Analysis of the COVID-19 Outbreak in New York City, New York

Caroline Denecke, Sally Dufek, Tyler Sagendorf, Mason Skaruppa

May 9, 2020

Abstract

Objective: To forecast the daily number of new COVID-19 cases, hospitalizations, and deaths in New York City, New York, from April 21, 2020 to May 5, 2020 based on models fit to prior data spanning from March 3, 2020 to April 20, 2020.

Methods: Autoregressive Integrated Moving Average (ARIMA) models were fit to the time series data from the New York City Department of Health and Mental Hygiene's coronavirus-data GitHub repository. These models were then used to forecast fifteen days in the future and the predictions were compared to the actual reported numbers for those days.

Results: The results suggested that the daily number of deaths and hospitalizations in NYC during the forecasted time period would both decrease, and the daily number of cases would repeatedly decrease and then increase with no change in the mean. When we compared these forecasts to the real data, we found that only the number of deaths closely matched our predictions. The forecasted daily number of new cases was always greater than the true daily number of new cases, and the forecast did not exhibit the decreasing trend observed in the real data; however, the predictions did follow the same pattern of decreasing and increasing over time. Also, the true values were almost always within the 80% prediction intervals, and were always within the 95% prediction intervals. Similarly, the forecasted daily number of hospitalizations was always less than the true daily number of hospitalizations, though the true values were always within the 80% prediction intervals.

Conclusions: When it comes to a pandemic like COVID-19, there are many factors that we cannot account for in our models. Despite these limitations, we were able to predict the daily number of COVID-19 cases, hospitalization, and deaths in New York City, New York, with a high or moderate degree of accuracy.

Introduction

Background

COVID-19 is a type of coronavirus that may cause symptoms such as coughing, respiratory difficulties, fever, and chills. It is especially deadly to those who may be more vulnerable due to age or preexisting conditions: older adults, those with asthma, and those who are immunocompromised (CDC 2020c, 2020a). The virus originated in Wuhan, Hubei Province, China at the end of 2019, and is thought to have initially spread through "animal-to-person" contact, since "many of the patients at the epicenter of the outbreak... had some link to a large seafood and live animal market" (CDC 2020b). Since then, it has rapidly spread from person-to-person to all parts of the world, and on March 11, 2020, the World Health Organization (WHO) officially classified the outbreak as a pandemic ("WHO Director-General's Opening Remarks at the Media Briefing on COVID-19 - 11 March 2020" n.d.). Despite global efforts to stop the spread of the disease, the number of cases continues to grow daily. As of May 6, 2020, 17:08 GMT, there were over 1.24 million confirmed cases of COVID-19 in the United States. This accounts for nearly one-third of all cases worldwide ("United States Coronavirus: 1,245,857 Cases and 73,145 Deaths - Worldometer" n.d.).

We chose to review the data associated with the COVID-19 outbreak in New York City (NYC), New York, because it was one of the first US cities to be affected, so it has some of the most complete and accurate data. Also, NYC has the largest population and highest population density of any US city at 8.2 million people and about 27,013 people per square mile, respectively ("U.S. Census Bureau QuickFacts: New York City, New York" n.d.). These two characteristics provide an ideal environment for COVID-19 to spread on a large scale, since the virus can be transmitted through person-to-person contact.

Data Sources and Collection

We used the "case-hosp-death.csv" and "probable-confirmed-dod.csv" files provided in the coronavirus-data GitHub repository by the New York City Department of Health and Mental Hygiene. The files that we used for this report were pulled on May 7, 2020, and only data from March 3 to May 5 was used. This is because the data is updated daily, and is "preliminary and subject to change," and it is mentioned that "the most recent data may be incomplete" ("README.md").

The "case-hosp-death.csv" file "contains daily counts of new confirmed cases, hospitalizations, and deaths" where "cases are by date of diagnosis," "hospitalizations are by date of admission," and "deaths are by date of death." The "probable_confirmed-dod.csv" file contains counts of confirmed and probable deaths, where a confirmed death means "the decedent was a New York City resident who had a positive... [COVID-19] laboratory test," and a probable death means "the death certificate lists as a cause of death 'COVID-19' or an equivalent," but the decedent "had no known positive laboratory test" for COVID-19 ("README.md").

GitHub Repository:

https://github.com/nychealth/coronavirus-data?files=1

Methods

We split the data from the "case-hosp-death.csv" file into training and testing sets. The training set consists of all observations from March 3 to April 20, and the testing set consists of all observations from April 21 to March 5. We then created three time series (TS) objects from each of the two data sets for the daily number of cases, hospitalizations, and deaths. The three TS objects from the training data would be used to fit ARIMA models while the three TS objects from the testing data would be used to compare to the values forecasted by the ARIMA models.

To determine the forms of the ARIMA models for the number of cases, hospitalizations, and deaths in NYC, we used the `auto.arima` function in R. We then assessed these models by using the `Box.test` function to check whether there was evidence of a lack of fit. If the p-value reported by the test was large, that means there was no evidence that the model in question is a poor fit for the data (the residuals are likely uncorrelated). Of the models suggested by the `auto.arima` function, only the ARIMA(0,2,1) model for the daily number of deaths was a good fit. To find the model forms for the number of cases and hospitalizations, we created ARIMA models for all possible combinations of reasonable values for the parameters p, d, and q, and calculated the Ljung-Box test p-value for each of these models. Models with p-values less than 0.05 were immediately discarded, and the remaining models were sorted in descending order by p-value (Tables 1 and 2 in the Appendix). The final models for the number of cases and hospitalizations were the ones that maximized the Ljung-Box test p-value and minimized the sum of p and q (fewest number of terms).

Once the final models were chosen, we predicted values of each time series from April 21 to May 5 (15 days beyond the scope of the training data). We also compared our forecasts to the true values reported for those days to assess the accuracy of our predictions.

Results

Exploratory Data Analysis

We plotted the time series data for new cases, hospitalizations, and deaths (confirmed, probable, and combined) from March 3 to April 20. These plots are shown below.

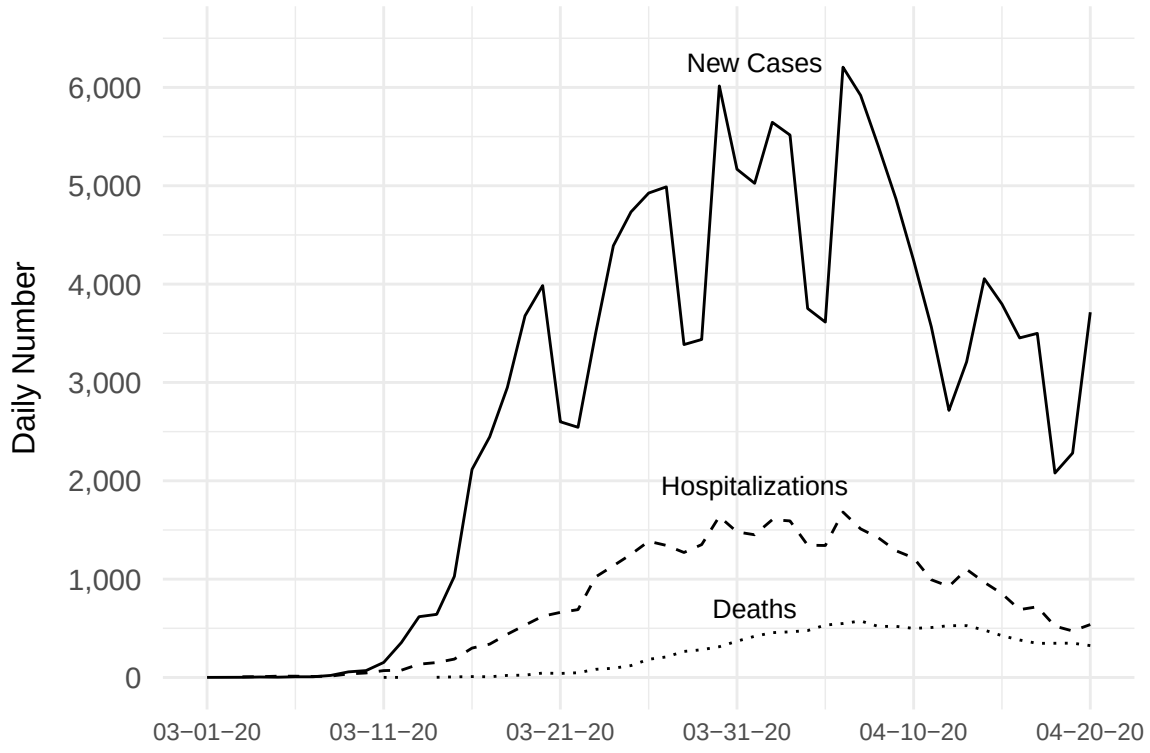

Figure 1: Daily COVID-19 Cases, Hospitalizations, and Deaths in NYC

From Figure 1, we can see that the number of new cases, hospitalizations, and deaths caused by COVID-19 appears to peak around the beginning of April. Generally, the number of deaths tends to be lower than the number of hospitalizations, and both are always lower than the number of new cases each day. None of these time series are stationary.

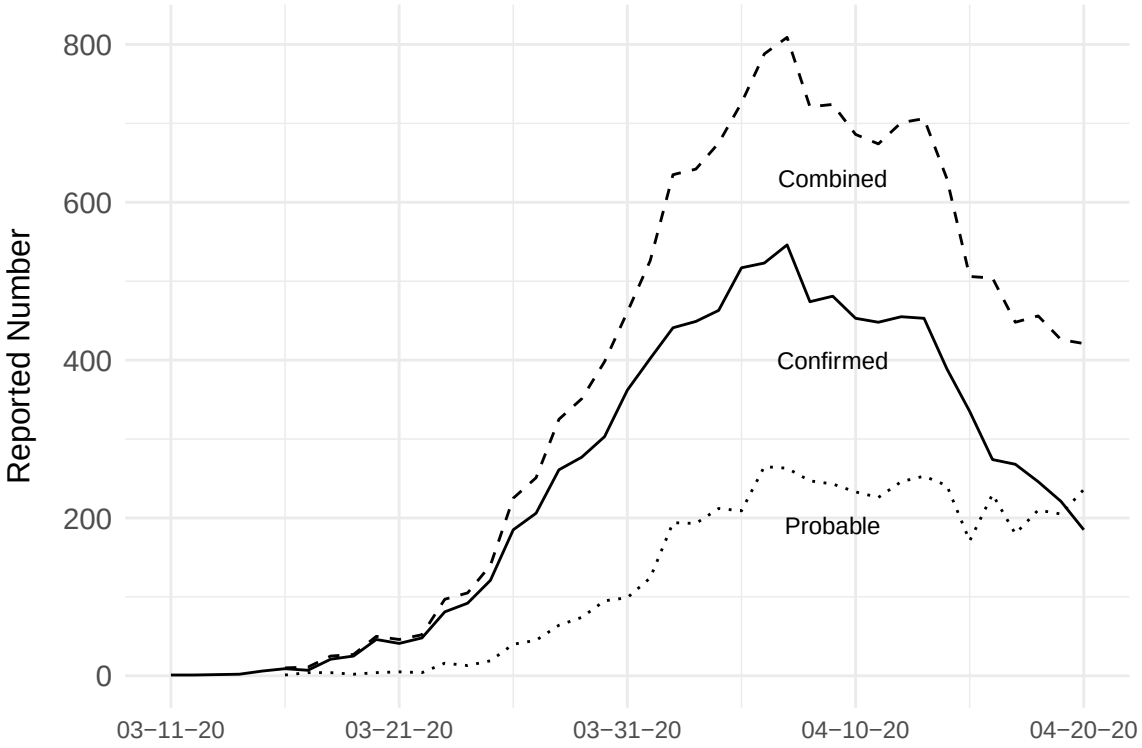

Figure 2: Daily COVID-19 Deaths in NYC by Category

In Figure 2, we can see the number of deaths attributed to COVID-19 broken down by category. The "Probable" time series shows daily death counts where the death certificate listed "COVID-19" as the cause of death, but the decedent had no positive laboratory test for COVID-19. The "Confirmed" time series is the daily number of deaths where the decedent did test positive for COVID-19. The "Combined" time series is the daily number of confirmed and probable deaths. We chose to use the series for the number of confirmed deaths, since we wanted our analysis to be as accurate as possible.

Modeling

Below are plots of the original time series data for the daily number of cases, hospitalizations, and deaths in NYC ranging from March 3 to April 20, along with their fitted ARIMA models. The characteristic equations of each model are provided along with insights on the quality of fit.

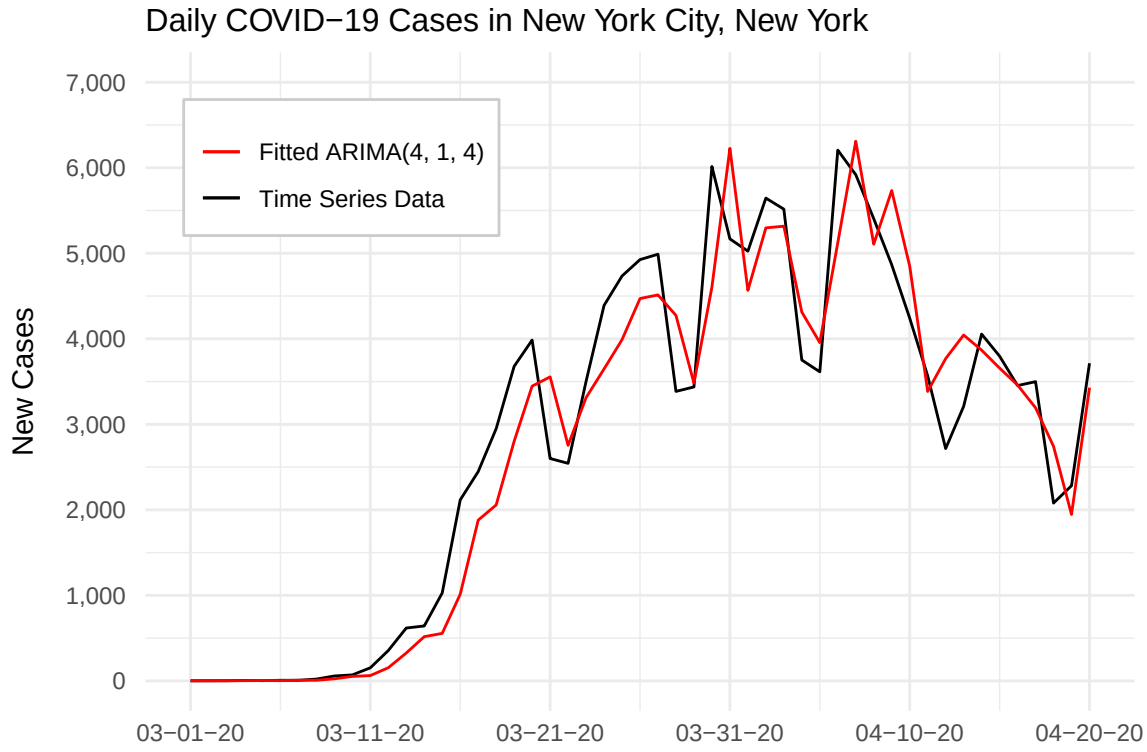

Figure 3: Daily COVID-19 Cases with Fitted Line

Figure 3 shows the original time series data for the number of new cases each day (black) along with the values obtained by fitting an ARIMA(4,1,4) model to the data (red). The characteristic equation for this model is

$$(1-0.8586B+1.4181B^2-0.7978B^3+0.9174B^4)(1-B)Y_t = (1-1.0535B+1.254B^2-0.6946B^3+0.7441B^4)\varepsilon_t$$

where Y_t is the number of cases at time t, and ε_t is white noise. This model was obtained by calling the final ARIMA model for cases. This is included in the Appendix under the "Final Models" subsection.

We can see that the fitted line tends to be below the actual time series data, though we know that there is no evidence that this model is a poor fit, based on the Ljung-Box test output (p-value = 0.6648). See "Ljung-Box Tests" in the Appendix for details.

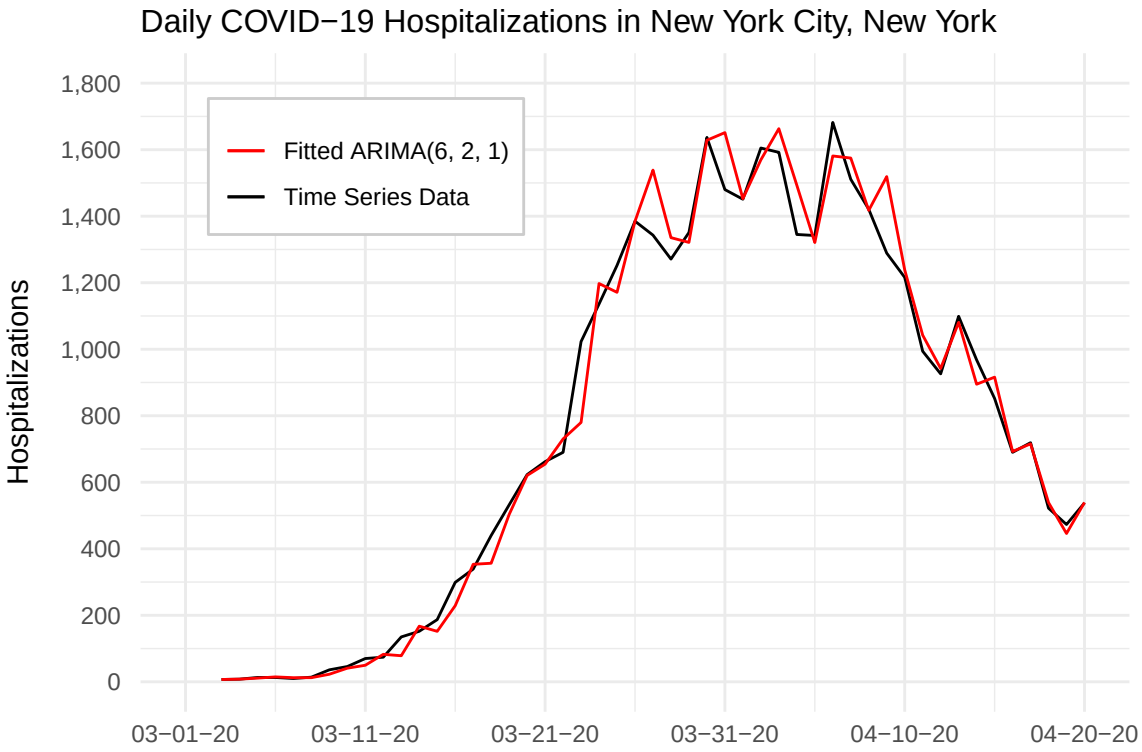

Figure 4: Daily COVID-19 Hospitalizations with Fitted Line

As with the number of cases, Figure 4 shows the original time series data for the number of daily hospitalizations in NYC with the fitted ARIMA(6,2,1) line. The characteristic equation for the fitted model is

$$(1 + 1.2546B + 1.209B^2 + 1.1486B^3 + 0.8146B^4 + 0.6526B^5 + 0.6546B^6)(1 - B)^2 Y_t = (1 + 0.7588B)\varepsilon_t$$

Visually, it appears that this model is a very good fit for the original data, and this is confirmed by the Ljung-Box test output in the Appendix (p-value = 0.3358).

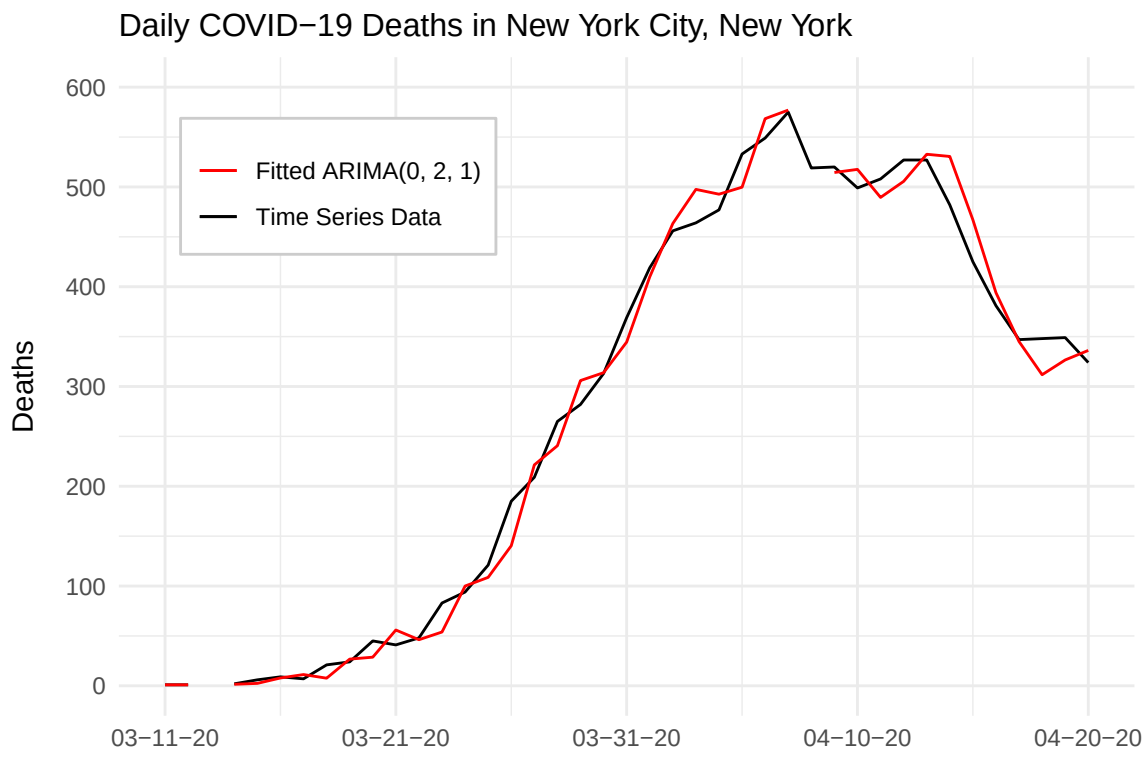

Figure 5: Daily COVID-19 Deaths with Fitted Line

Figure 5 is a plot of the original time series data for the number of daily COVID-19 deaths in NYC with the fitted ARIMA(0, 2, 1) line. The characteristic equation for the fitted model is

$$(1 + 0.6185B)(1 - B)^2 Y_t = \varepsilon_t$$

This was the only final model that was chosen by the `auto.arima` function since its Ljung-Box test p-value was 0.2212, which suggests that there is no evidence that the model residuals are uncorrelated (that they aren't white noise and the model is a poor fit).

This model, unlike the other final models, was produced by the `auto.arima` function. Its Ljung-Box test p-value is fairly large (0.2212), which suggests that there is no evidence that the model residuals are correlated.

Since all three models appear to accurately capture the behaviors of the original time series, we can proceed with forecasting.

Forecasting

We forecasted values 15 days beyond the scope of the original data for each of the three time series (cases, hospitalizations, and deaths). Then, we graphically compared these predictions to the actual data for that time period.

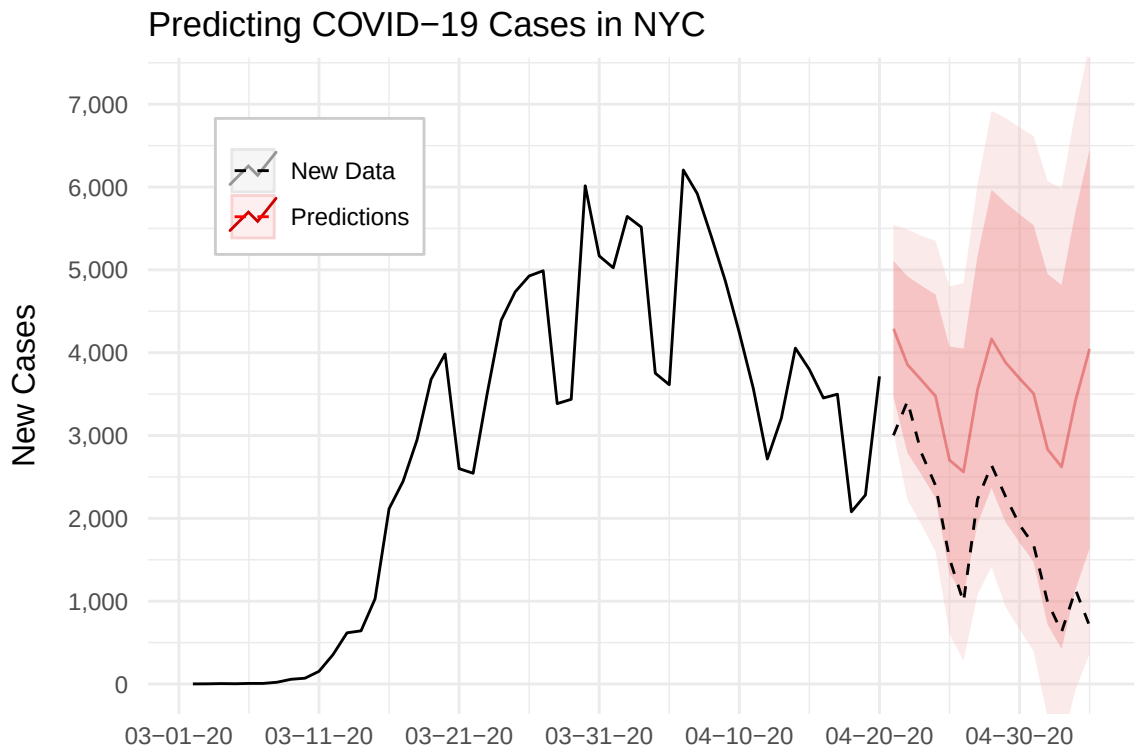

Figure 6: Actual and Forecasted COVID-19 Cases in NYC

From Figure 6, we can see that the model overpredicted the new daily number of cases. Also, the predictions do not seem to follow an overall decreasing trend like that of the actual data. However, the predictions do follow a similar pattern of increase and decrease.

It is likely that the tail end of the training data (solid black line) heavily influenced the predictions, because it appears that from April 12 to April 20, the number of cases remained fairly constant.

The real data was, for the most part, contained within the 80% prediction bands, and was always within the 95% prediction bands. The point forecasts, actual data, and prediction intervals are available in the Appendix under "Forecasts" for each of the three series.

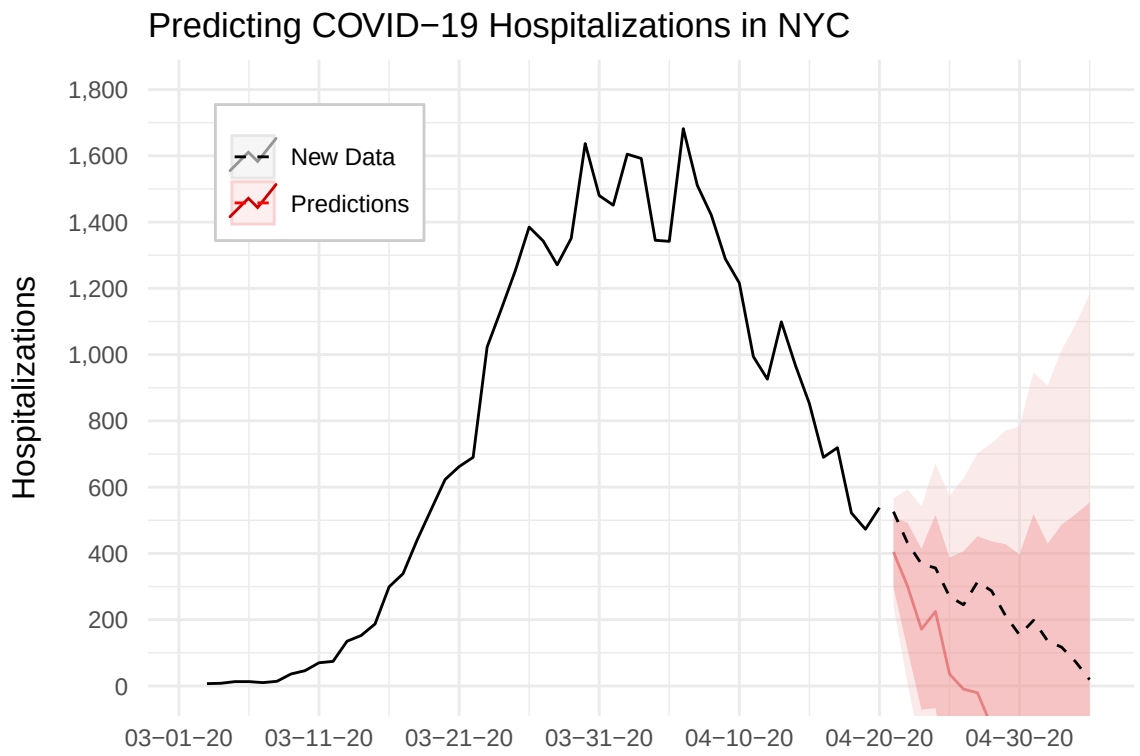

Figure 7: Actual and Forecasted COVID-19 Hospitalizations in NYC

From Figure 7, we can see that the predicted number of hospitalizations was always lower than the actual number of hospitalizations. In fact, the model predicted a negative number of new COVID-19 hospitalizations. In context, we would interpret any negative values as zero, since new hospitalizations cannot be negative. Even though the actual values were higher than the predicted values, the true values were still within the 80% prediction bands for every day except April 21, and the true value that day was still within its respective 95% prediction interval.

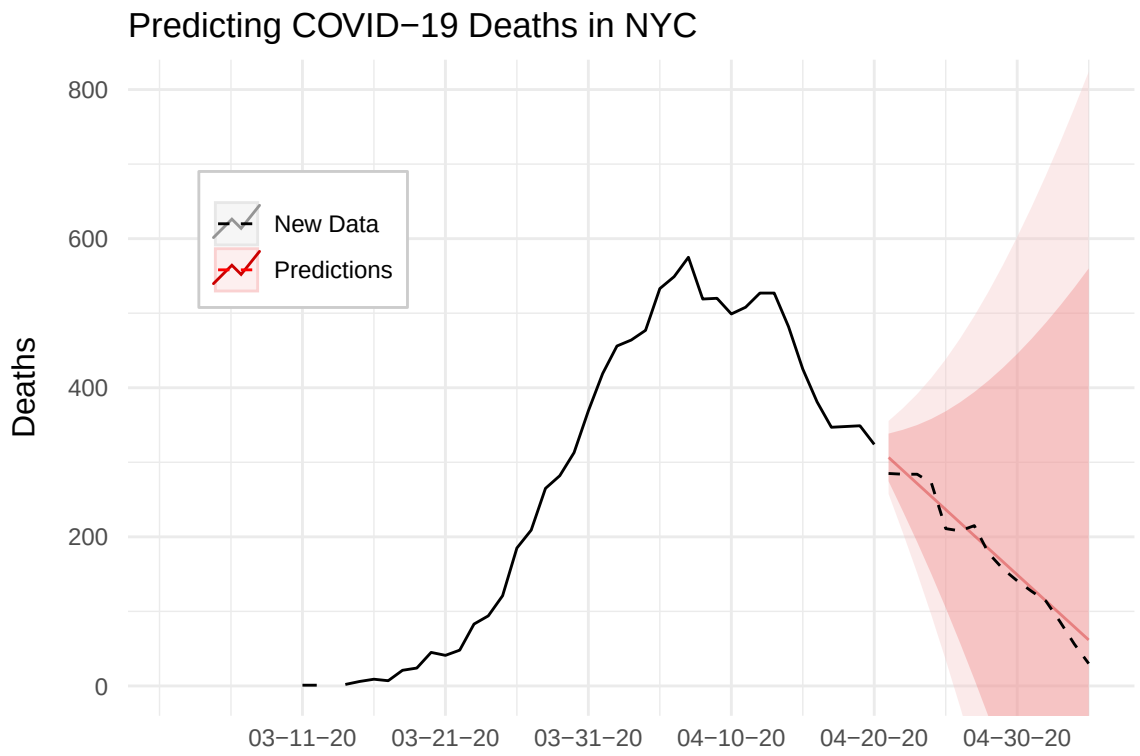

Figure 8: Actual and Forecasted COVID-19 Deaths in NYC

From Figure 8, we can see that the predicted and actual number of daily COVID-19 deaths in NYC are nearly identical. The model accurately predicted the true values from April 21 to May 5.

Conclusion

When it comes to a pandemic, there are many factors that we cannot account for in our models. We cannot predict human behavior or how policies to combat COVID-19 may change over time, and we cannot capture how the virus may naturally mutate into different strains over time as well. Despite these limitations, however, we were able to predict trends in the daily number of COVID-19 cases, hospitalization, and deaths in New York City, New York, with a high or moderate degree of accuracy. Fortunately, it seems like new COVID-19 cases, hospitalizations, and deaths in New York City are decreasing every day, based on a visual inspection of the most recent data in Figures 6-8.

References

CDC. 2020a. "Coronavirus Disease 2019 (COVID-19)." *Centers for Disease Control and Prevention*. https://www.cdc.gov/coronavirus/2019-ncov/need-extra-precautions/people-at-higher-risk.html.

———. 2020b. "Coronavirus Disease 2019 (COVID-19) Situation Summary." *Centers for Disease Control and Prevention*. https://www.cdc.gov/coronavirus/2019-ncov/cases-updates/summary.html.

———. 2020c. "Coronavirus Disease 2019 (COVID-19) – Symptoms." *Centers for Disease Control and Prevention*. https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html.

Grolemund, Garrett, and Hadley Wickham. 2011. "Dates and Times Made Easy with lubridate." *Journal of Statistical Software* 40 (3): 1–25. http://www.jstatsoft.org/v40/i03/.

Hyndman, Rob. 2018. *Fpp2: Data for "Forecasting: Principles and Practice" (2nd Edition)*. https://CRAN.R-project.org/package=fpp2.

Hyndman, Rob J, and Yeasmin Khandakar. 2008. "Automatic Time Series Forecasting: The Forecast Package for R." *Journal of Statistical Software* 26 (3): 1–22. http://www.jstatsoft.org/article/view/v027i03.

"United States Coronavirus: 1,245,857 Cases and 73,145 Deaths - Worldometer." n.d. Accessed May 6, 2020. https://www.worldometers.info/coronavirus/country/us/.

"U.S. Census Bureau QuickFacts: New York City, New York." n.d. Accessed May 6, 2020. https://www.census.gov/quickfacts/fact/table/newyorkcitynewyork/POP010210.

"WHO Director-General's Opening Remarks at the Media Briefing on COVID-19 - 11 March 2020." n.d. Accessed May 8, 2020. https://www.who.int/dg/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Wickham, Hadley, Jim Hester, and Romain Francois. 2018. *Readr: Read Rectangular Text Data*. https://CRAN.R-project.org/package=readr.

Wickham, Hadley, and Dana Seidel. 2019. *Scales: Scale Functions for Visualization*. https://CRAN.R-project.org/package=scales.

Zhu, Hao. 2019. *KableExtra: Construct Complex Table with 'Kable' and Pipe Syntax*. https://CRAN.R-project.org/package=kableExtra.

Appendix

Table 1: ARIMA Models for COVID-19 Hospitalizations

p	d	q	fit.df	p.value
6	2	1	7	0.3357596
6	2	2	8	0.2396983
6	2	6	12	0.2032512
6	2	3	9	0.1710120
6	2	4	10	0.1664724
6	2	5	11	0.0941617
6	2	0	6	0.0521037

Table 2: ARIMA Models for COVID-19 Cases

p	d	q	fit.df	p.value
4	1	4	8	0.6648433
1	1	4	5	0.0881854
0	0	4	4	0.0784502
2	1	3	5	0.0747694
4	1	3	7	0.0671327

Ljung-Box Tests

Deaths

Model 1

```
##
##  Box-Ljung test
##
## data:  deathARIMA1$residuals
## X-squared = 23.374, df = 19, p-value = 0.2212
```

Hospitalizations

Model 1

```
##
##  Box-Ljung test
##
## data:  hospARIMA1$residuals
## X-squared = 52.429, df = 17, p-value = 1.759e-05
```

Model 2

```
##
##  Box-Ljung test
##
## data:  hospARIMA2$residuals
## X-squared = 14.558, df = 13, p-value = 0.3358
```

Cases

Model 1

```
##
##  Box-Ljung test
##
## data:  casesARIMA1$residuals
## X-squared = 33.83, df = 18, p-value = 0.01322
```

Model 2

```
##
##  Box-Ljung test
##
## data:  casesARIMA2$residuals
## X-squared = 9.4413, df = 12, p-value = 0.6648
```

Final Models

Cases

```
## Series: cases_train_ts
## ARIMA(4,1,4)
##
## Coefficients:
##          ar1      ar2     ar3      ar4      ma1    ma2      ma3     ma4
##       0.8586  -1.4181  0.7978  -0.9174  -1.0535  1.254  -0.6946  0.7441
## s.e.  0.0780   0.1119  0.1033   0.0669   0.1718  0.262   0.2514  0.1693
##
## sigma^2 estimated as 394038:  log likelihood=-384.13
## AIC=786.25   AICc=790.87   BIC=803.28
```

Hospitalizations

```
## Series: hosp_train_ts
## ARIMA(6,2,1)
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5      ar6     ma1
##       -1.2546  -1.2090  -1.1486  -0.8146  -0.6526  -0.6546  0.7588
## s.e.   0.1190   0.1926   0.2298   0.2245   0.1818   0.1052  0.1537
##
## sigma^2 estimated as 6801:  log likelihood=-273.54
## AIC=563.07   AICc=566.86   BIC=577.87
```

Deaths

```
## Series: deaths_train_ts
## ARIMA(0,2,1)
##
## Coefficients:
##           ma1
##       -0.6185
## s.e.   0.1335
##
## sigma^2 estimated as 622.9:  log likelihood=-180.03
```

```
## AIC=364.07   AICc=364.4   BIC=367.39
```

Forecasts

Table 3: Actual and Forecasted COVID-19 Deaths with Prediction Intervals

Date	Actual	Forecasted	Lo 80	Hi 80	Lo 95	Hi 95
2020-04-21	285	306.51230	274.526745	338.4978	257.59462	355.4300
2020-04-22	284	289.02459	234.476061	343.5731	205.59981	372.4494
2020-04-23	284	271.53689	193.082202	349.9916	151.55080	391.5230
2020-04-24	272	254.04919	149.839782	358.2586	94.67466	413.4237
2020-04-25	211	236.56148	104.702978	368.4200	34.90130	438.2217
2020-04-26	208	219.07378	57.723007	380.4246	-27.69093	465.8385
2020-04-27	215	201.58608	8.973625	394.1985	-92.98925	496.1614
2020-04-28	178	184.09837	-41.469482	409.6662	-160.87789	529.0746
2020-04-29	157	166.61067	-93.534887	426.7562	-231.24763	564.4690
2020-04-30	141	149.12296	-147.157062	445.4030	-303.99823	602.2442
2020-05-01	127	131.63526	-202.276443	465.5470	-379.03862	642.3091
2020-05-02	114	114.14756	-258.838988	487.1341	-456.28614	684.5813
2020-05-03	86	96.65985	-316.795589	510.1153	-535.66568	728.9854
2020-05-04	56	79.17215	-376.101485	534.4458	-617.10879	775.4531
2020-05-05	30	61.68445	-436.715734	560.0846	-700.55285	823.9217

Table 4: Actual and Forecasted COVID-19 Hospitalizations with Prediction Intervals

Date	Actual	Forecasted	Lo 80	Hi 80	Lo 95	Hi 95
2020-04-21	526	404.228271	298.54069	509.9159	242.593061	565.8635
2020-04-22	432	301.578173	110.67585	492.4805	9.618258	593.5381
2020-04-23	368	171.165117	-71.75677	414.0870	-200.351850	542.6821
2020-04-24	356	224.744676	-66.83368	516.3230	-221.185938	670.6753
2020-04-25	270	35.965910	-315.68649	387.6183	-501.840020	573.7718
2020-04-26	245	-9.575833	-425.69817	406.5465	-645.980024	626.8284
2020-04-27	314	-20.719834	-493.14352	451.7039	-743.229511	701.7898
2020-04-28	287	-121.961944	-680.53240	436.6085	-976.221731	732.2978
2020-04-29	212	-220.747522	-869.10543	427.6104	-1212.325380	770.8303
2020-04-30	154	-332.150190	-1061.17788	396.8775	-1447.101827	782.8014
2020-05-01	198	-290.065747	-1098.73355	518.6021	-1526.816421	946.6849
2020-05-02	136	-470.918209	-1371.28516	429.4487	-1847.910631	906.0742
2020-05-03	117	-508.877513	-1504.34171	486.5867	-2031.308629	1013.5536
2020-05-04	73	-565.214996	-1648.57160	518.1416	-2222.065946	1091.6360
2020-05-05	19	-633.601337	-1820.82672	553.6240	-2449.305867	1182.1032

Table 5: Actual and Forecasted COVID-19 Cases with Prediction Intervals

Date	Actual	Forecasted	Lo 80	Hi 80	Lo 95	Hi 95
2020-04-21	3002	4289.705	3469.8010	5109.609	3035.76999	5543.640
2020-04-22	3414	3855.659	2789.2443	4922.074	2224.71835	5486.600
2020-04-23	2787	3667.520	2527.1169	4807.924	1923.42375	5411.617
2020-04-24	2398	3475.725	2249.1943	4702.255	1599.90846	5351.541
2020-04-25	1514	2703.442	1333.0102	4073.873	607.54765	4799.335
2020-04-26	990	2560.416	1070.1011	4050.732	281.17585	4839.657
2020-04-27	2230	3552.345	1934.0567	5170.633	1077.38671	6027.303
2020-04-28	2641	4166.706	2366.4884	5966.923	1413.51079	6919.901
2020-04-29	2264	3881.961	1952.4874	5811.434	931.08582	6832.836
2020-04-30	1922	3688.794	1707.9059	5669.681	659.28735	6718.300
2020-05-01	1681	3506.867	1474.2599	5539.474	398.26272	6615.471
2020-05-02	989	2833.832	717.0551	4950.609	-403.49902	6071.164
2020-05-03	630	2621.053	424.5506	4817.555	-738.20731	5980.312
2020-05-04	1137	3424.835	1141.6839	5707.985	-66.94302	6916.612
2020-05-05	699	4046.688	1639.9823	6453.394	365.94919	7727.427

R Code

```
knitr::opts_chunk$set(echo = FALSE, message = FALSE, warning = FALSE,
                      fig.width = 6, fig.height = 4, fig.pos = 'H',
                      out.extra = '')
# Required packages
library(forecast)
library(fpp2)
library(ggplot2); theme_set(theme_minimal()) # Set theme for all plots
library(kableExtra)
library(lubridate)
library(readr)
library(tidyverse)

# --- Data Cleaning ---
# Cases, Hospitalizations, and Deaths ---
# Read the data
case_hosp_death <- read_csv("case-hosp-death.csv")

# Training Data
case_hosp_death_train <- case_hosp_death[1:50,]

# Change variable names and date format
daily_df_train <- case_hosp_death_train %>%
  transmute(Date = mdy(DATE_OF_INTEREST),
            Cases = CASE_COUNT,
            Hosp = HOSPITALIZED_COUNT,
            Death = DEATH_COUNT)

# Testing Data
case_hosp_death_test <- case_hosp_death[51:65,]
# Change variable names and date format
daily_df_test<- case_hosp_death_test %>%
  transmute(Date = mdy(DATE_OF_INTEREST),
            Cases = CASE_COUNT,
            Hosp = HOSPITALIZED_COUNT,
            Death = DEATH_COUNT)

# Deaths by Category ---
# Read the data
probable_confirmed_dod <- read_csv("probable-confirmed-dod.csv")[1:40,]
# Change variable names and date format
probable_df <- probable_confirmed_dod %>%
  transmute(Date = mdy(DATE_OF_INTEREST),
            Confirmed = CONFIRMED_DEATHS,
            Probable = PROBABLE_DEATHS)
# Combined deaths = confirmed deaths + probable deaths
probable_df['Combined'] <- probable_df$Confirmed + probable_df$Probable

# --- Time Series Objects ---
# Training Data ---
# Cases
cases_train_ts <- ts(daily_df_train$Cases)
```

```r
# Hospitalizations
hosp_train_ts <- ts(daily_df_train$Hosp)
# Deaths
deaths_train_ts <- ts(daily_df_train$Death)

# Testing Data ---
# Cases
cases_test_ts <- ts(daily_df_test$Cases)
# Hospitalizations
hosp_test_ts <- ts(daily_df_test$Hosp)
# Deaths
deaths_test_ts <- ts(daily_df_test$Death)

# Dates for labeling
dates <-  c(ymd("2020-03-01"), ymd("2020-03-11"),
            ymd("2020-03-21"), ymd("2020-03-31"),
            ymd("2020-04-10"), ymd("2020-04-20"))

# --- EDA: Daily Reported Numbers ---
ggplot(daily_df_train) +
  geom_line(aes(x=Date, y=Cases), lty=1) +
  geom_line(aes(x=Date, y=Hosp), lty=2) +
  geom_line(aes(x=Date, y=Death), lty=3) +
  annotate("text",
           x = c(mdy("4/1/2020")),
           y = c(6250, 1950, 700),
           label = c("New Cases", "Hospitalizations", "Deaths"),
           color = c("black", "black", "black"), size = 3.5) +
  scale_y_continuous(name = "Daily Number",
                     limits = c(0, 6500),
                     breaks = seq(0, 6000, 1000),
                     labels = scales::unit_format(big.mark = ",", unit = "")) +
  scale_x_date(labels = scales::date_format("%m-%d-%y"),
               breaks = dates) +
  theme(axis.title.x = element_blank(),
        axis.text.y = element_text(size=11),
        axis.title.y = element_text(size=12,
                                    margin=ggplot2::margin(r=10)),
        plot.title = element_blank())

# --- EDA: Deaths by Category ---
ggplot(probable_df) +
  geom_line(aes(x=Date, y=Confirmed), lty=1) +
  geom_line(aes(x=Date, y=Probable), lty=3) +
  geom_line(aes(x=Date, y=Combined), lty=2) +
  annotate("text", size=3,
           x=c(mdy("4/9/2020")),
           y=c(400, 190, 630),
           label=c("Confirmed","Probable", "Combined")) +
  scale_y_continuous(name = "Reported Number",
                     limits = c(0, 810),
                     breaks = seq(0, 800, 200),
                     labels = seq(0, 800, 200)) +
```

```
  scale_x_date(labels = scales::date_format("%m-%d-%y"),
               breaks = dates[2:6]) +
  theme(axis.title.x = element_blank(),
        axis.text.y = element_text(size=11),
        axis.title.y = element_text(size=12,
                                    margin=ggplot2::margin(r=10)),
        plot.title = element_blank())

# --- Modeling of Deaths, Hospitalizations, and Cases ---
## Models
model_death <- auto.arima(deaths_train_ts)
model_hosp <- auto.arima(hosp_train_ts)
model_cases <- auto.arima(cases_train_ts)

## Modeling based on the auto.arima() function
deathARIMA1 <- Arima(deaths_train_ts, order=c(0,2,1),
                     include.mean=FALSE)
hospARIMA1 <- Arima(hosp_train_ts, order=c(2,2,1),
                    include.mean=FALSE)
casesARIMA1 <- Arima(cases_train_ts, order=c(0,1,2),
                     include.mean=FALSE)

# --- Ljung-Box Tests ---
# Deaths ---
Box.test(deathARIMA1$residuals, lag = 20, fitdf = 1, type = "Ljung-Box")
## Our p-value is 0.2212, indicating that we do not
## have autocorrelation, so this is a good model.

# Hospitalizations ---
# Model 1
Box.test(hospARIMA1$residuals, lag = 20, fitdf = 3, type = "Ljung-Box")
## There is still some autocorrelation in our residuals,
## so this is not a good model. Let's try something else.
# Hospitalizations - All Reasonable Models ---
# All combinations of p, d, and q
params <- expand.grid(p=seq(0,6), d=seq(0,2), q=seq(0,6))
# fitdf for Box.test() is equal to p+q
params$fit.df <- params$p + params$q
# Empty vector for p-values from Ljung-Box test
params$p.value <- rep(NA, nrow(params))

# Create an ARIMA model and apply the Ljung-Box test
# for all combinations in params
for(i in 1:nrow(params)) {
  # Model
  hospARIMA_temp <- Arima(hosp_train_ts,
                      order = c(params$p[i], params$d[i], params$q[i]),
                      include.mean = FALSE, method="ML")
  # Ljung-Box test p-value
  params$p.value[i] <- Box.test(hospARIMA_temp$residuals,
                                lag=20,
                                fitdf = params$fit.df[i],
                                type="Ljung-Box")$p.value
```

```
}

# Hospitalizations - All Models Table ---
# ARIMA models with Ljung-Box test p-values > 0.05
params <- params %>%
  filter(p.value > 0.05) %>%
  arrange(desc(p.value))

# Display table
kable(params, format="latex", align = rep('c',5),
      caption = "ARIMA Models for COVID-19 Hospitalizations") %>%
  kable_styling(latex_options = "HOLD_position")

# The simplest model is the ARIMA(6,2,0), but its
# p-value is quite low, so we will go with the ARIMA(6,2,1)
# model because it only has one more term and its p-value
# is quite large, which indicates that its residuals
# are uncorrelated.

# Hospitalizations ---
# Final Model
hospARIMA2 <- Arima(hosp_train_ts, order=c(6,2,1),
                    include.mean = FALSE)

# Hospitalizations ---
Box.test(hospARIMA2$residuals, lag=20, fitdf = 7, type="Ljung-Box")
# There is no evidence that the residuals are correlated.
# This is a good model, and it has the largest p-value
# of any that were tested.

# New Cases ---
# Model 1
Box.test(casesARIMA1$residuals, lag = 20, fitdf = 2, type = "Ljung-Box")
# Since the p-value is small, we should try a different model.

# Cases - All Reasonable Models ---
# All combinations of p, d, and q
params2 <- expand.grid(p=seq(0,4), d=seq(0,2), q=seq(0,4))
# fitdf for Box.test() is equal to p+q
params2$fit.df <- params2$p + params2$q
# Empty vector for p-values from Ljung-Box test
params2$p.value <- rep(NA, nrow(params2))

# Create an ARIMA model and apply the Ljung-Box test
# for all combinations in params
for(i in 1:nrow(params2)) {

  # Model
  casesARIMA_temp <- Arima(cases_train_ts,
                       order = c(params2$p[i], params2$d[i], params2$q[i]),
                       include.mean = FALSE, method = "ML")

  # Ljung-Box test p-value
```

```
  params2$p.value[i] <- Box.test(casesARIMA_temp$residuals,
                                 lag=20,
                                 fitdf = params2$fit.df[i],
                                 type="Ljung-Box")$p.value
}

# Cases - All Models Table ---
# ARIMA models with Ljung-Box test p-values > 0.05
params2 <- params2 %>%
  filter(p.value > 0.05) %>%
  arrange(desc(p.value))

# Display table
kable(params2, format = "latex", align = rep('c',5),
      caption = "ARIMA Models for COVID-19 Cases") %>%
  kable_styling(latex_options = "HOLD_position")
# New Cases ---
# Final Model
casesARIMA2 <- Arima(cases_train_ts, order=c(4,1,4),
                     include.mean = FALSE, method="ML")

# Cases
Box.test(casesARIMA2$residuals, lag=20, fitdf = 8, type="Ljung-Box")
# There is no evidence that the residuals are correlated.
# This is a good model, and it has the largest p-value
# of any that were tested.

# Cases ---

# Data frames for plotting
daily_df_train$fit.cases <- fitted(casesARIMA2)

# TS with fitted line: ARIMA(4,1,4)
ggplot(daily_df_train) +
  geom_line(aes(x=Date, y=Cases, color="Time Series Data")) +
  geom_line(aes(x=Date, y=fit.cases, color="Fitted ARIMA(4, 1, 4)")) +
  scale_color_manual(values = c("red", "black")) +
  scale_x_date(labels = scales::date_format("%m-%d-%y"),
               breaks = dates) +
  scale_y_continuous(name = "New Cases",
                     limits = c(0, 7000),
                     breaks = seq(0, 7000, 1000),
                     labels = scales::unit_format(big.mark = ",", unit = "")) +
  labs(title = "Daily COVID-19 Cases in New York City, New York") +
  theme(legend.position = c(0.198, 0.825),
        legend.background = element_rect(fill = "white", color = "grey80"),
        legend.title = element_blank(),
        legend.direction = "vertical",
        axis.title.y = element_text(size=11,
                                    margin=ggplot2::margin(r=10)),
        axis.title.x = element_blank(),
        plot.title = element_text(size=12))
```

```r
# Hospitalizations ---

# Data frames for plotting
daily_df_train$fit.hosp <- fitted(hospARIMA2)

# TS with fitted line: ARIMA(6,2,1)
ggplot(daily_df_train) +
  geom_line(aes(x=Date, y=Hosp, color = "Time Series Data")) +
  geom_line(aes(x=Date, y=fit.hosp, color = "Fitted ARIMA(6, 2, 1)")) +
  scale_color_manual(values = c("red", "black")) +
  scale_y_continuous(name = "Hospitalizations",
                     limits = c(0, 1800),
                     breaks = seq(0, 1800, 200),
                     labels = scales::unit_format(big.mark = ",", unit = "")) +
  scale_x_date(labels = scales::date_format("%m-%d-%y"),
               breaks = dates) +
  labs(title = "Daily COVID-19 Hospitalizations in New York City, New York") +
  theme(legend.position = c(0.228, 0.828),
        legend.background = element_rect(fill = "white", color = "grey80"),
        legend.title = element_blank(),
        legend.direction = "vertical",
        axis.title.y = element_text(size=11,
                                    margin=ggplot2::margin(r=10)),
        axis.title.x = element_blank(),
        plot.title = element_text(size=12))

# --- Time Series Plots with Overlayed Models ---
# Deaths ---

# Data frames for plotting
daily_df_train$fit.death <- fitted(deathARIMA1)
daily_deaths_train <- daily_df_train[10:50,] # Get rid of leading NA's

# TS with fitted line: ARIMA(0,2,1)
ggplot(daily_deaths_train) +
  geom_line(aes(x=Date, y=Death, color="Time Series Data")) +
  geom_line(aes(x=Date, y=fit.death, color="Fitted ARIMA(0, 2, 1)")) +
  scale_color_manual(values = c("red", "black")) +
  scale_y_continuous(name = "Deaths",
                     limits = c(0, 600),
                     breaks = seq(0, 600, 100),
                     labels = seq(0, 600, 100)) +
  scale_x_date(labels = scales::date_format("%m-%d-%y"),
               breaks = dates[2:6]) +
  labs(title = "Daily COVID-19 Deaths in New York City, New York") +
  theme(legend.position = c(0.216, 0.804),
        legend.background = element_rect(fill = "white", color = "grey80"),
        legend.title = element_blank(),
        legend.direction = "vertical",
        axis.title.y = element_text(size=11,
                                    margin=ggplot2::margin(r=10)),
        axis.title.x = element_blank(),
        plot.title = element_text(size=12))
```

```r
# --- Time Series Objects for New Data (Testing Data) ---
# Cases
cases_test_ts <- ts(daily_df_test$Cases, start=51)
# Hospitalizations
hosp_test_ts <- ts(daily_df_test$Hosp, start=51)
# Deaths
deaths_test_ts <- ts(daily_df_test$Death, start=51)

# Predictions based on final models
deaths.predict <- forecast(deathARIMA1, h=nrow(daily_df_test))
hosp.predict <- forecast(hospARIMA2, h=nrow(daily_df_test))
cases.predict <- forecast(casesARIMA2, h=nrow(daily_df_test))

# --- Prediction Plots ---
# Dates for labeling
dates2 <- c("03-01-20", "03-11-20", "03-21-20",
            "03-31-20", "04-10-20", "04-20-20", "04-30-20")

# Cases ---
autoplot(cases_train_ts) +
  autolayer(cases.predict, alpha=0.35, series = "Predictions") +
  autolayer(cases_test_ts, lty=2, series = "New Data") +
  labs(title = "Predicting COVID-19 Cases in NYC") +
  scale_color_manual(guide = 'legend',
                     labels = c("New Data", "Predictions"),
                     values = c("black", "red")) +
  scale_y_continuous(name = "New Cases",
                     breaks = seq(0, 7000, 1000),
                     labels = scales::unit_format(big.mark = ",", unit = "")) +
  scale_x_continuous(name = "", breaks = seq(0,60,10),
                     labels = dates2) +
  coord_cartesian(ylim = c(0, 7200)) +
  theme(legend.position = c(0.174, 0.804),
        legend.background = element_rect(fill = "white", color = "grey80"),
        legend.title = element_blank(),
        legend.direction = "vertical",
        axis.title.y = element_text(size=12,
                                    margin=ggplot2::margin(r=10)),
        axis.title.x = element_blank())

# Hospitalizations ---
autoplot(hosp_train_ts) +
  autolayer(hosp.predict, alpha=0.35, series = "Predictions") +
  autolayer(hosp_test_ts, lty=2, series = "New Data") +
  labs(title = "Predicting COVID-19 Hospitalizations in NYC") +
  scale_color_manual(guide = 'legend',
                     labels = c("New Data", "Predictions"),
                     values = c("black", "red")) +
  scale_y_continuous(name = "Hospitalizations",
                     breaks = seq(0, 1800, 200),
                     labels = scales::unit_format(big.mark = ",", unit = "")) +
  scale_x_continuous(name = "", breaks = seq(0,60,10),
                     labels = dates2) +
```

```
  coord_cartesian(ylim = c(0, 1800)) +
  theme(legend.position = c(0.174, 0.828),
        legend.background = element_rect(fill = "white", color = "grey80"),
        legend.title = element_blank(),
        legend.direction = "vertical",
        axis.title.y = element_text(size=12,
                                    margin=ggplot2::margin(r=10)),
        axis.title.x = element_blank())

# Deaths ---
autoplot(deaths_train_ts) +
  autolayer(deaths.predict, alpha=0.35, series = "Predictions") +
  autolayer(deaths_test_ts, lty=2, series = "New Data") +
  labs(title = "Predicting COVID-19 Deaths in NYC") +
  scale_color_manual(guide = 'legend',
                     labels = c("New Data", "Predictions"),
                     values = c("black", "red")) +
  scale_y_continuous(name = "Deaths",
                     breaks = seq(0, 800, 200),
                     labels = scales::unit_format(big.mark = ",", unit = "")) +
  scale_x_continuous(name = "", breaks = seq(10,60,10),
                     labels = dates2[2:length(dates2)]) +
  coord_cartesian(ylim = c(0, 800)) +
  theme(legend.position = c(0.174, 0.726),
        legend.background = element_rect(fill = "white", color = "grey80"),
        legend.title = element_blank(),
        legend.direction = "vertical",
        axis.title.y = element_text(size=12,
                                    margin=ggplot2::margin(r=10)),
        axis.title.x = element_blank())
# Cases ---
casesARIMA2

# Hospitalizations ---
hospARIMA2

# --- Final Models ---
# Deaths ---
deathARIMA1

# --- Predictions ---
# Deaths ---
deaths.predict2 <- as.data.frame(deaths.predict) %>%
  # Add dates and actual cases
  mutate(Date = daily_df_test$Date,
         Actual = daily_df_test$Death)
# Reorder columns
deaths.predict2 <- deaths.predict2[,c(6,7,seq(1,5))]

# Rename columns
names1 <- colnames(deaths.predict2)
colnames(deaths.predict2) <- c("Date", "Actual", "Forecasted", names1[4:7])
```

```r
# Display
kable(deaths.predict2, format="latex", align = c('l', rep('c', 6)),
      caption = "Actual and Forecasted COVID-19 Deaths with Prediction Intervals") %>%
  kable_styling(latex_options = "HOLD_position")
# Hospitalizations
hosp.predict2 <- as.data.frame(hosp.predict) %>%
  # Add dates and actual cases
  mutate(Date = daily_df_test$Date,
         Actual = daily_df_test$Hosp)

# Reorder columns
hosp.predict2 <- hosp.predict2[,c(6,7,seq(1,5))]

# Rename columns
colnames(hosp.predict2) <- c("Date", "Actual", "Forecasted", names1[4:7])

# Display
kable(hosp.predict2, format="latex", align = c('l', rep('c', 6)),
      caption = "Actual and Forecasted COVID-19 Hospitalizations with Prediction Intervals") %>%
  kable_styling(latex_options = "HOLD_position")
# Cases ---
cases.predict2 <- as.data.frame(cases.predict) %>%
  # Add dates and actual cases
  mutate(Date = daily_df_test$Date,
         Actual = daily_df_test$Cases)
# Reorder columns
cases.predict2 <- cases.predict2[,c(6,7,seq(1,5))]

# Rename columns
colnames(cases.predict2) <- c("Date", "Actual", "Forecasted", names1[4:7])

# Display
kable(cases.predict2, format="latex", align = c('l', rep('c', 6)),
      caption = "Actual and Forecasted COVID-19 Cases with Prediction Intervals") %>%
  kable_styling(latex_options = "HOLD_position")
```