

# Histogram

```
In [1]: import pandas as pd
df = pd.read_excel('data.xlsx', sheet_name='income')
```

```
In [2]: df
```

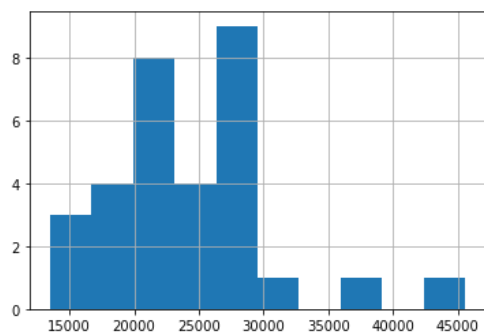
```
Out[2]:
```

	จังหวัด	รายได้เฉลี่ยต่อครัวเรือนปี พ.ศ. 2558 (บาท)
0	กรุงเทพมหานคร	45572
1	กาฬสินธุ์	15452
2	กำแพงเพชร	20140
3	ฉะเชิงเทรา	27555
4	ชลบุรี	27257
5	ชุมพร	29306
6	เชียงราย	13497
7	ตรัง	23309
8	ตราด	25333
9	นครนายก	23555
10	นครพนม	15804
11	นครราชสีมา	26376
12	นนทบุรี	36884
13	น่าน	17886
14	พระนครศรีอยุธยา	28379
15	พังงา	28842
16	พัทลุง	21478
17	พิษณุโลก	19235
18	ภูเก็ต	31500
19	มหาสารคาม	18540
20	มุกดาหาร	22870
21	ยโสธร	19518
22	ร้อยเอ็ด	19990
23	ระนอง	22035
24	ลำปาง	20466
25	ลำพูน	23834
26	สงขลา	27660
27	สมุทรสาคร	29347
28	สระบุรี	29413
29	สุโขทัย	22252
30	หนองคาย	21207

```
In [3]: df.rename(columns={'รายได้เฉลี่ยต่อครัวเรือนปี พ.ศ. 2558 (บาท)': 'income'}, inplace=True)
```

```
In [4]: df['income'].hist()
```

```
Out[4]: <AxesSubplot:>
```



# Scatter plot

```
In [5]: import pandas as pd
df1 = pd.read_csv('df1', index_col=0)
```

```
In [6]: df1
```

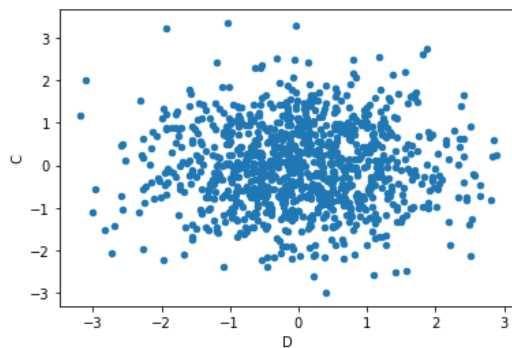
```
Out[6]:
```

	A	B	C	D
2000-01-01	1.339091	-0.163643	-0.646443	1.041233
2000-01-02	-0.774984	0.137034	-0.882716	-2.253382
2000-01-03	-0.921037	-0.482943	-0.417100	0.478638
2000-01-04	-1.738808	-0.072973	0.056517	0.015085
2000-01-05	-0.905980	1.778576	0.381918	0.291436
...	...	...	...	...
2002-09-22	1.013897	-0.288680	-0.342295	-0.638537
2002-09-23	-0.642659	-0.104725	-0.631829	-0.909483
2002-09-24	0.370136	0.233219	0.535897	-1.552605
2002-09-25	0.183339	1.285783	-1.052593	-2.565844
2002-09-26	0.775133	-0.850374	0.486728	-1.053427

1000 rows x 4 columns

```
In [7]: df1.plot.scatter(x='D', y='C')
```

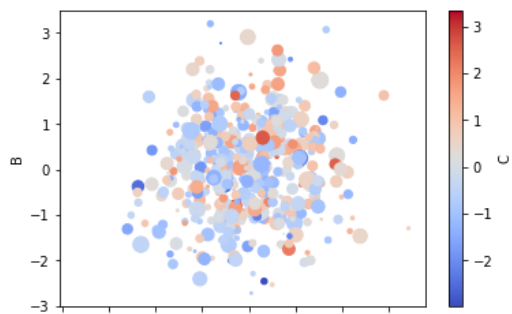
```
Out[7]: <AxesSubplot: xlabel='D', ylabel='C'>
```



```
In [8]: df1.plot.scatter(x='A', y='B', c='C', cmap='coolwarm', s=df1['D']*50)
```

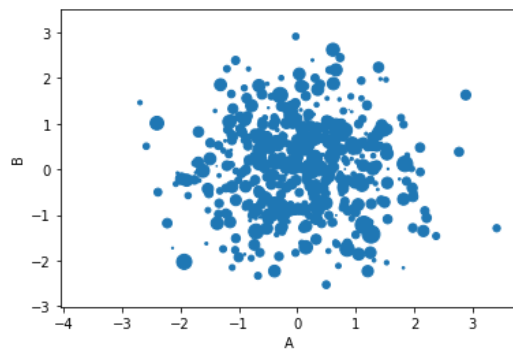
```
/Users/jakapongtosunpul/opt/anaconda3/lib/python3.9/site-packages/matplotlib/collections.py:982: RuntimeWarning: invalid value encountered in sqrt
  scale = np.sqrt(self._sizes) * dpi / 72.0 * self._factor
```

```
Out[8]: <AxesSubplot: xlabel='A', ylabel='B'>
```



```
In [9]: df1.plot.scatter(x='A', y='B', s=df1['C']*50)
```

```
Out[9]: <AxesSubplot: xlabel='A', ylabel='B'>
```



## Boxplot

```
In [10]: df2 = pd.read_csv('df2')
```

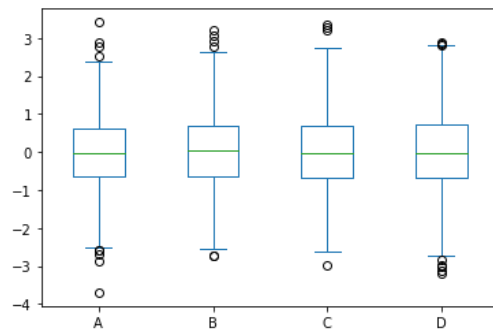
```
In [11]: df2
```

```
Out[11]:
```

	a	b	c	d
0	0.039762	0.218517	0.103423	0.957904
1	0.937288	0.041567	0.899125	0.977680
2	0.780504	0.008948	0.557808	0.797510
3	0.672717	0.247870	0.264071	0.444358
4	0.053829	0.520124	0.552264	0.190008
5	0.286043	0.593465	0.907307	0.637898
6	0.430436	0.166230	0.469383	0.497701
7	0.312296	0.502823	0.806609	0.850519
8	0.187765	0.997075	0.895955	0.530390
9	0.908162	0.232726	0.414138	0.432007

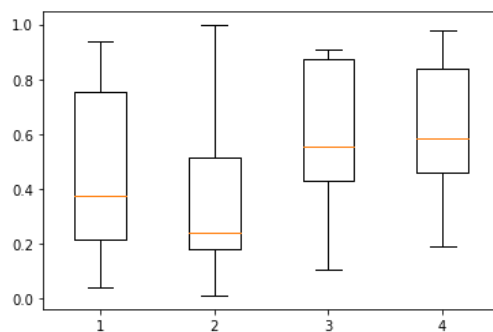
```
In [12]: df1.plot.box() # Can also pass a by= argument for groupby
```

```
Out[12]: <AxesSubplot:>
```



## Using 'matplotlib'

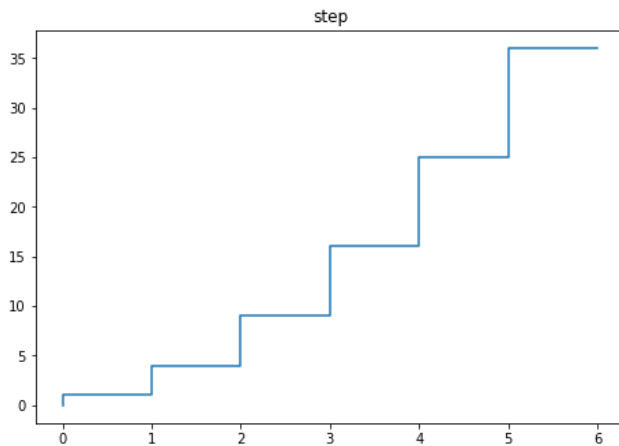
```
In [13]: # Box plot
import matplotlib.pyplot as plt
plt.boxplot(df2);
```



## Step plot

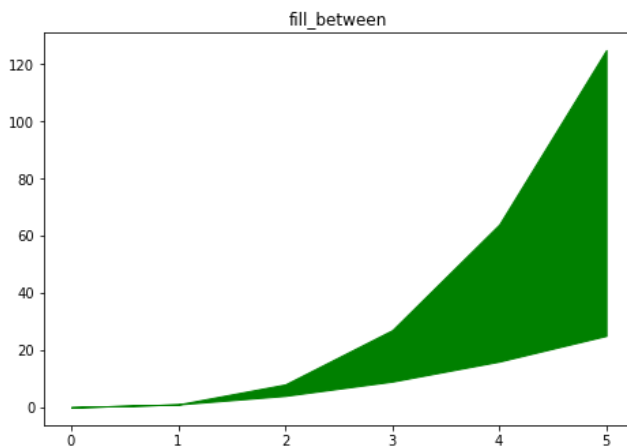
```
In [14]: import numpy as np
x = np.arange(0,7)
y = x**2
fig = plt.figure()
axes = fig.add_axes([0,0,1,1])
axes.step(x, y)
axes.set_title("step")
```

Out[14]: Text(0.5, 1.0, 'step')



## Fill between graph

```
In [15]: import numpy as np
x = np.arange(0,6)
y = x**2
y2= x**3
fig = plt.figure()
axes = fig.add_axes([0,0,1,1])
axes.fill_between(x, y, y2, color="green");
axes.set_title("fill_between");
```



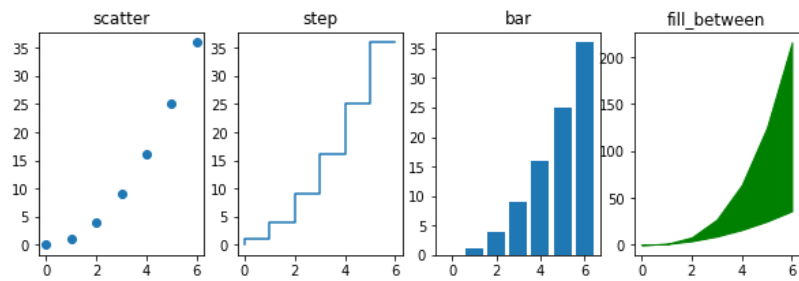
```
In [16]: import numpy as np
x = np.arange(0,7)
y = x**2
y2= x**3
fig, axes = plt.subplots(1, 4, figsize=(10,3))

axes[0].scatter(x, y)
axes[0].set_title("scatter")

axes[1].step(x, y)
axes[1].set_title("step")

axes[2].bar(x, y)
axes[2].set_title("bar")

axes[3].fill_between(x, y, y2, color="green");
axes[3].set_title("fill_between");
```

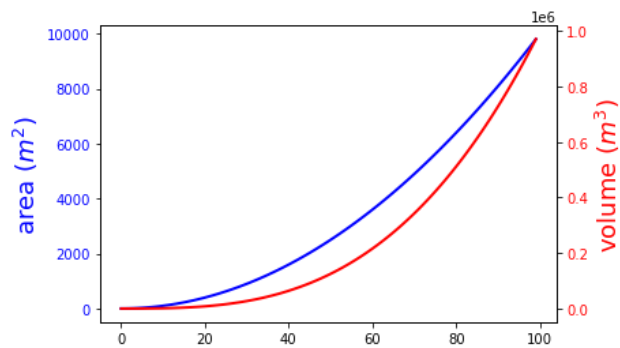


## Plot twin axes

```
In [17]: import numpy as np
x = np.arange(0,100)

import matplotlib.pyplot as plt
fig, ax1 = plt.subplots()
ax1.plot(x, x**2, lw=2, color="blue")
ax1.set_ylabel("area $(m^2)$", fontsize=18, color="blue")
for label in ax1.get_yticklabels():
    label.set_color("blue")

ax2 = ax1.twinx()
ax2.plot(x, x**3, lw=2, color="red")
ax2.set_ylabel("volume $(m^3)$", fontsize=18, color="red")
for label in ax2.get_yticklabels():
    label.set_color("red")
```



In [ ]: