IMPORT LIBRARIES

In [127…
```
import pandas as pd
```

LOAD DATA

In [128…
```
df = pd.read_csv('sales_data_week1_500rows.csv')
```

In [129…
```
df.head()
```

Out[129…

	CustomerID	Name	Age	Product	Purchase_Amount	Purchase Date	Region
0	1000	Steve Davis	51.0	Laptop	NaN	2024-04-20	South
1	1001	Jane Miller	36.0	Tablet	1805.62	2024-12-22	South
2	1002	Bob Smith	46.0	Tablet	168.44	2024-04-20	South
3	1003	Emma Brown	51.0	Smartphone	NaN	2024-01-28	West
4	1004	Sara Miller	50.0	Tablet	267.39	2024-03-15	South

In [130…
```
df.shape
```

Out[130…
```
(500, 7)
```

MISSING VALUES

In [131…
```
df.isnull().sum()
```

Out[131…
```
CustomerID          0
Name                4
Age                21
Product            87
Purchase_Amount    26
Purchase Date       0
Region            102
dtype: int64
```

In [132…
```
# Drop rows with missing 'Name' or 'Product'
df=df.dropna(subset=['Name', 'Product'])
df.head()
```

Out[132...

	CustomerID	Name	Age	Product	Purchase_Amount	Purchase Date	Region
0	1000	Steve Davis	51.0	Laptop	NaN	2024-04-20	South
1	1001	Jane Miller	36.0	Tablet	1805.62	2024-12-22	South
2	1002	Bob Smith	46.0	Tablet	168.44	2024-04-20	South
3	1003	Emma Brown	51.0	Smartphone	NaN	2024-01-28	West
4	1004	Sara Miller	50.0	Tablet	267.39	2024-03-15	South

In [133...
```
df.isnull().sum()
```

Out[133...
```
CustomerID          0
Name                0
Age                17
Product             0
Purchase_Amount    22
Purchase Date       0
Region             80
dtype: int64
```

In [134...
```
df= df.fillna({'Region':'Unknown'})
df.isnull().sum()
```

Out[134...
```
CustomerID          0
Name                0
Age                17
Product             0
Purchase_Amount    22
Purchase Date       0
Region              0
dtype: int64
```

In [135...
```
# Fill missing 'Purchase_Amount' with the mean
df=df.fillna({'Purchase_Amount':df['Purchase_Amount'].mean()})
```

In [136...
```
df.isnull().sum()
```

Out[136...
```
CustomerID          0
Name                0
Age                17
Product             0
Purchase_Amount     0
Purchase Date       0
Region              0
dtype: int64
```

DATA TYPE AND CONVERSION

In [137...
```
# Convert 'Purchase Date' to datetime format
df['Purchase Date'] = pd.to_datetime(df['Purchase Date'], errors='coerce')
```

```
# Create a new column 'Purchase_Year'
df['Purchase_Year'] = df['Purchase Date'].dt.year
df.head()
```

Out[137…

	CustomerID	Name	Age	Product	Purchase_Amount	Purchase Date	Region	Purchas
0	1000	Steve Davis	51.0	Laptop	1057.789098	2024-04-20	South	
1	1001	Jane Miller	36.0	Tablet	1805.620000	2024-12-22	South	
2	1002	Bob Smith	46.0	Tablet	168.440000	2024-04-20	South	
3	1003	Emma Brown	51.0	Smartphone	1057.789098	2024-01-28	West	
4	1004	Sara Miller	50.0	Tablet	267.390000	2024-03-15	South	

COLUMN RENAMING AND FORMATTING

In [138…

```
# Rename columns to lowercase and replace spaces with underscores
df.columns = df.columns.str.lower().str.replace(' ', '_')

# Rename 'purchase_amount' to 'amount_usd'
df.rename(columns={'purchase_amount': 'amount_usd'})
```

Out[138...

	customerid	name	age	product	amount_usd	purchase_date	region	purc
0	1000	Steve Davis	51.0	Laptop	1057.789098	2024-04-20	South	
1	1001	Jane Miller	36.0	Tablet	1805.620000	2024-12-22	South	
2	1002	Bob Smith	46.0	Tablet	168.440000	2024-04-20	South	
3	1003	Emma Brown	51.0	Smartphone	1057.789098	2024-01-28	West	
4	1004	Sara Miller	50.0	Tablet	267.390000	2024-03-15	South	
...	...	...	...	...	...	...	...	
492	1492	Sara Johnson	33.0	Tablet	966.200000	2024-07-20	East	
493	1493	Emma Davis	36.0	Smartphone	317.660000	2024-10-19	North	
495	1495	Tom Wilson	32.0	Smartphone	1304.230000	2024-07-24	South	
496	1496	Sara Miller	59.0	Tablet	672.670000	2024-04-20	East	
498	1498	Steve Ali	NaN	Headphones	114.320000	2024-09-17	West	

410 rows × 8 columns

In [139...

```
# Rename 'purchase_amount' to 'amount_usd'
df.rename(columns={'purchase_amount': 'amount_usd'}, inplace=True)
```

In [140...

```
df.head()
```

Out[140...

	customerid	name	age	product	amount_usd	purchase_date	region	purchase_
0	1000	Steve Davis	51.0	Laptop	1057.789098	2024-04-20	South	
1	1001	Jane Miller	36.0	Tablet	1805.620000	2024-12-22	South	
2	1002	Bob Smith	46.0	Tablet	168.440000	2024-04-20	South	
3	1003	Emma Brown	51.0	Smartphone	1057.789098	2024-01-28	West	
4	1004	Sara Miller	50.0	Tablet	267.390000	2024-03-15	South	

DATA FILTERING AND SORTING

In [141…
```
# Filter the DataFrame
df_filtered = df[df['amount_usd'] > 1000]


# Display the first few rows of the filtered DataFrame
display(df_filtered.head())
```

	customerid	name	age	product	amount_usd	purchase_date	region	purchase_y
0	1000	Steve Davis	51.0	Laptop	1057.789098	2024-04-20	South	20
1	1001	Jane Miller	36.0	Tablet	1805.620000	2024-12-22	South	20
3	1003	Emma Brown	51.0	Smartphone	1057.789098	2024-01-28	West	20
7	1007	Linda Davis	25.0	Tablet	1486.850000	2024-09-21	East	20
8	1008	Alice Lee	46.0	Smartphone	1272.280000	2024-09-06	East	20

In [142…
```
# Sort the filtered DataFrame by Purchase_Amount in descending order
df_sorted = df_filtered.sort_values(by='amount_usd', ascending=False)

# Display the first few rows of the sorted DataFrame
display(df_sorted.head())
```

	customerid	name	age	product	amount_usd	purchase_date	region	purcha
469	1469	Sara Davis	23.0	Smartphone	1991.76	2024-07-22	North	
435	1435	Jane Ali	NaN	Laptop	1987.35	2024-02-09	East	
315	1315	Bob Lee	57.0	Smartphone	1979.56	2024-10-19	Unknown	
258	1258	Emma Brown	22.0	Laptop	1975.28	2024-01-13	West	
310	1310	Linda Lee	50.0	Laptop	1974.03	2024-08-20	West	

DATA AGGREAGATION

In [143…
```
# Group the sorted DataFrame by 'Region'
region_grouped = df_sorted.groupby('region')

# Calculate total purchases per region
total_purchases = region_grouped.size()

# Calculate average purchase amount per region
average_purchase_amount = region_grouped['amount_usd'].mean()
```

```
# Combine the results into a new DataFrame
result_df = pd.DataFrame({'Total_Purchases': total_purchases,
                          'Average_Purchase_Amount': average_purchase_amount})

# Reset the index to make 'Region' a column
result_df = result_df.reset_index()

# Display the resulting DataFrame
display(result_df)
```

	region	Total_Purchases	Average_Purchase_Amount
0	East	53	1427.695970
1	North	54	1465.870472
2	South	34	1414.957835
3	Unknown	49	1430.978832
4	West	46	1470.220633

DATA WRANGLING

In [144…

```
# Create the 'category' column based on 'PPurchase_Amount'
df_sorted['category'] = pd.cut(df_sorted['amount_usd'], bins=[-float('inf'), 500

# Display the first few rows to verify
display(df_sorted.head())
```

	customerid	name	age	product	amount_usd	purchase_date	region	purcha
469	1469	Sara Davis	23.0	Smartphone	1991.76	2024-07-22	North	
435	1435	Jane Ali	NaN	Laptop	1987.35	2024-02-09	East	
315	1315	Bob Lee	57.0	Smartphone	1979.56	2024-10-19	Unknown	
258	1258	Emma Brown	22.0	Laptop	1975.28	2024-01-13	West	
310	1310	Linda Lee	50.0	Laptop	1974.03	2024-08-20	West	

EXPORT DATAFRAME TO CSV FILE

In [146…

```
df.to_csv('Downloads/new_sales_data.csv', index=False)
```

In [ ]:

In [ ]: