

# Esempio di Beta-Bimomiale—

*Fulvia Pennoni*

*20 Novembre 2017*

Nell'esempio del Bayes'billard delle dispense (Parte1\_Inferenza\_Bayesiana\_Pennoni\_17\_18.pdf) si suppone ad esempio un numero di palline blu pari pari a  $n = 10$  e  $p$  posizione delle palline blu e  $X$  la variabile casuale che identifica il numero di palline blu che si posizionano prima della pallina gialla. La densità marginale di  $X$  è un binomiale, mentre la densità a posteriori  $p|X = k$  è una Beta.

Si verifica il risultato precedente simulando il lancio delle palline tantissime volte:

Si simulano i valori di  $p$

```
nsim <- 10^6
p <- runif(nsim)
length(p)

## [1] 1000000
head(p)

## [1] 0.9224716 0.6213083 0.1839489 0.8968556 0.5447851 0.1295377
```

Si simulano lo stesso numero di valori per la distribuzione condizionata di  $X|p$  fissando  $n$

```
n <- 10
xcond <- rbinom(nsim,n,p)
head(xcond)

## [1] 9 6 2 9 7 1
```

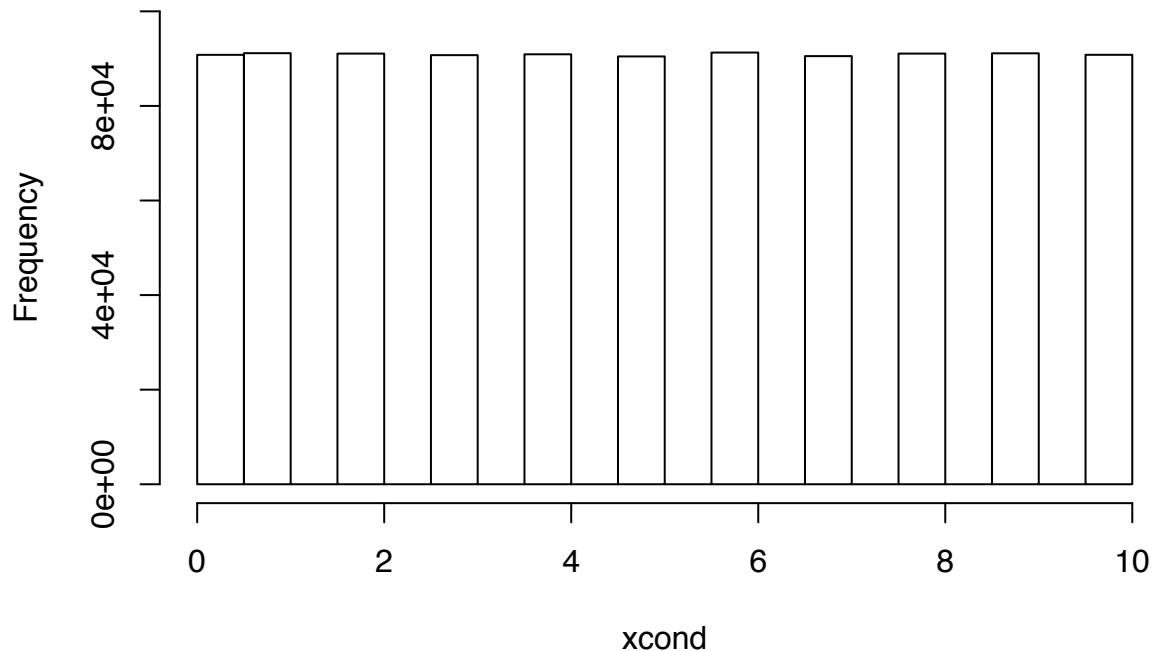
pertanto il primo elemento di xcond e' stato generato utilizzando il primo elemento di  $p$ , il secondo con il secondo elemento di  $p$  etc..

Nell'esempio visto a lezione la distribuzione marginale di  $X$  e' una distribuzione uniforme. Come dall'equazione illustrata nelle dispense  $P(X = k) = \frac{1}{n+1} = \frac{1}{11} = 0.09090909$  si nota che la distribuzione è assimilabile a quella di un'uniforme in  $(0,10)$  e che le densità sono proporzionali a  $\frac{1}{n+1}$

Per visualizzare:

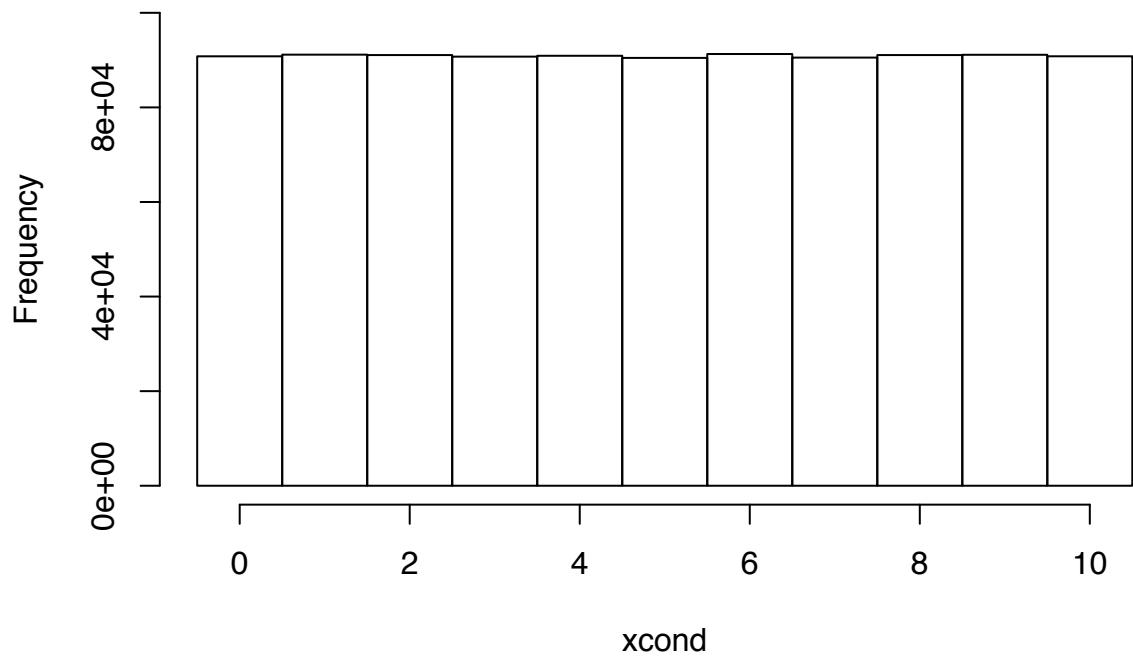
```
hist(xcond, ylim= c(0,100000))
```

### Histogram of xcond



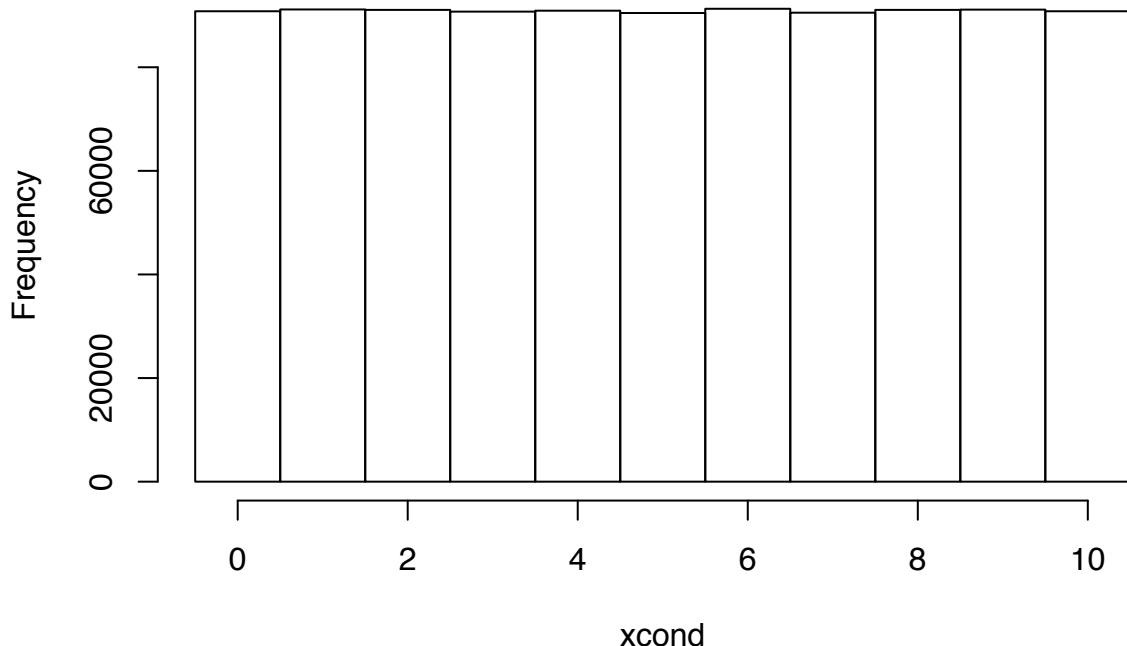
```
hist(xcond,
  breaks = seq(-.5,n+0.5,1),
  ylim= c(0,100000),
  main = "Dist. f(X|p)" )
```

### Dist. f(X|p)



```
h <- hist(xcond, breaks = seq(-.5,n+0.5,1),
           main = "Dist. f(X|p)")
```

**Dist. f(X|p)**



```
h$density
```

```
## [1] 0.090808 0.091154 0.091073 0.090741 0.090921 0.090470 0.091291
## [8] 0.090535 0.091071 0.091126 0.090810
```

dove nel secondo istogramma ogni asta e' centrata sull'intero.

Per cui si nota che la distribuzione potrebbe essere uniforme discreta.

Per rappresentare la distribuzione a posteriori di  $p$  dato  $X = 3$  si considerano ad esempio solo i valori realizzati di  $X$  il cui valore e' 3

```
ppost <- p[xcond==3]
length(ppost)
```

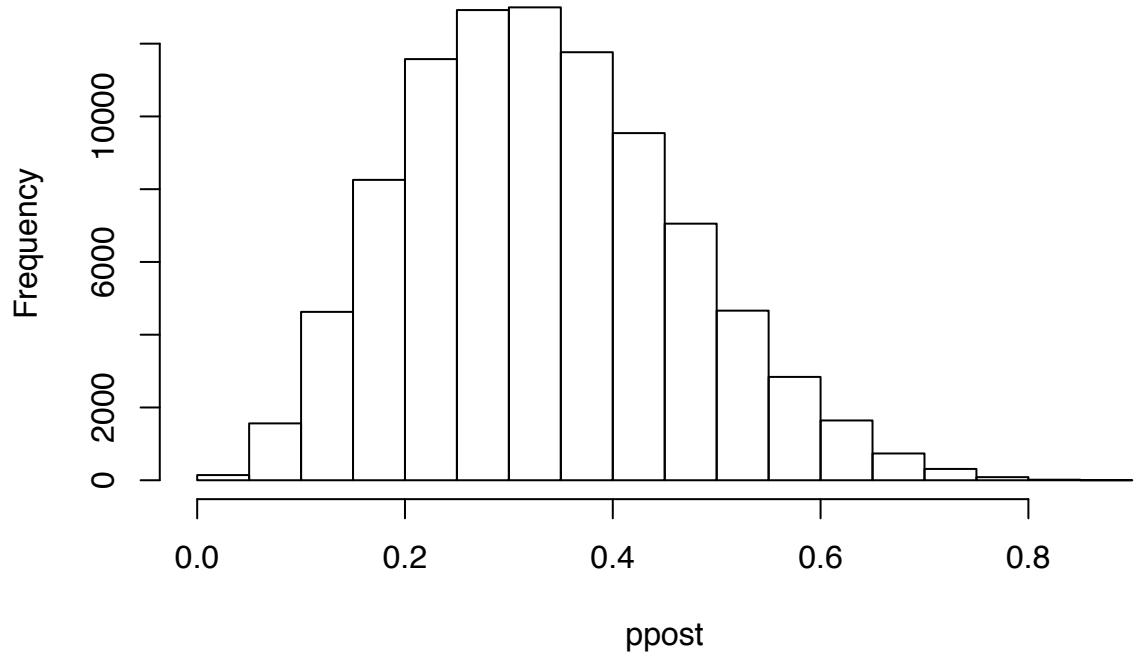
```
## [1] 90741
```

Come illustrato nell'esempio la distribuzione a posteriori e' una distribuzione  $Beta(\alpha + k, \beta + n - k)$  ponendo  $\alpha = 1$  e  $\beta = 1$  si ha  $Beta(4, 8)$  per  $k = 2$ .

Per i valori empirici generati in precedenza si ottiene:

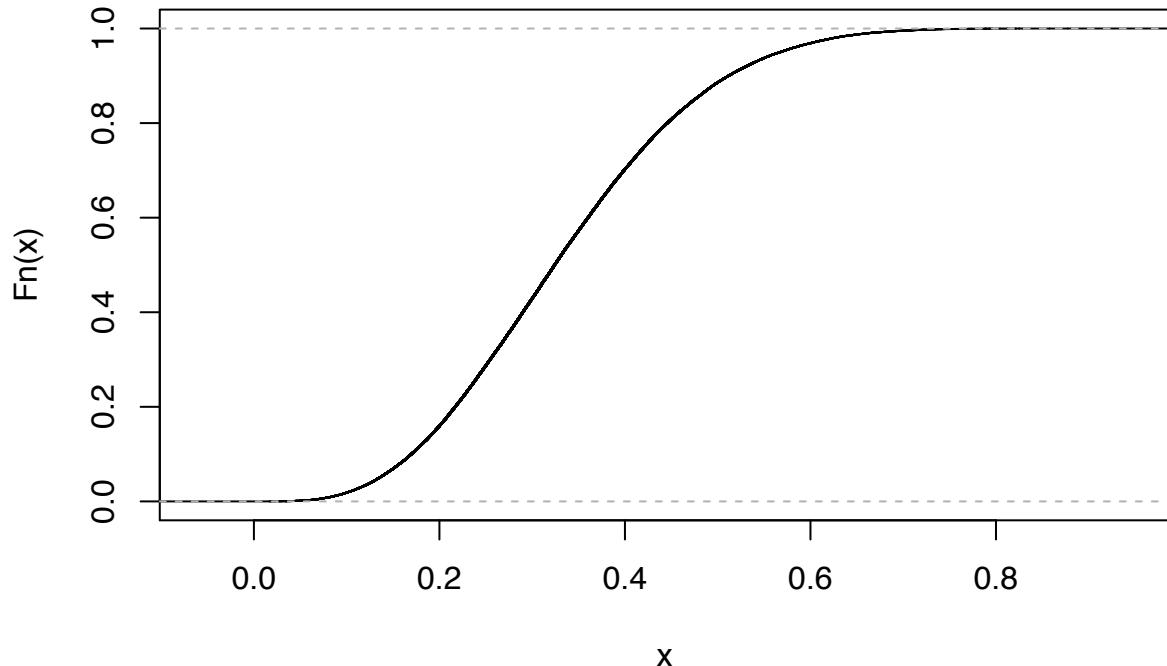
```
hist(ppost, main = "Dist. f(p|X=3)")
```

### Dist. f(p|X=3)



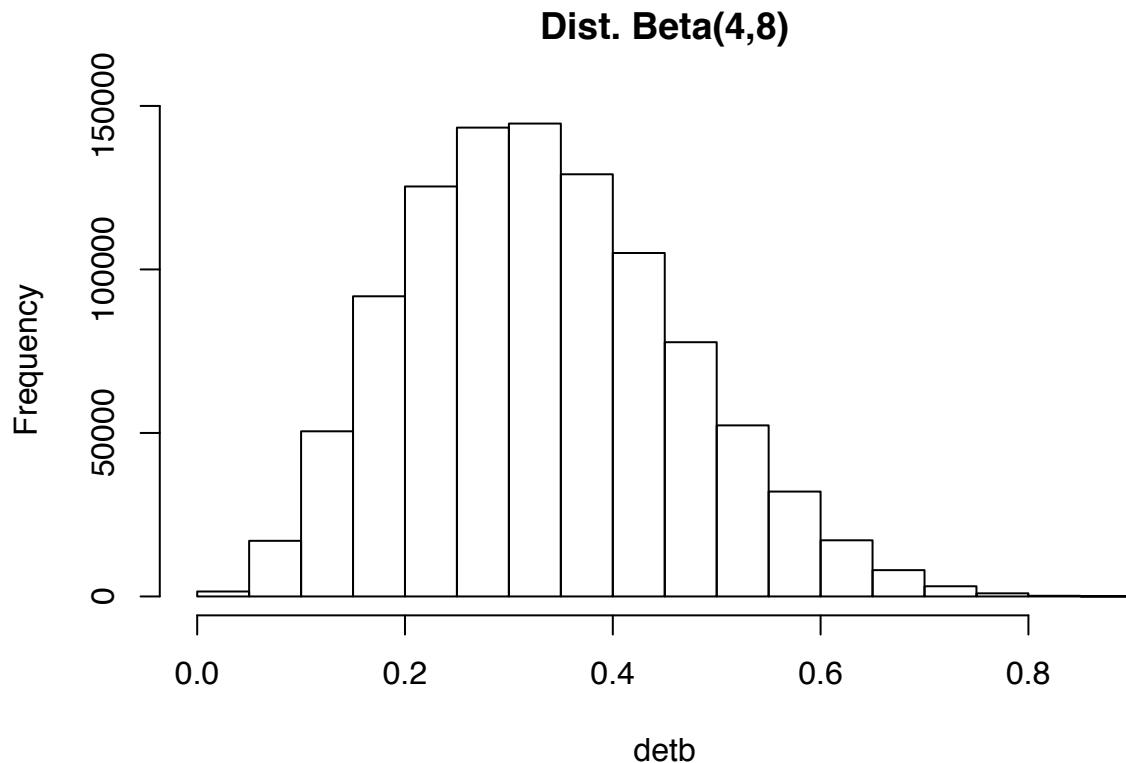
```
plot(ecdf(ppost), main = "Dist. f(p|X=3)")
```

### Dist. f(p|X=3)

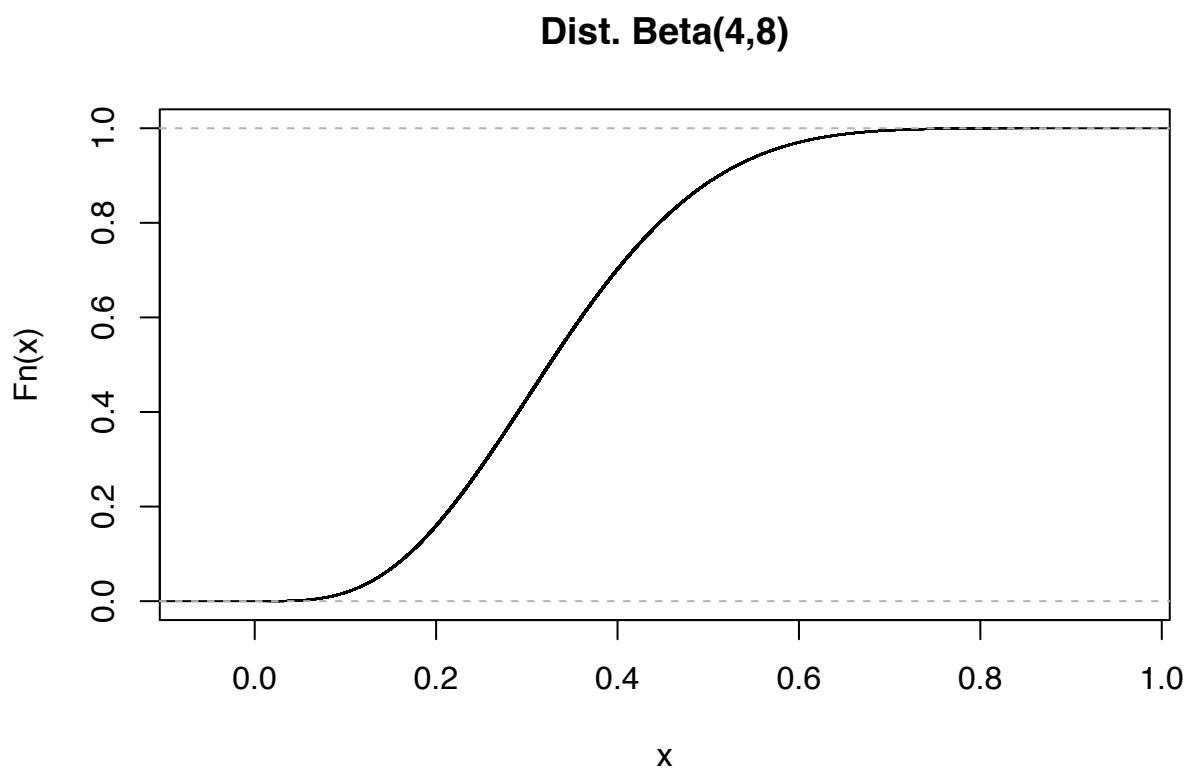


Per i valori a livello teorico ci si riferisce alla seguente

```
detb <- rbeta(10^6, 4,8)
hist(detb, main = "Dist. Beta(4,8)")
```



```
plot(ecdf(detb), main = "Dist. Beta(4,8)")
```



# Esempio di assegnazione della probabilità a priori

Fulvia Pennoni

22 Novembre 2017

## Esempio di famiglia coniugata

Supponendo di voler ottenere una stima per la proporzione del numero di ore di sonno in un giorno lavorativo degli studenti del nostro ateneo (ben 32.458 nell'accademico 2015/16) si ipotizza che la distribuzione della proporzione possa essere una distribuzione Beta( $\alpha, \beta$ ).

Si ipotizza che tale proporzione  $0 < p < 1$  possa essere ragionevolmente compresa tra 0 e 0.5 ( $8/24 = 0.3333$ ) ed inoltre che (da rilevazioni precedenti) la metà degli studenti dorma fino ad 8 ore, e solo un 10% di essi dorma più di 12 ore.

In questo caso è possibile determinare i parametri della distribuzione a priori per la proporzione ad esempio impiegando nel modo seguente la funzione della libreria `LearnBayes`.

Se si ritiene ad esempio che 0.3 sia un valore plausibile per la mediana della distribuzione (distribuzione a priori per il parametro di interesse) delle ore di sonno e che il 90-esimo percentile sia pari a circa 0.5 si ricostruiscono i parametri  $\alpha$  e  $\beta$  tali che la distribuzione risultante rispetti questi vincoli con la funzione di R della libreria `learnBayes::beta.select`.

Le quantità di interesse devono essere definite come una lista nel modo seguente per fissare la mediana `quantile1` ed il novantesino centile `quantile2` della distribuzione

```
quantile1 <- list(p = 0.5, x=0.3); quantile1  
  
## $p  
## [1] 0.5  
##  
## $x  
## [1] 0.3  
quantile2 <- list(p = 0.9, x=0.5)
```

Poi sfruttiamo la funzione `beta.select` che restituisce i valori dei due parametri che rispettano i vincoli imposti

```
require(LearnBayes)  
  
## Loading required package: LearnBayes  
beta.select(quantile1, quantile2)  
  
## [1] 3.26 7.19
```

da cui risulta che la  $f(p) \sim Beta(3.26, 7.19)$  e la proporzione media attesa è pari a

```
3.26/(3.26+7.19)
```

```
## [1] 0.3119617
```

Supponendo di utilizzare una recente intervista effettuata su un campione di 27 studenti di cui 11 dichiarano di aver dormito fino a 8 ore la notte precedente nel seguito si utilizza  $s$  (successi) per indicare il numero di coloro che dichiarano di aver dormito fino a 8 ore e  $f = (n - s)$  (insuccessi).

Dato che si tratta di una famiglia coniugata la distribuzione a posteriori per il parametro è ancora una distribuzione Beta, i cui parametri sono dati da quelli della prior e dal numero di successi campionari.

Nel seguito (cf. dispense) si disegnano le funzioni di densità: **prior**, **verosimiglianza** e **posterior**

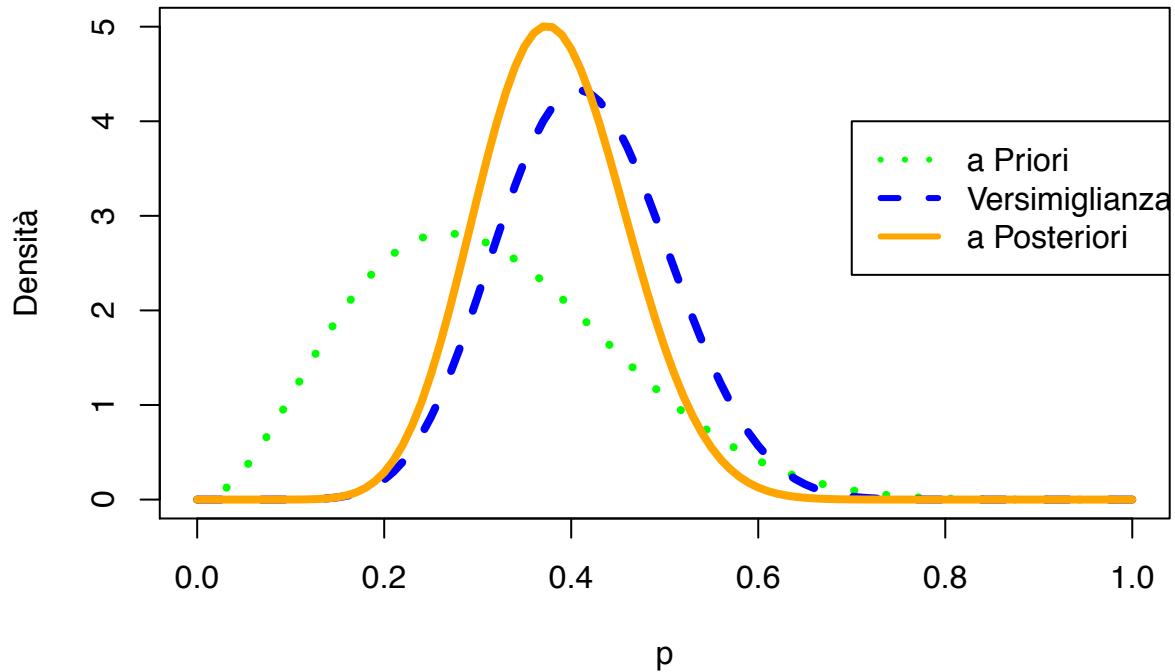
```
a <- 3.26
b <- 7.19
s <- 11
f <- 16

curve(dbeta(x,a,b),
       xlab="p",
       ylab="Densità",
       ylim=c(0,5),
       lty=3,
       lwd=4,
       col="green")

curve(dbeta(x,s+1,f+1),
      lty=2,
      lwd=4,
      col= "blue",
      add=TRUE)

curve(dbeta(x,a+s,b+f),
      from=0, to=1,
      lty=1,lwd=4,
      col="orange",
      add=TRUE)

legend(.7,4,
       c("a Priori","Versimiglianza","a Posteriori"),
       lty=c(3,2,1),
       lwd=c(3,3,3),
       col=c("green", "blue","orange" ))
```



Si nota che la distribuzione di riferimento per l'inferenza (a posteriori) è un compromesso tra il ragionamento iniziale che ha portato ad ipotizzare certi valori per il parametro (prior) e le rilevazioni effettuate (verosimiglianza) e presenta una variabilità di ridotta rispetto alla distribuzione iniziale del parametro.

# Esempi: matrici di transizione e passeggiata aleatoria

Fulvia Pennoni

4 Dicembre 2017

## Matrici di transizione

Nel seguito si richiamano le funzione di R per il prodotto matriciale. Si considera una matrice stocastica A di dimensione  $m = 3$  e  $n = 3$

```
A <- matrix(c(1/4,1/2,1/4,
              0,1/2,1/2,
              1/8,3/4,1/8),
             nrow = 3, ncol = 3, byrow =TRUE); A

##      [,1] [,2] [,3]
## [1,] 0.250 0.50 0.250
## [2,] 0.000 0.50 0.500
## [3,] 0.125 0.75 0.125
```

Essendo tutti valori per riga positivi, si verificano le somme per riga.

```
apply(A,MARGIN = 1, sum)

## [1] 1 1 1
```

da cui risulta che la somma di ogni riga è pari a 1.

Il prodotto matriciale in R si ottiene con `%*%`.

Si nota che moltiplicando la matrice per se stessa si ottiene ancora una matrice stocastica

```
A2 <- A%*%A; A2

##      [,1]      [,2]      [,3]
## [1,] 0.093750 0.562500 0.343750
## [2,] 0.062500 0.625000 0.312500
## [3,] 0.046875 0.531250 0.421875

apply(A2,MARGIN = 1, sum)

## [1] 1 1 1
```

La matrice  $A^n$  (di ordine  $n$ ) per  $n$  elevato definisce i valori della distribuzione di **equilibrio** del processo. Ad esempio per  $n = 10$  si ottiene come

```

A3 <- A2%*%A; A3

##          [,1]      [,2]      [,3]
## [1,] 0.06640625 0.5859375 0.3476562
## [2,] 0.05468750 0.5781250 0.3671875
## [3,] 0.06445312 0.6054688 0.3300781

A4 <- A3%*%A; A4

##          [,1]      [,2]      [,3]
## [1,] 0.06005859 0.5869141 0.3530273
## [2,] 0.05957031 0.5917969 0.3486328
## [3,] 0.05737305 0.5825195 0.3601074

A5 <- A4%*%A; A5

##          [,1]      [,2]      [,3]
## [1,] 0.05914307 0.5882568 0.3526001
## [2,] 0.05847168 0.5871582 0.3543701
## [3,] 0.05935669 0.5900269 0.3506165

A6 <- A5%*%A; A6

##          [,1]      [,2]      [,3]
## [1,] 0.05886078 0.5881500 0.3529892
## [2,] 0.05891418 0.5885925 0.3524933
## [3,] 0.05866623 0.5876541 0.3536797

A7 <- A6%*%A; A7

##          [,1]      [,2]      [,3]
## [1,] 0.05883884 0.5882473 0.3529139
## [2,] 0.05879021 0.5881233 0.3530865
## [3,] 0.05887651 0.5884199 0.3527036

A8 <- A7%*%A; A8

##          [,1]      [,2]      [,3]
## [1,] 0.05882394 0.5882285 0.3529476
## [2,] 0.05883336 0.5882716 0.3528950
## [3,] 0.05880708 0.5881759 0.3530170

A9 <- A8%*%A; A9

##          [,1]      [,2]      [,3]
## [1,] 0.05882443 0.5882369 0.3529387
## [2,] 0.05882022 0.5882238 0.3529560
## [3,] 0.05882890 0.5882543 0.3529168

A10 <- A9%*%A; A10

##          [,1]      [,2]      [,3]
## [1,] 0.05882344 0.5882347 0.3529419
## [2,] 0.05882456 0.5882390 0.3529364
## [3,] 0.05882183 0.5882292 0.3529490

```

Si nota che al crescere di  $n$  la matrice stocastica  $A$  tende ad avere stessi valori per ogni riga.

## Distribuzione di equilibrio e scomposizione spettrale

La distribuzione stazionaria si ricava anche considerando gli **autovalori**. Prima si traspone la matrice e poi si determinano gli autovettori e agli autovalori.

La funzione `base::eigen` permette di estrarre i valori

```
eiA <- eigen(t(A))
```

Per la distribuzione di equilibrio sono di interesse solo gli autovettori corrispondenti all'autovalore massimo (pari a 1) che devono essere normalizzati in modo da ottenere 1 per loro somma nel modo seguente

```
vA <- eiA$vector[,1]; vA  
## [1] 0.08543577 0.85435766 0.51261459  
sum(vA)  
## [1] 1.452408  
vA/sum(vA)  
## [1] 0.05882353 0.58823529 0.35294118  
sum(vA/sum(vA))  
## [1] 1
```

si ottengono le probabilità che definiscono la distribuzione stazionaria di  $A$ .

# Esempio di passeggiata aleatoria a tempo discreto

Fulvia Pennoni

4 Dicembre 2017

## Esempio di percorso nel caso di un processo random walk

Nel seguente codice si simulano delle realizzazioni per una passeggiata casuale (illustrata nella parte II delle dispense). Si considera un processo stocastico con un numero di stati da 0 a 15.

Si definisce la probabilità costante  $p$  per incrementare o decrementare la posizione rispetto ad uno stato e stabilendo uno stato di inizio si simula un certo numero possibili stati che il processo può assumere. Si noti che come mostrato nella matrice di transizione della parte delle dispense lo stato 0 e lo stato 15 sono stati **assorbenti**.

Nel seguente ciclo iterativo si definiscono i costrutti condizionali **if** per gli stati assorbenti e **else** per quelli transienti.

```
ini <- 0
fine <- 15
p <- 0.8
nsim <- 150
x <- rep(ini,nsim)
x[1] <- 5
set.seed(172)
for(i in 2:nsim){
  if(x[i-1]==ini || x[i-1]==fine){
    x[i] <- x[i-1]
  }
  else{
    x[i] <- x[i-1] + sample(c(1,-1), 1, prob=c(p,1-p))
  }
}
x

## [1] 5 6 7 6 7 8 9 10 11 12 13 14 15 15 15 15 15 15 15 15 15 15
## [24] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [47] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [70] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [93] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [116] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
## [139] 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
```

Per campionare un valore (1) oppure (-1) si utilizza la funzione **sample** in base alla probabilità definita di spostarsi in una direzione o in un'altra direzione.

Dalle realizzazioni notiamo che partendo dalla stato 5 si precipita abbastanza velocemente allo stato 0 essendo la probabilità di passaggio pari a 0.2.

# Esempio di stima del modello di transizione

*Fulvia Pennoni*

*5 Dicembre 2017*

## Esempio modello di transizione

I dati presenti nel dataset `datadr.Rdata` (dati simulati) riguardano una survey più recente rispetto a quella dell'esempio precedente (`data_drug` cf. codice di R). Sono dati longitudinali in cui le unità statistiche sono disposte per riga mentre per colonna vi sono i periodi temporali e ogni riga rappresenta la categoria di risposta della variabile consumo di cannabis per ogni anno della survey.

```
load("datadr.Rdata")
dim(datadr)

## [1] 1000     4

head(datadr)

##      [,1] [,2] [,3] [,4]
## [1,]    1    0    2    0
## [2,]    0    2    2    1
## [3,]    0    2    2    0
## [4,]    1    2    2    2
## [5,]    0    0    2    0
## [6,]    1    0    2    2
```

Si nota che ad esempio il primo individuo era un consumatore occasionale (categoria 1) alla prima survey, poi non più utilizzatore (categoria 0) alla seconda survey, poi era consumatore abituale (categoria 2) alla terza survey e poi non era più consumatore (categoria 0) alla quarta survey.

E' possibile ottenere la tabella frequenze di riposta osservate per ogni anno.

```
table(datadr[,1])

##
##   0   1   2
## 511 300 189

table(datadr[,2])

##
##   0   1   2
## 247 69 684

table(datadr[,3])

##
##   0   1   2
## 275 65 660
```

```
table(data[,4])
```

```
##  
##    0    1    2  
## 233 295 472
```

Dalla tabella si nota che le frequenze assolute dei consumatori saltuari in particolare sono molto diverse nei vari anni.

Il modello si stima con la funzione seguente presente nella libreria `LMest::est_mc_basic` che richiede le configurazioni di risposta e le rispettive frequenze e l'ipotesi sulle probabilità di transizione. L'opzione `mod=1` specifica l'ipotesi di catena di Markov omogenea nel tempo

```
require(LMest)
```

```
## Loading required package: LMest  
## Loading required package: MASS  
## Loading required package: MultiLCIRT  
## Loading required package: limSolve  
mod1 = est_mc_basic(Sconf,yvconf,mod=1)  
mod1  
  
## Call:  
## est_mc_basic(S = Sconf, yv = yvconf, mod = 1)  
##  
## Convergence info:  
##      LogLik np      AIC      BIC  
## [1,] -3694.146 8 7404.292 7443.554
```

```
summary(mod1)
```

```
## Call:  
## est_mc_basic(S = Sconf, yv = yvconf, mod = 1)  
##  
## Coefficients:  
##  
## Initial probabilities:  
##   est_piv  
## 0 0.511  
## 1 0.300  
## 2 0.189  
##  
## Transition probabilities:  
##           category  
## category      0      1      2  
## 0 0.1123 0.1152 0.7725  
## 1 0.2834 0.1429 0.5737  
## 2 0.3366 0.1618 0.5016
```

All'inzio della survey si nota che i non consumatori di cannabis sono circa la metà. Durante il secondo, terzo e quarto anno dell'indagine i non consumatori hanno una probabilità piuttosto bassa (0.11).

All'inzio della survey circa il 30% sono consumatori occasionali. Nei sondaggi successivi al primo i consumatori occasionali mostrano bassa persistenza in questo stato.

All'inzio della survey circa il 19% sono consumatori abituali. Coloro che dichiarano di utilizzare la cannabis in modo continuativo hanno alta probabilità (0.5) di farne un utilizzo continuativo anche negli anni consecutivi al primo sondaggio.

```
mod1$Fy
```

```
##      category
## time      0      1      2
##   1 0.5110000 0.3000000 0.1890000
##   2 0.2060219 0.1322989 0.6616792
##   3 0.2833478 0.1496759 0.5669763
##   4 0.2650793 0.1457457 0.5891750
```

La distribuzione congiunta delle variabili risposta permette di evidenziare che la probabilità riferita ai non consumatori decresce nel tempo mentre quella dei consumatori occasionali cresce così come quella dei consumatori abituali che cresce notevolmente dalla prima alla seconda survey a poi decresce leggermente durante la terza o quarta survey rimandando tuttavia la più elevata tra quelle stimate.

# Esempio di stima del modello di latente di Markov

*Fulvia Pennoni*

*11 Dicembre 2017*

## Esempio modello di stima del modello latente di Markov

Il seguente esempio è tratto Bartolucci, Pandolfi, Pennoni (2017) (Sezione 3.1) (cf. dispense parte teorica). I dati si riferiscono alle risposte ad una domanda del questionario rivolto ai maggiorenni in un'indagine russa (Russian Longitudinal Monitoring Survey) circa il livello di soddisfazione riscontrato rispetto al lavoro principale. Si tratta delle risposte fornite da 1718 persone intervistate per 7 anni consecutivi dal 2008 al 2014.

Nel questionario il livello di soddisfazione complessivo doveva essere espresso su una scala da 1 a 5 con: 1 completamente soddisfatto, 2 abbastanza soddisfatto, 3 nè soddisfatto nè insoddisfatto, 4 non molto soddisfatto, 5 particolarmente insoddisfatto.

```
library("LMest")
## Loading required package: MASS
## Loading required package: MultiLCIRT
## Loading required package: limSolve
data("RLMSdat", package = "LMest")
head(RLMSdat)

##    IKSJQ IKSJR IKSJS IKSJT IKSJU IKSJV IKSJW
## 1     2     2     2     2     1     1     2
## 2     2     2     3     2     2     2     2
## 3     2     4     4     2     3     4     2
## 4     2     3     2     2     2     2     2
## 5     2     2     3     2     2     2     2
## 6     3     4     3     2     2     4     3
```

I dati mostrano che il pattern di risposta del primo individuo oscilla tra abbastanza soddisfatto e molto soddisfatto e che solo 2 individui (id 3 e id 6) nei primi 6 si dichiarano non molto soddisfatti negli anni 1999, 2000 e 2013.

Utilizzando questi pattern di risposta è possibile ottenere i pattern di frequenza con la funzione `agg_data`, che restituisce la matrice `S` dei pattern ed il vettore delle rispettive frequenze.

```
out <- agg_data(RLMSdat)
yv <- out$freq
S <- 5 - out$data_dis
head(S)

##    IKSJQ IKSJR IKSJS IKSJT IKSJU IKSJV IKSJW
## [1,]    3     3     3     3     4     4     3
## [2,]    3     3     2     3     3     3     3
## [3,]    3     1     1     3     2     1     3
## [4,]    3     2     3     3     3     3     3
```

```

## [5,]    2    1    2    3    3    1    2
## [6,]    4    3    2    1    1    3    1
head(yv)

## [1] 1 13 1 14 1 1

```

Si noti che cambiando la scala con cui sono misurate le risposte adesso nella matrice  $S$  risultano i valori da 0 a 4 che prima erano da 1 a 5 si ha che (1) 4; (2) 3; (3) 2; (4) 1; (5) 0.

$S$  rappresenta la matrice delle configurazioni di risposta rispetto ai 7 anni della survey (dato che viene tolto 5 ad ogni elemento: la codifica 0 indica assolutamente insoddisfatto mentre 4 indica completamente insoddisfatto). Il vettore  $yv$  indica le corrispondenti numerosità osservate la cui somma fornisce il numero totale delle unità. Si nota che ci sono 14 individui che prima si dichiarano nè soddisfatti nè insoddisfatti e poi abbastanza soddisfatti.

```

max(yv)

## [1] 168
which(yv==168)

## [1] 23
S[23,]

## IKSJQ IKSJR IKSJS IKSJT IKSJU IKSJV IKSJW
##     3     3     3     3     3     3     3

```

L'individuazione dei pattern di risposta più frequenti e delle corrispondenti frequenze osservate per i dati disponibili si effettua sul vettore  $yv$ . Si nota che vi sono 168 individui che si ritengono sempre abbastanza soddisfatti del loro lavoro.

La funzione che permette di stimare il modello `est_lm_basic` richiede come parametri di input: la matrice delle configurazioni di risposta  $S$  ed il vettore delle frequenze  $yv$ , il numero di stati latenti  $k \leq 5$  e `mod=1` per assumere l'omogeneità della matrice di transizione, `start=0` per inizializzare il modo deterministico l'algoritmo di stima Expectation-Maximization.

```
mod1 <- est_lm_basic(S, yv, k = 3, mod = 1, start = 0)
```

mod	k	start	step	lk	lk-lko	discrepancy
1	3	0	0	-15606.8		
1	3	0	10	-13577.1	4.87573	0.00765358
1	3	0	20	-13559.7	0.533973	0.00266113
1	3	0	30	-13557.6	0.073773	0.000832573
1	3	0	40	-13557.3	0.01444436	0.000366457
1	3	0	50	-13557.2	0.00374981	0.000198311
1	3	0	60	-13557.2	0.00116625	0.000106657
1	3	0	70	-13557.2	0.000413452	5.7188e-05
1	3	0	80	-13557.2	0.000164074	3.06113e-05
1	3	0	83	-13557.2	0.000126816	2.53723e-05

```

## -----|-----|-----|-----|-----|-----|-----|
mod1

## Call:
## est_lm_basic(S = S, yv = yv, k = 3, start = 0, mod = 1)
##
## Convergence info:
##      LogLik np      AIC      BIC
## [1,] -13557.21 20 27154.41 27263.39

```

Fissando  $k = 3$  si intende individuare 3 sottopolazioni che sono clusterizzate in base al livello di soddisfazione.

L'oggetto in output restituisce il valore della verosimiglianza a convergenza, il numero dei parametri liberi del modello ed i corrispondenti valori di AIC e BIC.

La funzione **summary** permette di stampare le stime ottenute dal modello che riguardano le probabilità condizionate  $\hat{\phi}_{y|u}$  e le probabilità del processo latente iniziali  $\hat{\pi}_u$  e di transizione  $\hat{\pi}_{u|\bar{u}}$

```
summary(mod1)
```

```

## Call:
## est_lm_basic(S = S, yv = yv, k = 3, start = 0, mod = 1)
##
## Coefficients:
## 
## Initial probabilities:
##      est_piv
## [1,] 0.3638
## [2,] 0.4623
## [3,] 0.1739
## 
## Transition probabilities:
##      state
## state   1     2     3
##   1 0.8863 0.0991 0.0145
##   2 0.0472 0.9335 0.0193
##   3 0.0469 0.0774 0.8757
## 
## Conditional response probabilities:
## , , item = 1
## 
##      state
## category 1     2     3
##   0 0.0731 0.0000 0.0031
##   1 0.2442 0.0224 0.0233
##   2 0.4421 0.0965 0.0448
##   3 0.2197 0.8166 0.3480
##   4 0.0209 0.0645 0.5808

```

Per interpretare i 3 stati latenti si ricorre alle probabilità stimate riferite al modello di misura  $\hat{\phi}_{y|u}$ : probabilità di ogni categoria di risposta condizionata allo stato latente. Rispetto alle quali si evince che il primo gruppo è riferito a coloro che mostrano insoddisfazione o che non sono né soddisfatti né insoddisfatti (2 o 3); il

# Esempio di famiglia coniugata e inferenza predittiva

Fulvia Pennoni

9 Gennaio 2018

## Stima del tasso di mortalità rispetto a trapianti di cuore

Il seguente esempio tratto da Albert (2009) mostra un applicazione delle distribuzioni della famiglia coniugata ed l'utilizzo della distribuzione predittiva. Si dispone del numero  $n$  di trapianti di cuore eseguiti presso un certo ospedale e del numero  $y$  di decessi entro 30 giorni. Per le  $Y_1, \dots, Y_n$  si assume una distribuzione di Poisson dove il parametro è definito da  $\lambda e$  ovvero dal tasso di mortalità  $\lambda$  in base al numero di unità esposte  $e$ . La stima di massima verosimiglianza per il parametro potrebbe essere imprecisa quando le realizzazioni sono prossime a zero.

Si utilizza un'informazione a priori per il tasso di mortalità nella famiglia coniugata  $p(\lambda) \sim \text{Gamma}(\alpha, \beta)$  dove i valori di  $\alpha$  e  $\beta$  possono essere determinati in base ai dati osservati sui trapianti effettuati in altri ospedali simili tra loro per tasso di mortalità. Ad esempio se si dispone delle rilevazioni riferite a 10 ospedali si considera

$$\alpha = \sum_{j=1}^{10} z_j \quad \beta = \sum_{j=1}^{10} o_j$$

dove  $z_j$  e  $o_j$ ,  $j = 1, \dots, J$  rappresentano rispettivamente il numero complessivo di decessi nell'ospedale  $j$ -esimo ed il numero dei pazienti esposti al trapianto. Supponendo ad esempio

$$\sum_{j=1}^{10} z_j = 16$$

e

$$\sum_{j=1}^{10} o_j = 15174.$$

risulta  $p(\lambda) \sim \text{Gamma}(16, 15174)$ . In base alla distribuzione di Poisson la distribuzione a posteriori è  $\text{Gamma}(\alpha + \sum_i^n y_i, \beta + e)$  (cf. parte teorica delle dispense).

La distibuzione prevista per il numero di decessi  $f(y)$  è una distribuzione di Poisson. Con la distribuzione predittiva  $f(y)$  si può valutare la validità del modello proposto. Se i valori osservati sono compatibili con quelli ottenuti dalla distribuzione marginale a posteriori il modello assunto è ragionevole. Altrimenti se ad esempio i dati osservati sono nelle code della distribuzione predittiva il modello assunto per la prior potrebbe non essere adeguato, oppure la distribuzione delle verosimiglianza non è stata correttamente specificata.

Supponendo che per l'ospedale di interesse si riscontra 1 decesso entro 30 giorni e rispetto ad un nuemro di operazioni eseguite pari a 66 il tasso di mortalità risulta pari a  $1/66$ . Supponendo un valore medio per  $\lambda = \frac{\alpha}{\beta}$  la distribuzione predittiva  $f(y)$  si determina

$$f(y_{new}|y) = \frac{f(y|\lambda)p(\lambda)}{f(\lambda|y)}$$

ovvero dal prodotto della Poisson (prior per  $\lambda$ ) e della  $\text{Gamma}(\alpha, \beta)$  (verosimiglianza), rispetto alla  $\text{Gamma}(\alpha + \sum_i y_i, \beta + e)$  (distribuzione a posteriori).

Nel seguente si ottiene la distribuzione predittiva definendo i parametri della distribuzione a priori

```
alpha <- 16
beta <- 15174
```

il numero osservato di decessi e l'esposizione per l'ospedale considerato e un insieme di valori plausibile per l'evento decesso

```
yobs <- 1
ex <- 66
y <- 0:10
```

il valore medio per  $\lambda$

```
lam <- alpha/beta; lam
## [1] 0.001054435
```

```
fy <- dpois(y, lam*ex)*dgamma(lam,
                                 shape=alpha,
                                 rate = beta)/dgamma(lam,
                                 shape = alpha+y,
                                 rate = beta +ex)
cbind(y, round(fy,3))

##      y
## [1,] 0 0.933
## [2,] 1 0.065
## [3,] 2 0.002
## [4,] 3 0.000
## [5,] 4 0.000
## [6,] 5 0.000
## [7,] 6 0.000
## [8,] 7 0.000
## [9,] 8 0.000
## [10,] 9 0.000
## [11,] 10 0.000
```

da cui si evince che la densità predittiva si concentra nei valori 0 e 1 e permette di validare le scelte sulle distribuzioni implicate nel modello Bayesiano.

# Regione con la massima densita' a posteriori

Fulvia Pennoni

9 Gennaio 2018

## Esempio: distribuzione Beta-Binomiale

Nel seguente esempio si utilizzano metodi numerici per determinare la densità con la massima probabilità per il parametro caso in cui la distribuzione a posteriori deriva dalla famiglia coniugata Beta-Binomiale.

Nel procedimento che segue si fissa un valore  $k$

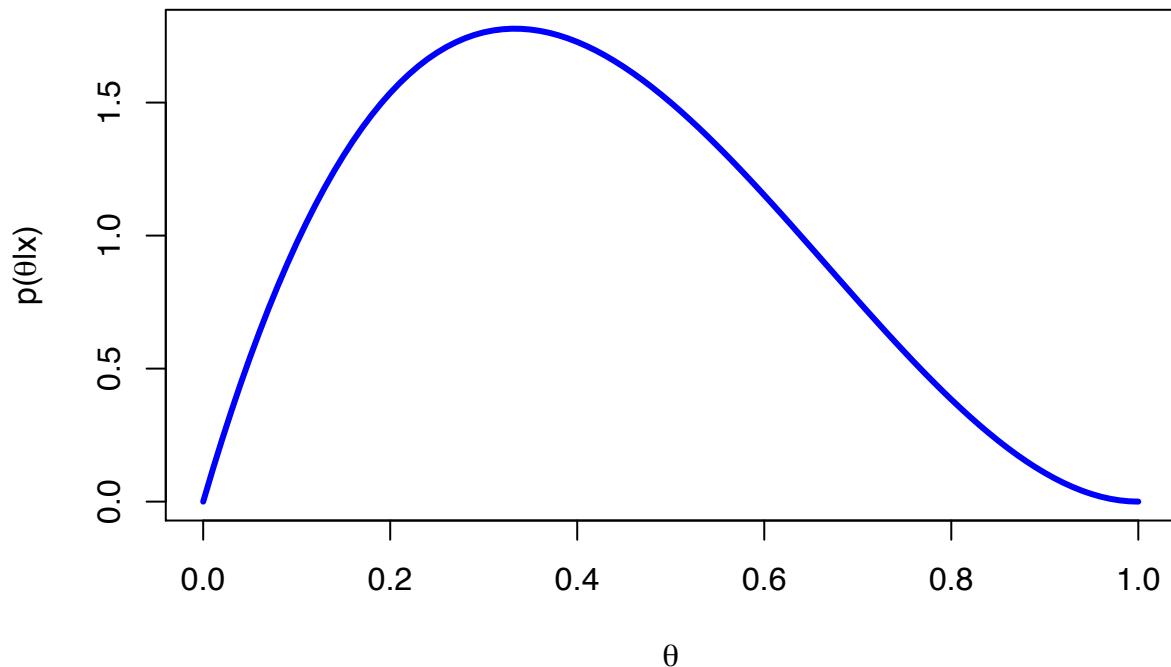
- $k$  è un valore inferiore al massimo della  $p(\theta|x)$  e si determinano le radici  $a, b$  dell'equazione  $p(\theta|x) = k(\alpha)$
- si cerca la regione  $C = P(\theta \in a, b)$  considerando  $\tilde{k}$  come il valore che risolve l'equazione  $\mathbf{C} = 1 - \alpha$  tale che si ottiene l'intervallo  $(\theta_a(\tilde{k}), \theta_b(\tilde{k}))$  con massima probabilità per  $\theta$ .

Esempio per una funzione di densità a posteriori Beta(2,3) (cf. esempio sulle ore di sonno degli studenti) la densità è la seguente

```
densbeta <- function(x,a,b){x^(a-1)*(1-x)^(b-1)/beta(a,b)}
```

```
curve(densbeta(x,a=2,b=3),0,1,
      ylab=expression(paste(p, "(", theta, "| x"))),
      xlab=expression(theta),
      col ="blue",
      lwd = 3 )
```



Se la regione di credibilità viene determinata con il metodo del percentile non è una regione con massima densità a posteriori per i valori del parametro: per  $\alpha = 0.05$  l'intervallo di confidenza risulta pari a

```
ci <- qbeta(c(0.025,0.975),2,3); ci
```

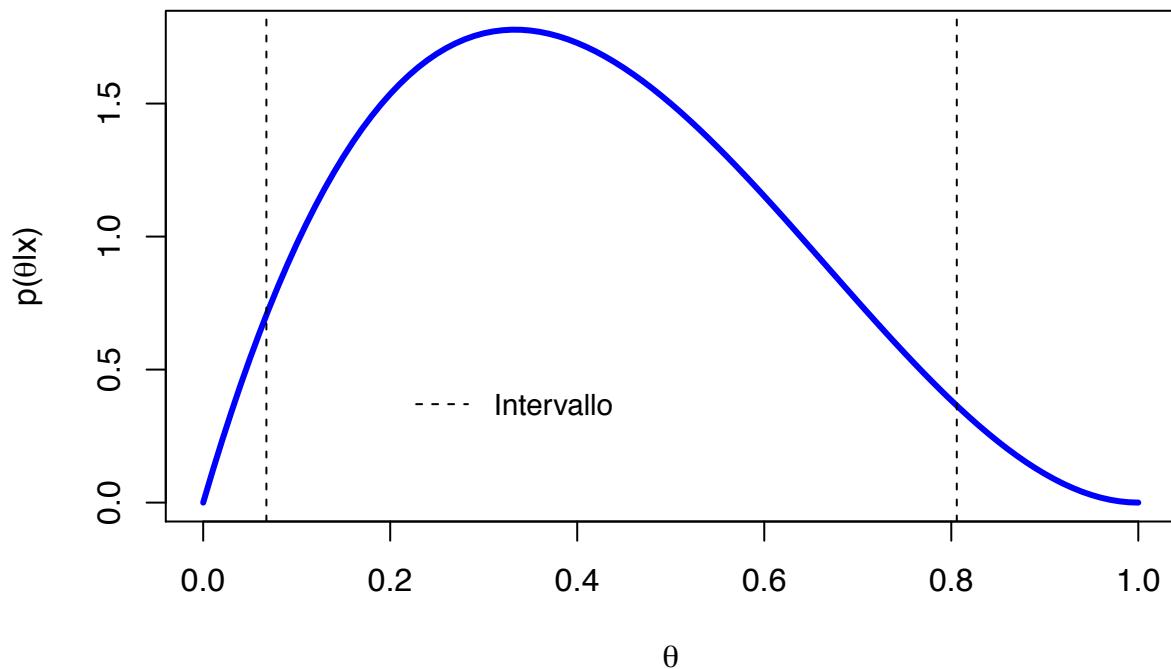
```
## [1] 0.06758599 0.80587955
```

come evidenziato nella seguente figura l'intervallo non è centrato rispetto alla moda

```
ci <- qbeta(c(0.025,0.975),2,3); ci
```

```
## [1] 0.06758599 0.80587955
```

```
curve(densbeta(x,a=2,b=3),0,1,
      ylab=expression(paste(p,"(",theta,"|x)")),
      xlab=expression(theta),
      col ="blue",
      lwd = 3)
abline(v=ci,lty=2)
legend(0.2,0.5,"Intervallo",lty=2,bty="n",cex=0.9)
```



Si noti che il massimo della funzione può essere ricavato utilizzando la funzione `optimize` che richiede come input la funzione ed il supporto del parametro

```
densbeta1<-function(x){x^(2-1)*(1-x)^(3-1)/beta(2,3)}
optimise(densbeta1, c(0,1), maximum = TRUE)
```

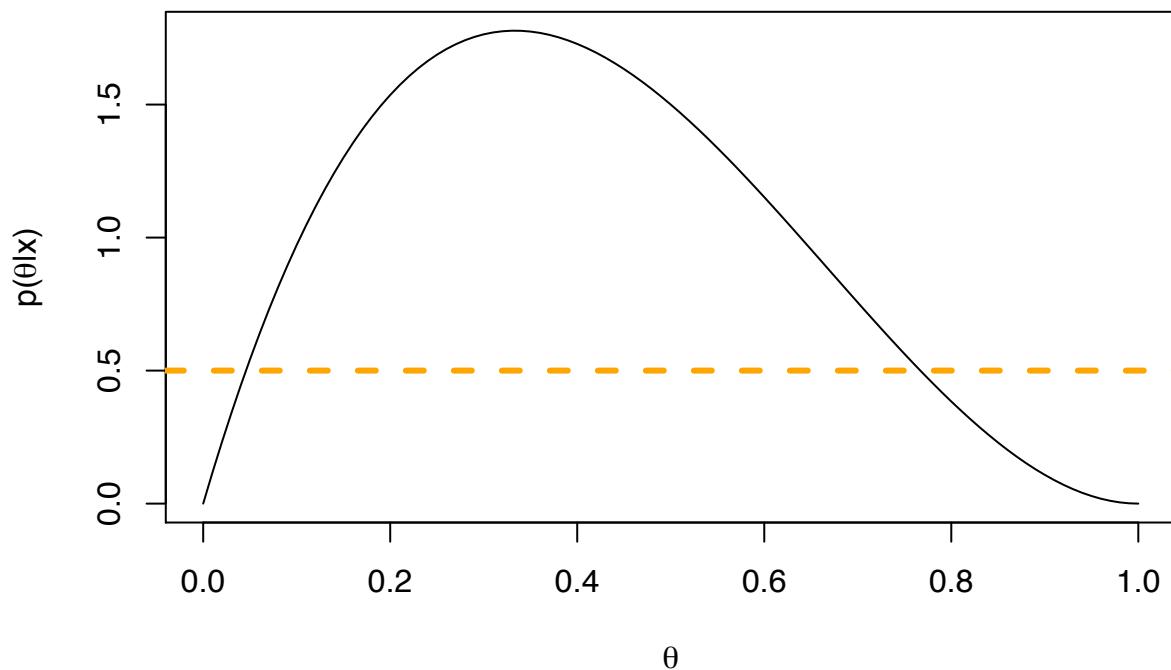
```
## $maximum
## [1] 0.3333205
```

```
##  
## $objective  
## [1] 1.777778
```

e restituisce come primo argomento i valori corrispondenti al punto di massimo `objective` sull'asse riferito a  $\theta$  `maximum`.

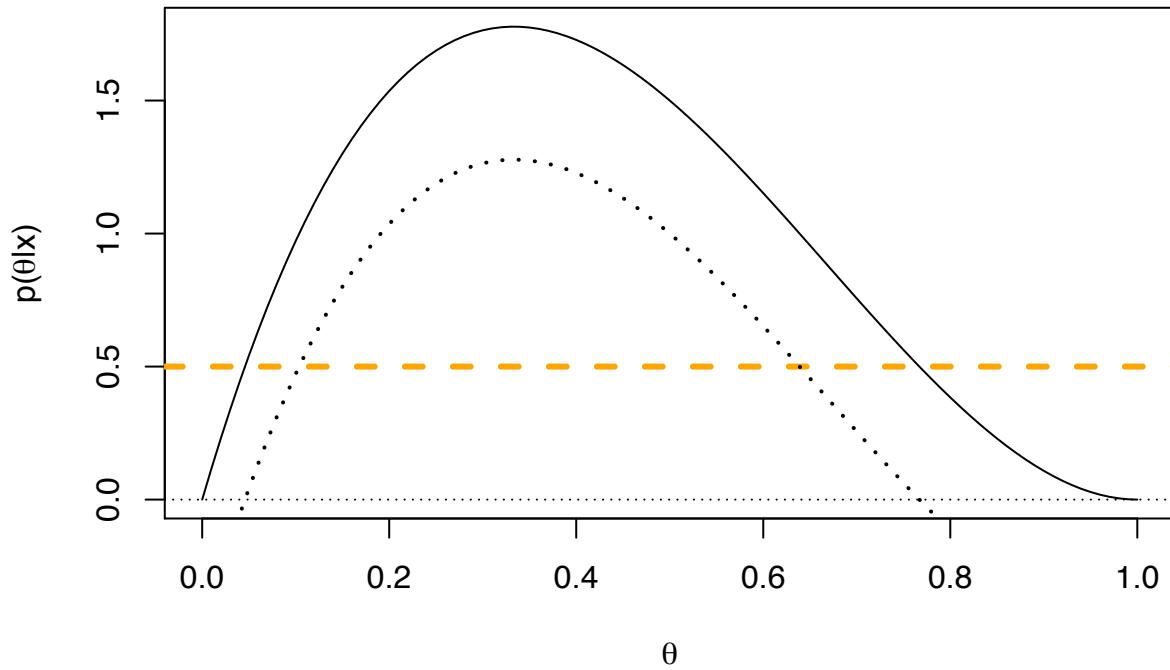
Per determinare l'**intervallo HPD** (highest posterior density) occorre cercare il valore di  $\tilde{k}$  partendo da un valore iniziale ad esempio  $k = 0.5$

```
curve(densbeta(x,a=2,b=3),0,1,  
      ylab=expression(paste(p,"(",theta,",|x)")),  
      xlab=expression(theta))  
abline(h=0.5,lty=2, col ="orange", lwd=3)
```



Si scala la funzione rispetto a  $k = 0.5$  e si disegnano le due densità

```
betat <- function(x,a,b) densbeta(x,a,b)-0.5  
  
curve(densbeta(x,a=2,b=3),0,1,  
      ylab=expression(paste(p,"(",theta,",|x)")),  
      xlab=expression(theta))  
abline(h=0.5,lty=2, col ="orange", lwd=3)  
curve(betat(x ,a=2,b=3),add=T,lty=3,lwd=2)  
abline(h=0,lty=3)
```



I valori  $\theta_a$  e  $\theta_b$  nella funzione scalata rappresentano gli estremi della regione con la massima probabilità a posteriori e si ottengono con la funzione `uniroot` che richiede la funzione ed il campo di variazione per le due radici

```
hpd1 <- uniroot(betat,c(0.001,0.1),a=2,b=3);
hpd2 <- uniroot(betat,c(0.7,0.9),a=2,b=3)

h1 <- hpd1$root; h1
## [1] 0.04576017
h2 <- hpd2$root; h2
## [1] 0.7669086
```

Si calcola l'area complessiva della regione e si valuta rispetto al valore scelto per la credibilità ad esempio  $1 - \alpha = 0.95$

```
integrate(densbeta,lower=h1,upper=h2,a=2,b=3)
## 0.9463883 with absolute error < 1.1e-14
```

Dato che risulta inferiore a 0.95 per considerare i valori fino al valore di  $1 - \alpha$  si ripete il procedimento per  $k \in [0.45, 0.5]$ : si generano più valori di  $k$  e si definisce una matrice (`risultati`) che contiene tutti i valori degli estremi della regione con massima densità a posteriori

```

k <- seq(0.45,0.5,by=0.001)
risultati <- matrix(NA,ncol=3,nrow=length(k))

for(i in 1:length(k)){
  traslata <- function(x,a,b)densbeta(x,a,b)-k[i]
  hpd1 <- uniroot(traslata,c(0.001,0.1),a=2,b=3)
  h1 <- hpd1$root
  hpd2 <- uniroot(traslata,c(0.7,0.9),a=2,b=3)
  h2 <- hpd2$root
  int <- integrate(densbeta,
                    lower=h1,
                    upper=h2,
                    a=2,b=3)
  int1<-int$value
  fine<-i
  if(int1 <= 0.95) break
  risultati[i,] <- c(h1,h2,int1)
}

fine

## [1] 32
risultati

##          [,1]      [,2]      [,3]
## [1,] 0.04075332 0.7808503 0.9553878
## [2,] 0.04085231 0.7805672 0.9552157
## [3,] 0.04095135 0.7802843 0.9550432
## [4,] 0.04105042 0.7800016 0.9548705
## [5,] 0.04114955 0.7797191 0.9546974
## [6,] 0.04124871 0.7794368 0.9545240
## [7,] 0.04134792 0.7791547 0.9543503
## [8,] 0.04144718 0.7788727 0.9541763
## [9,] 0.04154648 0.7785909 0.9540019
## [10,] 0.04164582 0.7783093 0.9538273
## [11,] 0.04174520 0.7780279 0.9536523
## [12,] 0.04184464 0.7777466 0.9534770
## [13,] 0.04194411 0.7774656 0.9533013
## [14,] 0.04204363 0.7771847 0.9531254
## [15,] 0.04214319 0.7769039 0.9529491
## [16,] 0.04224280 0.7766234 0.9527725
## [17,] 0.04234245 0.7763430 0.9525956
## [18,] 0.04244215 0.7760628 0.9524184
## [19,] 0.04254189 0.7757828 0.9522409
## [20,] 0.04264168 0.7755029 0.9520630
## [21,] 0.04274151 0.7752233 0.9518848
## [22,] 0.04284138 0.7749437 0.9517063
## [23,] 0.04294130 0.7746644 0.9515275
## [24,] 0.04304127 0.7743852 0.9513483
## [25,] 0.04314128 0.7741062 0.9511689
## [26,] 0.04324133 0.7738274 0.9509891
## [27,] 0.04334143 0.7735487 0.9508090
## [28,] 0.04344158 0.7732702 0.9506285
## [29,] 0.04354177 0.7729918 0.9504478

```

```

## [30,] 0.04364200 0.7727137 0.9502667
## [31,] 0.04374228 0.7724356 0.9500853
## [32,]      NA      NA      NA
## [33,]      NA      NA      NA
## [34,]      NA      NA      NA
## [35,]      NA      NA      NA
## [36,]      NA      NA      NA
## [37,]      NA      NA      NA
## [38,]      NA      NA      NA
## [39,]      NA      NA      NA
## [40,]      NA      NA      NA
## [41,]      NA      NA      NA
## [42,]      NA      NA      NA
## [43,]      NA      NA      NA
## [44,]      NA      NA      NA
## [45,]      NA      NA      NA
## [46,]      NA      NA      NA
## [47,]      NA      NA      NA
## [48,]      NA      NA      NA
## [49,]      NA      NA      NA
## [50,]      NA      NA      NA
## [51,]      NA      NA      NA

hpd <- risultati[fine-1,-3];hpd
```

```
## [1] 0.04374228 0.77243565
```

Si noti che il ciclo for si interrompe quando il contatore arriva al valore 32 di k ovvero per un valore di  $\tilde{k}$  uguale a

```

k <- seq(0.45,0.5,by=0.001)
k[32]
```

```
## [1] 0.481
```

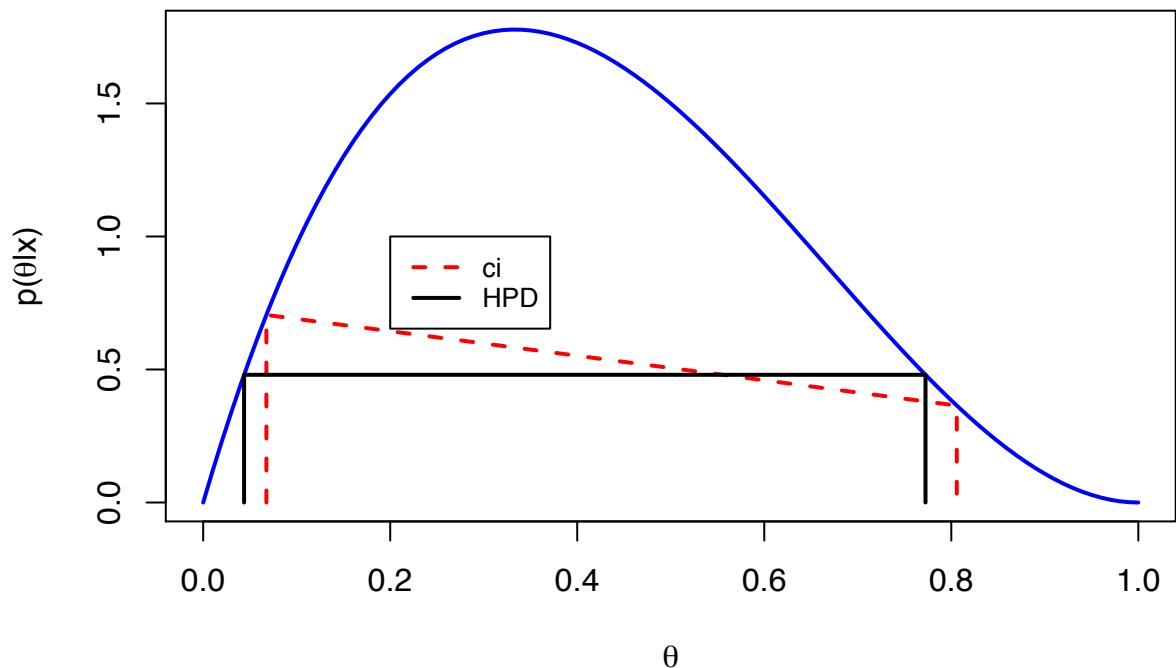
che assicura che l'area sottostante sia pari al livello di credibilità desiderato.

Tracciando i segmenti rispetto all'intervallo HPD e all'intervallo ottenuto con il metodo del percentile si nota di come quest'ultimo e' influenzato dall'asimmetria della distribuzione.

```

curve(densbeta(x,a=2,b=3),0,1,
      ylab=expression(paste(p,"(",theta,"|x)")),
      xlab=expression(theta),lwd=2, col ="blue")
lines( x=c(ci[1],ci[1],ci[2],ci[2]),
       y=c(0,dbeta(c(ci[1],ci[2]),2,3),0),
       col="red",
       lwd=2,
       lty=2)
lines( x=c(hpd[1],hpd[1],hpd[2],hpd[2]),
       y=c(0,dbeta(c(hpd[1],hpd[2]),2,3),0),
       col=1,
```

```
lwd=2)
legend(0.2,1,c("ci","HPD"),
       col=c("red",1),
       cex=0.8,
       lwd=2,
       lty=c(2,1))
```



# Scelta della prior ed inferenza robusta

*Fulvia Pennoni*

*10 Gennaio 2018*

## Esempio famiglia coniugata Gaussiana e T di Student

Nel seguente esempio si valutano due diverse prior (Distribuzione di Gauss e T di Student) per la statistica media campionaria quando si assume nota la varianza della popolazione.

Si noti che la media campionaria può essere scritta come combinazione lineare della media nelle prime  $n - 1$  osservazioni nel modo seguente

$$\bar{x} = \frac{n-1}{n}\bar{x}_{(n-1)} + \frac{1}{n}x_n.$$

Nel modello coniugato Gaussiano-Gaussiano (cf. parte teorica delle dispense) se  $l(\bar{X}; \theta) \sim N(\theta, \sigma^2/n)$  e  $p(\theta) \sim N(\mu, \tau^2)$

il valore atteso della distribuzione a posteriori si può scrivere in funzione di  $x_n$  e di  $\bar{x}_{(n-1)}$

$$E[\theta | \bar{x}_{(n-1)}, x_n] = \frac{x_n}{\sigma^2 + n\tau^2} + \frac{(n-1)\bar{x}_{(n-1)} + n\tau^2 + \mu\sigma^2}{\sigma^2 + n\tau^2}$$

Supponendo di sapere in fase di elicazione che il punteggio mediano ad un test è pari a 100 e che un valore plausibile per il 95-esimo percentile della distribuzione è pari a 120 la funzione `normal.select` della libreria `LearnBayes` determina i valori dei parametri  $\mu, \tau$  della distribuzione a priori

```
require(LearnBayes)

## Loading required package: LearnBayes
quantile1 <- list(p=.5,x=100);
quantile2 <- list(p=.95,x=120)
ris <- normal.select(quantile1, quantile2);
mu <- ris$mu; mu

## [1] 100
tau <- ris$sigma; tau

## [1] 12.15914
```

Pertanto  $p(\theta) \sim N(100, 12.16^2)$ .

Supponendo  $n = 4$ , ed assumendo  $\sigma$  (deviazione standard) nota e pari a 15 nel seguente esempio (Albert, 2009) si mostra come cambia la distribuzione a posteriori quando il punteggio medio risulta pari a  $\bar{x}_1 = 110$ ,  $\bar{x}_2 = 125$  e  $\bar{x}_3 = 140$ .

Nel seguito si applicano le formule (cf. parte teorica delle dispense) per ricavare le stime dei parametri  $\tau_1$  e  $\sigma_1$  per ogni valore realizzato del punteggio medio nel modo seguente:

```

sigma <- 15
se <- sigma/sqrt(4)
xnn <- c(110, 125, 140)
tau1 <- 1/sqrt(1/se^2 + 1/tau^2)
mu1 <- (xnn/se^2 + mu/tau^2) * tau1^2
summ1 <- cbind(xnn, mu1, tau1)
summ1

##      xnn     mu1     tau1
## [1,] 110 107.2439 6.383344
## [2,] 125 118.1098 6.383344
## [3,] 140 128.9757 6.383344

```

La distribuzione a priori esercita una notevole influenza sulla distribuzione a posteriori anche quando come nel caso dell'osservazione  $\bar{x}_2 = 125$  il punteggio medio è lontano dal valore medio stabilito a priori pari a 100.

Nel seguente esempio si evidenzia come cambia la distribuzione a posteriori quando la prior è la distribuzione  $T$  di Student con parametro di locazione  $\mu$  e di scala pari a  $\tau$  e con 2 gradi di libertà. Il parametro di scala  $\tau$  deve essere ricavato rispetto ai valori elicitati per la prior che sono  $\mu = 100$  e 95esimo percentile pari a 120. Da cui

$$P(\theta < 120) = 0.95$$

$$P\left(\frac{\theta - \mu}{\tau} < \frac{120 - 100}{\tau}\right) = 0.95.$$

Fissando i gradi di libertà pari a 2

$$P\left(T_2 < \frac{20}{\tau}\right) = 0.95$$

il quantile  $t_2(p)$  (di ordine  $p$ ) della distribuzione  $T_2$  si ottiene dalla funzione `qt`

```
qt(0.95,2)
```

```
## [1] 2.919986
```

ed il valore del parametro di scala della distribuzione si ottiene dall'equazione  $\tau = \frac{20}{t_2(p)}$  per cui risulta

```
taut <- 20/qt(0.95,2); taut
## [1] 6.849349
```

La seguente figura mette in evidenza le differenze tra la distribuzione di Gauss ( $\mu = 100$  e  $\tau = 15$ ) e la distribuzione T con 2 gradi di libertà

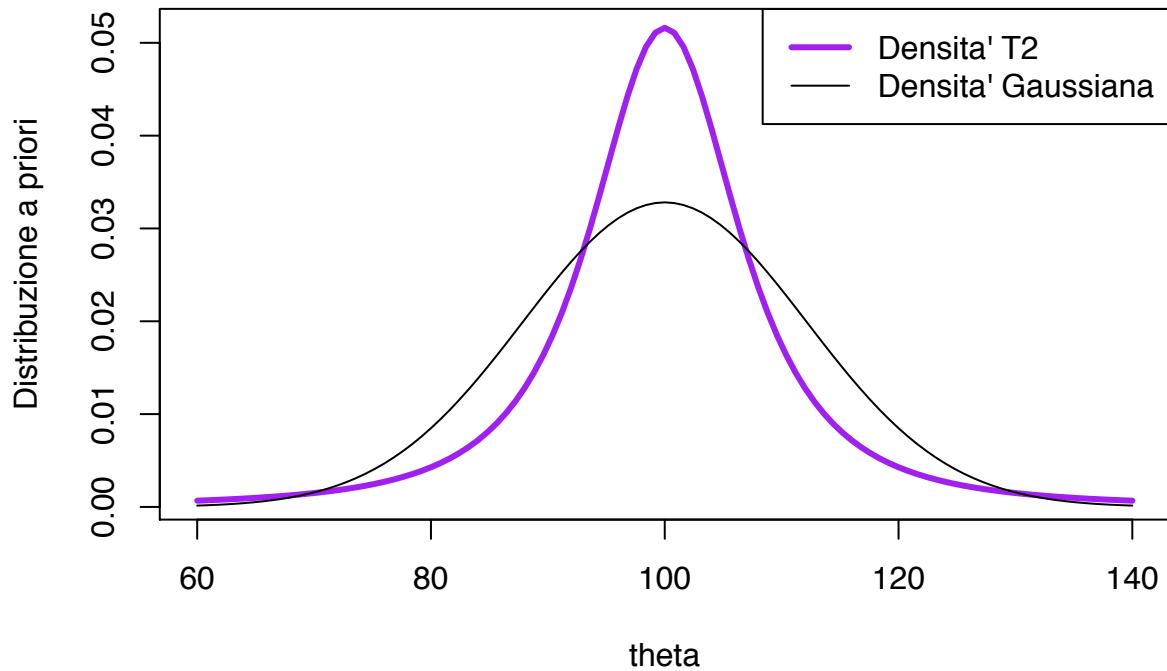
```
par(mfrow=c(1,1))
curve(1/taut*dt((x-mu)/taut,2),
```

```

from=60,
to=140,
xlab="theta",
ylab="Distribuzione a priori",
main ="Confronto delle distribuzioni per il punteggio medio",
col ='purple',
lwd=3)
curve(dnorm(x,mean=mu, sd=tau),
      add=TRUE,
      lwd=1)
legend("topright",
       legend=c("Densita' T2","Densita' Gaussiana"),
       lwd=c(3,1),
       col=c("purple", "black"))

```

## Confronto delle distribuzioni per il punteggio medio



La distribuzione a posteriore

$$p(\theta|X = x) = \phi(\bar{x}|\theta, \sigma/\sqrt{n})g_T(\theta|2, \mu, \tau)$$

si ottiene nel modo seguente approssimando la distribuzione continua con una sua discretizzazione. Si genera una sequenza di valori (griglia di valori possibili)

```

theta <- seq(60, 180, length = 500)
summary(theta)

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
##        60        90       120       120       150       180

```

Si calcolano i valori della verosimiglianza  $p(X|\theta)$  riferiti al valore medio considerando  $n = 4$

```
n <- 4
like <- dnorm(theta,mean=xnn,sd=sigma/sqrt(n))
summary(like)

##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.
## 0.000e+00 0.000e+00 1.687e-05 8.317e-03 7.223e-03 5.318e-02
```

Si determinano i valori del parametro di posizione e di scala della distribuzione a posteriori nel modo seguente

```
prior <- dt((theta - mu)/taut, 2)
post <- prior * like
post <- post/sum(post)
posizione <- sum(theta * post)
scala <- sqrt(sum(theta^2 * post) - posizione^2)
```

La seguente funzione in input prende il vettore delle osservazioni ed in output i valori attesi (locazione e scala) della distribuzione a posteriori

```
norm.t.compute <- function(xnn){
  theta <- seq(60, 180, length = 500)
  like <- dnorm(theta,mean=xnn,sd=sigma/sqrt(n))
  prior <- dt((theta - mu)/taut, 2)
  post <- prior * like
  post <- post/sum(post)
  posizione <- sum(theta * post)
  scala <- sqrt(sum(theta^2 * post) - posizione^2)
  c(xnn, posizione, scala)
}
```

Si applica la funzione al vettore  $xnn$

```
summ2<- t(sapply(c(110, 125, 140),norm.t.compute))
dimnames(summ2)[[2]]=c("xnn","mu1 t","tau1 t")
summ2

##      xnn     mu1 t     tau1 t
## [1,] 110 105.2921 5.841676
## [2,] 125 118.0841 7.885174
## [3,] 140 135.4134 7.973498

cbind(summ1,summ2)

##      xnn     mu1     tau1 xnn     mu1 t     tau1 t
## [1,] 110 107.2439 6.383344 110 105.2921 5.841676
## [2,] 125 118.1098 6.383344 125 118.0841 7.885174
## [3,] 140 128.9757 6.383344 140 135.4134 7.973498
```

Si nota rispetto ai valori ricavati nell'esempio precedente che il la mediana è rimasta invariata nel caso di

$\bar{x}_1 = 125$  ma è diminuita per  $\bar{x}_2 = 110$  e cresciuta per  $\bar{x}_3 = 140$ .

Nel grafico seguente si confrontano le distribuzioni a posteriori per il punteggio medio al test  $\bar{x}_3 = 140$  ovvero quando l'osservazione è molto lontana dal valore medio assegnato a priori

```

theta <- seq(60, 180, length=500)
mu1[3]

## [1] 128.9757
tau1

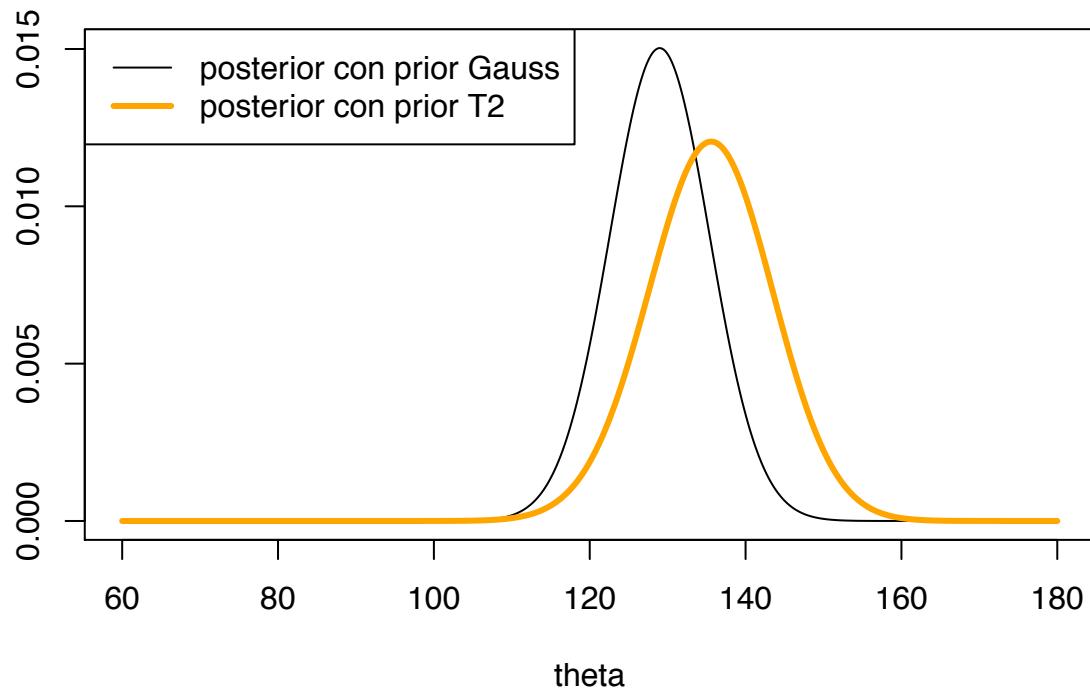
## [1] 6.383344

normpost <- dnorm(theta, mu1[3], tau1)
normpost <- normpost/sum(normpost)
plot(theta,normpost,
      type="l",
      lwd=1,
      ylab=" ")

like <- dnorm(theta,mean=140,sd=sigma/sqrt(n))
prior <- dt((theta - mu)/taut, 2)
tpost <- prior * like / sum(prior * like)

lines(theta,tpost, col="orange", lwd=3)
legend("topleft",
       legend=c("posterior con prior Gauss","posterior con prior T2"),
       lwd=c(1,3),
       col = c( "black","orange"))

```



Per valori medi osservati lontani da 100 la scelta di prior differenti conduce a distribuzioni a posteriori che possono abbastanza diverse, pertanto l'inferenza non è particolarmente robusta.

# Esempio di applicazione dell'algoritmo Metropolis-Hastings—

*Bayesian Inference –Fulvia Pennoni–*

15 Gennaio 2018

## Famiglia coniugata Gaussiana

Nel caso della famiglia coniugata Normale in cui il parametro di interesse e' la media con varianza nota è possibile ricavare in forma analitica i parametri della distribuzione a posteriori. Tuttavia come illustrato nella parte teorica delle dispense è possibile sviluppare la procedura Metropoli-Hastings nel modo che segue.

Si ipotizza un valore iniziale per  $\theta$  denotato come  $\theta^0$ . Si stabilisce la regola che alla prima simulazione

$$\theta^1 = \theta^0 + \epsilon$$

con  $\epsilon \sim N(0, d^2)$  ad esempio  $d = 1$ .

Si considerano i valori dei parametri della distribuzione a priori  $\mu$  e  $\tau$  e si considera nota la varianza nella popolazione  $\sigma = 1$ . Si procede con le simulazioni nel modo seguente

```
theta0 <- 3
mu <- 0
tau <- 2
sigma <- 1
d <- 1
```

Si sceglie il numero di iterazioni `nsim` e si salvano i valori per  $\theta$

```
nsim <- 10
theta <- rep(0,nsim)
```

L'algoritmo e' il seguente

```
theta[1] <- theta0
set.seed(173)
for(i in 2:nsim){
  theta.p <- theta[i-1] + rnorm(1,0,d)
  ratio <- dnorm(theta0, theta.p, sigma) *
    dnorm(theta.p, mu, tau) /
    (dnorm(theta0, theta[i-1], sigma) *
      dnorm(theta[i-1], mu, tau))
  print(ratio)

  evento <- rbinom(1,1,min(ratio,1))
```

```

print(evento)

theta[i] <- if(evento==1) theta.p else theta[i-1]
}

## [1] 0.2265882
## [1] 0
## [1] 0.5808754
## [1] 0
## [1] 1.23536
## [1] 1
## [1] 0.8725684
## [1] 1
## [1] 1.126791
## [1] 1
## [1] 0.7768596
## [1] 1
## [1] 0.95837
## [1] 0
## [1] 1.105759
## [1] 1
## [1] 0.3801577
## [1] 0

print(theta)

## [1] 3.000000 3.000000 3.000000 2.547716 2.889819 2.621191 1.727008
## [8] 1.727008 1.859567 1.859567

```

Con `print(ratio)` si ottengono i valori del rapporto di Metropolis-Hastings definito nell'equazione (1) delle dispense. Si nota che i 9 valori possono essere anche superiori a 1 e l'evento successo stabilisce se questi possano o meno essere accettati come realizzazioni a posteriori.

L'evento e' scelto in modo indipendente dal comportamento della catena ma la probabilita' di successo dell'evento e' scelta in base alle realizzazioni del rapporto (ratio). Il jump (ovvero la scelta di aggiungere a  $\theta$  il valore proposto) viene fatta se si realizza l'evento successo altrimenti non avviene il passo successivo.

Se si sceglie di applicare il **burn in** si eliminano le prime osservazioni. Si nota che se si eliminano la metà dei valori le realizzazioni attese sono quelle in `theta1`

```

theta1 <- theta[-(1:(nsim/2))]; theta1

## [1] 2.621191 1.727008 1.727008 1.859567 1.859567

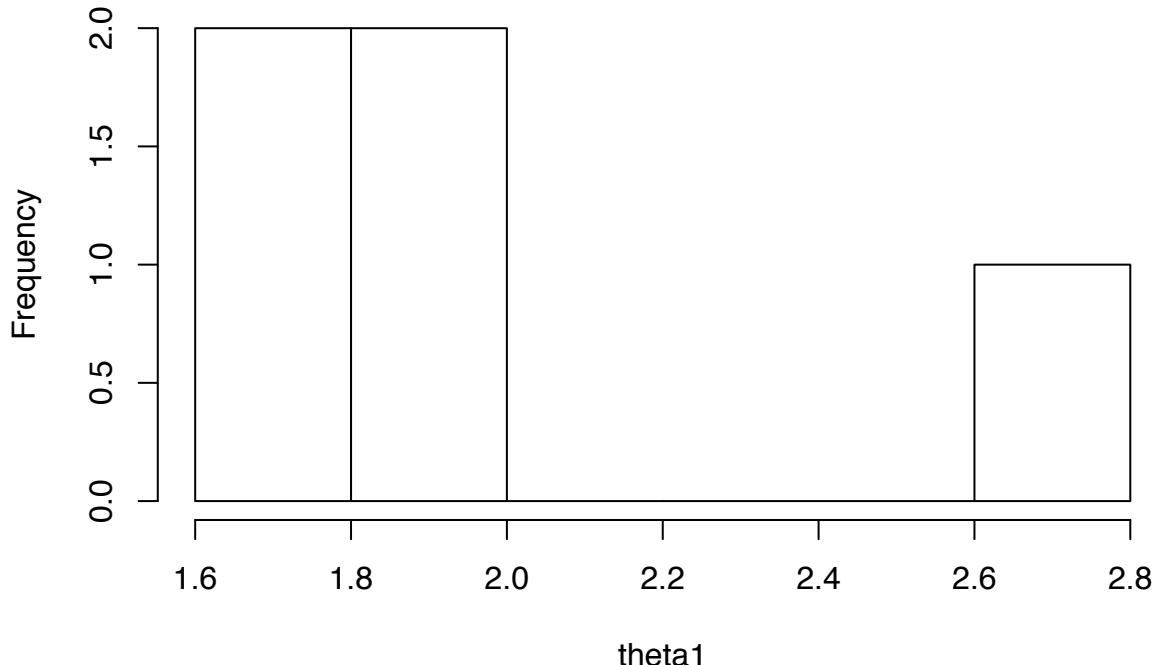
summary(theta1)

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
##      1.727    1.727   1.860    1.959   1.860    2.621

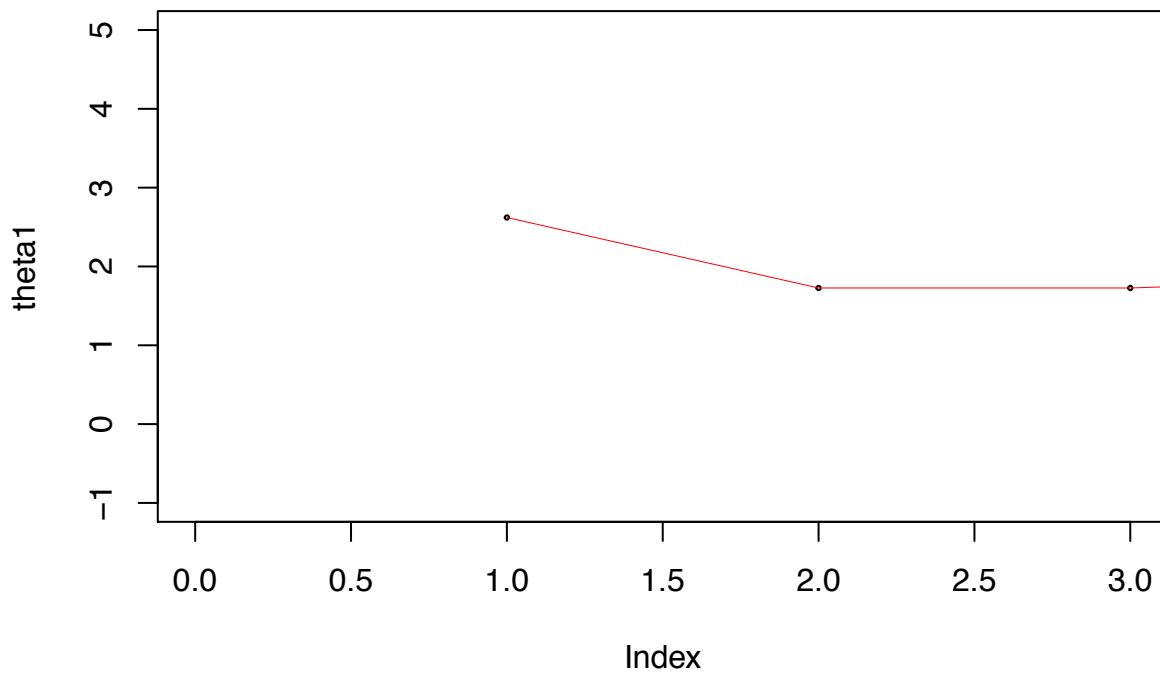
```

```
hist(theta1)
```

Histogram of theta1

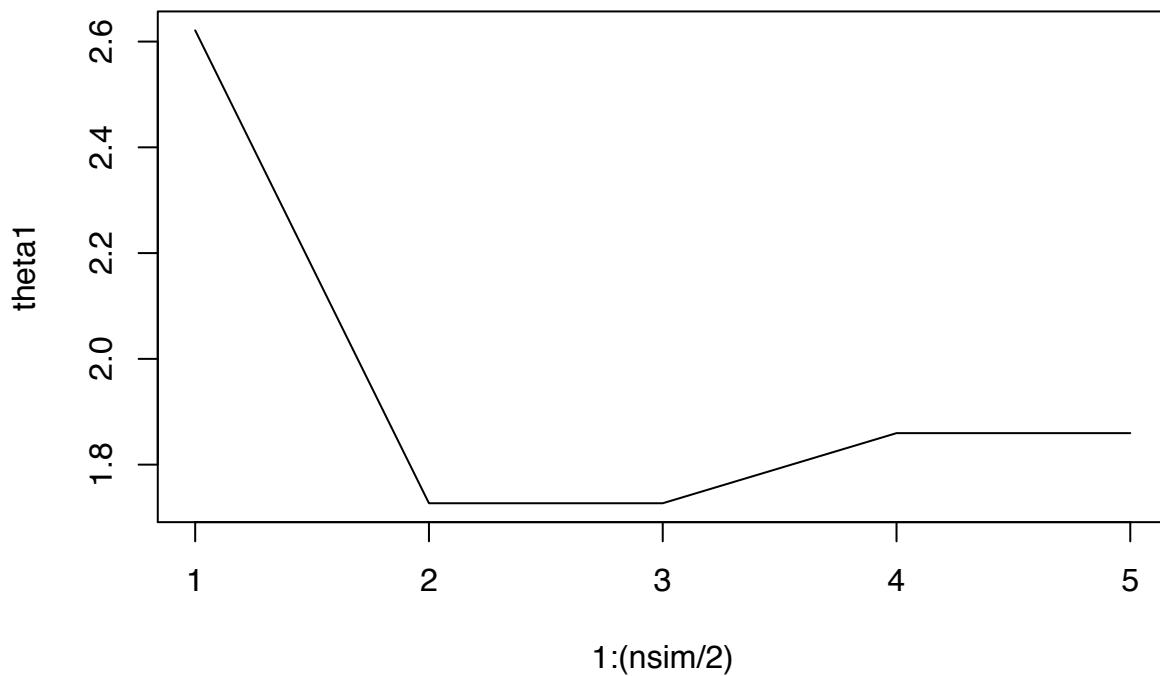


```
plot(theta1,pch=1,cex=0.3,xlim=c(0,3),ylim=c(-1,5))  
lines(theta1,lwd=0.5,col="red")
```



```
plot(1:(nsim/2),theta1, type ='l', main = "d=1" )
```

**d=1**



Rispetto ai valori teorici si sa che

$$E(\theta|\theta_0 = 3) = 0 + \frac{1/4}{1 + 1/4} \times 3 = 2.8$$

e

$$Var(\theta|\theta_0 = 3) = \frac{1}{1 + 1/4} = 0.8$$

Occorre stabilire un numero di realizzazioni `nsim` molto elevato in quanto occorre essere certi di raggiungere la steady state distribuzione del processo stocastico.

Nel seguente esempio utilizzando `nsim = 10^6` si riporta l'istogramma rispetto ai valori accettati, il trace plot.

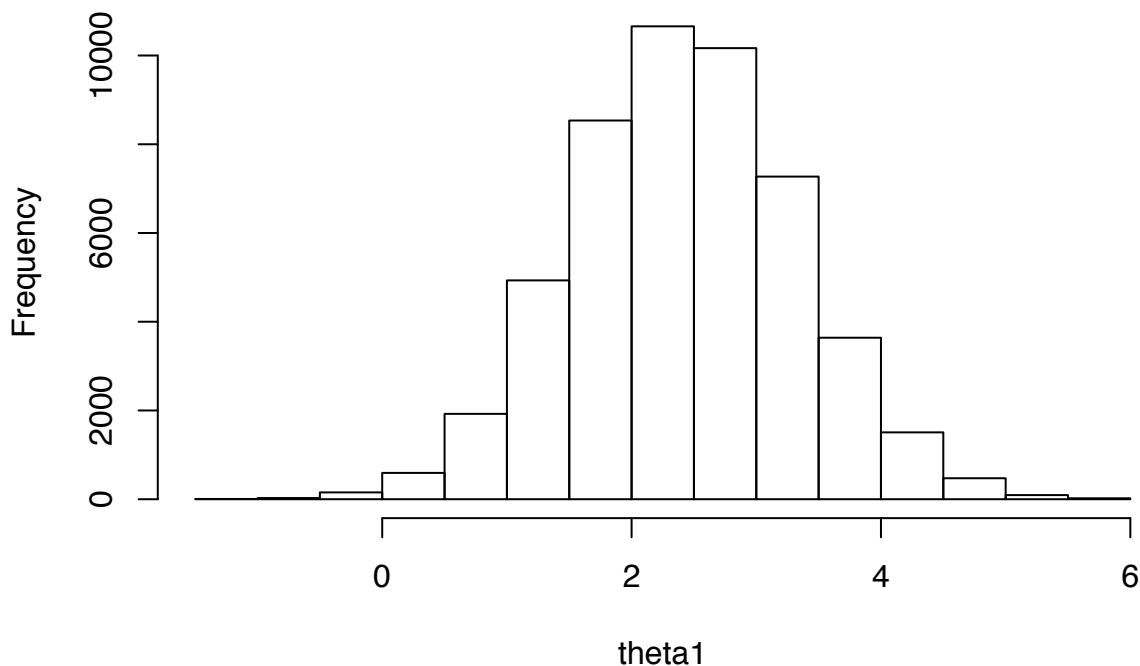
```
nsim <- 10^5
theta[1] <- theta0
set.seed(173)
for(i in 2:nsim){
  theta.p <- theta[i-1] + rnorm(1,0,d)
  ratio <- dnorm(theta0, theta.p, sigma)*
    dnorm(theta.p, mu, tau) /
    (dnorm(theta0, theta[i-1], sigma) *
      dnorm(theta[i-1], mu, tau))
  evento <- rbinom(1,1,min(ratio,1))

  theta[i] <- if(evento==1) theta.p else theta[i-1]
}

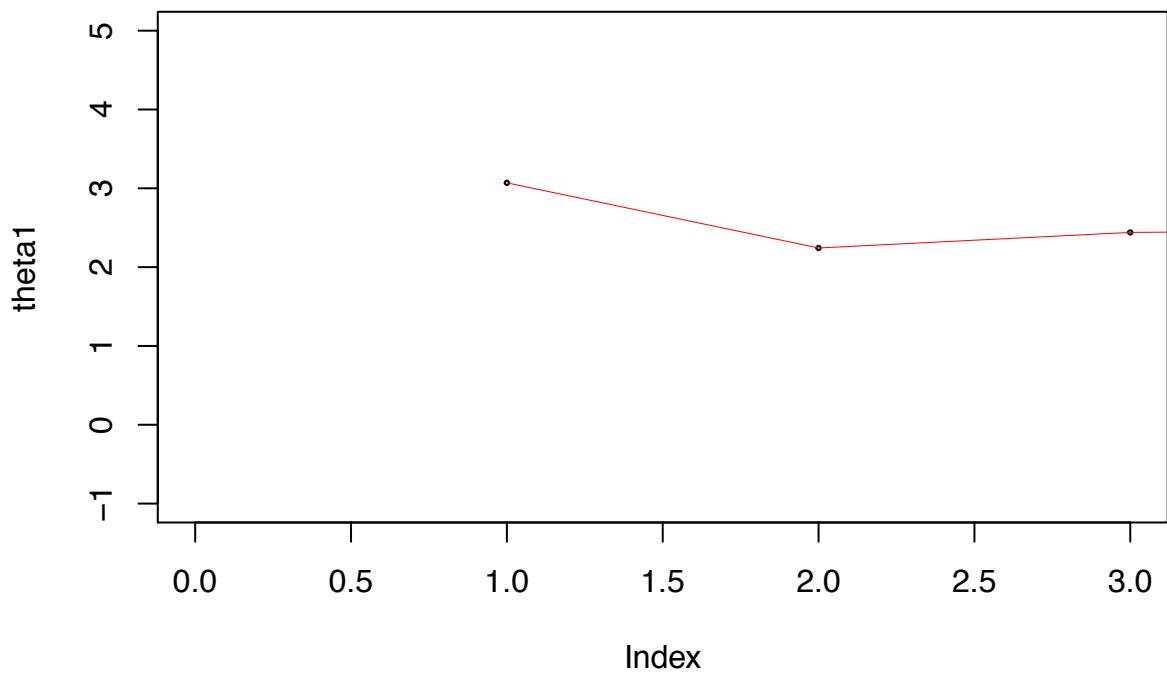
theta1 <- theta[-(1:(nsim/2))]
summary(theta1)
```

```
##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
## -1.404   1.809   2.416   2.422   3.026   5.977
var(theta1)
## [1] 0.8129868
hist(theta1)
```

Histogram of theta1

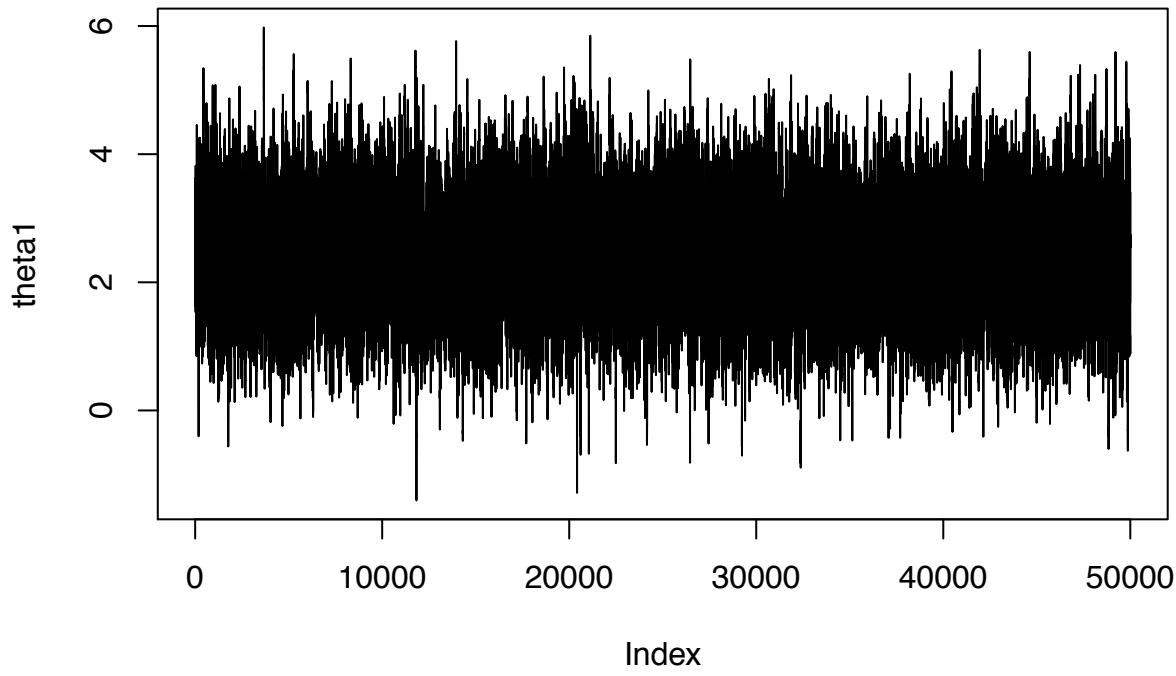


```
plot(theta1,pch=1,cex=0.3,xlim=c(0,3),ylim=c(-1,5))
lines(theta1,lwd=0.5,col="red")
```



```
plot(theta1,type ='l', main = "d=1")
```

**d=1**



Si nota dal summary che i valori per la media e la deviazione standard sono simili a quelli riferiti ai valori teorici.