TeamJBA_Project_Stage_II

March 14, 2023

[204]:
```
import pandas as pd
import numpy as np
```

0.0.1 USA Data Manipulation for country

[205]:
```
superdata = pd.read_csv('covid19_superdata.csv')
superdata.head(3)
```

[205]:
```
   countyFIPS            County Name State  StateFIPS  2020-01-22_x  \
0           0  statewide unallocated    AL          1             0
1        1001         autauga county    AL          1             0
2        1003         baldwin county    AL          1             0

   2020-01-23_x  2020-01-24_x  2020-01-25_x  2020-01-26_x  2020-01-27_x  …  \
0             0             0             0             0             0  …
1             0             0             0             0             0  …
2             0             0             0             0             0  …

   2023-01-28_y  2023-01-29_y  2023-01-30_y  2023-01-31_y  2023-02-01_y  \
0             0             0             0             0             0
1           230           230           230           230           230
2           723           723           723           723           723

   2023-02-02_y  2023-02-03_y  2023-02-04_y  2023-02-05_y  population
0             0             0             0             0           0
1           230           230           230           230       55869
2           723           723           723           723      223234

[3 rows x 2227 columns]
```

[206]:
```
superdata.iloc[:,865:1079]
```

[206]:
```
      2022-06-01_x  2022-06-02_x  2022-06-03_x  2022-06-04_x  2022-06-05_x  \
0                0             0             0             0             0
1            15969         15978         15978         15978         15978
2            56580         56648         56648         56648         56648
3             5710          5714          5714          5714          5714
```

```
4              6508          6512          6512          6512          6512
…              …             …             …             …             …
3184          11178         11178         11178         11178         11178
3185          10229         10229         10229         10229         10229
3186           5681          5681          5681          5681          5681
3187           2369          2369          2369          2369          2369
3188           1594          1594          1594          1594          1594

      2022-06-06_x  2022-06-07_x  2022-06-08_x  2022-06-09_x  2022-06-10_x  \
0                0             0             0             0             0
1            16032         16052         16065         16084         16095
2            56895         56955         57024         57079         57166
3             5719          5733          5734          5744          5748
4             6534          6535          6540          6544          6547
…              …             …             …             …             …
3184         11178         11178         11178         11234         11234
3185         10229         10229         10229         10403         10403
3186          5681          5681          5681          5702          5702
3187          2369          2369          2369          2371          2371
3188          1594          1594          1594          1604          1604

      …  2022-12-22_x  2022-12-23_x  2022-12-24_x  2022-12-25_x  \
0     …             0             0             0             0
1     …         18961         18961         18961         18961
2     …         67496         67496         67496         67496
3     …          7027          7027          7027          7027
4     …          7692          7692          7692          7692
…   …            …             …             …             …
3184  …         12394         12394         12394         12394
3185  …         11997         11997         11997         11997
3186  …          6303          6303          6303          6303
3187  …          2717          2717          2717          2717
3188  …          1876          1876          1876          1876

      2022-12-26_x  2022-12-27_x  2022-12-28_x  2022-12-29_x  2022-12-30_x  \
0                0             0             0             0             0
1            18961         18961         18961         18961         18961
2            67496         67496         67496         67496         67496
3             7027          7027          7027          7027          7027
4             7692          7692          7692          7692          7692
…              …             …             …             …             …
3184         12394         12394         12394         12394         12394
3185         11997         11997         11997         11997         11997
3186          6303          6303          6303          6303          6303
3187          2717          2717          2717          2717          2717
3188          1876          1876          1876          1876          1876
```

```
      2022-12-31_x
0                0
1            18961
2            67496
3             7027
4             7692
…                …
3184         12394
3185         11997
3186          6303
3187          2717
3188          1876

[3189 rows x 214 columns]
```

[207]:
```
superdata.iloc[:,1976:2190]
```

[207]:
```
      2022-06-01_y  2022-06-02_y  2022-06-03_y  2022-06-04_y  2022-06-05_y  \
0                0             0             0             0             0
1              216           216           216           216           216
2              683           683           683           683           683
3               99            99            99            99            99
4              105           105           105           105           105
…                …             …             …             …             …
3184           126           126           126           126           126
3185            16            16            16            16            16
3186            39            39            39            39            39
3187            44            44            44            44            44
3188            18            18            18            18            18

      2022-06-06_y  2022-06-07_y  2022-06-08_y  2022-06-09_y  2022-06-10_y  \
0                0             0             0             0             0
1              216           216           216           217           217
2              683           683           683           683           683
3               99            99            99            99            99
4              105           105           105           105           105
…                …             …             …             …             …
3184           126           126           126           126           126
3185            16            16            16            16            16
3186            39            39            39            39            39
3187            44            44            44            44            44
3188            18            18            18            19            19

      …  2022-12-22_y  2022-12-23_y  2022-12-24_y  2022-12-25_y  \
0     …             0             0             0             0
1     …           230           230           230           230
2     …           719           719           719           719
```

```
3      …          103           103           103           103
4      …          108           108           108           108
…    …           …            …            …            …
3184   …          136           136           136           136
3185   …           16            16            16            16
3186   …           43            43            43            43
3187   …           47            47            47            47
3188   …           22            22            22            22

      2022-12-26_y  2022-12-27_y  2022-12-28_y  2022-12-29_y  2022-12-30_y  \
0                0             0             0             0             0
1              230           230           230           230           230
2              719           719           719           719           719
3              103           103           103           103           103
4              108           108           108           108           108
…              …            …            …            …            …
3184           136           136           136           136           136
3185            16            16            16            16            16
3186            43            43            43            43            43
3187            47            47            47            47            47
3188            22            22            22            22            22

      2022-12-31_y
0                0
1              230
2              719
3              103
4              108
…              …
3184           136
3185            16
3186            43
3187            47
3188            22

[3189 rows x 214 columns]
```

[208]:
```
#parenthesis because its a paramater inside the function/method, square because␣
↪it's a list
#including axis = 1 here so that they are concat on the vertical way
df = pd.concat([superdata.iloc[:, 0:4],superdata.iloc[:,865:1079],
                superdata.iloc[:,1976:2190],
               superdata.iloc[:, -1]], axis=1)
df.head()
```

[208]:
```
   countyFIPS            County Name State  StateFIPS  2022-06-01_x  \
0           0  statewide unallocated    AL          1             0
```

```
1         1001          autauga county    AL           1         15969
2         1003          baldwin county    AL           1         56580
3         1005          barbour county    AL           1          5710
4         1007             bibb county    AL           1          6508

   2022-06-02_x  2022-06-03_x  2022-06-04_x  2022-06-05_x  2022-06-06_x  …  \
0             0             0             0             0             0  …
1         15978         15978         15978         15978         16032  …
2         56648         56648         56648         56648         56895  …
3          5714          5714          5714          5714          5719  …
4          6512          6512          6512          6512          6534  …

   2022-12-23_y  2022-12-24_y  2022-12-25_y  2022-12-26_y  2022-12-27_y  \
0             0             0             0             0             0
1           230           230           230           230           230
2           719           719           719           719           719
3           103           103           103           103           103
4           108           108           108           108           108

   2022-12-28_y  2022-12-29_y  2022-12-30_y  2022-12-31_y  population
0             0             0             0             0           0
1           230           230           230           230       55869
2           719           719           719           719      223234
3           103           103           103           103       24686
4           108           108           108           108       22394

[5 rows x 433 columns]
```

[209]:
```
df.shape
```

[209]:
```
(3189, 433)
```

[210]:
```
def county_state(x):
    return x[0] +"_"+ x[1]
```

[211]:
```
#.apply is iterating across the horizontal axis
df["county_state"] = df[["County Name", "State"]].apply(county_state, axis=1)
```

[212]:
```
#value_counts here is counting the times each value category is appearing, this␣
↪gives a unique in
df["county_state"].value_counts()
```

[212]:
```
statewide unallocated_AL    1
licking county_OH           1
lorain county_OH            1
lucas county_OH             1
```

```
madison county_OH           1
                           ..
johnson county_KY           1
kenton county_KY            1
knott county_KY             1
knox county_KY              1
weston county_KY            1
Name: county_state, Length: 3189, dtype: int64
```

[213]:
```
#X is confirmed cases
#Y is deaths

#transpose this: set the index value = statename
```

[214]:
```
df.index=df["county_state"]
```

[215]:
```
#transpose capital T here, making a copy here so they don't share a same place␣
↪in memory
superdataT = df.T.copy()
```

[216]:
```
superdataT.head()
```

[216]:
```
county_state statewide unallocated_AL autauga county_AL baldwin county_AL  \
countyFIPS                          0              1001              1003
County Name     statewide unallocated    autauga county    baldwin county
State                              AL                AL                AL
StateFIPS                           1                 1                 1
2022-06-01_x                        0             15969             56580

county_state barbour county_AL bibb county_AL blount county_AL  \
countyFIPS                1005           1007             1009
County Name     barbour county    bibb county    blount county
State                       AL             AL               AL
StateFIPS                    1              1                1
2022-06-01_x              5710           6508            15077

county_state bullock county_AL butler county_AL calhoun county_AL  \
countyFIPS                1011             1013              1015
County Name     bullock county    butler county    calhoun county
State                       AL               AL                AL
StateFIPS                    1                1                 1
2022-06-01_x              2337             5091             32596

county_state chambers county_AL  … niobrara county_WY park county_WY  \
countyFIPS                 1017  …              56027          56029
County Name     chambers county  …    niobrara county    park county
State                        AL  …                 WY             WY
```

```
StateFIPS                        1  …                 56              56
2022-06-01_x                  8551  …                708            6871

county_state platte county_WY sheridan county_WY sublette county_WY  \
countyFIPS              56031              56033              56035
County Name     platte county    sheridan county    sublette county
State                      WY                 WY                 WY
StateFIPS                  56                 56                 56
2022-06-01_x             1929               8150               1936

county_state sweetwater county_WY teton county_WY uinta county_WY  \
countyFIPS                  56037           56039           56041
County Name     sweetwater county    teton county    uinta county
State                          WY              WY              WY
StateFIPS                      56              56              56
2022-06-01_x                11178           10229            5681

county_state washakie county_WY weston county_WY
countyFIPS                56043            56045
County Name     washakie county    weston county
State                        WY               WY
StateFIPS                    56               56
2022-06-01_x               2369             1594

[5 rows x 3189 columns]
```

[217]:
```
superdataT.index
```

[217]:
```
Index(['countyFIPS', 'County Name', 'State', 'StateFIPS', '2022-06-01_x',
       '2022-06-02_x', '2022-06-03_x', '2022-06-04_x', '2022-06-05_x',
       '2022-06-06_x',
       …
       '2022-12-24_y', '2022-12-25_y', '2022-12-26_y', '2022-12-27_y',
       '2022-12-28_y', '2022-12-29_y', '2022-12-30_y', '2022-12-31_y',
       'population', 'county_state'],
      dtype='object', length=434)
```

[218]:
```
superdataT["Date"] = superdataT.index
```

[219]:
```
#append here is sticking an item into the last position of a list
to_remove = list(superdataT.index[:4])
to_remove.append(superdataT.index[-2])
to_remove.append(superdataT.index[-1])
to_remove
```

[219]:
```
['countyFIPS',
 'County Name',
 'State',
 'StateFIPS',
 'population',
 'county_state']
```

[220]:
```
def new_death(x):
    if x[-2:]=="_x":
        return "new"
    elif x[-2:]=="_y":
        return "death"
```

[221]:
```
superdataT["new_death"]=superdataT["Date"].apply(new_death)
```

[222]:
```
superdataT["new_death"].value_counts().index
```

[222]:
```
Index(['new', 'death'], dtype='object')
```

[223]:
```
#getting rid of other title names, cleaning/removing _x & _y

def clean_date(x,l=to_remove):
        if x in l:
            return np.nan
        else:
            return x[:-2]
```

[224]:
```
#apply here is iterating through every row, it is a recursion function, and␣
 ↪return without _x & _y
#clean_date doesn't have the parenthesis here, function has been refrenced here
#refrenced - not used in the moment, -> Ask Rob here about a function being␣
 ↪used as an attribute?
superdataT["Date"] = superdataT["Date"].apply(clean_date,l=to_remove)
```

[225]:
```
superdataT["Week"]=pd.DatetimeIndex(superdataT["Date"]).week
```

```
<ipython-input-225-c694bb34b93b>:1: FutureWarning: weekofyear and week have been
deprecated, please use DatetimeIndex.isocalendar().week instead, which returns a
Series. To exactly reproduce the behavior of week and weekofyear and return an
Index, you may call pd.Int64Index(idx.isocalendar().week)
  superdataT["Week"]=pd.DatetimeIndex(superdataT["Date"]).week
```

[226]:
```
superdataT["Week"].value_counts()
#series - left: value - right:counts of that value
#always does it from max to min count, that's why this index is out of order
```

[226]:
```
37.0    14
23.0    14
49.0    14
```

```
48.0    14
47.0    14
46.0    14
45.0    14
44.0    14
43.0    14
42.0    14
41.0    14
40.0    14
39.0    14
38.0    14
36.0    14
51.0    14
35.0    14
34.0    14
33.0    14
32.0    14
31.0    14
30.0    14
29.0    14
28.0    14
27.0    14
26.0    14
25.0    14
24.0    14
50.0    14
52.0    12
22.0    10
Name: Week, dtype: int64
```

[227]:
```
superdataT["Date"]
```

[227]:
```
countyFIPS               NaN
County Name              NaN
State                    NaN
StateFIPS                NaN
2022-06-01_x    2022-06-01
                     …
2022-12-29_y    2022-12-29
2022-12-30_y    2022-12-30
2022-12-31_y    2022-12-31
population               NaN
county_state             NaN
Name: Date, Length: 434, dtype: object
```

[228]:
```
superdataT["Date"]=pd.DatetimeIndex(superdataT["Date"])
```

```
[229]: superdataT["Date"]
```

```
[229]: countyFIPS             NaT
       County Name            NaT
       State                  NaT
       StateFIPS              NaT
       2022-06-01_x   2022-06-01
                         …
       2022-12-29_y   2022-12-29
       2022-12-30_y   2022-12-30
       2022-12-31_y   2022-12-31
       population             NaT
       county_state           NaT
       Name: Date, Length: 434, dtype: datetime64[ns]
```

```
[230]: superdataT.head()
```

```
[230]: county_state statewide unallocated_AL autaugua county_AL baldwin county_AL  \
       countyFIPS                           0              1001              1003
       County Name      statewide unallocated    autauga county    baldwin county
       State                               AL                AL                AL
       StateFIPS                            1                 1                 1
       2022-06-01_x                         0             15969             56580

       county_state barbour county_AL bibb county_AL blount county_AL  \
       countyFIPS                1005           1007             1009
       County Name     barbour county    bibb county   blount county
       State                       AL             AL               AL
       StateFIPS                    1              1                1
       2022-06-01_x              5710           6508            15077

       county_state bullock county_AL butler county_AL calhoun county_AL  \
       countyFIPS                1011             1013              1015
       County Name     bullock county    butler county    calhoun county
       State                       AL               AL                AL
       StateFIPS                    1                1                 1
       2022-06-01_x              2337             5091             32596

       county_state chambers county_AL  … sheridan county_WY sublette county_WY  \
       countyFIPS                 1017  …              56033              56035
       County Name     chambers county  …    sheridan county    sublette county
       State                        AL  …                 WY                 WY
       StateFIPS                     1  …                 56                 56
       2022-06-01_x               8551  …               8150               1936

       county_state sweetwater county_WY teton county_WY uinta county_WY  \
       countyFIPS                  56037           56039           56041
```

```
County Name     sweetwater county    teton county    uinta county
State                          WY              WY              WY
StateFIPS                      56              56              56
2022-06-01_x                11178           10229            5681

county_state washakie county_WY weston county_WY       Date new_death  Week
countyFIPS                56043            56045        NaT      None   NaN
County Name     washakie county    weston county        NaT      None   NaN
State                        WY               WY        NaT      None   NaN
StateFIPS                    56               56        NaT      None   NaN
2022-06-01_x               2369             1594 2022-06-01       new  22.0

[5 rows x 3192 columns]
```

[231]:
```
#when there's no parenthesis, you simply refrence it, when you add parenthesis␣
↪you are actually using it
#when you add parenthesis, it is boolean function
superdataT["Date"].notnull()
```

[231]:
```
countyFIPS      False
County Name     False
State           False
StateFIPS       False
2022-06-01_x     True
                ...
2022-12-29_y     True
2022-12-30_y     True
2022-12-31_y     True
population      False
county_state    False
Name: Date, Length: 434, dtype: bool
```

[232]:
```
#logical indexing because what is inside the first set of square brackets is a␣
↪boolean list
#I'm selecting all the true values here
superdataT=superdataT[superdataT["Date"].notnull()]
```

[233]:
```
superdataT.head()
```

[233]:
```
county_state statewide unallocated_AL autaugа county_AL baldwin county_AL  \
2022-06-01_x                        0             15969              56580
2022-06-02_x                        0             15978              56648
2022-06-03_x                        0             15978              56648
2022-06-04_x                        0             15978              56648
2022-06-05_x                        0             15978              56648

county_state barbour county_AL bibb county_AL blount county_AL  \
2022-06-01_x                5710          6508            15077
2022-06-02_x                5714          6512            15084
2022-06-03_x                5714          6512            15084
2022-06-04_x                5714          6512            15084
2022-06-05_x                5714          6512            15084

county_state bullock county_AL butler county_AL calhoun county_AL  \
2022-06-01_x                2337              5091              32596
2022-06-02_x                2337              5094              32604
2022-06-03_x                2337              5094              32604
2022-06-04_x                2337              5094              32604
2022-06-05_x                2337              5094              32604

county_state chambers county_AL  … sheridan county_WY sublette county_WY  \
2022-06-01_x                8551  …                8150               1936
2022-06-02_x                8553  …                8150               1936
2022-06-03_x                8553  …                8150               1936
2022-06-04_x                8553  …                8150               1936
2022-06-05_x                8553  …                8150               1936

county_state sweetwater county_WY teton county_WY uinta county_WY  \
2022-06-01_x                 11178           10229            5681
2022-06-02_x                 11178           10229            5681
2022-06-03_x                 11178           10229            5681
2022-06-04_x                 11178           10229            5681
2022-06-05_x                 11178           10229            5681

county_state washakie county_WY weston county_WY       Date new_death  Week
2022-06-01_x                2369             1594 2022-06-01       new  22.0
2022-06-02_x                2369             1594 2022-06-02       new  22.0
2022-06-03_x                2369             1594 2022-06-03       new  22.0
2022-06-04_x                2369             1594 2022-06-04       new  22.0
2022-06-05_x                2369             1594 2022-06-05       new  22.0

[5 rows x 3192 columns]
```

[234]:
```
superdataT.index = superdataT["Date"]
```

[235]:
```
superdataT.head()
```

[235]:
```
county_state statewide unallocated_AL autauga county_AL baldwin county_AL  \
Date
2022-06-01                         0            15969             56580
2022-06-02                         0            15978             56648
2022-06-03                         0            15978             56648
2022-06-04                         0            15978             56648
2022-06-05                         0            15978             56648
```

```
county_state barbour county_AL bibb county_AL blount county_AL  \
Date
2022-06-01                5710           6508             15077
2022-06-02                5714           6512             15084
2022-06-03                5714           6512             15084
2022-06-04                5714           6512             15084
2022-06-05                5714           6512             15084

county_state bullock county_AL butler county_AL calhoun county_AL  \
Date
2022-06-01                2337             5091              32596
2022-06-02                2337             5094              32604
2022-06-03                2337             5094              32604
2022-06-04                2337             5094              32604
2022-06-05                2337             5094              32604

county_state chambers county_AL  … sheridan county_WY sublette county_WY  \
Date                             …
2022-06-01                 8551  …              8150               1936
2022-06-02                 8553  …              8150               1936
2022-06-03                 8553  …              8150               1936
2022-06-04                 8553  …              8150               1936
2022-06-05                 8553  …              8150               1936

county_state sweetwater county_WY teton county_WY uinta county_WY  \
Date
2022-06-01                  11178           10229             5681
2022-06-02                  11178           10229             5681
2022-06-03                  11178           10229             5681
2022-06-04                  11178           10229             5681
2022-06-05                  11178           10229             5681

county_state washakie county_WY weston county_WY       Date new_death  Week
Date
2022-06-01                 2369             1594 2022-06-01       new  22.0
2022-06-02                 2369             1594 2022-06-02       new  22.0
2022-06-03                 2369             1594 2022-06-03       new  22.0
2022-06-04                 2369             1594 2022-06-04       new  22.0
2022-06-05                 2369             1594 2022-06-05       new  22.0

[5 rows x 3192 columns]
```

1 Q1 - Compare the weekly statistics (mean, median, mode) for number of new deaths across US.

[236]:
```
#: all columns, sum accorss horizontally rows
superdataT["Total"]=superdataT.iloc[:,:-3].sum(axis=1)
```

[237]:
```
superdataT.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 428 entries, 2022-06-01 to 2022-12-31
Columns: 3193 entries, statewide unallocated_AL to Total
dtypes: datetime64[ns](1), float64(2), object(3190)
memory usage: 10.4+ MB
```

[238]:
```
superdataT.columns[:-3]
```

[238]:
```
Index(['statewide unallocated_AL', 'autauga county_AL', 'baldwin county_AL',
       'barbour county_AL', 'bibb county_AL', 'blount county_AL',
       'bullock county_AL', 'butler county_AL', 'calhoun county_AL',
       'chambers county_AL',
       …
       'park county_WY', 'platte county_WY', 'sheridan county_WY',
       'sublette county_WY', 'sweetwater county_WY', 'teton county_WY',
       'uinta county_WY', 'washakie county_WY', 'weston county_WY', 'Date'],
      dtype='object', name='county_state', length=3190)
```

[239]:
```
for feat in superdataT.columns[:-4]:
    superdataT[feat]=superdataT[feat].astype("int")
```

[240]:
```
superdataT.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 428 entries, 2022-06-01 to 2022-12-31
Columns: 3193 entries, statewide unallocated_AL to Total
dtypes: datetime64[ns](1), float64(2), int64(3189), object(1)
memory usage: 10.4+ MB
```

[241]:
```
#rows first, all columns up until 3
superdataT["Total"]=superdataT.iloc[:,:-4].sum(axis=1)
```

[242]:
```
superdataT["new_death"].isnull().sum()
```

[242]:
```
0
```

[243]:
```
superdataT.reset_index(drop=True,inplace=True)
```

[244]:
```
superdataT["new_death"]=="death"
```

[244]:
```
0      False
1      False
2      False
3      False
4      False
       …
423     True
424     True
425     True
426     True
427     True
Name: new_death, Length: 428, dtype: bool
```

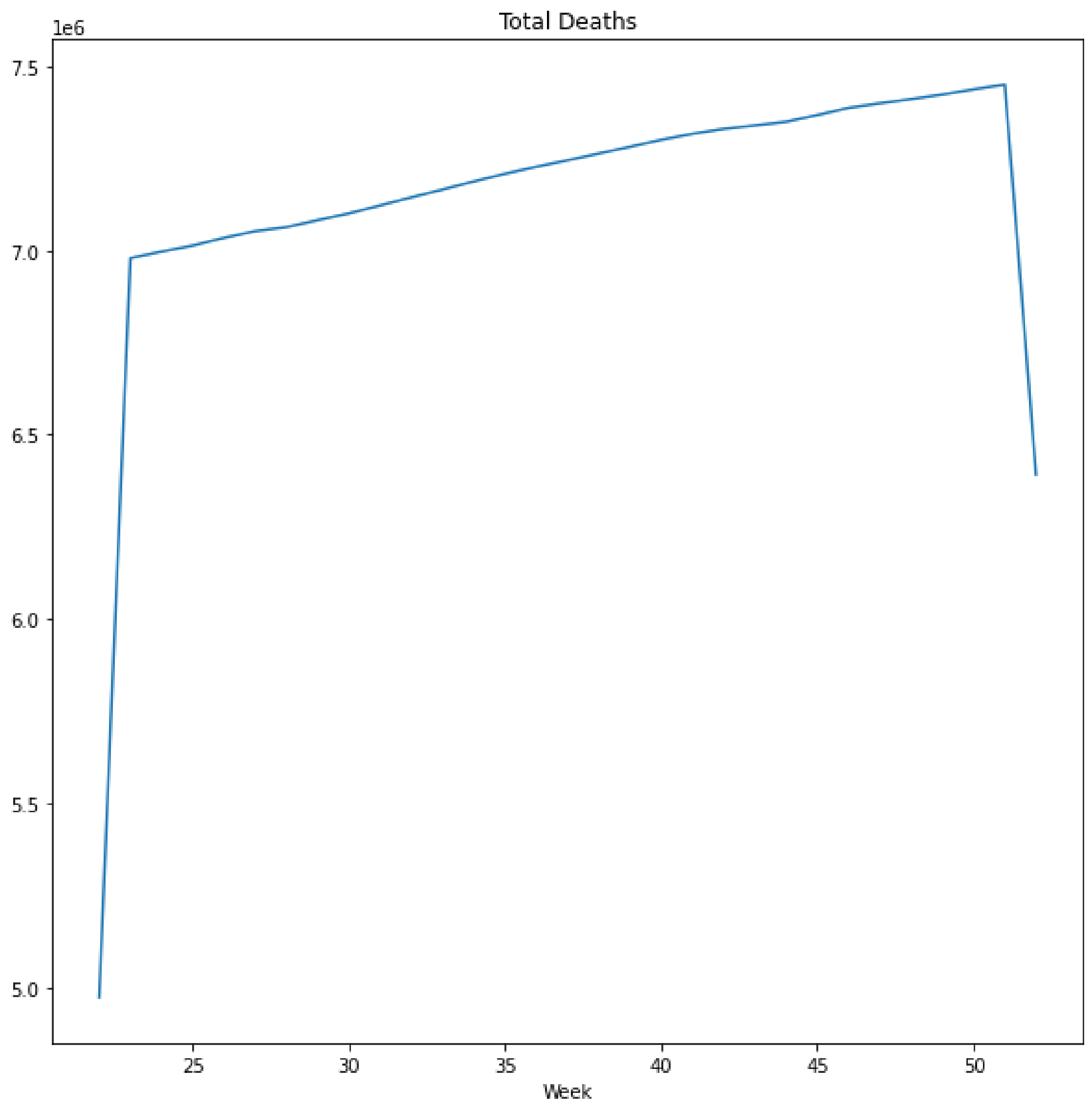
[245]:
```
superdataT.shape
```

[245]:
```
(428, 3193)
```

[246]:
```
#filtered only the ones that are new, and grouped by only the week and got a␣
↪total
superdataT[superdataT["new_death"]=="death"].groupby("Week").sum()["Total"].
↪plot(title="Total Deaths", figsize=(10,10))
```

```
<ipython-input-246-d8517f0364c8>:2: FutureWarning: The default value of
numeric_only in DataFrameGroupBy.sum is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only
columns which should be valid for the function.
  superdataT[superdataT["new_death"]=="death"].groupby("Week").sum()["Total"].pl
ot(title="Total Deaths", figsize=(10,10))
```

[246]:
```
<Axes: title={'center': 'Total Deaths'}, xlabel='Week'>
```

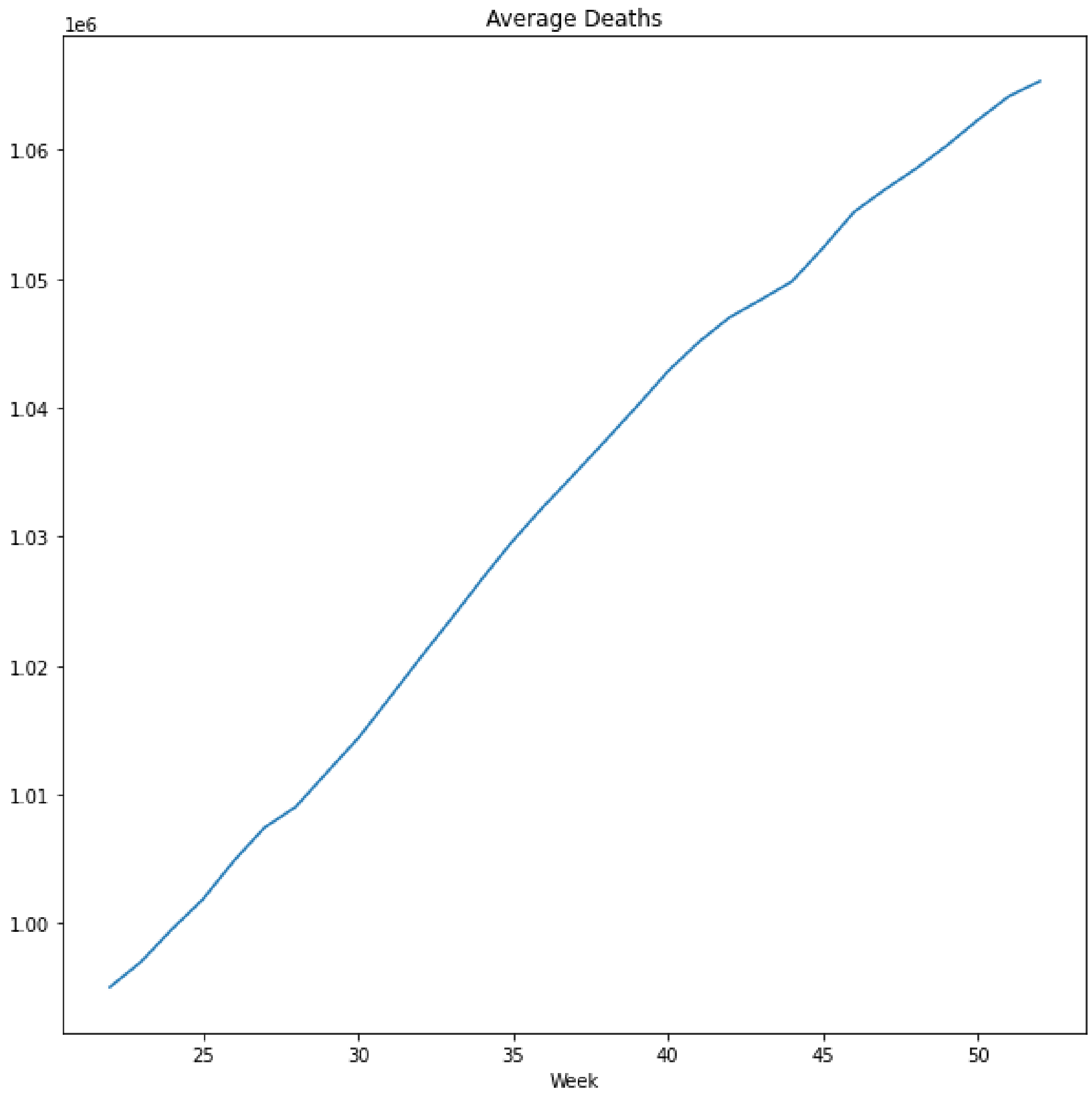


1.0.1 US (new_deaths) - median

```
[247]: superdataT[superdataT["new_death"]=="death"].groupby("Week").mean()["Total"].
        ↪plot(title="Average Deaths", figsize=(10,10))
```

```
<ipython-input-247-b169653b3b88>:1: FutureWarning: The default value of
numeric_only in DataFrameGroupBy.mean is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only
columns which should be valid for the function.
  superdataT[superdataT["new_death"]=="death"].groupby("Week").mean()["Total"].p
lot(title="Average Deaths", figsize=(10,10))
```

```
[247]: <Axes: title={'center': 'Average Deaths'}, xlabel='Week'>
```

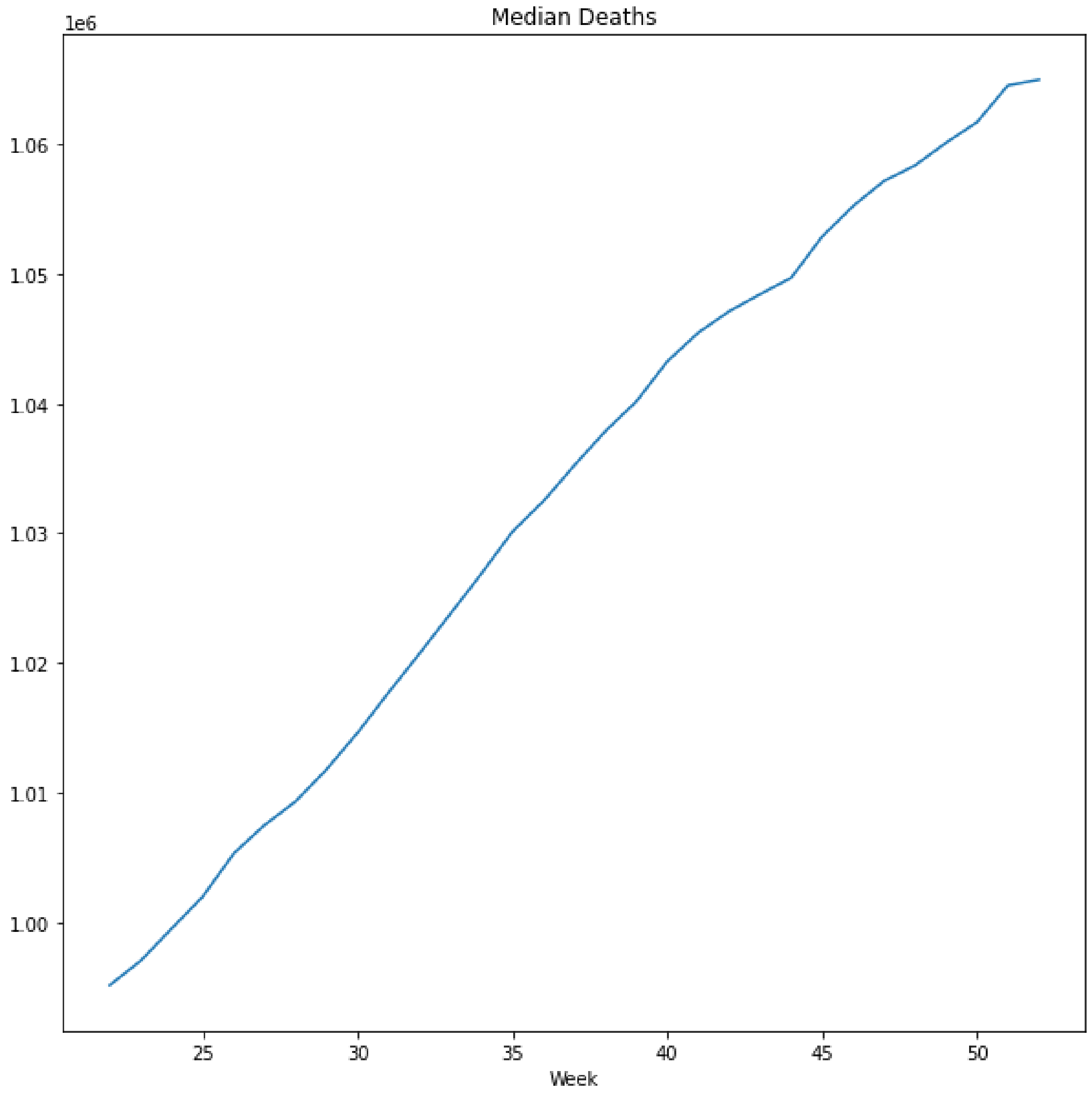


1.0.2 US (new_deaths) - mean

```
[248]: superdataT[superdataT["new_death"]=="death"].groupby("Week").median()["Total"].
        ↪plot(title="Median Deaths", figsize=(10,10))
```

```
<ipython-input-248-1f2695927164>:1: FutureWarning: The default value of
numeric_only in DataFrameGroupBy.median is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only
columns which should be valid for the function.
  superdataT[superdataT["new_death"]=="death"].groupby("Week").median()["Total"]
.plot(title="Median Deaths", figsize=(10,10))
```

```
[248]: <Axes: title={'center': 'Median Deaths'}, xlabel='Week'>
```



1.0.3 US (new_deaths) - mode

```
[249]: superdataT[superdataT["new_death"]=="death"].groupby(["Week"]).agg(lambda x:x.
        ↪value_counts().index[0])["Total"]
```

```
[249]: Week
       22.0     994605
       23.0     995529
       24.0     998197
       25.0    1000752
```

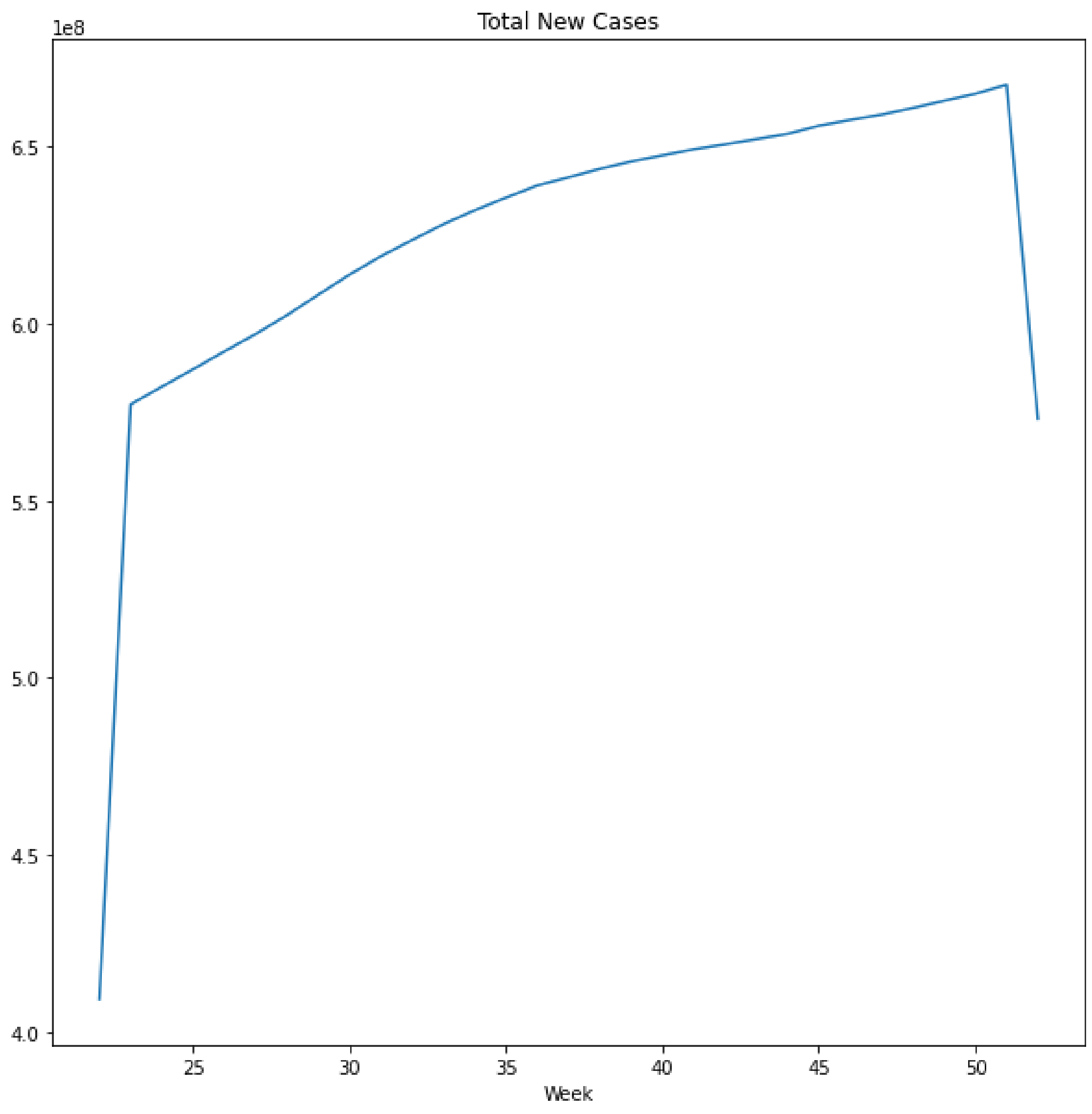
```
26.0    1002744
27.0    1006187
28.0    1007564
29.0    1010301
30.0    1012789
31.0    1015481
32.0    1018657
33.0    1021770
34.0    1024739
35.0    1028067
36.0    1031183
37.0    1033505
38.0    1036036
39.0    1038769
40.0    1041280
41.0    1043893
42.0    1046182
43.0    1047388
44.0    1049282
45.0    1053839
46.0    1055594
47.0    1057165
48.0    1059349
49.0    1061367
50.0    1063436
51.0    1064489
52.0    1064501
Name: Total, dtype: int64
```

2 Compare the weekly statistics (mean, median, mode) for number of new cases across US)

[250]:
```
superdataT[superdataT["new_death"]=="new"].groupby("Week").sum()["Total"].
 ↪plot(title="Total New Cases", figsize=(10,10))
```

```
<ipython-input-250-57e0fb381db1>:1: FutureWarning: The default value of
numeric_only in DataFrameGroupBy.sum is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only
columns which should be valid for the function.
  superdataT[superdataT["new_death"]=="new"].groupby("Week").sum()["Total"].plot
(title="Total New Cases", figsize=(10,10))
```

[250]:
```
<Axes: title={'center': 'Total New Cases'}, xlabel='Week'>
```

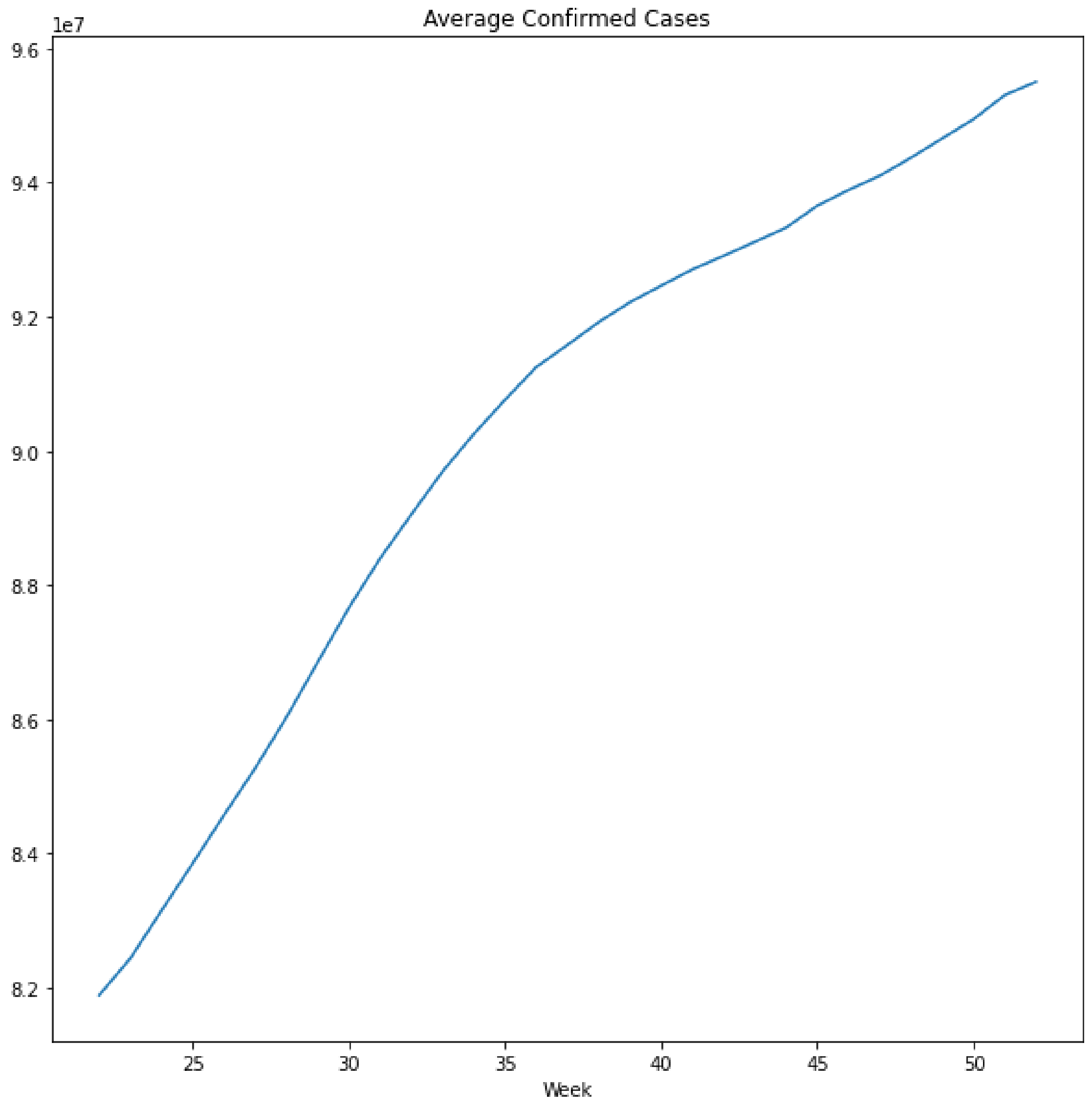


2.0.1 US (new_cases) - mean

```
[251]: superdataT[superdataT["new_death"]=="new"].groupby("Week").mean()["Total"].
        ↪plot(title="Average Confirmed Cases", figsize=(10,10))
```

```
<ipython-input-251-dacf60ed971a>:1: FutureWarning: The default value of
numeric_only in DataFrameGroupBy.mean is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only
columns which should be valid for the function.
  superdataT[superdataT["new_death"]=="new"].groupby("Week").mean()["Total"].plo
t(title="Average Confirmed Cases", figsize=(10,10))
```

```
[251]: <Axes: title={'center': 'Average Confirmed Cases'}, xlabel='Week'>
```



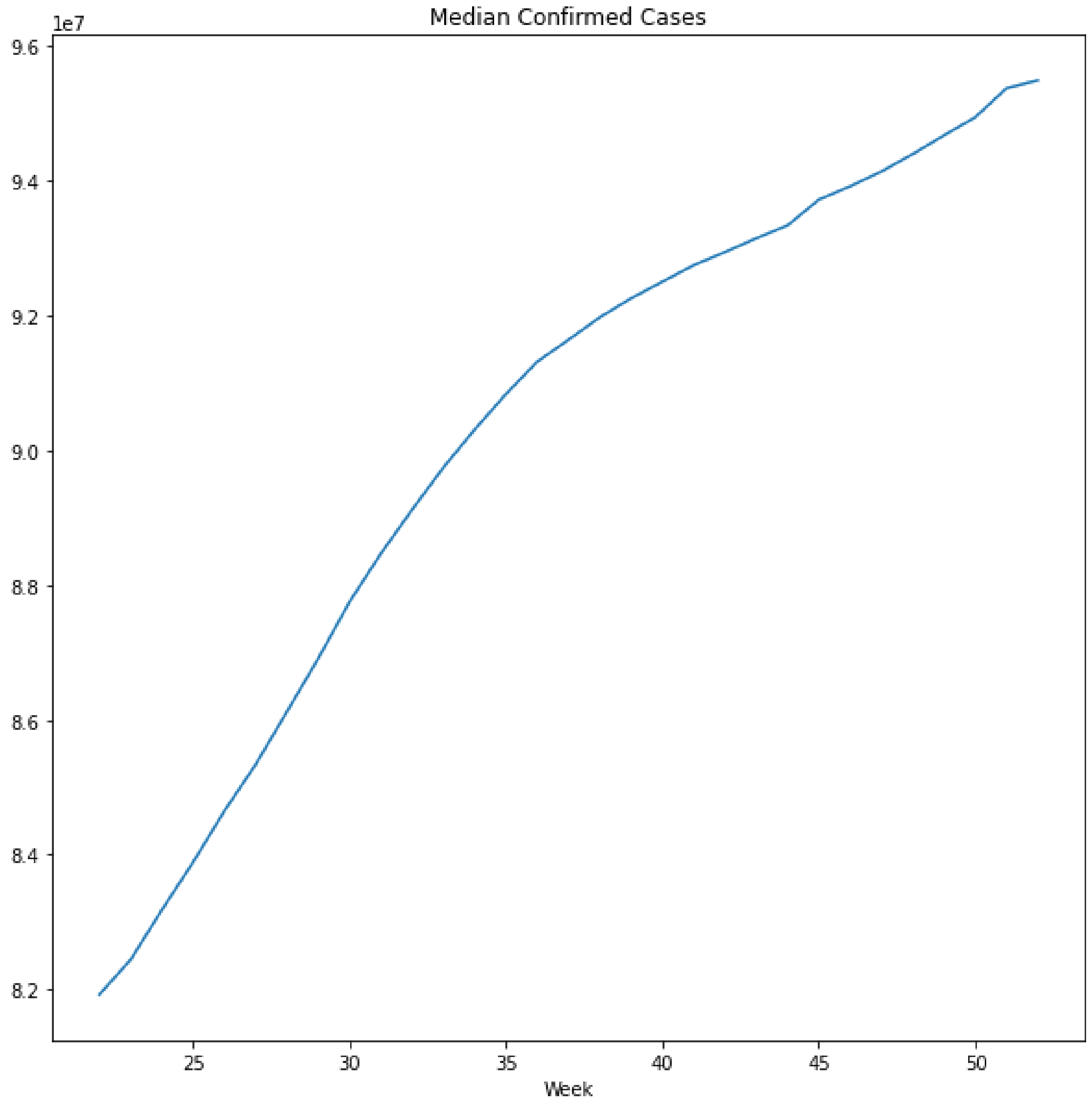
2.0.2 US (new_cases) - median

```
[252]: superdataT[superdataT["new_death"]=="new"].groupby("Week").median()["Total"].
        ↪plot(title="Median Confirmed Cases", figsize=(10,10))
```

```
<ipython-input-252-045212907803>:1: FutureWarning: The default value of
numeric_only in DataFrameGroupBy.median is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only
columns which should be valid for the function.
  superdataT[superdataT["new_death"]=="new"].groupby("Week").median()["Total"].p
```

```
lot(title="Median Confirmed Cases", figsize=(10,10))
```

```
[252]: <Axes: title={'center': 'Median Confirmed Cases'}, xlabel='Week'>
```



2.0.3 US (new_cases) - mode

```
[253]: superdataT[superdataT["new_death"]=="new"].groupby(["Week"]).agg(lambda x:x.
        ↪value_counts().index[0])["Total"]
```

```
[253]: Week
       22.0     81744241
       23.0     82153138
```

```
24.0    82784731
25.0    83507331
26.0    84205911
27.0    84880981
28.0    85668412
29.0    86453204
30.0    87274173
31.0    88049729
32.0    88716326
33.0    89407555
34.0    89969729
35.0    90544597
36.0    90995772
37.0    91404127
38.0    91747226
39.0    92093640
40.0    92334136
41.0    92585662
42.0    92823236
43.0    93028912
44.0    93211488
45.0    93442092
46.0    93791648
47.0    93976424
48.0    94458920
49.0    94764616
50.0    95083722
51.0    95122405
52.0    95391562
Name: Total, dtype: int64
```

3 Q2 Rounded Mean of BOTH New Cases & Deaths

[254]:
```
#rounding means, then creating two dataframes
#rounded mean of deaths per week = df_d
df_d=superdataT[superdataT["new_death"]=="death"].groupby("Week").
 ↪mean()["Total"].round()
```

```
<ipython-input-254-e18b089eff88>:3: FutureWarning: The default value of
numeric_only in DataFrameGroupBy.mean is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only
columns which should be valid for the function.
  df_d=superdataT[superdataT["new_death"]=="death"].groupby("Week").mean()["Tota
l"].round()
```

```
[255]: #rounded mean of new confirmed cases per week = df_n
df_n=superdataT[superdataT["new_death"]=="new"].groupby("Week").mean()["Total"].
↪round()
```

```
<ipython-input-255-20feb969d117>:2: FutureWarning: The default value of
numeric_only in DataFrameGroupBy.mean is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only
columns which should be valid for the function.
  df_n=superdataT[superdataT["new_death"]=="new"].groupby("Week").mean()["Total"
].round()
```

```
[256]: print("mean of deaths: ",df_d.mean())
print("median of deaths: ",df_d.median())
#indexing into the output of mode, it is a list here, so gotta index here
print("mode of deaths: ",df_d.mode()[1])
```

```
mean of deaths:  1032631.3870967742
median of deaths:  1034881.0
mode of deaths:  996977.0
```

```
[257]: print("mean of new confirmed cases: ",df_n.mean())
print("median of new confirmed cases: ",df_n.median())
#indexing into the output of mode, it is a list here, so gotta index here
print("mode of new confirmed cases: ",df_n.mode()[1])
```

```
mean of new confirmed cases:  90252963.25806452
median of new confirmed cases:  91582128.0
mode of new confirmed cases:  82437206.0
```

4 Comparing Countries

```
[258]: df_temp = pd.read_csv('owid-covid-data.csv')
```

```
[259]: df_Ind = df_temp[df_temp["location"]=="Indonesia"].copy()
df_Pak = df_temp[df_temp["location"]=="Pakistan"].copy()
df_Ni = df_temp[df_temp["location"]=="Nigeria"].copy()
```

```
[260]: df_Ind.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1160 entries, 118285 to 119444
Data columns (total 67 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   iso_code                                    1160 non-null   object
 1   continent                                   1160 non-null   object
 2   location                                    1160 non-null   object
```

```
3   date                                      1160 non-null   object
4   total_cases                               1101 non-null   float64
5   new_cases                                 1159 non-null   float64
6   new_cases_smoothed                        1154 non-null   float64
7   total_deaths                              1092 non-null   float64
8   new_deaths                                1160 non-null   float64
9   new_deaths_smoothed                       1155 non-null   float64
10  total_cases_per_million                   1101 non-null   float64
11  new_cases_per_million                     1159 non-null   float64
12  new_cases_smoothed_per_million            1154 non-null   float64
13  total_deaths_per_million                  1092 non-null   float64
14  new_deaths_per_million                    1160 non-null   float64
15  new_deaths_smoothed_per_million           1155 non-null   float64
16  reproduction_rate                         1023 non-null   float64
17  icu_patients                             0 non-null      float64
18  icu_patients_per_million                 0 non-null      float64
19  hosp_patients                            0 non-null      float64
20  hosp_patients_per_million                0 non-null      float64
21  weekly_icu_admissions                    0 non-null      float64
22  weekly_icu_admissions_per_million        0 non-null      float64
23  weekly_hosp_admissions                   0 non-null      float64
24  weekly_hosp_admissions_per_million       0 non-null      float64
25  total_tests                              371 non-null    float64
26  new_tests                                369 non-null    float64
27  total_tests_per_thousand                 371 non-null    float64
28  new_tests_per_thousand                   369 non-null    float64
29  new_tests_smoothed                       365 non-null    float64
30  new_tests_smoothed_per_thousand          365 non-null    float64
31  positive_rate                            365 non-null    float64
32  tests_per_case                           365 non-null    float64
33  tests_units                              372 non-null    object
34  total_vaccinations                       456 non-null    float64
35  people_vaccinated                        500 non-null    float64
36  people_fully_vaccinated                  502 non-null    float64
37  total_boosters                           121 non-null    float64
38  new_vaccinations                         409 non-null    float64
39  new_vaccinations_smoothed                735 non-null    float64
40  total_vaccinations_per_hundred           456 non-null    float64
41  people_vaccinated_per_hundred            500 non-null    float64
42  people_fully_vaccinated_per_hundred      502 non-null    float64
43  total_boosters_per_hundred               121 non-null    float64
44  new_vaccinations_smoothed_per_million    735 non-null    float64
45  new_people_vaccinated_smoothed           735 non-null    float64
46  new_people_vaccinated_smoothed_per_hundred  735 non-null    float64
47  stringency_index                          1075 non-null   float64
48  population_density                        1160 non-null   float64
49  median_age                                1160 non-null   float64
50  aged_65_older                             1160 non-null   float64
```

```
 51  aged_70_older                             1160 non-null   float64
 52  gdp_per_capita                            1160 non-null   float64
 53  extreme_poverty                           1160 non-null   float64
 54  cardiovasc_death_rate                     1160 non-null   float64
 55  diabetes_prevalence                       1160 non-null   float64
 56  female_smokers                            1160 non-null   float64
 57  male_smokers                              1160 non-null   float64
 58  handwashing_facilities                    1160 non-null   float64
 59  hospital_beds_per_thousand                1160 non-null   float64
 60  life_expectancy                           1160 non-null   float64
 61  human_development_index                   1160 non-null   float64
 62  population                                1160 non-null   float64
 63  excess_mortality_cumulative_absolute      0 non-null      float64
 64  excess_mortality_cumulative               0 non-null      float64
 65  excess_mortality                          0 non-null      float64
 66  excess_mortality_cumulative_per_million   0 non-null      float64
dtypes: float64(62), object(5)
memory usage: 616.2+ KB
```

[261]:
```
df_Ind["date"]=pd.to_datetime(df_Ind["date"])
df_Pak["date"]=pd.to_datetime(df_Pak["date"])
df_Ni["date"]=pd.to_datetime(df_Ni["date"])
```

[262]:
```
#checking to see if it's datetime
df_Ind.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1160 entries, 118285 to 119444
Data columns (total 67 columns):
 #   Column                                   Non-Null Count  Dtype
---  ------                                   --------------  -----
 0   iso_code                                 1160 non-null   object
 1   continent                                1160 non-null   object
 2   location                                 1160 non-null   object
 3   date                                     1160 non-null   datetime64[ns]
 4   total_cases                              1101 non-null   float64
 5   new_cases                                1159 non-null   float64
 6   new_cases_smoothed                       1154 non-null   float64
 7   total_deaths                             1092 non-null   float64
 8   new_deaths                               1160 non-null   float64
 9   new_deaths_smoothed                      1155 non-null   float64
 10  total_cases_per_million                  1101 non-null   float64
 11  new_cases_per_million                    1159 non-null   float64
 12  new_cases_smoothed_per_million           1154 non-null   float64
 13  total_deaths_per_million                 1092 non-null   float64
 14  new_deaths_per_million                   1160 non-null   float64
 15  new_deaths_smoothed_per_million          1155 non-null   float64
 16  reproduction_rate                        1023 non-null   float64
```

```
17  icu_patients                               0 non-null      float64
18  icu_patients_per_million                   0 non-null      float64
19  hosp_patients                              0 non-null      float64
20  hosp_patients_per_million                  0 non-null      float64
21  weekly_icu_admissions                      0 non-null      float64
22  weekly_icu_admissions_per_million          0 non-null      float64
23  weekly_hosp_admissions                     0 non-null      float64
24  weekly_hosp_admissions_per_million         0 non-null      float64
25  total_tests                                371 non-null    float64
26  new_tests                                  369 non-null    float64
27  total_tests_per_thousand                   371 non-null    float64
28  new_tests_per_thousand                     369 non-null    float64
29  new_tests_smoothed                         365 non-null    float64
30  new_tests_smoothed_per_thousand            365 non-null    float64
31  positive_rate                              365 non-null    float64
32  tests_per_case                             365 non-null    float64
33  tests_units                                372 non-null    object
34  total_vaccinations                         456 non-null    float64
35  people_vaccinated                          500 non-null    float64
36  people_fully_vaccinated                    502 non-null    float64
37  total_boosters                             121 non-null    float64
38  new_vaccinations                           409 non-null    float64
39  new_vaccinations_smoothed                  735 non-null    float64
40  total_vaccinations_per_hundred             456 non-null    float64
41  people_vaccinated_per_hundred              500 non-null    float64
42  people_fully_vaccinated_per_hundred        502 non-null    float64
43  total_boosters_per_hundred                 121 non-null    float64
44  new_vaccinations_smoothed_per_million      735 non-null    float64
45  new_people_vaccinated_smoothed             735 non-null    float64
46  new_people_vaccinated_smoothed_per_hundred 735 non-null    float64
47  stringency_index                           1075 non-null   float64
48  population_density                         1160 non-null   float64
49  median_age                                 1160 non-null   float64
50  aged_65_older                              1160 non-null   float64
51  aged_70_older                              1160 non-null   float64
52  gdp_per_capita                             1160 non-null   float64
53  extreme_poverty                            1160 non-null   float64
54  cardiovasc_death_rate                      1160 non-null   float64
55  diabetes_prevalence                        1160 non-null   float64
56  female_smokers                             1160 non-null   float64
57  male_smokers                               1160 non-null   float64
58  handwashing_facilities                     1160 non-null   float64
59  hospital_beds_per_thousand                 1160 non-null   float64
60  life_expectancy                            1160 non-null   float64
61  human_development_index                    1160 non-null   float64
62  population                                 1160 non-null   float64
63  excess_mortality_cumulative_absolute       0 non-null      float64
64  excess_mortality_cumulative                0 non-null      float64
```

```
 65  excess_mortality                           0 non-null     float64
 66  excess_mortality_cumulative_per_million    0 non-null     float64
dtypes: datetime64[ns](1), float64(62), object(4)
memory usage: 616.2+ KB
```

[263]:
```
#because its in Datetime data type
#two logic statements
df_Ind=df_Ind[(df_Ind["date"]>="2022-06-01")&(df_Ind["date"]<="2022-12-31")].
 ↪copy()
df_Pak=df_Pak[(df_Pak["date"]>="2022-06-01")&(df_Pak["date"]<="2022-12-31")].
 ↪copy()
df_Ni=df_Ni[(df_Ni["date"]>="2022-06-01")&(df_Ni["date"]<="2022-12-31")].copy()
```

[264]:
```
#asking for attribute
df_Ind["week"]=pd.DatetimeIndex(df_Ind["date"]).week
df_Pak["week"]=pd.DatetimeIndex(df_Pak["date"]).week
df_Ni["week"]=pd.DatetimeIndex(df_Ni["date"]).week
```

```
<ipython-input-264-a6a44c638604>:2: FutureWarning: weekofyear and week have been
deprecated, please use DatetimeIndex.isocalendar().week instead, which returns a
Series. To exactly reproduce the behavior of week and weekofyear and return an
Index, you may call pd.Int64Index(idx.isocalendar().week)
  df_Ind["week"]=pd.DatetimeIndex(df_Ind["date"]).week
<ipython-input-264-a6a44c638604>:3: FutureWarning: weekofyear and week have been
deprecated, please use DatetimeIndex.isocalendar().week instead, which returns a
Series. To exactly reproduce the behavior of week and weekofyear and return an
Index, you may call pd.Int64Index(idx.isocalendar().week)
  df_Pak["week"]=pd.DatetimeIndex(df_Pak["date"]).week
<ipython-input-264-a6a44c638604>:4: FutureWarning: weekofyear and week have been
deprecated, please use DatetimeIndex.isocalendar().week instead, which returns a
Series. To exactly reproduce the behavior of week and weekofyear and return an
Index, you may call pd.Int64Index(idx.isocalendar().week)
  df_Ni["week"]=pd.DatetimeIndex(df_Ni["date"]).week
```

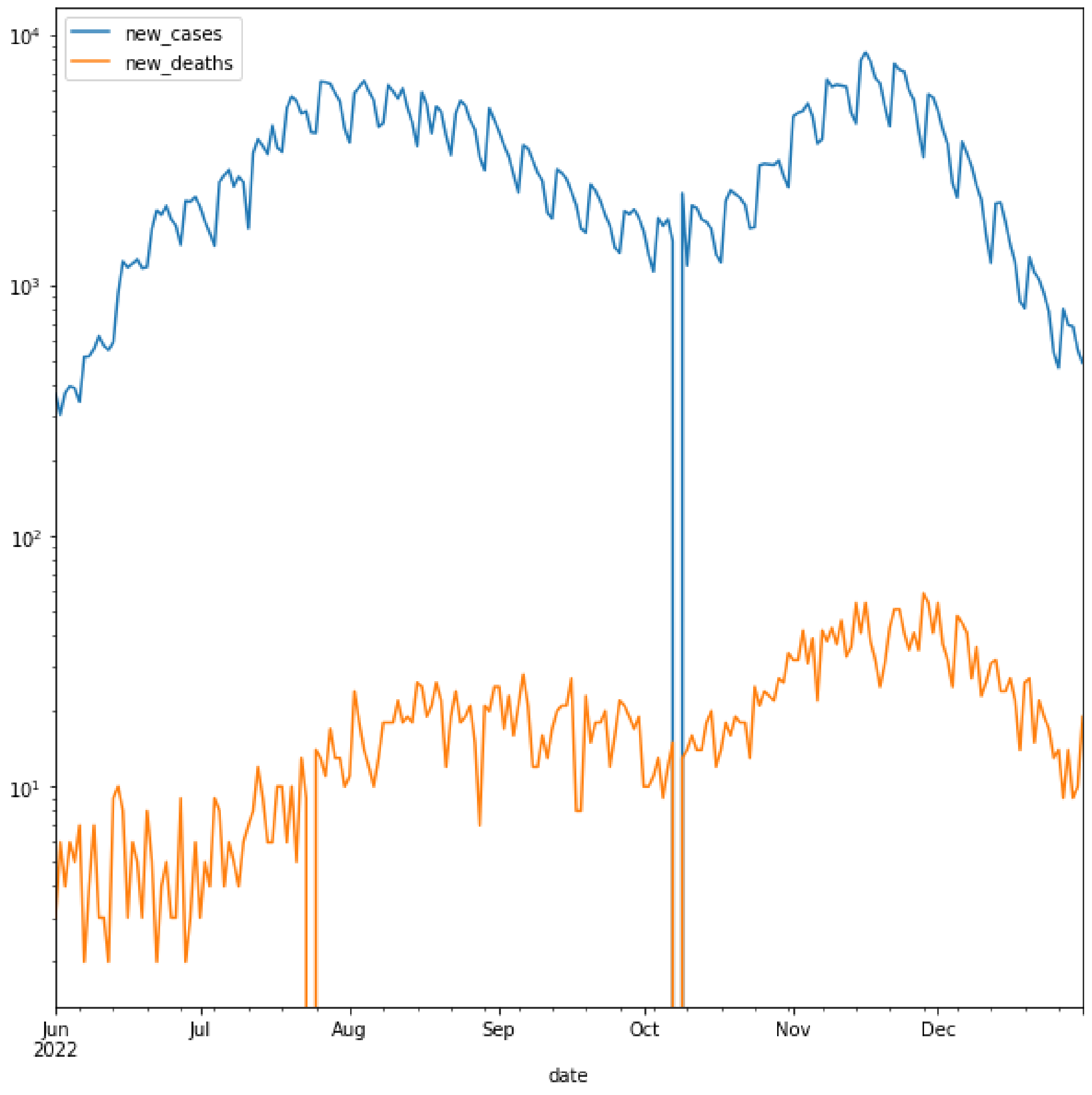
[265]:
```
df_Ind["date"]=pd.to_datetime(df_Ind["date"])
df_Pak["date"]=pd.to_datetime(df_Pak["date"])
df_Ni["date"]=pd.to_datetime(df_Ni["date"])
```

[266]:
```
#setting index to datetime
df_Ind.index=df_Ind["date"]
df_Pak.index=df_Pak["date"]
df_Ni.index=df_Ni["date"]
```

4.0.1 Observing Indonesia

```
[267]: #x-axis here is row values
       #y-axis here is log of new cases & deaths - making extreme values less extreme
       df_Ind[["new_cases","new_deaths"]].plot(logy=True, figsize=(10,10))
```

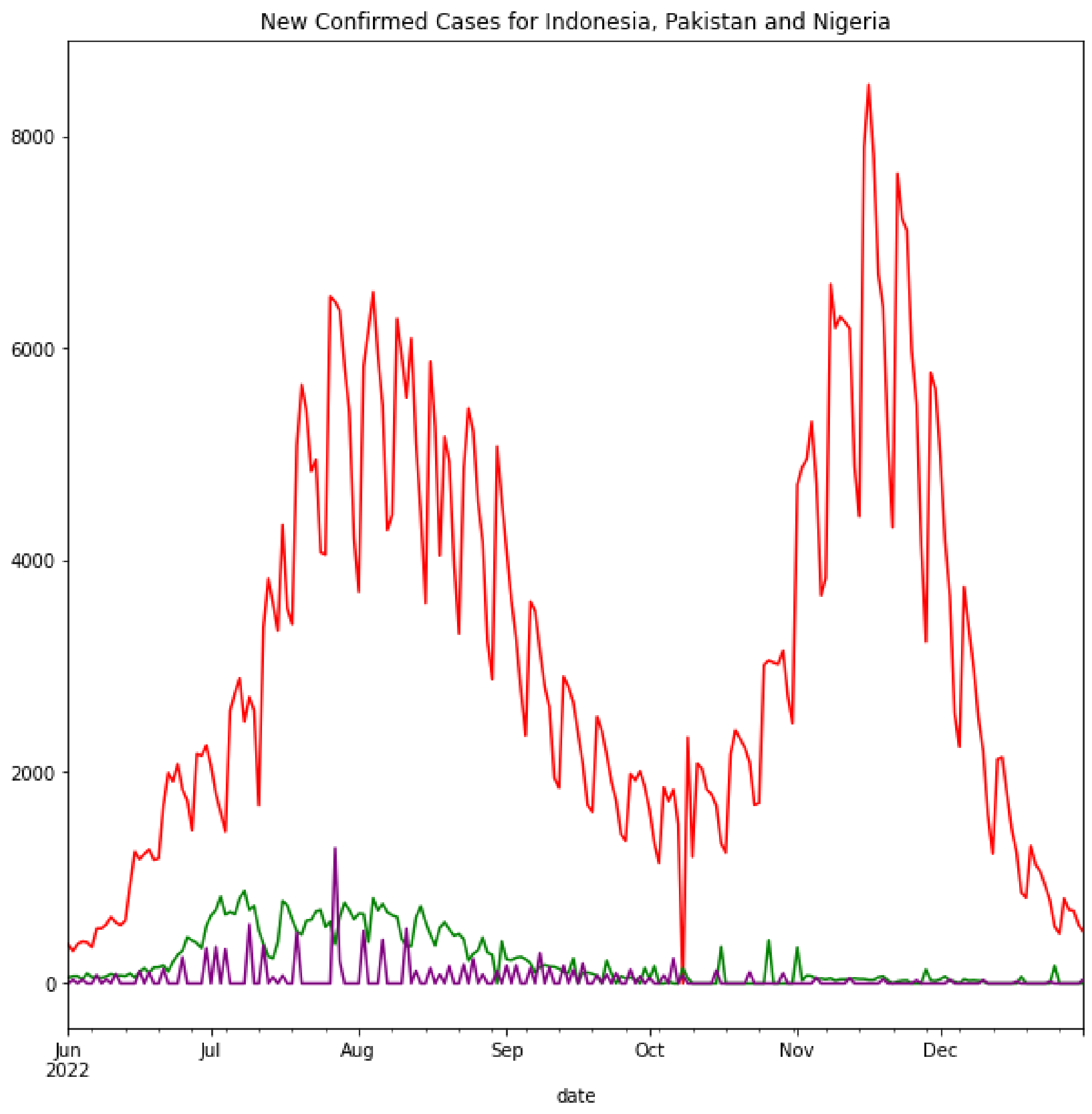
```
[267]: <Axes: xlabel='date'>
```



4.0.2 New Confirmed Cases for Indonesia, Pakistan and Nigeria - plot

[268]:
```
#subplot(in matplatlib) in pandas version as ax objects
#Indonesia is RED, Pakistan is GREEN, Nigeria is PURPLE
#ax=ax meaning here to put the Pakistan plot in ax plot
ax = df_Ind["new_cases"].plot(color="red")
df_Pak["new_cases"].plot(color="green", ax=ax)
df_Ni["new_cases"].plot(color="purple", ax=ax, title = "New Confirmed Cases for␣
↪Indonesia, Pakistan and Nigeria", figsize=(10,10))
```

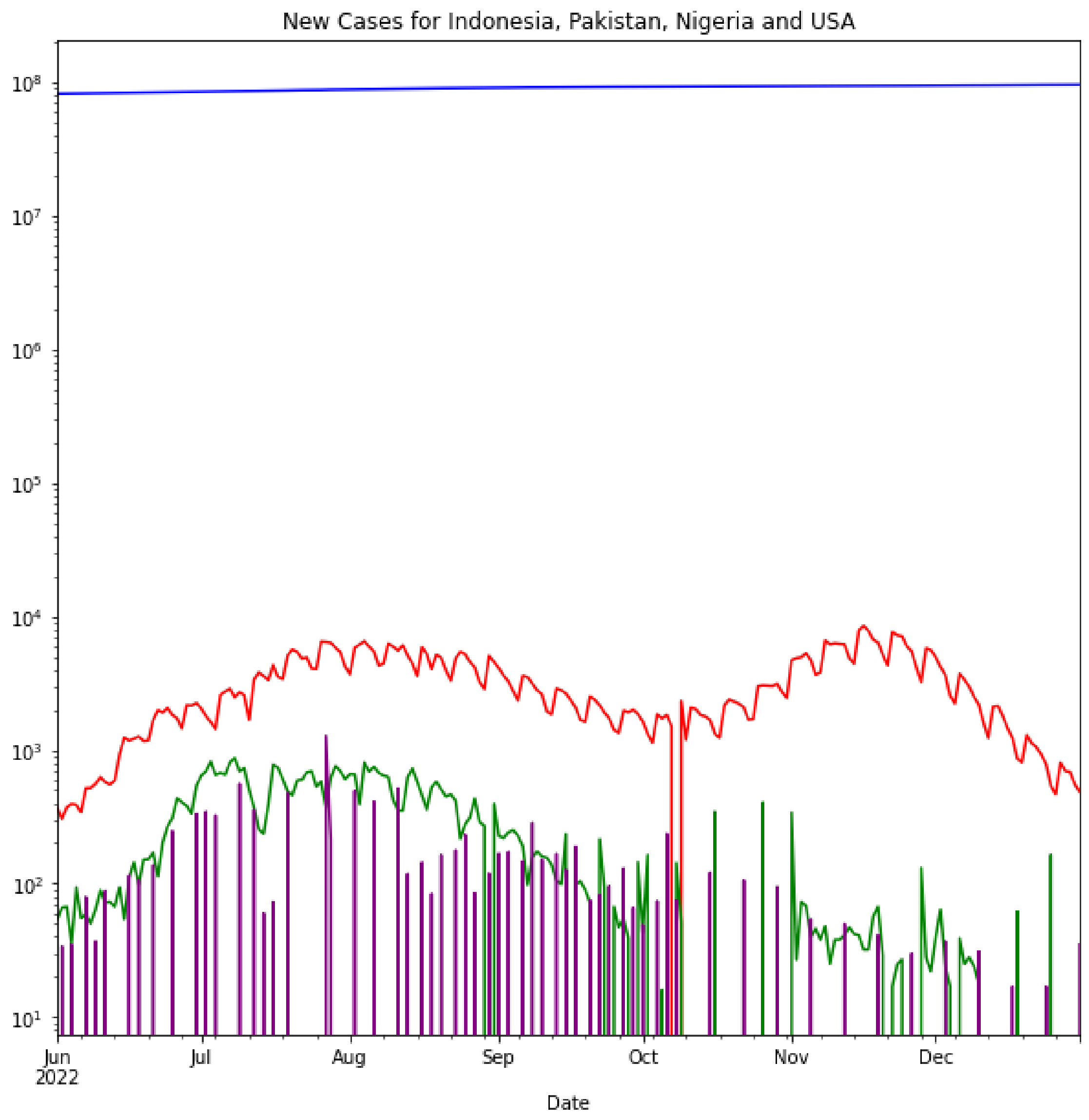
[268]:
```
<Axes: title={'center': 'New Confirmed Cases for Indonesia, Pakistan and
Nigeria'}, xlabel='date'>
```



```
[269]: #preparing superdataT for full country comparison
       superdataT.index = superdataT["Date"]
```

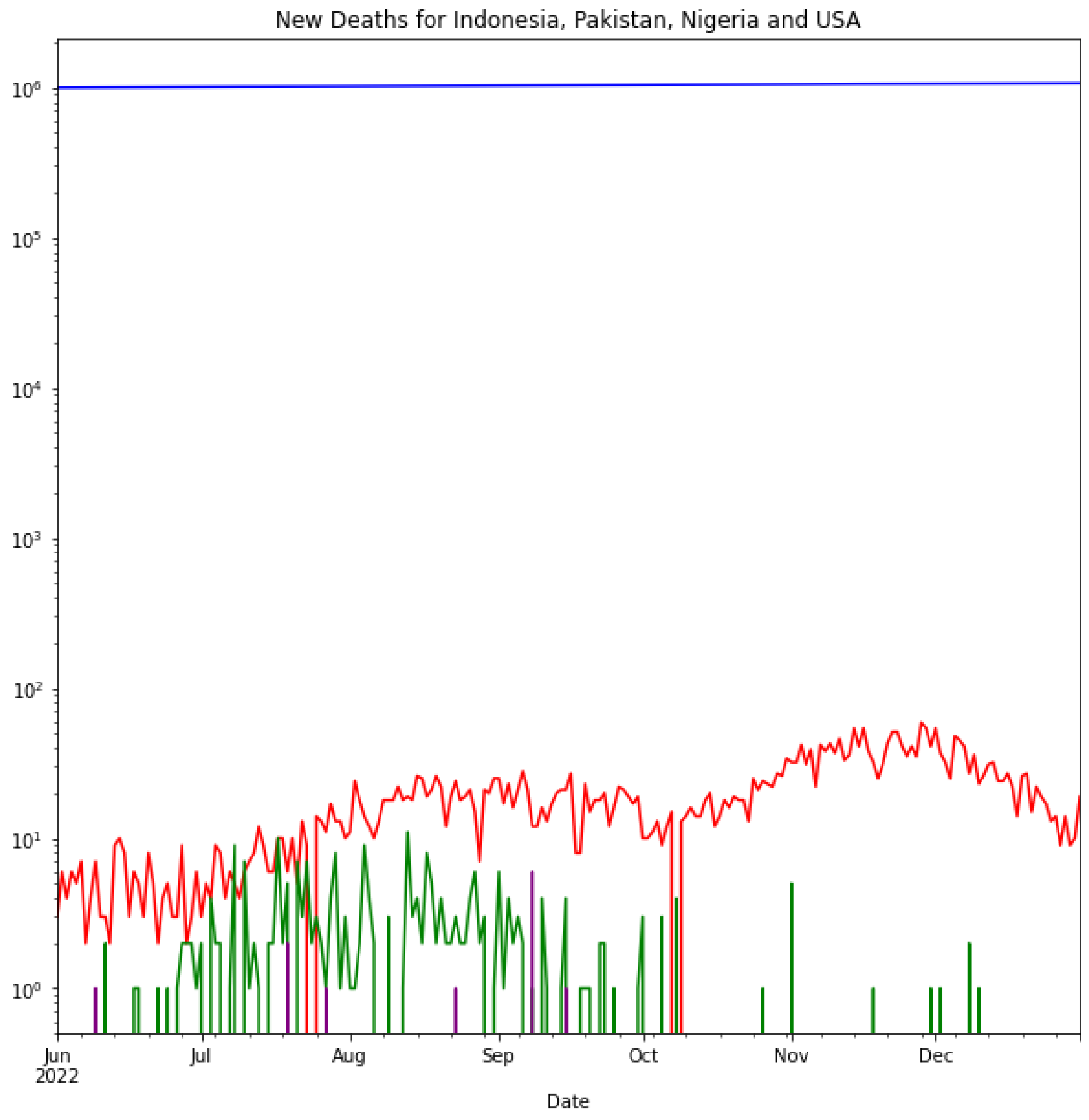
```
[270]: ax = df_Ind["new_cases"].plot(color="red", title = "New Cases for Indonesia,␣
        ↪Pakistan, Nigeria and USA")
       df_Pak["new_cases"].plot(color="green", ax=ax)
       df_Ni["new_cases"].plot(color="purple", ax=ax)
       #logical indexing
       superdataT[superdataT["new_death"]=="new"]["Total"].plot(color="blue", ax=ax,␣
        ↪logy=True, figsize=(10,10))
```

```
[270]: <Axes: title={'center': 'New Cases for Indonesia, Pakistan, Nigeria and USA'},
       xlabel='Date'>
```



```
[271]: ax = df_Ind["new_deaths"].plot(color="red", title = "New Deaths for Indonesia,␣
        ↪Pakistan, Nigeria and USA")
       df_Pak["new_deaths"].plot(color="green", ax=ax)
       df_Ni["new_deaths"].plot(color="purple", ax=ax)
       #logical indexing
       superdataT[superdataT["new_death"]=="death"]["Total"].plot(color="blue", ax=ax,␣
        ↪logy=True, figsize=(10,10))
```

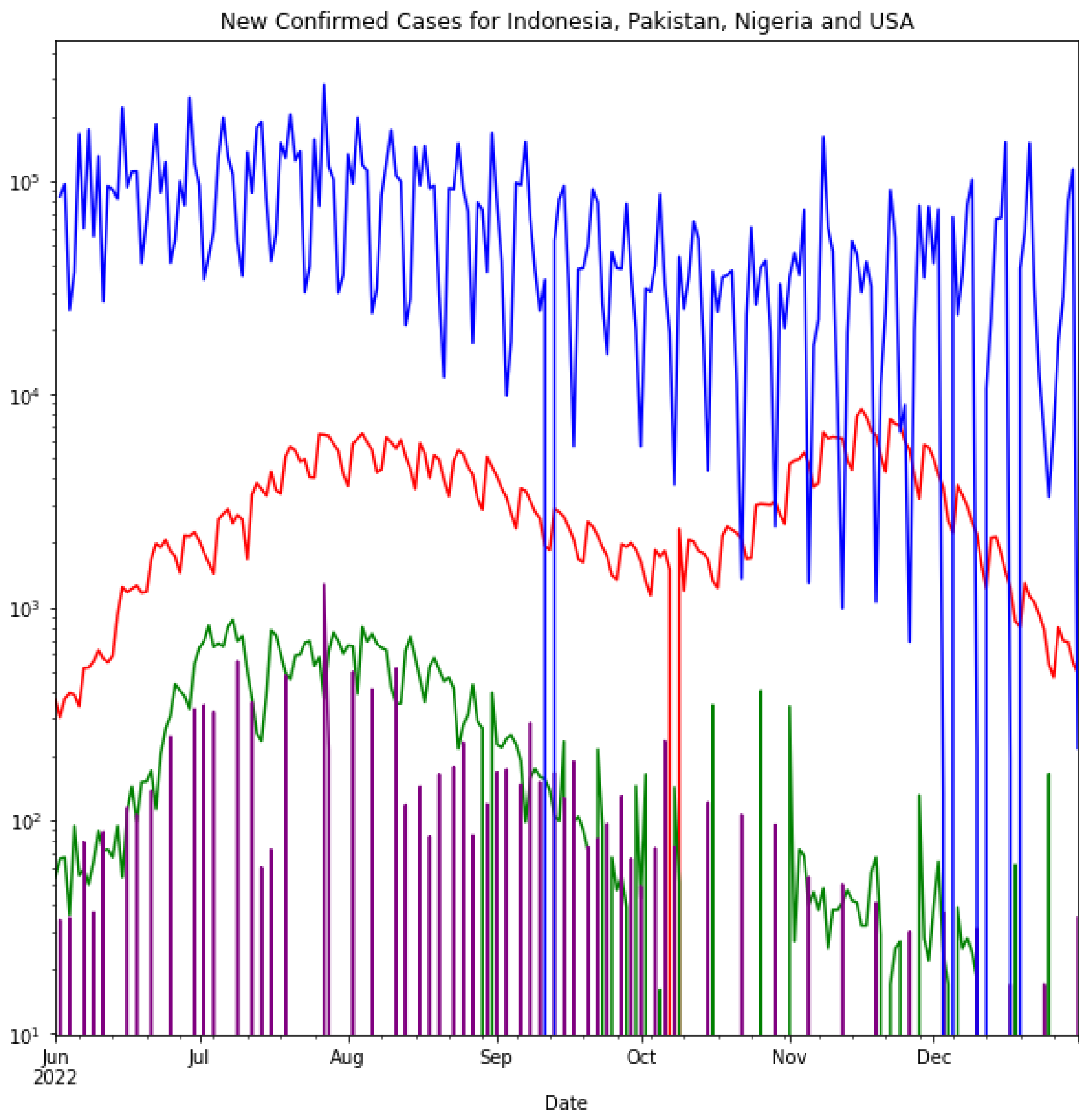
```
[271]: <Axes: title={'center': 'New Deaths for Indonesia, Pakistan, Nigeria and USA'},
       xlabel='Date'>
```



```
[272]: #difference between rows -> Difference between day/daily change
       superdataT["Total_dif"]=superdataT["Total"].diff()
```

```
[273]: ax = df_Ind["new_cases"].plot(color="red", title = "New Confirmed Cases for␣
        ↪Indonesia, Pakistan, Nigeria and USA")
       df_Pak["new_cases"].plot(color="green", ax=ax)
       df_Ni["new_cases"].plot(color="purple", ax=ax)
       #logical indexing
       superdataT[superdataT["new_death"]=="new"]["Total_dif"].plot(color="blue",␣
        ↪ax=ax, logy=True, figsize=(10,10))
```

```
[273]: <Axes: title={'center': 'New Confirmed Cases for Indonesia, Pakistan, Nigeria
       and USA'}, xlabel='Date'>
```



4.0.3 Peak weeks

[274]:
```
print('US peak death is: ')
superdataT[superdataT["new_death"]=="death"].groupby("Week").sum()["Total"].
  ↪max()
```

```
US peak death is:

<ipython-input-274-87f7836dd0ba>:2: FutureWarning: The default value of
numeric_only in DataFrameGroupBy.sum is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only
columns which should be valid for the function.
  superdataT[superdataT["new_death"]=="death"].groupby("Week").sum()["Total"].ma
x()
```

[274]:
```
7449099
```

[275]:
```
print('US peak cases is: ')
superdataT[superdataT["new_death"]=="new"].groupby("Week").sum()["Total"].max()
```

```
US peak cases is:

<ipython-input-275-edd7c2698e66>:2: FutureWarning: The default value of
numeric_only in DataFrameGroupBy.sum is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only
columns which should be valid for the function.
superdataT[superdataT["new_death"]=="new"].groupby("Week").sum()["Total"].max()
```

[275]:
```
667137461
```

[276]:
```
print('Indonesia peak death is: ')
df_Ind[["new_deaths"]].max()
```

```
Indonesia peak death is:
```

[276]:
```
new_deaths    59.0
dtype: float64
```

[277]:
```
print('Indonesia peak cases is: ')
df_Ind[["new_cases"]].max()
```

```
Indonesia peak cases is:
```

[277]:
```
new_cases    8486.0
dtype: float64
```

```
[278]: print('Pakistan peak death is: ')
       df_Pak["new_deaths"].max()

       Pakistan peak death is:

[278]: 11.0

[279]: print('Pakistan peak cases is: ')
       df_Pak["new_cases"].max()

       Pakistan peak cases is:

[279]: 872.0

[280]: print('Nigeria peak death is: ')
       df_Ni["new_deaths"].max()

       Nigeria peak death is:

[280]: 6.0

[281]: print('Nigeria peak cases is: ')
       df_Ni["new_cases"].max()

       Nigeria peak cases is:

[281]: 1279.0
```

4.0.4 Background research for weekly trends

Prior to the project as a team we assumed that peaks of data would be higher during the holidays, including last week of December, and month of January.

Some links relating to the data ar included in the following: Population data link: "https://www.indexmundi.com/g/r.aspx"

This link was used in order to select countries with a similar population density.

US: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9169704/#:~:text=The%20largest%20peak%20in%20hospit

This link describes peak hospitalizations which was used to assume the peak times of cases and deaths.

Indonesia:https://www.researchgate.net/publication/46395110_Multi_input_intervention_model_for_evaluatin

This link describes the Indonesian travel time which corelates to the peak Indonesian cases and deaths.

Pakistan: https://covid19.healthdata.org/pakistan?view=cumulative-deaths&tab=trend

This link provides a graphical analysis of the peak covid times for Pakistan

Nigeria: https://www.premiumtimesng.com/news/headlines/478855-covid-19-nigeria-records-790-new-cases-wednesday-highest-daily-infections-in-six-months.html?tztc=1

This link provides an article that shows the increase of covid cases relating to a surge in the same time the data shows a peak for Nigeria.