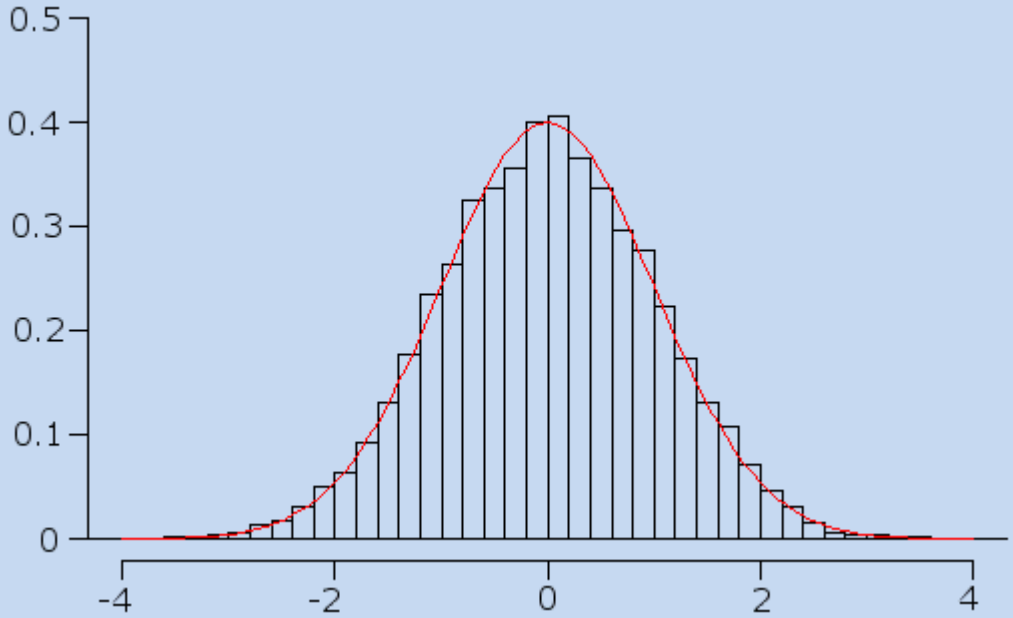



# Calcolo delle probabilità con il linguaggio R

## (con un'introduzione al linguaggio)

## Prima parte

©2015 Paolo Lazzarini
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 =

| 1 | 2 | 3 | 2 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|---|
| 1° elem. | 2° elem. | 3° elem. | 4° elem. | 5° elem. | 6° elem. | 7° elem. | 8° elem. | 9° elem. |

lunghezza(u)=9

x =

| 1 | 2 | 3 |
|---|---|---|

y =

| 2 | 3 | 1 |
|---|---|---|

x è diverso da y

r =

| A | A | B |
|---|---|---|

s =

| A | B |
|---|---|

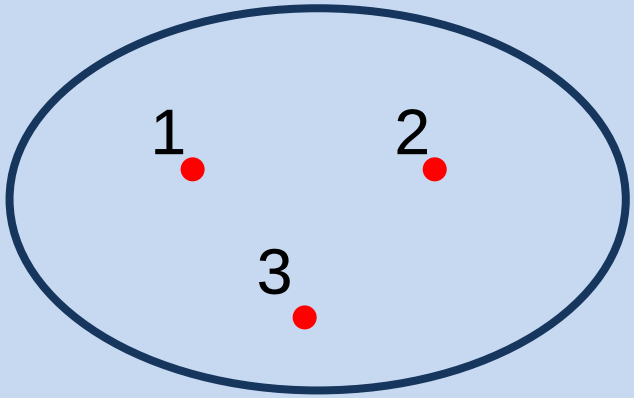
r è diverso da s

# 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
```

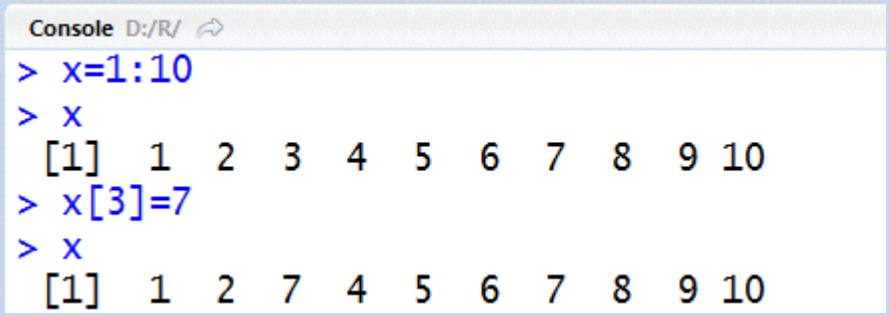
```
Console D:/R/
> x[x>3] #estrae gli elementi maggiori di 3
[1] 5 4 7 9 5 6 6
> length(x[x>3]) #quanti sono?
[1] 7
```

```
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:



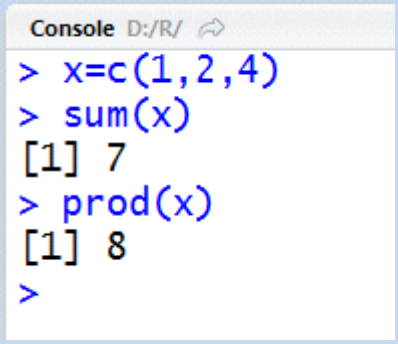

Qui viene cambiato il valore dell'elemento di x al posto 3; si può anche creare un elemento non presente, ad esempio x[11]=1.

**Esempio 2** Eliminare un dato elemento:

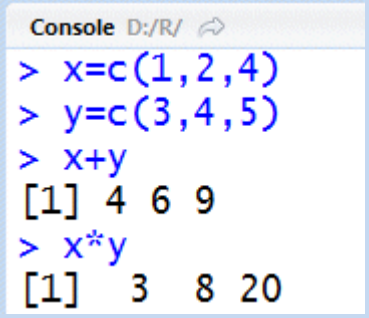

Qui viene eliminato l'elemento di x al posto 3 (notare la riassegnazione).

# Somme e prodotti

**Esempio 1** Somma e prodotto degli elementi di un vettore:

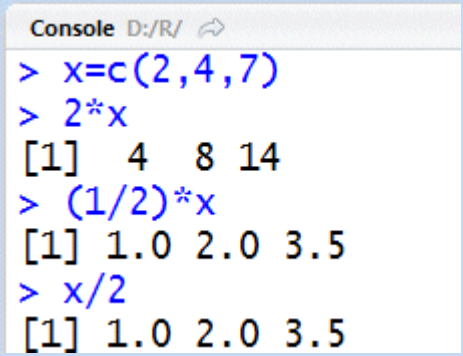



**Esempio 2** Somma e prodotto di vettori:



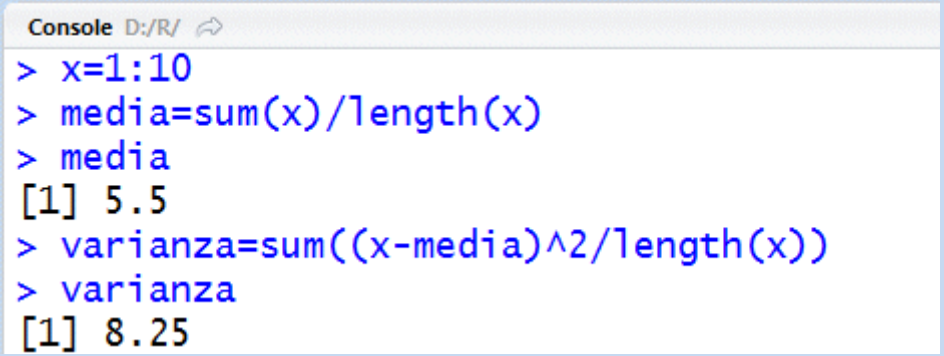

Attenzione, anche il prodotto è elemento per elemento; per il prodotto scalare si usa un altro simbolo.

**Esempio 3** Prodotto di un numero per un vettore:



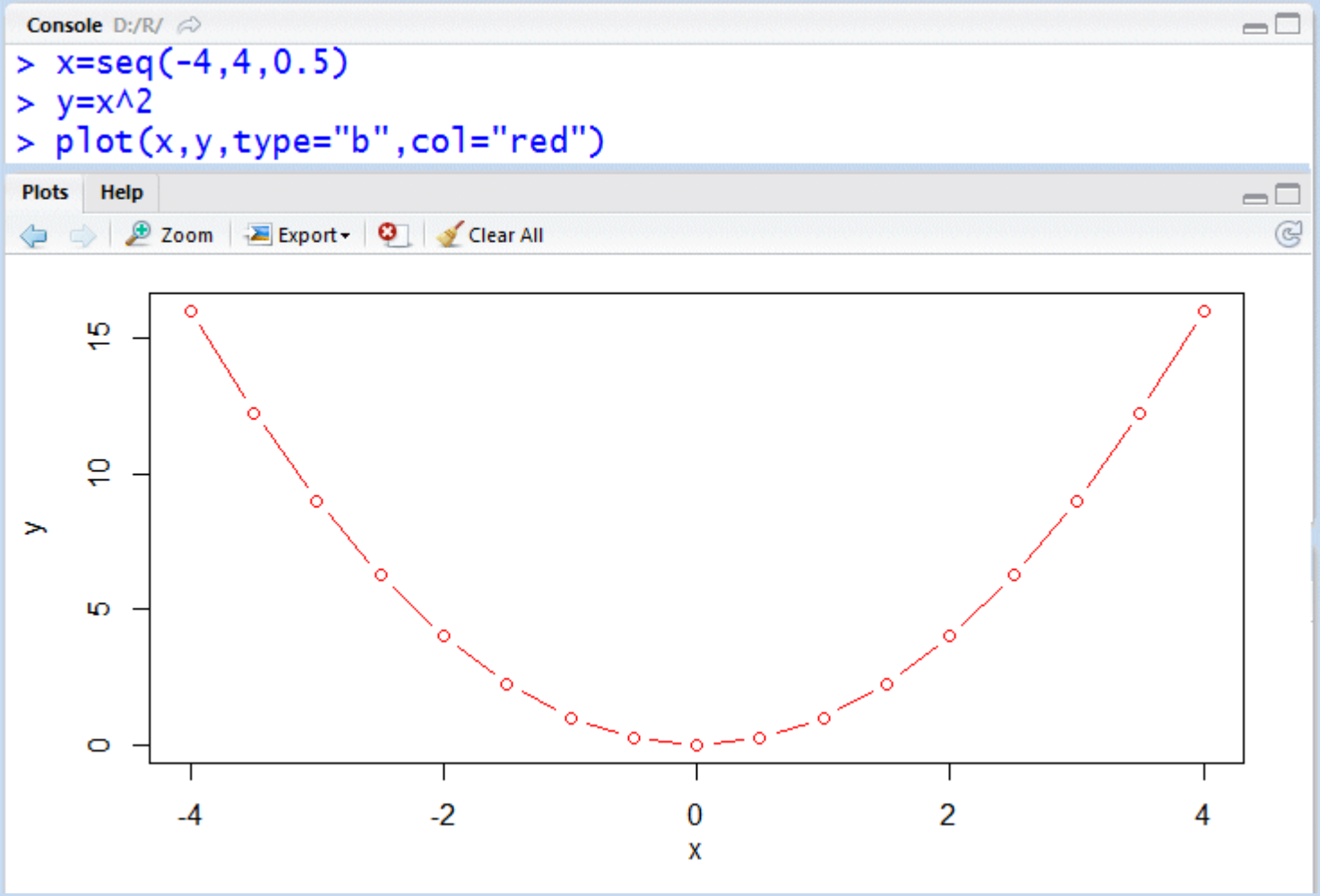



**Esempio 4** Media e varianza degli elementi di un vettore:




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

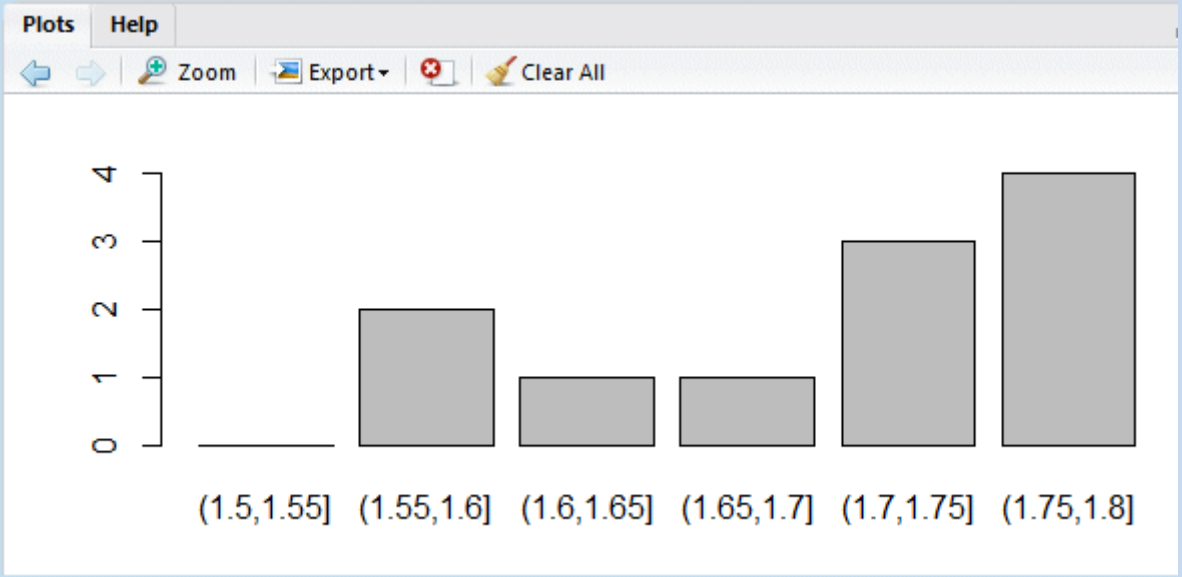
Per tabulare le frequenze di classe utilizzeremo le funzioni *cut()* e *table()*

**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:

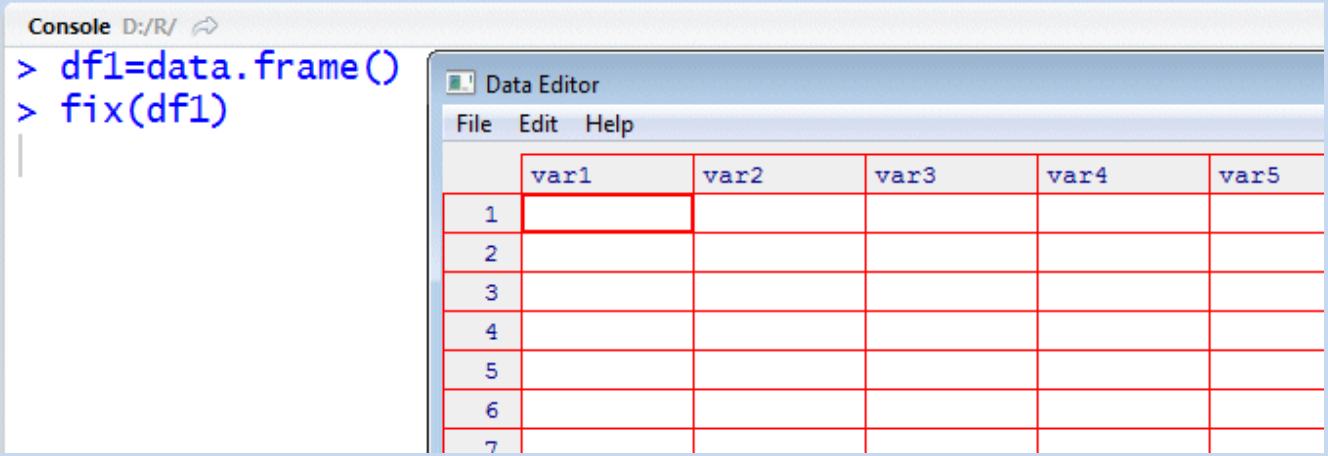
```
Console D:/R/
> subset(df,subset=(altezza>154 & sesso=="F"))
   nome altezza peso sesso
3 Maria     158   48      F
```

**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

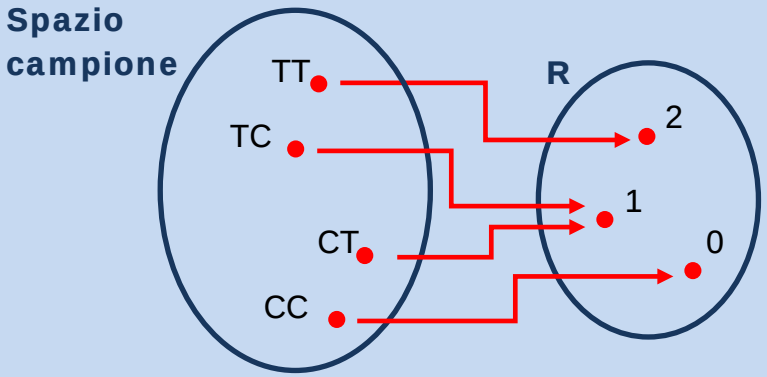
| **M** | 0 | 1 |
|---|---|---|
| **prob.** | 1/2 | 1/2 |

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:

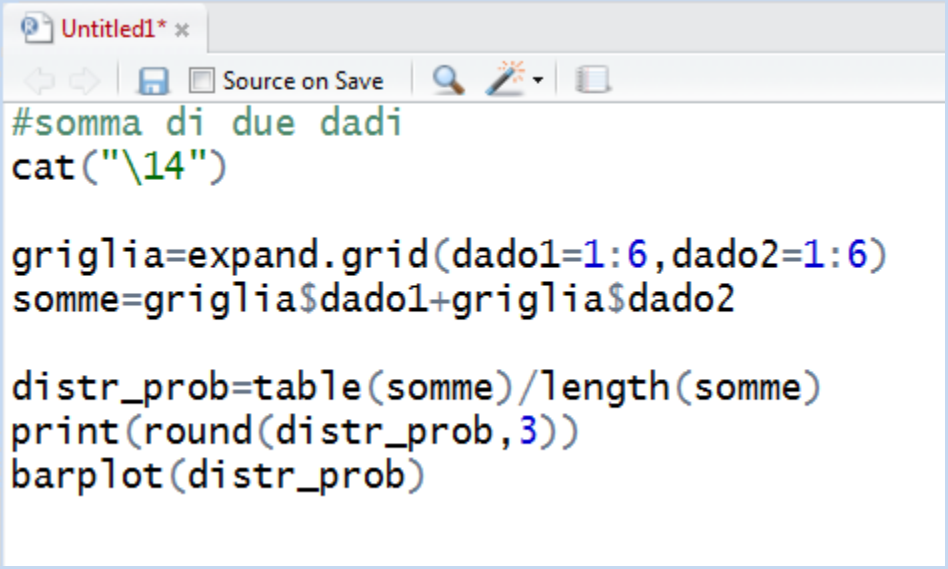
$$p(X=0) + p(X=1) + p(X=2) = 1/4 + 1/2 + 1/4 = 1$$

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:





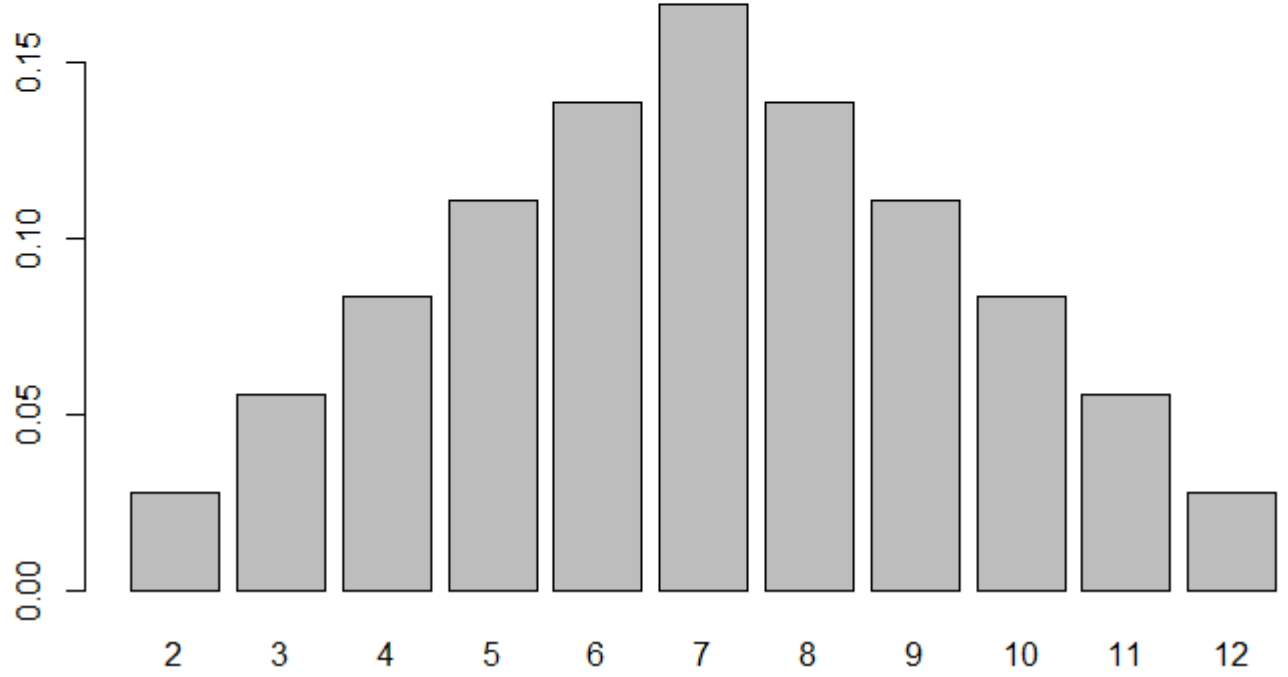

E questo è l’output:

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:

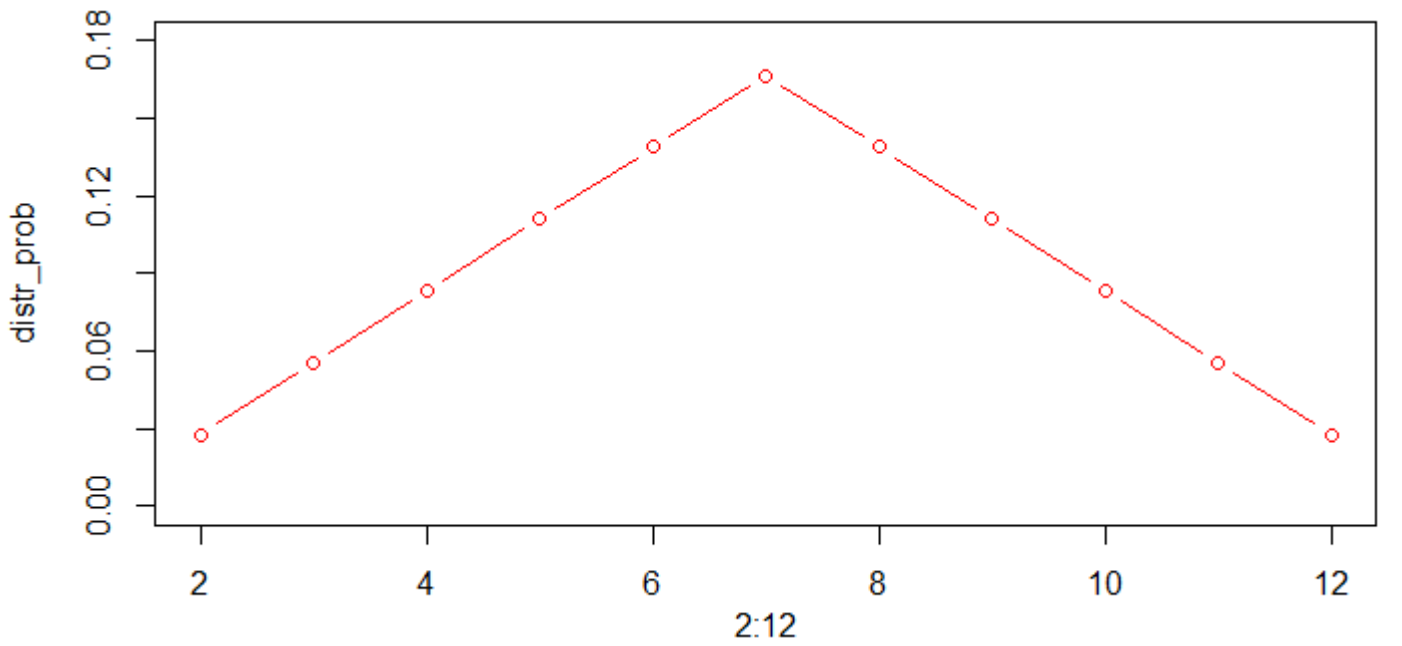
Console ~/R/

```
> 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():

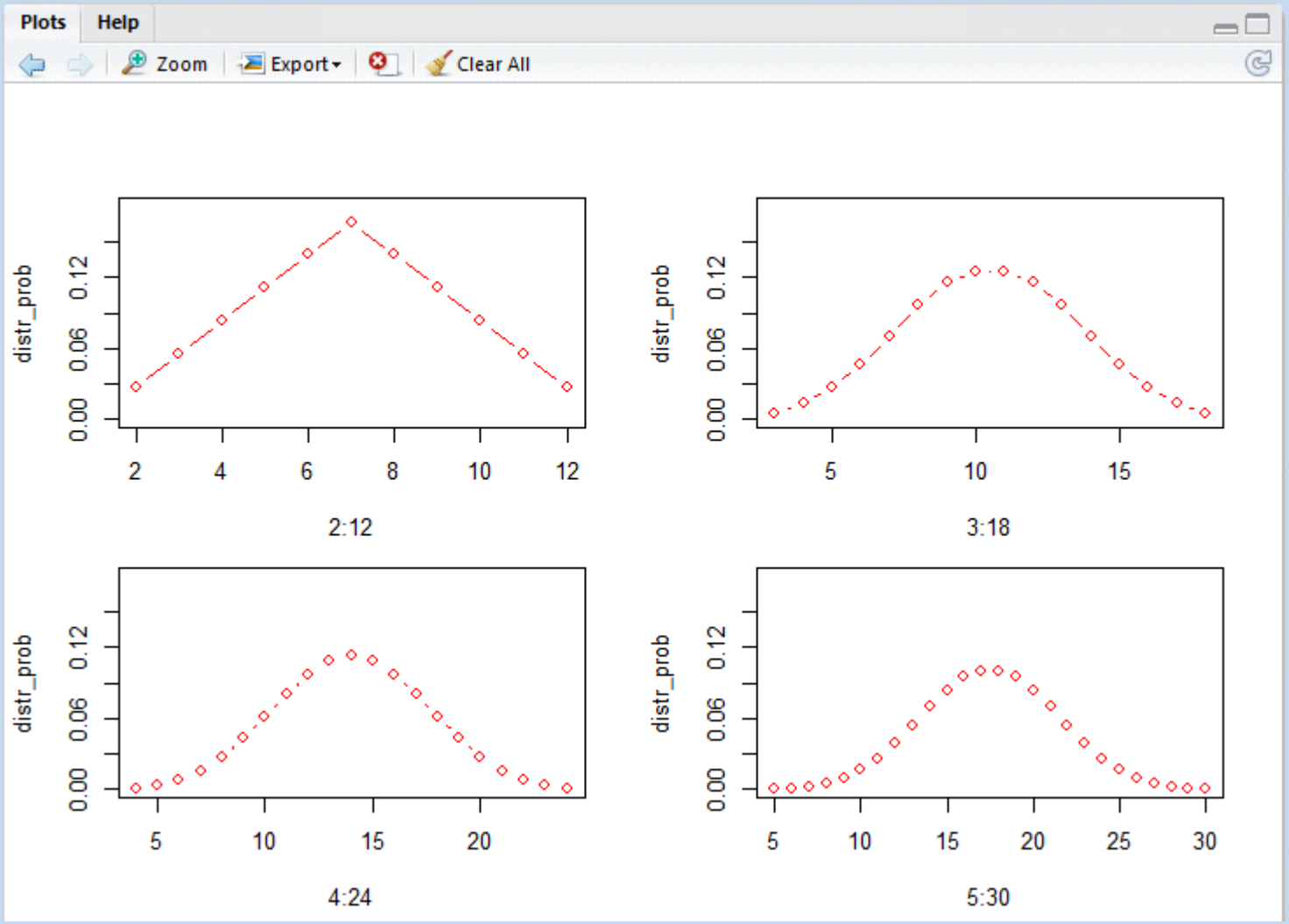
```
Console D:/R/
> plot(2:12,distr_prob,type="b",col="red",ylim=c(0,0.18))
> axis(2,at=seq(0,0.18,0.03))
```




**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):

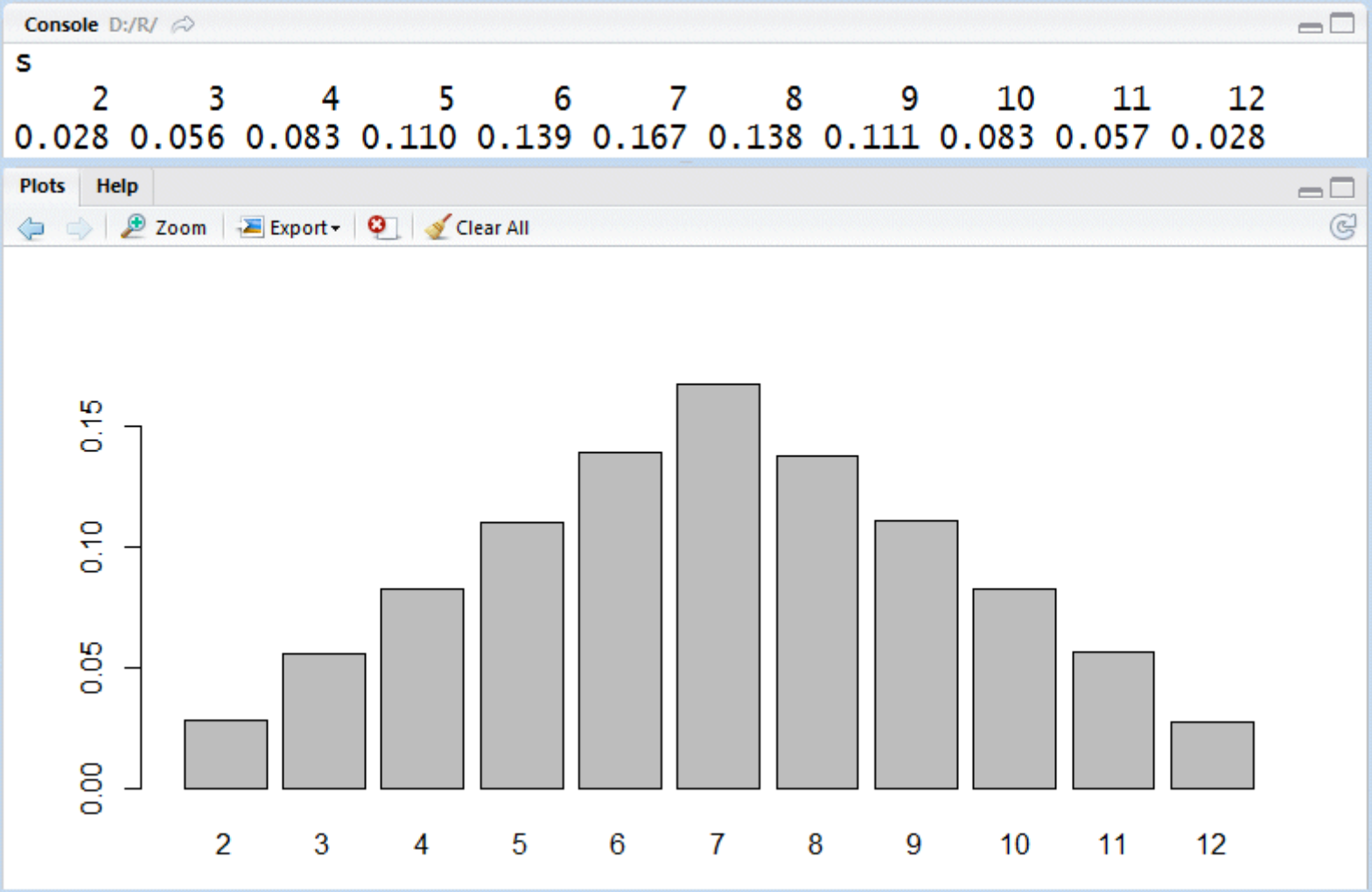


somma_due_dadi_simulazione.R

Source on Save

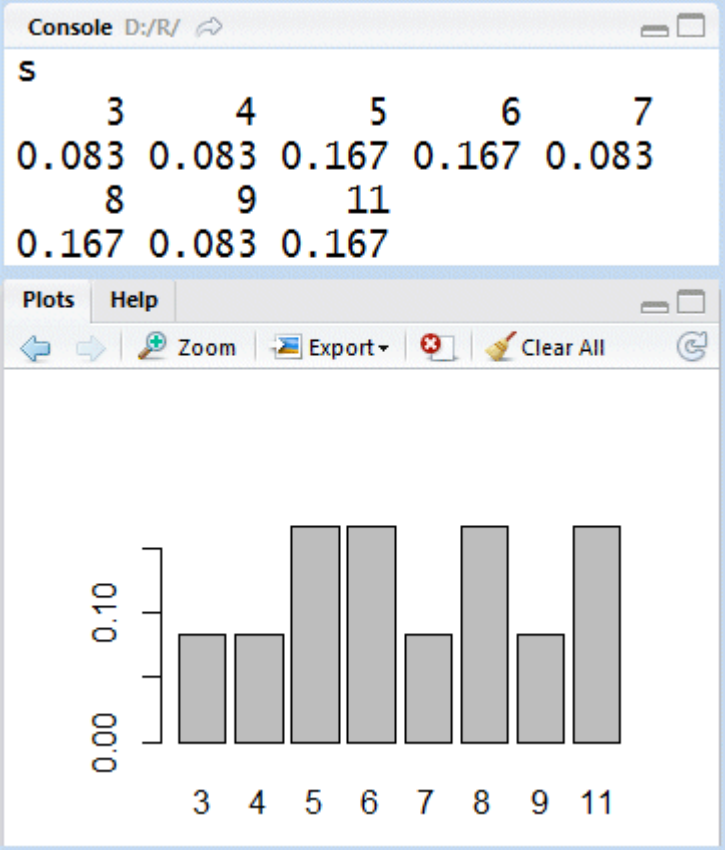
```
#somma di due dadi (simulazione)
cat("\14")
nrepliche=100000

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:




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 campiono, 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

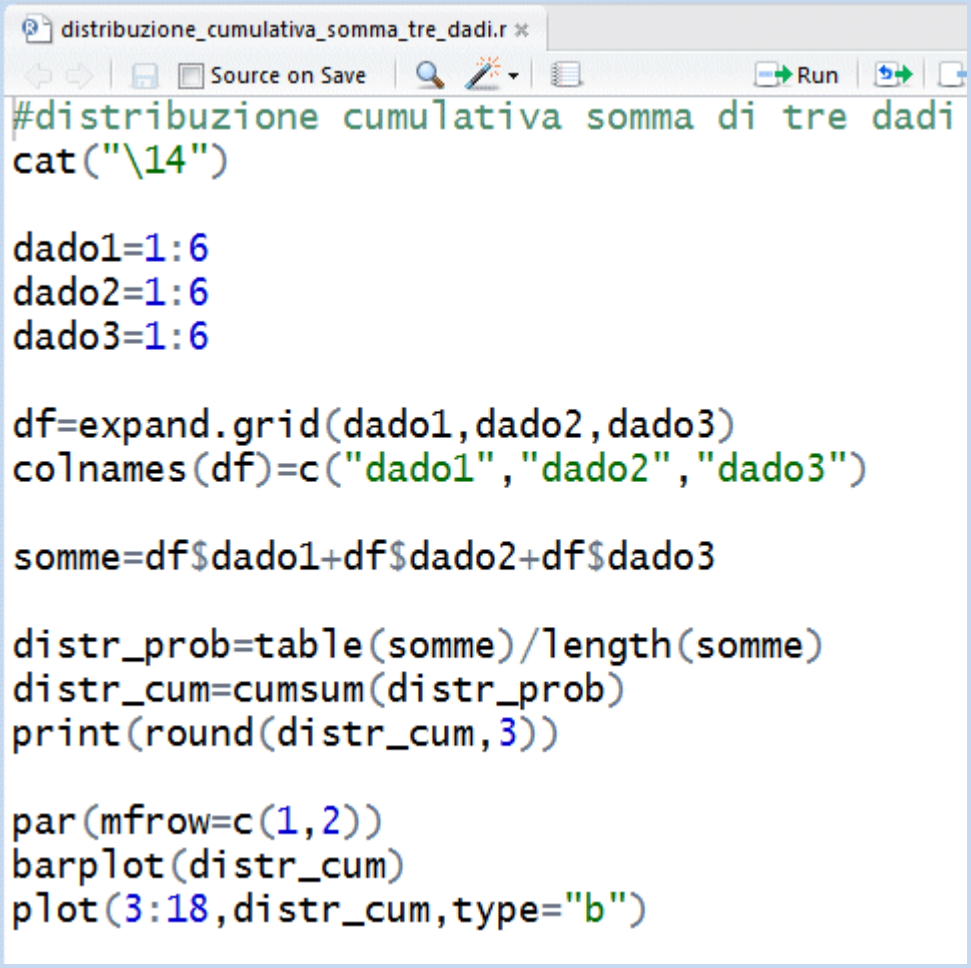
*S = factor(X1+X2, levels=1:12)*

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

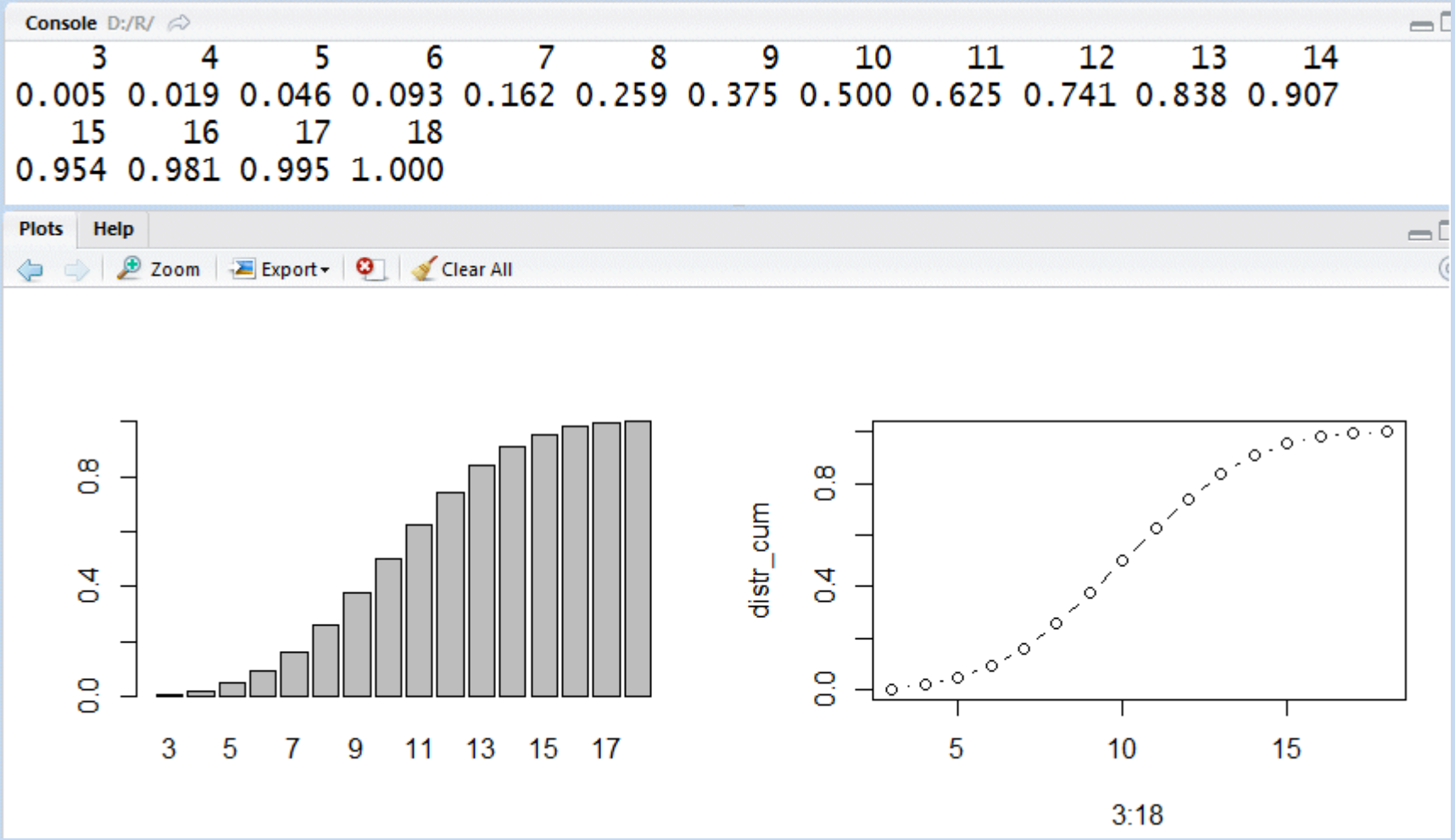
Il prossimo problema introduce la nozione di *funzione di ripartizione* o *funzione di distribuzione cumulativa* di una variabile casuale S: F(s)=prob(S≤s).

**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:



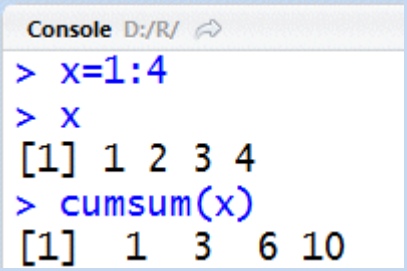
Output:




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

```r
#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)
```

Console D:/R/

```
X
      0       1       2       3
0.01794 0.26784 0.53725 0.17697
```

Fantastico: rispetto al programma precedente è bastato cambiare <u>unicamente</u>

*replace=TRUE*

in

*replace=FALSE*

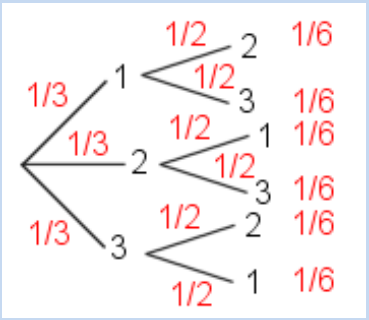
Sarebbe stato così semplice utilizzando un linguaggio di programmazione di tipo generale (come il BASIC, che conoscete, o java o C)?

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"

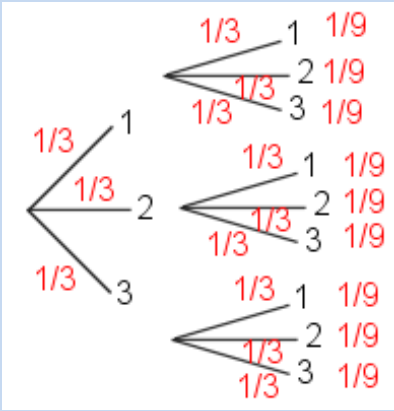



La probabilità dell'evento A è evidentemente p(A)=1/3. La probabilità dell'evento B, se non abbiamo alcuna informazione sulla prima estrazione, è di nuovo 1/3: p(B)=1/3. Siete convinti? Se non lo siete fate una simulazione oppure ragionate sul diagramma ad albero qui a fianco.

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.

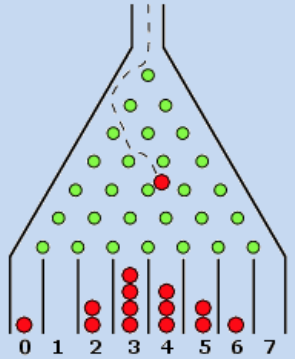


a) Si lancia 10 volte una moneta equa, quante volte si presenta TESTA? [X ~ B(10, 1/2)]

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)*.

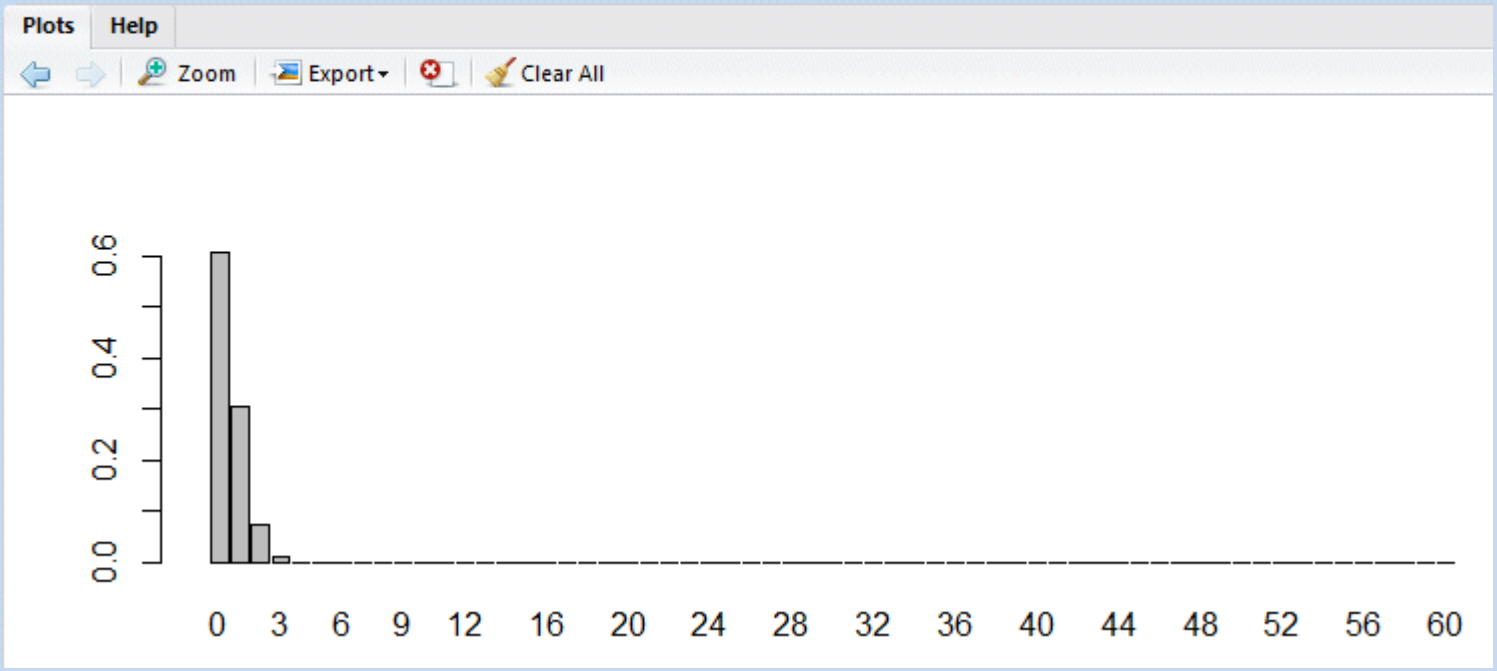


**Esempio 1** La distribuzione B(3, 5/8) dell'esempio 6, 5 biglie bianche, 3 biglie nere, 3 estrazioni con reimmissione:

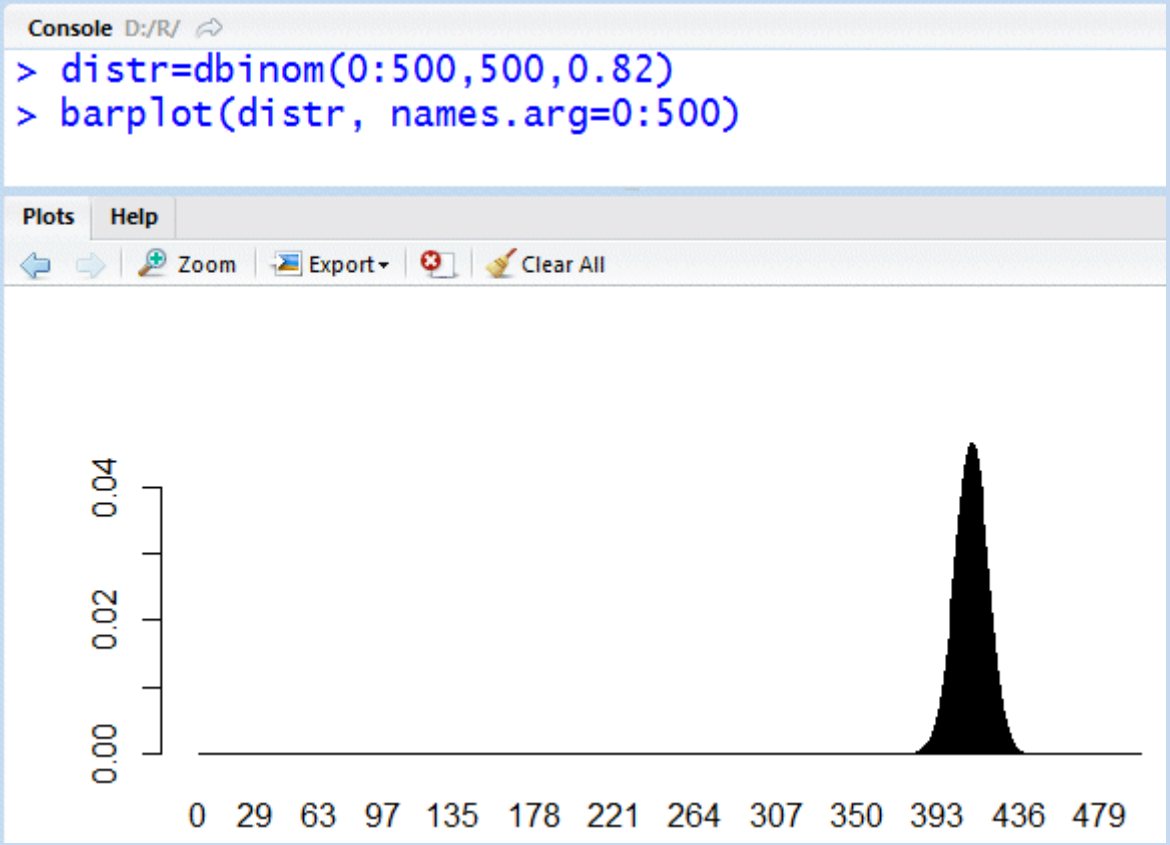
```
Console D:/R/
> dbinom(0:3,3,5/8) # la distribuzione di prob. di X~B(3,5/8)
[1] 0.05273438 0.26367188 0.43945312 0.24414062
> dbinom(2,3,5/8) # probabilità che sia X=2
[1] 0.4394531
```

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

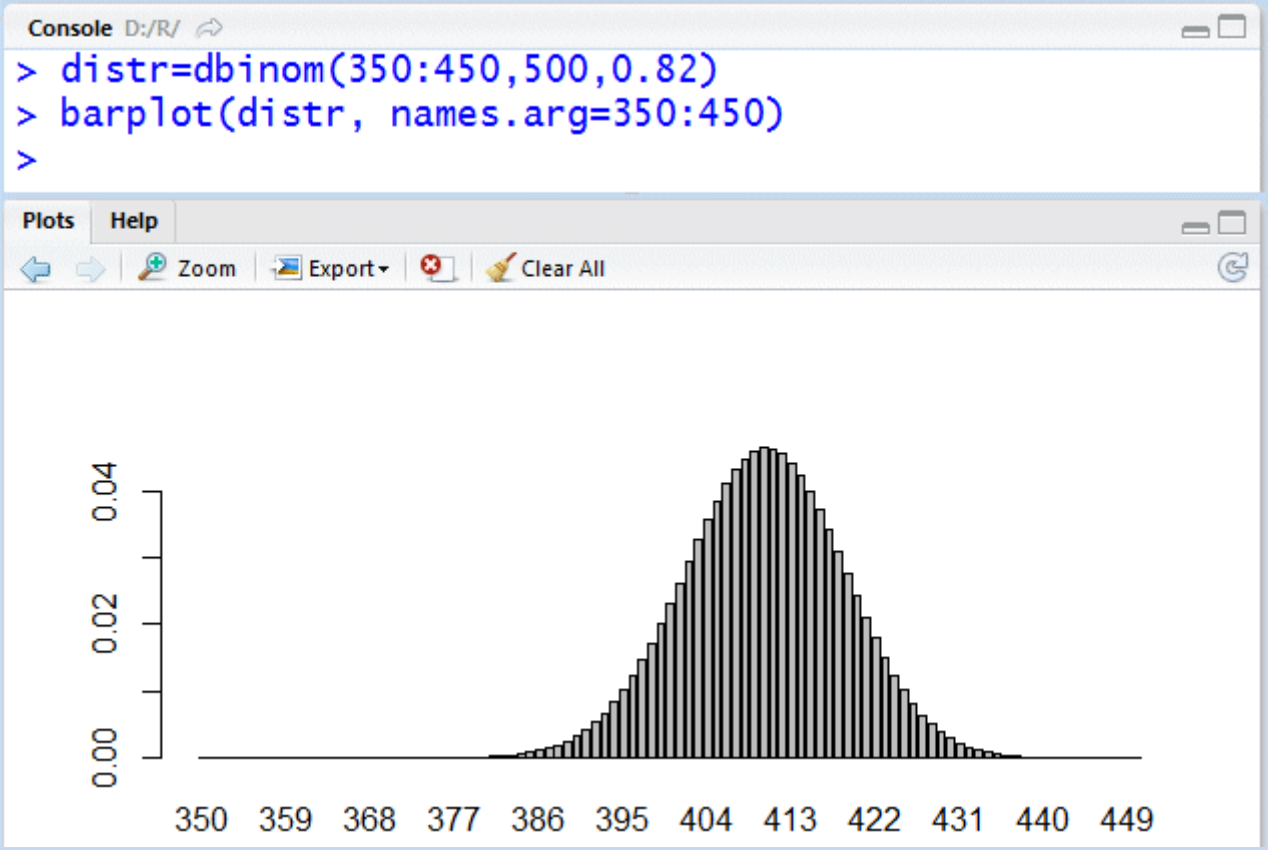
```
Console D:/R/
> distr=dbinom(0:60,60,1/120)
> barplot(distr, names.arg=0:60)
```



**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 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”



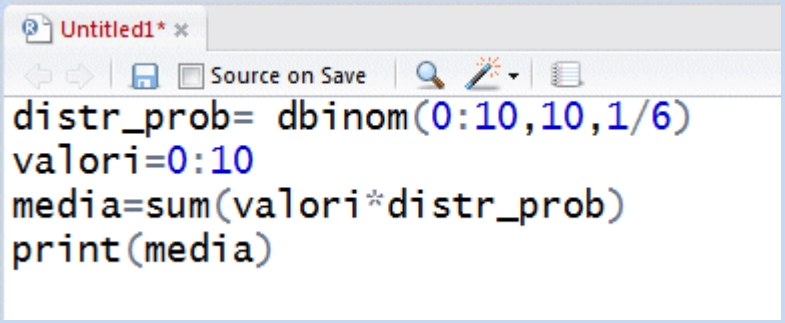
I valori possibili per X sono i numeri interi da 0 a 10 e riconosciamo per X la distribuzione binomiale B(10, 1/6), quindi siamo in grado di calcolare, con R, la probabilità

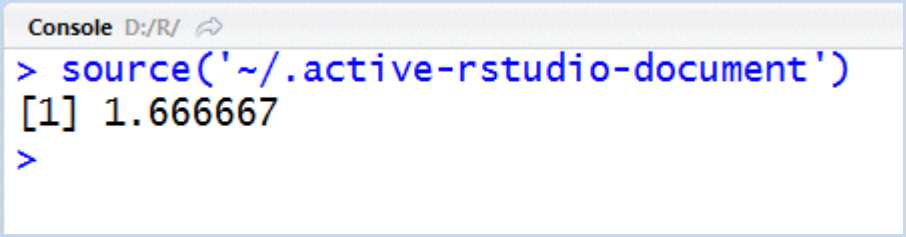
$p(X=i)$

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:

$$\text{media} = (3+1+3+2+5)/5 = 2{,}8$$
$$\text{varianza} = [(3-2{,}8)^2+(1-2{,}8)^2+(3-2{,}8)^2+(2-2{,}8)^2+(5-2{,}8)^2]/5 = 1{,}76$$

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
>
```

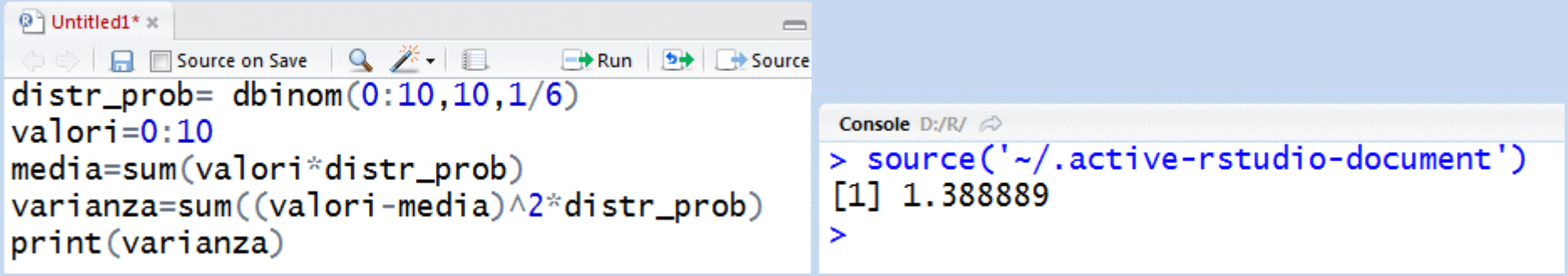
La varianza, calcolata sperimentalmente, è quindi circa 1,4. Cerchiamo ora di capire come si può procedere teoricamente (via calcolo delle probabilità). Sembra sensato definire la ***varianza*** della variabile casuale

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

$x_1, x_2, \ldots, x_n$

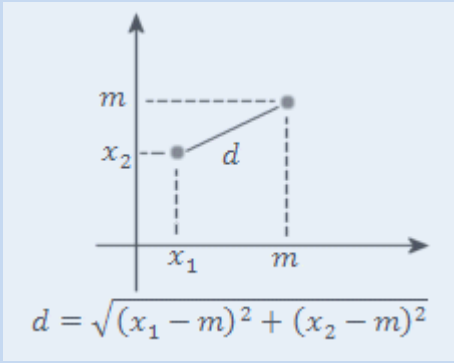
con probabilità $p_1, p_2, \ldots, p_n$ e sia $m=E(X)$; allora

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

Quindi possiamo pensare a $\sigma(X)$ 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 $\sigma$ 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

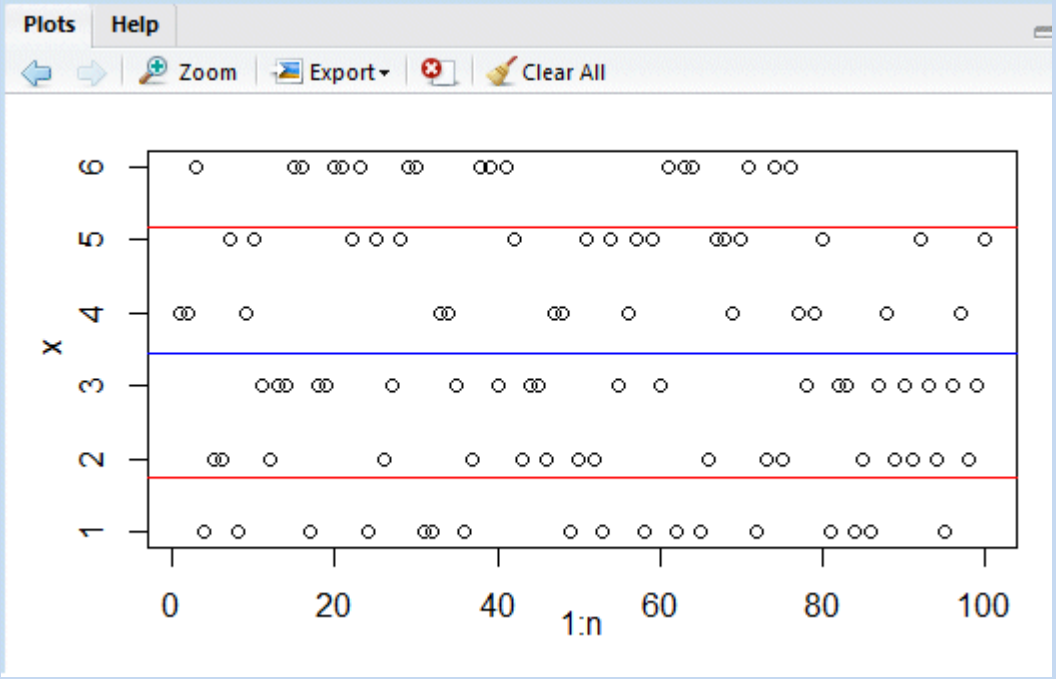
```
cat("\14")
n=100
x=sample(1:6,n,replace=TRUE)
media=mean(x)
varianza=sum((x-media)^2)/n
devst=sqrt(varianza)
sub_x=x[media-devst<=x & x<=media+devst]
freq_rel=length(sub_x)/n
cat("media=", media, "\n")
cat("varianza=", varianza, "\n")
cat("dev. st.=", devst, "\n")
cat("freq. rel.=",freq_rel,"\n")

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.



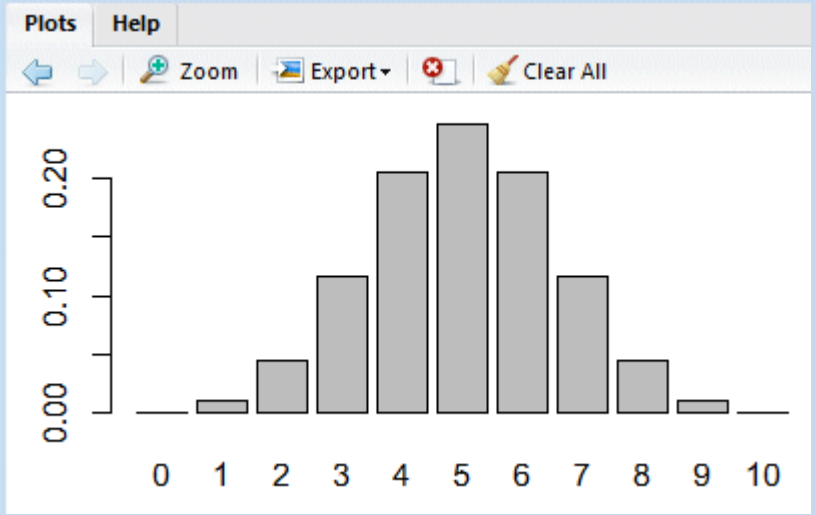
deviazione standard (2).R*

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 $\sigma$ 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.

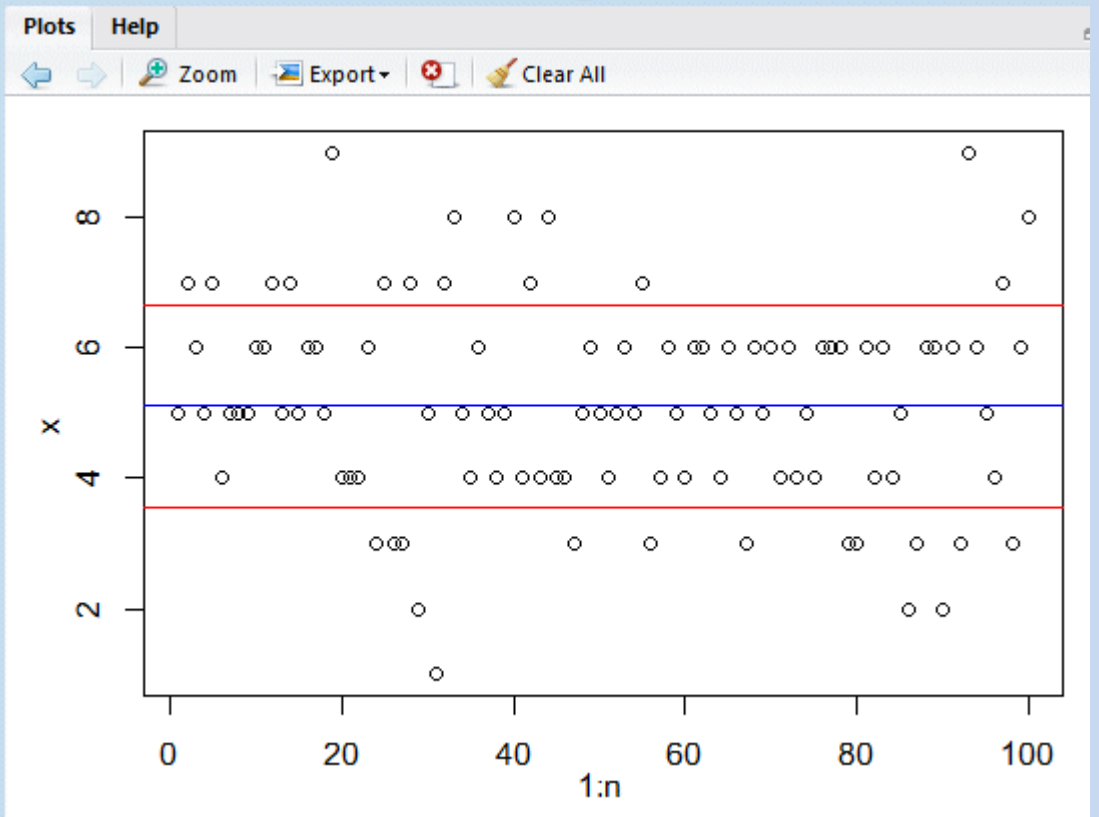
simulazione deviazione standard (2).R*

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***:

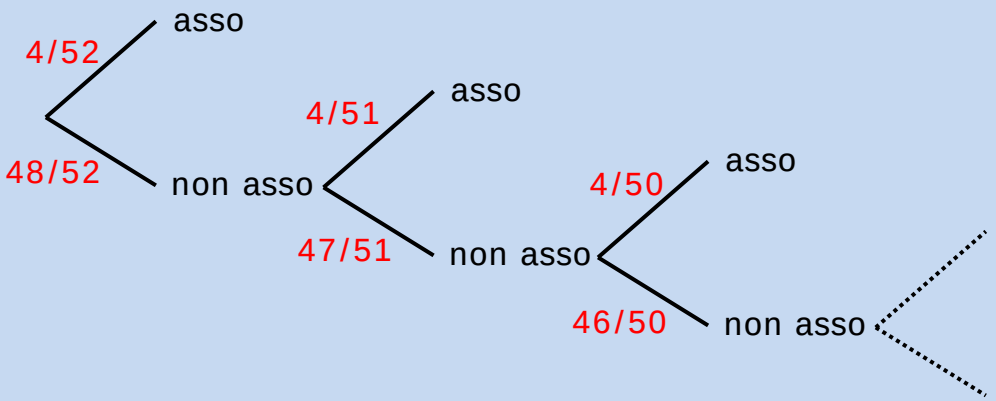
$$Nel\ primo\ caso\ \frac{\sigma(X)}{|E(X)|} = \frac{1{,}7078}{3{,}5} = 0{,}488 = 48{,}8\%$$

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

**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

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

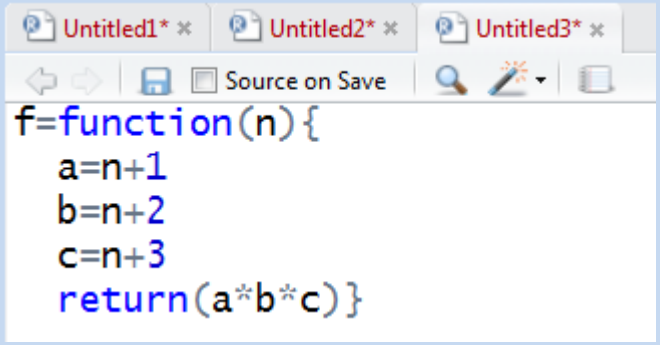
Abbiamo finito? Non proprio! Il calcolo di E(X), un calcolo elementare in cui entrano in gioco solo prodotti e somme di frazioni, è troppo lungo per essere fatto a mano anche con l'aiuto di una calcolatrice non programmabile; dobbiamo scrivere un programma, vediamo come procedere con R.

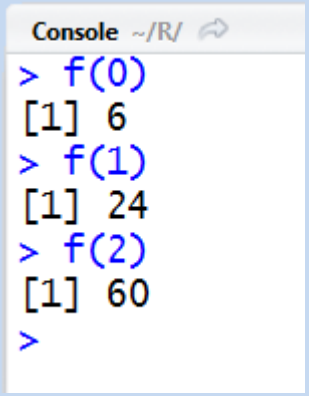
Poiché ci farà comodo definire la funzione p(n) che ci fornisce la probabilità che sia X=n, vediamo prima di tutto come definire una ***funzione*** con R. La struttura per definire una funzione che ad esempio chiamiamo f e che dipende dalla variabile n è la seguente:

```
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
>
```

Qui abbiamo utilizzato il comando *prod(v)* che fornisce il prodotto di tutti gli elementi del vettore v. Ad esempio *prod(1:4)=24*.

Tenete 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?

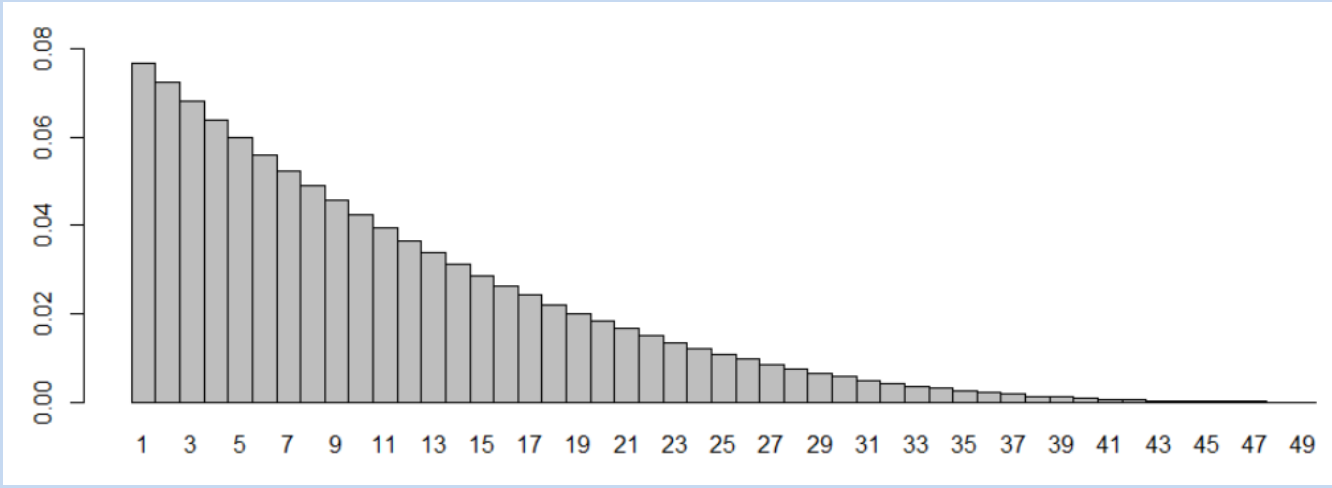
Visto che abbiamo a disposizione la funzione p(n) è interessante ottenere sia la tabella con la distribuzione di probabilità di X sia la rappresentazione grafica di tale distribuzione. Ecco il codice:

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 | | | |

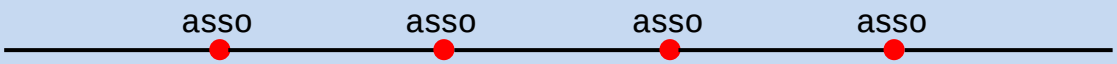



Come si vede la probabilità che il primo asso si presenti all'n-esima estrazione decresce molto rapidamente al crescere di n. E ciò si poteva intuire, ad esempio si capisce che è molto improbabile che il primo asso si presenti alla 30-esima estrazione (la probabilità è 0,0057). Però riflettete sul fatto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