Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Τομέας Ηλεκτρονικής και Υπολογιστών

9ο Εξάμηνο
Φθινόπωρο 2020

R Tutorial

1 Εισαγωγή στην R

Η γλώσσα προγραμματισμού R αποτελεί open source λογισμικό το οποίο είναι διαθέσιμο στην ιστοσελίδα *http://www.r-project.org/*. Αν και μπορεί να χρησιμοποιηθεί κανονικά απευθείας μετά την εγκατάσταση της, υπάρχει μία επίσης open source διεπαφή, φιλική προς τον χρήστη, το RStudio (http://www.rstudio.com/) το οποίο περιέχει κάποιες διευκολύνσεις.

Οι συναρτήσεις στην R είναι διαμοιρασμένες σε πακέτα (packages) τα βασικότερα από τα οποία φορτώνονται αυτόματα κατά την έναρξη. Κάποια από αυτά είναι τα "base", "utils", "graphics" και "stats" τα οποία περιέχουν πολλές από τις πιο συχνά χρησιμοποιούμενες συναρτήσεις. Ωστόσο, προκειμένου να χρησιμοποιηθούν άλλες χρήσιμες συναρτήσεις θα πρέπει ο χρήστης να κατεβάσει επιπλέον packages. Αυτό πραγματοποιείται εύκολα μέσα από το περιβάλλον της RGui. Εάν για παράδειγμα θέλουμε να κατεβάσουμε το πακέτο "e1071" το οποίο περιέχει χρήσιμες συναρτήσεις για τα SVM, κάνουμε click στο "Packages" -> " Install Packages"->... το οποίο βρίσκεται στο βασικό menu του RGui. Αντίστοιχα από το ίδιο menu μπορούμε να φορτώσουμε κάποιο package όταν το χρειαζόμαστε. Τις ίδιες λειτουργίες μπορούμε να κάνουμε και με τις εντολές `install.packages("e1071")` και `library("e1071")` για εγκατάσταση και φόρτωση του "e1071" αντίστοιχα. Με την εντολή `library()` εμφανίζονται όλες οι βιβλιοθήκες οι οποίες έχουν εγκατασταθεί.

Προκειμένου να προβάλλουμε το path στο οποίο βρισκόμαστε πληκτρολογούμε την εντολή `getwd()`, ενώ η αλλαγή του path γίνεται με την `setwd("C:\\...")`. Με την εντολή `ls()` μπορούμε να δούμε όλα τα αντικείμενα που βρίσκονται στο workspace μας. Ενώ με την εντολή `rm(list=ls())` γίνεται καθαρισμός ολόκληρου του workspace. Η σύντομευση `Ctrl + L` πραγματοποιεί καθαρισμό της οθόνης.

Για την προβολή του documentation μίας συνάρτησης π.χ. της `sum()`, μπορούμε να πληκτρολογήσουμε `help(sum)` ή `?sum`. Επιπλέον, για κάποιες συναρτήσεις μας δίνεται η δυνατότητα να δούμε ένα παράδειγμα που χρησιμοποιεί την συγκεκριμένη συνάρτηση που θέλουμε, με την εντολή `example("function")`, π.χ. `example("plot")`. Η χρήση σχολίων επιτρέπεται με τον χαρακτήρα `#`, ενώ συχνά ο χαρακτήρας `<-` χρησιμοποιείται συχνά αντί του `=`.

Εναλλακτικά προκειμένου να τρέξουμε ή και να σώσουμε τον κωδικά μας για μελλοντική χρήση, αντί να γράψουμε απευθείας στην κονσόλα πάμε File->New script. Από εκεί επιλέγουμε τις γραμμές κώδικα που θέλουμε να τρέξουμε ή και ολόκληρο τον κώδικα και πατάμε Ctrl-R. Διαφορετικά μπορούμε να επιλέξουμε Edit->Run all.

2 Βασικές δομές με παραδείγματα

2.1 Διανύσματα

Να δημιουργηθεί το διάνυσμα A = (10, 5, 3, 100, -2, 5, -50) και να τυπωθεί στην οθόνη.

```
> A = c(10,5,3,100,-2,5,-50)
> A
```

Επιλογή των στοιχείων 1, 3, 4 και 5 του διανύσματος A.

```
> A[c(1,3:5)]
```

Ποια στοιχεία του A έχουν τιμή μεγαλύτερη του 5;

```
> A > 5
> which(A>5)   # Επιστρέφει τις θέσεις των στοιχείων του A για τα
                                    #οποία ισχύει η ανισότητα
```

Επιλογή μόνο των στοιχείων του A που έχουν τιμή θετική.

```
> S = A > 0     # Εύρεση με ετικέτα (TRUE, FALSE) των στοιχείων του A
                                        # που είναι θετικοί αριθμοί.
> positives = A[S]          # Επιλογή μόνο των θετικών στοιχείων του A
> positives            # Εκτύπωση στην οθόνη της μεταβλητής positives
```

ή εναλλακτικά

```
> positives = A[A>0]
> positives
```

Για να δούμε πόσα είναι αυτά αρκεί να χρησιμοποιήσουμε τη `length`.

Δημιούργια ενός δεύτερου διανύσματος B και ένωσή του με το διάνυσμα A.

```
> A = c(10,5,3,100,-2,5,-50)
> B = c(1,2,5,6,9,0,100)
> cbind(A,B)                    # Προσθήκη σαν στήλη
> rbind(A,B)                    # Προσθήκη σαν γραμμή
```

2.2 Πίνακες

Να δημιουργηθεί πίνακας με 3 σειρές που να περιέχει τους αριθμούς 1 εώς 9.

```
> m = matrix(1:9,byrow = TRUE, nrow=3)
> m
```

Να δημιουργηθεί ένας δεύτερος πίνακας ιδίας διάστασης και να ενοποιηθεί με τον παραπάνω πίνακα. Στον νέο πίνακα που δημιουργήθηκε να υπολογιστούν τα αθροίσματα ανά γραμμή, ανά στήλη καθώς και ο μέσος όρος.

```
> m2 = matrix(10:18,byrow = TRUE, nrow=3)
> Mat = cbind(m,m2)     # ή rbind() για σύνδεση των πινάκων κατά σειρά
> Mat
> rowSums(Mat)
> colSums(Mat)
> mean(Mat)
```

Να δημιουργηθούν 2 πίνακες a και b και να υπολογιστεί το αποτέλεσμα του πολλαπλασιασμού των στοιχείων των δύο πινάκων ένα προς ένα, όπως και επίσης το αποτέλεσμα του πολλαπλασιασμού των δύο πινάκων.

```
> a = matrix(10:18,byrow = TRUE, nrow=3)
> b = matrix(c(3,6,7,10,8,1,2,3,2),byrow = TRUE, nrow=3)
> a*b             # Πολλαπλασιασμός στοιχείο προς στοιχείο
> a%*%b           # Πολλαπλασιασμός πινάκων
```

2.3 Data frames

Αποτελεί μία δομή συχνά χρησιμοποιούμενη στην R, για την αποθήκευση πινάκων δεδομένων. Αποτελεί γενίκευση των πινάκων και μπορεί να συνδυάζει διαφορετικές μεταξύ τους δομές (π.χ. διανύσματα με χαρακτήρες, πίνακες με αριθμητικά κλπ).

Παράδειγμα δημιουργίας ενός data frame και προσπέλαση των στοιχείων του.

```
> n = c(2, 3, 5)
> s = c("aa", "bb", "cc")
```

```
> b = c(TRUE, FALSE, TRUE)

# Δημιουργία του data frame df και προβολή του στην οθόνη.

> df = data.frame(n, s, b)

> df

# Προβολή της στήλης n του data frame με χρήση του χαρακτήρα '$'

> df$n
```

3 Programming Tools

3.1 File I/O

Για να δούμε το path στο οποίο βρισκόμαστε πληκτρολογούμε την εντολή `getwd()`, ενώ η αλλαγή του path γίνεται με την `setwd("C:\\...")`.

Για να διαβάσουμε τα δεδομένα ενός αρχείου από το δίσκο μπορούμε να χρησιμοποιήσουμε τη συνάρτηση `read.csv`. Έστω ότι έχουμε το αρχείο `people.txt` με δεδομένα:

```
Age, Height, Weight
25, 1.95, 85
18, 1.82, 80
44, 1.75, 82
26, 1.78, 75
34, 1.77, 78
```

Μπορούμε να το διαβάσουμε με την εντολή:

```
> people = read.csv("people.txt")
```

Η συνάρτηση μας επιτρέπει να επιλέξουμε το χαρακτήρα οριοθέτησης με την παράμετρο `sep` (το default είναι το κόμμα) ενώ η παράμετρος `header` ελέγχει αν η πρώτη γραμμή είναι το header. Επίσης μπορούμε να θέσουμε ονόματα στηλών με την παράμετρο `col.names`.

Για τα παρακάτω δεδομένα:

```
25; 1.95; 85
18; 1.82; 80
44; 1.75; 82
26; 1.78; 75
34; 1.77; 78
```

εκτελούμε την εντολή:

```
> people = read.csv("people.txt", sep = ';', header = FALSE, col.names
  = c("Age", "Height", "Weight"))
```

Σε περίπτωση που λείπουν κάποια δεδομένα (έχουμε missing values), μπορούμε να τα αντικαταστήσουμε με τη διάμεσο. Π.χ. για το παρακάτω dataset

```
25; 1.95; 85
18; 1.82; 80
44; 1.75;
26; 1.78; 75
34; 1.77; 78
```

αφού φορτώσουμε τα δεδομένα, μπορούμε να εκτελέσουμε τις παρακάτω εντολές:

```
m = median(people$Weight, na.rm = TRUE)

people$Weight[is.na(people$Weight)] = m
```

3.2 Βρόχοι

Δημιουργία ενός βρόχου for που τυπώνει όλες τις ακέραιες τιμές από 1 εώς 10.

```
> for(i in 1:10) {

+ print(i)

+ }
```

Εκτύπωση όλων των άρτιων αριθμών στο διάστημα [0,100]

```
> for(i in 0:100){

+ if(i %% 2 ==0){

+ print(i)

+ }}
```

4 Συναρτήσεις Στατιστικής

Τρέχοντας την εντολή `data()` μπορούμε να δούμε κάποια datasets που περιλαμβάνονται στην R, και μπορούμε να τα χρησιμοποιήσουμε άμεσα. Π.χ. για να δούμε το iris dataset, αρκεί να γράψουμε `iris`.

Μπορούμε να δούμε λεπτομέρειες για κάποιο data frame (ή για οποιοδήποτε type της R), τρέχοντας την εντολή `str`.

```
> str(iris)
```

Για μια λεπτομερή στατιστική αναφορά, χρησιμοποιούμε την εντολή summary:

```
> summary(iris)
```

Υπάρχουν επίσης οι μεμονωμένες συναρτήσεις για τα γνωστά στατιστικά μεγέθη (πολλές από αυτές όμως εφαρμόζονται μόνο σε 1d δεδομένα):

```
> sl = iris$Sepal.Length  # Επιλογή της στήλης Sepal.Length του iris
> mean(sl)
> median(sl)
> min(sl)
> max(sl)
> sd(sl)      # Standard Deviation
> var(sl)     # Variance
> range(sl)
```

Επίσης, υποστηρίζονται πιο σύνθετες συναρτήσεις όπως για παράδειγμα το cross correlation:

```
> cor(iris[1:4])
```

Τέλος, για απλές γραφικές παραστάσεις χρησιμοποιείται η συνάρτηση `plot` για την οποία μπορείτε να δείτε πληροφορίες πατώντας `help(plot)`.

5 Πρόβλημα για Εξάσκηση

Χρησιμοποιήστε το dataset AirPassengers της R που αναφέρεται σε δεδομένα πλήθους επιβατών για κάθε μήνα από το έτος 1949 μέχρι και το έτος 1960. Μπορείτε να φορτώσετε το dataset ως data frame με τις παρακάτω εντολές:

```
> dn = list(paste("Y", as.character(1949:1960), sep = ""), month.abb)
> airmat = matrix(AirPassengers, 12, byrow = TRUE, dimnames = dn)
> air = as.data.frame(t(airmat))
```

Στη συνέχεια, απαντήστε στα παρακάτω ερωτήματα:

- Πόσοι επιβάτες ταξίδεψαν κατά μέσο όρο για το έτος 1951;
- Ποιος είναι ο μέγιστος αριθμός επιβατών για τους μήνες Ιανουάριο και Φεβρουάριο;
- Υπολογίστε το cross correlation για όλες τις διαστάσεις του data frame. Τι παρατηρείτε;
- Υπολογίστε το άθροισμα για κάθε έτος και αποθηκεύστε το αποτέλεσμα σε ένα vector.
- Σχεδιάστε σε ένα διάγραμμα το vector που κατασκευάσατε.
- Επαναλάβετε τα δύο τελευταία ερωτήματα για κάθε μήνα και για όλα τα έτη.

Υπόδειξη: για να μετατρέψετε μια γραμμή (όχι στήλη) του data frame σε vector μπορείτε να χρησιμοποιήσετε την `unlist` (π.χ. `unlist(air["Jan",])`).