Analítica de datos y extracción del conocimiento

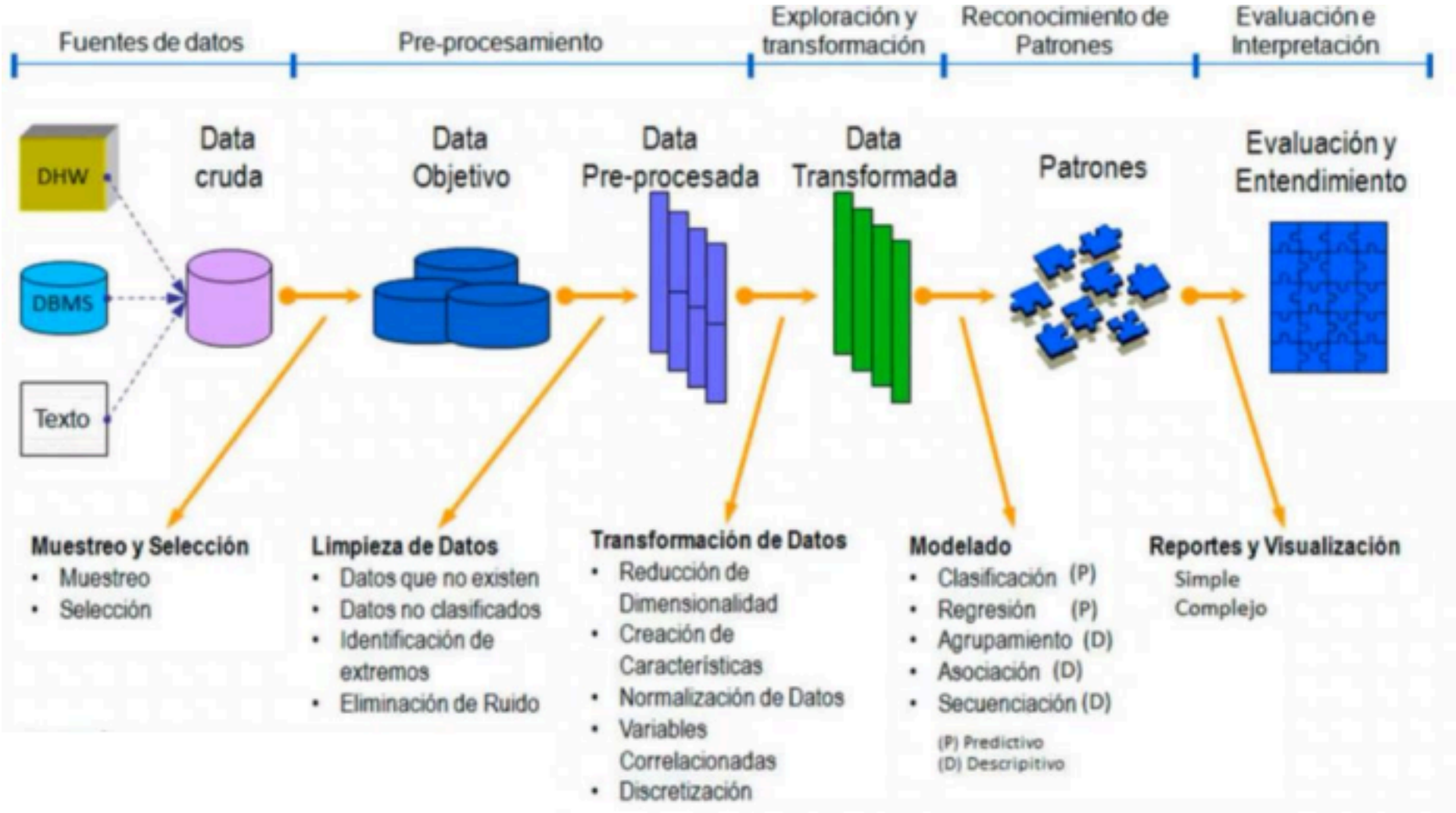
- Machine Learning

- Data Visualization
- Databases

- Deep Learning
- Computer Vision
- Data Signals Processing
- Data Cleaning
- IA
- Natural Language
- Geospatial Analysis

Nos ayuda a descubrir información útil con los datos

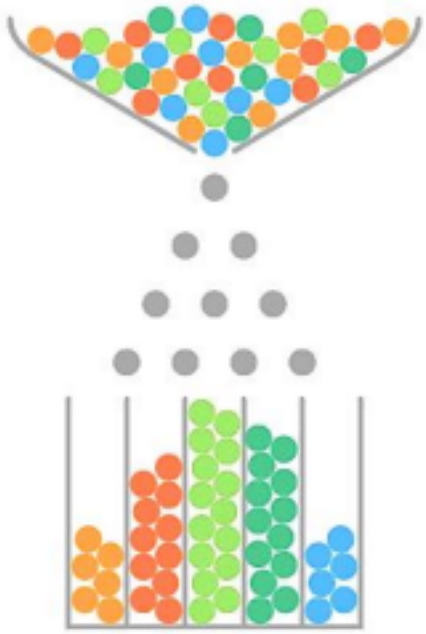
Reconocimiento de patrones



Estandarización de datos

1. Data cleaning: la limpieza de datos elimina ruido y resuelve las inconsistencias en los datos.

- Imputación de datos faltantes • Variables categóricas
- Dummy Encoding



- Escalamiento de datos

2. Data integration: con la Integración de datos se migran datos de varias fuentes a una fuente coherente como un Data Warehouse.

3. Data transformation: la transformación de datos sirve para normalizar datos de cualquier tipo.

4. Data reduction: la reducción de datos reduce el tamaño de los datos agregándolos.

PROCESO ETL

PROCESO ETL

Ejemplo preprocesamiento de

País	Edad	Salario	Compró
Francia	44	72000	No
España	27	48000	Yes
Alemania	30	54000	No
España	38	61000	No
Alemania	40		Yes
Francia	35	58000	Yes
España		52000	No
Francia	48	79000	Yes
Alemania	50	83000	No
Francia	37	67000	Yes

datos

Imputación de datos faltantes media - mediana

País	Edad	Salario	Compró
Francia	44	72000	No
España	27	48000	Yes
Alemania	30	54000	No
España	38	61000	No
Alemania	40	**63777.77**	Yes
Francia	35	58000	Yes
España	**38.77**	52000	No
Francia	48	79000	Yes
Alemania	50	83000	No
Francia	37	67000	Yes

Variables

Variables categóricas

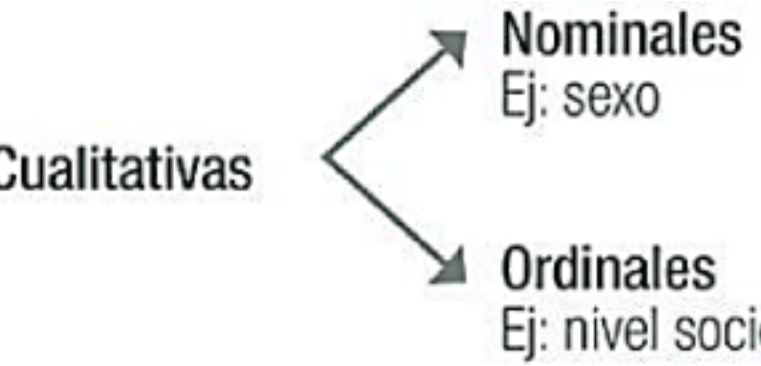
Las variables categóricas también se denominan variables cualitativas o variables de atributos. Los valores de una variable categórica son categorías o grupos mutuamente excluyentes. Los datos categóricos pueden tener o no tener un orden lógico.

Variables cuantitativas

Los valores de una variable cuantitativa son números que suelen representar un contro o una medición.

Asunto del análisis	Variables categóricas posibles	Variables cuantitativas posibles
Ventas de electrodomésticos	Tipo de electrodoméstico; Marca y modelo	Precio de venta
Pintura para carrocería de automóviles	Desperfectos de la pintura (descarapelada, raya, mancha, otros); Colores de la pintura	Temperatura del metal que se va a pintar; Espesor de la capa de pintura

Variables, tipos de dato



A B C D

A B C D

I II III IV

1 2 3 4

I II III IV

1

2 3 4

1.5 2.6 3.4

1.5 2.6 3.4

Jerarquizar Contar

Medir

Clasificar

Variables, tipos de dato

Ejemplos de variables categóricas

Tipo de datos	Ejemplos
Numérico	•Sexo (1 = Mujer, 2 = Hombre) •Resultados de una encuesta (1 = De acuerdo, 2 = Neutral, 3 = En desacuerdo)

Texto	•Formas de pago (Efectivo o Crédito) •Configuraciones de una máquina (Bajo, Medio, Alto) •Tipos de producto (Madera, Plástico, Metal)
Fecha/hora	•Días de la semana (lunes, martes, miércoles) •Meses del año (enero, febrero, marzo)

Variables, tipos de dato

Ejemplos de variables cuantitativas

Tipo de datos	Ejemplos
Numérico	•Número de quejas de clientes •Proporción de clientes elegibles para un reembolso •Peso de llenado de una caja de cereales
Fecha/hora	•Fecha y hora en que se recibió el pago •Fecha y hora del incidente de soporte técnico

Variables categóricas

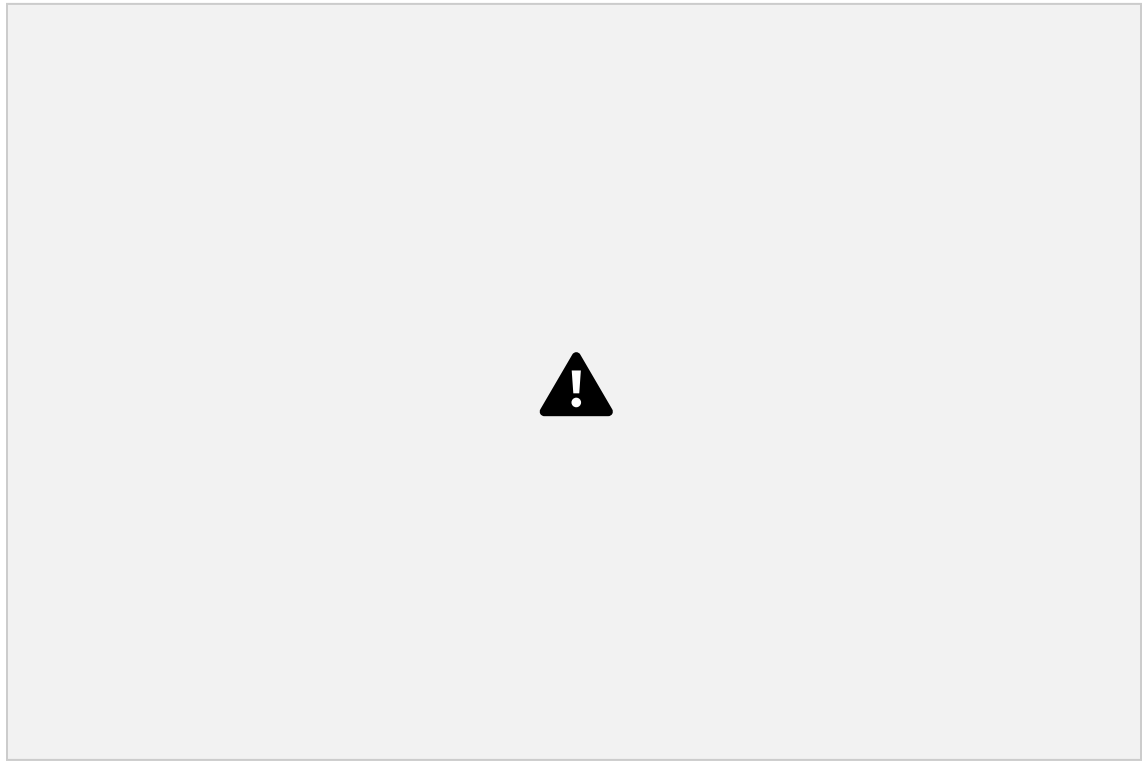
Surge un pequeño problema, parece que le asignamos mayor peso a un país que a otro

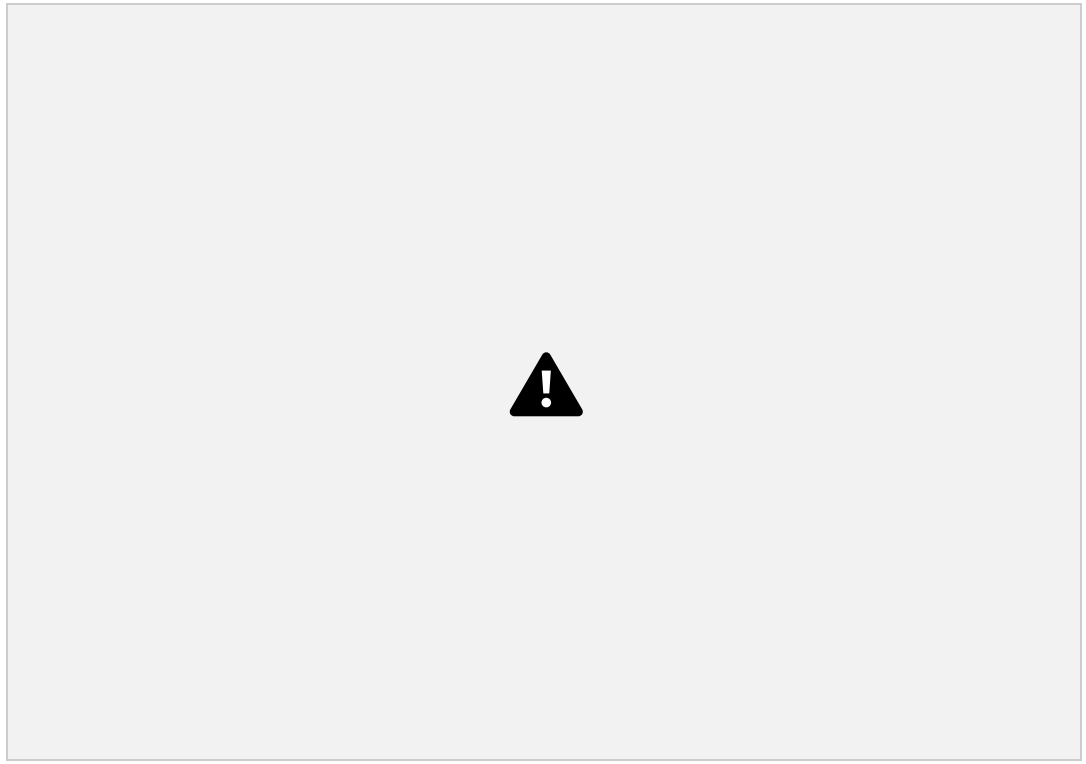
País	Edad	Salario	Compró
Francia	44	72000	No
España	27	48000	Yes
Alemania	30	54000	No
España	38	61000	No
Alemania	40	63777.77	Yes
Francia	35	58000	Yes
España	38.77	52000	No
Francia	48	79000	Yes
Alemania	50	83000	No
Francia	37	67000	Yes

País	Edad	Salario	Compró
0	44	72000	No
1	27	48000	Yes
2	30	54000	No
1	38	61000	No
2	40	63777.77	Yes
0	35	58000	Yes
1	38.77	52000	No
0	48	79000	Yes
2	50	83000	No
0	37	67000	Yes

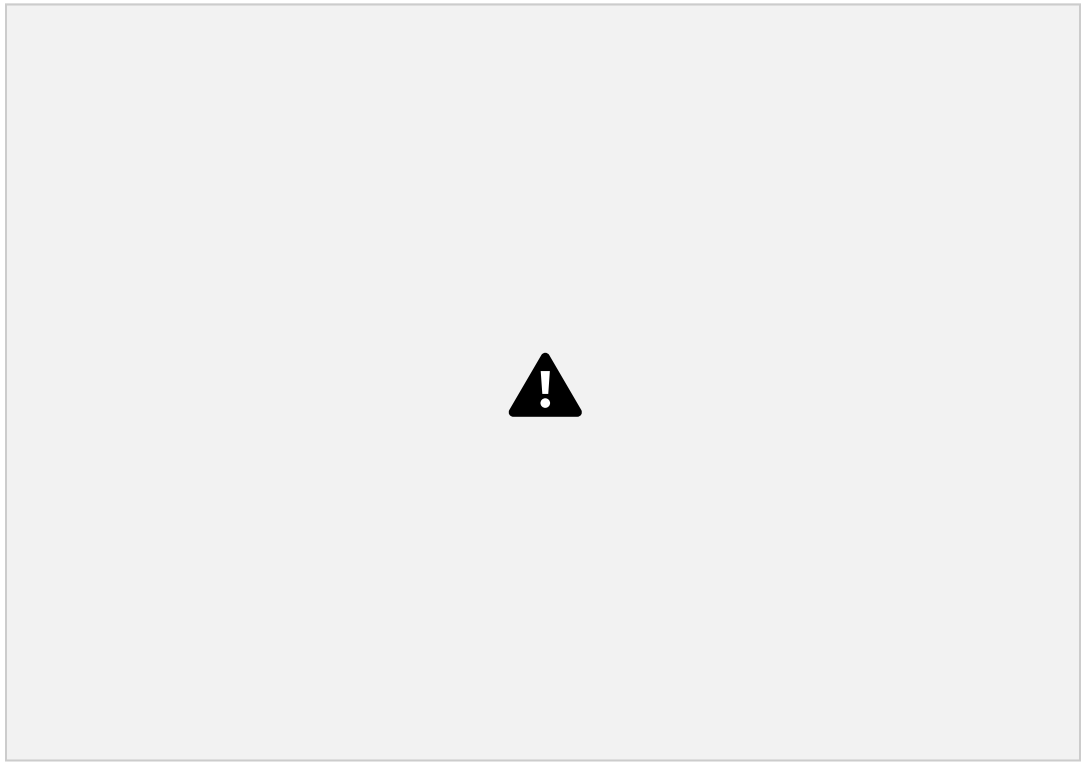
Cambio de valores cualitativos a numéricos

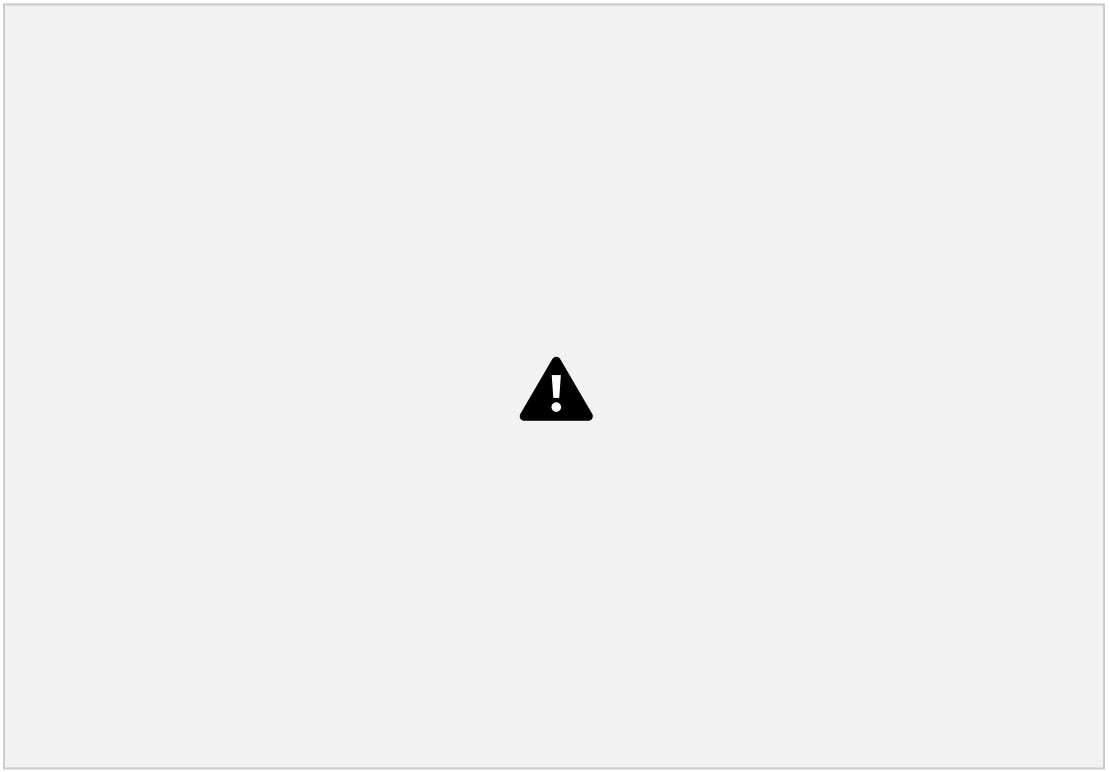
Dummy Encoding





Dummy Encoding



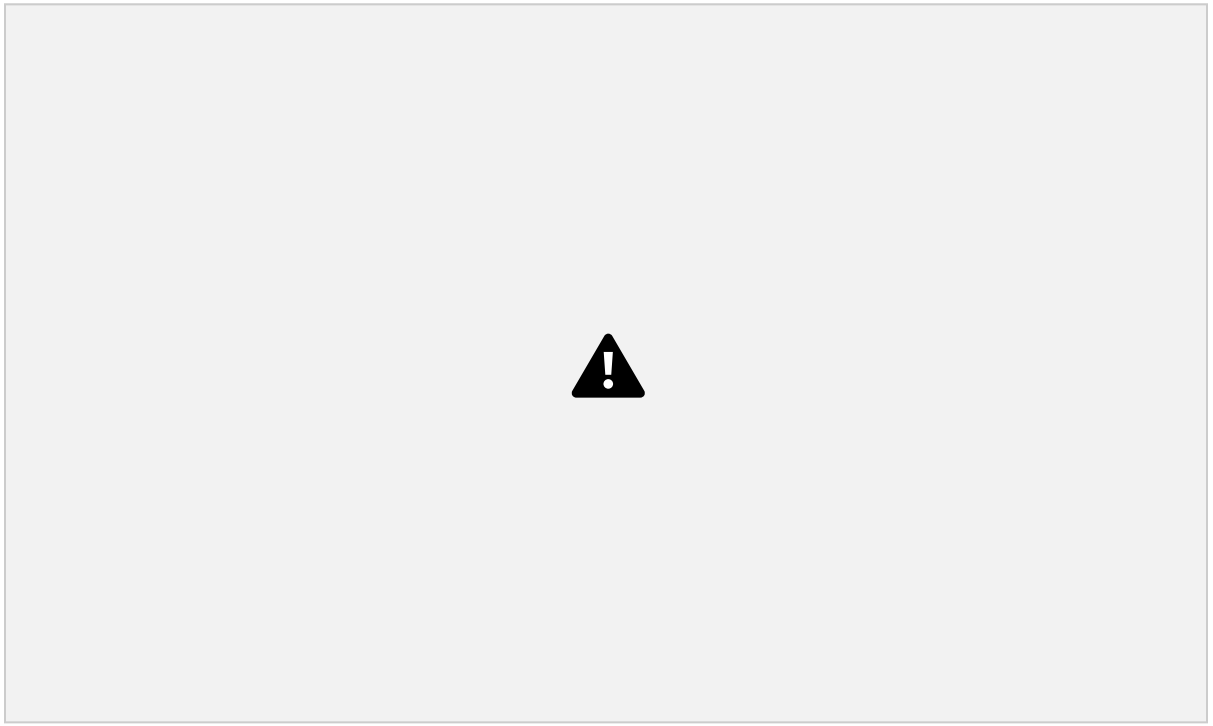


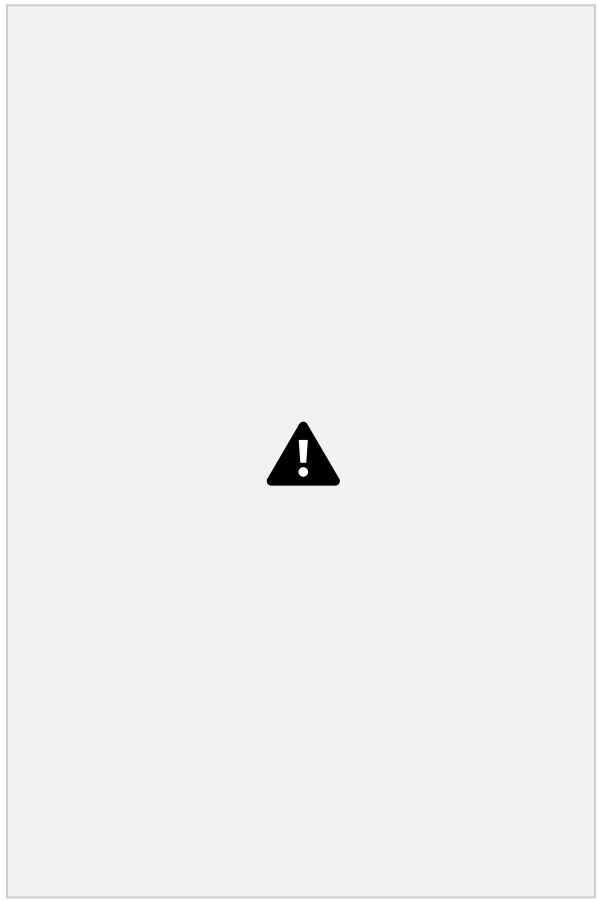
Transformación de escalas

Cambiar las escalas de los valores sin afectar la distribución y sin afectar la distancia entre los valores de la variable

Transformación de escalas

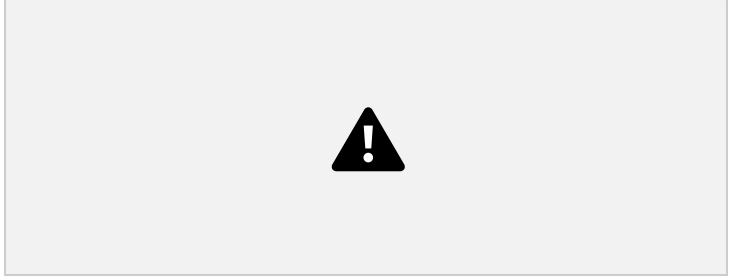
Escalamiento de características





Analítica de datos y extracción del conocimiento

Es una plataforma con recursos para aprender Machine Learning y Ciencia de datos



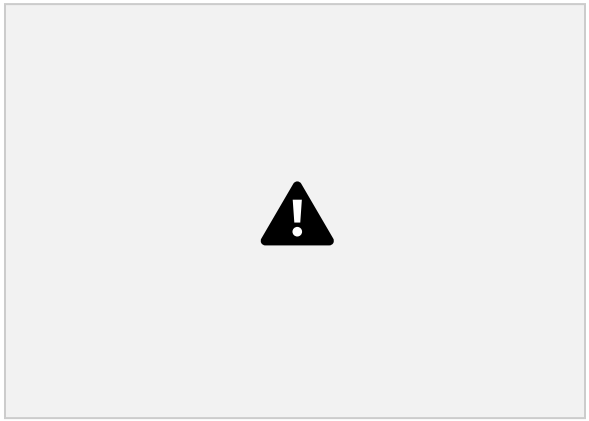
Competiciones que

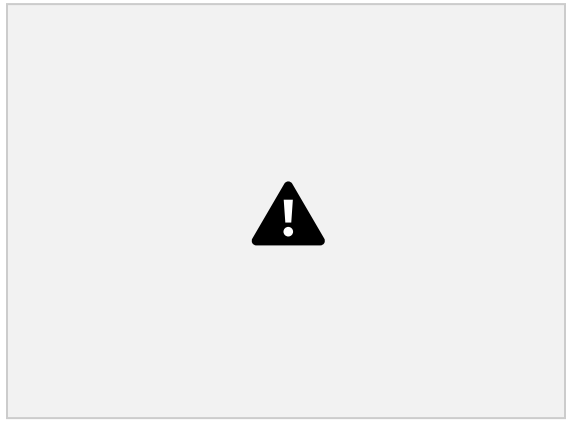
millón de dólares

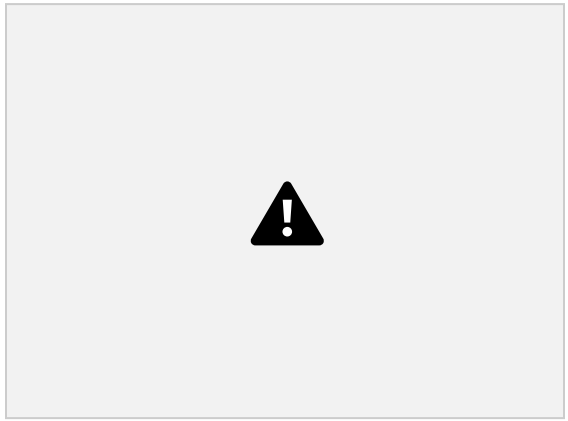
Datasets

Cursos

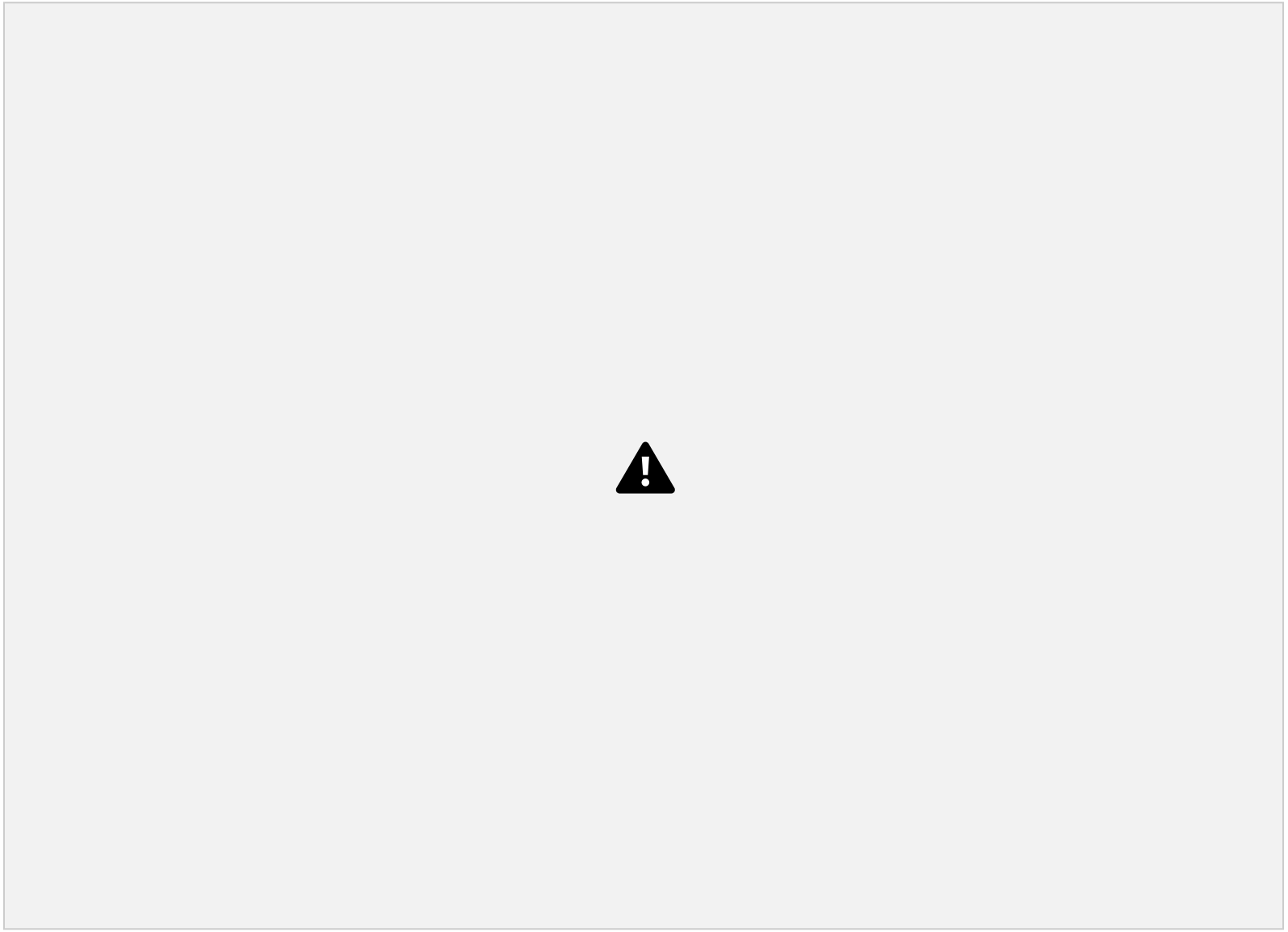
premian con más de 1



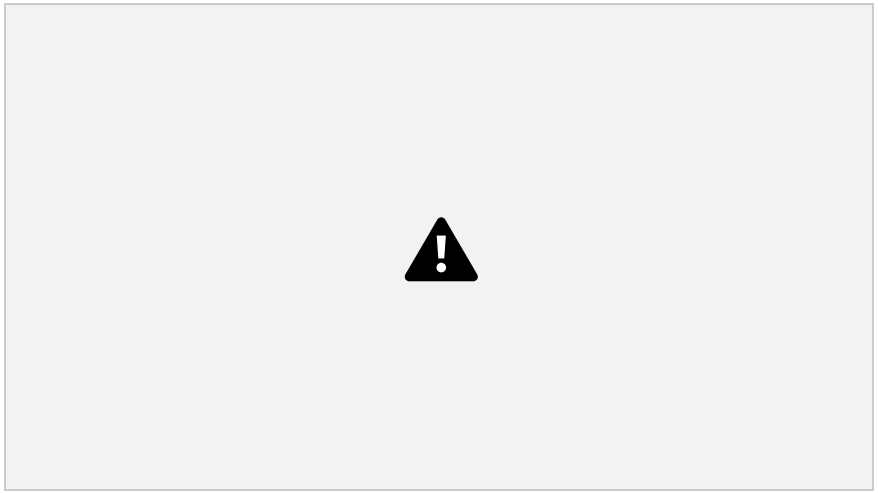




Dataset de kaggle para clase



Analítica de datos y extracción del conocimiento



Dos lenguajes de programación que permiten realizar análisis a un paquete de datos

Las archivos que se trabajan en Python se conocen como notebooks

Librería

s

Es una colección de funciones y métodos que permite realizar acciones sin necesidad de escribir código, tiene módulos integrados que proporcionan diferentes funcionalidades a usar

Librerías de Python

Librerías de datos

SciPy

Estructuras de datos y herramientas llamadas dataframes para manipulación y análisis de datos de manera efectiva.

Si principal instrumento es una tabla bidimensional que consiste en etiquetas de columnas y filas

Utiliza matrices para sus entradas y salidas, por lo que se puede procesar matrices de manera fácil y ágil.

Incluye funciones para problemas matemáticos avanzados

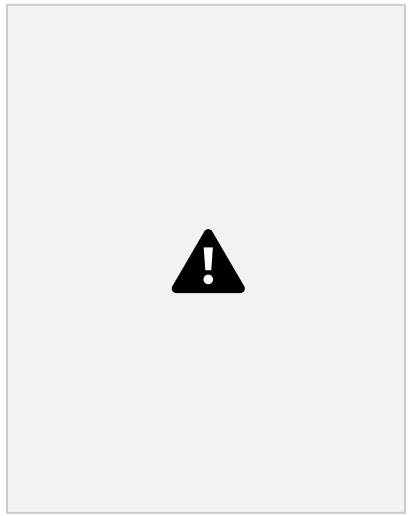
Como integrales

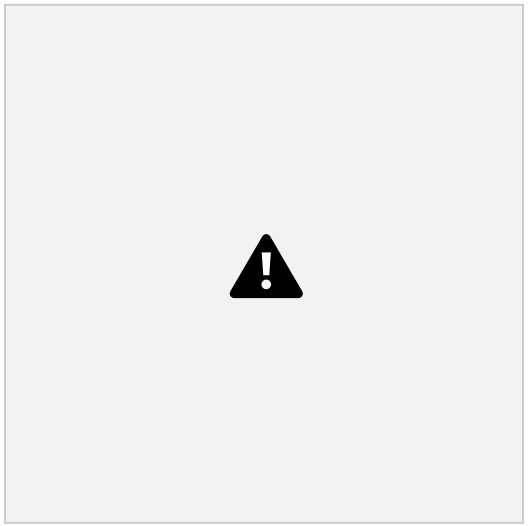
Ecuaciones diferenciales

Librerías de Python

Librerías de visualización

Comunican los resultados de los análisis en forma gráfica





Visualización de datos, ideal para realizar gráficos y tramas

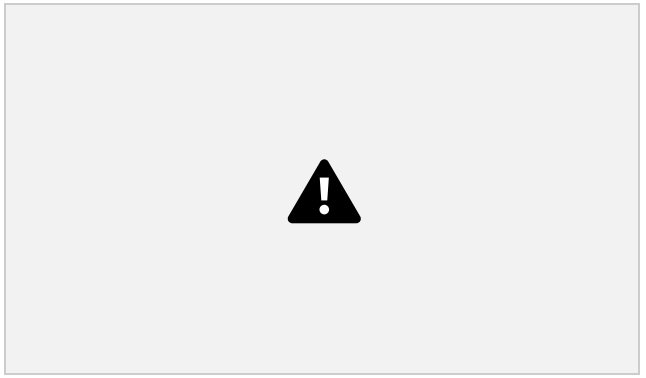
Se basa en matplotlib,

se pueden generar headmaps, series de tiempo, etc.

Librerías de Python

Librerías de algoritmos

Desarrollar modelos don el conjunto de datos para obtener predicciones



Contiene herramientas para modelado estadístico, incluido regresión, clasificación, agrupación, etc

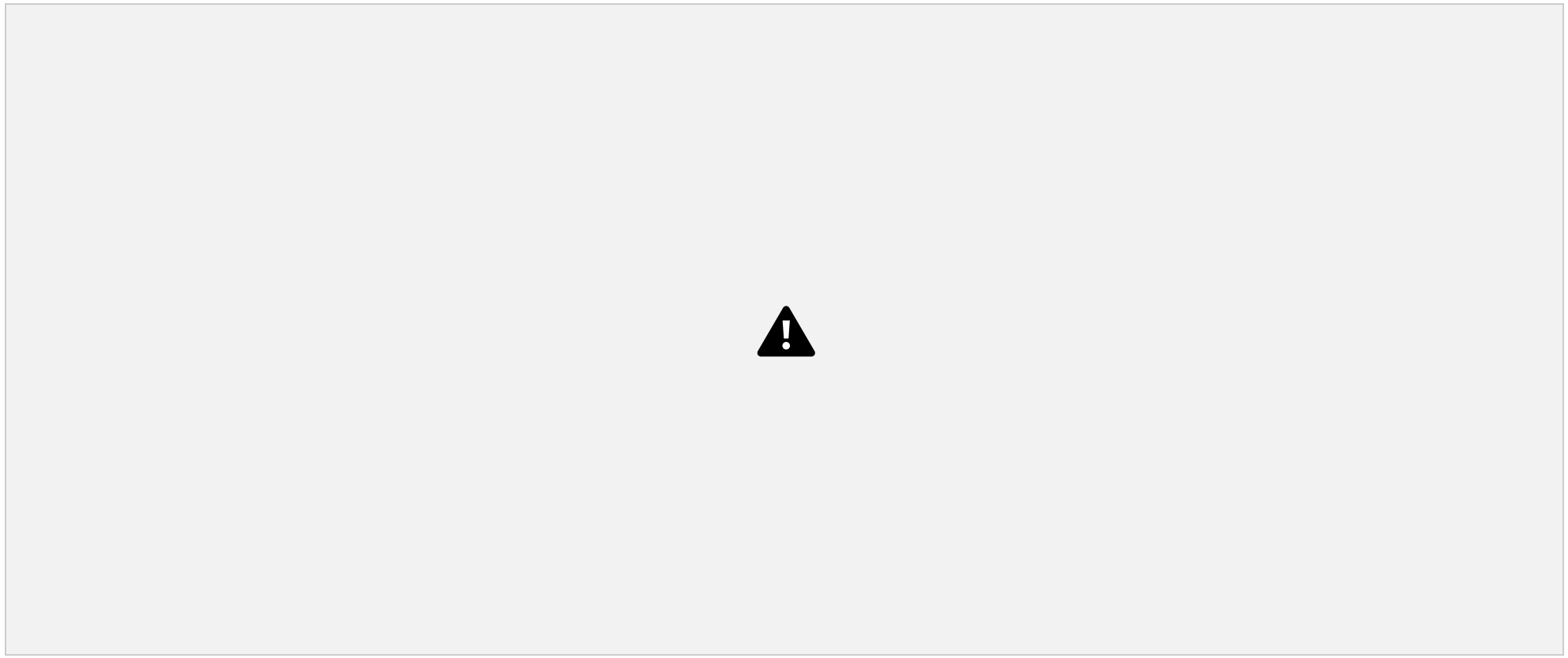
Se basa en

numpy, scipy y

matplotlib

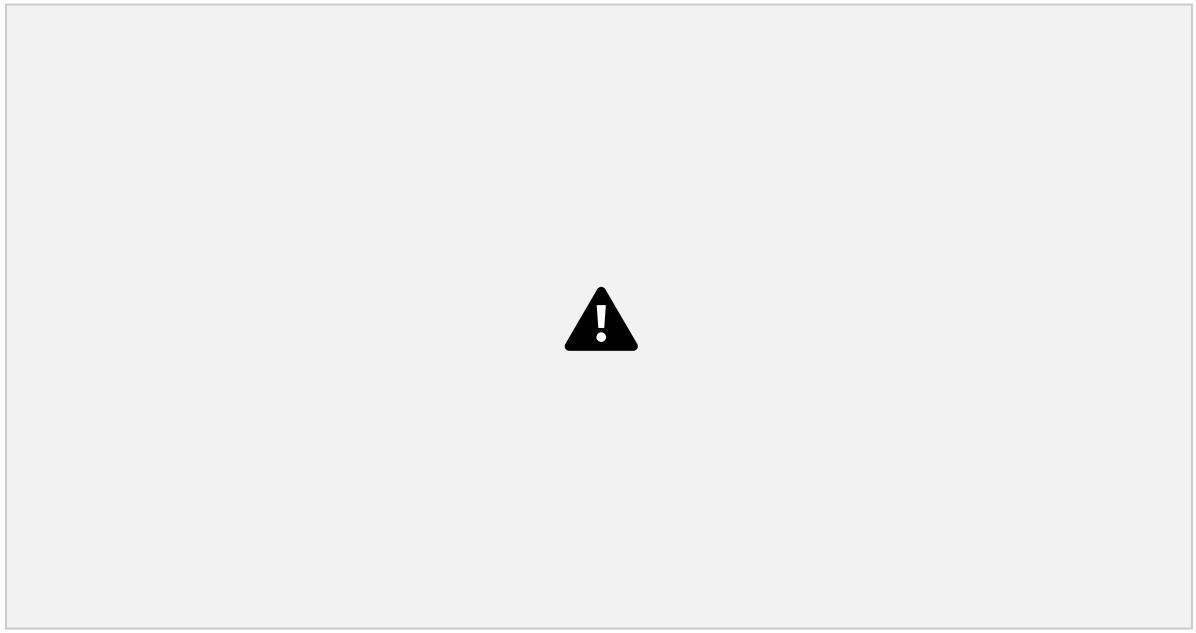
Modulo de Python que permite explorar datos, estimar estadísticas y modelos y realizar pruebas estadísticas

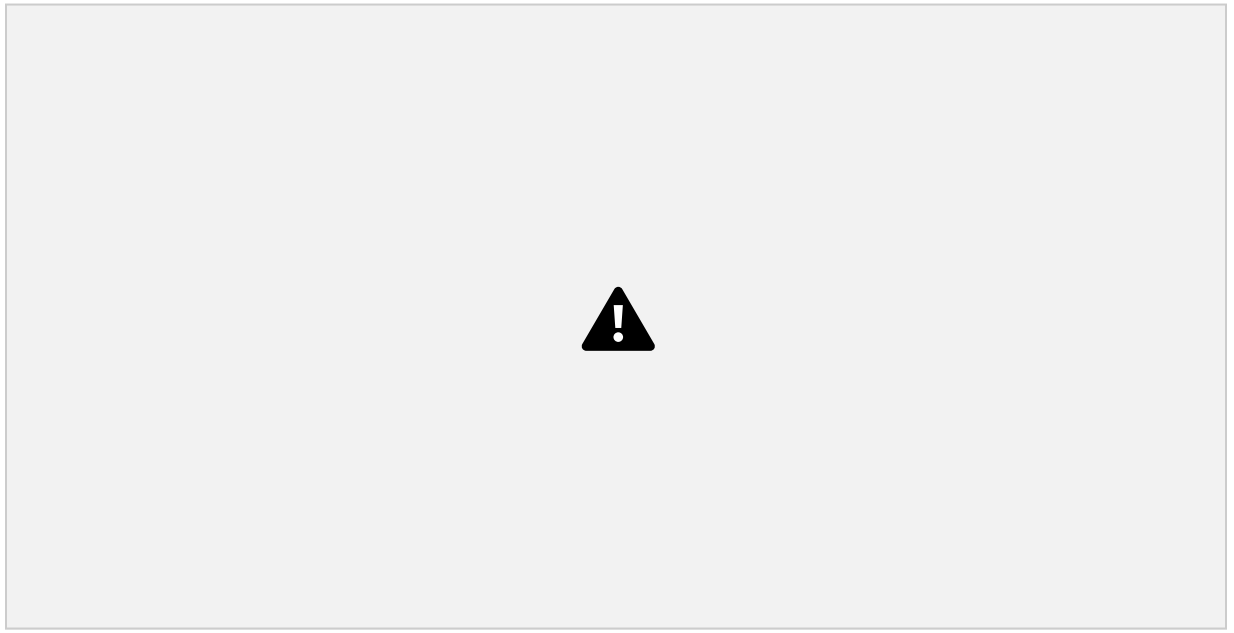
Librería pandas



https://aprendeconalf.es/docencia/python/manual/pandas/

Librería pandas





https://pandas.pydata.org/

Librería pandas

Pandas es una librería de Python especializada en el manejo y análisis de estructuras de

datos. 

Características

- Define nuevas estructuras de datos basadas en los arrays de la librería NumPy pero con nuevas funcionalidades.
- Permite leer y escribir fácilmente ficheros en formato CSV, Excel y bases de datos SQL.
- Permite acceder a los datos mediante índices o nombres para filas y columnas.
- Ofrece métodos para reordenar, dividir y combinar conjuntos de datos.
- Permite trabajar con series temporales.
- Realiza todas estas operaciones de manera muy eficiente.

Tipos de datos en pandas

Pandas dispone de tres estructuras de datos

diferentes: • Series: Estructura de una dimensión.

• DataFrame: Estructura de dos dimensiones (tablas).

• Panel: Estructura de tres dimensiones (cubos).

Estructuras se construyen a partir de *arrays* de la librería NumPy, añadiendo nuevas funcionalidades.

Objetos Series

Son *estructuras similares* a los arrays de una dimensión. Son homogéneas, es decir, sus elementos tienen que ser del mismo tipo, y su tamaño es inmutable, es decir, no se puede cambiar, aunque si su contenido.

Dispone de un índice que asocia un nombre a cada elemento del la serie, a través de la cuál se accede al elemento.

Creación de una Serie

Series(data=lista, index=indices, dtype=tipo) :

```
>>> import pandas as pd
>>> s = pd.Series(['Matemáticas', 'Historia', 'Economía', 'Programación', 'Inglés'],
dtype='string’)  >>> print(s)
0 Matemáticas
1 Historia
2 Economía
3 Programación
4 Inglés
dtype: string
```

Atributos de una Serie

Existen varias propiedades o métodos para ver las características de una serie.

- s.size : Devuelve el número de elementos de la serie s.
- s.index : Devuelve una lista con los nombres de las filas del DataFrame s.
- s.dtype : Devuelve el tipo de datos de los elementos de la serie s.

```
>>> import pandas as pd
>>> s = pd.Series([1, 2, 2, 3, 3, 3, 4, 4, 4, 4])
>>> s.size 10
>>> s.index
RangeIndex(start=0, stop=10, step=1)
>>> s.dtype
dtype('int64')
```

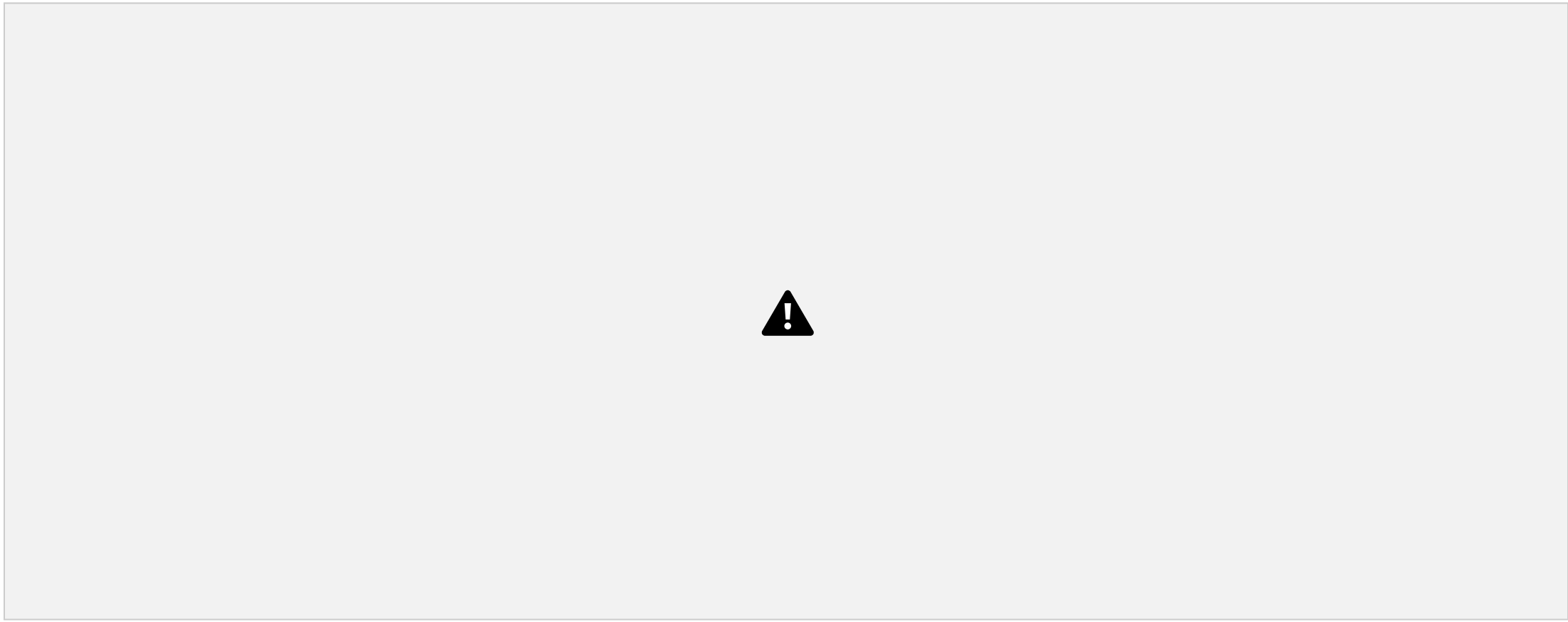
Funciones aplicables a una Serie

- s.count() : Devuelve el número de elementos que no son nulos ni NaN en la serie s. •

s.sum() : Devuelve la suma de los datos de la serie s cuando los datos son de un tipo numérico, o la concatenación de ellos cuando son del tipo cadena str.

- s.cumsum() : Devuelve una serie con la suma acumulada de los datos de la serie s cuando los datos son de un tipo numérico.
- s.value_counts() : Devuelve una serie con la frecuencia (número de repeticiones) de cada valor de la serie s.
- s.min() : Devuelve el menor de los datos de la serie s.
- s.max() : Devuelve el mayor de los datos de la serie s.
- s.mean() : Devuelve la media de los datos de la serie s cuando los datos son de un tipo numérico.
- s.std() : Devuelve la desviación típica de los datos de la serie s cuando los datos son de un tipo numérico.
- s.describe(): Devuelve una serie con un resumen descriptivo que incluye el número de datos, su suma, el mínimo, el máximo, la media, la desviación típica y los cuartiles.

Objetos DataFrame



Un *DataFrame* contiene dos índices, uno para las filas y otro para las columnas, y se puede acceder a sus elementos mediante los nombres de las filas y las columnas.

Atributos de un DataFrame

- df.info() : Devuelve información (número de filas, número de columnas, índices, tipo de las columnas y memoria usado) sobre el DataFrame df.
- df.shape : Devuelve una tupla con el número de filas y columnas del DataFrame df.
- df.size : Devuelve el número de elementos del DataFrame.
- df.columns : Devuelve una lista con los nombres de las columnas del DataFrame df.
- df.index : Devuelve una lista con los nombres de las filas del DataFrame df.
- df.dtypes : Devuelve una serie con los tipos de datos de las columnas del DataFrame df.
- df.head(n) : Devuelve las n primeras filas del DataFrame df.
- df.tail(n) : Devuelve las n últimas filas del DataFrame df.

Funciones aplicables a un DataFrame

df.count() : Devuelve una serie número de elementos que no son nulos ni NaN en cada columna del DataFrame df.

df.sum() : Devuelve una serie con la suma de los datos de las columnas del DataFrame df cuando los datos son de un tipo numérico, o la concatenación de ellos cuando son del tipo cadena str.

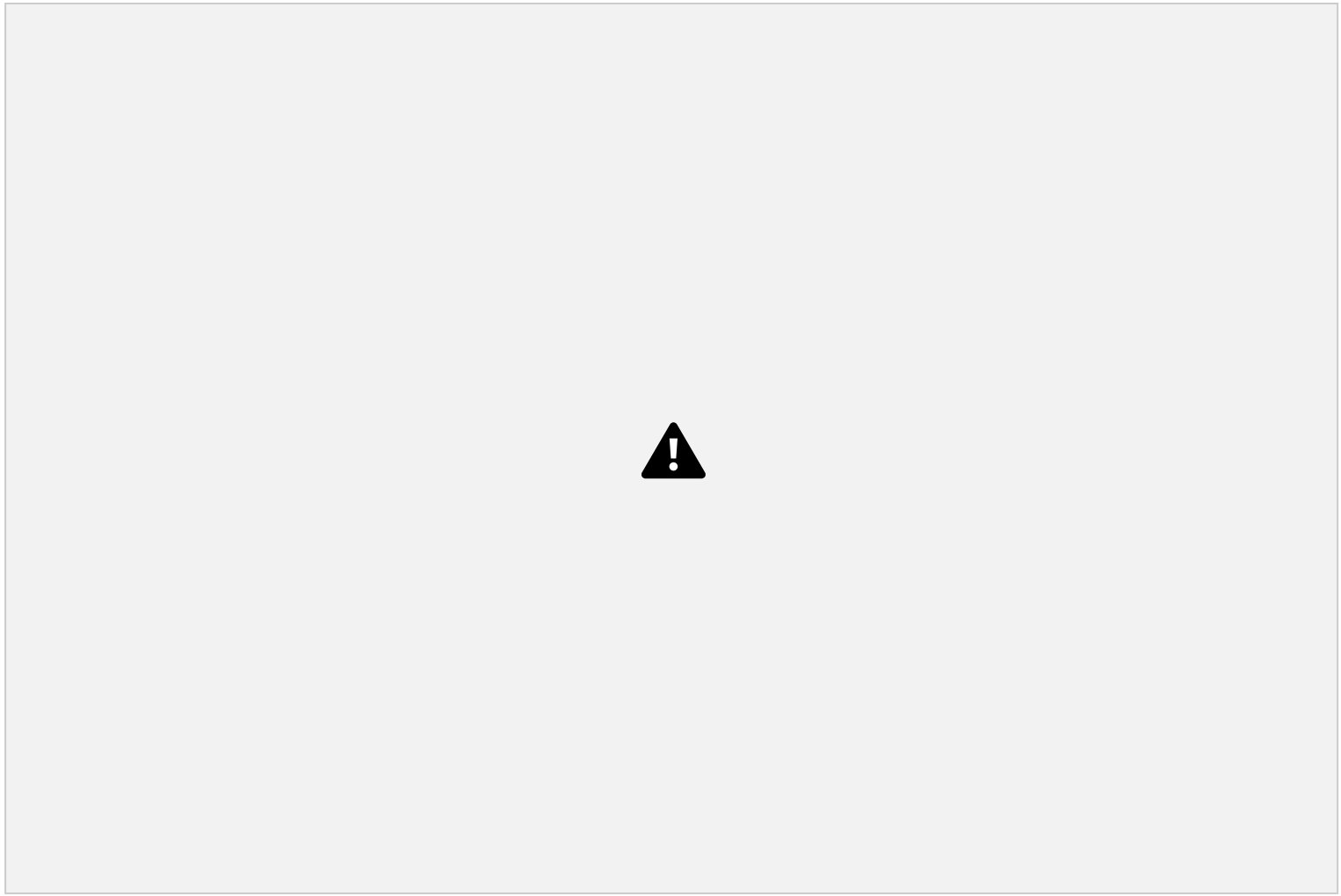
df.cumsum() : Devuelve un DataFrame con la suma acumulada de los datos de las columnas del DataFrame df cuando los datos son de un tipo numérico.

df.min() : Devuelve una serie con los menores de los datos de las columnas del DataFrame df. df.max() : Devuelve una serie con los mayores de los datos de las columnas del DataFrame df. df.mean() : Devuelve una serie con las media de los datos de las columnas del DataFrame df cuando los datos son de un tipo numérico.

df.std() : Devuelve una serie con las desviaciones típicas de los datos de las columnas del DataFrame df cuando los datos son de un tipo numérico.

df.describe(include = tipo) : Devuelve un DataFrame con un resumen estadístico de las columnas del DataFrame df del tipo tipo. Para los datos numéricos (number) se calcula la media, la desviación típica, el mínimo, el máximo y los cuartiles de las columnas numéricas. Para los datos no numéricos (object) se calcula el número de valores, el número de valores distintos, la moda y su frecuencia. Si no se indica el tipo solo se consideran las columnas numéricas.

Anaconda. ¿Qué es el Proyecto Jupyter?



Tomado de:

https://www.youtube.com/watch?v=Gi92BhWuuT0

Jupyter python

Jupyter es un proyecto heredado de la consola IPython la cual ha evolucionado y ha integrado sus notebooks en él.

Jupyter Python desarrollado por un colombiano

Fernando Pérez nació en Medellín, Colombia. Realizó su pregrado en Física en la Universidad de Antioquia y su maestría en Física en la misma universidad colombiana. Posteriormente obtuvo el título de doctorado en física de partículas de la Universidad de Colorado en Boulder, donde trabajó en simulaciones numéricas en Lattice QCD. Se trasladó a California en el 2008, donde trabaja actualmente para la universidad de

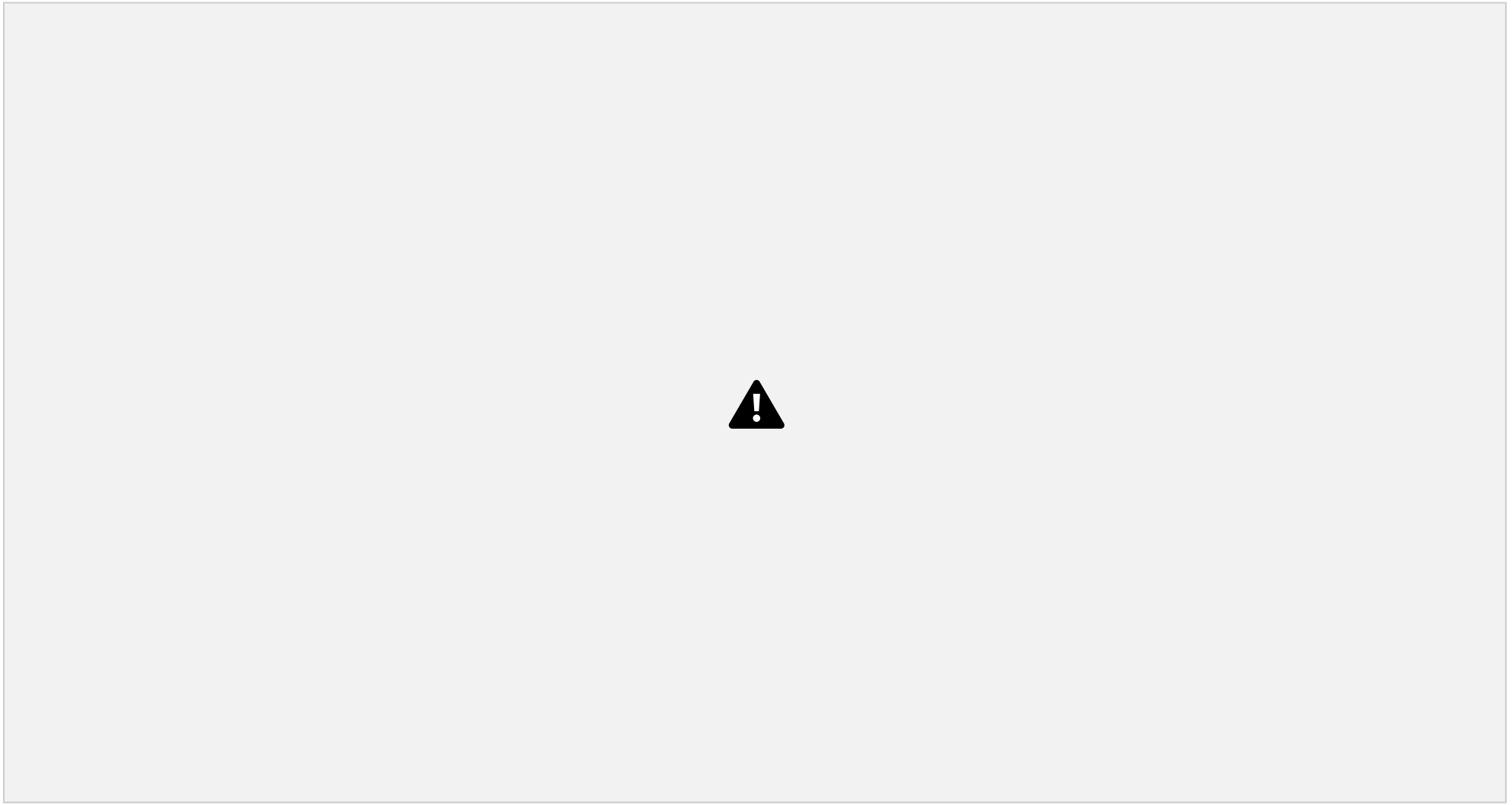
California en Berkeley en el departamento de estadística.

Antes fue parte de la plantilla de científicos del Laboratorio Nacional Lawrence Berkeley y había sido investigador asociado en Berkeley Institute for Data Science.

Pérez empezó a trabajar en IPython como proyecto particular en el 2001.

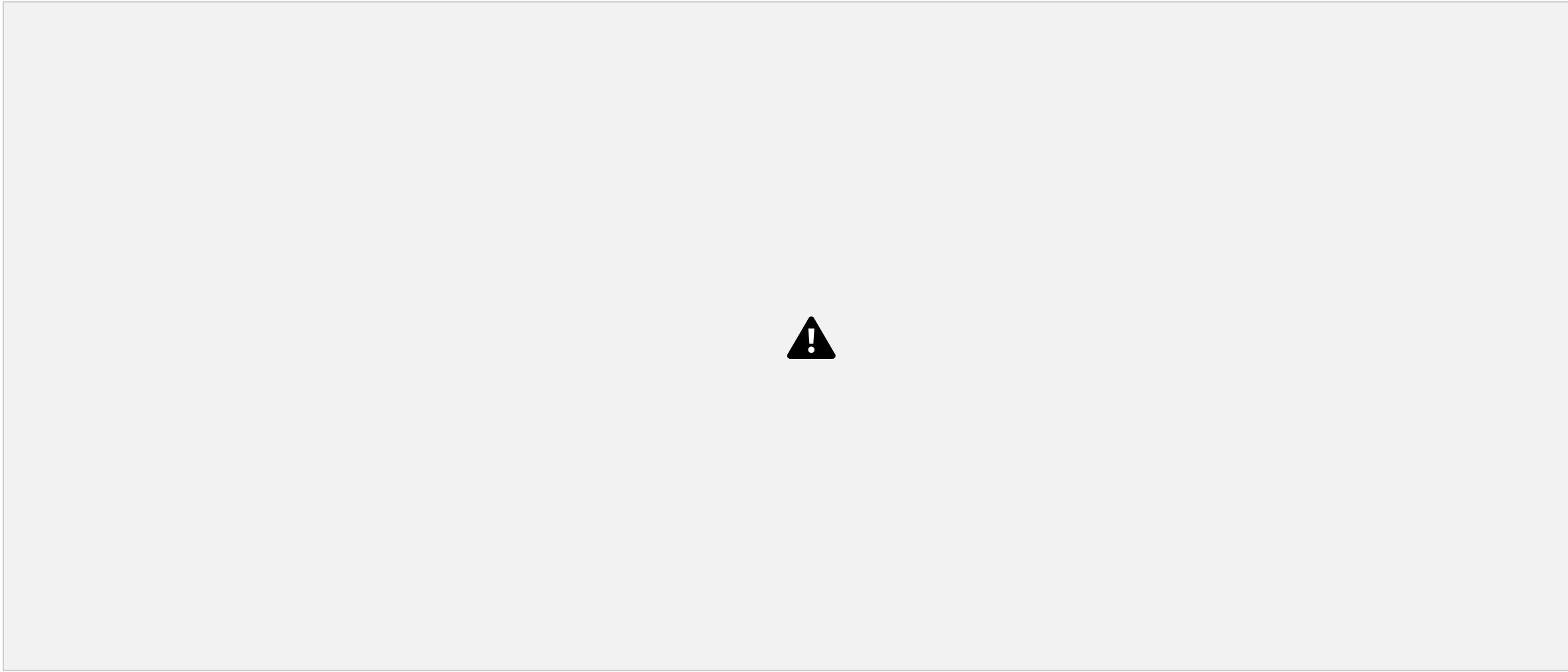
Tomado de: https://es.wikipedia.org/wiki/Fernando_P%C3%A9rez_(programador)

Instalación Jupyter python

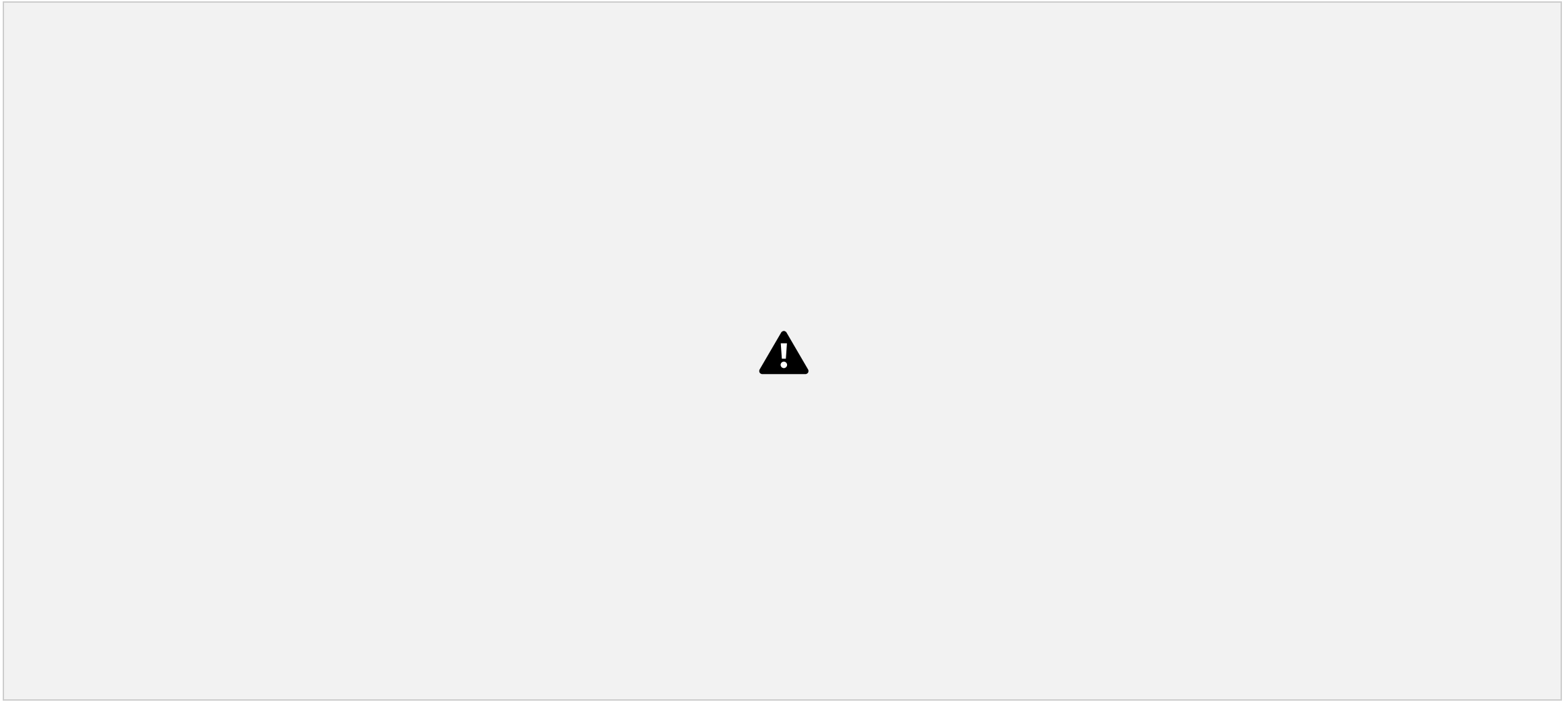


https://www.anaconda.com/

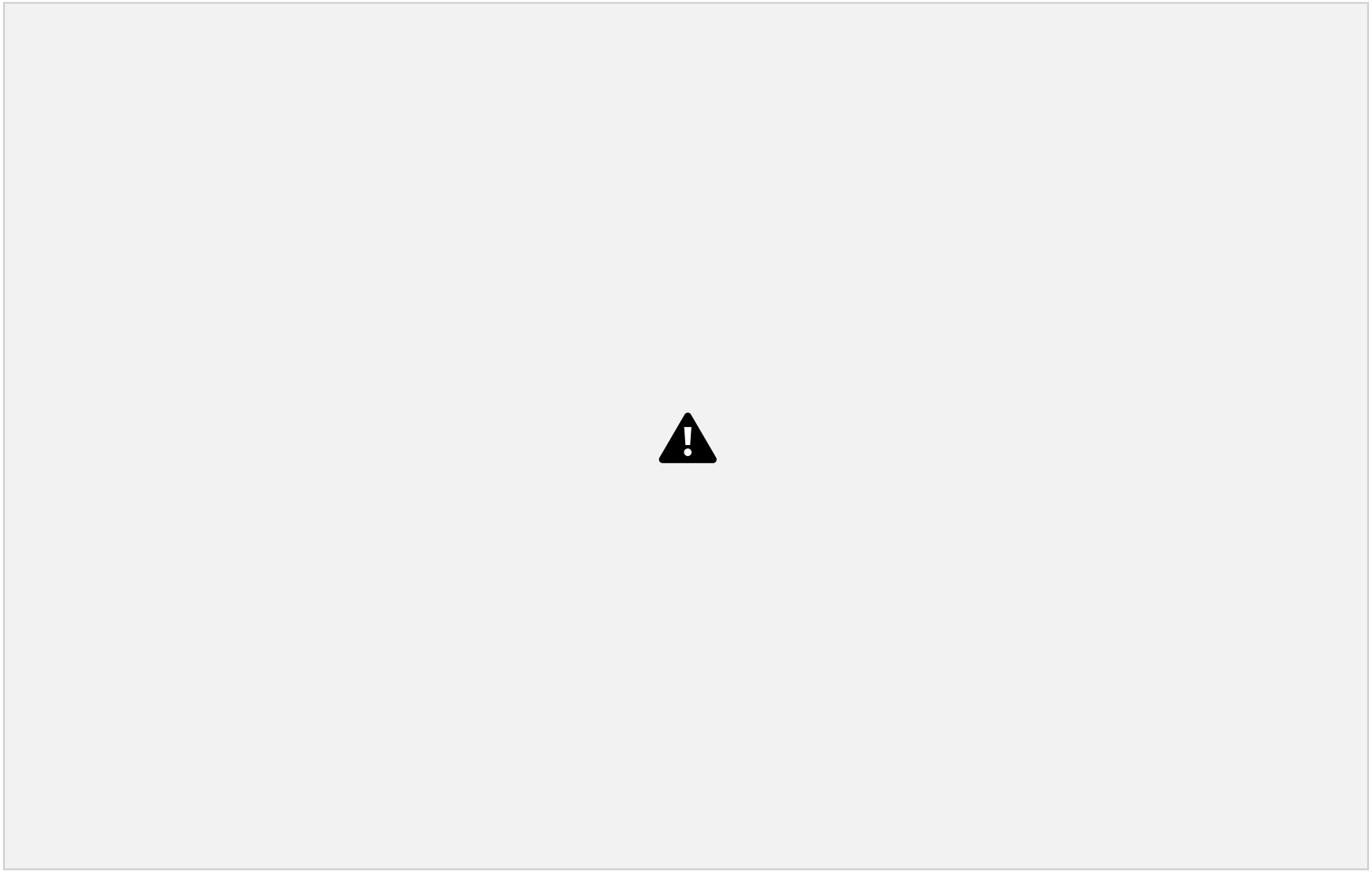
Instalación Jupyter python



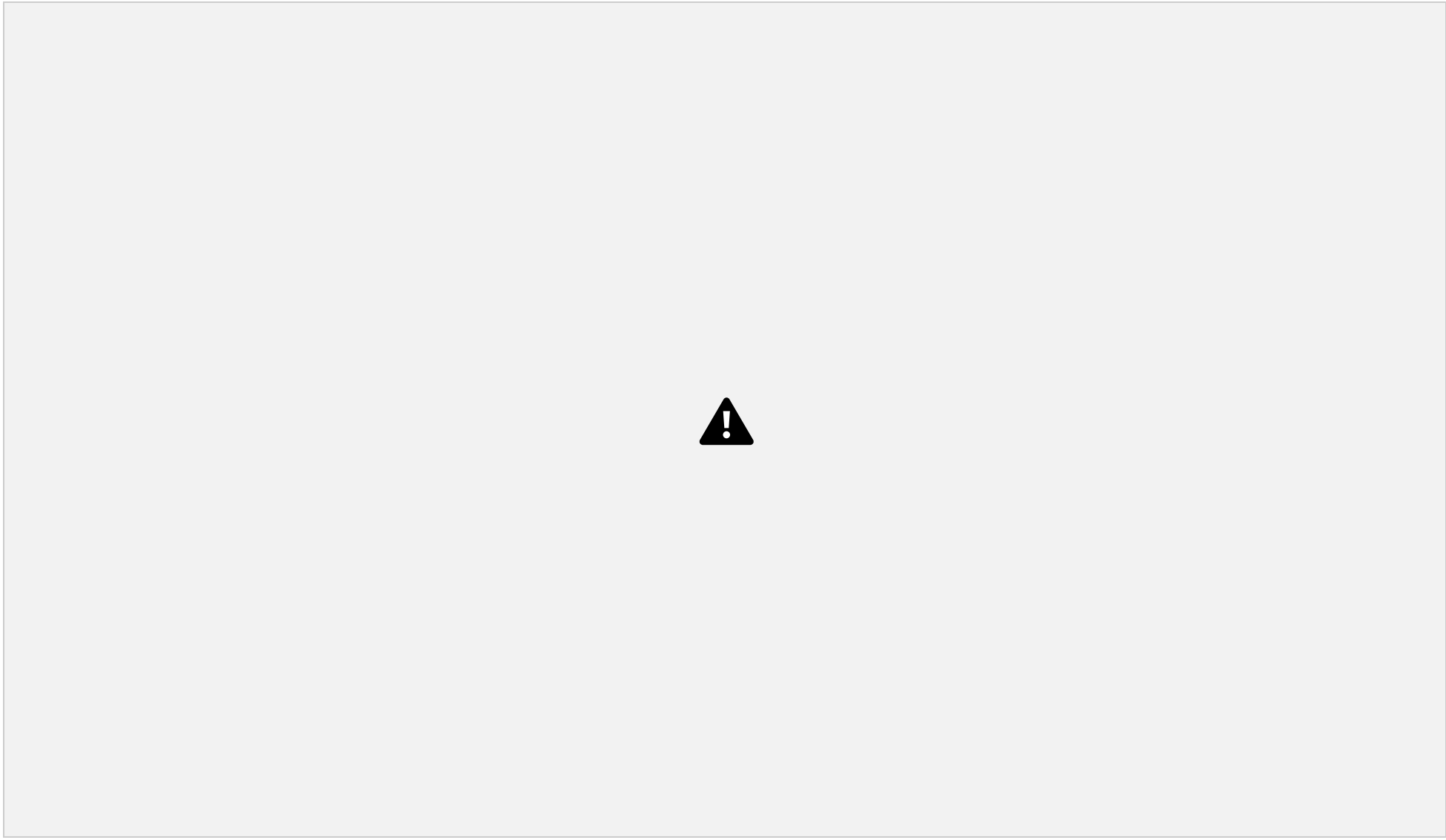
Instalación Jupyter python



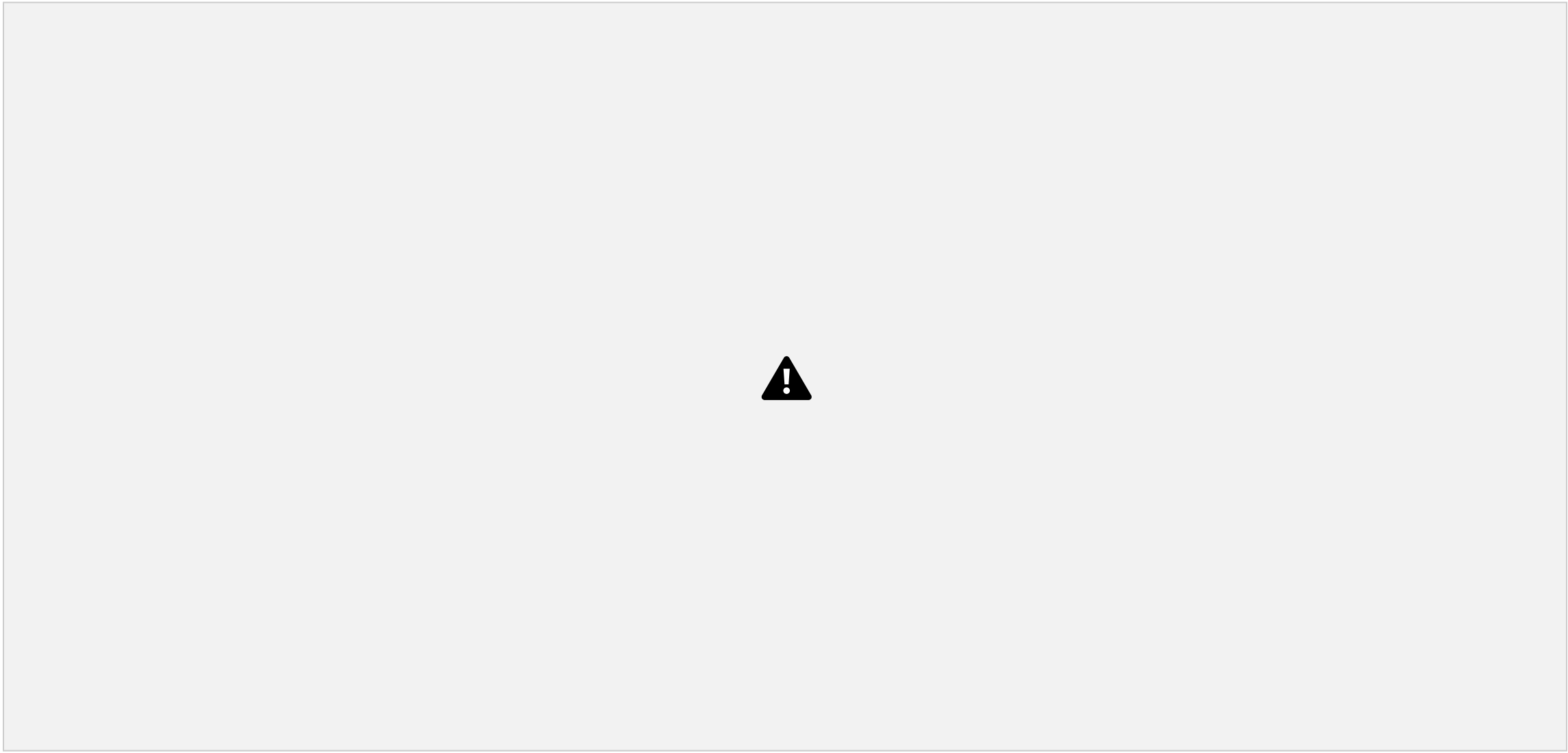
Instalación Jupyter python



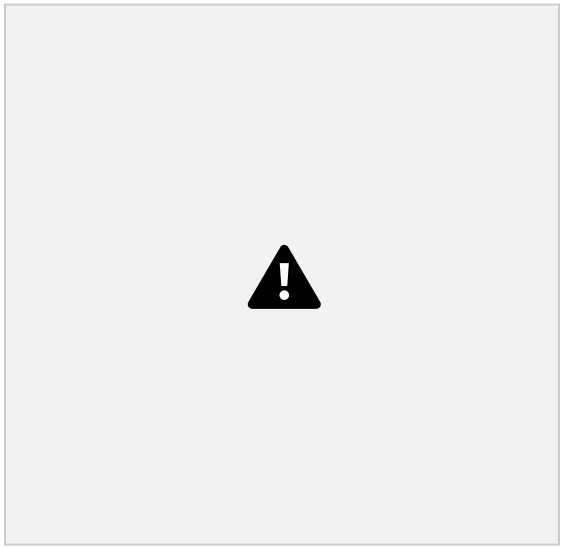
Instalación Jupyter python

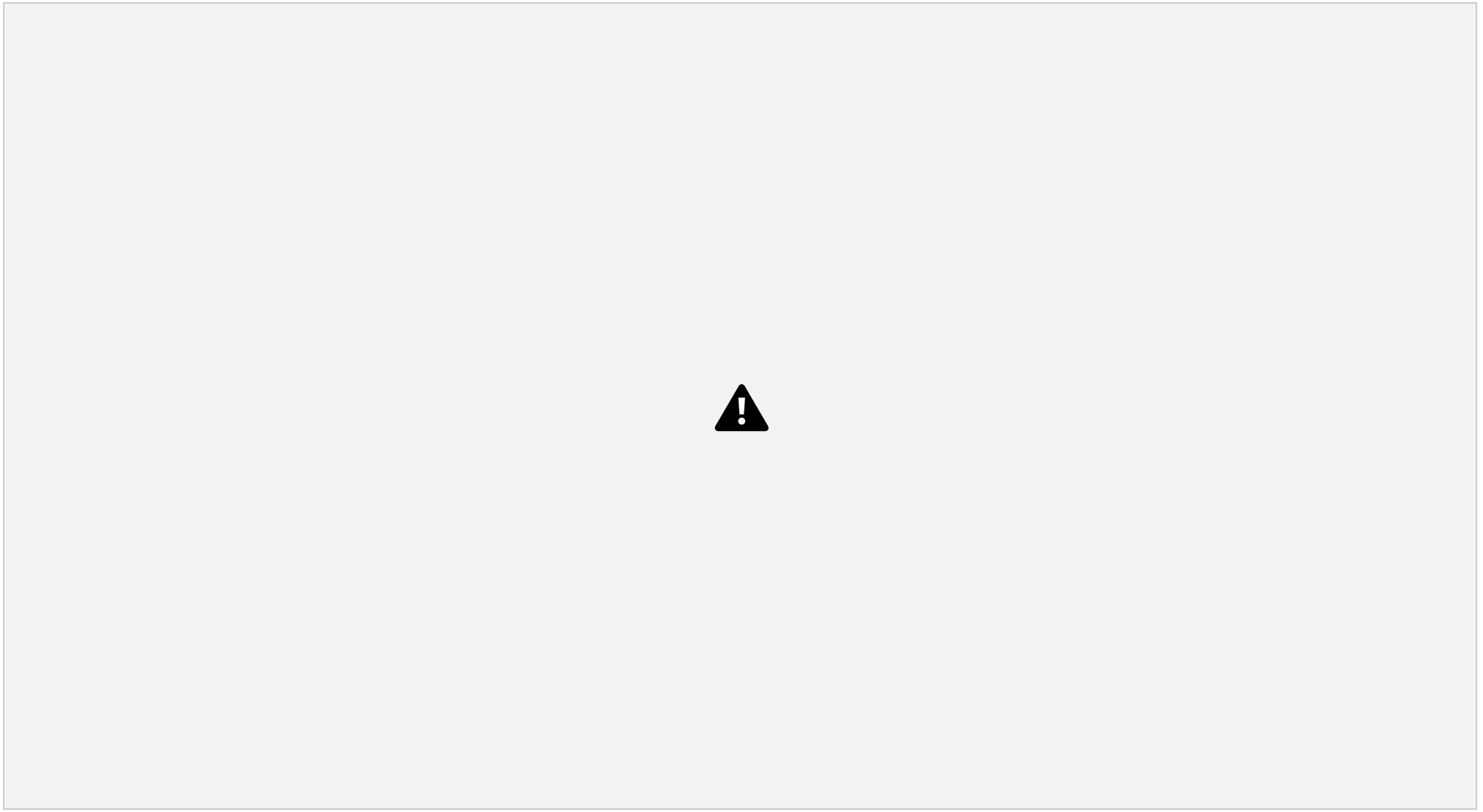


Instalación Jupyter python

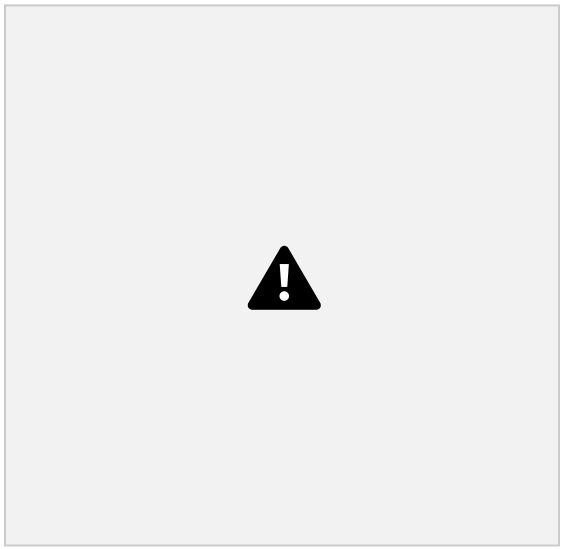


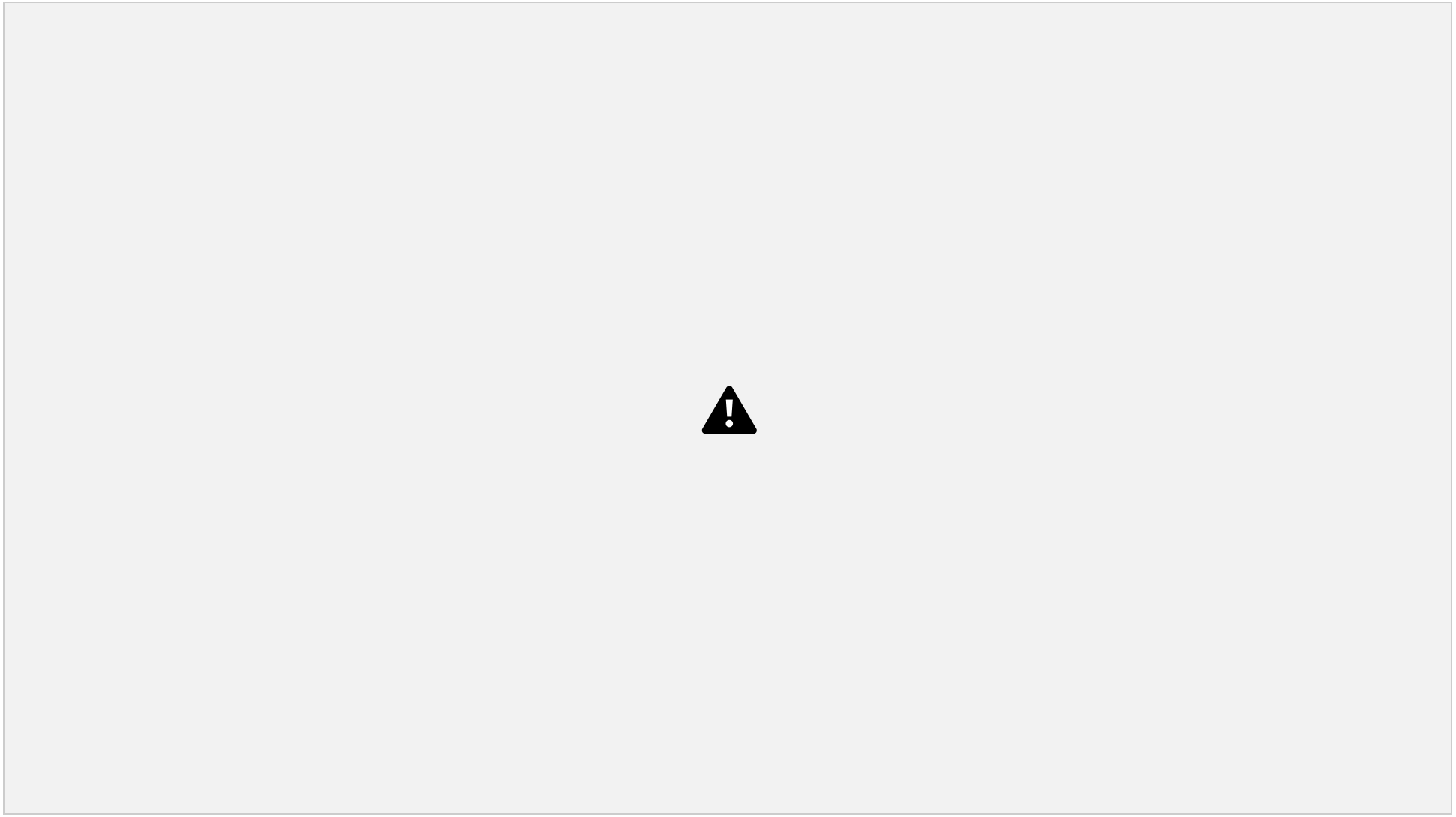
Jupyter notebook python





Anaconda GUI





Tarea

1. Realice un programa que sume, reste, multiplique y divide los elementos de dos matrices uno a uno, usando la librería numpy de Python,

• Para el ingreso de datos, utilice la captura de los mismos por teclado. • Para el calculo de las operaciones matemáticas utilice funciones. • Para mostrar el resultado en pantalla en forma que debe estar en forma de matriz realice otra función

A B

4

7

2	3
5	6
8	9

9	8	7
6	5	4
3	2	1

2. Realice la instalación de jupyter notebook en su computador y muestre en una presentación de diapositivas los pantallazos de la misma.

3. Realice en Google Colab la normalización y estandarización de los datos propuestos en está presentación