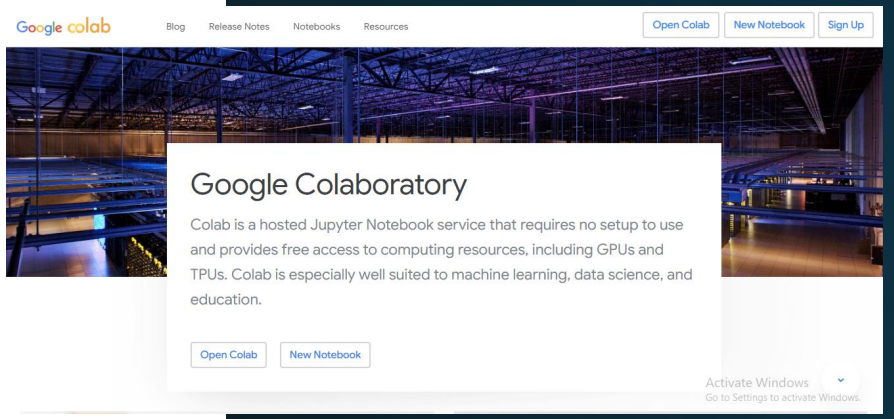
Data Analysis on PHYTON

Step # 1

Sign into gmail account

Step 2

Click on the link
https://colab.google/



Google colab

Blog Release Notes Notebooks Resources

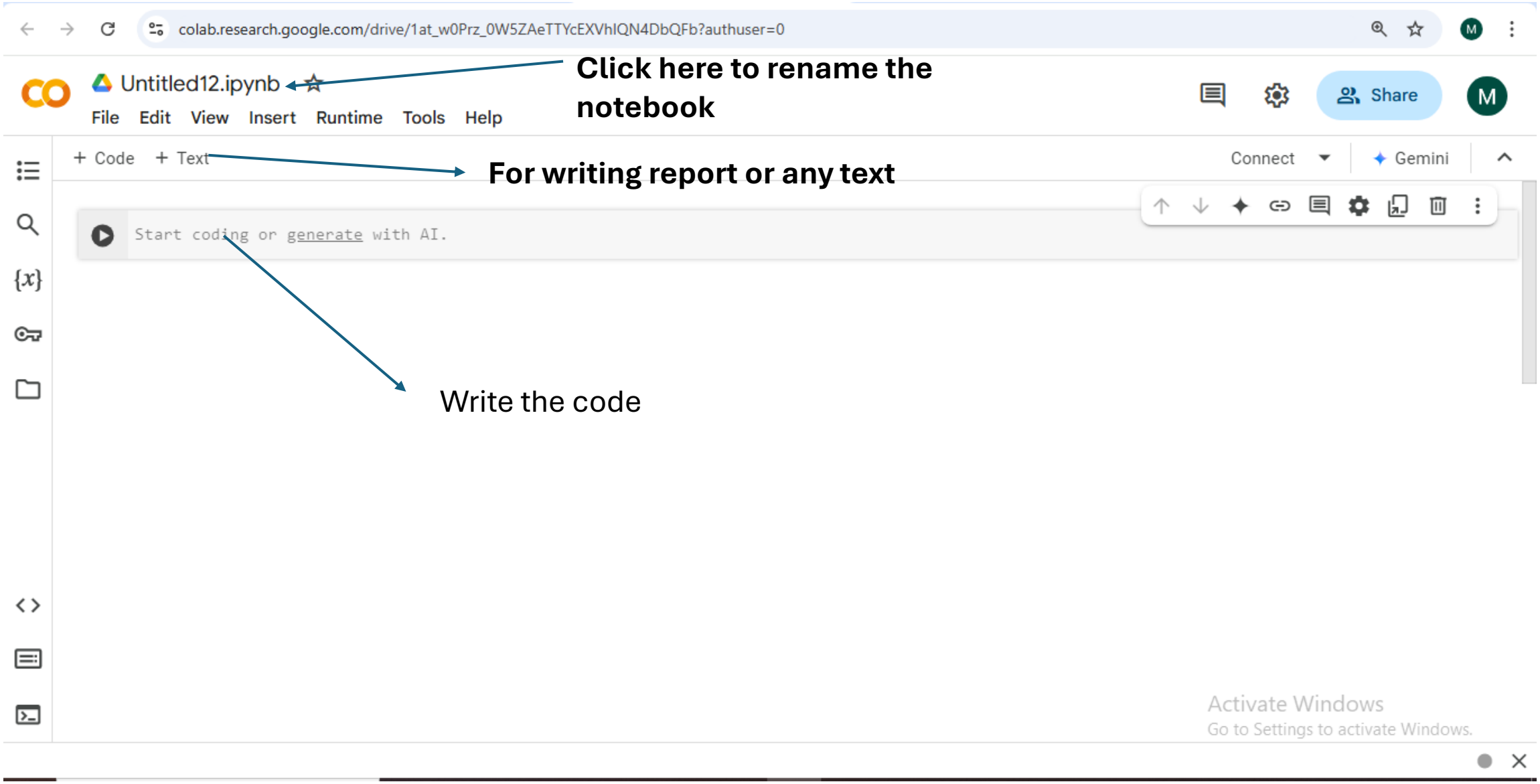
Open Colab New Notebook Sign Up

Google Colaboratory

Colab is a hosted Jupyter Notebook service that requires no setup to use and provides free access to computing resources, including GPUs and TPUs. Colab is especially well suited to machine learning, data science, and education.

Click on this

Open Colab New Notebook


colab.research.google.com/drive/1at_w0Prz_0W5ZAeTTYcEXVhIQN4DbQFb?authuser=0
Untitled12.ipynb
File Edit View Insert Runtime Tools Help
Share
M
+ Code + Text
Connect
Gemini
Start coding or generate with AI.
Click here to rename the notebook
For writing report or any text
Write the code
Activate Windows
Go to Settings to activate Windows.

```
[1] 2+3
```

```
5
```

To run the cell press shift+enter

```
[2] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
```

Import important libraries that will require in data analysis

- **Creating this data frame on PHYTON by using PANDAS**

ID	Gender	Weight	Height	Virus
202	Male	64	182	No
208	Female	56	178	Yes
213	Male	66	177	Yes
221	Male	109	180	No
225	Male	100	156	No
228	Female	59	143	No
232	Male	56	160	No
247	Female	54	165	Yes
248	Male	64	171	No
252	Male	65	150	Yes

Creating DataFrame

```
ID=[202,208,213,221,225,228,232,247,248,252]
Gender=["Male","Female","Male","Male","Male","Female","Male","Female","Male","Male"]
Weight=[64,56,66,109,100,59,56,54,64,65]
Height=[182,178,177,180,156,143,160,165,171,150]
Virus=["No","Yes","Yes","No","No","No","No","Yes","No","Yes"]
df=pd.DataFrame({"ID":ID,"Gender":Gender,"Weight":Weight,"Height":Height,"Virus":Virus})
```

```
[6] print (df)
```

```
    ID  Gender  Weight  Height Virus
0  202    Male      64     182    No
1  208  Female      56     178   Yes
2  213    Male      66     177   Yes
3  221    Male     109     180    No
4  225    Male     100     156    No
5  228  Female      59     143    No
6  232    Male      56     160    No
7  247  Female      54     165   Yes
8  248    Male      64     171    No
9  252    Male      65     150   Yes
```

```
[7] df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   ID      10 non-null     int64
 1   Gender  10 non-null     object
 2   Weight  10 non-null     int64
 3   Height  10 non-null     int64
 4   Virus   10 non-null     object
dtypes: int64(3), object(2)
memory usage: 532.0+ bytes
```

This gives the information about the data

Given data has 10 rows (observation/samples) and 5 columns (variables)

No column has missing value in this dataset

```
[8] df.describe()
```

This gives summary statistics of the numeric variables of the data

	ID	Weight	Height
count	10.000000	10.000000	10.000000
mean	227.600000	69.300000	166.200000
std	17.353834	19.143029	13.595751
min	202.000000	54.000000	143.000000
25%	215.000000	56.750000	157.000000
50%	226.500000	64.000000	168.000000
75%	243.250000	65.750000	177.750000
max	252.000000	109.000000	182.000000

Q_1

$Q_2 = median$

Q_3

```
[25] df['Gender'].value_counts()
```

Gives frequency of each unique value in Gender column

Gender	count
Male	7
Female	3

dtype: int64

```
[26] df['Gender'].unique()
```

Gives number of unique values in Gender column

```
array(['Male', 'Female'], dtype=object)
```

```
[27] df['Gender'].nunique()
```

Counts the number of unique values in Gender column

```
2
```

```
[10] df['Height'].mean()
```

```
166.2
```

This gives the mean of Height column/variable

```
[13] df['Height'].median()
```

```
168.0
```

This gives the median of Height column/variable

Note: in this manner, you can find mean and median of any numeric variable. For example, if we want to find the mean of weights, write df[‘Weight].mean()

```
[18] df['Gender'].mode()
```

	Gender
0	Male

dtype: object

This gives the mode (in object format) of Gender column/variable

```
[19] df['Gender'].mode()[0]
```

'Male'

This will extract the value of mode of gender column/variable

```python
import numpy as np
Q1=np.percentile(df['Height'],25)
Q3=np.percentile(df['Height'],75)
IQR=Q3-Q1
print(Q1)
print(Q3)
print(IQR)
```

```
157.0
177.75
20.75
```

This is syntax of finding Q1, Q3, and IQR of any numeric variable.
In this manner you can find any quartile, percentile and decile of any numeric variable

```
df["Height"].std()
```

```
13.59575097022759
```

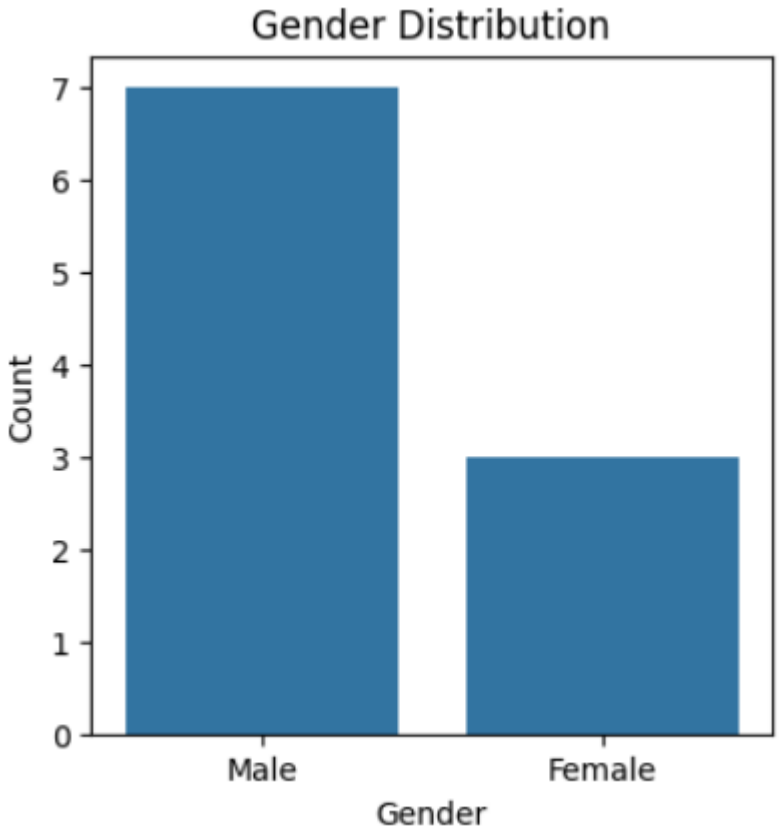
```
[23] df["Height"].var()
```

```
184.84444444444443
```

This is syntax of finding Standard deviation and variance of any numeric variable.

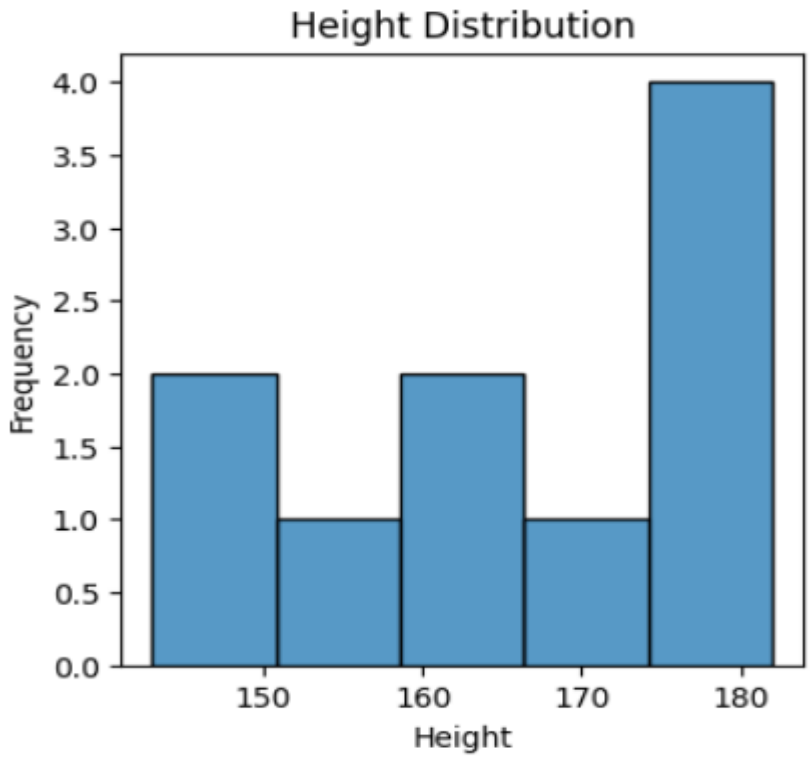
```
[27] plt.figure(figsize=(4, 4))
     sns.countplot(x='Gender', data=df)
     plt.title('Gender Distribution')
     plt.xlabel('Gender')
     plt.ylabel('Count')
     plt.show()
```

This is syntax of plotting bar chart of any categorical variable



```python
plt.figure(figsize=(4, 4))
sns.histplot(df['Height'])
plt.title('Height Distribution')
plt.xlabel('Height')
plt.ylabel('Frequency')
plt.show()
```

This is syntax of plotting histogram of any numeric variable

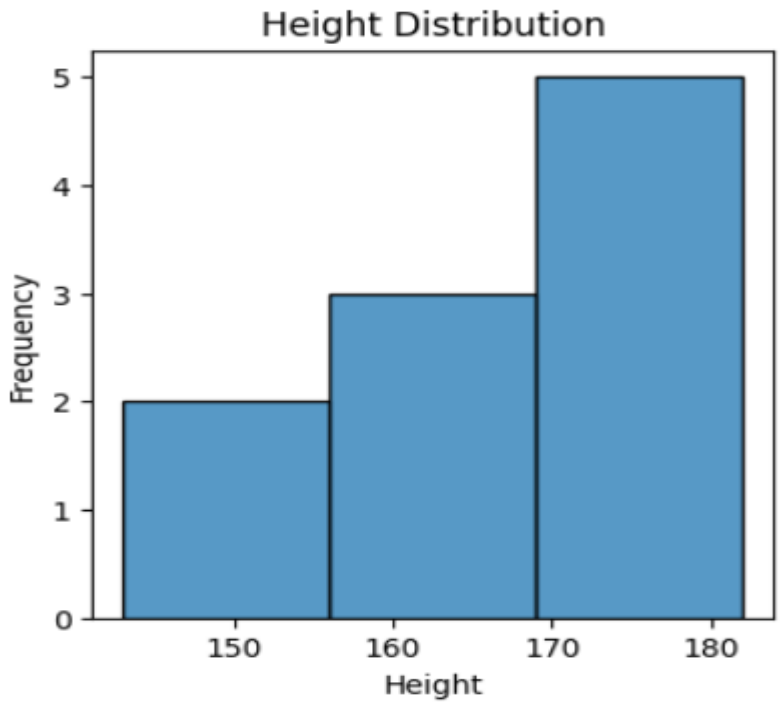


```
[33] plt.figure(figsize=(4, 4))
     sns.histplot(df['Height'], bins=3)
     plt.title('Height Distribution')
     plt.xlabel('Height')
     plt.ylabel('Frequency')
     plt.show()
```

This is syntax of plotting histogram of any numeric variable

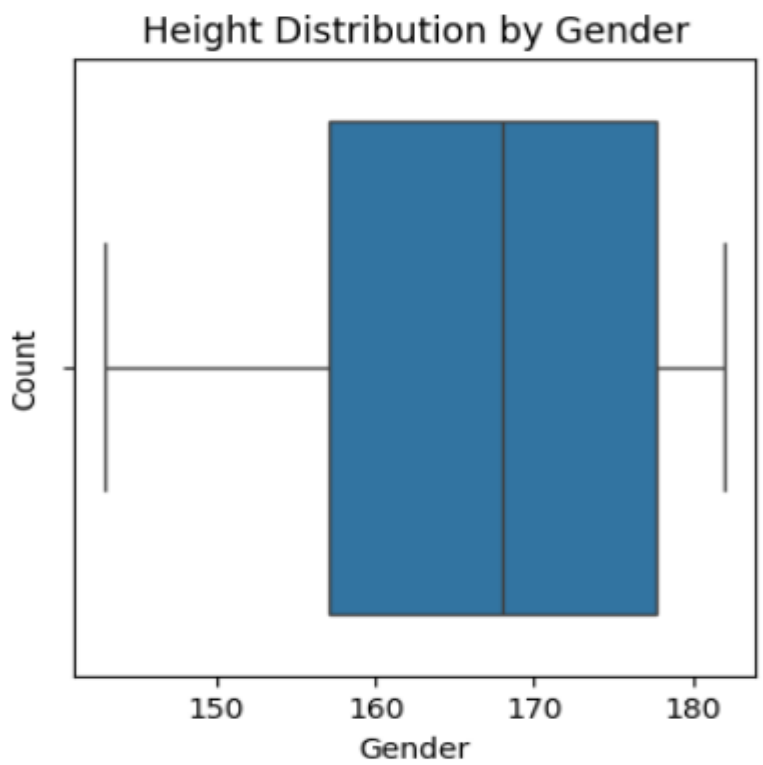
Bins=number of classes

by default, number of classes =5

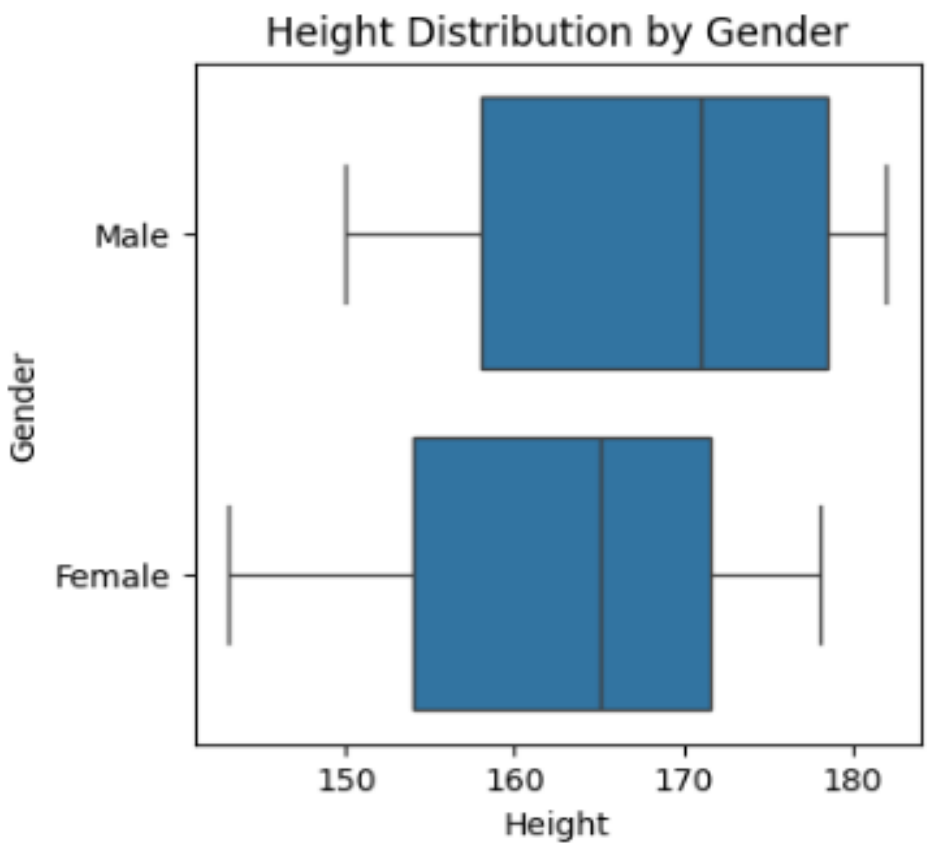


```
[37] plt.figure(figsize=(4, 4))
     sns.boxplot(x='Height', data=df)
     plt.title('Height Distribution by Gender')
     plt.xlabel('Gender')
     plt.ylabel('Count')
     plt.show()
```

This is syntax of plotting box plot of any numeric variable.



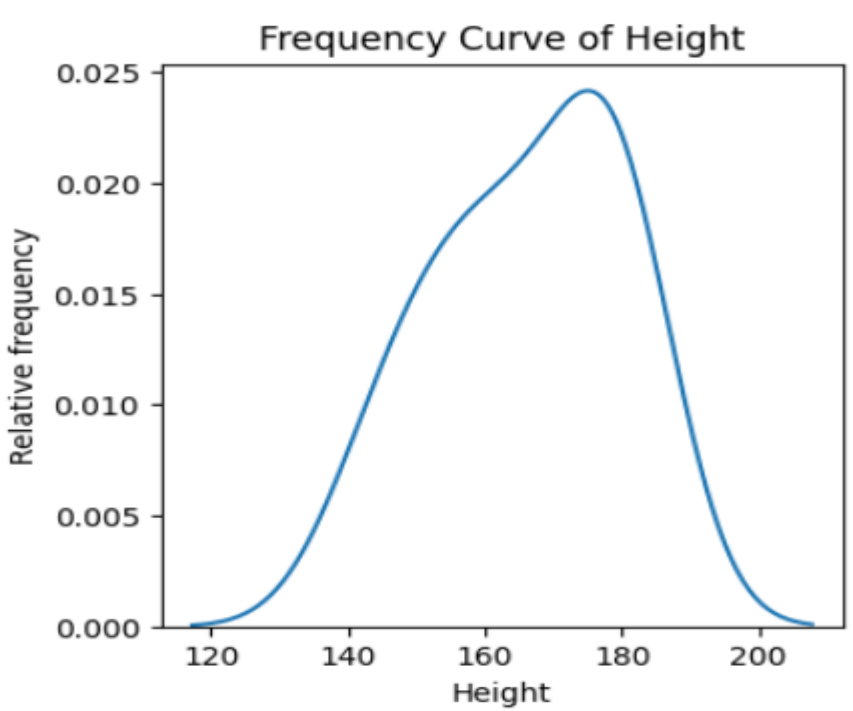
```python
plt.figure(figsize=(4, 4))
sns.boxplot(x='Height',y='Gender', data=df)
plt.title('Height Distribution by Gender')
plt.xlabel('Height')
plt.ylabel('Gender')
plt.show()
```



This is syntax of plotting box plot of 2 variables
in this example two variables are Gender and Height

```python
plt.figure(figsize=(4, 4))
sns.kdeplot(df['Height'])
plt.title('Frequency Curve of Height')
plt.xlabel('Height')
plt.ylabel('Relative frequency')
plt.show()
```

This is syntax of plotting frequency curve of any numeric variable.



```
[57] new_df = df.groupby('Gender')[['Height', 'Weight']].mean()
```

This is syntax of groupby command use it when needed in data analysis

```
new_df
```

	Height	Weight
Gender		
Female	162.0	56.333333
Male	168.0	74.857143

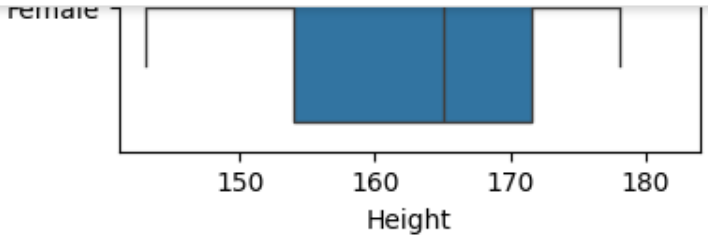
How to upload csv/excel file on PHYTON using Pandas

Data Analysis.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

[43]




```
[57] new_df = df.groupby('Gender')[['Height', 'Weight']].mean()
```

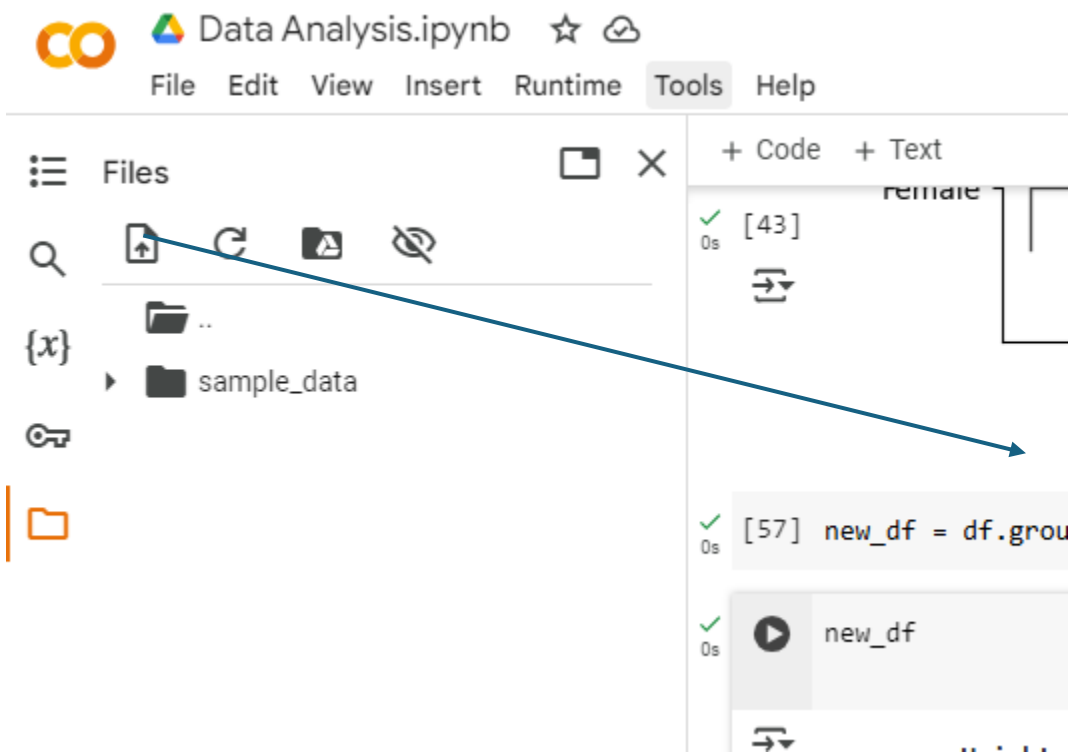
```
new_df
```

Gender	Height	Weight
Female	162.0	56.333333
Male	168.0	74.857143

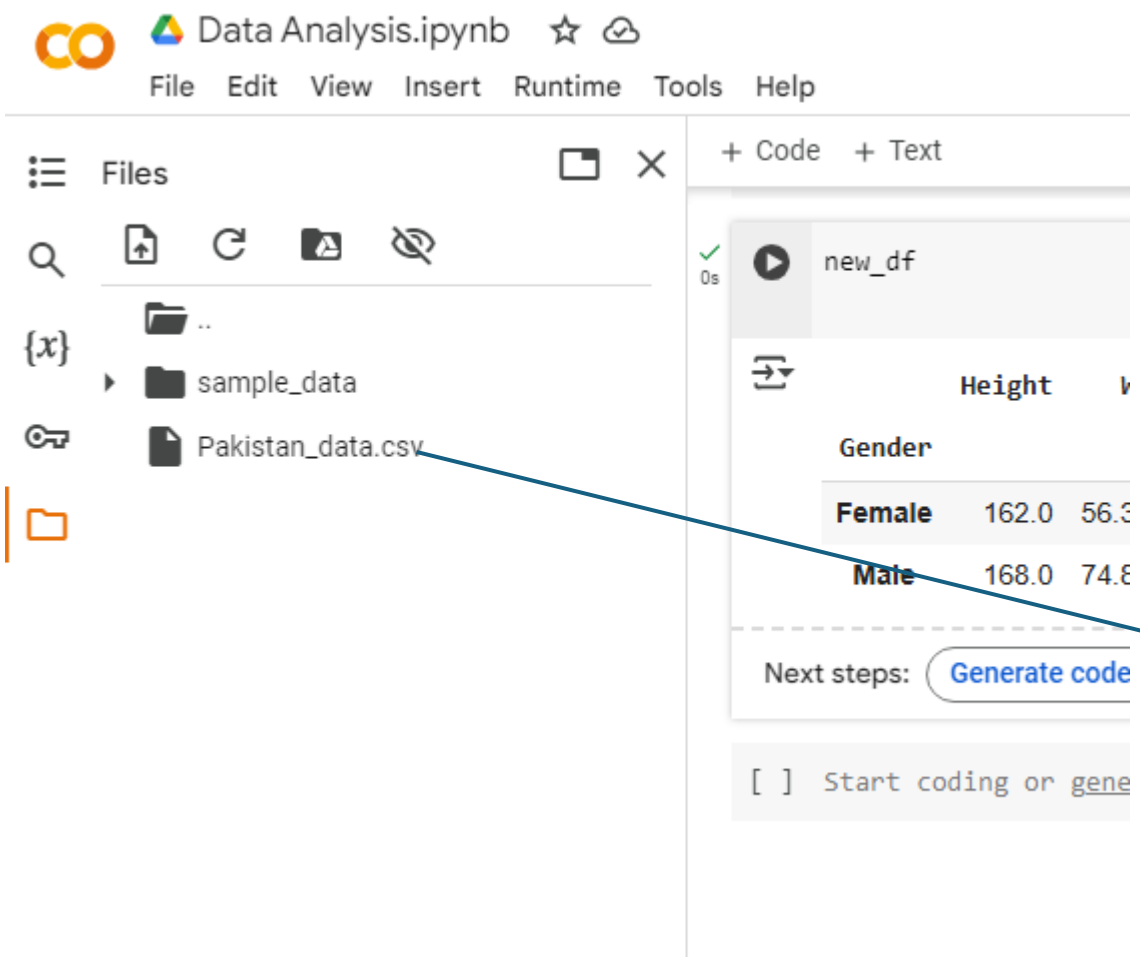
Click here

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

[ ] Start coding or generate with AI.



Click here and upload the file



Your file will appear here

```
[59] data=pd.read_csv("Pakistan_data.csv")
```

data variable save the uploaded dataframe (csv file)

```
[60] data.head()
```

This show the above 5 rows of the data frame

	Year	Qty(MT)	Rs(Crore)	% Share(Qty)	% Share(Rs)	MT per Crore
0	1987-88	794.83	0.51	0.076	0.067	1558.49
1	1988-89	4332.62	2.09	0.478	0.261	2073.02
2	1990-91	2310.84	2.03	0.236	0.220	1138.34
3	1991-92	3208.26	2.33	0.141	0.129	1376.94
4	1992-93	6234.20	3.85	0.429	0.189	1619.27

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

```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37 entries, 0 to 36
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Year          37 non-null     object
 1   Qty(MT)       37 non-null     float64
 2   Rs(Crore)     37 non-null     float64
 3   % Share(Qty)  37 non-null     float64
 4   % Share(Rs)   37 non-null     float64
 5   MT per Crore  37 non-null     float64
dtypes: float64(5), object(1)
memory usage: 1.9+ KB
```

Data has 37 rows and 6 columns

```
[62] data['Year'].isna()
```

	Year
0	False
1	False
2	False
3	False
4	False
5	False
6	False
7	False
8	False
9	False
10	False
11	False
12	False
13	False

This will check the missing values in year column

False indicate no missing value at respective index

```
data['Qty(MT)'].fillna(data['Qty(MT)'].mean(), inplace=True)
```

Syntax of filling missing value of any numeric/categorical column with its mean (for categorical use Mode)

inplace=True will update the column and fill the missing value in original dataframe

Now perform data analysis as perform in the earlier slides

For download your work

File -> Download -> Download.ipynb