Hypotheses

In [ ]:
```
!pip install statsmodels==0.13.5
```

```
Requirement already satisfied: statsmodels==0.13.5 in /root/venv/lib/python3.9/sit
e-packages (0.13.5)
Requirement already satisfied: scipy>=1.3 in /shared-libs/python3.9/py/lib/python
3.9/site-packages (from statsmodels==0.13.5) (1.9.3)
Requirement already satisfied: patsy>=0.5.2 in /root/venv/lib/python3.9/site-packa
ges (from statsmodels==0.13.5) (0.5.3)
Requirement already satisfied: numpy>=1.17 in /shared-libs/python3.9/py/lib/python
3.9/site-packages (from statsmodels==0.13.5) (1.23.4)
Requirement already satisfied: pandas>=0.25 in /shared-libs/python3.9/py/lib/pytho
n3.9/site-packages (from statsmodels==0.13.5) (1.2.5)
Requirement already satisfied: packaging>=21.3 in /shared-libs/python3.9/py-core/l
ib/python3.9/site-packages (from statsmodels==0.13.5) (21.3)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /shared-libs/python3.9/
py-core/lib/python3.9/site-packages (from packaging>=21.3->statsmodels==0.13.5)
(3.0.9)
Requirement already satisfied: python-dateutil>=2.7.3 in /shared-libs/python3.9/py
-core/lib/python3.9/site-packages (from pandas>=0.25->statsmodels==0.13.5) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /shared-libs/python3.9/py/lib/pytho
n3.9/site-packages (from pandas>=0.25->statsmodels==0.13.5) (2022.5)
Requirement already satisfied: six in /shared-libs/python3.9/py-core/lib/python3.
9/site-packages (from patsy>=0.5.2->statsmodels==0.13.5) (1.16.0)
WARNING: You are using pip version 22.0.4; however, version 23.1.1 is available.
You should consider upgrading via the '/root/venv/bin/python -m pip install --upgr
ade pip' command.
```

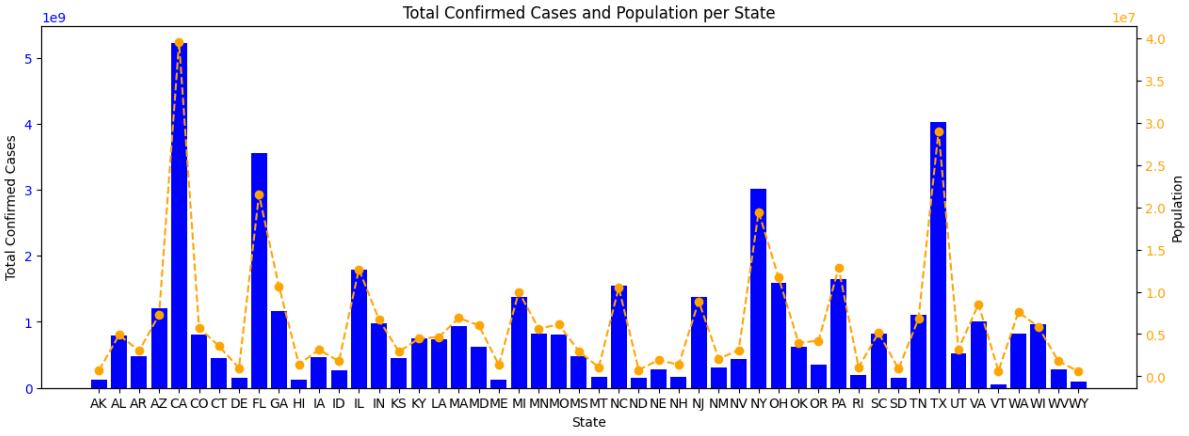
In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from math import sqrt
from sklearn.svm import SVR
from statsmodels.tsa.arima.model import ARIMA
import warnings
warnings.filterwarnings('ignore')
```

In [ ]:
```
confirmed_cases = pd.read_csv('data/covid_confirmed_usafacts.csv')
pop_for_us = pd.read_csv('data/covid_county_population_usafacts.csv', usecols= ['co
state_areas = pd.read_csv('state-areas.csv')


state_abbr = {
    'Alabama': 'AL',
    'Alaska': 'AK',
    'Arizona': 'AZ',
    'Arkansas': 'AR',
```

```
    'California': 'CA',
    'Colorado': 'CO',
    'Connecticut': 'CT',
    'Delaware': 'DE',
    'Florida': 'FL',
    'Georgia': 'GA',
    'Hawaii': 'HI',
    'Idaho': 'ID',
    'Illinois': 'IL',
    'Indiana': 'IN',
    'Iowa': 'IA',
    'Kansas': 'KS',
    'Kentucky': 'KY',
    'Louisiana': 'LA',
    'Maine': 'ME',
    'Maryland': 'MD',
    'Massachusetts': 'MA',
    'Michigan': 'MI',
    'Minnesota': 'MN',
    'Mississippi': 'MS',
    'Missouri': 'MO',
    'Montana': 'MT',
    'Nebraska': 'NE',
    'Nevada': 'NV',
    'New Hampshire': 'NH',
    'New Jersey': 'NJ',
    'New Mexico': 'NM',
    'New York': 'NY',
    'North Carolina': 'NC',
    'North Dakota': 'ND',
    'Ohio': 'OH',
    'Oklahoma': 'OK',
    'Oregon': 'OR',
    'Pennsylvania': 'PA',
    'Rhode Island': 'RI',
    'South Carolina': 'SC',
    'South Dakota': 'SD',
    'Tennessee': 'TN',
    'Texas': 'TX',
    'Utah': 'UT',
    'Vermont': 'VT',
    'Virginia': 'VA',
    'Washington': 'WA',
    'West Virginia': 'WV',
    'Wisconsin': 'WI',
    'Wyoming': 'WY'
}

state_areas['State Abbreviation'] = state_areas['state'].map(state_abbr)

covid_data = pd.merge(confirmed_cases, pop_for_us, on= 'countyFIPS', suffixes="_cas
covid_data.drop(covid_data[covid_data["County Name"].str.contains("Statewide")==Tru
state_pop = covid_data.groupby('State').sum()['population']
covid_data = covid_data.groupby('State')[list(covid_data.columns[4:])].sum().reset_
covid_data['cases_total'] = covid_data.iloc[:, 1:-1].sum(axis=1)
```

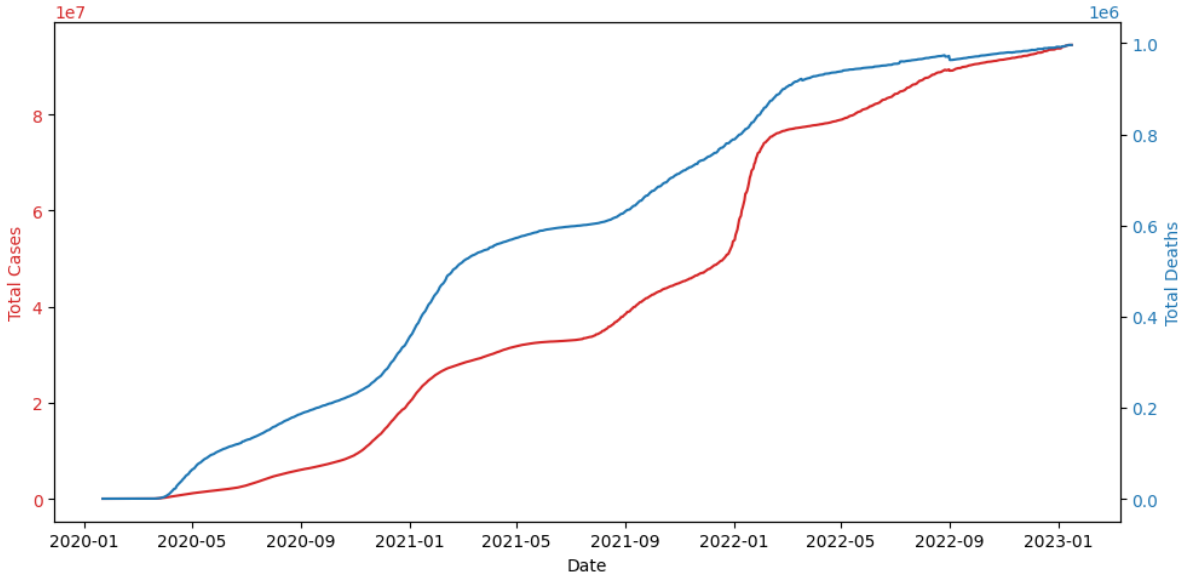
```
state_cases = covid_data.merge(state_areas, left_on='State', right_on='State Abbrev
state_cases.drop('State Abbreviation', axis=1, inplace=True)
state_cases.drop('state', axis=1, inplace=True)

state_cases['density'] = state_cases['population'] / state_cases["area (sq. mi)"]

state_data = state_cases.groupby("State").agg({"population": "sum", "density": "mea

correlation = state_data["density"].corr(state_data["cases_total"])

print("Correlation coefficient between population density and confirmed cases:", co

fig, ax1 = plt.subplots(figsize=(15, 5))

# plot total confirmed cases on primary y-axis
ax1.bar(state_data.index, state_data['cases_total'], color='blue')
ax1.set_xlabel('State')
ax1.set_ylabel('Total Confirmed Cases')
ax1.tick_params(axis='y', labelcolor='blue')

# create a secondary y-axis for population
ax2 = ax1.twinx()
ax2.plot(state_data.index, state_data['population'], color='orange', marker='o', li
ax2.set_ylabel('Population')
ax2.tick_params(axis='y', labelcolor='orange')

plt.title('Total Confirmed Cases and Population per State')
plt.show()
```

```
Correlation coefficient between population density and confirmed cases: 0.21037154
624992097
```



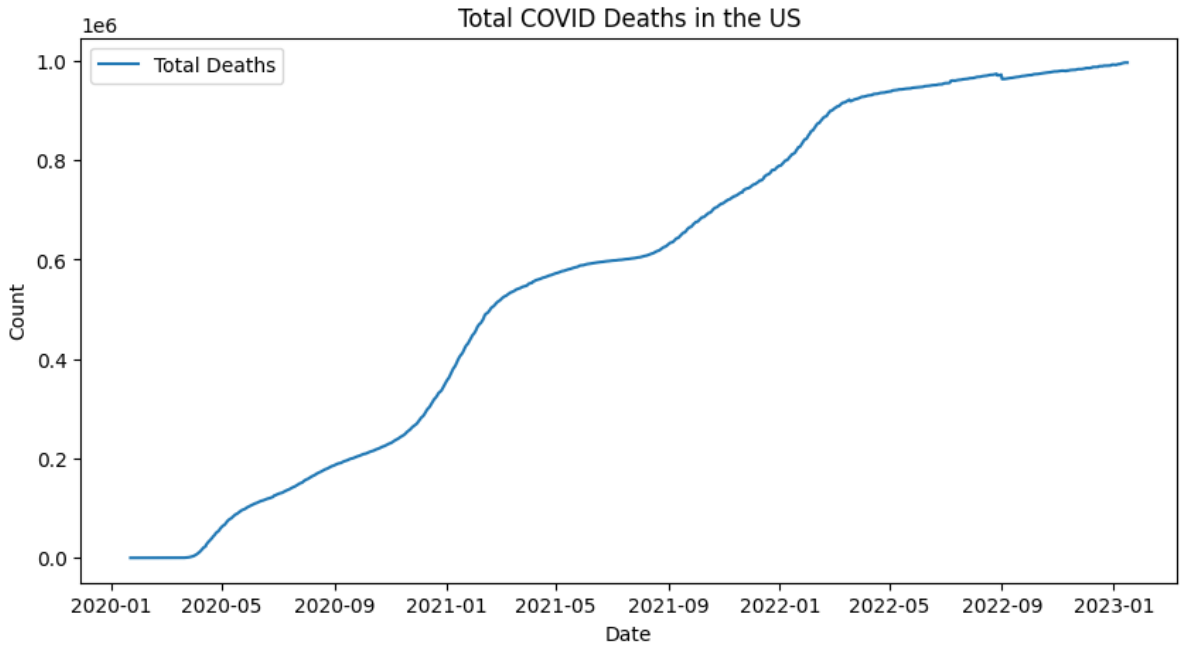
In [ ]:
```
cases_data = pd.read_csv('data/covid_confirmed_usafacts.csv')
cases_data.drop(cases_data[cases_data["County Name"].str.contains("Statewide")==Tru
cases_data = cases_data.iloc[:, 4:].sum(axis=0).reset_index()

deaths_data = pd.read_csv('data/covid_deaths_usafacts.csv')
deaths_data.drop(deaths_data[deaths_data["County Name"].str.contains("Statewide")==
deaths_data = deaths_data.iloc[:, 4:].sum(axis=0).reset_index()

# Rename the columns
cases_data.columns = ['date', 'total_cases']
deaths_data.columns = ['date', 'total_deaths']
```

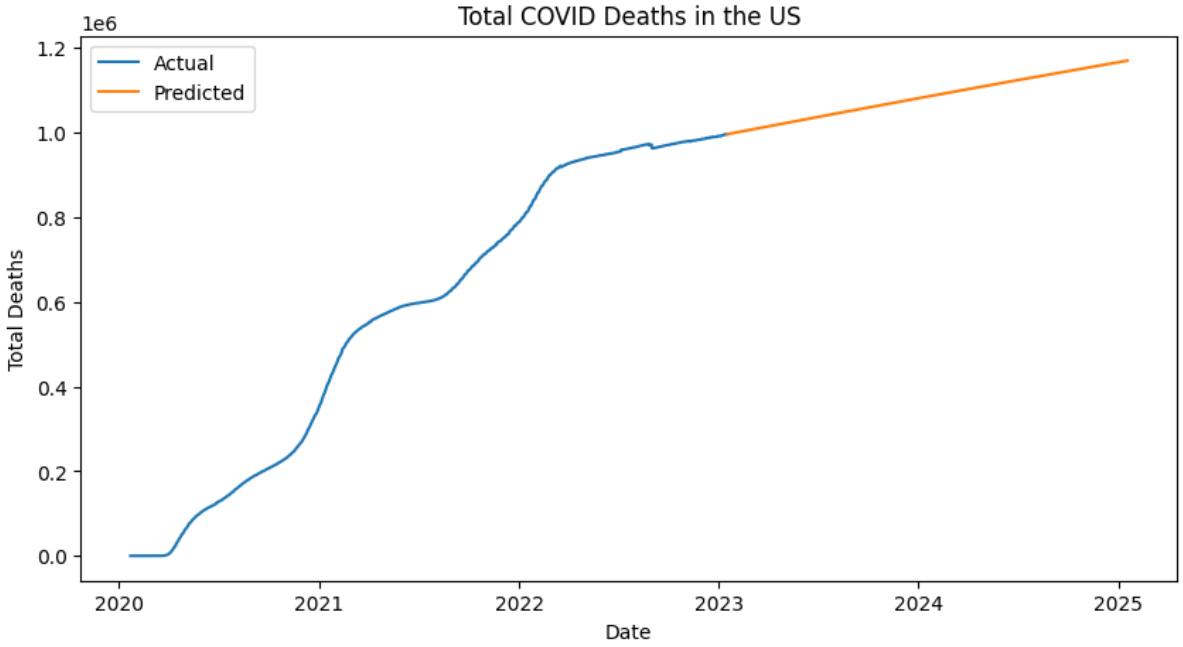
```
# Convert the date column to a datetime object
cases_data['date'] = pd.to_datetime(cases_data['date'])
deaths_data['date'] = pd.to_datetime(deaths_data['date'])

fig, ax1 = plt.subplots(figsize=(10,5))

color = 'tab:red'
ax1.set_xlabel('Date')
ax1.set_ylabel('Total Cases', color=color)
ax1.plot(cases_data['date'], cases_data['total_cases'], color=color)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx()

color = 'tab:blue'
ax2.set_ylabel('Total Deaths', color=color)
ax2.plot(deaths_data['date'], deaths_data['total_deaths'], color=color)
ax2.tick_params(axis='y', labelcolor=color)

fig.tight_layout()
plt.show()
```



In [ ]:
```
fig, ax = plt.subplots(figsize=(10,5))
ax.plot(deaths_data['date'], deaths_data['total_deaths'], label='Total Deaths')
ax.set_xlabel('Date')
ax.set_ylabel('Count')
ax.set_title('Total COVID Deaths in the US')
ax.legend()
plt.show()

model = ARIMA(deaths_data['total_deaths'], order=(2, 1, 2))
model_fit = model.fit()

start_date = deaths_data['date'].max() + pd.DateOffset(days=1)
end_date = start_date + pd.DateOffset(years=1)
future_dates = pd.date_range(start_date, end_date, freq='D')
forecast = model_fit.forecast(steps=len(future_dates))
```

```
fig, ax = plt.subplots(figsize=(10, 5))
ax.plot(deaths_data['date'], deaths_data['total_deaths'], label='Actual')
ax.plot(future_dates, forecast, label='Predicted')
ax.set_xlabel('Date')
ax.set_ylabel('Total Deaths')
ax.set_title('Total COVID Deaths in the US')
ax.legend()
plt.show()
```

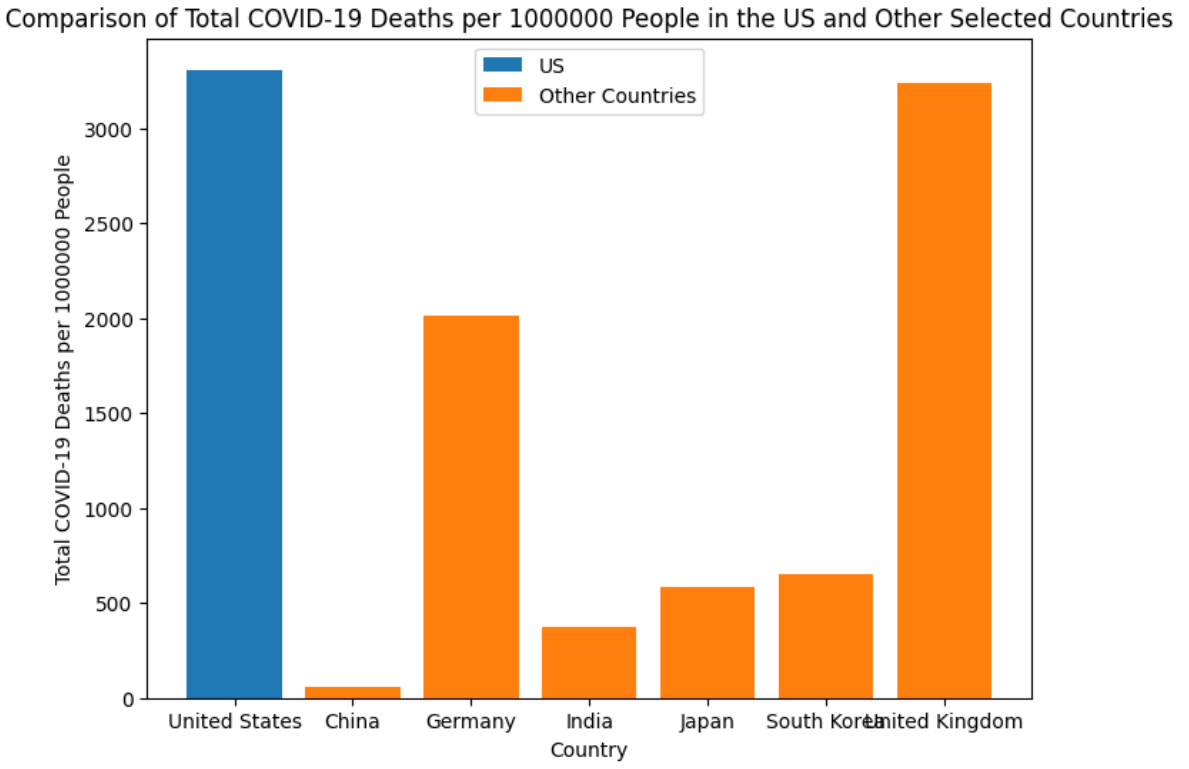




Comparing the U.S. to other Countries

In [ ]:
```
world_df = pd.read_csv("data/owid-covid-data-cases.csv", usecols=['location','date'
world_df = world_df.fillna(0) #fill our NaN values to zero.
```

```
FACTOR = 1000000

us_covid_data = world_df[world_df['location'] == 'United States']
other_countries_covid_data = world_df[world_df['location'].isin(['China', 'India',

# Calculate the statistics per million population
us_covid_data['total_cases_per_million'] = us_covid_data['total_cases'] / (us_covid
us_covid_data['new_cases_per_million'] = us_covid_data['new_cases'] / (us_covid_dat
us_covid_data['total_deaths_per_million'] = us_covid_data['total_deaths'] / (us_cov
us_covid_data['new_deaths_per_million'] = us_covid_data['new_deaths'] / (us_covid_d

other_countries_covid_data['total_cases_per_million'] = other_countries_covid_data[
other_countries_covid_data['new_cases_per_million'] = other_countries_covid_data['n
other_countries_covid_data['total_deaths_per_million'] = other_countries_covid_data
other_countries_covid_data['new_deaths_per_million'] = other_countries_covid_data['

fig, ax = plt.subplots(figsize=(8,6))

ax.bar(us_covid_data['location'], us_covid_data['total_deaths_per_million'], label=
ax.bar(other_countries_covid_data['location'], other_countries_covid_data['total_de

ax.set_xlabel('Country')
ax.set_ylabel(f'Total COVID-19 Deaths per {FACTOR} People')
ax.set_title(f"Comparison of Total COVID-19 Deaths per {FACTOR} People in the US an
ax.legend()

plt.show()
```



Created in Deepnote