

Lecture 2



Class: SY BSc

Subject : Statistical modelling in R

Chapter: Unit 1 Chp 2

Chapter Name: Data Management

Today's Agenda

1. Creating DataFrame
2. Subsetting
3. Grouping data
4. Apply function
5. Data cleaning
6. Working with datasets

Data frames

- A data frame is the most common way of storing data in R,
- A data frame is a list of equal-length vectors.
- This makes it a 2-dimensional structure,
- so it shares properties of both the matrix and the list.
- This means that a data frame has `names()`, `colnames()`, and `rownames()`,
- although `names()` and `colnames()` are the same thing.
- The `length()` of a data frame is the length of the underlying list
- and so is the same as `ncol()`; `nrow()` gives the number of rows.

Dataframes

- `> df`
`=data.frame(roll=c(1:5),sname=c("Sunil","Raj","Rohit","Rohan","Suresh"),grade=c("A","A+","B","B+","C"))`
- `> df`
- roll sname grade
- 1 1 Sunil A
- 2 2 Raj A+
- 3 3 Rohit B
- 4 4 Rohan B+
- 5 5 Suresh C

Default behavior to convert strings to factors

- `> str(df)`
- `'data.frame':` 5 obs. of 3 variables:
- `$ roll : int` 1 2 3 4 5
- `$ sname: Factor w/ 5 levels` "Raj","Rohan",...: 4 1 3 2 5
- `$ grade: Factor w/ 5 levels` "A","A+","B","B+",...: 1 2 3 4 5

Prevent conversion of strings to factors

- `> df1 =data.frame(roll=c(1:5),sname=c("Sunil","Raj","Rohit","Rohan","Suresh"),
grade=c("A","A+","B","B+","C"),stringsAsFactors =FALSE)`
- `> str(df1)`
- `'data.frame': 5 obs. of 3 variables:`
- `$ roll : int 1 2 3 4 5`
- `$ sname: chr "Sunil" "Raj" "Rohit" "Rohan" ...`
- `$ grade: chr "A" "A+" "B" "B+" ...`

Dataframes

- `> df1$grade=as.factor(df1$grade)`
- `> str(df1)`
- 'data.frame': 5 obs. of 3 variables:
- `$ roll : int 1 2 3 4 5`
- `$ sname: chr "Sunil" "Raj" "Rohit" "Rohan" ...`
- `$ grade: Factor w/ 5 levels "A","A+","B","B+",...: 1 2 3 4 5`

Converting into data frame

- `> as.data.frame(mat_d)`
- V1 V2 V3
- 1 1.0 4.6 8.2
- 2 2.8 6.4 10.0

Combining dataframes

Column bind

- `> df1 =data.frame(roll=c(1:5),sname=c("Sunil","Raj","Rohit","Rohan","Suresh"),`
- `+ grade=c("A","A+","B","B+","C"),stringsAsFactors =FALSE )`
- `> > marks<-c(80,90,45,67,98)`
- `> cbind(df1,marks)`
- | | roll | sname | grade | marks |
|-----|------|-------|-------|-------|
| • 1 | 1 | Sunil | A | 80 |
| • 2 | 2 | Raj | A+ | 90 |
| • 3 | 3 | Rohit | B | 45 |

Row bind

- `rbind(df1,data.frame(roll=6,sname="Priya",grade="A",marks=78))`
- roll sname grade marks
- 1 1 Sunil A 80
- 2 2 Raj A+ 90
- 3 3 Rohit B 45
- 4 4 Rohan B+ 67
- 5 5 Suresh C 98
- 6 6 Priya A 78



Subsetting vectors in R

Subsetting vectors in R

There are five ways to subset vectors:

1

Using positive integers

Using negative integers

2

3

Using logical operators

With nothing

4

5

With zero

Private and Confidential

Using positive integers

- `> li`
- `[1] 2 4 6 8 10 12 14 16 18 20`
- `> li[3]`
- `[1] 6`
- `> li[c(3,4)]`
- `[1] 6 8`
- `> li[c(2.9,3.4)]`
- `[1] 4 6`
- `> li[c(1,1)]`
- `[1] 2 2`

Using negative integer

- `> li[-5]`
- `[1] 2 4 6 8 12 14 16 18 20`
- `> li[-c(2,5)]`
- `[1] 2 6 8 12 14 16 18 20`

Using logical operators

- `> li`
- `[1] 2 4 6 8 10 12 14 16 18 20`
- `> li[li>5]`
- `[1] 6 8 10 12 14 16 18 20`
- `> li[li>5 & li<10]`
- `[1] 6 8`

With nothing

```
> 1 1 [] [1] 2 4 6 8 10 12 14 16  
18 20
```

With 0

- 0 position contains the type information about a vector
- `li[0]`
- `numeric(0)`
- `> li`
- `[1] 2 4 6 8 10 12 14 16 18 20`
- `> li1=c("Mumbai","kolkata","Bangalore")`
- `> li1[0]`
- `character(0)`

Dataframes and subsetting

Creating data frame using student records

- `> name=c('Tom','James','Ricky','Vin','Steve','Smith','Jack')`
- `> age=c(25,26,25,23,30,29,23)`
- `> rating=c(4.23,3.24,3.98,2.56,3.20,4.6,3.8)`
- `> df_stud=as.Data.Frame(Name=name,Age=age,Rating=rating)`

Displaying Data Frame

- `df_stud=data.frame("Name"=name,"Age"=age,"Rating"=rating)`
- `> df_stud`
- Name Age Rating
- 1 Tom 25 4.23
- 2 James 26 3.24
- 3 Ricky 25 3.98
- 4 Vin 23 2.56
- 5 Steve 30 3.20
- 6 Smith 29 4.60
- 7 Jack 23 3.80

Transposing the Data Frame

- > #transposing
- > t(df_stud)
- [,1] [,2] [,3] [,4] [,5] [,6] [,7]
- Name "Tom" "James" "Ricky" "Vin" "Steve" "Smith" "Jack"
- Age "25" "26" "25" "23" "30" "29" "23"
- Rating "4.23" "3.24" "3.98" "2.56" "3.20" "4.60" "3.80"

Viewing the structure of Data Frame

- `> str(df_stud)`
- 'data.frame': 7 obs. of 3 variables:
- \$ Name : Factor w/ 7 levels "Jack","James",...: 6 2 3 7 5 4 1
- \$ Age : num 25 26 25 23 30 29 23
- \$ Rating: num 4.23 3.24 3.98 2.56 3.2 4.6 3.8

Subsetting using numeric indexing

#select first 2 rows using numeric indexing

- `> df_stud[1:2,]`
- Name Age Rating
- 1 Tom 25 4.23
- 2 James 26 3.24

#select 2nd and 4th rows using numeric indexing

- `> df_stud[c(2,4),]`
- Name Age Rating
- 2 James 26 3.24
- 4 Vin 23 2.56

#select all rows 3rd column using numeric indexing

- `> df_stud[,3]`
- `[1] 4.23 3.24 3.98 2.56 3.20 4.60 3.80`

#select all rows 1st and 3rd column using numeric indexing

- `> df_stud[,c(1,3)]`
- Name Rating
- 1 Tom 4.23
- 2 James 3.24
- 3 Ricky 3.98
- 4 Vin 2.56
- 5 Steve 3.20
- 6 Smith 4.60
- 7 Jack 3.80

#select all rows and all columns except 2nd column using numeric indexing

- `> df_stud[, -2]`
- Name Rating
- 1 Tom 4.23
- 2 James 3.24
- 3 Ricky 3.98
- 4 Vin 2.56
- 5 Steve 3.20
- 6 Smith 4.60
- 7 Jack 3.80

#select all rows and all rows except 2nd and 4th row using numeric indexing

- `> df_stud[c(-2,-4),]`
- Name Age Rating
- 1 Tom 25 4.23
- 3 Ricky 25 3.98
- 5 Steve 30 3.20
- 6 Smith 29 4.60
- 7 Jack 23 3.80

Subsetting using condition

Subset(),which(),indexing

- `> subset(df_stud,Name=="Vin",select=1:3)`
- Name Age Rating
- 4 Vin 23 2.56
- `> df_stud[which(df_stud$Name=="Vin"),]`
- Name Age Rating
- 4 Vin 23 2.56
- `> df_stud[df_stud$Name=="Vin",]`
- Name Age Rating
- 4 Vin 23 2.56

> #drop the row which contains ricky

- > df_stud1=subset(df_stud,First_Name!="Ricky",1:3)
- > df_stud1
- First_Name Age Rating
- 1 Tom 25 4.23
- 2 James 26 3.24
- 4 Vin 23 2.56
- 5 Steve 30 3.20
- 6 Smith 29 4.60
- 7 Jack 23 3.80

> #drop age column

- > df_stud2=subset(df_stud,,select=-2)
- > df_stud2
- First_Name Rating
- 1 Tom 4.23
- 2 James 3.24
- 3 Ricky 3.98
- 4 Vin 2.56
- 5 Steve 3.20
- 6 Smith 4.60
- 7 Jack 3.80