PH Dengue Cases (2016-2021)

April 1, 2024

1 Philippine Dengue Cases from 2016 - 2021

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

1.0.1 Read the file analyze its data

[2]:
```
# Data from https://data.humdata.org/dataset/philippine-dengue-cases-and-deaths
dengue_data = pd.read_csv('ph_dengue_2016-2021.csv')
dengue_data.head()
```

[2]:

	Location	Cases	Deaths	Date	Region
0	ALBAY	15	0.0	01/10/2016	REGION V-BICOL REGION
1	ALBAY	13	0.0	1/17/2016	REGION V-BICOL REGION
2	ALBAY	9	0.0	1/24/2016	REGION V-BICOL REGION
3	ALBAY	14	0.0	1/31/2016	REGION V-BICOL REGION
4	ALBAY	9	0.0	02/07/2016	REGION V-BICOL REGION

[3]:
```
dengue_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32701 entries, 0 to 32700
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Location  32701 non-null  object
 1   Cases     32701 non-null  int64
 2   Deaths    32700 non-null  float64
 3   Date      32701 non-null  object
 4   Region    32701 non-null  object
dtypes: float64(1), int64(1), object(3)
memory usage: 1.2+ MB
```

1.0.2 Data Cleaning

[4]:
```
# Check if there is null value
dengue_data.isnull().sum()
```

[4]:
```
Location    0
Cases       0
Deaths      1
Date        0
Region      0
dtype: int64
```

[5]:
```
# Check the NaN value
null_data = dengue_data[dengue_data.isna().any(axis=1)]
null_data.head()
```

[5]:

	Location	Cases	Deaths	Date	Region
12774	QUEZON CITY	48	NaN	05/07/2017	NATIONAL CAPITAL REGION

[6]:
```
# Drop the null value and store it in a variable
dengue_data = dengue_data.dropna()
dengue_data.head()
```

[6]:

	Location	Cases	Deaths	Date	Region
0	ALBAY	15	0.0	01/10/2016	REGION V-BICOL REGION
1	ALBAY	13	0.0	1/17/2016	REGION V-BICOL REGION
2	ALBAY	9	0.0	1/24/2016	REGION V-BICOL REGION
3	ALBAY	14	0.0	1/31/2016	REGION V-BICOL REGION
4	ALBAY	9	0.0	02/07/2016	REGION V-BICOL REGION

[7]:
```
# Check if there is a data duplication
dengue_data.duplicated().sum()
```

[7]:
```
0
```

1.0.3 Convert columns to correct data type

[8]:
```
# Convert Deaths column into int
dengue_data['Deaths'] = dengue_data['Deaths'].astype(int)
```

[9]:
```
# Convert Date column into datetime
dengue_data['Date'] = pd.to_datetime(dengue_data['Date'])
```

[10]:
```
dengue_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 32700 entries, 0 to 32700
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype
```

```
---  ------    --------------  -----
 0   Location  32700 non-null  object
 1   Cases     32700 non-null  int64
 2   Deaths    32700 non-null  int32
 3   Date      32700 non-null  datetime64[ns]
 4   Region    32700 non-null  object
dtypes: datetime64[ns](1), int32(1), int64(1), object(2)
memory usage: 1.4+ MB
```

1.0.4 Augement data with additional columns

[11]:
```
# Add Year and Month Column
dengue_data['Year'] = dengue_data['Date'].dt.year
dengue_data['Month'] = dengue_data['Date'].dt.month
```

[12]:
```
dengue_data.head()
```

[12]:

	Location	Cases	Deaths	Date	Region	Year	Month
0	ALBAY	15	0	2016-01-10	REGION V-BICOL REGION	2016	1
1	ALBAY	13	0	2016-01-17	REGION V-BICOL REGION	2016	1
2	ALBAY	9	0	2016-01-24	REGION V-BICOL REGION	2016	1
3	ALBAY	14	0	2016-01-31	REGION V-BICOL REGION	2016	1
4	ALBAY	9	0	2016-02-07	REGION V-BICOL REGION	2016	2

1.0.5 1. What is the annual pattern of dengue infections, and in which year were the highest number of dengue cases recorded? In which year were the highest number of deaths recorder?

[13]:
```
# Use the groupby function to select the Year data
dengue_per_year = dengue_data.groupby('Year').sum(numeric_only = True)
```

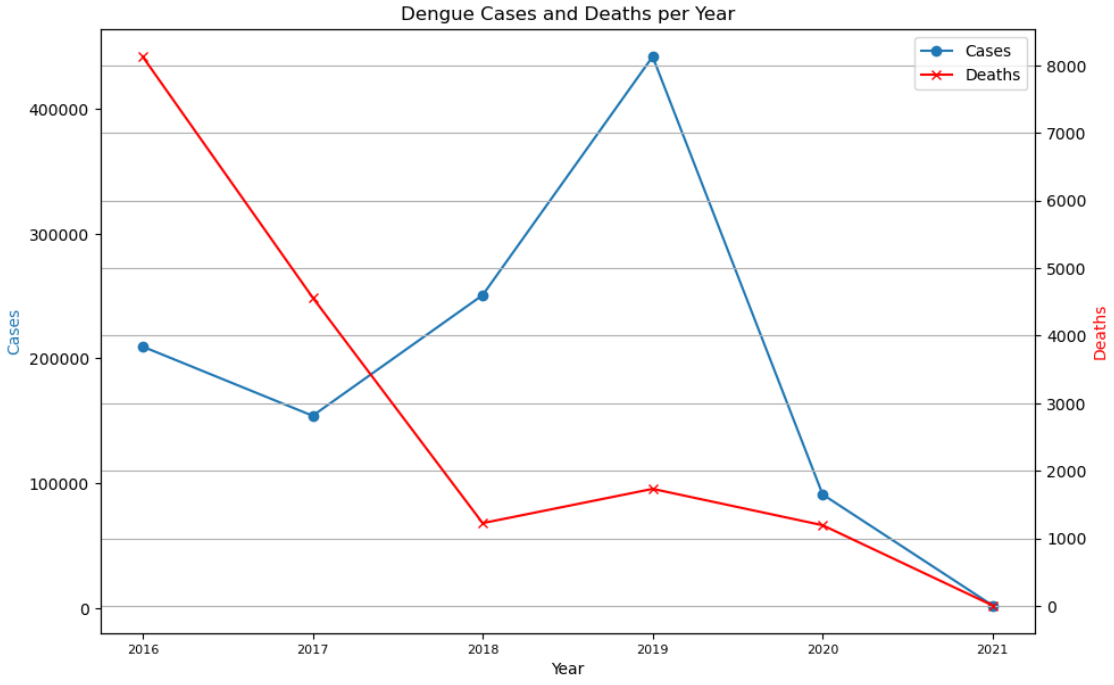
[14]:
```
dengue_per_year
```

[14]:

Year	Cases	Deaths	Month
2016	209544	8127	41244
2017	154107	4563	43139
2018	250783	1226	43344
2019	441902	1733	42966
2020	91041	1195	42208
2021	2087	8	252

[15]:
```
# Plot figure size
fig, ax1 = plt.subplots(figsize=(10, 6))

ax2 = ax1.twinx()

# X values
```

```
years = range(2016,2022)

# Plot Cases on the first y-axis (left)
ax1.plot(years, dengue_per_year['Cases'], color='#1f77b4', marker='o',␣
 ↪label='Cases')
ax1.set_xlabel('Year')
ax1.set_ylabel('Cases', color='#1f77b4')
ax1.set_xticks(years)
ax1.set_xticklabels(years, size=8)

# Plot Deaths on the second y-axis (right)
ax2.plot(years, dengue_per_year['Deaths'], color='r', marker='x',␣
 ↪label='Deaths')
ax2.set_ylabel('Deaths', color='r')

# Combine both datasets' legends
lines, labels = ax1.get_legend_handles_labels()
lines2, labels2 = ax2.get_legend_handles_labels()
ax2.legend(lines + lines2, labels + labels2, loc=0)

# Adjust layout
plt.tight_layout()

# Show the plot
plt.title('Dengue Cases and Deaths per Year')
plt.grid(True)
plt.show()
```



1.0.6 From the graph, it is evident that the year 2019 had the highest overall number of cases, whereas 2021 exhibited the lowest count. Also, it is evident that the year 2016 had the highest overall number of deaths, whereas 2021 exhibited the lowest count

1.0.7 2. During which month do dengue cases and deaths typically peak in terms of their frequency?

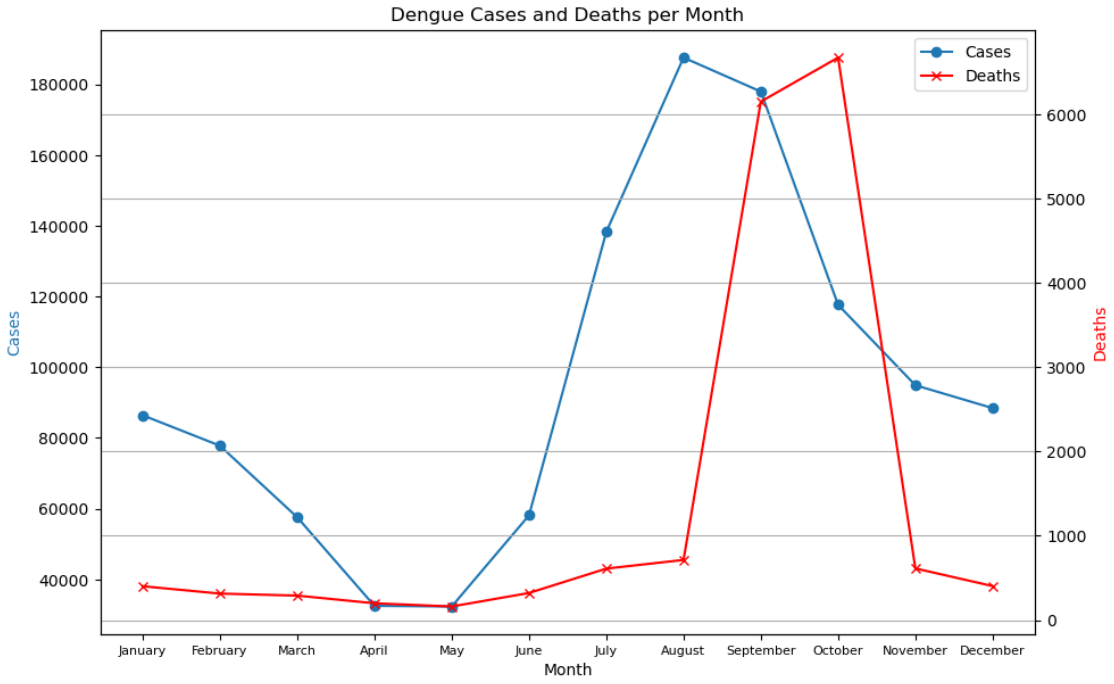
[16]:
```
# Use the groupby function to select the Month data
dengue_per_month = dengue_data.groupby('Month').sum(numeric_only = True)
dengue_per_month
```

[16]:

Month	Cases	Deaths	Year
1	86415	402	5584319
2	77801	315	5077296
3	57576	291	5586210
4	32508	200	5585706
5	32339	162	5581799
6	58110	322	5331690
7	138242	611	5837706
8	187554	714	5331816
9	177943	6148	5585958
10	117645	6670	5583438
11	94900	613	5063463
12	88431	404	5840100

[17]:
```
# Plot figure size
fig, ax1 = plt.subplots(figsize=(10, 6))

ax2 = ax1.twinx()

# X values
months = [
    "January", "February", "March", "April",
    "May", "June", "July", "August",
    "September", "October", "November", "December"]

# Plot Cases on the first y-axis (left)
ax1.plot(months, dengue_per_month['Cases'], color='#1f77b4', marker='o',␣
 ↪label='Cases')
ax1.set_xlabel('Month')
ax1.set_ylabel('Cases', color='#1f77b4')
ax1.set_xticks(months)
ax1.set_xticklabels(months, size=8)
```

```
# Plot Deaths on the second y-axis (right)
ax2.plot(months, dengue_per_month['Deaths'], color='r', marker='x',␣
 ↪label='Deaths')
ax2.set_ylabel('Deaths', color='r')

# Combine both datasets' legends
lines, labels = ax1.get_legend_handles_labels()
lines2, labels2 = ax2.get_legend_handles_labels()
ax2.legend(lines + lines2, labels + labels2, loc=0)

# Adjust layout
plt.tight_layout()

# Show the plot
plt.title('Dengue Cases and Deaths per Month')
plt.grid(True)
plt.show()
```



1.0.8 From the graph, it is evident that the month of August had the highest overall number of cases, whereas month of April and May has the lowest count. Also, it is evident that the month of October had the highest overall number of deaths, whereas month of April and May has the lowest count

1.0.9 3. What is the total number of cases per region, and what region accumulates the highest number of dengue cases and deaths?

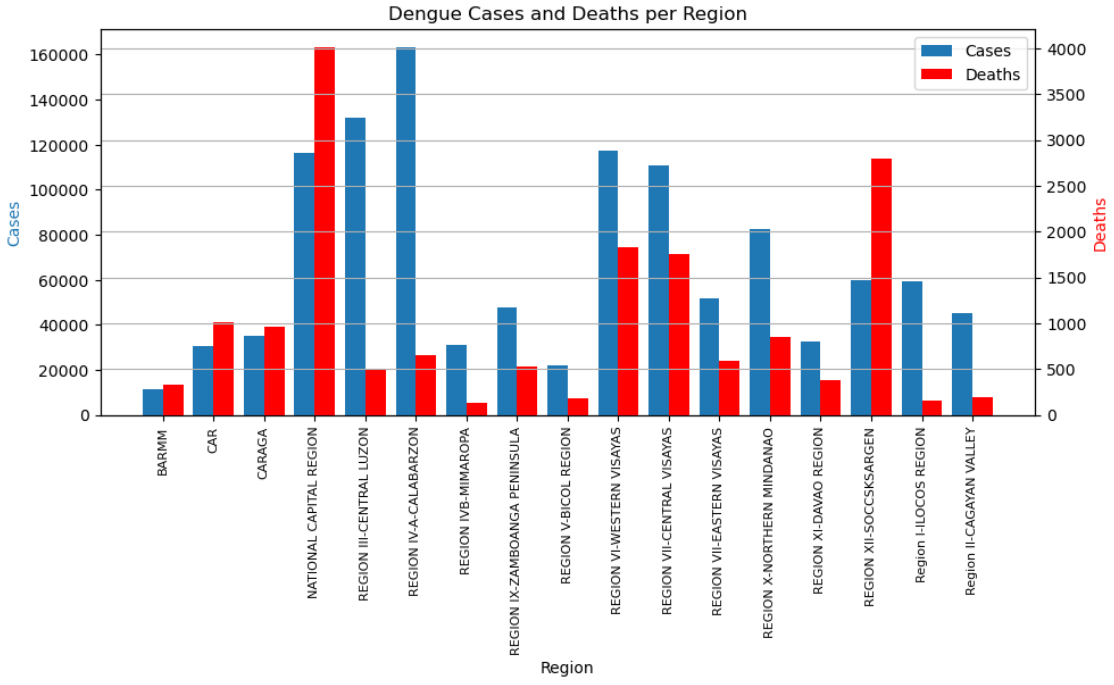
[18]:
```
# Use the groupby function to select the Region data
dengue_per_region = dengue_data.groupby('Region').sum(numeric_only = True)
dengue_per_region
```

[18]:

Region	Cases	Deaths	Year	Month
BARMM	11540	332	2613340	8420
CAR	30608	1008	3658676	11788
CARAGA	35305	966	5749348	18524
NATIONAL CAPITAL REGION	116099	4009	8883339	28623
REGION III-CENTRAL LUZON	132158	485	4740336	15354
REGION IV-A-CALABARZON	163177	653	3160224	10236
REGION IVB-MIMAROPA	30887	131	3160224	10236
REGION IX-ZAMBOANGA PENINSULA	47794	532	2623435	8475
REGION V-BICOL REGION	22195	185	3686928	11942
REGION VI-WESTERN VISAYAS	117563	1825	4197496	13560
REGION VII-CENTRAL VISAYAS	110726	1760	3672809	11865
REGION VII-EASTERN VISAYAS	51990	585	4197496	13560
REGION X-NORTHERN MINDANAO	82528	848	3672809	11865
REGION XI-DAVAO REGION	32564	385	3043289	9835
REGION XII-SOCCSKSARGEN	59860	2798	3136008	10104
Region I-ILOCOS REGION	59312	157	2633520	8530
Region II-CAGAYAN VALLEY	45158	193	3160224	10236

[19]:
```
# X values
region_group = dengue_data.groupby('Region')
region_cases = region_group.sum(numeric_only = True)['Cases']
region_group = dengue_data.groupby('Region')
region_deaths = region_group.sum(numeric_only = True)['Deaths']

# Plot figure size
fig, ax1 = plt.subplots(figsize=(10, 6))

ax2 = ax1.twinx()

# Bar width
bar_width = 0.4

# Create positions for the bars
keys = [pair for pair, df in region_group]
```

```
position1 = range(len(keys))
position2 = [pos + bar_width for pos in position1]

# Plot Cases on the first y-axis (left)
ax1.bar(position1, region_cases, width=bar_width, color='#1f77b4',␣
 ↪label='Cases')
ax1.set_xlabel('Region')
ax1.set_ylabel('Cases', color='#1f77b4')
ax1.set_xticks([pos + bar_width / 2 for pos in position1])
ax1.set_xticklabels(keys, rotation='vertical', size=8)

# Plot Deaths on the second y-axis (right)
ax2.bar(position2, region_deaths, width=bar_width, color='r', label='Deaths')
ax2.set_ylabel('Deaths', color='r')

# Combine both datasets' legends
lines, labels = ax1.get_legend_handles_labels()
lines2, labels2 = ax2.get_legend_handles_labels()
ax2.legend(lines + lines2, labels + labels2, loc=0)

# Adjust layout
plt.tight_layout()

# Show the plot
plt.title('Dengue Cases and Deaths per Region')
plt.grid(True)
plt.show()
```



1.0.10 From the graph, it is evident that the CALABARZON Region had the highest overall number of cases, whereas BARMM Region has the lowest count. Also, it is evident that the NCR Region had the highest overall number of deaths, whereas MIMAROPA Region has the lowest count.

1.0.11 4. What is the total number of cases per location, and what location accumulates the highest number of dengue cases and deaths?

[20]:
```
# Use the groupby function to select the Location data
dengue_per_location = dengue_data.groupby('Location').sum(numeric_only = True)
dengue_per_location
```

[20]:

Location	Cases	Deaths	Year	Month
CAGAYAN DE ORO CITY	14211	170	524687	1695
NEGROS ORIENTAL	20581	284	524687	1695
ABRA	3177	60	522668	1684
AGUSAN DEL NORTE	2822	97	522668	1684
AGUSAN DEL SUR	6530	112	522668	1684
…	…	…	…	…
ZAMBALES	6466	17	526704	1706
ZAMBOANGA CITY	18009	169	524687	1695
ZAMBOANGA DEL NORTE	10290	128	524687	1695
ZAMBOANGA DEL SUR	8472	126	524687	1695
ZAMBOANGA SIBUGAY	8410	74	524687	1695

[126 rows x 4 columns]

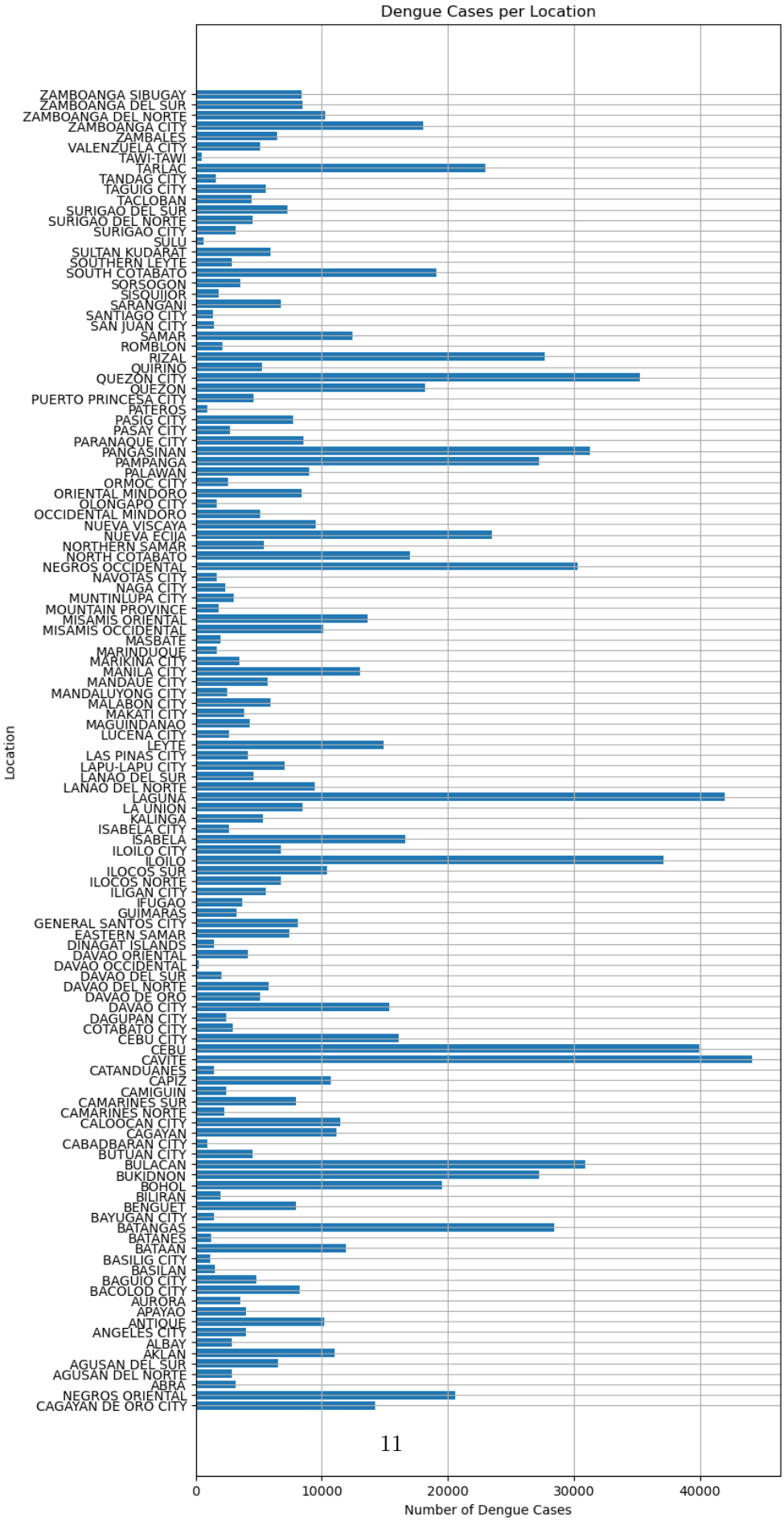
[21]:
```
# To find the highest and lowest data without using data visualization
dengue_sort = dengue_per_location.sort_values(by='Cases', ascending=False)
dengue_sort
```

[21]:

Location	Cases	Deaths	Year	Month
CAVITE	44169	170	526704	1706
LAGUNA	42013	108	526704	1706
CEBU	39957	688	524687	1695
ILOILO	37155	287	524687	1695
QUEZON CITY	35232	1208	520651	1679
…	…	…	…	…
PATEROS	926	29	522668	1684
CABADBARAN CITY	908	27	522668	1684
SULU	642	33	522668	1684
TAWI-TAWI	460	5	522668	1684
DAVAO OCCIDENTAL	207	8	419854	1360

```
[126 rows x 4 columns]
```

[22]:
```python
# Y values
loc_group = dengue_data.groupby('Location')
loc_cases = loc_group.sum(numeric_only = True)['Cases']

# Create and Display the plot using horizontal bar chart
plt.figure(figsize=(8, 20))
keys = [pair for pair, df in loc_group]
plt.barh(keys, loc_cases)
plt.title('Dengue Cases per Location')
plt.xlabel('Number of Dengue Cases')
plt.ylabel('Location')
plt.grid(True)
plt.show()
```


Dengue Cases per Location
Location
ZAMBOANGA SIBUGAY
ZAMBOANGA DEL SUR
ZAMBOANGA DEL NORTE
ZAMBOANGA CITY
ZAMBALES
VALENZUELA CITY
TAWI-TAWI
TARLAC
TANDAG CITY
TAGUIG CITY
TACLOBAN
SURIGAO DEL SUR
SURIGAO DEL NORTE
SURIGAO CITY
SULU
SULTAN KUDARAT
SOUTHERN LEYTE
SOUTH COTABATO
SORSOGON
SISQUIJOR
SARANGANI
SANTIAGO CITY
SAN JUAN CITY
SAMAR
ROMBLON
RIZAL
QUIRINO
QUEZON CITY
QUEZON
PUERTO PRINCESA CITY
PATEROS
PASIG CITY
PASAY CITY
PARANAQUE CITY
PANGASINAN
PAMPANGA
PALAWAN
ORMOC CITY
ORIENTAL MINDORO
OLONGAPO CITY
OCCIDENTAL MINDORO
NUEVA VISCAYA
NUEVA ECIJA
NORTHERN SAMAR
NORTH COTABATO
NEGROS OCCIDENTAL
NAVOTAS CITY
NAGA CITY
MUNTINLUPA CITY
MOUNTAIN PROVINCE
MISAMIS ORIENTAL
MISAMIS OCCIDENTAL
MASBATE
MARINDUQUE
MARIKINA CITY
MANILA CITY
MANDAUE CITY
MANDALUYONG CITY
MALABON CITY
MAKATI CITY
MAGUINDANAO
LUCENA CITY
LEYTE
LAS PINAS CITY
LAPU-LAPU CITY
LANAO DEL SUR
LANAO DEL NORTE
LAGUNA
LA UNION
KALINGA
ISABELA CITY
ISABELA
ILOILO CITY
ILOILO
ILOCOS SUR
ILOCOS NORTE
ILIGAN CITY
IFUGAO
GUIMARAS
GENERAL SANTOS CITY
EASTERN SAMAR
DINAGAT ISLANDS
DAVAO ORIENTAL
DAVAO OCCIDENTAL
DAVAO DEL SUR
DAVAO DEL NORTE
DAVAO DE ORO
DAVAO CITY
DAGUPAN CITY
COTABATO CITY
CEBU CITY
CEBU
CAVITE
CATANDUANES
CAPIZ
CAMIGUIN
CAMARINES SUR
CAMARINES NORTE
CALOOCAN CITY
CAGAYAN
CABADBARAN CITY
BUTUAN CITY
BULACAN
BUKIDNON
BOHOL
BILIRAN
BENGUET
BAYUGAN CITY
BATANGAS
BATANES
BATAAN
BASILIG CITY
BASILAN
BAGUIO CITY
BACOLOD CITY
AURORA
APAYAO
ANTIQUE
ANGELES CITY
ALBAY
AKLAN
AGUSAN DEL SUR
AGUSAN DEL NORTE
ABRA
NEGROS ORIENTAL
CAGAYAN DE ORO CITY
0
10000
20000
30000
40000
Number of Dengue Cases

1.0.12 From the graph, it is evident that the Cavite had the highest overall number of cases, whereas Davao Occidental has the lowest count.

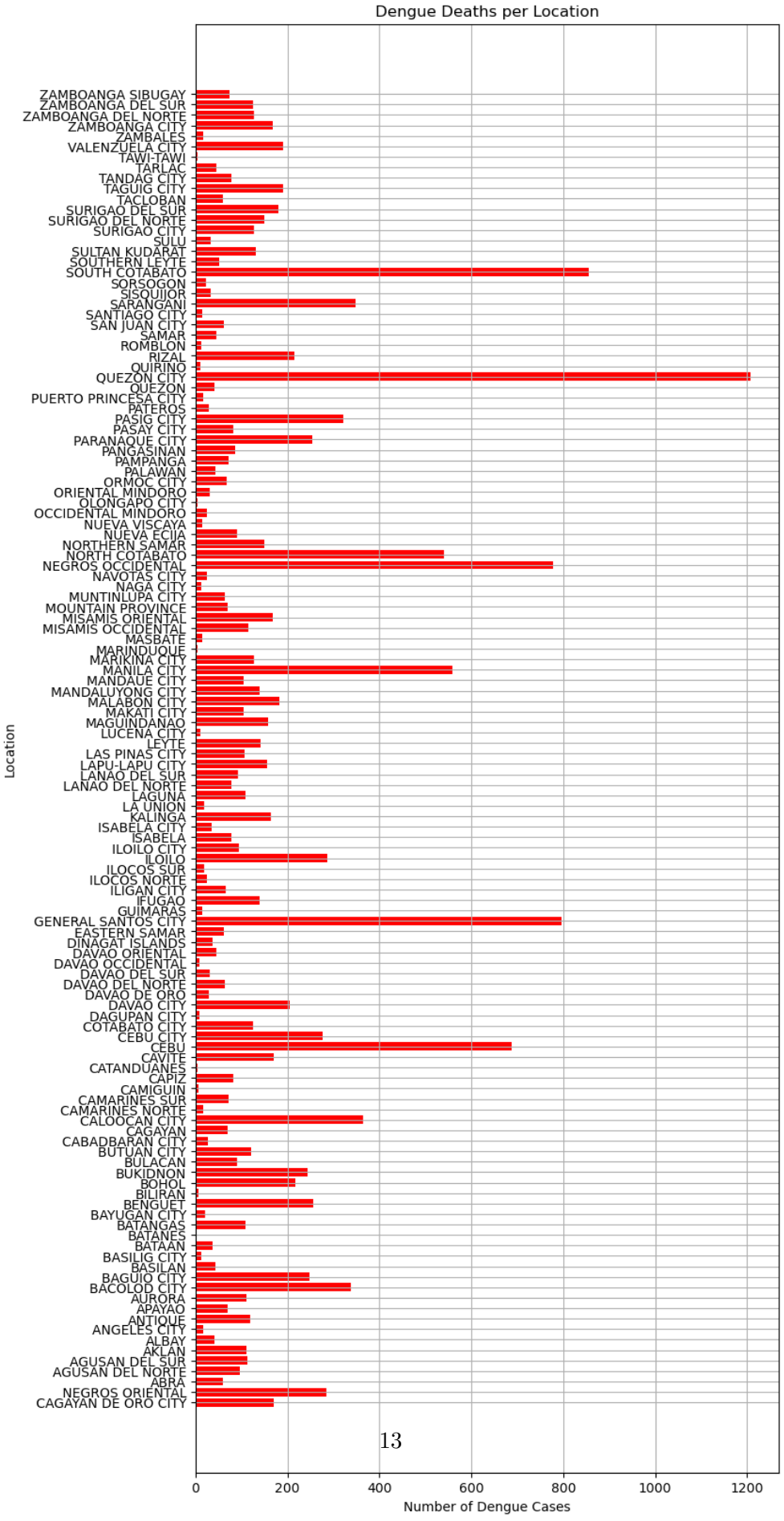
[23]:
```
# To find the highest and lowest data without using data visualization
dengue_sort = dengue_per_location.sort_values(by='Deaths', ascending=False)
dengue_sort
```

[23]:

Location	Cases	Deaths	Year	Month
QUEZON CITY	35232	1208	520651	1679
SOUTH COTABATO	19114	855	522668	1684
GENERAL SANTOS CITY	8106	797	522668	1684
NEGROS OCCIDENTAL	30323	778	524687	1695
CEBU	39957	688	524687	1695
…	…	…	…	…
TAWI-TAWI	460	5	522668	1684
OLONGAPO CITY	1652	5	526704	1706
MARINDUQUE	1696	4	526704	1706
CATANDUANES	1412	4	526704	1706
BATANES	1186	3	526704	1706

[126 rows x 4 columns]

[24]:
```
# Y values
loc_group = dengue_data.groupby('Location')
loc_deaths = loc_group.sum(numeric_only = True)['Deaths']

# Create and Display the plot using horizontal bar chart
plt.figure(figsize=(8, 20))
keys = [pair for pair, df in loc_group]
plt.barh(keys, loc_deaths, color = 'r')
plt.title('Dengue Deaths per Location')
plt.xlabel('Number of Dengue Cases')
plt.ylabel('Location')
plt.grid(True)
plt.show()
```


Dengue Deaths per Location
Location
Number of Dengue Cases
0
200
400
600
800
1000
1200
ZAMBOANGA SIBUGAY
ZAMBOANGA DEL SUR
ZAMBOANGA DEL NORTE
ZAMBOANGA CITY
ZAMBALES
VALENZUELA CITY
TAWI-TAWI
TARLAC
TANDAG CITY
TAGUIG CITY
TACLOBAN
SURIGAO DEL SUR
SURIGAO DEL NORTE
SURIGAO CITY
SULU
SULTAN KUDARAT
SOUTHERN LEYTE
SOUTH COTABATO
SORSOGON
SISQUIJOR
SARANGANI
SANTIAGO CITY
SAN JUAN CITY
SAMAR
ROMBLON
RIZAL
QUIRINO
QUEZON CITY
QUEZON
PUERTO PRINCESA CITY
PATEROS
PASIG CITY
PASAY CITY
PARANAQUE CITY
PANGASINAN
PAMPANGA
PALAWAN
ORMOC CITY
ORIENTAL MINDORO
OLONGAPO CITY
OCCIDENTAL MINDORO
NUEVA VISCAYA
NUEVA ECIJA
NORTHERN SAMAR
NORTH COTABATO
NEGROS OCCIDENTAL
NAVOTAS CITY
NAGA CITY
MUNTINLUPA CITY
MOUNTAIN PROVINCE
MISAMIS ORIENTAL
MISAMIS OCCIDENTAL
MASBATE
MARINDUQUE
MARIKINA CITY
MANILA CITY
MANDAUE CITY
MANDALUYONG CITY
MALABON CITY
MAKATI CITY
MAGUINDANAO
LUCENA CITY
LEYTE
LAS PINAS CITY
LAPU-LAPU CITY
LANAO DEL SUR
LANAO DEL NORTE
LAGUNA
LA UNION
KALINGA
ISABELA CITY
ISABELA
ILOILO CITY
ILOILO
ILOCOS SUR
ILOCOS NORTE
ILIGAN CITY
IFUGAO
GUIMARAS
GENERAL SANTOS CITY
EASTERN SAMAR
DINAGAT ISLANDS
DAVAO ORIENTAL
DAVAO OCCIDENTAL
DAVAO DEL SUR
DAVAO DEL NORTE
DAVAO DE ORO
DAVAO CITY
DAGUPAN CITY
COTABATO CITY
CEBU CITY
CEBU
CAVITE
CATANDUANES
CAPIZ
CAMIGUIN
CAMARINES SUR
CAMARINES NORTE
CALOOCAN CITY
CAGAYAN
CABADBARAN CITY
BUTUAN CITY
BULACAN
BUKIDNON
BOHOL
BILIRAN
BENGUET
BAYUGAN CITY
BATANGAS
BATANES
BATAAN
BASILIG CITY
BASILAN
BAGUIO CITY
BACOLOD CITY
AURORA
APAYAO
ANTIQUE
ANGELES CITY
ALBAY
AKLAN
AGUSAN DEL SUR
AGUSAN DEL NORTE
ABRA
NEGROS ORIENTAL
CAGAYAN DE ORO CITY

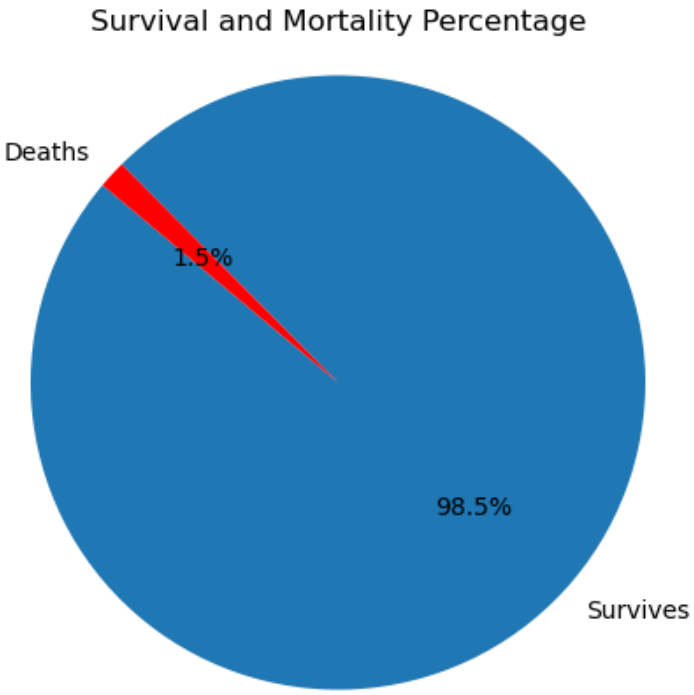
1.0.13 From the graph, it is evident that the Quezon City had the highest overall number of deaths, whereas Batanes has the lowest count.

1.0.14 5. What is the percentage of people who died and survived?

[25]:
```
# Get the total number of cases and deaths using sum()
total_cases = dengue_data['Cases'].sum()
total_deaths = dengue_data['Deaths'].sum()
# Subtract number of deaths from number of cases to get the total survives
total_survives = total_cases - total_deaths
# Display the result
print(f'Total cases are {total_cases}')
print(f'Total deaths are {total_deaths}')
print(f'Total survives are {total_survives}')
```

```
Total cases are 1149464
Total deaths are 16852
Total survives are 1132612
```

[26]:
```
# Create and Display the plot using Pie Graph
labels = ['Survives','Deaths']
sizes = [total_survives,total_deaths]
color = ['#1f77b4','r']
plt.figure(figsize=(8, 5))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140,colors = color)
plt.axis('equal')
plt.title("Survival and Mortality Percentage")
plt.show()
```



1.0.15 From the graph, the percentage of people who survived is 98.5% while the people who died is 1.50%

[ ]: