Excess Deaths in the United States during the COVID-19 Pandemic

1. Introduction:

The COVID-19 pandemic caused considerable turmoil and loss of life in the United States and around the world. The frequently cited death toll from the pandemic records only individuals who died from COVID with a laboratory-confirmed positive test result. However, many individuals who died, especially early in the pandemic, were never tested due to the shortage of tests. Therefore, the COVID death total may be an underestimate of the true number of COVID-associated deaths during the pandemic. One commonly used method to estimate the number of deaths from any mass-fatality event is examining excess deaths.[1] That is, the difference between the recorded number of deaths and the expected number of deaths during some given period. The purpose of this project is to analyze excess mortality in the United States during the pandemic.

2. Methods:

The data used for this project was collected from the National Center for Health Statistics (NCHS) mortality surveillance from October 2009 to February 2022.[2] This dataset includes the total number of deaths (from all causes) in the United States for each week. The data was split into a training dataset (October 2009 to January 2020) and a testing dataset (January 2020 to February 2022). The split between the two datasets in January 2020 represents the first recorded COVID death in the United States. The training dataset was used to develop a model for U.S. mortality prior to the pandemic. This model was then applied to the time period of the testing dataset to estimate hypothetical weekly deaths. These predictions were compared with recorded total deaths in the testing dataset.

Since weekly deaths are highly correlated, traditional linear regression / generalized linear model approaches cannot be used. Instead, time series methods were investigated and applied to the data. Specifically, ARIMA modeling on the testing dataset was used, which combines autoregressive and moving average models.[3] In order to apply such methods, the data must be stationary; that is, the statistical properties of the time series are independent of the position in the time series. Since deaths in the United States clearly follow a seasonal pattern, seasonal differencing was required to render the time series stationary. There are many different statistical tests to assess stationarity. For this analysis, the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test was used. Under the null hypothesis, the time series is stationary. Under the alternative hypothesis, the time series is not stationary. The URCA package in R was used to perform this test.[4]

Once stationarity was achieved, model fitting could begin. The model of interest was a seasonal ARIMA model, which incorporates additional seasonal terms to the classical ARIMA models. First, an autoregressive model of order p has the form:[3]

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t$$

This model is called an AR(p) model and looks like a multiple linear regression, but the predictors are lagged values y_t. The ε_t term represents the “white noise.”

Next, a moving average model of order q has the form:[3]

$$y_t = c + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \cdots + \theta_q\varepsilon_{t-q}$$

This model is called a MA(q) model and looks like a multiple linear regression, but the predictors are lagged values of the error ε_t terms.

An autoregressive model can be combined with a moving average model to obtain a non-seasonal AutoRegressive Integrated Moving Average (ARIMA) model, which has the form:[3]

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1\varepsilon_{t-1} + \cdots + \theta_q\varepsilon_{t-q} + \varepsilon_t$$

This model is called an ARIMA(p, d, q) model. Note that y'_t represents the differenced series, and the order of differencing is denoted by d.

Finally, a seasonal ARIMA model has additional seasonal terms, which are represented by $(P, D, Q)_m$, where m is the number of observations per period. For this project, m = 52 since there are (roughly) 52 weeks per year.

$$\text{ARIMA} \quad \underbrace{(p, d, q)}_{\text{Non-seasonal part of the model}} \quad \underbrace{(P, D, Q)_m}_{\text{Seasonal part of of the model}}$$

For the classical (non-seasonal) component of the ARIMA model, autoregressive models of order 0 to 3 and moving average models of order 0 to 3 were considered. For the seasonal component, autoregressive models of order 0 to 2 and moving average models of order 0 to 2 were considered. Examining all possible combinations, there were a total of 144 models constructed. The arima function in R was used to fit these models.[5] To determine the final model, AIC was used. The model with the smallest AIC was selected. Subsequently, model diagnostics were performed to ensure that the selected model was adequate.

Next, the selected ARIMA model was applied to the time period of the testing dataset. The forecast package in R was used for this analysis.[6] The point estimates, 80% prediction intervals, and 95% prediction intervals for weekly deaths were computed. To calculate excess deaths, the difference between the observed number of weekly deaths from the testing dataset and the upper bound of the 95% prediction interval from the model was taken. Excess deaths were calculated for each week in the testing dataset. Laboratory-confirmed COVID-19 deaths were extracted from the testing dataset and compared with computed excess deaths.

3. Results:

The entire dataset consists of 648 weeks of observations, with 535 weeks in the training dataset (before the pandemic) and 113 weeks in the testing dataset (during the pandemic). Figure 1 displays weekly deaths in the United States from October 2009 to February 2022. Prior to the pandemic, weekly deaths followed a clear seasonal pattern, peaking in the winter and bottoming out in the summer. Some years were more severe than others with weekly deaths peaking at

higher levels, likely due to harsh flu epidemics that tend to peak in the winter months. This typical pattern of deaths was broken during the pandemic with deaths much higher than usual and peaking at unusual times of the year.

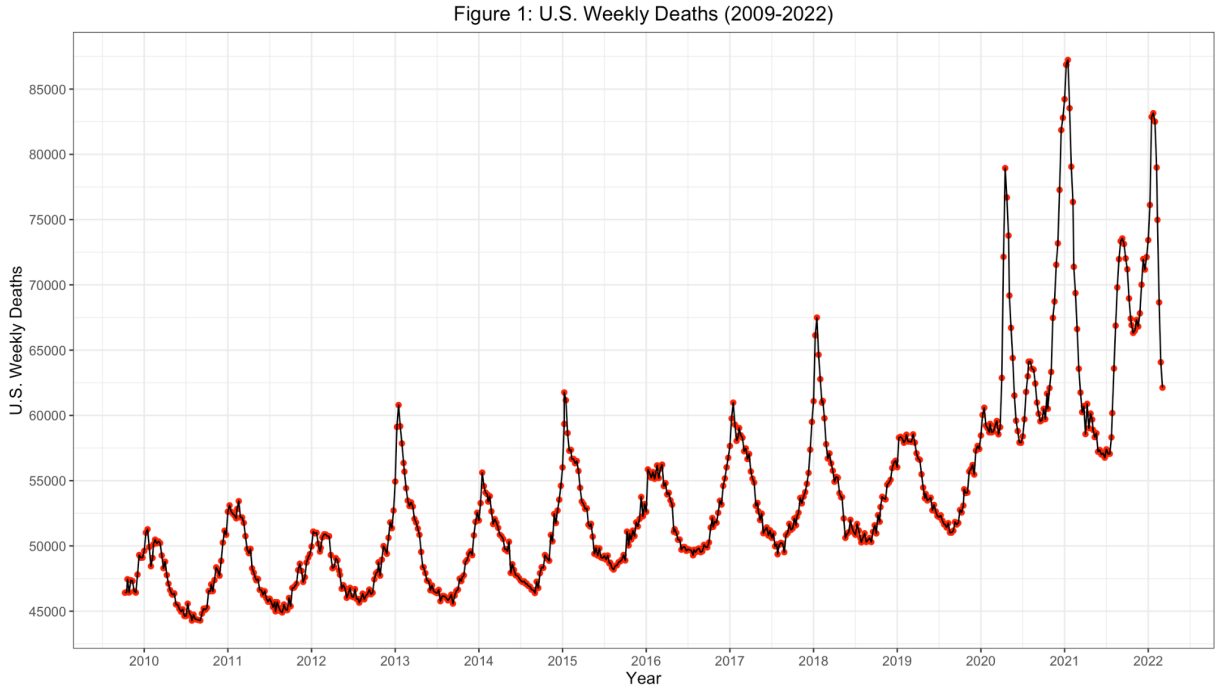


Figure 1: Weekly all-cause deaths in the United States from October 2009 to February 2022. Includes both the training dataset and testing dataset. Observe the clear seasonal pattern in weekly deaths, which tend to peak in the winter and reach their nadir in the summer. Weekly deaths spiked during the COVID-19 pandemic starting in spring 2020.

Before the COVID-19 pandemic (October 2009 to January 2020), the median number of weekly deaths in the United States was 50,749 with interquartile range (47912, 53468). The maximum number of deaths in a single week was 67,495, which occurred in January 2018. During the first two years of the pandemic (January 2020 to February 2022), the median number of U.S. weekly deaths was 63,576 with interquartile range (59570, 71969).

Using the original (non-differenced) training dataset, the autocorrelation graph is displayed in Figure A1. Notice the strong harmonic pattern with autocorrelation peaking at lags of 1.0, 2.0, and 3.0 (which correspond to 1, 2, and 3 years respectively). This indicates the presence of seasonality in the data. The partial autocorrelation plot in Figure A2 shows spikes at lags of multiples of 52 weeks (i.e. multiples of 1 year).

By visual inspection, stationarity is clearly violated for the training dataset. Using the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test for stationarity, the KPSS test statistic was 3.31, which corresponds to a p-value < 0.01. Therefore, the null hypothesis was rejected, which means there is sufficient evidence to suggest that the time series is not stationary. Thus, differencing was necessary. Due to the clear seasonal pattern in the data, the first seasonal difference was

applied, which is graphed in Figure A3. After taking the seasonal difference, the KPSS test statistic was 0.1032, which corresponds to a p-value > 0.10. As a result, the seasonal differenced time series was deemed stationary, and model selection could begin.

Using the seasonal differenced time series from the training data, model selection was performed. A summary of the model selection process is highlighted in Table A1. Note that some models did not converge in R. These models are marked with "Did Not Converge" in the AIC column. The model selection criterion was AIC; namely, the objective was to choose the model with the smallest AIC. The selected model was M513: ARIMA(3, 0, 2)$(0, 1, 2)_{52}$ with an AIC of 7753.536. The parameter estimates of this model are specified in Table A2.

Model diagnostics for the chosen model are displayed in Figure A4. The top graph displays model residuals. Since the model used seasonal differencing, the first 52 weeks of the dataset have no values, which explains the flat line of model residuals from October 2009 to October 2010. Note that there are a handful of observations with large residuals (in absolute values). The bottom plots display autocorrelation and partial autocorrelation, where the lags are measured in weeks. These plots look fairly good with only a few observations extending outside the blue horizontal dashed lines. There do not appear to be any major violations.

Next, the selected model was used to forecast weekly deaths for the next 113 weeks (i.e. the number of weeks in the testing dataset). This forecast represents what U.S. weekly deaths may have looked like if the COVID-19 pandemic never happened.

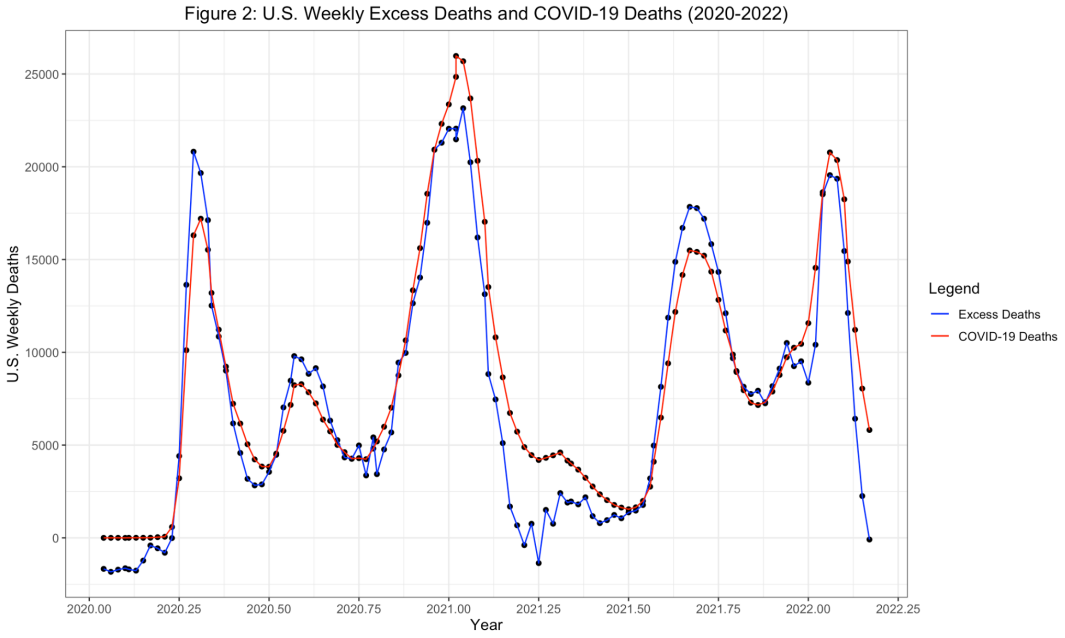


Figure 2: Weekly excess deaths (blue curve) and weekly laboratory-confirmed COVID-19 deaths (red curve) in the United States from January 2020 to February 2022. Excess deaths and COVID deaths are highly correlated, which supports the claim that a large proportion of the excess deaths were related to COVID.

Figure 2 displays calculated weekly excess deaths (blue curve) and observed COVID-19 deaths (red curve) in the United States from January 2020 to February 2022. Noticeable peaks in both occurred in April 2020 (initial pandemic wave), January 2021 (Alpha variant wave), September 2021 (Delta variant wave), and January 2022 (Omicron variant wave). The Pearson correlation coefficient between excess deaths and COVID-19 deaths was approximately 0.951 (95% CI: [0.930, 0.966]), which implies very strong correlation.

According to this methodology, there were a total of 833,510 excess deaths in the United States from January 2020 to February 2022. In comparison, there were 981,196 COVID-19 deaths during this time period. The fact that recorded COVID deaths exceeded excess deaths may have been due to a reduction in deaths from other causes, such as influenza and accidents, during the pandemic when individuals practiced non-pharmaceutical interventions (which prevented the spread of flu) and stayed home (which reduced car accidents). Thus, the baseline number of expected deaths should have been lowered to account for this, which would increase the number of excess deaths.

4. Discussion and Conclusion:

The pattern of weekly excess deaths in the United States closely resembles observed COVID-19 deaths, with peaks in April 2020, January 2021, September 2021, and January 2022. While this analysis does not attribute all of these excess deaths to COVID-19, the strong correlation between excess deaths and COVID-19 deaths supports the notion that most of the excess deaths were related to the virus.

There are several limitations of this work. First, the definition of excess deaths could be improved. This project defined excess deaths as the difference between observed deaths and the upper bound of the 95% prediction interval. More advanced statistical analysis should be implemented to determine a more precise definition. Next, this work only took the seasonal difference to render stationary the time series from the testing dataset. However, the original time series has a slight upward trend over time, so it may be better to take the first difference as well (in addition to the seasonal difference) in order to detrend the series. Further, there are a few potential outliers (from a visual perspective) in the training dataset due to severe flu epidemics in early 2013, 2015, and 2018. Removing some of these weeks and re-running the model without them would be a good idea as part of a sensitivity analysis.

Further research directions include performing this analysis for individual states, counties, or cities. This would allow for comparisons between states or cities to be made. For example, one might be interested in determining whether the gap between confirmed COVID-19 deaths and excess deaths differs between states. In addition, other time series methods besides ARIMA models could also be explored.

5. References:

1. Wang H, Paulson KR, Pease SA, et al. Estimating excess mortality due to the COVID-19 pandemic: a systematic analysis of COVID-19-related mortality, 2020–21. *The Lancet*. 2022;399(10334):1513-1536. doi:https://doi.org/10.1016/s0140-6736(21)02796-3
2. CDC. Weekly U.S. Influenza Surveillance Report. Centers for Disease Control and Prevention. Published May 18, 2022. https://www.cdc.gov/flu/weekly/index.htm
3. Chapter 8 ARIMA models | Forecasting: Principles and Practice. Otexts.com. Published 2019. https://otexts.com/fpp2/arima.html
4. Pfaff B, Zivot E, Stigler M. urca: Unit Root and Cointegration Tests for Time Series Data. R-Packages. Published August 29, 2022. Accessed December 5, 2023. https://cran.r-project.org/web/packages/urca/index.html
5. arima function - RDocumentation. www.rdocumentation.org. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/arima
6. forecast package - RDocumentation. www.rdocumentation.org. Accessed December 9, 2023. https://www.rdocumentation.org/packages/forecast/versions/8.21.1

6. Appendix:

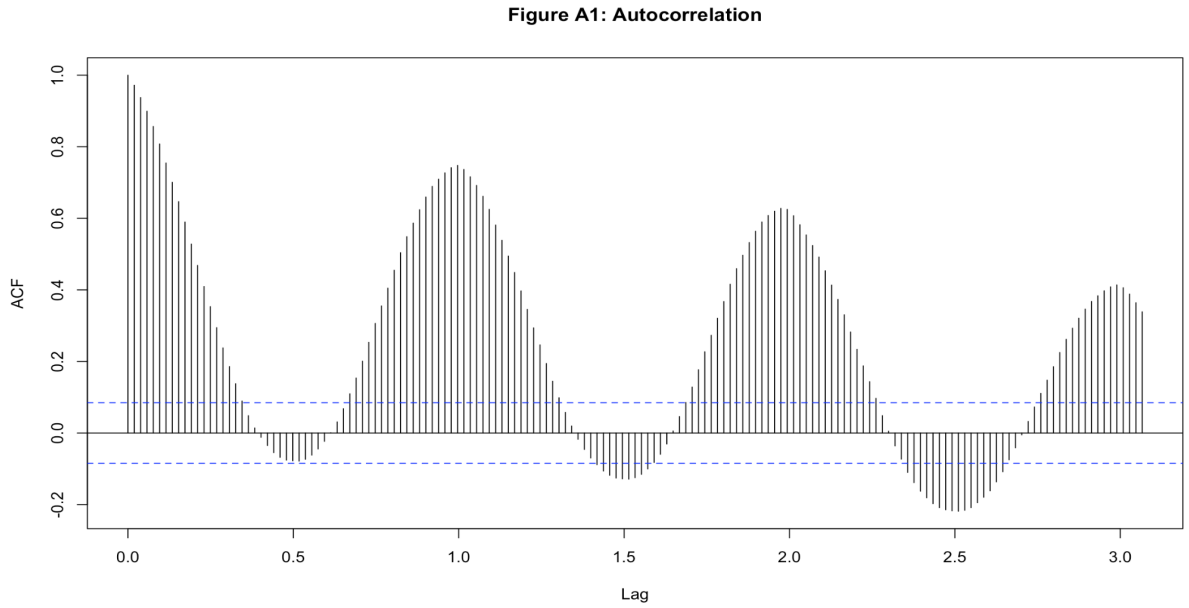


Figure A1: Autocorrelation of the original (non-differenced) training data time series vs. lag (measured in years). Autocorrelation measures the linear relationship between lagged values of a time series. Notice the periodic pattern where autocorrelations peak at the seasonal lags (i.e. at multiples of the seasonal frequency), which indicates seasonality.

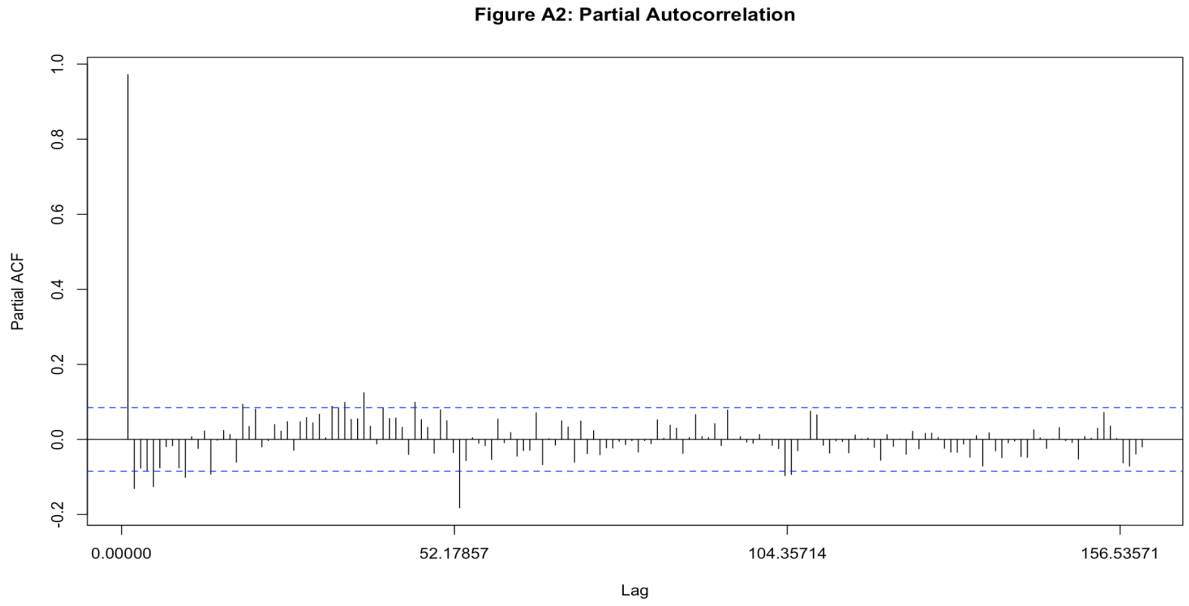


Figure A2: Partial autocorrelation of the original (non-differenced) training data time series vs. lag (measured in weeks). Partial autocorrelation measures the relationship between y_t and y_{t-k} after removing (adjusting for) the effects of lags 1, 2, 3, …, k − 1. Observe the spikes at the seasonal lags (multiples of 52 weeks).

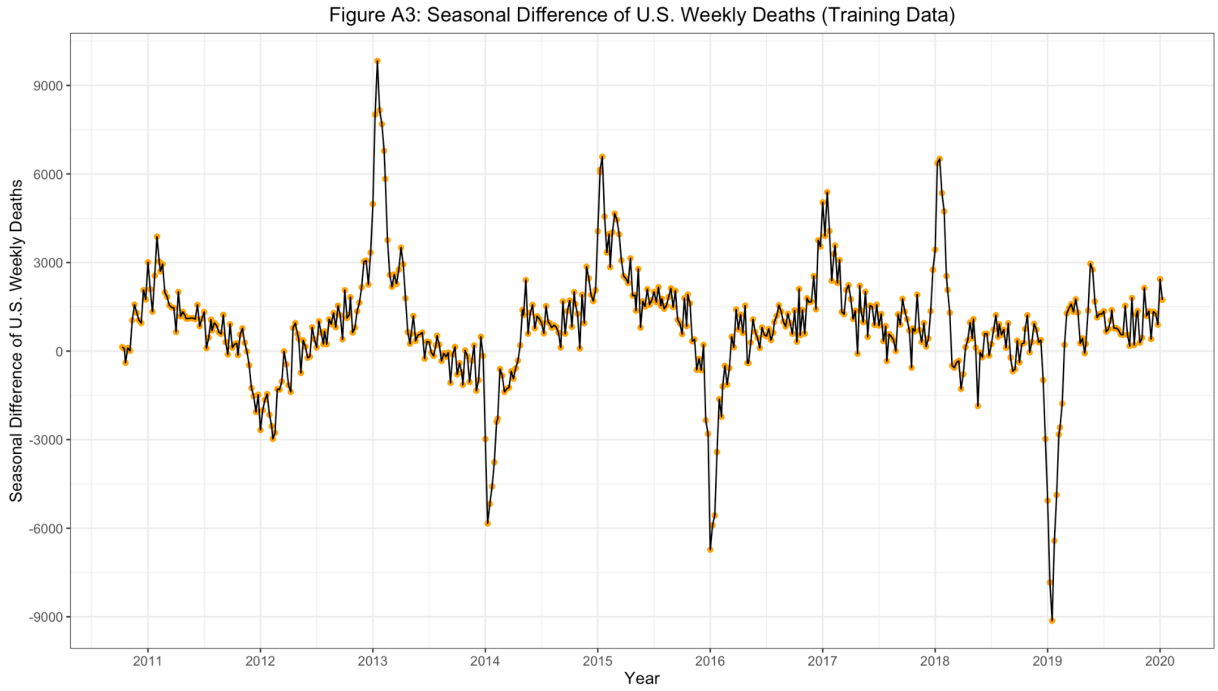


Figure A3: Seasonal difference of weekly deaths in the United States from the training dataset. Since the data are recorded weekly and there are (approximately) 52 weeks per year, seasonal differencing takes the difference of y_t and y_{t-52}. Thus, the seasonal difference cannot be computed for the first 52 weeks. Hence, the first data point occurs in October 2010 (as opposed to October 2009). Notice that this seasonal differenced time series appears much closer to stationary than the original time series, which is supported by the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test.

Table A1: Comparison of Seasonal ARIMA Models

Model Number	Model	Degrees of Freedom	AIC
M000	ARIMA(0, 0, 0)(0, 1, 0)$_{52}$	1	8746.842
M001	ARIMA(1, 0, 0)(0, 1, 0)$_{52}$	2	7940.984
M002	ARIMA(0, 0, 1)(0, 1, 0)$_{52}$	2	8402.390
M003	ARIMA(1, 0, 1)(0, 1, 0)$_{52}$	3	7942.347
M004	ARIMA(2, 0, 0)(0, 1, 0)$_{52}$	3	7942.133
M005	ARIMA(0, 0, 2)(0, 1, 0)$_{52}$	3	8202.089
M006	ARIMA(2, 0, 1)(0, 1, 0)$_{52}$	4	7940.837
M007	ARIMA(1, 0, 2)(0, 1, 0)$_{52}$	4	7931.020

M008	ARIMA(2, 0, 2)$(0, 1, 0)_{52}$	5	7932.988
M009	ARIMA(3, 0, 0)$(0, 1, 0)_{52}$	4	7932.904
M010	ARIMA(0, 0, 3)$(0, 1, 0)_{52}$	4	8070.911
M011	ARIMA(3, 0, 1)$(0, 1, 0)_{52}$	5	7934.869
M012	ARIMA(1, 0, 3)$(0, 1, 0)_{52}$	5	7932.969
M013	ARIMA(3, 0, 2)$(0, 1, 0)_{52}$	6	7931.101
M014	ARIMA(2, 0, 3)$(0, 1, 0)_{52}$	6	7920.020
M015	ARIMA(3, 0, 3)$(0, 1, 0)_{52}$	7	7933.017
M100	ARIMA(0, 0, 0)$(1, 1, 0)_{52}$	2	8689.448
M101	ARIMA(1, 0, 0)$(1, 1, 0)_{52}$	3	7813.772
M102	ARIMA(0, 0, 1)$(1, 1, 0)_{52}$	3	8301.447
M103	ARIMA(1, 0, 1)$(1, 1, 0)_{52}$	4	7812.578
M104	ARIMA(2, 0, 0)$(1, 1, 0)_{52}$	4	7811.740
M105	ARIMA(0, 0, 2)$(1, 1, 0)_{52}$	4	8092.940
M106	ARIMA(2, 0, 1)$(1, 1, 0)_{52}$	5	7810.306
M107	ARIMA(1, 0, 2)$(1, 1, 0)_{52}$	5	7806.952
M108	ARIMA(2, 0, 2)$(1, 1, 0)_{52}$	6	Did Not Converge
M109	ARIMA(3, 0, 0)$(1, 1, 0)_{52}$	5	7807.252
M110	ARIMA(0, 0, 3)$(1, 1, 0)_{52}$	5	7976.747
M111	ARIMA(3, 0, 1)$(1, 1, 0)_{52}$	6	7809.256
M112	ARIMA(1, 0, 3)$(1, 1, 0)_{52}$	6	7808.804
M113	ARIMA(3, 0, 2)$(1, 1, 0)_{52}$	7	Did Not Converge
M114	ARIMA(2, 0, 3)$(1, 1, 0)_{52}$	7	7810.849
M115	ARIMA(3, 0, 3)$(1, 1, 0)_{52}$	8	7797.641
M200	ARIMA(0, 0, 0)$(0, 1, 1)_{52}$	2	8734.368

M201	ARIMA(1, 0, 0)$(0, 1, 1)_{52}$	3	7770.048
M202	ARIMA(0, 0, 1)$(0, 1, 1)_{52}$	3	8336.841
M203	ARIMA(1, 0, 1)$(0, 1, 1)_{52}$	4	7768.149
M204	ARIMA(2, 0, 0)$(0, 1, 1)_{52}$	4	7767.669
M205	ARIMA(0, 0, 2)$(0, 1, 1)_{52}$	4	8124.815
M206	ARIMA(2, 0, 1)$(0, 1, 1)_{52}$	5	7768.095
M207	ARIMA(1, 0, 2)$(0, 1, 1)_{52}$	5	7768.216
M208	ARIMA(2, 0, 2)$(0, 1, 1)_{52}$	6	7765.523
M209	ARIMA(3, 0, 0)$(0, 1, 1)_{52}$	5	7767.988
M210	ARIMA(0, 0, 3)$(0, 1, 1)_{52}$	5	7993.769
M211	ARIMA(3, 0, 1)$(0, 1, 1)_{52}$	6	7769.817
M212	ARIMA(1, 0, 3)$(0, 1, 1)_{52}$	6	7769.407
M213	ARIMA(3, 0, 2)$(0, 1, 1)_{52}$	7	7755.157
M214	ARIMA(2, 0, 3)$(0, 1, 1)_{52}$	7	Did Not Converge
M215	ARIMA(3, 0, 3)$(0, 1, 1)_{52}$	8	Did Not Converge
M300	ARIMA(0, 0, 0)$(1, 1, 1)_{52}$	3	8673.578
M301	ARIMA(1, 0, 0)$(1, 1, 1)_{52}$	4	7769.074
M302	ARIMA(0, 0, 1)$(1, 1, 1)_{52}$	4	8299.999
M303	ARIMA(1, 0, 1)$(1, 1, 1)_{52}$	5	7766.596
M304	ARIMA(2, 0, 0)$(1, 1, 1)_{52}$	5	7766.039
M305	ARIMA(0, 0, 2)$(1, 1, 1)_{52}$	5	8093.601
M306	ARIMA(2, 0, 1)$(1, 1, 1)_{52}$	6	7766.536
M307	ARIMA(1, 0, 2)$(1, 1, 1)_{52}$	6	7766.632
M308	ARIMA(2, 0, 2)$(1, 1, 1)_{52}$	7	7768.240
M309	ARIMA(3, 0, 0)$(1, 1, 1)_{52}$	6	7766.381

M310	ARIMA(0, 0, 3)(1, 1, 1)$_{52}$	6	7978.525
M311	ARIMA(3, 0, 1)(1, 1, 1)$_{52}$	7	7768.288
M312	ARIMA(1, 0, 3)(1, 1, 1)$_{52}$	7	7767.956
M313	ARIMA(3, 0, 2)(1, 1, 1)$_{52}$	8	7770.086
M314	ARIMA(2, 0, 3)(1, 1, 1)$_{52}$	8	7754.658
M315	ARIMA(3, 0, 3)(1, 1, 1)$_{52}$	9	7772.102
M400	ARIMA(0, 0, 0)(2, 1, 0)$_{52}$	3	8650.304
M401	ARIMA(1, 0, 0)(2, 1, 0)$_{52}$	4	7772.077
M402	ARIMA(0, 0, 1)(2, 1, 0)$_{52}$	4	8296.168
M403	ARIMA(1, 0, 1)(2, 1, 0)$_{52}$	5	7769.793
M404	ARIMA(2, 0, 0)(2, 1, 0)$_{52}$	5	Did Not Converge
M405	ARIMA(0, 0, 2)(2, 1, 0)$_{52}$	5	8092.877
M406	ARIMA(2, 0, 1)(2, 1, 0)$_{52}$	6	Did Not Converge
M407	ARIMA(1, 0, 2)(2, 1, 0)$_{52}$	6	7768.843
M408	ARIMA(2, 0, 2)(2, 1, 0)$_{52}$	7	Did Not Converge
M409	ARIMA(3, 0, 0)(2, 1, 0)$_{52}$	6	Did Not Converge
M410	ARIMA(0, 0, 3)(2, 1, 0)$_{52}$	6	7978.392
M411	ARIMA(3, 0, 1)(2, 1, 0)$_{52}$	7	Did Not Converge
M412	ARIMA(1, 0, 3)(2, 1, 0)$_{52}$	7	7770.843
M413	ARIMA(3, 0, 2)(2, 1, 0)$_{52}$	8	7773.102
M414	ARIMA(2, 0, 3)(2, 1, 0)$_{52}$	8	7763.488
M415	ARIMA(3, 0, 3)(2, 1, 0)$_{52}$	9	Did Not Converge
M500	ARIMA(0, 0, 0)(0, 1, 2)$_{52}$	3	8625.560
M501	ARIMA(1, 0, 0)(0, 1, 2)$_{52}$	4	7768.477
M502	ARIMA(0, 0, 1)(0, 1, 2)$_{52}$	4	8266.953

M503	ARIMA(1, 0, 1)(0, 1, 2)$_{52}$	5	7765.876
M504	ARIMA(2, 0, 0)(0, 1, 2)$_{52}$	5	7765.302
M505	ARIMA(0, 0, 2)(0, 1, 2)$_{52}$	5	8082.029
M506	ARIMA(2, 0, 1)(0, 1, 2)$_{52}$	6	7765.858
M507	ARIMA(1, 0, 2)(0, 1, 2)$_{52}$	6	7765.894
M508	ARIMA(2, 0, 2)(0, 1, 2)$_{52}$	7	7763.507
M509	ARIMA(3, 0, 0)(0, 1, 2)$_{52}$	6	7765.701
M510	ARIMA(0, 0, 3)(0, 1, 2)$_{52}$	6	7969.441
M511	ARIMA(3, 0, 1)(0, 1, 2)$_{52}$	7	7767.603
M512	ARIMA(1, 0, 3)(0, 1, 2)$_{52}$	7	7767.312
M513	**ARIMA(3, 0, 2)(0, 1, 2)$_{52}$**	**8**	**7753.536**
M514	ARIMA(2, 0, 3)(0, 1, 2)$_{52}$	8	7753.720
M515	ARIMA(3, 0, 3)(0, 1, 2)$_{52}$	9	7755.209
M600	ARIMA(0, 0, 0)(2, 1, 1)$_{52}$	4	8570.239
M601	ARIMA(1, 0, 0)(2, 1, 1)$_{52}$	5	7768.834
M602	ARIMA(0, 0, 1)(2, 1, 1)$_{52}$	5	8245.514
M603	ARIMA(1, 0, 1)(2, 1, 1)$_{52}$	6	7766.317
M604	ARIMA(2, 0, 0)(2, 1, 1)$_{52}$	6	Did Not Converge
M605	ARIMA(0, 0, 2)(2, 1, 1)$_{52}$	6	8066.560
M606	ARIMA(2, 0, 1)(2, 1, 1)$_{52}$	7	Did Not Converge
M607	ARIMA(1, 0, 2)(2, 1, 1)$_{52}$	7	7766.405
M608	ARIMA(2, 0, 2)(2, 1, 1)$_{52}$	8	Did Not Converge
M609	ARIMA(3, 0, 0)(2, 1, 1)$_{52}$	7	Did Not Converge
M610	ARIMA(0, 0, 3)(2, 1, 1)$_{52}$	7	7976.575
M611	ARIMA(3, 0, 1)(2, 1, 1)$_{52}$	8	Did Not Converge

M612	ARIMA(1, 0, 3)(2, 1, 1)$_{52}$	8	7868.078
M613	ARIMA(3, 0, 2)(2, 1, 1)$_{52}$	9	7771.360
M614	ARIMA(2, 0, 3)(2, 1, 1)$_{52}$	9	Did Not Converge
M615	ARIMA(3, 0, 3)(2, 1, 1)$_{52}$	10	7773.885
M700	ARIMA(0, 0, 0)(1, 1, 2)$_{52}$	4	8556.570
M701	ARIMA(1, 0, 0)(1, 1, 2)$_{52}$	5	7769.871
M702	ARIMA(0, 0, 1)(1, 1, 2)$_{52}$	5	8217.642
M703	ARIMA(1, 0, 1)(1, 1, 2)$_{52}$	6	7767.186
M704	ARIMA(2, 0, 0)(1, 1, 2)$_{52}$	6	7766.587
M705	ARIMA(0, 0, 2)(1, 1, 2)$_{52}$	6	8049.606
M706	ARIMA(2, 0, 1)(1, 1, 2)$_{52}$	7	7767.139
M707	ARIMA(1, 0, 2)(1, 1, 2)$_{52}$	7	7767.111
M708	ARIMA(2, 0, 2)(1, 1, 2)$_{52}$	8	7768.820
M709	ARIMA(3, 0, 0)(1, 1, 2)$_{52}$	7	7769.947
M710	ARIMA(0, 0, 3)(1, 1, 2)$_{52}$	7	7950.357
M711	ARIMA(3, 0, 1)(1, 1, 2)$_{52}$	8	7768.878
M712	ARIMA(1, 0, 3)(1, 1, 2)$_{52}$	8	7768.580
M713	ARIMA(3, 0, 2)(1, 1, 2)$_{52}$	9	7755.255
M714	ARIMA(2, 0, 3)(1, 1, 2)$_{52}$	9	7755.506
M715	ARIMA(3, 0, 3)(1, 1, 2)$_{52}$	10	7757.917
M800	ARIMA(0, 0, 0)(2, 1, 2)$_{52}$	5	8536.601
M801	ARIMA(1, 0, 0)(2, 1, 2)$_{52}$	6	7769.004
M802	ARIMA(0, 0, 1)(2, 1, 2)$_{52}$	6	8214.657
M803	ARIMA(1, 0, 1)(2, 1, 2)$_{52}$	7	7765.665
M804	ARIMA(2, 0, 0)(2, 1, 2)$_{52}$	7	Did Not Converge

M805	ARIMA(0, 0, 2)(2, 1, 2)$_{52}$	7	8048.237
M806	ARIMA(2, 0, 1)(2, 1, 2)$_{52}$	8	Did Not Converge
M807	ARIMA(1, 0, 2)(2, 1, 2)$_{52}$	8	7766.157
M808	ARIMA(2, 0, 2)(2, 1, 2)$_{52}$	9	Did Not Converge
M809	ARIMA(3, 0, 0)(2, 1, 2)$_{52}$	8	Did Not Converge
M810	ARIMA(0, 0, 3)(2, 1, 2)$_{52}$	8	7950.768
M811	ARIMA(3, 0, 1)(2, 1, 2)$_{52}$	9	Did Not Converge
M812	ARIMA(1, 0, 3)(2, 1, 2)$_{52}$	9	7767.784
M813	ARIMA(3, 0, 2)(2, 1, 2)$_{52}$	10	7771.438
M814	ARIMA(2, 0, 3)(2, 1, 2)$_{52}$	10	Did Not Converge
M815	ARIMA(3, 0, 3)(2, 1, 2)$_{52}$	11	7774.244

Table A2: Model Parameters for Selected Model M513

Component	Point Estimate	Standard Error
AR1	–0.8976	0.0762
AR2	0.8908	0.0263
AR3	0.8405	0.0653
MA1	1.8074	0.0980
MA2	0.8188	0.0963
SMA1	–0.7180	0.0499
SMA2	0.1034	0.0533
$\sigma^2 = 495982$ Log Likelihood = –3868.77 AIC = 7753.536		

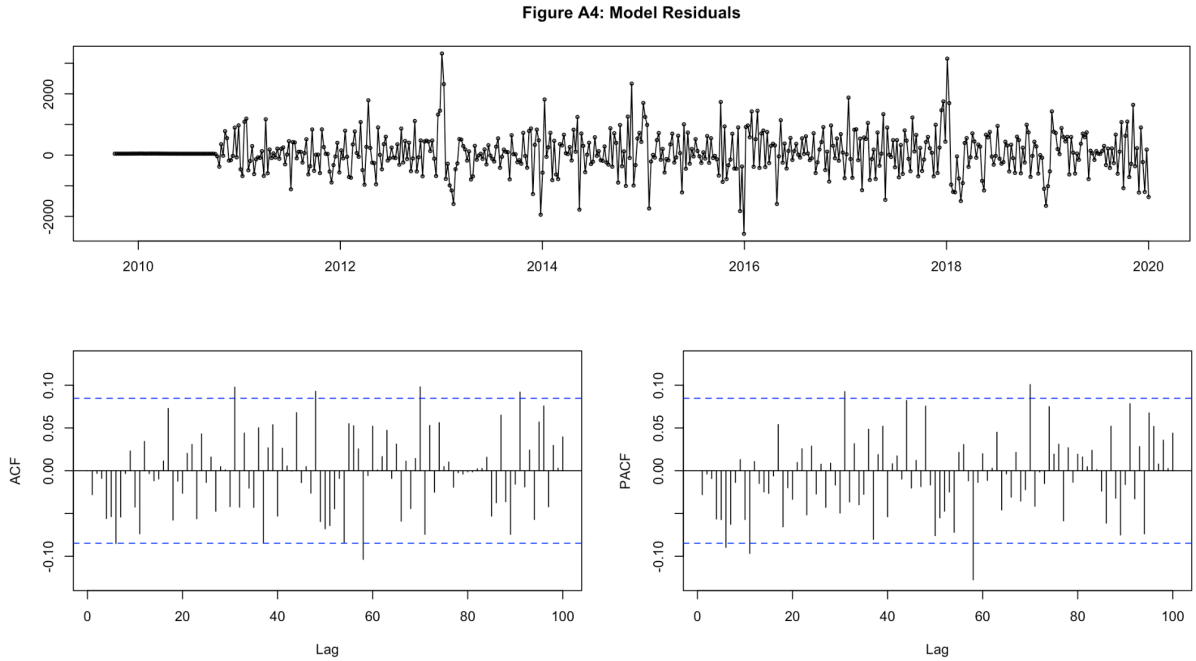


Figure A4: For the selected model M513 using the training data, plots of model residuals (top), model autocorrelation (bottom left), and model partial autocorrelation (bottom right). For the autocorrelation and partial autocorrelation plots, the lags are measured in weeks. Since the model used seasonal differencing, the first 52 weeks of the dataset have no values, which explains the flat line of model residuals from October 2009 to October 2010. There are a handful of observations with large residuals (in absolute values). The autocorrelation and partial autocorrelation plots look fairly good with only a few observations extending outside the blue horizontal dashed lines. There do not appear to be any major violations.

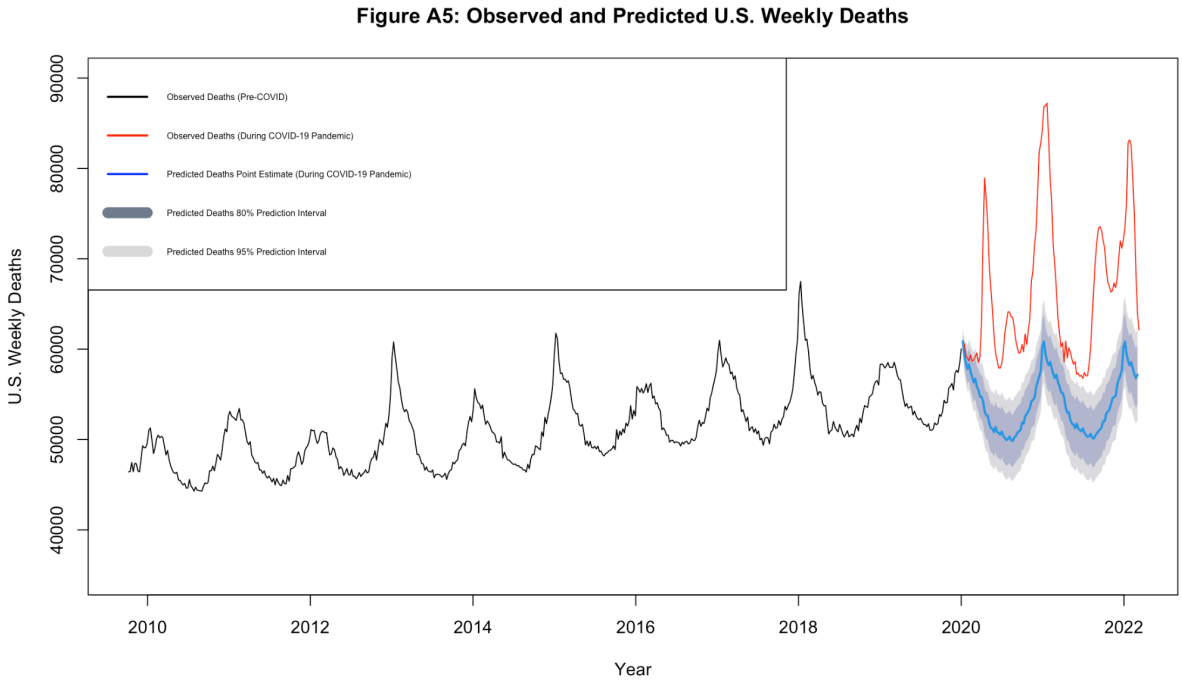


Figure A5: Observed weekly all-cause deaths before the pandemic (i.e. in the training dataset) (black curve), observed weekly all-cause deaths during the pandemic (i.e. in the testing dataset) (red curve), and forecast weekly all-cause deaths during the pandemic (point estimates) using the model developed from the training dataset (blue curve). Note the dark gray shading surrounding the blue curve represents the 80% prediction interval, and the light gray shading represents the 95% prediction interval.

Final Project (BS 845)

Jason Rutberg

2023-12-15

Import Packages

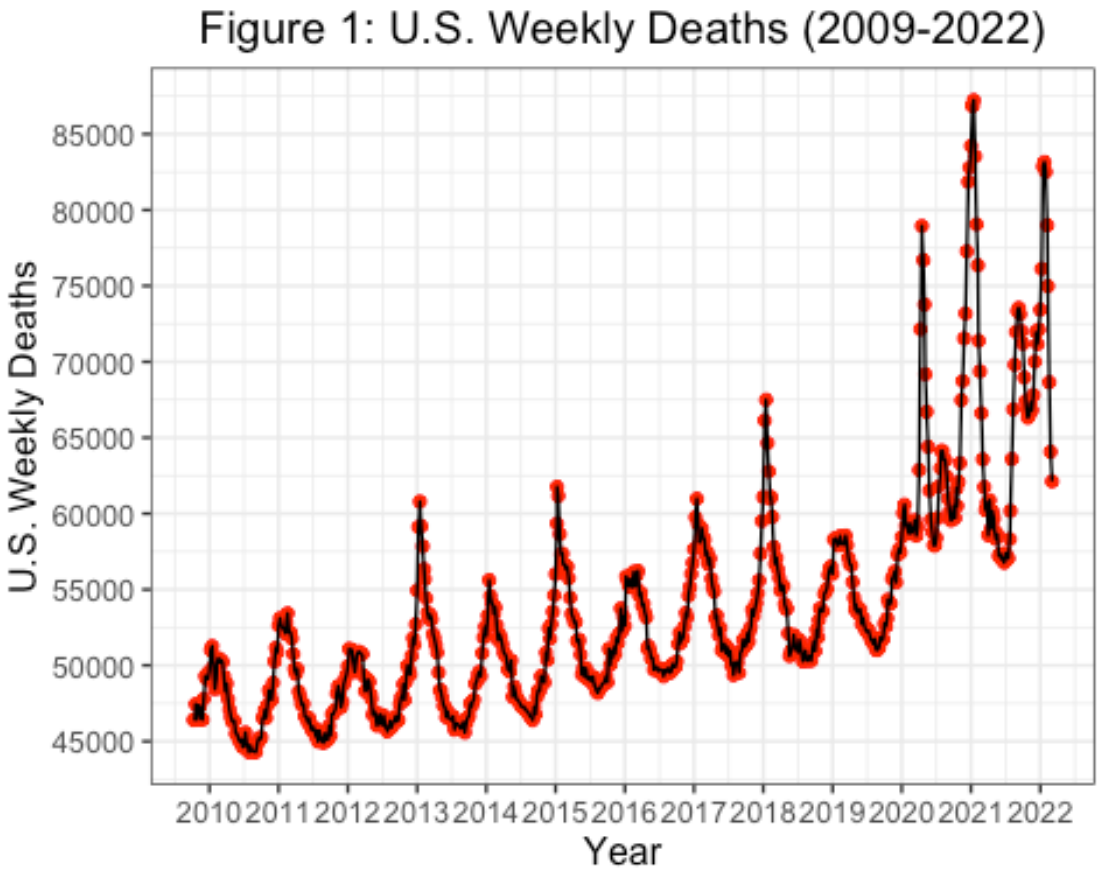
```
library(ggplot2)
library(forecast)

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

library(urca)
```

Import Data and Split into Training and Testing Data Sets

```
data <- read.csv("US_Weekly_Deaths.csv", as.is=T)
training_data <- as.ts(data[1:535,]) # Before COVID deaths
training_data <- ts(training_data, start = 2009.7665982, frequency = 52.17857
)
testing_data <- as.ts(data[536:dim(data)[1],]) # Week 536 has first COVID dea
th
testing_data <- ts(testing_data, start = 2020.03832991, frequency = 52.17857)
```

Figure 1

```
# Plot of US Weekly Deaths (2009-2022)
ggplot(data, aes(x = Date, y = All.Deaths)) +
  geom_point(col = "red") +
  geom_line() +
  labs(x = "Year",
       y = "U.S. Weekly Deaths",
       title = "Figure 1: U.S. Weekly Deaths (2009-2022)") +
  scale_x_continuous(breaks = seq(2009, 2022, by = 1)) +
  scale_y_continuous(breaks = seq(45000, 85000, by = 5000)) +
  theme_bw() +
  theme(text = element_text(size = 12),
        plot.title = element_text(hjust = 0.5))
```




```
# Time Series for All Deaths
Deaths <- training_data[,"All.Deaths"]
Year <- training_data[,"Year"]
Week <- training_data[,"Week"]
```

Summary Statistics

```
summary(Deaths) # Summary of Weekly Deaths Pre-COVID (Training Dataset)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   44281   47912   50749   51070   53468   67495

summary(testing_data[,"All.Deaths"]) # Summary of Weekly Deaths During COVID
(Testing Dataset)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   56764   59570   63576   66336   71969   87233
```

Figure A1

```
# Autocorrelation
acf(Deaths, lag = 160, main = 'Figure A1: Autocorrelation') # acf
```

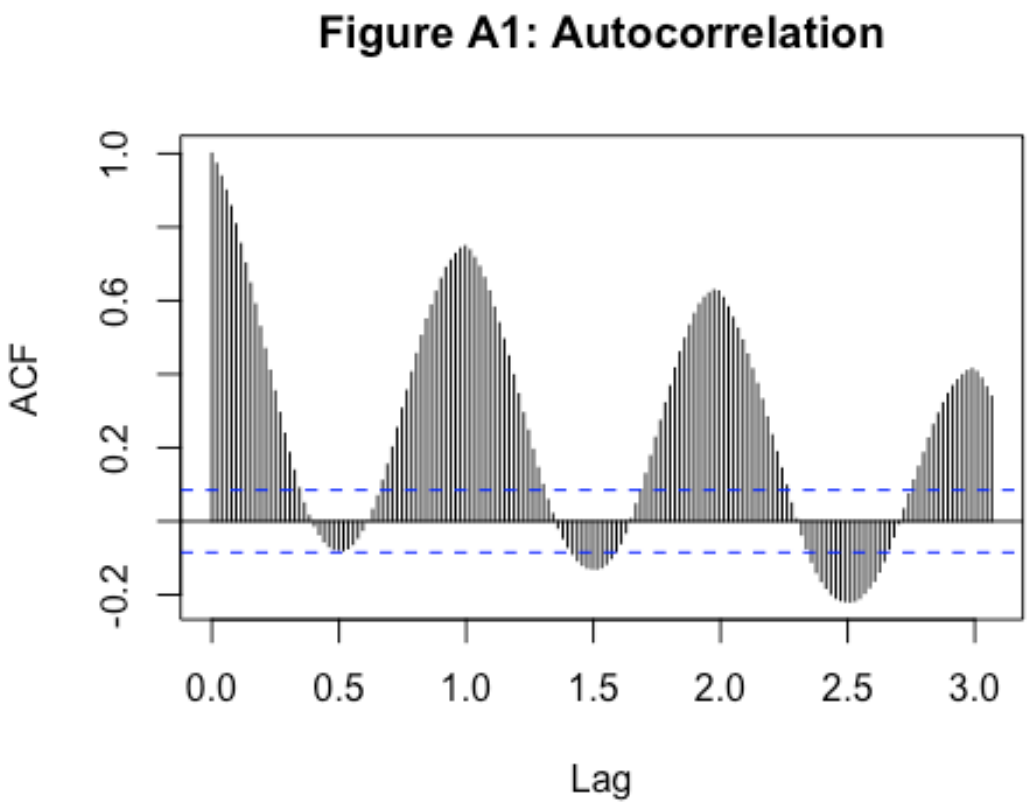
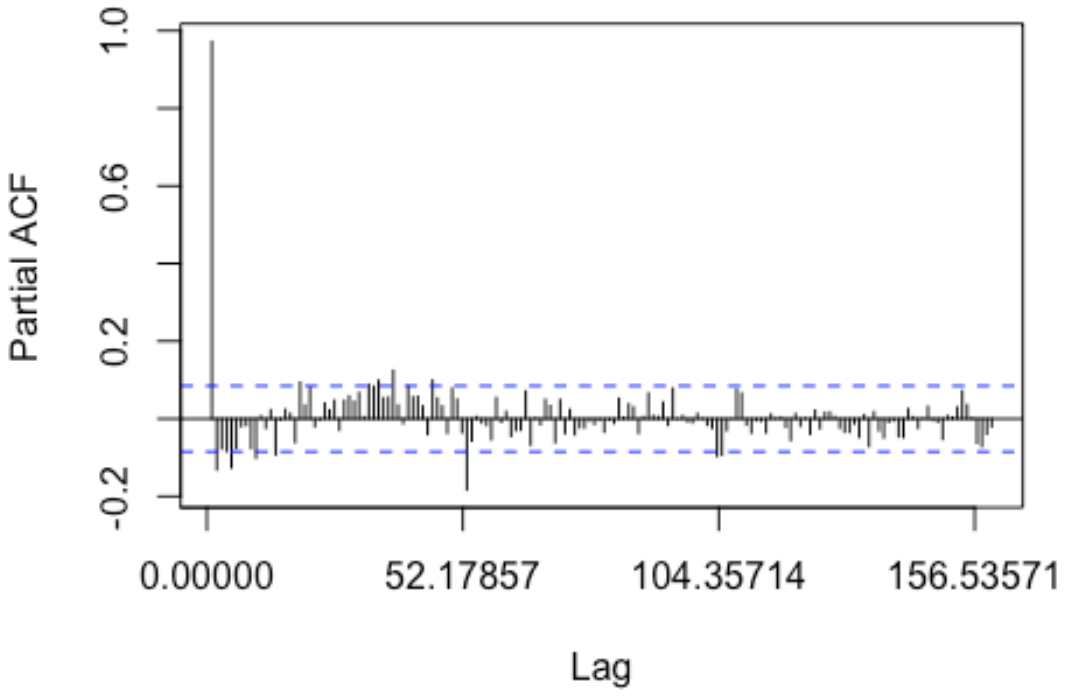


Figure A2

```
# Partial Autocorrelation
Pacf(Deaths, lag = 160, main = 'Figure A2: Partial Autocorrelation')
```

Figure A2: Partial Autocorrelation



Test for Stationarity Using Original Data from Training Set

```
# Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test for Stationarity
summary(ur.kpss(Deaths)) # reject null (NOT stationary)
```

```
##
## ########################
## # KPSS Unit Root Test #
## ########################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 3.3102
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

Take Seasonal Difference of Weekly Deaths for Training Data Set

```
SD_Deaths <- diff(Deaths, lag = 52, differences = 1)
```

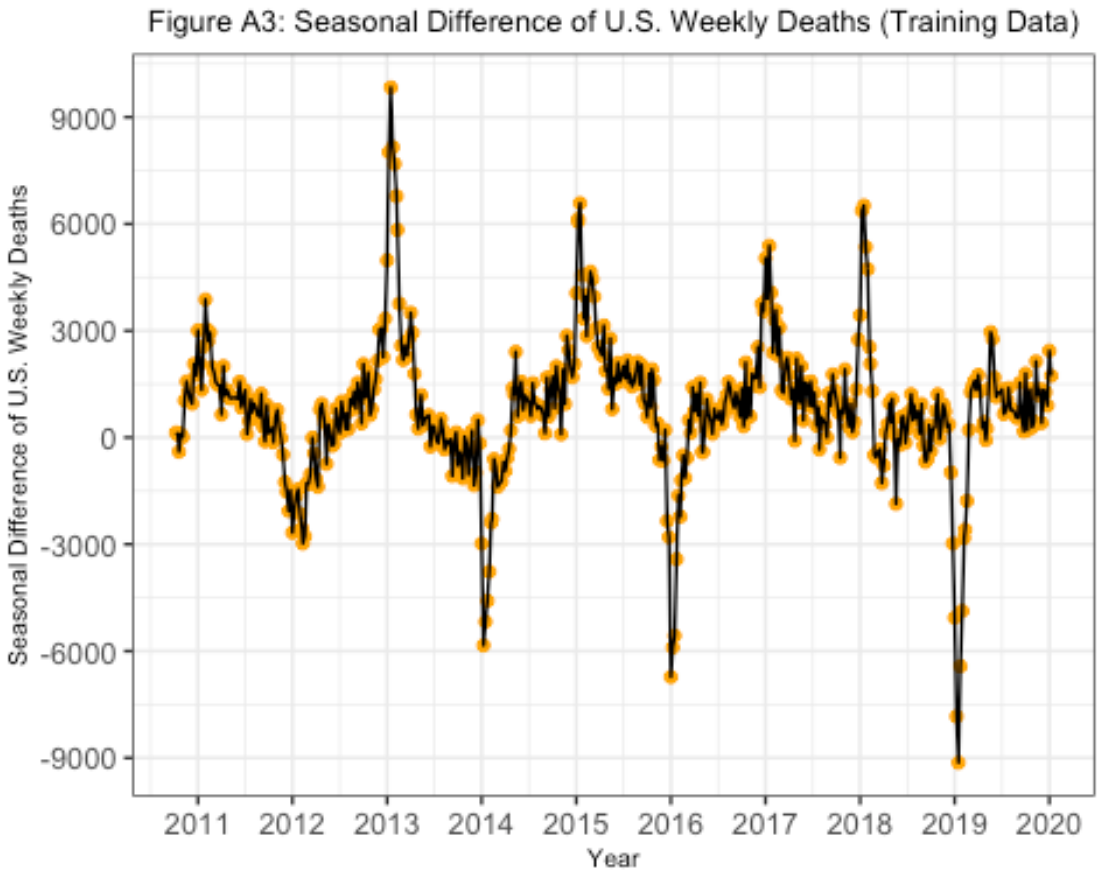
Test for Stationarity Using Seasonally-Differenced Data from Training Set

```
summary(ur.kpss(SD_Deaths)) # stationary
```

```
##
## ########################
## # KPSS Unit Root Test #
## ########################
##
## Test is of type: mu with 5 lags.
##
## Value of test-statistic is: 0.1032
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

Figure A3

```
# Seasonal Difference of U.S. Weekly Deaths
SD_deaths_data <- data.frame(Year = training_data[53:length(Deaths),"Date"],
SD_Deaths)

ggplot(SD_deaths_data, aes(x = Year, y = SD_Deaths)) +
  geom_point(col = "orange") +
  geom_line() +
  labs(x = "Year",
       y = "Seasonal Difference of U.S. Weekly Deaths",
       title = "Figure A3: Seasonal Difference of U.S. Weekly Deaths (Trainin
g Data)") +
  scale_x_continuous(breaks = seq(2009, 2020, by = 1)) +
  scale_y_continuous(breaks = seq(-15000, 15000, by = 3000)) +
  theme_bw() +
  theme(text = element_text(size = 12),
        plot.title = element_text(hjust = 0.5),
        title = element_text(size = 8))
```



Model Selection

```
# Seasonal Component (0, 1, 0)
M000 <- arima(Deaths, order = c(0,0,0), seasonal = list(order = c(0,1,0), per
iod = 52))
M001 <- arima(Deaths, order = c(1,0,0), seasonal = list(order = c(0,1,0), per
iod = 52))
M002 <- arima(Deaths, order = c(0,0,1), seasonal = list(order = c(0,1,0), per
iod = 52))
M003 <- arima(Deaths, order = c(1,0,1), seasonal = list(order = c(0,1,0), per
iod = 52))
M004 <- arima(Deaths, order = c(2,0,0), seasonal = list(order = c(0,1,0), per
iod = 52))
M005 <- arima(Deaths, order = c(0,0,2), seasonal = list(order = c(0,1,0), per
iod = 52))
M006 <- arima(Deaths, order = c(2,0,1), seasonal = list(order = c(0,1,0), per
iod = 52))
M007 <- arima(Deaths, order = c(1,0,2), seasonal = list(order = c(0,1,0), per
iod = 52))
M008 <- arima(Deaths, order = c(2,0,2), seasonal = list(order = c(0,1,0), per
iod = 52))
M009 <- arima(Deaths, order = c(3,0,0), seasonal = list(order = c(0,1,0), per
```

```
iod = 52))
M010 <- arima(Deaths, order = c(0,0,3), seasonal = list(order = c(0,1,0), per
iod = 52))
M011 <- arima(Deaths, order = c(3,0,1), seasonal = list(order = c(0,1,0), per
iod = 52))
M012 <- arima(Deaths, order = c(1,0,3), seasonal = list(order = c(0,1,0), per
iod = 52))
M013 <- arima(Deaths, order = c(3,0,2), seasonal = list(order = c(0,1,0), per
iod = 52))
M014 <- arima(Deaths, order = c(2,0,3), seasonal = list(order = c(0,1,0), per
iod = 52))
M015 <- arima(Deaths, order = c(3,0,3), seasonal = list(order = c(0,1,0), per
iod = 52))

# Seasonal Component (1, 1, 0)
M100 <- arima(Deaths, order = c(0,0,0), seasonal = list(order = c(1,1,0), per
iod = 52))
M101 <- arima(Deaths, order = c(1,0,0), seasonal = list(order = c(1,1,0), per
iod = 52))
M102 <- arima(Deaths, order = c(0,0,1), seasonal = list(order = c(1,1,0), per
iod = 52))
M103 <- arima(Deaths, order = c(1,0,1), seasonal = list(order = c(1,1,0), per
iod = 52))
M104 <- arima(Deaths, order = c(2,0,0), seasonal = list(order = c(1,1,0), per
iod = 52))
M105 <- arima(Deaths, order = c(0,0,2), seasonal = list(order = c(1,1,0), per
iod = 52))
M106 <- arima(Deaths, order = c(2,0,1), seasonal = list(order = c(1,1,0), per
iod = 52))
M107 <- arima(Deaths, order = c(1,0,2), seasonal = list(order = c(1,1,0), per
iod = 52))
M108 <- arima(Deaths, order = c(2,0,2), seasonal = list(order = c(1,1,0), per
iod = 52))
M109 <- arima(Deaths, order = c(3,0,0), seasonal = list(order = c(1,1,0), per
iod = 52))
M110 <- arima(Deaths, order = c(0,0,3), seasonal = list(order = c(1,1,0), per
iod = 52))
M111 <- arima(Deaths, order = c(3,0,1), seasonal = list(order = c(1,1,0), per
iod = 52))
M112 <- arima(Deaths, order = c(1,0,3), seasonal = list(order = c(1,1,0), per
iod = 52))
M113 <- arima(Deaths, order = c(3,0,2), seasonal = list(order = c(1,1,0), per
iod = 52))
M114 <- arima(Deaths, order = c(2,0,3), seasonal = list(order = c(1,1,0), per
iod = 52))
M115 <- arima(Deaths, order = c(3,0,3), seasonal = list(order = c(1,1,0), per
iod = 52))

# Seasonal Component (0, 1, 1)
M200 <- arima(Deaths, order = c(0,0,0), seasonal = list(order = c(0,1,1), per
```

```
iod = 52))
M201 <- arima(Deaths, order = c(1,0,0), seasonal = list(order = c(0,1,1), per
iod = 52))
M202 <- arima(Deaths, order = c(0,0,1), seasonal = list(order = c(0,1,1), per
iod = 52))
M203 <- arima(Deaths, order = c(1,0,1), seasonal = list(order = c(0,1,1), per
iod = 52))
M204 <- arima(Deaths, order = c(2,0,0), seasonal = list(order = c(0,1,1), per
iod = 52))
M205 <- arima(Deaths, order = c(0,0,2), seasonal = list(order = c(0,1,1), per
iod = 52))
M206 <- arima(Deaths, order = c(2,0,1), seasonal = list(order = c(0,1,1), per
iod = 52))
M207 <- arima(Deaths, order = c(1,0,2), seasonal = list(order = c(0,1,1), per
iod = 52))
M208 <- arima(Deaths, order = c(2,0,2), seasonal = list(order = c(0,1,1), per
iod = 52))
M209 <- arima(Deaths, order = c(3,0,0), seasonal = list(order = c(0,1,1), per
iod = 52))
M210 <- arima(Deaths, order = c(0,0,3), seasonal = list(order = c(0,1,1), per
iod = 52))
M211 <- arima(Deaths, order = c(3,0,1), seasonal = list(order = c(0,1,1), per
iod = 52))
M212 <- arima(Deaths, order = c(1,0,3), seasonal = list(order = c(0,1,1), per
iod = 52))
M213 <- arima(Deaths, order = c(3,0,2), seasonal = list(order = c(0,1,1), per
iod = 52))
M214 <- arima(Deaths, order = c(2,0,3), seasonal = list(order = c(0,1,1), per
iod = 52))
M215 <- arima(Deaths, order = c(3,0,3), seasonal = list(order = c(0,1,1), per
iod = 52))

# Seasonal Component (1, 1, 1)
M300 <- arima(Deaths, order = c(0,0,0), seasonal = list(order = c(1,1,1), per
iod = 52))
M301 <- arima(Deaths, order = c(1,0,0), seasonal = list(order = c(1,1,1), per
iod = 52))
M302 <- arima(Deaths, order = c(0,0,1), seasonal = list(order = c(1,1,1), per
iod = 52))
M303 <- arima(Deaths, order = c(1,0,1), seasonal = list(order = c(1,1,1), per
iod = 52))
M304 <- arima(Deaths, order = c(2,0,0), seasonal = list(order = c(1,1,1), per
iod = 52))
M305 <- arima(Deaths, order = c(0,0,2), seasonal = list(order = c(1,1,1), per
iod = 52))
M306 <- arima(Deaths, order = c(2,0,1), seasonal = list(order = c(1,1,1), per
iod = 52))
M307 <- arima(Deaths, order = c(1,0,2), seasonal = list(order = c(1,1,1), per
iod = 52))
M308 <- arima(Deaths, order = c(2,0,2), seasonal = list(order = c(1,1,1), per
```

```
iod = 52))
M309 <- arima(Deaths, order = c(3,0,0), seasonal = list(order = c(1,1,1), per
iod = 52))
M310 <- arima(Deaths, order = c(0,0,3), seasonal = list(order = c(1,1,1), per
iod = 52))
M311 <- arima(Deaths, order = c(3,0,1), seasonal = list(order = c(1,1,1), per
iod = 52))
M312 <- arima(Deaths, order = c(1,0,3), seasonal = list(order = c(1,1,1), per
iod = 52))
M313 <- arima(Deaths, order = c(3,0,2), seasonal = list(order = c(1,1,1), per
iod = 52))
M314 <- arima(Deaths, order = c(2,0,3), seasonal = list(order = c(1,1,1), per
iod = 52))
M315 <- arima(Deaths, order = c(3,0,3), seasonal = list(order = c(1,1,1), per
iod = 52))

# Seasonal Component (2, 1, 0)
M400 <- arima(Deaths, order = c(0,0,0), seasonal = list(order = c(2,1,0), per
iod = 52))
M401 <- arima(Deaths, order = c(1,0,0), seasonal = list(order = c(2,1,0), per
iod = 52))
M402 <- arima(Deaths, order = c(0,0,1), seasonal = list(order = c(2,1,0), per
iod = 52))
M403 <- arima(Deaths, order = c(1,0,1), seasonal = list(order = c(2,1,0), per
iod = 52))
M404 <- arima(Deaths, order = c(2,0,0), seasonal = list(order = c(2,1,0), per
iod = 52))
M405 <- arima(Deaths, order = c(0,0,2), seasonal = list(order = c(2,1,0), per
iod = 52))
M406 <- arima(Deaths, order = c(2,0,1), seasonal = list(order = c(2,1,0), per
iod = 52))
M407 <- arima(Deaths, order = c(1,0,2), seasonal = list(order = c(2,1,0), per
iod = 52))
M408 <- arima(Deaths, order = c(2,0,2), seasonal = list(order = c(2,1,0), per
iod = 52))
M409 <- arima(Deaths, order = c(3,0,0), seasonal = list(order = c(2,1,0), per
iod = 52))
M410 <- arima(Deaths, order = c(0,0,3), seasonal = list(order = c(2,1,0), per
iod = 52))
M411 <- arima(Deaths, order = c(3,0,1), seasonal = list(order = c(2,1,0), per
iod = 52))
M412 <- arima(Deaths, order = c(1,0,3), seasonal = list(order = c(2,1,0), per
iod = 52))
M413 <- arima(Deaths, order = c(3,0,2), seasonal = list(order = c(2,1,0), per
iod = 52))
M414 <- arima(Deaths, order = c(2,0,3), seasonal = list(order = c(2,1,0), per
iod = 52))
M415 <- arima(Deaths, order = c(3,0,3), seasonal = list(order = c(2,1,0), per
iod = 52))
```

```r
# Seasonal Component (0, 1, 2)
M500 <- arima(Deaths, order = c(0,0,0), seasonal = list(order = c(0,1,2), per
iod = 52))
M501 <- arima(Deaths, order = c(1,0,0), seasonal = list(order = c(0,1,2), per
iod = 52))
M502 <- arima(Deaths, order = c(0,0,1), seasonal = list(order = c(0,1,2), per
iod = 52))
M503 <- arima(Deaths, order = c(1,0,1), seasonal = list(order = c(0,1,2), per
iod = 52))
M504 <- arima(Deaths, order = c(2,0,0), seasonal = list(order = c(0,1,2), per
iod = 52))
M505 <- arima(Deaths, order = c(0,0,2), seasonal = list(order = c(0,1,2), per
iod = 52))
M506 <- arima(Deaths, order = c(2,0,1), seasonal = list(order = c(0,1,2), per
iod = 52))
M507 <- arima(Deaths, order = c(1,0,2), seasonal = list(order = c(0,1,2), per
iod = 52))
M508 <- arima(Deaths, order = c(2,0,2), seasonal = list(order = c(0,1,2), per
iod = 52))
M509 <- arima(Deaths, order = c(3,0,0), seasonal = list(order = c(0,1,2), per
iod = 52))
M510 <- arima(Deaths, order = c(0,0,3), seasonal = list(order = c(0,1,2), per
iod = 52))
M511 <- arima(Deaths, order = c(3,0,1), seasonal = list(order = c(0,1,2), per
iod = 52))
M512 <- arima(Deaths, order = c(1,0,3), seasonal = list(order = c(0,1,2), per
iod = 52))
M513 <- arima(Deaths, order = c(3,0,2), seasonal = list(order = c(0,1,2), per
iod = 52))
M514 <- arima(Deaths, order = c(2,0,3), seasonal = list(order = c(0,1,2), per
iod = 52))
M515 <- arima(Deaths, order = c(3,0,3), seasonal = list(order = c(0,1,2), per
iod = 52))

# Seasonal Component (2, 1, 1)
M600 <- arima(Deaths, order = c(0,0,0), seasonal = list(order = c(2,1,1), per
iod = 52))
M601 <- arima(Deaths, order = c(1,0,0), seasonal = list(order = c(2,1,1), per
iod = 52))
M602 <- arima(Deaths, order = c(0,0,1), seasonal = list(order = c(2,1,1), per
iod = 52))
M603 <- arima(Deaths, order = c(1,0,1), seasonal = list(order = c(2,1,1), per
iod = 52))
M604 <- arima(Deaths, order = c(2,0,0), seasonal = list(order = c(2,1,1), per
iod = 52))
M605 <- arima(Deaths, order = c(0,0,2), seasonal = list(order = c(2,1,1), per
iod = 52))
M606 <- arima(Deaths, order = c(2,0,1), seasonal = list(order = c(2,1,1), per
iod = 52))
M607 <- arima(Deaths, order = c(1,0,2), seasonal = list(order = c(2,1,1), per
```

```r
iod = 52))
M608 <- arima(Deaths, order = c(2,0,2), seasonal = list(order = c(2,1,1), per
iod = 52))
M609 <- arima(Deaths, order = c(3,0,0), seasonal = list(order = c(2,1,1), per
iod = 52))
M610 <- arima(Deaths, order = c(0,0,3), seasonal = list(order = c(2,1,1), per
iod = 52))
M611 <- arima(Deaths, order = c(3,0,1), seasonal = list(order = c(2,1,1), per
iod = 52))
M612 <- arima(Deaths, order = c(1,0,3), seasonal = list(order = c(2,1,1), per
iod = 52))
M613 <- arima(Deaths, order = c(3,0,2), seasonal = list(order = c(2,1,1), per
iod = 52))
M614 <- arima(Deaths, order = c(2,0,3), seasonal = list(order = c(2,1,1), per
iod = 52))
M615 <- arima(Deaths, order = c(3,0,3), seasonal = list(order = c(2,1,1), per
iod = 52))

# Seasonal Component (1, 1, 2)
M700 <- arima(Deaths, order = c(0,0,0), seasonal = list(order = c(1,1,2), per
iod = 52))
M701 <- arima(Deaths, order = c(1,0,0), seasonal = list(order = c(1,1,2), per
iod = 52))
M702 <- arima(Deaths, order = c(0,0,1), seasonal = list(order = c(1,1,2), per
iod = 52))
M703 <- arima(Deaths, order = c(1,0,1), seasonal = list(order = c(1,1,2), per
iod = 52))
M704 <- arima(Deaths, order = c(2,0,0), seasonal = list(order = c(1,1,2), per
iod = 52))
M705 <- arima(Deaths, order = c(0,0,2), seasonal = list(order = c(1,1,2), per
iod = 52))
M706 <- arima(Deaths, order = c(2,0,1), seasonal = list(order = c(1,1,2), per
iod = 52))
M707 <- arima(Deaths, order = c(1,0,2), seasonal = list(order = c(1,1,2), per
iod = 52))
M708 <- arima(Deaths, order = c(2,0,2), seasonal = list(order = c(1,1,2), per
iod = 52))
M709 <- arima(Deaths, order = c(3,0,0), seasonal = list(order = c(1,1,2), per
iod = 52))
M710 <- arima(Deaths, order = c(0,0,3), seasonal = list(order = c(1,1,2), per
iod = 52))
M711 <- arima(Deaths, order = c(3,0,1), seasonal = list(order = c(1,1,2), per
iod = 52))
M712 <- arima(Deaths, order = c(1,0,3), seasonal = list(order = c(1,1,2), per
iod = 52))
M713 <- arima(Deaths, order = c(3,0,2), seasonal = list(order = c(1,1,2), per
iod = 52))
M714 <- arima(Deaths, order = c(2,0,3), seasonal = list(order = c(1,1,2), per
iod = 52))
M715 <- arima(Deaths, order = c(3,0,3), seasonal = list(order = c(1,1,2), per
```

```
iod = 52))

# Seasonal Component (2, 1, 2)
M800 <- arima(Deaths, order = c(0,0,0), seasonal = list(order = c(2,1,2), per
iod = 52))
M801 <- arima(Deaths, order = c(1,0,0), seasonal = list(order = c(2,1,2), per
iod = 52))
M802 <- arima(Deaths, order = c(0,0,1), seasonal = list(order = c(2,1,2), per
iod = 52))
M803 <- arima(Deaths, order = c(1,0,1), seasonal = list(order = c(2,1,2), per
iod = 52))
M804 <- arima(Deaths, order = c(2,0,0), seasonal = list(order = c(2,1,2), per
iod = 52))
M805 <- arima(Deaths, order = c(0,0,2), seasonal = list(order = c(2,1,2), per
iod = 52))
M806 <- arima(Deaths, order = c(2,0,1), seasonal = list(order = c(2,1,2), per
iod = 52))
M807 <- arima(Deaths, order = c(1,0,2), seasonal = list(order = c(2,1,2), per
iod = 52))
M808 <- arima(Deaths, order = c(2,0,2), seasonal = list(order = c(2,1,2), per
iod = 52))
M809 <- arima(Deaths, order = c(3,0,0), seasonal = list(order = c(2,1,2), per
iod = 52))
M810 <- arima(Deaths, order = c(0,0,3), seasonal = list(order = c(2,1,2), per
iod = 52))
M811 <- arima(Deaths, order = c(3,0,1), seasonal = list(order = c(2,1,2), per
iod = 52))
M812 <- arima(Deaths, order = c(1,0,3), seasonal = list(order = c(2,1,2), per
iod = 52))
M813 <- arima(Deaths, order = c(3,0,2), seasonal = list(order = c(2,1,2), per
iod = 52))
M814 <- arima(Deaths, order = c(2,0,3), seasonal = list(order = c(2,1,2), per
iod = 52))
M815 <- arima(Deaths, order = c(3,0,3), seasonal = list(order = c(2,1,2), per
iod = 52))
```

Compare Model AIC's

```
# Compare Models (choose one with lowest AIC)
AIC(M000, M001, M002, M003, M004, M005, M006, M007,
    M008, M009, M010, M011, M012, M013, M014, M015,
    M100, M101, M102, M103, M104, M105, M106, M107,
    M108, M109, M110, M111, M112, M113, M114, M115,
    M200, M201, M202, M203, M204, M205, M206, M207,
    M208, M209, M210, M211, M212, M213, M214, M215,
    M300, M301, M302, M303, M304, M305, M306, M307,
    M308, M309, M310, M311, M312, M313, M314, M315,
    M400, M401, M402, M403, M404, M405, M406, M407,
    M408, M409, M410, M411, M412, M413, M414, M415,
```

```
    M500, M501, M502, M503, M504, M505, M506, M507,
    M508, M509, M510, M511, M512, M513, M514, M515,
    M600, M601, M602, M603, M604, M605, M606, M607,
    M608, M609, M610, M611, M612, M613, M614, M615,
    M700, M701, M702, M703, M704, M705, M706, M707,
    M708, M709, M710, M711, M712, M713, M714, M715,
    M800, M801, M802, M803, M804, M805, M806, M807,
    M808, M809, M810, M811, M812, M813, M814, M815)
```

Final Model: M513: ARIMA(3, 0, 2)(0, 1, 2) [52]

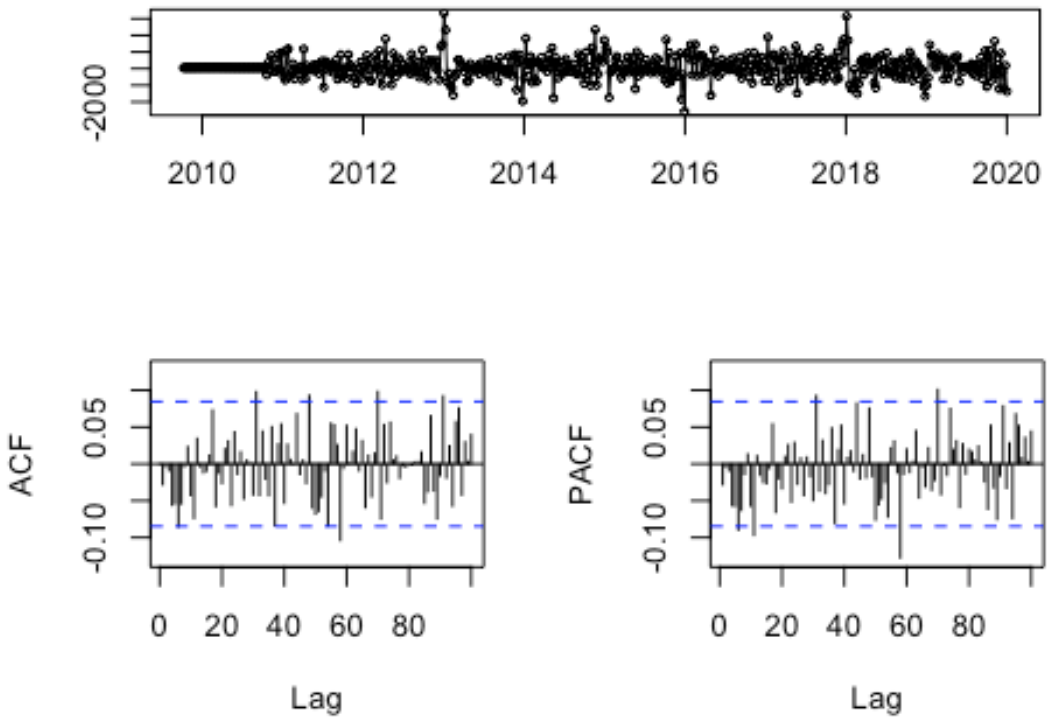
```
M513 <- arima(Deaths, order = c(3,0,2), seasonal = list(order = c(0,1,2), per
iod = 52))
summary(M513)
```

```
##
## Call:
## arima(x = Deaths, order = c(3, 0, 2), seasonal = list(order = c(0, 1, 2),
period = 52))
##
## Coefficients:
##           ar1     ar2     ar3     ma1     ma2     sma1    sma2
##       -0.8976  0.8908  0.8405  1.8074  0.8188  -0.7180  0.1034
## s.e.   0.0762  0.0263  0.0653  0.0980  0.0963   0.0499  0.0533
##
## sigma^2 estimated as 495982:  log likelihood = -3868.77,  aic = 7753.54
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE      MAPE     MASE
## Training set 80.57158 669.3204 477.5416 0.1457515 0.9107069 0.776708
##                      ACF1
## Training set -0.02794821
```

Figure A4

```
tsdisplay(residuals(M513), lag.max = 100, main='Figure A4: Model Residuals')
```

Figure A4: Model Residuals



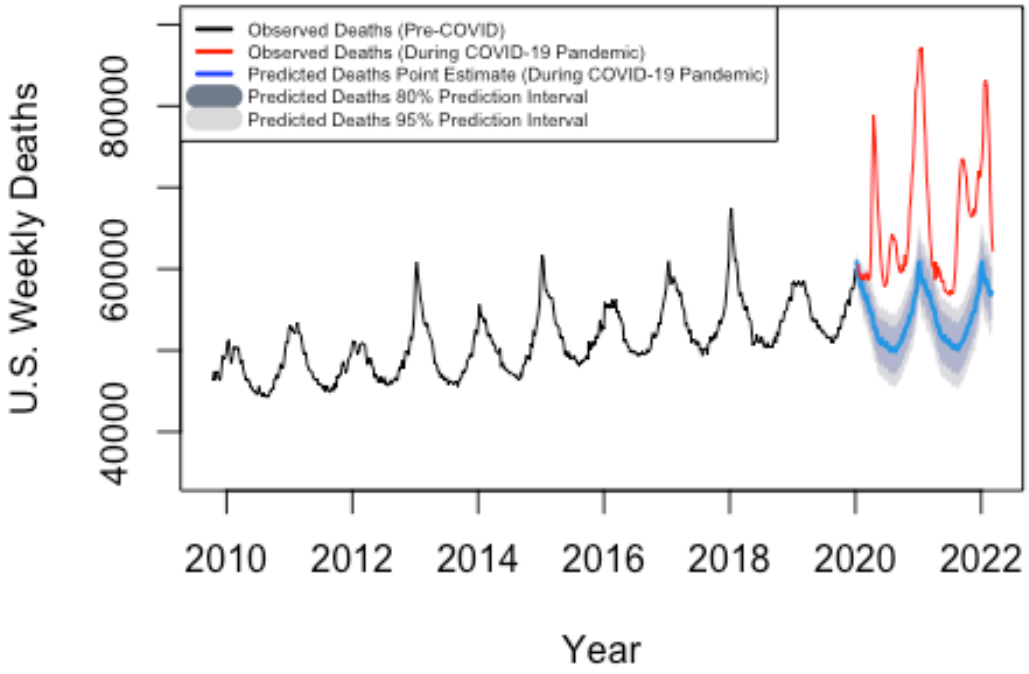
Model Forecasting

```
forecast <- forecast(M513, h = 113) # 113 weeks in the testing data set
```

Figure A5

```
plot(forecast, xlab = "Year",
     ylab = "U.S. Weekly Deaths",
     main = "Figure A5: Observed and Predicted U.S. Weekly Deaths",
     ylim = c(35000, 90000), cex.main = 0.85)
lines(testing_data[,"All.Deaths"], col = "red")
legend("topleft", legend = c("Observed Deaths (Pre-COVID)", "Observed Deaths
(During COVID-19 Pandemic)", "Predicted Deaths Point Estimate (During COVID-1
9 Pandemic)", "Predicted Deaths 80% Prediction Interval", "Predicted Deaths 9
5% Prediction Interval"),
       lty = 1, lwd = c(2, 2, 2, 10, 10), col = c("black", "red", "blue", "li
ghtsteelblue4", "gray85"), cex = 0.5)
```

Figure A5: Observed and Predicted U.S. Weekly Deaths



Compute Excess Deaths

```
point_estimate <- forecast$mean
upper_bound <- forecast$upper[,2]
excess_deaths <- as.numeric(testing_data[,"All.Deaths"]) - as.numeric(upper_b
ound)
excess_deaths_data <- data.frame(Year = testing_data[,"Date"], COVID_deaths =
testing_data[,"COVID.19.Deaths"], excess_deaths)
```

Total Excess Deaths

```
sum(excess_deaths) # total number of excess deaths

## [1] 883510.4
```

Total COVID-19 Deaths

```
sum(testing_data[,"COVID.19.Deaths"]) # total number of COVID deaths

## [1] 981196
```

Correlation Between Excess Deaths and COVID-19 Deaths

```
round(cor(excess_deaths, testing_data[,"COVID.19.Deaths"]), 3)

## [1] 0.951

cor.test(excess_deaths, testing_data[,"COVID.19.Deaths"])

##
##  Pearson's product-moment correlation
##
## data:  excess_deaths and testing_data[, "COVID.19.Deaths"]
## t = 32.442, df = 111, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9297232 0.9660933
## sample estimates:
##       cor
## 0.9511033
```

Figure 2

```
ggplot(excess_deaths_data, aes(x = Year, y = excess_deaths)) +
  geom_point(data = excess_deaths_data, aes(x = Year, y = excess_deaths), col
or = "black") +
  geom_point(data = excess_deaths_data, aes(x = Year, y = COVID_deaths), colo
r = "black") +
  geom_line(data = excess_deaths_data, aes(x = Year, y = excess_deaths, color
= 'Excess Deaths')) +
  geom_line(data = excess_deaths_data, aes(x = Year, y = COVID_deaths, color
= 'COVID-19 Deaths')) +
  labs(x = "Year",
       y = "U.S. Weekly Deaths",
       title = "Figure 2: U.S. Weekly Excess Deaths and COVID-19 Deaths (2020
-2022)") +
  scale_x_continuous(breaks = seq(2020, 2022.5, by = 0.5)) +
  scale_y_continuous(breaks = seq(0, 30000, by = 5000)) +
  theme_bw() +
  theme(text = element_text(size = 7),
        plot.title = element_text(hjust = 0.5)) +
  scale_color_manual(name = 'Legend',
                     breaks=c("Excess Deaths", "COVID-19 Deaths"),
                     values=c("Excess Deaths" = "blue", "COVID-19 Deaths" = "
red"))
```

Figure 2: U.S. Weekly Excess Deaths and COVID-19 Deaths (2020-2022)