

Avance II. Producto Integrador de Aprendizaje

Equipo 02 - Grupo 002

Frecuencia: Miércoles 7pm-10pm

Integrantes:

1802920 Juarez Rodríguez Flor Karina

1815973 Mendoza Álvarez Pilar Abigail

1802473 Ordaz Ruiz Margarita

“Suicide Rates Overview 1985 to 2016”

Descripción de los datos

El conjunto de datos que muestra la base de datos fue extraído de otros cuatro conjuntos de datos vinculados por tiempo y lugar, y se creó con la finalidad de encontrar señales correlacionadas con el aumento de las tasas de suicidio entre diferentes cohortes a nivel mundial, en todo el espectro socioeconómico.

Los datos se muestran en formato de tabla, la cual cuenta con 27,820 filas y 12 columnas con las siguientes descripciones: país (donde ocurrieron los suicidios), año (año en que ocurrieron los suicidios), sexo (contiene los valores “male” y female”), grupo de edad (rango de edad de 10 años de las personas que se suicidaron), recuento de suicidios (número de suicidios ocurridos), población (número de habitantes en ese año), tasa de suicidio (número de suicidios ocurridos entre la población dividida entre mil), composite key de país-año (muestra el país seguido por el año), índice de desarrollo humano (HDI) por año (porcentaje), PIB por año, PIB per cápita (cantidades en dólares) y generación (promedio de agrupación de edad, nombre de la generación de las personas que se suicidaron).

Lo primero que necesitamos para comenzar a trabajar con la base de datos es importar las librerías que nos ayudarán en el proceso de limpieza de datos y leer nuestra base de datos, nuestro tipo de archivo es un csv *(Coma Separated Value)*, para esto ocupamos el siguiente código:

```
In [1]: import pandas as pd
import numpy as np

datos = pd.read_csv("SUICIDIOS.csv")
```

La función *shape* nos devuelve (filas,columnas) con las que cuenta nuestra base de datos.

```
In [2]: datos.shape
```

```
Out[2]: (27820, 12)
```

Los nombres de las columnas son:

```
In [3]: datos.columns
```

```
Out[3]: Index(['country', 'year', 'sex', 'age', 'suicides_no', 'population',
              'suicides/100k pop', 'country-year', 'HDI for year',
              ' gdp_for_year ($) ', 'gdp_per_capita ($)', 'generation'],
              dtype='object')
```

Los rangos de edades que tenemos son:

```
In [4]: print(datos.age.unique())
print("Los rangos de edades que tenemos son: ",len(datos.age.unique()))
```

```
['15-24 years' '35-54 years' '75+ years' '25-34 years' '55-74 years'
 '5-14 years']
```

Los rangos de edades que tenemos son: 6

Los países que tenemos son:

```
In [5]: print(datos.country.unique())
print("El número de países que tenemos es: ",len(datos.country.unique()))
```

```
['Albania' 'Antigua and Barbuda' 'Argentina' 'Armenia' 'Aruba' 'Australia'
'Austria' 'Azerbaijan' 'Bahamas' 'Bahrain' 'Barbados' 'Belarus' 'Belgium'
'Belize' 'Bosnia and Herzegovina' 'Brazil' 'Bulgaria' 'Cabo Verde'
'Canada' 'Chile' 'Colombia' 'Costa Rica' 'Croatia' 'Cuba' 'Cyprus'
'Czech Republic' 'Denmark' 'Dominica' 'Ecuador' 'El Salvador' 'Estonia'
'Fiji' 'Finland' 'France' 'Georgia' 'Germany' 'Greece' 'Grenada'
'Guatemala' 'Guyana' 'Hungary' 'Iceland' 'Ireland' 'Israel' 'Italy'
'Jamaica' 'Japan' 'Kazakhstan' 'Kiribati' 'Kuwait' 'Kyrgyzstan' 'Latvia'
'Lithuania' 'Luxembourg' 'Macau' 'Maldives' 'Malta' 'Mauritius' 'Mexico'
'Mongolia' 'Montenegro' 'Netherlands' 'New Zealand' 'Nicaragua' 'Norway'
'Oman' 'Panama' 'Paraguay' 'Philippines' 'Poland' 'Portugal'
'Puerto Rico' 'Qatar' 'Republic of Korea' 'Romania' 'Russian Federation'
'Saint Kitts and Nevis' 'Saint Lucia' 'Saint Vincent and Grenadines'
'San Marino' 'Serbia' 'Seychelles' 'Singapore' 'Slovakia' 'Slovenia'
'South Africa' 'Spain' 'Sri Lanka' 'Suriname' 'Sweden' 'Switzerland'
'Thailand' 'Trinidad and Tobago' 'Turkey' 'Turkmenistan' 'Ukraine'
'United Arab Emirates' 'United Kingdom' 'United States' 'Uruguay'
'Uzbekistan']
```

El número de países que tenemos es: 101

Aqui se presentan los años que contiene nuestra base de datos.

```
In [6]: print(datos.year.unique())
print("El número total de años que abarca nuestra base es: ",len(datos.year.unique()))
```

```
[1987 1988 1989 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002
 2003 2004 2005 2006 2007 2008 2009 2010 1985 1986 1990 1991 2012 2013
 2014 2015 2011 2016]
```

El número total de años que abarca nuestra base es: 32

Abajo se presentan los nombres de las generaciones que tiene nuestra base.

```
In [7]: print(datos.generation.unique())
print("Tenemos en total ",len(datos.generation.unique())," generaciones.")
```

```
['Generation X' 'Silent' 'G.I. Generation' 'Boomers' 'Millenials'
'Generation Z']
```

Tenemos en total 6 generaciones.

```
In [8]: datos.dtypes
```

```
Out[8]: country          object
year              int64
sex              object
age              object
suicides_no       int64
population        int64
suicides/100k pop float64
country-year      object
HDI for year      float64
  gdp_for_year ($)  object
  gdp_per_capita ($) int64
generation        object
dtype: object
```

Justificación del uso de datos

La elección de esta base de datos se debió a que nos llamó la atención que tratara sobre el suicidio dado que es un tema muy importante y del cuál se debería de hablar más para saber de él y tratar de prevenirlo, además, dado que estamos atravesando una pandemia y muchas personas nos encontramos en confinamiento los problemas de salud mental se vieron incrementados y entre ellos se encuentra el suicidio. Luego de que abrimos y analizamos un poco más la base de datos vimos que aparte de ser un gran tema el que trata, tenía mucha información con la cuál trabajar y esto nos beneficiaría a la hora de elegir nuestro objetivo.

Esperamos que trabajar con este conjunto de datos nos ayude a entender más sobre el tema del suicidio y los diferentes factores que influyen en él para así poder crear un poco más de conciencia y ver cómo esto se puede implementar en los programas sociales que existen para la prevención de este.

Objetivos

Nuestro objetivo principal se basa en que, se observan variaciones en las tasas de suicidios, y queremos encontrar si existe relación alguna entre las variables que se presentan en la base de datos que se utilizará a lo largo del proyecto (presentadas como columnas), y determinar cuáles son aquellas características que podrían ser factores en común en los que cuales pueden enfocarse alternativas y/o programas sociales con la finalidad de decrementar en la medida de lo posible estas tasas.

Además de nuestro objetivo principal, también queremos enfocarnos en un objetivo secundario el cual sería investigar si los tiempos de crisis son un factor detonante a la alteración de las tasas de suicidio, incluyendo también la región/país en la que se encuentren.

Limpieza de Datos

Una vez explicada nuestra base de datos y lo que queremos lograr con ella iniciaremos con la

limpieza de datos y lo primero que queremos hacer es ver nuestros datos y con el siguiente código vamos a mostrar los primeros 5 renglones de nuestra base de datos sin manipulación.

In [9]: `datos.head()`

Out[9]:

	country	year	sex	age	suicides_no	population	suicides/100k pop	country- year	HDI for year	gdp_for
0	Albania	1987	male	15- 24 years	21	312900	6.71	Albania1987	NaN	2,156,62
1	Albania	1987	male	35- 54 years	16	308000	5.19	Albania1987	NaN	2,156,62
2	Albania	1987	female	15- 24 years	14	289700	4.83	Albania1987	NaN	2,156,62
3	Albania	1987	male	75+ years	1	21800	4.59	Albania1987	NaN	2,156,62
4	Albania	1987	male	25- 34 years	9	274300	3.28	Albania1987	NaN	2,156,62

El siguiente paso es eliminar las columnas que no necesitamos, para poder optimizar el código, en este caso nosotras decidimos eliminar la columna **population** esto debido a que ya tenemos el porcentaje de suicidios por población así que se nos hizo redundante, la columna **country-year** porque al igual que la columna **population** es redundante, debido a que ya tenemos tanto el país como el año, **HDI for year** debido a que no nos es útil para nuestros objetivos y la columna **gdp_for_year** esto porque preferimos basarnos en la columna **gdp_per_capita**.

Para esto utilizamos la función `drop()`, cabe mencionar que estos cambios decidimos almacenarnos en otra variable, para poder seguir teniendo nuestros datos originales.

In [10]: `ds = datos.drop(['country-year', 'population', 'gdp_for_year ($)', 'HDI for year'])`
`ds.head()`

Out[10]:

	country	year	sex	age	suicides_no	suicides/100k pop	gdp_per_capita (\$)	generation
0	Albania	1987	male	15-24 years	21	6.71	796	Generation X
1	Albania	1987	male	35-54 years	16	5.19	796	Silent
2	Albania	1987	female	15-24 years	14	4.83	796	Generation X
3	Albania	1987	male	75+ years	1	4.59	796	G.I. Generation
4	Albania	1987	male	25-34 years	9	3.28	796	Boomers

Ahora vamos a cambiarle el nombre a las columnas para que nos sea más fácil manipular los datos.

```
In [11]: ds.rename(columns={'suicides/100k pop':'porcentaje_suicidios',
                           'gdp_per_capita ($)':'per_capita'},
                  inplace=True)

ds.head()
```

```
Out[11]:
```

	country	year	sex	age	suicides_no	porcentaje_suicidios	per_capita	generation
0	Albania	1987	male	15-24 years	21	6.71	796	Generation X
1	Albania	1987	male	35-54 years	16	5.19	796	Silent
2	Albania	1987	female	15-24 years	14	4.83	796	Generation X
3	Albania	1987	male	75+ years	1	4.59	796	G.I. Generation
4	Albania	1987	male	25-34 years	9	3.28	796	Boomers

Otro detalle que notamos es que los datos del 2016 están incompletos, y eso afectaba la parte de la visualización, así que decidimos eliminar las filas que tuvieran el año 2016.

```
In [12]: ds = ds.drop(ds[ds['year']==2016].index)
```

Lo siguiente es ver si nuestra base contiene datos nulos, a simple vista no se ve ninguno, pero lo verificamos con la siguiente función:

```
In [13]: ds.isnull().any()
```

```
Out[13]: country          False
year          False
sex           False
age           False
suicides_no   False
porcentaje_suicidios False
per_capita    False
generation    False
dtype: bool
```

Ahora procederemos a guardar estos datos en un nuevo archivo csv.

```
In [14]: ds.to_csv('Datos_Suicidios_limpios.csv', index=False)
```

Como hemos visto tenemos muchos datos y esto nos dificulta el estudio estadístico de los mismos así que decidimos agrupar los datos por edades, esto porque nos interesa ver como es que varían los números de suicidios de acuerdo a las edades. Para esto primero vemos cuales son los valores únicos de edades.

```
In [15]: edad = ds.age.unique()
print(edad)

['15-24 years' '35-54 years' '75+ years' '25-34 years' '55-74 years'
 '5-14 years']
```

Después con un ciclo for le asignamos cada agrupación de datos por rango de año a los elementos de una lista.

```
In [16]: col = []
for i in range(0,6):
    col.append(ds[ds['age']==edad[i]])

col[0].head()
```

Out[16]:

	country	year	sex	age	suicides_no	porcentaje_suicidios	per_capita	generation
0	Albania	1987	male	15-24 years	21	6.71	796	Generation X
2	Albania	1987	female	15-24 years	14	4.83	796	Generation X
13	Albania	1988	male	15-24 years	17	5.33	769	Generation X
17	Albania	1988	female	15-24 years	8	2.71	769	Generation X
28	Albania	1989	male	15-24 years	12	3.71	833	Generation X

Estadística básica

- Para la estadística básica decidimos usar la función *describe()* dado que ésta nos arroja un resumen estadístico de una columna que en este caso fue la de *suicides_no*

```
In [17]: ds['suicides_no'].describe()
```

```
Out[17]: count    27660.000000
mean         243.413485
std          904.490185
min           0.000000
25%           3.000000
50%          25.000000
75%         132.000000
max         22338.000000
Name: suicides_no, dtype: float64
```

-Lo que nos pareció más relevante de este resumen estadístico fueron los valores mínimo y máximo ya que nos muestra que los datos tiene mucha variabilidad, lo que nos reafirma que es más conveniente trabajar por agrupaciones.

- Dado que los valores tienen mucho variabilidad, decidimos ver que tanto varían por rango de edad.

5 – 14

```
In [18]: col[5]['suicides_no'].describe()
```

```
Out[18]: count      4610.000000
mean         11.337093
std          30.907550
min           0.000000
25%           0.000000
50%           2.000000
75%           8.000000
max          404.000000
Name: suicides_no, dtype: float64
```

15 – 24

```
In [19]: col[0]['suicides_no'].describe()
```

```
Out[19]: count      4610.000000
mean        175.129501
std         523.256414
min           0.000000
25%           6.000000
50%          33.000000
75%          120.000000
max        6945.000000
Name: suicides_no, dtype: float64
```

25 – 34

```
In [20]: col[3]['suicides_no'].describe()
```

```
Out[20]: count      4610.000000
mean        243.349675
std         763.770222
min           0.000000
25%           8.000000
50%          42.000000
75%         154.750000
max        10310.000000
Name: suicides_no, dtype: float64
```

35 – 54


```
In [21]: col[1]['suicides_no'].describe()
```

```
Out[21]: count      4610.000000
mean        530.667028
std         1646.488573
min          0.000000
25%         12.000000
50%         87.000000
75%        343.000000
max       22338.000000
Name: suicides_no, dtype: float64
```

55 – 74

```
In [22]: col[4]['suicides_no'].describe()
```

```
Out[22]: count      4610.000000
mean        358.726247
std         1020.669146
min          0.000000
25%          5.000000
50%         56.000000
75%        244.000000
max       11431.000000
Name: suicides_no, dtype: float64
```

75+

```
In [23]: col[2]['suicides_no'].describe()
```

```
Out[23]: count      4610.000000
mean        141.271367
std         365.940282
min          0.000000
25%          1.000000
50%         19.000000
75%         94.000000
max       3171.000000
Name: suicides_no, dtype: float64
```

```
<div class=text-justify>
```

-De los resúmenes estadísticos obtenidos por rango de edad podemos concluir dos cosas, la primera es que efectivamente cada rango tiene la misma cantidad de elementos y la segunda es que podemos ver que su variabilidad es más aceptable y fácil de manejar si se compara con la obtenida en la columna de suicides_no.

\$\$\$\$

Esto nos ayuda a ver la gran utilidad que será trabajar en agrupaciones y dado que es un proceso repetitivo decidimos no obtener los resúmenes estadísticos de todas las agrupaciones a utilizar pero se mostrarán gráficamente más adelante.

- Además, decidimos aplicar la función *skew()* para obtener el valor de asimetría de la columna *suicides_no*

```
In [24]: ds['suicides_no'].skew()
```

```
Out[24]: 10.325864499495285
```

-Como el valor de asimetría es mayor que 1 y por lo tanto positivo, podemos decir que la columna tiene una distribución extremadamente sesgada a la izquierda y ésto se confirmará con las gráficas.

Visualización de datos

Importamos las librerías que vamos a utilizar para realizar las gráficas.

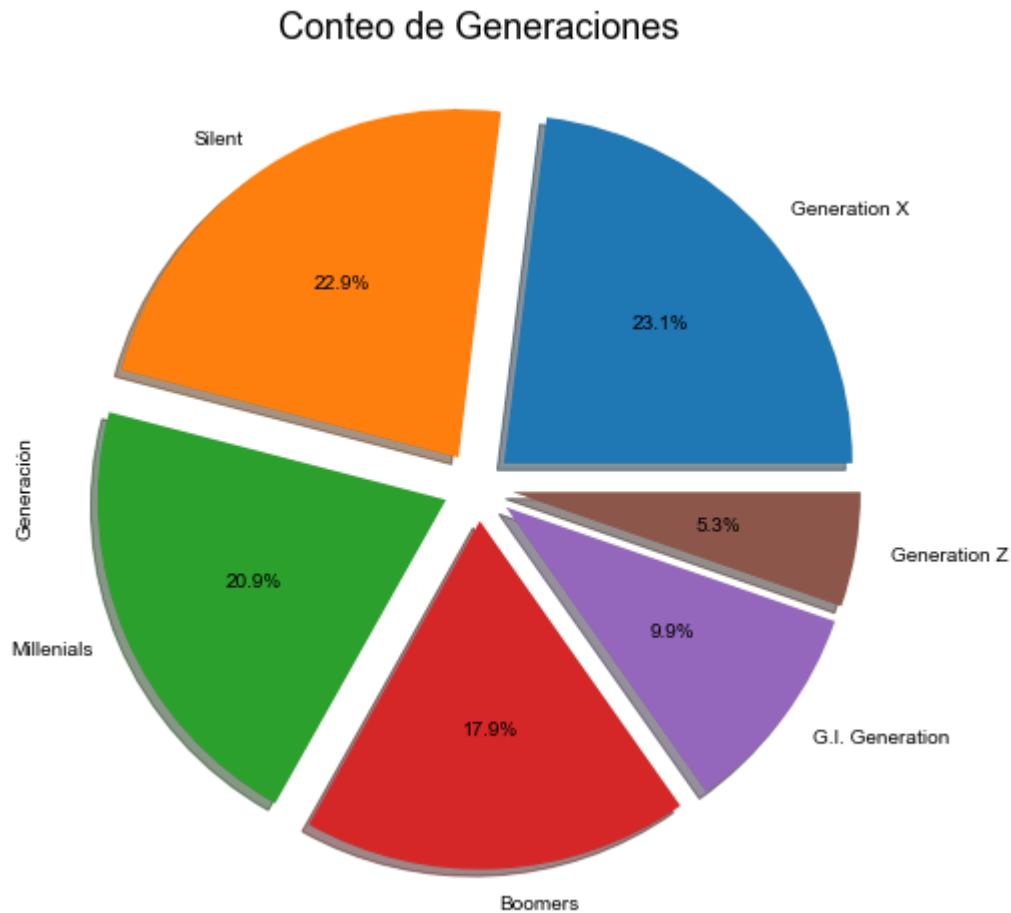
```
In [25]: import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px #https://plotly.com/python/plotly-express/
```

Como nuestro objetivo es determinar los factores que afecten más al número de suicidios decidimos realizar primero una gráfica de pastel en donde buscamos visualizar primeramente como se reparten nuestros datos en cuanto a las generaciones:

```
In [26]: f,ax=plt.subplots(1,1,figsize=(18,8))

ds['generation'].value_counts().plot.pie(explode=[0.1,0.1,0.1,0.1,0.1,0.1], autopct=
color = sns.color_palette("husl", 6)
sns.set(style="darkgrid")
plt.title('Conteo de Generaciones',size = 18)
plt.ylabel('Generación')
```

Out[26]: Text(0, 0.5, 'Generación')



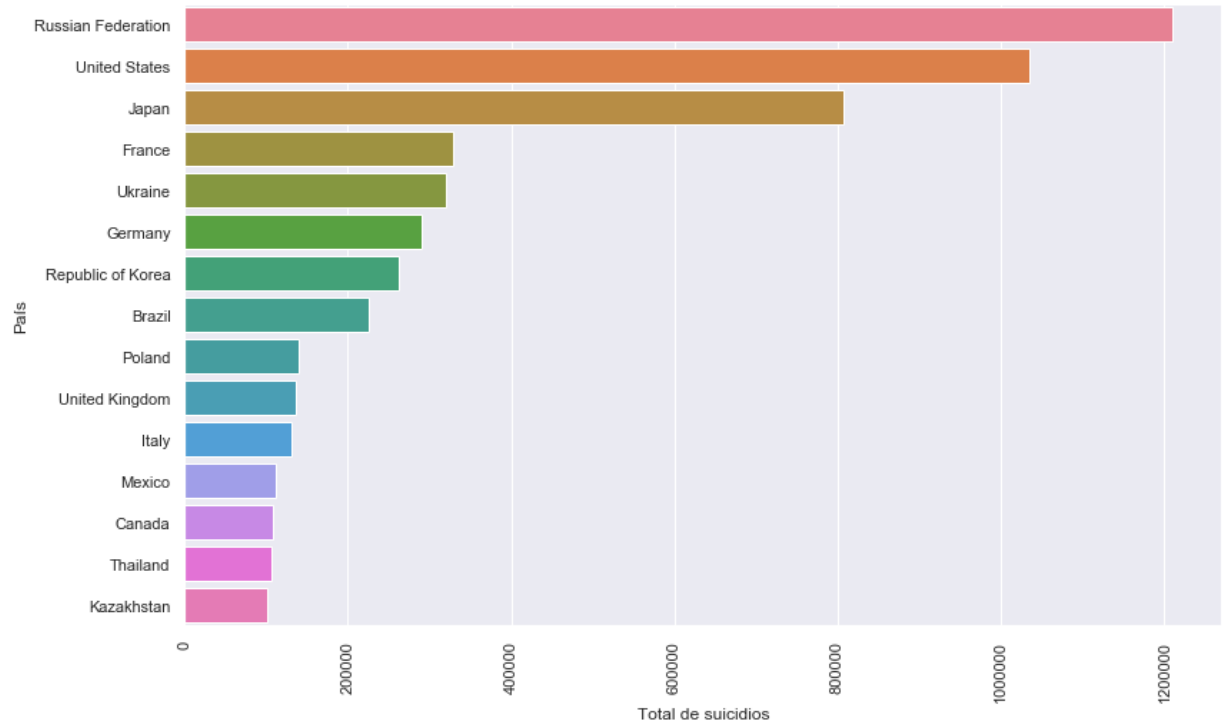
Ahora buscaremos los países con mayor número de suicidios, para evaluar el valor máximo, y para esto nos apoyaremos de una gráfica de barras y un mapa.

```
In [27]: Totalpais = ds.groupby(by=['country']).agg({'suicides_no': ['sum']})
Totalpais.columns = ['Tsuicidios']
Totalpais.reset_index(inplace=True)
Totalpais = Totalpais.sort_values(by=['Tsuicidios'], ascending=False).head(15)

color = sns.color_palette("husl", 15)
sns.set(style="darkgrid")
plt.figure(figsize=(13, 8))
sns.barplot(x='Tsuicidios', y='country', data=Totalpais, palette=color)
plt.xticks(rotation=90)

plt.figtext(.5,.91,'Top 15 paises total de suicidios (1985 - 2015)', color='b',
plt.xlabel('Total de suicidios')
plt.ylabel('País')
plt.show()
```

Top 15 paises total de suicidios (1985 - 2015)



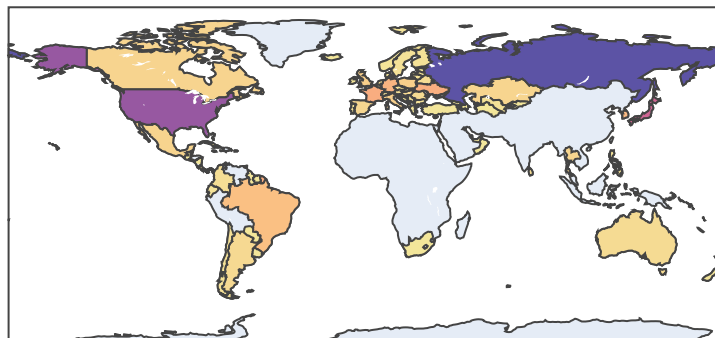
```
In [28]: #para este gráfico se INSTALA e IMPORTA la libreria plotly https://plotly.com/py
mapita = ds.groupby(by=['country']).agg({"suicides_no": ['sum']})
mapita.columns = ['total_suicidios']
mapita.reset_index(inplace=True)
mapita

fig = px.choropleth(mapita, locations="country", locationmode='country names',
                    color="total_suicidios",
                    hover_name="country",
                    color_continuous_scale='sunset')

fig.update_layout(
    title="Mapa de suicidios (1985 - 2015)",
    font=dict(
        family="Courier New, monospace",
        size=15,
        color="RebeccaPurple"
    )
)

fig.show()
```

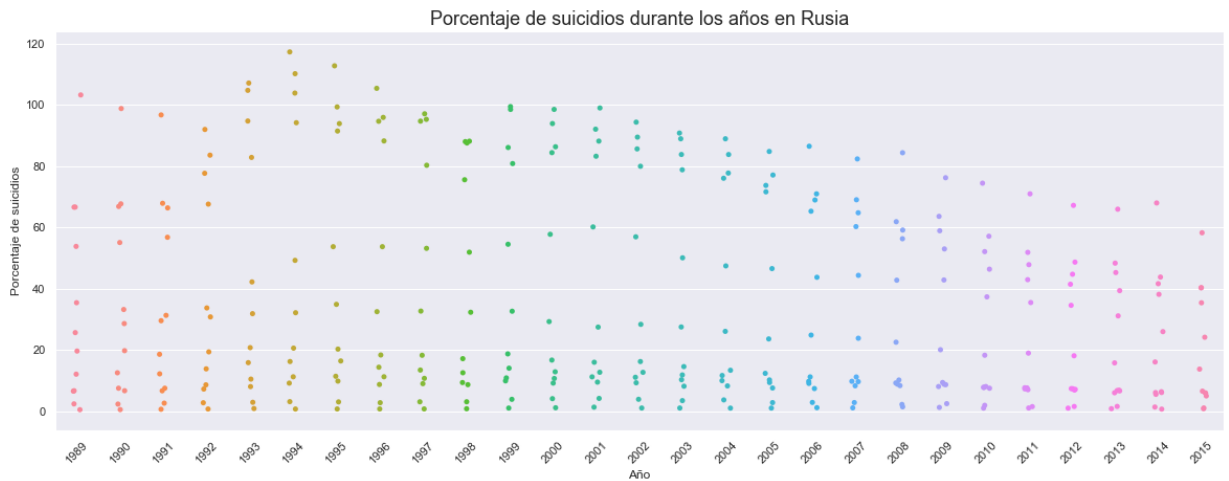
Mapa de suicidios (1985 - 2015)



Como vemos que Rusia es el país con más suicidios decidimos buscar cual es el año en el cual

hubo más suicidios en este país.

```
In [29]: rusia = ds[ds['country']=='Russian Federation']
plt.figure(figsize=(20,7))
sns.stripplot(x="year",y='porcentaje_suicidios',data=rusia)
plt.xlabel('Año')
plt.ylabel('Porcentaje de suicidios')
plt.title('Porcentaje de suicidios durante los años en Rusia',size = 18)
plt.xticks(rotation=45)
plt.show()
```

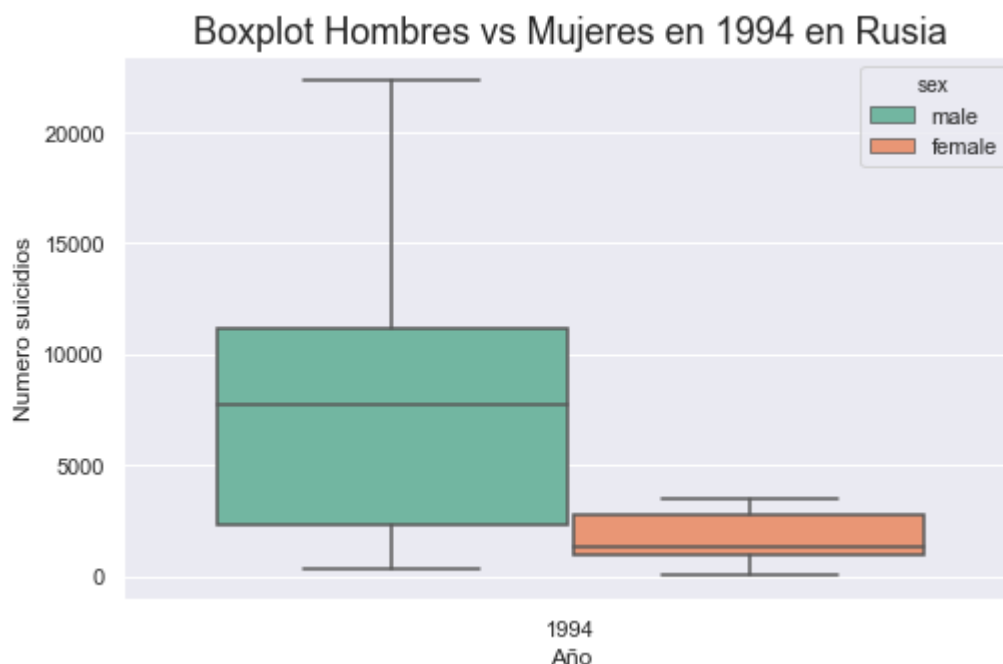


Como podemos observar el año que tuvo más porcentaje de suicidios fue 1994, así que decidimos enfocarnos en este año en Rusia para ver que tanto afecta a los géneros.

```
In [30]: f, ax = plt.subplots(1,1, figsize=(8,5))
mayor=ds[ds['year']==1994]
mayor.head()
ax = sns.boxplot(x='year', y='suicides_no',hue='sex', data=mayor[mayor['country']
sns.set(style="darkgrid")

plt.xlabel('Año',size=12)
plt.ylabel('Numero suicidios')
plt.title('Boxplot Hombres vs Mujeres en 1994 en Rusia',size = 18)
```

Out[30]: Text(0.5, 1.0, 'Boxplot Hombres vs Mujeres en 1994 en Rusia')

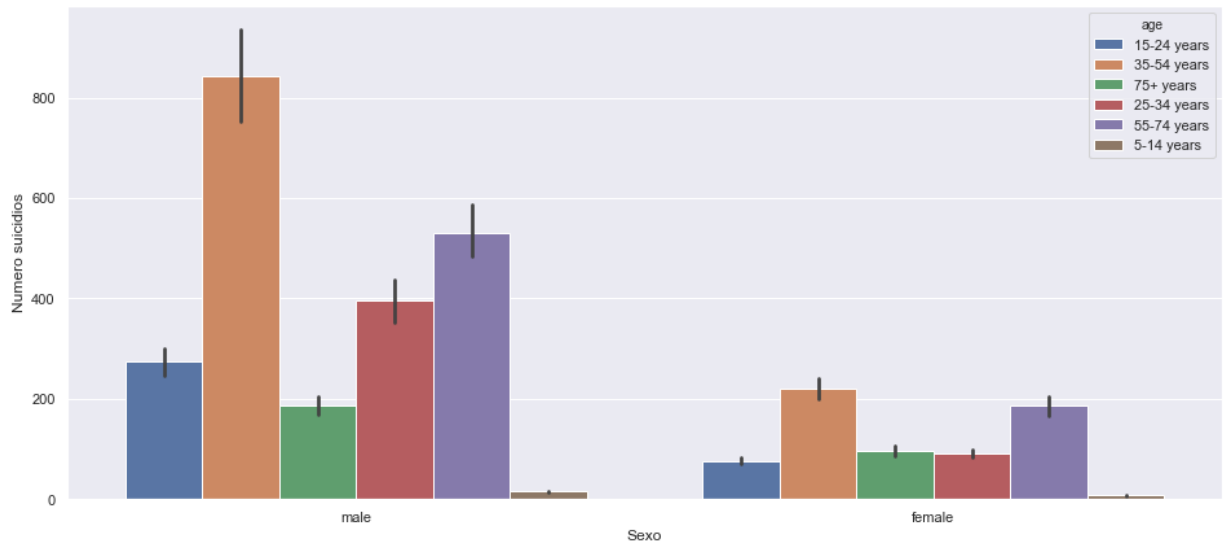


Esto que hicimos previamente nos ayudará para poder investigar desde un punto más específico, el contexto en el que se vivió en los años con un mayor número de suicidios en Rusia, como en el año 1994, para determinar si existieron algunas situaciones que detonaran este incremento. Esto con el fin, de poder tener una conclusión.

Y, ahora para intentar verlo desde otra perspectiva, haremos una serie de gráficos con un enfoque de dividir nuestros datos por rangos de edades y separados también por sexo ya que con el gráfico anterior podemos ver como si existe una diferencia considerable.

```
In [31]: plt.figure(figsize=(16,7))
gb_edad = sns.barplot(x = 'sex', y = 'suicides_no', hue = 'age',data = ds)
plt.xlabel('Sexo')
plt.ylabel('Numero suicidios')
```

```
Out[31]: Text(0, 0.5, 'Numero suicidios')
```



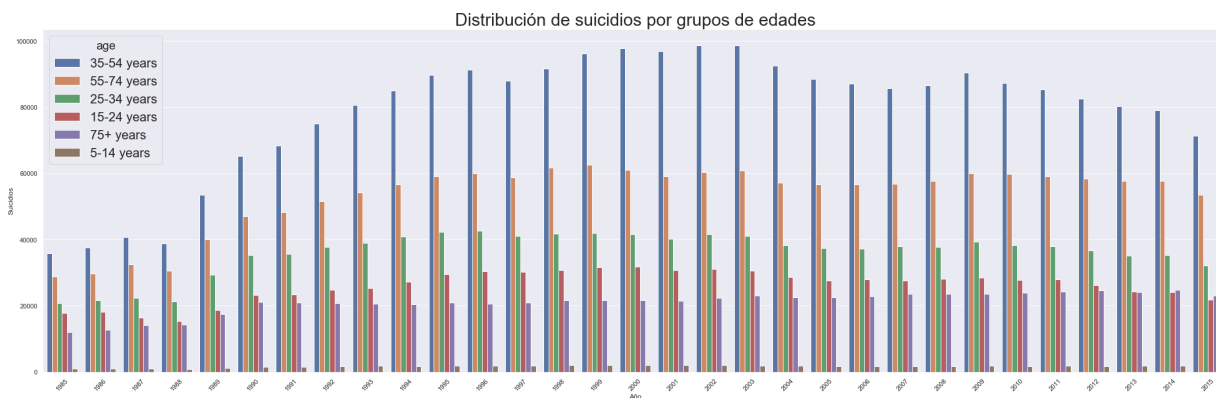
Como podemos ver en el gráfico anterior, según los datos que estamos analizando, los hombres son el género con un mayor número de suicidios y además el rango de edad donde hay más suicidios es entre los 35 y 54 años de edad seguido por los de 55-74 años.


```
In [32]: S_year = pd.DataFrame(ds['suicides_no'].groupby([ds['age'],ds['year']]).sum())
S_year = S_year.reset_index().sort_index(by='suicides_no',ascending=False)
yr = S_year
fig = plt.figure(figsize=(30,10))
plt.title('Distribución de suicidios por grupos de edades',size = 30)

sns.set(font_scale=2)
sns.barplot(y='suicides_no',x='year',hue='age',data=yr,palette='deep');
plt.ylabel('Suicidios')
plt.xlabel('Año')
plt.xticks(rotation=45)
plt.tight_layout()
```

C:\Users\maggy\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: FutureWarning:

by argument to sort_index is deprecated, please use .sort_values(by=...)



Esta gráfica nos ayuda mucho con nuestros objetivos, dado que se ve un aumento muy significativo los primeros años, en especial de las personas con edades de entre 35 y 54 años, así que esto nos ayuda para identificar una posible causa de que las personas de pronto empezaran a cometer más suicidios y ver que hechos históricos afectaron.

Segundo Objetivo

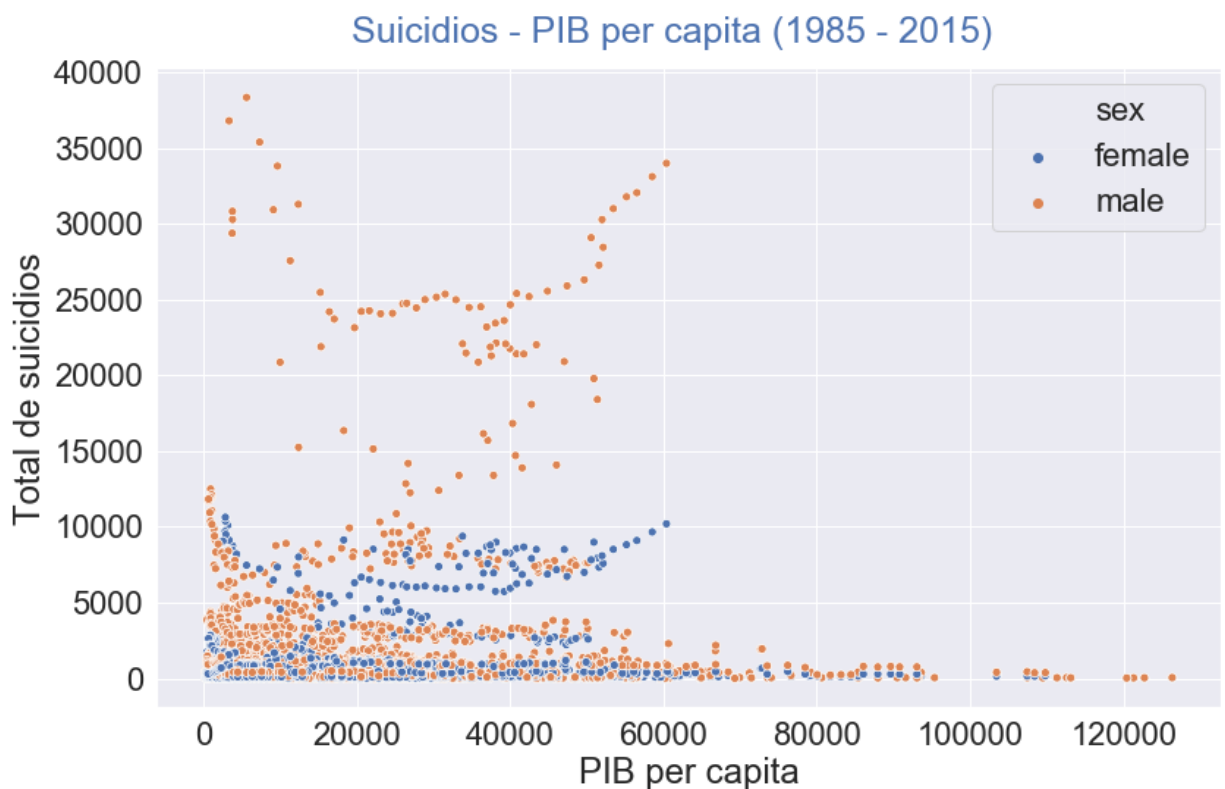
Por último, también consideramos importante (nuestro segundo objetivo) el buscar si afectaba los recursos económicos con los que se cuenta en cada país, así como también mostrar una gráfica individual de dispersión para los países con mayor número de suicidios.

```
In [33]: gdp_percapita = ds.groupby(by=['country', 'year', 'sex', 'per_capita']).agg({"su":
gdp_percapita.columns = ["totalsuicide"]
gdp_percapita.reset_index(inplace=True)

gdp_percapita

gdp_percapita_cng = gdp_percapita[gdp_percapita['totalsuicide'] < 40000]
plt.figure(figsize=(13, 8))
sns.scatterplot(x="per_capita", y="totalsuicide", hue="sex", data=gdp_percapita_cng)

plt.figtext(.5,.91,'Suicidios - PIB per capita (1985 - 2015)', color='b', fontsize=14)
plt.ylabel('Total de suicidios')
plt.xlabel('PIB per capita')
plt.show()
plt.show()
```



Como se puede ver los datos están más acumulados hacia la izquierda de la gráfica, lo cual indica que existe un mayor número de suicidios cuando los recursos económicos son más bajos y esto le afecta más a los hombres.

Ahora quisimos ver como se comportaba esto en los países con mayor número de suicidios.

```
In [34]: plt.figure(figsize=(30, 15))
países = ['Russian Federation', 'United States', 'Japan', 'France']
for i, column in enumerate(países):
    plt.subplot(2, 2, i+1)

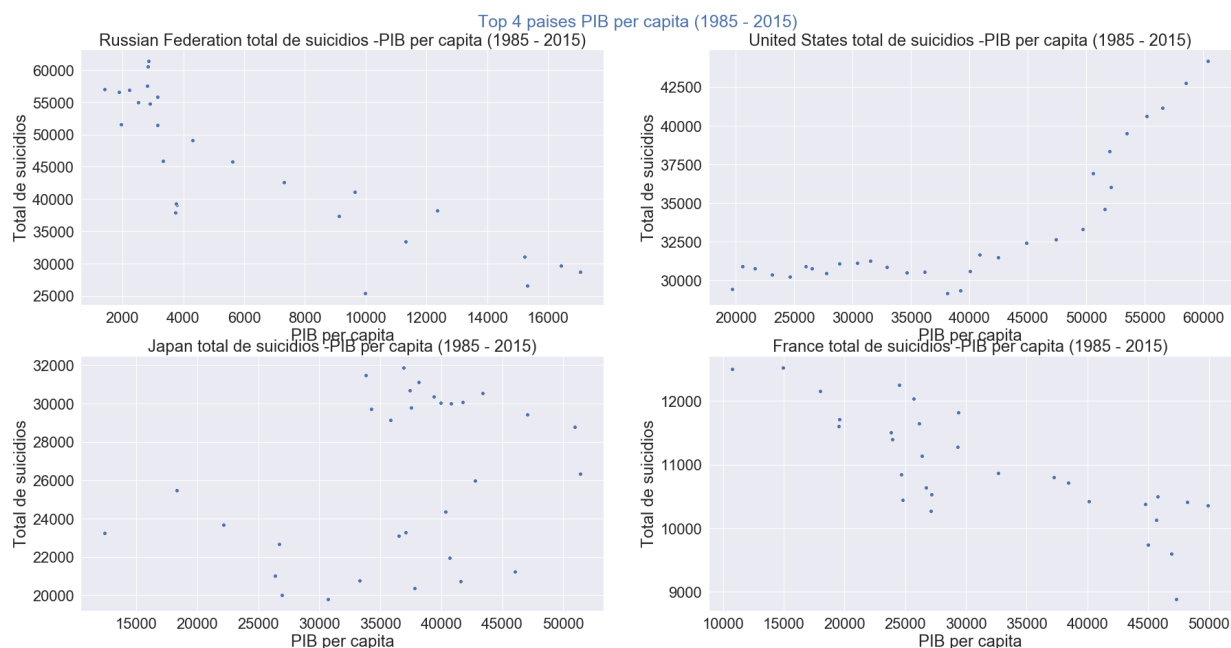
    t4países_gdp_pc = ds.groupby(by=['country', 'year', 'per_capita']).agg({"suicidios": "sum"})
    t4países_gdp_pc.columns = ["total_suicide"]
    t4países_gdp_pc.reset_index(inplace=True)

    gdp_cn = t4países_gdp_pc[t4países_gdp_pc['country'] == column]

    sns.scatterplot(x="per_capita", y="total_suicide", data=gdp_cn)
    plt.title(f"{column} total de suicidios - PIB per capita (1985 - 2015)")
    plt.xlabel('PIB per capita')
    plt.ylabel('Total de suicidios')

plt.figtext(.5, .91, 'Top 4 países PIB per capita (1985 - 2015)', color='b', fontstyle='italic')
```

```
Out[34]: Text(0.5, 0.91, 'Top 4 países PIB per capita (1985 - 2015)')
```



Gracias a esta gráfica pudimos ver como afecta el nivel de ingresos en los distintos países, vemos que en Rusia y Francia el mayor número de suicidios ocurren cuando los ingresos son menores, contrario de Estados Unidos donde se observa que mientras más riqueza se tenga mayor es el número de suicidios, por su parte Japón muestra los datos muy dispersos, por lo cual no podemos decir si existe una relación entre el PIB per cápita y el total de suicidios.

Conclusión General

Después de haber visualizado las gráficas y haber analizado los datos estadísticos llegamos a la conclusión de que el enfoque principal de los programas sociales para la prevención del suicidio

debe de ser dirigido a los hombres de entre 35 y 54 años, porque pudimos observar que estos son los que presentan mayor número de suicidios.

Durante la investigación de nuestro objetivo, nos encontramos con algo que nos llamó la atención que fue el alto número de suicidios que tenía Rusia y por ello decidimos indagar más a fondo en ese país y nos dimos cuenta que en 1994 fue el año más alarmante en cuanto a suicidios. Después de investigar, relacionamos este hecho con que a partir de 1992 se empezaron a aplicar políticas de carácter estructural, con el objetivo de construir una economía de mercado, y fue en 1994 cuando comenzó la etapa de transición, esto provocó que la producción industrial en ese año disminuyera en 50% de en comparación de la generada en 1990, lo cual tuvo como resultado una fuerte crisis en Rusia.

Por último, nuestras gráficas fueron enfocadas en nuestro objetivo secundario por lo que se tomó la columna del PIB per cápita y nos dimos cuenta que de manera general, entre más bajo los recursos económicos más suicidios ocurren. Al analizar el top 4 de países con más suicidios de manera individual, se observó que en dos de los cuatro, entre más alto el PIB menos suicidios lo que sigue la tendencia que se obtuvo de manera general pero hubo uno que nos arrojó lo contrario, entre más alto el PIB más suicidios, y también hubo una que mostraba los datos muy dispersos. Todo eso nos dice que la relación entre la economía y el número de suicidios es mejor analizarla por país dado que existe variación entre cada uno.