Mt1 2018

Irlanda Ayon-Moreno

5/4/2018

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
```

```
library(ggplot2)
```

Explained Errors

```
dat <- data.frame(
  first = c("Jon", "Arya", "Tyrion", "Daenerys", "Yara"),
  last = c("Snow", "Stark", "Lannister", "Targaryen", "Greyjoy"),
  gender = c("male", "female", "male", "female", "female"),
  title = c("lord", "princess", "master", "khaleesi", "princess"),
  gpa = c(2.8, 3.5, 2.9, 3.7, NA),
  stringsAsFactors = FALSE
)
dat
```

```
##      first      last gender    title gpa
## 1      Jon      Snow   male     lord 2.8
## 2     Arya     Stark female princess 3.5
## 3   Tyrion Lannister   male   master 2.9
## 4 Daenerys Targaryen female khaleesi 3.7
## 5     Yara   Greyjoy female princess  NA
```

value of 'first' for maximum 'gpa'

```
max_gpa <- max(dat$gpa, na.rm = TRUE)
max_gpa
```

```
## [1] 3.7
```

```
which_max_gpa <- dat$gpa = max_gpa
```

```
## Error in which_max_gpa <- dat$gpa = max_gpa: object 'which_max_gpa' not found
```

```
# corrected
which_max_gpa <- dat$gpa == max_gpa
which_max_gpa
```

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

```
dat$first(which_max_gpa)
```

```
## Error in eval(expr, envir, enclos): attempt to apply non-function
```

```
# corrected
dat$first[which_max_gpa]
```

```
## [1] "Daenerys" NA
```

gpa of title lord

```
dat[ , c("title", "gpa")]
```

```
##      title gpa
## 1     lord 2.8
## 2 princess 3.5
## 3   master 2.9
## 4 khaleesi 3.7
## 5 princess  NA
```

```
dat[ , "title"]
```

```
## [1] "lord"     "princess" "master"   "khaleesi" "princess"
```

```
dat[ , c("title", "gpa")] == "lord"
```

```
##      title   gpa
## [1,]  TRUE FALSE
## [2,] FALSE FALSE
## [3,] FALSE FALSE
## [4,] FALSE FALSE
## [5,] FALSE    NA
```

```
dat[ , "title"] = "lord"
dat
```

```
##      first      last gender title gpa
## 1      Jon      Snow   male  lord 2.8
## 2     Arya     Stark female  lord 3.5
## 3   Tyrion Lannister   male  lord 2.9
## 4 Daenerys Targaryen female  lord 3.7
## 5     Yara   Greyjoy female  lord  NA
```

```
dat$gpa[dat[ , "title"] = "lord"]
```

```
## Error: <text>:1:25: unexpected '='
## 1: dat$gpa[dat[ , "title"] =
##                            ^
```

```
# corrected
dat$gpa[dat[ , "title"] == "lord"]
```

```
## [1] 2.8 3.5 2.9 3.7  NA
```

median gpa of each gender

```
which_males <- dat$gender == 'male'
which_females <- dat$gender == 'female'
```

```
# with mistake
median_females <- median(dat$gpa[which_males])
median_males <- median(dat$gpa[which_males])
c(median_females, median_males)
```

```
## [1] 2.85 2.85
```

```
# corrected
median_females <- median(dat$gpa[which_females], na.rm = TRUE)
median_males <- median(dat$gpa[which_males])
c(median_females, median_males)
```

```
## [1] 3.60 2.85
```

Command Outputs

```
student <- list(
  name = "Anakin Skywalker",
  gpa = 4,
  major_minor = c(major1 = "jedi studies", major2 = "sith studies", minor = "galactic policies"
  grades = data.frame(
    course = c("force-101", "podracing", "light-sabers"),
    score = c(9.3, 10.0, 8.5),
    stringsAsFactors = FALSE
  )
)
student
```

```
## $name
## [1] "Anakin Skywalker"
##
## $gpa
## [1] 4
##
## $major_minor
##              major1              major2               minor
##      "jedi studies"      "sith studies" "galactic policies"
##
## $grades
##         course score
```

```
## 1    force-101   9.3
## 2    podracing  10.0
## 3 light-sabers   8.5
```

```
length(student$major_minor)
```

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

```
student$gpa < 2.5
```

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

```
names(student$major_minor)
```

```
## [1] "major1" "major2" "minor"
```

```
student$grades[2]
```

```
##   score
## 1   9.3
## 2  10.0
## 3   8.5
```

```
student$grades[[2]]
```

```
## [1]  9.3 10.0  8.5
```

```
student$grades[[2]][2]
```

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

```
rep(student$grades[[2]][2], student$gpa)
```

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

```
student$grades %>% arrange(score)
```

```
##         course score
## 1 light-sabers   8.5
## 2    force-101   9.3
## 3    podracing  10.0
```

Data Types

```
uno <- c(TRUE, FALSE)
uno
```

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

```
typeof(uno)
```

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

```r
dos <- uno + 1L
dos
```

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

```r
typeof(dos)
```

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

```r
tres <- c(uno, dos, "3.0", "4.0")
tres
```

```
## [1] "TRUE"  "FALSE" "2"     "1"     "3.0"   "4.0"
```

```r
typeof(tres)
```

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

```r
cuatro <- as.factor(tres)
cuatro
```

```
## [1] TRUE  FALSE 2     1     3.0   4.0
## Levels: 1 2 3.0 4.0 FALSE TRUE
```

```r
typeof(cuatro)
```

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

```r
uno[1]
```

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

```r
dos[uno[1]]
```

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

```r
cinco <- tres[dos[uno[1]]]
cinco
```

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

```r
typeof(cinco)
```

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

Markdown

Examples

Text in italics:

italics

Text in bold:

bold

Text that is associated to a (hyper)link:

link title

Text in code format:

- code chunck: Make a code chunk with three back ticks followed by an r in braces. End the chunk with three back ticks

```
four <- 4
```

- inline code: use back ticks around the code. include an r after the first back tick so that the code updates itself.

  `four` or 4

How to write unordered bulleted list:

Use - or + or * & end each line with 2 spaces

How to write ordered bulleted list:

Use numbers & end each line with 2 spaces

Vector Subsetting

```
lord <- c("v", "o", "l", "d", "e", "m", "o", "r", "t")
```

```
names(lord) = 1:9
lord
```

```
##   1   2   3   4   5   6   7   8   9
## "v" "o" "l" "d" "e" "m" "o" "r" "t"
```

```
lord[length(lord)]
```

```
##   9
## "t"
```

```
as.logical(-1:-4)
```

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

```
lord[as.logical(-1:-4)]
```

```
##   1   2   3   4   5   6   7   8   9
## "v" "o" "l" "d" "e" "m" "o" "r" "t"
```

```
lord[seq(from = 1, to = 9, by = 2)]
```

```
##   1   3   5   7   9
## "v" "l" "e" "o" "t"
```

```
lord[lord != "e" | lord == "o"]
```

```
##   1   2   3   4   6   7   8   9
## "v" "o" "l" "d" "m" "o" "r" "t"
```

```
lord[3:1]
```

```
##   3   2   1
## "l" "o" "v"
```

```
lord[3:1] == lord[3:1]
```

```
##    3    2    1
## TRUE TRUE TRUE
```

Write code that returns the elements: “d” “o” “o” “r”

```
lord[c(4, 2, 7, 8)]
```

```
##   4   2   7   8
## "d" "o" "o" "r"
```

Explain Commands

What does `sum(!is.finite(x))` say about `x`?

```
x <- seq(from = 0, to = 12, by = 2)
!is.finite(x)
```

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

```
sum(!is.finite(x))
```

```
## [1] 0
```

```
x <- c(0, 1, 2, 3, 4, pi/0, 1/0) # a non-zero number divided by 0 creates infinity
!is.finite(x)
```

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

```
sum(!is.finite(x))
```

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

```
# Tells the number of infinite values.
# 0 if x is finite
```

COnsider a numeric vector `x`. What type of subsetting operaations are involved in the command: `x[-which(x > 0)]`?

```
# numeric subsetting

x <- c(-4, -3, -2, -1, 0, 1, 2, 3, 4)
x > 0
```

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

```
which(x > 0)
```

```
## [1] 6 7 8 9
```

```
-which(x > 0)
```

```
## [1] -6 -7 -8 -9
```

```
x[-which(x > 0)]
```

```
## [1] -4 -3 -2 -1  0
```

A data frame `A` and a matrix `B` contain exactly the same tabular data. The first column of `A` and `B` is named `Col1`. Why does `B$Col1` return an error?

```
B <- matrix(c(1,2,3, 11,12,13), nrow = 2, ncol = 3, byrow = TRUE,
            dimnames = list(c("row1", "row2"),
                            c("Col1", "Col2", "Col3"))
)
B
```

```
##      Col1 Col2 Col3
## row1    1    2    3
## row2   11   12   13
```

```
A <- data.frame(B)
A
```

```
##      Col1 Col2 Col3
## row1    1    2    3
## row2   11   12   13
```

```
# $ is used on data frames (lists)
# [] may be uces on data frames (lists) & On Matrices
#     [row idex, column index]

A$Col1
```

```
## [1]  1 11
```

```
B$Col1
```

```
## Error in B$Col1: $ operator is invalid for atomic vectors
```

```
A[ , 1]
```

```
## [1]  1 11
```

```
B[ , 1]
```

```
## row1 row2
##    1   11
```

dplyr

```
autos <- data.frame(
  Model = as.character(c("pontiac firebird", "pontiac safari", NA, "opel 1900", "peugeot 304",
  MPG = as.double(c(19.0, NA, 30.0, NA, 30.0, 27.0, 35.0, NA)),
  Cylinders = as.integer(c(6, 8, rep(4, 6))),
  Weight = as.integer(c(NA, 5140, 2065, 2123, 2074, 1834, 1613, 2130)),
  Accelerate = as.double(c(15.0, 12.0, 14.5, NA, 19.5, NA, 18.0, 14.5)),
  Origin = as.character(c(rep("American", 2), rep("European", 4), rep("Japanese", 2)))
)
autos
```

```
##                  Model MPG Cylinders Weight Accelerate   Origin
## 1     pontiac firebird  19         6     NA       15.0 American
## 2       pontiac safari  NA         8   5140       12.0 American
## 3                 <NA>  30         4   2065       14.5 European
## 4           opel 1900  NA         4   2123         NA European
## 5         peugeot 304  30         4   2074       19.5 European
## 6 volkswagen model 111  27         4   1834         NA European
## 7                 <NA>  35         4   1613       18.0 Japanese
## 8         datsun p1510  NA         4   2130       14.5 Japanese
```

Display the `Model` and `Weight` of all `European` cars with `MPG` larger than 25. The output should be ordered from largest to smallest (in weight).

```
autos %>%
  filter(Origin == "European" & MPG > 25) %>%
  select(c(Model, Weight)) %>%
  arrange(desc(Weight))
```

```
##                  Model Weight
## 1         peugeot 304   2074
## 2                 <NA>   2065
## 3 volkswagen model 111   1834
```

ggplot2

Create a scatterplot of `mpg` and `hp`, with the size of the dots depending on `cyl`, and faceting by `transmission`.

```
dataset <- data.frame(
  row.names = c("Mazda RX4", "Mazda RX4 Wag", "Dotsun 710", "Hornest 4 Drive", "Hornest Sportal
  mpg = c(21, 21, 22.8, 21.4, 18.7),
  cyl = c(6, 6, 4, 6, 8),
  hp = c(110, 110, 93, 110, 175),
  transmission = c(rep("automatic", 3), rep("manual", 2))
)
dataset
```

```
##                    mpg cyl  hp transmission
## Mazda RX4         21.0   6 110    automatic
## Mazda RX4 Wag     21.0   6 110    automatic
## Dotsun 710        22.8   4  93    automatic
## Hornest 4 Drive   21.4   6 110       manual
## Hornest Sportabout 18.7   8 175       manual
```

```
ggplot(data = dataset, mapping = aes(x = mpg, y = hp)) +
  geom_point(aes(size = cyl)) +
  facet_grid(.~transmission)
```