Pandas

The Pandas library is a Python open-source library that offers high-performance, user-friendly data structures and powerful data analysis tools. Pandas is one of the most important tools in the Data Science toolkit, mainly due to its simplicity and scalability across a wide range of data types.

Why is pandas important in Data Science?

Data Wrangling: Pandas makes data cleaning and manipulation easier, so analysts and data scientists can spend more time working with their data.

Data Analysis: Pandas comes with built-in functions to aggregate, plot, and abstract high-level data.

Interoperability: Pandas works well with other libraries and is highly compatible with data from various sources.

Speed: Pandas is based on NumPy and is written in Python and Cython. It is relatively fast and is suitable for real-time data analysis in both business and academia.

Custom Operations: Pandas supports the implementation of custom operations and functions in datasets and includes features for handling missing data and aggregating data effectively.

How to Use pandas:

```
import pandas as pd

# Load a CSV file into a DataFrame
df = pd.read_csv('data.csv')
```

note(If you have a large DataFrame with many rows, Pandas will only return the first 5 rows, and the last 5 rows)

we can returen the max number of rows in different wanys

1-

```
import pandas as pd
print(pd.options.display.max_rows)
```

2-

```
df = pd.read_csv(‘file.csv’)
 Rows =df.shape[0]
```

To display the head

```
print(df.head())
```

but the first 5 rows of the DataFrame

to show all the rows

```
pd.set_option('display.max_rows', None)
```

To display the tail

```
Print(df.tail())
```

but the last 5 rows of the DataFrame

pandas help to dealing with each column separately can involve a variety of operations such as data transformation, analysis, cleaning, and visualization

like

Calculate the mean of the Age column

```
average_age = df['column name'].mean()
```

Convert to uppercase

```
df['City'] = df['column name'].str.upper()
```

Add a new column

```
df['Age_Next_Year'] = df['Age'] + 1
```

Get all entries where Age is above 30

```
older_than_30 = df[df['Age'] > 30]
```

When analyzing large sets of data, one of the most effective initial steps is to organize the data into meaningful groups. This process, known as **grouping**, helps us understand patterns, compare subsets of data, and make more informed decisions based on the similarities and differences between these groups. For example, in a business context like a restaurant, grouping data by items on the menu allows us to see which items are most popular, which are least ordered, and how items compare in terms of sales. This information is crucial for inventory management, pricing strategies, and marketing campaigns.

pandas, a powerful data manipulation library in Python, provides a straightforward way to group data using the `.groupby()` method. This method enables us to easily segment data into groups based on one or more columns and then perform operations on these groups. Let's delve into how this works with a practical example, focusing on a dataset from a Chipotle restaurant.
Example

```
import pandas as pd
item_orders = chipo.groupby('item_name').quantity.sum()
```