Basi di Dati e Data Analytics - Modulo B

Strumenti per probabilità e statistica

Nicola Torelli, Gioia Di Credico

Dipartimento di Scienze Economiche, Aziendali,
Matematiche e Statistiche B. de Finetti

Università degli studi di Trieste

April 13, 2023

Contents

1 Distribuzioni di probabilità in R

1.1 Distribuzione Binomiale e Poisson

1. Possiamo utilizzare R per valutare la probabilità che $Pr(X = 5)$ dove X è una variabile aleatoria che segue una distribuzione Binomiale $X \sim Bi(20, 0.5)$

```
  choose(20,5)*0.5^5*
  (1-0.5)^15

## [1] 0.01478577
```

La funzione `choose(n,r)` restituisce il coefficiente binomiale $\binom{n}{r}$.
E' utile ricordare che la funzione `factorial(n)` calcola $n!$.

Ora possiamo calcolare la funzione di probabilità per $X \sim Bi(20, 0.5)$

```
#definiamo una sequenza di interi da 0 a 20
x=c(0:20)

#valutiamo ed assegnamo al vettore 'pbin' la P(X=x) per x:0,...,20
pbin=choose(20,x)*.50^x*(1-.50)^(20-x)
pbin

##  [1] 9.536743e-07 1.907349e-05 1.811981e-04 1.087189e-03 4.620552e-03
##  [6] 1.478577e-02 3.696442e-02 7.392883e-02 1.201344e-01 1.601791e-01
## [11] 1.761971e-01 1.601791e-01 1.201344e-01 7.392883e-02 3.696442e-02
## [16] 1.478577e-02 4.620552e-03 1.087189e-03 1.811981e-04 1.907349e-05
## [21] 9.536743e-07
```

Per verificare che si tratta di una funzione di probabilità verichiamo che la somma sia 1.

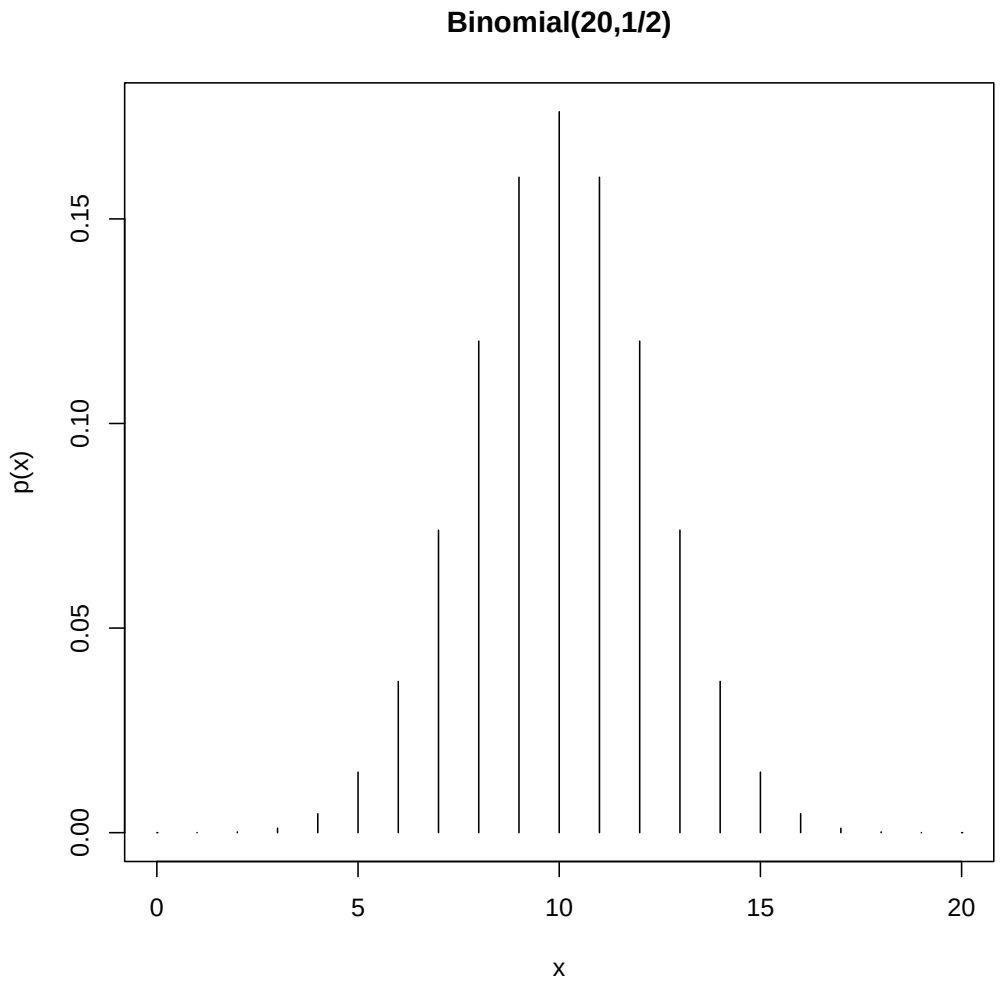
```
  sum(pbin)

## [1] 1
```

Esercizio

Disegna la funzione di probabilità di X.
Aggiungi al grafico la funzione di probabilità della variabile $X \sim Bi(20, 0.25)$

utilizzando la funzione `points()`. *Per distinguere le due funzioni imposta due colori diversi per le due funzioni di probabilità.*

```
plot(0:20,pbin,type="h",main="Binomial(20,1/2)",xlab="x",
     ylab="p(x)")
```



Creiamo una funzione che calcola la funzione di probabilità della Binomiale.

```
binomiale=function(x,n,p){choose(n,x)*p^x*(1-p)^(n-x)}
```

La funzione `binomiale` è ora disponibile e ci permette di calcolare le probabilità per ogni coppia di valori n e p.

Esercizio

Prova a valutare nuovamente $P(X = 5)$ per una Binomiale con $n = 20$ e $p = 0.5$ utilizzando la nuova funzione.
Calcola poi la probabilità di ottenere due volte 6 lanciando 15 volte un dado regolare a 6 facce?

2. Quello che abbiamo fatto finora può essere fatto con funzioni già disponibili in R per valutare le quantità più rilevanti per le principali famiglie parametriche di variabili casuali. E' possibile ottenere facilmente la funzione di probabilità (o densità), la funzione di ripartizione, la funzione quantile e generare numeri casuali per distribuzioni di probabilità, come Binomiale, Poisson, Normale, Gamma e molte altre.
   Le funzioni definite in R solitamente contengono il nome della famiglia di variabili aleatorie preceduto da 4 diversi prefissi che indicano il tipo di funzione. Quando il nome della famiglia è preceduto da `"d"` viene calcolata la probabilità (o la densità), da `"p"` la funzione di ripartizione, da `"q"` la funzione quantile, mentre `"r"` genera valori casuali.
   Utilizzando l'help di R è possibile elencare tutte le distribuzioni di probabilità implementate. Basta scrivere `help(Distributions)` oppure `?Distributions`. Ad esempio, la funzione `pnorm(x,m,s)` valuta la funzione di ripartizione in x per una gaussiana con media m e deviazione standard s.

Esercizio

Verifica che la funzione `dbinom` è equivalente alla funzione `binomiale` definita precedentemente.

3. Proviamo ora a generare valori casuali da un dato modello aleatorio. Potremmo simulare una sequenza di 100 lanci di una moneta ('testa' o 'croce') generando una realizzazione di Bernoulli con $p = 0.5$. La funzione `rbinom()` potrebbe essere usata come segue

```
prova=rbinom(100,1,.5)
prova
```

```
##   [1] 1 0 1 0 0 0 0 1 0 0 0 0 1 0 0 1 1 1 0 1 0 0 0 0 0 1 1 1 1 0 0 1 1 1 1
##  [38] 0 0 1 0 1 0 1 1 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 0 1 1 0 0 1 0 0 0 1 1
##  [75] 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 1 0 1 0
```

Esercizio

4. Si provi a replicare quanto fatto per la distribuzione Binomiale per la distribuzione di Poisson:
- scrivere una funzione che valuti la funzione di probabilità di una Poisson
- rappresentare graficamente la funzione di probabilità
- dimostrare che una distribuzione Binomiale con n grande e p (o $1-p$) può essere ben approssimata da una Poisson con media np. Si utilizzi ad esempio $n = 100$ e $p = 0.02$ e si confrontino i risultati graficamente.

1.2 Distribuzioni di probabilità continue

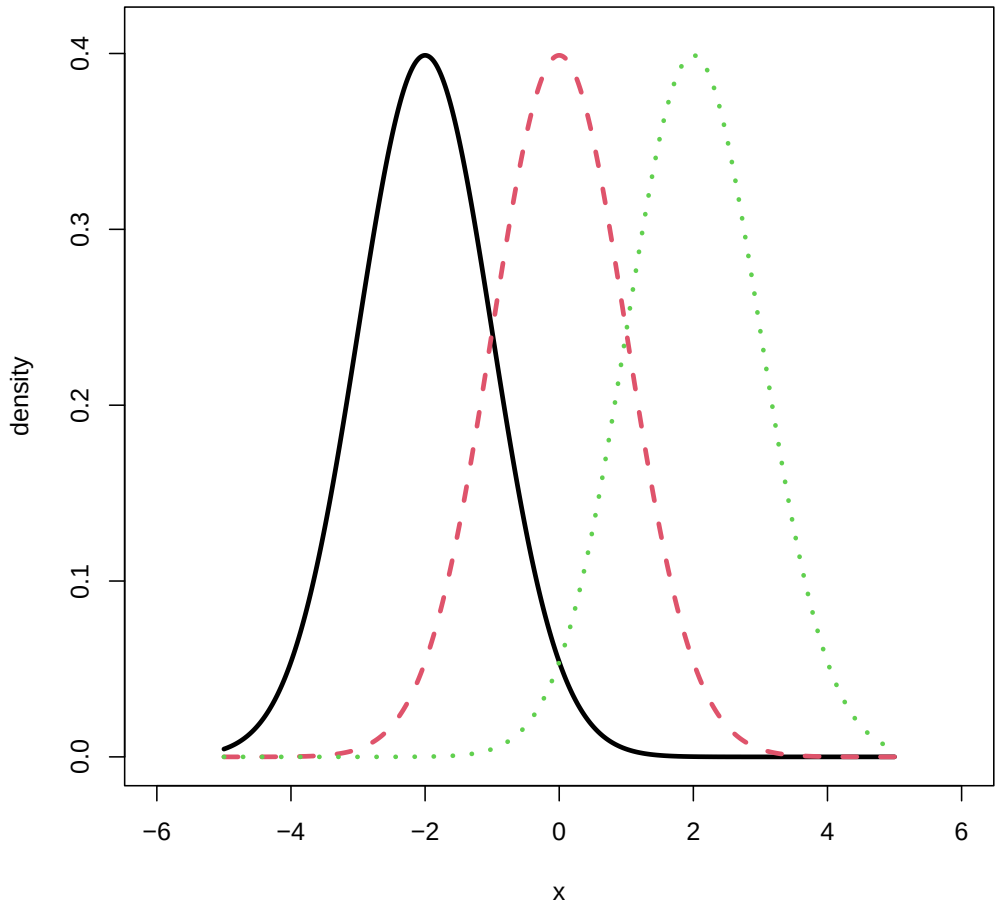
1. Alcune delle distribuzioni di probabilità disponibili in R sono:

Distribuzioni	nome in R	argomenti
beta	beta	shape1, shape2
binomiale	binom	size, prob
Cauchy	cauchy	location, scale
chi-quadro	chisq	df
esponenziale	exp	rate
F	f	df1, df2
gamma	gamma	shape, scale
geometrica	geom	prob
ipergeometrica	hyper	m, n, k
log-normale	lnorm	meanlog, sdlog
binomiale negativa	nbinom	size, prob
normale	norm	mean, sd
Poisson	pois	lambda
Student's t	t	df,
uniforme	unif	min, max
Weibull	weibull	shape, scale

Queste funzioni possono essere utilizzate per ottenere il grafico della funzione di densità di una variabile aleatoria continua. Utilizzando la funzione `plot()`, possiamo rappresentare il valore della funzione di densità per una sequenza di punti x. Proviamo ad ottenere il grafico di diverse funzioni di densità per delle Normali con stessa media ma diversa varianza.

```
xx=seq(-5,5,.01)
plot(xx,dnorm(xx,-2,1), type="l", xlim=c(-6,6), ylim=c(0,0.41),
     lwd=3, xlab="x", ylab="density")
points(xx,dnorm(xx,0,1), type="l", lty=2, col=2, lwd=3)
```

```
points(xx,dnorm(xx,2,1), type="l", lty=3, col=3, lwd=3)
```



In alternativa

```
curve(dnorm(x,-1,1),col=4, lwd=3, lty=4, add=T)
```

Si provi ad aggiungere al grafico densità di normali con la stessa varianza ma diversa media.

2. Per ogni famiglia di variabili aleatorie la funzione di ripartizione e la funzione quantile possono essere facilmente valutate anteponendo al nome che identifica la famiglia la "p" o la "q".

Si provi a calcolare le sequenti quantità per $X \sim N(170, 100)$:
$Pr(X \leq 185)$;
$Pr(165 \leq X \leq 190)$;
scarto interquartile di X;
il 99-esimo percentile di X.

3. Aggiungendo il prefisso "r" viene generato un vettore di numeri (pseudo) casuali. Questa funzione è molto utile, per esercizi di simulazione.

   Come primo esempio, potremmo provare a generare numeri casuali da una data distribuzione e possiamo verificare se questi numeri si comportano come previsto.
   Se generiamo un vettore di numeri casuali da una distribuzione uniforme $X \sim R(0, 1)$), quale sarà la proporzione dei valori che si troveranno al di sotto del valore 0.4? La risposta è, banalmente, circa il 40% dei valori. Vediamo se funziona.

```
#500 are drawn from a R(0,1)
simu=runif(500,0,1)
sum(simu<.4)/length(simu)

## [1] 0.39
```

   La proporzione che abbiamo trovato è molto vicina a 0.4 e differisce solo perché abbiamo generato solo un numero limitato di valori casuali.

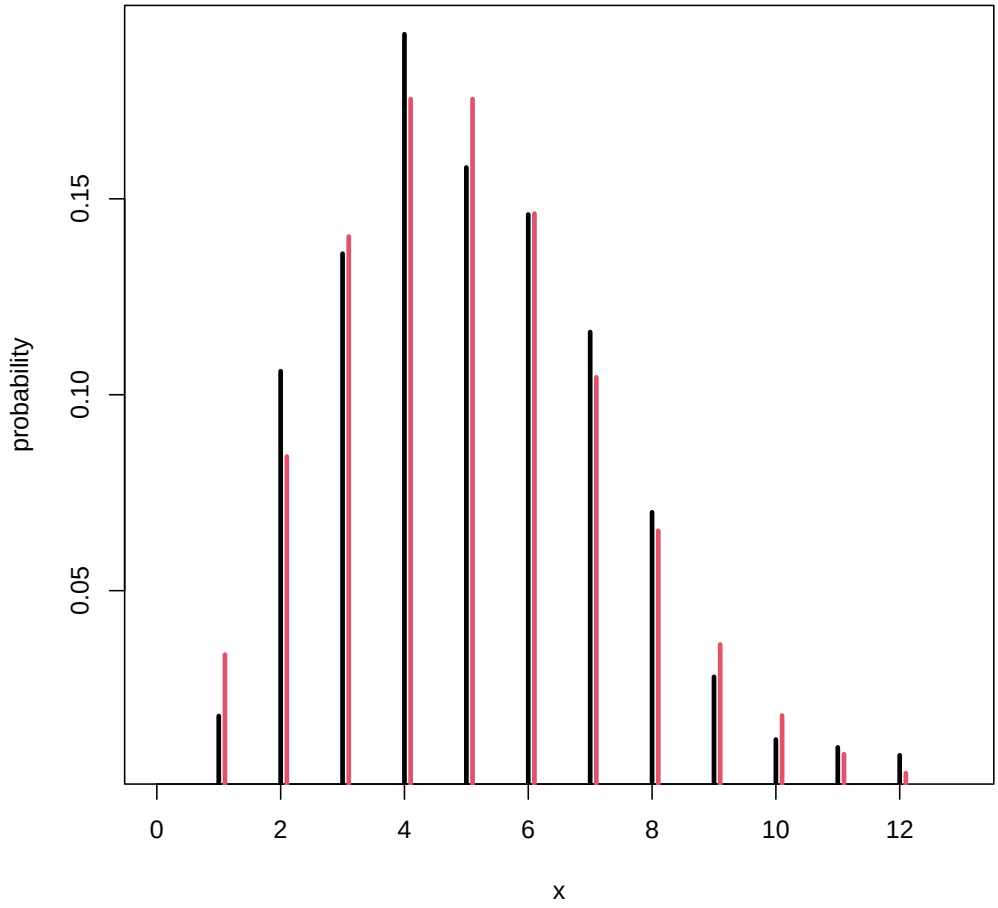
   Ora proviamo a generare numeri casuali da un Poisson con media 5, e confrontiamo il grafico della distribuzione di frequenza dei valori generati con le probabilità teoriche di una Poisson.

```
set.seed(3)
n = 500
#generiamo 500 valori casuali da una Poisson con media 5
dati.pois = rpois(n,5)
#otteniamo la distribuzione di frequenze
```

```r
tab = table(dati.pois)
tab
```

```
## dati.pois
##  1  2  3  4  5  6  7  8  9 10 11 12
##  9 53 68 96 79 73 58 35 14  6  5  4
```

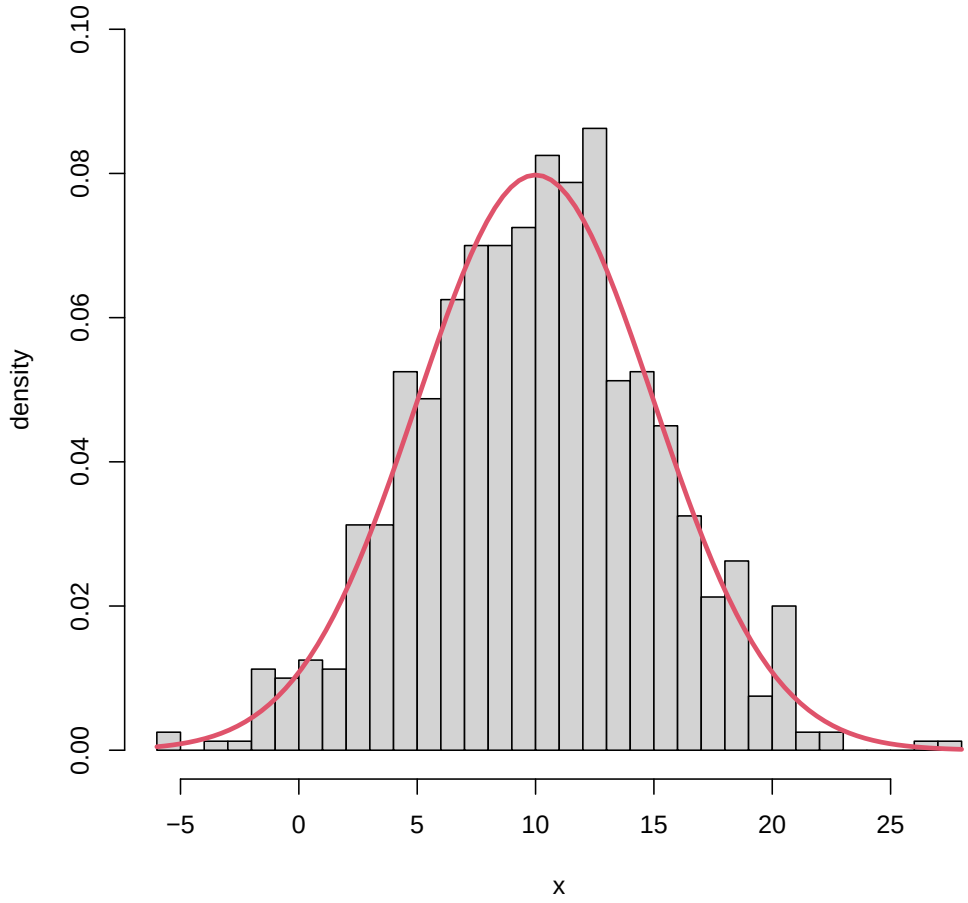
```r
# creiamo un data frame con due variabili:
#i valori (di tipo 'character') e le frequenze relative
tabo = as.data.frame(tab)
# convertiamo i valori in 'numeric'
punti = as.numeric(levels(tabo$dati.pois))
freq = as.numeric(tab/n)
# rappresentiamo graficamente le frequenze
plot(punti, freq,type="h", ylab="probability", xlab="x",
     xlim=c(0,13), lwd=3)

#aggiungiamo la funzione di probabilità per una Poisson con media 5 solo per
# i punti osservati
points(punti+.1,dpois(punti,5),type="h",col=2, lwd=3)
```



4. Ora generiamo 800 valori casuali da una Gaussiana con media 10 e varianza 25.

   Possiamo quindi confrontare l'istogramma dei valori ottenuti con la funzione di densità teorica della Normale.

```
x=rnorm(800, 10,sqrt(25))
# breaks specifica il numero di classi
hist(x, freq=F, ylim=c(0,0.1), breaks=30, main="", ylab="density")
curve(dnorm(x, 10,sqrt(25)),add=T, col=2, lwd=3)
```


density
0.00
0.02
0.04
0.06
0.08
0.10
−5
0
5
10
15
20
25
x