

**clusterAI 2023**  
**ciencia de datos en ingeniería industrial**  
**UTN BA**  
**curso I5521**

**clase\_00**



**clusterAI**

## Arte generativo



Mario Klingemann

- Consignas de la materia, presentación del equipo docente
- Features, samples, the curse of dimensionality
- Tipos de Aprendizaje, Supervisado vs No Supervisado
- Tipos de Datos
- Formato de los datos
- Data Science Workflow
- Primeras prácticas con Python

# Propuesta de valor del curso

Preparar a los futuros ingenier@s, profesionales, entusiastas y emprendedores para lidiar con complejidad en el contexto de la 4ta revolución industrial\*.

\*<https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/>

# Misión del curso

Lograr que el estudiante termine el curso incorporando:

1. Conocimiento básico-intermedio de análisis de datos con python
2. Fundamentos y Métodos clásicos y no tanto de machine learning.
3. Realización y comunicación de un proyecto con modelos y técnicas de data science y machine learning sobre datos reales.
4. Incorporándose a una comunidad activa y creciente de ciencia de datos.

# Herramientas del curso

- Comunicación semanal: por medio del google groups del curso.
- Comunicación diaria y comunitaria: canal de slack.
- Contexto del contenido: Diapositivas de clase (apuntes solamente de soporte) + jupyter notebooks + canal de youtube.
- Programar: Python (Anaconda, Jupyter) para programar.
- Teoría de base: Libros.
- Ejercitación: Guía de ejercicios.

# docentes clusterAI



## **Martin Palazzo**

Docente coordinador ClusterAI  
Head of Machine Learning in Stamm Biotech  
PhD Engineering  
Université de Technologie de Troyes & UTN FRBA  
Master OSS (UTN-UTT)  
Ingeniero Industrial UTN BA



## **Nicolas Aguirre**

Docente ClusterAI  
Machine Learning Engineer  
Phd Engineering  
UTN-Université de Technologie de Troyes  
Master OSS (UTN-UTT)  
Ingeniero Industrial UTN BA



## **Santiago Chas**

Docente Trabajos Prácticos ClusterAI  
Data Scientist AlixPartners  
Master candidate UBA-Data mining  
Ingeniero Industrial UTN BA

# Cluster online



**clusteraigroup@gmail.com**



**facebook.com/clusteraigroup/**



**twitter.com/clusteraigroup**



**github.com/clusteraigroup**



**linkedin.com/company/clusteraigroup**



# Bibliografia

**Python Data Science handbook**

<https://jakevdp.github.io/PythonDataScienceHandbook/>

**An introduction to statistical learning**

<https://www.statlearning.com/>

**Probabilistic Machine Learning**

<https://probml.github.io/pml-book/book1.html>

**Deep Learning book**

<https://www.deeplearningbook.org>

**Elements of Statistical Learning (Tibshirani)**

**Hands on Machine Learning with Scikit Learn (Geron)**

**Machine Learning & Pattern Recognition (Bishop)**

# Estructura del curso

**Datos  
Estructurados**

Análisis  
**Exploratorio**  
de Datos

Aprendizaje  
Supervisado:  
**Clasificación y  
regresión.**

Reducción  
de dim. +  
regularización

Aprendizaje  
No  
supervisado:  
**clustering**

**Redes neuronales**

Redes  
Neuronales:  
**Clasificación**

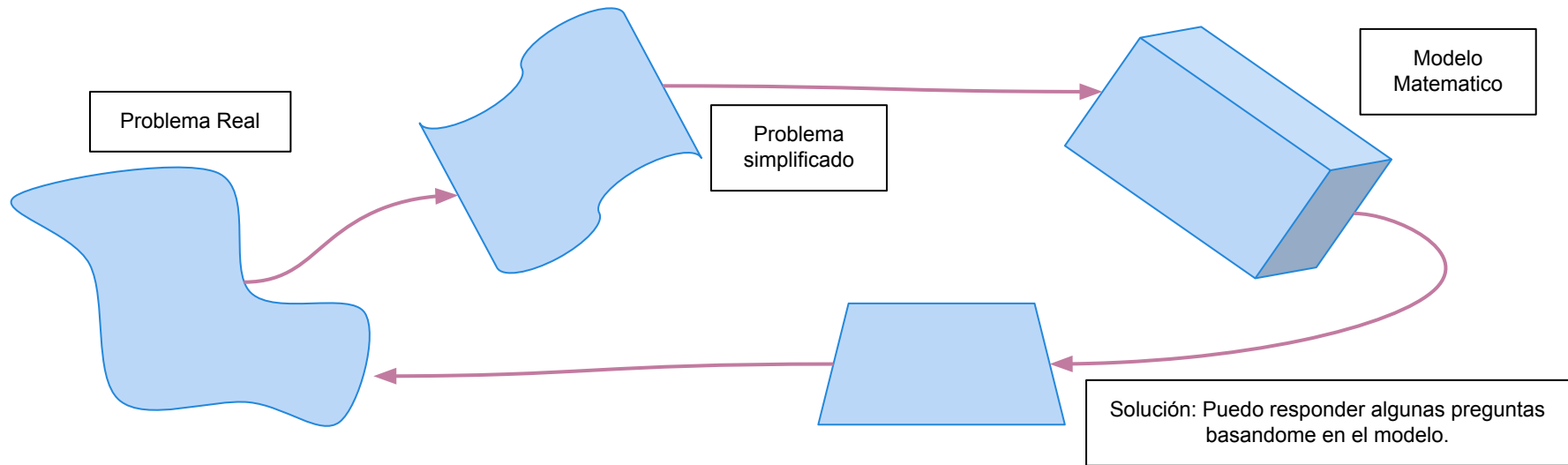
Redes  
Neuronales:  
**Autoencoders**

# Requisitos de aprobación

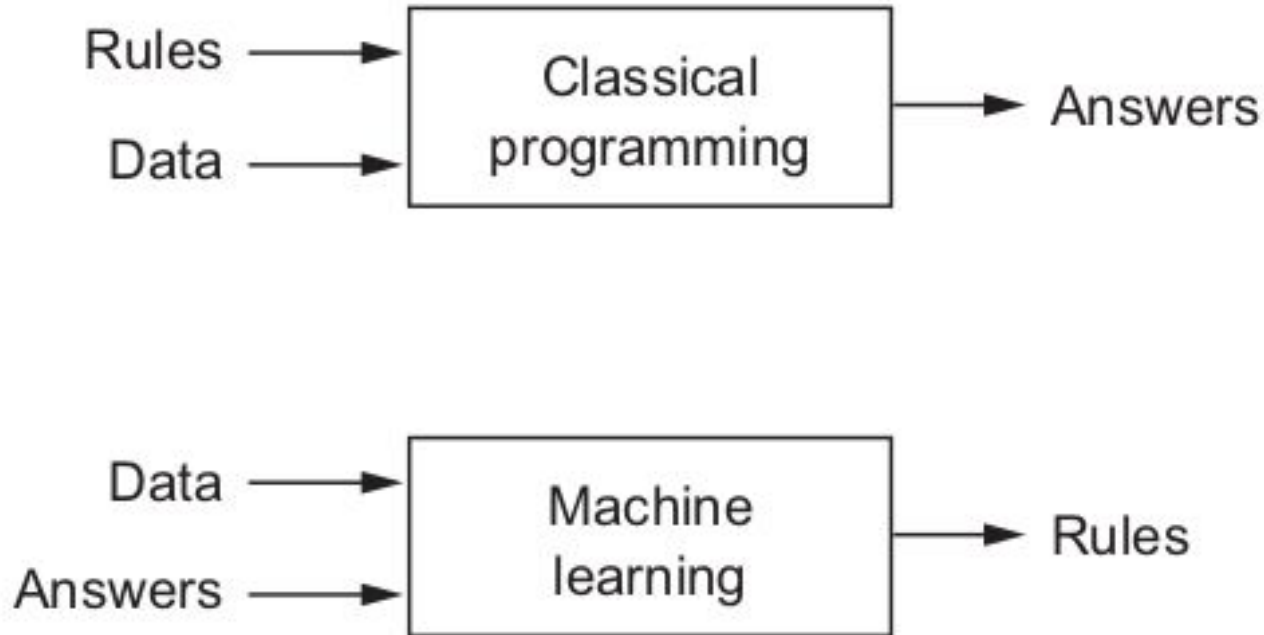
- *Asistencia y seguimiento de las clases (4 faltas máximo).*
- Trabajos prácticos (20%)
- Parcial teórico (40%)
- Trabajo práctico final (40%)

**aprender y construir modelos  
desde los datos.**

# Modelado matematico



# machine learning in a nutshell

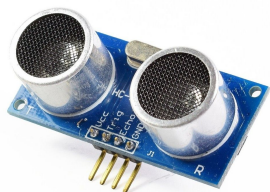


# Samples (instancia) & Features (atributos)

Un dataset es un conjunto de instancias obtenidas a partir de un sensor.

## Samples/instancias

Las samples corresponden a las instancias que obtenemos de una **muestra** de datos a partir de un sensor. Dicha muestra pertenece a una población que generalmente no conocemos por completo. Nuestro set de datos tendrá una cantidad  $n$  de samples.



$$S : \{x_1, x_2, x_3, \dots, x_n\}$$

# Samples & Features

$$x \in \mathbb{R}^d \quad x_i^T = [x_{i1}, x_{i2}, x_{i3}, \dots, x_{id}, ]$$

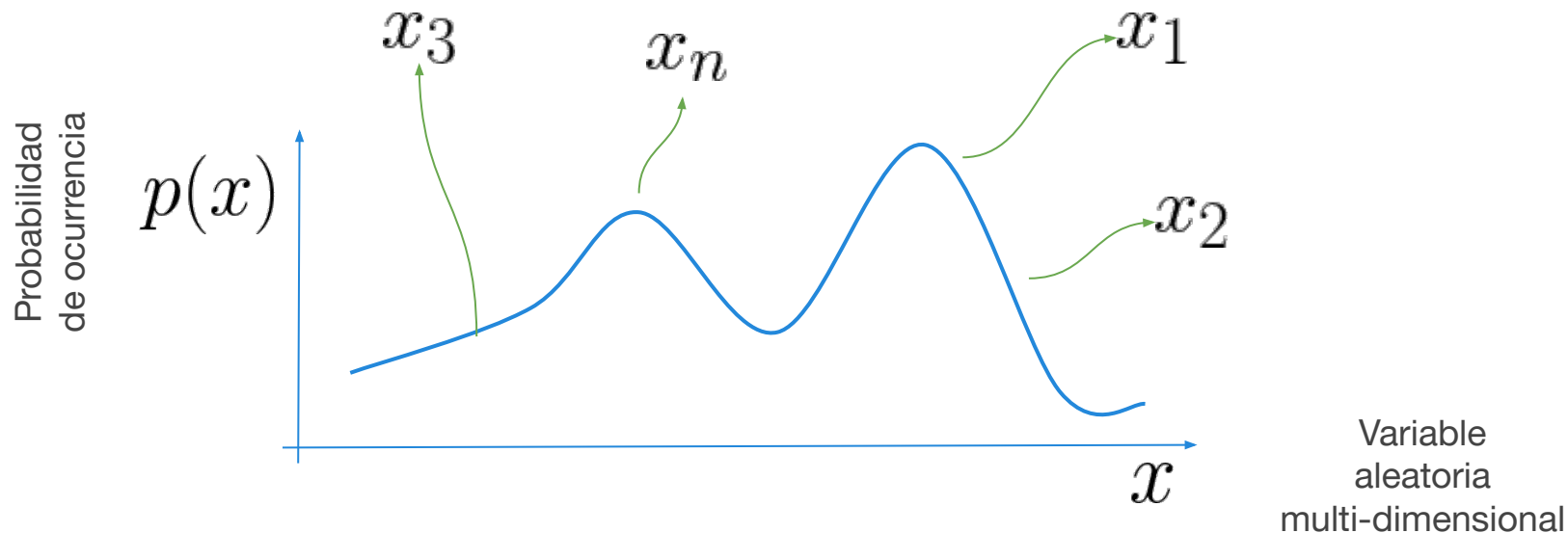
←----- Dimensiones -----→

## Features/variables/atributos/dimensiones

Denominamos features (atributos o mediciones) a las **variables** que definen y caracterizan a cada sample (instancia). La cantidad de features/variables que posea un sample es equivalente a la cantidad de **dimensiones** que describen a esa instancia en un espacio vectorial de ***d*** dimensiones. Entonces cada sample podemos considerarla un vector de ***d*** dimensiones. Nuestros datos “viven” en un espacio d-dimensional. Además, cada una de la dimensiones es considerada una variable aleatoria que sigue una distribución de probabilidad conocida o desconocida.



# Muestreo desde una función de densidad de probabilidad (desconocida)

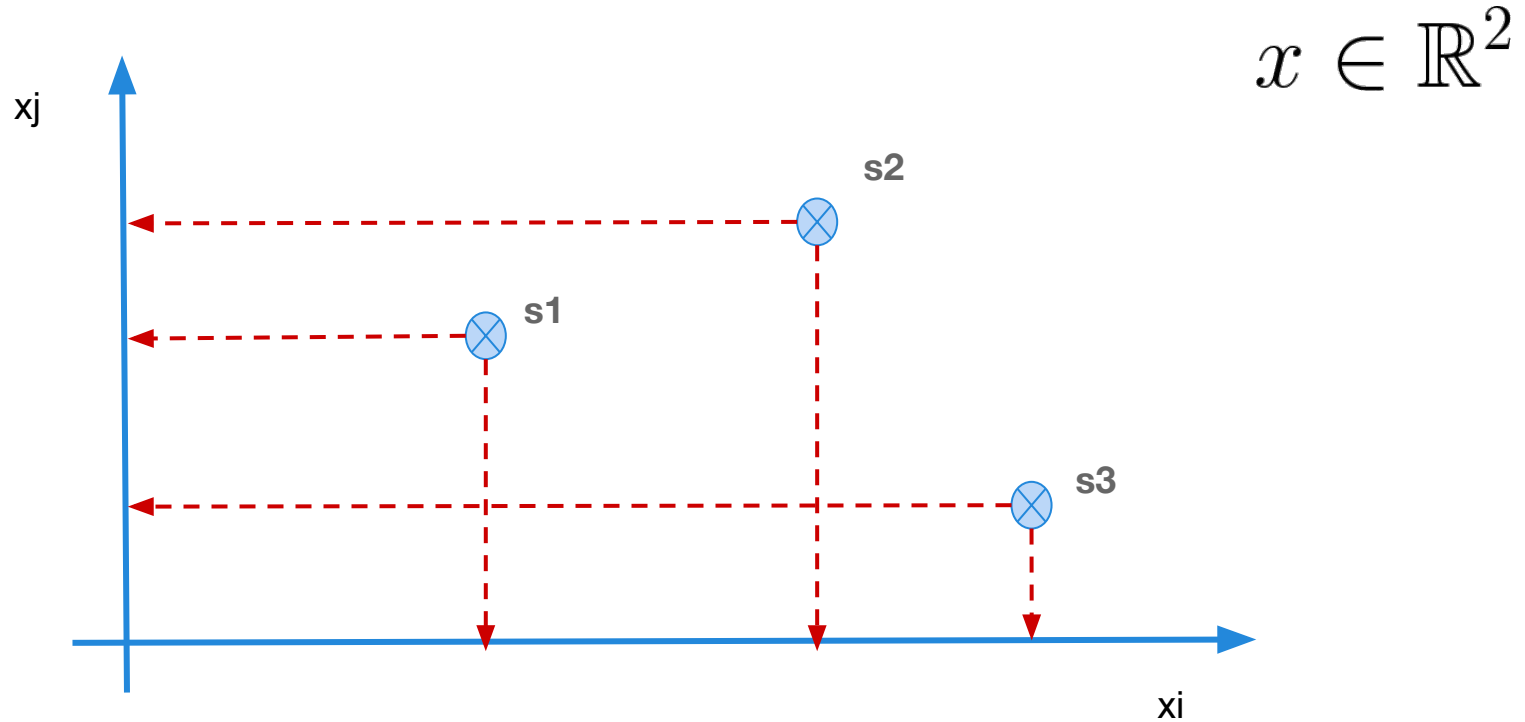


Mediante sensores vamos a realizar multiples mediciones de un sistema, es decir vamos a **muestrear** multiples veces una variable aleatoria y obtener distintas realizaciones.

[1] [https://en.wikipedia.org/wiki/Maximum\\_likelihood\\_estimation](https://en.wikipedia.org/wiki/Maximum_likelihood_estimation)

[2] [https://en.wikipedia.org/wiki/Kernel\\_density\\_estimation](https://en.wikipedia.org/wiki/Kernel_density_estimation).


# Samples (instancia) & Features (atributos)

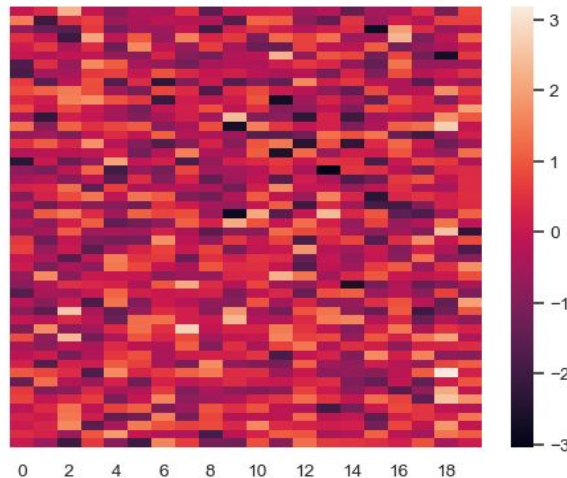


¿Cuántas features/variables/dimensiones y cuántas instancias/samples hay en este ejemplo?

# Data matrix

$$S : \{x_1, x_2, x_3, \dots, x_n\}$$


$$\mathbf{X}_{(n,d)} = \begin{bmatrix} x_{00} & x_{01} & \dots & x_{0d} \\ x_{10} & x_{11} & \dots & x_{1d} \\ \dots & \dots & \dots & \dots \\ x_{n0} & x_{n1} & \dots & x_{nd} \end{bmatrix}$$



# Sample-to-feature ratio

“n” samples

“d” features

$$\mathbf{X}_{(n,d)} = \begin{bmatrix} x_{00} & x_{01} & \dots & x_{0d} \\ x_{10} & x_{11} & \dots & x_{1d} \\ \dots & \dots & \dots & \dots \\ x_{n0} & x_{n1} & \dots & x_{nd} \end{bmatrix}$$
$$S2FR = \frac{n}{d}$$

$S2FR \gg 1$

$S2FR \ll 1$

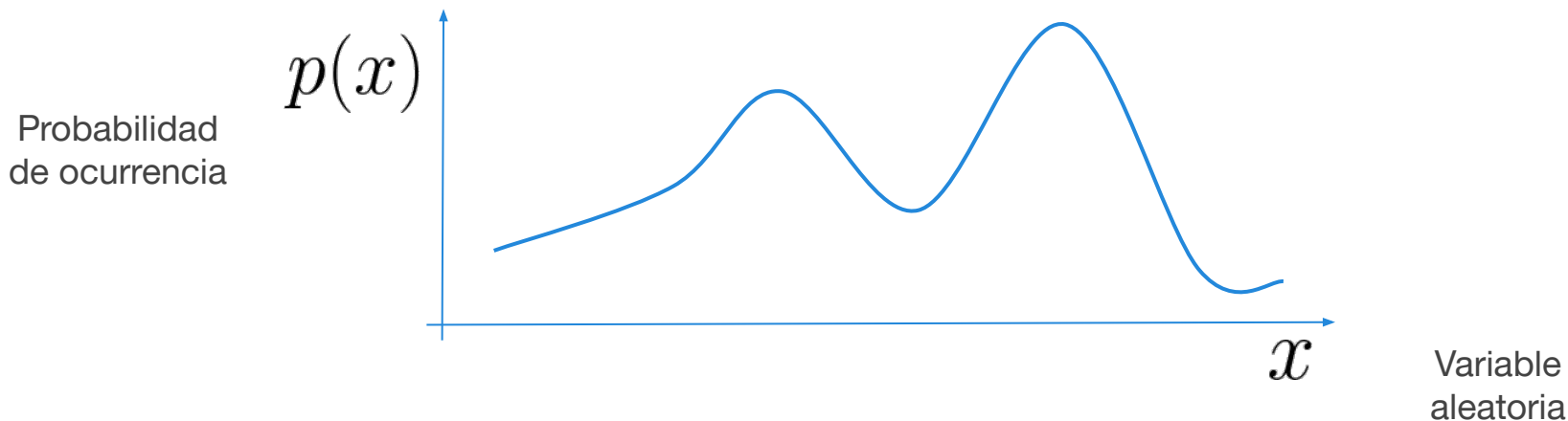
Cuando el número  $n$  de samples < número de features  $d$  la relación entre instancias y dimensiones es menor a uno y eso implica mayor dificultad para poder explicar y describir la distribución de las muestras/instancias.

# Distribuciones de Probabilidad y variables aleatorias

Primer caso: univariadas

# Distribución de probabilidad

La distribución de probabilidad es la **función** que asigna probabilidades de ocurrencia a distintos estados posibles de un experimento [1]. Es la **descripción** de un fenómeno **aleatorio** en términos de un espacio de muestreos y probabilidades de eventos.



# Funciones de densidad de probabilidad

Función de densidad de probabilidad discreta (izq) y continua (der).

$$\sum_{i=1}^n p(x_i) = 1$$

$$\int_{\mathcal{X}} f(x) dx = 1$$

# Funciones acumuladas de probabilidad

Función de densidad acumulada

$$F(x) = P(X \leq x) \quad \forall x \in \mathbb{R}$$

Función de densidad acumulada **discreta**

$$F(x) = \sum_{x_k \leq x} P(x_k)$$

Función de densidad acumulada **continua**

$$F(b) = P(x \leq b) = \int_{-\infty}^b f(x) dx$$



# Esperanza y Varianza de una VA

**Valor Esperado** de una variable aleatoria discreta (izq) y continua (der):

$$E(X) = \sum xP(x)$$

$$E(X) = \int_{-\infty}^{+\infty} x \cdot f(x) dx$$

**Varianza:** Se utilizan para describir la variabilidad de una variable aleatoria en referencia a su esperanza.

$$var(X) = E(X - E(x))^2$$

# Función de probabilidad empírica

$$P_{\text{teorica}}(x = a) = f(x = a)$$

$$P_{\text{empirica}}(x = a) = \frac{\sum_{i=1}^n \delta(x_i = a)}{n}$$

# Ejemplo proba empírica

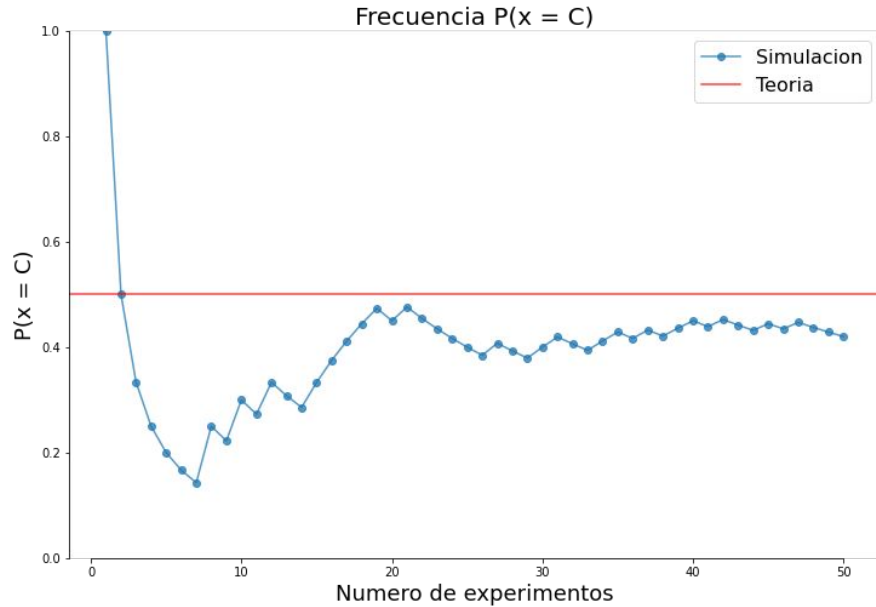
Supongamos que tenemos una moneda con 2 caras perfectamente balanceada donde la probabilidad teórica de obtener una cara es  $P(x = C) = 0.5$ .

Vamos a estimar en Python la probabilidad teórica con la probabilidad empírica mediante experimentos. En este caso  $n = 20$ .

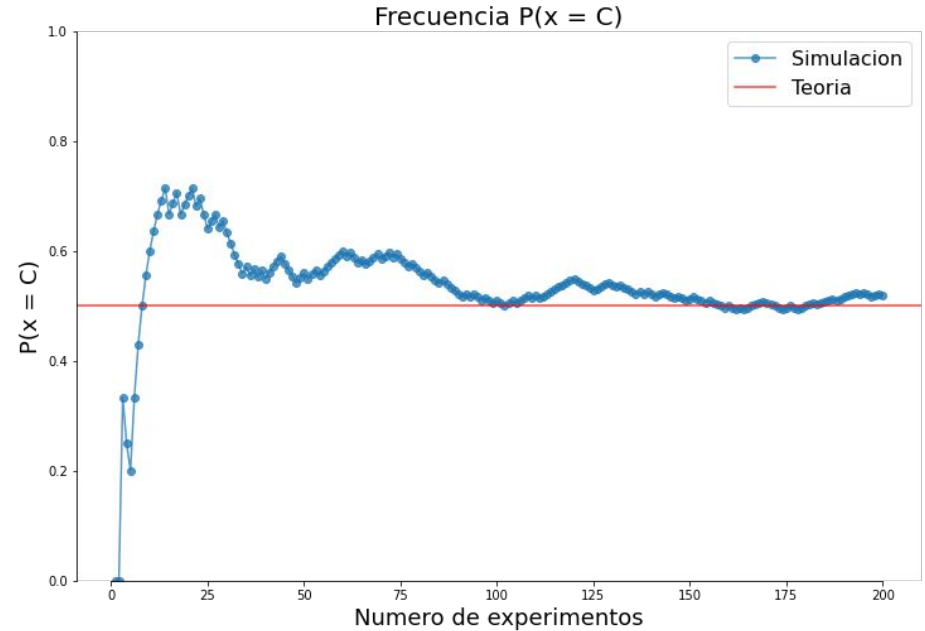
```
Experimentos: C C S C S S C S C C C S S S S S S S S C  
Numero de caras: 8  
 $P(x=C) = 0.4$  (Numero de caras/Total experimentos)
```

Luego de 20 iteraciones/sampleos del fenómeno a estudiar (moneda) observamos que la probabilidad empírica  $P(x = C) = 0.4$ . Que sucedió?

# Ejemplo prueba empírica



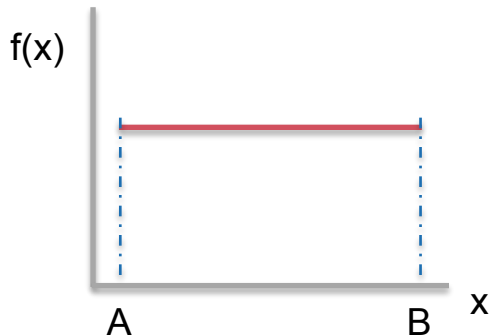
Prueba empírica luego de 50 iteraciones



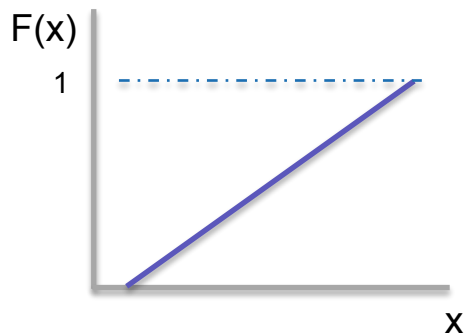
Prueba empírica luego de 200 iteraciones

# distribución uniforme

Distribución de densidad de probabilidad.



Distribución de probabilidad acumulada.



Rango de valores posibles.

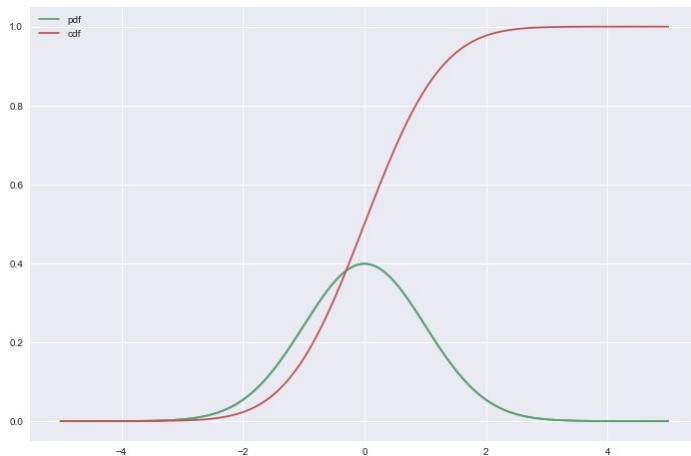
$$f(x) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq x \leq b \\ 0 & \text{else} \end{cases}$$

$$F(x) = \begin{cases} 0 & \text{if } x < a \\ \frac{x-a}{b-a} & \text{if } a \leq x \leq b \\ 1 & \text{if } x > b \end{cases}$$

La distribución de probabilidad uniforme asigna la misma probabilidad de ocurrencia a cada valor dentro del rango que puede generar una variable aleatoria.

# Distribución gaussiana - normal

Distribución de densidad (verde) y acumulada (roja) de probabilidad.



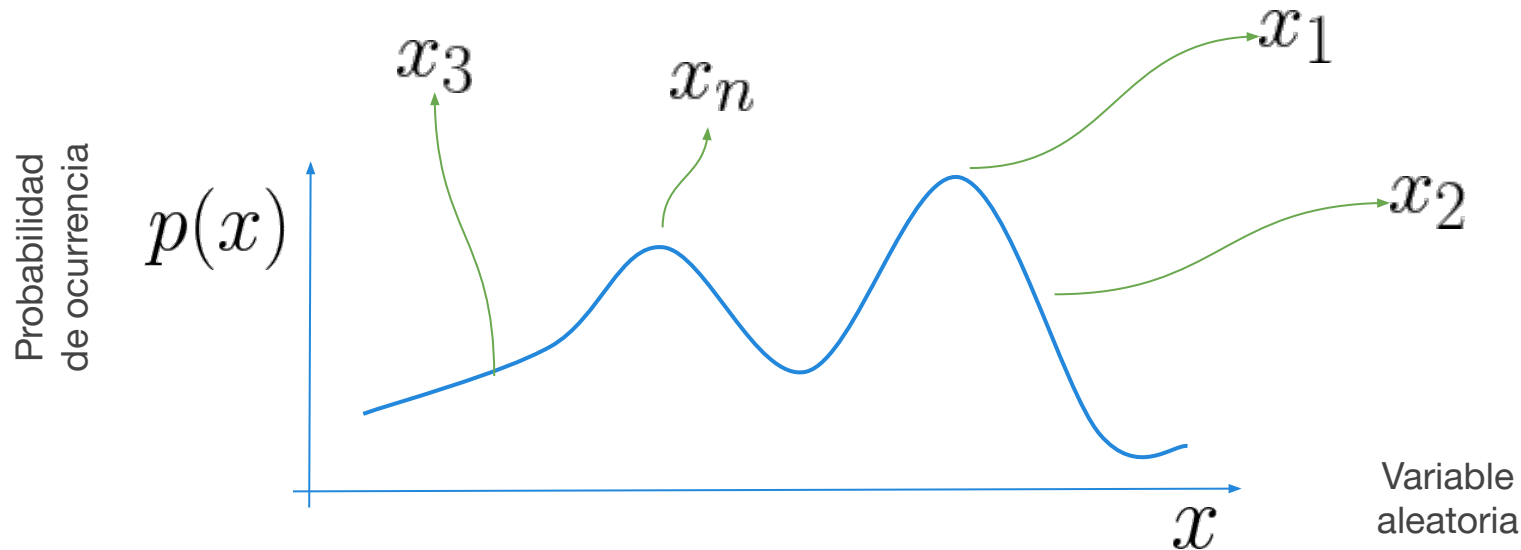
Rango de valores posibles de la VA.

Distribución simétrica de VA continua. El parámetro  $\mu$  define la esperanza y el sigma el desvío standard.

Suele utilizarse para modelar procesos reales en ciencias naturales, sociales, etc.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$$

# Muestreo desde una función de densidad de probabilidad



Suponiendo que **conocemos** la función de densidad de probabilidad de una variable aleatoria, vamos a **muestrear** multiples veces dicha funcion y obtener distintos valores de la variable aleatoria a observar. En el caso contrario si solo tenemos los datos y no conocemos la funcion de densidad que los genero se abordaran estrategias de maxima verosimilitud o metodos de estimacion no parametrica de la densidad [2]

[1] [https://en.wikipedia.org/wiki/Maximum\\_likelihood\\_estimation](https://en.wikipedia.org/wiki/Maximum_likelihood_estimation)

[2] [https://en.wikipedia.org/wiki/Kernel\\_density\\_estimation](https://en.wikipedia.org/wiki/Kernel_density_estimation) .

# Distribuciones de Probabilidad y variables aleatorias

Segundo caso: multivariadas



# Variables aleatorias multivariadas

$$p(x = \text{cancer}) = f(???) = f(x_1, x_2, \dots, x_n)$$

$$p(x = + \text{covid}) = f(???) = f(x_1, x_2, \dots, x_n)$$

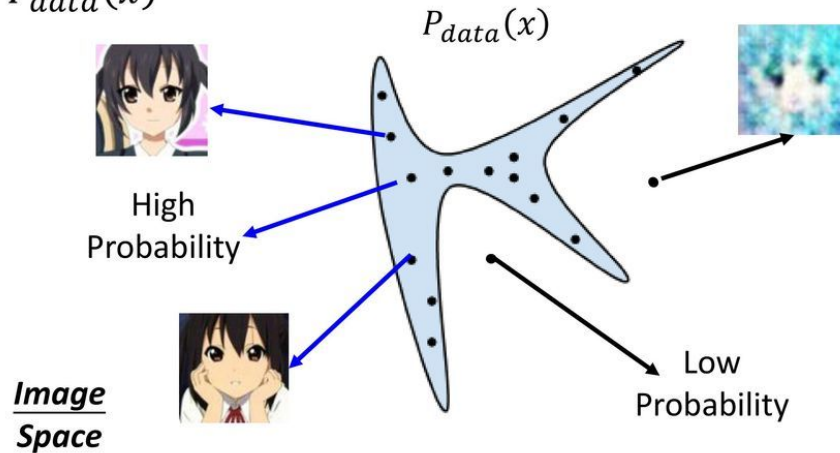
$$p(x = \text{cruzar a un conocido}) = f(x_1, x_2, \dots, x_n)$$

En los problemas reales existen variables aleatorias multi-variadas con distribuciones de densidad de probabilidad complejas.

# Variables aleatorias multivariadas


## Basic Idea of GAN

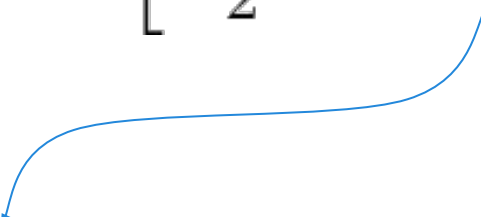
- The data we want to generate has a distribution  $P_{data}(x)$




# Distribución gaussiana bivariada

$$p(X) \sim (2\pi)^{-d/2} |\Sigma|^{-1/2} \exp \left[ -\frac{1}{2} (X - M)^t \Sigma^{-1} (X - M) \right]$$

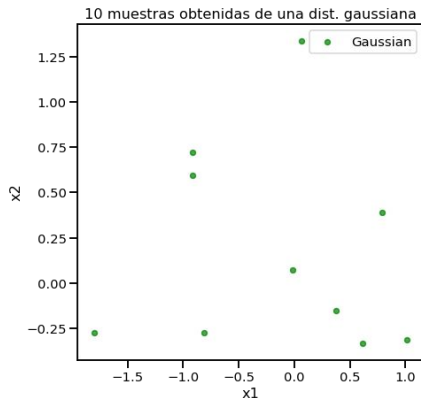

$$X = [x_1, x_2]$$


$$M = [\mu_1, \mu_2]$$

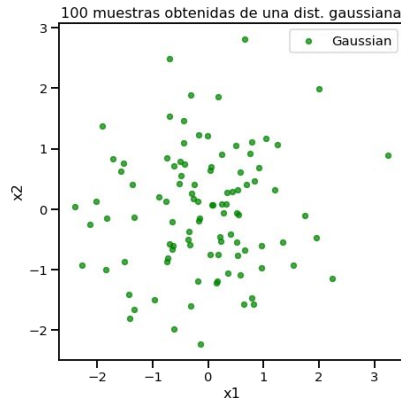

$$\Sigma = \begin{bmatrix} \sigma_{00}^2 & \sigma_{01}^2 \\ \sigma_{10}^2 & \sigma_{11}^2 \end{bmatrix}$$

# Muestreando una PDF gaussian bi-variada

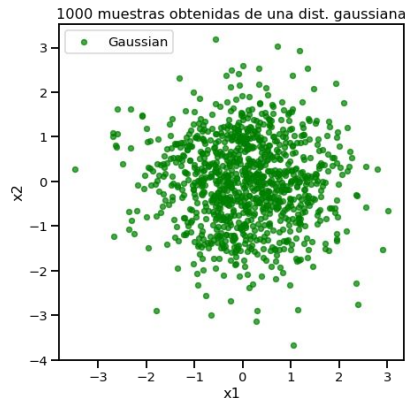
$n = 10$



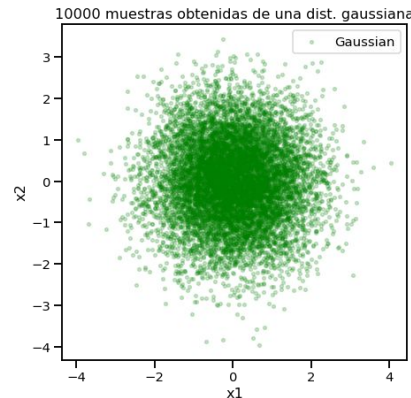
$n = 100$



$n = 1000$

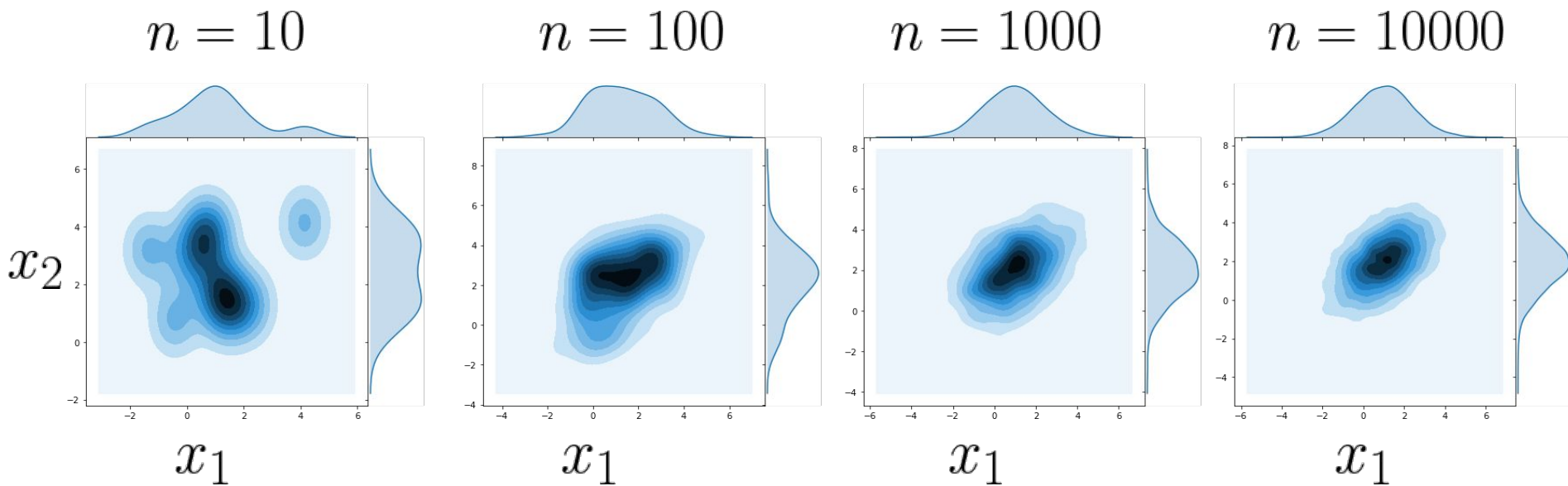


$n = 10000$



Scatterplot para visualizar las muestras/instancias obtenidas de una distribución de probabilidad gaussiana bivariada ( $d=2$ ) para distintos valores de  $n$ .

# Muestreando una PDF gaussian bi-variada



Density map realizado a partir de muestras obtenidas de una distribución de probabilidad gaussiana bivariada ( $d = 2$ ) con distintos valores de  $n$ .

# Aprender de datos

# Aprender de datos

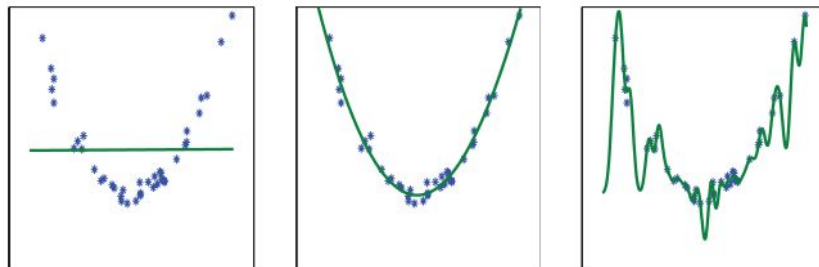
Nuestro problema tendrá una característica particular: no conocemos la distribución de densidad de probabilidad (PDF) que genera nuestros datos. Los datos difícilmente coincidan con una distribución normal o una exponencial. Lo más probable es que pertenezcan a una distribución compleja en alta dimensión.

- Una opción es aproximarlos a distribuciones que ya conozcamos (metodo de maxima verosimilitud) aunque no necesariamente esa es la mejor opción.
- La alternativa de **Aprendizaje Estadístico / Machine Learning** es aprender la distribución de los datos y/o su estructura en el espacio donde habitan con el fin de facilitar una tarea “aguas abajo” (downstream task).

# Aprender de datos

Caso aprendizaje supervisado:

- Partiendo de un set de datos  $S$  aprenderemos una función “ $f(x)$ ” desconocida y será el estimador que utilizaremos.
- Nunca llegaremos a una “ $f(x)$ ” ideal que explique a la perfección nuestros datos, por ende tendremos cierto grado de error. La función  $f(x)$  supone una distribución de probabilidad “ $p(x)$ ” que es incierta.
- Vamos a querer aprender una función que **generalice** bien para futuros datos nunca vistos. Es decir, una vez que encontramos el patrón en los datos disponibles de *entrenamiento*, esperamos que la  $f(x)$  siga encontrando los mismos patrones para datos futuros nunca vistos.





# Tipos de Aprendizaje Automático

## **Aprendizaje:**

- **Supervisado**
- **No Supervisado**
- **Semi-supervisado**
- **Por refuerzo**

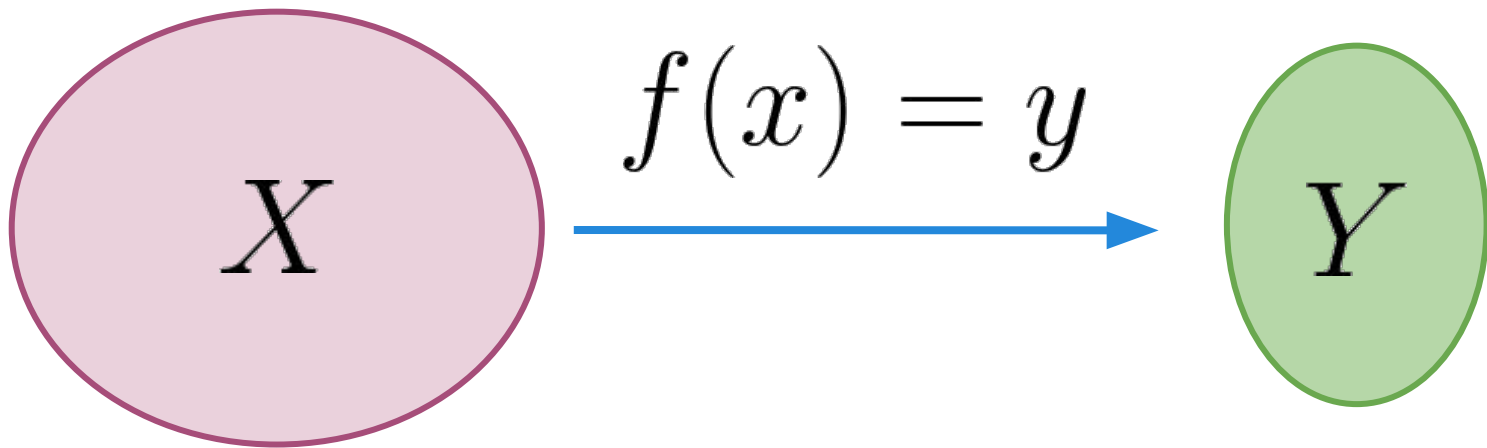
# Aprendizaje Supervisado

$$S = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$$

Suponemos un dataset con observaciones/samples  $S$ , donde  $X_i$  es un vector de features e  $Y_i$  es una label (etiqueta) asociada a cada observación  $X_i$ .

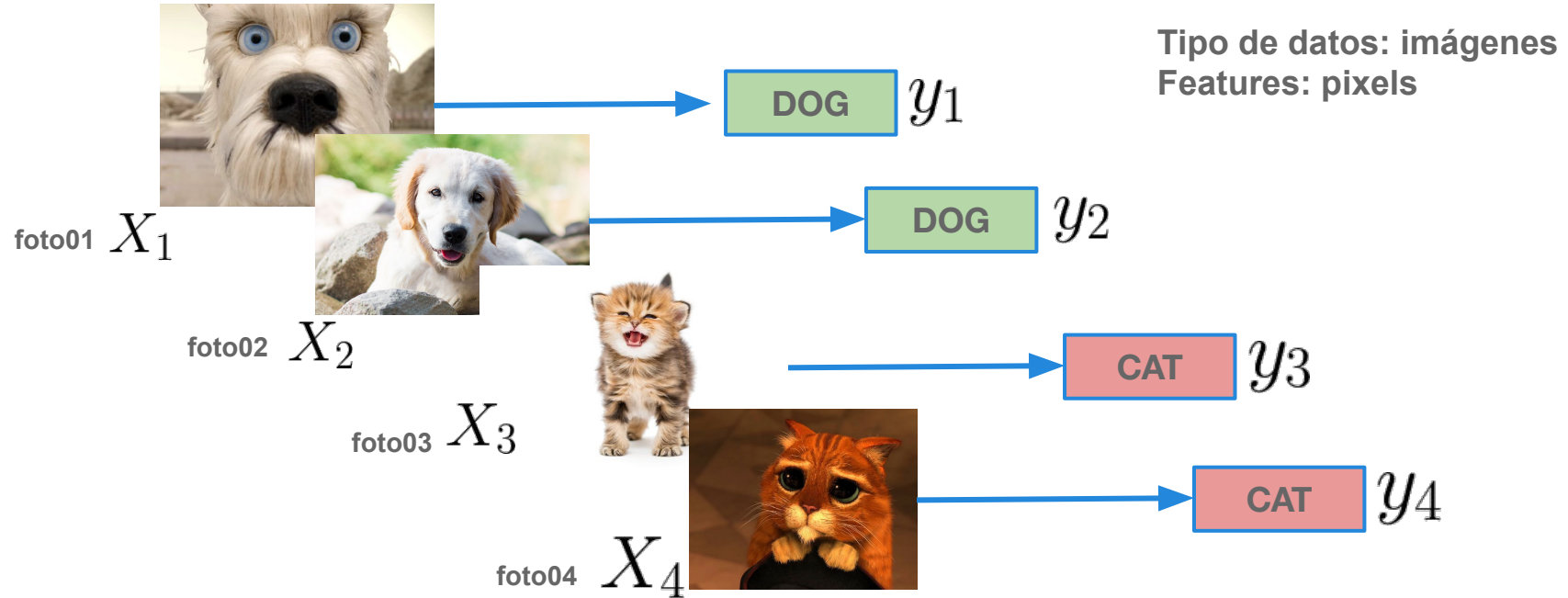
Solemos denominar a cada observación  $X_i$  como “sample” y a cada etiqueta  $Y_i$  como “label”.

# Aprendizaje Supervisado



Suponemos que la variable  $Y$  es dependiente de  $X$ . Esto quiere decir que  $Y$  está condicionada y es consecuencia de  $X$ . **Lo que no conocemos es la función  $y = f(x)$**  y es  $f(x)$  lo que queremos aprender desde los datos.

# Tipos de Aprendizaje: Supervisado



Cada instancia (sample) viene acompañada de una etiqueta (label).

## Tipos de Aprendizaje: Supervisado

**Tipo de datos: ADN**  
**Features: mutaciones**

A diagram showing a blue arrow pointing into a light green rectangular block labeled "CONTROL SANO". To the right of the block is the variable  $y_1$ .

paciente01  $X_1$

[illegible]

FORMAT	NAG0001	NAG0002	NAG0003
GT:QQ:DP:HQ	010:48:1:51,51	110:48:8:51,51	1/0:43:5:..
GT:QQ:DP:HQ	010:49:3:58,50	011:3:5:65,3	0/0:41:3:5
GT:QQ:DP:HQ	112:21:6:23,27	211:2:0:18,2	2/2:35:4
GT:QQ:DP:HQ	010:54:7:56,60	010:48:4:51,51	0/0:61:2
GT:QQ:DP	0/1:35:4	0/2:17:2	1/1:40:3

paciente02  $X_2$

[illegible]

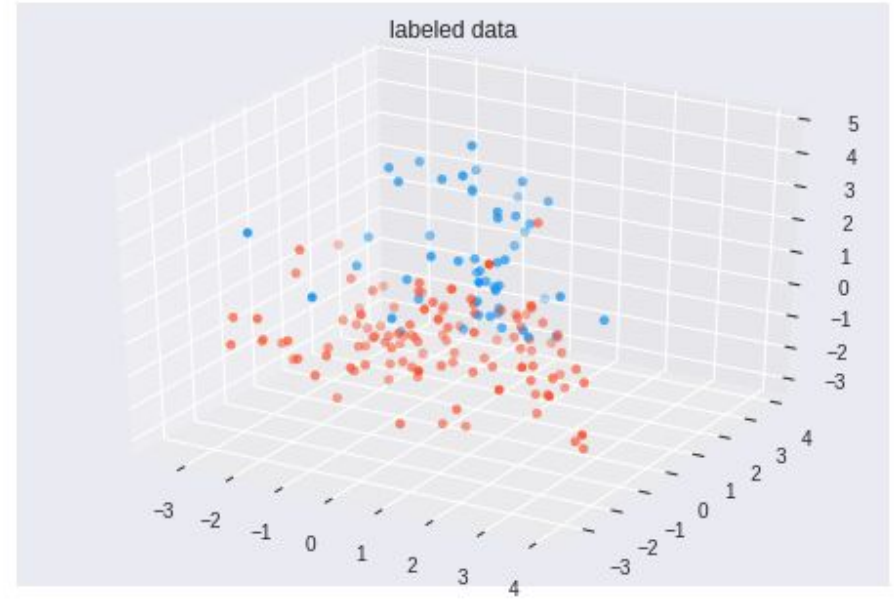
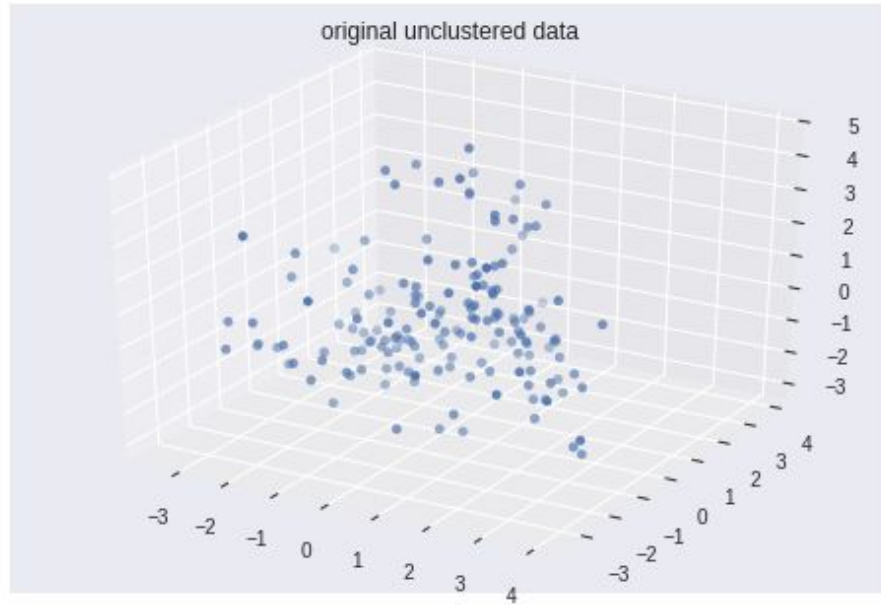
FORMAT	NA00001	NA00002	NA00003
GT:GQ:DP:HQ	010:48:1:51,51	110:48:8:51,51	1/1:43:15:...
GT:GQ:DP:HQ	010:49:3:58,50	011:3:65:3	0/0:41:3
GT:GQ:DP:HQ	112:21:6:23,27	211:21:0:18,2	2/2:36:14
GT:GQ:DP:HQ	010:54:7:56,60	101:04:4:51,51	0/0:61:2
GT:GQ:DP	0/1:35:4	0/2:17:2	1/1:40:3

paciente03  $X_3$

**CANCER**  $y_2$

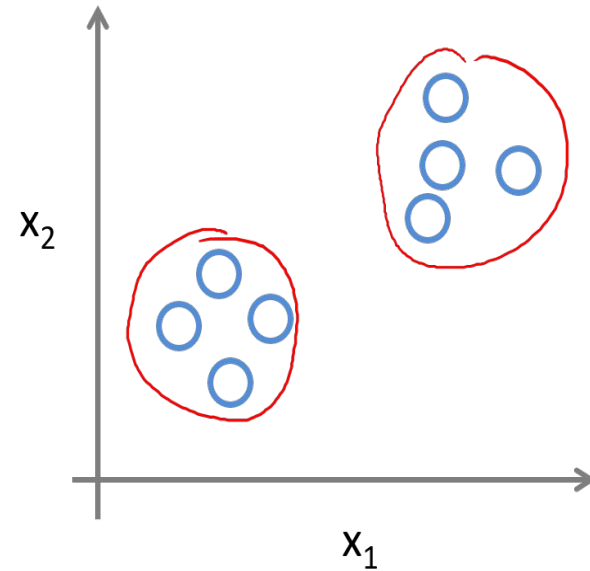
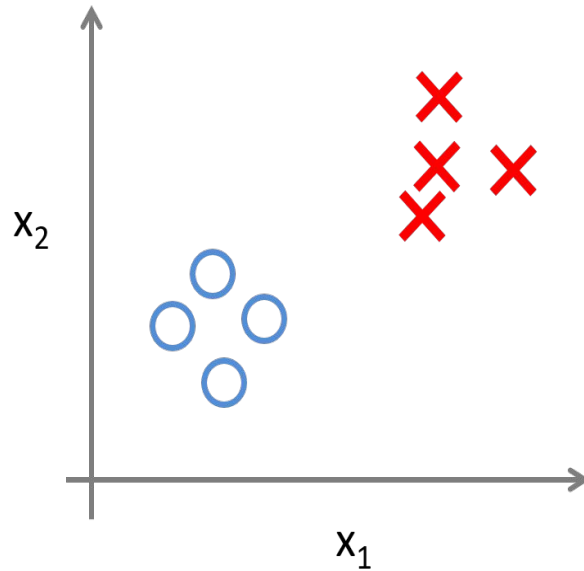
CANCER  $y_3$

# Tipos de Aprendizaje: No supervisado



Cada instancia (sample) **no posee** etiqueta (izq). Los modelos a aplicar en estos casos buscan encontrar estructuras o grupos implícitas en los datos (ej. clusters)

# Supervisado vs No supervisado



Izquierda: Datos etiquetados. Derecha: Datos sin etiquetar estructurados en clusters.

# Tipos de Datos

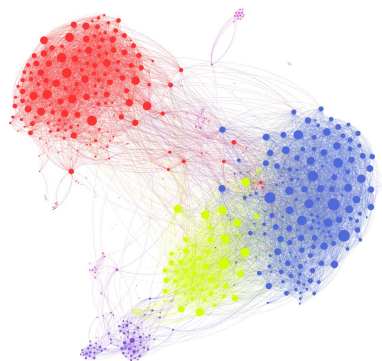
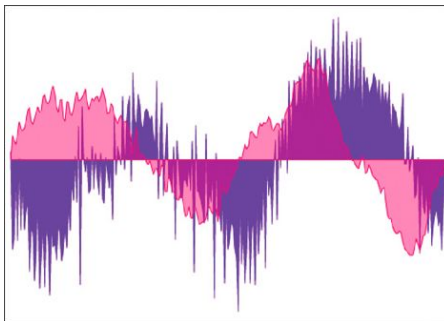
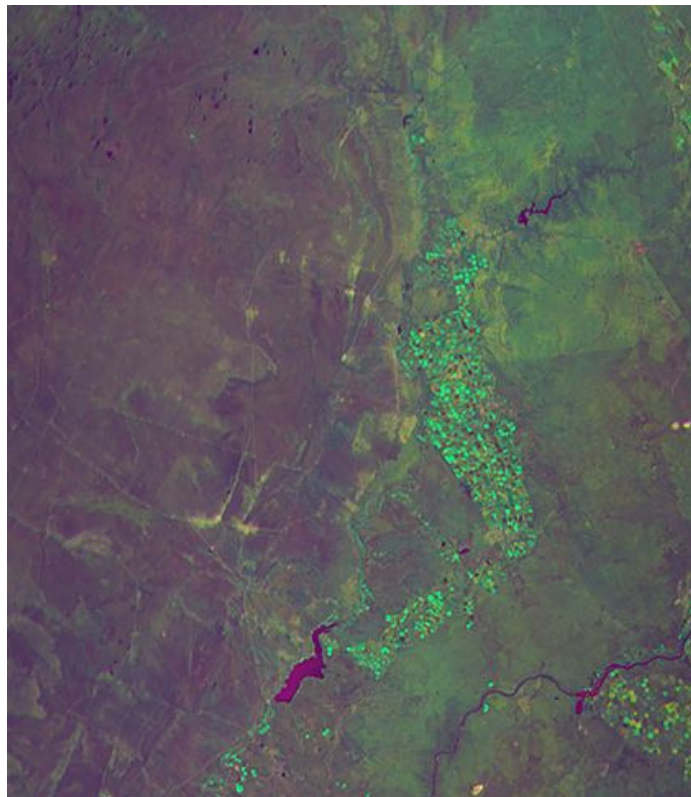
- Estructurados / tabulares
- Imágenes
- Grafos-redes
- Lenguaje Natural (texto)
- Audio / Series de tiempo



# Formato de los datos

- .CSV
  - .xlsx
    - .txt
      - .tsv
        - .jpeg
          - SQL query

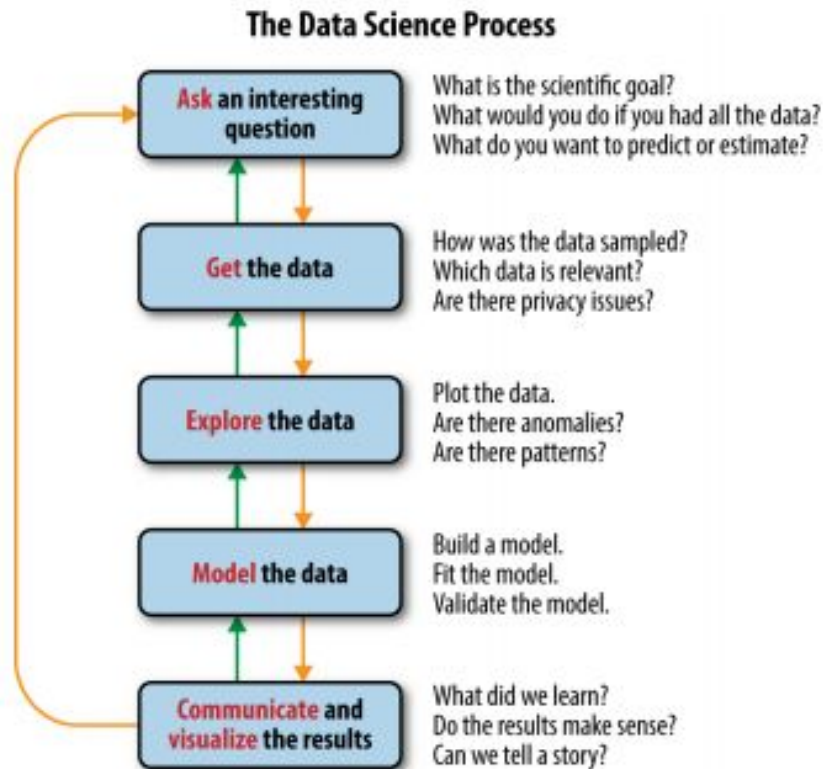
# Tipos de Datos



text  
as  
data  
researchers  
quantitative  
analysis  
textual  
methods  
sciences  
scaling  
approach  
develop  
users  
training  
techniques  
models  
social  
research  
software  
book-length  
Finally  
analyzing  
indicators

	A	B	C	D	E
1	Record Id	Date	Month	High	Low
2	4450	1	6	31	22
3	4451	15	6	35	21
4	4452	30	6	28	21
5	4453	1	7	34	25
6	4454	15	7	33	26
7	4455	30	7	30	21
8	4456	1	8	32	20
9	4457	15	8	36	23
10	4458	30	8	36	24
11	4459	1	9	37	20
12	4460	15	9	40	25
13	4461	30	9	37	22
14	4462	1	10	35	23
15	4463	15	10	32	25
16	4464	30	10	35	21
17	4465	1	11	32	24
18	4466	15	11	39	24
19	4467	30	11	33	23
20	4468	1	12	33	21
21	4469	15	12	34	24
22	4470	30	12	38	24

# Data Science Workflow



\*Development Workflows for Data Scientists

# 1) Data Science Workflow: get the data

The Kaggle logo, featuring the word "kaggle" in a light blue, lowercase, sans-serif font.The Datos Argentina logo, featuring the text "Datos Argentina" in white on a blue background with a circular pattern. Below the text, it says "Portal de datos abiertos del Gobierno de la República Argentina. Acá encontrarás información pública, herramientas y recursos para desarrollar aplicaciones, visualizaciones y más."

**Buenos Aires Data**



Iniciativa de Datos Públicos y Transparencia de la Ciudad Autónoma de Buenos Aires.

Durante el curso trataremos de utilizar repositorios de datos abiertos, principalmente aquellos de la Ciudad de Buenos Aires, Provincia de Buenos Aires o Nación.

## 2) Data Science Workflow: Explore

### Pre-processing

- Clean samples with NaNs
- Transform features
- Normalize data

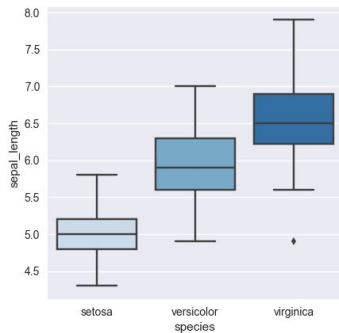
### Exploratory Data Analysis

- Realizar estadísticas descriptivas
- Quitar outliers estadísticos
- Visualizar con Bar-plots, Box-plots, Scatter-plots, Count-plots

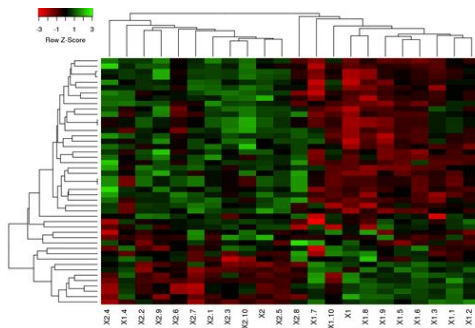
## 2) Exploratory Data Analysis (EDA)

- Importar datos.
- Revisar si hay NaNs o valores faltantes.
- Filtrar los datos de interés.
- Transformar los datos (ej, tabla pivote)
- Computar estadísticas descriptivas (media, dev. std, percentiles)
- Medir correlación entre variables de interés
- Visualizar:

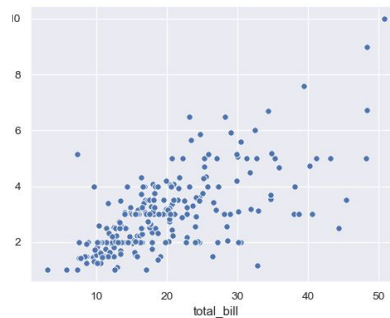
Boxplot



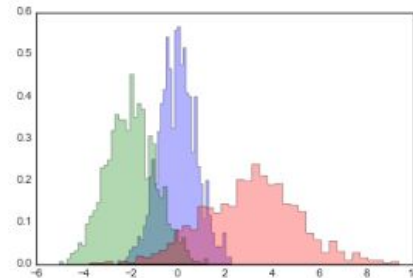
Heatmap



Scatter plot



Histogram



### 3) Data Science: statistical learning

**clasificación**

**regresión**

**Reducción de dimensionalidad**

**Detección de anomalías**

**clustering**

## Poster explicando el proyecto

# Reporte técnico del proyecto

## Exposición de posters



## Repositorio en github del proyecto

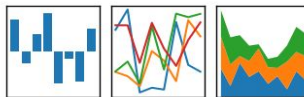
[illegible]



# Tecnologías que utilizaremos



## Librerías:



# Numpy vs Pandas



## Calculo con matrices

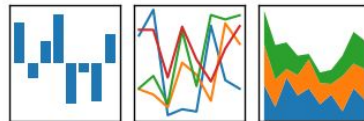
- No admite nombres en cols
- No admite nombres en filas
- Diversidad en aplicaciones de cálculo
- Útil para lidiar con álgebra y operaciones matriciales

## Atajos de Numpy acá:

[https://s3.amazonaws.com/assets.datacamp.com/blog\\_assets/Numpy\\_Python\\_Cheat\\_Sheet.pdf](https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Numpy_Python_Cheat_Sheet.pdf)

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



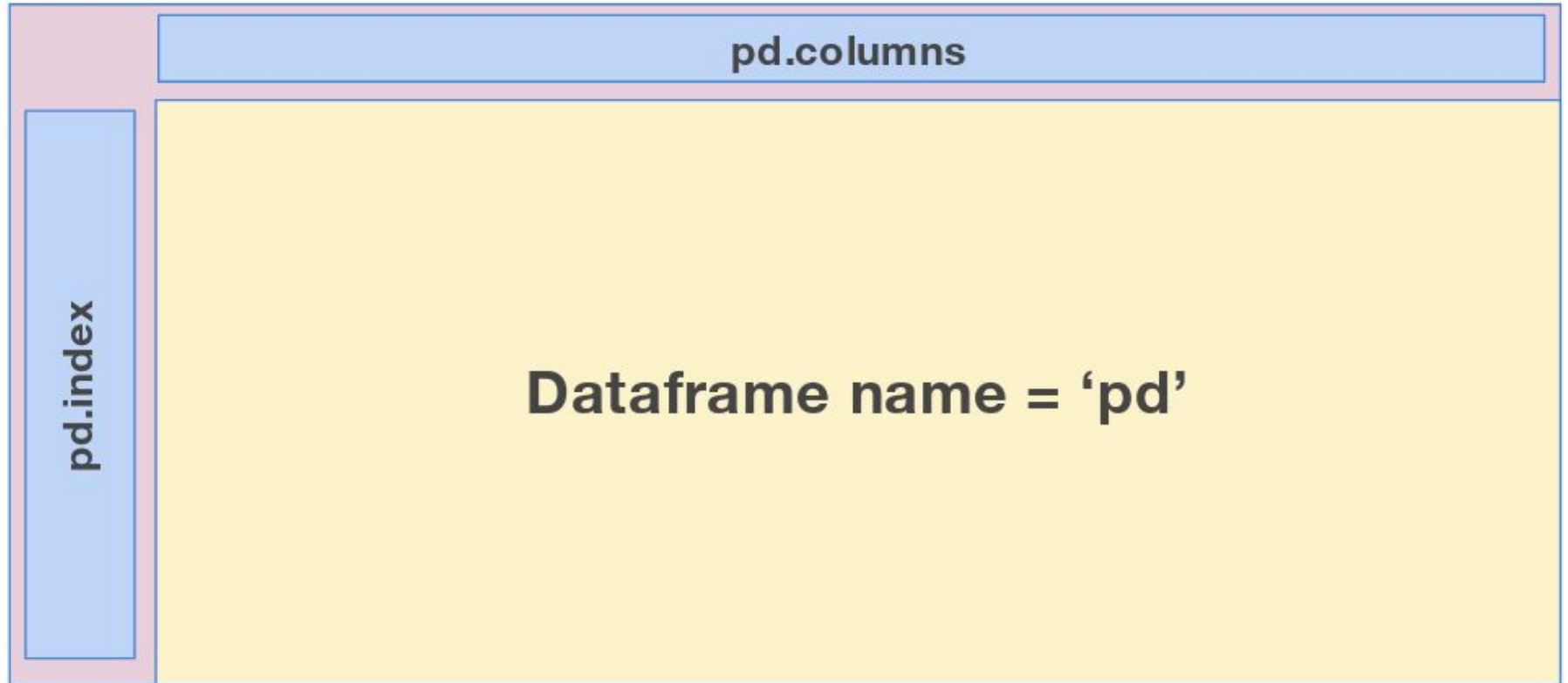
## Gestor de datasets en Dataframes (DFs)

- Admite nombre de columnas
- Admite nombre de filas
- Diversas funciones sobre DFs.
- Útil para lidiar con datos, limpiar, pre procesar.

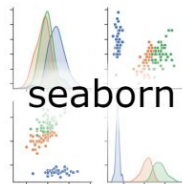
## Atajos de Pandas acá:

[https://pandas.pydata.org/Pandas\\_Cheat\\_Sheet.pdf](https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf)

# Pandas Dataframe



# Visualización




## Librerías de visualización de datos

- Matplotlib es la principal librería de visualización en Python.
- Seaborn corre sobre matplotlib y posee algunas mejoras de estética.
- Tipos de gráficos a realizar:
  - Countplot (graficos de barra)
  - Heatmap (mapas de calor)
  - Boxplot (diagrama de cajas y bigote)
  - Series de tiempo
  - Scatter plot (diagrama de puntos)
  - Distplot (distribuciones y densidades)

Atajos de Matplotlib acá:

[https://s3.amazonaws.com/assets.datacamp.com/blog\\_assets/Python\\_Matplotlib\\_Cheat\\_Sheet.pdf](https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Python_Matplotlib_Cheat_Sheet.pdf)

# Tipos de variables en Python



```
a = 3
b = 0.4
c = False
d = "Quiero analizar datos"
e = [2,3,4,5,6]
f = [[2,3,4],[1,0,40]]
```

**a = Integer**

**b = Float**

**c = Boolean**

**d = String**

**e = Numpy Array (1,5)**

**f = Numpy Array (2,3)**

# A agarrar la PyLA



# Exploratory Data Analysis

