R - Esercitazione 1

Lorenzo Di Biagio
dibiagio@mat.uniroma3.it

Università Roma Tre

30 Settembre 2013

Introduzione a R

R è un software open-source, per Linux , Mac OS X, Windows, distribuito secondo la licenza *GNU - GPL*.

Essendo libero è largamente utilizzato e in continua evoluzione grazie al contributo di migliaia di ricercatori e sviluppatori in tutto il mondo.

Si scarica dal CRAN collegandosi a

`http://www.r-project.org`

e scegliendo un mirror: per Linux , Mac OS X, Windows si possono scaricare versioni direttamente installabili.

Introduzione a RStudio

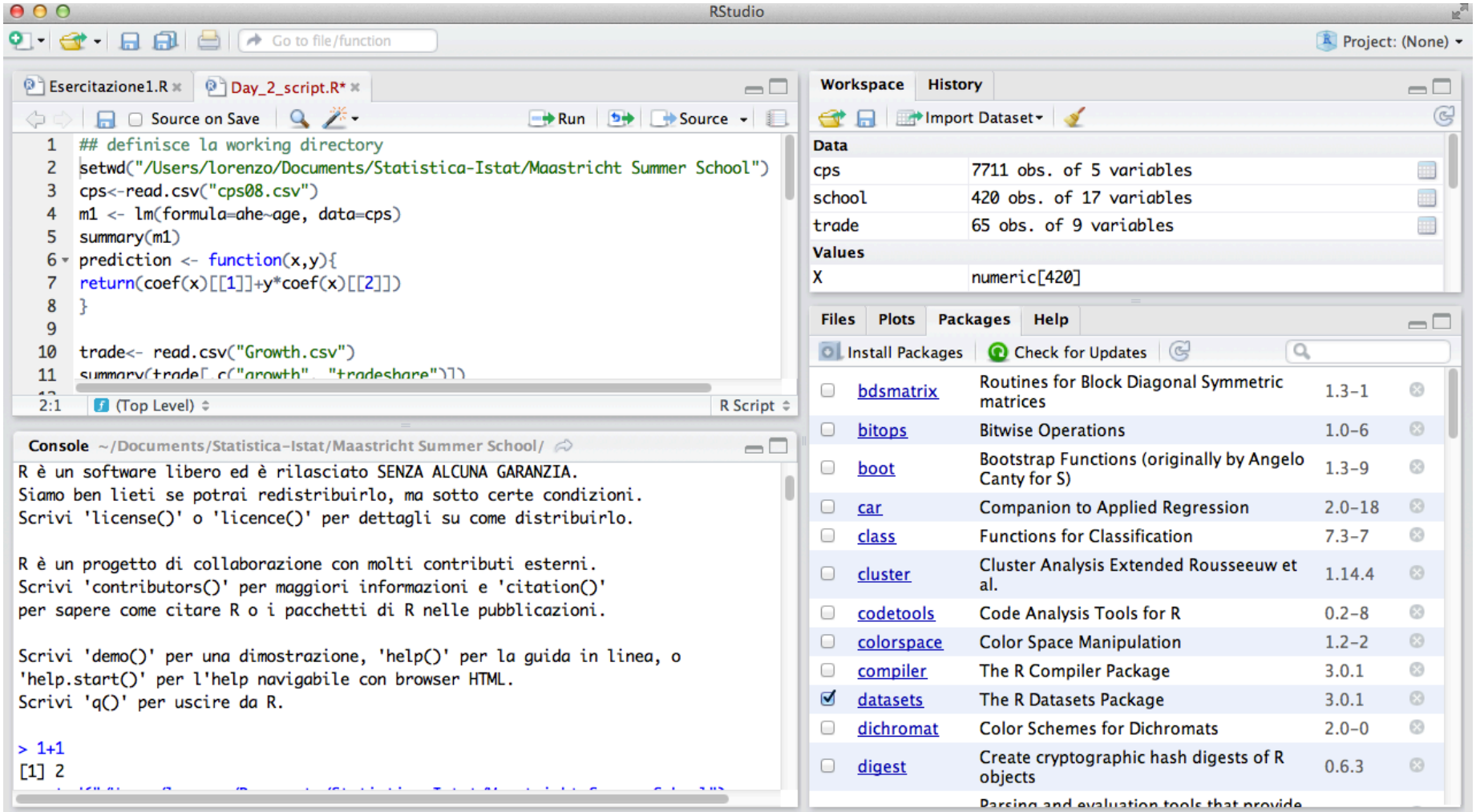
RStudio è un ambiente di sviluppo integrato (IDE) per R. È un software open-source che gira su Linux, Mac OS X e Windows.

RStudio si pu scaricare da

`http://www.rstudio.com`

L'ambiente di lavoro di RStudio è costituito da quattro finestre:

1. la finestra del codice (scrivere-eseguire script);
2. la finestra della console (riga di comando - output);
3. la finestra degli oggetti (elenco oggetti-cronologia dei comandi);
4. la finestra dei pacchetti-dei grafici-dell'aiuto in linea.


RStudio
Go to file/function
Project: (None)
Esercitazione1.R
Day_2_script.R*
Source on Save
Run
Source
1 ## definisce la working directory
2 setwd("/Users/lorenzo/Documents/Statistica-Istat/Maastricht Summer School")
3 cps<-read.csv("cps08.csv")
4 m1 <- lm(formula=ahe~age, data=cps)
5 summary(m1)
6 prediction <- function(x,y){
7 return(coef(x)[[1]]+y*coef(x)[[2]])
8 }
9
10 trade<- read.csv("Growth.csv")
11 summary(trade[,c("growth","tradeshare")])
2:1
(Top Level)
R Script
Console ~/Documents/Statistica-Istat/Maastricht Summer School/
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.
> 1+1
[1] 2
Workspace
History
Import Dataset
Data
cps 7711 obs. of 5 variables
school 420 obs. of 17 variables
trade 65 obs. of 9 variables
Values
X numeric[420]
Files
Plots
Packages
Help
Install Packages
Check for Updates
bdsmatrix Routines for Block Diagonal Symmetric matrices 1.3-1
bitops Bitwise Operations 1.0-6
boot Bootstrap Functions (originally by Angelo Canty for S) 1.3-9
car Companion to Applied Regression 2.0-18
class Functions for Classification 7.3-7
cluster Cluster Analysis Extended Rousseeuw et al. 1.14.4
codetools Code Analysis Tools for R 0.2-8
colorspace Color Space Manipulation 1.2-2
compiler The R Compiler Package 3.0.1
datasets The R Datasets Package 3.0.1
dichromat Color Schemes for Dichromats 2.0-0
digest Create cryptographic hash digests of R objects 0.6.3

Primi passi con R

I principali oggetti "atomici" di R sono:

1. numeri a precisione doppia (e.g.: 123)
2. numeri complessi (e.g.: 1+7i)
3. stringhe (e.g. "ciao")
4. valori logici (TRUE o FALSE)

Per assegnare un valore ad una variabile si usa l'operatore: <- oppure: =

Vi sono delle differenze tra i due operatori di assegnazione. Nella comunità di R si preferisce utilizzare <- e limitare l'uso di = per assegnare valori ai parametri di una funzione.

L'aiuto in linea

L'ambiente R dispone di un help in linea molto efficiente.

`help.start()` apre la pagina principale dell'help di R.

`help.search("parolachiave")` o `??parolachiave` cerca "parolachiave" nell'help.

`?funzione` o `help(funzione)` apre la pagina help del comando "funzione".

`?"operatore"` o `help("operatore")` apre la pagina help dell'operatore "operatore".

Sul CRAN sono disponibili numerose dispense e manuali di R, anche in italiano. Ad esempio:
`http://cran.r-project.org/doc/contrib/Mineo-dispensaR.pdf`

Collezioni di oggetti “atomici”

Gli oggetti “atomici” si possono raggruppare in:

1. vettori (elementi concatenati di un solo tipo)
2. matrici (vettori di vettori di uguale lunghezza)
3. fattori (collezione di dati categoriali)
4. data frames (insieme di vettori di uguale lunghezza ma eventualmente di tipi differenti)

Vettori (1)

Per creare un vettore con più di un elemento i dati vanno *concatenati* con la funzione c .

Vettori di sequenze di numeri si possono creare più velocemente con:

```
> x<-1:10
> x<-seq(1,100,10)
```

I singoli elementi di un vettore si estraggono con `[]`

Esercizio 1

1. Di che tipo è il vettore `x<-c("A", 1, TRUE)` ? E il vettore `x<-c(1,2,FALSE)` ?
2. Definire il vettore $x = (a, 1, b, 2, c, 3, \ldots, z, 21)$. (Utilizzare il vettore `letters` — Attenzione a "j", "k", "w", "x", "y").

Vettori (2)

Esercizio 2

Sia $x = (7, 9, 15)$ il vettore delle realizzazioni campionarie di un campione casuale di ampiezza 3. Si calcolino la media campionaria, il secondo momento campionario e la varianza campionaria di x.

Attenzione:

- Le operazioni elementari tra vettori sono svolte componente per componente.
- Si utilizzano differenti operatori per l'algebra vettoriale.
- In molte operazioni: se un vettore è troppo corto, R lo "ricicla" per renderlo di lunghezza uguale al vettore più lungo.

Fattori

I fattori immagazzinano dati categoriali come, ad esempio, “sì” e “no”; “maschio” e “femmina”; “insufficiente”, “sufficiente”, “buono”, “ottimo”.

I fattori si creano con il comando `factor` applicato a un vettore; si usa `ordered` per creare un fattore ordinato: se non è specificato l'ordine dei livelli viene usato l'ordine alfabetico.

Una semplice analisi delle frequenze di un fattore x si ottiene con:

> `table(x)` per le frequenze assolute.

> `table(x)/length(x)` per le frequenze relative.

> `pie(table(x))` per una rappr. grafica a torta.

> `barplot(table(x))` per una rappr. grafica con grafico a barre.

Data frames (1)

Un data frame è una matrice "generalizzata" in quanto può contenere allo stesso tempo vettori di tipo numerico, logico o fattore. Per questa sua caratteristica, il data frame è la struttura R più adatta per la memorizzazione e la gestione di data set.

Osserviamo 5 individui e registriamone sesso e età:

```
> x<-factor(c("M", "F", "M", "M", "F"))
> y<-c(29,40,23,62,60)
```

Definiamo il dataset delle nostre osservazioni:

```
data<-data.frame(sesso=x, eta=y)
```

Data frames (2)

Esercizio 3

1. Aggiungere la colonna istruzione = (13,16,18,21,11) a "data".
2. Aggiungere l'osservazione ("M",80,8) a "data".
3. Determinare il valore dell'età della quarta osservazione.
4. Calcolare la media delle età.
5. Calcolare la media degli anni di istruzione per sesso.

Data frames (3)

Esercizio 4

Aprire il database `http://people.stern.nyu.edu/wgreene/Text/Edition7/TableF4-3.csv` (alcuni dati su film usciti negli USA). Conservare solo le prime 5 variabili:

- Box: ricavi al botteghino (negli USA, in dollari).
- MPRating: classificazione della MPAA: 1=G, 2=PG, 3=PG13, 4=R.
- Budget: costi per la produzione (in milioni di dollari).
- Starpowr: valutazione complessiva degli attori che recitano nel film.
- Sequel: 1 se il film è un sequel, 0 se non lo è.

Studiare la struttura del database.

Data frames (4)

Per leggere un database si usa:
`> read.table(file, header = FALSE, sep = " ", dec = ".", skip=0, ...)`

dove "file" è il percorso (o l'url) tra virgolette ; "header" indica se la prima riga contiene il nome delle variabili o no (di default: no); "sep" indica il separatore dei dati (di default: lo spazio); "dec" indica il simbolo dei decimali (di default: il punto), "skip" indica il numero di righe da saltare (a partire dall'inizio) (di default: 0)

Se il file è *comma separated values* si può usare direttamente:
`>read.csv(file, header = TRUE, sep = ",", ...)`

comando identico al precedente, salvo per alcuni valori predefiniti.

Data frames (5)

Esercizio 4 - continua

Trasformare la variabile "BOX" in milioni di dollari; ricodificare la variabile "MPRATING" come un fattore ordinato.

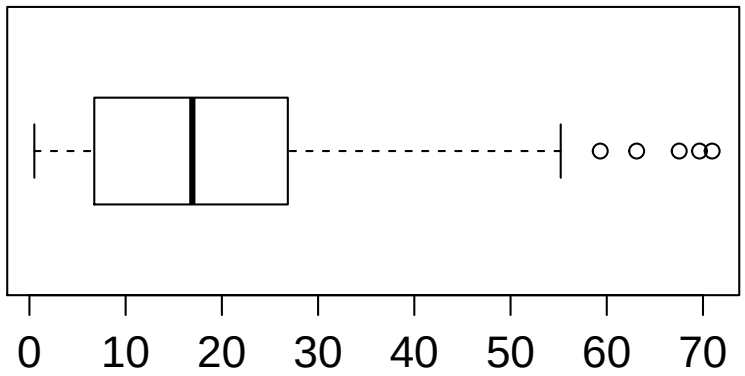
Studiare graficamente la distribuzione della variabile "BOX" attraverso un boxplot e un istogramma.

Un boxplot (o grafico a scatola) è uno strumento grafico di sintesi dei dati, che rende visivamente chiari semplici indici di posizione e di dispersione e l'eventuale asimmetria della distribuzione.

In R:

```
boxplot(...)
```

Data frames (5) - Boxplot

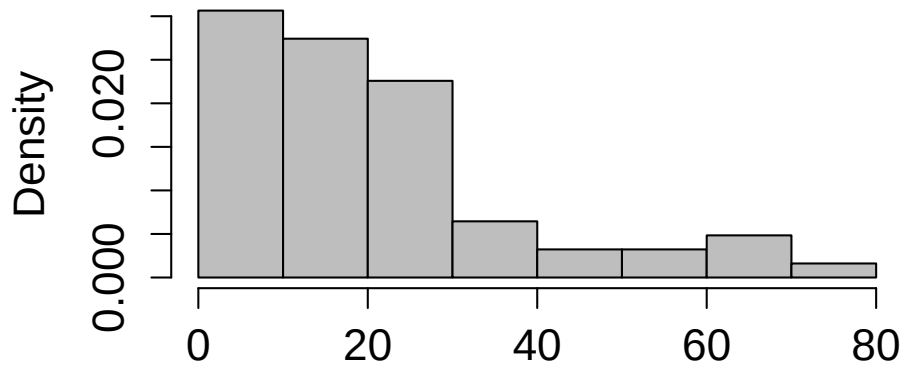


I bordi della scatola corrispondono al primo e terzo quartile.
All'interno è segnata la mediana.
Viene aggiunto un "baffo" sinistro (o inferiore) fino all'osservazione più piccola (se maggiore: sino a $\mathrm{Q1} - 1.5 \times (\mathrm{Q3} - \mathrm{Q1})$). Gli outliers sono segnalati a parte.
Analogamente per il baffo destro (o superiore).

Data frames (6) - Istogrammi

Un istogramma è una rappresentazione grafica di una distribuzione di frequenze di caratteri quantitativi (virtualmente) continui:
è costituito da rettangoli adiacenti;
ogni rettangolo ha base di lunghezza pari all'ampiezza della corrispondente classe; l'altezza invece è calcolata come densità di frequenza: è pari al rapporto fra la frequenza (relativa) associata alla classe e l'ampiezza della classe;
l'area della superficie di ogni rettangolo coincide con la frequenza (relativa) associata alla classe cui il rettangolo si riferisce;
l'area totale dell'istogramma è uguale 1.

Data frames (7) - Istogrammi



In R:
`hist(..., freq=T/F,breaks=c(...),...)` dove:
"freq"=FALSE garantisce che l'altezza di ogni rettangolo sia pari alla densità di probabilità di ogni classe (frequenza relativa di ogni classe/ampiezza della classe).
"breaks": serve a scegliere le classi; ponendolo uguale a un numero fissato si impone il numero delle classi (tutte di uguale ampiezza), ponendolo uguale a un vettore si impongono gli estremi delle classi. Valore predefinito: numero di classi suggerito dalla funzione `nclass.Sturges()`

Salvataggio

Prima di chiudere la sessione potrebbe essere necessario salvare il proprio lavoro.

Si consiglia di definire nello script la directory di lavoro con `setwd("percorso")`

Con RStudio si possono facilmente salvare (e riaprire):

1. gli script (.R)
2. gli oggetti dell'ambiente di lavoro (.RData)
3. la cronologia dei comandi (.Rhistory)