semana 3

Luis Ambrocio

5/8/2021

Contents

Subconjunto y clasificación

asignacion a dataframes ya creados

```
set.seed(13435)
X <- data.frame("var1"=sample(1:5),"var2"=sample(6:10),"var3"=sample(11:15))
X <- X[sample(1:5),]; X$var2[c(1,3)] = NA
X
```

```
##   var1 var2 var3
## 5    2   NA   11
## 4    4   10   12
## 1    3   NA   14
## 2    1    7   15
## 3    5    6   13
```

extrayendo columnas

```
X[,1]
```

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

```
X[,"var1"]
```

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

```
X[1:2,"var2"]
```

```
## [1] NA 10
```

extrayendo con condiciones

```
X[(X$var1 <= 3 & X$var3 > 11),]
```

```
##   var1 var2 var3
## 1    3   NA   14
```

```
## 2    1    7   15
```

```
X[(X$var1 <= 3 | X$var3 > 15),]
```

```
##   var1 var2 var3
## 5    2   NA   11
## 1    3   NA   14
## 2    1    7   15
```

lidiando con valores NA

```
X[which(X$var2 > 8),]
```

```
##   var1 var2 var3
## 4    4   10   12
```

ordenar

```
sort(X$var1)
```

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

```
sort(X$var1,decreasing=TRUE)
```

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

```
sort(X$var2,na.last=TRUE)
```

```
## [1]  6  7 10 NA NA
```

con la funcion order

```
X[order(X$var1),]
```

```
##   var1 var2 var3
## 2    1    7   15
## 5    2   NA   11
## 1    3   NA   14
## 4    4   10   12
## 3    5    6   13
```

```
X[order(X$var1,X$var3),]
```

```
##   var1 var2 var3
## 2    1    7   15
## 5    2   NA   11
## 1    3   NA   14
## 4    4   10   12
## 3    5    6   13
```

ordenando usando plyr

```
library(plyr)
arrange(X,var1)
```

```
##   var1 var2 var3
## 1    1    7   15
## 2    2   NA   11
## 3    3   NA   14
## 4    4   10   12
## 5    5    6   13
```

```
arrange(X,desc(var1))
```

```
##   var1 var2 var3
## 1    5    6   13
## 2    4   10   12
## 3    3   NA   14
## 4    2   NA   11
## 5    1    7   15
```

añadiendo filas y columnas

```
X$var4 <- rnorm(5)
X
```

```
##   var1 var2 var3       var4
## 5    2   NA   11 -0.4150458
## 4    4   10   12  2.5437602
## 1    3   NA   14  1.5545298
## 2    1    7   15 -0.6192328
## 3    5    6   13 -0.9261035
```

```
Y <- cbind(X,rnorm(5))
Y
```

```
##   var1 var2 var3       var4    rnorm(5)
## 5    2   NA   11 -0.4150458 -0.66549949
## 4    4   10   12  2.5437602 -0.02166735
## 1    3   NA   14  1.5545298 -0.17411953
## 2    1    7   15 -0.6192328  0.23900438
## 3    5    6   13 -0.9261035 -1.83245959
```

Notas y otros recursos

- Programación R en la pista de ciencia de datos
- Notas de la conferencia de Andrew Jaffe http://www.biostat.jhsph.edu/~ajaffe/lec_winterR/Lecture%202.pdf

Resumiendo datos

```
data("iris")
restData <-iris
```

viendo un trozo de los datos

```
head(restData,n=3)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
```

```
tail(restData,n=3)
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 148          6.5         3.0          5.2         2.0 virginica
## 149          6.2         3.4          5.4         2.3 virginica
## 150          5.9         3.0          5.1         1.8 virginica
```

hacer resumen

```
summary(restData)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  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
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

Información más detallada

```
str(restData)
```

```
## '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","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

quantiles de las variables cuantitativas

```
quantile(restData$Sepal.Widtht,na.rm=TRUE)
```

```
##   0%  25%  50%  75% 100%
##   NA   NA   NA   NA   NA
```

```
quantile(restData$Sepal.Width,probs=c(0.5,0.75,0.9))
```

```
##  50%  75%  90%
## 3.00 3.30 3.61
```

hacer tablas

```
table(restData$Species,useNA="ifany")
```

```
##
##     setosa versicolor  virginica
##         50         50         50
```

Make table

```
table(restData$Sepal.Width,restData$Species)
```

```
##
##       setosa versicolor virginica
##   2        0          1         0
##   2.2      0          2         1
##   2.3      1          3         0
##   2.4      0          3         0
```

```
##   2.5      0          4         4
##   2.6      0          3         2
##   2.7      0          5         4
##   2.8      0          6         8
##   2.9      1          7         2
##   3        6          8        12
##   3.1      4          3         4
##   3.2      5          3         5
##   3.3      2          1         3
##   3.4      9          1         2
##   3.5      6          0         0
##   3.6      3          0         1
##   3.7      3          0         0
##   3.8      4          0         2
##   3.9      2          0         0
##   4        1          0         0
##   4.1      1          0         0
##   4.2      1          0         0
##   4.4      1          0         0
```

```
colSums(is.na(restData))
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
##            0            0            0            0            0
```

```
all(colSums(is.na(restData))==0)
```

```
## [1] TRUE
```

valores especificos en columnas

```
table(restData$Species %in% c("setosa"))
```

```
##
## FALSE  TRUE
##   100    50
```

```
table(restData$Species %in% c("setosa","versicolor"))
```

```
##
## FALSE  TRUE
##    50   100
```

Tabulaciones cruzadas

```
data(UCBAdmissions)
DF = as.data.frame(UCBAdmissions)
summary(DF)
```

```
##       Admit       Gender   Dept       Freq
##  Admitted:12   Male  :12   A:4   Min.   :  8.0
##  Rejected:12   Female:12   B:4   1st Qu.: 80.0
##                            C:4   Median :170.0
##                            D:4   Mean   :188.6
##                            E:4   3rd Qu.:302.5
##                            F:4   Max.   :512.0
```

```
xt <- xtabs(Freq ~ Gender + Admit,data=DF)
xt
```

```
##         Admit
## Gender   Admitted Rejected
##   Male       1198     1493
##   Female      557     1278
```

tablas planas

```
warpbreaks$replicate <- rep(1:9, len = 54)
xt = xtabs(breaks ~.,data=warpbreaks)
xt
```

```
## , , replicate = 1
##
##     tension
## wool  L  M  H
##    A 26 18 36
##    B 27 42 20
##
## , , replicate = 2
##
##     tension
## wool  L  M  H
##    A 30 21 21
##    B 14 26 21
##
## , , replicate = 3
##
##     tension
## wool  L  M  H
##    A 54 29 24
##    B 29 19 24
##
## , , replicate = 4
##
##     tension
## wool  L  M  H
##    A 25 17 18
##    B 19 16 17
##
## , , replicate = 5
##
##     tension
## wool  L  M  H
##    A 70 12 10
##    B 29 39 13
##
## , , replicate = 6
##
##     tension
## wool  L  M  H
##    A 52 18 43
##    B 31 28 15
##
## , , replicate = 7
##
```

```
##     tension
## wool  L  M  H
##    A 51 35 28
##    B 41 21 15
##
## , , replicate = 8
##
##     tension
## wool  L  M  H
##    A 26 30 15
##    B 20 39 16
##
## , , replicate = 9
##
##     tension
## wool  L  M  H
##    A 67 36 26
##    B 44 29 28
```

```
ftable(xt)
```

```
##              replicate  1  2  3  4  5  6  7  8  9
## wool tension
## A    L                 26 30 54 25 70 52 51 26 67
##      M                 18 21 29 17 12 18 35 30 36
##      H                 36 21 24 18 10 43 28 15 26
## B    L                 27 14 29 19 29 31 41 20 44
##      M                 42 26 19 16 39 28 21 39 29
##      H                 20 21 24 17 13 15 15 16 28
```

tamaño de un dataset

```
fakeData = rnorm(1e5)
object.size(fakeData)
```

```
## 800048 bytes
```

```
print(object.size(fakeData),units="Mb")
```

```
## 0.8 Mb
```

Cambiar la forma de los datos

El objetivo son datos ordenados

1. Cada variable forma una columna
2. Cada observación forma una fila
3. Cada tabla / archivo almacena datos sobre un tipo de observación (por ejemplo, personas / hospitales).

```
library(reshape2)
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
```

```
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

fusion de dataframes

```
mtcars$carname <- rownames(mtcars)
carMelt <- melt(mtcars,id=c("carname","gear","cyl"),measure.vars=c("mpg","hp"))
head(carMelt,n=3)
```

```
##         carname gear cyl variable value
## 1     Mazda RX4    4   6      mpg  21.0
## 2 Mazda RX4 Wag    4   6      mpg  21.0
## 3    Datsun 710    4   4      mpg  22.8
```

```
tail(carMelt,n=3)
```

```
##          carname gear cyl variable value
## 62  Ferrari Dino    5   6       hp   175
## 63 Maserati Bora    5   8       hp   335
## 64    Volvo 142E    4   4       hp   109
```

casting de dataframe

```
cylData <- dcast(carMelt, cyl ~ variable)
cylData
```

```
##   cyl mpg hp
## 1   4  11 11
## 2   6   7  7
## 3   8  14 14
```

```
cylData <- dcast(carMelt, cyl ~ variable,mean)
cylData
```

```
##   cyl      mpg        hp
## 1   4 26.66364  82.63636
## 2   6 19.74286 122.28571
## 3   8 15.10000 209.21429
```

Valores promediados

```
head(InsectSprays)
```

```
##   count spray
## 1    10     A
## 2     7     A
## 3    20     A
## 4    14     A
## 5    14     A
## 6    12     A
```

```
tapply(InsectSprays$count,InsectSprays$spray,sum)
```

```
##   A   B   C   D   E   F
## 174 184  25  59  42 200
```

otra forma con split y apply

```
spIns =  split(InsectSprays$count,InsectSprays$spray)
spIns
```

```
## $A
```

```
##  [1] 10  7 20 14 14 12 10 23 17 20 14 13
##
## $B
##  [1] 11 17 21 11 16 14 17 17 19 21  7 13
##
## $C
##  [1] 0 1 7 2 3 1 2 1 3 0 1 4
##
## $D
##  [1]  3  5 12  6  4  3  5  5  5  5  2  4
##
## $E
##  [1] 3 5 3 5 3 6 1 1 3 2 6 4
##
## $F
##  [1] 11  9 15 22 15 16 13 10 26 26 24 13
```

```
sprCount = lapply(spIns,sum)
sprCount
```

```
## $A
## [1] 174
##
## $B
## [1] 184
##
## $C
## [1] 25
##
## $D
## [1] 59
##
## $E
## [1] 42
##
## $F
## [1] 200
```

con sapply

```
unlist(sprCount)
```

```
##   A   B   C   D   E   F
## 174 184  25  59  42 200
```

```
sapply(spIns,sum)
```

```
##   A   B   C   D   E   F
## 174 184  25  59  42 200
```

con paquete pylr

```
ddply(InsectSprays,.(spray),summarize,sum=sum(count))
```

```
##   spray sum
## 1     A 174
## 2     B 184
## 3     C  25
```

```
## 4     D  59
## 5     E  42
## 6     F 200
```

creando nuevas variables

```
spraySums <- ddply(InsectSprays,.(spray),summarize,sum=ave(count,FUN=sum))
dim(spraySums)
```

```
## [1] 72  2
```

```
head(spraySums)
```

```
##   spray sum
## 1     A 174
## 2     A 174
## 3     A 174
## 4     A 174
## 5     A 174
## 6     A 174
```

más informacion

- A tutorial from the developer of plyr - http://plyr.had.co.nz/09-user/
- A nice reshape tutorial http://www.slideshare.net/jeffreybreen/reshaping-data-in-r
- A good plyr primer - http://www.r-bloggers.com/a-quick-primer-on-split-apply-combine-problems/
- See also the functions
  - acast - for casting as multi-dimensional arrays
  - arrange - for faster reordering without using order() commands
  - mutate - adding new variables