Proiect la statistică

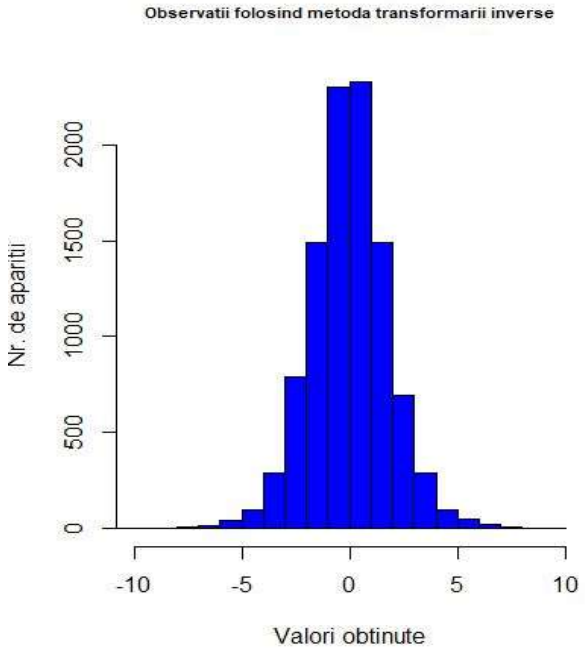
Ivan Andrei Valeriu

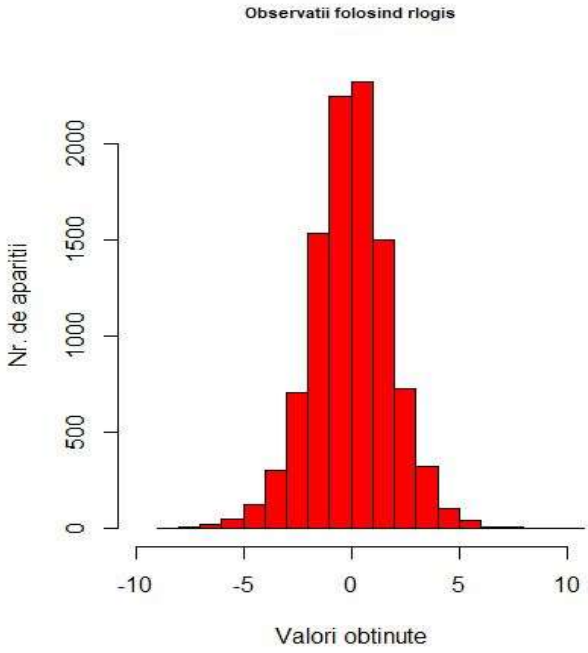
Grupa 321

Exercițiul 1:

a) Funcția de repartiție este continuă, deci inversa ei este f(y)=μ+β*ln(u/(1-u)), unde μ=0 și β=1.

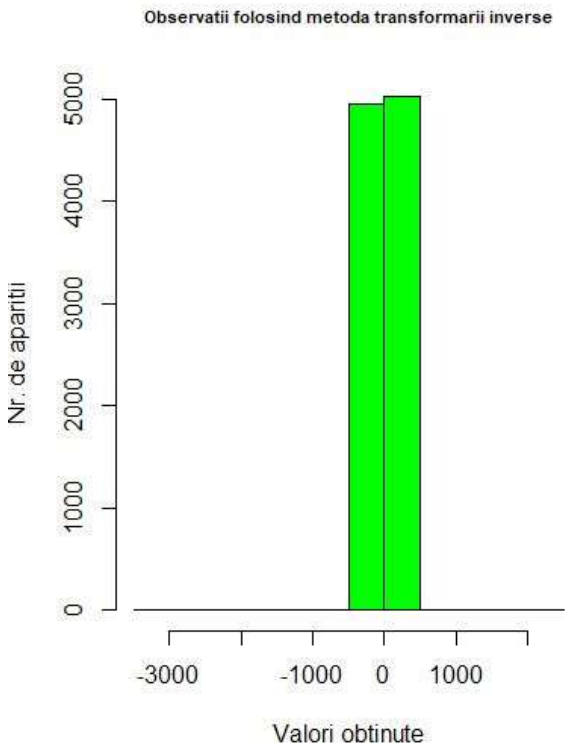
```
n=10000

valRepLogistica=function(nr) {

  u=runif(nr) #10000 de observatii dintr-o uniforma
  return (log(u/(1-u))) #inversa functiei de repartitie
}
test1=valRepLogistica(n)
test2=rlogis(n)

par(mfrow=c(1,2)) #afisam 2 grafice pe o linie

hist(test1,
     main="Observatii folosind metoda transformarii inverse",
     xlab="Valori obtinute",
     ylab="Nr. de aparitii",
     xlim=c(-10, 10),
     cex.main=0.7,
     col="blue");
hist(test2,
     main="Observatii folosind rlogis",
     xlab="Valori obtinute",
     ylab="Nr. de aparitii",
     xlim=c(-10, 10),
     cex.main=0.7,
     col="red");
```

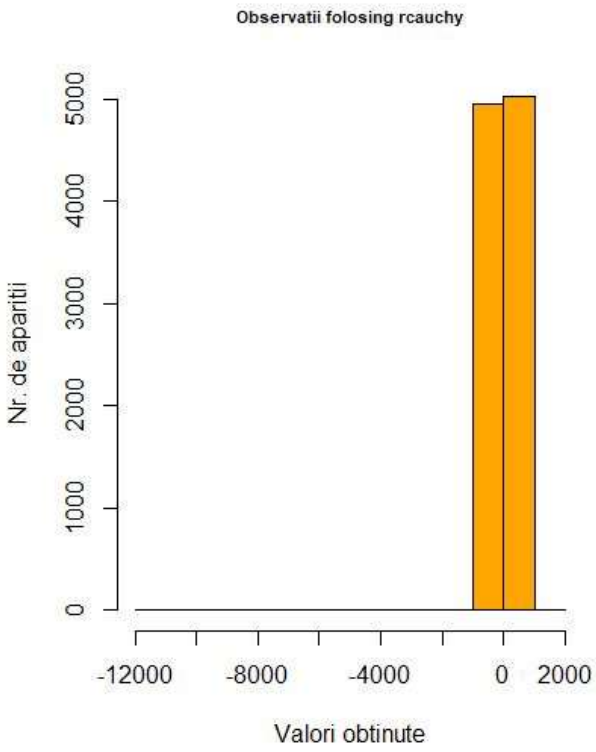




b) De asemenea, funcția de repartiție este continua, având inversa $f(y)=\beta*\tan(\pi*(u-1/2))+\mu$, unde $\beta=1$, $\mu=0$.

```
n=10000

valorRepCauchy=function(nr){

  u=runif(nr) #10000 de observatii dintr-o uniforma
  return(tan(pi*(u-1/2))) #inversa functiei date
}

test1=valorRepCauchy(n) #observatii metoda inversei
test2=rcauchy(n,0,1) #observatii metoda rcauchy

par(mfrow=c(1,2)) #afisam 2 grafice pe o linie

hist(test1,
     main="Observatii folosind metoda transformarii inverse",
     xlab="Valori obtinute",
     ylab="Nr. de aparitii",
     cex.main=0.7,
     col="green");

hist(test2,
     main="Observatii folosing rcauchy",
     xlab="Valori obtinute",
     ylab="Nr. de aparitii",
     cex.main=0.7,
     col="orange");
```





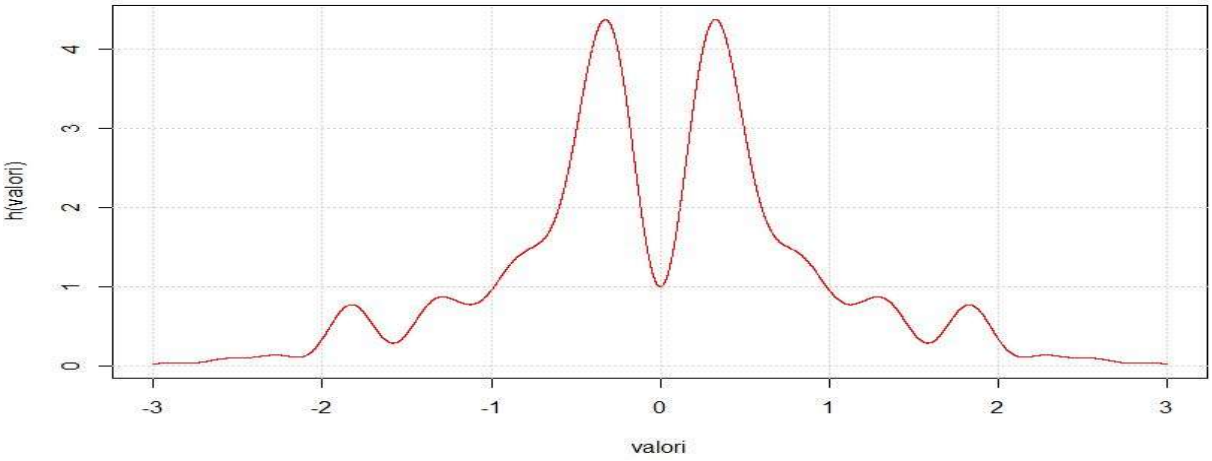
Exercițiul 2:

Am notat expresia $e^{(-x^2)/2}[\sin(6x)^2+3\cos(x)^2\sin(4x)^2+1]$ cu h(x). Considerăm f(x) proporțională cu h(x). Deci, f(x)=C*h(x).

Pentru a determina constanta de normalizare, generăm mai întâi observații sub aria determinată de h(x).

Graficul lui h(x):

```
h=function(x){
  return (exp(-x^2/2)*(sin(6*x)^2+3*(cos(x^2))^2*(sin(4*x))^2+1))
}
valori=seq(-3,3,0.0005)
plot(valori, h(valori), type="l", col="red")
grid(nx=NULL, col="lightgray", lty="dotted",lwd=par("lwd"),equilogs=TRUE)
```



Pentru a găsi constanta M, trebuie să studiem maximul funcției dată de raportul:

$\frac{h(x)}{g(x)}$=$\sqrt{2\pi}$[sin(6x)^2+3cos(x)^{2}sin(4x)^2+1]

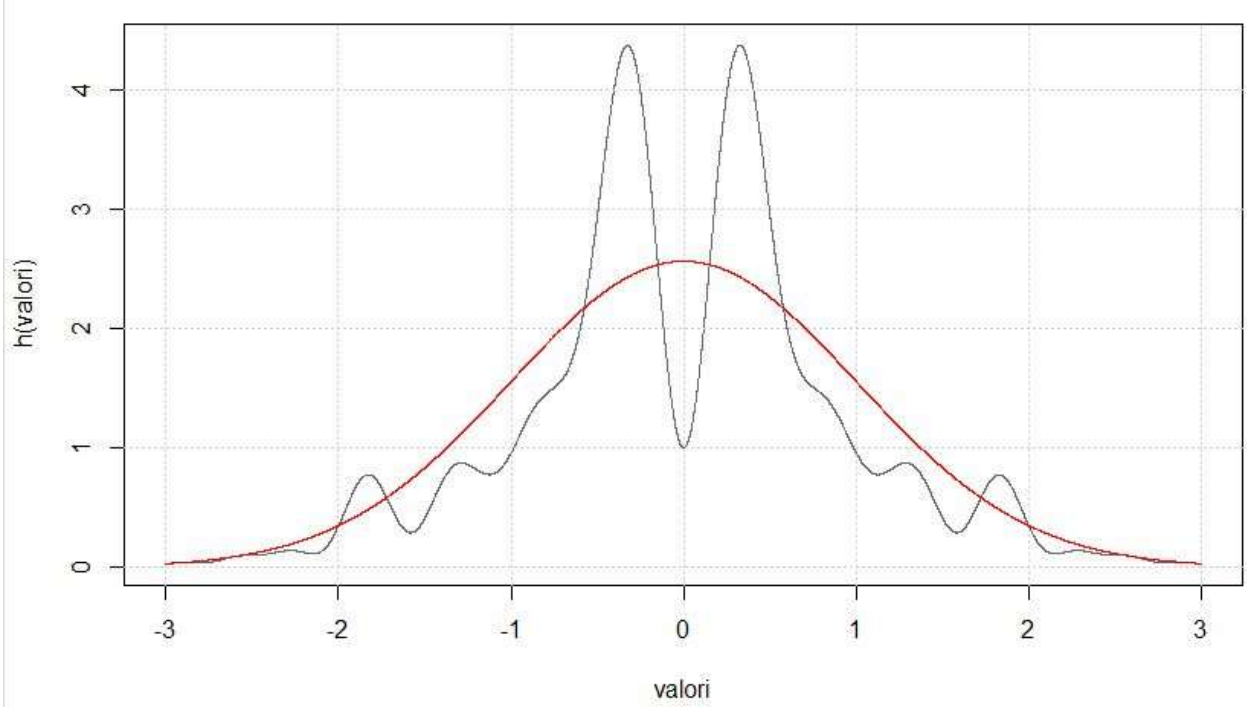
În acest sens, vom folosi funcția predefinită OPTIMISE, așa cum ni s-a sugerat.

```
raport = function(x) {
  return ( sqrt(2*pi) * ( sin(6*x)^2 + 3*(cos(x))^2 * (sin(4*x))^2 + 1) )
}
M = optimise(raport, c(-0.3,0), maximum = TRUE)
M[2]
```

$`objective`
[1] 10.84551

Ne vom folosi de valoarea găsită pentru M ca să mărginim graficul funcției h(x).

```
raport = function(x) {
  return ( sqrt(2*pi) * ( sin(6*x^2) + 3*cos(x^2) * sin(4*x^2) + 1) )
}
M=optimise(raport, c(-0.3,0), maximum = TRUE)
valori = seq(-3,3, 0.0005)
plot(valori, h(valori), type="l", col = "dimgray")
grid(nx = NULL, col = "lightgray", lty = "dotted",
     lwd = par("lwd"), equilogs = TRUE)
lines(valori, dnorm(valori)*M[[2]], col = "red")
```



Urmează să ne folosim de metoda respingerii și să păstrăm valorile acceptate.

```
valoriRetinute = c()
n = 2500 #numar de incercari
contor = 0 #numaram incercarile bune
i = 1
while( i <= n ) {
  u = runif(1,0,1) #generez o observatie din uniforma
  x = rnorm(1,0,1) #generez o observatie din normala standard
  if( u <= h(x)/(M[[2]]*dnorm(x))) {
    valoriRetinute[contor] = x
    contor = contor + 1
  }
  i=i + 1
}
#Astfel procentul de valori obtinute este:
p = contor/n
p
```

[1] 0.8056

După cum am notat, $\int h(x)=\int \frac{f(x)}{C}$. Cum $\int f(x)=1$ și $\int h(x)=\int \frac{f(x)}{c}$, atunci pM=$\frac{1}{C}$, deci C=$\frac{1}{pM}$.

```
M[[2]]*p #integrala lui h
```

[1] 5.18215

```
integrate(h, -Inf, Inf)
```

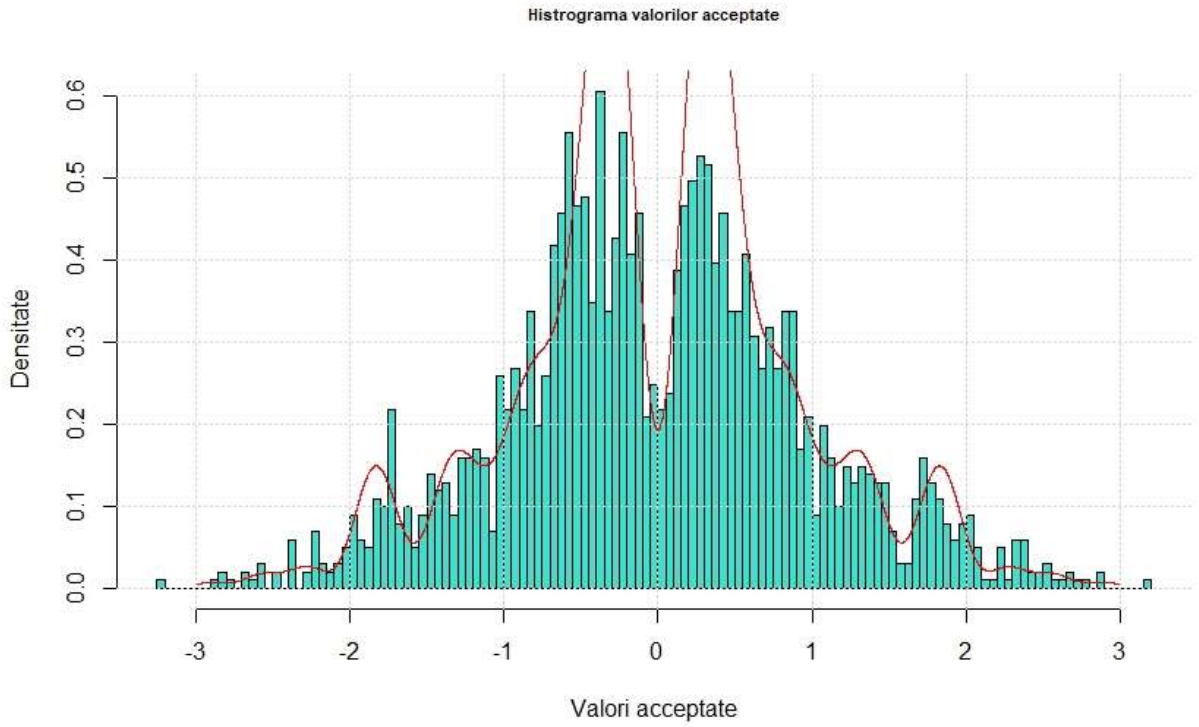
6.252033 with absolute error < 0.00037

```
normF = 1/(M[[2]]*p) #calcuez valoarea aproximativa a constantei
normF
```

[1] 0.1929701

Știind constanta C, putem afla f(x).

```
hist(valoriRetinute, breaks = 100, freq = FALSE, col = "turquoise",
     xlab = "Valori acceptate",
     ylab = "Densitate",
     main = "Histograma valorilor acceptate",
     cex.main = 0.7) #histograma valorilor acceptate
lines(valori, normF*h(valori), col = "red", type="l") #graficul functiei normalizate
grid(nx = NULL, col = "lightgray", lty = "dotted",
     lwd = par("lwd"), equilogs = TRUE)
```



Histograma valorilor obținute nu se așează perfect sub graficul funcției normalizate, deoarece nu am generat observații sub graficul lui h(x).

Exercițiul 3:

Scris analitic:

$$\int_0^1 h(x) = \int_0^1 \cos^2(50x) + \sin^2(20x) + 2\sin(20x)\cos(50x)dx$$

$$\int_0^1 \cos^2(50x) = \int_0^1 \frac{cos(100x)+1}{2}dx = \frac{x}{2} + \frac{1}{200}\sin(100x)\Big|_0^1 = \frac{1}{200}(100 + \sin(100)) \approx 0.49$$

$$\int_0^1 \sin^2(20x) = \int_0^1 \frac{1-\cos(40x)}{2} = \frac{x}{2} - \frac{1}{80}\sin(40x)\Big|_0^1 = \frac{1}{2} - \frac{\sin(40)}{80} \approx 0.49$$

$$\int_0^1 2\sin(20x)cos(50x) = 2\Big(\frac{1}{60}\cos(30x) - \frac{1}{140}\cos(70x)\Big)\Big|_0^1 \approx -0.02$$

$$\Rightarrow \int_0^1 h(x) \approx 0.96$$

Aplicând metoda Monte-Carlo în R vom obține:

```
nrUnif = 2000
funMonteCarlo = function(x) {
  return ( (cos(50*x) + sin(20*x))^2 ) #functia de la care plecam(vrem sa o integram)
}
Sn = 0#toate observatiile
valInti = c() #integrala calculata pentru i observatii (i se afla intre 1 si n)
valInti[0] = 0
for( i in 1:nrUnif) {
  Xn = runif(1,0,1) #generez o observatie
  Sn = Sn + funMonteCarlo(Xn) #fac suma, adunand h(Xn), unde h este funtia de mai devreme
  valInti[i] = Sn/i
}
valIntMC = Sn/nrUnif #valoarea integralei
valIntMC
```

[1] 0.9647803

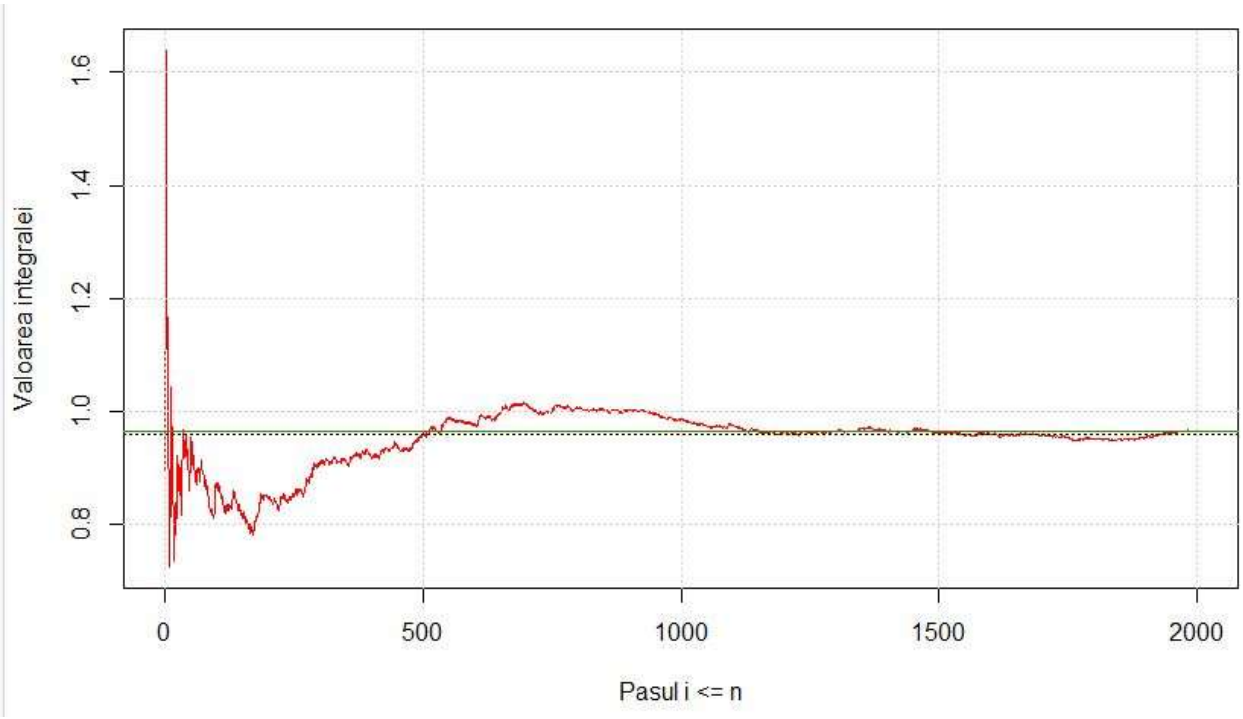
Folosindu-ne de INTEGRATE, obținem:

```
valR = integrate(funMonteCarlo, 0, 1)
valR
```

0.9652009 with absolute error < 1.9e-10

Pentru a ne da seama de eficiența acestei metode, reprezentăm grafic șirul de aproximări alături de valorile obținute anterior.

```
plot(valInti, type = "l", col = "red",
     xlab = "Pasul i <= n",
     ylab = "Valoarea integralei")
grid(nx = NULL, col = "lightgray", lty = "dotted",
     lwd = par("lwd"), equilogs = TRUE)
abline(h = valR[[1]], lty = "solid", col = "chartreuse4") #valoarea data de integrate()
abline(h = 0.96, lty = "dotted") #valoarea analitica
```



Exercițiul 4:

Pentru acest exercițiu am ales setul de date **"mtcars"** disponibil in seturile de date predefinite in R. Aceste înregistrări reprezintă o listă de mașini alături de care găsim o listă de parametri atât de design, cât și de performanță pentru autovehiculele respective.

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

În cele ce urmează, voi încerca să fac o analiză a datelor prezente în tabel și să interpretez pe cât posibil datele obținute:

a) Media

```
> apply(mtcars,2,mean)
      mpg       cyl      disp        hp      drat        wt      qsec        vs        am      gear      carb
 20.090625  6.187500 230.721875 146.687500  3.596563  3.217250 17.848750  0.437500  0.406250  3.687500  2.812500
```

Ni se afișează valoarea medie pentru fiecare dintre parametrii prezenți în tabel.

b) Suma

Se va afișa suma tuturor parametrilor pentru fiecare mașină în parte, însă nu am putut identifica vreo importanță pentru analiza respectivului rezultat.

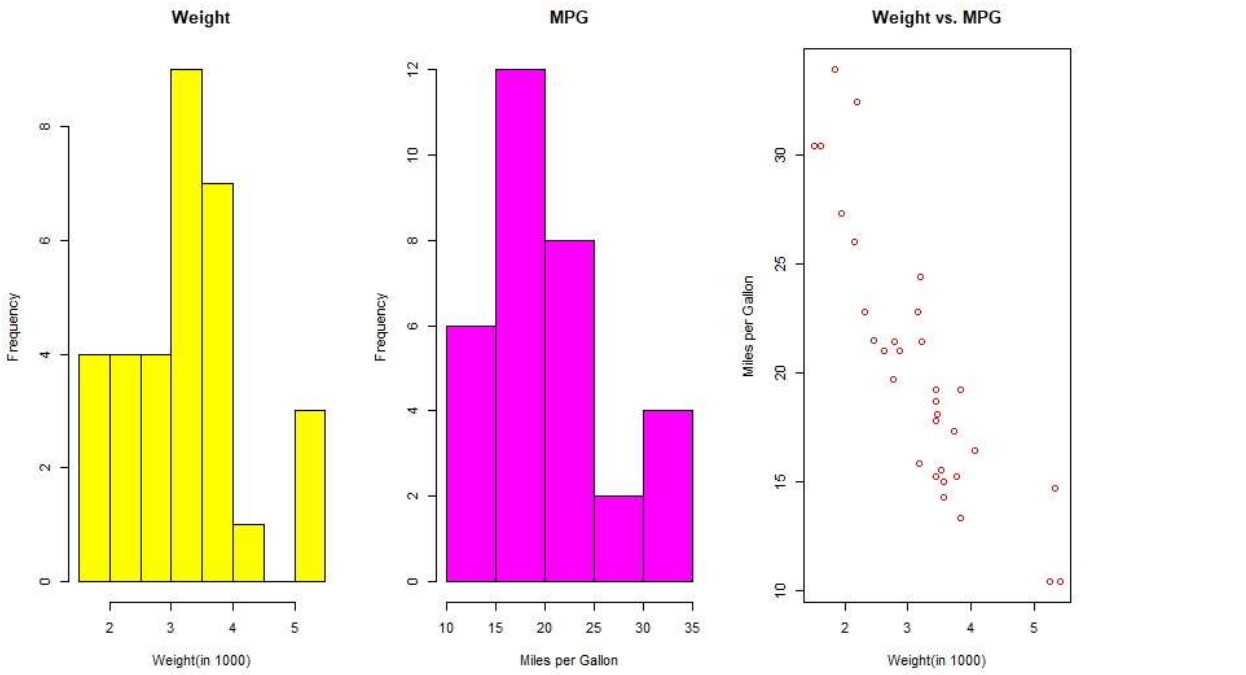
c) Cuantilele

```
> apply(mtcars,2,quantile,probs=c(0,0.25,0.50,0.75,1),na.rm=TRUE)
        mpg cyl    disp    hp  drat      wt    qsec vs am gear carb
0%   10.400   4  71.100  52.0 2.760 1.51300 14.5000  0  0    3    1
25%  15.425   4 120.825  96.5 3.080 2.58125 16.8925  0  0    3    2
50%  19.200   6 196.300 123.0 3.695 3.32500 17.7100  0  0    4    2
75%  22.800   8 326.000 180.0 3.920 3.61000 18.9000  1  1    4    4
100% 33.900   8 472.000 335.0 4.930 5.42400 22.9000  1  1    5    8
```

Minimul, prima cuantilă, a doua, a treia și maximul.

d) Histograme și comparația dintre ele

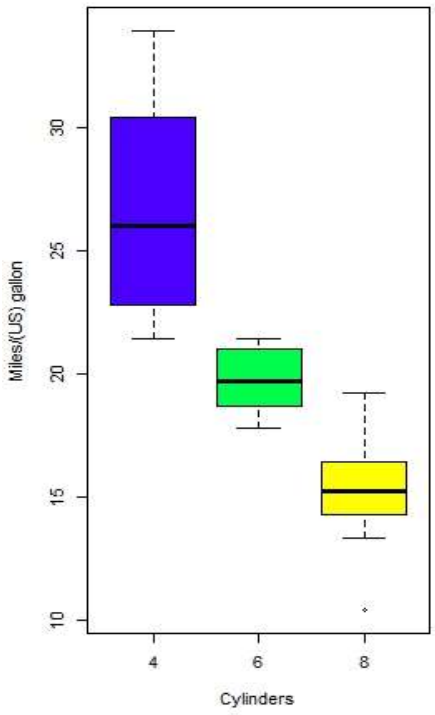
```
> par(mfrow=c(1,3))
> hist(mtcars$wt,main="Weight",xlab="Weight(in 1000)",col="yellow")
> hist(mtcars$mpg,main="MPG",xlab="Miles per Gallon",col="magenta")
> plot(mtcars$wt, mtcars$mpg, main="Weight vs. MPG", xlab= "Weight(in 1000)", ylab= "Miles per Gallon",col="red")
```



Primele două histograme analizează frecvența aparițiilor unor diverși parametri (în particular, Weight și MPG), iar cel de-al treilea plot realizează o analiză comparativă între cele două histograme.

e) Boxplot

```
boxplot(mpg~cyl,
        xlab="Cylinders", ylab="Miles/(US) gallon",
        col=topo.colors(3))
```



După ce am studiat boxplotul comparativ, am ajuns la concluzia că mașinile în 4 cilindri au consumul cel mai scăzut, deoarece parcurg mai multe mile cu un singur galon de combustibil, iar mediana este cea mai mare.