

# Predict Subs on Open Orders: KMEANS Algorithm, Aug. 2016

## loadData

In [27]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from matplotlib import style
style.use('ggplot')
from sklearn.cluster import KMeans

x = [136, 112, 108, 108, 105, 104, 104, 100, 91, 90, 90, 88, 87, 87, 86, 85, 80, 80, 80, 80, 80, 78, 78, 78,
77, 76, 76, 75, 75,
74]
y = [9400, 6800, 9000, 6300, 8900, 6800, 8000, 6700, 6800, 9200, 6600, 6000, 6600, 6800, 6800, 6700, 6800, 6
200, 6000, 9200,
8900, 9500, 6800, 6800, 6800, 8900, 8900, 9200, 6000, 6600]
```

## cleanData

In [28]:

```
X = np.array([[136, 9400],
               [112, 6800],
               [108, 9000],
               [108, 6300],
               [105, 8900],
               [104, 6800],
               [104, 8000],
               [100, 6700],
               [91, 6800],
               [90, 9200],
               [90, 6600],
               [88, 6000],
               [87, 6600],
               [87, 6800],
               [86, 6800],
               [85, 6700],
               [80, 6800],
               [80, 6200],
               [80, 6000],
               [80, 9200],
               [80, 8900],
               [78, 9500],
               [78, 6800],
               [78, 6800],
               [77, 6800],
               [76, 8900],
               [76, 8900],
               [75, 9200],
               [75, 6000],
               [74, 6600]])
```

## buildModel

In [29]:

```
kmeans = KMeans(n_clusters = 3)
kmeans.fit(X)

centroids = kmeans.cluster_centers_
labels = kmeans.labels_

print(centroids)
print(labels)
```

```
[[ 86.2      6100.      ]
 [ 91.63636364 9009.090909]
 [ 87.78571429 6742.85714286]]
[1 2 1 0 1 2 1 2 2 1 2 0 2 2 2 2 2 0 0 1 1 1 2 2 2 1 1 1 0 2]
```

## showCentroids

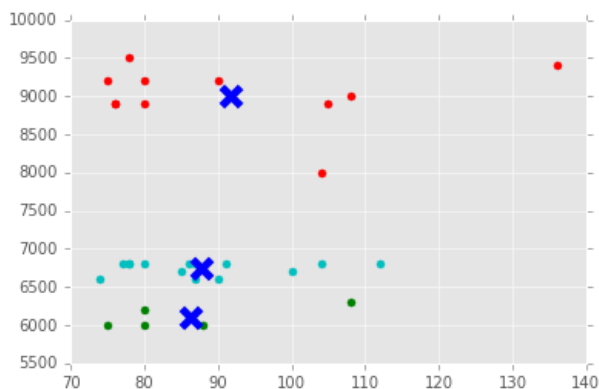
In [31]:

```
colors = ['g.', 'r.', 'c.']

for i in range(len(X)):
    print('coordinate:', X[i], 'label:', labels[i])
    plt.plot(X[i][0], X[i][1], colors[labels[i]], markersize = 10)

plt.scatter(centroids[:, 0], centroids[:, 1], marker = 'x', s = 150, linewidths = 5, zorder = 10)
plt.show()
```

```
('coordinate:', array([ 136, 9400]), 'label:', 1)
('coordinate:', array([ 112, 6800]), 'label:', 2)
('coordinate:', array([ 108, 9000]), 'label:', 1)
('coordinate:', array([ 108, 6300]), 'label:', 0)
('coordinate:', array([ 105, 8900]), 'label:', 1)
('coordinate:', array([ 104, 6800]), 'label:', 2)
('coordinate:', array([ 104, 8000]), 'label:', 1)
('coordinate:', array([ 100, 6700]), 'label:', 2)
('coordinate:', array([ 91, 6800]), 'label:', 2)
('coordinate:', array([ 90, 9200]), 'label:', 1)
('coordinate:', array([ 90, 6600]), 'label:', 2)
('coordinate:', array([ 88, 6000]), 'label:', 0)
('coordinate:', array([ 87, 6600]), 'label:', 2)
('coordinate:', array([ 87, 6800]), 'label:', 2)
('coordinate:', array([ 86, 6800]), 'label:', 2)
('coordinate:', array([ 85, 6700]), 'label:', 2)
('coordinate:', array([ 80, 6800]), 'label:', 2)
('coordinate:', array([ 80, 6200]), 'label:', 0)
('coordinate:', array([ 80, 6000]), 'label:', 0)
('coordinate:', array([ 80, 9200]), 'label:', 1)
('coordinate:', array([ 80, 8900]), 'label:', 1)
('coordinate:', array([ 78, 9500]), 'label:', 1)
('coordinate:', array([ 78, 6800]), 'label:', 2)
('coordinate:', array([ 78, 6800]), 'label:', 2)
('coordinate:', array([ 77, 6800]), 'label:', 2)
('coordinate:', array([ 76, 8900]), 'label:', 1)
('coordinate:', array([ 76, 8900]), 'label:', 1)
('coordinate:', array([ 75, 9200]), 'label:', 1)
('coordinate:', array([ 75, 6000]), 'label:', 0)
('coordinate:', array([ 74, 6600]), 'label:', 2)
```



## writeUp

This is a K-Means algorithm that groups a set of objects into subsets such that all elements within a group are more similar among them than they are to the others. Subscribers with higher total order count values are targeted first for marketing irregardless of principal location.