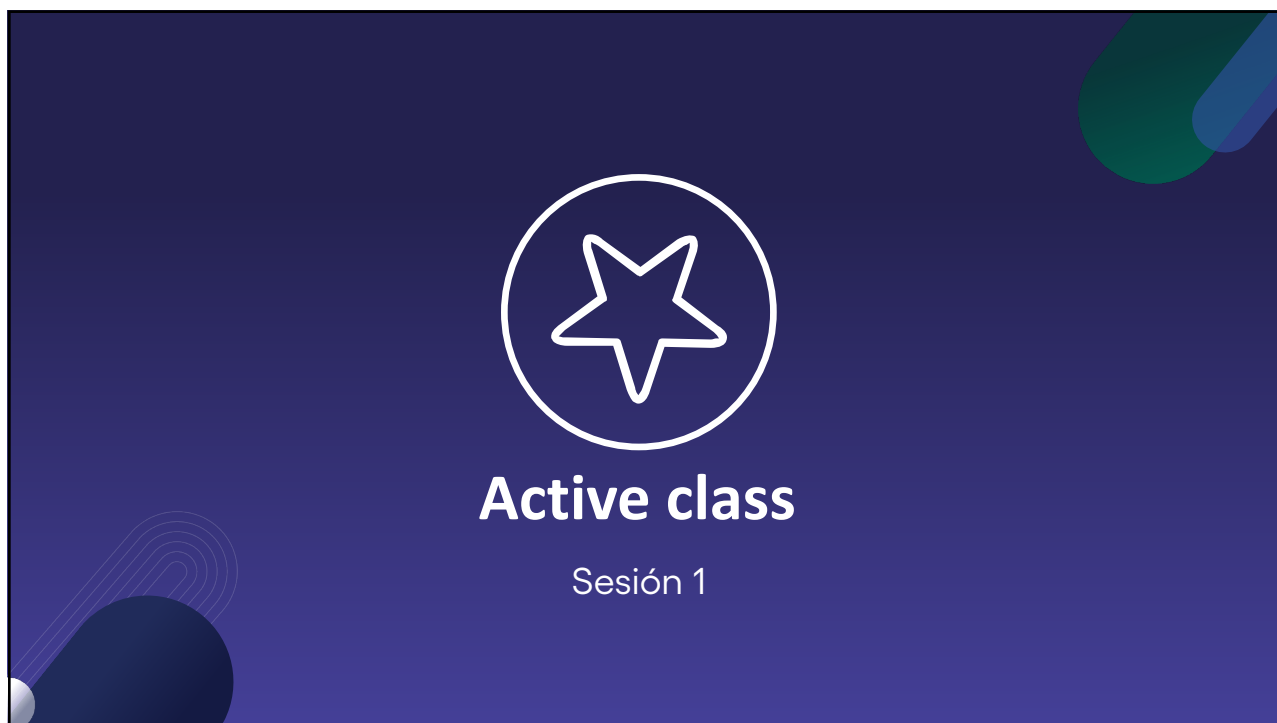


1



2



Módulo 2

Conceptos de almacenamiento y recuperación de información

3

Análisis numérico en Python

- **NumPy** (*Numerical Python*) es una librería de Python, que agrega soporte para arreglos y matrices.
- Incorpora una gran cantidad de funciones matemáticas de alto nivel para operar en estas estructuras.

<https://numpy.org>



4

Análisis de datos en Python

- **pandas** (*panel data*) es una librería de código abierto para el análisis y manejo de datos en Python.
- Permite acceder a los datos mediante índices o nombres (tanto para filas como para columnas), en concordancia con los **formatos tabulares** más usados.

<https://pandas.pydata.org/>



5

dataframe

Es la estructura más importante en pandas. Tiene **dos dimensiones** de datos etiquetados.

ROWS

COLUMNS

	Personnel	Position
0	Leslie Knope	Deputy Director
1	Ron Swanson	Director
2	Ann Perkins	Health Representative
3	Tom Haverford	Administrator
4	Mark Brendanawicz	City Planner
5	April Ludgate	Assistant - Director
6	Andy Dwyer	Assistant - Deputy Director
7	Ben Wyatt	Deputy City Manager
8	Chris Traeger	City Manager
9	Jerry Gergich	Administrator
10	Donna Meagle	Office Manager
11	Craig Middlebrooks	Assistant Office Manager

Las columnas pueden almacenar cualquier tipo de datos. Cada columna en un dataframe es una **serie**.

6

Lectura en pandas

- La **lectura** de datos tabulares es la forma más frecuente de encontrar y obtener información, independientemente de su origen (archivos o bases de datos) o formato.
- Para la lectura en pandas se utilizan las funciones **read_*** donde el * representa el formato del origen de la información. Por ejemplo: csv, excel, html, json, entre otras.
- El resultado de la lectura queda almacenado en un **dataframe**.



	Name	Symbol	Shares
0	Microsoft Corporation	MSFT	100
1	Google, LLC	GOOG	50
2	Tesla, Inc.	TSLA	150
3	Apple Inc.	AAPL	200
4	Netflix, Inc.	NFLX	80

7

dataframe (df)

Atributos

`df.shape` – Dimensionalidad
`df.columns` – Identificadores de columnas
`df.index` – Identificadores de filas
`df.dtypes` – Tipos de datos

Métodos

`df.head(6)` – Primeros 6 registros
`df.tail(3)` – Últimos 3 registros
`df.set_index('col_name')` – La columna queda como índice
`df.reset_index()` – Se reinicia el índice
`df.nunique()` – Cantidad de valores únicos por columna
`df['col_name'].unique()` – Valores únicos de la columna
`df['col_name'].value_counts()` – Frecuencia de valores
`df.sort_values('col_name')` – Ordena por valores de columna
`df.isna().sum()` – Valores faltantes por columna



8

Estadísticas descriptivas

```
df['col_name'].count()  
df['col_name'].min()  
df['col_name'].max()  
df['col_name'].mean()  
df['col_name'].median()  
df['col_name'].std()  
df['col_name'].quatile(.25)
```

La estructura *dataframe* de Pandas agrupa estas medidas en la función `describe()`, que genera estadísticas descriptivas para todas las columnas numéricas.

* Todas las anteriores **excluyen** los valores `NaN`

9

Estadísticas descriptivas

También se puede usar para las columnas de texto como:

```
describe(include='object')
```

En este caso incluye:

- El conteo (`count`)
- La cantidad de valores únicos (`unique`)
- El valor más frecuente (`top`) y su frecuencia (`freq`)

10

Agrupamiento

El método `groupby()` reorganiza en el sentido lógico el dataframe, formando particiones o grupos, de manera que dentro de cada grupo todas las filas tengan el mismo valor en la columna especificada.

Enseguida se aplica una función de agregado a cada **grupo** del dataframe dividido (y no a cada fila del dataframe original).

Cada resultado de la función de agregado debe producir **un solo valor**.

```
df.groupby('col_name').mean(numeric_only=True)
```

11

API plot de pandas



El panorama de visualización de Python puede parecer abrumador al principio.

Incluso se ha creado **PyViz.org**, un sitio para ayudar a los usuarios a decidir cuáles son las mejores herramientas de visualización de código abierto de Python para sus propósitos.

<https://pyviz.org/overviews/index.html>

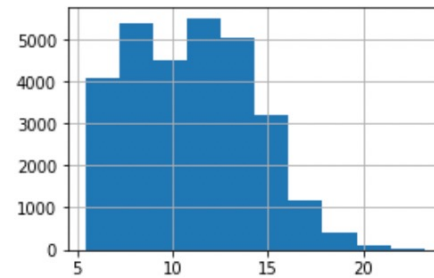
Una de las más antiguas es la **API plot** de pandas. Esta interfaz renderiza gráficos estáticos en libretas de Jupyter o para exportar desde Python, con un comando que puede ser tan simple como:

```
df.plot()
```

12

Histogramas

- Es una representación en barras de la **distribución** de los datos.
- En el eje horizontal se indican los valores o subrango de valores de las variables y en el vertical sus frecuencias.
- Utiliza este tipo de diagrama cuando desees observar el grado de homogeneidad o variabilidad de las columnas cuantitativas continuas del dataframe.



```
df['col_name'].plot(kind='hist')  
df['col_name'].plot.hist()
```

13



Tecnológico
de Monterrey

D.R.© Tecnológico de Monterrey, México, 2022.
Prohibida la reproducción total o parcial
de esta obra sin expresa autorización del
Tecnológico de Monterrey.

14