

## TAREA 4: Análisis de Estimaciones de Pobreza Comunal en Chile - 2022

Autores: Simón Ramirez - Victor Saldivia - Carlos Saquel  
Asignatura: Visualización de la Información

Este análisis tiene como objetivo explorar las estimaciones de pobreza por ingresos y pobreza multidimensional en las comunas de Chile durante el año 2022. Los datos fueron obtenidos a partir de los conjuntos de datos proporcionados por el Ministerio de Desarrollo Social y Familia de Chile a través del Ministerio de Desarrollo Social y Familia. Se presenta un análisis descriptivo y visualizaciones para comprender de mejor forma las correlaciones y características de la pobreza en las comunas.

### IMPORTACIÓN DE LIBRERÍAS

```
In [1]: import pandas as pd
import re
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import matplotlib.pyplot as plt
from matplotlib.collections import PatchCollection
from matplotlib.colors import LinearSegmentedColormap

from mpl_toolkits.axesartist.grid_finder import DictFormatter
import mpl_toolkits.axesartist.floating_axes as floating_axes
```

### CARGA DEL DATASET

El archivo que vamos a utilizar contiene información sobre las estimaciones de pobreza en Chile para el año 2022, tanto de ingresos como multidimensional.

```
In [2]: file_path_1 = './data/Estimaciones_Indice_Pobreza_Multidimensional_Comunas_2022.xlsx'
file_path_2 = './data/Estimaciones_Indice_Pobreza_Ingresos_Comunas_2022.xlsx'

In [3]: # Leer ambos archivos
df_pobreza_ingresos = pd.read_excel(file_path_1)
df_pobreza_multidimensional = pd.read_excel(file_path_2)
```

### FUNCIONES DE LIMPIEZA DE LOS DATOS

```
In [4]: # Funciones de limpieza de columnas y acentos
def accent_replace(col):
    reemplazos = {
        'á': 'a', 'é': 'e', 'í': 'i', 'ó': 'o', 'ú': 'u',
        'â': 'a', 'ê': 'e', 'î': 'i', 'ô': 'o', 'û': 'u',
    }
    for acento, sin_acento in reemplazos.items():
        col = col.replace(acento, sin_acento)
    return col

def columns_clean(columns):
    columns_limpias = []
    for col in columns:
        col = accent_replace(col).lower().replace('\n', '')
        col = re.sub(r'[^a-z0-9_]', '', col).replace(' ', '_')
        col = re.sub(r'^(?=[0-9_])', '', col).rstrip('_')
    columns_limpias.append(col)
    return columns_limpias

def get_clean_data(file, get_rows):
    df = pd.read_excel(file, skiprows=get_rows[0])
    df = df.iloc[get_rows[1]:]
    df.columns = columns_clean(df.columns)
    return df
```

### LIMPIEZA, CARGA Y AJUSTES DE COLUMNAS DEL DATASET DE POBREZA POR INGRESOS

```
In [5]: df_1 = get_clean_data(file='./data/Estimaciones_Indice_Pobreza_Ingresos_Comunas_2022.xlsx', get_rows=[2, 345])
df_1.columns = ['codigo', 'region', 'comuna', 'poblacion', 'personas_pobreza_por_ingresos', 'porcentaje_pobreza_por_ingresos_2022', 'limite_inferior_ingreso', 'limite_superior_ingreso', 'presencia_comuna_casen', 'tipo_de_estimacion_sae']
df_1.head()
```

	codigo	region	comuna	poblacion	personas_pobreza_por_ingresos	porcentaje_pobreza_por_ingresos_2022	limite_inferior_ingreso	limite_superior_ingreso	presencia_comuna_casen	tipo_de_estimacion_sae
0	1101	Tarapacá	Iquique	229674.0	18122.0	0.078904	0.064903	0.092905	SI	Directa y Sintético (Fay-Herriot)
1	1107	Tarapacá	Alto Hospicio	138527.0	21144.0	0.152634	0.137397	0.167872	SI	Directa y Sintético (Fay-Herriot)
2	1401	Tarapacá	Pozo Almonte	18290.0	2805.0	0.153344	0.115596	0.191092	SI	Directa y Sintético (Fay-Herriot)
3	1402	Tarapacá	Camíña	1380.0	332.0	0.240365	0.161336	0.319394	SI	Sintético
4	1403	Tarapacá	Colchane	1575.0	350.0	0.222535	0.147399	0.297672	SI	Sintético

### ANÁLISIS DESCRIPTIVO DE LOS DATOS

```
In [6]: df_1.describe(include='all')
```

	codigo	region	comuna	poblacion	personas_pobreza_por_ingresos	porcentaje_pobreza_por_ingresos_2022	limite_inferior_ingreso	limite_superior_ingreso	presencia_comuna_casen	tipo_de_
count	345.0	345	345	345.000000	345.000000	345.000000	345.000000	345.000000	345	
unique	345.0	16	345	NaN	NaN	NaN	NaN	NaN	2	Dir
top	1101.0	Metropolitana	Iquique	NaN	NaN	NaN	NaN	NaN	SI	
freq	1.0	52	1	NaN	NaN	NaN	NaN	NaN	335	
mean	NaN	NaN	NaN	57618.115942	3748.527536	0.090339	0.051886	0.129155	NaN	
std	NaN	NaN	NaN	91321.953828	5168.776002	0.043544	0.034849	0.054402	NaN	
min	NaN	NaN	NaN	207.000000	6.000000	0.008747	0.000000	0.021982	NaN	
25%	NaN	NaN	NaN	9959.000000	985.000000	0.059865	0.026897	0.092199	NaN	
50%	NaN	NaN	NaN	20525.000000	1863.000000	0.085873	0.047585	0.123562	NaN	
75%	NaN	NaN	NaN	56542.000000	3888.000000	0.115698	0.070589	0.163677	NaN	
max	NaN	NaN	NaN	661900.000000	33344.000000	0.242779	0.191087	0.319394	NaN	

### Información General del Dataset

```
In [7]: df_1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 345 entries, 0 to 344
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype  ---
 0   codigo                345 non-null    object
 1   region                345 non-null    object
 2   comuna                345 non-null    object
 3   poblacion             345 non-null    float64
 4   personas_pobreza_por_ingresos  345 non-null    float64
 5   porcentaje_pobreza_por_ingresos_2022  345 non-null    float64
 6   limite_inferior_ingreso  345 non-null    float64
 7   limite_superior_ingreso  345 non-null    float64
 8   presencia_comuna_casen  345 non-null    object
 9   tipo_de_estimacion_sae  345 non-null    object
dtypes: float64(5), object(5)
memory usage: 27.1+ KB
```

### LIMPIEZA, CARGA Y AJUSTES DE COLUMNAS DEL DATASET DE POBREZA MULTIDIMENSIONAL

```
In [8]: df_2 = get_clean_data(file='./data/Estimaciones_Indice_Pobreza_Multidimensional_Comunas_2022.xlsx', get_rows=[2, 345])
df_2.columns = ['codigo', 'region', 'comuna', 'poblacion', 'personas_pobreza_multidimensional', 'porcentaje_pobreza_multidimensional_2022', 'limite_inferior_multidimensional', 'limite_superior_multidimensional', 'presencia_comuna_casen', 'tipo_de_estimacion_sae']
df_2.head(10)
```

	codigo	region	comuna	poblacion	personas_pobreza_multidimensional	porcentaje_pobreza_multidimensional_2022	limite_inferior_multidimensional	limite_superior_multidimensional	presencia_comuna_casen
0	1101	Tarapacá	Iquique	229674.0	41966.566241	0.182722	0.162429	0.203015	
1	1107	Tarapacá	Alto Hospicio	138527.0	45162.442981	0.328019	0.294776	0.357263	
2	1401	Tarapacá	Pozo Almonte	18290.0	4563.395313	0.249502	0.199334	0.299671	
3	1402	Tarapacá	Camíña	1380.0	307.626999	0.222918	0.137766	0.308071	
4	1403	Tarapacá	Colchane	1575.0	472.752772	0.300160	0.187495	0.412826	
5	1404	Tarapacá	Huara	3072.0	1184.719019	0.385651	0.318693	0.452609	
6	1405	Tarapacá	Pica	6164.0	1042.267243	0.168219	0.119494	0.216945	
7	2101	Antofagasta	Antofagasta	438942.0	73102.886971	0.166543	0.140852	0.192234	
8	2102	Antofagasta	Mejillones	15502.0	3077.761164	0.198540	0.126538	0.270541	
9	2103	Antofagasta	Sierra Gorda	1790.0	294.946135	0.164774	0.074272	0.255277	

```
In [9]: df_2.describe(include='all')
```

	codigo	region	comuna	poblacion	personas_pobreza_multidimensional	porcentaje_pobreza_multidimensional_2022	limite_inferior_multidimensional	limite_superior_multidimensional	presencia_
count	345.0	345	345	345.000000	345.000000	345.000000	345.000000	345.000000	
unique	345.0	16	345	NaN	NaN	NaN	NaN	NaN	
top	1101.0	Metropolitana	Iquique	NaN	NaN	NaN	NaN	NaN	
freq	1.0	52	1	NaN	NaN	NaN	NaN	NaN	
mean	NaN	NaN	NaN	57618.115942	9763.030150	0.194504	0.138041	0.251170	
std	NaN	NaN	NaN	91321.953828	15099.360264	0.070852	0.060731	0.063519	
min	NaN	NaN	NaN	207.000000	0.234564	0.001133	0.000000	0.014602	
25%	NaN	NaN	NaN	9959.000000	2051.968205	0.052218	0.050272	0.200601	
50%	NaN	NaN	NaN	20525.000000	4087.274968	0.188434	0.130985	0.244170	
75%	NaN	NaN	NaN	56542.000000	9050.028970	0.228267	0.165782	0.294340	
max	NaN	NaN	NaN	661900.000000	125234.776994	0.586389	0.434880	0.738099	

### Información General del Dataset

```
In [10]: df_2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 345 entries, 0 to 344
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype  ---
 0   codigo                345 non-null    object
 1   region                345 non-null    object
 2   comuna                345 non-null    object
 3   poblacion             345 non-null    float64
 4   personas_pobreza_multidimensional  345 non-null    float64
 5   porcentaje_pobreza_multidimensional_2022  345 non-null    float64
 6   limite_inferior_multidimensional  345 non-null    float64
 7   limite_superior_multidimensional  345 non-null    float64
 8   presencia_comuna_casen  345 non-null    object
 9   tipo_de_estimacion_sae  345 non-null    object
dtypes: float64(5), object(5)
memory usage: 27.1+ KB
```

### COMBINACIÓN DE DATASETS

```
In [11]: df_merge = df_1.merge(df_2[['codigo', 'porcentaje_pobreza_multidimensional_2022', 'limite_inferior_multidimensional', 'limite_superior_multidimensional']], on='codigo')
df_merge
```

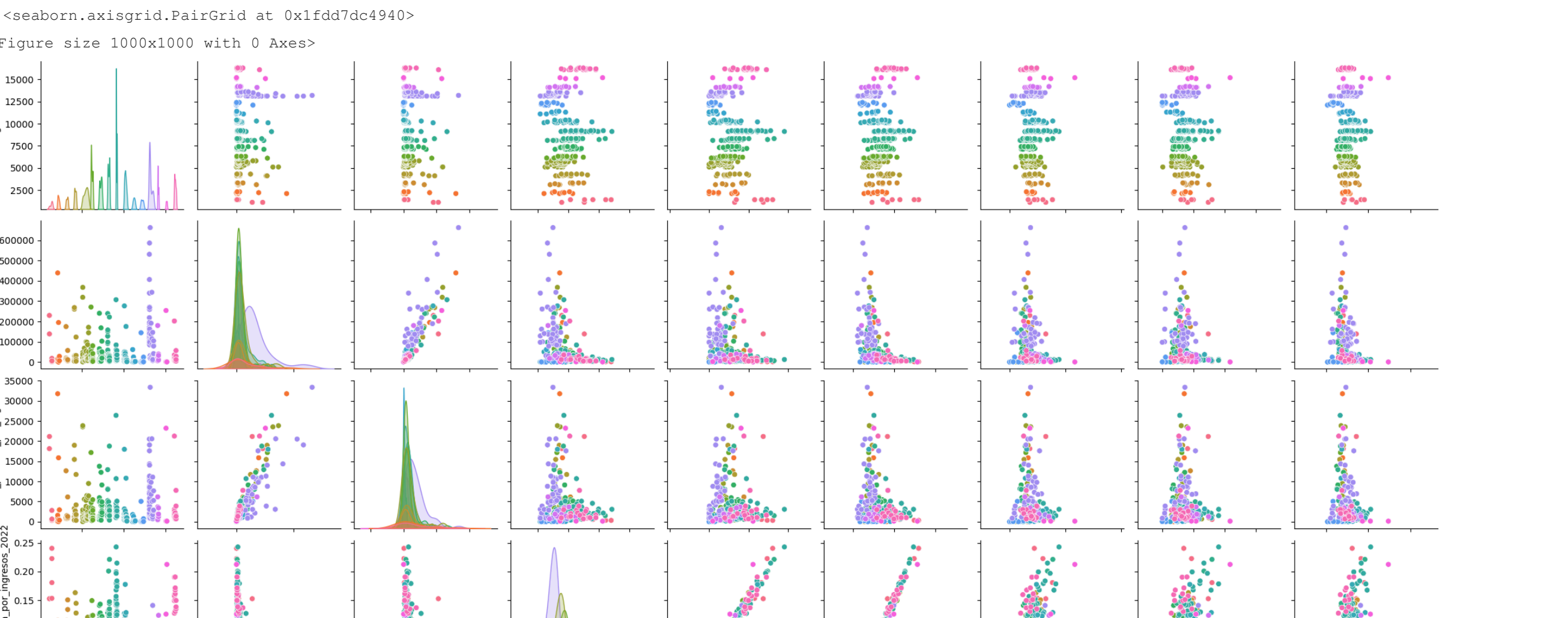
	codigo	region	comuna	poblacion	personas_pobreza_por_ingresos	porcentaje_pobreza_por_ingresos_2022	limite_inferior_ingreso	limite_superior_ingreso	presencia_comuna_casen	tipo_de_estimacion_sae
0	1101	Tarapacá	Iquique	229674.0	18122.0	0.078904	0.064903	0.092905	SI	Directa y Sintético (Fay-Herriot)
1	1107	Tarapacá	Alto Hospicio	138527.0	21144.0	0.152634	0.137397	0.167872	SI	Directa y Sintético (Fay-Herriot)
2	1401	Tarapacá	Pozo Almonte	18290.0	2805.0	0.153344	0.115596	0.191092	SI	Directa y Sintético (Fay-Herriot)
3	1402	Tarapacá	Camíña	1380.0	332.0	0.240365	0.161336	0.319394	SI	Sintético
4	1403	Tarapacá	Colchane	1575.0	350.0	0.222535	0.147399	0.297672	SI	Sintético
...	...	...	...	...	...	...	...	...	...	...
340	16301	Nuble	San Carlos	56782.0	7775.0	0.136924	0.103947	0.169902	SI	Directa y Sintético (Fay-Herriot)
341	16302	Nuble	Cholchagua	28817.0	3887.0	0.134886	0.092175	0.177596	SI	Directa y Sintético (Fay-Herriot)
342	16303	Nuble	Niquén	11543.0	1598.0	0.138420	0.090545	0.186295	SI	Directa y Sintético (Fay-Herriot)
343	16304	Nuble	San Fabián	4738.0	709.0	0.149570	0.100026	0.198115	SI	Directa y Sintético (Fay-Herriot)
344	16305	Nuble	San Nicolás	12365.0	1333.0	0.107768	0.061146	0.154390	SI	Directa y Sintético (Fay-Herriot)

345 rows × 13 columns

### VISUALIZACIÓN DE CORRELACIONES

```
In [12]: sns.heatmap(df_merge.corr(numeric_only=True), annot=True)
```

```
Out[12]: <Axes: >
```



### Visualización de relaciones múltiples: Pairplot por región

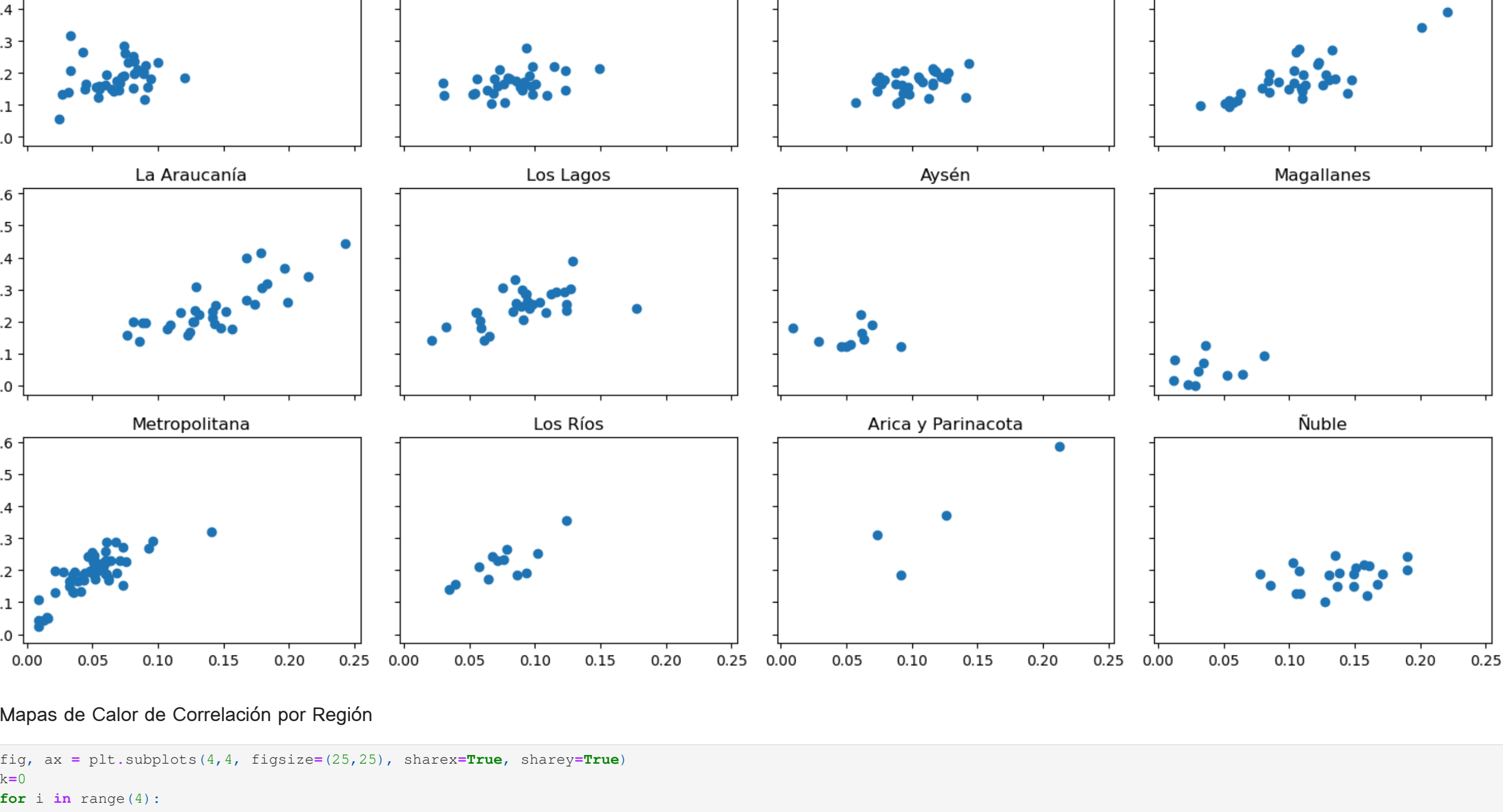
```
In [13]: plt.figure(figsize=(10,10))
sns.pairplot(df_merge, hue='region')
```

```
Out[13]: <seaborn.axisgrid.PairGrid at 0x1fd7dc4910>
```



### Scatter Plot de Pobreza por Ingresos vs Pobreza Multidimensional, Segmentado por Región

```
In [14]: fig, ax = plt.subplots(4, 4, figsize=(15,10), sharex=True, sharey=True)
k=0
for i in range(4):
    for j in range(4):
        df_merge_region = df_merge[df_merge['region']==df_merge.region.unique()[k]]
        ax[i,j].scatter(df_merge_region['porcentaje_pobreza_por_ingresos_2022'], y=df_merge_region['porcentaje_pobreza_multidimensional_2022'])
        ax[i,j].set_title(df_merge_region.unique()[k])
        k+=1
plt.tight_layout()
plt.show()
```



### Mapas de Calor de Correlación por Región

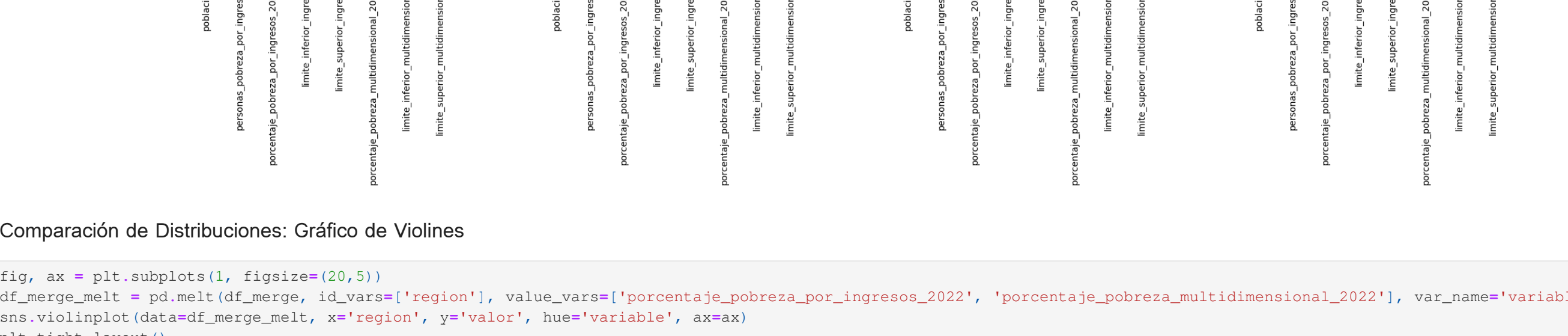
```
In [15]: fig, ax = plt.subplots(4, 4, figsize=(15,25), sharex=True, sharey=True)
```

```
for i in range(4):
    for j in range(4):
        df_merge_region = df_merge[df_merge['region']==df_merge.region.unique()[k]]
        sns.heatmap(df_merge_region.corr(numeric_only=True), annot=True, ax=ax[i,j])
        ax[i,j].set_title(df_merge_region.unique()[k])
        k+=1
plt.tight_layout()
plt.show()
```



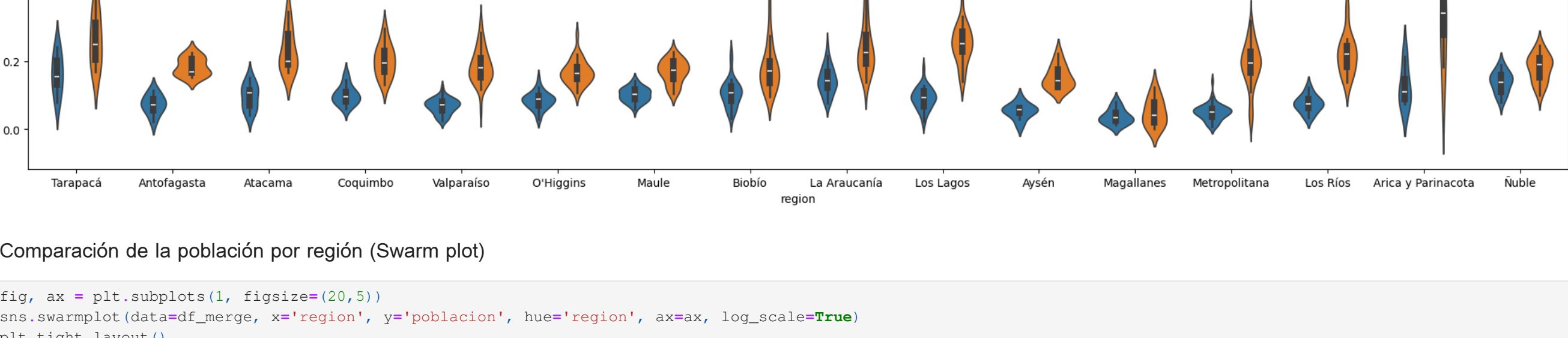
### Comparación de Distribuciones: Gráfico de Violines

```
In [16]: fig, ax = plt.subplots(1, 1, figsize=(20,5))
df_merge_melt = pd.melt(df_merge, id_vars='region', value_vars=['porcentaje_pobreza_por_ingresos_2022', 'porcentaje_pobreza_multidimensional_2022'], var_name='variable')
sns.violinplot(data=df_merge_melt, x='region', y='value', bw='variable', ax=ax)
plt.tight_layout()
plt.show()
```



### Comparación de la población por región (Swarm plot)

```
In [17]: fig, ax = plt.subplots(1, 1, figsize=(20,5))
sns.swarmplot(data=df_merge, x='region', y='poblacion', hue='region', axaxx, log_scale=True)
plt.tight_layout()
plt.show()
```



### EXPORTAMOS DATAFRAME COMBINADO COMO ARCHIVO CSV

```
In [18]: df_merge.to_csv('./data/df_merge_preliminar.csv', index=False)
```