

Capitolo 1

R come (super-)calcolatrice

1.1 Operazioni elementari

Lanciando il programma R¹ compare sul *desktop* un'interfaccia grafica con al suo interno una *console*, ovvero la finestra dalla quale si danno i comandi, come mostrato in figura 1.1. Il simbolo '>' nell'ultima riga della console è il cosiddetto *prompt*, ovvero il carattere che R visualizza per far capire che è pronto a ricevere comandi. Ora sta a noi. Cominciamo con qualcosa di banale, come '2 + 2'. Basta digitare tale espressione, dare Invio e il programma ci fornisce l'atteso risultato:

```
> 2+2
```

```
[1] 4
```

ove l'1 fra parentesi quadre indica che il risultato è costituito di un solo elemento.

A questo punto abbiamo a disposizione una calcolatrice, come vedremo, assai potente e versatile. Seguiamo quindi con una carrellata di operazioni autoesplicative:

```
> 3*4
```

```
[1] 12
```

```
> 12-3
```

```
[1] 9
```

```
> 20/4
```

```
[1] 5
```

```
> 20/7
```

```
[1] 2.857143
```

```
> 3^2
```

```
[1] 9
```

```
> 3^3
```

```
[1] 27
```

```
> 2^10
```

```
[1] 1024
```

```
> 2*pi
```

```
[1] 6.283145
```

Dall'ultima operazione impariamo che R conosce *pi greco* e questo ci potrà far comodo. E, ovviamente, conosce tale fondamentale costante un po' meglio delle 6 cifre decimali con cui il risultato ci è stato mostrato. Approfittiamo quindi per imparare l'opzione che ci permette di cambiare il nu-

¹Per cominciare e tenendo conto del target di questo testo, assumiamo che il lettore stia utilizzando Windows.

Per quanto riguarda l'installazione, vedi ad esempio sul sito <http://www.roma1.infn.it/~dagos/RaScuola/>, che contiene anche ulteriori informazioni riguardanti questo testo, inclusi link, esempi e figure a colori (ove i colori sono importanti).

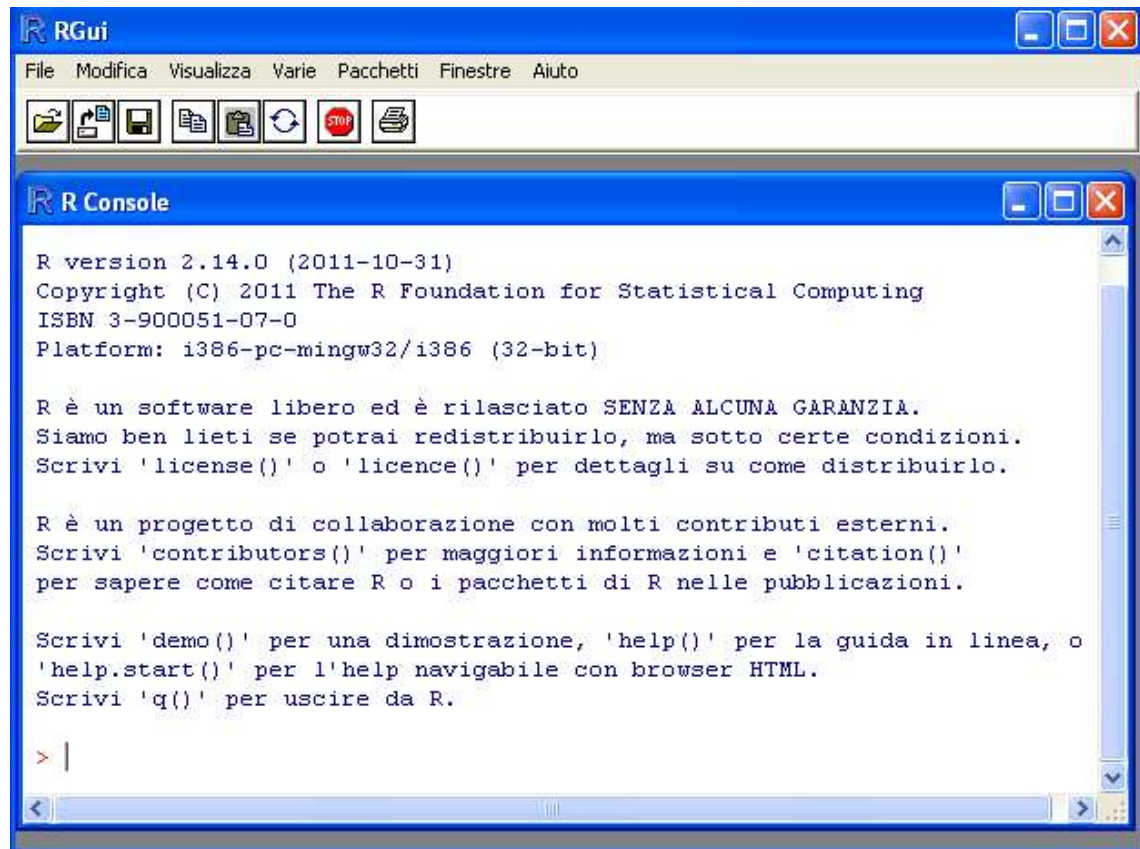


Figura 1.1: Interfaccia grafica ('GUI', ovvero *graphical user interface*, con all'interno la *console*).

mero di cifre con cui i risultati vengono *visualizzati*. Ad esempio se vogliamo π a 10 cifre (totali), diamo

```
> options(digits=10)
```

e controlliamo:

```
> pi
```

```
[1] 3.141592654
```

Risettiamo quindi il valore iniziale:²

```
> options(digits=7)
```

Approfittiamo per imparare un piccolo e utile trucco per non dover riscrivere l'intero comando. Mediante le frecce \uparrow e \downarrow della tastiera possiamo ripercorrere i comandi precedenti e riproporli, eventualmente modificati.

Un'altra importante operazione è l'estrazione di radice quadrata, per la quale non c'è un sim-

²Le opzioni vengono resettate quando si esce da R. In ogni caso, prima di cambiare un'opzione è bene conoscere, ed eventualmente memorizzare, l'opzione standard ('default').

Per avere la 'lista' (nel senso del paragrafo 2.3.3) delle opzioni:

```
> options()
```

Quindi, anticipando alcuni concetti che vedremo nel seguito, questi sono i comandi per salvare il valore di default dell'opzione `digits` e successivamente ripristinarlo:

```
> cifre.default <- options()$digits
```

```
.....
```

```
> options(digit=cifre.default)
```

bolo particolare (visto che le tastiere non hanno niente di mnemonicamente equivalente) ma dobbiamo usare un'apposita *funzione*:

```
> sqrt(25)
[1] 5
> sqrt(30)
[1] 5.477226
```

Gli stessi risultati potevano essere ottenuti elevando i numeri alla 1/2 (o alla 0.5). Usando questa ben nota proprietà delle potenze, possiamo ottenere le altre radici di interesse, ad esempio

```
> 25^(1/2)
[1] 5
> 30^0.5
[1] 5.477226
> 8^(1/3)
[1] 2.
```

Le parentesi dopo il simbolo di potenza sono importanti per far capire a R che vogliamo fare, ad esempio, 25 alla 1/2 e non 25 alla 1 e dividere il risultato per due, operazione che darebbe invece

```
> 25^1/2
[1] 12.5
```

Anche per il fattoriale,³ usualmente indicato con $n!$ bisogna far ricorso a una apposita funzione. Ad esempio,⁴

```
> factorial(4)
[1] 24
```

La lista delle funzioni matematiche elementari⁵ conosciute da R è mostrata in tabella 1.1.⁶

³Il fattoriale del numero naturale (ovvero intero positivo) n , è pari al prodotto di tutti i naturali da 1 a n . Ad esempio $4! = 1 \times 2 \times 3 \times 4 = 24$. Da cui segue la proprietà $n! = n \times (n-1)!$, estesa a tutti gli interi definendo per convenzione $0! = 1$, come possiamo verificare con R:

```
> factorial(0)
[1] 1
```

⁴Qualcuno si stupirà nello scoprire che R accetta anche fattoriali di numeri non interi:

```
> factorial(pi)
[1] 7.188083
```

In realtà R usa la funzione speciale *gamma*, per la quale sussiste, per n intero, la relazione $\Gamma(n+1) = n!$, come si può verificare con qualche esempio, come

```
> gamma(4 + 1)
[1] 24
```

o ancora più chiaramente mediante

```
> factorial
function (x)
gamma(x + 1)
```

da cui impariamo che `factorial(x)` non è altro che `gamma(x+1)` – in R se immettiamo una funzione senza parentesi otteniamo la funzione stessa o comunque informazioni sulla funzione.

⁵Si presti attenzione al fatto che l'aggettivo 'significativo' è usato in questa funzione con un'accezione diversa dal quella che si intende in fisica e nelle altre scienze sperimentali, la quale peraltro non sarebbe implementabile in una funzione senza ulteriori informazioni. Questa funzione considera soltanto se le cifre sono 'matematicamente' rilevanti e quindi ad esempio `signif(pi,3)` dà 3.14 e `signif(5.01,3)` dà 5.01, mentre `signif(5.00,3)` dà 5, benché nelle scienze sperimentali scrivere $r = 5$ cm sia ben diverso da scrivere $r = 5.00$ cm in quanto nel secondo caso l'informazione è molto più precisa del primo caso. Insomma, nel preparare una relazione, non 'filtrate' i risultati con `signif`, pretendendo poi che le cifre ottenute siano quelle veramente 'significative' perché "l'ha detto R".

⁶Il lettore non si spaventi se qualcuna non gli è nota. Come detto nella prefazione, questo è un testo *multilevel* ed è quindi normale che alcune cose non siano ancora state incontrate. Si prenda questa tabella come quella in cui vengono descritte le funzioni di una calcolatrice scientifica.

$+$, $-$, $*$, $/$ x^y $\%/\%$ $n\%m$ $\%*\%$ $>$, $<$, $>=$, $<=$, $==$, $!$, $\&$, $\&\&$, $ $, $ $	operazioni aritmetiche elementari elevamento a potenza, x^y parte intera della divisione (es. $16\%/\% 3 = 5$) n modulo m (abbr. ‘ $n \bmod m$ ’) (es. $17\% 5 = 2$) prodotto fra matrici operatori logici
$\text{sqrt}(x)$ $\text{factorial}(x)$ $\text{abs}(x)$ $\text{sin}(x)$, $\text{cos}(x)$, $\text{tan}(x)$ $\text{asin}(x)$, $\text{acos}(x)$, $\text{atan}(x)$ $\text{atan}(x,y)$ $\text{sinh}(x)$, $\text{cosh}(x)$, $\text{tanh}(x)$ $\text{asinh}(x)$, $\text{acosh}(x)$, $\text{atanh}(x)$ $\text{exp}(x)$ $\text{log}(x)$ $\text{log10}(x)$, $\text{log2}(x)$, $\text{log}(x,b)$ $\text{trunc}(x)$ $\text{round}(x,n)$ $\text{signif}(x,n)$ $\text{floor}(x,n)$ $\text{ceiling}(x,n)$	radice quadrata fattoriale valore assoluto (‘modulo’) funzioni trigonometriche (x in radianti) funzioni trigonometriche inverse angolo tra il vettore (x, y) e l’asse delle ordinate funzioni iperboliche funzioni iperboliche inverse esponenziale (e^x , o $\text{exp}(x)$) logaritmo in base e logaritmi in base 10, 2 e b tronca la parte decimale arrotonda a n cifre decimali arrotonda a n cifre ‘significative’ arrotonda all’intero più basso arrotonda all’intero più alto

Tabella 1.1: Operatori e funzioni matematiche elementari di R (si veda la nota 5 riguardo le ‘cifre significative’).

1.2 Variabili

Ora che abbiamo imparato ad usare R come calcolatrice, vediamo come si fa, per usare il gergo delle calcolatrici tascabili, a “mettere un numero in memoria”. Ad esempio, immaginiamo di essere interessati ad usare nel seguito i risultati di due operazioni, ad esempio ‘ $3+4*2$ ’ e ‘ $(3+4)*2$ ’. Potremmo chiamarli quindi ‘ris1’ e ‘ris2’ ed *assegnare* ad essi i risultati nel modo seguente:

```
> ris1 <- 3 + 4 * 2
> ris2 <- (3 + 4) * 2
```

Per mostrare il contenuto di una *variabile* è sufficiente dare come *comando* il suo nome, ad esempio

```
> ris1
[1] 11
> ris2
[1] 12
```

In realtà, R accetta per le assegnazioni anche il simbolo di uguale, anche se nel seguito l’operatore standard di R, ‘<-’, ha un significato più intuitivo ed inoltre può essere usato anche ‘verso destra’, con ‘->’ con significato immutato: “metti il risultato dell’operazione in questa variabile”.