

La base de datos fue extraída a través de una PQRS a la Cámara de Comercio del Cauca.

Repositorio del data set:

https://docs.google.com/spreadsheets/d/1t4qwqMq3Q0amMjjuPfGd_8lyYI_H5FWc/edit?usp=sharing&ouid=106123015324002929106&rtpof=true&sd=true

El proposito general de este proyecto es analizar los datos proporcionados por la Cámara de Comercio del Cauca a través de una PQRS respecto al turismo en el departamento del Cauca y sus municipios, a continuación se describe que se encontrara en cada columna para dar un poco mas de contexto.

Año: Se refiere al Año en que se realizo la encuesta.

mes: Se refiere al mes en que se realizo la encuesta

Pais de residencia: Nombre del pais de donde la persona reside

Ciudad: Es el nombre de la ciudad de Popayán o municipios del departamento del Cauca que que la persona visito.

Departamento: Se encontrara solamente el dato de Cauca.

Motivo del viaje: Se encontraran los datos de.

1_Vacaciones, recreo y ocio

2_Visitas a familiares y amigos

3_Educación y formación

4_Salud y atención médica

5_Religión y peregrinaciones 7_Tránsito 8_Otros motivos 9_Negocios y motivos profesionales

Cantidad de extranjeros no residentes: La cantidad de extrajeros no residentes que visitaron


Estudiante: Steven Alejandro Holguin Pabón

Talento tech 21/05/2024

✓ TURISMO INTELIGENTE

```
#Se importan las librerias que se utilizaran para realizar el analisis de los datos
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import folium
import geopandas as gpd
from folium.plugins import MarkerCluster
```


```
#Importamos el modulo de drive para extraer el dataset desde la drive
from google.colab import drive
drive.mount('/content/drive')
```

 Mounted at /content/drive

```
#Creamos una variable para asignarle la ruta que apunta en la Drive al dataset
ruta_archivo = '/content/drive/MyDrive/BD_turismo/4051978DocumentoSalida.xlsx'
```


```
#Se crea una variable
#Se lee el dataset utilizando
df_turismo = pd.read_excel(ruta_archivo)
```

```
#Se monta y desde monta el dataset por temas posibles conflictos al momento de abrir el data
!fusermount -u drive
!google-drive-ocamlfuse drive
```


 /bin/bash: line 1: google-drive-ocamlfuse: command not found

ANALISIS EXPLORATORIO

```
#A través de un impresion en consola se muestra cuantas filas y columnas tiene el dataset
print(df_turismo.shape)
```

 (9857, 7)

```
df_turismo.info()
```

 <class 'pandas.core.frame.DataFrame'>
 RangeIndex: 9857 entries, 0 to 9856
 Data columns (total 7 columns):

#	Column	Non-Null Count	Dtype
0	Año	9857 non-null	int64
1	Mes	9857 non-null	object
2	PaisOEEResidencia	9857 non-null	object

```
3  Ciudad          9857 non-null  object
4  Departamento    9857 non-null  object
5  MotivoViaje      9857 non-null  object
6  Cant Extranjeros no Residentes  9857 non-null  int64
dtypes: int64(2), object(5)
memory usage: 539.2+ KB
```

```
#Se cambia el nombre de algunos campos de la base de datos
```

```
nuevos_nombres = {
    'Año': 'anio',
    'Mes': 'mes',
    'PaisOEEResidencia': 'pais_de_residencia',
    'Ciudad': 'ciudad',
    'Departamento': 'departamento',
    'MotivoViaje': 'motivo_viaje',
    'Cant Extranjeros no Residentes': 'cantidad_de_Extranjeros'
}
```

```
#Se manda a llamar el diccionario nuevos nombre y elimina la fila completa donde encuentre c
df_turismo.rename(columns=nuevos_nombres, inplace=True)
```

```
#Consultar funcion de python para mostrar todas las filas y columnas
df_turismo.head(len(df_turismo))
```



	anio	mes	pais_de_residencia	ciudad	departamento	motivo_viaje	cantidad
0	2015	Ene	Alemania	Popayán	Cauca	1_Vacaciones, recreo y ocio	
1	2015	Ene	Argentina	Popayán	Cauca	1_Vacaciones, recreo y ocio	
2	2015	Ene	Argentina	Timbío	Cauca	1_Vacaciones, recreo y ocio	
3	2015	Ene	Argentina	Popayán	Cauca	8_Otros motivos	
4	2015	Ene	Australia	Popayán	Cauca	1_Vacaciones, recreo y ocio	
5	2015	Ene	Austria	Popayán	Cauca	1_Vacaciones, recreo y ocio	
6	2015	Ene	Bélgica	Popayán	Cauca	1_Vacaciones, recreo y ocio	
7	2015	Ene	Bolivia	Popayán	Cauca	1_Vacaciones, recreo y ocio	
8	2015	Ene	Brasil	La Vega	Cauca	1_Vacaciones, recreo y ocio	
9	2015	Ene	Brasil	Popayán	Cauca	1_Vacaciones, recreo y ocio	
10	2015	Ene	Brasil	Popayán	Cauca	3_Educación y formación	
11	2015	Ene	Brasil	Popayán	Cauca	8_Otros motivos	
12	2015	Ene	Canadá	Popayán	Cauca	1_Vacaciones, recreo y ocio	
13	2015	Ene	Canadá	Popayán	Cauca	8_Otros motivos	
14	2015	Ene	Chile	Popayán	Cauca	1_Vacaciones, recreo y ocio	
15	2015	Ene	Corea del Sur	Popayán	Cauca	1_Vacaciones, recreo y ocio	
16	2015	Ene	Costa Rica	Miranda	Cauca	1_Vacaciones, recreo y ocio	
17	2015	Ene	Costa Rica	Popayán	Cauca	1_Vacaciones, recreo y ocio	
18	2015	Ene	Cuba	Popayán	Cauca	1_Vacaciones, recreo y ocio	
						8_Otros	

5799	2021	Mar	Argentina	Popayán	Cauca	1_Vacaciones, recreo y ocio
5800	2021	Mar	Argentina	Popayán	Cauca	8_Otros motivos
5801	2021	Mar	Bélgica	Popayán	Cauca	1_Vacaciones, recreo y ocio

```
df_turismo.info()
```

```
>>> <class 'pandas.core.frame.DataFrame'>
RangeIndex: 9857 entries, 0 to 9856
Data columns (total 7 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   anio                                9857 non-null   int64
1   mes                                9857 non-null   object
2   pais_de_residencia                 9857 non-null   object
3   ciudad                             9857 non-null   object
4   departamento                       9857 non-null   object
5   motivo_viaje                       9857 non-null   object
6   cantidad_de_Extranjeros            9857 non-null   int64
dtypes: int64(2), object(5)
memory usage: 539.2+ KB
```

```
#Mostrar estadísticas de las columnas no numericas
df_turismo.describe()
```

```
>>>
```

	anio	cantidad_de_Extranjeros
count	9857.000000	9857.000000
mean	2019.331845	3.772750
std	2.607579	10.346664
min	2015.000000	1.000000
25%	2017.000000	1.000000
50%	2019.000000	1.000000
75%	2022.000000	3.000000
max	2023.000000	274.000000

```
#Se revisan los sub-niveles que tiene cada columna
cols_cat = ['anio', 'mes', 'pais_de_residencia', 'ciudad', 'motivo_viaje', 'cantidad_de_Extra
for col in cols_cat:
    print(f'columna {col}: {df_turismo[col].nunique()} subniveles')
```

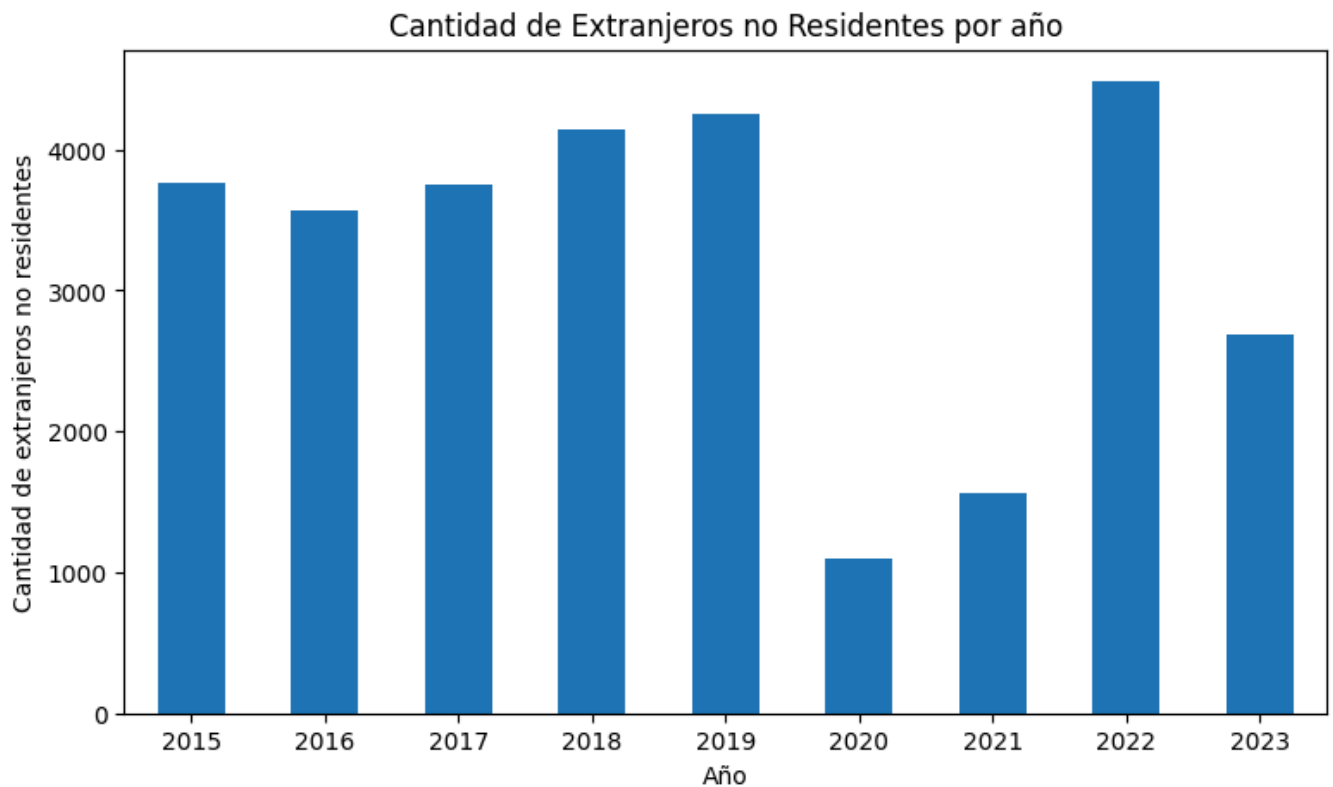
```
>>> columna anio: 9 subniveles
columna mes: 12 subniveles
```

```
columna pais_de_residencia: 118 subniveles  
columna ciudad: 44 subniveles  
columna motivo_viaje: 9 subniveles  
columna cantidad_de_Extranjeros: 105 subniveles
```

```
df_turismo.columns
```

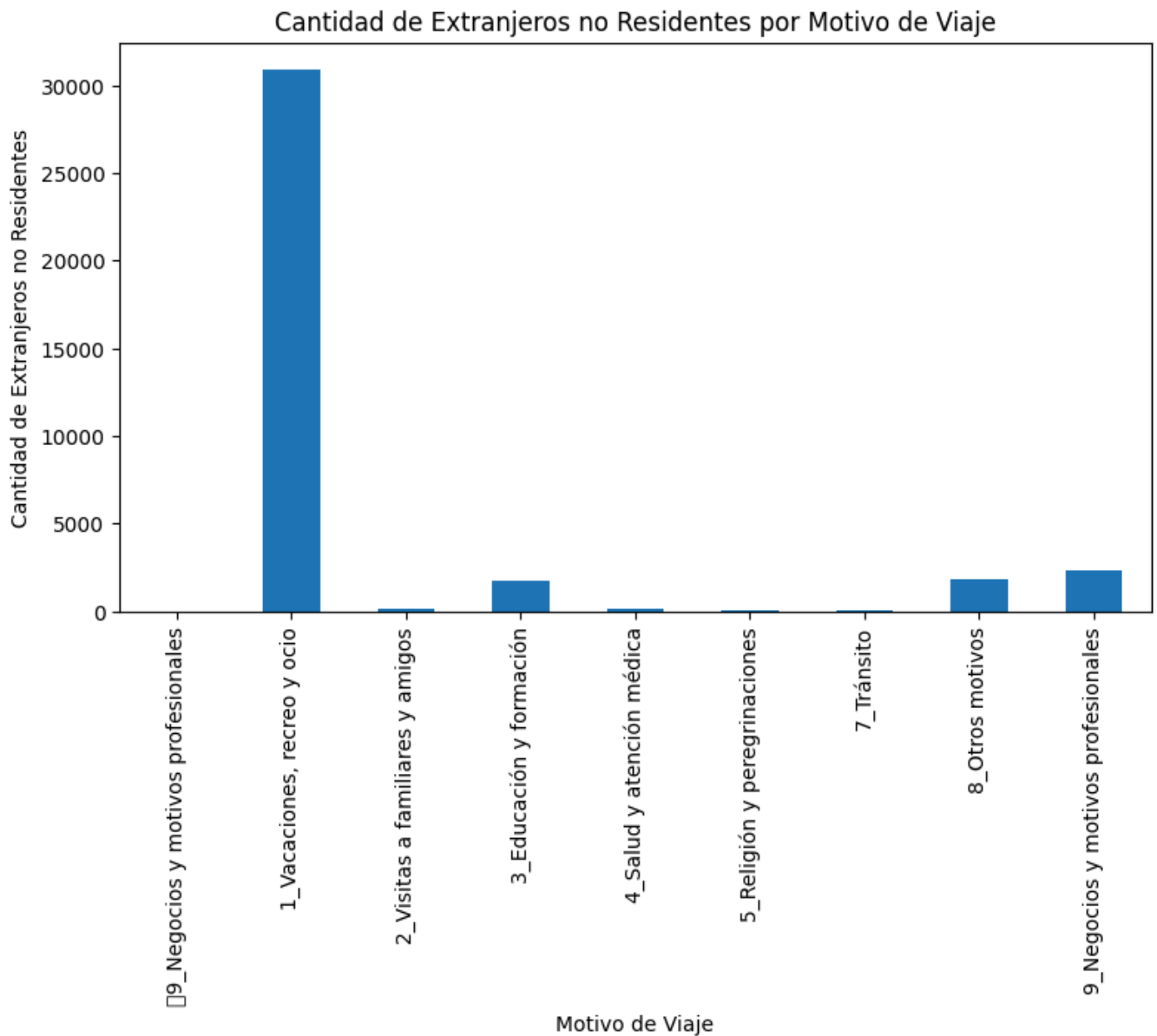
```
Index(['anio', 'mes', 'pais_de_residencia', 'ciudad', 'departamento',  
      'motivo_viaje', 'cantidad_de_Extranjeros'],  
      dtype='object')
```

```
plt.figure(figsize=(9, 5))  
df_turismo_filtro= df_turismo[(df_turismo["ciudad"] == "Popayán")]  
df_turismo_filtro.groupby("anio")["cantidad_de_Extranjeros"].sum().plot(kind="bar")  
plt.xlabel("Año")  
plt.ylabel("Cantidad de extranjeros no residentes")  
plt.title("Cantidad de Extranjeros no Residentes por año")  
plt.xticks(rotation=0)  
plt.show()
```



```
# Graficar la columna "MotivoViaje" como un gráfico de barras
plt.figure(figsize=(9, 5))
df_turismo.groupby("motivo_viaje")["cantidad_de_Extranjeros"].sum().plot(kind="bar")
plt.xlabel("Motivo de Viaje")
plt.ylabel("Cantidad de Extranjeros no Residentes")
plt.title("Cantidad de Extranjeros no Residentes por Motivo de Viaje")
plt.xticks(rotation=90)
plt.show()
```

➔ /usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Gly
fig.canvas.print_figure(bytes_io, **kw)

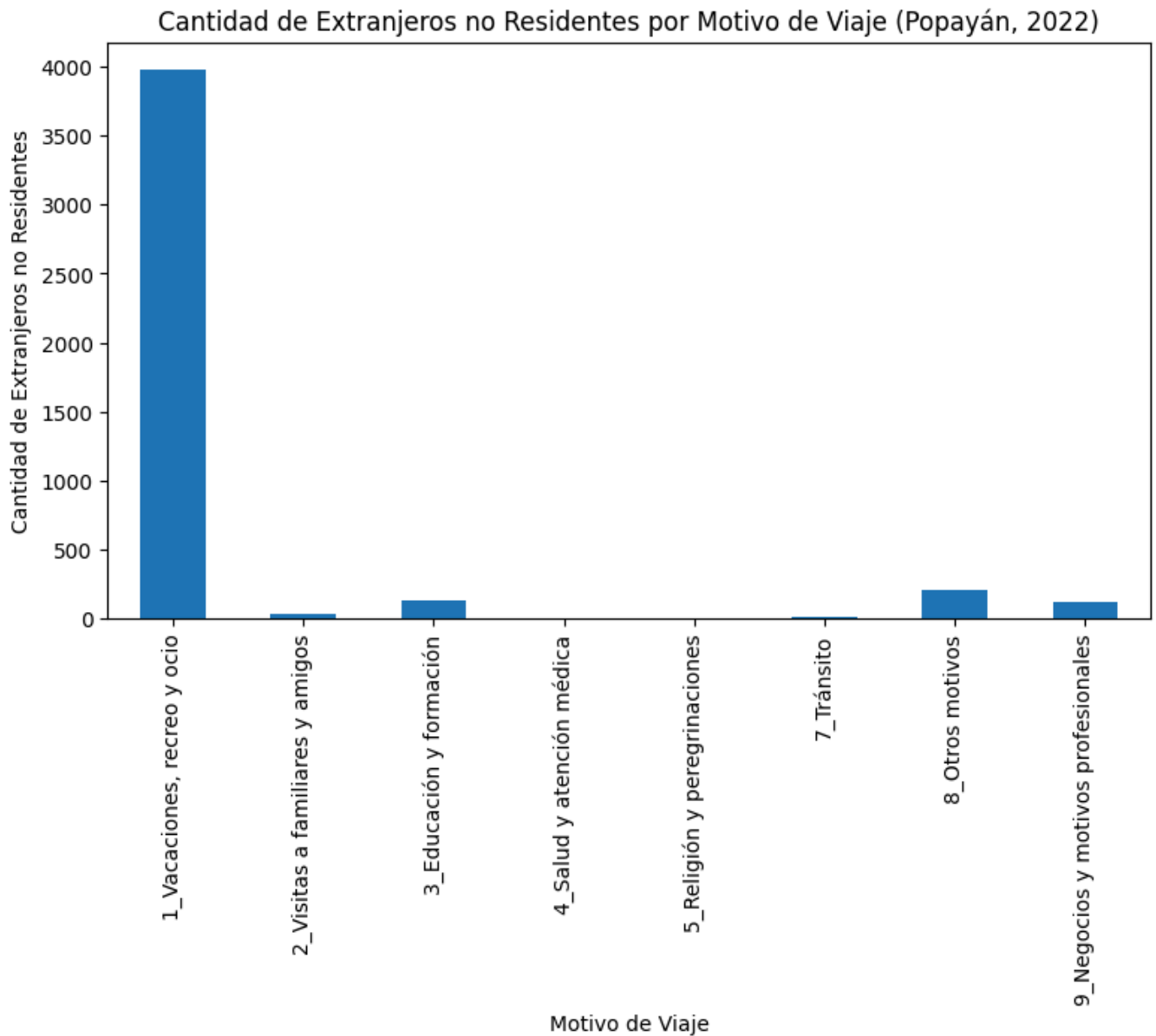


LIMPIEZA DE LA INFORMACIÓN

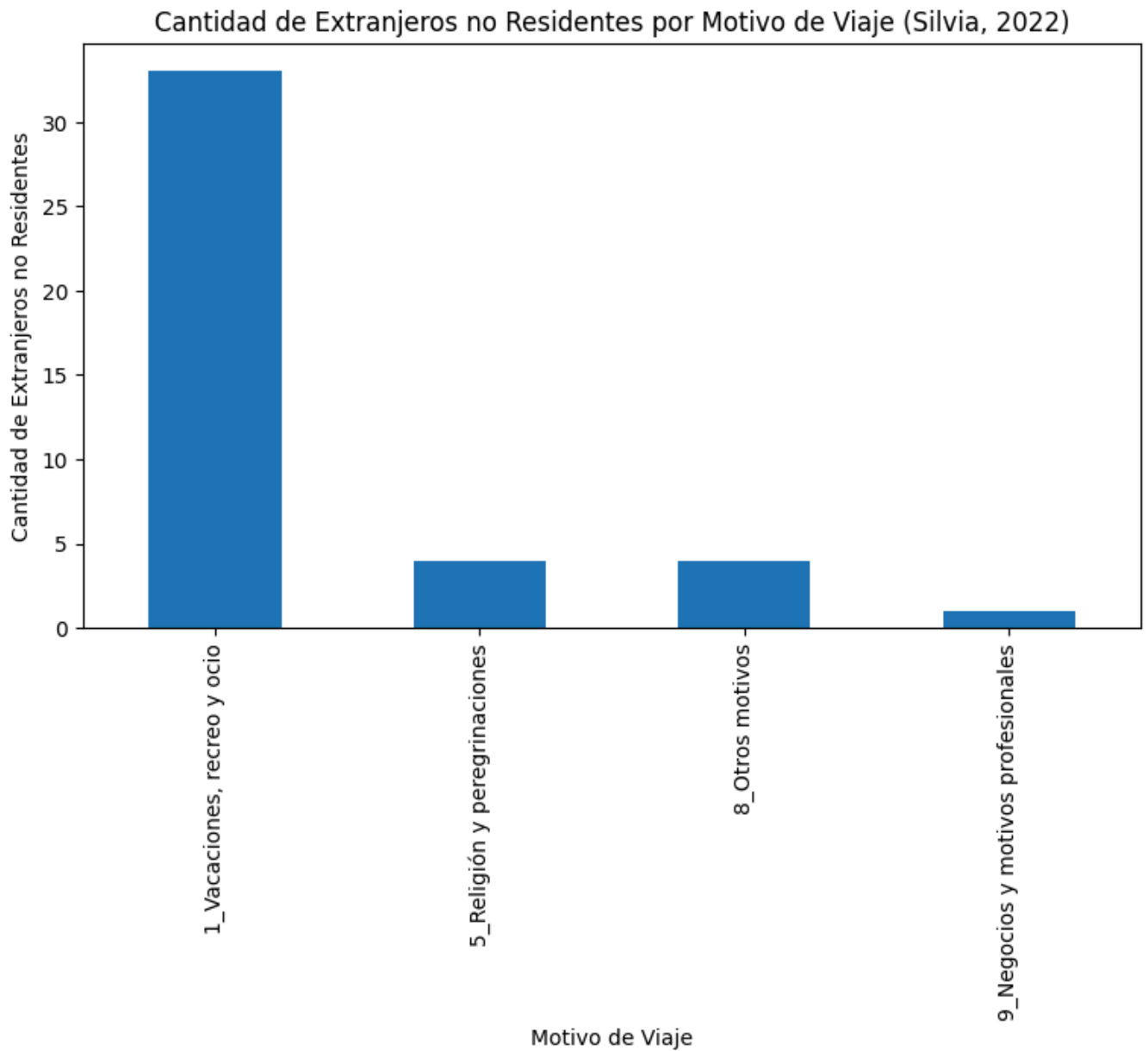
```
#Se eliminan los espacios en blanco de la columna motivo_viaje
df_turismo['motivo_viaje'] = df_turismo['motivo_viaje'].str.strip()

# Eliminar la columna departamento ya que el valor es siempre el mismo y es Cauca
df_turismo = df_turismo.drop('departamento', axis=1)

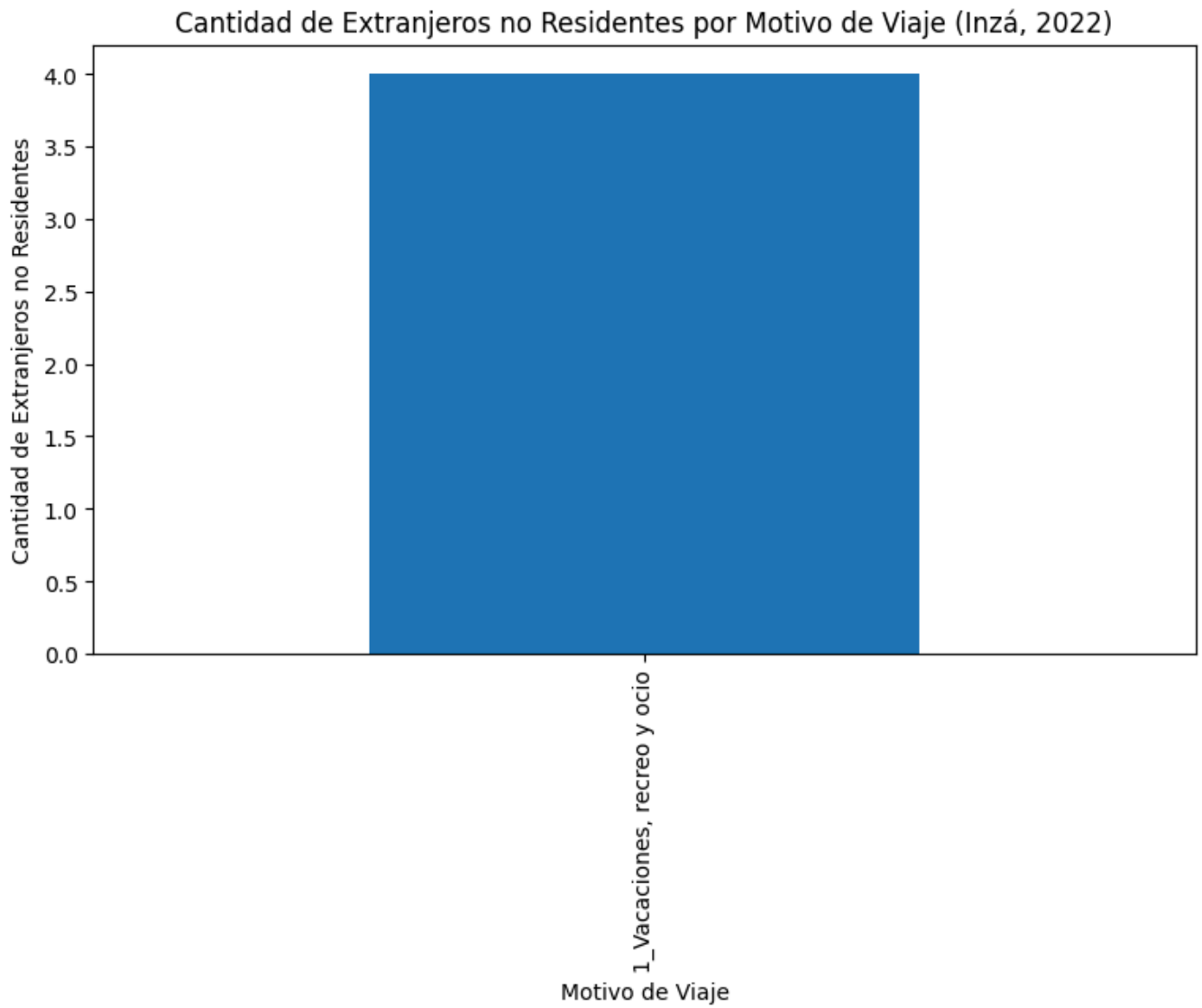
df_popayan_2022 = df_turismo[(df_turismo["ciudad"] == "Popayán") & (df_turismo["anio"] == 2022)]
plt.figure(figsize=(9, 5))
df_popayan_2022.groupby("motivo_viaje")["cantidad_de_Extranjeros"].sum().plot(kind="bar")
plt.xlabel("Motivo de Viaje")
plt.ylabel("Cantidad de Extranjeros no Residentes")
plt.title("Cantidad de Extranjeros no Residentes por Motivo de Viaje (Popayán, 2022)")
plt.xticks(rotation=90)
plt.show()
```

```
df_silvia_2022 = df_turismo[(df_turismo["ciudad"] == "Silvia") & (df_turismo["anio"] == 2022)]
plt.figure(figsize=(9,5))
df_silvia_2022.groupby("motivo_viaje")["cantidad_de_Extranjeros"].sum().plot(kind="bar")
plt.xlabel("Motivo de Viaje")
plt.ylabel("Cantidad de Extranjeros no Residentes")
plt.title("Cantidad de Extranjeros no Residentes por Motivo de Viaje (Silvia, 2022)")
plt.xticks(rotation=90)
plt.show()
```



```
df_inza_2022 = df_turismo[(df_turismo["ciudad"] == "Inzá") & (df_turismo["anio"] == 2022)]
plt.figure(figsize=(9, 5))
df_inza_2022.groupby("motivo_viaje")["cantidad_de_Extranjeros"].sum().plot(kind="bar")
plt.xlabel("Motivo de Viaje")
plt.ylabel("Cantidad de Extranjeros no Residentes")
plt.title("Cantidad de Extranjeros no Residentes por Motivo de Viaje (Inzá, 2022)")
plt.xticks(rotation=90)
plt.show()
```



VISUALIZACIÓN DE LOS DATOS

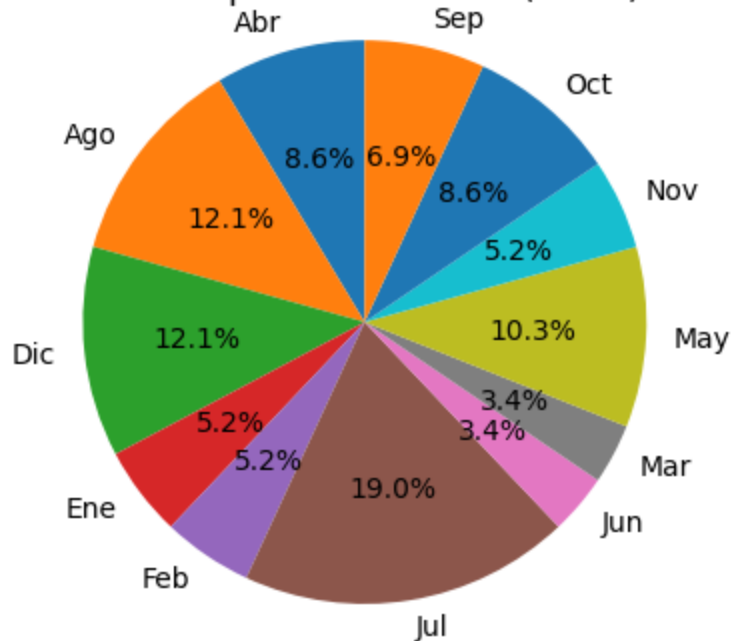
```
# Filtra los datos para los países de interés (por ejemplo, Brasil y Argentina)
países_interes = ['Brasil', 'Argentina', 'Francia', 'España', 'Estados Unidos']
df_filtrado = df_turismo[df_turismo['país_de_residencia'].isin(países_interes) & (df_turismo['mes'] == '2022')]

# Agrupa los datos por país y mes y calcula la suma de visitantes
df_agrupado = df_filtrado.groupby(['país_de_residencia', 'mes'])['cantidad_de_Extranjeros'].sum()

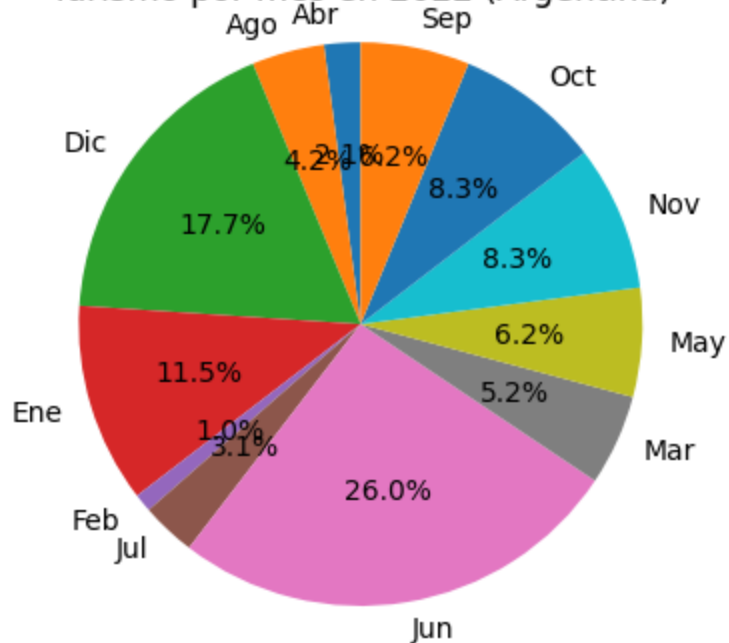
# Crea un gráfico de torta para cada país
for país in países_interes:
    datos_país = df_agrupado.loc[país]
    plt.figure(figsize=(4, 4))
    plt.pie(datos_país, labels=datos_país.index, autopct='%1.1f%%', startangle=90)
    plt.title(f"Turismo por mes en 2022 ({país})")
    plt.axis('equal')
    plt.show()
```



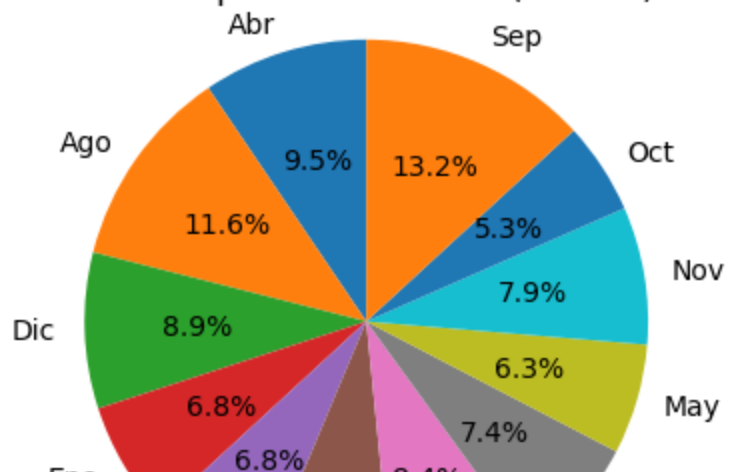
Turismo por mes en 2022 (Brasil)

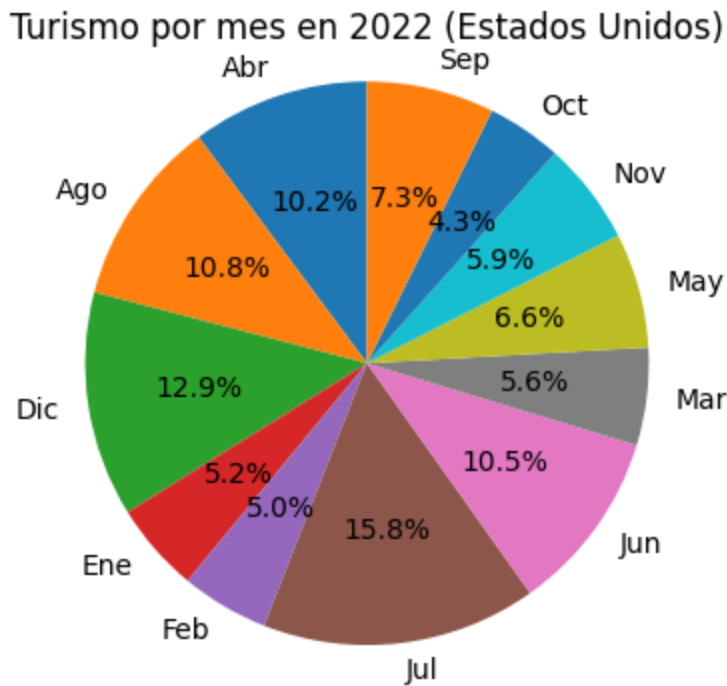
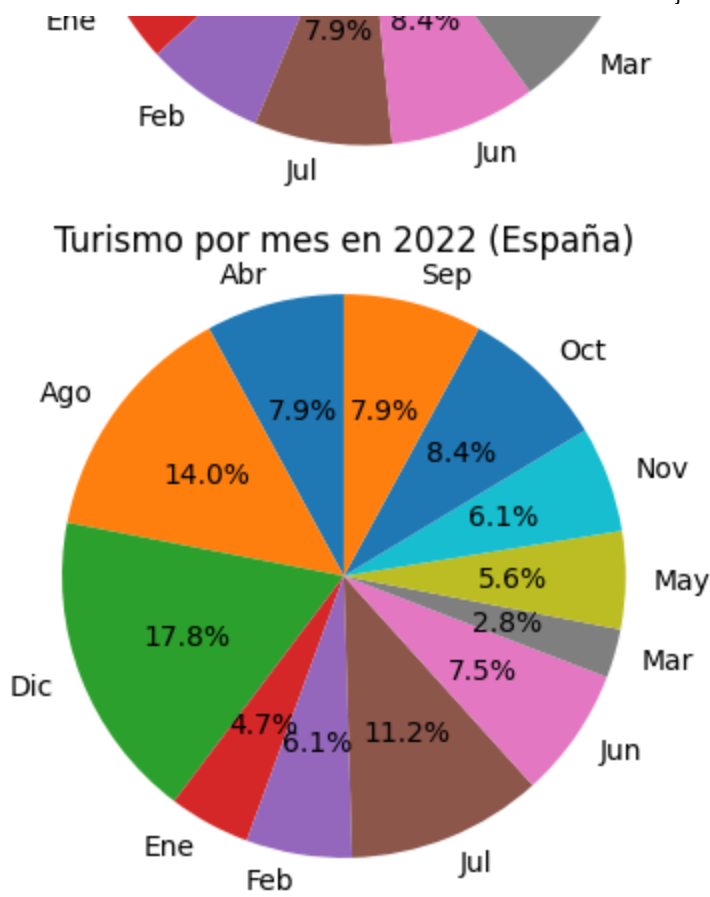


Turismo por mes en 2022 (Argentina)



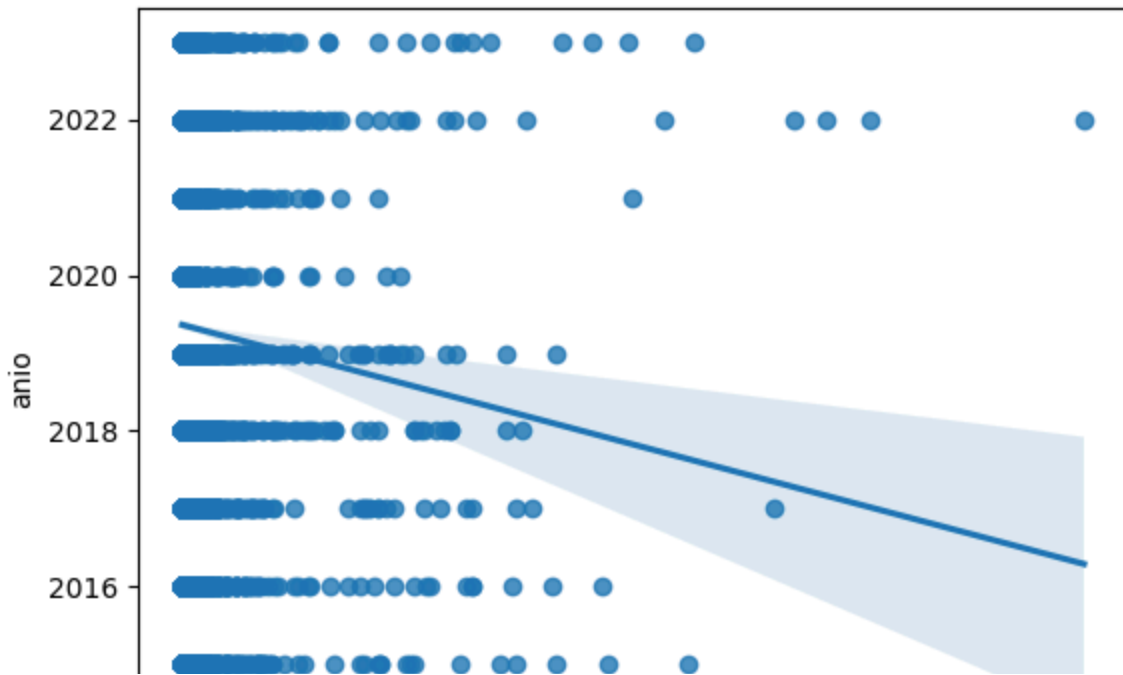
Turismo por mes en 2022 (Francia)





```
sns.regplot(x="cantidad_de_Extranjeros", y="anio", data=df_turismo)
```

↩ <Axes: xlabel='cantidad_de_Extranjeros', ylabel='anio'>



```
df_2022 = df_turismo[df_turismo["mes"] == "Mar"]
sns.regplot(x="cantidad_de_Extranjeros", y="anio", data=df_2022)
plt.xlabel("Cantidad de Extranjeros no Residentes")
plt.ylabel("Año")
plt.title("Regresión Lineal: Cantidad de Extranjeros por Año (2022)")
plt.show()
```

↩

