USA_cleaning

May 14, 2021

```
[1]: import pandas as pd
     import numpy as np
     import math
     import matplotlib.pyplot as plt
     import datetime
     from datetime import datetime as dt
```

```
[2]: from google.colab import drive
     drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
[3]: %cd '/content/drive/MyDrive/CSE544_PROJECT'
     %ls -l
```

```
/content/drive/.shortcut-targets-by-
id/1YQyVsZWGB7sACOZzGl1QA0QwFc_E5Nb1/CSE544_PROJECT
total 972
-rw------- 1 root root  13113 May  8 15:12  14.csv
-rw------- 1 root root   9808 May 10 08:37 '2a EWMA.ipynb'
-rw------- 1 root root 219146 May 10 08:38  2c.ipynb
-rw------- 1 root root  10403 May  8 19:17  August.csv.xlsx
-rw------- 1 root root   4319 May  8 21:23  August_Final.csv
-rw------- 1 root root  27056 May 10 05:29  clean.csv
-rw------- 1 root root  19894 May 10 03:40  CSE544_PROJECT.ipynb
-rw------- 1 root root  10672 May 10 22:46  fuel_clean.csv
-rw------- 1 root root  49009 May 10 22:47  fuel_cleaning.ipynb
-rw------- 1 root root   5099 May 10 22:09  fuel_unclean.csv
-rw------- 1 root root  14155 May  8 21:06  OCT_NOV_DEC.xlsx
-rw------- 1 root root  29958 May 10 20:39  post-cleaning.ipynb
-rw------- 1 root root  20198 May  8 15:35  sample.csv
-rw------- 1 root root  19754 May 10 08:34  SNEH_clean.csv
-rw------- 1 root root 153968 May 10 08:44  Sneh_trial.ipynb
-rw------- 1 root root   3807 May  8 22:40  temp2.csv
-rw------- 1 root root   3819 May  8 23:22  temp3.csv
-rw------- 1 root root   3814 May 10 00:18  temp.csv
-rw------- 1 root root  10849 May 10 23:30  USA_clean.csv
-rw------- 1 root root 130264 May 11 02:06  USA_cleaning.ipynb
```

```
-rw------- 1 root root 131790 May 10 22:49  US_confirmed.csv
-rw------- 1 root root  99103 May 10 23:04  US_deaths.csv
```

[4]:

```
df = pd.read_csv('US_confirmed.csv')
df.head()
```

[4]:

```
  State  2020-01-22  2020-01-23  …  2021-04-01  2021-04-02  2021-04-03
0    AK           0           0  …       60628       60823       60823
1    AL           0           0  …      515893      516309      516662
2    AR           0           0  …      330611      330756      330972
3    AZ           0           0  …      842273      843174      844328
4    CA           0           0  …     3570718     3573028     3577951

[5 rows x 439 columns]
```

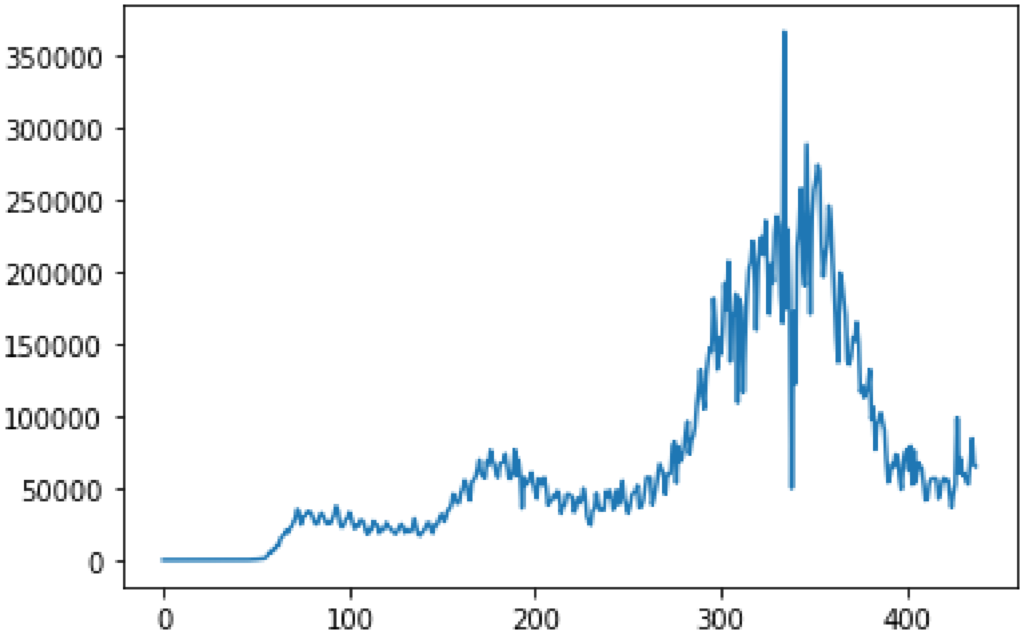
[5]:

```
# Add data of all states to get total US cases

data = df.values
data = np.sum(data, axis=0)
data = data[1:]
print(len(data))
```

```
438
```

[6]:

```
# Find daily cases

data_daily = np.zeros(len(data))
data_daily[0] = data[0]
for x in range(1, len(data)):
  data_daily[x] = data[x] - data[x-1]
plt.plot(data_daily)
# print(len(data_daily))
cases = data_daily
```



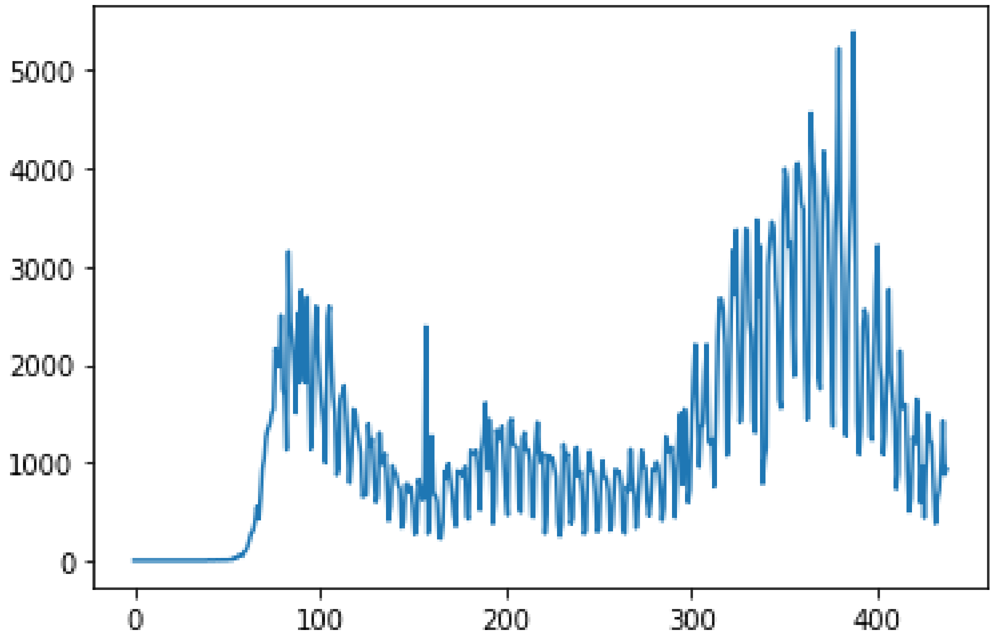
[7]:
```
# Create a list of dates

start_date = datetime.datetime(2020, 1, 22)
end_date = datetime.datetime(2021, 4, 3)
delta = datetime.timedelta(days=1)
date = []

while start_date<=end_date:
  date.append(start_date)
  start_date += delta

print(len(date))
```

```
438
```

[8]:
```
# Follow the same procedure for deaths

df = pd.read_csv('US_deaths.csv')
df.head()

# Add data of all states to get total US deaths
data = df.values
data = np.sum(data, axis=0)
data = data[1:]
print(len(data))
```

```
# Find daily deaths
data_daily = np.zeros(len(data))
data_daily[0] = data[0]
for x in range(1, len(data)):
  data_daily[x] = data[x] - data[x-1]
plt.plot(data_daily)
# print(len(data_daily))
death = data_daily
```

```
438
```



[9]:

```
# apply tukeys rule for removing outliers

import statistics

def tukey(price_list):
  month_price_list = []
  # lst1 = price_list[i:i+30] for i in range(0,len(df)-30+1,30)]
  for i in range(0,len(price_list)-30+1,30):
    month_price_list.append(price_list[i:i+30])
  month_price_list.append(price_list[420:])
  price_list_tukey = []
  for month in month_price_list:
    median = statistics.median(month)
    month_sorted = np.sort(month)
    q25 = month_sorted[math.ceil((25/100)*len(month))-1]
    q75 = month_sorted[math.ceil((75/100)*len(month))-1]
    iqr = q75 - q25
    cut_off = iqr * 1.5
    lower, upper = q25 - cut_off, q75 + cut_off
    numchanges = 0
    for i, x in enumerate(month):
        if x < lower or x > upper:
            month[i] = median
            numchanges += 1
    print("outliers = ", numchanges)
    price_list_tukey.extend(month)
    # plt.plot(price_list_tukey)
  return price_list_tukey
```

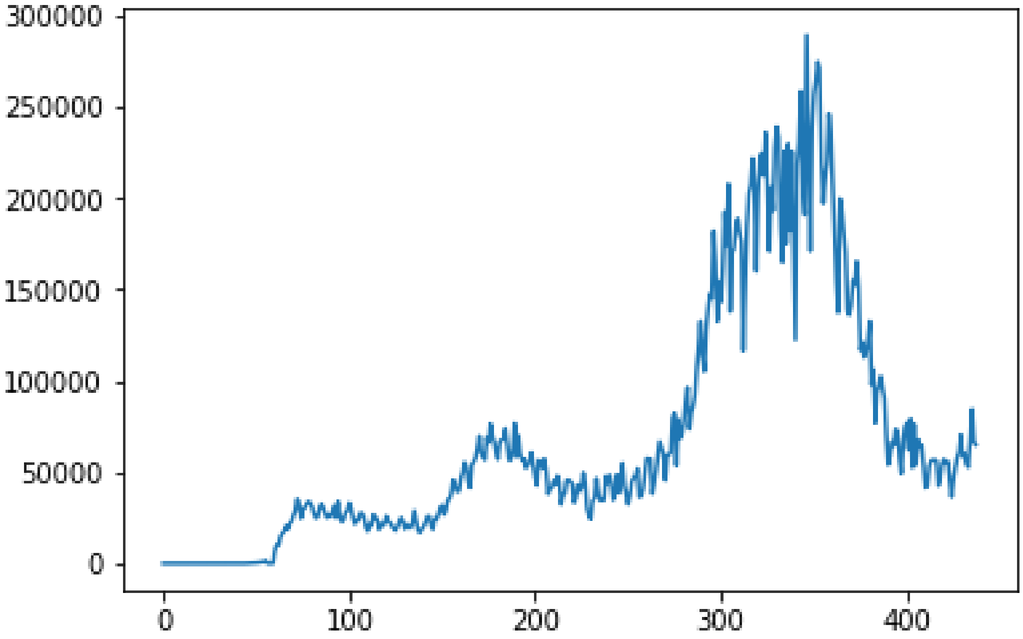
[10]:

```
cases = tukey(cases)
plt.plot(cases)
```

```
outliers =  1
outliers =  4
outliers =  0
outliers =  1
outliers =  0
outliers =  0
outliers =  1
outliers =  0
outliers =  0
outliers =  0
outliers =  1
outliers =  2
outliers =  0
outliers =  0
outliers =  1
```

[10]:

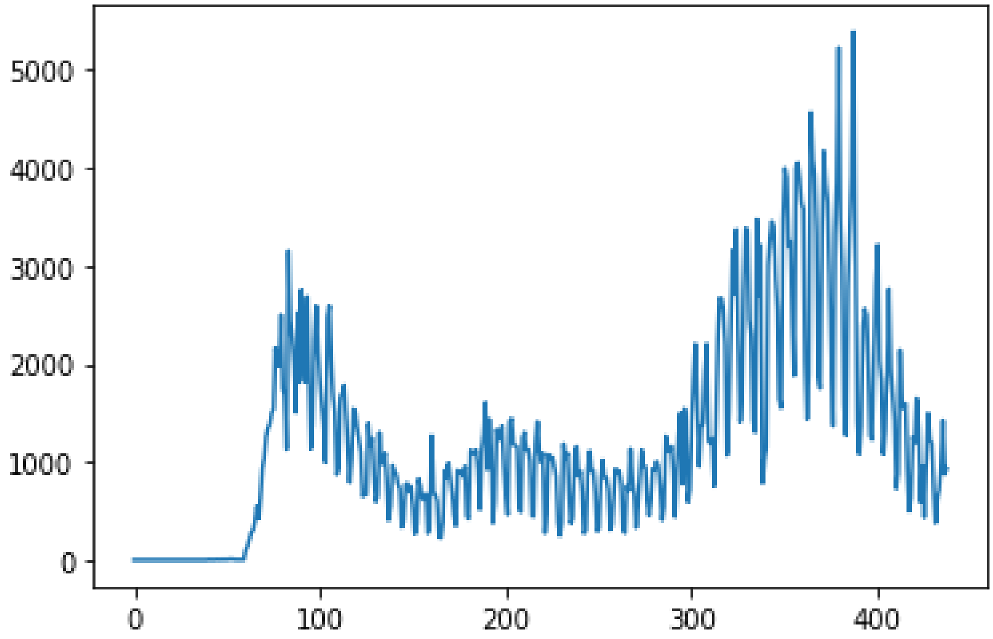
```
[<matplotlib.lines.Line2D at 0x7f4f21266290>]
```



[11]:
```
death = tukey(death)
plt.plot(death)
```

```
outliers =  2
outliers =  6
outliers =  0
outliers =  0
outliers =  0
outliers =  1
outliers =  0
outliers =  0
outliers =  0
outliers =  0
outliers =  0
outliers =  0
outliers =  0
outliers =  0
outliers =  0
```

[11]:
```
[<matplotlib.lines.Line2D at 0x7f4f21191b10>]
```



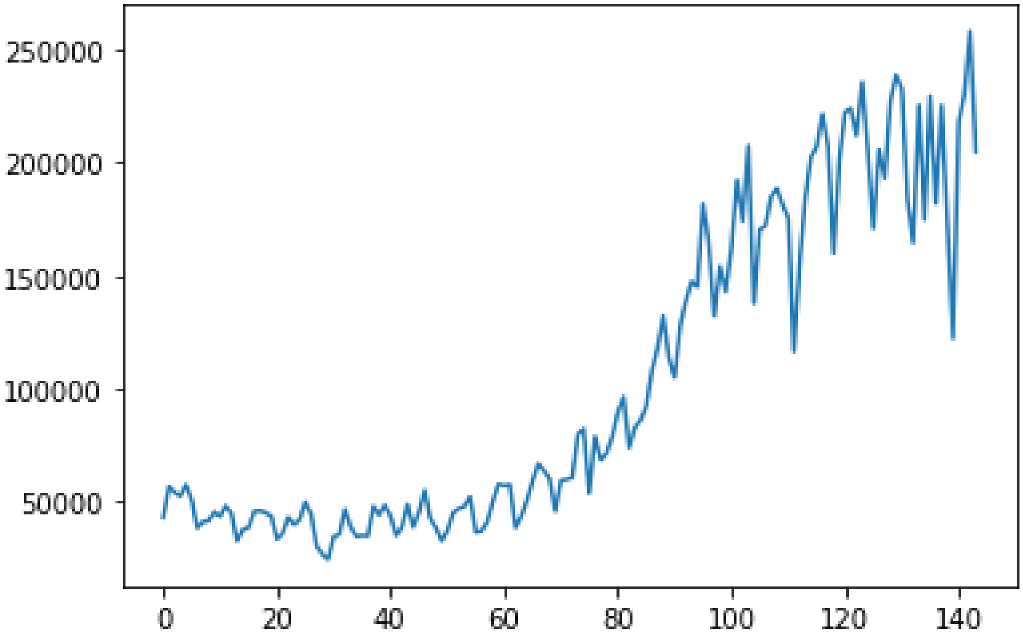
```
[43]: dict = {'Date': date, 'Cases': [int(i) for i in cases], 'Death': [int(i) for i␣
       ↪in death]}
      df1 = pd.DataFrame(dict)
```

```
[44]: df1.to_csv('USA_clean.csv')
```

```
[45]: start = datetime.datetime(2020, 8, 10)
      end = datetime.datetime(2020, 12, 31)
      df_clean = pd.read_csv('USA_clean.csv')

      def get_data(start, end, df_clean):
        cases = [int(df_clean['Cases'][i]) for i in range(0, len(df_clean['Date']))␣
       ↪if dt.strptime(df_clean['Date'][i], "%Y-%m-%d")>=start and dt.
       ↪strptime(df_clean['Date'][i], "%Y-%m-%d")<=end]
        death = [int(df_clean['Death'][i]) for i in range(0, len(df_clean['Date']))␣
       ↪if dt.strptime(df_clean['Date'][i], "%Y-%m-%d")>=start and dt.
       ↪strptime(df_clean['Date'][i], "%Y-%m-%d")<=end]
        # MT_daily_death = [int(df_clean['MT daily death'][i]) for i in range(0,␣
       ↪len(df_clean['Date'])) if dt.strptime(df_clean['Date'][i], "%m/%d/
       ↪%Y")>=start and dt.strptime(df_clean['Date'][i], "%m/%d/%Y")<=end]
        # NC_daily_death = [int(df_clean['NC daily death'][i]) for i in range(0,␣
       ↪len(df_clean['Date'])) if dt.strptime(df_clean['Date'][i], "%m/%d/
       ↪%Y")>=start and dt.strptime(df_clean['Date'][i], "%m/%d/%Y")<=end]
        return cases, death
```

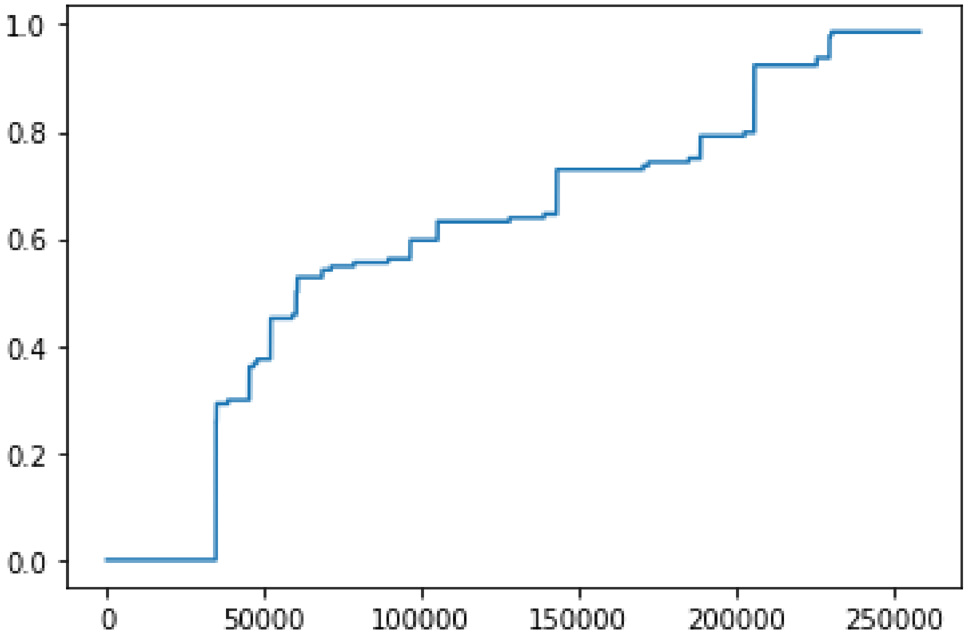
```
cases, death = get_data(start, end, df_clean)
```

[50]:
```
plt.plot(cases)
y1 = []
total=0
for x in cases:
  total += x
  y1.append(total)
# y1 = [y1[i]/total for i in range(len(y1))]
# print(y1[])
# plt.plot(y1)
```



[46]:
```
ecdf = np.ones(max(cases))
for x in range(max(cases)):
  ecdf[x] = np.searchsorted(cases,x,side='right')/len(cases)
plt.plot(ecdf)
```

[46]:
```
[<matplotlib.lines.Line2D at 0x7f4f1c408890>]
```


1.0
0.8
0.6
0.4
0.2
0.0
0
50000
100000
150000
200000
250000

```
[ ]:
```