Università degli Studi di Milano

Laurea Specialistica in Genomica Funzionale e Bioinformatica

Corso di Linguaggi di Programmazione per la Bioinformatica

Data frame

Giorgio Valentini

e –mail: *valentini@dsi.unimi.it*

DSI – Dipartimento di Scienze dell' Informazione
Università degli Studi di Milano

Data frame come struttura per rappresentare insiemi di dati eterogenei (1)

- 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

Data frame come struttura per rappresentare insiemi di dati eterogenei (2)

- 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):

Data frame

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

Matrice (array bidimensionale)

Dati:	val.num	val.num	val.num	val.num	val.num
Dato1	3.45	1.20	2.54	0.45	1.45
Dato2	5.67	2.95	5.44	0.91	0.95
Dato3	1.45	5.21	3.60	3.78	6.78
Dato4	4.56	8.00	2.12	8.03	8.73
Dato5	8.09	1.65	6.00	8.09	6.65
Dato6	3.11	2.58	3.98	4.56	3.56
Dato7	1.40	0.95	2.72	1.80	5.21
Dato8	7.73	8.82	4.43	5.90	3.18

• Un data frame può essere visualizzato come una matrice e si può accedere ai suoi elementi utilizzando indici (come per le matrici ordinarie)

3

Componenti dei data frame

- Formalmente i data frame sono *liste* di *classe* data.frame
- I *componenti* (colonne) del data frame possono essere costituiti da:
  - Vettori (numerici, a caratteri, logici)
  - Fattori
  - Matrici numeriche
  - Liste
  - Altri data frame

4

Caratteristiche dei componenti dei data frame

- I vettori numerici, logici ed i fattori sono inclusi direttamente come variabili (colonne) del data frame, mentre i vettori a caratteri sono forzati a fattori.
- Le matrici forniscono tante variabili al data frame quante sono le rispettive colonne
- Le liste forniscono tante variabili quanti sono i suoi componenti
- I data frame quanti sono i componenti

Restrizioni sulle componenti del data frame:

- I vettori componenti devono avere tutti la stessa lunghezza, mentre le matrici devono avere tutte lo stesso numero di righe
- I componenti delle liste incluse nel data frame devono rispettare le restrizioni di cui al punto precedente
- Le componenti del data frame A incluso nel data frame B devono essere conformi alle componenti del data frame B.

5

Costruzione dei 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
```

```
 mode(da.fr)
[1] "list"
> attributes(da.fr)
$names
[1] "x" "y" "z"
$row.names
[1] "1" "2" "3" "4"
$class
[1] "data.frame"
```

6

I data frame possono essere costruiti con matrici

Le matrici componenti il data frame devono avere lo stesso numero di righe:

```
> m1 <-matrix(1:12,nrow=2)
> m2 <-matrix(13:18,nrow=2)
> daf<-data.frame(m1,m2)
> daf
  X1 X2 X3 X4 X5 X6 X1 X2 X3
1  1  3  5  7  9 11 13 15 17
2  2  4  6  8 10 12 14 16 18
> m3 <-matrix(1:12,nrow=4)
> daf2<-data.frame(m1,m3)
Error in data.frame(m1, m3):
arguments imply differing
number of rows: 2, 4
```

Si possono utilizzare insieme matrici e vettori, purchè il numero delle righe delle matrici sia uguale alla lunghezza dei vettori:

```
> 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
```

7

I data frame possono essere costruiti con liste e con altri data frame

I componenti delle liste devono essere "compatibili":

```
> li <-list(a=matrix(1:12,nrow=3),
+ v=c("G","G","C"))
> m <- matrix(13:18,nrow=3)
> daf <- data.frame(li,m)
> daf
  a.1 a.2 a.3 a.4 v X1 X2
1   1   4   7  10 G 13 16
2   2   5   8  11 G 14 17
3   3   6   9  12 C 15 18
> m1 <- matrix(13:18,nrow=2)
> daf <- data.frame(li,m1)
Error in data.frame(li, m1) :
arguments imply differing number
of rows: 3, 2
```

Si possono utilizzare come componenti liste e data frame (ed anche matrici vettori e fattori), purchè compatibili:

```
> li <- list(a=matrix(1:12,nrow=3),
+ v=c("G","G","C"))
>daf.comp<-data.frame(v1=
+ c("A","B","C"),v2=c("D","E","F"))
> daf2<-data.frame(li,daf.comp)
> daf2
  a.1 a.2 a.3 a.4 v v1 v2
1   1   4   7  10 G  A  D
2   2   5   8  11 G  B  E
3   3   6   9  12 C  C  F
```

8

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.

9

Accesso alle componenti dei data frame mutuati dalle liste

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"

Es:

```
> x<-1:4; y<-5:8
> z<-paste("A",1:4,sep="")
> da.fr<-data.frame(x,y,z)
```

1. Accesso tramite indice numerico:

```
> da.fr[[1]]
[1] 1 2 3 4
> da.fr[1]
  x
1 1
2 2
3 3
4 4
```

2. Accesso tramite il nome delle componenti:

```
> da.fr$x
[1] 1 2 3 4
```

3. Accesso tramite indice "a caratteri":

```
> da.fr["x"]
  x
1 1
2 2
3 3
4 4
> da.fr[["x"]]
[1] 1 2 3 4
```

10

Accesso alle componenti tramite vettori di indici

Sono definiti operatori di accesso specifici per la classe *data.frame*: si tratta di vettori di indici con una semantica simile a quella delle matrici ordinarie:

Es:

```
> 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
```

```
> da.fr[1,2]
[1] 5
> da.fr[2,2:3]
  y  z
2 6 A2
> da.fr[3,]
  x y  z
3 3 7 A3
> da.fr[2:4,1:2]
  x y
2 2 6
3 3 7
4 4 8
```

11

Esempi di accesso alle componenti di un data frame

```
> mat<-matrix(c(rep("A",3),rep("T",3),rep("G",3),rep("C",3)),nrow=2)
> li <- list(v1=rnorm(2),m=matrix(rnorm(6),nrow=2))
> daf <-data.frame(mat,li) # costruzione data frame
> daf
  X1 X2 X3 X4 X5 X6         v1        m.1       m.2       m.3
1  A  A  T  G  G  C -0.8058378 -0.2722994 0.5641271 2.4615146
2  A  T  T  G  C  C  1.6268044 -0.7586567 0.9504489 0.6681619
```

Esempi di accesso alle componenti:

Accesso " a modo" lista

```
> daf[2]
  X2
1  A
2  T
> daf[[2]]
[1] A T
Levels: A T
> daf["v1"]
         v1
1 -0.8058378
2  1.6268044
> daf$m.1
[1] -0.2722994 -0.7586567
```

Accesso " a modo" matrice

```
> daf[[1,2]]
[1] 1
> daf[,5]
[1] G C
Levels: C G
> daf[,7]
[1] -0.8058378  1.6268044
> daf[,7:length(daf)]
    v1        m.1        m.2       m.3
1  -0.8058378 -0.2722994 0.5641271 2.4615146
2   1.6268044 -0.7586567 0.9504489 0.6681619
> daf[1,5:7]
  X5 X6         v1
1  G  C -0.8058378
```

12

Estrazione “logica” di osservazioni da data frame

```
> daf[daf$m.3>1,] # estrai da daf solo le osservazioni la cui
                  # variabile m.3 > 1
  X1 X2 X3 X4 X5 X6         v1        m.1       m.2      m.3
1  A  A  T  G  G  C -0.8058378 -0.2722994 0.5641271 2.461515
```

Equivalentemente si può usare la funzione subset:

```
subset(daf,m.3>1)
  X1 X2 X3 X4 X5 X6         v1        m.1       m.2      m.3
1  A  A  T  G  G  C -0.8058378 -0.2722994 0.5641271 2.461515
```

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

```
> subset(daf,X2 %in% "A")
  X1 X2 X3 X4 X5 X6         v1        m.1       m.2      m.3
1  A  A  T  G  G  C -0.8058378 -0.2722994 0.5641271 2.461515
> subset(daf,X2 %in% c("A","T"))
  X1 X2 X3 X4 X5 X6         v1        m.1       m.2       m.3
1  A  A  T  G  G  C -0.8058378 -0.2722994 0.5641271 2.4615146
2  A  T  T  G  C  C  1.6268044 -0.7586567 0.9504489 0.6681619
```

13

La funzione str

La funzione str(*oggetto*) fornisce una serie minima di informazione su *oggetto*.

Es.

```
> data(iris) # caricamento di un data frame da un file
             # contenuto in un package precedentemente caricato
> mode(iris)
[1] "list"
> class(iris)
[1] "data.frame"
> iris
...
> str(iris)
`data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versic..",..: 1 1 1 1
   1 1 1 1 1 1 ...
```

14

La funzione summary

La funzione summary(*oggetto*) forinisce una serie di informazioni statistiche su *oggetto*.

Es:

```
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```


Una nota sull' accesso alle variabili

Un data frame è una lista di "colonne":

```
> data( Formaldehyde); str( Formaldehyde)
`data.frame':   6 obs. of  2 variables:
 $ carb  : num  0.1 0.3 0.5 0.6 0.7 0.9
 $ optden: num  0.086 0.269 0.446 0.538 0.626 0.782
> Formaldehyde$optden
[1] 0.086 0.269 0.446 0.538 0.626 0.782
> Formaldehyde[["optden"]]
[1] 0.086 0.269 0.446 0.538 0.626 0.782
> Formaldehyde[[2]]
[1] 0.086 0.269 0.446 0.538 0.626 0.782
>  Formaldehyde[,2]
[1] 0.086 0.269 0.446 0.538 0.626 0.782
> Formaldehyde[2]
  optden
1  0.086
2  0.269
3  0.446
4  0.538
5  0.626
6  0.782
> mode(Formaldehyde[2])
[1] "list"
```

Accesso all' oggetto (vettore in questo caso) del data frame

Accesso al componente (una lista) del data frame

“Dropping” delle dimensioni

Si è visto che con:

```
>  Formaldehyde[,2]
[1] 0.086 0.269 0.446 0.538 0.626 0.782
```

si accede all’ oggetto del data frame (variabile di 6 osservazioni interpretata come vettore numerico).

Come fare a mantenere la struttura data.frame ?

```
> str(Formaldehyde[,2])  # vettore
 num [1:6] 0.086 0.269 0.446 0.538 0.626 0.782
```

si usa il parametro drop

```
> str(Formaldehyde[,2, drop=FALSE])
`data.frame':   6 obs. of  1 variable:
 $ optden: num  0.086 0.269 0.446 0.538 0.626 0.782
> dim(Formaldehyde[,2]) # un vettore non ha attributo dimensioni
NULL
> dim( Formaldehyde[,2,drop=FALSE]) # un data frame sì
[1] 6 1
```

17

Le funzioni attach e detach (1)

La notazione *oggetto$componente* utilizzata per liste e data frame in alcuni contesti può essere eccessivamente verbosa e poco conveniente.

La funzione attach “rende disponibili” nel cammino di ricerca corrente i nomi delle componenti come se fossero variabili “stand alone”:

```
> da.fr<-data.frame(x=1:2,y=3:4,z=paste("C",1:2,sep=""))
> da.fr
  x y  z
1 1 3 C1
2 2 4 C2
> attach(da.fr)
> x # la componente x di da.fr è accessibile direttamente
[1] 1 2
> z # la componente z di da.fr è accessibile direttamente
[1] C1 C2
Levels: C1 C2
```

Assegnamenti o modifiche sulle variabili “estratte” dal data frame con detach non hanno effetto sul data frame stesso. Per modificare le componenti è necessario utilizzare la notazione *oggetto$componente* :

```
> x<-y
> da.fr # da.fr immutato
  x y  z
1 1 3 C1
2 2 4 C2
> da.fr$x<-y
> da.fr # da.fr modificato
  x y  z
1 3 3 C1
2 4 4 C2
```

18

Le funzioni attach e detach (2)

La funzione detach elimina dal cammino di ricerca le componenti delle liste o data frame precedentemente rese disponibili dalla funzione attach:

```
> da.fr<-data.frame(x=1:2,y=3:4,z=paste("C",1:2,sep=""))
> da.fr
  x y  z
1 1 3 C1
2 2 4 C2
> attach(da.fr)
> x # la componente x di da.fr è accessibile direttamente
[1] 1 2
> z # la componente z di da.fr è accessibile direttamente
[1] C1 C2
Levels: C1 C2
> detach(da.fr)
> x  # la variabile x non è più visibile
Error: Object "x" not found
> y # la variabile y non è più visibile
Error: Object "y" not found
```

19

Esercizi

1. Costruire un data frame da.fr che abbia come componenti un vettore numerico casuale v di lunghezza 20, una matrice casuale m con 4 colonne ed una lista i cui componenti siano 3 matrici a piacere.
2. 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?
3. Si consideri il *data frame daf della slide 12*.
   (a) Estrarre da daf l' ultima colonna
   (b) Estrarre da daf le righe la cui variabile X2 sia uguale ad "A".
   (c) Estrarre da daf un data frame composto solo dalle colonne 4,5,6 e 7.
   (d) Modificare l' ultima colonna di daf in <0,0>
   (e) Aggiungere al data frame una nuova colonna i cui valori rappresentino la somma delle colonne v1, m1, m2 ed m3.
4. Selezionare dal data set *iris* le osservazioni relative alle specie "virginica" con Petal.Length>5.890.
5. Tremite la funzione *summary* ricavare informazioni statistiche di base sulla specie "versicolor" del data set *iris*.
6. Si dispone di un insieme di dati sperimentali (ad es: dati clinici e dati bio-molecolari) da utilizzare a fini diagnostici, relativi ad un insieme di pazienti. Si discuta se ed in quali condizioni i dati siano rappresentabili tramite data frame.

20