

```
#####
##### Lezione 3 - ITIS Marconi (Verona) #####
##### Marco Minozzo - 19 marzo 2018 #####
#####

# N.B. cercare di non utilizzare caratteri accentati nei commenti
# per evitare di dover scegliere un "encoding" per il salvataggio

# controllare la working directory e cambiarla se necessario

getwd()

# creare un file dove salvare quanto scritto nel pannello sorgente

# controllare i packages caricati e caricare quelli che serviranno

search()

# ".GlobalEnv"          "tools:rstudio"          "package:RevoUtils"
# "package:stats"       "package:graphics"       "package:grDevices"
# "package:utils"       "package:datasets"       "package:RevoUtilsMath"
# "package:methods"     "Autoloads"          "package:base"

loadedNamespaces()

# "compiler"    "RevoUtils"    "graphics"    "tools"    "utils"
# "grDevices"   "stats"        "datasets"    "methods"   "base"
# "RevoUtilsMath"

# library(ggplot2)

# in R sono caricati un certo numero di data set di prova

library(help = "datasets")
data()

##### Esercizio 4 (probability integral transform) #####
prove <- 100000; lambda <- 1.0; u <- 0; y <- 0

set.seed(1325)
t0 <- proc.time()
for (i in 1:prove) {
  u[i] <- runif(1, min=0, max=1)
  y[i] <- -(1/lambda)*log(1-u[i])
}
t1 <- proc.time() - t0; t1

##### alternativamente, definendo una nostra funzione #####
# miaformula <- function(ufun, lambdafun) {-(1/lambdafun)*log(1-ufun)}
#
# t0 <- proc.time()
# for (i in 1:prove) {
#   u[i] <- runif(1, min=0, max=1)
#   y[i] <- miaformula(u[i], lambda)
#   # y[i] <- miaformula(lambdafun=lambda, ufun=u[i])
#   # questo e' un modo alternativo di chiamare la funzione
# }
# t1 <- proc.time() - t0; t1
```

```
#####

umedia <- mean(u); umedia
ymedia <- mean(y); ymedia
yvar <- var(y)*((prove-1)/prove); yvar
# R fornisce la varianza campionaria (dividendo per n-1)
ysqm <- sqrt(yvar); ysqm

hist(u, 20)
hist(u, 20, prob=TRUE)
plot(ecdf(u),do.points=FALSE,verticals=TRUE)

max(y)
hist(y, seq(0, 13, 0.5), prob=TRUE)
lines(density(y,bw=0.1))
rug(y)

plot(ecdf(y),do.points=FALSE,verticals=TRUE)
x <- seq(0, 13, 0.05)
lines(x, pexp(x, rate=lambda), lty=3)
#####

#### come definire un operatore binario #####
"%marco%" <- function(x,y) { (x+y)^2 }
x <- c(1,2,3,4)
y <- c(4,3,5,2)
x %marco% y
#####

#### distribuzione geometrica #####
k<- 0:40
p<-dgeom(k,1/8)
plot(k,p,type="h",lwd=3)
xg <- rgeom(10000, 1/8)
min(xg); max(xg)
hist(xg, seq(-0.5,100.5,1), prob=TRUE)
#####

#### Esercizio 8 (incasso giornaliero) #####
prove <- 100000; prob <- 0.03; lambda <- 0.04; incasso <- 0

t0 <- proc.time()
for (i in 1:prove) {
  numclienti <- rgeom(1, prob) + 1
  scontrini <- rexp(numclienti, lambda)
  incasso[i] <- sum(scontrini)
}
t1 <- proc.time() - t0; t1

incassomedio <- mean(incasso); incassomedio
varincasso <- var(incasso)*((prove-1)/prove); varincasso
# R fornisce la varianza campionaria (dividendo per n-1)
sqmincasso <- sqrt(varincasso); sqmincasso

incasso500 <- incasso>500;
sum(incasso500)
prob500 <- sum(incasso500)/prove; prob500
```

```

hist(incasso, 20)
hist(incasso, 20, prob=TRUE)

max(incasso)
hist(incasso, seq(0,9500,500), prob=TRUE)
lines(density(incasso,bw=100))
rug(incasso)

plot(ecdf(incasso),do.points=FALSE,verticals=TRUE)

x <- seq(0, 8000, 50)
lines(x, pexp(x, rate=0.004), lty=3)
lines(x, pexp(x, rate=1/mean(incasso)), lty=3)
#####

#### Esercizio 7 (Bonus Malus) #####
prove <- 100000; sinistro <- c("NO","SI"); BManno3 <- 0;
premioanno3 <- 0; premi2anni <- 0; rimborsi <- 0; differenza <- 0
# lambda <- 0.006 # parametro della distribuzione esponenziale

t0 <- proc.time()
for (i in 1:prove) {
  annol <- sample(sinistro, size=1, prob=c(0.99,0.01))
  if (annol=="NO")
    anno2 <- sample(sinistro, size=1, prob=c(0.99,0.01))
  if (annol=="SI")
    anno2 <- sample(sinistro, size=1, prob=c(0.99,0.01))

BManno3[i]<-10+2*(annol=="SI")-(annol=="NO")+2*(anno2=="SI")-
(anno2=="NO")
premioanno3[i] <- 10*BManno3[i]
premi2anni[i] <- 100+10*(10+2*(annol=="SI")-(annol=="NO"))

  rimborsi[i] <- runif(1,min=0,max=10000)*(annol=="SI")
  +runif(1,min=0,max=10000)*(anno2=="SI") # uniforme
# rimborsi[i] <- rexp(1,rate=lambda)*(annol=="SI")
# +rexp(1,rate=lambda)*(anno2=="SI") # esponenziale

  differenza[i] <- premi2anni[i] - rimborsi[i]
}
t1 <- proc.time() - t0; t1

differenzamedia <- mean(differenza); differenzamedia
vardifferenza <- var(differenza)*((prove-1)/prove); vardifferenza
# R fornisce la varianza campionaria (dividendo per n-1)
sqmdifferenza <- sqrt(vardifferenza); sqmdifferenza

hist(differenza, 20)

quantile(differenza,probs=c(0, 0.0001, 0.001, 0.01, 0.0195, 0.02, 0.5,
1))

sum(differenza < 0); prove
probperdita <- sum(differenza < 0)/prove; probperdita

sum(differenza < -1000); prove
probperdital000 <- sum(differenza < -1000)/prove; probperdital000

```

```
hist(differenza, 20)
hist(differenza, 20, prob=TRUE)
#####
```