ASSQ3

April 12, 2024

1 Question3 a

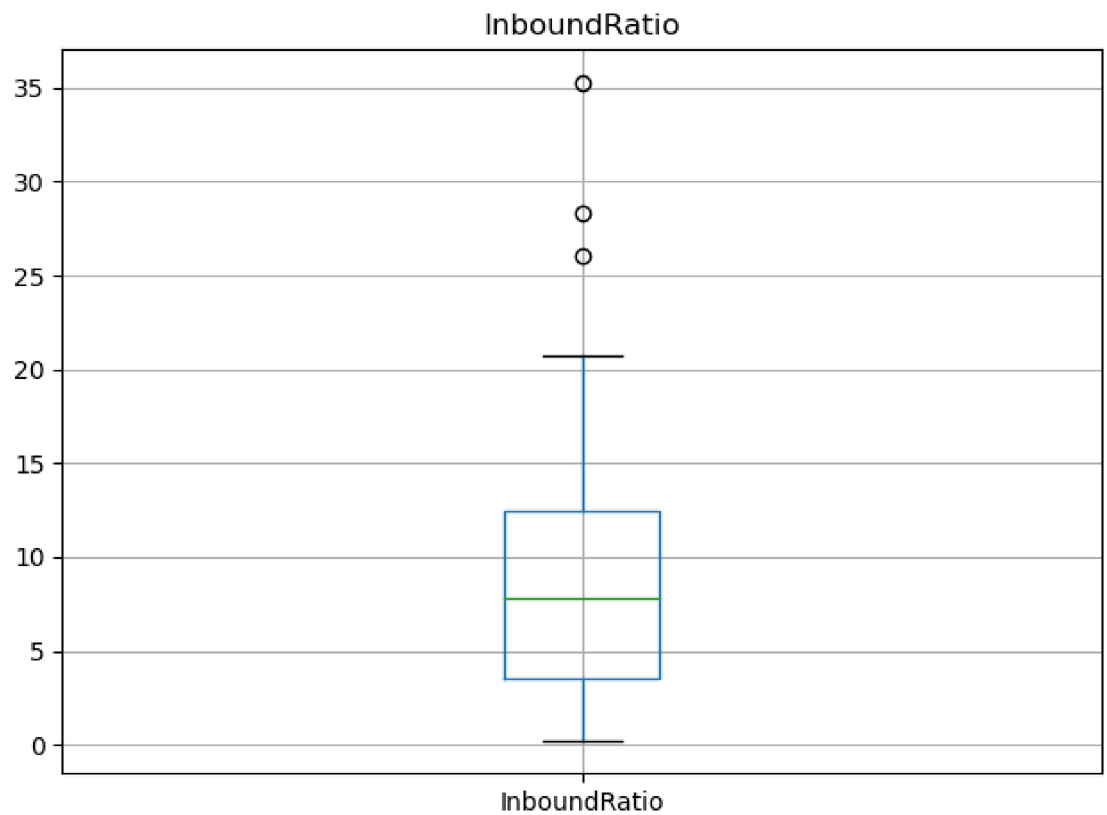
[1]:
```
# Reading the data into Python
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from matplotlib.ticker import MaxNLocator
from sklearn.decomposition import PCA
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.utils import resample
import warnings
df = pd.read_excel("data_q3.xlsx");
warnings.filterwarnings("ignore")
```

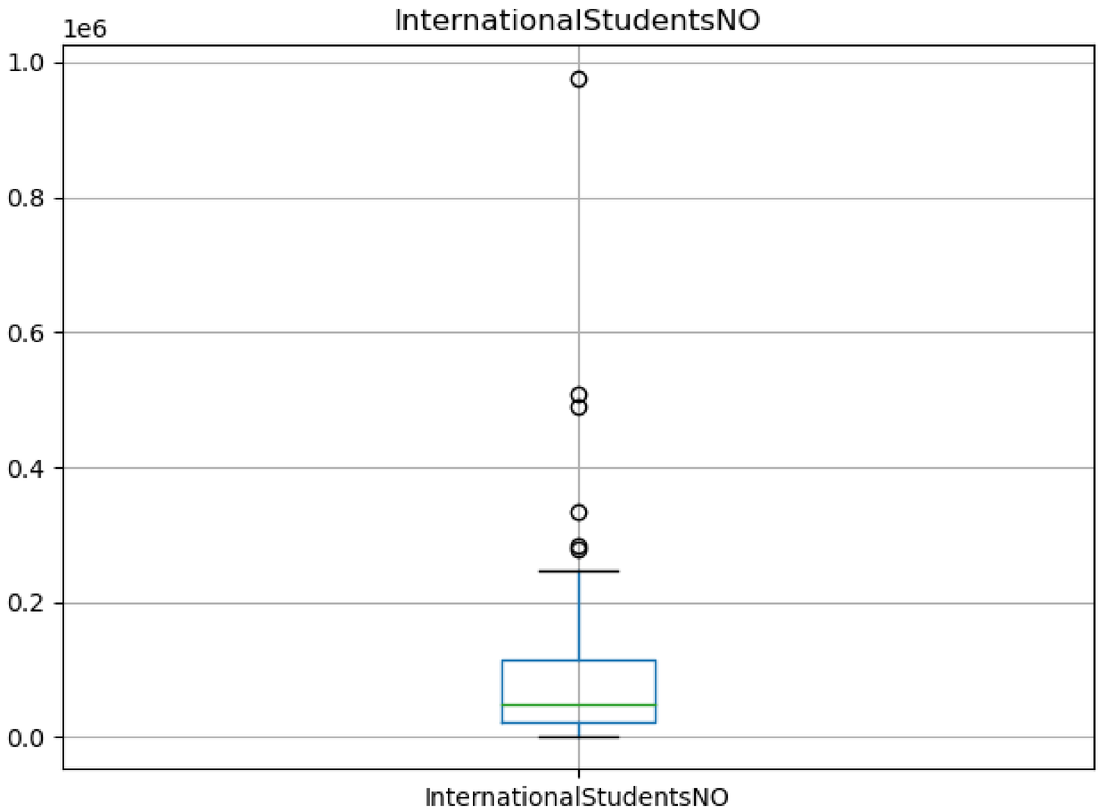
[2]:
```
# Q3a Data preprocessing.
print("Number of observation: ", df.shape[0])      # check dimension
print("Any NA value:", df.isnull().values.any()); # Check for missing values
print("Any row duplictaes:",df.duplicated().any());# check for dupllicates rows
df = df.dropna()
df.reset_index(drop=True, inplace=True)
#Check for data error(negative values)
num_error = (df.select_dtypes(include=['float64', 'int64']) < 0).sum()
print(num_error)
#Check datatype
print(df.dtypes)
#check outliers
interest = ["InboundRatio", "InternationalStudentsNO", "KOFPoGI", "KOFEcGI",␣
 ↪"KOFSoGI", "top_50_count",
                   "top_100_count", "top_500_count", "top_1000_count"]
for i in interest:
    df.boxplot(i)
    plt.title(i)
    plt.tight_layout()
    plt.show()
```

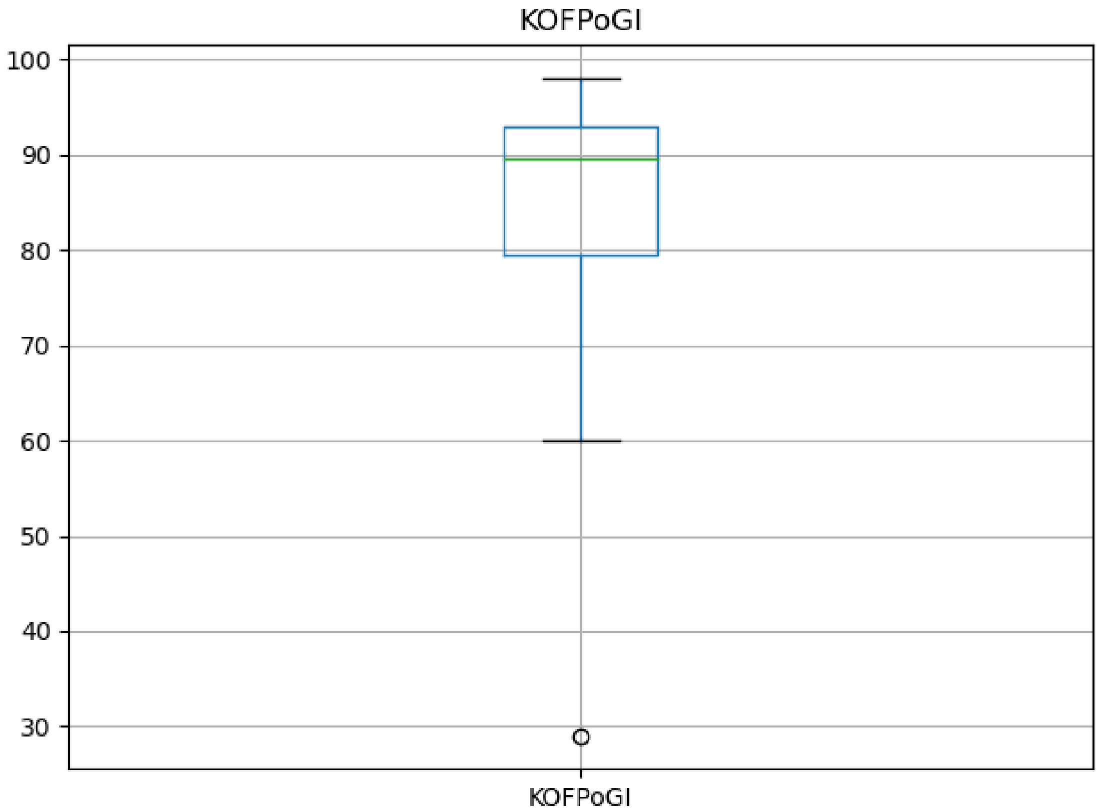
```python
# check normalization
df[["InboundRatio","InternationalStudentsNO","KOFPoGI","KOFEcGI","KOFSoGI","ISCED5␣
↪Percentage","ISCED6 Percentage","ISCED7 Percentage","ISCED8 Percentage",
    "top_50_count","top_100_count","top_500_count","top_1000_count"]].describe()
```

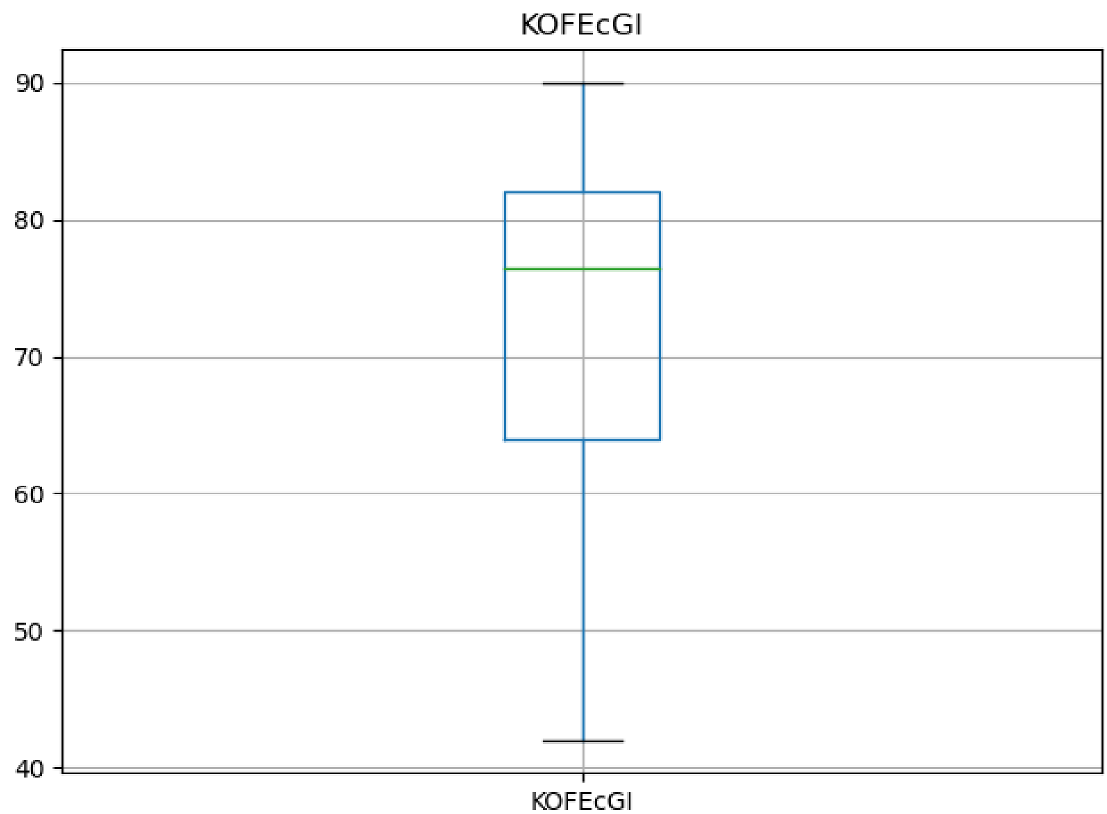
```
Number of observation:  49
Any NA value: True
Any row duplictaes: False
Tertiary Percentage              0
ISCED5 Percentage                0
ISCED6 Percentage                0
ISCED7 Percentage                0
ISCED8 Percentage                0
year                             0
InternationalStudentsNO          0
KOFGI                            0
KOFGIdf                          0
KOFGIdj                          0
KOFPoGI                          0
KOFPoGIdf                        0
KOFPoGIdj                        0
KOFSoGI                          0
KOFSoGIdf                        0
KOFSoGIdj                        0
KOFInGI                          0
KOFInGIdf                        0
KOFInGIdj                        0
KOFIpGI                          0
KOFIpGIdf                        0
KOFIpGIdj                        0
KOFCuGI                          0
KOFCuGIdf                        0
KOFCuGIdj                        0
KOFEcGI                          0
KOFEcGIdf                        0
KOFEcGIdj                        0
KOFTrGI                          0
KOFTrGIdf                        0
KOFTrGIdj                        0
KOFFiGI                          0
KOFFiGIdf                        0
KOFFiGIdj                        0
KOFSoGI_WithoutInterpersonal     0
InboundRatio                     0
top_50_count                     0
top_100_count                    0
top_500_count                    0
top_1000_count                   0
```

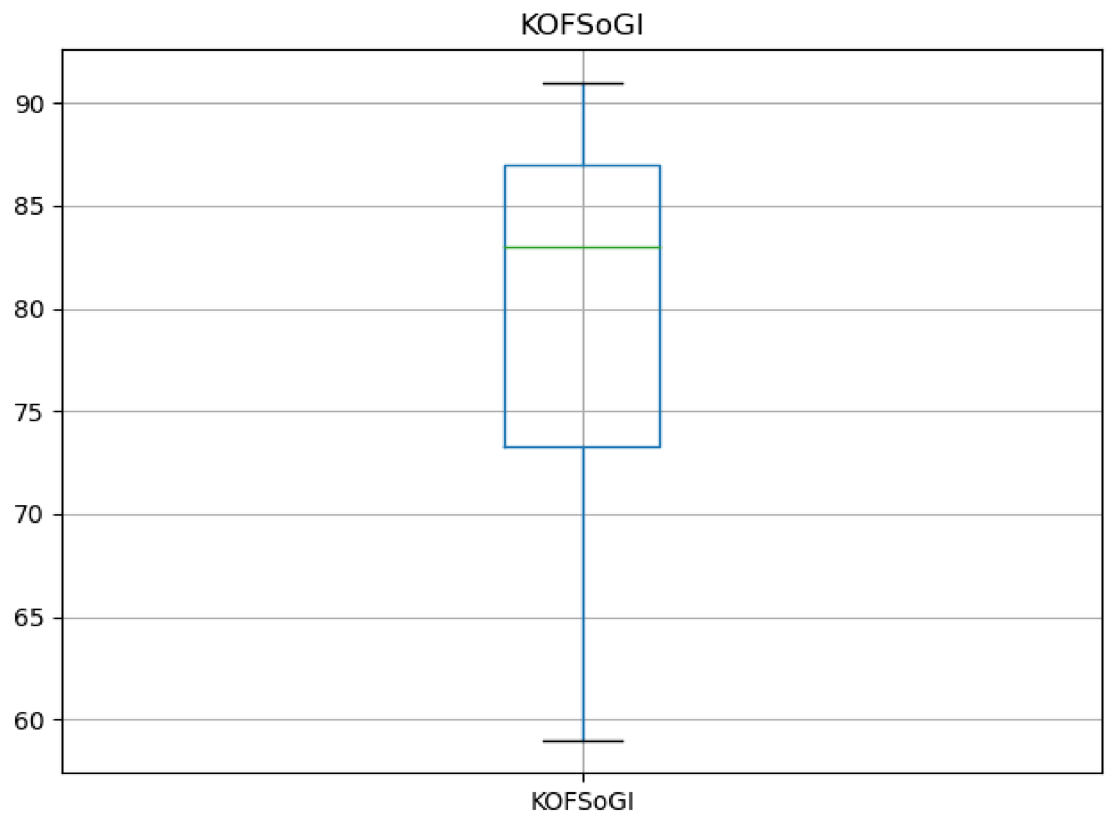
```
total_ranked_universities       0
dtype: int64
country_x                        object
code                             object
Tertiary Percentage             float64
ISCED5 Percentage               float64
ISCED6 Percentage               float64
ISCED7 Percentage               float64
ISCED8 Percentage               float64
country_y                        object
year                              int64
InternationalStudentsNO           int64
KOFGI                             int64
KOFGIdf                           int64
KOFGIdj                           int64
KOFPoGI                           int64
KOFPoGIdf                         int64
KOFPoGIdj                         int64
KOFSoGI                           int64
KOFSoGIdf                         int64
KOFSoGIdj                         int64
KOFInGI                           int64
KOFInGIdf                         int64
KOFInGIdj                         int64
KOFIpGI                           int64
KOFIpGIdf                         int64
KOFIpGIdj                         int64
KOFCuGI                           int64
KOFCuGIdf                         int64
KOFCuGIdj                         int64
KOFEcGI                           int64
KOFEcGIdf                         int64
KOFEcGIdj                         int64
KOFTrGI                           int64
KOFTrGIdf                         int64
KOFTrGIdj                         int64
KOFFiGI                           int64
KOFFiGIdf                         int64
KOFFiGIdj                         int64
KOFSoGI_WithoutInterpersonal    float64
InboundRatio                    float64
top_50_count                      int64
top_100_count                     int64
top_500_count                     int64
top_1000_count                    int64
total_ranked_universities         int64
WESP                             object
dtype: object
```

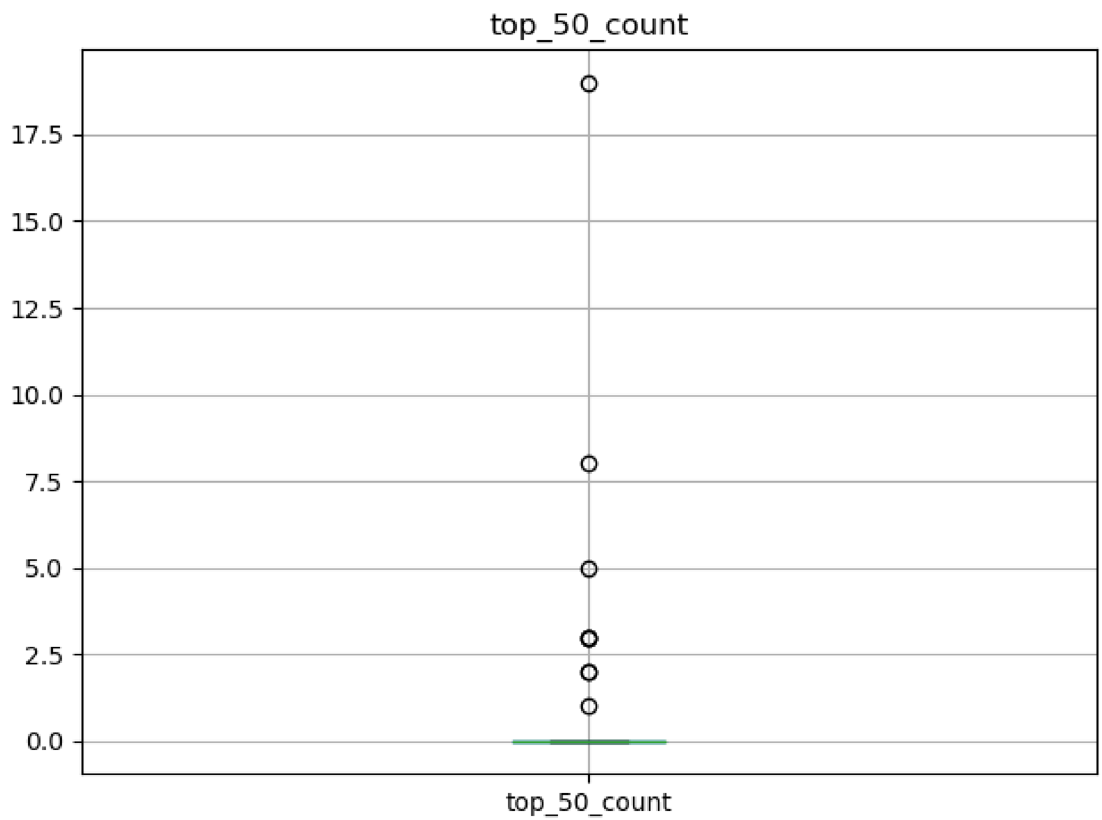

InboundRatio
35
30
25
20
15
10
5
0
InboundRatio

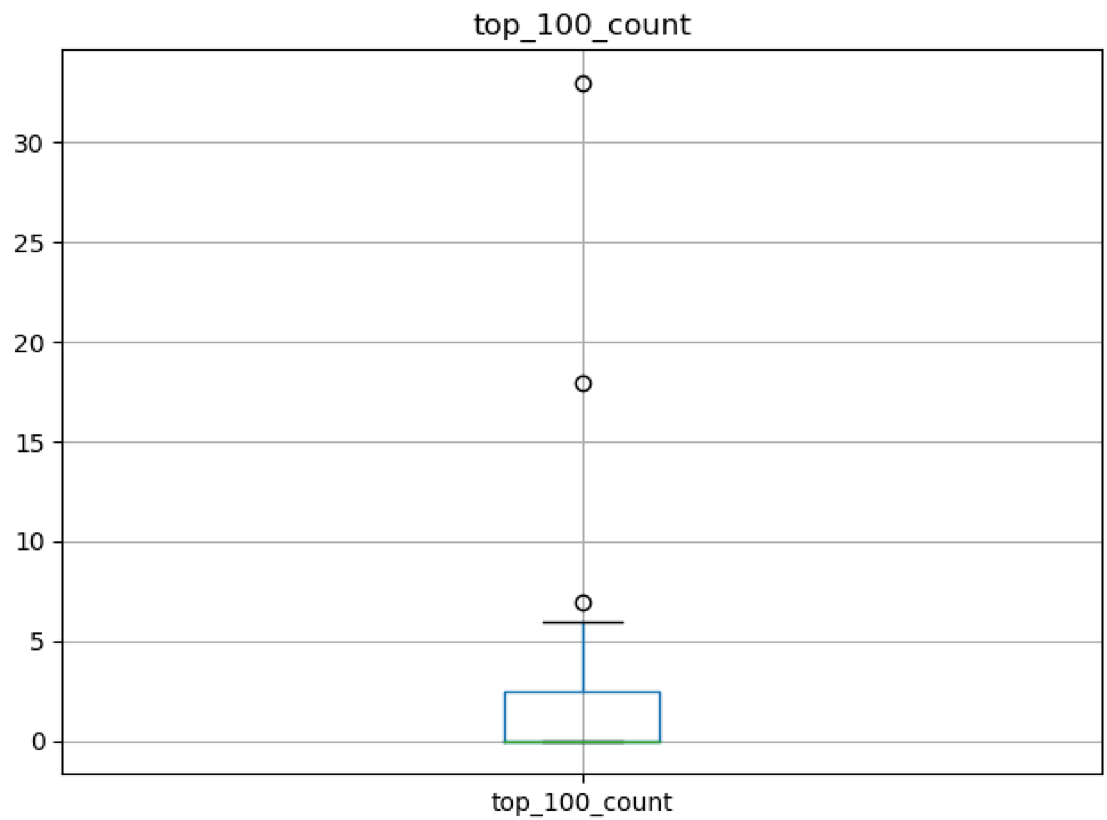

InternationalStudentsNO
1e6
1.0
0.8
0.6
0.4
0.2
0.0
InternationalStudentsNO

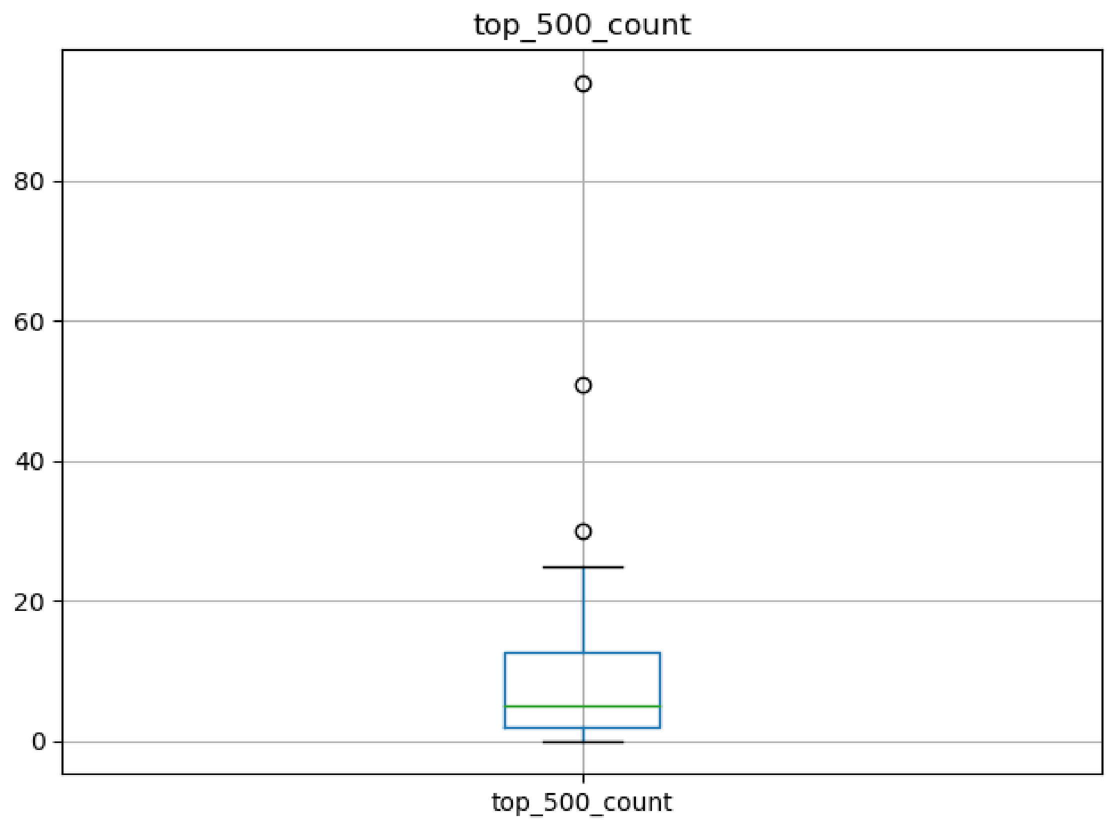

KOFPoGI
100
90
80
70
60
50
40
30
KOFPoGI

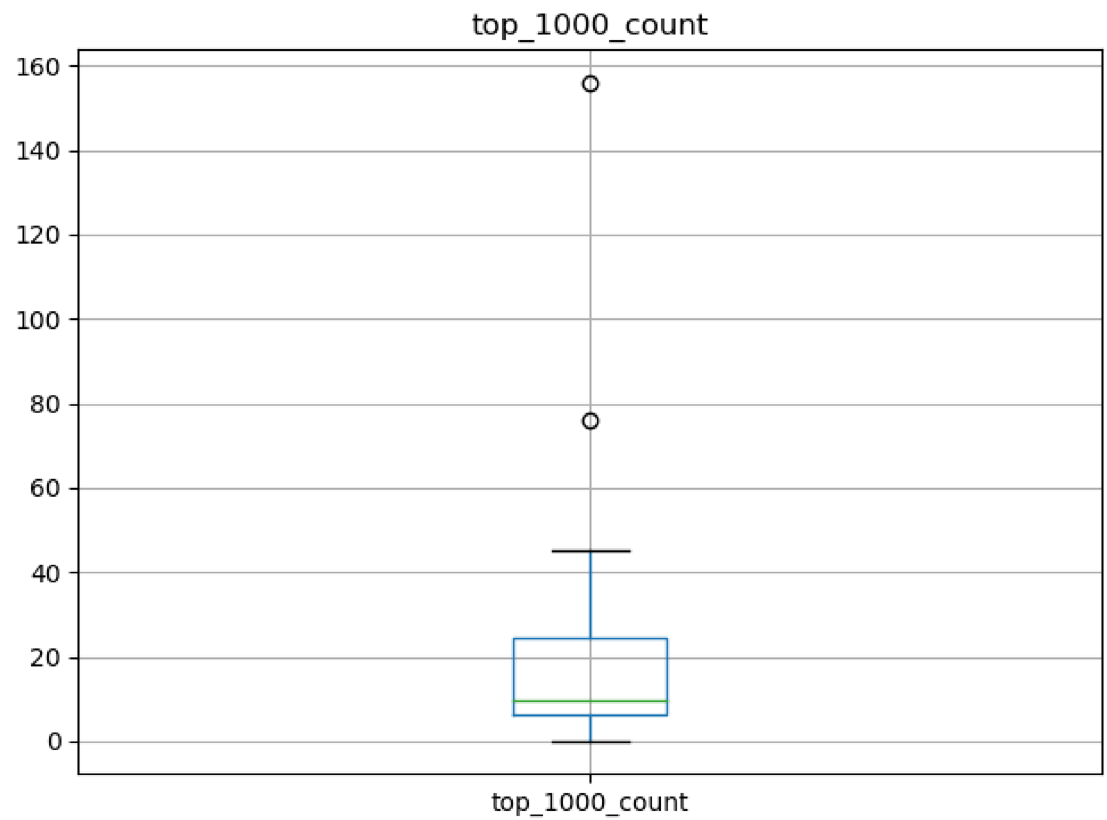

KOFEcGI
90
80
70
60
50
40
KOFEcGI


KOFSoGI
90
85
80
75
70
65
60
KOFSoGI


top_50_count
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
top_50_count


top_100_count
30
25
20
15
10
5
0
top_100_count


top_500_count
80
60
40
20
0
top_500_count




```
[2]:         InboundRatio  InternationalStudentsNO    KOFPoGI    KOFEcGI    KOFSoGI  \
count      42.000000                42.000000  42.000000  42.000000  42.000000
mean        9.368033            117317.380952  84.952381  71.976190  79.976190
std         8.016693            183894.022375  13.510524  12.994348   9.358674
min         0.219050              1546.000000  29.000000  42.000000  59.000000
25%         3.549540             22034.250000  79.500000  64.000000  73.250000
50%         7.800560             49007.000000  89.500000  76.500000  83.000000
75%        12.455103            114335.750000  93.000000  82.000000  87.000000
max        35.293780            976562.000000  98.000000  90.000000  91.000000

       ISCED5 Percentage  ISCED6 Percentage  ISCED7 Percentage  \
count          42.000000          42.000000          42.000000
mean           10.626414          45.236110          14.233167
std             9.801015          13.083961           8.697049
min             0.004350          12.319206           1.083925
25%             2.523087          38.851575           6.738658
50%             8.476903          44.474409          14.806317
75%            16.899843          54.239022          21.464752
max            41.863344          68.238077          35.507974

       ISCED8 Percentage  top_50_count  top_100_count  top_500_count  \
```

```
count        42.000000    42.000000    42.000000    42.000000
mean          2.098529     1.095238     2.261905    10.214286
std           1.353961     3.259579     5.793434    16.543418
min           0.000000     0.000000     0.000000     0.000000
25%           0.804222     0.000000     0.000000     2.000000
50%           2.085667     0.000000     0.000000     5.000000
75%           2.887539     0.000000     2.500000    12.750000
max           5.152113    19.000000    33.000000    94.000000

       top_1000_count
count       42.000000
mean        18.642857
std         26.709660
min          0.000000
25%          6.250000
50%          9.500000
75%         24.250000
max        156.000000
```

In this dataset, we have 49 observations with missing values and no row duplicates. There is no negative value in the numeric variables. We also observe a few outliers in those numeric variables, as indicated by box plots. Since the standard deviation of those variables is quite different, we have to standardize them(This is done in later parts), which is a crucial step for cluster analysis as the distance between data points is a major determinant. Variables on different scales will result in a bias for cluster analysis. Data balancing is not needed for this cluster analysis question. We begin our analysis by dropping all the NA values.

2 Question3 b

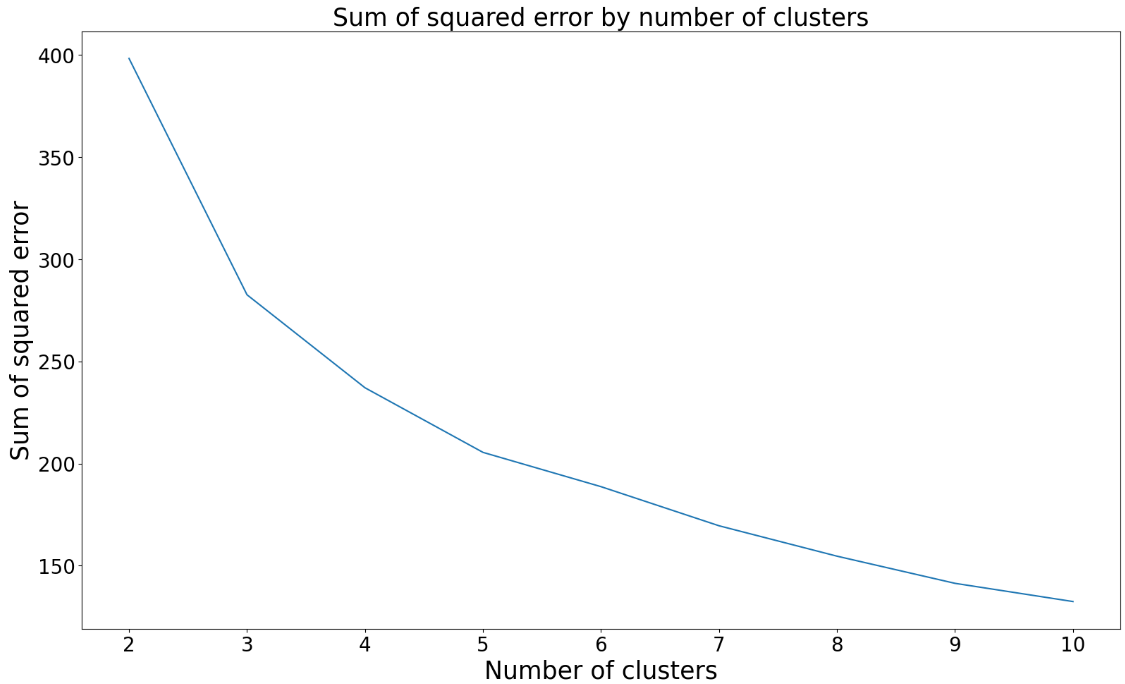
[3]:

```
# Standardising all continious variables
variables_ofstd =␣
 ↪["InboundRatio","InternationalStudentsNO","KOFPoGI","KOFEcGI","KOFSoGI","ISCED5␣
 ↪Percentage","ISCED6 Percentage","ISCED7 Percentage","ISCED8 Percentage",
    "top_50_count","top_100_count","top_500_count","top_1000_count"];
std= df[variables_ofstd];
scaler = StandardScaler(); # creating object
fitted = scaler.fit(std);
df_std  = pd.DataFrame(fitted.transform(std))
```

[4]:

```
# Elbow method
np.random.seed(1)
from sklearn.cluster import KMeans
def wcss(x, kmax):#wcss Function: The wcss function calculates the␣
 ↪within-cluster sum
#of squares (WCSS) for different numbers of clusters.
    wcss_s = [] #wcss_s: This list will store the WCSS values for different␣
 ↪numbers of clusters.
```

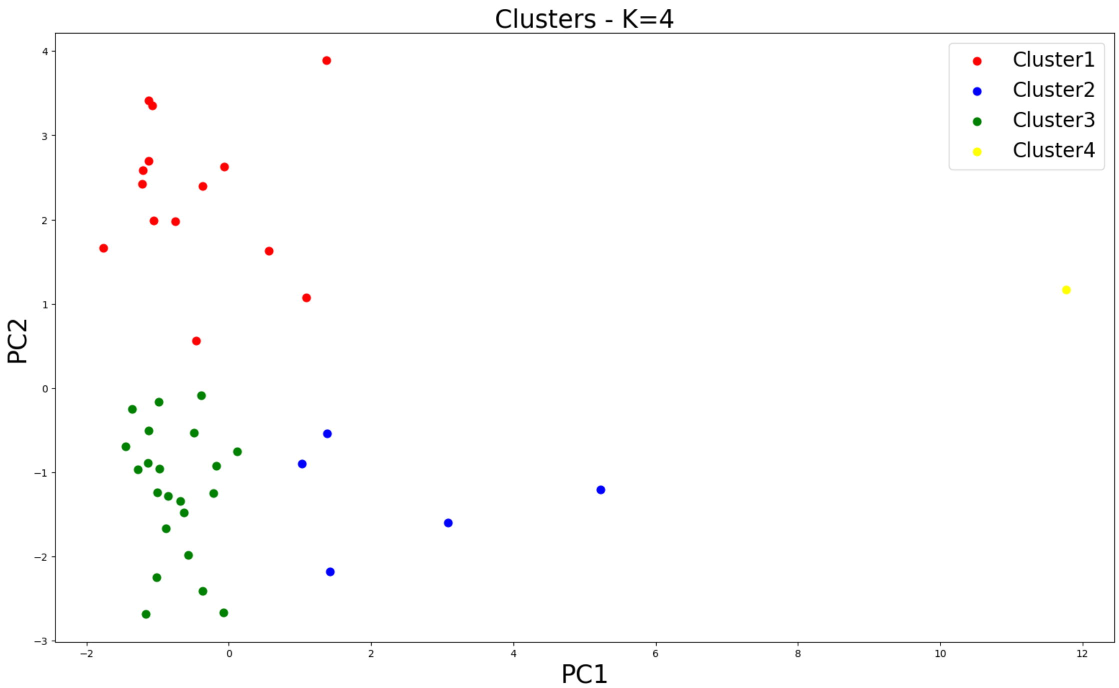
```
    for k in range(2, kmax + 1):
        kmeans = KMeans(n_clusters = k);
        kmeans.fit(x);
        wcss_s.append(kmeans.inertia_);# sample distances to closest cluster␣
 ↪center
    return wcss_s
# Plot
from matplotlib import pyplot as plt
fig = plt.figure(figsize = (19,11));
ax = fig.add_subplot(1,1,1);
kmax = 10; # maximum number of clusters
ax.plot(range(2, kmax + 1), wcss(df_std, kmax));
ax.tick_params(axis="both", which="major", labelsize=20);
ax.set_xlabel("Number of clusters", fontsize = 25);
ax.set_ylabel("Sum of squared error", fontsize = 25);
ax.set_title("Sum of squared error by number of clusters", fontsize = 25);
plt.show();
#Silihouse score
np.random.seed(100)
def Silhouette(x, kmax):
    sil = []
    for k in range(2, kmax+1):
        kmeans = KMeans(n_clusters = k).fit(x)
        sil.append(silhouette_score(x, kmeans.labels_, metric = "euclidean"))
    return sil
# Plot
fig = plt.figure(figsize = (19,11));
ax = fig.add_subplot(1,1,1);
ax.plot(range(2,kmax+1) , Silhouette(df_std,kmax));
ax.tick_params(axis="both", which="major", labelsize=20);
ax.set_xlabel("Number of clusters", fontsize = 25);
ax.set_ylabel("Silhouette score", fontsize = 25);
ax.set_title("Silhouette score by number of clusters", fontsize = 25);
ax.xaxis.set_major_locator(MaxNLocator(integer=True)) # to force intergers in␣
 ↪x-axis
plt.show();
```



It is not quite intuitive where the elbow effect happens, but it could be at K=3, 4, or 5 since the SSE decreased quite significantly with respect to lower K values. For K values greater than 5, the SSE decreases, but not as dramatically. The Silhouette scores plot favors K=2 and 4, but the SSEs for

K=2 are too high in the Elbow method. Therefore, we propose K=4(four clusters for the dataset). We can reduce the dimensionality of the data via PCA for visual inspection(See later parts).

[5]:
```python
# We now perform visiual inspection via reducing dimesionality (PCA)
from sklearn.decomposition import PCA
pca = PCA(n_components=2);# First two components
principalComponents = pca.fit_transform(df_std);
PCs = pd.DataFrame(data = principalComponents, columns = ["PC1", "PC2"]);
kmeans = KMeans(n_clusters = 4, init = "k-means++", random_state = 42);
y_kmeans = kmeans.fit_predict(df_std); # predictions of clusters
# Plotting PCs
fig = plt.figure(figsize = (19,11));
ax = fig.add_subplot(1,1,1);
plt.scatter(PCs.iloc[y_kmeans == 0, 0], PCs.iloc[y_kmeans == 0, 1], s=60,
c="red", label = "Cluster1");
plt.scatter(PCs.iloc[y_kmeans == 1, 0], PCs.iloc[y_kmeans == 1, 1], s=60,
c="blue", label = "Cluster2");
plt.scatter(PCs.iloc[y_kmeans == 2, 0], PCs.iloc[y_kmeans == 2, 1], s=60,
c="green", label = "Cluster3");
plt.scatter(PCs.iloc[y_kmeans == 3, 0], PCs.iloc[y_kmeans == 3, 1], s=60,
c="yellow", label = "Cluster4");
plt.xlabel("PC1", fontsize = 25);
plt.ylabel("PC2", fontsize = 25);
ax.set_title("Clusters - K=4", fontsize = 25);
plt.legend(fontsize = 20);
plt.show();
# Total variability explained by first two
print(" The variability explained by first two principal components is " +␣
↪str(np.sum(pca.explained_variance_ratio_)*100) + "%")
```




```
The variability explained by first two principal components is
64.97013208955245%
```

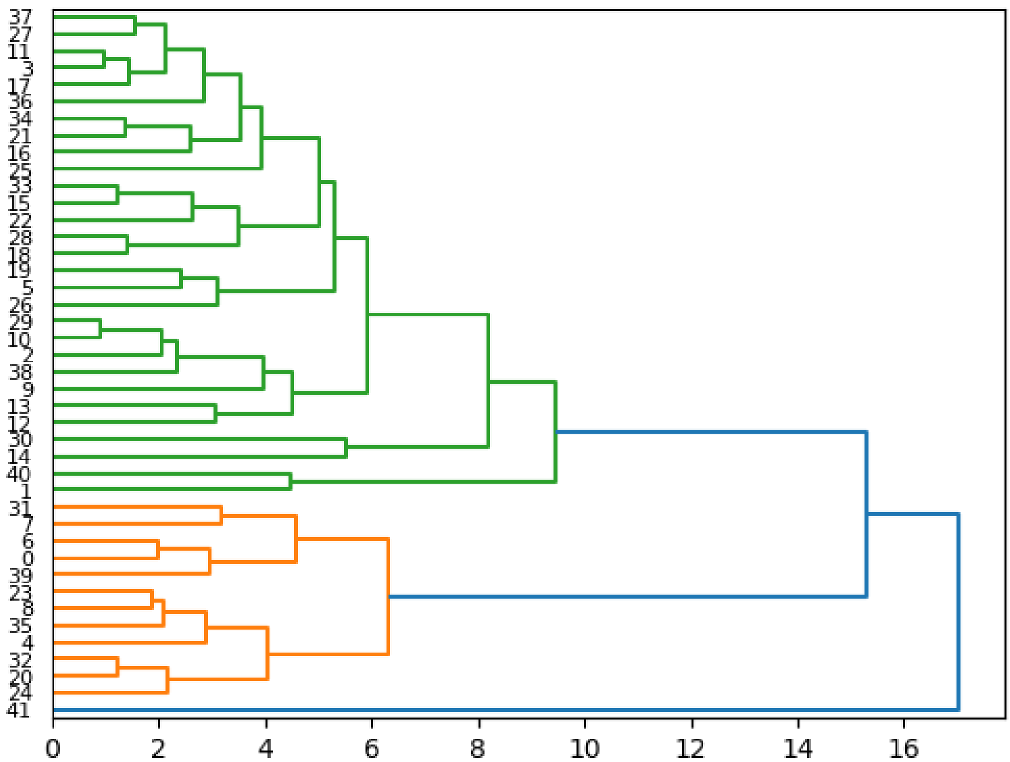
[6]:
```
# print out countries in differnt clusters.
df["Cluster_Kmean"] = pd.DataFrame(y_kmeans);
print("Cluster 1:\n", list(df["country_x"][(df["Cluster_Kmean"]==0)]));
print("Cluster 2:\n", list(df["country_x"][(df["Cluster_Kmean"]==1)]));
print("Cluster 3:\n", list(df["country_x"][(df["Cluster_Kmean"]==2)]));
print("Cluster 4:\n", list(df["country_x"][(df["Cluster_Kmean"]==3)]));
```

```
Cluster 1:
 ['Argentina', 'Brazil', 'Chile', 'China', 'Colombia', 'Japan', 'Kazakhstan',
'Malaysia', 'Mexico', 'Mongolia', 'Russia', 'Saudi Arabia', 'South Africa',
'Turkey']
Cluster 2:
 ['Australia', 'Canada', 'France', 'Germany', 'United Kingdom']
Cluster 3:
 ['Austria', 'Belgium', 'Cyprus', 'Czech Republic', 'Denmark', 'Hong Kong',
'Hungary', 'Iceland', 'Ireland', 'Italy', 'Latvia', 'Netherlands', 'New
Zealand', 'Norway', 'Poland', 'Portugal', 'Qatar', 'Slovak Republic',
'Slovenia', 'Spain', 'Sweden', 'Switzerland']
Cluster 4:
 ['USA']
```

We could observe a clear separation between the clusters, and there is no overlap in the figure. It seems that $K = 4$ also represents the definition of the clusters quite well.The first 2 principal *components explain approx 65% of the variability of the data. More importantly, the 4 clusters are well-defined in the PC1 and PC2 scatter plots.*

3 Question3 c

[7]:
```
#Q3C agglomerative cluster analysis
from scipy.cluster.hierarchy import dendrogram, linkage;
dendrogram(linkage(df_std, method="ward"), orientation = "right",# Generating␣
 ↪dendrogram
labels = None);
plt.show()
```



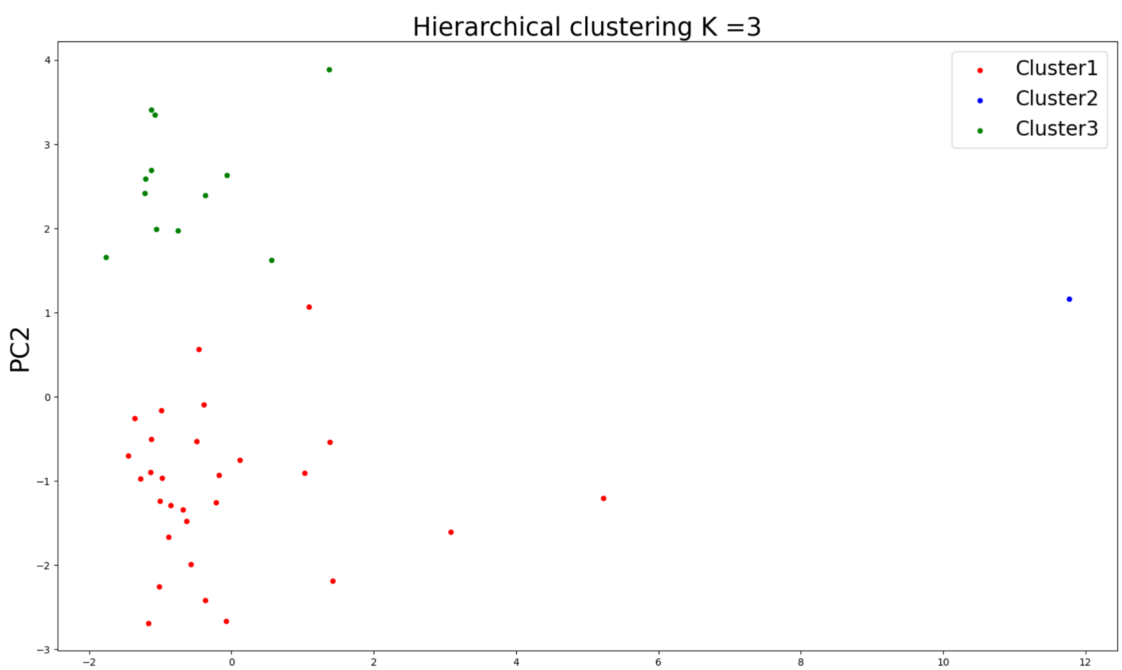
The largest distance can be found between approximately 9 and 15, generating 3 clusters(Vertical line at 9).Hence, we propose 3 clusters for this method.

[8]:
```
#We now perform visiual inspection via reducing dimesionality (PCA)
from sklearn.cluster import AgglomerativeClustering
model = AgglomerativeClustering(n_clusters=3, linkage="ward",
compute_distances = True);
model.fit(df_std);
df["Cluster_Agg"] = pd.DataFrame(model.labels_);
clusters3 = model.labels_;
```

```
pca = PCA(n_components=2);
principalComponents = pca.fit_transform(df_std);
print("Variability explained by first 2 PCs: ", round(np.sum(pca.
 ↪explained_variance_ratio_),2))
PCs = pd.DataFrame(data = principalComponents, columns = ["PC1", "PC2"]);

# Plotting PCs
fig = plt.figure(figsize = (19,11));
ax = fig.add_subplot(1,1,1);
plt.scatter(PCs.iloc[clusters3 == 0, 0], PCs.iloc[clusters3 == 0, 1], s=20,␣
 ↪c="red", label = "Cluster1");
plt.scatter(PCs.iloc[clusters3 == 1, 0], PCs.iloc[clusters3 == 1, 1], s=20,␣
 ↪c="blue", label = "Cluster2");
plt.scatter(PCs.iloc[clusters3 == 2, 0], PCs.iloc[clusters3 == 2, 1], s=20,␣
 ↪c="green", label = "Cluster3");
plt.ylabel("PC2", fontsize = 25);
ax.set_title("Hierarchical clustering K =3", fontsize = 25);
plt.legend(fontsize = 20);
plt.show();
```

```
Variability explained by first 2 PCs:  0.65
```



Since we can not plot all the variables simultaneously, we would reduce the dimensionality of the dataset through PCA. The first 2 principal components explain 65% of the variability of the data. More importantly, the 3 clusters are well-defined in the PC1 and PC2 scatter plots.

4 Question3 d

[9]:
```python
# print out countries in differnt clusters to describe.
print("Cluster 1:\n", list(df["country_x"][(df["Cluster_Agg"]==0)]));
print("Cluster 2:\n", list(df["country_x"][(df["Cluster_Agg"]==1)]));
print("Cluster 3:\n", list(df["country_x"][(df["Cluster_Agg"]==2)]));
```

```
Cluster 1:
 ['Australia', 'Austria', 'Belgium', 'Canada', 'Cyprus', 'Czech Republic',
'Denmark', 'France', 'Germany', 'Hong Kong', 'Hungary', 'Iceland', 'Ireland',
'Italy', 'Japan', 'Latvia', 'Malaysia', 'Netherlands', 'New Zealand', 'Norway',
'Poland', 'Portugal', 'Qatar', 'Slovak Republic', 'Slovenia', 'Spain', 'Sweden',
'Switzerland', 'United Kingdom']
Cluster 2:
 ['USA']
Cluster 3:
 ['Argentina', 'Brazil', 'Chile', 'China', 'Colombia', 'Kazakhstan', 'Mexico',
'Mongolia', 'Russia', 'Saudi Arabia', 'South Africa', 'Turkey']
```

The analysis of K-mean and agglomerative cluster analysis suggested a slightly different number of clusters (4 and 3), indicating that globalization of the country and education system are indeed quite complex as there is a lot factors influencing them. Interestingly, the USA itself was identified as one cluster in this case, meaning that the USA has unique characteristics not shared by other countries, as well as the dominance of the USA, such as a higher globalization index, international students' mobility, economic globalization index, etc. Moreover, Countries like Australia, Canada and France were in one cluster alone by the K-mean method; then they were put in the same cluster as countries like Spain and Japan. This potentially implies that those emerging countries are getting better and better at their education system and attracting more international students to come to their country.This agrees with what we have from the scatter plot, as we could observe a point on the far right end(USA).