Functions

```
import pandas as pd
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()

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

df["sex"] = df["sex"].map({"male":0, "female":1})
df.head()
```

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

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

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

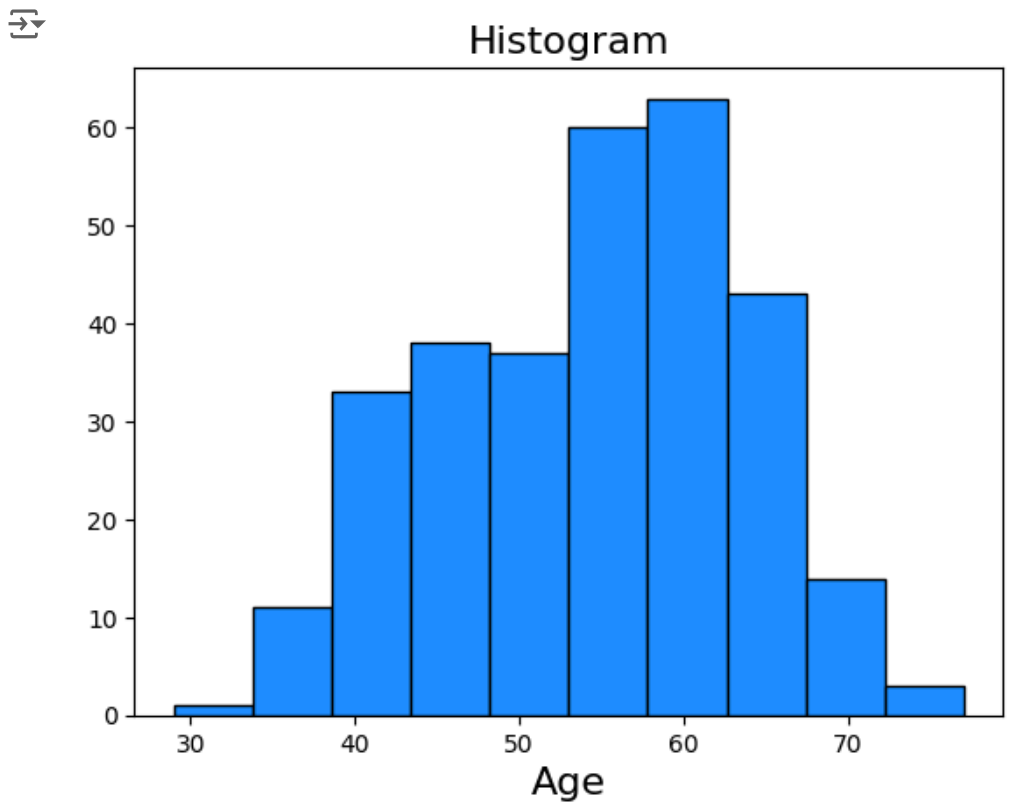
```
age = df["age"]
```

Histogram

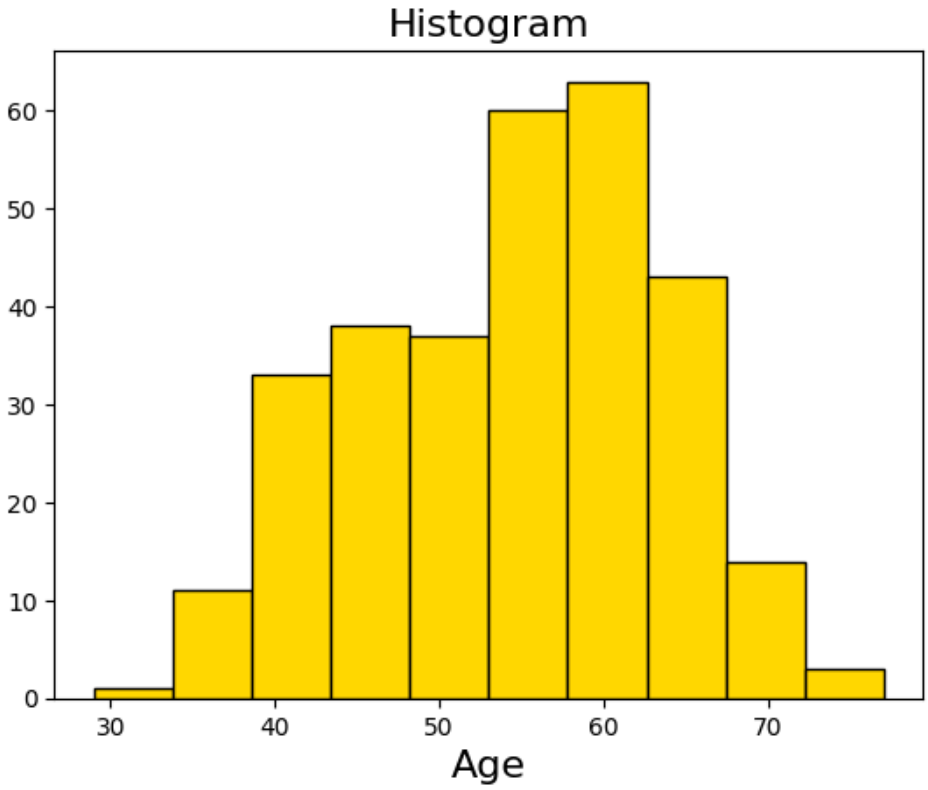
```
def hist(data, color="dodgerblue", label="Age"):
    import matplotlib.pyplot as plt

    fig, ax = plt.subplots()

    ax.hist(data, color=color, edgecolor="black")

    ax.set_title('Histogram', fontsize=16)
    ax.set_xlabel(label, fontsize=16);
```

Call the hist() function

```
# use default color
hist(age)
```




```
# use custom color
hist(age, color="gold")
```



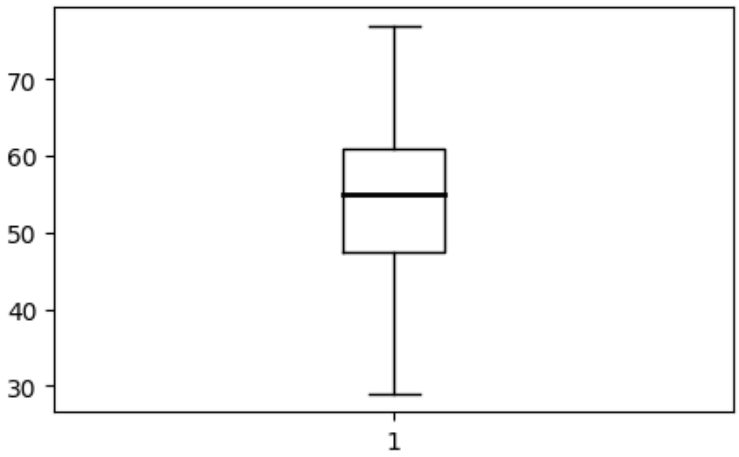
Boxplot

```
def boxplot(data, labels=[1], median_color="black", patch_artist=False, box_color=["dodgerblue"]
    import matplotlib.pyplot as plt

    fig, ax = plt.subplots(figsize = (5, 3))

    bplot = ax.boxplot(data, patch_artist=patch_artist,
#                            medianprops={"color":median_color, "linewidth":2}, tick_labels=labe
                           medianprops={"color":median_color, "linewidth":2})

    if patch_artist:
        for patch, color in zip(bplot['boxes'], box_color):
            patch.set_facecolor(color);
```

Call the boxplot() function

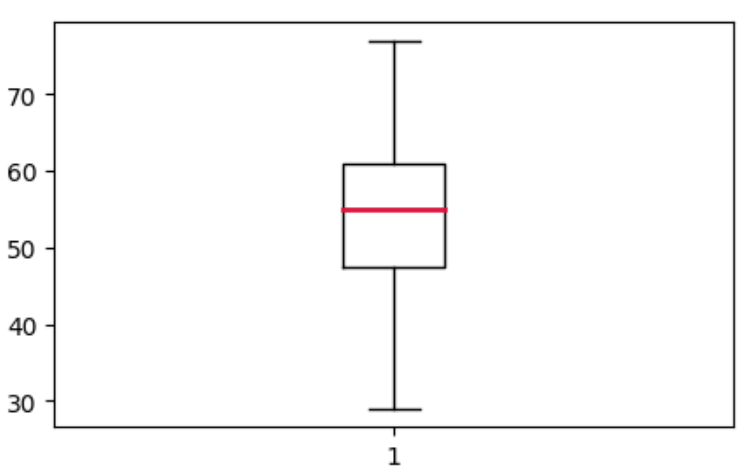
```
# use defaults
boxplot(age)
ax.set_xticklabels(labels)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-17-067ba165a817> in <cell line: 3>()
      1 # use defaults
      2 boxplot(age)
----> 3 ax.set_xticklabels(labels)

NameError: name 'ax' is not defined
```

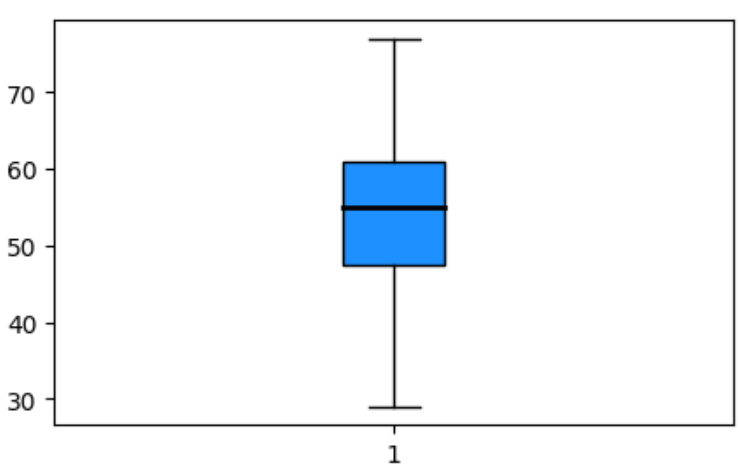


Next steps: Explain error

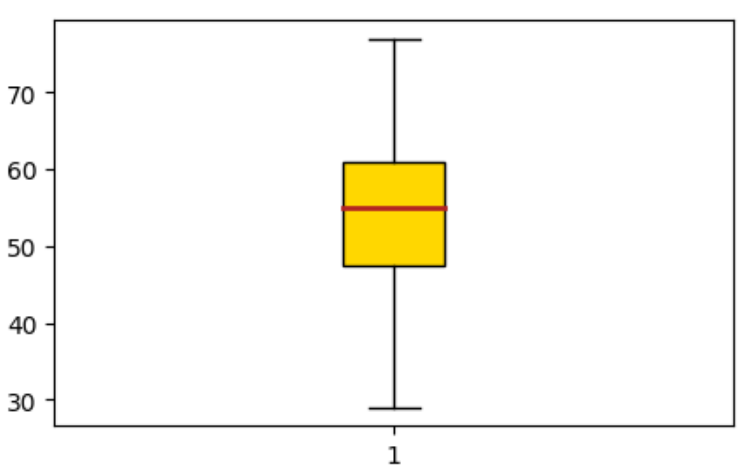
```
# set the label and the median color
boxplot(age, labels=["age"], median_color="crimson")
```




```
# use default box color
boxplot(age, labels=["age"], patch_artist=True)
```




```
# use custom median and box colors
boxplot(age, labels=["age"], median_color="firebrick", patch_artist=True, box_color=["gold"])
```

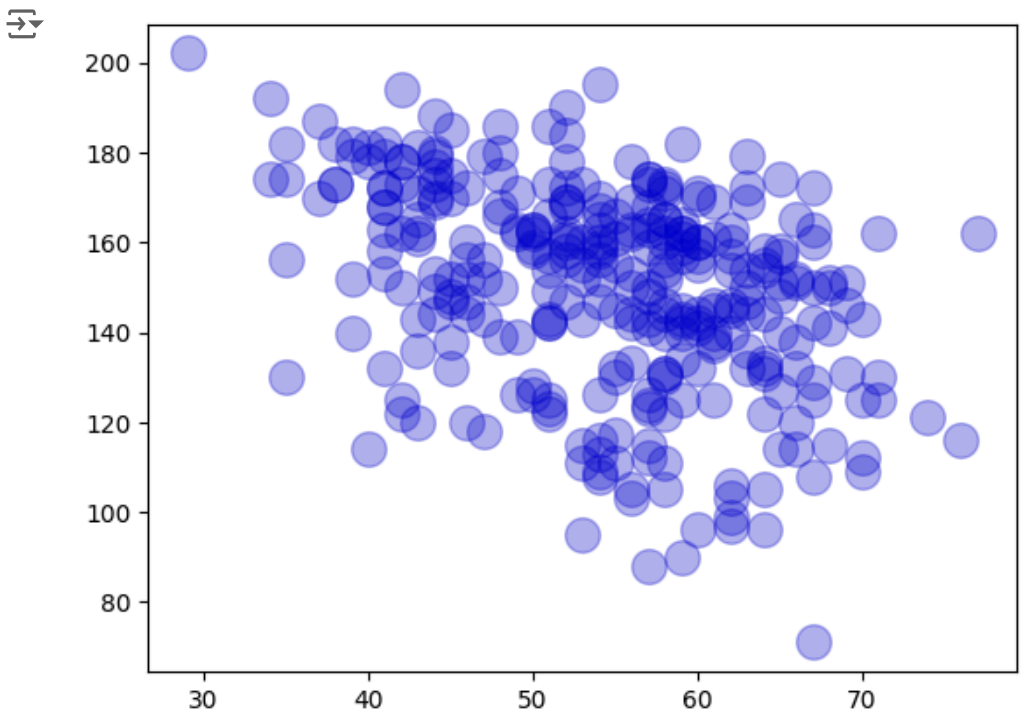


Scatterplot

```
def scatter(x, y, alpha=.3, size=200, color="mediumblue"):
    import matplotlib.pyplot as plt

    fig, ax = plt.subplots()

    ax.scatter(x=x, y=y, alpha=alpha, s = size, c = color);


age = df["age"]
max_hr = df["max_hr"]
sex = df["sex"]
```

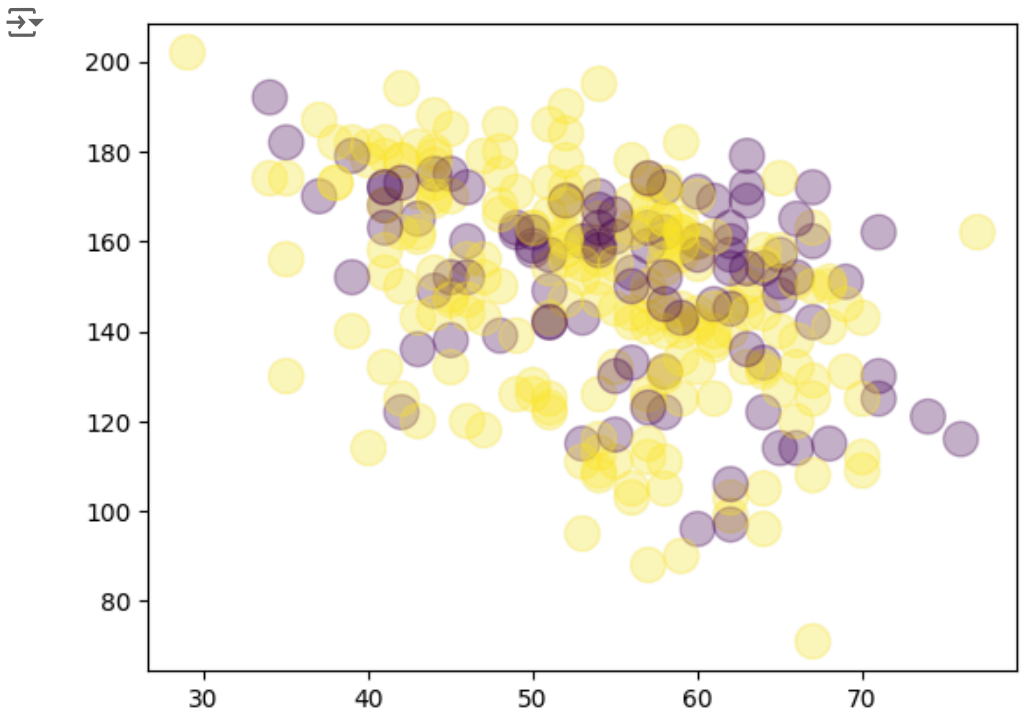
Call the scatter() function

```
# use default color
scatter(age, max_hr)
```



Map the color to a categorical column

```
scatter(age, max_hr, color=sex)
```



Barplot

```
def bar(labels, height, color="deepskyblue"):
    import matplotlib.pyplot as plt

    fig, ax = plt.subplots()
    ax.bar(x=labels, height=height, color=color, edgecolor="black")
```

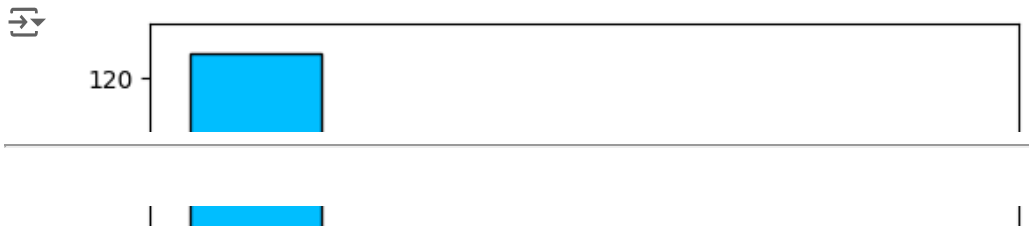
```
# make "age" a categorical variable

df["age_groups"] = pd.cut(df["age"], [29, 39, 49, 59, 69, 79],
                          labels=["thirties","forties","fifties","sixties", "seventies"])

height = df["age_groups"].value_counts()
height
```

age_groups	count
fifties	125
sixties	80
forties	72
thirties	15
seventies	10

dtype: int64

Call the bar() function

```
# use default color
bar(labels=["thirties","forties","fifties","sixties", "seventies"], height=height)
```



Pie Chart

```
def pie(labels, values, colors):
    import matplotlib.pyplot as plt

    fig, ax = plt.subplots(figsize = (15, 5))
    ax.pie(x=values, labels=labels, colors=colors, autopct='%.1f%%',
        wedgeprops = {"edgecolor" : "black",
                      'linewidth': 1,
                      'antialiased': True});
```

```
values = df["age_groups"].value_counts()
values
```

age_groups	count
fifties	125
sixties	80
forties	72
thirties	15
seventies	10

dtype: int64