Pandas Data Visualization

```
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
--------------
```

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

df.head()
```

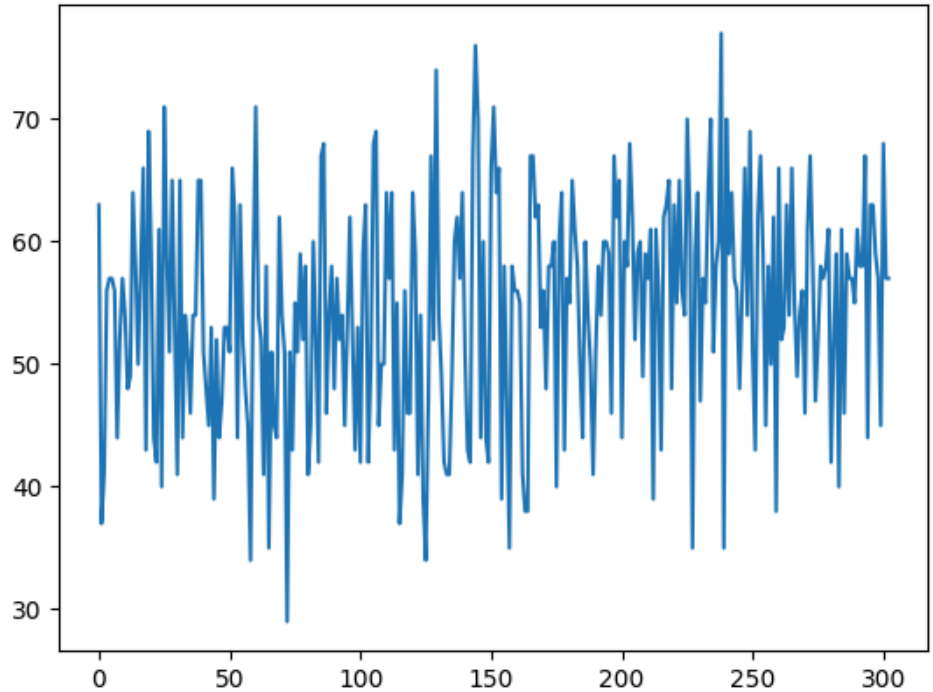
	age	sex	chest_pain	rest_bp	chol	max_hr	st_depr	heart_disease
0	63	female	3	145	233	150	2.3	1
1	37	female	2	130	250	187	3.5	1
2	41	male	1	130	204	172	1.4	1
3	56	female	1	120	236	178	0.8	1
4	57	male	0	120	354	163	0.6	1

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

Line Plot

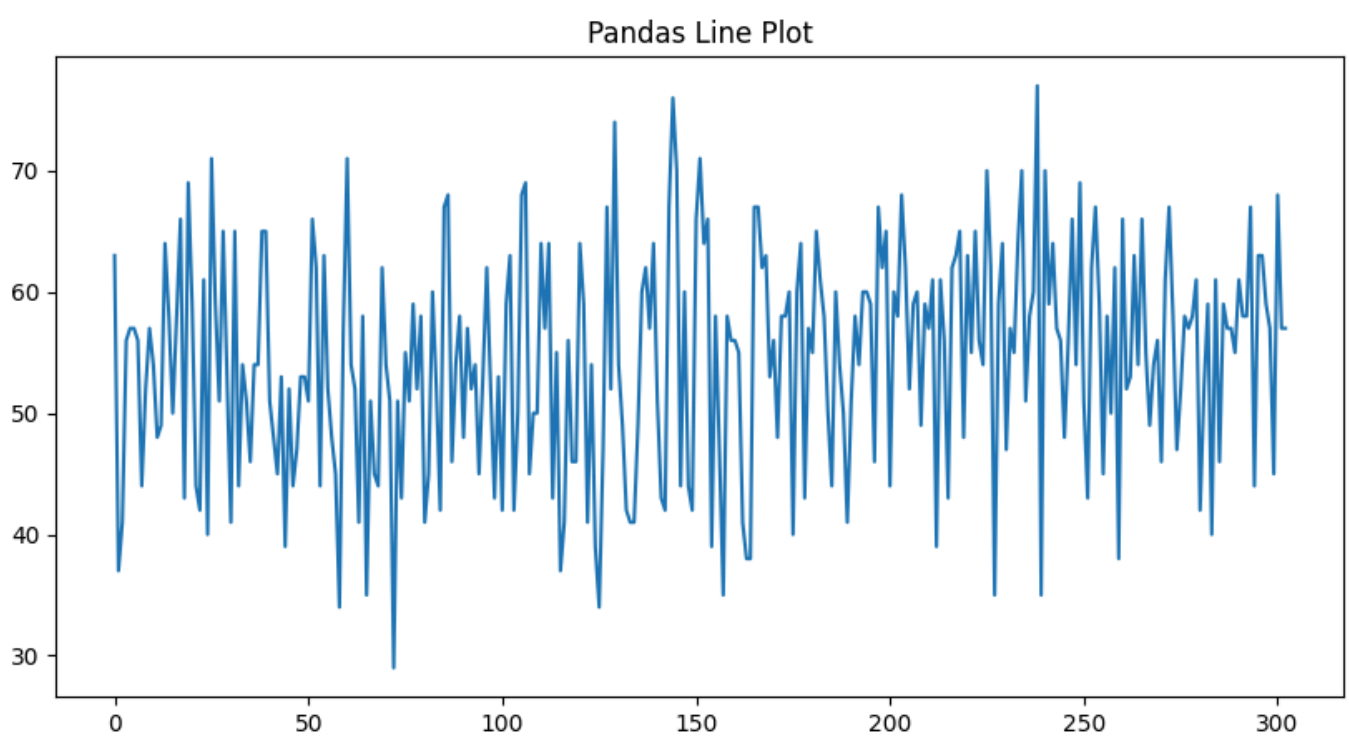
Column

```
# x-axis defaults to the row's index number

df["age"].plot(kind="line");
```



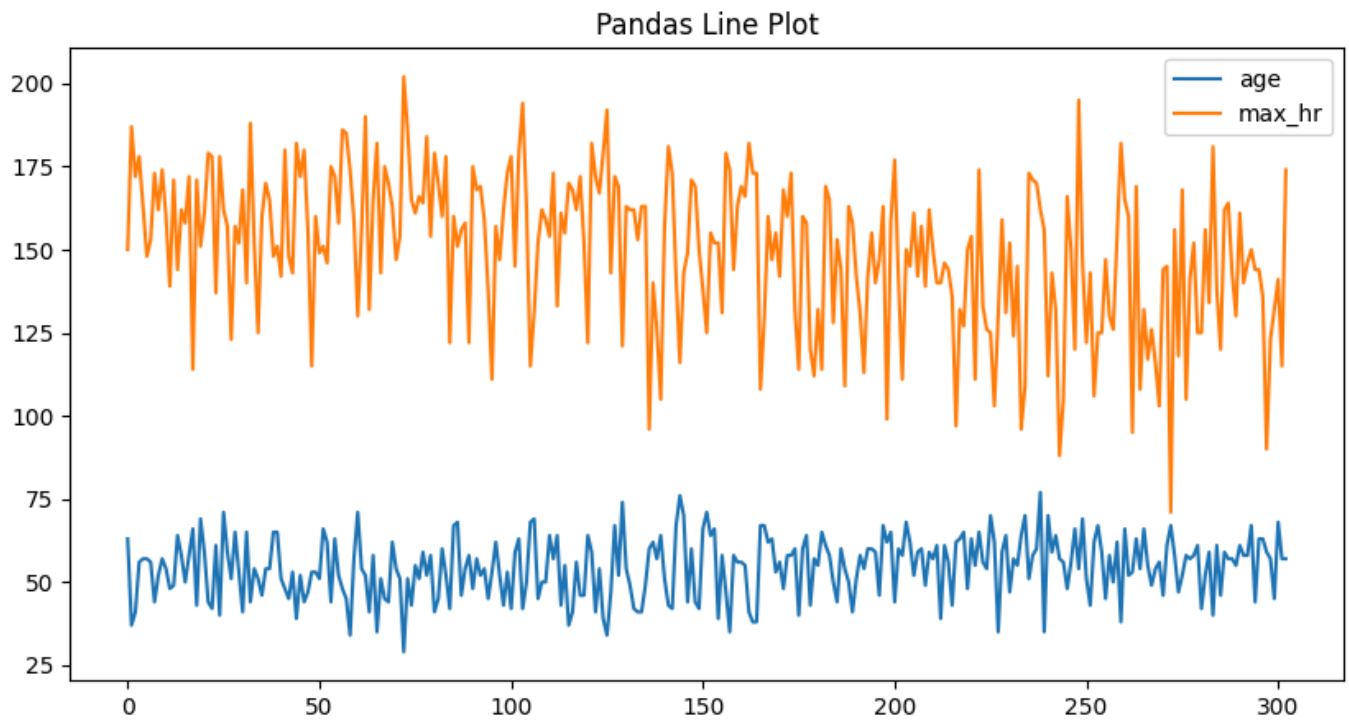
You can set many of the plot's parameters within the plot() method.

```
df["age"].plot(kind="line", figsize=(10,5), title="Pandas Line Plot");
```



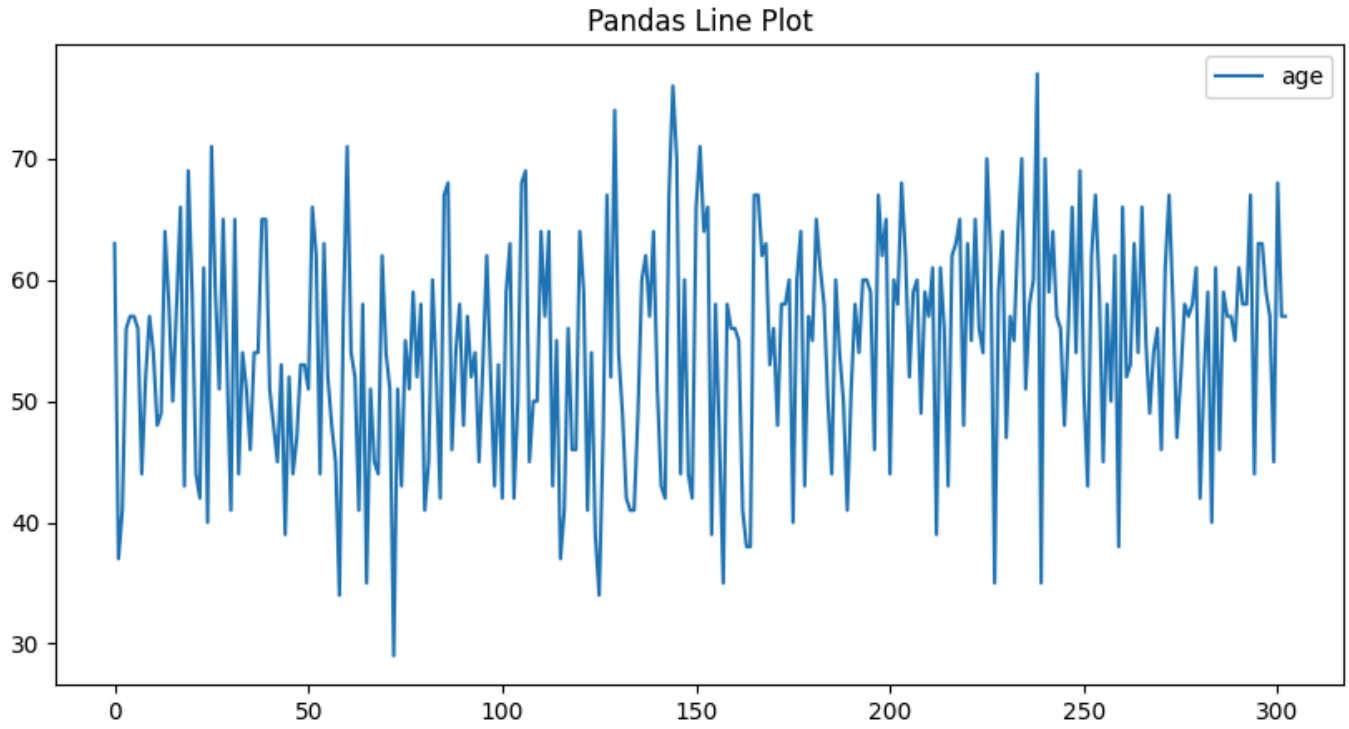
Multiple columns

```
df[["age", "max_hr"]].plot(kind="line", figsize=(10,5), title="Pandas Line Plot");
```



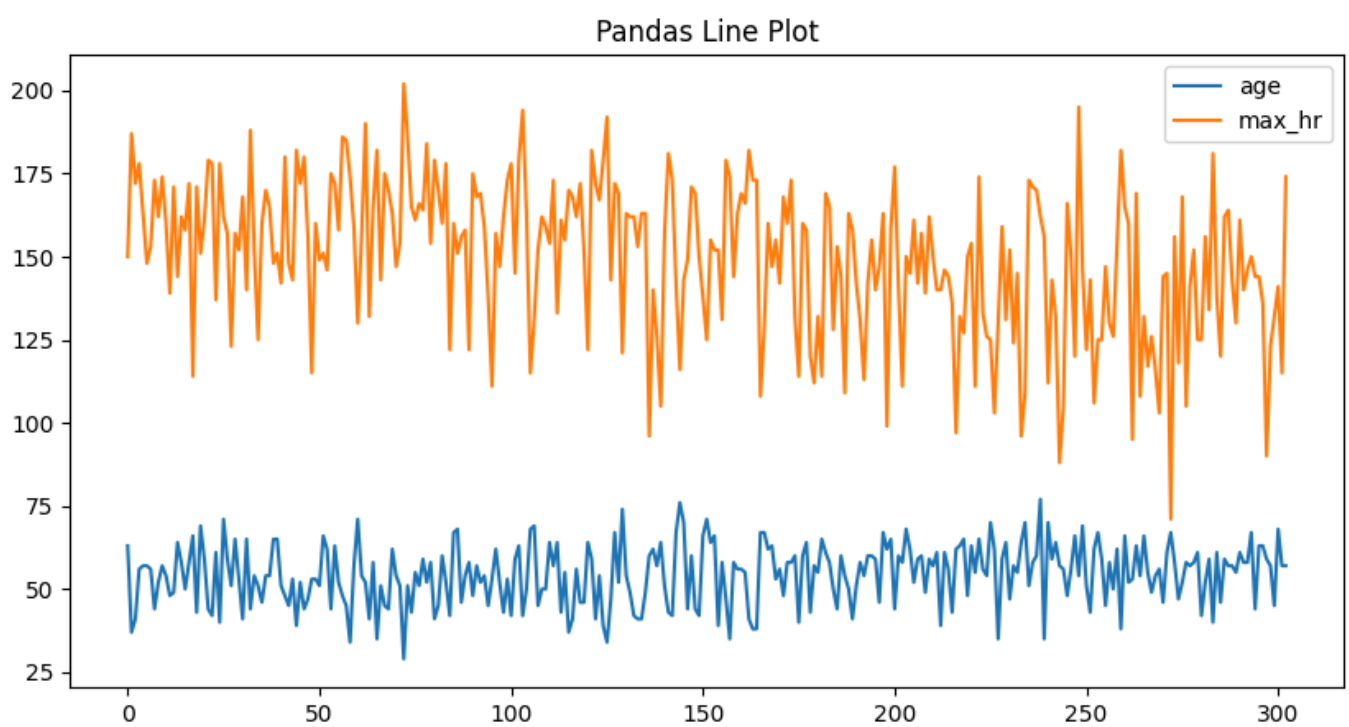
DataFrame

```
# x-axis defaults to the row's index number

df.plot(y="age", kind="line", figsize=(10,5), title="Pandas Line Plot");
```



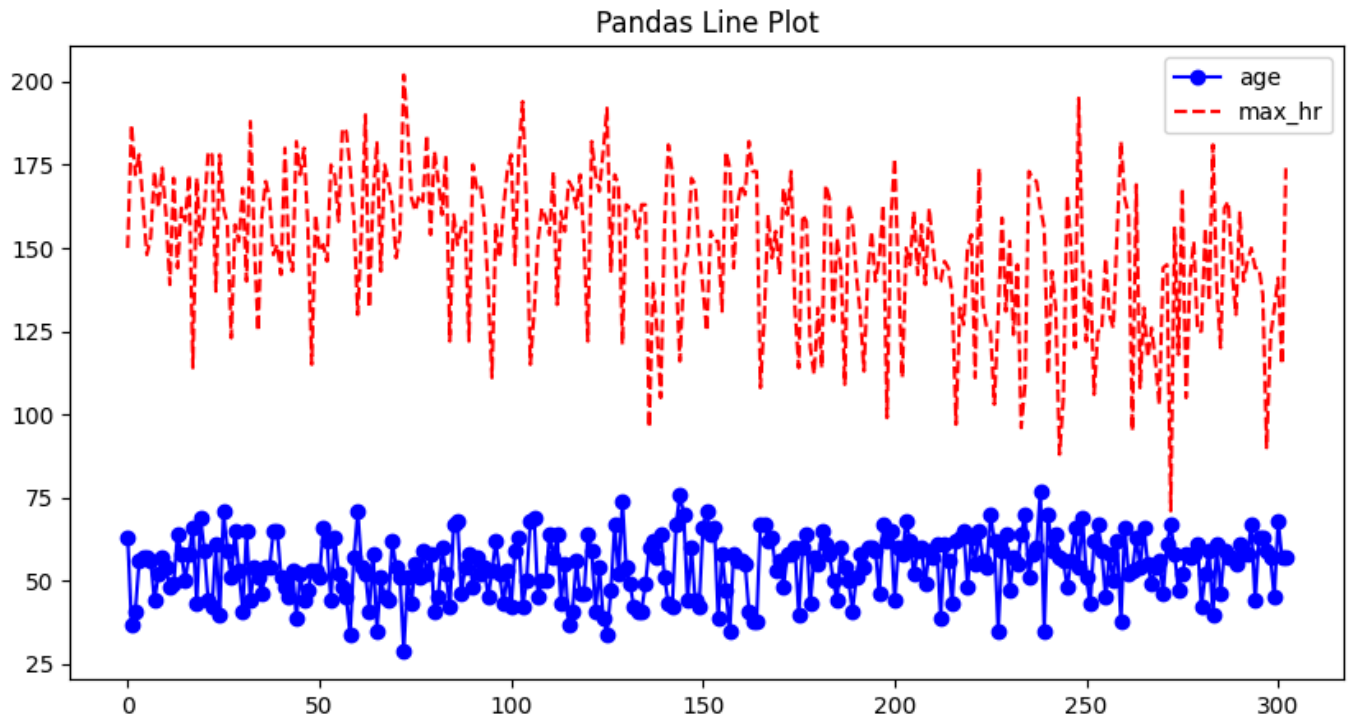
Multiple columns

```
df.plot(y=["age", "max_hr"], kind="line", figsize=(10,5), title="Pandas Line Plot");
```



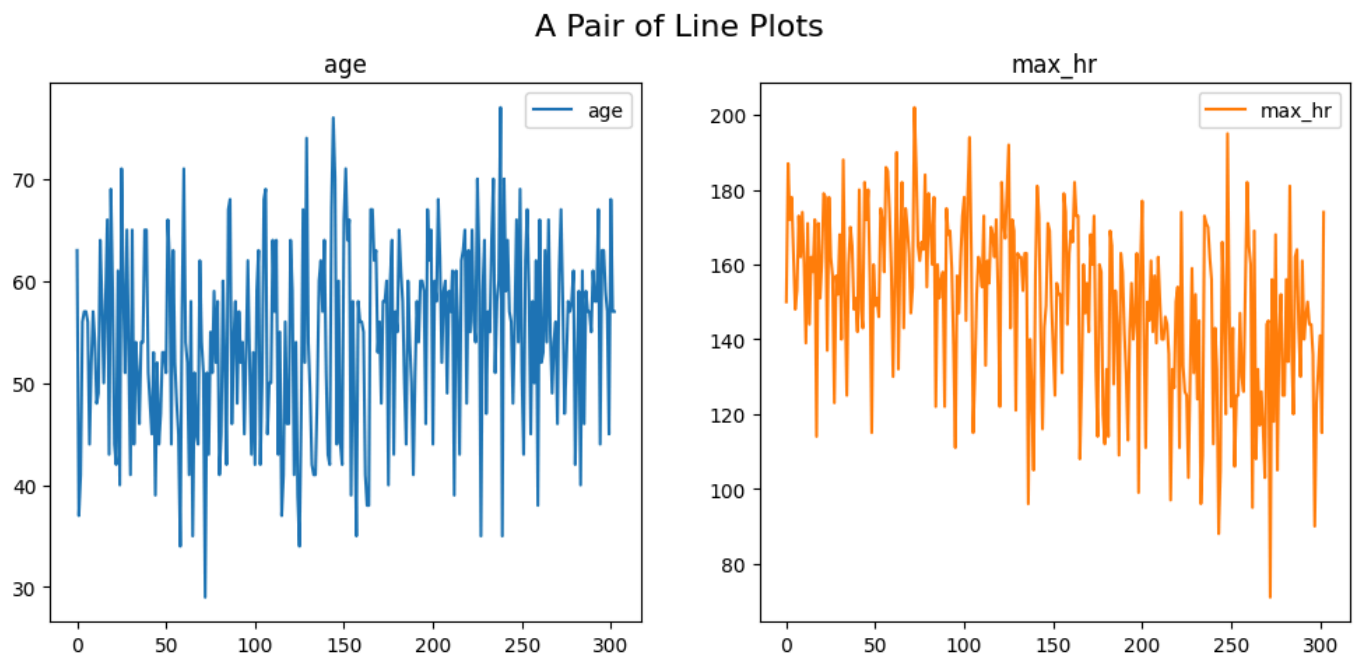
Set a custom style

```
df.plot(kind="line", y=["age", "max_hr"],
             figsize=(10,5), title="Pandas Line Plot",
             style=['b-o', 'r--']); # 'b:', 'r-.'
```



Define subplots to render plots in separate axes.

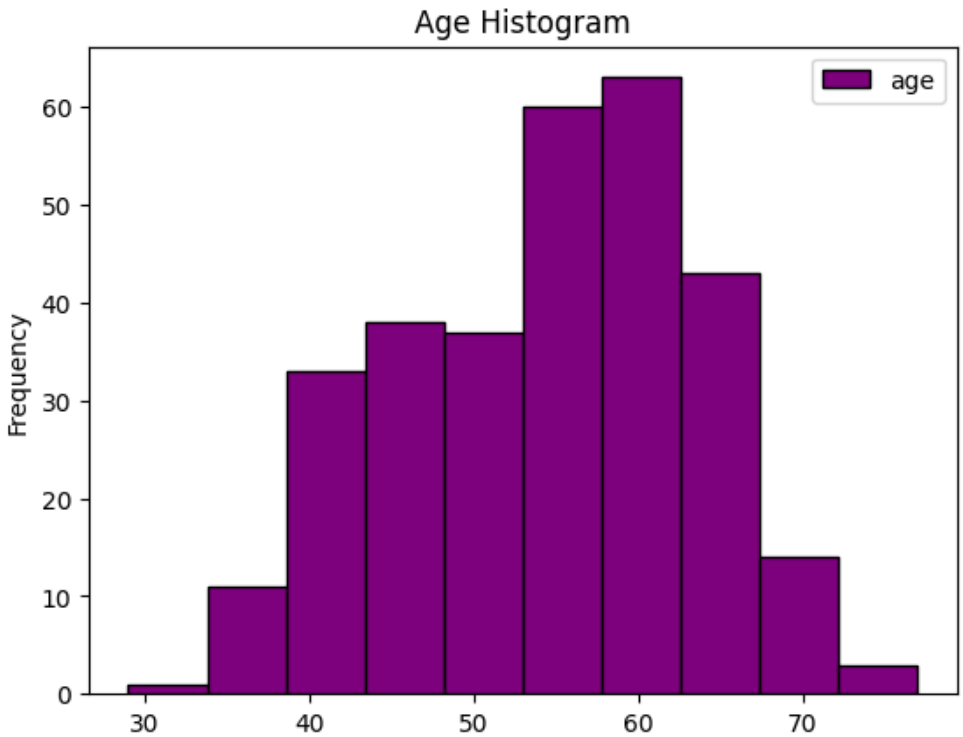
```
df.plot(kind="line",  y=["age", "max_hr"],
             figsize=(12,5), title=["age", "max_hr"],
             subplots=True, layout=(1,2))

# Set the title of the whole figure
plt.suptitle("A Pair of Line Plots", fontsize=16);
```



Histogram

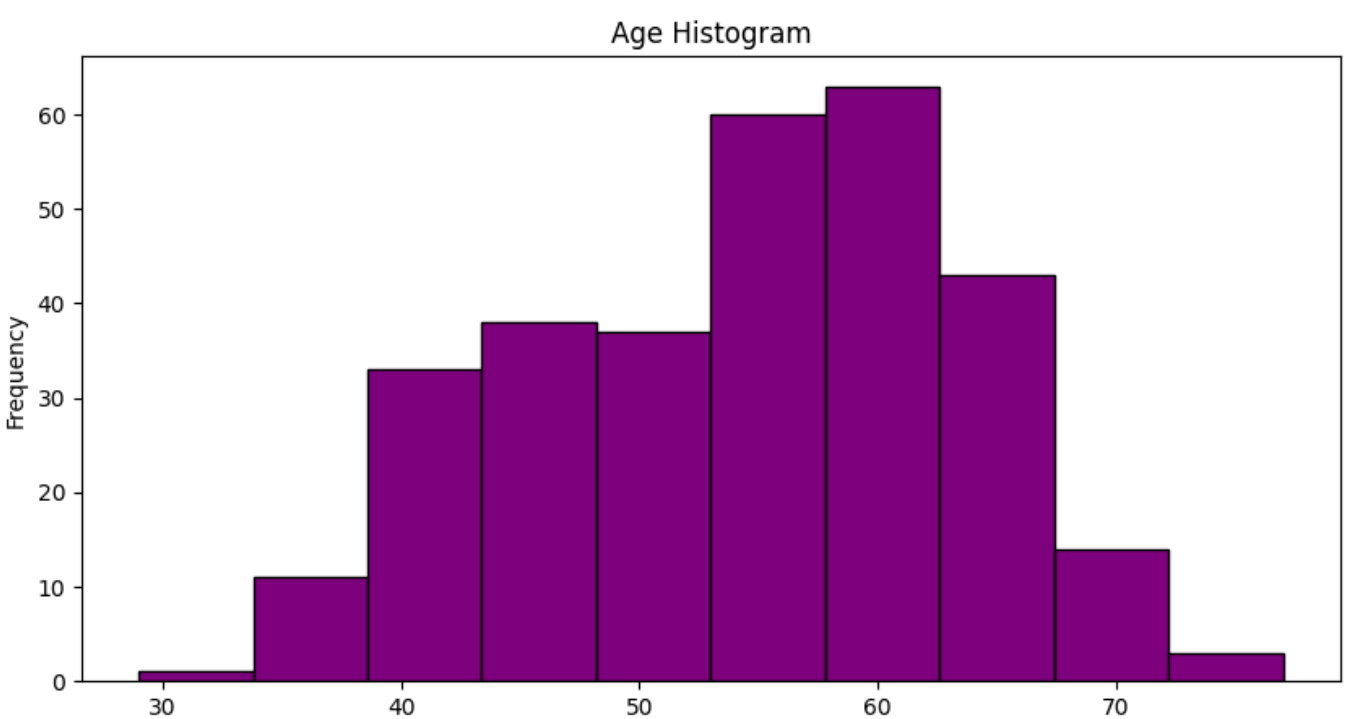
Column

```
# Set legend=True to display a legend (defaults to False when plotting from a column)

df["age"].plot(kind="hist", color="purple",
               edgecolor='black', title="Age Histogram", legend=True);
```



DataFrame

```
# Set legend=False to not display a legend (defaults to True when plotting from a DataFrame)

df.plot(y="age", kind="hist", color="purple",
        edgecolor='black', title="Age Histogram", figsize=(10, 5), legend=False);
```



Bar Plot

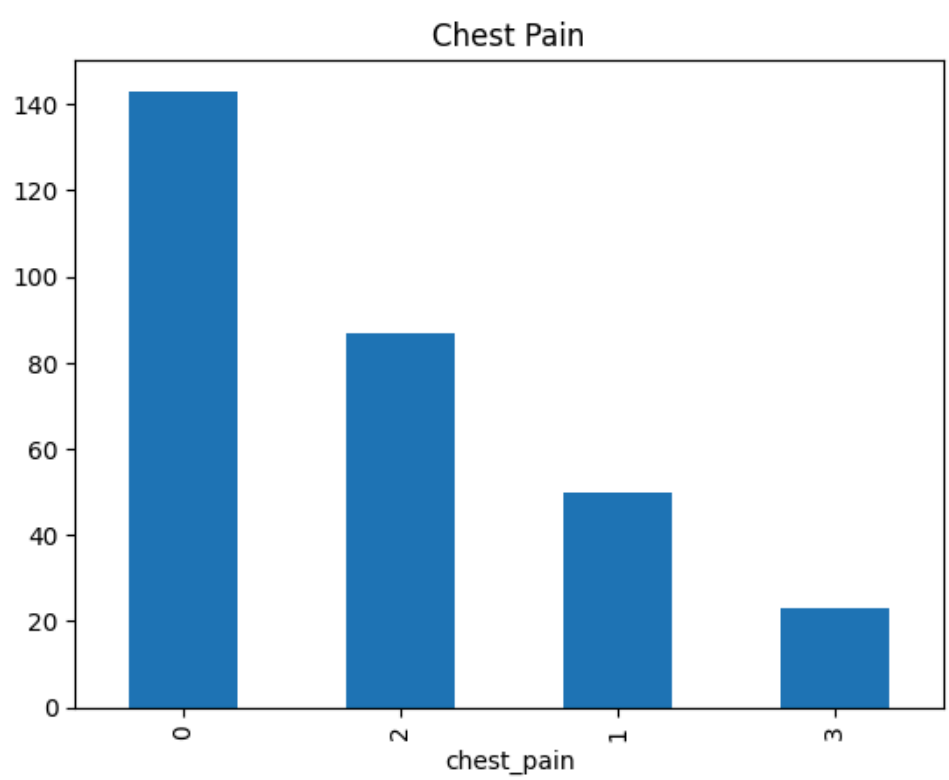
Categorical variable

```
df["chest_pain"].value_counts()
```

chest_pain	count
0	143
2	87
1	50
3	23

dtype: int64

```
df["chest_pain"].value_counts().plot(kind="bar", title="Chest Pain");
```

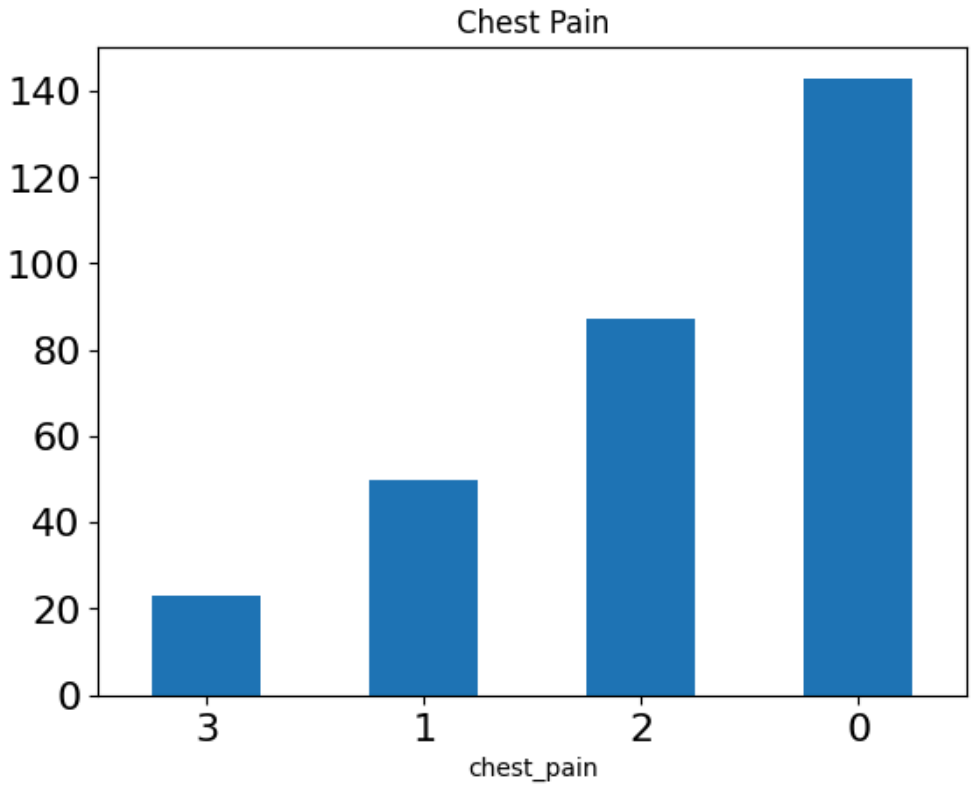


Rotate the ticks and increase font size

```
df["chest_pain"].value_counts().plot(kind="bar", title="Chest Pain",
                                     rot="horizontal", fontsize=16);
```

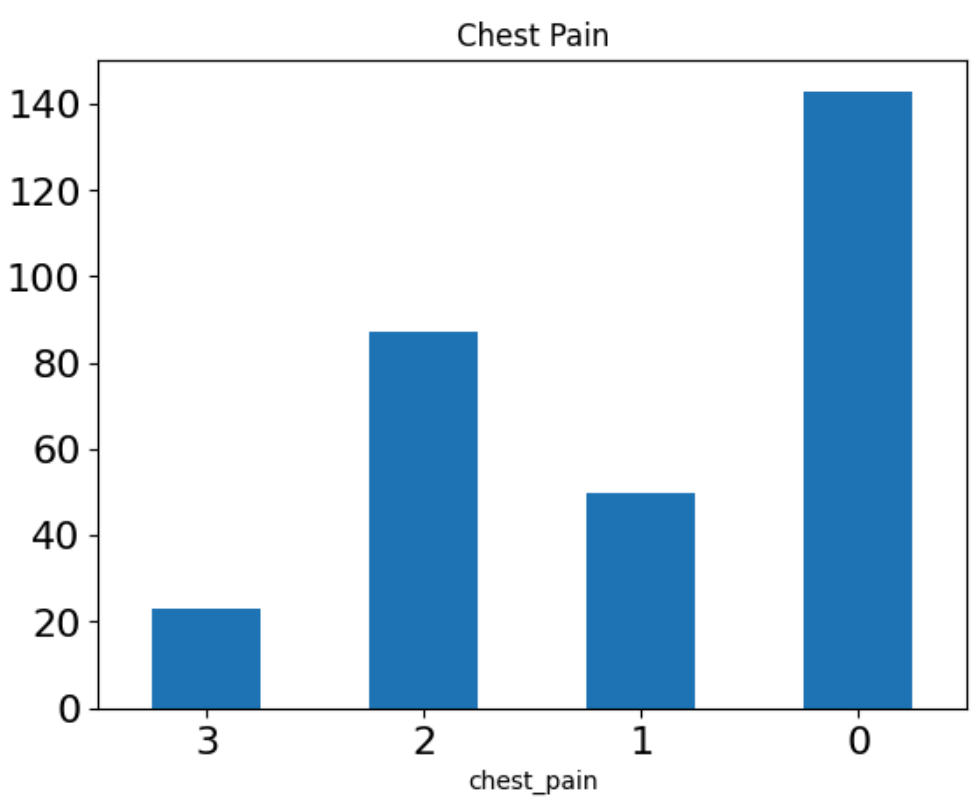
Sort in ascending order

```
# Set ascending to True
df["chest_pain"].value_counts(ascending=True).plot(kind="bar", title="Chest Pain",
                                                   rot="horizontal", fontsize=16);
```



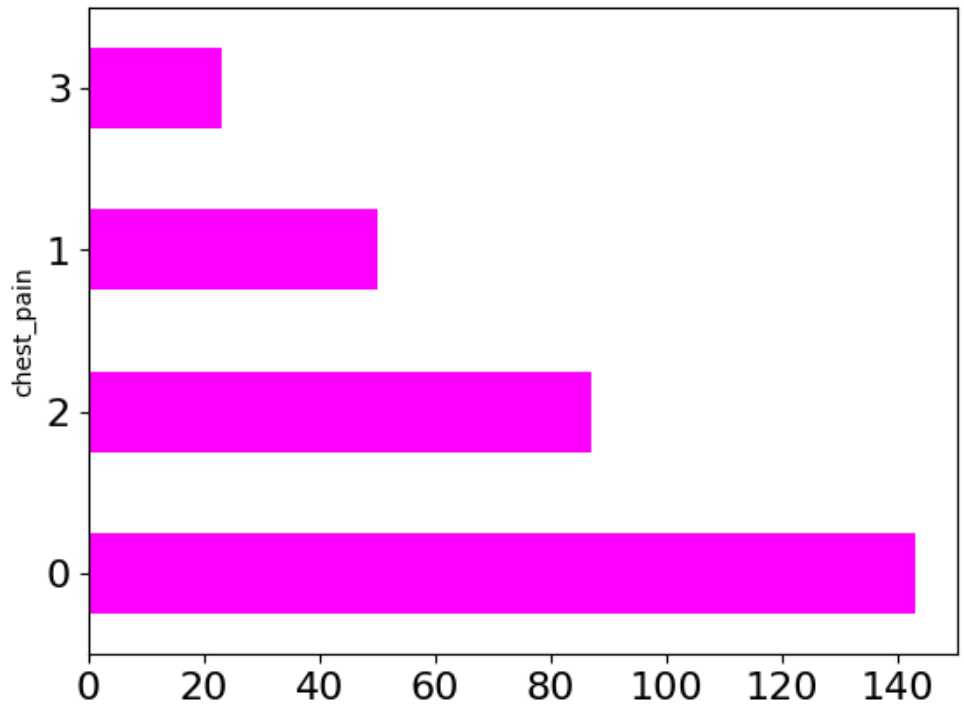
Display in order of index position

```
# Set sort to False
df["chest_pain"].value_counts(sort=False).plot(kind="bar", title="Chest Pain",
                                rot="horizontal", fontsize=16);
```



Horizontal bar plot

```
df["chest_pain"].value_counts().plot(kind="barh", color="magenta", fontsize=16);
```



Joint: categorical x aggregated continuous variable

Groupby and Aggregate

Group by a categorical variable and aggregate over a continuous variable

```
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

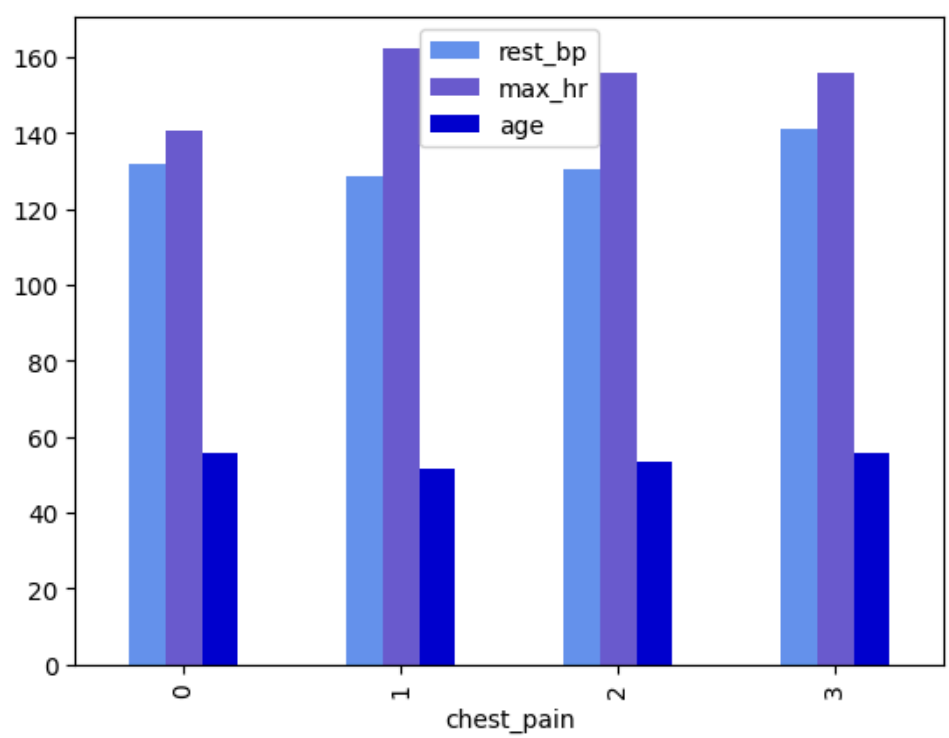
groupby()

```
# 1 categorical variable and 3 continuous variables
# Get the mean value of the continuous variables for each category of the categorical variable

   # Categorical    # Continuous variables
df[["chest_pain", "rest_bp", "max_hr", "age"]].groupby(["chest_pain"]).mean()
```

chest_pain	rest_bp	max_hr	age
0	132.020979	140.538462	55.692308
1	128.400000	162.420000	51.360000
2	130.379310	155.609195	53.517241
3	140.869565	155.956522	55.869565

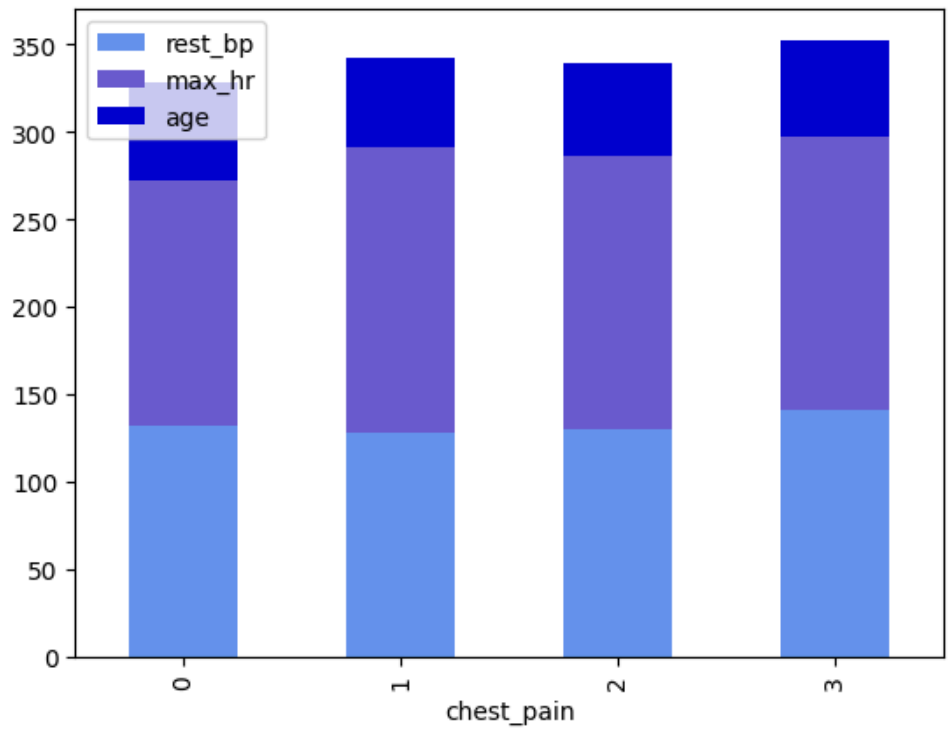
Bar plot

```
colors=["cornflowerblue", "slateblue", "mediumblue"]

   # Categorical    # Continuous variables
df[["chest_pain", "rest_bp", "max_hr", "age"]].groupby(["chest_pain"]).mean().plot(kind="bar", c
```



Stacked bar plot

```
colors=["cornflowerblue", "slateblue", "mediumblue"]

   # Categorical    # Continuous variables
df[["chest_pain", "rest_bp", "max_hr", "age"]].groupby(["chest_pain"]).mean().plot(kind="bar",
                                                                                    color=colors,
```



Relocate the legend

```
colors=["cornflowerblue", "slateblue", "mediumblue"]

   # Categorical    # Continuous variables
df[["chest_pain", "rest_bp", "max_hr", "age"]].groupby(["chest_pain"]).mean().plot(kind="bar",
                                                                                    color=colors,

# Display a legend (just outside of the plot; up 1, and over to the right 1)
plt.legend(bbox_to_anchor=(1, 1));
```