**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

# «Στατιστική Συμπερασματολογία με Χρήση της Γλώσσας Προγραμματισμού R»

## ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Της

**Τσιάντα Χρυσοβαλάντου**

(ΑΕΜ: 2733)

**Επιβλέπων: Βασιλειάδης Γεώργιος**

Επίκουρος Καθηγητής

Καστοριά Οκτωβριος 2024

Η παρούσα σελίδα σκοπίμως παραμένει λευκή

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

# Στατιστική Συμπερασματολογία με Χρήση της Γλώσσας Προγραμματισμού R

## ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Της

**Τσιάντα Χρυσοβαλάντου**

(ΑΕΜ: 2733)

**Επιβλέπων: Βασιλειάδης Γεώργιος**

Επίκουρος Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 2 Οκτωβρη 2024

| | | |
|---|---|---|
| Γεώργιος Βασιλειάδης<br>Επίκουρος Καθηγητής | Δημήτριος Βέργαδος<br>Επίκουρος Καθηγητής | Τουλόπουλος Δημήτριος<br>Επίκουρος Καθηγητής |

Καστοριά Οκτώβρης 2024

# Ευχαριστίες

# Περίληψη

Η συγκεκριμένη εργασία εκπονήθηκε στα πλαίσια του μαθήματος «Πτυχιακή Εργασία» για το ακαδημαϊκό χειμερινό εξάμηνο 2023-2024, με θέμα την γλώσσα προγραμματισμού R και την Στατιστική Συμπερασματολογία. Αποτελείται από ένα μεγάλο θεωρητικό κομμάτι όπου αναφέρονται πληροφορίες σχετικά με την γλώσσα R και από διάφορα πρακτικά παραδείγματα καθ'όλη την έκταση της εργασίας. Για την μελέτη και την συγγραφή του θεωρητικού μέρους, χρησιμοποιήθηκαν βιβλιογραφικές και διαδικτυακές πηγές και όσον αφορά το πρακτικό μέρος χρησιμοποιήθηκε το ελεύθερο λογισμικό R Studio.

Αποτέλεσμα αυτής της πτυχιακής εργασίας, είναι να έχουμε αποκτήσει μια βασική γνώση για την γλώσσα προγραμματισμού R και σε ποιους τομείς χρησιμοποιείται και μεγαλύτερη εξοικείωση στην δημιούργια διαφόρων παραδειγμάτων με την χρήση αυτής.

***Λέξεις Κλειδιά:***

Γλώσσα Προγραμματιμού R, Στατιστική Συμπερασματολογία, R Studio, Διανύσματα, Λίστες, Πίνακες, Πίνακες Πολλαπλών Διαστάσεων, Πλαίσια Δεδομένων, Παράγοντες, Περιγραφική Στατιστική, Δείγματα, Πίνακες Συνάφειας, Έλεγχος Υποθέσεων, Έλεγχος Καλής Προσαρμογής, Έλεγχος Ανεξαρτησίας.

# Abstract

This dissertation was prepared in the context of the course «Final Thesis» for the winter semester 2023-2024 on R programming language and Statistical Inference. It is consisted of a big theoretical part reporting information about the R programming language and a practical one with examples of mini scripts throughout the thesis. Literature reviews and online sources were used for both research and writing of the theoretical section. The open source software R Studio was used for the practical section as well.

As a result of this project we will be able to have some basic information about the R programming language and in which fields is preferred to be used and be more accustomed to creating some scripts in the R Studio.

***Key Words:***

R Programming Language, Statistical Inference, R Studio, Vectors, Lists, Matrices, Arrays, Data Frames, Factors, Descriptive Statistics, Samples, Relevance Tables, Hypothesis Testing, Good-Adaptation Testing, Independence Testing.

# Πίνακας Περιεχομένων

# Λίστα Εικόνων

# Λίστα Πινάκων

# Εισαγωγή

Το θέμα της παρούσας εργασίας είναι η γλώσσα προγραμματισμού R και η Στατιστική Συμπερασματολογία. Η γλώσσα R είναι μια ελεύθερου και ανοικτού κώδικα γλώσσα, η οποία χρησιμοποιείται σε διάφορους επιστημονικούς τομείς και όχι μόνο καθώς και σε πολλές από τις εφαρμογές που χρησιμοποιούν πλέον όλοι οι χρήστες της τεχνολογίας στην καθημερινότητα τους. Εφαρμόζεται τόσο από έναν εξειδικευμένο προγραμματιστή, όσο και από έναν απλό χρήστη με την βοήθεια του λογισμικού R Studio.

Στόχος της συγκεκριμένης εργασίας είναι η ενημέρωση σχετικά με την γλώσσα R, τι δυνατότητες προσφέρει, γιατί να μην προτιμήσουμε κάποια άλλη γλώσσα προγραμματισμού, καθώς επίσης τους τρόπους με τους οποίους μπορούμε να αναπτύξουμε διάφορα δείγματα και να εξάγουμε συμπεράσματα και στοιχεία μέσα από αυτά.

Στο πρώτο μέρος της εργασίας, θα εξετάσουμε τι είναι η γλώσσα προγραμματισμού R και πότε δημιουργήθηκε, μέσα από μια σύντομη ιστορική αναδρομή, ποια είναι τα πλεονεκτήματα και τα μειονεκτήματα που πρέπει να λάβει υπόψιν του ο κάθε χρήστης πριν την επιλέξει καθώς θα δούμε πως είναι και το περιβάλλον του R Studio. Επιπλέον, θα αναφέρουμε πληροφορίες για το συγκεκριμένο λογισμικό όσον αφορά την δημιουργία και ανάπτυξη του. Περαιτέρω, θα κάνουμε μια αναφορά σε διάφορα πακέτα και βιβλιοθήκες της R, όπου κάθε μία εξειδικεύεται σε διαφορετικό τύπο χρηστών και απευθύνεται για συγκεκριμένες υπηρεσίες. Τέλος, θα κατηγοριοποιήσουμε τα αντικείμενα δεδομένων και τις δομές δεδομένων της γλώσσας, θα δούμε μέσα από πρακτικά παραδείγματα πως μπορούμε να δημιουργήσουμε/τροποποιήσουμε/επεξεργαστούμε τις παραπάνω δομές καθώς και που χρησιμοποιείται η κάθε μια.

Στο δεύτερο μέρος της εργασίας, θα παρουσιάσουμε διάφορες πληροφορίες για το κομμάτι της Στατιστικής Συμπερασματολογίας, τι είναι η Περιγραφική Στατιστική, θα αναλύσουμε συμπερασματολογία για ένα αλλά και παραπάνω δείγματα δεδομένων και τέλος θα δούμε τι είναι οι Πίνακες Συνάφειας, ο Έλεγχος Καλής Προσαρμογής, ο Έλεγχος Υποθέσεων και ο Έλεγχος Ανεξαρτησίας. Καθ'όλη την διάρκεια της εργασίας ακολουθούν παραδείγματα με την χρήση του λογισμικού R Studio.

# 1. Η Γλώσσα Προγραμματισμού R

## 1.1 Εισαγωγή

### 1.1.1 Ορισμός της R

Η R είναι μια ανοικτού κώδικα γλώσσα προγραμματισμού αλλά και ένα περιβάλλον κατάλληλο για στατιστικούς υπολογισμούς, δημιουργία γραφημάτων καθώς και για ανάλυση δεδομένων. Διανέμεται σαν ελεύθερο λογισμικό από το Ίδρυμα Ελεύθερου Λογισμικού γνωστό ως FSF (Free Software Foundation) και διατίθεται κάτω από την άδεια GNU General Public License. Ως εκ τούτου, μπορεί να θεωρηθεί και ως GNU λογισμικό, το οποίο είναι κυρίως γραμμένο σε C, Fortran αλλά και αυτούσιο R κώδικα. Ο όρος αυτός υποδηλώνει ότι ο πηγαίος κώδικας της γλώσσας, είναι δωρεάν και προσβάσιμος από όλους τους τύπους χρηστών, δίνοντας τους έτσι την ευκαιρία να μπορούν να βασίζονται σε αυτόν για οποιαδήποτε μακροπρόθεσμη ή βραχυπρόθεσμη μελέτη. Οι προγραμματιστές από την πλευρά τους, μπορούν να εξετάσουν τον τρόπο με τον οποίο λειτουργεί ο κώδικας και να κάνουν τις απαραίτητες αλλαγές σε δυσλειτουργικές ή προβληματικές πλευρές του, ώστε να είναι αποδοτικότερος.

Παρέχει ένα μεγάλο εύρος στατιστικών τεχνικών στις οποίες περιλαμβάνονται:

- Γραμμική και Μη Γραμμική Μοντελοποίηση (Linear and Nonlinear Modelling),
- Γραμμική Παλινδρόμηση (Linear Regression),
- Κλασσικές Στατιστικές Δοκιμές (Classical Statistical Tests),
- Ανάλυση Χρονοσειρών (Time-Series Analysis) και πολλών ακόμα.

Κατατάσσεται στις συναρτησιακές γλώσσες, όμως το περιβάλλον της συμπεριλαμβάνει και ορισμένα εργαλεία υποστήριξης αντικειμενοστραφούς προγραμματισμού, για πιο πολύπλοκες υπολογιστικές διεργασίες. Επιπλέον, διαθέτει και διάφορα εργαλεία τα οποία είναι αναγκαία για μια στατιστική ανάλυση. Συνοπτικά, κάποια από αυτά είναι:

1. Δημιουργία Τυχαίων Δειγμάτων.
2. Διακριτές και Συνεχείς Μεταβλητές (Poisson, Gamma).
3. Έλεγχοι Υποθέσεων.
4. Στατιστικά Τεστ (Kolmogorov-Smirnoff).
5. Δημιουργία Γραφημάτων (ιστόγραμμα, qq plot, pie chart, bar chart).

## 1.2 Ιστορική Αναδρομή

Η γλώσσα προγραμματισμού R, εφαρμόστηκε για πρώτη φορά στις αρχές του 1990 στο Πανεπιστήμιο του Auckland της Νέας Ζηλανδίας. Αφετηρία για την ανάπτυξη της αποτέλεσε η γλώσσα S, η οποία χρησιμοποιούνταν τόσο για στατιστική ανάλυση όσο και σχεδίαση γραφικών. Η S αναπτύχθηκε το 1976 στα εργαστήρια Bell Laboratories από τον John Chambers, όμως έγινε δημοσίως διαθέσιμη στις αρχές του 1980. Γνώρισε μεγάλη άνθηση στα τέλη της ίδιας δεκαετίας έως τις αρχές της δεκαετίας του 1990, αφού θεωρούνταν μια γλώσσα υψηλού επιπέδου για τα δεδομένα εκείνης της εποχής. Μέχρι τότε, μεγάλο μέρος της στατιστικής ανάλυσης πραγματοποιούνταν καλώντας άμεσα μια καλώς-δομημένη βιβλιοθήκη της Fortran, γνωστή ως SCS (Statistical Computing Subroutines).

Η βιβλιοθήκη SCS είχε αναπτυχθεί με την πάροδο των χρόνων για να παρέχει μια ευέλικτη βάση σε πολλές στατιστικές έρευνες. Λόγω του ότι τα τμήματα στατιστικής έρευνας ήταν επιφορτισμένα με την ανάπτυξη νέων μεθοδολογιών, συχνά για την αντιμετώπιση μη τυπικών καταστάσεων δεδομένων, η ικανότητα εκτέλεσης των σωστών υπολογισμών ήταν υψίστης σημασίας. Έτσι, η S σχεδιάστηκε για να προσφέρει μια εναλλακτική και πιο δια δραστική προσέγγιση, παρόλο που οι πρώιμες εκδόσεις της δεν περιείχαν συναρτήσεις ειδικές για στατιστική μοντελοποίηση. Οι ιδρυτές της ήθελαν να είναι μια γλώσσα η οποία θα έκανε την ανάλυση δεδομένων ευκολότερη και αποτελεσματικότερη, καθιστώντας την διαχειρίσιμη από έναν απλό χρήστη έως έναν πιο εξειδικευμένο.

Το 1984 ο πηγαίος κώδικας της αδειοδοτήθηκε από την εταιρεία AT&T Software Sales, κυρίως για εκπαιδευτικούς και επιχειρηματικούς σκοπούς. Μέχρι το 1988, η γλώσσα S υπέστη αρκετές αλλαγές, κυρίως στην σύνταξη της. Μετά την δημοσίευση της καινούργιας και πιο συγκεκριμένα τρίτης έκδοσης της, πολλοί από τους χρήστες δυσκολευόταν να την κατανοήσουν σε βάθος καθώς αρκετές από τις εντολές της

έπρεπε να ανασυνταχθούν. Μια από τις βασικότερες αλλαγές που επικράτησαν ήταν η επανεγγραφή πολλών εσωτερικών λειτουργιών από την γλώσσα Fortran στην γλώσσα C. Επίσης, την ίδια χρονιά ξεκίνησε να μετατρέπεται στο εμπορικό πακέτο S- PLUS. Η τέταρτη έκδοση, η οποία είναι αυτή που χρησιμοποιούμε μέχρι και σήμερα, έγινε διαθέσιμη το 1998.

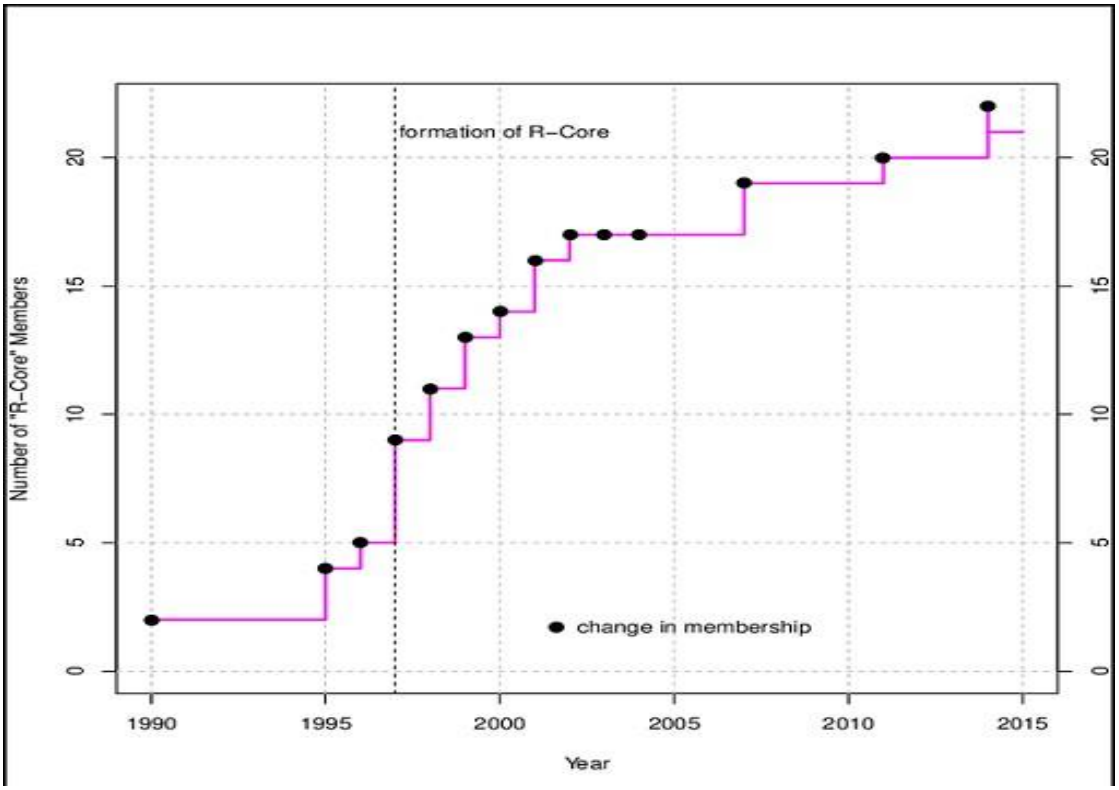
Το 1991, οι Ross Ihaka και Robert Gentleman έγραψαν την αρχική έκδοση ενός νέου πακέτου λογισμικού της S. Βέβαια, στα χρόνια που ακολούθησαν, υπήρξαν κι' άλλες ομάδες προγραμματιστών οι οποίες συνείσφεραν σημαντικά στην ανάπτυξη αυτού του πακέτου. Ήταν εξ ολοκλήρου ανεξάρτητο από το S-PLUS και κυκλοφόρησε για πρώτη φορά το 1993 ως γλώσσα R. Οι δημιουργοί του έδωσαν το συγκεκριμένο όνομα βασιζόμενοι στα αρχικά των ονομάτων τους, αλλά και στο λογοπαίγνιο, λόγω του μονογράμματος του πρωτεύοντος πακέτου. Μολονότι η R βασίστηκε στην S, έχει ανάλογα χαρακτηριστικά, χωρίς όμως να είναι πανομοιότυπη με αυτήν. Από τις κυριότερες διαφορές τους αποτέλεσε ο τρόπος διαχείρισης της διαθέσιμης μνήμης του υπολογιστή από την κάθε μία, ενώ ταυτόχρονα, η πρώτη μπορούσε να εκτελέσει τις ίδιες διεργασίες με την δεύτερη, αλλά σε μικρότερο κομμάτι κώδικα.

Οι Ihaka και Gentleman, πεπεισμένοι από τον Martin Macler, ξεκίνησαν να διανέμουν το πακέτο αυτό δωρεάν, δημιουργώντας συγχρόνως και μια λίστα χρηστών. Η φιλοσοφία αυτής της λίστας ήταν να έχουν όλοι οι χρήστες την δυνατότητα να επεμβαίνουν στον πηγαίο κώδικα, είτε για να βελτιώσουν προγραμματιστικά σφάλματα είτε για να κάνουν όποια τροποποίηση θεωρούσαν απαραίτητη. Η ανάπτυξη της γλώσσας δεν άργησε να έρθει. Πιο συγκεκριμένα, όλο και περισσότεροι χρήστες προσπαθούσαν να στείλουν και να ενσωματώσουν τους κώδικες τους. Η διαρκής αυτή συνεισφορά προκάλεσε τον διαχωρισμό της λίστας σε δύο μικρότερες, με την μια να εξειδικεύεται στην συνεχής εξέλιξη και την άλλη για οποιαδήποτε βοήθεια.

Έτσι, το 1995 καθιερώθηκε ως ελεύθερο και ανοικτού κώδικα λογισμικό και διατέθηκε υπό την άδεια GNU GPL. Το γεγονός αυτό, αρχικά δημιούργησε περιορισμούς σε εμπορικές ή εταιρικές χρήσεις, διότι δεν ήταν εμφανές που άνηκε η ευθύνη σε περίπτωση βλάβης. Το 1996 δημιουργήθηκε το πρώτο αποθετήριο χρηστών για προγράμματα που βασίζονταν σε κώδικα R, ενώ, από το 1997, η R Core

Group διαχειρίζεται το έργο της. Την ίδια περίοδο, ανακοινώθηκε επίσημα και το Ολοκληρωμένο Δίκτυο Αρχειοθέτησης της R (Comprehensive R Archive Network ή CRAN), το οποίο συμπεριελάμβανε μόλις 12 πακέτα. Τέλος, τον Φεβρουάριο του 2000, διατέθηκε ελεύθερα η πρώτη έκδοση της, γνωστή ως 1.0.

Στην εικόνα 1, παρατηρείται τόσο η ανάπτυξη των μελών της ομάδας R Core Group όσο και η ανοδική πορεία του έργου της ομάδας αυτής από το 1990 έως και το 2015.

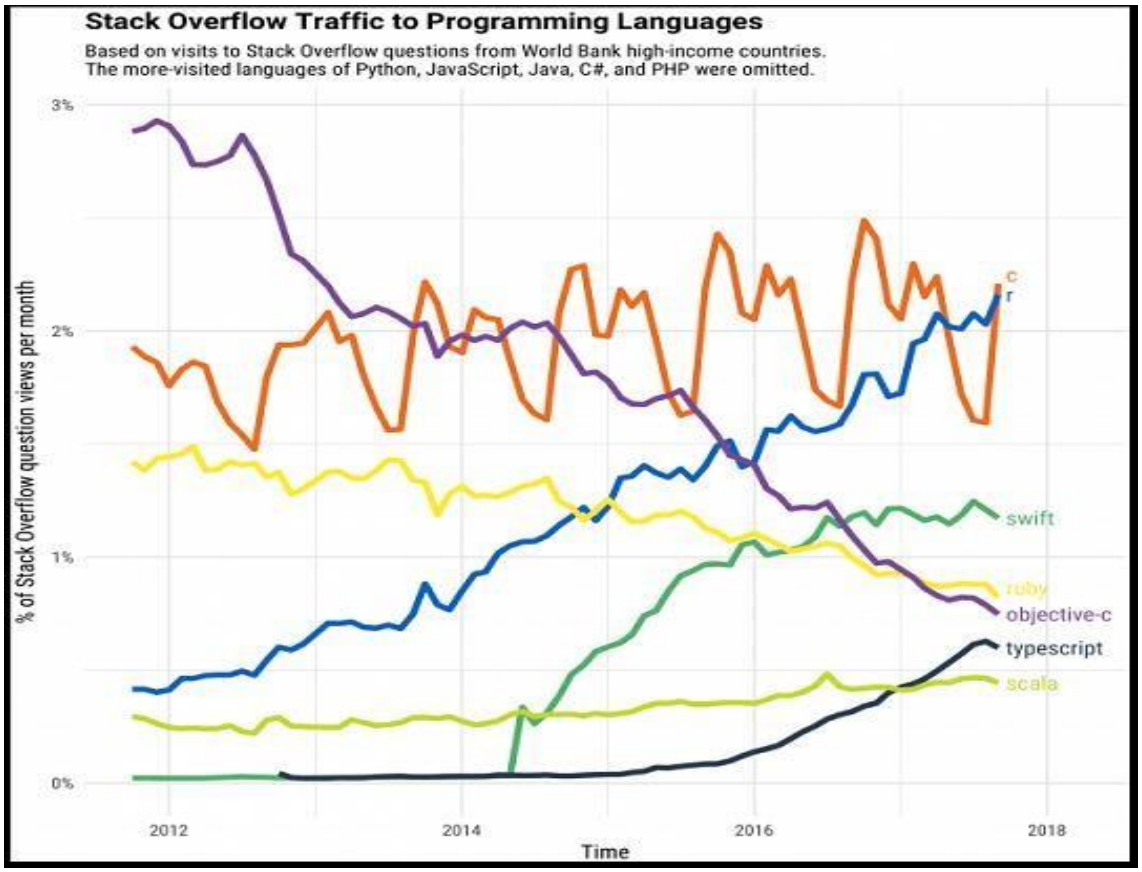



*Εικόνα 1: Ανάπτυξη των μελών και του έργου της ομάδας R Core Group.*

Η R εξακολουθεί να θεωρείται μια δωρεάν και ανοικτού κώδικα γλώσσα, ενώ νέες βελτιώσεις, προγράμματα και διανομές της συνεχίζουν μέχρι και σήμερα. Η πλειοψηφία των εταιρειών προσφέρει πλέον πιστοποίηση, ώστε να αναλάβουν την ευθύνη οποιασδήποτε βλάβης, οδηγώντας συνεχώς σε μεγαλύτερη χρήση. Τον Μάρτιο του 2015, τα διαθέσιμα πακέτα ξεπερνούσαν τα 6.200, καθώς η ολοένα και περισσότερη εξέλιξη της γλώσσας συνέβαλε στην ανάπτυξη νέων μεθόδων και καινούργιων εργαλείων. Από τον Σεπτέμβριο του 2018, έχουν δημοσιευτεί πάνω από 15.000 πακέτα, τα οποία διανέμονται ελεύθερα μέσω CRAN, GitHub και αρκετών ακόμα δημόσιων ή ιδιωτικών αποθετηρίων στο Διαδίκτυο.

Αξίζει επίσης να σημειωθεί, πως πολλοί οργανισμοί, ανάμεσα στους οποίους Google, LinkedIn και Facebook προτιμούν την συγκεκριμένη γλώσσα για ανάλυση δεδομένων. Ταυτόχρονα, εφαρμόζεται και σε πολλούς επιστημονικούς κλάδους. Παραδείγματα αυτών αποτελούν η οικονομία, το μάρκετινγκ, οι θετικές επιστήμες και οι τομείς υγείας. Η νεότερη έκδοση της γλώσσας, η οποία είναι η 4.0, ανακοινώθηκε τον Απρίλιο του 2020.

Κλείνοντας, η R αλληλεπιδρά πλέον και μέσω της διεπαφής εντολών (command line interface), ενώ παράλληλα και με περιβάλλον ανάπτυξης IDE. Τα λογισμικά αυτά αναπτύσσονται όλο και περισσότερο, με πιο ευρέως γνωστό το RStudio, για το οποίο θα αναφερθούν περαιτέρω πληροφορίες στην συνέχεια της εργασίας. Τέλος, στην εικόνα 2, παρατηρείται η όλο και αυξανόμενη επισκεψιμότητα της R στο Stack Overflow, σε σχέση με άλλες γλώσσες προγραμματισμού, από το 2012 έως και το 2018. Η απεικόνιση της φαίνεται με μπλε χρώμα.




*Εικόνα 2: Απεικόνιση της αυξανόμενης επισκεψιμότητας της R στο Stack Overflow.*

## 1.3 Οφέλη & Περιορισμοί

### 1.3.1 Πλεονεκτήματα

- ✓ Μπορεί να εκτελέσει λειτουργίες σε διανύσματα, πίνακες και διάφορα άλλα αντικείμενα δεδομένων ανεξαρτήτου μεγέθους.

- ✓ Υποστηρίζεται από λειτουργικά συστήματα Windows, MacOS, αλλά και από αρκετές διανομές συστημάτων UNIX.

- ✓ Είναι υψηλής συμβατότητας, επιτρέποντας στον χρήστη να την διαχειρίζεται εν συναρτήσει με άλλα εργαλεία όπως:
  - Γλώσσες Προγραμματισμού (Python, Java, C /C++)
  - Πακέτα Λογισμικού Στατιστικής Ανάλυσης (SPSS)
  - Προγράμματα Λογιστικών Φύλλων (Excel, Libreoffice Calc)
  - Σύστηματα Διαχείρισης Βάσεων Δεδομένων (Access)

- ✓ Διαθέτει πακέτα κατάλληλα για την μετατροπή μη ομαδοποιημένων δεδομένων (messy data) σε μια πιο ολοκληρωμένη και δομημένη μορφή. Η διαδικασία αυτή είναι γνωστή ως data wrangling.

- ✓ Είναι ιδανική για δημιουργία πιο «ελκυστικών» σχεδιαστικά γραφημάτων, συγκριτικά με άλλες γλώσσες προγραμματισμού, υλοποιημένων με ειδικές βιβλιοθήκες.

- ✓ Συμβάλλει στην ανάπτυξη λειτουργιών μηχανικής μάθησης (machine learning operations), με κυριότερες τόσο την παλινδρόμηση (regression) όσο και την κατηγοριοποίηση (classification).

- ✓ Κάνει χρήση του Rd, ενός δικού της συστήματος συγγραφής και τεκμηρίωσης κειμένου, το οποίο είναι παρόμοιας μορφής με το LaTeX. Μπορεί να εφαρμοστεί εγγράφως αλλά και μέσω διαδικτύου.

### 1.3.2 ΜειΟΝεκτήματα

- Όπως προαναφέραμε και στην ιστορική αναδρομή, βάση για την δημιουργία της γλώσσας R αποτέλεσε η S. Ωστόσο, η δεύτερη θεωρείται μια «παλιά» γλώσσα προγραμματισμού. Το ίδιο ισχύει και για το κύριο πακέτο της. Το γεγονός αυτό, φέρει αρνητικό αντίκτυπο στην R, αφού πρακτικά σημαίνει πως δεν έρχεται με εκτεταμένη υποστήριξη για 3D ή δυναμικά γραφικά. Για να καταστεί δυνατή αυτή η επιλογή, πρέπει να εγκατασταθούν επιπλέον πακέτα.

- Η πλειοψηφία των γλωσσών προγραμματισμού, ανάμεσα τους και η R, στερείται της βασικής ασφάλειας. Ως αποτέλεσμα, δεν μπορεί να ενσωματωθεί σε διαδικτυακές εφαρμογές και λόγω αυτού, δημιουργούνται αρκετοί περιορισμοί.

- Δεν υφίσταται για ανάλυση μεγάλων δεδομένων (Big Data), διότι καταλαμβάνει πολύ χώρο στη μνήμη.

- Είναι case sensitive, επιτρέπει δηλαδή, την διαφοροποίηση μεταξύ πεζών και κεφαλαίων γραμμάτων.

- Θεωρείται μια περίπλοκη γλώσσα. Συνέπεια αυτού, είναι να υπάρχει δυσκολία εκμάθησης, ειδικά για χρήστες οι οποίοι δεν έχουν ιδιαίτερη επαφή με τον προγραμματισμό.

- Σε αντίθεση με την Python ή την Matlab, η R είναι μια «αργή» γλώσσα. Άρα, ως προς τον χρόνο εκτέλεσης των εντολών, δεν είναι αρκετά αποδοτική.

- Διαθέτει ένα εύρος αλγορίθμων, κατανεμημένων σε διάφορα πακέτα. Προγραμματιστές, οι οποίοι δεν είναι εξοικειωμένοι με τα πακέτα αυτά, ενδεχομένως να μην μπορούν να υλοποιήσουν τους συγκεκριμένους αλγόριθμους με ευκολία, με αποτέλεσμα να μην είναι και οι ίδιοι παραγωγικοί.

## 1.4 Εισαγωγή στο RStudio

### 1.4.1 Ορισμός & Λίγα Λόγια για το RStudio

Το RStudio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE), κατάλληλο για ανάλυση δεδομένων, με χρήση της γλώσσας προγραμματισμού R, και όχι μόνο. Πρόκειται για ένα ανοικτού κώδικα λογισμικό, του οποίου η εφαρμογή ξεκίνησε τον Δεκέμβριο του 2010, όμως η πρώτη του έκδοση δημοσιεύτηκε επίσημα τον Φεβρουάριο του 2011. Η ανάπτυξη του πραγματοποιείται από την εταιρεία RStudio Inc, η οποία αργότερα, έχοντας ως ιδρυτή της τον J.J.Allaire, μετονομάστηκε και πλέον είναι γνωστή ως RStudio PBC. Διατίθεται με την άδεια GNU Affero General Public License v3 (AGPLv3), όπου εγγυάται την ελεύθερη χρήση/τροποποίηση/διανομή του πηγαίου κώδικα. Είναι κυρίως υλοποιημένο σε Java, όμως συμπεριλαμβάνει κομμάτια τα οποία οι κατασκευαστές του έχουν δημιουργήσει και αναπτύξει σε C++ και JavaScript.

Απαρτίζεται από πολλά δικά του πακέτα, πέρα από αυτά που έχει ήδη η γλώσσα R. Περιλαμβάνει εργαλεία για έλεγχο και εντοπισμό σφαλμάτων, σχεδίαση γραφημάτων, ιστορικό, βοήθεια, καθώς επίσης, υποστηρίζει άμεση εκτέλεση κώδικα. Επιπλέον, προσφέρεται στον χρήστη σε διάφορες διανομές, όπου η κάθε μία εξειδικεύεται σε διαφορετικό τύπο χρηστών είτε για προσωπική είτε για εμπορική χρήση. Οι πιο γνωστές είναι:

- RStudio Desktop,

  όπου εγκαθίσταται και δουλεύει τοπικά στον υπολογιστή όπως οι περισσότερες εφαρμογές και

- RStudio Server/Server Pro,

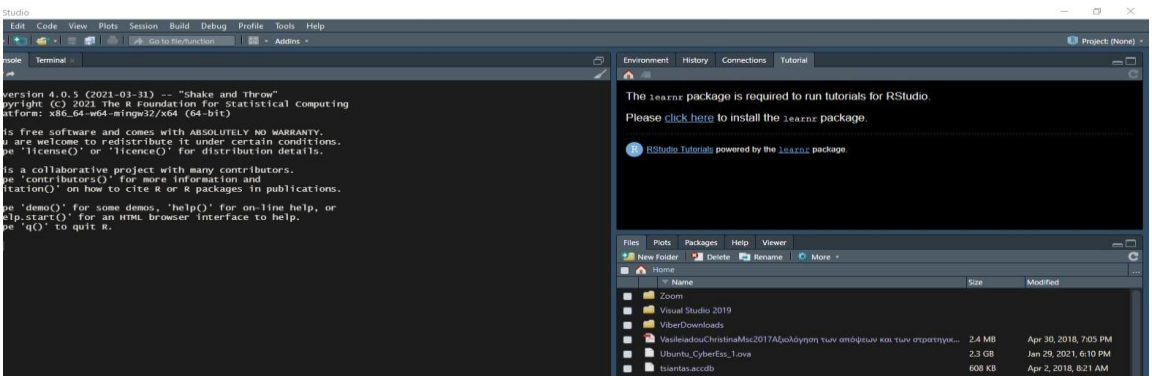
  το οποίο εκτελείται σε απομακρυσμένο διακομιστή (server), επιτρέποντας την πρόσβαση στο RStudio, χρησιμοποιώντας έναν οποιοδήποτε φυλλομετρητή (web browser).

Ακόμη, μπορεί να εγκατασταθεί σε μεγάλη ποικιλία λειτουργικών συστημάτων ανάλογα με τις παραπάνω εκδόσεις. Πιο συγκεκριμένα, σε Windows, Linux και MacOS για την πρώτη, αλλά και Debian/Ubuntu, Red Hat/CentOS, openSUSE και SUSE Linux για την δεύτερη. Η νεότερη έκδοση που χρησιμοποιούμε σήμερα, τόσο

για το RStudio Desktop, όσο και το RStudio Server είναι η 1.4.1106, όπου ανακοινώθηκε τον Μάρτιο του 2021.

### 1.4.2 Περιγραφή Περιβάλλοντος RStudio

Όπως αναφέρθηκε και στην ενότητα 1.2 (Ιστορική αναδρομή), η γλώσσα R λειτουργεί μέσω cli αλλά και με ολοκληρωμένα περιβάλλοντα ανάπτυξης (IDE). Στην συγκεκριμένη ενότητα, θα αναπτυχθεί μόνο το δεύτερο κομμάτι δηλαδή αυτό του λογισμικού RStudio, χρησιμοποιώντας την έκδοση RStudio Desktop, καθώς και τι περιλαμβάνει μέσα το περιβάλλον. Όσον αφορά τον προγραμματισμό της R, θα ακολουθήσει σε επόμενο κεφάλαιο. Με μια πρώτη ματιά, όταν ο χρήστης ανοίγει το λογισμικό βλέπει πως αποτελείται από τρία διαφορετικά παράθυρα. Ένα στιγμιότυπο του λογισμικού παρουσιάζεται παρακάτω στην εικόνα 3.



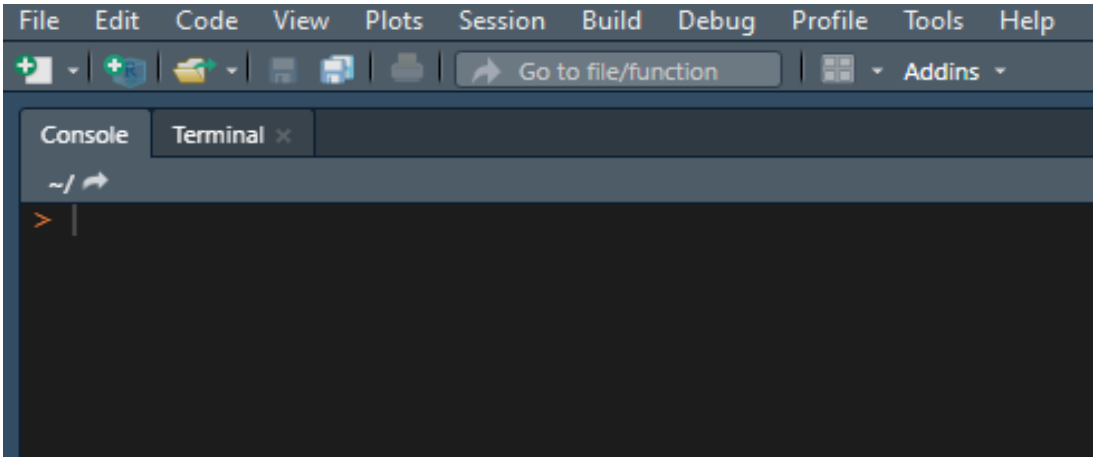

*Εικόνα 3: Στιγμιότυπο του λογισμικού RStudio*

Αρχικά, στην αριστερή πλευρά υπάρχει το κεντρικό μενού το οποίο αποτελείται από τις επιλογές «File», «Edit», «Code», «View», «Plots», «Session», «Build», «Debug», «Profile», «Tools» και «Help». Η κάθε επιλογή περιλαμβάνει αντιστοίχως τις δικές τις λειτουργίες, με πιο βασική την πρώτη, αφού από εκεί ανοίγουν είτε καινούργια, είτε προ υπάρχοντα αρχεία. Έπειτα, διακρίνεται το πρώτο παράθυρο της αριστερής πλευράς, στο οποίο υπάρχουν δύο καρτέλες:

- Console: όπου είναι το βασικό μας παράθυρο, αφού εκεί εκτελούνται οι εντολές (μετά το σύμβολο “>”) και έπειτα τυπώνονται τα αντίστοιχα αποτελέσματα. Δίπλα από την καρτέλα Console, υπάρχει η καρτέλα

- Terminal: όπου είναι κατάλληλη για shell scripts.

Στην εικόνα 4, παρουσιάζεται ένα στιγμιότυπο του πρώτου τμήματος με το μενού και τις παραπάνω καρτέλες.

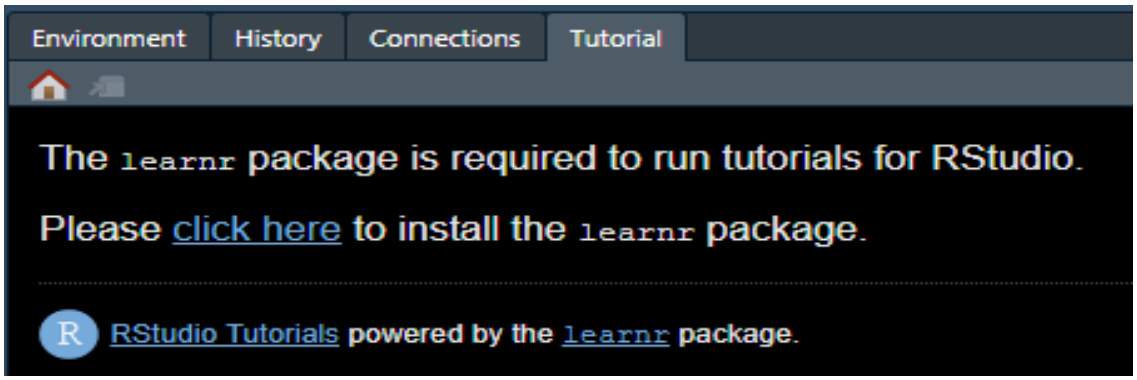



*Εικόνα 4: Στιγμιότυπο αριστερού τμήματος του RStudio.*

Στην δεξιά πλευρά του λογισμικού, φαίνεται πως υπάρχουν δύο ξεχωριστά τμήματα, δηλαδή δύο διαφορετικά παράθυρα. Στο παράθυρο του επάνω τμήματος, υπάρχουν τέσσερις καρτέλες:

- Environment: όπου εμφανίζονται όλες οι μεταβλητές που έχουν χρησιμοποιηθεί στον κώδικα,
- History: όπου υπάρχει όλο το ιστορικό των εντολών που εκτελέστηκαν καθ' όλη την διάρκεια που ο προγραμματιστής δουλεύει στο RStudio,
- Connections: δείχνει τις ενεργές και μη συνδέσεις, που πραγματοποιούνται σε διαφορετικές πηγές δεδομένων, καθώς επίσης τους τύπους των παραπάνω συνδέσεων αλλά και τις ιδιότητες των δεδομένων αυτών. Τέλος, στη καρτέλα
- Tutorial: υπάρχουν ποίκιλα εκπαιδευτικά υλικά τα οποία ο χρήστης έχει την δυνατότητα να συμβουλευτεί. Για να επιτευχθεί αυτό, είναι απαραίτητη η εγκατάσταση του πακέτου learnr.

Στο στιγμιότυπο της εικόνας 5, διακρίνεται το επάνω τμήμα με τις καρτέλες και πιο συγκεκριμένα η τελευταία, στην οποία διακρίνεται ξεκάθαρα το μήνυμα εγκατάστασης του πακέτου που προαναφέρθηκε. Οι υπόλοιπες καρτέλες προς στιγμήν, είναι κενές, εφόσον δεν έχει εκτελεστεί ακόμα κάποιος κώδικας.

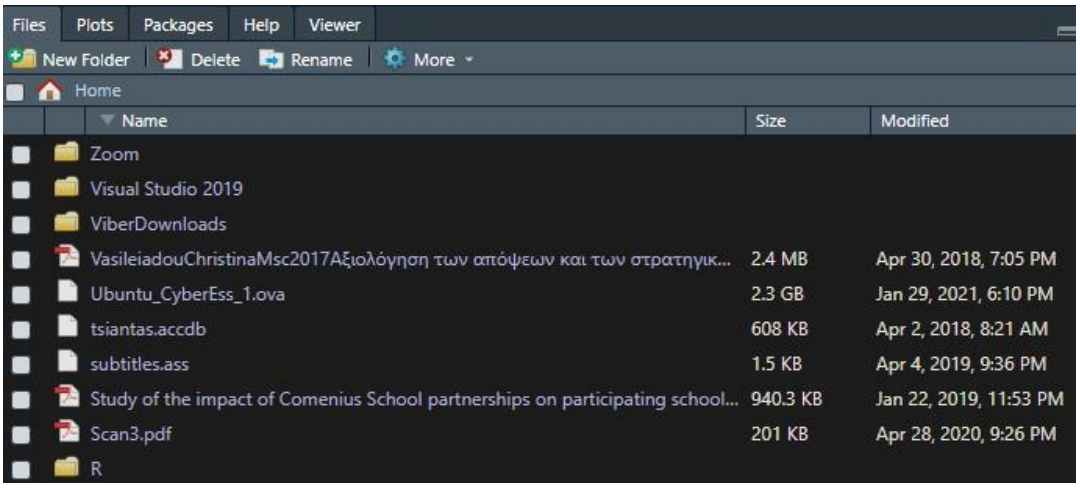



*Εικόνα 5:Στιγμιότυπο δεξιού τμήματος (επάνω τμήμα).*

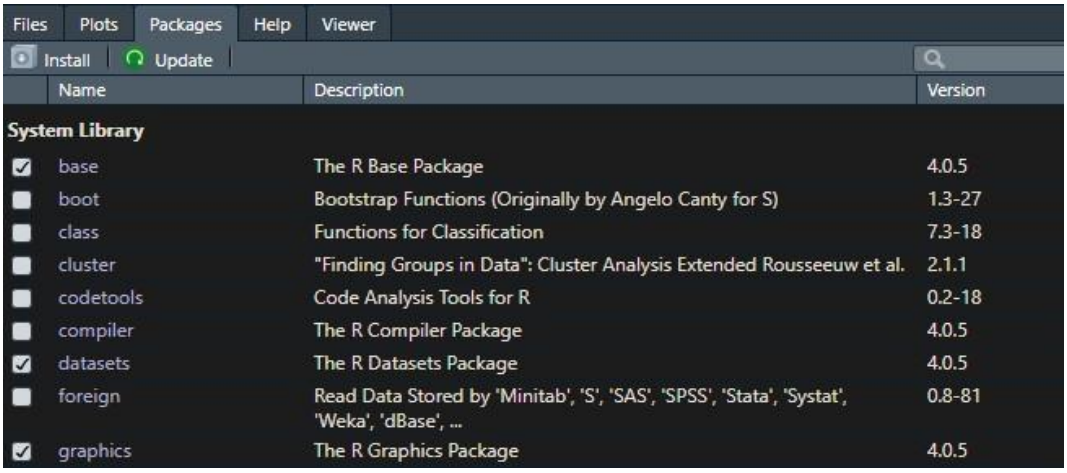
Στην συνέχεια, στο παράθυρο του κάτω τμήματος του δεξιού μέρους του λογισμικού, υπάρχουν πέντε καρτέλες:

- Files: όπου εμφανίζονται όλα τα διαθέσιμα αρχεία και φάκελοι που μπορούν να φορτωθούν στο περιβάλλον εργασίας της R (βλ. εικόνα 6),
- Plots: εκεί δημιουργούνται οι γραφικές παραστάσεις κάθε φορά που εκτελείται ένα πρόγραμμα,
- Packages: όπου φαίνονται όλα τα πακέτα τα οποία είναι ήδη εγκατεστημένα στο λογισμικό, αλλά και ποια είναι διαθέσιμα για μελλοντική εγκατάσταση και μετέπειτα χρήση (βλ. εικόνα 7),
- Help: όπου προσφέρει οποιαδήποτε βοήθεια, είτε για σύνταξη εντολών είτε παρέχοντας χρήσιμα links με οδηγίες κλπ. (βλ. εικόνα 8) και τέλος η καρτέλα
- Viewer: προβάλλει διαδικτυακές εφαρμογές ή σελίδες για τις οποίες γίνεται αναφορά μέσα στο πρόγραμμα.

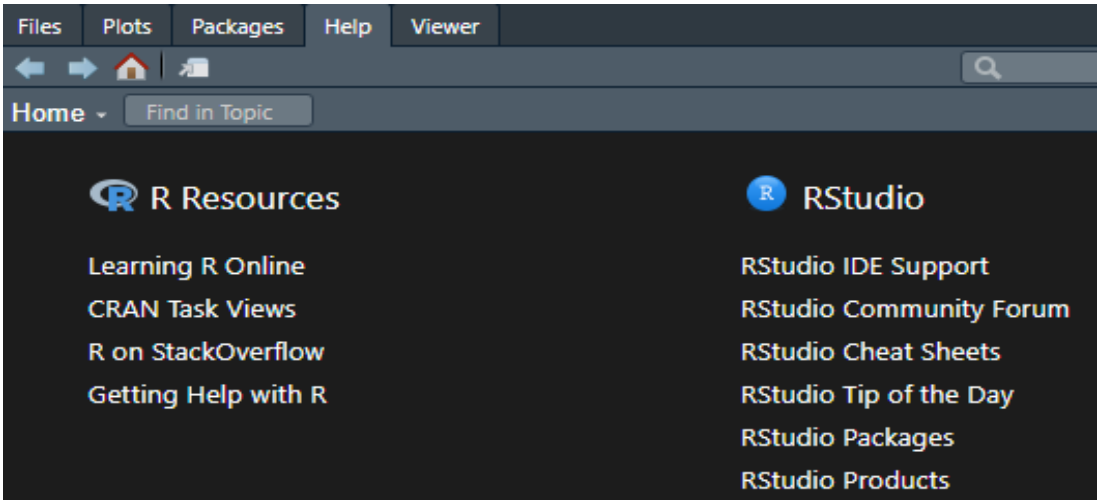
Στις καρτέλες Viewer και Plots δεν υπάρχει τίποτα προς προβολή, διότι δεν έχει εκτελεστεί ακόμα κάποιο πρόγραμμα.




*Εικόνα 6: Στιγμιότυπο της καρτέλας Files (κάτω τμήμα).*




*Εικόνα 7: Στιγμιότυπο της καρτέλας Packages (κάτω τμήμα).*




*Εικόνα 8: Στιγμιότυπο της καρτέλας Help (κάτω τμήμα).*

### 1.4.3 Πακέτα της R

Όσον αφορά τα πακέτα της R, στις προηγούμενες ενότητες σημειώθηκε ότι υπάρχει μια πληθώρα αυτών, όπου το καθένα χρησιμοποιείται για διαφορετική λειτουργία. Στην ουσία, τα πακέτα αυτά είναι κάποιες επεκτάσεις οι οποίες περιέχουν μια συλλογή συναρτήσεων της γλώσσας R, εκτελέσιμο κώδικα και δείγματα δεδομένων. Αποτελούν ένα πολύ σημαντικό μέρος της, αφού μπορούν να βελτιώσουν τις λειτουργίες της, είτε προσθέτοντας καινούργιες, είτε ενισχύοντας τις προ υπάρχουσες. Με αυτόν τον τρόπο, η γλώσσα γίνεται πιο δυναμική. Τα πακέτα αυτά, μπορούν να εγκατασταθούν από τους χρήστες, συνήθως μέσω ενός αποθετηρίου.

Έπειτα της διαδικασίας εγκατάστασης, η αποθήκευση τους πραγματοποιείται σε έναν συγκεκριμένο φάκελο με όνομα «library», δηλαδή βιβλιοθήκη. Η βιβλιοθήκη αυτή αποτελείται από διάφορα πακέτα συναρτήσεων. Το πακέτο με το όνομα «base» θεωρείται, όπως υποδηλώνει και το όνομα του, ο «πυρήνας» της R και συμπεριλαμβάνει τις κυριότερες συναρτήσεις αυτής της γλώσσας προγραμματισμού:

1. Ανάγνωση και χειρισμός των δεδομένων,
2. Δημιουργία ορισμένου τύπου γραφικών και
3. Πραγματοποίηση τριών βασικών στατιστικών αναλύσεων.

Για να εγκατασταθεί οποιοδήποτε πακέτο στο λογισμικό RStudio, εκτελείται η εντολή:

- install.packages(“package”),

ενώ αν ο χρήστης επιθυμεί να φορτώσει ένα πακέτο από τον φάκελο αποθήκευσης του, χρησιμοποιείται η εντολή:

- library(package).

Με το όνομα «package», υποδηλώνεται το όνομα του πακέτου που πρόκειται να εγκατασταθεί ή φορτωθεί. Για να υπάρχει μεγαλύτερη ακρίβεια, στο παράδειγμα των παρακάτω στιγμιότυπων, φαίνεται πως πραγματοποιείται η εγκατάσταση και στην συνέχεια η χρήση του πακέτου ggplot2 (βλ. εικόνα 9), το οποίο θα αναλυθεί πιο κάτω. Ακόμη, εκτελώντας εντολή search() μπορεί κανείς να δει τα πακέτα που είναι ήδη φορτωμένα στη βιβλιοθήκη (βλ. εικόνα 10).

```
> install.packages("ggplot2")
WARNING: Rtools is required to build R packages but is not currently installed. Please d
ownload and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/pc/Documents/R/win-library/4.0'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.0/ggplot2_3.3.3.zip'
Content type 'application/zip' length 4070058 bytes (3.9 MB)
downloaded 3.9 MB

package 'ggplot2' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\pc\AppData\Local\Temp\RtmpkPq6O4\downloaded_packages
> library(ggplot2)
```

*Εικόνα 9: Στιγμιότυπο εγκατάστασης και μετέπειτα χρήσης του πακέτου ggplot2.*

```
> search()
 [1] ".GlobalEnv"        "package:ggplot2"   "tools:rstudio"
 [4] "package:stats"     "package:graphics"  "package:grDevices"
 [7] "package:utils"     "package:datasets"  "package:methods"
[10] "Autoloads"         "package:base"
>
```

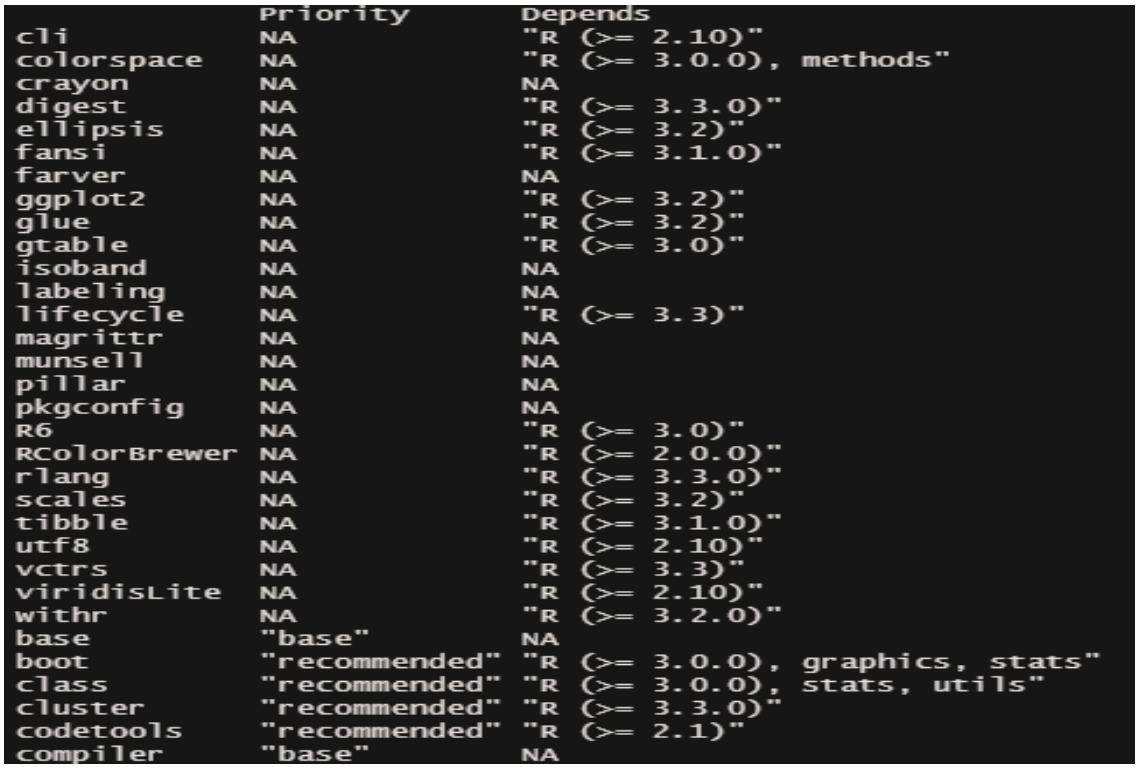
*Εικόνα 10: Στιγμιότυπο προβολής των φορτωμένων πακέτων στη βιβλιοθήκη.*

Ένα σύνολο των πακέτων, εγκαθίστανται από προεπιλογή, παράλληλα με την εγκατάσταση της R. Υπάρχει όμως και ένας μεγάλος αριθμός στατιστικών πακέτων ο οποίος περιλαμβάνεται ήδη στη βασική έκδοση της R και είναι διαθέσιμος μετά την εγκατάστασή της. Η πρόσβαση σε αυτή τη λίστα των πακέτων επιτυγχάνεται με τη βοήθεια της εντολής:

- installed.packages().

Καθώς έχει πραγματοποιηθεί για πρώτη φορά η εγκατάσταση της R στον υπολογιστή, κάποια από τα αυτομάτως εγκατεστημένα πακέτα, δεν είναι άμεσα διαθέσιμα για χρήση. Η λύση που προτείνεται στην συγκεκριμένη περίπτωση για ορισμένα από αυτά, είναι η υποχρεωτική φόρτωση τους. Ο τρόπος με τον οποίο μπορεί ένας χρήστης να διακρίνει για ποια πακέτα έχει γίνει ή όχι προεπιλογή φόρτωσης, είναι να εκτελέσει την εντολή που προ αναφέρθηκε και να παρατηρήσει την στήλη «priority», δηλαδή προτεραιότητα.

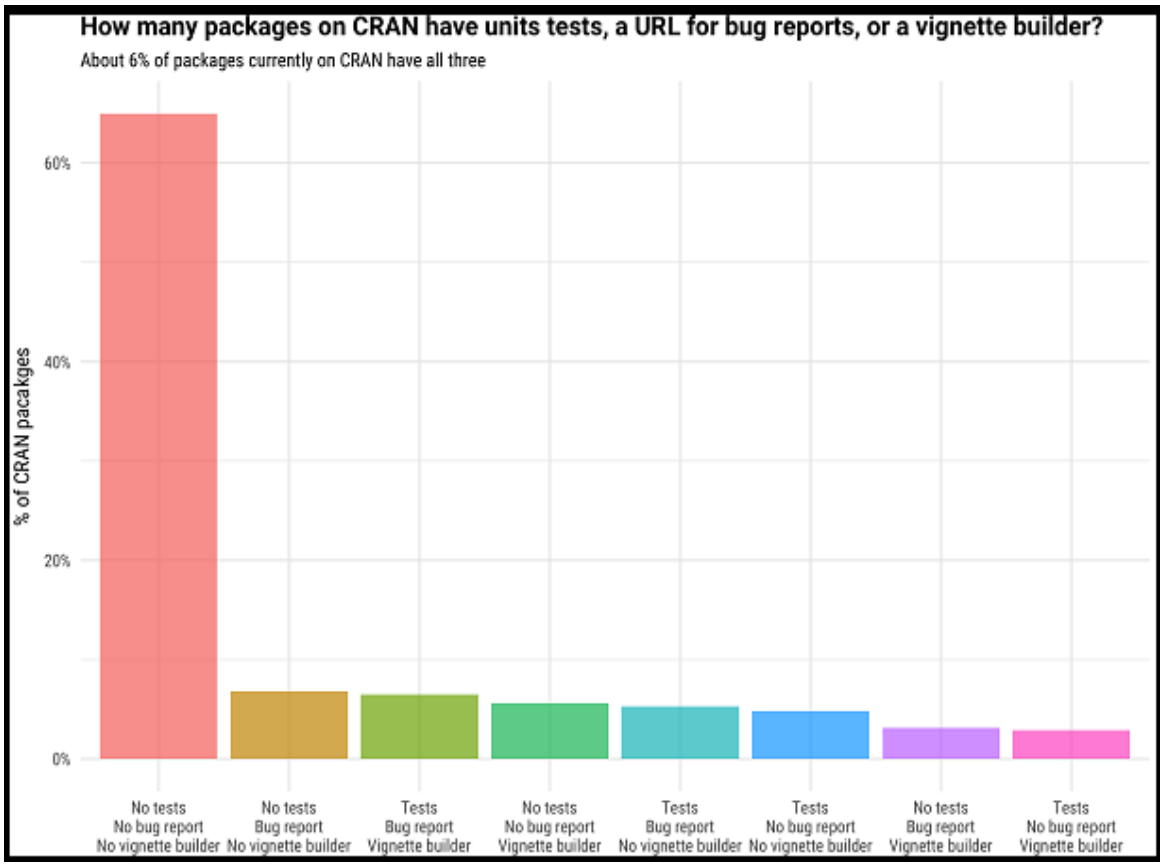
Όσα από τα πακέτα έχουν προτεραιότητα με όνομα «NA» δηλαδή «Not Available» ή «recommended» είναι αναγκαία η φόρτωση στην μνήμη με χρήση της συνάρτησης library(). Αντ' αυτού, τα πακέτα των οποίων η προτεραιότητα έχει όνομα «base» μπορούν να χρησιμοποιηθούν κατευθείαν. Στο στιγμιότυπο της παρακάτω εικόνας, φαίνεται ένα παράδειγμα της προτεραιότητας των πακέτων.



| | Priority | Depends |
|---|---|---|
| cli | NA | "R (>= 2.10)" |
| colorspace | NA | "R (>= 3.0.0), methods" |
| crayon | NA | NA |
| digest | NA | "R (>= 3.3.0)" |
| ellipsis | NA | "R (>= 3.2)" |
| fansi | NA | "R (>= 3.1.0)" |
| farver | NA | NA |
| ggplot2 | NA | "R (>= 3.2)" |
| glue | NA | "R (>= 3.2)" |
| gtable | NA | "R (>= 3.0)" |
| isoband | NA | NA |
| labeling | NA | NA |
| lifecycle | NA | "R (>= 3.3)" |
| magrittr | NA | NA |
| munsell | NA | NA |
| pillar | NA | NA |
| pkgconfig | NA | NA |
| R6 | NA | "R (>= 3.0)" |
| RColorBrewer | NA | "R (>= 2.0.0)" |
| rlang | NA | "R (>= 3.3.0)" |
| scales | NA | "R (>= 3.2)" |
| tibble | NA | "R (>= 3.1.0)" |
| utf8 | NA | "R (>= 2.10)" |
| vctrs | NA | "R (>= 3.3)" |
| viridisLite | NA | "R (>= 2.10)" |
| withr | NA | "R (>= 3.2.0)" |
| base | "base" | NA |
| boot | "recommended" | "R (>= 3.0.0), graphics, stats" |
| class | "recommended" | "R (>= 3.0.0), stats, utils" |
| cluster | "recommended" | "R (>= 3.3.0)" |
| codetools | "recommended" | "R (>= 2.1)" |
| compiler | "base" | NA |

*Εικόνα 11: Στιγμιότυπο προβολής προτεραιότητας πακέτων.*

Τα πακέτα που υπάρχουν ήδη στο λογισμικό και η εγκατάσταση τους έχει γίνει μέσω του αποθετηρίου CRAN, έχουν προτεραιότητα «NA», οπότε πρέπει να γίνει πρώτα η φόρτωση των πακέτων, προκειμένου να χρησιμοποιηθούν οι αντίστοιχες συναρτήσεις τους. Στην εικόνα 12 φαίνεται πως μόνο το 6% των αρχειοθετημένων R πακέτων στο αποθετήριο CRAN, περιλαμβάνουν τρεις από τις αποτελεσματικότερες λειτουργίες της γλώσσας, το μεγαλύτερο ποσοστό (πάνω από 60%) δεν έχει καμία ενώ μεταξύ 0 και περίπου 10% έχουν ορισμένες από αυτές.




*Εικόνα 12: Ποσοστό R πακέτων στο CRAN με συμπεριλαμβόμενες τις αποτελεσματικότερες λειτουργίες της γλώσσας*

Εν κατακλείδι, μερικά παραδείγματα από τα πιο σημαντικά πακέτα της R, τα οποία χρησιμοποιούνται κυρίως στον τομέα της ανάλυσης δεδομένων παρουσιάζονται παρακάτω:

1. ggplot2

   Από τα πιο γνωστά πακέτα, κατάλληλο για δημιουργία και σχεδίαση γραφικών. Μπορεί να κατασκευάσει γραφήματα με μία, δύο και τρεις μεταβλητές σε συνδυασμό με αριθμητικά δεδομένα και η ομαδοποίηση αυτών μπορεί να πραγματοποιηθεί βάση κάποιου συμβόλου, μεγέθους, ή και χρώματος. Σε περίπτωση πιο δια δραστικών ή τρισδιάστατων γραφημάτων, μπορούν να χρησιμοποιηθούν συνδυαστικά με το παραπάνω, τα πακέτα plotly (το οποίο μπορεί να χρησιμοποιηθεί και μαζί με την γλώσσα JavaScript, παίρνοντας την μορφή plotly.js) και plot3D αντίστοιχα.

2. dplyr

   Ιδανικό για να διαχειρίζεται πλαίσια δεδομένων (data frames). Το συγκεκριμένο πακέτο, περιλαμβάνει μια τεράστια ομάδα συναρτήσεων οι οποίες έχουν την μορφή ρημάτων και βοηθούν σε θέματα διαχείρισης των δεδομένων αυτών. Οι 5 κυριότερες από αυτές είναι:

   - filter(),
   - select(),
   - arrange(),
   - mutate() και summarize()

3. tidyr

   Το πακέτο αυτό, περιλαμβάνει διάφορα εργαλεία τα οποία βοηθούν ώστε τα δεδομένα να είναι πιο ομαδοποιημένα. Επίσης, μπορεί να χρησιμοποιηθεί άριστα, σε συνδυασμό με το πακέτο dplyr.

4. Shiny

   Χρησιμοποιείται για δημιουργία και ανάπτυξη δια δραστικών εφαρμογών, εγκατεστημένων σε τοποθεσίες στο διαδίκτυο, χωρίς να είναι απαραίτητη η χρήση της γλώσσας JavaScript. Παράλληλα με τις εφαρμογές αυτές, είναι κατάλληλο και για δημιουργία ορισμένων ταμπλό (dashboard). Για πιο εκτεταμένες δυνατότητες, μπορεί να αλληλεπιδρά με γλώσσες προγραμματισμού/τεχνολογίες όπως:

   - Θέματα Μορφοποίησης της CSS,
   - Συγκεκριμένες Ενέργειες της JavaScript αλλά και με
   - Htmlwidgets τα οποία παρέχουν ειδικά framework για την R.

5. mlr

   Πακέτο κατάλληλο αλλά και ένα από τα πιο επεκτάσιμα frameworks για απλές ή πιο πολύπλοκες διεργασίες μηχανικής μάθησης (machine learning), όπως είναι η κατηγοριοποίηση (classification), η παλινδρόμηση (regression) και η ομαδοποίηση (clustering).

6. data.table

   Χρησιμοποιείται από πολλούς τομείς, που ως επί το πλείστων, περιλαμβάνουν, επιχειρήσεις προγνωστικών δεδομένων αλλά και ιατρικούς οργανισμούς. Θεωρείται το πιο γρήγορο από όλα τα πακέτα, αφού μπορεί να διαχειρίζεται ταυτόχρονα μεγάλο όγκο δεδομένων.

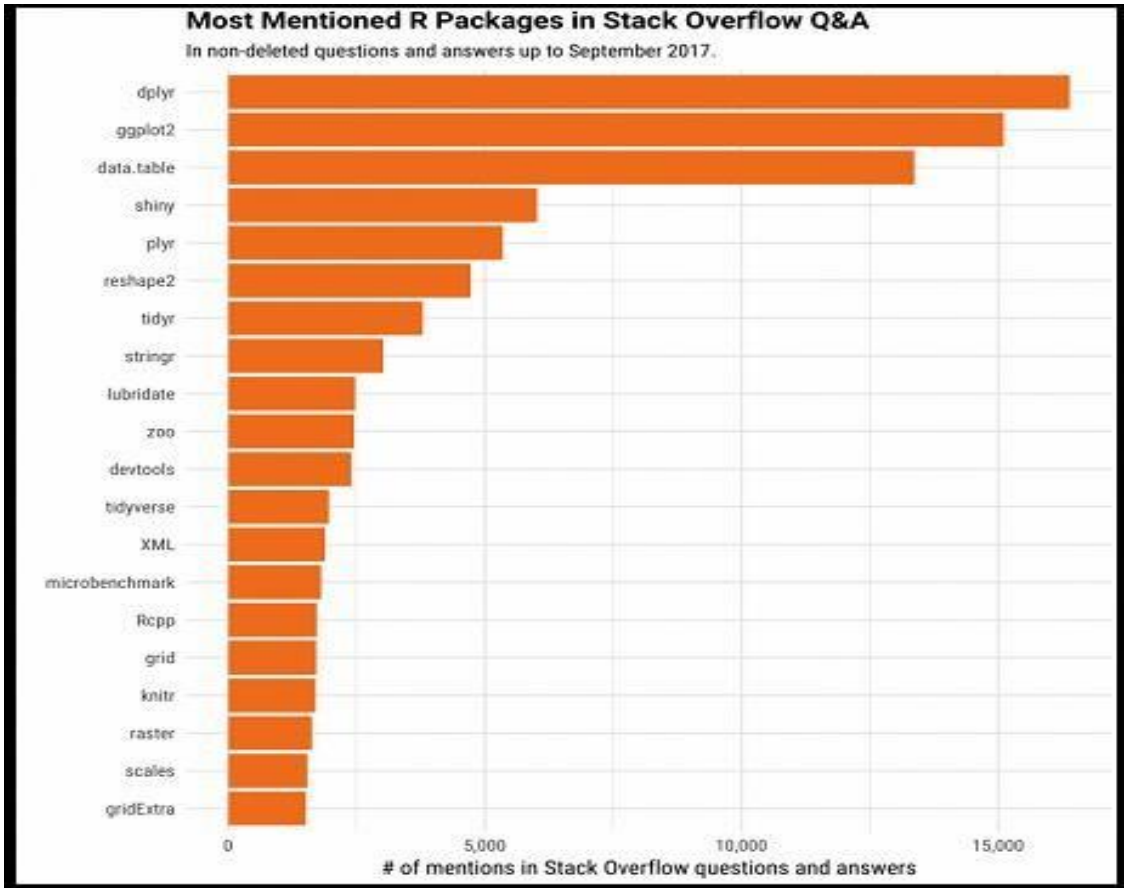
7. stringr

   Διαθέτει μια ομάδα συναρτήσεων ώστε να γίνεται όσο το δυνατόν ευκολότερη η διαχείριση σε αλφαριθμητικά δεδομένα.

8. knitr

   Πακέτο το οποίο χρησιμοποιείται κατά κύριο λόγο για ερευνητικούς σκοπούς. Από τα βασικότερα πλεονεκτήματα του αποτελεί το γεγονός πως χρησιμοποιείται για την δημιουργία αναφορών, ενώ παράλληλα, ενσωματώνεται με διάφορους τύπους δομών κώδικα όπως για παράδειγμα LaTeX, HTML, Markdown, LyX κλπ.

Τέλος, στην εικόνα 13, διακρίνονται τα πιο διαδεδομένα πακέτα της γλώσσας προγραμματισμού R, σύμφωνα με ένα ερωτηματολόγιο του Stack Overflow, με τις απαντήσεις των χρηστών μέχρι και τον Σεπτέμβριο του 2017. Σε όλη την έκταση αλλά κυρίως στην κορυφή της λίστας, βρίσκονται αρκετά από τα πακέτα που σημειώθηκαν παραπάνω και πιο συγκεκριμένα, το πακέτο dplyr με τις αναφορές να ξεπερνούν τις 15.000, το ggplot2 με 15.000 και το data.table με παραπάνω από 10.000 αναφορές.



*Εικόνα 13: Δημοφιλέστερα πακέτα της R βάση ερωτηματολόγιου του Stack Overflow.*

# 2. Αντικείμενα Δεδομένων της R

## 2.1 Τύποι Δεδομένων

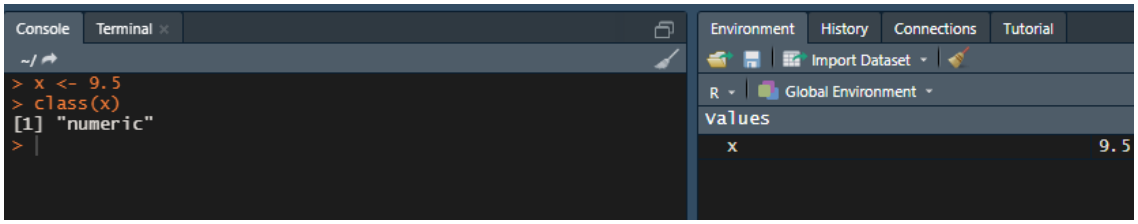
Ένα από τα πιο καίρια σημεία του προγραμματισμού, είναι οι μεταβλητές. Οι μεταβλητές είναι δεσμευμένες θέσεις μνήμης κατάλληλες για την αποθήκευση τιμών. Αυτό σημαίνει ότι, όταν δημιουργείτε μια μεταβλητή, δεσμεύεται χώρος στη διαθέσιμη μνήμη του υπολογιστή. Για την παρατήρηση των δεδομένων που περιέχονται μέσα στις μεταβλητές αυτές, χρησιμοποιούνται κάποιοι τύποι, οι οποίοι είναι γνωστοί ως τύποι δεδομένων (Data Types). Οι προαναφερόμενοι, διαφοροποιούνται ανάλογα με την γλώσσα προγραμματισμού, αφού η κάθε μια, διαχειρίζεται τους δικούς της. Όσον αφορά την R, αλλά και άλλες γλώσσες όπως η Python και η PHP, οι μεταβλητές δεν δηλώνονται με έναν τύπο δεδομένων, αλλά παίρνουν αυτόματα τον τύπο δεδομένων του αντικειμένου R που τους έχει ανατεθεί. Υπάρχουν πέντε κύριοι τύποι δεδομένων, οι οποίοι αναλύονται παρακάτω και σε αυτούς περιλαμβάνονται:

- Αριθμητικοί (Numeric),
- Ακέραιοι (Integers),
- Μιγαδικοί (Complex),
- Χαρακτήρες (Characters) και
- Λογικοί (Logical).

### 2.1.1 Αριθμητικοί (Numeric)

Οι αριθμητικοί τύποι είναι οι πιο συνήθης στην R και λειτουργούν ως προεπιλεγμένοι. Όπως υποδηλώνει και το όνομα τους, με τον όρο «αριθμητικοί» εννοείται οτιδήποτε μπορεί να θεωρηθεί αριθμητική τιμή. Χρησιμοποιούν ένα μεγάλο εύρος αριθμών και αριθμητικών τάξεων στους οποίους περιλαμβάνονται δεκαδικοί (float), δεκαδικοί διπλής ακρίβειας (double) αλλά και ακέραιοι (integers). Για τους περισσότερους στατιστικούς υπολογισμούς, προτιμάται η χρήση των δεκαδικών αριθμών. Διαφορετικά, θα πρέπει οι δεκαδικοί να μετατραπούν σε ακέραιοι.

Το γεγονός αυτό, οφείλεται στο ότι οι υπολογισμοί αυτοί, έχουν να κάνουν με αριθμούς με έως και δύο δεκαδικά ψηφία, ενώ ταυτόχρονα η αποθήκευση είναι ευκολότερη με δεκαδικούς αριθμούς από την μετατροπή των ακέραιων αριθμών σε αριθμητικούς όποτε είναι αυτοί αναγκαίοι για υπολογισμούς. Στην εικόνα 14, διακρίνεται η ανάθεση (η οποία υποδηλώνεται με το βέλος το οποίο δείχνει προς την μεταβλητή) μιας αριθμητικής τιμής και πιο ειδικά ενός δεκαδικού αριθμού σε μια μεταβλητή. Επιπρόσθετα, με την εκτέλεση της συνάρτησης class() θα τυπωθεί ο τύπος της μεταβλητής που ανατέθηκε και πλέον είναι ευδιάκριτη και στο παράθυρο environment.




*Εικόνα 14: Στιγμιότυπο ανάθεσης αριθμητικής τιμής σε μεταβλητή.*

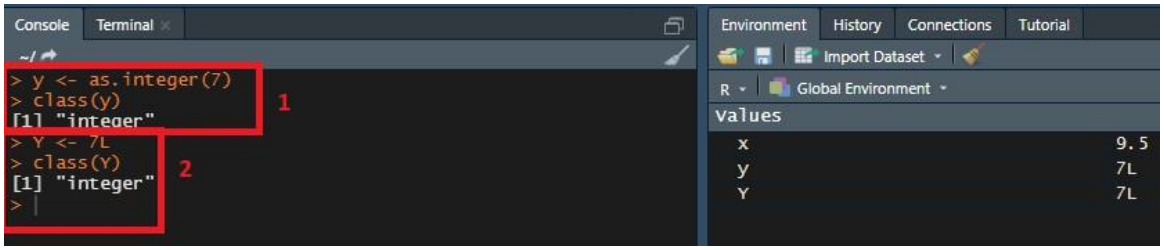
### 2.1.2 Ακέραιοι (Integers)

Ένας ακέραιος είναι ένας από τους αριθμητικούς τύπους δεδομένων που δεν αποτελείται από δεκαδικά ψηφία. Ο συγκεκριμένος τύπος χρησιμοποιείται κυρίως σε περιπτώσεις που η μεταβλητή (όπως αναφέρθηκε παραπάνω), δεν χρειάζεται να μετατραπεί. Επιπλέον, είναι χρήσιμος αν ο προγραμματιστής επιθυμεί να περάσει κάποιο αντικείμενο R σε μια συνάρτηση C ή FORTRAN που αναμένει μια ακέραια τιμή. Οι ακέραιες μεταβλητές μπορούν να οριστούν με δύο τρόπους στην γλώσσα προγραμματισμού R:

1. Καλώντας την συνάρτηση as.integer(), ή εναλλακτικά,
2. Χρησιμοποιώντας το κεφαλαίο γράμμα "L", μαζί με τον αριθμό.

Ο πρώτος τρόπος είναι προτιμότερος από τον δεύτερο. Ο λόγος που συμβαίνει αυτό, είναι διότι ότι ο τελευταίος αποτελεί συνδυασμό ενός αριθμού με έναν χαρακτήρα και είναι πιθανό να δημιουργηθούν ασάφειες σε κάποιον ο οποίος διαβάζει τον κώδικα.

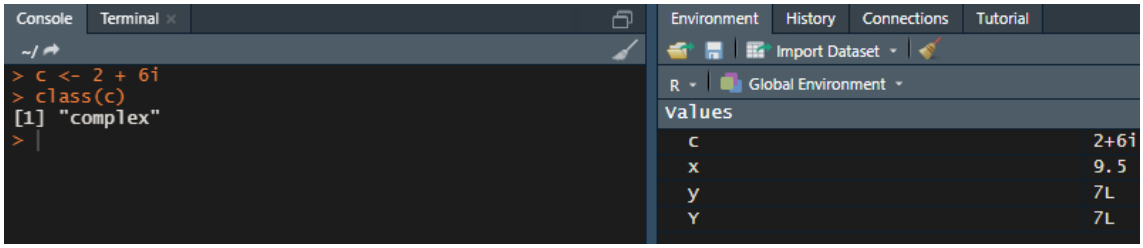
Στην εικόνα 15, παρουσιάζονται οι δύο αυτοί τρόποι ανάθεσης με τους αριθμούς 1 και 2 αντίστοιχα.




*Εικόνα 15: Στιγμιότυπο ανάθεσης ακέραιας τιμής σε μεταβλητή.*

### 2.1.3 Μιγαδικοί (Complex)

Ο συγκεκριμένος τύπος δεδομένων χρησιμοποιείται για μιγαδικούς αριθμούς ή για αριθμούς με «φανταστική» τιμή. Η τιμή αυτή, υποδηλώνεται με το γράμμα "i" και αποκαλείται «φανταστική», μόνο όταν συνοδεύει έναν αριθμό. Εναλλακτικά, μπορεί να είναι μια οποιαδήποτε μεταβλητή. Οι μιγαδικοί αριθμοί έχουν την μορφή "2+6i" με τους αριθμούς 2 και 6 να αποκαλούνται πραγματικοί. Ένα στιγμιότυπο ανάθεσης μιγαδικής τιμής σε μεταβλητή παρουσιάζεται πιο κάτω στην εικόνα 16.




*Εικόνα 16: Στιγμιότυπο ανάθεσης μιγαδικής τιμής σε μεταβλητή.*

### 2.1.4 Χαρακτήρες (Characters)

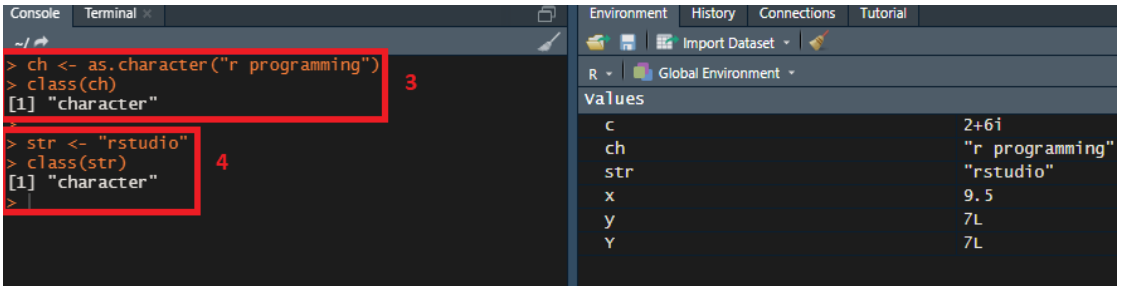
Οι χαρακτήρες, είναι ο δεύτερος πιο χρησιμοποιούμενος τύπος δεδομένων σε οποιαδήποτε γλώσσα προγραμματισμού. Σκοπός των τύπων αυτών, είναι η αποθήκευση σε δεδομένα χαρακτήρων ή συμβολοσειρών (string). Στην R, οι μεταβλητές χαρακτήρων μπορούν να δημιουργηθούν με δύο διαφορετικές μεθόδους:

- Είτε καλώντας την συνάρτηση as.character(),

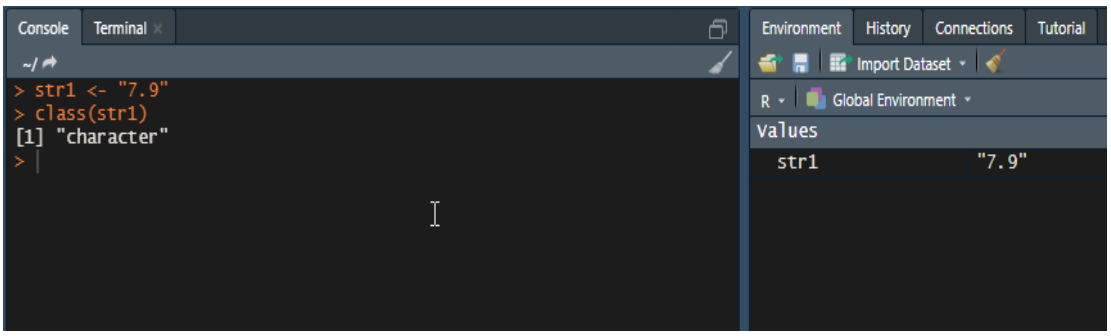
- Είτε χρησιμοποιώντας τες μέσα σε διπλά ή μονά εισαγωγικά.

Επίσης, μια ακόμη χρήση της συνάρτησης as.character() είναι ότι έχει την δυνατότητα να μετατρέψει οποιοδήποτε άλλο τύπο δεδομένων σε δεδομένα χαρακτήρων. Από την άλλη πλευρά, το βασικότερο πλεονέκτημα της δεύτερης μεθόδου, είναι ότι μπορεί να προσθέσει σχεδόν οτιδήποτε κάτω από αυτά τα διπλά ή μεμονωμένα εισαγωγικά και να εμφανιστεί ως χαρακτήρας. Αν για παράδειγμα, προστεθούν αριθμητικές τιμές κάτω από αυτά, θα θεωρηθούν ως χαρακτήρες και όχι πια ως αριθμοί.

Στην εικόνα 17, φαίνονται οι τρόποι ανάθεσης αυτών των μεταβλητών με τους αριθμούς 3 και 4 αντίστοιχα. Επιπλέον, στο στιγμιότυπο της εικόνας 18, φαίνεται η ανάθεση ενός αριθμού ως χαρακτήρα.




*Εικόνα 17: Στιγμιότυπο ανάθεσης αλφαριθμητικής τιμής σε μεταβλητή*

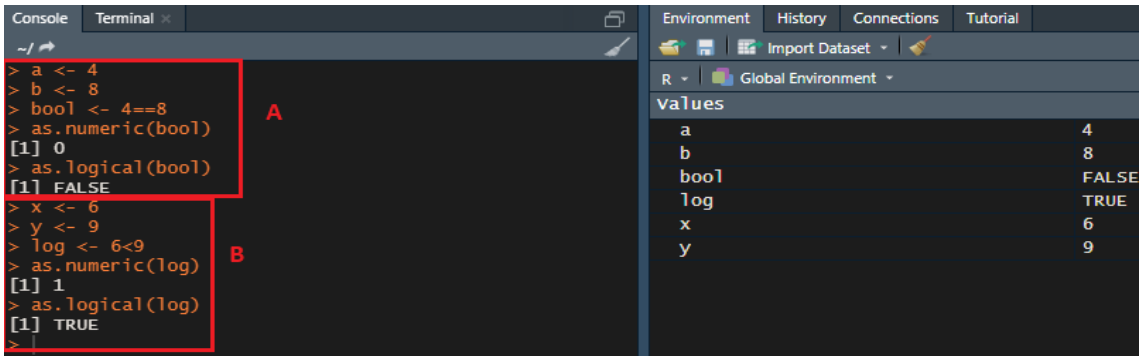



*Εικόνα 18: Στιγμιότυπο ανάθεσης αριθμού ως χαρακτήρας.*

### 2.1.5 Λογικοί (Logical)

Σε αυτήν την κατηγορία τύπων, οι λογικές μεταβλητές μπορούν να πάρουν μια από τις δύο πιθανές τιμές, συνήθως αναφερόμενες ως TRUE ή FALSE. Οι τιμές αυτές είναι γνωστές στην πληροφορική ως τελεστές αληθείας ή τελεστές boolean και εφαρμόζονται σε περιπτώσεις σύγκρισης μεταξύ δύο ή περισσότερων μεταβλητών. Για παράδειγμα, εάν δύο αριθμοί είναι ίσοι ή όχι, εάν ο πρώτος αριθμός είναι μεγαλύτερος ή μικρότερος από τον δεύτερο και ούτω καθεξής.

Οι λογικές μεταβλητές μπορούν να μετατραπούν σε αριθμητικές, με την βοήθεια της συνάρτησης as.numeric(), όπου ο τελεστής FALSE παίρνει την τιμή μηδέν (0), ενώ, ο τελεστής TRUE την τιμή ένα (1). Η αντίστροφη διαδικασία επιτυγχάνεται με την συνάρτηση as.logical(). Παρακάτω στην εικόνα 19, παρουσιάζονται δύο παραδείγματα ανάθεσης και έπειτα σύγκρισης αριθμών, σε συνδυασμό με τις μετατροπές (A και B αντίστοιχα) που αναφέρθηκαν.




*Εικόνα 19: Στιγμιότυπο ανάθεσης λογικής τιμής σε μεταβλητή.*

## 2.2 Δομές της R

Στην γλώσσα προγραμματισμού R, υπάρχουν διάφορα αντικείμενα, γνωστά ως δομές δεδομένων ή αλλιώς data structures, τα οποία παρέχουν κάποιες μεθόδους. Κύριος σκοπός των μεθόδων αυτών, είναι τα δεδομένα, έπειτα από κατάλληλη διαρρύθμιση, να έχουν μια πιο ταξινομημένη μορφή. Πρακτικά, είναι σαν να λέμε ότι έχουμε μια «συλλογή» από δεδομένα που πρέπει να τους δοθεί μια πιο επιθυμητή και σωστή μορφοποίηση. Η ταξινόμηση πραγματοποιείται ανάλογα με τις διαστάσεις που έχουν οι συγκεκριμένες δομές, δηλαδή:

- ✓ 1d για μονοδιάστατη,
- ✓ 2d για δισδιάστατη και
- ✓ nd για πολλαπλή διάσταση.

Στην R, δεν υπάρχει μηδενική διάσταση (0d). Επίσης, μπορούν να καταταχθούν σε δύο κατηγορίες, ανάλογα με την ομογένεια ή ετερογένεια τους. Στην ομογενής κατηγοριοποίηση, οι δομές έχουν την δυνατότητα αποθήκευσης μόνο ενός συγκεκριμένου τύπου δεδομένων, ο οποίος μπορεί να είναι, είτε ακέραιος, είτε χαρακτήρας κλπ. Αντίθετα, στην περίπτωση της ετερογενής κατηγοριοποίησης, οι δομές μπορούν να αποθηκεύσουν την ίδια στιγμή, παραπάνω από ένα τύπο δεδομένων.

Επιπρόσθετα, στην R δεν υπάρχει ούτε κλιμακωτός τύπος δεδομένων, ενώ, οι μεταβλητές που εμπεριέχουν μοναδικές τιμές, αποτελούν μοναδιαία διανύσματα (έχουν δηλαδή μήκος το οποίο ισούται με 1). Υπάρχουν έξι βασικοί τύποι δομών δεδομένων στην συγκεκριμένη γλώσσα προγραμματισμού, στους οποίους συμπεριλαμβάνονται:

- ❖ Διανύσματα (Vectors),
- ❖ Λίστες (Lists),
- ❖ Πίνακες (Matrices),
- ❖ Πλαίσια Δεδομένων (Data Frames) αλλά και
- ❖ Πίνακες Πολλαπλών Διαστάσεων (Arrays)

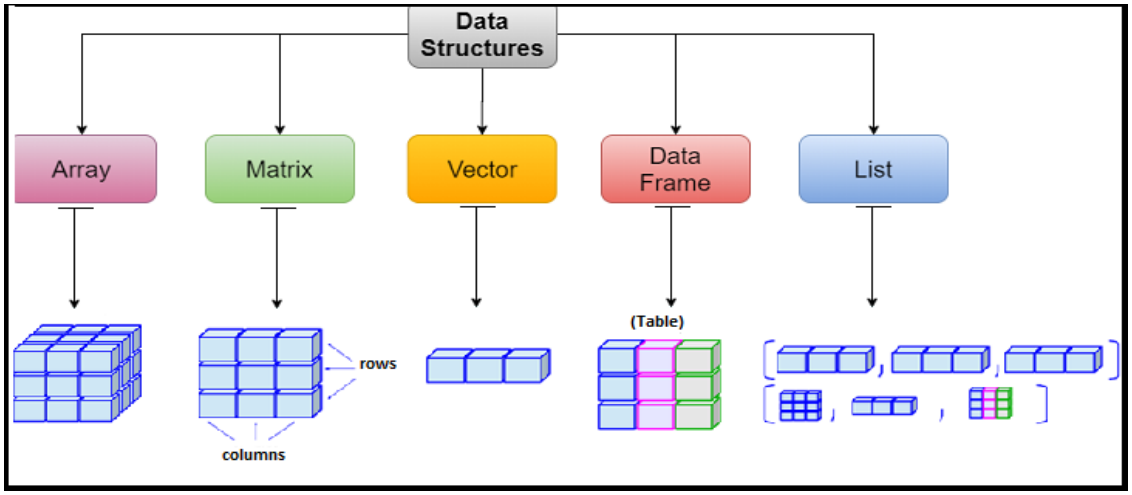
όπου αναλύονται όλοι παρακάτω.

Πριν όμως από αυτό το βήμα, στον παρακάτω πίνακα ταξινομούνται όλες αυτές οι δομές ανάλογα με τις ιδιότητες που έχουν και την κατηγορία στην οποία κατατάσσονται βάση των πληροφοριών που περιγράφηκαν παραπάνω.

*Πίνακας 1: Ταξινόμηση των δομών της R ανάλογα με τις ιδιότητες και την κατηγορία τους.*

| Ιδιότητες / Δομές | Διανύσματα (Vector) | Λίστες (Lists) | Πλαίσια Δεδομένων (Data Frames) | Πίνακες Πολλαπλών Διαστάσεων (Arrays) | Πίνακες (Matrices) |
|---|---|---|---|---|---|
| 0d | ✗ | ✗ | ✗ | ✗ | ✗ |
| 1d | ✓ | ✓ | ✗ | ✗ | ✗ |
| 2d | ✗ | ✗ | ✓ | ✗ | ✓ |
| nd | ✗ | ✗ | ✗ | ✓ | ✗ |
| Ομογενής | ✓ | ✗ | ✗ | ✓ | ✓ |
| Ετερογενής | ✗ | ✓ | ✓ | ✗ | ✗ |

Τέλος, στην εικόνα 20, απεικονίζεται ξεκάθαρα η δομή τους.




*Εικόνα 20: Απεικόνιση της δομής των δεδομένων.*

### 2.2.1 Διανύσματα (Vectors)

Τα διανύσματα είναι η απλούστερη μορφή δομών δεδομένων στην γλώσσα προγραμματισμού R. Διάνυσμα ονομάζεται μια ακολουθία από στοιχεία τα οποία έχουν τον ίδιο τύπο δεδομένων. Τα στοιχεία αυτά αποτελούν τα περιεχόμενα του διανύσματος. Τα διανύσματα της R είναι αντίστοιχης φιλοσοφίας με τους πίνακες (arrays) της γλώσσας προγραμματισμού C, με την διαφορά ότι στην R η τιμή του διανύσματος ξεκινάει από το 1 και όχι από το 0. Χωρίζονται σε δύο κύριες κατηγορίες:

- Ατομικά Διανύσματα (Atomic Vectors) και
- Λίστες (Lists)

καθώς επίσης έχουν τρεις κοινές ιδιότητες:

- τον τύπος της συνάρτησης,
- το μήκος της, πόσα στοιχεία δηλαδή εμπεριέχονται σε αυτήν και
- τα γνωρίσματα ή αλλιώς τα χαρακτηριστικά της.

Πέρα όμως από τα κοινά χαρακτηριστικά τους, μοιράζονται και μια διαφορά όσον αφορά τον τύπο των στοιχείων τους. Τα στοιχεία του ατομικού διανύσματος πρέπει να έχουν οπωσδήποτε τον ίδιο τύπο δεδομένων σε αντίθεση με τα στοιχεία της λίστας που επιτρέπεται να έχουν διαφορετικό. Οι τύποι αυτοί (όπως αναφέρθηκε στην ενότητα 2.2) μπορεί να είναι:

1. Αριθμητικοί (Numeric)
2. Ακέραιοι (Integers)
3. Μιγαδικοί (Complex)
4. Χαρακτήρες (Characters) και
5. Λογικοί (Logical).

Στα ατομικά διανύσματα, χρησιμοποιούνται όλοι οι παραπάνω τύποι, σπανίως όμως οι μιγαδικοί. Στην R, οι πιο περίπλοκες δομές δεδομένων κατασκευάζονται με διανύσματα ως δομικά στοιχεία. Τα διανύσματα μπορούν να δημιουργηθούν με αρκετούς διαφορετικούς τρόπους στην συγκεκριμένη γλώσσα προγραμματισμού, όπως επίσης να διαγραφούν, να τροποποιηθούν, να ταξινομηθούν κλπ. Αναλυτικότερα, υπαγορεύονται όλες αυτές οι επιλογές παρακάτω.

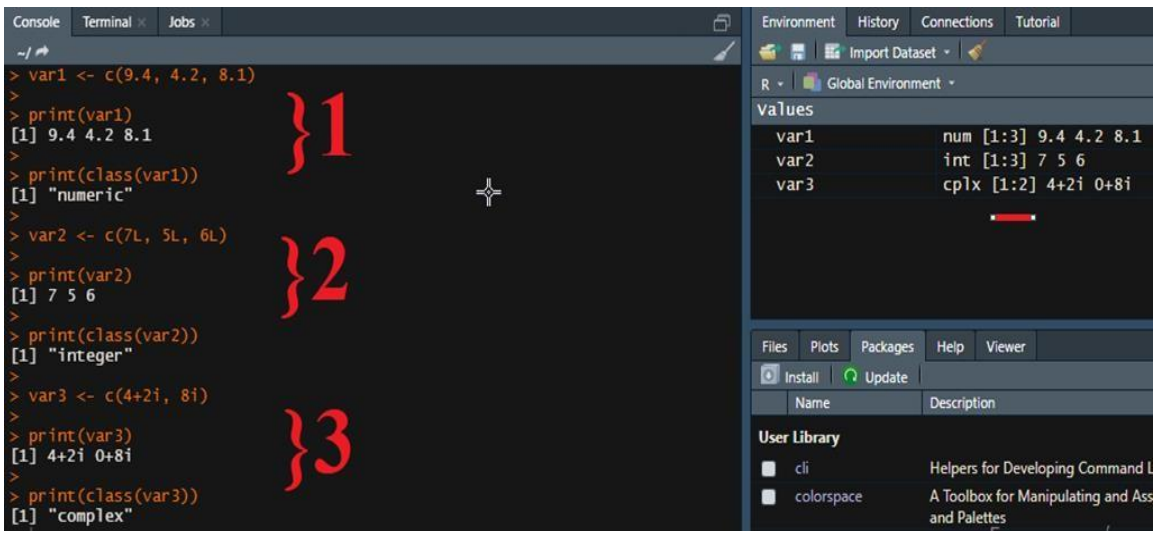
#### 2.2.1.1 ΔημΙΟυργία ΔΙανυσμάτων

Τα ατομικά διανύσματα στην γλώσσα προγραμματισμού R δημιουργούνται με τέσσερις διαφορετικούς τρόπους:

- **Χρησιμοποιώντας την συνάρτηση c()**

Η συγκεκριμένη συνάρτηση, η οποία ονομάζεται αλλιώς και συνάρτηση συνδυασμού, χρησιμοποιείται όταν ο χρήστης θέλει να αποθηκεύσει πολλαπλές τιμές κάτω από μια μεταβλητή. Για να δημιουργηθεί το συγκεκριμένο διάνυσμα με χρήση της συνάρτησης c(), τα στοιχεία πρέπει να διαχωρίζονται το ένα από το άλλο με κόμμα. Ακόμα και αν ο χρήστης αποθηκεύσει έναν απλό αριθμό, αυτό εξακολουθεί να θεωρείται διάνυσμα, με την διαφορά πως ονομάζεται μοναδιαίο διάνυσμα, αποτελείται δηλαδή από ένα στοιχείο. Μπορούν να χρησιμοποιηθούν ακέραιοι αριθμοί, δεκαδικοί κλπ. ενώ με το που πραγματοποιηθεί η ανάθεση των τιμών, το διάνυσμα είναι ευδιάκριτο και στο παράθυρο environment.

Με το που εκτελεστεί η εντολή: print(variable) θα τυπωθεί στην οθόνη το αντίστοιχο διάνυσμα, ενώ αν χρησιμοποιηθεί η εντολή: print(class(variable)), θα τυπωθεί και ο τύπος του συγκεκριμένου διανύσματος. Ένα παράδειγμα ανάθεσης αριθμητικής (}1), ακέραιης (}2) και μιγαδικής τιμής (}3) παρουσιάζεται παρακάτω στο στιγμιότυπο της εικόνας 21. Αντίστοιχα, ισχύει και για τις λογικές τιμές.

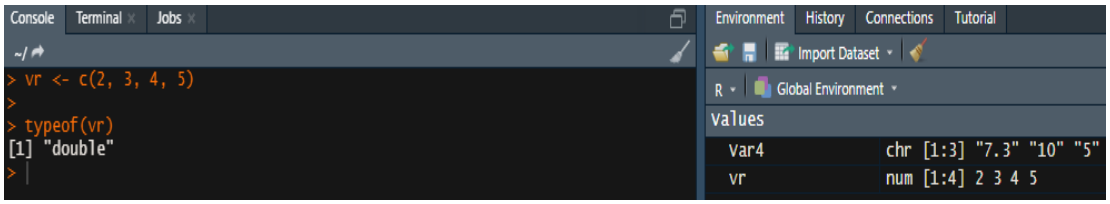



*Εικόνα 21: Στιγμιότυπο δημιουργίας αριθμητικού, ακέραιου και μιγαδικού διανύσματος με την μέθοδο c()*

Ένας εναλλακτικός τρόπος με τον οποίο μπορεί να προβληθεί ο τύπος του διανύσματος, είναι εκτελώντας την συνάρτηση:

- typeof(variable).

Η διαφορά σε σχέση με τον παραπάνω τρόπο, είναι ότι όταν υπάρχουν αριθμητικές τιμές θα επιστρέψει σαν τύπο διανύσματος "double" δηλαδή δεκαδικό διπλής ακρίβειας και όχι "numeric", παρόλο που η τάξη του διανύσματος είναι αριθμητική (βλ. εικόνα 22).




*Εικόνα 22: Στιγμιότυπο χρήσης της συνάρτησης typeof().*

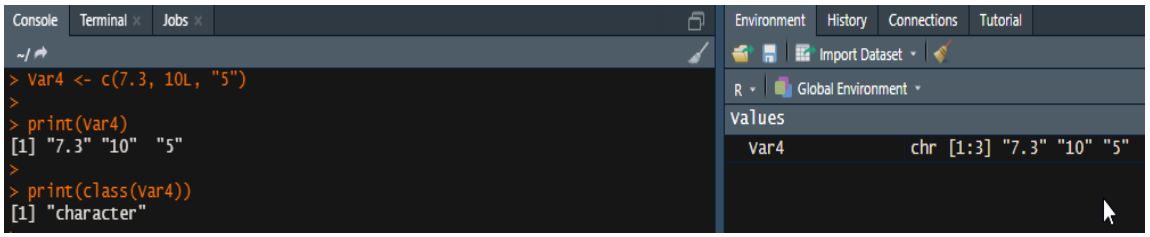
Αξίζει επίσης να σημειωθεί, ότι στο παράθυρο environment πέρα από την ανάθεση του κάθε διανύσματος, διακρίνεται και το μήκος του. Όπως φαίνεται στο παραπάνω στιγμιότυπο (21), υπογραμμισμένο με κόκκινο χρώμα, το μήκος του αριθμητικού αλλά και του ακέραιου διανύσματος είναι [1:3] δηλαδή περιέχουν το καθένα από τρία στοιχεία, ενώ το μήκος του μιγαδικού διανύσματος είναι [1:2] δηλαδή περιέχει δύο στοιχεία.

Όσον αφορά τα διανύσματα χαρακτήρων, έχουν ένα επιπλέον πλεονέκτημα σε σχέση με τα υπόλοιπα. Όπως σημειώθηκε και στην αρχή της ενότητας, τα στοιχεία που υπάρχουν μέσα σε ένα διάνυσμα έχουν κοινό τύπο δεδομένων. Το προαναφερόμενο πλεονέκτημα, γνωστό ως coercion, είναι ότι αν μέσα σε ένα διάνυσμα υπάρχει παραπάνω από ένας και διαφορετικός τύπος δεδομένων, η συνάρτηση c() θα μετατρέψει εξ ολοκλήρου το διάνυσμα και αυτό θα πάρει αυτόματα τον τύπο που έχει την μεγαλύτερη προτεραιότητα. Η προτεραιότητα αυτή, από τον χαμηλότερο προς τον υψηλότερο τύπο έχει ως εξής:

**Λογικοί < Ακέραιοι < Δεκαδικοί Διπλής Ακρίβειας < Μιγαδικοί < Χαρακτήρες.**

Για να υπάρχει μεγαλύτερη σαφήνεια, αν σε ένα διάνυσμα το οποίο αποτελείται από πολλούς και διαφορετικούς τύπους δεδομένων, προστεθεί έστω και ένας χαρακτήρας το διάνυσμα θα μετατραπεί και πλέον θα θεωρείται διάνυσμα χαρακτήρα

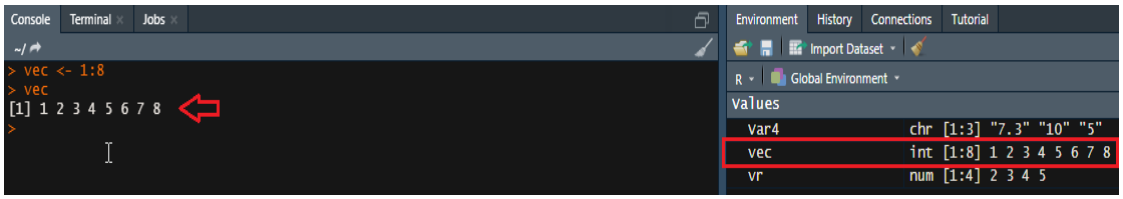
(εφόσον αυτό έχει την υψηλότερη προτεραιότητα). Στο στιγμιότυπο της εικόνας 23, παρουσιάζεται ένα τέτοιο παράδειγμα.




*Εικόνα 23: Στιγμιότυπο μετατροπής διανύσματος χαρακτήρα*

- **Χρησιμοποιώντας τον (:) τελεστή**

Ο δεύτερος και πολύ απλούστερος τρόπος με τον οποίο μπορεί να δημιουργηθεί ένα διάνυσμα στην γλώσσα προγραμματισμού R, είναι χρησιμοποιώντας τον (:) τελεστή. Το μόνο που έχει να κάνει ο χρήστης ή ο προγραμματιστής είναι να αναθέσει σε μια μεταβλητή δύο αριθμούς, διαχωρισμένους από τον συγκεκριμένο τελεστή. Με το που πραγματοποιηθεί η ανάθεση, έχει δημιουργηθεί το διάνυσμα με μήκος όσο υποδηλώνει ο τελευταίος αριθμός. Για παράδειγμα, αν χρησιμοποιηθούν οι αριθμοί 1:8, το διάνυσμα θα περιλαμβάνει μέσα οκτώ στοιχεία (1,2,3,4,5,6,7,8). Το παράδειγμα αυτό, παρουσιάζεται αναλυτικότερα παρακάτω στο στιγμιότυπο της εικόνας 24.

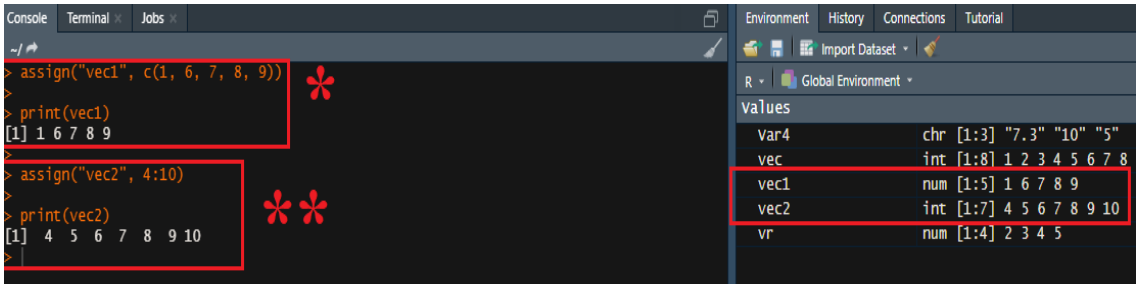



*Εικόνα 24: Στιγμιότυπο δημιουργίας διανύσματος χρησιμοποιώντας τον : τελεστή.*

- **Χρησιμοποιώντας την συνάρτηση assign()**

Ο επόμενος τρόπος να δημιουργηθεί ένα διάνυσμα στην γλώσσα προγραμματισμού R, είναι καλώντας την συνάρτηση assign(). Μέσα στην συνάρτηση assign(), είναι απαραίτητη προϋπόθεση να οριστεί το όνομα του καινούργιου διανύσματος με διπλά

εισαγωγικά αλλά και οι τιμές που πρόκειται να αποθηκευτούν μέσα σε αυτό. Οι τιμές αυτές διαχωρίζονται με κόμμα, έπειτα από το όνομα του διανύσματος και δηλώνονται είτε με χρήση της συνάρτησης c() που αναφέρθηκε παραπάνω σαν α΄ τρόπος δημιουργίας, είτε με τον δεύτερο τρόπο, δηλαδή τον (:) τελεστή. Στο στιγμιότυπο της εικόνας 25 με (*) και (**) υπάρχουν αντίστοιχα τα παραδείγματα με τις παραπάνω δύο εκδοχές.




*Εικόνα 25: Στιγμιότυπο δημιουργίας διανύσματος με χρήση της συνάρτησης assign().*

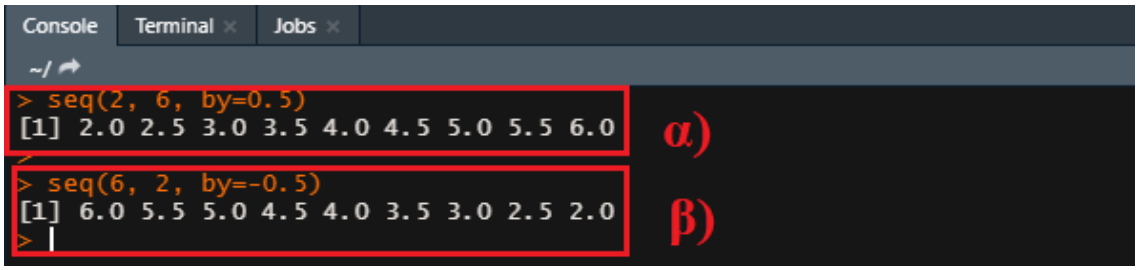
- **Χρησιμοποιώντας τις συναρτήσεις seq() και rep()**

Ο επόμενος τρόπος με τον οποίο μπορεί να δημιουργηθεί ένα R διάνυσμα, είναι χρησιμοποιώντας σε πρώτη φάση την συνάρτηση ακολουθίας seq(). Η συγκεκριμένη συνάρτηση είναι για πιο πολύπλοκες περιπτώσεις, όπως είναι η κατασκευή μιας ακολουθίας αριθμών με οποιαδήποτε διαφορά μεταξύ των τιμών. Περιλαμβάνει με την σειρά της δύο επιμέρους τρόπους εκτέλεσης:

1. ορίζοντας το βήμα του διανύσματος με χρήση της παραμέτρου by ή
2. ορίζοντας το μήκος του διανύσματος με την λειτουργία length.out.

Όσον αφορά την πρώτη επιλογή, η παράμετρος by επιτρέπει να οριστεί το βήμα του διανύσματος, δηλαδή την ποσότητα με την οποία οι αριθμοί θα αυξάνονται ή θα μειώνονται κάθε φορά. Το βήμα μπορεί να έχει είτε θετική είτε αρνητική τιμή. Για παράδειγμα, από το 2 μέχρι το 6 με βήμα 0.5, το διάνυσμα που θα δημιουργηθεί θα είναι: 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6 (βλ. εικόνα 26 α). Αντίστοιχα, από το 6 μέχρι το 2 με βήμα -0.5 το διάνυσμα θα είναι: 6, 5.5, 5, 4.5, 4, 3.5, 3, 2.5, 2 (βλ. εικόνα 26 β).

Αντιθέτως, για την δεύτερη επιλογή η λειτουργία length.out ορίζει το μήκος του διανύσματος, ενώ, παράλληλα η ίδια η γλώσσα R υπολογίζει από μόνη της ποιο θα είναι το μέγεθος του βήματος.

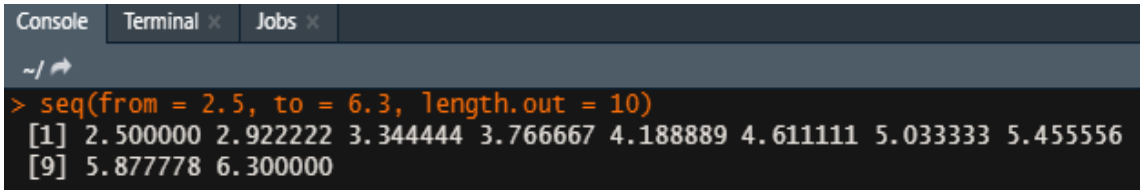



*Εικόνα 26: Στιγμιότυπο δημιουργίας διανύσματος με χρήση seq() και by.*

Το κοινό χαρακτηριστικό που έχουν οι δύο αυτοί τρόποι, είναι πως για να οριστεί ο πρώτος αλλά και ο τελευταίος αριθμός χρησιμοποιούνται τα arguments:

- from και
- to αντίστοιχα.

Η μοναδική μικρή διαφορά είναι ότι όταν χρησιμοποιείται η μέθοδος του βήματος, δεν αποτελεί απαραίτητη προϋπόθεση ο χρήστης να γράψει τα συγκεκριμένα arguments, αρκεί να έχει δηλώσει τις τιμές με την σωστή σειρά. Π.χ. το διάνυσμα: seq(2, 6, 0.5) θα δημιουργηθεί και θα λειτουργεί ακριβώς με τον ίδιο τρόπο με το διάνυσμα: seq(from=2, to=6, by=0.5). Στην αντίθετη περίπτωση, δηλαδή όταν χρησιμοποιείται η μέθοδος του μήκους, είναι αναγκαία η δήλωση των συγκεκριμένων arguments (βλ. εικόνα 27).

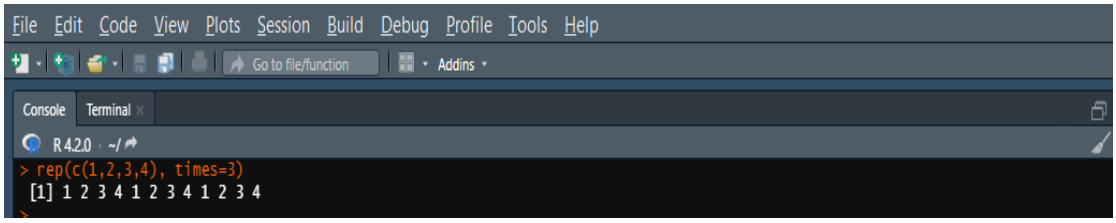



*Εικόνα 27: Στιγμιότυπο δημιουργίας διανύσματος με χρήση seq() και length.out*

Τελευταίος τρόπος αποτελεί η συνάρτηση επανάληψης rep(), η οποία βοηθάει και διευκολύνει τον χρήστη σε περιπτώσεις κατασκευής διανυσμάτων με απαραίτητη προϋπόθεση την επανάληψη ορισμένων τιμών. Αντίστοιχα με την συνάρτηση ακολουθίας seq(), η rep() συμπεριλαμβάνει και αυτή δύο τρόπους εκτέλεσης:

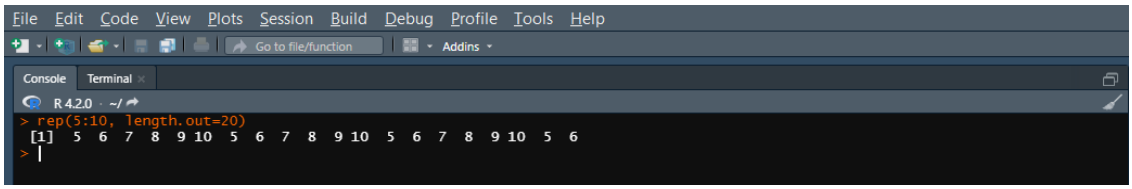
1. ορίζοντας τον αριθμό των επαναλήψεων με χρήση της παραμέτρου times ή
2. ορίζοντας το μέγεθος του διανύσματος με χρήση της παραμέτρου length.out.

Στην πρώτη περίπτωση, το όρισμα times υποδηλώνει πόσες φορές θα επαναληφθούν οι τιμές του διανύσματος που έχουν ανατεθεί. Παραδείγματος χάρη, αν ο χρήστης θέλει να δημιουργήσει ένα διάνυσμα που περιλαμβάνει τους αριθμούς 1,2,3,4 και θέλει να επαναληφθεί τρεις φορές, το όρισμα times θα ισούται με την τιμή 3. Έτσι, το τελικό αποτέλεσμα θα είναι της μορφής: 1 2 3 4 1 2 3 4 1 2 3 4. (βλ. εικόνα 28)




*Εικόνα 28: Στιγμιότυπο δημιουργίας διανύσματος με χρήση rep() και times.*

Κλείνοντας, στην δεύτερη περίπτωση το όρισμα length.out λειτουργεί παρομοίως με την περίπτωση της συνάρτησης seq(). Ορίζει το μέγεθος του επαναλαμβανόμενου διανύσματος. Αν για παράδειγμα (βλ. εικόνα 29) ο χρήστης δημιουργήσει ένα διάνυσμα με τις τιμές 5,6,7,8,9,10 και η παράμετρος μήκους length ανατεθεί ίση με 20, αυτό σημαίνει ότι οι παραπάνω τιμές θα επαναλαμβάνονται με την σειρά που είναι δηλωμένες, μέχρι στο τυπωμένο αποτέλεσμα να υπάρχουν είκοσι τιμές.

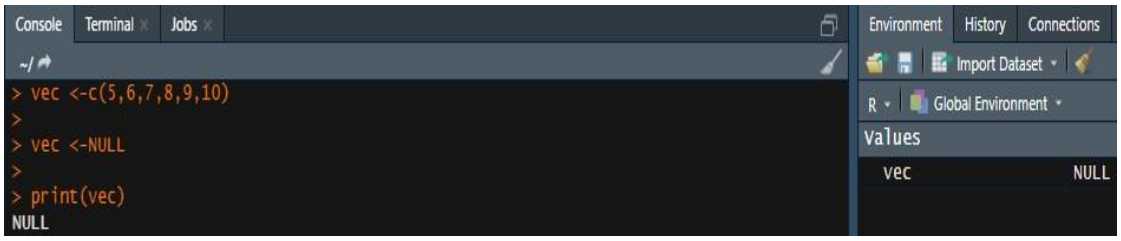



*Εικόνα 29: Δημιουργία διανύσματος με χρήση rep() και length.out*

### 2.2.1.2 Διαγραφή Διανυσμάτων

Με τον όρο διαγραφή ενός R διανύσματος, εννοείται η διαδικασία κατά την οποία αφαιρούνται όλα ή ορισμένα από τα στοιχεία τα οποία υπάρχουν μέσα σε ένα διάνυσμα. Η κάθε μια περίπτωση ακολουθεί και διαφορετική μέθοδο. Προκειμένου να επιτευχθεί είτε η μια επιλογή είτε η άλλη, πρέπει αρχικά το διάνυσμα να έχει δημιουργηθεί. Έπειτα της διαδικασίας δημιουργίας, το μόνο που μένει για γίνει αρχικά η πρώτη περίπτωση δηλαδή αυτή της ολοκληρωτικής διαγραφής του διανύσματος, είναι να του ανατεθεί η τιμή NULL δηλαδή μηδέν. Στο στιγμιότυπο της

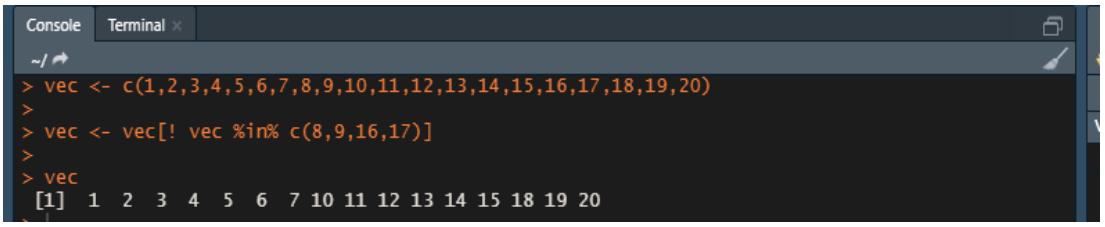
εικόνας 30 αποτυπώνεται ένα αντίστοιχο παράδειγμα. Όπως είναι ευδιάκριτο και το παράθυρο environment, το διάνυσμα θεωρείται πλέον κενό.




*Εικόνα 30: Στιγμιότυπο ολοκληρωτικής διαγραφής ενός διανύσματος.*

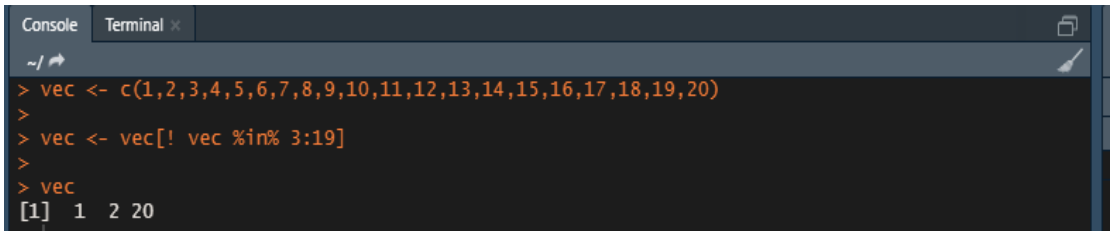
Στην δεύτερη περίπτωση, δηλαδή για να εξαλειφθούν μερικά συγκεκριμένα στοιχεία τα οποία υπάρχουν στο διάνυσμα, χρησιμοποιείται συνδυαστικά η μέθοδος %in% και άλλες ορισμένες διαδικασίες που αναφέρθηκαν και στην ενότητα 2.2.1.1. Πιο συγκεκριμένα:

1. Με την συνάρτηση δημιουργίας c(), όταν πρόκειται να δηλωθούν μεμονωμένα τα προς απαλοιφή στοιχεία. Για παράδειγμα, αν το διάνυσμα αποτελείται από τους αριθμούς 1 έως και το 20 και ο προγραμματιστής επιθυμεί να αφαιρέσει τα στοιχεία 8, 9, 16 και 17 θα πρέπει να εκτελέσει τον κώδικα της εικόνας 31.



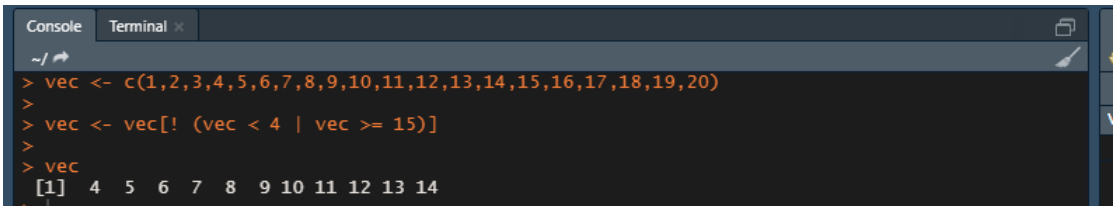

*Εικόνα 31: Στιγμιότυπο αφαίρεσης συγκεκριμένων στοιχείων από διάνυσμα με την μέθοδο c().*

2. Με τον τελεστή (:), όταν πρόκειται να διαγραφεί ένα μεγάλο εύρος στοιχείων ώστε να μην πληκτρολογούνται ένα προς ένα. Ως παράδειγμα για τον συγκεκριμένο τρόπο, αποτελεί το στιγμιότυπο της εικόνας 32, όπου εν τέλη θα αφαιρεθούν τα στοιχεία από το 3 έως και το 19.




*Εικόνα 32: Στιγμιότυπο αφαίρεσης συγκεκριμένων στοιχείων από διάνυσμα με τον : τελεστή.*

3. Με τους τελεστές σύγκρισης >, <, >= ή <=, σε περίπτωση που επρόκειτο να διαγραφεί ένα μεγαλύτερο, μικρότερο ή ως συνδυαστικό και των δύο εύρος στοιχείων. Η μοναδική διαφορά σε σχέση με τους παραπάνω δύο τρόπους, είναι ότι εδώ δεν χρησιμοποιείται η μέθοδος %in%. Λόγου χάρη, αν πρόκειται να απαλειφθούν τα στοιχεία που βρίσκονται πριν το 4ο και από το 15ο στοιχείο και μετά, εκτελείται ο κώδικας της εικόνας 33.




*Εικόνα 33: Στιγμιότυπο αφαίρεσης συγκεκριμένων στοιχείων από διάνυσμα με τους τελεστές σύγκρισης.*

### 2.2.2 Λίστες (Lists)

Μέχρι στιγμής, τα διανύσματα που αναλύθηκαν παραπάνω αποτελούν ατομικά διανύσματα, εμπεριέχουν δηλαδή δεδομένα μιας μορφής. Υπάρχουν όμως και ειδικές περιπτώσεις στις οποίες είναι απαραίτητη η δημιουργία δεδομένων πολλαπλών και διαφόρων μορφών. Λύση στην συγκεκριμένη περίπτωση, αποτελεί η χρήση λίστας. Με λίγα λόγια, λίστα ονομάζεται ένα διάνυσμα του οποίου τα στοιχεία επιτρέπεται να έχουν διαφορετικό τύπο δεδομένων. Οι τύποι αυτοί πέραν από τους ήδη γνωστούς, μπορεί να είναι πίνακες, πίνακες πολλαπλών διαστάσεων, συναρτήσεις ή ακόμα και κάποια άλλη/ες λίστα/ες.

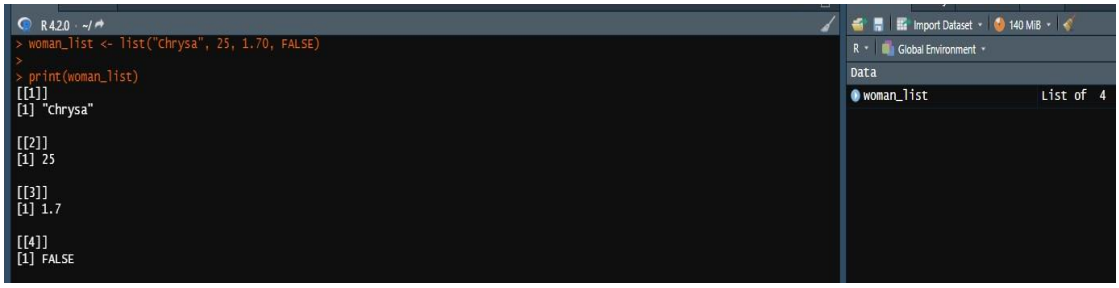
#### 2.2.2.1 ΔημΙΟυργία Λίστας

Στην υπό ενότητα 2.2.1.1. αναλύθηκε πως δημιουργείται ένα ατομικό διάνυσμα, με όλους τους πιθανούς τρόπους με τους οποίους μπορεί να καταστεί αυτό δυνατό. Ένας από τους παραπάνω αυτούς τρόπους συμπεριελάμβανε την χρήση ειδικών συναρτήσεων. Παρομοίως, η δημιουργία λίστας στην γλώσσα προγραμματισμού R επιτυγχάνεται με την βοήθεια μιας συνάρτησης η οποία έχει αντίστοιχο όνομα, δηλαδή list(). Ως ορίσματα της συγκεκριμένης συνάρτησης μπορούν να χρησιμοποιηθούν ταυτόχρονα διαφορετικοί τύποι δεδομένων. Για μεγαλύτερη σαφήνεια, ακολουθεί το παρακάτω παράδειγμα.

Έστω ότι ένας προγραμματιστής/χρήστης θέλει να δημιουργήσει μία λίστα η οποία να εμπεριέχει στοιχεία για την περιγραφή ενός ανθρώπου π.χ. μιας γυναίκας. Οι πληροφορίες που επιθυμεί να δηλώσει είναι:

- ✓ Το όνομα της, που αποτελεί ένα χαρακτήρα
- ✓ Την ηλικία της, η οποία θεωρείται μια ακέραια θετική τιμή
- ✓ Το ύψος της, ως αριθμητική τιμή δεκαδικού αριθμού και τέλος
- ✓ Το αν έχει ή όχι παιδιά, σαν λογική τιμή TRUE ή FALSE .

Η διαδικασία που έπεται να ακολουθήσει, είναι να αναθέσει μια μεταβλητή στην παραπάνω συνάρτηση και στην συνέχεια να δηλώσει όλες τις τιμές ως ορίσματα στην παρένθεση της συνάρτησης. Για να τυπωθεί το αποτέλεσμα και να εμφανιστούν όλα τα στοιχεία που δηλώθηκαν στην λίστα, αρκεί να εκτελεστεί η εντολή: print(list) όπου list το όνομα της μεταβλητής που έχει τεθεί. Επίσης, κάτι που αξίζει να σημειωθεί είναι πως πριν ο χρήστης τυπώσει το αποτέλεσμα, όταν δηλαδή πρώτο ορίζει τη λίστα, στο διπλανό παράθυρο (environment), φαίνεται πως αυτή έχει δηλωθεί στο πρόγραμμα καθώς και πόσα στοιχεία περιέχει. Στην συγκεκριμένη περίπτωση 4. Το παράδειγμα αυτό, απεικονίζεται στο στιγμιότυπο 34.



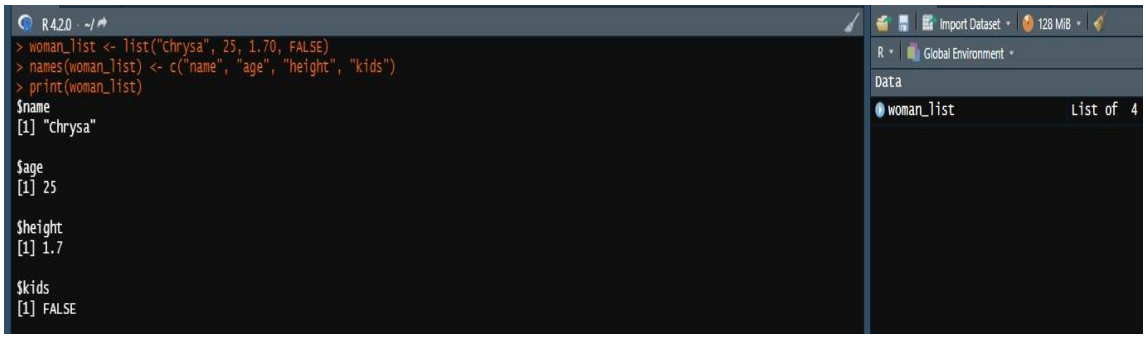

*Εικόνα 34: Στιγμιότυπο δημιουργίας λίστας με την συνάρτηση list()*

Άλλη μια διαδικασία η οποία θεωρείται αρκετά χρήσιμη, είναι ότι ο χρήστης έχει την δυνατότητα να δώσει ονόματα στα ορίσματα που έχει θέσει. Έπειτα, μπορεί να έχει πρόσβαση στα ορίσματα αυτά χρησιμοποιώντας απλά και μόνο τα ονόματα. Η όλη διαδικασία μπορεί να επιτευχθεί είτε μετά την δημιουργία της λίστας είτε κατά την διάρκεια κατασκευής της. Έστω ότι ο χρήστης επιθυμεί αρχικά να θέσει ονόματα έπειτα της δημιουργία της λίστας. Με βάση το παραπάνω παράδειγμα:

Έχοντας δημιουργηθεί η λίστα με τις αντίστοιχες πληροφορίες της, ο χρήστης θέλει να δώσει ονόματα ώστε να φαίνεται ξεκάθαρα σε τι μορφής κατηγορία ανήκει η κάθε πληροφορία. Για να μπορέσουν να τεθούν τα ονόματα γίνεται χρήση μιας συνάρτησης με όνομα names(). Ως όρισμα της συγκεκριμένης συνάρτησης, χρησιμοποιείται η μεταβλητή που έχει οριστεί για την δημιουργία της αρχικής λίστας, στην συγκεκριμένη περίπτωση woman_list. Στην συνέχεια, χρησιμοποιώντας ως καινούργια μεταβλητή την συνάρτηση names(woman_list), δημιουργείται ένα νέο διάνυσμα με χρήση της συνάρτησης c(), μέσα στην οποία δηλώνονται ως χαρακτήρες, οι κατηγορίες των ονομάτων για την καθεμία πληροφορία. Οι κατηγορίες στο παράδειγμα αυτό θα είναι:

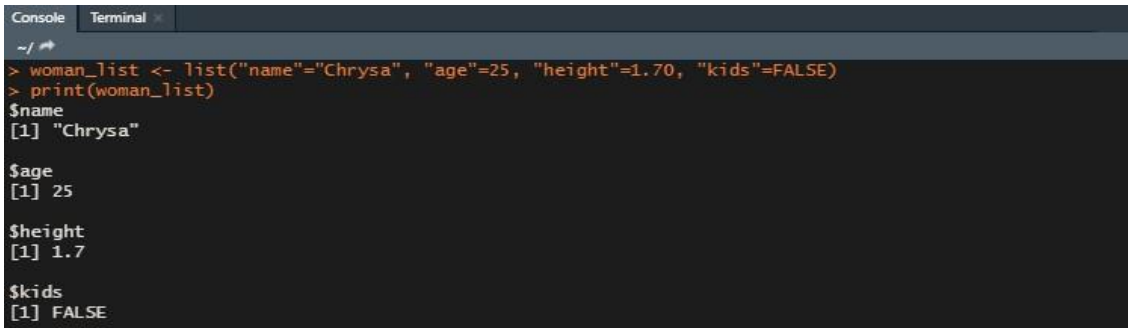
- Name
- Age
- Height &
- Kids

Τελικό βήμα, είναι να εκτελεστεί το πρόγραμμα με την εντολή: print(woman_list). Αφού εκτελεστεί ο κώδικας, θα τυπωθούν οι πληροφορίες που εμπεριέχονται στη λίστα μαζί με το όνομα της αντίστοιχης τους κατηγορίας. Το κάθε όνομα εμφανίζεται πριν από την κάθε πληροφορία, ακολουθούμενο από το σύμβολο
$. Στο στιγμιότυπο 35 απεικονίζεται το παράδειγμα.




*Εικόνα 35: Στιγμιότυπο δημιουργία λίστας δηλώνοντας ονόματα στα ορίσματα.*

Περνώντας στην δεύτερη περίπτωση, όπου ο χρήστης επιθυμεί να θέσει τα ονόματα ταυτόχρονα με την κατασκευή της λίστας. Η διαδικασία δεν διαφέρει αισθητά με την πρώτη περίπτωση αφού το μονό που έχει να κάνει είναι να θέσει τις μεταβλητές: Name, Age, Height & Kids ίσες με τα ορίσματα μέσα στην λίστα. Ωστόσο, όταν τυπωθεί το τελικό αποτέλεσμα στην οθόνη τα ονόματα θα εμφανίζονται πάλι μετά το σύμβολο $. (βλ. εικόνα 36)




*Εικόνα 36: Στιγμιότυπο δημιουργίας λίστας δηλώνοντας ονόματα στα ορίσματα κατά την δημιουργία της λίστας.*

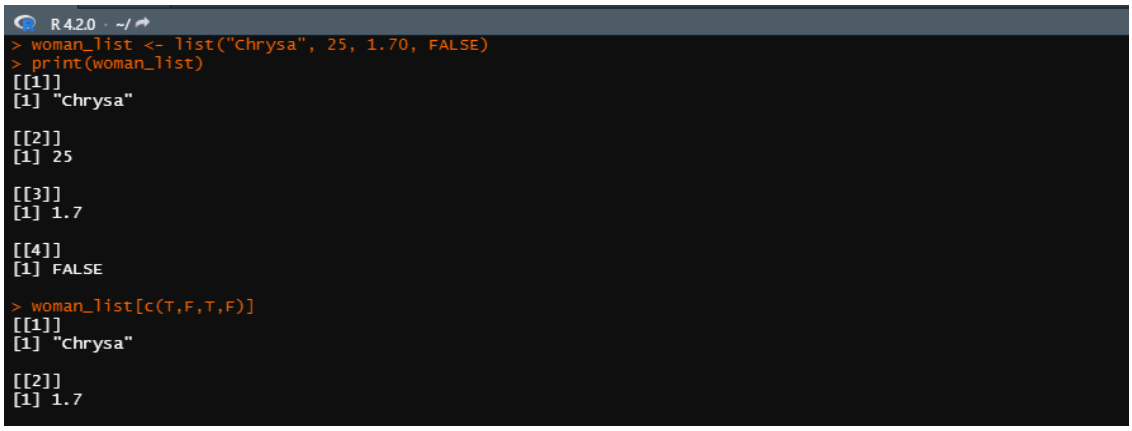
#### 2.2.2.2 Αντιστοίχιση Στοιχείων στη Λίστας

Σε μια δομή δεδομένων όπως είναι η λίστα, τα στοιχεία είναι πάντα αριθμημένα. Η έννοια «αντιστοίχιση των στοιχείων της λίστας» ορίζει μια διαδικασία κατά την οποία ο χρήστης επιλέγει ένα υποσύνολο στοιχείων από αυτήν με σκοπό να τα αναλύσει περαιτέρω ή ενδεχομένως να τα τροποποιήσει. Ο προαναφερόμενος, έχει την δυνατότητα να αναφερθεί στα στοιχεία μιας λίστας με τρόπο ανάλογο με τα ατομικά διανύσματα της R, χρησιμοποιώντας λογικά, ακέραια (με θετική ή και αρνητική τιμή) ή διανύσματα χαρακτήρων.

Για να αναφερθούν τα στοιχεία του υποσυνόλου χρησιμοποιούνται διπλές ή μεμονωμένες αγκύλες ([] ή [[]]). Οι απλές αγκύλες, επιστρέφουν κάθε φορά μια λίστα με μόνο ένα στοιχείο. Έτσι, έχουν την μορφή: list[[1]], list[[2]] κλπ. Αντιθέτως, οι διπλές επιστρέφουν ένα αντικείμενο της κλάσης του στοιχείου που συμπεριλαμβάνεται στην λίστα. Σε αυτή τη περίπτωση, η μορφή θα είναι η εξής: list[[1]][1] κοκ. Τέλος, υπάρχει και περίπτωση του συμβόλου "$" το οποίο επιστρέφει στοιχεία των οποίων τα ονόματα συσχετίζονται με την λίστα, όχι όμως απαραίτητα στην ίδια κλάση. Χρησιμοποιώντας το παραπάνω παράδειγμα:

Έστω ότι ο χρήστης θέλει να αντιστοιχίσει τις πληροφορίες της λίστας woman_list με τις λογικές τιμές TRUE/FALSE ενός διανύσματος. Τα ορίσματα της λίστας woman_list είναι συνολικά 4. Άρα, για να γίνει εφικτή η αντιστοίχιση, πρέπει και τα λογικά ορίσματα του δεύτερου διανύσματος να είναι εξίσου 4. Στις θέσεις τις οποίες η λογική τιμή ισούται με FALSE, δεν μπορεί να γίνει η αντιστοίχιση, οπότε δεν τυπώνεται το όρισμα της λίστας woman_list που βρίσκεται σε ανάλογη θέση. Αντιθέτως, σε κάθε θέση όπου υπάρχει η λογική τιμή TRUE, πραγματοποιείται η αντιστοίχιση. Συνεπώς, τυπώνεται και το όρισμα.

Στο στιγμιότυπο της παρακάτω εικόνας, διακρίνεται πως στην 2$^{η}$ και 4$^{η}$ θέση τα ορίσματα έχουν λάβει την τιμή TRUE, ενώ στην 1$^{η}$ και την 3$^{η}$ την τιμή FALSE. Έτσι, τα ορίσματα της λίστας woman_list που θα αντιστοιχηθούν και έπειτα τυπωθούν θα είναι το 1$^{ο}$ που δείχνει το όνομα και το 3$^{ο}$ που υποδηλώνει το ύψος.

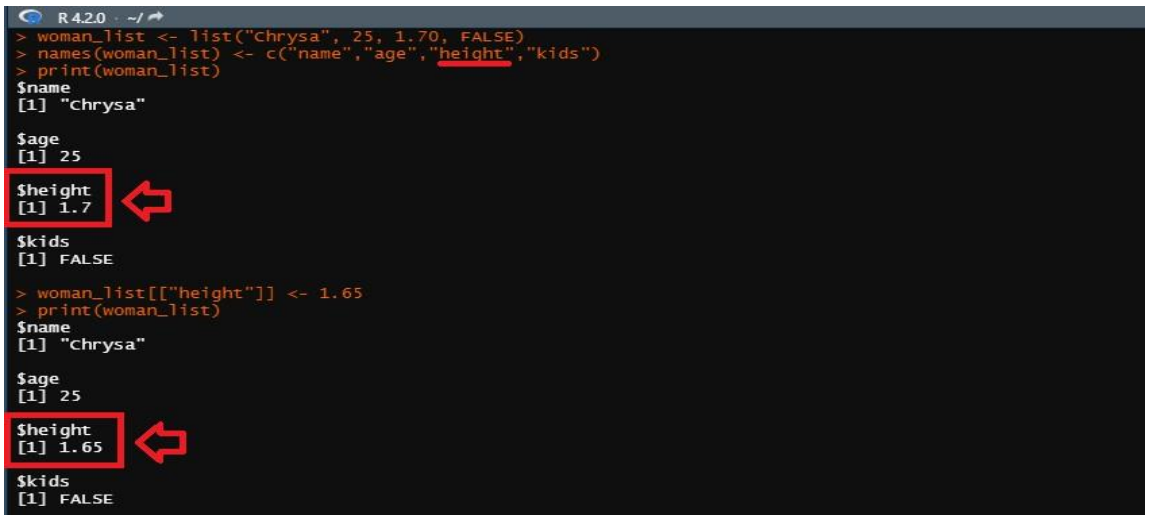



*Εικόνα 37: Στιγμιότυπο αντιστοίχισης στοιχείων σε μια λίστα.*

### 2.2.2.3 Τροποποίηση, Προσθήκη καI Διαγραφή ΣτοIχείων της Λίστας

Όσον αφορά το κομμάτι της τροποποίησης των στοιχείων μιας λίστας, πρόκειται για μια πολύ απλή και συγχρόνως γρήγορη διαδικασία. Ο εκάστοτε προγραμματιστής, μπορεί να τροποποιήσει ένα ή περισσότερα από τα στοιχεία που υπάρχουν μέσα στη λίστα, αποκτώντας σε πρώτη φάση πρόσβαση σε αυτά μέσω της διαδικασίας της αντιστοίχισης και έπειτα εκχωρώντας τους εκ νέου τιμές. Στο παράδειγμα της εικόνας 35, έκτος από την δημιουργία της λίστας woman_list, πραγματοποιείται και η δημιουργία/αντιστοίχιση ονομάτων για το κάθε στοιχείο.

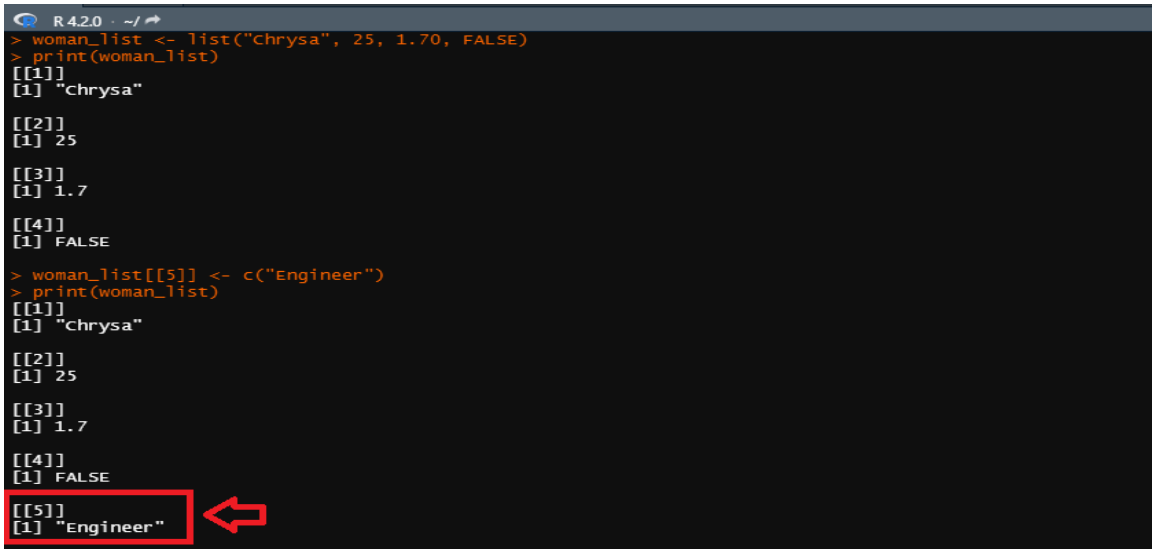
Έστω ότι ο χρήστης επιθυμεί να τροποποιήσει την μεταβλητή του ύψους του οποίου η αρχική τιμή ισούται με 1,70. Η καινούργια τιμή που θέλει να δώσει είναι ίση με 1,65. Για να το κάνει αυτό, αρκεί να χρησιμοποιήσει το όρισμα “height” μέσα σε διπλές αγκύλες ([[]]) και στην συνέχεια να του αναθέσει την νέα τιμή. Κατόπιν εκτέλεσης του προγράμματος, φαίνεται ξεκάθαρα στην οθόνη του λογισμικού πως η τιμή έχει όντως αλλάξει. Στο στιγμιότυπο 38, απεικονίζεται αναλυτικότερα ο κώδικας και το αποτέλεσμα.




*Εικόνα 38: Στιγμιότυπο τροποποίησης στοιχείων σε μια λίστα.*

Για να προστεθεί κάποιο νέο στοιχείο ή στοιχεία στην λίστα, αρκεί ο χρήστης να το δηλώσει, εκχωρώντας το σε μια κενή, μη αντιστοιχισμένη θέση. Στην λίστα woman_list, υπάρχουν 4 στοιχεία. Ο χρήστης επιθυμεί να προσθέσει και ένα 5ο στοιχείο το οποίο θα αναφέρεται στον τίτλο της δουλειάς της συγκεκριμένης

γυναίκας. Εκχωρεί αρχικά μια 5η τιμή μέσα σε διπλές αγκύλες ([[]]) και το επόμενο του βήμα είναι να την αντιστοιχίσει με ένα νέο-κατασκευασμένο διάνυσμα χαρακτήρα το οποίο θα εμπεριέχει την μεταβλητή "Engineer". Το αποτέλεσμα που προκύπτει, αποτυπώνεται στο στιγμιότυπο 39.




*Εικόνα 39: Στιγμιότυπο προσθήκης νέου στοιχείου στη λίστα*

Τέλος, σε περίπτωση που ο χρήστης επιθυμεί να διαγράψει οριστικά κάποιο ή κάποια από τα στοιχεία της λίστας, το μόνο που έχει να κάνει είναι να του αναθέσει την τιμή NULL. Για παράδειγμα, έστω ότι θέλει να διαγράψει το στοιχείο της λίστας που προστέθηκε πιο πάνω ως νέο, δηλαδή το "Engineer". Χρησιμοποιεί το 5ο στοιχείο που βρίσκεται εκχωρημένο στις διπλές αγκύλες και έπειτα του αντιστοιχίζεται η μηδενική μεταβλητή. Έτσι, η λίστα μένει ξανά με τα 4 αρχικά της στοιχεία (βλ. εικόνα 40).


*Εικόνα 40: Στιγμιότυπο διαγραφής στοιχείου από τη λίστα.*

### 2.2.3 Πίνακες (Matrices)

Όπως προαναφέρθηκε στην ενότητα 2.2, οι δομές δεδομένων έχουν συγκεκριμένες διαστάσεις και μπορούν να καταταχθούν σε δύο κατηγορίες βάση της ομογένειας ή της ετερογένειας τους. Ο ψηφιοπίνακας (matrix) είναι μια δομή δεδομένων, η οποία είναι εξ ολοκλήρου διαφορετική από αυτή των διανυσμάτων. Ο λόγος για τον οποίο διαφέρουν είναι διότι οι πίνακες είναι ομογενής και δισδιάστατοι. Αυτό πρακτικά σημαίνει, ότι έχουν δύο διαστάσεις, γραμμές και στήλες αντίστοιχα.

Σκοπός του κάθε πίνακα, είναι να αποθηκεύει και να ταξινομεί τιμές δεδομένων στις παραπάνω γραμμές και στήλες, οι οποίες αποτελούνται από τους βασικούς τύπους δεδομένων (αριθμητικοί, δεκαδικοί κλπ.). Παρ ’όλα αυτά, ένας πίνακας δεν αποκλείεται να αποτελείται από ένα συνδυασμό δύο ή περισσότερων πινάκων. Επίσης, μπορεί να περιλαμβάνει λογικά δεδομένα ή δεδομένων χαρακτήρων, ωστόσο δεν είναι τόσο συχνά χρησιμοποιούμενα. Στον κλάδο της ανάλυσης δεδομένων, οι τιμές των δεδομένων αποθηκεύονται στις γραμμές ενώ οι αντίστοιχες μεταβλητές τους στις στήλες του ορθογώνιου πίνακα.

#### 2.2.3.1 ΔημΙΟυργία Πίνακα

Η δημιουργία ενός ψηφιοπίνακα (matrix) μπορεί να γίνει με τρεις διαφορετικούς τρόπους στην γλώσσα R, όπου όλοι περιλαμβάνουν την χρήση συναρτήσεων.

- **Με την συνάρτηση matrix()**

Όσον αφορά την πρώτη και πιο βασική περίπτωση, αναφέρεται στην χρήση μιας συνάρτησης με όνομα αντίστοιχο με αυτό της δομής, δηλαδή, matrix(). Η σύνταξη της συγκεκριμένης συνάρτησης πραγματοποιείται με την βοήθεια κάποιων δικών της arguments τα οποία δηλώνονται με συγκεκριμένη σειρά ως εξής:

- matrix(data, byrow, nrow, ncol, dimnames)

και αποτελούν:

- data: Τα στοιχεία δεδομένων του πίνακα.
- byrow: Θεωρείται μια μεταβλητή λογικής τιμής. Σε περίπτωση που λάβει την τιμή αληθείας TRUE, τα δεδομένα του πίνακα θα πρέπει να ταξινομηθούν με

βάση τις γραμμές του, αν και από προεπιλογή ταξινομούνται κατά στήλες (byrow = FALSE).

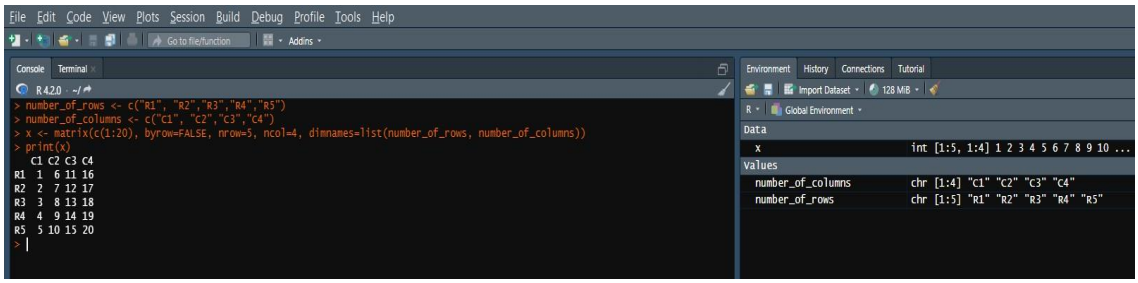
- nrow: Ο αριθμός γραμμών του προς δημιουργία πίνακα.
- ncol: Ο αριθμός στηλών του προς δημιουργία πίνακα.
- dimnames: Ορίζει τα ονόματα που πρόκειται να υπαγορευθούν στις παραπάνω γραμμές και στήλες. Δηλώνεται πάντα κατά την δημιουργία του ψηφιοπίνακα.

Σχετικά με τις μεταβλητές nrow και ncol, δεν είναι αναγκαία προϋπόθεση να οριστούν και οι δύο μεταβλητές, αρκεί να έχει δηλωθεί απλά η μία. Η γλώσσα R, έχει την δυνατότητα να υπολογίζει από μόνη της την δεύτερη μεταβλητή βασιζόμενη στις τιμές των δεδομένων που έχει εισάγει ο χρήστης στο πρόγραμμα. Ακόμα, για να μπορέσουν να οριστούν τα ονόματα στις γραμμές και στήλες του ψηφιοπίνακα, πρέπει αρχικά να έχουν δημιουργηθεί ως διανύσματα και έπειτα η μεταβλητή dimnames που θα τα συμπεριλαμβάνει, να χρησιμοποιηθεί μέσα στην συνάρτηση list(), ώστε να τυπωθούν ταξινομημένα. Στα παρακάτω στιγμιότυπα (37 & 38) απεικονίζεται το ίδιο παράδειγμα με κάποιες παραλλαγές.

Ξεκινώντας με την πρώτη εικόνα, ο προγραμματιστής δημιουργεί στο πρόγραμμα δύο διανύσματα: number_of_rows και number_of _columns, τα οποία δηλώνουν τα ονόματα των γραμμών και στηλών του πίνακα αντίστοιχα. Τα ορίσματα των γραμμών είναι: “R1”,”R2”,”R3”,R4”,”R5” διότι ο πίνακας πρόκειται να έχει 5 γραμμές και τα ορίσματα των στηλών είναι: “C1”,”C2”,”C3”,”C4” εφόσον ο χρήστης επιθυμεί ο πίνακας να έχει 4 στήλες. Ακολουθεί η δήλωση της συνάρτησης matrix(), η οποία συμπεριλαμβάνει:

- τιμές από το 1 έως και το 20,
- τη μεταβλητή byrow ίση με την προεπιλεγμένη τιμή αληθείας FALSE ώστε ο ορθογώνιος πίνακας να δημιουργηθεί με βάση τις στήλες,
- nrow=5 και ncol=4 όπως προ αναφέρθηκε και τέλος
- τα ονόματα (dimnames) των γραμμών και των στηλών που δημιουργήθηκαν παραπάνω μέσα σε μορφή λίστας.

Εκτελώντας τον κώδικα, χρησιμοποιώντας την εντολή: print(matrix), τυπώνεται στην οθόνη ο πίνακας ενώ παράλληλα είναι ευδιάκριτος και στο παράθυρο environment.




*Εικόνα 41: Στιγμιότυπο δημιουργίας matrix με χρήση συνάρτησης matrix().*

Κλείνοντας, στο δεύτερο στιγμιότυπο, το παράδειγμα παραμένει το ίδιο με την διαφορά ότι η τιμή αληθείας byrow αλλάζει από FALSE σε TRUE ώστε αυτή τη φορά ο πίνακας να δημιουργηθεί βάση των γραμμών του. Επιπρόσθετα, παραλείπετε η μεταβλητή ncol διότι όπως σημειώθηκε, ο χρήστης έχει δηλώσει τα δεδομένα και τον αριθμό των γραμμών, οπότε το πρόγραμμα μπορεί να τυπώσει ακριβώς το ίδιο αποτέλεσμα και χωρίς αυτή.


*Εικόνα 42: Στιγμιότυπο δημιουργίας matrix με χρήση συνάρτησης matrix() (παραλλαγή).*

Ένας εναλλακτικός τρόπος με τον οποίο μπορούν να ονομαστούν οι γραμμές και οι στήλες του πίνακα είναι με την βοήθεια των συναρτήσεων rownames() και colnames() αντίστοιχα. Η διαφορά μεταξύ των συναρτήσεων αυτών με την μεταβλητή dimnames, είναι πως η συναρτήσεις αυτές χρησιμοποιούνται μετά την δημιουργία του πίνακα. Για παράδειγμα:

Έστω ότι ο χρήστης επιθυμεί να δημιουργήσει ένα πίνακα με 9 στοιχεία, τα οποία θα κατηγοριοποιηθούν σε 3 γραμμές και 3 στήλες. Κάνοντας χρήση της συνάρτησης matrix(), τυπώνεται ο παρακάτω πίνακας (βλ. εικόνα 39 (1)). Σειρά έχουν οι συναρτήσεις rownames() και colnames(). Ως όρισμα των συναρτήσεων χρησιμοποιείται ο ήδη δημιουργημένος πίνακας (matrix1) και έπειτα δημιουργούνται δύο διανύσματα που περιέχουν τα ονόματα “row1”, “row2”, “row3” για τις γραμμές

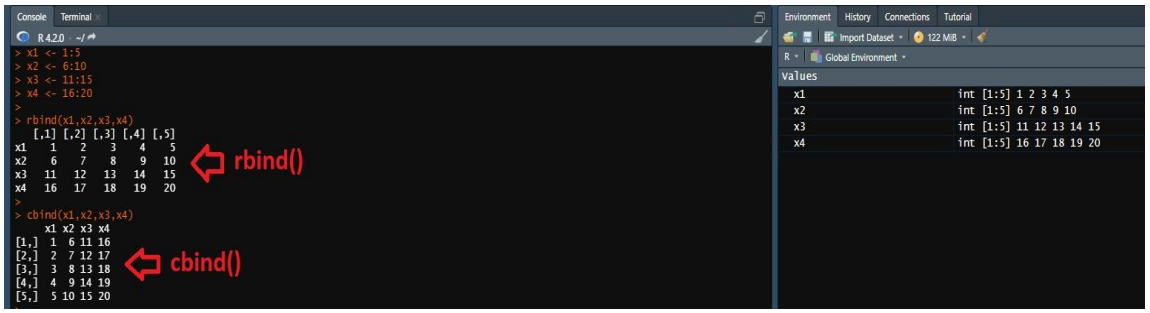
και “col”, “col2” και “col3” για τις στήλες. Έτσι, το αποτέλεσμα που προκύπτει είναι το εξής. (βλ. εικόνα 39 (2)).


*Εικόνα 43: Στιγμιότυπο δημιουργίας ονομάτων γραμμών και στηλών με χρήση των συναρτήσεων rownames() και colnames().*

- **Με τις συναρτήσεις rbind() και cbind().**

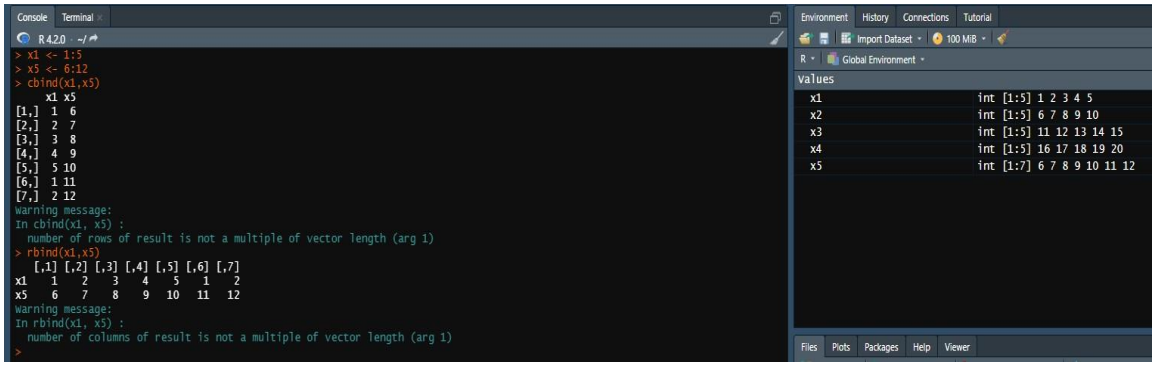
Στην ανάλυση δεδομένων, δεν είναι λίγες οι φορές που για να προκύψει ένας καινούργιος πίνακας πρέπει να γίνει σύνδεση ανάμεσα σε δύο ή περισσότερους πίνακες/διανύσματα ίσου μήκους. Η δημιουργία αυτού, πραγματοποιείται με την βοήθεια των συναρτήσεων rbind() και cbind(). Αυτό που κάνει τις δύο αυτές συναρτήσεις να ξεχωρίζουν μεταξύ τους, είναι πως η rbind() ενώνει τα διαφορετικά διανύσματα/πίνακες με σκοπό να γίνουν οι γραμμές του νέου πίνακα ενώ η cbind() τα συνδυάζει ώστε να αποτελέσουν τις στήλες του νέου πίνακα.

Έστω ότι ο ψηφιοπίνακας που επιθυμεί ο χρήστης να δημιουργήσει, περιλαμβάνει τιμές από το 1 έως και το 20. Σε πρώτη φάση, δημιουργούνται τα διανύσματα x1,x2,x3 και x4 με μήκος ίσο με 5 το καθένα. Με βάση την εικόνα 40, συνδυάζοντας τα τέσσερα αυτά διανύσματα με τις συναρτήσεις rbind() και cbind() δημιουργείται ο εκάστοτε πίνακας όπου στην πρώτη περίπτωση καθορίζεται από τις γραμμές (όπως φαίνεται και με κόκκινα βέλη), ενώ στην δεύτερη από τις στήλες.




*Εικόνα 44: Στιγμιότυπο δημιουργίας matrix με χρήση rbind() & cbind().*

Συχνή είναι και περίπτωση, όπου τα αρχικά διανύσματα δεν είναι ισοδύναμου μεγέθους. Εκτελώντας είτε την μια συνάρτηση είτε την άλλη, αυτό που θα προκύψει σαν αποτέλεσμα, είναι πως θα εισαχθούν κανονικά οι τιμές του μεγαλύτερου διανύσματος αλλά οι τιμές του μικρότερου θα επαναλαμβάνονται κυκλικά μέχρι να συμπληρωθεί εντελώς ο πίνακας. Παρ' όλα αυτά, το πρόγραμμα θα εμφανίσει προειδοποιητικό μήνυμα για την παραπάνω ανισότητα. Έστω ότι το διάνυσμα x1 περιέχει 5 τιμές ενώ το διάνυσμα x5, 7 τιμές. Αν για παράδειγμα εκτελεστεί η συνάρτηση cbind(), οι δύο πρώτες τιμές του διανύσματος x1 θα ξανά εισαχθούν μέχρι να θεωρηθεί ότι το μήκος του είναι 7 και όχι 5. (βλ. εικόνα 41).



*Εικόνα 45: Στιγμιότυπο δημιουργίας matrix διαφορετικού μεγέθους με χρήση rbind() & cbind().*

- **Χρήση της συνάρτησης dim().**

Ο τελευταίος τρόπος για την δημιουργία ενός matrix, είναι με την βοήθεια της συνάρτησης dim() η οποία ονομάζεται και συνάρτηση διάστασης. Η διαδικασία με την οποία λειτουργεί η συγκεκριμένη συνάρτηση είναι τροποποιώντας τις διαστάσεις ενός προ υπάρχων διανύσματος με σκοπό να μετατραπεί εκ νέου σε ένα ψηφιοπίνακα.

Στο διάνυσμα αυτό, εκχωρούνται δύο ακέραιες τιμές ώστε η μια να αντιστοιχεί στις γραμμές του νέου πίνακα ενώ η δεύτερη στις στήλες του. Για παράδειγμα, έστω το διάνυσμα με τιμές από το 1 έως και το 25 το οποίο πρόκειται να αλλαχτεί σε πίνακα με διαστάσεις 5x5. Στο στιγμιότυπο 42, απεικονίζεται το αντίστοιχο πρόγραμμα.

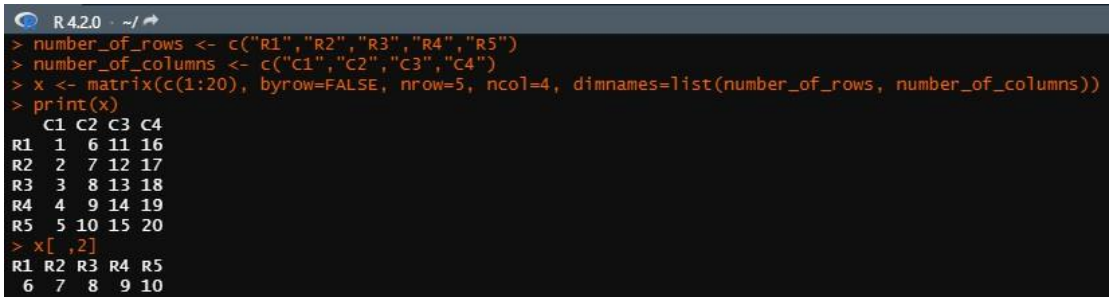

*Εικόνα 46: Στιγμιότυπο δημιουργίας matrix με χρήση συνάρτησης dim().*

### 2.2.3.2 Πρόσβαση σε Στοιχεία του Matrix

Ο τρόπος με τον οποίο μπορεί ένας χρήστης να αποκτήσει πρόσβαση στα στοιχεία ενός matrix είναι κάνοντας χρήση των απλών αγκυλών ([]) στην μορφή:

- matrix[row,col]

όπου row και col τα διανύσματα γραμμών και στηλών αντίστοιχα. Όπως και στα ατομικά διανύσματα, μπορούν να χρησιμοποιηθούν οι ίδιες τεχνικές αντιστοίχισης και εδώ. Ο χρήστης, έχει την δυνατότητα να καθορίσει την ποσότητα των γραμμών και των στηλών σε ένα διάνυσμα και στην συνέχεια να την χρησιμοποιήσει ώστε να αντιστοιχίσει ένα matrix. Σε περίπτωση που μέσα στις αγκύλες παραλείψει είτε τον αριθμό των γραμμών είτε των στηλών και το πεδίο παραμείνει άδειο, η γλώσσα R θα επιλέξει από μόνη όλες τις γραμμές ή τις στήλες. (βλ. εικόνα 47)




*Εικόνα 47: Στιγμιότυπο αντιστοίχισης στοιχείων matrix.*

Βέβαια όπως φαίνεται και στην προηγούμενη εικόνα, αν πραγματοποιηθεί κάτι τέτοιο, το τελικό αποτέλεσμα που θα προκύψει θα είναι διάνυσμα, και όχι πίνακας. Για να αποφευχθεί αυτή η συμπεριφορά, γίνεται χρήση ενός argument με όνομα drop το οποίο λαμβάνει την λογική τιμή FALSE. (βλ. εικόνα 48)


*Εικόνα 48: Στιγμιότυπο αντιστοίχισης στοιχείων matrix (2).*

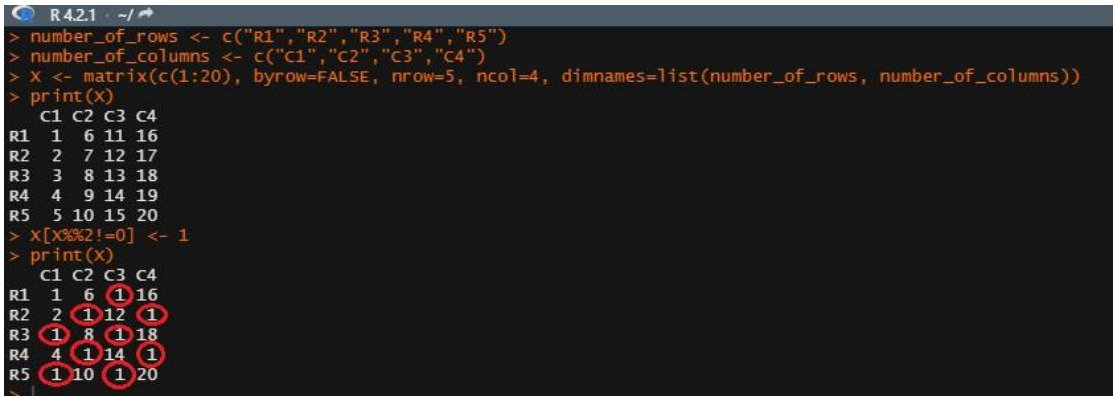
Ακόμη, αν ο χρήστης θέλει να εξαιρέσει κάποιες γραμμές ή στήλες από τον πίνακα, μπορεί να χρησιμοποιήσει αρνητικές τιμές ακέραιων αριθμών. Όταν χρησιμοποιεί διανύσματα λογικών τιμών ισοδύναμου μήκους με το μήκος του πίνακα, μπορεί να καθορίσει τις απαιτούμενες γραμμές και στήλες. Μια τέτοια ενέργεια, εξαιρεί όσες από τις διαστάσεις αντιστοιχούν στην τιμή αληθείας FALSE και συμπεριλαμβάνει μόνο όσες αντιστοιχούν στην τιμή TRUE. Από την άλλη πλευρά, όταν τα λογικά διανύσματα είναι διαφορετικού μήκους με το μέγεθος του πίνακα, η R ανακυκλώνει το διάνυσμα. Όσον αφορά τα διανύσματα που περιλαμβάνουν γραμμές και στήλες οι οποίες έχουν μετονομαστεί, ο χρήστης μπορεί να αποκτήσει πρόσβαση στις παραπάνω γραμμές και στήλες χρησιμοποιώντας μόνο τα ονόματα. Τέλος, όλες οι μέθοδοι αντιστοίχισης για το matrix μπορούν να χρησιμοποιηθούν συνδυαστικά μεταξύ τους. (βλ. εικόνα 49)


*Εικόνα 49: Στιγμιότυπο αντιστοίχισης στοιχείων matrix (3).*

#### 2.2.3.3 Τροποποίηση ενός Matrix

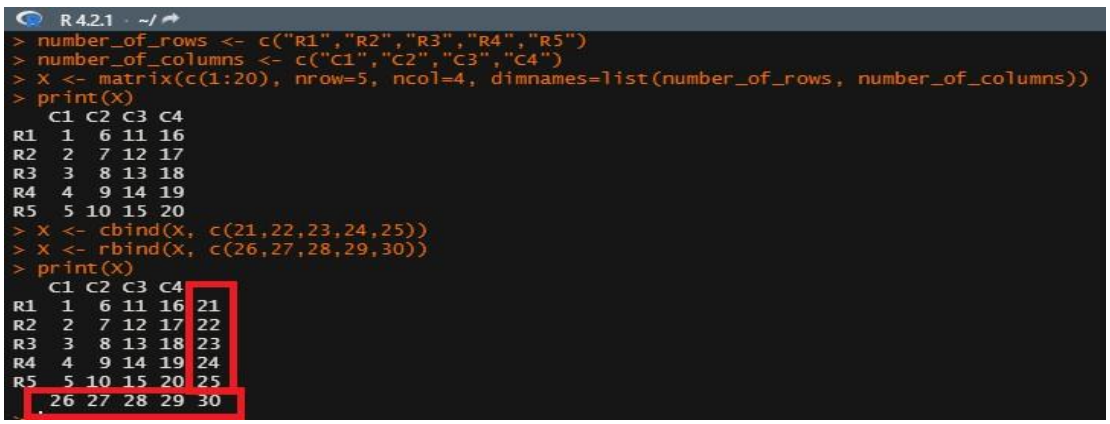
Για να επιτευχθεί η τροποποίηση ενός πίνακα, αρκεί να συνδυαστούν οι τεχνικές αντιστοίχισης και ο τελεστής %%. Με βάση το matrix που έχει ήδη δημιουργήσει ο χρήστης:

Έστω ότι ο χρήστης θέλει να τροποποιήσει δια του δύο όλα τα στοιχεία που υπάρχουν στις γραμμές και στις στήλες του πίνακα. Η εν λόγω τροποποίηση δεν θα είναι ακριβής εφόσον ο πίνακας έχει και μονά και ζυγά στοιχεία. Για τον λόγο αυτό, χρησιμοποιώντας τον τελεστή %% , θα δηλώσει στο πρόγραμμα πως όσα από τα στοιχεία είναι άρτιου αριθμού θα παραμείνουν ως έχουν, ενώ όσα από τα στοιχεία αποτελούν στοιχεία περιττού αριθμού, την θέση τους θα πάρει ο αριθμός 1. Το τελικό αποτέλεσμα διακρίνεται στην εικόνα 50.




*Εικόνα 50: Στιγμιότυπο τροποπίιησης matrix.*

Μια εναλλακτική μέθοδος τροποποίησης ενός matrix, είναι η προσθήκη καινούργιων γραμμών και στηλών. Η μέθοδος αυτή επιτυχάνεται με τις συναρτήσεις rbind() και cbind() οι οποίες αναλύθηκαν παραπάνω και ως τρόπος δημιουργίας ενός πίνακα. Αν για παράδειγμα, ο χρήστης θέλει να προσθέσει στον παραπάνω πίνακα μια έξτρα στήλη αποτελούμενη από τους αριθμούς: 21,22,23,24,25 και μια ακόμα γραμμή με συμπεριλαμβανόμενους τους αριθμούς: 26,27,28,29,30 θα εκτελέσει τον κώδικα της εικόνας 51.




*Εικόνα 51: Στιγμιότυπο τροποποίσης matrix (2).*

### 2.2.4 Πλαίσια Δεδομένων (Data Frames)

Τα πλαίσια δεδομένων ή αλλιώς data frames είναι μια δομή δεδομένων που κατηγοριοποιούνται ως μια ειδική περίπτωση λίστας. Αυτό που τα ξεχωρίζει από την κλασσική δομή της λίστας είναι πως όλα τα στοιχεία έχουν ισοδύναμο μήκος. Ωστόσο, είναι δισδιάστατα με κάθε στήλη να αντιστοιχίζεται στην τιμή μιας μεταβλητής, ενώ κάθε γραμμή αντιστοιχίζεται σε ένα σύνολο τιμών από την κάθε στήλη. Τα πλαίσια δεδομένων έχουν κάποια συγκεκριμένα χαρακτηριστικά που οφείλουν να τηρούνται και σε αυτά συμπεριλαμβάνονται:

- τα αποθηκευμένα δεδομένα μπορεί να είναι αριθμητικά ή ακόμα να έχουν την μορφή χαρακτήρα,
- το όνομα της κάθε στήλης απαγορεύεται να είναι κενό ενώ ταυτόχρονα πρέπει να συμπεριλαμβάνει τον ίδιο αριθμό αντικείμενων και τέλος
- το όνομα της κάθε γραμμής πρέπει να είναι μοναδικό.

#### 2.2.4.1 Δημιουργία Πλαισίου Δεδομένων

Για να κατασκευαστούν στην γλώσσα R τα πλαίσια δεδομένων υπάρχουν τρεις τρόποι και πιο συγκεκριμένα τρεις συναρτήσεις οι οποίες είναι:

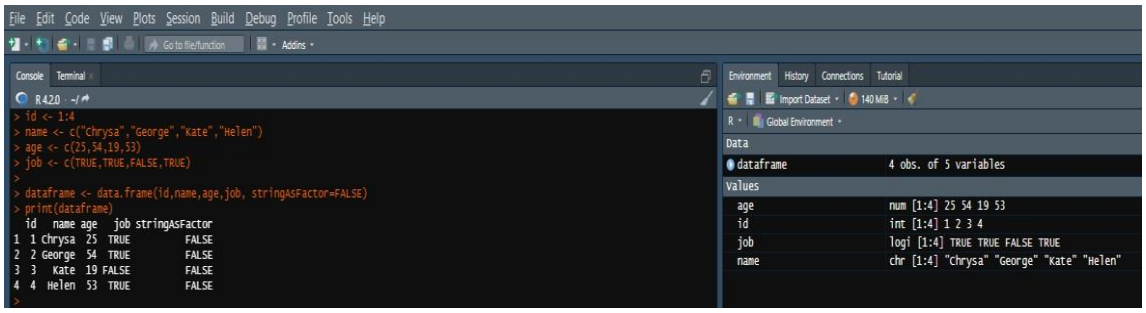
1. as.data.frame(), η οποία αποτελεί μια συνάρτηση αλλαγής καθώς ο τρόπος που κατασκευάζει πλαίσια δεδομένων είναι μετατρέποντας αντικείμενα μιας συγκεκριμένης μορφής σε αντικείμενα της τάξης data frame.
2. read.table(), η οποία αντλεί και διαβάζει δεδομένα από κάποιο εξωτερικό αρχείο και τα μετατρέπει εξίσου σε πλαίσιο δεδομένων και τέλος
3. η συνάρτηση data.frame().

Για λόγους ευκολίας αλλά και έκτασης της εργασίας θα αναλυθεί περισσότερο μόνο ο τελευταίος τρόπος που είναι και ο βασικότερος.

Για να χρησιμοποιηθεί η συνάρτηση data.frame(), πρέπει αρχικά ο χρήστης να δημιουργήσει κάποια διανύσματα με τα δεδομένα που επρόκειτο να περιέχονται μετά στο πλαίσιο δεδομένων. Έστω ότι θέλει να κατασκευάσει 4 από αυτά, με πληροφορίες για κάποιους ανθρώπους όπως το όνομα, την ηλικία, το αν δουλεύουν ή

όχι καθώς θα έχουν και από μια αναγνωριστική μεταβλητή ο καθένας (id). Στην συνέχεια, αυτές οι πληροφορίες θα εισαχθούν ως ορίσματα στην συνάρτηση του πλαισίου.

Ένα μειονέκτημα της συνάρτησης data.frame() είναι πως έχει την ιδιότητα, από προεπιλογή, να μετατρέπει τα δεδομένα χαρακτήρων σε παράγοντες, μια δομή η οποία πρόκειται να αναλυθεί παρακάτω. Ωστόσο, η λογική μεταβλητή stringAsFactors αποτελεί λύση στην περίπτωση αυτή. Θέτοντας την ίση με την τιμή FALSE μπορεί να αποφευχθεί η παραπάνω επιλογή. Παρακάτω στην εικόνα 52, φαίνεται το παραπάνω παράδειγμα.




*Εικόνα 52: Στιγμιότυπο δημιουργίας data frame με την συνάρτηση data.frame().*

### 2.2.4.2 Πρόσβαση σε Στοιχεία του Data Frame

Για να αποκτήσει ένας χρήστης πρόσβαση στα στοιχεία ενός πλαισίου δεδομένων χρησιμοποιεί όπως και στις λίστες είτε απλές ([]), είτε διπλές αγκύλες ([[]]), είτε το σύμβολο $. Με βάση το παράδειγμα του στιγμιότυπου 51:

Έστω ότι ο χρήστης θέλει να αποκτήσει πρόσβαση στα παραπάνω ονόματα του πλαισίου δεδομένων. Χρησιμοποιώντας την μεταβλητή dataframe με τις μονές αγκύλες να συμπεριλαμβάνουν το όρισμα name σε διπλά εισαγωγικά, το αποτέλεσμα που θα εμφανιστεί στην οθόνη θα είναι σε μορφή λίστας. Αντιθέτως, αν χρησιμοποιηθούν οι διπλές αγκύλες με διπλά εισαγωγικά ή το σύμβολο $ με μονά εισαγωγικά, θα εμφανιστούν σε μορφή διανύσματος. (βλ. εικόνα 53)

```
> id <- 1:4
> name <- c("Chrysa","George","Kate","Helen")
> age <- c(25,54,19,53)
> job <- c(TRUE,TRUE,FALSE,TRUE)
> dataframe <- data.frame(id,name,age,job, stringAsFactor=FALSE)
> print(dataframe)
  id   name age   job stringAsFactor
1  1 Chrysa  25  TRUE          FALSE
2  2 George  54  TRUE          FALSE
3  3   Kate  19 FALSE          FALSE
4  4  Helen  53  TRUE          FALSE
> dataframe["name"]
    name
1 Chrysa
2 George
3   Kate
4  Helen
> dataframe[["name"]]
[1] "Chrysa" "George" "Kate"   "Helen"
> dataframe$'name'
[1] "Chrysa" "George" "Kate"   "Helen"
```

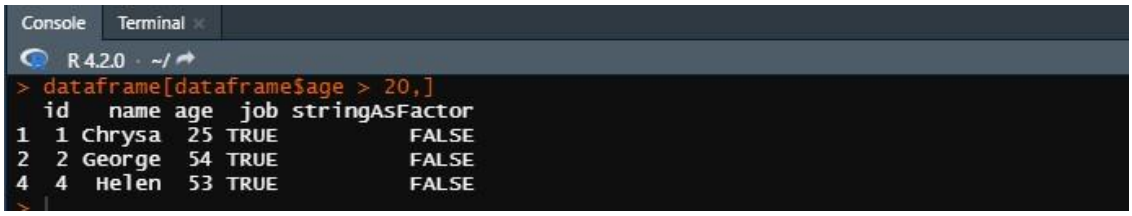
*Εικόνα 53: Στιγμιότυπο αντιστοίχισης στοιχείων σε ένα data frame.*

Μια ακόμη χρήσιμη πληροφορία όσον αφορά την πρόσβαση στοιχείων ενός data frame, είναι πως ο χρήστης μπορεί να χρησιμοποιήσει ορισμένους δείκτες στις γραμμές ή τις στήλες με σκοπό να εισέλθει σε συγκεκριμένα στοιχεία. Για παράδειγμα, έστω ότι θέλει να αποκτήσει πρόσβαση στις 2 πρώτες γραμμές και στις 2 πρώτες στήλες του πλαισίου δεδομένων. Θα πρέπει να χρησιμοποιήσει τον δείκτη (:) στην μεταβλητή dataframe σε μονές αγκύλες. Ο δείκτης αυτός δηλώνει το εύρος της γραμμής ή της στήλης των στοιχείων που πρόκειται να εμφανιστεί. Έτσι, θα εκτελέσει τον κώδικα της εικόνας 54.

```
> dataframe[1:2,1:2, drop=FALSE]
  id   name
1  1 Chrysa
2  2 George
>
```

*Εικόνα 54: Στιγμιότυπο αντιστοίχισης στοιχείων σε ένα data frame (2).*

Μια τελευταία δυνατότητα που παρέχεται στον χρήστη για σκοπούς πρόσβασης, είναι πως μπορεί να χρησιμοποιήσει ορισμένα queries. Τα συγκεκριμένα, έχουν την μορφή ερωτημάτων με στόχο να επιλεχτούν μόνο συγκεκριμένα στοιχεία από την λίστα του πλαισίου. Εν παραδείγματι, έστω πως επιθυμεί να εμφανίσει κάποιες πληροφορίες σχετικά με τα στοιχεία του ορίσματος της ηλικίας. Για να επιτευχθεί αυτό, το ερώτημα που πρέπει να τεθεί στο πρόγραμμα είναι “Τύπωσε όλους τους χρήστες των οποίων η ηλικία ξεπερνάει τα 20 έτη”. Θα χρειαστεί το όρισμα “age” της μεταβλητής dataframe να εκχωρηθεί μέσα σε αγκύλες της μορφής [dataframe$age] και να συγκριθεί με τον αριθμό 20, χρησιμοποιώντας τον τελεστή ανισότητας “>”. Τέλος, θα πρέπει μετά το σύμβολο της ανισότητας να ακολουθήσει “,”. Ο κώδικας και το τυπωμένο αποτέλεσμα ακολουθούν στην εικόνα 55.

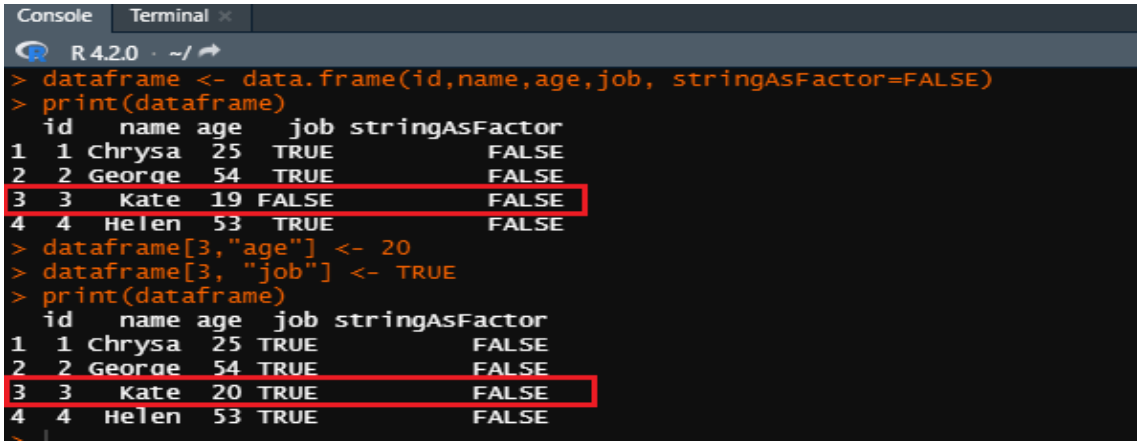



*Εικόνα 55: Στιγμιότυπο αντιστοίχισης στοιχείων σε ένα data frame (3).*

### 2.2.4.3 Τροποποίηση Τιμών ενός Data Frame

Για να επιτευχθεί η διαδικασία της τροποποίησης συνδυάζονται για άλλη μια φορά οι τεχνικές αντιστοίχισης και στην συνέχεια μετονομασίας ή αλλαγής τιμής. Μέσω του παρακάτω παραδείγματος:

Έστω ότι στο πλαίσιο δεομένων που έχει κατασκευάσει ο χρήστης, επιθυμεί να τροποποιήσει κάποιες πληροφορίες του χρήστη "Kate". Όπως μπορεί να παρατηρήσει κανείς, η συγκεκριμένη γυναίκα είναι 19 ετών και προς το παρόν άνεργη. Συνεπώς, αυτές είναι και οι πληροφορίες που πρόκειται να αλλάξουν. Η ηλικία επρόκειτο να τροποποιηθεί από 19 σε 20, ενώ η επαγγελματική της κατάσταση από άνεργη σε εργαζόμενη. Αρχικά, θα πρέπει στο πλαίσιο dataframe να εκχωρηθεί μέσα σε μονές αγκύλες και να χρησιμοποιηθεί ως πρώτο όρισμα ο αριθμός της γραμμής στην οποία βρίσκεται το στοιχείο που θα τροποποιηθεί ενώ ως δεύτερο όρισμα μέσα σε διπλά εισαγωγικά και στην ίδια αγκύλη το στοιχείο. Τέλος, δίνεται η νέα επιθυμητή ηλικία. Η ίδια μέθοδος ακολουθείται και για την νέα τιμή της επαγγελματικής κατάστασης. (βλ. εικόνα 56)

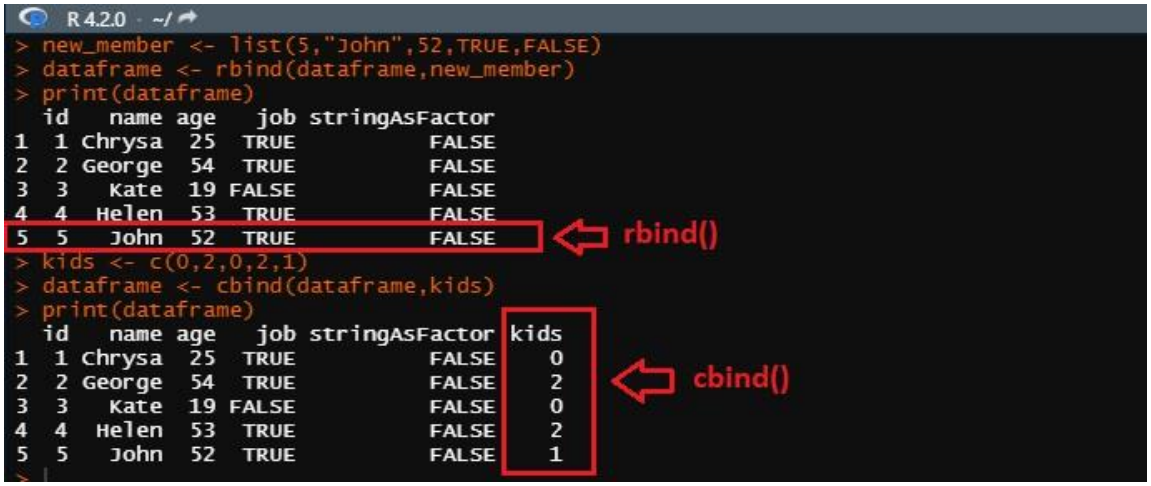



*Εικόνα 56: Στιγμιότυπο τροποποίησης στοιχείων σε ένα πλαίσιο δεδομένων.*

#### 2.2.4.4 Προσθήκη & Διαγραφή Γραμμών & Στηλών

Για να καταφέρει ένας χρήστης να προσθέσει επιπλέον γραμμές σε ένα πλαίσιο δεδομένων, αρκεί να κάνει χρήση της συνάρτησης rbind() η οποία είναι ήδη γνωστή από την ενότητα 2.2.3.1. Στο παράδειγμα dataframe, υπάρχουν 4 γραμμές. Ο χρήστης επιθυμεί να προσθέσει ακόμη μια. Η έξτρα αυτή γραμμή, θα περιλαμβάνει έναν 5ο χρήστη αντίστοιχα. Θα πρέπει σε πρώτη φάση να δημιουργηθεί ο νέος χρήστης και μέσα σε μορφή λίστας με χρήση της συνάρτησης list() και θα συμπεριλαμβάνει όλες του τις πληροφορίες, οι οποίες θα είναι παρόμοιου τύπου με αυτές των προηγούμενων χρηστών. Θα περιλαμβάνει δηλαδή ένα αναγνωριστικό id, όνομα, ηλικία, επαγγελματική κατάσταση καθώς και το όρισμα stringAsFactors ίσο με FALSE. Έπειτα, η συνάρτηση rbind() θα δεχθεί ως ορίσματα τον νέο χρήστη και το ήδη υπάρχων πλαίσιο δεδομένων ώστε να δημιουργηθεί το καινούργιο με την παραπάνω γραμμή.

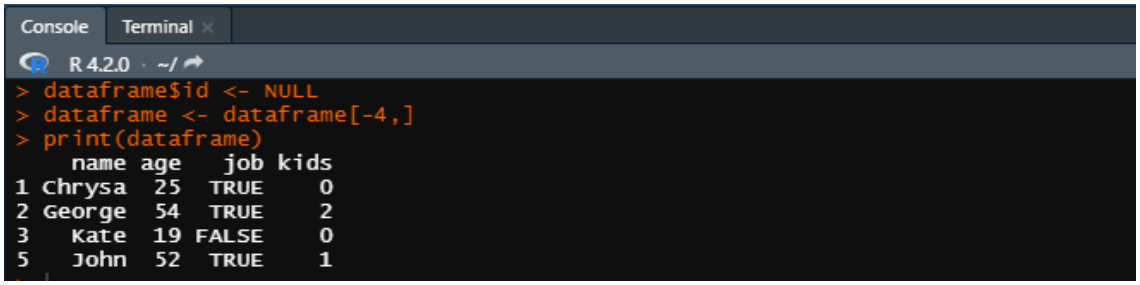
Αντίστοιχα, σε περίπτωση που ο χρήστης θέλει να προσθέσει και κάποια επιπλέον στήλη στο dataframe, θα χρησιμοποιήσει την εξίσου γνωστή συνάρτηση cbind(). Έστω ότι η έξτρα στήλη που επιθυμεί να προσθέσει θα αναφέρεται σε πληροφορίες οικογενειακής κατάστασης, αν δηλαδή οι χρήστες έχουν παιδιά και αν ναι πόσα. Θα δημιουργηθεί αρχικά η στήλη σε μορφή διανύσματος με χρήση της συνάρτησης c() και θα περιλαμβάνει ως ορίσματα ακέραιες τιμές που θα αντιστοιχούν στον αριθμό τον παιδιών. Τέλος, θα χρησιμοποιηθεί η συνάρτηση cbind() με ορίσματα το πλαίσιο δεδομένων και την καινούργια στήλη. Στο στιγμιότυπο 57 διακρίνεται το παράδειγμα αυτό με τυπωμένο στην οθόνη το data frame με τις νέες διαστάσεις.




*Εικόνα 57: Στιγμιότυπο προσθήκης γραμμών και στηλών σε data frame.*

Το κομμάτι της διαγραφής μιας στήλης από το πλαίσιο δεδομένων, αποτελεί μια πολύ απλή διαδικασία καθώς το μόνο που έχει να κάνει ο χρήστης είναι να αναθέσει προς διαγραφή στήλη ίση με την τιμή NULL. Για παράδειγμα, έστω ότι θέλει να σβήσει την στήλη που περιλαμβάνει τα αναγνωριστικά id του κάθε χρήστη. Θα επιλέξει την στήλη id του πλαισίου και θα την εξισώσει απλά με την μηδενική μεταβλητή.

Από την άλλη πλευρά, η διαγραφή γραμμών από το πλαίσιο δεδομένων θεωρείται λίγο διαφορετική. Λόγος για αυτό, αποτελεί το γεγονός πως εδώ δεν χρησιμοποιείται η μεταβλητή NULL, αλλά μια αρνητική τιμή ακέραιου αριθμού μέσα σε αγκύλες. Ο αριθμός αυτός καθορίζεται ανάλογα με το ποια γραμμή επιθυμεί ο χρήστης να σβήσει. Με απλά λόγια, αν ο χρήστης θέλει να διαγράψει την 4η γραμμή η οποία περιλαμβάνει όλες τις πληροφορίες του χρήστη "Helen", ο αρνητικός αριθμός που θα δηλώσει στο πρόγραμμα θα είναι [-4, ]. Δίπλα από την αρνητική τιμή, θα ακολουθήσει "," και μια κενή θέση η οποία θα δηλώνει ότι θα πρέπει να τροποποιηθεί κάποια άλλη στήλη πέρα από αυτήν που έχει τεθεί προς διαγραφή. Η εικόνα 58 που ακολουθεί παρουσιάζει ένα αντίστοιχο παράδειγμα διαγραφής ορισμένων διαστάσεων.




*Εικόνα 58: Στιγμιότυπο διαγράφης γραμμών και στηλών σε data frame.*

### 2.2.5 Πίνακες Πολλαπλών Διαστάσεων(Arrays)

Οι πίνακες πολλαπλών διαστάσεων ή αλλιώς arrays αποτελούν μια εκ φύσεως ομογενή δομή δεδομένων, με τα δεδομένα να αποθηκεύονται σε μορφή πολυδιάστατου ψηφιοπίνακα. Αυτό σημαίνει ότι δεν υπάρχει κανένας περιορισμός στον αριθμό των διαστάσεων του πίνακα, συνεπώς και στα δεδομένα που είναι εφικτό να αποθηκεύσει. Παρ 'όλα αυτά οι τιμές των δεδομένων αυτών είναι απαραίτητο να έχουν το ίδιο τύπο.

#### 2.2.5.1 ΔημΙΟυργία Array

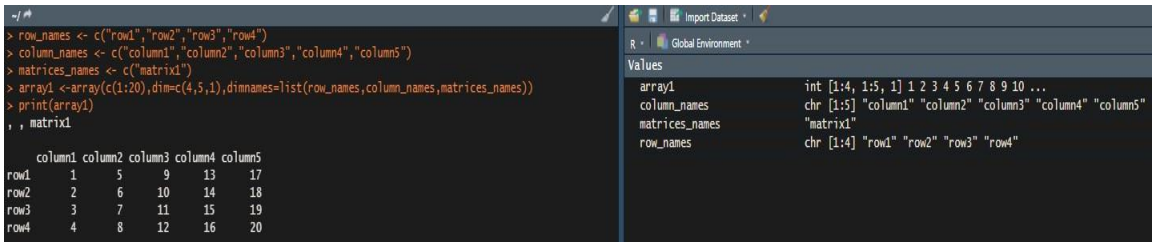
Υπάρχουν διάφοροι τρόποι για να δημιουργηθεί ένας πίνακας πολλαπλών διαστάσεων στην γλώσσα R. Ο πιο γνωστός από αυτούς, θεωρείται η συνάρτηση array(). Όπως και στην κατασκευή ενός matrix, η σύνταξη της συνάρτησης array() έχει επίσης δικά της arguments. Η βασική μορφή της είναι:

- array(data,dim=c(row_size,column_size,matrices),dimnames=list(row_names, column_names,matrices_names))

όπου:

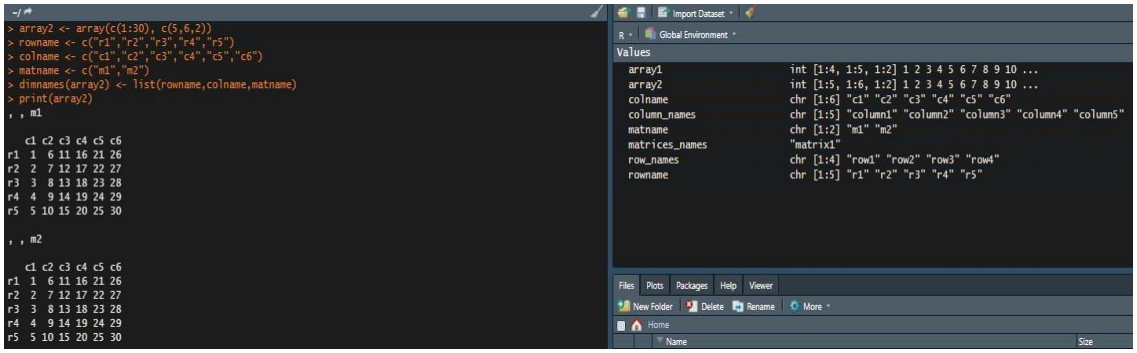
- data: Τα δεδομένα που θα υπάρχουν μέσα στον πίνακα.
- dim: Διάνυσμα διαστάσεων του πίνακα.
- row_size: Ο συνολικός αριθμός των γραμμών του προς κατασκευή πίνακα.
- column_size: Ο συνολικός αριθμός των στηλών του προς κατασκευή πίνακα.
- matrices: Αν και πόσους ψηφιοπίνακες περιλαμβάνει.
- dimnames: Λίστα με κατηγοριοποιημένα τα ονόματα των διαστάσεων του array. Ορίζεται κατά την διάρκεια κατασκευής του πίνακα.
- row_names: Τα ονόματα όλων των γραμμών σε μορφή διανύσματος.
- column_names: Τα ονόματα όλων των στηλών σε μορφή διανύσματος.
- matrices_names: Διάνυσμα που συμπεριλαμβάνει τα ονόματα όσων matrix υπάρχουν στον πίνακα.

Για να δημιουργηθεί το array αρκεί πρώτα να δηλωθούν τα ονόματα (dimnames) που θα αποτελούν τις γραμμές, τις στήλες αλλά και τους ορθογώνιους πίνακες, αν υπάρχουν. Έτσι, δημιουργούνται τα διανύσματα: row_names, column_names και matrices names αντίστοιχα. Στο παράδειγμα που ακολουθεί ο χρήστης έχει επιλέξει να κατασκευάσει έναν πίνακα με 4 γραμμές, 5 στήλες καθώς και 1 matrix. Η υπόλοιπη διαδικασία είναι πανομοιότυπη με αυτή της προηγούμενης δομής πίνακα, δηλαδή του matrix. Στο στιγμιότυπο 59 φαίνεται το παράδειγμα.



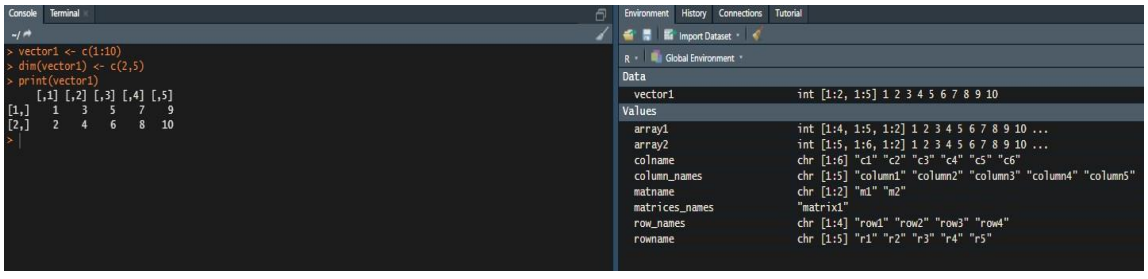

*Εικόνα 59: Στιγμιότυπο δημιουργίας array με χρήση της συνάρτησης array().*

Επίσης, η μετονομασία των γραμμών, των στηλών αλλά και των matrix μπορεί να πραγματοποιηθεί με μια παραλλαγή της μεταβλητής dimnames. Χρησιμοποιώντας την αυτήν την φορά ως συνάρτηση. Το γεγονός που κάνει τον πρώτο τρόπο να διαφοροποιείται από τον δεύτερο, είναι πως όταν χρησιμοποιείται η dimnames ως συνάρτηση, τα ονόματα δηλώνονται έπειτα της κατασκευής του πίνακα. (βλ. εικόνα 60).




*Εικόνα 60: Στιγμιότυπο δημιουργίας array με εξωτερική μετονομασία με χρήση συνάρτησης dimnames().*

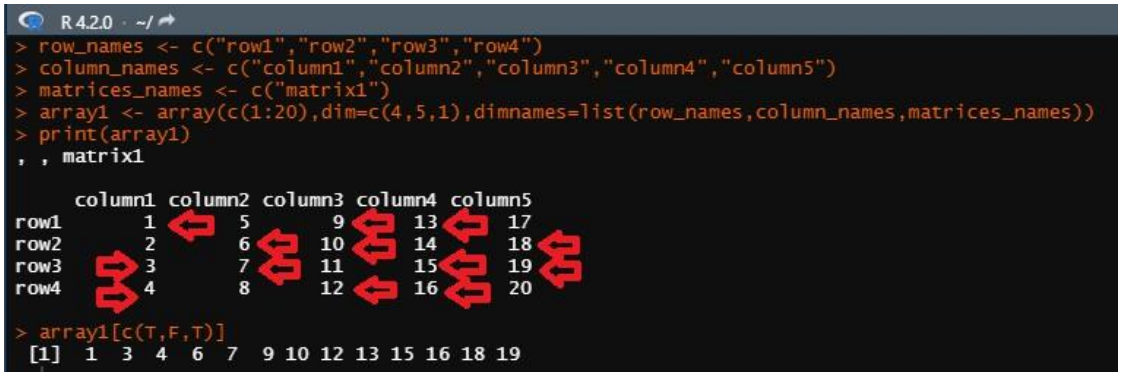
Τέλος, όπως αναφέρθηκε στους τρόπους δημιουργίας ενός ψηφιοπίνακα, συμπεριλαμβάνεται και η συνάρτηση διάστασης dim(). Το ίδιο συμβαίνει και εδώ. Με την βοήθεια της συγκεκριμένης συνάρτησης, οποιοδήποτε διάνυσμα μπορεί να μετατραπεί σε πίνακα πολλαπλών διαστάσεων. (βλ. εικόνα 61).




*Εικόνα 61: Στιγμιότυπο δημιουργίας array με χρήση συνάρτησης dim().*

#### 2.2.5.2 Αντιστοίχιση Στοιχείων του Array

Ένας χρήστης μπορεί να αποκτήσει πρόσβαση στα στοιχεία ενός πίνακα πολλαπλών διαστάσεων και να τα αντιστοιχίσει, χρησιμοποιώντας τα μέσα σε μονές αγκύλες ([]). Και εδώ, μπορούν να χρησιμοποιηθούν χαρακτήρες, λογικές τιμές, θετικοί ακέραιοι και αρνητικοί ακέραιοι. Η διαδικασία που ακολουθείται είναι ακριβώς ίδια με την αντιστοίχιση στοιχείων που συμβαίνει και στην περίπτωση της λίστα ή του πίνακα που αναφέρθηκαν παραπάνω (βλ. εικόνα 62).




*Εικόνα 62: Στιγμιότυπο αντιστοίχισης στοιχείων σε ένα array.*

Κάτι το οποίο ο χρήστης πρέπει να δώσει ιδιαίτερη σημασία όταν δουλεύει με αντιστοίχιση λογικών στοιχείων σε έναν πίνακα πολλαπλών διαστάσεων, είναι πως σε περίπτωση που τα στοιχεία αυτά είναι μικρότερου μεγέθους από τις διαστάσεις του array, ανακυκλώνονται. Όπως δηλαδή, συμβαίνει και στην δημιουργία ενός matrix με ένωση δύο ή περισσότερων πινάκων διαφορετικού μεγέθους. Έτσι, αν επιλέξει μόνο μια καταχώρηση οποιασδήποτε διάστασης ενός πίνακα, το αποτέλεσμα που θα προκύψει θα είναι matrix. Για να μην συμβεί αυτό, γίνεται χρήση του ορίσματος drop=FALSE.

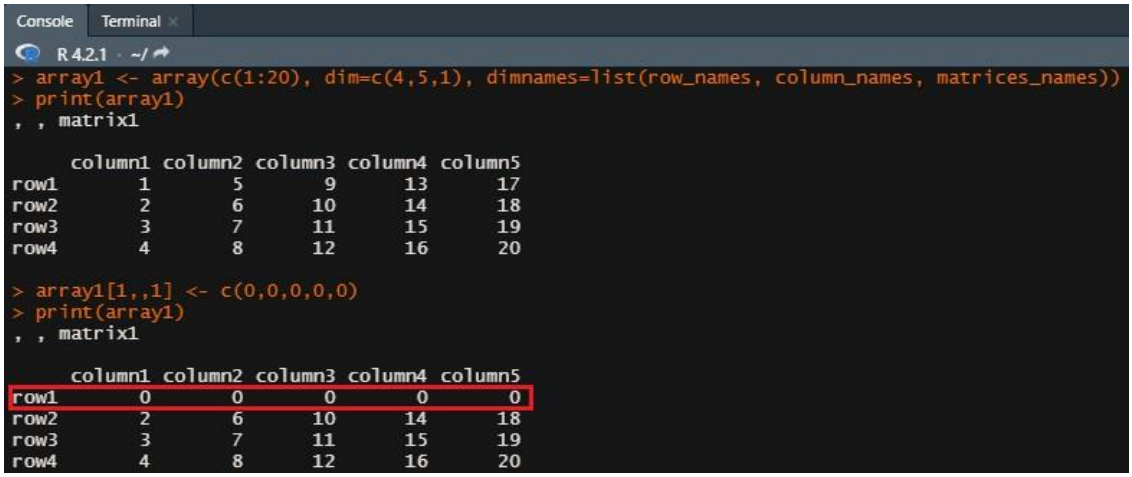
#### 2.2.5.3 Τροποποίηση ενός Array

Ο πιο κοινός τρόπος στην γλώσσα προγραμματισμού R ώστε να τροποποιηθεί ένας πίνακας πολλαπλών διαστάσεων ή αλλιώς ένα array, είναι ο ίδιος τρόπος με τον οποίο συμπεριφέρεται ο χρήστης και στις υπόλοιπες δομές.

Έστω ότι χρήστης θέλει να τροποποιήσει το array1 που δημιουργήθηκε παραπάνω. Η τροποποίηση που επιθυμεί να πραγματοποιήσει, είναι πάνω στην πρώτη γραμμή του πίνακα. Πιο συγκεκριμένα, πρόκειται να μηδενίσει όλες τις τιμές της μεταβλητής “row1”. Για να το επιτύχει θα χρησιμοποιήσει μία αγκύλη της μορφής:

- array1[1,,1]

όπου ο πρώτος άσσος αντιστοιχεί στον αριθμό της γραμμής, ο τελευταίος στον αριθμό του matrix που έτσι και αλλιώς υπάρχει μόνο ένα, ενώ η δεύτερη θέση παραμένει κενή διότι καμία στήλη δεν σκοπεύει να υποστεί τροποποίηση από τον προγραμματιστή. Τέλος, στο array ανατίθεται ένα διάνυσμα c() με όλες τις μηδενικές τιμές όπως προβάλεται και στην εικόνα 63.




*Εικόνα 63: Στιγμιότυπο τροποποίησης array.*

### 2.2.6 Παράγοντες (Factors)

Όσον αφορά τους παράγοντες είναι ένας ολοκληρωτικά διαφορετικός τύπος από όλους όσους προαναφέρθηκαν. Ο λόγος για τον οποίο συμβαίνει αυτό, είναι διότι αρκετές είναι οι φορές που στον τομέα της ανάλυσης δεδομένων, οι μεταβλητές αντί για ποσοτικές είναι ποιοτικές/κατηγορικές. Ως ποσοτικές μεταβλητές ορίζονται οι μεταβλητές που επιτρέπεται να πάρουν οποιαδήποτε συγκεκριμένη τιμή από αυτές που αναλύθηκαν παραπάνω (αριθμητικές, ακέραιες, χαρακτήρες). Παραδείγματα ποσοτικών μεταβλητών μπορούν να θεωρηθούν:

- ονόματα,
- η ηλικία ενός ανθρώπου,
- το άθροισμα δύο ή περισσότερων αριθμών κλπ.

Αντιθέτως, ως κατηγορικές ή ποιοτικές ορίζονται οι μεταβλητές οι οποίες μπορούν να λάβουν μια πιθανή τιμή από ένα συγκεκριμένο σύνολο στο οποίο ανήκουν. Οι τιμές μπορούν να είναι λογικές, χαρακτήρες ή και αριθμοί. Τα σύνολα

των πιθανών αυτών τιμών ονομάζονται επίπεδα. Οι παράγοντες στην γλώσσα προγραμματισμού R μπορούν να έχουν διαφορετικό τύπο, όμως οι τιμές που επιτρέπεται να πάρουν είναι αναγκαίο να βρίσκονται στα επίπεδα αυτά. Λόγου χάρη, αν ο χρήστης εισάγει μια τιμή η οποία δεν περιλαμβάνεται στα επίπεδα ενός παράγοντα, ο παράγοντας θα πάρει την τιμή NA δηλαδή Not Available. Πιο συγκεκριμένα, παραδείγματα ποσοτικών/κατηγορικών μεταβλητών αποτελούν:

- το φύλο ενός ανθρώπου (Άντρας ή Γυναίκα),
- λογικές Τιμές (True ή False),
- το μέγεθος ενός ρούχου (S, M, L ή XL) κλπ.

#### 2.2.6.1 ΔημΙΟυργία Παραγόντων

Προκειμένου να κατασκευαστεί ένας παράγοντας γίνεται χρήση μιας συνάρτησης με όνομα factor(). Η σύνταξη της δεν διαφέρει καθόλου από αυτή του matrix ή του array διότι και εδώ υπάρχουν ορισμένα arguments. Η κύρια μορφή στην οποία στηρίζεται η σύνταξη της είναι:

- factor(x=character(), levels, labels, exclude, ordered, nmax)

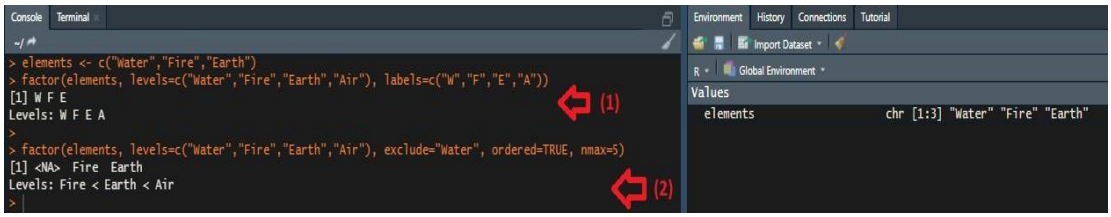
με τα arguments να ερμηνεύονται ως:

- x: Τα στοιχεία δεδομένων του παράγοντα.
- levels: Αποτελεί ένα διάνυσμα το οποίο συμπεριλαμβάνει τις πιθανές τιμές που είναι δυνατό να λάβει ο παράγοντας. Η χρήση του είναι προαιρετική.
- labels: Διάνυσμα εξίσου μη υποχρεωτικό που όταν χρησιμοποιείται περιέχει ορισμένα ονόματα τα οποία αντιστοιχούν στα επίπεδα του παράγοντα.
- exclude: Μια συλλογή τιμών εκτός επιπέδων.
- ordered: Μεταβλητή η οποία προσδιορίζει αν ο παράγοντας θεωρείται ή όχι ταξινομημένος. Λαμβάνει λογικές τιμές.
- nmax: Οριοθετεί τον αριθμό των επιπέδων.

Έστω ότι ο χρήστης θέλει να δημιουργήσει έναν παράγοντα, ο οποίος θα περιέχει ως κατηγορικές μεταβλητές στοιχεία της φύσης. Αρχικά, θα πρέπει να κατασκευαστεί ένα διάνυσμα με όνομα “elements” που ως πιθανές τιμές θα συμπεριλαμβάνει τα στοιχεία “Water”, “Fire” και “Earth”. Τα επίπεδα του παράγοντα θα είναι “Water”,

“Fire”, “Earth” και “Air”. Γίνεται έτσι αντιληπτό, πως από τις αρχικές τιμές παραλείπεται το τέταρτο στοιχείο, δηλαδή “Air”, χωρίς όμως να παραλείπεται ταυτόχρονα και από τα επίπεδα. Ως labels για τα συγκεκριμένα επίπεδα δίνονται προαιρετικά οι τιμές “W”, “F”, “E” και “A” αντίστοιχα. Δημιουργώντας και εκτελώντας την συνάρτηση: factor(), τυπώνεται μέχρι στιγμής το παρακάτω αποτέλεσμα (βλ. εικόνα 64 (1)).

Εν συνεχεία, ο χρήστης θα χρησιμοποιήσει την μεταβλητή exclude ώστε να παραλείψει κάποια τιμή από τα παραπάνω επίπεδα. Για παράδειγμα, γίνεται παράληψη της μεταβλητής του νερού, δηλαδή “Water”. Τέλος, θέτει την μεταβλητή ordered ίση με την τιμή αληθείας TRUE ώστε ο παράγοντας να είναι ταξινομημένος και ως ανώτατη τιμή των επιπέδων ορίζει nmax=5. Στην εικόνα 64 (2) προβάλετε το τελικό αποτέλεσμα. Κάτι που αξίζει να παρατηρηθεί είναι πως εφόσον η μεταβλητή “Water” θεωρείται πλέον μη διαθέσιμη λόγω παράλειψης από τα επίπεδα του παράγοντα, στην θέση εμφανίζεται η τιμή NA. (βλ. εικόνα 64 (2)).

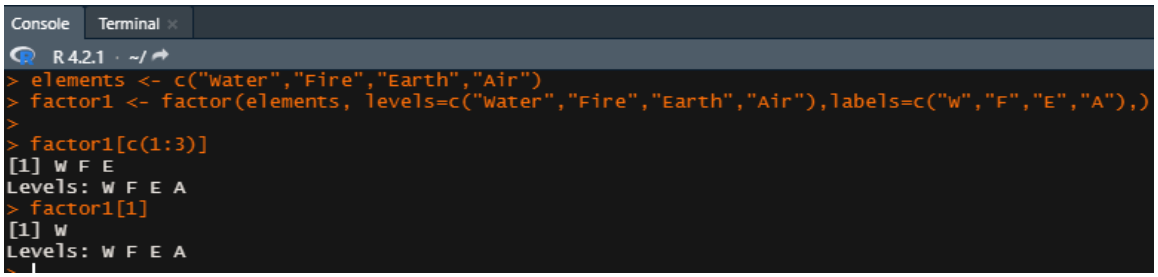



*Εικόνα 64: Στιγμιότυπο δημιουργίας παράγοντα με χρήση της συνάρτησης factor().*

#### 2.2.6.2 Αντιστοίχιση Στοιχείων ενός Παράγοντα

Άλλη μια διαδικασία παρόμοια με τις υπόλοιπες δομές, είναι η αντιστοίχιση των παραγόντων. Γίνεται χρήση ακριβώς της ίδιας τεχνικής ώστε ο χρήστης να αποκτήσει πρόσβαση στα στοιχεία των επιπέδων του παράγοντα. Και εδώ, ισχύουν οι τιμές της μορφής αρνητικών και θετικών ακέραιων, λογικών ή χαρακτήρων. Με βάση το παραπάνω παράδειγμα με τα στοιχεία της φύσης:

Έστω ότι ο χρήστης θέλει να αποκτήσει πρόσβαση στα 3 πρώτα στοιχεία του παράγοντα με όνομα factor1. Για να το επιτύχει αυτό, θα χρησιμοποιήσει μέσα σε αγκύλες, την συνάρτηση c(1:3) ώστε να εμγανίσει τα στοιχεία νερού, φωτιάς και αέρα. Αντίστοιχα, αν θέλει να εμφανίσει μόνο το 1$^{ο}$ στοιχείο, θα γράψει μέσα στις αγκύλες: [1] (βλ. εικόνα 65).



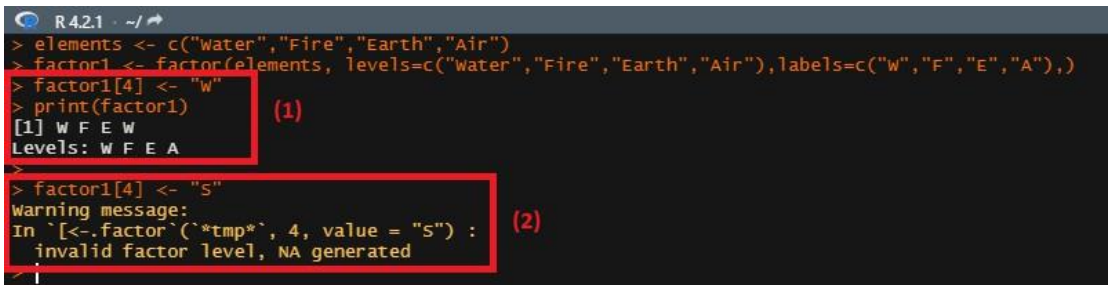

*Εικόνα 65: Στιγμιότυπο αντιστοίχισης παραγόντων.*

### 2.2.6.3 Τροποποίηση ενός Παράγοντα

Η τροποποίηση ενός παράγοντα είναι η διαδικασία κατά την οποία ο χρήστης μέσω της διαδικασίας αντιστοίχισης αλλάζει τις ήδη υπάρχουσες τιμές με καινούργιες. Οι νέες αυτές τιμές, είναι απαραίτητο να έχουν δηλωθεί μέσα στα επίπεδα του παράγοντα. Διαφορετικά, θα εμφανιστεί στην οθόνη του προγράμματος μη έγκυρη εισαγωγή τιμής.

Έστω, για παράδειγμα, ότι ο προγραμματιστής επιθυμεί να τροποποιήσει το τελευταίο στοιχείο δηλαδή εκείνο του αέρα. Αρχικά, θα χρειαστεί να εισάγει σε αγκύλες τον αριθμό της θέσης δήλωσης του στοιχείου, στην προκειμένη περίπτωση,
4. Στην συνέχεια, θα εισάγει την νέα τιμή από αυτές που υπάρχουν στα επίπεδα. Επιλέγει να εισάγει στην θέση του αέρα, το στοιχείο του νερού, οπότε δηλώνει την τιμή: “W”. Με αυτόν τον τρόπο, το αποτέλεσμα αλλάζει και διακρίνεται στην εικόνα 66(1). Σε άλλη περίπτωση:

Έστω πως στην θέση του 4ου στοιχείου θέλει να δηλώσει ένα καινούργιο που δεν βρίσκεται στα επίπεδα του factor. Για περάδειγμα, το στοιχείο “S”, δηλαδή stone. Ακολουθώντας τα ίδια βήματα, δεν θα μπορέσει να πραγνατοποιηθεί η τροποποποίηση και το μήνυμα που εμφανίζεται φαίνεται στην εικόνα 66(2).



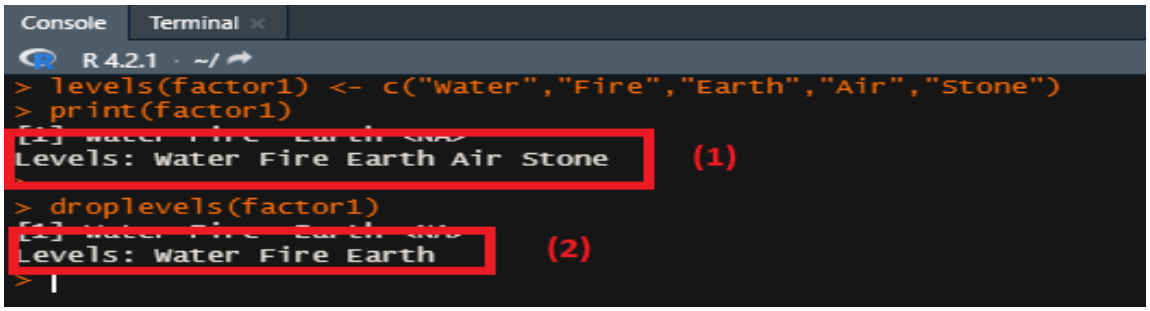

*Εικόνα 66: Στιγμιότυπο τροποποίησης παράγοντα.*

#### 2.2.6.4 Προσθήκη & Διαγραφή Επιπέδων ενός Παράγοντα

Όπως προανφέρθηκε στην προηγούμενη ενότητα, ο χρήστης δεν έχει την δυνατότητα να τροποποιήσει μια τιμή με μια άλλη, η οποία δεν βρίσκεται στα επίπεδα. Για να γίνει προσθήκη μιας τέτοιας τιμής, θα πρέπει να γίνει πρώτα προσθήκη νέου/ων επιπέδου/ων στον παράγοντα. Αν για παράδειγμα, ο χρήστης θέλει να αντικαταστήσει/προσθέσει το στοιχείο "Stone", δηλαδή πέτρα, του οποίου έγινε προσπάθεια προσθήκης πριν, θα χρησιμοποιήσει τον κώδικα της εικόνας 67(1), ώστε να κατασκευάσει αρχικά το αντίστοιχο επίπεδο. Τέλος, σε περίπτωση που θέλει να διαγράψει κάποιο ή ακόμη και κάποια από τα ήδη υπάρχοντα επίπεδα του παράγοντα, το μόνο που έχει να κάνει είναι να κάνει χρήση της συνάρτησης:

- droplevels().

Η διαδικασία αυτή διακρίνεται στην εικόνα 67(2) όπου διαγράφονται στα δύο τελευταία επίπεδα, δηλαδή του αέρα και της πέτρας.




*Εικόνα 67: Στιγμιότυπο προσθήκης και διαγραφής επιπέδων ενός παράγοντα.*

# 3. Στατιστική Συμπερασματολογία

## 3.1 Εισαγωγή στην Στατιστική Συμπερασματολογία

### 3.1.1 Ορισμός Στατιστικής Συμπερασματολογίας

Ένας από τους σημαντικότερους τομείς του κλάδου των μαθηματικών, αποτελεί η επιστήμη της στατιστικής. Η έννοια «στατιστική», αναφέρεται στο κομμάτι της συλλογής, ανάλυσης και ερμηνείας αριθμητικών ή μη δεδομένων. Με απλά λόγια, μπορεί κανείς να πεί, πως αποτελεί μια ομάδα ποσοτικών δεδομένων, τα οποία εξάγωνται δειγματοληπτικά, με σκοπό να εκτιμηθούν τα χαρακτηριστικά ενός πληθυσμού από τον οποίο ελήφθη το δείγμα. Διακρίνεται σε δύο επιμέρους τμήματα:

1. Περιγραφική Στατιστική, η οποία όπως υποδηλώνει και το όνομα της, χρησιμεύει στην περιγραφή των δεδομένων και
2. Συμπερασματική Στατιστική, της οποίας η χρήση είναι να καταλήγει σε ένα συμπέρασμα για έναν πληθυσμό από τον οποίο έχει συλλεχθεί ένα συγκεκριμένο δείγμα.

Έτσι, ο όρος στατιστική συμπερασματολογία αποτελεί μια διαδικασία κατά την οποία είναι εφικτή η λήψη αποφάσεων σχετικά με τις παραμέτρους ενός πληθυσμού οι οποίες βασίζονται σε τυχαία δείγματα αλλά και το πόσο βέβαια είναι τα δείγματα αυτά. Η στατιστική συμπερασματολογία εξετάζει τα δεδομένα με μεγαλύτερη αποτελεσματικότητα και ακρίβεια σε σχέση με άλλες μεθόδους παρόμοιας φιλοσοφίας. Από τους πιο σημαντικούς τύπους με τους οποίους πραγματοποιείται η εξαγωγή των συμπερασμάτων αποτελούν: διαστήματα εμπιστοσύνης, έλεγχος υποθέσεων ενός δείγματος, συσχέτιση Pearson, t-test ή ANOVA κλπ. Ακόμη, εφαρμόζεται σε διάφορους τομείς όπως για παράδειγμα: στην τεχνητή νοημοσύνη, την μηχανική μάθηση, σε επιχειρηματικές και χρηματοοικονομικές αναλύσεις, σε φαρμακευτικούς τομείς καθώς και για εντοπισμό απάτης. Η εκτίμηση των δεδομένων αυτών εξαρτάται από τρεις συγκεκριμένες μορφές:

- εκτίμηση σημείου,
- εκτίμηση διαστήματος και
- έλεγχος υποθέσεων.

Επίσης, για να δημιουργηθεί ένα στατιστικό συμπέρασμα απαίτουνται άλλα δύο χαρακτηριστικά:

1. να υπάρχει ένα δείγμα μεταβλητής τιμής και
2. το μέγεθος του δείγματος.

## 3.2 Περιγραφική Στατιστική

Όπως προαναφέρθηκε και στην υποενότητα 3.1.1, η περιγραφική στατιστική ασχολείται με την περιγραφή και την περιληπτική παρουσίαση ενός συνόλου δεδομένων. Αξίζει να σημειωθεί, πως η Περιγραφική Στατιστική κάνει χρήση διαφόρων αριθμητικών τεχνικών με σκοπό την περιγραφή ορισμένων χαρακτηριστικών των δεδομένων, όπως είναι: η διακύμανση, η μέση τιμή, η διάμετρος αλλά και η διαγραμματική απεικόνιση. Βέβαια, η περιγραφή αυτών των χαρακτηριστικών, δεν είναι δυνατό να γενικευτεί για όλον το πληθυσμό. Περιλαμβάνει τις εξής τεχνικές:

- ✓ Γραφικές Παραστάσεις,
- ✓ Στατιστικούς Πίνακες και
- ✓ Αριθμητικά Περιγραφικά Μέτρα.

Ένα από τα κομμάτια που απασχολεί περισσότερο, είναι αυτή των γραφημάτων. Τα γραφήματα διευκολύνουν τον χρήστη να έρθει στο συμπέρασμα αν ισχύουν ή όχι οι υποθέσεις των στατιστικών μοντέλων. Οι πιο σημαντικοί τύποι γραφημάτων περιλαμβάνουν:

- QQ plots (quantile-quantile plots),
- Ιστογράμματα (histograms),
- Γραφήματα Πυκνότητας (density plots) αλλά και
- Κυτιογράματα (boxplots).

## 3.3 Έλεγχος Υποθέσεων

Ένα από τα κυριότερα θέματα του τομέα της Στατιστικής Συμπερασματολογίας είναι ο Έλεγχος Υποθέσεων και πιο συγεκεριμένα ο Έλεγχος Στατιστικών Υποθέσεων. Με την έννοια της λέξης «υπόθεση» ορίζουμε μια άποψη ή ένα ερευνητικό ερώτημα του οποίου η αξιολόγηση χρήζει περαιτέρω διερεύνησης. Ο

έλεγχος υποθέσεων και οι διαδικασίες που αυτός περιλαμβάνει μας επιτρέπει να καθορίσουμε κατά πόσο τέτοιες ερευνητικές υποθέσεις ικανοποιούνται με τα δεδομένα της κάθε έρευνας. Οι ερευνητικές αυτές υποθέσεις, όταν διατυπωθούν, οδηγούν στις στατιστικές υποθέσεις που ελέγχονται με μαθηματικές δαδικασίες ως προς την ορθότητα τους. Στο μεγαλύτερο κομμάτι των ελέγχων υποθέσεων, χρησιμοποιείται η διαδικασία της μέσης τιμής ενός ή δύο δειγμάτων καθώς και οι ελέγχοι καλής προσαρμογής και ανεξαρτησίας, οι οποίοι αναλύονται και παρακάτω. Το σημείο που διαφοροποεί τους δύο αυτούς ελέγχους, είναι πως ο έλεγχος καλής προσαρμογής εφαρμόζεται για ελέγχους μίας ποιοτικής μεταβλητής ενώ ο έλεγχος ανεξαρτησίας για δύο ποιοτικές μεταβλητές.

Πιο συγκεριμένα όμως τώρα, ο έλεγχος υποθέσεων είναι μια στατιστική διαδικασία που χρησιμοποιείται για να αποφασιστεί αν υπάρχουν αρκετά στοιχεία σε ένα δείγμα δεδομένων για να υποστηρίξουν μια συγκεκριμένη υπόθεση σχετικά με τον πληθυσμό από τον οποίο προέρχεται το δείγμα. Στην R, ο έλεγχος υποθέσεων μπορεί να πραγματοποιηθεί χρησιμοποιώντας διάφορες εντολές και πακέτα, ανάλογα με τον τύπο του ελέγχου που θέλουμε να πραγματοποιήσουμε π.χ.

- t-test, χρησιμοποιείται για να συγκρίνει τις μέσες τιμές δύο ομάδων και να αποφασιστεί αν οι διαφορές μεταξύ τους είναι στατιστικά σημαντικές.
- x^2-test, χρησιμοποιείται για να εξετάσει τις σχέσεις μεταξύ κατηγορηματικών δεδομένων.
- ANOVA, η ανάλυση διακύμανσης δηλαδή, χρησιμοποιείται για να συγκρίνει τις μέσες τιμές μεταξύ τριών ή περισσότερων ομάδων.
- z-test, χρησιμοποιείται όταν έχουμε ένα μεγάλο δείγμα και θέλουμε να συγκρίνουμε μια μέση τιμή με μια γνωστή τιμή.

**Βήματα για τον έλεγχο υποθέσεων:**

- Διαμόρφωση της μηδενικής υπόθεσης (H0) και της εναλλακτικής υπόθεσης (H1).
- Επιλογή κατάλληλου τεστ ανάλογα με τα δεδομένα και την υπόθεση.
- Υπολογισμός της στατιστικής τιμής (π.χ. t-value, z-value).
- Εύρεση της p-τιμής που θα καθορίσει αν θα απορρίψουμε ή όχι τη μηδενική υπόθεση.
- Λήψη απόφασης με βάση το επίπεδο σημαντικότητας (συνήθως 0.05).

Στην συνέχεια της εργασίας θα ακολουθήσουν πιο αναλυτικά παραδείγματα ελέγχου υποθέσεων όπως ο έλεγχος μέσης τιμής, διαφορά μεταξύ δύο τιμών κλπ. με σκοπό να υπάρχει μεγαλύτερη επεξήγηση.

## 3.4 Έλεγχος Μέσης Τιμής, Ποσοστού, Διαφοράς Μέσων Τιμών και Διαστήματα Εμπιστοσύνης

Ο έλεγχος μέσης τιμής στην γλώσσα R, γνωστός και ως one-sample t-test, είναι μια στατιστική διαδικασία που χρησιμοποιείται για να προσδιορίσει αν η μέση τιμή ενός δείγματος διαφέρει σημαντικά από μια προκαθορισμένη τιμή ή υπόθεση, γνωστή ως υποθετική μέση τιμή (μ0). Αυτή η προκαθορισμένη τιμή μπορεί να είναι, για παράδειγμα, μια αναμενόμενη τιμή βασισμένη σε προηγούμενα δεδομένα ή μια θεωρητική τιμή που θέλουμε να ελέγξουμε αν ισχύει στον πληθυσμό από τον οποίο προέρχεται το δείγμα. Ο σκοπός του ελέγχου μέσης τιμής είναι να διαπιστώσει αν η παρατηρούμενη μέση τιμή ενός δείγματος είναι συμβατή με μια προκαθορισμένη τιμή ή αν αποκλίνει σημαντικά από αυτήν. Αυτός ο έλεγχος είναι ιδιαίτερα χρήσιμος όταν θέλουμε να ελέγξουμε υποθέσεις σχετικά με τη μέση τιμή ενός πληθυσμού, χρησιμοποιώντας τα δεδομένα ενός δείγματος.

**Βήματα του Ελέγχου:**

1. Καθορισμός των Υποθέσεων:
    - Μηδενική υπόθεση (H0): Η μέση τιμή του πληθυσμού (μ) είναι ίση με την υποθετική τιμή (μ0).
    - Εναλλακτική υπόθεση (H1): Η μέση τιμή του πληθυσμού (μ) είναι διαφορετική από την υποθετική τιμή (μ0).
2. Υπολογισμός του Στατιστικού t:

Το στατιστικό t μετρά την απόσταση μεταξύ της μέσης τιμής του δείγματος και της υποθετικής μέσης τιμής, εκφρασμένη σε μονάδες τυπικής απόκλισης. Υπολογίζεται ως:

$$t = \frac{\bar{x} - \mu_0}{\frac{s}{\sqrt{n}}}$$

όπου $\bar{x}$ είναι η μέση τιμή του δείγματος, s είναι η τυπική απόκλιση του δείγματος, και n το μέγεθος του δείγματος.

3. Υπολογισμός του p-value:

Το p-value είναι η πιθανότητα να παρατηρήσουμε τα δεδομένα μας (ή κάτι πιο ακραίο) αν η μηδενική υπόθεση (H0) είναι αληθής. Με άλλα λόγια, το p-value μετράει πόσο πιθανό είναι τα αποτελέσματα που βλέπουμε να έχουν προκύψει τυχαία αν δεν υπάρχει πραγματική διαφορά ή επίδραση. Η ερμηνεία του είναι η εξής:

- μικρό p-value (συνήθως $\alpha < 0.05$): Υπάρχουν ισχυρά στοιχεία υπέρ της απόρριψης της μηδενικής υπόθεσης. Αυτό σημαίνει ότι τα αποτελέσματα που παρατηρήσαμε είναι πολύ απίθανο να έχουν συμβεί τυχαία αν η μηδενική υπόθεση είναι αληθής.
- μεγάλο p-value (συνήθως $\alpha \geq 0.05$): Δεν υπάρχουν επαρκή στοιχεία για την απόρριψη της μηδενικής υπόθεσης. Αυτό σημαίνει ότι τα δεδομένα δεν παρέχουν ισχυρή ένδειξη ότι η μηδενική υπόθεση είναι ψευδής.

Όπου α, ένα προκαθορισμένο επίπεδο σημαντικότητας, το οποίο είναι συνήθως 0.05. Το επίπεδο σημαντικότητας είναι το όριο που θέτουμε για να αποφασίσουμε αν το p-value είναι αρκετά μικρό ώστε να απορρίψουμε τη μηδενική υπόθεση.

- Αν το p-value < α: Απορρίπτουμε τη μηδενική υπόθεση.
- Αν το p-value ≥ α: Δεν απορρίπτουμε τη μηδενική υπόθεση.

4. Λήψη Απόφασης:

Αν το p-value είναι μικρότερο από το προκαθορισμένο επίπεδο σημαντικότητας (συνήθως $\alpha = 0.05$), απορρίπτουμε τη μηδενική υπόθεση. Διαφορετικά, δεν απορρίπτουμε τη μηδενική υπόθεση. Σημαντικό είναι να κατανοήσουμε πως το p- value **δεν** είναι η πιθανότητα η μηδενική υπόθεση να είναι αληθής. Αντίθετα, είναι η πιθανότητα να παρατηρήσουμε τα δεδομένα μας αν η μηδενική υπόθεση ήταν αληθής. Ένα μικρό p-value υποδηλώνει στατιστική σημαντικότητα, αλλά δεν εγγυάται ότι το αποτέλεσμα είναι σημαντικό στην πραγματική ζωή (πρακτική σημαντικότητα). Το p-value είναι επομένως ένα εργαλείο που βοηθάει στη λήψη

αποφάσεων για το αν θα απορρίψουμε ή όχι τη μηδενική υπόθεση σε έναν στατιστικό έλεγχο.

5. Διάστημα Εμπιστοσύνης:

Τα διαστήματα εμπιστοσύνης (confidence intervals) είναι ένα σημαντικό εργαλείο στη στατιστική ανάλυση, καθώς παρέχουν ένα εύρος τιμών μέσα στο οποίο αναμένεται να βρίσκεται η πραγματική τιμή μιας παραμέτρου του πληθυσμού, με ένα συγκεκριμένο επίπεδο εμπιστοσύνης. Στην R, τα διαστήματα εμπιστοσύνης υπολογίζονται συχνά κατά τη διάρκεια στατιστικών ελέγχων, όπως ο έλεγχος μέσης τιμής. Πιο συγκεκριμένα, Ένα διάστημα εμπιστοσύνης είναι ένα εύρος τιμών που έχει έναν συγκεκριμένο βαθμό βεβαιότητας (συνήθως 95%) ότι περιλαμβάνει την πραγματική τιμή της παραμέτρου του πληθυσμού, όπως η μέση τιμή, το ποσοστό ή η διακύμανση.

- Ένα 95% διάστημα εμπιστοσύνης σημαίνει ότι, αν παίρναμε πολλά δείγματα και υπολογίζαμε το διάστημα εμπιστοσύνης για το καθένα, το 95% αυτών των διαστημάτων θα περιείχε την πραγματική τιμή της παραμέτρου.
- Το διάστημα εμπιστοσύνης εξαρτάται από το μέγεθος του δείγματος και την διακύμανση των δεδομένων: μεγαλύτερα δείγματα και μικρότερη διακύμανση οδηγούν σε στενότερα (πιο ακριβή) διαστήματα εμπιστοσύνης.

**Παράδειγμα με χρήση της R:**

Υποθέτουμε ότι ένας καθηγητής πανεπιστημίου, θέτει την μέση βαθμολογία των φοιτητών του σε ένα συγκεκριμένο διαγώνισμα προόδου ίση με 80. Μετά το πέρας της εξέτασης έχει τις βαθμολογίες από ένα δείγμα 10 φοιτητών και θέλει να ελέγξει αν η μέση βαθμολογία του δείγματος διαφέρει σημαντικά από το 80. Οι βαθμολογίες είναι ίσες με: 85, 77, 93, 89, 75, 81, 78, 76, 84, 81.

Ως βήμα πρώτο, δημιουργούμε τα δεδομένα της R εισάγωντας τις βαθμολογίες και έπειτα γίνεται εκτέλεση του one-sample t-test όπου χρησιμοποιούμε τη συνάρτηση t.test() για να ελέγξουμε αν η μέση τιμή του δείγματος διαφέρει από την υποθετική τιμή των 80. Όταν εκτελέσουμε τον παραπάνω κώδικα, η R θα επιστρέψει τα αποτελέσματα του ελέγχου.

```
> grades_of_students <- c(85,77,93,89,75,81,78,76,84,81)
>
> result_of_test <- t.test(grades_of_students, mu=80)
>
> print(result_of_test)

        One Sample t-test

data:  grades_of_students
t = 1.0223, df = 9, p-value = 0.3333
alternative hypothesis: true mean is not equal to 80
95 percent confidence interval:
 77.69552 86.10448
sample estimates:
mean of x
     81.9
```

*Εικόνα 68: Παράδειγμα ελέγχου μέσης τιμής με χρήση της γλώσσας R.*

**Ανάλυση των Αποτελεσμάτων:**

- στατιστικό t: Το t-value είναι 1.0223, το οποίο δείχνει τη διαφορά μεταξύ της μέσης τιμής του δείγματος και της υποθετικής μέσης τιμής, σε μονάδες τυπικής απόκλισης.
- p-value: Το p-value είναι 0.3333. Επειδή το p-value είναι μεγαλύτερο από το επίπεδο σημαντικότητας 0.05, δεν έχουμε επαρκή στοιχεία για να απορρίψουμε τη μηδενική υπόθεση. Δηλαδή, δεν μπορούμε να πούμε ότι η μέση τιμή των βαθμολογιών του δείγματος διαφέρει σημαντικά από 80.
- διάστημα εμπιστοσύνης (95%): Το διάστημα εμπιστοσύνης είναι από 77.69 έως 86.10. Αυτό σημαίνει ότι, με 95% εμπιστοσύνη, η πραγματική μέση τιμή του πληθυσμού θα βρίσκεται εντός αυτού του εύρους.
- μέση τιμή του δείγματος: Η μέση τιμή του δείγματος είναι 81.9.

Καταλήγουμε στο συμπέρασμα πως, με βάση το p-value που υπολογίστηκε, δεν υπάρχει στατιστικά σημαντική διαφορά μεταξύ της μέσης τιμής του δείγματος και της υποθετικής τιμής των 80. Άρα, δεν έχουμε επαρκή στοιχεία για να απορρίψουμε τη μηδενική υπόθεση ότι η μέση βαθμολογία είναι 80.

Ο έλεγχος ποσοστού, γνωστός ως one-sample proportion test, είναι μια στατιστική μέθοδος που χρησιμοποιείται για να ελέγξει αν το παρατηρούμενο ποσοστό επιτυχιών σε ένα δείγμα διαφέρει σημαντικά από ένα προκαθορισμένο ποσοστό στον πληθυσμό. Αυτή η μέθοδος είναι ιδιαίτερα χρήσιμη όταν θέλουμε να συγκρίνουμε το ποσοστό ενός δείγματος με μια υποθετική τιμή για να δούμε αν υπάρχει στατιστικά σημαντική

διαφορά. Ο κύριος σκοπός του ελέγχου ποσοστού είναι να εξετάσει αν το παρατηρούμενο ποσοστό επιτυχιών (ή αποτυχιών) σε ένα δείγμα ανταποκρίνεται σε ένα συγκεκριμένο ποσοστό που έχει οριστεί ως υπόθεση. Για παράδειγμα, μπορεί να θέλουμε να ελέγξουμε αν το ποσοστό επιτυχίας ενός διαγωνίσματος σε μια εξεταστική είναι 70%, βασισμένοι σε δεδομένα που συλλέξαμε από ένα δείγμα φοιτητών. Οι βασικές υποθέσεις είναι:

- Μηδενική υπόθεση (H0): Το παρατηρούμενο ποσοστό στο δείγμα είναι ίσο με το υποθετικό ποσοστό ($p=p0$).
- Εναλλακτική υπόθεση (H1): Το παρατηρούμενο ποσοστό στο δείγμα είναι διαφορετικό από το υποθετικό ποσοστό ($p\neq p0$).

**Βήματα του ελέγχου:** Όπως και στον έλεγχο μέσης τιμής

1. Συλλογή Δεδομένων: Συλλέγουμε δεδομένα από το δείγμα μας και καταγράφουμε τον αριθμό των επιτυχιών (ή αποτυχιών).
2. Υπολογισμός του Στατιστικού: Υπολογίζουμε ένα στατιστικό που θα μας επιτρέψει να συγκρίνουμε το παρατηρούμενο ποσοστό με το υποθετικό ποσοστό. Ο πιο κοινός τρόπος για να γίνει αυτό είναι με τη χρήση του στατιστικού x^2.
3. Υπολογισμός του p-value: Το p-value μας λέει πόσο πιθανό είναι να παρατηρήσουμε το δεδομένο ποσοστό ή ένα ποσοστό πιο ακραίο αν η μηδενική υπόθεση είναι αληθής.
4. Λήψη Απόφασης: Συγκρίνουμε το p-value με το προκαθορισμένο επίπεδο σημαντικότητας (συνήθως 0.05) για να αποφασίσουμε αν θα απορρίψουμε ή όχι τη μηδενική υπόθεση.

**Παράδειγμα με χρήση της R:**

Υποθέτουμε ότι μια σχολή ισχυρίζεται ότι το 60% των φοιτητών της έχουν πετύχει στην τελευταία εξεταστική περίοδο. Για να ελέγξουμε αυτόν τον ισχυρισμό, συλλέγουμε δεδομένα από ένα δείγμα 140 φοιτητών και διαπιστώνουμε ότι 80 από αυτούς δήλωσαν πως πέρασαν.

Ως βήμα πρώτο, δημιουργούμε τα δεδομένα της R με τον αριθμό των πετυχημένων φοιτητών και έπειτα πραγματοποείαται εκτέλεση του ελέγχου ποσοστού, κάνοντας χρήση της συνάρτηση prop.test() για να ελέγξουμε αν το παρατηρούμενο ποσοστό των επιτυχημεώνων φοιτητών διαφέρει σημαντικά από το υποθετικό ποσοστό του 60%. Όταν εκτελεστεί ο παραπάνω κώδικας, η R θα εμφανίσει τα αποτελέσματα του ελέγχου.

```
> qualified_students <- 80
>
> n <- 140
>
> p0 <- 0.60
>
> test_result <- prop.test(qualified_students, n, p=p0)
>
> print(test_result)

        1-sample proportions test with continuity correction

data:  qualified_students out of n, null probability p0
X-squared = 0.36458, df = 1, p-value = 0.546
alternative hypothesis: true p is not equal to 0.6
95 percent confidence interval:
 0.4850705 0.6538076
sample estimates:
        p
0.5714286
```

*Εικόνα 69: Παράδειγμα ελέγχου ποσοστού με χρήση της γλώσσας R.*

**Ανάλυση των Αποτελεσμάτων:**

- p-value: Το p-value είναι 0.546, το οποίο δεν είναι πολύ κοντά στο επίπεδο σημαντικότητας 0.05. Αυτό σημαίνει ότι δεν είμαστε πολύ κοντά στο να απορρίψουμε τη μηδενική υπόθεση.
- x^2: Το στατιστικό x^2 δείχνει την απόσταση μεταξύ του παρατηρούμενου ποσοστού και του υποθετικού ποσοστού. Αν η διαφορά αυτή είναι μεγάλη, το x^2 θα είναι υψηλό, και πιθανότατα θα απορρίψουμε τη μηδενική υπόθεση.
- διάστημα εμπιστοσύνης: Το διάστημα εμπιστοσύνης (95%) κυμαίνεται από 48.5% έως 65.3%. Αυτό σημαίνει ότι είμαστε 95% σίγουροι πως το πραγματικό ποσοστό των ικανοποιημένων φοιτητών βρίσκεται εντός αυτού του εύρους.

Καταλήγουμε στο συμπέρασμα πως με βάση το p-value που είναι μεγαλύτερο από την τιμή του 0.05, μπορούμε να πούμε ότι δεν υπάρχει ισχυρή στατιστική ένδειξη ότι το ποσοστό των επιτυχημένων φοιτητών διαφέρει σημαντικά από το 60%.

Τέλος, ο έλεγχος για την διαφορά μέσων τιμών, (γνωστός και ως two-sample t-test ή independent t-test) είναι μια στατιστική μέθοδος που χρησιμοποιείται για να συγκρίνει τις μέσες τιμές δύο ανεξάρτητων δειγμάτων, με σκοπό να διαπιστωθεί αν υπάρχει στατιστικά σημαντική διαφορά μεταξύ τους. Ο σκοπός του ελέγχου για τη διαφορά μέσων τιμών είναι να εξεταστεί αν οι μέσες τιμές δύο ομάδων είναι ίδιες ή διαφορετικές. Για παράδειγμα, μπορεί να θέλουμε να συγκρίνουμε την απόδοση δύο διαφορετικών τμημάτων σε μια εξέταση.

**Βασικές υποθέσεις:**

1. Μηδενική υπόθεση **(**H0**)**: Οι μέσες τιμές των δύο πληθυσμών είναι ίδιες (μ1 =μ2).
2. Εναλλακτική υπόθεση **(**H1**)**: Οι μέσες τιμές των δύο πληθυσμών είναι διαφορετικές (μ1≠μ2).

**Προϋποθέσεις για τον έλεγχο:**

Πριν πραγματοποιηθεί ο έλεγχος, πρέπει να βεβαιωθούμε ότι πληρούνται ορισμένες προϋποθέσεις:

1. Ανεξαρτησία των δειγμάτων: Τα δύο δείγματα πρέπει να είναι ανεξάρτητα το ένα από το άλλο.
2. Κανονικότητα: Τα δεδομένα σε κάθε δείγμα πρέπει να ακολουθούν την κανονική κατανομή, ειδικά αν το μέγεθος του δείγματος είναι μικρό.
3. Ομοιογένεια διακυμάνσεων: Οι διακυμάνσεις των δύο πληθυσμών πρέπει να είναι ίδιες. Αυτό μπορεί να ελεγχθεί με τη βοήθεια του ελέγχου Levene.

**Βήματα του έλεγχου:**

- Στατιστικό t: Το t-value δείχνει την απόσταση μεταξύ των δύο μέσων τιμών, σε μονάδες τυπικής απόκλισης.
- p-value: Αν το p-value είναι μικρότερο από το επίπεδο σημαντικότητας (συνήθως 0.05), τότε υπάρχει στατιστικά σημαντική διαφορά μεταξύ των μέσων τιμών των δύο ομάδων.

- Διάστημα εμπιστοσύνης (95%): Το διάστημα εμπιστοσύνης δείχνει το εύρος τιμών εντός του οποίου είναι πιθανό να βρίσκεται η πραγματική διαφορά των μέσων τιμών.
- Μέσες τιμές των δύο ομάδων: Η R θα επιστρέψει επίσης τις εκτιμώμενες μέσες τιμές για κάθε ομάδα.

**Παράδειγμα με χρήση της R:**

Ας υποθέσουμε ότι θέλουμε να συγκρίνουμε τις μέσες τιμές των επιδόσεων δύο διαφορετικών ομάδων φοιτητών σε ένα διαγώνισμα. Ως αρχικό βήμα, δημιουργούμε τα δεδομένα της R, δηλαδή τις δύο ομάδες. Έπειτα, γίνεται εκτέλεση του two-sample t-test, με χρήση της συνάρτησης t.test() για να συγκρίνουμε τις μέσες τιμές των δύο ομάδων. Όταν εκτελέσουμε τον παραπάνω κώδικα, η R θα επιστρέψει τα εξής αποτελέσματα:

```
> teamA <- c(90,88,85,89,93,86,87,83,78,80)
>
> teamB <- c(85,82,78,84,92,77,88,79,76,81)
>
> teams_result <- t.test(teamA,teamB)
>
> print(teams_result)

        Welch Two Sample t-test

data:  teamA and teamB
t = 1.7037, df = 17.785, p-value = 0.1059
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.8667204  8.2667204
sample estimates:
mean of x mean of y
     85.9      82.2
```

*Εικόνα 70: Παράδειγμα ελέγχου για την διαφορά μέσων τιμών με χρήση της γλώσσας R.*

**Ανάλυση των Αποτελεσμάτων:**

- p-value: Το p-value είναι 0.1059, το οποίο είναι μεγαλύτερο από 0.05. Αυτό σημαίνει ότι δεν υπάρχει στατιστικά σημαντική διαφορά μεταξύ των μέσων τιμών των δύο ομάδων.
- Διάστημα εμπιστοσύνης: Το διάστημα εμπιστοσύνης (95%) κυμαίνεται από -0.866 έως 8.26, κάτι που σημαίνει ότι η πραγματική διαφορά μεταξύ των μέσων τιμών θα μπορούσε να είναι οποιαδήποτε τιμή εντός αυτού του εύρους.
- Μέσες τιμές: Οι μέσες τιμές των δύο ομάδων είναι 82.2 και 85.9 αντίστοιχα.

Καταλήγουμε στο συμπέρασμα πως με βάση τα αποτελέσματα, δεν έχουμε επαρκή στοιχεία για να υποστηρίξουμε ότι υπάρχει στατιστικά σημαντική διαφορά μεταξύ των μέσων τιμών των δύο ομάδων.

## 3.5 Έλεγχος Καλής Προσαρμογής

Στην συγκεκριμένη κατηγορία ελέγχου καλής προσαρμογής, υπάρχει ο έλεγχος υποθέσεων, (Kolmogorov-Smirnov test), κατά τον οποίον ελέγχουμε την μηδενική υπόθεση ότι δηλαδή τα δεδομένα ακολουθούν μια συγκεκριμένη κατανομή, με εναλλακτική ότι δεν την ακολουθούν. Με βάση λοιπόν την τιμή του εν λόγω ελέγχου, φτάνουμε σε τελικά συμπεράσματα σε σχέση με την καταλληλότητα ή μη του μοντέλου (κατανομή) που έχουμε επιλέξει.

Ο έλεγχος καλής προσαρμογής, για μια ποιοτική μεταβλήτη, όπως αναφέραμε παραπάνω, ελέγχει αν υπάρχουν στατιστικές διαφορές μεταξύ των κατηγοριών του δείγματος, έχοντας ως μέτρο σύγκρισης, το πόσο συχνά εμφανίζεται η κάθε κατηγορία του δείγματος, με το πόσο συχνά περιμέναμε να εμφανιστεί ως αποτέλεσμα η κάθε μια. Η συχνότητες εμφάνισης ονομάζονται πραγματικές συχνότητες ενώ η συχνότητες που περιμέναμε ονομάζονται αναμενόμενες συχνότητες, όταν βέβαια ισχύει η μηδενική υπόθεση του ελέγχου, δηλαδή όταν με βάση την υπόθεση όλες οι κατηγορίες των δειγμάτων έχουν ίδια συχνότητα εμφάνισης. Ο πιο συνηθισμένος έλεγχος καλής προσαρμογής είναι ο x^2 έλεγχος.

**Υποθέσεις στον Έλεγχο Καλής Προσαρμογής:**

- Μηδενική Υπόθεση (H0): Τα δεδομένα ακολουθούν την θεωρητική κατανομή.
- Εναλλακτική Υπόθεση (H1): Τα δεδομένα δεν ακολουθούν την θεωρητική κατανομή.

**Παράδειγμα με χρήση της R:**

Στο συγκεκριμένο παράδειγμα θα χρησιμοποιησουμε τον x^2 έλεγχο καλής προσαρμογής για να ελέγξουμε αν τα δεδομένα από ένα πείραμα ακολουθούν μια

ομοιόμορφη κατανομή. Έστω ότι έχουμε ένα ζάρι που το ρίξαμε 30 φορές και καταγράψαμε τα εξής αποτελέσματα:

- 1: 10 φορές
- 2: 8 φορές
- 3: 6 φορές
- 4: 2 φορές
- 5: 2 φορές
- 6: 2 φορές

Όπου οι αριθμοί 1,2,3,4,5 και 6 καθορίζουν πόσες φορές όταν ρίξαμε το ζάρι φέραμε ως αποτέλεσμα 1,2,3,4,5 και 6 αντίστοιχα. Θέλουμε να ελέγξουμε αν το ζάρι είναι «τυχερό», δηλαδή αν κάθε πλευρά έχει την ίδια πιθανότητα να εμφανιστεί.

Θα δημιουργήσουμε μια μεταβλητή που θα καθορίζει τις πραγματικές συχνότητες με όνομα real_dice και μια η οποία θα καθορίζει τις αναμενόμενες συχνότητες με όνομα expect_dice. Αρχικά, καταχωρούμε τις παρατηρούμενες συχνότητες στην R. Εάν το ζάρι είναι «τυχερό», κάθε πλευρά θα πρέπει να εμφανίζεται 30/6 = 5 φορές. Χρησιμοποιούμε τη συνάρτηση chisq.test() για να συγκρίνουμε τις πραγματικές συχνότητες με τις αναμενόμενες συχνότητες.

```
> real_dice <- c(10,8,6,2,2,2)
>
> expect_dice <- rep(5,6)
>
> result_of_dice <- chisq.test(real_dice, p=expect_dice/sum(expect_dice))
>
> print(result_of_dice)

        Chi-squared test for given probabilities

data:  real_dice
X-squared = 12.4, df = 5, p-value = 0.0297
```

*Εικόνα 71: Παράδειγμα ελέγχου καλής προσαρμογής με χρήση της γλώσσας R.*

**Ανάλυση των Αποτελεσμάτων:**

- x^2 = 12.4: Αυτή είναι η υπολογισμένη τιμή του x^2 για τον έλεγχο.
- p-value = 0.0297: Η p-value δείχνει την πιθανότητα να παρατηρήσουμε μια τόσο ακραία x^2 τιμή ή πιο ακραία, υπό την υπόθεση ότι η μηδενική υπόθεση είναι αληθής. Στην περίπτωση αυτή, με p-value = 0.0297 < 0.05, καταλήγουμε στο συμπέρασμα ότι το ζάρι δεν είναι «τυχερό».

## 3.6 Έλεγχος Ανεξαρτησίας & Πίνακες Συνάφειας

Ο έλεγχος ανεξαρτησίας εφαρμόζεται για να εξετάσουμε τη συνάφεια μεταξύ δύο ποιοτικών μεταβλητών με την έννοια της ανεξαρτησίας μεταξύ των γραμμών και στηλών του πίνακα διπλής εισόδου (ή συνάφειας) των δύο μεταβλητών. Η βασική ιδέα είναι να διαπιστώσουμε πόσο πιθανό είναι να παρατηρήσουμε τις συχνότητες ενός πίνακα με δεδομένο ότι η μηδενική υπόθεση είναι αληθής. Υπάρχουν οι έννοιες της

- Μηδενικής Υπόθεσης η οποία δηλώνει ότι δεν υπάρχει συσχέτιση μεταξύ των δύο μεταβλητών στον πληθυσμό, άρα είναι στατιστικά ανεξάρτητες, κάθως και η έννοια της
- Εναλακτικής Υπόθεσης η οποία προτείνει ότι οι δύο μεταβλητές σχετίζονται στον πληθυσμό και επομένως οι μεταβλητές εξαρτώνται στατιστικά.

Ο έλεγχος ανεξαρτησίας προβάλλει τα δεδομένα ως ένα δείγμα στο οποίο κάθε άτομο ταξινομείται σε δύο διαφορετικές μεταβλητές. Τα δεδομένα παρουσιάζονται συνήθως σε ένα πίνακα (πίνακας συνάφειας) με τις κατηγορίες της μίας μεταβλητής να καθορίζουν τις σειρές και τις κατηγορίες της δεύτερης μεταβλητής να καθορίζουν τις στήλες. Τα δεδομένα, που ονομάζονται παρατηρούμενες συχνότητες, δείχνουν απλά πόσα άτομα από το δείγμα βρίσκονται σε κάθε κελί του πίνακα συνάφειας. Η μηδενική υπόθεση χρησιμοποιείται για την κατασκευή μιας εξιδανικευμένης κατανομής δείγματος των αναμενόμενων συχνοτήτων που περιγράφει τον τρόπο εμφάνισης του δείγματος εάν τα δεδομένα ήταν σε απόλυτη συμφωνία με την μηδενική υπόθεση. Ίσχύουν οι κανόνες που ισχύουν και στον έλεγχο καλής προσαρμογής.

**Παράδειγμα με χρήση της R:**

Έστω ότι έχουμε δεδομένα από μια έρευνα 140 ανθρώπων, όπου συμμετέχοντες ταξινομούνται ανάλογα με την προτίμηση χρώματος αυτοκινήτου και το φύλο του αγοραστή. Θέλουμε να ελέγξουμε αν η προτίμηση χρώματος αυτοκινήτου είναι ανεξάρτητη από το φύλο του αγοραστή. Τα δεδομένα είναι τα εξής:

- 40 άνδρες προτιμούν το κόκκινο αυτοκίνητο.
- 20 άνδρες προτιμούν το μπλε αυτοκίνητο.
- 30 γυναίκες προτιμούν το κόκκινο αυτοκίνητο.

- 50 γυναίκες προτιμούν το μπλε αυτοκίνητο.

Ως πρώτο βήμα, δημιουργούμε τον πίνακα συνάφειας για αυτά τα δεδομένα. Στη συνέχεια, θα εκτελέσουμε το x^2 τεστ για να εξετάσουμε την ανεξαρτησία μεταξύ του φύλου και της προτίμησης χρώματος αυτοκινήτου. Έπειτα θα πάρουμε τα παρακάτω αποτελέσματα.

```
> car_color <- matrix(c(40,20,30,50), nrow=2, byrow=TRUE)
>
> rownames(car_color) <- c("Men","Women")
>
> colnames(car_color) <- c("Red", "Blue")
>
> print(car_color)
      Red Blue
Men    40   20
Women  30   50
>
> car_result <- chisq.test(car_color)
>
> print(car_result)

        Pearson's Chi-squared test with Yates' continuity
        correction

data:  car_color
X-squared = 10.529, df = 1, p-value = 0.001175
```

*Εικόνα 72: Παράδειγμα ελέγχου ανεξαρτησίας & πινάκων συνάφειας με χρήση της γλώσσας R.*

**Ανάλυση των Αποτελεσμάτων:**

- x^2: Το x^2 στατιστικό είναι 10.529
- p-value = 0.0001: Η τιμή p είναι 0.001175, που είναι πολύ μικρότερη από 0.05. Αυτό σημαίνει ότι έχουμε αρκετές αποδείξεις για να απορρίψουμε τη μηδενική υπόθεση, και συνεπώς υπάρχει στατιστικά σημαντική σχέση μεταξύ του φύλου και της προτίμησης χρώματος αυτοκινήτου.

Με βάση αυτό το τεστ, μπορούμε να πούμε ότι η προτίμηση χρώματος αυτοκινήτου δεν είναι ανεξάρτητη από το φύλο, δηλαδή το φύλο του ατόμου επηρεάζει την προτίμησή του για το χρώμα του αυτοκινήτου.

# Συμπεράσματα

Με βάση την παραπάνω εργασία, καταλήγουμε στο συμπέρασμα να έχουμε αποκτήσει μία βασική γνώση με σκοπό να κατανοήσουμε πλήρως την γλώσσα προγραμματισμού R, να μάθουμε ποια είναι τα οφέλη της ώστε να την προτιμήσουμε για χρήση, αλλά αντίστοιχα και τι πρέπει να αποφύγουμε. Επίσης, στο πρακτικό κομμάτι, ένας από τους κυριότερους στόχους είναι να εξοικειωθούμε με το περιβάλλον του λογισμικού R Studio, να μπορούμε να αναπτύξουμε βασικές λειτουργείες της γλώσσας πάνω σε πίνακες, διανύσματα αλλα και άλλες μορφές δεδομένων και να μπορούμε να τις εφαρμόσουμε σε οποιοδήποτε παράδειγμα. Ακόμη, όσον αφορά την στατιστική συμπερασματολογία, να μάθουμε την έννοια της και σε ποιους τομείς χρησιμοποιείται, τι είναι η περιγραφική στατιστική, αλλά και να μπορέσουμε να ξεχωρίσουμε τις διαφορές του ελέγχου ανεξαρτησίας (με την βοήθεια των πινάκων συνάφειας), του ελέγχου υποθέσεων, καθώς και του ελέγχου καλής προσαρμογής.

# Βιβλιογραφία


1. Ross I. (1998) Past and Future History A Draft of a Paper for Interface ’98 https://www.stat.auckland.ac.nz/~ihaka/downloads/Interface98.pdf
2. Κ. Φωκιανός Χ. Χαραλάμπους Κεφάλαιο 7 Στατιστική Συμπερασματολογία http://www.mas.ucy.ac.cy/~fokianos/GreekRbook/dialeksi7.pdf
3. Educational Research Techniques (2015) The history & characteristics of R https://educationalresearchtechniques.com/2015/06/11/the-history-and-characteristics-of-r/
4. Akshat S. (2016) Understanding Decision Tree Algorithm by Using R Programming Language https://themys.sid.uncu.edu.ar/~rpalma/Toma_Decision/Papers/R-decision-tree.pdf
5. Ντούφρας Ι. Καρλής Δ. (2015) Εισαγωγή στον Προγραμματισμό και στη Στατιστική Ανάλυση με R file:///C:/Users/pc/Downloads/Εισαγωγή%20στον%20προγραμματισμό%20και%20στη%20στατιστική%20ανάλυση%20με%20R.pdf
6. Ρουμπίνη Σ. (2016) Δηπλωματική Εργασία file:///C:/Users/pc/Downloads/ΔΙΠΛΩΜΑΤΙΚΗ%20ΕΡΓΑΣΙΑ-ΡΟΥΜΠΙΝΑ%20ΣΟΦΡΑ.Pdf
7. Guru 99 What is R Programming Language? Introduction & Basics of R https://www.guru99.com/r-programming-introduction-basics.html
8. R-project.org What is R? https://www.r-project.org/about.html
9. Wikipedia R (Γλώσσα Προγραμματισμού) https://el.wikipedia.org/wiki/R_(γλώσσα_προγραμματισμού)
10. Wikipedia R (Ιστορική Αναδρομή) https://en.wikipedia.org/wiki/R_(programming_language)#History
11. Wikipedia S (Γλώσσα Προγραμματισμού) https://en.wikipedia.org/wiki/S_(programming_language)
12. Data Flair Training Pros and Cons of R Programming Language https://data-flair.training/blogs/pros-and-cons-of-r-programming-language/
13. Javatpoint R advantages and disadvantages https://www.javatpoint.com/r-advantages-and-disadvantages
14. Techvidvan R Data Types https://techvidvan.com/tutorials/r-data-types/
15. W3Schools R Data Types https://www.w3schools.com/r/r_data_types.asp
16. TutorialSPoint R Data Types https://www.tutorialspoint.com/r/r_data_types.html
17. R-Coder R Data Types https://r-coder.com/data-types-r/
18. Analytic Steps Data Types in R https://www.analyticssteps.com/blogs/data-types-r
19. RStudio.com R Packages

https://www.rstudio.com/products/rpackages/
20. DataCamp Top 10 Most Important Packages in R for Data Science https://www.datacamp.com/community/tutorials/top-ten-most-important-packages-in-r-for-data-science
21. Free Code Camp (2021) https://www.youtube.com/watch?v=_V8eKsto3Ug (free code camp)
22. Data Camp R Packages Guide https://www.datacamp.com/community/tutorials/r-packages-guide
23. TutorialSPoint R Packages https://www.tutorialspoint.com/r/r_packages.html
24. Data Flair Training R Packages https://data-flair.training/blogs/r-packages-tutorial/
25. Τρέβεζας Σ. (2016-2017) Μια πολύ σόντομη εισαγωγή στην R https://eclass.uoa.gr/modules/document/file.php/MATH452/Εργαστήρια%20R/Intro-R.pdf
26. OutlierAnalytics (2019) Top 8 R Packages-Libraries for Data Science in 2019 https://blog.outliersanalytics.com/top-8-r-packages-libraries-for-data-science-in-2019/
27. Techvidvan R Data Structures https://techvidvan.com/tutorials/r-data-structures/
28. W3Schools R Vectos https://www.w3schools.com/r/r_vectors.asp
29. Παράδης Ε. (2012) Analysis of Phylogenetics and Evolution in R Second Edition https://books.google.gr/books?hl=el&lr=&id=FsScWggkW_cC&oi=fnd&pg=PR3&dq=evolution+of+r+programming+language&ots=EwwXAQmnj5&sig=wOzArrOJiQsUoK47gTrayGVhWpI&redir_esc=y#v=onepage&q=evolution%20of%20r%20programming%20language&f=false
30. Data Flair Training R Vector https://data-flair.training/blogs/r-vector/
31. Medium.com Nerd for Tech/R Tutorial Apply Function https://medium.com/nerd-for-tech/r-tutorial-apply-function-family-in-r-cbbd4b6781cd
32. Rstudio.com Products of R https://www.rstudio.com/products/rstudio/
33. Techvidvan RStudio Tutorial https://techvidvan.com/tutorials/rstudio-tutorial/
34. Wikipedia RStudio https://en.wikipedia.org/wiki/RStudio
35. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.651.1157&rep=rep1&type=pdf#page=14
36. Geeks for Geeks Introduction to R Studio
37. https://www.geeksforgeeks.org/introduction-to-r-studio/
38. Science Direct Statistical Inference https://www.sciencedirect.com/topics/neuroscience/statistical-inference
39. Καραβασίλης Γ. Στατιστική Μεθολογία https://users.sch.gr/gkaravasilis/R_DescrStat.pdf

40. Φουσκάκης Δ. Ανάαλυση Δεδομένων με Χρήση Στατιστικού Πακέτου R Εθνικό Μετσόβειο Πολυτεχνείο Σχολή Εφαρμοσμένων Μαθηματικών & ΦυσικώνΕπιστημών http://www.math.ntua.gr/~fouskakis/Data_Analysis/Slides/06.pdf
41. Μάρκος Α. Συνάφεια Μεταξύ Ποιοτικών Μεταβλητών http://www.amarkos.gr/material/Συνάφεια.pdf
42. Δημητριάδης Ε. Ποσοτικές Μέθοδοι για Στελέχη Επειχηρήσεων, χ2 έλεγχος Ανεξαρτησίας, Μεταπτυχιακό Πρόγραμμα Σπουδών, ΙΗU https://eclass.mst.duth.gr/modules/document/file.php/MBA-TOURISM109/ΔΙΑΛΕΞΕΙΣ_POWER%20POINT_ΠΟΣΟΤΙΚΕΣ%20ΜΕΘΟΔΟΙ/QUANTITATIVE%20METHODS_7_MBA_CHI-SQUARE.pdf
43. Παπαδόπουλος Γ. ‘Ελεγχος Καλής Προσαρμογής , Ανεξαρτησίας και Ομογένειας https://www.aua.gr/gpapadopoulos/files/14x2test14-15.pdf

# Παράρτημα Α.