cosijoeza

November 23, 2024

[212]:
```
import pandas as pd
import seaborn as sns
import numpy as np
from matplotlib import pyplot as plt
from statsmodels.graphics.gofplots import qqplot
```

1 Hipertensión Arterial en México

1.1 Lectura y despliegue de información de la base de datos

[213]:
```
dataHipertention = pd.read_csv('hipertension-arterial-mexico.csv')
```

[214]:
```
dataHipertention.head()
```

[214]:
```
         FOLIO_I  sexo  edad  concentracion_hemoglobina  temperatura_ambiente  \
0  2022_01001004     2    41                       14.2                    22
1  2022_01001009     2    65                       14.1                     9
2  2022_01001012     2    68                       14.2                    22
3  2022_01001013     1    35                       15.7                    11
4  2022_01001015     2    65                       12.7                     7

   valor_acido_urico  valor_albumina  valor_colesterol_hdl  \
0                4.8             4.0                    34
1                4.4             3.8                    73
2                4.8             4.0                    34
3                6.5             4.1                    49
4                4.2             4.2                    41

   valor_colesterol_ldl  valor_colesterol_total  …  segundamedicion_peso  \
0                  86.0                     139  …                 64.70
1                 130.0                     252  …                 96.75
2                  86.0                     139  …                 68.70
3                 107.0                     203  …                 64.70
4                  76.0                     145  …                 97.15

   segundamedicion_estatura  distancia_rodilla_talon  \
0                     154.0                     48.5
```

```
1                     152.2                     44.5
2                     144.8                     42.3
3                     154.0                     48.5
4                     161.3                     49.6

   circunferencia_de_la_pantorrilla  segundamedicion_cintura  \
0                              33.5                      0.0
1                              41.1                    113.7
2                              37.8                    103.7
3                              33.5                      0.0
4                              42.0                    118.9

   tension_arterial  sueno_horas  masa_corporal  actividad_total  \
0               107            4      32.889389              120
1               104            2       1.000000              240
2               105            1       1.000000              480
3               117            5      26.265339              275
4               123            2       1.000000              255

   riesgo_hipertension
0                    1
1                    0
2                    0
3                    1
4                    0

[5 rows x 36 columns]
```

```
[215]: dataHipertention.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4363 entries, 0 to 4362
Data columns (total 36 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   FOLIO_I                           4363 non-null   object
 1   sexo                              4363 non-null   int64
 2   edad                              4363 non-null   int64
 3   concentracion_hemoglobina         4363 non-null   float64
 4   temperatura_ambiente              4363 non-null   int64
 5   valor_acido_urico                 4363 non-null   float64
 6   valor_albumina                    4363 non-null   float64
 7   valor_colesterol_hdl              4363 non-null   int64
 8   valor_colesterol_ldl              4363 non-null   float64
 9   valor_colesterol_total            4363 non-null   int64
 10  valor_creatina                    4363 non-null   float64
 11  resultado_glucosa                 4363 non-null   float64
 12  valor_insulina                   4363 non-null   float64
 13  valor_trigliceridos              4363 non-null   int64
 14  resultado_glucosa_promedio       4363 non-null   int64
 15  valor_hemoglobina_glucosilada    4363 non-null   float64
 16  valor_ferritina                  4363 non-null   float64
 17  valor_folato                     4363 non-null   float64
 18  valor_homocisteina               4363 non-null   float64
 19  valor_proteinac_reactiva         4363 non-null   float64
 20  valor_transferrina               4363 non-null   float64
 21  valor_vitamina_bdoce             4363 non-null   float64
 22  valor_vitamina_d                 4363 non-null   float64
 23  peso                             4363 non-null   float64
 24  estatura                         4363 non-null   float64
 25  medida_cintura                   4363 non-null   float64
 26  segundamedicion_peso             4363 non-null   float64
 27  segundamedicion_estatura         4363 non-null   float64
 28  distancia_rodilla_talon          4363 non-null   float64
 29  circunferencia_de_la_pantorrilla 4363 non-null   float64
 30  segundamedicion_cintura          4363 non-null   float64
 31  tension_arterial                 4363 non-null   int64
 32  sueno_horas                      4363 non-null   int64
 33  masa_corporal                    4363 non-null   float64
 34  actividad_total                  4363 non-null   int64
 35  riesgo_hipertension              4363 non-null   int64
dtypes: float64(24), int64(11), object(1)
memory usage: 1.2+ MB
```

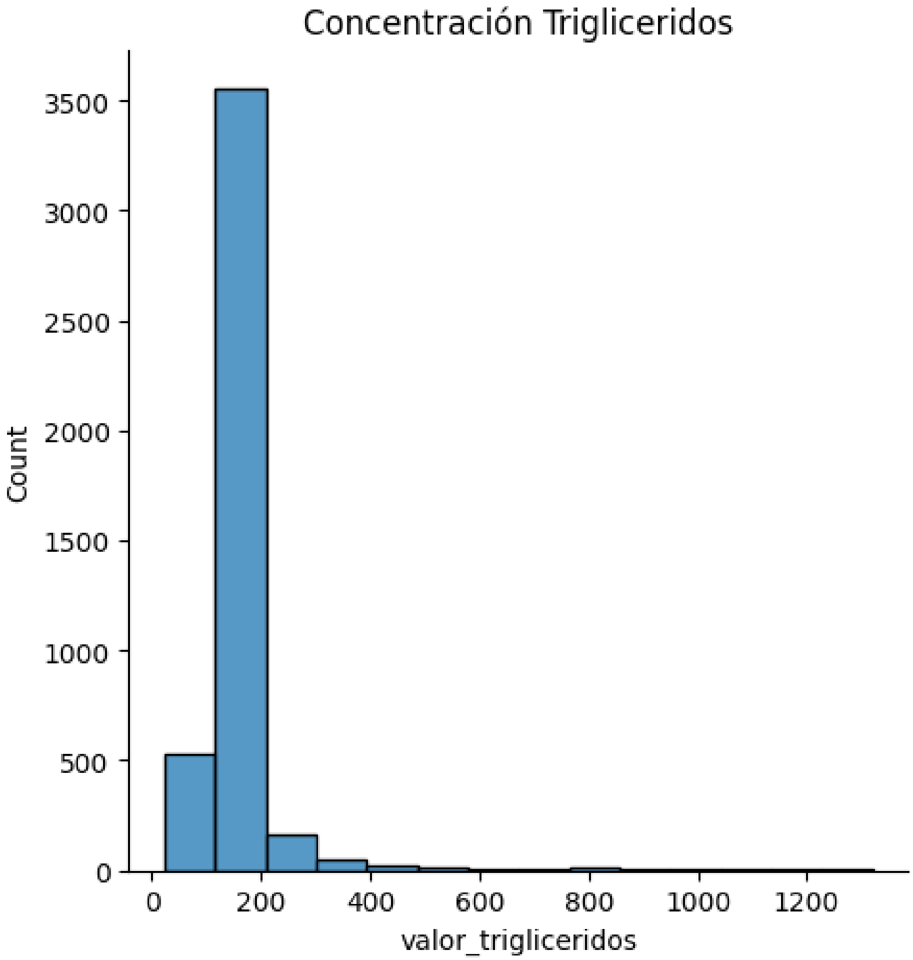
1.2 Grafica Concentración de Trigliceridos

Esta representación nos muestra que los valores de los trigliceridos tiene más valores entre 0 y 200.

[216]:
```
df = dataHipertention['valor_trigliceridos']
```

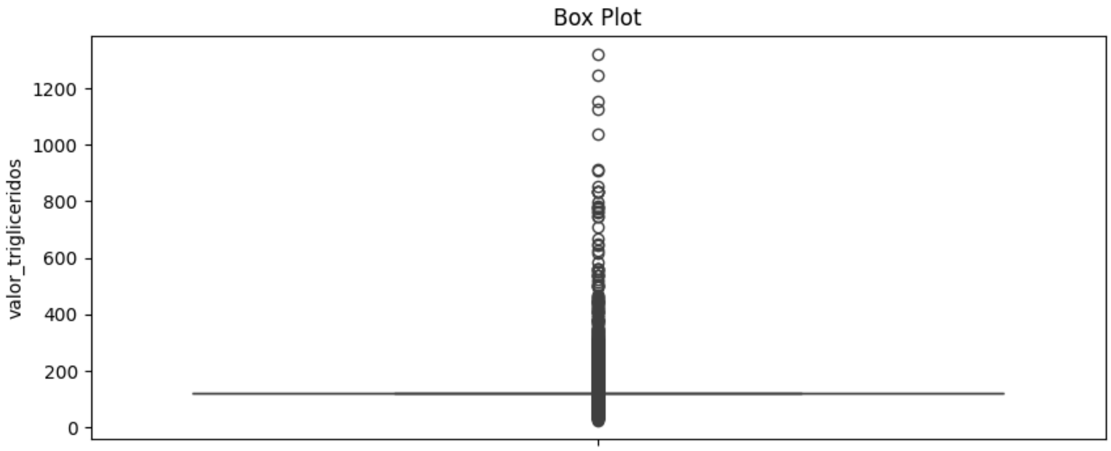
[217]:
```
plt.figure(figsize=(10,4))
sns.displot(df)
plt.title("Concentración Trigliceridos")
sns.despine()
plt.show()
```

```
<Figure size 1000x400 with 0 Axes>
```



1.3 Busqueda de Outliers en los Trigliceridos

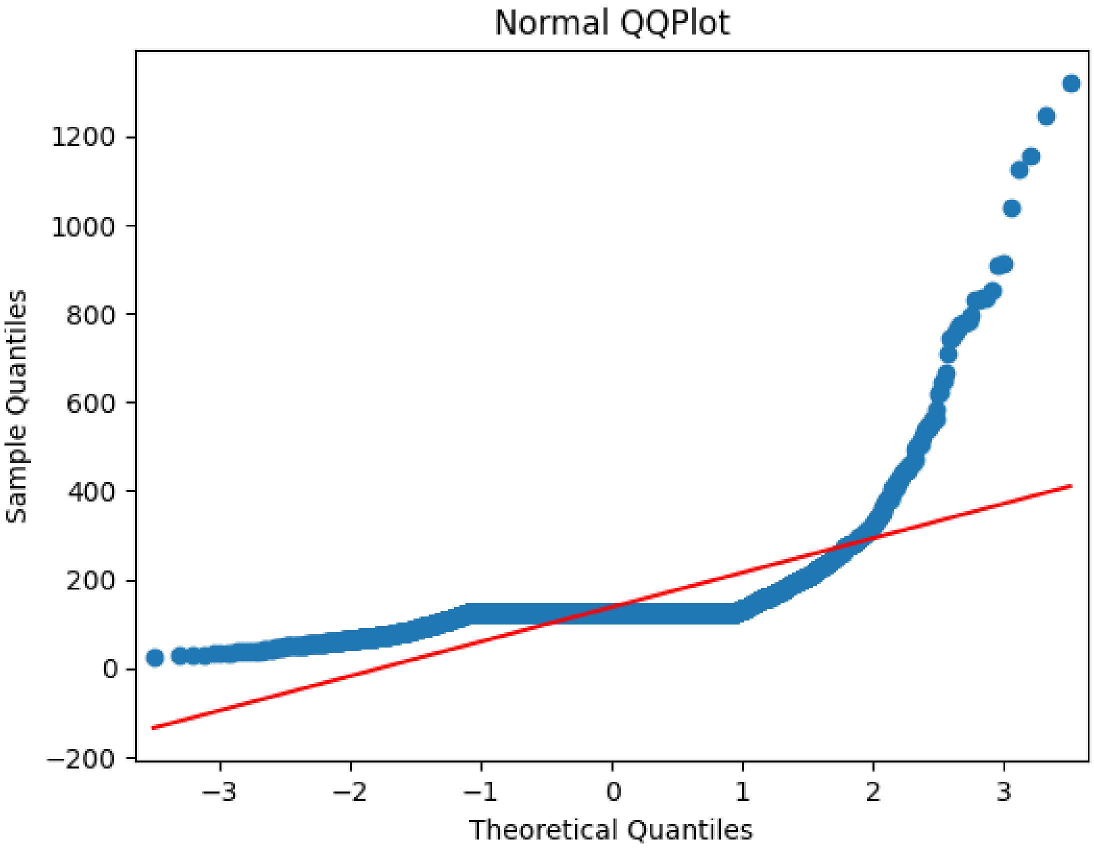
[218]:
```
plt.figure(figsize=(10,4))
plt.title("Box Plot")
sns.boxplot(df)
plt.show()
```



[219]:

```
plt.figure(figsize=(10,4))
qqplot(df,line='s')
plt.title("Normal QQPlot")
plt.show()
```

```
<Figure size 1000x400 with 0 Axes>
```



1.4 ¿Cuáles son los outliers y cuántos son?

[220]:

```
out = []
def Zscore_outlier(df,umbral):
  mean = np.mean(df)
  standarDesviation = np.std(df)
  for i in df:
      z = (i - mean) / standarDesviation
      if np.abs(z) > umbral:
        out.append(i)
  print("Outliers: ",out)
  return out
outliers = Zscore_outlier(df,umbral=3)
```

```
Outliers:  [376, 379, 452, 382, 507, 443, 438, 563, 773, 563, 408, 445, 423,
431, 835, 563, 440, 1040, 390, 832, 453, 456, 393, 500, 797, 446, 777, 582, 550,
432, 408, 382, 767, 373, 550, 1245, 758, 624, 445, 832, 499, 524, 540, 423, 375,
744, 835, 409, 470, 649, 745, 516, 417, 463, 461, 784, 618, 1320, 408, 852, 778,
1124, 910, 441, 501, 540, 418, 494, 407, 1154, 382, 400, 466, 667, 710, 536,
644, 381, 914, 537]
```

[221]:

```
print(len(outliers))
```

```
80
```

1.5 Gráfica Concentración de Tensión Arterial

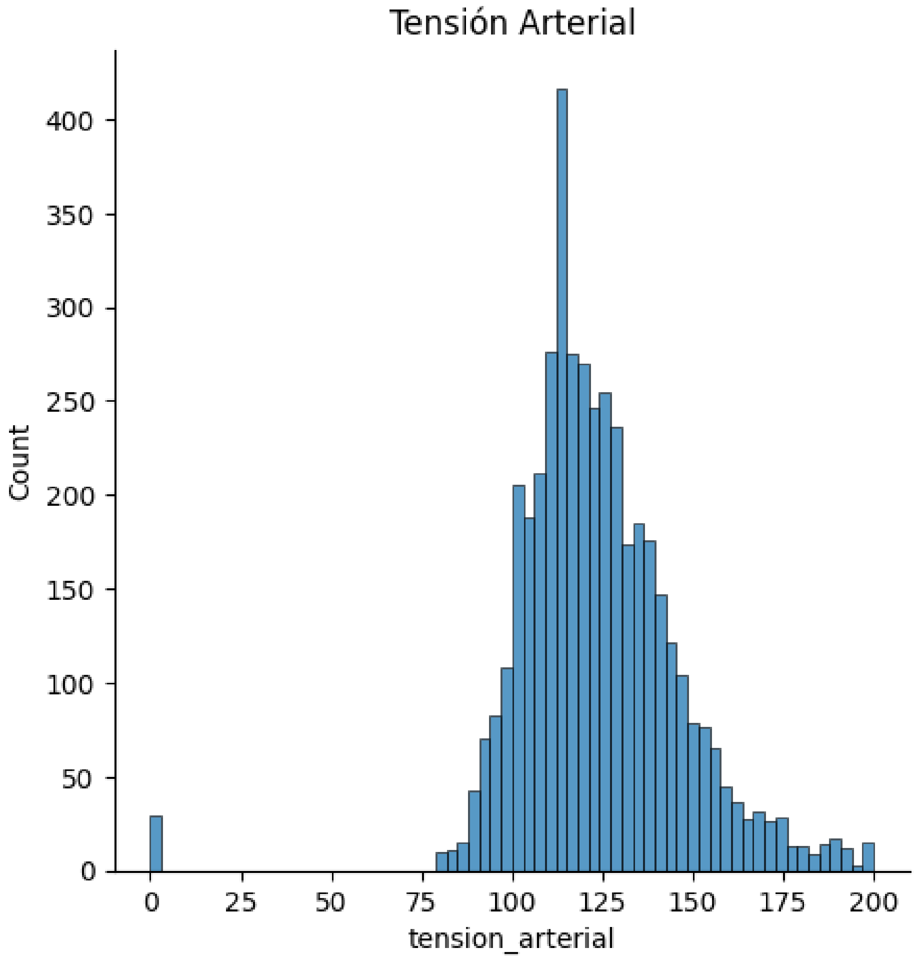
[222]:

```
df = dataHipertention['tension_arterial']
```

[223]:

```
plt.figure(figsize=(10,4))
sns.displot(df)
plt.title("Tensión Arterial")
sns.despine()
plt.show()
```

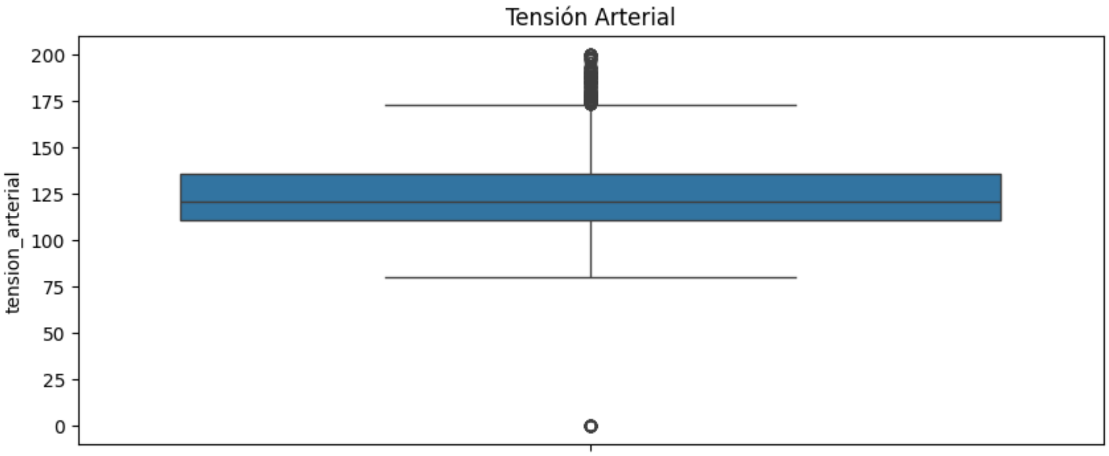
```
<Figure size 1000x400 with 0 Axes>
```



1.6 Busqueda de Outliers en la Tensión Arterial

[224]:

```
plt.figure(figsize=(10,4))
plt.title("Tensión Arterial")
sns.boxplot(df)
plt.show()
```



[225]:
```
plt.figure(figsize=(10,4))
sns.distplot(df)
plt.title("Tensión Arterial")
sns.despine()
plt.show()
```

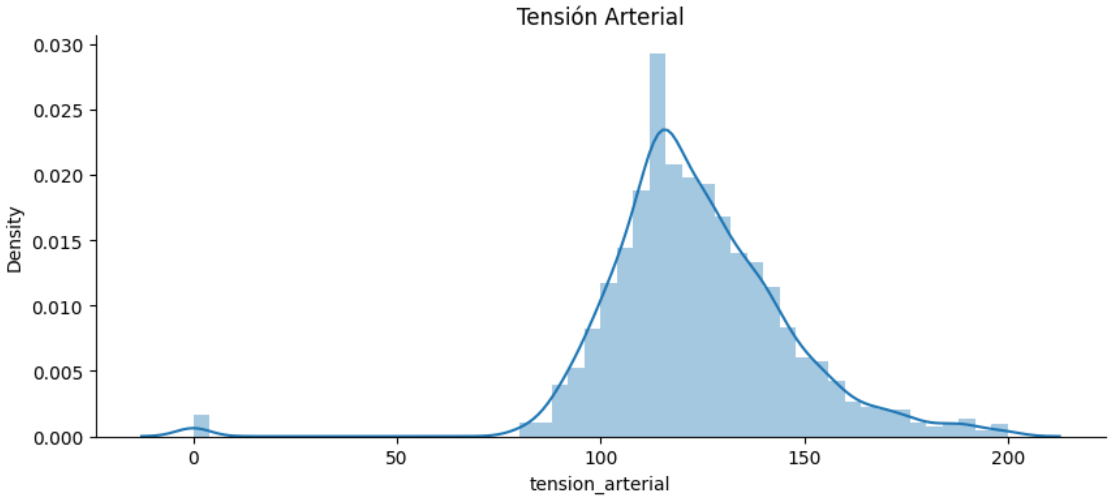
```
<ipython-input-225-fbbd1529f507>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df)
```



1.7 ¿Cuáles son los outliers y cuántos son?

[226]:

```
outliers = Zscore_outlier(df,umbral=3)
```

```
Outliers:  [376, 379, 452, 382, 507, 443, 438, 563, 773, 563, 408, 445, 423,
431, 835, 563, 440, 1040, 390, 832, 453, 456, 393, 500, 797, 446, 777, 582, 550,
432, 408, 382, 767, 373, 550, 1245, 758, 624, 445, 832, 499, 524, 540, 423, 375,
744, 835, 409, 470, 649, 745, 516, 417, 463, 461, 784, 618, 1320, 408, 852, 778,
1124, 910, 441, 501, 540, 418, 494, 407, 1154, 382, 400, 466, 667, 710, 536,
644, 381, 914, 537, 0, 0, 0, 193, 198, 0, 0, 0, 200, 193, 193, 200, 200, 0, 0,
0, 198, 199, 194, 0, 200, 0, 0, 192, 0, 0, 0, 199, 0, 197, 194, 0, 0, 0, 0, 0,
198, 192, 0, 200, 0, 197, 0, 200, 200, 0, 196, 0, 0, 200, 0, 0, 193]
```

[227]:

```
outliers.sort()
print(outliers)
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 192, 192, 193, 193, 193, 193, 194, 194, 196, 197, 197, 198, 198, 198,
199, 199, 200, 200, 200, 200, 200, 200, 200, 200, 373, 375, 376, 379, 381, 382,
382, 382, 390, 393, 400, 407, 408, 408, 408, 409, 417, 418, 423, 423, 431, 432,
438, 440, 441, 443, 445, 445, 446, 452, 453, 456, 461, 463, 466, 470, 494, 499,
500, 501, 507, 516, 524, 536, 537, 540, 540, 550, 550, 563, 563, 563, 582, 618,
624, 644, 649, 667, 710, 744, 745, 758, 767, 773, 777, 778, 784, 797, 832, 832,
835, 835, 852, 910, 914, 1040, 1124, 1154, 1245, 1320]
```

2 COVID-19 en Brazil

2.1 Lectura y despliegue de información de la base de datos

```
[228]: brazilCovid = pd.read_csv('brazil-covid19.csv')
```

```
[236]: brazilCovid.head()
```

```
[236]:         date   hour              state  suspects  refuses cases  deaths
       0  2020-01-30  16:00       Minas Gerais         1        0     0       0
       1  2020-01-30  16:00     Rio de Janeiro         1        0     0       0
       2  2020-01-30  16:00     Santa Catarina         0        2     0       0
       3  2020-01-30  16:00          São Paulo         3        1     0       0
       4  2020-01-30  16:00  Rio Grande do Sul         2        2     0       0
```

```
[230]: brazilCovid.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1008 entries, 0 to 1007
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   date       1008 non-null   object
 1   hour       684 non-null    object
 2   state      1008 non-null   object
 3   suspects   1008 non-null   int64
 4   refuses    1008 non-null   int64
 5   cases      1008 non-null   object
 6   deaths     1008 non-null   int64
dtypes: int64(3), object(4)
memory usage: 55.2+ KB
```

3 Terrorismo Global

3.1 Lectura y despliegue de información de la base de datos

```
[232]: terrorismoGlobal = pd.read_csv('global-terrorism.csv',encoding='latin-1')
```

```
<ipython-input-232-de8b786cd3cf>:1: DtypeWarning: Columns
(4,31,33,62,76,79,94,96,121) have mixed types. Specify dtype option on import or
set low_memory=False.
  terrorismoGlobal = pd.read_csv('global-terrorism.csv',encoding='latin-1')
```

```
[233]: terrorismoGlobal.head()
```

```
[233]:       eventid  iyear  imonth  iday approxdate  extended resolution  country  \
       0  197000000001   1970       7     2        NaN         0        NaN       58
       1  197000000002   1970       0     0        NaN         0        NaN      130
```

```
2  197001000001   1970          1     0        NaN         0        NaN      160
3  197001000002   1970          1     0        NaN         0        NaN       78
4  197001000003   1970          1     0        NaN         0        NaN      101

          country_txt  region  … addnotes scite1 scite2  scite3  dbsource  \
0  Dominican Republic       2  …      NaN    NaN    NaN     NaN      PGIS
1              Mexico       1  …      NaN    NaN    NaN     NaN      PGIS
2         Philippines       5  …      NaN    NaN    NaN     NaN      PGIS
3              Greece       8  …      NaN    NaN    NaN     NaN      PGIS
4               Japan       4  …      NaN    NaN    NaN     NaN      PGIS

   INT_LOG  INT_IDEO INT_MISC INT_ANY  related
0      0.0       0.0      0.0     0.0      NaN
1      0.0       1.0      1.0     1.0      NaN
2     -9.0      -9.0      1.0     1.0      NaN
3     -9.0      -9.0      1.0     1.0      NaN
4     -9.0      -9.0      1.0     1.0      NaN

[5 rows x 135 columns]
```

[239]:

```
terrorismoGlobal.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18870 entries, 0 to 18869
Columns: 135 entries, eventid to related
dtypes: float64(57), int64(22), object(56)
memory usage: 19.4+ MB
```

[ ]: