

Importare i dati in R

In R, l'[inserimento diretto dei dati](#) in un [Dataframe](#) non è una operazione semplicissima. L'importazione dei dati diventa quindi il modo più efficiente per costruire un dataset, o per utilizzarne uno disponibile in altro formato.

R prevede diverse funzioni di importazione dei dati, anche con finalità diverse (ad es.: importare dati e tabelle dal web), ma la procedura più semplice da seguire è:

- Se il file è in Excel:
 - **Controllarlo** con Excel o Open Calc, e salvarlo in formato Excel (xlsx);
 - **Importare il file con RStudio**, o direttamente in R **con il pacchetto readxl**, o **con RCommander**.
- Con il copia e incolla:
 - Vai a [Importare i dati copiati da altri programmi](#)
- Se il file è in altri formati di testo (csv o txt):
 - se necessario, importarlo e **controllarlo** con Excel o Open Calc, e salvarlo in formato Excel (xlsx), oppure [in un altro formato adatto](#) all'importazione in R;
 - **Importare il file con la procedura più adatta al formato**.
- Se il file è in altri formati (es. SPSS o SAS):
 - **Importare il file con la procedura più adatta al formato**.

Vedi anche:

- [Importare tabella da file PDF](#)
- [Importare fogli Google](#)
- [importare i dati con protocollo SDMX](#) (Istat, Eurostat, OECD, ecc)
- [Importare documenti di testo](#)

Preparazione della base dati

Prima di importare i dati in un dataframe, bisogna prepararli, facendo attenzione a:

- la **riga di intestazione**, che deve contenere i nomi delle variabili, e deve essere una sola;
- l'appropriatezza dei **nomi delle variabili**;
- il **formato numerico** adottato ed eventualmente la **codifica del file** (impostazioni di lingua);
- che ci sia una variabile per gli **identificativi di caso** (la numerazione delle righe è relativa alla posizione, l'identificativo del caso deve essere univoco);
- **tipo di variabili**: numeriche e stringhe (vedi: [Variabili e vettori in R](#));
- trattamento dei **valori mancanti**: poiché R richiede un particolare trattamento dei valori mancanti, può essere utile, prima dell'importazione, distinguere fra:
 - mancate risposte = dati non disponibili, ovvero valori mancanti veri e proprio, da

codificare come NA;

- risposte non dovute = filtrate da domande precedenti.

Con Excel

- Tutti i dati devono essere in un singolo foglio di calcolo di un singolo file;
- Nel foglio non ci devono essere righe vuote;
- Se i dati sono raggruppati (come nell'esempio sotto), costruire una variabile che contenga l'informazione dell'appartenenza al gruppo

Esempio ¹⁾

Questa tabella NON va bene

Male		Female	
ID	Salary	ID	Salary
1	32000	2	23000
3	37000	4	54000

Va trasformata così

ID	Salary	Gender
1	32000	Male
2	23000	Female
3	37000	Male
4	54000	Female

Con Office Calc

Valgono le stesse indicazioni per Excel.

Formati dei files Excel e Calc

Una volta terminati i controlli sui file, i default delle procedure di esportazione dei file Excel e Calc possono essere utilizzati per salvare il dataset ed importarlo in R.

Sarà cioè possibile utilizzare il comando da menu “Salva con nome ...” e selezionare il formato di output, senza ulteriori modifiche.

Excel

In Excel è possibile salvare i files in vari formati. Oltre al formato Excel (xlsx), suggeriamo di utilizzare i seguenti:

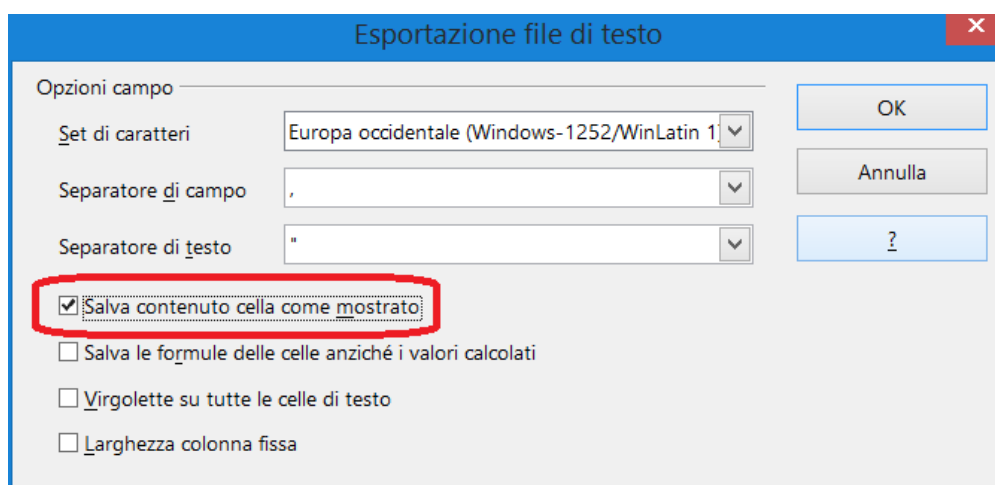
formato	sep. campo	sep. testo	sep. decimali	comando import.
txt (tab delimited)	tabulazione	nessuno	, (virgola)	read.delim2
csv	; (punto e virgola)	nessuno	, (virgola)	read.csv2

Attenzione a non confondere il formato **CSV** con il formato **CSV UTF-8**.

Calc

Anche in OpenOffice, è possibile salvare i files in vari formati, fra i quali sono da preferire il formato xlsx e quello csv (non esiste un filtro per l'importazione diretta dei formati open office):

formato	sep. campo	sep. testo	sep. decimali	comando import.
csv	, (virgola)	nessuno	come mostrato	read.csv



Esporta i dati da Calc

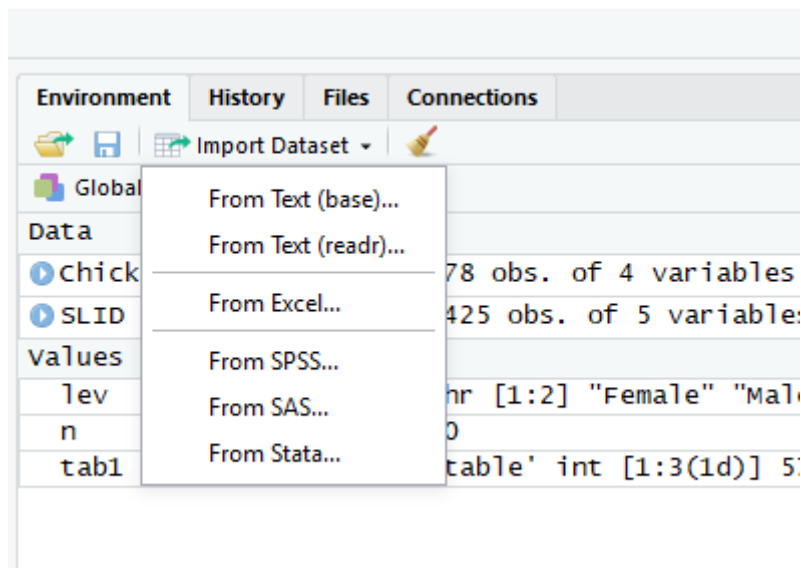
In questo caso, di default, il separatore dei decimali (e delle migliaia), è quello utilizzato nel foglio di calcolo stesso, quindi quello di sistema (in Italia, dovrebbe essere la virgola).

Importazione: procedure

Definiamo come [directory di lavoro](#) quella in cui collochiamo il file esportato, per maggiore comodità.

Con RStudio

RStudio prevede una semplice procedura di importazione dei dati da Excel e da altri formati. Le funzioni usate da RStudio sono sia quelle della distribuzione base, sia quelle del pacchetto *readr* (*tidyverse*).



Importazione con RStudio

In [questo video](#) viene mostrata ad esempio l'importazione di un file csv da url (YouTube).

Con RCommander

Vedi: [Importazione dei dati con RCommander](#)

Funzioni di R

Da Excel

Per importare i file di Excel in formato xls exlsx, è molto comodo e semplice da usare il pacchetto *readxl*²⁾:

[es_readxl.R](#)

```
# installare e caricare il pacchetto
install.packages("readxl")
library("readxl")
```

```
# importare il file
mydata <- read_excel("myfile.xlsx")

# oppure
mydata <- read_excel(file.choose())
```

Gli argomenti della funzione consentono di indicare il foglio di lavoro e il range delle celle da importare.

```
# argomenti di read_excel
read_excel(path, sheet = NULL, range = NULL, col_names = TRUE,
  col_types = NULL, na = "", trim_ws = TRUE, skip = 0, n_max = Inf,
  guess_max = min(1000, n_max))
```

Vedi il [sito web del pacchetto](#)

da csv

Se il file ha il **formato numerico italiano**, usiamo la funzione `read.csv2`:

```
# argomenti di read.csv2
read.csv2(file, header = TRUE, sep = ";", quote = "\"",
  dec = ",", fill = TRUE, comment.char = "", ...)
```

Di default, quindi, `read.csv2` legge i file con ";" come separatore di campo (`sep=";"`) e separatore di decimale "," (`dec = ","`). Digiteremo dunque il comando:

[es_csv2.R](#)

```
mydata <- read.csv2("myfile.csv")
```

Se il file usa un **formato numerico non italiano** (il punto come separatore di decimali), usiamo la funzione `read.csv`, che, di default, legge i file con "," come separatore di campo (`sep=","`) ma **separatore di decimale "."** (`dec = "."`). In teoria, però, nel nostro file dovremmo avere le virgole come separatori dei decimali.

Digiteremo dunque:

```
mydata <- read.csv("myfile.csv")
```

da formato di testo delimitato

I file in formato testo possono avere altri delimitatori di campo (spazio, tabulazione, ecc.). I file Excel esportati in formato **txt**, ad esempio usano la tabulazione (**tab delimited**). Se il file ha il **formato numerico italiano**, usiamo la funzione `read.delim2`:

```
# argomenti di read.delim2
read.delim2(file, header = TRUE, sep = "\t", quote = "\"",
            dec = ",", fill = TRUE, comment.char = "", ...)
```

Di default, la funzione legge i file con le tabulazioni come separatore di campo (`sep = "\t"`) e separatore di decimale `","` (`dec = ","`). Sarà quindi sufficiente digitare il comando:

[es_delim2.R](#)

```
mydata <- read.delim2("myfile.txt")
```

Queste funzioni fanno parte ed hanno gli stessi argomenti della funzione generica di importazione dei dati [read.table](#).

Copia e incolla

È possibile importare in R tabelle e dati copiati da un altro programma.

- Selezionare la tabella da copiare, dopo aver controllato che il formato sia adatto all'importazione;
- Copiare (da menu o tastiera)
- Scrivere le righe di comando:

```
library(readr)
nuovo_oggetto <- read_delim(clipboard())
```

Entrambe le funzioni `read_delim()` e `clipboard()` appartengono al pacchetto **readr**, usato anche da RStudio per l'importazione dei dati.

da Spss

La funzione più semplice da usare è `spss.get` nel pacchetto **Hmisc**.

```
# argomenti di spss.get
spss.get(file, lowernames=FALSE, datevars = NULL,
        use.value.labels = TRUE, to.data.frame = TRUE,
```

charfactor=FALSE)

datevars	variabili da convertire in date
use.value.labels	se TRUE (default), le etichette vengono trasformate in livelli dei fattori
to.data.frame	se TRUE (default), i dati vengono importati in un Dataframe
charfactor	se FALSE (default), le variabili di testo NON vengono convertite in fattori

Per altri argomenti, cfr. la guida di Hmisc.

Es.:

```
require(Hmisc)
mydata <- spss.get("mydata.sav",
datevars=c("datanascita", "datainterv"))
```

da SAS

```
# argomenti di sasxport.get
sasxport.get(file, lowernames=TRUE, force.single = TRUE,
             method=c('read.xport', 'dataload', 'csv'), formats=NULL,
allow=NULL,
             out=NULL, keep=NULL, drop=NULL, as.is=0.5, FUN=NULL)
```

Richiede il file di esportazione di SAS (.xpt).

Esempio:

```
require(Hmisc)
mydata <- sasxport.get("mydata.xpt")
```

Per altri argomenti e opzioni, cfr. l'help di Hmisc.

Dopo l'importazione

- Controllare ed eventualmente correggere il nome delle variabili (colnames) - vedi [Variabili e vettori: etichette](#);
- Controllare che sia presente una variabile ID per l'identificatore dei casi - vedi [sopra](#);
- Controllare ed eventualmente convertire il formato dei dati (fattori, fattori ordinati, date) - vedi [Ricodificare le variabili](#).

[Inserimento dei dati](#), [Importazione](#), [Gestione dei dati](#), [Excel](#), [Calc](#)

1)

tratto da [R4Stats](#)

2)

Ha. Wickham & J. Bryan (2018). readxl: Read Excel Files. R package version 1.1.0.
<https://CRAN.R-project.org/package=readxl>.

From:

<https://www.agnesevardanega.eu/wiki/> - **Ricerca Sociale con R**

Permanent link:

https://www.agnesevardanega.eu/wiki/r/gestione_dei_dati/importare_i_dati

Last update: **05/09/2025 16:53**

