In [2]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:

```python
df = pd.read_csv('pmc_1.csv')
df
```

Out[3]:

	Unnamed: 0	dailysamples	dailyconfirmed	dailyrecovered	dailydeceased	totalcritical
0	1	153.0	15.0	0.0	2.0	NaN
1	2	115.0	33.0	1.0	2.0	4.0
2	3	157.0	44.0	1.0	4.0	6.0
3	4	220.0	55.0	1.0	4.0	5.0
4	5	265.0	65.0	0.0	5.0	7.0
...	...	...	...	...	...	...
748	752	NaN	69.0	NaN	0.0	NaN
749	753	NaN	57.0	NaN	0.0	NaN
750	754	NaN	30.0	NaN	0.0	NaN
751	755	NaN	31.0	NaN	0.0	NaN
752	756	NaN	13.0	NaN	0.0	NaN

753 rows × 19 columns

In [4]:

```python
df.head()
```

Out[4]:

	Unnamed: 0	dailysamples	dailyconfirmed	dailyrecovered	dailydeceased	totalcritical	ve
0	1	153.0	15.0	0.0	2.0	NaN	
1	2	115.0	33.0	1.0	2.0	4.0	
2	3	157.0	44.0	1.0	4.0	6.0	
3	4	220.0	55.0	1.0	4.0	5.0	
4	5	265.0	65.0	0.0	5.0	7.0	

In [5]:

```python
df.describe()
```

Out[5]:

	Unnamed: 0	dailysamples	dailyconfirmed	dailyrecovered	dailydeceased	totalcritica
count	753.000000	723.000000	751.000000	722.000000	751.000000	710.00000
mean	377.111554	6302.919779	872.615180	894.757618	12.426099	398.85211
std	217.703483	4992.899174	1392.477493	1389.869347	15.518404	355.43056
min	1.000000	106.000000	4.000000	0.000000	0.000000	0.00000
25%	189.000000	3106.000000	129.000000	149.250000	2.000000	160.00000
50%	377.000000	5239.000000	273.000000	293.500000	6.000000	234.00000
75%	565.000000	7451.000000	934.000000	968.500000	17.000000	583.25000
max	756.000000	27986.000000	8301.000000	8215.000000	67.000000	1415.00000

In [6]:

```
print("Missing values in the dataset:")
df = df.drop(columns=['Date'])
print(df.isnull().sum())
```

```
Missing values in the dataset:
Unnamed: 0              0
dailysamples           30
dailyconfirmed          2
dailyrecovered         31
dailydeceased           2
totalcritical          43
ventilatorpatients    414
totalsamples           30
totalconfirmed          2
totalhospital          31
totalrecovered         30
totaldeceased           4
totalhousesurvey      605
populationcovered     605
housescovered         605
flu                   615
active_hosp           188
active_home           191
dtype: int64
```

In [7]:

```
df.fillna(df.mean(), inplace=True)
```

In [8]:

```
selected_attributes = ['dailysamples', 'dailyconfirmed', 'dailyrecovered', 'dailyde

print("Sum of selected attributes:")
print(df[selected_attributes].sum())

print("Mean of selected attributes:")
print(df[selected_attributes].mean())

print("Median of selected attributes:")
print(df[selected_attributes].median())

print("Standard Deviation of selected attributes:")
print(df[selected_attributes].std())
```

```
Sum of selected attributes:
dailysamples      4.746099e+06
dailyconfirmed    6.570792e+05
dailyrecovered    6.737525e+05
dailydeceased     9.356852e+03
totalcritical     3.003356e+05
dtype: float64
Mean of selected attributes:
dailysamples      6302.919779
dailyconfirmed     872.615180
dailyrecovered     894.757618
dailydeceased       12.426099
totalcritical      398.852113
dtype: float64
Median of selected attributes:
dailysamples      5488.0
dailyconfirmed     274.0
dailyrecovered     317.0
dailydeceased        6.0
totalcritical      273.0
dtype: float64
Standard Deviation of selected attributes:
dailysamples      4892.293174
dailyconfirmed    1390.624561
dailyrecovered    1360.920290
dailydeceased       15.497754
totalcritical      345.119081
dtype: float64
```

```
In [9]: df
```

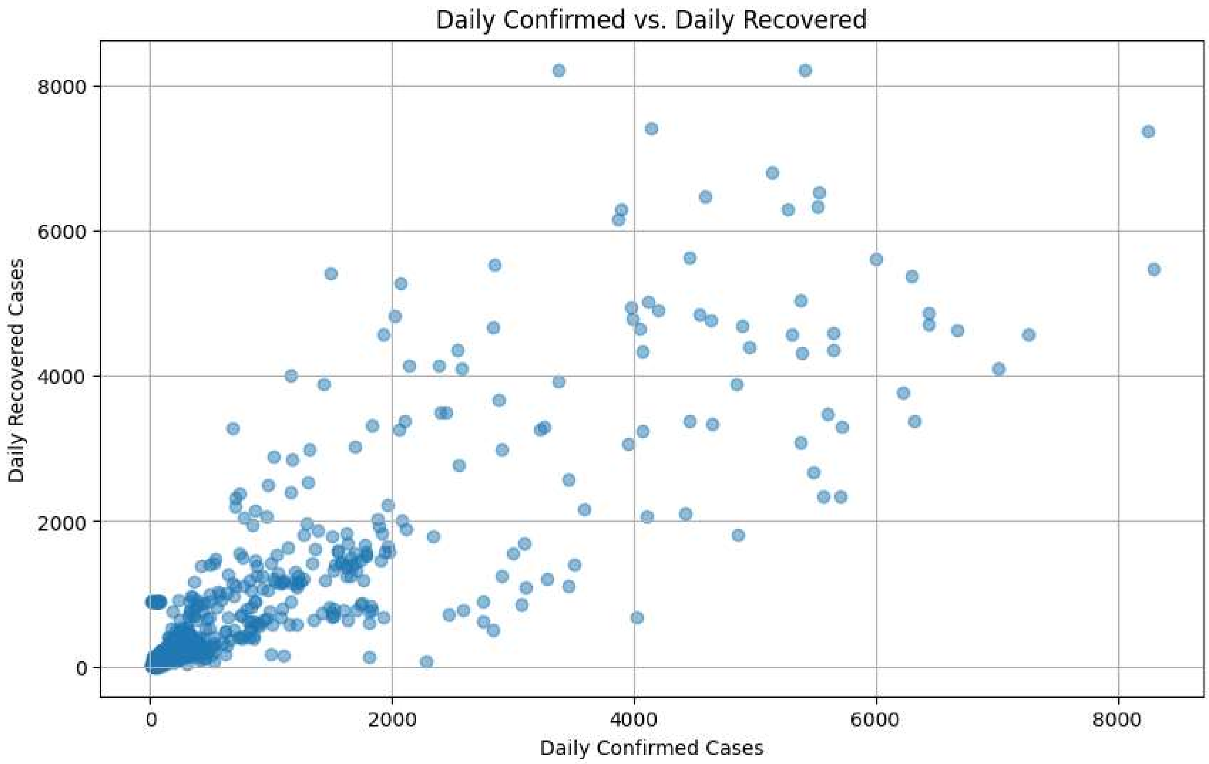
Out[9]:

	Unnamed: 0	dailysamples	dailyconfirmed	dailyrecovered	dailydeceased	totalcritical
0	1	153.000000	15.0	0.000000	2.0	398.852113
1	2	115.000000	33.0	1.000000	2.0	4.000000
2	3	157.000000	44.0	1.000000	4.0	6.000000
3	4	220.000000	55.0	1.000000	4.0	5.000000
4	5	265.000000	65.0	0.000000	5.0	7.000000
...	...	...	...	...	...	...
748	752	6302.919779	69.0	894.757618	0.0	398.852113
749	753	6302.919779	57.0	894.757618	0.0	398.852113
750	754	6302.919779	30.0	894.757618	0.0	398.852113
751	755	6302.919779	31.0	894.757618	0.0	398.852113
752	756	6302.919779	13.0	894.757618	0.0	398.852113

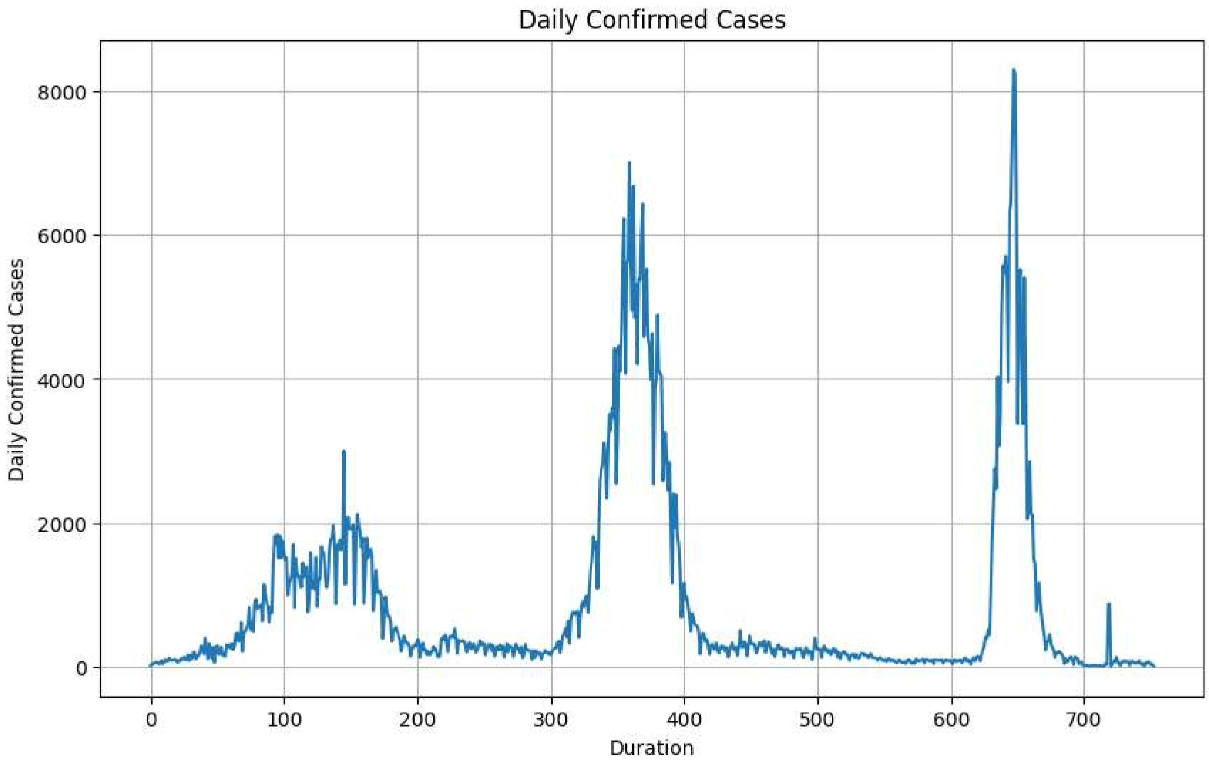
753 rows × 18 columns

In [10]:

```
x = df['dailyconfirmed']
y = df['dailyrecovered']

plt.figure(figsize=(10, 6))
plt.scatter(x, y, alpha=0.5)
plt.title('Daily Confirmed vs. Daily Recovered')
plt.xlabel('Daily Confirmed Cases')
plt.ylabel('Daily Recovered Cases')
plt.grid(True)

plt.show()
```

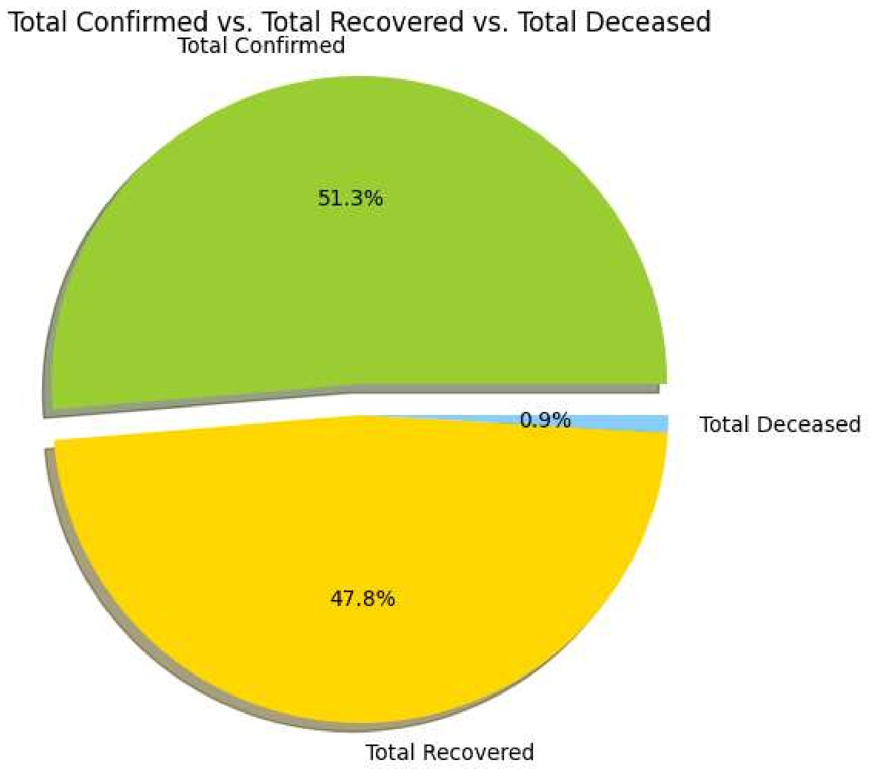


In [11]:
```
# plot daily confirmed cases
plt.figure(figsize=(10, 6))
plt.plot(df['dailyconfirmed'])
plt.title('Daily Confirmed Cases')
plt.xlabel('Duration')
plt.ylabel('Daily Confirmed Cases')
plt.grid(True)

plt.show()
```



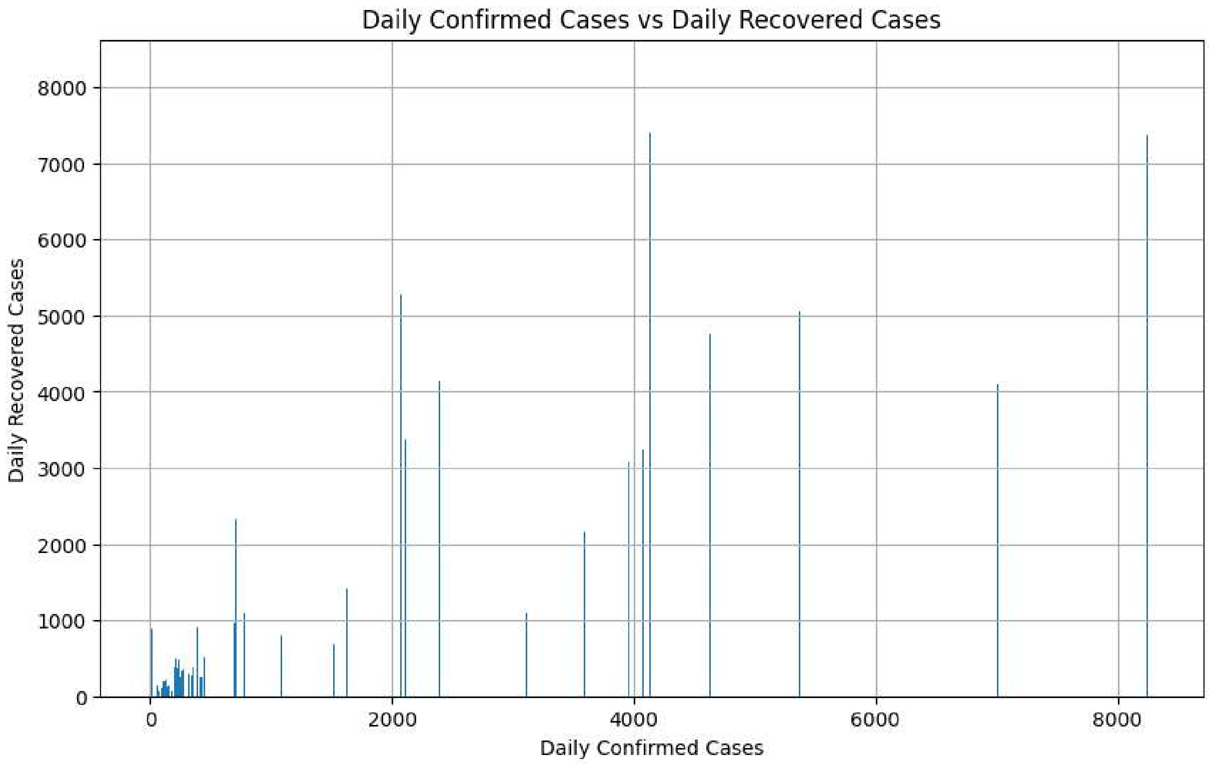
In [16]:

```
# use total confirmed cases and total recovered cases to plot a pie chart
total_confirmed = df['totalconfirmed'].sum()
total_recovered = df['totalrecovered'].sum()
total_deceased = df['totaldeceased'].sum()

labels = ['Total Confirmed', 'Total Recovered', 'Total Deceased']
sizes = [total_confirmed, total_recovered, total_deceased]
colors = ['yellowgreen', 'gold', 'lightskyblue']
explode = (0.1, 0, 0)

plt.figure(figsize=(10, 6))
plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%', sh
plt.axis('equal')
plt.title('Total Confirmed vs. Total Recovered vs. Total Deceased')
plt.show()
```

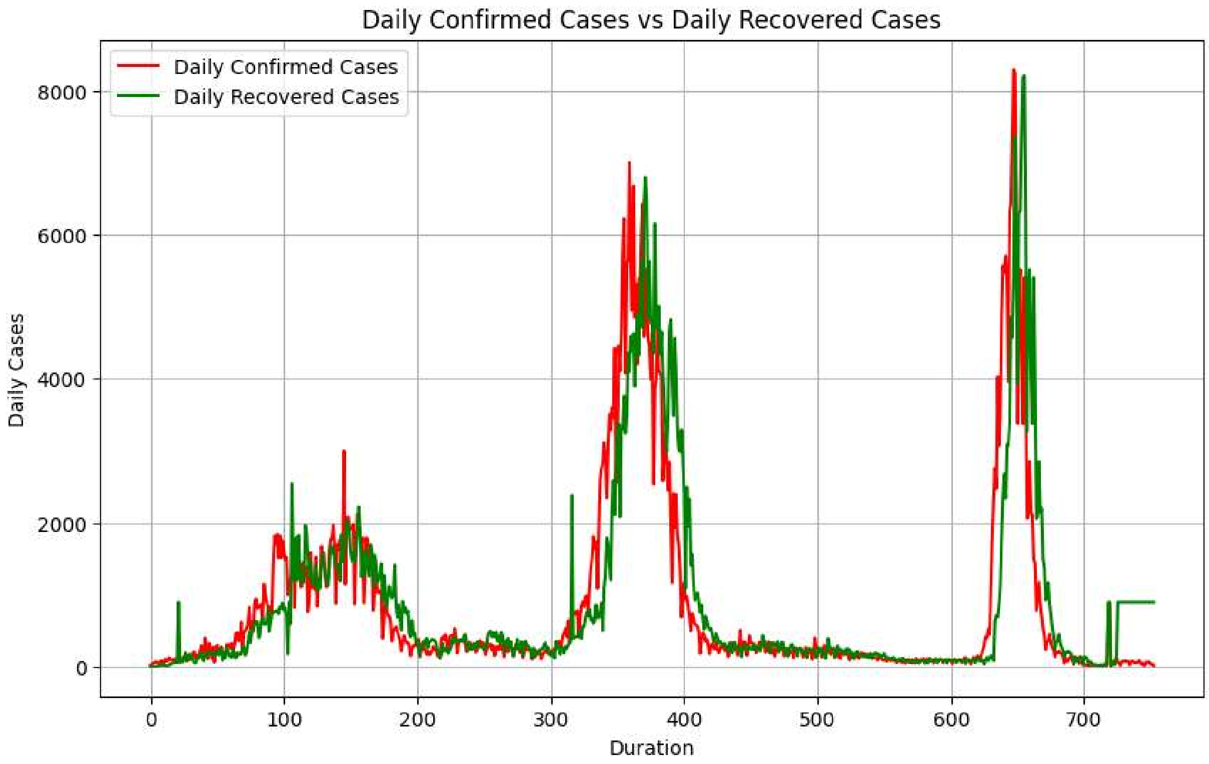


In [12]:

```
# bar plot of daily confirmed cases vs daily recovered cases
plt.figure(figsize=(10, 6))
plt.bar(df['dailyconfirmed'], df['dailyrecovered'])
plt.title('Daily Confirmed Cases vs Daily Recovered Cases')
plt.xlabel('Daily Confirmed Cases')
plt.ylabel('Daily Recovered Cases')
plt.grid(True)

plt.show()
```



In [15]:
```
# compund line plot of daily confirmed cases vs daily recovered cases
plt.figure(figsize=(10, 6))
plt.plot(df['dailyconfirmed'], color='red', label='Daily Confirmed Cases')
plt.plot(df['dailyrecovered'], color='green', label='Daily Recovered Cases')
plt.title('Daily Confirmed Cases vs Daily Recovered Cases')
plt.xlabel('Duration')
plt.ylabel('Daily Cases')
plt.grid(True)
plt.legend()

plt.show()
```


Daily Confirmed Cases vs Daily Recovered Cases
Daily Confirmed Cases
Daily Recovered Cases
Daily Cases
0
2000
4000
6000
8000
Duration
0
100
200
300
400
500
600
700