

```
import numpy as np
N=np.array([1,2])
S=np.array([1,6])
print("S multiplied by N: ",N*S)
print("S divided by N: ",S/N)
print("S raised to the power of N: ",S**N)
```

S multiplied by N: [1 12]
S divided by N: [1. 3.]
S raised to the power of N: [1 36]

+ Code + Text

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

# Create two arrays
array1 = np.array([1, 2, 3])
array2 = np.array([4, 5, 6])

# Perform addition
result_array = array1 + array2
print(result_array)
```

[5 7 9]

```
import pandas as pd
import numpy as np

# Define an array
```

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

```
import pandas as pd
import numpy as np

# Define an array
data = np.array([1, 2, 3, 4, 5])

# Apply functions
result_sin = np.sin(data)
result_log = np.log(data)
result_log2 = np.log2(data)
result_exp = np.exp(data)
print(result_sin)
print(result_log)
print(result_log2)
print(result_exp)
```

[0.84147098 0.90929743 0.14112001 -0.7568025 -0.95892427]
[0. 0.69314718 1.09861229 1.38629436 1.60943791]
[0. 1. 1.5849625 2. 2.32192809]
[2.71828183 7.3890561 20.08553692 54.59815003 148.4131591]

```
#graphs:
import pandas as pd
import matplotlib.pyplot as plt

# Define the dataset
people = ['kiran', 'arun', 'vijay', 'varun']
age = [25, 30, 35, 40]
height = [145, 151, 165, 173]
weight = [45, 55, 65, 75]

# Create a DataFrame
df = pd.DataFrame({'Name': people, 'Age': age, 'Height': height, 'Weight': weight})

# Scatter plot: Age vs Height
plt.figure(figsize=(8, 6))
plt.scatter(df['Age'], df['Height'], color='blue')
plt.title('Age vs Height')
plt.xlabel('Age')
plt.ylabel('Height')
plt.grid(True)
plt.show()

# Scatter plot: Age vs Weight
plt.figure(figsize=(8, 6))
plt.scatter(df['Age'], df['Weight'], color='red')
plt.title('Age vs Weight')
plt.xlabel('Age')
plt.ylabel('Weight')
plt.grid(True)
plt.show()
```

```
import pandas as pd
import numpy as np

# Create two arrays
array1 = np.array([1, 2, 3])
array2 = np.array([4, 5, 6])

# Perform addition
result_array = array1 + array2
print("The addition of two arrays is",result_array)

The addition of two arrays is [5 7 9]

[31] import pandas as pd
import numpy as np

# Define an array
data = np.array([1, 2, 3, 4, 5])

# Apply functions
result_sin = np.sin(data)
result_log = np.log(data)
result_log2 = np.log2(data)
result_exp = np.exp(data)
print("the sin of data", result_sin)
print("the log of data is",result_log)
print("the log2 of data is",result_log2)
```

```
import pandas as pd

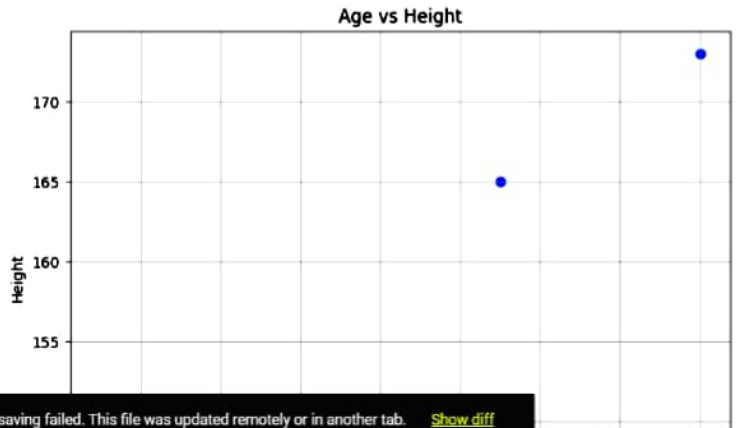
people = ['kiran', 'arun', 'vijay', 'varun']
age = [25, 30, 35, 40]
height = [145, 151, 165, 173]
weight = [45, 55, 65, 75]

# Create a dictionary to hold the data
health_data = {
    'Name': people,
    'Age': age,
    'Height': height,
    'Weight': weight
}

print(health_data)

{'Name': ['kiran', 'arun', 'vijay', 'varun'], 'Age': [25, 30, 35, 40], 'Height': [145, 151, 165, 173], 'Weight': [45, 55, 65, 75]}
```

```
# Scatter plot: Height vs Weight
plt.figure(figsize=(8, 6))
plt.scatter(df['Height'], df['Weight'], color='green')
plt.title('Height vs Weight')
plt.xlabel('Height')
plt.ylabel('Weight')
plt.grid(True)
plt.show()
```



Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

✓ 0s completed at 8:17 PM

▶ `import pandas as pd`

```
people = ['kiran', 'arun', 'vijay', 'varun']  
age = [25, 30, 35, 40]  
height = [145, 151, 165, 173]  
weight = [45, 55, 65, 75]
```

`# Create a dictionary to hold the data`

```
health_data = {  
    'Name': people,  
    'Age': age,  
    'Height': height,  
    'Weight': weight  
}  
print(health_data)
```

⇒ `{'Name': ['kiran', 'arun', 'vijay', 'varun'], 'Age': [25, 30, 35, 40], 'Height': [145, 151, 165, 173], 'Weight': [45, 55, 65, 75]}`

Automatic saving failed. This file was updated remotely or in another tab.

[Show diff](#)

✓ 0s comp

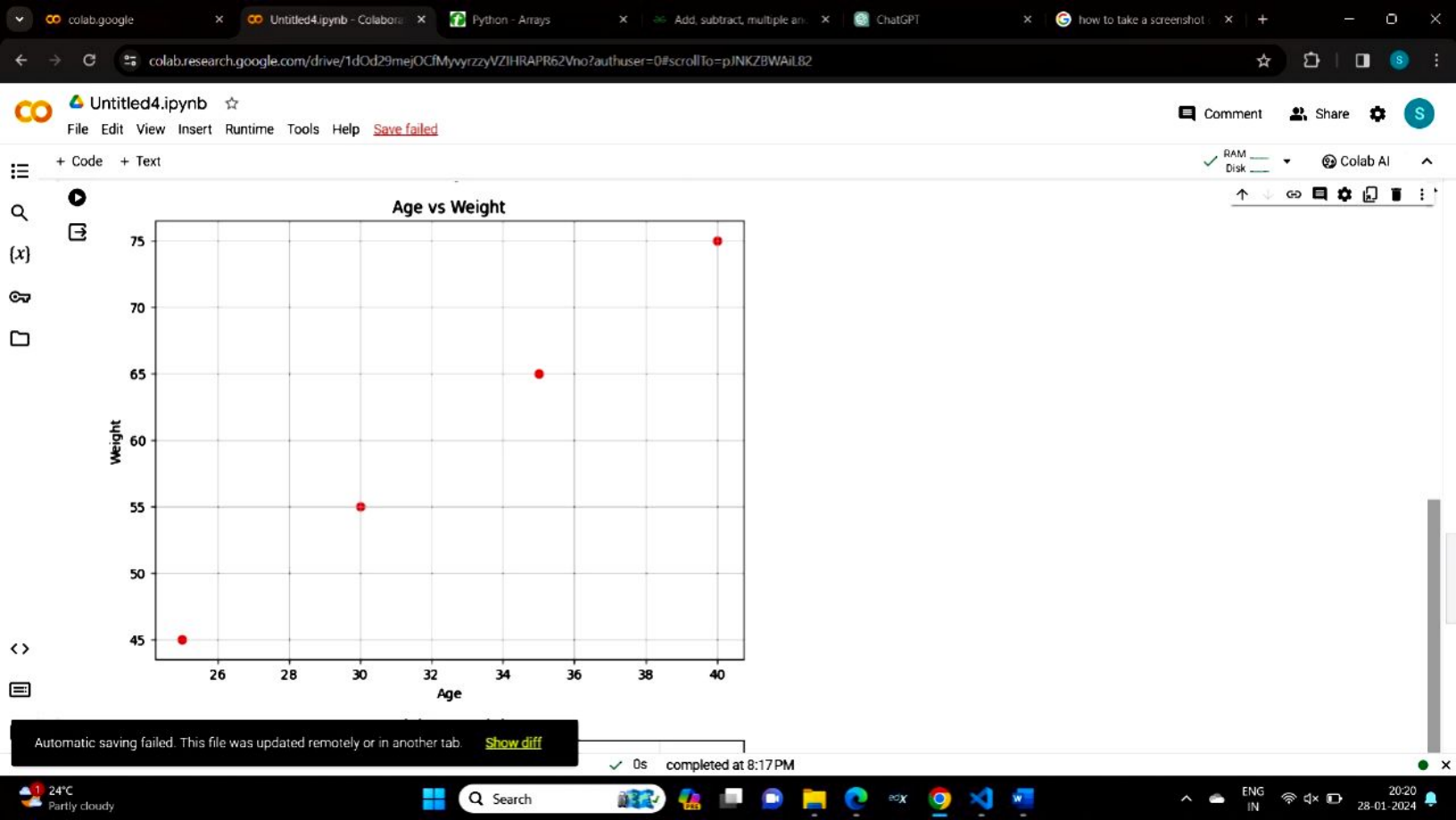
```
graphs:
import pandas as pd
import matplotlib.pyplot as plt

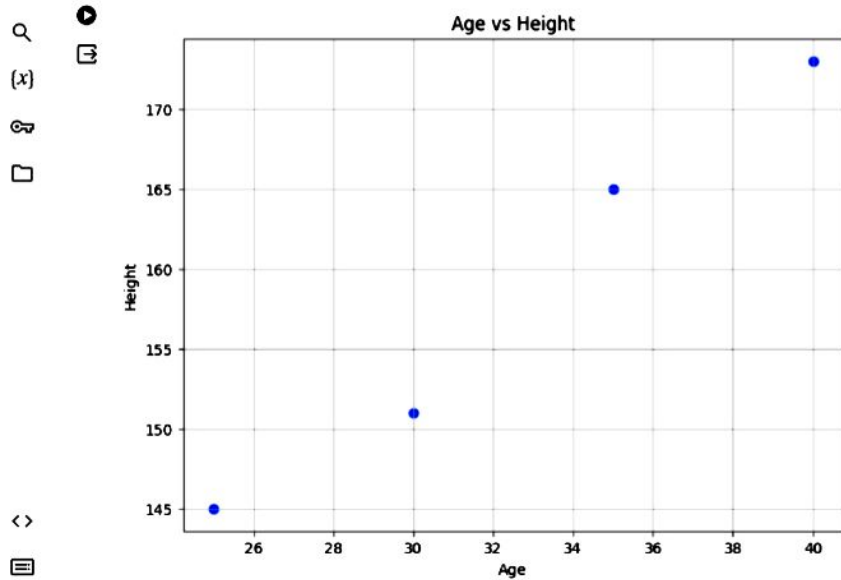
# Define the dataset
people = ['kiran', 'arun', 'vijay', 'varun']
age = [25, 30, 35, 40]
height = [145, 151, 165, 173]
weight = [45, 55, 65, 75]

# Create a DataFrame
df = pd.DataFrame({'Name': people, 'Age': age, 'Height': height, 'Weight': weight})

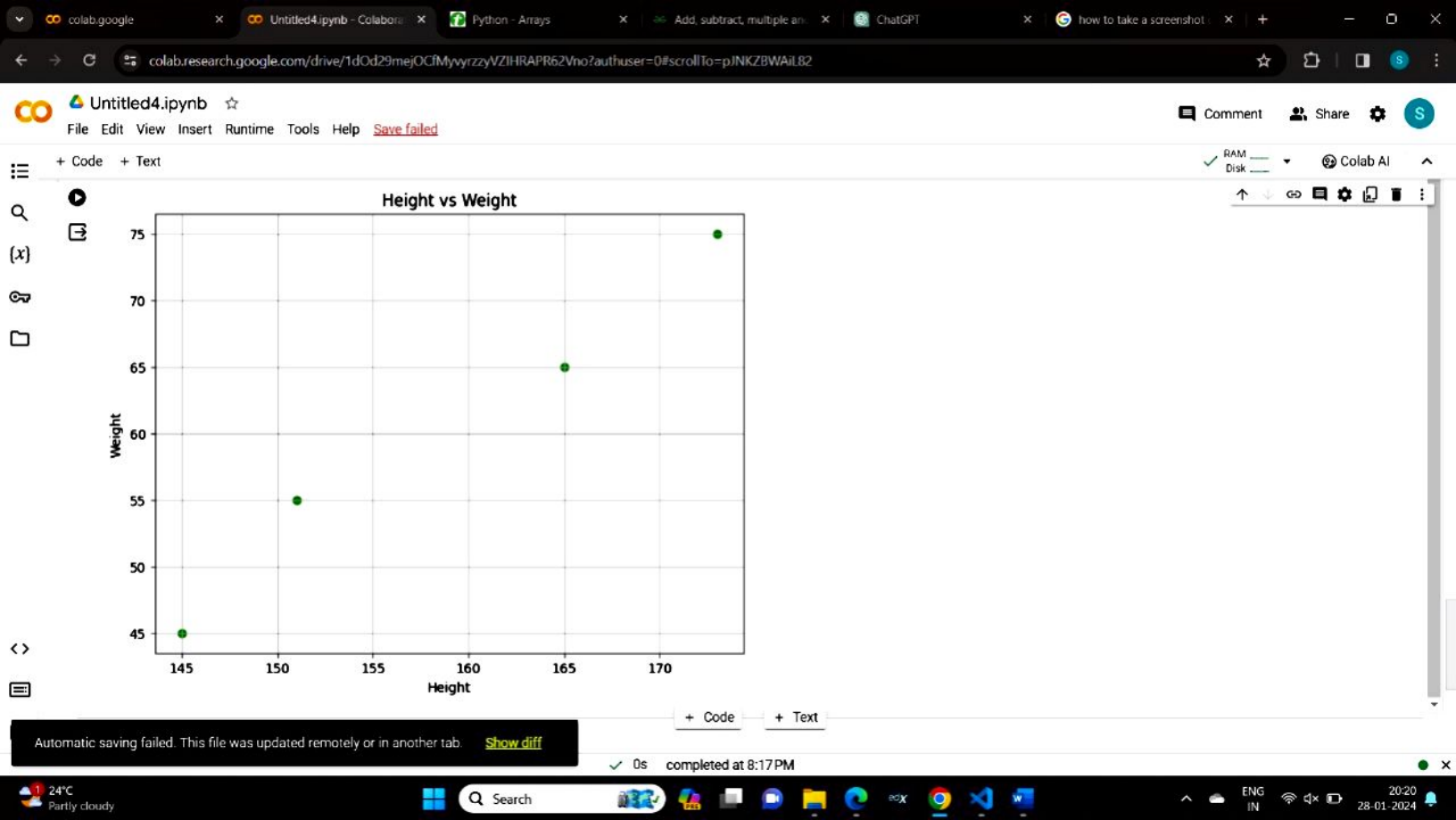
# Scatter plot: Age vs Height
plt.figure(figsize=(8, 6))
plt.scatter(df['Age'], df['Height'], color='blue')
plt.title('Age vs Height')
plt.xlabel('Age')
plt.ylabel('Height')
plt.grid(True)
plt.show()

# Scatter plot: Age vs Weight
plt.figure(figsize=(8, 6))
plt.scatter(df['Age'], df['Weight'], color='red')
plt.title('Age vs Weight')
plt.xlabel('Age')
plt.ylabel('Weight')
plt.grid(True)
```



Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)



```
+ Code + Text
import pandas as pd
import matplotlib.pyplot as plt

# Fix the mismatch in the lengths of lists
people = ['kiran', 'arun', 'vijay', 'varun']
age = [25, 30, 35, 40]
height = [145, 151, 165, 173]
weight = [45, 55, 65, 75]

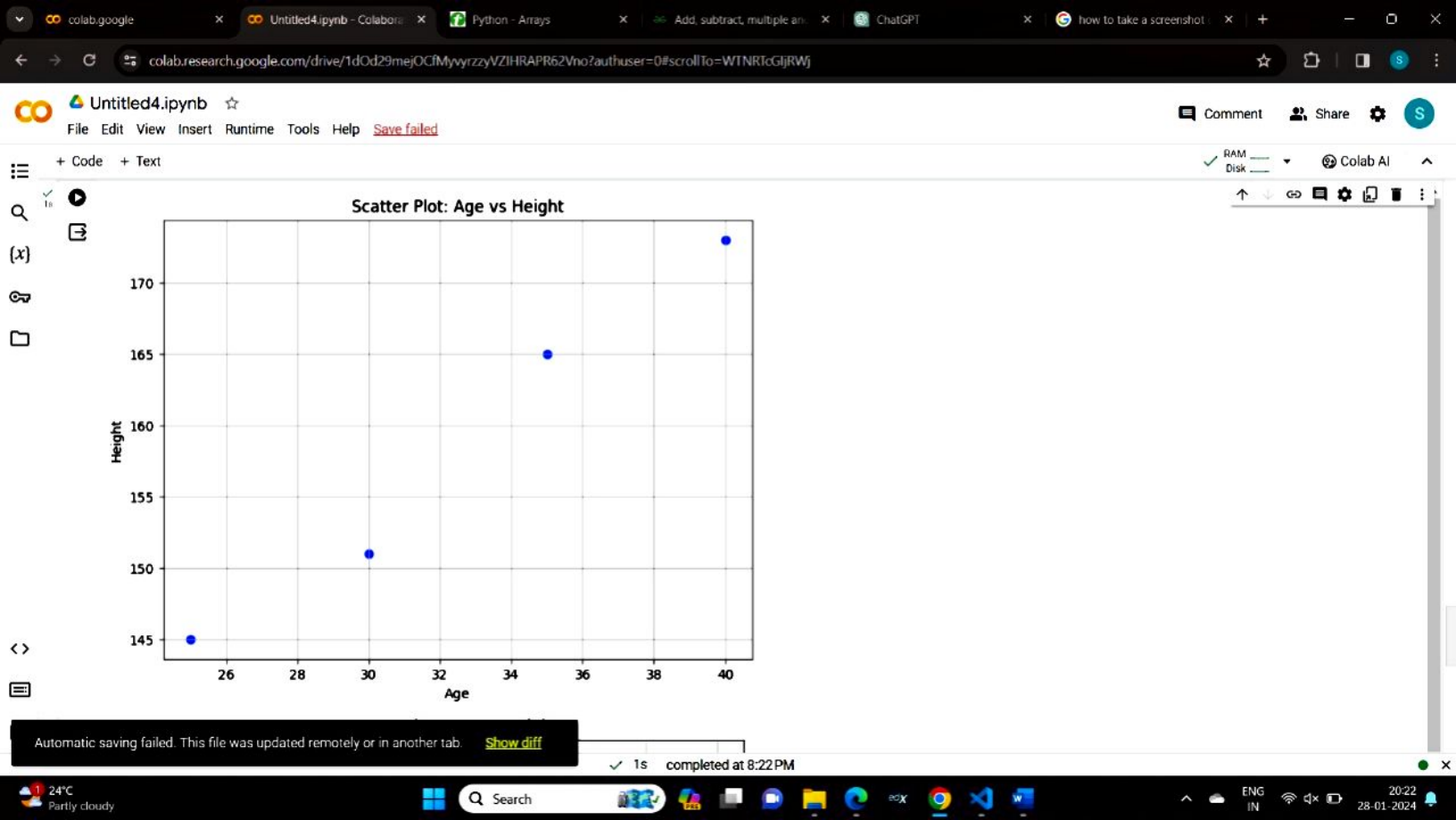
# Create a Dataframe
health_data = {
    'Name': people,
    'Age': age,
    'Height': height,
    'Weight': weight
}

df = pd.DataFrame(health_data)

# Scatter plot: Age vs Height
plt.figure(figsize=(8, 6))
plt.scatter(df['Age'], df['Height'], color='blue')
plt.title('Scatter Plot: Age vs Height')
plt.xlabel('Age')
plt.ylabel('Height')
plt.grid(True)
plt.show()

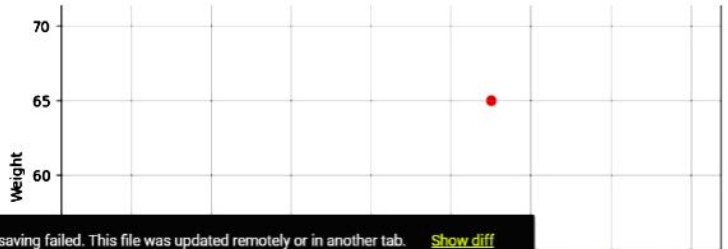
# Scatter plot: Age vs Weight
plt.figure(figsize=(8, 6))
plt.scatter(df['Age'], df['Weight'], color='red')
```

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)



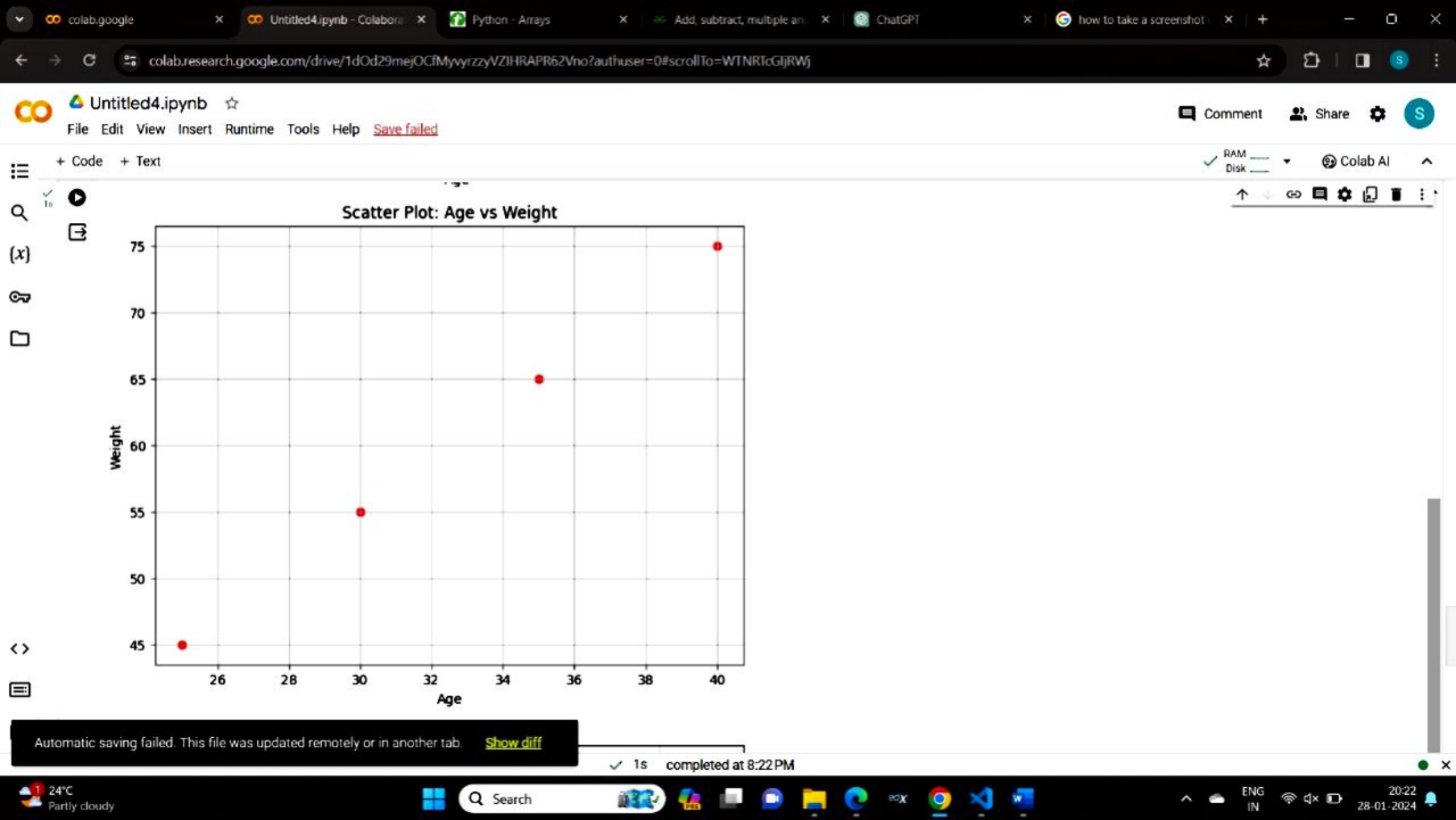
```
# Scatter plot: Age vs Weight
plt.figure(figsize=(8, 6))
plt.scatter(df['Age'], df['Weight'], color='red')
plt.title('Scatter Plot: Age vs Weight')
plt.xlabel('Age')
plt.ylabel('Weight')
plt.grid(True)
plt.show()

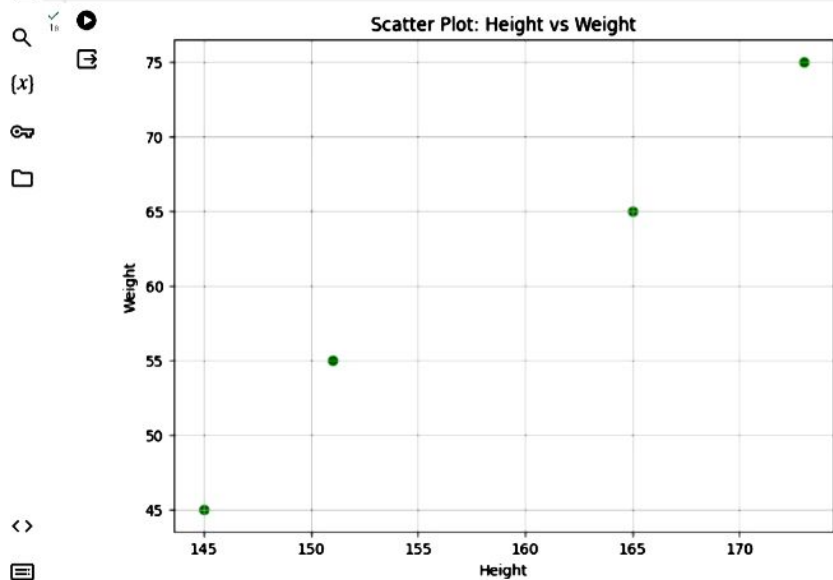
# Scatter plot: Height vs Weight
plt.figure(figsize=(8, 6))
plt.scatter(df['Height'], df['weight'], color='green')
plt.title('Scatter Plot: Height vs Weight')
plt.xlabel('Height')
plt.ylabel('Weight')
plt.grid(True)
plt.show()
```



Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

✓ 1s completed at 8:22 PM





Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

✓ 1s completed at 8:22 PM


```
#Bar chart:|
import pandas as pd
import matplotlib.pyplot as plt

# Fix the mismatch in the lengths of lists
people = ['kiran', 'arun', 'vijay', 'varun']
age = [25, 30, 35, 40]
height = [145, 151, 165, 173]
weight = [45, 55, 65, 75]

# Create a DataFrame
health_data = {
    'Name': people,
    'Age': age,
    'Height': height,
    'Weight': weight
}
df = pd.DataFrame(health_data)

# Bar chart for age
plt.figure(figsize=(8, 6))
plt.bar(df['Name'], df['Age'], color='blue')
plt.title('Bar Chart: Age')
plt.xlabel('Name')
plt.ylabel('Age')
plt.grid(axis='y')
plt.show()

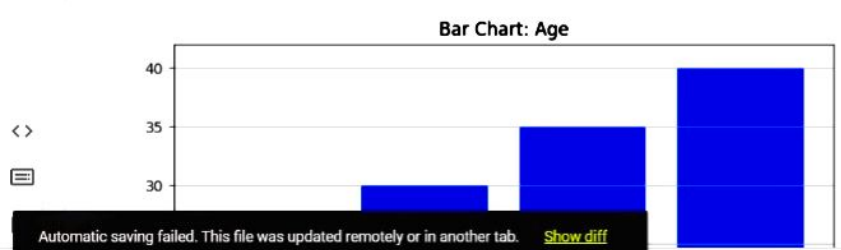
# Bar chart for height
```

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#) 0s completed at 8:24 PM

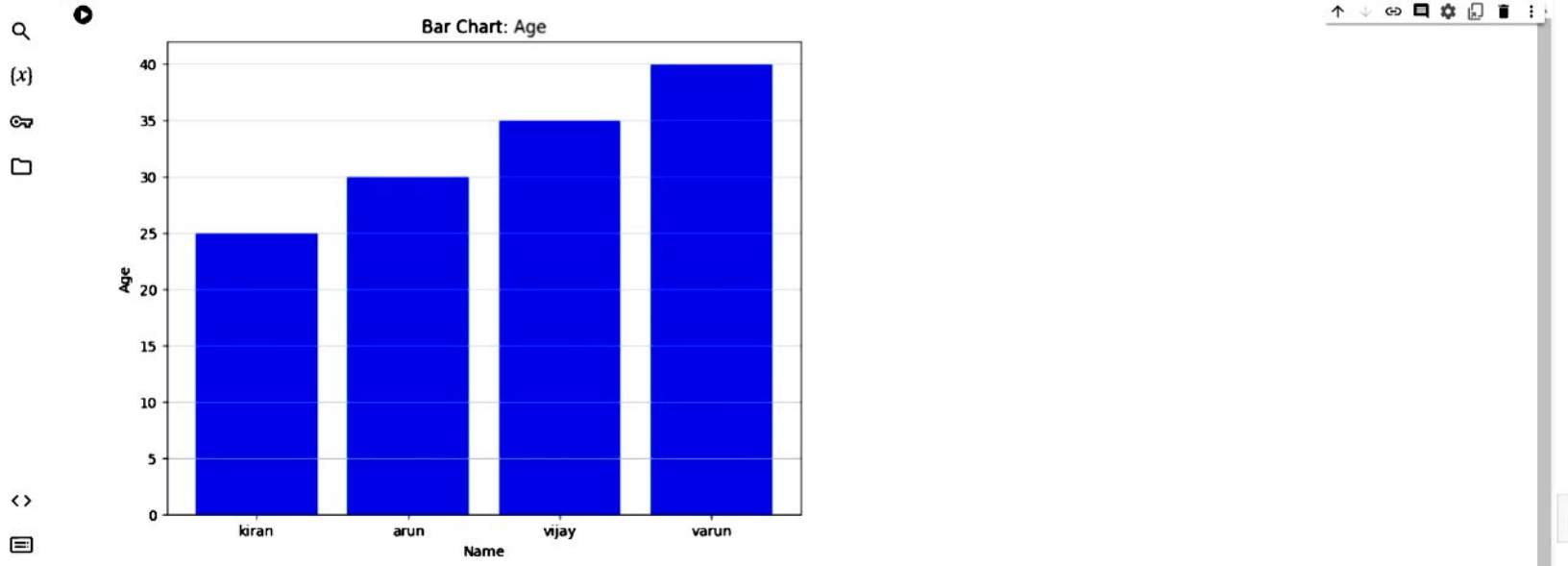

 Colab AI

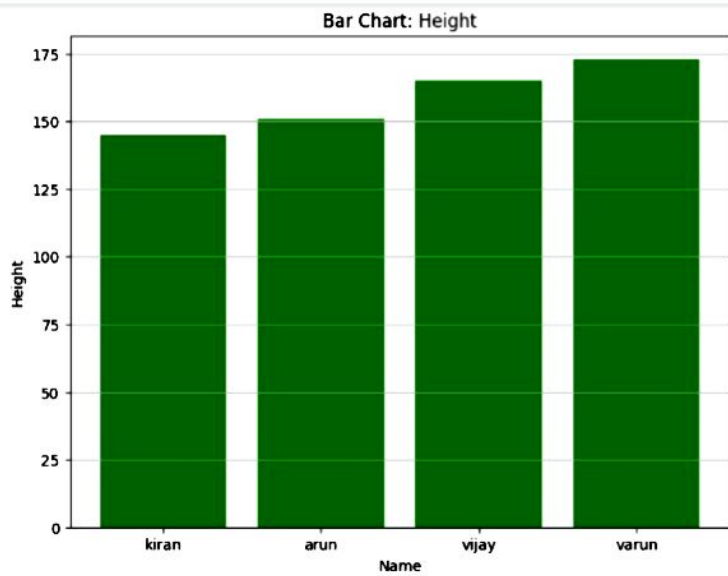
```
# Bar chart for height
plt.figure(figsize=(8, 6))
plt.bar(df['Name'], df['Height'], color='green')
plt.title('Bar Chart: Height')
plt.xlabel('Name')
plt.ylabel('Height')
plt.grid(axis='y')
plt.show()

# Bar chart for weight
plt.figure(figsize=(8, 6))
plt.bar(df['Name'], df['Weight'], color='red')
plt.title('Bar Chart: Weight')
plt.xlabel('Name')
plt.ylabel('Weight')
plt.grid(axis='y')
plt.show()
```



Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)





Bar Chart: Weight

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

0s completed at 8:24 PM

+ Code + Text

✓ RAM Disk Colab AI ^

```
#Histogram:
import pandas as pd
import matplotlib.pyplot as plt

# Fix the mismatch in the lengths of lists
people = ['kiran', 'arun', 'vijay', 'varun']
age = [25, 30, 35, 40]
height = [145, 151, 165, 173]
weight = [45, 55, 65, 75]

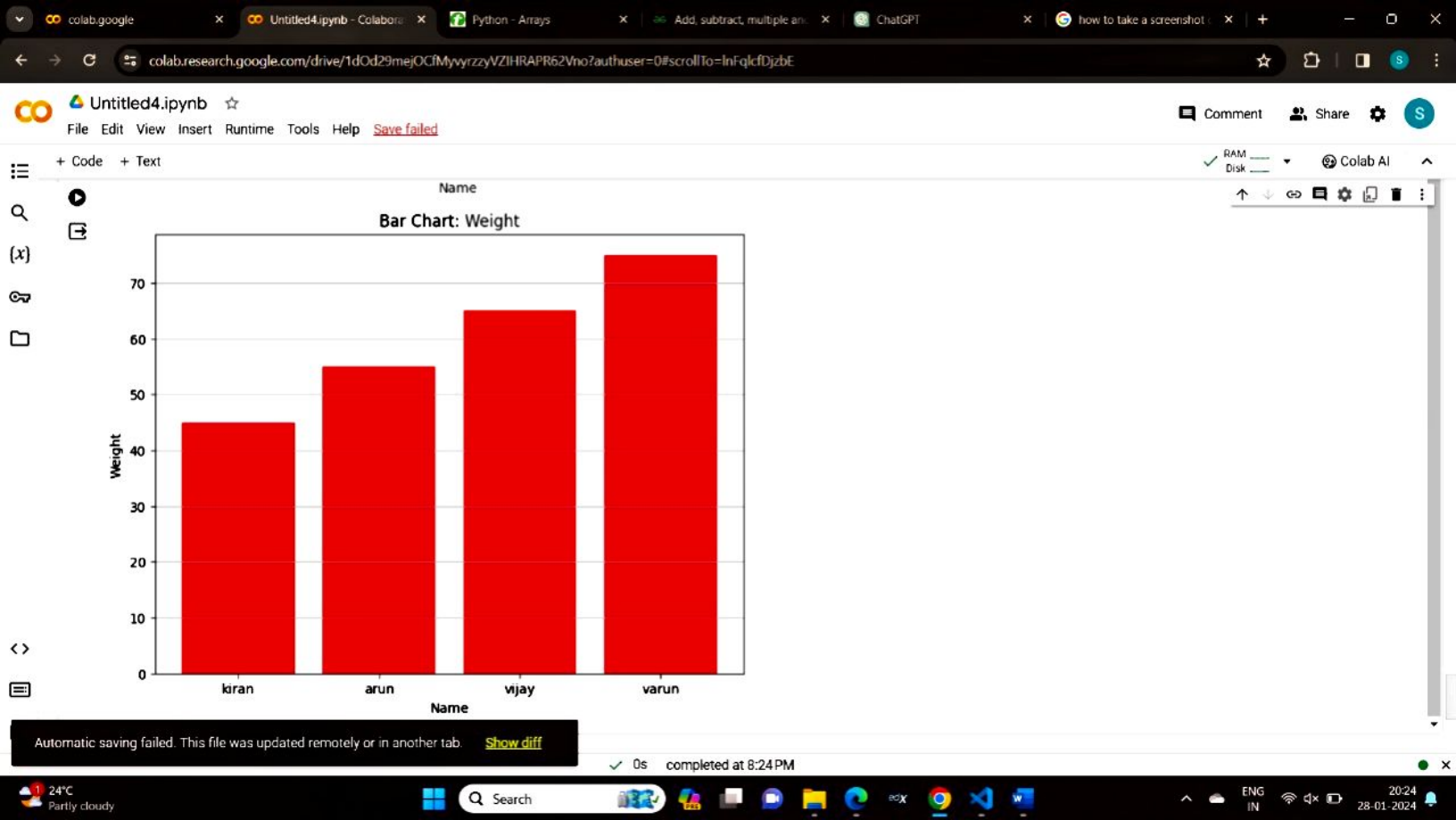
# Create a DataFrame
health_data = {
    'Name': people,
    'Age': age,
    'Height': height,
    'Weight': weight
}
df = pd.DataFrame(health_data)

# Histogram for Age
plt.figure(figsize=(8, 6))
plt.hist(df['Age'], bins=5, color='blue', edgecolor='black')
plt.title('Histogram of Age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()

# Histogram for Height
```

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

✓ 1s completed at 8:27 PM



colab.google

Untitled4.ipynb - Colabora

Python - Arrays

Add, subtract, multiple an

ChatGPT

how to take a screenshot

colab.research.google.com/drive/1dOd29mejOCfMyvryzyVZIHRApR62Vno?authuser=0#scrollTo=-IMZj2DfKQ--

Untitled4.ipynb

File Edit View Insert Runtime Tools Help Save failed

+ Code + Text

RAM Disk

Colab AI

Histogram for Height
plt.figure(figsize=(8, 6))
plt.hist(df['Height'], bins=5, color='green', edgecolor='black')
plt.title('Histogram of Height')
plt.xlabel('Height')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()

Histogram for Weight
plt.figure(figsize=(8, 6))
plt.hist(df['Weight'], bins=5, color='red', edgecolor='black')
plt.title('Histogram of Weight')
plt.xlabel('Weight')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()

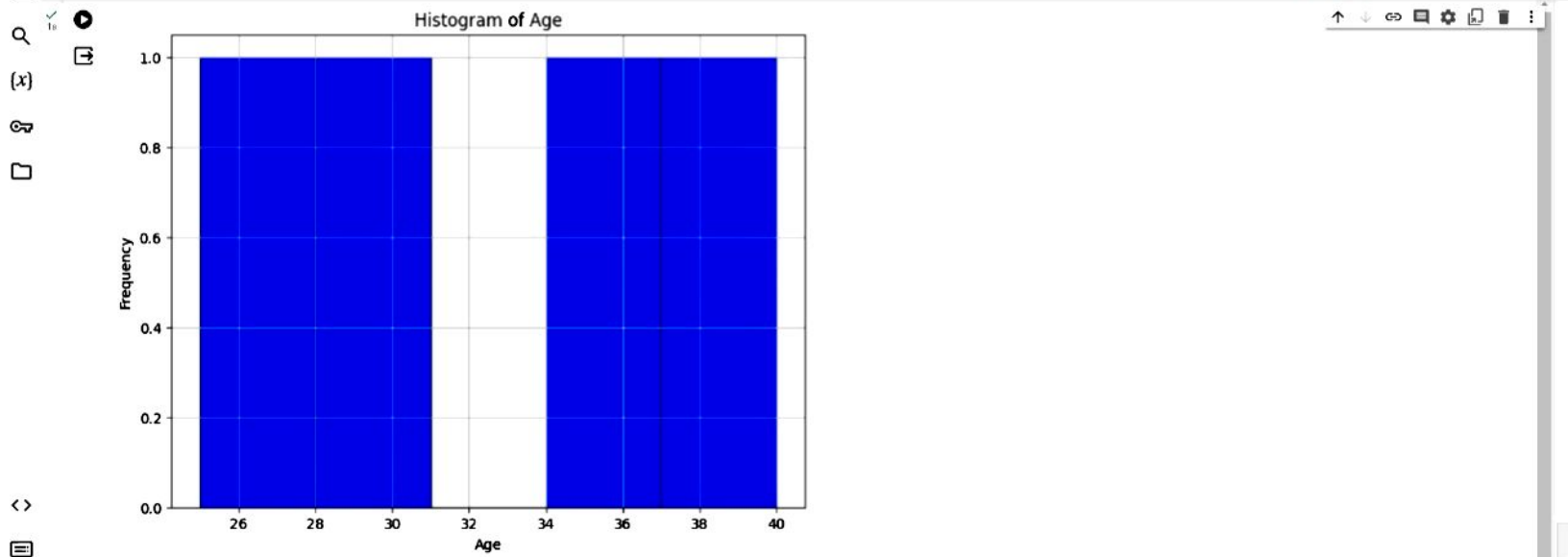
Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

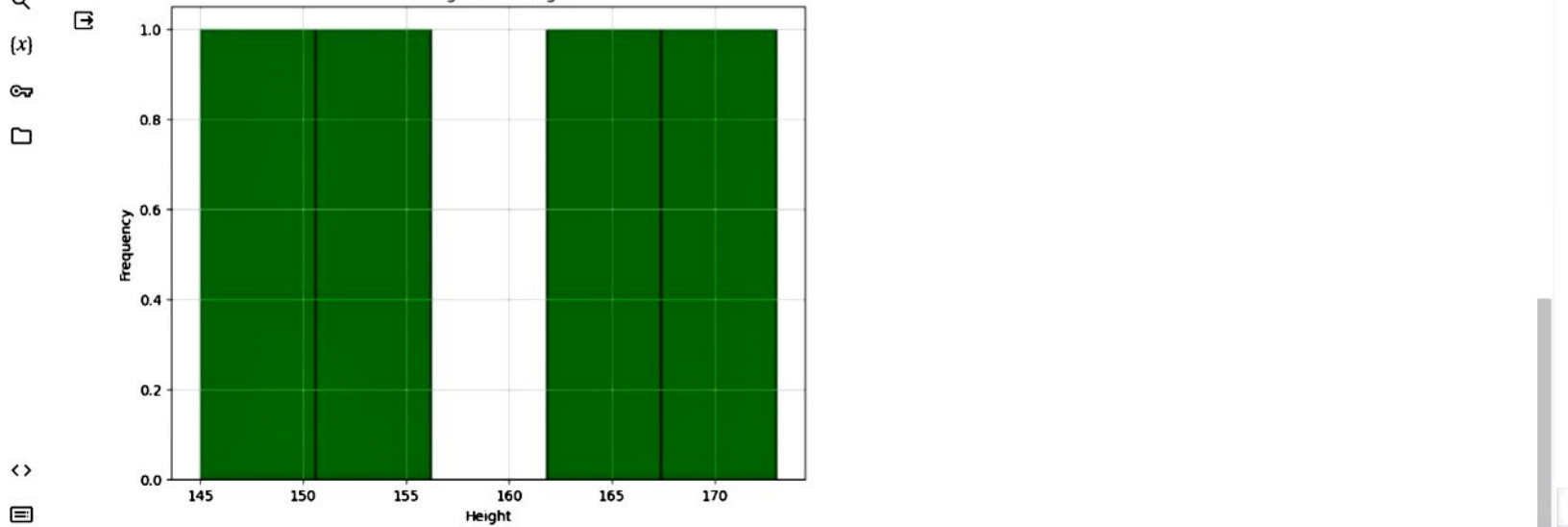
1s completed at 8:27 PM

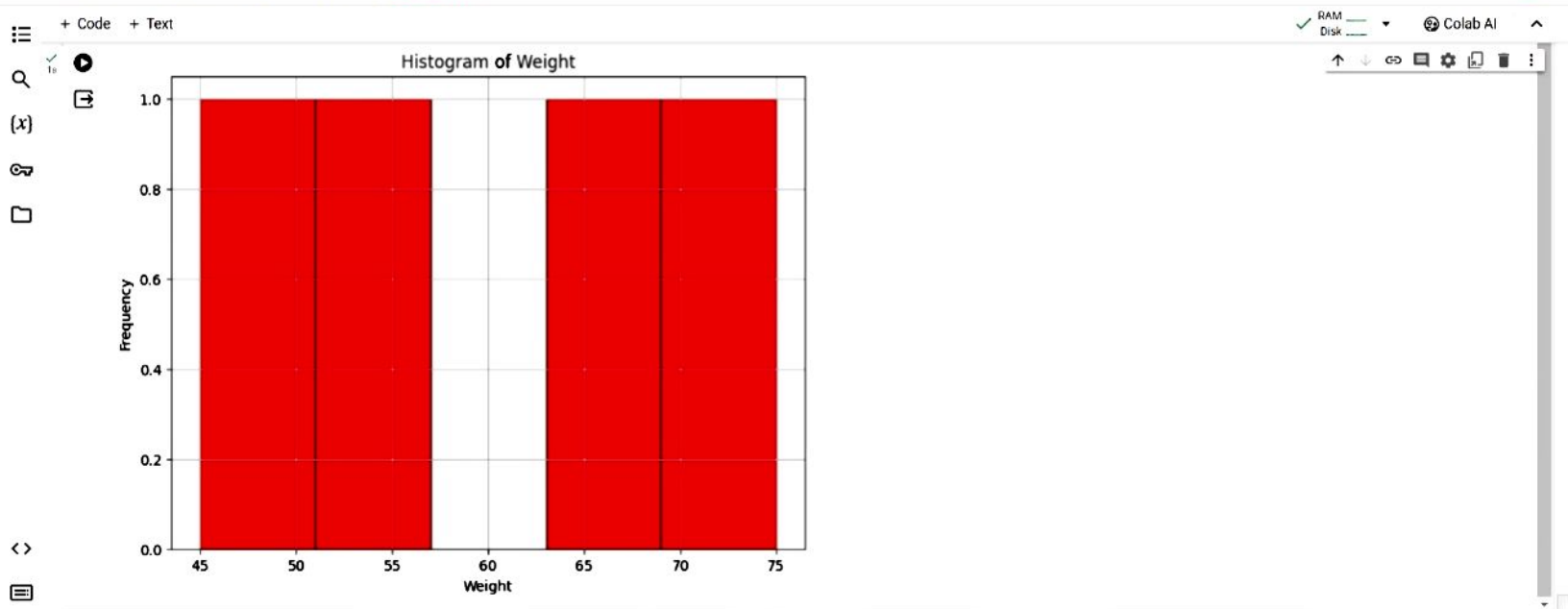
24°C
Partly cloudy

Search

ENG
IN20:27
28-01-2024







+ Code + Text

Find

RAM
Disk
No results
Colab AI

Provide Graph Title, labels for X, Y axis with proper justification and explanation of the graph.

import pandas as pd

import matplotlib.pyplot as plt

Fix the mismatch in the lengths of lists

people = ['kiran', 'arun', 'vijay', 'varun']

age = [25, 30, 35, 40]

height = [145, 151, 165, 173]

weight = [45, 55, 65, 75]

Create a DataFrame

health_data = {

'Name': people,

'Age': age,

'Height': height,

'Weight': weight

}

df = pd.DataFrame(health_data)

Histogram for Age

plt.figure(figsize=(8, 6))

plt.hist(df['Age'], bins=5, color='blue', edgecolor='black')

plt.title('Age Distribution')

plt.xlabel('Age')

plt.ylabel('Frequency')

plt.grid(True)

plt.show()

Explanation for Histogram of Age

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

individuals in the healthcare dataset.")

✓ 0s completed at 8:30 PM

```
# Explanation for Histogram of Age
print("\nExplanation for Age Distribution Histogram:")
print("The histogram illustrates the distribution of ages among individuals in the healthcare dataset.")
print("The X-axis represents age values, and the Y-axis represents the frequency of individuals in each age group.")
print("This histogram allows us to understand the age demographics of the dataset.")

# Histogram for Height
plt.figure(figsize=(8, 6))
plt.hist(df['Height'], bins=5, color='green', edgecolor='black')
plt.title('Height Distribution')
plt.xlabel('Height (cm)')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()

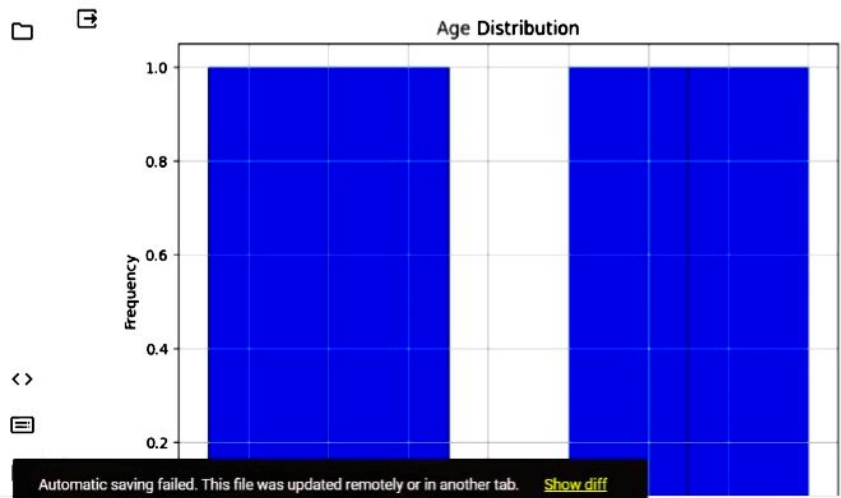
# Explanation for Histogram of Height
print("\nExplanation for Height Distribution Histogram:")
print("The histogram visualizes the distribution of heights among individuals in the healthcare dataset.")
print("The X-axis represents height values in centimeters, and the Y-axis represents the frequency of individuals in each height range.")
print("This histogram enables us to analyze the distribution of heights within the dataset.")

# Histogram for Weight
plt.figure(figsize=(8, 6))
plt.hist(df['Weight'], bins=5, color='red', edgecolor='black')
plt.title('Weight Distribution')
plt.xlabel('Weight (kg)')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

✓ 0s completed at 8:30 PM

```
# Explanation for Histogram of Weight
print("\nExplanation for Weight Distribution Histogram:")
print("The histogram showcases the distribution of weights among individuals in the healthcare dataset.")
print("The x-axis represents weight values in kilograms, and the y-axis represents the frequency of individuals in each weight range.")
print("This histogram provides insights into the distribution of weights within the dataset.")
```



Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

colab.google

Untitled4.ipynb - Colabora

Python - Arrays

Add, subtract, multiple an

ChatGPT

how to take a screenshot

colab.research.google.com/drive/1dOd29mejOCfMyvyrzzyVZIHRApR62Vno?authuser=0#scrollTo=GcjfhFvIAmG

☆

Untitled4.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Explanation for Height Distribution Histogram:
The histogram visualizes the distribution of heights among individuals in the healthcare dataset.
The X-axis represents height values in centimeters, and the Y-axis represents the frequency of individuals in each height range.
This histogram enables us to analyze the distribution of heights within the dataset.

Weight Distribution

Weight Range (kg)	Frequency
45 - 50	1.0
50 - 55	1.0
65 - 70	1.0
70 - 75	1.0

0s

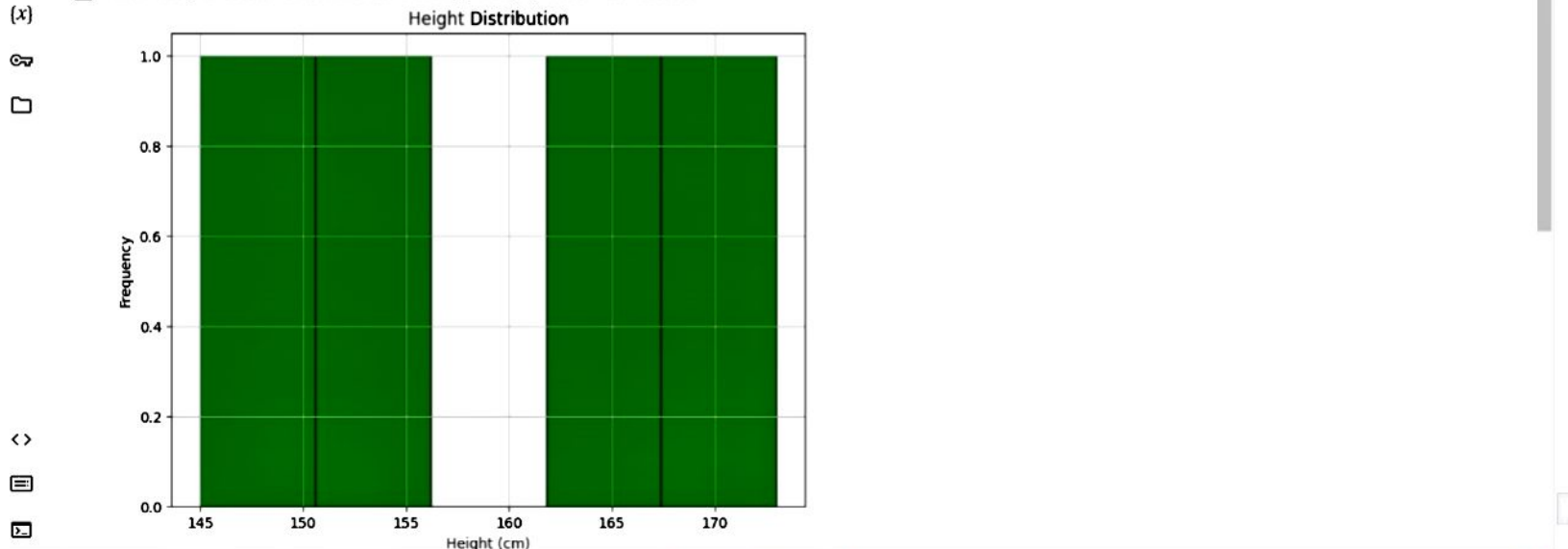
completed at 8:30 PM

25°C Haze

Search

20:31 28-01-2024

Explanation for Age Distribution Histogram:
The histogram illustrates the distribution of ages among individuals in the healthcare dataset.
The X-axis represents age values, and the Y-axis represents the frequency of individuals in each age group.
This histogram allows us to understand the age demographics of the dataset.



colab.research.google.com/drive/1fb77z_e6wE2jlwsEGA_sqcG1Rv_m6QXN#scrollTo=8JRzvsik-Jf

Untitled6.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk Colab AI

0s

import pandas as pd
df=pd.read_csv('/CarPrice_Assignment.csv')
df.head()

	car_ID	symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewheel	engine	location	wheelbase	...	enginesize	fuelsystem	bore	ratio	stroke	compressionratio
0	1	3	alfa-romero glulia	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47	2.68	9.0		
1	2	3	alfa-romero stelvio	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47	2.68	9.0		
2	3	1	alfa-romero Quadrifoglio	gas	std	two	hatchback	rwd	front	94.5	...	152	mpfi	2.68	3.47	9.0		
3	4	2	audi 100 ls	gas	std	four	sedan	fwd	front	99.8	...	109	mpfi	3.19	3.40	10.0		
4	5	2	audi 100ls	gas	std	four	sedan	4wd	front	99.4	...	136	mpfi	3.19	3.40	8.0		

5 rows x 26 columns

[]

Start coding or generate with AI.

0s

completed at 10:34AM

colab.research.google.com/drive/1fb77z_e6wE2jlwsEGA_sqcG1Rv_m6QXN#scrollTo=rQyl7I9DlcnT

Untitled6.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
import pandas as pd
df=pd.read_excel('/Salary_dataset.xls')
df.head()
```

Unnamed: 0	YearsExperience	Salary
0	0	1.2 39344
1	1	1.4 46206
2	2	1.6 37732
3	3	2.1 43526
4	4	2.3 39892

Files

libx32

media

mnt

opt

proc

python-apt

root

run

sbin

srv

sys

tmp

tools

usr

var

CarPrice_Assignment.csv

NGC-DL-CONTAINER-LICENSE

Salary_dataset.csv

Salary_dataset.xls

cuda-keyring_1.0-1_all.deb

Disk 81.41 GB available

0s completed at 10:41 AM

23°C Haze

Search

edX

10:41 29-01-2024

```
import pandas as pd
df=pd.read_csv('/salary_dataset.txt')
df.head()
```

	\tYearsExperience\tSalary
0	0\t1.2\t39344
1	1\t1.4\t46206
2	2\t1.6\t37732
3	3\t2.1\t43526
4	4\t2.3\t39892

- proc
- python-apt
- root
- run
- sbin
- srv
- sys
- tmp
- tools
- usr
- var
- CarPrice_Assignment.csv
- NGC-DL-CONTAINER-LICENSE
- Salary_dataset.csv
- Salary_dataset.txt
- Salary_dataset.xls
- cuda-keyring_1.0-1_all.deb
- even_odd.cpp
- even_odd.exe
- evennn.txt

URL

✓
0s

```
import pandas as pd
X=pd.read_excel("/content/Salary_dataset.xls")
X.head()
```

Unnamed: 0 YearsExperience Salary



0	0	1.2	39344
1	1	1.4	46206
2	2	1.6	37732
3	3	2.1	43526
4	4	2.3	39892



Double-click (or enter) to edit

✓ 0s completed at 3:25 PM



Search



ENG
IN

15:30

30-01-2024