Obtención de datos estadisticos

In [22]:
```
import pandas as pd
import seaborn as sb
import numpy as np; np.random.seed(0)
import matplotlib.pyplot as plt

#import matplotlib.pyplot as plt
import seaborn as sb
import sklearn
from sklearn.cluster import KMeans
from sklearn.metrics import pairwise_distances_argmin_min
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.preprocessing import scale
import sklearn.metrics as sm
from sklearn import datasets
%matplotlib inline
from mpl_toolkits.mplot3d import Axes3D

import os
nb_path = os.path.abspath("Datos_estadisticos_national_MPI_Rural.ipynb")
gdp_csv = os.path.join(os.path.dirname(nb_path), "Datasets/MPI_national.c
datos = pd.read_csv(gdp_csv, engine='python')

from matplotlib import cm
plt.rcParams["figure.figsize"] = (30,22)
plt.style.use("ggplot")
```

Out[22]: (102, 8)

In [23]:
```
datos
```

Out[23]:

	ISO	Country	MPI Urban	Headcount Ratio Urban	Intensity of Deprivation Urban	MPI Rural	Headcount Ratio Rural	Intensity of Deprivation Rural
0	KAZ	Kazakhstan	0.000	0.0	33.3	0.000	0.09	33.3
1	SRB	Serbia	0.000	0.1	41.4	0.002	0.50	40.3
2	KGZ	Kyrgyzstan	0.000	0.1	40.2	0.003	0.70	37.1
3	TUN	Tunisia	0.000	0.1	35.6	0.012	3.18	38.7
4	ARM	Armenia	0.001	0.2	33.3	0.001	0.39	36.9
...	...	...	...	...	...	...	...	...
97	CAF	Central African Republic	0.289	58.2	49.7	0.519	89.79	57.8
98	LBR	Liberia	0.290	60.5	48.0	0.481	84.86	56.6
99	SOM	Somalia	0.293	55.9	52.4	0.651	96.92	67.2

In [24]:
```
filas=len(datos)
filas
```

Out[24]:
```
102
```

In [25]:
```
tipo_datos = pd.read_csv(gdp.csv)
```

In [26]:
```
tipo_datos.dtypes
```

Out[26]:
```
ISO                                 object
Country                             object
MPI Urban                          float64
Headcount Ratio Urban              float64
Intensity of Deprivation Urban     float64
MPI Rural                          float64
Headcount Ratio Rural              float64
Intensity of Deprivation Rural     float64
dtype: object
```

In [27]:
```
datos['MPI Urban'].describe()
```

Out[27]:
```
count    102.000000
mean       0.078343
std        0.093693
min        0.000000
25%        0.007250
50%        0.034500
75%        0.125750
max        0.459000
Name: MPI Urban, dtype: float64
```

In [28]:
```
datos['Headcount Ratio Urban'].describe()
```

Out[28]:
```
count    102.000000
mean      16.809804
std       18.498448
min        0.000000
25%        1.950000
50%        8.400000
75%       27.575000
max       82.500000
Name: Headcount Ratio Urban, dtype: float64
```

In [29]:
```
datos['Intensity of Deprivation Urban'].describe()
```

Out[29]:
```
count    102.000000
mean      41.678431
std        5.135908
min       33.300000
25%       37.200000
50%       41.550000
75%       45.675000
max       55.700000
Name: Intensity of Deprivation Urban, dtype: float64
```

Segunda ´Parte

In [30]:
```
datos['MPI Rural'].describe()
```

Out[30]:
```
count    102.000000
mean       0.214676
std        0.201208
min        0.000000
25%        0.025000
50%        0.160000
75%        0.384500
max        0.669000
Name: MPI Rural, dtype: float64
```

In [31]:
```
datos['MPI Rural'].median()
```

Out[31]:
```
0.16
```

In [32]:
```
datos['Headcount Ratio Rural'].describe()
```

Out[32]:
```
count    102.000000
mean      40.036176
std       33.270714
min        0.090000
25%        6.745000
50%       36.055000
75%       70.130000
max       96.920000
Name: Headcount Ratio Rural, dtype: float64
```

In [33]:
```
datos['Headcount Ratio Rural'].median()
```

Out[33]:
```
36.055
```

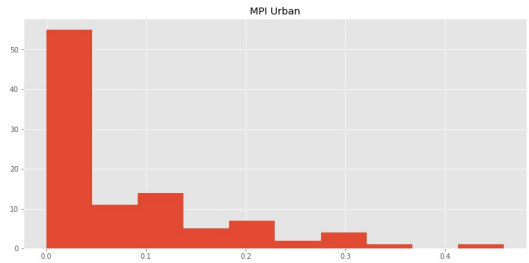
In [34]:
```
datos['Intensity of Deprivation Rural'].describe()
```

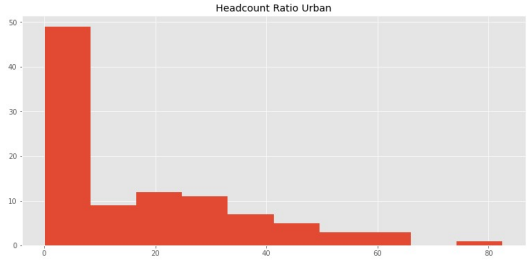
Out[34]:
```
count    102.000000
mean      46.824510
std        8.783191
min       33.300000
25%       40.225000
50%       44.800000
75%       53.425000
max       69.500000
Name: Intensity of Deprivation Rural, dtype: float64
```

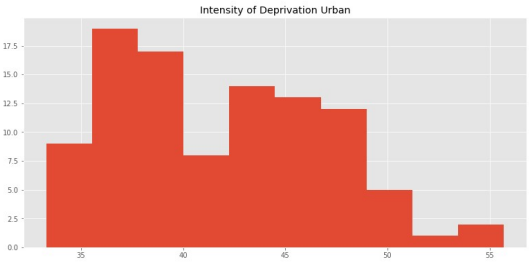
In [35]:
```
datos['Intensity of Deprivation Rural'].median()
```

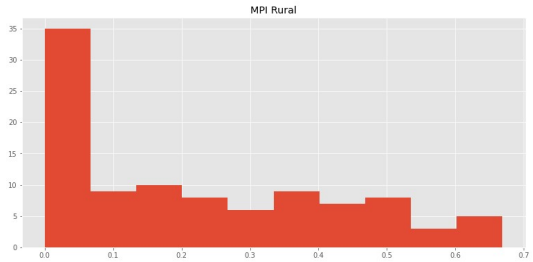
Out[35]:
```
44.8
```

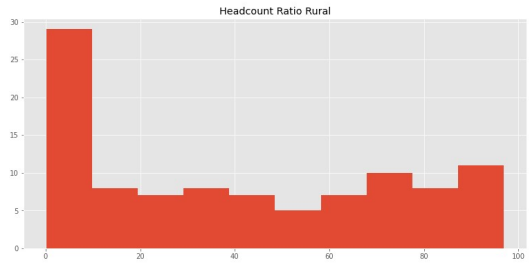
In [36]:
```
datos.drop([0,1]).hist()
plt.show()
```

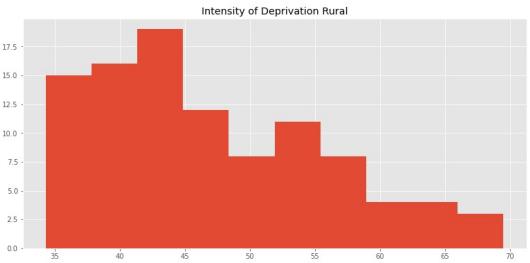




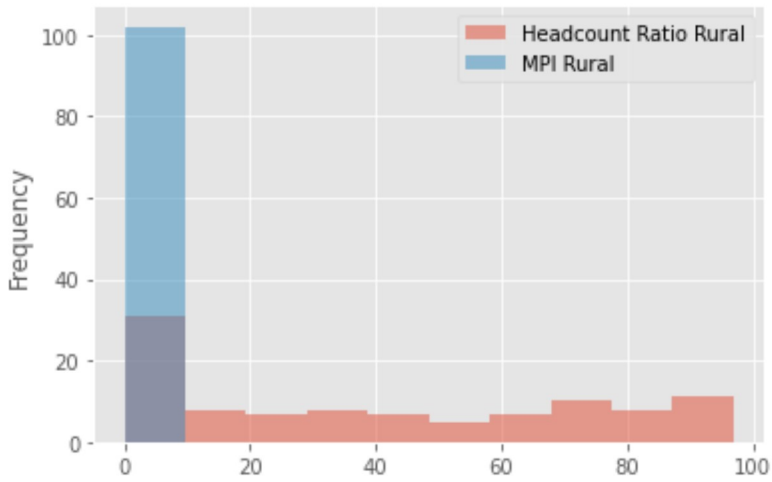




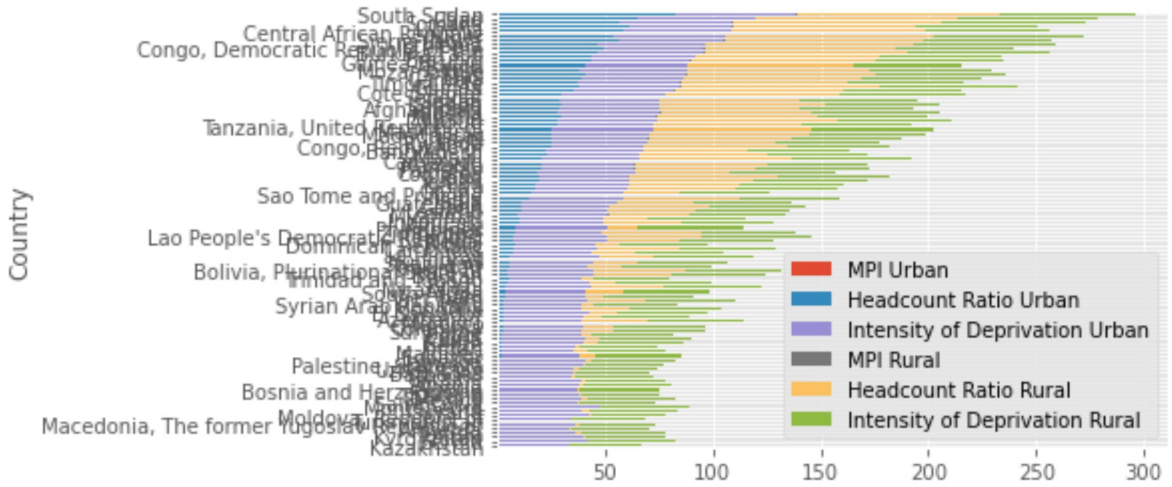




In [37]:
```
%matplotlib inline
datos[["Headcount Ratio Rural", "MPI Rural"]].plot.hist(bins=10,alpha=0.5
plt.show()
```

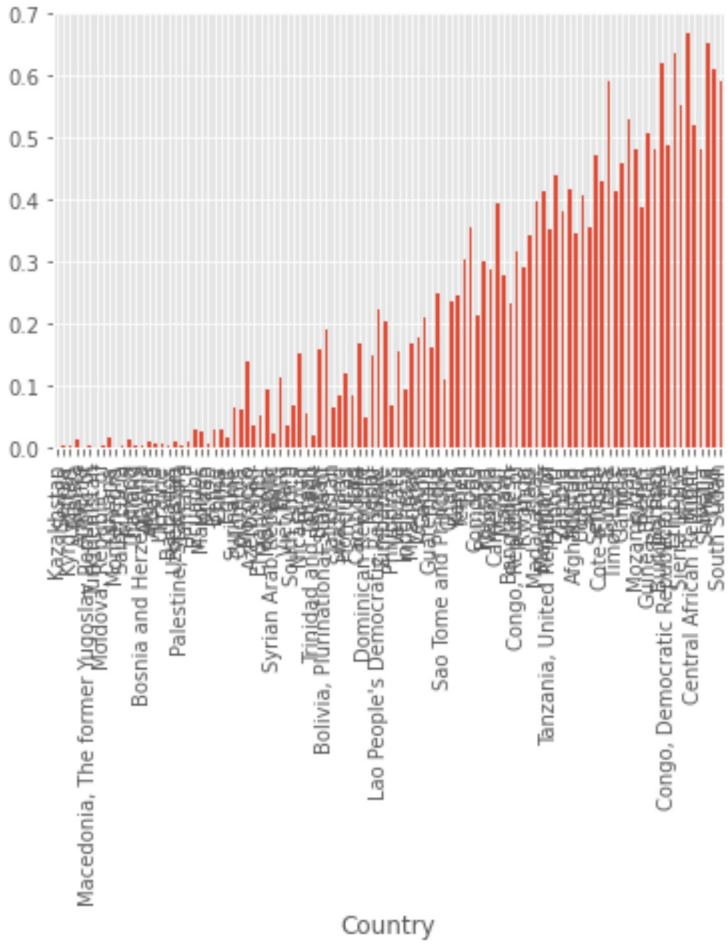


In [38]:

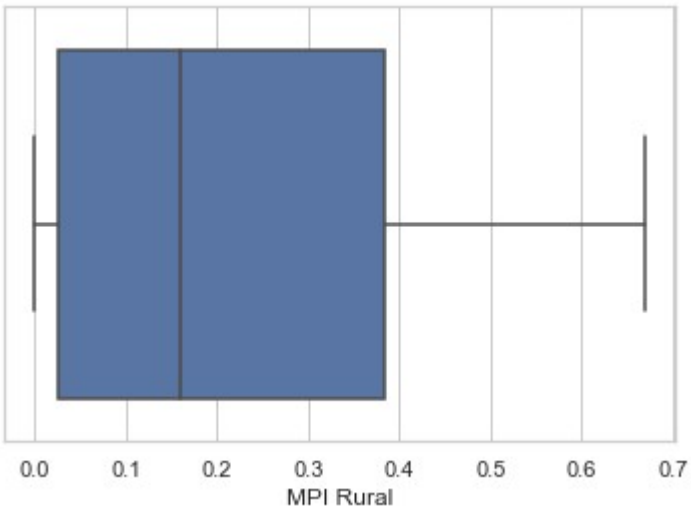

Country
MPI Urban
Headcount Ratio Urban
Intensity of Deprivation Urban
MPI Rural
Headcount Ratio Rural
Intensity of Deprivation Rural
50
100
150
200
250
300

In [ ]:

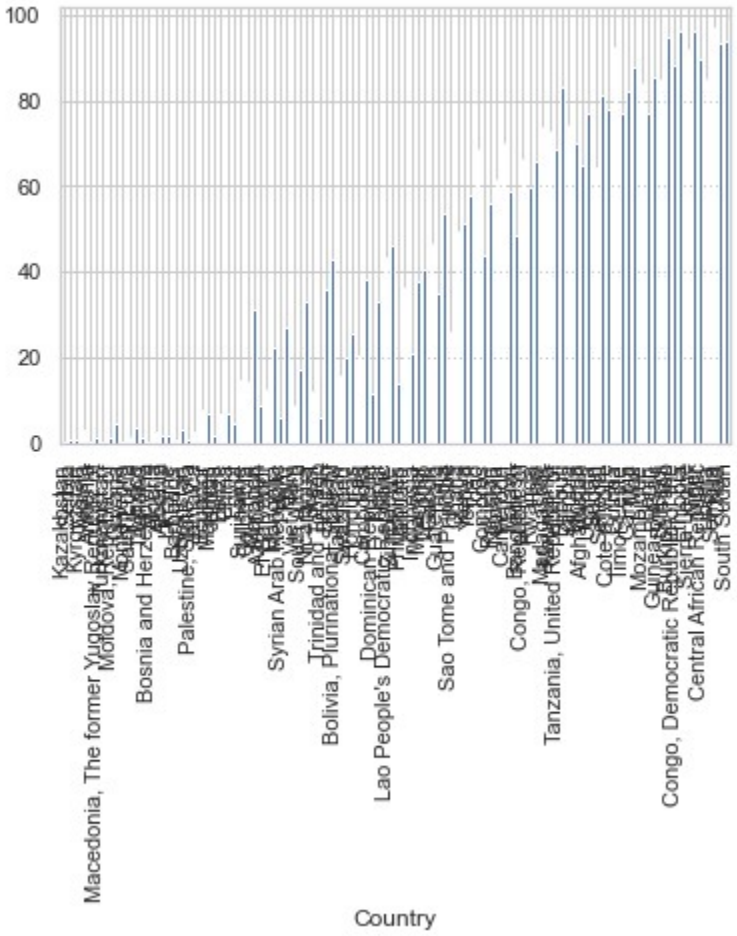
In [39]:


0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Country

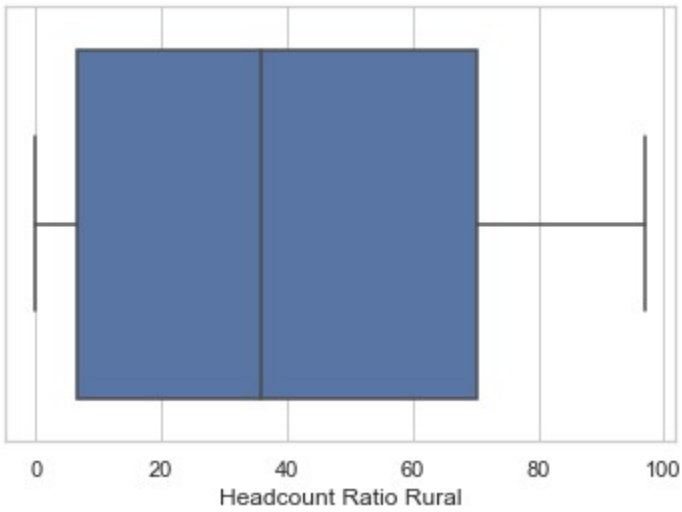
In [40]:
```
sb.set_theme(style="whitegrid")
```



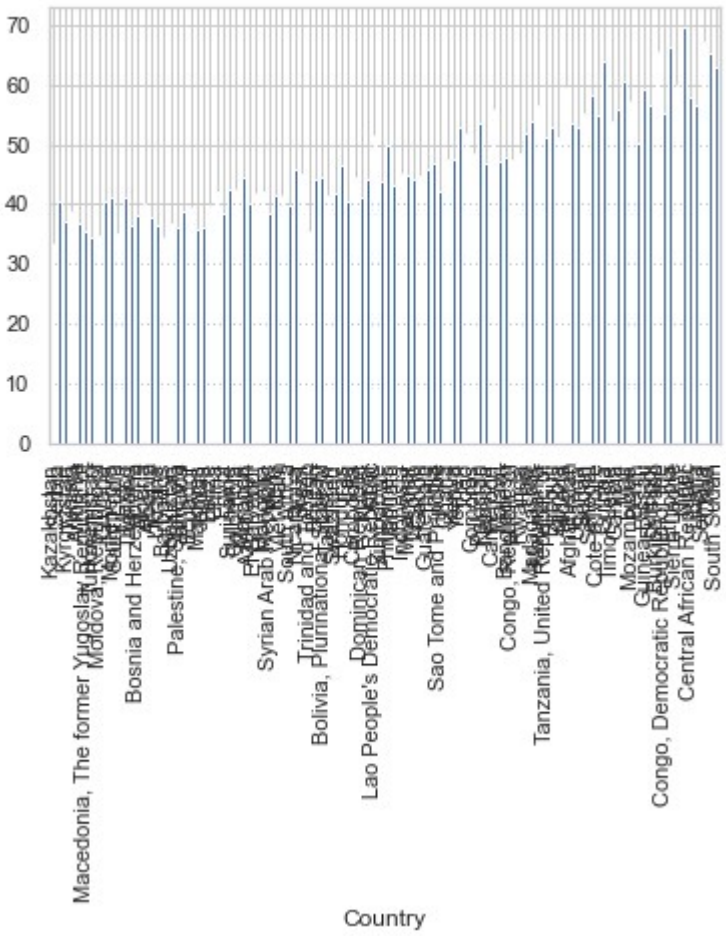
In [41]:



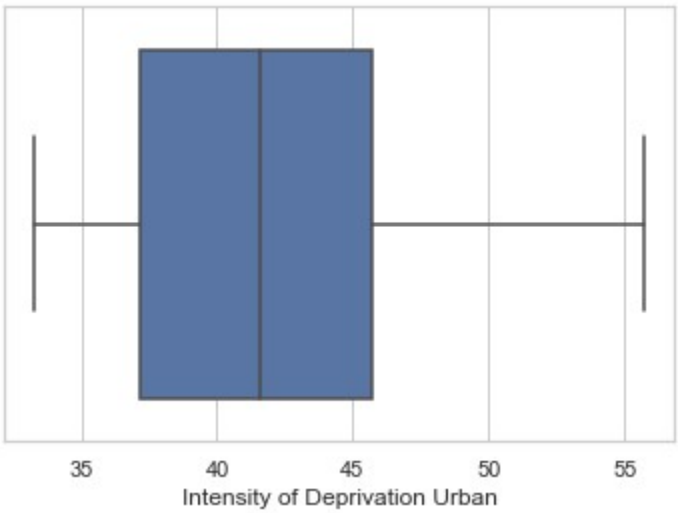
In [42]:
```
sb.set_theme(style="whitegrid")
```



In [43]:



In [44]:
```
sb.set_theme(style="whitegrid")
```



In [45]:
```
datos
```

Out[45]:

	ISO	Country	MPI Urban	Headcount Ratio Urban	Intensity of Deprivation Urban	MPI Rural	Headcount Ratio Rural	Intensity of Deprivation Rural
0	KAZ	Kazakhstan	0.000	0.0	33.3	0.000	0.09	33.3
1	SRB	Serbia	0.000	0.1	41.4	0.002	0.50	40.3
2	KGZ	Kyrgyzstan	0.000	0.1	40.2	0.003	0.70	37.1
3	TUN	Tunisia	0.000	0.1	35.6	0.012	3.18	38.7
4	ARM	Armenia	0.001	0.2	33.3	0.001	0.39	36.9
...	...	...	...	...	...	...	...	...
97	CAF	Central African Republic	0.289	58.2	49.7	0.519	89.79	57.8
98	LBR	Liberia	0.290	60.5	48.0	0.481	84.86	56.6
99	SOM	Somalia	0.293	55.9	52.4	0.651	96.92	67.2
100	TCD	Chad	0.351	64.8	54.1	0.609	93.41	65.2
101	SSD	South Sudan	0.459	82.5	55.7	0.591	94.00	62.8

102 rows × 8 columns

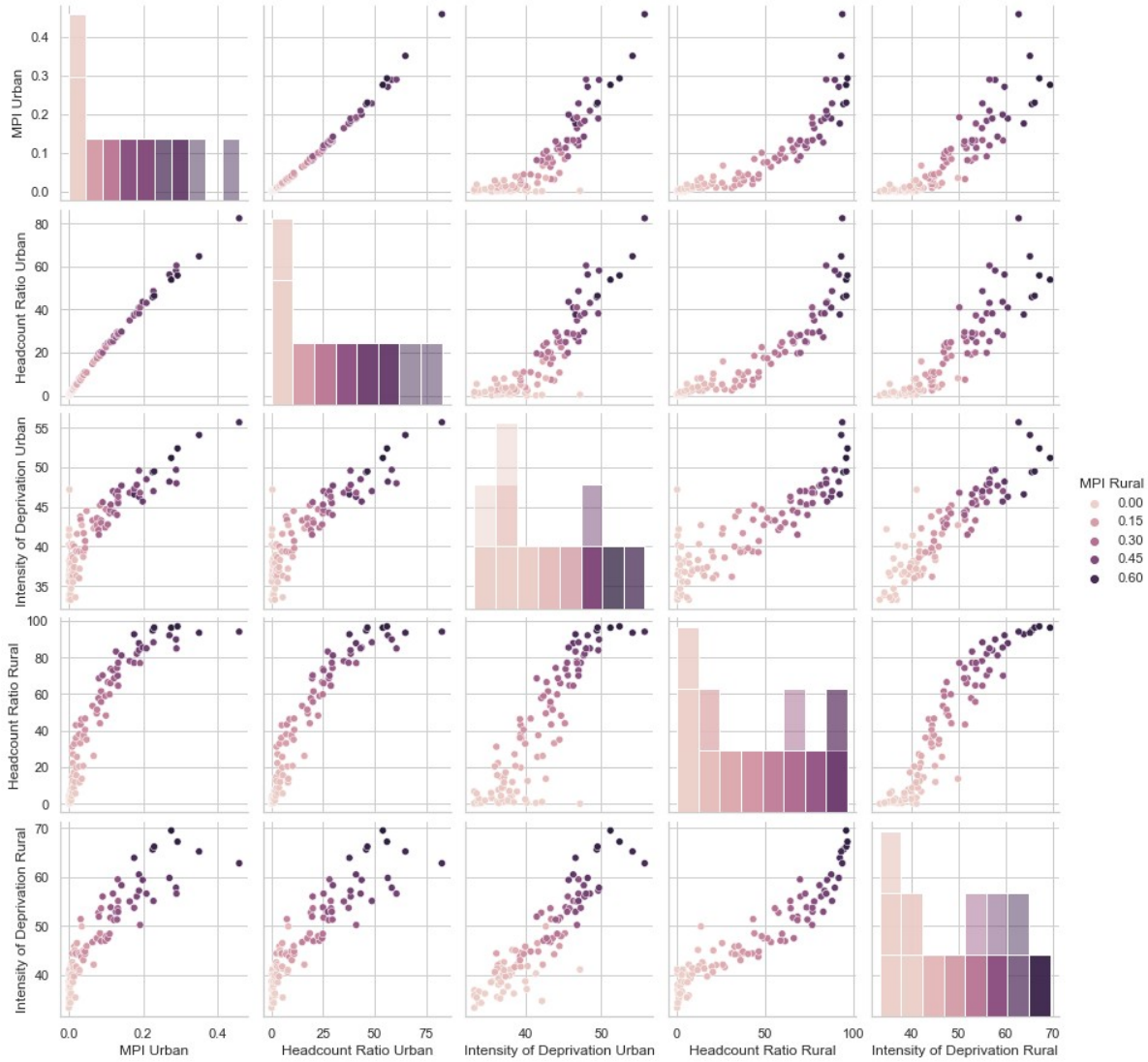
In [46]:
```
dataframe = pd.read_csv(gdp_csv)
dataframe.head()
```

Out[46]:

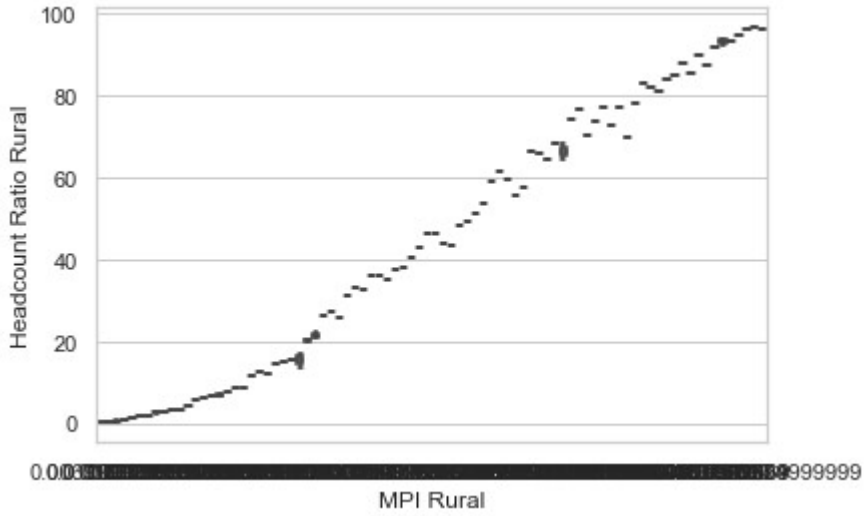
	ISO	Country	MPI Urban	Headcount Ratio Urban	Intensity of Deprivation Urban	MPI Rural	Headcount Ratio Rural	Intensity of Deprivation Rural
0	KAZ	Kazakhstan	0.000	0.0	33.3	0.000	0.09	33.3
1	SRB	Serbia	0.000	0.1	41.4	0.002	0.50	40.3

	ISO	Country	MPI Urban	Headcount Ratio Urban	Intensity of Deprivation Urban	MPI Rural	Headcount Ratio Rural	Intensity of Deprivation Rural
2	KGZ	Kyrgyzstan	0.000	0.1	40.2	0.003	0.70	37.1

In [47]:



In [48]:

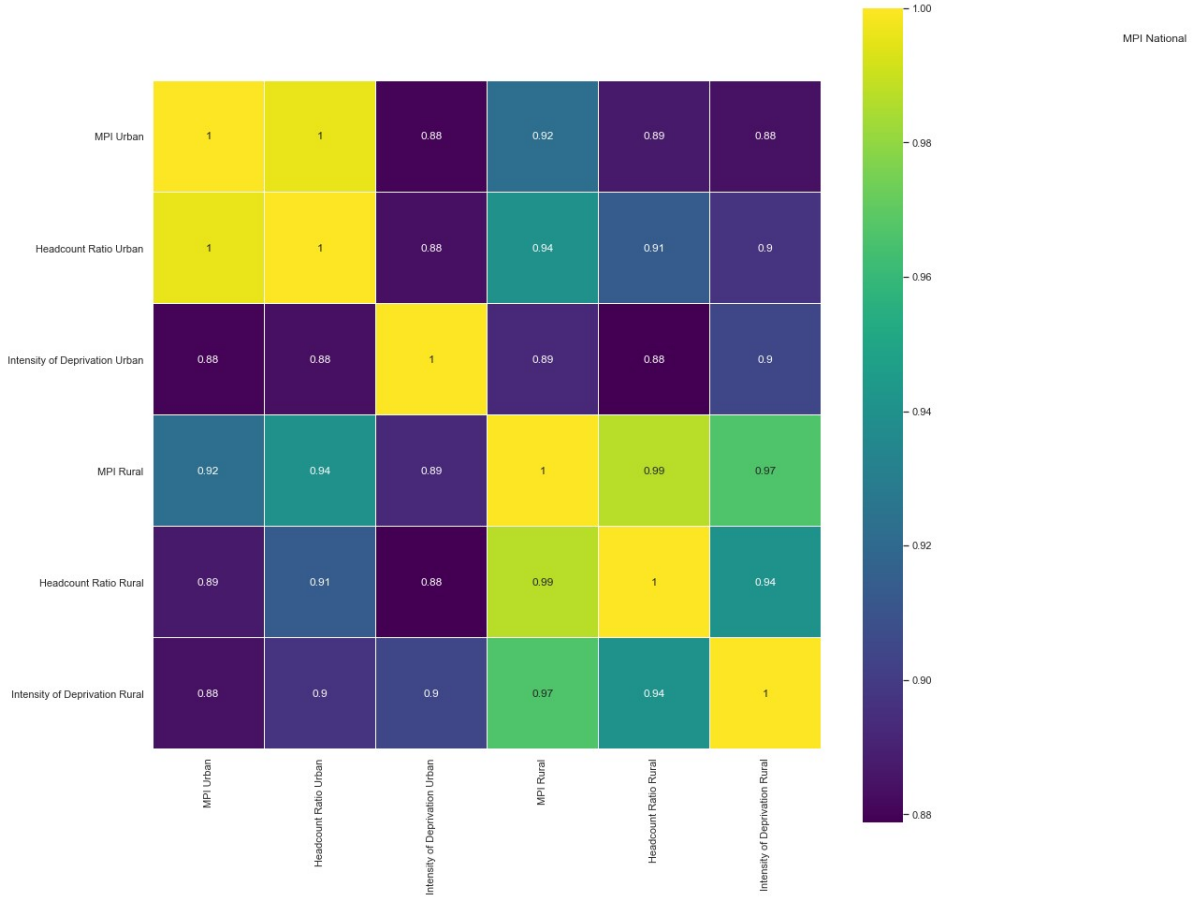


In [49]:

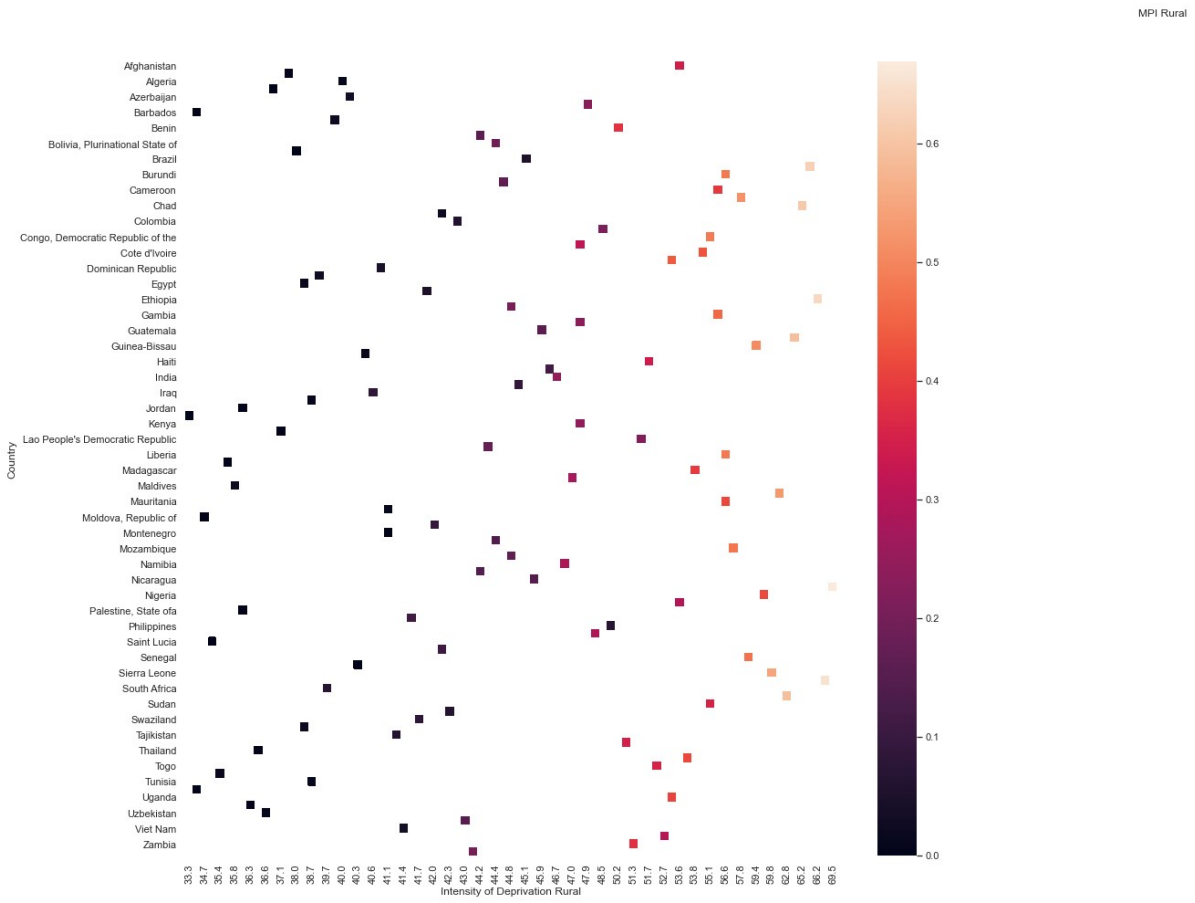
Out[49]:

	MPI Urban	Headcount Ratio Urban	Intensity of Deprivation Urban	MPI Rural	Headcount Ratio Rural	Intensity of Deprivation Rural
MPI Urban	1.000000	0.981233	0.705851	0.856086	0.854226	0.775371
Headcount Ratio Urban	0.981233	1.000000	0.684539	0.847173	0.847278	0.760670
Intensity of Deprivation Urban	0.705851	0.684539	1.000000	0.696237	0.686322	0.744445
MPI Rural	0.856086	0.847173	0.696237	1.000000	0.975488	0.843500
Headcount Ratio Rural	0.854226	0.847278	0.686322	0.975488	1.000000	0.817546
Intensity of Deprivation Rural	0.775371	0.760670	0.744445	0.843500	0.817546	1.000000

In [99]:
```
datos1=datos.drop(columns=['ISO','Country'])
colormap = plt.cm.viridis
plt.figure(figsize=(16,16))
plt.title("MPI National", y=1.05, x=1.5)
sb.heatmap(datos1.astype(float).corr(),linewidths=0.1,vmax=1.0, square=Tr
```



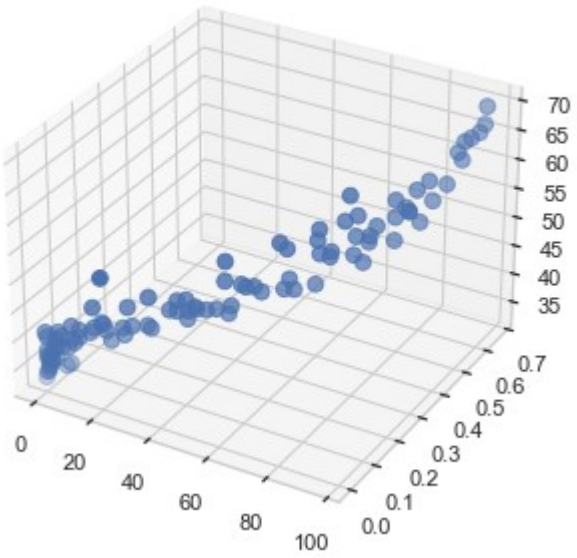
In [106]:
```
datos3 = datos.pivot("Country","Intensity of Deprivation Rural","MPI Rura
plt.figure(figsize=(16,16))
plt.title("MPI Rural", y=1.05, x=1.5)
```



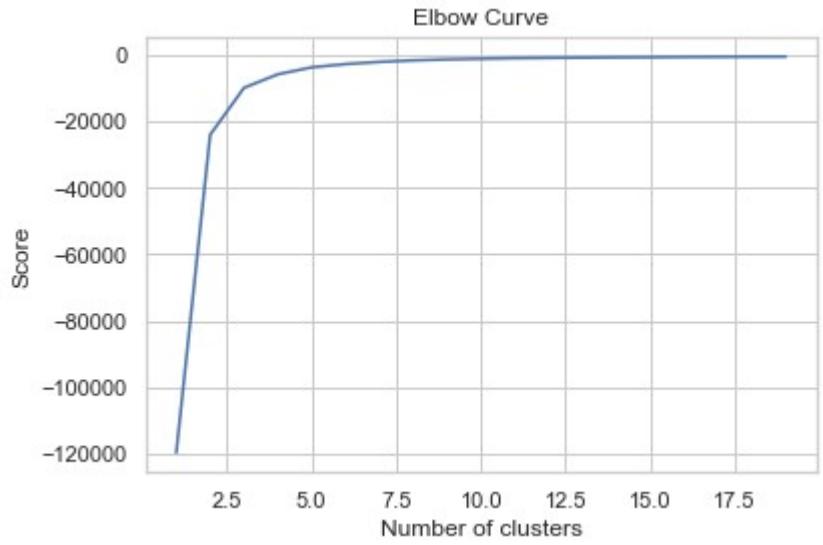
In [76]:
```
X = np.array(dataframe[["Headcount Ratio Rural","MPI Rural","Intensity of
y = np.array([x for x in range(len(dataframe.index))])
```

Out[76]: (102, 3)

In [78]:
```
fig = plt.figure()
ax = Axes3D(fig)
```



In [82]:
```
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
plt.plot(Nc,score)
plt.xlabel('Number of clusters')
plt.ylabel('Score')
plt.title('Elbow Curve')
```

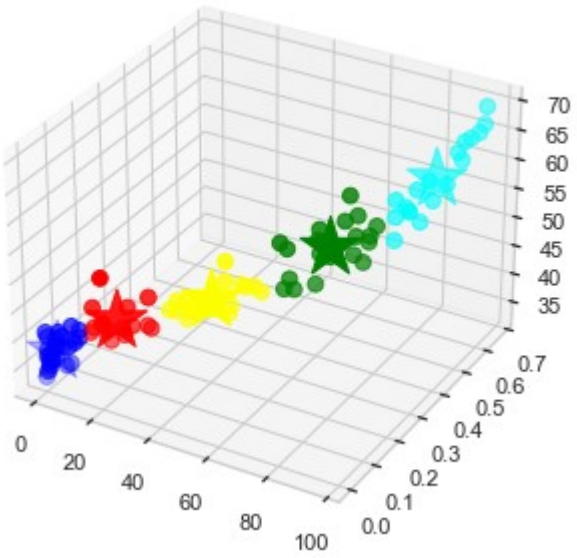


In [86]:
```
kmeans= KMeans(n_clusters=5).fit(X)
centroids = kmeans.cluster_centers_
```

```
[[1.82766667e+01 7.82000000e-02 4.28800000e+01]
 [6.81100000e+01 3.57950000e-01 5.24300000e+01]]
```

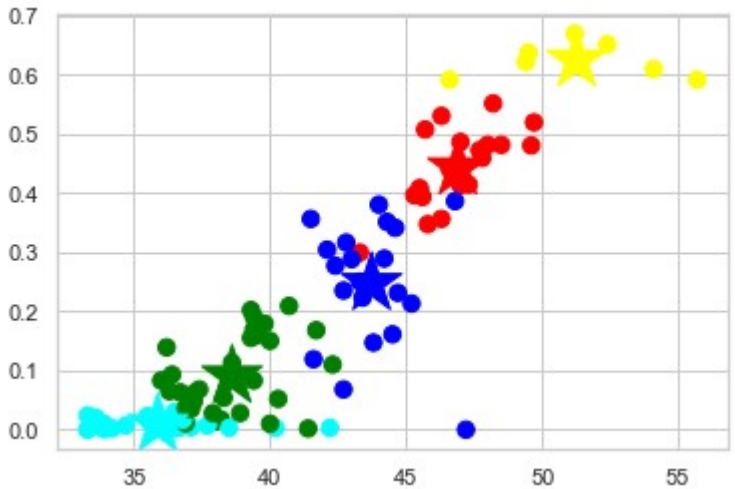
In [87]:
```
labels = kmeans.predict(X)
C = kmeans.cluster_centers_
colores=['red','green','blue','cyan','yellow']
asignar=[]
for row in labels:
    asignar.append(colores[row])

fig=plt.figure()
ax = Axes3D(fig)
ax.scatter(X[:,0],X[:,1],X[:,2], c=asignar, s=60)
```



In [56]:
```
f1= dataframe["Intensity of Deprivation Urban"].values
f2= dataframe['MPI Rural'].values

plt.scatter(f1,f2,c=asignar,s=70)
plt.scatter(C[:,0],C[:,1], marker='*', c=colores, s=1000)
```



In [80]:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	1
1	0.03	1.00	0.06	1
2	0.00	0.00	0.00	1
3	0.05	1.00	0.09	1
4	0.00	0.00	0.00	1
5	0.00	0.00	0.00	1
6	0.00	0.00	0.00	1
7	0.00	0.00	0.00	1
8	0.00	0.00	0.00	1
9	0.00	0.00	0.00	1
10	0.00	0.00	0.00	1
11	0.00	0.00	0.00	1
12	0.00	0.00	0.00	1
13	0.00	0.00	0.00	1
14	0.00	0.00	0.00	1
15	0.00	0.00	0.00	1
16	0.00	0.00	0.00	1

In [ ]: