**# A Simple Example of Clustering**

You are given much more country data. Using the same methodology as the one in the lecture, group all the countries in 2 clusters.

Try with other numbers of clusters and see if they match your expectations. Maybe 7 is going to be a cool one!

Plot the data using the c parameter to separate the data by the clusters we defined.

Note: c stands for color

**## Import the relevant libraries**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

sns.set()

from sklearn.cluster import KMeans

**## Load the data**

Load data from the csv file: <i> 'Countries\_exercise.csv'</i>.

**# Load the data**

raw\_data = pd.read\_csv('Countries\_exercise.csv')

**# Check the data**

raw\_data

**Remove the duplicate index column from the dataset.**

data = raw\_data.copy()

**## Plot the data**

plt.scatter(data['Longitude'], data['Latitude'])

plt.xlim(-180,180)

plt.ylim(-90, 90)

plt.show()

**## Select the features**

Create a copy of that data and remove all parameters apart from *Longitude* and *Latitude*.

x = data.iloc[:,1:3]

x

**## Clustering**

**Here's the actual solution:**

**Simply change kmeans = KMeans(2) to kmeans = KMeans(3)**

**Then run the remaining kernels until the end.**

kmeans = KMeans(3)

kmeans.fit(x)

**### Clustering Results**

identified\_clusters = kmeans.fit\_predict(x)

identified\_clusters

data\_with\_clusters = data.copy()

data\_with\_clusters['Cluster'] = identified\_clusters

data\_with\_clusters

plt.scatter(data['Longitude'], data['Latitude'],c=data\_with\_clusters['Cluster'], cmap = 'rainbow')

plt.xlim(-180,180)

plt.ylim(-90, 90)

plt.show()