Pandemic_GroupProject_DATA201A

May 30, 2022

[1]:
```python
#Importing library
# Dalit, Mahshid, and Patrali

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm
import calendar

%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('seaborn-whitegrid')
import numpy as np
import pandas as pd
import seaborn as sns

#reading as a df
history = pd.read_csv('test.csv')
```

[2]:
```python
#viewing columns in df
history.columns
```

[2]:
```
Index(['date', 'death', 'deathIncrease', 'inIcuCumulative', 'inIcuCurrently',
       'hospitalizedIncrease', 'hospitalizedCurrently',
       'hospitalizedCumulative', 'negative', 'negativeIncrease',
       'onVentilatorCumulative', 'onVentilatorCurrently', 'positive',
       'positiveIncrease', 'states', 'totalTestResults',
       'totalTestResultsIncrease', 'month'],
      dtype='object')
```

[3]:
```python
#making the date tad a datetime dtype
history['date']= pd.to_datetime(history['date'])
#making total tests results a numeric dtype
history['totalTestResultsIncrease'] = pd.
 ↪to_numeric(history['totalTestResultsIncrease'])
```

[4]:

```
#printing information on each variable
history.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 420 entries, 0 to 419
Data columns (total 18 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   date                      420 non-null    datetime64[ns]
 1   death                     420 non-null    int64
 2   deathIncrease             420 non-null    int64
 3   inIcuCumulative           420 non-null    int64
 4   inIcuCurrently            420 non-null    int64
 5   hospitalizedIncrease      420 non-null    int64
 6   hospitalizedCurrently     420 non-null    int64
 7   hospitalizedCumulative    420 non-null    int64
 8   negative                  420 non-null    int64
 9   negativeIncrease          420 non-null    int64
 10  onVentilatorCumulative    420 non-null    int64
 11  onVentilatorCurrently     420 non-null    int64
 12  positive                  420 non-null    int64
 13  positiveIncrease          420 non-null    int64
 14  states                    420 non-null    int64
 15  totalTestResults          420 non-null    int64
 16  totalTestResultsIncrease  420 non-null    int64
 17  month                     420 non-null    object
dtypes: datetime64[ns](1), int64(16), object(1)
memory usage: 59.2+ KB
```

[5]:

```
#two neginc values were miss entered as negative
history[history['negativeIncrease'] < 0 ] #95 and 319
```

[5]:

```
          date   death  deathIncrease  inIcuCumulative  inIcuCurrently  \
95  2020-12-02  264990           2811            31038           19687
319 2020-04-22   44827           2150             2370           15705

     hospitalizedIncrease  hospitalizedCurrently  hospitalizedCumulative  \
95                   5238                 100327                  467773
319                  1591                  59204                   34423

     negative  negativeIncrease  onVentilatorCumulative  \
95   50105551           -658774                    3252
319   1667505           -213981                     227

     onVentilatorCurrently  positive  positiveIncrease  states  \
95                    6855  13925720            203429      56
319                   5474    838190             29273      56
```

```
     totalTestResults  totalTestResultsIncrease   month
95          201554613                   1587969  Dec-20
319           4797247                    335191  Apr-20
```

[6]:

```
#fixing the values that has data entered wrong
history['negativeIncrease'] = history['negativeIncrease'].abs()
history[history['negativeIncrease'] < 0 ] #fixed
```

[6]:

```
Empty DataFrame
Columns: [date, death, deathIncrease, inIcuCumulative, inIcuCurrently,
hospitalizedIncrease, hospitalizedCurrently, hospitalizedCumulative, negative,
negativeIncrease, onVentilatorCumulative, onVentilatorCurrently, positive,
positiveIncrease, states, totalTestResults, totalTestResultsIncrease, month]
Index: []
```

[7]:

```
## USED IN PPT
#this is used for indexing
dates = [5, 3, 9, 0, 11, 8, 7, 1, 14, 13, 12, 2, 6, 4, 10]
#mean of death increase by month
month_death = history.groupby('month', as_index = False)['deathIncrease'].mean()
#then sorting by the months with the lowest to highest increase in death
md_ss = month_death.sort_values('deathIncrease')
#making months the index to help when plotting if needed if not md_ss will be␣
 ↪used
md_s = md_ss.reindex(dates)
md_s = md_s.set_index('month')
#visualizing the grouped and sorted tabel
md_ss
```

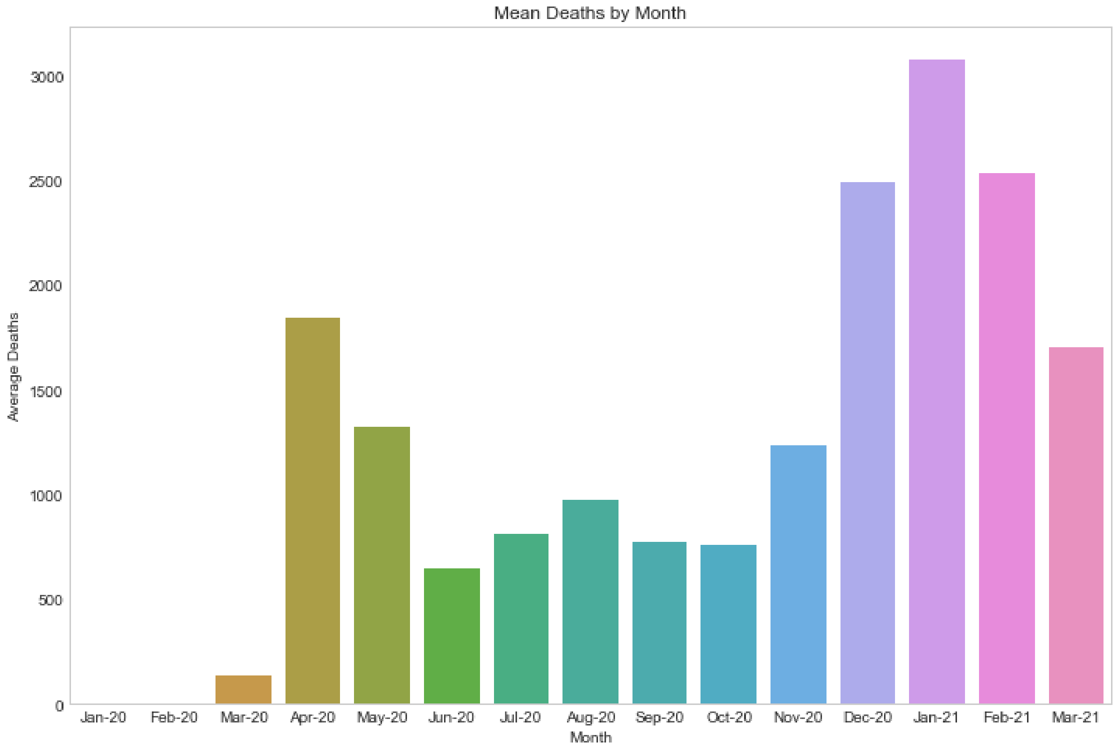
[7]:

```
     month  deathIncrease
5   Jan-20       0.000000
3   Feb-20       0.172414
9   Mar-20     139.548387
8   Jun-20     649.166667
13  Oct-20     759.516129
14  Sep-20     777.633333
7   Jul-20     814.483871
1   Aug-20     975.612903
12  Nov-20    1235.500000
11  May-20    1327.000000
10  Mar-21    1700.571429
0   Apr-20    1843.833333
2   Dec-20    2487.483871
4   Feb-21    2537.785714
6   Jan-21    3077.000000
```

[8]:

```
## USED IN PPT
# plotting average death increase by month
#using the table that has months as an index so they show up as labels in X axis
plt.figure(figsize = (12,8)) #making th eplot nice and big and readable
sns.barplot(md_s.index, "deathIncrease", data=md_s, ci=95) #plotting data
plt.title('Mean Deaths by Month') #title
plt.xlabel('Month') #x-label
plt.ylabel('Average Deaths') #y-label
axes = plt.gca()
axes.yaxis.grid() #making a grid to better see where the bars are at on the y␣
 ↪axis
plt.show() #showing plot
```

```
/Users/dalithendel/opt/anaconda3/lib/python3.8/site-
packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variables
as keyword args: x, y. From version 0.12, the only valid positional argument
will be `data`, and passing other arguments without an explicit keyword will
result in an error or misinterpretation.
  warnings.warn(
```



[9]:

```
## USE IN PPT
#average positive and negative cases by Month
dates = [5, 3, 9, 0, 11, 8, 7, 1, 14, 13, 12, 2, 6, 4, 10]
```

```
#mean of neg and pos tests grouped by month
month_test = history.groupby('month', as_index = False)['negativeIncrease',␣
 ↪'positiveIncrease'].mean()
#reindexing with dates list
mt = month_test.reindex(dates)
#setting the months as the index
mt = mt.set_index('month')
#sorting the values
mt
```

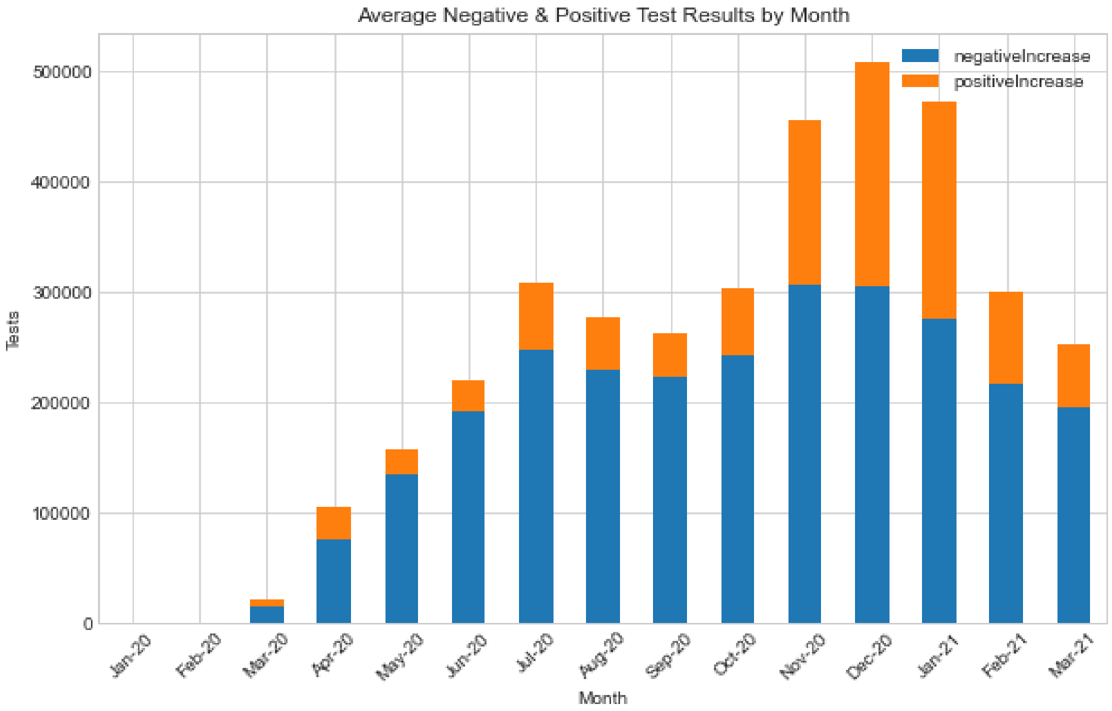
```
<ipython-input-9-4547a87feef2>:5: FutureWarning: Indexing with multiple keys
(implicitly converted to a tuple of keys) will be deprecated, use a list
instead.
  month_test = history.groupby('month', as_index = False)['negativeIncrease',
'positiveIncrease'].mean()
```

[9]:

```
        negativeIncrease  positiveIncrease
month
Jan-20          0.000000          0.105263
Feb-20          0.000000          0.551724
Mar-20      14858.129032       6353.129032
Apr-20      75698.966667      29209.300000
May-20     134563.580645      23167.903226
Jun-20     192426.733333      27719.900000
Jul-20     247878.193548      61296.129032
Aug-20     230209.870968      47006.870968
Sep-20     222806.166667      39755.433333
Oct-20     242665.387097      61032.741935
Nov-20     307506.733333     149199.700000
Dec-20     305196.612903     203976.322581
Jan-21     275074.516129     197179.741935
Feb-21     215890.750000      84794.392857
Mar-21     195200.142857      57900.000000
```

[10]:

```
## USE IN PPT
#plotting the figure
plt.rcParams["figure.figsize"] = [10, 6] #making th eplot nice and big and␣
 ↪readable
mt.plot(kind='bar', stacked=True) #plotting a stacked bar graph
plt.title('Average Negative & Positive Test Results by Month') #title
plt.xticks(rotation=45) #rotating the x labels
plt.ylabel('Tests') #ylabel
plt.xlabel('Month') #xlabel
plt.show()
```



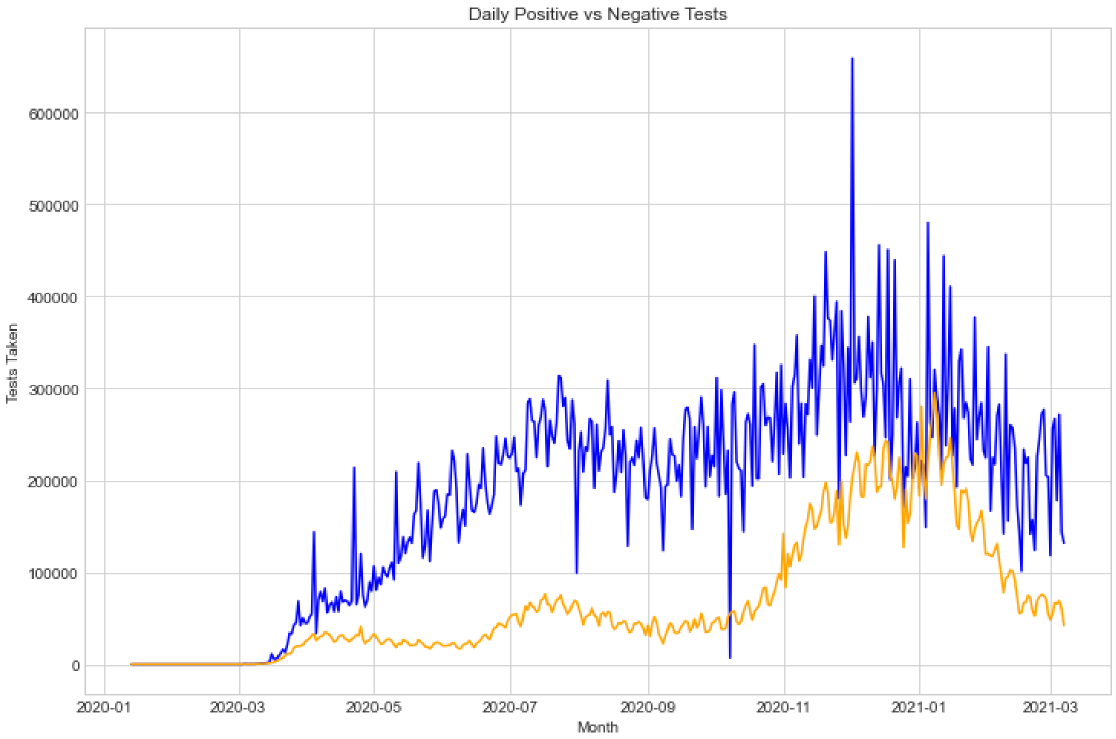
[11]:
```
## USE IN PPT
#calculating some stats
dates = [5, 3, 9, 0, 11, 8, 7, 1, 14, 13, 12, 2, 6, 4, 10] #used for indexing
#neg pos sum by month
month_test = history.groupby('month', as_index = False)['negativeIncrease',␣
 ↪'positiveIncrease'].sum()
me = month_test.reindex(dates) #reindexing with dates
me = me.set_index('month') #setting months as index
mee = me.sort_values('positiveIncrease') #this line acctualy I never used
me.negativeIncrease.sum() #76328335
me.positiveIncrease.sum() #28756489

diff = me.negativeIncrease.sum() - me.positiveIncrease.sum() #47571846

diff
```

```
<ipython-input-11-79cfe09a3459>:5: FutureWarning: Indexing with multiple keys
(implicitly converted to a tuple of keys) will be deprecated, use a list
instead.
  month_test = history.groupby('month', as_index = False)['negativeIncrease',
'positiveIncrease'].sum()
```

[11]:
```
47571846
```

[12]:

```
## USE IN PPT
#plotting the figure
plt.figure(figsize = (12,8)) # making plot bigger
plt.plot(history.date, history.negativeIncrease, color = 'blue') #plotting a␣
 ↪line for neg tests taken
plt.plot(history.date, history.positiveIncrease, color = 'orange') #plotting a␣
 ↪line for pos tests taken
plt.title('Daily Positive vs Negative Tests') #title
plt.ylabel('Tests Taken') #ylabel
plt.xlabel('Month') #xlabel
plt.show()
```



[ ]:

```

```

[13]:

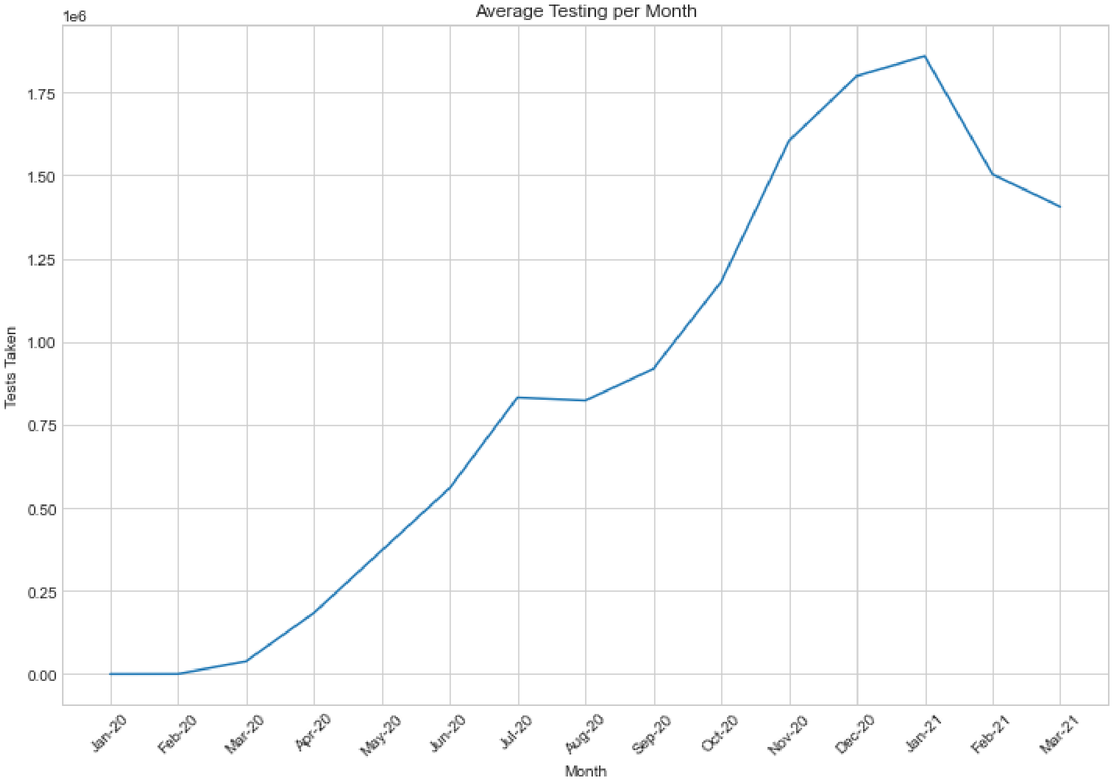
```
#mean total test results grouped by month
monthatest = history.groupby('month', as_index =␣
 ↪False)['totalTestResultsIncrease'].mean()
#used for indexing
dates = [5, 3, 9, 0, 11, 8, 7, 1, 14, 13, 12, 2, 6, 4, 10]
mat = monthatest.reindex(dates) #reindexing with dates
mat = mat.set_index('month') #making month the index
mat
```

[13]:

```
        totalTestResultsIncrease
month                           
Jan-20              4.210526e-01
Feb-20              2.257586e+02
Mar-20              3.796752e+04
Apr-20              1.834222e+05
May-20              3.714054e+05
Jun-20              5.587559e+05
Jul-20              8.318542e+05
Aug-20              8.225482e+05
Sep-20              9.176842e+05
Oct-20              1.179577e+06
Nov-20              1.604208e+06
Dec-20              1.799268e+06
Jan-21              1.859263e+06
Feb-21              1.503458e+06
Mar-21              1.405887e+06
```

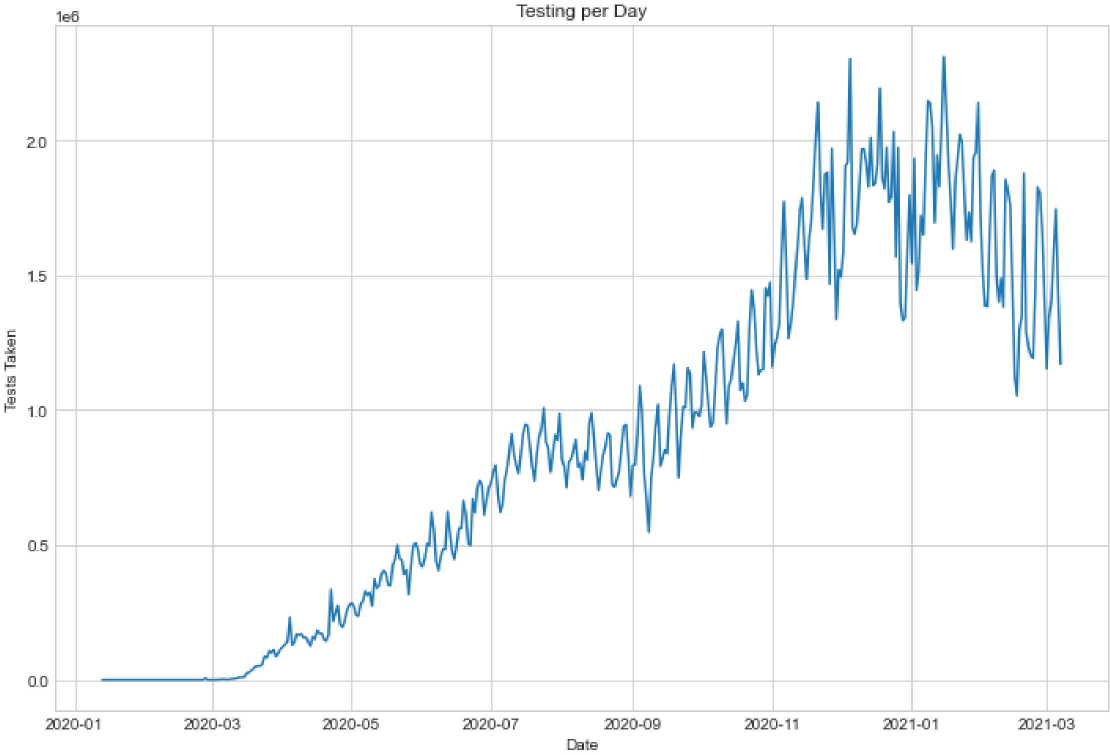
[14]:

```
## USED IN PPT
# plotting avg tests taken per month
plt.figure(figsize=(12,8)) #making figure bigger
plt.xticks(rotation = 45) #rotating x labels
plt.plot(mat.index, 'totalTestResultsIncrease', data=mat) #plotting
plt.title('Average Testing per Month') #title
plt.ylabel('Tests Taken') #ylabel
plt.xlabel('Month') #xlabel
plt.show()
```



[15]:

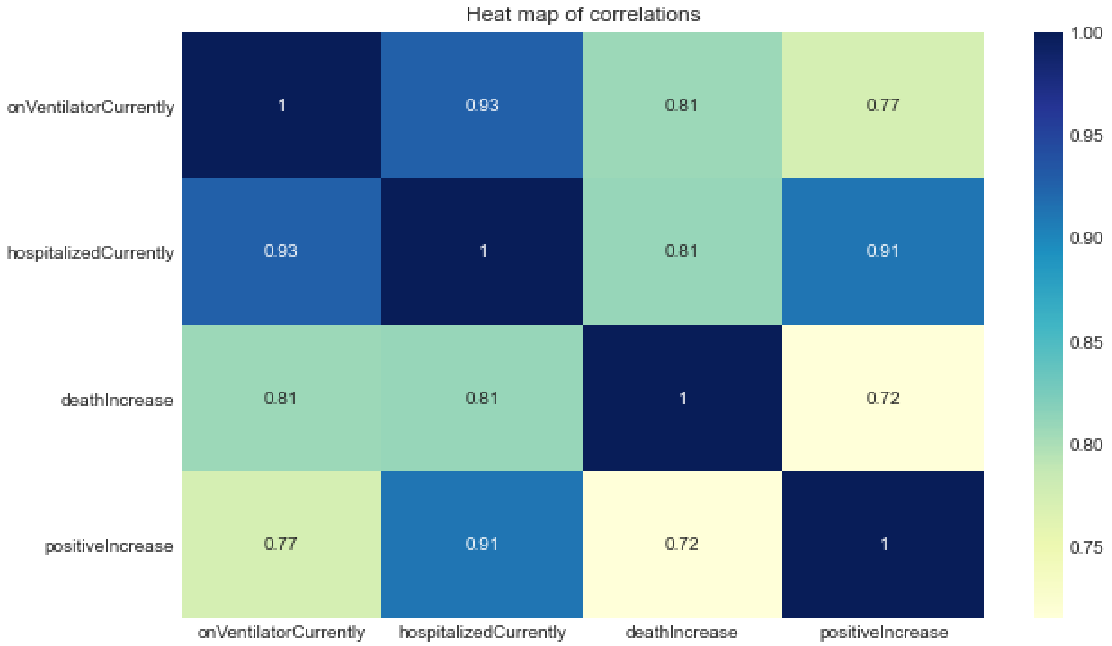
```
## USED IN PPT
# plotting tests taken per day
plt.figure(figsize=(12,8)) #mak eplot bigger
plt.plot('date', 'totalTestResultsIncrease', data=history) #plotting
plt.title('Testing per Day') #title
plt.ylabel('Tests Taken') #ylabel
plt.xlabel('Date') #xlabel
plt.show()
```



```
[16]: #total amount of tests in inests taken by day
      history.totalTestResultsIncrease.sum()
```

```
[16]: 363825123
```

```
[17]: #correlatin on ven vurrently, hospitalized currently, death increase, and pos␣
       ↪tests
      #heatmap
      history2 = history[['onVentilatorCurrently', 'hospitalizedCurrently',␣
       ↪'deathIncrease', 'positiveIncrease']]      #Dropping Year column as that␣
       ↪data is not significant
      plt.figure(figsize = (10,6))
      corrplot = sns.heatmap(history2.corr(), cmap="YlGnBu", annot=True)        ␣
       ↪#Making a heatmap for correlation between variables
      plt.title('Heat map of correlations')
      plt.show()
      history2.corr() #corr tabel
```



```
[17]:                        onVentilatorCurrently  hospitalizedCurrently  \
      onVentilatorCurrently               1.000000               0.926970
      hospitalizedCurrently               0.926970               1.000000
      deathIncrease                       0.808064               0.810131
      positiveIncrease                    0.773144               0.912692

                             deathIncrease  positiveIncrease
      onVentilatorCurrently       0.808064          0.773144
      hospitalizedCurrently       0.810131          0.912692
      deathIncrease               1.000000          0.715154
      positiveIncrease            0.715154          1.000000
```

```
[18]: ###### merged notebooks with group members
```

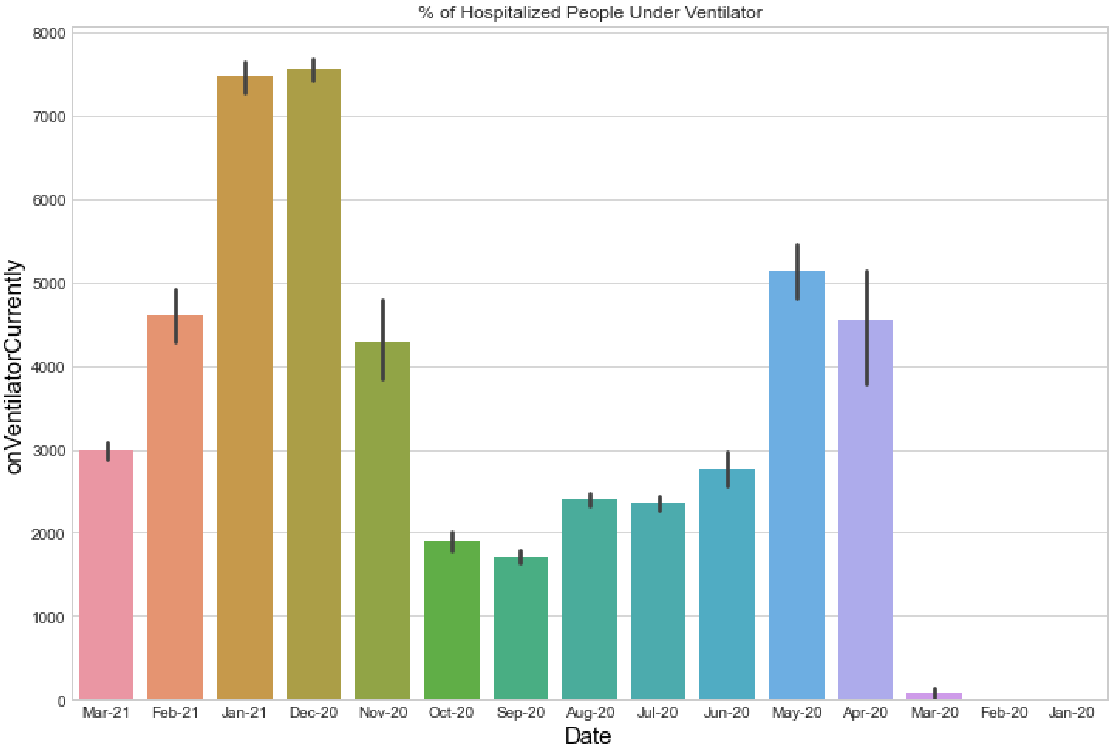
```
[19]: history.columns
```

```
[19]: Index(['date', 'death', 'deathIncrease', 'inIcuCumulative', 'inIcuCurrently',
             'hospitalizedIncrease', 'hospitalizedCurrently',
             'hospitalizedCumulative', 'negative', 'negativeIncrease',
             'onVentilatorCumulative', 'onVentilatorCurrently', 'positive',
             'positiveIncrease', 'states', 'totalTestResults',
             'totalTestResultsIncrease', 'month'],
            dtype='object')
```

```
[20]: #NOT USED IN PPT
      #plotting what percent of hospitalized people were on ventilators each month
```

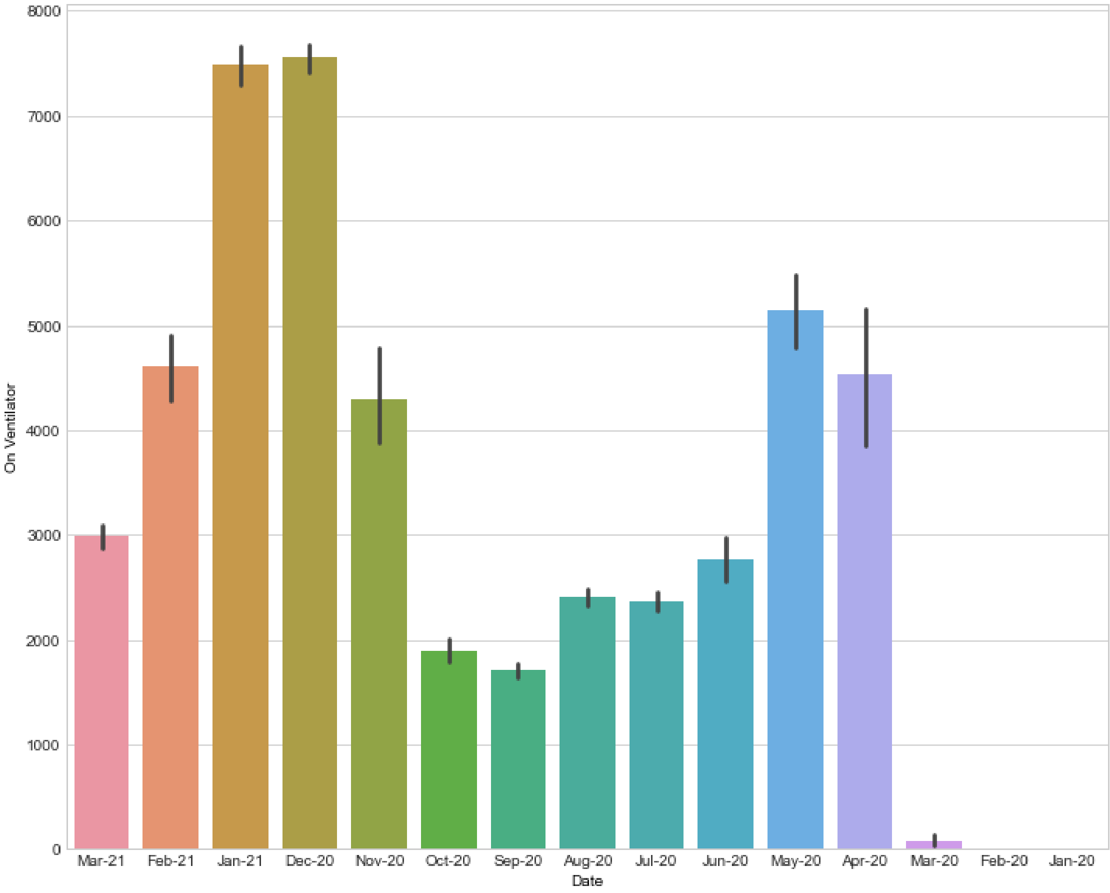
```
plt.figure(figsize = (12,8)) #increase plot size
plt.ylabel('% Hospitalized people on ventilator',size=15, color='black') #ylabel
sns.barplot(history['month'],history.onVentilatorCurrently , data = history)␣
 ↪#plttin as a bargraph with error bars
plt.xlabel('Date',size=15, color='black') #xlabel
plt.title('% of Hospitalized People Under Ventilator') #title
plt.show()
```

```
/Users/dalithendel/opt/anaconda3/lib/python3.8/site-
packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variables
as keyword args: x, y. From version 0.12, the only valid positional argument
will be `data`, and passing other arguments without an explicit keyword will
result in an error or misinterpretation.
  warnings.warn(
```



[21]:
```
#NOT USED IN PPT
#Number of people on ventilator over time
plt.figure(figsize = (12,10))    #make fig larger
sns.barplot(history['month'], history['onVentilatorCurrently'], data=history)␣
 ↪#barplot with arror bars
plt.xlabel('Date',size=10, color='black') #xlabel
plt.ylabel('On Ventilator',size=10, color='black') #ylabel
plt.show()
```

```
/Users/dalithendel/opt/anaconda3/lib/python3.8/site-
packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variables
as keyword args: x, y. From version 0.12, the only valid positional argument
will be `data`, and passing other arguments without an explicit keyword will
result in an error or misinterpretation.
  warnings.warn(
```



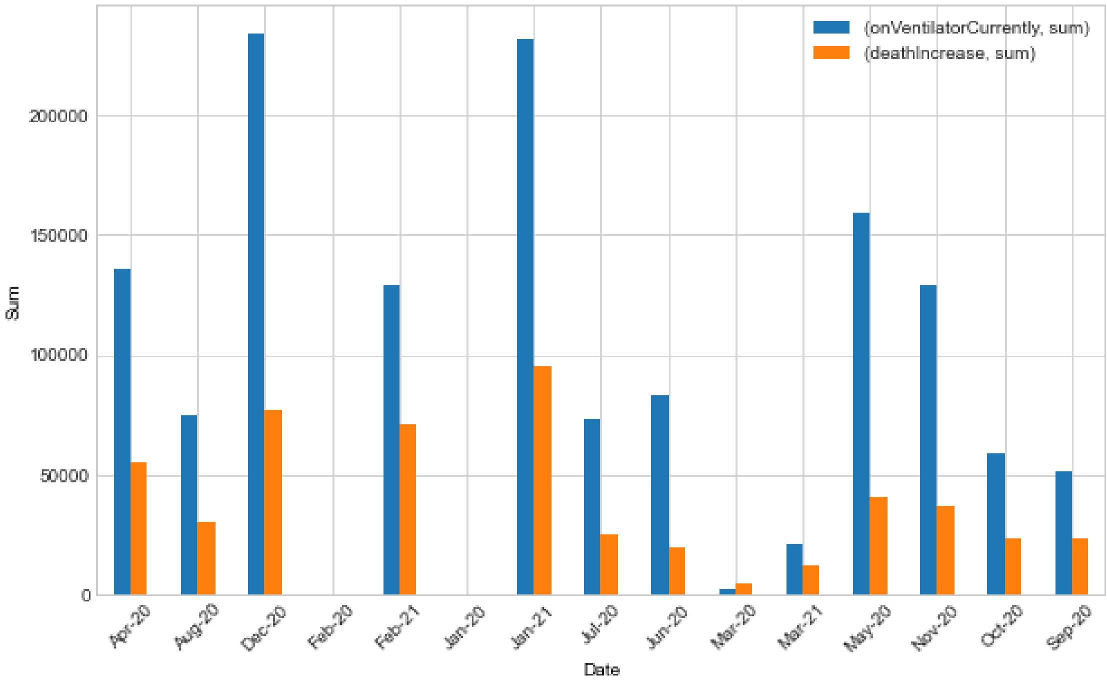
[22]:

```
# Hospitalized and Death per Months   #Used in ppt
plt.figure(figsize = (20,10)) #making fig bigger
National_history_rev = history.groupby(history['month'], as_index=␣
 ↪False)['onVentilatorCurrently','deathIncrease'].agg(['sum']) #grouppint on␣
 ↪vent and deaths by month
#dates = [5, 3, 9, 0, 11, 8, 7, 1, 14, 13, 12, 2, 6, 4, 10]
#National_history_rev = National_history_rev.reindex(dates)
#National_history_rev.sort_values(['month'])
#National_history_rev = National_history_rev.set_index(['month'])
#plotting a bar plot with the variables side by side for comparison
ax = National_history_rev.plot.bar(rot=0)
plt.xticks(rotation=45) #xlabels rotation
plt.xlabel('Date',size=10, color='black') #xlabels
plt.ylabel('Sum',size=10, color='black') #ylabels
ax.get_legend().remove()
plt.legend() #plotting a legend
```

```
<ipython-input-22-8cf6bce6f340>:3: FutureWarning: Indexing with multiple keys
(implicitly converted to a tuple of keys) will be deprecated, use a list
instead.
  National_history_rev = history.groupby(history['month'], as_index=
False)['onVentilatorCurrently','deathIncrease'].agg(['sum']) #grouppint on vent
and deaths by month
```

```
[22]: <matplotlib.legend.Legend at 0x7feda1c21430>
```

```
<Figure size 1440x720 with 0 Axes>
```



```
[23]: #groupby to visualize table for those currently on ventilators and those␣
       ↪currently hospitalized
      a = history.
       ↪groupby(history['month'])['onVentilatorCurrently','hospitalizedCurrently'].
       ↪agg(['sum'])
      a
```

```
<ipython-input-23-27cb52c8b9c2>:2: FutureWarning: Indexing with multiple keys
(implicitly converted to a tuple of keys) will be deprecated, use a list
instead.
```

```
  a = history.groupby(history['month'])['onVentilatorCurrently','hospitalizedCur
rently'].agg(['sum'])
```

[23]:

```
        onVentilatorCurrently hospitalizedCurrently
                          sum                   sum
month
Apr-20                 136194               1510437
Aug-20                  74602               1392199
Dec-20                 234228               3467428
Feb-20                      0                     0
Feb-21                 128960               1930850
Jan-20                      0                     0
Jan-21                 231906               3743799
Jul-20                  73163               1613867
Jun-20                  82814                926523
Mar-20                   2496                 97142
Mar-21                  20909                306901
May-20                 159333               1375940
Nov-20                 128835               2168935
Oct-20                  58676               1173562
Sep-20                  51288                935831
```

[24]:

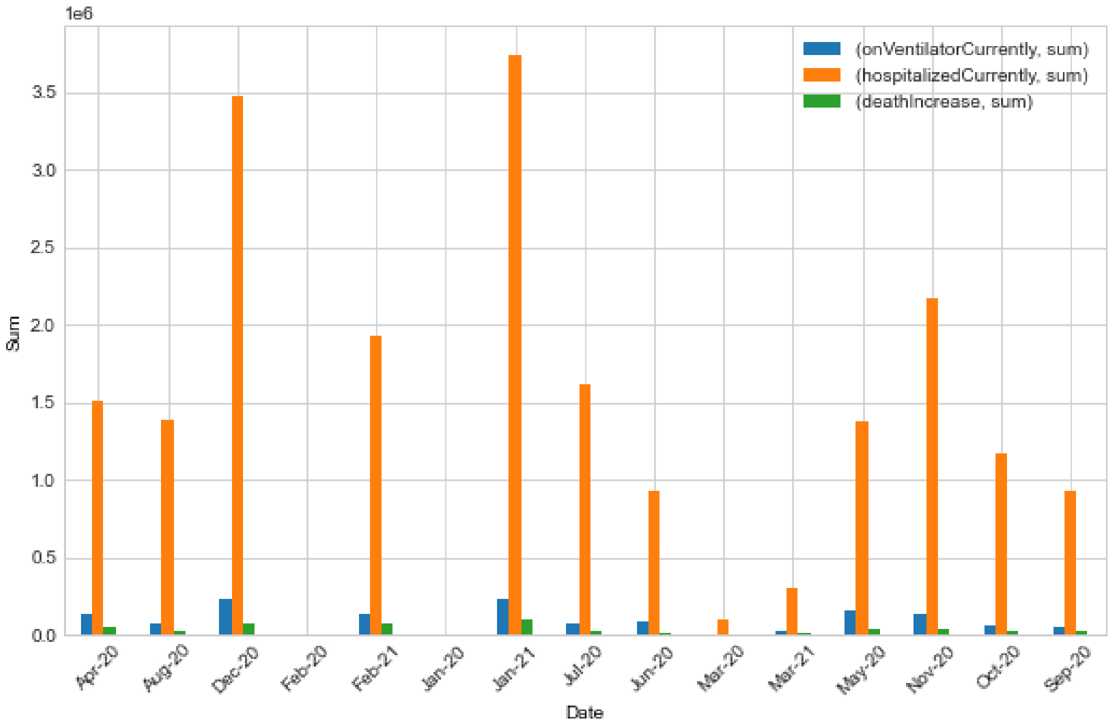
```
# Hospitalized, Ventilator, and Death per Months   #Used in ppt
plt.figure(figsize = (20,10)) #making the fig bigger
#groupby hospitalized, death increase, and on ventilator by month
National_history_rev = history.
 ↪groupby(history['month'])['onVentilatorCurrently','hospitalizedCurrently',␣
 ↪'deathIncrease'].agg(['sum'])
#National_history_rev.sort_values(['month'])
#National_history_rev = National_history_rev.set_index(['month'])
#National_history_rev

ax = National_history_rev.plot.bar(rot=0)
plt.xticks(rotation=45) #changing label rotation on
plt.xlabel('Date',size=10, color='black') #xlabel
plt.ylabel('Sum',size=10, color='black') #ylabel
ax.get_legend().remove()
plt.legend()
```

```
<ipython-input-24-f47a683c0bca>:4: FutureWarning: Indexing with multiple keys
(implicitly converted to a tuple of keys) will be deprecated, use a list
instead.
  National_history_rev = history.groupby(history['month'])['onVentilatorCurrentl
y','hospitalizedCurrently', 'deathIncrease'].agg(['sum'])
```

[24]:

```
<matplotlib.legend.Legend at 0x7feda124dd90>
```

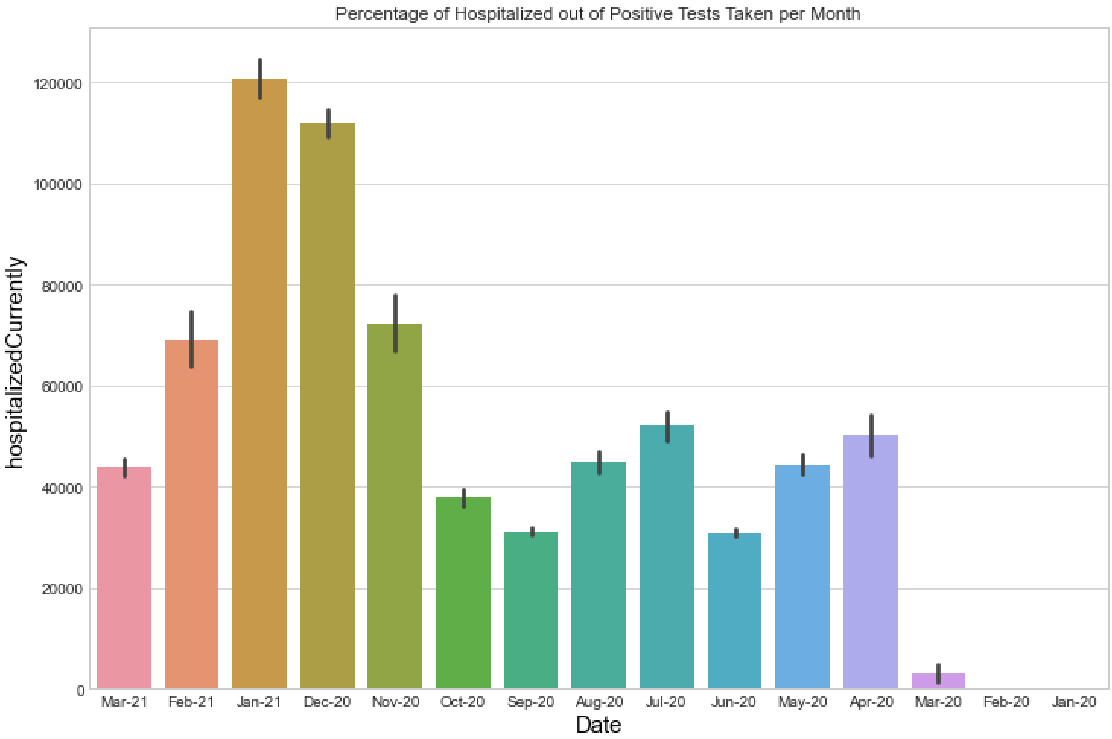
```
<Figure size 1440x720 with 0 Axes>
```



[25]:

```
#what percentage of positive testing people end up in hospital each month ␣
 ↪#used in ppt
plt.figure(figsize = (12,8)) #increase fig size
plt.ylabel('Percentage of Hospitalized out of Positive Tests',size=15,␣
 ↪color='black') #plot as bar plt
sns.barplot(history['month'], history.hospitalizedCurrently, data = history)␣
 ↪#xlabel
plt.xlabel('Date',size=15, color='black') #ylabel
plt.title('Percentage of Hospitalized out of Positive Tests Taken per Month')␣
 ↪#title
plt.show()
```

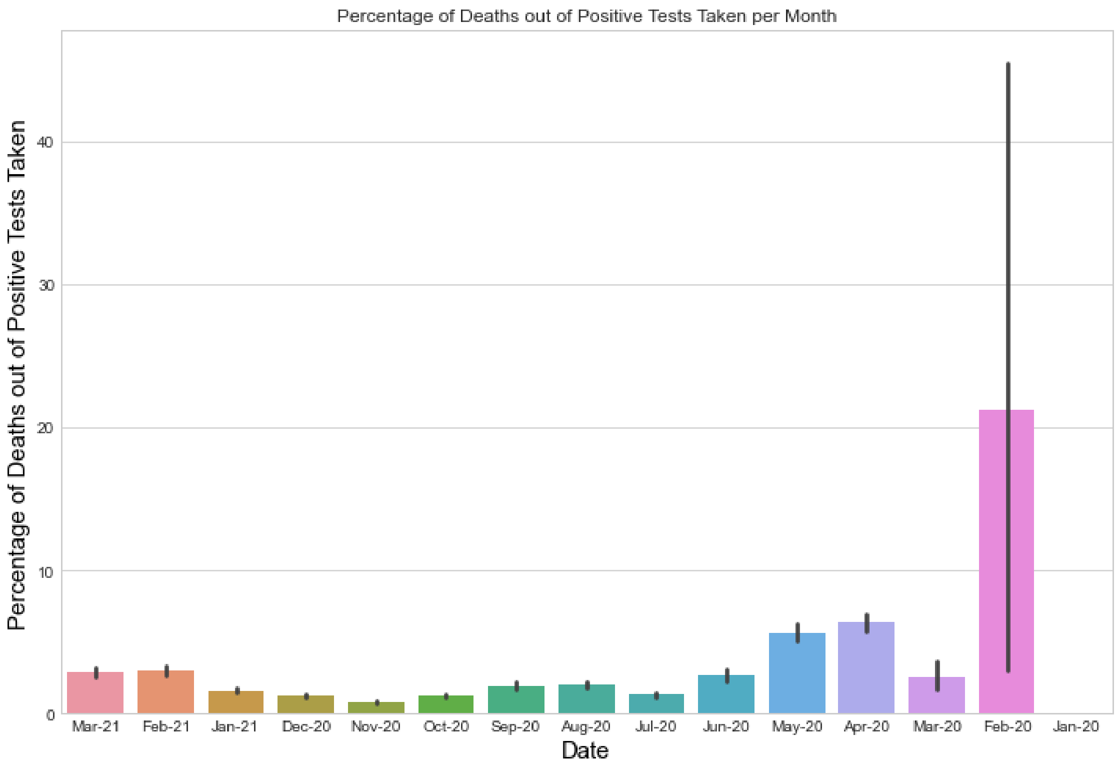
```
/Users/dalithendel/opt/anaconda3/lib/python3.8/site-
packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variables
as keyword args: x, y. From version 0.12, the only valid positional argument
will be `data`, and passing other arguments without an explicit keyword will
result in an error or misinterpretation.
  warnings.warn(
```



[26]:

```
#what percentage of positive testing people die each month    #used in ppt
Die = (history['deathIncrease']/history['positiveIncrease'])*100 #getting a␣
 ↪percentage of death out of positive cases
plt.figure(figsize = (12,8)) #making fig bigger
plt.ylabel('Percentage of Deaths out of Positive Tests Taken',size=15,␣
 ↪color='black') #ylabel
sns.barplot(history['month'], Die, data = history) #plotting bar plot
plt.xlabel('Date',size=15, color='black') #xlabel
plt.title('Percentage of Deaths out of Positive Tests Taken per Month') #title
plt.show()
```

```
/Users/dalithendel/opt/anaconda3/lib/python3.8/site-
packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variables
as keyword args: x, y. From version 0.12, the only valid positional argument
will be `data`, and passing other arguments without an explicit keyword will
result in an error or misinterpretation.
  warnings.warn(
```



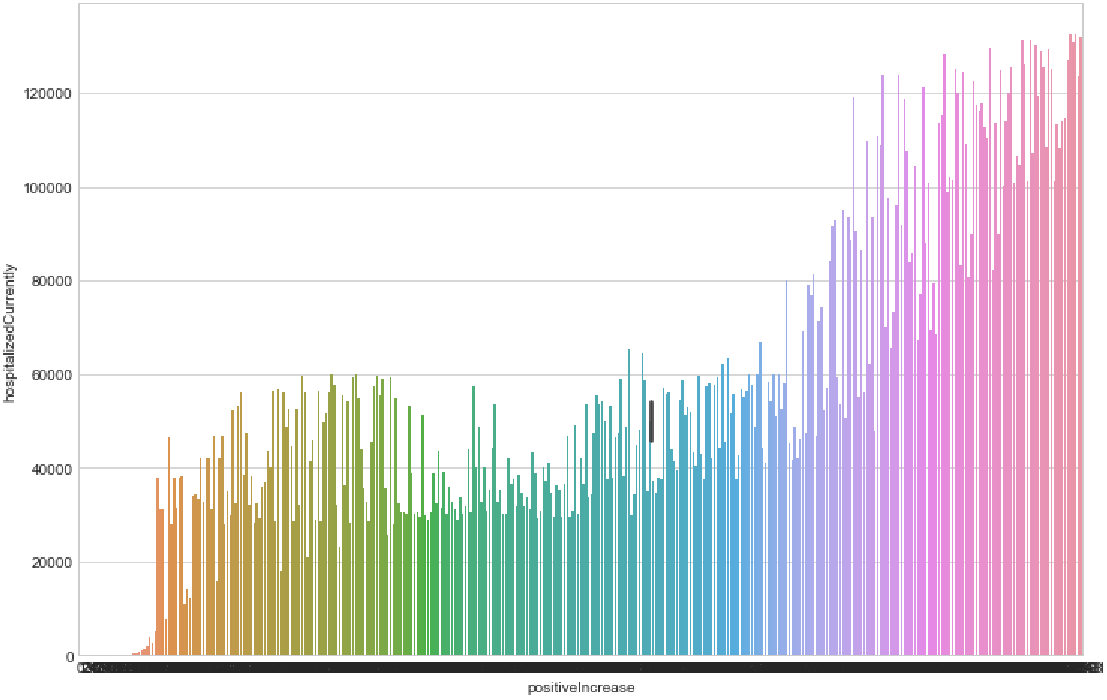
[27]:

```
history['hospitalizedIncrease'] = abs(history['hospitalizedIncrease'])
```

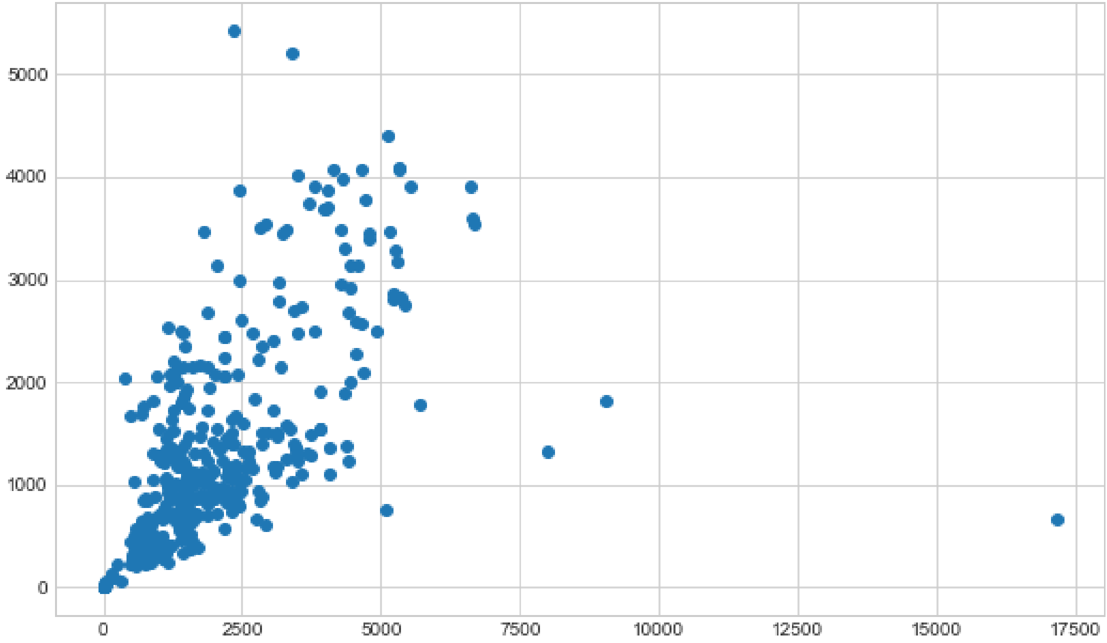
[28]:

```
# This graph shows the cumulative ICU patients per month
plt.figure(figsize = (12,8))
sns.barplot( "positiveIncrease","hospitalizedCurrently", data=history)
plt.show()
```

```
/Users/dalithendel/opt/anaconda3/lib/python3.8/site-
packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variables
as keyword args: x, y. From version 0.12, the only valid positional argument
will be `data`, and passing other arguments without an explicit keyword will
result in an error or misinterpretation.
  warnings.warn(
```



[29]:
```
plt.scatter("hospitalizedIncrease","deathIncrease", data=history)
plt.show()
```



[30]:

```
#looking at unique
history.nunique()
# The dates column only has unique values
```

[30]:

```
date                        420
death                       376
deathIncrease               344
inIcuCumulative             349
inIcuCurrently              346
hospitalizedIncrease        347
hospitalizedCurrently       357
hospitalizedCumulative      367
negative                    373
negativeIncrease            372
onVentilatorCumulative      322
onVentilatorCurrently       333
positive                    386
positiveIncrease            375
states                       16
totalTestResults            398
totalTestResultsIncrease    380
month                        15
dtype: int64
```

[31]:

```
history.death.describe()
# the average death in given timeline is 174729.957143, with a high standard␣
 ↪deviation of 145225.627340
```

[31]:

```
count       420.000000
mean     174729.957143
std      145225.627340
min           0.000000
25%       52407.500000
50%      154802.000000
75%      248777.250000
max      515151.000000
Name: death, dtype: float64
```

[32]:

```
history.positiveIncrease.describe()
#the mean positive increase in cases is 68467.830952 with a standard deviation␣
 ↪higher than the mean at 68682.221496,
#which means that the distribution of data is abnormal, which sort of␣
 ↪corresponds with how the covid 19 cases increased
```

[32]:

```
count      420.000000
mean     68467.830952
std      68682.221496
```

```
min            0.000000
25%        22462.500000
50%        44664.500000
75%        89684.250000
max       295121.000000
Name: positiveIncrease, dtype: float64
```

[33]:
```
history.negativeIncrease.describe()
#the mean negative increase in cases is 181734.130952 with a standard deviation␣
 ↪of 115222.401108 . In this dataset,
#the maximum increase in negativity is 658774.
```

[33]:
```
count       420.000000
mean     181734.130952
std      115222.401108
min           0.000000
25%       80563.000000
50%      208387.500000
75%      260898.750000
max      658774.000000
Name: negativeIncrease, dtype: float64
```

[34]:
```
history.negativeIncrease.describe()
```

[34]:
```
count       420.000000
mean     181734.130952
std      115222.401108
min           0.000000
25%       80563.000000
50%      208387.500000
75%      260898.750000
max      658774.000000
Name: negativeIncrease, dtype: float64
```

[35]:
```
#the mean negative increase in cases is 181734.130952 with a standard deviation␣
 ↪of 115222.401108 . In this dataset,
#the maximum increase in negativity is 658774.
history.totalTestResultsIncrease.describe()
```

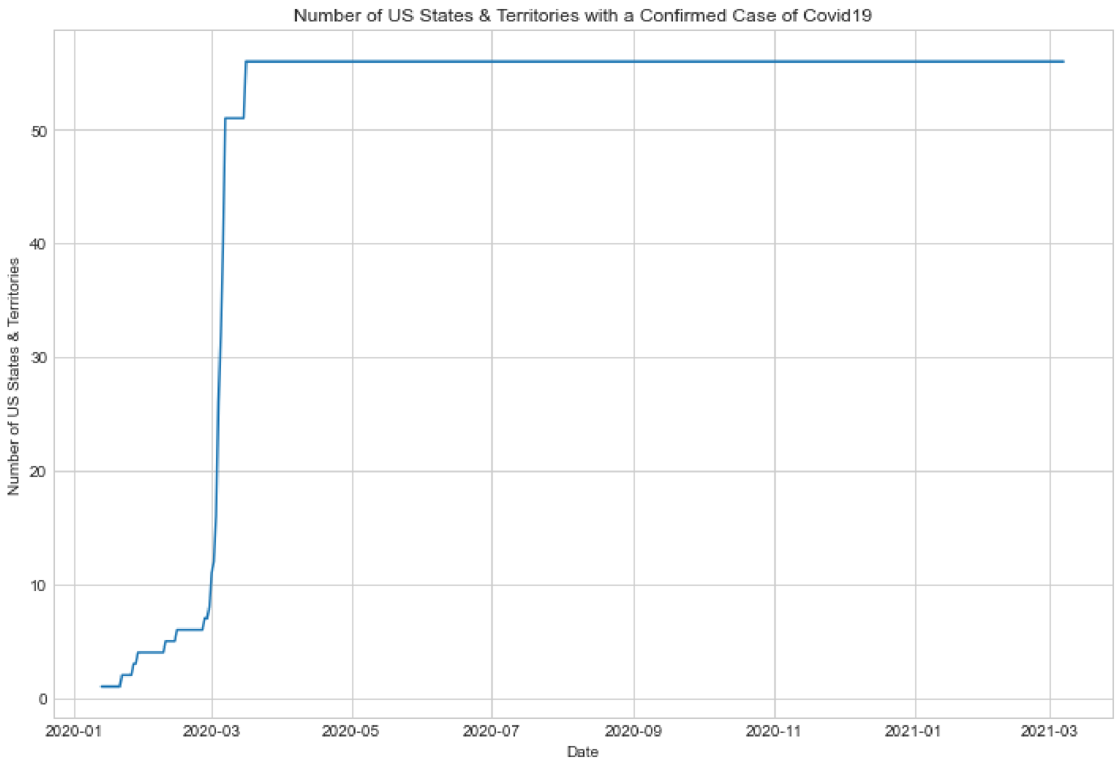
[35]:
```
count    4.200000e+02
mean     8.662503e+05
std      6.579460e+05
min      0.000000e+00
25%      2.388558e+05
50%      8.223685e+05
75%      1.401706e+06
max      2.309884e+06
```

```
Name: totalTestResultsIncrease, dtype: float64
```

```
[36]: history.hospitalizedIncrease.describe()
      #The mean increase in hospitalization is 1865.66904. The standard deviation is␣
       ↪1689.626250. The highest increase in
      #hospitalization was 17155.
```

```
[36]: count      420.000000
      mean      1865.669048
      std       1689.626250
      min          0.000000
      25%        774.250000
      50%       1492.500000
      75%       2488.000000
      max      17155.000000
      Name: hospitalizedIncrease, dtype: float64
```

```
[38]: plt.figure(figsize=(12,8))
      plt.plot('date', 'states', data=history)
      plt.title('Number of US States & Territories with a Confirmed Case of Covid19')
      plt.ylabel('Number of US States & Territories')
      plt.xlabel('Date')

      plt.show()
```



[ ]:

```
```