Informazioni su oggetti R

• Vettori ed array multi-dimensionali sono oggetti R che servono a rappresentare oggetti omogenei
• Se *a* è un oggetto R, per avere informazioni su *a* esistono alcuni comandi. '**mode**' restituisce il tipo dell'oggetto**, 'str'**I invece visualizza in modo succinto la struttura di un oggetto R

```
>  a <- c("a", "b", "c")
> mode(a)
[1] "character"
> b <- seq(1, 5, by=2)
> mode(b)
[1] "numeric"
```

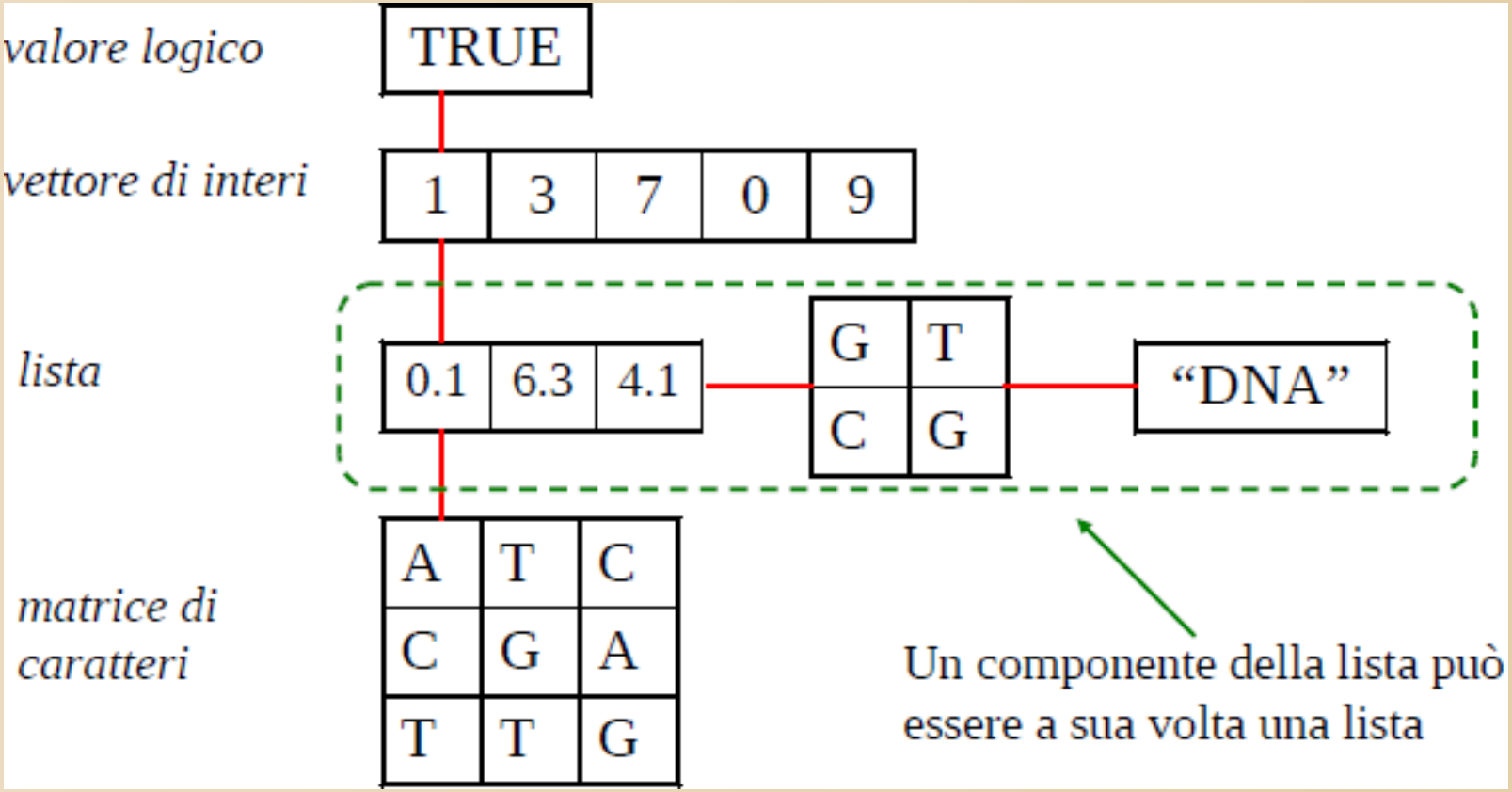
```
> str(a)
 chr [1:3] "a" "b" "c"

> str(b)
 num [1:25] 1 3 5 7 9 11 13 15 17 19 ...
```

Rappresentazioni di dati eterogenei: Liste

- La "lista" è uno dei tipi di oggetto R per rappresentare dati eterogenei (es. dati numerici e caratteri)
- Le liste rappresentano un *insieme ordinato* (nel senso che i suoi elementi sono ancora indicizzabili) *di oggetti* (componenti)
- Le componenti possono non essere dello stesso tipo o modo
- Le componenti possono essere ad es.: *un vettore numerico*, *un valore logico*, *una matrice*, *un array di caratteri*, *una funzione* o anche un' altra lista.

Liste

• La lista è quindi una *struttura dati ricorsiva,* poiché una componente può essere una lista



Costruzione di una lista

• Per costruire una lista si usa la funzione **list**:

```
> li <-list(c("s1", "s2"), 1:6)
> li
[[1]]
[1] "s1" "s2"

[[2]]
[1] 1 2 3 4 5 6
```

• Le componenti di una lista sono indicizzate, e sono a loro volta delle sottoliste di lunghezza 1. E' possibile usare l'operatore [] per accedere ad una sottolista, oppure l'operatore [[]] per accedere all'oggetto R contenuto nella sottolista

```
> li[1]
[[1]]
 [1] "s1" "s2"

> li[[1]]
[1] "s1" "s2"

> mode(li[1])
[1] "list"

> mode(li[[1]])
[1] "character"
```

Accesso mediante nomi: l'operatore $

• E' però possibile assegnare alle componenti un nome, a cui poi accedere mediante l'operatore **$**

```
> li <- list(val=TRUE, vector=c(1,3,7,0,9))

> li
$val
[1] TRUE

$vector
[1] 1 3 7 0 9
```

```
> li$val
[1] TRUE

> li$vector
[1] 1 3 7 0 9
```

Accesso mediante nomi: l'operatore $

• E' possibile assegnare i nomi alle componenti di una lista anche successivamente con il comando **names**:

```
> li <- list(TRUE, c(1,3,7,0,9))
>  names(li)<- c("val", "vector")
> li
$val
[1] TRUE

$vector
[1] 1 2 3 4 5 6

> li$val
[1] TRUE
```

```
> li[[1]]
[1] TRUE

> li[1]
$val
[1] TRUE
```

- l' eventuale nome associato alla componente viene incluso
- Quindi li$val è equivalente a li[[1]] e li$vector a li[[2]]

Accesso tramite indice “a caratteri”

• Se le componenti hanno un nome è possibile accedere ad esse tramite indice “a caratteri”

```
> li <- list(val = TRUE, vector = c(1,3,7,0,9),  m = matrix(1:12,nrow=2))
> li[["m"]]
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]  1    3    5    7    9    11
[2,]  2    4    6    8   10    12
```

• Questa modalità può essere utile quando il nome della componente è memorizzato in un’ altra variabile:

```
> v <- "vector"; li[[v]]
[1] 1 3 7 0 9
```

Allungare e concatenare liste

• Come qualsiasi altro oggetto accessibile tramite indici, le liste possono essere estese, aggiungendo specifiche componenti:

```
> li<-list(0.6798,
      paste(c(rep("A",4),
   rep("T",4)), collapse=""))
> li[3] <- list(TRUE)
> li
[[1]]
[1] 0.6798
[[2]]
[1] "AAAATTTT"
[[3]]
[1] TRUE
```

• La concatenazione di liste è possibile tramite la funzione **c**

```
> li1 <- list(TRUE,2)
> li2 <- list("pippo")
> li3<-list( c(1,2,3),
c("T","A","C"))
> li123 <- c(li1,li2,li3)
> li123
[[1]]
[1] TRUE
[[2]]
[1] 2
[[3]]
[1] "pippo"
[[4]]
[1] 1 2 3
[[5]]
[1] "T" "A" "C"
```

Esercizi

1) Siano dati i seguenti nominativi di donatori di sangue ed i corrispondenti gruppi sanguigni (GS):
   - *Nomi : Nome1, Nome2, ..., Nome16*
   - *GS: 0, A, 0, A, B, A, A, 0, 0, B, A , 0, A, A, B, B*

   a) Rappresentare i gruppi sanguigni in una tabella delle frequenze
   b) Rappresentare i gruppi sanguigni in una tabella delle frequenze relative
   c) Creare anche un vettore di cognomi:

   Congome1, Congnome2, ..., Cognome 16

   ed inserire nomi, cognomi e GS in una lista con nomi
   d) Restituire i nomi ed i cognomi delle persone che hanno gruppo sanguigno B

Esercizi

2) Costruire una lista *li* composta da una matrice numerica 4X4, da un vettore di caratteri con 32 elementi, dalla stringa “topo”, e da un’ ulteriore lista composta da un vettore di 10 elementi numerici e dal valore logico FALSE.

3) Si estragga dalla lista
li<-list(*m*=matrix(rnorm(64), nrow=8),*s*=c(rep("T",3),rep("G",5)))
- la seconda colonna della matrice *m*

4)Accedi in 3 modi diversi alla seconda componente della lista *li* dell’ es. 3)

Rappresentazioni di dati eterogenei: Data frame

• Un data frame può essere considerato come una matrice le cui colonne rappresentano dati eterogenei:

Dati:	val.num	val.car.	val. log.	val.num	val.car.
Dato1	3.45	ATTA	TRUE	0.45	CCAT
Dato2	5.67	TGAT	FALSE	0.91	TATT
Dato3	1.45	TATA	FALSE	3.78	CCCC
Dato4	4.56	TGAG	TRUE	8.03	GAGA
Dato5	8.09	CCCG	TRUE	8.09	AGAG
Dato6	3.11	CATG	TRUE	4.56	ATAT
Dato7	1.40	TGAG	FALSE	1.80	GCTA
Dato8	7.73	GGAC	TRUE	5.90	TGAT

- Formalmente è una lista di classe *data.frame*
- Le colonne del data frame rappresentano variabili i cui modi ed attributi possono essere differenti
- Le matrici e gli array sono invece costituiti da elementi omogenei per modo ed attributo

Componenti dei data frame

- Le colonne del data frame possono essere costituiti da:
  - Vettori (numerici, a caratteri, logici)
  - Fattori
  - Matrici numeriche
  - Liste
  - Altri data frame
- I data frame sono costuiti tramite la funzione **data.frame**:

```
> x <- 1:4
> y <- 5:8
> z <- paste("A",1:4,sep="")
> da.fr <- data.frame(x,y,z)
> da.fr
  x y  z
1 1 5 A1
2 2 6 A2
3 3 7 A3
4 4 8 A4
```

Vincolo sulle righe

```
> m1 <-matrix(1:12,nrow=2)
> v <- c("A","C")
> daf3<-data.frame(m1,v)
> daf3
  X1 X2 X3 X4 X5 X6 v
1  1  3  5  7  9 11 A
2  2  4  6  8 10 12 C
> v1<- c("A","C","G")
> daf4<-data.frame(m1,v1)
Error in data.frame(m1, v1) :
arguments imply differing number of
rows: 2, 3
```

- Il numero di righe di ogni componente deve essere lo stesso

```
> x <- rep(1, 2);
> daf3b <- data.frame(daf3, x) #
> daf3b
  X1 X2 X3 X4 X5 X6  v  x
1  1  3  5  7  9 11  A  1
2  2  4  6  8 10 12  C  1
```

Accesso alle componenti ed agli elementi dei data frame

- Esistono due modalità generali di accesso alle componenti ed agli elementi dei data frame:

1. I data frame sono liste, e quindi è possibile accedere ad essi secondo le *modalità di accesso tipiche delle liste* stesse

2. Come classe data frame, sono definiti *operatori di accesso tramite vettori di indici*, simili a quelli utlizzati per le matrici e gli array.

Accesso alle componenti dei data frame

Essendo liste, è possibile accedere alle componenti dei data frame secondo le modalità tipiche delle liste:

1. Accesso tramite indice numerico
2. Accesso tramite il nome delle componenti
3. Accesso tramite indice "a caratteri"

```
> x<-1:2; y<-5:6;
> z <- c("A1", "A2")
> da.fr<-data.frame(x,y,z)
```

1. Accesso tramite indice numerico:

```
> > da.fr[[2]]
[1] 5 6
> da.fr[2]
  y
1 5
2 6
```

2. Accesso tramite il nome delle componenti

```
> da.fr$y
[1] 5 6
```

3. Accesso tramite indice "a caratteri"

```
> da.fr["y"]
  y
1 5
2 6
> da.fr[["y"]]
[1] 5 6
```

Accesso alle componenti tramite vettori di indici

Sono definiti operatori di accesso specifici per la classe *data.frame*: mediante indici simili a quelli delle matrici ordinarie:

```
> x <- 1:2; y<-5:6; z <- c(0.5, 1.5)
> w <- paste("A",1:2,sep="")
> da.fr <- data.frame(x,y,z,w)
> da.fr
  x y   z  w
1 1 5 0.5 A1
2 2 6 1.5 A2
```

```
> da.fr[1,2]
[1] 5

> da.fr[2,2:3]
  y   z
2 6 1.5
> da.fr[1,]
  x y z   W
1 1 5 0.5 A1

> da.fr[,1:2]
  x y
1 1 5
2 2 6
```

Estrazione “logica” di osservazioni da data frame

```
> da.fr[da.fr$x>1,] # estrai da da.fr solo le osservazioni la cui variabile x > 1
  x y  z  w
2 2 6 1.5 A2
```

- **Equivalentemente si può usare la funzione subset:**

```
> subset(da.fr, x>1)
  x y  z  w
2 2 6 1.5 A2
```

- **Se si vogliono selezionare elementi da un insieme si può usare l’operatore %in%:**

```
> subset(da.fr,w %in% "A1")
  x  y  z   w
1 1  5 0.5 A1
> subset(da.fr,w %in% c("A1","A2"))
  x y  z   w
1 1 5 0.5  A1
2 2 6 1.5  A2
```

Esercizi

5) Ripetere l'esercizio 1) delle liste usando un data frame

6)Cosa è possibile rappresentare con una lista e non con un data frame?

7) Costruire una lista che abbia come componenti 3 vettori a caratteri. Trasformare la lista in un data frame tramite la funzione *as.data.frame*. Quali sono le restrizioni che si devono applicare alle liste perchè siano dei data frame?