4.DatosNA_leerArchivos_funcionesLoop

Abel Isaias Gutierrez-Cruz

9/8/2021

Remover valores NA

```
x <- c(1, 2, NA, 4, NA, 5)
bad <- is.na(x)
print(bad)
```

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

```
print(!bad)
```

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

```
x[!bad]
```

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

Usando la función `complete.cases`

```
x <- c(1, 2, NA, 4, NA, 5)
y <- c("a", "b", NA, "d", NA, 8)
good <- complete.cases(x, y)
print(y[good])
```

```
## [1] "a" "b" "d" "8"
```

```
print(x[good])
```

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

```
x <- c(1, 2, NA, 4, NA, 5)
y <- c("a", "b", NA, "d", NA, NA)
good <- complete.cases(x, y)
print(x[good])
```

```
## [1] 1 2 4
```

```
print(y[good])
```

```
## [1] "a" "b" "d"
```

Eliminar NA de dataframes

```
data("airquality")
good <- complete.cases(airquality)
datosLimpios <- airquality[good, ]
```

Descargar y leer archivos

Descargar

```
url <- "https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2Fss06hid.csv"
download.file(url, destfile = "../Data/SurveyHousing.csv", mode = "wb")
```

Leer archivos csv

```
data <- read.csv("../Data/SurveyHousing.csv")
```

Leer archivos separados por tabulaciones

```
data <- read.table("../Data/separadoPorTabulaciones.txt", sep = "\t", header = TRUE)
```

```
## Warning in read.table("../Data/separadoPorTabulaciones.txt", sep =
## "\t", : incomplete final line found by readTableHeader on '../Data/
## separadoPorTabulaciones.txt'
```

Leer archivos xlsx

```
fileUrl2 <- "https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2FDATA.gov_NGAP.xlsx"
# se tiene que establecer el mode = "wb"
download.file(fileUrl2, destfile = "../Data/NaturalGasAquisition.xlsx", mode = "wb")

library(xlsx)
```

```
## Warning: package 'xlsx' was built under R version 4.0.5
```

```
data <- read.xlsx("../Data/NaturalGasAquisition.xlsx", sheetIndex = 1,
                            header = TRUE, rowIndex = c(18, 19, 20, 21, 22, 23),
                            colIndex = c(7, 8, 9, 10 ,11, 12, 13, 14, 15))
```

Funciones para explorar archivos

```
print(head(data, 1))
```

```
##     Zip CuCurrent PaCurrent PoCurrent      Contact Ext          Fax email
## 1 74136         0         1         0 918-491-6998   0 918-491-6659    NA
##   Status
## 1      1
```

```
tail(data, 1)
```

```
##     Zip CuCurrent PaCurrent PoCurrent      Contact Ext  Fax email Status
## 5 80120         1         0         0 345-098-8890 456 <NA>    NA      1
```

```
summary(data)
```

```
##       Zip          CuCurrent     PaCurrent     PoCurrent   Contact
##  Min.   :30329   Min.   :0.0   Min.   :0.0   Min.   :0   Length:5
##  1st Qu.:74136   1st Qu.:0.0   1st Qu.:0.0   1st Qu.:0   Class :character
##  Median :74136   Median :1.0   Median :0.0   Median :0   Mode  :character
##  Mean   :67785   Mean   :0.6   Mean   :0.4   Mean   :0
##  3rd Qu.:80120   3rd Qu.:1.0   3rd Qu.:1.0   3rd Qu.:0
```

```
##  Max.   :80203   Max.   :1.0   Max.   :1.0   Max.   :0
##
##       Ext           Fax              email              Status
##  Min.   :  0   Length:5           Mode:logical   Min.   :1
##  1st Qu.:  0   Class :character   NA's:5         1st Qu.:1
##  Median :  0   Mode  :character                  Median :1
##  Mean   :114                                    Mean   :1
##  3rd Qu.:114                                    3rd Qu.:1
##  Max.   :456                                    Max.   :1
##  NA's   :1
```

```
data("airquality")
airquality
```

```
##     Ozone Solar.R Wind Temp Month Day
## 1      41     190  7.4   67     5   1
## 2      36     118  8.0   72     5   2
## 3      12     149 12.6   74     5   3
## 4      18     313 11.5   62     5   4
## 5      NA      NA 14.3   56     5   5
## 6      28      NA 14.9   66     5   6
## 7      23     299  8.6   65     5   7
## 8      19      99 13.8   59     5   8
## 9       8      19 20.1   61     5   9
## 10     NA     194  8.6   69     5  10
## 11      7      NA  6.9   74     5  11
## 12     16     256  9.7   69     5  12
## 13     11     290  9.2   66     5  13
## 14     14     274 10.9   68     5  14
## 15     18      65 13.2   58     5  15
## 16     14     334 11.5   64     5  16
## 17     34     307 12.0   66     5  17
## 18      6      78 18.4   57     5  18
## 19     30     322 11.5   68     5  19
## 20     11      44  9.7   62     5  20
## 21      1       8  9.7   59     5  21
## 22     11     320 16.6   73     5  22
## 23      4      25  9.7   61     5  23
## 24     32      92 12.0   61     5  24
## 25     NA      66 16.6   57     5  25
## 26     NA     266 14.9   58     5  26
## 27     NA      NA  8.0   57     5  27
## 28     23      13 12.0   67     5  28
## 29     45     252 14.9   81     5  29
## 30    115     223  5.7   79     5  30
## 31     37     279  7.4   76     5  31
## 32     NA     286  8.6   78     6   1
## 33     NA     287  9.7   74     6   2
## 34     NA     242 16.1   67     6   3
## 35     NA     186  9.2   84     6   4
## 36     NA     220  8.6   85     6   5
## 37     NA     264 14.3   79     6   6
## 38     29     127  9.7   82     6   7
## 39     NA     273  6.9   87     6   8
## 40     71     291 13.8   90     6   9
```

```
## 41     39     323 11.5   87     6  10
## 42     NA     259 10.9   93     6  11
## 43     NA     250  9.2   92     6  12
## 44     23     148  8.0   82     6  13
## 45     NA     332 13.8   80     6  14
## 46     NA     322 11.5   79     6  15
## 47     21     191 14.9   77     6  16
## 48     37     284 20.7   72     6  17
## 49     20      37  9.2   65     6  18
## 50     12     120 11.5   73     6  19
## 51     13     137 10.3   76     6  20
## 52     NA     150  6.3   77     6  21
## 53     NA      59  1.7   76     6  22
## 54     NA      91  4.6   76     6  23
## 55     NA     250  6.3   76     6  24
## 56     NA     135  8.0   75     6  25
## 57     NA     127  8.0   78     6  26
## 58     NA      47 10.3   73     6  27
## 59     NA      98 11.5   80     6  28
## 60     NA      31 14.9   77     6  29
## 61     NA     138  8.0   83     6  30
## 62    135     269  4.1   84     7   1
## 63     49     248  9.2   85     7   2
## 64     32     236  9.2   81     7   3
## 65     NA     101 10.9   84     7   4
## 66     64     175  4.6   83     7   5
## 67     40     314 10.9   83     7   6
## 68     77     276  5.1   88     7   7
## 69     97     267  6.3   92     7   8
## 70     97     272  5.7   92     7   9
## 71     85     175  7.4   89     7  10
## 72     NA     139  8.6   82     7  11
## 73     10     264 14.3   73     7  12
## 74     27     175 14.9   81     7  13
## 75     NA     291 14.9   91     7  14
## 76      7      48 14.3   80     7  15
## 77     48     260  6.9   81     7  16
## 78     35     274 10.3   82     7  17
## 79     61     285  6.3   84     7  18
## 80     79     187  5.1   87     7  19
## 81     63     220 11.5   85     7  20
## 82     16       7  6.9   74     7  21
## 83     NA     258  9.7   81     7  22
## 84     NA     295 11.5   82     7  23
## 85     80     294  8.6   86     7  24
## 86    108     223  8.0   85     7  25
## 87     20      81  8.6   82     7  26
## 88     52      82 12.0   86     7  27
## 89     82     213  7.4   88     7  28
## 90     50     275  7.4   86     7  29
## 91     64     253  7.4   83     7  30
## 92     59     254  9.2   81     7  31
## 93     39      83  6.9   81     8   1
## 94      9      24 13.8   81     8   2
```

```
## 95     16      77  7.4   82     8   3
## 96     78      NA  6.9   86     8   4
## 97     35      NA  7.4   85     8   5
## 98     66      NA  4.6   87     8   6
## 99    122     255  4.0   89     8   7
## 100    89     229 10.3   90     8   8
## 101   110     207  8.0   90     8   9
## 102    NA     222  8.6   92     8  10
## 103    NA     137 11.5   86     8  11
## 104    44     192 11.5   86     8  12
## 105    28     273 11.5   82     8  13
## 106    65     157  9.7   80     8  14
## 107    NA      64 11.5   79     8  15
## 108    22      71 10.3   77     8  16
## 109    59      51  6.3   79     8  17
## 110    23     115  7.4   76     8  18
## 111    31     244 10.9   78     8  19
## 112    44     190 10.3   78     8  20
## 113    21     259 15.5   77     8  21
## 114     9      36 14.3   72     8  22
## 115    NA     255 12.6   75     8  23
## 116    45     212  9.7   79     8  24
## 117   168     238  3.4   81     8  25
## 118    73     215  8.0   86     8  26
## 119    NA     153  5.7   88     8  27
## 120    76     203  9.7   97     8  28
## 121   118     225  2.3   94     8  29
## 122    84     237  6.3   96     8  30
## 123    85     188  6.3   94     8  31
## 124    96     167  6.9   91     9   1
## 125    78     197  5.1   92     9   2
## 126    73     183  2.8   93     9   3
## 127    91     189  4.6   93     9   4
## 128    47      95  7.4   87     9   5
## 129    32      92 15.5   84     9   6
## 130    20     252 10.9   80     9   7
## 131    23     220 10.3   78     9   8
## 132    21     230 10.9   75     9   9
## 133    24     259  9.7   73     9  10
## 134    44     236 14.9   81     9  11
## 135    21     259 15.5   76     9  12
## 136    28     238  6.3   77     9  13
## 137     9      24 10.9   71     9  14
## 138    13     112 11.5   71     9  15
## 139    46     237  6.9   78     9  16
## 140    18     224 13.8   67     9  17
## 141    13      27 10.3   76     9  18
## 142    24     238 10.3   68     9  19
## 143    16     201  8.0   82     9  20
## 144    13     238 12.6   64     9  21
## 145    23      14  9.2   71     9  22
## 146    36     139 10.3   81     9  23
## 147     7      49 10.3   69     9  24
## 148    14      20 16.6   63     9  25
```

```
## 149    30     193  6.9   70     9  26
## 150    NA     145 13.2   77     9  27
## 151    14     191 14.3   75     9  28
## 152    18     131  8.0   76     9  29
## 153    20     223 11.5   68     9  30
```

```
head(airquality)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

```
tail(airquality)
```

```
##     Ozone Solar.R Wind Temp Month Day
## 148    14      20 16.6   63     9  25
## 149    30     193  6.9   70     9  26
## 150    NA     145 13.2   77     9  27
## 151    14     191 14.3   75     9  28
## 152    18     131  8.0   76     9  29
## 153    20     223 11.5   68     9  30
```

```
print(summary(airquality))
```

```
##      Ozone           Solar.R           Wind             Temp
##  Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00
##  1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00
##  Median : 31.50   Median :205.0   Median : 9.700   Median :79.00
##  Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88
##  3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00
##  Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00
##  NA's   :37       NA's   :7
##      Month            Day
##  Min.   :5.000   Min.   : 1.0
##  1st Qu.:6.000   1st Qu.: 8.0
##  Median :7.000   Median :16.0
##  Mean   :6.993   Mean   :15.8
##  3rd Qu.:8.000   3rd Qu.:23.0
##  Max.   :9.000   Max.   :31.0
##
```

```
str(airquality)
```

```
## 'data.frame':    153 obs. of  6 variables:
##  $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
##  $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
##  $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
##  $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
##  $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
quantile(airquality, na.rm = TRUE, probs = c(0.2, 0.4, 0.7))
```

```
##   20%   40%   70%
```

```
##  7.40 12.12 72.00
```

```
table(airquality$Ozone)
```

```
##
##   1   4   6   7   8   9  10  11  12  13  14  16  18  19  20  21  22  23  24  27
##   1   1   1   3   1   3   1   3   2   4   4   4   4   1   4   4   1   6   2   1
##  28  29  30  31  32  34  35  36  37  39  40  41  44  45  46  47  48  49  50  52
##   3   1   2   1   3   1   2   2   2   2   1   1   3   2   1   1   1   1   1   1
##  59  61  63  64  65  66  71  73  76  77  78  79  80  82  84  85  89  91  96  97
##   2   1   1   2   1   1   1   2   1   1   2   1   1   1   1   2   1   1   1   2
## 108 110 115 118 122 135 168
##   1   1   1   1   1   1   1
```

```
data <- read.csv("../Data/titanic.csv")
table(data$Sex, data$Survived)
```

```
##
##            0   1
##   female  81 233
##   male   468 109
```

```
table(data$Sex)
```

```
##
## female   male
##    314    577
```

Funciones bucle (loop functions)

Lapply

Obtener el promedio de cada uno de los elementos de la lista

```
x <- list(a = 1:5, b = rnorm(10))
print(x)
```

```
## $a
## [1] 1 2 3 4 5
##
## $b
##  [1] -2.52798487 -0.65349686  1.03009314 -0.04429052 -0.56446608  1.02959949
##  [7]  0.39343343  0.10596717 -0.15721573  1.74272369
```

```
print(lapply(x, mean))
```

```
## $a
## [1] 3
##
## $b
## [1] 0.03543629
```

Forma con la función Lapply

```
print(x)
```

```
## $a
## [1] 1 2 3 4 5
##
## $b
```

```
##  [1] -2.52798487 -0.65349686  1.03009314 -0.04429052 -0.56446608  1.02959949
##  [7]  0.39343343  0.10596717 -0.15721573  1.74272369
```

```
print("")
```

```
## [1] ""
```

```
print(lapply(x, runif, min = 0, max = 10))
```

```
## $a
## [1] 7.188122 1.101717 3.344020 6.017340 3.260226
##
## $b
##  [1] 0.12057034 1.64435215 7.53727477 0.06289342 3.21742475 5.01498132
##  [7] 6.95518197 9.44616955 2.53544228 1.55789189
```

```
resultado <- lapply(x, runif, min = 0, max = 10)
class(resultado)
```

```
## [1] "list"
```

Forma manual:

```
runif(1, min = 0, max = 10)
```

```
## [1] 2.346744
```

```
runif(2, min = 0, max = 10)
```

```
## [1] 0.7440182 3.8379511
```

```
runif(3, min = 0, max = 10)
```

```
## [1] 3.086248 6.550231 1.359636
```

```
runif(4, min = 0, max = 10)
```

```
## [1] 3.3665871 6.4943117 0.5934895 6.7119261
```

Sapply

```
# aplicando la función lapply

x <- rep(2, 4)

resultadoL <- lapply(x, runif, min = 0, max = 10)
print(class(resultadoL))
```

```
## [1] "list"
```

```
resultadoS <- sapply(x, runif, min = 0, max = 10)
print(class(resultadoS))
```

```
## [1] "matrix" "array"
```

Apply

Obtener el promedio de cada una de las columnas

```
x <- matrix(rnorm(200), 20, 10)
apply(x, 2, mean)
```

```
##  [1] -0.072187458  0.013169574  0.050229863  0.044309356 -0.001335803
##  [6] -0.210722653  0.110647259 -0.257474558 -0.113667336  0.193949854
```

Obtener el promedio de cada una de las filas

```
apply(x, 1, mean)
```

```
##  [1]  0.36079836  0.11943886  0.03329672 -0.20411877 -0.12163022  0.34010075
##  [7] -0.26319356 -0.27196002 -0.31533449 -0.27697017  0.20150062 -0.04023657
## [13]  0.33859140  0.04780774 -0.50047033 -0.12464818 -0.25475454  0.12523711
## [19] -0.30953883  0.62992033
```

Calcular cuartiles

```
x <- matrix(rnorm(200), 20, 10)
apply(x, 1, quantile, probs = c(0.25, 0.75))
```

```
##            [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
## 25% -0.07175727 -0.7854470 -0.3755239 -0.4395449 -0.9355069 -0.9510869
## 75%  0.45691346  0.3249956  0.9119073  0.8534294  1.2921422  0.8471406
##           [,7]       [,8]      [,9]       [,10]      [,11]      [,12]
## 25% -0.5999727 -0.5021437 0.1093405 -0.09940288 -0.9473942 -0.2780352
## 75%  0.2388903  0.2464658 0.6402450  0.47671238  0.4525667  1.1599879
##          [,13]        [,14]      [,15]      [,16]      [,17]      [,18]
## 25% -0.2822969 -0.701921399 -1.0022498 -1.2057799 -0.5825279 -0.3525442
## 75%  0.6378600 -0.001265098  0.2014583  0.1787133  0.5909597  0.1608935
##         [,19]      [,20]
## 25% -0.429629 -0.9896017
## 75%  0.498245  0.3341165
```

Mapply

Generar una lista de numeros, queremos que se repite cuatro veces el número 1, que se repita 3 veces el numero 2, dos veces el número 3 y una vez el número 4 Forma tediosa de hacerlo:

```
print(list(rep(1, 4), rep(2, 3), rep(3, 2), rep(4, 1)))
```

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

Forma fácil al utilizar la función mapply

```
mapply(rep, 1:4, 4:1)
```

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

```
## [[3]]
## [1] 3 3
##
## [[4]]
## [1] 4
```

Tapply

```
# Generar 30 elementos números
x <- c(rnorm(10), runif(10), rnorm(10, 1))
f <- gl(3, 10)
tapply(x, f, mean)
```

```
##          1          2          3
## -0.5249881  0.5067349  1.2080481
```

Split

```
# Generar 30 elementos números
x <- c(rnorm(10), runif(10), rnorm(10, 1))
f <- gl(3, 10)
# separación del vector en grupos
split(x, f)
```

```
## $`1`
##  [1] -0.09380522  0.72006660  0.05787103  1.27234351  0.93183456  0.82133519
##  [7]  0.28756458  1.08132546  2.75707936  1.11415104
##
## $`2`
##  [1] 0.86699098 0.44881757 0.37362355 0.56619083 0.66350723 0.43250440
##  [7] 0.03190223 0.02509599 0.72912543 0.95690105
##
## $`3`
##  [1]  1.0415623352 -0.3320746638  1.3278159932  2.1061552040  0.3608814006
##  [6]  2.6185170365  0.8571643273  1.6438850074  1.0671115761 -0.0001441013
```

```
lapply(split(x, f), mean)
```

```
## $`1`
## [1] 0.8949766
##
## $`2`
## [1] 0.5094659
##
## $`3`
## [1] 1.069087
```

```
data("airquality")
# separar dataframe conforme a los valores en la variable "Month"
s <- split(airquality, airquality$Month)
lapply(s, function(x) colMeans(x[, c("Ozone", "Solar.R", "Wind")], na.rm = TRUE))
```

Aplicación de la función split en un dataframe

```
## $`5`
```

```
##     Ozone   Solar.R      Wind
##  23.61538 181.29630  11.62258
##
## $`6`
##     Ozone   Solar.R      Wind
##  29.44444 190.16667  10.26667
##
## $`7`
##      Ozone    Solar.R       Wind
##  59.115385 216.483871   8.941935
##
## $`8`
##      Ozone    Solar.R       Wind
##  59.961538 171.857143   8.793548
##
## $`9`
##     Ozone   Solar.R      Wind
##  31.44828 167.43333  10.18000
```