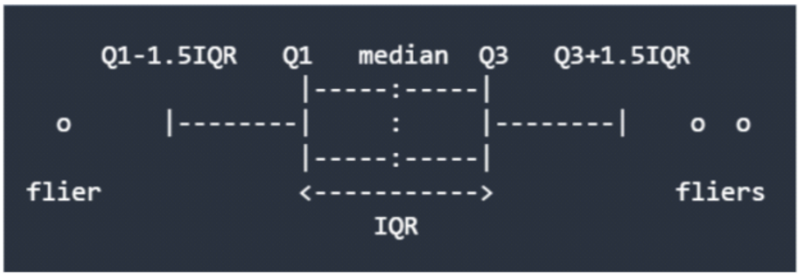
CN-Lec31-Matplotlib & Seaborn - 2

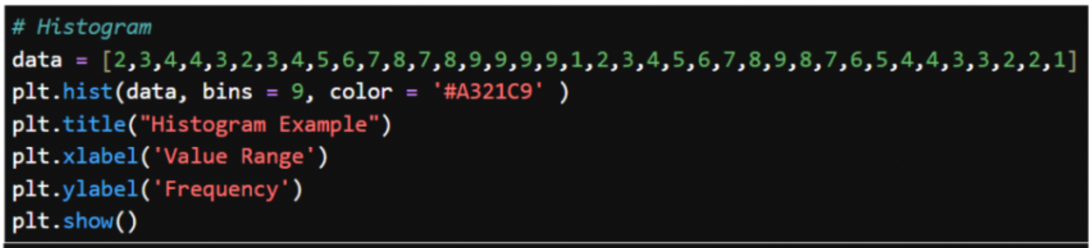
Matplotlib & Seaborn - 2

🎯 Session Objectives:

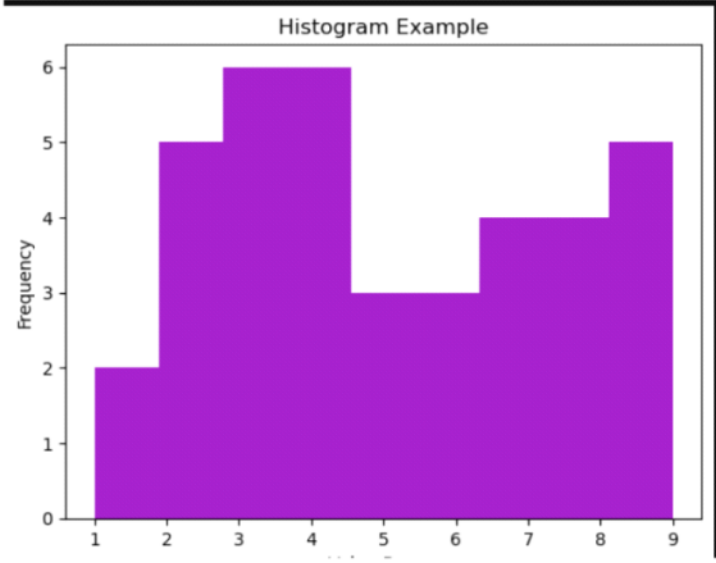
- ✅ Understand what data visualization is and why it matters
- ✅ Use Matplotlib to plot different types of charts
- ✅ Customize plots with markers, colors, linewidth, and line styles
- ✅ Integrate Matplotlib with NumPy and Pandas
- ✅ Understand why Seaborn is important in visualization
- ✅ Recognize common Seaborn plot types

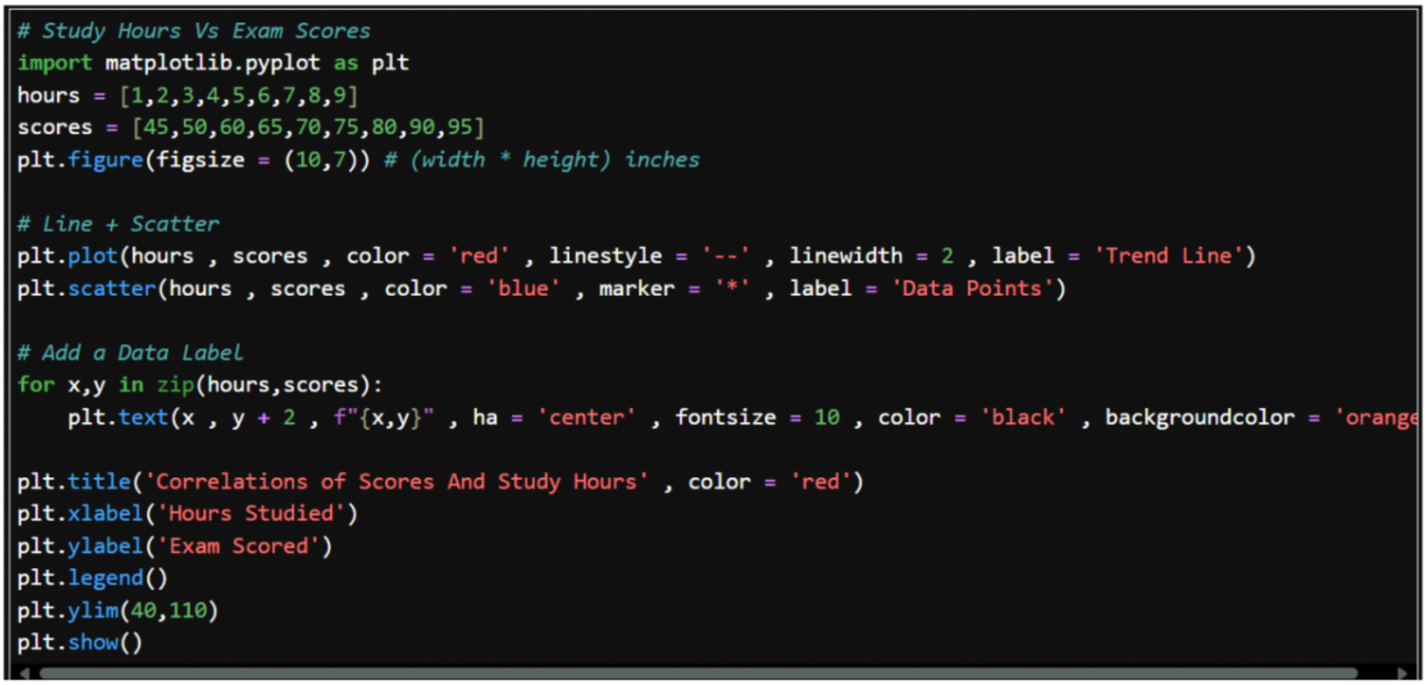


```
     Q1-1.5IQR   Q1   median  Q3   Q3+1.5IQR
                      |-----:-----|
  o      |--------|     :     |--------|    o  o
                      |-----:-----|
flier                 <----------->            fliers
                           IQR
```

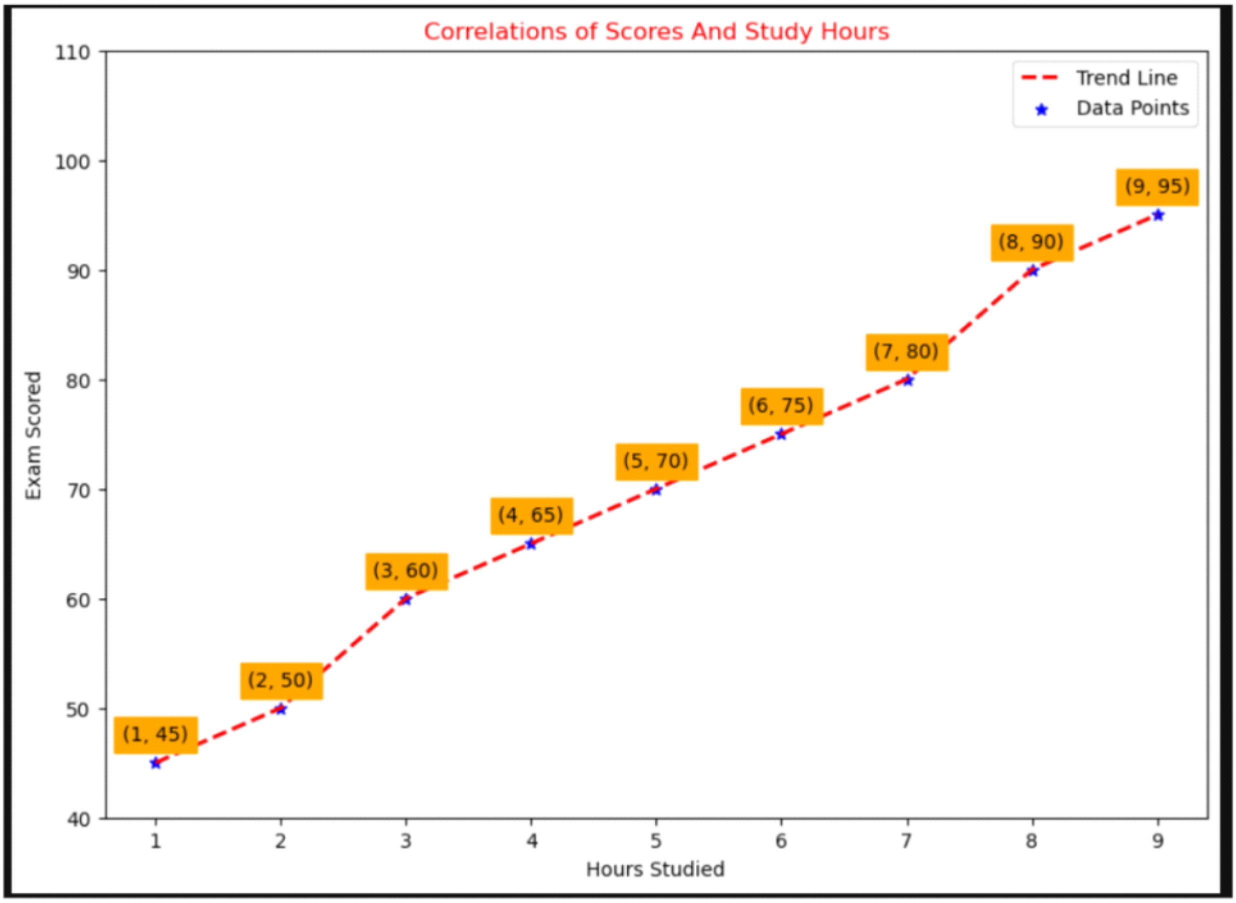


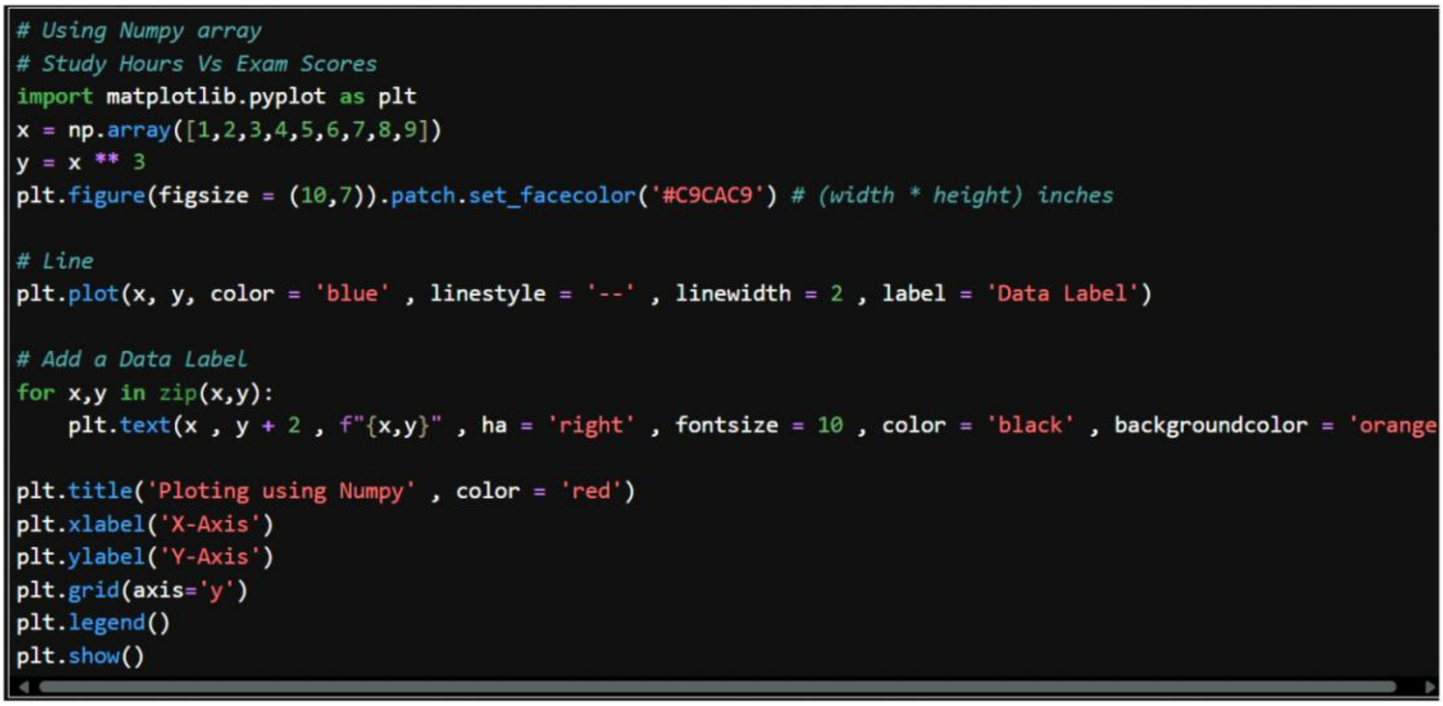
```
# Histogram
data = [2,3,4,4,3,2,3,4,5,6,7,8,7,8,9,9,9,9,1,2,3,4,5,6,7,8,9,8,7,6,5,4,4,3,3,2,2,1]
plt.hist(data, bins = 9, color = '#A321C9' )
plt.title("Histogram Example")
plt.xlabel('Value Range')
plt.ylabel('Frequency')
plt.show()
```



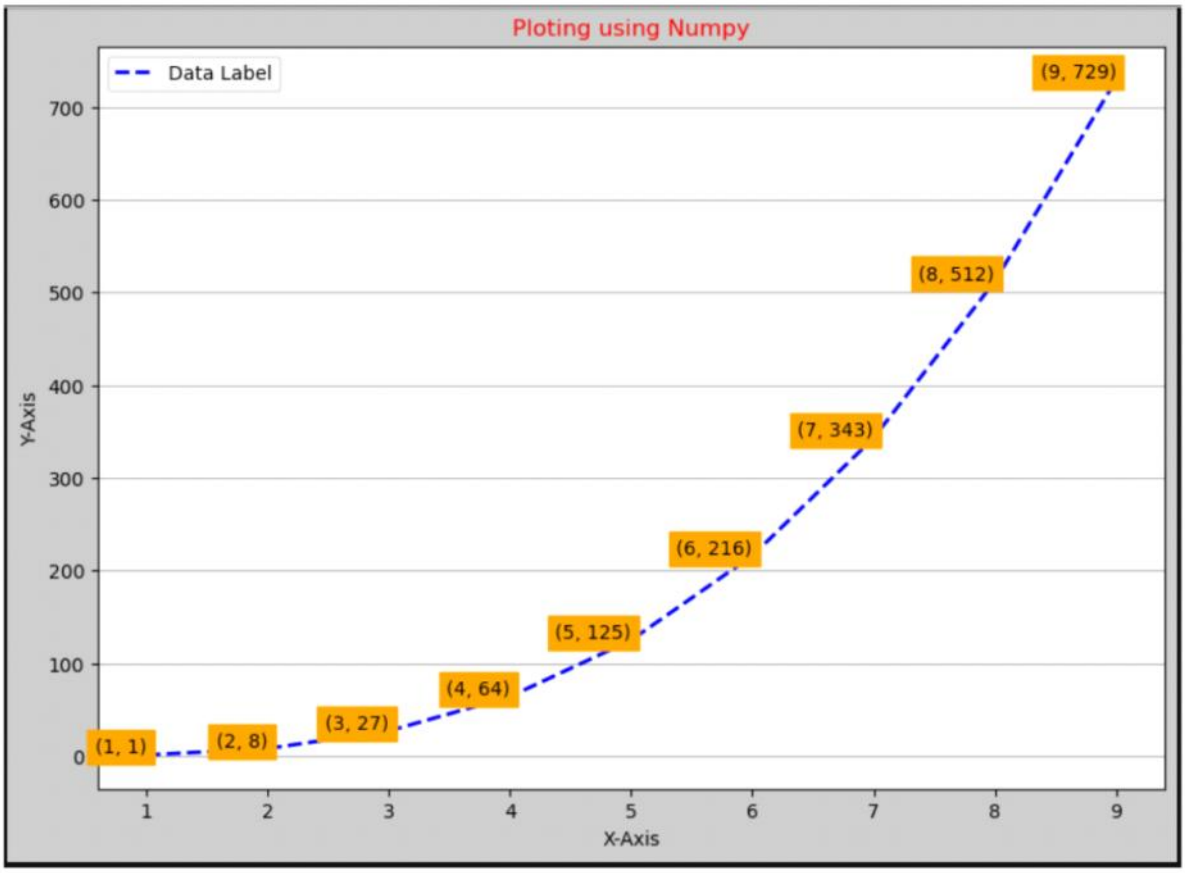


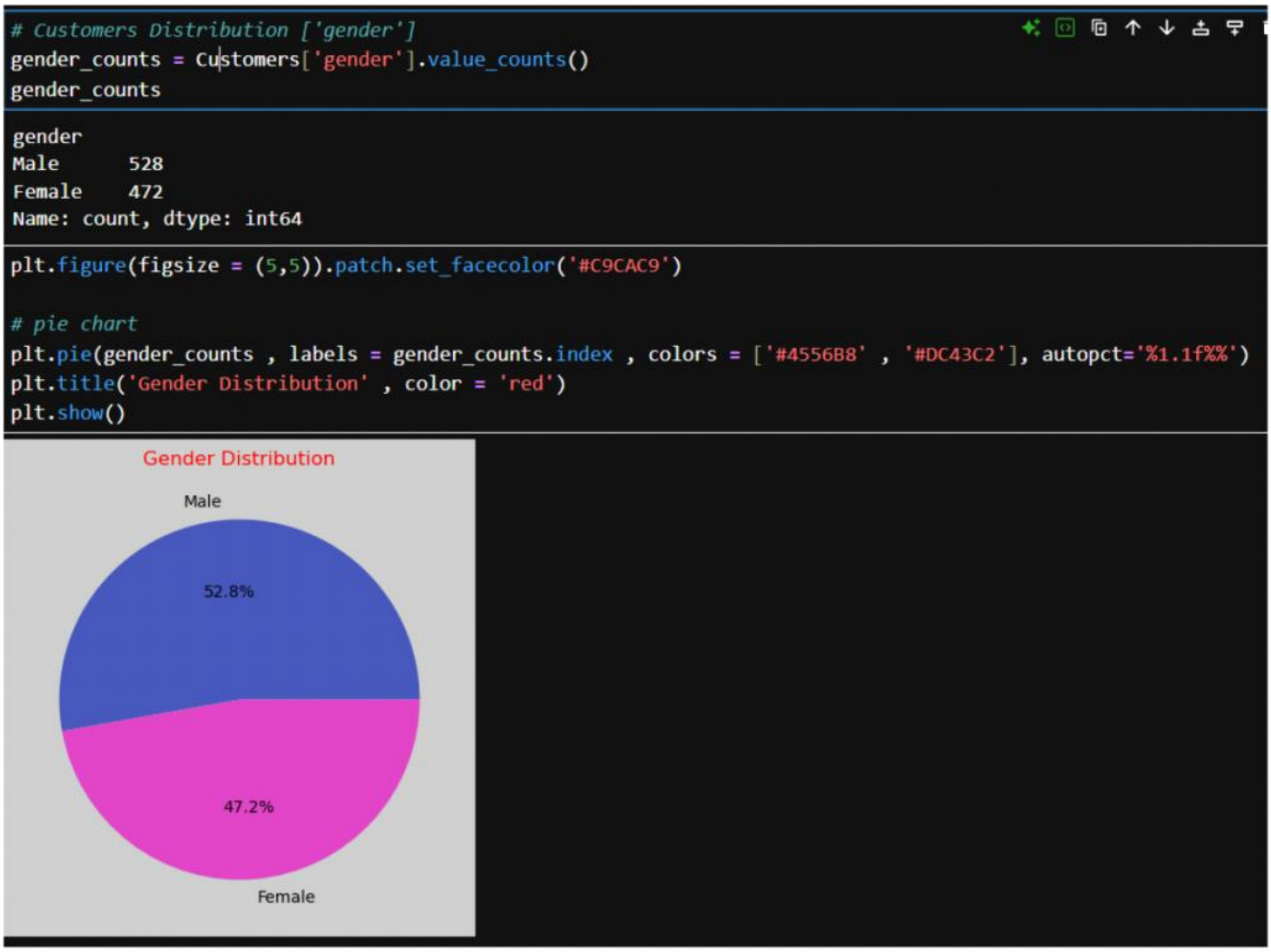
```python
# Study Hours Vs Exam Scores
import matplotlib.pyplot as plt
hours = [1,2,3,4,5,6,7,8,9]
scores = [45,50,60,65,70,75,80,90,95]
plt.figure(figsize = (10,7)) # (width * height) inches

# Line + Scatter
plt.plot(hours , scores , color = 'red' , linestyle = '--' , linewidth = 2 , label = 'Trend Line')
plt.scatter(hours , scores , color = 'blue' , marker = '*' , label = 'Data Points')

# Add a Data Label
for x,y in zip(hours,scores):
    plt.text(x , y + 2 , f"{x,y}" , ha = 'center' , fontsize = 10 , color = 'black' , backgroundcolor = 'orange

plt.title('Correlations of Scores And Study Hours' , color = 'red')
plt.xlabel('Hours Studied')
plt.ylabel('Exam Scored')
plt.legend()
plt.ylim(40,110)
plt.show()
```





```
# Using Numpy array
# Study Hours Vs Exam Scores
import matplotlib.pyplot as plt
x = np.array([1,2,3,4,5,6,7,8,9])
y = x ** 3
plt.figure(figsize = (10,7)).patch.set_facecolor('#C9CAC9') # (width * height) inches

# Line
plt.plot(x, y, color = 'blue' , linestyle = '--' , linewidth = 2 , label = 'Data Label')

# Add a Data Label
for x,y in zip(x,y):
    plt.text(x , y + 2 , f"{x,y}" , ha = 'right' , fontsize = 10 , color = 'black' , backgroundcolor = 'orange

plt.title('Ploting using Numpy' , color = 'red')
plt.xlabel('X-Axis')
plt.ylabel('Y-Axis')
plt.grid(axis='y')
plt.legend()
plt.show()
```

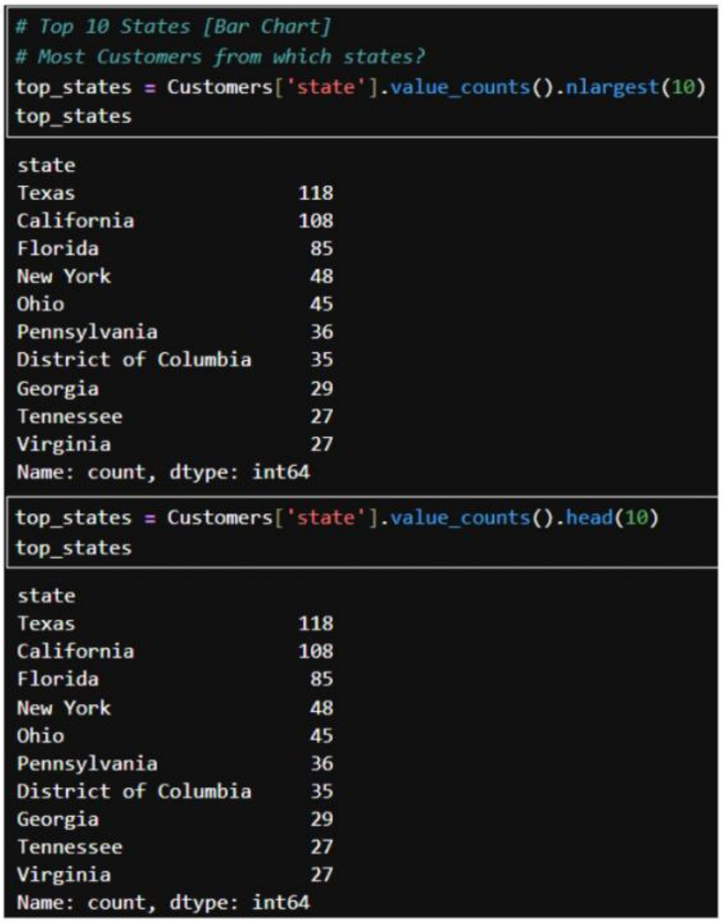




```
# Customers Distribution ['gender']
gender_counts = Customers['gender'].value_counts()
gender_counts
```

```
gender
Male      528
Female    472
Name: count, dtype: int64
```

```
plt.figure(figsize = (5,5)).patch.set_facecolor('#C9CAC9')

# pie chart
plt.pie(gender_counts , labels = gender_counts.index , colors = ['#4556B8' , '#DC43C2'], autopct='%1.1f%%')
plt.title('Gender Distribution' , color = 'red')
plt.show()
```

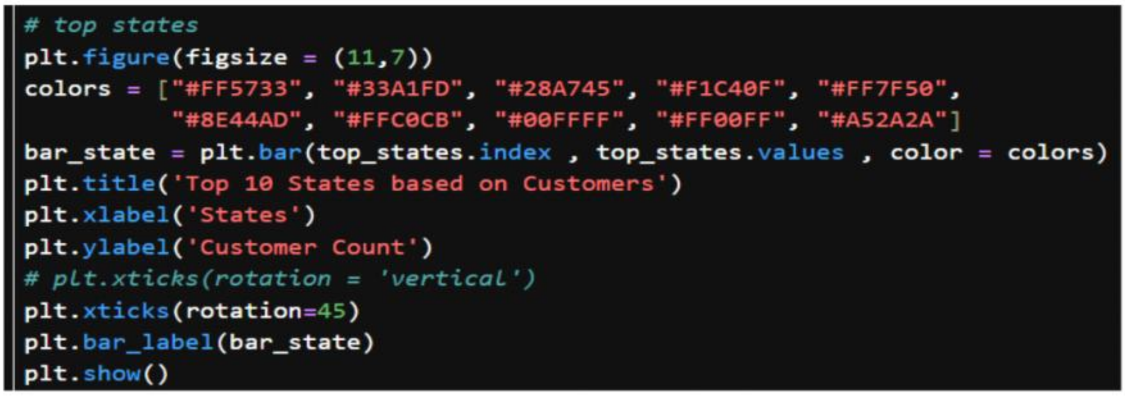


```
# Top 10 States [Bar Chart]
# Most Customers from which states?
top_states = Customers['state'].value_counts().nlargest(10)
top_states
```

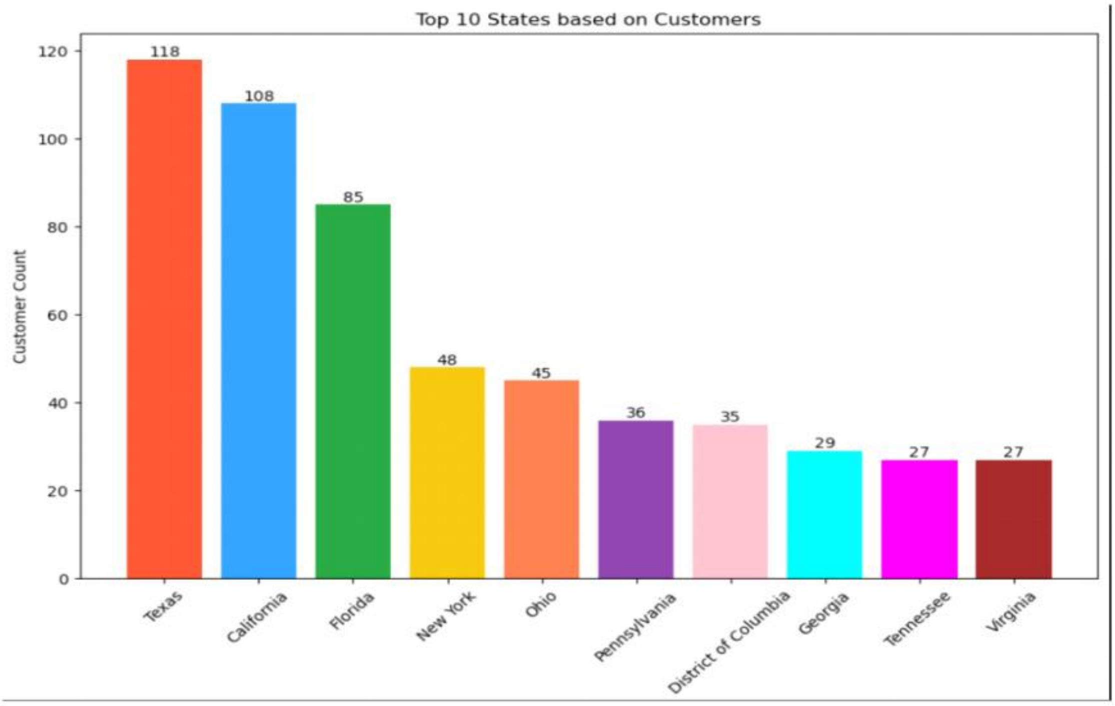
```
state
Texas                   118
California              108
Florida                  85
New York                 48
Ohio                     45
Pennsylvania             36
District of Columbia     35
Georgia                  29
Tennessee                27
Virginia                 27
Name: count, dtype: int64
```

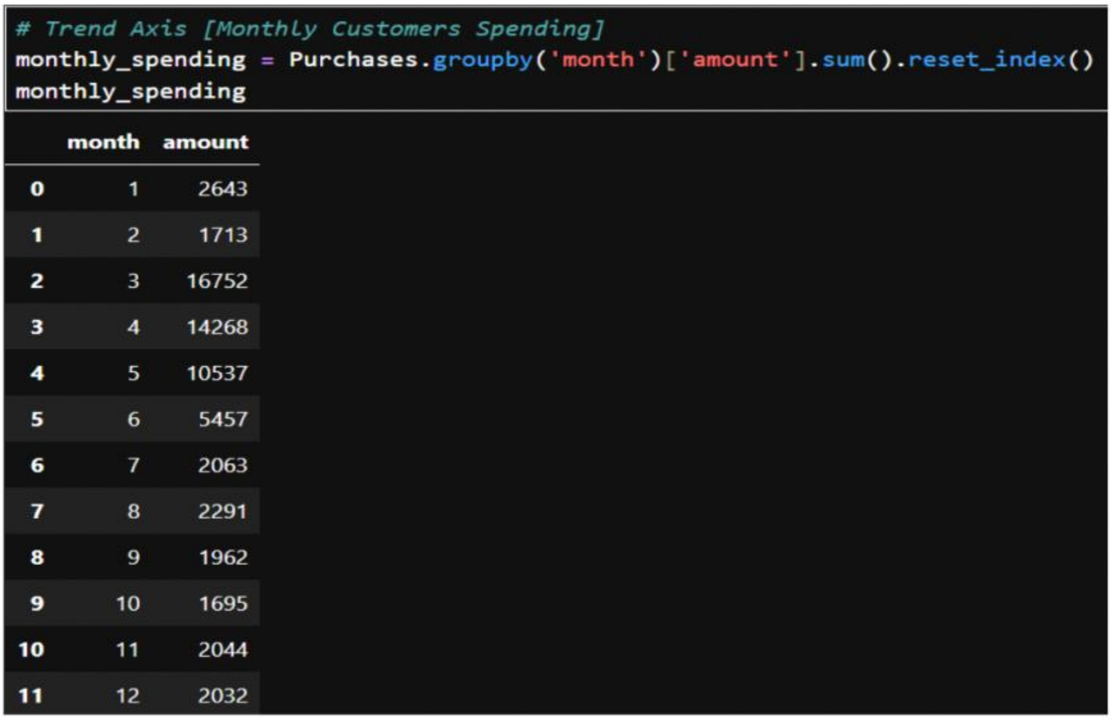
```
top_states = Customers['state'].value_counts().head(10)
top_states
```

```
state
Texas                   118
California              108
Florida                  85
New York                 48
Ohio                     45
Pennsylvania             36
District of Columbia     35
Georgia                  29
Tennessee                27
Virginia                 27
Name: count, dtype: int64
```



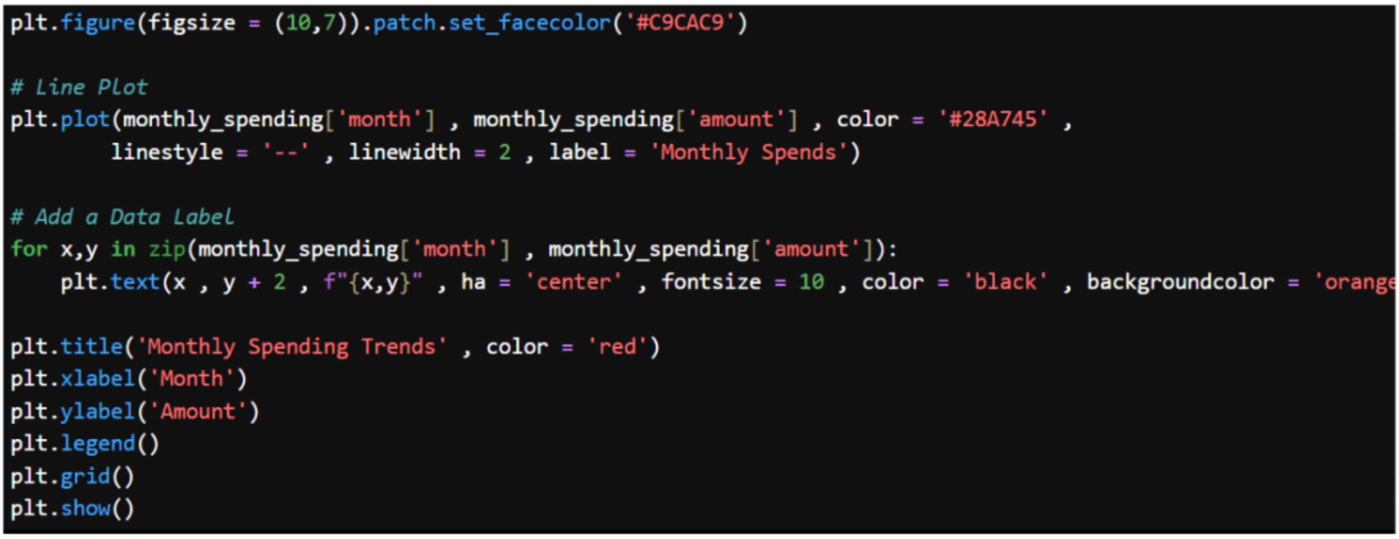
```
# top states
plt.figure(figsize = (11,7))
colors = ["#FF5733", "#33A1FD", "#28A745", "#F1C40F", "#FF7F50",
          "#8E44AD", "#FFC0CB", "#00FFFF", "#FF00FF", "#A52A2A"]
bar_state = plt.bar(top_states.index , top_states.values , color = colors)
plt.title('Top 10 States based on Customers')
plt.xlabel('States')
plt.ylabel('Customer Count')
# plt.xticks(rotation = 'vertical')
plt.xticks(rotation=45)
plt.bar_label(bar_state)
plt.show()
```



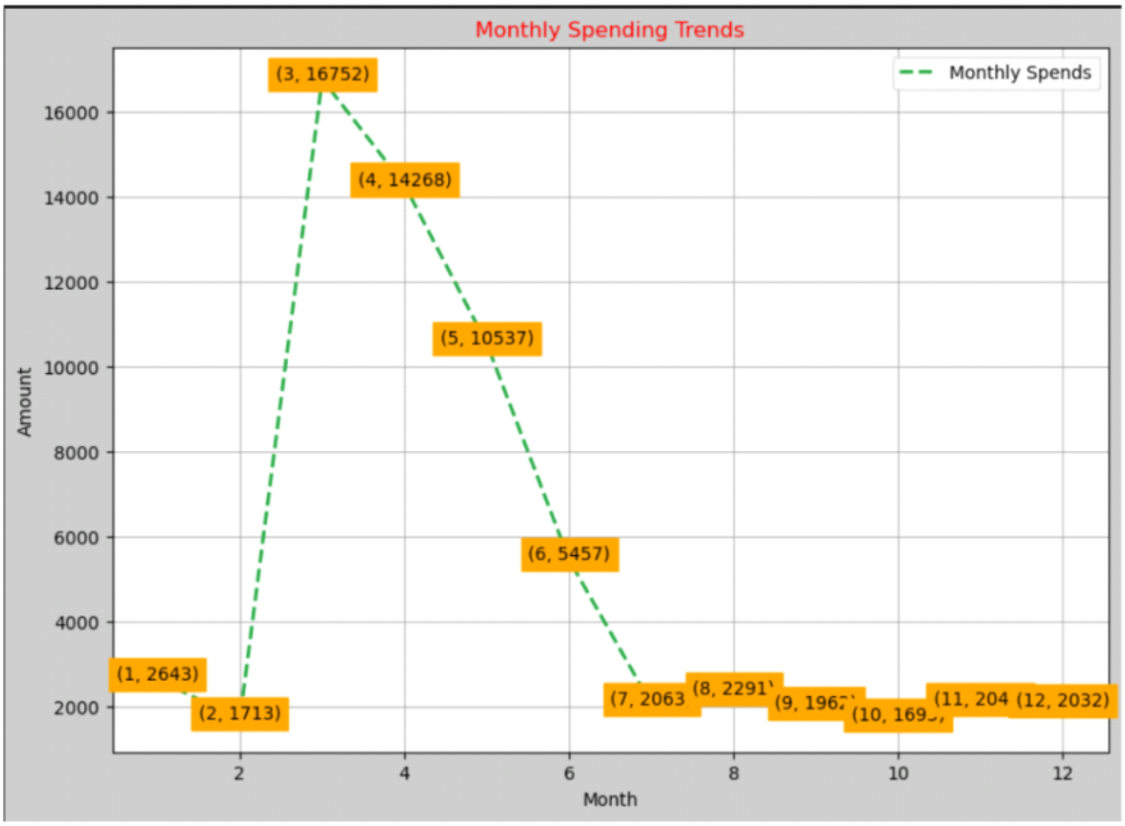


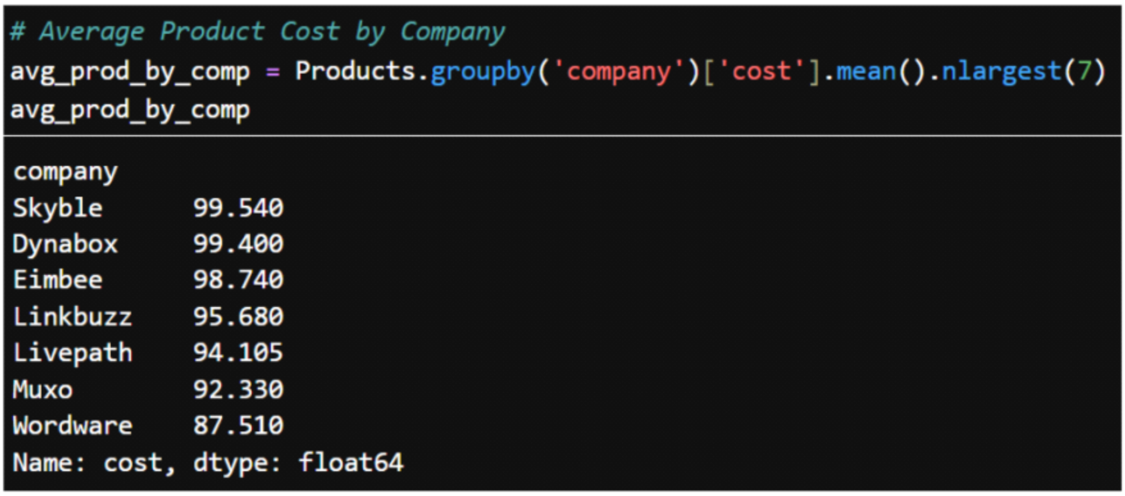
```
# Trend Axis [Monthly Customers Spending]
monthly_spending = Purchases.groupby('month')['amount'].sum().reset_index()
monthly_spending
```

	month	amount
0	1	2643
1	2	1713
2	3	16752
3	4	14268
4	5	10537
5	6	5457
6	7	2063
7	8	2291
8	9	1962
9	10	1695
10	11	2044
11	12	2032



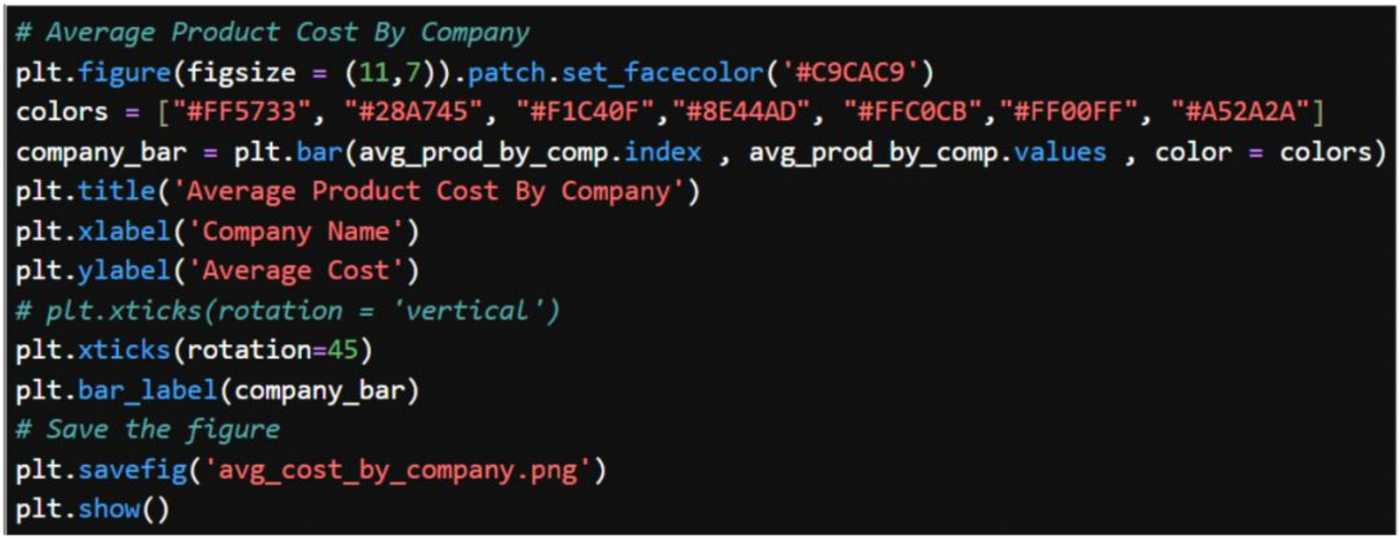
```python
plt.figure(figsize = (10,7)).patch.set_facecolor('#C9CAC9')

# Line Plot
plt.plot(monthly_spending['month'] , monthly_spending['amount'] , color = '#28A745' ,
        linestyle = '--' , linewidth = 2 , label = 'Monthly Spends')

# Add a Data Label
for x,y in zip(monthly_spending['month'] , monthly_spending['amount']):
    plt.text(x , y + 2 , f"{x,y}" , ha = 'center' , fontsize = 10 , color = 'black' , backgroundcolor = 'orange

plt.title('Monthly Spending Trends' , color = 'red')
plt.xlabel('Month')
plt.ylabel('Amount')
plt.legend()
plt.grid()
plt.show()
```



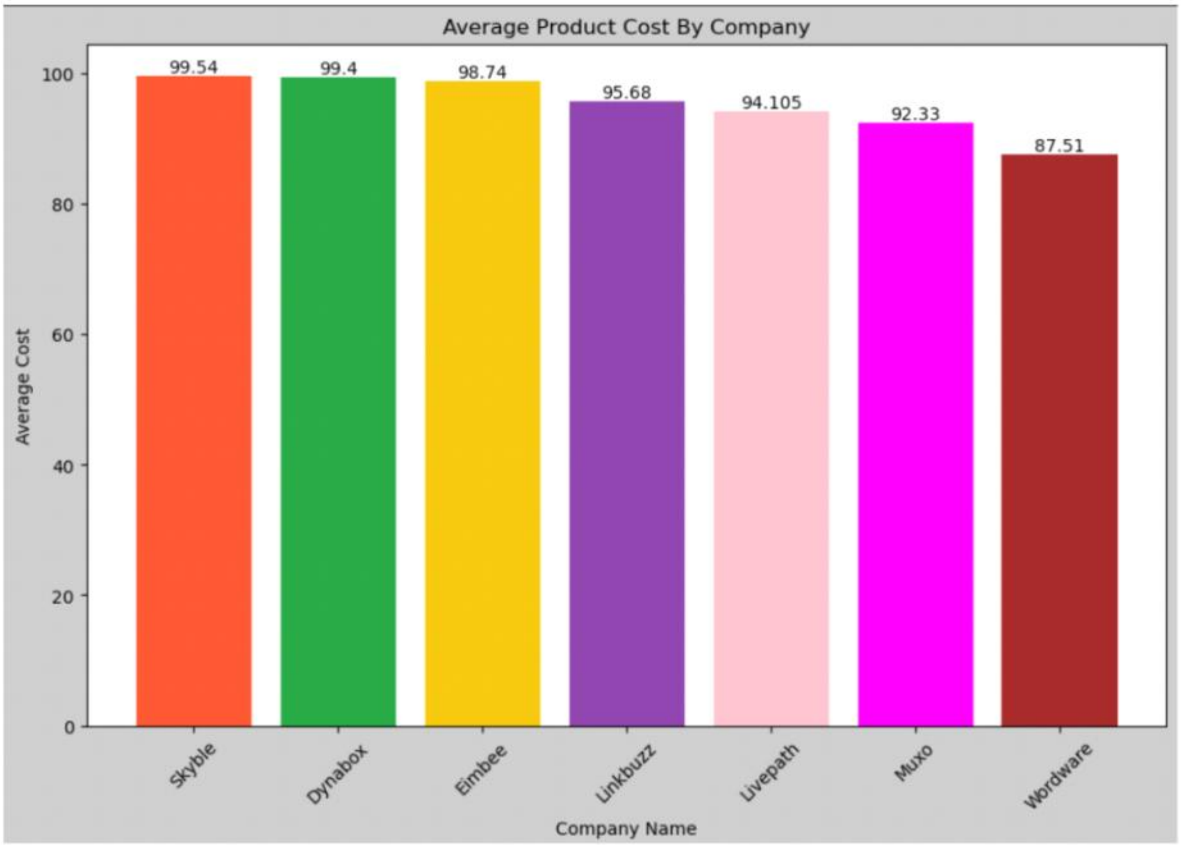


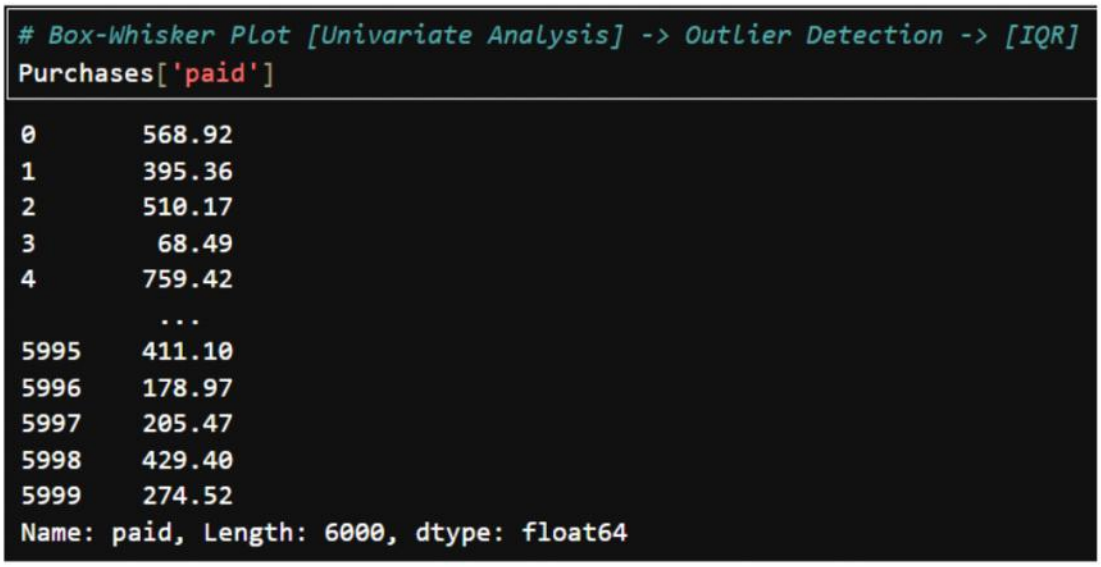
```python
# Average Product Cost by Company
avg_prod_by_comp = Products.groupby('company')['cost'].mean().nlargest(7)
avg_prod_by_comp
```

```
company
Skyble      99.540
Dynabox     99.400
Eimbee      98.740
Linkbuzz    95.680
Livepath    94.105
Muxo        92.330
Wordware    87.510
Name: cost, dtype: float64
```



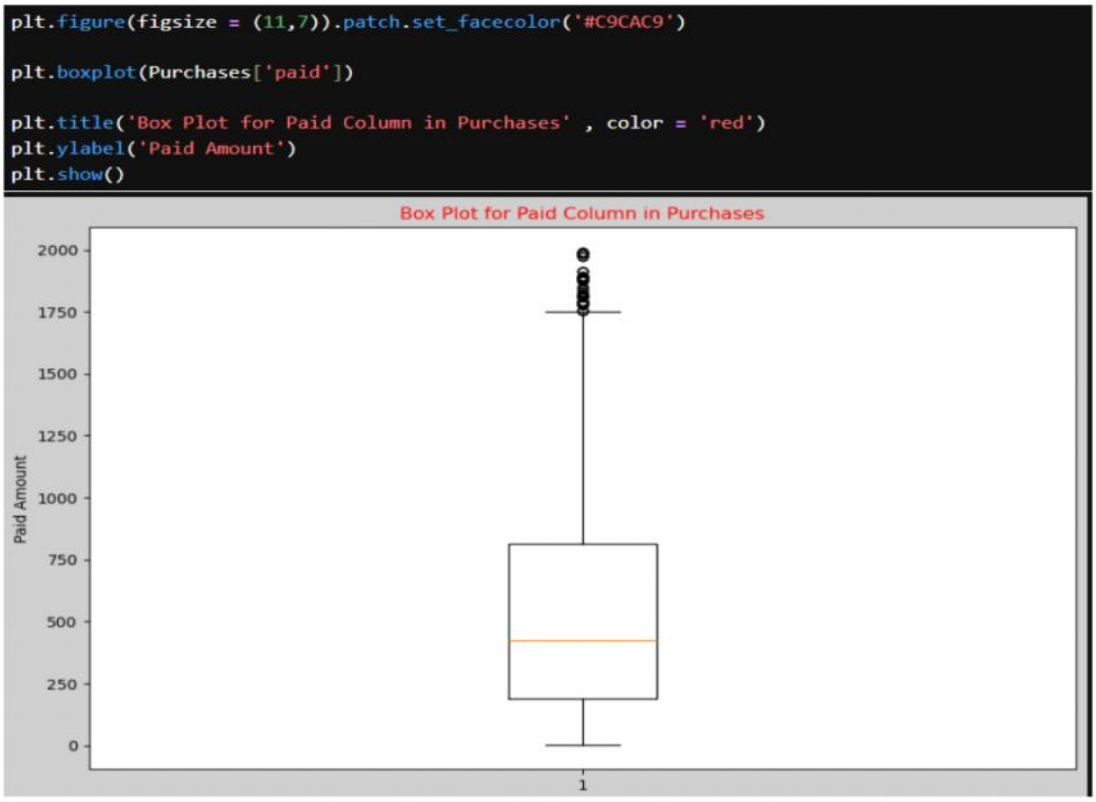
```python
# Average Product Cost By Company
plt.figure(figsize = (11,7)).patch.set_facecolor('#C9CAC9')
colors = ["#FF5733", "#28A745", "#F1C40F","#8E44AD", "#FFC0CB","#FF00FF", "#A52A2A"]
company_bar = plt.bar(avg_prod_by_comp.index , avg_prod_by_comp.values , color = colors)
plt.title('Average Product Cost By Company')
plt.xlabel('Company Name')
plt.ylabel('Average Cost')
# plt.xticks(rotation = 'vertical')
plt.xticks(rotation=45)
plt.bar_label(company_bar)
# Save the figure
plt.savefig('avg_cost_by_company.png')
plt.show()
```



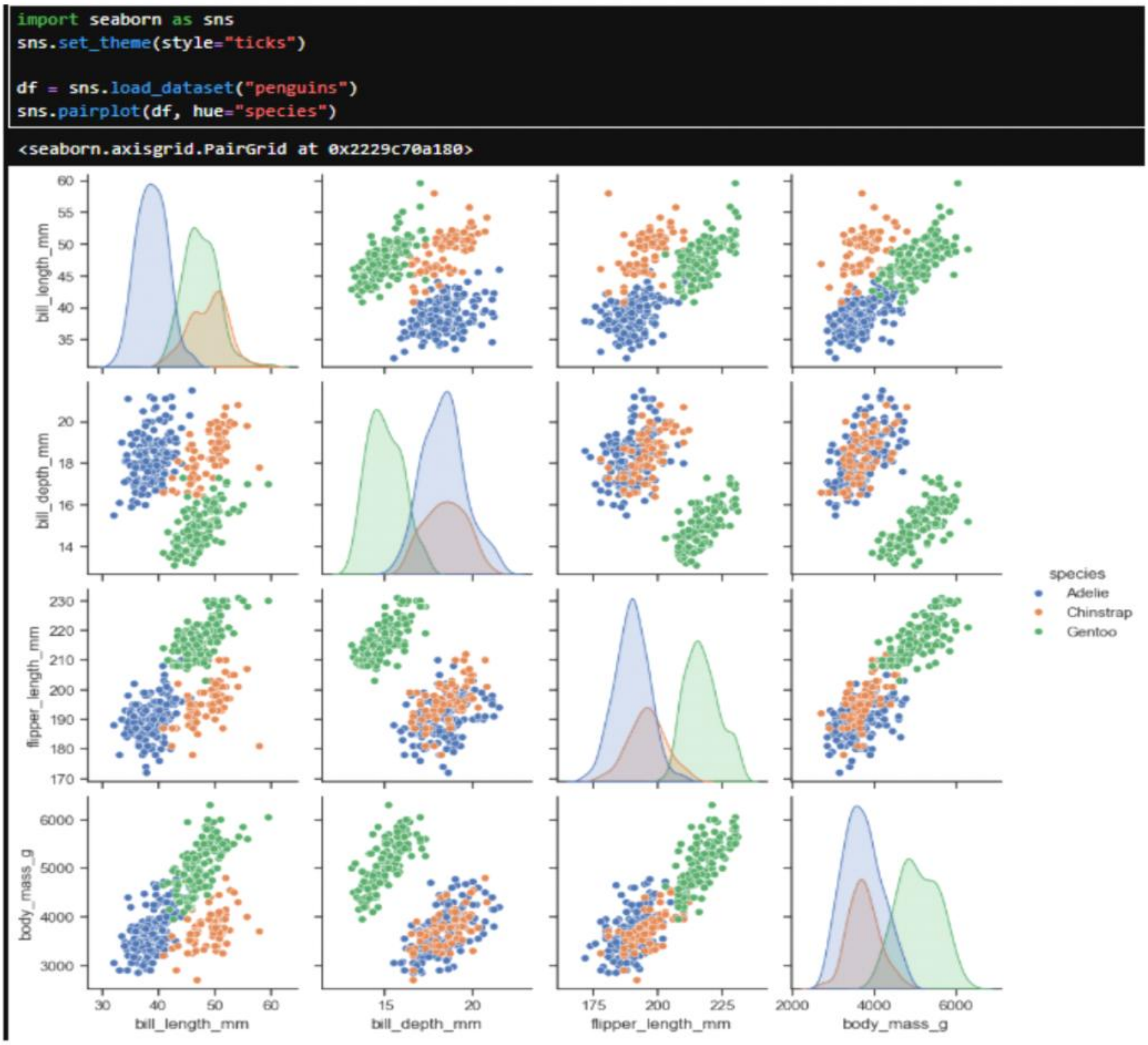


```python
# Box-Whisker Plot [Univariate Analysis] -> Outlier Detection -> [IQR]
Purchases['paid']
```

```
0       568.92
1       395.36
2       510.17
3        68.49
4       759.42
         ...
5995    411.10
5996    178.97
5997    205.47
5998    429.40
5999    274.52
Name: paid, Length: 6000, dtype: float64
```

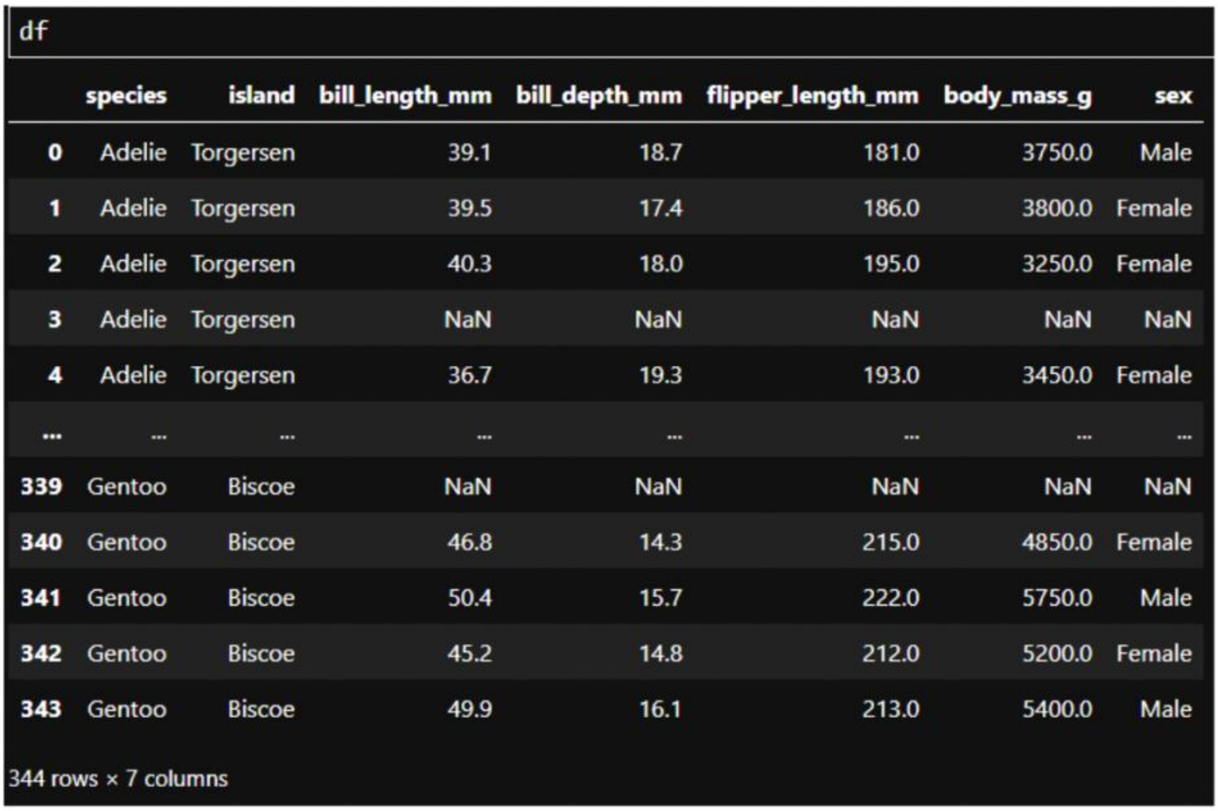


```
plt.figure(figsize = (11,7)).patch.set_facecolor('#C9CAC9')

plt.boxplot(Purchases['paid'])

plt.title('Box Plot for Paid Column in Purchases' , color = 'red')
plt.ylabel('Paid Amount')
plt.show()
```



```
import seaborn as sns
sns.set_theme(style="ticks")

df = sns.load_dataset("penguins")
sns.pairplot(df, hue="species")
```

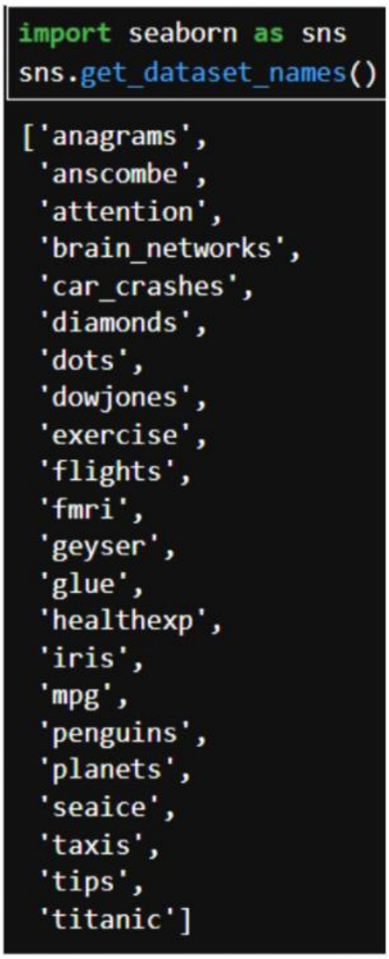
```
<seaborn.axisgrid.PairGrid at 0x2229c70a180>
```



```
df
```

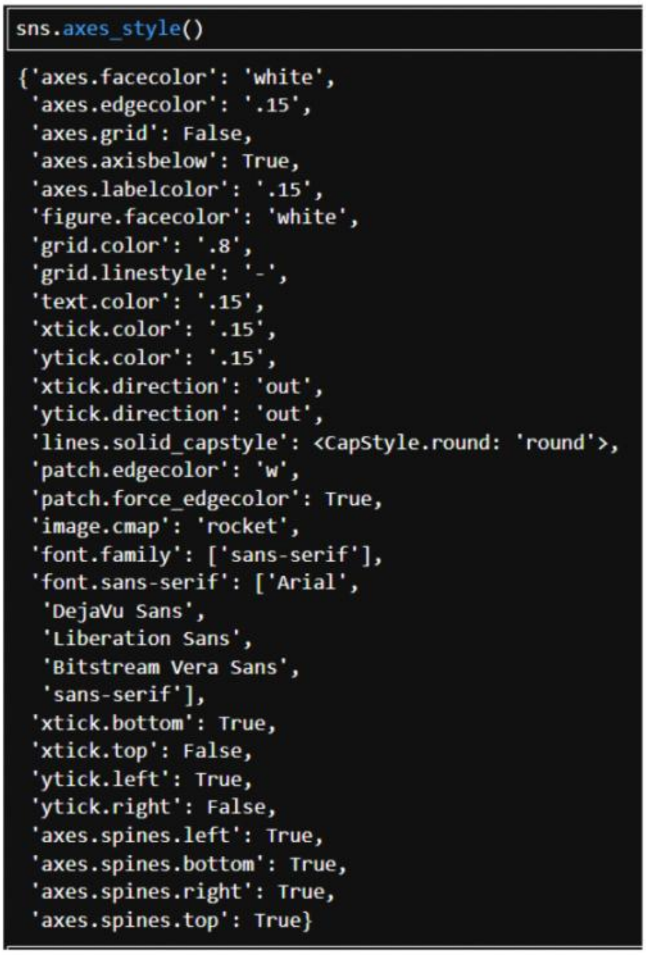
	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	Male
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	Female
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	Female
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	Female
...	...	...	...	...	...	...	...
339	Gentoo	Biscoe	NaN	NaN	NaN	NaN	NaN
340	Gentoo	Biscoe	46.8	14.3	215.0	4850.0	Female
341	Gentoo	Biscoe	50.4	15.7	222.0	5750.0	Male
342	Gentoo	Biscoe	45.2	14.8	212.0	5200.0	Female
343	Gentoo	Biscoe	49.9	16.1	213.0	5400.0	Male

344 rows × 7 columns



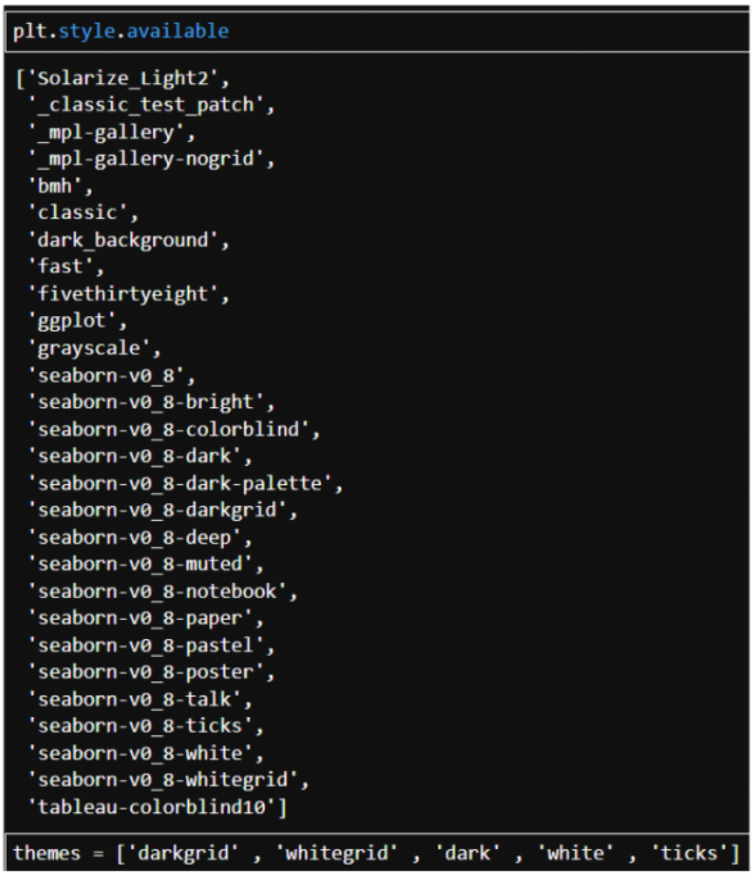
```
import seaborn as sns
sns.get_dataset_names()
```

```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'dowjones',
 'exercise',
 'flights',
 'fmri',
 'geyser',
 'glue',
 'healthexp',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'seaice',
 'taxis',
 'tips',
 'titanic']
```



```
sns.axes_style()
```

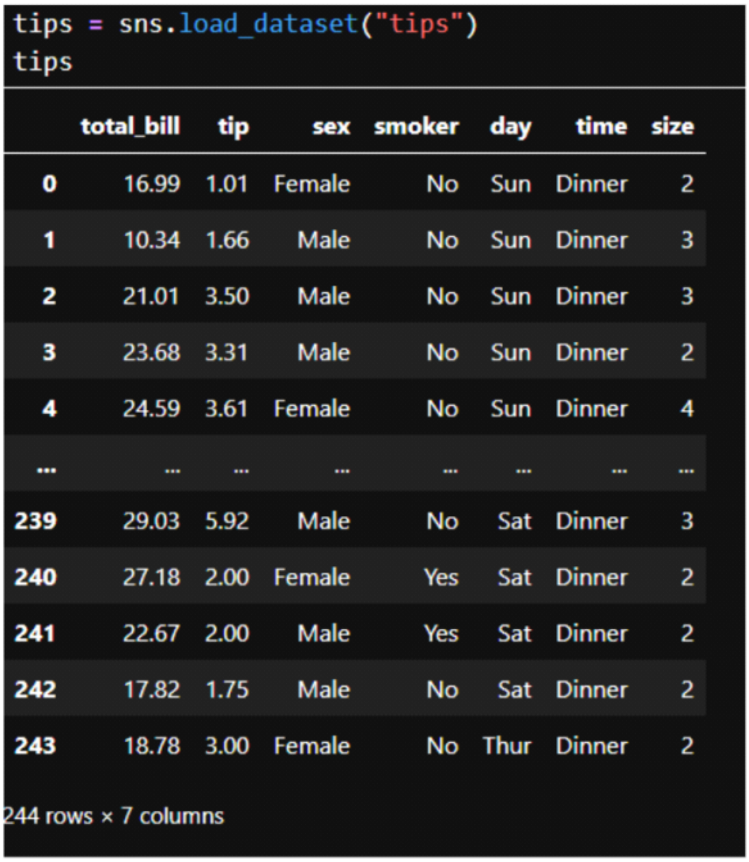
```
{'axes.facecolor': 'white',
 'axes.edgecolor': '.15',
 'axes.grid': False,
 'axes.axisbelow': True,
 'axes.labelcolor': '.15',
 'figure.facecolor': 'white',
 'grid.color': '.8',
 'grid.linestyle': '-',
 'text.color': '.15',
 'xtick.color': '.15',
 'ytick.color': '.15',
 'xtick.direction': 'out',
 'ytick.direction': 'out',
 'lines.solid_capstyle': <CapStyle.round: 'round'>,
 'patch.edgecolor': 'w',
 'patch.force_edgecolor': True,
 'image.cmap': 'rocket',
 'font.family': ['sans-serif'],
 'font.sans-serif': ['Arial',
  'DejaVu Sans',
  'Liberation Sans',
  'Bitstream Vera Sans',
  'sans-serif'],
 'xtick.bottom': True,
 'xtick.top': False,
 'ytick.left': True,
 'ytick.right': False,
 'axes.spines.left': True,
 'axes.spines.bottom': True,
 'axes.spines.right': True,
 'axes.spines.top': True}
```



```
plt.style.available
```

```
['Solarize_Light2',
 '_classic_test_patch',
 '_mpl-gallery',
 '_mpl-gallery-nogrid',
 'bmh',
 'classic',
 'dark_background',
 'fast',
 'fivethirtyeight',
 'ggplot',
 'grayscale',
 'seaborn-v0_8',
 'seaborn-v0_8-bright',
 'seaborn-v0_8-colorblind',
 'seaborn-v0_8-dark',
 'seaborn-v0_8-dark-palette',
 'seaborn-v0_8-darkgrid',
 'seaborn-v0_8-deep',
 'seaborn-v0_8-muted',
 'seaborn-v0_8-notebook',
 'seaborn-v0_8-paper',
 'seaborn-v0_8-pastel',
 'seaborn-v0_8-poster',
 'seaborn-v0_8-talk',
 'seaborn-v0_8-ticks',
 'seaborn-v0_8-white',
 'seaborn-v0_8-whitegrid',
 'tableau-colorblind10']
```

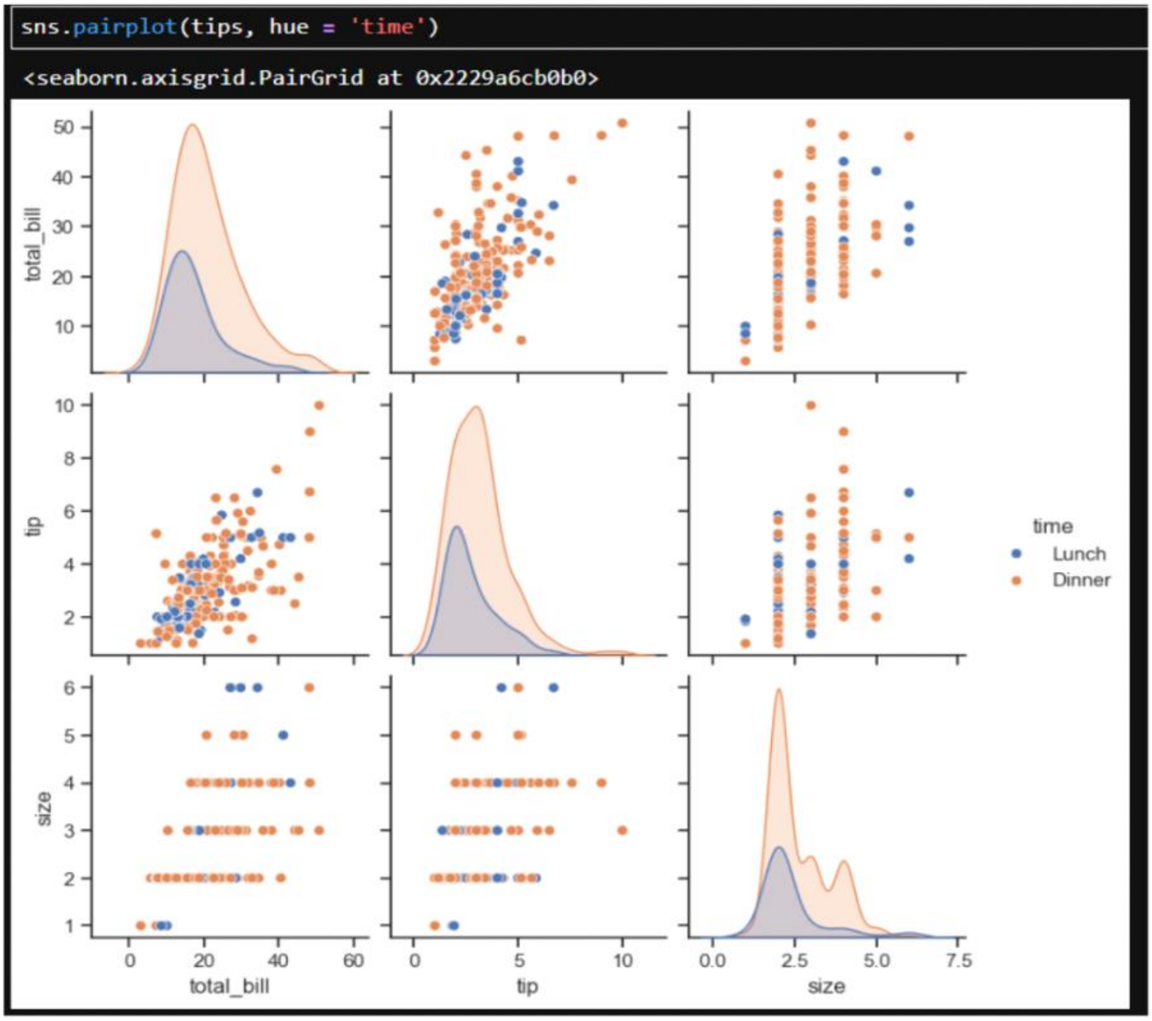
```
themes = ['darkgrid' , 'whitegrid' , 'dark' , 'white' , 'ticks']
```



```
tips = sns.load_dataset("tips")
tips
```

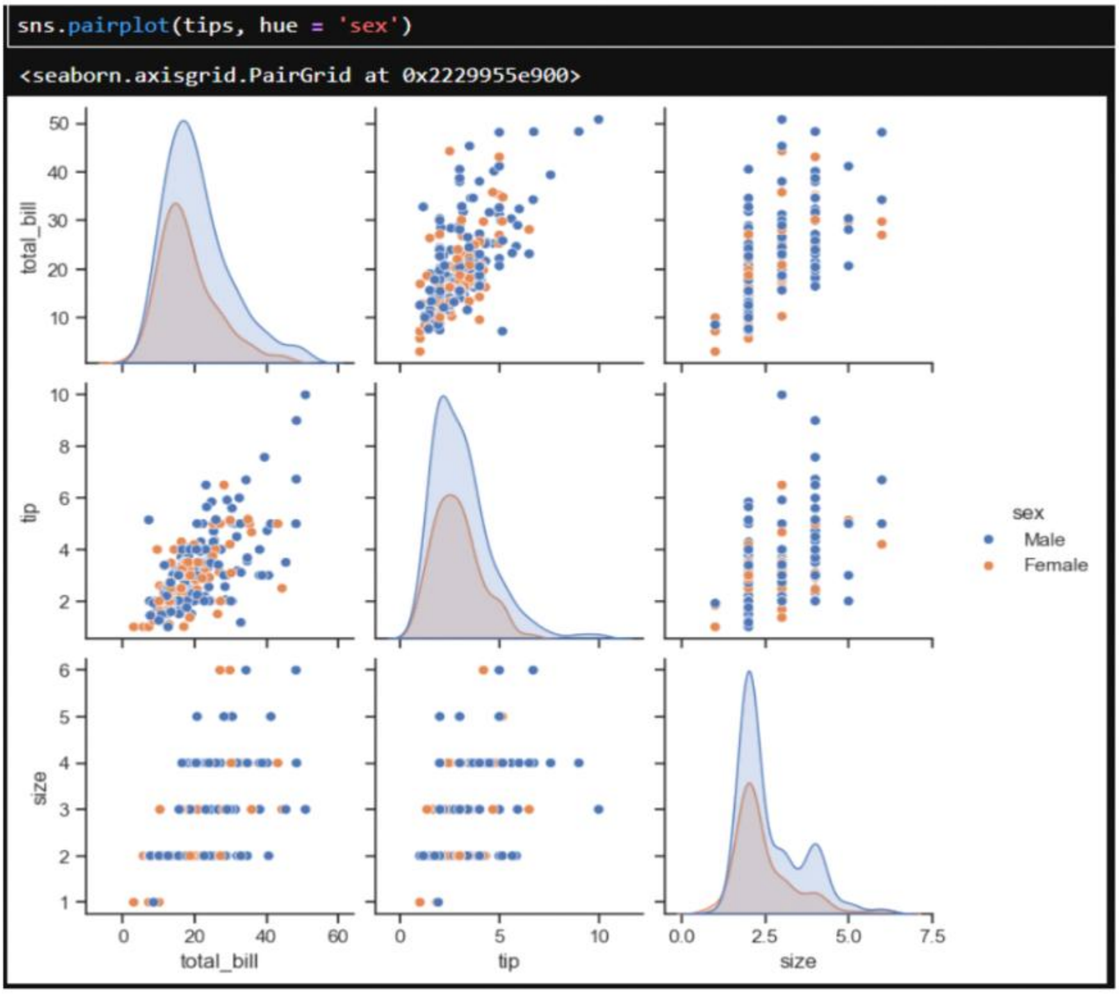
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns



```
sns.pairplot(tips, hue = 'time')
```

```
<seaborn.axisgrid.PairGrid at 0x2229a6cb0b0>
```



```
sns.pairplot(tips, hue = 'sex')
```

```
<seaborn.axisgrid.PairGrid at 0x2229955e900>
```