# K-Means Clustering Algorithm Example-3

ယနေ့ခေတ်သည် ကုန်ပစ္စည်းများကို အပြိုင်အဆိုင် ရောင်းချသည့် ခေတ်ဖြစ်သည်။ အရောင်းမြှင့်တင်ခြင်း(sale promotion) ၊ ဈေးကွက်ထိုးဖောက်ခြင်း(marketing) ၊ ကုန်ပစ္စည်းထုတ်လုပ်သည့် နည်းဗျူဟာများ(product development strategies) ရှာဖွေခြင်းတို့ ပြုလုပ်သည့်အခါ AI နည်းပညာများကို အသုံးပြုလာကြသည်။ ထို့ကြောင့် စားသုံးသူများ၏ အလေ့အကျင့် စရိုက်(customer behavior) ၊ ဈေးဝယ်သည့် အလေ့အကျင့် စရိုက်(buying behavior) စသည်တို့ကို လေ့လာပြီး အသုံးချကြသည်။

ယခု ဥပမာတွင် e-commerce site တစ်ခု အတွက် K-Means Clustering Algorithm ဖြင့် customer segmentation လုပ်သည့်နည်းကို ဖော်ပြ ရှင်းလင်းထားသည်။

မိသားစု (၃၀၀) သို့မဟုတ် အိမ်ထောင်ဦးစီး (၃၀၀)ယောက်၏ နှစ်စဉ ဝင်ငွေ(annual income,in USdollar 000) နှင့် နှစ်စဉ အသုံးစရိတ် (annual spend, in USdollar 000)ကို မှတ်တမ်းတင်ထားသည့် dataset ကို အခြေခံ၍ optimum number of clusters ရှာဖွေပုံကို ဖော်ပြထားသည်။

numpy နှင့် pandas libraries များကို သုံး၍ python ဖြင့် ရေးထားသည့် code များကို အတတ်နိုင်ဆုံး သေးစိတ် ရှင်းပြသွားမည်။

### Load the required packages

Matrix များကို တွက်ရန် numpy နှင့် pandas ကို import လုပ်သည်။
ဂရပ်ပုံများဆွဲတွက်ရန် matplotlib ကို import လုပ်သည်။

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings('ignore')#သတိပေးချက်များကို မဖော်ပြရန်

#Plot styling
import seaborn as sns; sns.set()  # for plot styling
%matplotlib inline

plt.rcParams['figure.figsize'] = (16, 9) # ဂရပ်အရွယ်အစား သတ်မှတ်သည်။
plt.style.use('ggplot')  # အသုံးပြုလိုသည့် စတိုင် သတ်မှတ်သည်။
```

### Dataset ကို ဖတ်သည်။

In [2]:
```
dataset=pd.read_csv('CLV.csv') #http://www.acmv.org/MachineLearning/kMeans/kMeans_Eg3/CLV.csv
```

In [3]:
```
dataset.head()  # dataset ထဲမှ ပထမဆုံး 5 rows ကို ကြည့်သည်။
```

Out[3]:

| | INCOME | SPEND |
|---|---|---|
| **0** | 233 | 150 |
| **1** | 250 | 187 |
| **2** | 204 | 172 |
| **3** | 236 | 178 |
| **4** | 354 | 163 |

In [4]:
```
len(dataset) # dataset ထဲတွင် row မည်မျှရှိသည်ကို စစ်ဆေးသည်။
```

Out[4]: 303

Dataset ထဲမှ statistical အချက်အလက်များကို ကြည့်သည်။ အလျားလိုက်မြင် ရရန် .transpose() လုပ်သည်။

In [5]:
```
dataset.describe().transpose() #descriptive statistics of the dataset
```

Out[5]:

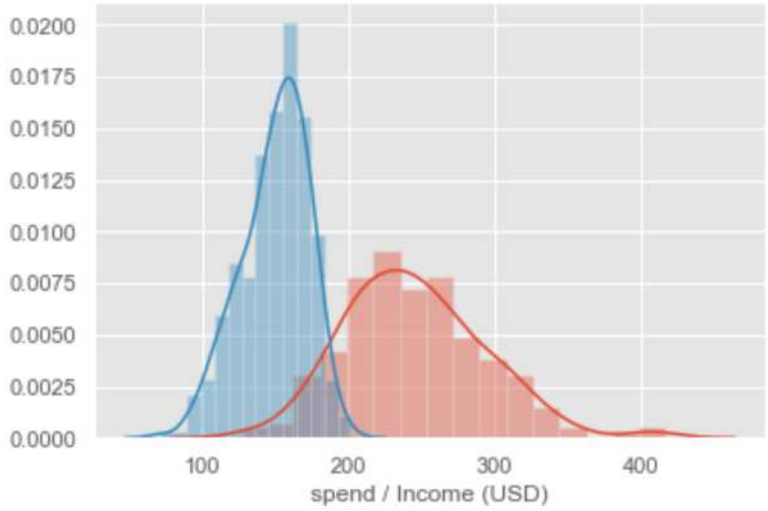
| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **INCOME** | 303.0 | 245.273927 | 48.499412 | 126.0 | 211.0 | 240.0 | 274.0 | 417.0 |
| **SPEND** | 303.0 | 149.646865 | 22.905161 | 71.0 | 133.5 | 153.0 | 166.0 | 202.0 |

Dataset ထဲတွင် 303 row ရှိသည်။ ပျမ်းမျှ နှစ်စဉ ဝင်ငွေမှာ (mean annual income ) ဒေါ်လာ 245000 ဖြစ်သည်။ ပျမ်းမျှ နှစ်စဉ အသုံးစရိတ်မှာ (mean annual spend ) ဒေါ်လာ 149000 ဖြစ်သည်။ Dataset ထဲတွင် 303 row ရှိသည်။ ပျမ်းမျှ နှစ်စဉ ဝင်ငွေမှာ (mean annual income ) ဒေါ်လာ 245000 ဖြစ်သည်။ ပျမ်းမျှ နှစ်စဉ အသုံးစရိတ်မှာ (mean annual spend ) ဒေါ်လာ 149000 ဖြစ်သည်။

distplot နှင့် violinplot တို့ ဖြင့် ဂရပ်ပုံများ ဆွဲ၍ data visualization လုပ်သည်။

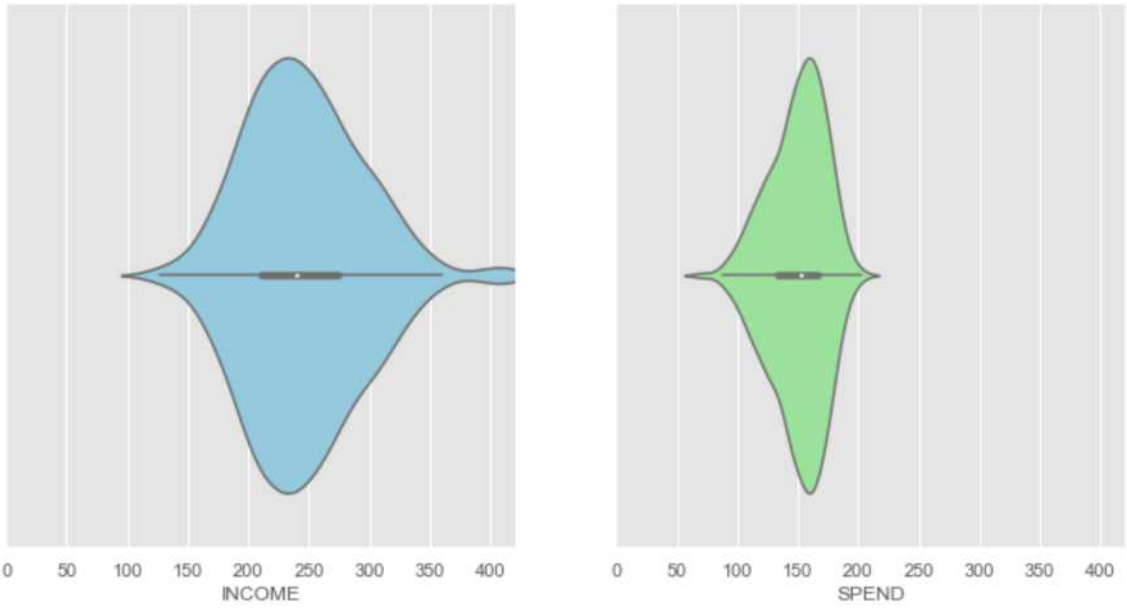
In [6]:
```
#Visualising the data
plot_income = sns.distplot(dataset["INCOME"])
plot_spend = sns.distplot(dataset["SPEND"])
plt.xlabel('spend / Income (USD)')
```

Out[6]: Text(0.5, 0, 'spend / Income (USD)')




In [7]:
```
#Violin plot
f, axes = plt.subplots(1,2, figsize=(12,6), sharex=True, sharey=True)
v1 = sns.violinplot(data=dataset, x='INCOME', color="skyblue",ax=axes[0])
v2 = sns.violinplot(data=dataset, x='SPEND',color="lightgreen", ax=axes[1])
v1.set(xlim=(0,420))
```

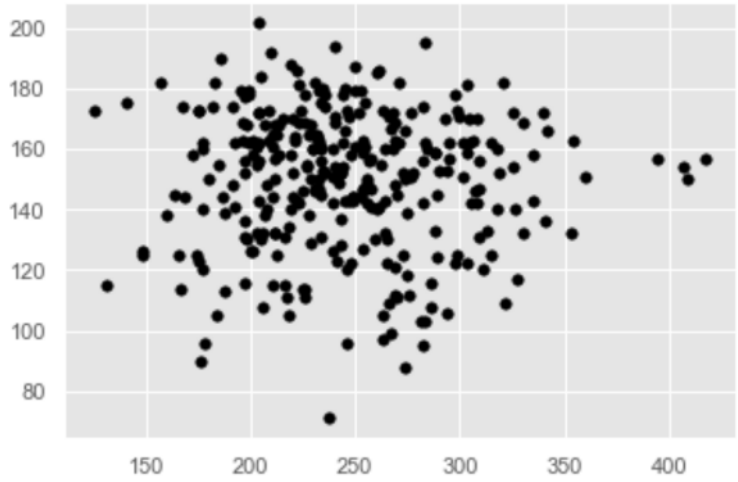
Out[7]: [(0, 420)]




Income ပမာဏကို X ဝင်ရိုးတွင်လည်းကောင်း Spend ပမာဏကို Y ဝင်ရိုးတွင်လည်းကောင်းထား၍ scatter ဂရပ် ဆွဲသည်။

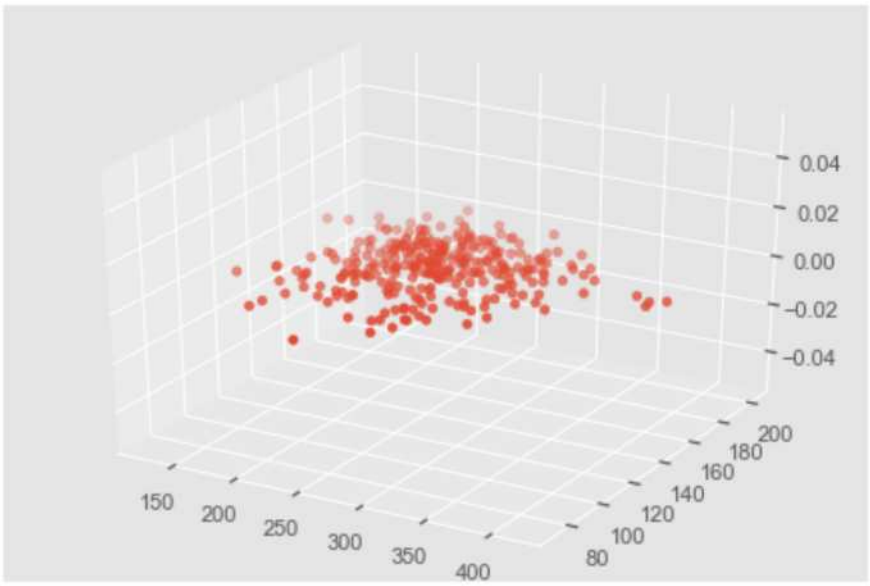
In [8]:
```
# Plotting the values to understand the spread
Income = dataset['INCOME'].values
Spend = dataset['SPEND'].values
X = np.array(list(zip(Income, Spend)))
plt.scatter(Income, Spend, c='black', s=30)
```

Out[8]: <matplotlib.collections.PathCollection at 0x62dfde4588>




In [9]:
```
# Plot in 3D space
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(X[:, 0], X[:, 1])
```

Out[9]: <mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x62dfe88d68>



### Clustering fundamentals

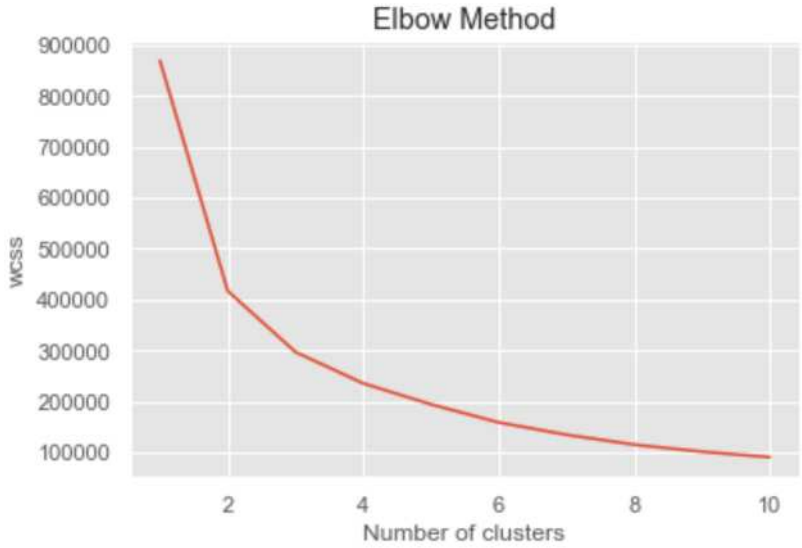
k အရေအတွက် သည် clustering လုပ်လိုသည့် အစု အရေအတွက် ဖြစ်သည်။ k အရေအတွက်ကို ဆုံးဖြတ်ခြင်းသည် ရိုးရှင်းလွယ်ကူသည့် ကိစ္စ တစ်ခု မဟုတ်ပါ။ k အရေအတွက် မည်မျှဖြစ်မည်ကို အလွယ်တကူ ဆုံးဖြတ်ရန် ခက်ခဲသောကြောင့် k အရေအတွက် ၂ မှ ၁၀ အထိကို loop ပတ်၍ တွက်သည်။

အကောင်းဆုံး K အရေအတွက်ကို ရွေးရန်(to choose the optimal K) elbow method ကို အသုံးပြုနိုင်သည်။ elbow method ဆိုသည်မှာ အောက်ပုံတွင် ဂရပ်လိုင်းသည် လက်မောင်း နှင့် တံတောင်ဆစ်ကဲ့သို့ ကွေးဆင်း သွားသကဲ့သို့ ဖြစ်နေသည်။ တံတောင်ဆစ် ကွေးနေရာသည် အကောင်းဆုံး K အရေအတွက်ကို ဖော်ပြပေးသည့် နေရာ ဖြစ်သည်ဟု ဆိုလိုသည်။ ထိုနေရာကို "elbow point" ဟုခေါ်ဆိုသည်။

"Elbow point," မတိုင်ခင်နေရာသည် K အရေအတွက်သည်နှင့်အမျှ claster ဖွဲ့စည်းမှု ပိုကျစ်လစ်လာသည် ။ သို့သော် "elbow point," ကျော်သွားသည့်အခါ K အရေအတွက် ပိုများအောင် လုပ်သော်လည်း claster ဖွဲ့စည်းမှု မဆိုစလောက်သာ ကောင်းလာသောကြောင့် ဖြစ်သည်။

In [10]:
```
X=dataset.iloc[:,[0,1]].values
```

In [11]:
```
#Using the elbow method to find the ideal number of clusters
from sklearn.cluster import KMeans
wcss = []
for i in range(1,11):
    km=KMeans(n_clusters=i,init='k-means++', max_iter=300, n_init=10, random_state=0)
    km.fit(X)
    wcss.append(km.inertia_)
plt.plot(range(1,11),wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('wcss')
plt.show()
```



### Calculating the Silhoutte Coefficient

Silhouette value သို့မဟုတ် silhoutte coefficient သည် cluster တစ်ခုမှ object တစ်ခုသည့် တခြားသော cluster မှ object များနှင့် မည်မျှဆင်တူသည်ကို ဖော်ပြသည်။ (The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). )

Silhouette value သို့မဟုတ် silhoutte coefficient တန်ဖိုးမှ −1 မှ +1 အတွင်း ဖြစ်သည်။ တန်းဖိုးမြင့်လေ မိမိ cluster မှ အဖွဲ့များနှင့် တူညီ(similar)ပြီး တခြား cluster မှ အဖွဲ့များနှင့် ကွဲပြားလေ ဖြစ်သည်။(The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.)

silhoutte coefficient တန်ဖိုးမြင့်လျှင် clustering လုပ်ခြင်း မှန်ကန်သည်ဟုဆိုနိုင်သည်။

In [12]:
```
from sklearn.metrics import silhouette_score
from sklearn.cluster import KMeans

for n_cluster in range(2, 11):
    kmeans = KMeans(n_clusters=n_cluster).fit(X)
    label = kmeans.labels_
    sil_coeff = silhouette_score(X, label, metric='euclidean')
    print("For n_clusters={}, The Silhouette Coefficient is {}".format(n_cluster, sil_coeff))
```

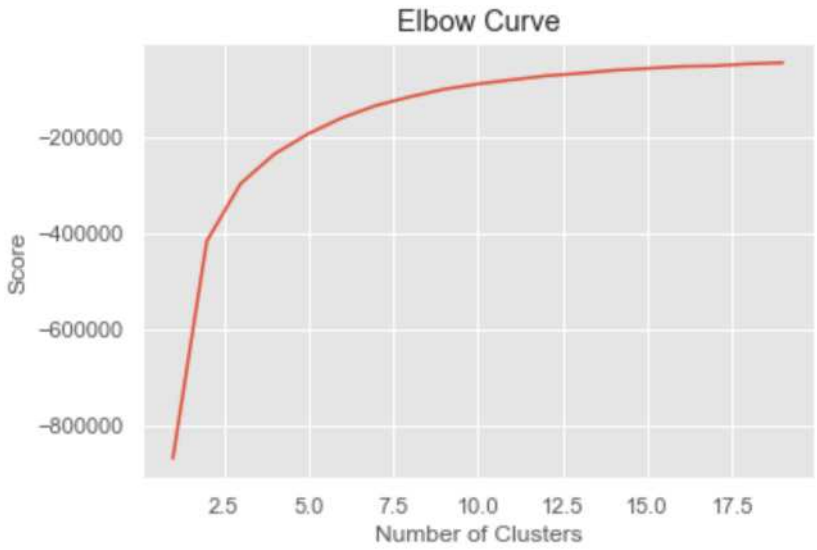
```
For n_clusters=2, The Silhouette Coefficient is 0.44006694211403197
For n_clusters=3, The Silhouette Coefficient is 0.35962629048722355
For n_clusters=4, The Silhouette Coefficient is 0.35271446789203426
For n_clusters=5, The Silhouette Coefficient is 0.3599559651419018
For n_clusters=6, The Silhouette Coefficient is 0.36926344785902604
For n_clusters=7, The Silhouette Coefficient is 0.35766486276444615
For n_clusters=8, The Silhouette Coefficient is 0.3603118868409036
For n_clusters=9, The Silhouette Coefficient is 0.34239615698190273
For n_clusters=10, The Silhouette Coefficient is 0.35582331950254165
```

### Elbow Curve

In [13]:
```
import pylab as pl
from sklearn.decomposition import PCA

Nc = range(1, 20)
kmeans = [KMeans(n_clusters=i) for i in Nc]
kmeans
score = [kmeans[i].fit(X).score(X) for i in range(len(kmeans))]
score
pl.plot(Nc,score)
pl.xlabel('Number of Clusters')
pl.ylabel('Score')
pl.title('Elbow Curve')
pl.show()

print(score)
```




[-868805.478547855, -416914.67764462164, -297098.0103782924, -235435.61549707665, -193333.51167275754, -159962.83332028287, -134591.5198951492, -116451.45762093064, -100848.46839767715, -89879.29129370357, -81345.96663614169, -73007.51368515502, -67773.06437686866, -61501.28785502304, -57438.70698604744, -53464.5695792062, -52137.05447260316, -47978.358200691 45, -45656.69334312072]

**interia**

In [14]:
```
for k in range (1, 11):
    kmeans_model = KMeans(n_clusters=k, random_state=1).fit(X)
    labels = kmeans_model.labels_
    interia = kmeans_model.inertia_
    print ("k:",k, " cost:", interia)
print()
```

```
k: 1  cost: 868805.4785478548
k: 2  cost: 416914.67764462065
k: 3  cost: 297101.3764201943
k: 4  cost: 235568.75630353513
k: 5  cost: 193333.51167275637
k: 6  cost: 158999.20745160058
k: 7  cost: 135314.10167803388
k: 8  cost: 115686.57935998778
k: 9  cost: 102546.93727877043
k: 10  cost: 91556.28599714936
```

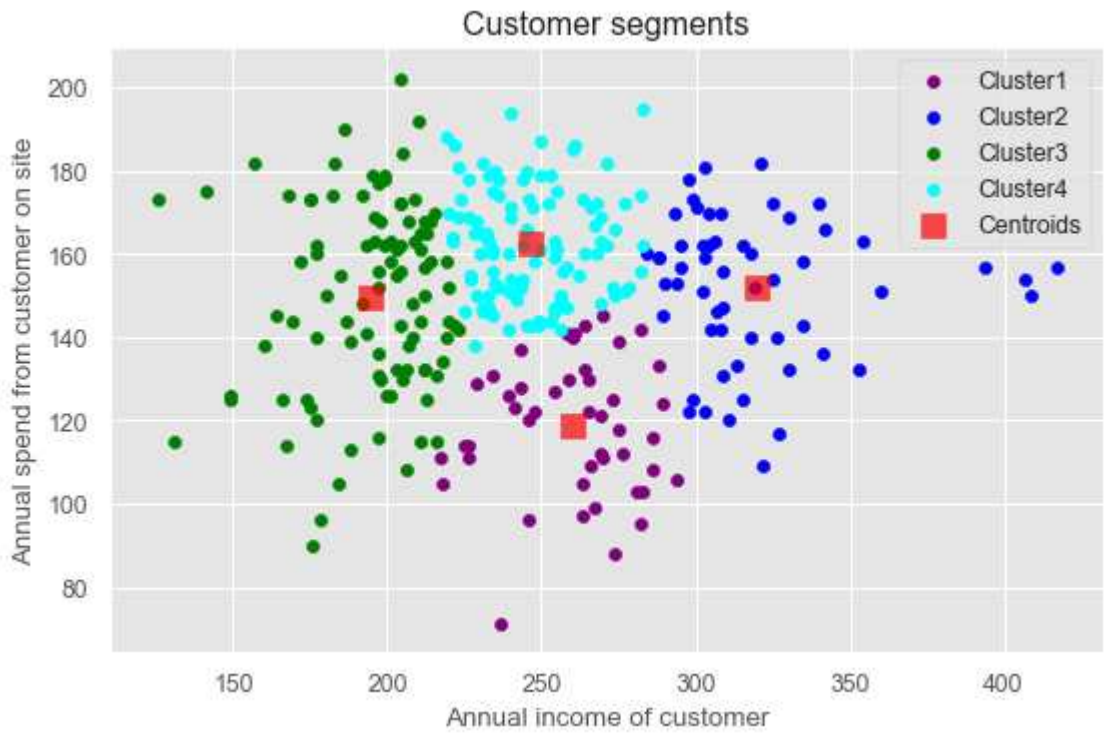
In [15]:
```
##Fitting kmeans to the dataset
km4=KMeans(n_clusters=4, init='k-means++', max_iter=300, n_init=10, random_state=0)
y_means = km4.fit_predict(X)
```

**Cluster လေးခု(k=4) ခွဲထားပုံကို ဂရပ်ဖြင့် ဖော်ပြခြင်း**

In [16]:
```
from matplotlib.pyplot import figure
figure(num=None, figsize=(8, 5), dpi=80, facecolor='w', edgecolor='k')

plt.scatter(X[y_means==0,0],X[y_means==0,1],s=30, c='purple',label='Cluster1')
plt.scatter(X[y_means==1,0],X[y_means==1,1],s=30, c='blue',label='Cluster2')
plt.scatter(X[y_means==2,0],X[y_means==2,1],s=30, c='green',label='Cluster3')
plt.scatter(X[y_means==3,0],X[y_means==3,1],s=30, c='cyan',label='Cluster4')

plt.scatter(km4.cluster_centers_[:,0], km4.cluster_centers_[:,1],s=100,marker='s', c='red', alpha=0.
7, label='Centroids')
plt.title('Customer segments')
plt.xlabel('Annual income of customer')
plt.ylabel('Annual spend from customer on site')
plt.legend()
plt.show()
```




Ref: https://github.com/sowmyacr/kmeans_cluster (https://github.com/sowmyacr/kmeans_cluster)

In [ ]: