

```
In [29]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sqlalchemy
import mysql.connector
import warnings
import os
import json
warnings.filterwarnings('ignore')
```

Conexión a la Base de Datos

```
In [30]: # Conexión a la Base de Datos
# Valores de Configuración
HOST='localhost'
USER='erikssonherlo'
PASS='201830459'
DATABASE='dataset'

# Establece la conexión
connection = mysql.connector.connect(host=HOST, user=USER, password=PASS, da

# Crea un objeto cursor para ejecutar consultas SQL
cursor = connection.cursor()

# Resumen de Muertes Global
query1 = 'SELECT * FROM death WHERE register_type = 1'
cursor.execute(query1)
results1 = cursor.fetchall()

# Resumen de Muertes por Municipio
query2 = 'SELECT * FROM death WHERE register_type = 2'
cursor.execute(query2)
results2 = cursor.fetchall()

# Cierra el cursor y la conexión
cursor.close()
connection.close()

# Convierte los resultados en DataFrames de Pandas
column_names1 = [i[0] for i in cursor.description]
df_summary = pd.DataFrame(results1, columns=column_names1)

column_names2 = [i[0] for i in cursor.description]
df_municipality = pd.DataFrame(results2, columns=column_names2)
```

Carga de Datos, desde la Base de Datos

Datos del Resumen de Muertes Globales

```
In [31]: # Muestra los DataFrames
print(df_summary)
```

	id	date	country	departament	municipality	population	\
0	1	2020-01-03	Guatemala	TOTAL	TOTAL	17109746	
1	2	2020-01-04	Guatemala	TOTAL	TOTAL	17109746	
2	3	2020-01-05	Guatemala	TOTAL	TOTAL	17109746	
3	4	2020-01-06	Guatemala	TOTAL	TOTAL	17109746	
4	5	2020-01-07	Guatemala	TOTAL	TOTAL	17109746	
...	
93029	179558	2020-12-27	Guatemala	TOTAL	TOTAL	17109746	
93030	179559	2020-12-28	Guatemala	TOTAL	TOTAL	17109746	
93031	179560	2020-12-29	Guatemala	TOTAL	TOTAL	17109746	
93032	179561	2020-12-30	Guatemala	TOTAL	TOTAL	17109746	
93033	179900	2020-12-31	Guatemala	TOTAL	TOTAL	17109746	

	new_deaths	cumulative_deaths	register_type
0	0	0	summary
1	0	0	summary
2	0	0	summary
3	0	0	summary
4	0	0	summary
...
93029	6	4763	summary
93030	5	4768	summary
93031	5	4773	summary
93032	8	4781	summary
93033	22	4803	summary

[93034 rows x 9 columns]

Datos de Muertes por Municipios

```
In [32]: print(df_municipality)
```

	id	date	country	departament	municipalit
y \	73	2020-03-15	Guatemala	HUEHUETENANGO	TODOS SANTOS CUCHUMATA
0	74	2020-03-15	Guatemala	HUEHUETENANGO	BARILLA
N	75	2020-03-15	Guatemala	QUETZALTENANGO	CAJOL
1	76	2020-03-15	Guatemala	JUTIAPA	EL PROGRES
S	77	2020-03-15	Guatemala	HUEHUETENANGO	SAN MIGUEL ACATA
2
A	86861	179895	2020-12-31	Guatemala	CHIQUIMULA
3	86862	179896	2020-12-31	Guatemala	JUTIAPA
0	86863	179897	2020-12-31	Guatemala	ZACAPA
4	86864	179898	2020-12-31	Guatemala	TOTONICAPAN
N	86865	179899	2020-12-31	Guatemala	BAJA VERAPAZ
...
...	86861	23435	0	0	municipality
A	86862	36047	0	0	municipality
A	86863	38270	0	0	municipality
N	86864	25461	0	0	municipality
A	86865	35895	0	0	municipality
J

	population	new_deaths	cumulative_deaths	register_type
0	38303	0	0	municipality
1	122466	0	0	municipality
2	18896	0	0	municipality
3	25189	0	0	municipality
4	35071	0	0	municipality
...
86861	23435	0	0	municipality
86862	36047	0	0	municipality
86863	38270	0	0	municipality
86864	25461	0	0	municipality
86865	35895	0	0	municipality

[86866 rows x 9 columns]

EDA Monovariable

El análisis exploratorio de datos (EDA) monovariable realizado revela información valiosa sobre diversas variables cuantitativas y cualitativas en el conjunto de datos. A continuación, se presenta un resumen descriptivo de los hallazgos:

Datos Cuantitativos

Variables Cuantitativa - Nuevas Muertes (new_deaths):

Estadísticos Descriptivos:

- Descripción

```
In [33]: # Variables Cuantitativas
df_new_deaths = df_summary[['new_deaths']]
df_new_deaths.describe()
```

```
Out[33]:
```

new_deaths	
count	93034.000000
mean	13.273481
std	14.776865
min	0.000000
25%	0.000000
50%	9.000000
75%	22.000000
max	61.000000

- Valores Unicos:

```
In [34]: df_summary['new_deaths'].unique()
```

```
Out[34]: array([ 0,  1,  2,  3,  5,  4, 12, 10,  6,  8,  7, 20, 15, 58, 14, 22, 27,
        18, 17, 16, 19, 34, 31, 35, 49, 21, 44, 26, 37, 40, 23, 39, 47, 33,
        25, 48, 54, 53, 29, 42, 59, 30, 32, 57, 38, 24, 11, 45, 28, 13,  9,
        61, 41])
```

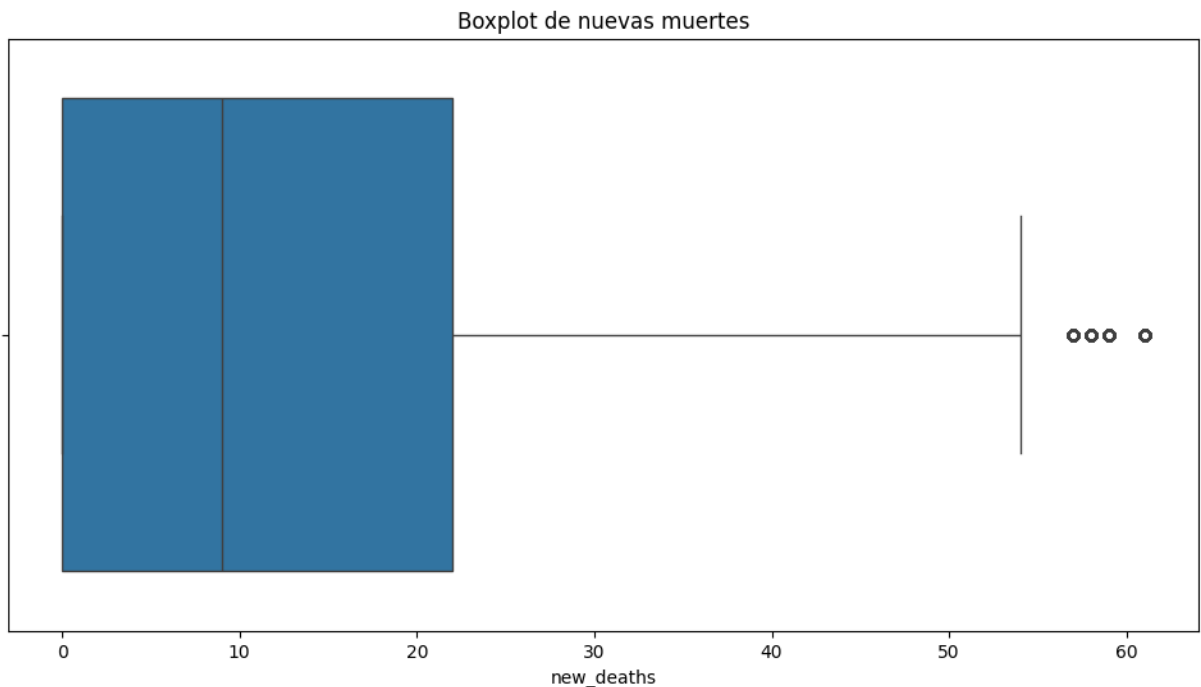
Grafica de Caja (Boxplot):

Un boxplot es una representación gráfica que nos proporciona información sobre la distribución de un conjunto de datos.

Elementos del Boxplot

- Caja (Box): La caja del boxplot representa el rango intercuartílico (IQR), que es la distancia entre el primer cuartil (Q1) y el tercer cuartil (Q3). La línea dentro de la caja indica la mediana (Q2).
- Bigotes (Whiskers): Los bigotes se extienden desde la caja hasta los valores más extremos dentro de un cierto rango llamado "longitud del bigote". Los valores fuera de este rango se consideran "valores atípicos" y se representan como puntos individuales.
- Líneas Adicionales: Pueden haber líneas adicionales dentro del boxplot para representar valores específicos o umbrales.

```
In [35]: # Palette
my_palette = ['#221f1f', '#b20710', '#e50914', '#f5f5f1']
plt.figure(figsize=(12, 6))
ax = sns.boxplot(x='new_deaths', data=df_new_deaths, orient='h')
ax.set_title('Boxplot de nuevas muertes')
plt.show()
```



Descripción del Boxplot

- Inicio en 0 y Finalización en 25: La caja y los bigotes se extienden desde 0 hasta 20, cubriendo el rango completo de datos.
- Línea entre 0 y 10: Dentro del boxplot, hay una línea horizontal que representa algún valor específico o umbral. En este caso, la línea está entre 0 y 10. Esto podría ser una referencia importante, como un límite crítico o un valor de interés.

Interpretación Visual

En la visualización del boxplot, podemos observar una caja que abarca desde 0 hasta 25 en el eje y. En nuestro caso, al no haber valores atípicos, no se visualiza ningún tipo de punteo más allá de los bigotes. Además, podemos notar una línea vertical entre 0 y 10, que podría indicar un rango específico de interés o algún límite relevante. En nuestro caso, dicho valor es la mediana, que se sitúa en 22, lo que indica que es el valor medio de la distribución, y es un valor importante para comprender la ubicación central de nuestros datos.

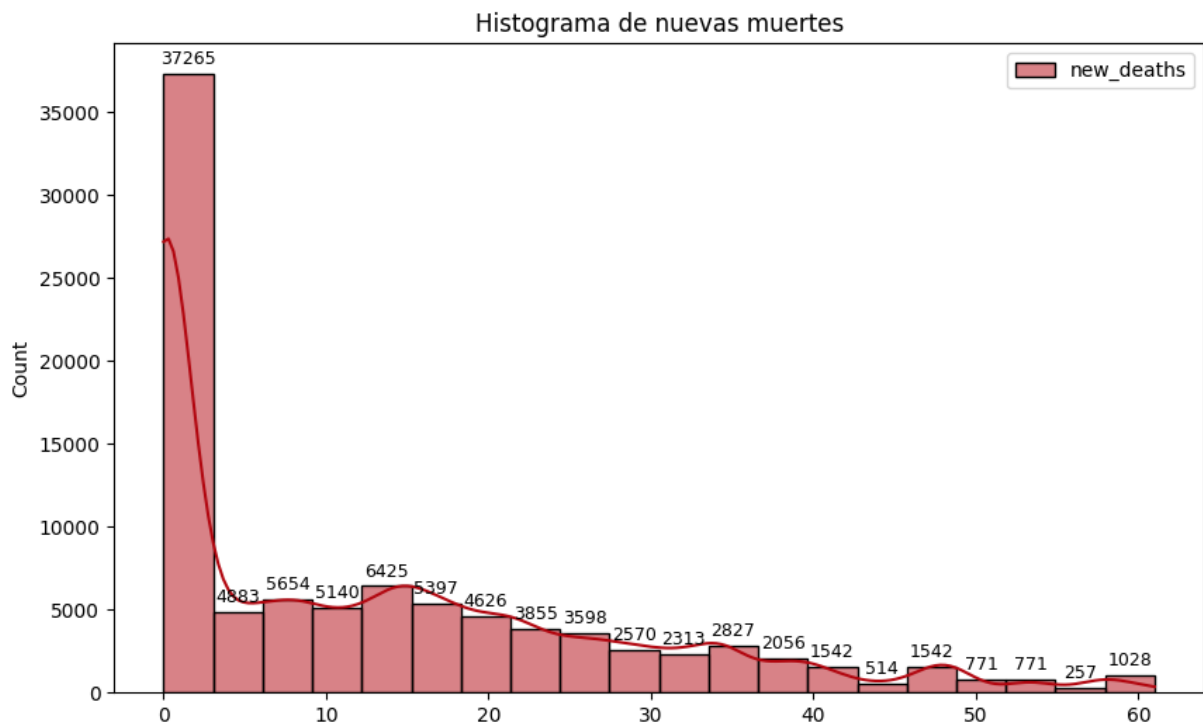
Transformación: Si bien es cierto, podemos ver que fuera de los bigotes se tienen algunos valores atípicos, estos nos muestran los famosos picos, los cuales

indican aumentos significativos de un día para otro en los valores de las muertes, por lo tanto no se realiza ningún tipo de transformación.

Histograma

- Eje X (Horizontal): Representa los valores posibles de la variable que estás observando (en nuestro caso, 'Nuevas Muertes').
- Eje Y (Vertical): Indica la frecuencia o densidad de ocurrencia de esos valores.

```
In [36]: plt.figure(figsize=(10, 6))
ax = sns.histplot(df_new_deaths, kde=True, bins=20, palette=my_palette[1:])
for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2.,
p.get_height()),
ha='center', va='baseline', fontsize=9, color='black',
xytext=(0, 5),
textcoords='offset points')
ax.set_title('Histograma de nuevas muertes')
plt.show()
```



Visualización de la Información: Podemos observar dónde se concentran la mayoría de las nuevas muertes.

- Si hay una cola hacia la derecha, podría haber casos de días con un gran número de nuevas muertes.
- La mediana (valor medio) se encuentra en la región más alta del histograma.

En resumen, el histograma nos brinda una representación visual de la distribución de tus datos, ayudándote a identificar patrones, tendencias y características importantes.

Variables Cuantitativa - Muertes Acumuladas (Cumulative Deaths):

Estadísticos Descriptivos:

- **Descripción**

```
In [37]: # Variables Cuantitativas
df_cumulative_deaths = df_summary[['cumulative_deaths']]
df_cumulative_deaths.describe()
```

Out[37]:

	cumulative_deaths
--	-------------------

count	93034.000000
mean	1604.917127
std	1694.676097
min	0.000000
25%	1.000000
50%	830.000000
75%	3261.000000
max	4803.000000

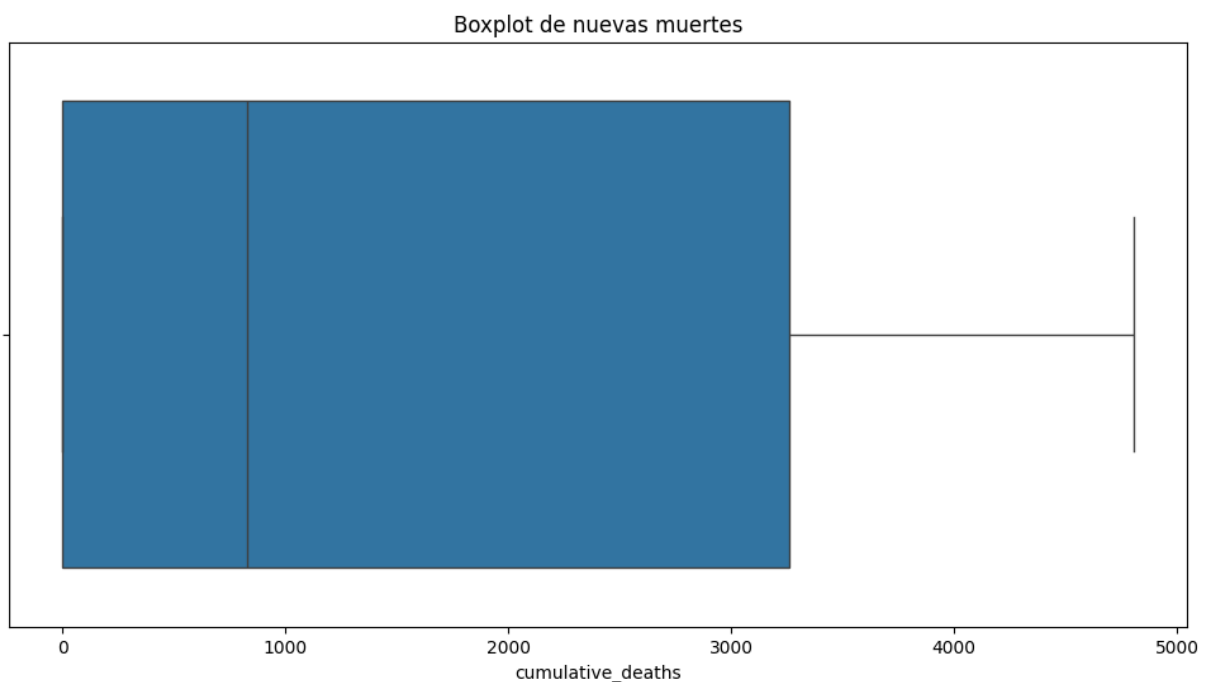
- **Valores Unicos**

```
In [38]: df_summary['cumulative_deaths'].unique()
```

```
Out[38]: array([ 0,  1,  2,  3,  5,  7,  9, 10, 11, 13, 15,
                16, 18, 17, 19, 21, 23, 24, 26, 27, 29, 30,
                33, 35, 38, 43, 45, 48, 53, 57, 58, 59, 63,
                68, 80, 90, 102, 108, 116, 123, 143, 158, 216, 230,
                252, 267, 289, 316, 334, 351, 367, 384, 399, 418, 432,
                449, 483, 514, 531, 547, 582, 601, 623, 672, 706, 727,
                746, 773, 817, 843, 880, 920, 947, 981, 1004, 1053, 1092,
                1139, 1172, 1219, 1244, 1302, 1350, 1404, 1443, 1449, 1502, 1531,
                1573, 1632, 1669, 1699, 1734, 1761, 1782, 1835, 1867, 1924, 1957,
                1995, 2013, 2037, 2072, 2119, 2168, 2197, 2211, 2222, 2233, 2267,
                2296, 2341, 2355, 2379, 2389, 2419, 2467, 2506, 2532, 2580, 2594,
                2611, 2630, 2662, 2685, 2709, 2728, 2740, 2760, 2778, 2790, 2804,
                2825, 2845, 2852, 2862, 2890, 2897, 2918, 2929, 2949, 2957, 2972,
                2984, 3009, 3036, 3076, 3105, 3119, 3124, 3137, 3154, 3170, 3186,
                3213, 3229, 3234, 3238, 3246, 3261, 3267, 3285, 3293, 3302, 3310,
                3335, 3347, 3356, 3365, 3384, 3387, 3410, 3430, 3453, 3478, 3515,
                3530, 3541, 3546, 3567, 3580, 3594, 3609, 3644, 3651, 3665, 3682,
                3704, 3714, 3729, 3738, 3748, 3752, 3766, 3794, 3811, 3821, 3823,
                3832, 3845, 3858, 3880, 3920, 3932, 3938, 3947, 4008, 4050, 4067,
                4074, 4076, 4092, 4099, 4107, 4133, 4141, 4161, 4166, 4171, 4178,
                4191, 4209, 4224, 4239, 4250, 4274, 4286, 4311, 4345, 4376, 4405,
                4423, 4445, 4476, 4510, 4551, 4589, 4624, 4656, 4688, 4718, 4739,
                4749, 4757, 4763, 4768, 4773, 4781, 4803])
```

Boxplot

```
In [39]: #BoxPlot
plt.figure(figsize=(12, 6))
ax = sns.boxplot(x='cumulative_deaths', data=df_cumulative_deaths, orient='h')
ax.set_title('Boxplot de nuevas muertes')
plt.show()
```

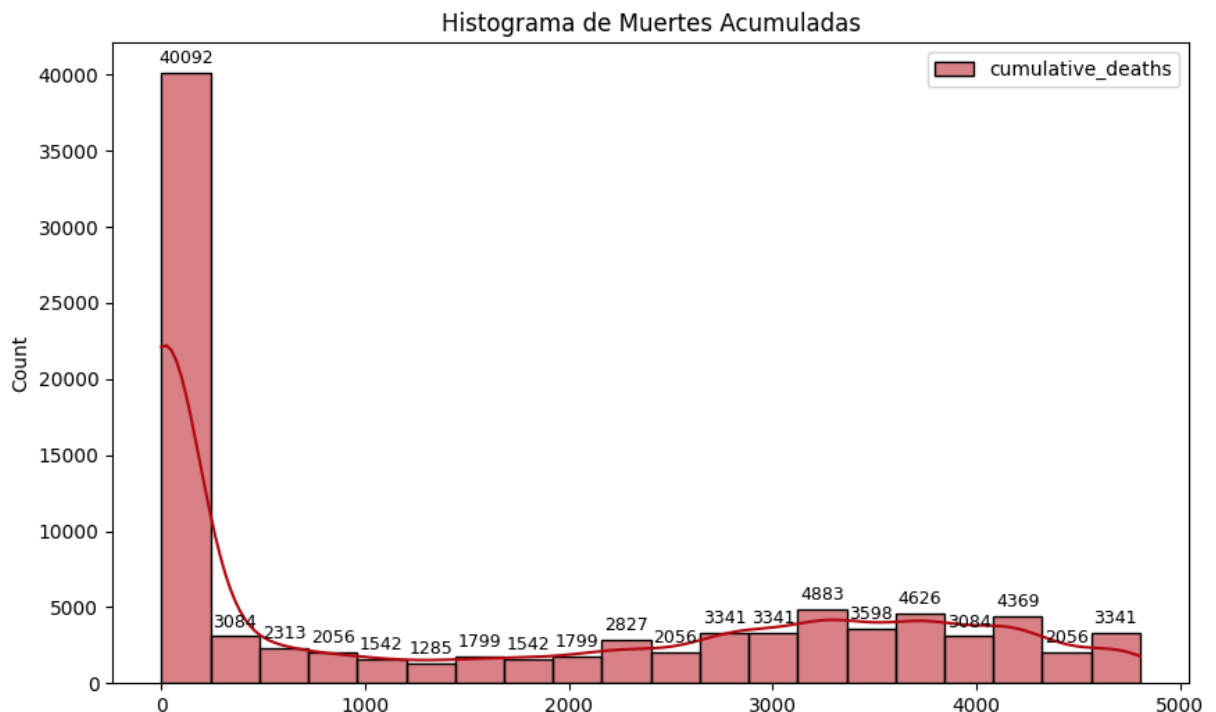


Interpretación Visual

En la visualización del boxplot, podemos observar una caja que abarca desde 0 hasta mas de 3000 en el eje y. En nuestro caso, al no haber valores atípicos, no se visualiza ningun tipo de punteo más allá de los bigotes. Además, podemos observar una línea vertical entre 0 y 1000, que podría indicar un rango específico de interés o algún límite relevante. En nuestro caso, dicho valor es la mediana, que se situa en 776, lo que indica que es el valor medio de la distribución, y es un valor importante para comprender la ubicación central de nuestros datos

Histograma

```
In [40]: plt.figure(figsize=(10, 6))
ax = sns.histplot(df_cumulative_deaths, kde=True, bins=20, palette=my_palette)
for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2.,
    p.get_height()),
    ha='center', va='baseline', fontsize=9, color='black',
    xytext=(0, 5),
    textcoords='offset points')
ax.set_title('Histograma de Muertes Acumuladas')
plt.show()
```



Podemos visualizar que los datos de variables acumuladas se mantuvieron durante varios días, por lo que significa que no existieron cambios significativos dentro de dichos días, en cuanto a muertes acumuladas.

Variables Cuantitativa - Población de Municipios (Population):

Estadísticos Descriptivos:

- **Descripción**

```
In [41]: df_municipality_population = df_municipality[['population']]  
df_municipality_population.describe()
```

```
Out[41]:
```

	population
count	8.686600e+04
mean	4.982360e+04
std	8.138646e+04
min	0.000000e+00
25%	1.739300e+04
50%	3.086200e+04
75%	5.852600e+04
max	1.205668e+06

- **Valores Unicos**

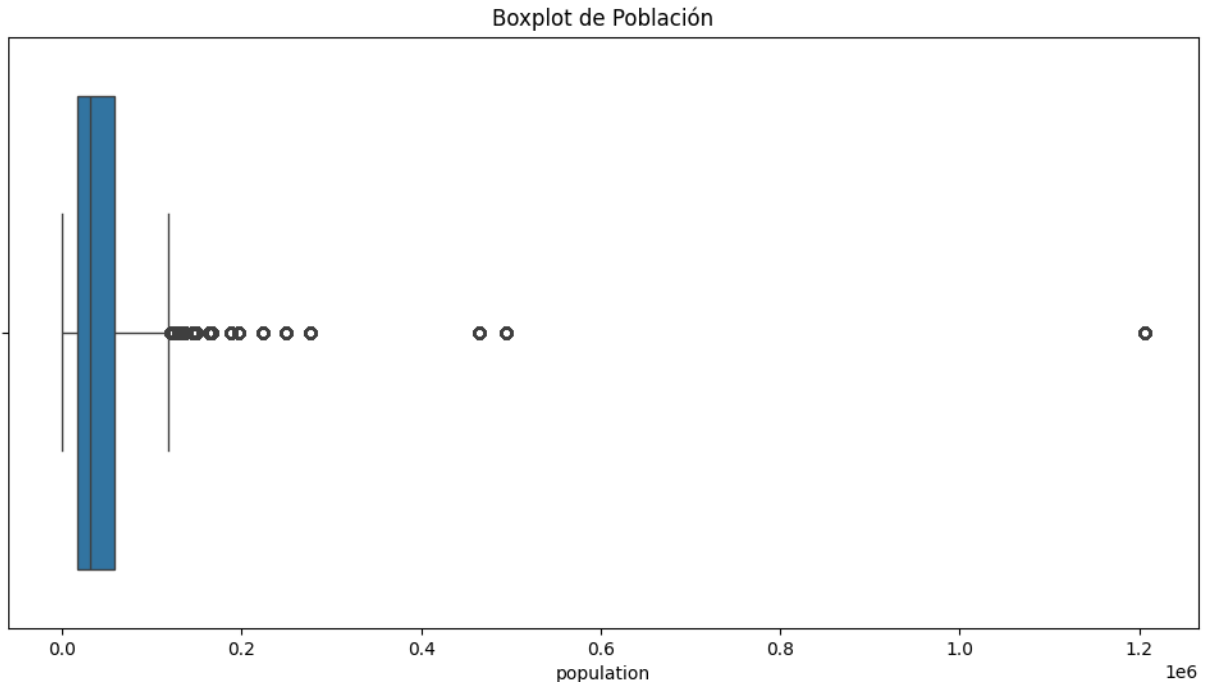
```
In [42]: df_municipality['population'].unique()
```

```
Out[42]: array([[ 38303, 122466, 18896, 25189, 35071, 8766, 13971,
 32038, 27591, 28890, 50093, 26146, 75979, 30134,
 26170, 23394, 50171, 44819, 10200, 58526, 46350,
 35616, 17811, 42667, 72955, 81005, 30912, 7103,
 10229, 14380, 51770, 20579, 10225, 51975, 37419,
 11870, 66883, 39350, 30017, 4480, 81385, 8724,
 13785, 10859, 10996, 30444, 35895, 18222, 58174,
 19514, 13811, 23435, 36047, 38270, 25461, 12577,
 5128, 97296, 76128, 79777, 13384, 15639, 91523,
 23499, 17393, 9088, 14440, 29121, 18085, 7851,
 8317, 16339, 12261, 8139, 25716, 39269, 32963,
 13620, 60848, 6861, 11057, 54382, 26714, 9238,
 0, 9652, 21416, 8272, 7817, 7462, 10340,
 15586, 9607, 7950, 23858, 12599, 50814, 30205,
 46629, 28097, 22423, 20384, 27522, 6706, 57277,
 17322, 14522, 20299, 42375, 64292, 19778, 16786,
 41035, 34823, 12509, 21695, 10341, 13786, 29283,
 23166, 104861, 10646, 2563, 25332, 18291, 21855,
 12374, 13124, 25859, 2911, 24580, 9361, 54623,
 25475, 12573, 12674, 87912, 17957, 163665, 25165,
 20245, 98886, 69107, 26560, 17465, 92103, 80234,
 112778, 86150, 25479, 65178, 65594, 45937, 48644,
 26191, 84350, 17833, 117579, 29659, 27001, 51253,
 60376, 41997, 79360, 42601, 29285, 30812, 50348,
 18128, 63897, 46489, 13803, 23021, 45434, 24199,
 32104, 22048, 22012, 61547, 23851, 13535, 34591,
 148943, 31500, 30973, 43067, 117014, 109753, 165026,
 248880, 123979, 40493, 494561, 464528, 65730, 196867,
 64441, 8519, 58338, 166078, 61664, 13508, 69560,
 54588, 70118, 1205668, 28445, 117483, 15231, 83448,
 145417, 17569, 38981, 13294, 7945, 44025, 26472,
 73811, 47441, 108405, 11980, 108134, 67994, 36409,
 17116, 121721, 59496, 16679, 45249, 276836, 147604,
 108893, 60200, 14121, 66314, 70066, 91954, 97974,
 49299, 38386, 15835, 62329, 76047, 12905, 20032,
 11948, 97003, 59788, 20851, 7544, 83375, 79308,
 129232, 37283, 15840, 29377, 57977, 16817, 15225,
 36117, 49410, 41552, 31581, 82876, 76020, 12330,
 8658, 32771, 29356, 11600, 73093, 19080, 41304,
 136909, 21906, 17234, 55723, 27307, 14640, 39337,
 26686, 45162, 13071, 43622, 54845, 62895, 63841,
 51299, 24289, 15414, 17923, 37915, 43512, 13123,
 27787, 224109, 42342, 79867, 48188, 19693, 23160,
 64224, 45870, 23030, 12438, 23311, 19389, 52439,
 55133, 60791, 28043, 60355, 33764, 51024, 22618,
 40083, 104428, 55290, 42354, 107265, 11635, 38006,
 41984, 34866, 21088, 188522, 26350, 12569, 27567,
 38219, 30776, 28473, 25677, 41224, 106156, 45229,
 34981])
```

Boxplot

```
In [43]: #BoxPlot
plt.figure(figsize=(12, 6))
ax = sns.boxplot(x='population', data=df_municipality_population, orient='h')
```

```
ax.set_title('Boxplot de Población')
plt.show()
```

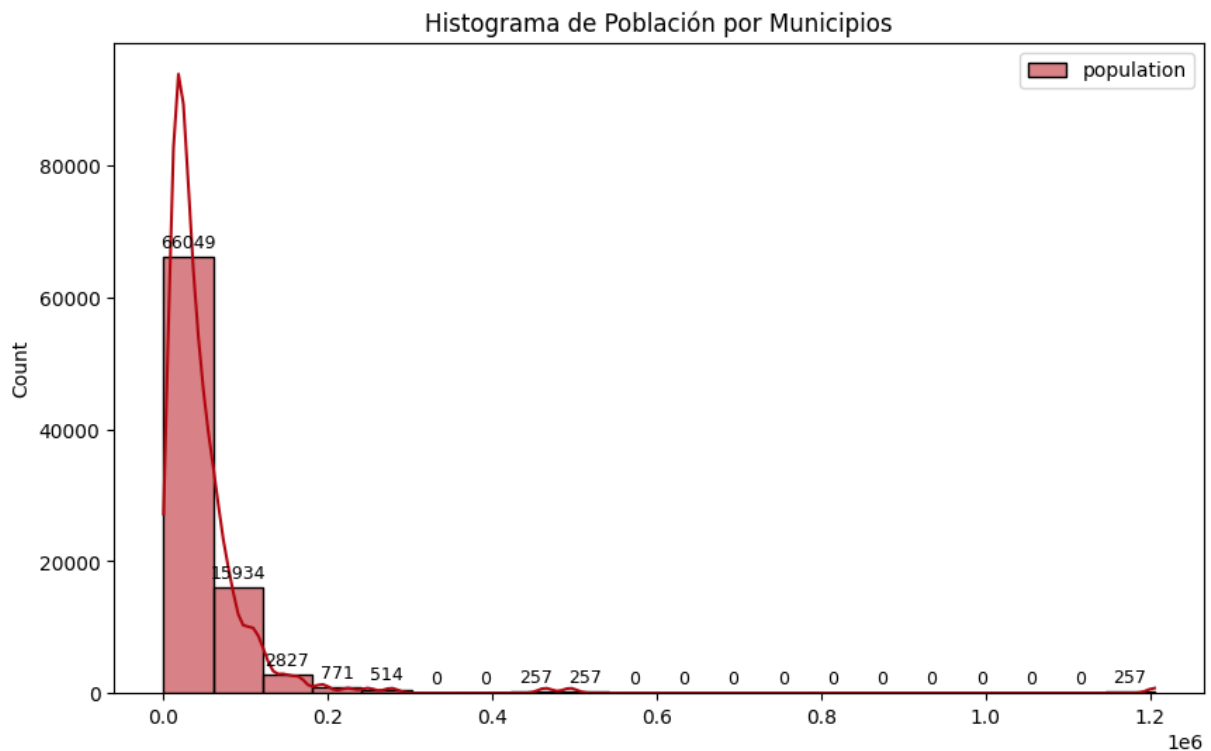


El Boxplot contiene datos atípicos, que dado el contexto geográfico indican que hay regiones con una alta densidad poblacional, lo que provoca que la información no pueda ser visualizada de manera correcta.

Transformación: a pesar de existir dichos valores atípicos, estos no pueden sufrir ningún tipo de transformación, ya que esto los alejaría de la representación fidedigna de la realidad.

Histograma

```
In [44]: plt.figure(figsize=(10, 6))
ax = sns.histplot(df_municipality_population, kde=True, bins=20, palette=my_
for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2.,
p.get_height()),
ha='center', va='baseline', fontsize=9, color='black',
xytext=(0, 5),
textcoords='offset points')
ax.set_title('Histograma de Población por Municipios')
plt.show()
```



Al igual que con el boxplot, al haber datos atípicos, no podemos visualizar correctamente el histograma

Datos Cualitativos

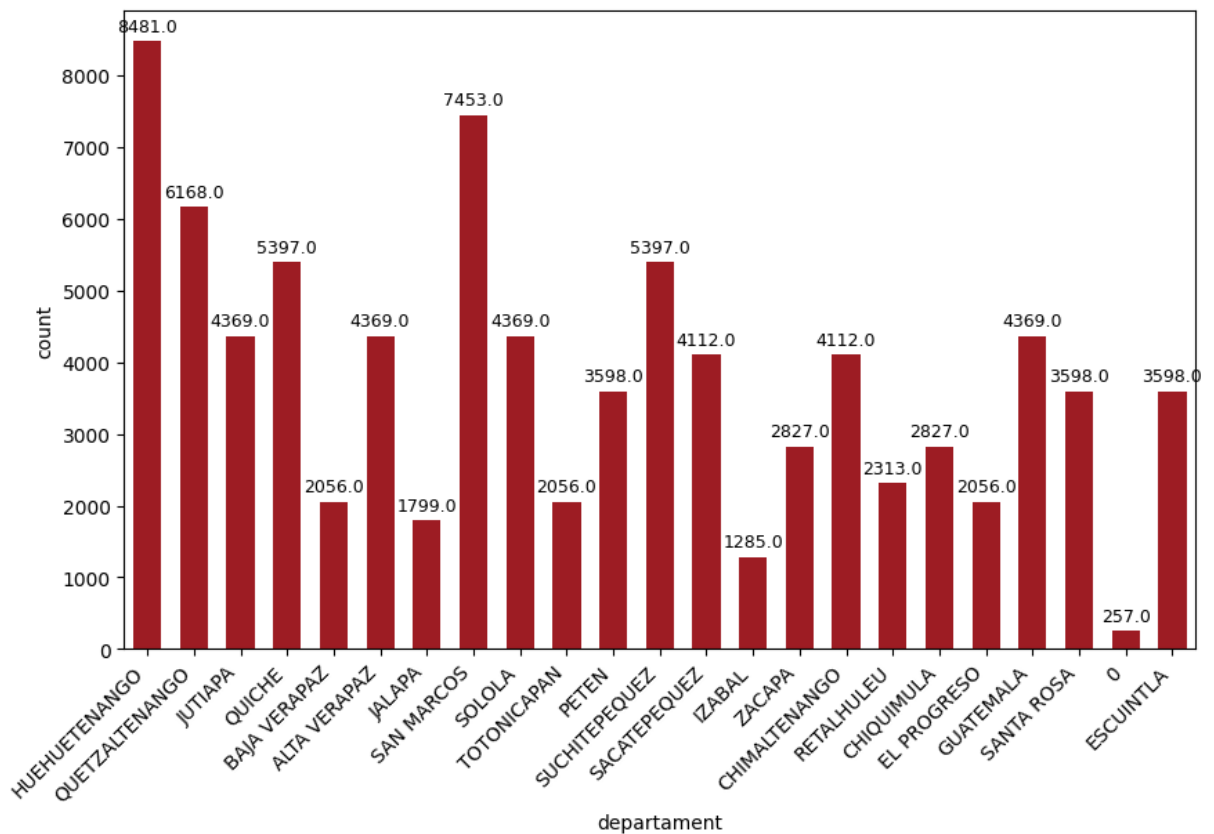
Variables Cualitativa - Departamentos (Departaments)

Diagrama de Barras:

```
In [45]: plt.figure(figsize=(10, 6))
ax = sns.countplot(x='departament', data=df_municipality, width=0.6, palette=
for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.get_
        ha='center', va='baseline', fontsize=9, color='black', xytext=
        textcoords='offset points')

ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha="right")

plt.show()
```



Es posible observar que los 2 departamentos con mas registros son:

- Huehuetenango
- San Marcos

Esto se debe a que son los 2 departamentos que tienen mayor numero de municipios en el pais.

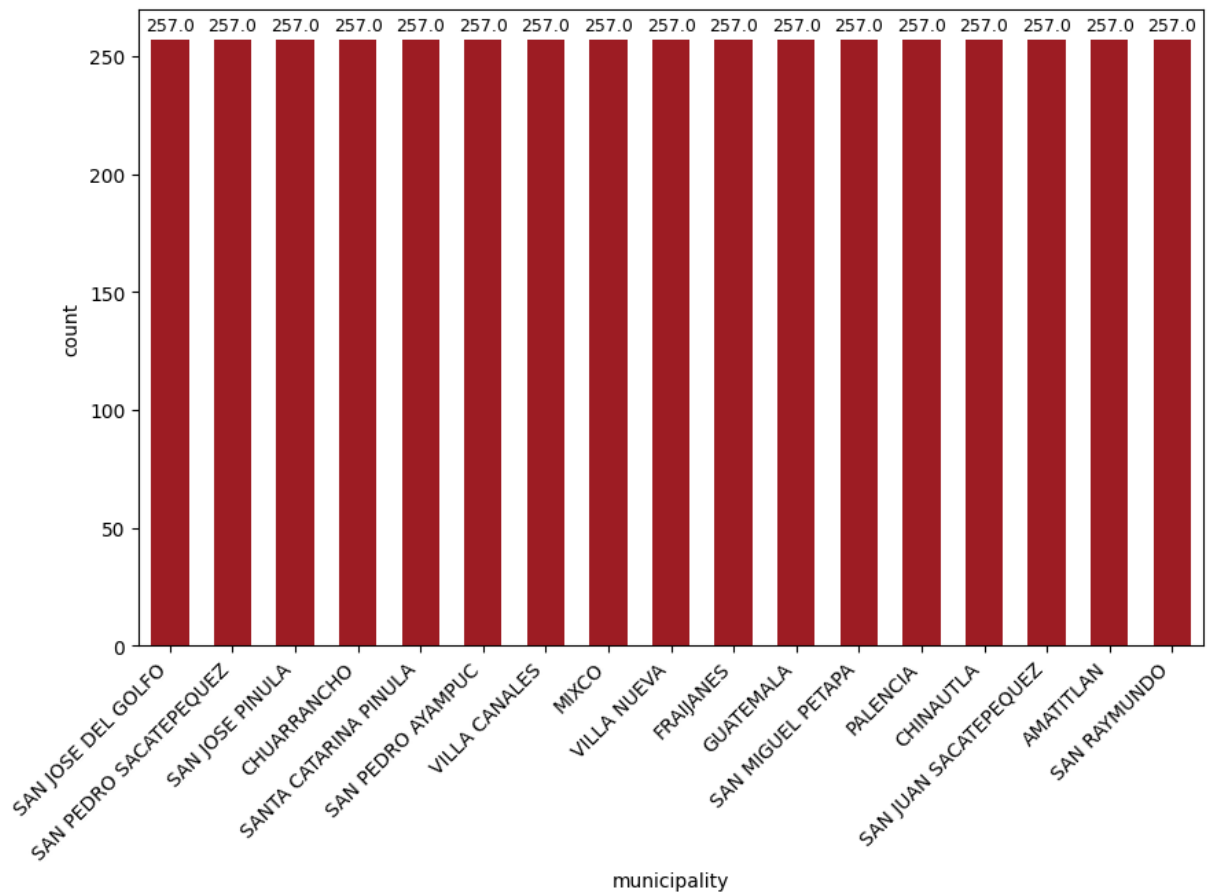
Nota: esta grafica no representa la cantidad de muertes por departamento, sino la cantidad de registros existentes por departamento.

Diagrama de Barras - Municipios

```
In [46]: df_departament_GT = df_municipality[df_municipality['departament'] == 'GUATEMALA']
plt.figure(figsize=(10, 6))
ax = sns.countplot(x='municipality', data=df_departament_GT, width=0.6, palette='magma')

for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='baseline', fontsize=9, color='black', xytext=(p.get_x() + p.get_width() / 2., p.get_height() + 5),
                textcoords='offset points')

ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha="right")
plt.show()
```

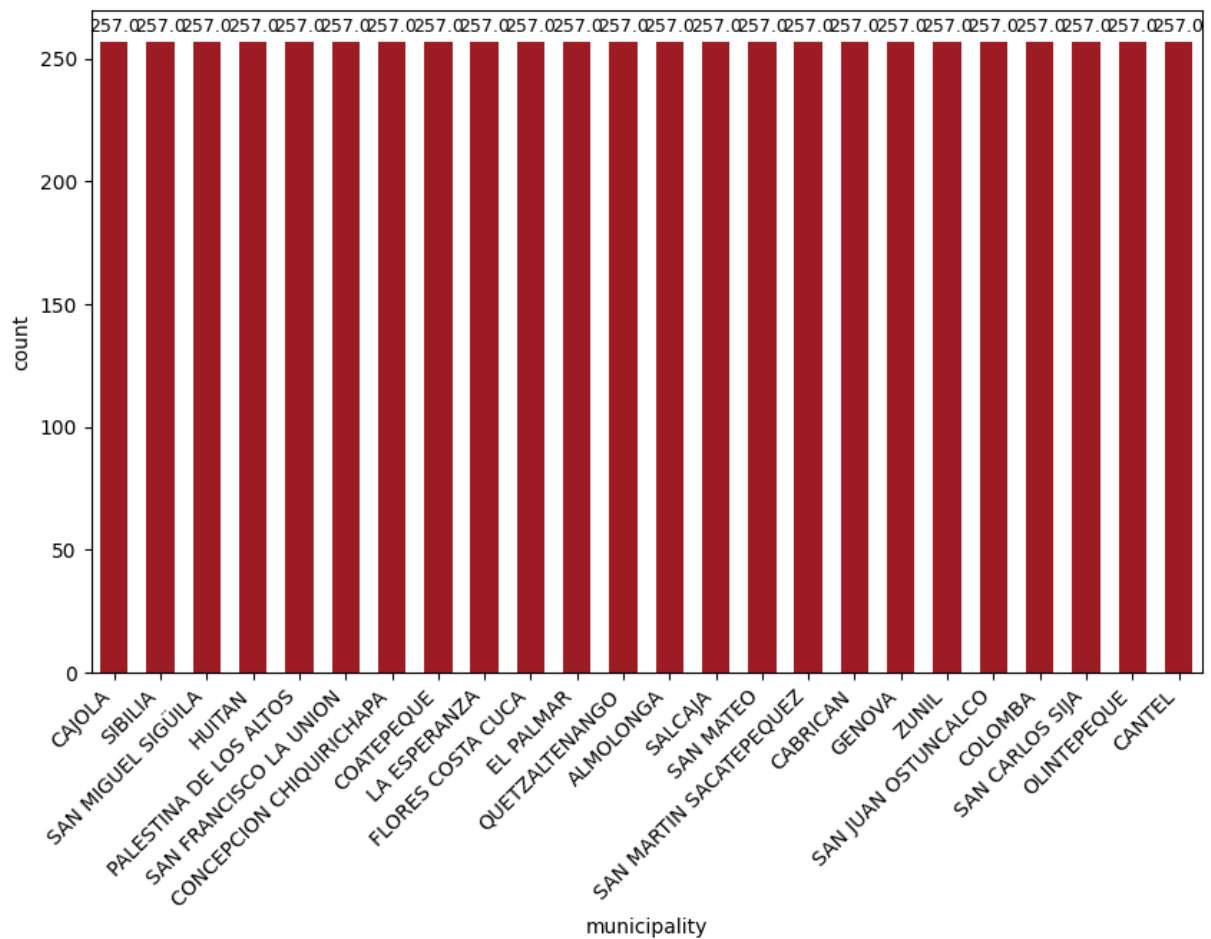


```
In [47]: df_departament_GT = df_municipality[df_municipality['departament'] == 'QUETZ
plt.figure(figsize=(10, 6))
ax = sns.countplot(x='municipality', data=df_departament_GT, width=0.6, pale

for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.get_
                ha='center', va='baseline', fontsize=9, color='black', xytex
                textcoords='offset points')

ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha="right")

plt.show()
```

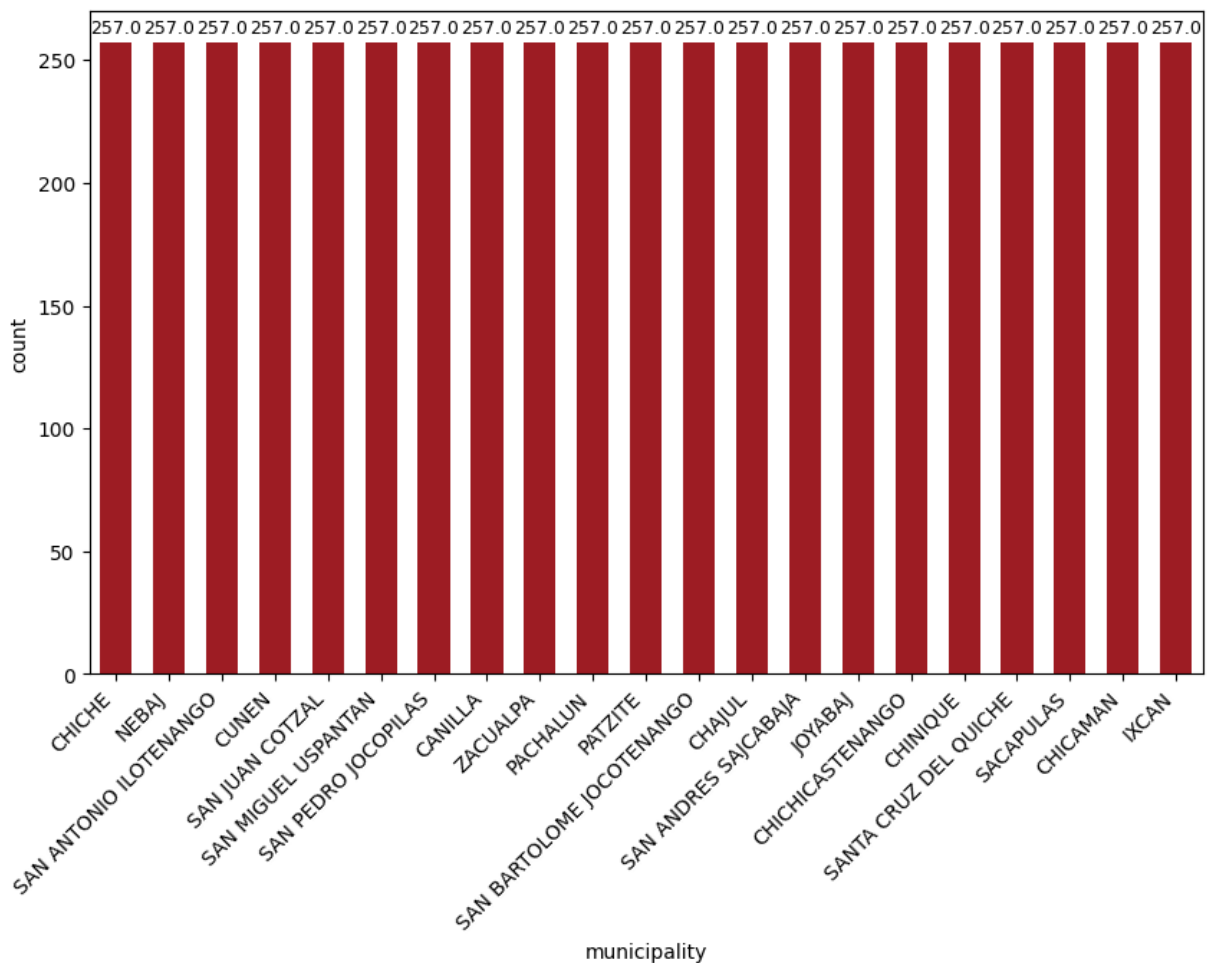


```
In [48]: df_departament_GT = df_municipality[df_municipality['departament'] == 'QUICH
plt.figure(figsize=(10, 6))
ax = sns.countplot(x='municipality', data=df_departament_GT, width=0.6, pale

for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.get_
        ha='center', va='baseline', fontsize=9, color='black', xytex
        textcoords='offset points')

ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha="right")

plt.show()
```

Analisis de Datos Cualitativos sobre Municipios

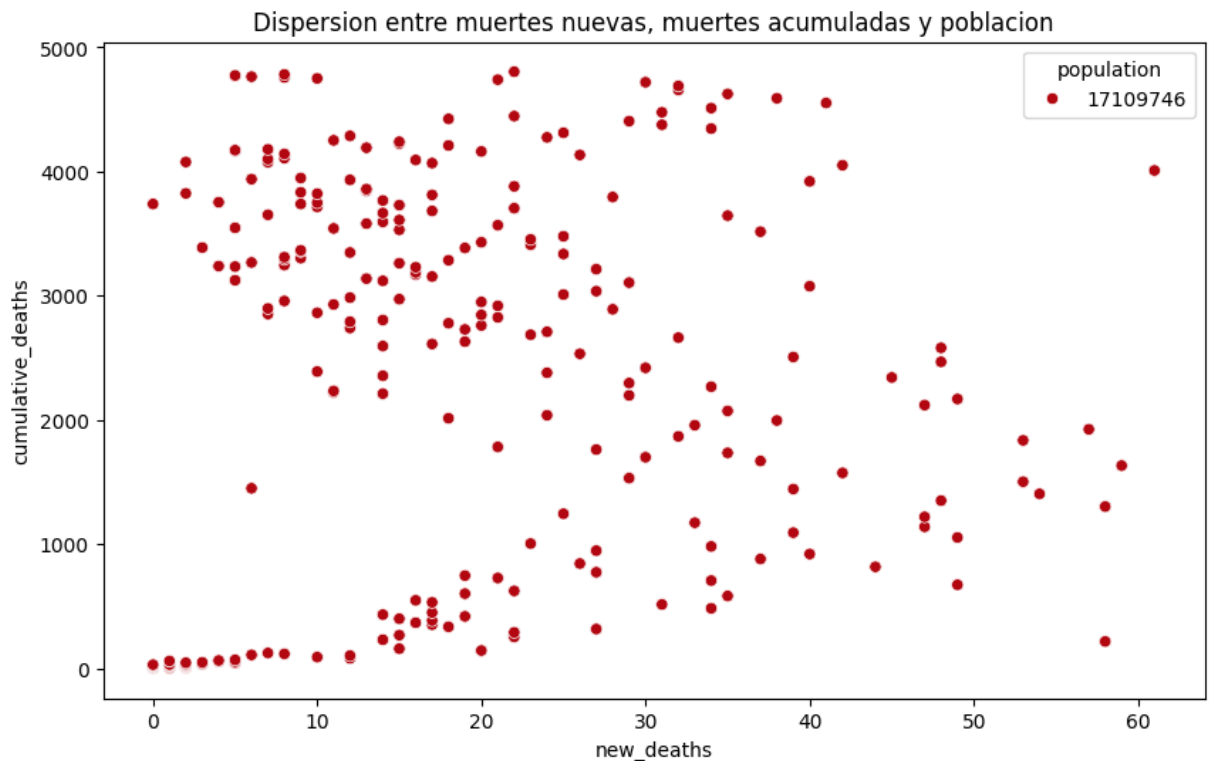
El conteo de los registros por municipio es de 257 para todos, ya que los datos proporcionados fueron datos de muertes por fecha para todo un año en específico, sin embargo algunos datos faltantes evitan poder visualizar toda la línea de tiempo, por lo que las graficas de los municipios para los siguientes departamentos fueron: Guatemala , Quetzaltengo y Quiché

Esto con la finalidad de visualizar la cantidad de muertes, para cada registro de cada departamento del interior y la ciudad capital

Analisis EDA MultiVariable

Dispersion entre variables muertes, muertes acumuladas y poblacion

```
In [49]: plt.figure(figsize=(10, 6))
ax = sns.scatterplot(x='new_deaths', y='cumulative_deaths', data=df_summary,
ax.set_title('Dispersion entre muertes nuevas, muertes acumuladas y poblacion
plt.show()
```



Observación: En la grafica se puede observar las muertes se dieron en cantidades menores y de manera continua al inicio de la pandemia; es decir cuando las muertes acumuladas eran menores, mientras que las cantidades mas grandes de muertes se dieron cuando las muertes acumuladas eran mayores.

Mapa de calor entre variables muertes, muertes acumuladas y poblacion

```
In [50]: plt.figure(figsize=(10, 6))
ax = sns.heatmap(df_summary[['new_deaths', 'cumulative_deaths', 'population']])
ax.set_title('Mapa de calor entre muertes nuevas, muertes acumuladas y poblacion')
plt.show()
```



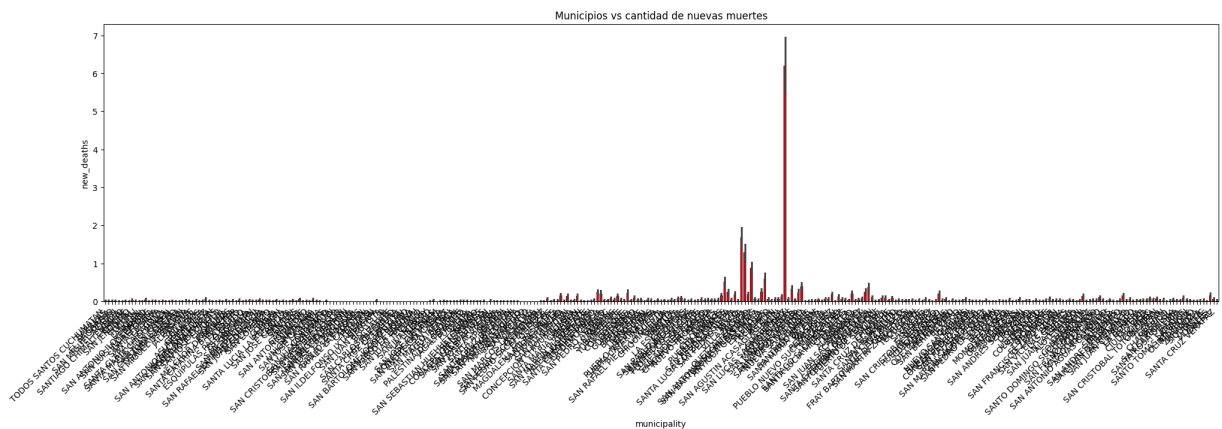
Analisis sobre mapa de calor entre variables muertes, muertes acumuladas y poblacion

En el mapa no se puede observar con claridad la relación directa entre la población y las variables de mortandad, sin embargo podemos concluir que la población tiene una correlacion alta con ambas variables, sin embargo la correlacion mayor es con las muertes acumuladas. Por su parte correlacion entre muertes y muertes acumuladas es moderada.

Estas correlaciones nos indican que a mayor población mayor cantidad de muertes acumuladas y nuevas muertes.

Gráfico de barras: Municipios vs cantidad de nuevas muertes

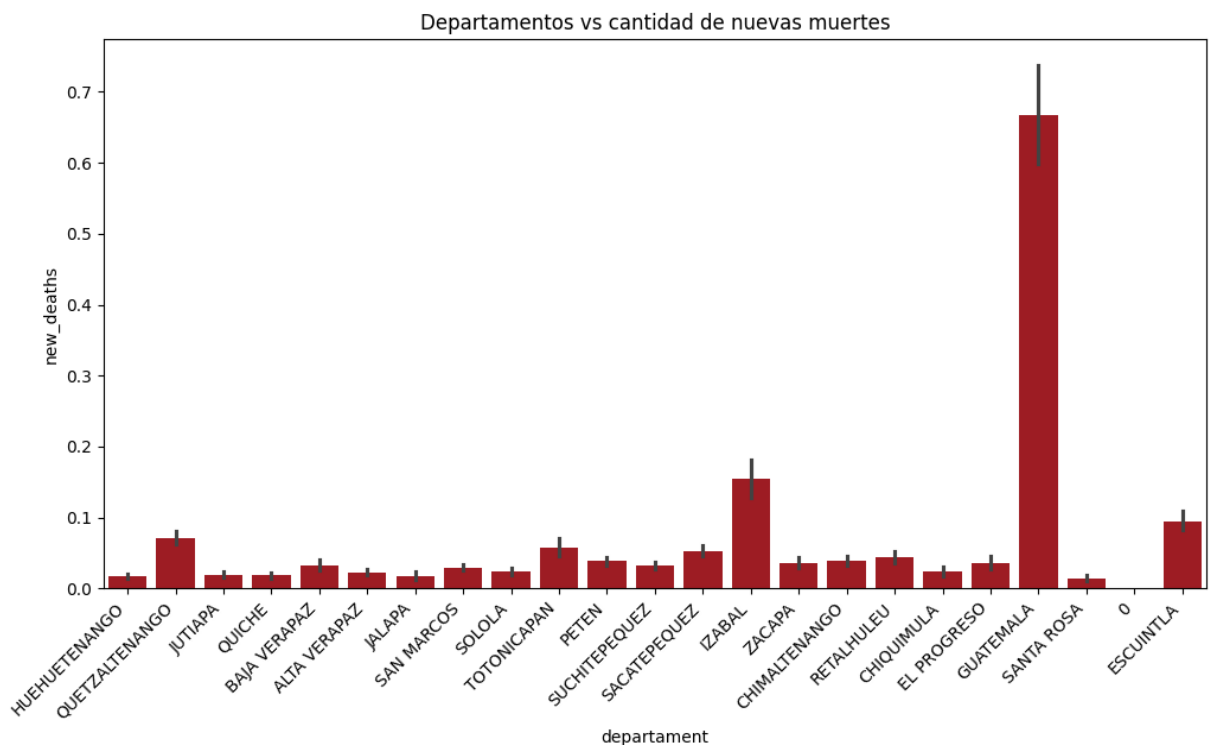
```
In [51]: plt.figure(figsize=(24, 6))
ax = sns.barplot(x='municipality', y='new_deaths', data=df_municipality, pal
ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha="right")
ax.set_title('Municipios vs cantidad de nuevas muertes')
plt.show()
```



Dicha grafica es muy complicada de interpretar, debido a la gran cantidad de datos que presenta

Grafica de barras: Departamentos vs cantidad de nuevas muertes

```
In [52]: plt.figure(figsize=(12, 6))
ax = sns.barplot(x='departament', y='new_deaths', data=df_municipality, palette='magma')
ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha="right")
ax.set_title('Departamentos vs cantidad de nuevas muertes')
plt.show()
```



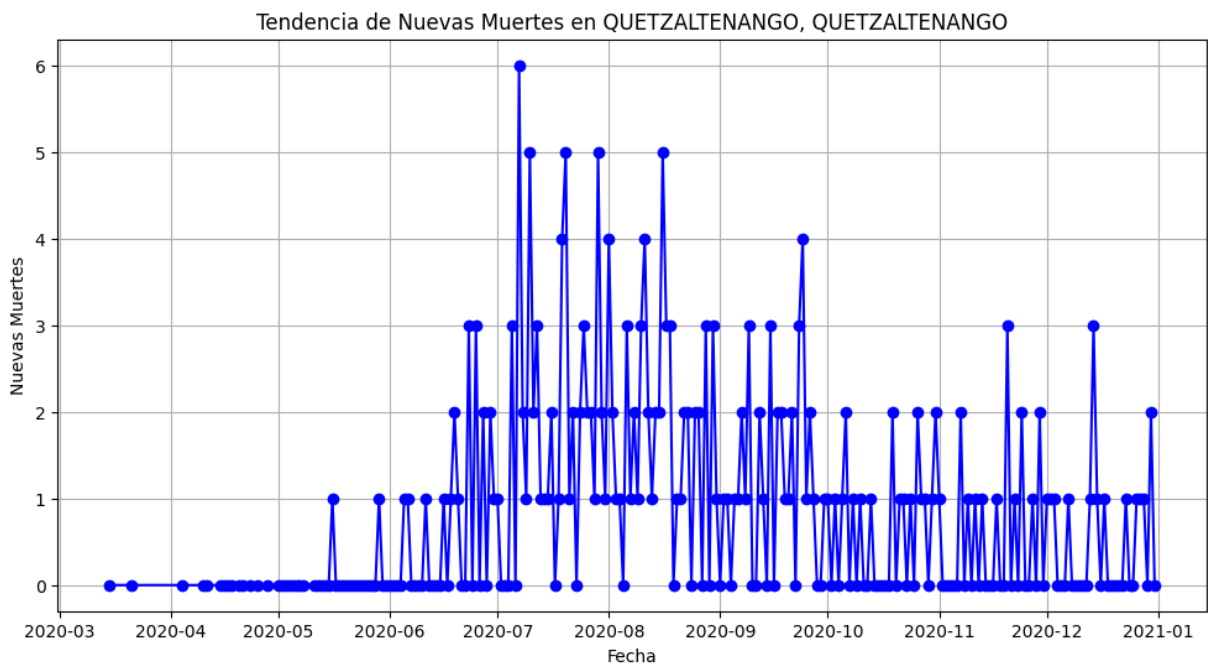
Observaciones: Es posible observar que el departamento con mayor cantidad de muertes, es el departamento de Guatemala, esto se debe a que es uno de los municipios con mayor densidad poblacional.

```
In [53]: # Filtra los datos para un municipio y departamento específicos
municipio_especifico = 'QUETZALTENANGO'
departamento_especifico = 'QUETZALTENANGO'
df_municipio_departamento = df_municipality[(df_municipality['municipality'] == municipio_especifico) & (df_municipality['department'] == departamento_especifico)]

# Convierte la columna 'date' a tipo datetime para asegurarte de que esté en el formato correcto
df_municipio_departamento['date'] = pd.to_datetime(df_municipio_departamento['date'])

# Ordena el DataFrame por fecha para asegurarte de que la gráfica muestre la tendencia correctamente
df_municipio_departamento = df_municipio_departamento.sort_values('date')

# Crea la gráfica de tendencia
plt.figure(figsize=(12, 6))
plt.plot(df_municipio_departamento['date'], df_municipio_departamento['new_deaths'])
plt.title(f'Tendencia de Nuevas Muertes en {municipio_especifico}, {departamento_especifico}')
plt.xlabel('Fecha')
plt.ylabel('Nuevas Muertes')
plt.grid(True)
plt.show()
```

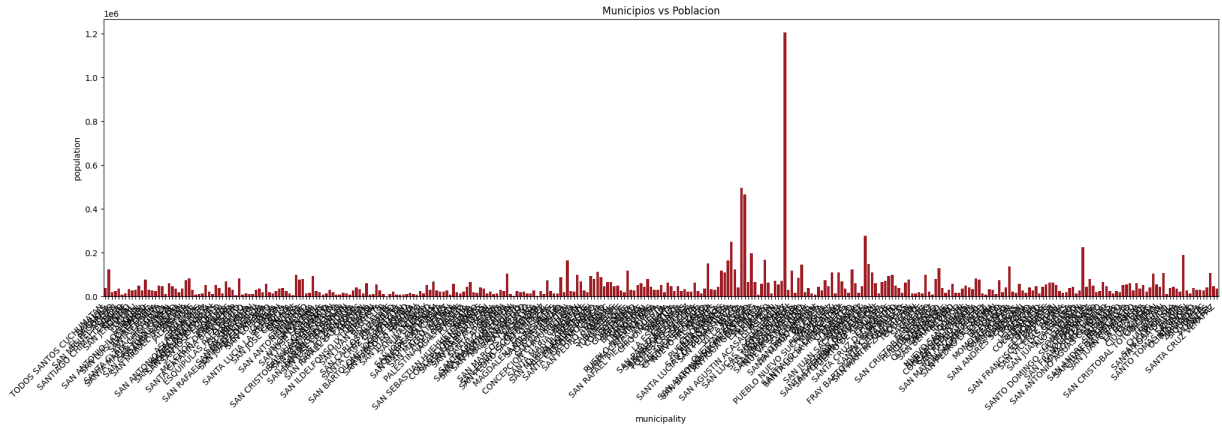


Observaciones: es posible visualizar los famosos "picos" de muertes entre los meses de julio y Septiembre, justo cuando las medidas sanitarias empezaron a flexibilizarse.

Grafica de barras municipios vs poblacion

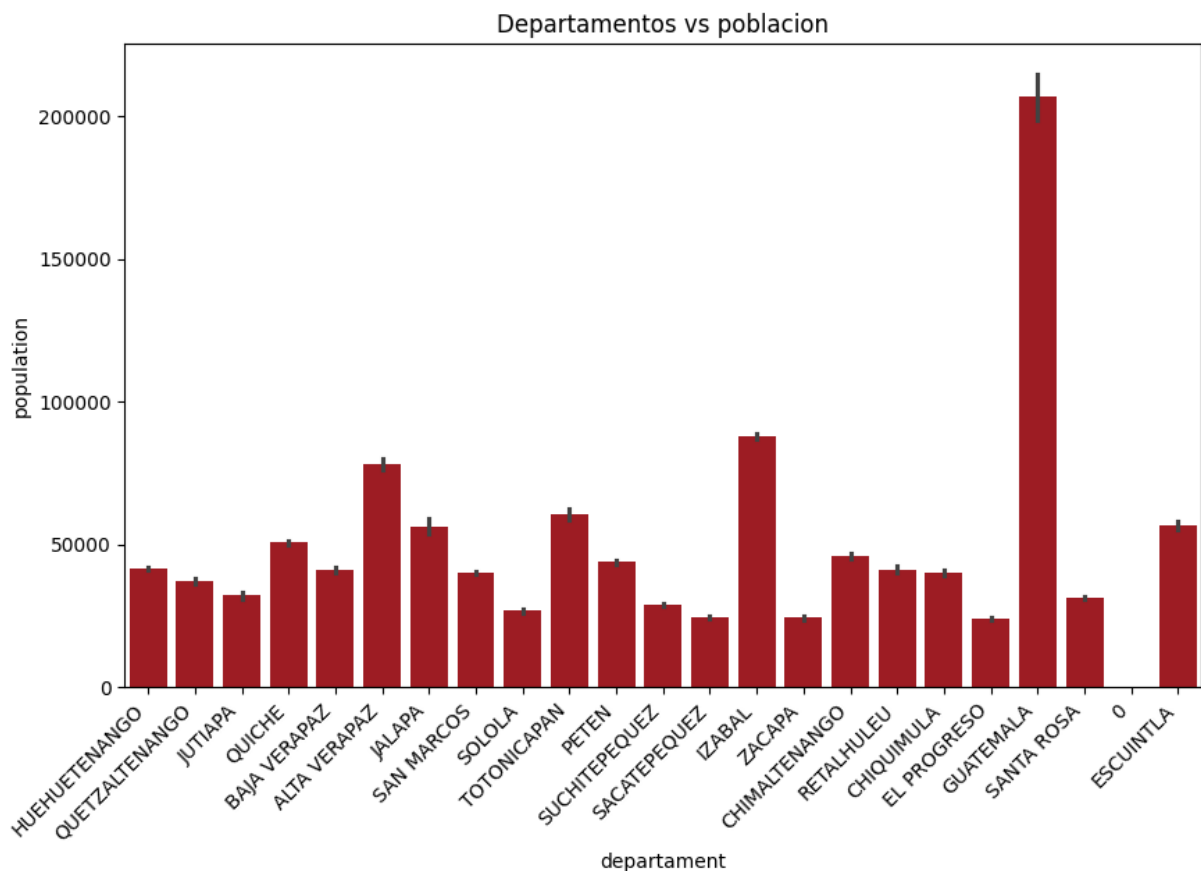
```
In [55]: plt.figure(figsize=(24, 6))
ax = sns.barplot(x='municipality', y='population', data=df_municipality, palette='magma')
ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha="right")
```

```
ax.set_title('Municipios vs Poblacion')
plt.show()
```



Grafica de barras departamentos vs poblacion

```
In [56]: plt.figure(figsize=(10, 6))
ax = sns.barplot(x='departament', y='population', data=df_municipality, palette='magma')
ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha="right")
ax.set_title('Departamentos vs poblacion')
plt.show()
```



Observación: nuevamente es posible visualizar a Guatemala como el departamento con mayor densidad poblacional.

Conclusiones Generales

En base a los resultados obtenidos en los analisis de datos, se puede concluir en que:

- El departamento de Guatemala es el que mayor cantidad de muertes acumula, esto debido a que es el departamento con más población en todo el país.
- Además de ser el departamento con mayor población, Guatemala tambien es el que tiene concentrado el trafico aereo del aeropuerto, por lo que gran parte de los contagios se disparan debido a esta variable que no puede ser medida de manera cuantitativa.
- Las muertes acumuladas estan relacionadas directamente con las nuevas muertes, por lo que al observar un descenso en la cantidad de muertes nuevas, podemos observar como las muertes acumuladas siguen una tendencia horizontal, indicando la normalización de las muertes.
- Las Nuevas muertes van aumentando con el paso del tiempo, por lo que se debe tomar en cuenta que las medidas sanitarias de contingencia deben ser mas estrictas al inicio de una pandemia para tratar de mantener las muertes al minimo posible.