Generate import pandas Close

< 1 of 1 > Undo Changes Use code with caution

```
# prompt: import pandas
from google.colab import files

import pandas as pd
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats

# Optional for advanced visualizations
import plotly.express as px
import missingno as msno




# Upload the file from local system
uploaded = files.upload()
```

Choose Files BrandEDA.csv

- **BrandEDA.csv**(text/csv) - 1458 bytes, last modified: 3/3/2023 - 100% done

```
Saving BrandEDA.csv to BrandEDA (2).csv
```

```
# Read the uploaded CSV file into a pandas DataFrame
df = pd.read_csv('BrandEDA.csv')
#checking the data info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 10 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Brand                30 non-null     object 
 1   Item ID              30 non-null     int64  
 2   Ram(Gb)              29 non-null     object 
 3   StoraGe(Gb)          28 non-null     float64
 4   Price(Rs)            30 non-null     int64  
 5   Processor speed(GHz) 30 non-null     float64
 6    Touch               30 non-null     object 
 7   Color                30 non-null     object 
 8   Weight               27 non-null     float64
 9   DisplAy size(Inch)   30 non-null     float64
dtypes: float64(4), int64(2), object(4)
memory usage: 2.5+ KB
```

```
#removing the empty cell in the data
df.dropna(inplace=True)
df.head()
```

	Brand	Item ID	Ram(Gb)	StoraGe(Gb)	Price(Rs)	Processor speed(GHz)	Touch	Color	Weight	DisplAy size(Inch)
0	Apple	32	16	256.0	315000	2.2	No	Space grey	1.83	15.4
1	Apple	8	8	512.0	232000	3.2	No	Dark Black	1.29	13.3
2	Apple	32	16	1024.0	382000	2.3	No	Silver	2.00	16.0
4	Apple	32	4	128.0	92400	1.8	No	Silver	1.35	11.6
5	Apple	4	k	256.0	60000	2.6	No	Silver	1.00	13.3

Next steps: Generate code with df View recommended plots New interactive sheet

```
#1. Data Analysis Using Statistical Methods:What is the average price (Price(Rs)) of laptops for each brand??

# Calculate the average price for each brand
average_price = df.groupby('Brand')['Price(Rs)'].mean()
```

```
# Display the results
print("Average Price of Laptops for Each Brand:")
average_price
```

```
Average Price of Laptops for Each Brand:
```

	Price(Rs)
Brand	
Apple	199554.545455
Dell	137053.846154

dtype: float64

```
# Ensure 'Ram(Gb)' column is in string format before using .str accessor
df["Ram(Gb)"] = df["Ram(Gb)"].astype(str)

# Remove "GB" from the 'Ram(Gb)' column
df["Ram(Gb)"] = df["Ram(Gb)"].str.replace("GB", "", regex=False)

# Convert 'Ram(Gb)' back to numeric, invalid values (e.g., "k") will become NaN
df["Ram(Gb)"] = pd.to_numeric(df["Ram(Gb)"], errors="coerce")

# Now filter for Apple laptops with more than 8GB of RAM and a price greater than 200,000 Rs
apple_laptops = df[(df["Brand"] == "Apple") & (df["Ram(Gb)"] > 8) & (df["Price(Rs)"] > 200000)]

# Display the filtered data
apple_laptops
```

	Brand	Item ID	Ram(Gb)	StoraGe(Gb)	Price(Rs)	Processor speed(GHz)	Touch	Color	Weight	DisplAy size(Inch)
0	Apple	32	16.0	256.0	315000	2.2	No	Space grey	1.83	15.4
2	Apple	32	16.0	1024.0	382000	2.3	No	Silver	2.00	16.0

Next steps: Generate code with apple_laptops | View recommended plots | New interactive sheet

```
#Question 3: How many Dell laptops have a processor speed greater than 2.5 GHz and a weight less than 1.5 kg?
dell_laptops = df[(df["Brand"] == "Dell") & (df["Processor speed(GHz)"] > 2.5) & (df["Weight"] < 1.5)]
dell_laptops
```

	Brand	Item ID	Ram(Gb)	StoraGe(Gb)	Price(Rs)	Processor speed(GHz)	Touch	Color	Weight	DisplAy size(Inch)
18	Dell	8	8.0	256.0	224000	2.8	Yes	Silver	1.2	13.4

```
#4. Data Analysis by Grouping Data
#Question: What is the total storage capacity (StoraGe(Gb)) of laptops grouped by brand?
total_storage_by_brand = df.groupby("Brand")["StoraGe(Gb)"].sum()
total_storage_by_brand
```

	StoraGe(Gb)
Brand	
Apple	3712.0
Dell	7044.0

dtype: float64

```
# 5.Which are the top 5 most expensive laptops in the dataset when sorted in descending order of price?
top_5_most_expensive = df.nlargest(5, "Price(Rs)")
top_5_most_expensive
```

	Brand	Item ID	Ram(Gb)	StoraGe(Gb)	Price(Rs)	Processor speed(GHz)	Touch	Color	Weight	DisplAy size(Inch)
2	Apple	32	16.0	1024.0	382000	2.3	No	Silver	2.00	16.0
17	Dell	26	16.0	1024.0	340000	2.3	No	Silver	2.11	17.0
0	Apple	32	16.0	256.0	315000	2.2	No	Space grey	1.83	15.4
19	Dell	26	16.0	512.0	299000	2.6	No	Black	2.50	15.6
23	Apple	4	8.0	256.0	255000	2.3	No	Silver	1.37	13.3

Next steps: Generate code with `top_5_most_expensive` View recommended plots New interactive sheet

```
#6-7. Data Analysis Using Combination of Sorting, Filtering, and Grouping
```

```
#Question 6: Among laptops with a display size greater than 14 inches, which brand offers the most expensive option?
expensive_laptops = df[df["DisplAy size(Inch)"] > 14].nlargest(1, "Price(Rs)")
expensive_laptops
```

	Brand	Item ID	Ram(Gb)	StoraGe(Gb)	Price(Rs)	Processor speed(GHz)	Touch	Color	Weight	DisplAy size(Inch)
2	Apple	32	16.0	1024.0	382000	2.3	No	Silver	2.0	16.0

```
#Question 7: What is the average processor speed for laptops with a weight less than 2 kg, grouped by brand?
average_speed_by_brand = df[df["Weight"] < 2].groupby("Brand")["Processor speed(GHz)"].mean()
average_speed_by_brand
```

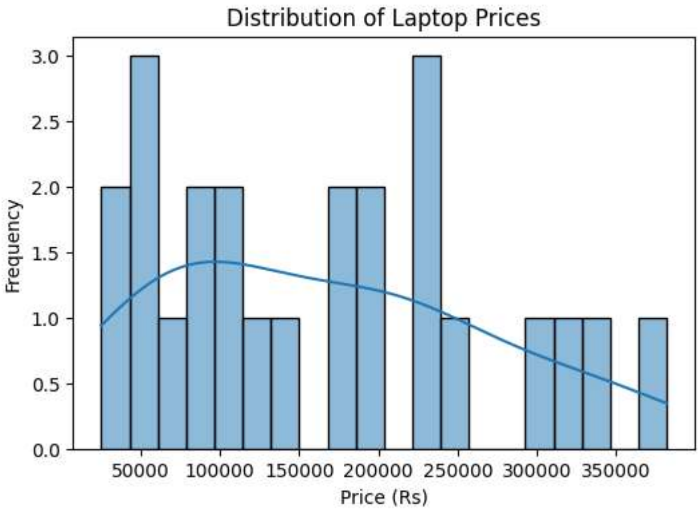
	Processor speed(GHz)
Brand	
Apple	2.070000
Dell	2.205556

dtype: float64

```
#8-10. Data Visualization with Charts
```

```
#Question 8: Visualize the price distribution of laptops using a histogram.
plt.figure(figsize=(6, 4))
sns.histplot(df["Price(Rs)"], bins=20, kde=True)
plt.title("Distribution of Laptop Prices")
plt.xlabel("Price (Rs)")
plt.ylabel("Frequency")
plt
```

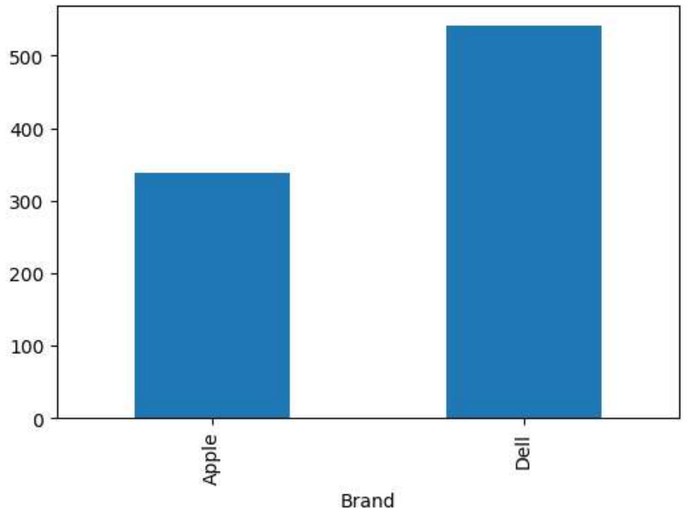
```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.10/dist-packages/matplotlib/pyplot.py'>
```



```
#9Create a bar chart to compare the average storage capacity (StoraGe(Gb)) across different brands.
```

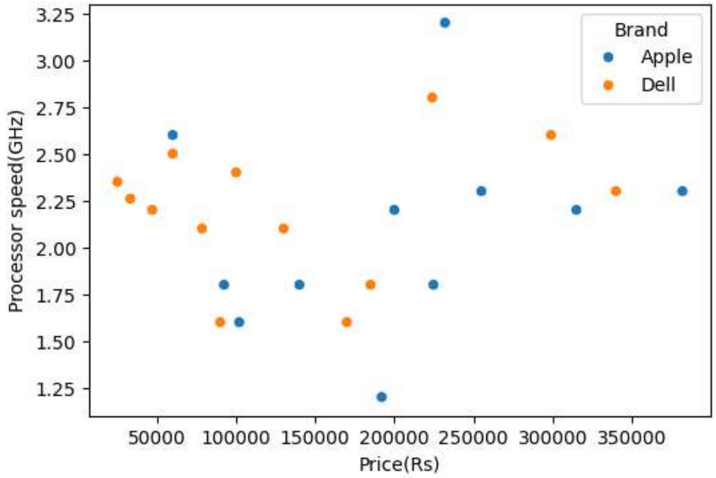
```
plt.figure(figsize=(6, 4))
average_storage_by_brand = df.groupby("Brand")["StoraGe(Gb)"].mean()
average_storage_by_brand.plot(kind="bar")
```

```
<Axes: xlabel='Brand'>
```



```
#10Plot a scatter plot showing the relationship between price (Price(Rs)) and processor speed (Processor speed(GHz)), with different colors f
plt.figure(figsize=(6, 4))
sns.scatterplot(data=df, x="Price(Rs)", y="Processor speed(GHz)", hue="Brand")
plt
```

```
<module 'matplotlib.pyplot' from '/usr/local/lib/python3.10/dist-packages/matplotlib/pyplot.py'>
```



```
#
```