

Manejo de Archivos

Herramientas
computacionales: el arte
de la analítica

Semana Tec



Base de datos a utilizar (Dataset)

- Historical data for bike sharing in London 'Powered by TfL Open Data'
 - <https://www.kaggle.com/datasets/hmavrodiev/london-bike-sharing-dataset/data>

Metadata:

"timestamp" - *timestamp field for grouping the data*

"cnt" - *the count of a new bike shares*

"t1" - *real temperature in C*

"t2" - *temperature in C "feels like"*

"hum" - *humidity in percentage*

"wind_speed" - *wind speed in km/h*

"weather_code" - *category of the weather*

"is_holiday" - *boolean field - 1 holiday / 0 non holiday*

"is_weekend" - *boolean field - 1 if the day is weekend*

"season" - *category field meteorological seasons: 0-spring ; 1-summer; 2-fall; 3-winter.*

"weathe_code" category description:

*1 = Clear ; mostly clear but have some values with haze/fog/patches of fog/ fog in vicinity 2 = scattered clouds / few clouds 3 = Broken clouds
4 = Cloudy 7 = Rain/ light Rain shower/ Light rain 10 = rain with thunderstorm 26 = snowfall 94 = Freezing Fog*

Lectura de archivos CSV

- CSV (Comma Separated Values)
- Se le debe de indicar en dónde está el archivo (en qué carpeta)
 - Si no se le dice, asume que está en la misma carpeta en la que está trabajando RStudio

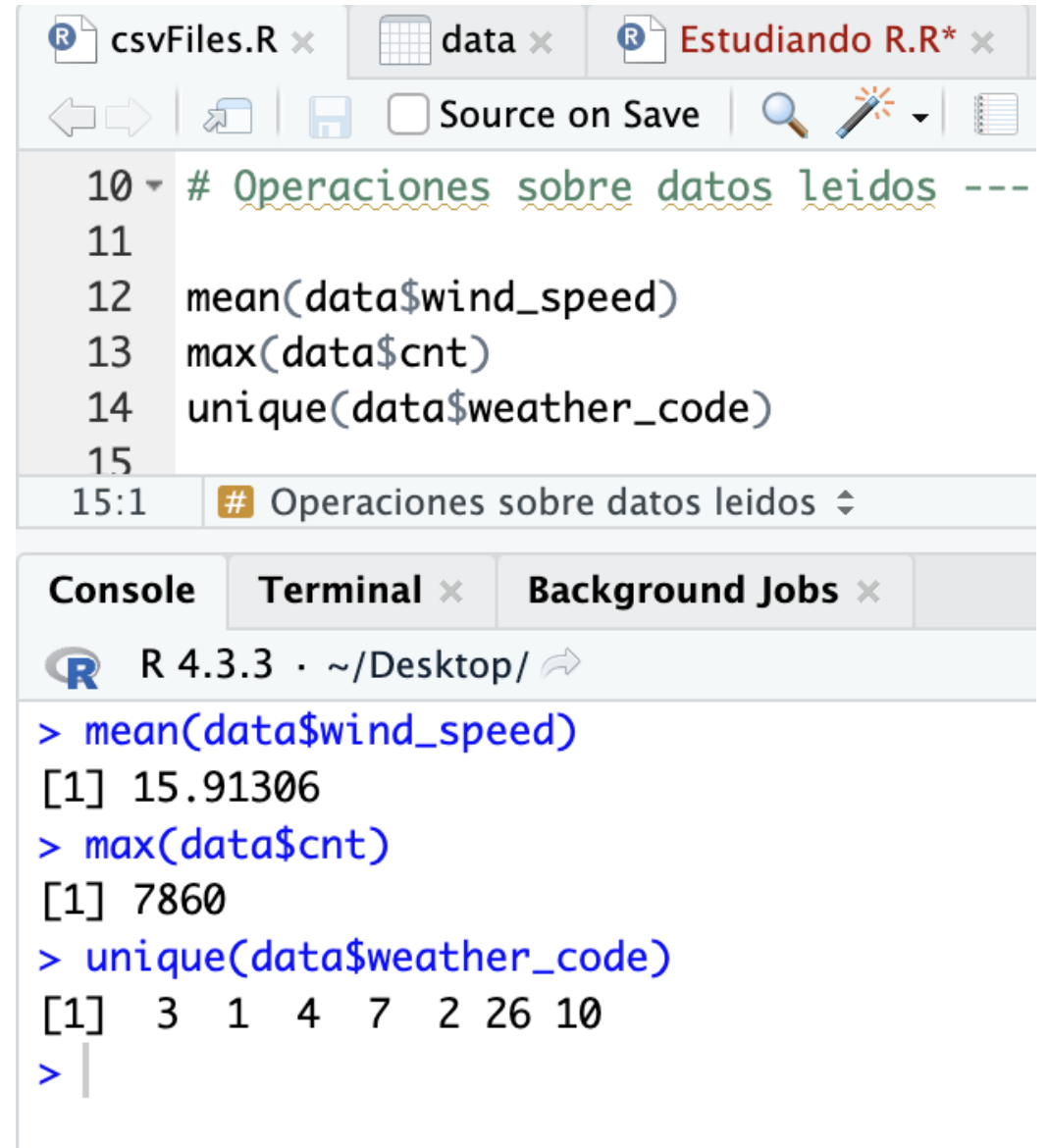
```
1  
2 # Abrir archivo CSV -----  
3  
4 getwd()  
5 setwd("/Users/dr.jorge/Desktop")  
6 getwd()  
7 data = read.csv("./london_merged.csv")  
8 View(data)  
9
```

	timestamp	cnt	t1	t2	hum	wind_speed	weather_code	is_holiday	is_weekend	season
1	2015-01-04 00:00:00	182	3.0	2.0	93.0	6.0	3	0	1	3
2	2015-01-04 01:00:00	138	3.0	2.5	93.0	5.0	1	0	1	3
3	2015-01-04 02:00:00	134	2.5	2.5	96.5	0.0	1	0	1	3
4	2015-01-04 03:00:00	72	2.0	2.0	100.0	0.0	1	0	1	3
5	2015-01-04 04:00:00	47	2.0	0.0	93.0	6.5	1	0	1	3
6	2015-01-04 05:00:00	46	2.0	2.0	93.0	4.0	1	0	1	3
7	2015-01-04 06:00:00	51	1.0	-1.0	100.0	7.0	4	0	1	3
8	2015-01-04 07:00:00	75	1.0	-1.0	100.0	7.0	4	0	1	3
9	2015-01-04 08:00:00	131	1.5	-1.0	96.5	8.0	4	0	1	3
10	2015-01-04 09:00:00	301	2.0	-0.5	100.0	9.0	3	0	1	3
11	2015-01-04 10:00:00	528	3.0	-0.5	93.0	12.0	3	0	1	3
12	2015-01-04 11:00:00	727	2.0	-1.5	100.0	12.0	3	0	1	3
13	2015-01-04 12:00:00	862	2.0	-1.5	96.5	13.0	4	0	1	3
14	2015-01-04 13:00:00	916	3.0	-0.5	87.0	15.0	3	0	1	3
15	2015-01-04 14:00:00	1039	2.5	0.0	90.0	8.0	3	0	1	3
16	2015-01-04 15:00:00	869	2.0	-1.5	93.0	11.0	3	0	1	3
17	2015-01-04 16:00:00	737	3.0	0.0	93.0	12.0	3	0	1	3

Showing 1 to 17 of 17,414 entries, 10 total columns

Operaciones sobre los datos

- Se pueden realizar las mismas operaciones que con los vectores
- Tan solo se debe de indicar el nombre de la columna (característica) que se desea analizar



The screenshot shows the RStudio interface with three tabs: 'csvFiles.R', 'data', and 'Estudiando R.R*'. The 'data' tab is active, displaying the following R code:

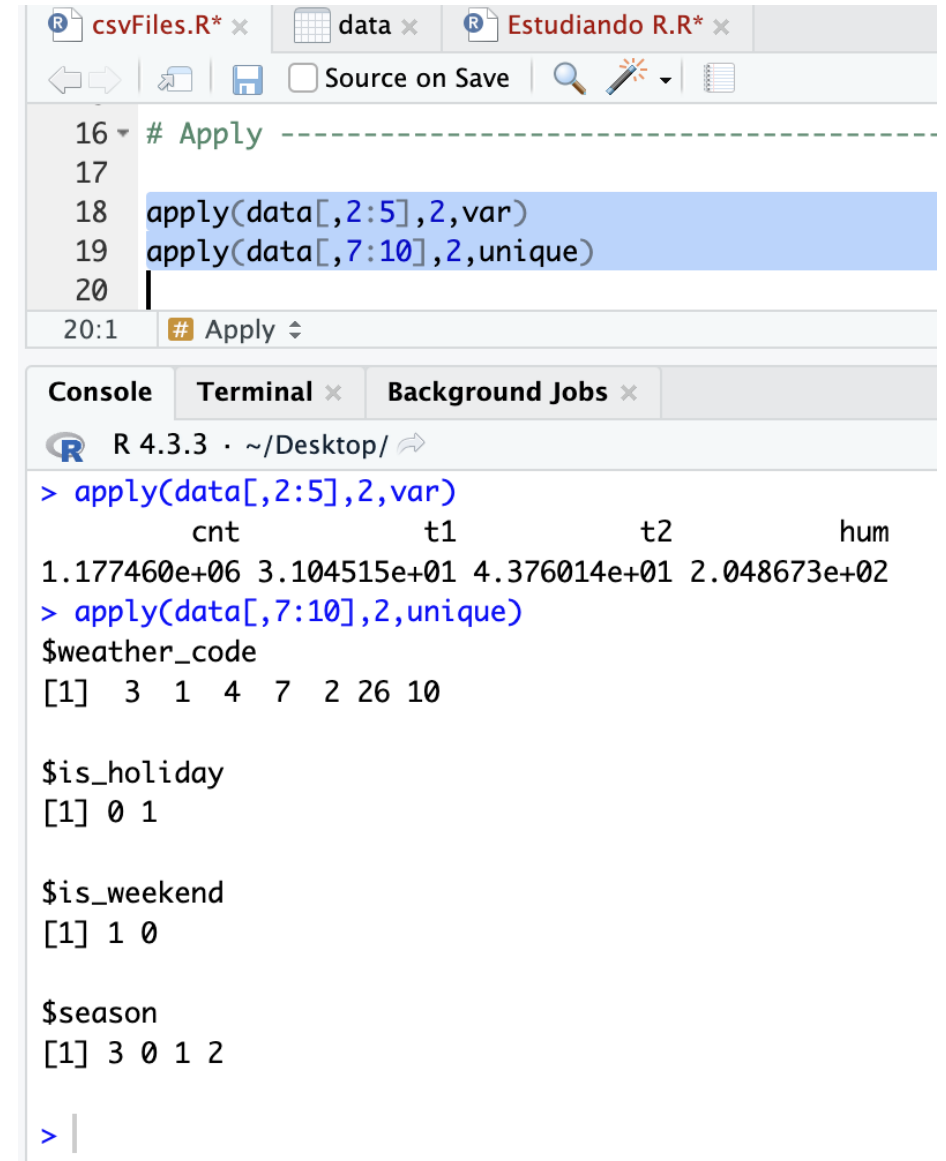
```
10 # Operaciones sobre datos leídos ---  
11  
12 mean(data$wind_speed)  
13 max(data$cnt)  
14 unique(data$weather_code)  
15
```

Below the code editor, the 'Console' tab is active, showing the output of the executed code:

```
R 4.3.3 · ~/Desktop/  
> mean(data$wind_speed)  
[1] 15.91306  
> max(data$cnt)  
[1] 7860  
> unique(data$weather_code)  
[1] 3 1 4 7 2 26 10  
> |
```

Misma función sobre varias columnas

- `apply(x, margen, funcion)`
 - `x` corresponde a los datos
 - `margen`
 - 1 sobre los renglones
 - 2 sobre las columnas
 - `c(1,2)` sobre renglones y columnas
 - `funcion` indica cuál función aplicar a los datos



The screenshot shows an RStudio interface with three tabs: 'csvFiles.R*', 'data', and 'Estudiando R.R*'. The 'Estudiando R.R*' tab is active, displaying the following R code in the editor:

```
16 # Apply -----  
17  
18 apply(data[,2:5],2,var)  
19 apply(data[,7:10],2,unique)  
20
```

The console output shows the results of these two `apply` functions:

```
> apply(data[,2:5],2,var)  
      cnt      t1      t2      hum  
1.177460e+06 3.104515e+01 4.376014e+01 2.048673e+02  
> apply(data[,7:10],2,unique)  
$weather_code  
[1] 3 1 4 7 2 26 10  
  
$is_holiday  
[1] 0 1  
  
$is_weekend  
[1] 1 0  
  
$season  
[1] 3 0 1 2  
  
>
```

Preparación de los datos

VALIDAZ DE LOS DATOS



Base de datos a utilizar (Dataset)

Rain in Australia

Predict next-day rain in Australia



[Data Card](#) [Code \(644\)](#) [Discussion \(21\)](#) [Suggestions \(0\)](#)

About Dataset

Context

Predict **next-day rain** by training classification models on the target variable **RainTomorrow**.

Content

This dataset contains about 10 years of daily weather observations from many locations across Australia.

RainTomorrow is the target variable to predict. It means -- did it rain the next day, Yes or No? This column is Yes if the rain for that day was 1mm or more.

Source & Acknowledgements

Observations were drawn from numerous weather stations. The daily observations are available from <http://www.bom.gov.au/climate/data>. An example of latest weather observations in Canberra: <http://www.bom.gov.au/climate/dwo/IDCJDW2801.latest.shtml>

Definitions adapted from <http://www.bom.gov.au/climate/dwo/IDCJDW0000.shtml> Data source: <http://www.bom.gov.au/climate/dwo/> and <http://www.bom.gov.au/climate/data>.

Copyright Commonwealth of Australia 2010, Bureau of Meteorology.

Usability ⓘ

10.00

License

Other (specified in description)

Expected update frequency

Never

Tags

Earth and Nature

Classification

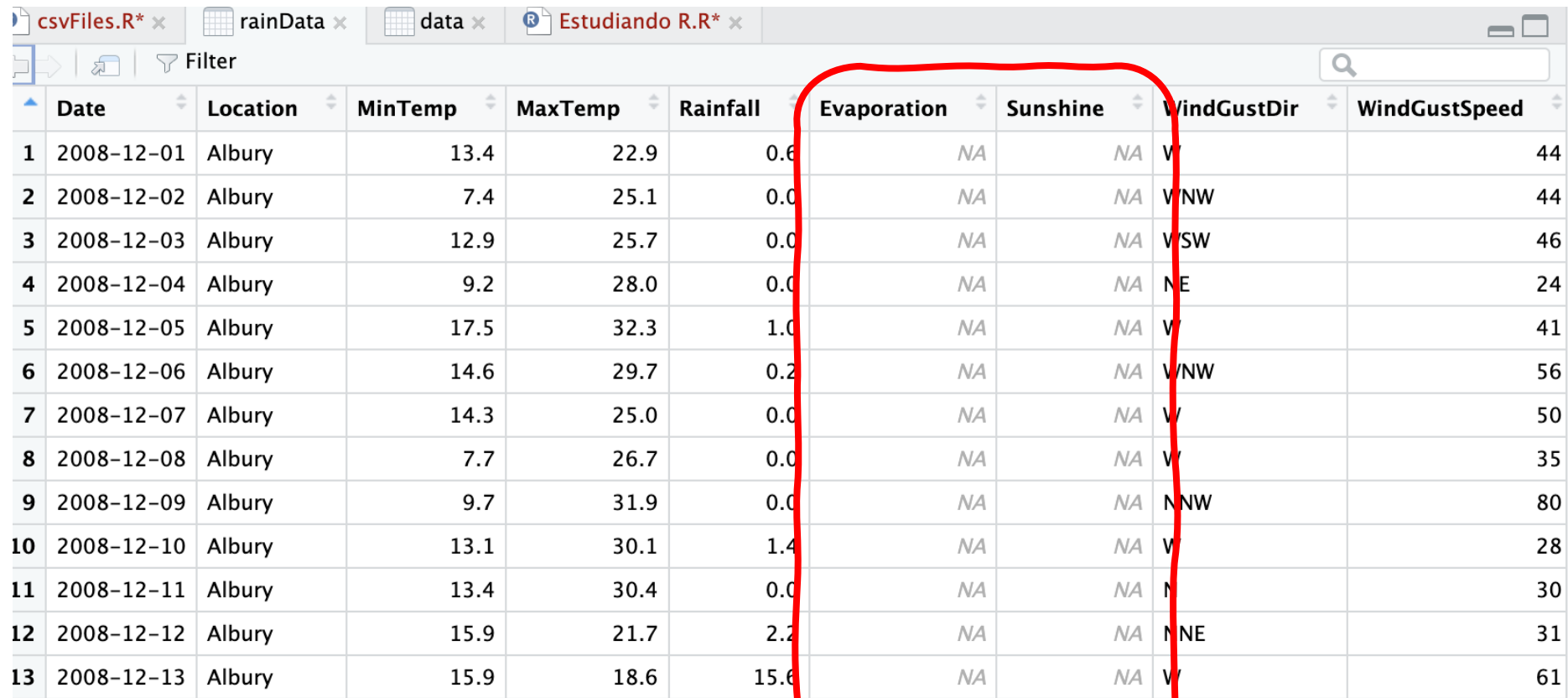
Binary Classification

Weather and Climate

- Rain in Australia: Predict next-day rain in Australia
 - <https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package>

Valores nulos

- Representados como NA
- No aportan al análisis de los datos

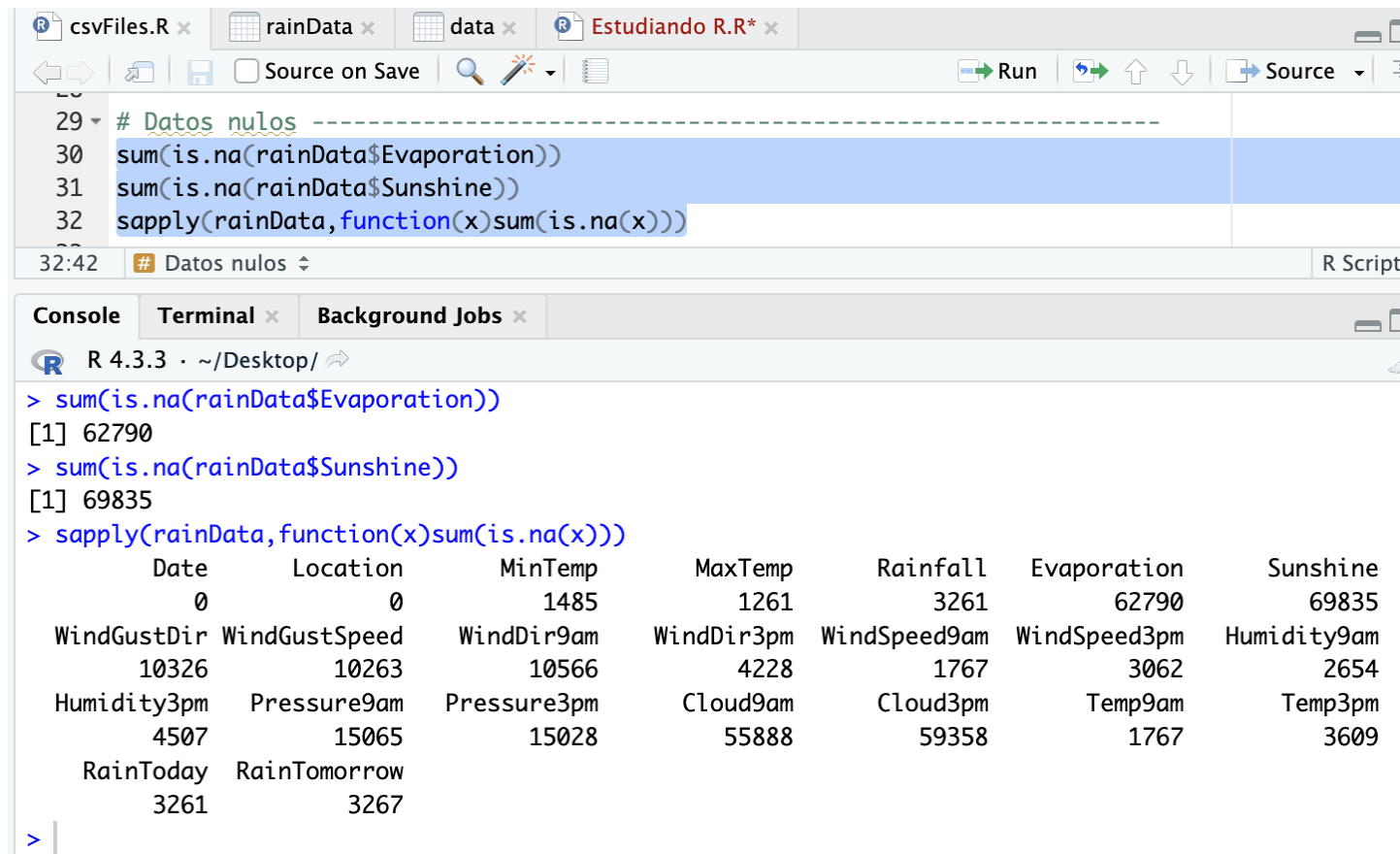


The screenshot shows an RStudio window with a data table. The table has 10 columns: Date, Location, MinTemp, MaxTemp, Rainfall, Evaporation, Sunshine, WindGustDir, and WindGustSpeed. The 'Evaporation' and 'Sunshine' columns are circled in red, indicating they contain NA values. The table contains 13 rows of data for the month of December 2008.

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed
1	2008-12-01	Albury	13.4	22.9	0.6	NA	NA	W	44
2	2008-12-02	Albury	7.4	25.1	0.0	NA	NA	WNW	44
3	2008-12-03	Albury	12.9	25.7	0.0	NA	NA	WSW	46
4	2008-12-04	Albury	9.2	28.0	0.0	NA	NA	NE	24
5	2008-12-05	Albury	17.5	32.3	1.0	NA	NA	W	41
6	2008-12-06	Albury	14.6	29.7	0.2	NA	NA	WNW	56
7	2008-12-07	Albury	14.3	25.0	0.0	NA	NA	W	50
8	2008-12-08	Albury	7.7	26.7	0.0	NA	NA	W	35
9	2008-12-09	Albury	9.7	31.9	0.0	NA	NA	NNW	80
10	2008-12-10	Albury	13.1	30.1	1.4	NA	NA	W	28
11	2008-12-11	Albury	13.4	30.4	0.0	NA	NA	N	30
12	2008-12-12	Albury	15.9	21.7	2.2	NA	NA	NNE	31
13	2008-12-13	Albury	15.9	18.6	15.6	NA	NA	W	61

Contar datos nulos

- `sapply(X,funcion)`
 - regresa un vector con los resultados de aplicar la *función* a todos los datos *X*



The screenshot shows the R Studio interface with the following components:

- Script Editor:** Contains R code for counting missing values in the `rainData` dataset. Lines 29-32 are highlighted in blue.
- Console:** Shows the execution of the code and the resulting output, including a data frame with 10 columns and 1 row.

```
# Datos nulos -----
sum(is.na(rainData$Evaporation))
sum(is.na(rainData$Sunshine))
sapply(rainData,function(x)sum(is.na(x)))
```

```
> sum(is.na(rainData$Evaporation))
[1] 62790
> sum(is.na(rainData$Sunshine))
[1] 69835
> sapply(rainData,function(x)sum(is.na(x)))
```

Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine
0	0	1485	1261	3261	62790	69835
WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed9am	WindSpeed3pm	Humidity9am
10326	10263	10566	4228	1767	3062	2654
Humidity3pm	Pressure9am	Pressure3pm	Cloud9am	Cloud3pm	Temp9am	Temp3pm
4507	15065	15028	55888	59358	1767	3609
RainToday	RainTomorrow					
3261	3267					

Eliminar NA

- Se eliminan los NA de una columna
- Las demás columnas pueden seguir teniendo NA
- Necesario eliminar los NA de cada columna

```
csvFiles.R x rainData x data x Estudiando R.R* x
Source on Save Run
34 # Eliminar NA -----
35
36 newRData <- rainData[!is.na(rainData$Evaporation), ]
37 sum(is.na(newRData$Evaporation))
38 sapply(newRData, function(x) sum(is.na(x)))
39
40 newRData <- newRData[!is.na(newRData$Sunshine), ]
41 sapply(newRData, function(x) sum(is.na(x)))
42
42:1 ## Eliminar NA <->
```

```
R 4.3.3 · ~/Desktop/
> newRData <- rainData[!is.na(rainData$Evaporation), ]
> sum(is.na(newRData$Evaporation))
[1] 0
> sapply(newRData, function(x) sum(is.na(x)))
```

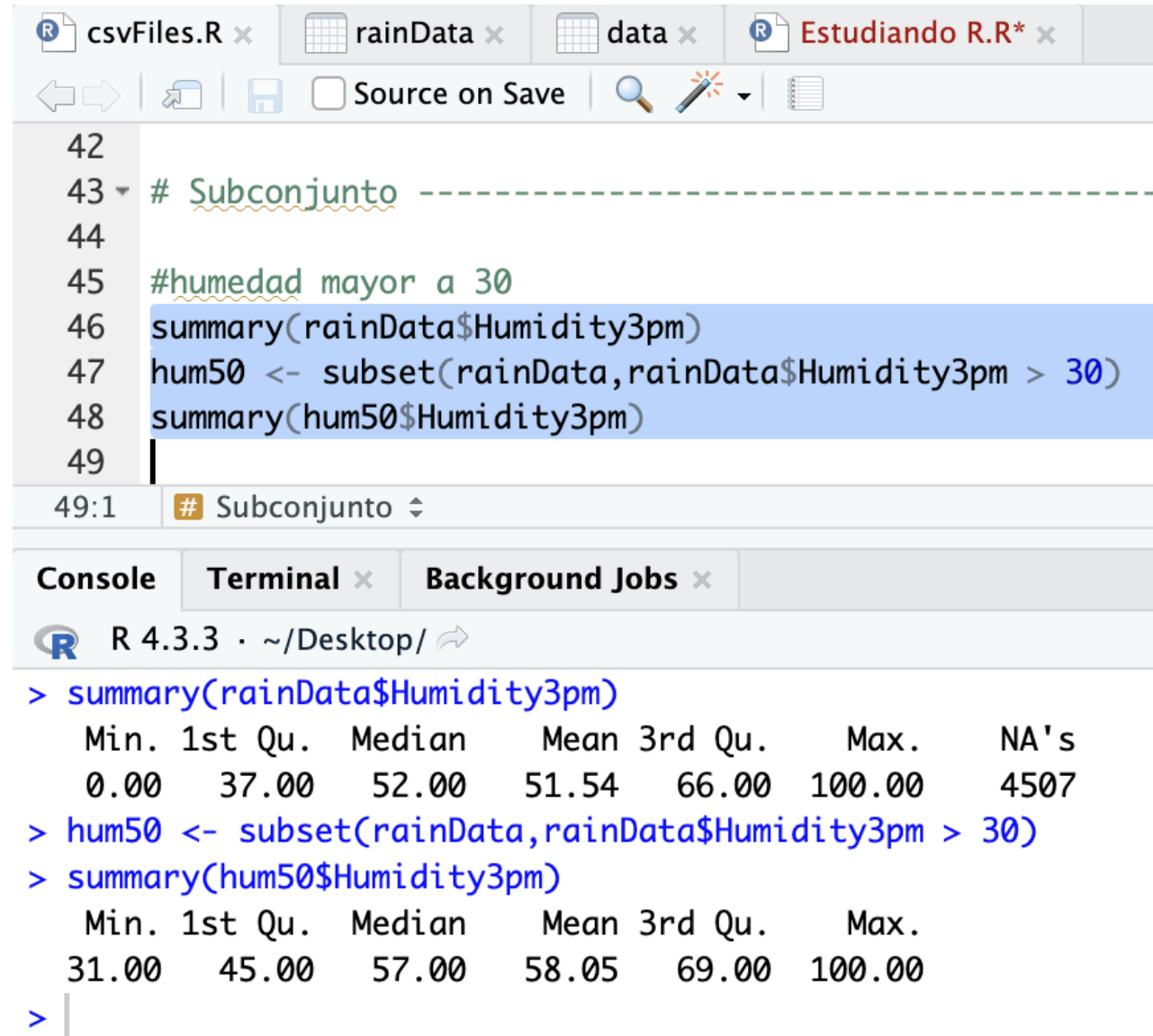
Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation
0	0	658	645	959	0
Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed9am
11288	5052	5025	3632	1284	500
WindSpeed3pm	Humidity9am	Humidity3pm	Pressure9am	Pressure3pm	Cloud9am
919	1079	2180	930	933	10389
Cloud3pm	Temp9am	Temp3pm	RainToday	RainTomorrow	
13054	697	1753	959	1320	

```
> newRData <- newRData[!is.na(newRData$Sunshine), ]
> sapply(newRData, function(x) sum(is.na(x)))
```

Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation
0	0	617	589	911	0
Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed9am
0	4351	4328	2617	718	215
WindSpeed3pm	Humidity9am	Humidity3pm	Pressure9am	Pressure3pm	Cloud9am
432	913	1089	653	660	7052
Cloud3pm	Temp9am	Temp3pm	RainToday	RainTomorrow	
8581	652	847	911	964	

```
>
```

Crear subconjunto de un dataset



The screenshot shows the RStudio IDE with the following elements:

- Top Panel:** Tab titles include 'csvFiles.R', 'rainData', 'data', and 'Estudiando R.R*'. The toolbar shows navigation and execution icons.
- Source Editor:** Contains R code for creating a subset based on humidity. Lines 46-48 are highlighted in blue.
- Console:** Shows the output of the executed commands, including summary statistics for the original and subsetted data.

```
42  
43 # Subconjunto -----  
44  
45 #humedad mayor a 30  
46 summary(rainData$Humidity3pm)  
47 hum50 <- subset(rainData, rainData$Humidity3pm > 30)  
48 summary(hum50$Humidity3pm)  
49 |
```

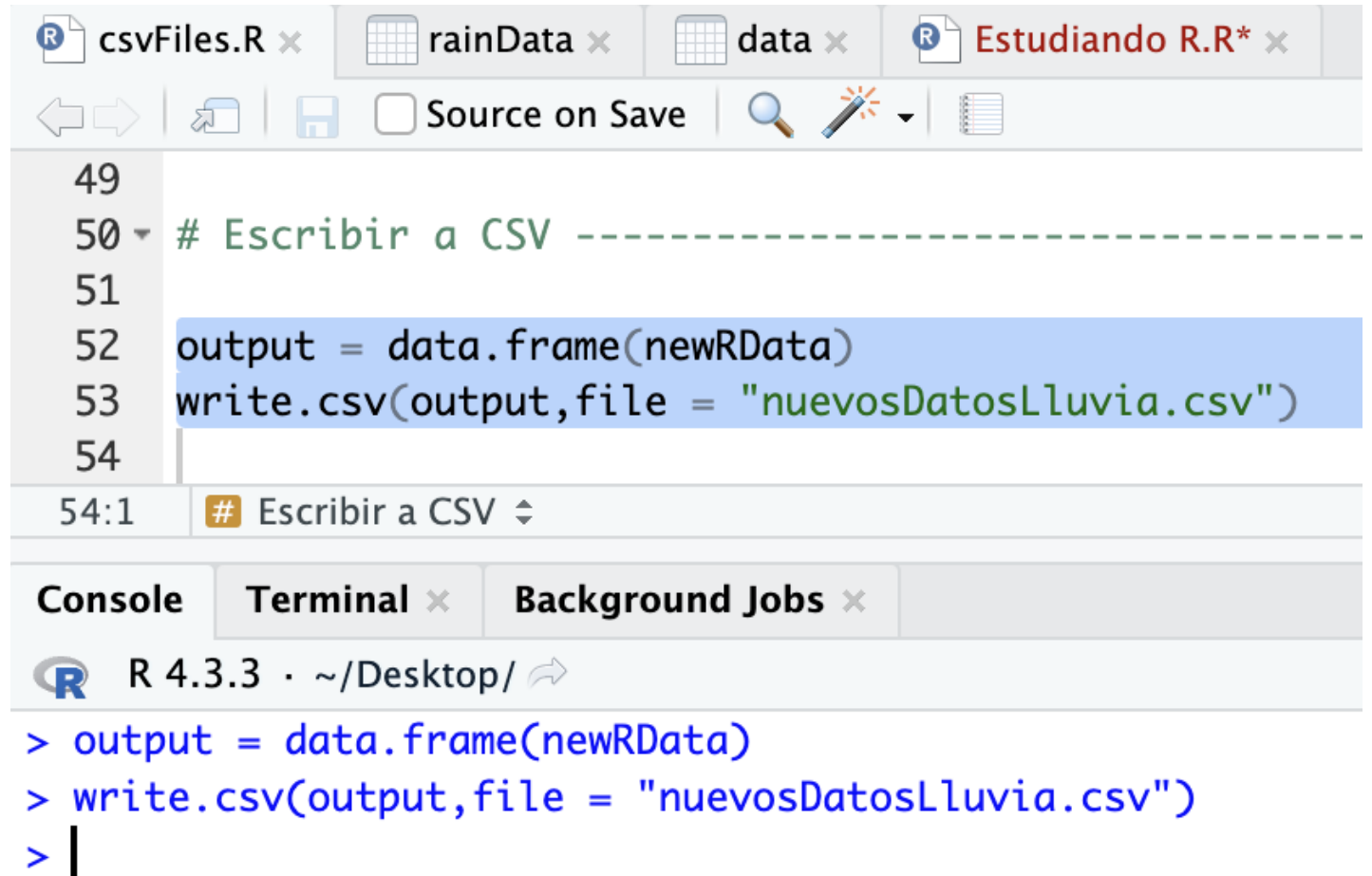
49:1 # Subconjunto

Console Terminal Background Jobs

R 4.3.3 · ~/Desktop/

```
> summary(rainData$Humidity3pm)  
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's  
  0.00   37.00   52.00   51.54   66.00   100.00  4507  
> hum50 <- subset(rainData, rainData$Humidity3pm > 30)  
> summary(hum50$Humidity3pm)  
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
 31.00   45.00   57.00   58.05   69.00   100.00  
> |
```

Escribir
dataset a un
CSV



The screenshot shows the RStudio IDE interface. The top toolbar includes icons for file operations and a 'Source on Save' checkbox. The script editor displays R code with line numbers 49 to 54. Lines 52 and 53 are highlighted in blue. The console at the bottom shows the execution of the code, with the prompt character '>' at the start of each line.

```
49  
50 # Escribir a CSV -----  
51  
52 output = data.frame(newRData)  
53 write.csv(output, file = "nuevosDatosLluvia.csv")  
54
```

54:1 # Escribir a CSV

Console Terminal Background Jobs

R 4.3.3 · ~/Desktop/

```
> output = data.frame(newRData)  
> write.csv(output, file = "nuevosDatosLluvia.csv")  
> |
```

Ejercicio integrador

- Utiliza el dataset “Climate Change: Earth Surface Temperature Data”
 - <https://www.kaggle.com/datasets/berkeleyearth/climate-change-earth-surface-temperature-data>
- Realiza un análisis en R de los datos descargados (utiliza estadísticos y gráficas)
- Entrega
 - Código Fuente
 - Documento PDF con lo siguiente:
 - Discusión de los resultados arrojados en el análisis en R
 - Conclusión
 - Selecciona 3 países y compara cómo han cambiado las temperaturas en esos 3 países.

Climate Change: Earth Surface Temperature Data

2241

New Notebook

Download

Data Card

Code (632)

Discussion (9)

Suggestions (0)

GlobalLandTemperaturesByCity.csv (532.83 MB)

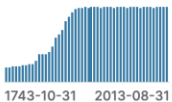
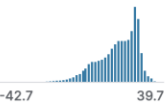


Download

Detail

Compact

Column

7 of 7 columns

dt	# AverageTemperature	# AverageTemperature...	City	Country	Latitude
			3448 unique values		36.17N 34.56N Other (78:...
1749-11-01			Århus	Denmark	57.05N
1749-12-01			Århus	Denmark	57.05N
1750-01-01	1.699	1.013	Århus	Denmark	57.05N
1750-02-01	3.9610000000000003	2.3609999999999998	Århus	Denmark	57.05N
1750-03-01	5.182	3.48	Århus	Denmark	57.05N
1750-04-01	7.197	0.732	Århus	Denmark	57.05N