sir

June 7, 2023

[2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.integrate import odeint
```

[3]:
```
# Read the data from the CSV files
airport_data = pd.read_csv('/content/airport_df.csv')
airport_data.head()
```

[3]:

	ID	Name	City	\
0	1	Goroka Airport	Goroka	
1	2	Madang Airport	Madang	
2	3	Mount Hagen Kagamuga Airport	Mount Hagen	
3	4	Nadzab Airport	Nadzab	
4	5	Port Moresby Jacksons International Airport	Port Moresby	

	Country	IATA	ICAO	Lat	Long	Alt	Timezone	DST	\
0	Papua New Guinea	GKA	AYGA	-6.081690	145.391998	5282	10	U	
1	Papua New Guinea	MAG	AYMD	-5.207080	145.789001	20	10	U	
2	Papua New Guinea	HGU	AYMH	-5.826790	144.296005	5388	10	U	
3	Papua New Guinea	LAE	AYNZ	-6.569803	146.725977	239	10	U	
4	Papua New Guinea	POM	AYPY	-9.443380	147.220001	146	10	U	

	Tz database time zone	type	source
0	Pacific/Port_Moresby	airport	OurAirports
1	Pacific/Port_Moresby	airport	OurAirports
2	Pacific/Port_Moresby	airport	OurAirports
3	Pacific/Port_Moresby	airport	OurAirports
4	Pacific/Port_Moresby	airport	OurAirports

[8]:
```
# Extract the relevant columns from the datasets
airport_passengers = airport_data['Alt'].values
```

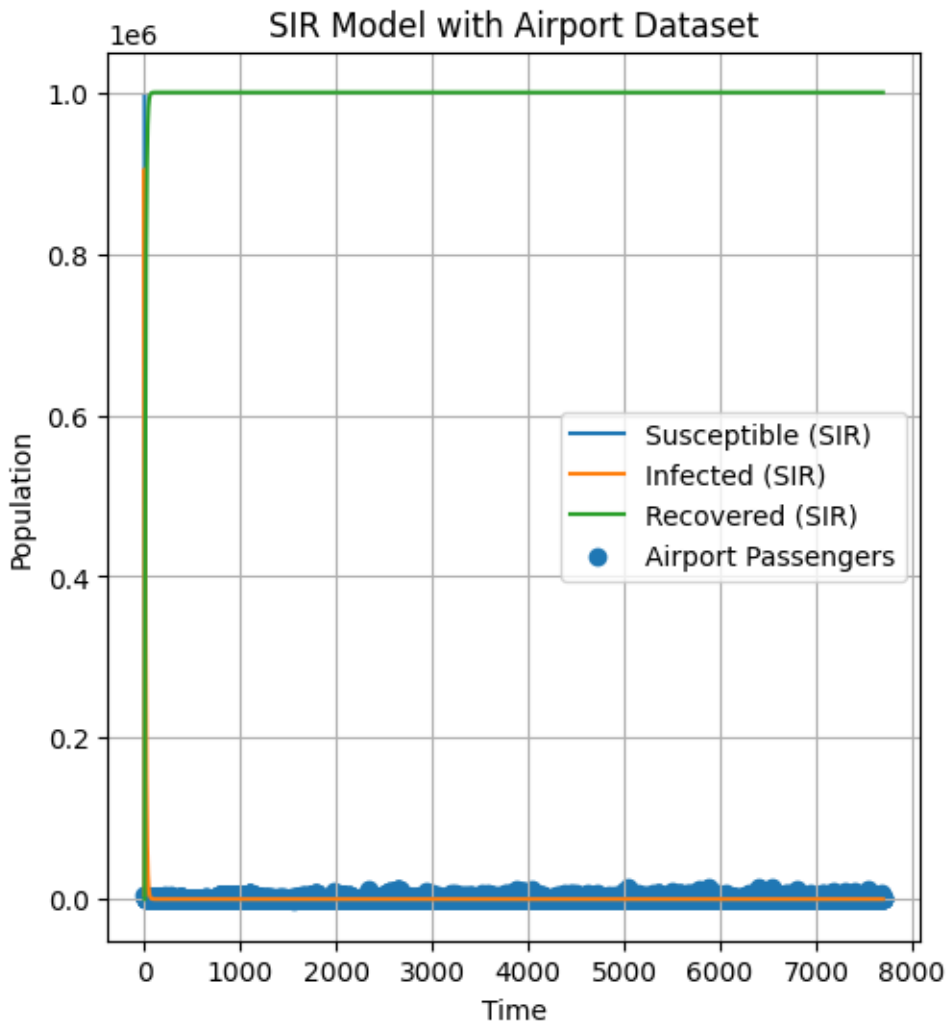
[9]:
```
# Parameters
population_size = 1000000
contact_rate = 0.3
recovery_rate = 0.1
```

```
incubation_rate = 0.2
duration = len(airport_passengers) - 1
```

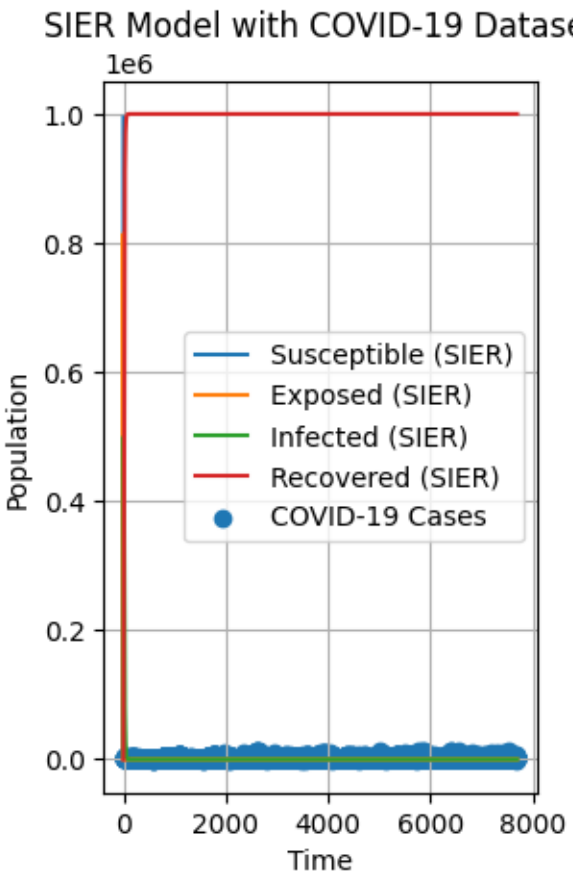
[10]:
```
# SIR model
def sir_model(y, t, contact_rate, recovery_rate):
    S, I, R = y
    dSdt = -contact_rate * S * I
    dIdt = contact_rate * S * I - recovery_rate * I
    dRdt = recovery_rate * I
    return [dSdt, dIdt, dRdt]

# SIER model
def sier_model(y, t, contact_rate, recovery_rate, incubation_rate):
    S, E, I, R = y
    dSdt = -contact_rate * S * I
    dEdt = contact_rate * S * I - incubation_rate * E
    dIdt = incubation_rate * E - recovery_rate * I
    dRdt = recovery_rate * I
    return [dSdt, dEdt, dIdt, dRdt]
```

[11]:
```
# Initial conditions
initial_infected = airport_passengers[0]
y0_sir = [population_size - initial_infected, initial_infected, 0]
y0_sier = [population_size - initial_infected, 0, initial_infected, 0]

# Time points
t = np.arange(duration + 1)

# Solve ODEs
solution_sir = odeint(sir_model, y0_sir, t, args=(contact_rate, recovery_rate))
solution_sier = odeint(sier_model, y0_sier, t, args=(contact_rate,␣
 ↪recovery_rate, incubation_rate))
```

[12]:
```
# Plot results
plt.figure(figsize=(12, 6))

# SIR model
plt.subplot(1, 2, 1)
plt.plot(t, solution_sir[:, 0], label='Susceptible (SIR)')
plt.plot(t, solution_sir[:, 1], label='Infected (SIR)')
plt.plot(t, solution_sir[:, 2], label='Recovered (SIR)')
plt.scatter(t, airport_passengers, label='Airport Passengers')
plt.xlabel('Time')
plt.ylabel('Population')
plt.title('SIR Model with Airport Dataset')
plt.legend()
plt.grid(True)
```



[13]:
```
# SIER model
plt.subplot(1, 2, 2)
plt.plot(t, solution_sier[:, 0], label='Susceptible (SIER)')
plt.plot(t, solution_sier[:, 1], label='Exposed (SIER)')
plt.plot(t, solution_sier[:, 2], label='Infected (SIER)')
plt.plot(t, solution_sier[:, 3], label='Recovered (SIER)')
plt.scatter(t, airport_passengers, label='COVID-19 Cases')
plt.xlabel('Time')
plt.ylabel('Population')
plt.title('SIER Model with COVID-19 Dataset')
plt.legend()
plt.grid(True)
```


SIER Model with COVID-19 Dataset
1e6
1.0
0.8
0.6
0.4
0.2
0.0
Population
0
2000
4000
6000
8000
Time
Susceptible (SIER)
Exposed (SIER)
Infected (SIER)
Recovered (SIER)
COVID-19 Cases