Categorical Variables

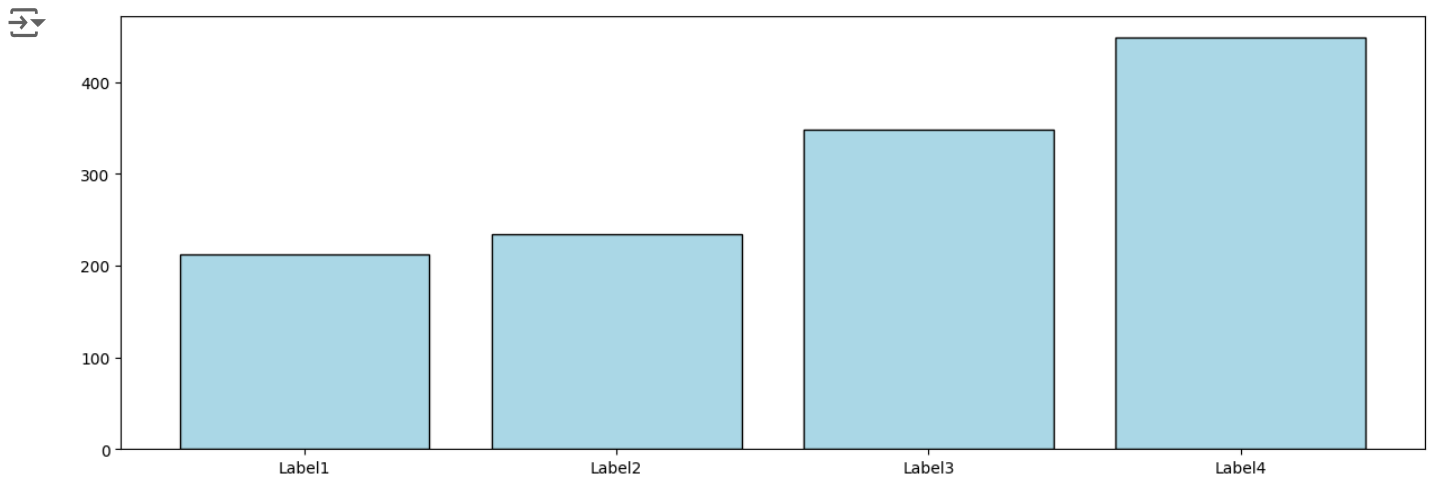
```
import pandas as pd
import matplotlib.pyplot as plt


from google.colab import drive
import os

drive.mount('/content/drive')
os.chdir('/content/drive/MyDrive/')
for item in os.listdir():
  print(item)
print("--------------")
os.chdir('/content/drive/MyDrive/cloud/GitHub/AdvDataViz/Notebooks/')
for item in os.listdir():
  print(item)
print("--------------")
notebooks = "/content/drive/MyDrive/cloud/GitHub/AdvDataViz/Notebooks"
print(os.listdir(notebooks))
print("--------------")

file = "heart-disease.csv"
file_path = os.path.join(notebooks, file)
with open(file_path, "r") as f:
  contents = f.read()
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/c
learningStore
healthyCar
startup
cloud
Artificial Intelligence
--------------
03 Matplotlib  - Exercise.ipynb
02 Matplotlib.ipynb
01 Python_Pandas.ipynb
04 Continuous Variables - Histogram .ipynb
05 Continuous Variables - Histogram - Exercise .ipynb
07 Continuous Variables - Boxplot - Exercise .ipynb
03 Matplotlib - Exercise Solutions.ipynb
05 Continuous Variables - Histogram - Exercise Solutions.ipynb
06 Continuous Variables - Boxplot.ipynb
08 Continuous Variables - Scatterplot.ipynb
07 Continuous Variables - Boxplot - Exercise Solutions.ipynb
09 Continuous Variables - Scatterplot - Exercise Solutions.ipynb
09 Continuous Variables - Scatterplot - Exercise .ipynb
10 Categorical Variables - Bar_Pie.ipynb
12 Seaborn.ipynb
11 Pandas Data Visualization.ipynb
13 Seaborn - Exercise .ipynb
Top 50 US Tech Companies.csv
13 Seaborn - Exercise Solution.ipynb
15 Custom Modules.ipynb
14 Functions.ipynb
churn.csv
student_performance.csv
myplotlib.py
--------------
['heart-disease.csv']
--------------
```

```
employee_attrition_.csv
heart-disease.csv
--------------
['03 Matplotlib  - Exercise.ipynb', '02 Matplotlib.ipynb', '01 Python_Pandas.ipynb', '04 Cont
--------------
```

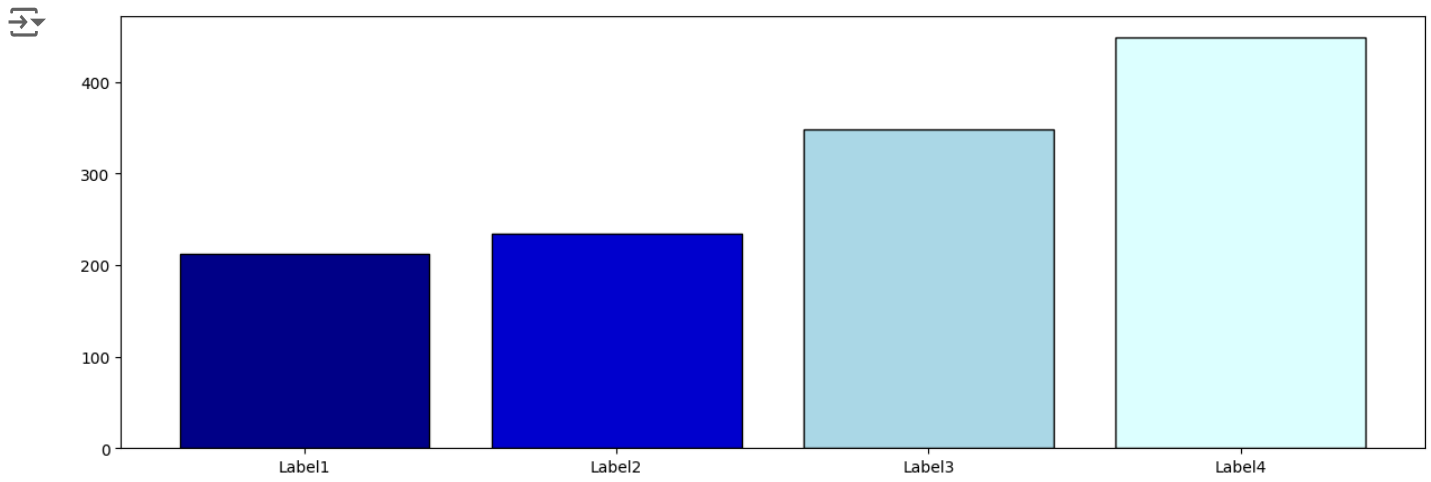
Bar plot

```
fig, ax = plt.subplots(figsize = (15, 5))

      # the labels to go beneath the bars             # the height of each bar
ax.bar(x=["Label1", "Label2","Label3", "Label4"], height=[212, 234, 348, 449],
      color="lightblue", edgecolor="black");
```



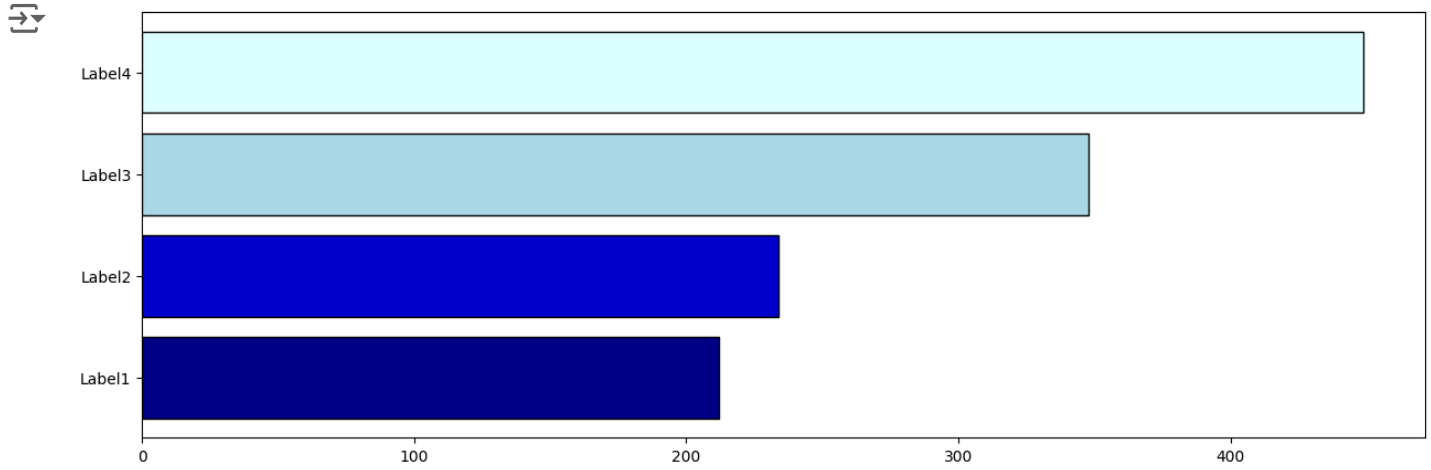
Display each bar with a different color

```
fig, ax = plt.subplots(figsize = (15, 5))

      # the labels to go beneath the bars             # the height of each bar
ax.bar(x=["Label1", "Label2","Label3", "Label4"], height=[212, 234, 348, 449],
      color=["darkblue", "mediumblue", "lightblue", "lightcyan"], edgecolor="black");
```



Horizontal bar plot

```
fig, ax = plt.subplots(figsize = (15, 5))

ax.barh(y=["Label1", "Label2","Label3", "Label4"], width=[212, 234, 348, 449],
        color=["darkblue", "mediumblue", "lightblue", "lightcyan"], edgecolor="black");
```



Dataset: Heart Disease

```
#df = pd.read_csv("heart-disease.csv")
df = pd.read_csv(file_path)

df.head()
```

	age	sex	chest_pain	rest_bp	chol	max_hr	st_depr	heart_disease
0	63	female	3	145	233	150	2.3	1
1	37	female	2	130	250	187	3.5	1
2	41	male	1	130	204	172	1.4	1
3	56	female	1	120	236	178	0.8	1
4	57	male	0	120	354	163	0.6	1

Next steps: Generate code with df | View recommended plots | New interactive sheet

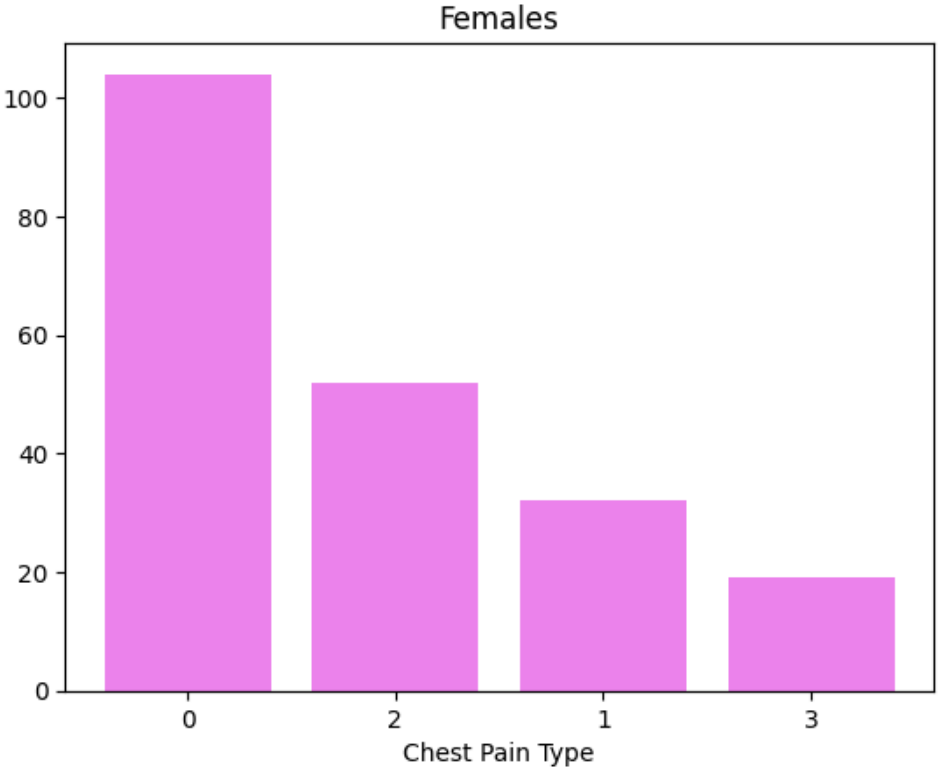
Count of chest pain type for females

```
females_pain = df.loc[df["sex"] == "female", ["chest_pain"]].value_counts()
females_pain
```

chest_pain	count
0	104
2	52
1	32
3	19

dtype: int64

```
fig, ax = plt.subplots()

# the labels to go beneath the bars   # the height of each bar
ax.bar(x = ['0', '2', '1', '3'], height=females_pain, color='violet')

ax.set_xlabel("Chest Pain Type")
ax.set_title("Females");
```



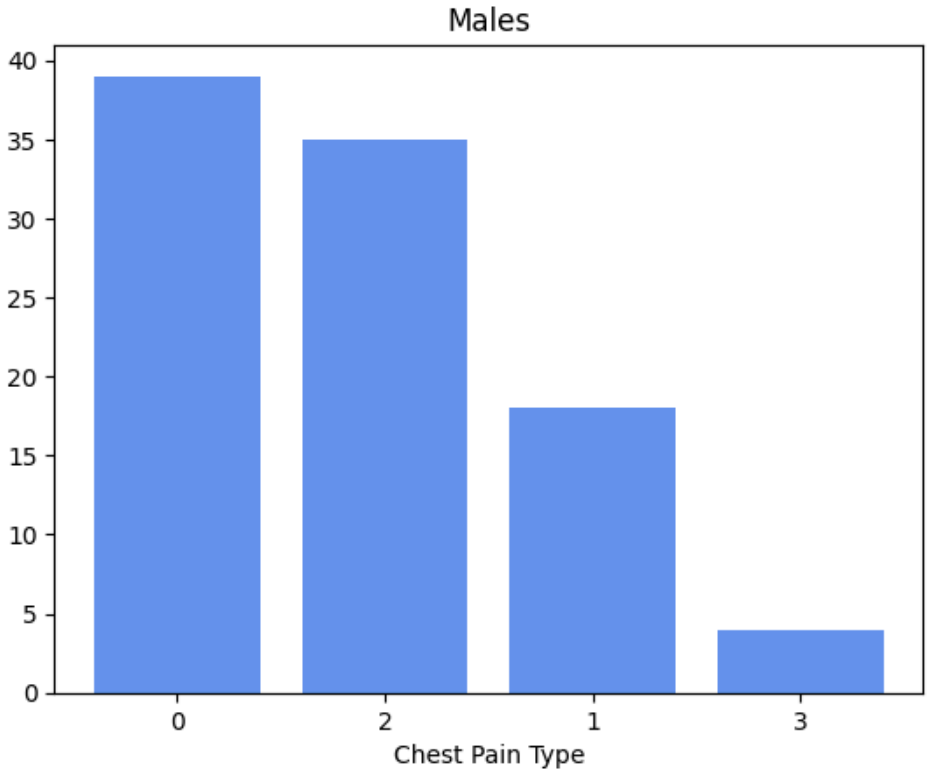
Count of chest pain type for males

```
males_pain = df.loc[df["sex"] == "male", ["chest_pain"]].value_counts()
males_pain
```

chest_pain	count
0	39
2	35
1	18
3	4

dtype: int64

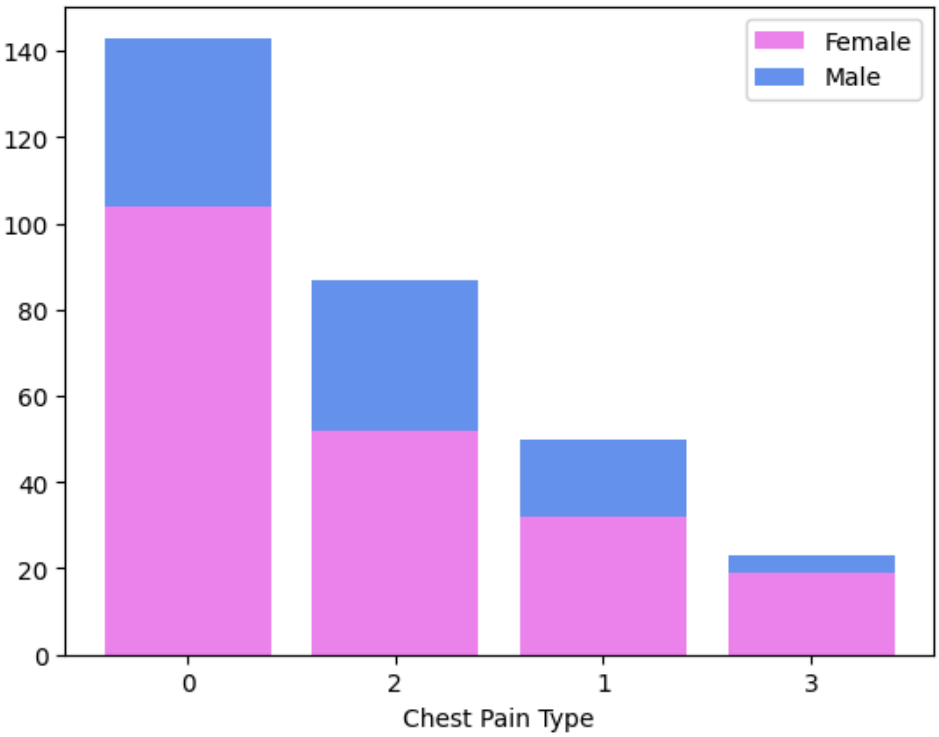
```
fig, ax = plt.subplots()

# the labels to go beneath the bars   # the height of each bar
ax.bar(x = ['0', '2', '1', '3'],  height=males_pain, color='cornflowerblue')

ax.set_xlabel("Chest Pain Type")
ax.set_title("Males");
```



Joint: categorical x categorical

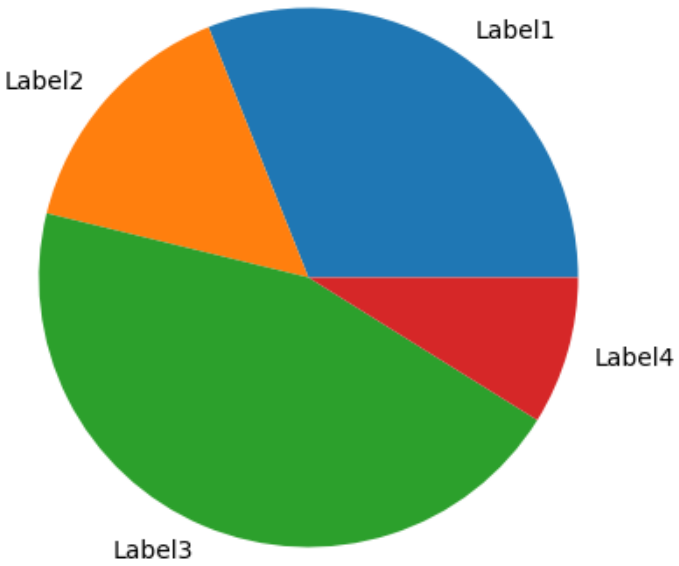
Stacked bar chart with legend

```
fig, ax = plt.subplots()

ax.bar(x = ['0', '2', '1', '3'], height=females_pain, color='violet')
                                                                # set the first plot on the "bottom"
ax.bar(x = ['0', '2', '1', '3'], height=males_pain, color='cornflowerblue', bottom=females_pain)


ax.set_xlabel("Chest Pain Type")

plt.legend(["Female", "Male"]);
```



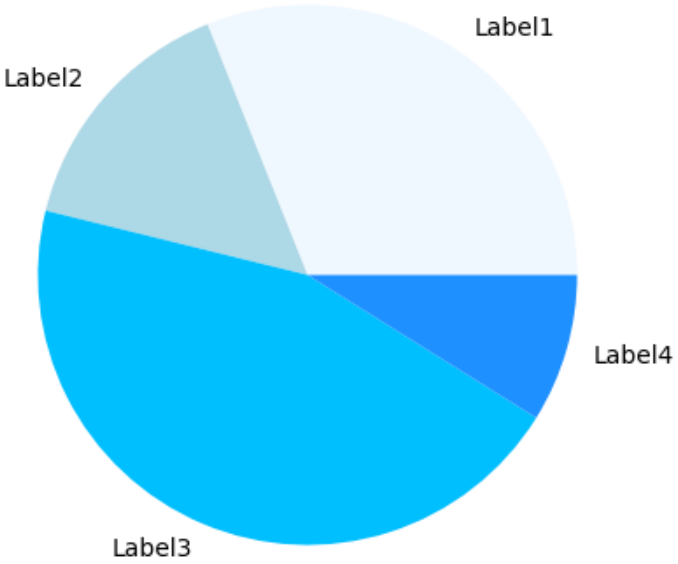
Pie chart

```
fig, ax = plt.subplots(figsize = (15, 5))

       # the value of each slice      # the labels to go with the slices
ax.pie(x=[154, 75, 223, 44], labels = ["Label1", "Label2","Label3", "Label4"]);
```



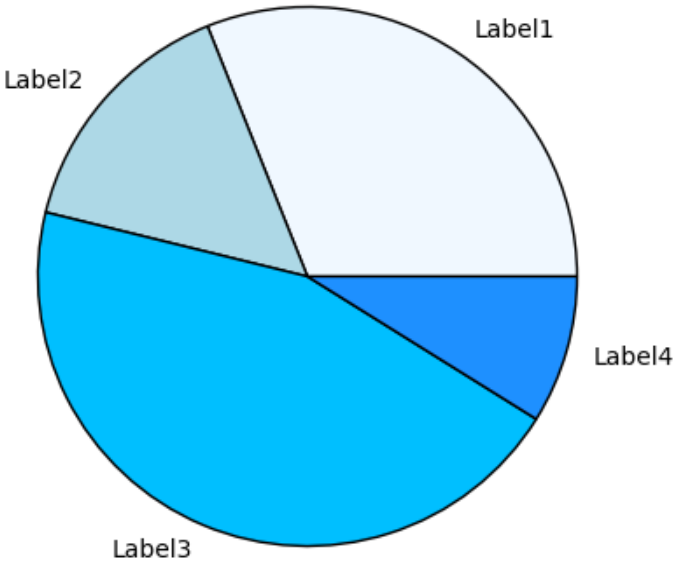
Set the colors

```
fig, ax = plt.subplots(figsize = (15, 5))

ax.pie(x=[154, 75, 223, 44], labels = ["Label1", "Label2","Label3", "Label4"],
       colors= ["aliceblue", "lightblue", "deepskyblue", "dodgerblue"]);
```



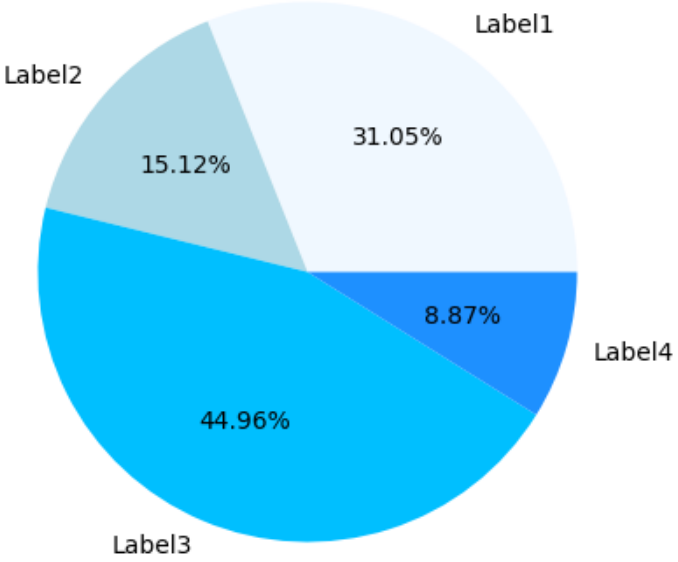
Set the wedge properties

```
fig, ax = plt.subplots(figsize = (15, 5))

ax.pie(x=[154, 75, 223, 44], labels = ["Label1", "Label2","Label3", "Label4"],
       colors= ["aliceblue", "lightblue", "deepskyblue", "dodgerblue"],
       wedgeprops = {"edgecolor" : "black",
                     'linewidth': 1,
                     'antialiased': True});
```



autopct

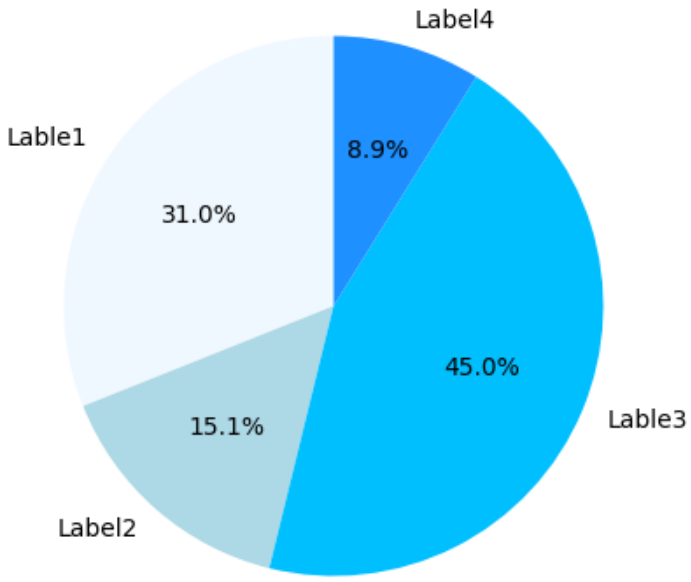
```
fig, ax = plt.subplots(figsize = (15, 5))

ax.pie(x=[154, 75, 223, 44], labels = ["Label1", "Label2","Label3", "Label4"],
       colors= ["aliceblue", "lightblue", "deepskyblue", "dodgerblue"],
       autopct='%.2f%%'); # format values as a float with 2 decimal places.
```



Set the start angle

Rotates so that Label1 is at 90 degrees

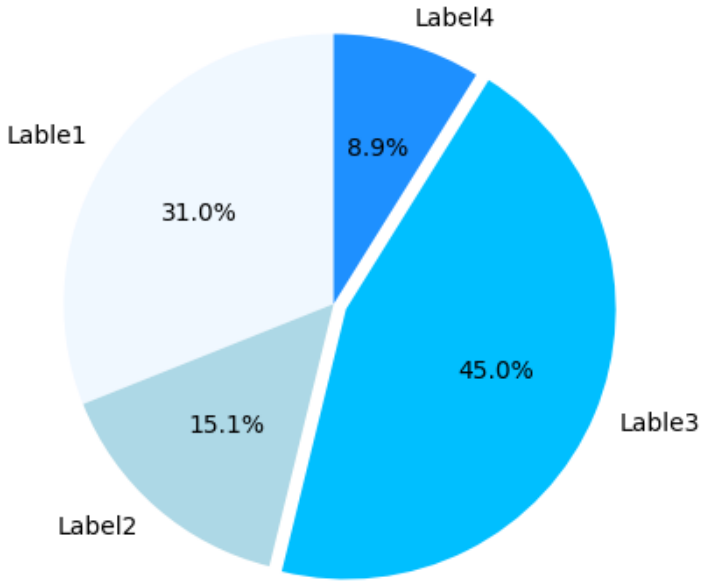
```
fig, ax = plt.subplots(figsize = (15, 5))

ax.pie(x=[154, 75, 223, 44], labels = ["Lable1", "Label2","Lable3", "Label4"],
       colors= ["aliceblue", "lightblue", "deepskyblue", "dodgerblue"],
       autopct='%.1f%%', startangle=90);
```



explode

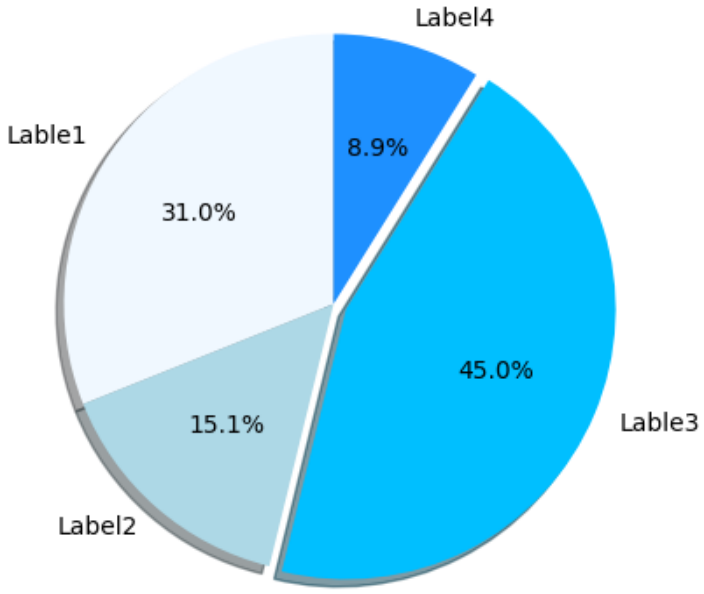
Separates out the indicated slice

```
fig, ax = plt.subplots(figsize = (15, 5))

ax.pie(x=[154, 75, 223, 44], labels = ["Lable1", "Label2","Lable3", "Label4"],
       colors=["aliceblue", "lightblue", "deepskyblue", "dodgerblue"],
       autopct='%.1f%%', startangle=90, explode = [0, 0, 0.05, 0]); # separate out Label 3 slic
```



shadow

```
fig, ax = plt.subplots(figsize = (15, 5))

ax.pie(x=[154, 75, 223, 44], labels = ["Lable1", "Label2","Lable3", "Label4"],
        colors= ["aliceblue", "lightblue", "deepskyblue", "dodgerblue"],
        autopct='%.1f%%', startangle=90, explode = [0, 0, 0.05, 0], shadow=True);
```



Dataset: Top 50 US Tech Companies

```
file = "Top 50 US Tech Companies.csv"
file_path = os.path.join(notebooks, file)
with open(file_path, "r") as f:
  contents = f.read()


#df = pd.read_csv("Top 50 US Tech Companies.csv")
df = pd.read_csv(file_path)

df.head()
```

	Company Name	Industry	Sector	HQ State	Founding Year	Annual Revenue 2022-2023 (USD in Billions)	Market Cap (USD in Trillions)	Stock Name	Ann Inc Tax 2022-2 (USD Billio
0	Apple Inc.	Technology	Consumer Electronics	California	1976	387.53	2.520	AAPL	18.
1	Microsoft Corporation	Technology	Software Infrastructure	Washington	1975	204.09	2.037	MSFT	15.
2	Alphabet (Google)	Technology	Software Infrastructure	California	1998	282.83	1.350	GOOG	11.
3	Amazon	Technology	Software Application	Washington	1994	513.98	1.030	AMZN	-3.
4	NVIDIA Corporation	Technology	Semiconductors	California	1993	26.97	0.653	NVDA	0.

Next steps: Generate code with df | View recommended plots | New interactive sheet

Unique categories

```
df["HQ State"].unique()
```

```
array(['California', 'Washington', 'Texas', 'New York', 'Connecticut',
       'Massachusetts', 'New Jersey', 'Wisconsin', 'Idaho', 'Montana',
       'Florida', 'Arizona', 'North Carolina'], dtype=object)
```

Value counts

```
df["HQ State"].value_counts()
```

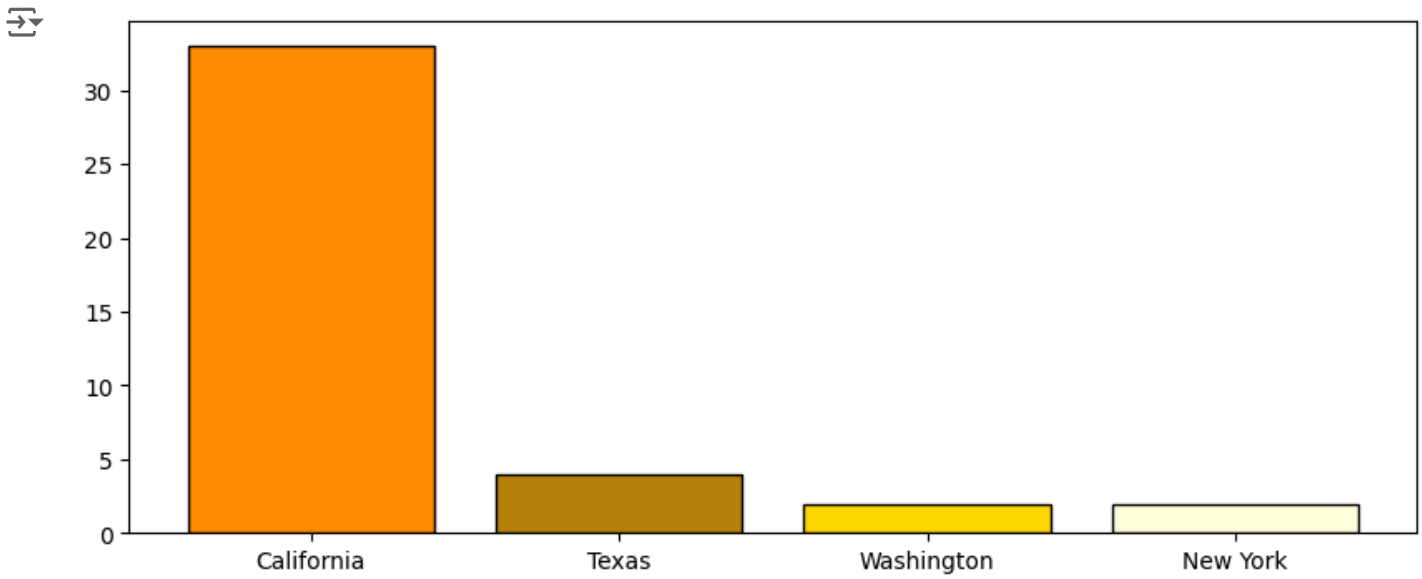
HQ State	count
California	33
Texas	4
Washington	2
New York	2
Connecticut	1
Massachusetts	1
New Jersey	1
Wisconsin	1
Idaho	1
Montana	1
Florida	1
Arizona	1
North Carolina	1

dtype: int64

Plot the top 4 states for tech company headquarters

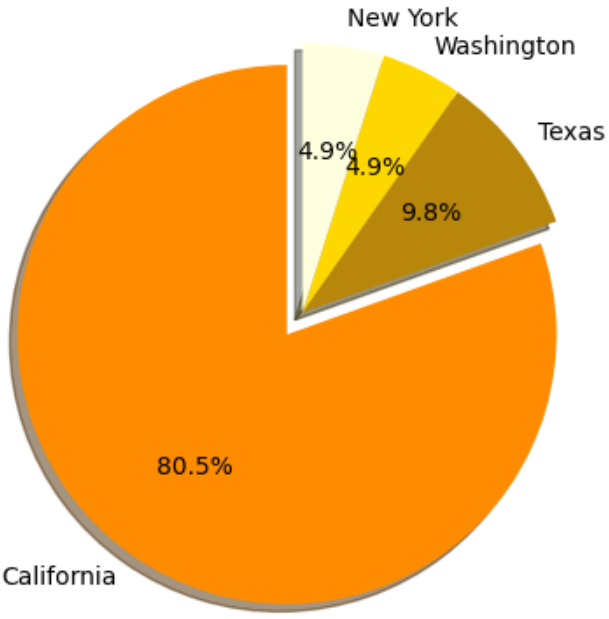
Bar plot

```
data = df["HQ State"].value_counts()[:4]
labels = ["California", "Texas", "Washington", "New York"]
colors = ["darkorange", "darkgoldenrod", "gold", "lightyellow"]


fig, ax = plt.subplots(figsize = (10, 4))

ax.bar(x=labels, height=data, color=colors, edgecolor="black");
```



Pie Chart

```
fig, ax = plt.subplots(figsize = (15, 5))

ax.pie(x=data, labels=labels, colors=colors,
       autopct='%.1f%%', startangle=90, explode = [0.1, 0, 0, 0], shadow=True);
```



Transforming a continuous variable into a categorical variable

Discretizing

Transforming from continuous to discrete variable

```
file = "churn.csv"
file_path = os.path.join(notebooks, file)
with open(file_path, "r") as f:
  contents = f.read()


#df = pd.read_csv("churn.csv")
df = pd.read_csv(file_path)

df.head()
```

	CreditScore	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	Estima
0	619	1	42	2	0.00	1	1	1	
1	608	1	41	1	83807.86	1	0	1	
2	502	1	42	8	159660.80	3	1	0	
3	699	1	39	1	0.00	2	0	0	
4	850	1	43	2	125510.82	1	1	1	

Next steps: Generate code with df | View recommended plots | New interactive sheet

Binning

```
df["Credit Category"] = pd.cut(df["CreditScore"], [0, 579, 669, 739, 799, 850],
                                        labels=["Poor","Fair","Good","Very Good", "Excellent"])
df["Credit Category"].head(10)
```

Credit Category

Count each category

```
df["Credit Category"].value_counts()
```

Credit Category	count
Fair	3331
Good	2428