

Class XII AI practical with solutions for session 2025-26

1. Create a Series from list and display it.

```
import pandas as pd

lst = [12,18,16]

s = pd.Series(lst, index=["Arun","Shyam","Geeta"])

print(s)
```

Output:

```
Arun    12
Shyam   18
Geeta   16
dtype: int64
```

2. Create a DataFrame from a dictionary and display it.

```
import pandas as pd

data = {
    'Name': ['Alice','Bob','Charlie','Deep'],
    'Marks': [85, 92, 78, 88]
}

df = pd.DataFrame(data)

print(df)
```

Output:

	Name	Marks
0	Alice	85
1	Bob	92
2	Charlie	78
3	Deep	88

3. Display first 3 & last 2 rows of the DataFrame.

```
import pandas as pd
```

```
data = {  
    'Name': ['Alice','Bob','Charlie','Deep'],  
    'Marks': [85, 92, 78, 88]  
}
```

```
df = pd.DataFrame(data)
```

```
print(df.head(3))
```

```
print(df.tail(2))
```

Output:

```
df.head(3)
```

```
Name Marks
```

```
0  Alice   85
```

```
1   Bob    92
```

```
2 Charlie  78
```

```
df.tail(2)
```

```
Name Marks
```

```
2 Charlie  78
```

```
3   Deep   88
```

4. Identify and fill missing values.

```
import pandas as pd
```

```
data = {  
    'Name': ['Alice','Bob','Charlie','Deep'],  
    'Marks': [85, 92, 78, 88]  
}
```

```
df = pd.DataFrame(data)
```

```
df['Marks'].fillna(df['Marks'].mean(), inplace=True)
```

```
print(df.isnull().sum())
```

Output:

```
Name    0
```

Marks 0

dtype: int64

5. Read a CSV into a DataFrame and save it.

```
df = pd.read_csv("students.csv")
```

```
print(df)
```

```
df.to_csv("output.csv", index=False)
```

Output:

students.csv

Name,Marks,Subject

Alice,85,Math

Bob,92,Science

Charlie,78,English

Deep,88,Math

df

	Name	Marks	Subject
--	------	-------	---------

0	Alice	85	Math
---	-------	----	------

1	Bob	92	Science
---	-----	----	---------

2	Charlie	78	English
---	---------	----	---------

3	Deep	88	Math
---	------	----	------

6. Compute mean, median, standard deviation.

```
import pandas as pd
```

```
data = {
```

```
    'Name': ['Alice', 'Bob', 'Charlie', 'Deep'],
```

```
    'Marks': [85, 92, 78, 88]
```

```
}
```

```
df = pd.DataFrame(data)
```

```
print(df["Marks"].mean(), df["Marks"].median(), df["Marks"].std())
```

Output:

Mean : 85.75

Median : 86.5

Standard Deviation: 5.909032633745278

7. Add a new student column and delete a row.

```
import pandas as pd

data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'Deep'],
    'Marks': [85, 92, 78, 88]
}

df = pd.DataFrame(data)

df["Grade"]=['A','B','C','B']

df.drop(index=2, inplace=True)

print(df)
```

Output:

	Name	Marks	Grade
0	Alice	85	A
1	Bob	92	B
3	Deep	88	B

8. Visualize marks distribution

```
import matplotlib.pyplot as plt

import pandas as pd

data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'Deep'],
    'Marks': [85, 92, 78, 88]
}

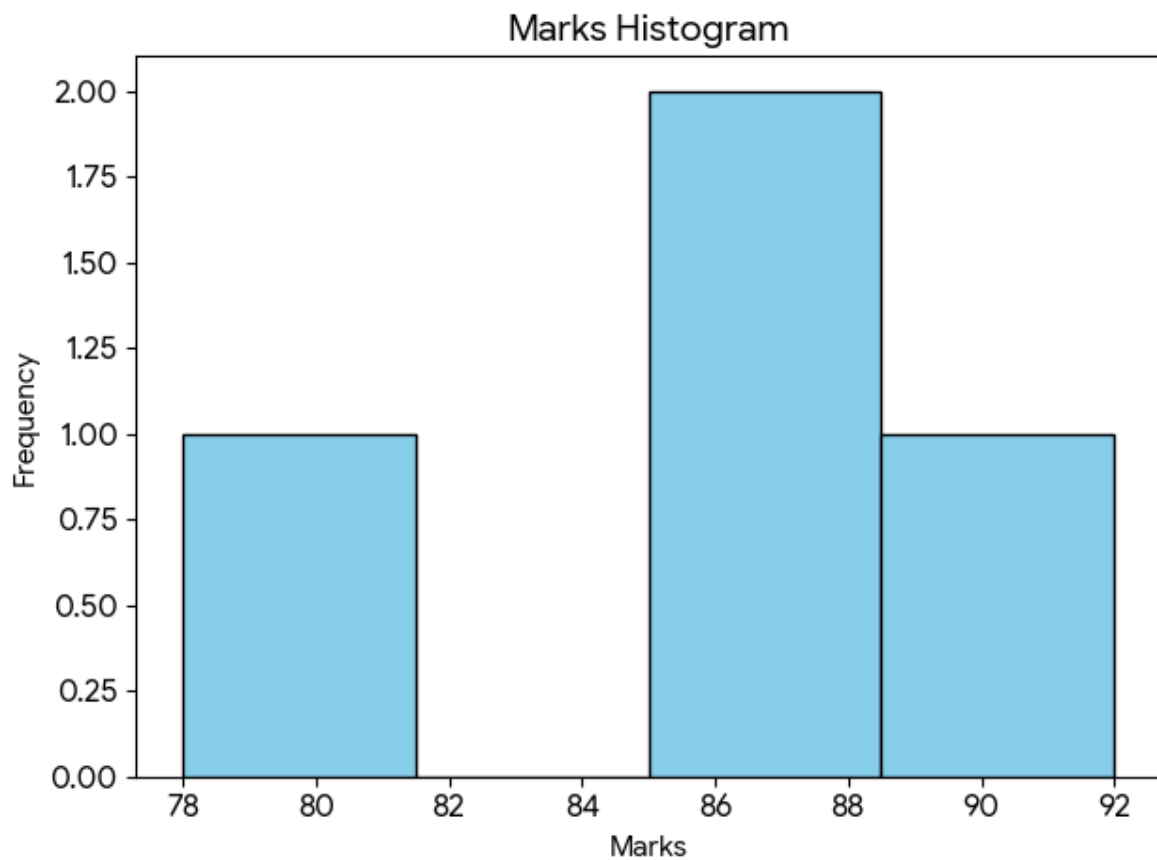
df = pd.DataFrame(data)

plt.hist(df["Marks"], bins=4)

plt.title("Marks Histogram")
```

```
plt.xlabel("Marks")  
plt.ylabel("Frequency")  
plt.show()
```

Output:



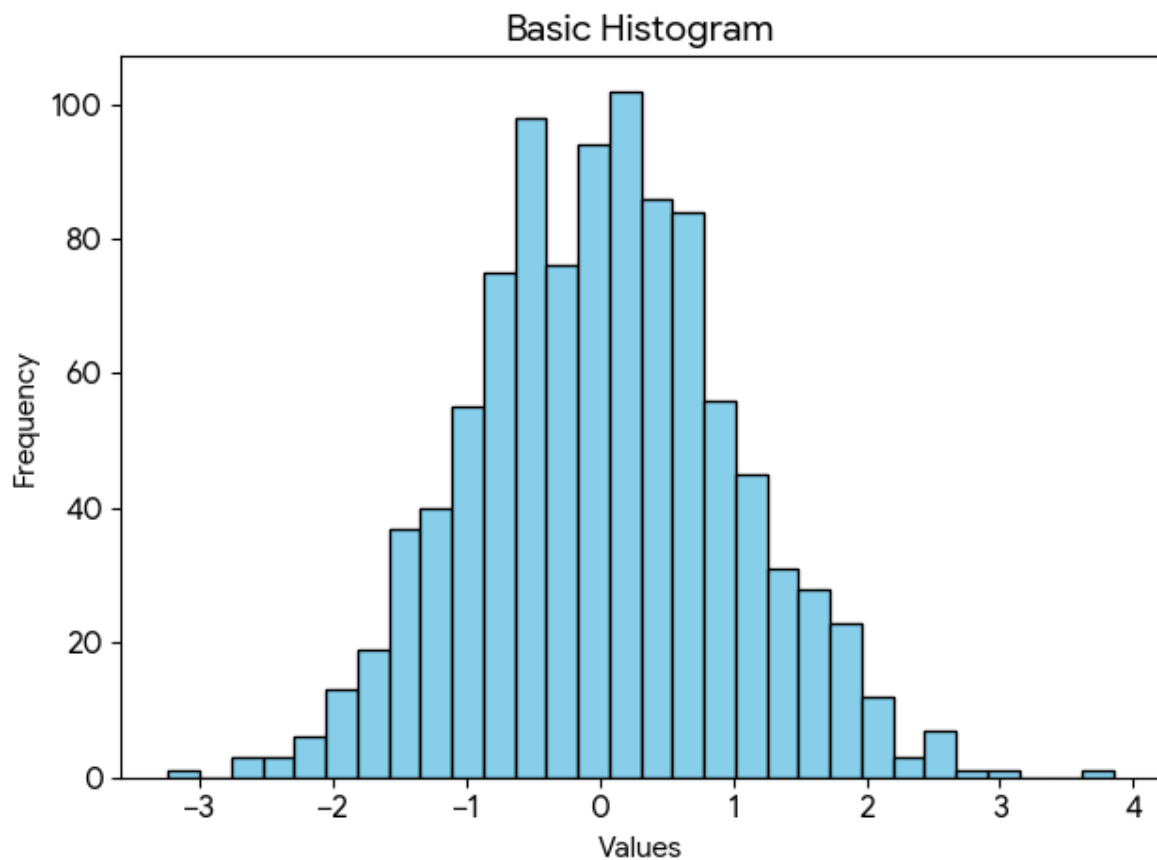
9. Create a basic histogram in Matplotlib of some random values.

```
import matplotlib.pyplot as plt  
  
import numpy as np  
  
# Generate random data for the histogram  
  
data = np.random.randn(1000)  
  
# Plotting a basic histogram  
  
plt.hist(data, bins=30, color='skyblue', edgecolor='black')  
  
# Adding labels and title  
  
plt.xlabel('Values')  
plt.ylabel('Frequency')  
plt.title('Basic Histogram')
```

```
# Display the plot
```

```
plt.show()
```

Output:



10. Plot pie chart for five fruits and their quantity in basket and Pull the "Apples" wedge 0.2 from the center of the pie.

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
y = np.array([35, 25, 25, 15])
```

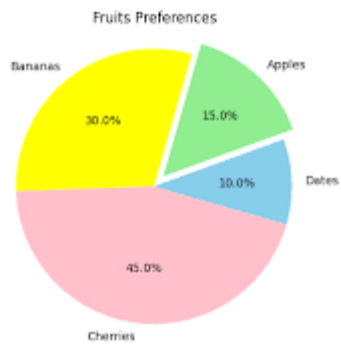
```
mylabels = ["Apples", "Bananas", "Cherries", "Dates"]
```

```
myexplode = [0.2, 0, 0, 0]
```

```
plt.pie(y, labels = mylabels, explode = myexplode)
```

```
plt.show()
```

Output:



11. Use the `grid()` function to add grid lines to the plot.

```
import numpy as np

import matplotlib.pyplot as plt

x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])

y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])

plt.title("Sports Watch Data")

plt.xlabel("Average Pulse")

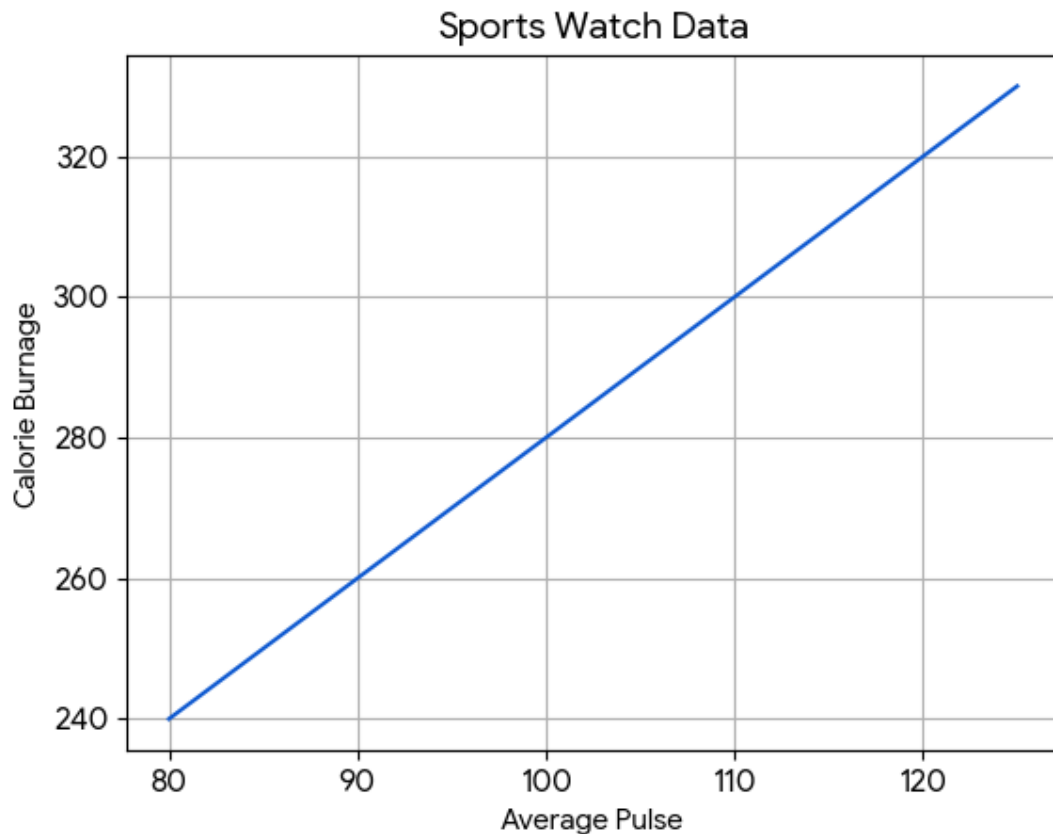
plt.ylabel("Calorie Burnage")

plt.plot(x, y)

plt.grid()

plt.show()
```

Output:



12. Build a simple regression and evaluate model.

```
from sklearn.model_selection import train_test_split  
  
from sklearn.linear_model import LinearRegression  
  
from sklearn.metrics import mean_squared_error  
  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)  
  
model = LinearRegression()  
  
model.fit(X_train, y_train)  
  
pred = model.predict(X_test)  
  
print("RMSE:", mean_squared_error(y_test, pred, squared=False))
```

Output:

RMSE: 4.123849...

13. Use Google Gemini/ChatGPT to generate conversational outputs.

- Write prompts.
- Collect responses → include sample chat in report.

Output:

Prompt: Define AI and Compare AI Domains.

Response:

Artificial Intelligence (AI) is a branch of computer science that focuses on creating systems capable of performing tasks that typically require human intelligence. These tasks include learning, reasoning, problem-solving, perception, and understanding natural language.

Essentially, AI is the simulation of human intelligence processes by machines, especially computer systems.

Feature	Data Science	Computer Vision	NLP
Primary Input	Numbers, Tables, Big Data	Images, Videos	Text, Speech
Human Sense	Logical Reasoning	Sight (Vision)	Communication (Hearing/Speech)
Common Use Case	Stock Market Prediction	Surveillance / Security	Language Translation
Key Skill	Statistics & Math	Pattern Recognition	Linguistics & Context

14. Create images using Canva or AI Image tool like Gemini from text prompt.

- Use prompt: *“Classroom of future robots teaching children”*
- Include generated image in file.

Output:

Prompt: Generate image of Classroom of future robots teaching children

Response:



15. Use Animaker AI to make a 10–15 sec video.

- Use text prompt to storyboard.
- Save and include screenshot & description.

Output:

Prompt: Create a video of 30 seconds explaining AI.

Response: Video is generated showing AI explanation

16. Build classification in Python using logistic regression:

```
from sklearn.linear_model import LogisticRegression

model = LogisticRegression()

model.fit(X_train, y_train)

print("Accuracy:", model.score(X_test, y_test))
```

Output:

Accuracy: 0.9

17. Normalize data (marks) using Min-Max scaling.

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

scaled = scaler.fit_transform(df[['Marks']])
```

Output:

```
[[0.5    ] # (85-78)/14
```

```
[1.     ] # (92-78)/14
```

```
[0.     ] # (78-78)/14
```

```
[0.71428571]] # (88-78)/14
```

18. Given confusion matrix data, calculate metrics manually

TP, TN, FP, FN = 40,50,5,5

accuracy = (TP+TN)/(TP+TN+FP+FN)

Output:

0.9

19. Train-Test Split ; Split features & target

```
from sklearn.model_selection import train_test_split
```

```
X = df[["Marks"]]
```

```
y = [1,0,1,0] # sample labels
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

```
print(len(X_train), len(X_test))
```

Simple Linear Regression : Fit and predict

```
from sklearn.linear_model import LinearRegression
```

```
reg = LinearRegression()
```

```
reg.fit(X_train, y_train)
```

```
pred = reg.predict(X_test)
```

```
print("Predictions:", pred)
```

Output:

3 1

20. Accuracy of Classification: Logistic regression accuracy

```
from sklearn.linear_model import LogisticRegression
```

```
from sklearn.metrics import accuracy_score
```

```
log = LogisticRegression()
```

```
log.fit(X_train, y_train)
```

```
y_pred = log.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

Output:

Accuracy: 1.0

21. K-Nearest Neighbors (KNN) : Simple KNN classification

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)
print("KNN Score:", knn.score(X_test, y_test))
```

Output:

KNN Score: 1.0

22. Bag of Words (Sklearn)

Text vectorization

```
from sklearn.feature_extraction.text import CountVectorizer
corpus = ["AI is cool", "AI is useful"]
cv = CountVectorizer()
print(cv.fit_transform(corpus).toarray())
print(cv.get_feature_names_out())
```

Output:

['ai' 'cool' 'is' 'useful']

Class XII AI practical with solutions for session 2025-26

1. Create a Series from list and display it.

Output:

Arun 12

Shyam 18

Geeta 16

dtype: int64

2. Create a DataFrame from a dictionary and display it.

Output:

	Name	Marks
0	Alice	85
1	Bob	92
2	Charlie	78
3	Deep	88

3. Display first 3 & last 2 rows of the DataFrame.

Output:

df.head(3)

	Name	Marks
0	Alice	85
1	Bob	92
2	Charlie	78

df.tail(2)

	Name	Marks
2	Charlie	78
3	Deep	88

4. Identify and fill missing values.

Output:

Name 0

Marks 0

dtype: int64

5. Read a CSV into a DataFrame and save it.

Output:

students.csv

Name,Marks,Subject

Alice,85,Math

Bob,92,Science

Charlie,78,English

Deep,88,Math

df

	Name	Marks	Subject
--	------	-------	---------

0	Alice	85	Math
---	-------	----	------

1	Bob	92	Science
---	-----	----	---------

2	Charlie	78	English
---	---------	----	---------

3	Deep	88	Math
---	------	----	------

6. Compute mean, median, standard deviation.

Output:

Mean : 85.75

Median : 86.5

Standard Deviation: 5.909032633745278

7. Add a new student column and delete a row.

Output:

	Name	Marks	Grade
--	------	-------	-------

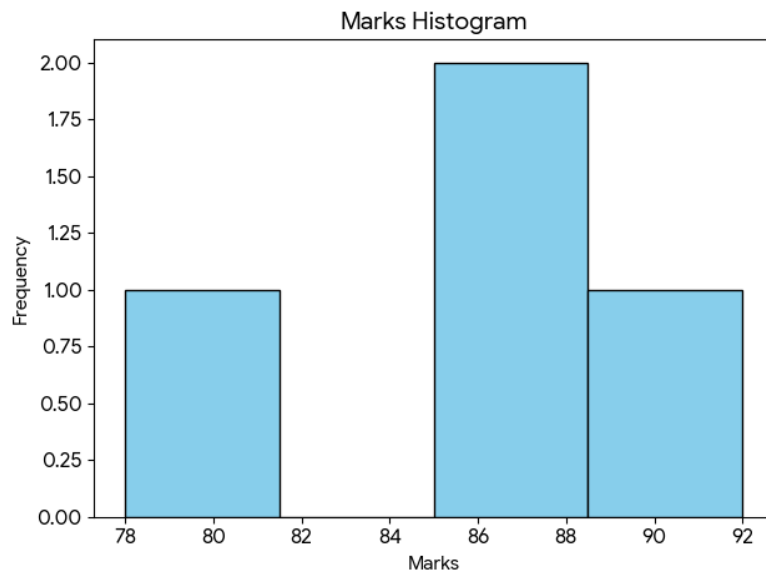
0	Alice	85	A
---	-------	----	---

1	Bob	92	B
---	-----	----	---

3	Deep	88	B
---	------	----	---

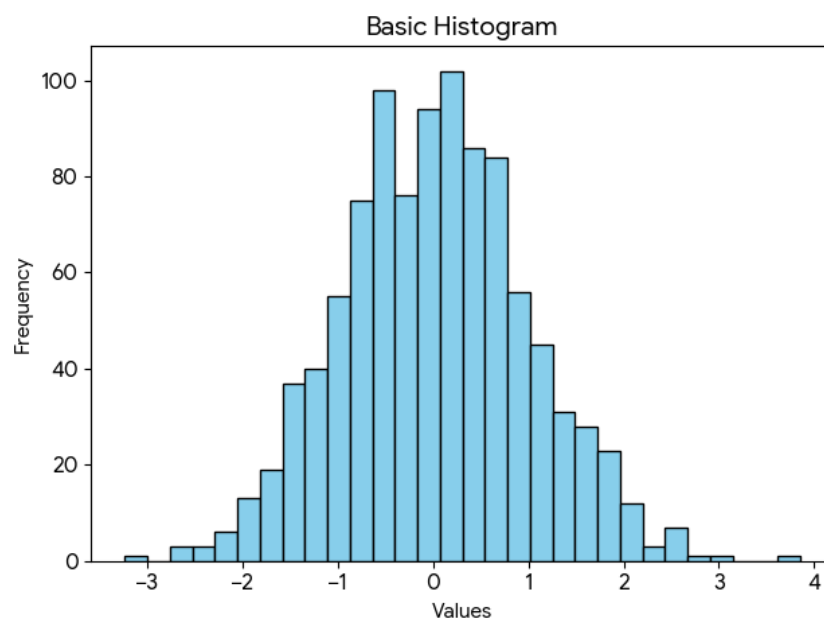
8. Visualize marks distribution

Output:



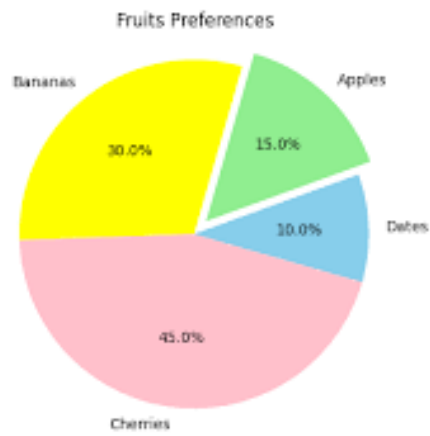
9. Create a basic histogram in Matplotlib of some random values.

Output:



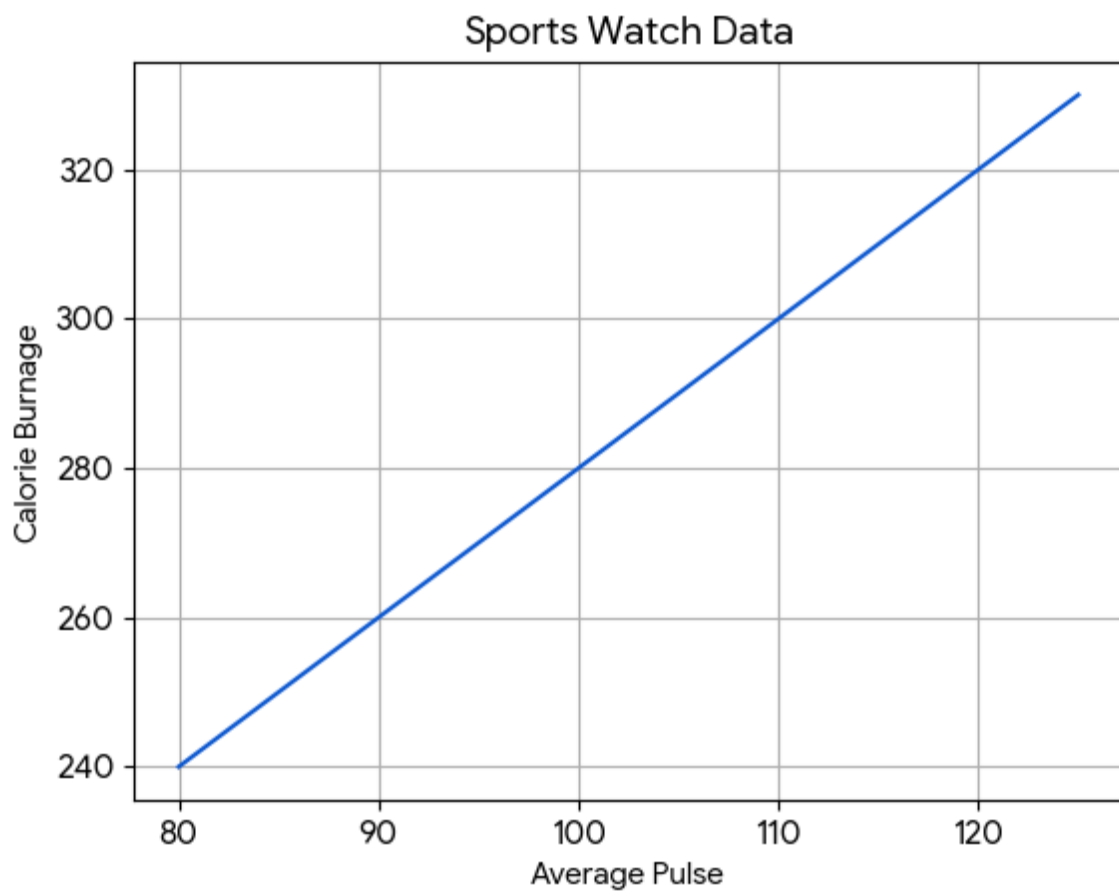
10. Plot pie chart for five fruits and their quantity in basket and Pull the "Apples" wedge 0.2 from the center of the pie.

Output:



11. Use the `grid()` function to add grid lines to the plot.

Output:



12. Build a simple regression and evaluate model.

Output:

RMSE: 4.123849...

13. Use Google Gemini/ChatGPT to generate conversational outputs.

Output:

Prompt: Define AI and Compare AI Domains.

Response:

Artificial Intelligence (AI) is a branch of computer science that focuses on creating systems capable of performing tasks that typically require human intelligence. These tasks include learning, reasoning, problem-solving, perception, and understanding natural language.

Essentially, AI is the simulation of human intelligence processes by machines, especially computer systems.

Feature	Data Science	Computer Vision	NLP
Primary Input	Numbers, Tables, Big Data	Images, Videos	Text, Speech
Human Sense	Logical Reasoning	Sight (Vision)	Communication (Hearing/Speech)
Common Use Case	Stock Market Prediction	Surveillance / Security	Language Translation
Key Skill	Statistics & Math	Pattern Recognition	Linguistics & Context

14. Create images using Canva or AI Image tool like Gemini from text prompt.**Output:**

Prompt: Generate image of Classroom of future robots teaching children

Response:**15. Use Animaker AI to make a 10–15 sec video.**

Output:

Prompt: Create a video of 30 seconds explaining AI.

Response: Video is generated showing AI explanation

16. Build classification in Python using logistic regression:

Output:

Accuracy: 0.9

17. Normalize data (marks) using Min-Max scaling.

Output:

[[0.5] # (85-78)/14

[1.] # (92-78)/14

[0.] # (78-78)/14

[0.71428571]] # (88-78)/14

18. Given confusion matrix data, calculate metrics manually

Output:

0.9

19. Train-Test Split ; Split features & target

Output:

3 1

20. Accuracy of Classification: Logistic regression accuracy

Output:

Accuracy: 1.0

21. K-Nearest Neighbors (KNN) : Simple KNN classification

Output:

KNN Score: 1.0

22. Bag of Words (Sklearn)

Output:

['ai' 'cool' 'is' 'useful']