# United States COVID-19 Hospital Resource Forecasting

By Jerad Williams

## Problem

During times of crisis, hospitals, policy makers, and the public need crucial information about how expected need of resources align with existing resources, so that cities and states can best prepare. We will produce forecasts which show hospital bed use, need for intensive care beds, and invasive ventilator use due to COVID-19, based on projected deaths for all 50 U.S. states. These projections are produced by models and data from IHME (Institute for Health Metrics and Evaluation) based on observed death rates from COVID-19, and include uncertainty intervals.

## Hypothesis

NULL Hypothesis ($H_0$): Input time series is stationary
Alternate Hypothesis ($H_1$): Input time series is non-stationary
The confidence level will be at 95%.

After this is verified by different statistical tests, we will check the forecasts by comparing them with the data given by IHME and see how closely they correlate.

## Proposed Approach

IHME's database has three columns for each dimension. One representing the mean of the dimension in different locations distributed by date, second having lower uncertainty bound, and third having upper uncertainty bound corresponding to each dimension.
In this database, we are targeting three dimensions:

- All beds - Number of beds needed by day
  - Mean
  - Lower uncertainty bound
  - Upper uncertainty bound
- ICU beds - Number of intensive care unit beds needed by day
  - Mean
  - Lower uncertainty bound
  - Upper uncertainty bound
- Invasive Ventilators - Number of invasive ventilators needed by day
  - Mean
  - Lower uncertainty bound
  - Upper uncertainty bound

All these columns are distributed by location and date. Clearly, each of these columns form a time-series with time being represented by date column and value being represented by each of the above mentioned columns.

The proposed idea is to use time-series forecasting using ARIMA and SARIMAX in order to see how many COVID beds, ICU beds, and invasive ventilators will be needed in the coming year and how far the predicted upper and lower forecast values go from the given upper and lower uncertainty bounds in the dataset.

Also, I will be predicting the mean values of each of these dimensions for the upcoming three months, which will then help us and authorities prepare for additional amenities that should be available in times of need.

### Evaluation (Testing for Accuracy)

For accuracy testing, we will implement the following:

- Comparing the observed data points and forecasts, as well as calculating the Mean Squared Error and Root Mean Squared Error. The less they are, the better the model.
- Running model diagnosis and observing the Q-Q plots and observing how much closer the points are to the theoretical quantiles, histogram of forecasts and residuals and seeing if they're having normal distribution.
- Checking the Akaike Information Criterion (AIC) value. After running grid search, the parameters which have lowest AIC value corresponding to them will be used for model building.
- After executing the above, I can then obtain quarterly, biannual, or annual forecasts and compare the already acquired data from the source for upcoming months and compare the same with the forecasts derived from the model.

# Data Wrangling

### What kind of cleaning steps were performed?

- First, checking that the data was distributed on the basis of states inside the US. I wanted to see the figures of the US as a country, so I removed this column and grouped the data by date.
    - For mean, I aggregated the mean columns by grouping them by date and taking the mean of the mean columns.
    - For the upper uncertainty bound, I grouped by date and aggregated it as sum.
    - For the lower uncertainty bound, I grouped by date and aggregated the column as sum.
- The next step is to look for any missing values now.

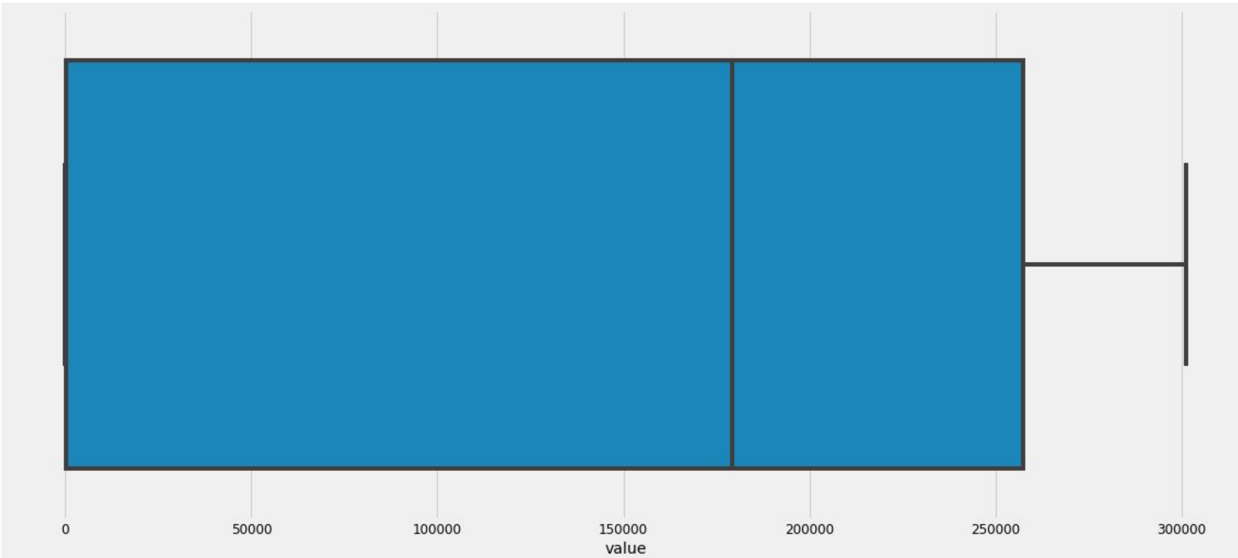
    - Firstly, I checked the distribution of each of these columns and saw what shape each of them were taking.
    - If the distribution was normal, I did missing value imputation by mean. That is, I replaced the missing values with the mean of the other values of the same column.
    - If the distribution was skewed, I replaced them by median.
- Then, I looked for any outliers.
    - I did box-plot analysis of each of the columns and saw if the columns were having values which were lying outside of the whiskers.
    - If yes, I took out the quantiles corresponding to each of these columns and replaced the outliers with the most optimum quantile values - generally, it's 96 percentile value in case of outliers above upper whisker and 4 percentile value in case of those below lower whisker.
    - I also used z-scores to handle them. I replaced all values having z-scores above +3 or below -3 with +3 and -3 respectively.

**How did you deal with missing values and outliers, if any?**

- Firstly, I checked the distribution of each of these columns and saw what shape each of them were taking.
- If the distribution was normal, I did missing value imputation by mean. That is, I replaced the missing values with the mean of the other values of the same column.
- If the distribution was skewed, I replaced them by median.
- Also, I used Imputation Using Multivariate Imputation by Chained Equation (MICE) and then saw which of the three methods were giving me better results and chose that method.

**Were there outliers, and how did you handle them?**

- I did the box-plot analysis of each of the columns and saw if the columns were having values which were lying outside of the whiskers.
- If yes, I took out the quantiles corresponding to each of these columns and replaced the outliers with the most optimum quantile values - generally, it's 96 percentile value in case of outliers above upper whisker and 4 percentile value in case of those below lower whisker.




- I also used z-scores to handle them. I replaced all values having z-scores above +3 or below -3 with +3 and -3 respectively.
  - After executing z-score, every value has z-score less than +3 or greater than -3. Thus, no outliers are present in the data.

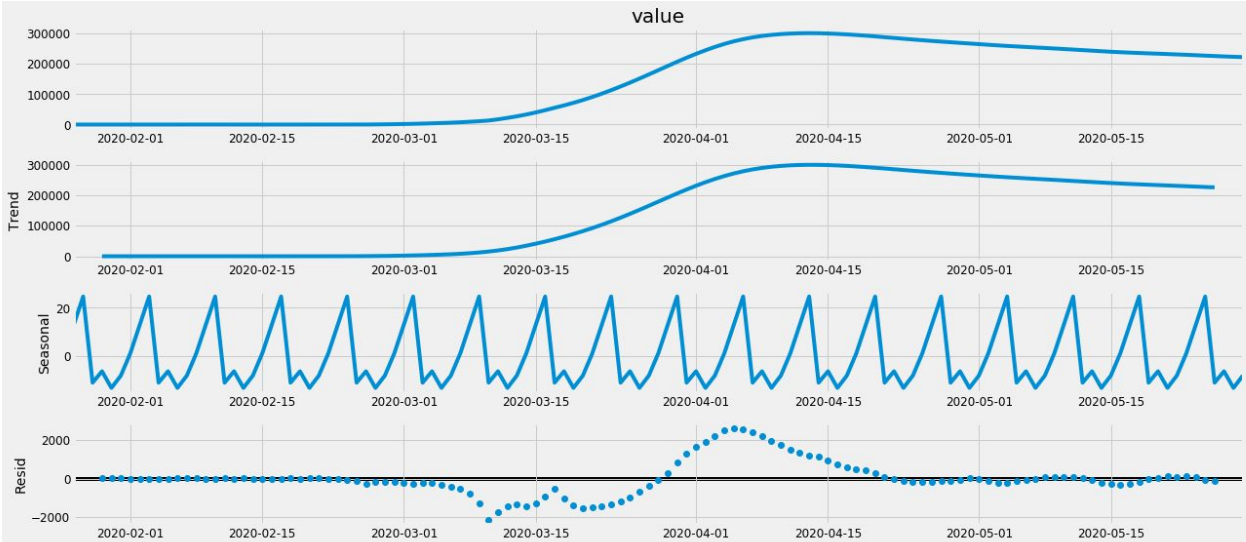
# Statistical Data Analysis

### Extracting Time-Series from Dataframe

We first need to extract the dataframe of all three dimensions (All Beds, ICU Beds, Invasive Ventilators) and convert them into a time-series. When extracted, plotting to begin visualization of said data showed that the time-series stays close to 0 need of resources till the end of February 2020, and begins increasing after that. It keeps on increasing till mid April where it reaches the highest point and then begins decreasing.

### Decomposing the Time-Series

Now that we've visualized the data, we need to be able to identify and filter through its characteristics. We will use the decomposition model, which will narrow the time-series to the following four components: Observed, Trend, Seasonal, and Residual.

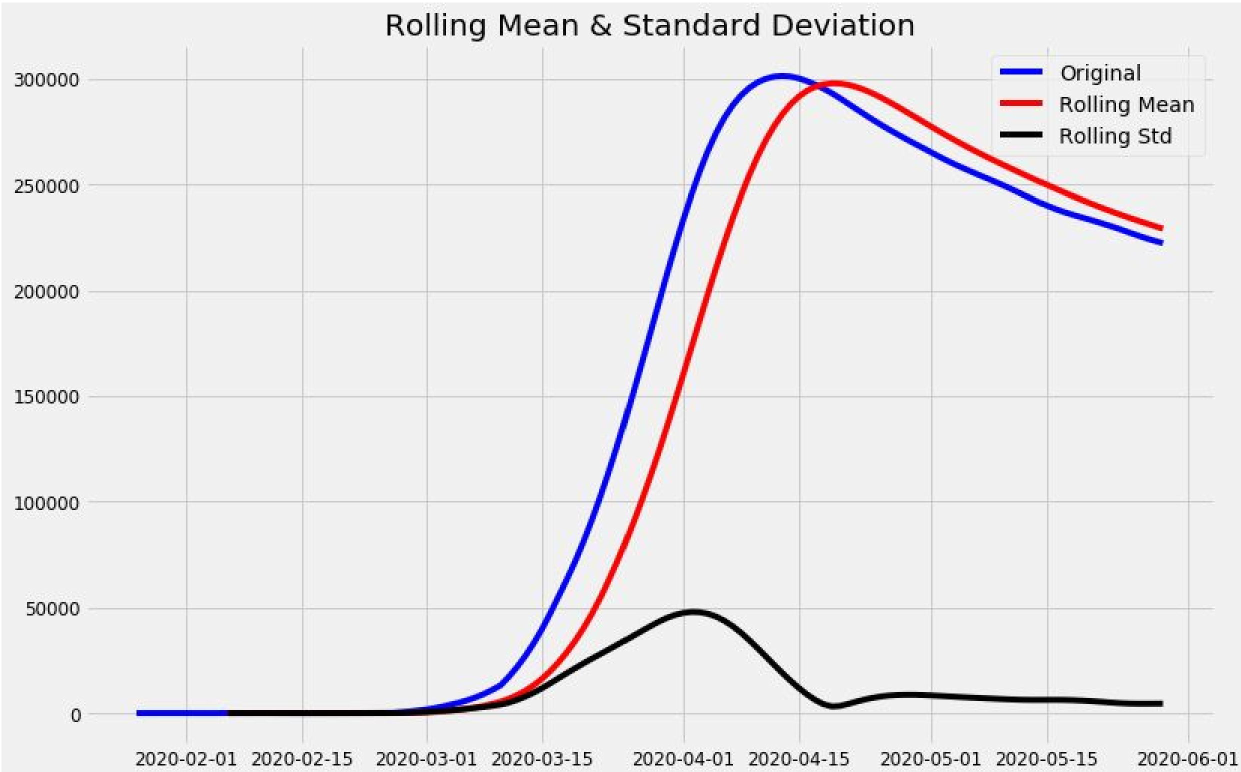



By implementing the decomposition model, we can see that besides trend, there is a high amount of seasonality within the data.

**Checking Stationarity**

In a time series, we know that observations are time dependent. It turns out that a lot of nice results that hold for independent random variables hold for stationary random variables. So by making the data stationary, we can actually apply regression techniques to this time dependent variable.

- The next step is to check if the series was stationary or not. A series is said to be stationary if over time, it satisfies following three conditions:
  - Constant mean
  - Constant variance
  - An autocovariance that doesn't depend upon time.
- To check for stationarity, I used two procedures:
  - Plotting rolling statistics
  - Dickey-fuller test
- For plotting rolling averages, I used both simple moving averages and exponential weighted moving averages and saw which of them gave me constant mean and variance and better Dickey-Fuller statistics figure and subtracted the resultant series from the original series to obtain stationary series.
- I also checked for trend and seasonality components and if they're present, I removed them from the original series to get a stationary series since both of them make a series non-stationary.

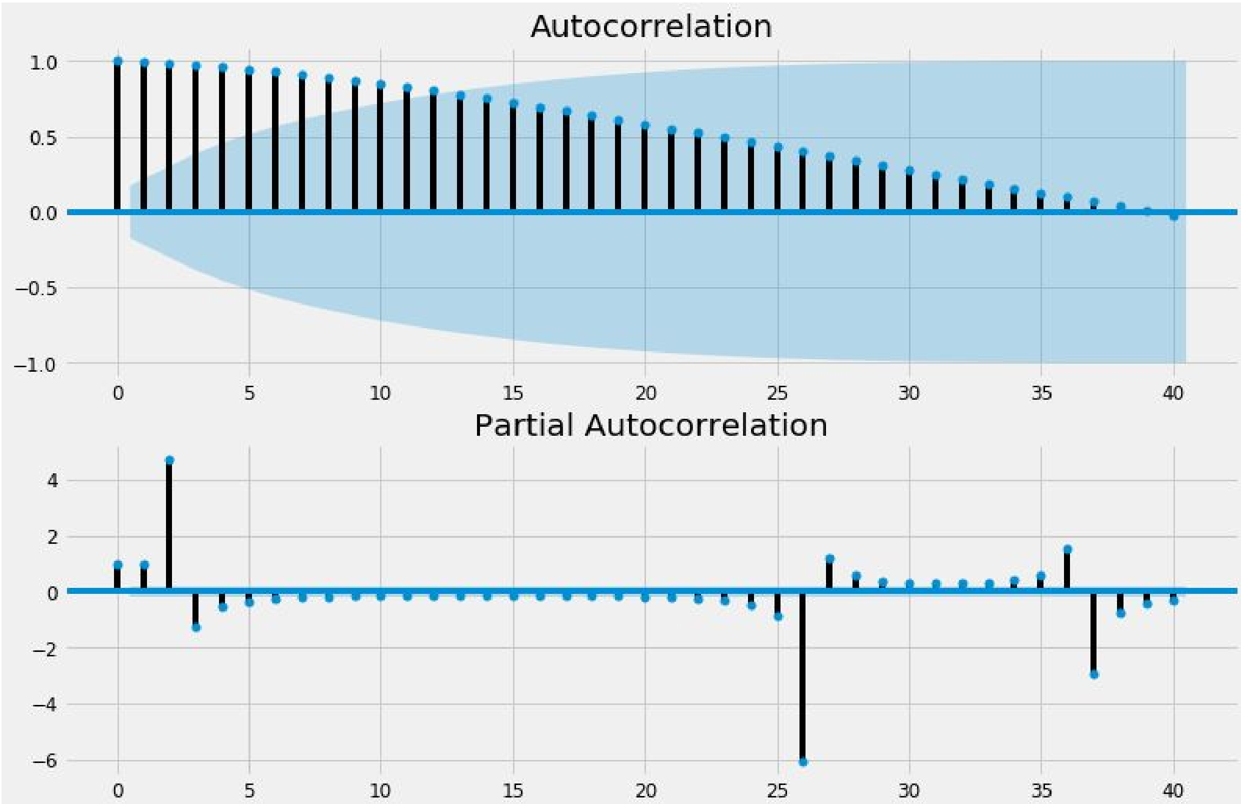



After determining/plotting the rolling statistic, plotting the standard deviation, and performing the Dickey-Fuller test, we can see that this time-series is non-stationary with p value of 0.58.

ARIMA converts a non-stationary series to stationary by taking the difference between the values, so no transformations are needed.

**Plotting ACF and PACF**

ACF and PACF functions to find the autocorrelation and partial autocorrelation plots to obtain q(ACF lag) and p(PACF lag) values by looking at the lags where each of these charts cross the upper interval for the first time.




Since seasonality is also present in the data, as can be seen from above graphs, after 7 days, we're seeing some sine wave like pattern, let's consider SARIMA, but will also consider ARIMA.

# Forecasting Time-Series with ARIMA

ARIMA consist of 2 terms (i) AR (ii) MA

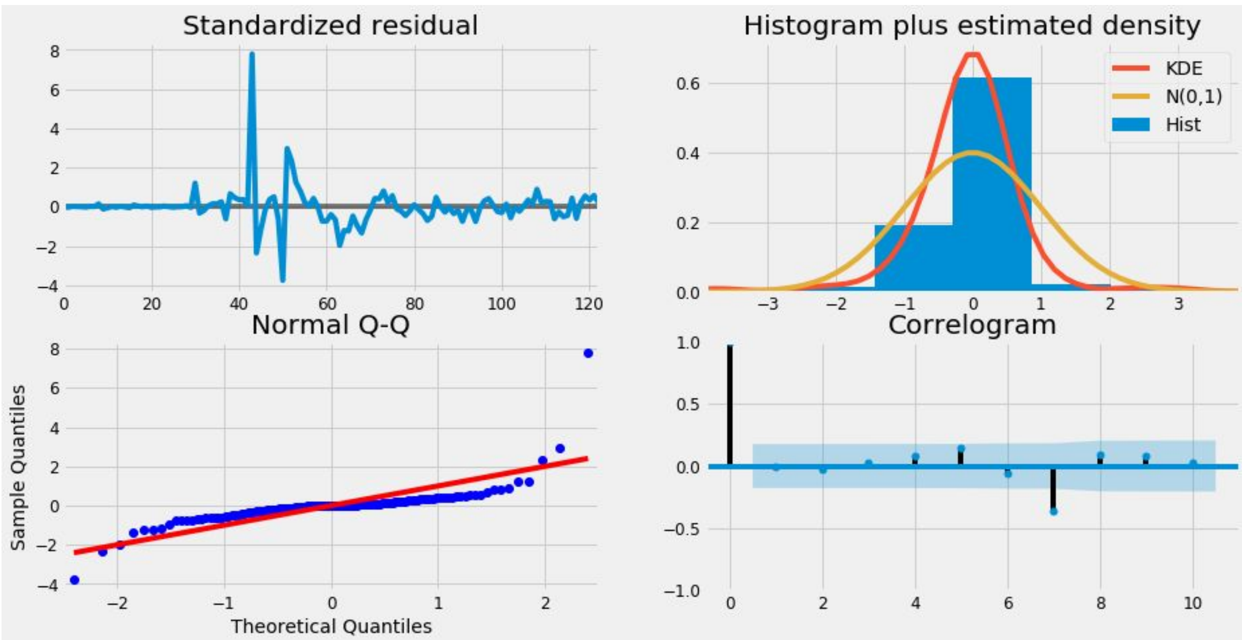
- AR corresponds to the difference value. This is today's value minus yesterday's value or value-on-value change.
- MA corresponds to moving average terms.

Three integers (p, d, q) are typically used to parametrize ARIMA models.

- p: number of autoregressive terms (AR order)
- d: number of nonseasonal differences (differencing order)
- q: number of moving-average terms (MA order)

When running a parameter search for the best p,q and d and AIC (Akaike Information Criteria), we found that (2, 2, 2) - AIC:1684.3964 corresponds to the lowest AIC value of 1684.3964. So we trained our model on the value of p = 2, q = 2 and d = 2.

We then plotted the properties of the model:

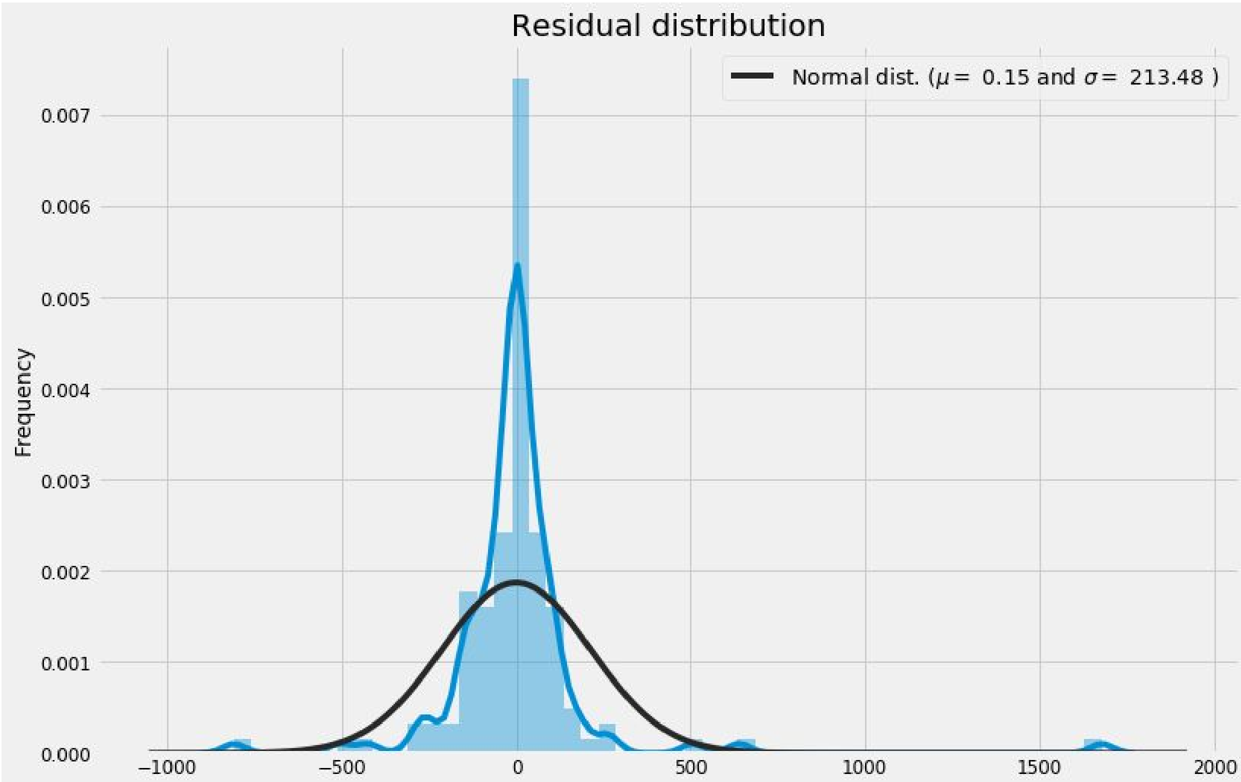



The above plots imply that:

- The Standardized Residual follows random behaviour and thus corresponds to random noise.
- The Histogram suggests that the residuals have normal distribution.
- The Normal Q-Q plot suggests that the theoretical and sample quantiles are very close to each other. The more close the sample quantiles to the line, the more normal their distribution will be. The points approximately lie on the red line.
- The Correlogram has all autocorrelation of different lags in between the blue shaded area.

The diagnosis says that the model is quite good.

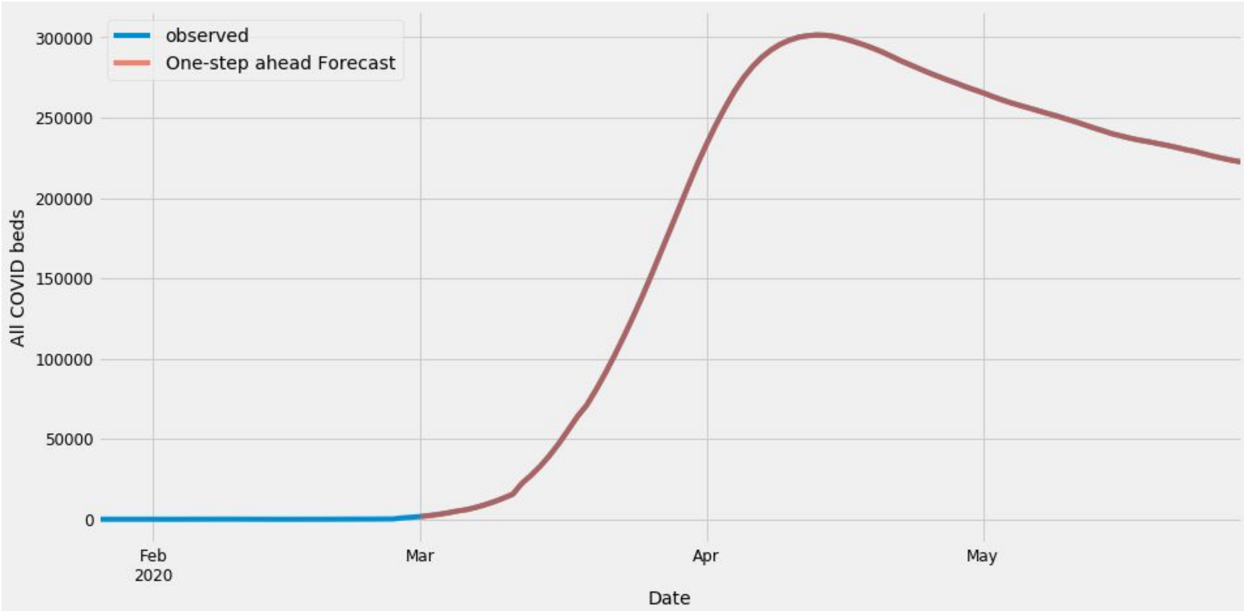
**Residual Distribution**




We can see that the residuals are near normally distributed with kurtosis being a little high.

**Predicting Values**

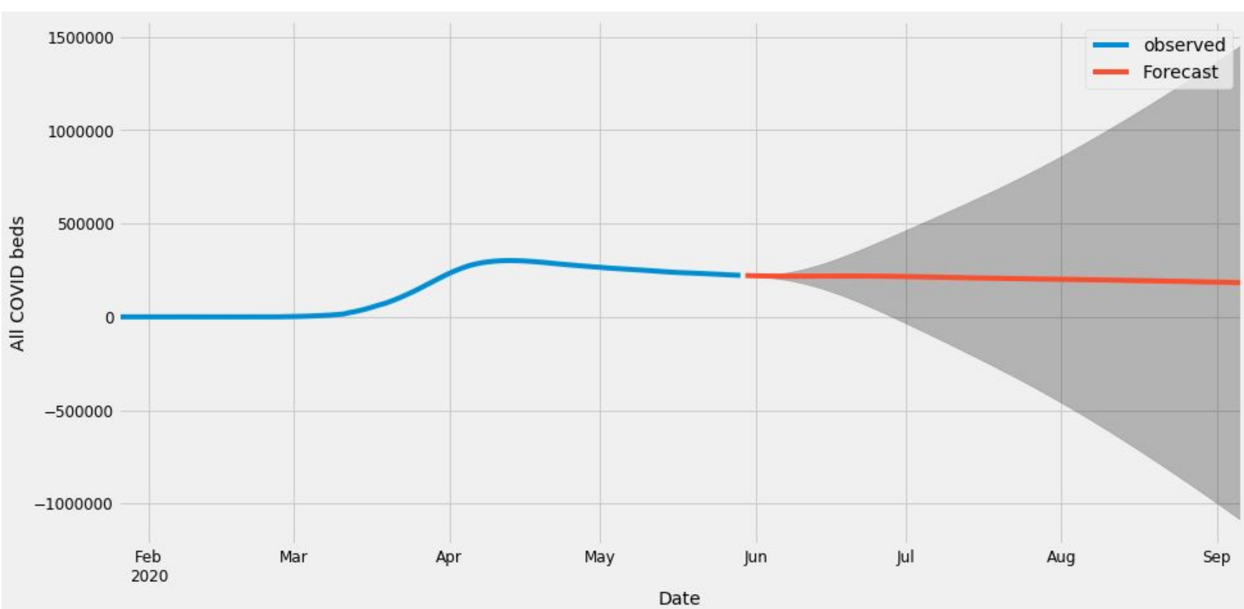
Further on, we wanted to validate the model by plotting the predicted values on the data after the 1st of March 2020.




The line plot is showing the observed values compared to the rolling forecast predictions. Overall, our forecasts align with the true values very well.

**We also calculated the Mean-Squared Error which was close to 0 which again implies that the model is quite good.**

Finally, we predicted the requirements of the number of COVID-19 beds, ICU beds, and ventilators after 29th May 2020 with 95% confidence interval.



# Forecasting Time-Series with SARIMA

An extension to ARIMA that supports the direct modeling of the seasonal component of the series is called SARIMA (Seasonal Autoregressive Integrated Moving Average).

SARIMA adds three new hyperparameters to specify the autoregression (AR), differencing (I) and moving average (MA) for the seasonal component of the series, as well as an additional parameter for the period of the seasonality.
Configuring a SARIMA model requires selecting hyperparameters for both the trend and seasonal elements of the series.

**Trend Elements**

They are the same as the ARIMA model. Specifically:

- p: Trend autoregression order. (AR)
- d: Trend difference order.
- q: Trend moving average order. (MA)

Seasonal Elements

There are four seasonal elements that are not part of ARIMA that must be configured; they are:

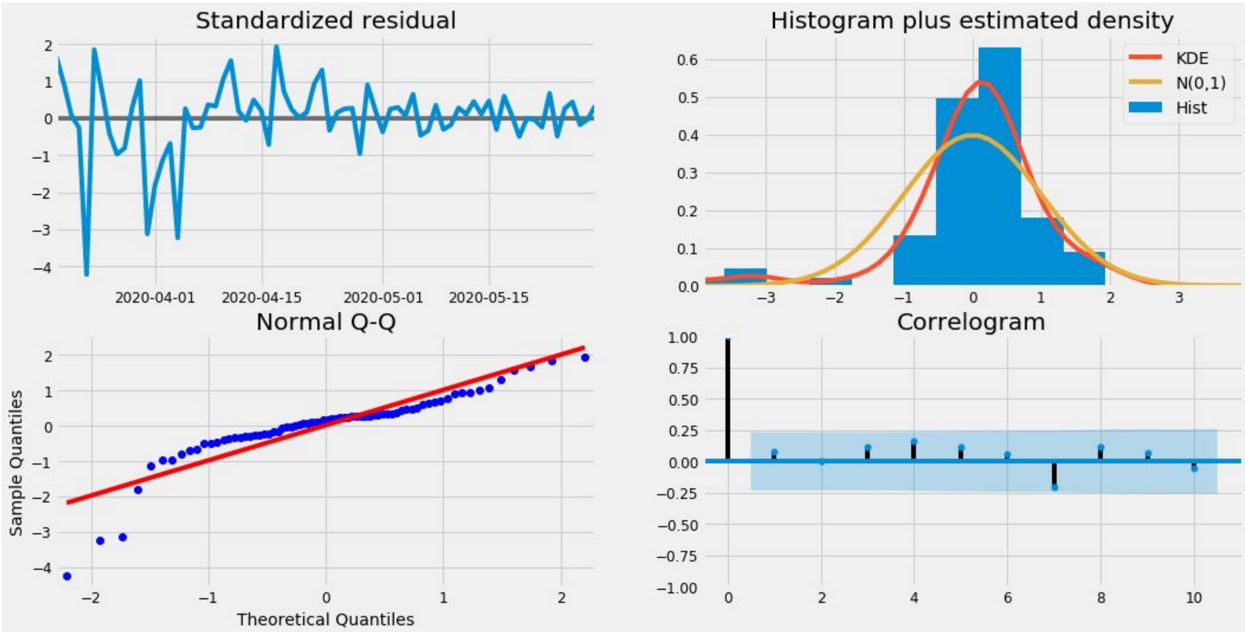
- P: Seasonal autoregressive order.
- D: Seasonal difference order.
- Q: Seasonal moving average order.
- m: The number of time steps for a single seasonal period.

We decided to do the parameter search for (p,q, d) and (P, D, Q, m), and found the following best model:

**SARIMA (1, 2, 2)x(0, 2, 2, 12)12 - AIC:1045.545**

We can see that (1, 2, 2)x(0, 2, 2, 12)12 - AIC:1045.545 corresponds to the lowest AIC value of 1045.544.

We then plotted the properties of the model:



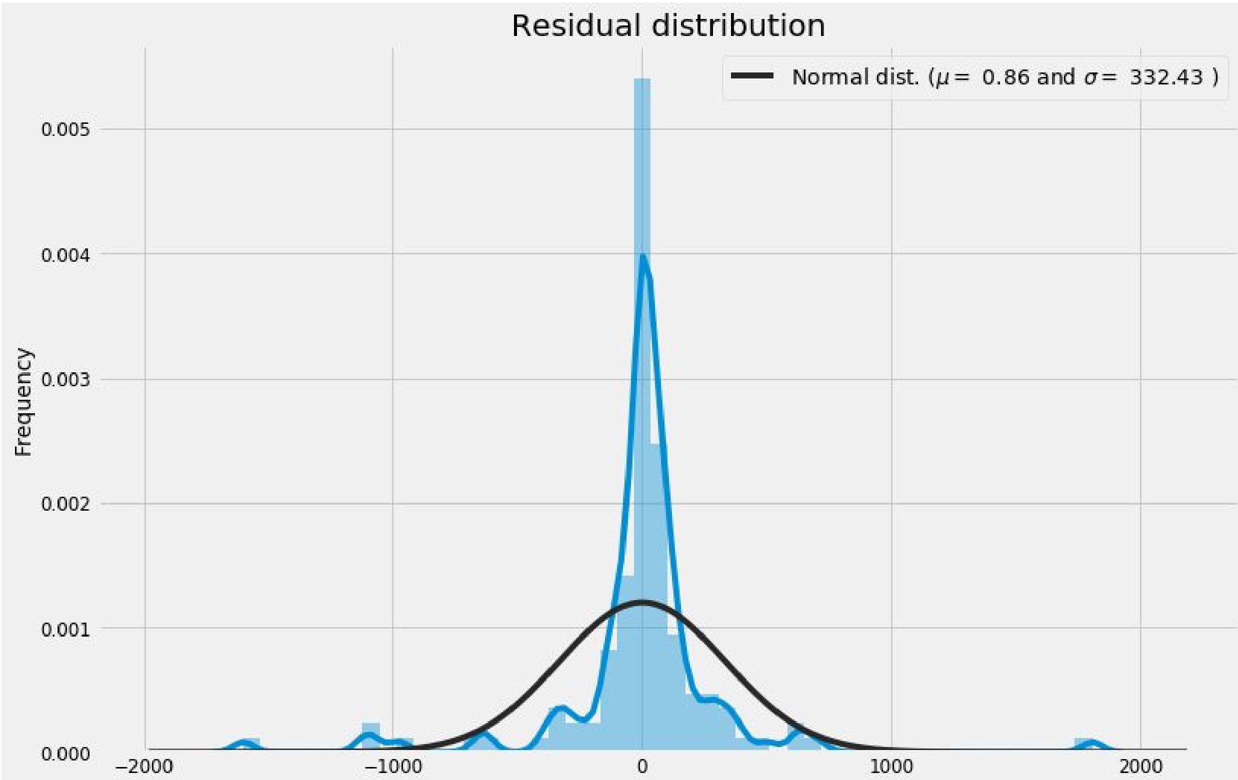

The above plots implies that:

- The standardized residual follow random behaviour and thus corresponding to random noise.
- The histogram suggests that the residuals have normal distribution.
- The Normal Q-Q plot suggests that the theoretical and sample quantiles are very close to each other. The more close the sample quantiles to the line, the more normal their distribution will be. The points approximately lie on the red line.
- The correlogram has all autocorrelation of different lags in between the blue shaded area.

Hence, the diagnosis implies that model is still quite good.

We wanted to validate the model by plotting the predicted values on the data after the 1st of March 2020

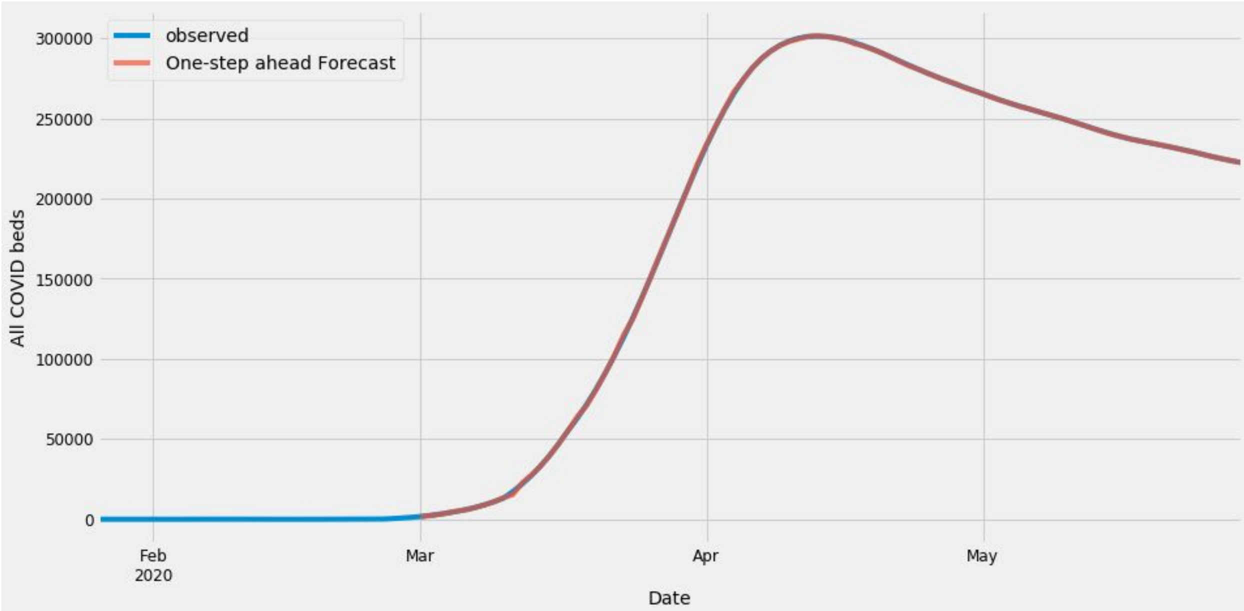
**Residual Distribution**




We can see that the residuals are near normally distributed with kurtosis being a little high.

**Predicting Values**

Further on, we wanted to validate the model by plotting the predicted values on the data after the 1st of March 2020.

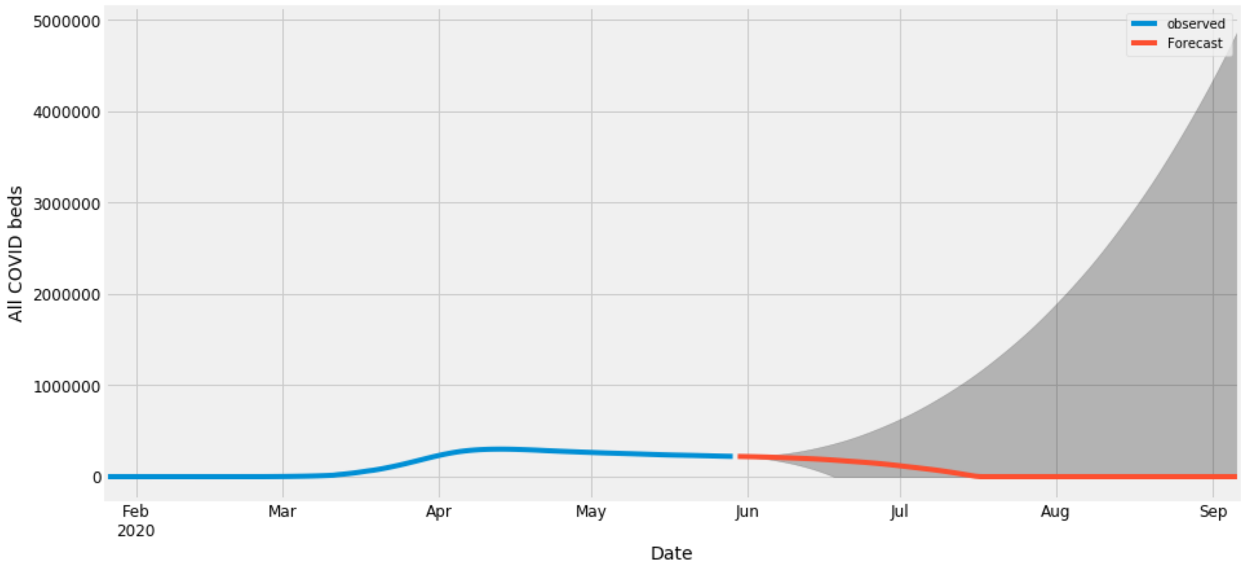



The line plot is showing the observed values compared to the rolling forecast predictions. Overall, our forecasts align with the true values very well.

We found that the predicted value is very close to the one-step ahead forecast.

**However, we calculated the Mean-Squared Error 916.37 which is bigger than the ARIMA model, which implies that the ARIMA model is better than SARIMA.**

Finally we predicted the requirements of the number of the bed after 29th May 2020 with 95% confidence interval.



5000000
4000000
3000000
2000000
1000000
0
All COVID beds
observed
Forecast
Feb
2020
Mar
Apr
May
Jun
Jul
Aug
Sep
Date

## Acknowledgements


1. IHME: COVID-19 Projections. (2020). Retrieved May 3, 2020, from https://covid19.healthdata.org/united-states-of-america
2. Hyndman, R. J., & Athanasopoulos, G. (2018, May 6). Forecasting: Principles and Practice. Retrieved May 10, 2020, from https://otexts.com/fpp2/arima.html
3. Perktold, J., Skipper, S., & Taylor, J. (2019). SARIMAX: Introduction¶. Retrieved May 10, 2020, from https://www.statsmodels.org/dev/examples/notebooks/generated/statespace_sarimax_stata.html