```
#Cateva problema interesante
#1. Problema cu vineri 13
fullyears <- 2000:2399
months <- 1:12
testthirteenth <- NULL
for(i in 1:length(fullyears)) {
  for(j in 1:12){
    testthirteenth <- c(testthirteenth,weekdays(as.Date(paste(
      fullyears[i],"/", months[j],"/13",sep=""),"%Y/%m/%d")))
  }
}
#functia paste face concatenare
#functia as.Date face conversie la tipul Date

t2 <- table(testthirteenth) #tabel de frecventa

#testthirteenth
#TEMA: Puteti eficientiza codul? --for-ul trebuie eliminat
fullyears <- 2000:2399
months <- 1:12
dates <- as.Date(paste(rep(fullyears, each = 12), rep(months, times = length(fullyears)), "13", sep = "/"), "%Y/%m/%d")
testthirteenth <- weekdays(dates)

t2 <- table(testthirteenth)

#Intr-o clasa sunt 24 de studenti.
#I. Generam esantionul de date de nastere:
#exploram functia sample
#consola: ? sample
#prototipul functiei: sample(x, size, replace = FALSE, prob = NULL)
#Pentru reproductibilitatea rezultatelor folosim functia set.seed()
set.seed(4)
sample(1:10,5)
#sample(1:10,200) - asta da eroare
sample(1:10,200,replace=T)

set.seed(6)
zi <- sample(1:31,24,replace=T)
luna <- sample(1:12,24,replace=T)
an <- sample(2002:2003,24,replace=T)
zi_nastere <- as.Date(paste(
  an[1],"/", luna[1],"/",zi[1],sep=""));
for(i in 1:24) {
   zi_nastere <- c(zi_nastere,as.Date(paste(an[i],"/", luna[i],"/",zi[i],sep="")))
}
zi_nastere
#A. Se alege un student la intamplare
  #a. Calculati probabilitatea ca ziua sa de nastere sa pice intr-o joi
  #din setul de date de nastere calculam cate pica intr-o joi si impartim la nr total de zile de nastere
  zi_saptamana <- weekdays(zi_nastere)
  prob <- length(zi_saptamana[zi_saptamana=="Thursday"])/length(zi_nastere)
  #b. Calculati probabilitatea ca ziua sa de nastere sa pice intr-o joi stiind ca nu e nascut intr-o marti.

#B Se aleg 2 studenti la intamplare
  #a. Calculati probabilitatea ca ambii studenti sa fie nascuti in aceeasi zi a saptamanii
  #b. Calculati probabilitatea ca cei doi studenti sa fie nascuti in zile consecutive ale saptamanii
#RESTUL E TEMA


#2. #Afisati valoarea factorialului pentru numerele de la 1 la 20
factorial(5)
  f <- factorial(1:20)
#sau
f <- sapply(1:20, factorial)

x <- list(a = 1:10, beta = exp(-3:3), logic = c(TRUE,FALSE,FALSE,TRUE))
x1 <- list(1:10, exp(-3:3), c(TRUE,FALSE,FALSE,TRUE))

# compute the list mean for each list element
lapply(x, mean)
lapply(x1, mean)
sapply(x1,mean)
# median and quantiles for each list element

v <- sample(1:300,20)
quantile(v,1/4)
quantile(v,3/4)

lapply(x, quantile, probs = 1:3/4)
sapply(x, quantile)

#3. Avem n bile numerotate de la 1 la n pe care le punem, in mod aleator in n urne disponibile.
#Cu ce probabilitate fiecare urna contine o bila?
#n! deoarece bilele sunt numerotate!!!
#


n <- 1:20
prob_n <- factorial(n)/n^n
plot(n,prob_n,type="h") #,main="Tralalala"
title("Probabilitatea ca toate urnele sa contina o bila")

#Tipurile din plot

#implicit type="p" <- puncte
plot(1:5,1:5,type="p")
#type="l" <- linii
plot(1:5,1:5,type="l")
#type="b" <- both, atat punctele cat si liniile dintre ele
plot(1:5,1:5,type="b")
#type="o" <- overlay, pune linii peste puncte
```

```
plot(1:5,1:5,type="o")
#type="h" <- deseneaza bare verticale
plot(1:5,1:5,type="h")
#type="n" <- nu deseneaza nimic, doar axele
plot(1:5,1:5,type="n")

# TEMA: Avem n bile numerotate de la 1 la n pe care le punem, in mod aleator, in N urne disponibile.
#Cu ce probabilitate prima urna contine k bile?
k <- c(2, 5, 10, 20, 30, 40, 50)

probdiff <- sapply(k, function(kk) prod(365:(365 - kk + 1)) / (365 ^ kk))
probat2lafel <- 1 - probdiff

plot(k, probat2lafel, xlab = "Numarul de studenti", ylab = "Probabilitatea zilei de nastere", col = "green", type = "l")
lines(k, probdiff, col = "red", type = "l")
legend(10, 1, "Zilele de nastere nu coincid", box.lty = NULL)
legend(30, 0.7, "Zilele de nastere coincid", box.lty = NULL)
title("Problema zilelor de nastere")
abline(v = 23)
abline(h = 0.5)

#4 Problema zilelor de nastere
#Avem un grup de k persoane si 365 de zile de nastere posibile.
#Determinati probabilitatea ca toate persoanele sa aiba zile de nastere distincte.
#Determinati probabilitatea ca cel putin 2 persoane sa aiba aceeasi data de nastere.
k <- c(2,5,10,20,30,40,50)
probdiff <- c()
probat2lafel <- c()
for(i in 1:length(k)) {
  kk <- k[i]
  probdiff[i] <- prod(365:(365-kk+1))/(365 ^kk)
  probat2lafel[i] <- 1- prod(365:(365-kk+1))/(365 ^ kk)
}
plot(k,probat2lafel,xlab="Numarul de studenti",ylab="Probabilitatea zilei de nastere",col="green","l")
lines(k,probdiff,col="red","l")
legend(10,1,"Zilele de nastere nu coincid",box.lty=NULL)
legend(30,.7,"Zilele de nastere coincid",,box.lty=NULL)
title("Problema zilelor de nastere")
abline(v=23)
abline(h=0.5)
#Tema: de eficientizat acest cod
k <- c(2, 5, 10, 20, 30, 40, 50)

probdiff <- sapply(k, function(kk) prod(365:(365 - kk + 1)) / (365 ^ kk))
probat2lafel <- 1 - probdiff

plot(k, probat2lafel, xlab = "Numarul de studenti", ylab = "Probabilitatea zilei de nastere", col = "green", type = "l")
lines(k, probdiff, col = "red", type = "l")
legend(10, 1, "Zilele de nastere nu coincid", box.lty = NULL)
legend(30, 0.7, "Zilele de nastere coincid", box.lty = NULL)
title("Problema zilelor de nastere")
abline(v = 23)
abline(h = 0.5)


#Functia choose calculeaza combinari de n luate cate k
x <- choose(20,3)
y <- choose(10,2)
z <- choose(30,5)

a <- choose(20,0)
b <- choose(10,5)
c <- choose(20,1)
d <- choose(20,1)
e <- choose(20,4)
f <- choose(10,2)
x*y/z
```