Pandas

Semestre 02, 2025

Introducción

Pandas es una librería externa para Python de código abierto. Para poder utilizarse es necesario instalarla primero ya que no está incluida en la instalación de Python.

🐼 El nombre viene de PANel DAta.

Es útil para análisis y manipulación de datos.

Instalación

Pandas se puede instalar usando el manejador de paquetes de Thonny.

Prueba de instalación

Deben correr el siguiente comando ya sea desde una terminal, Thonny, un script de Python o un Jupyter Notebook. Si el comando no da error quiere decir que está correctamente instalado.

```

  import pandas as pd

```

CSV

CSV significa Comma-Separated Values o valores separados por comas. Es un formato de archivo de texto plano que se utiliza para almacenar datos en forma tabular, como si fuera una hoja de cálculo.

Es posible verlos desde Excel o cualquier editor de texto.

```
nombre,edad,ciudad

Ana,23,Guatemala

Luis,30,Quetzaltenango

Marta,27,Antigua

```

En este archivo tenemos una tabla de 3 columnas: Nombre, Edad y Ciudad.

Usamos CSV porque:

- Es fácil de crear y leer.
- Es compatible con Excel, Google Sheets, bases de datos y lenguajes de programación.
- Es ligero y no depende de ningún software específico.

Data Frames

Son las estructuras de daos que usa Pandas. Constan de 3 componentes:

1. Columnas
2. Índices
3. Datos

Crear un data frame es muy parecido a crear un diccionario

```
df = pd.DataFrame({

"Carnet": [123, 456, 789, 987, 654],

"Hoja de trabajo 1": [80, 85, 75, 40, 100],

"Hoja de trabajo 2": [90, 90, 75, 61, 100],

```

```
10   "Hoja de trabajo 3": [80, 80, 0, 50, 85],
11
12   })
13
```

Imprimir un data muestra de forma ordenada la información.

```
1
2   print(df)
3
```

Instrucciones básicas

La forma del data frame nos da información sobre la cantidad de información que este tiene.

```
1
2   print(df.shape)
3
```

Si se requiere estadística básica es posible utilizar la función describe.

```
1
2   print(df.describe())
3
```

Es posible sacar estadística de forma más específica:

Promedio:

```
1
2   print(df.mean())
3
```

Cantidad de registros

```
1
2   print(df.count())
3
```

Máximo

```
print(df.max())
```

Mínimo

```
print(df.min())
```

Desviación estándard

```
print(df.std())
```

Solo una columna

```
print(df["Hoja de trabajo 1"])
```

Solo el promedio de una columna

```
print(df["Hoja de trabajo 1"].mean())
```

También es posible generar sub data frames con ciertas columnas

```
df2 = df[["Hoja de trabajo 1", "Hoja de trabajo 3"]]


print(df2)


print(df2.shape)
```

Si se requiere obtener un valor específico es posible hacerlo:

```
valor = df["Hoja de trabajo 1"][0]


print(valor)

```

Búsqueda de información

Poder acceder a información específica es uno de los poderes más grandes de Pandas. Para lograr esto se puede utilizar LOC ya que permite acceder a un grupo de filas y columnas usando etiquetas.

```
print(df.loc[df["Hoja de trabajo 1"] >= 61])

```

También es posible filtrar la información en el dataframe para limitar el resultado.

```
distinguidos = df.loc[df["Hoja de trabajo 1"] >= 80]


print(distinguidos)


print("\n--\n")


print(distinguidos[["Carnet", "Hoja de trabajo 1"]])

```

Agregar información

```
nuvas_notas = [100, 100, 100, 100, 100]


df["Hoja de trabajo 4"] = nuvas_notas

```

Actualizar información

Cambiarle el nombre a una columna:

```
df = df.rename(columns={"Hoja de trabajo 1": "Tarea 1"})


df = df.rename(columns={
"Hoja de trabajo 2": "Tarea 2",
"Hoja de trabajo 3": "Tarea 3"
})

```

cambiar el valor:

```
df.loc[df["Carnet"] == 123, "Hoja de trabajo 2"] = 95

```

Eliminar información

```
del df["Hoja de trabajo 4"]

```

Leer archivos

Previo a leer archivos es una buena práctica usar getcwd (get current working directory). Esto nos da el directorio actual en donde está corriendo el script.

```
import os


path = os.getcwd()


print(path)

```

Esto permite ahorrar tiempo en definir la ruta completa del archivo que queremos leer.

```
file = path + "/canciones.csv"


datos = pd.read_csv(file)


print(datos)

```

En datos se creo un dataframe con la información del archivo CSV. Ahora ya se puede hacer análisis completo sobre los datos.

Escribir archivos

Para escribir a un archivo se convierte un dataframe a un csv.

```

```

```
import os

import pandas as pd


path = os.getcwd()


canciones = [

{"titulo": "Cancion 1", "autor": "Autor 1", "duracion": 3.5},

{"titulo": "Cancion 2", "autor": "Autor 1", "duracion": 2.5},

{"titulo": "Cancion 3", "autor": "Autor 3", "duracion": 4.2},

{"titulo": "Cancion 4", "autor": "Autor 1", "duracion": 1.9},

]


df = pd.DataFrame(canciones)


file = path + "/canciones.csv"


df.to_csv(file, index=False)


print("Archivo escrito extiosamente")

```