R - seconda lezione

1 Simulazione e rappresentazioni grafiche in R

R contiene funzioni interne che fanno riferimento a distribuzioni note. Relativamente a una fissata distribuzione, possiamo valutarne la distribuzione cumulata o la densità, o uno o più quantili, oppure generare k realizzazioni indipendenti da tale distribuzione. Vedremo come disegnare funzioni di ripartizione e di densità, tramite curve o grafici "a bastoncino" e come rappresentare le distribuzioni empiriche di vettori di dati simulati tramite istogrammi e grafici "a bastoncino".

1.1 Generazione di campioni i.i.d. da distribuzioni note

Per ognuna delle distribuzioni viste a lezione (e per tante altre), R puo' generare un valore casuale dalla distribuzione prescelta, o calcolare la densita' (o la massa di probabilita') in un punto, la funzione di ripartizione, i quantili di livello q in [0,1].

Ogni distribuzione e' contraddistinta da un nome "mnemonico", e ciascuna delle funzioni sopra indicate viene individuata tramite un prefisso che va ad aggiungersi al nome della distribuzione, come di seguito elencato:

d - densita'

p - funzione di ripartizione

q - quantile

r - numero casuale dalla distribuzione prescritta

Esempi: `dbinom(),rexp(),qnorm()`.

Ad esempio, per la distribuzione normale standardizzata abbiamo le seguenti funzioni: `dnorm(x)` calcola il valore della densita' N(0,1) in x

Oss: `dnorm(x)=dnorm(x,mean=0,sd=1)`

`pnorm(x)` calcola il valore della funzione di ripartizione in x.
`qnorm(a)` calcola il quantile di livello `a`
`rnorm(n)` genera un campione di dimensione n da una normale standard .
Naturalmente `rnorm(100,10,1)` genera 100 realizzazioni indipendenti da una distribuzione N(10,1). Distribuzioni disponibili:

Distrib.	Nome R	Parametri	Defaults
beta	beta	shape1, shape2,ncp	
binomial	binom	size, prob	
Cauchy	cauchy	location, scale	0,1
chi-squared	chisq	df, ncp	
exponential	exp	rate	1
F	f	df1, df1, ncp	
gamma	gamma	shape, scale	scale=1
geometric	geom	prob	
hypergeometric	hyper	m, n, k	
log-normal	lnorm	meanlog, sdlog	0,1
logistic	logis	location, scale	0,1
neg. binomial	nbinom	size, prob	
normal	norm	mean, sd	0,1
Poisson	pois	lambda	
Student t	t	df, ncp	
uniform	unif	min, max	0,1
Weibull	weibull	shape, scale	scale=1

Per vedere l'andamento della funzione di ripartizione di una normale:

a) standard

```
> x <- seq(-5,5,length=100)
> ripx <- pnorm(x)
> plot(x, ripx, type="l")
```

b) di media 2 e varianza 0.5^2

```
> ripx1 <- pnorm(x, 2, 0.5)
> lines(x, ripx1, col=2) #questo comando AGGIUNGE un grafico al precedente
```

il comando

```
> points(x, ripx1, type="l",col=2)
```

ha lo stesso effetto del precedente.

La probabilita' di ottenere valori fra 1 e 3 per una normale standard e':

```
> pnorm(3) - pnorm(1)
```

La stessa probabilita' per una normale di media 2 e varianza $(0.5)^2$ e' invece:

```
> pnorm(3,2,0.5) - pnorm(1,2,0.5)
```

Per disegnare un diagramma a bastoncini per la distribuzione binomiale di parametri n=15 e p=0.8, e, successivamente, n=50, p=0.8. Sovrappone a questo grafico la curva di una densità normale con media e varianza adeguate (`meanbin`,sdbin).

```
> x <- 0:15
> y <- dbinom(x,15,0.8)
> plot(x,y,type='h')
> points(x,y,pch=20)
> title('Distribuzione Binomiale-n=15,p=0.8')
```

aggiungiamo la curva della densita' normale:

```
> x1 <- seq(0,15,0.1)
> lines(x1,dnorm(x1,meanbin,sdbin),col=2)
```

facciamo aumentare n:

```
> x <- 0:50
> y <- dbinom(x,50,0.8)
> plot(x,y,type='h')
> points(x,y,pch=20)
> title('Distribuzione Binomiale-n=50,p=0.8')
```

aggiungiamo la curva della densita' normale:

```
> x1 <- seq(0, 50, 0.1)
> lines(x1,dnorm(x1,meanbin,sdbin),col=2)
```

La probabilita' sotto la binomiale con n=15 e p=0.8 di ottenere un numero di successi minore o uguale a 2 e'

```
> pbinom(2,15,0.8)
```

ovvero

```
> dbinom(0,15,0.8) + dbinom(1,15,0.8) + dbinom(2,15,0.8)
```

Se fossimo interessati a 7 realizzazioni indipendenti di una variabile con distribuzione di Poisson di parametro $\lambda = 2$:

```
> rpois(7,2)
```

eccetera...

1.2 Campionamento da urne o popolazioni finite

Il comando `sample()` permette di estrarre (con o senza reinserimento) un certo numero di valori da un insieme prefissato. Per generare i possibili risultati di 10 lanci di un dado equilibrato, possiamo fare cosi':

```
> sample(1:6,10,replace=T)
```

Possiamo anche creare un'urna con 4 palline bianche e 3 nere:

```
> urn <- c('b1','b2','b3','b4','n1','n2','n3')
```

Estraiamo due palline senza reinserimento:

```
> sample(urn,2)
```

con reinserimento:

```
a> sample(urn,2,replace=T)
```

1.2.1 Confronto fra distribuzioni empiriche e teoriche

Per vedere l'accostamento di variabili discrete a modelli teorici di riferimento, usiamo un metodo grafico.

Esempio 1: generiamo 100 dati da una Poisson di parametro $\lambda = 2$.

```
> x <- rpois(100,2)
```

Il comando `table()` ci da' la distribuzione delle frequenze assolute; quindi otteniamo le frequenze relative con il comando:

```
> x <- rpois(100,2)
> relfreq <- table(x)/100
> relfreq
x
   0    1    2    3    4    5    6    8
0.15 0.28 0.22 0.17 0.10 0.06 0.01 0.01 attr(,"class") [1] "table"
> names(relfreq)
[1] "0" "1" "2" "3" "4" "5" "6" "8"
> as.numeric(names(relfreq))
[1] 0 1 2 3 4 5 6 8
> as.numeric(relfreq)
[1] 0.15 0.28 0.22 0.17 0.10 0.06 0.01 0.01
```

Per il grafico della distribuzione di frequenze relative (diagramma a bastoncini):

```
> plot(as.numeric(relfreq),type="h")
```

Dal momento che alcuni valori potrebbero avere frequenza nulla (nel caso del nostro esempio, 8), è senz'altro meglio operare diversamente, come indicato di seguito:

```
> plot(as.numeric(names(relfreq)),relfreq, type="h")
> points(as.numeric(names(relfreq)),relfreq, pch=20)
```

Nota: `as.numeric(x)` fornisce la versione numerica di x.
Nel caso esaminato, ad esempio, `names(relfreq)` è una variabile alfanumerica, e `as.numeric(names(relfreq)` la rende numerica, il che ci consente di utilizzarla per rappresentare le ascisse del grafico.
Per confrontare la distribuzione empirica con quella teorica di una Poisson di parametro lambda=2, salviamo la probabilita' di ottere 0,1,2,3,4,...,8 sotto una Poisson(2)

```
> prob <- dpois(0:8,2)
```

Quindi confrontiamo le frequenze relative empiriche con le probabilita' teoriche.
Per effettuare il confronto grafico:

```
> lines((0:8)+0.1, prob,lty=4,type="h",col=2)
> points((0:8)+0.1, prob,pch=3,col=2)
```

Esempio 2: Ripetiamo ora lo stesso procedimento per una variabile esponenziale negativa di parametro lambda=2

```
> x <- rexp(1000,2)
> mean(x)
[1] 0.4946056
```

Dato che si tratta di una variabile continua, possiamo rappresentare le frequenze osservate attraverso un istogramma, con il comando `hist(x)`. Per ottenere l'istogramma delle frequenze relative aggiungiamo l'opzione `prob=T` :

```
> hist(x,prob=T,nclass=15)
```

Sovrapponiamo la densita' esponenziale di parametro 2:

```
> xexp <- seq(0,30,0.1)
> yexp <- dexp(xexp,2)
> lines(xexp,yexp)
```

Per un confronto grafico tra le distribuzioni possiamo anche utilizzare i comandi `qqplot()` e `qqpoints()` come si vedrà nel seguito.