Continuous Variables - Histogram

```
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
Artificial Intelligence
cloud
--------------
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',
---------------
```

Dataset: Heart Disease

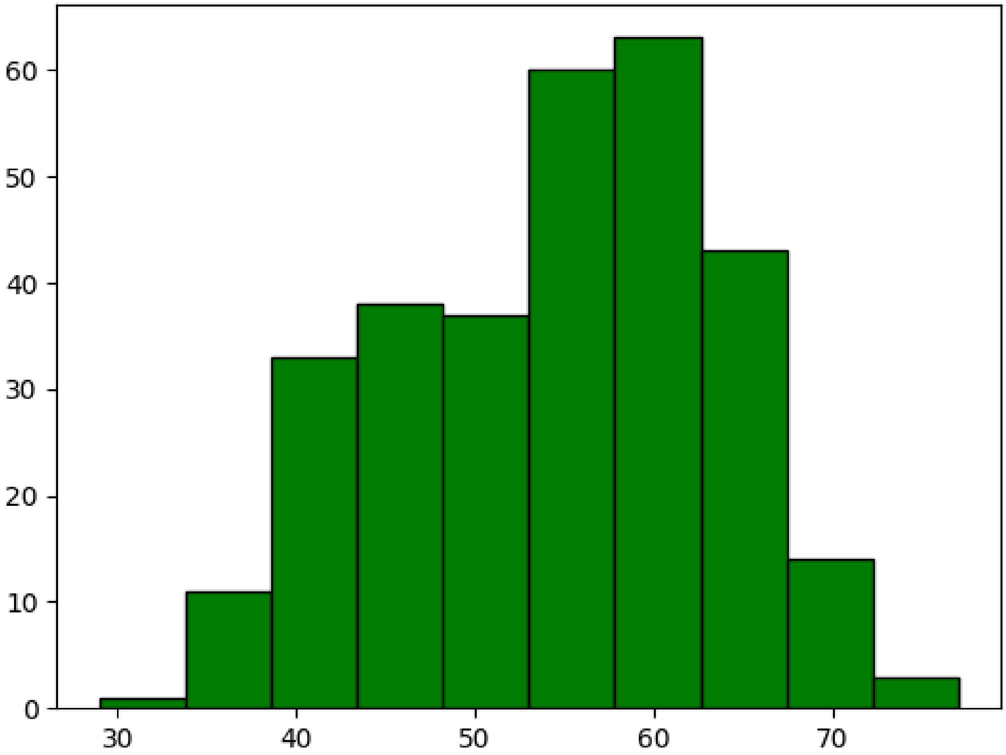
```
#df = pd.read_csv("heart-disease.csv")
df = pd.read_csv(file_path)

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

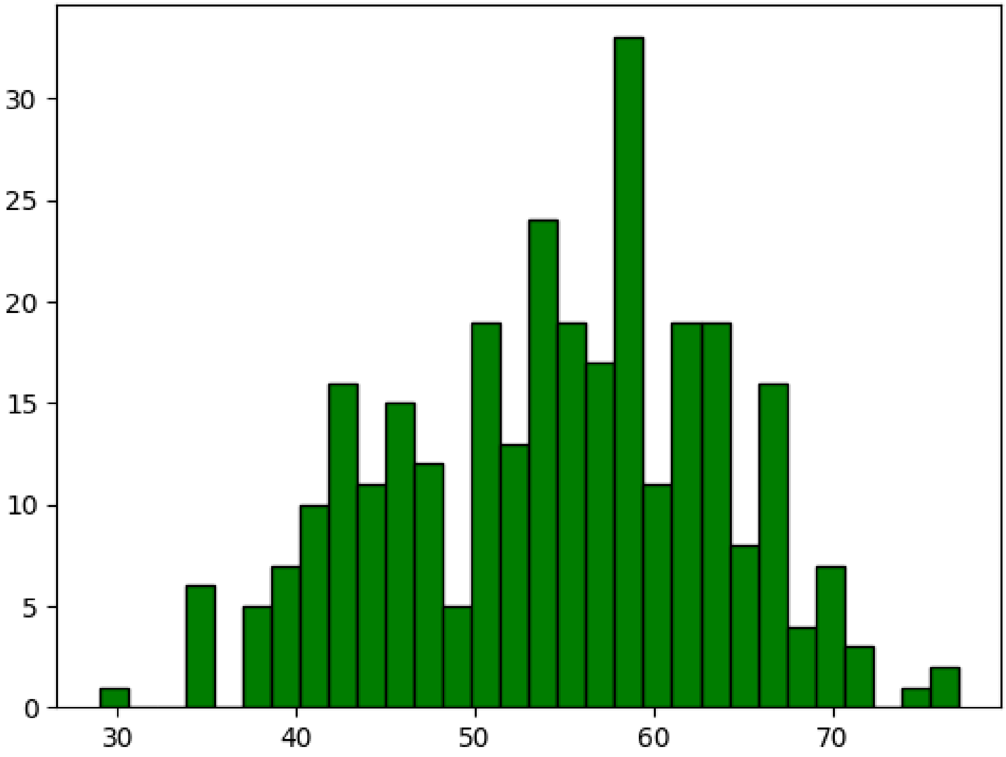
Histogram

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

ax.hist(df['age'], color='g', edgecolor='black'); # the default number of bins is 10

# The basic built-in colors:
# b: blue
# g: green
# r: red
# c: cyan
# m: magenta
# y: yellow
# k: black
# w: white
```



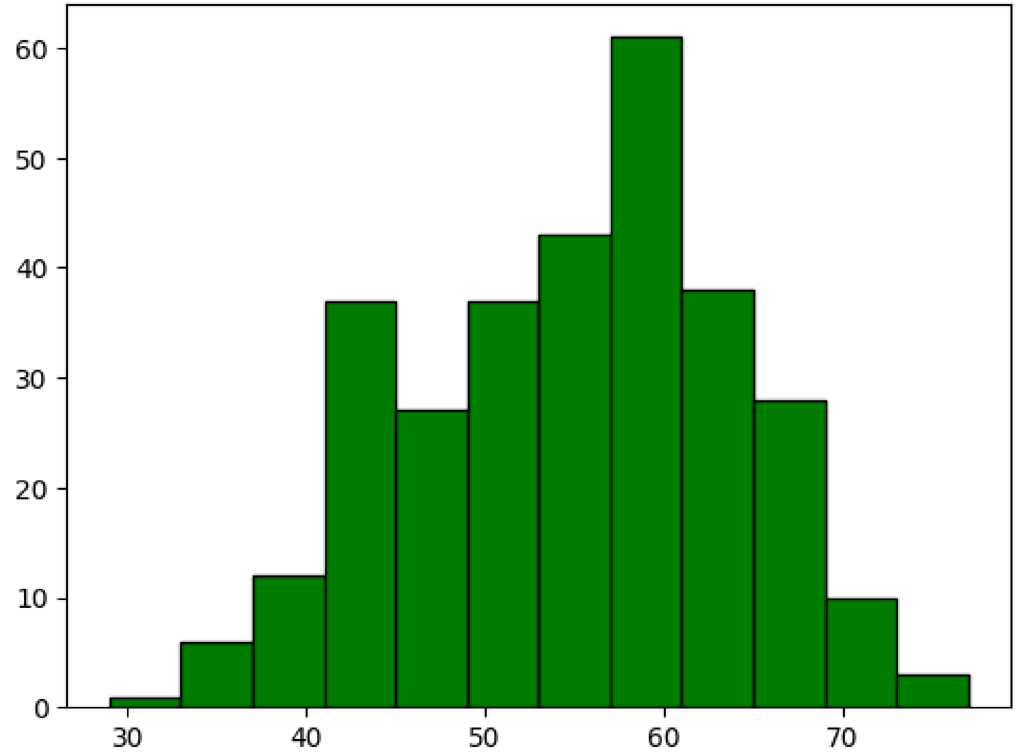
Set the number of bins to display

```
fig, ax = plt.subplots()
                                                # set the number of bins
ax.hist(df['age'], color='g', edgecolor='black', bins = 30);
```



Auto-set the number of bins to display

```
fig, ax = plt.subplots()
                                                # auto-set the number of bins
                                                # "ideal" number of bins that most faith
ax.hist(df['age'], color='g', edgecolor='black', bins = "auto");
```

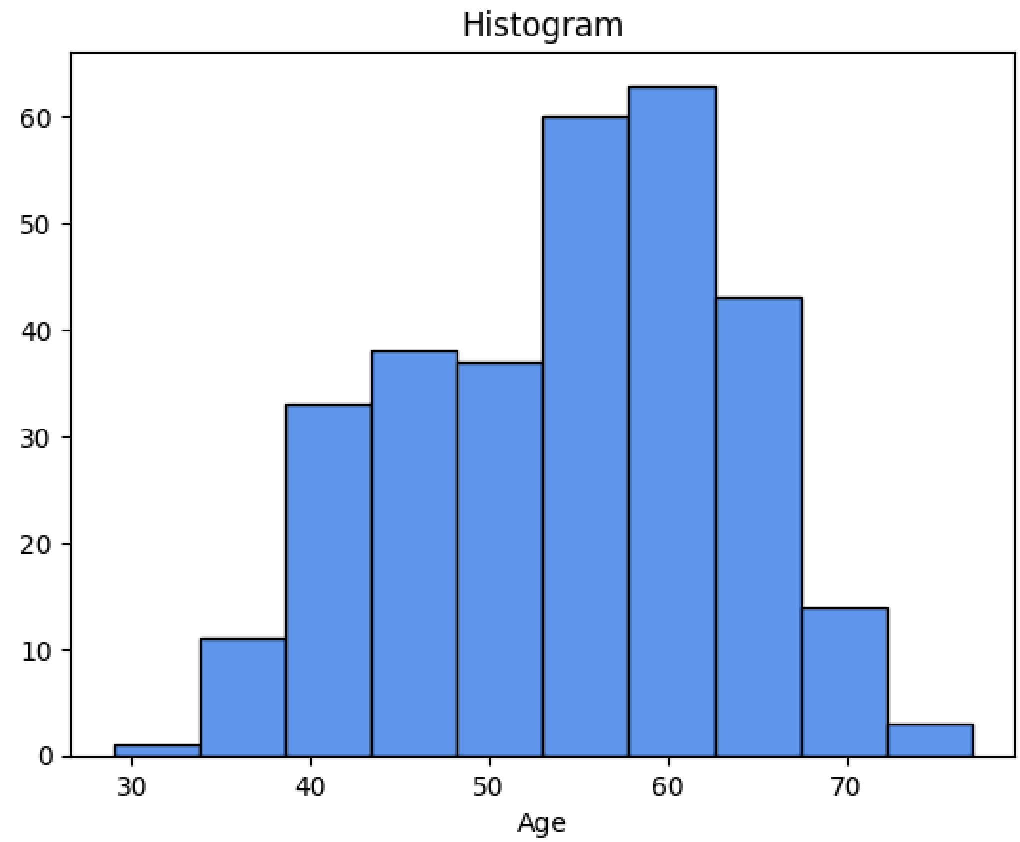


Set a custom color

[Named Colors]

[HTML Color Codes]

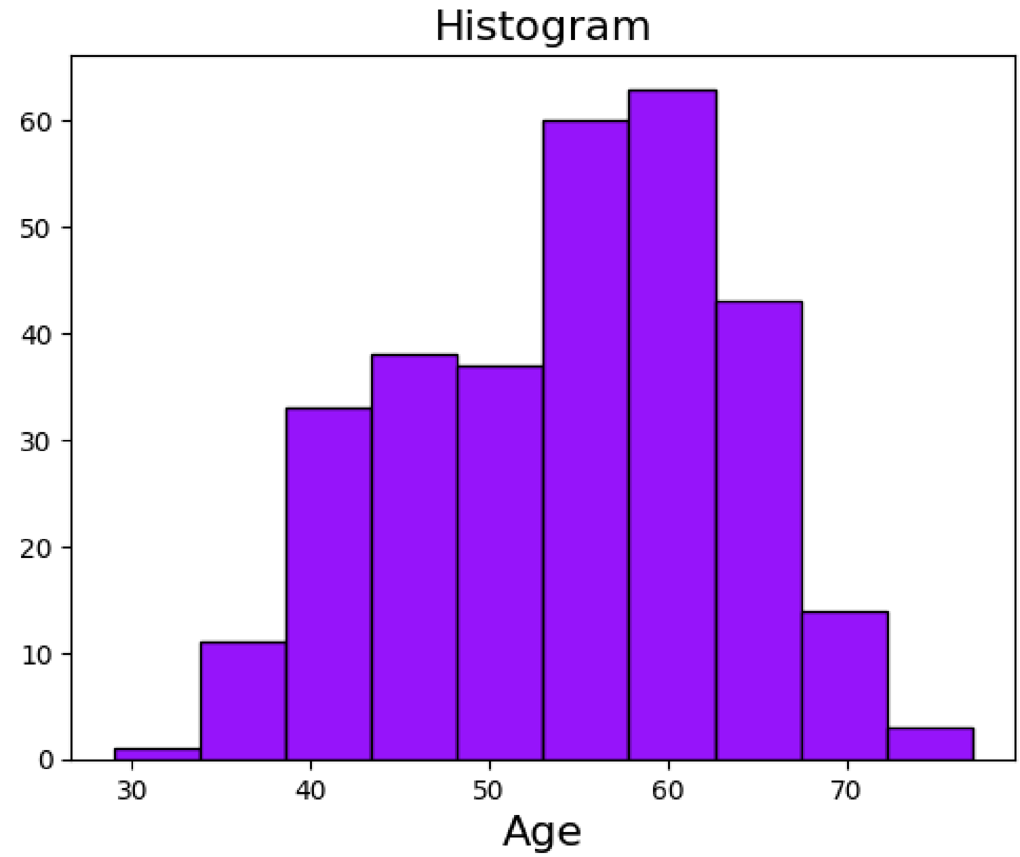
```
fig, ax = plt.subplots()

                        # custom colors
ax.hist(df['age'], color = '#6399EB', edgecolor='black')

ax.set_title("Histogram")
ax.set_xlabel("Age");
```



Style the chart

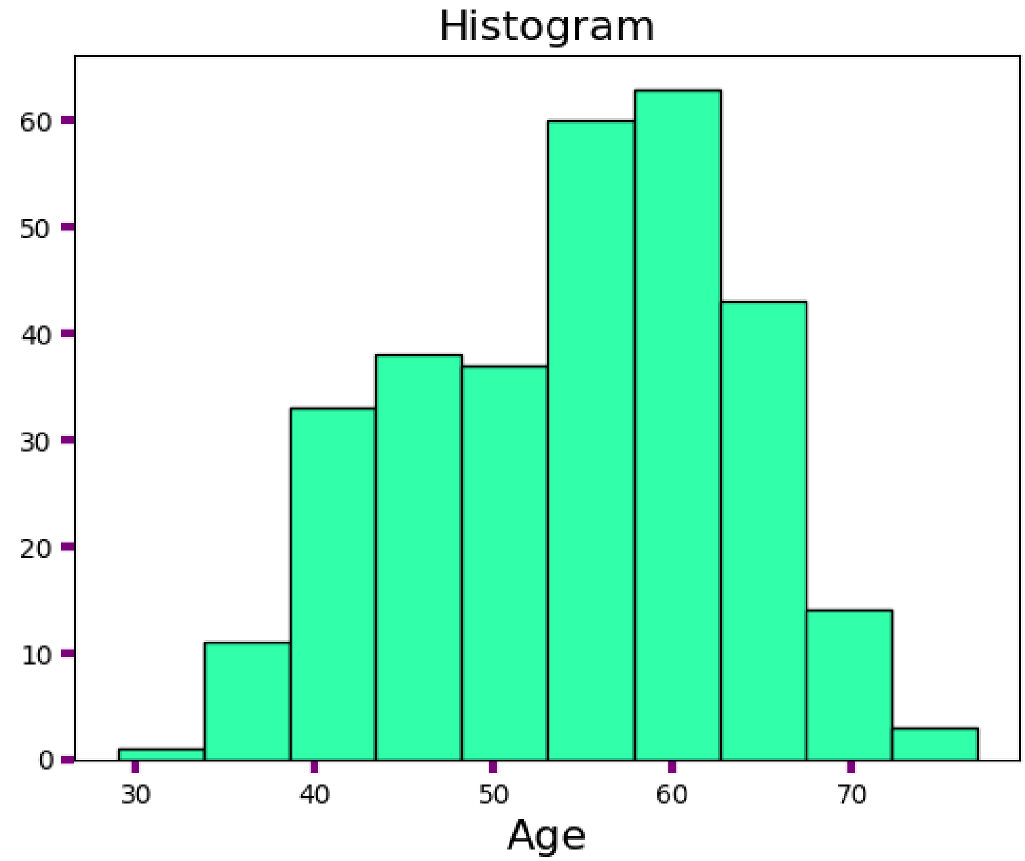
Format the labels (increase font size)

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

ax.hist(df['age'], color = '#9616FA', edgecolor='black')

# Increase the font size of the labels
ax.set_title('Histogram', fontsize=16)
ax.set_xlabel('Age', fontsize=16);
```



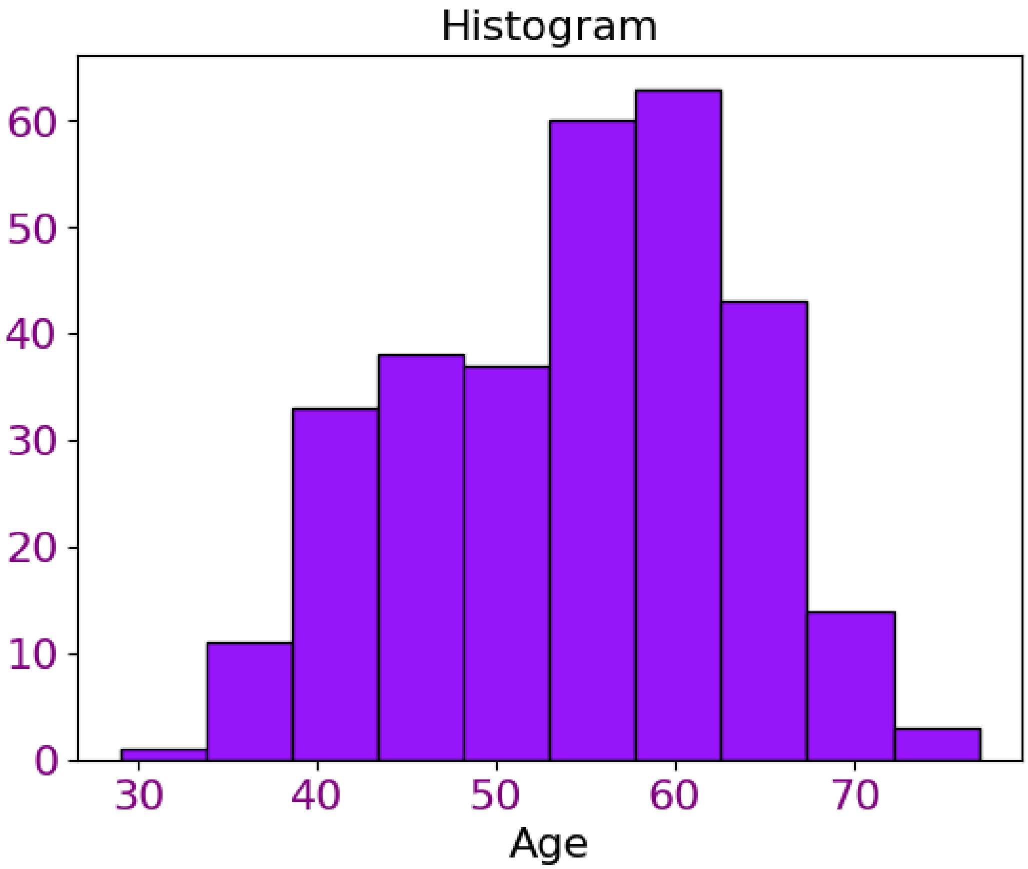
Customize the ticks

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

ax.hist(df['age'], color = '#33ffac', edgecolor='black')

ax.set_title('Histogram', fontsize=16)
ax.set_xlabel('Age', fontsize=16)

# Customize the ticks
ax.tick_params(color="purple", width=3, length=5);
```



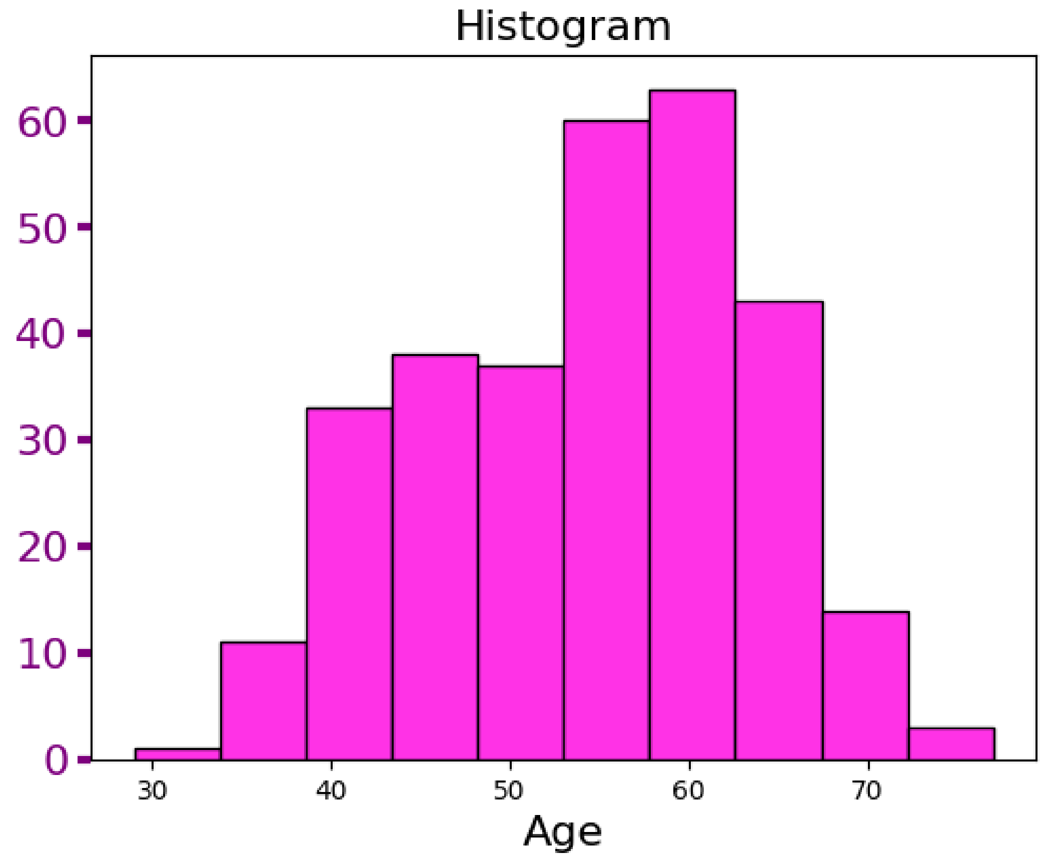
Customize the tick labels (color and size)

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

ax.hist(df['age'], color = '#9616FA', edgecolor='black')

ax.set_title('Histogram', fontsize=16)
ax.set_xlabel('Age', fontsize=16)

# Customize the tick labels
ax.tick_params(labelsize=16, labelcolor="purple");
```



Customize only a specific axis

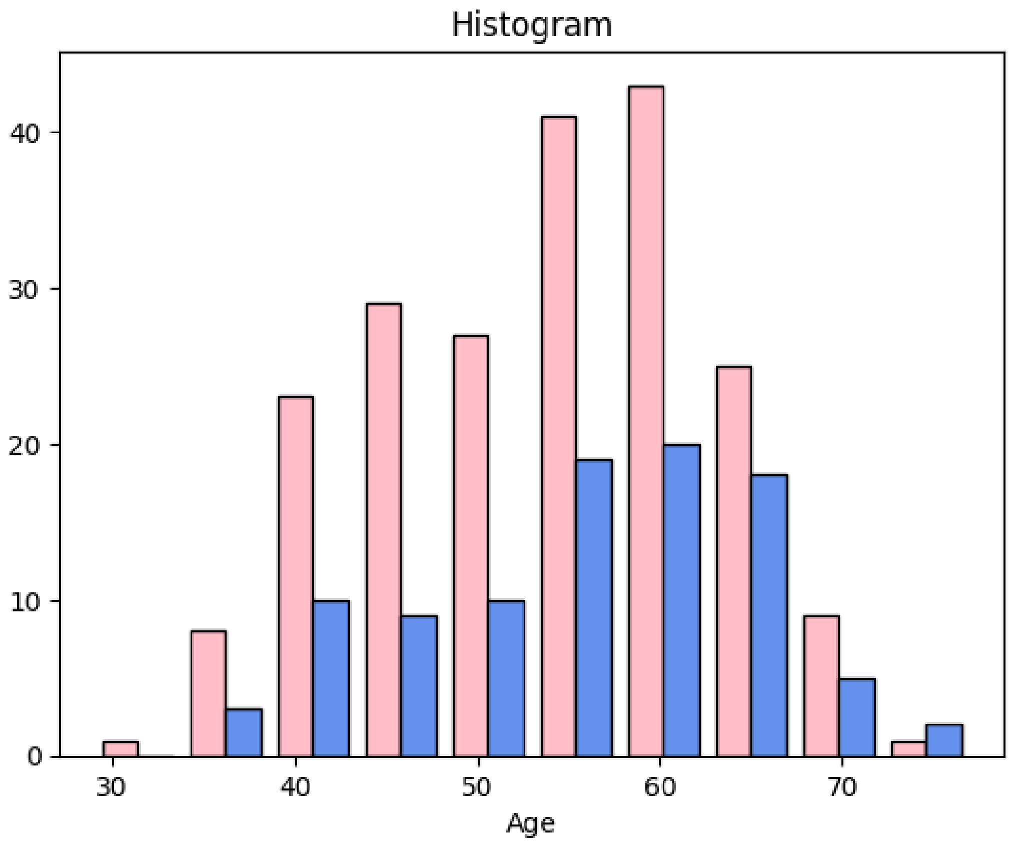
```
fig, ax = plt.subplots()

ax.hist(df['age'], color = '#ff33ea', edgecolor='black')

ax.set_title('Histogram', fontsize=16)
ax.set_xlabel('Age', fontsize=16)

# Customize the ticks of a specific axis
ax.tick_params(axis='y', labelsize=16, labelcolor="purple", color="purple", width=3, len
```



Continuous x categorical (age x gender)

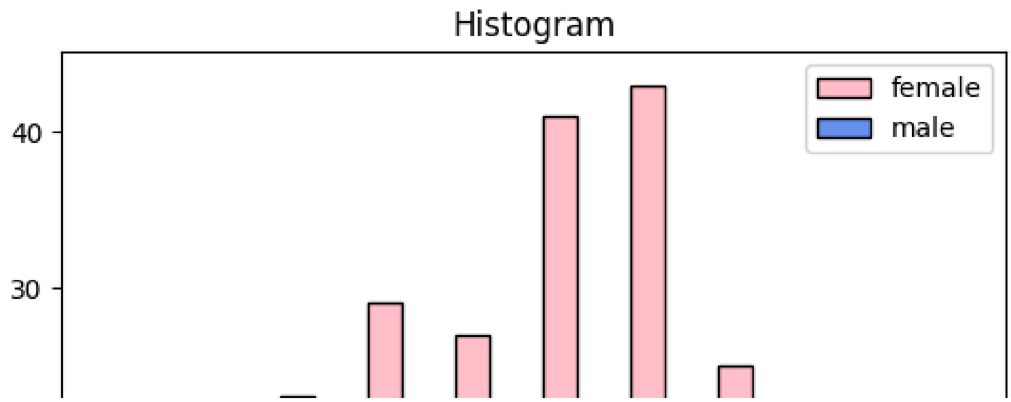
Create paired histograms

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

# The data
female_ages = df.loc[df['sex']=="female", "age"]
male_ages = df.loc[df['sex']=="male", "age"]

        # List of ages by gender (for paired histogram)
ax.hist([female_ages, male_ages], color=['pink', "cornflowerblue"], edgecolor='black')

ax.set(xlabel="Age", title="Histogram");
```



Add a legend

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

# The data
females_age = df.loc[df['sex']=="female", "age"]
males_age = df.loc[df['sex']=="male", "age"]

ax.hist([females_age, males_age], color=['pink',"cornflowerblue"], edgecolor='black')

ax.set(xlabel="Age", title="Histogram")

plt.legend(["female", "male"]);  # <-- display a legend
```


Histogram
40
30
female
male