

Laborator 1

1 R și RStudio

R este un mediu dedicat analizei statistice introdus în 1996; este întreținut și dezvoltat în lumea academică și are avantajul de a fi open-source spre deosebire de alte pachete statistice cunoscute (Minitab, SPSS etc).

Cu R se poate lucra din linia de comandă sau cu ajutorul unei interfețe grafice; în cele ce urmează vom folosi o astfel de interfață grafică: RStudio care este open-source și poate fi folosită pe Linux, Windows sau Mac.

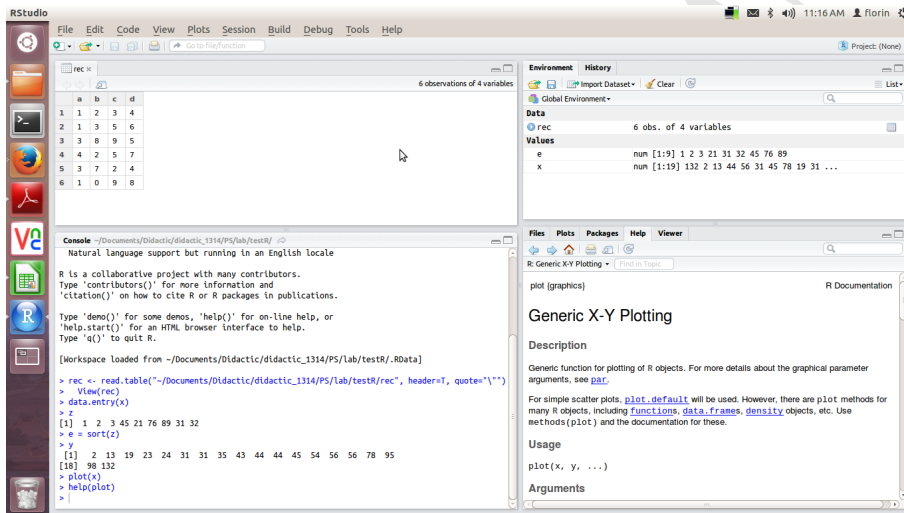


Figure 1: RStudio screenshot

RStudio conține (vezi figura de mai sus) uzual patru panouri (începând din stânga sus, în sens orar): unul în care se pot edita și executa scripturile R (care conțin funcții și comenzi) sau fișierele de date, unul dedicat vizualizării variabilelor, istoricului comenzilor etc., în al treilea apar graficele, se poate parcurge help-ul atașat comenzilor și conținutul fișierelor din directorul curent, iar ultimul panel este cel care conține prompterul liniei de comandă (>) - aici se pot executa comenzi R.

1.1 Sesiunea RStudio

O sesiune trebuie să înceapă prin setarea directorului de lucru: **Session** → **Set Working Directory** → **Choose Directory** și se va termina prin salvarea spațiului de lucru (în fereastra de dialog, la întrebarea "Save workspace image to `/.RData`?" alegeți "Save" din **Session** → **Save Workspace As**).

1.2 Variabile și tipuri

În R variabilele sunt uzual vectori sau matrici. Orice variabilă poate fi vizualizată prin simpla apelare a numelui. Tipurile folosite sunt numeric, șiruri de caractere (de exemplu " a43fdt") și boolean (TRUE sau T, FALSE sau F)

Asignarea Există două simboluri pentru asignare în R: `=` și `<-` (fără spații, se recomandă folosirea lui pentru compatibilitate cu versiuni mai vechi de R).

Crearea unui vector Avem mai jos trei metode diferite de a crea un vector:

- prin concatenare folosind funcția `c()`,
- ca o secvență de numere întregi consecutive - pentru acest exemplu am afișat conținutul vectorului care este indexat începând cu 1 (la afișare fiecare linie va fi introdusă prin indicele primului element de pe acea linie - dacă vectorul încapă pe o singură linie va apare doar [1])
- sau construit ca o secvență a cărei îi sunt indicate numărul inițial, cel final și numărul de termeni cu ajutorul funcției `seq()`

```
> x = c(1, 3, 2, 15, 6, 21, 34, 54, 7)
> x = c(T, T, F, T, F)
> x
[1] TRUE TRUE FALSE TRUE FALSE
> x = -5:13
> x
[1] -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9
[16] 10 11 12 13
> x = seq(-3, 3, length=100)
```

Elementele unui vector pot fi accesate în felul următor

```
> x = c(23, 21, 32, 25, 34, 19, 32, 45, 67)
> x[4] # al 4-lea element
[1] 25
x[2:6] # elementele de la 2 până la 6 inclusiv
[1] 21 32 25 34 19
x[-3] # toate elementele mai puțin al 3-lea
[1] 23 21 25 34 19 32 45 67
```

Un vector poate fi editat folosind funcția `data.entry(vector)`.

1.3 Operații aritmetice și funcții predefinite

Se pot face direct din linia de comandă folosind valori numerice sau variabile

```
> sin(1)
[1] 0.841471
> log(2)
[1] 0.6931472
> x = 3
> x^2
[1] 9
> exp(x)
[1] 20.08554
```

Operațiile efectuate cu vectori sunt de obicei la nivelul fiecărei componente

```
> x = c(1, 3, 2, 15, 6, 21, 34, 54, 7)
> y = c(22, 11, 32, 25, 54, 13, 27, 36, 2)
> x^2
[1] 1 9 4 225 36 441 1156 2916 49
> x + y
[1] [1] 23 14 34 40 60 34 61 90 9
```

R conține în general funcții matematice și statistice care pot manipula vectori, matrici sau variabile simple

```
> x <- c(1, 3, 2, 15, 6, 21, 34, 54, 7)
> length(x)
> [1] 9
> sort(x)
> [1] 1 2 3 6 7 15 21 34 54
> sqrt(x)
[1] 1.000000 1.732051 1.414214 3.872983 2.449490
[6] 4.582576 5.830952 7.348469 2.645751
> exp(x)
[1] 2.718282e+00 2.008554e+01 7.389056e+00
[4] 3.269017e+06 4.034288e+02 1.318816e+09
[7] 5.834617e+14 2.830753e+23 1.096633e+03
```

Informații despre o funcție pot fi obținute folosind `help(ume_funcție)` în linia de comandă.

1.4 Funcții definite de către utilizator

O funcție poate fi definită din linia de comandă astfel: să presupunem că dorim să calculăm dispersia unei distribuții

```
> dispersie = function (x, p) {
+ media = sum(p*x);
+ dispersie = sum(p*(x - media)^2);
+ return (dispersie)
+ }
> y = c(23, 32, 31, 27, 27, 33, 25, 21)
> q = c(1/8, 1/16, 1/8, 1/16, 1/8, 1/16, 1/8, 5/16)
> dispersie(y, q)
```

Dar, o astfel de funcție poate fi scrisă într-un script astfel **File** → **New File** → **R Script** și în fereastra de editare se scrie codul

```
dispersie = function(x, p) {
  media = sum(p*x);
  dispersie = sum(p*(x - media)^2);
  return (dispersie)
}
```

RStudio. După editare, scriptul este salvat (**Ctrl+S**) cu un nume de tipul "my_script.R" și este încărcat cu **Code** → **Source File** (**Ctrl+Shift+O**¹ sau **Ctrl+Shift+S**²) sau din linia de comandă cu `source(script_file)`.

În același script funcția astfel scrisă poate fi și executată pentru anumite argumente, de exemplu putem adăuga la script

```
> y = c(23, 32, 31, 27, 27, 33, 25, 21)
> q = c(1/8, 1/16, 1/8, 1/16, 1/8, 1/16, 1/8, 5/16)
> dispersie(y, q)
```

¹Ubuntu Linux.

²Windows.

RStudio. O dată încărcat scriptul, o funcție care face parte din acest script se poate executa din linia de comandă: **dispersie(y, q)** sau din fereastra de editare astfel: se selectează liniile dorite a fi executate și **Ctrl+Enter**, iar scriptul în întregime se execută cu **Ctrl+Alt+R**.

O funcție poate fi modificată în fereastra de editare a scriptului sau din linia de comandă cu *fix(function_name)*

```
> fix(dispersie)
```

1.5 Manipularea fișierelor cu date

Dacă un fișier "my_file" (aflat în directorul de lucru, altfel trebuie adăugată și calea relativă) conține un singur vector de date (fără antet), el poate fi citit și transformat într-un vector astfel

```
> x = scan("my_file")
```

Dacă fișierul conține un antet (să spunem că două dintre coloane sunt numite "col1" și "col2"), atunci procedăm astfel

```
> y = read.table("my_file", header = T) # acest obiect conține și antetul  
> x1 = y[["col1"]] # acest vector conține doar datele numerice din coloana "col1"  
> x2 = y[["col2"]] # acest vector conține doar datele numerice din coloana "col2"
```

Pot fi citite și fișiere de tip .csv (comma separated values):

```
> x = read.csv(file="date.csv", header = T)
```

1.6 Structuri iterative și de control

R conține structuri standard pentru iterații și de control utile în scrierea funcțiilor:

```
> if (condition){  
+   statement  
> }else  
+   {  
+   alternative  
+   }
```

```
> for (var in sequence){  
+   statement  
> }
```

```
> while (condition){  
+   statement  
> }
```

Următoarea funcție folosește astfel de structuri:

```
vector_sqrt = function(x) {  
  for(i in 1:length(x)) {  
    if(x[i] > 0)  
      x[i] = sqrt(x[i])  
    else)  
      x[i] = sqrt(-x[i])  
  }  
}
```

2 Simularea variabilelor aleatoare (Ilustrări ale LNM și TLC)

2.1 Distribuții continue remarcabile

Exercițiu rezolvat. Reprezentați grafic funcția de densitate a distribuției exponențiale, $Exp(\lambda)$ ($\lambda > 0$).

Această distribuție se anulează pe toată semiaxa negativă, deci este suficient să o reprezentăm pe semiaxa pozitivă (vom folosi, de fapt, un interval de forma $[0, a]$).

```
density_exponential = function(lambda, n, a) {  
  x = seq(0, a, n);  
  y = dexp(x, lambda);  
  plot(x, y, type = 'l');  
}
```

Exercițiu propus.

1.1. Scrieți o funcție care să reprezinte grafic densitățile următoarelor distribuții:

- (a) $Gamma(\alpha, \lambda)$.
- (b) $Student(r)$.
- (c) $N(\mu, \sigma^2)$.

2.2 Legea numerelor mari (LNM).

Fie X_i , $1 \leq i \leq n$, un șir de variabile aleatoare independente și identic distribuite, media lor aritmetică este

$$\bar{x}_n = \frac{X_1 + X_2 + \dots + X_n}{n},$$

Atunci, conform LNM, $\bar{x}_n \rightarrow \mu$, unde $\mu = \mathbb{E}[X_i]$, $\forall 1 \leq i \leq n$.

Exercițiu rezolvat. Verificați LNM utilizând șirul de variabile $X_i : Poisson(\lambda)$, $1 \leq i \leq n$. (Se știe că $\mathbb{E}[X_i] = \lambda$.)

```
LLN_Poisson = function(lambda, n) {  
  sum = 0;  
  for(i in 1:n) {  
    u = rpois(1, lambda);  
    sum = sum + u;  
  }  
  return(sum/n);  
}
```

O variantă mai simplă și mai rapidă:

```
LLN_Poisson = function(lambda, n) {  
  return(mean(rpois(n, lambda)));  
}
```

Exercițiu rezolvat. Verificați LNM utilizând șirul de variabile $X_i : Gamma(\alpha, \lambda)$, $1 \leq i \leq n$. (Se știe că $\mathbb{E}[X_i] = \alpha/\lambda$.)

Folosim aici doar varianta mai simplă:

```
LLN_Gamma = function(alfa, lambda, n) {  
  return(mean(rgamma(n, alfa, lambda)));  
}
```

Exerciții propuse.

2.1. Scrieți câte o funcție care să verifice LNM pentru următoarele șiruri de variabile aleatoare

(a) $X_i : Exponential(\lambda)$, $1 \leq i \leq n$. (Știm că $\mathbb{E}[X_i] = 1/\lambda$.)

(b) $X_i : B(m, p)$, $1 \leq i \leq n$. (Știm că $\mathbb{E}[X_i] = mp$.)

2.2 Rezolvați același exercițiu pentru șirul de variabile aleatoare $X_i : Student(r)$, $1 \leq i \leq n$, pentru care se știe că $\mathbb{E}[X_i] = 0$. Comparați rezultatele cu valorile exacte pentru următorii parametri: $n \in \{1000, 10000, 100000, 1000000\}$ and $r \in \{2, 3, 4, 5\}$.

2.3 Teorema Limită Centrală (TLC).

Fie X_i , $1 \leq i \leq n$, un șir de variabile aleatoare independente și identic distribuite și fie

$$\bar{x}_n = \frac{X_1 + X_2 + \dots + X_n}{n}$$

media lor aritmetică, atunci, conform TLC, \bar{x}_n (pentru valori foarte mari ale lui n) urmează o distribuție $N(\mu, \sigma^2/n)$.

După standardizarea mediei de selecție obținem că $\frac{\bar{x}_n - \mu}{\sigma/\sqrt{n}} : N(0, 1)$.

Exercițiu rezolvat. Verificați TLC folosind șirul de variabile aleatoare $X_i : Poisson(\lambda)$, $1 \leq i \leq n$. (Se știe că $\mathbb{E}[X_i] = \lambda$ și $Var[X_i] = \lambda$.)

TLC spune că

$$P\left(\frac{\bar{x}_n - \mu}{\sigma/\sqrt{n}} \leq z\right) \cong P(Z \leq z),$$

Unde $z \in \mathbb{R}$ și $Z : N(0, 1)$ - legea normală standard. Probabilitatea din dreapta este $pnorm(z)$, în timp ce probabilitatea din stânga poate fi aproximată astfel: fie N (un număr foarte mare) de eșantioane aleatoare simple, fiecare de dimensiune n , $(X_i^k)_{i=1, n}^{k=1, N}$, apoi calculăm

$$P^N(z) = \frac{|\{k : \bar{x}_n^k \leq z\sigma/\sqrt{n} + \mu\}|}{N}.$$

(X_i^k sunt valori simulate conform distribuției date - Poisson în cazul exercițiului de față). Se compară apoi această probabilitate cu $pnorm(z)$.

```

CLT_Poisson = function(lambda, n, N, z) {
  expectation = lambda;
  st_dev = sqrt(lambda);
  upper_bound = z * st_dev/sqrt(n) + expectation;
  sum = 0;
  for(i in 1:N) {
    x_n = mean(rpois(n, lambda));
    if(x_n <= upper_bound) {
      sum = sum + 1;
    }
  }
  return(sum/N);
}

```

Remarcă: n ar trebui să fie cel puțin 30; e. g., $n = 30$, $N = 10000$, $z \in \{0, 1, 1.5, 2\}$.

Exerciții propuse.

- 3.1. Scrieți o funcție care să verifice TLC folosind șirul de variabile aleatoare $X_i : Exponential(\lambda)$, $1 \leq i \leq n$. (Știm că $\mathbb{E}[X_i] = 1/\lambda$, $Var[X_i] = 1/\lambda^2$.)
- 3.2 Rezolvați același exercițiu pentru șirul de variabile aleatoare $X_i : Gamma(\alpha, \lambda)$, $1 \leq i \leq n$. (We know that $\mathbb{E}[X_i] = \alpha/\lambda$ and $Var[X_i] = \alpha/\lambda^2$.) Choose $n = 50$, $N \in \{5000, 10000, 20000\}$, and $z \in \{-1.5, 0, 1.5\}$.