

INFERRE

PROBABILITY AND MATHEMATICAL STATISTICS

4

Direttore / Editor in chief

Corrado TANASI

Università degli Studi di Palermo

Comitato Scientifico / Editorial Board

Giuseppe RAO

Università degli Studi di Palermo

Francesco RUSSO

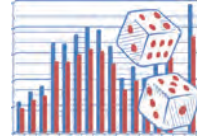
University of Cape Town

Francesco TULONE

Università degli Studi di Palermo

INFERRE

PROBABILITY AND MATHEMATICAL STATISTICS



Credo che il calcolo delle probabilità sia l'unica branca della matematica in cui buoni autori ottengono spesso risultati completamente sbagliati.

Charles PIERCE

Scopi e obiettivi

La collana è orientata verso due direzioni diverse ma connesse tra loro. La prima ha come obiettivo quello di trasferire gradualmente al lettore (studente o meno) un messaggio preciso, anche se a prima vista ovvio: i metodi della matematica pura sono fondamentali e formano il tessuto connettivo essenziale al calcolo delle probabilità e alla statistica. Ciò non è scontato, perché si nota una carenza significativa nel panorama editoriale italiano di testi orientati in tal senso. Questa chiave di lettura, quindi, privilegia testi con linguaggio formale della matematica pura nella teoria delle probabilità e della statistica matematica, senza escludere le applicazioni. Alcuni volumi sono pensati per lettori con una solida base in matematica, e possono adattarsi molto bene a laureati e studenti universitari in matematica.

La seconda direzione, usando un linguaggio meno formale ma pur sempre essenziale e rigoroso, riguarda l'applicazione del calcolo delle probabilità e della statistica descrittiva a specifici temi, quali la teoria dei giochi, le scienze naturali, la chimica, la farmacia, la medicina, l'ingegneria, la biologia, la sociologia, l'economia e la finanza.

Aims and scopes

The present series focuses mainly on two aspects, which are connected but different at the same time. The first aspect deals with the fact that the methods of the pure mathematics are fundamental for the essential structures of probability calculus and statistics. The present

series wants to give this precise message to the readers (students or not), because it is not an obvious aspect. There are in fact few textbooks, which are written in this perspective, that is, based on the formal language of the pure mathematics in theory of probability and statistics (together with its applications).

The second aspect is placed at a different level, which is based on the applications of the probability calculus and statistics to some specific subjects, like game theory, natural sciences, chemistry, pharmacy, medicine, engineering, biology, sociology, economics and finance. Some volumes are designed for readers with a solid basis in mathematics and they can fit very well to graduate and undergraduate students in mathematics.



Vai al contenuto multimediale

Mariantonietta Ruggieri

Laboratorio informatico–statistico con R





Aracne editrice

www.aracneeditrice.it
info@aracneeditrice.it

Copyright © MMXVIII
Gioacchino Onorati editore S.r.l. – unipersonale

www.gioacchinoonoratieditore.it
info@gioacchinoonoratieditore.it

via Vittorio Veneto, 20
00020 Canterano (RM)
(06) 45551463

ISBN 978-88-255-2045-3

*I diritti di traduzione, di memorizzazione elettronica,
di riproduzione e di adattamento anche parziale,
con qualsiasi mezzo, sono riservati per tutti i Paesi.*

*Non sono assolutamente consentite le fotocopie
senza il permesso scritto dell'Editore.*

I edizione: dicembre 2018

*A Federica,
perchè cresca con la consapevolezza
che studiare è un diritto più che un dovere.*

II Premessa

Modulo I

15 Capitolo I *L'ambiente R*

1.1. Cosa è R, 15 – 1.1.1. *Introduzione all'ambiente statistico R*, 15 – 1.1.2. *Come iniziare con R*, 15 – 1.2. Gli oggetti, 17 – 1.2.1. *Vettori e variabili*, 17 – 1.2.2. *Matrici*, 19 – 1.3. *Matematica con R*, 21 – 1.3.1. *Operazioni con vettori e matrici*, 21 – 1.3.2. *Risoluzione di equazioni e sistemi*, 24 – 1.3.3. *Massimi e minimi di una funzione*, 26 – 1.4. La gestione degli oggetti, 27 – 1.4.1. *Estrarre elementi da vettori o matrici*, 27 – 1.4.2. *Il workspace*, 29 – 1.4.3. *Tipologie di oggetti*, 31 – 1.5. *Funzioni, script o programmi*, 35 – 1.6. *Test e cicli*, 36 – 1.7. *Input e Output*, 38 – 1.7.1. *Come salvare l'output*, 38 – 1.7.2. *Importare dati in formato testo*, 39 – 1.8. *Packages di R*, 40.

41 Capitolo II *Statistica descrittiva con R*

2.1. *Immissione dei dati*, 41 – 2.1.1. *Classificazione delle variabili*, 41 – 2.1.2. *Dataset statistici*, 43 – 2.2. *Serie di dati e distribuzioni di frequenza*, 46 – 2.2.1. *Distribuzioni di frequenza per valori singoli*, 46 – 2.2.2. *Distribuzioni di frequenza per classi di valori*, 48 – 2.3. *Rappresentazioni grafiche*, 51 – 2.3.1. *Grafico a colonne*, 51 – 2.3.2. *Grafico a torta*, 52 – 2.3.3. *Diagramma cartesiano*, 52 – 2.3.4. *Istogramma e poligono di frequenza*, 53 – 2.3.5. *Funzione di ripartizione*, 56 – 2.3.6. *Boxplot*, 60 – 2.4. *Indici di sintesi*, 62 – 2.4.1. *Le medie*, 62 – 2.4.2. *Gli indici di eterogeneità*, 65 – 2.4.3. *Gli indici di variabilità*, 66 – 2.4.4. *Indici di asimmetria e curtosi*, 68.

71 Capitolo III *Relazioni tra variabili*

3.1. *L'analisi delle tabelle di contingenza*, 71 – 3.2. *La regressione lineare semplice*, 78 – 3.3. *La dipendenza in media: il rapporto di correlazione*, 87.

Modulo II

91 Capitolo IV

Calcolo delle probabilità con R

4.1. Esperimenti casuali, 91 – 4.2. Calcolo combinatorio, 92 – 4.3. Valore atteso e varianza di variabili casuali, 93 – 4.4. Variabili casuali discrete, 98 – 4.4.1. *V.c. binomiale*, 98 – 4.4.2. *V.c. bernoulliana*, 100 – 4.4.3. *V.c. geometrica*, 101 – 4.4.4. *V.c. binomiale negativa*, 102 – 4.4.5. *V.c. ipergeometrica*, 103 – 4.4.6. *V.c. di Poisson*, 104 – 4.5. Variabili casuali continue, 106 – 4.5.1. *V.c. normale o di Gauss*, 106 – 4.5.2. *V.c. uniforme*, 111 – 4.5.3. *V.c. chi-quadrato*, 112 – 4.5.4. *V.c. t di Student*, 114 – 4.5.5. *V.c. F*, 115 – 4.5.6. *V.c. esponenziale*, 116 – 4.5.7. *V.c. gamma*, 117 – 4.5.8. *V.c. beta*, 119 – 4.6. Simulazione di variabili casuali, 121.

125 Capitolo V

Campionamento e inferenza

5.1. Legge dei grandi numeri, 125 – 5.2. Teorema centrale del limite, 128 – 5.3. Funzione di (log)verosimiglianza, 130 – 5.4. Intervalli di confidenza, 135 – 5.4.1. *IC per la media*, 135 – 5.4.2. *IC per una proporzione*, 139 – 5.4.3. *IC per la varianza*, 141 – 5.5. Verifica di ipotesi, 142 – 5.5.1. *Verifica di ipotesi per la media*, 143 – 5.5.2. *Verifica di ipotesi per una proporzione*, 144 – 5.5.3. *Verifica di ipotesi per la varianza*, 146.

149 Bibliografia

Appendice

153 *Esercizi modulo I*

159 *Esercizi modulo II*

165 *Soluzioni esercizi modulo I*

185 *Soluzioni esercizi modulo II*

Premessa

Il corso di “Laboratorio informatico–statistico” si propone di fornire strumenti informatici di supporto agli argomenti di Statistica descrittiva (modulo I) e inferenziale (modulo II), pertanto prevede che tali conoscenze siano preliminari.

In Appendice si riportano alcuni esercizi con le relative soluzioni; per scaricare i dataset utilizzati, collegarsi al seguente link e scegliere la cartella “DATASET_R”: https://www.dropbox.com/sh/adwmg3yi29auesk/AAATkW_WInB-7KgR9vuXO4d4a?dl=0.

MODULO I

L'ambiente R

1.1. Cosa è R

1.1.1. Introduzione all'ambiente statistico R

R più che un software si può definire un ambiente statistico di programmazione a oggetti¹, che consente di manipolare quantità di dati consistenti, effettuare calcoli, visualizzare informazioni e grafici, trattare modelli statistici anche complessi.

Come S-PLUS ha origine dal linguaggio S². Il nome “R” deriva probabilmente dalle iniziali del nome dei due sviluppatori originari: Robert Gentleman e Ross Ihaka.

R è un linguaggio *free*, sotto la licenza GNU — General Public Licence — della Free Software Foundation, dunque si può scaricare gratuitamente da Internet all'indirizzo: <http://cran.r-project.org/>.

È curato da statistici ed esperti di informatica soprattutto di fama internazionale ed è in costante sviluppo. Il gruppo di sviluppatori di R costituisce l'R Development Core Team.

R è *cross-platform*, ossia è compatibile con vari processori e sistemi operativi ed è *open-source*, cioè è possibile accedere al codice interno e proporre modifiche.

1.1.2. Come iniziare con R

Se R è stato installato correttamente, al suo avvio si apre una finestra, chiamata “R Console”, che fornisce le seguenti informazioni:

1. Vettori, funzioni, tabelle, grafici.
2. Il nome deriva dal linguaggio C, in inglese “SEE”.

R version 3.5.0 (2018-04-23) – “Joy in Playing”
 Copyright (C) 2018 The R Foundation for Statistical Computing
 Platform: x86_64-w64-mingw32/x64 (64-bit)

R è un software libero ed è rilasciato “senza alcuna garanzia”.
 Siamo ben lieti se potrai redistribuirlo, ma sotto certe condizioni.
 Scrivi “license()” o “licence()” per dettagli su come distribuirlo.

R è un progetto di collaborazione con molti contributi esterni.
 Scrivi “contributors()” per maggiori informazioni e “citation()”
 per sapere come citare R o i pacchetti di R nelle pubblicazioni.

Scrivi “demo()” per una dimostrazione, “help()” per la guida in linea, o
 “help.start()” per l’*help* navigabile con browser HTML.
 Scrivi “q()” per uscire da R.

>

Quest’ultimo è il “prompt” di R, dopo il quale possono essere inseriti tutti i comandi.

I comandi possono essere separati da un punto e virgola o semplicemente essere immessi in una nuova linea, digitando il tasto “Invio”.

Il carattere + sta a indicare che il comando continua nella linea successiva.

I comandi già eseguiti possono essere richiamati da tastiera mediante i tasti freccia verticali (↑, ↓) e modificati mediante i tasti freccia orizzontali (←, →).

Bisogna tener presente che R è *case-sensitive*, cioè i caratteri minuscoli sono considerati diversi dai caratteri maiuscoli.

All’avvio di R vengono fornite alcune indicazioni. Ad esempio viene suggerito il comando per uscire dall’ambiente R:

```
> q()
```

Tale procedura è consentita anche dal menu dell’interfaccia grafica (File – Exit oppure ×).

È possibile ottenere istruzioni sui comandi, attraverso l’*help* online, fortemente consigliato. Le due istruzioni sono equivalenti:

```
> help(nome comando)
```

```
>?nome comando
```


esempi:

```
> help(q)
>?q
```

Alcuni comandi necessitano delle virgolette, come ad esempio *for*:

```
> help("for")
>?"for"
```

Informazioni più dettagliate su come utilizzare l'*help*, si possono ottenere digitando il comando:

```
>?help
```

È possibile aprire una pagina di *help* in formato *html* attraverso il comando:

```
> help.start()
```

Per alcune dimostrazioni si può utilizzare il comando:

```
> demo()
```

ad esempio

```
> demo(graphics)
```

mostra una serie di grafici che è possibile costruire con R.

1.2. Gli oggetti

1.2.1. Vettori e variabili

Gli elementi che R consente di creare e gestire sono gli *oggetti*. Gli oggetti più semplici sono gli scalari e i vettori.

L'istruzione di assegnazione è `<-`; anche il simbolo `=` è consentito.

Dunque, volendo assegnare a una variabile *x* il valore 7, il comando da utilizzare è il seguente:

```
> x<-7
```

Per visualizzare il contenuto dell'oggetto `x`, basta richiamare `x`:

```
> x
[1] 7
```

L'output è preceduto da `[1]`, che sta a indicare la posizione del primo elemento del vettore `x`. `x`, sebbene sia uno scalare, viene interpretato come vettore di lunghezza 1.

Creiamo un vettore con più elementi. Le seguenti tre istruzioni sono equivalenti, ma la seconda e la terza non sono consigliate:

```
> y<-c(2, 6, 8, 9, 3)
> c(2, 6, 8, 9, 3)-> y
> assign("y", c(2, 6, 8, 9, 3))
```

Il comando "`c`" viene utilizzato per concatenare gli elementi.

Gli elementi possono anche essere inseriti uno per volta con l'istruzione "`scan()`", separati da spazio o da un Invio, di cui l'ultimo deve essere a vuoto:

```
> y<-scan()
1: 2 6 8 9 3
6:
Read 5 items
```

```
> y<-scan()
1: 2
2: 6
3: 8
4: 9
5: 3
6:
Read 5 items
```

Se richiamiamo `y`, otteniamo i valori in esso contenuti:

```
> y
[1] 2 6 8 9 3
```

Per creare un vettore x che contiene i primi 7 numeri naturali, utilizziamo la seguente istruzione:

```
> x<-1:7
> x
[1] 1 2 3 4 5 6 7
```

Una successione di numeri può anche essere creata col comando “seq”, che consente di specificare anche il passo o il numero n (length) degli elementi della successione³:

```
> x<-seq(1, 7)
> x
[1] 1 2 3 4 5 6 7
```

```
> x<-seq(-2, 8, 2)
> x
[1] -2 0 2 4 6 8
```

```
> y<-seq(0, 1, length = 11)
> y
[1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

1.2.2. *Matrici*

Costruiamo un vettore z che contiene i primi 20 numeri naturali:

```
> z<-1:20
> z
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

Volendo creare una matrice A , che contiene gli stessi elementi del vettore z , organizzati in 5 righe e 4 colonne, il comando da utilizzare è “matrix”:

```
> A<-matrix(z,5, 4)
> A
```

3. Ricordiamo che il passo di una successione è dato da: $(x_{(n)} - x_{(1)})/(n - 1)$, dove $x_{(1)}$ ed $x_{(n)}$ sono i valori estremi della serie ordinata.