In [2]:
```python
import pandas as pd
import statistics as stat
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:
```python
base = pd.read_csv('../Data/base_set_joined.csv')

def cumulate_weekly_data_columns(start_index, end_index):
    '''
    Given a start and end column, this function sums up
    each column in between and then groups days by week.
    This function is used to read data from the base set
    containing data from the US.
    :param start_index: The starting columnn.
    :param end_index: The ending column.
    :return: Two lists; one containing weeks and the other
    containing the summation of the columns of that week.
    '''
    # calculate summation of previous week
    prev_sum = 0
    prev_day = base.iloc[:, start_index - 8].sum()
    for i in range(start_index - 7, start_index):
        col = base.iloc[:, i]
        curr_day = col.sum()
        prev_sum += curr_day - prev_day
        prev_day = curr_day

    curr_sum = 0
    days_passed = 0
    weekly_x = []
    weekly_y = []

    # cumulate weekly data
    for i in range(start_index, end_index + 1):
        col = base.iloc[:, i]
        curr_day = col.sum()
        curr_sum += curr_day - prev_day
        prev_day = curr_day
        days_passed += 1
        if days_passed == 7:
            weekly_y.append(curr_sum)
            curr_sum = 0
            days_passed = 0
            begin_week = base.columns[i - 6]
            weekly_x.append(begin_week[-5:])

    return weekly_x, weekly_y
```

In [11]:
```python
world_data = pd.read_csv('../Data/world_data.csv')

def cumulate_weekly_data_rows(start_index, end_index, column):
    '''
    Given a start row, end row, and column, this function
    sums up the values of the column for all rows between
    start and end rows and splits the data into weeks. This
    function is used to read data from the world data .csv file.
    :param start_index: The starting row.
    :param end_index: The ending row.
    :param column: The column to sum up.
    :return: Two lists; one containing weeks and the other
    containing the summation of the rows of that week.
    '''

    # calculate summation for the previous week
    prev_sum = 0
    prev_day = world_data.iloc[start_index - 8][column].sum()
    for i in range(start_index - 7, start_index):
        prev_sum += world_data.iloc[i][column]

    curr_sum = 0
    days_passed = 0
    weekly_x = []
    weekly_y = []

    # cumulate weekly data
    for i in range(start_index, end_index + 1):
        val = world_data.iloc[i][column]
        curr_sum += val
        days_passed += 1
        if days_passed == 7:
            weekly_y.append(curr_sum)
            curr_sum = 0
            days_passed = 0
            begin_week = world_data.iloc[i - 6]['date']
            weekly_x.append(begin_week[-5:])

    return weekly_x, weekly_y
```

In [5]:
```python
def print_stats(data, country, data_type):
    '''
    Prints mean, median, and mode about the
    given dataset.
    :param data: A list containing the data.
    :param country: The name of the country
    the data belongs to.
    :param data_type: What type of data is
    contained in data.
    '''

    print('%s weekly %s mean: %15.2f' % (country, data_type, stat.mean(data)))
    print('%s weekly %s median: %13.2f' % (country, data_type, stat.median(data)))
    print('%s weekly %s mode: %15.2f' % (country, data_type, stat.mode(data)))
```

In [6]:
```python
base = pd.read_csv('../Data/base_set_joined.csv')

# get US population
us_population = base['population'].sum()

# get start and end indices
start_index = base.columns.get_loc("cases 2022-06-01")
end_index = base.columns.get_loc("cases 2022-12-31")

us_cases_x, us_cases_y = cumulate_weekly_data_columns(start_index, end_index)

print_stats(us_cases_y, 'US', 'cases')
```

```
US weekly cases mean:       451748.00
US weekly cases median:     366562.50
US weekly cases mode:       626152.00
```

In [7]:
```python
base = pd.read_csv('../Data/base_set_joined.csv')

# get start and end indices
start_index = base.columns.get_loc("deaths 2022-06-01")
end_index = base.columns.get_loc("deaths 2022-12-31")

us_deaths_x, us_deaths_y = cumulate_weekly_data_columns(start_index, end_index)

print_stats(us_deaths_y, 'US', 'deaths')
```

```
US weekly deaths mean:         2308.00
US weekly deaths median:       2401.00
US weekly deaths mode:         1830.00
```

In [13]:
```python
world_data = pd.read_csv('../Data/world_data.csv')

# get Canada indices and population
start_row = world_data.index[(world_data['date'] == '2022-06-01') & (world_data['location'] == 'Canada')].tolist()[0]
end_row = world_data.index[(world_data['date'] == '2022-12-31') & (world_data['location'] == 'Canada')].tolist()[0]
canada_population = world_data.iloc[start_row]['population']

# Canada cases
canada_cases_x, canada_cases_y = cumulate_weekly_data_rows(start_row, end_row, 'new_cases')

# Canada deaths
canada_deaths_x, canada_deaths_y = cumulate_weekly_data_rows(start_row, end_row, 'new_deaths')

print_stats(canada_cases_y, 'Canada', 'cases')
print_stats(canada_deaths_y, 'Canada', 'deaths')
```

```
Canada weekly cases mean:        20233.50
Canada weekly cases median:      18911.50
Canada weekly cases mode:        13035.00
Canada weekly deaths mean:          267.40
Canada weekly deaths median:        282.50
Canada weekly deaths mode:          290.00
```

In [14]:
```python
world_data = pd.read_csv('../Data/world_data.csv')

# get France indices and population
start_row = world_data.index[(world_data['date'] == '2022-06-01') & (world_data['location'] == 'France')].tolist()[0]
end_row = world_data.index[(world_data['date'] == '2022-12-31') & (world_data['location'] == 'France')].tolist()[0]
france_population = world_data.iloc[start_row]['population']

# Canada cases
france_cases_x, france_cases_y = cumulate_weekly_data_rows(start_row, end_row, 'new_cases')

# Canada deaths
france_deaths_x, france_deaths_y = cumulate_weekly_data_rows(start_row, end_row, 'new_deaths')

print_stats(france_cases_y, 'France', 'cases')
print_stats(france_deaths_y, 'France', 'deaths')
```

```
France weekly cases mean:       316450.87
France weekly cases median:     295226.50
France weekly cases mode:       153574.00
France weekly deaths mean:          425.63
France weekly deaths median:        428.50
France weekly deaths mode:          254.00
```

In [16]:
```python
world_data = pd.read_csv('../Data/world_data.csv')

# get Mexico indices and population
start_row = world_data.index[(world_data['date'] == '2022-06-01') & (world_data['location'] == 'Mexico')].tolist()[0]
end_row = world_data.index[(world_data['date'] == '2022-12-31') & (world_data['location'] == 'Mexico')].tolist()[0]
mexico_population = world_data.iloc[start_row]['population']

# Canada cases
mexico_cases_x, mexico_cases_y = cumulate_weekly_data_rows(start_row, end_row, 'new_cases')

# Canada deaths
mexico_deaths_x, mexico_deaths_y = cumulate_weekly_data_rows(start_row, end_row, 'new_deaths')

print_stats(mexico_cases_y, 'Mexico', 'cases')
print_stats(mexico_deaths_y, 'Mexico', 'deaths')
```

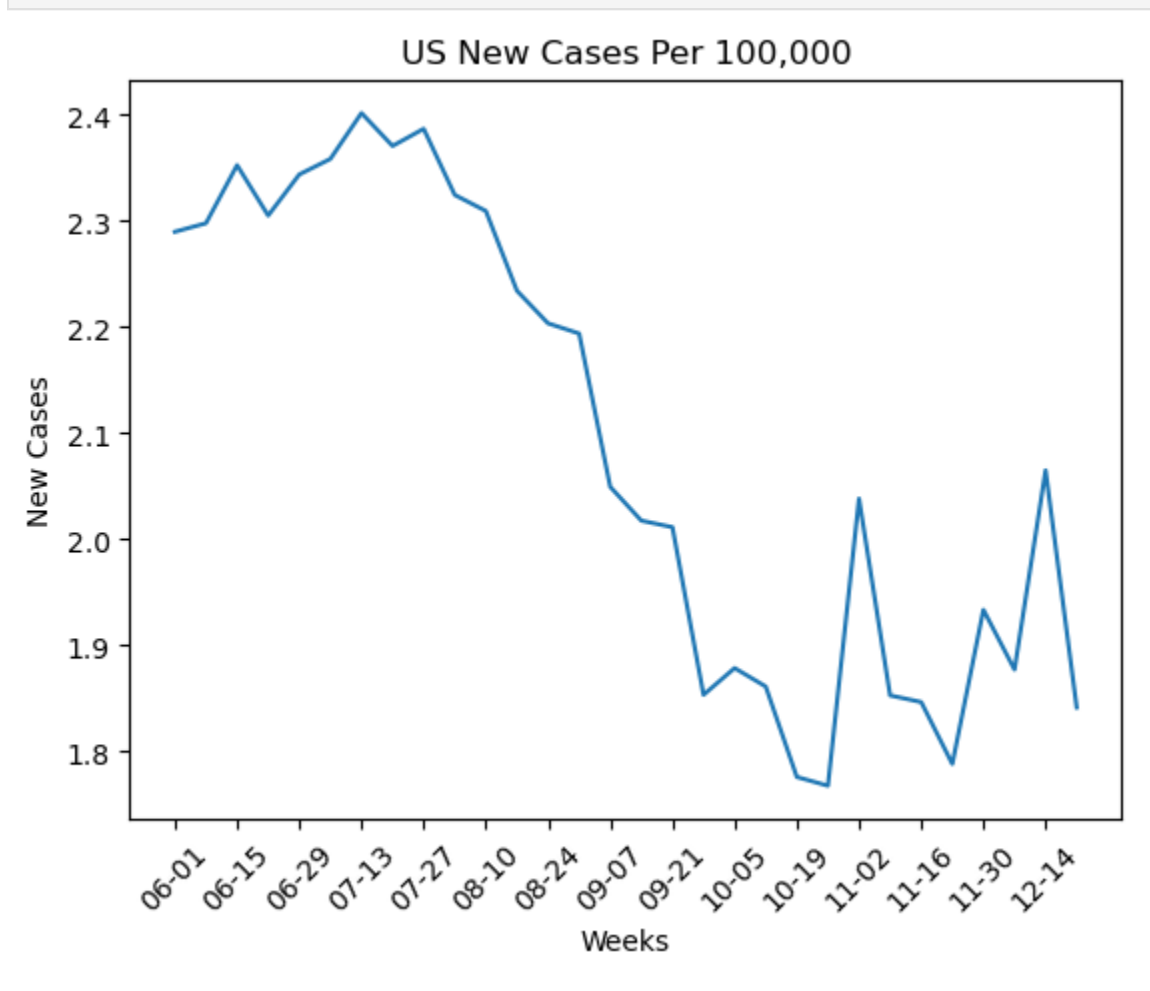
```
Mexico weekly cases mean:        47853.50
Mexico weekly cases median:      20270.50
Mexico weekly cases mode:        20583.00
Mexico weekly deaths mean:          173.60
Mexico weekly deaths median:         78.00
Mexico weekly deaths mode:           25.00
```

In [22]:
```python
# normalize US cases data and log the data
us_cases_normalized = []

for data in us_cases_y:
    us_cases_normalized.append(data / us_population * 100000)

us_cases_normalized = np.log10(us_cases_normalized)

plt.plot(us_cases_x, us_cases_normalized)
plt.xlabel('Weeks')
plt.ylabel('New Cases')
plt.title('US New Cases Per 100,000')
plt.xticks(us_cases_x[::2], rotation = 45)
plt.show()
```

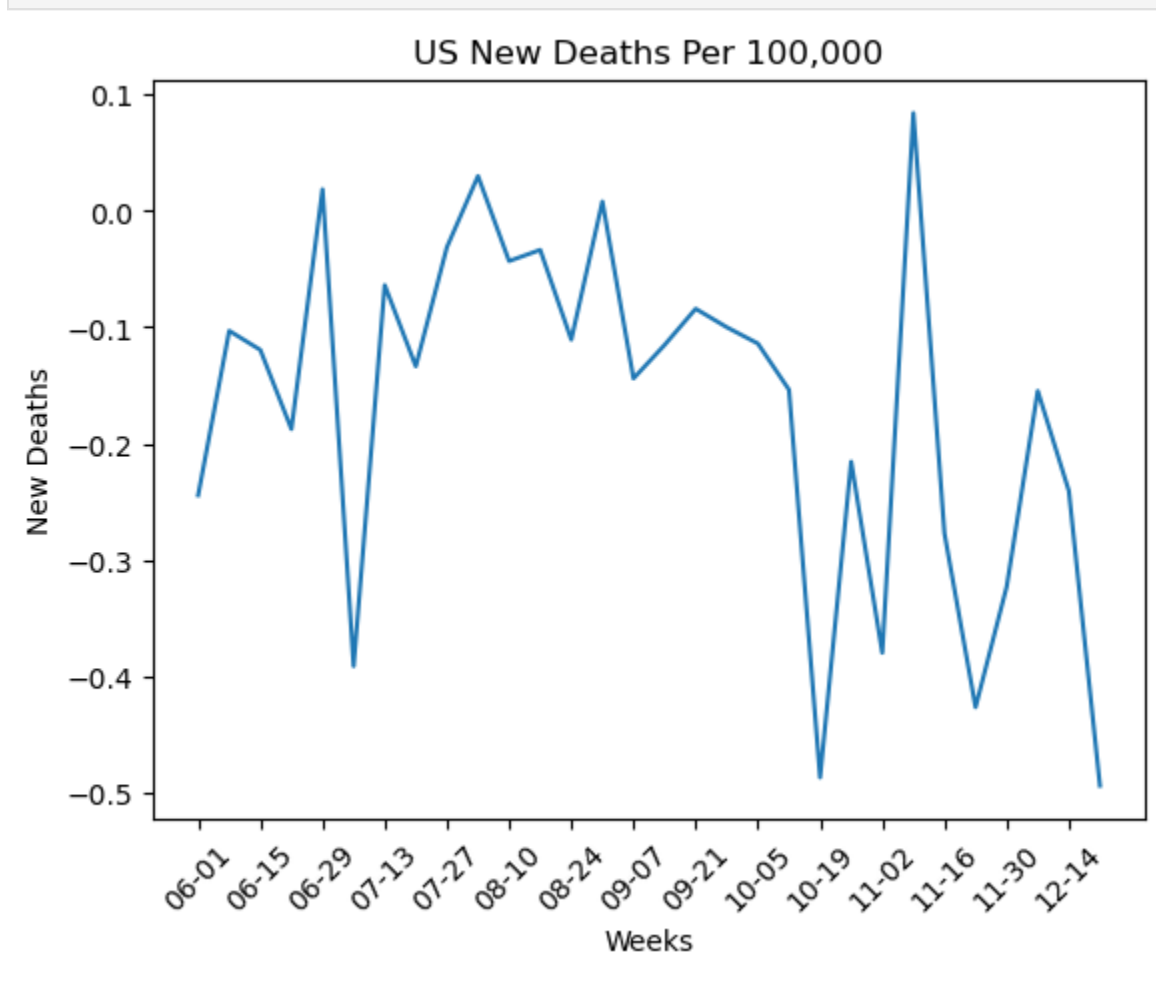


In [23]:
```python
# normalize US deaths data and log the data
us_deaths_normalized = []

for data in us_deaths_y:
    us_deaths_normalized.append(data / us_population * 100000)

us_deaths_normalized = np.log10(us_deaths_normalized)

plt.plot(us_deaths_x, us_deaths_normalized)
plt.xlabel('Weeks')
plt.ylabel('New Deaths')
plt.title('US New Deaths Per 100,000')
plt.xticks(us_deaths_x[::2], rotation = 45)
plt.show()
```

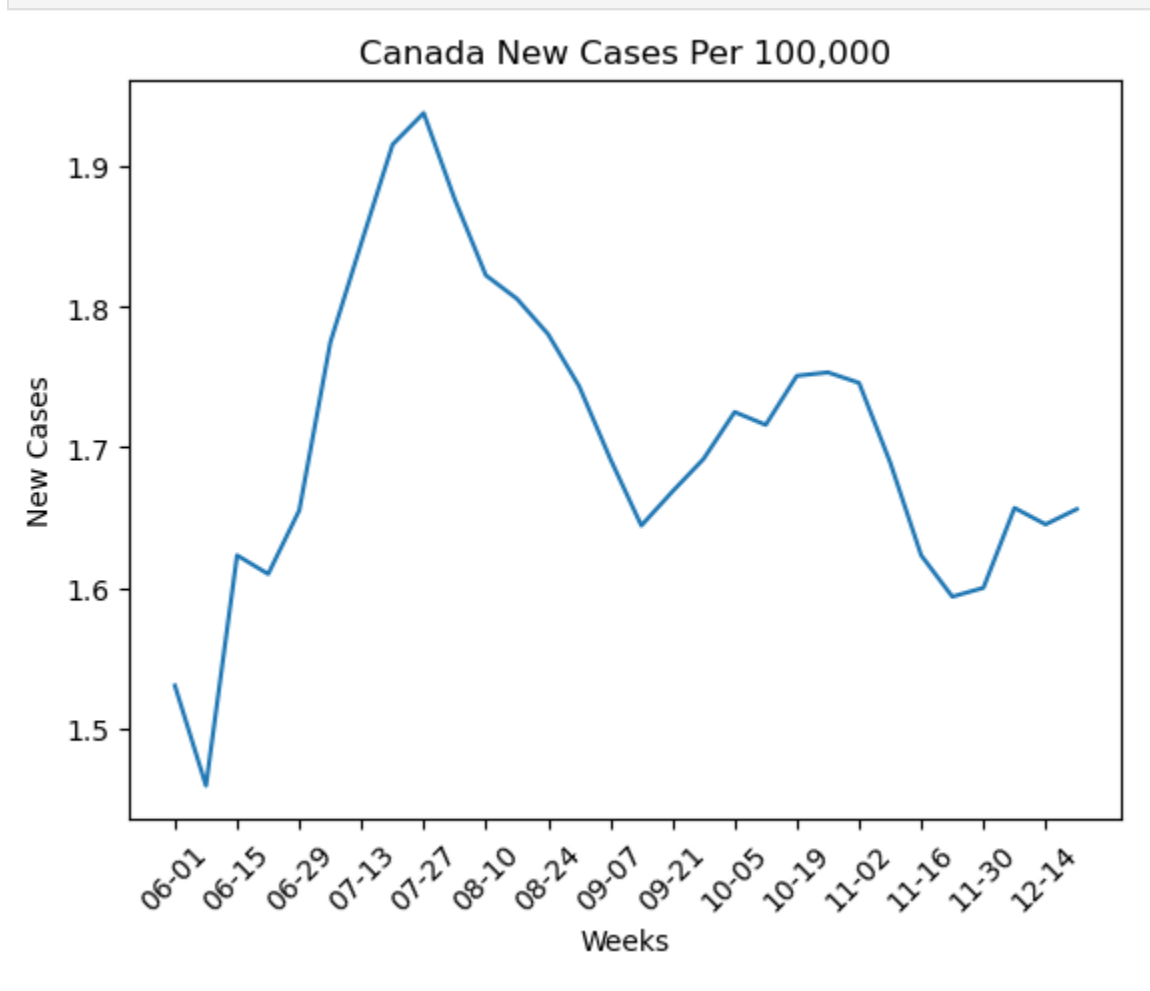


In [21]:
```python
# normalize Canada cases data and log the data
canada_cases_normalized = []

for data in canada_cases_y:
    canada_cases_normalized.append(data / canada_population * 100000)

canada_cases_normalized = np.log10(canada_cases_normalized)

# plot Canada cases
plt.plot(canada_cases_x, canada_cases_normalized)
plt.xlabel('Weeks')
plt.ylabel('New Cases')
plt.title('Canada New Cases Per 100,000')
plt.xticks(canada_cases_x[::2], rotation = 45)
plt.show()
```

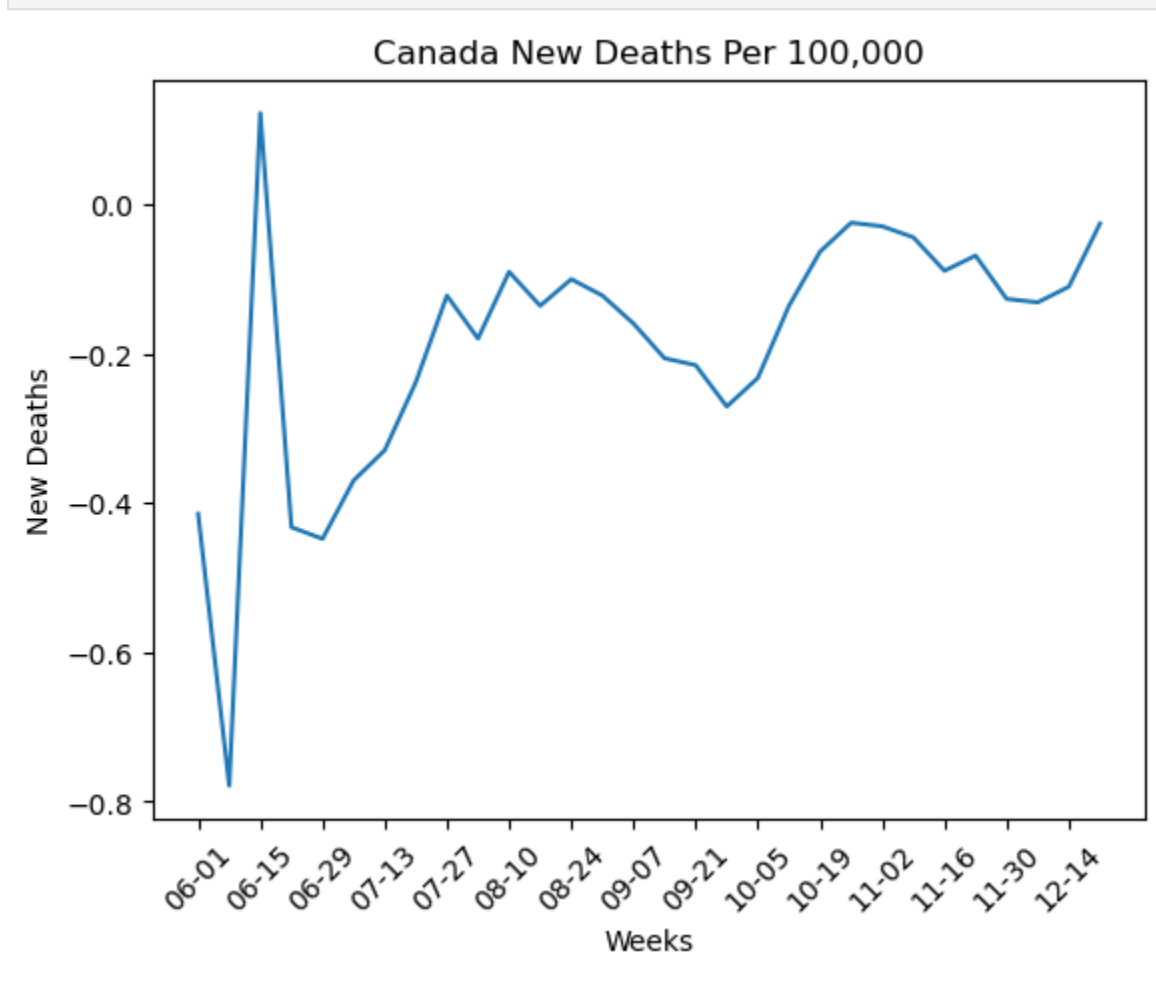


In [20]:
```python
# normalize Canada cases data and log the data
canada_deaths_normalized = []

for data in canada_deaths_y:
    canada_deaths_normalized.append(data / canada_population * 100000)

canada_deaths_normalized = np.log10(canada_deaths_normalized)

# plot Canada deaths
plt.plot(canada_deaths_x, canada_deaths_normalized)
plt.xlabel('Weeks')
plt.ylabel('New Deaths')
plt.title('Canada New Deaths Per 100,000')
plt.xticks(canada_deaths_x[::2], rotation = 45)
plt.show()
```

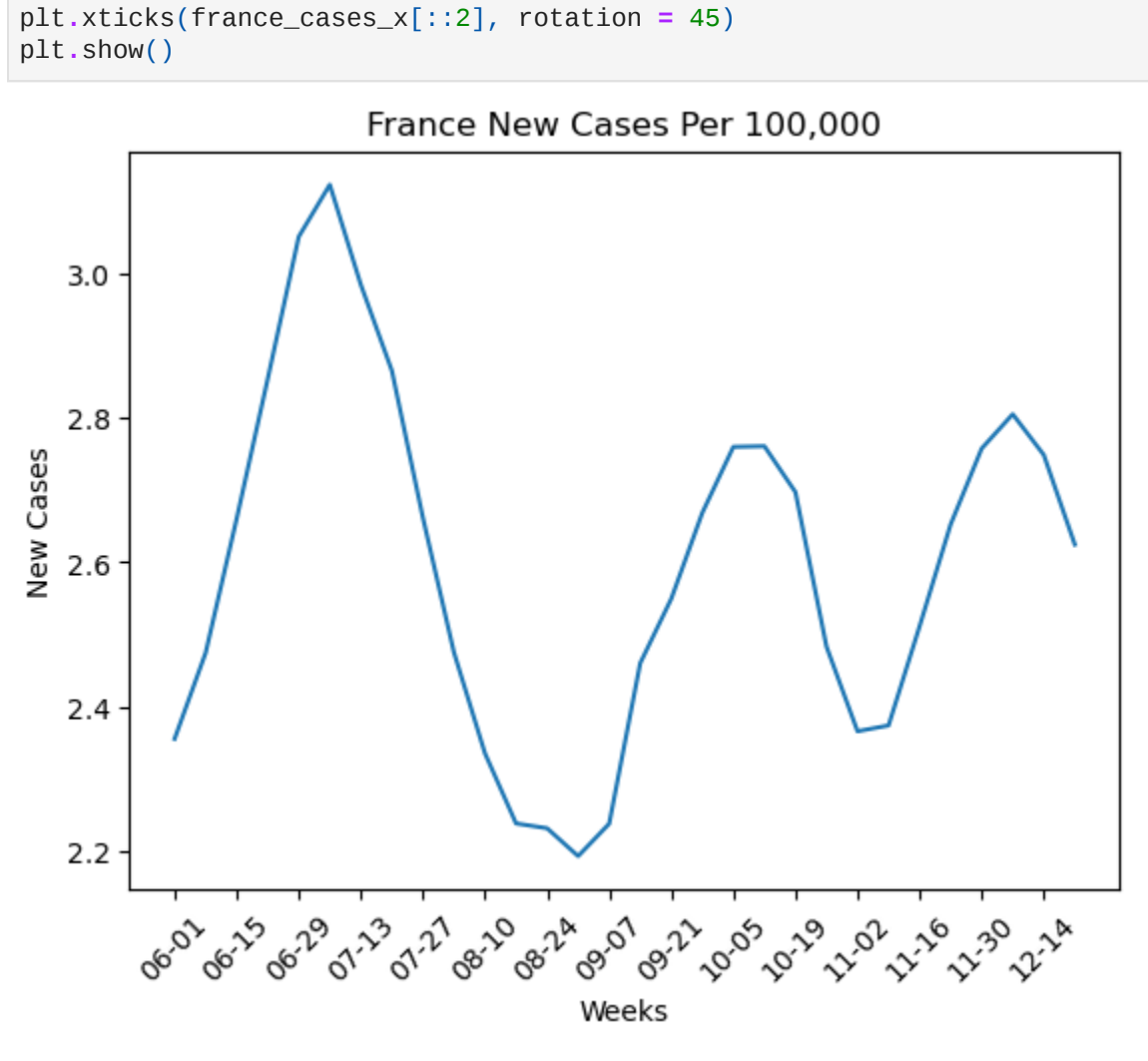


In [25]:
```python
# normalize France cases data and log the data
france_cases_normalized = []

for data in france_cases_y:
    france_cases_normalized.append(data / france_population * 100000)

france_cases_normalized = np.log10(france_cases_normalized)

# plot France cases
plt.plot(france_cases_x, france_cases_normalized)
plt.xlabel('Weeks')
plt.ylabel('New Cases')
plt.title('France New Cases Per 100,000')
plt.xticks(france_cases_x[::2], rotation = 45)
plt.show()
```

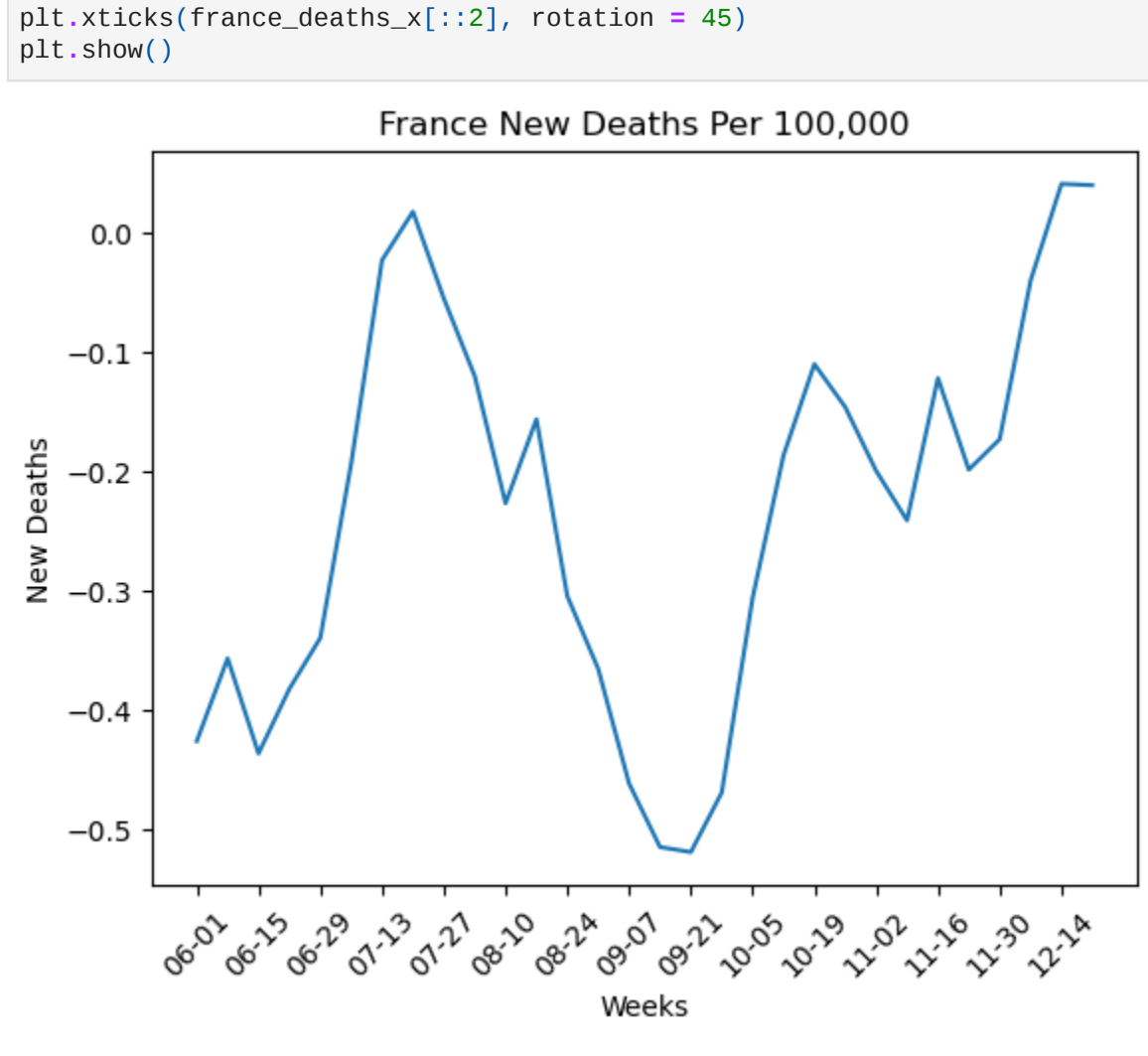


In [26]:
```python
# normalize France cases data and log the data
france_deaths_normalized = []

for data in france_deaths_y:
    france_deaths_normalized.append(data / france_population * 100000)

france_deaths_normalized = np.log10(france_deaths_normalized)

# plot France deaths
plt.plot(france_deaths_x, france_deaths_normalized)
plt.xlabel('Weeks')
plt.ylabel('New Deaths')
plt.title('France New Deaths Per 100,000')
plt.xticks(france_deaths_x[::2], rotation = 45)
plt.show()
```

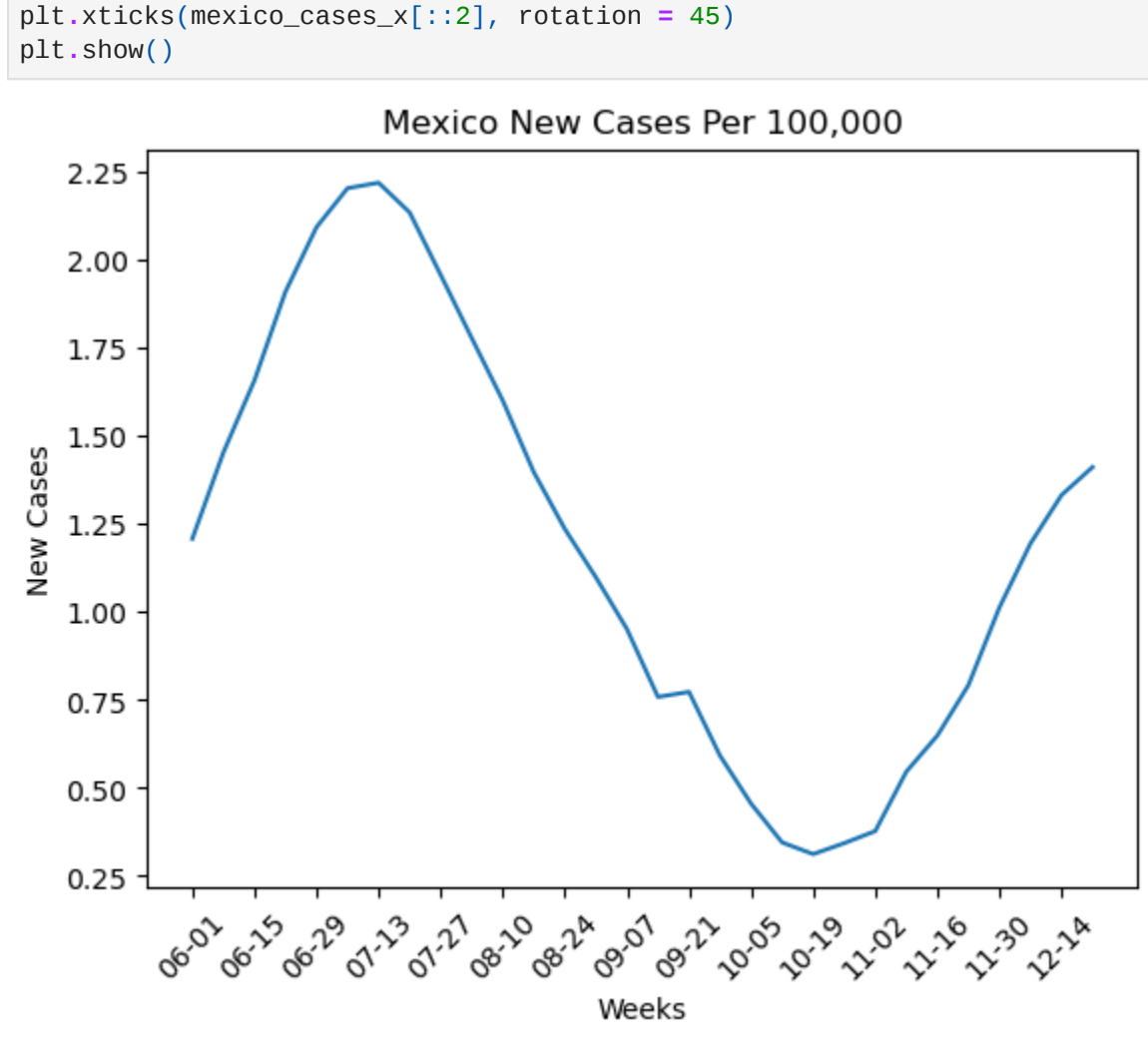


In [27]:
```python
# normalize Mexico cases data and log the data
mexico_cases_normalized = []

for data in mexico_cases_y:
    mexico_cases_normalized.append(data / mexico_population * 100000)

mexico_cases_normalized = np.log10(mexico_cases_normalized)

# plot Mexico cases
plt.plot(mexico_cases_x, mexico_cases_normalized)
plt.xlabel('Weeks')
plt.ylabel('New Cases')
plt.title('Mexico New Cases Per 100,000')
plt.xticks(mexico_cases_x[::2], rotation = 45)
plt.show()
```

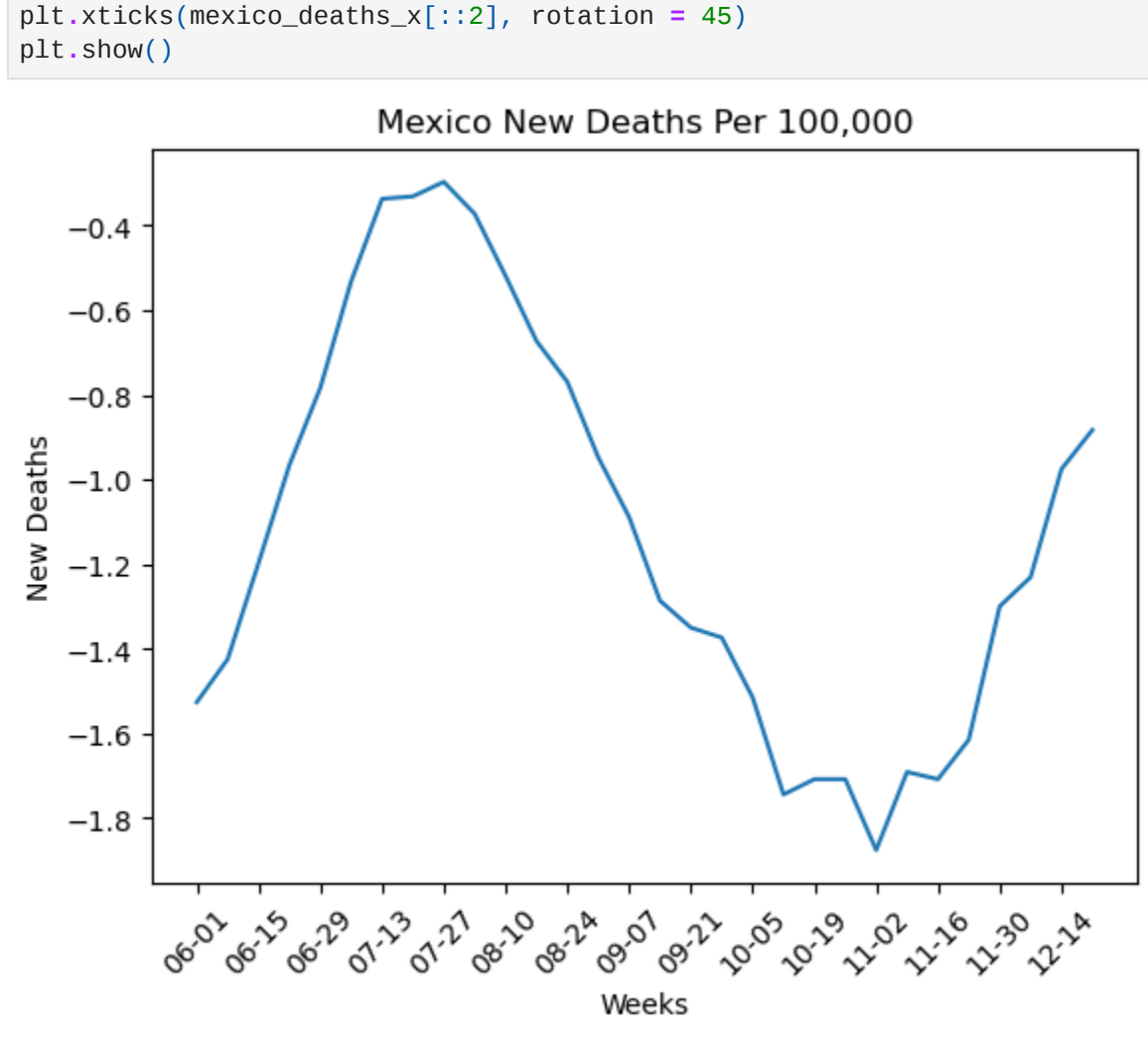


In [28]:
```python
# normalize Mexico cases data and log the data
mexico_deaths_normalized = []

for data in mexico_deaths_y:
    mexico_deaths_normalized.append(data / mexico_population * 100000)

mexico_deaths_normalized = np.log10(mexico_deaths_normalized)

# plot Mexico deaths
plt.plot(mexico_deaths_x, mexico_deaths_normalized)
plt.xlabel('Weeks')
plt.ylabel('New Deaths')
plt.title('Mexico New Deaths Per 100,000')
plt.xticks(mexico_deaths_x[::2], rotation = 45)
plt.show()
```



Cases

Each nation loosely followed the same pattern for cases, with a sharp increase in cases in the beginning of June, and falling steadily until the Winter holidays Thanksgiving and Christmas. The sharp increase around Thanksgiving and Christmas is obvious, as families travel around the world to congregate and spend time with one another it makes sense that Covids transmissions would skyrocket. The exception to this high rate increase is Canada, which did have an increase around the holidays but had a pretty high rate of infection throughout the latter half of 2022.

The peak of covid from 2020 to 2023 was the summer of 2022, and the reason is not as clear as a spike around the holidays. Many people were vaccinated around this time, and receiving boosters, citation here.. The reason is most likely that the vaccinations do not prevent you from getting sick, but prevent you from showing symptoms, and a new variant surfaced. This meant that fully vaccinated individuals were spreading the disease without necessarily showing many symptoms, which would make individuals realizing they are sick take longer. In addition, those who already caught it would have their natural immunity negated, meaning they could get sick to this new variant. (Citation is in the section in June 2022).

Deaths

Each nations death rate follows the same loose pattern as the rate of new cases. This is obviously because as people are getting sick with covid they have a higher chance of dying from other complications. The United States pattern is more erratic, but is still following the loose pattern set by the cases graph.

We have already pointed this out numerous times, but the data pertaining to the US seems to be mismanaged. There are gaps where days go by without any new cases, and the numbers sometimes decrease from day to day. As such, we believe the erratic nature of the pattern reflects largely upon how deaths and cases are reported and recorded in the spreadsheet rather than revealing of the actual pattern of death rates.

There isn't much to say about Mexico; it's pattern is pretty standard. It has two peaks: one in the Summer, and one near Christmas. This is likely due to tourism and Christmas itself respectively.

France is a more interesting case. While it is slightly erratic, there are three peaks that follow the cases curve. The one in the Summer and the one around Christmas can likely be explained by tourism and Christmas, but the one at the beginning of October is more difficult to explain. The likely cause is a three-day holiday called La Fête de la Gastronomie, a festival dedicated to good food in France.

Canada's death rates are more difficult to explain. The curve generally follows that of the cases curve, with two notable peaks, one during August, and one in October. The August peak is likely explained by Summer tourism. Like in the U.S., Summer is a key time for traveling in Canada. The peak in October is less obvious, but doing some research, the cause may be that Canada ended its international Covid travel restrictions October 1st, 2022. This might have lead to a peak in both cases and deaths.