

NotasR

AVallarino

Verificar instalación de paquete

Valida si un paquete se encuentra instalado. Si no lo está, lo instala.

```
if(!require(dplyr, quietly = TRUE, warn.conflicts = FALSE) ){  
  install.packages('dplyr',  
    dependencies = TRUE,  
    repos = "http://cran.us.r-project.org")  
}
```

```
## Warning: package 'dplyr' was built under R version 3.3.2
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.3.2
```

```
## Loading tidyverse: ggplot2  
## Loading tidyverse: tibble  
## Loading tidyverse: tidyr  
## Loading tidyverse: readr  
## Loading tidyverse: purrr
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
## Warning: package 'tibble' was built under R version 3.3.2
```

```
## Warning: package 'tidyr' was built under R version 3.3.2
```

```
## Warning: package 'readr' was built under R version 3.3.2
```

```
## Warning: package 'purrr' was built under R version 3.3.2
```

```
## Conflicts with tidy packages -----
```

```
## filter(): dplyr, stats  
## lag():    dplyr, stats
```

Caputrar excepción al leer un archivo:

Lee un archivo en el directorio de trabajo. Si el archivo no existe, lo descarga al directorio y lo abre.

```
# Armo array con nombres:
c_names <- c("CRIM", "ZN", "INDUS", "CHAS", "NOX", "RM", "AGE", "DIS", "RAD", "TAX", "PTRATIO", "B", "LSTAT", "MEDV")

# Cargo archivo de datos y agrego nombres de columnas:
dat_housing <- try(read.table("housing.dat", header = FALSE, col.names = c_names))

if (inherits(dat_housing, "try-error")) {
  # Descargo datos en directorio de trabajo - Indico nombre destino:
  download.file(
    "https://archive.ics.uci.edu/ml/machine-learning-databases/housing/housing.data",
    "housing.dat")
  dat_housing <- read.table("housing.dat", header = FALSE, col.names = c_names)
}

head(dat_housing)
```

```
##      CRIM  ZN  INDUS  CHAS    NOX    RM  AGE    DIS  RAD  TAX  PTRATIO    B
## 1 0.00632 18   2.31     0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90
## 2 0.02731  0   7.07     0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90
## 3 0.02729  0   7.07     0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83
## 4 0.03237  0   2.18     0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63
## 5 0.06905  0   2.18     0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90
## 6 0.02985  0   2.18     0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12
##    LSTAT  MEDV
## 1   4.98 24.0
## 2   9.14 21.6
## 3   4.03 34.7
## 4   2.94 33.4
## 5   5.33 36.2
## 6   5.21 28.7
```

Descargo .ZIP y lo descomprimo

Opción 1:

Del .zip obtengo lista de archivos .csv:

```
# URL con .zip
url <- "https://www.dropbox.com/s/7q8ohggjm8bw5m2/02-tarea.zip?dl=1"

# Defino directorios temporales:
temp <- tempfile()
temp2 <- tempfile()

# Obtengo archivo .zip y lo descomprimo:
download.file(url, temp)
unzip(zipfile = temp, exdir = temp2)

# Obtengo listado de archivos:
temp3 <- paste(temp2, "/02-tarea", sep = "")
csv_list <- list.files(temp3, pattern = "*.csv")

print("- Lista de archivos .CSV:")
```

```
## [1] "- Lista de archivos .CSV:"
```

```
csv_list
```

```
## [1] "2000.csv" "2001.csv" "2002.csv" "2003.csv" "2004.csv" "2005.csv"
## [7] "2006.csv" "2007.csv" "2008.csv"
```

Cargo listado de archivos del directorio descomprimido

```
# Obtengo listado de archivos (nombre completo)
paths <- dir(temp3, pattern = "\\*.csv$", full.names = TRUE)
paths <- set_names(paths, basename(paths))

# Cargo los datos de todos los .csv en un solo DataFrame:
data_200x_orig <- map_df(paths, ~read_csv(., col_types = cols(
  edo = col_character(),
  trans_Total = col_double(),
  trans_Mujeres = col_double(),
  trans_Hombres = col_double(),
  noTrans_Total = col_double(),
  noTrans_Mujeres = col_double(),
  noTrans_Hombres = col_double(),
  lesiones_Total = col_double(),
  lesiones_Mujeres = col_double(),
  lesiones_Hombres = col_double()))), .id = "filename")

head(data_200x_orig)
```

```
## # A tibble: 6 x 11
##   filename          edo trans_Total trans_Mujeres trans_Hombres
##   <chr>            <chr>      <dbl>      <dbl>      <dbl>
## 1 2000.csv      Aguascalientes    63.0      54.8      72.5
## 2 2000.csv      Baja California    89.2      63.9     115.1
## 3 2000.csv Baja California Sur    76.6      61.5      91.3
## 4 2000.csv      Campeche        54.6      44.9      64.1
## 5 2000.csv      Coahuila        60.0      51.7      69.3
## 6 2000.csv      Colima         61.1      51.0      72.1
## # ... with 6 more variables: noTrans_Total <dbl>, noTrans_Mujeres <dbl>,
## #   noTrans_Hombres <dbl>, lesiones_Total <dbl>, lesiones_Mujeres <dbl>,
## #   lesiones_Hombres <dbl>
```

Opción 2:

Cargo archivos .csv del directorio descomprimido:

```
# URL con .zip
url <- "https://data.diegovalle.net/hoyodecrimen/cuadrantes.csv.zip"

# Defino directorios temporales:
temp <- tempfile()
temp2 <- tempfile()

# Obtengo archivo .zip y lo descomprimo:
download.file(url, temp)
unzip(zipfile = temp, exdir = temp2)

# Obtengo archivos con los datos:
crime_lat_long <- read_csv(file.path(temp2, "/clean-data/crime-lat-long.csv"), col_types
= cols())
```

```
## Warning in rbind(names(probs), probs_f): number of columns of result is not
## a multiple of vector length (arg 1)
```

```
## Warning: 10 parsing failures.
## row # A tibble: 5 x 5 col      row    col   expected actual expected   <int> <chr>
<chr> <chr> actual 1 20223  hour time like  21510 file 2 59310  hour time like  15470
row 3 61494  hour time like  23190 col 4 61496  hour time like  00250 expected 5 61497
  hour time like  10490 actual # ... with 1 more variables: file <chr>
## ... ..
.....
.....
.....
.....
## See problems(...) for more details.
```

```
crime_lat_long <- arrange(crime_lat_long, date, cuadrante, crime)
head(crime_lat_long, n=10)
```

```
## # A tibble: 10 x 8
##   cuadrante      crime      date      hour  year
##   <chr>          <chr>    <date>    <time> <int>
## 1 C-2.1.13      ROBO A NEGOCIO C.V. 2013-01-01 22:20:00 2013
## 2 C-2.1.18      ROBO A NEGOCIO C.V. 2013-01-01 22:56:00 2013
## 3 C-2.1.4       ROBO A NEGOCIO C.V. 2013-01-01 11:05:00 2013
## 4 C-2.1.8       ROBO A TRANSEUNTE C.V. 2013-01-01 04:10:00 2013
## 5 C-2.4.15 ROBO DE VEHICULO AUTOMOTOR S.V. 2013-01-01 19:05:00 2013
## 6 C-2.4.8       LESIONES POR ARMA DE FUEGO 2013-01-01 17:31:00 2013
## 7 C-2.5.3       ROBO A TRANSEUNTE C.V. 2013-01-01 11:33:00 2013
## 8 C-2.5.4       ROBO A TRANSEUNTE C.V. 2013-01-01 21:57:00 2013
## 9 N-1.1.18      VIOLACION 2013-01-01 10:07:00 2013
## 10 N-1.1.9      ROBO A TRANSEUNTE C.V. 2013-01-01 11:18:00 2013
## # ... with 3 more variables: month <int>, lat <dbl>, long <dbl>
```

```
cuad_hdcrime <- read_csv(file.path(temp2, "/clean-data/cuadrantes-hoyodecrimen.csv"), col_types = cols())
cuad_hdcrime <- arrange(cuad_hdcrime, date, cuadrante, crime)
head(cuad_hdcrime)
```

```
## # A tibble: 6 x 16
##   municipio      cuadrante      crime      date
##   <chr>          <chr>          <chr>    <date>
## 1 <NA> (NO ESPECIFICADO) HOMICIDIO DOLOSO 2013-01-01
## 2 <NA> (NO ESPECIFICADO) LESIONES POR ARMA DE FUEGO 2013-01-01
## 3 <NA> (NO ESPECIFICADO) ROBO A BORDO DE METRO C.V. 2013-01-01
## 4 <NA> (NO ESPECIFICADO) ROBO A BORDO DE METRO S.V. 2013-01-01
## 5 <NA> (NO ESPECIFICADO) ROBO A BORDO DE MICROBUS C.V. 2013-01-01
## 6 <NA> (NO ESPECIFICADO) ROBO A BORDO DE MICROBUS S.V. 2013-01-01
## # ... with 12 more variables: count <int>, year <int>, zona <chr>,
## #   cve_zona <chr>, no_region <int>, sector1 <chr>, sector2 <chr>,
## #   sector <chr>, cve_sector <int>, no_cuadrante <int>, population <int>,
## #   cve_mun <int>
```

Paquetes:

.- lubridate: librería para trabajar con Fechas y Tiempo.

.- profvis: La función system.time supone que sabes donde buscar, es decir, que expresiones debes evaluar, una función que puede ser más útil cuando uno desconoce cuál es la función que alenta un programa es profvis() (paquete `profvis`).

.-