In [1]:
```
!pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\dell\appdata\local\programs\python
\python311\lib\site-packages (1.26.4)
```

```
[notice] A new release of pip available: 22.3 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [2]:
```
!pip install matplotlib
```

```
Requirement already satisfied: matplotlib in c:\users\dell\appdata\local\programs\py
thon\python311\lib\site-packages (3.9.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\dell\appdata\local\progr
ams\python\python311\lib\site-packages (from matplotlib) (1.3.0)
Requirement already satisfied: cycler>=0.10 in c:\users\dell\appdata\local\programs
\python\python311\lib\site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\dell\appdata\local\prog
rams\python\python311\lib\site-packages (from matplotlib) (4.54.1)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\dell\appdata\local\prog
rams\python\python311\lib\site-packages (from matplotlib) (1.4.7)
Requirement already satisfied: numpy>=1.23 in c:\users\dell\appdata\local\programs\p
ython\python311\lib\site-packages (from matplotlib) (1.26.4)
Requirement already satisfied: packaging>=20.0 in c:\users\dell\appdata\roaming\pyth
on\python311\site-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=8 in c:\users\dell\appdata\local\programs\pyt
hon\python311\lib\site-packages (from matplotlib) (10.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\dell\appdata\local\progr
ams\python\python311\lib\site-packages (from matplotlib) (3.1.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\dell\appdata\roaming
\python\python311\site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\users\dell\appdata\roaming\python\pyth
on311\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
```

```
[notice] A new release of pip available: 22.3 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [3]:
```
!pip install pandas
```

```
Requirement already satisfied: pandas in c:\users\dell\appdata\local\programs\python
\python311\lib\site-packages (2.2.3)
Requirement already satisfied: numpy>=1.23.2 in c:\users\dell\appdata\local\programs
\python\python311\lib\site-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\dell\appdata\roami
ng\python\python311\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\dell\appdata\local\programs
\python\python311\lib\site-packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\dell\appdata\local\program
s\python\python311\lib\site-packages (from pandas) (2024.2)
Requirement already satisfied: six>=1.5 in c:\users\dell\appdata\roaming\python\pyth
on311\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

```
[notice] A new release of pip available: 22.3 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [4]:
```
!pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in c:\users\dell\appdata\local\programs
\python\python311\lib\site-packages (1.5.2)
Requirement already satisfied: numpy>=1.19.5 in c:\users\dell\appdata\local\programs
\python\python311\lib\site-packages (from scikit-learn) (1.26.4)
Requirement already satisfied: scipy>=1.6.0 in c:\users\dell\appdata\local\programs
\python\python311\lib\site-packages (from scikit-learn) (1.14.1)
Requirement already satisfied: joblib>=1.2.0 in c:\users\dell\appdata\local\programs
\python\python311\lib\site-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\dell\appdata\local\p
rograms\python\python311\lib\site-packages (from scikit-learn) (3.5.0)
```

```
[notice] A new release of pip available: 22.3 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [1]:

```python
import numpy as np
import math
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.patches as patches
from sklearn.cluster import KMeans
```

Create input function from given text

- Ex: '1 2 3
  4 5 6 =>[[1,2,3],[4,5,6],[7,8,9]]
  7 8 9'

In [2]:

```python
#Create an inputting function for a 2D array from text
def inputX(x): #x is str
    x=x.split('\n')
    res=[]
    for i in x:
        i=i.split()
        temp=[]
        for j in i:
            temp.append(int(j))
        res.append(temp)
    return res
```

In [3]:

```python
s='''65 55
70 52
45 42
61 34
37 31'''
arr=inputX(s)
```

In [4]:

```python
print('output:\n {}'.format(arr))
```

```
output:
 [[65, 55], [70, 52], [45, 42], [61, 34], [37, 31]]
```

Doing all the statistic calculation in K-Means (Means, Variance,...)

In [5]:
```
arr=np.array(arr) #Set up numpy array
```

In [6]:
```
sumCols=np.sum(arr, axis=0) #Sum by cols
```

In [7]:
```
meanCols=np.mean(arr, axis=0) #Mean by cols
```

In [8]:
```
meanAll=np.mean(arr) #Meam of the dataset
```

In [9]:
```
#Defined n and other variables
n=arr.shape[0] #Row
m=arr.shape[1] #Col
```

In [10]:
```
#Make all the available for all function
global sumCols, meanCols, meanAll, n, m
```

In [11]:
```
print('Sum of col: {} \n Mean of col: {}'.format(sumCols, meanCols))
print('Mean: {}'.format(meanAll))
print(f'Shape: {m}, {n}')
```

```
Sum of col: [278 214]
 Mean of col: [55.6 42.8]
Mean: 49.2
Shape: 2, 5
```

> **Variance**

> Population Variance:
>
> - $$\sigma_j^2 = \frac{\sum_{i=1}^{N}(x_{ij} - \mu_j)^2}{N}$$

> Sample Variance:
>
> - $$s_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2$$

> Covariance:
>
> - $$s_{jk} = \frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)$$

In [12]:
```
#Variance
```

In [13]:
```
def variance(arr):
    var=[]
    for i in range(m):
        var.append(math.sqrt(np.sum((arr[:,i]-meanCols[i])**2)/(m)))
        print('Variance of col #{}: {}'.format(i+1, var[i]))
    return var
```

In [14]:
```
#Sample Variance
```

In [15]:
```
def sample_variance(arr):
    sV=[]
    for i in range(m):
        sumVals=0 #Restoring sum of square difs

        for j in arr[:,i]: #Taking only columns in the array
            sumVals+=(j-meanCols[i])**2

        sV.append(math.sqrt(sumVals/(n-1)))
        print('Variance of col #{}: {}'.format(i+1, sV[i]))
    return sV
```

In [16]:
```
#Covariance *Note: Only make calculation by given j, and k
```

In [17]:
```
# c=c1=0 #Checking variables (Conds Vars)
# while c==c1 or (c not in range(m) or c1 not in range(m)): # Only allow j<>k and j
#     c=j_val=int(input('Enter j: '))-1
#     c1=k_val=int(input('Enter k: '))-1

def covariance(arr, j_val, k_val):
    coVariance=0
    sumVals=0 #Restoring sum of (xij-x¯j)(xik-x¯k)
    for i in range(n): #Interate through each row
        sumVals+=(arr[i][j_val]-meanCols[j_val])*(arr[i][k_val]-meanCols[k_val]) #(

    coVariance=sumVals/(n-1)
    print('Covariance of values on col #{} and col #{}: {}'.format(j_val, k_val,coV
    return coVariance
```

> **Normalization, Standardization**
>
> - > standardization:
>
> $$x' = \frac{x - \bar{x}}{\sigma}$$
>
> - > min-max normalization:
>
> $$x' = \frac{\sum_{i=1}^{n}(x_{ij} - x_{min})}{x_{max} - x_{min}}$$

In [18]:
```
#Normalization
```

In [19]:
```
def normalization(arr):
    norm=[] #Create empty restoring place
    for i in range(m):
        print(f'Col #{i}:', end=' ')
        norm.append((arr[:,i]-arr[:,i].min())/(arr[:,i].max()-arr[:,i].min()))
        print(norm[i])
    return norm
```

In [20]:
```
#Standardization
```

In [21]:
```
def standardization(arr):
    standardization=[] #z-scores
    for i in range(m):
        print(f'Col #{i}:', end=' ')
        standardization.append((arr[:,i]-meanCols[i])/sample_variance[i])
        print(standardization[i])
    return standardization
```

In [22]:
```
#Standard deviation
std=variance(arr)
```

```
Variance of col #1: 19.788885769542457
Variance of col #2: 15.013327412669051
```

**Euclidean Distance **

> - $$d(x,y)=\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\cdots+(x_n-y_n)^2}$$

In [23]:
```
dists=[np.linalg.norm(arr[i]-meanAll) for i in range(m)] #Calculate the distance fr
```

In [24]:
```
print('Dists: {}'.format(dists))
```

```
Dists: [16.830923919975394, 20.987615395751845]
```

Plotting

> - Only for 2 first cols

In [25]:
```
#Generating all ele by funcs
norm=np.array(normalization(arr))
standardization=np.std(norm, axis=1)
meanNor=np.mean(norm, axis=1)
```

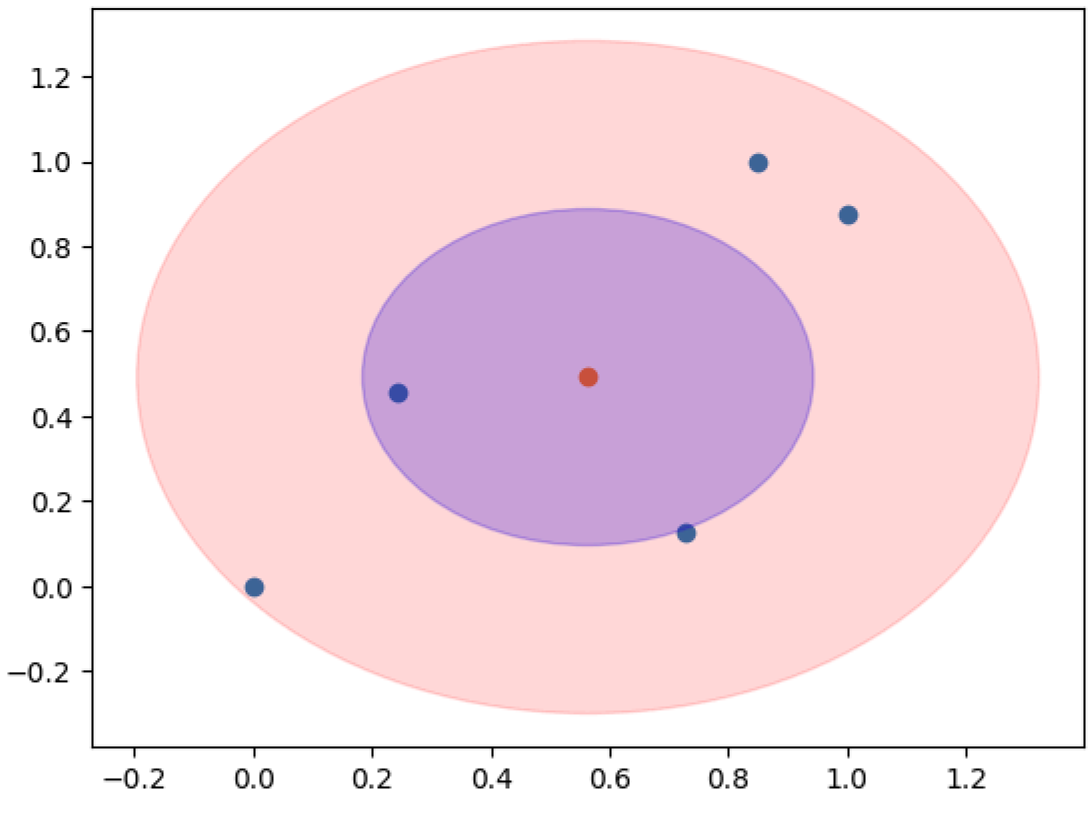
```
Col #0: [0.84848485 1.         0.24242424 0.72727273 0.        ]
Col #1: [1.         0.875      0.45833333 0.125      0.        ]
```

In [26]:
```
ellipse = patches.Ellipse(
    [mean for mean in meanNor],
    standardization[0] * m, # Access the first element of the array
    standardization[1] * m, # Access the first element of the array
    alpha=0.25,
    color='blue'
)

#The second standard deviation can cover the whole data account for all the outlier
ellipse2 = patches.Ellipse(
    [mean for mean in meanNor],
    2*standardization[0] * m, # Access the first element of the array
    2*standardization[1] * m, # Access the first element of the array
    alpha=0.15,
    color='red'
)

fig, graph=plt.subplots()

graph.scatter(norm[0], norm[1]) # Change normalization to norm
graph.scatter(meanNor[0], meanNor[1])
graph.add_patch(ellipse)
graph.add_patch(ellipse2)
plt.show()
```



Using with REAL DATA!

> Link to the dataset: here

> Description: happyscore_income.csv, has been downloaded and saved to local drive.

> Path: /content/drive/MyDrive/K-Mean_Dataset/happyscore_income.csv

In [27]:
```
#Using pandas as data normalizer tool.
dt=pd.read_csv('D:\K-Means\happyscore_income.csv')
dt.head()
```

Out[27]:

	country	adjusted_satisfaction	avg_satisfaction	std_satisfaction	avg_income	median_ir
0	Armenia	37.0	4.9	2.42	2096.76	1731.5
1	Angola	26.0	4.3	3.19	1448.88	1044.2
2	Argentina	60.0	7.1	1.91	7101.12	5109.4
3	Austria	59.0	7.2	2.11	19457.04	16879.6
4	Australia	65.0	7.6	1.80	19917.00	15846.0

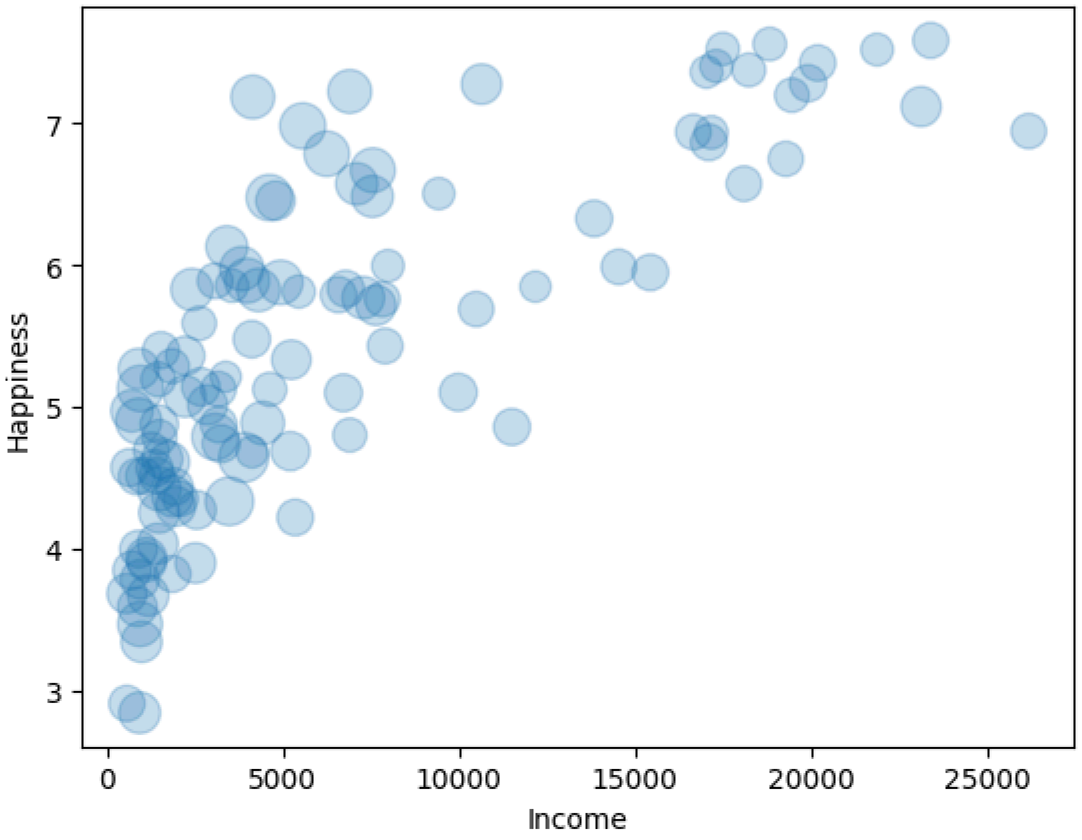
In [28]:
```
dt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 111 entries, 0 to 110
Data columns (total 11 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   country                111 non-null    object 
 1   adjusted_satisfaction  111 non-null    float64
 2   avg_satisfaction       111 non-null    float64
 3   std_satisfaction       111 non-null    float64
 4   avg_income             111 non-null    float64
 5   median_income          111 non-null    float64
 6   income_inequality      111 non-null    float64
 7   region                 111 non-null    object 
 8   happyScore             111 non-null    float64
 9   GDP                    111 non-null    float64
 10  country.1              111 non-null    object 
dtypes: float64(8), object(3)
memory usage: 9.7+ KB
```

In [29]:
```
#Scatter the dataset for overall look with the relationship of happy and income
happy=dt['happyScore']
income=dt['avg_income']
iqe=dt['income_inequality']

plt.xlabel('Income')
plt.ylabel('Happiness')
plt.scatter(income,happy,s=iqe*5, alpha=0.25)
```

Out[29]:
```
<matplotlib.collections.PathCollection at 0x1e69753f750>
```



In [72]:
```
#Create some filters
dt.sort_values('avg_income', inplace=True)
richest=dt[dt['avg_income']>15000]
```

In [73]:
```
richest.iloc[:]
```

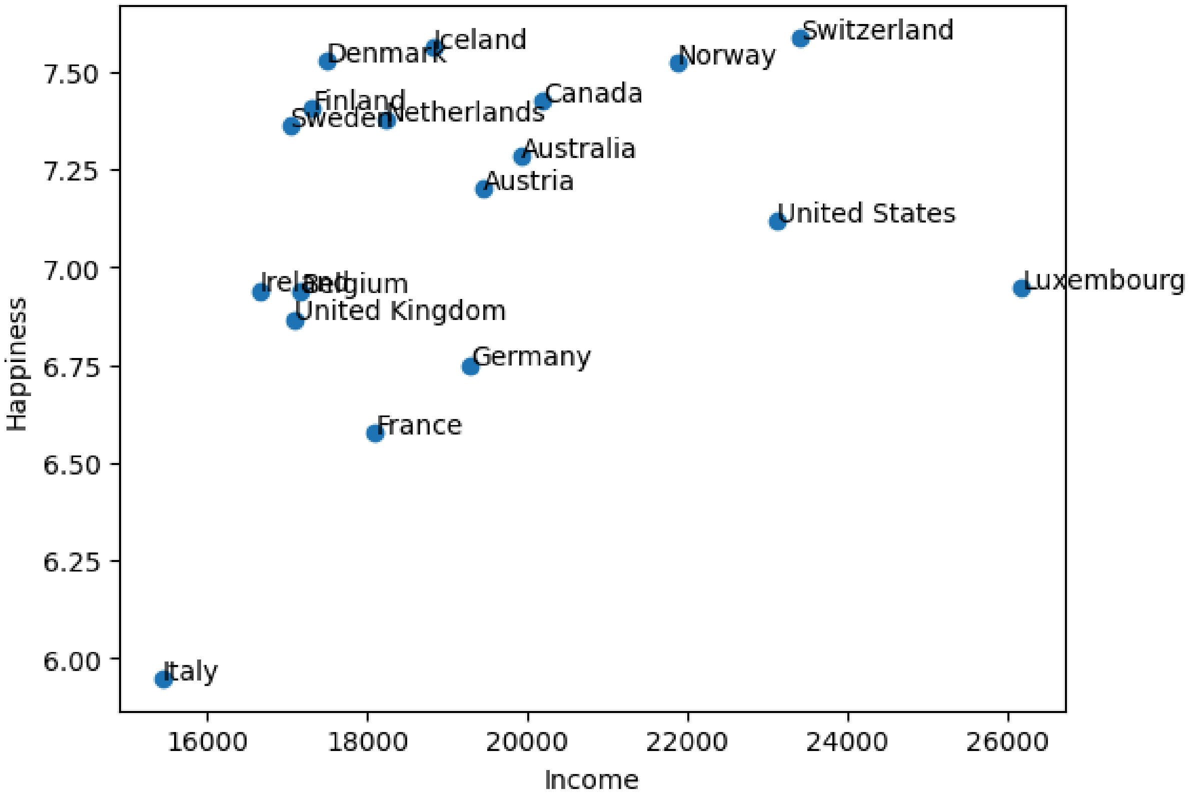
Out[73]:

	country	adjusted_satisfaction	avg_satisfaction	std_satisfaction	avg_income	me
49	Italy	57.0	6.6	1.81	15437.595000	1
44	Ireland	64.0	7.5	1.85	16657.770000	1
90	Sweden	67.0	7.8	1.72	17032.755000	1
36	United Kingdom	60.0	7.1	1.98	17099.550000	1
7	Belgium	63.0	7.2	1.72	17168.505000	1
33	Finland	70.0	7.9	1.53	17310.195000	1
27	Denmark	74.0	8.4	1.53	17496.510000	1
34	France	52.0	6.4	2.15	18096.788571	1
76	Netherlands	69.0	7.6	1.38	18234.435000	1
48	Iceland	71.0	8.1	1.64	18828.345000	1
25	Germany	61.0	7.2	1.99	19285.960000	1
3	Austria	59.0	7.2	2.11	19457.040000	1
4	Australia	65.0	7.6	1.80	19917.000000	1
16	Canada	69.0	8.0	1.71	20190.780000	1
77	Norway	70.0	8.0	1.62	21877.710000	1
105	United States	62.0	7.3	1.92	23127.000000	1
17	Switzerland	70.0	8.0	1.62	23400.040000	1
61	Luxembourg	66.0	7.7	1.76	26182.275000	2

In [74]:
```
richMean=np.mean(richest['avg_income'])
allMean=np.mean(dt['avg_income'])
```

In [75]:
```
plt.scatter(richest['avg_income'], richest['happyScore'])
plt.xlabel('Income')
plt.ylabel('Happiness')

for id,inf in richest.iterrows():
    plt.text(inf['avg_income'], inf['happyScore'], inf['country'])
```



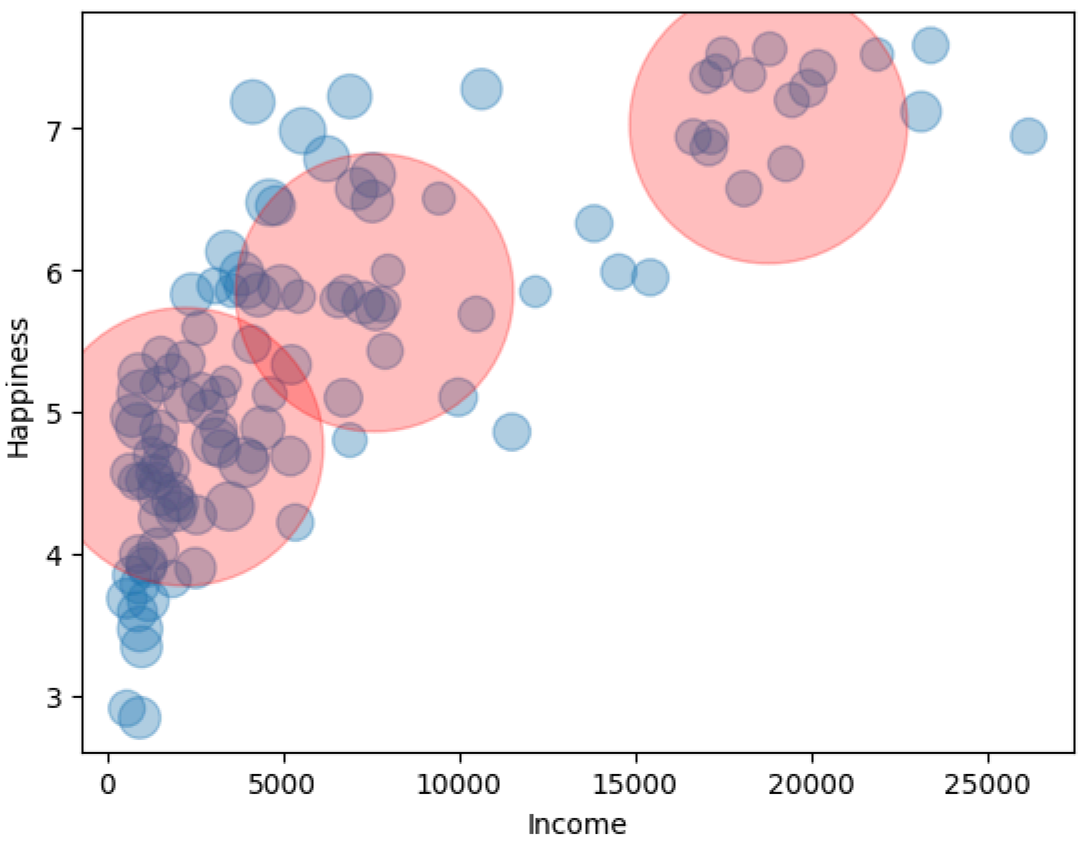
In [76]:
```
income_happy=np.column_stack((income, happy)) #KMeans required difference shape of
km_res=KMeans(n_clusters=3).fit(income_happy)
```

In [77]:
```
km_res.cluster_centers_ #(income, happy)
```

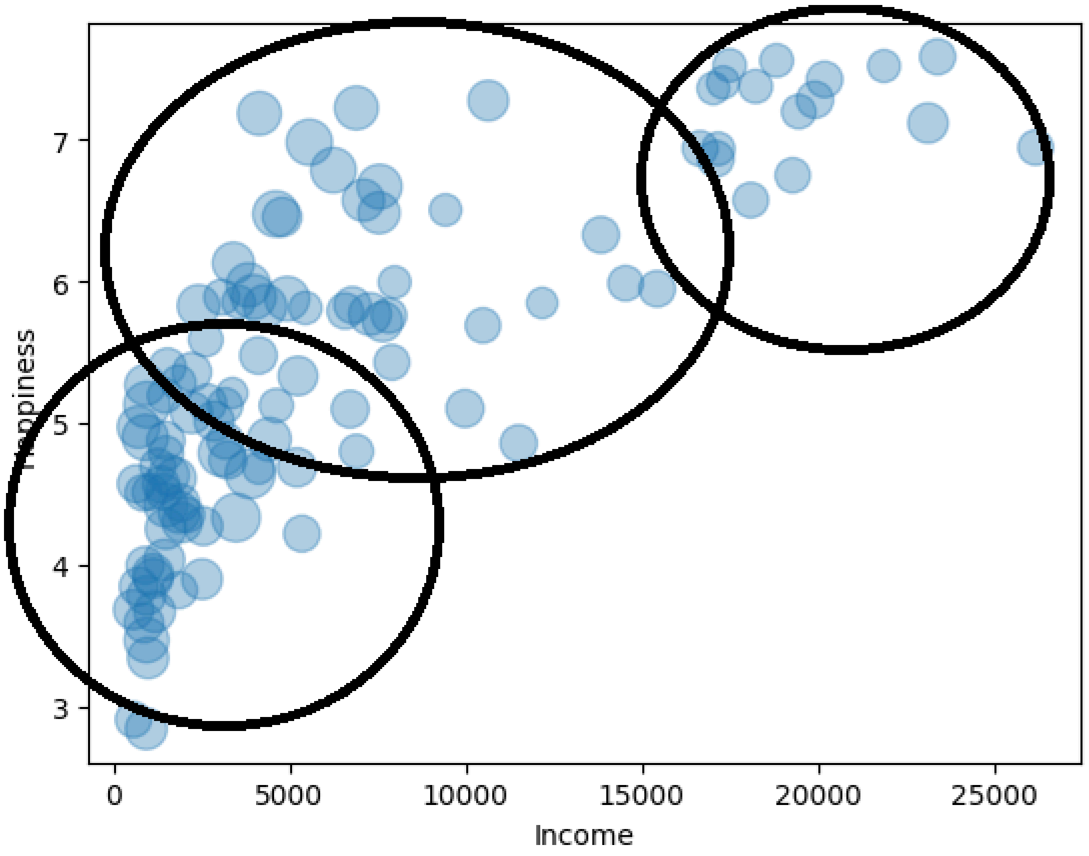
Out[77]:
```
array([[2.19912121e+03, 4.75472308e+00],
       [1.87593022e+04, 7.03270000e+00],
       [7.57755711e+03, 5.85080769e+00]])
```

In [78]:
```
#Scatter the dataset for overall look with the relationship of happy and income
happy=dt['happyScore']
income=dt['avg_income']
iqe=dt['income_inequality']

plt.xlabel('Income')
plt.ylabel('Happiness')
plt.scatter(income,happy,s=iqe*5, alpha=0.35)
plt.scatter(km_res.cluster_centers_[:,0], km_res.cluster_centers_[:,1], color='red'
Out[78]: <matplotlib.collections.PathCollection at 0x1e69d1b9fd0>
```



As we can easily recognize, the **avg_income** contain large integers, or the amount of money. Hence, it can affect the performance of K-Means. As well, I want my final result to be closer to this pic below which is more accurate.



Therefore, we apply normalization to the data to get more accurate data rather than the raw input

In [79]:
```
#Using normalization function, created before.
norm=np.array(normalization(income_happy))

#standardization for more visualize the spread of points from the centre.
standardization=np.std(norm, axis=1)
meanNor=np.mean(norm, axis=1)
```

```
Col #0: [0.00000000e+00 5.15435839e-05 3.13010128e-03 5.53859238e-03
 5.68228964e-03 1.08241526e-02 1.16347926e-02 1.29022962e-02
 1.31108134e-02 1.31764144e-02 1.41932287e-02 1.43900315e-02
 1.45899581e-02 1.49898114e-02 1.62502863e-02 1.86587774e-02
 2.16170667e-02 2.19528810e-02 2.36163330e-02 2.70392955e-02
 3.06515636e-02 3.33955566e-02 3.42061966e-02 3.47909820e-02
 3.58321624e-02 3.58790202e-02 3.59165064e-02 3.60863660e-02
 3.78324973e-02 3.92106100e-02 4.66891155e-02 4.99722856e-02
 5.06001801e-02 5.08547742e-02 5.27056574e-02 5.40577654e-02
 5.95047247e-02 6.44913322e-02 6.47621703e-02 7.20954165e-02
 7.60689583e-02 7.75262360e-02 8.02098493e-02 8.20309444e-02
 8.94984048e-02 9.77922360e-02 9.79156282e-02 1.01574832e-01
 1.01873551e-01 1.04586618e-01 1.09675375e-01 1.10819226e-01
 1.13700460e-01 1.16449972e-01 1.27405327e-01 1.29500912e-01
 1.33264765e-01 1.38528068e-01 1.38886530e-01 1.39601892e-01
 1.46031824e-01 1.50643153e-01 1.57956986e-01 1.58419540e-01
 1.64768438e-01 1.70470251e-01 1.81267851e-01 1.82346622e-01
 1.86726004e-01 1.90596199e-01 1.95020421e-01 2.21695202e-01
 2.34663329e-01 2.40147095e-01 2.42734356e-01 2.47119726e-01
 2.47214104e-01 2.54915823e-01 2.62167068e-01 2.72225096e-01
 2.72755760e-01 2.76239052e-01 2.83302280e-01 2.86373224e-01
 2.89484266e-01 3.45889663e-01 3.67443081e-01 3.87399819e-01
 3.93307222e-01 4.26979435e-01 4.53032373e-01 5.18173506e-01
 5.45499806e-01 5.80439913e-01 6.28085513e-01 6.42727991e-01
 6.45336214e-01 6.48028780e-01 6.53561515e-01 6.60836775e-01
 6.84276554e-01 6.89651396e-01 7.12842494e-01 7.30711522e-01
 7.37391883e-01 7.55352479e-01 7.66043087e-01 8.31914616e-01
 8.80697104e-01 8.91358816e-01 1.00000000e+00]
Col #1: [0.01390059 0.17733783 0.36478517 0.44903117 0.21187869 0.35130581
 0.15754002 0.24347094 0.43365628 0.51158382 0.         0.19839933
 0.13184499 0.48230834 0.10551811 0.35235889 0.22999158 0.22430497
 0.17438922 0.39069082 0.36352148 0.35278012 0.2514743  0.49599832
 0.29759899 0.42902275 0.33277169 0.4100674  0.53917439 0.3778433
 0.37299916 0.51537489 0.2064027  0.32224094 0.33635215 0.30707666
 0.31823926 0.4705139  0.5309604  0.62952822 0.22262005 0.30160067
 0.57919124 0.48462511 0.457877   0.64237574 0.41048863 0.48125527
 0.42860152 0.40016849 0.49978939 0.69313395 0.31444819 0.63521483
 0.66048863 0.37973884 0.64258635 0.55560236 0.38795282 0.915754
 0.62868576 0.43091828 0.76621735 0.48104465 0.76158382 0.64005897
 0.3890059  0.52506318 0.29043808 0.626369   0.87278854 0.83129739
 0.62173547 0.47577928 0.6305813  0.92396799 0.41301601 0.78664701
 0.61731255 0.76790227 0.80686605 0.60593934 0.61499579 0.54549284
 0.66470093 0.77211457 0.47662174 0.60025274 0.93491997 0.42502106
 0.63374052 0.73504634 0.66301601 0.65480202 0.8637321  0.95303286
 0.8483572  0.86310025 0.96187869 0.9873631  0.78685762 0.95598147
 0.99452401 0.82371525 0.918492   0.93618366 0.9663016  0.98631003
 0.90143218 1.         0.86499579]
```

In [80]:
```python
ellipse = patches.Ellipse(
    [mean for mean in meanNor],
    standardization[0] * m, # Access the first element of the array
    standardization[1] * m, # Access the first element of the array
    alpha=0.25,
    color='blue'
)
```

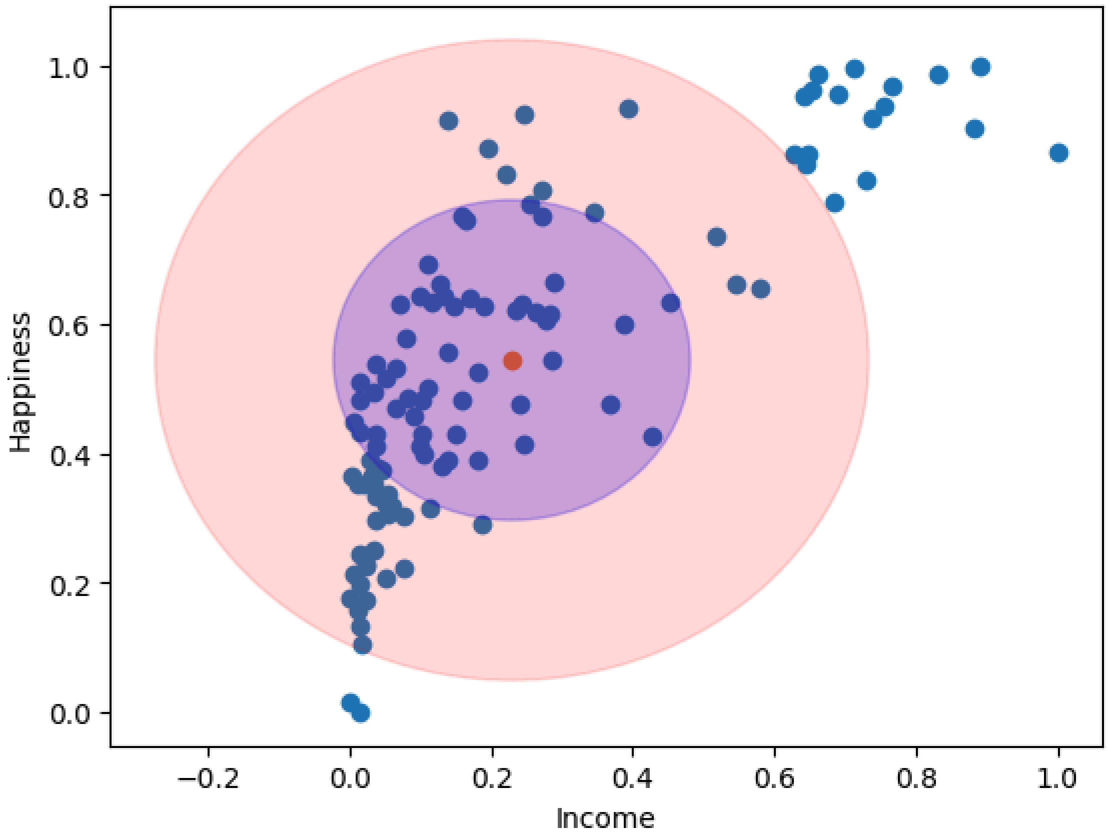
```
#The second standard deviation can cover the whole data account for all the outlier
ellipse2 = patches.Ellipse(
    [mean for mean in meanNor],
    2*standardization[0] * m, # Access the first element of the array
    2*standardization[1] * m, # Access the first element of the array
    alpha=0.15,
    color='red'
)

fig, graph=plt.subplots()

plt.xlabel('Income')
plt.ylabel('Happiness')

graph.scatter(norm[0], norm[1]) # Change normalization to norm
graph.scatter(meanNor[0], meanNor[1])

graph.add_patch(ellipse)
graph.add_patch(ellipse2)
plt.show()
```



With the illustration above, we can confirm that the original data is transformed from 3->7 to the scale of 0->1, and so on with the avg_icome

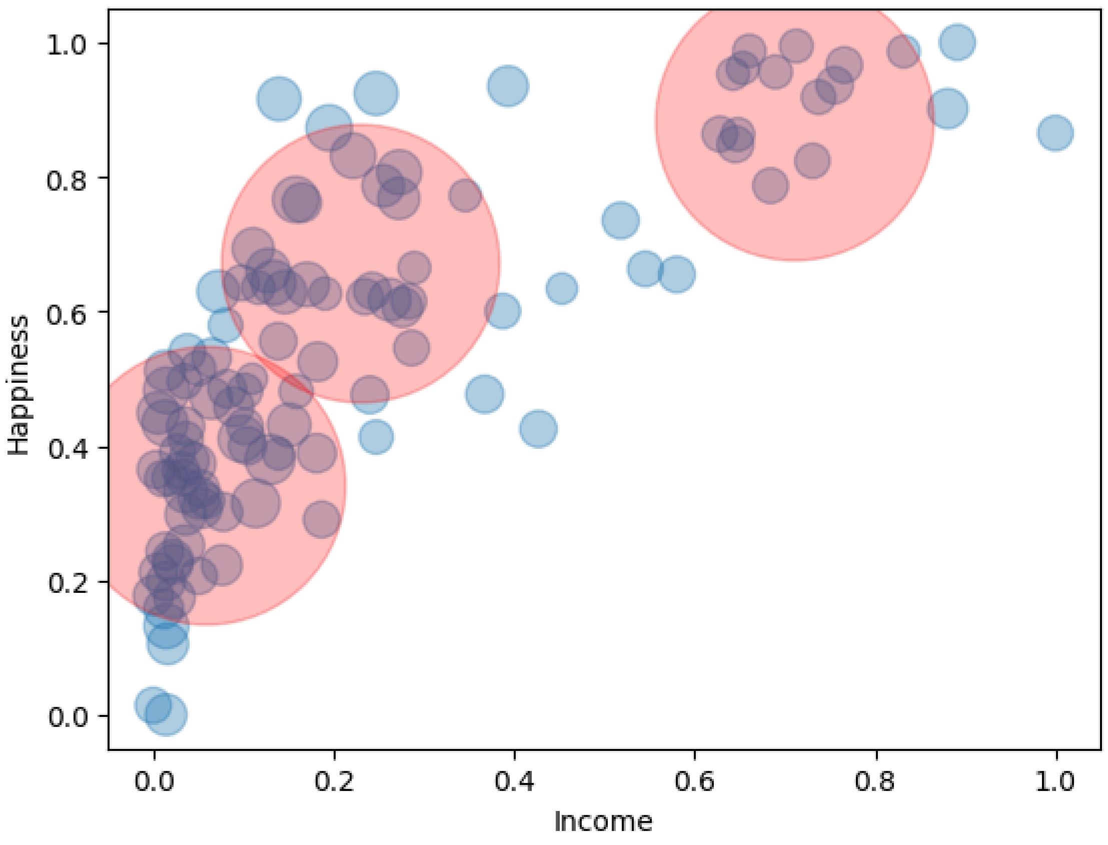
In [81]:
```
def reshaped(arr):
    arr=np.array(arr)
    col=len(arr)
    row=len(arr[0])
    res=np.array([[None]*col]*row)
    for i in range(len(arr)):
        res[:,i]=arr[i]
    return res
```

In [82]:
```
#Because, with my norm function it returned the array with shape [[values of col 1]
reshaped_norm=reshaped(norm) #Reshaped the data from [[a1, a2, ..., an], [b1, b2, .
#to
# [[a1, b1],
#  [a2, b2],
#  ...
#  [an, bn]]
km_res_new=KMeans(n_clusters=3).fit(reshaped_norm)
```

Plotting and Enjoy!

In [83]:
```
plt.xlabel('Income')
plt.ylabel('Happiness')
plt.scatter(reshaped_norm[:,0],reshaped_norm[:,1],s=iqe*5, alpha=0.35)
plt.scatter(km_res_new.cluster_centers_[:,0], km_res_new.cluster_centers_[:,1], col
```

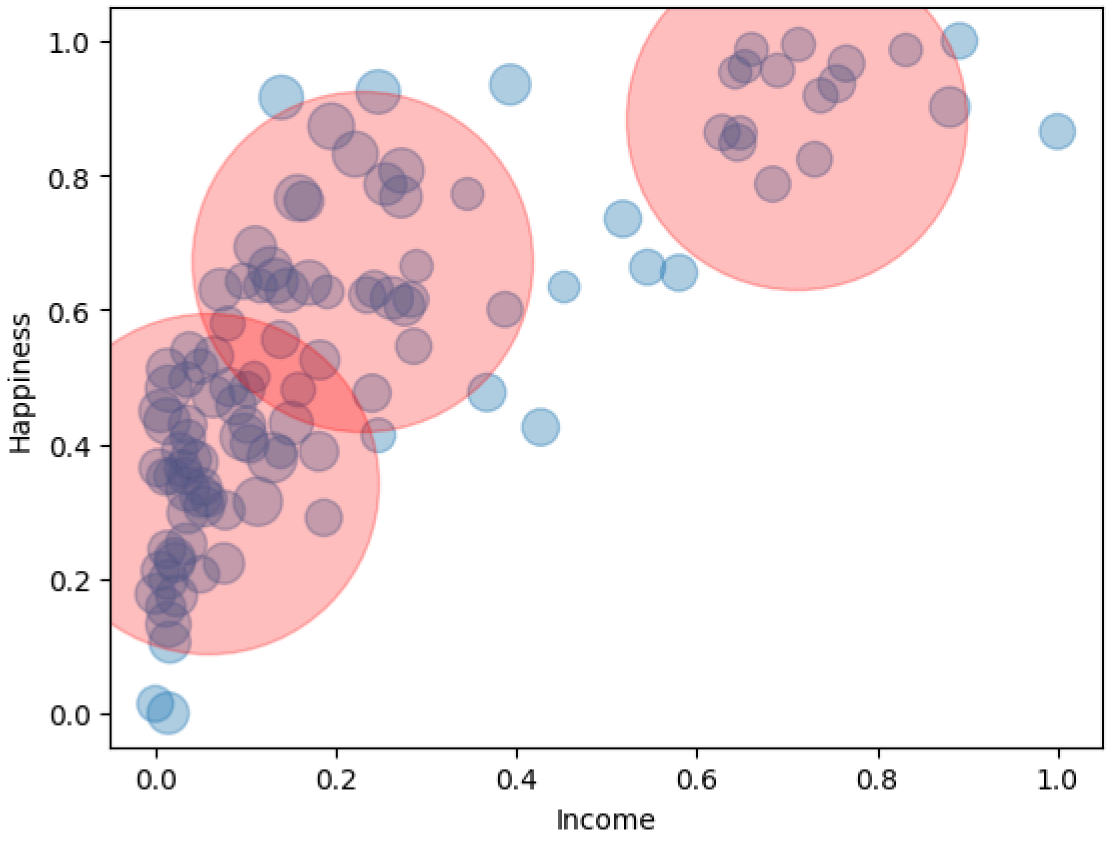
Out[83]:
```
<matplotlib.collections.PathCollection at 0x1e69d2b8210>
```



We can clearly see it has improved much better!

In [84]:
```
plt.xlabel('Income')
plt.ylabel('Happiness')
plt.scatter(reshaped_norm[:,0],reshaped_norm[:,1],s=iqe*5, alpha=0.35)
plt.scatter(km_res_new.cluster_centers_[:,0], km_res_new.cluster_centers_[:,1], col
```

Out[84]:
```
<matplotlib.collections.PathCollection at 0x1e69d2f1750>
```



Now increase the size of the clusters to see the final result more clearly.

In [85]:
```
cluster_labels=km_res_new.labels_
for i, label in enumerate(cluster_labels):
    print(f"Data point {i} is in cluster {label}")
```

```
Data point 0 is in cluster 0
Data point 1 is in cluster 0
Data point 2 is in cluster 0
Data point 3 is in cluster 0
Data point 4 is in cluster 0
Data point 5 is in cluster 0
Data point 6 is in cluster 0
Data point 7 is in cluster 0
Data point 8 is in cluster 0
Data point 9 is in cluster 0
Data point 10 is in cluster 0
Data point 11 is in cluster 0
Data point 12 is in cluster 0
Data point 13 is in cluster 0
Data point 14 is in cluster 0
Data point 15 is in cluster 0
Data point 16 is in cluster 0
Data point 17 is in cluster 0
Data point 18 is in cluster 0
Data point 19 is in cluster 0
Data point 20 is in cluster 0
Data point 21 is in cluster 0
Data point 22 is in cluster 0
Data point 23 is in cluster 0
Data point 24 is in cluster 0
Data point 25 is in cluster 0
Data point 26 is in cluster 0
Data point 27 is in cluster 0
Data point 28 is in cluster 0
Data point 29 is in cluster 0
Data point 30 is in cluster 0
Data point 31 is in cluster 0
Data point 32 is in cluster 0
Data point 33 is in cluster 0
Data point 34 is in cluster 0
Data point 35 is in cluster 0
Data point 36 is in cluster 0
Data point 37 is in cluster 0
Data point 38 is in cluster 0
Data point 39 is in cluster 2
Data point 40 is in cluster 0
Data point 41 is in cluster 0
Data point 42 is in cluster 2
Data point 43 is in cluster 0
Data point 44 is in cluster 0
Data point 45 is in cluster 2
Data point 46 is in cluster 0
Data point 47 is in cluster 0
Data point 48 is in cluster 0
Data point 49 is in cluster 0
Data point 50 is in cluster 0
Data point 51 is in cluster 2
Data point 52 is in cluster 0
Data point 53 is in cluster 2
Data point 54 is in cluster 2
Data point 55 is in cluster 0
Data point 56 is in cluster 2
Data point 57 is in cluster 2
Data point 58 is in cluster 0
Data point 59 is in cluster 2
Data point 60 is in cluster 2
Data point 61 is in cluster 0
Data point 62 is in cluster 2
Data point 63 is in cluster 0
Data point 64 is in cluster 2
Data point 65 is in cluster 2
Data point 66 is in cluster 0
Data point 67 is in cluster 2
Data point 68 is in cluster 0
Data point 69 is in cluster 2
Data point 70 is in cluster 2
Data point 71 is in cluster 2
Data point 72 is in cluster 2
Data point 73 is in cluster 2
Data point 74 is in cluster 2
Data point 75 is in cluster 2
Data point 76 is in cluster 0
Data point 77 is in cluster 2
Data point 78 is in cluster 2
Data point 79 is in cluster 2
Data point 80 is in cluster 2
Data point 81 is in cluster 2
Data point 82 is in cluster 2
Data point 83 is in cluster 2
Data point 84 is in cluster 2
Data point 85 is in cluster 2
Data point 86 is in cluster 2
Data point 87 is in cluster 2
Data point 88 is in cluster 2
Data point 89 is in cluster 2
Data point 90 is in cluster 2
Data point 91 is in cluster 1
Data point 92 is in cluster 1
Data point 93 is in cluster 1
Data point 94 is in cluster 1
Data point 95 is in cluster 1
Data point 96 is in cluster 1
Data point 97 is in cluster 1
Data point 98 is in cluster 1
Data point 99 is in cluster 1
Data point 100 is in cluster 1
Data point 101 is in cluster 1
Data point 102 is in cluster 1
Data point 103 is in cluster 1
Data point 104 is in cluster 1
Data point 105 is in cluster 1
Data point 106 is in cluster 1
Data point 107 is in cluster 1
Data point 108 is in cluster 1
Data point 109 is in cluster 1
Data point 110 is in cluster 1
```

And we can see points belong to different clusters.

In [86]:
```python
#Create a clusters label for our dataset and apply the clusters.
dt['Cluster']=cluster_labels
dt.head()
```

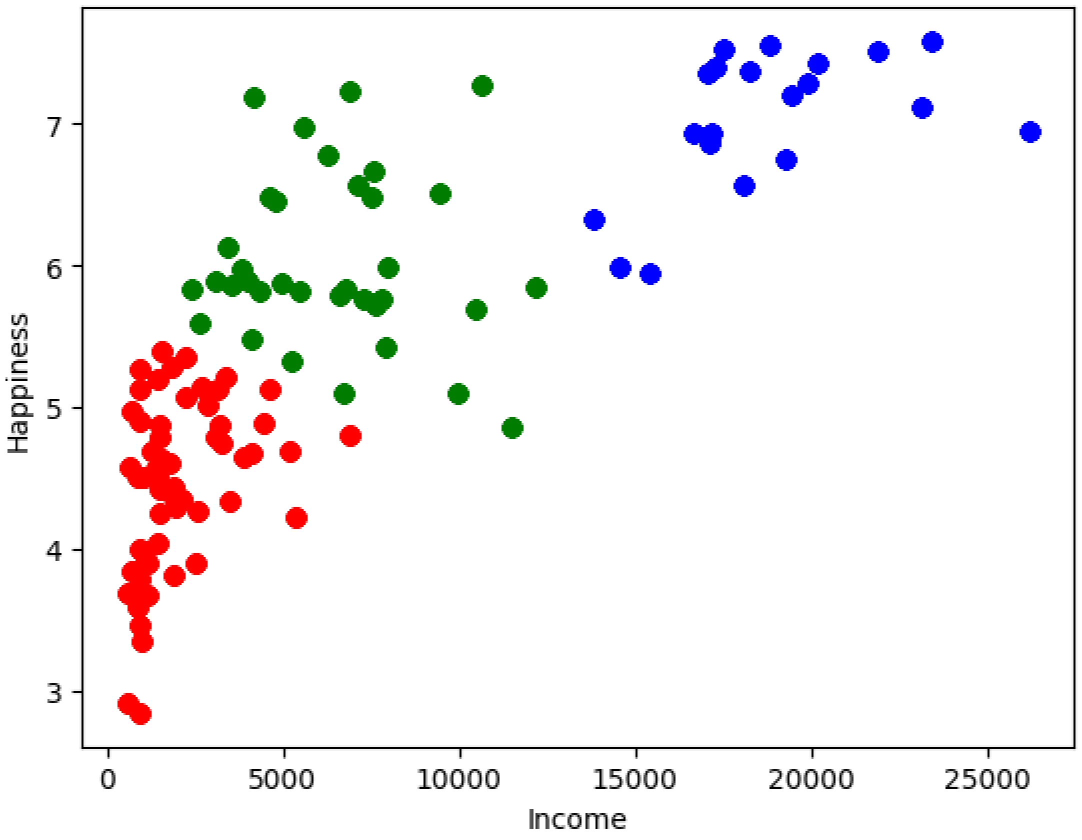
Out[86]:

	country	adjusted_satisfaction	avg_satisfaction	std_satisfaction	avg_income	medi
10	Burundi	25.0	2.9	1.96	572.88	
65	Madagascar	33.0	3.7	1.86	574.20	
58	Liberia	37.0	4.4	2.02	653.04	
72	Mozambique	34.0	3.8	1.76	714.72	
73	Niger	34.0	3.8	1.75	718.40	

Then, we attach the cluster labels to our original dataframe for better understanding.

Final Touch Down!

In [87]:
```python
plt.xlabel('Income')
plt.ylabel('Happiness')
colors=['red','blue','green']
# Loop through each cluster
for i in cluster_labels:
    cluster = dt[dt['Cluster'] == i]
    for id, inf in cluster.iterrows():
        plt.scatter(inf['avg_income'], inf['happyScore'], color=colors[i])
```



EXPORT TO PDF

```
In [94]: !pip install nbconvert PyPDF2
```

```
Requirement already satisfied: nbconvert in c:\users\dell\appdata\local\programs\pyt
hon\python311\lib\site-packages (7.16.4)
Requirement already satisfied: PyPDF2 in c:\users\dell\appdata\local\programs\python
\python311\lib\site-packages (3.0.1)
Requirement already satisfied: beautifulsoup4 in c:\users\dell\appdata\local\program
s\python\python311\lib\site-packages (from nbconvert) (4.12.3)
Requirement already satisfied: bleach!=5.0.0 in c:\users\dell\appdata\local\programs
\python\python311\lib\site-packages (from nbconvert) (6.1.0)
Requirement already satisfied: defusedxml in c:\users\dell\appdata\local\programs\py
thon\python311\lib\site-packages (from nbconvert) (0.7.1)
Requirement already satisfied: jinja2>=3.0 in c:\users\dell\appdata\local\programs\p
ython\python311\lib\site-packages (from nbconvert) (3.1.4)
Requirement already satisfied: jupyter-core>=4.7 in c:\users\dell\appdata\roaming\py
thon\python311\site-packages (from nbconvert) (5.7.2)
Requirement already satisfied: jupyterlab-pygments in c:\users\dell\appdata\local\pr
ograms\python\python311\lib\site-packages (from nbconvert) (0.3.0)
Requirement already satisfied: markupsafe>=2.0 in c:\users\dell\appdata\local\progra
ms\python\python311\lib\site-packages (from nbconvert) (2.1.5)
Requirement already satisfied: mistune<4,>=2.0.3 in c:\users\dell\appdata\local\prog
rams\python\python311\lib\site-packages (from nbconvert) (3.0.2)
Requirement already satisfied: nbclient>=0.5.0 in c:\users\dell\appdata\local\progra
ms\python\python311\lib\site-packages (from nbconvert) (0.10.0)
Requirement already satisfied: nbformat>=5.7 in c:\users\dell\appdata\local\programs
\python\python311\lib\site-packages (from nbconvert) (5.10.4)
Requirement already satisfied: packaging in c:\users\dell\appdata\roaming\python\pyt
hon311\site-packages (from nbconvert) (24.1)
Requirement already satisfied: pandocfilters>=1.4.1 in c:\users\dell\appdata\local\p
rograms\python\python311\lib\site-packages (from nbconvert) (1.5.1)
Requirement already satisfied: pygments>=2.4.1 in c:\users\dell\appdata\roaming\pyth
on\python311\site-packages (from nbconvert) (2.18.0)
Requirement already satisfied: tinycss2 in c:\users\dell\appdata\roaming\python\pyth
on311\site-packages (from nbconvert) (1.3.0)
Requirement already satisfied: traitlets>=5.1 in c:\users\dell\appdata\roaming\pytho
n\python311\site-packages (from nbconvert) (5.14.3)
Requirement already satisfied: six>=1.9.0 in c:\users\dell\appdata\roaming\python\py
thon311\site-packages (from bleach!=5.0.0->nbconvert) (1.16.0)
Requirement already satisfied: webencodings in c:\users\dell\appdata\roaming\python
\python311\site-packages (from bleach!=5.0.0->nbconvert) (0.5.1)
Requirement already satisfied: platformdirs>=2.5 in c:\users\dell\appdata\roaming\py
thon\python311\site-packages (from jupyter-core>=4.7->nbconvert) (4.3.6)
Requirement already satisfied: pywin32>=300 in c:\users\dell\appdata\roaming\python
\python311\site-packages (from jupyter-core>=4.7->nbconvert) (306)
Requirement already satisfied: jupyter-client>=6.1.12 in c:\users\dell\appdata\roami
ng\python\python311\site-packages (from nbclient>=0.5.0->nbconvert) (8.6.3)
Requirement already satisfied: fastjsonschema>=2.15 in c:\users\dell\appdata\roaming
\python\python311\site-packages (from nbformat>=5.7->nbconvert) (2.20.0)
Requirement already satisfied: jsonschema>=2.6 in c:\users\dell\appdata\local\progra
ms\python\python311\lib\site-packages (from nbformat>=5.7->nbconvert) (4.23.0)
Requirement already satisfied: soupsieve>1.2 in c:\users\dell\appdata\roaming\python
\python311\site-packages (from beautifulsoup4->nbconvert) (2.6)
Requirement already satisfied: attrs>=22.2.0 in c:\users\dell\appdata\local\programs
\python\python311\lib\site-packages (from jsonschema>=2.6->nbformat>=5.7->nbconvert)
(24.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in c:\users\dell
\appdata\local\programs\python\python311\lib\site-packages (from jsonschema>=2.6->nb
format>=5.7->nbconvert) (2023.12.1)
```

```
Requirement already satisfied: referencing>=0.28.4 in c:\users\dell\appdata\local\pr
ograms\python\python311\lib\site-packages (from jsonschema>=2.6->nbformat>=5.7->nbco
nvert) (0.35.1)
Requirement already satisfied: rpds-py>=0.7.1 in c:\users\dell\appdata\roaming\pytho
n\python311\site-packages (from jsonschema>=2.6->nbformat>=5.7->nbconvert) (0.20.0)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\dell\appdata\roami
ng\python\python311\site-packages (from jupyter-client>=6.1.12->nbclient>=0.5.0->nbc
onvert) (2.9.0.post0)
Requirement already satisfied: pyzmq>=23.0 in c:\users\dell\appdata\roaming\python\p
ython311\site-packages (from jupyter-client>=6.1.12->nbclient>=0.5.0->nbconvert) (2
4.0.1)
Requirement already satisfied: tornado>=6.2 in c:\users\dell\appdata\roaming\python
\python311\site-packages (from jupyter-client>=6.1.12->nbclient>=0.5.0->nbconvert)
(6.4.1)
```

```
[notice] A new release of pip available: 22.3 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [96]:
```
!set PATH=/Library/TeX/texbin:$PATH
```

In [ ]:
```
```