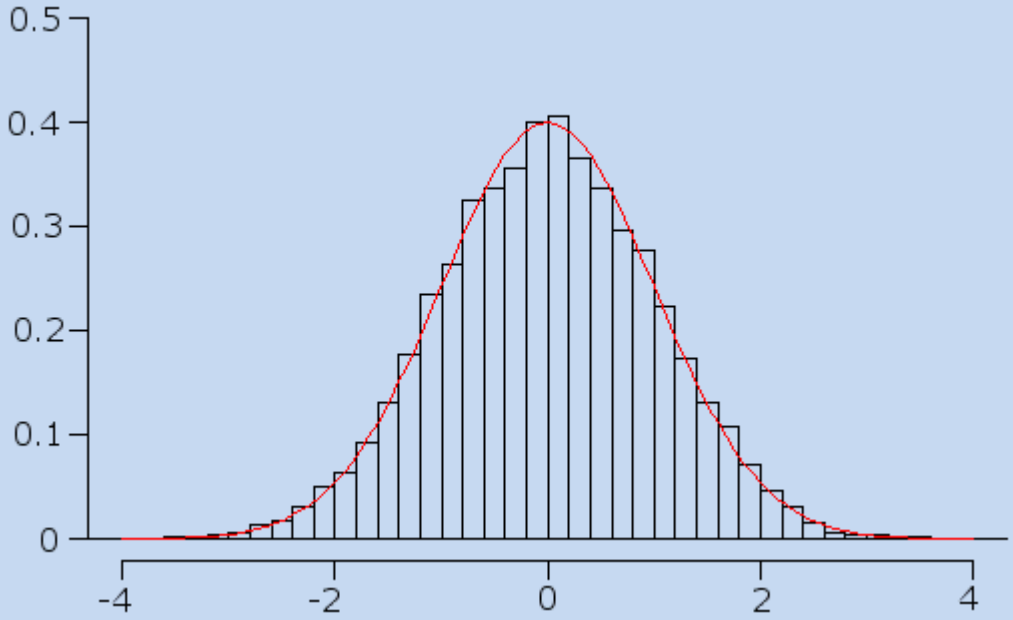


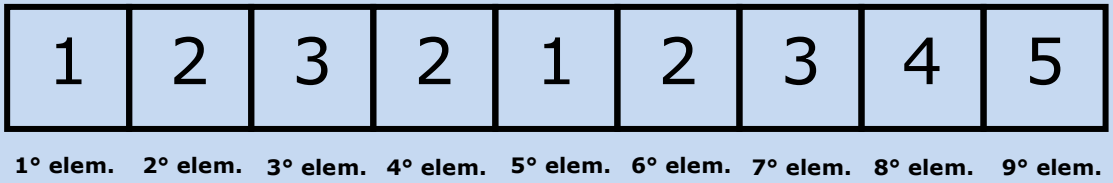
Calcolo delle probabilità con il linguaggio R

(con un'introduzione "per esempi" al linguaggio)

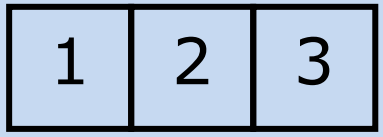
paolo@paololazzarini.it

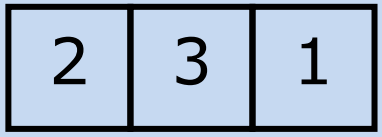
Vettori

Un *vettore*, in informatica, è una struttura ordinata di dati: potete pensare ad una sequenza di celle numerate, ognuna delle quali contiene un elemento. Gli elementi possono essere *numeri* o *stringhe* ma devono essere tutti dello stesso tipo. Ecco qualche esempio:

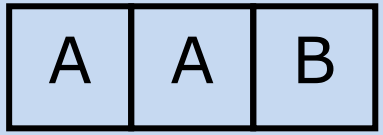
u= 

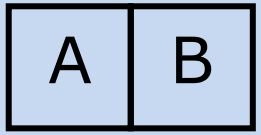
lunghezza(u)=9

x = 

y = 

x è diverso da y

r = 

s = 

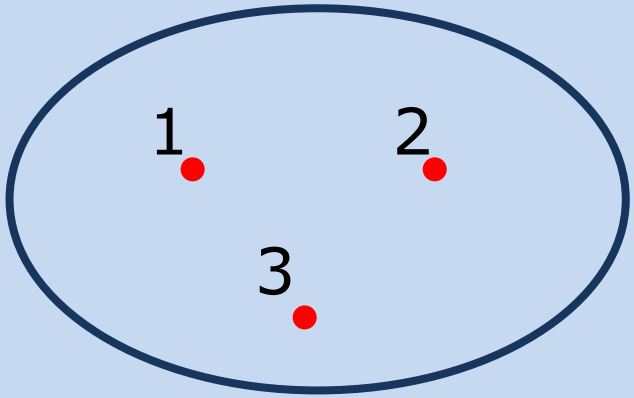
r è diverso da s

Non confondere vettori con insiemi

$A = \{1, 2, 3\}$ $B = \{2, 3, 1\}$

A è uguale a B

A=B



In un diagramma di Venn è ininfluente l'ordine con cui sono disposti gli elementi dell'insieme.

$A = \{a, a, b\}$ $B = \{a, b\}$

A è uguale a B

La definizione formale di uguaglianza di insiemi: A=B se ogni elemento di A appartiene a B e, viceversa, ogni elemento di B appartiene ad A.

Come creare vettori in R

Esempio 1 Uso dei due punti:

```
Console D:/R/
> x=1:10
> x
 [1]  1  2  3  4  5  6  7  8  9 10
```

Esempio 2 Concatenazione *c(…)*:

```
Console D:/R/
> x=c(1,3,12,20)
> x
[1]  1  3 12 20
> y=c("paolo","fabio","giovanni")
> y
[1] "paolo"    "fabio"    "giovanni"
```

Nota: gli elementi del vettore x sono numeri, gli elementi del vettore y sono stringhe (rappresentate tra virgolette).

Esempio 3 Funzione *seq(…)* (sequenza):

```
Console D:/R/
> x=seq(from=0, to=1, by=0.1)
> x
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
> length(x)
[1] 11
```


Nota: i nomi dei parametri, nel nostro caso *from*, *to*, *by*, possono essere omessi; avremmo potuto scrivere *x=seq(0, 1, 0.1)*, rispettando l'ordine dei parametri.
Nota: la funzione *length(x)* fornisce la lunghezza del vettore x.

Esempio 4 Funzione *rep(…)* (ripetizione):

```
Console D:/R/
> a=rep("paolo", times=3)
> a
[1] "paolo" "paolo" "paolo"
> noquote(a)
[1] paolo paolo paolo
> b=c(rep(1,3),rep(2,4))
> b
[1] 1 1 1 2 2 2 2
```

Nota: la funzione *noquote()* consente di rappresentare gli elementi del vettore *a* (che sono stringhe) privi di virgolette.

Test uguaglianza

Due modi sostanzialmente diversi per verificare l'uguaglianza di due oggetti:

segno di uguaglianza == (valutazione elemento per elemento)
funzione *identical()* (valutazione "in blocco")

Esempio 1

```
Console D:/R/
> x=c(1,2,3); y=c(2,1,3)
> x==y
[1] FALSE FALSE  TRUE
> identical(x,y)
[1] FALSE
```

```
Console D:/R/
> x=c(1,2,1,3,3)
> x==1
[1]  TRUE FALSE  TRUE FALSE FALSE
```

```
Console D:/R/
> x=2
> x==2
[1] TRUE
```

```
Console D:/R/
> (2+3)^2==2^2+3^2
[1] FALSE
> (2*3)^2==2^2*3^2
[1] TRUE
```

Attenzione: il simbolo = indica un'assegnazione, il simbolo == indica un test di uguaglianza.

Come estrarre elementi di un vettore

Esempio 1 Estrazione dell'elemento i-esimo del vettore x: *x[i]*

```
Console D:/R/
> x=c(2,3,5,7,11)
> x
[1]  2  3  5  7 11
> x[2]
[1] 3
> x[5]
[1] 11
```

Esempio 2 Estrazione degli elementi di posto i:j, cioè da i a j:

```
Console D:/R/
> x=c(5,3,7,1,4,3,7)
> x
[1] 5 3 7 1 4 3 7
> x[3:6]
[1] 7 1 4 3
```

Esempio 3 Estrazione mediante una condizione:

```
Console D:/R/
> x=c(1,2,3,1,1,5,4,3,2,1,1,7,9,5,6,6,3,2,1)
> x[x==2] #estrae gli elementi uguali a 2
[1] 2 2 2
> length(x[x==2]) #quanti sono?
[1] 3
```

```
Console D:/R/
> x[x>3] #estrae gli elementi maggiori di 3
[1] 5 4 7 9 5 6 6
> length(x[x>3]) #quanti sono?
[1] 7
```

```
Console D:/R/
> x[x<3 | x>6] # | è l'operatore logico OR
 [1] 1 2 1 1 2 1 1 7 9 2 1
> x[3<=x & x<=6] # & è l'operatore logico AND
[1] 3 5 4 3 5 6 6 3
```

Nota: qui si è usato il simbolo # per introdurre un commento, inoltre sono stati utilizzati gli operatori logici OR e AND rappresentati rispettivamente dai simboli "|" e "&" che si trovano sulla tastiera.

Come modificare vettori in R

Esempio 1 Cambiare il valore di un dato elemento:

```
Console D:/R/
> x=1:10
> x
 [1]  1  2  3  4  5  6  7  8  9 10
> x[3]=7
> x
 [1]  1  2  7  4  5  6  7  8  9 10
```

Qui viene cambiato il valore dell'elemento di x al posto 3; si può anche creare un elemento non presente, ad esempio x[11]=1.

Esempio 2 Eliminare un dato elemento:

```
Console D:/R/
> x=1:10
> x
 [1]  1  2  3  4  5  6  7  8  9 10
> x=x[-3]
> x
[1]  1  2  4  5  6  7  8  9 10
> length(x)
[1] 9
```

Qui viene eliminato l'elemento di x al posto 3 (notare la riassegnazione).

Somme e prodotti

Esempio 1 Somma e prodotto degli elementi di un vettore:

```
Console D:/R/
> x=c(1,2,4)
> sum(x)
[1] 7
> prod(x)
[1] 8
>
```

Esempio 2 Somma e prodotto di vettori:

```
Console D:/R/
> x=c(1,2,4)
> y=c(3,4,5)
> x+y
[1] 4 6 9
> x*y
[1]  3  8 20
```

Attenzione, anche il prodotto è elemento per elemento; per il prodotto scalare si usa un altro simbolo.

Esempio 3 Prodotto di un numero per un vettore:

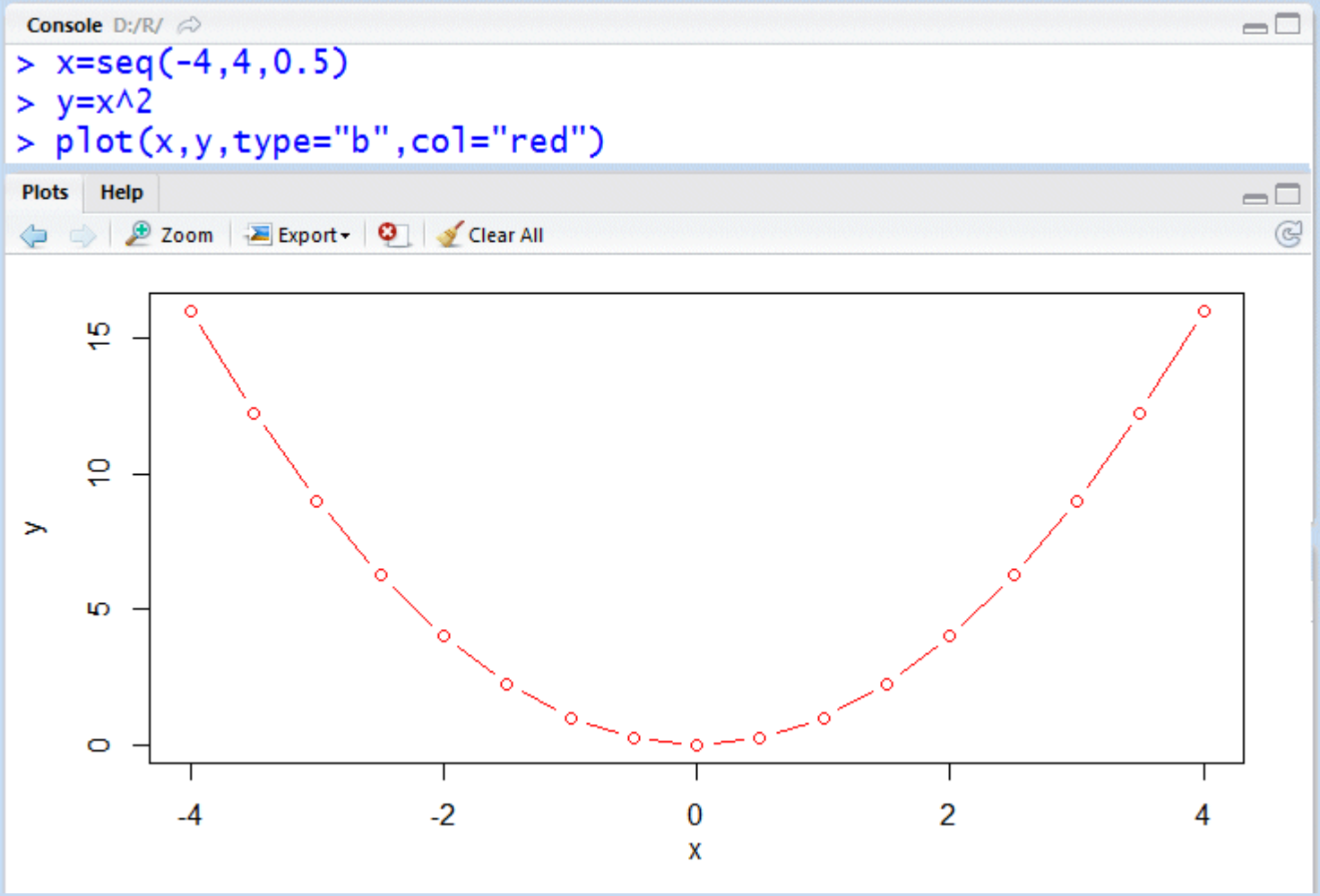
```
Console D:/R/
> x=c(2,4,7)
> 2*x
[1]  4  8 14
> (1/2)*x
[1] 1.0 2.0 3.5
> x/2
[1] 1.0 2.0 3.5
```


Esempio 4 Media e varianza degli elementi di un vettore:

```
Console D:/R/
> x=1:10
> media=sum(x)/length(x)
> media
[1] 5.5
> varianza=sum((x-media)^2/length(x))
> varianza
[1] 8.25
```

Nota: *x−media* è un operazione del tipo *vettore – numero*, ad ogni elemento del vettore sarà sottratto il numero.
Nota: per la media e la varianza ci sono due funzioni primitive di R, *mean()* e *var()*; però, attenzione, per R la varianza è quella campionaria e non quella della popolazione.

Esempio 5 Grafico, per punti, di una funzione:



Nota: qui abbiamo utilizzato la funzione

plot(x, y)

dove x e y sono due vettori di stessa lunghezza. Gli elementi del vettore y, nel nostro caso, sono i quadrati degli elementi del vettore x. Il parametro *type* è impostabile a "p" (solo punti), "l" (solo segmenti che collegano i punti), "b" (punti e segmenti). Il parametro *col* serve a impostare il colore del grafico.

Tabella delle frequenze

Vedi anche: Tabella delle frequenze di classe

Per ottenere una tabella di frequenze assolute utilizzare la funzione *table(x)* dove x è un vettore; è anche facile ottenere una tabella di frequenze relative dividendo *table(x)* per *length(x)*.

Esempio 1 Tabella di frequenze assolute:

```
Console D:/R/
> x=c(1,2,1,3,4,3,3,5,7,5,1,3,4,3,4)
> tabella=table(x)
> tabella
x
1 2 3 4 5 7
3 1 5 3 2 1
```

```
Console D:/R/
> length(x)
[1] 15
> sum(tabella)
[1] 15
```

Nota: la somma delle frequenze assolute è uguale al numero di elementi del vettore x.

Esempio 2 Tabella di frequenze relative:

```
Console D:/R/
> x=c(1,2,1,3,4,3,3,5,7,5,1,3,4,3,4)
> tabella=table(x)/length(x)
> tabella
x
         1          2          3          4          5          7
0.20000000 0.06666667 0.33333333 0.20000000 0.13333333 0.06666667
> round(tabella,3)
x
    1     2     3     4     5     7
0.200 0.067 0.333 0.200 0.133 0.067
> sum(tabella)
[1] 1
```

Nota: la somma delle frequenze relative è uguale a 1.
Nota: la funzione *round()* ci consente di effettuare degli arrotondamenti, nel nostro caso alla terza cifra decimale.

Esempio 3 Tabella di frequenze per un carattere qualitativo, ad esempio *colore occhi* (A=azzurri, M=marroni, ecc.):

```
Console D:/R/
> x=c("A","M","M","N","V","M","A","A","V","N")
> table(x)
x
A M N V
3 3 2 2
```

Tabella delle frequenze di classe

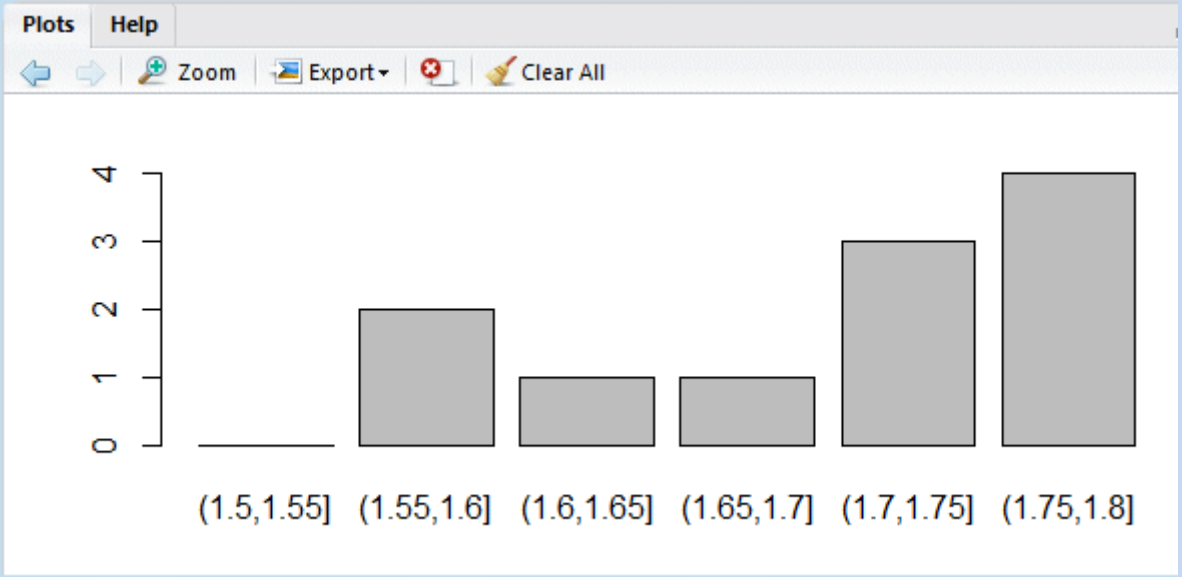
Per tabulare le frequenze di classe utilizzeremo le funzioni *cut()* e *table()*

Esempio 1 I valori da tabulare sono le altezze di 11 ragazzi:

```
Console D:/R/
> x=c(1.58, 1.77, 1.73, 1.75, 1.61, 1.80, 1.76, 1.70, 1.59, 1.71, 1.76)
> classi=cut(x,breaks=seq(1.50,1.80,0.05))
> classi
 [1] (1.55,1.6] (1.75,1.8] (1.7,1.75] (1.7,1.75] (1.6,1.65]
 [6] (1.75,1.8] (1.75,1.8] (1.65,1.7] (1.55,1.6] (1.7,1.75]
[11] (1.75,1.8]
6 Levels: (1.5,1.55] (1.55,1.6] (1.6,1.65] ... (1.75,1.8]
```


Il parametro *breaks* della funzione *cut()* consente di impostare le classi, nel nostro caso si va da 1.50 a 1.80 con passo 0.05; le classi sono aperte a sinistra e chiuse a destra (*right=T*) per cui il primo valore 1.5, se presente, non verrebbe considerato (*include.lowest=F)*. Volendo si può modificare questa impostazione mediante il parametro *right=F/T* e il parametro *include.lowest=F/T*. Notare che la funzione *cut* crea per ogni valore la sua classe, per cui le classi sono tante quante i valori e possono ripetersi. Le classi sono chiamate *livelli* e sono indicate nell'ultima riga dell'output. Ora è facile ottenere la tabella e il relativo diagramma a barre (nella finestra dell'output grafico di R Studio):

```
Console D:/R/
> tavola=table(classi)
> tavola
classi
(1.5,1.55] (1.55,1.6] (1.6,1.65] (1.65,1.7] (1.7,1.75] (1.75,1.8]
         0          2          1          1          3          4
> barplot(tavola)
```



Campionamento casuale

La funzione *sample()* consente l'estrazione di un campione casuale, di dimensione specificata (parametro *size*), da un vettore x; tale estrazione può essere con o senza reimmissione (parametro *replace=T/F*). Se non è specificata una distribuzione di probabilità (parametro *prob*) ogni elemento di x ha la stessa probabilità di essere estratto (distribuzione uniforme).

Esempio 1 Simula il lancio di una moneta equa per 100 volte:

```
Console D:/R/
> x=c("T","C")
> campione=sample(x,size=100,replace=TRUE)
> campione
  [1] "T" "T" "C" "T" "T" "T" "T" "C" "C" "C" "T" "T" "T" "C" "C" "T" "T" "T" "C"
 [20] "T" "C" "C" "T" "C" "C" "C" "C" "T" "C" "C" "T" "T" "T" "T" "T" "C" "T" "T"
 [39] "T" "C" "C" "T" "C" "T" "C" "T" "C" "C" "C" "C" "C" "T" "T" "T" "C" "C" "C"
 [58] "T" "T" "C" "T" "T" "C" "T" "T" "T" "T" "C" "T" "T" "C" "C" "C" "T" "T" "T"
 [77] "C" "C" "C" "T" "T" "T" "C" "T" "C" "C" "T" "C" "C" "C" "T" "C" "C" "C" "T"
 [96] "T" "T" "C" "T" "T"
```

Nota: in questo caso deve essere evidentemente *replace=TRUE*, altrimenti alla terza estrazione nel vettore x non avremmo più elementi (perciò, se poniamo *replace=FALSE*, R fornisce un messaggio d'errore).

Ora è facile determinare le frequenze assolute e relative:

```
Console D:/R/
> table(campione)
campione
 C  T
47 53
> table(campione)/length(campione)
campione
   C    T
0.47 0.53
```


Esempio 2 Simula il lancio di una moneta di trucco p=0.4 per 100 volte:

```
Console D:/R/
> x=c("T","C")
> distr_prob=c(0.4,0.6)
> campione=sample(x,size=100,replace=TRUE,prob=distr_prob)
> campione
  [1] "T" "C" "C" "T" "C" "T" "C" "C" "T" "T" "C" "T" "T" "T" "C"
 [16] "C" "T" "C" "C" "C" "T" "T" "T" "C" "T" "C" "T" "T" "C" "C"
 [31] "C" "C" "C" "T" "C" "C" "C" "T" "C" "T" "C" "C" "C" "T" "C"
 [46] "C" "C" "C" "C" "C" "C" "T" "T" "C" "C" "T" "C" "C" "C" "T"
 [61] "C" "C" "C" "T" "T" "T" "C" "C" "T" "C" "T" "T" "C" "T" "C"
 [76] "C" "C" "T" "C" "T" "T" "C" "C" "T" "C" "T" "T" "T" "T" "C"
 [91] "T" "C" "T" "C" "T" "T" "C" "T" "T" "T"
> table(campione)/length(campione)
campione
   C    T
0.55 0.45
```

Esempio 3 Simula l'estrazione di 3 biglie da un'urna che contiene 5 biglie nere e 3 biglie bianche:

```
Console D:/R/
> x=c(rep("N",5),rep("B",3))
> x
[1] "N" "N" "N" "N" "N" "B" "B" "B"
> sample(x,size=3,replace=FALSE)
[1] "B" "N" "N"
> sample(x,size=3,replace=FALSE)
[1] "B" "B" "N"
> sample(x,size=3,replace=FALSE)
[1] "N" "B" "N"
> sample(x,size=3,replace=FALSE)
[1] "N" "B" "N"
> sample(x,size=3,replace=FALSE)
[1] "N" "N" "B"
> sample(x,size=3,replace=FALSE)
[1] "N" "N" "N"
```


Nota: qui il comando sample viene eseguito più volte, ogni volta sono estratte 3 biglie senza reimmissione (replace=FALSE); vedremo tra poco come rendere automatico il processo di ripetizione dell'esperimento (ciclo *for …*).

Esempio 4 Simula la variabile casuale discreta X

valori	X=1	X=2	X=3	X=4	X=5
prob.	0.1	0.3	0.3	0.1	0.2

con distribuzione non uniforme:


```
Console D:/R/
> x=1:5
> distr_prob=c(0.1, 0.3, 0.3, 0.1, 0.2)
> sum(distr_prob)
[1] 1
> X=sample(x, size=1, prob=distr_prob)
> X
[1] 3
> X=sample(x, size=1, prob=distr_prob)
> X
[1] 2
> X=sample(x, size=1, prob=distr_prob)
> X
[1] 2
```

Nota: R è un linguaggio case sensitive: qui la variabile x (minuscolo) indica il vettore dei valori che la variabile casuale X può assumere, mentre X (maiuscolo) indica una particolare realizzazione di X.

E’ anche facile rappresentare graficamente, con un diagramma a barre, la distribuzione di probabilità della v.c. X:

```
Console D:/R/
> barplot(distr_prob,names.arg=1:5)
```

Questo è l’output nella finestra grafica di R Studio:

Nota: il parametro *names.arg* è il vettore delle etichette che saranno visualizzate sotto ciascuna barra.

Dataframe

Un oggetto *dataframe* è una tabella costituita da più vettori di stessa lunghezza (colonne della tabella). Per creare un dataframe useremo la funzione *data.frame()* oppure l'editor fornito da R.

Esempio 1 Dataframe costituito da 4 colonne (vettori): nome, altezza, peso, sesso:

```
Console D:/R/
> nome=c("Paolo","Fabio","Maria","Luca","Elena")
> altezza=c(173,177,158,170,154)
> peso=c(72,65,48,56,50)
> sesso=c("M","M","F","M","F")
> df=data.frame(nome,altezza,peso,sesso,stringsAsFactors=FALSE)
> df
   nome altezza peso sesso
1 Paolo     173   72     M
2 Fabio     177   65     M
3 Maria     158   48     F
4  Luca     170   56     M
5 Elena     154   50     F
```

Nota: ai nostri fini è opportuno porre il parametro *stringsAsFactors* uguale a FALSE.

Per ottenere le singole colonne (vettori) utilizzeremo l'operatore $ o l'operatore [[…]]:

```
Console ~/R/
> df$nome
[1] "Paolo" "Fabio" "Maria" "Luca"  "Elena"
> df$sesso
[1] "M" "M" "F" "M" "F"
> df$nome[1]
[1] "Paolo"
> df$nome[3]
[1] "Maria"
> df[[1]]
[1] "Paolo" "Fabio" "Maria" "Luca"  "Elena"
> df[[4]]
[1] "M" "M" "F" "M" "F"
> df[[1]][1]
[1] "Paolo"
> df[[1]][3]
[1] "Maria"
```


Per selezionare parti di una dataframe utilizzeremo la funzione *subset()* impostando il parametro *subset* (condizione) ed eventualmente il parametro *select* (per selezionare colonne, ad esempio *select=c(nome, sesso)*).

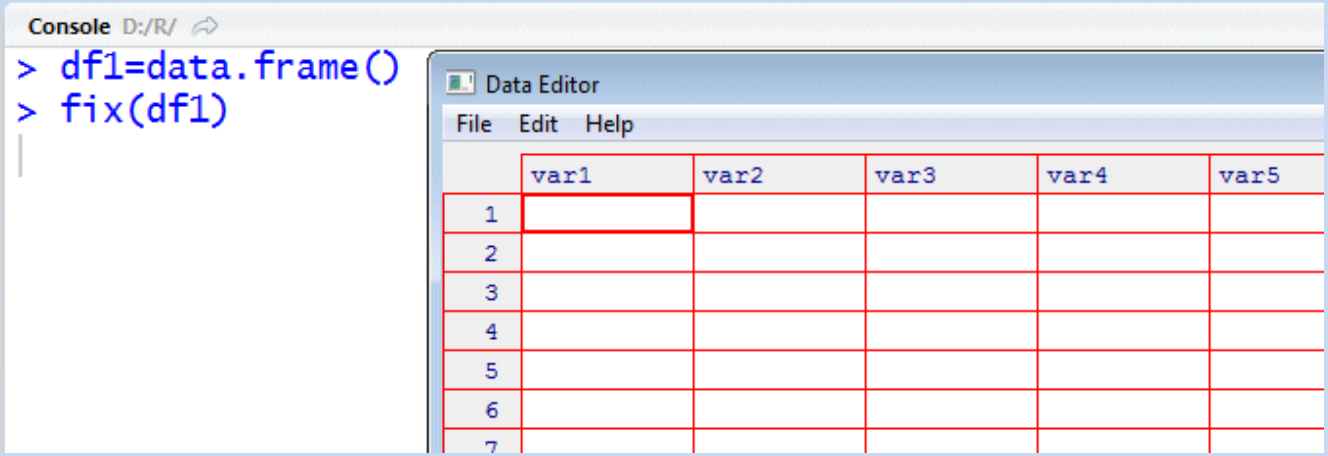
Esempio 2 Estraiamo dal dataframe *df* dell'es. 1 i dati riguardanti le femmine con altezza maggiore di 154:

```
Console D:/R/
> subset(df,subset=(altezza>154 & sesso=="F"))
   nome altezza peso sesso
3 Maria     158   48      F
```

Esempio 3 Riferendoci al dataframe df dell'es. 1, calcoliamo l'altezza media dei maschi:

```
Console D:/R/
> sub_df=subset(df,subset=sesso=="M")
> mean(sub_df$altezza)
[1] 170
```


Esempio 4 Utilizziamo la funzione *fix()* per creare un nuovo dataframe mediante un editor:



Variabili casuali (o variabili aleatorie)

Esempio 1 X = "valore che si presenta lanciando un dado equo"

La *distribuzione di probabilità* della variabile X è

X	1	2	3	4	5	6
prob.	1/6	1/6	1/6	1/6	1/6	1/6

Esempio 2 Si lancia una moneta equa. Poniamo M=1 se esce TESTA altrimenti M=0.

La distribuzione di probabilità della variabile M è

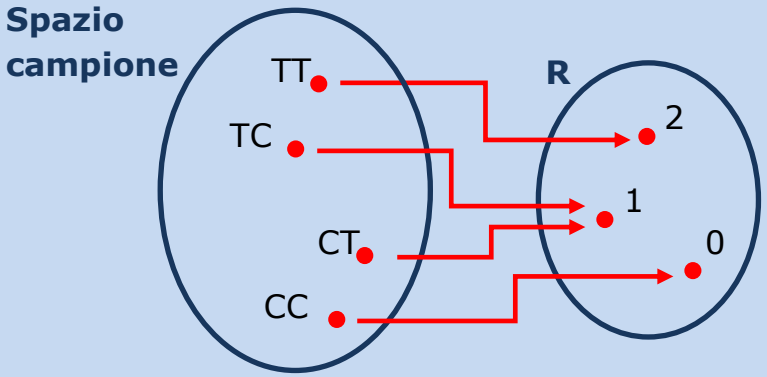
M	0	1
prob.	1/2	1/2

Entrambe le variabili X ed M sono *discrete* (assumono un numero finito o numerabile di valori) e hanno distribuzione di probabilità *uniforme* (i valori o *realizzazioni* della variabile hanno tutte la stessa probabilità). Notare che la somma delle probabilità di una distribuzione è sempre 1 (condizione di normalizzazione). Nei prossimi esempi studieremo, con l'aiuto di R, la distribuzione di probabilità di alcune v.c. (e, come vedremo, saranno distribuzioni non uniformi).

Nota Dietro ad ogni variabile casuale c'è un esperimento aleatorio; **ad ogni** risultato dell'esperimento la variabile casuale associa un numero reale. In questo senso una variabile casuale è una funzione. Consideriamo ad esempio l'esperimento aleatorio che consiste nel lanciare due monete eque e la variabile casuale

X = "numero di TESTA che si presentano"

La situazione è illustrata nel diagramma seguente:



X	0	1	2
prob.	1/4	2/4	1/4

Osservate che i tre eventi X=0, X=1, X=2 esauriscono tutte le possibilità (uno di questi eventi deve necessariamente verificarsi), ne segue che la loro unione è l'evento certo; sono inoltre eventi evidentemente disgiunti (incompatibili). Quindi la somma delle probabilità di questi tre eventi è uguale a 1:

$$p(X=0) + p(X=1) + p(X=2) = 1/4 + 1/2 + 1/4 = 1$$

Questo fatto è vero in generale per qualsiasi v.c.: la somma delle probabilità di una distribuzione è sempre 1 (condizione di normalizzazione).

Esempio 3 Calcolare la distribuzione di probabilità della variabile casuale S = "somma dei valori che si presentano lanciando due dadi equi"

E' arrivato il momento di scrivere il nostro primo programma (script) in R; digiteremo il programma nella finestra di scripting di R Studio:

Untitled1* ×

Source on Save

```
#somma di due dadi
cat("\14")

griglia=expand.grid(dado1=1:6,dado2=1:6)
somme=griglia$dado1+griglia$dado2

distr_prob=table(somme)/length(somme)
print(round(distr_prob,3))
barplot(distr_prob)
```

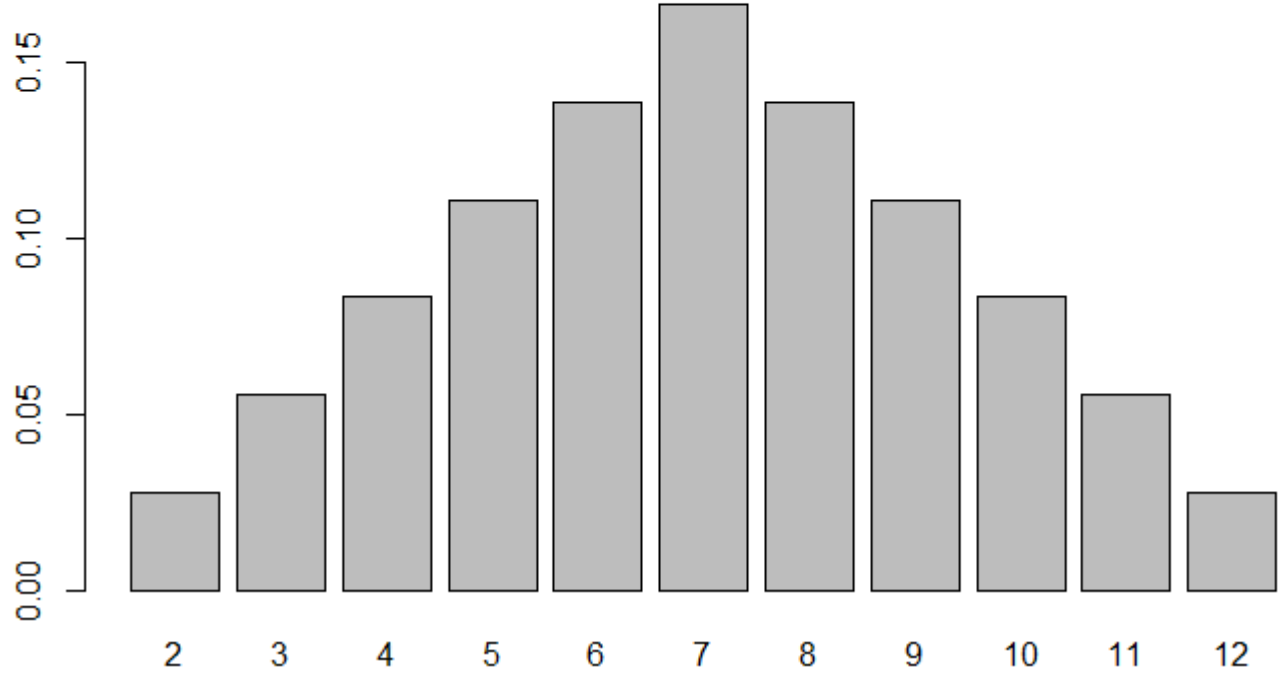
E questo è l'output:

Console D:/Testi/Circolo 2014-15/

```
somme
    2     3     4     5     6     7     8     9    10    11    12
0.028 0.056 0.083 0.111 0.139 0.167 0.139 0.111 0.083 0.056 0.028
```

Plots Help

Zoom Export Clear All



Esaminiamo in ora in dettaglio il programma.

1. Le funzioni *cat()* e *print()* servono entrambe per visualizzare dati nella console, tuttavia solo la seconda garantisce una corretta rappresentazione dell'oggetto da stampare (ad esempio una tabella). La funzione *cat()* consente di concatenate sulla stessa riga di output numeri e stringhe. Nel nostro caso il comando *cat("\14")* serve semplicemente a ripulire la console.

2. La funzione *expand.grid()* prende in input i due vettori *dado1* e *dado2*, cioè i due vettori di valori da 1 a 6, e fornisce in output tutte le 36 possibili coppie ordinate di valori nella forma di dataframe; qui a fianco vediamo concretamente qual è l'output. E' importante capire che ognuna di queste 36 coppie ha la stessa probabilità di presentarsi tenendo conto che i due dadi sono equi e che il risultato ottenuto con un dado non condiziona il risultato ottenuto col secondo (indipendenza). Osservare però che una stessa somma può essere ottenuta più volte, ad esempio le coppie (6, 3), (3, 6), (4, 5), (5, 4) generano tutte una somma pari a 9; quindi la probabilità che la somma sia 9 è 4/36=1/9=0.1111…

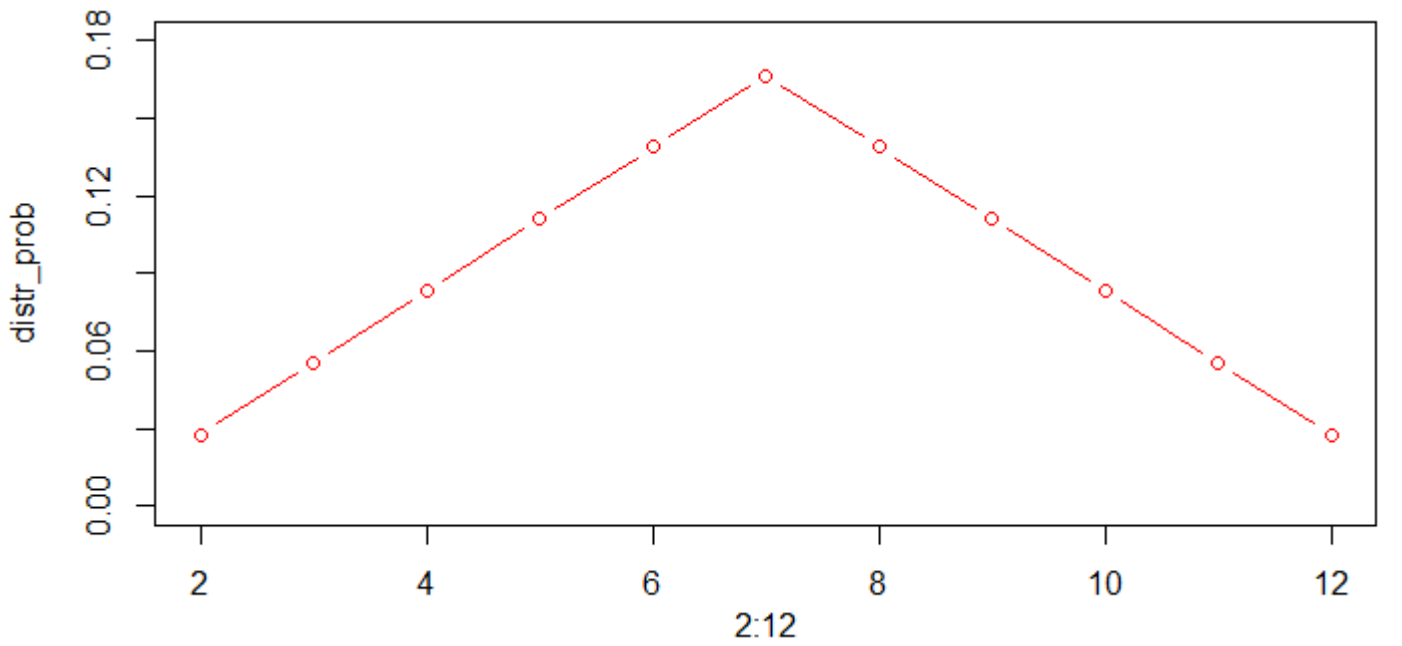
3. *somme* è il vettore delle somme degli elementi di ciascuna coppia:

```
Console ~/R/
> griglia
   dado1 dado2
1      1     1
2      2     1
3      3     1
4      4     1
5      5     1
6      6     1
7      1     2
8      2     2
9      3     2
10     4     2
11     5     2
12     6     2
13     1     3
14     2     3
15     3     3
16     4     3
17     5     3
18     6     3
19     1     4
20     2     4
21     3     4
22     4     4
23     5     4
24     6     4
25     1     5
26     2     5
27     3     5
28     4     5
29     5     5
30     6     5
31     1     6
32     2     6
33     3     6
34     4     6
35     5     6
36     6     6
```

```
Console D:/R/
> somme
 [1]  2  3  4  5  6  7  3  4  5  6  7  8  4  5  6  7  8  9  5  6  7
[22]  8  9 10  6  7  8  9 10 11  7  8  9 10 11 12
```

E’ facile modificare il programma per ottenere la distribuzione di probabilità delle somme di 3, 4, 5 dadi (provate a farlo!). Un altro tipo di grafico utile a rappresentare le distribuzioni di probabilità si può ottenere mediante il comando *plot()*:

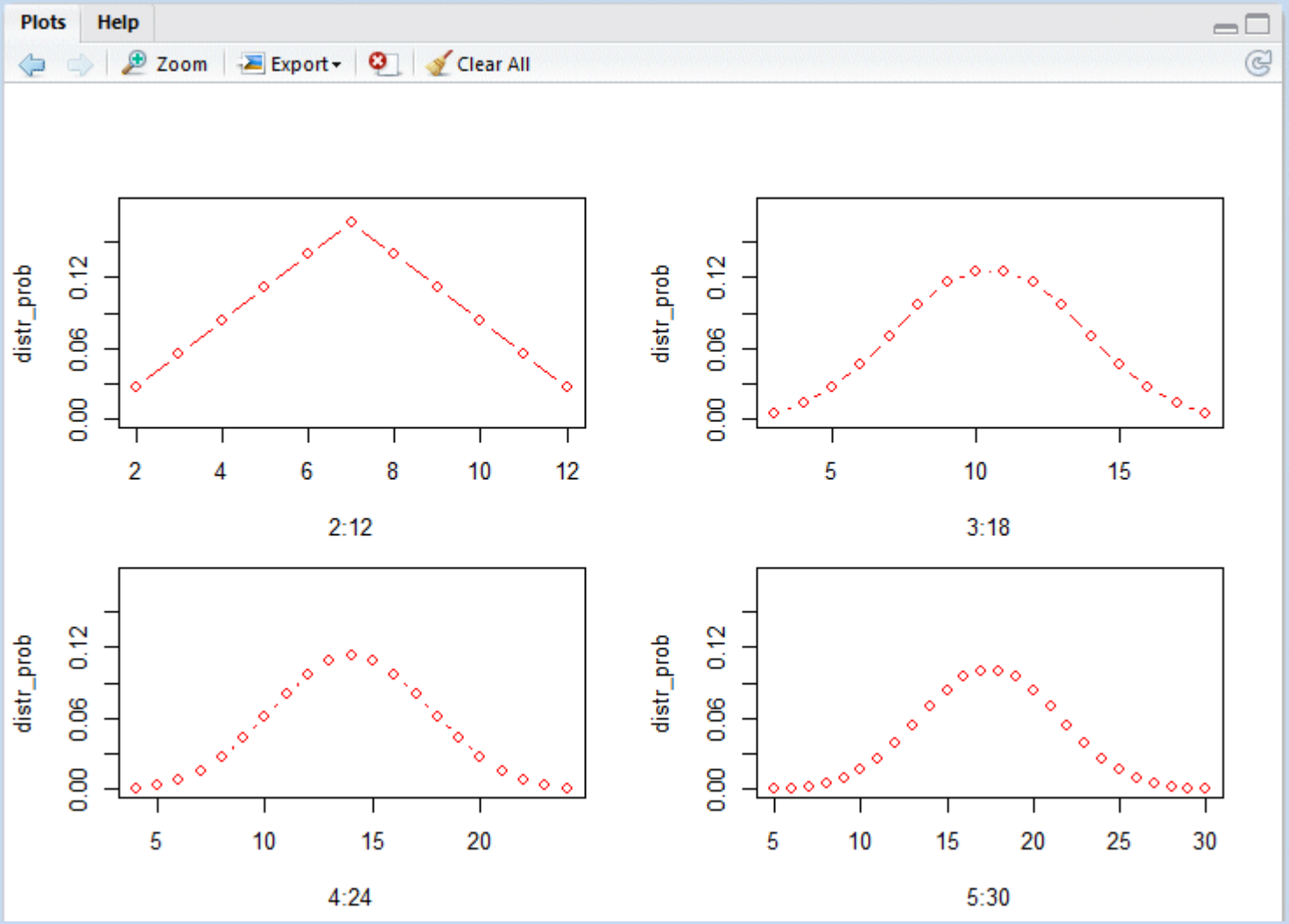
```
Console D:/R/
> plot(2:12,distr_prob,type="b",col="red",ylim=c(0,0.18))
> axis(2,at=seq(0,0.18,0.03))
```



Il parametro *ylim* della funzione *plot()* serve ad impostare l’escursione sull’asse delle y, per noi da 0 a 0.18.

Il comando *axis()* serve a posizionare le graduazioni sugli assi. Il primo parametro (*side*) serve ad individuare l’asse (1 per orizzontale, 2 per verticale).
Il parametro *at* (vettore) serve a impostare la posizione delle graduazioni, nel nostro caso da 0 a 0.18 con passo 0.03.

Nella schermata seguente vedete i grafici delle distribuzioni di probabilità per le somme di 2, 3, 4, 5 dadi. Cosa osservate?



Esempio 4 Simulare per n volte il lancio di due dadi equi e calcolare la distribuzione delle frequenze relative delle somme. Verificare che al crescere di n (n=100, 1000, 10000, 100000) la distribuzione di frequenze relative tende alla distribuzione di probabilità della variabile S dell'esempio 3.

Ecco il codice (che, a questo punto, non dovrebbe aver bisogno di commenti):

somma_due_dadi_simulazione.R

Source on Save

```
#somma di due dadi (simulazione)
cat("\14")
nrepliche=100000

X1=sample(1:6,nrepliche,replace=T)
X2=sample(1:6,nrepliche,replace=T)
S=X1+X2
#sarebbe meglio sostituire la riga precedente
#con la riga:
#S=factor(X1+X2,levels=2:12)

distr_freq=table(S)/nrepliche
print(round(distr_freq,3))
barplot(distr_freq)
```

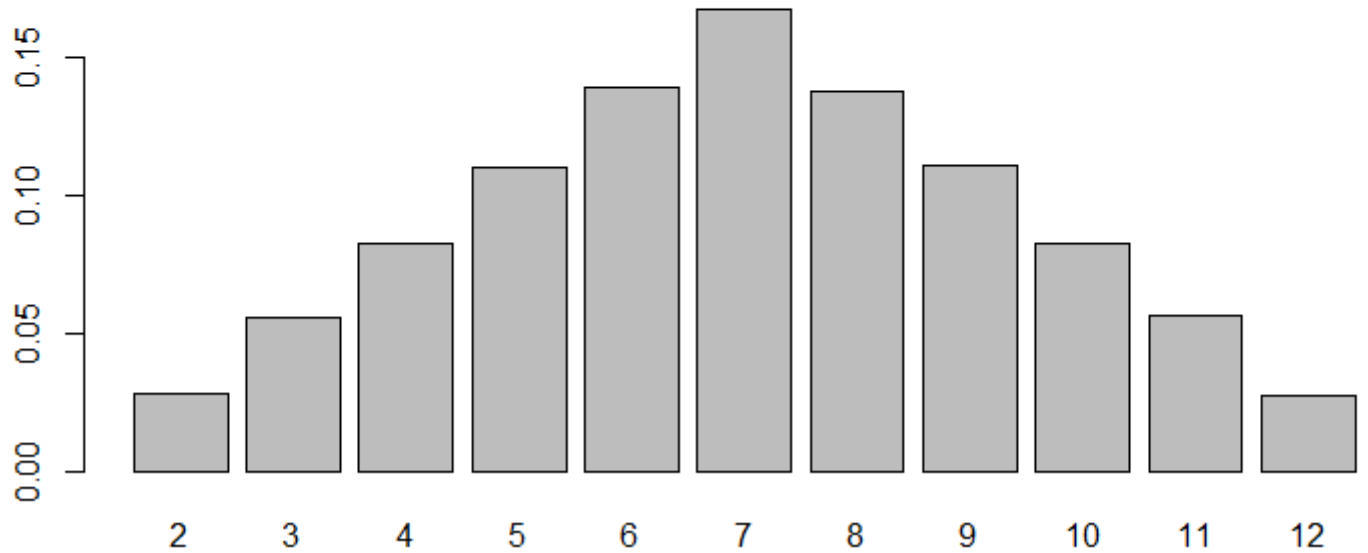
E questo è l'output:

Console D:/R/

```
S
     2      3      4      5      6      7      8      9     10     11     12
0.028  0.056  0.083  0.110  0.139  0.167  0.138  0.111  0.083  0.057  0.028
```

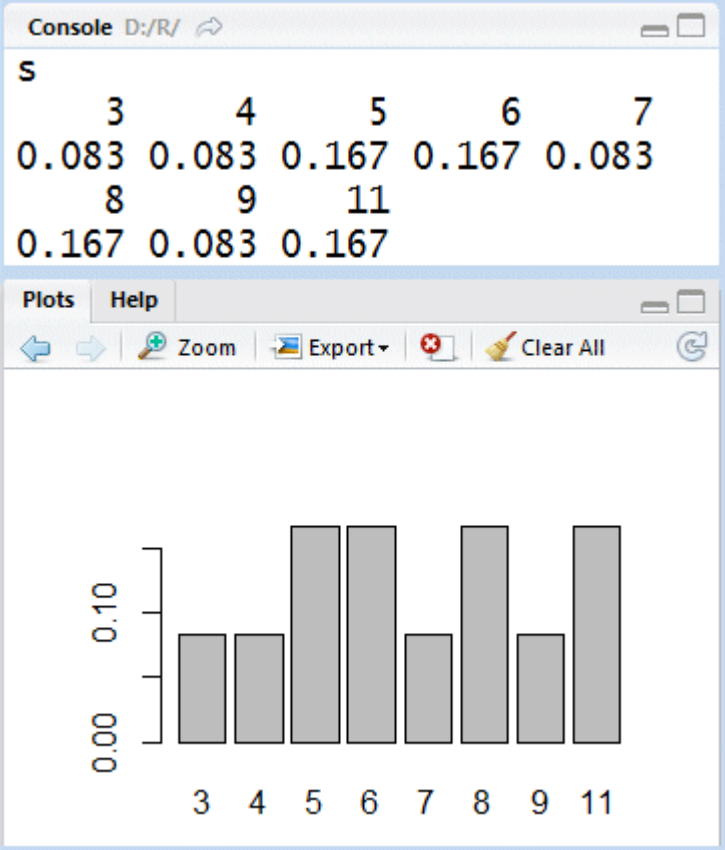
Plots Help

Zoom Export Clear All



Osservazioni? I risultati ottenuti, con n=100000, sono praticamente uguali a quelli calcolati nell'esempio 3; riflettete però sulla profonda differenza del modo con cui li abbiamo determinati: nel primo caso sulla base di un ragionamento teorico (calcolo delle probabilità), nel secondo sulla base di una simulazione cioè di un esperimento.

Osservazione. Il nostro programma ha un difetto, vediamo cosa succede se il numero delle repliche dell'esperimento è piccolo, ad esempio *nrepliche=20*:



Come si vede nella schermata a fianco, in questa simulazione non si sono presentati tutti i possibili 11 valori di somma (valori da 2 a 12), ad esempio non si è mai presentata una somma uguale 2; ciò non ci meraviglia data l'aleatorietà della simulazione e il piccolo numero di repliche dell'esperimento. Però avremmo preferito che fosse esplicitamente indicata, per queste somme che non compaiono, la frequenza 0. Come fare? Il linguaggio R fornisce una comoda soluzione: trasformiamo la variabile S, che per il momento è un semplice vettore di somme, in una variabile di tipo *factor* cioè una variabile che ci consente di indicare anche tutti i possibili *livelli*, cioè per noi tutti i possibili valori da 2 a 12, che una somma potrebbe assumere. Ecco come fare, sostituiamo la riga di programma

S = X1+X2

con la riga

S = factor(X1+X2, levels=1:12)

Ora il comando *table(S)* sarà in grado di indicare anche le somme con frequenza nulla (provare!).

Il prossimo problema introduce la nozione di *funzione di ripartizione* o *funzione di distribuzione cumulativa* di una variabile casuale S: F(s)=prob(S≤s).

Esempio 5 Qual è la probabilità che la somma S di tre dadi equi sia minore o uguale a 9? In generale: qual è la probabilità che la somma di tre dadi sia minore o uguale a n (3 ≤ n ≤ 18)?

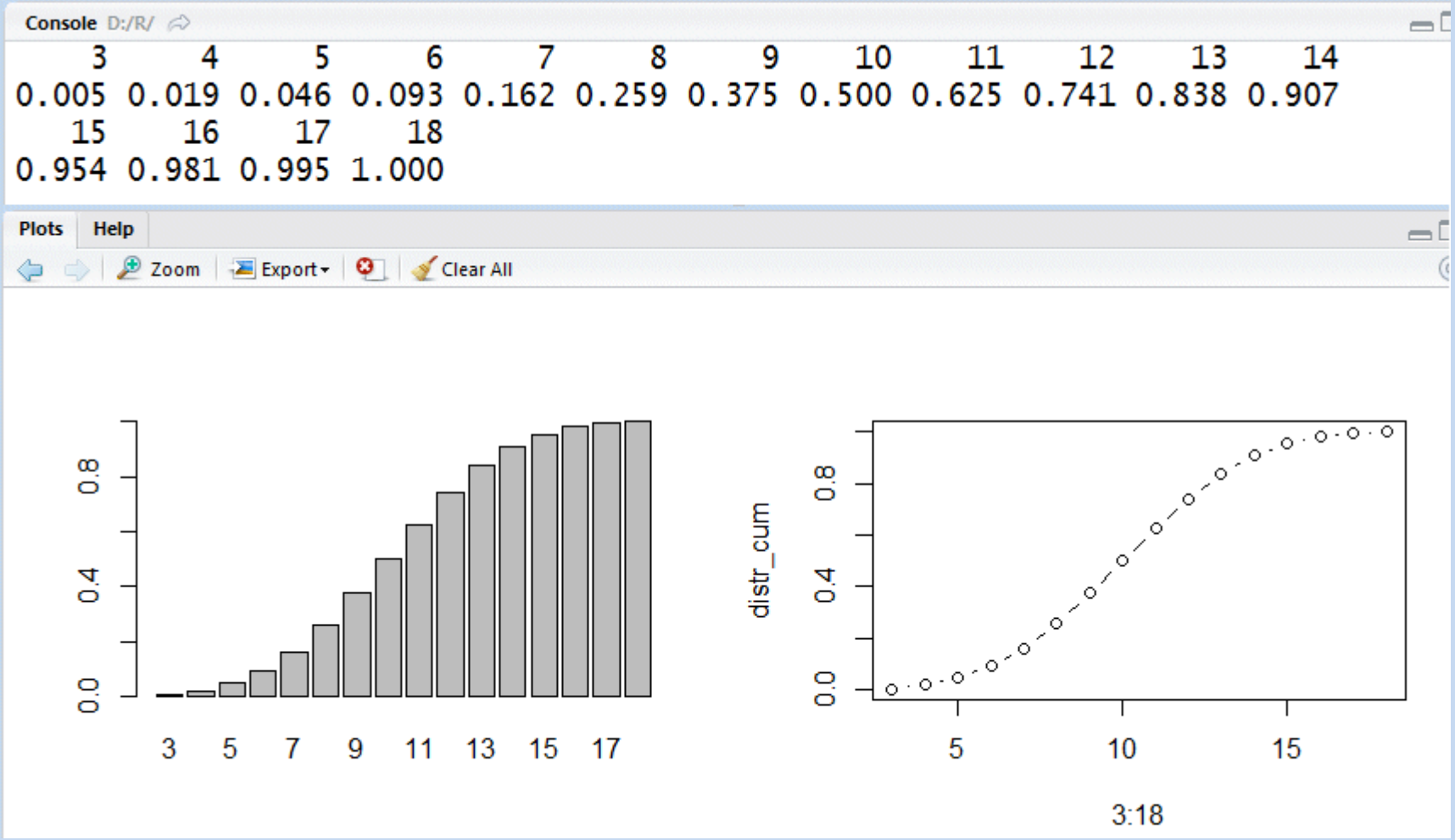
Ecco il codice:

distribuzione_cumulativa_somma_tre_dadi.r

Source on Save | Run

```
#distribuzione cumulativa somma di tre dadi
cat("\14")

dado1=1:6
dado2=1:6
dado3=1:6

df=expand.grid(dado1,dado2,dado3)
colnames(df)=c("dado1","dado2","dado3")

somme=df$dado1+df$dado2+df$dado3

distr_prob=table(somme)/length(somme)
distr_cum=cumsum(distr_prob)
print(round(distr_cum,3))

par(mfrow=c(1,2))
barplot(distr_cum)
plot(3:18,distr_cum,type="b")
```

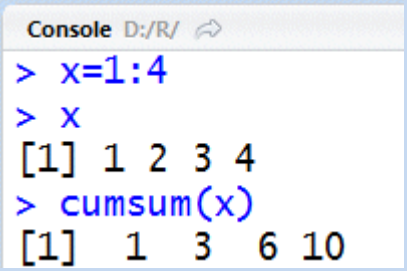
Output:



Come si vede dalla tabella, la probabilità che la somma dei dadi sia minore o uguale a 9 è 0.375 (provate a realizzare una simulazione per verificare questo risultato). Nel pro-

gramma ci sono solo un paio di cose da segnalare:

1. La funzione *cumsum(x)*, dove x è un vettore, ci fornisce il vettore delle somme cumulate degli elementi di x; ad esempio



Le somme cumulate di x:
x[1] = 1
x[1]+x[2] = 3
x[1]+x[2]+x[3] = 6
x[1]+x[2]+x[3]+x[4] = 10

Ora la somma cumulata è proprio quella che ci serve. Qual è, ad esempio, la probabilità che la somma S di tre dadi sia minore o uguale a 5? Dobbiamo sommare le tre probabilità

prob(S=3) + prob(S=4) + prob(S=5)

cioè

distr_prob[1] + distr_prob[2] + distr_prob[3] = 0,046

(qui le probabilità semplicemente si sommano perché gli eventi S=3, S=4, S=5 sono evidentemente incompatibili).

2. La funzione *par()* serve a impostare i parametri grafici. Nel nostro caso vogliamo mostrare due grafici affiancati nella finestra di output grafico; questo si fa settando il parametro *mfrow=c(1, 2)* in modo da avere i grafici su 1 riga e 2 colonne. Se volessimo quattro grafici: *mfrow=c(2, 2)* cioè grafici su 2 righe e 2 colonne (fare delle prove!).

I prossimi due problemi riguardano l'estrazione casuale di biglie da un'urna, estrazione che può essere con o senza reimmissione; le due diverse modalità introducono due importanti distribuzioni di probabilità che sono rispettivamente la distribuzione binomiale e la distribuzione ipergeometrica.

Esempio 6 Un'urna contiene 5 biglie bianche e 3 biglie nere. Si estraggono a caso, in sequenza, tre biglie, rimettendo ogni volta la biglia estratta nell'urna. Calcolare la distribuzione di probabilità della variabile casuale X="numero di biglie bianche estratte". Verificare il risultato con una simulazione.

I valori possibili per la variabile casuale X sono evidentemente 0, 1, 2, 3. Esaminiamo i vari casi, tenendo presente che qui le estrazioni sono indipendenti perché ogni volta viene ripristinato lo stato iniziale dell'urna.

X=0 L'unica sequenza possibile è

N	N	N

ed ha probabilità (3/8)·(3/8)·(3/8) = 0.375·0.375·0.375 ≅ 0.05273. Qui vale la regola di moltiplicazione: il 37.5% delle volte la prima estrazione è N (probabilisticamente), il 37.5% del 37.5% delle volte anche la seconda è N e il 37.5% del 37.5% del 37.5% tutte e tre le estrazioni sono N.

X=1 Le sequenze possibili sono

B	N	N

N	B	N

N	N	B

e hanno tutte probabilità (5/8)·(3/8)·(3/8); poiché le sequenze sono 3 e rappresentano eventi incompatibili (quindi le probabilità si sommano), la probabilità che sia X=1 è 3·(5/8)·(3/8)·(3/8) ≅ 0.26367.

X=2 Le sequenze possibili sono

B	B	N

B	N	B

N	B	B

e hanno tutte probabilità (5/8)·(5/8)·(3/8); poiché le sequenze sono 3 e rappresentano eventi incompatibili, la probabilità che sia X=2 è 3·(5/8)·(5/8)·(3/8) ≅ 0.43945.

X=3 L'unica sequenza possibile è

B	B	B

ed ha probabilità (5/8)·(5/8)·(5/8) ≅ 0.24414.

Facciamo la verifica (con R):

(3/8)·(3/8)·(3/8) + 3·(5/8)·(3/8)·(3/8) + 3·(5/8)·(5/8)·(3/8) + (5/8)·(5/8)·(5/8) = 1

(condizione di normalizzazione: la somma delle probabilità di una distribuzione deve sempre essere 1).

Ora la simulazione:

simulazione estrazione biglie con rimessa.R*

Source on Save | Run

```
#Simulazione della variabile casuale
# X="numero di biglie bianche estratte"
# estrazione con reinserimento

cat("\14")
nbianche=5          #numero biglie bianche nell'urna
nnere=3             #numero biglie nere nell'urna
nestratte=3         #numero biglie estratte ad ogni prova
nrepliche=100000    #numero prove replicate

urna=c(rep("B",nbianche),rep("N",nnere))

X=rep(0,nrepliche) #inizializzazione del vettore X
for (i in 1:nrepliche)
  {estratte=sample(urna,nestratte,replace=TRUE)
   X[i]=length(estratte[estratte=="B"])}
distr_frequenze=table(X)/nrepliche
print(round(distr_frequenze,5))
barplot(distr_frequenze)
```

Console D:/R/

```
X
      0       1       2       3
0.05257 0.26129 0.44100 0.24514
```

Osservazioni sul programma.

1. Per la prima volta abbiamo utilizzato un ciclo *for*: la sintassi è questa

for (i in vettore) {gruppo di comandi da ripetere}

Se il vettore fosse *1:n* i comandi verrebbero ripetuti *n* volte, con *i* che va da 1 a n. Attenti alle parentesi. La variabile *i*, naturalmente, può essere sostituita da qualsiasi variabile. Nel nostro caso il ciclo *for* serve a gestire la ripetizione dell'estrazione delle tre biglie.

2. Il vettore X serve a memorizzare, nel suo elemento X[i], il numero di biglie bianche estratte alla i-esima prova; all'inizio, prima del ciclo for, tutti gli elementi di X sono posti uguali a zero (si parla di *inizializzazione* della variabile).

3. Come facciamo a sapere quante sono le biglie bianche estratte? Il vettore *estratte* potrebbe essere, ad esempio, "N", "B", "B" o anche "B", "N", "B". Qui entra in gioco la potenza di R, il comando

estratte[estratte=="B"]

seleziona gli elementi del vettore *estratte* che sono uguali a "B"; in entrambi i due casi di esempio citati, tale comando fornirebbe in uscita il vettore "B", "B" la cui lunghezza è, appunto, il numero di biglie bianche cercato.

Esempio 7 Un'urna contiene 5 biglie bianche e 3 biglie nere. Si estraggono a caso tre biglie, le biglie sono estratte in un sol colpo oppure una dopo l'altra senza però rimettere la biglia estratta nell'urna. Calcolare la distribuzione di probabilità della variabile casuale X="numero di biglie bianche estratte". Verificare il risultato con una simulazione.

I valori possibili per la variabile casuale X sono 0, 1, 2, 3 (questa volta la cosa è meno evidente che nell'esempio precedente, perché?). Esaminiamo i vari casi, tenendo presente che qui le estrazioni sono dipendenti perché ad ogni estrazione si modifica la composizione dell'urna. Nel ragionamento che svilupperemo ci fa comodo pensare che le biglie siano estratte una dopo l'altra, per cui parleremo di prima, seconda, terza biglia; tuttavia non cambia nulla se le biglie sono estratte insieme: quando le ho in mano e stringo il pugno ne avrò una a sinistra (la prima), una al centro (la seconda), una a destra (la terza).

I casi possibili:

X=0 L'unica sequenza possibile è

N	N	N

ed ha probabilità $(3/8)\cdot(2/7)\cdot(1/6) \cong 0.01786$. Attenzione, qui entrano in gioco, per la seconda e terza estrazione, probabilità condizionate: se la prima biglia estratta è nera, la probabilità di estrarre

una seconda biglia nera è 2/7 perché nell'urna sono rimaste due biglie nere su un totale di 7 biglie. In modo analogo anche 1/6 è una probabilità condizionata: se le prime due estrazioni sono due biglie nere, nell'urna rimane una sola biglia nera su 6.

X=1 Le sequenze possibili sono

B	N	N

N	B	N

N	N	B

e hanno tutte, per la proprietà commutativa del prodotto, probabilità $(5/8)\cdot(3/7)\cdot(2/6) = (5\cdot3\cdot2)/(8\cdot7\cdot6)$; poiché le sequenze sono 3 e rappresentano eventi incompatibili (quindi le probabilità si sommano), la probabilità che sia X=1 è $3\cdot(5/8)\cdot(3/7)\cdot(2/6) \cong 0.26786$.

X=2 Le sequenze possibili sono

B	B	N

B	N	B

N	B	B

e hanno tutte probabilità $(5/8)\cdot(4/7)\cdot(3/6)$; poiché le sequenze sono 3 e rappresentano eventi incompatibili, la probabilità che sia X=2 è $3\cdot(5/8)\cdot(4/7)\cdot(3/6) \cong 0.53571$.

X=3 L'unica sequenza possibile è

B	B	B

ed ha probabilità $(5/8)\cdot(4/7)\cdot(3/6) \cong 0.17857$.

Facciamo la verifica (con R):

$$(3/8)\cdot(2/7)\cdot(1/6) + 3\cdot(5/8)\cdot(3/7)\cdot(2/6) + 3\cdot(5/8)\cdot(4/7)\cdot(3/6) + (5/8)\cdot(4/7)\cdot(3/6) = 1$$

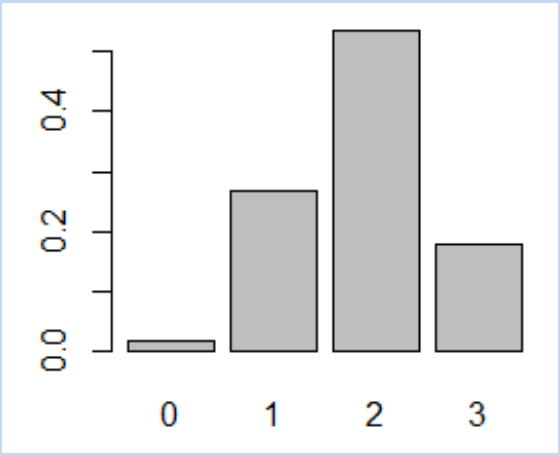
Ora la simulazione:

simulazione estrazione biglie senza rimessa.R

Source on Save | Run

```
#Simulazione della variabile casuale
# X="numero di biglie bianche estratte"
# estrazione senza reinserimento

cat("\14")
nbianche=5          #numero biglie bianche nell'urna
nnere=3             #numero biglie nere nell'urna
nestratte=3         #numero biglie estratte ad ogni prova
nrepliche=100000    #numero prove replicate

urna=c(rep("B",nbianche),rep("N",nnere))

X=rep(0,nrepliche) #inizializzazione del vettore X
for (i in 1:nrepliche)
  {estratte=sample(urna,nestratte,replace=FALSE)
   X[i]=length(estratte[estratte=="B"])}
distr_frequenze=table(X)/nrepliche
print(round(distr_frequenze,5))
barplot(distr_frequenze)
```

```
X
      0       1       2       3
0.01765 0.26737 0.53586 0.17912
```



Fantastico: rispetto al programma precedente è bastato cambiare unicamente

replace=TRUE

in

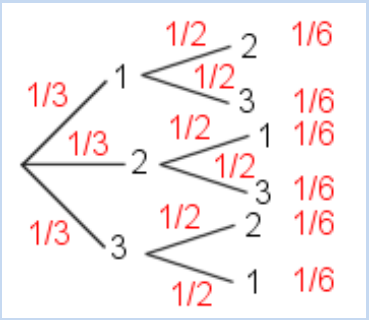
replace=FALSE

Osservate inoltre che cambiando i parametri potete simulare tutte le situazioni possibili (potenza dei parametri!), provate ad esempio così: *nbianche*=3, *nnere*=2, *nestratte*=3.
Quali sono, in questo caso, i valori possibili per la v.c. X?

Probabilità condizionata, dipendenza e indipendenza

L'esempio 7 di pag. 41 ha introdotto alcune nozioni chiave del calcolo delle probabilità: *probabilità condizionata*, *dipendenza di eventi*, *indipendenza di eventi*. Mettiamo a fuoco questi concetti ragionando su una situazione concreta: in un urna ci sono tre biglie numerate da 1 a 3. Si estraggono due biglie senza reimmissione. Consideriamo i due eventi:

A = "il primo numero estratto è 2"
B = "il secondo numero estratto è 3"

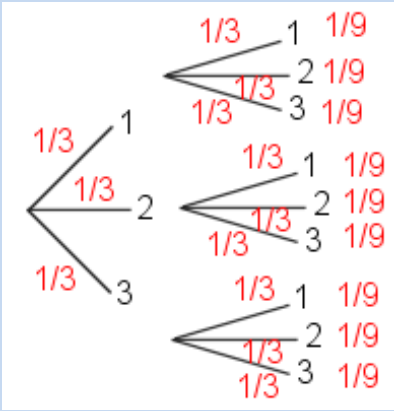


La probabilità dell'evento A è evidentemente p(A)=1/3. La probabilità dell'evento B, se non abbiamo alcuna informazione sulla prima estrazione, è di nuovo 1/3: p(B)=1/3. Siete convinti? Se non lo siete fate una simulazione oppure ragionate sul diagramma ad albero qui a fianco.

Supponiamo ora di sapere che si è verificato l'evento A: cosa possiamo dire dell'evento B? Il verificarsi di A condiziona il verificarsi di B (questa volta abbiamo un'informazione in più): scriveremo p(B|A)=1/2 cioè la probabilità di B dato A è 1/2 e parleremo di **probabilità condizionata**. I due eventi A e B sono **dipendenti** perché p(B) ≠ p(B|A).
Osservate inoltre che la probabilità che si verifichino congiuntamente gli eventi A e B è

p(A e B) = p(A)·p(B|A) = 1/6

(ragionate sul primo diagramma ad albero).



Consideriamo ora la stessa urna di prima ma dopo la prima estrazione rimettiamo la biglia estratta nell'urna (vedi secondo diagramma ad albero). Questa volta si ha

P(A) = 1/3, p(B) = 1/3, p(B|A) = p(B) = 1/3
p(A e B) = p(A)·p(B) = 1/9

e gli eventi A e B sono **indipendenti** perché p(B) = p(B|A)[1] cioè il verificarsi o meno di A non ha alcuna influenza sulla probabilità di B.

[1] Questa è una definizione: A e B sono **indipendenti** se p(B)=p(B|A) o, equivalentemente, se p(A e B)=p(A)·p(B).

La distribuzione binomiale

L'esempio 6 può essere generalizzato nel modo seguente: in un'urna ho n biglie di cui b sono bianche e $n-b$ sono nere ed estraggo, con reimmissione, k biglie. Qual è la distribuzione di probabilità della variabile casuale

X = "numero di biglie bianche estratte"?

Una situazione come questa è rappresentativa di un tipo di distribuzione di probabilità che prende il nome di ***distribuzione binomiale*** di parametri

k e $p = b/n$

(b/n è la probabilità di estrarre una biglia bianca).

Per indicare che la v. c. X ha distribuzione binomiale di parametri k e p scriveremo

$X \sim B(k, p)$

Vediamo, in astratto, qual è la caratterizzazione di una distribuzione binomiale di parametri *k* e *p* cioè di una distribuzione *B(k, p)*:

1. Si considera un esperimento aleatorio che abbia solo due esiti possibili che per comodità chiamiamo SUCCESSO e INSUCCESSO.

Nel caso dell'urna, SUCCESSO significa estrarre una biglia bianca, INSUCCESSO estrarre una biglia nera.

2. L'esperimento viene ripetuto *k* volte e le prove successive sono indipendenti.

Nel caso dell'urna, la biglia estratta viene rimessa nell'urna e ciò garantisce l'indipendenza delle estrazioni.

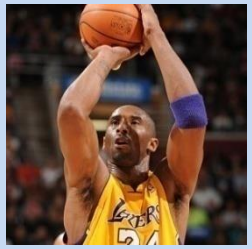
3. La probabilità di SUCCESSO rimane costantemente uguale a *p*.

Nel caso dell'urna la probabilità di SUCCESSO è b/n, dove n è il numero delle biglie, e rimane costante (dato il reinserimento).

4. La distribuzione binomiale B(k, p) è la distribuzione di probabilità della variabile casuale X="numero di SUCCESSI su k prove" dove i valori di X sono 0, 1, 2, …, k

Per la distribuzione binomiale c'è una formula interessante (che usa i coefficienti binomiali, li conoscete?) ma per il momento non ce ne occupiamo. Piuttosto provate ad indicare quale variabile casuale a distribuzione binomiale modellizza i seguenti fenomeni aleatori. Una variabile casuale rappresenta la risposta ad una domanda che non ammette una risposta deterministica. In tutti gli esempi che seguono la risposta è una variabile casuale X.

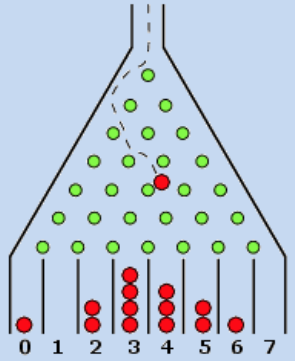
a) Si lancia 10 volte una moneta equa, quante volte si presenta TESTA? [X ~ B(10, 1/2)]



b) Si lancia 7 volte un dado equo, quante volte si presenta 3? [X ~ B(7, 1/6)]

c) Un giocatore di basket ha probabilità 0.78 di fare canestro in tiro libero. Su 6 tiri liberi, quante volte farà canestro? [X ~ B(6, 0.78)]

d) Un produttore di sementi fornisce semi per fiori con probabilità 0,82 di germogliare. Se vengono piantati 500 semi, quanti germoglieranno? [X ~ B(500, 0.82)]



e) Nella macchina di Galton in figura, la biglia ha la stessa probabilità di deviare a sinistra o a destra per ogni perno urtato. In quale slot, da 0 a 7, finisce la biglia? [X ~ B(7, 1/2)]

f) Una fabbrica di lampadine ha accertato per via statistica che una lampadina su 120 è difettosa. Quante lampadine difettose contiene una confezione di 60? [X ~ B(60, 1/120)]

Distribuzioni con R

R dispone di un set di comandi per gestire tutte le principali distribuzioni di probabilità, in particolare per la distribuzione binomiale ci sono i comandi

dbinom(x, size, prob)

che ci fornisce i valori della distribuzione (*size* è il nostro k, *prob* è il nostro p e x è un numero o un vettore, vedi esempio seguente) e il comando

rbinom(n, size, prob)

che ci consente di simulare n realizzazioni di una v.c. con distribuzione *B(size, prob)*.

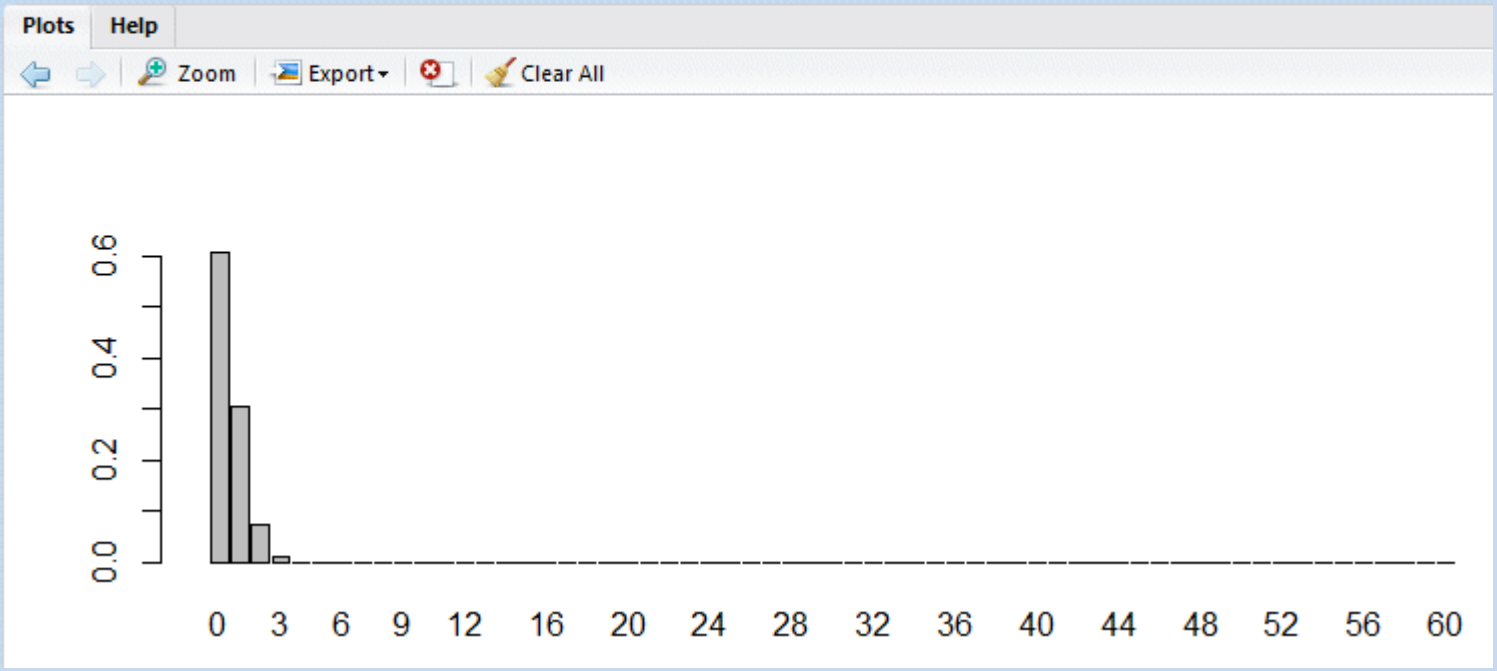
Esempio 1 La distribuzione B(3, 5/8) dell'esempio 6, 5 biglie bianche, 3 biglie nere, 3 estrazioni con reimmissione:

```
Console D:/R/
> dbinom(0:3,3,5/8) # la distribuzione di prob. di X~B(3,5/8)
[1] 0.05273438 0.26367188 0.43945312 0.24414062
> dbinom(2,3,5/8) # probabilità che sia X=2
[1] 0.4394531
```

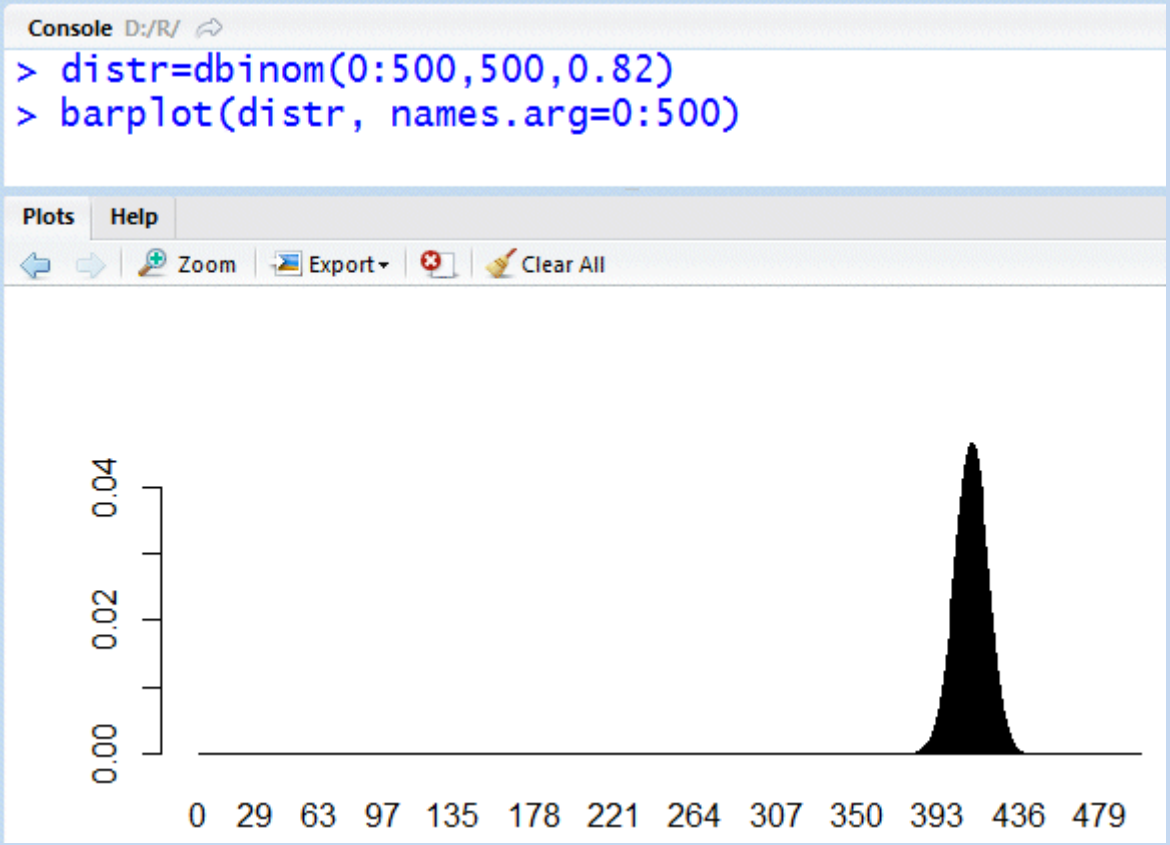
Esempio 2 La distribuzione B(60, 1/120) dell'esempio (f) di pag. 47:

Console D:/R/

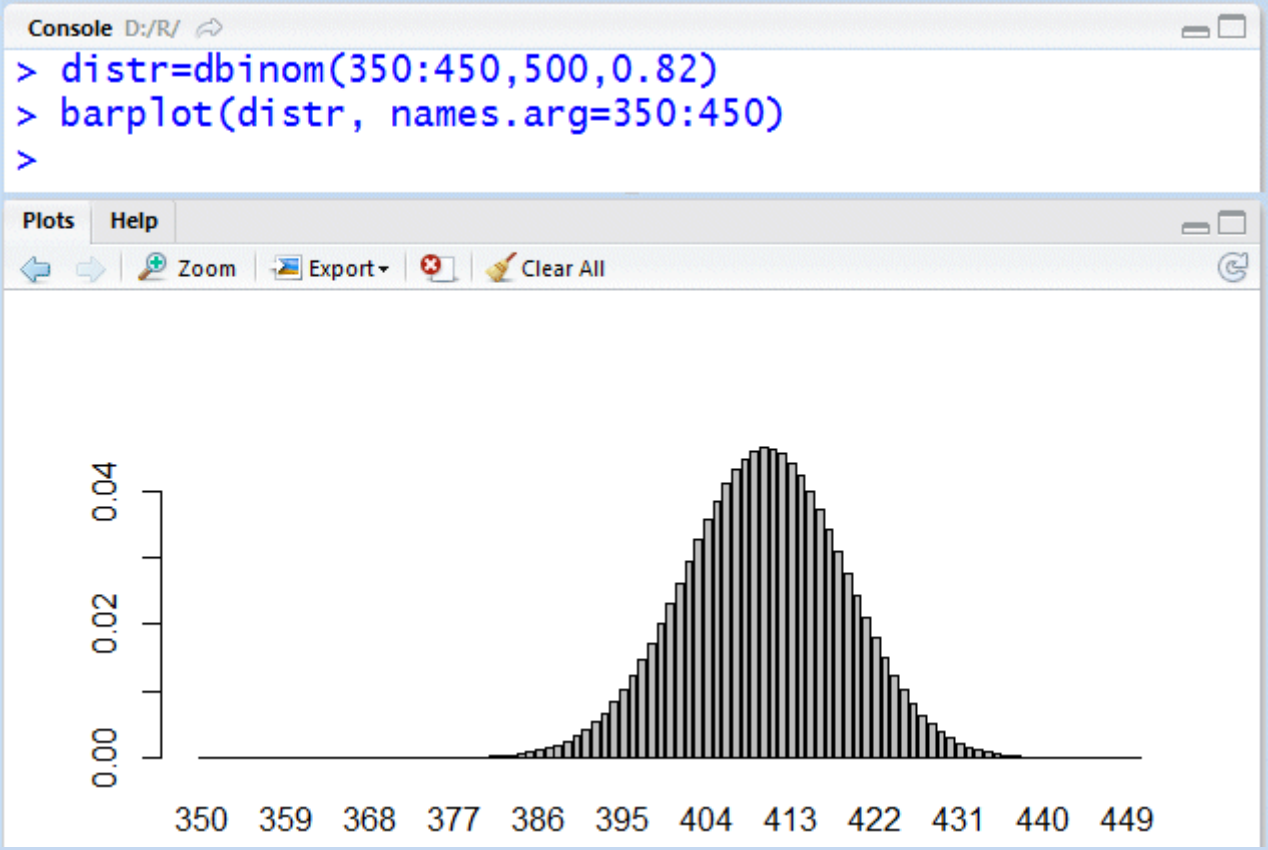
```
> distr=dbinom(0:60,60,1/120)
> barplot(distr, names.arg=0:60)
```



Esempio 3 La distribuzione B(500, 0.82) dell'esempio (d) di pag. 47:



Zoomiamo sulla distribuzione:



Tenete infine presente che per la **distribuzione ipergeometrica** (estrazione senza reinserimento) utilizzeremo il comando di R

dhyper(x, m, n, k)

dove x è un numero o un vettore, m è il numero di biglie bianche nell'urna, n il numero di biglie nere e k il numero di biglie estratte. Ad esempio il comando *dhyper(0:3, 5, 3, 3)* fornisce la distribuzione dell'esempio 7 di pag. 41 cioè la distribuzione della v.c. X="numero di biglie bianche estratte".

Valor medio di una variabile casuale

Il prossimo problema introduce un'idea fondamentale del calcolo delle probabilità: il *valor medio* o *valore atteso* di una variabile casuale.

Esempio 1 Paolo ha deciso di regalare a Francesco una certa somma aleatoria, procedendo in questo modo: Francesco lancia 10 volte un dado equo e ogni volta che esce 6 riceve 1 euro. Quale somma riceverà, in media, Francesco?

Premessa Come faccio a calcolare la media empirica di una serie di valori dati? Se ad esempio i voti negli scritti di matematica di uno studente, nell'arco dell'anno, sono 4, 3, 5, 5, 6, 6, 5, 6, 6, qual è la media? Sapete benissimo come fare: si sommano i voti e si divide per il loro numero, quindi la media è (4+3+5+5+6+6+5+6+6)/9 = 5.11. C'è però un altro modo per calcolarla, ai nostri fini preferibile: si moltiplica ciascun voto per la sua frequenza relativa, quindi

$$\text{media} = 4\cdot(1/9) + 3\cdot(1/9) + 5\cdot(3/9) + 6\cdot(4/9)$$

Il risultato ovviamente è lo stesso (aritmetica elementare). Notare che nel calcolo della media di dati empirici non c'è nulla di aleatorio.

Torniamo al nostro problema e, per il momento, risolviamolo con una simulazione, quindi sperimentalmente. Ecco il codice e l'output (qui l'esperimento è ripetuto solo 5 volte, per capir bene quello che succede).

simulazione valor medio (1) .R* ×

Source

```
cat("\14")
n=5
X=rep(0,n) #inizializzazione

for (i in 1:n)
  {dati=sample(1:6,10,replace=TRUE)
  X[i]=length(dati[dati==6])
  cat("Dati:",dati)
  cat("  Freq. del 6: ",X[i])
  cat("\n") #serve per andare a capo
  }

cat("Media=",mean(X))
```

Nel calcolo della media potremmo evitare di salvare nel vettore X la frequenza del 6 relativa a ciascuno degli n esperimenti. Perché?

Console D:/R/

```
Dati: 4 1 6 3 6 1 3 1 1 4  Freq. del 6:  2
Dati: 5 4 2 1 6 4 4 1 4 4  Freq. del 6:  1
Dati: 2 4 6 3 4 3 5 3 1 2  Freq. del 6:  1
Dati: 3 3 5 2 6 6 1 5 2 1  Freq. del 6:  2
Dati: 3 4 2 3 6 2 3 4 4 2  Freq. del 6:  1
Media= 1.4
```

Se eseguiamo molte volte l'esperimento, diciamo 1000 volte, otteniamo una buona approssimazione della media che è circa 1,7 euro.

Cerchiamo ora di capire come si può procedere teoricamente (via calcolo delle probabilità). Prima di tutto introduciamo la v.c. che modellizza la nostra situazione:

X = "somma ricevuta" =
= "numero di volte che si presenta sei su 10 lanci del dado"

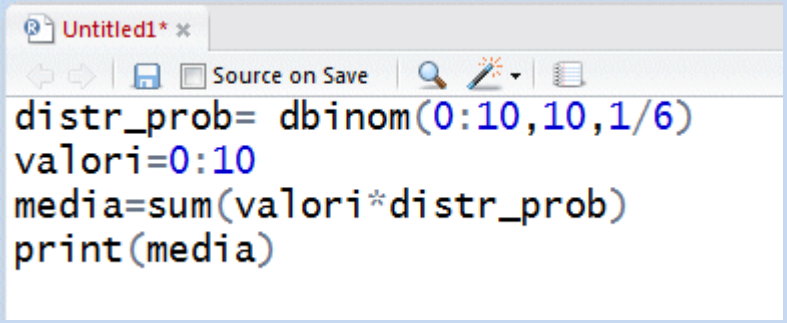
I valori possibili per X sono i numeri interi da 0 a 10 e riconosciamo per X la distribuzione binomiale B(10, 1/6), quindi siamo in grado di calcolare, con R, la probabilità

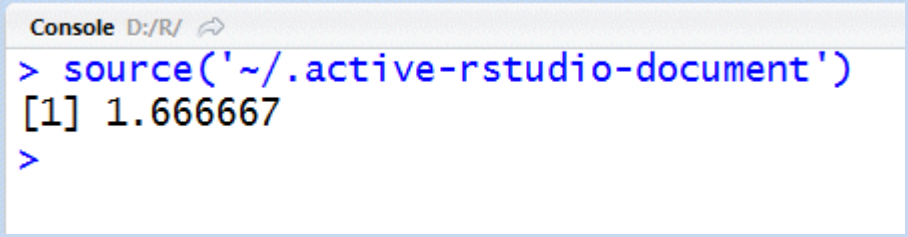
p(X=i)

cioè la probabilità dell'evento X=i, dove *i* rappresenta un numero da 0 a 10. Per calcolare la media di dati empirici moltiplicavamo ciascuno dei valori per la sua frequenza relativa (ricordate?); ora è naturale moltiplicare ciascuno dei valori possibili della v.c. X per la sua probabilità. Allora il *valor medio* di X, che si indica con E(X), è

$$E(X) = 0\cdot p(X=0) + 1\cdot p(X=1) + 2\cdot p(X=2) + \ldots + 10\cdot p(X=10)$$

Bene, facciamo il calcolo con R.





Come vedete il valor medio calcolato sperimentalmente con la simulazione di pag. 53 è una buona approssimazione del valor medio calcolato teoricamente (e l'approssimazione tende a migliorare se aumenta il numero di repliche dell'esperimento). L'esempio che abbiamo esaminato dà senso alla seguente definizione:

Se X è una variabile casuale discreta che assume i valori x_1, x_2, … x_n con probabilità p_1, p_2, …, p_n, si chiama ***valor medio*** o ***valore atteso*** di X il valore

$$E(X) = x_1 \cdot p_1 + x_2 \cdot p_2 + \ldots + x_n \cdot p_n$$

(la E in E(X) è l'iniziale di *expected value*, dall'inglese).

Varianza di una variabile casuale

Il prossimo problema introduce un'altra idea fondamentale del calcolo delle probabilità: la *varianza* di una variabile casuale.

Esempio 1 Riferendoci all'esempio precedente simuliamo per 10000 volte la somma x ricevuta da Francesco e calcoliamo la media *m* di tali somme ricevute. Ci chiediamo: quando dista in media la somma x effettivamente ricevuta da *m*?

La prima cosa che viene in mente per calcolare la distanza della somma x effettivamente ricevuta dalla somma media m è la formula

$$|x-m|$$

Tuttavia il valore assoluto introduce delle complicazioni tecniche che vogliamo evitare e allora consideriamo come "distanza" il quadrato di tale valore cioè

$$(x-m)^2$$

Quindi non ci resta che calcolare la media di tali valori $(x-m)^2$. Se ad esempio le somme in euro effettivamente ricevute da Francesco in 5 occasioni fossero:

3, 1, 3, 2, 5

il valor medio delle "distanze" dalla media, cioè quella che chiameremo *varianza empirica* di questi dati, si calcola così:

media = (3+1+3+2+5)/5 = 2,8

varianza = $[(3-2,8)^2+(1-2,8)^2+(3-2,8)^2+(2-2,8)^2+(5-2,8)^2]/5 = 1,76$

Ecco la simulazione:

simulazione varianza (1) .R*

Source on Save

```
cat("\14")
n=10000
x=rep(0,n) #inizializzazione

for (i in 1:n)
  {dati=sample(1:6,10,replace=TRUE)
  x[i]=length(dati[dati==6])
  }
media=mean(x)
varianza=sum((x-media)^2)/n
cat("Media=",media,"\n")
cat("Varianza=",varianza)
```

Console D:/Testi/Circolo 2014-15/

```
Media= 1.66
Varianza= 1.3844
>
```

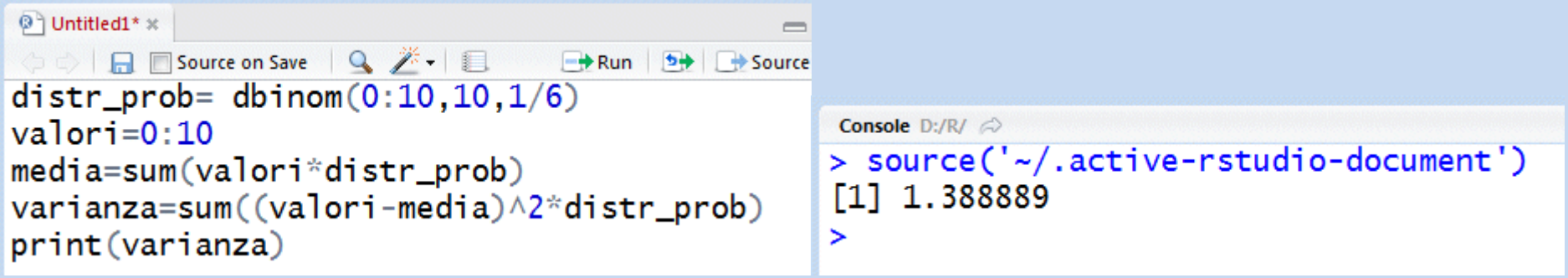
La varianza, calcolata sperimentalmente, è quindi circa 1,4. Cerchiamo ora di capire come si può procedere teoricamente (via calcolo delle probabilità). Appare sensato definire la ***varianza*** della variabile casuale

X = "somma ricevuta" =
= "numero di volte che si presenta sei su 10 lanci del dado"

come valor medio della nuova variabile casuale $(X-E(X))^2$ cioè

$$var(X) = E[(X-E(X))^2]$$

Siete convinti che $D = (X-E(X))^2$ è una nuova variabile casuale? Infatti E(X), il valor medio di X, è un valore costante non aleatorio ma $D=(X-E(X))^2$ è una variabile aleatoria perché lo è X (D è una funzione della v.c. X). E, se ci pensate, D è proprio la variabile aleatoria che risponde alla domanda da cui siamo partiti: qual è la "distanza" di X dal suo valor medio? Il valore atteso di D è precisamente quello che cerchiamo: la distanza media. Calcoliamo con R la varianza e confrontiamola con quella valutata sperimentalmente:



Deviazione standard di una variabile casuale

La varianza di una variabile casuale X

$$var(X) = E[(X - E(X))^2]$$

ci dà un'idea di quale sia la "distanza" media della variabile dal suo valor medio (quindi della dispersione dei valori di X attorno alla media); la varianza ha delle importanti proprietà che esamineremo più avanti ma, come ricorderete, nella sua definizione compare in realtà il quadrato di una distanza. Ciò ovviamente altera la vera natura della distanza media, pensate ad esempio che se i valori della v.c. X fossero misure in metri allora l'unità di misura della varianza sarebbe il metro quadro. Non a caso per la varianza abbiamo sempre parlato di "distanza" tra virgolette. Per mettere a posto le cose si introduce allora una nuova grandezza, la ***deviazione standard***, data dalla radice quadrata della varianza e indicata con σ(X)

$$\sigma(X) = \sqrt{var(X)}$$

La deviazione standard può essere interpretata come la distanza media di una variabile casuale X dal suo valor medio E(X); inoltre l'unità di misura della deviazione standard è la stessa dei valori che X può assumere.

Nota La deviazione standard assomiglia molto alla distanza euclidea in $\mathbb{R}^n$.

Supponiamo che la v.c. discreta X assuma i valori

$x_1, x_2, \ldots, x_n$

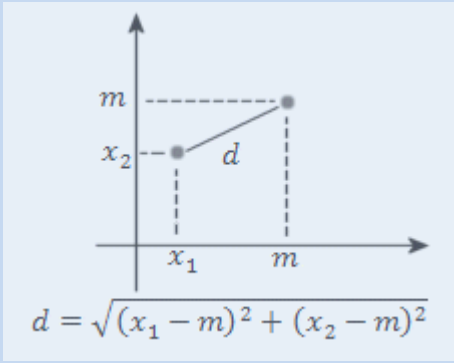
con probabilità $p_1, p_2, \ldots, p_n$ e sia m=E(X); allora

$$\sigma(X)=\sqrt{\sum_{i=1}^{n}(x_i-m)^2 p_i}$$

Quindi possiamo pensare a σ(X) come alla distanza euclidea tra i punti

$(x_1, x_2, \ldots, x_n)$ e $(m, m, \ldots, m)$

di $\mathbb{R}^n$ dove però ciascun addendo è "pesato" secondo la probabilità p_i.



Se la distribuzione di probabilità di X fosse uniforme avremmo

$$\sigma(X)=\sqrt{\frac{1}{n}}\sqrt{\sum_{i=1}^{n}(x_i-m)^2}$$

e questo valore è proprio la distanza euclidea tra i punti $(x_1, x_2, \ldots, x_n)$ e $(m, m, \ldots, m)$ di $\mathbb{R}^n$ a meno di un fattore $\frac{1}{\sqrt{n}}$.

Valore atteso, varianza, dev. standard: esperimenti

Esempio 1 Consideriamo la v.c. X="valore che si presenta lanciando un dado equo". Calcolare: il valore atteso, la varianza, la deviazione standard di X. Calcolare inoltre la probabilità dell'evento

$$E(X)-\sigma(X) \le X \le E(X)-\sigma(X)$$

cioè la probabilità che il valore di X sia compreso tra il valor medio di X meno la dev. st. di X e il valor medio di X più la dev. st. di X.

deviazione standard (1).R

Source on Save | Run | Source

```
# X="valore che si presenta
# lanciando un dado equo"
cat("\14")
x=1:6
distr_prob=rep(1/6,6)
media=sum(x*distr_prob)
varianza=sum((x-media)^2*distr_prob)
devst=sqrt(varianza)
sub_x=x[media-devst<=x & x<=media+devst]
p=length(sub_x)/6
cat("media=", media, "\n")
cat("varianza=", varianza, "\n")
cat("dev. st.=", devst, "\n")
cat("p(",media-devst,"<= X <=",media+devst,")=",p,"\n")
```

Console D:/temp/

```
media= 3.5
varianza= 2.916667
dev. st.= 1.707825
p( 1.792175 <= X <= 5.207825 )= 0.6666667
>
```

Esempio 2 Simulare n valori (realizzazioni) della v.c. X dell'esempio precedente e calcolare la media m, la varianza v, la deviazione standard σ dei dati ottenuti. Calcolare inoltre la frequenza relativa dei valori compresi tra $m-\sigma$ e $m+\sigma$ e rappresentare graficamente i dati che cadono in tale intervallo.

simulazione deviazione standard (1).R

Source on Save

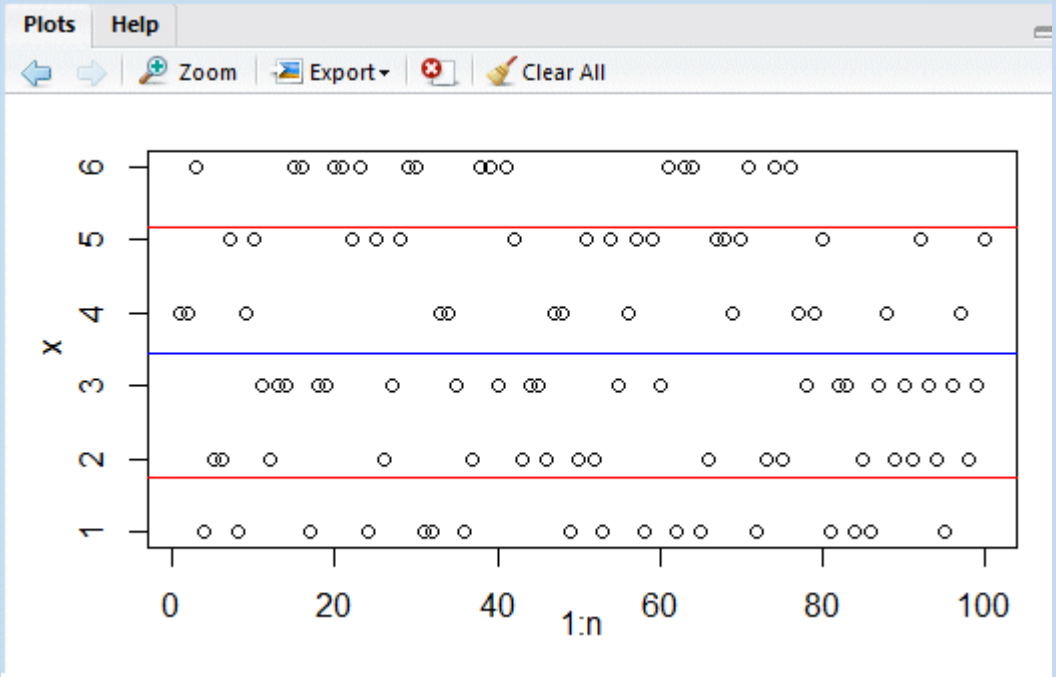
```
cat("\14")
n=100
x=sample(1:6,n,replace=TRUE)
media=mean(x)
varianza=sum((x-media)^2)/n
devst=sqrt(varianza)
sub_x=x[media-devst<=x & x<=media+devst]
freq_rel=length(sub_x)/n
cat("media=", media, "\n")
cat("varianza=", varianza, "\n")
cat("dev. st.=", devst, "\n")
cat("freq. rel.=",freq_rel,"\n")

plot(1:n,x)
abline(h=media, col="blue")
abline(h=media-devst, col="red")
abline(h=media+devst, col="red")
```

Console D:/R/

```
media= 3.45
varianza= 2.9475
dev. st.= 1.716828
freq. rel.= 0.66
>
```

abline(h=k) **traccia la retta oriz. y=k;** ***abline*****(v=k) traccia la retta vert. x=k;** ***abline(a=q, b=m)*** **traccia la retta di eq. y=mx+q.**

Le rette sono tracciate solo se aggiunte ad un grafico esistente.



Esempio 3 Si lancia 10 volte una moneta equa e si considera la v.c. X=”numero di volte che si presenta TESTA”. Calcolare: il valore atteso, la varianza, la deviazione standard di X. Calcolare inoltre la probabilità dell’evento

$$E(X)-\sigma(X) \le X \le E(X)-\sigma(X)$$

cioè la probabilità che il valore di X sia compreso tra il valor medio di X meno la dev. st. di X e il valor medio di X più la dev. st. di X. Rappresentare con un diagramma a barre la distribuzione di probabilità di X.

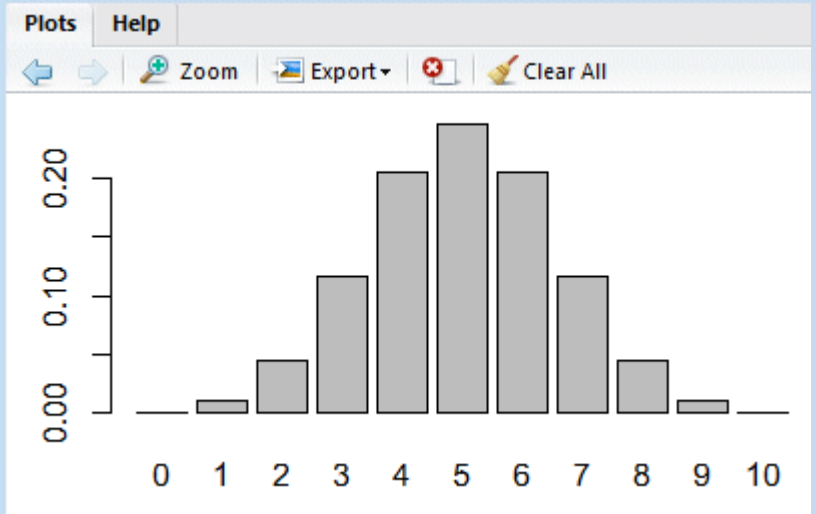
deviazione standard (2).R*

Source on Save | Run | Source

```
# X="numero di TESTA lanciando
# 10 volte una moneta equa"
cat("\14")
x=0:10
distr_prob=dbinom(0:10,10,1/2)
media=sum(x*distr_prob)
varianza=sum((x-media)^2*distr_prob)
devst=sqrt(varianza)
sub_x=x[media-devst<=x & x<=media+devst]
p=sum(distr_prob[sub_x+1]) #perché quel +1?
cat("media=", media, "\n")
cat("varianza=", varianza, "\n")
cat("dev. st.=", devst, "\n")
cat("p(",media-devst,"<= X <=",media+devst,")=",p,"\n")

barplot(distr_prob,names.arg=0:10)
```

Console D:/Testi/Circolo 2014-15/

```
media= 5
varianza= 2.5
dev. st.= 1.581139
p( 3.418861 <= X <= 6.581139 )= 0.65625
```



Esempio 4 Simulare n valori (realizzazioni) della v.c. X dell'esempio precedente e calcolare la media m, la varianza v, la deviazione standard σ dei dati ottenuti. Calcolare inoltre la frequenza relativa dei valori compresi tra $m-\sigma$ e $m+\sigma$ e rappresentare graficamente i dati che cadono in tale intervallo.

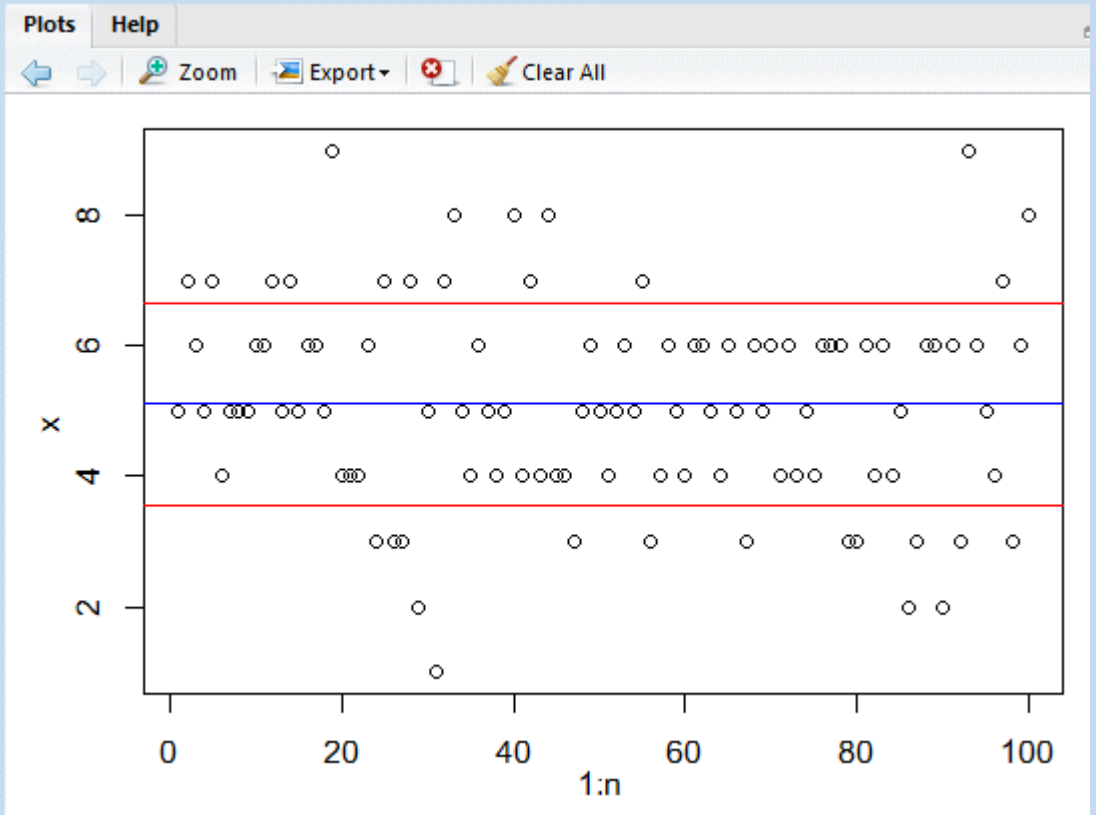
simulazione deviazione standard (2).R*

Source on Save

```
cat("\14")
n=100
x=rep(0,n)
for (i in 1:n){
  lanci=sample(0:1,10,replace=TRUE)
  x[i]=sum(lanci)}

media=mean(x)
varianza=sum((x-media)^2)/n
devst=sqrt(varianza)
sub_x=x[media-devst<=x & x<=media+devst]
freq_rel=length(sub_x)/n
cat("media=", media, "\n")
cat("varianza=", varianza, "\n")
cat("dev. st.=", devst, "\n")
cat("freq. rel.=",freq_rel,"\n")

plot(1:n,x)
abline(h=media, col="blue")
abline(h=media-devst, col="red")
abline(h=media+devst, col="red")
```

Console D:/R/

```
media= 5.11
varianza= 2.3979
dev. st.= 1.548515
freq. rel.= 0.69
>
```



Osservazioni

L'esempio 1 e l'esempio 3 mettono a confronto due diverse distribuzioni di probabilità, quella uniforme dell'esempio 1 e quella binomiale B(10, 1/2) dell'esempio 3. Nel primo caso circa 67 valori su 100 della variabile casuale X cadono, probabilisticamente, nell'intervallo [m-σ, m+σ] dove m è il valore atteso e σ la dev. standard di X; nel secondo caso la percentuale di valori che cadono nell'intervallo [m-σ, m+σ], dove m e σ si riferiscono questa volta alla seconda variabile casuale, è circa del 66%. In entrambi i casi dunque la "maggioranza" dei valori cade nel relativo intervallo [m-σ, m+σ] e ciò mette in luce il significato della dev. standard. Tuttavia le due distribuzioni sono molto diverse: nel secondo caso (esempio 3) i valori casuali si raccolgono molto più attorno alla media di quanto avvenga nel primo caso (esempio 1). Come possiamo renderci conto di questo fatto? Dobbiamo valutare la deviazione standard percentualmente rispetto alla media cioè il rapporto $\frac{\sigma(X)}{|E(X)|}$ che prende il nome di ***deviazione standard relativa***:

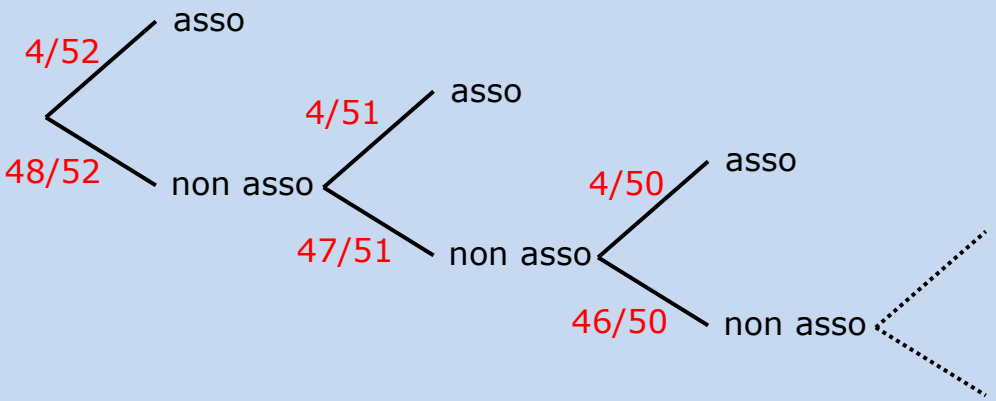
$$Nel\ primo\ caso\ \frac{\sigma(X)}{|E(X)|} = \frac{1{,}7078}{3{,}5} = 0{,}488 = 48{,}8\%$$

$$Nel\ secondo\ caso\ \frac{\sigma(X)}{|E(X)|} = \frac{1{,}5811}{5} = 0{,}316 = 31{,}6\%$$

Esempio 5 Quante carte devo alzare in media da un mazzo ben mescolato di 52 carte per ottenere un asso (il primo asso)?

Soluzione teorica

Quello che ci serve è il valore atteso della variabile casuale X che indica il numero di carte da alzare per ottenere un asso. E' chiaro che nel caso peggiore dovrò alzare 49 carte (perché può succedere che alzi 48 carte diverse da un asso ma, in questo caso, la 49-esima è necessariamente un asso). Quindi la variabile casuale X può assumere i valori da 1 a 49 e tali valori hanno naturalmente probabilità diverse. Indichiamo con p(n) la probabilità p(X=n) cioè la probabilità che il primo asso si presenti avendo alzato *n* carte. Ad esempio, p(2) indica la probabilità che il primo asso si presenti alla seconda carta. Come calcolare p(n)? Osserva il diagramma ad albero qui a fianco, sui cui rami sono indicate le probabilità condizionate. Allora si ha:



$p(1) = 4/52$
$p(2) = (48/52)(4/51)$
$p(3) = (48/52)(47/51)(4/50)$
$p(4) = (48/52)(47/51)(46/50)(4/49)$
…

Quindi il valor medio di X è

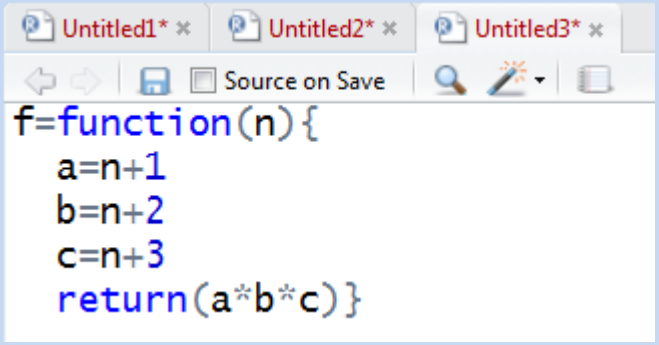
$$E(X) = 1 \cdot p(1)+2 \cdot p(2)+3 \cdot p(3)+\ldots+49 \cdot p(49)$$

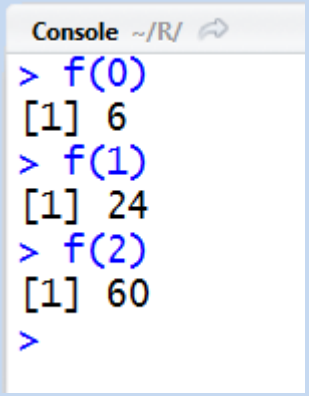
Abbiamo finito? Non proprio! Il calcolo di E(X), un calcolo elementare in cui entrano in gioco solo prodotti e somme di frazioni, è troppo lungo per essere fatto a mano anche con l'aiuto di una calcolatrice non programmabile; dobbiamo scrivere un programma, vediamo come procedere con R.

Poiché ci farà comodo definire la funzione p(n) che ci fornisce la probabilità che sia X=n, vediamo prima di tutto come definire una ***funzione*** con R. La struttura per definire una funzione che ad esempio chiamiamo f e che dipende dalla variabile n è la seguente:

f = function(n) {
comando
comando
…
return(valore in uscita)}

Ecco un semplice esempio:





Come vedete la funzione f ha in entrata il valore n e in uscita il valore (n+1)(n+2)(n+3); il valore in uscita è quello fornito dal comando *return*. Nel nostro caso potevamo mettere direttamente in uscita (n+1)*(n+2)*(n+3) ma abbiamo preferito eseguire tre assegnazioni intermedie, a=n+1, b=n+2, c=n+3, e poi mettere in uscita a*b*c. Da notare che tutte le variabili che intervengono nella funzione, compresa la variabile n, sono ***variabili locali*** cioè il loro valore esiste solo all'interno della funzione. Provate ad esempio, nella console, dopo aver utilizzato la funzione f, a chiedere il valore di n, a, b, c: otterrete in ogni caso il messaggio "object not found", cioè a tali variabili non risulta assegnato alcun valore (il valore è assegnato solo localmente, all'interno del blocco che definisce la funzione). Bene, ora possiamo procedere al calcolo del valore atteso E(X). Ecco il programma:

Untitled1* Untitled2* Untitled3*

Source on Save Run

```
cat("\14")

p=function(n){
    i=0:(n-2)
    a=4/52
    b=prod((48-i)/(52-i))*4/(52-n+1)
    if (n==1) return(a) else return(b)}

EX=0
for (i in 1:49)
     EX=EX+i*p(i)

cat("E(X)=", EX)
```

Console ~/R/

```
E(X)= 10.6
>
```

Nota Abbiamo utilizzato il comando *prod(v)* che fornisce il prodotto di tutti gli elementi del vettore v. Ad esempio *prod(1:4)=24*. Si è utilizzata inoltre la struttura di controllo *se … allora … altrimenti …*:

if *(condizione) comando* ***else*** *comando*

dove la condizione in questo caso è n==1 (infatti per n=1 il valore che la funzione deve fornire è *a* mentre in tutti gli altri casi è *b*).

Tenete infine presente che dopo aver lanciato questo script in cui è definita la funzione p(n), ogni altro script potrà utilizzare tale funzione. Per verificare che la funzione p(n) rimane in memoria fate clic sulla scheda "environment" di R studio.

In conclusione il valore atteso è 10.6; in concreto dobbiamo aspettarci di alzare in media 10 o 11 carte. Lo avreste detto?

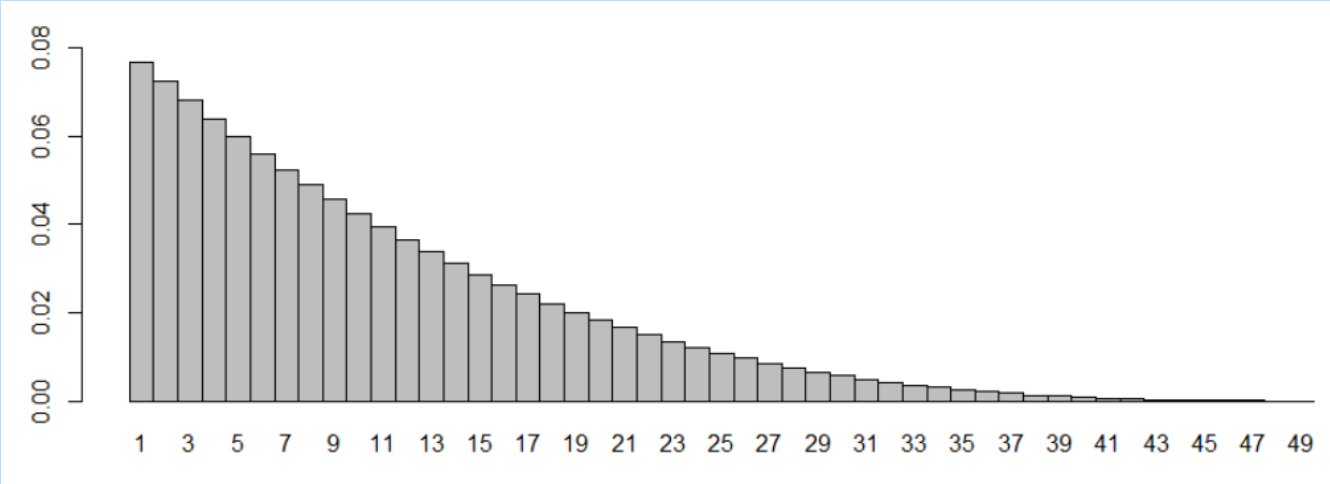
Visto che abbiamo a disposizione la funzione p(n) è interessante ottenere sia la tabella con la distribuzione di probabilità di X sia la rappresentazione grafica di tale distribuzione. Ecco qui a fianco il codice:

Untitled1* Untitled2*

Source on Save Run

```
distr_prob=c()
for (i in 1:49) distr_prob[i]=p(i)
print(cbind(1:49,round(distr_prob,4)))
barplot(distr_prob, names.arg=1:49,space=0,ylim=c(0,0.08))
```

	[,1]	[,2]			
[1,]	1	0.0769	[26,]	26	0.0096
[2,]	2	0.0724	[27,]	27	0.0085
[3,]	3	0.0681	[28,]	28	0.0075
[4,]	4	0.0639	[29,]	29	0.0065
[5,]	5	0.0599	[30,]	30	0.0057
[6,]	6	0.0561	[31,]	31	0.0049
[7,]	7	0.0524	[32,]	32	0.0042
[8,]	8	0.0489	[33,]	33	0.0036
[9,]	9	0.0456	[34,]	34	0.0030
[10,]	10	0.0424	[35,]	35	0.0025
[11,]	11	0.0394	[36,]	36	0.0021
[12,]	12	0.0365	[37,]	37	0.0017
[13,]	13	0.0338	[38,]	38	0.0013
[14,]	14	0.0312	[39,]	39	0.0011
[15,]	15	0.0287	[40,]	40	0.0008
[16,]	16	0.0264	[41,]	41	0.0006
[17,]	17	0.0242	[42,]	42	0.0004
[18,]	18	0.0221	[43,]	43	0.0003
[19,]	19	0.0202	[44,]	44	0.0002
[20,]	20	0.0183	[45,]	45	0.0001
[21,]	21	0.0166	[46,]	46	0.0001
[22,]	22	0.0150	[47,]	47	0.0000
[23,]	23	0.0135	[48,]	48	0.0000
[24,]	24	0.0121	[49,]	49	0.0000
[25,]	25	0.0108			



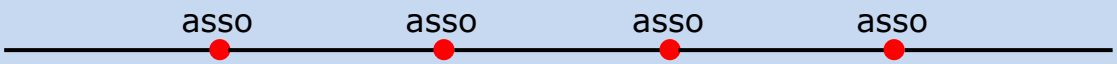
Nota: il comando *cbind(v1, v2, …)* crea una tabella le cui colonne sono formate rispettivamente dagli elementi dei vettori v1, v2, …

Come si vede la probabilità che il primo asso si presenti all'n-esima estrazione decresce rapidamente al crescere di n. E ciò si poteva intuire, ad esempio si capisce che è molto improbabile che il primo asso si presenti alla 30-esima estrazione (la probabilità è 0,0057). Però riflettete sul fatto che, ad esempio, è più probabile che la prima carta sia un asso piuttosto che la seconda carta sia il primo asso.

Soluzione intuitiva

C'è un modo intuitivo (non rigoroso) per renderci conto che il valor medio cercato è 10.6?

Possiamo ragionare così: è poco probabile che, in un mazzo ben mescolato, due assi siano estratti uno dopo l'altro (o comunque uno vicino all'altro), ancor meno probabile che siano estratti tre o addirittura quattro assi consecutivi (o comunque tra loro vicini); quindi dobbiamo aspettarci una situazione in cui, in media, gli assi siano equispaziati nel mazzo. La situazione è quella in figura: i 5 intervalli che si vengono a creare hanno ampiezza (52 – 4)/5 = 9.6 e quindi il primo asso si presenterà, in media, dopo 9.6+1 = 10.6 estrazioni.



Simulazione

Simulazione media per il primo asso.R*

Source on Save | Run

```
cat("\14")
mazzo=c(rep("*",48),rep("A",4))

nrepliche=10
X=rep(0,n)

for (i in 1:nrepliche) {
  carte=sample(mazzo,49,replace=F)
  X[i]=which(carte=="A")[1]
  sequenza=carte[1:X[i]]
  cat(noquote(sequenza),"  X=",X[i],"\n")}

cat("\n", "media=", mean(X))
```

Console ~/R/

```
* * A    X= 3
* * * * * * * * * A    X= 9
* * * * * * * * * * * * * * * * A    X= 16
* * * * * * * * * A    X= 10
* * * * * * * * * * * * * A    X= 13
* A    X= 2
* * * * * * * * * * * A    X= 12
* * * A    X= 4
* * * * * * * * * * A    X= 11
* * * * * * * * * * * * * * * * * * * A    X= 18

 media= 9.8
>
```

Esaminiamo ora in dettaglio il programma.

1. Il mazzo (vettore *mazzo*) viene simulato, ai nostri fini, con un vettore costituito da 48 asterischi (carte "non asso") e 4 lettere "A" (assi); il mazzo è dunque ordinato (e non ben mescolato) ma ciò è ininfluente perché poi il comando *sample* opererà un'estrazione casuale di 49 carte senza reinserimento (generando il vettore *carte*). E' sufficiente estrarre 49 carte per avere la certezza che sia stato estratto almeno un asso.

2. Come facciamo a sapere in quale posizione del vettore *carte* compare il primo asso? Qui entra in gioco il comando *which(carte=="A")* che ci fornisce il vettore delle posizioni della lettera "A" nel vettore carte. Se, ad esempio, il vettore *carte* fosse ***A****A******A, il comando which ci fornirebbe il vettore di posizioni 4, 9, 16; quindi

which(carte=="A")[1]

è il primo elemento di tale vettore cioè è la posizione del primo asso.

3. Nel vettore X, alla posizione i-esima, viene salvata la posizione del primo asso nell'esperimento i-esimo. Dunque il vettore X contiene proprio le realizzazioni della nostra v.c. X. Il vettore *sequenza* contiene la sequenza corrente di carte fino al primo asso, sequenza che viene visualizzata per ogni esperimento assieme al valore di X.

Matrici

Una matrice *mxn* (leggi "m per n") è una tabella ordinata di *m·n* numeri disposti su *m* righe e *n* colonne. Vediamo come creare matrici in R.

Esempio 1 Per creare una matrice utilizzare la funzione *matrix(data=vettore, nrow=m, ncol=n, byrow=F/T)*:


```
Console ~/R/
> v=1:12
> M=matrix(v, nrow=3)
> M
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
>
```

Notare che la matrice M viene riempita, colonna per colonna, utilizzando nell'ordine gli elementi del vettore v. Le righe impostate sono 3 e quindi le colonne, necessariamente, sono 4 (visto che il vettore v ha 12 elementi). Se vogliamo saturare la matrice per righe anziché per colonne dobbiamo assegnare al parametro *byrow* (per righe) il valore TRUE, come si vede nell'esempio seguente. Per default il parametro *byrow* è impostato a FALSE.

Esempio 2

```
Console ~/R/
> v=1:12
> M=matrix(v,nrow=3,byrow=TRUE)
> M
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
>
```

Esempio 3

```
Console ~/R/
> M=matrix(1:12,ncol=3)
> M
     [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3    7   11
[4,]    4    8   12
> M=matrix(1:12,ncol=3,byrow=TRUE)
> M
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
[4,]   10   11   12
>
```

Qui viene impostato il numero di colonne, il numero di righe è determinato di conseguenza.

Esempio 4 Ecco come creare rapidamente una matrice di soli 0:

```
Console ~/R/
> M=matrix(0,nrow=2,ncol=3)
> M
     [,1] [,2] [,3]
[1,]    0    0    0
[2,]    0    0    0
>
```


Esempio 5 Per estrarre l'elemento alla riga *i* e alla colonna *j* di una matrice *M* utilizzare *M[i, j]*:

```
Console ~/R/
> M=matrix(1:12,nrow=3)
> M
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
> M[2,3]
[1] 8
> M[3,2]
[1] 6
>
```

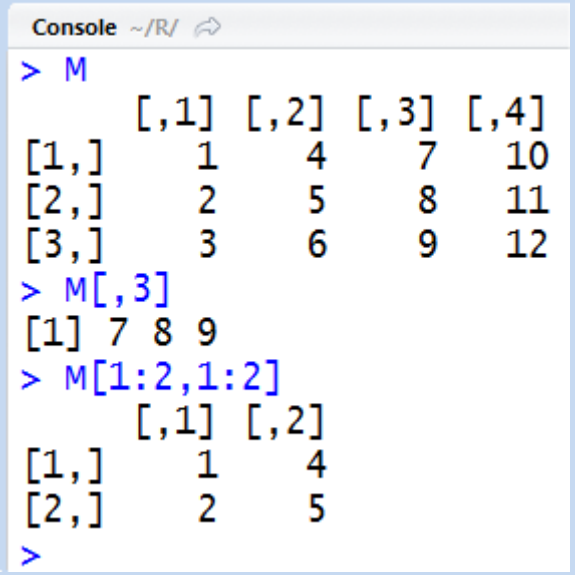
Esempio 6 Per assegnare il valore *x* all'elemento alla riga *i* e alla colonna *j* di una matrice *M* utilizzare *M[i, j]=x*:

```
Console ~/R/
> M
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
> M[2,3]=0
> M
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    0   11
[3,]    3    6    9   12
>
```

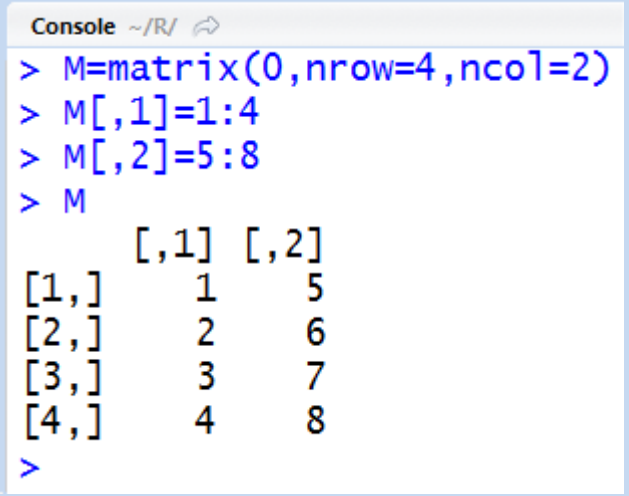
Esempio 7 Per estrarre la riga i-esima, cioè un vettore riga, dalla matrice *M* utilizzare *M[i,]* oppure *M[i, 1:n]* dove *n* è il numero delle colonne:

```
Console ~/R/
> M
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
> M[2,]
[1]  2  5  8 11
> M[3,1:4]
[1]  3  6  9 12
>
```

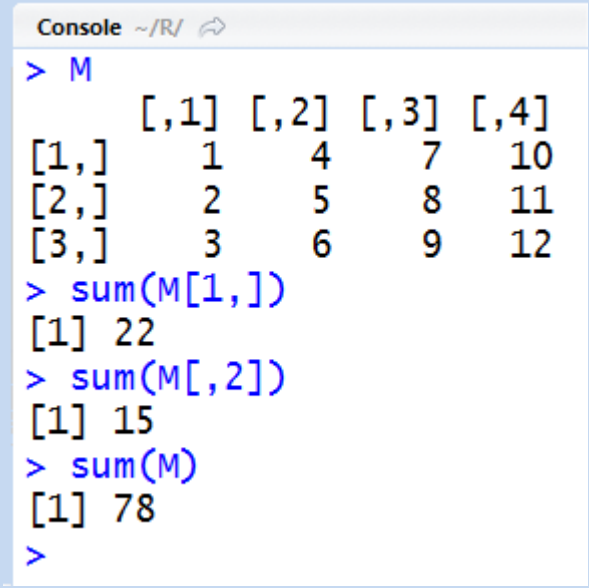
Esempio 8 Per estrarre la colonna i-esima, cioè un vettore colonna, dalla matrice *M* utilizzare *M[, i]* oppure *M[1:m, i]* dove *m* è il numero delle righe; in modo analogo potremo estrarre qualsiasi sottomatrice:



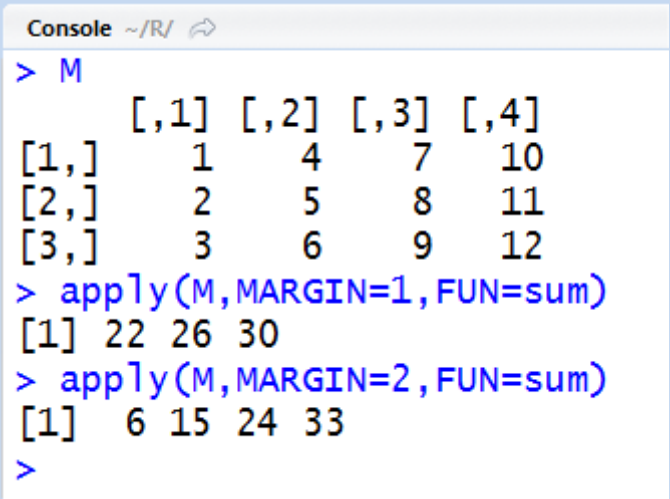
Esempio 9 Qui viene creata una matrice 4x2 impostando le due colonne con due vettori di lunghezza 4 (in modo analogo possiamo creare una matrice assegnando vettori riga):



Esempio 10 Qui vedi come eseguire tre diverse somme: la somma degli elementi della prima riga della matrice M, la somma degli elementi della seconda colonna della matrice M, la somma di tutti gli elementi della matrice M:



Esempio 11 La funzione *apply(X, MARGIN, FUN)* applica una funzione *FUN* alla matrice *X* (alle righe se *MARGIN=1*, alle colonne se *MARGIN=2*, a tutti gli elementi se *MARGIN=c(1,2)*). In questo esempio la funzione applicata alla matrice M è la funzione *sum*: il primo comando *apply* somma gli elementi di ciascuna riga (e restituisce il vettore delle somme di riga), il secondo comando *apply* somma gli elementi di ciascuna colonna (e restituisce il vettore delle somme di colonna):



Esempio 12 Possiamo creare matrici anche utilizzando le funzioni *rbind()* e *cbind()* che consentono di "unire" in una matrice più vettori della stessa lunghezza; *rbind()* dispone i vettori per riga, *cbind()* per colonna:

```
Console ~/R/
> x=1:4
> y=x^2
> z=x^3
> M=rbind(x,y,z)
> M
  [,1] [,2] [,3] [,4]
x    1    2    3    4
y    1    4    9   16
z    1    8   27   64
> N=cbind(x,y,z)
> N
     x  y  z
[1,] 1  1  1
[2,] 2  4  8
[3,] 3  9 27
[4,] 4 16 64
>
```

Nota: impostare il parametro *deparse.level* a 0 se si vuole che i "nomi" di riga o colonna "x", "y", "z" siano sostituiti dai soliti indici di riga o colonna (come negli esempi precedenti).

Esempio 13 Prodotto di due matrici *mxn* elemento per elemento (simbolo *) :

```
Console ~/R/
> A=matrix(c(1,2,3,0,1,2,-1,0,2),nrow=3)
> A
     [,1] [,2] [,3]
[1,]    1    0   -1
[2,]    2    1    0
[3,]    3    2    2
> B=matrix(c(-2,0,2,1,3,2,-3,0,-2),nrow=3)
> B
     [,1] [,2] [,3]
[1,]   -2    1   -3
[2,]    0    3    0
[3,]    2    2   -2
> A*B
     [,1] [,2] [,3]
[1,]   -2    0    3
[2,]    0    3    0
[3,]    6    4   -4
>
```

Esempio 14 Prodotto matriciale righe per colonne (simbolo %*%):

Nota Tale prodotto è applicabile solo se le dimensioni delle matrici A e B sono rispettivamente *mxn* e *nxp*.

```
Console ~/R/
> A %*% B
     [,1] [,2] [,3]
[1,]   -4   -1   -1
[2,]   -4    5   -6
[3,]   -2   13  -13
>
```

Distribuzioni di probabilità congiunte

Siano X e Y due variabili aleatorie discrete; siano rispettivamente $x_1, x_2, \ldots, x_m$ e $y_1, y_2, \ldots, y_n$ i valori che le variabili possono assumere. Si chiama ***distribuzione di probabilità congiunta*** la funzione

$$p(x_i, y_j) = p(X=x_i \text{ e } Y=y_j)$$

cioè la funzione che ad ogni coppia (x_i, y_j) associa la probabilità che sia $X=x_i$ e simultaneamente sia $Y=y_j$.

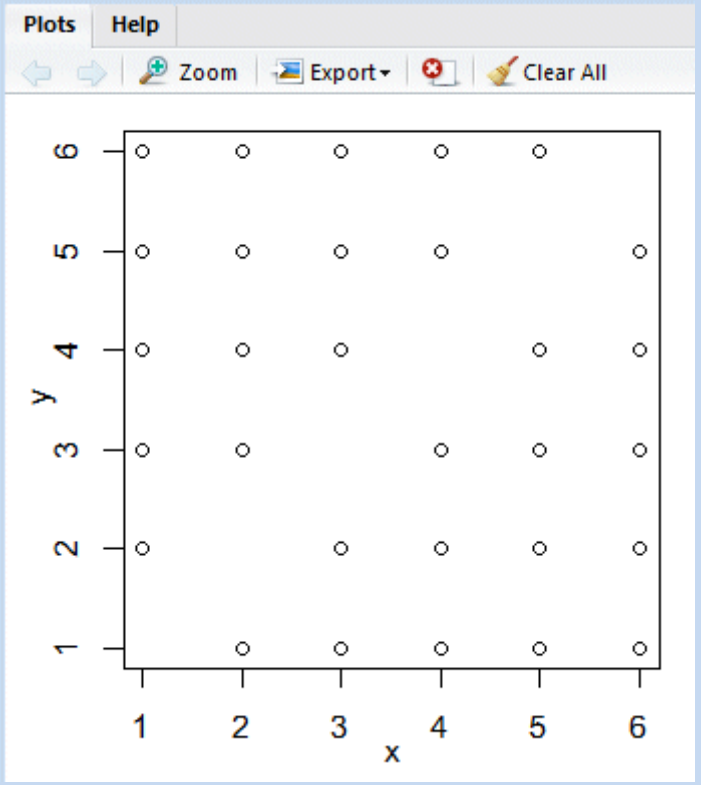
Esempio 1 Un urna contiene 6 biglie numerate da 1 a 6, se ne estraggono due senza reinserimento; sia X il primo numero estratto e Y il secondo. Determinare la distribuzione congiunta di X e Y.

I valori possibili per la v.c. X sono 1, 2, …, 6; stessa cosa per la v.c. Y, presa singolarmente (infatti se non abbiamo alcuna informazione sulla prima biglia estratta, la v.c. X, i valori possibili per la v.c. Y sono 1, 2, …, 6). Se però consideriamo le due v.c. congiuntamente è chiaro che non potremo mai avere le estrazioni (1, 1), (2, 2), …, (6, 6) tenendo conto del non reinserimento (tali coppie hanno probabilità zero di essere estratte). Le coppie (i, j) con $i \neq j$ hanno invece tutte la stessa probabilità di essere estratte pari a $(1/6)\cdot(1/5)=1/30$. Quindi nel nostro caso la distribuzione congiunta di X e Y è la funzione costante $p(x_i, y_j)=1/30$. Possiamo rappresentare facilmente con R, nel piano cartesiano,

le coppie (i, j) ammissibili, cioè con probabilità non nulla, e possiamo anche generare la tabella della distribuzione di probabilità congiunta (sarà una matrice).

Ecco il codice per la rappresentazione cartesiana delle coppie a probabilità non nulla:

```
x=c()
y=c()
for (i in 1:6) {
  for (j in 1:6) {
    if (i!=j) {
      x=c(x,i)
      y=c(y,j)}
  }
}
par(pty="s")
plot(x,y)
```



Nota I due comandi x=c() e y=c() servono ad inizializzare i due vettori x e y.

Qui si è utilizzata la struttura di controllo *se … allora …*:

if *(condizione) comando*

dove la condizione è *(i!=j)* cioè *i* diverso da *j.*

Il comando x=c(x, i) serve a ridefinire il vettore x, inizialmente vuoto, concatenando ad x il valore i; idem per il vettore y.

Come sapete la funzione *par()* serve ad impostare i parametri grafici; in questo caso il settaggio *pty="s"* imposta una regione quadrata (**s**quare) per il grafico.

Esempio 2 Costruire la tabella delle probabilità congiunte relative alle variabili casuali dell'esempio precedente:

temp1.R*

Source on Save

```
cat("\14")
m=matrix(0,nrow=6,ncol=6)
rownames(m)=c("x=1","x=2","x=3","x=4","x=5","x=6")
colnames(m)=c("y=1","y=2","y=3","y=4","y=5","y=6")
for (i in 1:6)
  for (j in 1:6)
    if (i!=j) m[i,j]=1/30 else m[i,j]=0
print(round(m,4))
```

Console ~/R/

```
     y=1    y=2    y=3    y=4    y=5    y=6
x=1 0.0000 0.0333 0.0333 0.0333 0.0333 0.0333
x=2 0.0333 0.0000 0.0333 0.0333 0.0333 0.0333
x=3 0.0333 0.0333 0.0000 0.0333 0.0333 0.0333
x=4 0.0333 0.0333 0.0333 0.0000 0.0333 0.0333
x=5 0.0333 0.0333 0.0333 0.0333 0.0000 0.0333
x=6 0.0333 0.0333 0.0333 0.0333 0.0333 0.0000
>
```


Nota sul codice Qui si è utilizzata la struttura di controllo *se … allora … altrimenti …*:

if *(condizione) comando* ***else*** *comando*

dove la condizione è *(i!=j)* cioè *i* diverso da *j*.
Le funzioni *rownames()* e *colnames()* ci consentono di assegnare dei nomi alle righe e alle colonne della tabella.

Nota sulla tabella La somma di tutti gli elementi della tabella è 1. Questo fatto deve essere chiaro sul piano probabilistico: la tabella contempla tutte le possibili coppie dei valori di X e di Y, una di queste coppie deve necessariamente verificarsi, quindi la probabilità dell'unione di tutti gli eventi incompatibili

$E_{i,j}$ = "si presenta la coppia (i, j)"

con i=1, …, 6, j=1, …, 6 è 1 (probabilità dell'evento certo). Verifichiamolo con R:

Console ~/R/

```
> sum(m)
[1] 1
>
```

La somma degli elementi della riga i-esima della tabella è p(X=i)=1/6=0.166…; ad esempio la somma degli elementi della prima riga rappresenta la probabilità dell'unione di tutti i possibili eventi $E_{1,j}$ in cui X=1, quindi rappresenta la probabilità che sia X=1, considerando singolarmente la v.c. X:

```
Console ~/R/
> m[1,]
       y=1        y=2        y=3        y=4        y=5        y=6
0.00000000 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333
> sum(m[1,])
[1] 0.1666667
>
```


Analogamente la somma degli elementi della colonna j-esima della tabella è p(Y=j)=1/6=0.166…; ad esempio la somma degli elementi della prima colonna rappresenta la probabilità dell'unione di tutti i possibili eventi $E_{j,1}$ un cui Y=1, quindi rappresenta la probabilità che sia Y=1, considerando singolarmente la v.c. Y:

```
Console ~/R/
> m[,1]
       x=1        x=2        x=3        x=4        x=5        x=6
0.00000000 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333
> sum(m[,1])
[1] 0.1666667
>
```

Quindi data la distribuzione congiunta delle v.c. X e Y possiamo ricostruire la distribuzione di X e Y prese singolarmente (ma non è vero il viceversa, come vedremo); quando la distribuzione di probabilità di X (o di Y) viene ottenuta in questo modo si parla di distribuzione di probabilità *marginale* (tale termine deriva dal fatto che le probabilità, ottenute sommando gli elementi delle righe o delle colonne, possono essere scritte ai margini della matrice).

Osserviamo che dalla tabella delle probabilità congiunte si deduce che non solo la v.c. X, ma anche Y, presa singolarmente, ha distribuzione di probabilità uniforme (ciò è già stato sottolineato nel commento all'esempio 1). Osserviamo infine che, considerando l'urna dell'esempio 1 e l'estrazione di due biglie senza rimessa e con rimessa, troveremmo due tabelle 6x6 di probabilità congiunta ovviamente diverse (una è quella dell'esempio 2, l'altra ha 36 elementi tutti uguali a 1/36); risalendo però alle probabilità marginali ritroveremmo in entrambi i casi distribuzioni di probabilità uniformi sia per X sia per Y. Ciò significa che dalla distribuzione di X e Y non possiamo derivare la distribuzione congiunta.

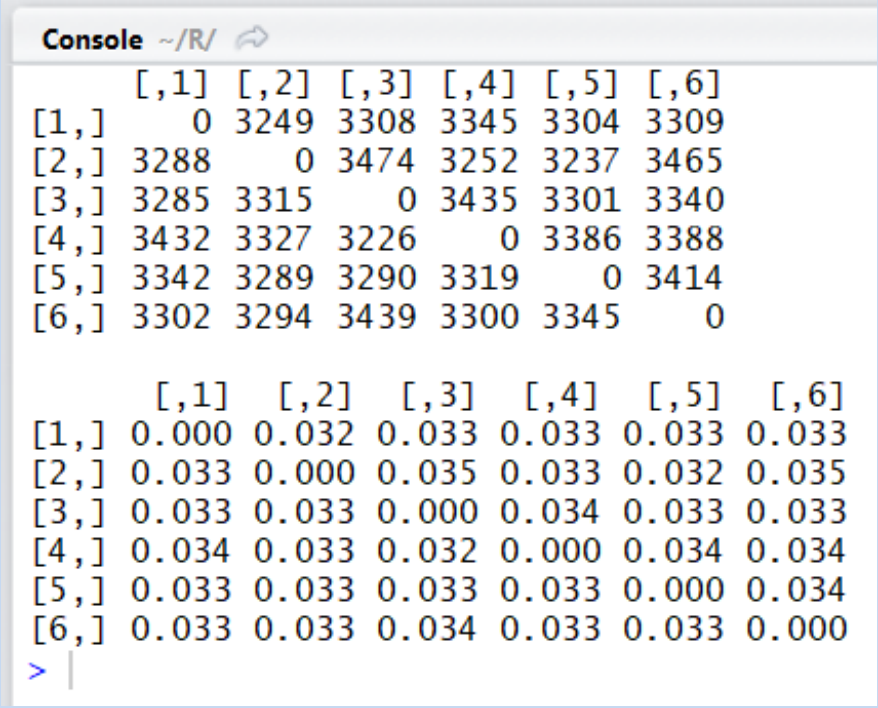
Esempio 3 Simulare l'esperimento dell'esempio 1 e determinare frequentisticamente la distribuzione di probabilità congiunta delle variabili X e Y.

tabella probabilità congiunte.R*

Source on Save

```
cat("\14")
n=100000
tabella_dati=matrix(0,nrow=6,ncol=6)
tabella_prob_congiunte=matrix(0,nrow=6,ncol=6)

for (i in 1:n)
  {urna=1:6
  biglie_estratte=sample(urna,size=2,replace=FALSE)
  biglia1=biglie_estratte[1]
  biglia2=biglie_estratte[2]
  tabella_dati[biglia1,biglia2]=tabella_dati[biglia1,biglia2]+1}

tabella_prob_congiunte=tabella_dati/n

print(tabella_dati)
cat("\n") # a capo
print(round(tabella_prob_congiunte,3))
```

Output:



Esempio 4 Dalla solita urna dell'esempio 1 sono estratte due biglie senza rimessa. Qual è la probabilità che i risultati delle due estrazioni differiscano al più di 2?

Modellizzando la situazione con le due v.c. X e Y dell'esempio 1, si tratta di calcolare $p(|X-Y|\leq 2)$. Noi conosciamo la tabella della distribuzione congiunta di X e Y, tutte le coppie (i, j) con $i \neq j$ hanno probabilità 1/30. Non ci resta che contare le coppie (i, j) ammissibili tali che $(|i-j|\leq 2$. Ecco il codice:

temp.R*

Source on Save

```
contatore=0
for (i in 1:6) {
  for (j in 1:6) {
    if (abs(i-j)<=2 & i!=j) contatore=contatore+1
  }}
print(contatore)
```

Console ~/R/

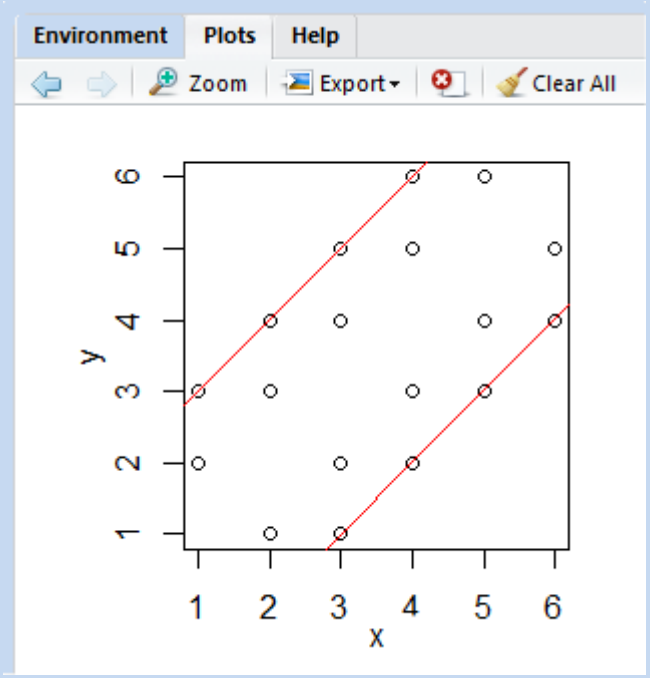
```
> source('~/.ac
[1] 18
```

Le coppie sono quindi 18 e la probabilità cercata è 18/30=3/5=0.6. Qui vedete il codice per rappresentare nel piano cartesiano le 18 coppie:

temp.R

Source on Save

```
x=c()
y=c()
for (i in 1:6) {
  for (j in 1:6) {
    if (i!=j & abs(i-j)<=2) {
      x=c(x,i)
      y=c(y,j)}
  }
}
par(pty="s")
plot(x,y)
abline(a=-2,b=1,col="red")
abline(a=2,b=1,col="red")
```



Notare che la condizione abs(i-j)≤2 equivale al sistema:

$$-2 \le i-j \le 2 \quad (-2 \le x-y \le 2)$$

le cui soluzioni sono i punti compresi tra e sulle rette y=x+2 e y=x-2.

Il prossimo esempio introduce l'importante nozione di *indipendenza di variabili aleatorie*.

Esempio 5 Data la seguente tabella di probabilità congiunta per le v.c. X e Y, simulare *n* valori congiunti (x, y) e verificare che la tabella delle frequenze relative congiunte (tabella di contingenza), al crescere di *n*, approssima la tabella delle probabilità congiunte. Mostrare che le v.c. X e Y sono dipendenti.

X Y→ ↓	**4**	**5**	**6**
1	0.20	0.05	0.10
2	0.01	0.08	0.01
3	0.10	0.20	0.25

temp3.R*

Source on Save

```
cat("\14")
v=c(0.2,0.01,0.1,0.05,0.08,0.2,0.1,0.01,0.25)
M=matrix(v,nrow=3)
rownames(M)=1:3
colnames(M)=4:6

distr_probX=apply(M,MARGIN=1,FUN=sum)
distr_probY=apply(M,MARGIN=2,FUN=sum)

cat("Distribuzione di probabilità
    congiunte di X e Y: \n")
print(M)
cat("\n")
cat("Distribuzione (marginale) di X: \n")
print(distr_probX)
cat("\n")
cat("Distribuzione (marginale) di Y: \n")
print(distr_probY)
cat("\n")
```

```
n=10
simulX=c()
simulY=c()
for (i in 1:n) {
  x=sample(1:3,1,prob=distr_probX)
  simulX[i]=x
  distr_condY=M[x,]/distr_probX[x]
  y=sample(4:6,1,prob=distr_condY)
  simulY[i]=y}
cat("Risultato della simulazione: \n")
print(cbind(simulX,simulY))
cat("\n")
cat("Distribuzione di frequenze rel.
congiunte di X e Y: \n")
t=table(simulX,simulY)/n
print(round(t,2))
```


Nota Dopo aver generato un valore x dobbiamo simulare un valore y condizionato dal fatto che si è presentato x, quindi dobbiamo determinare la distribuzione di probabilità di Y dato l'evento X=x. Terremo presente che p(X=x e Y=y) = M[x,y] = p(X=x)·p(Y=y|X=x) e quindi p(Y=y|X=x) = M[x,y]/p(X=x).

E questo è l'output:

```
Console ~/R/
Distribuzione di probabilità
congiunte di X e Y:
     4    5    6
1 0.20 0.05 0.10
2 0.01 0.08 0.01
3 0.10 0.20 0.25

Distribuzione (marginale) di X:
   1    2    3
0.35 0.10 0.55

Distribuzione (marginale) di Y:
   4    5    6
0.31 0.33 0.36

Risultato della simulazione:
      simulX simulY
 [1,]      3      6
 [2,]      2      5
 [3,]      3      4
 [4,]      3      5
 [5,]      1      4
 [6,]      2      5
 [7,]      3      6
 [8,]      3      6
 [9,]      1      4
[10,]      3      5

Distribuzione di frequenze rel.
congiunte di X e Y:
      simulY
simulX   4   5   6
     1 0.2 0.0 0.0
     2 0.0 0.2 0.0
     3 0.1 0.2 0.3
>
```

Cosa significa che due v.c. X e Y sono **indipendenti**? Dovresti aspettarti questa definizione: X e Y sono indipendenti se gli eventi X=x e Y=y sono indipendenti per ogni scelta dei valori x e y che le due variabili X e Y possono assumere. Intuitivamente: in nessun caso il valore assunto da X condiziona il valore assunto da Y (e viceversa). Noi sappiamo che gli eventi X=x e Y=y sono indipendenti se si ha $p(X=x \text{ e } Y=y) = p(X=x) \cdot p(Y=y)$. Verifichiamo la dipendenza, cioè la non indipendenza, delle nostre variabili X e Y con R:


```
Console ~/R/
> M
     4    5    6
1 0.20 0.05 0.10
2 0.01 0.08 0.01
3 0.10 0.20 0.25
> distr_probX
   1    2    3
0.35 0.10 0.55
> distr_probY
   4    5    6
0.31 0.33 0.36
> M[1,1]==distr_probX[1]*distr_probY[1]
    1
FALSE
>
```

Basta infatti verificare che in un sol caso non si ha $p(X=x \text{ e } Y=y) = p(X=x) \cdot p(Y=y)$; qui abbiamo verificato che $p(X=1, Y=4) \neq p(X=1) \cdot p(Y=4)$.

Proprietà del valore atteso

Se X e Y sono variabili casuali, dipendenti o indipendenti, e k una costante si ha:

$$E(kX) = kE(X)$$
$$E(X+Y)=E(X)+E(Y)$$

linearità

Se X e Y sono variabili casuali <u>indipendenti</u> si ha:

$$E(X \cdot Y) = E(X) \cdot E(Y)$$

La prima proprietà si dimostra facilmente (provate!). Le altre due non le dimostreremo ma vogliamo fare qualche verifica.

Esempio 1 Verificare la seconda proprietà nel caso delle due v.c. X e Y dell'esempio precedente di cui conosciamo la distribuzione congiunta. Le due variabili sono dipendenti e questa è la situazione in cui la proprietà è tutt'altro che evidente.

Cominciamo a calcolare il valore atteso di X e quello di Y:

```
> EX=1*0.35+2*0.1+3*0.55
> EX
[1] 2.2
```

```
> EY=4*0.31+5*0.33+6*0.36
> EY
[1] 5.05
```

Quindi E(X)+E(Y)=2.2+5.05=7.25.

Ora dobbiamo calcolare il valore atteso della nuova variabile casuale Z=X+Y. Qui a fianco vediamo due tabelle: la prima ci fornisce i valori di Z (tutte le somme possibili), la seconda ci fornisce la probabilità di ciascuna somma ed è proprio la tabella delle probabilità congiunte dell'esempio precedente.

X Y→ ↓	4	5	6
1	5	6	7
2	6	7	8
3	7	8	9

X Y→ ↓	4	5	6
1	0.20	0.05	0.10
2	0.01	0.08	0.01
3	0.10	0.20	0.25

Calcoliamo il valore atteso di Z:

```
> M
     4    5    6
1 0.20 0.05 0.10
2 0.01 0.08 0.01
3 0.10 0.20 0.25
```

```
> N=matrix(c(5,6,7,6,7,8,7,8,9),nrow=3)
> N
     [,1] [,2] [,3]
[1,]    5    6    7
[2,]    6    7    8
[3,]    7    8    9
> EZ=sum(N*M)
> EZ
[1] 7.25
```

Quindi E(Z)=E(X+Y)=7.25. Ricordate che qui R esegue il prodotto di matrici elemento per elemento.

Nota Per calcolare E(Z) potevamo procedere anche in un altro modo. I valori che Z può assumere sono 5, 6, 7, 8, 9. Calcoliamo la distribuzione di probabilità di Z, tenendo conto della distribuzione congiunta di X e Y:

Z (=X+Y)	5	6	7	8	9
prob.	0.2	0.05+0.01= 0.06	0.1+0.08+0.1= 0.28	0.01+0.2= 0.21	0.25

Quindi: $E(Z) = 5 \cdot 0.2+6 \cdot 0.06+7 \cdot 0.28+8 \cdot 0.21+9 \cdot 0.25 = 7.25$

Esempio 2 Considera la solita urna con sei biglie numerate da 1 a 6 e l'estrazione di due biglie nelle due modalità con rimessa e senza rimessa. Siano inoltre X e Y i numeri ottenuti alla prima e alla seconda estrazione. Mostra che nel caso di estrazioni con rimessa (variabili indipendenti) si ha $E(XY)=E(X)E(Y)$ mentre nel caso di estrazioni senza rimessa si ha $E(XY)\neq E(X)E(Y)$.

Primo caso, var. indipendenti Calcoliamo E(XY):

Untitled1*

Source on Save

```
cat("\14")
#N=matrice dei valori XY
N=matrix(0,nrow=6,ncol=6)
for (i in 1:6)
  for(j in 1:6)
    N[i,j]=i*j
cat("Matrice dei valori XY: \n")
print(N)

#M=matrice delle probabilità congiunte
M=matrix(rep(1/36,36),nrow=6)
cat("Matrice delle probabilità congiunte: \n")
print(round(M,4))

#valore atteso di Z=XY
cat("EZ= \n")
EZ=sum(N*M)
print(EZ)
```

Console ~/R/

```
Matrice dei valori XY:
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    2    3    4    5    6
[2,]    2    4    6    8   10   12
[3,]    3    6    9   12   15   18
[4,]    4    8   12   16   20   24
[5,]    5   10   15   20   25   30
[6,]    6   12   18   24   30   36
Matrice delle probabilità congiunte:
       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
[1,] 0.0278 0.0278 0.0278 0.0278 0.0278 0.0278
[2,] 0.0278 0.0278 0.0278 0.0278 0.0278 0.0278
[3,] 0.0278 0.0278 0.0278 0.0278 0.0278 0.0278
[4,] 0.0278 0.0278 0.0278 0.0278 0.0278 0.0278
[5,] 0.0278 0.0278 0.0278 0.0278 0.0278 0.0278
[6,] 0.0278 0.0278 0.0278 0.0278 0.0278 0.0278
EZ=
[1] 12.25
```

Calcoliamo E(X)E(Y):

```
Console ~/R/
> valori=1:6
> EX=sum(valori*1/6)
> EY=EX
> EY*EX
[1] 12.25
>
```


Secondo caso, var. dipendenti Calcoliamo E(XY)

Untitled1*

Source on Save

```
cat("\14")
#N=matrice dei valori XY
N=matrix(0,nrow=6,ncol=6)
for (i in 1:6)
  for(j in 1:6)
    N[i,j]=i*j
cat("Matrice dei valori XY: \n")
print(N)

#M=matrice delle probabilità congiunte
M=matrix(0,nrow=6,ncol=6)
for (i in 1:6)
  for (j in 1:6)
    if (i!=j) M[i,j]=1/30 else M[i,j]=0
cat("Matrice delle probabilità congiunte: \n")
print(round(M,4))

#valore atteso di Z=XY
cat("EZ: \n")
EZ=sum(N*M)
print(EZ)
```

```
Console ~/R/
Matrice dei valori XY:
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    2    3    4    5    6
[2,]    2    4    6    8   10   12
[3,]    3    6    9   12   15   18
[4,]    4    8   12   16   20   24
[5,]    5   10   15   20   25   30
[6,]    6   12   18   24   30   36
Matrice delle probabilità congiunte:
       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
[1,] 0.0000 0.0333 0.0333 0.0333 0.0333 0.0333
[2,] 0.0333 0.0000 0.0333 0.0333 0.0333 0.0333
[3,] 0.0333 0.0333 0.0000 0.0333 0.0333 0.0333
[4,] 0.0333 0.0333 0.0333 0.0000 0.0333 0.0333
[5,] 0.0333 0.0333 0.0333 0.0333 0.0000 0.0333
[6,] 0.0333 0.0333 0.0333 0.0333 0.0333 0.0000
EZ:
[1] 11.66667
```

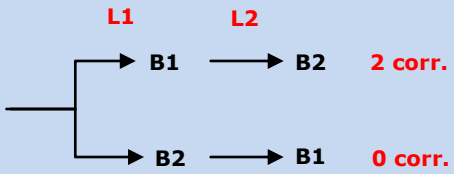
Poiché anche in questo caso si ha E(X)=E(Y)=3.5 e quindi E(X)E(Y)=12.25 abbiamo verificato che $E(XY) \neq E(X)E(Y)$

Il prossimo classico problema (matching problem) mostra quanto sia utile la linearità del valore atteso.

Esempio 3 (La segretaria maldestra) Una segretaria compila *n* lettere e *n* buste con il relativo indirizzo. Lettere e buste cadono dalla scrivania e si mescolano casualmente. La segretaria accoppia lettere e buste in modo casuale, nel senso che una lettera ha la stessa probabilità di finire in ognuna delle *n* buste. In media, quante lettere saranno recapitate al loro vero destinatario?

Tentativo ingenuo di soluzione mediante un diagramma ad albero

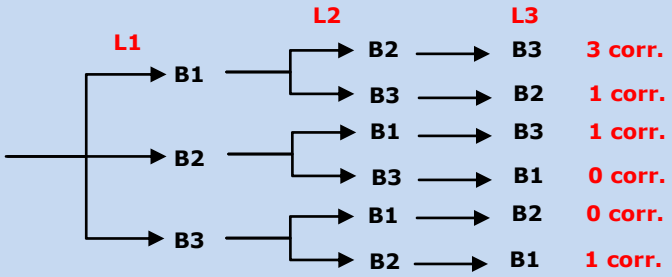
Supponiamo che le lettere e quindi le buste siano solo 2. Chiamiamo L1, L2 le lettere, B1, B2 le buste. La situazione è questa:



Due casi possibili ciascuno di prob. 1/2:
1. (L1, B1) e (L2, B2) cioè 2 corrispondenze
2. (L1, B2) e (L2, B1) cioè 0 corrispondenze

Ne segue: media = $0 \cdot 1/2 + 2 \cdot 1/2 = 1$

Se le lettere sono 3, la situazione è questa:



Sei casi possibili ciascuno di prob. 1/6:
1. (L1, B1) e (L2, B2) e (L3, B3) cioè 3 corrispondenze
2. (L1, B1) e (L2, B3) e (L3, B2) cioè 1 corrispondenza
3. (L1, B2) e (L2, B1) e (L3, B3) cioè 1 corrispondenze
4. (L1, B2) e (L2, B3) e (L3, B1) cioè 0 corrispondenze
5. (L1, B3) e (L2, B1) e (L3, B2) cioè 0 corrispondenze
6. (L1, B3) e (L2, B2) e (L3, B1) cioè 1 corrispondenza

Ne segue: media = $0 \cdot 2/6 + 1 \cdot 3/6 + 3 \cdot 1/6 = 1$

In entrambe le situazioni esaminate, *n=1* e *n=2*, il valore atteso cercato è 1. Ci rendiamo conto però che procedendo in questo modo le cose si complicano sempre di più e, dati i condizionamenti, è difficile determinare una regola generale che valga per ogni *n*. Ci serve uno strumento più potente di un diagramma ad albero per affrontare il problema.

Soluzione La variabile casuale che rappresenta la nostra situazione è:

X = "numero di lettere, delle n compilate, che finiscono nella busta corrispondente"

A noi interessa il valor medio di X. Ora l'idea è questa (ed è un'idea che potremo applicare in molte situazioni): esprimiamo la v.c. X come somma di variabili casuali più semplici:

$$X = X_1 + X_2 + \ldots + X_n$$

$$\text{dove } X_i = \begin{cases} 1 \text{ se la lettera i-esima finisce nella propria busta} \\ 0 \text{ altrimenti} \end{cases}$$

Notate che le variabili sono dipendenti: se ad es. $X_1=X_2=\ldots=X_{n-1}=1$ allora necessariamente $X_n=1$.
Per la linearità del valore atteso, anche nel caso di v.c. dipendenti, si ha

$$E(X) = E(X_1)+ E(X_2) + \ldots + E(X_n)$$

ed è facile calcolare il valore atteso di ogni X_i; infatti si ha $p(X_i=1)=1/n$ (qui il bello è che la probabilità è valutata per la variabile X_i presa singolarmente e una lettera ha la stessa probabilità di finire in ognuna delle *n* buste) quindi $E(X_i)=0\cdot p(X_i=0)+1\cdot p(X_i=1)=1/n$. In conclusione:

$$E(X) = \underbrace{1/n + 1/n + \ldots + 1/n}_{\text{n volte}} = n/n = 1$$

Dunque, per qualsiasi numero *n* di lettere, in media solo una lettera raggiungerà la propria destinazione. Lo avreste detto?

Simulazione con R

La segretaria maldestra.R*

Source on Save | Run

```
cat("\14")
nrepliche=5
nlettere=5
risultati=c()

# x=vettore lettere (numerate da 1 a nlettere)
# y=vettore casuale buste (permutazione casuale del vettore x)
x=1:nlettere
for (i in 1:nrepliche){
  y=sample(1:nlettere,nlettere,replace=FALSE)
  confronto=x==y
  print(rbind(x,y))
  cat(confronto,"\n\n")
  risultati[i]=length(confronto[confronto==TRUE])
}
cat("N. lettere nella busta giusta (per ogni esperimento): \n")
cat(risultati,"\n")
cat("Media= \n",mean(risultati))
```


Nota La variabile *confronto* è un vettore booleano infatti l'uguaglianza x==y viene testata per ogni elemento del vettore x e del corrispondente elemento del vettore y.

Ecco l'output (nel caso di 5 repliche dell'esperimento):

```
Console ~/R/
  [,1] [,2] [,3] [,4] [,5]
x    1    2    3    4    5
y    3    5    4    1    2
FALSE FALSE FALSE FALSE FALSE

  [,1] [,2] [,3] [,4] [,5]
x    1    2    3    4    5
y    1    4    5    2    3
TRUE FALSE FALSE FALSE FALSE

  [,1] [,2] [,3] [,4] [,5]
x    1    2    3    4    5
y    3    2    1    4    5
FALSE TRUE FALSE TRUE TRUE

  [,1] [,2] [,3] [,4] [,5]
x    1    2    3    4    5
y    1    2    4    5    3
TRUE TRUE FALSE FALSE FALSE

  [,1] [,2] [,3] [,4] [,5]
x    1    2    3    4    5
y    5    4    3    2    1
FALSE FALSE TRUE FALSE FALSE

N. lettere nella busta giusta (per ogni esperimento):
0 1 3 2 1
Media=
 1.4
>
```

Esempio 4 Se lancio 20 volte una moneta equa, quante volte devo aspettarmi TESTA? Se lancio 10 volte un dado equo, quante volte devo aspettarmi 6? In generale: se X è una variabile casuale con distribuzione binomiale B(k, p), qual è il suo valore atteso?

Conviene affrontare il caso generale, ragionando in astratto. Sappiamo che:

X="numero di successi su k prove indipendenti ciascuna con probabilità di successo p"

I valori che X può assumere sono 0, 1, 2, …, k.

Possiamo esprimere X così:

$$X = X_1 + X_2 + \ldots + X_k$$

con

$$X_i = \begin{cases} 1 \text{ se si verifica SUCCESSO all'i-esima prova} \\ 0 \text{ altrimenti} \end{cases}$$

e con $p(X_i=1)=p$, $i=1,2, \ldots, k$.

Il valore atteso di ogni X_i è $E(X_i) = 1 \cdot p + 0 \cdot (1-p) = p$. Per la linearità del valore atteso si ha:

$$E(X) = E(X_1 + X_2 + \ldots + X_k) = E(X_1) + E(X_2) + \ldots + E(X_k) = kp$$

Nel caso del lancio della moneta si ha k=20 e p=1/2, quindi il valore atteso è $kp = 20 \cdot (1/2)=10$.
Nel caso del lancio del dado si ha k=10 e p=1/6, quindi il valore atteso è $kp= 10 \cdot (1/6)=5/3=1.666\ldots$

Proprietà della varianza

Se X e Y sono variabili casuali, dipendenti o indipendenti, e k una costante si ha:

$$var(X+k) = var(X)$$
$$var(kX) = k^2 var(X)$$

Se X e Y sono variabili casuali indipendenti si ha:

$$var(X+Y) = var(X-Y) = var(X)+var(Y)$$

Inoltre c'è una formula alternativa per la varianza:

$$var(X) = E(X^2) - E(X)^2$$

Dimostriamo la formula alternativa, ricordando le proprietà del valore atteso, ponendo m=E(X) e ricordando che il valore atteso di una costante k è k (posso considerare una costante k come una variabile casuale che assume l'unico valore k con probabilità 1):

$var(X) = E[(X-m)^2] = E[X^2-2mX+m^2] = E(X^2)-2mE(X)+E(m^2) = E(X^2)-2m^2+m^2 = E(X^2)-E(X)^2$

Dimostriamo che var(X+k)=var(X), ponendo m=E(X):

$var(X+k) = E[(X+k - E(X+k))^2] = E[(X+k-m-k)^2] = E[(X-m)^2] = var(X)$

Dimostriamo che $var(kX)=k^2var(X)$, utilizzando la formula alternativa e ponendo m=E(X):

$var(kX) = E[(kX-E(kX))^2] = E[(kX-km)^2] = E[k^2X^2-2k^2mX+k^2m^2] = E(k^2X^2)-2k^2m^2+k^2m^2 = k^2E(X^2)-k^2m^2 = k^2(E(X^2)-E(X)^2) = k^2var(X)$

Il prossimo esempio introduce una questione di capitale importanza sia nel calcolo delle probabilità sia in statistica inferenziale.

Esempio 1 Lanciamo *n* volte un dado equo. Indichiamo con le v.c. X_1, X_2, …, X_n i valori ottenuti ad ogni lancio e consideriamo la v.c. $\bar{X} = \frac{X_1+X_2+\cdots+X_n}{n}$ che prende il nome di *media campionaria*. Calcolare valore atteso e varianza di $\bar{X}$ in funzione di n. Cosa si può dire della distribuzione di probabilità di $\bar{X}$?

Sappiamo che:

$$E(X_i) = 1 \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} + \cdots + 6 \cdot \frac{1}{6} = \frac{7}{2}$$

$$var(X_i) = \left(1 - \frac{7}{2}\right)^2 \cdot \frac{1}{6} + \left(2 - \frac{7}{2}\right)^2 \cdot \frac{1}{6} + \cdots + \left(6 - \frac{7}{2}\right)^2 \cdot \frac{1}{6} = \frac{35}{12}$$

Grazie alla linearità del valore atteso si ha:

$$E(\bar{X}) = \frac{1}{n}[E(X_1) + E(X_2) + \cdots + E(X_n)] = \frac{1}{n}\left[n\frac{7}{2}\right] = \frac{7}{2}$$

E per le proprietà della varianza nel caso di variabili indipendenti (come lo sono evidentemente le X_i):

$$var(\bar{X}) = \frac{1}{n^2}[var(X_1) + var(X_2) + \cdots + var(X_n)] = \frac{1}{n^2}\left[n\frac{35}{12}\right] = \frac{1}{n}\frac{35}{12}$$

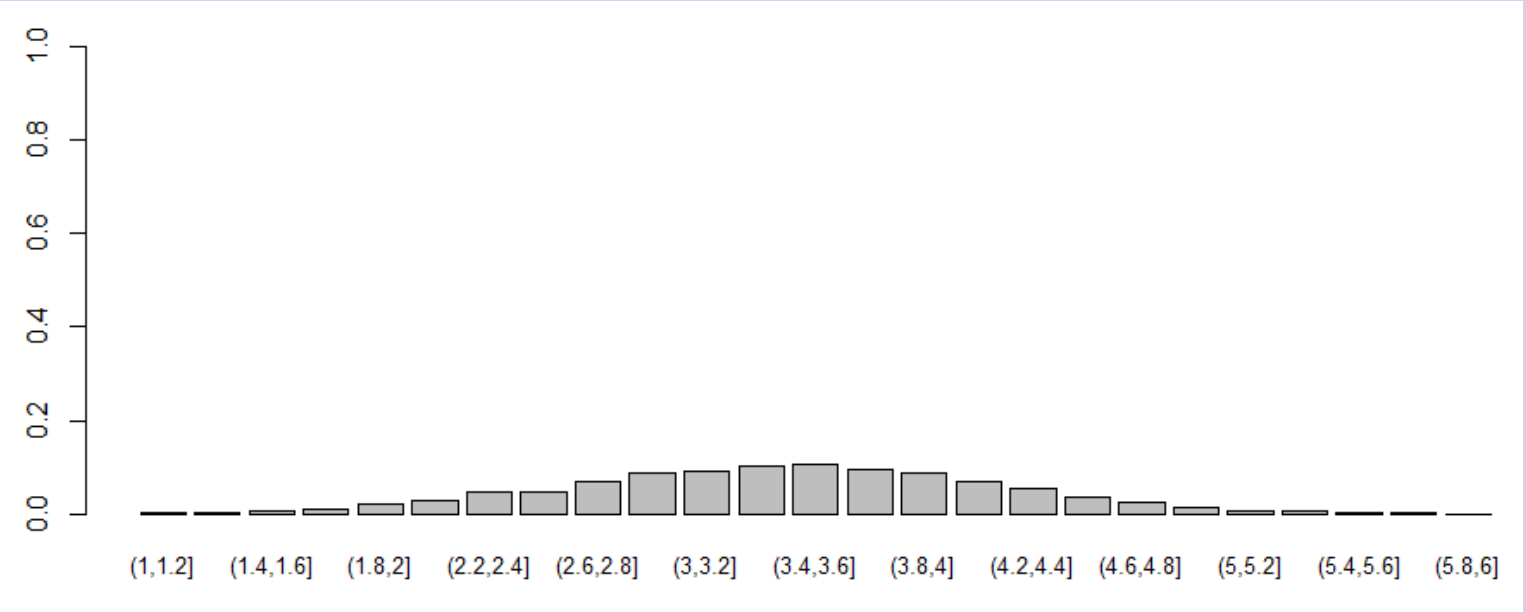
Quindi il valore atteso di $\bar{X}$ è proprio il valore atteso 7/2 nel lancio di un dado (cioè di ciascuna delle X_i) e non dipende da *n*, mentre la varianza di $\bar{X}$ è uguale alla varianza 35/12 nel lancio di un dado (cioè di ciascuna delle X_i) divisa per *n*. Questo è l'aspetto essenziale: la varianza decresce al crescere di *n* grazie al fattore *1/n* e quando *n* tende all'infinito la varianza tende a zero (e quindi anche la dev. standard di $\bar{X}$ tende a zero). Tenete presente che una v.c. con varianza nulla è necessariamente una costante[2]. Ciò ci fa intuire che al crescere di n, la distribuzione di probabilità di $\bar{X}$ debba "concentrarsi" attorno alla media 3,5. Verifichiamo questa intuizione con una simulazione.

[2] Si può dimostrare infatti che la varianza è nulla solamente quando la variabile assume *quasi certamente* un solo valore k, cioè se si ha p(X=k)=1. Nel caso di una v.c. X con un numero finito di valori, come quella di cui ci stiamo occupando, ciò significa che X è una costante.

Distribuzione media campionaria (simulazione).R*

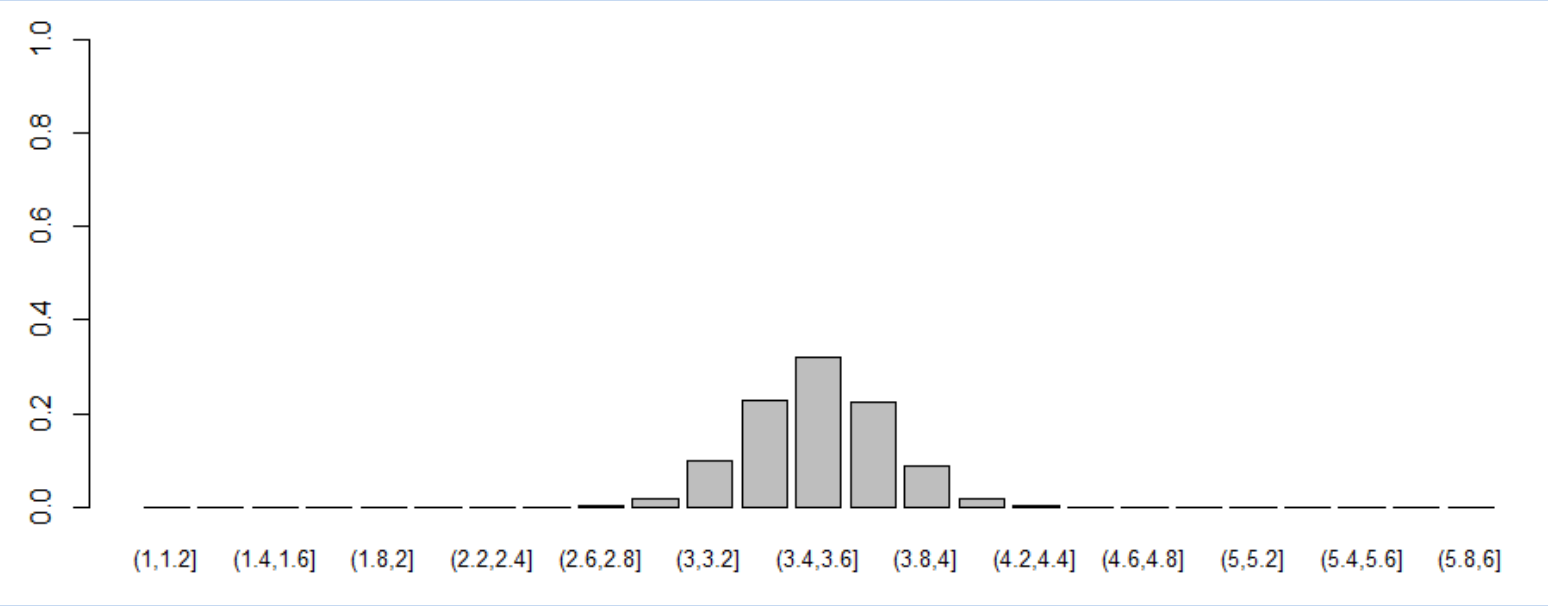
Source on Save | Run | Source

```
cat("\14")
nrepliche=5000
nlanci=5
VettMedieCamp=c()
for (i in 1:nrepliche){
  sequenza=sample(1:6,nlanci,replace=TRUE)
  MediaCamp=sum(sequenza)/nlanci
  VettMedieCamp=c(VettMedieCamp,MediaCamp)
}
classi=cut(VettMedieCamp,breaks=seq(1,6,0.2))
barplot(table(classi)/nrepliche,ylim=c(0,1),cex.names=0.8)
```

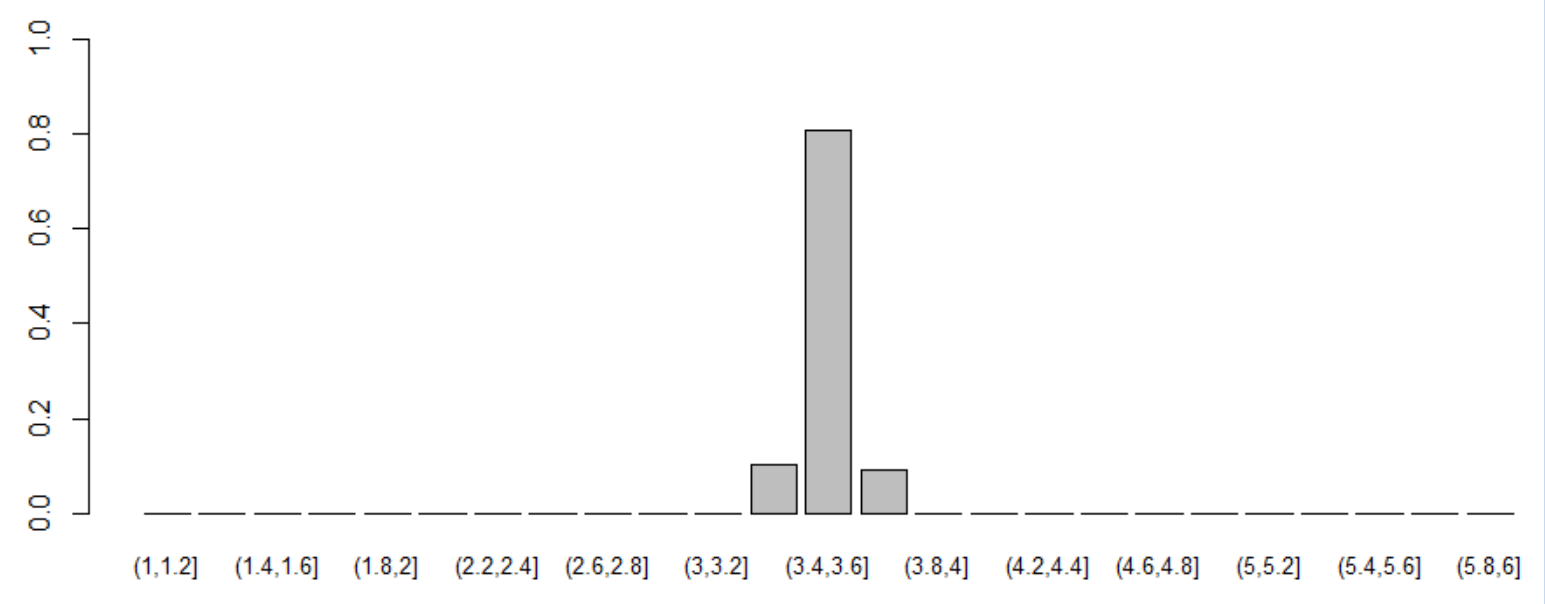


5 lanci

Notare la dispersione della distribuzione di frequenze rel. attorno alla classe (3.4, 3.6].



50 lanci



500 lanci

Nota Quanto visto ha delle implicazioni teoriche importanti. Consideriamo un esperimento casuale e un evento E relativo a tale esperimento; supponiamo che E abbia probabilità *p* di verificarsi. Replichiamo l'esperimento *n* volte, sempre nelle stesse condizioni. Consideriamo le v.c. indipendenti X_i con i=1, 2, …, *n*, così definite:

$$X_i = \begin{cases} 1 \text{ se l'evento E si verifica} \\ \\ 0 \text{ altrimenti} \end{cases}$$

La somma $S_n=X_1+X_2+\ldots+X_n$ è la v.c. che rappresenta il numero di volte che E si verifica su *n* prove e la v.c. $\bar{X}_n=S/n$ rappresenta la frequenza relativa dell'evento E. Ora noi sappiamo, per quanto visto nell'esempio precedente, che il valore atteso di $\bar{X}_n$ è uguale al valore atteso di ciascuna delle X_i, cioè è uguale a $1\cdot p+0\cdot(1-p)=p$, e la varianza di $\bar{X}_n$ è uguale alla varianza di ciascuna delle X_i divisa per *n* (dunque la varianza tende a zero al crescere di n). Tenendo conto di questi due fatti teorici:

1. Il valore atteso della v.c. $\bar{X}_n$ che rappresenta la frequenza relativa dell'evento E è *p*, cioè è proprio la probabilità di E
2. La varianza di $\bar{X}_n$ tende a zero quando il numero di prove tende all'infinito

possiamo avere un'idea di quanto, in modo più preciso, viene affermato dal *teorema debole dei grandi numeri* e cioè che

$$\lim_{n\to\infty} prob(|\bar{X}_n - p| \geq \varepsilon) = 0^3$$

[3] Qui si parla di convergenza in probabilità. Ricordando la nozione di limite propria dell'analisi, possiamo tradurre questa scrittura formale così: per ogni $\varepsilon_1>0$, per ogni $\varepsilon_2>0$ esiste un intero $n(\varepsilon_1, \varepsilon_2)$ tale che

$$prob(|\bar{X}_n - p| \geq \varepsilon_1) < \varepsilon_2$$

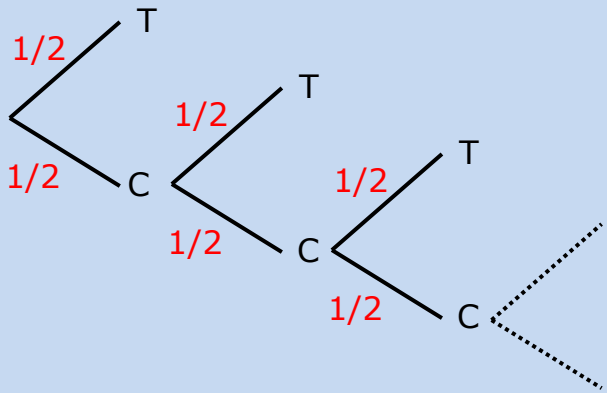
se $n > n(\varepsilon_1, \varepsilon_2)$. Attenzione ciò non implica che, per $n > n(\varepsilon_1, \varepsilon_2)$, sia necessariamente $|\bar{X}_n - p| < \varepsilon_1$ anche se è probabile che sia così.

La distribuzione geometrica

Finora abbiamo avuto a che fare con vari tipi di distribuzioni di probabilità (uniforme, binomiale, ipergeometrica ed anche altre come la distribuzione della somme di due o più dadi). In tutti questi casi le distribuzioni si riferivano a v.c. discrete finite, cioè a variabili che potevano assumere un numero finito di valori. Il prossimo esempio introduce invece una v.c. discreta che può assumere una infinità numerabile di valori cioè i valori possibili costituiscono un sottoinsieme infinito dell'insieme dei numeri interi.

Esempio 1 Si lancia più volte una moneta equa. Qual è la distribuzione di probabilità della variabile casuale X="numero di lanci per ottenere per la prima volta TESTA"?

Quali sono i valori possibili di X? X assume il valore 1 se al primo lancio ottengo TESTA, assume il valore 2 se al primo lancio ottengo CROCE e al secondo TESTA, assume il valore 3 se la sequenza è CCT e così così via; X può assumere qualsiasi valore intero maggiore di 0. Naturalmente è molto improbabile che sia, ad esempio, X=1000; ciò vorrebbe dire che per 999 volte è sempre uscita CROCE. Tuttavia dobbiamo contemplare, in linea di principio, anche casi via via più improbabili. X è dunque una variabile casuale discreta a infiniti valori. Tenendo presente il diagramma qui a fianco, è facile rendersi conto che



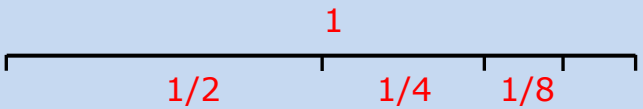
$p(X=1) = 1/2,$
$p(X=2) = (1/2)(1/2) = 1/2^2$
$p(X=3) = (1/2)\,(1/2)\,(1/2) = 1/2^3$
…

Quindi, per i=1, 2, 3, …

$P(X=i) = 1/2^i$

Se sommiamo tutte le probabilità di una distribuzione dobbiamo ottenere 1, ricordate? Nel nostro caso dobbiamo sommare <u>infiniti</u> termini

$1/2 + 1/4 + 1/8 + 1/16 + \ldots$



ma l'intuizione geometrica (vedi figura) ci dice che tale somma deve proprio essere 1. D'altra parte i termini della nostra somma sono in <u>progressione geometrica</u> di ragione r=1/2. La somma $s_n=r+r^2+r^3+\ldots+r^n$ è, come sapete[4],

$$s_n = \frac{r^{n+1}-r}{r-1}$$

Quindi nel nostro caso $s_n = 1 - \frac{1}{2n}$ e, per n che tende all'infinito, s_n tende ad 1.

La distribuzione di probabilità della v.c. X rientra nella classe delle ***distribuzioni geometriche***. Per renderci conto dell'andamento e della "forma" della distribuzione di X, tabuliamo p(X=i) facendo variare *i* da 1 a 12 (in realtà *i* dovrebbe variare da 1 a infinito) e rappresentiamo graficamente la tabella.

[4] Posto $s = r+r^2+r^3+\ldots+r^n$, moltiplicando entrambi i membri per r, si ha $rs = r^2+r^3+\ldots+r^{n+1}$; sottraendo la seconda equazione alla prima si ottiene $s-rs=r-r^{n+1}$ da cui segue $s=\frac{r^{n+1}-r}{r-1}$.

Ecco il codice:

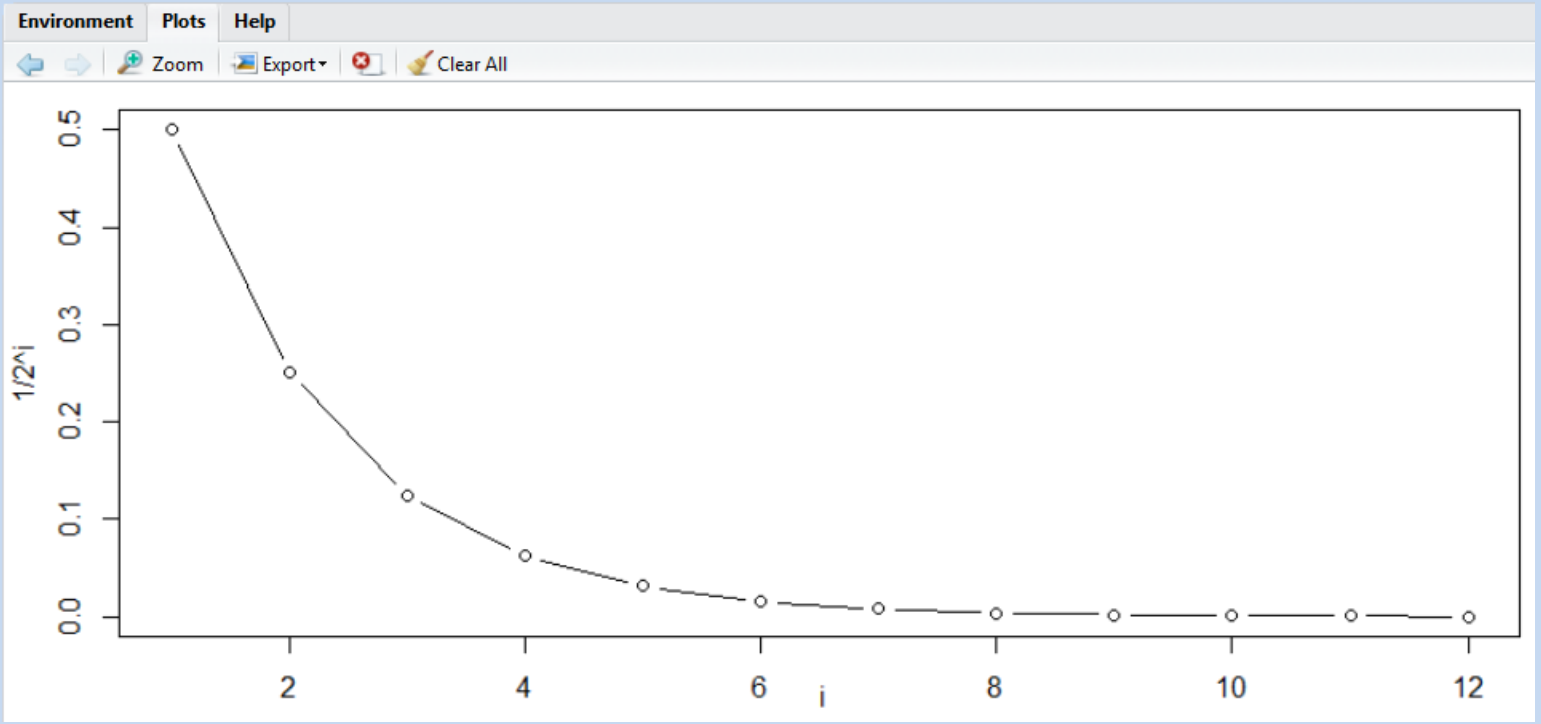
Untitled1*

Source on Save

```
cat("\14")
i=1:12
print(cbind(i,1/2^i))
plot(i,1/2^i,type="b")
```

Console ~/R/

```
      i
 [1,]  1 0.5000000000
 [2,]  2 0.2500000000
 [3,]  3 0.1250000000
 [4,]  4 0.0625000000
 [5,]  5 0.0312500000
 [6,]  6 0.0156250000
 [7,]  7 0.0078125000
 [8,]  8 0.0039062500
 [9,]  9 0.0019531250
[10,] 10 0.0009765625
[11,] 11 0.0004882812
[12,] 12 0.0002441406
>
```

Vediamo ora quali sono, in astratto (quindi in generale), le caratteristiche di una distribuzione geometrica.

Consideriamo un esperimento aleatorio che abbia solo due esiti possibili: successo e insuccesso, rispettivamente con probabilità *p* e *1-p*; replichiamo l'esperimento, sempre nelle stesse condizioni, assumendo cioè prove indipendenti, fino al primo successo. Diremo che la v.c.

X="numero di prove per ottenere il primo successo"

ha ***distribuzione geometrica*** di parametro *p* e scriveremo

$X \sim Geom(p)$

Si dimostra facilmente, con un diagramma ad albero analogo a quello dell'esempio precedente, che la distribuzione di probabilità di X è

$p(X=i) = (1-p)^{i-1}p$ con i=1, 2, …

Si può dimostrare inoltre che il valore atteso di X è

$E(X)=1/p$

e la varianza di X è

$var(X)=(1-p)/p^2$

Esempio 2 Quante volte devo lanciare in media una moneta equa per ottenere TESTA? Verificare il risultato teorico con una simulazione.

La v.c. X="numero di lanci per ottenere per la prima volta TESTA" ha una distribuzione Geom(p) con p=1/2, quindi il suo valore atteso è 1/p=2. Come vedete il valore atteso è proprio quello che ci aspettavamo intuitivamente. Ecco la simulazione:

Numero lanci per TESTA.R*

Source on Save

```
cat("\14")
nrepliche=10
X=rep(0,nrepliche)

for (i in 1:nrepliche){
  cont=0
  repeat {moneta=sample(c("T","C"),1)
          cat(moneta)
          cont=cont+1
          if (moneta=="T") break}
  cat("\n")
  X[i]=cont}
cat("Realizzazioni di X:",X,"\n")
cat("Media=",mean(X))
```

Console ~/R/

```
CCCT
T
T
CCCCCT
T
T
CT
T
T
T
Realizzazioni di X: 4 1 1 6 1 1 2 1 1 1
Media= 1.9
>
```

Nota Qui abbiamo utilizzato la struttura di controllo

```
repeat {  comando
          comando
          ...
          if (condizione) break}
```


che ci consente di eseguire ripetutamente un blocco di comandi fino a quando si verifica una certa condizione (quando la condizione è verificata si esce dal ciclo). A differenza del ciclo *for*, il numero di ripetizioni non è predeterminato.

Esempio 3 Un urna contiene 40 dischetti numerati da 1 a 40. Qual è la probabilità di ottenere il numero cinque (almeno una volta) in 10 estrazioni con reimmissione? Verificare il risultato teorico con una simulazione.

Il numero cinque può presentarsi per la prima volta alla prima estrazione, oppure alla seconda, alla terza e così via … (tenendo presente che le estrazioni sono con reimmissione). La v.c. X="numero di estrazioni per ottenere per la prima volta il cinque" ha distribuzione Geom(1/40) e si ha $p(X=i)=(1-1/40)^{i-1}1/40$. Quindi la probabilità cercata è

$$p(X=1)+p(X=2)+\ldots+p(X=10)$$

(il calcolo lo faremo con R). Ecco il codice per il calcolo della probabilità e per la simulazione:

temp.R* ×

Source on Save | Run

```
cat("\14")

p=function(i) (1-1/40)^(i-1)*(1/40)
prob=sum(p(1:10))
cat("Probabilità=",prob,"\n\n")

nrepliche=100000
urna=1:40
cont=0
for (i in 1:nrepliche){
  sequenza=sample(urna,10,replace=T)
  if (5 %in% sequenza) cont=cont+1}
freq_rel=cont/nrepliche
cat("Frequenza relativa estrazione
del 5 in 10 estrazioni=",freq_rel)
```

Console ~/R/

```
Probabilità= 0.2236704

Frequenza relativa estrazione
del 5 in 10 estrazioni= 0.22287
>
```


Nota Qui abbiamo utilizzato l'operatore *%in%* che ci consente di verificare se un certo valore è elemento di un vettore; ecco un esempio:

Console ~/R/

```
> x=1:5
> 3 %in% x
[1] TRUE
> 6 %in% x
[1] FALSE
>
```

Nota Lo stesso problema poteva essere risolto utilizzando la distribuzione binomiale. Possiamo infatti considerare la variabile casuale

X=”numero di volte che si presenta il cinque in 10 estrazioni”

Tenendo presente che le successive estrazioni del dischetto sono prove indipendenti con probabilità di successo p=1/40 (data la reimmissione), la distribuzione di probabilità di X è $X \sim B(10, 1/40)$. La probabilità cercata sarà quindi

$p(X=1)+p(X=2)+ \ldots + p(X=10)$

Infatti il cinque, in 10 estrazioni, può presentarsi esattamente una volta oppure esattamente due volte oppure … oppure esattamente dieci volte. In tutti questi casi e solo in questi casi si presenta almeno una volta.

Facciamo il calcolo con R:

```
Console ~/R/
> sum(dbinom(1:10,10,1/40))
[1] 0.2236704
>
```

Esempio 4 Quante volte bisogna lanciare in media un dado equo per ottenere tutte le facce?

Sia X = "numero di lanci per ottenere tutte le facce". Esprimiamo X come somma delle variabili casuali:

X_1 = 1 (numero di lanci per il primo valore, ovviamente basta un lancio)
X_2 = "numero addizionale di lanci fino al secondo nuovo valore"
…
X_6 = "numero addizionale di lanci fino al sesto nuovo valore"

$X = X_1 + \ldots + X_6$

Ora la v.c. X_2 ha una distribuzione geometrica Geom(5/6), X_3 ha la distribuzione Geom(4/6) e così via. Allora, grazie alla linearità del valore atteso,

$E(X) = E(\Sigma X_i) = \Sigma E(X_i) = 1 + 6/5 + 6/4 + 6/3 + 6/2 + 6 = 147/10 = 14,7$

Ecco il codice per la simulazione:

Quanti lanci per tutte le facce di un dado.R

Source on Save

```
cat("\14")
nrepliche=10
X=c()
for (i in 1:nrepliche){
  indicatore=rep(0,6)
  cont=0
  repeat {dado=sample(1:6,1)
          cat(dado)
          indicatore[dado]=1
          cont=cont+1
          if (identical(indicatore,rep(1,6)))
            {cat("\n")
            X=c(X,cont)
            break}}}
cat("\n")
cat("X=",X,"\n")
cat("media(X)=",mean(X),"\n")
```

Output:

Console ~/R/

```
543216
3254251251543423 46
612162664262455426453
55523334635631
136511636542
34234235621
45632523636421
2264364226161625
16212611134415
55552166351264

X= 6 18 21 14 12 11 14 16 14 14
media(X)= 14
>
```


Nota Ogni esperimento consiste nel lancio ripetuto di un dado (comando *repeat*) fino a quando non si presentano tutti i valori possibili. Per accertarsi che siano usciti tutti i valori si utilizza il vettore *indicatore* che all'inizio è 0, 0, 0, 0, 0, 0; ad ogni lancio, se si presenta un certo valore *dado*, si pone la relativa componente *dado* del vettore uguale a 1. Se, ad esempio, al primo lancio si presenta 3, il vettore *indicatore* sarà 0, 0, 1, 0, 0, 0. Si esce dal ciclo *repeat* quando *indicatore* è uguale a 1, 1, 1, 1, 1, 1. Il vettore X memorizza, per ogni esperimento, il numero di lanci effettuati e quindi rappresenta delle realizzazioni della nostra variabile casuale X.

La distribuzione di Poisson

Esempio 1 So, per esperienza, che ricevo in media una e-mail ogni due ore. Qual è la probabilità di ricevere al più 10 e-mail nelle prossime 24 ore?

A prima vista o meglio tenendo conto delle cose che già sappiamo un problema di questo tipo non ha soluzione. Posso dire che in media riceverò 12 mail al giorno, ma non conosco la distribuzione di probabilità della variabile casuale

X = ”numero di mail ricevute in 24 ore”

Dalla conoscenza del valore atteso λ=E(X) di X, nel nostro caso uguale a 12, non posso in generale risalire alla distribuzione di probabilità di X. Nel nostro caso però entra in gioco il **modello di Poisson** e dalla sola conoscenza del valor medio λ possiamo risalire alla distribuzione di probabilità di X che sarà

$$p(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}$$

Quindi nel nostro caso la probabilità cercata è

p(X=0) + p(X=1) + … + p(X=10) ≈ 0,347

(il calcolo si fa con R).

Ma quando possiamo applicare il modello (o paradigma) di Poisson? Quali proprietà caratterizzano tale modello?

Il significato della distribuzione di Poisson è questo: si vogliono contare i "successi" su un grande numero di eventi che possono avere due esiti possibili, essendo la probabilità di successo per ogni evento molto piccola (e in generale distinta per ogni evento). Ad esempio possiamo modellizzare con la distribuzione di Poisson il numero X di e-mail ricevuto nell'arco di un'ora (eventi "nel tempo"). Le persone che potenzialmente possono inviarci una mail sono moltissime ma l'evento che una ben determinata persona ci mandi una mail proprio in quell'arco di tempo è molto piccola. Un altro esempio (eventi "nello spazio"): X è il numero di errori di stampa presenti in una certa pagina di un certo libro. Anche qui gli errori potenzialmente possibili sono moltissimi, uno per ogni parola, ma la probabilità che una certa parola contenga un errore di stampa è molto piccola.

Caratteristiche del modello di Poisson

Siano E_1, E_2, … , E_n degli eventi e sia $p(E_i)=p_i$, assumiamo che *n* sia grande e ogni p_i piccola, assumiamo inoltre che gli E_i siano indipendenti (o debolmente dipendenti). In queste ipotesi il numero X di eventi che si verificano ha una distribuzione di probabilità **approssimativamente** uguale a

$$p(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}$$

dove λ è il valore atteso di X. Notare che la v.c. X è discreta a infiniti valori infatti nel modello astratto k può assumere tutti i valori 0, 1, 2, … (anche se nella realtà applicativa non sarà mai così, ovviamente). Si può inoltre dimostrare che anche la varianza di X è uguale a λ.

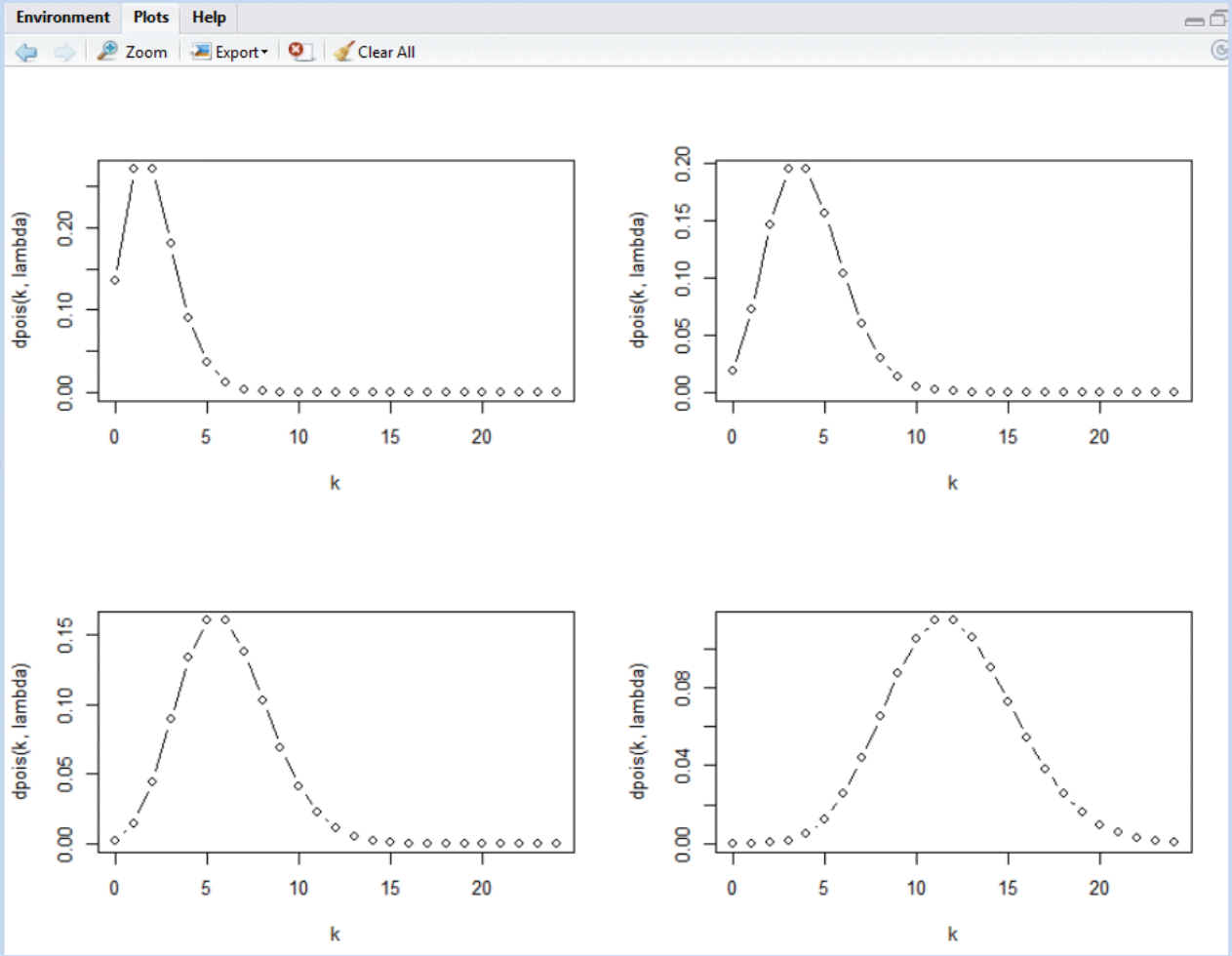
Forma della distribuzione

Tracciamo i grafici per punti delle funzioni di probabilità relative al parametro λ=2, λ=4, λ=6, λ=12 (in ascissa k, in ordinata p(X=k). Il comando di R da usare è *dpois(k, λ)* che ci fornisce p(X=k).

Untitled1* ×

Source on Save

```
k=0:24

par(mfrow=c(2,2))
lambda=2
plot(k,dpois(k,lambda),type="b")
lambda=4
plot(k,dpois(k,lambda),type="b")
lambda=6
plot(k,dpois(k,lambda),type="b")
lambda=12
plot(k,dpois(k,lambda),type="b")
```



Esempio 2 Vogliamo costruire artificialmente una situazione simile a quella descritta nell'esempio 1 e verificare, con una simulazione, che la distribuzione di Poisson la rappresenta con buona approssimazione.

Supponiamo che ci siano 100 fonti che possono inviarci una mail comprendendo lo spam; introduciamo 100 variabili casuali F_i che valgono 1 se la fonte i-esima ci invia una mail nell'arco delle prossime 24 ore e 0 altrimenti. Assumiamo inoltre che sia

$$p(F_i = 1) = \frac{1}{20+i} \quad i = 1, 2, \dots, 100$$

(quindi le probabilità di successo, cioè di invio di una mail, cambiano da fonte a fonte e sono abbastanza piccole variando da un massimo di 1/21 a un minimo di 1/120).
La variabile casuale X che ci interessa è

$$X = \text{"numero di mail ricevute nelle prossime 24 ore"} = \sum_{i=1}^{100} F_i$$

Il valore atteso di X è $\lambda = E(X) = \sum_{i=1}^{100} \frac{1}{20+i}$, calcoliamolo con l'aiuto di R:

```
Console ~/R/
> i=1:100
> lambda=sum(1/(20+i))
> lambda
[1] 1.771129
>
```

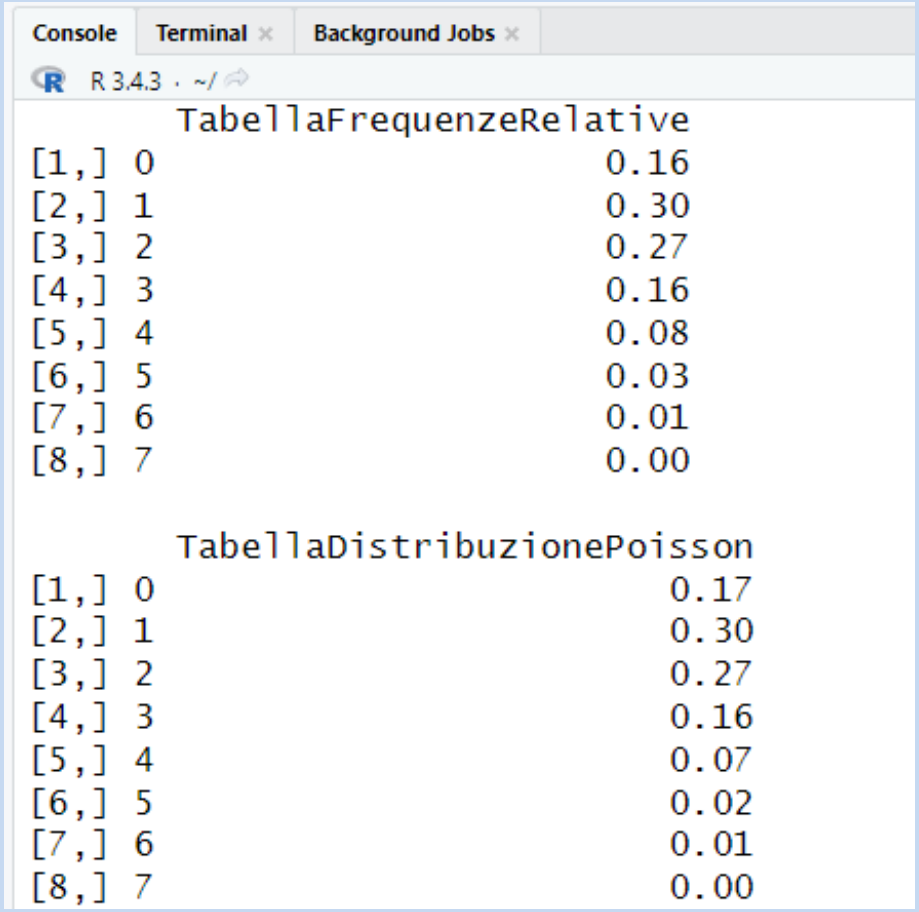
Quindi λ è circa 1,771.

Ecco il codice della simulazione:

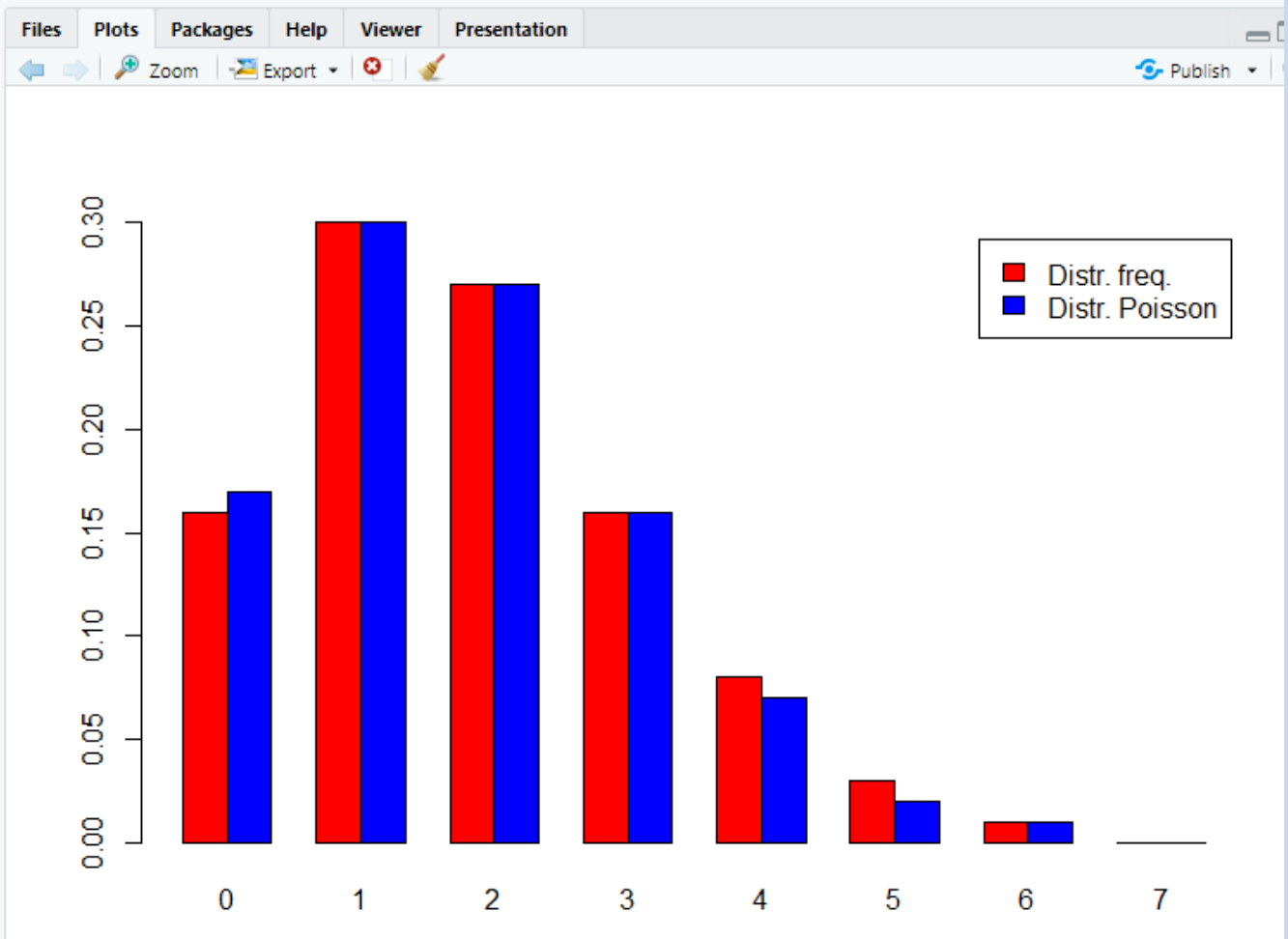
poisson.R

```
dati=c()
k=1:100
lambda=sum(1/(20 + k))

nrepliche=1000
for (n in 1:nrepliche) {
  num_mail=0
  for (i in 1:100) {
       p=1/(20+i)
       j=sample(0:1,size=1,prob=c(1-p,p))
       num_mail=num_mail+j}
  dati=c(dati,num_mail)}

tabella_freq_relative=round(table(dati)/nrepliche,2)
l=length(tabella_freq_relative)
TabellaFrequenzeRelative=as.vector(tabella_freq_relative)
print(cbind(0:(l-1), TabellaFrequenzeRelative))
cat("\n")

TabellaDistribuzionePoisson=round(dpois(0:(l-1),lambda),2)
print(cbind(0:(l-1), TabellaDistribuzionePoisson))

#grafico a barre
v=c()
for (i in 1:l)
  {v=c(v,TabellaFrequenzeRelative[i],TabellaDistribuzionePoisson[i])}
M=matrix(v,nrow=2)
barplot(M,beside=TRUE,names.arg = 0:(l-1),col=c("red","blue"),
        legend.text = c("Distr. freq.","Distr. Poisson"))
```

Output testuale:



Output grafico:



Nota: per generare il diagramma a barre in cui compaiono affiancate le due colonnine relative alla distribuzione di frequenze relative e alla distribuzione di probabilità, bisogna costruire una matrice M di due righe le cui colonne abbiano per elementi frequenza relativa e corrispondente probabilità.

Il teorema di Bayes

Dall'uguaglianza

$$\mathrm{p}(\mathrm{A}\cap\mathrm{B}) = \mathrm{p}(\mathrm{A}) \cdot \mathrm{p}(\mathrm{B}|\mathrm{A}) = \mathrm{p}(\mathrm{B}) \cdot \mathrm{p}(\mathrm{A}|\mathrm{B})$$

segue, se $p(B) \neq 0$,

$$\mathrm{p}(\mathrm{A}|\mathrm{B}) = \frac{\mathrm{p}(\mathrm{A})\cdot\mathrm{p}(\mathrm{B}|\mathrm{A})}{\mathrm{p}(\mathrm{B})}$$

e questo è il teorema, o meglio la formula, di Bayes.

Per il momento la formula non è ancora molto significativa: conviene dare un'interpretazione "nel tempo" agli eventi A e B (interpretazione diacronica). Ragioniamo su una situazione concreta.

Esempio 1 (La moneta truccata) Un'urna contiene 10 monete indistinguibili, 9 sono regolari e 1 è truccata (è bilanciata in modo che si presenti sempre TESTA). Si estrae una moneta dall'urna e si esegue l'esperimento consistente nel lanciarla 6 volte; il risultato è che si presenta per tutte le 6 volte TESTA. Cosa possiamo dire, dopo l'esperimento, sulla probabilità che la moneta estratta sia quella truccata?

Osserviamo che prima dell'esecuzione dell'esperimento, l'informazione a disposizione ci porta a dire che la probabilità che la moneta sia truccata è banalmente uguale a 1/10 (**probabilità a priori**). L'esecuzione dell'esperimento modifica però lo stato delle nostre informazioni e, intuitivamente, siamo spinti a ritenere che la probabilità che la moneta estratta sia quella truccata è aumentata, è ora maggiore di 1/10 (**probabilità a posteriori**, dopo l'esperimento). Si capisce che qui entra in gioco la probabilità condizionata. Come possiamo procedere per una valutazione numerica della probabilità?

Consideriamo i due eventi:

H = "la moneta estratta è truccata"
E = "lanciando 6 volte la moneta estratta si presenta TESTA per 6 volte"

Abbiamo indicato i due eventi rispettivamente con **H**, che sta per **IPOTESI** ed **E**, che sta per **ESPERIMENTO**.

Allora la probabilità che noi cerchiamo è

p(H|E) (probabilità a posteriori, la probabilità di H dato l'evento E)

Per la formula di Bayes:

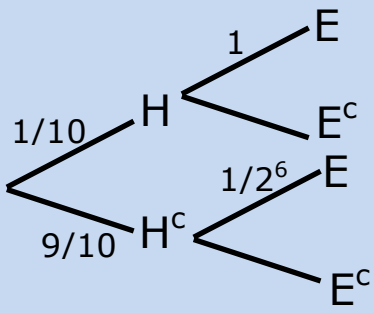
$$p(H|E) = \frac{p(H)\cdot p(E|H)}{p(E)}$$

Dove:

p(H) è la probabilità dell'ipotesi senza alcuna condizione, cioè prima di eseguire l'esperimento (probabilità a priori); quindi p(H)=1/10

p(E|H) è la probabilità di E data l'ipotesi, assumendo cioè che la moneta sia truccata, quindi p(E|H)=1

p(E) è la probabilità dell'evento E senza alcuna condizione, quindi devo tener conto di entrambi i casi, che si sia o non si sia verificato l'evento H (vedi diagramma di disintegrazione); quindi



$$p(E) = (1/10)\cdot 1 + (9/10)\cdot(1/2^6)$$

In conclusione:

$$p(H|E) = \frac{\frac{1}{10}\cdot 1}{\left(\frac{1}{10}\right)\cdot 1 + \left(\frac{9}{10}\right)\cdot\left(\frac{1}{2^6}\right)} = \frac{64}{73} \cong 0{,}877$$

Simulazione con R

Vogliamo verificare il risultato precedente con una simulazione. La logica della simulazione è questa:

1) si ripete per n volte l'estrazione dall'urna, la probabilità di estrarre la moneta truccata è 1/10 (in realtà si estrarrà un valore a caso del vettore 1:10, assumendo che il valore 1 corrisponda alla moneta truccata);

2) se la moneta estratta è quella truccata si aggiorna il contatore *contF* e si genera la stringa TTTTTT (perché lanciando la moneta truccata si presenta sempre TESTA); se la moneta estratta non è truccata si simula per sei volte il lancio di una moneta equa (probabilità di TESTA uguale a 1/2), si genera la stringa relativa e si tiene conto dei casi in cui si presenta la stringa TTTTTT (contatore *cont*);

3) si considera il rapporto $\frac{contF}{cont+contF}$ che approssima la probabilità cercata; qui è essenziale capire che il rapporto da valutare non è tra numero di volte che si è estratta la moneta truccata e numero n delle prove eseguite ma tra numero di volte che si è estratta una moneta truccata e numero totale di volte che si è presentata la stringa TTTTTT sia essa generata dalla moneta truccata o dalla moneta equa.

Bayes (1).R

Source on Save | Run | Source

```
cat("\14")
SeiVolteT=rep("T",6)

n=1000
cont=0
contF=0
for (i in 1:n)
  {s=sample(1:10,size=1,replace=TRUE)
  if (s==1)
    {cat("Moneta truccata: TTTTTT"); contF=contF+1}
  else
    {cat("Moneta equa:     ")
    s=sample(c("C","T"),size=6,replace=TRUE)
    cat(s,sep="")
    if (identical(s,SeiVolteT)) {cont=cont+1; cat("*")}
  }
  cat("\n")
  }
cat("\n")
cat("Frequenza di H dato E: ")
cat(contF/(contF+cont))
```

Console ~/R/

```
Moneta equa:      CTTCTT
Moneta equa:      CCTCCC
Moneta equa:      TTCTCC
Moneta equa:      CTTCTT
Moneta equa:      TCTCTT
Moneta equa:      TCTTTC
Moneta equa:      TCCTCC
Moneta equa:      TCTTCC
Moneta equa:      TCCTCT
Moneta equa:      CTCTTC
Moneta equa:      TTCCCC
Moneta equa:      TCCCCC
Moneta equa:      TTTTTT*
Moneta equa:      TTCCCT
Moneta equa:      TTCCCC
Moneta truccata: TTTTTT
Moneta equa:      TCCCCC
Moneta equa:      TCTCTC
Moneta equa:      TTCCTT
Moneta equa:      TCTTCC

Frequenza di H dato E: 0.8181818
```

Esempio 2 (Il problema dei biscotti) Ci sono due scatole di biscotti, la prima, la scatola A, contiene 10 biscotti al cioccolato e 30 biscotti alla vaniglia, la seconda, la scatola B, contiene 20 biscotti al cioccolato e 20 alla vaniglia. Paolo sceglie una scatola a caso e da questa prende un biscotto a caso. Se il biscotto che Paolo ha preso è alla vaniglia, qual è la probabilità che sia stato preso dalla scatola A?

Ipotesi:	H = ”Il biscotto proviene dalla scatola A”
Informazione (evidenza):	E = “Il biscotto estratto è alla vaniglia”
Cosa voglio trovare:	$p(H \mid E)$
Probabilità a priori dell’ipotesi:	$p(H)=1/2$
Probabilità di E data l’ipotesi:	$p(E \mid H) = 3/4$
Teorema di Bayes:	$p(H \mid E) = p(E \mid H)p(H)/p(E)$
Formula di disintegrazione:	$p(E) = p(E \mid H)p(H)+p(E \mid H^c)p(H^c) = 3/8+1/4 = 5/8$

Quindi: $p(H \mid E) = \frac{\frac{3}{4}\cdot\frac{1}{2}}{\frac{5}{8}} = \frac{3}{5}$

Formula di disintegrazione

$$p(E) = p(E \mid H)p(H)+p(E \mid H^c)p(H^c)$$