Welcome 2 Roboschool 2015 @Volgograd State Technical University

IPython Notebook

download @http://ipython.org/notebook.html (http://ipython.org/notebook.html)

This is

- very fun
- comfort
- intuitive

Basic Operations

In [7]:
```
a = 5
b = 8
c = a+b
print(c)


z = [1,2,3,4,5,6]

print(min(z))
```

```
13
1
```

API & Class implementation

In [3]:
```
class my_super_class:

    _some_parameter = None

    def __init__(self, some_parameter_):
        self._some_parameter = some_parameter_

    def get_some_parameter(self):
        return self._some_parameter
```

In [8]:
```
msc = my_super_class('Hello')

msc.get_some_parameter()
```

Out[8]:
```
'Hello'
```

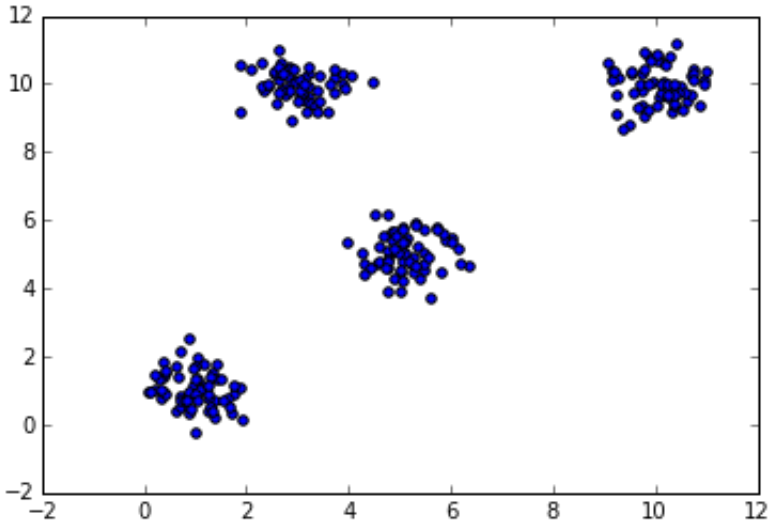
Data Science: CRISP - DM

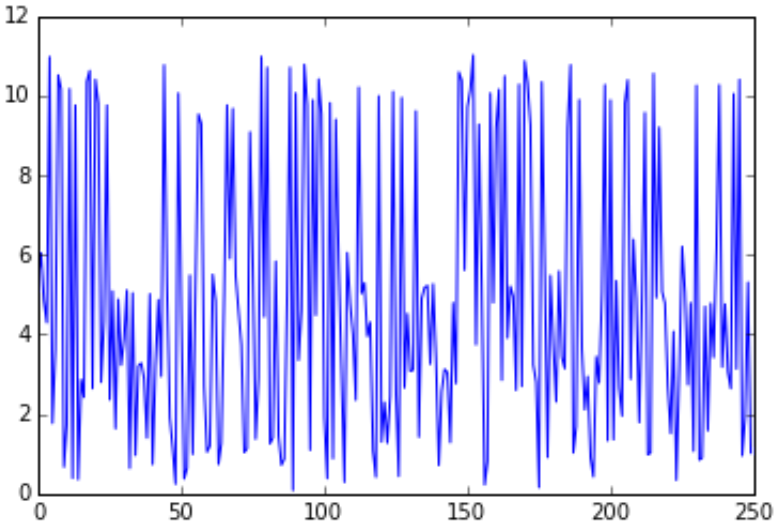
* *

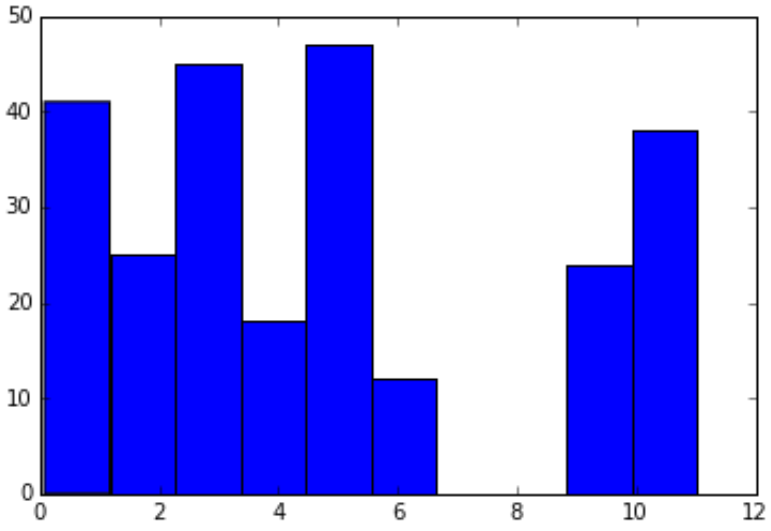
In [11]:
```
%matplotlib inline
import time
import numpy as np
import pandas as pd
from sklearn.cluster import MeanShift, estimate_bandwidth
from sklearn.datasets.samples_generator import make_blobs
import matplotlib.pyplot as plt
```

In [112]:
```
n_samples_ = 250
cluster_std_=0.5

colors = 100*['r.','g.','m.','c.','k.','y.','b.']
centers = [[1,1],[5,5],[3,10], [10,10]]
# centers = [[1,1],[5,5],[3,10], [10,10], [100,100]

X, _ = make_blobs(n_samples=n_samples_, centers=centers, cluster_std=clu
ster_std_)

# plot data
plt.figure()
plt.scatter(X[:,0], X[:,1])

plt.figure()
plt.plot(X[:,0])

plt.figure()
plt.hist(X[:,0])
```

```
Out[112]: (array([ 41.,  25.,  45.,  18.,  47.,  12.,   0.,   0.,  24.,  38.]),
           array([  0.08094375,   1.1743838 ,   2.26782385,   3.3612639 ,
                    4.45470395,   5.54814401,   6.64158406,   7.73502411,
                    8.82846416,   9.92190421,  11.01534427]),
           <a list of 10 Patch objects>)
```







Variations:

- add very far class

KMeans.

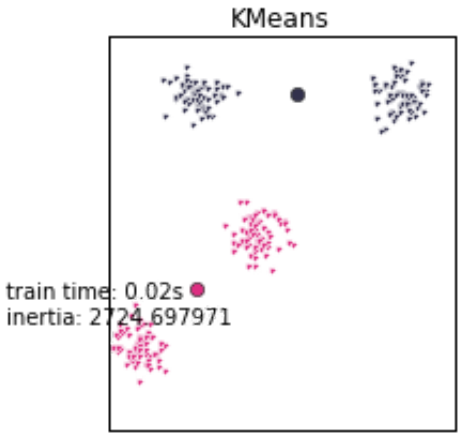
Explain using whiteboard

In [117]:
```
from sklearn.cluster import KMeans

n_clusters_ = 2#len(centers)

k_means = KMeans(init='k-means++', n_clusters=n_clusters_, n_init=10)

t0 = time.time()
k_means.fit(X)
t_batch = time.time() - t0

k_means_labels = k_means.labels_
k_means_cluster_centers = k_means.cluster_centers_
k_means_labels_unique = np.unique(k_means_labels)

print (t_batch)
```

```
0.02101278305053711
```

In [118]:
```
import random

def design_color():
    color = "#%06x" % random.randint(0, 0xFFFFFF)
    return color


fig = plt.figure(figsize=(8, 3))
fig.subplots_adjust(left=0.02, right=0.98, bottom=0.05, top=0.9)

colors_clusters = []
for i in range(n_clusters_):
    colors_clusters[len(colors_clusters):] = [design_color()]


# KMeans
ax = fig.add_subplot(1, 3, 1)
for k, col in zip(range(n_clusters_), colors_clusters):
    my_members = k_means_labels == k
    cluster_center = k_means_cluster_centers[k]
    ax.plot(X[my_members, 0], X[my_members, 1], 'w',
            markerfacecolor=col, marker='.')
    ax.plot(cluster_center[0], cluster_center[1], 'o', markerfacecolor=c
ol,
            markeredgecolor='k', markersize=6)
ax.set_title('KMeans')
ax.set_xticks(())
ax.set_yticks(())
plt.text(-3.5, 1.8,  'train time: %.2fs\ninertia: %f' % (
    t_batch, k_means.inertia_))


print (k_means_cluster_centers)
```

```
[[ 6.54702065  9.94365145]
 [ 3.06410974  3.01424752]]
```

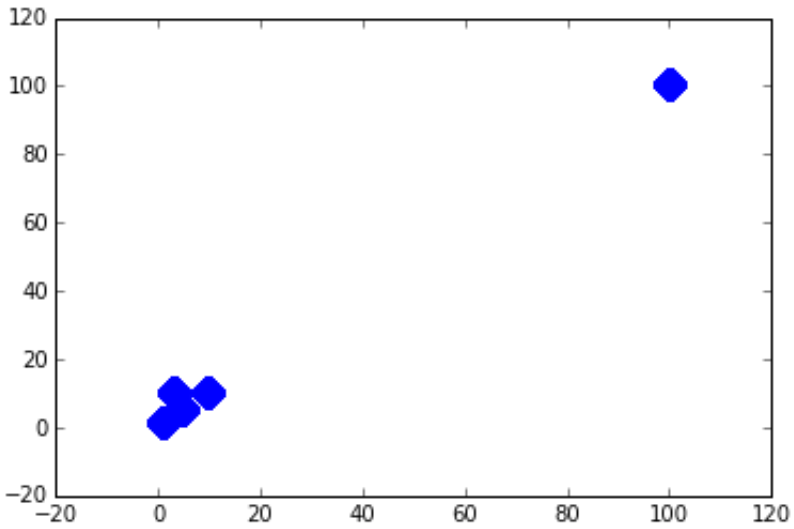


MeanShift

In [109]:
```
bandwidth = estimate_bandwidth(X, quantile=.2, n_samples=250)


bin_seeding_ = True
t0 = time.time()
cluster_instance = MeanShift(bandwidth=bandwidth, bin_seeding=bin_seedin
g_)
cluster_instance.fit(X)
labels = cluster_instance.labels_
cluster_centers = cluster_instance.cluster_centers_
n_cluster_ = len(np.unique(labels))
fit_time = time.time() - t0

print(cluster_centers)
```

```
[[   0.95442028    1.02027664]
 [   5.0050787     4.95717385]
 [ 100.08833359  100.01504268]
 [   3.0193204     9.95213805]
 [  10.00072579    9.94241852]]
```

In [120]:
```
for i in range(len(X)):
    plt.plot(X[i][0],X[i][1], colors[labels[i]], markersize = 3)
    plt.scatter(cluster_centers[:,0], cluster_centers[:,1],marker = "x",
s = 75, linewidths=10, zorder = 10)
plt.show()
```



You need

http://localhost:8888/notebooks/Documents/IPython%20Notebooks/A%20Method%20Automation%20F
(http://localhost:8888/notebooks/Documents/IPython%20Notebooks/A%20Method%20Automation%20

http://localhost:8888/notebooks/Strategway%20--%20Clustering.ipynb