Minería de Datos
Grupo: 1
Dr. Guillermo Gilberto Molero Castillo

Ibañez Guzman Osvaldo
Semestre 2023-1
Tarea #2

Objetivo

El objetivo de la práctica es identificar una fuente de datos e importar dichos datos desde alguna plataforma Git, para posteriormente realizar un proceso de Análisis exploratorio de datos (EDA).

Contexto

Datos recopilados de la encuesta mundial sobre la salud de estudiantes adolescentes basado en los datos de 27 países, que proporciona un valor de porcentaje por cada uno de los valores.

Variables

- Country – País de procedencia
- Year – Año que se realizado la encuesta
- Age Group – Grupo de edades
- Sex – Genero de personas que realizaron la encuesta
- Currently_Drink_Alcohol – Bebia Alcohol regularmente
- Really_Get_Drunk – Realmente se emborrachaba
- Overwieght - Sobrepeso
- Use_Marijuana – Consumía Mariguana
- Have_Understanding_Parents – Tenia padres comprensivos
- Missed_classes_without_permssion – Faltaba a clases sin permiso
- Had_sexual_relation – Tenía relaciones sexuales
- Smoke_cig_currently – Fumaba regularmente
- Had_fights – Tenia Peleas
- Bullied – Sufría acoso
- Got_Seriously_injured – Tenia heridas serias
- No_close_friends – No tenía amigos cercanos
- Attempted_suicide – Intentos previos de suicidios

Fuente de datos

https://www.kaggle.com/datasets/kashishnaqvi/suicidal-behaviours-among-adolescents

Liga de Datos en GitHub

https://github.com/OsvaldoIG/MineriaDatos/blob/main/Data/GHSH_Pooled_Data1.csv

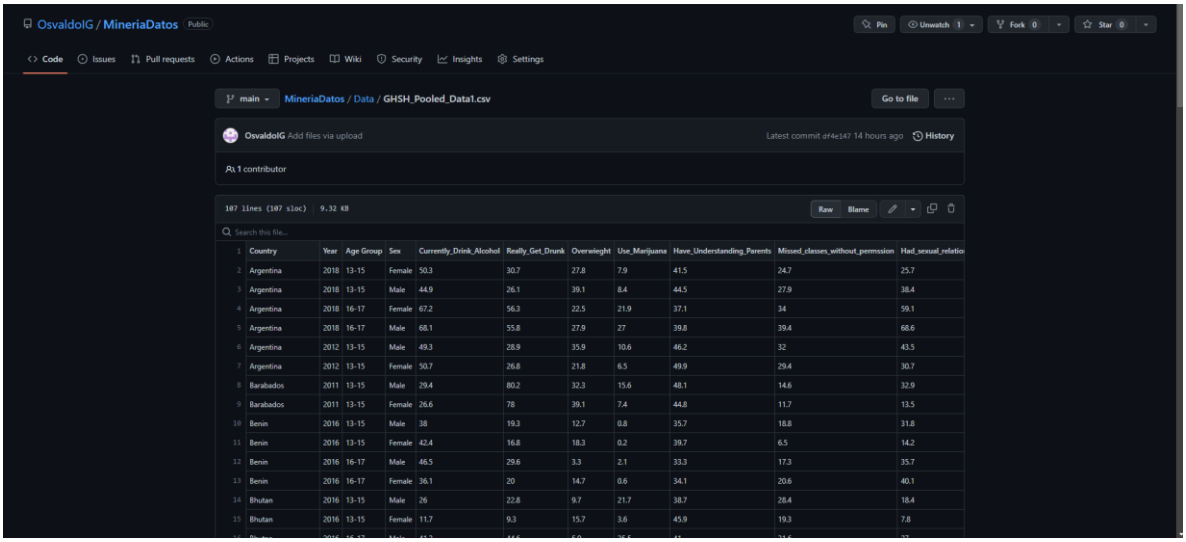
Desarrollo

La primer parte consistirá en la importación de datos a través de un repositorio en GitHub

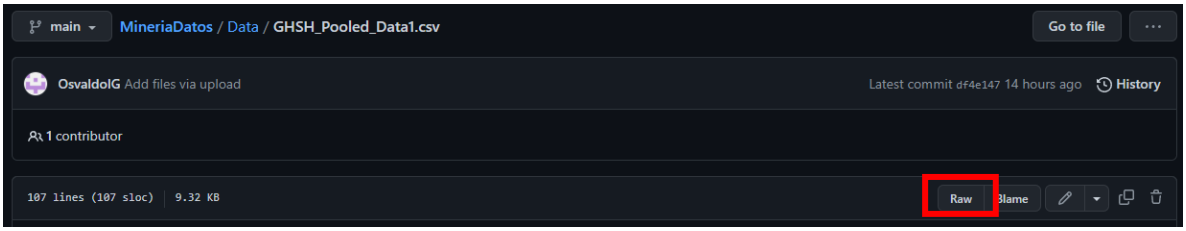
Para poder importar los datos será necesario importar la biblioteca “pandas” para la manipulación y análisis de los datos.

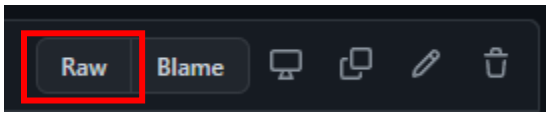
```
import pandas as pd
```

Al ya tener la librería necesitaremos el enlace de RAW de nuestro documento de GitHub, para obtener este link debemos abrir el documento desde nuestro GitHub.

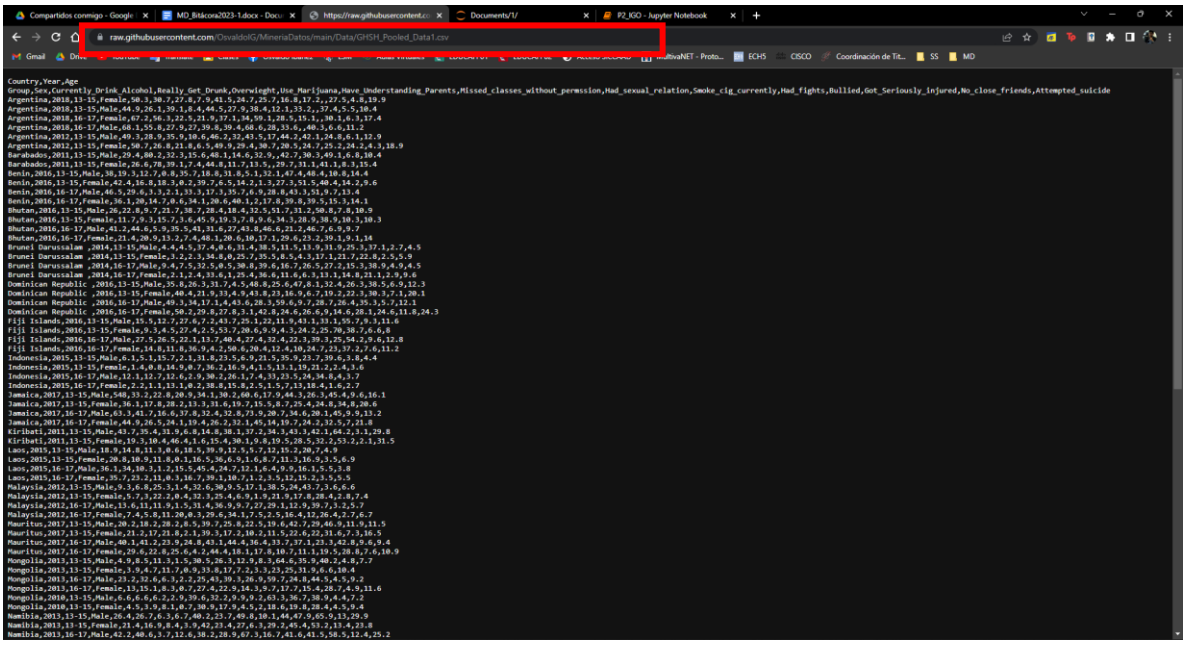


En la sección donde nos muestra la cantidad de líneas y el tamaño del archivo nos muestra dos opciones una que se llama Raw y otra Blame. Para abrir el enlace necesario debemos hacer click en Raw lo que nos desplegara la siguiente pantalla.





Después de esto se nos desplegará una pantalla con los datos y el link que necesitamos es el de esa página.



Liga de Datos RAW

https://raw.githubusercontent.com/OsvaldoIG/MineriaDatos/main/Data/GHSH_Pooled_Data1.csv

Ese enlace es que usaremos en los cuadernos de Jupyter de la siguiente manera. Con el enlace del documento CSV, usamos la función “read_csv” de la biblioteca pandas y finalmente la mostraremos en pantalla para ver los datos recuperados.

```
url = "https://raw.githubusercontent.com/OsvaldoIG/MineriaDatos/main/Data/GHSH_Pooled_Data1.csv"
DatosSuicidio = pd.read_csv(url)
DatosSuicidio
```

	Country	Year	Age Group	Sex	Currently_Drink_Alcohol	Really_Get_Drunk	Overweight	Use_Marijuana	Have_Understanding_Parents	Missed_classes_with
0	Argentina	2018	13-15	Female	50.3	30.7	27.8	7.9	41.5	
1	Argentina	2018	13-15	Male	44.9	26.1	39.1	8.4	44.5	
2	Argentina	2018	16-17	Female	67.2	56.3	22.5	21.9	37.1	
3	Argentina	2018	16-17	Male	68.1	55.8	27.9	27.0	39.8	
4	Argentina	2012	13-15	Male	49.3	28.9	35.9	10.6	46.2	
...	...	...	...	...	...	...	...	...	...	
101	Vanuatu	2011	13-15	Female	5.8	4.7	13.6	1.9	20.2	
102	Wallis and Futuna	2015	13-15	Male	32.2	35.5	60.5	4.0	36.3	
103	Wallis and Futuna	2015	13-15	Female	24.4	27.1	63.0	2.0	36.3	
104	Wallis and Futuna	2015	16-17	Male	48.3	53.7	57.8	10.1	36.5	
105	Wallis and Futuna	2015	16-17	Female	42.9	51.7	70.6	3.9	37.8	

106 rows × 17 columns

Una vez que contamos con los datos ya en la plataforma de Jupiter, procedemos a realizar el análisis exploratorio de los datos. Donde como sabemos lo primero es importar las bibliotecas, así como la importación de los datos.

El primer paso es la descripción de los datos, primeramente, conociendo el tamaño de nuestra matriz, lo cual lo haremos con el atributo *shape*, donde observaremos que tenemos una matriz con 106 registros y un total de 17 columnas.

```
DatosSuicidio.shape
```

```
(106, 17)
```

Posteriormente debemos identificar los tipos de variables con los que trabajaremos, usaremos la función *dtypes* que nos indica de que tipo de datos se trata cada una de las columnas, esto nos ayuda a identificar cuales son de tipo numéricos y cuales son texto. Observamos que trabajaremos con valores de tipo numérico y de tipo objeto (categóricas), donde esta última categoría únicamente son las columnas **Country**, **Age Group** y **Sex**.

```
DatosSuicidio.dtypes
```

```
Country                           object
Year                               int64
Age Group                         object
Sex                               object
Currently_Drink_Alcohol          float64
Really_Get_Drunk                 float64
Overwieght                       float64
Use_Marijuana                    float64
Have_Understanding_Parents       float64
Missed_classes_without_permssion float64
Had_sexual_relation              float64
Smoke_cig_currently              float64
Had_fights                       float64
Bullied                          float64
Got_Seriously_injured            float64
No_close_friends                 float64
Attempted_suicide                float64
dtype: object
```

Debemos identificar los datos faltantes en cada una de las columnas, usaremos la función de *pandas*, la cual nos devolverá el valor total de la cantidad de nulos existentes por cada una de las columnas *isnull().sum()*. Únicamente las columnas Smoke_cig_currently y Bullied contienen dos y cuatro valores nulos respectivamente, esto podría deberse a varias situaciones por ejemplo que el valor que debería estar colocado es 0, o que el registro estuvo incompleto al momento de realizarse.

```
DatosSuicidio.isnull().sum()
```

```
Country                             0
Year                                0
Age Group                           0
Sex                                 0
Currently_Drink_Alcohol             0
Really_Get_Drunk                    0
Overwieght                          0
Use_Marijuana                       0
Have_Understanding_Parents          0
Missed_classes_without_permssion    0
Had_sexual_relation                 0
Smoke_cig_currently                 2
Had_fights                          0
Bullied                             4
Got_Seriously_injured               0
No_close_friends                    0
Attempted_suicide                   0
dtype: int64
```

Una forma de unificar los dos pasos anteriores es con la función *info* que no muestra el tamaño de la tabla, los tipos de datos y los valores no nulos de cada registro.

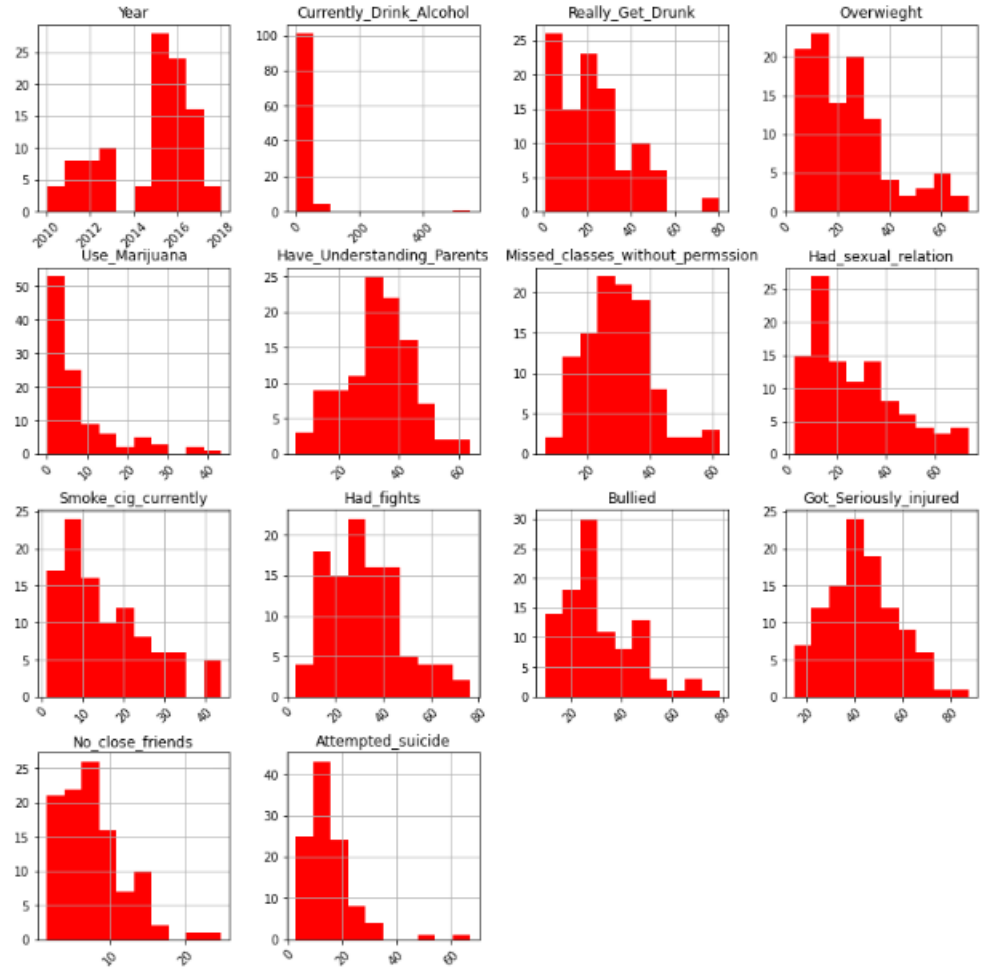
```
DatosSuicidio.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 106 entries, 0 to 105
Data columns (total 17 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   Country                           106 non-null    object
 1   Year                              106 non-null    int64
 2   Age Group                         106 non-null    object
 3   Sex                               106 non-null    object
 4   Currently_Drink_Alcohol           106 non-null    float64
 5   Really_Get_Drunk                  106 non-null    float64
 6   Overwieght                        106 non-null    float64
 7   Use_Marijuana                     106 non-null    float64
 8   Have_Understanding_Parents        106 non-null    float64
 9   Missed_classes_without_permssion  106 non-null    float64
 10  Had_sexual_relation               106 non-null    float64
 11  Smoke_cig_currently               104 non-null    float64
 12  Had_fights                        106 non-null    float64
 13  Bullied                           102 non-null    float64
 14  Got_Seriously_injured             106 non-null    float64
 15  No_close_friends                  106 non-null    float64
 16  Attempted_suicide                 106 non-null    float64
dtypes: float64(13), int64(1), object(3)
memory usage: 14.2+ KB
```

Procederemos a la detección de los valores atípicos, para las variables numéricas existen varias formas de hacerlo una de las más usadas son los histogramas, que muestra un grafica con los datos agrupados y así podremos identificar si existen datos atípicos o ver si existe un sesgo de alguno de ellos. Para esto debemos recordar que en los datos numéricos a excepción del año todos os datos están dados en porcentajes, por lo que es importante observar que todos estén en un rango de 0-100, y podemos observar que en **Currently_Drink_Alcohol**, existe por lo menos un

dato mayor al 400% lo cual no es algo lógico. Esto podría deberse a que alguien coloco mal un dato y deberemos rectificarlo o en su caso eliminarlo.

```
DatosSuicidio.hist(figsize=(14,14), xrot=45, color='red')
plt.show()
```



Otra forma de verlo es con ver los datos estadístico de las variables numéricas, usando la función *describe*, el cual nos mostrará la cantidad de datos registrados, el promedio de cada uno, así como su desviación estándar, además del mínimo y máximo y sus valores cada cuarto percentil. En dicha tabla podemos observar que el valor máximo de **Currently_Drink_Alcohol** es de 548%, esto como se menciono puede ser un error y en realidad podría ser 54.8%.

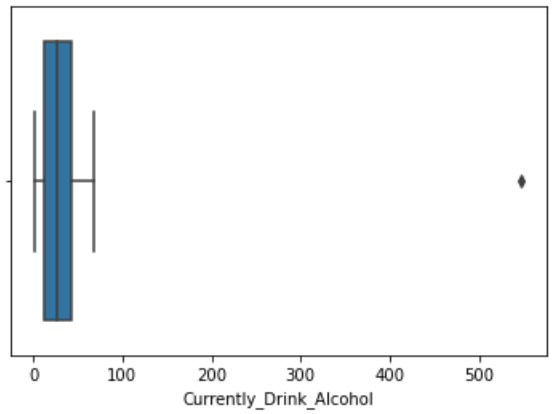
```
DatosSuicidio.describe()
```

	Year	Currently_Drink_Alcohol	Really_Get_Drunk	Overwieght	Use_Marijuana	Have_U
count	106.000000	106.000000	106.000000	106.000000	106.000000	
mean	2014.698113	31.815094	22.496226	23.694340	7.642453	
std	2.089292	53.454089	16.553129	15.764075	8.713536	
min	2010.000000	1.400000	0.800000	3.300000	0.000000	
25%	2013.000000	11.550000	9.000000	11.400000	2.025000	
50%	2015.000000	26.000000	19.650000	21.800000	4.350000	
75%	2016.000000	42.350000	30.475000	31.850000	9.575000	
max	2018.000000	548.000000	80.200000	70.600000	43.200000	

Otra forma de buscar datos atípicos es con diagramas de caja, que atreves de los dos procesos anteriores únicamente verificaremos **Currently_Drink_Alcohol** ya que es el único que presenta datos atípicos. En este caso podemos observar que efectivamente, únicamente contamos con un dato atípico de más del 500%

```
VariablesValoresAtipicos = ['Currently_Drink_Alcohol']
for col in VariablesValoresAtipicos:
  sns.boxplot(col, data=DatosSuicidio)
  plt.show()
```

```
D:\Users\osva_\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variable as a keyword arg: x. From version 0.12, the o
nly valid positional argument will be `data`, and passing other arguments witho
ut an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



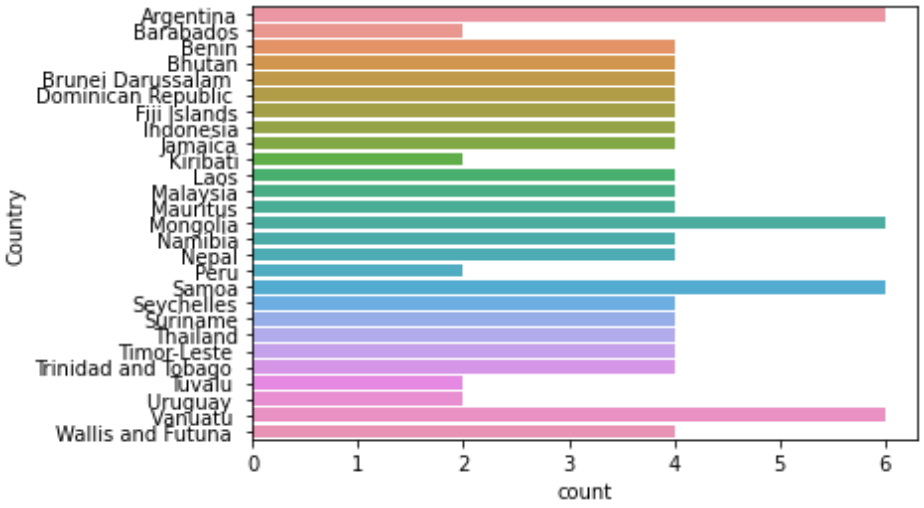
Una vez concluido con el análisis de las variables numéricas procederemos al análisis de las variables categóricas, donde mostraremos una tabla que muestra el recuento de los valores, así como las clases que existen, el valor con frecuencias mas alta y la frecuencia con la que aparece. En dicha tabla podemos observar que contaos con 27 países registrados, dos grupos de edad (13-15 y 16-17) y con dos valores para el sexo (Hombre y Mujer). Y en ninguna categoría se encuentran valores únicos.

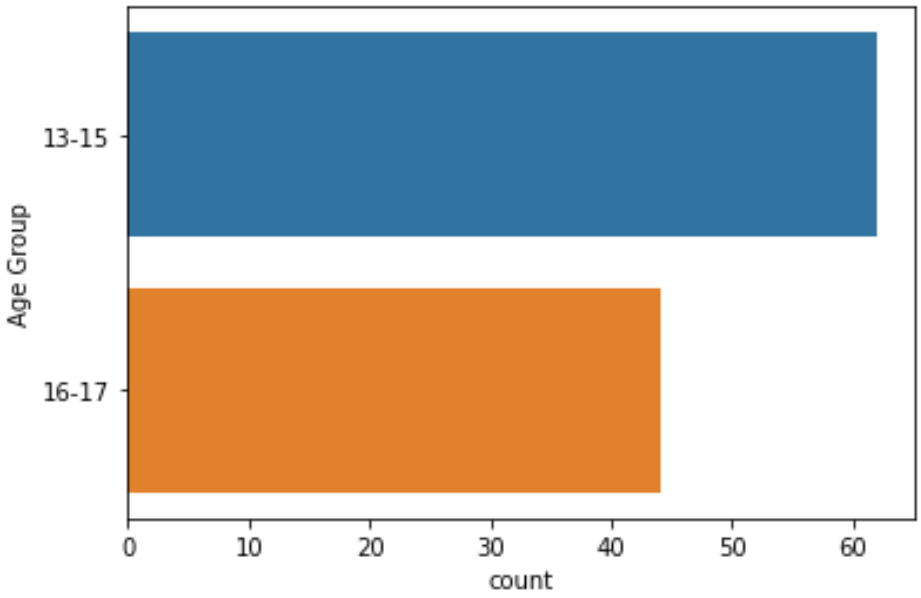
```
DatosSuicidio.describe(include='object')
```

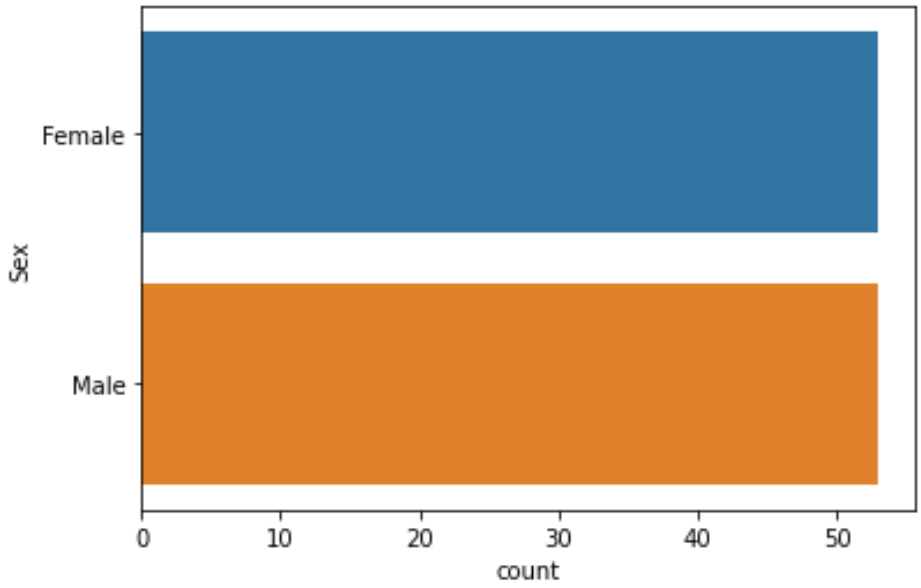
	Country	Age Group	Sex
count	106	106	106
unique	27	2	2
top	Argentina	13-15	Female
freq	6	62	53

Una forma más de observar estos datos, son con gráficos donde podamos observar la frecuencia de dichas variables, así podríamos observar que grupo de edades es donde se tiene mas registros o que países tienen más datos registrados. Al mostrar las graficas podemos observar que hay países como Argentina, Mongolia, Samoa y Vanuatu que cuentan con la mayor cantidad de registros (6) mientras que algunos otros países apenas cuentan con dos. Además, observaremos que tenemos un mayor registro de personas entre 13 y 15 años y la misma cantidad de hombres y mujeres en los registros.

```
for col in DatosSuicidio.select_dtypes(include='object'):
  if DatosSuicidio[col].all():sns.countplot(y=col, data=DatosSuicidio)
  plt.show()
```







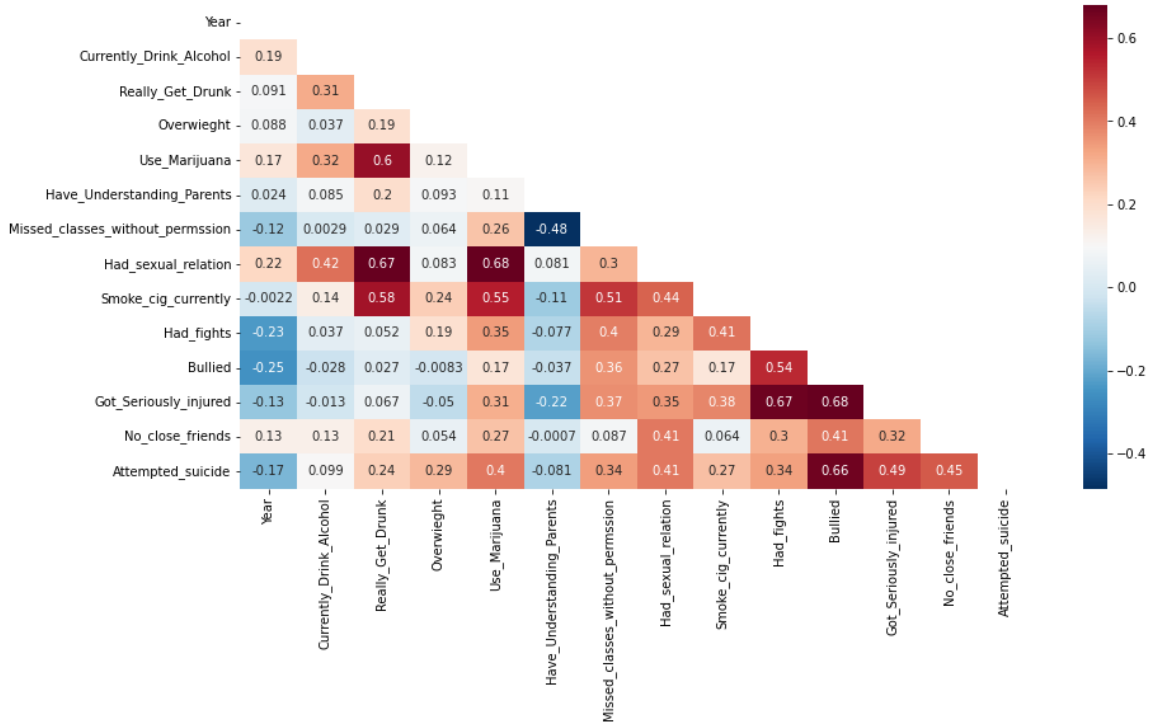
El siguiente paso es obtener una matriz de correlaciones entre las variables numéricas para identificar las relaciones que existen recordando que buscamos relaciones fuertes, para esto usaremos la función *corr()* que nos ayudara a obtener la correlación existente entre cada una de nuestras variables, para este análisis no considerare el valor del año ya que estoy buscando un análisis de las razone por las cuales los adolescentes han cometido suicidio, además de los que años varían según los registros de cada país.

```
DatosSuicidiosRec = DatosSuicidio.drop(columns='Year').corr()
DatosSuicidiosRec
```

	Currently_Drink_Alcohol	Really_Get_Drunk	Overwieght	Use_Marijuana	Have_Understanding_Parents
Currently_Drink_Alcohol	1.000000	0.311971	0.037212	0.318670	0.085079
Really_Get_Drunk	0.311971	1.000000	0.191082	0.604226	0.199064
Overwieght	0.037212	0.191082	1.000000	0.121040	0.092736
Use_Marijuana	0.318670	0.604226	0.121040	1.000000	0.105225
Have_Understanding_Parents	0.085079	0.199064	0.092736	0.105225	1.000000
Missed_classes_without_permssion	0.002931	0.029132	0.063789	0.261242	-0.483356
Had_sexual_relation	0.418399	0.674573	0.083223	0.675593	0.080762
Smoke_cig_currently	0.141118	0.584109	0.241447	0.554177	-0.114372
Had_fights	0.036944	0.052409	0.189777	0.346987	-0.077071
Bullied	-0.028085	0.026526	-0.008283	0.171492	-0.037443
Got_Seriously_injured	-0.012972	0.066561	-0.050310	0.310927	-0.221525
No_close_friends	0.131297	0.206292	0.053527	0.266877	-0.000703
Attempted_suicide	0.098731	0.235646	0.288114	0.403062	-0.081001

Esto es una forma complicada de visualizar los datos por lo que usaremos un mapa de calor donde valores mayorea a 0.66 o menores a -0.66 se consideraran correlaciones fuertes.

```
plt.figure(figsize=(14,7))
MatrizInf = np.triu(DatosSuicidio.corr())
sns.heatmap(DatosSuicidio.corr(),
            cmap='RdBu_r',
            annot=True,
            mask=MatrizInf)
plt.show()
```



Podemos observar una serie de relaciones fuertes, esto indica que son variables fuertemente relacionadas es decir si queremos reducir la dimensión de nuestros datos posiblemente podamos eliminarlos, entre estas relaciones fuertes tenemos

- Had_sexual_relation y Really_Get_Drunk
- Had_sexual_relation y Use_Marijuana
- Got_Seriously_injured y Had_fights
- Got_Seriously_injured y Bullied
- Attempted_suicide y Buillied

Para este caso sería bueno notar que la intención de suicidios previos esta fuertemente relacionada con ser una persona que sufre acoso y esta muy poco relacionada con el consumo de bebidas alcohólicas.

Enlace GitHub Tarea 2

Cuaderno Jupyter

https://github.com/OsvaldoIG/MineriaDatos/tree/main/T2

Liga de Datos en GitHub

https://github.com/OsvaldoIG/MineriaDatos/blob/main/Data/GHSH_Pooled_Data1.csv