Tema 2. Análisis de datos

Dr. Manuel Castillo-Cara
Intelligent Ubiquitous Technologies – Smart Cities (IUT-SCi)
Web: www.smartcityperu.org

Índice

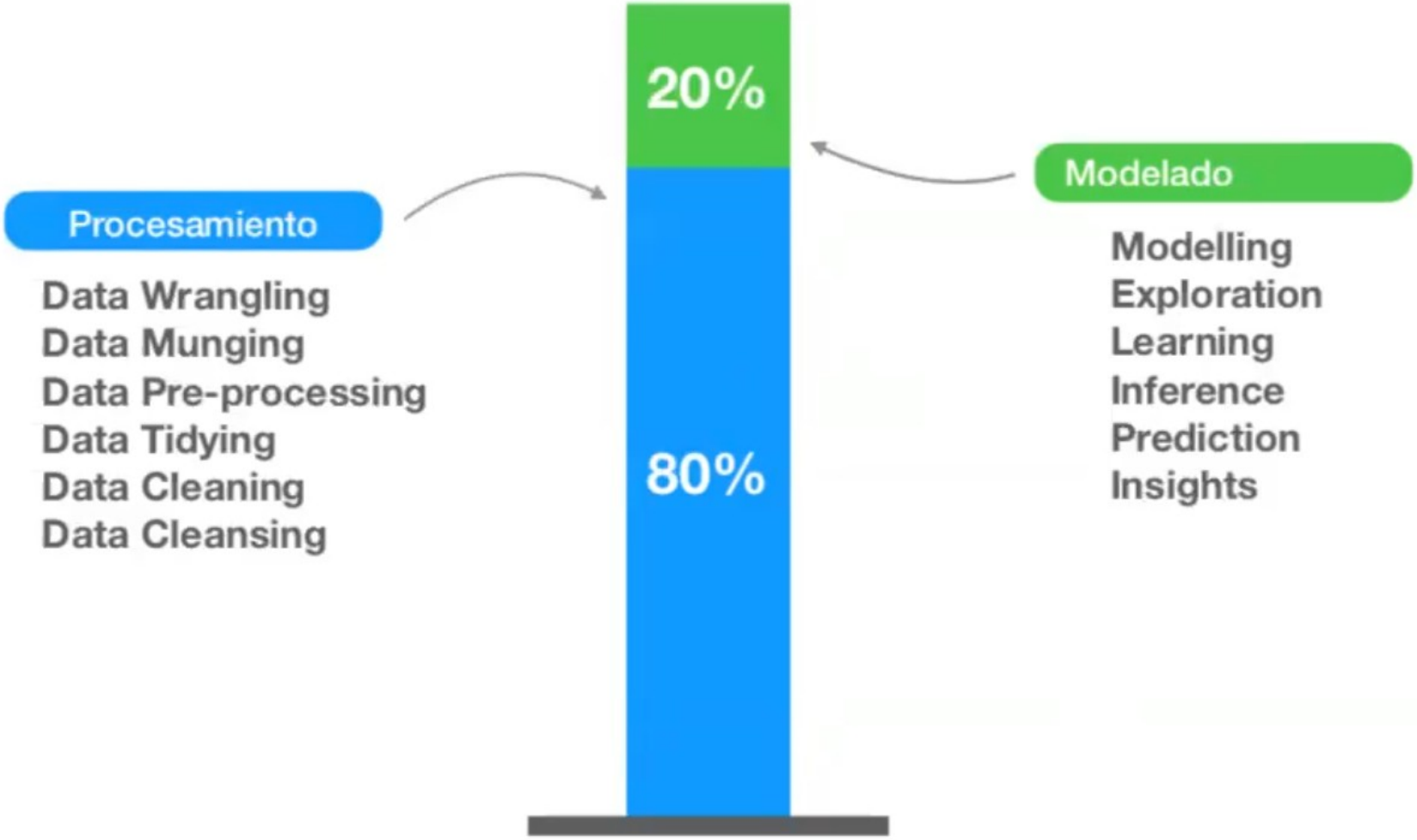
- Fuentes de datos.
- Cargar un dataset.
  - Descripción de los datasets.
- Estadística descriptiva.
- Visualización de datos.
  - Univariable.
  - Multivariable.

Smart City

LAB CTIC UNI

Fuentes de datos

1. Procesamiento de datos – Tarea que más tiempo consume



2. Tecnologías y técnicas

- Distintos perfiles profesionales:
  - **Data Scientist** → Más Desarrollador y analista
  - **Data Engineer** → Trabaja con las BBDD, ficheros en bruto… es decir trabaja con la tecnología
  - …
- Habilidades + objetivos:
  - Análisis descriptivo/exploratorio
  - Modelado/Inferencia
  - Machine Learning
  - Producto (servicio, herramienta visualización…)

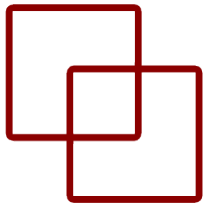
3. Orígenes de los datos

4. Formato de los datos

- Es un problema real, los datos originales no vendrán en un formato propicio para su análisis directo (estructurado)

Sepal LengthCM	Sepal WidthCm	Petal LengthCm	Petal WidthCm	Species
5.1	3.5	1.4	0.2	setosa
4.9	3	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3	1.4	0.1	setosa
4.3	3	1.1	0.1	setosa

5. Tipos de datos (I)

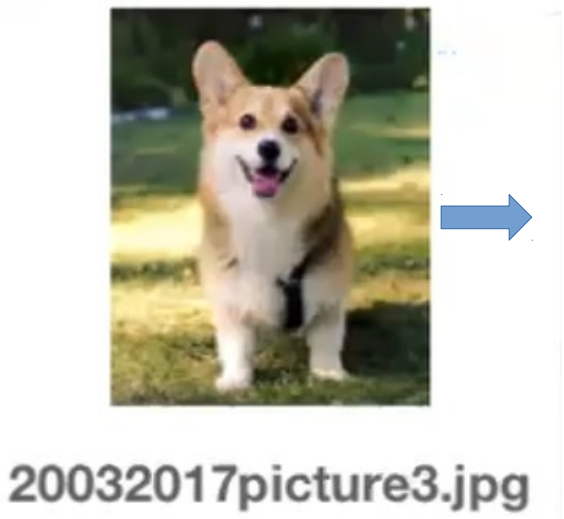


- Podemos clasificar tres formatos básicos de los datos conforme a la estructuración de la información:
  - **Datos estructurados**. Toda la información está identificada conforme a un patrón común y definido que representa un modelo. Óptima para ser analizada. Ej. Modelo relacional, base de datos SQL.
  - **Datos semi-estructurados**. La estructura de los datos está definida y es conocida, pero no responde a un modelo estricto. Óptima para ser procesada. Ej. JSON, XML.
  - **Datos no estructurados**. No representa un modelo y carece de estructura conocida. Requiere ser transformada para su análisis. Ej. Multimedia, texto libre.

5. Tipos de datos (II)

- Aproximadamente sólo un **10%-15%** de los datos disponibles son de tipo estructurado o semi.
- La mayor parte del tiempo dedicada al procesamiento de datos incluye la extracción de información y la **transformación** de los datos de un formato a otro.

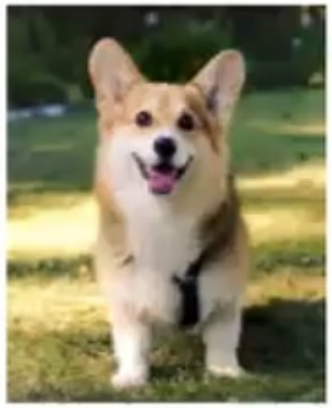
Dato no estructurado



5. Tipos de datos (II)

- Aproximadamente sólo un **10%-15%** de los datos disponibles son de tipo estructurado o semi.
- La mayor parte del tiempo dedicada al procesamiento de datos incluye la extracción de información y la **transformación** de los datos de un formato a otro.

Dato no estructurado **Dato semi-estructurado**



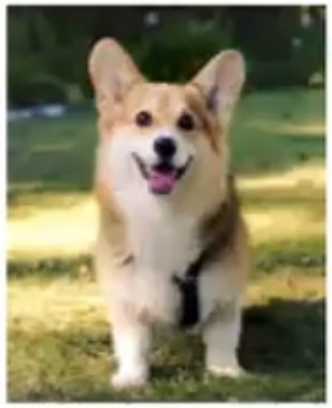
20032017picture3.jpg

```
{
 "file": "20032017picture3.jpg",
 "date": "20-03-2017"
 "tags": ["outdoors", "animal",
"canine", “corgi"]
}
```

5. Tipos de datos (II)

- Aproximadamente sólo un **10%-15%** de los datos disponibles son de tipo estructurado o semi.
- La mayor parte del tiempo dedicada al procesamiento de datos incluye la extracción de información y la **transformación** de los datos de un formato a otro.

Dato no estructurado



20032017picture3.jpg

Dato semi-estructurado

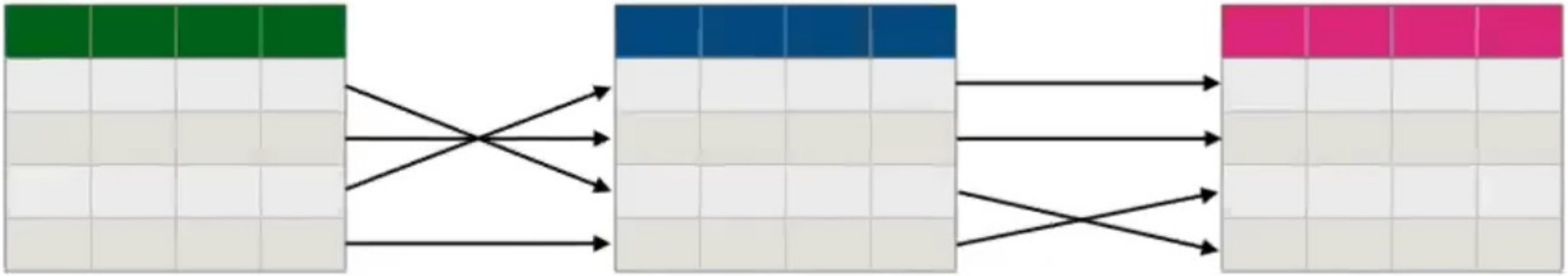
```
{
 "file": "20032017picture3.jpg",
 "date": "20-03-2017"
 "tags": ["outdoors", "animal",
"canine", “corgi"]
}
```

Dato estructurado

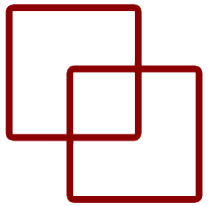
date	species	breed
20/03/2017	canine	corgi

6. Datos estructurados

- Definidos a partir de un modelo estricto:
  - Campos y variables.
  - Relaciones: índices y claves.
  - Restricciones: tipos, longitud y formato.
- Definidos a partir de un modelo estricto, contenido semántico.
- Generalmente asociados con bases de datos relacionales (RDBMS).
- Preparados para ser analizados y consultados mediante algoritmos y lenguajes (SQL)



7. Data wrangling: Transformando nuestros datos



- Objetivo: Transformar datos en bruto (**raw data**) en un formato más apropiado y valioso desde el puntos de vista del análisis.
- El proceso debe realizarse a partir de la forma más pura posible de los datos en bruto y debe documentarse para ser **reproducible**.
- No es suficiente con obtener un conjunto de datos estructurado ya que no siempre es óptimo para el análisis.

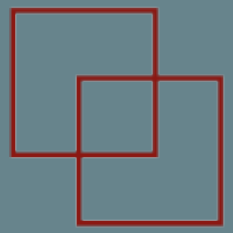
Datos estructurados =/= Buenos datos

- Generalmente buscamos que los datos mantengan unas propiedades deseables que conocemos como: **Tidy Data.**

8. Tidy data

> "Tidy data is a standard way of mapping the meaning of a dataset to its structure. A dataset is messy or tidy depending on how rows, columns and tables are matched up with observations, variables and types"
>
> Hadley Wickham 2014 (Tidy Data - Journal of Statistical Software)

- Un dataset "Tidy" mantiene las siguientes **propiedades**:
  - Cada variable representa una columna
  - Cada observación representa una fila
  - Cada unidad observacional representa una tabla
- Permite definir objetivos, estrategias y herramientas **estandarizadas** para la limpieza y transformación de datos.
- Permite definir un vocabulario y operadores de transformación desde un punto de vista **agnóstico a cualquier lenguaje**.
- Artículo: *Wickham, H. (2014). Tidy data. Journal of Statistical Software*

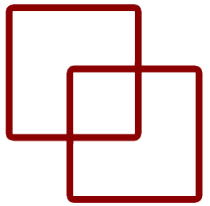


Smart City

LAB CTIC UNI

Cargar un dataset

1. Librería Standar



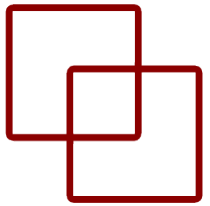
- La API de Python proporciona el módulo CSV y la función *reader()*.

```
# Load CSV Using Python Standard Library
import csv
import numpy

filename = 'data/pima-indians-diabetes.csv'
raw_data = open(filename, 'r')
reader = csv.reader(raw_data, delimiter= ',', quoting=csv.QUOTE_NONE)
x = list(reader)
data = numpy.array(x).astype('float')
print(data.shape)
```

```
(768, 9)
```

2. NumPy

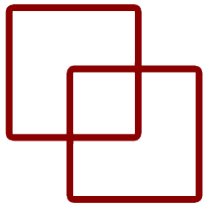


- Puede cargar sus datos CSV usando NumPy y la función *numpy.loadtxt()*.
- Esta función no supone una fila de encabezado y todos los datos tienen el mismo formato.

```
# Load CSV using NumPy
from numpy import loadtxt
filename = 'data/pima-indians-diabetes.csv'
raw_data = open(filename, 'rb')
data = loadtxt(raw_data, delimiter=",")
print(data.shape)
```

```
(768, 9)
```

2.3. Pandas (I)

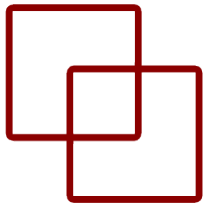


- Puede cargar sus datos CSV usando Pandas y la función *pandas.read_csv()*.
- La más recomendada.

```
# Load CSV using Pandas
import pandas as pd
filename = 'data/pima-indians-diabetes.csv'
names = ['preg', 'plas', 'pres', 'skin' ,'test', 'mass', 'pedi', 'age' ,'class']
data = pd.read_csv(filename, names=names)
print(data.shape)
```

```
(768, 9)
```

2.3. Pandas (II)



- También podemos modificar este ejemplo para cargar datos CSV directamente desde una URL.

```
import pandas as pd
# load dataset
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data'
dataframe = pd.read_csv(url, header=None)
print(dataframe.shape)
```

```
(4177, 9)
```

Smart City

LAB CTIC UNI

Conjunto de datos estándar

1. Iris

- Tipo: clasificación multiclase.
- Dimensiones: 150 instancias, 5 atributos.
- Entradas: Numéricas.
- Resultado: Categórico, 3 etiquetas de clase.

```
> data(iris)
> head(iris)
  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
```

CÓDIGO 2.25: Resumen del conjunto de datos Iris

2. Económica de Longley

- Descripción: Predecir el número de personas empleadas a partir de variables económicas.
- Tipo: Regresión.
- Dimensiones: 16 instancias, 7 atributos.
- Entradas: Numéricas.
- Salida: Numérico.

```
> data(longley)
> head(longley)
     GNP.deflator     GNP Unemployed Armed.Forces Population Year
1947         83.0 234.289      235.6        159.0    107.608 1947
1948         88.5 259.426      232.5        145.6    108.632 1948
1949         88.2 258.054      368.2        161.6    109.773 1949
1950         89.5 284.599      335.1        165.0    110.929 1950
1951         96.2 328.975      209.9        309.9    112.075 1951
1952         98.1 346.999      193.2        359.4    113.270 1952
```

CÓDIGO 2.26: Resumen del conjunto de datos Longley

3. Boston Housing

- Descripción: Predecir el precio medio de una casa en los suburbios de Boston.
- Tipo: Regresión.
- Dimensiones: 506 instancias, 14 atributos.
- Entradas: Numéricas.
- Resultado: Numérico.

```
> data(BostonHousing)
> head(BostonHousing)
     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
```

4. BreastCancer Wisconsin

- Descripción: Predecir si una muestra de tejido es maligna o benigna dadas propiedades sobre la muestra de tejido.
- Tipo: Clasificación Binaria.
- Dimensiones: 699 instancias, 11 atributos.
- Entradas: Entero (Nominal).
- Resultado: Categórica, 2 clases.

```
> data(BreastCancer)
> head(BreastCancer)
       Id Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size
1 1000025            5         1          1             1            2
2 1002945            5         4          4             5            7
3 1015425            3         1          1             1            2
4 1016277            6         8          8             1            3
5 1017023            4         1          1             3            2
6 1017122            8        10         10             8            7
  Bare.nuclei Bl.cromatin Normal.nucleoli Mitoses     Class
1           1           3                1       1    benign
2          10           3                2       1    benign
3           2           3                1       1    benign
4           4           3                7       1    benign
5           1           3                1       1    benign
6          10           9                7       1 malignant
```

5. Identificación de vidrio

- Descripción: Predecir el tipo de vidrio a partir de propiedades químicas.
- Tipo: Regresión.
- Dimensiones: 244 instancias, 10 atributos.
- Entradas: Numérico.
- Resultado: Categórica, 7 clases.

```
> data(Glass)
> head(Glass)
        RI    Na   Mg   Al    Si    K   Ca Ba   Fe Type
1 1.52101 13.64 4.49 1.10 71.78 0.06 8.75  0 0.00    1
2 1.51761 13.89 3.60 1.36 72.73 0.48 7.83  0 0.00    1
3 1.51618 13.53 3.55 1.54 72.99 0.39 7.78  0 0.00    1
4 1.51766 13.21 3.69 1.29 72.61 0.57 8.22  0 0.00    1
5 1.51742 13.27 3.62 1.24 73.08 0.55 8.07  0 0.00    1
6 1.51596 12.79 3.61 1.62 72.97 0.64 8.07  0 0.26    1
```

6. Ionosphere

- Descripción: Predecir estructuras de alta energía en la atmósfera a partir de datos de antena.
- Tipo: Regresión.
- Dimensiones: 351 instancias, 35 atributos.
- Entradas: Numérico.
- Resultado: Categórica, 2 clases.

```
> data(Ionosphere)
> head(Ionosphere)
  V1 V2      V3       V4       V5       V6       V7       V8      V9       V34 Class
1  1  0 0.99539 -0.05889  0.85243  0.02306  0.83398 -0.37708 1.00000   0.45300  good
2  1  0 1.00000 -0.18829  0.93035 -0.36156 -0.10868 -0.93597 1.00000   0.02447   bad
3  1  0 1.00000 -0.03365  1.00000  0.00485  1.00000 -0.12062 0.88965   0.38238  good
4  1  0 1.00000 -0.45161  1.00000  1.00000  0.71216 -1.00000 0.00000   1.00000   bad
5  1  0 1.00000 -0.02401  0.94140  0.06531  0.92106 -0.23255 0.77152   0.65697  good
6  1  0 0.02337 -0.00592 -0.09924 -0.11949 -0.00763 -0.11824 0.14706   0.12011   bad
```

7. Diabetes de Pima Indians

- Descripción: Predecir el inicio de la diabetes en mujeres indias pima a partir de datos de registros médicos.
- Tipo: Clasificación Binaria.
- Dimensiones: 768 instancias, 9 atributos.
- Entradas: Numérico.
- Resultado: Categórica, 2 clases.

```
> data(PimaIndiansDiabetes)
> head(PimaIndiansDiabetes)
  pregnant glucose pressure triceps insulin mass pedigree age diabetes
1        6     148       72      35       0 33.6    0.627  50      pos
2        1      85       66      29       0 26.6    0.351  31      neg
3        8     183       64       0       0 23.3    0.672  32      pos
4        1      89       66      23      94 28.1    0.167  21      neg
5        0     137       40      35     168 43.1    2.288  33      pos
6        5     116       74       0       0 25.6    0.201  30      neg
```

8. Sonar, Mines vs. Rocks

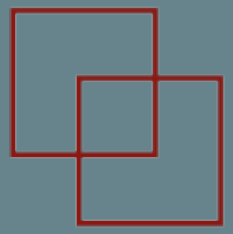
- Descripción: Predice los retornos de metal o roca a partir de los datos de retorno del sonar.
- Tipo: Clasificación Binaria.
- Dimensiones: 208 instancias, 61 atributos.
- Entradas: Numérico.
- Resultado: Categórica, 2 clases.

```
> data(Sonar)
> head(Sonar)
     V1     V2     V3     V4     V5     V6     V7     V8     V9    V10 Class
1 0.0200 0.0371 0.0428 0.0207 0.0954 0.0986 0.1539 0.1601 0.3109 0.2111     R
2 0.0453 0.0523 0.0843 0.0689 0.1183 0.2583 0.2156 0.3481 0.3337 0.2872     R
3 0.0262 0.0582 0.1099 0.1083 0.0974 0.2280 0.2431 0.3771 0.5598 0.6194     R
4 0.0100 0.0171 0.0623 0.0205 0.0205 0.0368 0.1098 0.1276 0.0598 0.1264     R
5 0.0762 0.0666 0.0481 0.0394 0.0590 0.0649 0.1209 0.2467 0.3564 0.4459     R
6 0.0286 0.0453 0.0277 0.0174 0.0384 0.0990 0.1201 0.1833 0.2105 0.3039     R
```

9. Base de datos Soya

- Descripción: Predecir problemas con cultivos de soja a partir de datos de cultivos.
- Tipo: Clasificación Multiclase.
- Dimensiones: 683 instancias, 26 atributos.
- Entradas: Entero (Nominal).
- Resultado: Categórica, 19 clases.

```
> data(Soybean)
> head(Soybean)
                  Class date plant.stand precip temp hail crop.hist
1 diaporthe-stem-canker    6           0      2    1    0         1
2 diaporthe-stem-canker    4           0      2    1    0         2
3 diaporthe-stem-canker    3           0      2    1    0         1
4 diaporthe-stem-canker    3           0      2    1    0         1
5 diaporthe-stem-canker    6           0      2    1    0         2
6 diaporthe-stem-canker    5           0      2    1    0         3
```



Smart City

LAB CTIC UNI

Estadística descriptiva

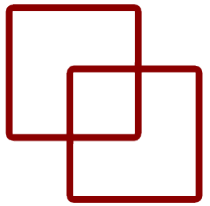
1. Introducción

- Necesidad de comprender los datos a la perfección.
  - Finalidad: Tener un modelo robusto y fiable
- Trabajaremos:
  - Trabajar con las técnicas principales para aprender de nuestros datos.
  - Entender nuestros datos a través de la observación estadística como: las dimensiones, tipo de datos, distribución de clases, entre otros.
  - Trabajar estadísticas avanzadas en el análisis de un conjunto de datos como desviaciones estándar y sesgo de los datos.
  - Entender las relaciones entre los atributos mediante el cálculo de correlaciones.

2. Entender nuestros datos

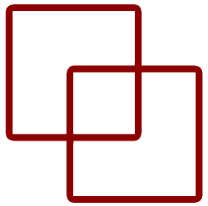
- Error típico: Trabajar directamente algoritmos sin conocer nuestros datos.
- Una buena cocina de datos (análisis y procesamiento) nos lleva a obtener mejores y más confiables resultados, entendiendo nuestro conjunto de datos.
- Las tareas a realizar son:
  - **Limpieza de datos:** Descubrimos datos *missing* o corruptos y/o marcamos o imputamos datos faltantes.
  - **Transformación de datos**: Descubrir distribuciones como la gaussiana o exponencial que nos genera una idea de qué técnica utilizar, p.e., escalamiento o transformación de datos.
  - **Modelado de datos**: A través de la estadística descriptiva obtenemos una idea de qué algoritmo poder utilizar.

3. Entender nuestros datos



- Python (y sus librerías) nos da una gran variedad de funciones para conocer en profundidad nuestros datos.
  - A través de ellas vamos a conocer muchos detalles del conjunto de datos.
  - Nos aproximamos a qué técnica dará mejor resultado.
- Deberemos entonces conocer:
  - La forma, el tamaño, el tipo y el diseño general que tienen los datos.

3.1. Conocer nuestros datos



- La función "*head()*" nos permite ver nuestras instancias.

```
# View first 20 rows
data.head(10)
```

	preg	plas	pres	skin	test	mass	pedi	age	class
0	6	148	72	35	0	33.6	627.0	50	1
1	1	85	66	29	0	26.6	351.0	31	0
2	8	183	64	0	0	23.3	672.0	32	1
3	1	89	66	23	94	28.1	167.0	21	0
4	0	137	40	35	168	43.1	2288.0	33	1
5	5	116	74	0	0	25.6	201.0	30	0

3.2. Dimensiones del dataset

- Especifica el número de atributos e instancias.
  - Muchas instancias entonces es convenientes disminuirlas (capacidad computacional).
  - Muchos atributos entonces problema de la dimensionalidad
- Utilizaremos la propiedad "*shape*".

```
# Dimensions of your data
data.shape
```

```
(768, 9)
```

3.3. Tipos de datos

- Nos proporcionará algunas pistas sobre qué:
  - Tipo de análisis utilizar,
  - Tipo de visualización del conjunto de datos
  - Algoritmo de ML puede ser un buen candidato.
- Por ejemplo, algoritmos tipo árbol prefieren atributos categóricos.
- Utilizaremos la propiedad "*dtypes*".

```
# Data Types for Each Attribute
data.dtypes
```

```
preg        int64
plas        int64
pres        int64
skin        int64
test        int64
mass      float64
pedi      float64
age         int64
class       int64
dtype: object
```

3.4. Resumen de los datos

- Muestra las estadísticas primarias para cada atributo. Muestra:
  - Mínimo (Min.) / Valor del 25 % Percentil (1st Qu.) / Mediana (Median) - Media (Mean) / Valor del 75 % Percentil (3st Qu.) / Máximo (Max.).
- Utilizaremos la función "*describe()*".

```
# Statistical Summary
data.describe()
```

	preg	plas	pres	skin	test	mass	pedi	age	class
count	768.000	768.000	768.000	768.000	768.000	768.000	768.000	768.000	768.000
mean	3.845	120.895	69.105	20.536	79.799	31.993	428.235	33.241	0.349
std	3.370	31.973	19.356	15.952	115.244	7.884	340.486	11.760	0.477
min	0.000	0.000	0.000	0.000	0.000	0.000	0.100	21.000	0.000
25%	1.000	99.000	62.000	0.000	0.000	27.300	205.000	24.000	0.000
50%	3.000	117.000	72.000	23.000	30.500	32.000	337.000	29.000	0.000
75%	6.000	140.250	80.000	32.000	127.250	36.600	591.500	41.000	1.000
max	17.000	199.000	122.000	99.000	846.000	67.100	2329.000	81.000	1.000

3.5. Distribución de clase

- Muchas veces un conjunto de datos para un problema de clasificación puede encontrarse desbalanceado:
  - No existe un equilibrio entre el número total de instancias para cada clase.
  - En este caso, se tendrán que aplicar técnicas de rebalanceo de clases.
- Función: "*groupby('class').size()*".

```
# Class Distribution
data.groupby('class').size()
```

```
class
0    500
1    268
dtype: int64
```

3.6. Correlaciones

- Se define la relación entre pares de atributos numéricos.
- Los valores superiores a aproximadamente 0,75 o inferiores a -0,75 son los más interesantes ya que muestran una alta correlación.
- 1 y -1 correlación positiva o negativa completa.
- Utilizaremos la función "*cor()*".

```
# Pairwise Pearson correlations
correlations = data.corr(method='pearson')
print(correlations)
```

```
        preg   plas   pres   skin   test   mass   pedi    age  class
preg   1.000  0.129  0.141 -0.082 -0.074  0.018 -0.026  0.544  0.222
plas   0.129  1.000  0.153  0.057  0.331  0.221  0.133  0.264  0.467
pres   0.141  0.153  1.000  0.207  0.089  0.282  0.051  0.240  0.065
skin  -0.082  0.057  0.207  1.000  0.437  0.393  0.154 -0.114  0.075
test  -0.074  0.331  0.089  0.437  1.000  0.198  0.185 -0.042  0.131
mass   0.018  0.221  0.282  0.393  0.198  1.000  0.104  0.036  0.293
pedi  -0.026  0.133  0.051  0.154  0.185  0.104  1.000  0.018  0.177
age    0.544  0.264  0.240 -0.114 -0.042  0.036  0.018  1.000  0.238
class  0.222  0.467  0.065  0.075  0.131  0.293  0.177  0.238  1.000
```

3.7. Asimetría

- Si una distribución parece casi gaussiana pero se empuja hacia la izquierda o hacia la derecha, es útil conocer el sesgo.
- Valores cercanos a cero tienen un menor sesgo, sin embargo, sesgo a la izquierda será con valores negativos y sesgo a la derecha serán valores positivos.
- Función: "*skew()*".

```
# Skew for each attribute
data.skew()
```

```
preg     0.901674
plas     0.173754
pres    -1.843608
skin     0.109372
test     2.272251
mass    -0.428982
pedi     1.561978
age      1.129597
class    0.635017
dtype: float64
```

UNIVERSIDAD PERUANA
CAYETANO HEREDIA

Visualización de datos

1. Introducción

- Mediante gráficos y diagramas de los datos en bruto.
- Los diagramas de atributos nos proporciona una gran ayuda a la hora de poder detectar datos atípicos, datos extraños o no válidos.
- Además, nos proporciona una idea de las posibles transformaciones de datos que podríamos aplicar en la fase de procesamiento de datos.

1.1. Visualización

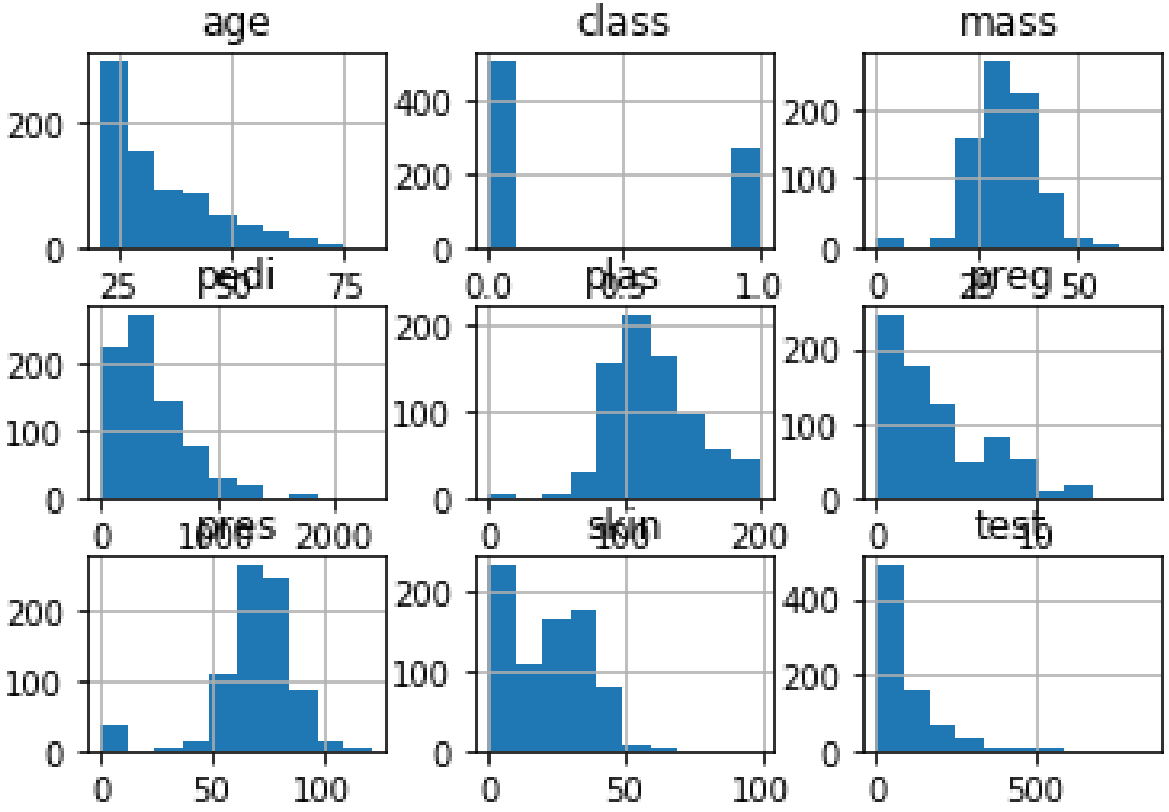
- Diferentes partes:
  - Paquetes de visualización.
  - Visualización univariable: mostrar atributos independientemente.
  - Visualización multivariable: mostrar diferentes atributos.
- Paquetes:
  - "*Matplotlib*": Excelente para gráficos rápidos y básicos de datos.
  - "*Seaborn*": Más profesionales y, a menudo, más útiles en la práctica.

UNIVERSIDAD PERUANA
CAYETANO HEREDIA

Visualización Univariable

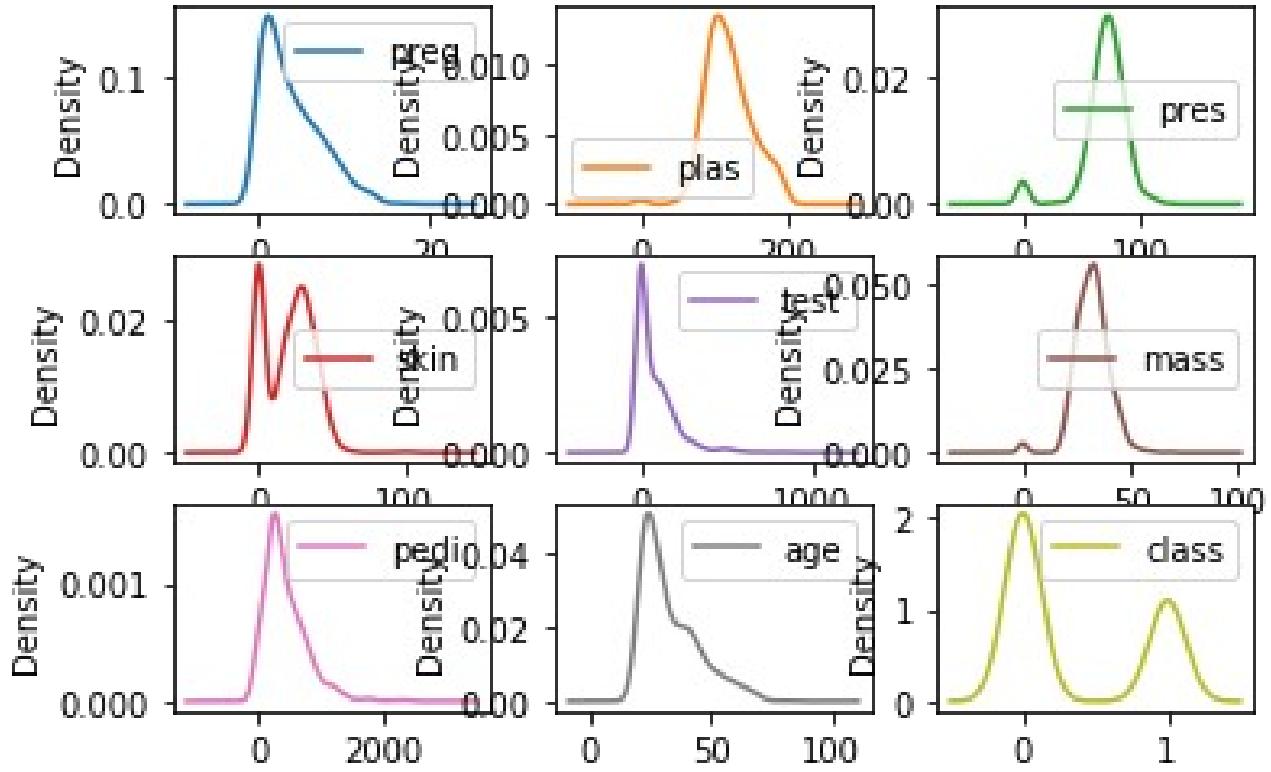
2.1. Histogramas

- Muestran un gráfico de barras a un atributo numérico.
- Su altura muestra el número de instancias
- Utilizaremos la función "*hist()*".



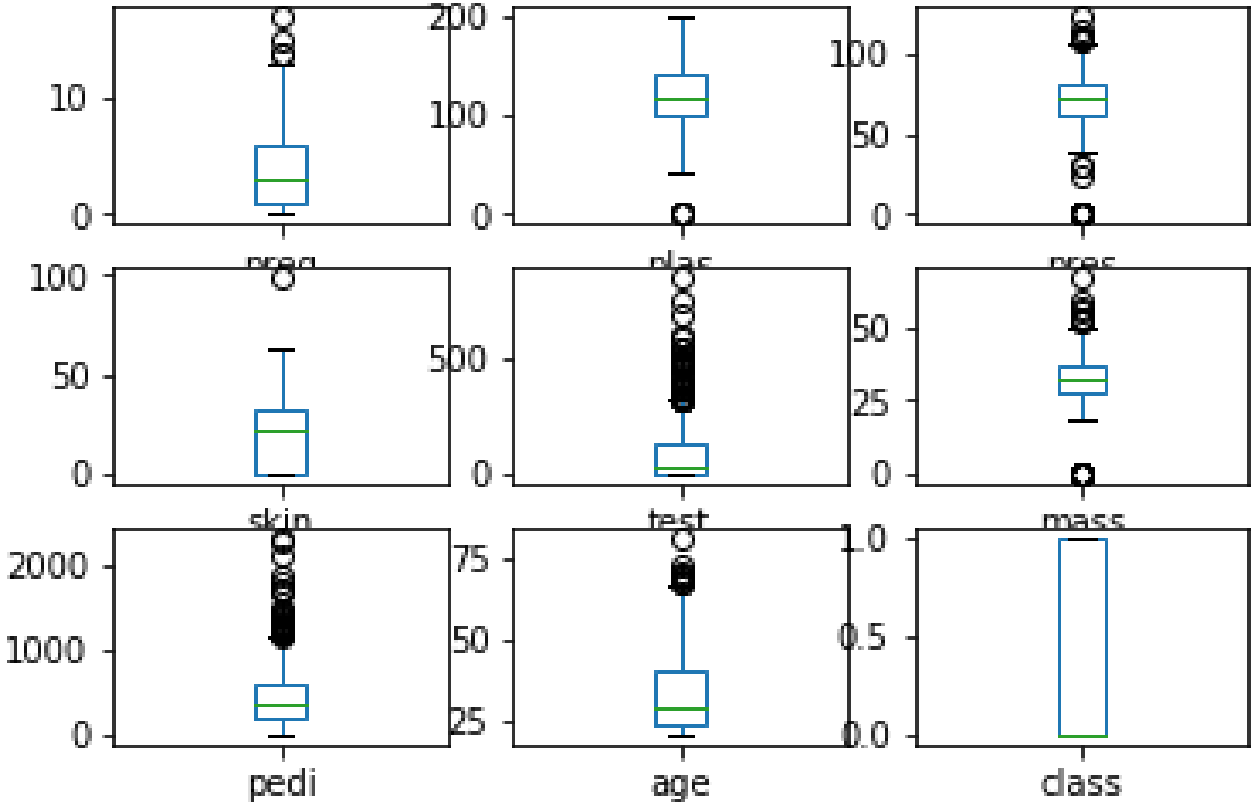
2.2. Densidad

- Visualizar los histogramas a líneas.
- Utilizaremos la función "*plot()*" con el parámetro *kind='density'*.
  - Puede observarse la la doble distribución gaussiana de *skin*.
  - Una posible distribución exponencial (similar a Lapacian) de *age*.
  - Un posible sesgo en *plas*.



2.3. Boxplot

- La representación de este tipo de gráfico es como sigue:
  - La caja captura el 50% medio de los datos.
  - La línea muestra la mediana.
  - Los bordes de los gráficos muestran la extensión razonable de los datos.
  - Cualquier punto fuera de los bordes es un buen candidato para los valores atípicos (*outlayers*).
- Utilizaremos la función "*plot()*" con el parámetro *kind='box'*.

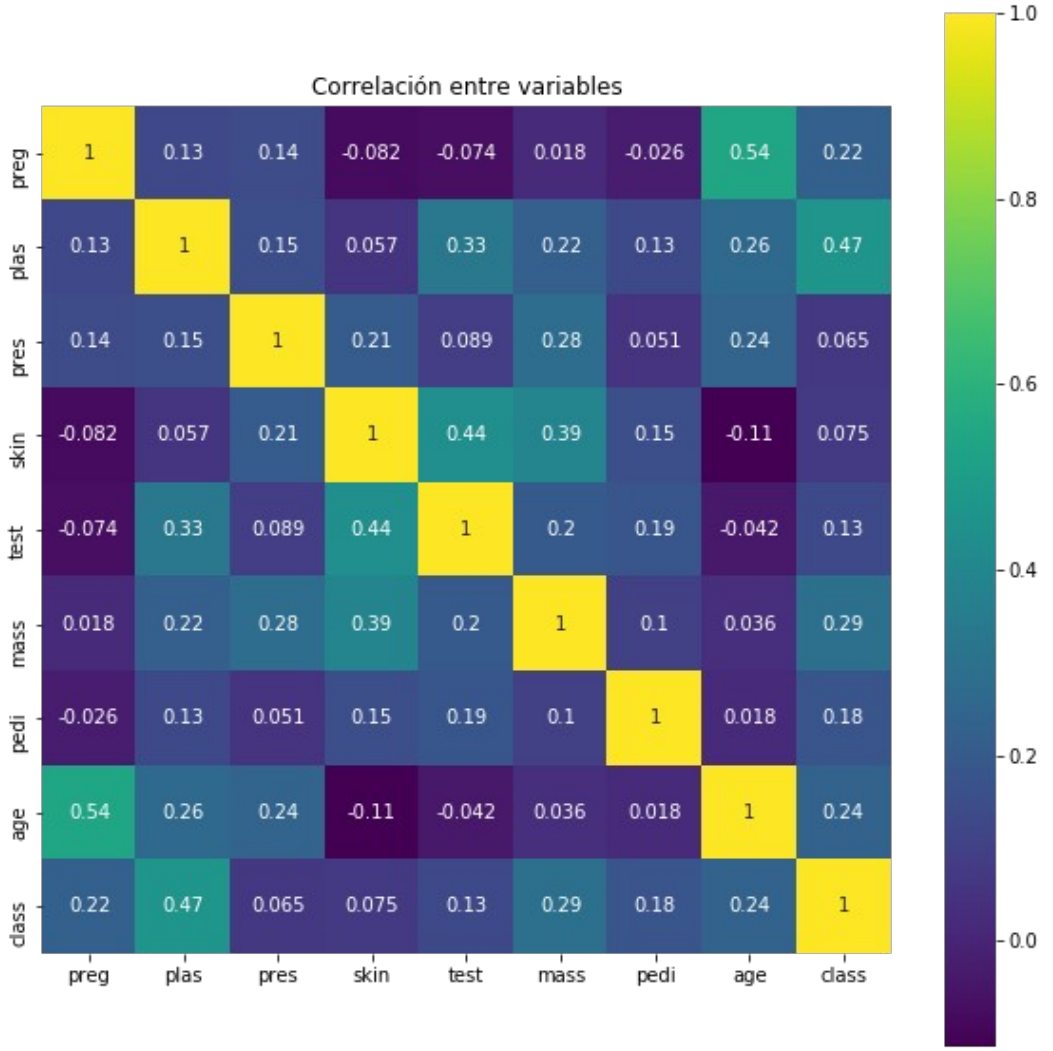


UNIVERSIDAD PERUANA
CAYETANO HEREDIA

Visualización Multivariable

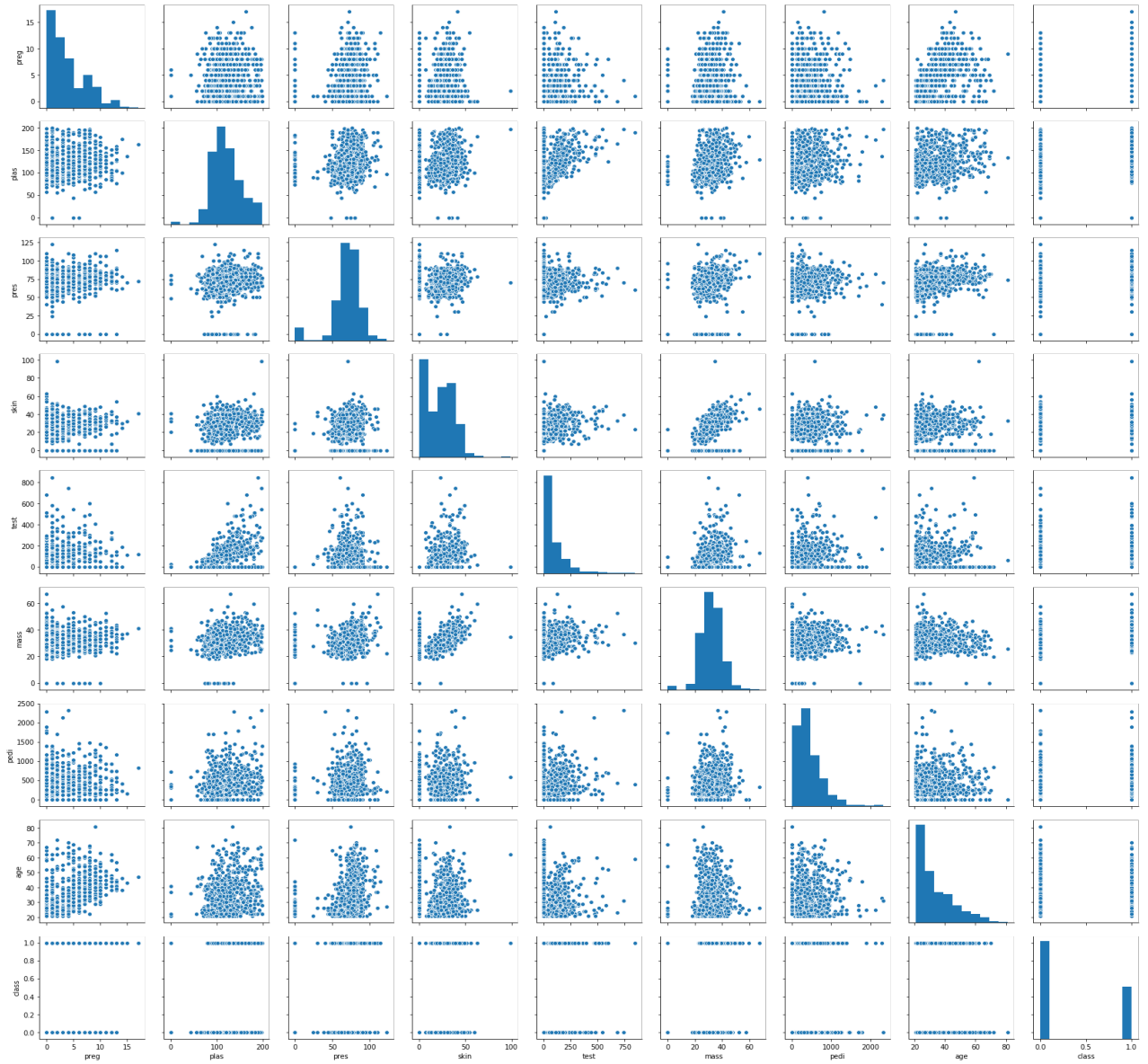
3.1. Correlación

- Evaluar la correlación entre nuestros atributos de tipo numérico, esto es, evaluar que atributos cambian juntos.
- Se puede observar:
  - Poca correlación entre atributos.
  - Casi siempre correlación positiva.



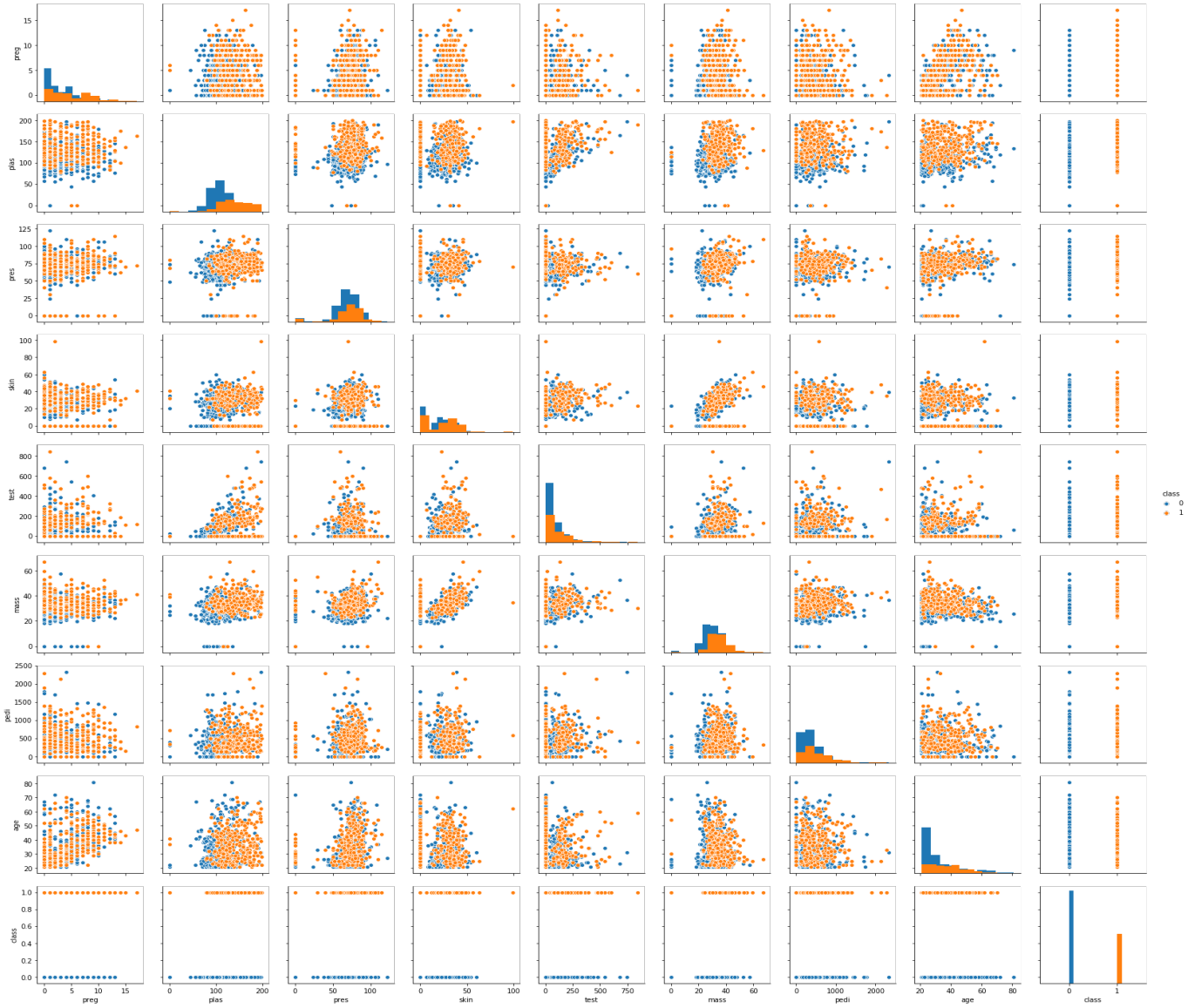
3.2. Matriz de dispersión

- Los diagramas de dispersión traza dos variables juntas, tanto en los ejes X e Y con puntos que muestran la interacción entre estos dos atributos.
- Son interesantes para notar comportamientos lineales o clusterizados entre variables
  - Una relación lineal (línea diagonal), que esas dos variables pueden ser predominantes a la hora de utilizar un algoritmo.



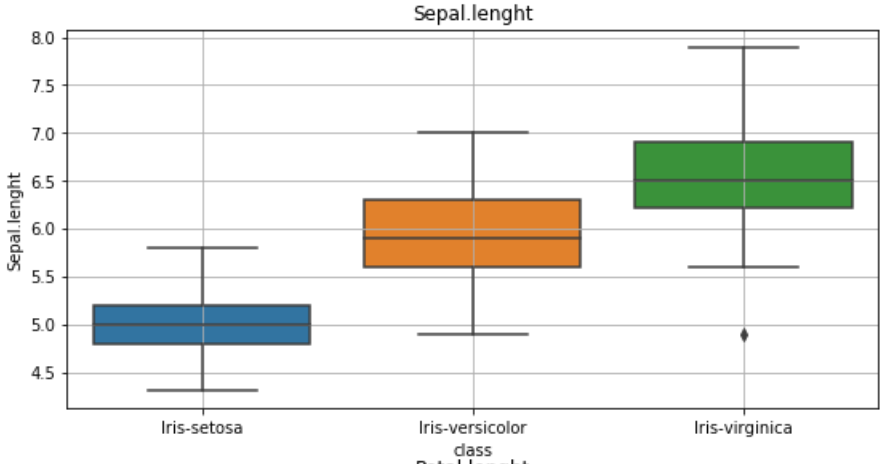
3.3. Matriz de dispersión por clase

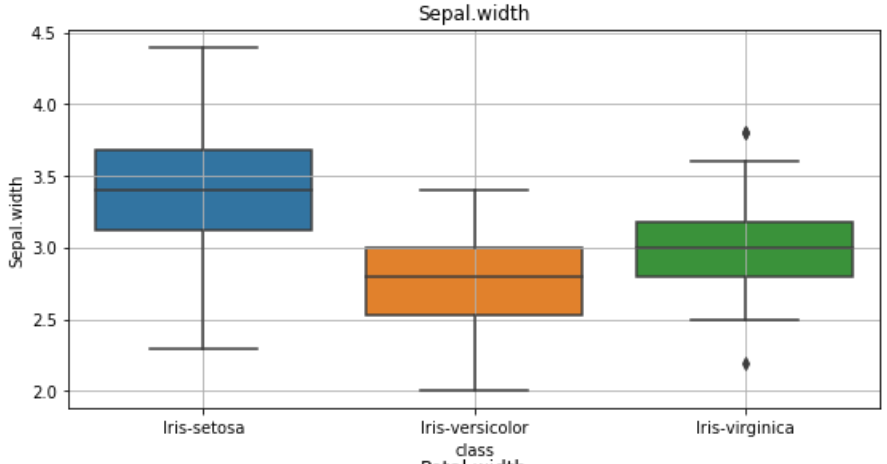
- Los puntos en una matriz de diagrama de dispersión pueden ser coloreados por la etiqueta de clase en problemas de clasificación.
  - Nos ayuda a detectar una separación clara (o no clara) de las clases y quizás a darnos una idea de lo difícil que puede ser el problema.
- Los diagramas están dados por la interacción de los pares de atributos pero, en este caso, teniendo en cuenta la etiqueta clase.

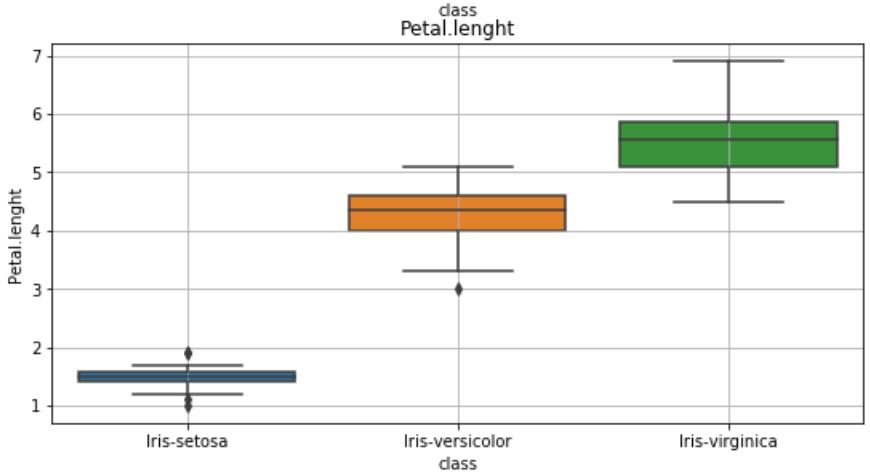


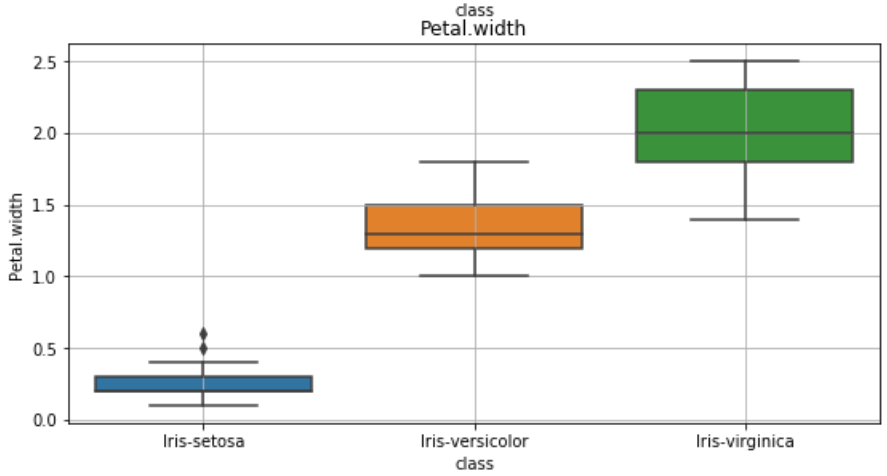
3.4. Boxplot por clase

- Nos permite comprender cómo se relaciona cada atributo con el valor de la clase.

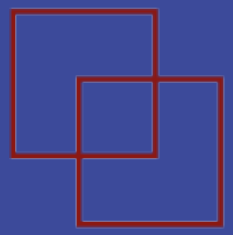








¡GRACIAS!



Smart City

LAB CTIC UNI

Dr. Manuel Castillo-Cara
Intelligent Ubiquitous Technologies – Smart Cities (IUT-SCi)
Web: www.smartcityperu.org