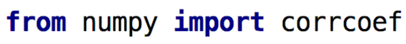
Lab Lecture Notes for Week 6
Calculating correlation and drawing scatter plots

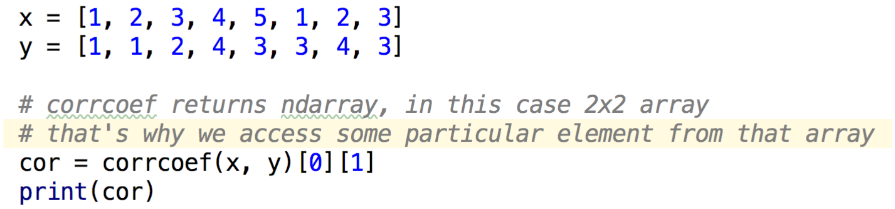
Correlation

For calculating Pearson correlation coefficient we are going to use `corrcoef` function from `numpy`. Let's import it:



```
from numpy import corrcoef
```

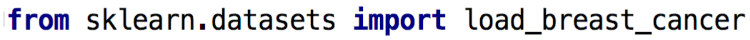
Let's run this function for arbitrary arrays first:



```
x = [1, 2, 3, 4, 5, 1, 2, 3]
y = [1, 1, 2, 4, 3, 3, 4, 3]

# corrcoef returns ndarray, in this case 2x2 array
# that's why we access some particular element from that array
cor = corrcoef(x, y)[0][1]
print(cor)
```

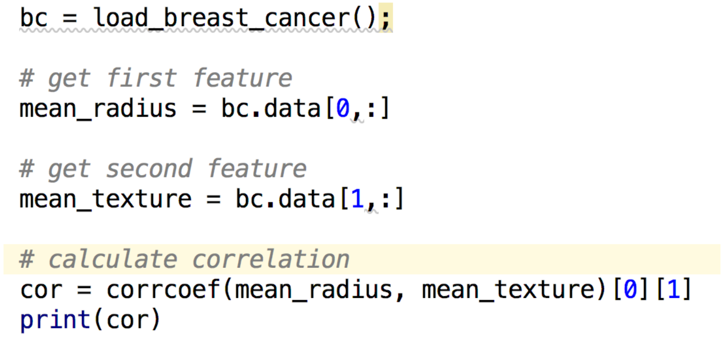
The result of this is `0.416475609064`.

Now let's use a built-in dataset. We are going to use Breast Cancer dataset. Let's import it:



```
from sklearn.datasets import load_breast_cancer
```

We are going to calculate correlation between mean radius and mean texture (first two features):

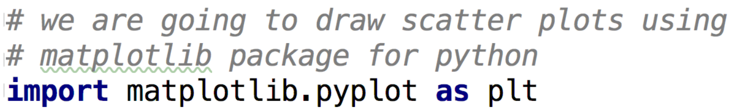


```
bc = load_breast_cancer();

# get first feature
mean_radius = bc.data[0,:]

# get second feature
mean_texture = bc.data[1,:]

# calculate correlation
cor = corrcoef(mean_radius, mean_texture)[0][1]
print(cor)
```

The result of this is `0.989278602681`.

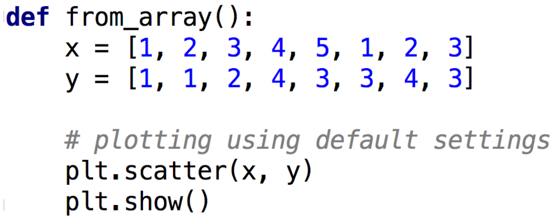
Scatter plots

For drawing scatter plots we are going to use `matplotlib` package. Let's import it:



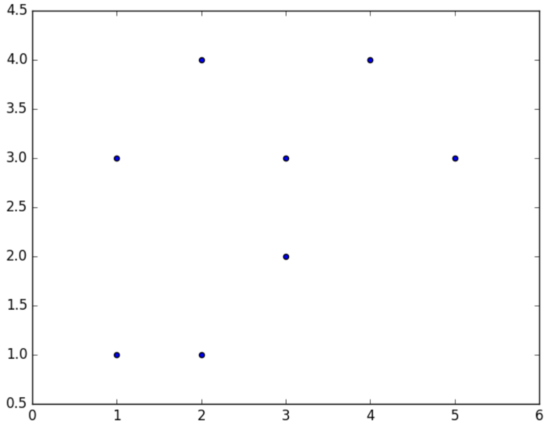
```python
# we are going to draw scatter plots using
# matplotlib package for python
import matplotlib.pyplot as plt
```

Now, let’s plot some arbitrary arrays:

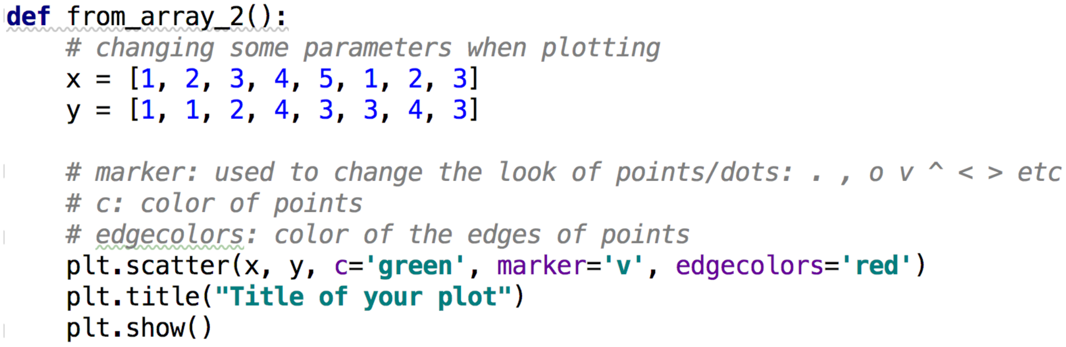


```python
def from_array():
    x = [1, 2, 3, 4, 5, 1, 2, 3]
    y = [1, 1, 2, 4, 3, 3, 4, 3]

    # plotting using default settings
    plt.scatter(x, y)
    plt.show()
```

The result of this is:

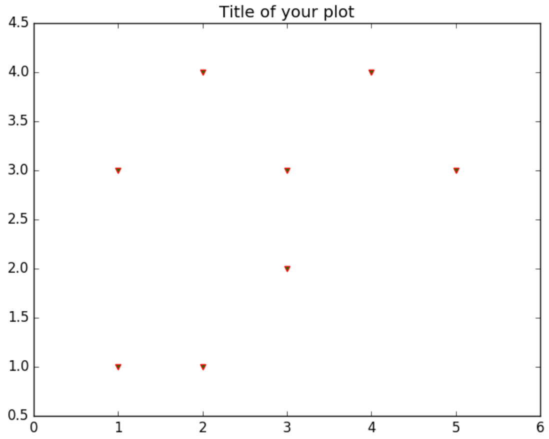


Now let’s change some parameters of scatter() function:

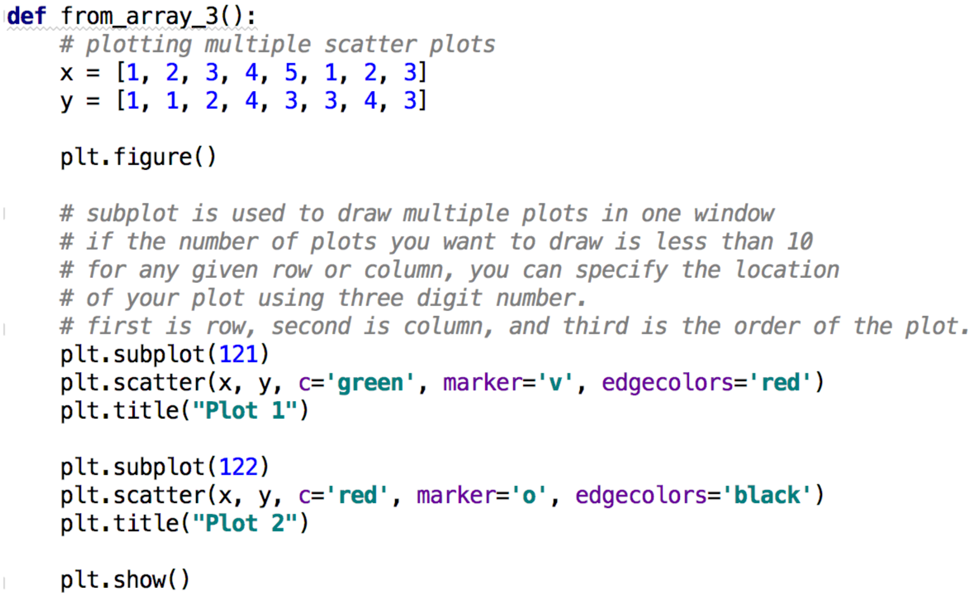


```python
def from_array_2():
    # changing some parameters when plotting
    x = [1, 2, 3, 4, 5, 1, 2, 3]
    y = [1, 1, 2, 4, 3, 3, 4, 3]

    # marker: used to change the look of points/dots: . , o v ^ < > etc
    # c: color of points
    # edgecolors: color of the edges of points
    plt.scatter(x, y, c='green', marker='v', edgecolors='red')
    plt.title("Title of your plot")
    plt.show()
```

The result of this is:

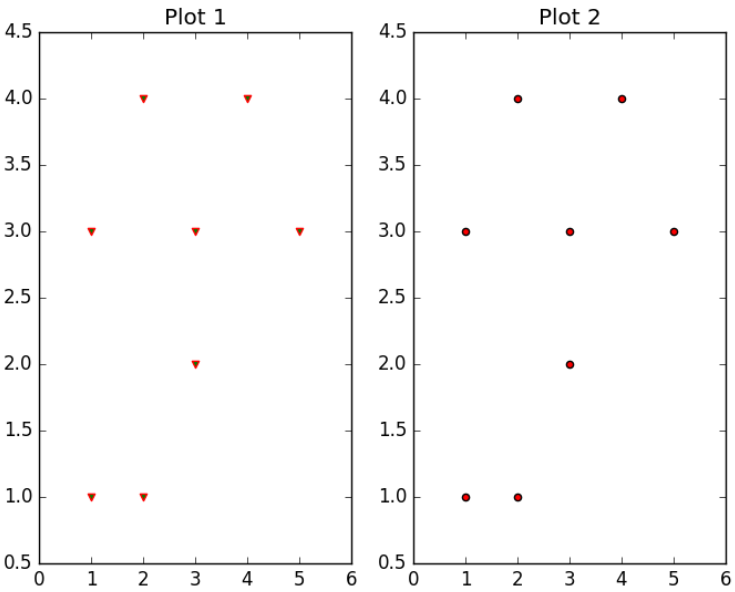


Moreover, you can draw multiple plots at the same time:

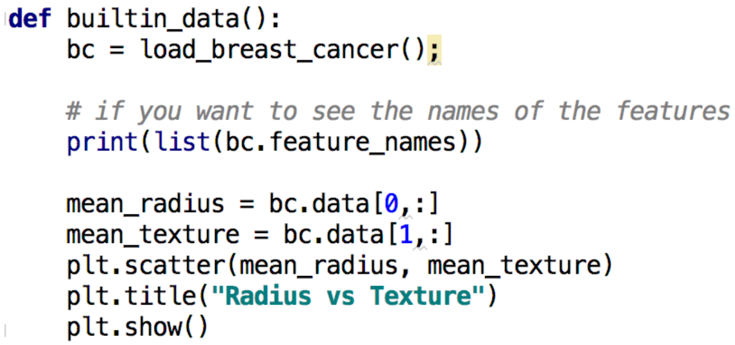


```
def from_array_3():
    # plotting multiple scatter plots
    x = [1, 2, 3, 4, 5, 1, 2, 3]
    y = [1, 1, 2, 4, 3, 3, 4, 3]

    plt.figure()

    # subplot is used to draw multiple plots in one window
    # if the number of plots you want to draw is less than 10
    # for any given row or column, you can specify the location
    # of your plot using three digit number.
    # first is row, second is column, and third is the order of the plot.
    plt.subplot(121)
    plt.scatter(x, y, c='green', marker='v', edgecolors='red')
    plt.title("Plot 1")

    plt.subplot(122)
    plt.scatter(x, y, c='red', marker='o', edgecolors='black')
    plt.title("Plot 2")

    plt.show()
```

Result of this is:

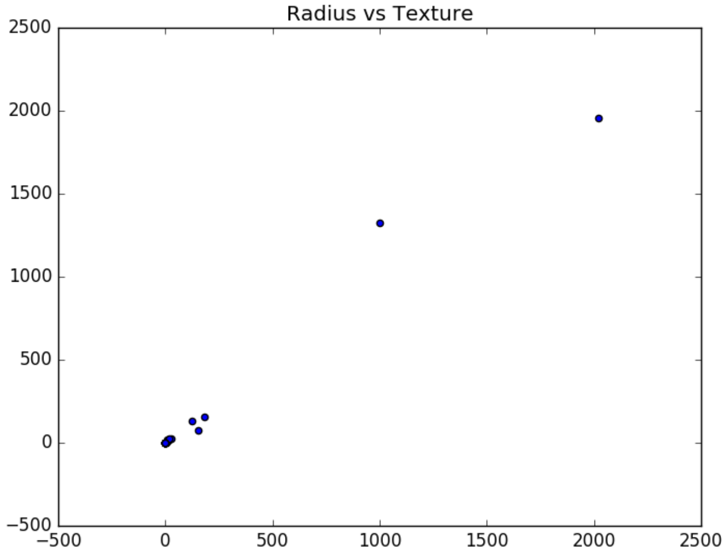


Now let’s use breast cancer data, and draw scatter plot of mean radius versus mean texture:



```
def builtin_data():
    bc = load_breast_cancer();

    # if you want to see the names of the features
    print(list(bc.feature_names))

    mean_radius = bc.data[0,:]
    mean_texture = bc.data[1,:]
    plt.scatter(mean_radius, mean_texture)
    plt.title("Radius vs Texture")
    plt.show()
```

The result is below:



We can see that there are two clear outliers in the data.

NOTE: Version of programs and packages used
Python: 3.5.2
Numpy: 1.11.1
Matplotlib: 1.5.3
Sklearn: 0.17.1