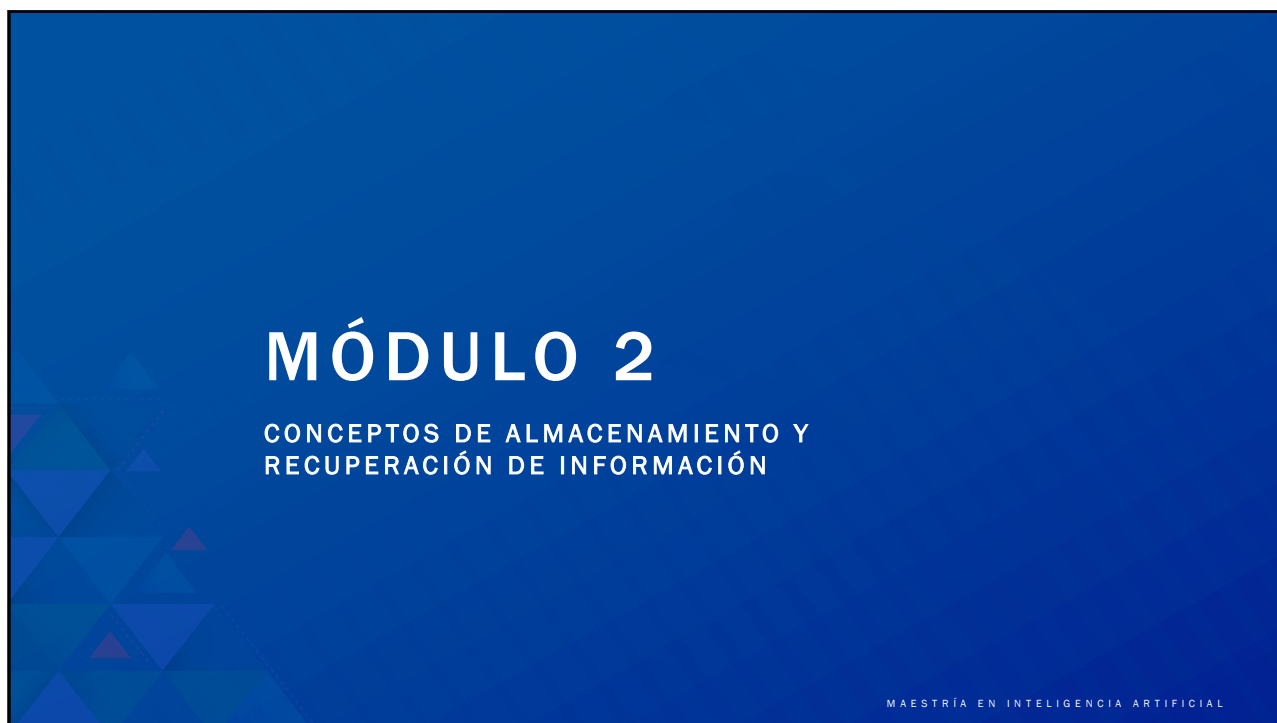




1



2

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>



MAESTRÍA EN INTELIGENCIA ARTIFICIAL

3

Análisis numérico

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/>



MAESTRÍA EN INTELIGENCIA ARTIFICIAL

4

dataframe

PANDAS

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

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

Diagram illustrating the structure of a DataFrame with rows and 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 Manger
9	Jerry Gergich	Administrator
10	Donna Meagle	Office Manger
11	Craig Middlebrooks	Assistant Office Manager

MAESTRÍA EN INTELIGENCIA ARTIFICIAL


5

Lectura de archivos

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**.

Diagram illustrating the process of reading a CSV file into a 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

MAESTRÍA EN INTELIGENCIA ARTIFICIAL

6

Algunos atributos y métodos

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

	Code	Name	Continent	Region	SurfaceArea	IndepYear	Population
0	ABW	Aruba	North America	Caribbean	193.0	NaN	103000
1	AFG	Afghanistan	Asia	Southern and Central Asia	652090.0	1919.0	22720000
2	AGO	Angola	Africa	Central Africa	1246700.0	1975.0	12878000
3	AIA	Anguilla	North America	Caribbean	96.0	NaN	8000
4	ALB	Albania	Europe	Southern Europe	28748.0	1912.0	3401200

MAESTRÍA EN INTELIGENCIA ARTIFICIAL

7

Estadísticas descriptivas

DESCRIBE()

```
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'].quantile(.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**

MAESTRÍA EN INTELIGENCIA ARTIFICIAL

8



Estadísticas descriptivas

DESCRIBE()

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 (nunique)
- El valor más frecuente (top) y su frecuencia (freq)

MAESTRÍA EN INTELIGENCIA ARTIFICIAL

9



Agrupamiento

GROUPBY()

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)
```

```
countries.groupby('Continent')['Population'].sum()
```

	Code	Name	Continent	Region	SurfaceArea	IndepYear	Population
0	ABW	Aruba	North America	Caribbean	193.0	NaN	103000
1	AFG	Afghanistan	Asia	Southern and Central Asia	652090.0	1919.0	22720000
2	AGO	Angola	Africa	Central Africa	1246700.0	1975.0	12878000
3	AIA	Anguilla	North America	Caribbean	96.0	NaN	8000
4	ALB	Albania	Europe	Southern Europe	28748.0	1912.0	3401200

Population	
Continent	
Africa	784475000
Asia	3705025700
Europe	730074600
North America	482993000
Oceania	30401150
South America	345780000

MAESTRÍA EN INTELIGENCIA ARTIFICIAL

10

Agrupamiento GROUPBY()

Se puede agrupar en **varios niveles** pasando más de una columna como parámetro.

El resultado tendrá un índice jerárquico
(**MultiIndex**)

```
countries.groupby(['Continent', 'Region'])
[['Population', 'GNP']].mean()
```

		Population	GNP
Continent	Region		
Africa	Central Africa	10628000.00	3659.78
	Eastern Africa	12999947.37	3680.26
	Northern Africa	24752285.71	34838.57
	Southern Africa	9377200.00	25386.20
	Western Africa	13039529.41	6277.12
	Eastern Asia	188416000.00	690610.62
Asia	Middle East	10465594.44	37625.56
	Southeast Asia	47140090.91	58422.09

Con **agg()** se pueden aplicar varias funciones al mismo tiempo, incluso distintas por columna:

```
countries.groupby(['Continent', 'Region']).agg({
    'Population': ['mean', 'sum', 'max'],
    'GNP': ['mean', 'min']})
```

		Population			GNP	
		mean	sum	max	mean	min
Continent	Region					
Africa	Central Africa	10628000.00	95652000	51654000	3659.78	6.00
	Eastern Africa	12999947.37	246999000	62565000	3680.26	0.00
	Northern Africa	24752285.71	173266000	68470000	34838.57	60.00
	Southern Africa	9377200.00	46886000	40377000	25386.20	1061.00
	Western Africa	13039529.41	221672000	111506000	6277.12	0.00
	Eastern Asia	188416000.00	1507328000	127758000	690610.62	1043.00
Asia	Middle East	10465594.44	188380700	66591000	37625.56	1813.00

MAESTRÍA EN INTELIGENCIA ARTIFICIAL

11

Formatos ANCHO Y LARGO

Ancho: Cada variable medida en diferentes condiciones o momentos se guarda en columnas separadas.

Es decir, un mismo sujeto (ejemplo: un alumno) aparece en una sola fila y sus mediciones en **columnas distintas**.

Se usa mucho para reportes porque es más legible a simple vista.

	Alumno	Enero	Febrero	Marzo
0	Ana	85	88	90
1	Luis	90	92	94
2	María	78	80	82

```
df_long = df_wide.melt(id_vars='Alumno',
    var_name='Mes',
    value_name='Calificacion')
```

← pivot()
melt() →

Largo: Las mediciones se guardan en filas adicionales en lugar de columnas.

Un mismo sujeto aparece en varias filas, cada una con una combinación de identificador + condición + valor.

Es más flexible para algunos análisis estadísticos y visualizaciones porque cada observación queda registrada como una **fila única**.

	Alumno	Mes	Calificacion
0	Ana	Enero	85
1	Luis	Enero	90
2	María	Enero	78
3	Ana	Febrero	88
4	Luis	Febrero	92
5	María	Febrero	80
6	Ana	Marzo	90
7	Luis	Marzo	94
8	María	Marzo	82

```
df_wide = df_long.pivot(index='Alumno',
    columns='Mes',
    values='Calificacion')
    .reset_index()
```

MAESTRÍA EN INTELIGENCIA ARTIFICIAL

12

API plot()

PANDAS

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

Incluso se ha creado [PyViz.org](https://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()
```



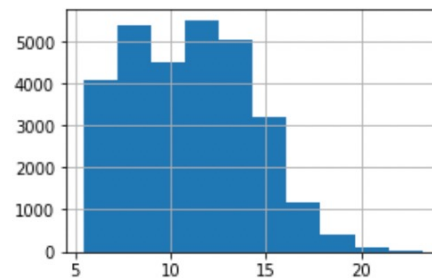
MAESTRÍA EN INTELIGENCIA ARTIFICIAL

13

Histogramas

API PLOT()

- 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()
```

MAESTRÍA EN INTELIGENCIA ARTIFICIAL

14



Derechos Reservados 2025 | Tecnológico de Monterrey |
Prohibida la reproducción total o parcial de esta obra sin
expresa autorización del Tecnológico de Monterrey.