Seminario de Lenguajes - Python

Cursada 2024

Introducción al análisis de datos

- Pandas
- Matplotlib

Ciencia de datos:

> En pocas palabras: se refiere al análisis significativo de datos aplicando técnicas de programación sobre un área de conocimiento específico.

Manejo de archivo de datos

- ¿Cuáles son las ventajas de utilizar Pandas sobre las estructuras de datos estándar de Python?
- ¿Cómo puede simplificar las operaciones realizadas con los módulos **csv** y **json**?
- ¿Qué otros formatos de archivos conocen para almacenar datos?
- ¿Cuáles son formatos abiertos?

Introducción a Pandas

- ¿Qué es Pandas y cuál es su importancia en el análisis de datos?
- ¿Cuáles son las principales estructuras de datos en Pandas?
- ¿Cómo importar y explorar un conjunto de datos usando Pandas?
- ¿Qué funcionalidades ofrece Pandas para conocer, filtrar y modificar los datos?

Pandas

- es una biblioteca de Python que proporciona estructuras de datos y herramientas de análisis de

datos de alto rendimiento y fáciles de usar.
- Esta orientada al análisis de datos porque permite manipular y analizar conjuntos de datos de manera eficiente.

In [ ]:
```
import pandas as....
```

In [ ]:
```
import pandas as pd
```

📌 Tipo de archivos y estructuras de datos en Pandas

¿Qué tipo de datos usaron con para reccorrer el contenido usando:

- csv
- json

Veamos un ejemplo

In [ ]:
```
from pathlib import  Path
file_route = Path('files')
file_data = 'ar-airports.csv'
df_airports = pd.read_csv(file_route/file_data)
```

In [ ]:
```
df_airports
```

In [ ]:
```

```

¿Qué tipo de datos es la variable utilizada?

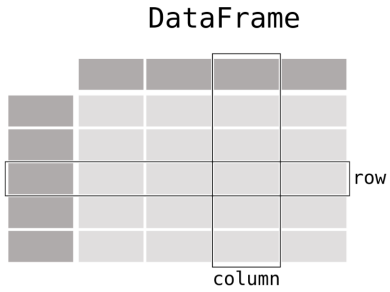
In [ ]:
```
type(df_airports)
```

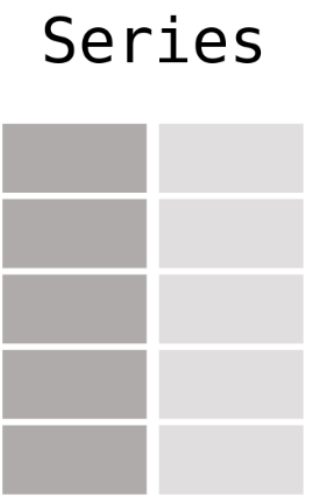
In [ ]:
```

```

Las principales estructuras de datos en Pandas

- **Series**: es un arreglo unidimensional de datos etiquetados.
- **DataFrame**: es una estructura de datos tabular bidimensional compuesta por filas y columnas.





Tipo de archivos y formas de abrir:

- **pd.read_csv()**: para leer archivos CSV (Comma Separated Values).
- **pd.read_excel()**: para leer archivos de Excel.
- **pd.read_json()**: para leer archivos JSON.
- **pd.read_html()**: para leer tablas HTML de una página web.
- **pd.read_sql()**: para leer datos de una base de datos SQL.
- **pd.read_hdf()**: para leer archivos HDF5 (Hierarchical Data Format version 5).
- Muchos mas

📌 Conociendo nuestros datos

Ver los primeros o últimos

```
df.head(number)
df.tail(number)
```

In [ ]:
```
df_airports.head(5)
```

Filas y columnas

In [ ]:
```
df_airports.shape
```

Nombres de las columnas

Si trabajamos con un archivo csv, ¿en dónde se encuentran los nombres de las columnas?

In [ ]:
```
df_airports.columns
```

Información de nuestros datos

- ¿Qué otra información nos puede servir, por ejemplo si vamos a buscar datos en la columna **municipality** del archivo de aeropuertos?
- ¿Recuerdan que problemas tenía esta columna?

In [ ]:
```
df_airports.info()
```

¿Cómo podemos obtener los valores de algunos cálculos estadísticos básicos, sobre las columnas con datos numéricos?

In [ ]:
```
df_airports.describe()
```

¿Qué es cada cálculo?

- cantidad de valores no nulos
- mínimo
- media
- desviación estándar: cuánto varían los valores en relación con la media.
- máximo
- 25%(primer cuartil): indica el valor por debajo del cual cae el 25% de los datos.
- 50%(mediana): es el valor que divide la serie en dos mitades iguales.
- 75%(tercer cuartil): indica el valor por debajo del cual cae el 75% de los datos.

In [ ]:
```
df_airports.dtypes
```

- **info**: cantidad de valores nulos por columnas y tipo de datos que pandas le asignó.
- **describe**: cálculos sobre las columnas que contienen datos numéricos.
- **dtypes**: solamente los tipos de datos asignados por pandas.
- Info tipos de datos

¿Cuánta memoria estamos usando con nuestro dataset?

In [ ]:
```
df_airports.info(verbose=False, memory_usage='deep')
```

Quiero saber la cantidad de aeropuertos que hay de cada tipo

1. Consulto los nombres de las columnas para saber donde está esa información

In [ ]:
```
df_airports.columns
```

2. Consulto cuáles son los valores únicos de la columna **type**

In [ ]:
```
df_airports.type.unique()
```

3. Calculo cuántos hay de cada tipo

In [ ]:
```
df_airports.type.value_counts()
```

Trabajando con datos de una sola columna

- Queremos guardarnos los nombres de todos los aeropuertos

In [ ]:
```
df_names = df_airports.name
```

In [ ]:
```
type(df_names)
```

¿Pero es lo mismo un Dataframe que una variable de tipo Series?

📌 Series

- Tipo de dato unidimensional, es como un arreglo con etiquetas de índice.
- Los tipos de operaciones pueden variar con respecto a un Dataframe:
  - modificación de columnas (cambiar orden)
  - modificaciones de columnas o filas (agregar, eliminar)
  - agrupación de de datos
  - otros...
- Info

📌 Acceso a datos por índice o etiquetas

In [ ]:
```
df_airports.iloc[0]
```

In [ ]:
```
df_airports.loc[0]
```

- **loc**: por label, en este caso el índice en un número
- **iloc**: por índice entero, es el número de la fila.

🧰 Manipulación de Datos con Pandas

- ¿Cómo seleccionar, filtrar y transformar datos?
- ¿Cuáles son las operaciones comunes para manipular y procesar datos?

📌 Filtrado de datos

- Filas que coincidan con con un valor específico de una columna.
- Filas que cumplan condición con operadores booleanos.
- Selección de columnas específicas.
- Filas que coincidan con algún string o elemento de una lista.

Queremos encontrar:

- los aeropuertos que coincidan con un tipo de aeropuerto dado.
- los aeropuertos que se encuentren en valores o rangos correspondientes a la elevación.
- los aeropuertos de tipo: large, medium, small.

Los aeropuertos que coincidan con un tipo de aeropuerto dado.

In [ ]:
```
df_airports[df_airports.type=='medium_airport'].head(3)
```

¿Por qué usamos dos veces la variable **df_airports**?

In [ ]:
```
df_airports.type=='medium_airport'
```

- Nos devuelve un array de booleanos, es en realidad una máscara.
- Luego se aplica esa máscara a las filas del Dataframe.

In [ ]:
```
mask = df_airports.type=='medium_airport'
df_airports[mask].head(3)
```

Los aeropuertos que se encuentren en valores o rangos correspondientes a la elevación.

In [ ]:
```
df_airports[df_airports.elevation_ft<131].head(3)
```

In [ ]:
```
df_airports[(df_airports.elevation_ft>131) & (df_airports.elevation_ft<903)].head(3)
```

¿Son los mismos operadores booleanos usados en Python?

📌 Operadores booleanos

|Operador|Python|pandas| |----------|---------|--------| |**and**|cond1 and cond2|(cond1) & (cond2)| |**or**|cond1 or cond2|(cond1) | (cond2)| |**not**|not cond|~cond|

🚨 Desafío

Utilizando pandas encontrar los lagos según los siguientes criterios:

- Lagos con una superficie menor o igual a 17 km²
- Lagos con una superficie mayor que 17 km² y menor o igual a 59 km²

- Lagos con una superficie mayor a 59 km²

Filtrado de los aeropuertos de tipo: large, medium, small

In [ ]:
```
df_airports.type.unique()
```

Hay dos posibles formas

- Filtrar los aeropuertos que contengan el string **airport**, ya que los demás no contienen ese string.
- Filtrar con una lista que contenga las categorías que estamos buscando.

Filtrar indicando el string airport, ya que los demás no lo contienen.

In [ ]:
```
df_airports[df_airports.type.str.contains('airport')].head(3)
```

`df.column.str.contains('string')`
Permite filtrar las filas que en la **columna** se encuentre el string dado

Filtrar con una lista que contenga las categorías que estamos buscando.

In [ ]:
```
df_airports.type.unique()
```

In [ ]:
```
airports_int = ['large_airport', 'medium_airport',  'small_airport']
```

In [ ]:
```
df_airports_int = df_airports[df_airports.type.isin(airports_int)]
```

In [ ]:
```
df_airports_int
```

`df.columna.isin(lista)`
Permite filtrar las filas cuyo contenido de la **columna** dada sea algunos de los elementos de la **lista** pasada como parámetro.

🚨 Encontrar los valores de los aeropuertos con las elevaciones 220, 290 o 639.

In [ ]:
```
elevations_ask = [220, 290,639]
df_airports[df_airports.elevation_ft.isin(elevations_ask)]
```

🎒 Gráficos con Matplotlib

¿Qué tipos de gráficos conocen?

Con Matplotlib se pueden realizar

- Barra
- Línea
- Torta
- Scatter
- Histograma
- Boxplot
- 3D
- y muchos más

Primero importamos:

In [ ]:
```
import matplotlib.pyplot as plt
```

Generar gráficos simples

Graficar los tipos de aeropuertos según el tipo

- barra
- torta

Utilizando plot directamente con los valores contados

In [ ]:
```
conteo_tipos = df_airports_int.type.value_counts()
```

Crear el gráfico de barra

In [ ]:
```
conteo_tipos.plot(kind='bar', xlabel='Tipo de aeropuerto', ylabel='Cantidad', title='(
plt.show()
```

Crear el gráfico de torta

In [ ]:
```
conteo_tipos.plot(kind='pie', autopct='%1.1f%%', title='Proporción de aeropuertos por

# Mostrar el gráfico de torta
plt.show()
```

Utilizando las funciones de Matplotlib

In [ ]:
```
values = df_airports_int.type.value_counts().values
labels = df_airports_int.type.value_counts().index
```

📌 Crear el gráfico de barras

In [ ]:
```
plt.bar(labels,values)
# Agregar etiquetas y título
plt.xlabel('Tipo de aeropuerto')
plt.ylabel('Cantidad')
plt.title('Cantidad de aeropuertos por tipo')

# Rotar etiquetas en el eje x para mejor legibilidad
plt.xticks(rotation=45)
# Mostrar el gráfico de barras
plt.show()
```

In [ ]:
```

```

📌Crear el gráfico de torta

In [ ]:
```
plt.pie(values, labels=labels, autopct='%1.1f%%')

# Agregar título
plt.title('Proporción de aeropuertos por tipo')

# Mostrar el gráfico de torta
plt.show()
```

In [ ]:
```
import squarify    # pip install squarify (se necesita instalar para generar gráficos

squarify.plot(sizes=values, label=labels, alpha=.7 )
plt.axis('off')
plt.show()
```

In [ ]:

Para pensar en casa

- ¿Cómo manejar valores faltantes y duplicados en un conjunto de datos?
- ¿Cómo agregamos columnas, y cómo podemos hacerlo aplicando una función?
- ¿Qué técnicas ofrece Pandas para realizar agregaciones y agrupaciones de datos?
- Nuestros datos y gráficos interactuando con Streamlit.

Seguimos la próxima ...

In [ ]: