Data manipulation: basics for data frame

Qiang Shen

Sept. 29, 2016

Data arrangement

- basics of data arrangement
- apply

Example

- gender different of leadership

```
##   manager     date gender age q1 q2 q3 q4 q5
## 1       1 10/24/08      M  32  5  4  5  5  5
## 2       2 10/28/08      F  56  3  5  2  5  5
## 3       3  10/1/08      F  25  3  5  5  5 NA
## 4       4 10/12/08      M  60  3  3  4 NA  3
## 5       5   5/1/09      F  99  2  2  1  2  1
```

Creating new variables

```
sum_q12<-(leadership$q1+leadership$q2)/2
sum_q12
```

```
## [1] 4.5 4.0 4.0 3.0 2.0
```

```
leadership
```

```
##   manager     date gender age q1 q2 q3 q4 q5
## 1       1 10/24/08      M  32  5  4  5  5  5
## 2       2 10/28/08      F  56  3  5  2  5  5
## 3       3  10/1/08      F  25  3  5  5  5 NA
## 4       4 10/12/08      M  60  3  3  4 NA  3
## 5       5   5/1/09      F  99  2  2  1  2  1
```

Arithmetic operators

Operator	Description
+	Addition
–	Subtraction
*	Multiplication
/	Division
^ or **	Exponentiation
x%%y	Modulus (x mod y) 5%%2 is 1
x%/%y	Integer division 5%/%2 is 2

Figure 1:

Creating new variables continued

```
attach(leadership)
leadership$sum_q12<-q1+q2
detach(leadership)
leadership

leadership_new<-transform(leadership,
                          sum_q12=(q1+q2),  ###not <-
                          mean_q12=(q1+q2)/2)
leadership_new
```

Recoding variables

-

Operator	Description
`<`	Less than
`<=`	Less than or equal to
`>`	Greater than
`>=`	Greater than or equal to
`==`	Exactly equal to
`!=`	Not equal to
`!x`	Not x
`x | y`	x or y
`x & y`	x and y
`isTRUE(x)`	Test if x is `TRUE`

Recoding variables

```
##   manager     date gender age q1 q2 q3 q4 q5
## 1       1 10/24/08      M  32  5  4  5  5  5
## 2       2 10/28/08      F  56  3  5  2  5  5
## 3       3  10/1/08      F  25  3  5  5  5 NA
## 4       4 10/12/08      M  60  3  3  4 NA  3
## 5       5   5/1/09      F  NA  2  2  1  2  1
```

```
##   manager     date gender age q1 q2 q3 q4 q5       agecat
## 1       1 10/24/08      M  32  5  4  5  5  5        Young
## 2       2 10/28/08      F  56  3  5  2  5  5 Middle Aged
## 3       3  10/1/08      F  25  3  5  5  5 NA        Young
## 4       4 10/12/08      M  60  3  3  4 NA  3 Middle Aged
## 5       5   5/1/09      F  NA  2  2  1  2  1         <NA>
```

Recoding variables continued

```
# compact version

leadership <- within(leadership, {
  agecat <- NA
  agecat[age > 75] <- "Elder"
  agecat[age >= 55 & age <= 75] <- "Middle Aged"
  agecat[age < 55] <- "Young"
})
leadership
```

Rename

```
names(leadership)[2]<- "testDate" #rename one varible
names(leadership)[5:9] <-c('aq1',"aq2","aq3","aq4","aq5") #
names(leadership)[names(leadership)=='gender']<-'Gender'
leadership
```

```
##   manager testDate Gender age aq1 aq2 aq3 aq4 aq5       a
## 1       1 10/24/08      M  32   5   4   5   5   5
## 2       2 10/28/08      F  56   3   5   2   5   5 Middle
## 3       3  10/1/08      F  25   3   5   5   5  NA
## 4       4 10/12/08      M  60   3   3   4  NA   3 Middle
## 5       5   5/1/09      F  NA   2   2   1   2   1
```

Rename continued

– reshape

```
if (!(require(reshape))) install.packages("reshape")
library(reshape)
leadership
rename(leadership, c(manager = "managerID", date = "testDat
```

Missing data: NA

```
mean(leadership$aq4)
mean(leadership$aq4,na.rm=T)
na.omit(leadership)
leadership
leadership[which(leadership$aq4!=NA),]
leadership[!is.na(leadership$aq4),]
```

Format convert

```
leadership$age <- as.numeric(leadership$age)
is.numeric(leadership$age)
leadership$age <- as.character(leadership$age)
is.character(leadership$age)
leadership$age <- as.numeric(leadership$age)
is.numeric(leadership$age)
```

missing data

```
value<-c(5,5,6,7,5,NA,3,5)
value==5
```

```
## [1]  TRUE  TRUE FALSE FALSE  TRUE    NA FALSE  TRUE
```

```
value!=5
```

```
## [1] FALSE FALSE  TRUE  TRUE FALSE    NA  TRUE FALSE
```

```
a <- c(1,2,2,5,1,NA,0,2)
b <- c(1,NA,4,7,1,NA,-1,2)
d <- c(1,1,NA,6,1,NA,1,2)
k<-data.frame(a,b,d)
logic<-sapply(k,is.na)
k$e<-rowSums(!logic)



k$value<-rowSums(k[,c('a','b','d')],na.rm=T)
k$value[k$e==0]<-NA
```

Sort data

```
attach(leadership)
newdata <- leadership[order(age), ]
newdata
detach(leadership)
attach(leadership)
newdata <- leadership[order(gender, -age), ]### blank on th
newdata
detach(leadership)
```

Select variables

```
newdata <- leadership[, c(5:9)]  #blank on the left side fo
newdata
myvars <- c("aq1", "aq2", "aq3", "aq4", "aq5")
newdata <- leadership[myvars]
newdata
myvars <- paste("aq", 1:5, sep = "")
myvars
newdata <- leadership[myvars]
```

Drop variables

```
newdata <- leadership[,c(-7, -8)]
newdata
newdata <- leadership[c(-7, -9)]
newdata <- leadership[,c(-7:-9)]

leadership[,-13]
leadership$season<-NULL
leadership

myvars <- names(leadership) %in% c("aq3", "aq4")
myvars
newdata <- leadership[!myvars]
```

Select Observations

```
newdata <- leadership[1:3, ]
newdata <- leadership[which(leadership$gender == "M" &
                            leadership$age > 30), ]
attach(leadership)
newdata <- leadership[gender == "M" & age > 30, ]
newdata <- leadership[which(gender == "M" & age > 30), ]
###blank on the right side for the observation selection
detach(leadership)
```

Subset with subset() function

```
newdata <- subset(leadership, age >= 35 | age < 24, c(aq1,
newdata <- subset(leadership, gender == "M" & age > 25, sel
newdata <- subset(leadership, rownames=1:3, select = Gender
```

Using SQL statements to manipulate data frames

```
if(!require(sqldf)) install.packages('sqldf')
newdf <- sqldf("select * from mtcars where carb=1 order by
                row.names = TRUE)
newdf
newdf <- sqldf("select avg(mpg) as avg_mpg, avg(disp) as av
    gear from mtcars where cyl in (4, 6) group by gear")
newdf
```