

Extracted Python Code

Python Programs

```
import pandas as pd
iris = pd.read_csv('iris.csv', names=['sepal_length', 'sepal_width', 'petal_length',
'petal_width', 'class'])
iris.tail()
iris.describe()

"""# New Section"""

import matplotlib.pyplot as plt
x=iris['petal_length']
y=iris['petal_width']
plt.scatter(x,y)
plt.title("Scatter plot Petal length vs width")
plt.xlabel("petal length")
plt.ylabel("Petal width")

import matplotlib.pyplot as plt
fig, ax = plt.subplots()
# scatter the sepal_length against the sepal_width
ax.scatter(iris['sepal_length'], iris['sepal_width'])
# set a title and labels
ax.set_title('Iris Dataset')
ax.set_xlabel('sepal_length')
ax.set_ylabel('sepal_width')

colors = {'Iris-setosa':'r', 'Iris-versicolor':'g', 'Iris-virginica':'b'}
species_colors = iris['class'].map(colors)
fig, ax = plt.subplots()
ax.scatter(iris['sepal_length'],iris['sepal_width'],c=species_colors)
ax.set_title("Scatter Plot of Iris Dataset")
ax.set_xlabel('Sepal Length')
ax.set_ylabel('Sepal width')

"""How many flowers belong to each species?"""

species_counts = iris['class'].value_counts()

plt.bar(species_counts.index, species_counts.values, color='lightblue')
plt.title("Number of Flowers by Species")
plt.xlabel("Species")
plt.ylabel("Count")
plt.grid(axis='y')

"""What is the distribution of petal length?"""

plt.hist(iris['petal_length'], bins=10, color='lightgreen', edgecolor='black')
plt.title("Distribution of Petal Length")
plt.xlabel("Petal Length (cm)")
```

Extracted Python Code

```
plt.ylabel("Frequency")
plt.grid(True)
plt.show()

"""Compare petal length between species."""

species = iris['class'].unique()
data = [iris[iris['class'] == s]['petal_length'] for s in species]

plt.boxplot(data, labels=species)
plt.title("Petal Length by Species")
plt.xlabel("Species")
plt.ylabel("Petal Length (cm)")
plt.grid(True)
plt.show()

"""What is the relationship between petal length and petal width?"""

plt.scatter(iris['petal_length'], iris['petal_width'], color='purple', alpha=0.6)
plt.title("Petal Length vs Petal Width")
plt.xlabel("Petal Length (cm)")
plt.ylabel("Petal Width (cm)")
plt.show()

"""Mtcars Dataset Operations"""

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv('/content/mtcars.csv')
df.head()

plt.scatter(df['wt'],df['mpg'])
plt.title('Scatter plot of wt vs mpg')
plt.xlabel('wt')
plt.ylabel('mpg')
plt.show()

"""**Bar plot to check the frequency distribution of transmission type of cars.**"""

auto_count = (df['am'] == 0).sum()
manual_count = (df['am'] == 1).sum()
labels = ['Automatic', 'Manual']
counts = [auto_count, manual_count]
plt.bar(labels, counts, color=['orange', 'blue'])
plt.title("Frequency of Transmission Types")
plt.xlabel("Transmission Type")
plt.ylabel("Number of Cars")
plt.grid(axis='y')
plt.show()

"""What is the distribution of fuel efficiency (mpg) across all cars?"""
```

Extracted Python Code

```
plt.hist(df['mpg'], bins=8, color='skyblue', edgecolor='black')
plt.title("Histogram of MPG")
plt.xlabel("Miles per Gallon")
plt.ylabel("Frequency")
plt.grid(True)
plt.show()

"""What is the relationship between car weight and mpg?"""

plt.scatter(df['wt'], df['mpg'], color='green')
plt.title("Car Weight vs MPG")
plt.xlabel("Weight (1000 lbs)")
plt.ylabel("Miles per Gallon")
plt.grid(True)
plt.show()

"""Which transmission type is more common (Manual or Automatic)?**bold text**"""

auto = (df['am'] == 0).sum()
manual = (df['am'] == 1).sum()

plt.bar(['Automatic', 'Manual'], [auto, manual], color=['orange', 'blue'])
plt.title("Frequency of Transmission Types")
plt.ylabel("Number of Cars")
plt.grid(axis='y')
plt.show()

"""Draw a box plot to visualize the distribution of the mpg (Miles per Gallon) values in
the mtcars dataset."""

plt.boxplot(df['mpg'])
plt.title("Box Plot of MPG")
plt.ylabel("Miles per Gallon")
plt.grid(True)
plt.show()

"""How does MPG vary by number of cylinders?"""

cylinders = sorted(df['cyl'].unique())
data = [df[df['cyl'] == c]['mpg'] for c in cylinders]

plt.boxplot(data, labels=cylinders)
plt.title("MPG by Number of Cylinders")
plt.xlabel("Cylinders")
plt.ylabel("MPG")
plt.grid(True)
plt.show()

"""How many cars have each gear count?"""

# Count number of cars by gear
```

Extracted Python Code

```
gear_counts = df['gear'].value_counts().sort_index()

# Bar plot
plt.bar(gear_counts.index.astype(str), gear_counts.values, color='lightgreen')
plt.title("Number of Cars by Gear Count")
plt.xlabel("Number of Gears")
plt.ylabel("Number of Cars")
plt.grid(axis='y')
plt.show()

"""How many cars in the mtcars dataset have 4, 6, and 8 cylinders?

Display the result using a bar plot.
"""

# Count cars by number of cylinders
cyl_counts = df['cyl'].value_counts().sort_index()

# Bar plot
plt.bar(cyl_counts.index.astype(str), cyl_counts.values, color='skyblue')
plt.title("Number of Cars by Cylinder Count")
plt.xlabel("Number of Cylinders")
plt.ylabel("Number of Cars")
plt.grid(axis='y')
plt.show()

"""Student Performance Data Set"""

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
sp=pd.read_csv('/content/StudentsPerformance.csv')
sp

"""How many students are male vs female?"""

gender_counts = sp['gender'].value_counts()
plt.bar(gender_counts.index, gender_counts.values, color=['blue', 'pink'])
plt.title("Student Count by Gender")
plt.xlabel("Gender")
plt.ylabel("Number of Students")
plt.grid(axis='y')
plt.show()

"""What is the distribution of math scores?"""

plt.hist(sp['math score'], bins=10, color='lightblue', edgecolor='black')
plt.title("Math Score Distribution")
plt.xlabel("Math Score")
plt.ylabel("Number of Students")
```

Extracted Python Code

```
plt.grid(True)
plt.show()

"""Compare math scores between genders (Box Plot)"""

genders = sp['gender'].unique()
data = [sp[sp['gender'] == g]['math score'] for g in genders]

plt.boxplot(data, labels=genders)
plt.title("Math Score by Gender")
plt.xlabel("Gender")
plt.ylabel("Math Score")
plt.grid(True)
plt.show()

"""How many students completed the test preparation course?"""

prep_counts = sp['test preparation course'].value_counts()
plt.bar(prepare_counts.index, prepare_counts.values, color=['green', 'orange'])
plt.title("Test Preparation Course Completion")
plt.xlabel("Preparation Status")
plt.ylabel("Number of Students")
plt.grid(axis='y')
plt.show()

"""Relationship between reading and writing scores (Scatter Plot)"""

plt.scatter(sp['reading score'], sp['writing score'], color='purple', alpha=0.5)
plt.title("Reading vs Writing Scores")
plt.xlabel("Reading Score")
plt.ylabel("Writing Score")
plt.grid(True)
plt.show()

import seaborn as sns
import numpy as np
# Generate random data for two different distributions
data1 = np.random.normal(0, 1, 1000)
data2 = np.random.normal(2, 1, 1000)
# Create a box plot to compare the two distributions
sns.boxplot(data=[data1, data2])
# Adding labels and a title
plt.xlabel('Distribution')
plt.ylabel('Value')
plt.title('Box Plot Comparison of Two Distributions')
# Show the plot
plt.show()
```