Introducción a R

Aplicaciones a la enseñanza de la Estadística

IV - Encuentro Colombiano de Educación Estocástica

Daniel Enrique González Gómez

2021-05-31

Base de datos

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.
Wikipedia

Una base de datos en estadística es un conjunto de información relacionada con una población organizada en filas y columnas. Las columnas corresponden a las variables y las filas están relacionadas con los individuos u objetos de estudio.

Existen repositorio de bases de datos para uso general

- dataset en RStudio
- Portal Bases de datos abiertos Colombia
- Datos Banco mundial
- Portal de Datos Abiertos de Esri España

[*] Open Data Barometer : https://opendatabarometer.org/4thedition/report/?lang=es

Leer datos

- Data set R : bases de datos al interior de los paquetes de R
- Utilizando menu de RStudio, datos en DD en formato:
  - Excel
  - csv
  - SPSS
  - SAS
  - Stata
- Utilizando funciones en linea de consula
- De forma automatica con paquete RSocrata

R Dataset

```
paquetes=library(help = "datasets")
head(paquetes$info[[2]])
```

```
## [1] "AirPassengers          Monthly Airline Passenger Numbers 1949-1960"
## [2] "BJsales                Sales Data with Leading Indicator"
## [3] "BOD                    Biochemical Oxygen Demand"
## [4] "CO2                    Carbon Dioxide Uptake in Grass Plants"
## [5] "ChickWeight            Weight versus age of chicks on different diets"
## [6] "DNase                  Elisa assay of DNase"
```

```
tail(paquetes$info[[2]])
```

```
## [1] "                       Trees"
## [2] "uspop                  Populations Recorded by the US Census"
## [3] "volcano                Topographic Information on Auckland's Maunga"
## [4] "                       Whau Volcano"
## [5] "warpbreaks             The Number of Breaks in Yarn during Weaving"
## [6] "women                  Average Heights and Weights for American Women"
```

https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/00Index.html

.footnote

R Dataset

```
data(iris) # dataset de R
```

```
head(iris) # visualiza las primeras 6 filas de la base
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Datos de iris (de Fisher o Anderson)

- longitud y ancho del sépalo
- largo y ancho de pétalos
- especies: setosa, versicolor y virginica.

Base de datos estadísticos : arreglo de filas y columnas (matriz) donde por lo general las columnas representan las variables y las filas los registros de los objetos de estudio

Base de datos estudianes Probabilidad y Estadistica

```
bd0052 = read_excel("data/bd0052.xlsx",col_types = c("numeric", "numeric", "text",
         "numeric", "text", "numeric"))
DT::datatable(head(bd0052,81),fillContainer = FALSE, options = list(pageLength = 6))
```

Show 6 entries Search:

	id	idgrup	grupo	promacum	carrera	matriculada
1	4	4	A	3.75	Biología	1
2	6	6	A	3.47	Biología	1
3	9	9	A	4.05	Biología	1
4	10	10	A	3.9	Biología	1
5	11	11	A	3.55	Biología	1
6	12	12	A	4.63	Biología	1

Showing 1 to 6 of 81 entries Previous 1 2 3 4 5 … 14 Next

```
library(readr)
data=read_csv("data/spi_global_rankings_intl.csv")
DT::datatable(head(data,218),fillContainer = FALSE, options = list(pageLength = 6))
```

Show 6 entries Search:

	rank	name	confed	off	def	spi
1	1	Spain	UEFA	3.54	0.39	93.99
2	2	Brazil	CONMEBOL	3.04	0.32	92.22
3	3	Belgium	UEFA	2.96	0.58	87.71
4	4	France	UEFA	2.9	0.55	87.64
5	5	England	UEFA	2.7	0.45	87.44
6	6	Argentina	CONMEBOL	2.54	0.42	86.59

Showing 1 to 6 of 218 entries Previous 1 2 3 4 5 … 37 Next

[*]https://data.fivethirtyeight.com/#soccer-spi

Importar datos de manera automatica

La API de datos abiertos de Socrata le permite acceder mediante programación a una gran cantidad de recursos de datos abiertos de gobiernos, organizaciones sin fines de lucro y ONG de todo el mundo. Haga clic en el enlace de abajo y pruebe un ejemplo en vivo ahora mismo.

https://dev.socrata.com/

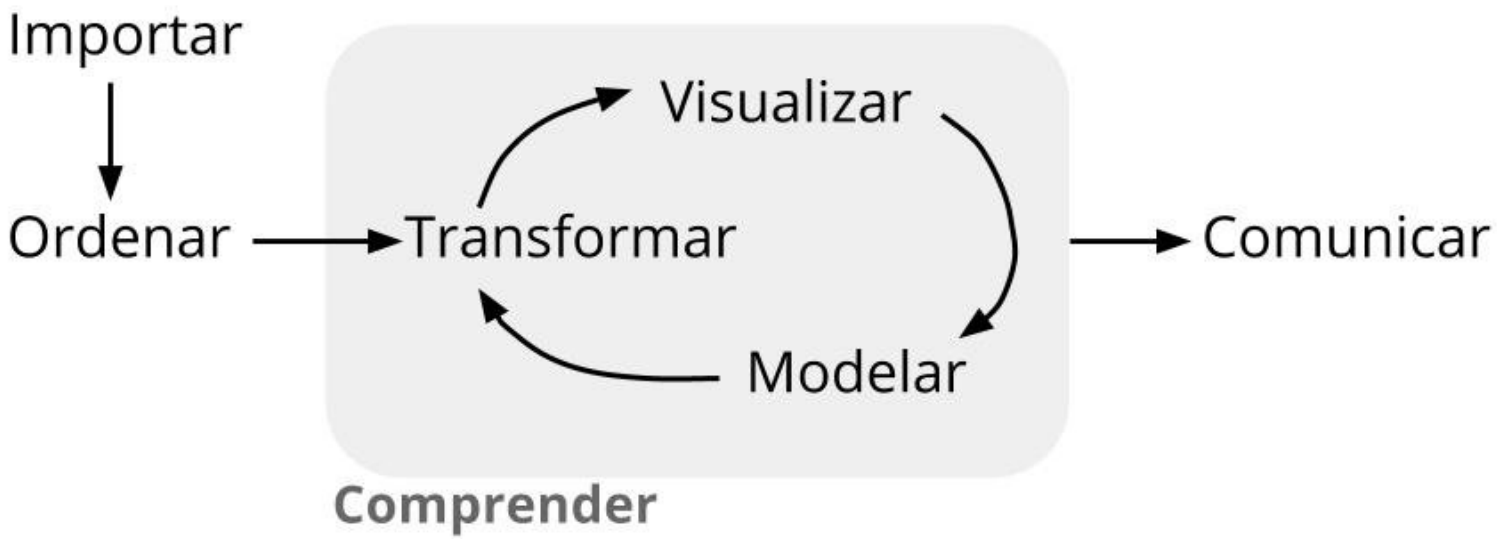
Cargar la base de datos de COVID-19 Colombia

```
# install.packages("RSocrata")  # instal paquete RSocrata
 library(RSocrata)
 token ="ew2rEMuESuzWPqMkyPfOSGJgE"
 Colombia= read.socrata("https://www.datos.gov.co/resource/gt2j-8ykr.json", app_token = token)
 saveRDS(Colombia,"Colombia.RDS")
```

https://www.datos.gov.co/ https://dev.socrata.com/foundry/www.datos.gov.co/gt2j-8ykr

[*] Se requiere solicitar token en la página de los datos - RDS formato de R para almacenar cualquier objeto - se lee con readRDS

Etapas del proceso de datos



[*] Imagen tomada de : https://bitsandbricks.github.io/ciencia_de_datos_gente_sociable/

Ordenar los datos

Es importante después de haber importado la base de datos, hacer una revisión de cada una de las variables con el fin de poder detectar:

- Datos faltantes (NA)
- Datos anómalos o raros
- Etiquetas mal colocadas (minúsculas, MAYÚSCULAS, Titulo...)

```
> table(Colombia$ubicacion)

       casa        Casa        CASA    Fallecido     Hospital Hospital UCI         N/A
      11012     3121567           7       85207       14568         4284       12788
```

```
     f       F       m       M
     3 1688878       2 1560550
```

Arreglo de la base de datos

```
library(stringr)
Colombia$sexo=str_to_lower(Colombia$sexo)
Colombia$estado[Colombia$estado=="N/A"]="NA"
Colombia$estado=str_to_lower(Colombia$estado)

Colombia$recuperado[Colombia$recuperado=="N/A"]="NA"
Colombia$recuperado=str_to_lower(Colombia$recuperado)

Colombia$fuente_tipo_contagio[Colombia$fuente_tipo_contagio=="N/A"]="NA"
Colombia$fuente_tipo_contagio=str_to_lower(Colombia$fuente_tipo_contagio)

Colombia$ubicacion[Colombia$ubicacion=="N/A"]="NA"
Colombia$ubicacion=str_to_lower(Colombia$ubicacion)
```


km/h
0
20
40
60
80
100
120
140
160
180
200
220
240
260
1/min*1000
0
1
2
3
4
Indicadores
+20.5 c
1049 km
km

Tablas de frecuencia

```
table(bd0052$carrera)
```

```
##
##                  Biología         Ingeniería Civil   Ingeniería de Sistemas   Ingeniería Electrónica
##                        23                      41                        4                        5
##      Ingeniería Mecánica Negocios Internacionales
##                        4                       4
```

```
data.frame(table(bd0052$carrera))
```

```
##                      Var1 Freq
## 1                 Biología   23
## 2         Ingeniería Civil   41
## 3   Ingeniería de Sistemas    4
## 4   Ingeniería Electrónica    5
## 5      Ingeniería Mecánica    4
## 6 Negocios Internacionales    4
```

```
t1=summarytools::freq(bd0052$carrera, cumul = FALSE, headings = FALSE)
t1
```

```
##
##                                Freq   % Valid   % Total
## ------------------------------ ------ --------- ---------
##                        Biología     23     28.40     28.40
##              Ingeniería Civil     41     50.62     50.62
##       Ingeniería de Sistemas      4      4.94      4.94
##       Ingeniería Electrónica      5      6.17      6.17
##          Ingeniería Mecánica      4      4.94      4.94
##     Negocios Internacionales      4      4.94      4.94
##                          <NA>      0                0.00
##                         Total     81    100.00    100.00
```

```r
library(agricolae)
h2=with(bd0052,graph.freq(promacum,plot=FALSE))
t2=table.freq(h2)
colnames(t2) = c("  LI  ", "  LS  ", "marca clase", "Frec.Abs","Frec.Rel", "Frec.Abs.Ac","Frec.
t2
```

```
##      LI     LS   marca clase Frec.Abs Frec.Rel Frec.Abs.Ac Frec.Rel.Ac
## 1   3.30   3.53         3.415       16     20.0          16        20.0
## 2   3.53   3.76         3.645       21     26.2          37        46.2
## 3   3.76   3.99         3.875       11     13.8          48        60.0
## 4   3.99   4.22         4.105       19     23.8          67        83.8
## 5   4.22   4.45         4.335        7      8.8          74        92.5
## 6   4.45   4.68         4.565        4      5.0          78        97.5
## 7   4.68   4.91         4.795        2      2.5          80       100.0
```

```
summarytools::descr(mtcars$mpg)
```

```
## Descriptive Statistics
## mtcars$mpg
## N: 32
##
##                      mpg
## ----------------- --------
##              Mean    20.09
##           Std.Dev     6.03
##               Min    10.40
##                Q1    15.35
##            Median    19.20
##                Q3    22.80
##               Max    33.90
##               MAD     5.41
##               IQR     7.38
##                CV     0.30
##          Skewness     0.61
##       SE.Skewness     0.41
##          Kurtosis    -0.37
##           N.Valid    32.00
##         Pct.Valid   100.00
```

```
summarytools::descr(bd0052$promacum)
```

```
## Descriptive Statistics
## bd0052$promacum
## N: 81
##
##                   promacum
## ----------------- ----------
##              Mean       3.86
##           Std.Dev       0.36
##               Min       3.34
##                Q1       3.55
##            Median       3.83
##                Q3       4.16
##               Max       4.83
##               MAD       0.43
##               IQR       0.61
##                CV       0.09
##          Skewness       0.50
##       SE.Skewness       0.27
##          Kurtosis      -0.58
##           N.Valid      80.00
##         Pct.Valid      98.77
```

Visualización

Gráficos variables cualitativas con R base

Gráfico de tortas | Diagrama de barras | Diag. de barras dos variables

```
cc=c(20, 10, 20, 20, 20, 20, 20, 20, 20, 30, 20, 20, 20, 10, 30, 20, 20, 30, 20, 30, 30, 20, 10
pie(table(cc), labels=labs, main=" Distribución por carrera")
```

Gráficas variables cuantitativas con R base

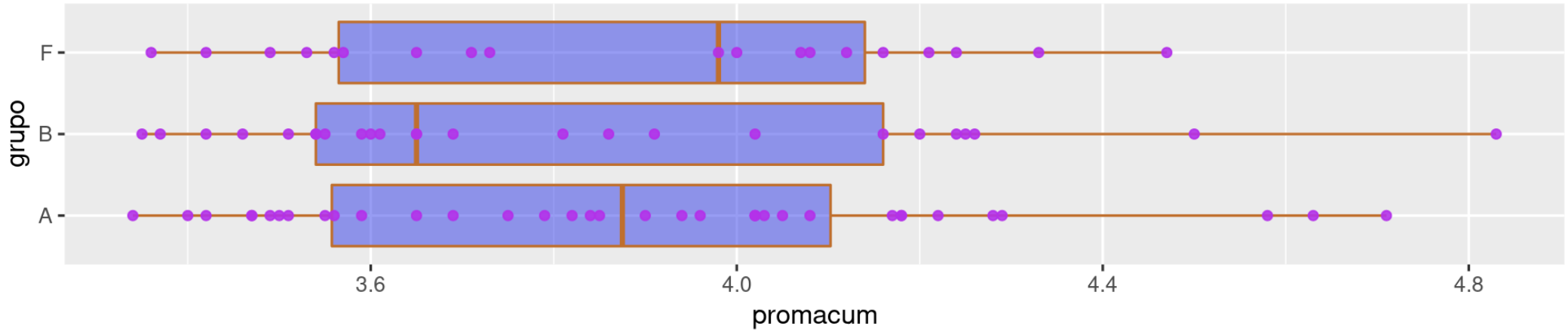
Diag.de arbol | Histograma | Diag.de Densidad | Diag.de Cajas | Diag.de cajas~factor | Diag.de Dispersión | Series de tiempo | Resumen

```
nf=c(4.1, 2.7, 3.1, 3.2, 3.0, 3.2, 2.0, 2.4, 1.6, 3.2, 3.1, 2.6, 2.0, 2.4, 2.8, 3.3, 4.0, 3.4,
stem(nf)
```

```
##
##   The decimal point is at the |
##
##   1 | 67
##   2 | 00012244444
##   2 | 555677777778
##   3 | 00000000000000111111112222222223344
##   3 | 55555556667777777888999
##   4 | 0111123
```

ggplot2


```
library(ggplot2)
ggplot(bd0052, aes(x=promacum, y=grupo)) +
  geom_boxplot(fill="#313ae8",                # color de relleno
               color="#bf6f2e",               # color de lineas
               alpha=0.5)+
  geom_point(color="#b431e8",alpha=0.9)
```



highcharter

https://jkunst.com/highcharter/

https://rstudio-pubs-static.s3.amazonaws.com/320413_6ab300527e8548b1a3cbd0d4c6200fcc.html

plotly

https://plotly.com/r/

https://plotly-r.com/

Shiny

- Genoma humano
- Paquetes de R
- Galeria

Práctica