1 COVID-19 Data Analysis and Forecasting

Author: Joseph Denney

Email: joseph.d.denney@gmail.com (mailto:joseph.d.denney@gmail.com)

github: www.github.com/josephdenney/Covid_Analysis (http://www.github.com/josephdenney/Covid_Analysis)

1.1 Introduction

1.1.1 Problem and Purpose

This project will use forecasting to model Covid-19 deaths based on current hospitalization, ventilator, and death data. I will be using API html links to bring in up to date data regularly. This project will use supervised learning in the form of SARIMA and SARIMAX in order to create time series death forecasts.

The purpose of this analysis is to provide an accurate forecast of Covid-19 related deaths as 2021 progresses.

Our challenges are -

*** 1. Create multiple forecasts by creating forecasts for specific states**

*** 2. Build a forecast for the United States as a whole**

*** 3. Provide insights as to the urgency of making changes to how we are operating as a country**

1.2 Table of Contents

1.2.1 Standard Libraries

In [31]:

```
from tensorflow.random import set_seed
import numpy as np

set_seed(42)
np.random.seed(42)
```

In [32]:
```
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D
import matplotlib
%matplotlib inline
import functools
from jupyter_plotly_dash import JupyterDash
import datetime as dt
from datetime import date
from datetime import datetime, timedelta
import pandas_datareader as pdr
import holidays
```

In [33]:
```
from statsmodels.tsa.statespace.tools import diff
from statsmodels.tsa.stattools import acovf, acf, pacf, pacf_yw, pacf_ols
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.stattools import adfuller
from statsmodels.tools.eval_measures import mse,rmse
from pmdarima import auto_arima
from statsmodels.tsa.seasonal import seasonal_decompose as sd
from statsmodels.tsa.statespace.varmax import VARMAX, VARMAXResults
from statsmodels.tsa.ar_model import AR, ARResults
from statsmodels.tsa.arima_model import ARMA, ARIMA, ARMAResults
from statsmodels.tsa.statespace.sarimax import SARIMAX
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from keras.preprocessing.sequence import TimeseriesGenerator
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout
from keras.callbacks import EarlyStopping
from keras.callbacks import ModelCheckpoint, TensorBoard
```

In [34]:
```
import warnings
warnings.filterwarnings('ignore')

import itertools
import statsmodels.api as sm
from matplotlib.pylab import rcParams
plt.style.use('ggplot')
```

In [35]:
```
scaler = MinMaxScaler()
standard_scaler = StandardScaler()
```

1.2.2 Custom Libraries

In [156]:
```
%load_ext autoreload
%autoreload 2
%reload_ext autoreload
from Cust_Func import *
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

Contents

1.3 Explore Data

Contents

1.3.1 Create New DataFrame

In [37]:
```
open_links = False
```

In [38]:
```
import webbrowser

if open_links == True:
    webbrowser.open("https://covidtracking.com/")
    webbrowser.open("https://covidtracking.com/data/api")
```

The Covid Tracking Project was organized by the news agency The Atlantic early in 2020 in an effort to provide as much data on the pandemic as possible. Coordination of state by state Covid data required building working relationships with state officials to obtain relevant state information. Above are links to the project that can provide further information regarding Covid-19. Additionally, it is worth noting that the project is coming to its end at the beginning of March 2021 as a result of improvements to Federal collection of data.

In [39]:
```
# set to true to fetch new data
get_data = True
```

In [40]:
```
if get_data == True:
    df_states = pd.read_csv('https://api.covidtracking.com/v1/states/dai
    df_whole_US = pd.read_csv('https://api.covidtracking.com/v1/us/daily
    df_states.to_csv('StateData.csv')
    df_whole_US.to_csv('USA.csv')
else:
    df_states = pd.read_csv('StateData.csv', index_col='date', parse_date
    df_whole_US = pd.read_csv('USA.csv', index_col='date', parse_dates=Tr
```

In [41]:
```
df_states.head()
```

Out[41]:

date	state	positive	probableCases	negative	pending	totalTestResultsSource	totalTestRe
2021-02-12	AK	54282.0	NaN	NaN	NaN	totalTestsViral	1584
2021-02-12	AL	478667.0	103040.0	1842516.0	NaN	totalTestsPeopleViral	2218
2021-02-12	AR	311608.0	64580.0	2322916.0	NaN	totalTestsViral	2569
2021-02-12	AS	0.0	NaN	2140.0	NaN	totalTestsViral	2
2021-02-12	AZ	793532.0	53218.0	2864855.0	NaN	totalTestsViral	7140

5 rows × 54 columns

In [42]:
```
1 df_states.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 19485 entries, 2021-02-12 to 2020-01-13
Data columns (total 54 columns):
#   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
0   state                       19485 non-null  object
1   positive                    19296 non-null  float64
2   probableCases               8390 non-null   float64
3   negative                    15509 non-null  float64
4   pending                     2024 non-null   float64
5   totalTestResultsSource      19485 non-null  object
6   totalTestResults            19383 non-null  float64
7   hospitalizedCurrently       16119 non-null  float64
8   hospitalizedCumulative      12017 non-null  float64
9   inIcuCurrently              10568 non-null  float64
10  inIcuCumulative             3516 non-null   float64
11  onVentilatorCurrently       8386 non-null   float64
12  onVentilatorCumulative      1201 non-null   float64
13  recovered                   13992 non-null  float64
```

Contents

In [43]:
```
1 df_whole_US.head()
```

Out[43]:

date	states	positive	negative	pending	hospitalizedCurrently	hospitalizedCumulative
2021-02-12	56	27266230.0	122400369.0	9434.0	71504.0	839119.0
2021-02-11	56	27165660.0	121833298.0	11981.0	74225.0	836774.0
2021-02-10	56	27063243.0	121244702.0	12079.0	76979.0	834314.0
2021-02-09	56	26968049.0	120859564.0	10516.0	79179.0	831088.0
2021-02-08	56	26875063.0	120367478.0	12114.0	80055.0	827944.0

5 rows × 24 columns

1.3.2 Plot of Ventilators and Deaths for Each State

In [44]:
```
df_states.columns.unique
```

Out[44]:
```
<bound method Index.unique of Index(['state', 'positive', 'probableCases',
'negative', 'pending',
       'totalTestResultsSource', 'totalTestResults', 'hospitalizedCurrentl
y',
       'hospitalizedCumulative', 'inIcuCurrently', 'inIcuCumulative',
       'onVentilatorCurrently', 'onVentilatorCumulative', 'recovered',
       'dataQualityGrade', 'lastUpdateEt', 'dateModified', 'checkTimeEt',
       'death', 'hospitalized', 'dateChecked', 'totalTestsViral',
       'positiveTestsViral', 'negativeTestsViral', 'positiveCasesViral',
       'deathConfirmed', 'deathProbable', 'totalTestEncountersViral',
       'totalTestsPeopleViral', 'totalTestsAntibody', 'positiveTestsAntibod
y',
       'negativeTestsAntibody', 'totalTestsPeopleAntibody',
       'positiveTestsPeopleAntibody', 'negativeTestsPeopleAntibody',
       'totalTestsPeopleAntigen', 'positiveTestsPeopleAntigen',
       'totalTestsAntigen', 'positiveTestsAntigen', 'fips', 'positiveIncrea
se',
       'negativeIncrease', 'total', 'totalTestResultsIncrease', 'posNeg',
       'deathIncrease', 'hospitalizedIncrease', 'hash', 'commercialScore',
       'negativeRegularScore', 'negativeScore', 'positiveScore', 'score',
       'grade'],
      dtype='object')>
```

In [45]:
```
df_states.isnull().sum()
```

```
totalTestsPeopleAntigen        18601
positiveTestsPeopleAntigen     18921
totalTestsAntigen              16559
positiveTestsAntigen           17563
fips                               0
positiveIncrease                   0
negativeIncrease                   0
total                              0
totalTestResultsIncrease           0
posNeg                             0
deathIncrease                      0
hospitalizedIncrease               0
hash                               0
commercialScore                    0
negativeRegularScore               0
negativeScore                      0
positiveScore                      0
score                              0
grade                          19485
```

In [46]:
```
# columns to keep in dataframe
columns = ['state','death','inIcuCurrently','onVentilatorCurrently','pos
```

In [47]:
```
for col in columns:
    df_states[col] = df_states[col].fillna(0)
```

Contents

In [48]:
```
df_states = sort_and_clean_df(dataframe=df_states, target_columns=columns
```

In [49]:
```
df_states.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 19485 entries, 2020-01-13 to 2021-02-12
Data columns (total 7 columns):
#   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
0   state                  19485 non-null  object
1   death                  19485 non-null  float64
2   inIcuCurrently         19485 non-null  float64
3   onVentilatorCurrently  19485 non-null  float64
4   positive               19485 non-null  float64
5   hospitalizedCurrently  19485 non-null  float64
6   deathIncrease          19485 non-null  int64
dtypes: float64(5), int64(1), object(1)
memory usage: 1.2+ MB
```

Contents

In [50]:
```
df_states.iloc[-50:].sort_values(by='death',ascending=False)
# # only graph the top 7
# that keep state ventilator data
```

Out[50]:

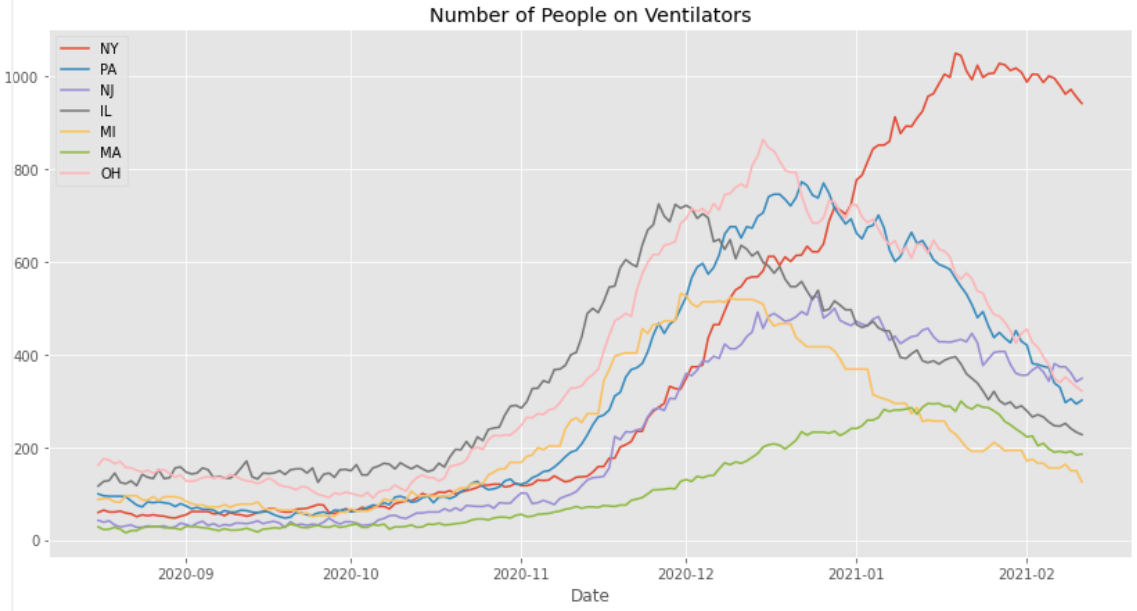
date	state	death	inIcuCurrently	onVentilatorCurrently	positive	hospitalizedCurrently
2021-02-12	CA	46002.0	2930.0	0.0	3381615.0	10505.0
2021-02-12	TX	40095.0	2582.0	0.0	2541845.0	8607.0
2021-02-12	NY	36882.0	1358.0	941.0	1512690.0	7068.0
2021-02-12	FL	29061.0	0.0	0.0	1781450.0	4825.0
2021-02-12	PA	22959.0	496.0	286.0	888256.0	2548.0
2021-02-12	NJ	22393.0	525.0	336.0	740062.0	2565.0

In [51]:
```
df_states['state'].unique() # list of states to iterate through
```

Out[51]:
```
array(['WA', 'MA', 'VA', 'FL', 'NJ', 'NE', 'IN', 'MI', 'RI', 'WY', 'NY',
       'PA', 'TX', 'VT', 'WI', 'IL', 'HI', 'NC', 'CO', 'CA', 'AZ', 'GA',
       'NH', 'OR', 'SC', 'MD', 'DC', 'NM', 'TN', 'OH', 'NV', 'IA', 'KY',
       'KS', 'AR', 'DE', 'AK', 'MN', 'WV', 'ID', 'LA', 'CT', 'AL', 'MO',
       'ME', 'MT', 'MS', 'UT', 'SD', 'ND', 'OK', 'GU', 'AS', 'MP', 'VI',
       'PR'], dtype=object)
```

In [52]:

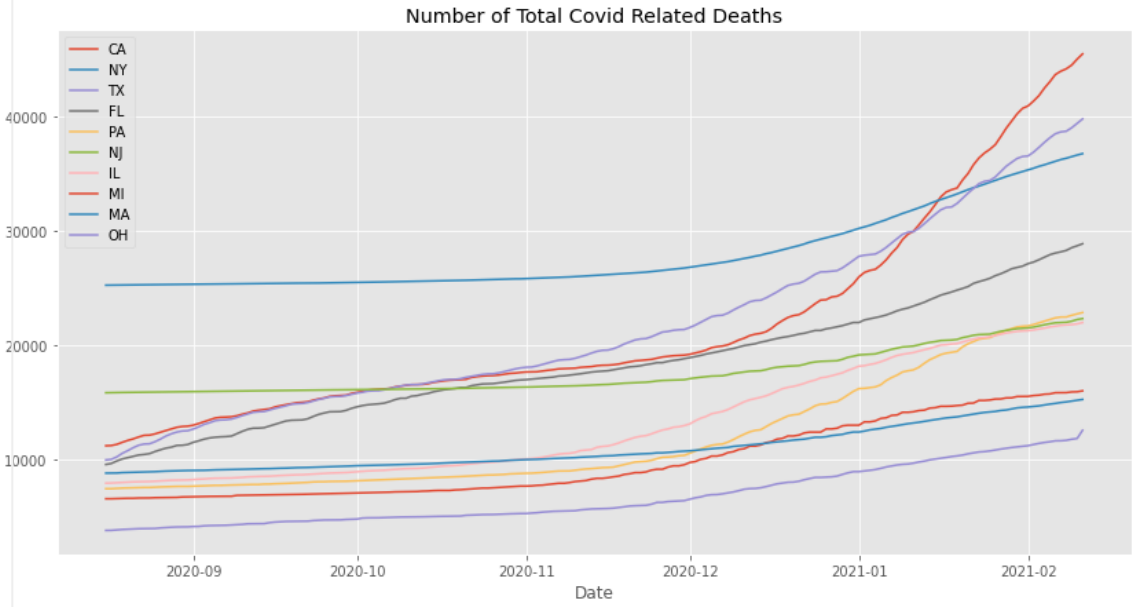
```
# for loop iterates through shortened list and prints ventilator usage
# for the trailing 180 day period.
state_postal = ['NY', 'PA', 'NJ','IL','MI','MA','OH']

fig = plt.figure(figsize=(14,7));

for state in state_postal:
    df_individual = df_states[df_states['state']==state]['onVentilatorCu
    df_plot = df_individual.iloc[(df_individual.index.argmax()-180):(df_
    plt.plot(df_plot,label=f'{state}');
    plt.title('Number of People on Ventilators')
    plt.xlabel('Date')
    plt.ylabel('People')
    plt.legend();
```



Contents

In [53]:

```
# same as above graph - all states have death data, this is a graph of th
# states with the highest covid mortality
state_postal = ['CA', 'NY', 'TX', 'FL', 'PA', 'NJ','IL','MI','MA','OH'] #
# some do not have ventilator data reported.

fig = plt.figure(figsize=(14,7));

for state in state_postal:
    df_individual = df_states[df_states['state']==state].death.sort_index
    df_plot = df_individual.iloc[(df_individual.index.argmax()-180):(df_i
    plt.plot(df_plot,label=f'{state}');
    plt.title('Number of Total Covid Related Deaths')
    plt.xlabel('Date')
    plt.ylabel('Deaths')
    plt.legend();
```



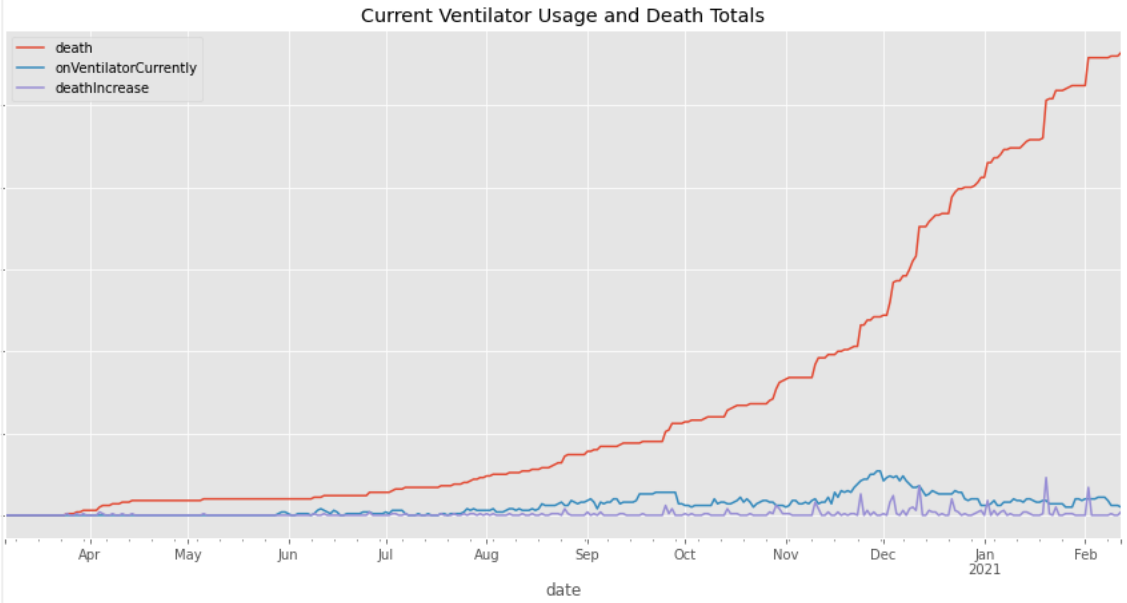
Contents

1.3.3 Plot Alaska Death Count

In [54]:

```
df_AK = df_states[df_states['state']=='AK'] # just look at Alaska for nou
```

In [55]:
```
fig = plt.figure(figsize=(15,7));

df_AK['death'].plot(legend=True,title='Current Ventilator Usage and Death
df_AK['onVentilatorCurrently'].plot(legend=True);
df_AK['deathIncrease'].plot(legend=True);
```



Ventilator usage in Alaska peaks right before the end of December. deathIncrease is the rate of death, or 'volume' of death. Spikes in that line correspond to a steeper increase in deaths along the red trend.

2 Modeling and Forecasts

2.0.1 Alaska SARIMA Model - Initial Modeling

In [56]:
```
df_AK = df_AK.sort_index()
```

Contents

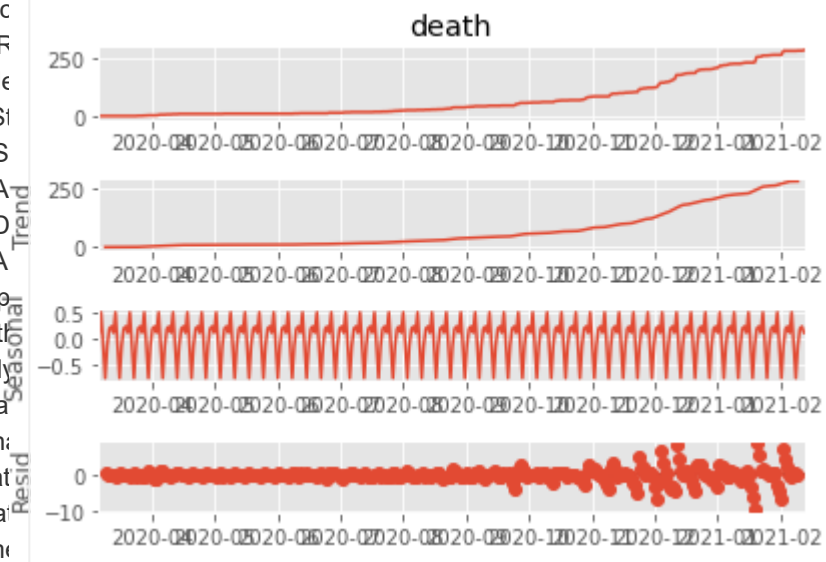
In [57]:
```
df_AK = df_AK.dropna(subset=['death'])
df_AK = df_AK.dropna(subset=['onVentilatorCurrently'])
```

In [58]:
```
df_alaska = pd.DataFrame(df_AK)
```

In [59]:
```
print(df_alaska.index.min())
print(df_alaska.index.max())
print('Length of dataframe: ' , len(df_alaska))
```

```
2020-03-06 00:00:00
2021-02-12 00:00:00
Length of dataframe:  344
```

In [60]:
```
sd(df_alaska['death'], model='additive').plot(); # alaska = seasonal
```



In [61]:
```
stepwise_fit = auto_arima(df_alaska['death'],start_p=0,start_q=0,max_p=10
                          max_q=10, seasonal=True, seasonal_test='ocsb',
                          n_jobs=-1,stepwise=True)
```

Contents

In [62]:
```
model = SARIMAX(df_alaska['death'], order=stepwise_fit.order,seasonal_or
model.summary()
```

Out[62]:

SARIMAX Results

Dep. Variable:	death	**No. Observations:**	344
Model:	SARIMAX(0, 2, 1)	**Log Likelihood**	-771.585
Date:	Sat, 13 Feb 2021	**AIC**	1547.170
Time:	10:52:41	**BIC**	1554.840
Sample:	03-06-2020	**HQIC**	1550.225
	- 02-12-2021		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-0.9661	0.010	-94.202	0.000	-0.986	-0.946
sigma2	5.2930	0.095	55.531	0.000	5.106	5.480

Ljung-Box (L1) (Q):	0.07	**Jarque-Bera (JB):**	19731.69
Prob(Q):	0.79	**Prob(JB):**	0.00
Heteroskedasticity (H):	102.42	**Skew:**	5.22
Prob(H) (two-sided):	0.00	**Kurtosis:**	38.72

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Contents

In [63]:
```
# dont forget get_predict
predictions_AK = model.get_forecast(30)
```

In [64]:
```
1 predictions_AK.predicted_mean
```

Out[64]:
```
2021-02-13    283.716071
2021-02-14    285.432141
2021-02-15    287.148212
2021-02-16    288.864283
2021-02-17    290.580353
2021-02-18    292.296424
2021-02-19    294.012495
2021-02-20    295.728565
2021-02-21    297.444636
2021-02-22    299.160706
2021-02-23    300.876777
2021-02-24    302.592848
2021-02-25    304.308918
2021-02-26    306.024989
2021-02-27    307.741060
2021-02-28    309.457130
2021-03-01    311.173201
2021-03-02    312.889272
2021-03-03    314.605342
```

Contents

In [65]:
```
1 predictions_AK.predicted_mean
2 predictions_AK.conf_int(alpha=.05)
```

Out[65]:

	lower death	upper death
2021-02-13	279.206865	288.225277
2021-02-14	278.946285	291.917998
2021-02-15	279.070661	295.225763
2021-02-16	279.381611	298.346954
2021-02-17	279.803728	301.356978
2021-02-18	280.299039	304.293809
2021-02-19	280.845396	307.179593
2021-02-20	281.428635	310.028495
2021-02-21	282.039103	312.850169
2021-02-22	282.669908	315.651505
2021-02-23	283.315951	318.437603

In [66]:
```
1 stepwise_fit.order
```

Out[66]: (0, 2, 1)

In [67]:
```
1 stepwise_fit.summary()
```

Out[67]:

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	344
Model:	SARIMAX(0, 2, 1)	**Log Likelihood**	-770.318
Date:	Sat, 13 Feb 2021	**AIC**	1546.637
Time:	10:52:43	**BIC**	1558.141
Sample:	0	**HQIC**	1551.220
	- 344		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	0.0067	0.004	1.851	0.064	-0.000	0.014
ma.L1	-0.9870	0.011	-88.631	0.000	-1.009	-0.965
sigma2	5.2394	0.186	28.104	0.000	4.874	5.605

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	20653.57
Prob(Q):	1.00	**Prob(JB):**	0.00
Heteroskedasticity (H):	76.47	**Skew:**	5.36
Prob(H) (two-sided):	0.00	**Kurtosis:**	39.53

Warnings:

[1] Covariance matrix calculated using the outer product of gradients (complex-step).

In [68]:
```
1 length = len(df_alaska)-45
```

In [69]:
```
1 train_data = df_alaska.iloc[:length]
2 test_data = df_alaska.iloc[length:]
```

Contents

In [70]:
```
model = sm.tsa.statespace.SARIMAX(train_data['death'], order=stepwise_fi
res = model.fit(disp=False)
print(res.summary()) # high p values indicate difficulty in modeling.
```

```
                               SARIMAX Results
==========================================================================
===
Dep. Variable:                    death   No. Observations:
299
Model:                SARIMAX(0, 2, 1)   Log Likelihood                -594.
695
Date:                 Sat, 13 Feb 2021   AIC                           1193.
389
Time:                         10:52:44   BIC                           1200.
777
Sample:                     03-06-2020   HQIC                          1196.
347
                          - 12-29-2020
Covariance Type:                   opg
==========================================================================
===
                 coef    std err          z      P>|z|      [0.025      0.9
75]
--------------------------------------------------------------------------
---
ma.L1         -0.9494      0.014    -66.756      0.000      -0.977      -0.
922
sigma2         3.1868      0.093     34.116      0.000       3.004       3.
370
==========================================================================
========
Ljung-Box (L1) (Q):                   0.47   Jarque-Bera (JB):
1391.62
Prob(Q):                              0.49   Prob(JB):
0.00
Heteroskedasticity (H):              70.71   Skew:
4.78
Prob(H) (two-sided):                  0.00   Kurtosis:
34.34
==========================================================================
========

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (comp
lex-step).
```

In [71]:
```
start = len(train_data)
end = len(train_data) + len(test_data) - 1
```

In [72]:
```
predictions_AK = res.predict(start,end,typ='endogenous').rename('SARIMAX
```

Contents

In [73]:
```
1 train_data.index
```

Out[73]:
```
DatetimeIndex(['2020-03-06', '2020-03-07', '2020-03-08', '2020-03-09',
               '2020-03-10', '2020-03-11', '2020-03-12', '2020-03-13',
               '2020-03-14', '2020-03-15',
               ...
               '2020-12-20', '2020-12-21', '2020-12-22', '2020-12-23',
               '2020-12-24', '2020-12-25', '2020-12-26', '2020-12-27',
               '2020-12-28', '2020-12-29'],
              dtype='datetime64[ns]', name='date', length=299, freq=None)
```


In [74]:
```
1 test_data.index
```

Out[74]:
```
DatetimeIndex(['2020-12-30', '2020-12-31', '2021-01-01', '2021-01-02',
               '2021-01-03', '2021-01-04', '2021-01-05', '2021-01-06',
               '2021-01-07', '2021-01-08', '2021-01-09', '2021-01-10',
               '2021-01-11', '2021-01-12', '2021-01-13', '2021-01-14',
               '2021-01-15', '2021-01-16', '2021-01-17', '2021-01-18',
               '2021-01-19', '2021-01-20', '2021-01-21', '2021-01-22',
               '2021-01-23', '2021-01-24', '2021-01-25', '2021-01-26',
               '2021-01-27', '2021-01-28', '2021-01-29', '2021-01-30',
               '2021-01-31', '2021-02-01', '2021-02-02', '2021-02-03',
               '2021-02-04', '2021-02-05', '2021-02-06', '2021-02-07',
               '2021-02-08', '2021-02-09', '2021-02-10', '2021-02-11',
               '2021-02-12'],
              dtype='datetime64[ns]', name='date', freq=None)
```

In [75]:
```
1 predictions_AK = pd.DataFrame(predictions_AK)
```

In [76]:
```
1 predictions_AK.index.name = 'date'
```

Compare Test Data with Predictions

In [77]:
```
1 train_data.index.freq = 'D'
2 test_data.index.freq = 'D' # -1D is reverse index, ie most recent date is
3 # perform sort_index on dataframe to correct. set frequencies to match fo
4 # on same visualization
```

In [78]:
```
1 pd.DataFrame(test_data['death']).info()
```

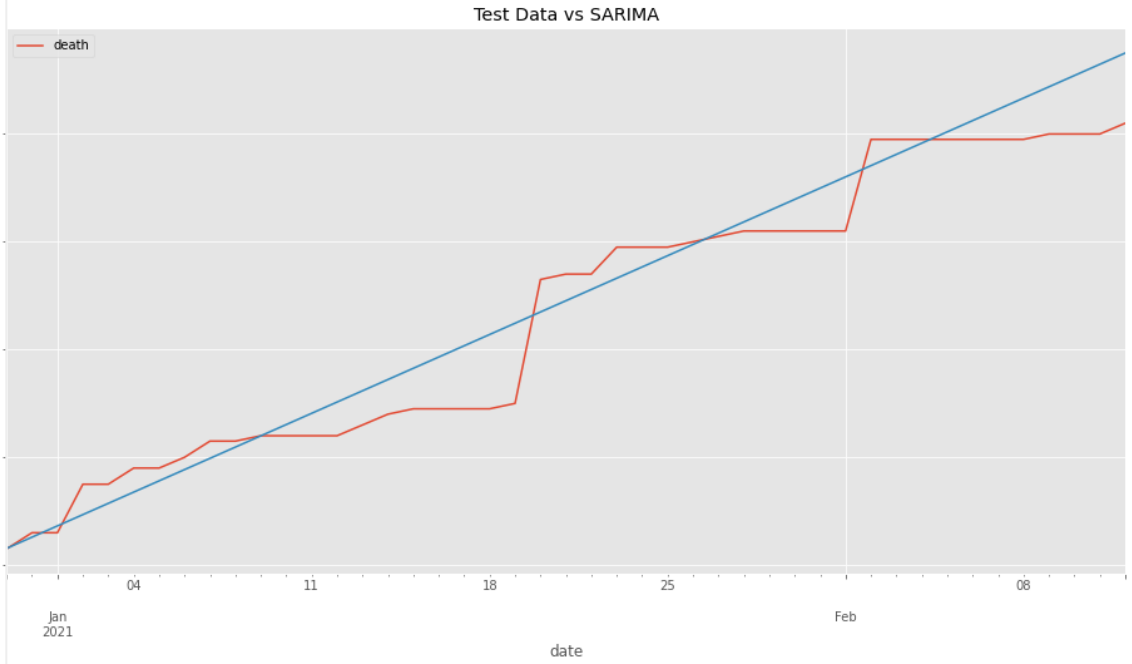
```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 45 entries, 2020-12-30 to 2021-02-12
Freq: D
Data columns (total 1 columns):
#   Column  Non-Null Count  Dtype
---  ------  --------------  -----
0   death   45 non-null     float64
dtypes: float64(1)
memory usage: 720.0 bytes
```

In [79]:
```
predictions_AK.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 45 entries, 2020-12-30 to 2021-02-12
Freq: D
Data columns (total 1 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   SARIMAX(0,2,1) Predictions  45 non-null     float64
dtypes: float64(1)
memory usage: 720.0 bytes
```

Contents

In [80]:
```
pd.DataFrame(test_data['death']).plot(figsize=(16,8),legend=True,title='
plt.plot(pd.DataFrame(predictions_AK))
plt.show()
```



In [81]:

```
model = sm.tsa.statespace.SARIMAX(df_alaska['death'], order=stepwise_fit
res = model.fit(disp=False)
print(res.summary())
```

```
                               SARIMAX Results
==========================================================================
===
Dep. Variable:                   death   No. Observations:
344
Model:               SARIMAX(0, 2, 1)   Log Likelihood                -771.
585
Date:                Sat, 13 Feb 2021   AIC                           1547.
170
Time:                        10:52:48   BIC                           1554.
840
Sample:                    03-06-2020   HQIC                          1550.
225
                         - 02-12-2021
Covariance Type:                 opg
==========================================================================
===
                 coef    std err          z      P>|z|      [0.025      0.9
75]
--------------------------------------------------------------------------
---
ma.L1         -0.9661      0.010    -94.202      0.000      -0.986      -0.
946
sigma2         5.2930      0.095     55.531      0.000       5.106       5.
480
==========================================================================
========
Ljung-Box (L1) (Q):                   0.07   Jarque-Bera (JB):
19731.69
Prob(Q):                              0.79   Prob(JB):
0.00
Heteroskedasticity (H):             102.42   Skew:
5.22
Prob(H) (two-sided):                  0.00   Kurtosis:
38.72
==========================================================================
========

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (comp
lex-step).
```

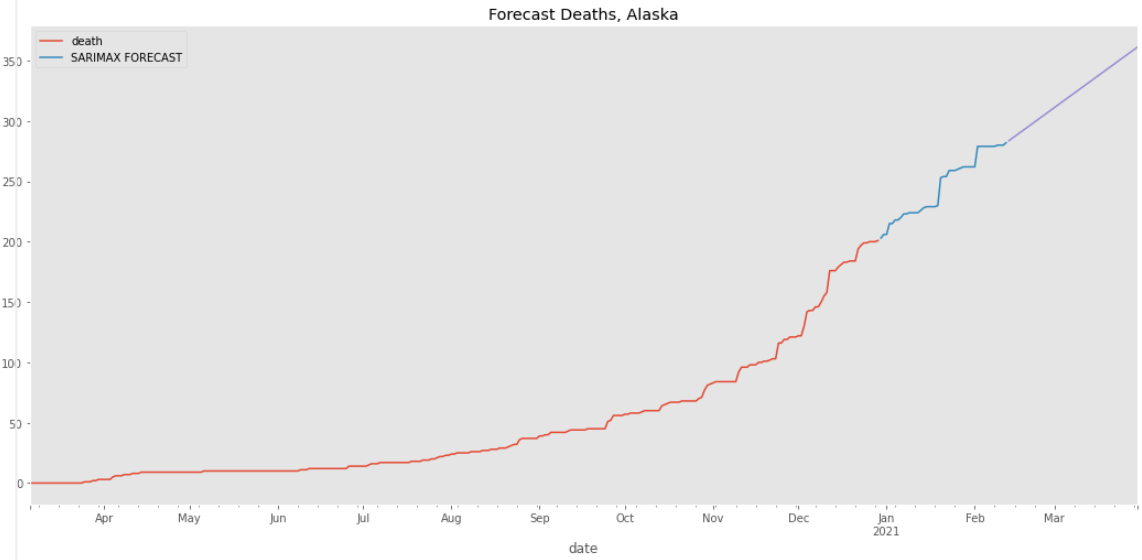
In [82]:

```
fcast = res.predict(start=len(df_AK),end=len(df_AK)+45, typ='endogenous'
```

Contents

In [83]:

```
fig, ax = plt.subplots()

train_data['death'].plot(figsize=(16,8),legend=True,ylabel='Deaths',title
test_data['death'].plot(grid=True);
fcast.plot(legend=True,figsize=(18,8));
ax.grid();
plt.show();
```

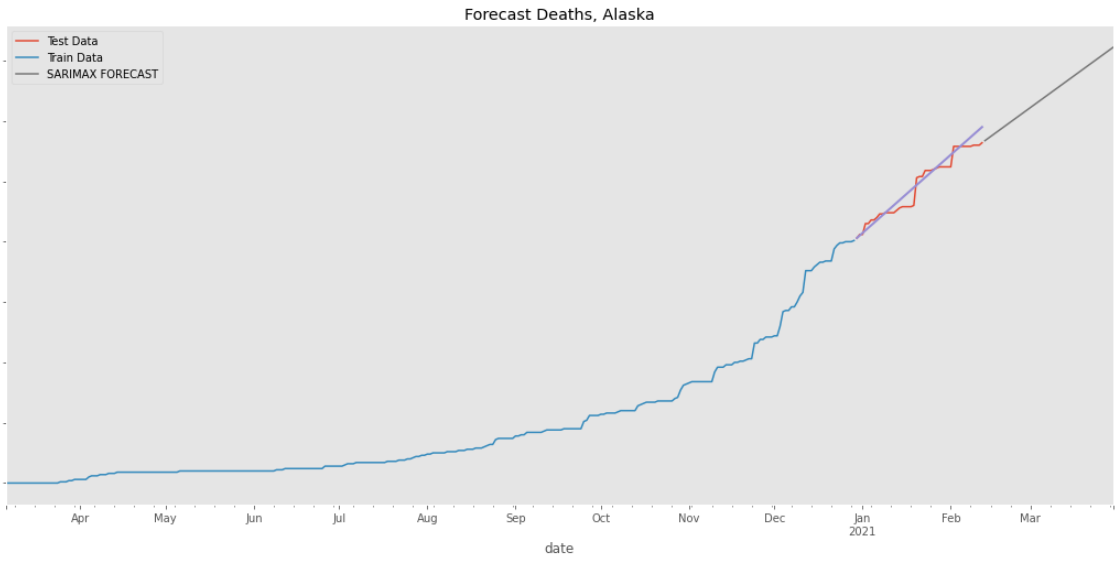


Contents

Below graph will show prediction data against test data as well as a separate future forecast.

In [84]:

```
fig, ax = plt.subplots()

test_data['death'].plot(figsize=(16,8),legend=True,title='Forecast Deaths
train_data['death'].plot(figsize=(16,8),legend=True,ylabel='Deaths',grid=
plt.plot(predictions_AK, linewidth=2); # 'PREDICTIONS' FROM END OF TRAIN
fcast.plot(legend=True,figsize=(18,8)); # SARIMA FORECAST
ax.grid();
plt.show();
```



Using Auto Arima to determine order and seasonal order for the SARIMA model is pretty effective here and provides a valid forecast. Given this run through the process, future forecasts will forecast an exogenous variable and plug that forecast back into the model.

2.1 SARIMAX Modeling

2.1.1 New York State Ventilator Usage Forecast

NY State was one of the first states to experience the worst of the pandemic. In addition to this, they are one of the most populous states in the country, and have had a spike in Covid cases post-holiday season 2020. California just passed barely passed New York state in total deaths *related (not directly caused) by COVID-19, making New York the state with the second most deaths in the United States.

Contents

In [85]:

```
# change to True, run cell to follow link
open_links = False

import webbrowser

if open_links == True:
    webbrowser.open("https://deadline.com/2021/02/california-south-africa
    webbrowser.open("https://www.cityandstateny.com/articles/politics/new
    webbrowser.open("http://www.op.nysed.gov/COVID-19_EO.html#") # new yo
    # the number of healtchare workers
```

Contents

In [86]:

```
df = state_dataframe(df_states, 'NY')
```

```
Successfully returned indexed dataframe for NY
```

In [87]:

```
df_ref = state_dataframe(df_states, 'NY')
```

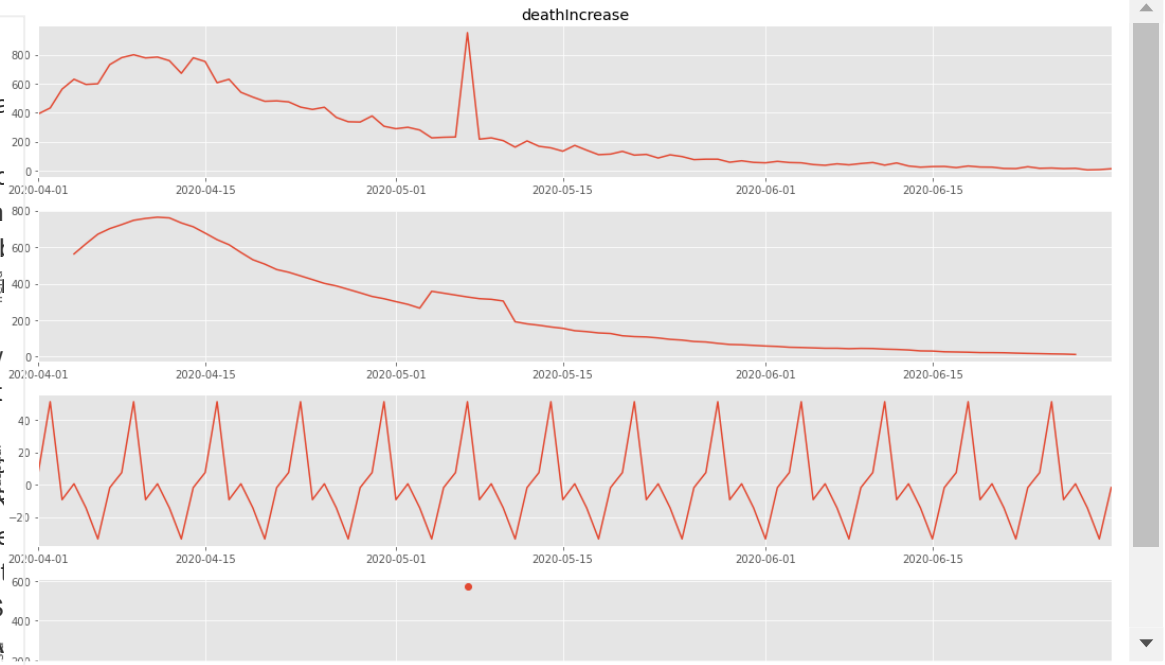
```
Successfully returned indexed dataframe for NY
```

In [88]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 348 entries, 2020-03-02 to 2021-02-12
Freq: D
Data columns (total 7 columns):
#   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
0   state                   348 non-null    object
1   death                   348 non-null    float64
2   inIcuCurrently          348 non-null    float64
3   onVentilatorCurrently   348 non-null    float64
4   positive                348 non-null    float64
5   hospitalizedCurrently   348 non-null    float64
6   deathIncrease           348 non-null    int64
dtypes: float64(5), int64(1), object(1)
memory usage: 21.8+ KB
```

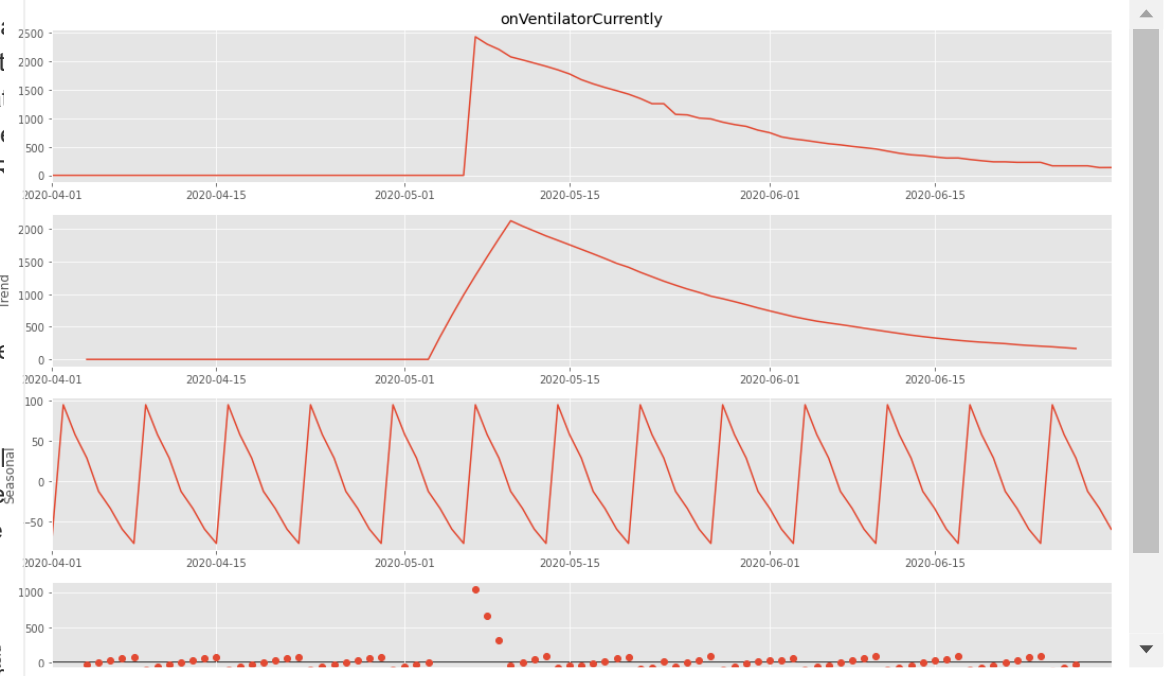
In [89]:
```
plt.rcParams['figure.figsize']=(15,10);
sd(df.loc['04-2020':'06-2020']['deathIncrease']).plot();
```



Contents

In [90]:
```
# death increase seasonal decomp plot shows that we have near weekly seas
```

In [91]:
```
plt.rcParams['figure.figsize']=(15,10);
sd(df.loc['04-2020':'06-2020']['onVentilatorCurrently']).plot();
```



In [92]:
```
# seasonality peaks match the peaks of the deathIncrease seasonal decomp
```

In [93]:

```
stepwise_fit, stepwise_full, results, results_full = arima_tune(df, 'onV
                                                                days_to_
                                                                verbose=
# train days arg defaults to 270 days, but can be changed. seasonality c
# see docstring for further details
# forecasting 30 days out into the future with a seasonality length of 6
```

```
ARIMA order is:  (0, 2, 2)
Seasonal ARIMA order is:  (0, 0, 1, 7)
Use ARIMA object stepwise_fit to store ARIMA and seasonal ARIMA orders in
variables.
```

SARIMAX Results

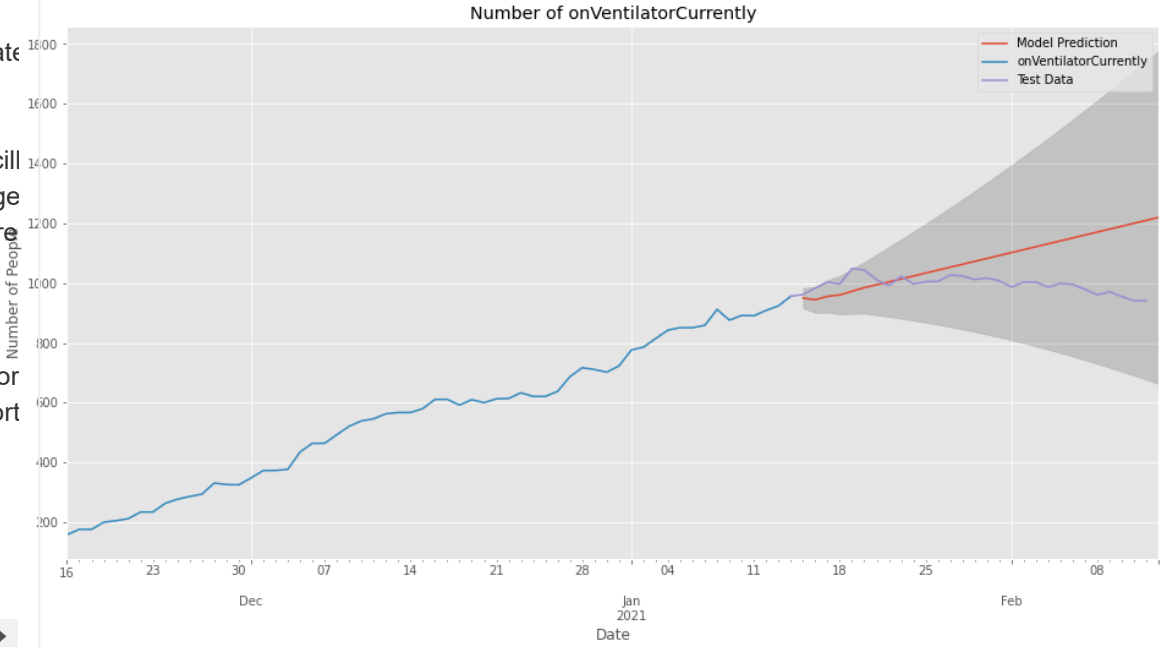
Dep. Variable:	onVentilatorCurrently	**No. Observations:**	240
Model:	SARIMAX(0, 2, 2)x(0, 0, [1], 7)	**Log Likelihood**	-881.528
Date:	Sat, 13 Feb 2021	**AIC**	1771.057
Time:	10:53:04	**BIC**	1784.774
Sample:	05-19-2020	**HQIC**	1776.591
	- 01-13-2021		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]

Reasonable q-q plot. Model summary is good overall.

In [94]:

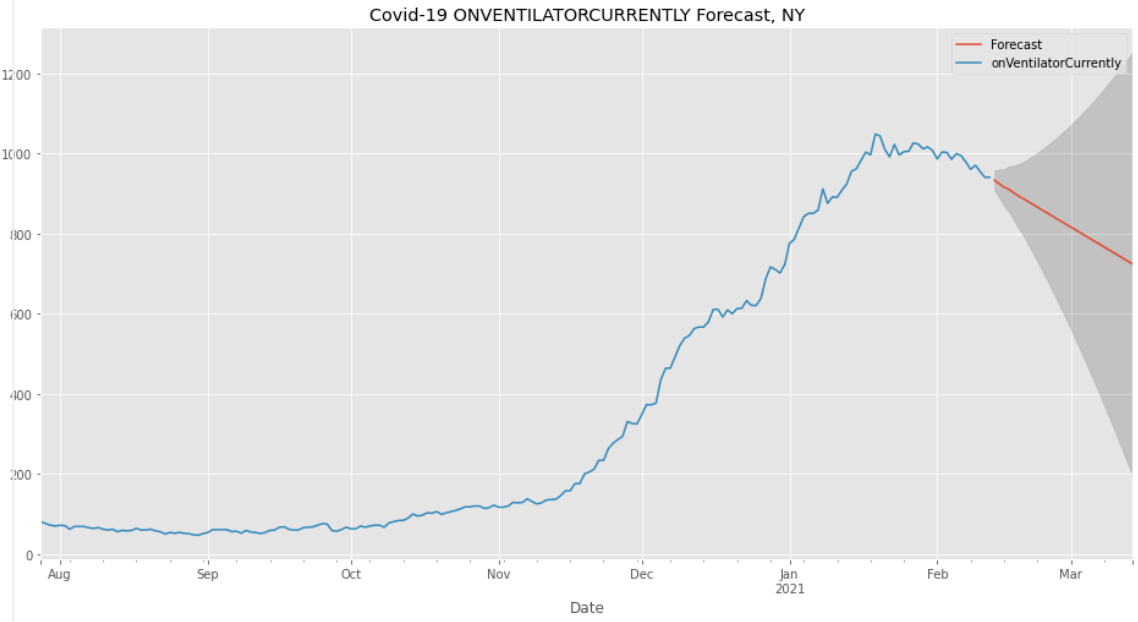
```
evaluate_predictions(results, df, 'onVentilatorCurrently', stepwise_fit=
                     alpha=.05, days_to_forecast=30)
# plot training time and test time
# this evaluates the model using a train test split while also providing
# a forecast of confidence intervals with an alpha of .05.
```



Contents

In [95]:

```
exog_forecast, forecast_obj = build_SARIMAX_forecast(model=results_full,
                                                     dataframe=df,
                                                     target_column='onVe
                                                     days_to_forecast=30
                                                     stepwise_fit=stepwi
                                                     alpha=.05, state_po

# this is a forecast of those people who are currently on ventilators in
# involving Covid-19. This forecast data will be used to enhance the ove
# forecast of death or deathIncrease (rate of death)
```

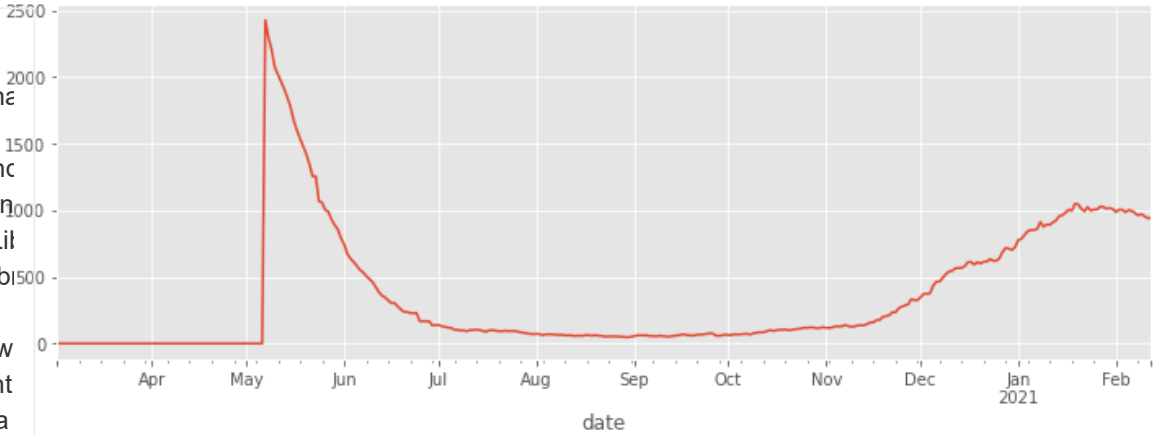



New York ventilator data has shown an improvement more recently. The recent downward trend here is encouraging, and it is likely to continue.

The initial spike in ventilator usage in New York (seen below) was likely a result of not knowing how to properly treat the virus. As time has progressed we have learned that ventilators should only be used in the most severe cases. Additional methods like keeping Covid patients on their stomachs while using the ventilator has been claimed to be more effective.

Finally, this forecast will be used to influence a forecast of deaths in New York.

In [96]:
```
df['onVentilatorCurrently'].plot(figsize=(12,4)); # see spike here coming
# from initial May 2020 reporting.
```



Contents

2.1.2 New York State Deaths Forecast

Modeled using New York's Ventilator Usage Forecast

In [97]:
```
stepwise_fit, df_forecast = get_exogenous_forecast_dataframe(dataframe=d
                                                             original_da
                                                             exog_foreca
                                                             target_colu
                                                             exogenous_c
                                                             days_to_for
                                                             m_periods=7
```

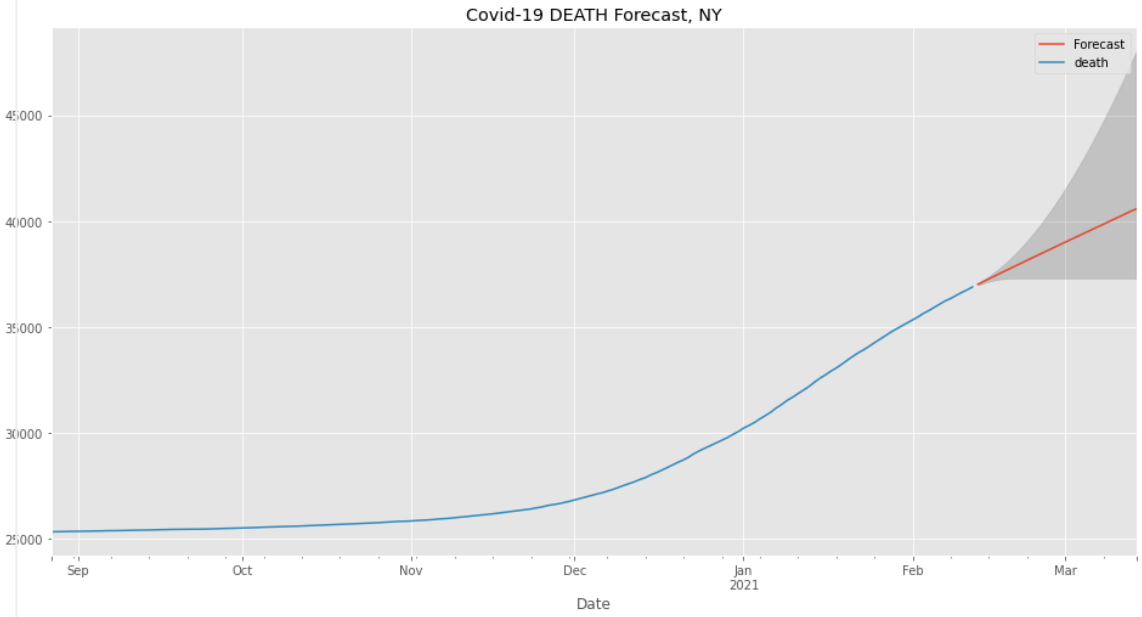
In [98]:
```
df_forecast.tail(5)
```

Out[98]:

	state	death	inIcuCurrently	onVentilatorCurrently	positive	hospitalizedCurrently	deathI
2021-03-10	NaN	NaN	NaN	752.720906	NaN	NaN	
2021-03-11	NaN	NaN	NaN	745.607978	NaN	NaN	
2021-03-12	NaN	NaN	NaN	738.495049	NaN	NaN	
2021-03-13	NaN	NaN	NaN	731.382121	NaN	NaN	
2021-03-14	NaN	NaN	NaN	724.269192	NaN	NaN	

In [99]:
```
# create model
full_exog_model = SARIMAX(df['death'],df['onVentilatorCurrently'],
                          order=stepwise_fit.order,seasonal_order=stepwi
# fit model
model = full_exog_model.fit()
```

In [100]:
```
exog_forecast, results_forecast = build_SARIMAX_forecast(model=model,
                                                          dataframe=df_fo
                                                          target_column='
                                                          days_to_forecas
                                                          stepwise_fit=ste
                                                          alpha=.05,
                                                          original_df=df_
                                                          exogenous_colum
                                                          state_postal_co
```



Above graph implements usage of a function that flattens the lower portion of the confidence interval instead of allowing it to decrease. Deaths remain flat in a best case scenario instead of 'decreasing' in the forecast.

2.1.3 New York Analysis and Recommendations

New York flattened its curve from the beginning of the pandemic until the Thanksgiving holidays when everything suddenly became more difficult. There is a clear increase in deaths beginning around mid November and into Christmas and the New Year, likely a result of family gatherings, social gatherings, and increased survivability in the cold.

There is every possibility that without continuing social distancing and enforcing the wearing of masks that the rate of death will once again increase.

Recommendations for the state include the following:

****Lower the number of people allowed at indoor private gatherings from the current amount of 10. Social gatherings are not economically essential. Certain states have limits of one or two households per private residence which has proven to limit the spread of the disease.***

Increase effort to improve awareness. Covering the mouth but not the nose does not limit the spread of Covid-19. Mandate signage depicting proper mask usage at public establishments.

Continue social distancing policies and reduce the number of outdoor events. The pandemic in New York is not under control.

Contents

2.1.4 California Deaths Forecast

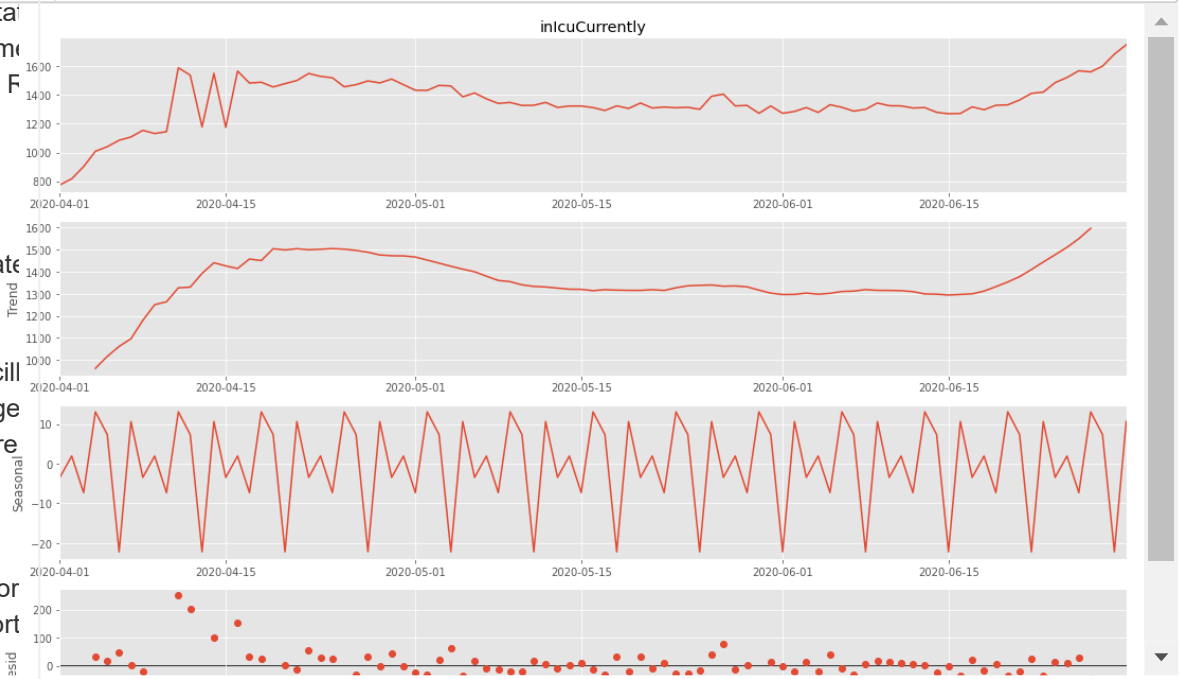
Modeled using ICU Forecast (ICU Forecast not shown)

In [142]:
```
# change to True, run cell to follow link
open_links = False

import webbrowser

if open_links == True:
    webbrowser.open("https://deadline.com/2021/02/california-south-africa
    webbrowser.open("https://covid19.ca.gov/?utm_source=google&utm_mediur
```

In [146]:
```
df_ref = state_dataframe(df_states, 'CA')
```

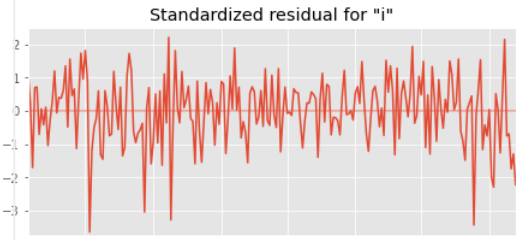
```
Successfully returned indexed dataframe for CA
```

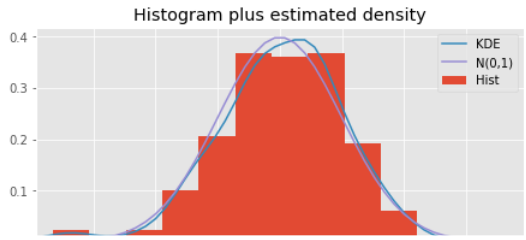
In [147]:
```
plt.rcParams['figure.figsize']=(15,10);
sd(df_ref.loc['04-2020':'06-2020']['inIcuCurrently']).plot();
```



In [148]:
```
state_dataframe, exog_forecast = create_exog_forecast(df_states, 'inIcuC
                                                      m_periods=7, state_
```

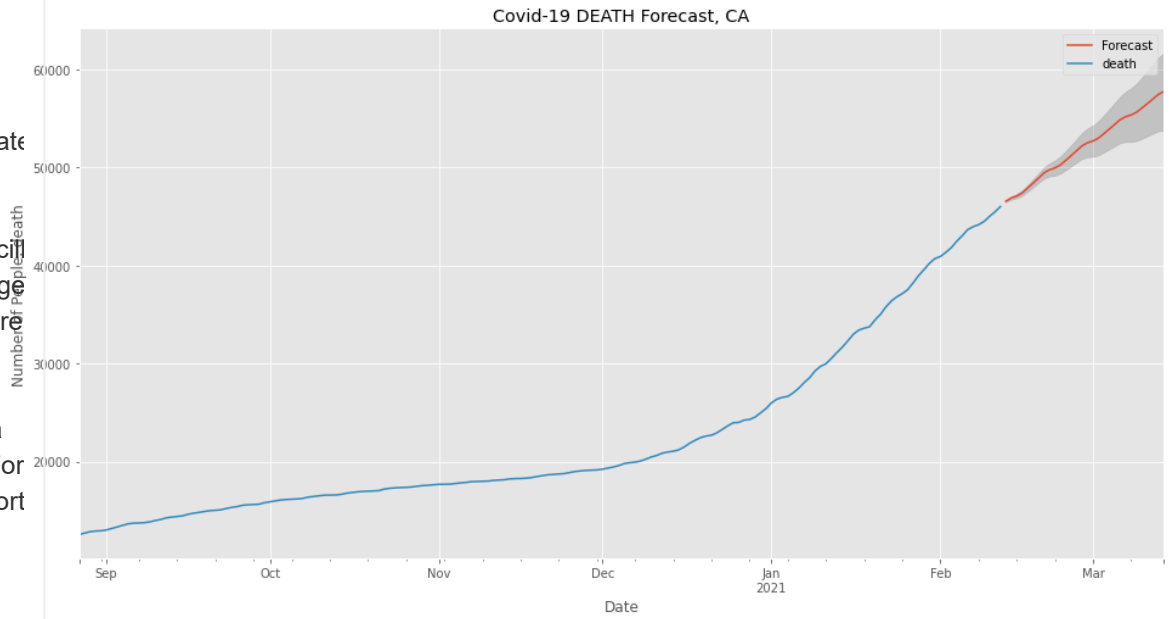
Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).





In [152]:
```
# normal q-q plot
```

In [149]:
```
forecast_object = graph_exog_forecast(dataframe=state_dataframe,
                                      target_column='death',
                                      exogenous_column='inIcuCurrently',
                                      exog_forecast=exog_forecast,
                                      df_ref=df_ref,
                                      alpha=.05, days_to_forecast=30,
                                      train_days=270, m_periods=7,
                                      state_postal_code='CA')
```



Contents

In [150]:
```
forecast_object.predicted_mean[-5:] # projected mean deaths
# by March 14th, 2021 stand at 57,730.
```

Out[150]:
```
2021-03-10    56063.879182
2021-03-11    56515.194316
2021-03-12    56978.468058
2021-03-13    57444.262203
2021-03-14    57729.645022
Freq: D, Name: predicted_mean, dtype: float64
```

In [153]:
```
forecast_object.conf_int()[-5:] # upper confidence interval of 95% foreca
# deaths of over 61,000 by March 14th, 2021.
```

Out[153]:

	lower death	upper death
2021-03-10	52931.486803	59196.271562
2021-03-11	53183.717245	59846.671386
2021-03-12	53443.382517	60513.553599
2021-03-13	53689.901401	61198.623006
2021-03-14	53749.443886	61709.846157

2.1.5 California Analysis and Recommendations

California has experienced an even more drastic increase in deaths since the holidays than any other state. What is happening is an emergency with over 540 Covid deaths on Feb. 11th, 2021 alone. The California state website talks about helping to slow the spread of Covid. Given the circumstances, this language is not urgently and clearly conveying the message that the situation needs immediate attention from each and every individual living in the state.

Recommendations for the state include the following:

* ***Require wearing a mask if an individual is not in or on their private property. Allow no exceptions.***

* ***Prohibit private and public gatherings of 5 or more people unless from the same household.***

* ***The spread of this disease in this state will continue to take lives if people are not made to understand the consequences of selfish behavior. Introduce visual evidence of the rammifications of the virus with an emphasis on personal stories on public social media and television.***

2.1.6 Texas Hospitalized Forecast

In [154]:

```
# change to True, run cell to follow link(s)
open_links = False

import webbrowser

if open_links == True:
    webbrowser.open("https://www.kvue.com/article/news/health/coronavirus
    webbrowser.open("https://www.khou.com/article/news/local/texas/texas-
    webbrowser.open("https://apps.texastribune.org/features/2020/texas-co
```

Contents

Texas is also a populous state and has been inclined to open back up quickly to keep the economy going. The state has not reported ICU or Ventilator numbers to the Covid Tracking Project. A KHOU article cites a decrease in hospitalizations while warning about the continued threat of breaching hospital capacity.

In [157]:

```
df = state_dataframe(df_states, 'TX')
```

```
Successfully returned indexed dataframe for TX
```

In [158]:

```
df_ref = state_dataframe(df_states, 'TX')
```

```
Successfully returned indexed dataframe for TX
```

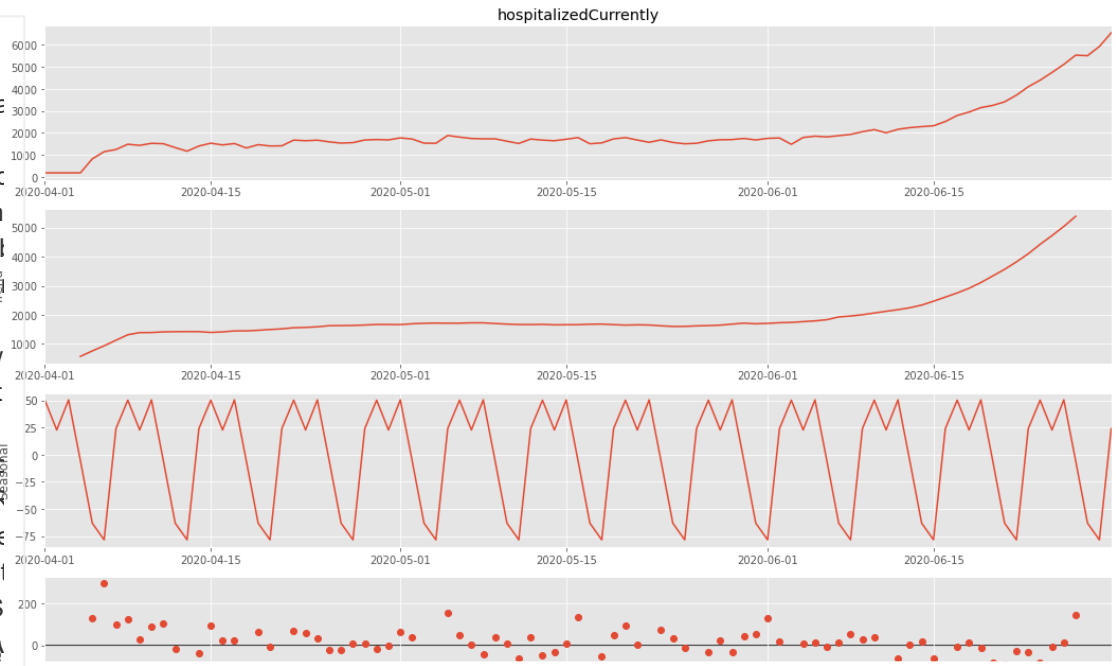
In [182]:

```
df.tail()
```

Out[182]:

date	state	death	inIcuCurrently	onVentilatorCurrently	positive	hospitalizedCurrently	de
2021-02-08	TX	38700.0	2667.0	0.0	2491227.0	9401.0	
2021-02-09	TX	39001.0	2777.0	0.0	2504556.0	9401.0	
2021-02-10	TX	39386.0	2740.0	0.0	2517453.0	9165.0	
2021-02-11	TX	39771.0	2703.0	0.0	2529343.0	8933.0	
2021-02-12	TX	40095.0	2582.0	0.0	2541845.0	8607.0	

In [193]:
```
plt.rcParams['figure.figsize']=(15,10);
sd(df.loc['04-2020':'06-2020']['hospitalizedCurrently']).plot();
```



Contents

In [194]:
```
stepwise_fit, stepwise_full, results, results_full = arima_tune(df, 'hosp
                                                                days_to_
                                                                verbose=
# train days arg defaults to 270 days, but can be changed. seasonality c
# see docstring for further details
```

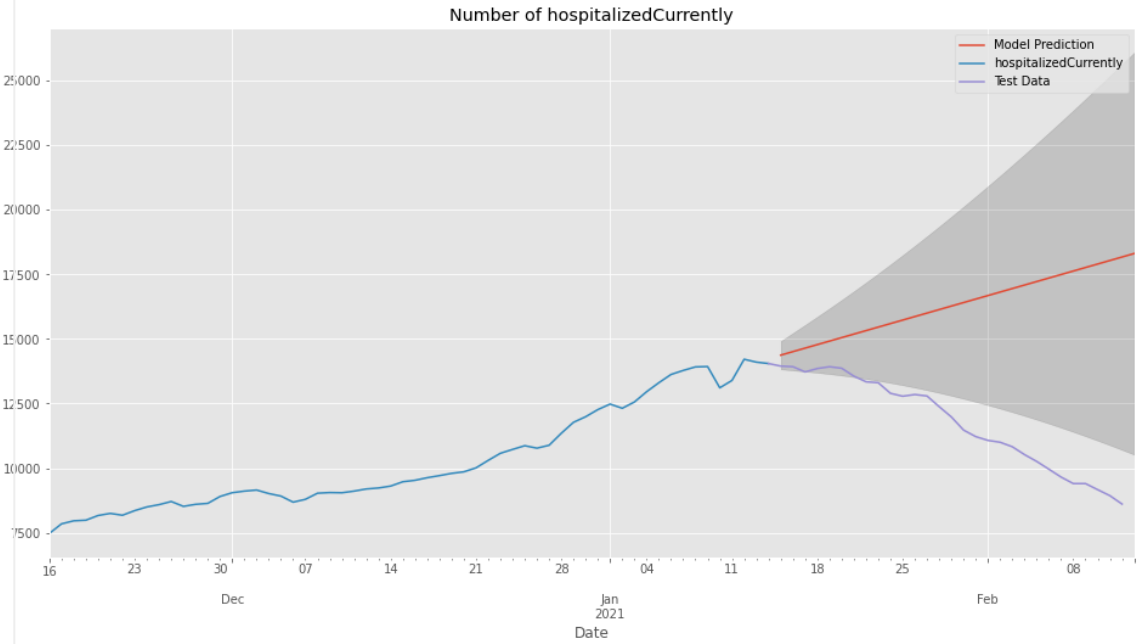
SARIMAX Results

Dep. Variable:	hospitalizedCurrently	**No. Observations:**	240
Model:	SARIMAX(0, 2, 1)	**Log Likelihood**	-1560.634
Date:	Sat, 13 Feb 2021	**AIC**	3125.269
Time:	12:43:32	**BIC**	3132.197
Sample:	05-19-2020	**HQIC**	3128.061
	- 01-13-2021		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]

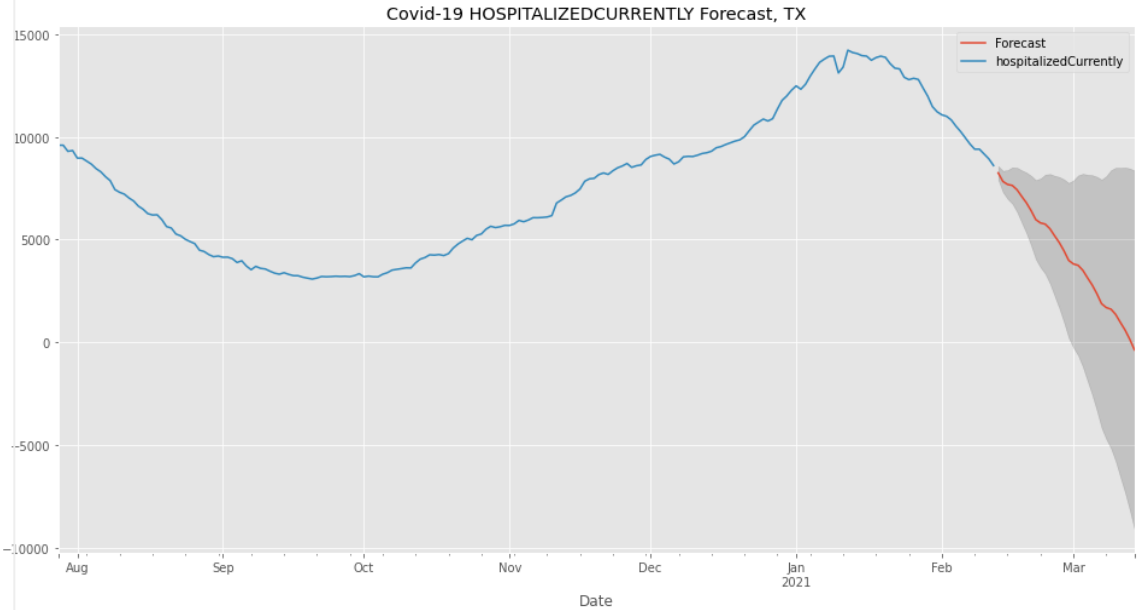
The ends of the q-q plot are not quite in line.

In [195]:
```
evaluate_predictions(results, df, 'hospitalizedCurrently',
                     stepwise_fit=stepwise_fit, alpha=.05, days_to_foreca
# plot training time and test time
```



Contents

In [196]:
```
exog_forecast, forecast_obj = build_SARIMAX_forecast(model=results_full,
                                                     dataframe=df,
                                                     target_column='hosp
                                                     days_to_forecast=30
                                                     alpha=.05,
                                                     state_postal_code='
```



In [197]:
```
# graph of those currently hospitalized and subsequent forecast above sho
# a declining rate of hospitalization with a chance to remain flat.
```


2.1.7 Texas Deaths Forecast

Modeled using Texas's Hospitalized Forecast

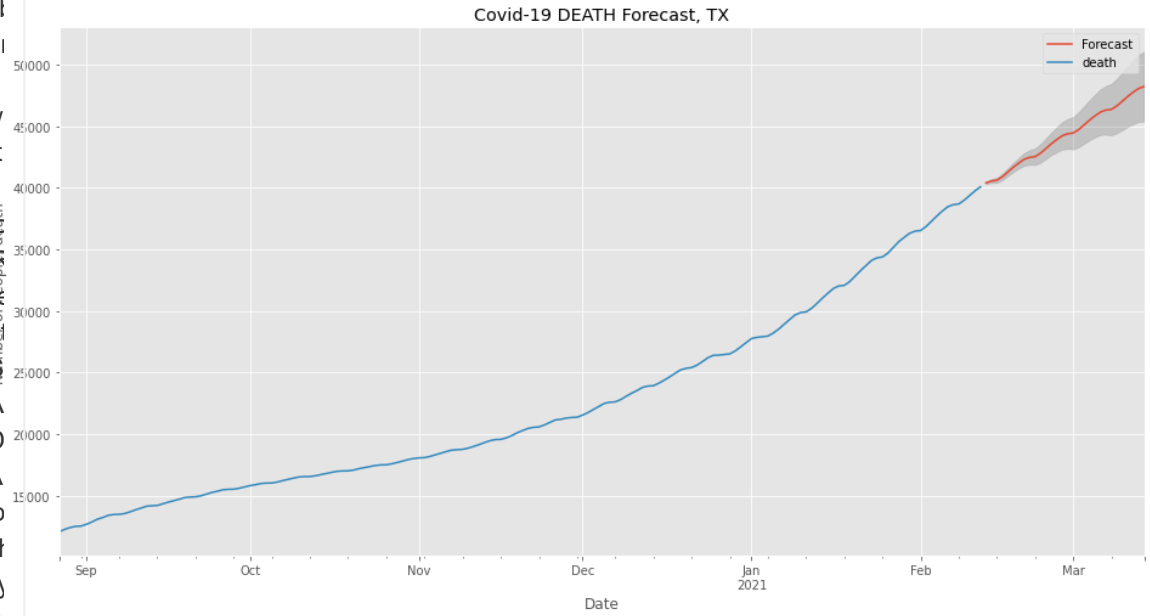
In [198]:
```
stepwise_fit, df_forecast = get_exogenous_forecast_dataframe(dataframe=d
                                                             original_da
                                                             exog_foreca
                                                             target_colu
                                                             exogenous_c
                                                             days_to_for
                                                             m_periods=7
```

In [199]:
```
# get exogenous forecast dataframe will return an extended dataframe
# containing the forecasted exogenous column from build_SARIMAX_forecast
# above after taking in the variable exog_forecast
```

In [200]:
```
full_exog_model = SARIMAX(df['death'],df['hospitalizedCurrently'],
                          order=stepwise_fit.order,seasonal_order=stepwi
```

In [201]:
```
# fit model
model = full_exog_model.fit()
```

In [202]:
```
exog_forecast, results_forecast = build_SARIMAX_forecast(model=model,
                                                          dataframe=df_fo
                                                          target_column='
                                                          days_to_forecas
                                                          stepwise_fit=ste
                                                          alpha=.05,
                                                          original_df=df_
                                                          exogenous_colum
                                                          state_postal_co
```



Contents

In [203]:
```
results_forecast.predicted_mean[-5:]
```

Out[203]:
```
2021-03-10    47041.298532
2021-03-11    47432.913649
2021-03-12    47774.846888
2021-03-13    48082.671287
2021-03-14    48239.111011
Freq: D, Name: predicted_mean, dtype: float64
```

2.1.8 Texas Analysis and Recommendations

The number of people currently hospitalized has decreased in recent weeks and I am forecasting that to continue. However, deaths are forecasted to slow only slightly over the next 30 days. There is some improvement, but it isn't enough.

Recommendations for the state include the following:

**** Limit private social gatherings.***

**** There are those in Texas who believe that the virus is a joke, that the rules don't apply to them, and that there are no consequences. Like California, an awareness campaign with personal stories and visual evidence of what Covid does could help contain the***

**** Make wearing a mask outside of one's private property required.***

Contents

2.1.9 Florida Deaths Forecast

Modeled using Hospitalized Currently Forecast

In [124]:
```
# change to True, run cell to follow link
open_links = False

import webbrowser

if open_links == True:
    webbrowser.open("https://www.newsweek.com/covid-florida-travel-advice
```

Florida currently has 300 new cases of the UK variant of Covid with 0 travel restrictions in place. The spread of the virus within Florida and throughout the rest of the United States as a result of a lack of travel bans and restraint is a serious issue. This section will forecast rate of death, total deaths, and will use the number of people currently hospitalized as an exogeous forecast. (data as of 2-10-2021)

In [125]:
```
df_ref = state_dataframe(df_states, 'FL')
```

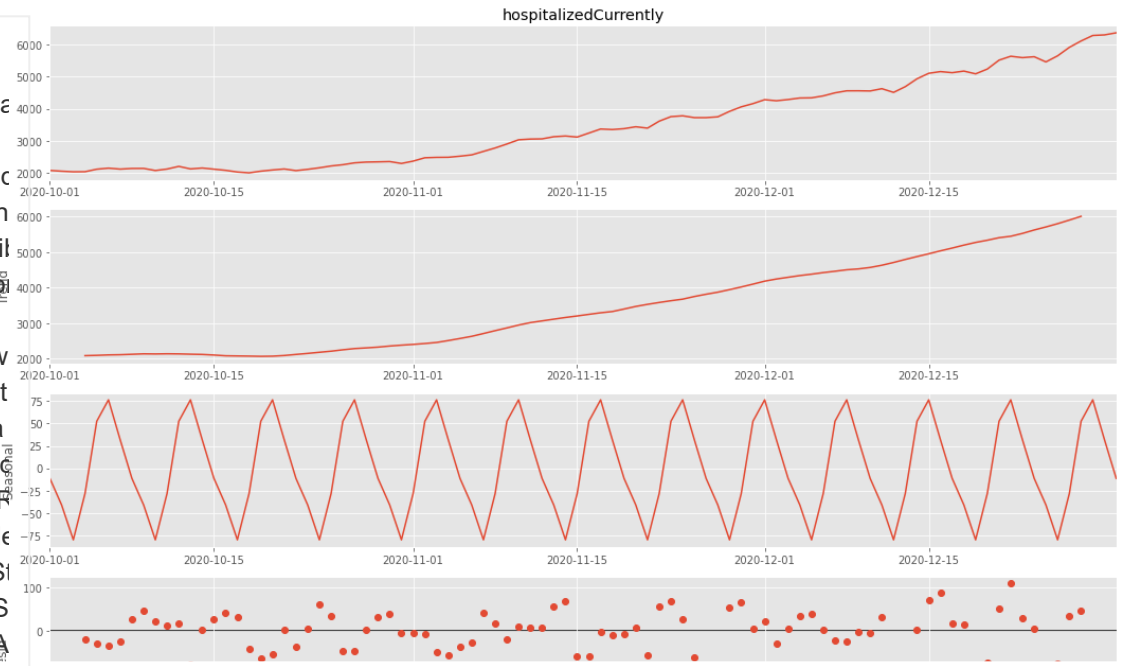
```
Successfully returned indexed dataframe for FL
```

In [126]:
```
df_ref.tail()
```

Out[126]:

date	state	death	inIcuCurrently	onVentilatorCurrently	positive	hospitalizedCurrently	de
2021-02-08	FL	28287.0	0.0	0.0	1751343.0	5381.0	
2021-02-09	FL	28526.0	0.0	0.0	1758254.0	5307.0	
2021-02-10	FL	28691.0	0.0	0.0	1765659.0	5129.0	
2021-02-11	FL	28871.0	0.0	0.0	1774013.0	4906.0	
2021-02-12	FL	29061.0	0.0	0.0	1781450.0	4825.0	

In [127]:
```
plt.rcParams['figure.figsize']=(15,10);
sd(df_ref.loc['10-2020':'12-2020']['hospitalizedCurrently']).plot(); # s
```



Contents

In [138]:
```
state_dataframe, exog_forecast = create_exog_forecast(df_states, 'hospita
                                                      days_to_forecast=45
                                                      state_postal_code=
```

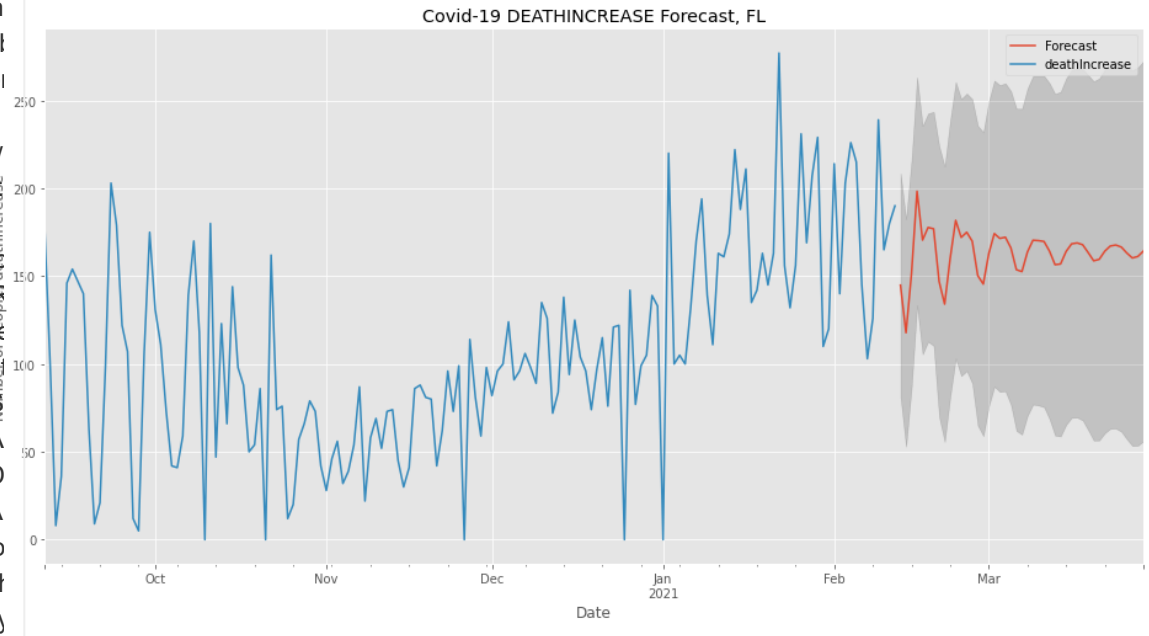
Model:	SARIMAX(0, 1, 0)x(0, 0, [1], 6)	**Log Likelihood**	-1650.364
Date:	Sat, 13 Feb 2021	**AIC**	3304.729
Time:	11:08:55	**BIC**	3311.489
Sample:	05-19-2020	**HQIC**	3307.460
	- 12-29-2020		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ma.S.L6	0.1485	0.234	0.634	0.526	-0.311	0.608
sigma2	2.362e+05	2300.288	102.690	0.000	2.32e+05	2.41e+05

Ljung-Box (L1) (Q):	0.71	**Jarque-Bera (JB):**	335172.89
Prob(Q):	0.40	**Prob(JB):**	0.00
Heteroskedasticity (H):	0.01	**Skew:**	13.52

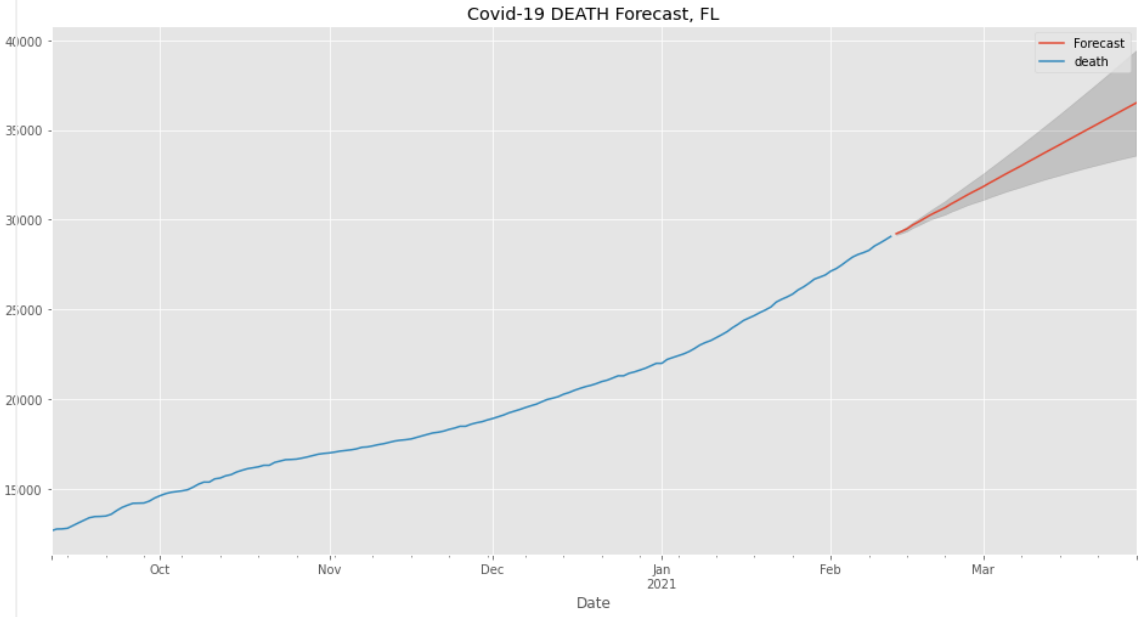
In [129]:

```
forecast_object = graph_exog_forecast(dataframe=state_dataframe,
                                      target_column='deathIncrease',
                                      exogenous_column='hospitalizedCurr
                                      exog_forecast=exog_forecast,
                                      df_ref=df_ref,
                                      alpha=.05, days_to_forecast=45,
                                      train_days=270, m_periods=7,
                                      state_postal_code='FL')
```



Contents

In [130]:
```
forecast_object = graph_exog_forecast(dataframe=state_dataframe,
                                      target_column='death',
                                      exogenous_column='hospitalizedCurr
                                      exog_forecast=exog_forecast,
                                      df_ref=df_ref,
                                      alpha=.05, days_to_forecast=45,
                                      train_days=270, m_periods=7,
                                      state_postal_code='FL')
```



In [131]:
```
forecast_object.predicted_mean[-5:] # projected mean deaths for
# by the end of March
```

Out[131]:
```
2021-03-25    35837.119536
2021-03-26    36003.597226
2021-03-27    36168.891724
2021-03-28    36333.435035
2021-03-29    36498.392480
Freq: D, Name: predicted_mean, dtype: float64
```

Contents

2.1.10 Florida Analysis and Recommendations

Florida does not have any travel restrictions in place. There have been and will continue to be a steady rate of Covid deaths, likely reaching over 35000 by the end of March.

Recommendations for the state include the following:

**** Florida is fully opened - implement travel restrictions and reduce the number of people allowed to privately gather.***

**** Pre and post-Super Bowl footage showed business as usual with zero mask usage. Implement and enforce laws requiring masks in public. There are plenty of states that sponsibly open that are mitigating the spread of this virus. Florida seems to be encouraging the spread.***

**** Common sense is easy. Don't breathe into peoples' faces, wear a mask, and keep your distance.***

Contents

2.1.11 United States Death Forecast

In [132]:

```
# change to True, run cell to follow link(s)
open_links = False

import webbrowser

if open_links == True:
    webbrowser.open("")
```

In [133]:

```
df_whole_US.head()
```

Out[133]:

date	states	positive	negative	pending	hospitalizedCurrently	hospitalizedCumulative
2021-02-12	56	27266230.0	122400369.0	9434.0	71504.0	839119.0
2021-02-11	56	27165660.0	121833298.0	11981.0	74225.0	836774.0
2021-02-10	56	27063243.0	121244702.0	12079.0	76979.0	834314.0
2021-02-09	56	26968049.0	120859564.0	10516.0	79179.0	831088.0
2021-02-08	56	26875063.0	120367478.0	12114.0	80055.0	827944.0

5 rows × 24 columns

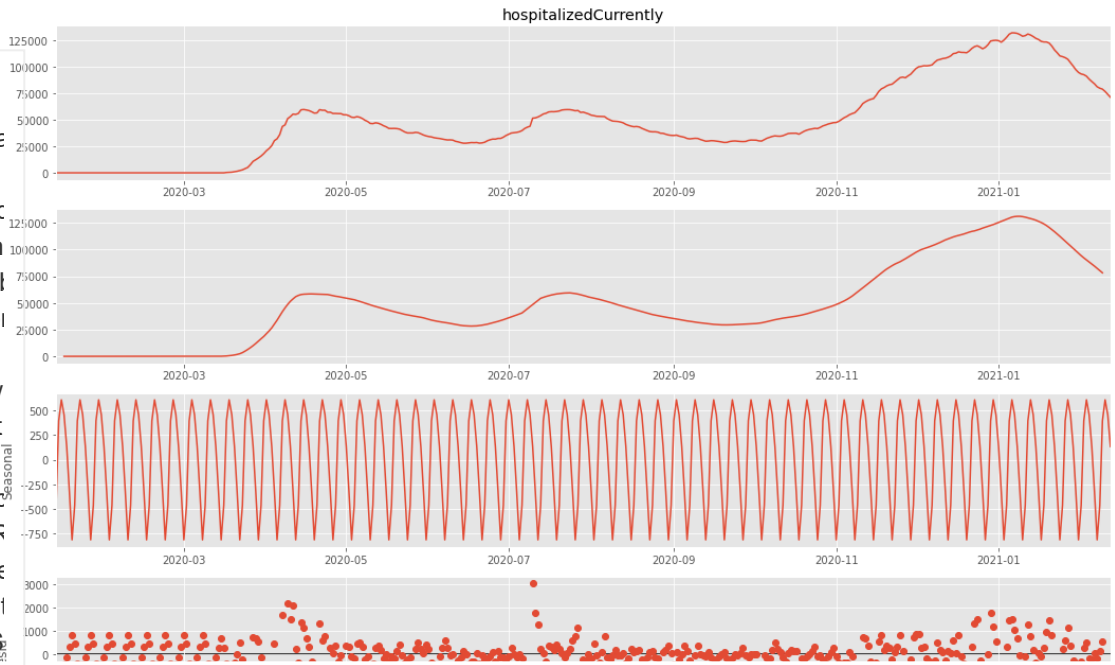
In [134]:

```
columns =  ['death',
            'inIcuCurrently',
            'onVentilatorCurrently',
            'positive',
            'hospitalizedCurrently',
            'deathIncrease']
            # sort_and_clean_df(df_whole_US, columns)
```

In [135]:

```
df_whole_US = sort_and_clean_df(df_whole_US, columns)
```

In [136]:
```
1 sd(df_whole_US['hospitalizedCurrently']).plot();
```



Contents

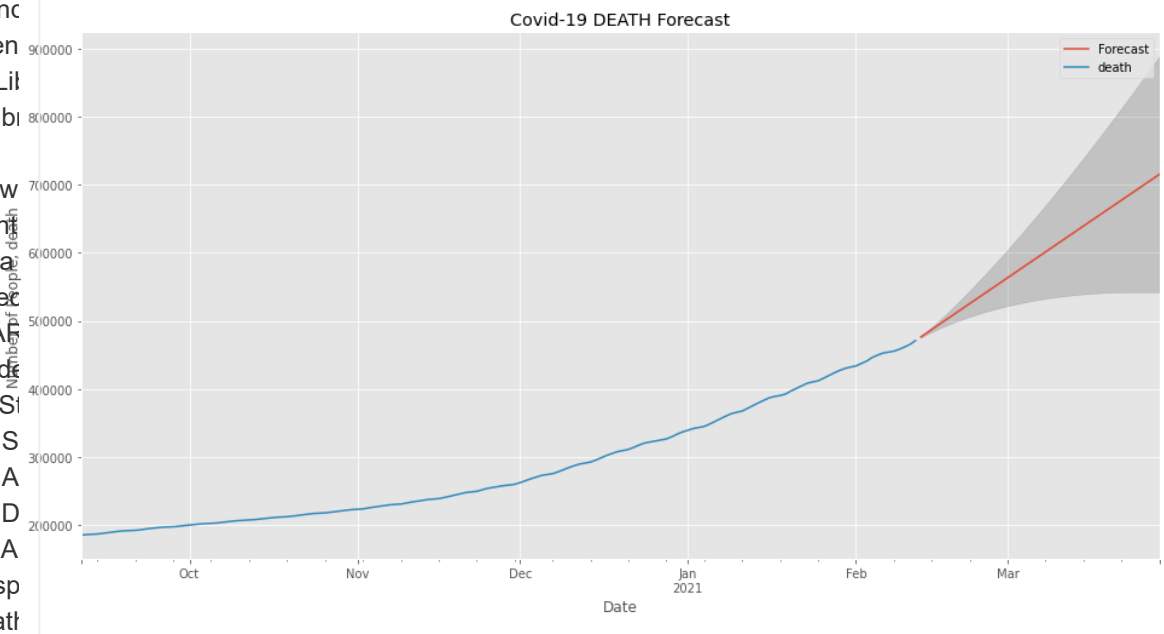
In [137]:
```
1 dataframe, exog_forecast = create_exog_forecast(df_whole_US, 'hospitalize
                                                  days_to_forecast=45, m_pe
```

Dep. Variable:	hospitalizedCurrently	**No. Observations:**	225
Model:	SARIMAX(1, 2, 1)x(1, 0, [], 7)	**Log Likelihood**	-1756.261
Date:	Sat, 13 Feb 2021	**AIC**	3520.522
Time:	11:09:29	**BIC**	3534.005
Sample:	05-19-2020	**HQIC**	3525.970
	- 12-29-2020		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.1993	0.074	2.693	0.007	0.054	0.344
ma.L1	-0.8657	0.055	-15.814	0.000	-0.973	-0.758
ar.S.L7	0.2738	0.035	7.762	0.000	0.205	0.343
sigma2	7.218e+05	2.78e+04	25.981	0.000	6.67e+05	7.76e+05

In [138]:

```
forecast_object_deaths = graph_exog_forecast(dataframe=dataframe,
                                             target_column='death',
                                             exogenous_column='hospitalizedCurr
                                             exog_forecast=exog_forecast,
                                             df_ref=df_ref,
                                             alpha=.05, days_to_forecast=45,
                                             train_days=270, m_periods=7)
```



Contents

In [139]:

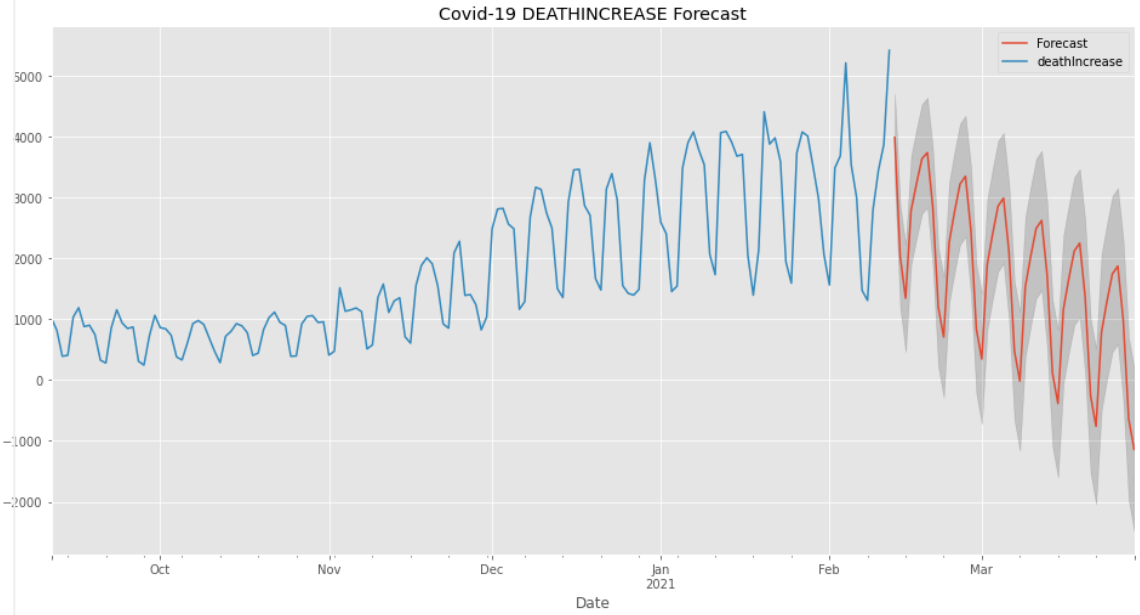
```
forecast_object_deaths.conf_int(alpha=.05)[-5:]
```

Out[139]:

	lower death	upper death
2021-03-25	542285.791255	845053.763793
2021-03-26	542201.557896	855979.672209
2021-03-27	542053.546031	866972.095110
2021-03-28	541839.945093	878027.715806
2021-03-29	541571.173458	889155.497596

In [140]:

```
forecast_object = graph_exog_forecast(dataframe=dataframe,
                                      target_column='deathIncrease',
                                      exogenous_column='hospitalizedCurr
                                      exog_forecast=exog_forecast,
                                      df_ref=df_ref,
                                      alpha=.025, days_to_forecast=45,
                                      train_days=270, m_periods=7)
```



Contents

2.1.12 United States Analysis and Recommendations

The holidays were literal killers. There is no getting around that fact, and without accelerated vaccinations we will continue to have increased deaths around future holidays. We have an opportunity to stop the spread of this virus with everyone's cooperation and resolve. My forecast for the country predicts there's a chance to truly slow and potentially stop the pandemic and Covid deaths by mid-March. There is also the possibility we continue to be irresponsible as a whole and deaths accelerate into the end of March.

I recommend that we come together as Americans in this one endeavor - to protect one another in what is a time of unmitigated risk.

**** WEAR A MASK***

**** STAY AWAY FROM PEOPLE YOU DONT LIVE WITH and respect their space. If you are someone who believes this is a hoax and you see someone with a mask on, just stay away from them.***

**** Use the drive-thru whenever possible, have food delivered. Let's do our best to keep the food service industry going without sacrificing common sense.***

**** Have your groceries delivered - grocery stores continue to stay extremely busy but there are services that are helping individuals pay their bills that can help limit the spread of this disease.***

**** Limit your time indoors with friends and family not in your household and wear a mask when taking that time with them.***

3 Summary Recommendations

3.1 State by State Recommendations

3.1.1 New York

Recommendations for the state include the following:

**** Lower the number of people allowed at indoor private gatherings from the current amount of 10. Social gatherings are not economically essential. Certain states have limits of one or two households per private residence which has proven to limit the spread of the disease.***

**** Increase effort to improve awareness. Covering the mouth but not the nose does not limit the spread of Covid-19. Mandate signage depicting proper mask usage at public establishments.***

**** Continue social distancing policies and reduce the number of outdoor events. The pandemic in New York is not under control.***

3.1.2 California

Recommendations for the state include the following:

**** Require wearing a mask if an individual is not in or on their private property. Allow no exceptions.***

**** Prohibit private and public gatherings of 5 or more people unless from the same household.***

**** The spread of this disease in this state will continue to take lives if people are not made to understand the consequences of selfish behavior. Introduce visual evidence of the rammifications of the virus with an emphasis on personal stories on public social media and television.***

3.1.3 Texas

Recommendations for the state include the following:

**** Limit private social gatherings.***

**** There are those in Texas who believe that the virus is a joke, that the rules don't apply to them, and that there are no consequences. Like California, an awareness campaign with personal stories and visual evidence of what Covid does could help contain the virus.***

**** Make wearing a mask outside of one's private property required.***

3.1.4 Florida

Recommendations for the state include the following:

**** Florida is fully opened - implement travel restrictions and reduce the number of people allowed to privately gather.***

**** Pre and post-Super Bowl footage showed business as usual with zero mask usage. Implement and enforce laws requiring masks in public. There are plenty of states that are responsibly open that are mitigating the spread of this virus. Florida seems to be encouraging the spread.***

**** Common sense is easy. Don't breathe into peoples' faces, wear a mask, and keep your distance.***

3.1.5 United States

I recommend that we come together as Americans in this one endeavor - to protect one another in what is a time of unmitigated risk.

**** WEAR A MASK***

**** STAY AWAY FROM PEOPLE YOU DONT LIVE WITH and respect their space. If you are someone who believes this is a hoax and you see someone with a mask on, just stay away from them.***

**** Use the drive-thru whenever possible, have food delivered. Let's do our best to keep the food service industry going without sacrificing common sense.***

**** Have your groceries delivered - grocery stores continue to stay extremely busy but there are services that are helping individuals pay their bills that can help limit the spread of this disease.***

**** Limit your time indoors with friends and family not in your household and wear a mask when taking that time with them.***

3.2 Conclusions

I chose to undertake this project for several reasons. It is relevant to what is happening now, and it has real implications peoples' lives. On a much more personal level, it is frustrating to have three grandparents in their 90's all of whom I am unable to see during this time. Additionally, mother has an auto-immune deficiency, which makes her risk around others who are irresponsible that much more real for me.

Contents

Most people seem to be respectful enough to wear a mask, but after nearly a year many are growing tired of the simple task. One hopes that respect for others would prevail over the desire to have 'freedom' to do as one pleases. The covenant that we enter into as citizens is with each other. It's to protect each other, be considerate, and be understanding that one person's wants (the desire to not wear a mask, to party, to have a good time) do not supercede the responsibility to protect our fellow Americans.

Furthermore, the argument for personal freedom is in this case ridiculous - it's akin to arguing that one should have the personal freedom to walk around shooting anyone you please simply because you deserve that 'freedom'. With vaccine distribution occurring, we don't have much longer to endure the difficulties.

3.3 Future Work

*** Continue to update the analysis until the project ceases functioning on March 7th, 2021.**

*** Find a future source of data to actively pull in and compare future actual data with the forecasts in this notebook.**

4 Appendix and Ancillary Code [...]