Continuous 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()
```

```
Mounted at /content/drive
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
```

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

Dataset: Heart Disease

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

df.head()
```

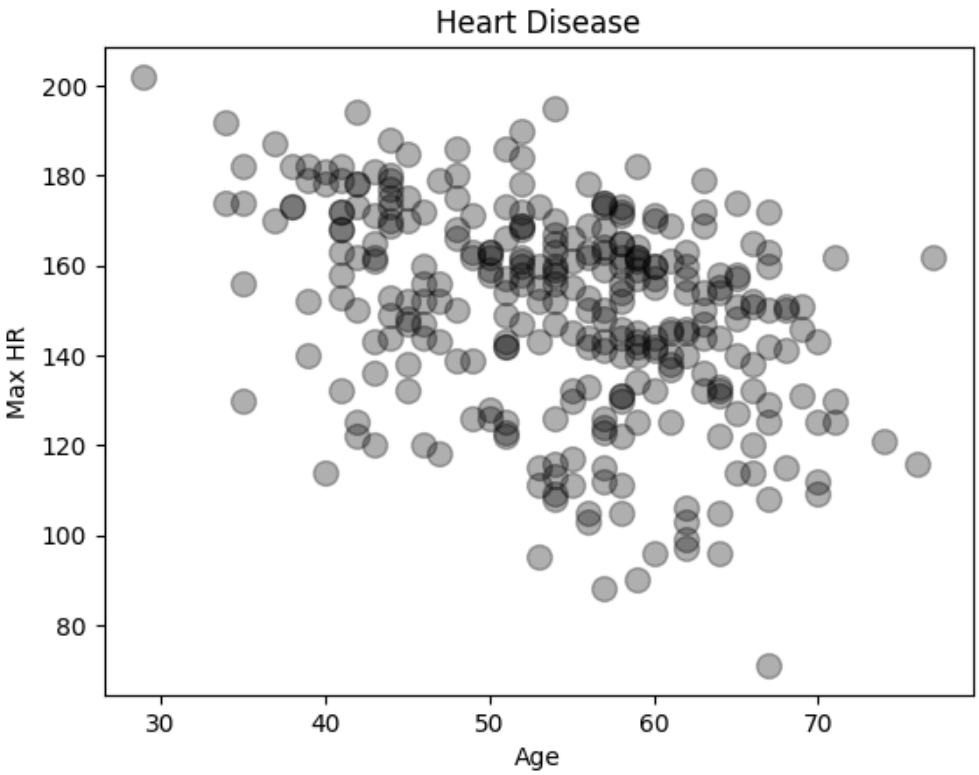
	age	sex	chest_pain	rest_bp	chol	max_hr	st_depr	heart_disease
0	63	female	3	145	233	150	2.3	1
1	37	female	2	130	250	187	3.5	1
2	41	male	1	130	204	172	1.4	1
3	56	female	1	120	236	178	0.8	1
4	57	male	0	120	354	163	0.6	1

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

Scatterplot

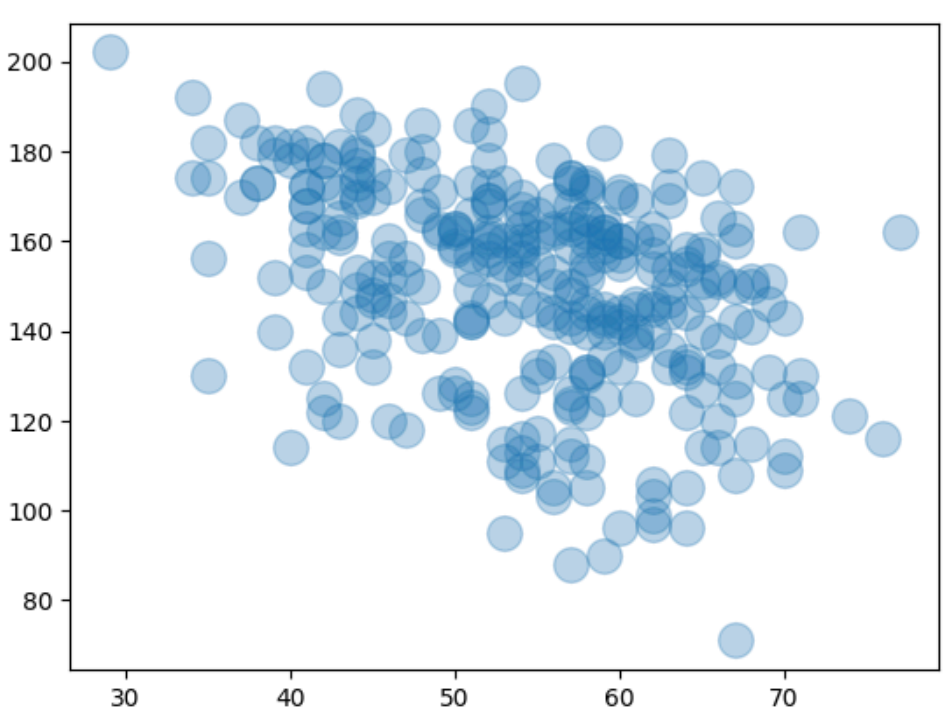
Joint: continuous x continuous

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

#              (x-axis,       y-axis,      alpha,    size,     color)
ax.scatter(x=df['age'], y=df['max_hr'], alpha=.3, s =100, c = "black")

# Set the properties
ax.set(title="Heart Disease", xlabel='Age', ylabel="Max HR");
```



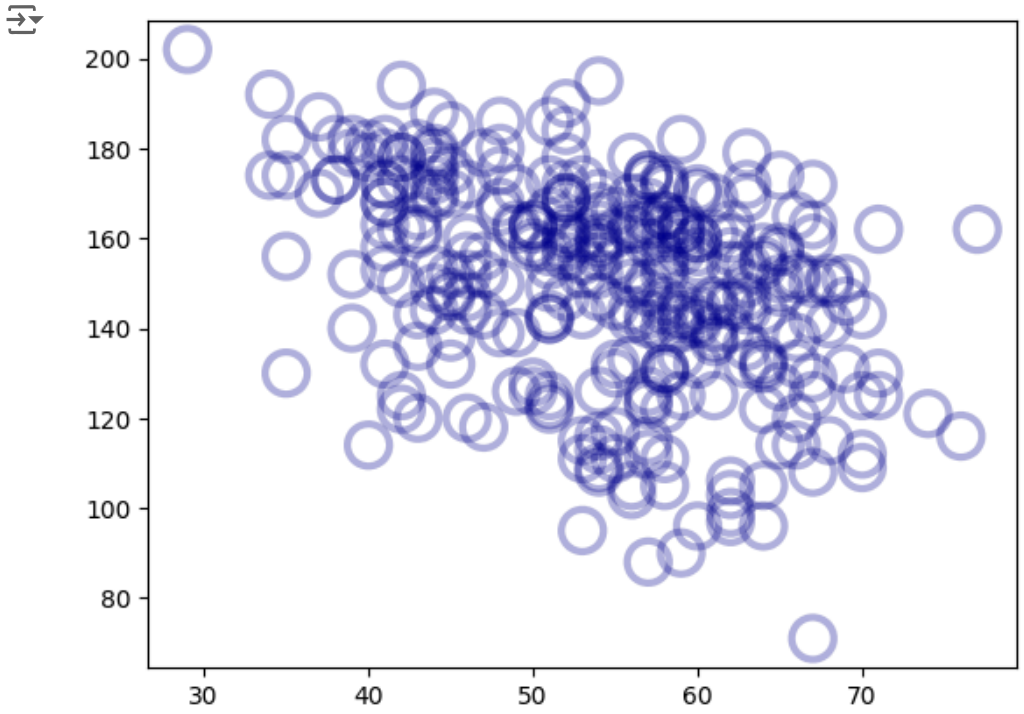
Modify the marker

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

# sample markers: "o"(default), "v", "^", x", "P", "d", "."
ax.scatter(df['age'], df['max_hr'], alpha=.3, s = 200, marker="o");
```



Modify the marker style

```
fig, ax = plt.subplots()
                                                    # no fill color
ax.scatter(df['age'], df['max_hr'], alpha=.3, s = 300, c = "none", edgecolors="darkblue", linewidt
```



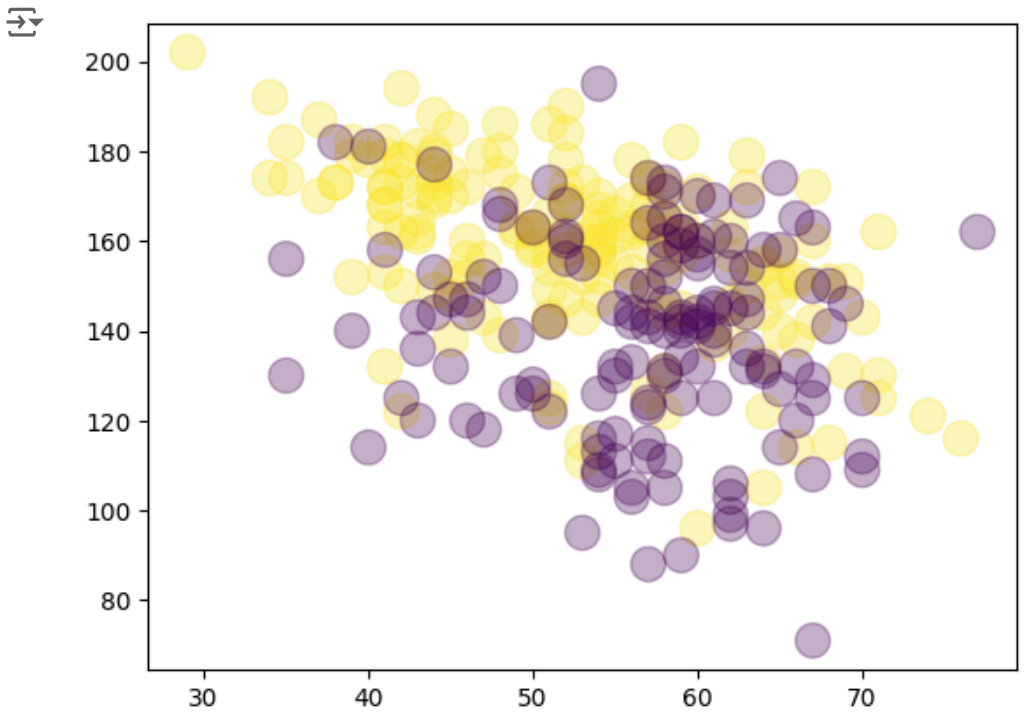
Joint: continuous x categorical

Age x MaxHR x Heart_Disease

Auto-set the colors based on class membership (categorical variable)

The target variable must be an integer.

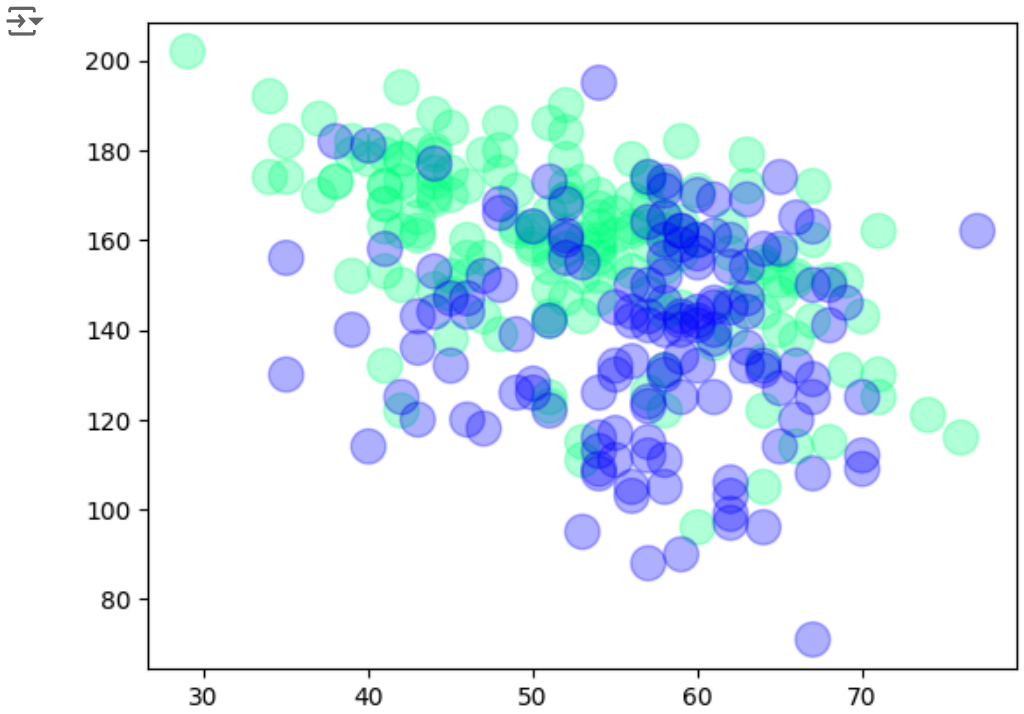
```
fig, ax = plt.subplots()

ax.scatter(df['age'], df['max_hr'], alpha=.3, s = 200, c = df["heart_disease"]);
```



Modify the colormap

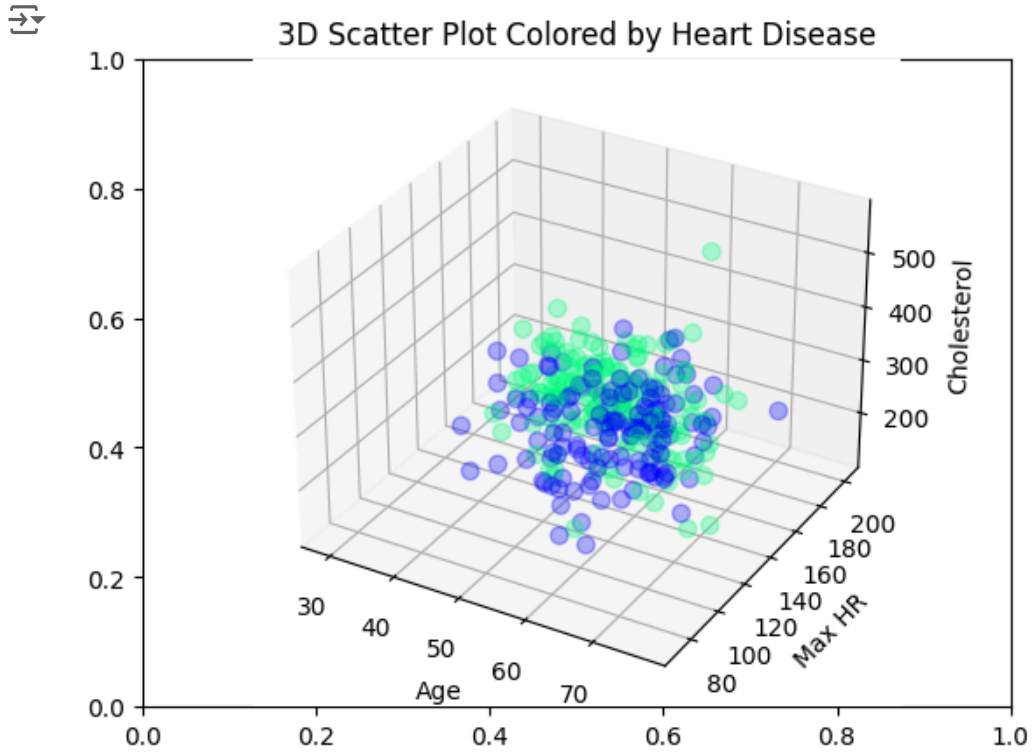
[Colormaps]

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

ax.scatter(df['age'], df['max_hr'], alpha=.3, s = 200, c = df["heart_disease"], cmap = "winter");
```



3D Scatterplot

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

# add 3D
ax = fig.add_subplot(111, projection='3d')

ax.scatter(df['age'], df['max_hr'], df['chol'], alpha=.3, s = 50, c = df["heart_disease"], cmap =

ax.set_title('3D Scatter Plot Colored by Heart Disease')
ax.set_xlabel('Age')
ax.set_ylabel('Max HR')
ax.set_zlabel('Cholesterol');
```

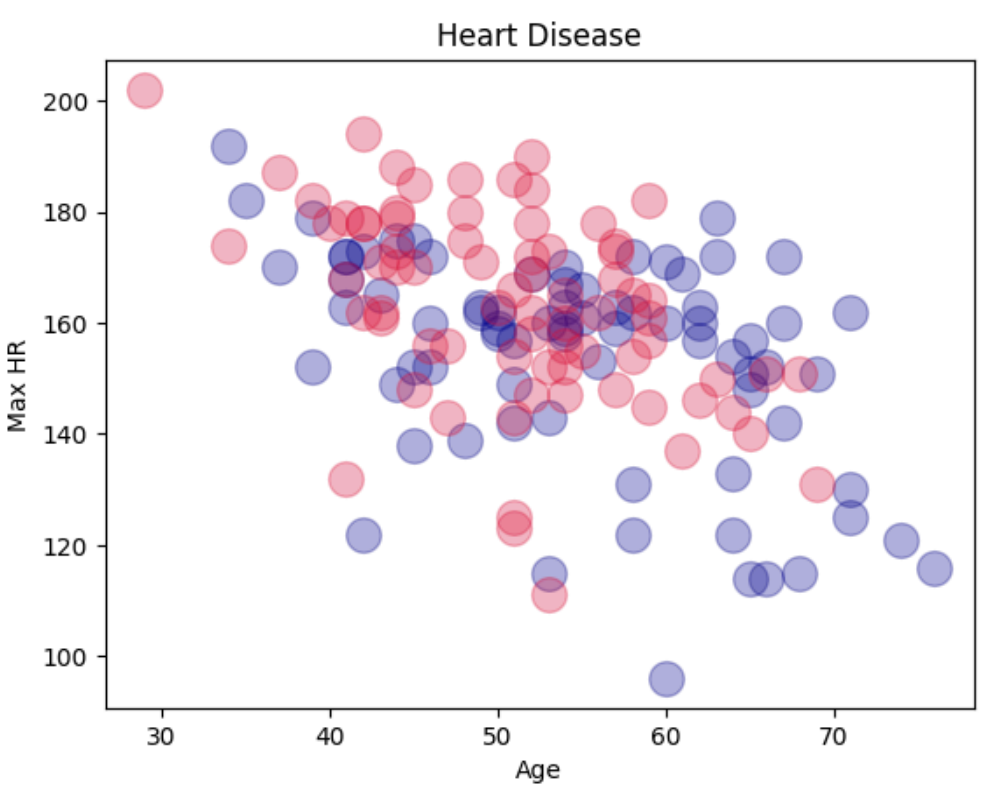


Manually set the colors based on class membership (categorical variable)

Age x MaxHR x Sex

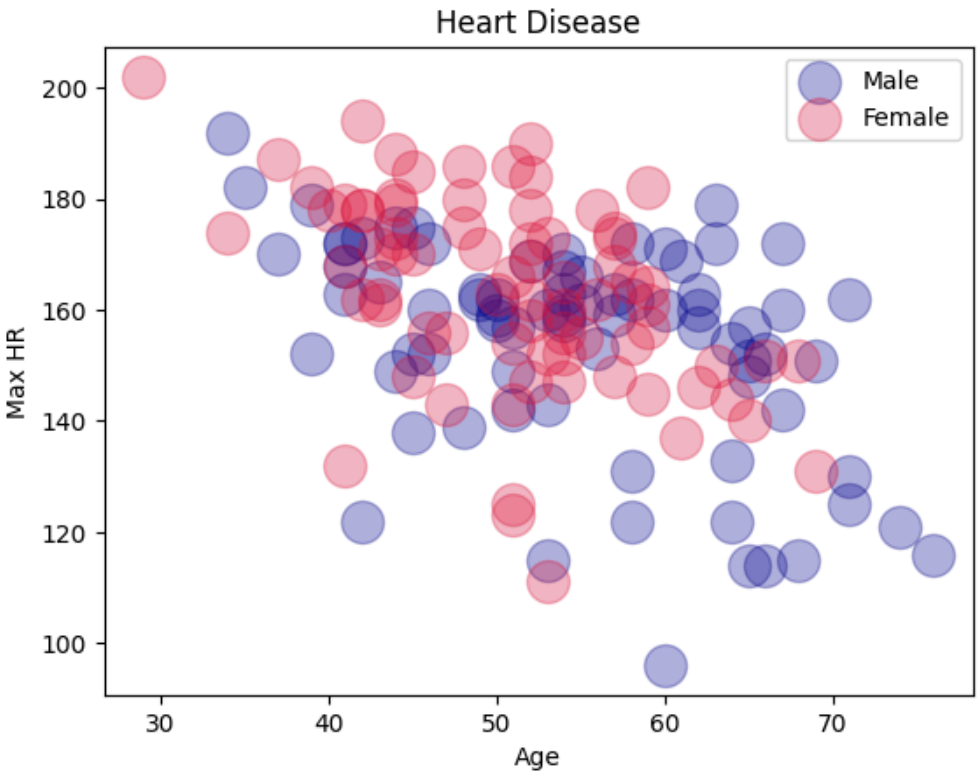
```
males_age = df.loc[df["sex"] == "male", 'age'][:75]
males_maxhr = df.loc[df["sex"] == "male", 'max_hr'][:75]
females_age = df.loc[df["sex"] == "female", 'age'][:75]
females_maxhr = df.loc[df["sex"] == "female", 'max_hr'][:75]


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

```
ax.scatter(males_age, males_maxhr, alpha=.3, s = 200, c = "darkblue")
ax.scatter(females_age, females_maxhr, alpha=.3, s = 200, c = "crimson")

ax.set(title="Heart Disease", xlabel='Age', ylabel="Max HR");
```



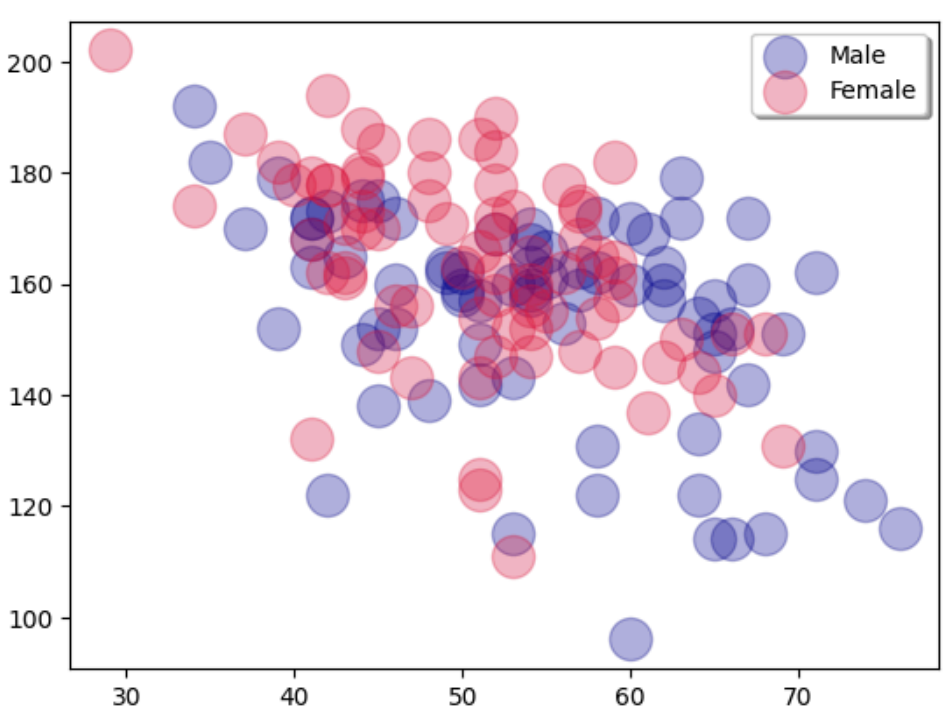
Add a legend

```
fig, ax = plt.subplots()
                                                                    # Add labels to be used by tl
ax.scatter(males_age, males_maxhr, alpha=.3, s = 300, c = "darkblue", label="Male")
ax.scatter(females_age, females_maxhr, alpha=.3, s = 300, c = "crimson", label="Female")

ax.set(title="Heart Disease", xlabel='Age', ylabel="Max HR");

# The legend will use the label within scatter()
ax.legend();
```



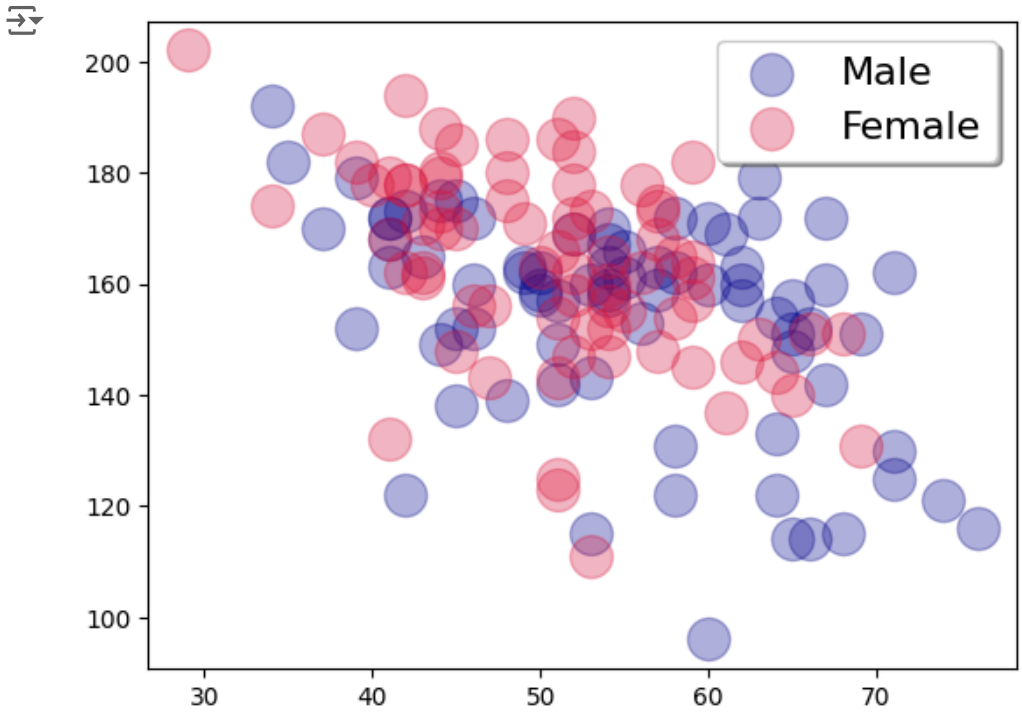
Add shadow to legend

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

ax.scatter(males_age, males_maxhr, alpha=.3, s = 300, c = "darkblue", label="Male")
ax.scatter(females_age, females_maxhr, alpha=.3, s = 300, c = "crimson", label="Female")

ax.legend(shadow=True);
```



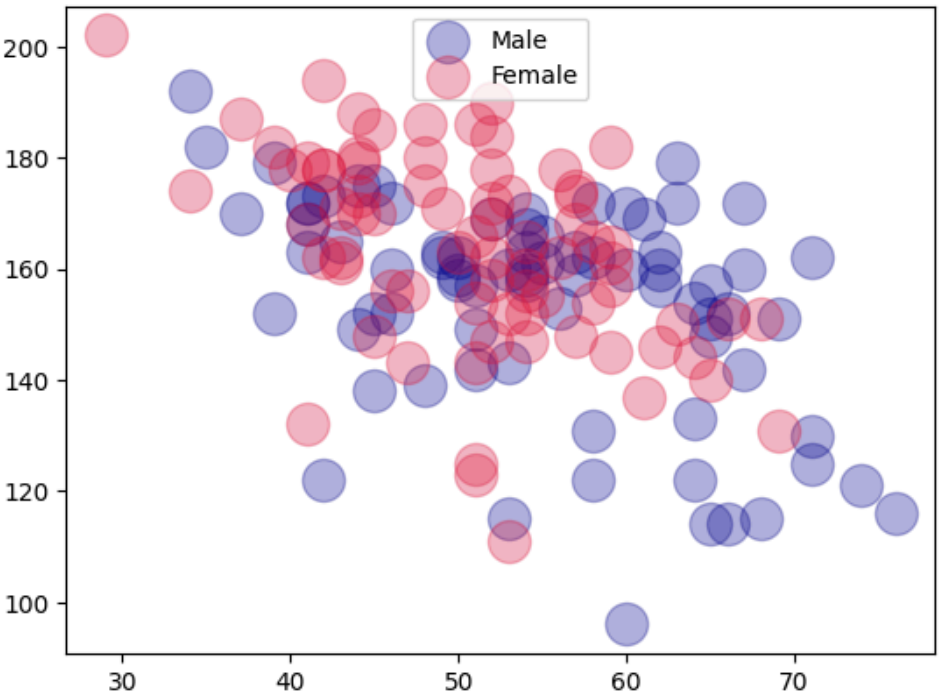
Increase legend font size

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

ax.scatter(males_age, males_maxhr, alpha=.3, s = 300, c = "darkblue", label="Male")
ax.scatter(females_age, females_maxhr, alpha=.3, s = 300, c = "crimson", label="Female")

ax.legend(shadow=True, fontsize=16);
```



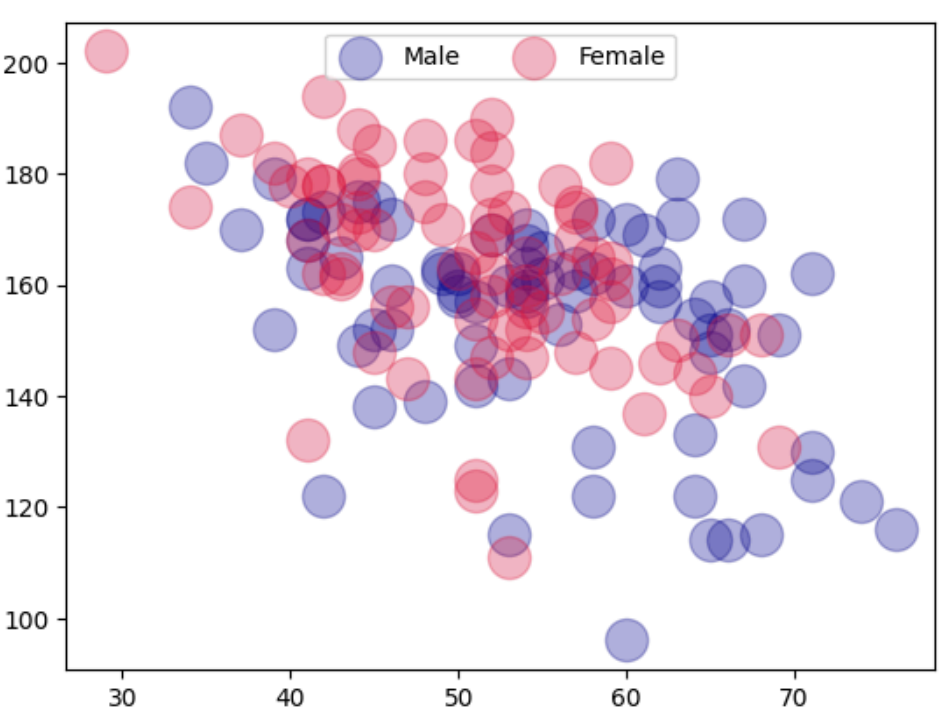
Modify the location of the legend

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

ax.scatter(males_age, males_maxhr, alpha=.3, s = 300, c = "darkblue", label="Male")
ax.scatter(females_age, females_maxhr, alpha=.3, s = 300, c = "crimson", label="Female")


# OPTIONS:
# 'upper left', 'upper right', 'lower left', 'lower right',
# 'upper center', 'lower center', 'center left', 'center right'

# The default is "best"
ax.legend(loc="upper center");
```



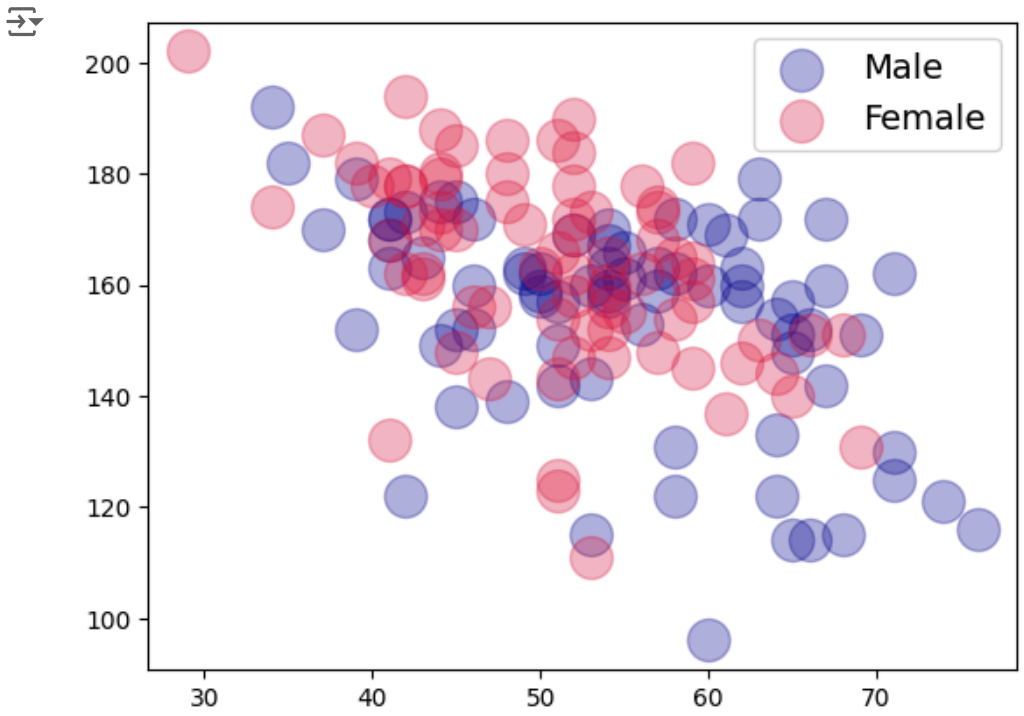
Modify the number of columns for the legend to display

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

ax.scatter(males_age, males_maxhr, alpha=.3, s = 300, c = "darkblue", label="Male")
ax.scatter(females_age, females_maxhr, alpha=.3, s = 300, c = "crimson", label="Female")

                        # Display legend as 2 columns
ax.legend(loc="upper center", ncol=2);
```



Customize legend

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

males = ax.scatter(males_age, males_maxhr, alpha=.3, s = 300, c = "darkblue", label="Male")
females = ax.scatter(females_age, females_maxhr, alpha=.3, s = 300, c = "crimson", label="Female'

# Customize a legend
ax.legend(loc='upper right', fontsize=14);
```

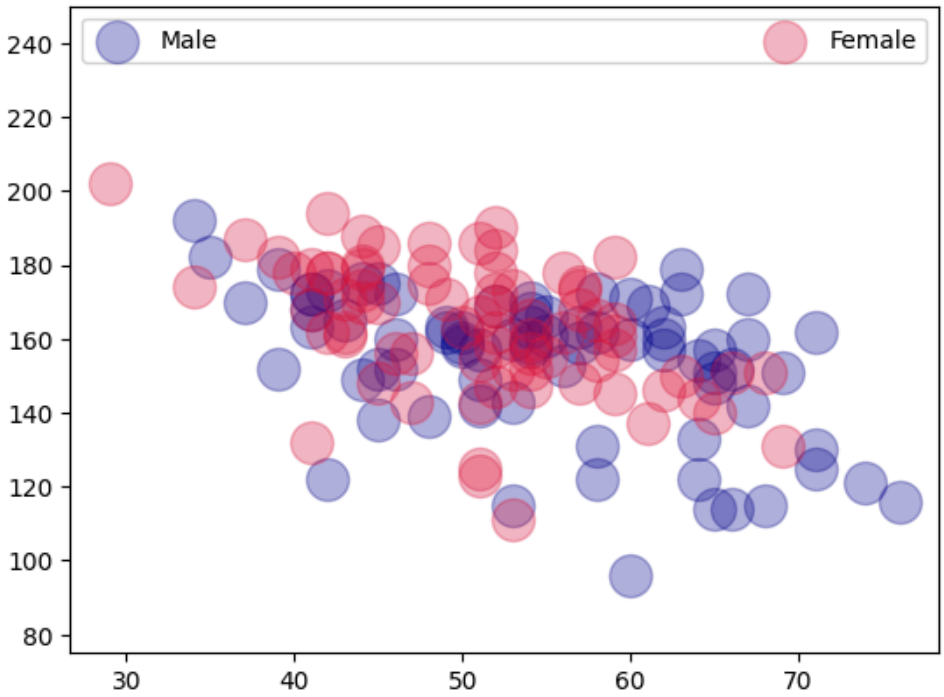


Display horizontally (as two columns)

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


males = ax.scatter(males_age, males_maxhr, alpha=.3, s = 300, c = "darkblue", label="Male")
females = ax.scatter(females_age, females_maxhr, alpha=.3, s = 300, c = "crimson", label="Female'

ax.legend(loc='upper right', ncol=2, fontsize=14);
```

Display horizontally (expanded)

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


males = ax.scatter(males_age, males_maxhr, alpha=.3, s = 300, c = "darkblue", label="Male")
females = ax.scatter(females_age, females_maxhr, alpha=.3, s = 300, c = "crimson", label="Female'

# Adjust the y-axis to make room for the leged to be centered
ax.set_ylim(75, 250)

ax.legend(loc='upper center', ncol=2, mode="expand", fontsize=10);
```



Anchor legend above the scatterplot and hide frame

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


males = ax.scatter(males_age, males_maxhr, alpha=.3, s = 300, c = "darkblue", label="Males")
females = ax.scatter(females_age, females_maxhr, alpha=.3, s = 300, c = "crimson", label="Females"

                        # x, y  x, y
ax.legend(bbox_to_anchor=(0, 1, 1, 0), # <- coordinates for placing legend
                    ncol=2, mode="expand", fontsize=14, frameon=False);
```

Add a title to the legend

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

males = ax.scatter(males_age, males_maxhr, alpha=.3, s = 300, c = "darkblue", label="Males")
females = ax.scatter(females_age, females_maxhr, alpha=.3, s = 300, c = "crimson", label="Females"

                     #  x, y  x, y
ax.legend(bbox_to_anchor=(0, 1, 1, 0),
                   ncol=2, mode="expand", fontsize=14, frameon=False, title="Heart Disease", titl
```