Matematica e Statistica con R

Federico Comoglio e Maurizio Rinaldi

23 ottobre 2015

Indice

Prefazione

Queste dispense devono la loro esistenza al corso di R che ormai da alcuni anni accompagna il corso di Matematica e Statistica. Esse seguono i principali temi del corso e fanno uso consistente di esempi. La maggior parte degli esempi sono stati pensati da noi, ma non rivendichiamo che essi siano esclusivi o abbiano una rilevanza particolare.
Inoltre, numerosi punti sono ancora carenti di adeguate referenze, che verranno aggiunte in maniera esauriente nelle prossime versioni.
Queste dispense sono state realizzate in LaTeX, con praticamente tutte le figure generate *on the fly* in R e incluse nel testo utilizzando `Sweave`.
Nonostante abbiamo speso tempo nel pensare ai concetti da includere ed organizzarne il contenuto, ci assumiamo le nostre responsabilità per errori che sono ancora presenti nel testo, figure o codice. A questo proposito ma non solo, ogni vostro suggerimento è ben accetto e potenzialmente molto prezioso.

Grazie,
Novara, 23 ottobre 2015
Maurizio e Federico

Capitolo 1

Introduzione ad R

R è un programma *freeware* (di distribuzione gratuita) e *open-source* (il cui codice sorgente può essere modificato e redistribuito) distribuito dalla "R foundation", utilizzato per il calcolo statistico ed in grado di svolgere calcoli matematici ed algebrici (sicuramente di livello superiore ad una normale calcolatrice tascabile) ma che non rientrano nello scopo primario di R. La versione cui sono riferite queste dispense è la 3 2.2. Assumiamo che lo studioso abbia scaricato dal sito `http://www.r-project.org/` tale versione di R e la abbia installata sul suo computer.

1.1 Introduzione

Il programma si presenta all'utente con una interfaccia per l'inserimento dei dati, una finestra chiamata "R console". Per esempio digitando a console (dopo il *prompt* >) il comando

```
> 1+1
```

e premendo il tasto di invio appare come risposta

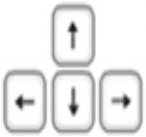
```
[1] 2
```

Oltre alla console R dispone di finestre di altra natura, per esempio finestre grafiche e finestre di *help*.

1.2 R basic

Iniziamo a descrivere alcune peculiarità di R.
L'insieme dei comandi eseguiti costituisce la storia o *cronologia* (*history*) di una sessione di lavoro. È possibile muoversi all'interno della cronologia, cioè richiamare i comandi precedenti e successivi utilizzando le frecce ↑ e ↓ della tastiera.

Possiamo scrivere diversi comandi prima di eseguirli (tasto di invio): basta separarli con un ";".
Si ricordi sempre che R è *c*ase sensitive ossia vi è differenza tra lettere minuscole e maiuscole. Questo è particolarmente importante in quanto ad esempio le variabili `a` ed `A` rappresentano entità distinte.



Help e guida on-line

Una qualunque richiesta di informazioni può essere effettuata digitando il comando: ?*argomento* (o `help`(*argomento*)). Ad esempio ?`sin` consente di ottenere informazioni sulla funzione seno e altre funzioni trigonometriche in quanto la guida è completamente indicizzata. È anche possibile utilizzare il menu a tendina (*help*), con modalità diverse a seconda del sistema operativo.
Infine, il comando `help.start()` consente di aprire l'*help* HTML di R da cui è poi possibile consultare la documentazione.

Inserire i commenti

I commenti sono frasi, annotazioni o codice da non eseguire momentaneamente molto utilizzati in programmazione. L'introduzione di un commento in R è possibile mediante il simbolo `#` (cancelletto singolo). Per evitare che un commento venga cancellato durante il normale *editing* del testo e che esso venga eseguito si consiglia di raddoppiare i simboli (`##`). Ad esempio:

```
> 1+1 # e  uguale a 2
[1] 2
```

non produrrà errore perché riconoscerà la stringa scritta dopo `#` come commento.

Quando un commento è utile e quando è superfluo?

A prescindere dalle preferenze personali, un commento dovrebbe essere inserito quando la porzione di codice che si sta considerando risulta a noi non ovvia, quando le dimensioni, tipo e ruolo delle variabili coinvolte non sono ovvie o semplicemente quando si ha bisogno di introdurre una annotazione che renda più facile la lettura del codice non solo a noi stessi ma anche ad altri. Quest'ultimo punto è di fondamentale importanza quando si vuole condividere il codice con altri membri della comunità scientifica. Per capire se i commenti che inserite sono superflui o mancanti:

> Ad un certo punto del corso salvate una porzione di codice (o una funzione) commentata in un file `.R`, lunga almeno 25 righe di codice. Aggiungete la data corrente al nome del file e in un periodo compreso tra 6 e 12 mesi semplicemente aprite il file e rileggetene il codice. Se siete in grado di capirne il contenuto, avete probabilmente fatto buon uso dei commenti.

Working directory

Il comando

```
> getwd()
```

indica la directory di lavoro mentre il comando

```
> setwd("indirizzo")
```

imposta la *working directory* nella posizione specificata da `"indirizzo"`. Lavorando con RStudio per impostare la directory si può usare il menu a tendina `Session` e in particolare posizionare la working directory in corrispondenza al file di codice sorgente.

Pacchetti

1.2.1 Pacchetti

Oltre all'aggiornamento costante del programma, lo sviluppo di R consiste nello creazione e aggiornamento continui da parte della comunità mondiale di una serie di pacchetti addizionali. Un pacchetto contiene in particolare un insieme di comandi finalizzati alla realizzazione di un determinato compito: esistono per esempio pacchetti per la risoluzione di equazioni differenziali, pacchetti per l'analisi statistica di *microarray*, pacchetti per il *clustering* di dati. Per non appesantire R i pacchetti (con poche eccezioni) vengono caricati solo se richiesto dalle esigenze dell'utente. L'utente deve quindi inizialmente localizzare (per esempio via web) il pacchetto che lo interessa e scaricarlo localmente. Per fare questo deve selezionare il sito CRAN da cui scaricarlo e poi (direttamente dal menu a tendina di R) scaricarlo (con annesse eventuali *dependencies*). Una volta scaricato in locale il pacchetto può essere messo a disposizione dell'utente con il comando

```
> library("nome.pacchetto")
```

Editori esterni

E' molto comodo scrivere le porzioni di codice che si intendono eseguire in R su un file di testo esterno in modo da disporre a fine sessione di un listato dei comandi usati (con eventuali commenti), scevro da errori e da operazioni superflue o ripetute. Gli editor esterni hanno in particolare la proprietà di

- Riconoscere la sintassi di R
- Interagire con R, consentendo l'esecuzione di comandi direttamente dall'editor.

Ne menzioniamo quattro di semplice utilizzo: Rstudio, Tinn-R, Komodo Edit e TextWrangler, i cui dettagli sono presenti in tabella 1.1.
A parte Tinn-R e RStudio, gli altri due editori necessitano di estensioni (*plug-in*) dedicate

Editor	Sistema Operativo	Website
RStudio	Mutlipiattaforma	`http://www.Rstudio.org`
Tinn-R	Windows	`http://www.sciviews.org/Tinn-R`
Komodo Edit	Multipiattaforma	`http://www.activestate.com/komodo-edit`
TextWrangler	Mac OSX	`http://www.barebones.com/products/textwrangler/`

Tabella 1.1: Alcune informazioni sugli editor citati nel testo (IDE)

per riconoscere la sintassi di R, segnalare errori, fornire suggerimenti ed eseguire comandi in R direttamente dall'editor. Il *plug-in* per Komodo Edit si chiama SciViews-K (`http://www.sciviews.org`), mentre il *plug-in* per TextWrangler è scaricabile dal sito `http://www.smalltime.com/gene/R.plist`. Quest'ultimo deve essere poi inserito nella directory (librerie globali): `~/Library/Application Support/TextWrangler/Language Modules/`.
A livello di sviluppo sono invece disponibili editori dedicati, che forniscono numerose funzioni di diagnostica e *debug* del codice. Senza qui addentrarci nei dettagli, Emacs (`http://www.gnu.org/software/emacs/`) con il *plug-in* ESS (Emacs *Speak Statistics*, `http://ess.r-project.org/`) ha trovato negli ultimi anni largo impiego anche come editor per R.

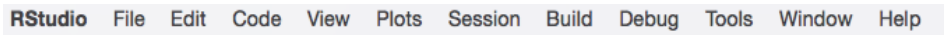


Figura 1.1: RStudio menubar

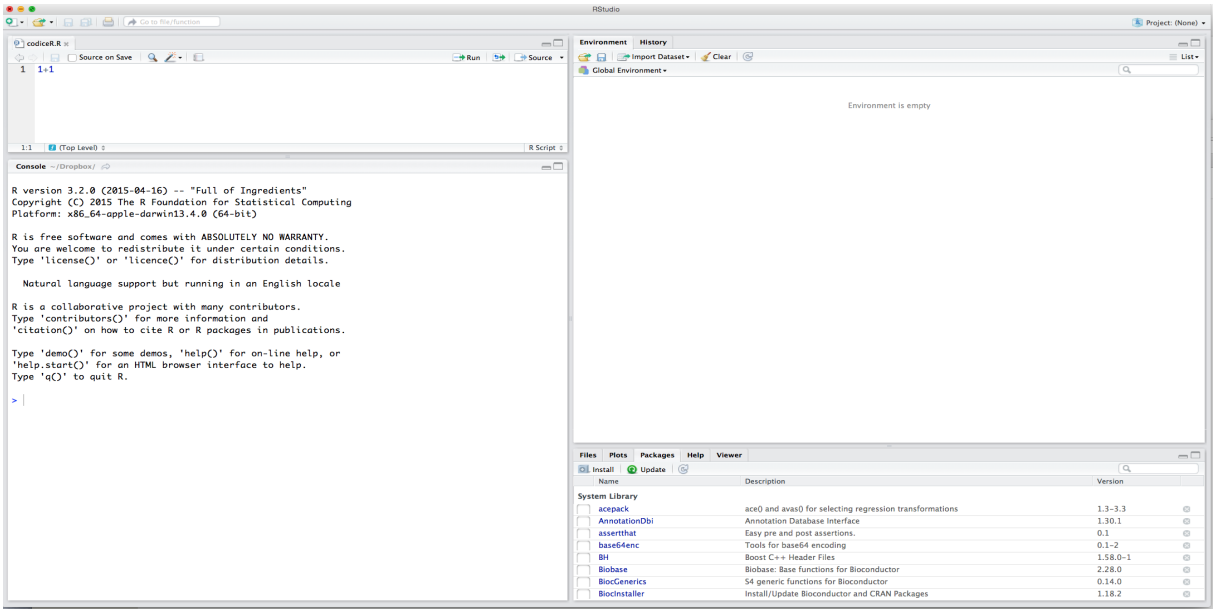


Figura 1.2: RStudio Screenshot

Un discorso più esteso spetta ad RStudio che presenta un ambiente di sviluppo integrato particolarmente adatto a fini didattici e come già detto è *free* e multipiattaforma.

Adotteremo in questo libro RStudio per la facilità d'uso e per il fatto che consente una visualizzazione completa contestuale di diversi aspetti di una sessione di lavoro. Se si lavora all'interno di RStudio il caricamento dei pacchetti è più semplice: in uno dei pannelli è presente il menu `Packages` dal quale l'utente può scaricare il pacchetto usando il sottomenu `Install` e caricarlo all'interno di R semplicemente barrandone la casellina relativa.

1.3 Rstudio

All'apertura di RStudio appaiono diversi pannelli.

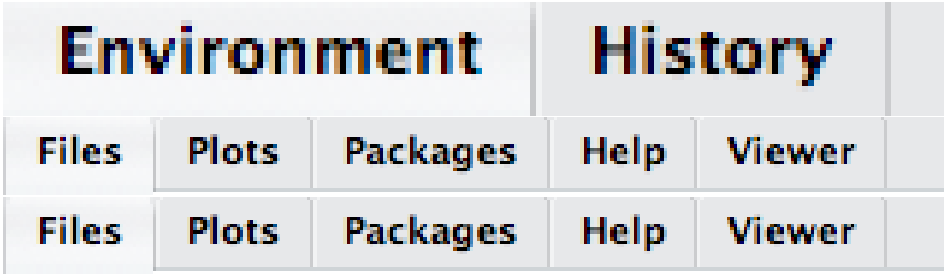


Figura 1.3: RStudio menubar

Il pannello fondamentale che appare aprendo RStudio è la classica console di R. Ci sono poi 2 altre finestre (bipartite) una con 2 linguette `Environment` e `History` la seconda con 5 linguette mostrate nella figura che segue.

1.3.1 Script di codice in R e RStudio

Ad esempio deve essere possibile inviare ad R porzioni di codice per la loro esecuzione R stesso include un editor (basta selezionare New document nel menu a tendina presente sotto File, in Rstudio invece sotto FIle basta selezionare NewScript.