CHAPTER

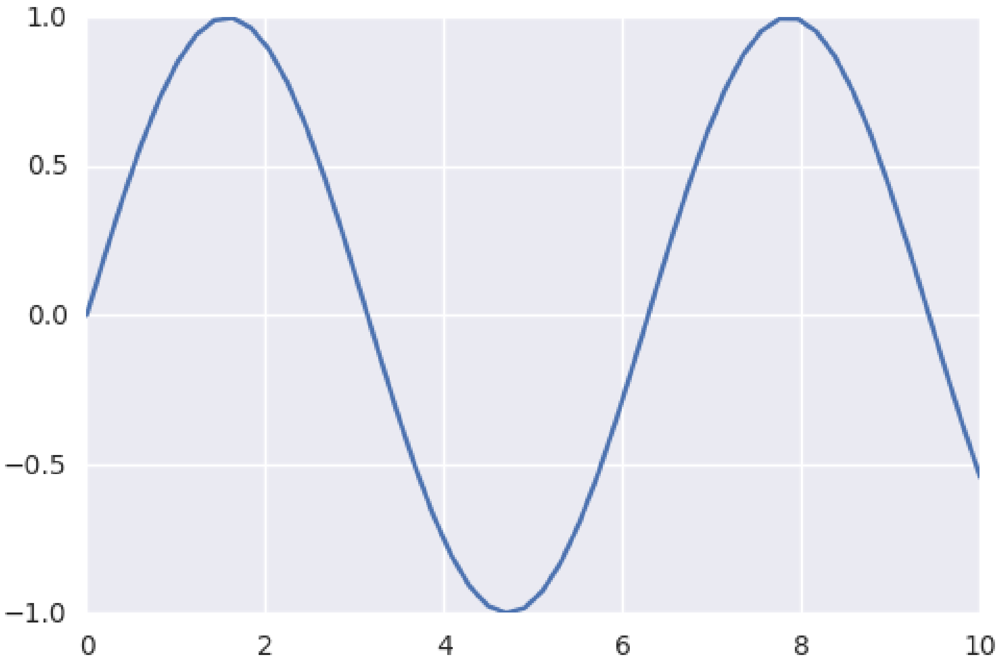
FIVE

MATPLOTLIB: DATA VISUALIZATION

Sources - Nicolas P. Rougier: http://www.labri.fr/perso/nrougier/teaching/matplotlib - https://www.kaggle.com/benhamner/d/uciml/iris/python-data-visualizations

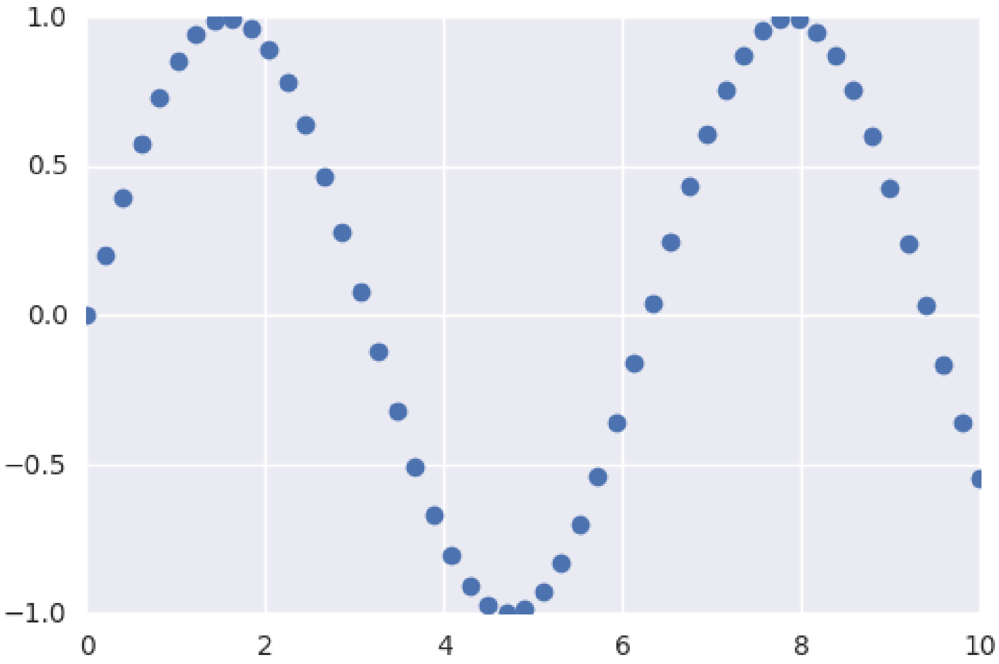
Basic plots

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

x = np.linspace(0, 10, 50)
sinus = np.sin(x)

plt.plot(x, sinus)
plt.show()
```

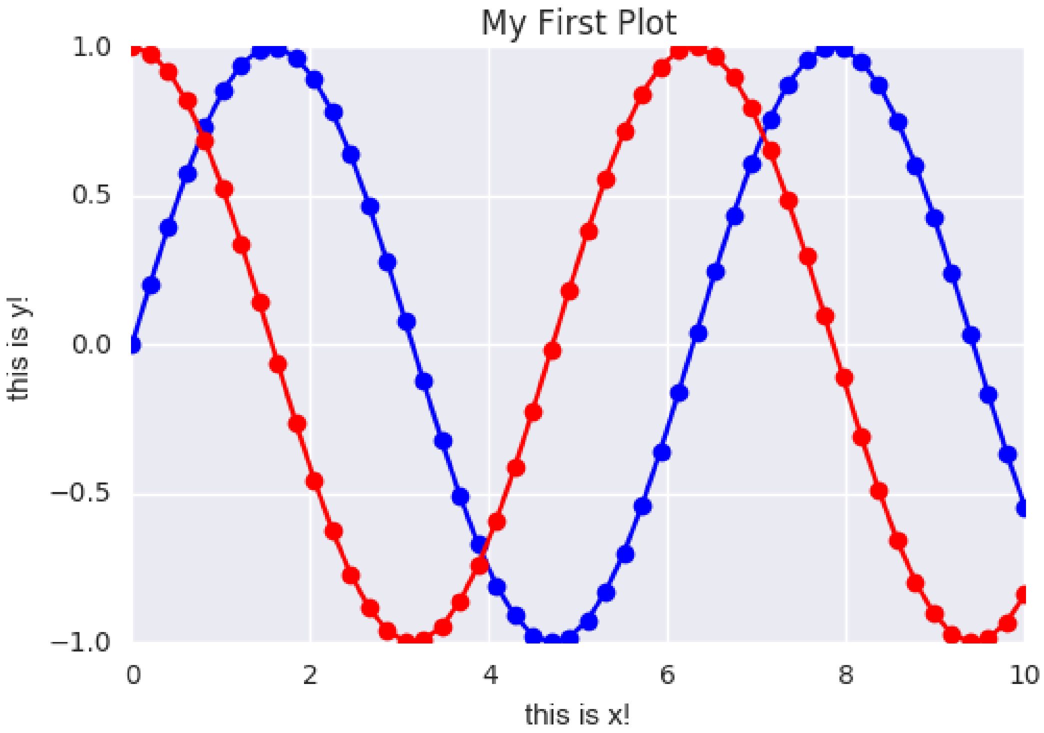


```
plt.plot(x, sinus, "o")
plt.show()
```

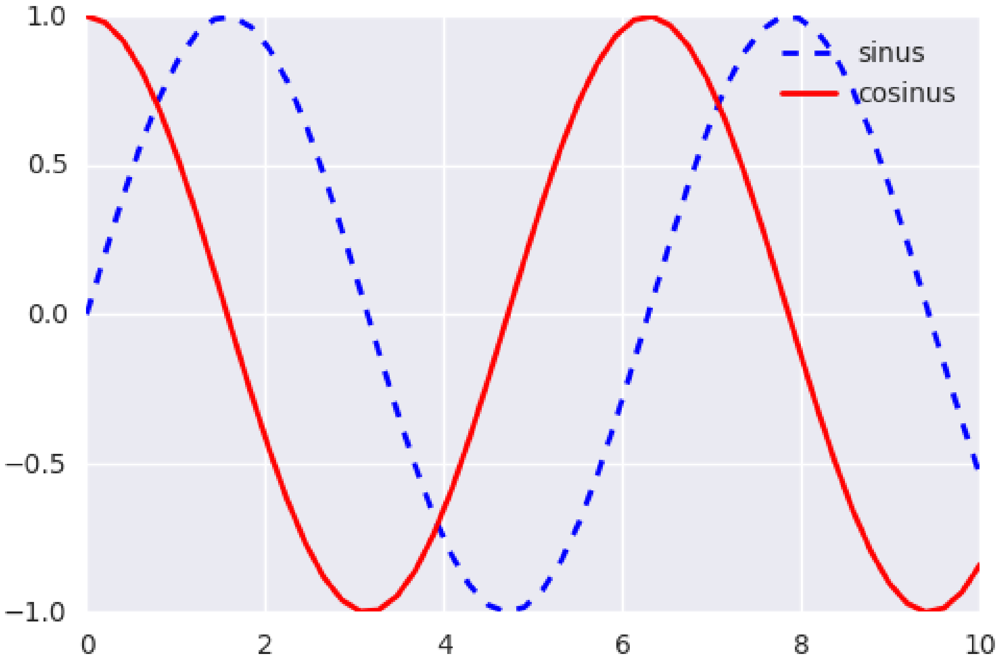
```
# use plt.plot to get color / marker abbreviations
```



```
# Rapid multiplot

cosinus = np.cos(x)
plt.plot(x, sinus, "-b", x, sinus, "ob", x, cosinus, "-r", x, cosinus, "or")
plt.xlabel('this is x!')
plt.ylabel('this is y!')
plt.title('My First Plot')
plt.show()
```



```
# Step by step
plt.plot(x, sinus, label='sinus', color='blue', linestyle='--', linewidth=2)
plt.plot(x, cosinus, label='cosinus', color='red', linestyle='-', linewidth=2)
plt.legend()
plt.show()
```



Scatter (2D) plots

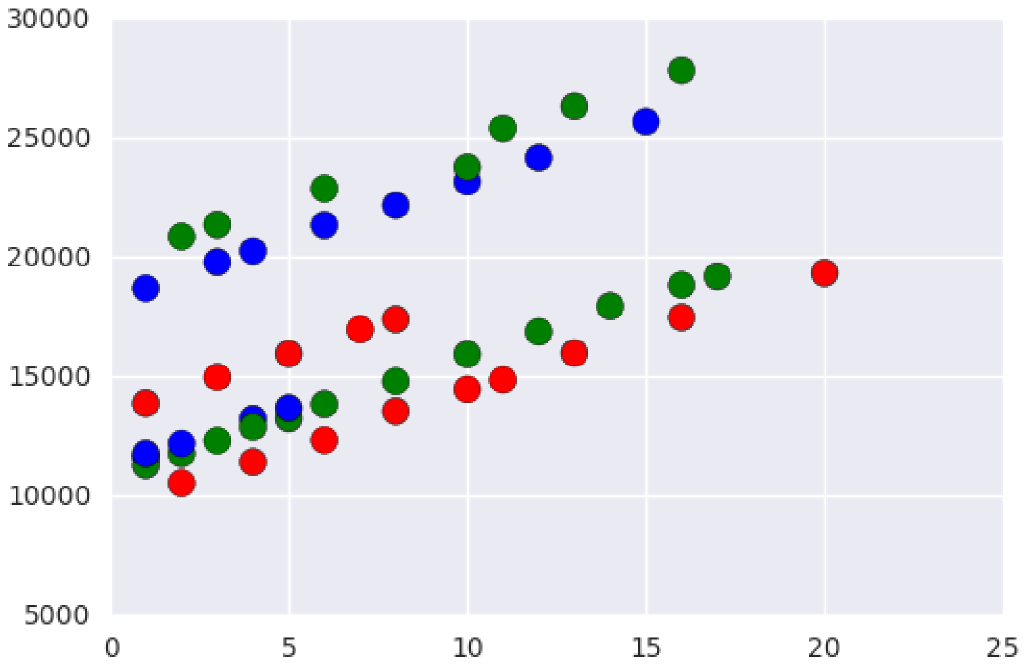
Load dataset

```
import pandas as pd
try:
    salary = pd.read_csv("../data/salary_table.csv")
except:
    url = 'https://raw.github.com/duchesnay/pylearn-doc/master/data/salary_table.csv'
    salary = pd.read_csv(url)

df = salary
```

Simple scatter with colors

```
colors = colors_edu = {'Bachelor':'r', 'Master':'g', 'Ph.D':'blue'}
plt.scatter(df['experience'], df['salary'], c=df['education'].apply(lambda x:␣
↪colors[x]), s=100)
```

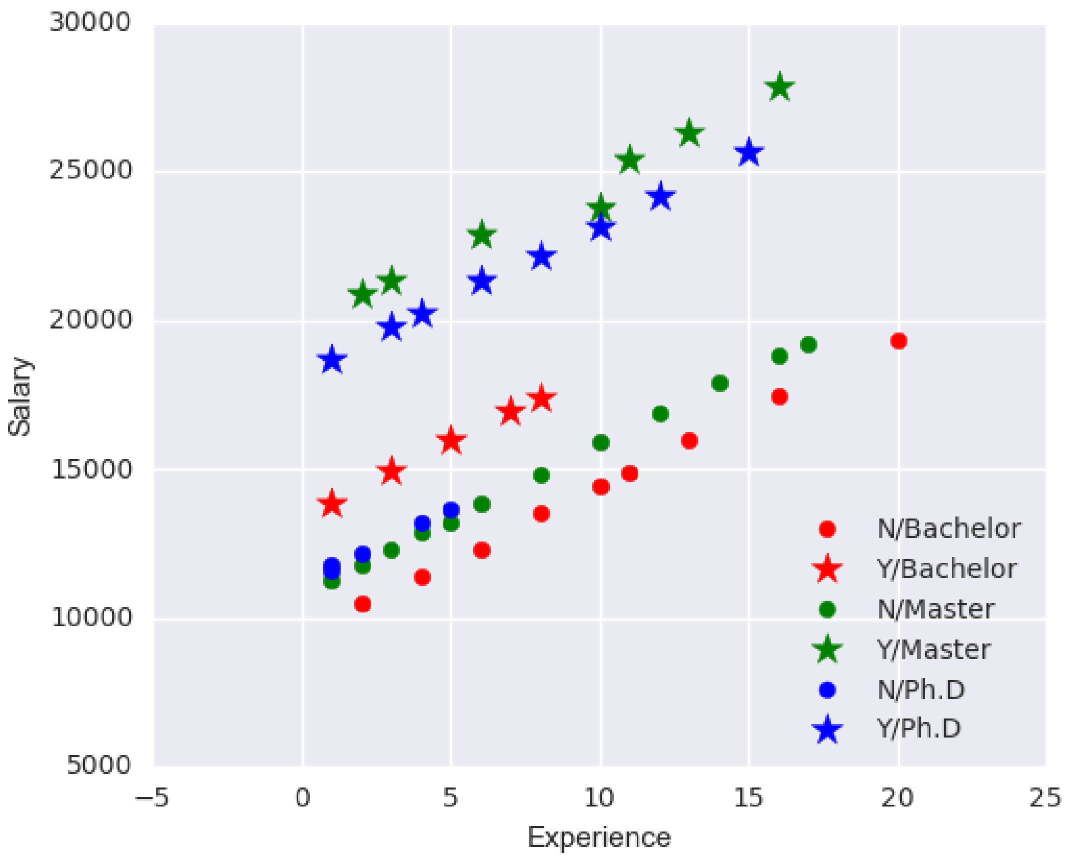
```
<matplotlib.collections.PathCollection at 0x7f78c2ab25f8>
```



Scatter plot with colors and symbols

```
## Figure size
plt.figure(figsize=(6,5))

## Define colors / symbols manually
symbols_manag = dict(Y='*', N='.')
colors_edu = {'Bachelor':'r', 'Master':'g', 'Ph.D':'blue'}
```

```
## group by education x management => 6 groups
for values, d in salary.groupby(['education','management']):
    edu, manager = values
    plt.scatter(d['experience'], d['salary'], marker=symbols_manag[manager],␣
↪color=colors_edu[edu],
                s=150, label=manager+"/"+edu)

## Set labels
plt.xlabel('Experience')
plt.ylabel('Salary')
plt.legend(loc=4)  # lower right
plt.show()
```



Saving Figures

```
### bitmap format
plt.plot(x, sinus)
plt.savefig("sinus.png")
plt.close()

# Prefer vectorial format (SVG: Scalable Vector Graphics) can be edited with
# Inkscape, Adobe Illustrator, Blender, etc.
plt.plot(x, sinus)
plt.savefig("sinus.svg")
plt.close()

# Or pdf
plt.plot(x, sinus)
plt.savefig("sinus.pdf")
plt.close()
```

Exploring data (with seaborn)

Sources: http://stanford.edu/~mwaskom/software/seaborn

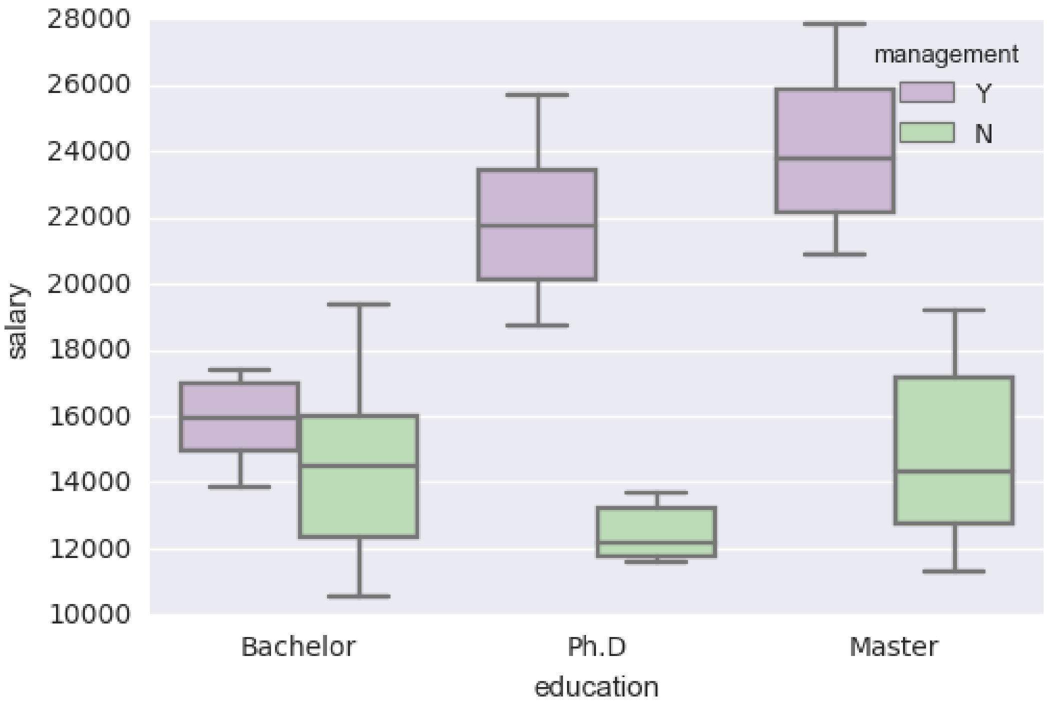
Install using: `pip install -U --user seaborn`

Boxplot

Box plots are non-parametric: they display variation in samples of a statistical population without making any assumptions of the underlying statistical distribution.

```
import seaborn as sns

sns.boxplot(x="education", y="salary", hue="management", data=salary, palette="PRGn")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f78c2ab44a8>
```



```
sns.boxplot(x="management", y="salary", hue="education", data=salary, palette="PRGn")
```

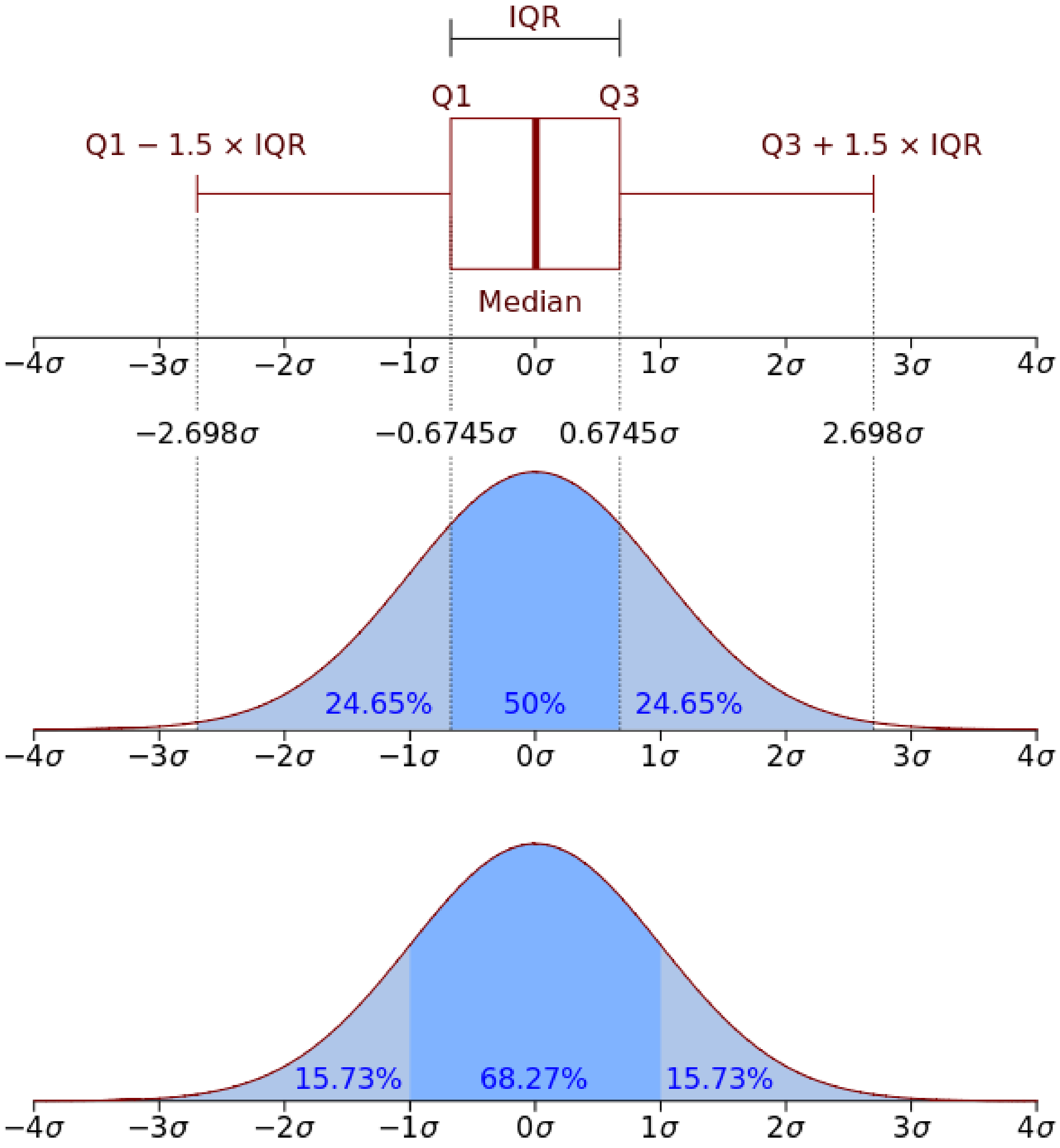
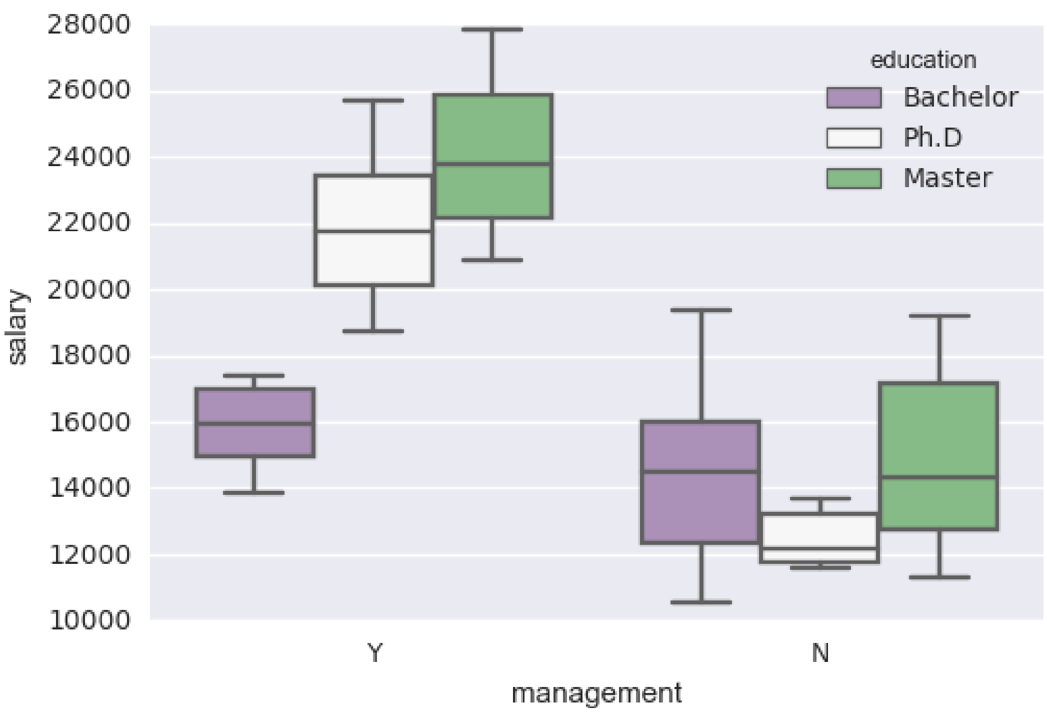

IQR
Q1
Q3
Q1 − 1.5 × IQR
Q3 + 1.5 × IQR
Median
−4σ
−3σ
−2σ
−1σ
0σ
1σ
2σ
3σ
4σ
−2.698σ
−0.6745σ
0.6745σ
2.698σ
24.65%
50%
24.65%
15.73%
68.27%
15.73%

Fig. 5.1: title

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f78c27d9358>
```



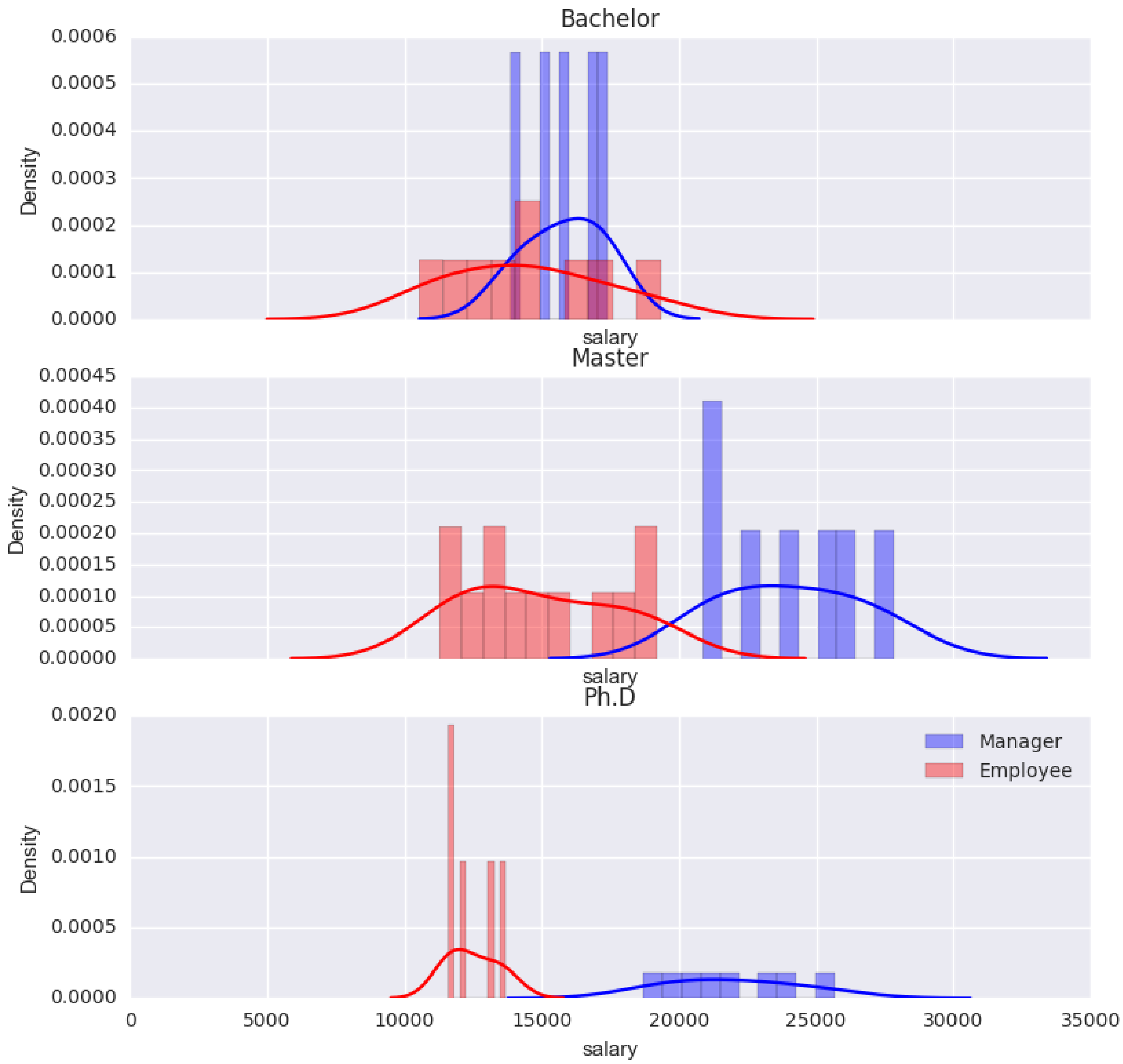
Density plot with one figure containing multiple axis

One figure can contain several axis, whose contain the graphic elements

```
# Set up the matplotlib figure: 3 x 1 axis

f, axes = plt.subplots(3, 1, figsize=(9, 9), sharex=True)

i = 0
for edu, d in salary.groupby(['education']):
    sns.distplot(d.salary[d.management == "Y"], color="b", bins=10, label="Manager",␣
↪ax=axes[i])
    sns.distplot(d.salary[d.management == "N"], color="r", bins=10, label="Employee",␣
↪ax=axes[i])
    axes[i].set_title(edu)
    axes[i].set_ylabel('Density')
    i += 1
plt.legend()
```

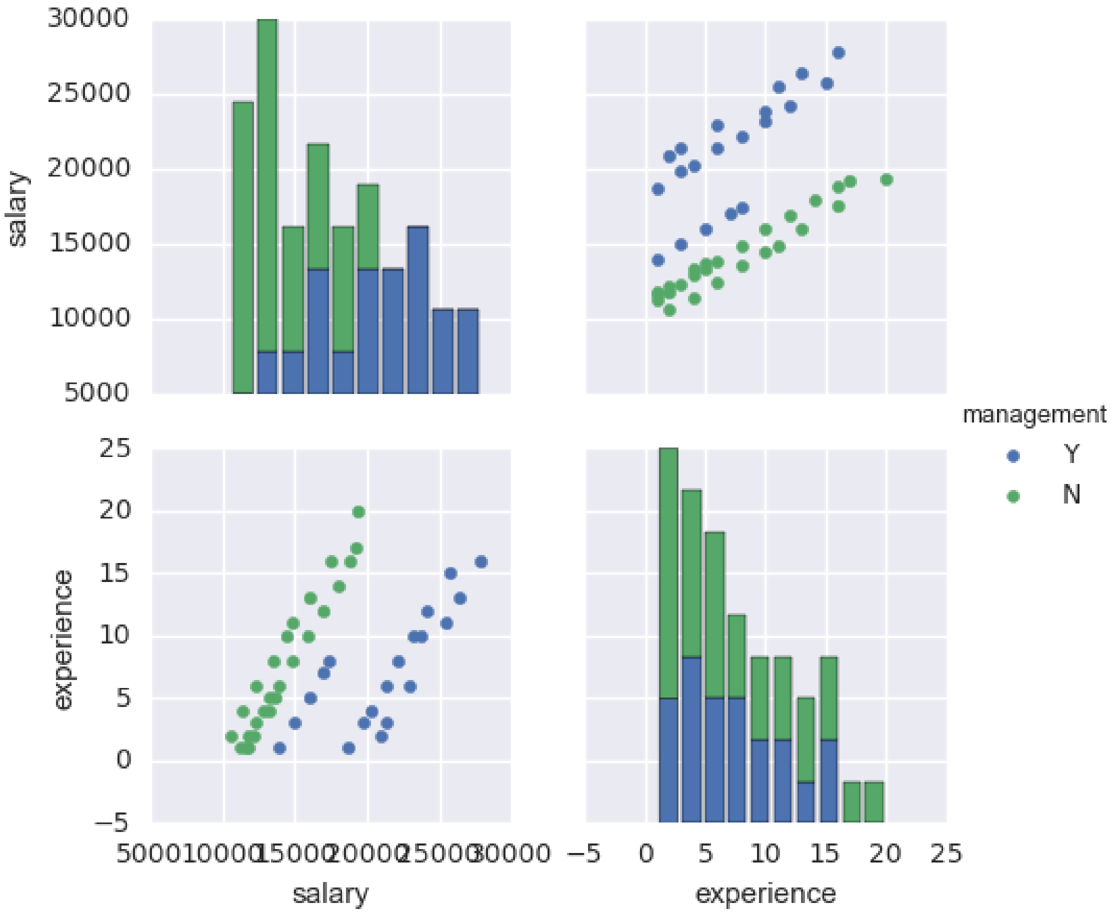
```
/usr/local/anaconda3/lib/python3.5/site-packages/statsmodels/nonparametric/kdetools.
↪py:20: VisibleDeprecationWarning: using a non-integer number instead of an integer␣
↪will result in an error in the future
  y = X[:m/2+1] + np.r_[0,X[m/2+1:],0]*1j
```

```
<matplotlib.legend.Legend at 0x7f78c32ca9b0>
```



```
g = sns.PairGrid(salary, hue="management")
g.map_diag(plt.hist)
g.map_offdiag(plt.scatter)
g.add_legend()
```

```
<seaborn.axisgrid.PairGrid at 0x7f78c2994dd8>
```


30000
25000
20000
15000
10000
5000
salary
25
20
15
10
5
0
−5
experience
5000
10000
15000
20000
25000
30000
salary
−5
0
5
10
15
20
25
experience
management
● Y
● N