Clase 4.0
Análisis

Marcos Rosetti y Luis Pacheco-Cobos

Estadística y Manejo de Datos con R (EMDR) — Virtual

Preprocesamiento

Preprocesamiento

- Para revisar los datos cargados usamos:

```
library(readr)
datos <- read.csv("mis_datos.csv")

head(datos) # primeras 10 líneas

tail(datos) # últimas 10 líneas

str(datos) # tipo, nombre y contenido del data frame

summary(datos) # breve resumen estadístico

dim(datos) # número de columnas y de filas
```

Preprocesamiento

- Errores comunes: Fechas

```
library(lubridate)
date1 <- c("12-01-99")
dmy(date1)
```

```
## [1] "1999-01-12"
```

```
date2 <- c("01/12/99")
mdy(date2)
```

```
## [1] "1999-01-12"
```

```
date3 <- c("January 12, 1999")
mdy(date3)
```

```
## [1] "1999-01-12"
```

Preprocesamiento

- Errores comunes: Hora

```
library(lubridate)
time1 <- c("15_30_30")
time1 <- hms(time1)
hour(time1)
```

```
## [1] 15
```

```
minute(time1)
```

```
## [1] 30
```

```
as.duration(time1)
```

```
## [1] "55830s (~15.51 hours)"
```

Preprocesamiento

- Errores comunes: Números como texto

```
library(readr)
x <- c("7", "7*", " 7.0", "7/0")
parse_number(x)
```

```
## [1] 7 7 7 7
```

Preprocesamiento

- Nombres de columnas de un conjunto datos cargado en R.

```
setwd("Clase4_files/")
data <- read.table("example.csv", sep = ",") # sin header
head(data, 3) # ¿cómo se ve el data frame?
data <- read.table("example2.csv" ,sep = ",", header = T) # con header
head(data, 3) # ¿cómo se ve data frame ahora?
```

Preprocesamiento

- Codificación de variables categóricas.

```r
sex <- c(0,1,1,1,0,1) # Bad
sex <- c("Female", "Male", "Male", "Male", "Female", "Male") # Good
cond <- c(0,1,1,1,0,1) # Bad
cond <- c("Ctrl", "Exp", "Exp", "Exp", "Ctrl", "Exp") # Good
```

Preprocesamiento

- Detección e imputación de datos faltantes.

```
naq <- airquality[complete.cases(airquality), ] # solucion paquete base
head(naq)
```

```
##   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
## 7    23     299  8.6   65     5   7
## 8    19      99 13.8   59     5   8
```

Preprocesamiento

- Detección e imputación de datos faltantes.

```
naq <- na.omit(airquality) # otra alternativa
head(naq)
```

```
##   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
## 7    23     299  8.6   65     5   7
## 8    19      99 13.8   59     5   8
```

Preprocesamiento

- Detección e imputación de datos faltantes.

```
naq <- airquality %>% drop_na() # solucion con dplyr
head(naq)
```

```
##   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    23     299  8.6   65     5   7
## 6    19      99 13.8   59     5   8
```

Preprocesamiento

- Detección e imputación de datos faltantes.

```
naq <- airquality
naq$Ozone[which(is.na(naq$Ozone))] <- mean(naq$Ozone, na.rm = TRUE) # inteta median()
head(naq)
```

```
##      Ozone Solar.R Wind Temp Month Day
## 1 41.00000     190  7.4   67     5   1
## 2 36.00000     118  8.0   72     5   2
## 3 12.00000     149 12.6   74     5   3
## 4 18.00000     313 11.5   62     5   4
## 5 42.12931      NA 14.3   56     5   5
## 6 28.00000      NA 14.9   66     5   6
```

Preprocesamiento

- Detección e imputación de datos faltantes.

```
library(Hmisc)
naq <- airquality
naq$Ozone <- impute(naq$Ozone, fun = mean) # usando la función impute()
head(naq)
```

```
##      Ozone Solar.R Wind Temp Month Day
## 1 41.00000     190  7.4   67     5   1
## 2 36.00000     118  8.0   72     5   2
## 3 12.00000     149 12.6   74     5   3
## 4 18.00000     313 11.5   62     5   4
## 5 42.12931      NA 14.3   56     5   5
## 6 28.00000      NA 14.9   66     5   6
```

Preprocesamiento

- Detección e imputación de datos faltantes.

```
naq <- airquality
# use randomly generated values
naq$Ozone[which(is.na(naq$Ozone))] <- rnorm(n =
                                        mean = mean(naq$Ozone, na.rm = T),
                                        sd = sd(naq$Ozone, na.rm = T))
head(naq)
```

```
##      Ozone Solar.R Wind Temp Month Day
## 1 41.00000     190  7.4   67     5   1
## 2 36.00000     118  8.0   72     5   2
## 3 12.00000     149 12.6   74     5   3
## 4 18.00000     313 11.5   62     5   4
## 5 52.93144      NA 14.3   56     5   5
## 6 28.00000      NA 14.9   66     5   6
```

Preprocesamiento

- Detección y `capping` de datos extremos.

```
x <- c(-30, 1:10, 20, 30)
```

```
boxplot(x)
```

```
boxplot.stats(x)$out
```

```
## [1] -30  20  30
```

```
boxplot.stats(x, coef = 2)$out # un criterio mas amplio
```

```
## [1] -30  30
```

Preprocesamiento

- Detección y **`capping`** de datos extremos.

```
x <- c(-30, 1:10, 20, 30)
qnt <- quantile(x, probs=c(.25, .75), na.rm = T) # cuantiles
caps <- quantile(x, probs=c(.15, .85), na.rm = T) # elige tis "caps"
H <- 1.5 * IQR(x, na.rm = T)
x[x < (qnt[1] - H)] <- caps[1] # remplaza outliers por max caps
x[x > (qnt[2] + H)] <- caps[2] # remplaza outliers por min caps
x
```

```
##  [1]  1.8  1.0  2.0  3.0  4.0  5.0  6.0  7.0  8.0  9.0 10.0 12.0 12.0
```

```
boxplot(x)
```

Preprocesamiento

- Estandarización y escalamiento con `scale`.

```
x <- c(1, 2, 3)
scale(x, center = T)[1:3]
```

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

```
(x - mean(x)) / sd(x) # manual
```

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

```
scale(x, center = F)[1:3]
```

```
## [1] 0.3779645 0.7559289 1.1338934
```

```
x / sqrt(sum(x^2) / (length(x) - 1)) # manual
```

```
## [1] 0.3779645 0.7559289 1.1338934
```

Licencia CC BY