UNIVERSIDAD DE COLIMA

University of Colima
Faculty of Mechanical and Electrical Engineering
Intelligent Computer Engineering

Analysis of Dengue Cases in Mexico Until February 2024
Data analysis and visualization

Ceballos Contreras Jesús Eduardo 20176724
Larios Bravo Cristian Armando 20188165
López Aguilar Alba Beth-birai 20186885
Olmos Romero Nydia Naomi 20184561

6°D

Place: Mexico, Colima, Coquimatlan.
Date: 27/02/2024.

Main goal of this activity:

Apply the knowledge acquired about Pandas, NumPy and Matplotlib to perform an exploratory analysis of a real data set on dengue cases reported up to February 2024.

S5-U2-AC-01 Analysis of Dengue Cases in Mexico Until February 2024

INTRODUCTION

Dengue fever, also known as breakbone fever, is a mosquito-borne viral infection that causes flu-like symptoms such as high fever, severe headache, pain behind the eyes, joint and muscle pain, rash, and mild bleeding. The dengue virus is primarily transmitted to humans through the bite of infected female Aedes mosquitoes, particularly Aedes aegypti and Aedes albopictus. Dengue fever can range from mild to severe and can even be fatal in some cases, especially if left untreated or if the patient develops dengue hemorrhagic fever (DHF) or dengue shock syndrome (DSS). This disease is a major public health concern, particularly in tropical and subtropical regions of the world where the Aedes mosquito is prevalent. Efforts

to control the spread of dengue fever include vector control measures such as eliminating mosquito breeding sites, using insecticides, and implementing community health education programs.

Most relevant findings

- Classic Dengue have more victims vs Hemorrhagic Dengue

DEVELOPMENT:

Data preparation:

1. Load the dataset into a Pandas DataFrame.

Exploratory analysis:

2. Review the first rows of the DataFrame to understand the structure of the data.
3. Check and handle missing data (NaN values) if necessary.
4. Obtain basic statistical information about numerical variables.
5. Identify the categorical variables in the dataset.

Numerical Analysis. Measures of central tendency

The measures of central tendency of a dataset are basically represented by three parameters:

- The arithmetic mean or average.
- Fashion
- The standard deviation

6. Create code cells and obtain the three measures of central tendency from the numerical cells of the dataset.
7. Count the number of cases per category for each categorical variable. For example men, women, children, adolescents, adults and older adults
8. Obtain the number of cases of classic dengue and dengue hemorrhagic fever
9. Obtain the number of patients who died from each of the two types of dengue and by age category.

Visualization:

10. Visualize the distribution of dengue cases by category using bar graphs.
11. Visualize the number of dengue cases over time using a line graph or bar graph grouped by month.

12. Visualize the distribution of dengue cases by states of the mexican republic using bar graphs.

🧐 👑 🎁 🎏 🎉 😳

🚀 HARD CHALLENGE!!! Additional analysis:

13. Explore the correlation between at least two variables from the dataset columns in the dataset. Pandas has a function to calculate correlation.

Data preparation:

1. Load the dataset into a Pandas DataFrame.

In [ ]:
```
# Import project libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [ ]:
```
# Load the data
df = pd.read_csv("dengue_abierto.csv")
df
```

Out[ ]:

	FECHA_ACTUALIZACION	ID_REGISTRO	SEXO	EDAD_ANOS	ENTIDAD_RES	MUNICII
0	19/02/2024	1304867	1	11	30	
1	19/02/2024	1304870	2	12	25	
2	19/02/2024	1304893	1	24	23	
3	19/02/2024	1304896	2	38	30	
4	19/02/2024	1304928	2	4	31	
...	...	...	...	...	...	
26556	19/02/2024	1329824	1	89	23	
26557	19/02/2024	1329843	1	6	12	
26558	19/02/2024	1329898	1	51	16	
26559	19/02/2024	1329930	2	40	12	
26560	19/02/2024	1329959	2	29	12	

26561 rows × 28 columns

Exploratory analysis:

2. Review the first rows of the DataFrame to understand the structure of the data.

3. Check and handle missing data (NaN values) if necessary.
4. Obtain basic statistical information about numerical variables.
5. Identify the categorical variables in the dataset.

In [ ]:
```
# 2. Review the first rows of the DataFrame to understand the structure of the data
df.head()
```

Out[ ]:

	FECHA_ACTUALIZACION	ID_REGISTRO	SEXO	EDAD_ANOS	ENTIDAD_RES	MUNICIPIO_R
0	19/02/2024	1304867	1	11	30	
1	19/02/2024	1304870	2	12	25	
2	19/02/2024	1304893	1	24	23	
3	19/02/2024	1304896	2	38	30	2
4	19/02/2024	1304928	2	4	31	

5 rows × 28 columns

In [ ]:
```
# 3. Check and handle missing data (NaN values) if necessary.
# df.isnull().sum()
print(f"Data before dropping NaN values: \n{df.isna().sum()}")
```

```
Data before dropping NaN values:
FECHA_ACTUALIZACION       0
ID_REGISTRO               0
SEXO                      0
EDAD_ANOS                 0
ENTIDAD_RES               0
MUNICIPIO_RES             0
HABLA_LENGUA_INDIG        0
INDIGENA                  0
ENTIDAD_UM_NOTIF          0
MUNICIPIO_UM_NOTIF        0
INSTITUCION_UM_NOTIF      0
FECHA_SIGN_SINTOMAS       0
TIPO_PACIENTE             0
HEMORRAGICOS              0
DIABETES                  0
HIPERTENSION              0
ENFERMEDAD_ULC_PEPTICA    0
ENFERMEDAD_RENAL          0
INMUNOSUPR                0
CIRROSIS_HEPATICA         0
EMBARAZO                  0
DEFUNCION                 0
DICTAMEN                  1
TOMA_MUESTRA              0
RESULTADO_PCR             0
ESTATUS_CASO              0
ENTIDAD_ASIG              0
MUNICIPIO_ASIG            0
dtype: int64
```

In [ ]:
```
df = df.dropna()
print(f"Data after dropping NaN values: {df.isna().sum()}")
```

```
Data after dropping NaN values: FECHA_ACTUALIZACION       0
ID_REGISTRO                0
SEXO                       0
EDAD_ANOS                  0
ENTIDAD_RES                0
MUNICIPIO_RES              0
HABLA_LENGUA_INDIG         0
INDIGENA                   0
ENTIDAD_UM_NOTIF           0
MUNICIPIO_UM_NOTIF         0
INSTITUCION_UM_NOTIF       0
FECHA_SIGN_SINTOMAS        0
TIPO_PACIENTE              0
HEMORRAGICOS               0
DIABETES                   0
HIPERTENSION               0
ENFERMEDAD_ULC_PEPTICA     0
ENFERMEDAD_RENAL           0
INMUNOSUPR                 0
CIRROSIS_HEPATICA          0
EMBARAZO                   0
DEFUNCION                  0
DICTAMEN                   0
TOMA_MUESTRA               0
RESULTADO_PCR              0
ESTATUS_CASO               0
ENTIDAD_ASIG               0
MUNICIPIO_ASIG             0
dtype: int64
```

In [ ]:
```
# 4. Obtain basic statistical information about numerical variables.
df.describe()
```

Out[ ]:

	ID_REGISTRO	SEXO	EDAD_ANOS	ENTIDAD_RES	MUNICIPIO_RES	HABLA_L
count	2.656000e+04	26560.000000	26560.000000	26560.000000	26560.000000	
mean	1.315828e+06	1.474360	26.363215	17.077146	31.168562	
std	8.243211e+03	0.499352	17.318120	8.058535	62.438824	
min	1.299508e+06	1.000000	0.000000	1.000000	1.000000	
25%	1.308845e+06	1.000000	12.000000	12.000000	2.000000	
50%	1.315956e+06	1.000000	23.000000	14.000000	7.000000	
75%	1.322932e+06	2.000000	38.000000	23.000000	38.000000	
max	1.329959e+06	2.000000	110.000000	35.000000	570.000000	

8 rows × 26 columns

In [ ]:
```
# 5. Identify the categorical variables in the dataset.
categorical = df.select_dtypes(include=['object'])
```

```
categorical
```

Out[ ]:

	FECHA_ACTUALIZACION	FECHA_SIGN_SINTOMAS
0	19/02/2024	09/01/2024
1	19/02/2024	05/01/2024
2	19/02/2024	06/01/2024
3	19/02/2024	07/01/2024
4	19/02/2024	06/01/2024
...	...	...
26556	19/02/2024	14/02/2024
26557	19/02/2024	12/02/2024
26558	19/02/2024	13/02/2024
26559	19/02/2024	13/02/2024
26560	19/02/2024	12/02/2024

26560 rows × 2 columns

Numerical Analysis. Measures of central tendency

The measures of central tendency of a dataset are basically represented by three parameters:

- The arithmetic mean or average.
- Fashion
- The standard deviation

6. Create code cells and obtain the three measures of central tendency from the numerical cells of the dataset.
7. Count the number of cases per category for each categorical variable. For example men, women, children, adolescents, adults and older adults
8. Obtain the number of cases of classic dengue and dengue hemorrhagic fever
9. Obtain the number of patients who died from each of the two types of dengue and by age category.

In [ ]:
```
# Measures of central tendency of the dataset
# MEAN
print(f"Mean of the dataset per column:")
mean = df.describe()
mean = mean.loc['mean']
mean
```

```
Mean of the dataset per column:
```

Out[ ]:
```
ID_REGISTRO               1.315828e+06
SEXO                      1.474360e+00
EDAD_ANOS                 2.636322e+01
ENTIDAD_RES               1.707715e+01
MUNICIPIO_RES             3.116856e+01
HABLA_LENGUA_INDIG        1.989910e+00
INDIGENA                  1.987688e+00
ENTIDAD_UM_NOTIF          1.707617e+01
MUNICIPIO_UM_NOTIF        3.304032e+01
INSTITUCION_UM_NOTIF      7.408923e+00
TIPO_PACIENTE             1.207003e+00
HEMORRAGICOS              1.992093e+00
DIABETES                  1.978991e+00
HIPERTENSION              1.986596e+00
ENFERMEDAD_ULC_PEPTICA    1.999511e+00
ENFERMEDAD_RENAL          1.997515e+00
INMUNOSUPR                1.998343e+00
CIRROSIS_HEPATICA         1.999322e+00
EMBARAZO                  1.987236e+00
DEFUNCION                 1.997139e+00
DICTAMEN                  4.996875e+00
TOMA_MUESTRA              1.501431e+00
RESULTADO_PCR             4.823983e+00
ESTATUS_CASO              1.564044e+00
ENTIDAD_ASIG              1.724157e+01
MUNICIPIO_ASIG            3.278983e+01
Name: mean, dtype: float64
```

In [ ]:
```python
# Measures of central tendency of the dataset
# FASHION
print(f"Fashion of the dataset per column:")
modeDf = df.mode()
modeDf[0:1]

# pd.Series(df.values.flatten()).mode()[0]
```

Fashion of the dataset per column:

Out[ ]:

	FECHA_ACTUALIZACION	ID_REGISTRO	SEXO	EDAD_ANOS	ENTIDAD_RES	MUNICIPIO_R
0	19/02/2024	1299508	1.0	12.0	12.0	

1 rows × 28 columns

In [ ]:
```python
# Measures of central tendency of the dataset
# THE STANDARD DEVIATION
print(f"Standard deviation of the dataset per column:")
std = df.describe()
std = std.loc['std']
std
```

Standard deviation of the dataset per column:

Out[ ]:
```
ID_REGISTRO               8243.211008
SEXO                         0.499352
EDAD_ANOS                   17.318120
ENTIDAD_RES                  8.058535
MUNICIPIO_RES               62.438824
HABLA_LENGUA_INDIG           0.099945
INDIGENA                     0.110275
ENTIDAD_UM_NOTIF             8.065218
MUNICIPIO_UM_NOTIF          81.467962
INSTITUCION_UM_NOTIF         5.207107
TIPO_PACIENTE                0.405165
HEMORRAGICOS                 0.088569
DIABETES                     0.143417
HIPERTENSION                 0.114998
ENFERMEDAD_ULC_PEPTICA       0.022119
ENFERMEDAD_RENAL             0.049788
INMUNOSUPR                   0.040669
CIRROSIS_HEPATICA            0.026025
EMBARAZO                     0.112255
DEFUNCION                    0.053417
DICTAMEN                     0.060353
TOMA_MUESTRA                 0.500007
RESULTADO_PCR                0.628530
ESTATUS_CASO                 0.777935
ENTIDAD_ASIG                 8.802535
MUNICIPIO_ASIG              75.170623
Name: std, dtype: float64
```

In [ ]:
```
# 7. Count the number of cases per category for each categorical variable. For exam
# Per Sex
women = df[df["SEXO"]==1].shape[0]
men = df[df["SEXO"]==2].shape[0]
print(f"Number of cases per Sex \n\
Women: {women} \n\
Men: {men}")
# df["SEXO"].value_counts()
```

```
Number of cases per Sex
Women: 13961
Men: 12599
```

In [ ]:
```
# Per Age
children = df[df['EDAD_ANOS'] < 12].shape[0]
adolescents = df[(df['EDAD_ANOS'] >= 13) & (df['EDAD_ANOS'] < 18)].shape[0]
adults = df[(df['EDAD_ANOS'] >= 18) & (df['EDAD_ANOS'] < 60)].shape[0]
olderAdults = df[(df['EDAD_ANOS'] >= 60)].shape[0]
print(f"Number of cases per Age \n\
Children(0-12): {children} \n\
Adolescents(13-17): {adolescents} \n\
Adults(18-59): {adults} \n\
Older Adults(60+): {olderAdults}")
```

```
Number of cases per Age
Children(0-12): 5829
Adolescents(13-17): 3561
Adults(18-59): 15031
Older Adults(60+): 1280
```

In [ ]:
```python
# 8. Obtain the number of cases of classic dengue and dengue hemorrhagic fever
dengueHemorrhagic = df[df["HEMORRAGICOS"]==1].shape[0]
classicDengue = df[df["HEMORRAGICOS"]==2].shape[0]
print(f"Number of cases of classic dengue and dengue hemorrhagic fever \n\
Classic Dengue: {classicDengue} \n\
Dengue Hemorrhagic: {dengueHemorrhagic}")
```

```
Number of cases of classic dengue and dengue hemorrhagic fever
Classic Dengue: 26350
Dengue Hemorrhagic: 210
```

In [ ]:
```python
# 9. Obtain the number of patients who died from each of the two types of dengue an
classicDengueChi = df[(df["HEMORRAGICOS"]==2) & (df["DEFUNCION"]==1) & (df['EDAD_AN
classicDengueAdo = df[(df["HEMORRAGICOS"]==2) & (df["DEFUNCION"]==1) & (df['EDAD_AN
classicDengueAdu = df[(df["HEMORRAGICOS"]==2) & (df["DEFUNCION"]==1) & (df['EDAD_AN
classicDengueOld = df[(df["HEMORRAGICOS"]==2) & (df["DEFUNCION"]==1) & (df['EDAD_AN

dengueHemoChi = df[(df["HEMORRAGICOS"]==1) & (df["DEFUNCION"]==1) & (df['EDAD_ANOS'
dengueHemoAdo = df[(df["HEMORRAGICOS"]==1) & (df["DEFUNCION"]==1) & (df['EDAD_ANOS'
dengueHemoAdu = df[(df["HEMORRAGICOS"]==1) & (df["DEFUNCION"]==1) & (df['EDAD_ANOS'
dengueHemoOld = df[(df["HEMORRAGICOS"]==1) & (df["DEFUNCION"]==1) & (df['EDAD_ANOS'

print(f"Number of deaths per Age and type of Dengue\n\
        Classic Dengue\n\
Children(0-12): {classicDengueChi} \n\
Adolescents(13-17): {classicDengueAdo} \n\
Adults(18-59): {classicDengueAdu} \n\
Older Adults(60+): {classicDengueOld}\n\n\
        Dengue Hemorrhagic\n\
Children(0-12): {dengueHemoChi} \n\
Adolescents(13-17): {dengueHemoAdo} \n\
Adults(18-59): {dengueHemoAdu} \n\
Older Adults(60+): {dengueHemoOld}")
```

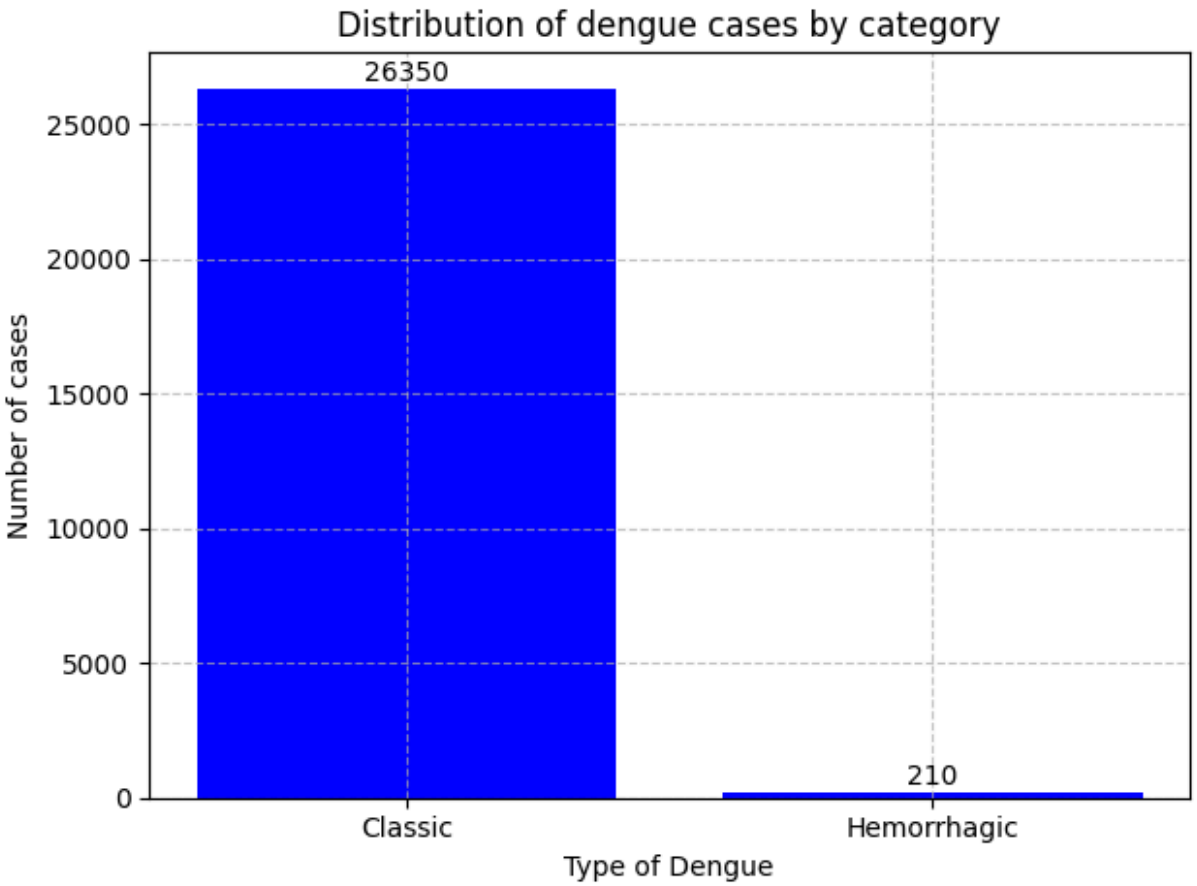
```
Number of deaths per Age and type of Dengue
        Classic Dengue
Children(0-12): 12
Adolescents(13-17): 8
Adults(18-59): 37
Older Adults(60+): 13

        Dengue Hemorrhagic
Children(0-12): 0
Adolescents(13-17): 1
Adults(18-59): 1
Older Adults(60+): 1
```

Visualization:

10. Visualize the distribution of dengue cases by category using bar graphs.

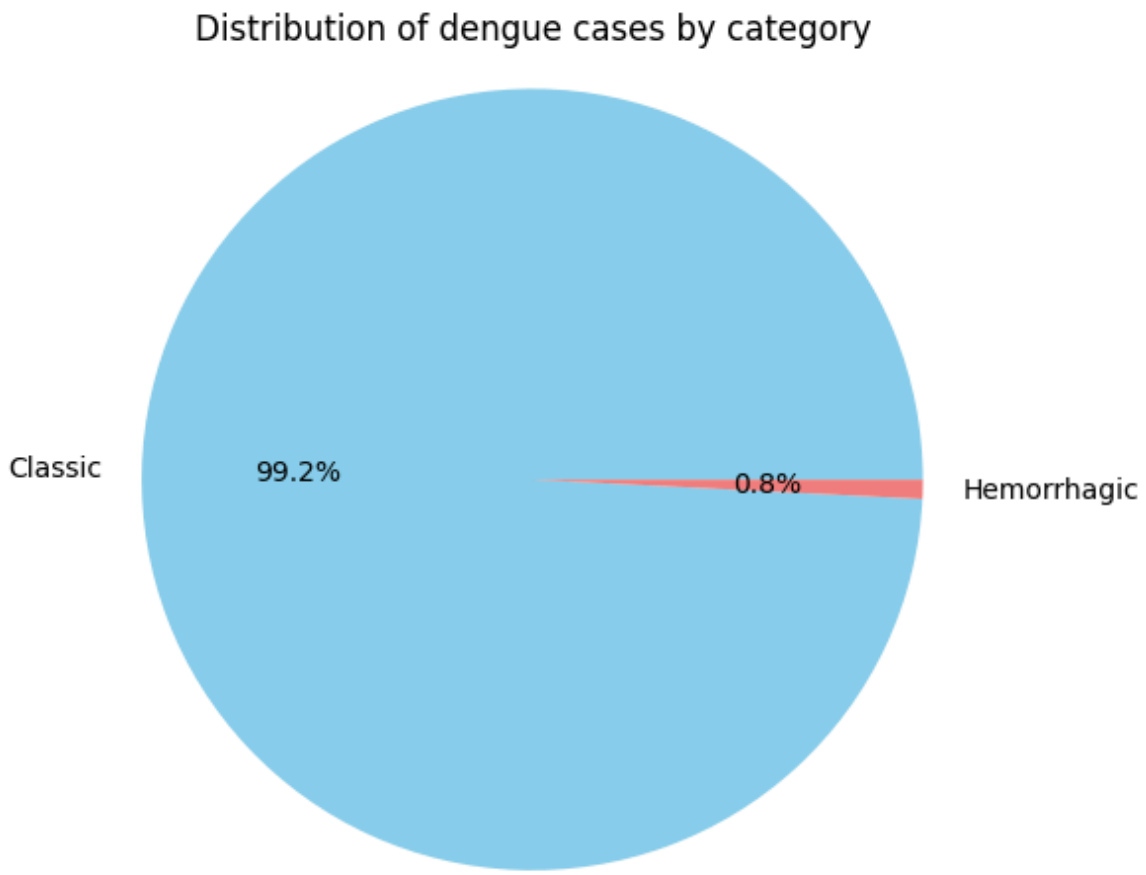
11. Visualize the number of dengue cases over time using a line graph or bar graph grouped by month.
12. Visualize the distribution of dengue cases by states of the mexican republic using bar graphs.

In [ ]:
```
# 10. Visualize the distribution of dengue cases by category using bar graphs.

labels = ["Classic", "Hemorrhagic"]

# Create the bar graph
plt.bar(labels,[classicDengue,dengueHemorrhagic], color='blue')
plt.title("Distribution of dengue cases by category")
plt.xlabel("Type of Dengue")
plt.ylabel("Number of cases")

# Set result values
for i, v in enumerate([classicDengue, dengueHemorrhagic]):
    plt.text(i, v + 50, str(v), ha='center', va='bottom')

# Mostrar la gráfica
plt.grid(True, linestyle="--", alpha=0.7) # Add grid
plt.tight_layout() # Design adjustments
plt.show()
```



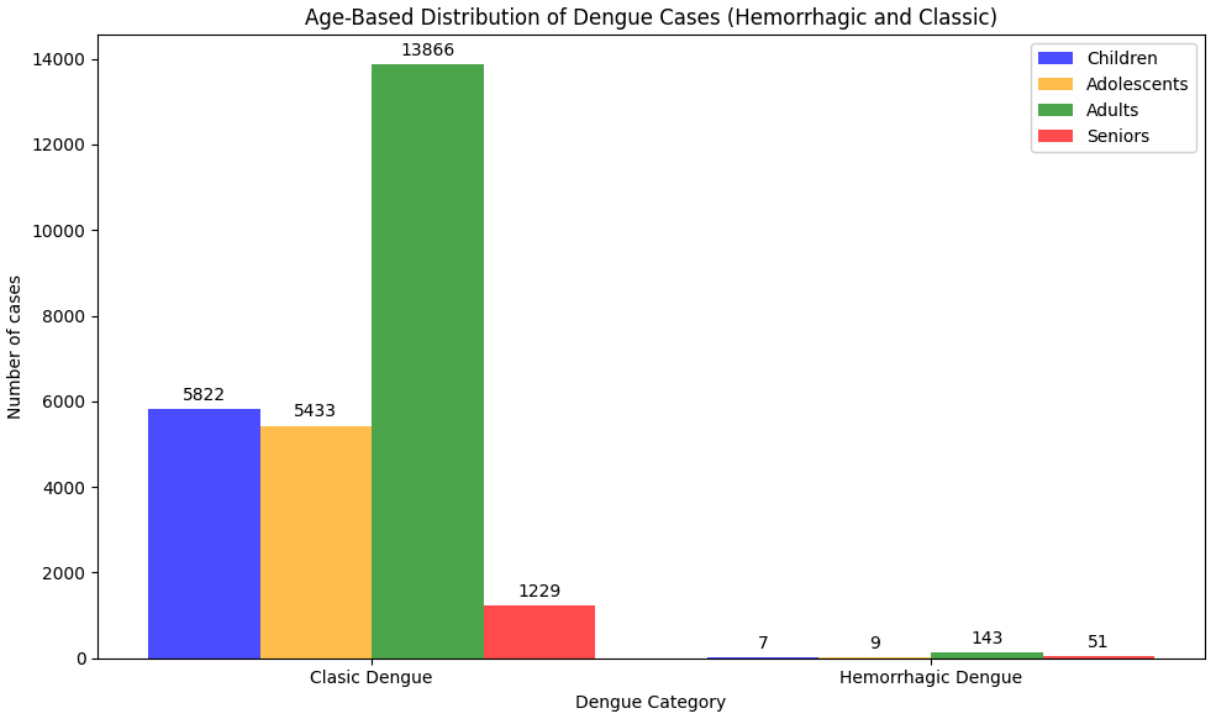
In [ ]:
```
# courtesy of KURT

labels = ["Classic", "Hemorrhagic"]
```

```
sizes = [classicDengue, dengueHemorrhagic]

# Create the pie chart
plt.pie(sizes, labels=labels, autopct='%1.1f%%', colors=['skyblue', 'lightcoral'])
plt.title("Distribution of dengue cases by category")

# Mostrar la gráfica
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.tight_layout() # Desgin adjustment
plt.show()
```



In [ ]:
```
children = df[df["EDAD_ANOS"] < 12]["HEMORRAGICOS"].value_counts()
adolescents = df[(df["EDAD_ANOS"] >= 12) & (df["EDAD_ANOS"] < 20)]["HEMORRAGICOS"].
adults = df[(df["EDAD_ANOS"] >= 20) & (df["EDAD_ANOS"] < 60)]["HEMORRAGICOS"].value
seniors = df[df["EDAD_ANOS"] >= 60]["HEMORRAGICOS"].value_counts()

plt.figure(figsize=(10, 6))

children_bar = plt.bar([0, 1], children, color='blue', alpha=0.7, width=0.2, label=
adolescents_bars = plt.bar([0.2, 1.2], adolescents, color='orange', alpha=0.7, widt
adults_bars = plt.bar([0.4, 1.4], adults, color='green', alpha=0.7, width=0.2, labe
seniors_bars = plt.bar([0.6, 1.6], seniors, color='red', alpha=0.7, width=0.2, labe

plt.title('Age-Based Distribution of Dengue Cases (Hemorrhagic and Classic)')
plt.xlabel('Dengue Category')
plt.ylabel('Number of cases')
plt.xticks([0.3, 1.3], ['Clasic Dengue', 'Hemorrhagic Dengue'])
plt.legend()
```

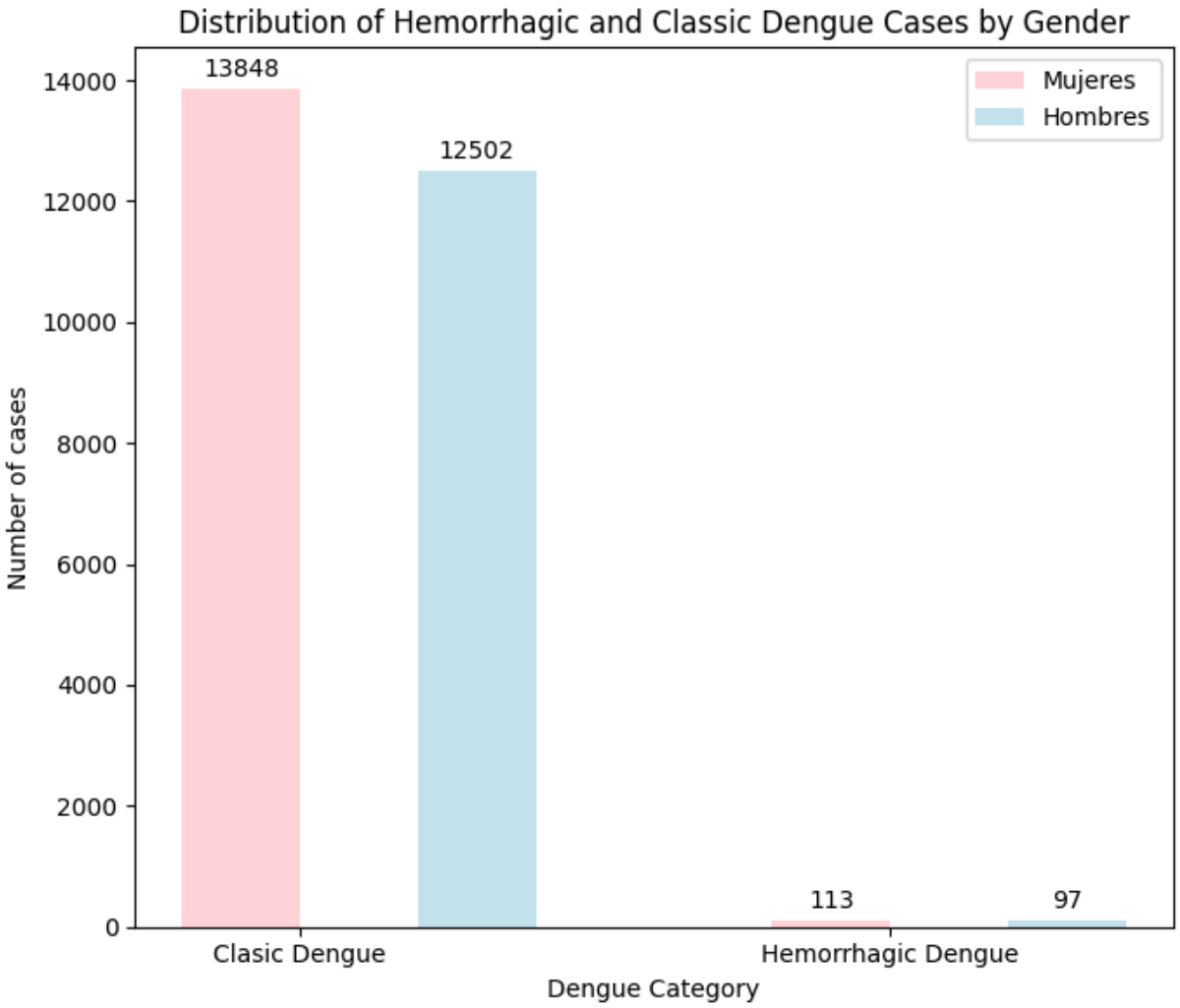
```
plt.tight_layout()

for bars in [children_bar, adolescents_bars, adults_bars, seniors_bars]:
    for bar in bars:
        height = bar.get_height()
        plt.annotate('{}'.format(height),
                     xy=(bar.get_x() + bar.get_width() / 2, height),
                     xytext=(0, 3),
                     textcoords="offset points",
                     ha='center', va='bottom')

plt.show()
```



In [ ]:

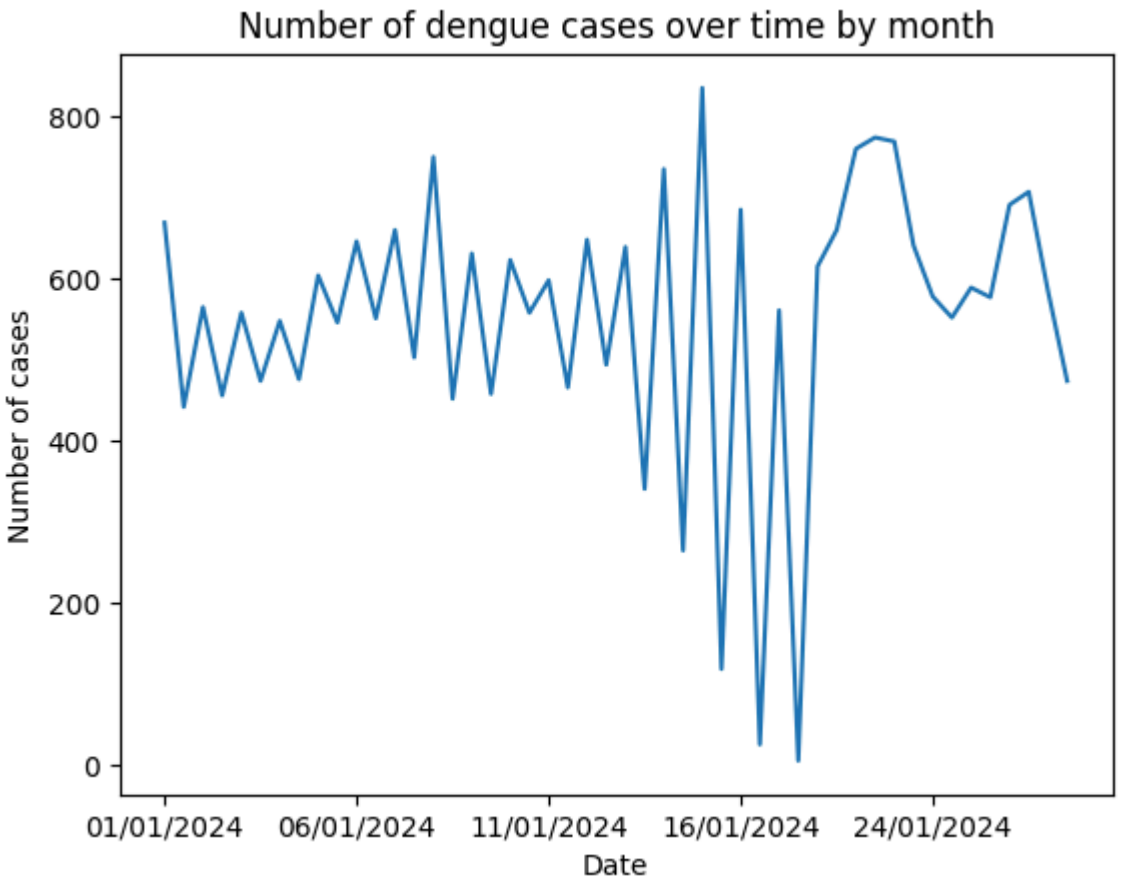
```
women = df[(df["SEXO"] == 1)]["HEMORRAGICOS"].value_counts()
men = df[(df["SEXO"] == 2)]["HEMORRAGICOS"].value_counts()

# Create the figure
plt.figure(figsize=(7, 6))

# Plot the data
women_bar = plt.bar([0.2, 1.2], women, color='pink', alpha=0.7, width=0.2, label='M
men_bar = plt.bar([0.6, 1.6], men, color='lightblue', alpha=0.7, width=0.2, label='

# Configure the chart
plt.title('Distribution of Hemorrhagic and Classic Dengue Cases by Gender')
plt.xlabel('Dengue Category')
plt.ylabel('Number of cases')
plt.xticks([0.3, 1.3], ['Clasic Dengue', 'Hemorrhagic Dengue'])
plt.legend()
plt.tight_layout()

 # Add number labels to each bar
for bars in [women_bar, men_bar]:
    for bar in bars:
```

```
        height = bar.get_height()
        plt.annotate('{}'.format(height),
                     xy=(bar.get_x() + bar.get_width() / 2, height),
                     xytext=(0, 3),
                     textcoords="offset points",
                     ha='center', va='bottom')

# Show the chart
plt.show()
```



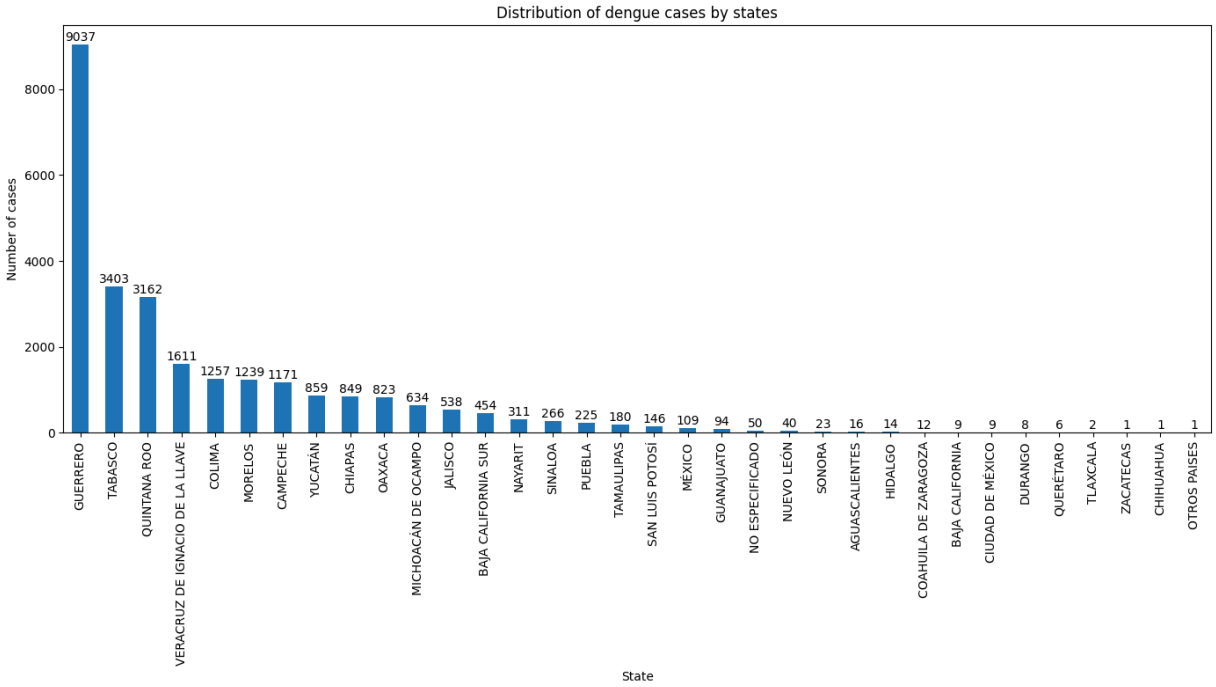
In [ ]:
```
# 11. Visualize the number of dengue cases over time using a line graph or bar grap
dengueCasesPerMonth = df["FECHA_SIGN_SINTOMAS"].value_counts().sort_index()
# dengueCasesPerMonth
dengueCasesPerMonth.plot(kind='line')
plt.title("Number of dengue cases over time by month")
plt.ylabel("Number of cases")
plt.xlabel("Date")
```

Out[ ]:
```
Text(0.5, 0, 'Date')
```



In [ ]:
```python
# 12. Visualize the distribution of dengue cases by states of the mexican republic
ENTIDAD = {
    1: 'AGUASCALIENTES',
    2: 'BAJA CALIFORNIA',
    3: 'BAJA CALIFORNIA SUR',
    4: 'CAMPECHE',
    5: 'COAHUILA DE ZARAGOZA',
    6: 'COLIMA',
    7: 'CHIAPAS',
    8: 'CHIHUAHUA',
    9: 'CIUDAD DE MÉXICO',
    10: 'DURANGO',
    11: 'GUANAJUATO',
    12: 'GUERRERO',
    13: 'HIDALGO',
    14: 'JALISCO',
    15: 'MÉXICO',
    16: 'MICHOACÁN DE OCAMPO',
    17: 'MORELOS',
    18: 'NAYARIT',
    19: 'NUEVO LEÓN',
    20: 'OAXACA',
    21: 'PUEBLA',
    22: 'QUERÉTARO',
    23: 'QUINTANA ROO',
    24: 'SAN LUIS POTOSÍ',
    25: 'SINALOA',
    26: 'SONORA',
```

```
    27: 'TABASCO',
    28: 'TAMAULIPAS',
    29: 'TLAXCALA',
    30: 'VERACRUZ DE IGNACIO DE LA LLAVE',
    31: 'YUCATÁN',
    32: 'ZACATECAS',
    33: 'ESTADOS UNIDOS DE AMERICA',
    34: 'OTROS PAISES DE AMERICA LATINA',
    35: 'OTROS PAISES',
    97: 'NO APLICA',
    98: 'SE IGNORA',
    99: 'NO ESPECIFICADO'
}
data = df.copy()
data["ENTIDAD_ASIG"] = df["ENTIDAD_ASIG"].map(ENTIDAD)
# df["ENTIDAD_ASIG"] = df["ENTIDAD_ASIG"].map(ENTIDAD)
dengueCasesByState = data["ENTIDAD_ASIG"].value_counts()
# dengueCasesByState.value_counts()
plt.figure(figsize=(14, 8))
dengueCasesByState.plot(kind='bar')
plt.title("Distribution of dengue cases by states")
plt.ylabel("Number of cases")
plt.xlabel("State")
# Set result values over the bars
for i, v in enumerate(dengueCasesByState):
    plt.text(i, v + 20, str(v), ha='center', va='bottom')

plt.xticks(rotation=90)  # Rotate eje x labels for better visualization
plt.tight_layout()
plt.show()
```



HARD CHALLENGE!!! Additional analysis:

13. Explore the correlation between at least two variables from the dataset columns in the dataset. Pandas has a function to calculate correlation.

In [ ]:
```
# 13. Explore the correlation between at least two variables from the dataset colum
# A good correlation depends on the use, but it is safe to say you have at least 0.
# The correlation of a variable with itself is 1.
# Correlation between Age and Hemorrhagic
data = pd.DataFrame(df, columns=["DEFUNCION", "EDAD_ANOS"])
data2 = pd.DataFrame(df, columns=["HEMORRAGICOS", "DEFUNCION"])

# Rename columns
data.columns = ["Death", "Age"]
data2.columns = ["Hemorrhagic", "Death"]

# correlation = df["DEFUNCION"].corr(df["HEMORRAGICOS"])
correlation = data.corr()
correlation2 = data2.corr()
print("Correlation between Death and Age: \n")
print(correlation)

print("\nCorrelation between Hemorrhagic and Death: \n")
print(correlation2)

df.drop(["FECHA_ACTUALIZACION","FECHA_SIGN_SINTOMAS"], axis=1, inplace=True)
print("\nCorrelation of the dataset: \n")
df.corr()
```

```
Correlation between Death and Age:

          Death       Age
Death  1.000000 -0.023745
Age   -0.023745  1.000000

Correlation between Hemorrhagic and Death:

             Hemorrhagic     Death
Hemorrhagic     1.000000  0.019093
Death           0.019093  1.000000

Correlation of the dataset:
```

```
C:\Users\nydia\AppData\Local\Temp\ipykernel_23308\1904365756.py:21: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  df.drop(["FECHA_ACTUALIZACION","FECHA_SIGN_SINTOMAS"], axis=1, inplace=True)
```

Out[ ]:

	ID_REGISTRO	SEXO	EDAD_ANOS	ENTIDAD_RES	MUNICIF
ID_REGISTRO	1.000000	-0.009409	-0.005020	-0.004411	-0
SEXO	-0.009409	1.000000	-0.073656	-0.009151	-0
EDAD_ANOS	-0.005020	-0.073656	1.000000	0.120655	0
ENTIDAD_RES	-0.004411	-0.009151	0.120655	1.000000	0
MUNICIPIO_RES	-0.001542	-0.018547	0.026521	0.190091	1
HABLA_LENGUA_INDIG	-0.002786	0.018958	-0.042673	0.007091	-0
INDIGENA	0.003002	0.010336	-0.031352	-0.007659	-0
ENTIDAD_UM_NOTIF	-0.004784	-0.010075	0.118882	0.992284	0
MUNICIPIO_UM_NOTIF	0.006170	-0.018326	0.029726	0.155925	0
INSTITUCION_UM_NOTIF	0.014436	-0.021142	-0.078818	-0.034346	0
TIPO_PACIENTE	-0.027127	-0.002983	-0.070429	-0.051296	0
HEMORRAGICOS	0.004290	0.002227	-0.109598	-0.019403	-0
DIABETES	0.009716	0.018240	-0.218348	-0.041927	-0
HIPERTENSION	0.008093	0.016964	-0.174439	-0.037360	-0
ENFERMEDAD_ULC_PEPTICA	0.000044	0.000568	-0.027943	-0.002323	-0
ENFERMEDAD_RENAL	0.001020	-0.007106	-0.061486	0.004325	-0
INMUNOSUPR	-0.002025	-0.015070	-0.028655	0.000045	0
CIRROSIS_HEPATICA	-0.004697	0.007355	-0.016496	-0.004598	-0
EMBARAZO	0.000916	0.108015	0.011139	-0.017059	-0
DEFUNCION	0.020970	-0.009809	-0.023745	-0.001237	-0
DICTAMEN	0.020030	-0.009530	-0.023914	-0.001749	-0
TOMA_MUESTRA	0.070744	0.017715	0.021675	-0.003536	0
RESULTADO_PCR	0.104175	-0.006162	0.066339	0.031829	0
ESTATUS_CASO	-0.181995	-0.025626	0.034621	0.043858	-0
ENTIDAD_ASIG	-0.004457	-0.003011	0.114836	0.904973	0
MUNICIPIO_ASIG	-0.000034	-0.008436	0.027439	0.152969	0

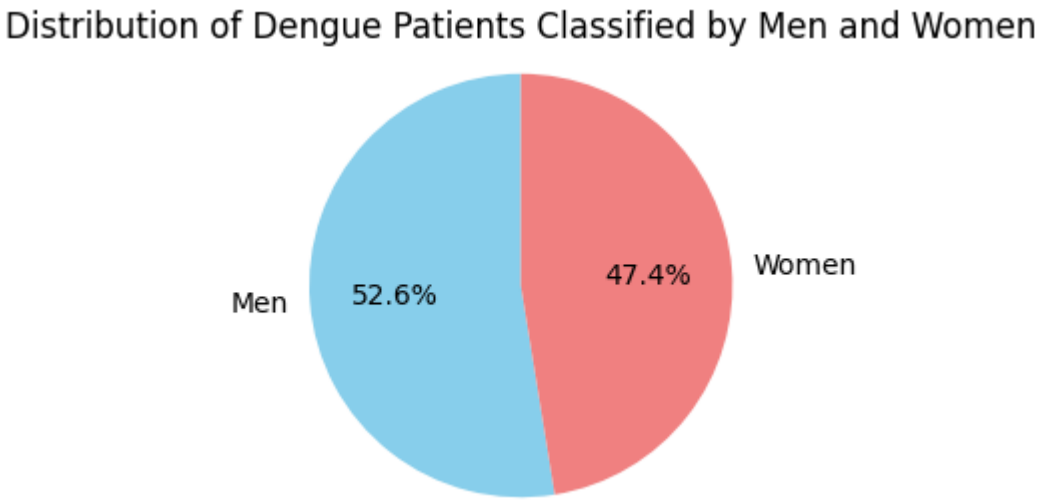
26 rows × 26 columns

In [ ]:
```
# Cases in Colima
colima = df['ENTIDAD_RES'].value_counts().get(6, 0)
print("Number of cases in the state of Colima, entity 6:", colima)
```

```
Number of cases in the state of Colima, entity 6: 1245
```

In [ ]:
```python
# Number of men and women in the dataset
# Count the values in the "Sexo" column
numperSex = df["SEXO"].value_counts()
print(numperSex)
# Create a pie chart
plt.figure(figsize=(3, 3))
plt.pie(numperSex, labels=['Men', 'Women'], autopct='%1.1f%%', colors=['skyblue', '
plt.title('Distribution of Dengue Patients Classified by Men and Women')
 # To make the pie chart circular
plt.axis('equal')
plt.show()
```

```
SEXO
1    13961
2    12599
Name: count, dtype: int64
```



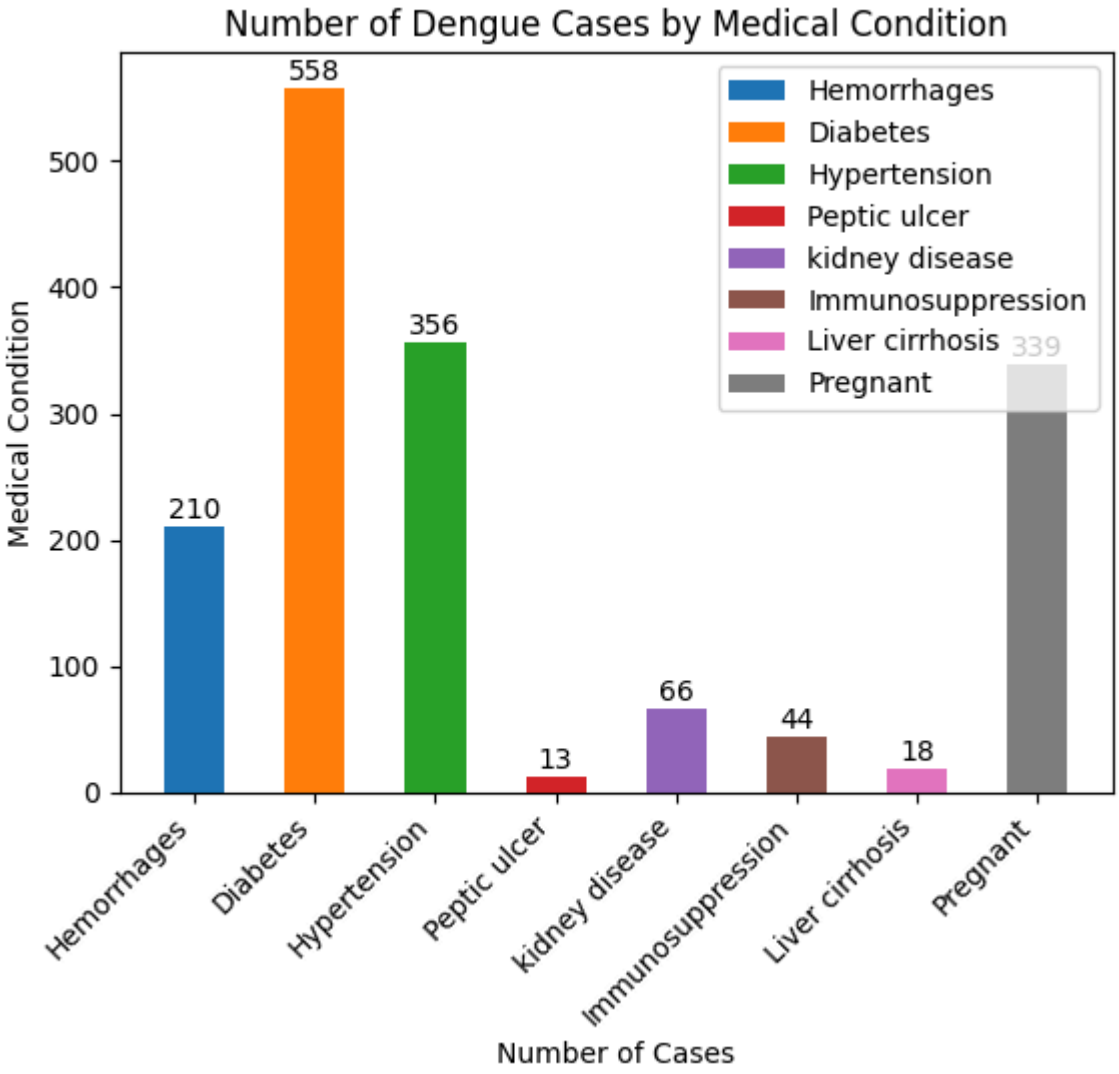
In [ ]:
```python
hemorrhagic = df[df["HEMORRAGICOS"] == 1]
num_hemorrhagic = len(hemorrhagic)
print("Number of cases with hemorrhages:", num_hemorrhagic)

diabetes = df[df["DIABETES"] == 1]
num_diabetes = len(diabetes)
print("Number of cases with diabetes:", num_diabetes)

hypertension = df[df["HIPERTENSION"] == 1]
num_hypertension = len(hypertension)
print("Number of cases with hypertension:", num_hypertension)

peptic_ulcer = df[df["ENFERMEDAD_ULC_PEPTICA"] == 1]
num_peptic_ulcer = len(peptic_ulcer)
print("Number of cases with peptic ulcer disease:", num_peptic_ulcer)

kidney = df[df["ENFERMEDAD_RENAL"] == 1]
num_kidney = len(kidney)
print("Number of cases with kidney disease:", num_kidney)

inmunosupr = df[df["INMUNOSUPR"] == 1]
```

```
num_inmunosupr = len(inmunosupr)
print("Number of cases with immunosuppression:", num_inmunosupr)

cirrhosis = df[df["CIRROSIS_HEPATICA"] == 1]
num_cirrhosis = len(cirrhosis)
print("Number of cases with liver cirrhosis:", num_cirrhosis)

pregnant = df[df["EMBARAZO"] == 1]
num_pregnant = len(pregnant)
print("Number of cases of pregnant patients:", num_pregnant)

total_cases = num_hemorrhagic + num_diabetes + num_hypertension + num_peptic_ulcer
print(f"Total number of patients with medical cases: {total_cases}")
```

```
Number of cases with hemorrhages: 210
Number of cases with diabetes: 558
Number of cases with hypertension: 356
Number of cases with peptic ulcer disease: 13
Number of cases with kidney disease: 66
Number of cases with immunosuppression: 44
Number of cases with liver cirrhosis: 18
Number of cases of pregnant patients: 339
Total number of patients with medical cases: 1604
```

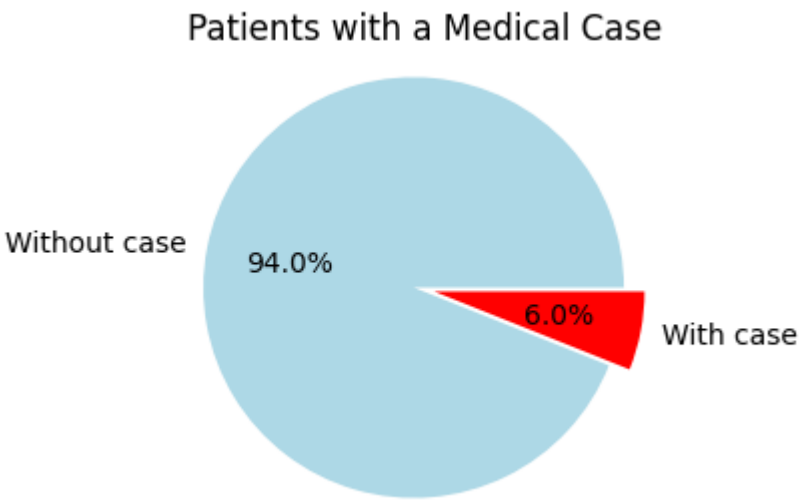
In [ ]:

```
fig, ax = plt.subplots()
ax.bar('Hemorrhages', num_hemorrhagic, 0.5, label='Hemorrhages')
ax.bar('Diabetes', num_diabetes, 0.5, label='Diabetes')
ax.bar('Hypertension', num_hypertension, 0.5, label='Hypertension')
ax.bar('Peptic ulcer', num_peptic_ulcer, 0.5, label='Peptic ulcer')
ax.bar('kidney disease', num_kidney, 0.5, label='kidney disease')
ax.bar('Immunosuppression', num_inmunosupr, 0.5, label='Immunosuppression')
ax.bar('Liver cirrhosis', num_cirrhosis, 0.5, label='Liver cirrhosis')
ax.bar('Pregnant', num_pregnant, 0.5, label='Pregnant')

ax.set_xlabel('Number of Cases')
ax.set_ylabel('Medical Condition')
ax.set_title('Number of Dengue Cases by Medical Condition')
# Add the quantities above each bar
for i, v in enumerate([num_hemorrhagic, num_diabetes, num_hypertension, num_peptic_
    ax.annotate(str(v), xy=(i, v), xytext=(0, 3), textcoords="offset points", ha='c
# at 45 degrees
plt.xticks(rotation=45, ha='right')
# Place the legend in the upper right corner
ax.legend(loc='upper right')
plt.show()
```



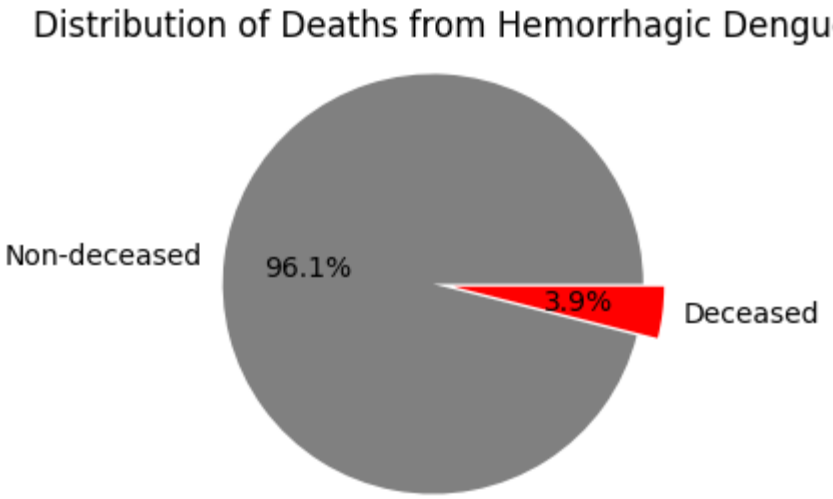
In [ ]:
```
# Determine the percentage represented by the total number of patients in the datas
labels = ['Without case', 'With case']
colors = ['lightblue', 'red']
# Print the number of deaths from hemorrhagic dengue

valor = [((df["EDAD_ANOS"].count())- total_cases), total_cases]
print(valor)
# Create the figure with a specific size
plt.figure(figsize=(3, 3))
# Highlight the deceased portion
explode = (0.1, 0)
# Plot pie chart
plt.pie(valor, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%')
plt.title('Patients with a Medical Case')
plt.axis('equal')
plt.show()
```

```
[24956, 1604]
```



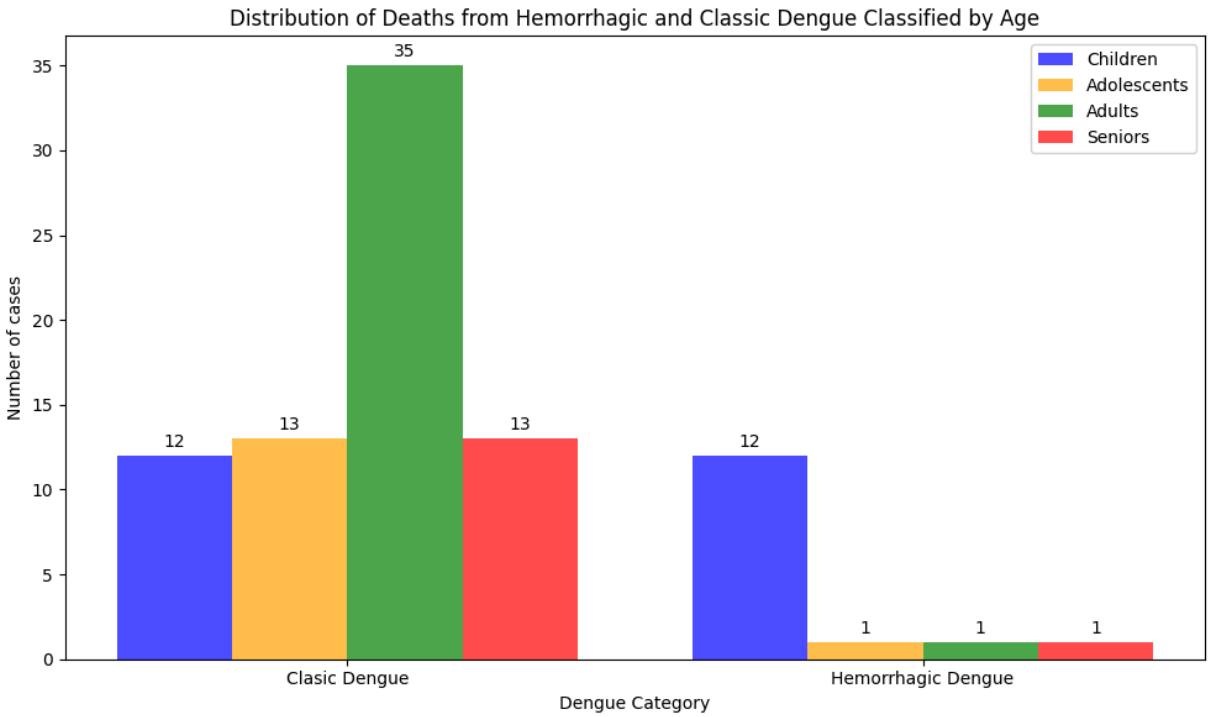
In [ ]:
```
# Calculate the percentage of deceased and non-deceased people using the "DEFUNCION
labels = ['Non-deceased','Deceased' ]
colors = ['grey', 'red']
# Print the number of deaths from hemorrhagic dengue
fallecidos = df[df['DEFUNCION'] == 1]['HEMORRAGICOS'].value_counts()
print("Deceased:{} ".format(fallecidos))

# Create a figure with a specific size
plt.figure(figsize=(3, 3))

# Highlight the deceased portion
explode = (0.1, 0)

# Plot pie chart
plt.pie(fallecidos, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%
plt.title('Distribution of Deaths from Hemorrhagic Dengue')
plt.axis('equal')
plt.show()
```

```
Deceased:HEMORRAGICOS
2    73
1     3
Name: count, dtype: int64
```



In [ ]:

```
children = df[(df["EDAD_ANOS"] < 12) & (df['DEFUNCION'] == 1)]["HEMORRAGICOS"].valu
adolescents = df[(df["EDAD_ANOS"] >= 12) & (df["EDAD_ANOS"] < 20) & df['DEFUNCION']
adults = df[(df["EDAD_ANOS"] >= 20) & (df["EDAD_ANOS"] < 60)& (df['DEFUNCION'] == 1
seniors = df[(df["EDAD_ANOS"] >= 60) & (df['DEFUNCION'] == 1)]["HEMORRAGICOS"].valu

# Create the figure
plt.figure(figsize=(10, 6))

# Plot the data
children_bar = plt.bar([0, 1], children, color='blue', alpha=0.7, width=0.2, label=
adolescents_bars = plt.bar([0.2, 1.2], adolescents, color='orange', alpha=0.7, widt
adults_bars = plt.bar([0.4, 1.4], adults, color='green', alpha=0.7, width=0.2, labe
seniors_bars = plt.bar([0.6, 1.6], seniors, color='red', alpha=0.7, width=0.2, labe

# Configure the chart
plt.title('Distribution of Deaths from Hemorrhagic and Classic Dengue Classified by
plt.xlabel('Dengue Category')
plt.ylabel('Number of cases')
plt.xticks([0.3, 1.3], ['Clasic Dengue', 'Hemorrhagic Dengue'])
plt.legend()
plt.tight_layout()

 # Add number labels to each bar
for bars in [children_bar, adolescents_bars, adults_bars, seniors_bars]:
    for bar in bars:
        height = bar.get_height()
        plt.annotate('{}'.format(height),
                     xy=(bar.get_x() + bar.get_width() / 2, height),
                     xytext=(0, 3),
                     textcoords="offset points",
                     ha='center', va='bottom')

# Show the chart
plt.show()
```



Box or Whisker Plots

Boxplots, also known as boxplots, are a graphical tool used to represent the distribution of a set of numerical data and display various important statistics. They consist of a box with extended lines (whiskers) at both ends. The main components of a box plot are:

1. **Box**: The box represents the interquartile range (IQR), which covers the middle 50% of the data. The bottom of the box indicates the first quartile (Q1) and the top indicates the third quartile (Q3). The height of the box shows the spread of the data within this range.

2. **Line in the box (Median)**: A line inside the box marks the median of the data, which is the value that divides the data set into two equal parts: the lower 50% and the lower 50%. superior.

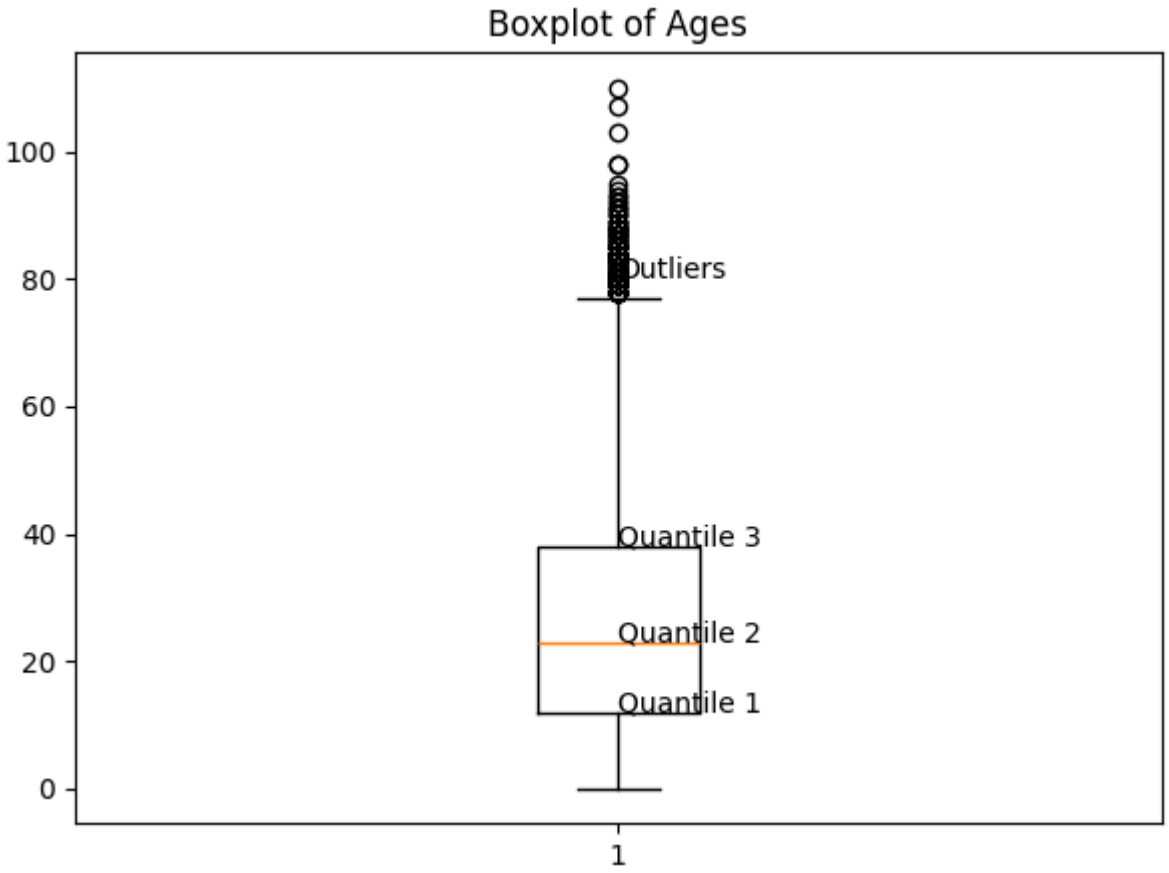
3. **Whiskers**: Whiskers extend from the box to the extreme values within a specific range. Commonly, mustaches account for 1.5 times the IQR. Values outside this range can be considered outliers.

Box plots are useful for identifying centrality, dispersion, and the presence of outliers in a data set in a visual and concise manner. In addition, they allow the distributions of different data groups to be compared effectively.

In [ ]:
```
# Calculate quartiles
median = df['EDAD_ANOS'].median()
q1 = df['EDAD_ANOS'].quantile(0.25)
q2 = df['EDAD_ANOS'].quantile(0.50)
q3 = df['EDAD_ANOS'].quantile(0.75)
```

```
print("Median:", median)
print("Quantile 1:", q1)
print("Quantile 2:", q2)
print("Quantile 3:", q3)


# Filter outliers
outliers = df[df['EDAD_ANOS'] >= 80]['EDAD_ANOS']
# Count the outliers
num_outliers = outliers.count()
# Show the number of outliers
print("Amound of outliers (EDAD_ANOS >= 80):", num_outliers)

# Create the boxplot
plt.figure(figsize=(7, 5))
plt.boxplot(df['EDAD_ANOS'], vert=True)
# Add labels to each quartile
plt.text(1, q1, 'Quantile 1')
plt.text(1, q2, 'Quantile 2')
plt.text(1, q3, 'Quantile 3')
# Add labels to outliers
plt.text(1, 80, 'Outliers')
plt.title('Boxplot of Ages')

plt.show()
```

```
Median: 23.0
Quantile 1: 12.0
Quantile 2: 23.0
Quantile 3: 38.0
Amound of outliers (EDAD_ANOS >= 80): 114
```



Conclusions obtained during the analysis

In the previous dataset, the information collected on Dengue disease was displayed. In this, there are two types, classic and hemorrhagic. In the case of Mexico, it was highly attacked by this virus and large numbers of cases were found in different people of different types of illnesses and ages.

It has been observed within the analysis of the data set that hemorrhagic cases are fewer than the classic ones, in addition, it is worth mentioning that these types of cases are mostly in adults, and are applied to people with pathological diseases. This disease attacked millions of people and to this day there are still cases of such disease

Findings

- There more number of cases Dengue on women than men.
- No are many cases on Older People(60+), the most cases are on Adults(18+), then on Children(-12) and then Adolescents(13+ / -17).
- There more Dengue Classic cases than Hemorrhagic Dengue, but Hemorrhagic Dengue it's more deadly.
- Most of those infected were adult men.
- People with diabetes were part of the large number of infected.
- Within the dengue hemorrhagic fever rate, 3.9% died.

- The largest number of cases are located in rocky and humid areas, the state with the highest rate of cases was "Guerrero"