



UNIVERSITÀ
DEGLI STUDI DELLA
TUSCIA

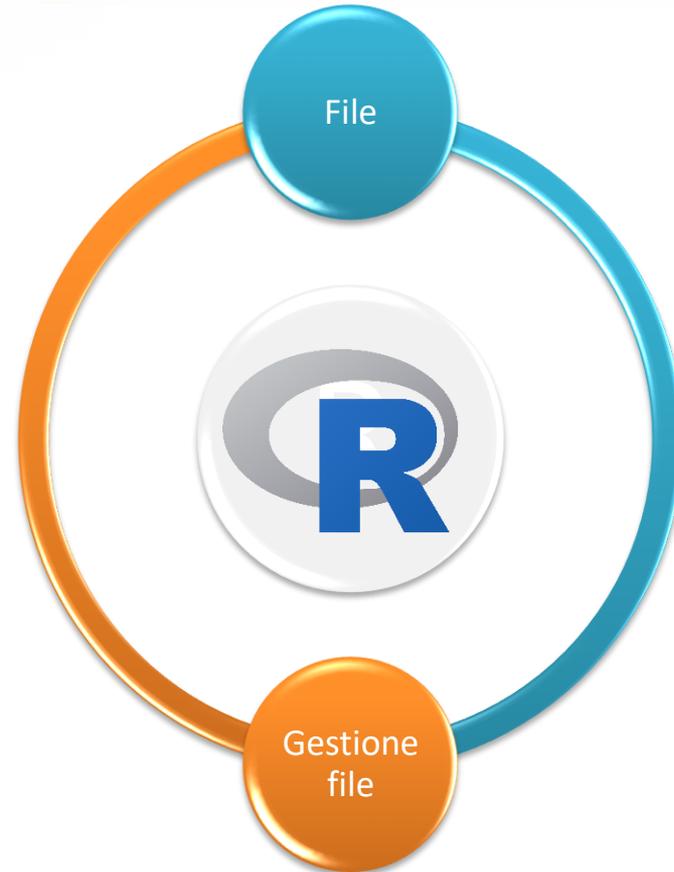
INFORMATICA

Linguaggio R
File

Dott. Franco Liberati
franco.liberati@unitus.it

LINGUAGGIO R

Argomenti del corso





LINGUAGGIO R

FILE

❑ Un documento digitale (**file**) è un informazione digitale costituita da parole binarie che possono assumere due connotazioni:

testuale o binaria

- ❑ In un documento testuale i byte sono interpretati come singoli caratteri alfabetici e si ha una rappresentazione in accordo alla codifica utilizzata (ASCII, ISO 8859 e EBCDIC)
- ❑ In un documento binario ogni record è un numero che ha un significato specifico (l'altezza di una immagine; l'intensità di una nota, le coordinate geografiche di un punto di un oggetto tridimensionale).



LINGUAGGIO R

FILE

- Un file è caratterizzato da un **nome** e da un **formato**: il nome identifica il documento (es.: DivinaCommedia.txt); mentre il formato definisce la modalità di rappresentazione, organizzazione e significato delle parole che lo compongono
- Il file ha degli **attributi** (dimensione, data creazione, data ultimo accesso) che sono gestiti dal file system

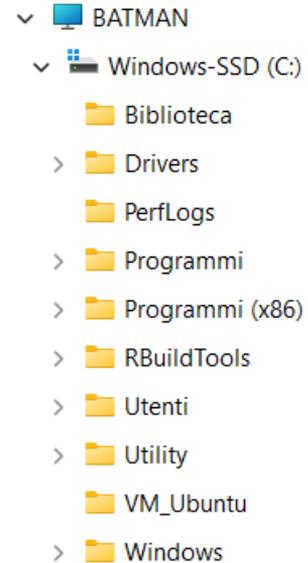
LINGUAGGIO R

FILE

□ Ogni file che risiede in maniera permanente su un supporto di memorizzazione è discriminato dalla posizione cioè il **percorso** (*path*) che ha inizio con l'identificativo del supporto di memorizzazione (di solito una lettera) fino al documento digitale intervallato

□ In Windows si ha \\ dopo l'etichetta che specifica il supporto di memorizzazione in cui risiede il file) e / per ogni cartella che lo contiene (es.:

C:\\Users\\flibe\\Documents\\Esercizi\\LIBRO\\Script_I_1.r)





LINGUAGGIO R

FILE

□ Un file può essere:

1. **creato** (create);
2. **aperto ed analizzato**, attraverso l'operazione di lettura (open/read);
3. **aperto e modificato**, mediante la funzione di scrittura (open/write);
4. **aperto ed incrementato** nei contenuti inserendo nuove informazioni alla fine dello stesso file (open/append);
5. **chiuso** (close)



File testuale



LINGUAGGIO R

FILE TESTUALE: generalità

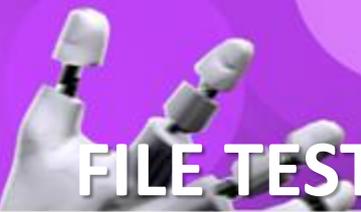
- ❑ Un documento testuale è costituito da una sequenza di parole la cui codifica segue, nella definizione canonica, lo standard ASCII per cui ogni parola è formata da 8bit, un byte, e ad ogni valore da 0 a 255 si associa una lettera, un simbolo o un carattere speciale (come il ritorno a capo e il tabulatore)



LINGUAGGIO R

FILE TESTUALE: funzioni di gestione di file in R

- ❑ Nel Linguaggio R la **gestione di documenti testuali** avviene sfruttando tre funzioni principali che consentono la creazione, la lettura o la scrittura di informazioni e la chiusura del file



LINGUAGGIO R

FILE TESTUALE: funzione di creazione e accesso

Per definire il file da creare o da gestire si ricorre alla funzione `file()` presente nel pacchetto base che restituisce un connettore (**handle**), cioè un oggetto che mantiene il collegamento a un'area di memoria in cui risiede il documento digitale:

```
file(description="", open="", blocking=TRUE, encoding=getOption("encoding"),  
raw=FALSE, method=getOption("url.method", "default"))
```

dove i campi fondamentali sono `description`, in cui specificare il percorso del file e il suo nome, e `open`, che indica il modo in cui deve essere gestito il file



LINGUAGGIO R

FILE TESTUALE: modi di accesso

Modalità di apertura di un file in modalità testuale

"r" "rt"

Lettura in modalità testuale

"w" "wt"

Scrittura in modalità testuale

"a" "at"

Aggiunta, a partire dall'ultimo carattere immesso, in un file testuale

"w+"

Lettura e scrittura di un file testuale



LINGUAGGIO R

FILE TESTUALE: creazione

La funzione di creazione avviene con la semplice specifica del percorso e del nome del file e il modo **w**

ESEMPIO

```
#Definizione del documento. Si usa un percorso relativo, ovvero si scrive nella cartella di lavoro
new_file <- "DivinaCommedia.txt"
#Verifica se nella cartella di lavoro esiste un documento con lo stesso nome
if (file.exists(new_file)) {
#Messaggio di avvertimento e terminazione del programma (non si crea il file)
cat("Il file già esiste")
} else {
#Creazione del file con sovrascrittura nel caso il file esista e restituzione del connettore
(file_handle)
file_handle<-file(new_file, "w")
#Chiusura del file creato
close(file_handle)
}
```



LINGUAGGIO R

FILE TESTUALE: scrittura

La **scrittura** di informazioni testuali all'interno di un file avviene sfruttando la funzione **cat()** dopo aver ottenuto il connettore e imposto il modo a **w**

La scrittura avviene in maniera sequenziale dopo ogni richiamo della funzione; l'andata a capo deve essere esplicitata dal carattere `\n`.

ESEMPIO

```
#Definizione del nome e del path del file
file_input <- "C:\\Letteratura/Italia//Dante/Inferno/Inferno1.txt«

#Restituzione connettore
file_handle <- file(file_input, "w")

#Scrittura con gli argomenti: testo da scrivere (con andata a capo) e connettore
cat("Nel mezzo del cammin di nostra vita\n", file = file_handle)

#Scrittura con gli argomenti: testo da scrivere (con andata a capo) e connettore
cat("mi ritrovai per una selva oscura\n", file = file_handle)

#Chiusura del file
close(file_handle)
```



LINGUAGGIO R

FILE TESTUALE: lettura

Il **prelievo** di informazioni digitali da un documento testuale richiede il connettore con il modo di lettura R e l'uso della funzione **readLines()** che estrapola dati fino al simbolo di fine file e li separa in stringhe per ogni ritorno a capo

ESEMPIO

```
#Definizione del nome e del path del file
file_input <- "C:\\Letteratura/Italia//Dante/Inferno/Inferno1.txt«

#Restituzione connettore
file_handle <- file(file_input, "r")

#Lettura del file (si parte dalla prima riga e si legge solamente la prima riga impostando n=1)
linea_output<-readLines(file_handle, n=1)

#Chiusura del file
close(file_handle)

print(linea_output)
[1] "Nel mezzo del cammin di nostra vita "
```



LINGUAGGIO R

FILE TESTUALE: modifica

La lettura e la scrittura di un documento testuale avviene sfruttando il modo w+

Il posizionamento ad un carattere nel testo si ottiene con la funzione:

```
seek(con, where = NA, origin = "start", rw = "", ...)
```

con: l'handle di connessione

where: un valore numerica che esplicità la posizione nella quale si vuole operare (la conta parte per default dall'origine del file)

rw: assume i valori "read" o "write" per eslicitare se si vuole leggere o scrivere dati

origin: assume i valori "start", "current", "end" per esplicitare il punto di partenza per il posizionamento (where)



LINGUAGGIO R

FILE TESTUALE: modifica

La lettura e la scrittura di un documento testuale avviene sfruttando il modo **w+**

Il posizionamento ad un carattere nel testo si ottiene con la funzione **seek**

ESEMPIO

```
#Definizione del nome e del path del file
file_input <- "C:\\Letteratura\\Italia\\Dante\\Inferno\\Inferno2.txt"
#Restituzione connettore
file_handle <- file(file_input, "w+")

#Scrittura con gli argomenti: testo da scrivere (con andata a capo) e connettore
cat("Lo giorno se n'andava, e l'aria bruno \n", file=file_handle)
cat("toglieva li animai che sono in terra \n", file=file_handle)
cat("da le fatiche loro; e io sol uno \n", file=file_handle)

# Posizionamento al quattordicesimo carattere a partire dall'inizio
seek(file_handle, 27, rw = "w")
#Modifica della parola non corretta (sita alla posizione 27)
cat("aere", file=file_handle)

#Posizionamento all'inizio
seek(file_handle, 0, rw = "r")
prima_riga<-readLines(file_handle, n=1)

#Chiusura del file
close(file_handle)

print(prima_riga)
[1] "Lo giorno se n'andava, e l'aere bruno "
```



LINGUAGGIO R

FILE TESTUALE: lettura

Una ultima funzione interessante è `shell.exec(path_nomefile)` che attiva il programma di default per la visualizzazione del file

Ad esempio

`shell.exec("C:\\Letteratura/Italia//Dante/Inferno/Inferno2.txt")` in Windows attiva l'apertura dell'applicativo di default (impostato dall'utente come ad esempio Notepad)



File testuale CSV



LINGUAGGIO R

FILE TESTUALE CSV: generalità

Un documento digitale in formato **Comma-Separated Values** (valori separati da virgola) è un documento testuale usato per rappresentare dati in forma tabellare

È un formato impiegato per l'interscambio dei dati tra sistemi informativi eterogenei (es.: da excel a calc)

Formalmente un documento digitale CSV è un file di testo, che utilizza un separatore (comunemente il simbolo di virgola) per distinguere i dati contenuti all'interno delle singole celle di una tabella

LINGUAGGIO R

FILE TESTUALE CSV: esempio

Un esempio è una scheda degli impiegati ad uso delle risorse umane di una azienda

IMPIEGATI.CSV

```
Id, Nome, Salario, Data_Contratto, Reparto
1, Franco, 623.3, 2012-01-01, ITC
2, Davide, 515.2, 2013-09-23, Operations
3, Fabio, 611, 2014-11-15, ITC
4, Gianmarco, 729.1, 2014-05-11, Sicurezza
5, Cristiano, 843.25, 2015-03-27, Sicurezza
6, Dalia, 578, 2013-05-21, ITC
7, Francesca, 632.8, 2013-07-30, Personale
8, Alessandra, 722.5, 2014-06-17, Contabilità
```

In alcuni documenti digitali CSV non sempre si rispetta la convenzione del separatore virgola ma si usa il carattere del punto e virgola (;) nonché spazi o tabulazioni (in quest'ultimo caso si parla di formato TSV). Queste differenze sono tutte derivanti dall'assenza di uno standard eccetto per un RFC4180 che definisce una serie di prassi comuni, più o meno consolidate (tra cui l'uso dei doppi apici per riportare le stringhe).



LINGUAGGIO R

FILE TESTUALE CSV: creazione

Per creare un documento digitale CSV si può sfruttare un qualsiasi editor di testo. Una volta redatti i dati, opportunamente separati, è sufficiente rinominare il file con l'estensione .CSV

La coerenza di un file CSV (non si può parlare di validità a causa dell'assenza di uno standard) può avvenire con strumenti presenti nella rete internet come CSV Lint o Flat File Checker.

Molti editor di fogli elettronici (es.: excel) consentono la conversione nel formato CSV



LINGUAGGIO R

FILE TESTUALE CSV: lettura

Il package **UTILS** presente nelle librerie preinstallate nell'ambiente R, contiene diverse funzioni valide per importare i dati da questo formato.

La **lettura del file CSV** avviene mediante la funzione `read.csv` il cui risultato è un dataframe

La funzione ha la seguente sintassi:

```
read.csv(percorso_file, sep = separatore, comment.char=car, stringsAsFactors = FALSE)
```

Per leggere un file CSV si specifica il percorso e il nome del file (`percorso_file`); il separatore usato (`sep`); escludendo i dati dopo il simbolo `car` del parametro `comment.char`

LINGUAGGIO R

FILE TESTUALE CSV: lettura esempio

```
impiegati<-read.csv("impiegati.csv", sep=";", stringsAsFactors = FALSE)
```

```
#Stampa impiegati con salario superiore ai 600 euro
```

```
print(subset(impiegati,impiegati$Salario>600))
```

```
Id Nome Salario Data_Contratto Reparto
1 1 Franco 623.3 2012-01-01 ITC
3 3 Davide 611.0 2014-11-15 ITC
7 7 Francesca 632.8 2013-07-30 Personale
8 8 Alessandra 722.5 2014-06-17 Contabilità
```

```
#Stampa dell'impiegato con salario massimo
```

```
print(subset(impiegati,impiegati$Salario==max(impiegati$Salario)))
```

```
Id Nome Salario Data_Contratto Reparto
8 8 Alessandra 722.5 2014-06-17 Contabilità
```

```
#Stampa dell'impiegati del reparto ITC e con reddito superiore a 620 euro
```

```
print(subset(impiegati,((impiegati$Reparto=="ITC")&(impiegati$Salario>600))))
```

```
Id Nome Salario Data_Contratto Reparto
1 1 Franco 623.3 2012-01-01 ITC
3 3 Davide 611.0 2014-11-15 ITC
```

```
#Assunti dal primo agosto 2014
```

```
print(subset(impiegati, as.Date(impiegati$Data_Contratto)>as.Date("2014-07-30")))
```

```
Id Nome Salario Data_Contratto Reparto
3 3 Davide 611.00 2014-11-15 ITC
5 5 Cristiano 543.25 2015-03-27 Sicurezza
```

Id	Nome	Salario	Data_Contratto	Reparto
1	Franco	623.3	2012-01-01	ITC
2	Davide	515.2	2013-09-23	Contabilità
3	Fabio	611	2014-11-15	ITC
4	Gianmarco	721.9	2014-05-11	Sicurezza
5	Cristiano	843.25	2015-03-27	Sicurezza
6	Dalia	578	2013-05-21	ITC
7	Francesca	632.8	2013-07-30	Personale
8	Alessandra	722.5	2014-06-17	Contabilità

La prima colonna *N* deriva dal numero/nome di riga inserito dall'ambiente R e può essere eliminata ponendo il parametro `row.names=FALSE`



LINGUAGGIO R

FILE TESTUALE CSV: scrittura

Da un dataframe si può creare un documento CSV attraverso il comando:

```
write.csv(dataframe,file=percorso_e_nome_file_csv, row.names = TRUE)
```

Ad esempio è possibile memorizzare un segmento di dati di studenti che può essere impiegato per elaborazioni successive

LINGUAGGIO R

FILE TESTUALE CSV: scrittura esempio

```
classe <- data.frame(  
  id = c(1:5),  
  name = c("Massimo", "Roberto", "Arianna", "Giorgia", "Simona"),  
  data_nascita = as.Date(c("2000-04-13", "2000-01-01", "2000-09-11", "2000-08-15", "2000-12-25")),  
  media = c(9.5, 8.7, 9.2, 4.5, 6.7),  
  stringsAsFactors = FALSE  
)
```

```
fileName2 <- "C:/Users/flibe/Documents/Esercizi/LIBRO/classe2.csv"
```

```
#Scrittura del dataframe classe nel file CSV
```

```
ret<-write.csv(classe, fileName2, row.names=FALSE)
```

```
#Lettura e stampa del file CSV creato
```

```
cl<-read.csv("C:/Users/flibe/Documents/Esercizi/LIBRO/classe2.csv", sep=",")
```

```
print(classe)
```

	id	name	data_nascita	media
1	1	Massimo	2000-04-13	9.5
2	2	Roberto	2000-01-01	8.7
3	3	Arianna	2000-09-11	9.2
4	4	Giorgia	2000-08-15	4.5
5	5	Simona	2000-12-25	6.7



Fine