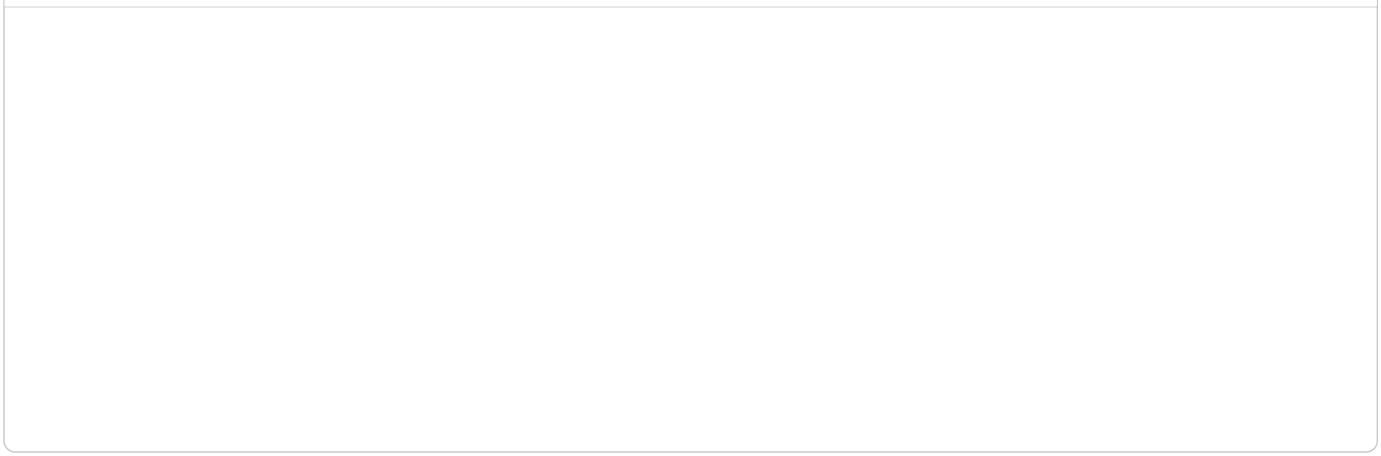
```
# matplotlib
# seaborn
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

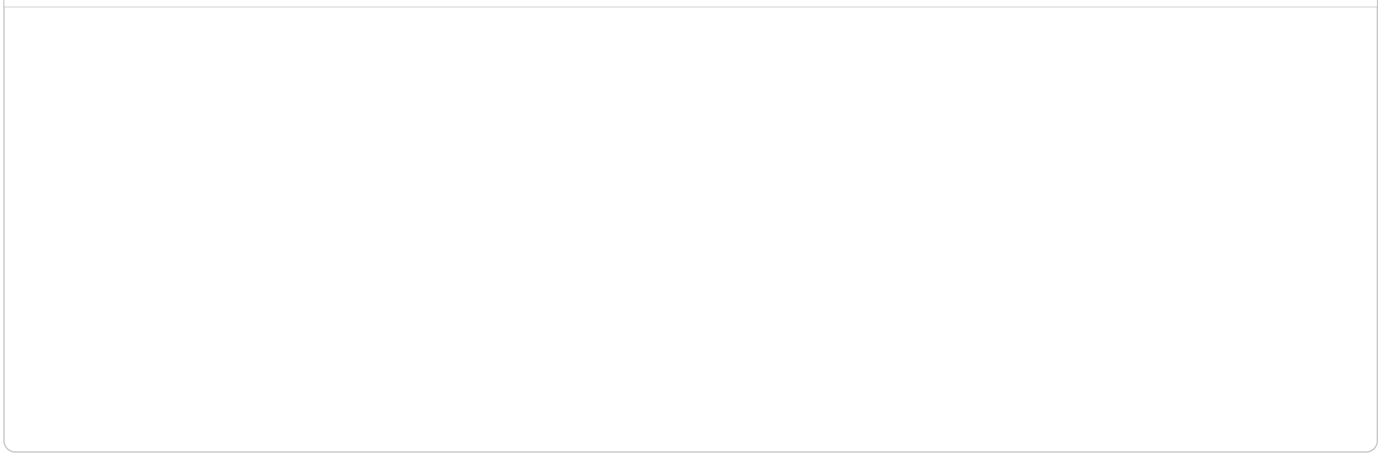
```
x = [1,2,3,4,5]
y = [10,20,30,40,50]
plt.figure(figsize=(6,3))
plt.plot(x,y, color = 'c',linewidth=3,linestyle="--",marker='o',markerfacecolor='red',alpha=1) # alpha = transparency
plt.grid()
plt.xlabel("X Values")
plt.ylabel("Y Values")
plt.title("line plot")
plt.show()
```



```
print(plt.style.available)
```

```
['Solarize_Light2', '_classic_test_patch', '_mpl-gallery', '_mpl-gallery-nogrid', 'bmh', 'classic', 'dark_background', 'fast', '
```

```
plt.style.use("dark_background")
x = [1,2,3,4,5]
y = [10,20,30,40,50]
plt.figure(figsize=(6,3))
plt.plot(x,y, color = 'c',linewidth=3,linestyle="--",marker='o',markerfacecolor='red',alpha=1) # alpha = transparency
plt.grid()
plt.xlabel("X Values")
plt.ylabel("Y Values")
plt.title("line plot")
plt.show()
```



```
x = ['A','B','C','D']
y = [20,40,60,70]
plt.barh(x,y,color='c',edgecolor='white',height=0.3,linewidth=1)
plt.xlabel("X Values")
plt.ylabel("Y Values")
```

```
plt.title("Bar plot")
plt.show()
```

```
x = ['A','B','C','D']
y = [20,40,60,70]
plt.bar(x,y,color='c',edgecolor='white',width=0.5,linewidth=1)
plt.xlabel("X Values")
plt.ylabel("Y Values")
plt.title("Bar plot")
plt.show()
```

```
# scatter plot
x= np.random.randint(1,50,20)
y= np.random.randint(1,50,20)
plt.scatter(x,y,alpha =1)
plt.xlabel("X Values")
plt.ylabel("Y Values")
plt.title("Scatter plot")
plt.show()
```

```
# histogram
x= np.random.rand(50)
plt.hist(x,bins=25)
plt.xlabel("X INdex")
plt.ylabel("Y Index")
plt.title("Histogram plot")
plt.show()
```

```
# subplots
x = [1,2,3,4,5]
y = [10,20,30,40,50]
z= [5,10,15,20,25]
# first subplot
plt.subplot(1,2,1)
plt.plot(x,y,color='red')

# second subplot
plt.subplot(1,2,2)
plt.plot(x,z,color='yellow')

plt.show()
```

```
x = np.random.rand(4,4)
x
```

```
array([[0.75589961, 0.51413269, 0.43934646, 0.64905235],
       [0.4952967 , 0.34121553, 0.97865533, 0.38653022],
       [0.25589315, 0.19097615, 0.40283675, 0.92776465],
       [0.1899779 , 0.87417721, 0.9597644 , 0.62363467]])
```

```
# heatmap
x = np.random.rand(4,4)
plt.imshow(x,cmap='hot',interpolation='bicubic')
plt.colorbar()
plt.show()
```

```
df = pd.read_csv('/content/sales_data.csv')
```

```
df.head()
```

```
df.shape
```

```
(240, 9)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 240 entries, 0 to 239
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype
```

```
---  ------            --------------  -----
 0   Transaction ID    240 non-null    int64
 1   Date              240 non-null    object
 2   Product Category  240 non-null    object
 3   Product Name      240 non-null    object
 4   Units Sold        240 non-null    int64
 5   Unit Price        240 non-null    float64
 6   Total Revenue     240 non-null    float64
 7   Region            240 non-null    object
 8   Payment Method    240 non-null    object
dtypes: float64(2), int64(2), object(5)
memory usage: 17.0+ KB
```

```
df.describe()
```

```
df['Region'].value_counts()
```

```
# total revenue based on product category
p= df.groupby("Product Category")["Total Revenue"].sum()
plt.figure(figsize=(10,5))
plt.bar(p.index,p.values)
plt.xlabel('Category')
plt.ylabel('Revenue')
plt.show()
```

```
# revenue over time
df['Date']= pd.to_datetime(df['Date'])
plt.figure(figsize=(10,6))
plt.plot(df['Date'],df['Total Revenue'],marker='o')
plt.xlabel('Date')
plt.ylabel('Total Revenue')
plt.title("revenue over time ")
plt.show()
```

```
# histogram for Total Revenue
plt.hist(df['Total Revenue'],bins=20,edgecolor='Black')
plt.show()
```

```
# pie chart
region_revenue= df.groupby('Region')["Total Revenue"].sum()
plt.pie(region_revenue,labels= region_revenue.index,autopct="%1.1f%%",colors= ['b','c','orange'])
plt.show()
```

```
region_revenue= df.groupby('Payment Method')["Total Revenue"].sum()
plt.pie(region_revenue,labels= region_revenue.index,autopct="%1.1f%%",colors= ['b','c','orange'])
plt.show()
```

```
# seaborn
import seaborn as sns
```

```
sns.histplot(df['Total Revenue'],bins=20)
plt.show()
```

```
# boxplot
plt.figure(figsize=(10,6))
```

```
sns.boxplot(x="Product Category",y="Total Revenue",data=df)
```

```
from numpy import mean
sns.barplot(x="Region",y="Total Revenue",data=df,estimator= mean)
```

```
# plot for units sold for payment methods
sns.barplot(x="Payment Method",y="Units Sold",data=df)
```

```
# count plot
sns.countplot(x="Region",data=df)
```

```
sns.countplot(x="Payment Method",data=df)
```

```
sns.countplot(x="Product Category",data=df)
```

```
# top 10 products based on revenue
top_prod= df.groupby("Product Name")["Total Revenue"].sum().nlargest(10).reset_index()
sns.barplot(data=top_prod,x="Total Revenue",y="Product Name")
```

```
type(top_prod)
```

```
# Average Unit Price By prod Category
avg_price= df.groupby("Product Category")["Unit Price"].mean().reset_index()
sns.barplot(data=avg_price,x="Unit Price",y="Product Category")
```

```
from numpy import mean

plt.figure(figsize=(10,6))

sns.barplot(x = "Product Category", y = "Unit Price", data = df, estimator=mean)

plt.show()
```

```
Start coding or generate with AI.
```