

Mapas de calor y boxplots.

Carga los datos usando tu lector de csv o con pandas. Es recomendable hacerlo con pandas. Verifica la cantidad de datos que tienes, las variables que contiene cada vector de datos e identifica el tipo de variables. Analiza las variables para saber que representa cada una y en que rangos se encuentran. Si la descripción del problema no te lo indica, utiliza el máximo y el mínimo para encontrarlo. Basándose en la media, mediana y desviación estándar de cada variable, que conclusiones puedes entregar de los datos. Realiza el análisis de las variables usando diagramas de cajas y bigotes, histogramas y mapas de calor.

Equipo:

Esteban López Alegría A01706956

Félix Javier Rojas Gallardo A01201946

Amanda María Real Núñez. A01367729

Preguntas:

- ¿Hay alguna variable que no aporta información?

En el análisis y creación de representaciones visuales de índice multidimensional de pobreza rural, que consiste en las columnas de MPI_Rural, Headcount_Ratio_Rural y Intensity_of_Deprivation_Rural, la variable que no nos aporte información necesaria fue la columna de ISO

- Si tuvieras que eliminar variables, ¿cuáles quitarías y por qué?

La variable principal es el índice de pobreza multidimensional (MPI), este es un índice compuesto, por lo cual las otras variables en nuestra base de datos son importantes para poder complementar y entender este índice. Por lo cual no sería conveniente eliminar ninguna variable. Afortunadamente las columnas que tenemos son las necesarias para poder ver el comportamiento del índice multidimensional de pobreza tanto urbana como rural, en las naciones que se encuentran en la base de datos, en lugar de eliminar alguna variable considero que podríamos sacar un mayor provecho de las columnas de ISO y Country si fueran variables de tipo float. Sin embargo el tener variables string son necesarias en la columna de los países, debido a que nos fueron útiles en las gráficas de barras.

- ¿Existen variables que tengan datos extraños?

Ninguna variable contenía datos anormales, debido a que los datos numéricos podían analizarse de manera óptima, al inicio del análisis modificamos nuestra base de datos en Excel para que no tuviera caracteres que nos fueran difíciles de utilizar en Jupyter, es por ello que la base no cuenta con acentos o caracteres especiales, todos los datos son numéricos, excepto por las primeras dos columnas que contienen texto (aun así son útiles).

- Si comparas las variables, ¿todas están en rangos similares? ¿Crees que esto afecte?

Los rangos de datos se encontraban dentro de un rango considerable, es decir la base contiene el mismo número de datos en todas las columnas, los datos numéricos de cada columna con datos tipo float, contienen las mismas unidades y por medio de la gráfica de caja y bigotes, logramos observar que

```

In [27]: 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

datos=pd.read_csv('MPI_national.csv')
from matplotlib import cm
plt.rcParams["figure.figsize"] = (40,22)
plt.style.use("ggplot")
datos.shape

```

Out[27]: (102, 8)

In [6]: datos

O...	ISO	Country	MPI_Urban	Headcount_Ratio_Urban	Intensity_of_Deprivation_Urban	MPI_Rural	He
0	KAZ	Kazakhstan	0.000	0.0	33.3	0.000	
1	SRB	Serbia	0.000	0.1	41.4	0.002	
2	KGZ	Kyrgyzstan	0.000	0.1	40.2	0.003	
3	TUN	Tunisia	0.000	0.1	35.6	0.012	
4	ARM	Armenia	0.001	0.2	33.3	0.001	
...	
97	CAF	Central African Republic	0.289	58.2	49.7	0.519	
98	LBR	Liberia	0.290	60.5	48.0	0.481	
99	SOM	Somalia	0.293	55.9	52.4	0.651	
100	TCD	Chad	0.351	64.8	54.1	0.609	
101	SSD	South Sudan	0.459	82.5	55.7	0.591	

102 rows × 8 columns

<  >

In [7]:

Out[7]:102

In [8]: tipo_datos = pd.read_csv ("MPI_national.csv")

In [22]: tipo_datos.dtypes

```
Out[22]: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 [25]: datos['MPI_Urban'].describe()

```
Out[25]: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 [23]: datos['Headcount_Ratio_Urban'].describe()

```
Out[23]: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 [24]: datos['Intensity_of_Deprivation_Urban'].describe()

```
Out[24]: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 [25]: datos['MPI_Rural'].describe()

```
Out[25]:count    102.000000
mean           0.214676
std            0.201208
```

```
In [33]: datos['MPI_Rural'].median()
```

```
Out[33]:0.16
```

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

```
Out[27]: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 [28]: datos['Headcount_Ratio_Rural'].median()
```

```
Out[28]:36.055
```

```
In [29]: datos['Intensity_of_Deprivation_Rural'].describe()
```

```
Out[29]: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
```

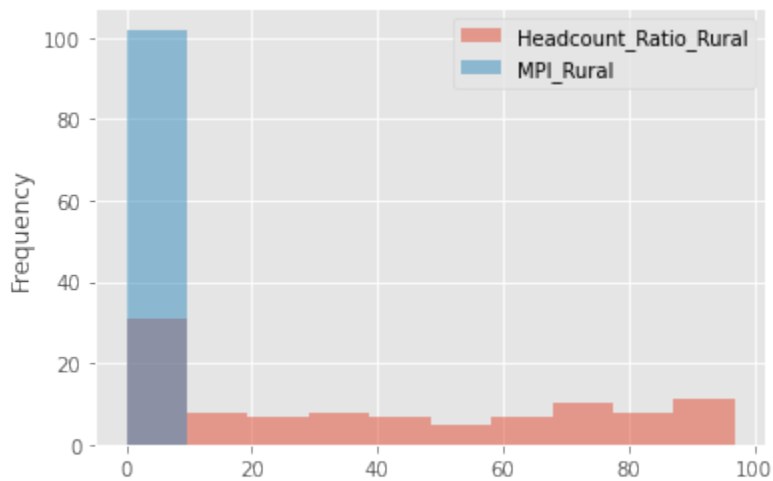
```
In [9]: datos['Intensity_of_Deprivation_Rural'].median()
```

```
Out[9]:44.8
```

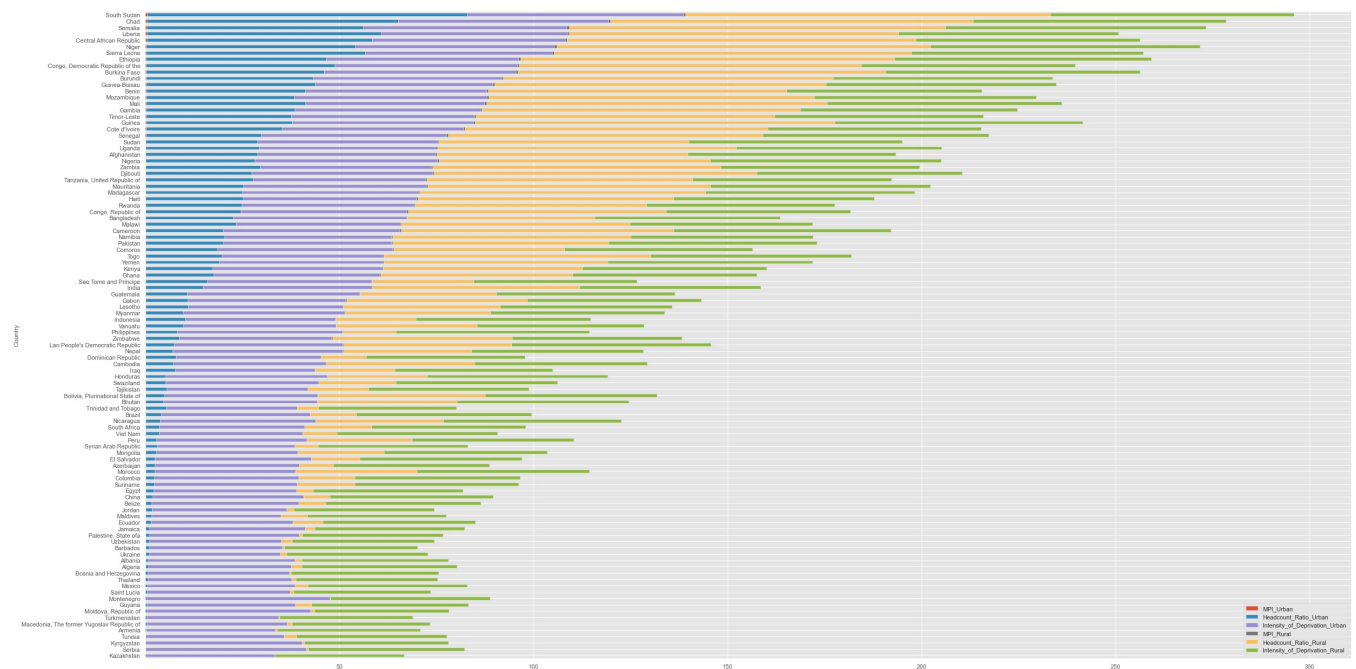
```
In [21]: datos.drop([0,1]).hist()
      plt.show()
```



```
In [2... %matplotlib inline
datos[["Headcount_Ratio_Rural", "MPI_Rural"]].plot.hist(bins=10,alpha=0.5)
plt.show()
```



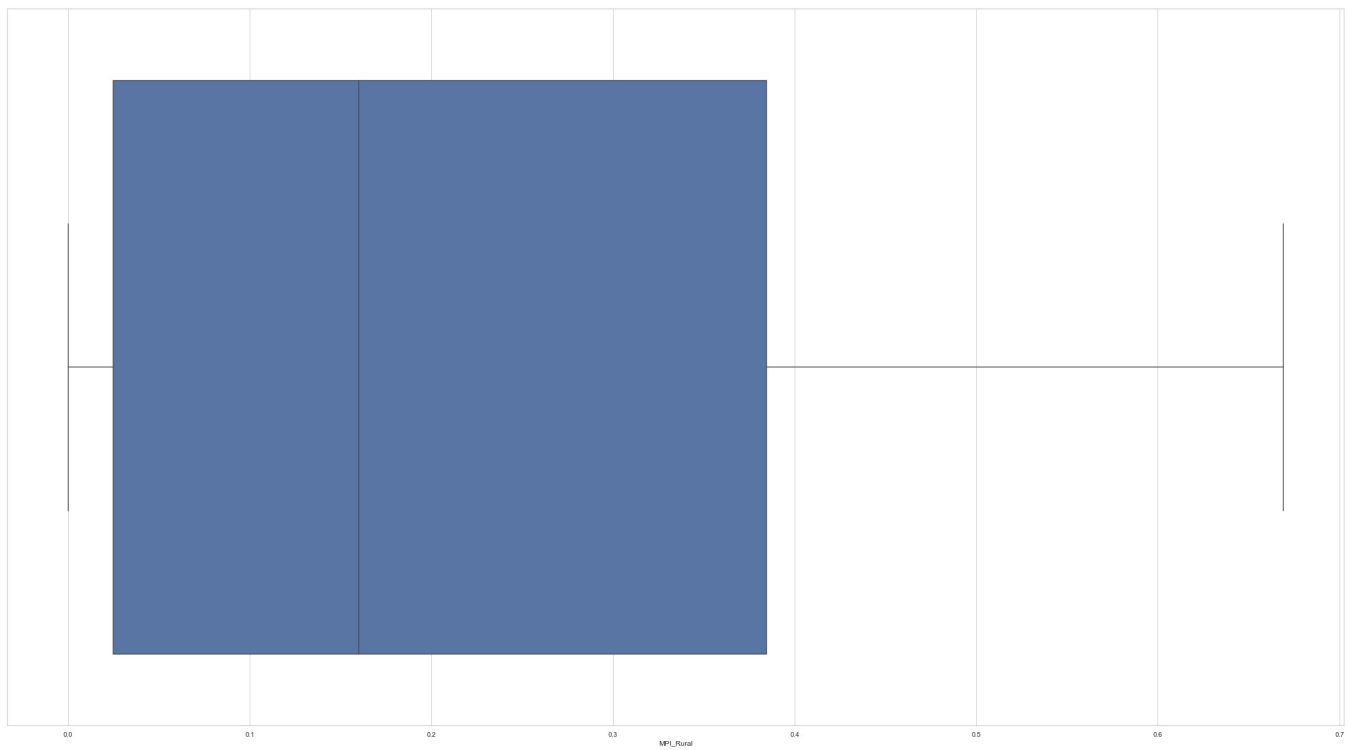
```
In [28]: datos.set_index("Country").plot.barh(stacked=True);
```



```
In [ ]:
```

```
In [29]: datos.set_index("Country")["MPI_Rural"].plot(kind="bar");
```

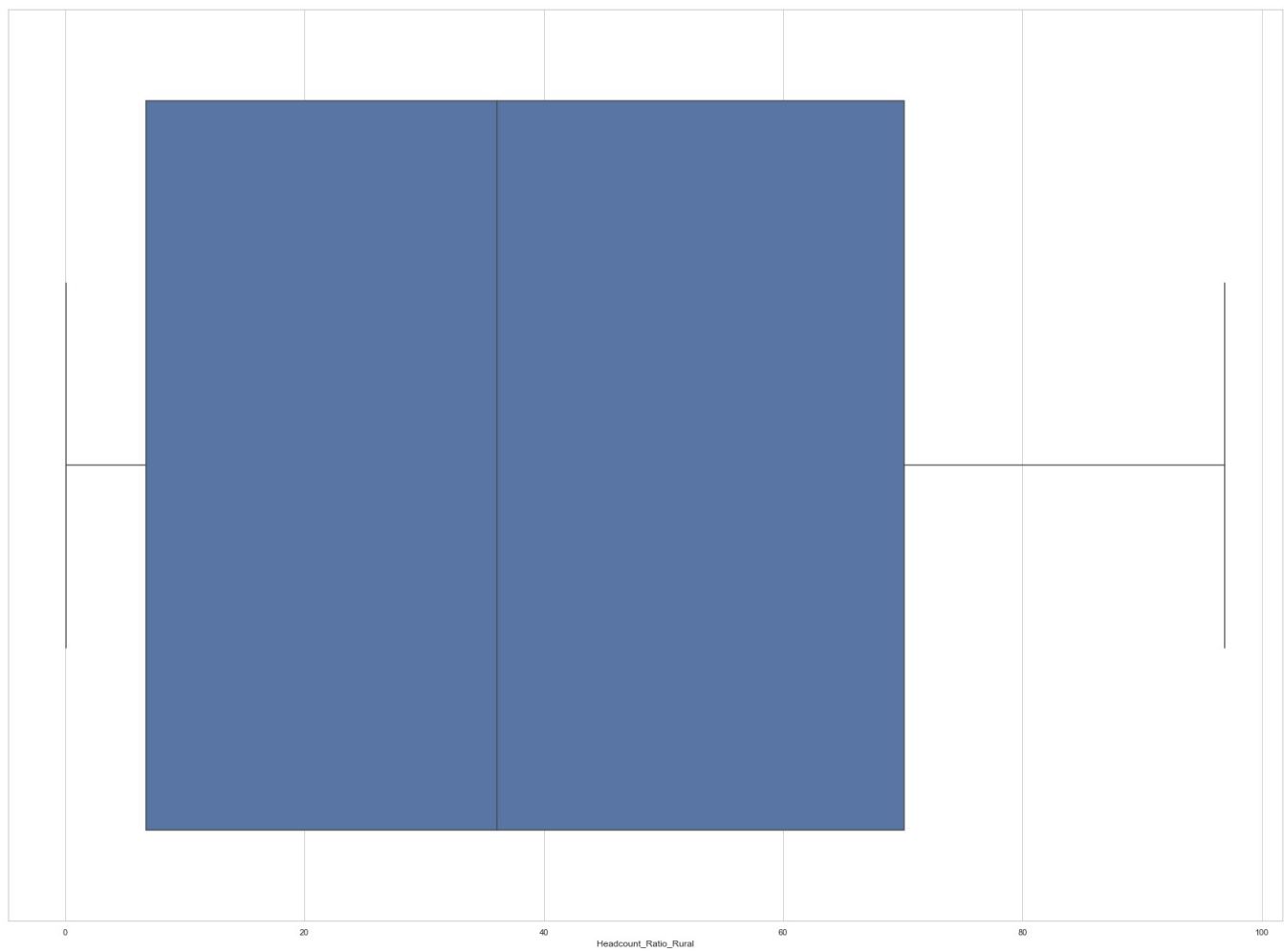
```
In [31]: sb.set_theme(style="whitegrid")
        ax = sb.boxplot(x=datos["MPI_Rural"])
```



```
In [ ]:
```

```
In [49]: datos.set_index("Country")["Headcount_Ratio_Rural"].plot(kind="bar");
```

```
In [60]: sb.set_theme(style="whitegrid")
         ax = sb.boxplot(x=datos["Headcount_Ratio_Rural"])
```



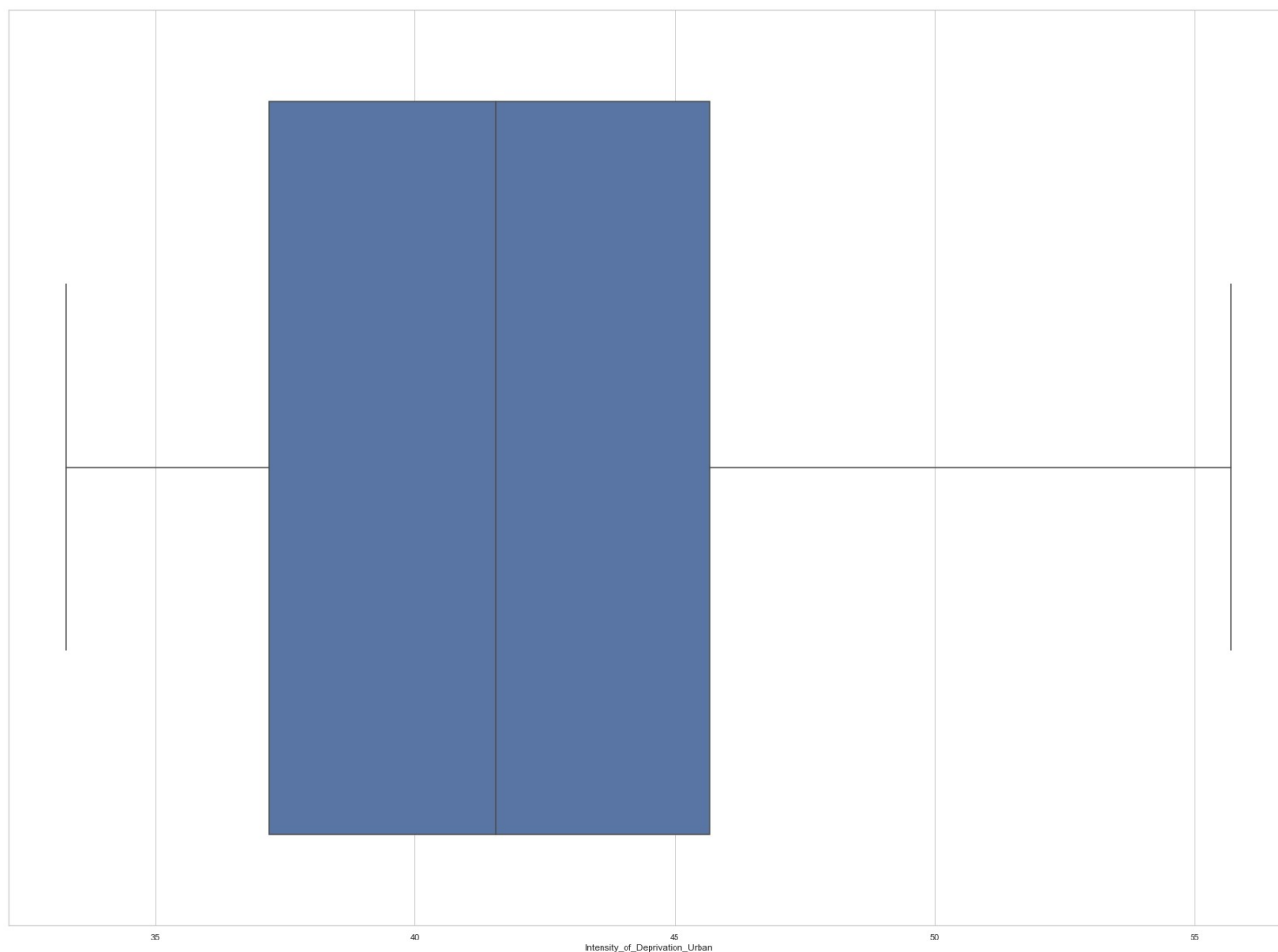
```
In [ ]:
```

```
In [4... datos.set_index("Country")["Intensity_of_Deprivation_Rural"].plot(kind="bar
```

^

v

```
In [61]: sb.set_theme(style="whitegrid")
         ax = sb.boxplot(x=datos["Intensity_of_Deprivation_Urban"])
```



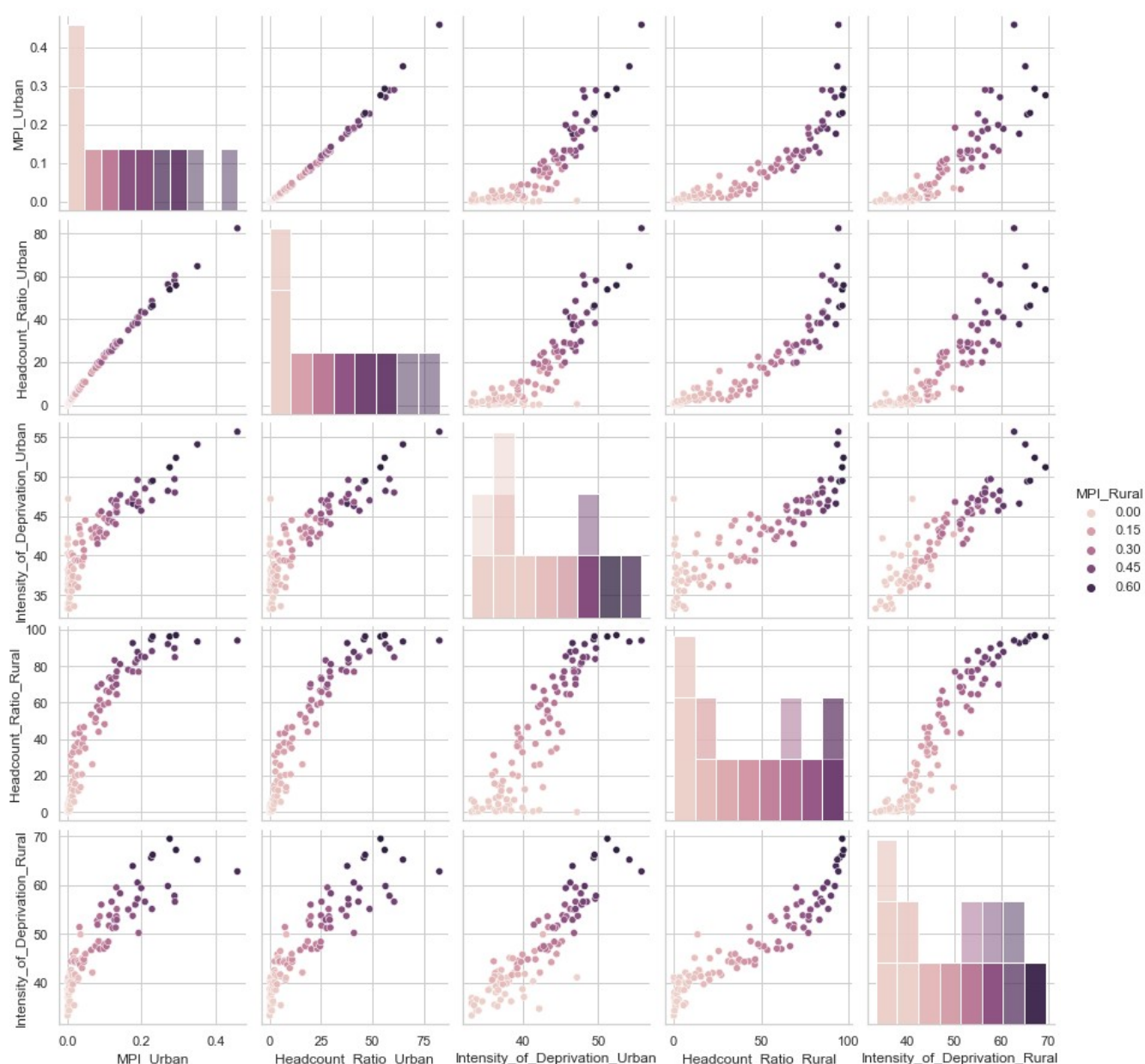
```
In [55]: datos
```

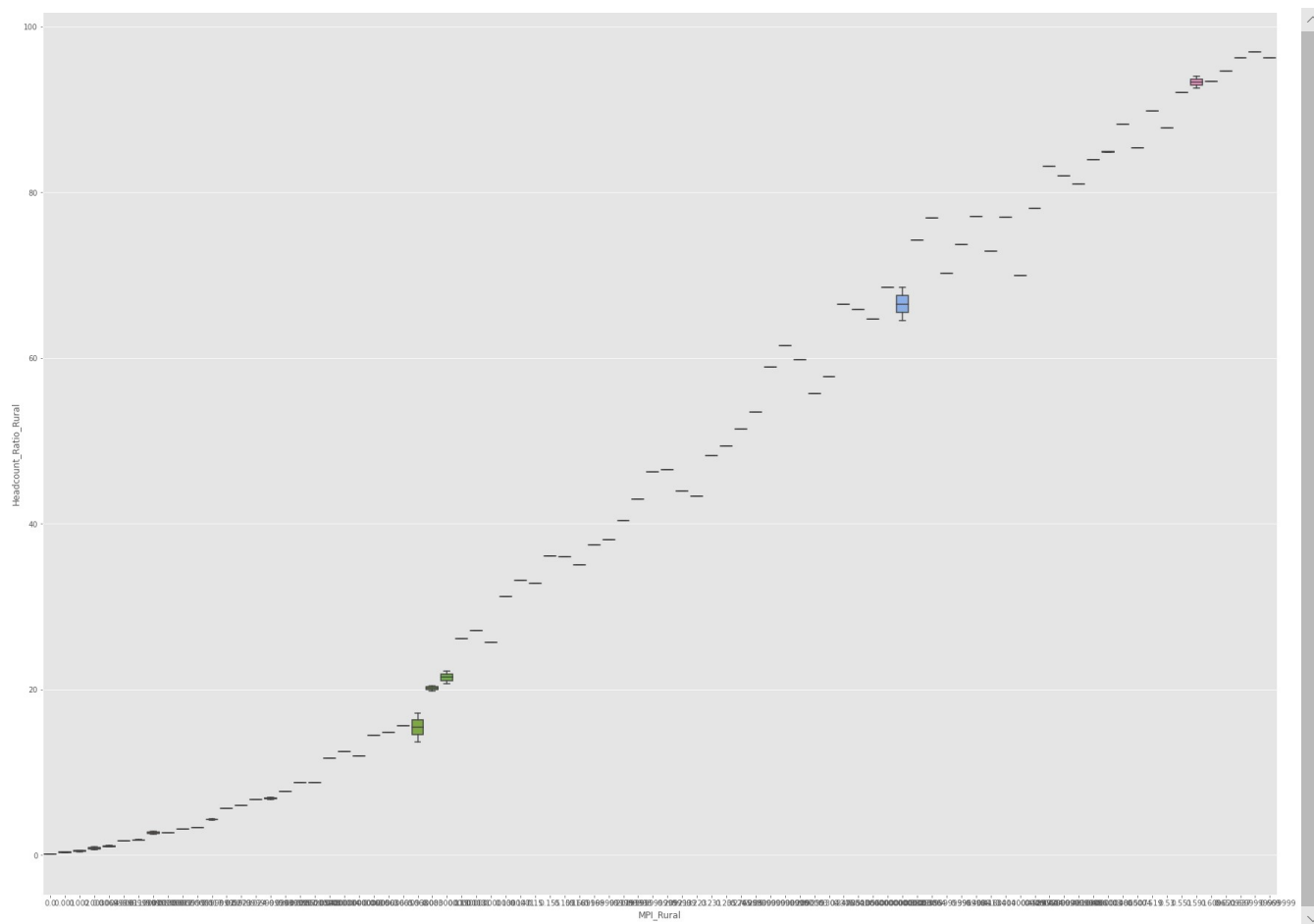
Ou...	ISO	Country	MPI_Urban	Headcount_Ratio_Urban	Intensity_of_Deprivation_Urban	MPI_Rural
0	KAZ	Kazakhstan	0.000	0.0	33.3	0.000
1	SRB	Serbia	0.000	0.1	41.4	0.002
2	KGZ	Kyrgyzstan	0.000	0.1	40.2	0.003
3	TUN	Tunisia	0.000	0.1	35.6	0.012
4	ARM	Armenia	0.001	0.2	33.3	0.001
...
97	CAF	Central African Republic	0.289	58.2	49.7	0.519
98	LBR	Liberia	0.290	60.5	48.0	0.481
99	SOM	Somalia	0.293	55.9	52.4	0.651


```
In [52]: dataframe = pd.read_csv(r"MPI_national.csv")
dataframe.head()
```

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	33.3
1	SRB	Serbia	0.000	0.1	41.4	0.002	41.4
2	KGZ	Kyrgyzstan	0.000	0.1	40.2	0.003	40.2
3	TUN	Tunisia	0.000	0.1	35.6	0.012	35.6
4	ARM	Armenia	0.001	0.2	33.3	0.001	33.3

```
In [62]: sb.pairplot(dataframe, hue="MPI_Rural", diag_kind="hist");
```





```
In [11]: datos.corr(method= "kendall")
```

Ou...	MPI_Urban	Headcount_Ratio_Urban	Intensity_of_Deprivation_Urban	MPI_R
MPI_Urban	1.000000	0.981233	0.705851	0.856
Headcount_Ratio_Urban	0.981233	1.000000	0.684539	0.847
Intensity_of_Deprivation_Urban	0.705851	0.684539	1.000000	0.696
MPI_Rural	0.856086	0.847173	0.696237	1.000
Headcount_Ratio_Rural	0.854226	0.847278	0.686322	0.971
Intensity_of_Deprivation_Rural	0.775371	0.760670	0.744445	0.843

```
l... #colormap = plt.cm.viridis
      #.figure(figsize=(12,12))
      #plt.title("MPI_National", y=1.05, x=15)
      #sb.heatmap(datos.astype(float).corr(),linewidths=0.1,vmax=1.0, square=True,

In ... datos1=datos.drop(columns=['ISO','Country'])
      colormap = plt.cm.viridis
      plt.figure(figsize=(50,30))
      plt.title("MPI_National", y=1.05, x=15)
      sb.heatmap(datos1.astype(float).corr(),linewidths=0.1,vmax=1.0, square=True,
```

jupyter Mapas de Calor y boxplots. Last Checkpoint: hace 10 minutos (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Félix Javier Rojas Gallardo A01201946

Amanda María Real Núñez. A01367729

Preguntas:

- ¿Hay alguna variable que no aporta información?

En el análisis y creación de representaciones visuales de índice multidimensional de pobreza rural, que consiste en las columnas de MPI_Rural, Headcount_Ratio_Rural y Intensity_of_Deprivation_Rural, la variable que no nos aporta información necesaria fue la columna de ISO

- Si tuvieras que eliminar variables, ¿cuáles quitarías y por qué?

La variable principal es el índice de pobreza multidimensional (MPI), este es un índice compuesto, por lo cual las otras variables en nuestra base de datos son importantes para poder complementar y entender este índice. Por lo cual no sería conveniente eliminar ninguna variable. Afortunadamente las columnas que tenemos son las necesarias para poder ver el comportamiento del índice multidimensional de pobreza tanto urbana como rural, en las naciones que se encuentran en la base de datos, en lugar de eliminar alguna variable considero que podríamos sacar un mayor provecho de las columnas de ISO y Country si fueran variables de tipo float. Sin embargo el tener variables string son necesarias en la columna de los países, debido a que nos fueron útiles en las gráficas de barras.

- ¿Existen variables que tengan datos extraños?

Ninguna variable contenía datos anormales, debido a que los datos numéricos podían analizarse de manera óptima, al inicio del análisis modificamos nuestra base de datos en Excel para que no tuviera caracteres que nos fueran difíciles de utilizar en Jupyter, es por ello que la base no cuenta con acentos o caracteres especiales, todos los datos son numéricos, excepto por las primeras dos columnas que contienen texto (aun así son útiles).

- Si comparas las variables, ¿todas están en rangos similares? ¿Crees que esto afecte?

Los rangos de datos se encontraban dentro de un rango considerable, es decir la base contiene el mismo número de datos en todas las columnas, los datos numéricos de cada columna con datos tipo float, contienen las mismas unidades y por medio de la gráfica de caja y bigotes, logramos observar que no contamos con datos atípicos la mayoría se encuentra dentro del rango determinado. Podemos decir que las variables tienen rangos numéricos distintos, pero para el objetivo y análisis que buscamos esto no afecta, esto afectaría en caso de que quisiéramos relacionar estas variables entre sí.

- ¿Puedes encontrar grupos que se parezcan? ¿Qué grupos son estos?

En nuestra base de datos se estructura en el orden del MPI de los países, por lo cual no hay grupos de datos similares como tal. Al analizar los datos del MPI Urbano y Rural logramos apreciar que se tienen datos no parecidos entre sí, pero fueron tomados con las mismas medidas multidimensionales de pobreza, debido a que nos permite hacer una comparación del índice de pobreza en lugares urbanos y rurales, todo esto por medio de las representaciones visuales, las cuales nos permiten ver el comportamiento de cada nación en estos dos ámbitos.

```
In [27]: 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
```

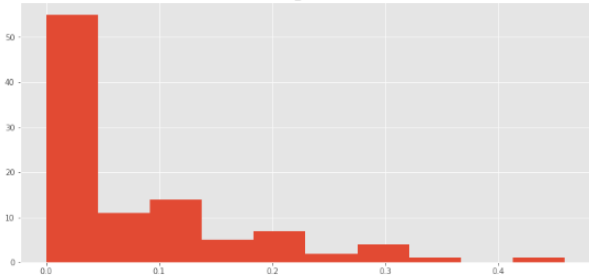
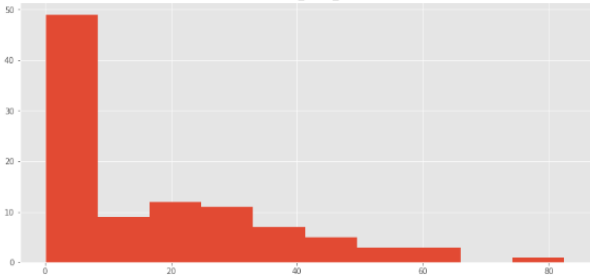
jupyter Mapas de Calor y boxplots. Last Checkpoint: hace 6 minutos (unsaved changes)

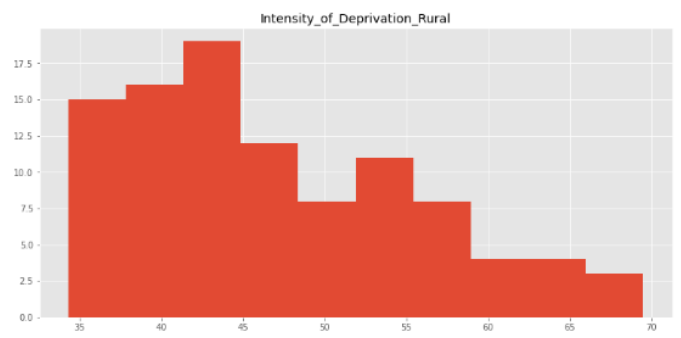
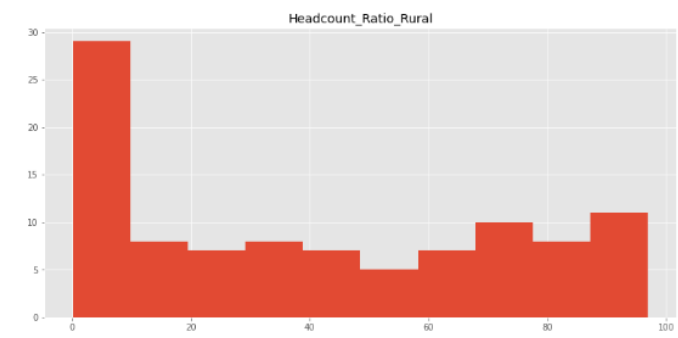
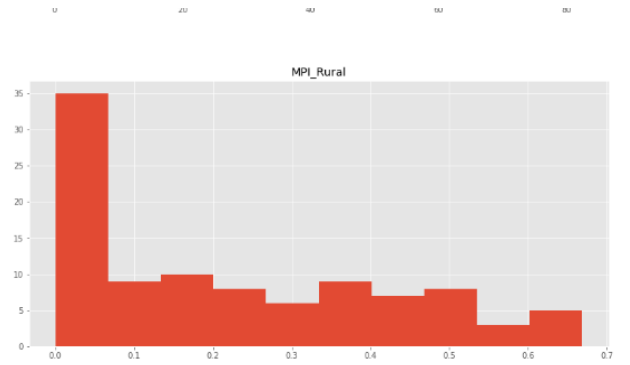
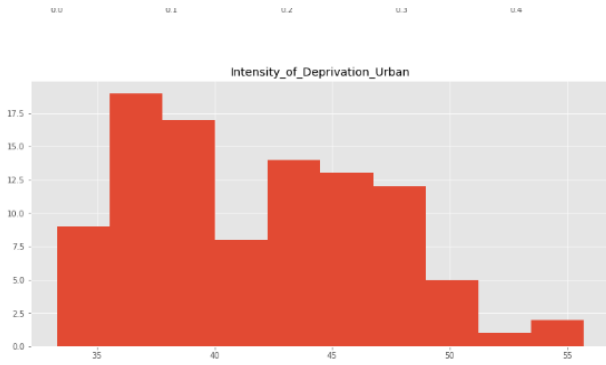
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

```
In [9]: datos['Intensity_of_Deprivation_Rural'].median()

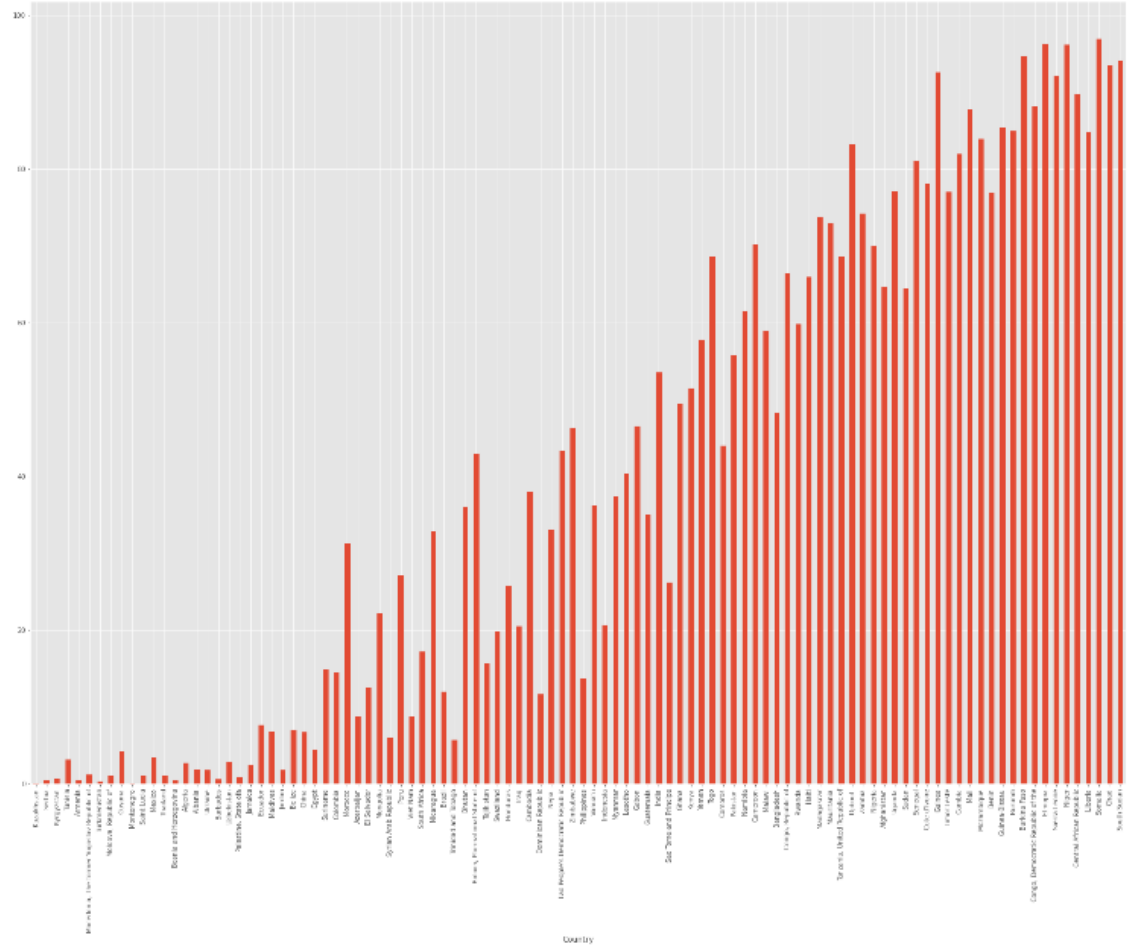
Out[9]: 44.8
```

```
In [21]: datos.drop([0,1]).hist()
plt.show()
```



```
In [49]: datos.set_index("Country")["Headcount_Ratio_Rural"].plot(kind="bar");
```



File Edit View Insert Cell Kernel Widgets Help

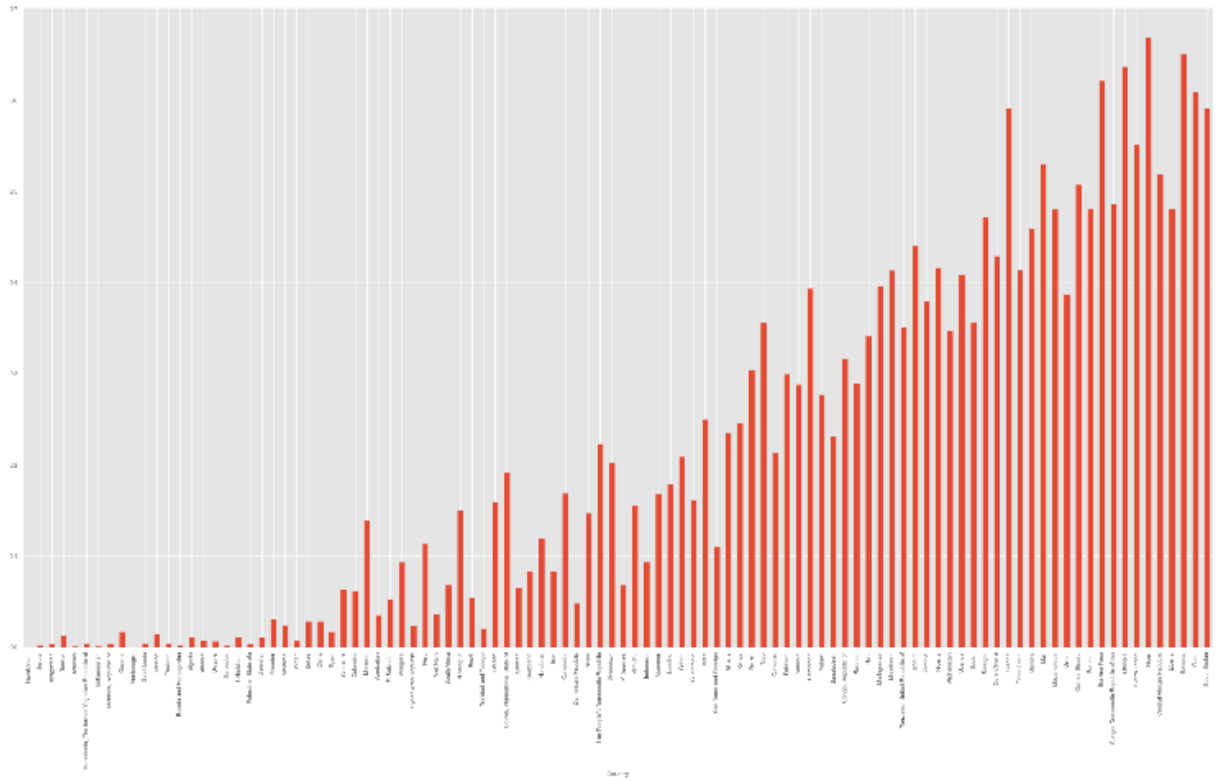
Not Trusted

Python 3

Run

In []:

In [29]: `datos.set_index("Country")["MPI_Rural"].plot(kind="bar");`



File Edit View Insert Cell Kernel Widgets Help

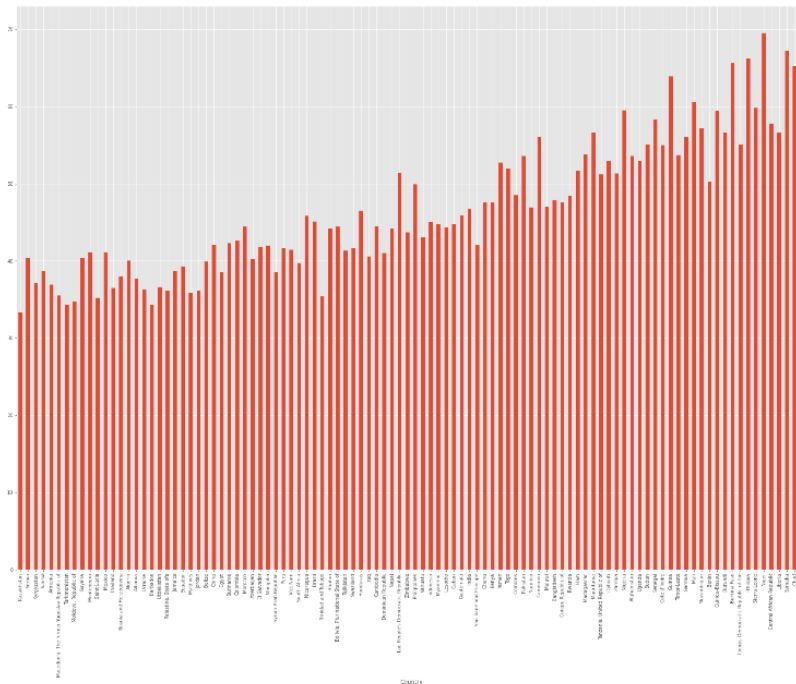
Not Trusted

Python 3

Run

In []:

In [40]: `datos.set_index("Country")["Intensity_of_Deprivation_Rural"].plot(kind="bar");`



In [32]:

```
datos1=datos.drop(columns=['ISO','Country'])
colormap = plt.cm.viridis
plt.figure(figsize=(50,30))
plt.title("MPI_National", y=1.05, x=15)
sb.heatmap(datos1.astype(float).corr(),linewidths=0.1,vmax=1.0, square=True, cmap=colormap, linecolor="white", annot=True)
```



Obtencion de datos estadisticos

Carga los datos usando tu lector de csv o con pandas. Es recomendable hacerlo con pandas. Verifica la cantidad de datos que tienes, las variables que contiene cada vector de datos e identifica el tipo de variables. Analiza las variables para saber que representa cada una y en que rangos se encuentran. Si la descripción del problema no te lo indica, utiliza el máximo y el mínimo para encontrarlo. Basándose en la media, mediana y desviación estándar de cada variable, que conclusiones puedes entregar de los datos.

```
In [1]: import pandas as pd
        datos=pd.read_csv('MPI_national.csv')
```

```
In [11]: datos
```

```
Out[11]:
```

	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 [3]: tipo_datos = pd.read_csv ("MPI_national.csv")
```

```
In [4]: tipo_datos.dtypes
```

```
Out[4]: 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 [5]: datos['Headcount Ratio Rural'].median()
```

```
Out[5]: 36.055
```

```
In [7]: import pandas as pd
import seaborn as sb
import numpy as np; np.random.seed(0)
import matplotlib.pyplot as plt
datos= pd.read_csv('MPI_national.csv')

from matplotlib import cm
plt.rcParams['figure.figsize']=(16,9)
plt.style.use('ggplot')

datos.shape
```

```
Out[7]: (102, 8)
```

```
In [38]: datos.describe()
```

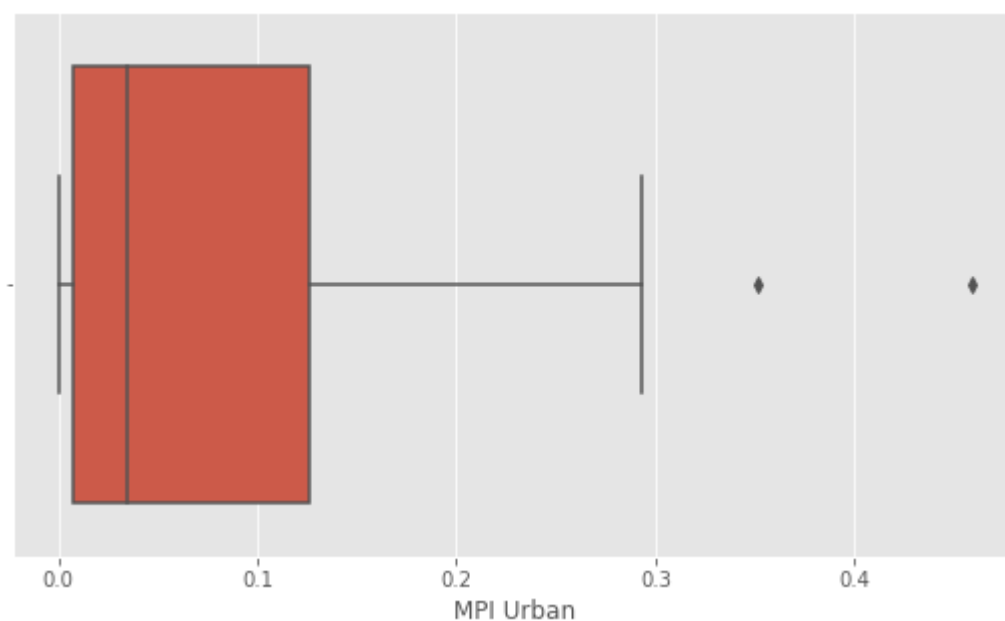
```
Out[38]:
```

	MPI Urban	Headcount Ratio Urban	Intensity of Deprivation Urban	MPI Rural	Headcount Ratio Rural	Intensity of Deprivation Rural
count	102.000000	102.000000	102.000000	102.000000	102.000000	102.000000
mean	0.078343	16.809804	41.678431	0.214676	40.036176	46.824510
std	0.093693	18.498448	5.135908	0.201208	33.270714	8.783191
min	0.000000	0.000000	33.300000	0.000000	0.090000	33.300000
25%	0.007250	1.950000	37.200000	0.025000	6.745000	40.225000
50%	0.034500	8.400000	41.550000	0.160000	36.055000	44.800000
75%	0.125750	27.575000	45.675000	0.384500	70.130000	53.425000
max	0.459000	82.500000	55.700000	0.669000	96.920000	69.500000

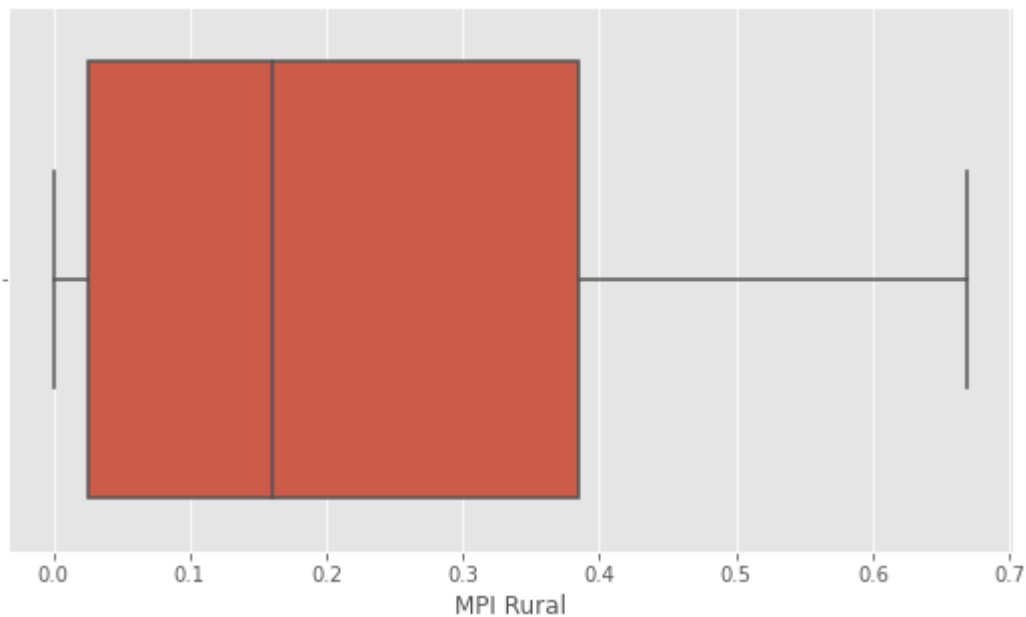
```
In [8]: datos.drop (1,0).hist()
plt.show()
```



```
In [40]: plt.figure(figsize=(9,5))
ax=sb.boxplot(x=datos["MPI Urban"])
```



```
In [9]: plt.figure(figsize=(9,5))
ax=sb.boxplot(x=datos["MPI Rural"])
```



```
In [ ]: ax= sb.boxplot(x="sepal_length" , y="sepal_width",datos )
```

```
In [10]: datos.corr(method='pearson')
```

```
Out[10]:
```

	MPI Urban	Headcount Ratio Urban	Intensity of Deprivation Urban	MPI Rural	Headcount Ratio Rural	Intensity of Deprivation Rural
MPI Urban	1.000000	0.995981	0.880024	0.922065	0.887147	0.884069
Headcount Ratio Urban	0.995981	1.000000	0.884032	0.939615	0.913555	0.896901
Intensity of Deprivation Urban	0.880024	0.884032	1.000000	0.892678	0.878833	0.904428
MPI Rural	0.922065	0.939615	0.892678	1.000000	0.986750	0.966458
Headcount Ratio Rural	0.887147	0.913555	0.878833	0.986750	1.000000	0.940608
Intensity of Deprivation Rural	0.884069	0.896901	0.904428	0.966458	0.940608	1.000000

```
In [44]: colormap=plt.cm.viridis
plt.figure(figsize=(12,12));
plt.title('Correlacion de Pearson MPI', y=1.05, size=15);
sb.heatmap(datos.astype(float).corr(),linewidth=0.1, vmax=1.0, square=True, cmap=colorm
```

