Programming for Data Science

Data Frame in R language

Marco Beccuti

Università degli Studi di Torino

Dipartimento di Informatica

October 2021

Data Frame in R

- It is used to storage data table in R;
- It can be considered as a matrix in which columns can contain different types;
- We can create data frames from pre-existing variables:

```
> name = c("GENE1","GENE2","GENE3")
> seq = c("ATCCT..","CCTTT..","CCAACT..")
> count = c(100,20,4)
> d = data.frame(name,seq,count)
> d
```

	name	seq	count
1	GENE1	ATCCT..	100
2	GENE2	CCTTT..	20
3	GENE3	CCACT..	4

Data Frame in R

Main operations:

- attributes(d) returns the data frame attributes:

  $> attributes(d)$
  $\$names$
  [1]"name" "seq" "count"
  $\$row.names$
  [1]1 2 3
  $\$class$ [1]"data.frame"

- colnames(d) returns the names of data frame columns:

  $> colnames(d)$
  [1]"name" "seq" "count"

  $> colnames(d) = c(\text{“}c1\text{”}, \text{“}c2\text{”}, \text{“}c3\text{”}, \text{“}c4\text{”})$ **change column names.**

- rownames(d) returns the names of data frame rows:

  $> rownames(d)$
  [1]1 2 3

Indexing Data Frame in R

- it is possible to use column name to access columns of a data frame.

  $> d$

	name	*seq*	*count*
1	*GENE*1	*ATCCT*..	100
2	*GENE*2	*CCTTT*..	20
3	*GENE*3	*CCACT*..	4

  $> d\$count$ **gives the values in the 3rd column of d.**

  [1]100 20 4

- Selecting all data for cases that satisfy some criterion.

  $> d[d\$count > 10,]$

	name	*seq*	*count*
1	*GENE*1	*ATCCT*..	100
2	*GENE*2	*CCTTT*..	20

Indexing Data Frame in R

- it is possible to use the same method of matrices to access values of a data frame.

$> d$

	name	*seq*	*count*
1	*GENE*1	*ATCCT*..	100
2	*GENE*2	*CCTTT*..	20
3	*GENE*3	*CCACT*..	4

$> d[2,2]$ **gives the value in the 2nd row and 2nd column of d.**
[1]*CCTTT*..

$> d[2,]$ **gives the values in the 2nd row of d.**
[1]*GENE*2 *CCTTT*.. 20

$> d[,3]$ **gives the values in the 3rd column of d.**
[1]100 20 4

Data Frame in R

Main operations(2):

- summary(d) returns a summary of data frame:

  $> summary(d)$

name	seq		count
GENE1 : 1	ATCCT.. : 1	Min. :	4.000
GENE2 : 1	CCTTT.. : 1	1stQu. :	12.00
GENE3 : 1	CCACT.. : 1	Median :	20.00
		Mean :	41.33
		3rdQu. :	60.00
		Max. :	100.00

- Observe text columns are converted in factor by default. Parameter stringsAsFactors to deal with this.

  $> d = data.frame(name, seq, count, stringAsFactors = FALSE)$

Data Frame in R

Main operations(4):

- subset(d,cond) returns a subset of rows according to condition:

$> subset(d, d[,3] > 10)$

	name	*seq*	*count*
1	*GENE*1	*ATCCT..*	100
2	*GENE*2	*CCTTT..*	20

$> subset(d, d\$count > 10)$

	name	*seq*	*count*
1	*GENE*1	*ATCCT..*	100
2	*GENE*2	*CCTTT..*	20

Data Frame in R

Main operations(5):

- which(condition) gives the TRUE indices of a logical object.
  Then, it answers to the question "Which indices are TRUE?"

  $> which(d[,3] > 10)$
  [1] 1 2

  $> which(d[,3] == 20)$
  [1] 2

  $> which(d[,3] \%in\% 1:20)$ **operator %in% tests which elements of d are in 1:20.**
  [1] 2 3

  $> which(d[,1] \%in\% c("GENE1","GENE3"))$
  [1] 1 3

Exercises on Data Frames

- Create a data frame called *D* with the following data:

Firstname	Lastname	Age	Gender	Points
Alice	Ryan	37	F	278
Paul	Collins	34	M	242
Jerry	Burke	26	M	312
Thomas	Dolan	72	M	740
Marguerite	Black	18	F	177
Linda	McGrath	24	F	195

- Store the points for every person into a vector called *pts*, then calculate the average number of points received.

- Store the data for the females only into a data frame called *fpoints*, then calculate the summary.

Exercises on Data Frames

- Create a data frame called D with the following data:

Firstname	Lastname	Age	Gender	Points
Alice	Ryan	37	F	278
Paul	Collins	34	M	242
Jerry	Burke	26	M	312
Thomas	Dolan	72	M	740
Marguerite	Black	18	F	177
Linda	McGrath	24	F	195

$> Firstname = c(\text{“}Alice\text{”}, \text{“}Paul\text{”}, \text{“}Jerry\text{”}, \text{“}Thomas\text{”}, \text{“}Marguerite\text{”}, \text{“}Linda\text{”})$
$> Lastname = c(\text{“}Ryan\text{”}, \text{“}Collins\text{”}, \text{“}Burke\text{”}, \text{“}Dolan\text{”}, \text{“}Black\text{”}, \text{“}McGrath\text{”})$
$> Age = c(37, 34, 26, 72, 18, 24)$
$> Gender = c(\text{“}F\text{”}, \text{“}M\text{”}, \text{“}M\text{”}, \text{“}M\text{”}, \text{“}F\text{”}, \text{“}F\text{”})$
$> Points = c(278, 242, 312, 740, 177, 195)$
$> D = data.frame(Firstname, Lastname, Age, Gender, Points)$ **vector names are used as column names.**

Exercises on Data Frames

- Store the points for every person into a vector called *pts*, then calculate the average number of points received.

```
> pts = D$Points
> pts
[1]278 242 312 740 177 195
> mean(pts)
[1]324
```

Exercises on Data Frames

- Store the data for the females only into a data frame called *fpoints*, then calculate the summary.

  $> fpoints = subset(D, D\$Gender == \text{“}F\text{”})$
  $summary(fpoints)$

Exercises on Data Frames

- The age for Paul Collins was entered incorrectly. Change his age to 48.
- Determine the maximum age of the males.
- Extract the data for people with more than 100 points and are over the age of 30.

Exercises on Data Frames

- The age for Paul Collins was entered incorrectly. Change his age to 48.

  $> D[2, 3] = 48$

Exercises on Data Frames

- Determine the maximum age of the males.

  $> max(subset(D, D\$Gender == \text{“}M\text{”})\$Age)$
  [1]72

Exercises on Data Frames

- Extract the data for people with more than 100 points and are over the age of 30.

  $> subset(D, D\$Age > 30 \& D\$Points > 100)$