



كلية العلوم
والتقنيات - مراكش
FACULTE DES SCIENCES
ET TECHNIQUES - MARRAKECH

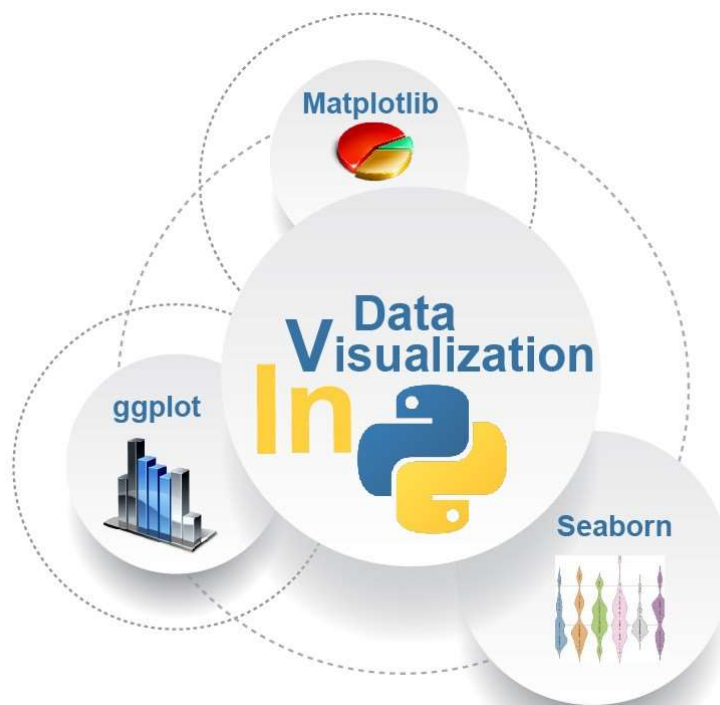


PYTHON FUNDAMENTALS: A COMPREHENSIVE TRAINING

(https://github.com/binatnadata/event2-python-ERME_CLUB/tree/main).

Python labs -By Binatna Data-

- Sessions of December 2023
- Session of 31/12/2023



(https://github.com/binatnadata/event2-python-ERME_CLUB/tree/main).

Exercise 1

1. Load the csv file named "energies_over_time.csv"
2. print the HEAD , the TAIL
3. Print this informations: columns names , dataframe size , dtype , chape , dataframe information
4. rename the column name "Energy" by "E"
5. check for duplacted rows , nan rows , if exist drop em
6. Create a line chart using Matplotlib to visualize the trend in solar panel energy production over the given dates . Label the x-axis as "Date" and the y-axis as "Energy Production (kWh)". Choose an appropriate title for the chart.

```
In [1]: # Import necessary Libraries
import pandas as pd
```

```
In [2]: # 1. Load the csv file named "energies_over_time.csv"
df = pd.read_csv("energies_over_time.csv")
```

```
In [3]: # 2. Print the HEAD and the TAIL
print("HEAD:")
print(df.head())
print("\nTAIL:")
print(df.tail())
```

HEAD:

	Date	Energy
0	1/2017	300
1	2/2017	200
2	3/2017	350
3	4/2017	180
4	5/2017	200

TAIL:

	Date	Energy
77	8/2023	2400
78	9/2023	2430
79	10/2023	2500
80	11/2023	2500
81	12/2023	2500

In [4]: *# 3. Print information about the dataframe*

```
print("\nColumn names:", df.columns)
print("Dataframe size:", df.size)
print("Data types:", df.dtypes)
print("Shape:", df.shape)
print("\nDataframe information:")
df.info()
```

Column names: Index(['Date', 'Energy'], dtype='object')

Dataframe size: 164

Data types: Date object

Energy int64

dtype: object

Shape: (82, 2)

Dataframe information:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 82 entries, 0 to 81

Data columns (total 2 columns):

#	Column	Non-Null Count	Dtype
---	--------	----------------	-------

---	-----	-----	-----
-----	-------	-------	-------

0	Date	82 non-null	object
---	------	-------------	--------

1	Energy	82 non-null	int64
---	--------	-------------	-------

dtypes: int64(1), object(1)

memory usage: 1.4+ KB

In [5]: *# 4. Rename the column name "energy" to "E"*

```
df.rename(columns={"Energy": "E"}, inplace=True)
```

In [6]: *# 6. Check for duplicated rows and NaN rows, if they exist, drop them*

```
print("\nChecking for duplicated rows:")
print("Number of duplicated rows:", df.duplicated().sum())
print("\nChecking for NaN values:")
print("Number of NaN rows:")
print(df.isna().sum())
df.drop_duplicates(inplace=True)
df.dropna(inplace=True)
print("\nDropped duplicated rows and NaN rows.")
```

Checking for duplicated rows:

Number of duplicated rows: 0

Checking for NaN values:

Number of NaN rows:

Date 0

E 0

dtype: int64

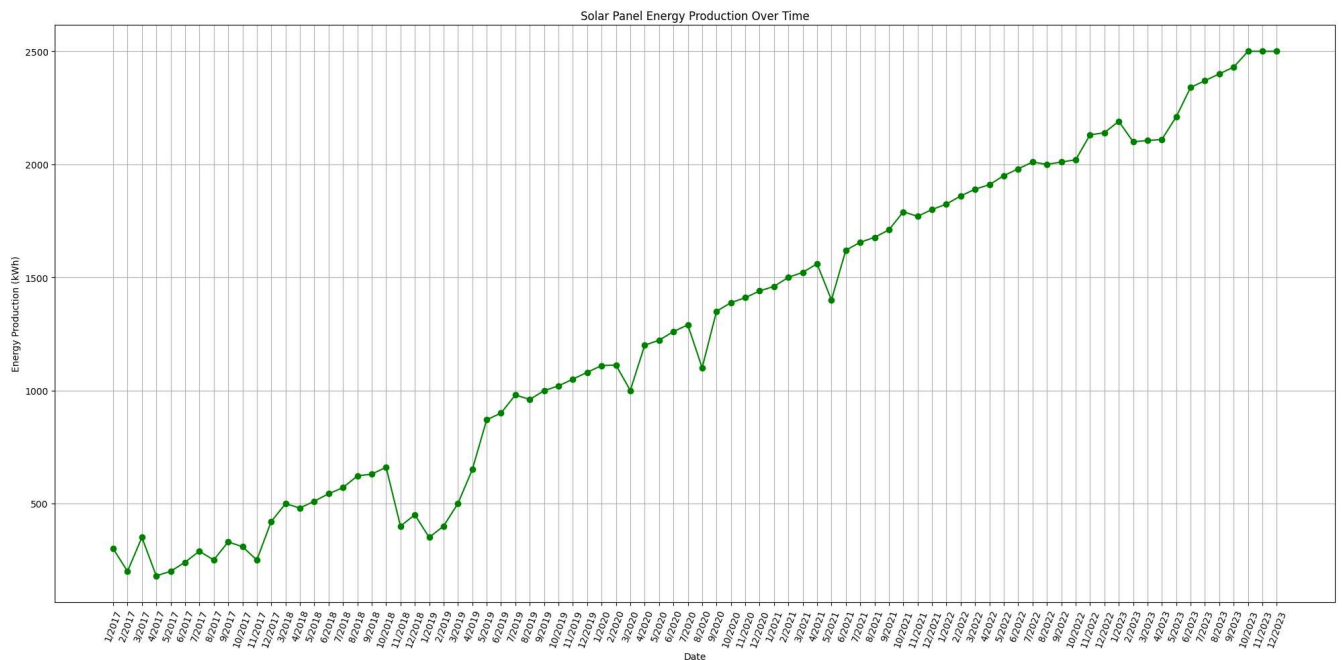
Dropped duplicated rows and NaN rows.

```
In [7]: import matplotlib.pyplot as plt

# create lists
dates = df["Date"]
energy_production = df["E"]

# Plotting the line chart
plt.figure(figsize=(20, 10))
plt.plot(dates, energy_production, marker='o', linestyle='-', color='g')
plt.title('Solar Panel Energy Production Over Time')
plt.xlabel('Date')
plt.ylabel('Energy Production (kWh)')
plt.xticks(rotation=70)
plt.grid(True)
plt.tight_layout()

# Show the plot
plt.show()
```



Exercise 2

1. Create new column named Year generated from the column Date .
2. Groupe by Year and calculate the sum of energies for each year, name the result as df2 .
3. Print the df2 dataframe (select only the column "E").
4. Use df2 to create a bar chart to compare the energy production on different Years . Each bar should represent a specific Year . Label the x-axis as "Date" and the y-axis as "Energy Production (kWh)". Provide a suitable title for the chart . Then Save the plot as a png image.
5. Create a pie chart instead of a bar chart , Then Save the plot as a png image.

```
In [8]: # create new column YEAR using split method
df['Year'] = df['Date'].str.split('/').str[1].astype(int)
```

```
In [9]: # create new column YEAR using datetime function
df['Year2'] = pd.to_datetime(df['Date'], format='%m/%Y').dt.year
```

```
In [10]: # printing dataframe data types
df.dtypes
```

```
Out[10]: Date      object
E          int64
Year       int32
Year2      int64
dtype: object
```

```
In [11]: # group by Year and sum the energies for each year, and select only the "E" column
df2 = df.groupby("Year").sum("E")["E"]
```

```
In [12]: # print df2
df2
```

```
Out[12]:
```

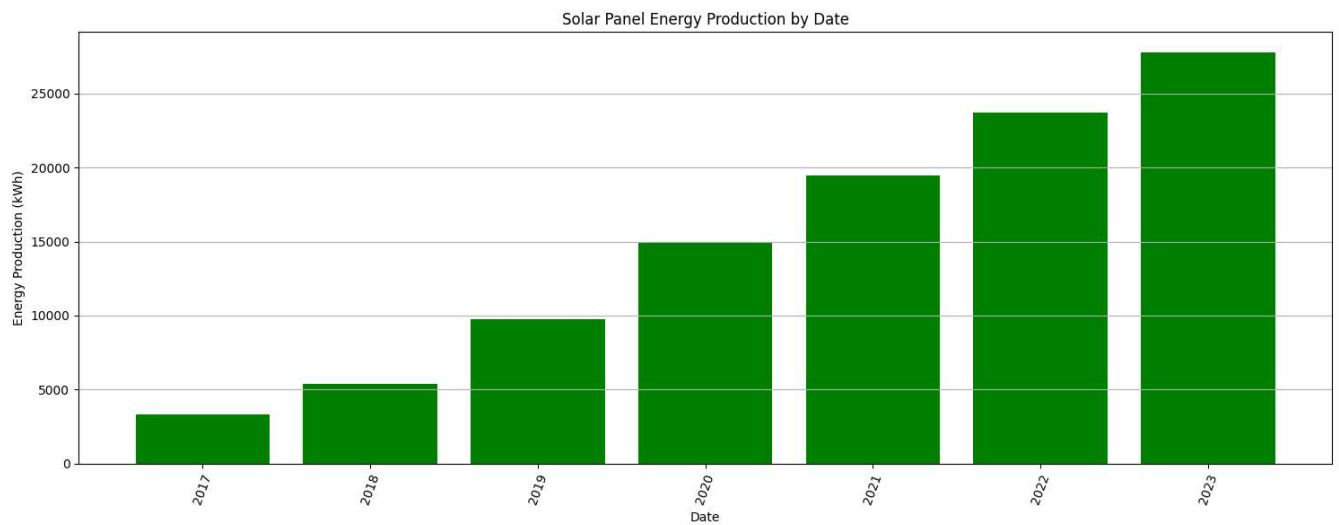
	E
Year	
2017	3318
2018	5364
2019	9759
2020	14882
2021	19464
2022	23724
2023	27755

```
In [13]: # Bar chart
import matplotlib.pyplot as plt

# create lists
dates = df2.index
energy_production = df2["E"]

# Plotting the bar chart
plt.figure(figsize=(15, 6))
plt.bar(dates, energy_production, color='green')
plt.title('Solar Panel Energy Production by Date')
plt.xlabel('Date')
plt.ylabel('Energy Production (kWh)')
plt.xticks(rotation=70)
plt.grid(axis='y')
plt.tight_layout()

# Save the plot as a png image
plt.savefig('my_bar_chart.png')
# Show the plot
plt.show()
```



```
In [14]: # Pie chart
import matplotlib.pyplot as plt

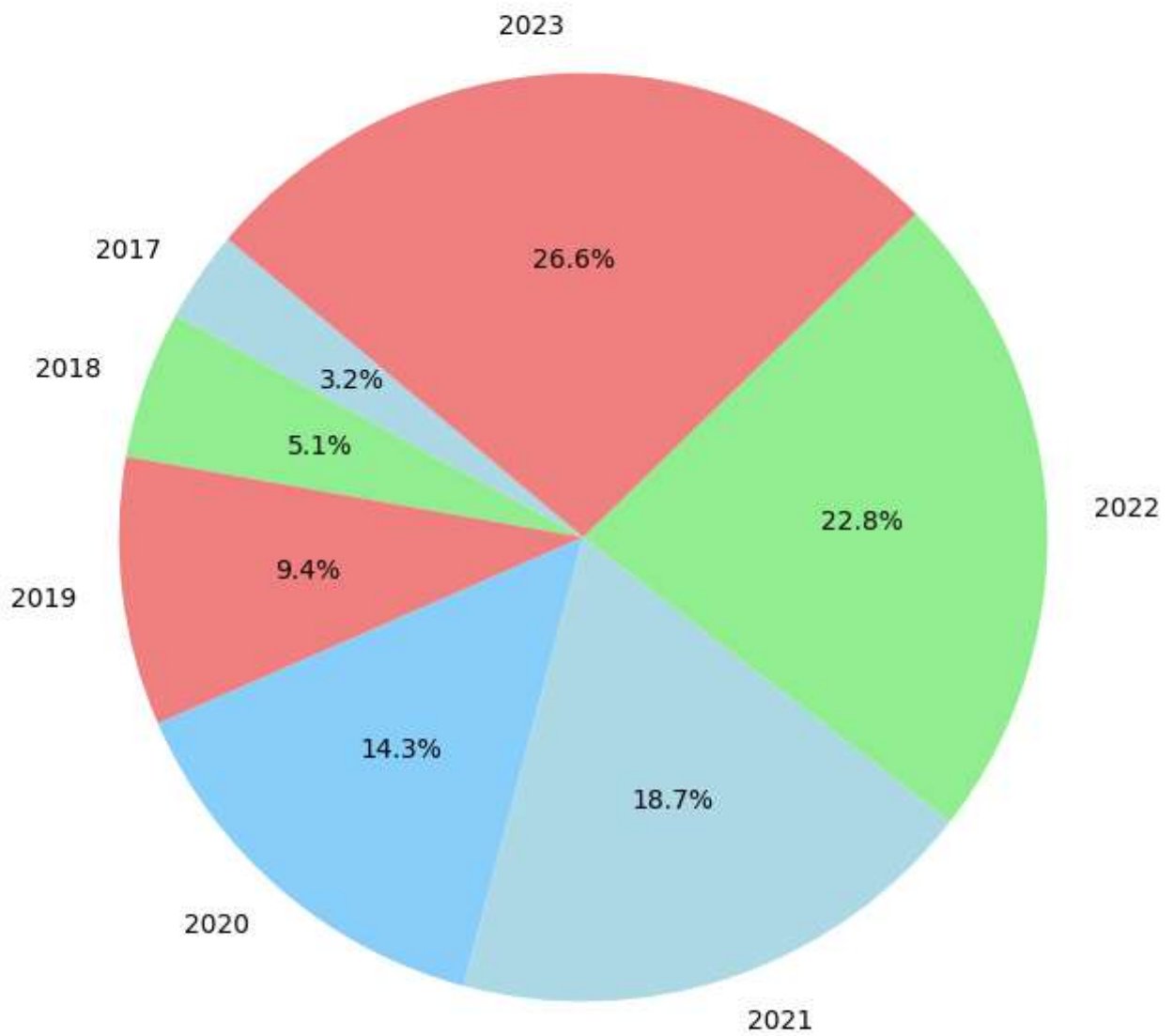
# Assuming you have a DataFrame df2 with columns 'Date' and 'E'
# create lists
dates = df2.index
energy_production = df2["E"]

# Plotting the pie chart
plt.figure(figsize=(8, 8))
plt.pie(energy_production, labels=dates, autopct='%1.1f%%', startangle=140, colors=['lightblue', 'lightgreen', 'lightcoral'])
plt.title('Solar Panel Energy Production by Date')

# Save the plot as a png image
plt.savefig('my_pie_chart.png')

# Show the plot
plt.show()
```

Solar Panel Energy Production by Date



Python Labs (C) 2023 [BinatnaData](https://www.linkedin.com/company/100193879/)
(<https://www.linkedin.com/company/100193879/>) Corporation



(<https://www.linkedin.com/company/100193879/>)