LABORATORIO 6 - Modelli

STATISTICA E LABORATORIO (CDL in INTERNET OF THINGS, BIG DATA, MACHINE LEARNING)

Anno Accademico 2022-2023

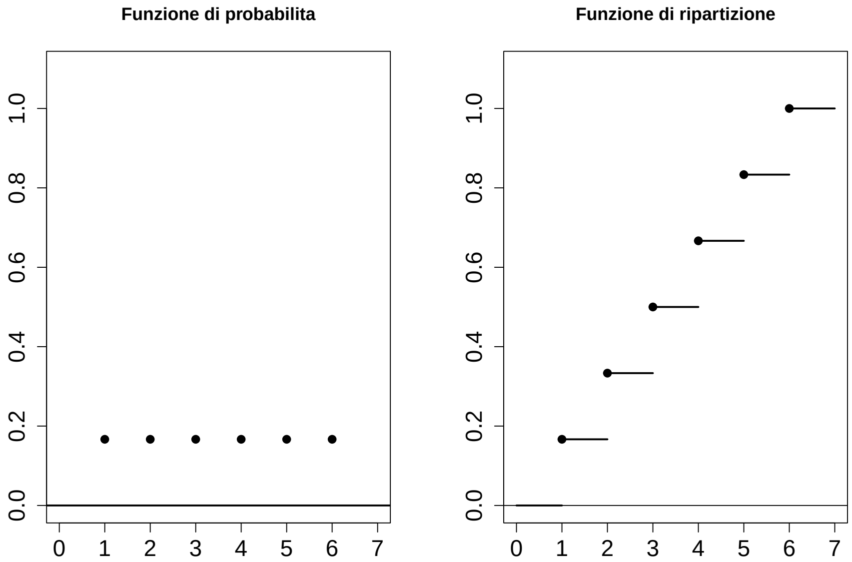
Section 1

Modelli discreti

Dado

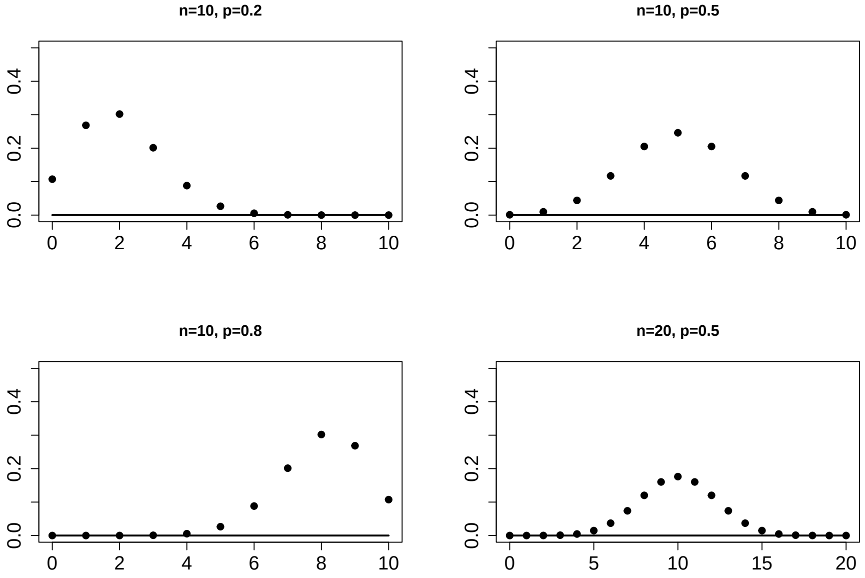
Si consideri il lancio di un dado regolare. La variabile casuale X, che indica la faccia uscita dopo il lancio, ha distribuzione di probabilità Ud(6).

```
par(mfrow=c(1,2))
xx <- c(1:6)
plot(xx,rep(1/6,6),pch=19,xlim=c(0,7),ylim=c(0,1.1),lwd=2,
     cex.axis=1.5, xlab=" ",ylab=" ",main = "Funzione di
     probabilita")
abline(0,0,lwd=2)
plot(xx,c(1,2,3,4,5,6)/6,pch=19,xlim=c(0,7),ylim=c(0,1.1),lwd=2,
     cex.axis=1.5, xlab=" ",ylab=" ",main = "Funzione di
     ripartizione")
segments(0,0,1,0,lwd=2)
segments(1,1/6,2,1/6,lwd=2)
segments(2,2/6,3,2/6,lwd=2)
segments(3,3/6,4,3/6,lwd=2)
segments(4,4/6,5,4/6,lwd=2)
segments(5,5/6,6,5/6,lwd=2)
segments(6,6/6,7,6/6,lwd=2)
```


Funzione di probabilita
0.0
0.2
0.4
0.6
0.8
1.0
0 1 2 3 4 5 6 7
Funzione di ripartizione
0.0
0.2
0.4
0.6
0.8
1.0
0 1 2 3 4 5 6 7

Modello binomiale

```
par(mfrow=c(2,2))
xx <- seq(0,10,1)
# dbinom(x,size,prob) fornisce i valori della funzione
# di probabilita' in x di una Bin(size,prob)
plot(xx,dbinom(xx,10,0.2),pch=19,ylim=c(0,0.5),lwd=2,cex.axis=1.5,
     xlab=" ",ylab=" ",main="n=10, p=0.2")
segments(0,0,10,0,lwd=2)
plot(xx,dbinom(xx,10,0.5),pch=19,ylim=c(0,0.5),lwd=2,cex.axis=1.5,
     xlab=" ",ylab=" ",main="n=10, p=0.5")
segments(0,0,10,0,lwd=2)
plot(xx,dbinom(xx,10,0.8),pch=19,ylim=c(0,0.5),lwd=2,cex.axis=1.5,
     xlab=" ",ylab=" ",main="n=10, p=0.8")
segments(0,0,10,0,lwd=2)
xx<-seq(0,20,1)
plot(xx,dbinom(xx,20,0.5),pch=19,ylim=c(0,0.5),lwd=2,cex.axis=1.5,
     xlab=" ",ylab=" ",main="n=20, p=0.5")
segments(0,0,20,0,lwd=2)
```


n=10, p=0.2
n=10, p=0.5
n=10, p=0.8
n=20, p=0.5
0.0
0.2
0.4
0
2
4
6
8
10
5
15
20

Atleti

Tra i 100 iscritti ad una associazione sportiva ci sono 30 più alti di 180 cm. Si estrae casualmente un campione di $n = 10$ atleti con reinserimento. La variabile casuale X che definisce il numero di atleti che, tra i 10 considerati, è più alto di 180 cm (successo) ha distribuzione Bi(10; 0.3). La probabilità di estrarre almeno un atleta più alto di 180 cm è

```
# dbinom(x,size,prob) fornisce i valori della funzione
# di probabilita' in x di una Bin(size,prob)
1-dbinom(0,10,0.3) # P(X>=1)
```

```
## [1] 0.9717525
```

La probabilità di estrarre due atleti più alti di 180 cm è

```
dbinom(2,10,0.3) # P(X=2)
```

```
## [1] 0.2334744
```

Infine, la probabilità di estrarne meno di 4 è

```
dbinom(0,10,0.3)+dbinom(1,10,0.3)+dbinom(2,10,0.3)+dbinom(3,10,0.3)
```

```
## [1] 0.6496107
```

```
# pbinom(q,size,prob) fornisce i valori della funzione
# di ripartizione in q di una Bin(size,prob)
pbinom(2,10,0.3) # P(X<=2)
```

```
## [1] 0.3827828
```

```
# qbinom(p,size,prob) fornisce il quantile di livello p
# di una Bin(size,prob)
qbinom(0.5,10,0.3) # mediana
```

```
## [1] 3
```

```
# rbinom(n,size,prob) simula n osservazioni da una Bin(size,prob)
rbinom(4,10,0.3)
```

```
## [1] 4 1 1 3
```

```
# per avere sempre gli stessi valori simulati e' necessario
# fissare il seed della procedura di simulazione
set.seed(11)
rbinom(4,10,0.3)
```

```
## [1] 2 0 3 0
```

Monitor

Per un inconveniente nella linea di produzione, su 100 monitor prodotti da una certa azienda 10 risultano difettosi. Un rivenditore ha, recentemente, acquistato cinquanta monitor da questa azienda. La variabile casuale X che descrive il numero di monitor che, tra i cinquanta venduti, verranno resi alla casa produttrice perché difettosi è una Bi(50; 0.1). La probabilità che nessun monitor sia difettoso è

```
dbinom(0,50,0.1) # P(x=0)
```

```
## [1] 0.005153775
```

```
1-pbinom(3.,50,0.1) # P(X>3)
```

```
## [1] 0.7497061
```

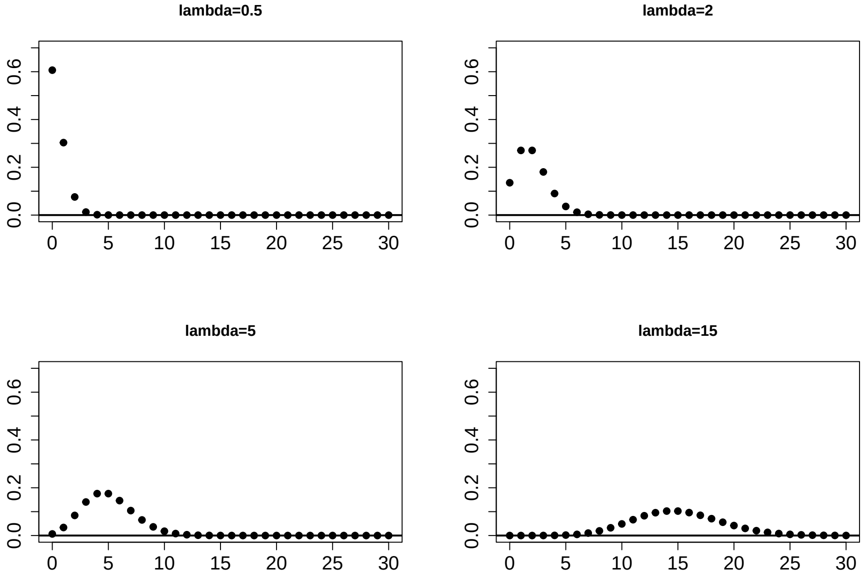
```
qbinom(c(0.25,0.5,0.75),50,0.1) # quartili
```

```
## [1] 3 5 6
```

Modello poisson

```
par(mfrow=c(2,2))
xx<-seq(0,30,1)
# dpois(x,lambda) fornisce i valori della funzione di
# probabilita' in x di una P(lambda)
plot(xx,dpois(xx,0.5),pch=19,ylim=c(0,0.7),lwd=2,cex.axis=1.5,
     xlab=" ",ylab=" ",main="lambda=0.5")
abline(0,0,lwd=2)
plot(xx,dpois(xx,2),pch=19,ylim=c(0,0.7),lwd=2,cex.axis=1.5,
     xlab=" ",ylab=" ",main="lambda=2")
abline(0,0,lwd=2)
plot(xx,dpois(xx,5),pch=19,ylim=c(0,0.7),lwd=2,cex.axis=1.5,
     xlab=" ",ylab=" ",main="lambda=5")
abline(0,0,lwd=2)
plot(xx,dpois(xx,15),pch=19,ylim=c(0,0.7),lwd=2,cex.axis=1.5,
     xlab=" ",ylab=" ",main="lambda=15")
abline(0,0,lwd=2)
```

```
par(mfrow=c(1,1))
```


lambda=0.5
lambda=2
lambda=5
lambda=15
0.0
0.2
0.4
0.6
0
5
10
15
20
25
30

Pronto soccorso

Pronto soccorso. Al Pronto Soccorso di un piccolo ospedale si presentano in media 3 pazienti ogni ora. Per predisporre il personale medico necessario, si vuole calcolare la probabilità che in un'ora arrivino esattamente 2 pazienti e la probabilità che in un'ora arrivino più di 2 pazienti.

```
# dpois(x,lambda) fornisce i valori della funzione di
#probabilita' in x di una P(lambda)
dpois(2,3) # P(X=2)
```

```
## [1] 0.2240418
```

```
# ppois(x,lambda) fornisce i valori della funzione di
# ripartizione in q di una P(lambda)
1-ppois(2,3) # P(X>2)=1-P(X<=2)
```

```
## [1] 0.5768099
```

```r
# qpois(p,lambda) fornisce il quantile di livello p di una P(lambda)
qpois(0.5,3) # mediana
```

```
## [1] 3
```

```r
qpois(c(0.25,0.5,0.75),3) # quartili
```

```
## [1] 2 3 4
```

```r
# rpois(n,lambda) simula n osservazioni
# da una P(lambda)
rpois(4,3)
```

```
## [1] 1 6 1 2
```

```r
# per avere sempre gli stessi valori simulati
# e' necessario fissare il seed della procedura di simulazione
set.seed(11)
rpois(4,3)
```

```
## [1] 2 0 3 0
```

Pezzi difettosi

Un certo macchinario produce un pezzo difettoso ogni cento. Si vuole calcolare la probabilità che, scegliendo a caso 100 pezzi, se ne trovino esattamente 3 difettosi.

```
dbinom(3,100,0.01) # P(X=3) esatta usando la Bi(100,0.01)
```

```
## [1] 0.06099917
```

```
n=100
p=0.01
lambda=n*p
lambda
```

```
## [1] 1
```

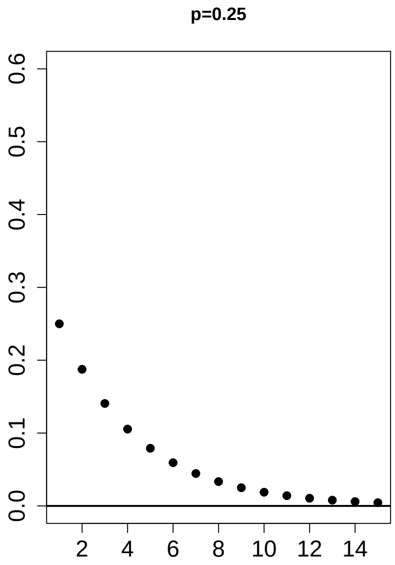
```
dpois(3,lambda) # P(X=3) approssimata usando la P(1)
```

```
## [1] 0.06131324
```

Modello geometrico

```
par(mfrow=c(1,2))
xx<-seq(1,15,1)
# dgeom(x,prob) fornisce i valori della funzione di
# probabilita' in x di una Ge(prob)
# si considera l'argomento x-1 perche' in R la
# distribuzione geometrica e' definita in modo equivalente
# considerando il numero di insuccessi (invece del
# numero di prove) prima del successo
plot(xx,dgeom(xx-1,0.25),pch=19,ylim=c(0,0.6),lwd=2,cex.axis=1.5,
     xlab=" ",ylab=" ",main="p=0.25")
abline(0,0,lwd=2)
plot(xx,dgeom(xx-1,0.5),pch=19,ylim=c(0,0.6),lwd=2,cex.axis=1.5,
     xlab=" ",ylab=" ",main="p=0.5")
abline(0,0,lwd=2)
```

```
par(mfrow=c(1,1))
```


p=0.25
0.0
0.1
0.2
0.3
0.4
0.5
0.6
2
4
6
8
10
12
14
p=0.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
2
4
6
8
10
12
14

Gioco del lotto

Si consideri il gioco del lotto. La probabilità che esca il tre in una singola estrazione su una ruota prefissata $1/18$. La variabile casuale X, che indica il numero di settimane necessarie affinché esca il numero tre sulla ruota di Napoli, ha distribuzione Ge($1/18$). Si calcola la probabilità che il tre esca alla trentesima settimana, se si è a conoscenza che non è uscito nelle prime dieci settimane.

```
# dgeom(x,prob) fornisce i valori della funzione di probabilita'
# in x di una Ge(prob)
# per quanto detto in precedenza si considera
#l'argomento x-1
# pgeom(q,prob) fornisce i valori della funzione di
# ripartizione in q di una Ge(prob)
# P(X=30|X>10)=P(X=30)/P(X>10)
dgeom(30-1,1/18)/(1-pgeom(10-1,1/18))
```

```
## [1] 0.01875337
```

```r
dgeom(20-1,1/18) # P(X=20)
```

```
## [1] 0.01875337
```

```r
# qgeom(p,prob) fornisce il quantile di livello p di una Ge(prob);
# per quanto detto in precedenza bisogna aggiungere 1
qgeom(0.5,1/18)+1 # mediana
```

```
## [1] 13
```

```r
qgeom(c(0.25,0.5,0.75),1/18)+1 # quartili
```

```
## [1]  6 13 25
```

```r
# rgeom(n,prob) simula n osservazioni da una Ge(prob);
# per quanto detto in precedenza bisogna aggiungere  1
rgeom(4,1/18)+1
```

```
## [1] 47 19 30  7
```

```r
# per avere sempre gli stessi valori simulati e'
# necessario fissare il seed della procedura di simulazione
set.seed(11)
rgeom(4,1/18)+1
```

```
## [1] 20 10 14  6
```

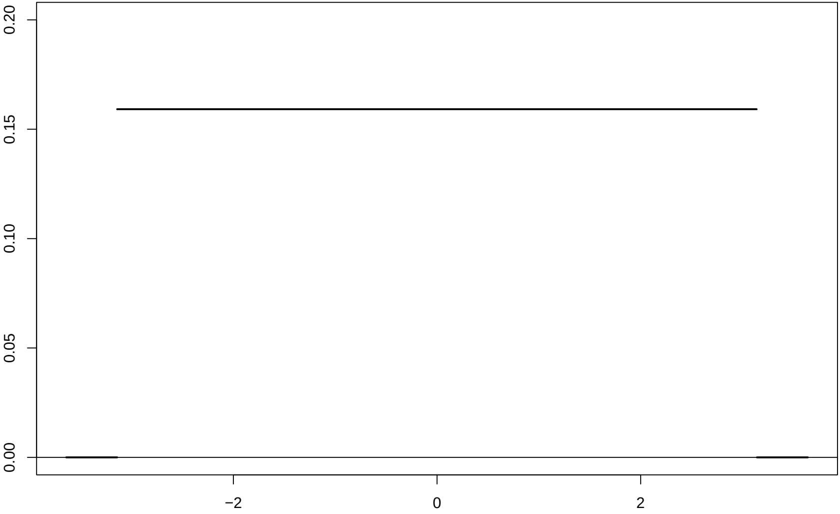
Section 2

Modelli continui

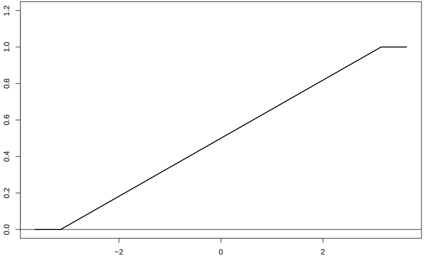
Criceti

Si predispone un esperimento per valutare il senso di orientamento dei criceti. Gli animali vengono posti al centro un contenitore circolare con un'unica via di uscita. Dopo averli bendati e disorientati, si osserva la direzione scelta da ciascun criceto. Sia $X \sim U(-\pi; \pi)$ la variabile casuale che esprime l'ampiezza in radianti dell'angolo tra la direzione scelta dall'animale e la direzione che porta all'uscita.

```
# dunif(x,min,max) fornisce i valori della funzione di
# densita' in x di una U(min,max)
x <- seq(-pi,pi,0.01)
plot(x,dunif(x,-pi,pi),type='l',lwd=2,xlim=c(-pi-0.5,pi+0.5),
     ylim=c(0,0.2),xlab="",ylab="")
abline(0,0,lwd=1)
segments(-0.5-pi,0,-pi,0,lwd=2)
segments(pi,0,pi+0.5,0,lwd=2)
```


0.00
0.05
0.10
0.15
0.20
−2
0
2

```r
# punif(x,min,max) fornisce i valori della funzione di ripartizione
# in q di una U(min,max)
x <- seq(-pi,pi,0.01)
plot(x,punif(x,-pi,pi),type='l',lwd=2,xlim=c(-pi-0.5,pi+0.5),
     ylim=c(0,1.2),xlab="",ylab="")
abline(0,0,lwd=1)
segments(-0.5-pi,0,-pi,0,lwd=2)
segments(pi,1,pi+0.5,1,lwd=2)
```



```r
punif(1,-pi,pi) # P(X<=1)
```

```
## [1] 0.6591549
```

```r
1-punif(1,-pi,pi) # P(X>1)
```

```
## [1] 0.3408451
```

```r
punif(1,-pi,pi)-punif(-1,-pi,pi) # P(-1<=X<=1)
```

```
## [1] 0.3183099
```

```r
# qunif(p,min,max) fornisce il quantile di livello p
# di una U(min,max)
qunif(0.5,-pi,pi) # mediana
```

```
## [1] 0
```

```r
qunif(c(0.25,0.5,0.75),-pi,pi) # quartili
```

```
## [1] -1.570796  0.000000  1.570796
```

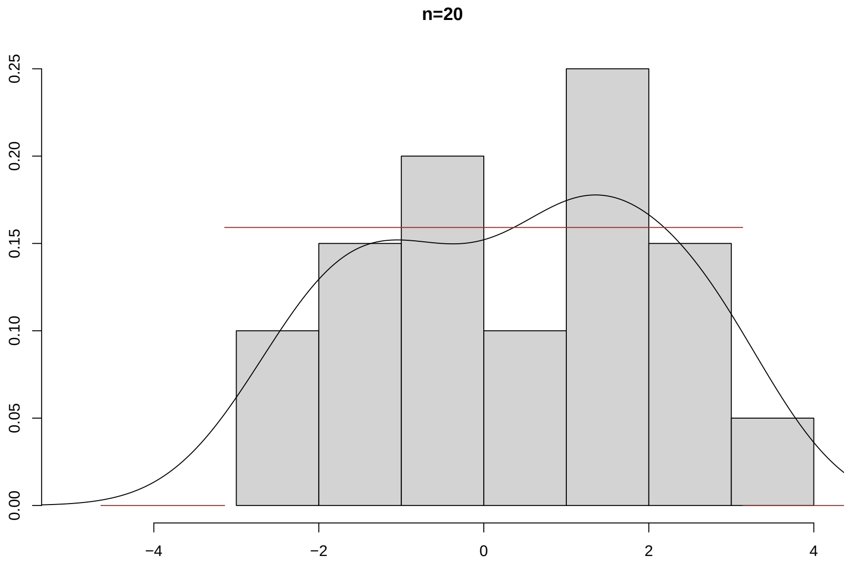
```r
# runif(n,min,max) simula n osservazioni da una
# U(min,max)
runif(4,-pi,pi)
```

```
## [1] -0.9402888 -2.7521078 -0.1067026 -0.6337014
```

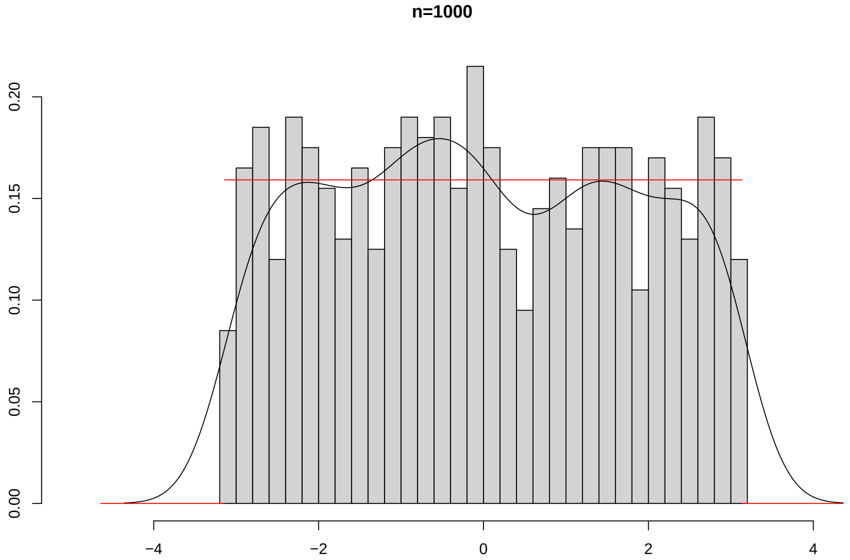
```r
# per avere sempre gli stessi valori simulati e'
# necessario fissare il seed della procedura di simulazione
set.seed(11)
runif(4,-pi,pi)
```

```
## [1] -1.39958082 -3.13833600  0.06665437 -3.05332704
```

```r
# si simulano n=20 osservazioni da una U(-pi,pi) e
# si rappresentano istogramma e stima della densita'
# in rosso la vera funzione di densita' U(-pi,pi)
set.seed(1)
x<-runif(20,-pi,pi)
hist(x,freq=FALSE,xlim=c(-5,4),main="n=20",xlab="",ylab="")
lines(density(x))
lines(seq(-pi,pi,0.01),dunif(seq(-pi,pi,0.01),-pi,pi),col='red')
segments(-1.5-pi,0,-pi,0,col='red')
segments(pi,0,pi+1.5,0,col='red')
```


n=20
0.00
0.05
0.10
0.15
0.20
0.25
−4
−2
0
2
4

```r
# si simulano n=1000 osservazioni da una U(-pi,pi) e
# si rappresentano istogramma e stima della densita'
# le stime sono molto piu' vicine alla vera funzione di densita'
# U(-pi,pi) (in rosso)
set.seed(1)
x<-runif(1000,-pi,pi)
hist(x,freq=FALSE,xlim=c(-5,4),breaks = 25,main="n=1000",xlab="",
     ylab="")
lines(density(x))
lines(seq(-pi,pi,0.01),dunif(seq(-pi,pi,0.01),-pi,pi),col='red')
segments(-1.5-pi,0,-pi,0,col='red')
segments(pi,0,pi+1.5,0,col='red')
```


n=1000
0.00
0.05
0.10
0.15
0.20
−4
−2
0
2
4

Circuito

Un circuito è costituito da due componenti dal funzionamento indipendente la cui vita operativa, misurata in anni, è descritta rispettivamente dalle variabili casuali $X_1 \sim Esp(0.2)$ e $X_2 \sim Esp(0.3)$. Si cerca la probabilità che funzionamento del circuito sia non superiore a 10 anni.

```
# dexp(x,rate) fornisce i valori della funzione di densita'
# in x di una Esp(rate)
# pexp(q,rate) fornisce i valori della funzione
# di ripartizione in q di una Esp(rate)
pexp(10,0.2)*pexp(10,0.3)
```

```
## [1] 0.8216156
```

```
# P(X_1<=10 e X_2<=10) (componenti in parallelo)
```

```r
# con l'opzione lower.tail=FALSE si ottiene P(X>q)=1-P(X<=q)=1-F(q)
1-pexp(10,0.2,lower.tail=FALSE)*pexp(10,0.3,lower.tail=FALSE)
```

```
## [1] 0.9932621
```

```r
# P(X_1<=10 o X_2<=10) (componenti in serie)

# in alternativa pexp(10,0.2)+pexp(10,0.3)-
#pexp(10,0.2)*pexp(10,0.3)
```

```r
# qexp(p,rate) fornisce il quantile di livello p di una Esp(rate)
qexp(0.5,0.2) # mediana
```

```
## [1] 3.465736
```

```r
qexp(c(0.25,0.5,0.75),0.2) # quartili
```

```
## [1] 1.438410 3.465736 6.931472
```

```r
qexp(0.5,0.3) # mediana
```

```
## [1] 2.310491
```

```r
qexp(c(0.25,0.5,0.75),0.3) # quartili
```

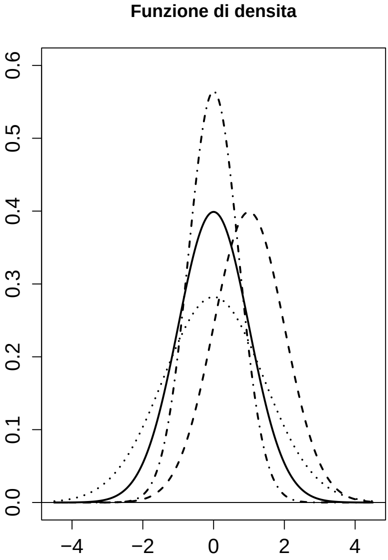
```
## [1] 0.9589402 2.3104906 4.6209812
```

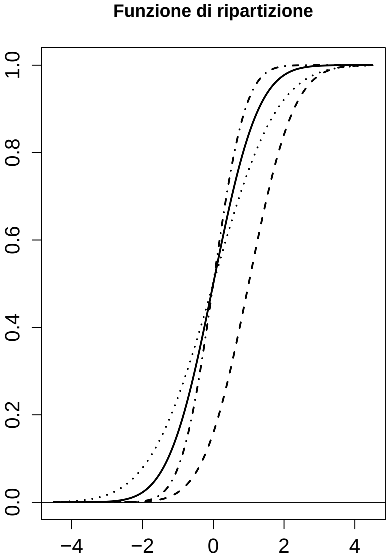
```r
# rexp(n,rate) simula n osservazioni da una Esp(rate)
rexp(4,0.2)
```

```
## [1] 0.3080879 1.8486090 6.1314038 0.1355105
```

Modello normale

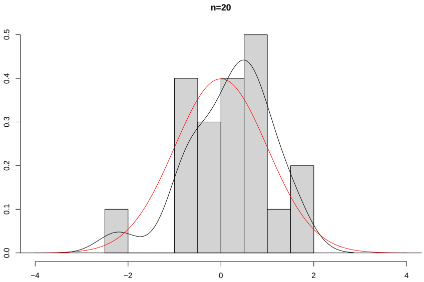
```
# dnorm(x,mean,sd) fornisce i valori della funzione di densita'
# in x di una N(mean,sd^2)
# pnorm(q,mean,sd) fornisce i valori della funzione di ripartizione
# in q di una N(mean,sd^2) si considera sd invece della varianza
par(mfrow=c(1,2))
xx<-seq(-4.5,4.5,0.001)
plot(xx,dnorm(xx,0,1),ylim=c(0,0.6),type='l',lwd=2,cex.axis=1.3,
     xlab="",ylab="",main="Funzione di densita")
lines(xx,dnorm(xx,1,1),lwd=2,lty=2)
lines(xx,dnorm(xx,0,sqrt(2)),lwd=2,lty=3)
lines(xx,dnorm(xx,0,sqrt(1/2)),lwd=2,lty=4)
abline(0,0,lwd=1)
plot(xx,pnorm(xx,0,1),ylim=c(0,1),type='l',lwd=2,cex.axis=1.3,
     xlab="",ylab="",main="Funzione di ripartizione")
lines(xx,pnorm(xx,1,1),lwd=2,lty=2)
lines(xx,pnorm(xx,0,sqrt(2)),lwd=2,lty=3)
lines(xx,pnorm(xx,0,sqrt(1/2)),lwd=2,lty=4)
abline(0,0,lwd=1)
```

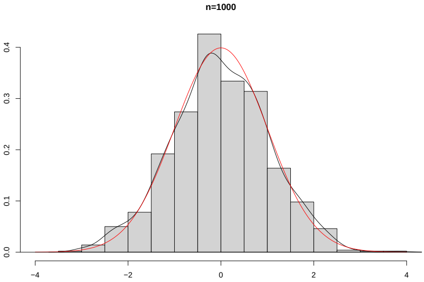

Funzione di densita
0.0
0.1
0.2
0.3
0.4
0.5
0.6
−4
−2
0
2
4


Funzione di ripartizione
0.0
0.2
0.4
0.6
0.8
1.0
−4
−2
0
2
4

```r
# si simulano n=20 osservazioni da una N(0,1) e
# si rappresentano istogramma e stima della densita'
# in rosso la vera funzione di densita' N(0,1)
set.seed(1)
x<-rnorm(20)
hist(x,freq=FALSE,xlim=c(-4,4),main="n=20",xlab="",ylab="")
lines(density(x))
lines(seq(-4,4,0.01),dnorm(seq(-4,4,0.01)),col='red')
abline(0,0,lwd=1)
```

```r
# si simulano n=1000 osservazioni da una N(0,1) e
# si rappresentano istogramma e stima della densita'
# le stime sono molto piu' vicine alla vera funzione di densita'
# N(0,1) (in rosso)
set.seed(1)
x<-rnorm(1000)
hist(x,freq=FALSE,xlim=c(-4,4),main="n=1000",xlab="",ylab="")
lines(density(x))
lines(seq(-4,4,0.01),dnorm(seq(-4,4,0.01)),col='red')
abline(0,0,lwd=1)
```


n=20
0.0
0.1
0.2
0.3
0.4
0.5
−4
−2
0
2
4


n=1000
0.0
0.1
0.2
0.3
0.4
−4
−2
0
2
4

Pressione

La variabile casuale X rappresenta la pressione sistolica, in mm di mercurio, di un generico individuo. Per la popolazione maschile italiana adulta si assume che $X \sim N(129; 392.04)$.

```
pnorm(135,129,sqrt(392.04)) # P(X<=135)
```

```
## [1] 0.6190666
```

```
pnorm(130,129,sqrt(392.04))-pnorm(120,129,sqrt(392.04)) # P(120<=X<=
```

```
## [1] 0.1954219
```

```
pnorm(129,129,sqrt(392.04),lower.tail=FALSE) # P(X>129)
```

```
## [1] 0.5
```

```
# P(120<=X<=150|X>129)
(pnorm(150,129,sqrt(392.04))-pnorm(120,129,sqrt(392.04)))/
  pnorm(129,129,sqrt(392.04),lower.tail=FALSE)
```

Barre

Per tagliare delle barre d'acciaio alla lunghezza nominale di 5 cm si utilizza un macchinario che fornisce barre con lunghezza $X \sim N(5.05; 0.01)$. Ad un successivo controllo di qualità, si scartano le barre che differiscono dalla lunghezza nominale per più di un mm. La probabilità che una generica barra soddisfi ai requisiti è

```
pnorm(5.1,5.05,sqrt(0.01))-pnorm(4.9,5.05,sqrt(0.01))
```

```
## [1] 0.6246553
```

```
# P(4.9<=X<=5.1) con X N(5.05,0.01)
```

Se fosse possibile ricalibrare la procedura di taglio in modo da avere $\mu = 5$, si avrebbe

```
pnorm(5.1,5,sqrt(0.01))-pnorm(4.9,5,sqrt(0.01)) # P(4.9<=X<=5.1) co
```

```
## [1] 0.6826895
```

Infine, se si aumenta anche la precisione dello strumento di modo che $\sigma = 0.05$, si ha

```
pnorm(5.1,5,sqrt(0.0025))-pnorm(4.9,5,sqrt(0.0025))
```

```
## [1] 0.9544997
```

```
# P(4.9<=X<=5.1) con X N(5,0.0025)
```

Riso

Un'industria alimentare confeziona pacchi di riso, con peso dichiarato pari a 500 gr, utilizzando un macchinario che fornisce pacchi con peso $X \sim N(500; 5)$. Si scelgono a caso 5 confezioni di riso e si vuole calcolare la probabilità che non ci siano pacchi con peso inferiore a quello dichiarato.

```
p = pnorm(500,500,sqrt(5)) # P(X<500)
dbinom(0,5,p)#P(Y=0) Y Bin(5,p)
```

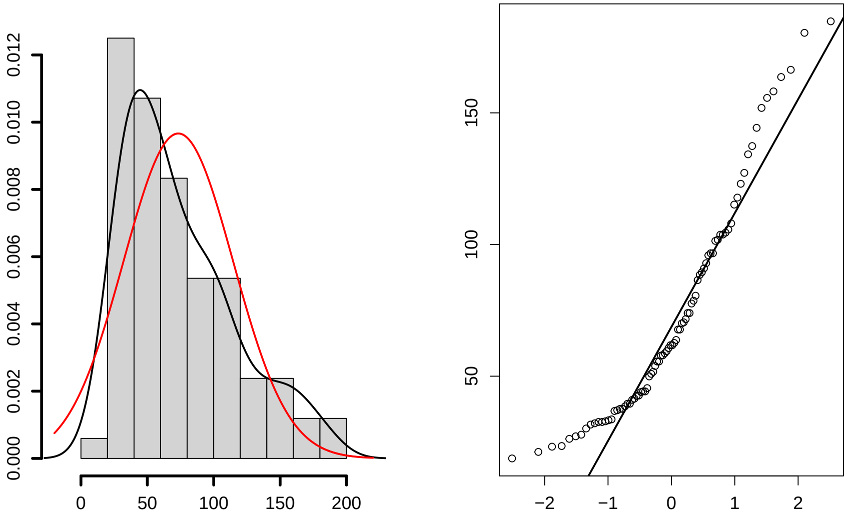
```
## [1] 0.03125
```

Cipolle

Si considerano $n = 84$ misurazioni sulla produzione (grammi per pianta) di una specie di cipolla bianca coltivata in Australia.

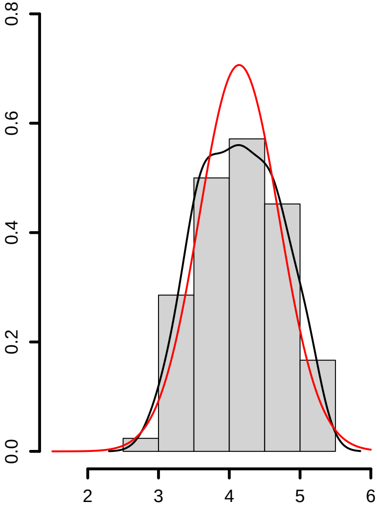
```
cipolle<-read.table("onions.dat",col.names=c("peso","dens","sede"))
par(mfrow=c(1,2))
hist(cipolle$peso,freq=F,lwd=3,xlab=" ",ylab=" ",xlim=c(-20,220),
     ylim=c(0,0.013),cex.axis=1.2,main=" ")
xx<-seq(-20,220,0.01)
lines(density(cipolle$peso),lwd=2)
lines(xx,dnorm(xx,mean(cipolle$peso),sqrt(var(cipolle$peso)*83/84)),
      lwd=2,col='red')
qqnorm(cipolle$peso,xlab=" ",ylab=" ",cex.axis=1.2,main=" ")
qqline(cipolle$peso,lwd=2)
```

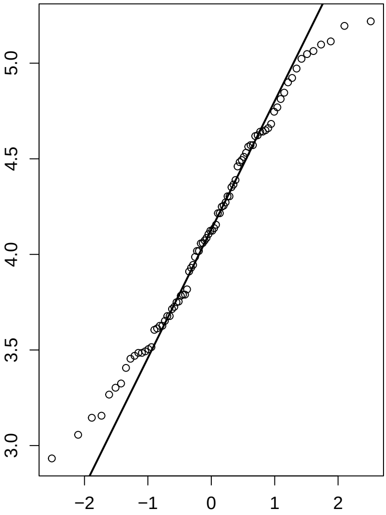
```
par(mfrow=c(1,1))
```


0.000
0.002
0.004
0.006
0.008
0.010
0.012
0
50
100
150
200
50
100
150
−2
−1
0
1
2

```
par(mfrow=c(1,2))
logpeso<-log(cipolle$peso)
hist(logpeso,freq=F,lwd=3,xlab=" ",ylab=" ",xlim=c(1.5,6),ylim=c(0,(
xx<-seq(1.5,6,0.01)
lines(density(logpeso),lwd=2)
lines(xx,dnorm(xx,mean(logpeso),sqrt(var(logpeso)*83/84)),lwd=2,col=
qqnorm(logpeso,xlab=" ",ylab=" ",cex.axis=1.2,main=" ")
qqline(logpeso,lwd=2)
```

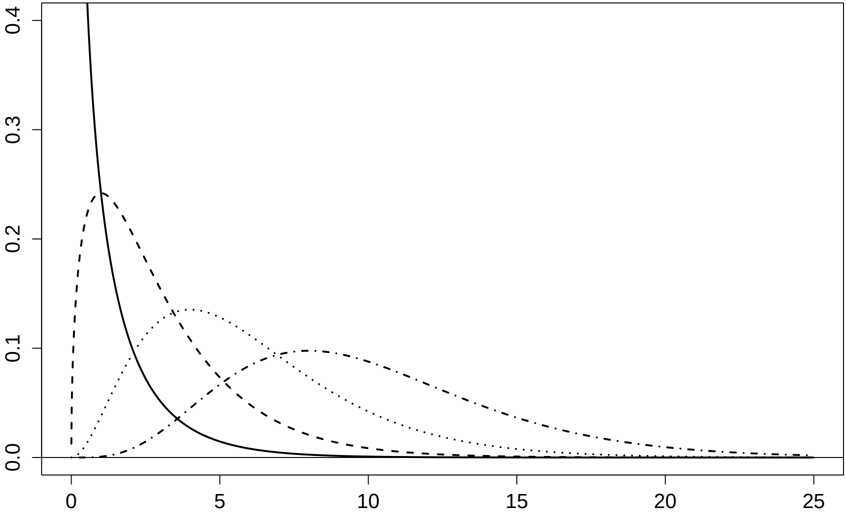
```
par(mfrow=c(1,1))
```


0.0
0.2
0.4
0.6
0.8
2
3
4
5
6


3.0
3.5
4.0
4.5
5.0
−2
−1
0
1
2

Modello chi-quadrato

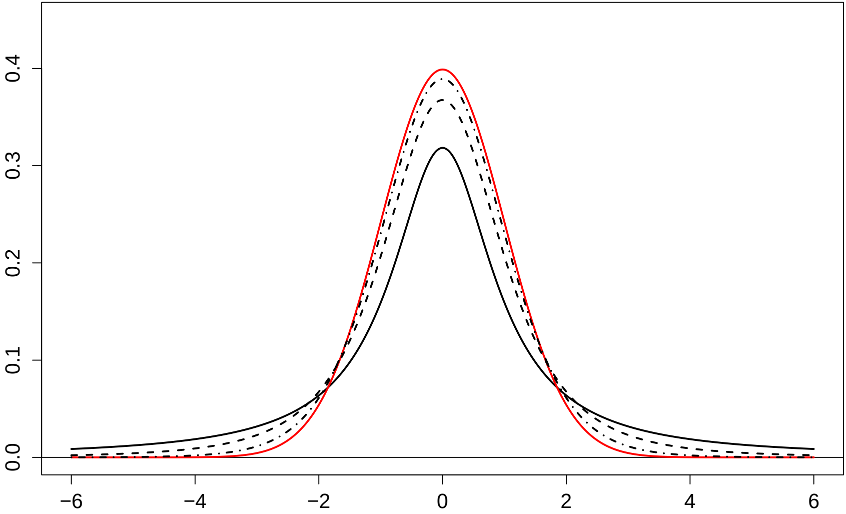
```
xx<-seq(0.001,25,0.01)
plot(xx,dchisq(xx,1),ylim=c(0,0.4),xlim=c(0,25),type='l',lwd=2,
     cex.axis=1.3,xlab=" ",ylab=" ")
lines(xx,dchisq(xx,3),lwd=2,lty=2)
lines(xx,dchisq(xx,6),lwd=2,lty=3)
lines(xx,dchisq(xx,10),lwd=2,lty=4)
abline(0,0,lwd=1)
```


0.0
0.1
0.2
0.3
0.4
0
5
10
15
20
25

##

Modello t di Student

```
xx<-seq(-6,6,0.01)
plot(xx,dt(xx,1),ylim=c(0,0.45),xlim=c(-6,6),type='l',lwd=2,
     cex.axis=1.3,xlab=" ",ylab=" ")
abline(0,0,lwd=1)
lines(xx,dnorm(xx),lwd=2,col='red')
lines(xx,dt(xx,3),lwd=2,lty=2)
lines(xx,dt(xx,10),lwd=2,lty=4)
```


0.0
0.1
0.2
0.3
0.4
−6
−4
−2
0
2
4
6

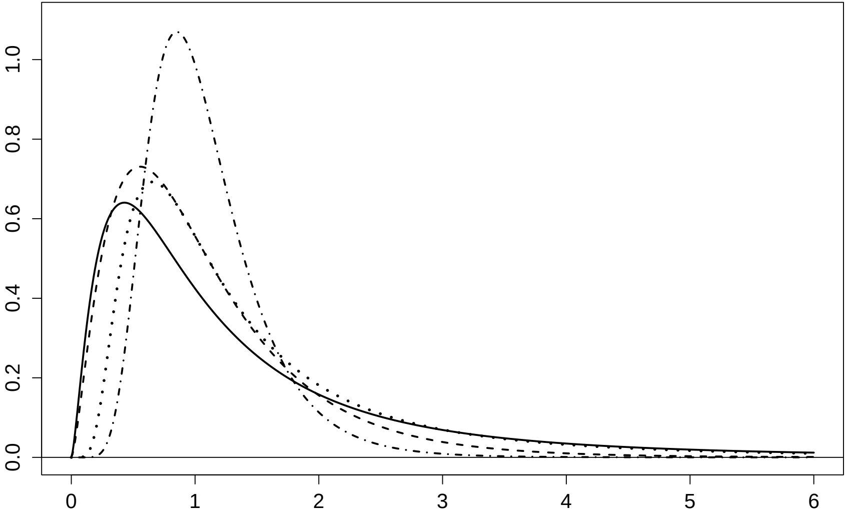
```r
# dt(x,df) fornisce i valori della funzione di densita'
# in x di una t(df)
# pt(q,df) fornisce i valori della funzione di
# ripartizione in q di una t(df)
# qt(p,df) fornisce il quantile di livello p di una t(df)
# rt(n,df) simula n osservazioni da una t(df)

# valori critici si una t(3)
qt(c(0.10,0.05,0.025,0.01,0.005,0.001),3,lower.tail=FALSE)
```

```
## [1]  1.637744  2.353363  3.182446  4.540703  5.840909 10.214532
```

Modello F di Fisher

```
xx<-seq(0,6,0.01)
plot(xx,df(xx,5,5),ylim=c(0,1.1),xlim=c(0,6),type='l',lwd=2,
     cex.axis=1.3,xlab=" ",ylab=" ")
abline(0,0,lwd=1)
lines(xx,df(xx,5,25),lwd=2,lty=2)
lines(xx,df(xx,25,5),lwd=3,lty=3)
lines(xx,df(xx,25,25),lwd=2,lty=4)
```


0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5
6

```
# df(x,df1,df2) fornisce i valori della funzione di densita'
# in x di una F(df1,df2)
# pf(q,df1,df2) fornisce i valori della funzione di ripartizione
# in q di una F(df1,df2)
# qf(p,df1,df2) fornisce il quantile di livello p
# di una F(df1,df2)
# rf(n,df1,df2) simula n osservazioni da una F(df1,df2)


# valori critici di una F(5,5)
qf(c(0.999,0.995,0.99,0.975,0.95,0.9,0.10,0.05,0.025,
     0.01,0.005,0.001),5,5,lower.tail=FALSE)
```

```
##  [1]  0.03361074  0.06693617  0.09118247  0.13993095  0.19800690
##  [7]  3.45298225  5.05032906  7.14638183 10.96702065 14.93960546
```