PYTHON FOR DATA ANALYSIS FUNDAMENTALS

1. Importing a CSV file to Google Colab via code

```
[1] Start coding or generate with AI.
```

```
from google.colab import files
upload=files.upload()
print(upload)
```

```
Choose files retail_sales_dataset.csv
• retail_sales_dataset.csv(text/csv) - 51673 bytes, last modified: 15/04/2025 - 100% done
Saving retail_sales_dataset.csv to retail_sales_dataset.csv
{'retail_sales_dataset.csv': b'Transaction ID,Date,Customer ID,Gender,Age,Product Category,Quantity,Price per Unit,Total Amount\r\n1,2023-11-24,
```

2. Importing Pandas and NumPy for data analysis and creating a path for the dataset.

```
[3] import pandas as pd
    import numpy as np
```

```
[4] path="/content/retail_sales_dataset.csv"
```

3. Checking if data is loaded successfully while limiting it to 10 rows.

```
[54] df=pd.read_csv(path)
     df.head(10)
```

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30
3	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500
4	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100
5	6	2023-04-25	CUST006	Female	45	Beauty	1	30	30

4. Investigating the properties of the dataset given.

```
[8] df.shape
```

```
(1000, 9)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Transaction ID    1000 non-null   int64
 1   Date              1000 non-null   object
 2   Customer ID       1000 non-null   object
 3   Gender            1000 non-null   object
 4   Age               1000 non-null   int64
 5   Product Category  1000 non-null   object
 6   Quantity          1000 non-null   int64
 7   Price per Unit    1000 non-null   int64
 8   Total Amount      1000 non-null   int64
dtypes: int64(5), object(4)
memory usage: 70.4+ KB
```

5. Investigating the properties of the dataset given.

```
[10] df.dtypes
```

	0
Transaction ID	int64
Date	object
Customer ID	object
Gender	object
Age	int64
Product Category	object
Quantity	int64
Price per Unit	int64
Total Amount	int64

dtype: object

```
[11] df.columns
```

```
Index(['Transaction ID', 'Date', 'Customer ID', 'Gender', 'Age',
       'Product Category', 'Quantity', 'Price per Unit', 'Total Amount'],
      dtype='object')
```

6. Investigating the properties of the dataset given, focusing on nulls, and describe statement.

```
[12] df.isnull().sum()
```

Show hidden output

```
df['Age'].describe()
```

	Age
count	1000.00000
mean	41.39200
std	13.68143
min	18.00000
25%	29.00000
50%	42.00000
75%	53.00000
max	64.00000

dtype: float64

7. Checking for any duplicates in the dataset.

```
[15] df.duplicated()
```

Show hidden output

```
[17] df.drop_duplicates()
```

Show hidden output

```
[18] df['Transaction ID'].duplicated().sum()
```

```
np.int64(0)
```

8. Syntax for age groups

```
[24] def categorize_age(age):
       if age < 12:
         return 'Child'
       elif age < 18:
         return 'Teen'
       elif age < 65:
         return 'Adult'
       else:
         return 'Senior'

     df['AgeGroup'] = df['Age'].apply(categorize_age)

     display(df)
```

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount	AgeGroup
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150	Adult
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000	Adult
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30	Adult
3	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500	Adult

9. Extracting the day of the week to analyse the best performing day of the week.

```
[46] import pandas as pd

     df['Date'] = pd.to_datetime(df['Date'])

     df['day of the week'] = df['Date'].dt.day_name()

     display(df)
```

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount	AgeGroup	Day of the Week	day_of_month	day of the week
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150	Adult	Friday	24	Friday
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000	Adult	Monday	27	Monday
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30	Adult	Friday	13	Friday
3	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500	Adult	Sunday	21	Sunday
4	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100	Adult	Saturday	6	Saturday

10. Deleting a duplicated column from the data frame.

```
[47] import pandas as pd
     df.drop(columns=['Day of the Week'], inplace=True)
     print(df)
```

Show hidden output

11. Extracting the day of the month from the date.

```
import pandas as pd

df['Date'] = pd.to_datetime(df['Date'])

df['day_of_month'] = df['Date'].dt.day

display(df)
```

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount	AgeGroup	day_of_month	day of the week
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150	Adult	24	Friday
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000	Adult	27	Monday
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30	Adult	13	Friday
3	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500	Adult	21	Sunday

12. Extracting the month of the year from the date to analyse the best-performing month in the dataset.

```
[49] import pandas as pd

df['Date'] = pd.to_datetime(df['Date'])
df['month'] = df['Date'].dt.month
display(df)
```

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount	AgeGroup	day_of_month	day of the week	month
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150	Adult	24	Friday	11
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000	Adult	27	Monday	2
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30	Adult	13	Friday	1
3	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500	Adult	21	Sunday	5
4	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100	Adult	6	Saturday	5
...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	996	2023-05-16	CUST996	Male	62	Clothing	1	50	50	Adult	16	Tuesday	5

13. Extracting the month and year from the dataset.

```
import pandas as pd
df['Date'] = pd.to_datetime(df['Date'])
df['year_month'] = df['Date'].dt.strftime('%Y-%m')
display(df)
```

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount	AgeGroup	day_of_month	day of the week	month	year_month
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150	Adult	24	Friday	11	2023-11
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000	Adult	27	Monday	2	2023-02
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30	Adult	13	Friday	1	2023-01
3	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500	Adult	21	Sunday	5	2023-05
4	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100	Adult	6	Saturday	5	2023-05
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	996	2023-05-16	CUST996	Male	62	Clothing	1	50	50	Adult	16	Tuesday	5	2023-05

14. Syntax for spending buckets where spending is grouped.

```
[53] def categorize_total_amount(total_amount):
       if total_amount <=99:
         return '0-99'
       elif total_amount <=199:
         return '100-199'
       elif total_amount <=299:
         return '200-299'
       elif total_amount <=499:
         return '300-499'
       else:
         return '500-2000'

     df['Spending_Bucket'] = df['Total Amount'].apply(categorize_total_amount)

     display(df)
```

15. The entire table after analysis has been performed.

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount	AgeGroup	day_of_month	day of the week	month	year_month	Spending_Bucket
0	1	2023-11-24	CUST001	Male	34	Beauty	3	50	150	Adult	24	Friday	11	2023-11	100-199
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500	1000	Adult	27	Monday	2	2023-02	500-2000
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30	30	Adult	13	Friday	1	2023-01	0-99
3	4	2023-05-21	CUST004	Male	37	Clothing	1	500	500	Adult	21	Sunday	5	2023-05	500-2000
4	5	2023-05-06	CUST005	Male	30	Beauty	2	50	100	Adult	6	Saturday	5	2023-05	100-199
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	996	2023-05-16	CUST996	Male	62	Clothing	1	50	50	Adult	16	Tuesday	5	2023-05	0-99
996	997	2023-11-17	CUST997	Male	52	Beauty	3	30	90	Adult	17	Friday	11	2023-11	0-99
997	998	2023-10-29	CUST998	Female	23	Beauty	4	25	100	Adult	29	Sunday	10	2023-10	100-199
998	999	2023-12-05	CUST999	Female	36	Electronics	3	50	150	Adult	5	Tuesday	12	2023-12	100-199
999	1000	2023-04-12	CUST1000	Male	47	Electronics	4	30	120	Adult	12	Wednesday	4	2023-04	100-199