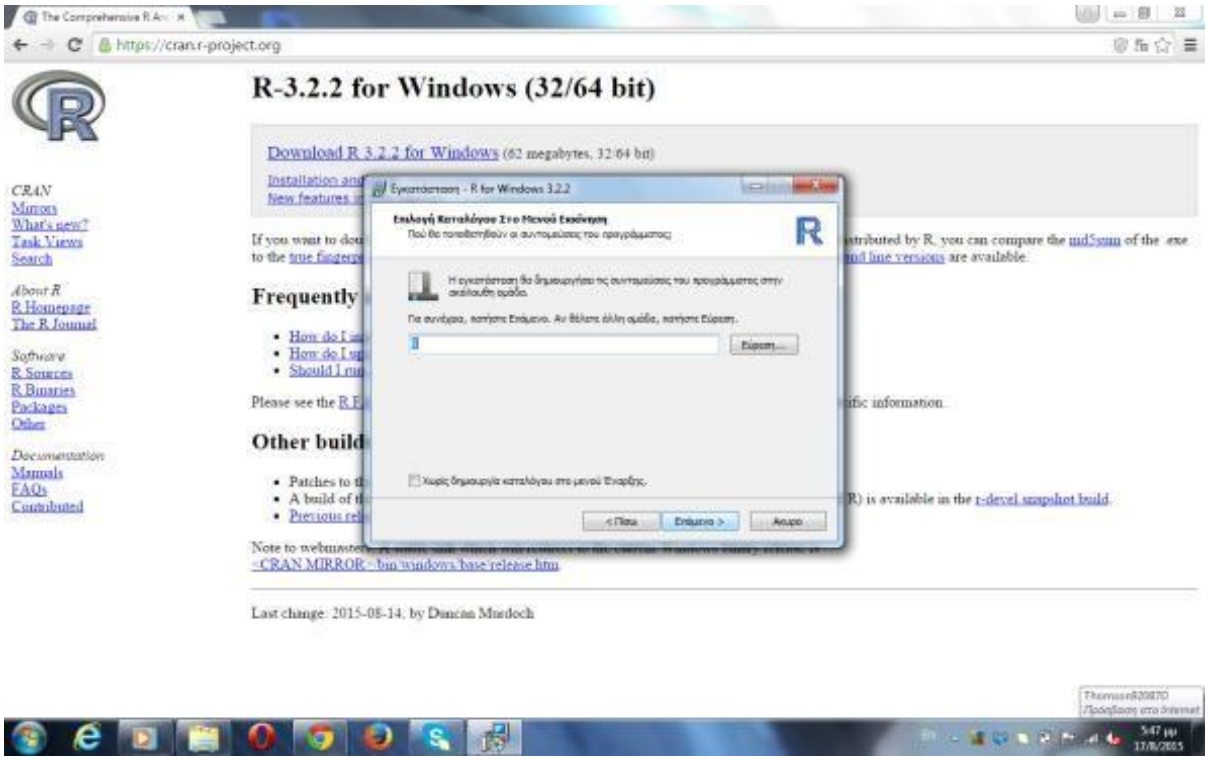

https://cran.r-project.org
R-3.2.2 for Windows (32/64 bit)
Download R 3.2.2 for Windows (62 megabytes, 32/64 bit)
CRAN
Mirrors
What's new?
Task Views
Search
About R
R Homepage
The R Journal
Software
R Sources
R Binaries
Packages
Other
Documentation
Manuals
FAQs
Contributed
Frequently
Other build
Last change: 2015-08-14, by Duncan Murdoch

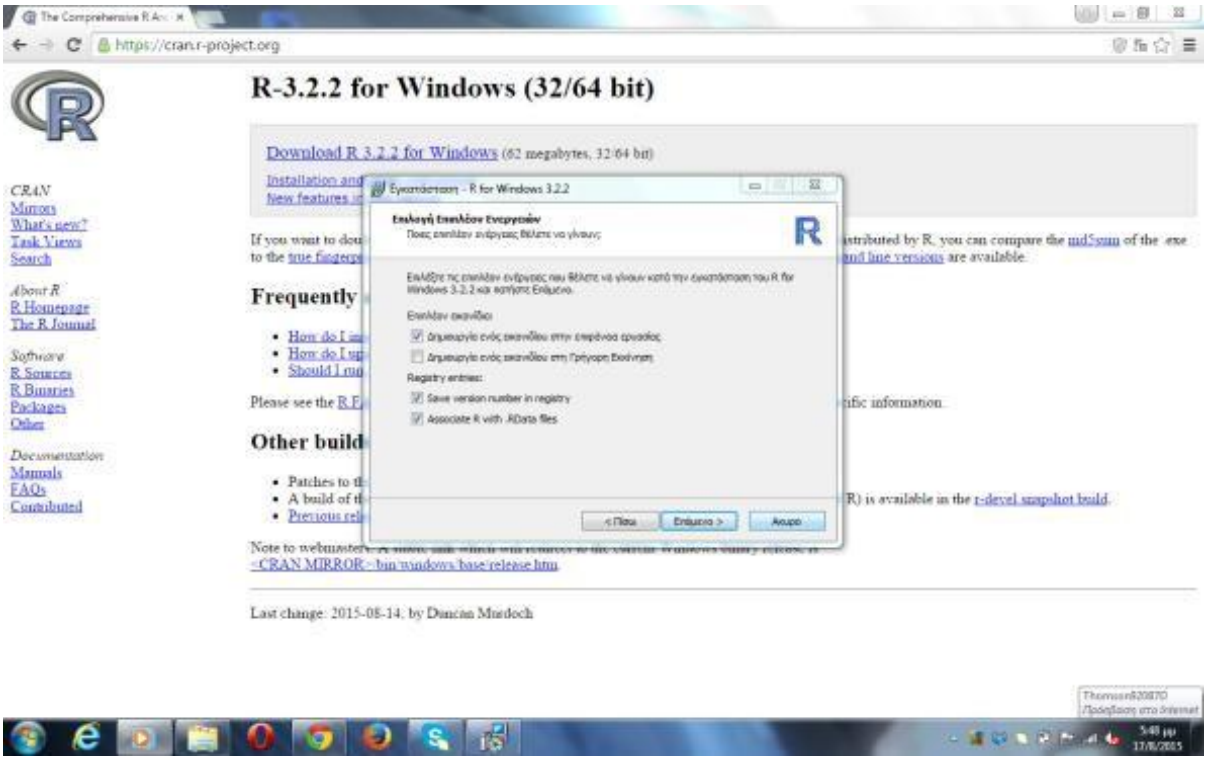

https://cran.r-project.org
R-3.2.2 for Windows (32/64 bit)
Download R 3.2.2 for Windows (62 megabytes, 32/64 bit)
Εγκατάσταση - R for Windows 3.2.2
Επιλογή Συστατικών
32-bit User installation
Core Files
32-bit Files
64-bit Files
Message translations
< Πίσω
Επόμενο >
Άκυρο
Frequently
Other build
Last change: 2015-08-14, by Duncan Murdoch

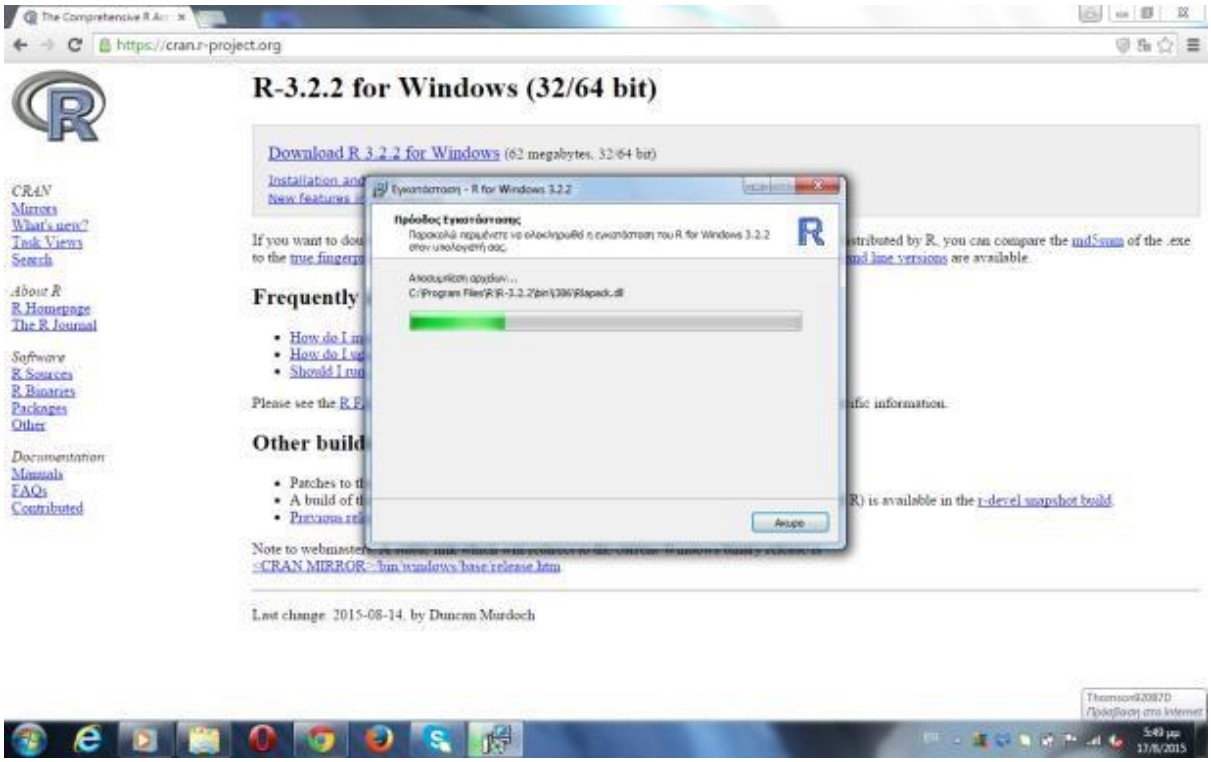

https://cran.r-project.org
R-3.2.2 for Windows (32/64 bit)
Download R 3.2.2 for Windows (62 megabytes, 32/64 bit)
Εγκατάσταση - R for Windows 3.2.2
Startup options
Do you want to customize the startup options?
Please specify yes or no, then click Next.
yes (customized startup)
no (accept defaults)
< Πίσω
Επόμενο >
Άκυρο
Frequently
Other build
Last change: 2015-08-14, by Duncan Murdoch


https://cran.r-project.org
R-3.2.2 for Windows (32/64 bit)
Download R 3.2.2 for Windows (62 megabytes, 32/64 bit)
Display Mode
Do you prefer the MDI or SDI interface?
Please specify MDI or SDI, then click Next.
MDI (one big window)
SDI (separate windows)
Frequently
Other build
Last change: 2015-08-14, by Duncan Murdoch

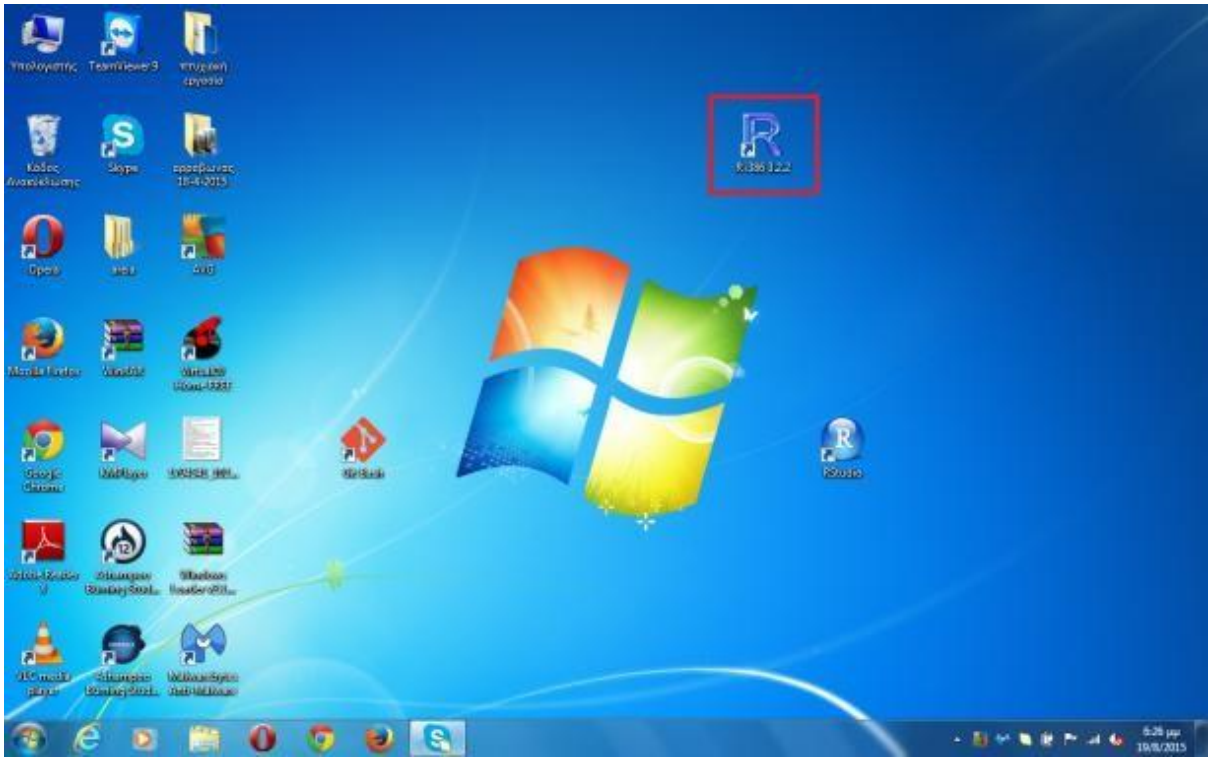

https://cran.r-project.org
R-3.2.2 for Windows (32/64 bit)
Download R 3.2.2 for Windows (62 megabytes, 32/64 bit)
Help Style
Which form of help display do you prefer?
Please specify plain text or HTML help, then click Next.
Plain text
HTML help
Frequently
Other build
Last change: 2015-08-14, by Duncan Murdoch

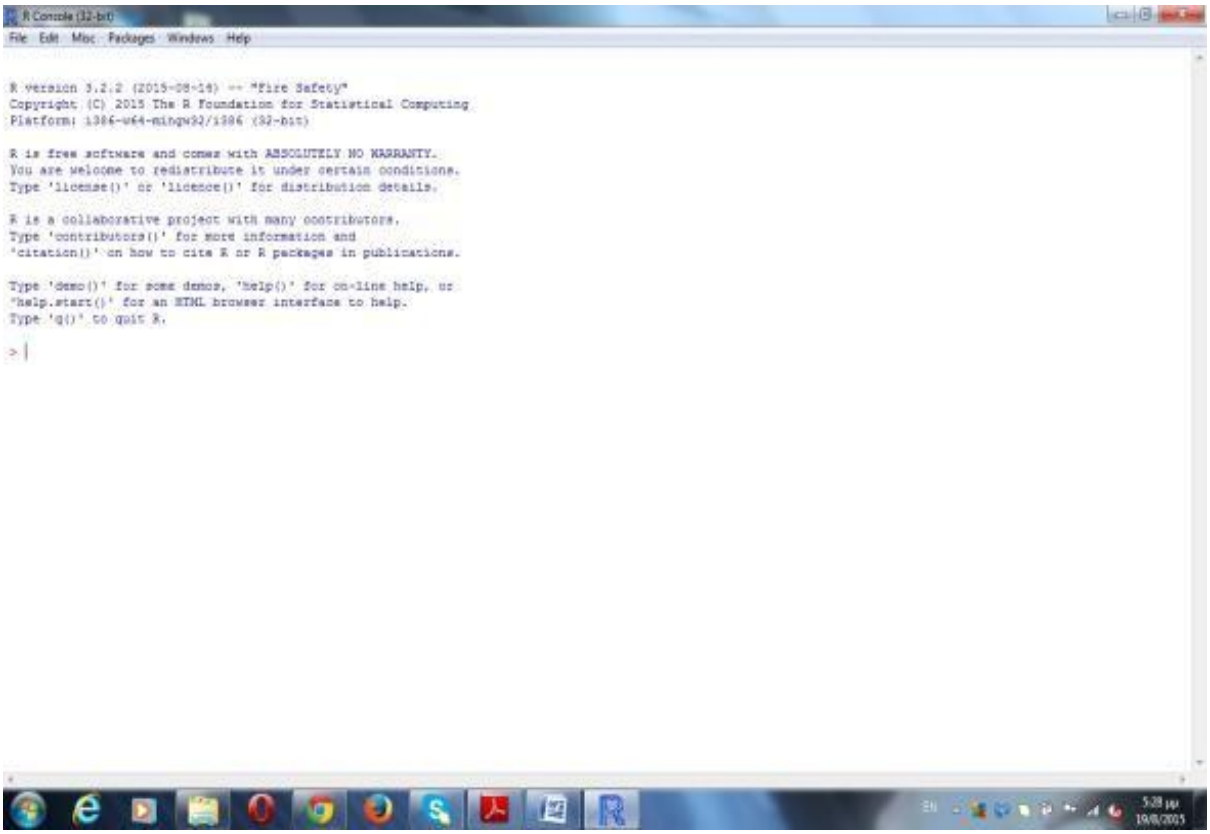

https://cran.r-project.org
R-3.2.2 for Windows (32/64 bit)
Download R 3.2.2 for Windows (62 megabytes, 32/64 bit)
CRAN
Mirrors
What's new?
Task Views
Search
About R
R Homepage
The R Journal
Software
R Sources
R Binaries
Packages
Other
Documentation
Manuals
FAQs
Contributed
Frequently
Other build
Last change: 2015-08-14, by Duncan Murdoch


https://cran.r-project.org
R-3.2.2 for Windows (32/64 bit)
Download R 3.2.2 for Windows (62 megabytes, 32/64 bit)
CRAN
Mirrors
What's new?
Task Views
Search
About R
R Homepage
The R Journal
Software
R Sources
R Binaries
Packages
Other
Documentation
Manuals
FAQs
Contributed
Frequently
Other build
Last change: 2015-08-14, by Duncan Murdoch


https://cran.r-project.org
R-3.2.2 for Windows (32/64 bit)
Download R 3.2.2 for Windows (62 megabytes, 32/64 bit)
CRAN
Mirrors
What's new?
Task Views
Search
About R
R Homepage
The R Journal
Software
R Sources
R Binaries
Packages
Other
Documentation
Manuals
FAQs
Contributed
Frequently
Other build
Last change: 2015-08-14, by Duncan Murdoch
Εγκατάσταση - R for Windows 3.2.2
Πρόοδος εγκατάστασης
Άκυρο


https://cran.r-project.org
R-3.2.2 for Windows (32/64 bit)
Download R 3.2.2 for Windows (62 megabytes, 32/64 bit)
Εγκατάσταση - R for Windows 3.2.2
Ολοκληρώνοντας τον Οδηγό Εγκατάστασης του R for Windows 3.2.2
Τέλος
Frequently
Other build
Last change: 2015-08-14. by Duncan Murdoch

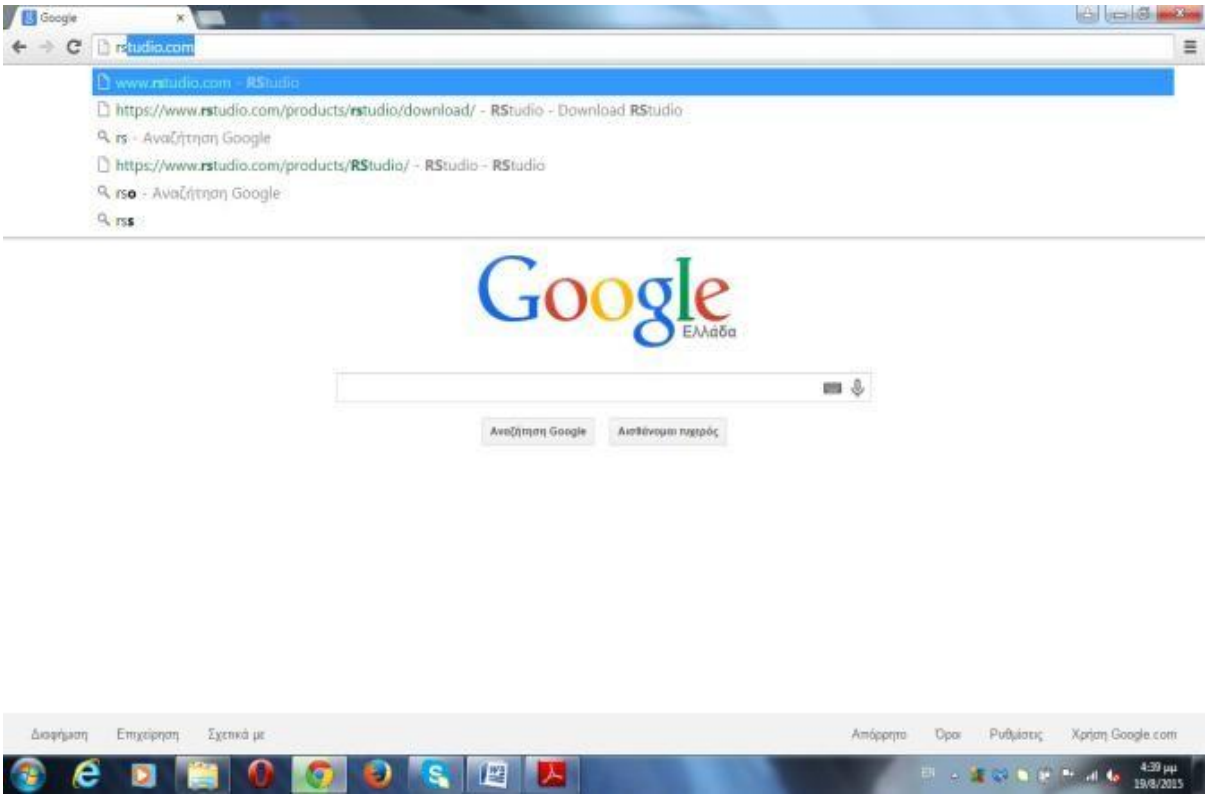




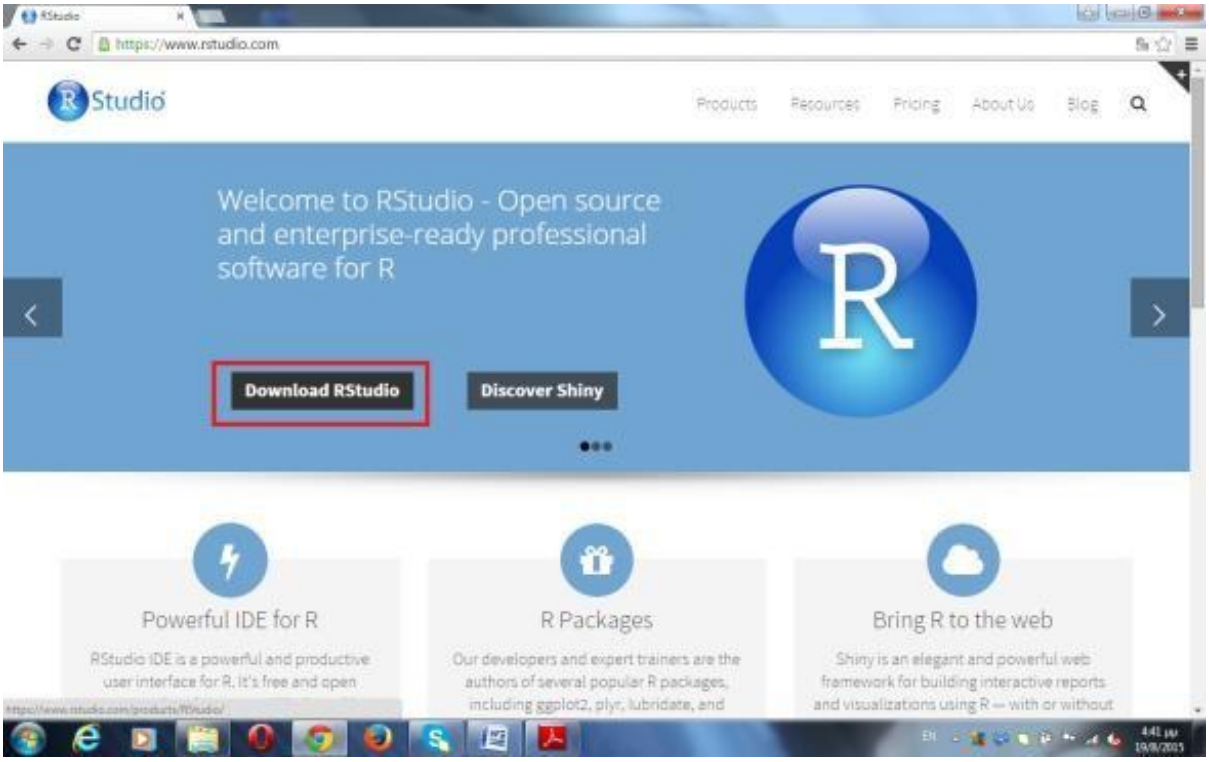
Η εγκατάσταση του R ολοκληρώθηκε όπως φαίνεται από την κονσόλα R παραπάνω.

1.2.2 Εγκατάσταση rstudio

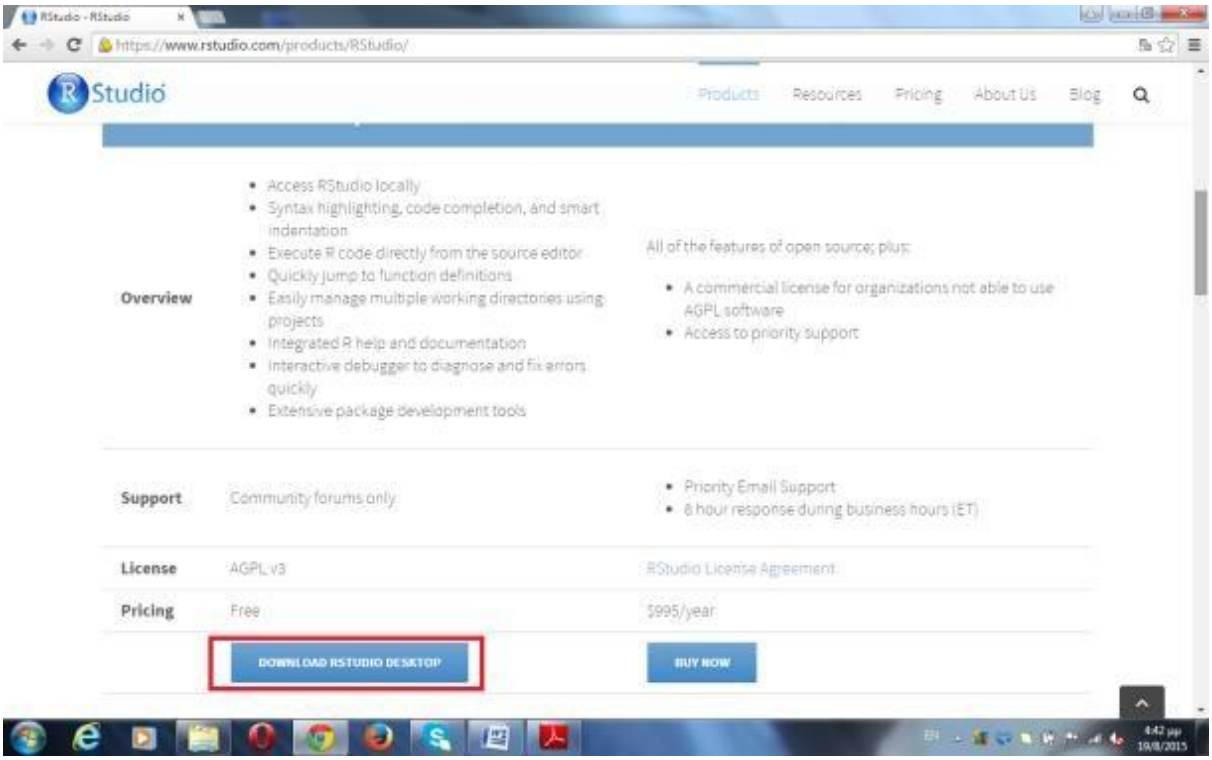
- πηγαίνουμε στη σελίδα www.rstudio.com



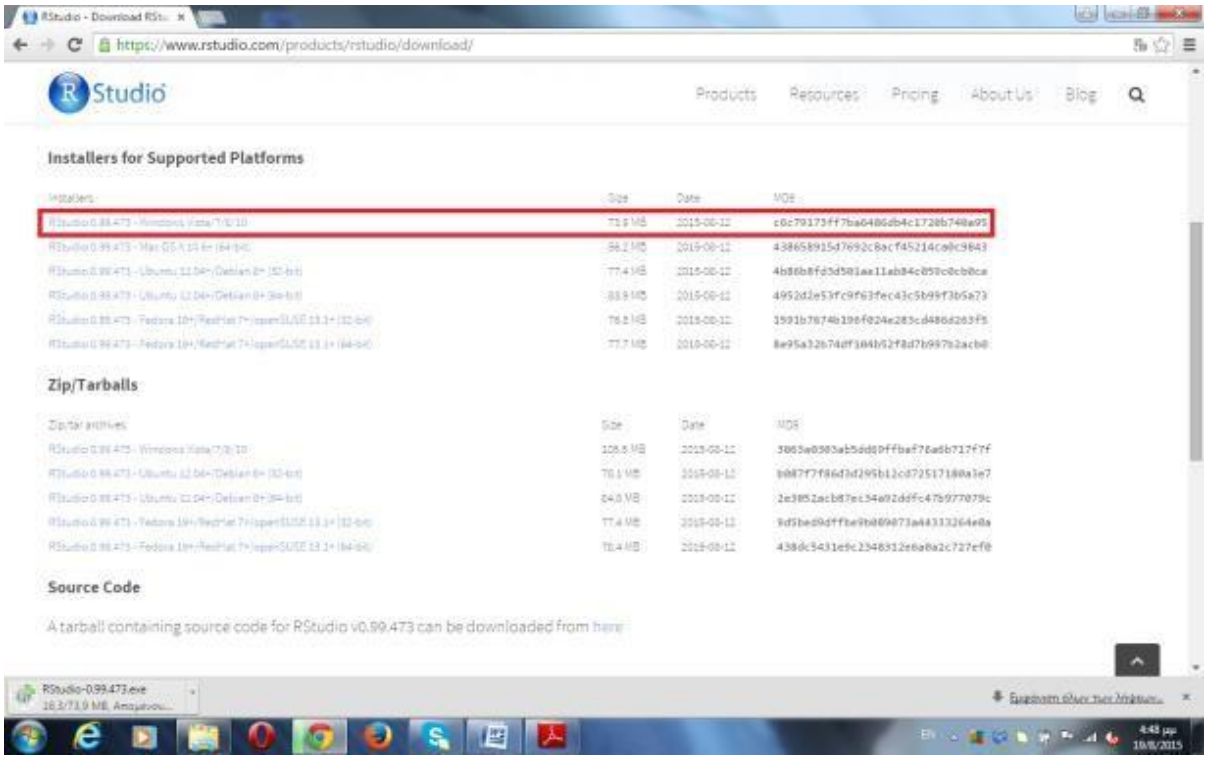
-κάνουμε κλικ στο Download RStudio



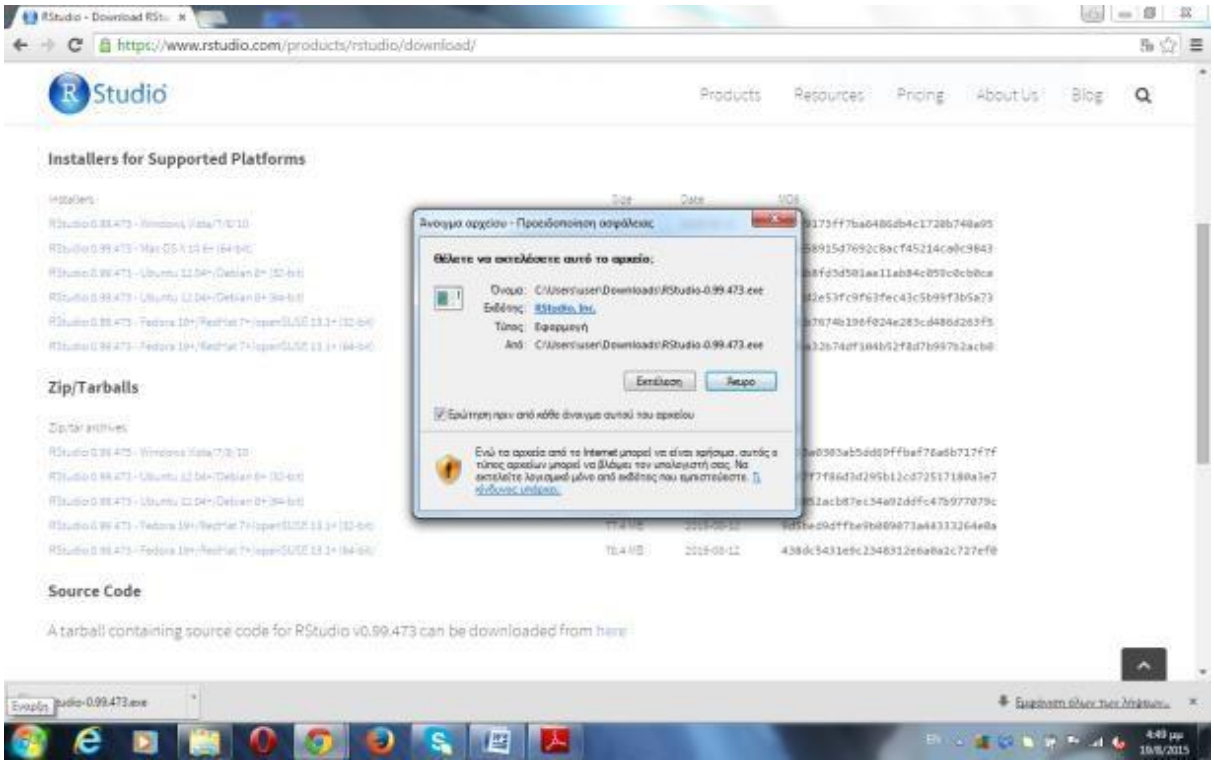
- κάνουμε κλικ στο DOWNLOAD RSTUDIO DESKTOP

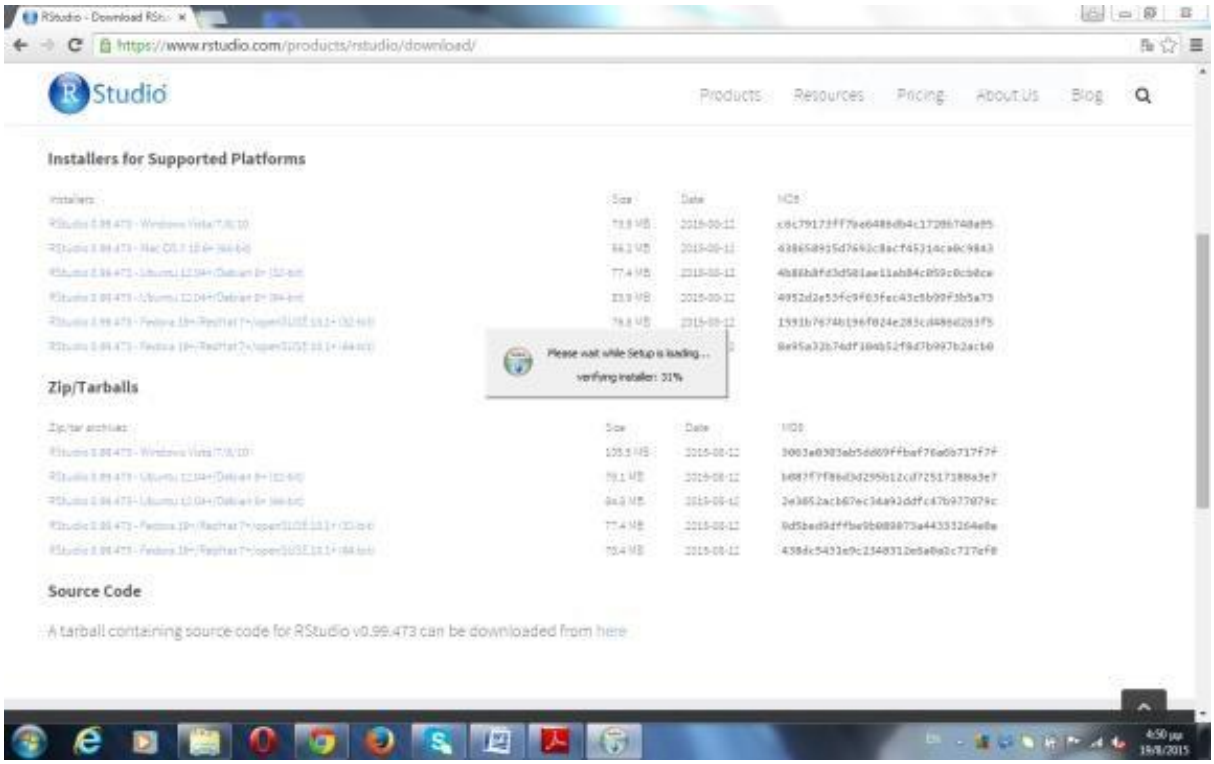


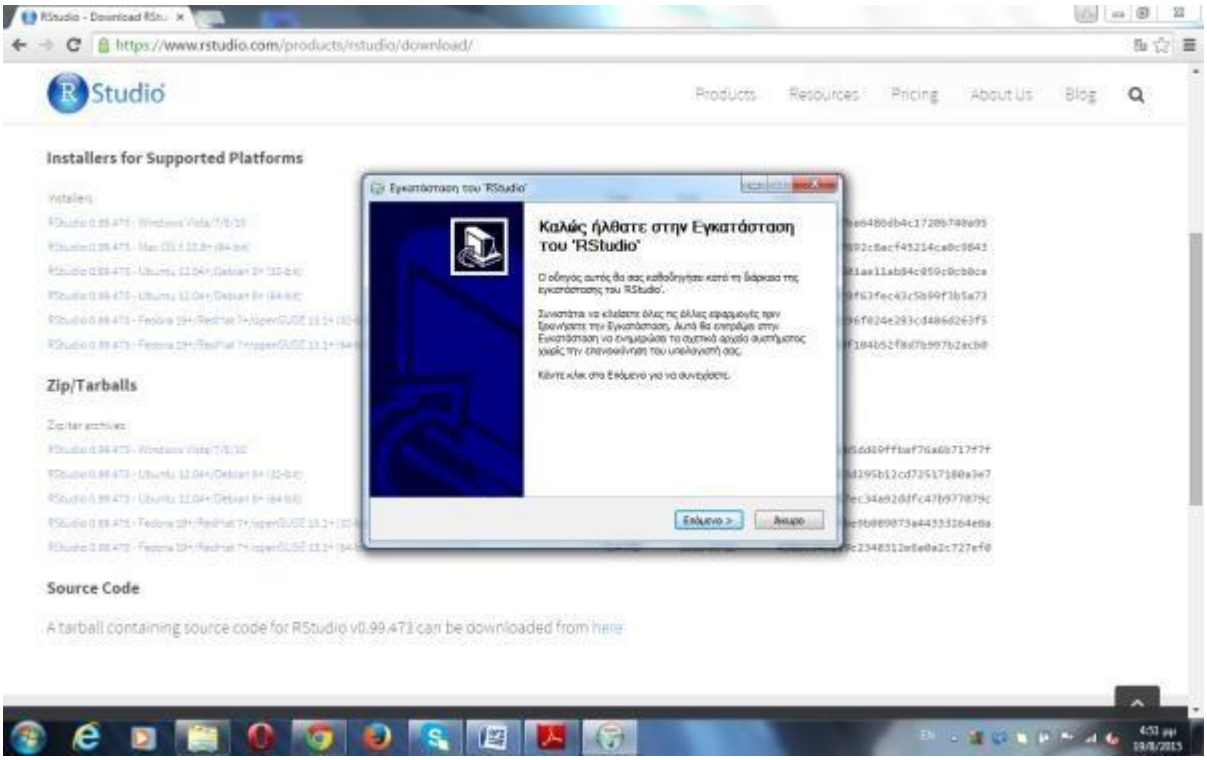
-κάνουμε κλικ για να κατεβάσουμε την έκδοση του RStudio Desktop

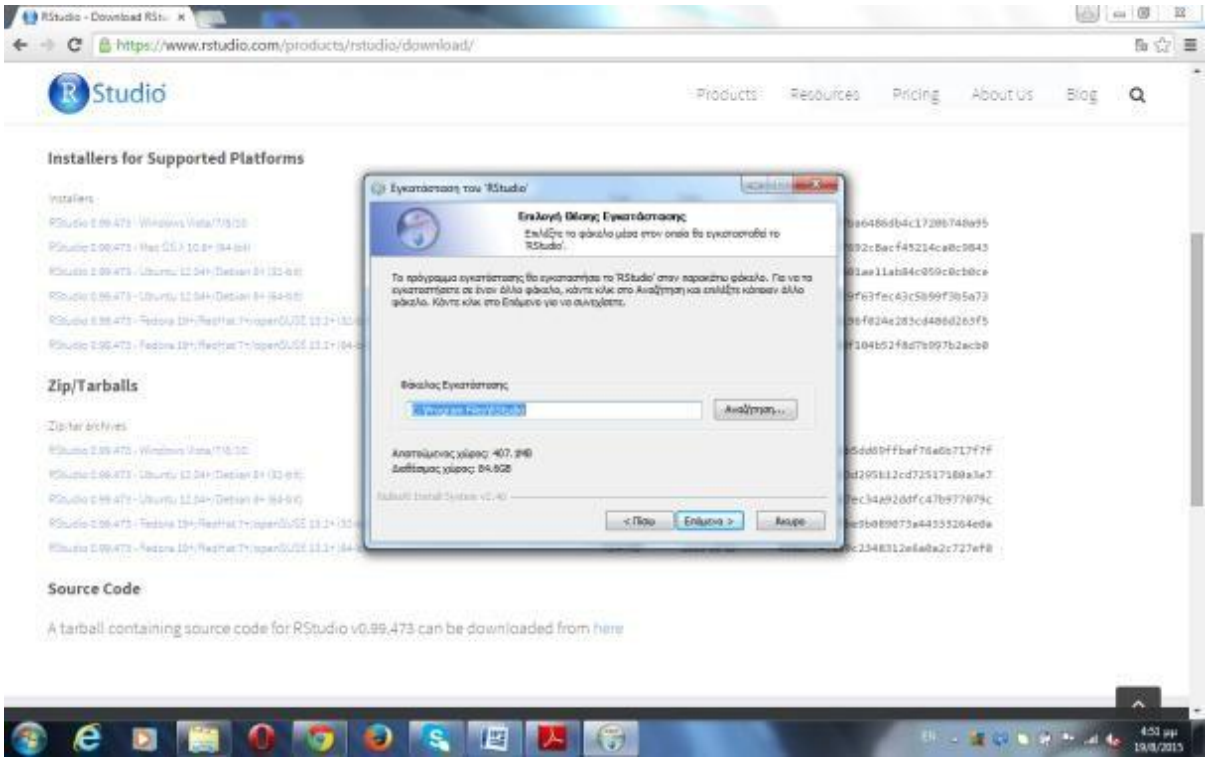


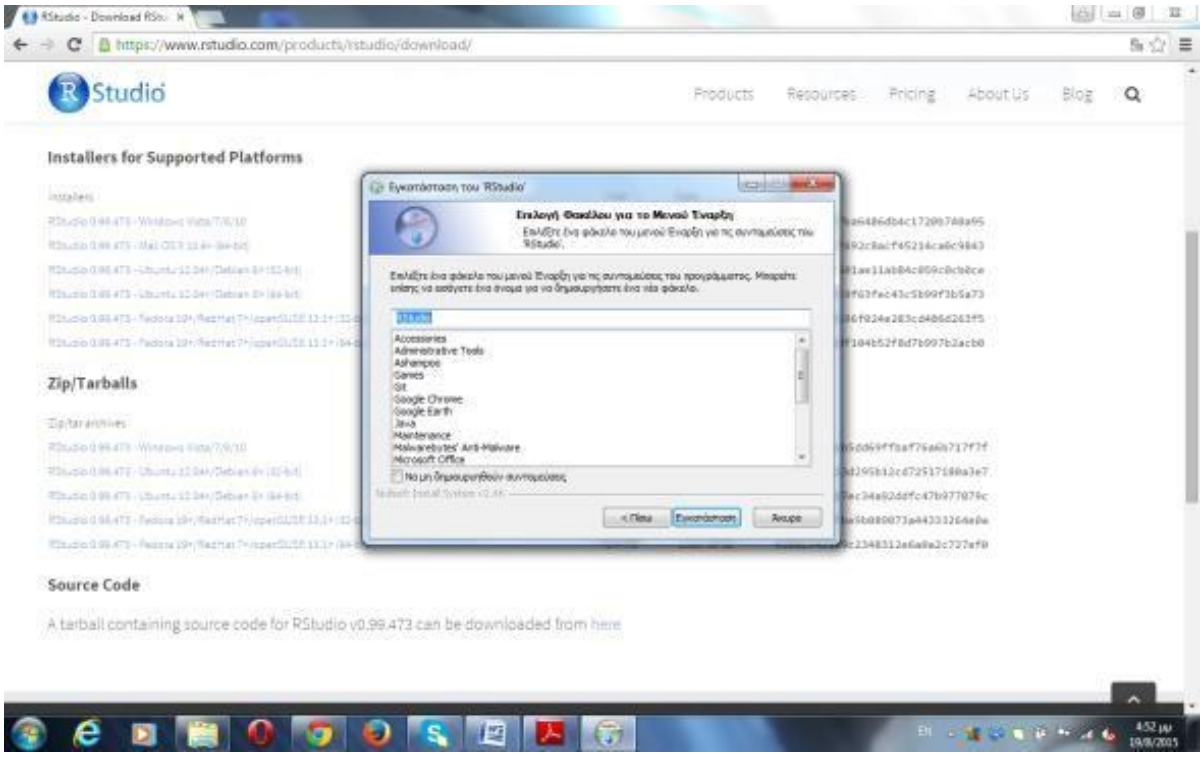
-ακολουθούμε τις προεπιλεγμένες επιλογές μέχρι ώσπου ολοκληρωθεί η εγκατάσταση

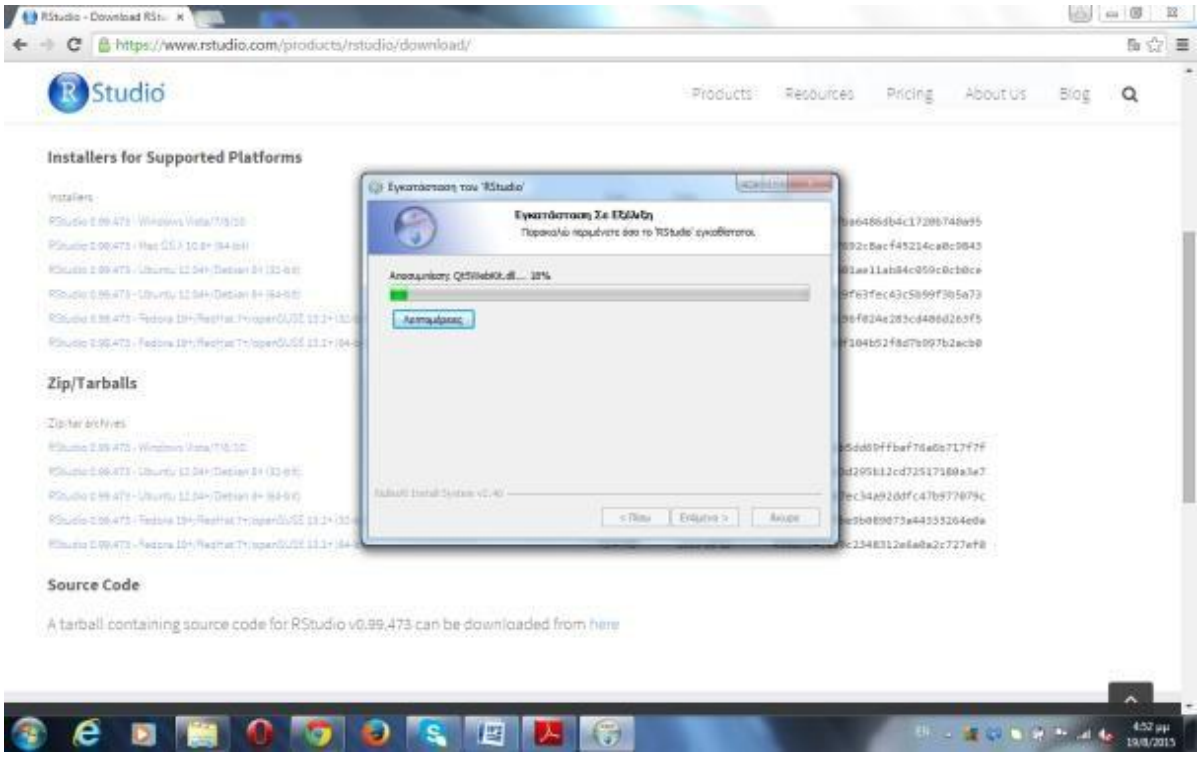


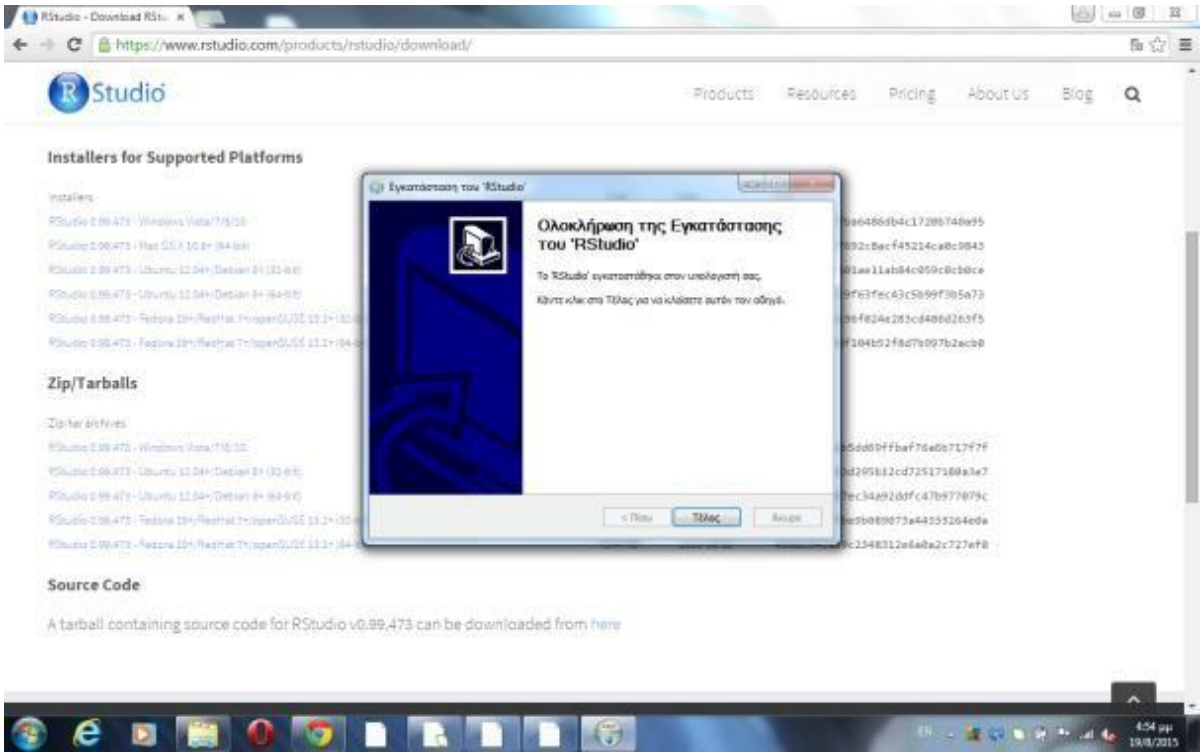


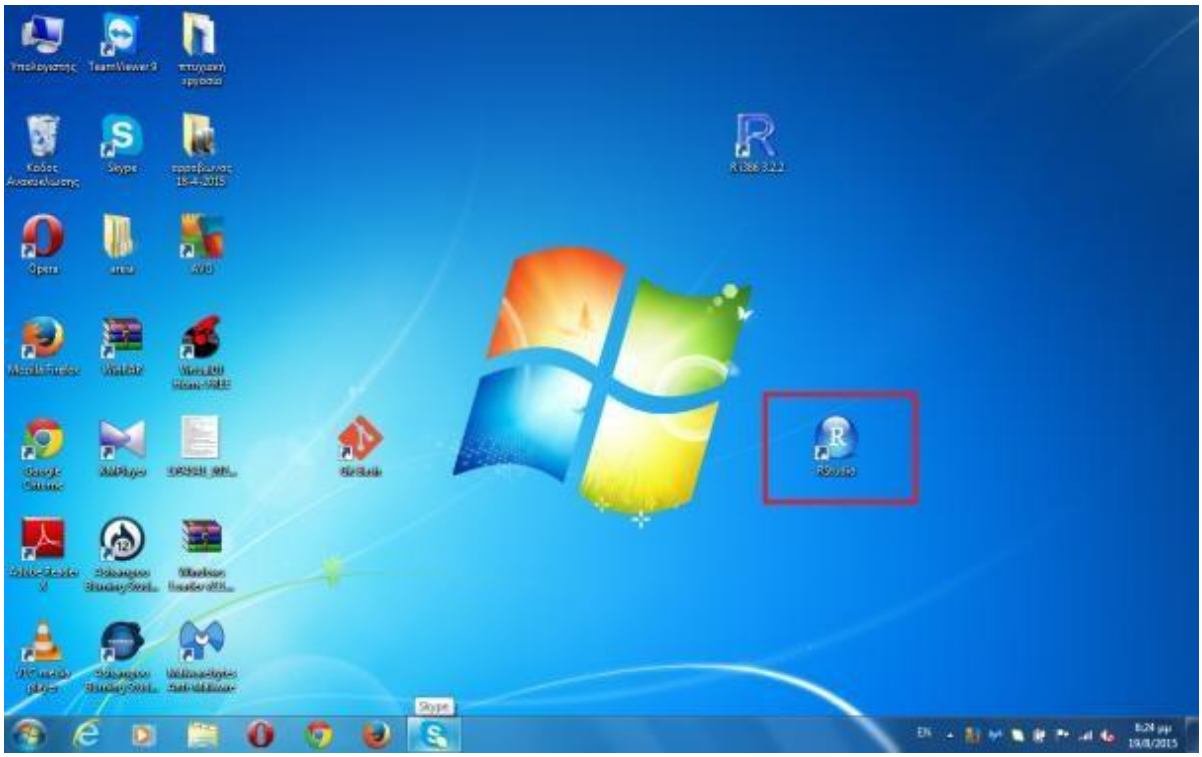

https://www.rstudio.com/products/rstudio/download/
Products
Resources
Pricing
About Us
Blog
Installers for Supported Platforms
Εγκατάσταση του 'RStudio'
Καλώς ήλθατε στην Εγκατάσταση του 'RStudio'
Επόμενο >
Άκυρο
Zip/Tarballs
Source Code
A tarball containing source code for RStudio v0.99.473 can be downloaded from here

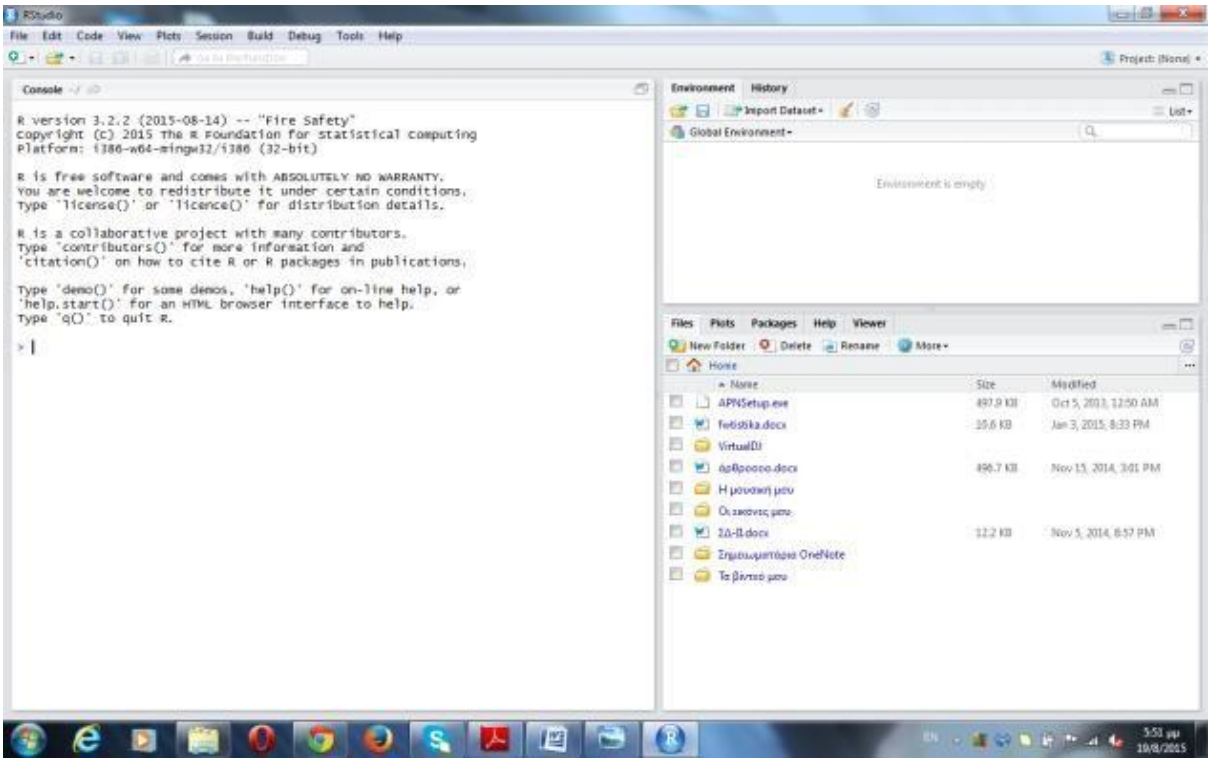

https://www.rstudio.com/products/rstudio/download/
Products
Resources
Pricing
About Us
Blog
Installers for Supported Platforms
Εγκατάσταση του 'RStudio'
Επιλογή Θέσης Εγκατάστασης
Φάκελος Εγκατάστασης
Αναζήτηση...
< Πίσω
Επόμενο >
Άκυρο
Zip/Tarballs
Source Code
A tarball containing source code for RStudio v0.99.473 can be downloaded from here


RStudio - Download RSt
https://www.rstudio.com/products/rstudio/download/
RStudio
Products
Resources
Pricing
About Us
Blog
Installers for Supported Platforms
Zip/Tarballs
Source Code
A tarball containing source code for RStudio v0.99.473 can be downloaded from here
Εγκατάσταση του 'RStudio'
Επιλογή Φακέλου για το Μενού Έναρξη
Επιλέξτε ένα φάκελο του μενού Έναρξη για τις συντομεύσεις του RStudio.
Επιλέξτε ένα φάκελο του μενού Έναρξη για τις συντομεύσεις του προγράμματος. Μπορείτε επίσης να εισάγετε ένα όνομα για να δημιουργήσετε ένα νέο φάκελο.
Accessories
Administrative Tools
Ashampoo
Games
Git
Google Chrome
Google Earth
Java
Maintenance
Malwarebytes' Anti-Malware
Microsoft Office
Να μη δημιουργηθούν συντομεύσεις
< Πίσω
Εγκατάσταση
Άκυρο


RStudio - Download RSt
https://www.rstudio.com/products/rstudio/download/
RStudio
Products
Resources
Pricing
About Us
Blog
Installers for Supported Platforms
Zip/Tarballs
Source Code
A tarball containing source code for RStudio v0.99.473 can be downloaded from here
Εγκατάσταση του 'RStudio'
Εγκατάσταση Σε Εξέλιξη
Παρακαλώ περιμένετε όσο το 'RStudio' εγκαθίσταται.
Αποσυμπίεση: Qt5WebKit.dll... 18%
Αντιγράφεται...
< Πίσω
Επόμενο >
Άκυρο


RStudio - Download RSt
https://www.rstudio.com/products/rstudio/download/
RStudio
Products
Resources
Pricing
About Us
Blog
Installers for Supported Platforms
Εγκατάσταση του 'RStudio'
Ολοκλήρωση της Εγκατάστασης του 'RStudio'
Το 'RStudio' εγκαταστάθηκε στον υπολογιστή σας.
Κάντε κλικ στο Τέλος για να κλείσετε αυτόν τον οδηγό.
< Πίσω
Τέλος
Άκυρο
Zip/Tarballs
Source Code
A tarball containing source code for RStudio v0.99.473 can be downloaded from here


Υπολογιστής
TeamViewer 9
Skype
Opera
Mozilla Firefox
Google Chrome
Skype

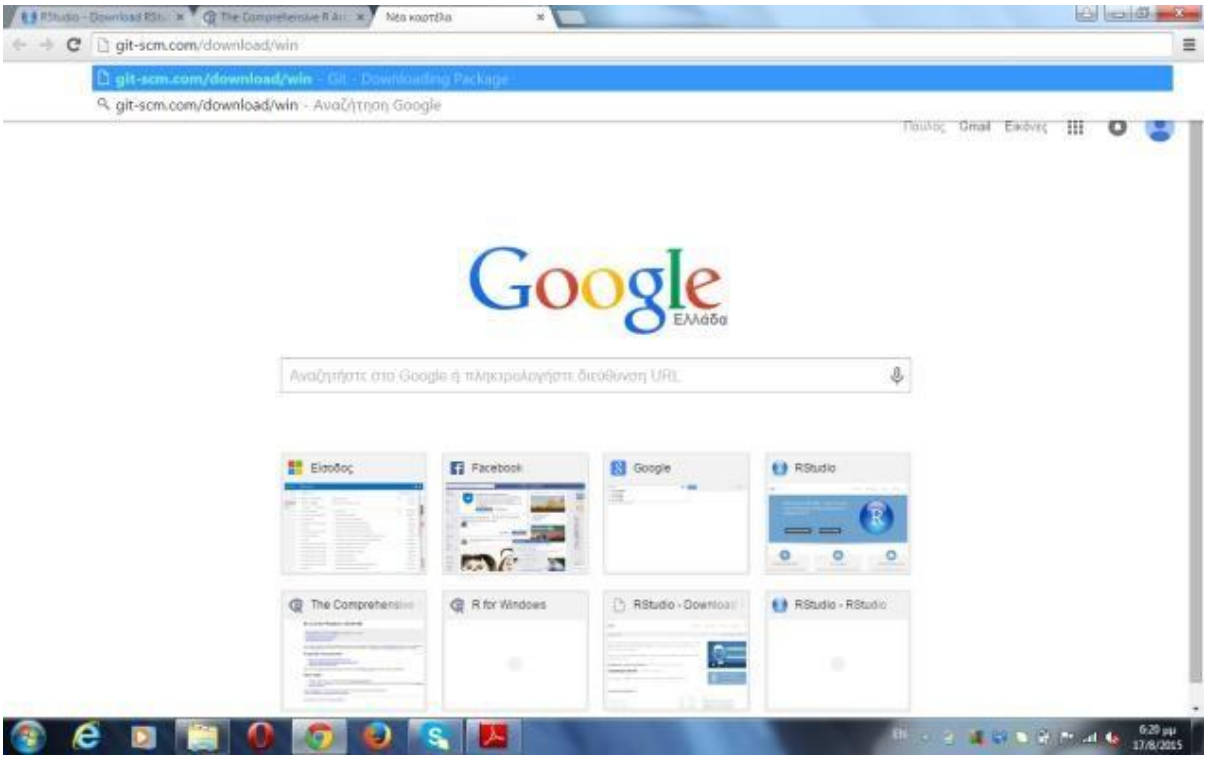


Η εγκατάσταση του RStudio ολοκληρώθηκε.

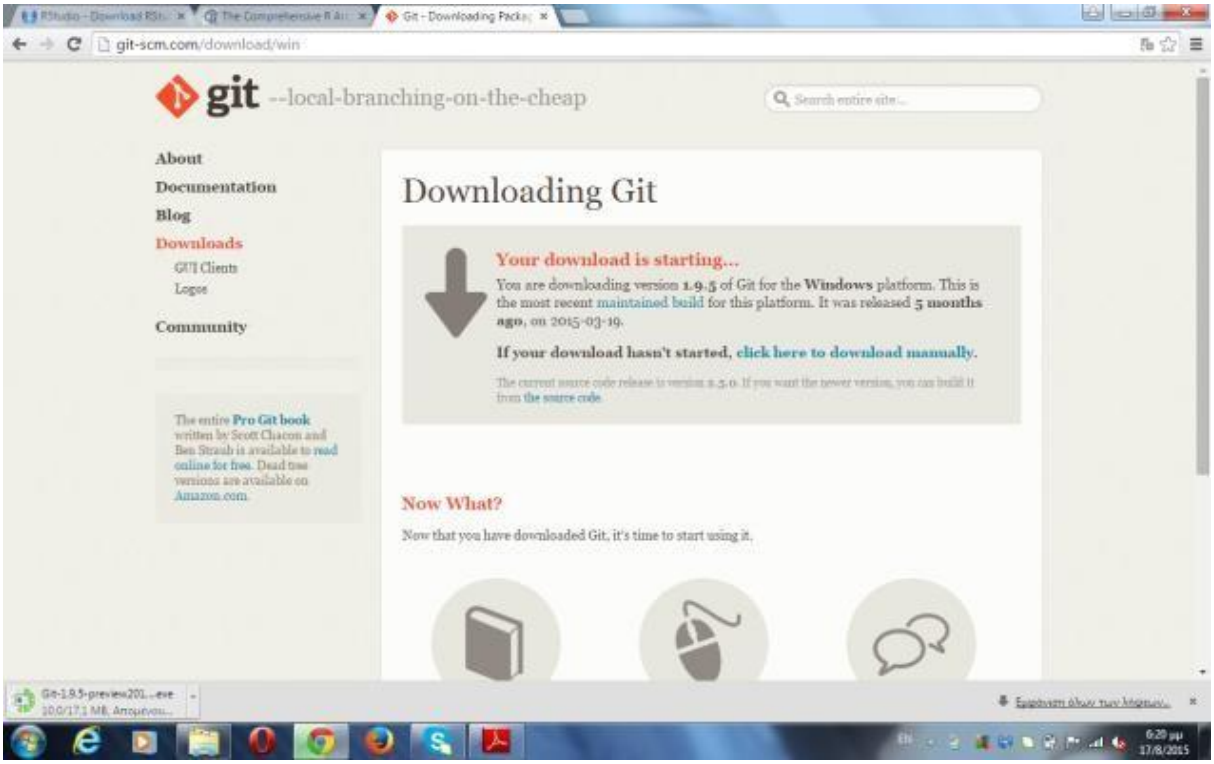
1.3 Εγκατάσταση Git και github

Το git είναι ένα σύστημα ελέγχου διαχείρισης μεταβολών (revision control system) για ένα ή περισσότερα αρχεία που όλα μαζί αποτελούν κάποιο project. Με έμφαση στην ταχύτητα και δυνατότητες κατανομής διαδικτυακά σε servers, διευκολύνει την ομαδική συνεργασία και αποτελεί σήμερα το πιο διαδεδομένο τέτοιο σύστημα σε χρήση.

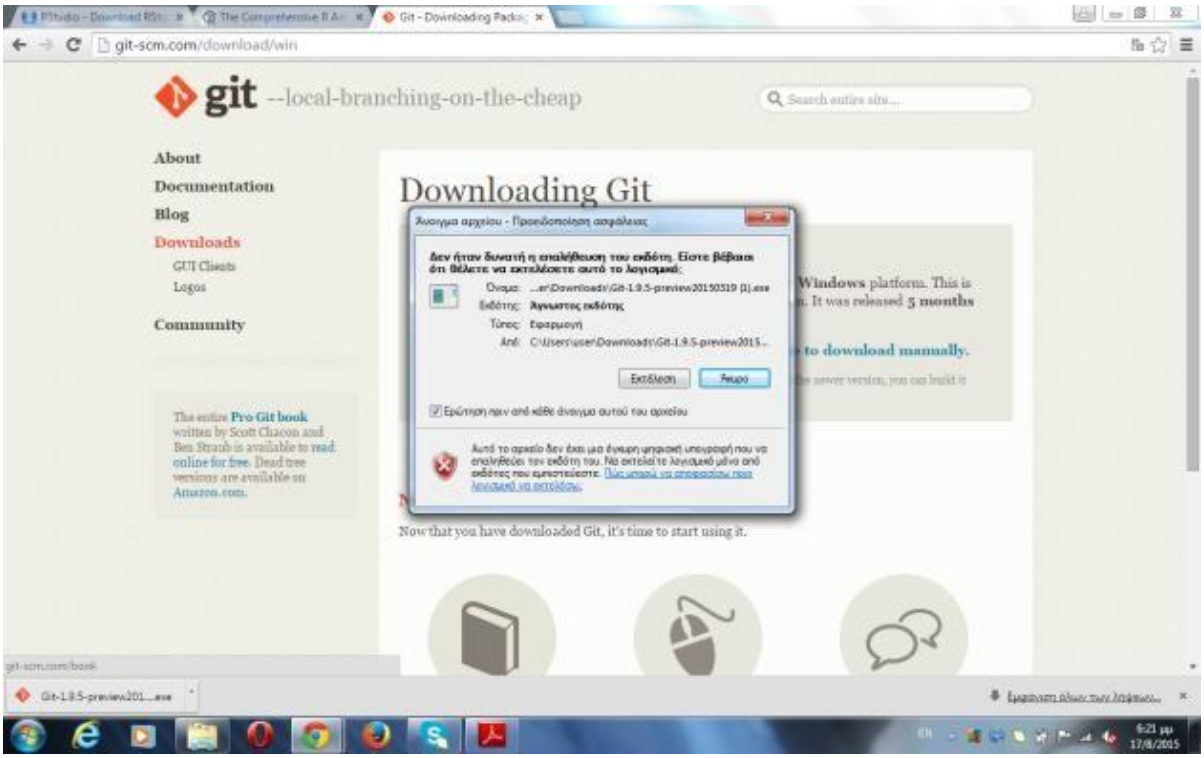
- πηγαίνουμε στη σελίδα git-scm.com/download/win



-ξεκινάει η λήψη

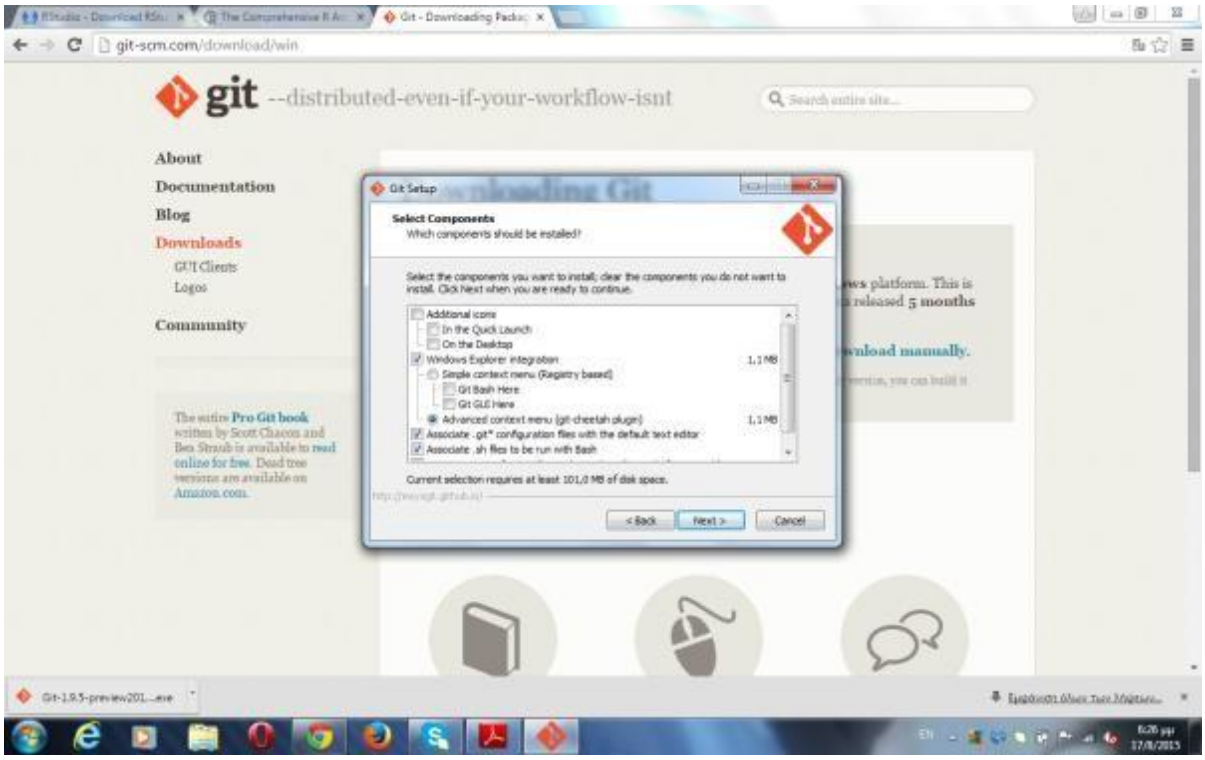


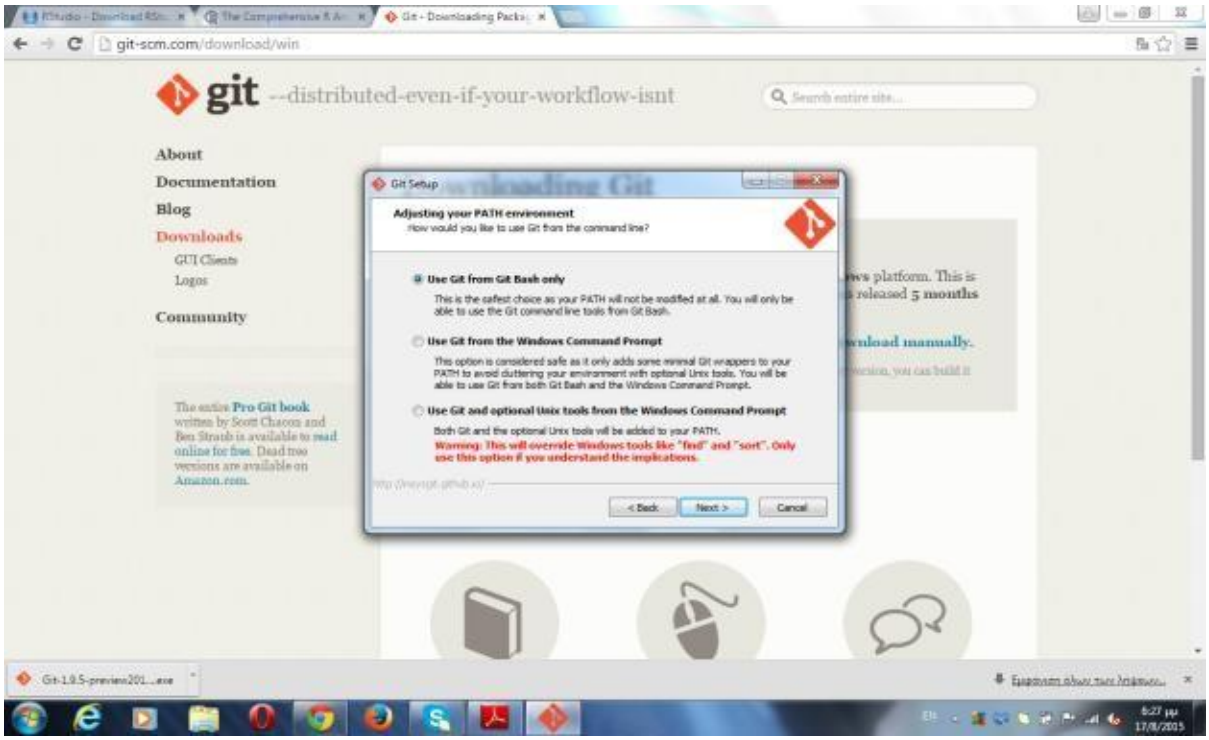
-ακολουθούμε τις προεπιλεγμένες επιλογές μέχρι ώσπου ολοκληρωθεί η εγκατάσταση

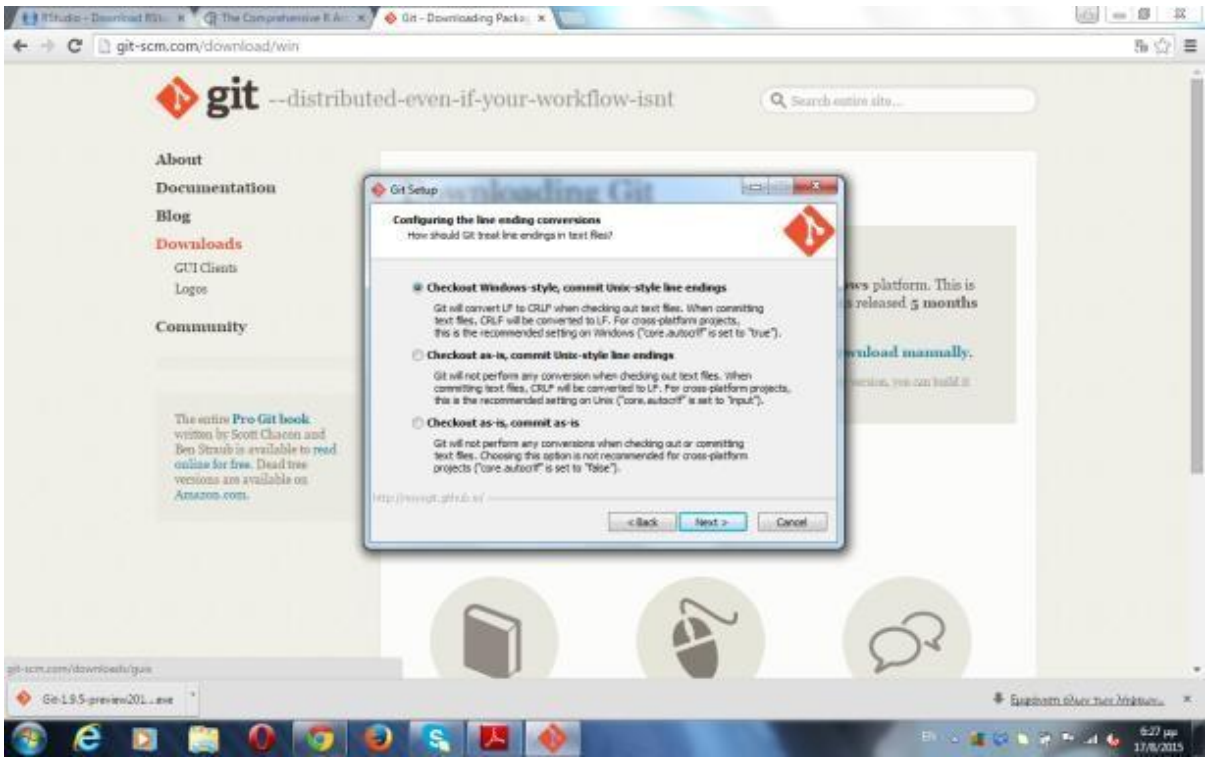

git-scm.com/download/win
git --local-branching-on-the-cheap
About
Documentation
Blog
Downloads
GUI Clients
Logos
Community
Downloading Git
Windows platform. This is
It was released 5 months
to download manually.
Now that you have downloaded Git, it's time to start using it.

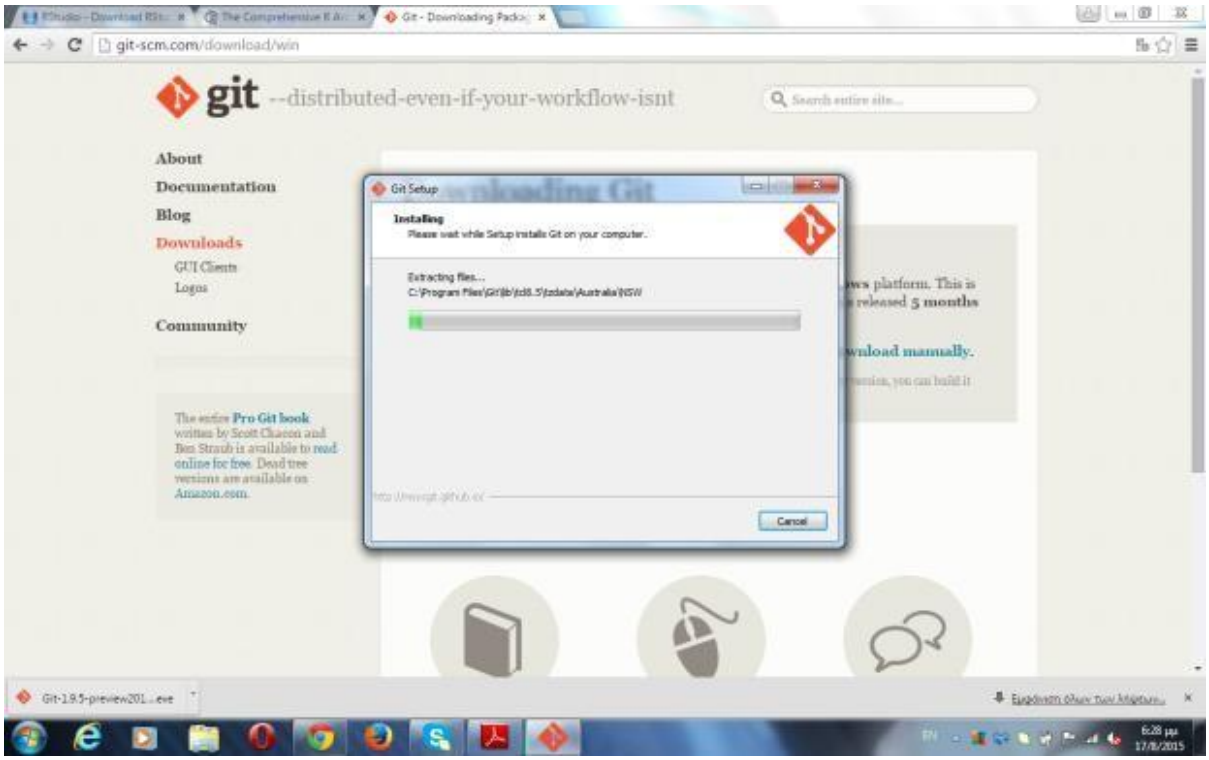

git-scm.com/download/win
git --distributed-even-if-your-workflow-isnt
About
Documentation
Blog
Downloads
GUI Clients
Logos
Community
Git Setup
Welcome to the Git Setup Wizard
Next >
Cancel


git-scm.com/download/win
git --distributed-even-if-your-workflow-isnt
About
Documentation
Blog
Downloads
GUI Clients
Logos
Community
Git Setup
Information
Please read the following important information before continuing.
When you are ready to continue with Setup, click Next.
GNU General Public License
Version 2, June 1991
Preamble
< Back
Next >
Cancel

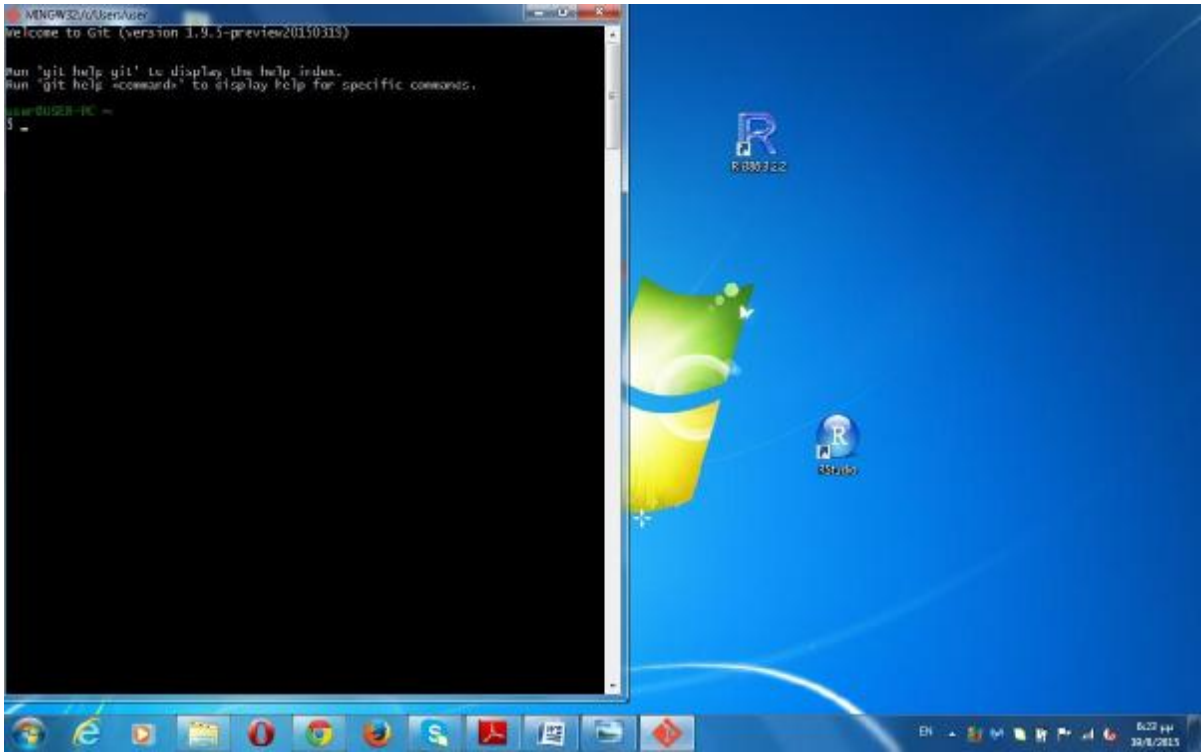

git-scm.com/download/win
git --distributed-even-if-your-workflow-isnt
About
Documentation
Blog
Downloads
GUI Clients
Logos
Community
Git Setup
Select Components
Which components should be installed?
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
Additional icons
On the Desktop
Windows Explorer integration
Git Bash Here
Git GUI Here
Associate .sh files to be run with Bash
Current selection requires at least 101,0 MB of disk space.
< Back
Next >
Cancel


git-scm.com/download/win
git --distributed-even-if-your-workflow-isnt
About
Documentation
Blog
Downloads
GUI Clients
Logos
Community
Git Setup
Adjusting your PATH environment
How would you like to use Git from the command line?
Use Git from Git Bash only
This is the safest choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.
Use Git from the Windows Command Prompt
This option is considered safe as it only adds some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from both Git Bash and the Windows Command Prompt.
Use Git and optional Unix tools from the Windows Command Prompt
Both Git and the optional Unix tools will be added to your PATH.
Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.
< Back
Next >
Cancel


git-scm.com/download/win
git --distributed-even-if-your-workflow-isnt
About
Documentation
Blog
Downloads
GUI Clients
Logos
Community
Git Setup
Configuring the line ending conversions
How should Git treat line endings in text files?
Checkout Windows-style, commit Unix-style line endings
Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").
Checkout as-is, commit Unix-style line endings
Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").
Checkout as-is, commit as-is
Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").
< Back
Next >
Cancel


git-scm.com/download/win
git --distributed-even-if-your-workflow-isnt
About
Documentation
Blog
Downloads
GUI Clients
Logos
Community
Git Setup
Installing
Please wait while Setup installs Git on your computer.
Extracting files...
Cancel


git-scm.com/download/win
git --distributed-even-if-your-workflow-isnt
About
Documentation
Blog
Downloads
GUI Clients
Logos
Community
Git Setup
Completing the Git Setup Wizard
Setup has finished installing Git on your computer. The application may be launched by selecting the installed icons.
Click Finish to exit Setup.
View ReleaseNotes.rtf
git
Finish



Η εγκατάσταση του Git ολοκληρώθηκε.

Κεφάλαιο 2: R Programming

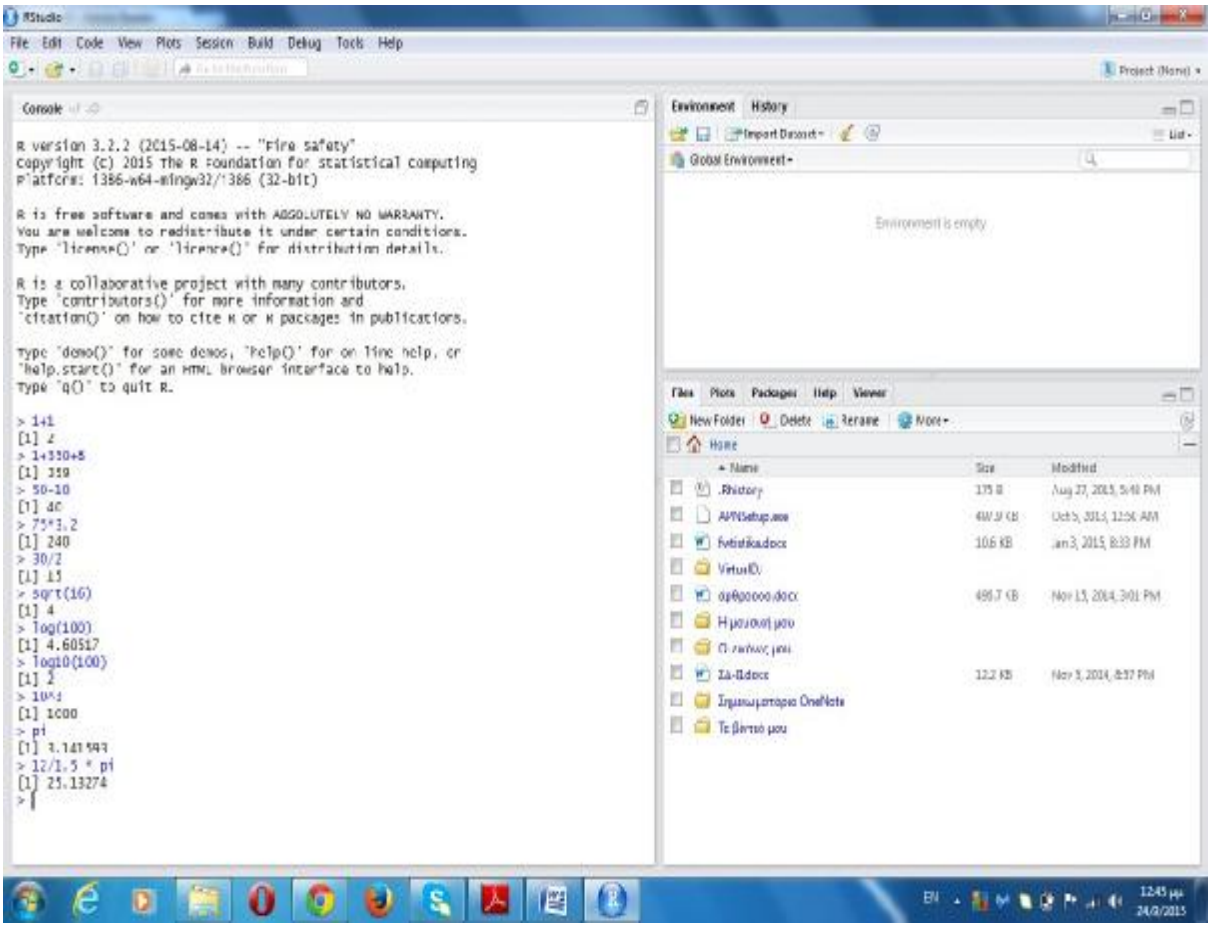
2.1 Εισαγωγικά

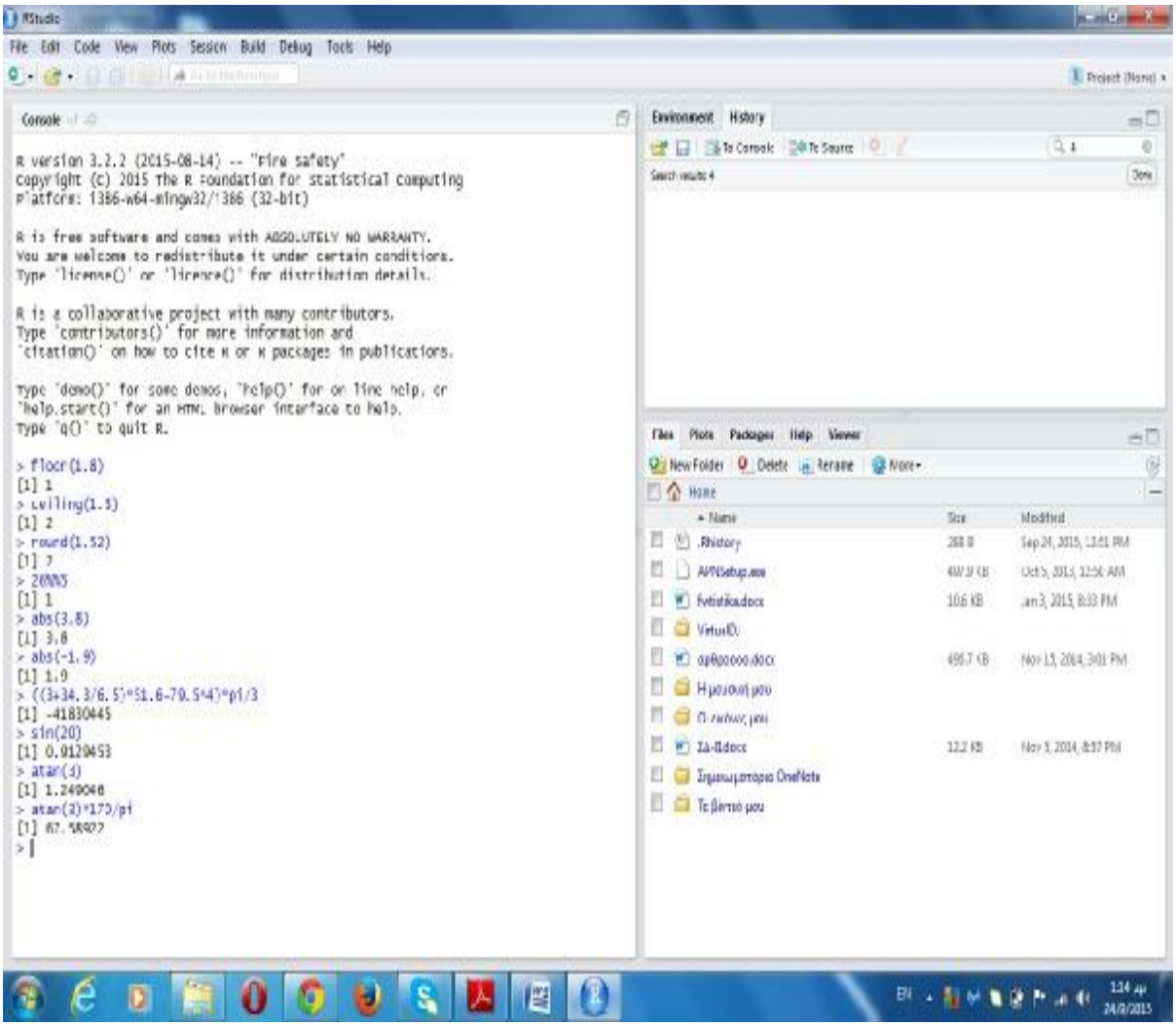
Αφού έχει γίνει με επιτυχία η εγκατάσταση των εργαλείων λογισμικού για χρήση της R μπορούμε π.χ. τη κονσόλα του Rstudio να τη χρησιμοποιήσουμε ως κομπιουτεράκι κάνοντας πράξεις. Ακολουθεί ένας πίνακας όπου φαίνονται τα σύμβολα που μπορούμε να χρησιμοποιήσουμε:

Αριθμητικοί και λογικοί τελεστές	
+	πρόσθεση
-	αφαίρεση
*	πολλαπλασιασμός
/	διαίρεση
b	ύψωση σε δύναμη
%%	υπόλοιπο διαιρέσεως
%/%	ακέραιο μέρος διαιρέσεως
==	λογική ισότητα
! =	λογική ανισότητα
>	λογικό μεγαλύτερο
<	λογικό μικρότερο
>=	λογικό μεγαλύτερο ή ίσον
<=	λογικό μικρότερο ή ίσον
|	λογικό είτε (OR) για διανύσματα
&	λογικό και (AND) για διανύσματα
||	λογικό είτε (OR) για στοιχεία
&&	λογικό και (AND) για στοιχεία

Όπως και στο κομπιουτεράκι μπορούμε να κάνουμε διάφορες πράξεις δηλαδή πρόσθεση, αφαίρεση, πολλαπλασιασμό και διαίρεση με τα ανάλογα σύμβολα που

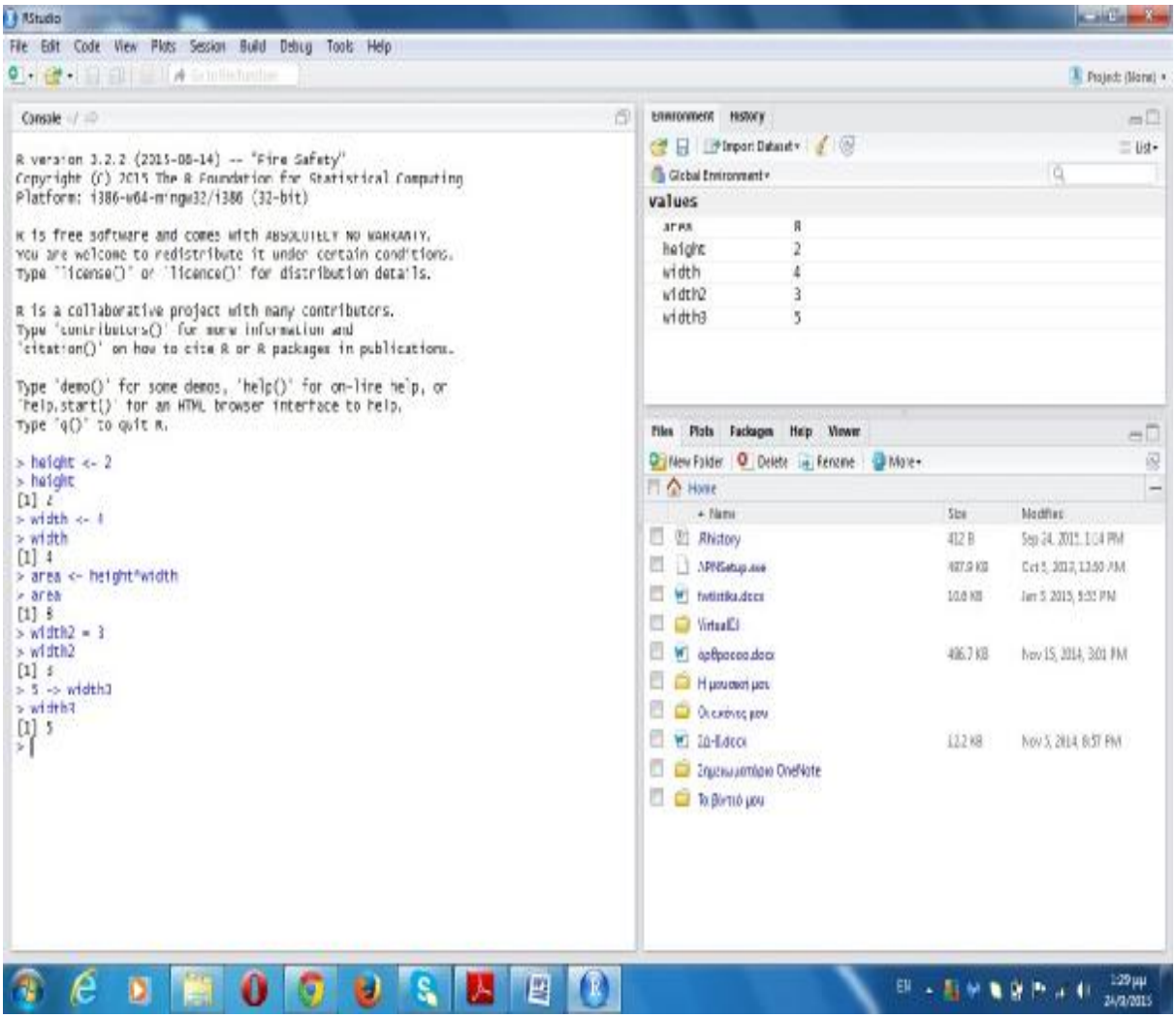
βλέπουμε παραπάνω (+,-,*,/). Μπορούμε ακόμα να χρησιμοποιήσουμε τις γνωστές συναρτήσεις, για παράδειγμα τετραγωνική ρίζα, sqrt(), square root, τη λογαριθμική συνάρτηση, log() και log10(), η πρώτη για φυσικούς λογαρίθμους και η δεύτερη για δεκαδικούς. Επίσης η ύψωση σε δύναμη, ^. Ο αριθμός π, (pi) με αρκετά δεκαδικά ψηφία υπάρχει διαθέσιμος και μπορούμε να τον χρησιμοποιούμε στις πράξεις μας. Αναφέρονται και οι συναρτήσεις για κόψιμο δεκαδικών, floor(), ή συμπλήρωση στον επόμενο ακέραιο, ceiling(), ή στρογγύλεμα, round(), είτε σε ακέραιο, είτε σε πραγματικό με επιθυμητό αριθμό δεκαδικών. Μπορούμε να ζητήσουμε περισσότερες πληροφορίες για κάποια συνάρτηση πληκτρολογώντας π.χ. ?round και μας εμφανίζονται πληροφορίες από το online manual. Υπάρχει και το υπόλοιπο διαιρέσεως, %%, καθώς και ο τρόπος υπολογισμού του ακεραίου μέρους, %/%. Τέλος είναι οι τριγωνομετρικές συναρτήσεις, μόνο που τα ορίσματα είναι σε rad, αλλά αν θέλουμε μοίρες κάνουμε την κατάλληλη μετατροπή.



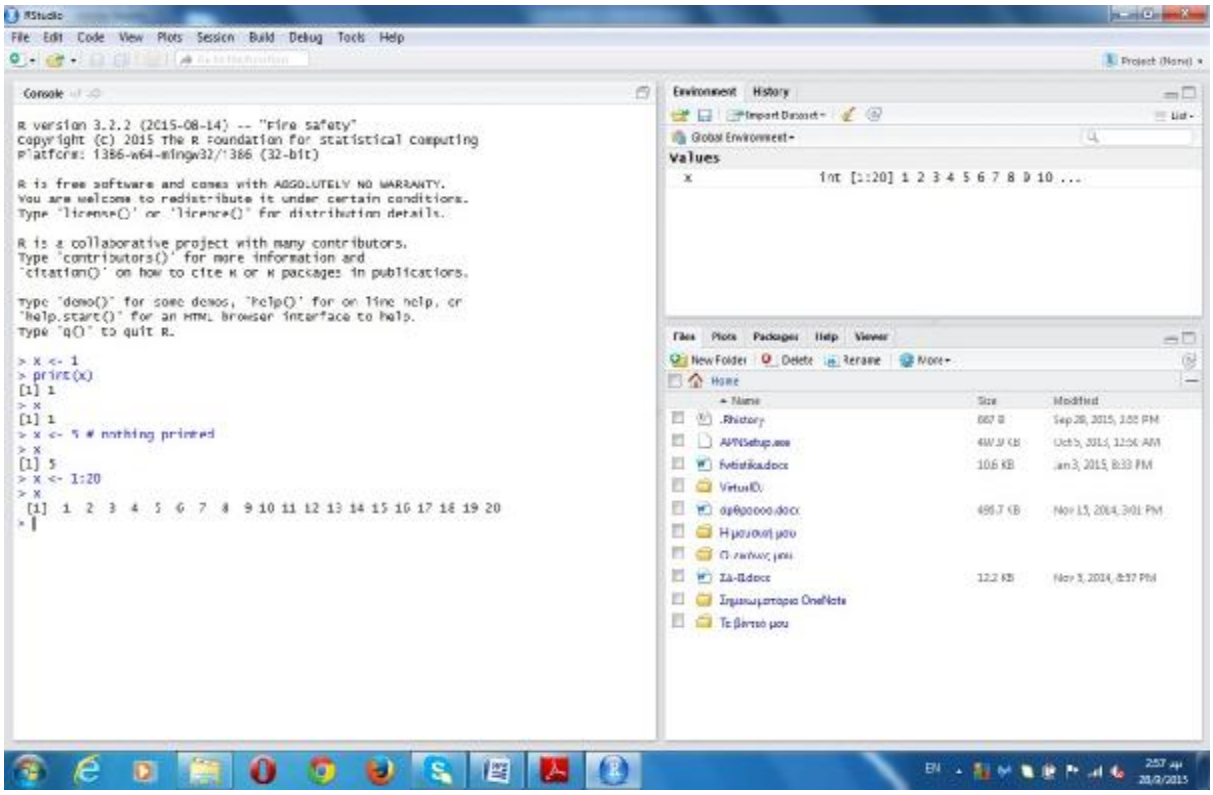


Στις παραπάνω φωτογραφίες γίνεται μια εφαρμογή στη κονσόλα του rstudio, όσων έχουν αναφερθεί παραπάνω. Δηλαδή γίνονται διάφορες πράξεις χρησιμοποιώντας τα σύμβολα για να πάρουμε τη σωστή πληροφορία.

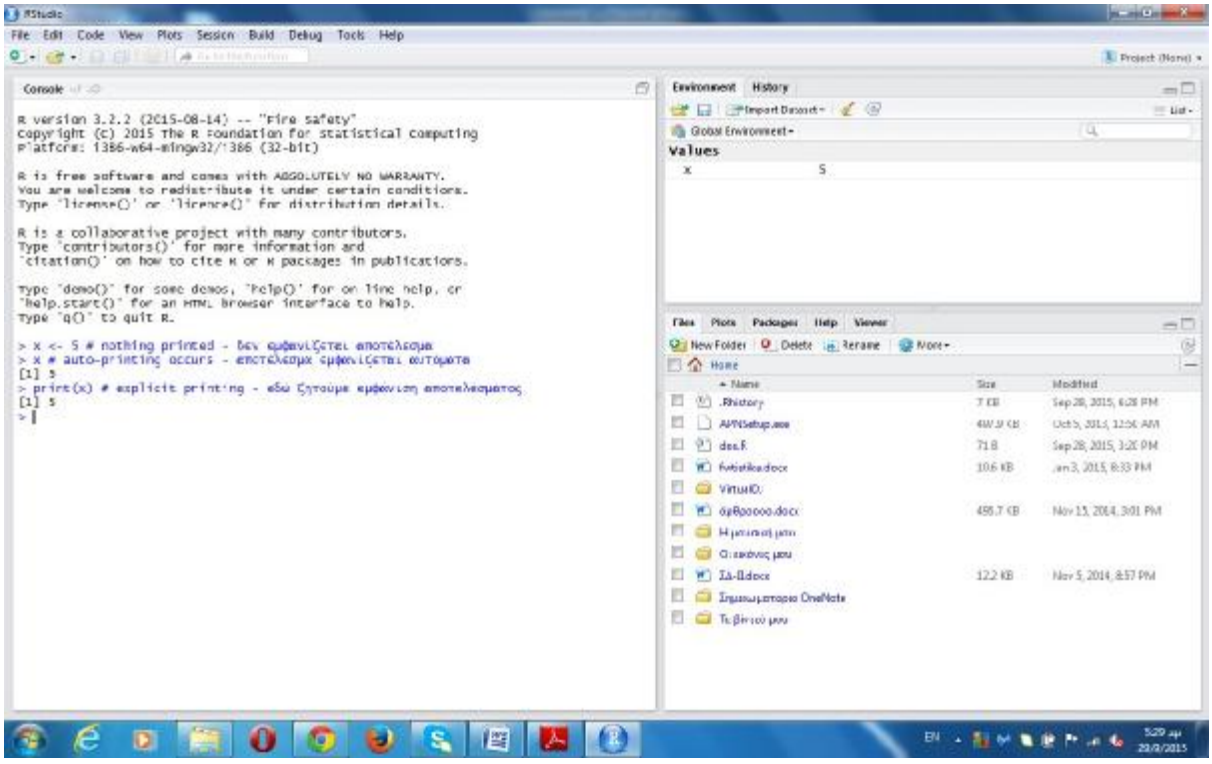
Επίσης μπορούμε να χρησιμοποιήσουμε τη κονσόλα για τη χρήση μεταβλητών με κατάλληλα ονόματα και οι μετέπειτα πράξεις να γίνονται με τις μεταβλητές. Δηλαδή ορίζουμε τις μεταβλητές, π.χ. height και width για ύψος και πλάτος ενός παραλληλογράμμου δίνοντάς τους τιμές με τον τελεστή <- και υπολογίζουμε το εμβαδόν, area. Τιμές σε μεταβλητές δίδονται από δεξιά στα αριστερά, όπως δείχνει το βελάκι. Παρατηρούμε επίσης ότι αν απλώς δηλώσουμε τη μεταβλητή δεν φαίνεται η τιμή της. Πρέπει να πληκτρολογήσουμε πάλι το όνομά της για να εμφανιστεί η τιμή της στην κονσόλα. Ο τελεστής <- είναι αντίστοιχος με το γνωστό = αλλά έτσι έχει επικρατήσει. Μπορεί επίσης να αντιστραφεί-> έτσι ώστε η τιμή αριστερά να δοθεί στη μεταβλητή δεξιά κάτι που δεν γίνεται με το =. Αυτό φαίνεται και από την παρακάτω φωτογραφία.



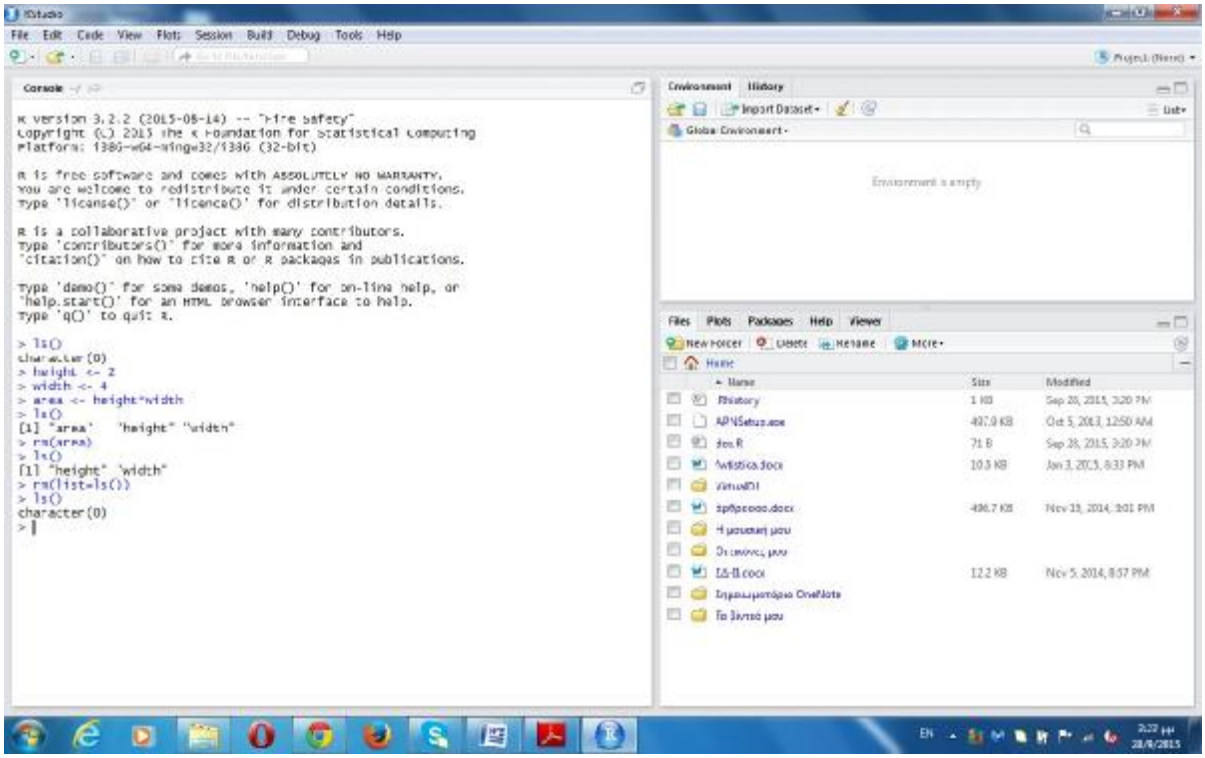
Επίσης με τη χρήση της συνάρτησης print() μπορεί να γίνει εκτύπωση αποτελεσμάτων ή τιμών μεταβλητών.



Όταν πληκτρολογήσουμε μια πλήρη εντολή, αυτή εκτελείται άμεσα και επιστρέφεται το αποτέλεσμα. Όπως φαίνεται και παρακάτω



Όπου το [1] σημαίνει ότι το αποτέλεσμα είναι διάνυσμα και το 5 είναι το πρώτο στοιχείο. Ο τελεστής που είδαμε παραπάνω σημαίνει τη δημιουργία ακολουθίας ακεραίων.



Όπως βλέπουμε στην παραπάνω εικόνα η ls() μας δείχνει λίστα με το τι μεταβλητές ή άλλα αντικείμενα βρίσκονται στο workspace. Μπορούμε να «σβήσουμε» μεταβλητές με τη συνάρτηση rm() όπου στο όρισμα έχουμε το όνομα της μεταβλητής. Μπορούμε επίσης να «σβήσουμε» και όλες τις μεταβλητές στον χώρο εργασίας.

2.1.1 Αντικείμενα (Objects)

Όλες οι μεταβλητές είναι αντικείμενα (objects) στη κονσόλα της R. Έχουμε πέντε βασικές (θεμελιώδεις) κλάσεις αντικειμένων:

- character - χαρακτήρες
- numeric (real numbers) - πραγματικοί αριθμοί
- integer - ακέραιοι
- complex - μιγαδικοί αριθμοί
- logical (True/False) - λογικές μεταβλητές (αληθή/ψευδή)

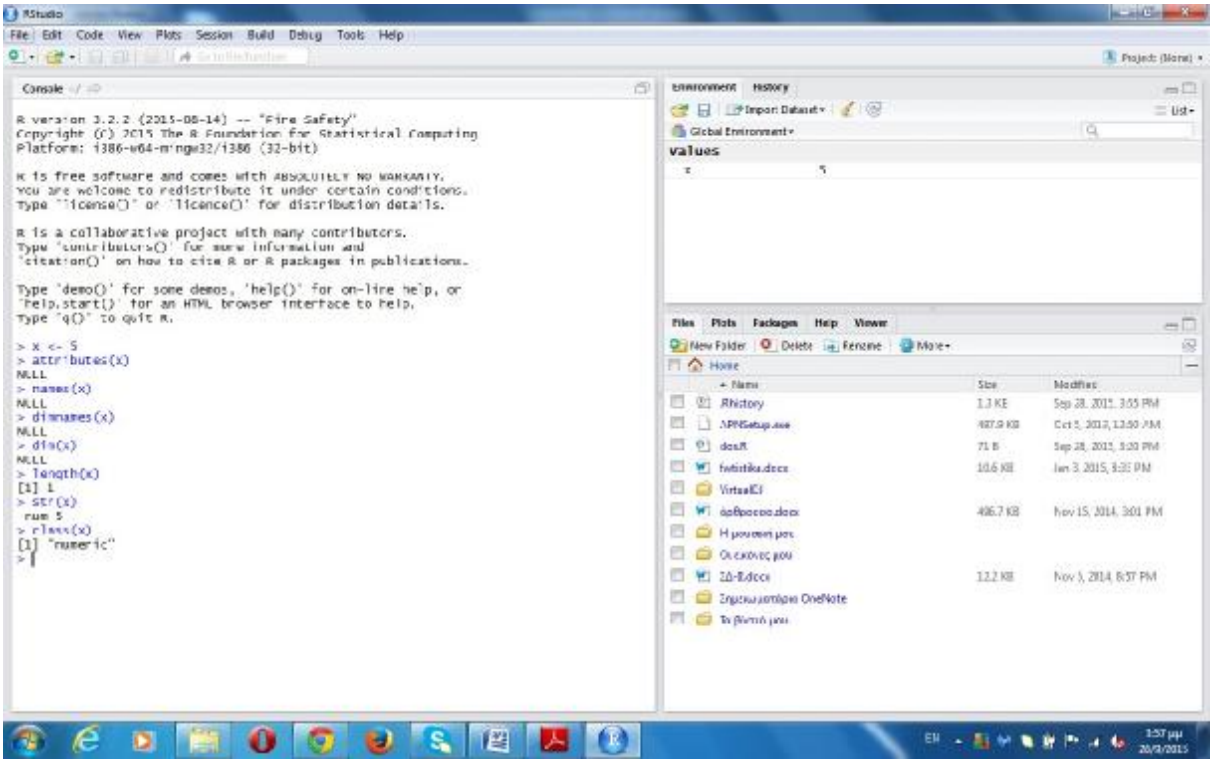
Επίσης όλα τα αντικείμενα είναι διανύσματα. Ας ξεκινήσουμε με διανύσματα με ένα στοιχείο (οι γνωστοί μας αριθμοί) και γενικεύουμε σε ακολουθίες αντικειμένων ίδιου τύπου/κλάσης σε μια διάσταση (vector) ή δυο (matrix). Η λίστα (list) γενικεύει το vector εφόσον μπορεί να περιέχει αντικείμενα διαφορετικών κλάσεων. Οι αριθμοί είναι αντικείμενα numeric (πραγματικοί αριθμοί διπλής ακριβείας - double precision).

Εάν δηλώνουμε ακέραιο το κάνουμε με το επίθεμα L, π.χ. 1L ο ακέραιος 1. Η R δέχεται και τα σύμβολα Inf (άπειρο), αποτέλεσμα της διαίρεσης με 0, καθώς και το NaN (Not a Number) μια απροσδιόριστη τιμή (π.χ. 0/0) ή τιμή που δεν υπάρχει (λείπει).

Τα αντικείμενα έχουν διάφορες χαρακτηριστικές ιδιότητες (attributes) όπως:

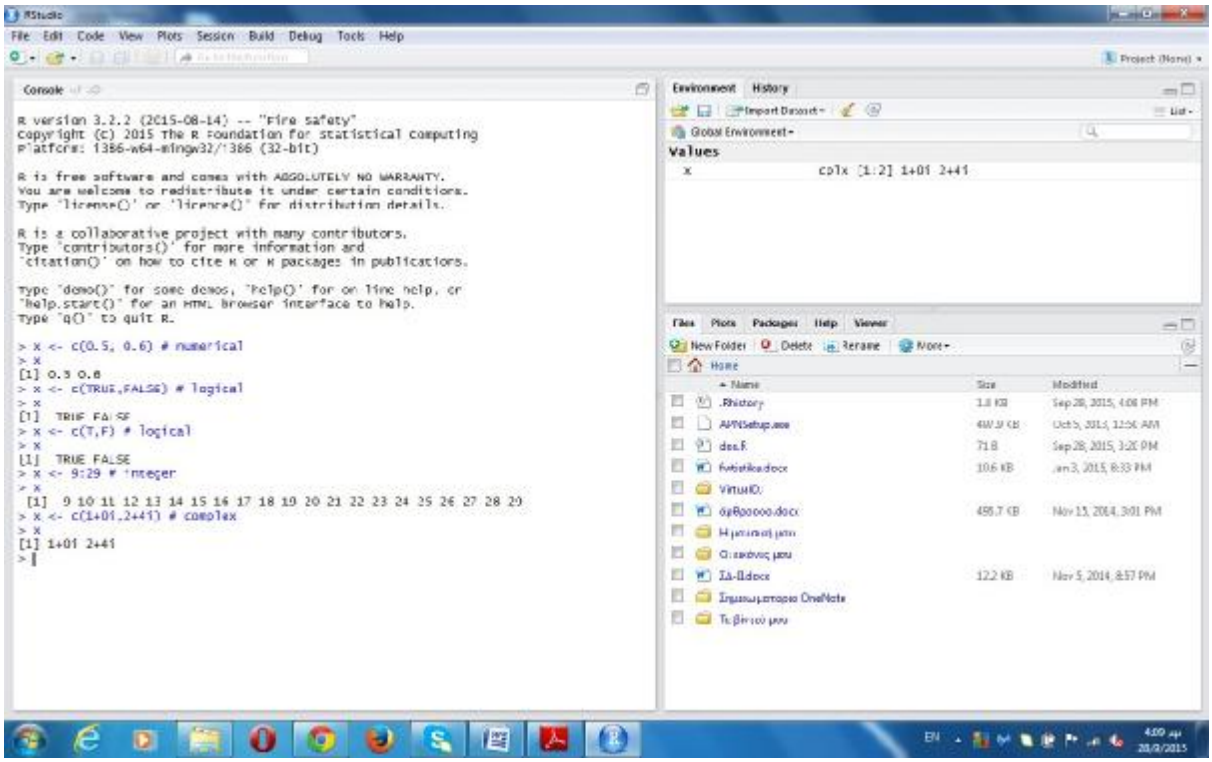
- names, dimnames (όνομα)
- dimensions (διαστάσεις) π.χ. για πίνακες (matrices, arrays)
- class (κλάση)
- length (μήκος)
- other (άλλες ιδιότητες)

Στη φωτογραφία που ακολουθεί θα δούμε ένα παράδειγμα με μια μεταβλητή x που έχει τιμή τον αριθμό 5. Επίσης βλέπουμε διάφορες συναρτήσεις που μας δίνουν πληροφορίες για το αντικείμενο x. Στο παρακάτω παράδειγμα φαίνεται πως η str() είναι πιο χρήσιμη και δηλώνει ότι το x είναι αντικείμενο numeric (num) με τιμή 5. Σε κάποια σημεία παίρνουμε τις απαντήσεις NULL που απλά σημαίνουν ότι το x δεν έχει ορισμένη αυτήν την ιδιότητα. Για πιο σύνθετα αντικείμενα οι απαντήσεις δεν θα είναι NULL.

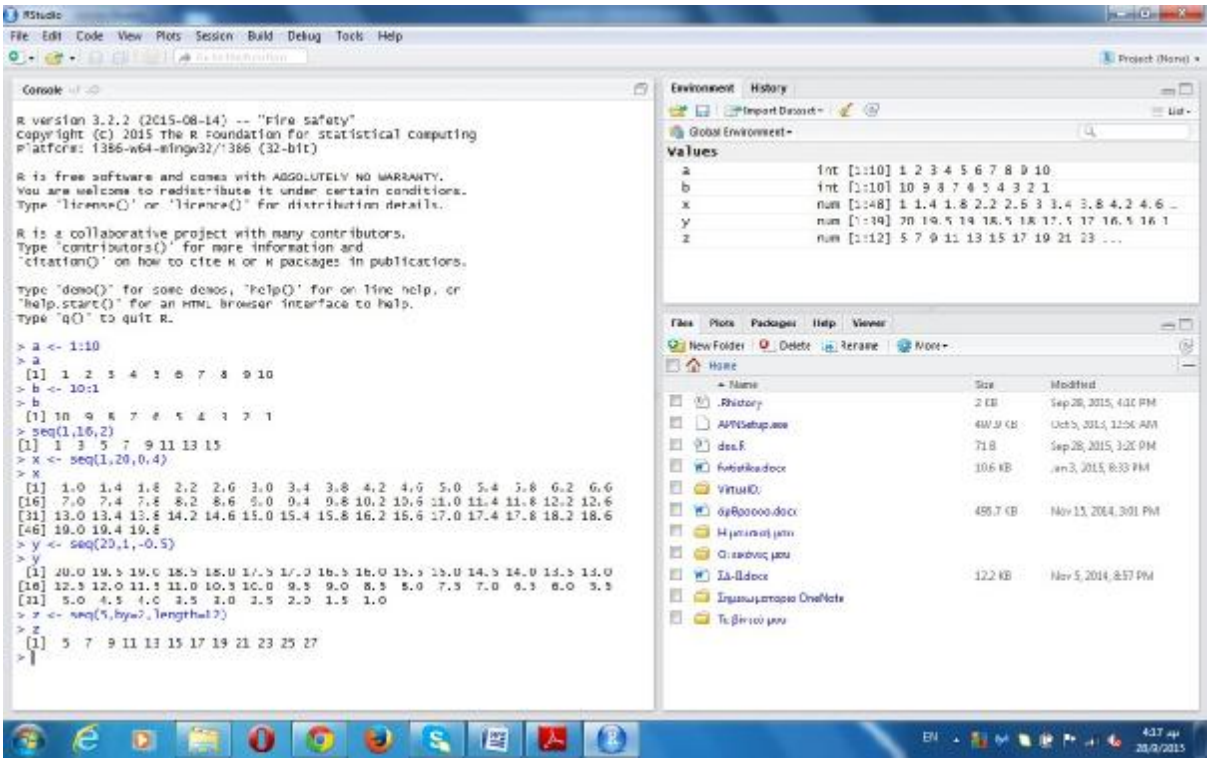


2.1.2 Δημιουργία διανυσμάτων

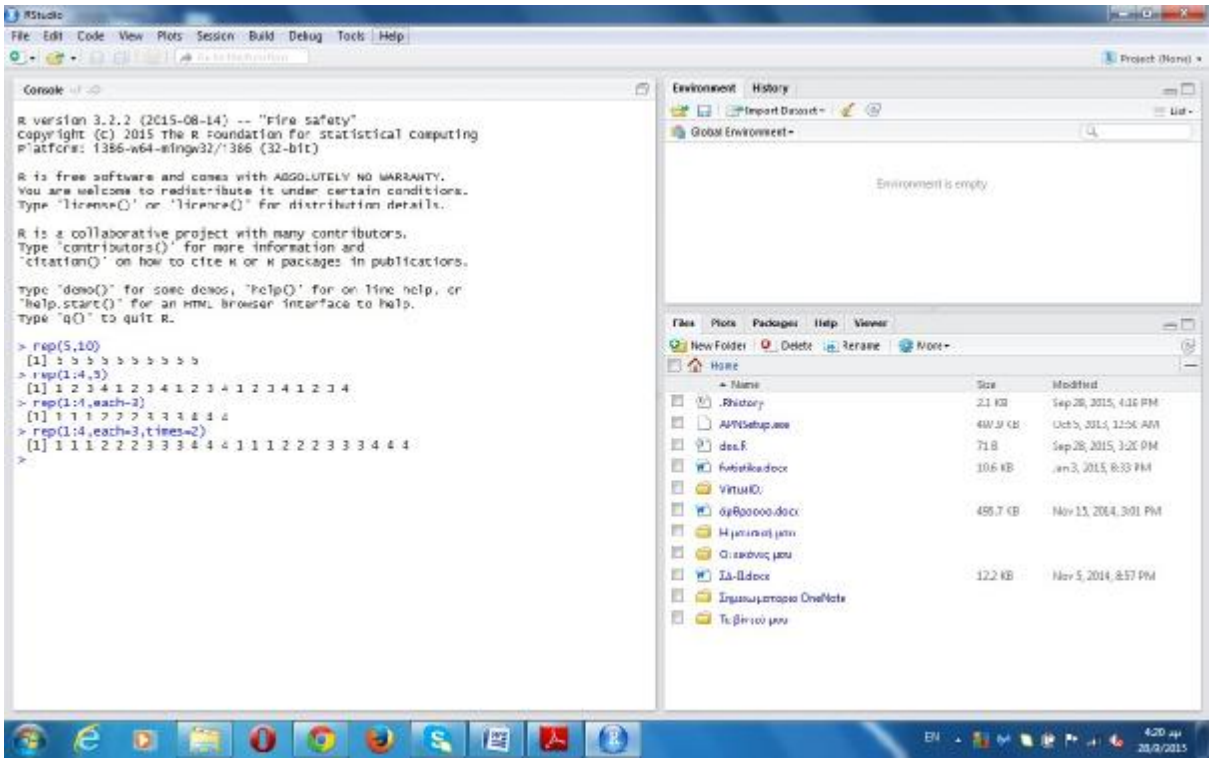
Η συνάρτηση c() (από τη λέξη combine) μπορεί να δημιουργήσει διανύσματα αντικειμένων, π.χ. διάφορα παραδείγματα μεταβλητών.



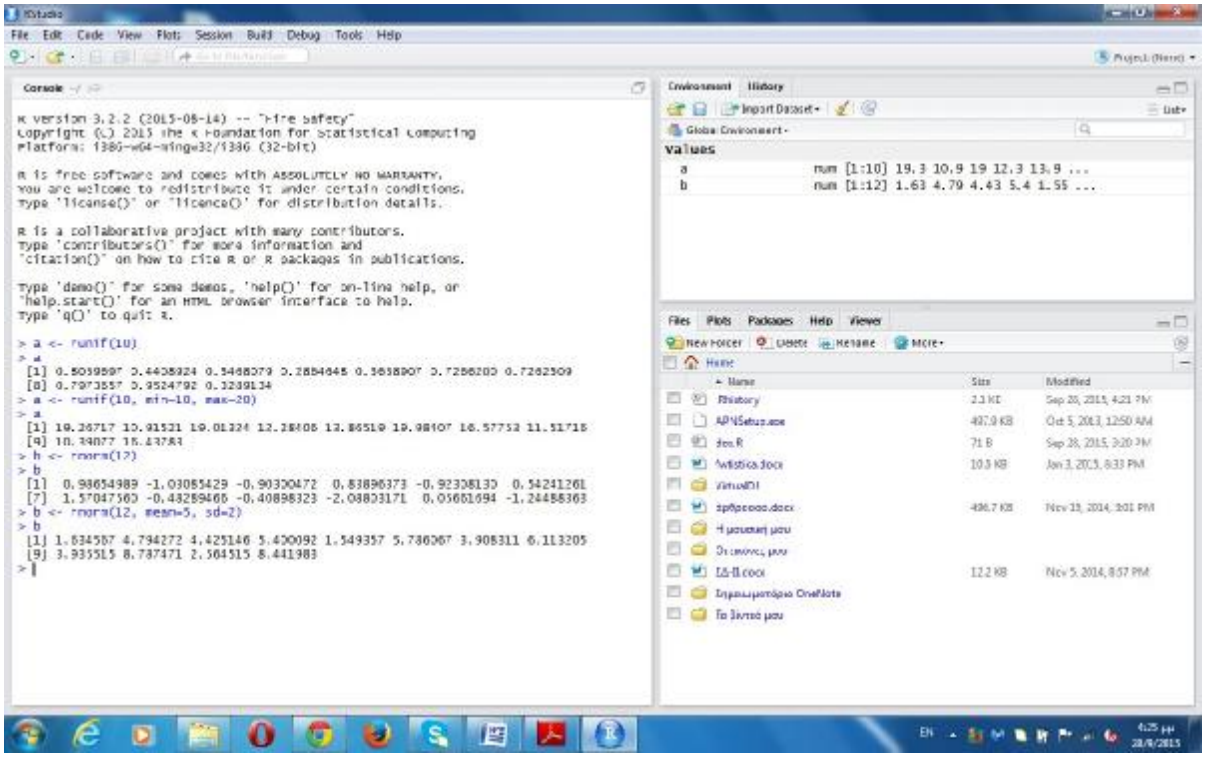
Είναι εύκολο να φτιάξουμε ακολουθίες αριθμών είτε με τον τελεστή : (για ακεραίους), είτε με την συνάρτηση seq() όπου ορίζουμε αρχικό στοιχείο, τελικό στοιχείο και βήμα (μπορεί να είναι και αρνητικό). Μπορούμε επίσης να δώσουμε μόνο αρχικό στοιχείο, βήμα και αριθμό στοιχείων στο διάνυσμα.



Όπως θα δούμε παρακάτω μπορούμε να επαναλάβουμε στοιχεία όσες φορές θέλουμε. Δηλαδή:

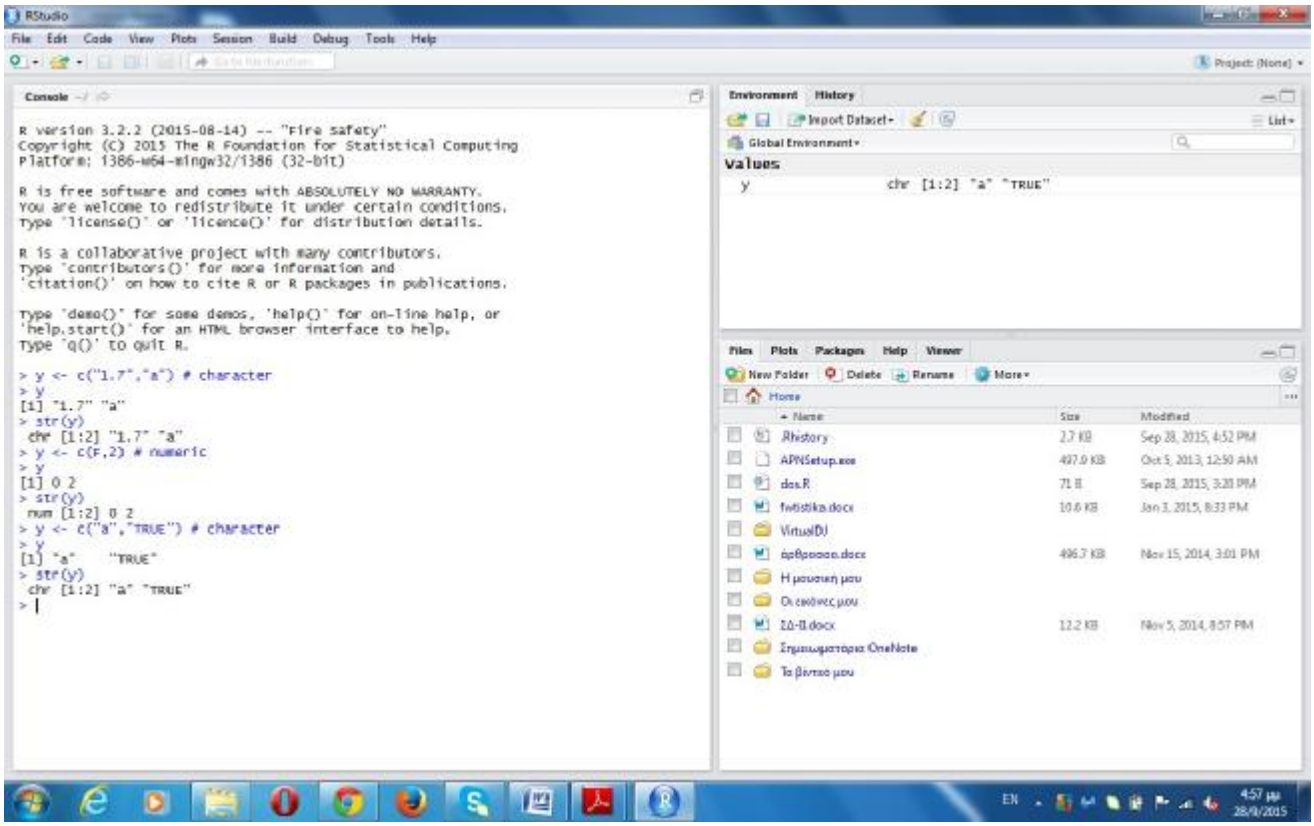


Μπορούμε επίσης να παράγουμε εύκολα ακολουθίες τυχαίων αριθμών. Βλέπουμε παρακάτω ακολουθίες από την ομοιόμορφη κατανομή, runif, μεταξύ 0 και 1 ή μεταξύ 10 και 20. Βλέπουμε επίσης και από την κανονική κατανομή με μέσο 0 και τυπική απόκλιση 1 ή με μέσο 5 και τυπική απόκλιση 2. Οι τυχαίοι αριθμοί που παράγονται είναι διαφορετικοί κάθε φορά που καλούμε τέτοιες συναρτήσεις επομένως οι αριθμοί θα είναι διαφορετικοί για σας, εκτός αν χρησιμοποιηθεί το ίδιο seed. Π.χ. set.seed(1).



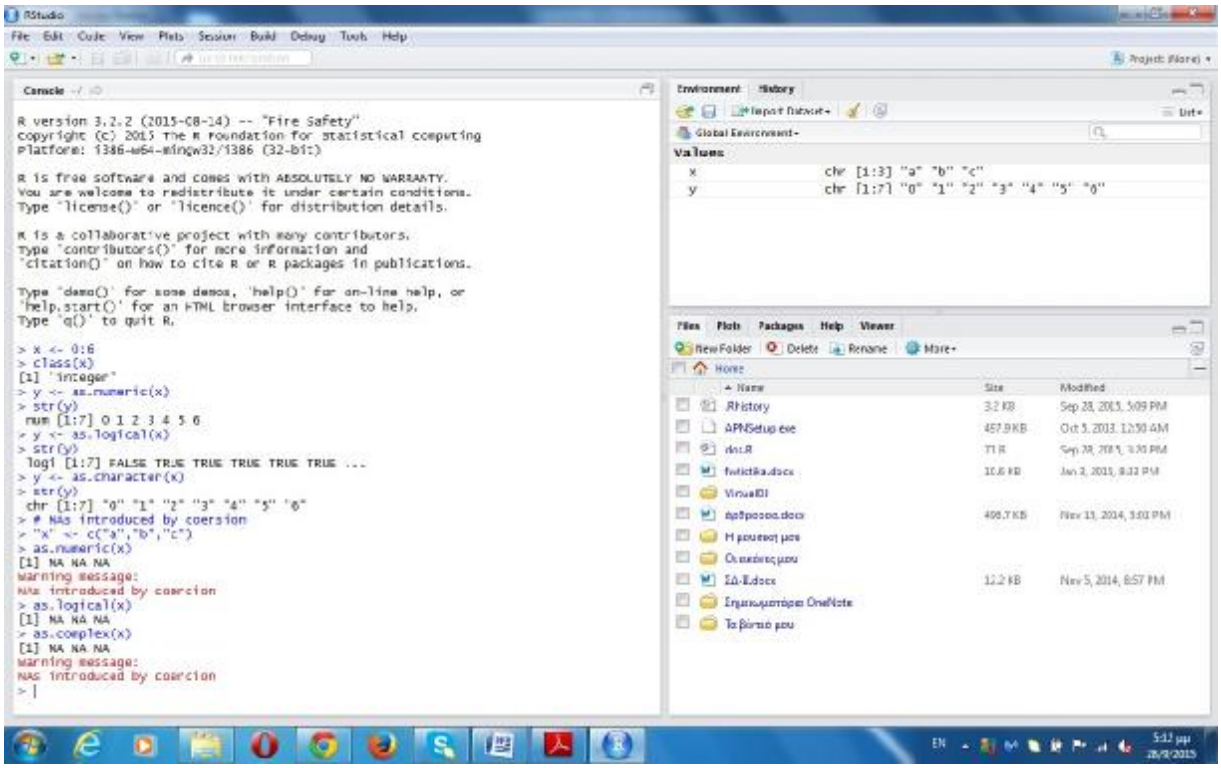
2.1.3 Μείξη αντικειμένων

Μπορούμε να ανακατέψουμε αντικείμενα διαφορετικών κλάσεων σε διάνυσμα όπου γίνεται αυτόματη (έμμεση) μετατροπή (implicit coercion) έτσι ώστε το διάνυσμα να αποτελείται από αντικείμενα της ίδιας κλάσης. Αυτό φαίνεται στην παρακάτω εικόνα.



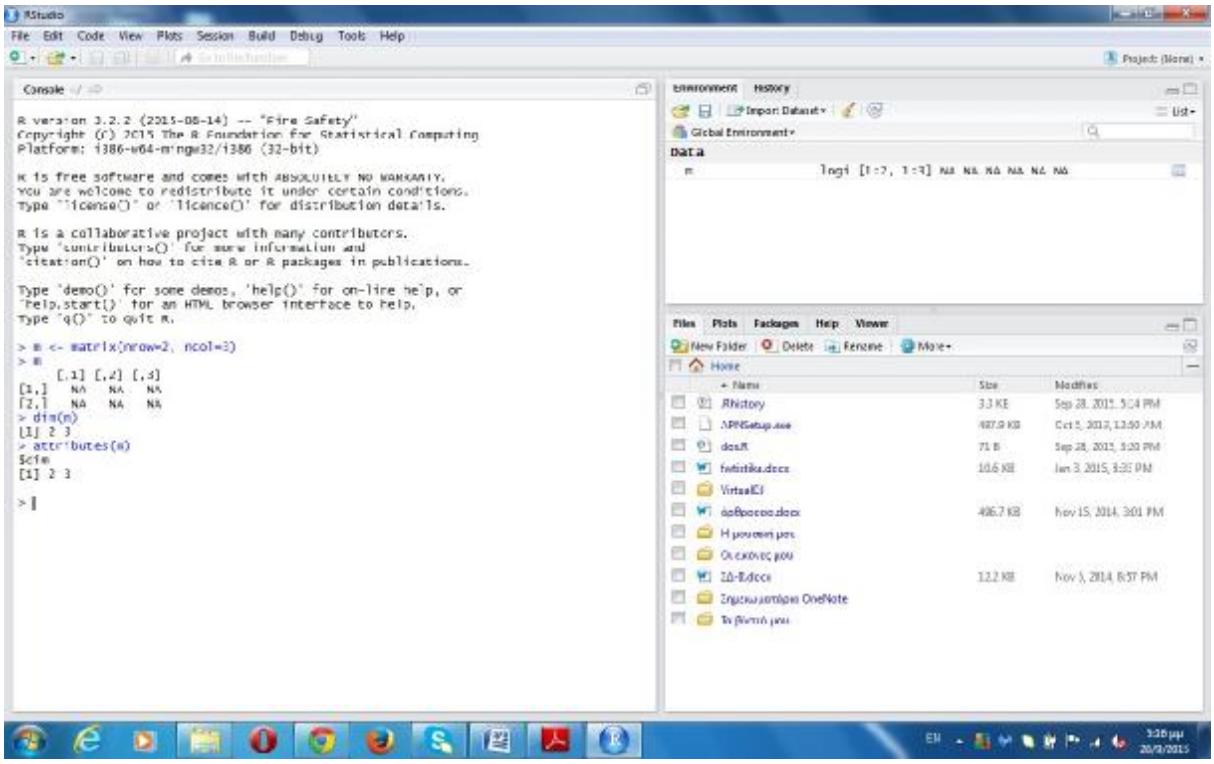
2.1.4 Άμεση μετατροπή (explicit coercion)

Μπορούμε να μετατρέψουμε αντικείμενα από μια κλάση σε άλλη με τις συναρτήσεις as.* όπου αυτό θα μας χρειαστεί. Στην εικόνα που ακολουθεί βλέπουμε τη μετατροπή διανυσμάτων σε διαφορετικές κλάσεις, και, ότι σε περιπτώσεις που δεν μπορεί να γίνει η μετατροπή το αποτέλεσμα μετατρέπεται σε NA.

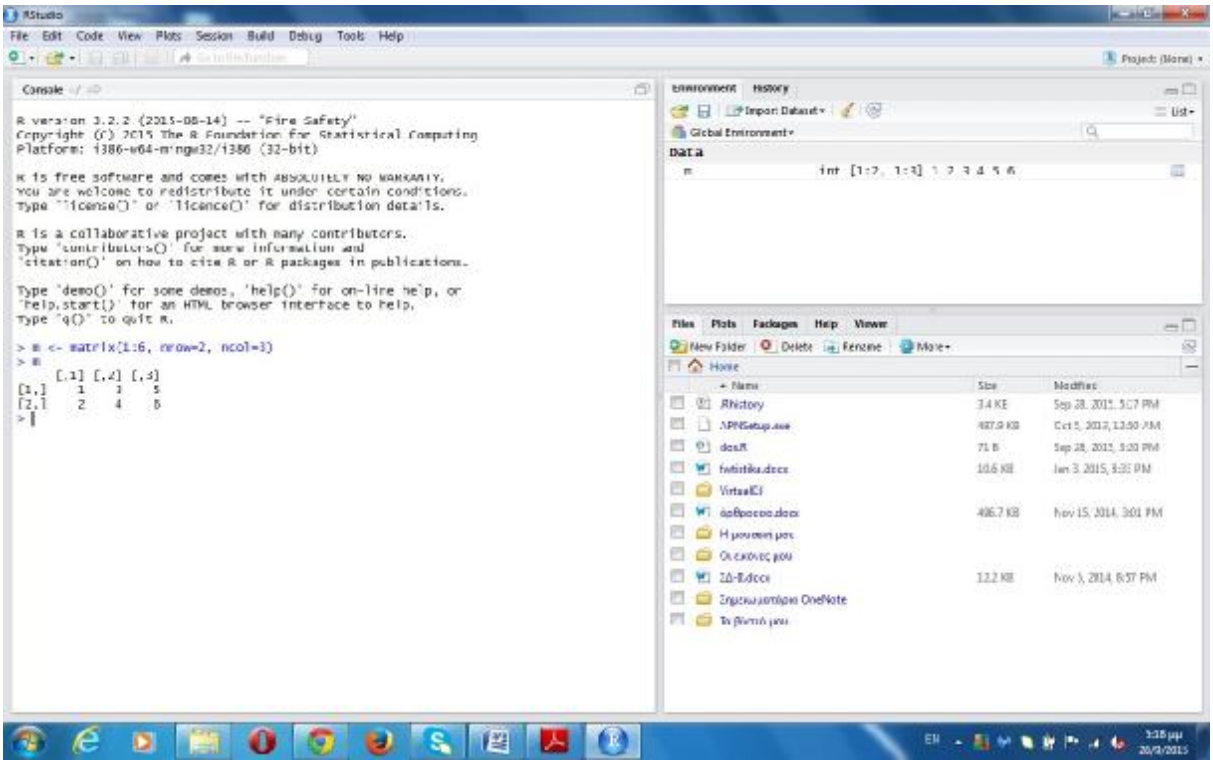


2.1.5 Πίνακες (Matrices)

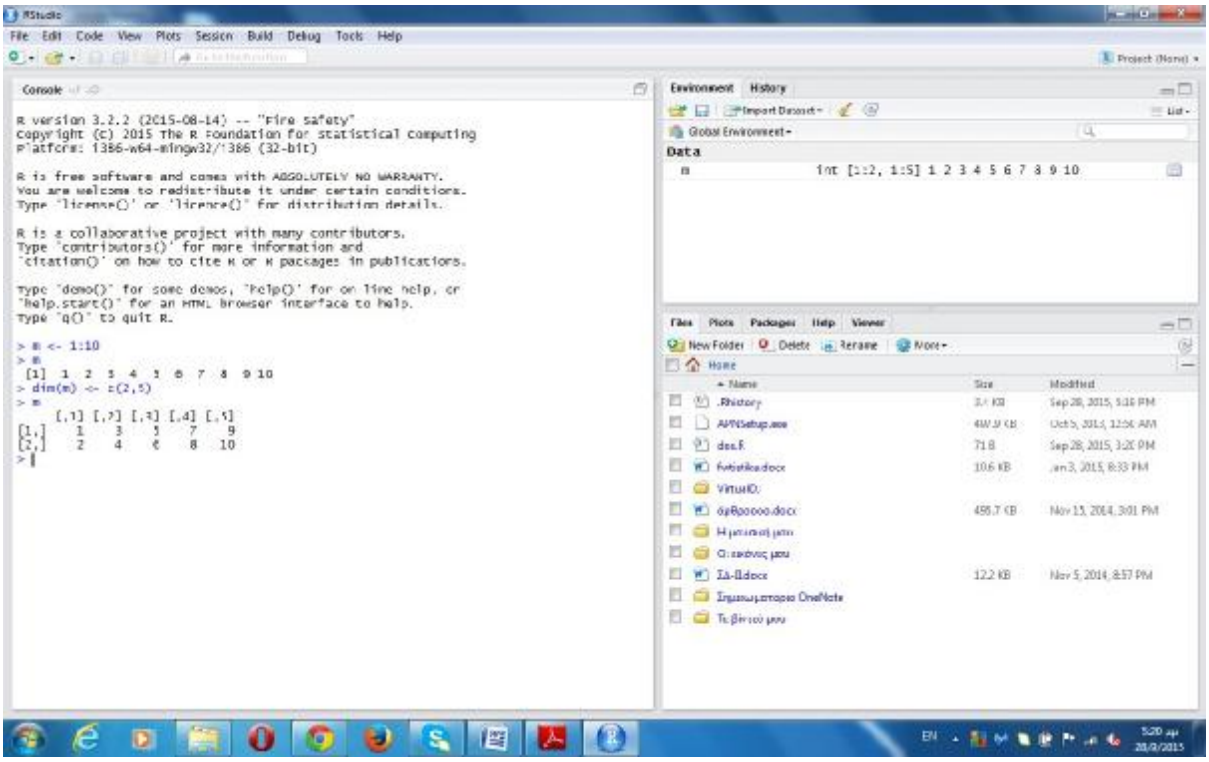
Οι πίνακες γενικεύουν την έννοια του διανύσματος στις 2 διαστάσεις. Η χαρακτηριστική ιδιότητα dimensions (διαστάσεις) εδώ έχει νόημα. Βλέπουμε τη δημιουργία ενός άδειου πίνακα και τις εφαρμογές των dim και attributes συναρτήσεων:



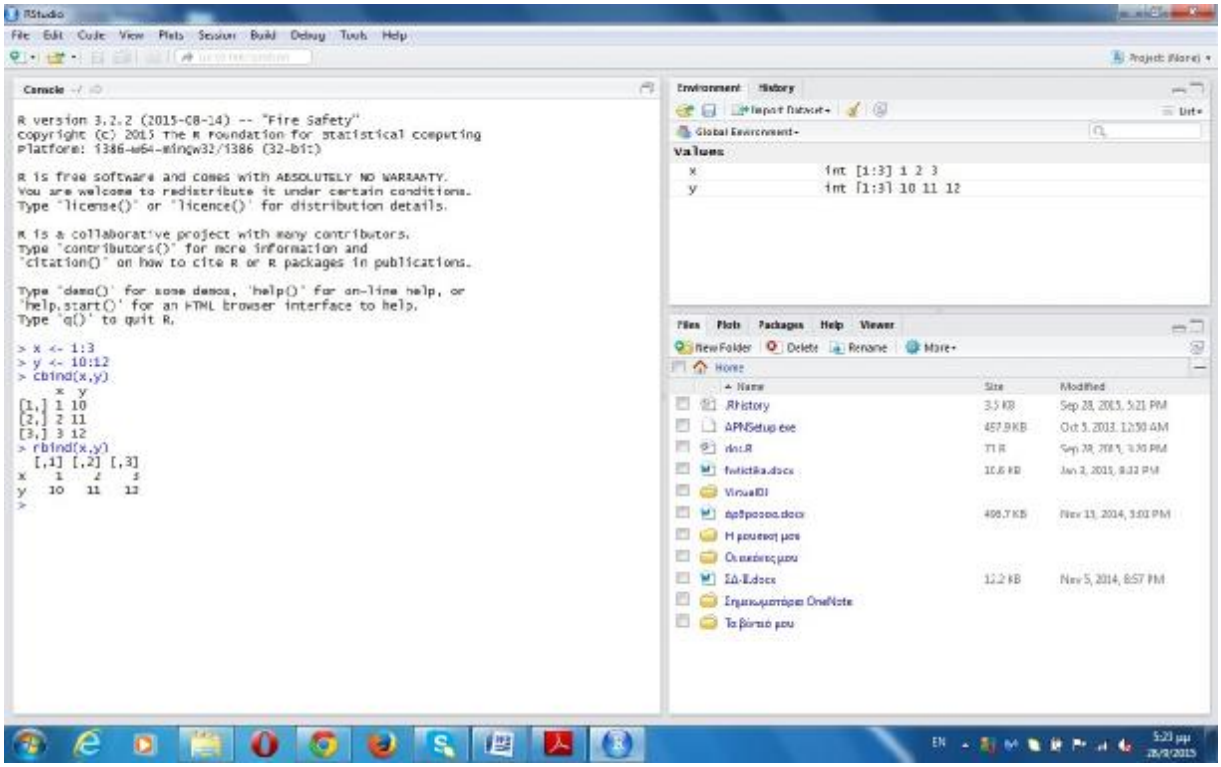
Μπορούμε να φτιάξουμε έναν πίνακα από διάνυσμα γεμίζοντας στήλες:



ή απλώς ορίζοντας τη χαρακτηριστική ιδιότητα διαστάσεων:

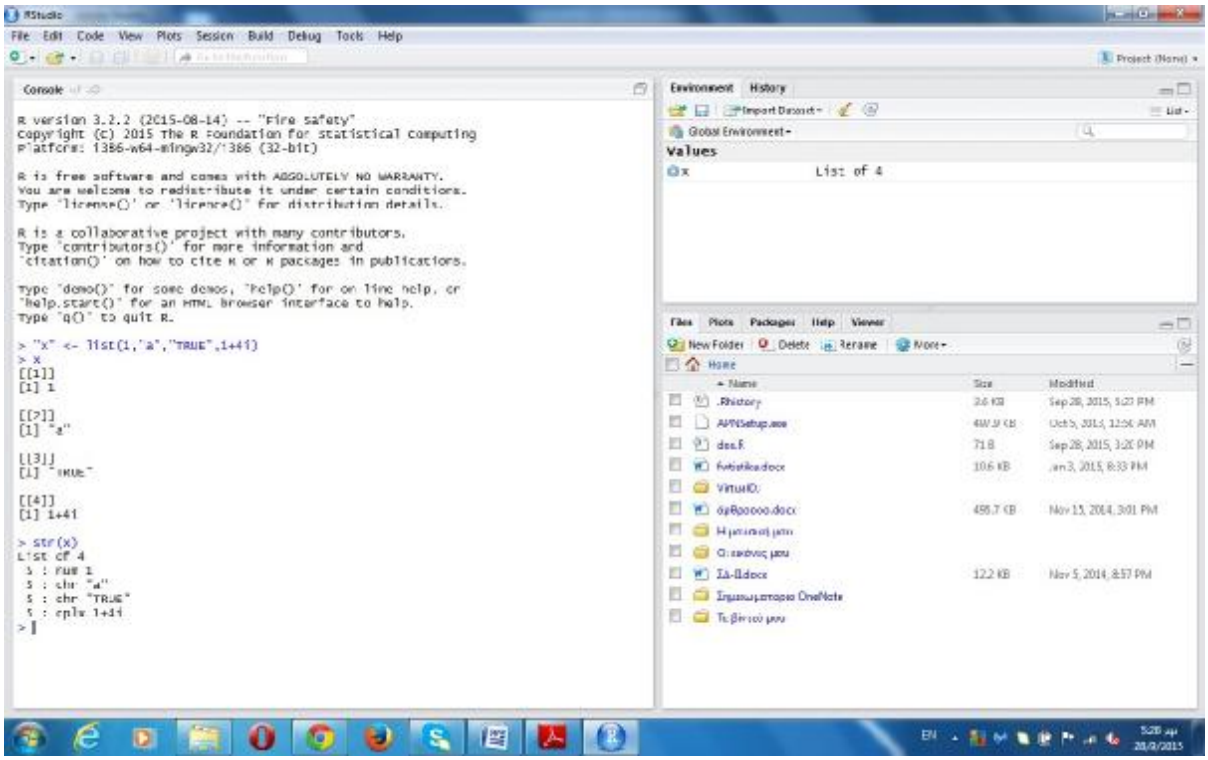


Εναλλακτικά, μπορούμε να ενώσουμε διανύσματα γραμμής ή στήλης:



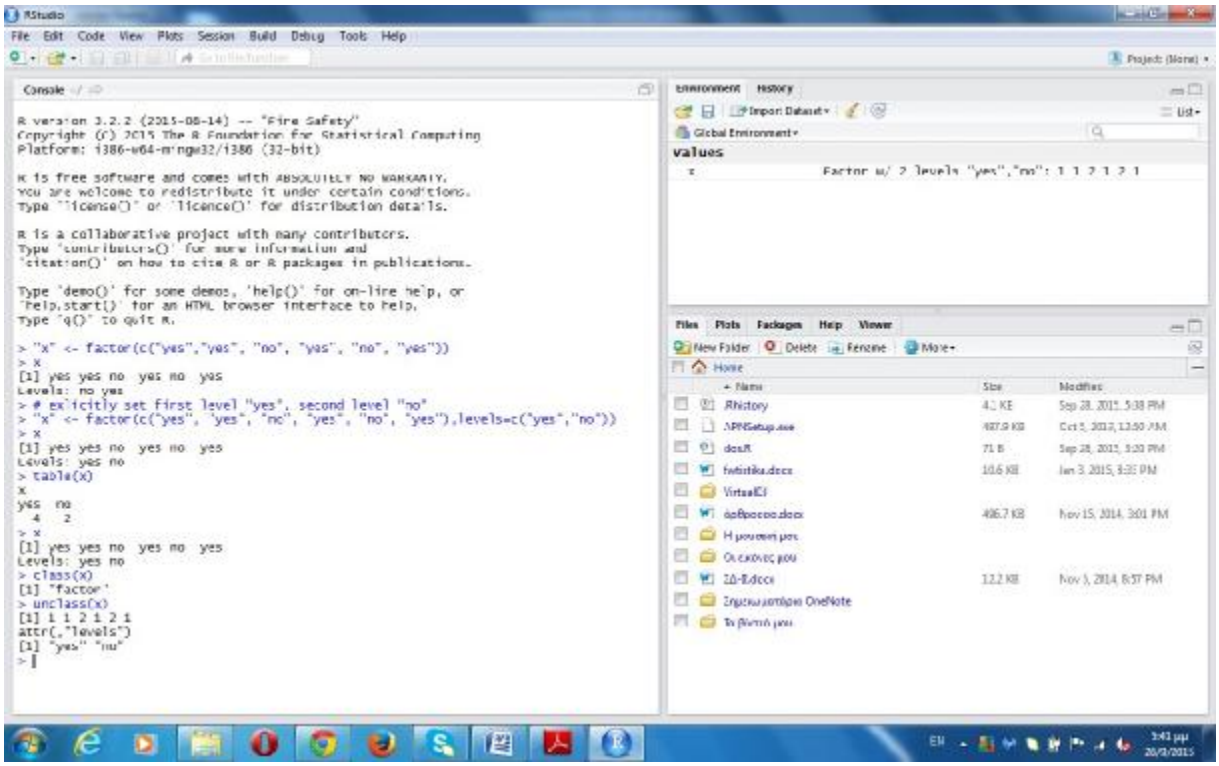
2.1.6 Λίστες (Lists)

Οι λίστες χρησιμοποιούνται ευρέως στη κονσόλα της R. Είναι σαν τα διανύσματα αλλά πιο γενικά αντικείμενα αφού μπορούν να περιέχουν αντικείμενα διαφορετικών κλάσεων. Σαν στοιχεία σε λίστες μπορούμε να έχουμε οποιοδήποτε αντικείμενο, διανύσματα, πίνακες, ακόμα και άλλες λίστες.



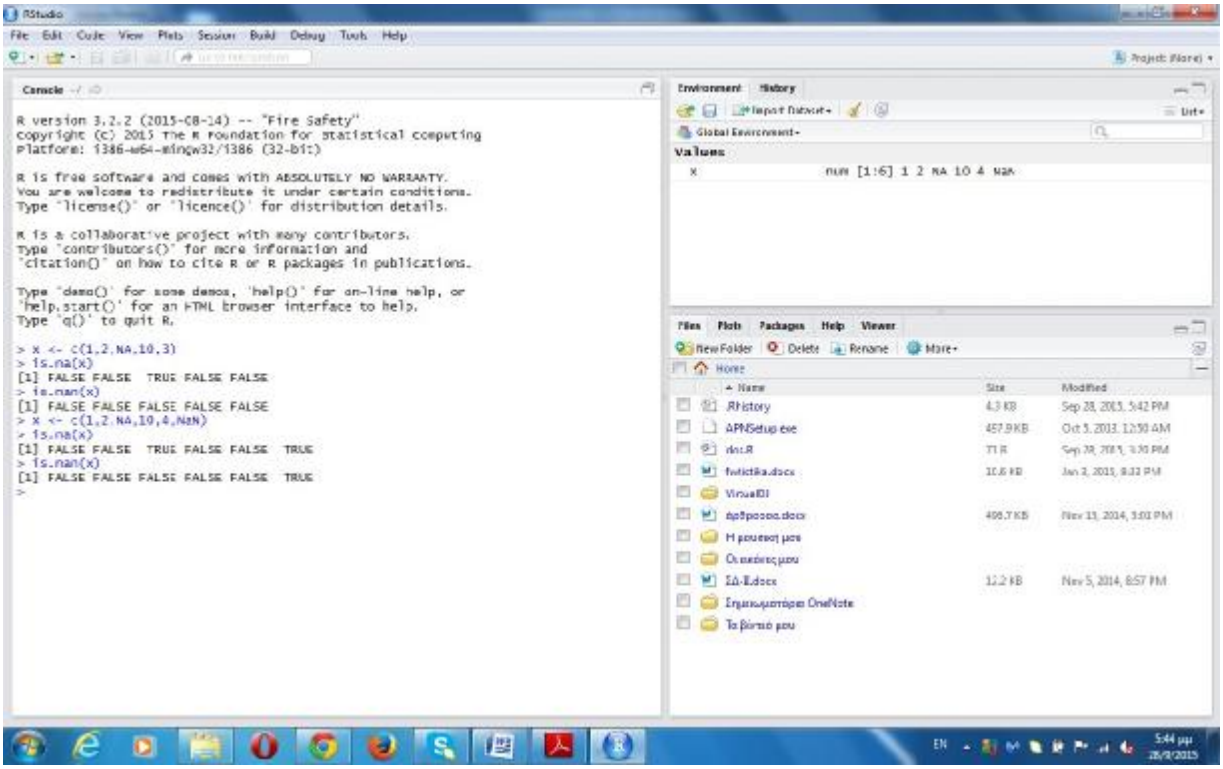
2.1.7 Factors

Τα factors είναι αντικείμενα που εκφράζουν δεδομένα κατηγοριών. Στην ουσία είναι ακέραια διανύσματα μόνο που αντί για αριθμό έχουν όνομα. Είναι χρήσιμα σε ανάλυση δεδομένων όπου π.χ. για κατηγορίες Άρρεν, Θήλυ είναι καλύτερα και πιο ξεκάθαρο να έχουμε τα ονόματα Άρρεν, Θήλυ παρά ακέραιους. Τα factors χρησιμοποιούνται με ιδιαίτερο τρόπο από συναρτήσεις μοντέλων όπως lm και glm (γραμμικά μοντέλα).



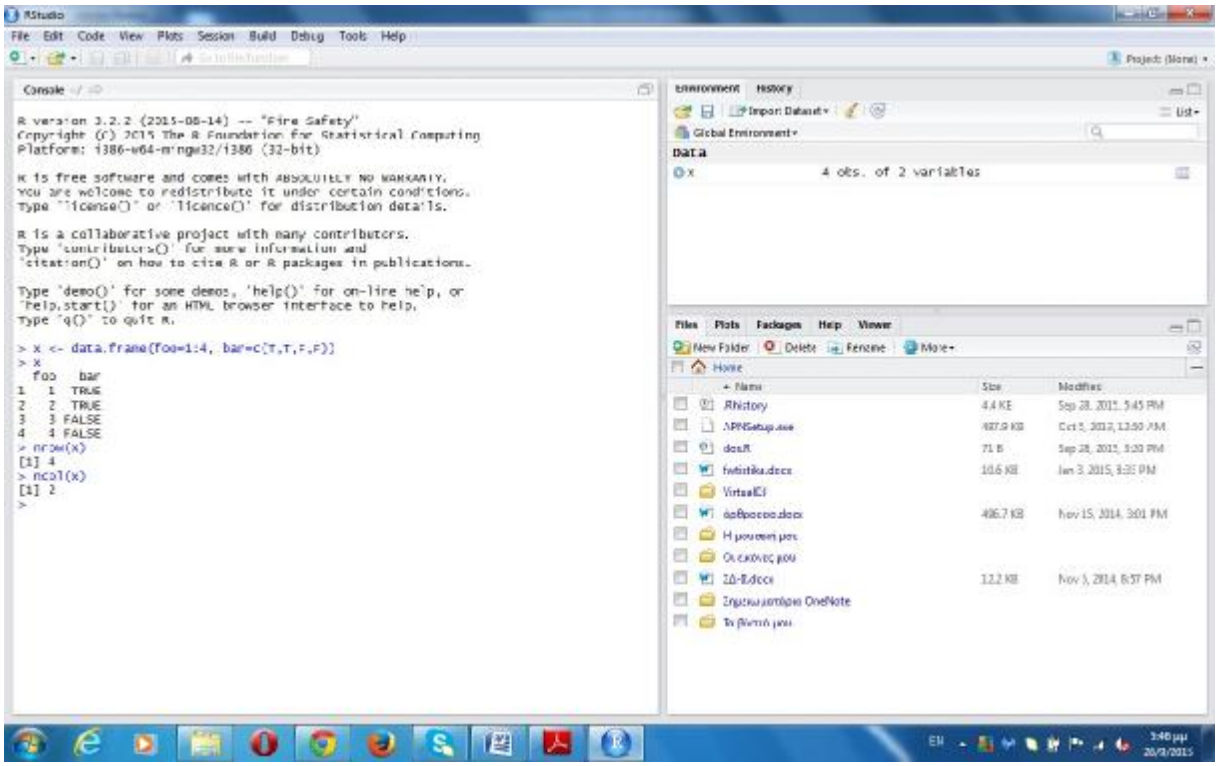
2.1.8 Τιμές που λείπουν (missing values)

Πολλές φορές σε δεδομένα θα έχουμε τιμές που λείπουν, NA ή τιμές απροσδιόριστες, NaN. Οι συναρτήσεις is.na() ή is.nan() χρησιμοποιούνται για να προσδιορίσουμε αυτές τις τιμές.

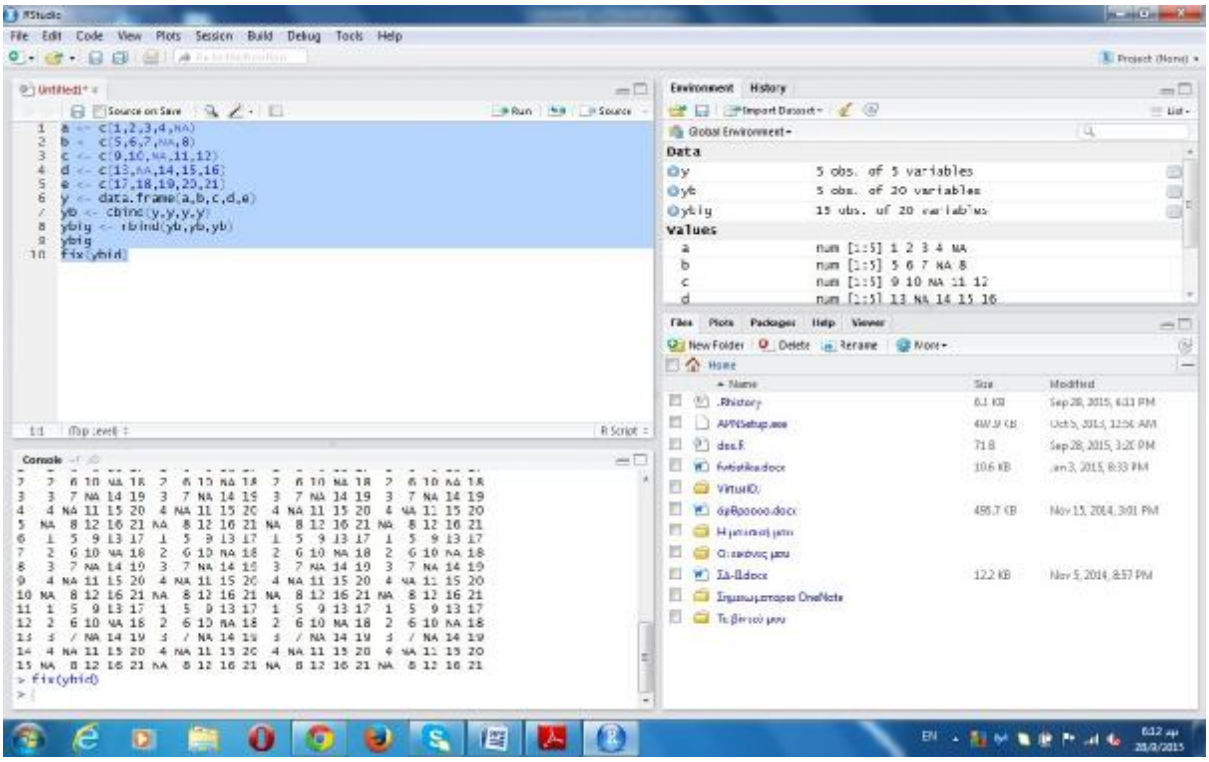


2.1.9 Data frames

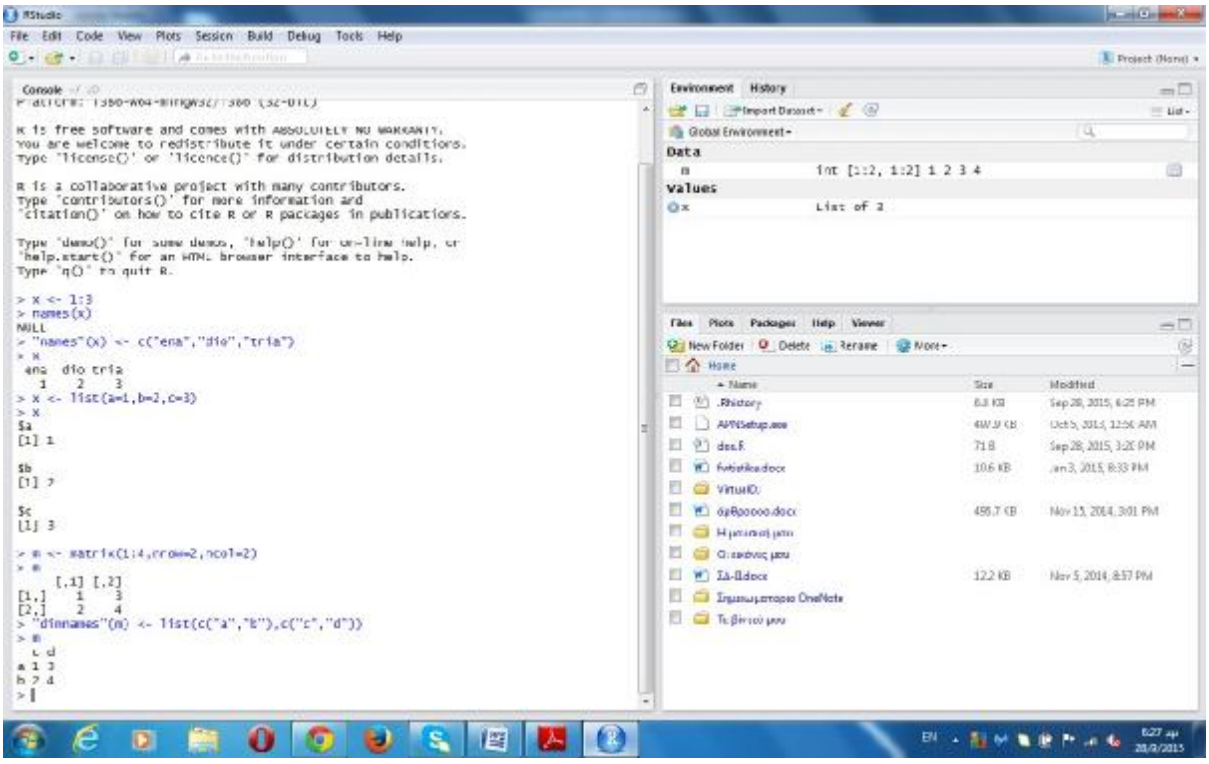
Τα αντικείμενα data frames είναι λίστες όπου κάθε στοιχείο της λίστας έχει το ίδιο μήκος. Είναι πολύ χρήσιμα αντικείμενα εφόσον τα περισσότερα δεδομένα που επεξεργαζόμαστε ανήκουν σε αυτήν την κατηγορία. Δημιουργούνται είτε απευθείας με τη συνάρτηση data.frame() είτε διαβάζοντας δεδομένα από εξωτερικά αρχεία με τις read.table() ή read.csv().



Μπορεί να χρειαστούμε να μετατρέψουμε ένα data frame σε vector (ομοειδή αντικείμενα). Γίνεται με την stack() και επαναφορά σε data frame με την unstack(). Μπορούμε να επιθέσουμε περισσότερα data frames κατά στήλη και γραμμή και μπορούμε επίσης να ανοίξουμε έναν data viewer με την fix(). Αυτό μας επιτρέπει να έχουμε μια γρήγορη εικόνα ενός μεγάλου data frame και να κάνουμε ίσως κάποια μικρή διόρθωση σε στοιχεία του.



2.1.10 Names Τα στοιχεία ενός R αντικειμένου μπορούν να έχουν και δικά τους ονόματα κάτι χρήσιμο για επεξεργασία και αυτοπροσδιορισμό.

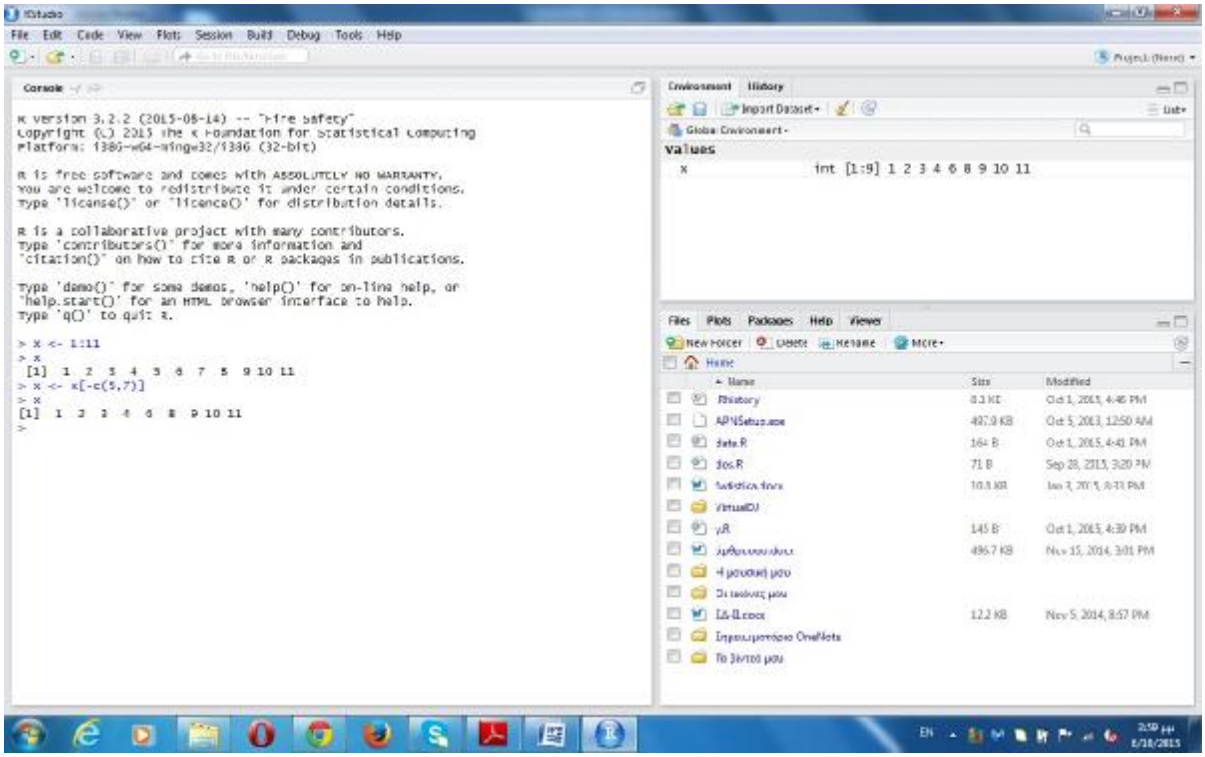


2.1.11 Διάβασμα/γράψιμο δεδομένων

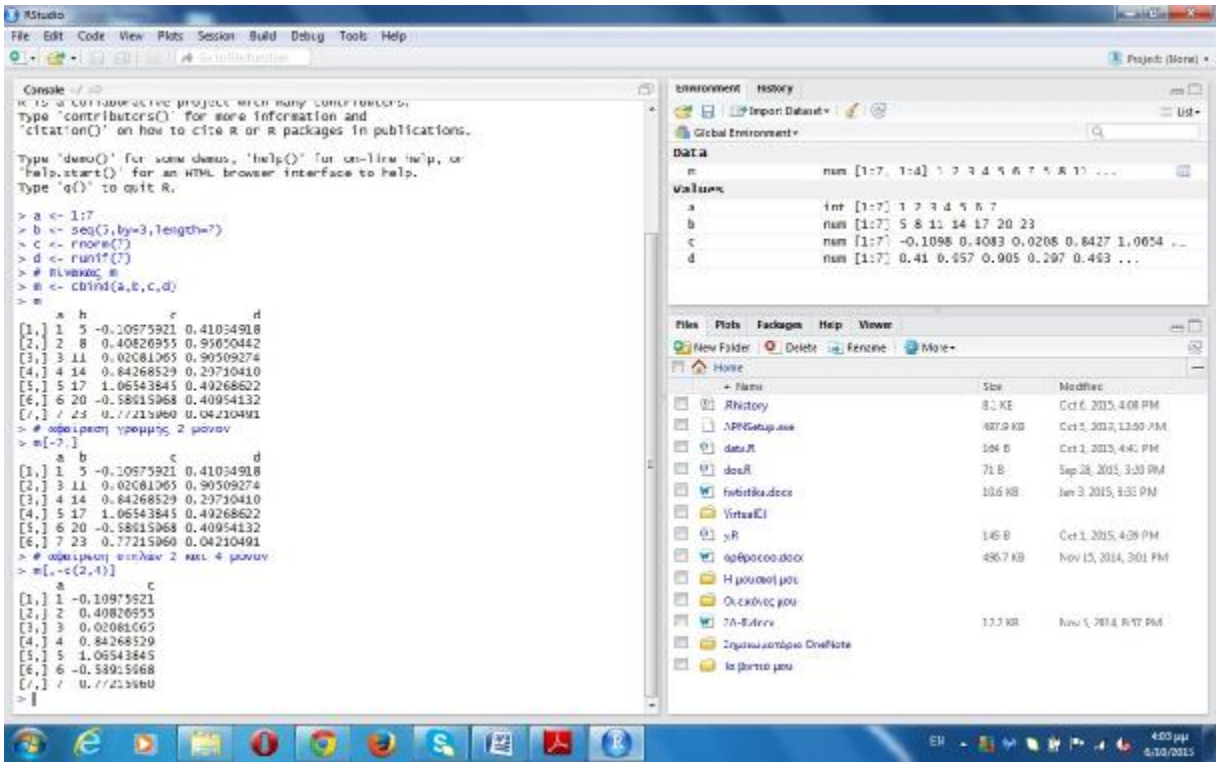
Διάβασμα δεδομένων με τις συναρτήσεις:

- read.table, read.csv - διαβάζουν δεδομένα από εξωτερικά αρχεία σε data frames
- readLines - διαβάζει γραμμές από αρχείο κειμένου text σε διάνυσμα character
- source - διαβάζει R κώδικα (αντίθετη της dump)
- dget - διαβάζει R κώδικα (αντίθετη της dput)
- load - διαβάζει workspaces που έχουν αποθηκευτεί
- unserialize - διαβάζει αντικείμενα από αρχεία binary Αντίστοιχες συναρτήσεις για γράψιμο:
- write.table, write.csv
- writeLines
- dump
- dput
- save
- serialize

2.1.12 Αφαίρεση στοιχείων Μπορούμε να αφαιρέσουμε συγκεκριμένα στοιχεία από διανύσματα, π.χ. το 5ο και το
7ο από το x με τη χρήση του -c()

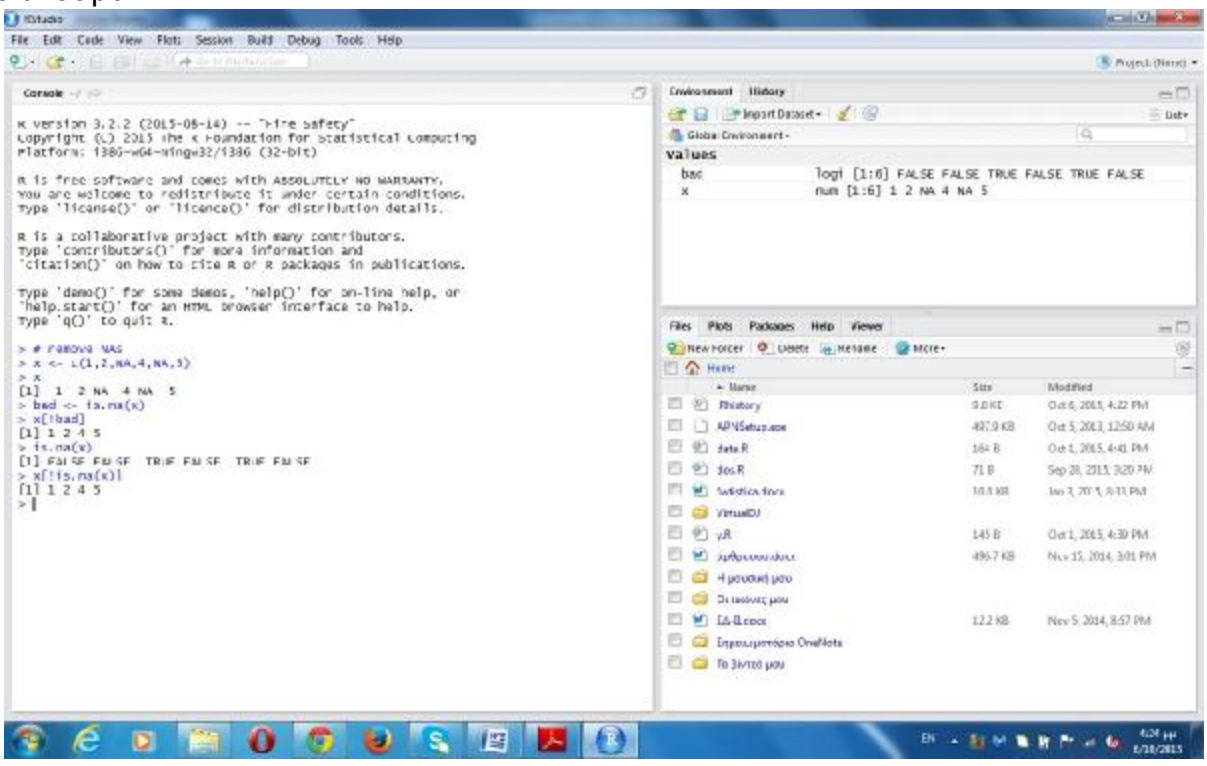


Για πίνακες η αφαίρεση συγκεκριμένων γραμμών ή στηλών γίνεται ως εξής:



2.1.13 Αφαίρεση NA

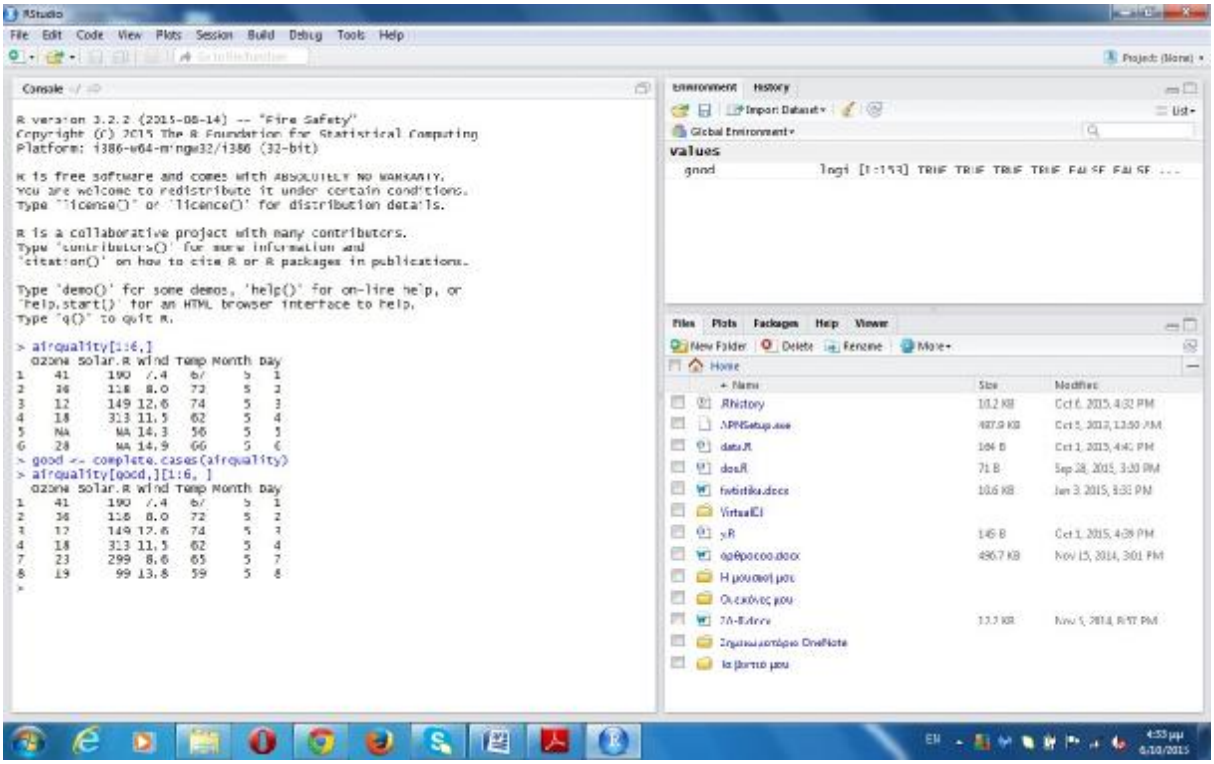
Με την εκλεξιμότητα μπορούμε να αφαιρούμε NA τιμές, π.χ. για κάποιο διάνυσμα:



Στην R υπάρχουν ενσωματωμένα κάποια αρχεία δεδομένων.

> datasets

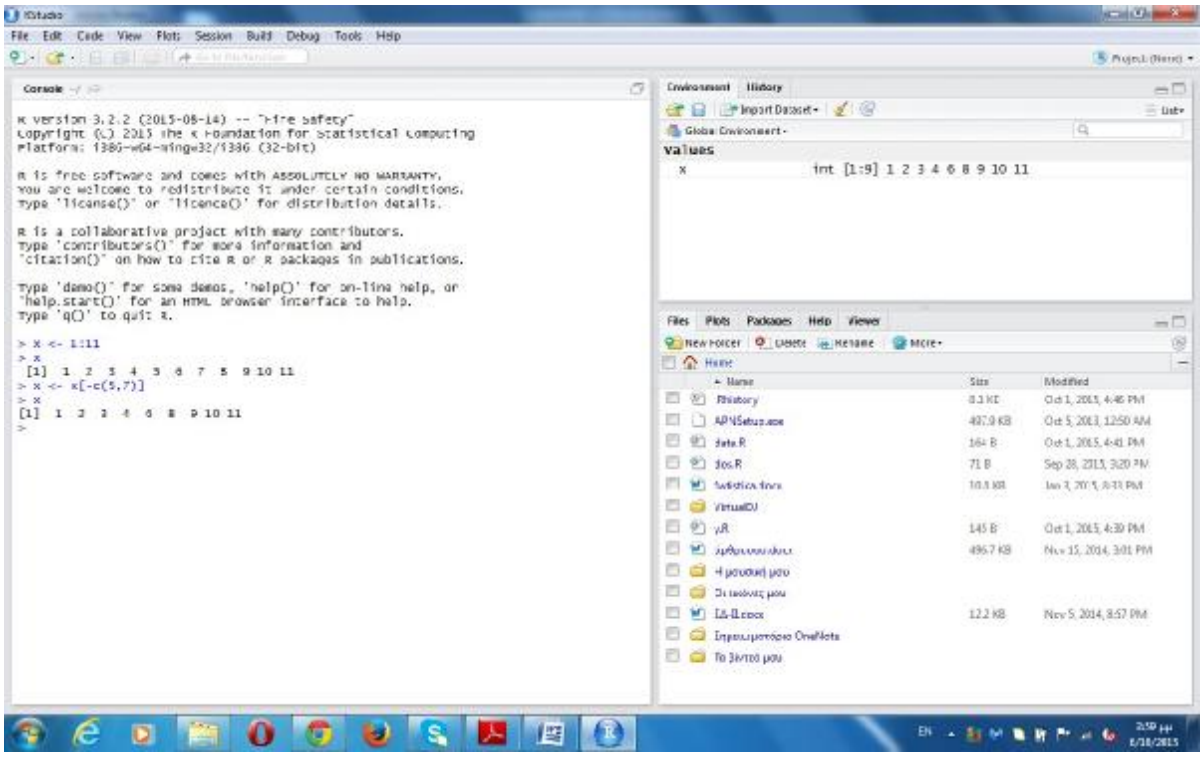
> library(help = ”datasets”) παρουσιάζει λίστα με το τι υπάρχει. Ένα από τα αρχεία δεδομένων είναι το airquality, ένα data frame που περιέχει μετρήσεις ποιότητας ατμοσφαιρικού αέρα στη Νέα Υόρκη (πληκτρολογείστε airquality για λεπτομέρειες). Επιλέγουμε τις πρώτες 6 γραμμές και βλέπουμε κάποια NA.



Με την complete.cases αφαιρούμε τις γραμμές με NA καθαρίζοντας έτσι τα δεδομένα για περαιτέρω επεξεργασία. Προσέξτε τον τρόπο με τον οποίο επιλέξαμε τις καθαρές γραμμές αφαιρώντας τις 5 και 6 που είχαν NA.

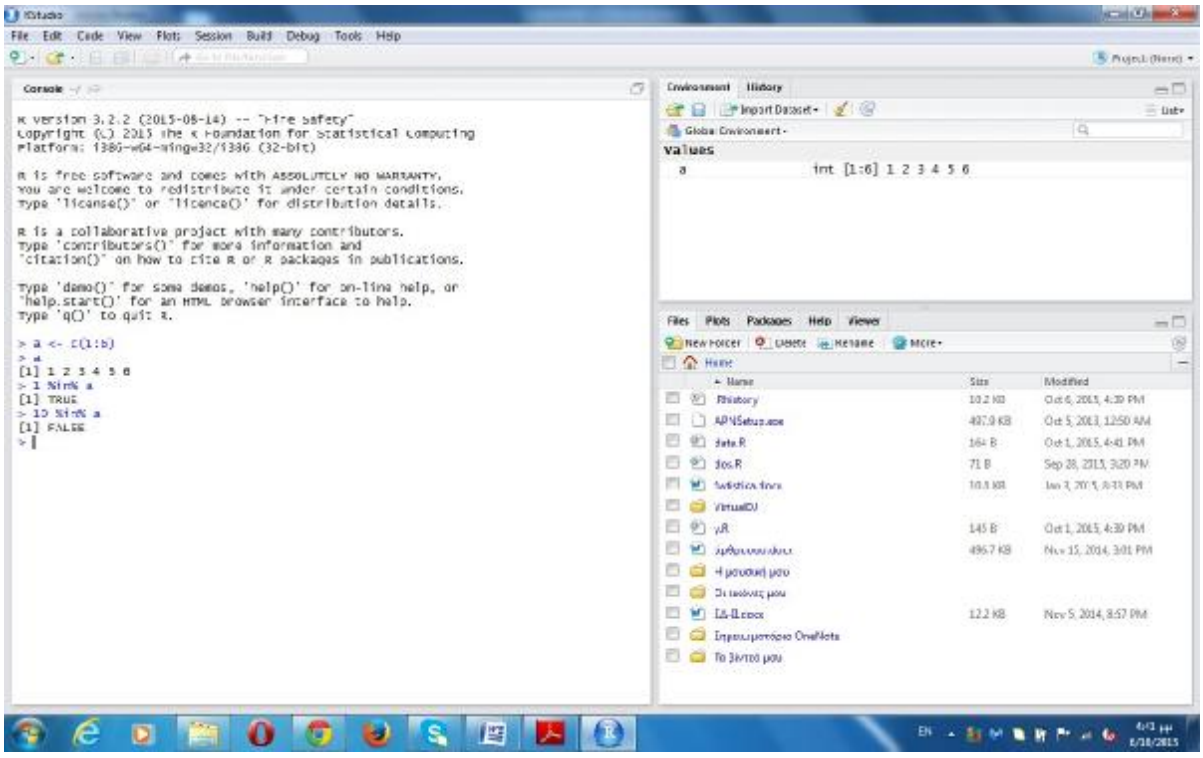
2.1.14 Διανυσματοποίηση πράξεων

Πολλές από τις πράξεις στην R είναι διανυσματοποιημένες με αποτέλεσμα πιο απλό κώδικα.

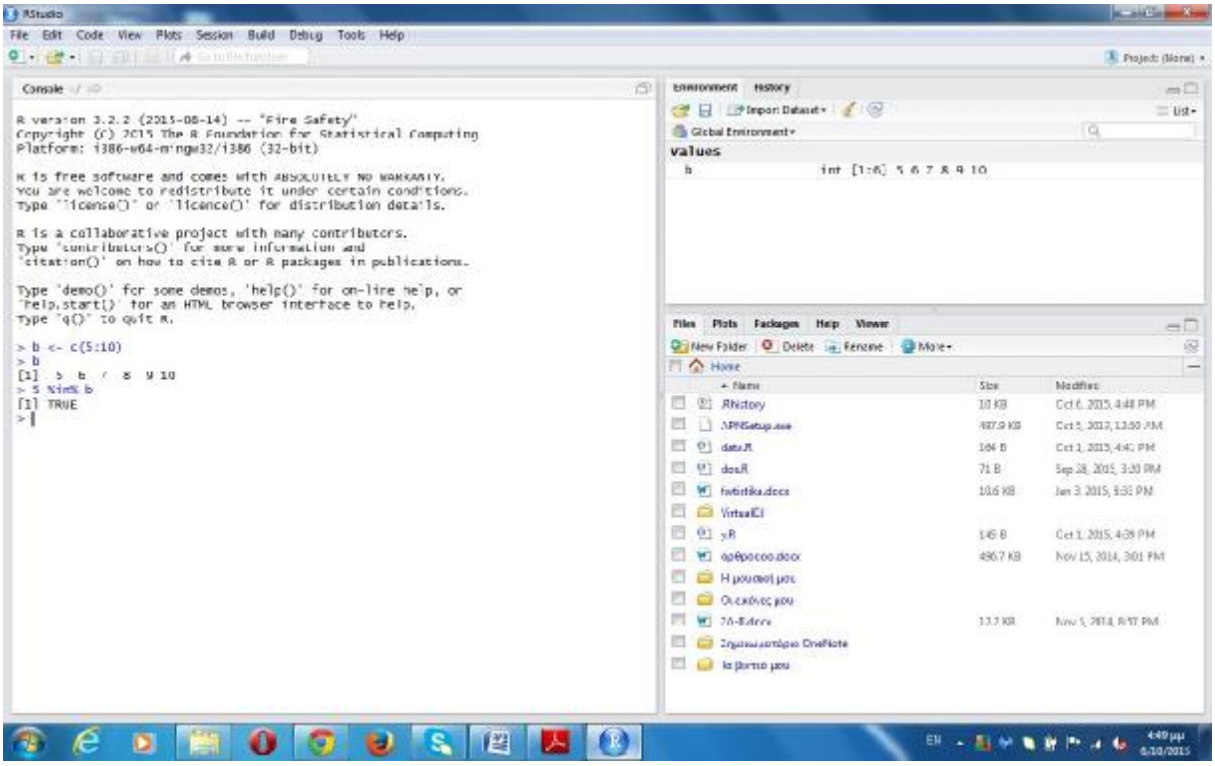


2.1.15 Επιπλέον πράξεις

Μπορούμε να ελέγξουμε αν κάποια τιμή βρίσκεται σε κάποιο αντικείμενο

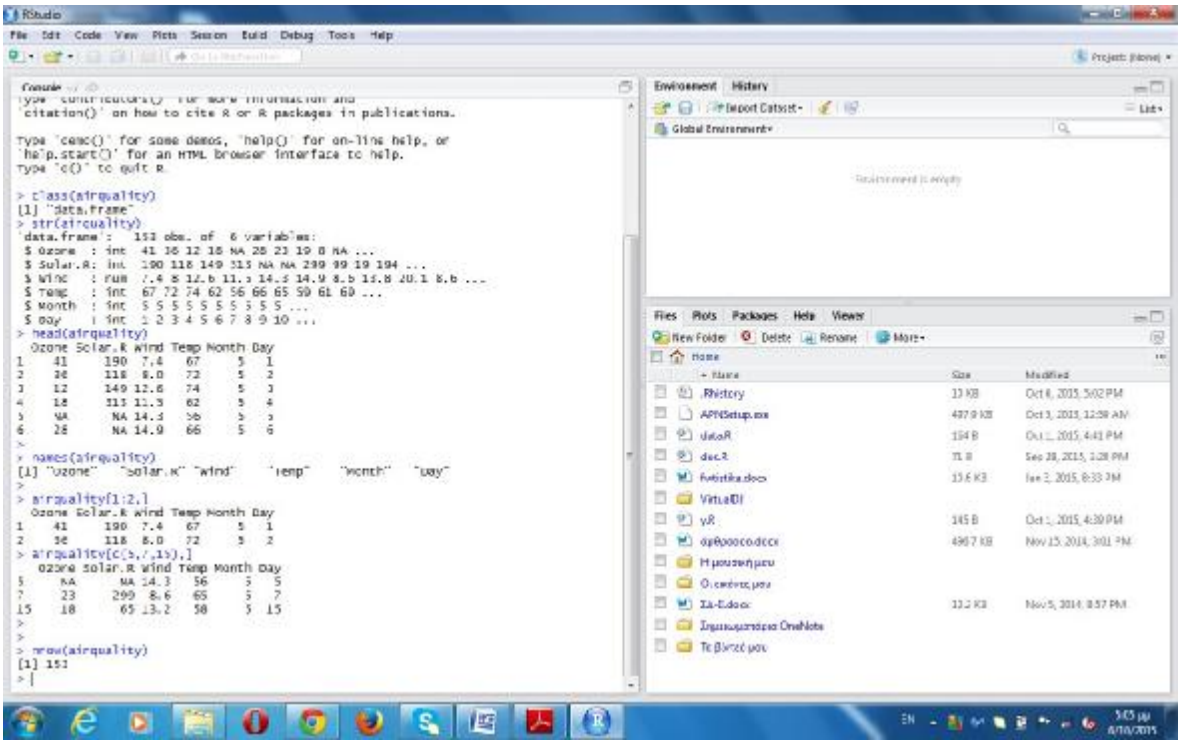


Ή εάν κάθε στοιχείο του a βρίσκεται στο b με επιστροφή ένα λογικό διάνυσμα μήκους length(a) με TRUE εάν το αντίστοιχο στοιχείο βρίσκεται στο b και FALSE εάν όχι.

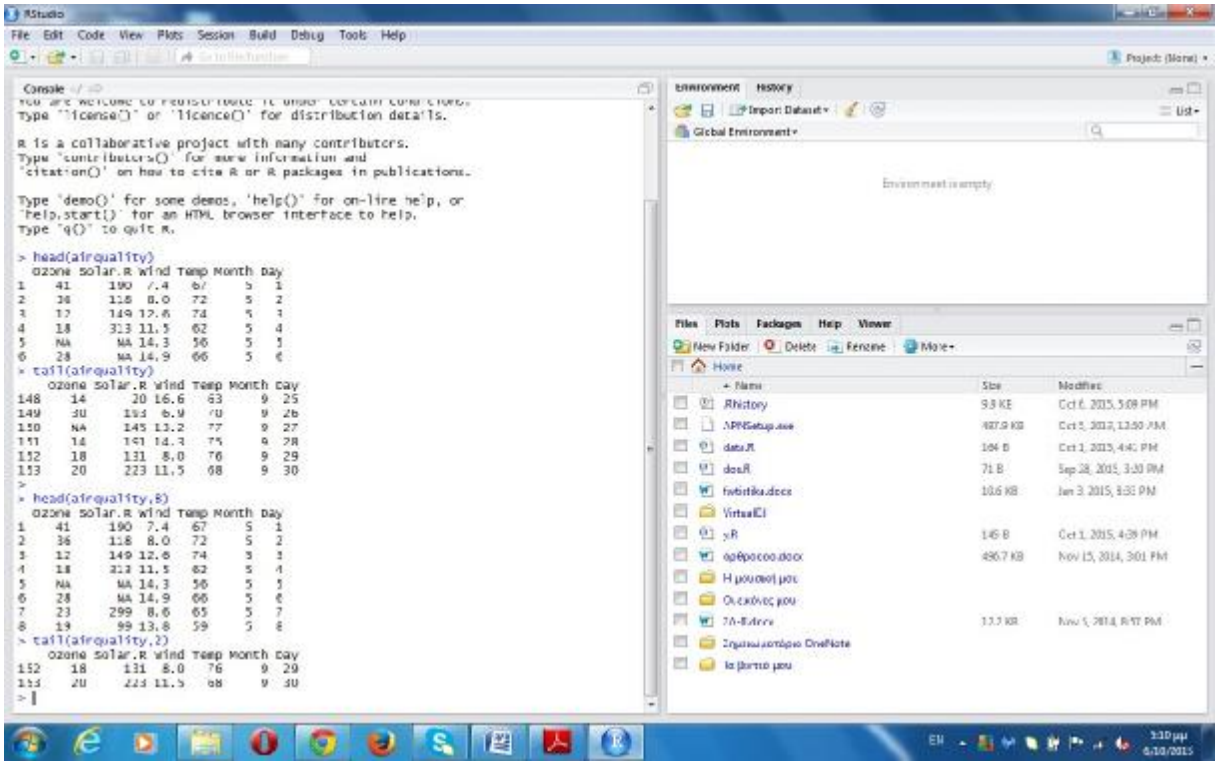


2.1.16 Η βάση δεδομένων airquality

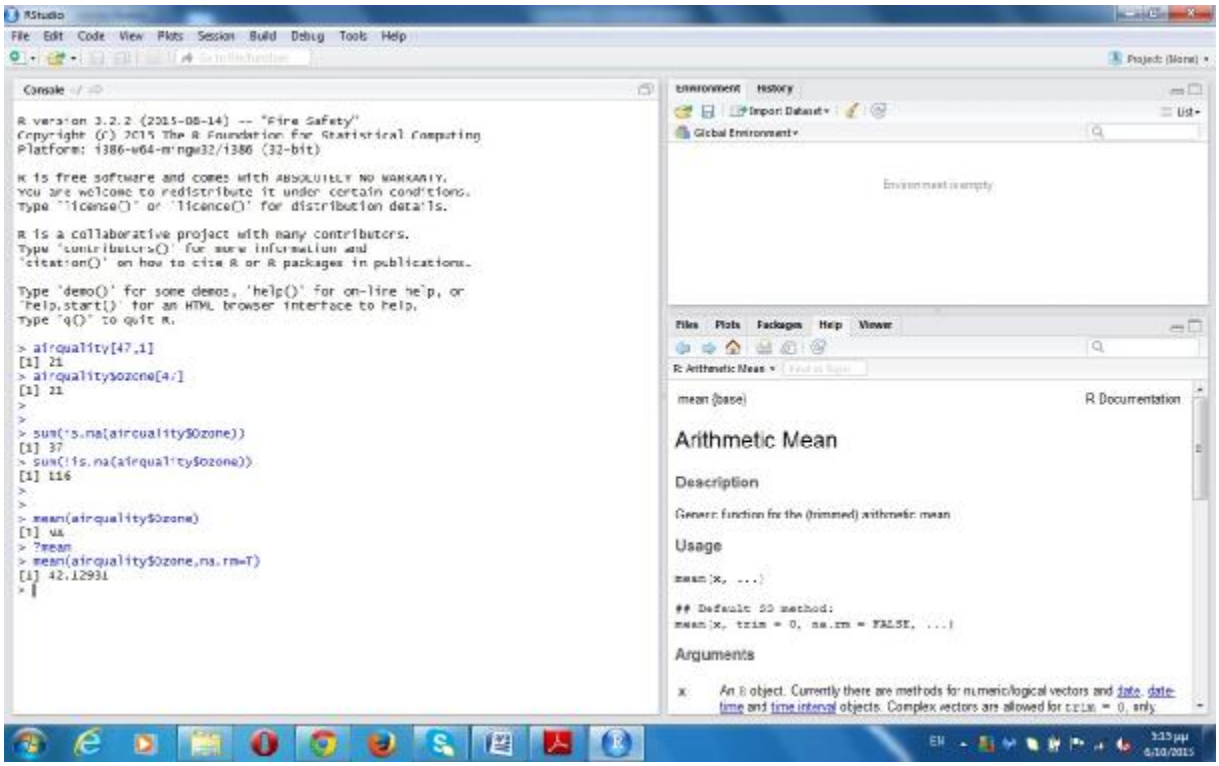
Θα δουλέψουμε λίγο με τη βάση airquality. Βλέπουμε ότι πρόκειται για data frame με 153 μετρήσεις 6 μεταβλητών. Τα ονόματα των μεταβλητών μπορούμε να τα επιλέξουμε όπως φαίνεται στην αρχή. Επίσης αν θέλουμε τις δυο πρώτες γραμμές. Ακόμα και αν θέλουμε τις γραμμές 5, 7, 15: Τέλος, πόσες μετρήσεις (γραμμές) υπάρχουν στη βάση;



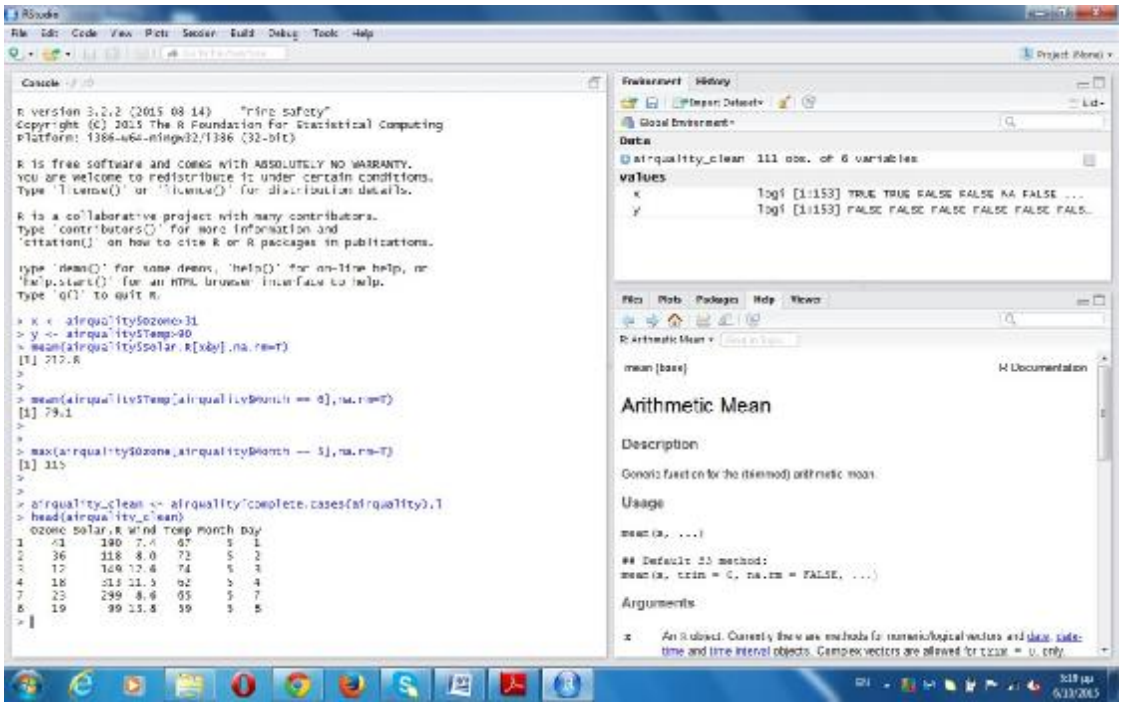
Οι συναρτήσεις head και tail μας δείχνουν τις πρώτες και τελευταίες τιμές της βάσης. Μπορούμε να δούμε και παραπάνω ή παρακάτω γραμμές:



Μπορούμε να δούμε ποια είναι η τιμή του όζοντος στη γραμμή 47 (δυο τρόποι); Πόσα NA έχει η στήλη του όζοντος και πόσες καθαρές τιμές; Και ποια είναι η μέση τιμή του όζοντος;



Εδώ χρησιμοποιήσαμε την συνάρτηση mean. Απλή χρήση μας δίνει NA γιατί η στήλη περιέχει τιμές NA. Κοιτάζοντας την σελίδα mean βλέπουμε ότι πρέπει να ορίσουμε την μεταβλητή na.rm σαν αληθή έτσι ώστε να αφαιρεθούν οι NA τιμές και να υπολογιστεί η σωστή μέση τιμή. Εξάγετε τις μετρήσεις όπου οι τιμές όζοντος είναι πάνω από 31 και θερμοκρασίας πάνω από 90. Ποια είναι η μέση ηλιακή ακτινοβολία σε αυτό το υποσύνολο; Ποια είναι η μέση θερμοκρασία για τον μήνα 6; Ποια είναι η μέγιστη τιμή όζοντος για τον μήνα 5; Καθαρίστε τα δεδομένα airquality από όλα τα NA. Πόσες είναι οι καθαρές μετρήσεις;



Καθαρίζουμε με την συνάρτηση complete.cases. Προσοχή γιατί το αποτέλεσμα complete.cases(airquality) θα δώσει απλώς μια στήλη με TRUE, FALSE όπου το TRUE αντιστοιχεί σε καθαρή μέτρηση και το FALSE σε μη. Τα στοιχεία της στήλης αντιστοιχούν στις γραμμές του data frame. Η airquality [complete.cases(airquality),] σημαίνει ότι θα εμφανίσει όλες τις καθαρές γραμμές και με το , θα δώσει επίσης όλες τις στήλες.

2.2 Προγραμματισμός με R

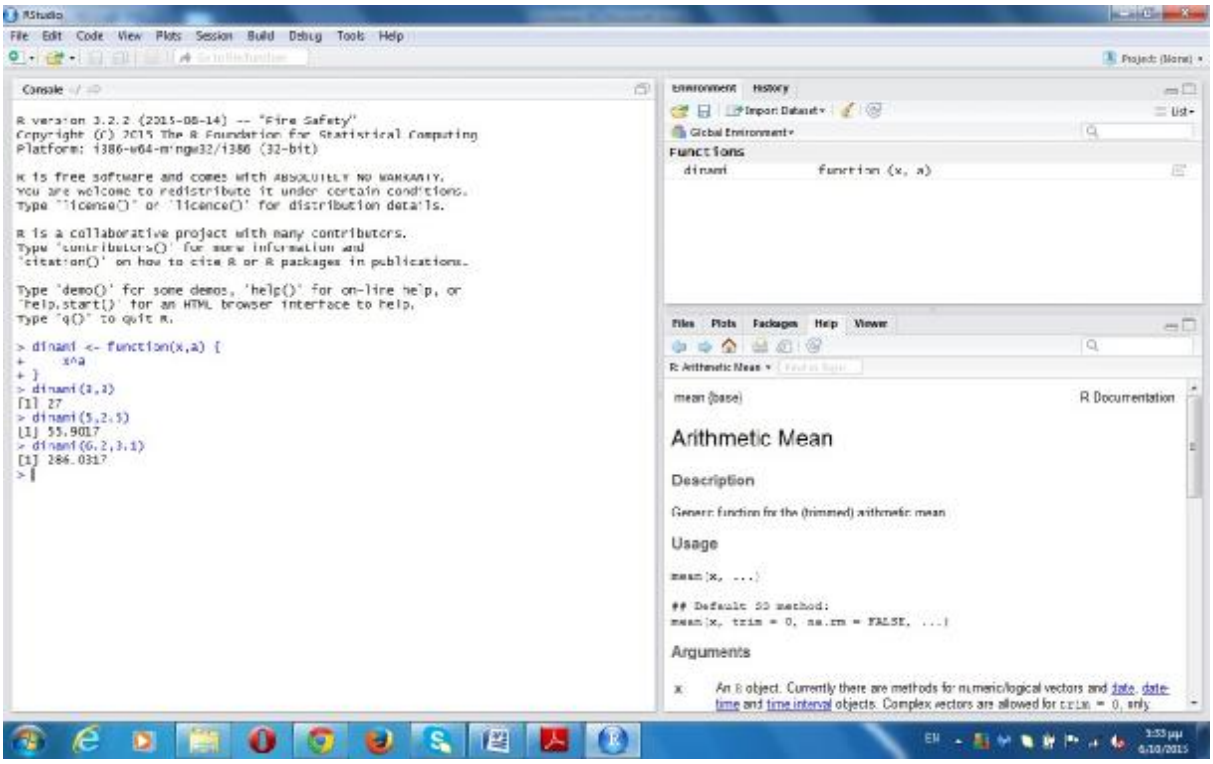
2.2.1 Δομές ελέγχου

Οι δομές ελέγχου επιτρέπουν τον έλεγχο της ροής ενός προγράμματος ανάλογα με διάφορες συνθήκες ροής του προγράμματος.

- if,else: ελέγχει μια κατάσταση αν είναι αληθής ή όχι και πράττει ανάλογα.
- for: εκτελεί βρόγχο επανάληψης συγκεκριμένες φορές
- while: εκτελεί βρόγχο επανάληψης εάν αληθεύει κάποια συνθήκη
- repeat: εκτελεί βρόγχο επανάληψης άπειρες φορές - χρειάζεται break για να σταματήσει
- break: σταματά έναν βρόγχο επανάληψης και το πρόγραμμα πάει στην αμέσως επόμενη εκτελέσιμη εντολή
- next: πηδά και δεν εκτελεί μια επανάληψη - συνεχίζει όμως το βρόγχο
- return: σημείο εξόδου συνάρτησης με κάποια τιμή

2.2.2 Συναρτήσεις – Functions Όποτε έχουμε ομάδα εντολών που εκτελούν συγκεκριμένη λειτουργικότητα μπορούμε να τις βάλουμε σε μια συνάρτηση.
f <- function(arguments) {statements}
Η συνάρτηση είναι και αυτή ένα αντικείμενο στην R και μπορεί να χρησιμοποιηθεί όπως και όλα τα άλλα αντικείμενα.

Παράδειγμα: *Βρείτε συνάρτηση που επιστρέφει τη δύναμη ενός αριθμού. Παράμετροι εισόδου ο αριθμός και ο εκθέτης. Έξοδος συνάρτησης, η δύναμη.*

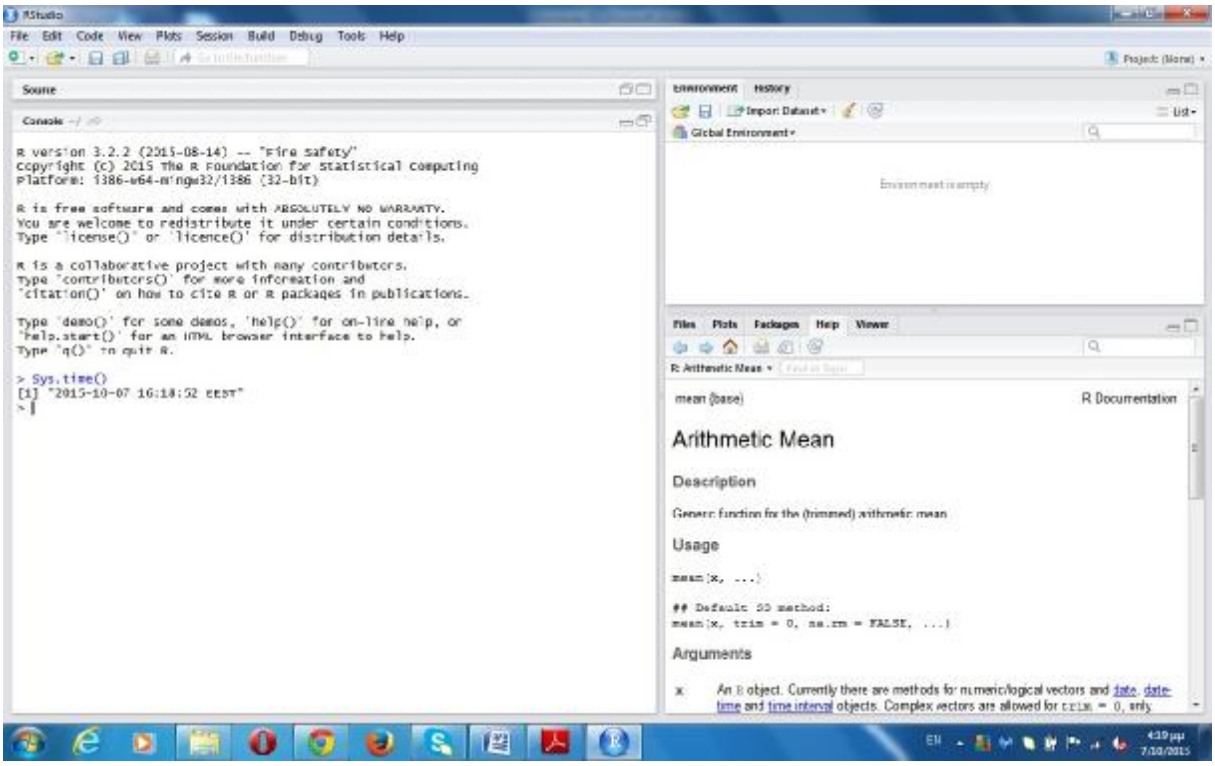


2.2.3 Dates Η R διαθέτει αντικείμενα τύπου (class) Date για την καλύτερη διαχείριση δεδομένων που εξαρτώνται από τον χρόνο.

Πίνακας: Μερικά σύμβολα για format string αντικείμενα Date

Σύμβολο	**Ερμηνεία**	**Παράδειγμα**
%a	συντόμευση ημέρας	Wed
%A	πλήρης ημέρα	Wednesday
%d	ημέρα του μήνα αριθμητική	31
%b	συντόμευση μηνός	Oct
%B	πλήρης μήνας	October
%m	μήνας αριθμητικός	03
%Y	χρόνος αριθμητικός 4ψήφιος	1932
%y	χρόνος αριθμητικός 2ψήφιος	95
%H	ώρα (00-24)	16
%I	ώρα (00-12)	09
%M	λεπτά	35
%S	δευτερόλεπτα	52

Εάν έχει κανείς Date αντικείμενα μπορεί να βρίσκει διαφορές και να κάνει πράξεις για καλύτερη διαχείριση αποτελεσμάτων. Υπάρχουν και άλλες συναρτήσεις. Μια χρήσιμη για σκοπούς τεκμηρίωσης είναι:



Όπου μας επιτρέπει να βάζουμε timestamp π.χ. για το πότε είχαμε πρόσβαση σε κάποια αρχεία από το internet.

2.3 Συναρτήσεις επανάληψης

Οι συναρτήσεις επανάληψης έχουν την ίδια λειτουργικότητα με τις δομές επανάληψης που είδαμε προηγουμένως, απλώς είναι πιο συνοπτικές και εύχρηστες για διαδραστική χρήση. Οι συναρτήσεις αυτές έχουν τη ρίζα apply στο όνομά τους και είναι:

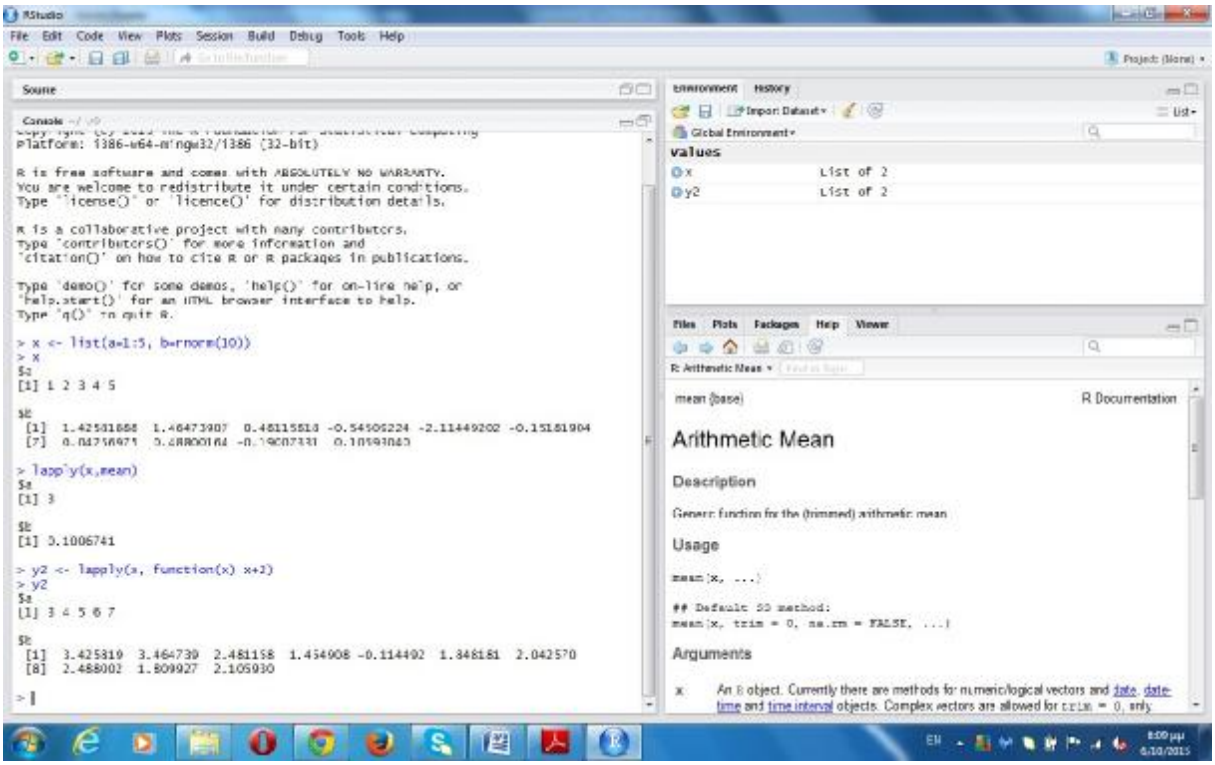
- lapply: Βρόγχος επανάληψης σε λίστα εφαρμόζοντας μια συνάρτηση σε κάθε στοιχείο της λίστας
- sapply: Ίδια λειτουργικότητα με την lapply απλώς με απλοποίηση του αποτελέσματος
- apply: Εφαρμόζει μια συνάρτηση σε γραμμές ή στήλες ενός πίνακα
- tapply: Εφαρμόζει μια συνάρτηση σε υποσύνολα διανύσματος
- mapply: Ίδια λειτουργικότητα με την lapply γενικευμένη σε πολυδιάστατα αντικείμενα

Η βοηθητική συνάρτηση split είναι χρήσιμη στις εφαρμογές των παραπάνω συναρτήσεων.

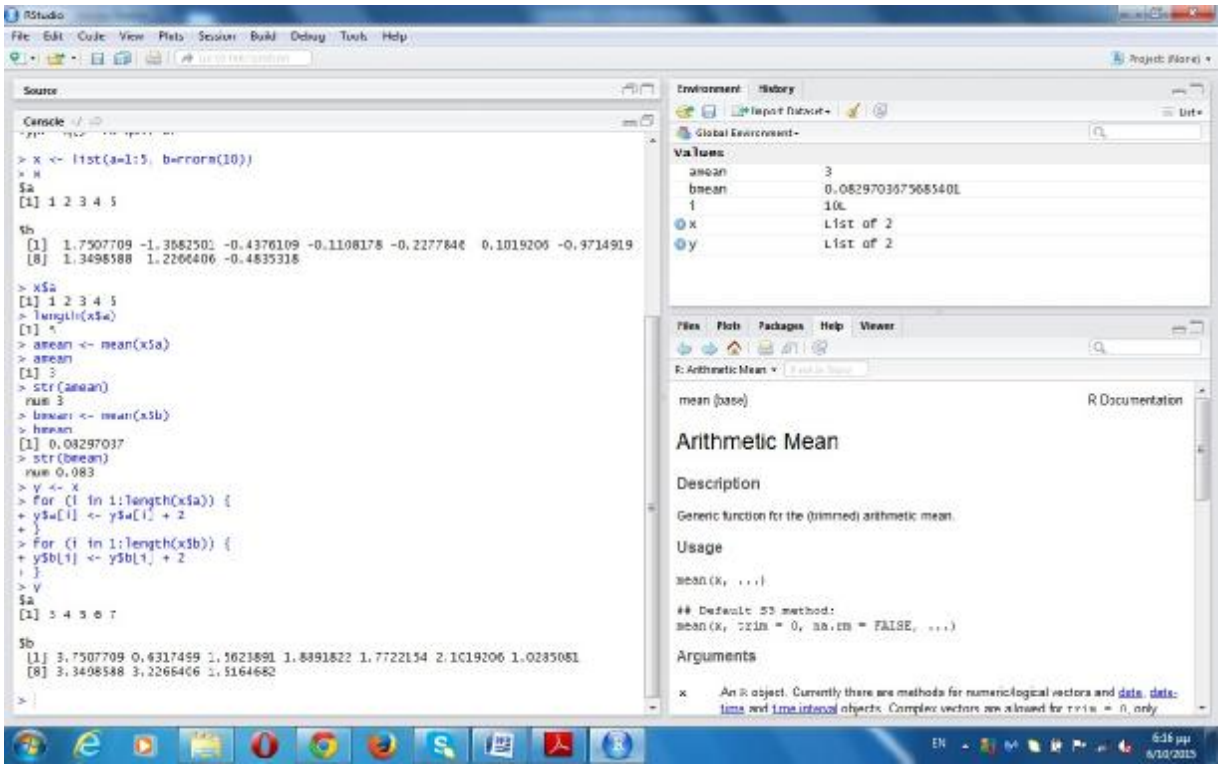
2.3.1 lapply

Η lapply έχει τρεις μεταβλητές εισόδου. Μια λίστα X, την συνάρτηση εφαρμογής FUN (ή το όνομά της) και τα ορίσματα της συνάρτησης. Εάν η X δεν είναι λίστα μετατρέπεται σε μια με την εσωτερική εφαρμογή της as.list(). Η lapply επιστρέφει σαν αποτέλεσμα πάντα μια λίστα.

Δημιουργούμε μια λίστα x με δυο στοιχεία. Το στοιχείο a, μια ακολουθία 5 ακεραίων και το στοιχείο b, μια ακολουθία 10 τυχαίων αριθμών από την κανονική κατανομή. Εφαρμόζουμε την συνάρτηση mean και βρίσκουμε τη μέση τιμή των δυο ακολουθιών. Επαναλαμβάνουμε με μια δική μας συνάρτηση που απλώς προσθέτει τον αριθμό 2 στα στοιχεία των a και b.

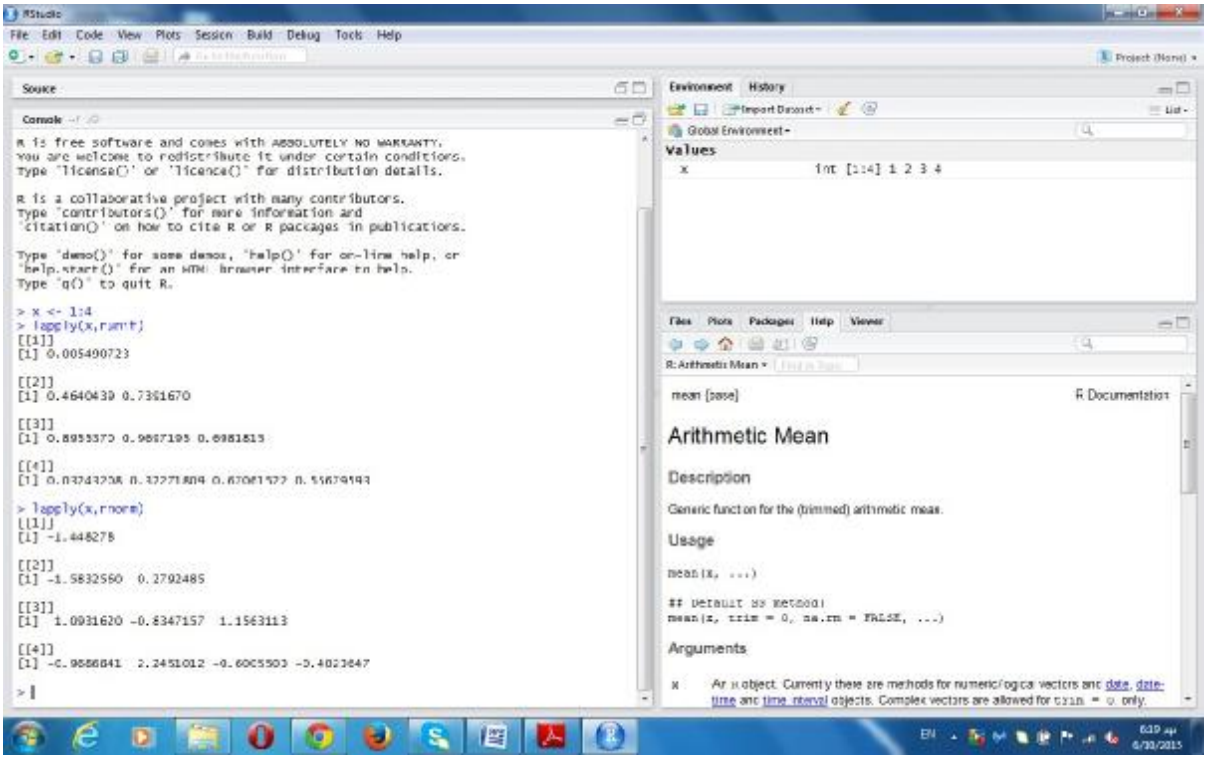


Η ίδια λειτουργικότητα με τη δομή for θα ήταν:

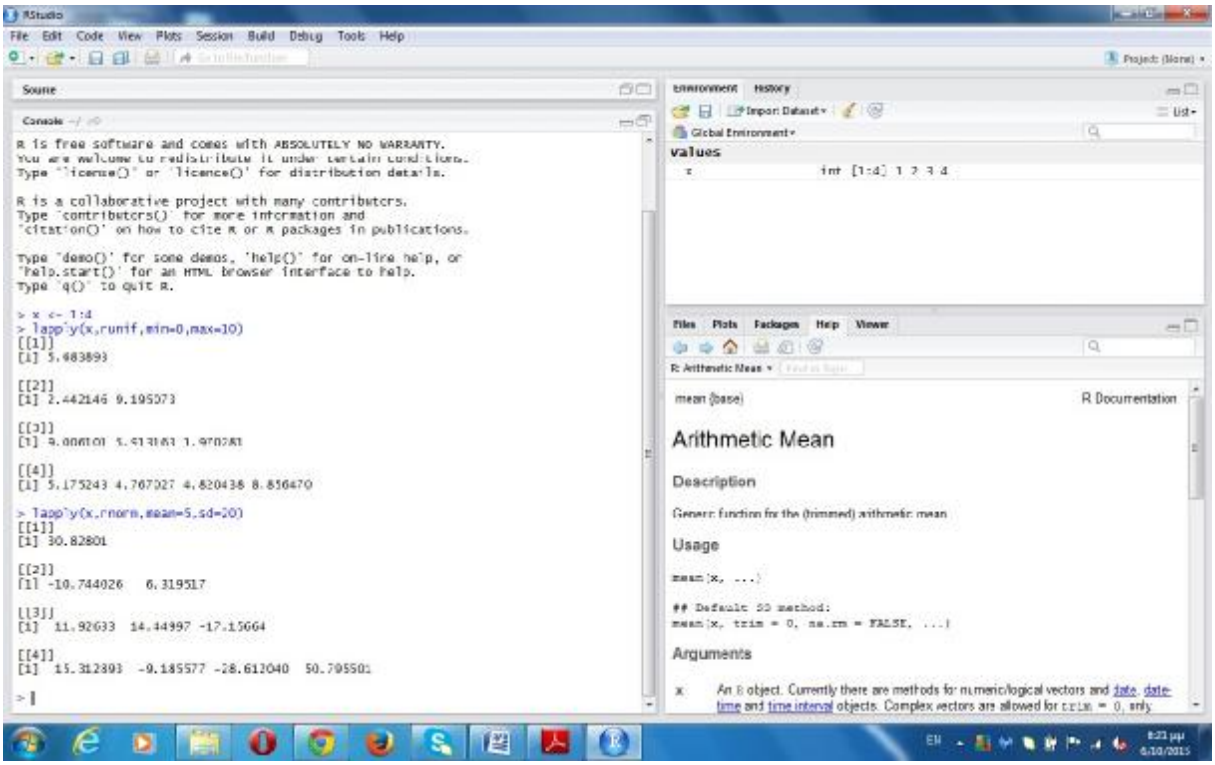


Περισσότερος κόπος και τα αποτελέσματα είναι αριθμητικά διανύσματα.

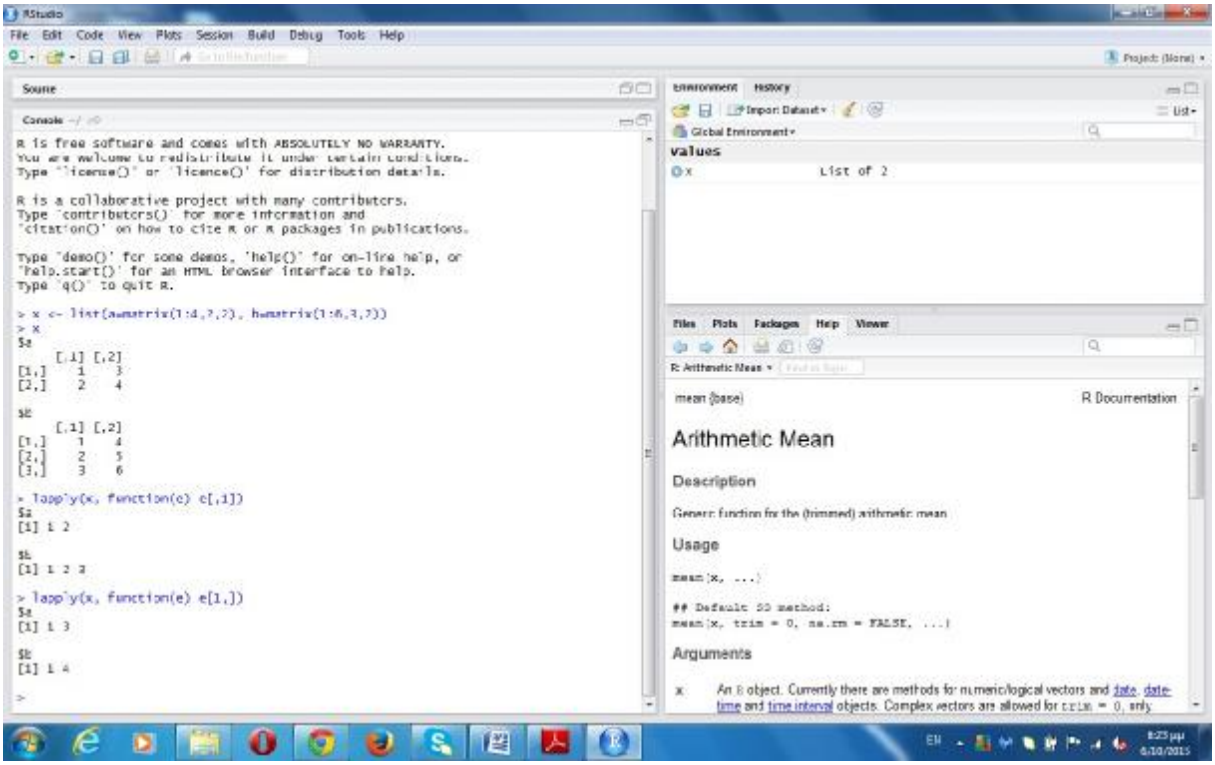
Την lapply μπορούμε να την χρησιμοποιήσουμε να φτιάξουμε π.χ. διαδοχικές ακολουθίες τυχαίων αριθμών με μεταβλητό μήκος. Δοκιμάζουμε την runif (γεννήτρια τυχαίων αριθμών ομοιόμορφης κατανομής) και την rnorm (γεννήτρια τυχαίων αριθμών κανονικής κατανομής).



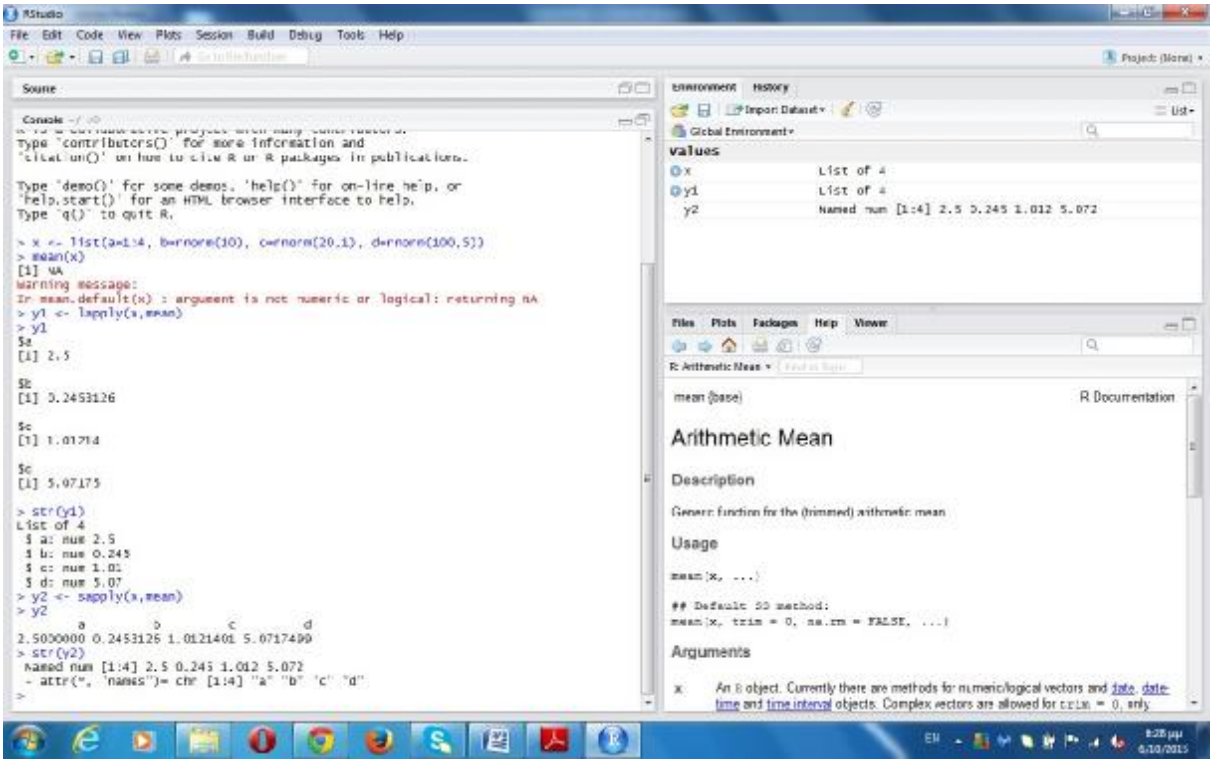
Το ίδιο παράδειγμα με διαφορετικά χαρακτηριστικά στις γεννήτριες.



Μπορούμε να χρησιμοποιήσουμε και εμείς δικές μας απλές συναρτήσεις. π.χ. να βγάλουμε την πρώτη στήλη ή πρώτη γραμμή δυο πινάκων:

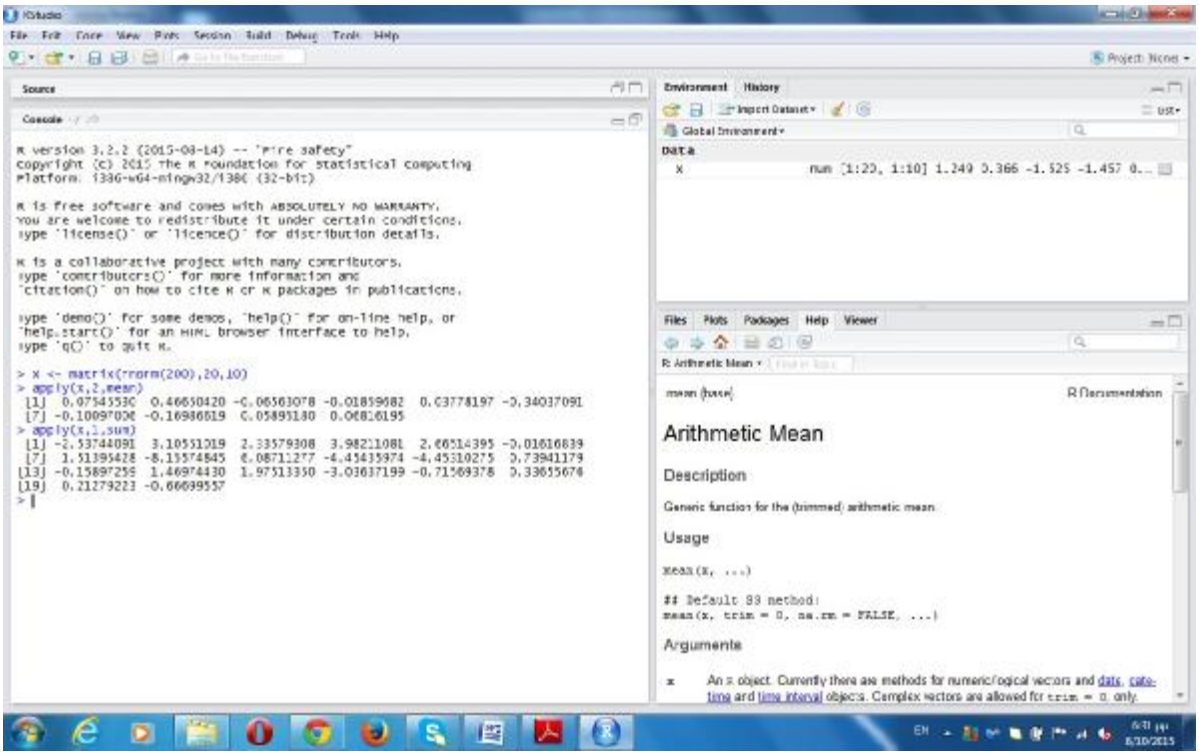


2.3.2 sapply Η sapply είναι σαν την lapply, απλώς προσπαθεί να απλοποιήσει το αποτέλεσμα



2.3.3 apply

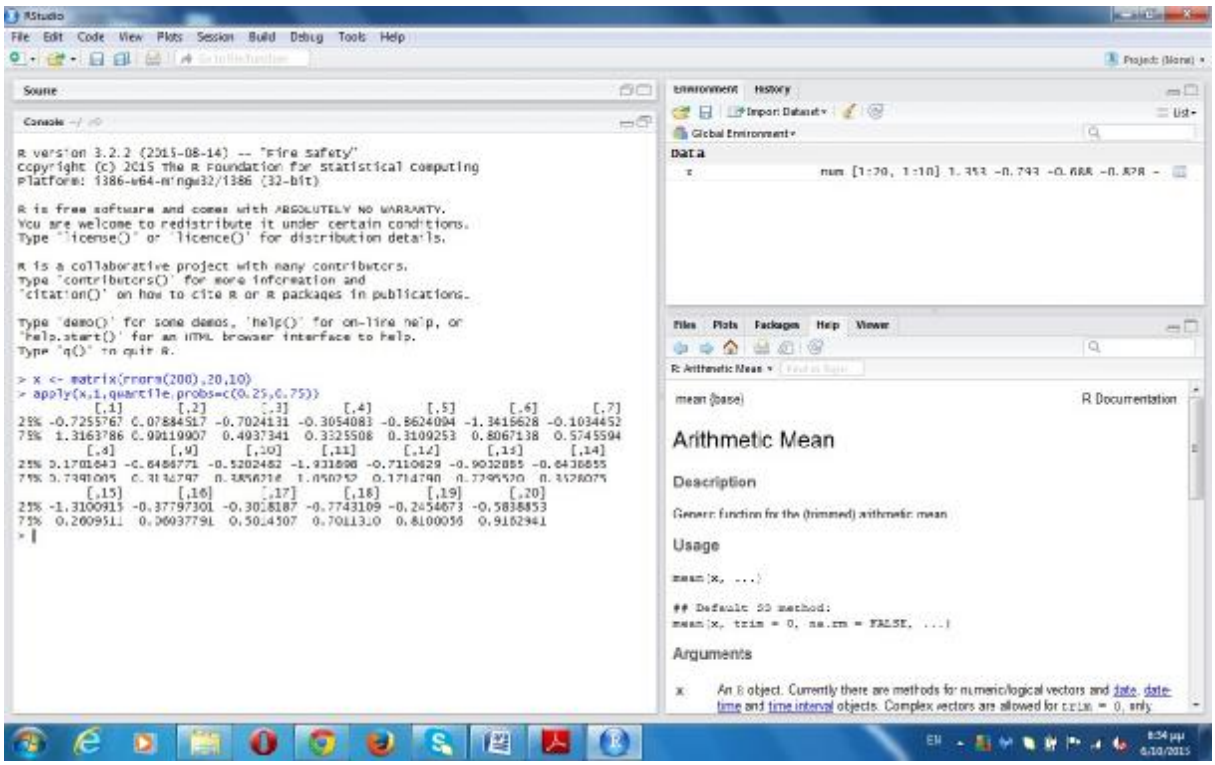
Η apply εφαρμόζει μια συνάρτηση σε γραμμές ή στήλες ενός πίνακα. Τη δοκιμάζουμε σε πίνακα 20_10 για υπολογισμό μέσης τιμής (στήλες) ή αθροίσματος (γραμμές). Το αν θα εφαρμόσουμε τη συνάρτηση στις γραμμές ή στήλες εξαρτάται από το δεύτερο όρισμα της apply. Τιμή 1 για γραμμές, τιμή 2 για στήλες.



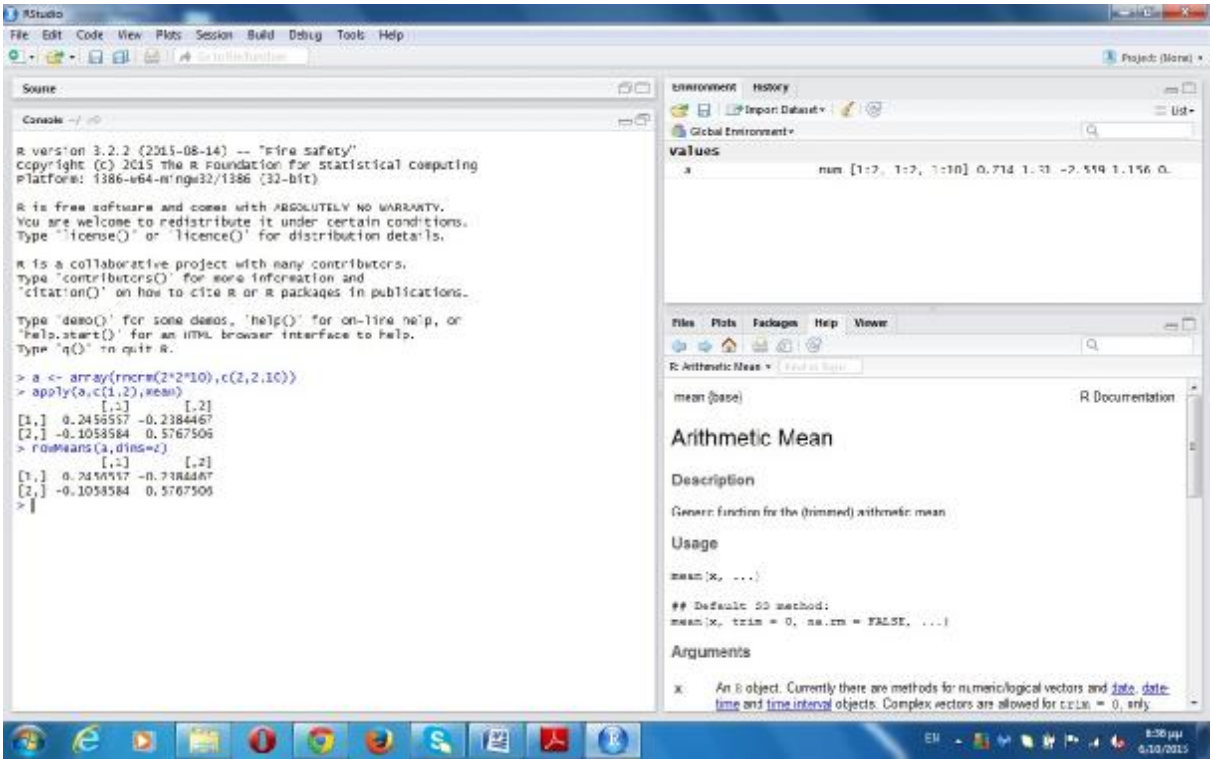
Υπάρχουν και οι συντομεύσεις:

- rowSums = apply(x,1,sum)
- rowMeans = apply(x,1,mean)
- colSums = apply(x,2, sum)
- colMeans = apply(x,2, mean)

Quantiles των γραμμών ενός πίνακα:

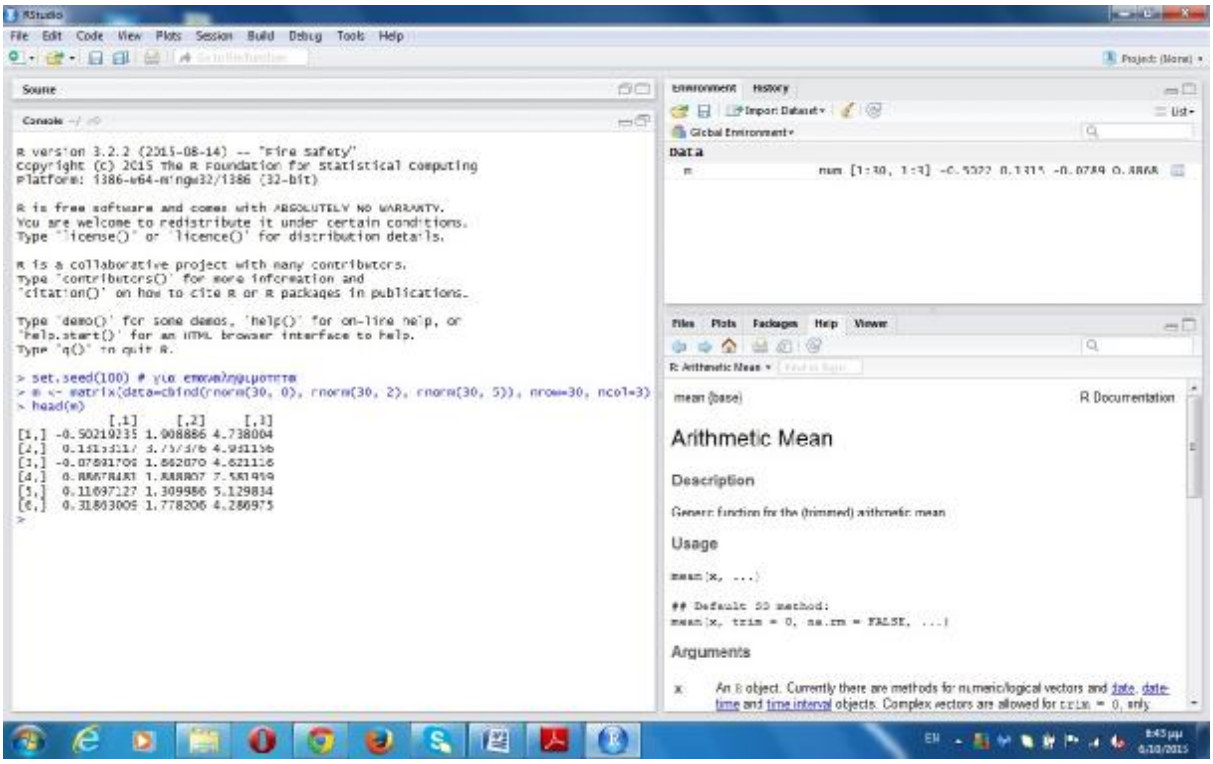


Μέση τιμή στοιχείων 3-διάστατου πίνακα κατά μήκος της τρίτης διάστασης:

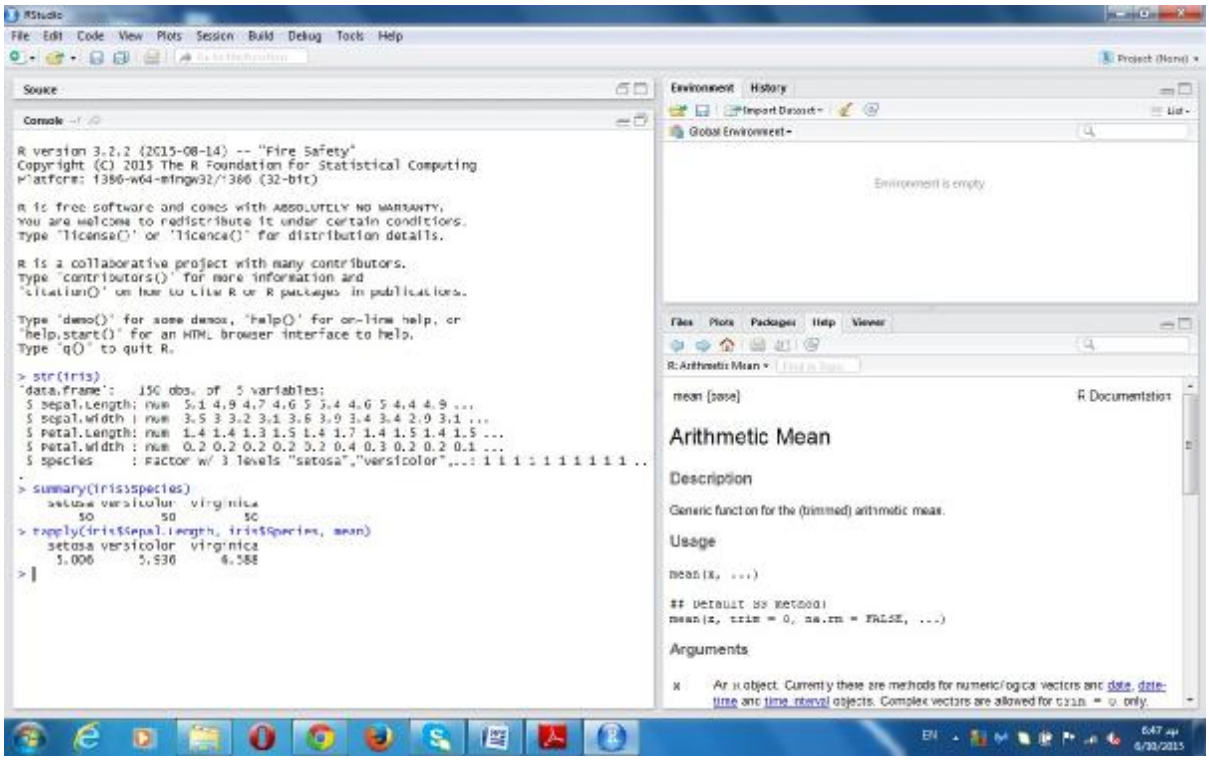


Ένα καλό παράδειγμα από την [11].

Έστω ότι έχουμε έναν πίνακα με 3 στήλες που αντιστοιχούν ίσως σε 3 μεταβλητές ή 3 διαφορετικούς τρόπους μιας μέτρησης. Έστω επίσης ότι έχουμε 30 γραμμές που αντιστοιχούν σε 30 μετρήσεις. Μπορούμε να εξομοιώσουμε έναν τέτοιο πίνακα π.χ. ως εξής:

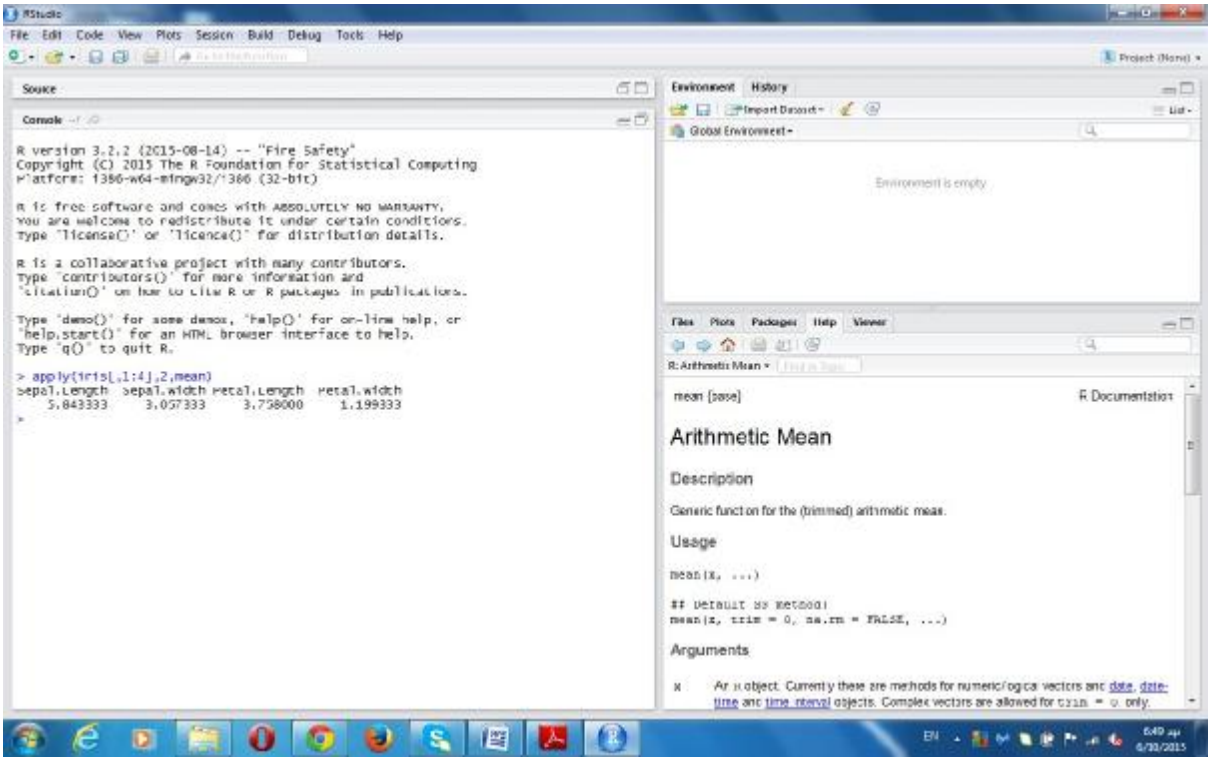


2.3.4 Παραδείγματα Παράδειγμα: Η ενσωματωμένη βάση iris μπορεί να φορτωθεί με τις εντολές:
library(datasets) data(iris)
Ποια είναι η μέση τιμή της μεταβλητής Sepal.Length για το Species virginica; Φορτώνουμε τη βάση και με τις str() και summary() βλέπουμε τη δομή της βάσης και ότι το 1/3 των δεδομένων ανήκουν στο Species virginica. Με την tapply βρίσκουμε τη μέση τιμή της μεταβλητής Sepal.Length και για τα 3 είδη Species.

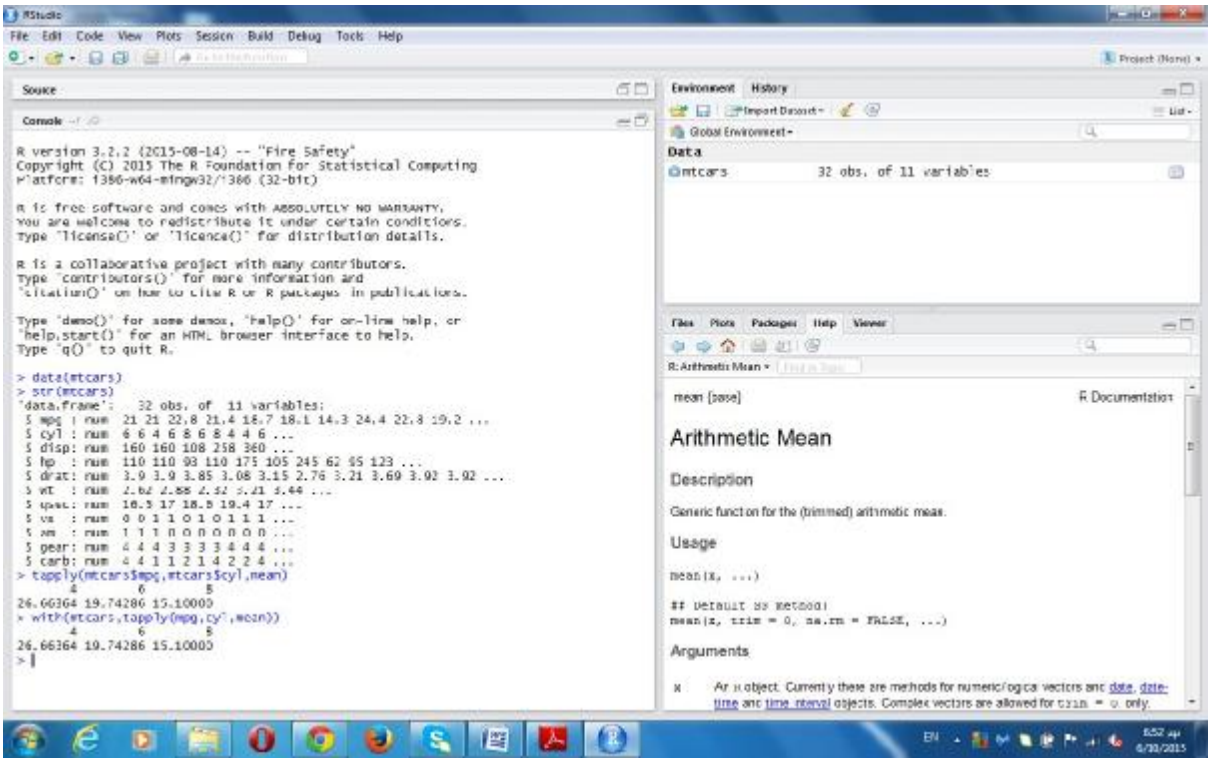


Το ζητούμενο είναι η τελευταία τιμή 6.588.

Παράδειγμα: Για την ίδια βάση iris ποια είναι η μέση τιμή των μεταβλητών Sepal.Length, Sepal.Width, Petal.Length, Petal.Width*;* Θέλουμε τη μέση τιμή για κάθε μια από τις 4 πρώτες στήλες. Εφαρμογή της apply για το αντικείμενο iris[,1:4] με παράμετρο 2 για στήλες και mean για τη συνάρτηση.



Παράδειγμα: *Μια άλλη βάση είναι η mtcars. Ποια είναι η μέση τιμή mpg (miles per gallon) ανάλογα με τους κυλίνδρους cyl του αμαξιού;* Πάλι κοιτάζουμε πρώτα τη δομή της βάσης με την str(). Θέλουμε τη μέση τιμή mpg ανάλογα με τον αριθμό cyl. Εφαρμογή της tapply με δυο ισοδύναμους τρόπους. Ο δεύτερος χρησιμοποιεί την συνάρτηση with() έτσι ώστε να μην χρειάζεται να επαναλαμβάνουμε το όνομα της βάσης πολλές φορές.



Φαίνεται πάντως ότι η μέση κατανάλωση καυσίμου είναι μεγαλύτερη για λιγότερους κυλίνδρους.

Κεφάλαιο 3: Εφαρμογή σε ανάλυση δεικτών νοσοκομείων

Στο assignment 3 του [2] γίνεται ανάλυση ποιότητας σε δείκτες νοσοκομείων των ΗΠΑ. Τα δεδομένα που χρησιμοποιούμε ανακτήθηκαν την 2015-08-17 21:17:44 EEST από: http://medicare.gov/download/HospitalCompare/2012/July/HOSArchive_Revised_Flatfiles_20120701.zip.

Το συμπιεσμένο αρχείο περιέχει μια βάση δεδομένων από διάφορους δείκτες ποιότητος για τα νοσοκομεία των ΗΠΑ για το 2012 από το κυβερνητικό πρόγραμμα ιατροφαρμακευτικής ασφάλισης medicare των ΗΠΑ.

Από το https://en.wikipedia.org/wiki/Medicare_(United_States):

In the United States, Medicare is a national social insurance program, administered by the U.S. federal government since 1966, currently using about 30 private insurance companies across the United States. Medicare provides health insurance for Americans aged 65 and older who have worked and paid into the system. It also provides health insurance to younger people with disabilities, end stage renal disease and amyotrophic lateral sclerosis. In 2010, Medicare provided health insurance to 48 million Americans—40 million people age 65 and older and eight million younger people with disabilities.

Η βάση αυτή αποτελείται από τα αρχεία readme.txt, Hospital_Revised_Flatfiles.pdf και 30 csv αρχεία με δεδομένα από διάφορους δείκτες νοσοκομειακής ποιότητας.

readme.txt
Hospital_Revised_Flatfiles.pdf
Agency_for_Health~h_and_Quality.csv
Agency_for_Health~ty_-_National.csv
Agency_for_Health~ality_-_State.csv
HCAHPS_Measures.csv
HCAHPS_Measures_-_National.csv
HCAHPS_Measures_-_State.csv
Healthcare_Associ~ed_Infections.csv
Healthcare_Associ~ections_State.csv
Hospital_Acquired_Condition.csv
Hospital_Acquired~on_-_National.csv
Hospital_Data.csv
Measure_Dates.csv
Medicare_Payment_~lume_Measures.csv
Medicare_Payment_~es_-_National.csv
Medicare_Payment_~sures_-_State.csv
Medicare_Spending_Per_Patient.csv
Outcome_of_Care_Measures.csv
Outcome_of_Care_M~es_-_National.csv
Outcome_of_Care_M~sures_-_State.csv
Outpatient_Imagin~ency_Measures.csv
Outpatient_Imagin~es_-_National.csv
Outpatient_Imagin~sures_-_State.csv
Process_of_Care_M~es_-_Children.csv

Process_of_Care_M~_Heart_Attack.csv
Process_of_Care_M~Heart_Failure.csv
Process_of_Care_M~es_-_National.csv
Process_of_Care_M~s_-_Pneumonia.csv
Process_of_Care_Measures_-_SCIP.csv
Process_of_Care_M~sures_-_State.csv
Structural_Measures.csv

Το αρχείο Hospital Revised Flatfiles.pdf (34 σελίδες) περιγράφει όλη τη βάση δεδομένων. Εμείς θα χρησιμοποιήσουμε μόνο δυο αρχεία:

- Outcome_of_Care_Measures.csv: Πληροφορίες για θνησιμότητα και επαναεισαγωγή μέσα σε 30 ημέρες λόγω καρδιακών προβλημάτων και πνευμονίας για πάνω από 4000 νοσοκομεία.

- Hospital_Data.csv Πληροφορίες για το κάθε νοσοκομείο.

Το άρθρο 11 του Hospital Data.csv και άρθρο 19 του Outcome of Care Measures.csv από το Hospital Revised Flatfiles.pdf περιγράφει με λεπτομέρεια το περιεχόμενο των παραπάνω αρχείων και αποτελεί ένα καλό παράδειγμα τεκμηρίωσης παρομοίων δεδομένων.

Παρόμοια δεδομένα είναι διαθέσιμα στο σύνδεσμο https://data.medicare.gov για όλο το σύστημα υγείας των ΗΠΑ.

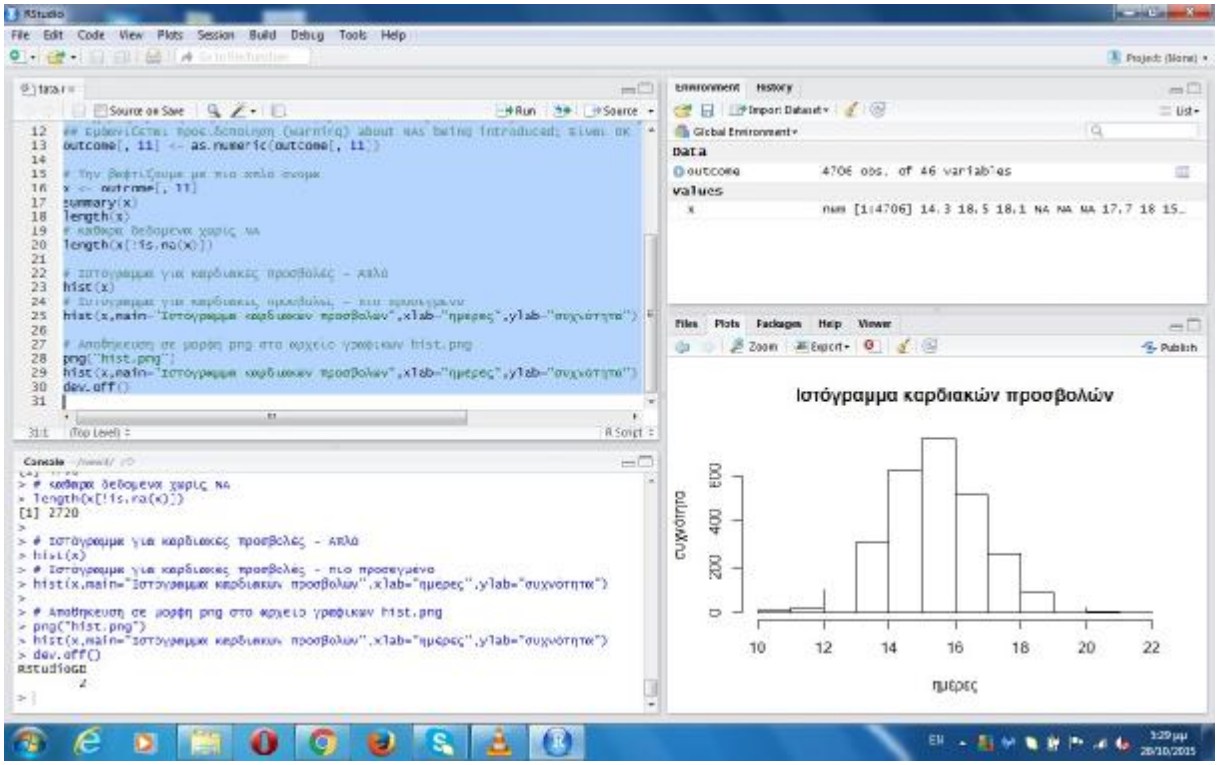
Στα παρακάτω θα χρησιμοποιήσουμε τα δεδομένα για να απαντήσουμε 4 ερωτήματα:

Ερώτημα 1

Ποια είναι η θνησιμότητα 30 ημερών για καρδιακές προσβολές;

Μας ενδιαφέρει εδώ να απαντήσουμε ποια είναι η θνησιμότητα που παρατηρείται σε ασθενείς που εισήχθησαν σε νοσοκομείο μέσα σε 30 ημέρες από την εισαγωγή τους. Προφανώς εάν η τιμή είναι μηδενική ο ασθενής επέζησε. Τα νοσοκομεία που έχουν μικρότερες τιμές αξιολογούνται ως καλύτερης ποιότητας στην παροχή φροντίδας.

Διαβάζουμε τα δεδομένα στο αρχείο Outcome of Care Measures.csv και εξετάζουμε τη δομή του. Δηλώνουμε αρχικά ότι όλα τα δεδομένα είναι character για να μην γίνει κάποια ανεπιθύμητη μετατροπή και αλλοιωθούν τα δεδομένα (π.χ. αν character/factor αρχίζει με 0 και η αυτόματη μετατροπή το θεωρήσει numeric το 0 θα εξαφανιστεί). Κάνουμε εμείς τη μετατροπή όταν χρειάζεται. Ο παρακάτω κώδικας είναι από script αρχείο και δεν φαίνεται η μεγάλη ποσότητα των ενδιάμεσων αποτελεσμάτων. Βλέπουμε ότι έχουμε 46 μεταβλητές/στήλες και 4706 μετρήσεις/γραμμές. Η 11η στήλη έχει όνομα Hospital 30 Day Death Mortality Rates from Heart Attack και αυτή μας ενδιαφέρει. Φτιάχνουμε το ιστόγραμμα και βλέπουμε πως αποθηκεύεται σε εξωτερικό αρχείο γραφικών.



Το ιστόγραμμα δείχνει ότι στα 2720 από τα 4706 νοσοκομεία (58%), οι περισσότεροι ασθενείς που εισάγονται με καρδιακή προσβολή καταλήγουν κατά μέσον όρο σε 15 περίπου ημέρες.

Ερώτημα 2

Ποιο είναι το καλύτερο νοσοκομείο σε κάποια πολιτεία;

Το καλύτερο νοσοκομείο σε μια πολιτεία στις ΗΠΑ θα είναι αυτό που θα έχει τους λιγότερους θανάτους για heart attack, heart failure ή pneumonia. Τα νοσοκομεία που δεν έχουν δεδομένα για αυτές τις περιπτώσεις δεν τα υπολογίζουμε στην ταξινόμηση.

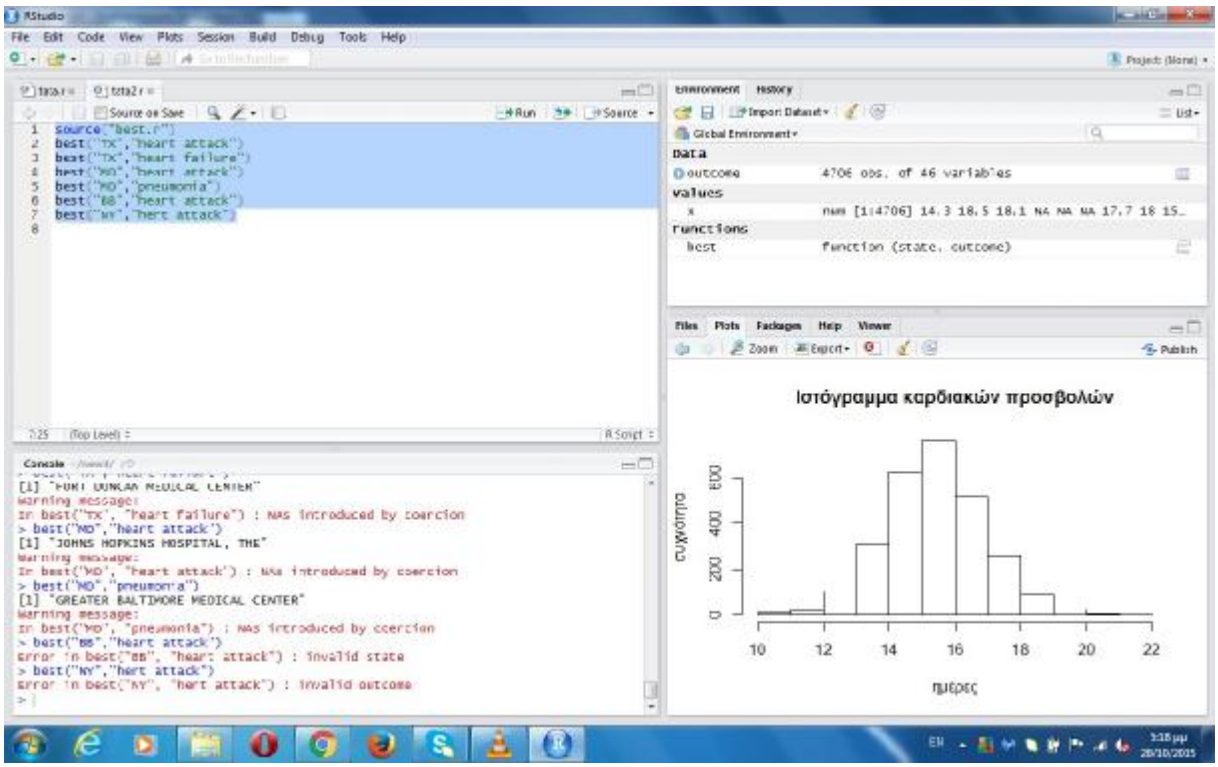
Φτιάχνουμε μια συνάρτηση best() με δυο παραμέτρους, την συντομογραφία της πολιτείας, state και την περίπτωση του ασθενούς, outcome. Η συνάρτηση θα πρέπει να ελέγχει την εγκυρότητα των παραμέτρων. Εάν η πολιτεία (state) είναι μη έγκυρη, δίνει το μήνυμα invalid state. Εάν η περίπτωση (outcome) είναι μη έγκυρη, δίνει το μήνυμα invalid outcome.

Η συνάρτηση best()

```
best <- function(state, outcome) {
   ## Read data     hdata <- read.csv("Outcome_of_Care_Measures.csv",
colClasses = "character")
...
```

```
## Check that state and outcome are valid
## Return hospital name in that state with lowest 30-day death rate


validState <- which(hdata[,7] == state)
if (length(validState) == 0) {
stop("invalid state")
} ....
if (outcome == "heart attack"){
    hdata[,11] <- as.numeric(hdata[,11])        # convert char to numeric
istate <- which(hdata[,7]==state)           # select state group        iclean
<- which(!is.na(hdata[istate,11]))    # clean out NAs        hmin <-
min(hdata[istate[iclean],11])       # minimum deaths in group        imin <-
which(hdata[istate[iclean],11]==hmin) # index of best        best1 <-
hdata[istate[iclean][imin],2]       # and the best is        best1
} else if (outcome == "heart failure"){
    hdata[,17] <- as.numeric(hdata[,17])        # convert char to numeric
istate <- which(hdata[,7]==state)           # select state group        iclean
<- which(!is.na(hdata[istate,17]))    # clean out NAs        hmin <-
min(hdata[istate[iclean],17])       # minimum deaths in group        imin <-
which(hdata[istate[iclean],17]==hmin) # index of best        best2 <-
hdata[istate[iclean][imin],2]       # and the best is        best2
} else if (outcome == "pneumonia") {
    hdata[,23] <- as.numeric(hdata[,23])        # convert char to numeric
istate <- which(hdata[,7]==state)           # select state group        iclean
<- which(!is.na(hdata[istate,23]))    # clean out NAs        hmin <-
min(hdata[istate[iclean],23])       # minimum deaths in group        imin <-
which(hdata[istate[iclean],23]==hmin) # index of best        best3 <-
hdata[istate[iclean][imin],2]       # and the best is        best3    } else {
    stop("invalid outcome")
} }
```

Τρέχουμε τη συνάρτηση με ορίσματα τον κωδικό της πολιτείας και τον δείκτη που εξετάζουμε για ορισμένες ενδεικτικές περιπτώσεις.



Δίνουμε επίτηδες και λάθος πολιτεία (BB - invalid state) καθώς και τυπογραφικό στο δείκτη θνησιμότητας (hert attack – invalid outcome).Τα αποτελέσματα συμφωνούν με

τις οδηγίες του assignment. Εμφανίζονται και Warning μηνύματα για NAs introduced by coercion τα οποία μπορούμε να αγνοήσουμε.

Ερώτημα 3

Ταξινόμηση νοσοκομείων βάσει δείκτου θνησιμότητας σε κάποια πολιτεία

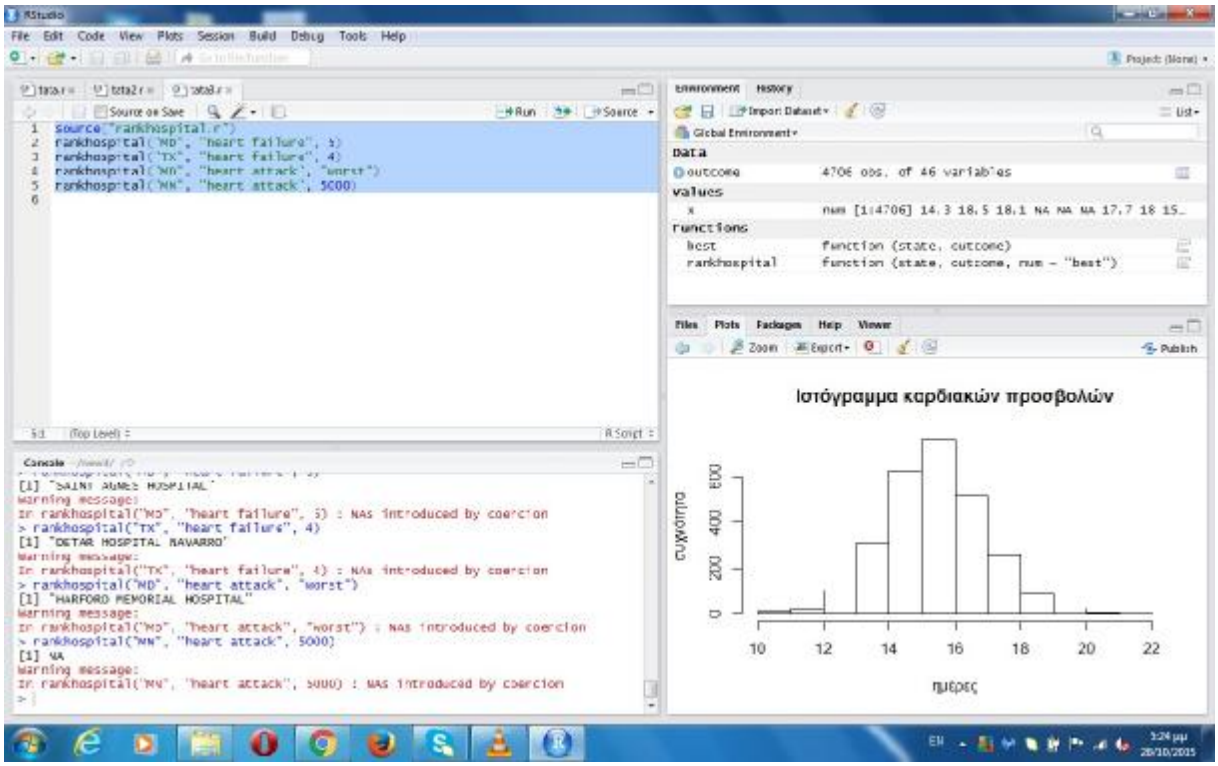
Εδώ φτιάχνουμε μια συνάρτηση rankhospital() με τρεις παραμέτρους. State, outcome και num την ταξινόμηση του νοσοκομείου σε εκείνο το state για αυτό το outcome.

Η συνάρτηση rankhospital()

```
rankhospital <- function(state, outcome, num = "best") {
   ## Read data
   hdata <- read.csv("Outcome_of_Care_Measures.csv", colClasses = "character")


   ## Check that state and outcome are valid
   ## Return hospital name in that state with the given rank 30-day death rate


   validState <- which(hdata[,7] == state)
   if (length(validState) == 0) {
stop("invalid state")
   }


   if (outcome == "heart attack"){
      hdata[,11] <- as.numeric(hdata[,11])         # convert char to numeric
istate <- which(hdata[,7]==state)            # select state group
      # order with tie breaking
      hos_ordered <- order(hdata[istate,11],hdata[istate,2],decreasing=FALSE,na.last=NA)
if (num == "best"){
         result <- hdata[istate,2][hos_ordered[1]]
      } else if (num == "worst"){
         result <- hdata[istate,2][hos_ordered[length(hos_ordered)]]
      } else {
         result <- hdata[istate,2][hos_ordered[num]]
      }
result


   } else if (outcome == "heart failure"){
      hdata[,17] <- as.numeric(hdata[,17])         # convert char to numeric
istate <- which(hdata[,7]==state)            # select state group
      # order with tie breaking
      hos_ordered <- order(hdata[istate,17],hdata[istate,2],decreasing=FALSE,na.last=NA)
if (num == "best"){
         result <- hdata[istate,2][hos_ordered[1]]
      } else if (num == "worst"){
         result <- hdata[istate,2][hos_ordered[length(hos_ordered)]]
      } else {
         result <- hdata[istate,2][hos_ordered[num]]
      }
      result


   } else if (outcome == "pneumonia") {
      hdata[,23] <- as.numeric(hdata[,23])         # convert char to numeric
istate <- which(hdata[,7]==state)            # select state group
      # order with tie breaking
      hos_ordered <- order(hdata[istate,23],hdata[istate,2],decreasing=FALSE,na.last=NA)
if (num == "best"){           result <- hdata[istate,2][hos_ordered[1]]
      } else if (num == "worst"){
         result <- hdata[istate,2][hos_ordered[length(hos_ordered)]]
      } else {
```

```
            result <- hdata[istate,2][hos_ordered[num]]
        }
        result


    }           else            {
print(outcome)
stop("invalid outcome")
    }
}
```

Δοκιμάζουμε για το 5ο νοσοκομείο στην πολιτεία Maryland (MD) ως προς heart failure, το 4ο στο Texas (TX), το χειρότερο στη Maryland και το 5000ο στη Minnesota (MN) το οποίο δεν υπάρχει και το πρόγραμμα μας βγάζει NA.



Τα αποτελέσματα πάλι συμφωνούν με τις οδηγίες του assignment. Εμφανίζονται και Warning μηνύματα για NAs introduced by coercion τα οποία μπορούμε να αγνοήσουμε.

Ερώτημα 4

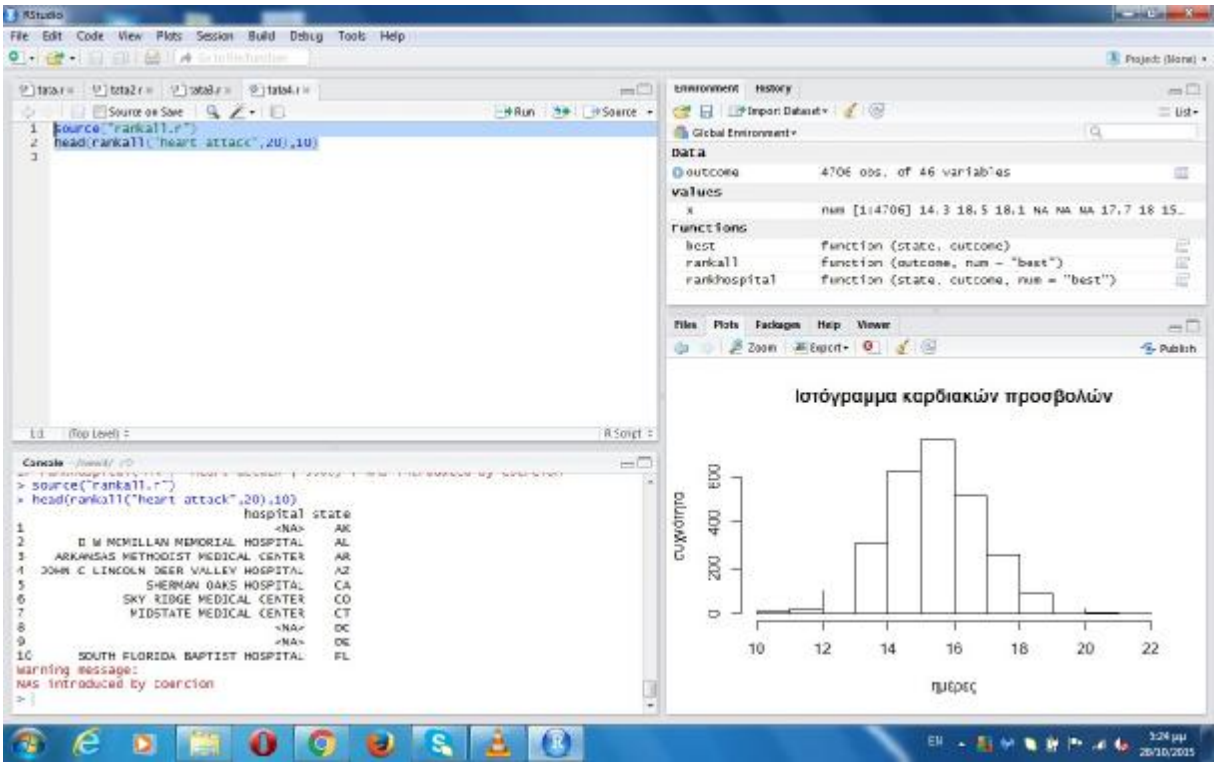
Ταξινόμηση νοσοκομείων σε όλες τις πολιτείες

Τέλος, φτιάχνουμε μια συνάρτηση rankall() με δυο παραμέτρους. Outcome και num. Η συνάρτηση εξετάζει τη βάση για τους δείκτες που αναφέραμε παραπάνω και επιστρέφει με το νοσοκομείο σε κάθε state που έχει τη σειρά num.

```
rankall <- function(outcome, num = "best") {
    ## Read data
    hdata <- read.csv("Outcome_of_Care_Measures.csv", colClasses = "character")


    ## Check that state and outcome are valid
```

```
   ## For each state, find the hospital of the given rank
   ## Return a data frame with the hospital names and the
   ## (abbreviated) state name


   allstates <- sort(unique(hdata[,7]))      N <- length(allstates) # they are 54      dframe
<- data.frame(hospital=rep(NA,N), state=rep("",2), stringsAsFactors=FALSE)
    i <-
0
   for (state in allstates){


       if (outcome == "heart attack"){
          hdata[,11] <- as.numeric(hdata[,11])         # convert char to numeric
istate <- which(hdata[,7]==state)            # select state group
          # order with tie breaking
          hos_ordered <- order(hdata[istate,11],hdata[istate,2],decreasing=FALSE,na.last=NA)
if (num == "best"){
             result <- hdata[istate,2][hos_ordered[1]]
          } else if (num == "worst"){
             result <- hdata[istate,2][hos_ordered[length(hos_ordered)]]
          } else {
             result <- hdata[istate,2][hos_ordered[num]]
          }


       } else if (outcome == "heart failure"){
          hdata[,17] <- as.numeric(hdata[,17])         # convert char to numeric
istate <- which(hdata[,7]==state)            # select state group
          # order with tie breaking
          hos_ordered <- order(hdata[istate,17],hdata[istate,2],decreasing=FALSE,na.last=NA)
if (num == "best"){
             result <- hdata[istate,2][hos_ordered[1]]
          } else if (num == "worst"){
             result <- hdata[istate,2][hos_ordered[length(hos_ordered)]]
          } else {
             result <- hdata[istate,2][hos_ordered[num]]
          }


       } else if (outcome == "pneumonia") {
          hdata[,23] <- as.numeric(hdata[,23])         # convert char to numeric
istate <- which(hdata[,7]==state)            # select state group
          # order with tie breaking
          hos_ordered <- order(hdata[istate,23],hdata[istate,2],decreasing=FALSE,na.last=NA)
if (num == "best"){
             result <- hdata[istate,2][hos_ordered[1]]
          } else if (num == "worst"){
             result <- hdata[istate,2][hos_ordered[length(hos_ordered)]]
          } else {
             result <- hdata[istate,2][hos_ordered[num]]
          }


       } else {
print(outcome)
stop("invalid outcome")
       }
       i <- i+1
       dframe[i,] <- c(result,state)
   }
   dframe
}
```

Τα αποτελέσματα πάλι συμφωνούν με τις οδηγίες του assignment. Εμφανίζονται και Warning μηνύματα για NAs introduced by coercion τα οποία μπορούμε να αγνοήσουμε.



ΣΥΜΠΕΡΑΣΜΑΤΑ

Μετά από λεπτομερειακή αναζήτηση πληροφοριών για τη χρήση του λογισμικού R για στατιστική ανάλυση βιοιατρικών δεδομένων, τα συμπεράσματα είναι τα εξής:

- η συγκεκριμένη πτυχιακή εργασία ασχολήθηκε κυρίως με τη χρήση του προγράμματος σε μια έρευνα σε νοσοκομεία για καρδιακά προβλήματα και πνευμονία. Θα μπορούσε όμως να αναφερθεί σε κάποιο άλλο θέμα όπως για «για χρήστες φακών επαφής» αλλά δεν έχει γίνει μια τόσο μεγάλη έρευνα,

- το πρόγραμμα έχει τη δυνατότητα να κάνει από απλές πράξεις (π.χ. κομπιουτεράκι) μέχρι πιο σύνθετες και εξειδικευμένες πράξεις-ασκήσεις με αξιόπιστα αποτελέσματα και σπάνια θα βγει μια λανθάνουσα απάντηση,

- μπορεί να χρησιμοποιηθεί για στατιστικές αναλύσεις, γραφήματα, μαθηματικές ασκήσεις, συναρτήσεις, διανύσματα, πίνακες και λίστες, και

- τέλος, η τεχνολογία έχει εξελιχθεί τόσο πολύ ανάλογα με τις ανθρώπινες ανάγκες και έτσι διευκολύνει πάρα πολύ την καθημερινότητα μας πληκτρολογώντας κάποια κουμπιά και όλα είναι έτοιμα.

Βιβλιογραφία

[1] Jeff Leek, Roger D. Peng, and Brian Caffo. *The Data Scientist's Toolbox*. https://www.coursera.org/course/datascitoolbox. July 2015.

[2] Jeff Leek, Roger D. Peng, and Brian Caffo. *R Programming*. https://www.coursera.org/course/course/rprog. July 2015.

[3] *R tutorials from scratch*. https://www.youtube.com/playlist?list=PLFAYD0dt5xCzTQHDhMPZwBoaAXWeVhZzg. Aug. 2015.