# Forecasting COVID-19 Cases and Deaths Using Time Series Analysis in R

Regina Duval, Practicum for MSDS692

6/27/2020

## Introduction

---

Since January of 2020, coronavirus cases and deaths have been tracked around the world. The COVID-19 global outbreak continues to impact daily life and everyone is wondering when it will end. The purpose of this project is to investigate the efficacy of time series analysis in forecasting cases and deaths at the country, state, and county level. Specifically, I have focused on six countries, seven states within the US, and nine counties within Minnesota, my home state.

Epidemiologic models are inherently complex because of the high number of variables needed to predict disease spread. Model accuracy is also dependent on knowledge of the disease, something that is still very limited when it comes to COVID-19. Finally, as of June 25, 2020, there are only 155 days of data to use in this analysis. As with any model, less data means less precise forecasts.

Recognizing these constraints, my analysis was narrow in scope since it was intended to utilize basic time series models such as ARIMA and Vector Autoregression. I initially planned to include constants at the county level like population, distance from the state population center, and presence of meat-packing plants. As I moved more deeply into the project, it became clear that these constants were difficult to incorporate into time series models with such a short span of data. Instead, I used them to run multiple linear regression in order to guide me in determining which counties might impact the case and death count of other counties regionally.

## Data Preparation and Exploration

---

### The Data

I began with a data set made available for the COVID19 Global Forecasting (Week 5) Kaggle competition found here. Unfortunately, this data only contained population, cases and deaths through early May (it has since been updated to include data through early June). To expand my data set, I found a site here (USAfacts) that daily updates confirmed cases and deaths for every county in the United States, but it tracks cases cumulatively. In addition to case and death count, I gathered county distance data from the National Bureau of Economic Research here and the United States Cities Database here. This data, combined with the county population data from my Kaggle data set, allowed me to identify each state's highest populated county and the distance of each other county in the state from that population center. Finally, I scoured the internet for lists of meat-packing plants in the US. Using lists found here, here, and here, I marked key counties with the presence of meat packing plants.

Ultimately, I used four files to conduct my time series analysis. These files can be found in the data folder of my Github repository at https://github.com/Reinalynn/MSDS692. The kaggle and county_data data sets are in long form while cases0624 and deaths0624 data are in wide form. This will be resolved later when the recent data is used to check the accuracy of my models and predict future cases and deaths for select models.

```r
library(astsa)
library(broom)
library(caret)
library(DAAG)
library(dplyr)
library(dynlm)
library(forecast)
library(fpp2)
library(funModeling)
library(knitr)
library(MARSS)
library(MTS)
library(quantmod)
library(readr)
library(reshape)
library(sf)
library(tidyverse)
library(tidycensus)
library(tmap)
library(tmaptools)
library(tseries)
library(urbnmapr)
library(urca)
library(varhandle)
library(vars)

# load files from github
kaggle <- read.csv(url("https://raw.githubusercontent.com/Reinalynn/MSDS692/master/Data/kaggle.csv"), he
county_data <- read.csv(url("https://raw.githubusercontent.com/Reinalynn/MSDS692/master/Data/County_data
cases0624 <- read.csv(url("https://raw.githubusercontent.com/Reinalynn/MSDS692/master/Data/cases0624.csv
deaths0624 <- read.csv(url("https://raw.githubusercontent.com/Reinalynn/MSDS692/master/Data/deaths0624.c

kable(head(kaggle))
```

**Install libraries and load data**

| County | Province_State | Country_Region | Population | Date | Cases | Deaths |
|---|---|---|---|---|---|---|
| | | Afghanistan | 27657145 | 2020-01-23 | 0 | 0 |
| | | Afghanistan | 27657145 | 2020-01-24 | 0 | 0 |
| | | Afghanistan | 27657145 | 2020-01-25 | 0 | 0 |
| | | Afghanistan | 27657145 | 2020-01-26 | 0 | 0 |
| | | Afghanistan | 27657145 | 2020-01-27 | 0 | 0 |
| | | Afghanistan | 27657145 | 2020-01-28 | 0 | 0 |

```r
#kable(head(county_data))
#kable(cases0624[1:5, c(1, 2, 3, 4, 5, 159)])
#kable(deaths0624[1:5, c(1, 2, 3, 4, 5, 159)])
```

## EDA

I wanted to better understand which counties had a high number of cases or deaths, so I first merged my county specific data with the total of cases and deaths by county as of June 24, 2020. All of my data exploration was conducted on US county level data because this was the lowest level data.

As a note, a meat_plant value of 0 means that no meat plant is present in the county and a value of 1 means there is a meat plant. For dist_cat, a value of 1 = 0 miles from population center, 2 = 0.1 to 25 miles, 3 = 25.1 to 75 miles, 4 = 75.1 to 150 miles, 5 = 150.1 to 300 miles, 6 = greater than 300 miles.

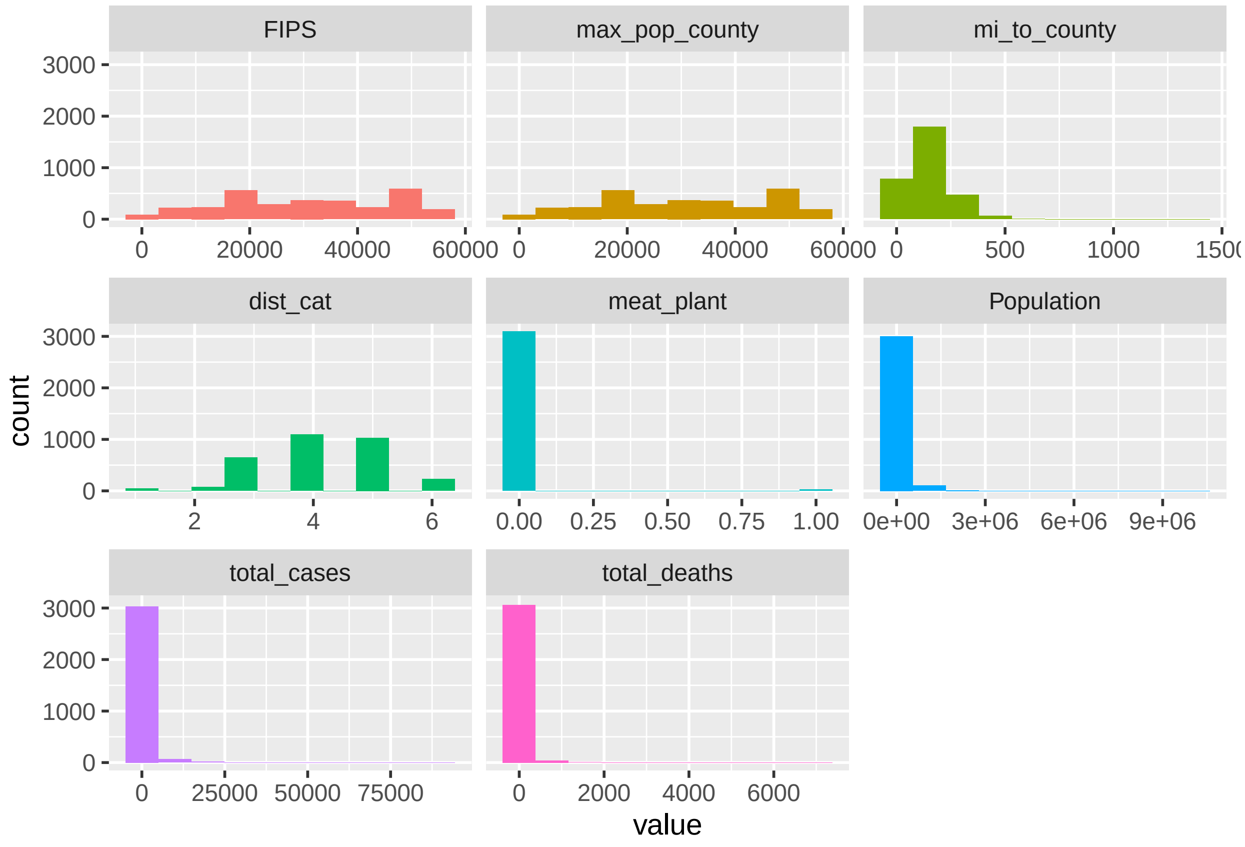
```
total_cases <- cases0624[, c(1, 159)] # first column is FIPS, last column is cumulative cases as of 062
total_deaths <- deaths0624[, c(1, 159)] # repeat for deaths
total <- merge(total_cases, total_deaths, by = "FIPS") # create new data frame with 3 columns
colnames(total) <- c("FIPS", "total_cases", "total_deaths") # name columns
data <- left_join(county_data, total, by = "FIPS") # add total cases and deaths to county data
summary(data)
```

### Create EDA data set

```
##       FIPS        countyname           state            concatenate
##  Min.   : 1001   Length:3124        Length:3124        Length:3124
##  1st Qu.:19006   Class :character   Class :character   Class :character
##  Median :29186   Mode  :character   Mode  :character   Mode  :character
##  Mean   :30478
##  3rd Qu.:45088
##  Max.   :56045
##
##  max_pop_county  max_pop_countyname  mi_to_county        dist_cat
##  Min.   : 1073   Length:3124        Min.   :   0.00   Min.   :1.000
##  1st Qu.:19153   Class :character   1st Qu.:  75.76   1st Qu.:4.000
##  Median :29510   Mode  :character   Median : 127.19   Median :4.000
##  Mean   :30505                      Mean   : 146.32   Mean   :4.177
##  3rd Qu.:45019                      3rd Qu.: 196.02   3rd Qu.:5.000
##  Max.   :56021                      Max.   :1368.11   Max.   :6.000
##
##    meat_plant         Population        total_cases       total_deaths
##  Min.   :0.000000   Min.   :      86   Min.   :    0.0   Min.   :   0.00
##  1st Qu.:0.000000   1st Qu.:   10948   1st Qu.:   15.0   1st Qu.:   0.00
##  Median :0.000000   Median :   25812   Median :   66.0   Median :   1.00
##  Mean   :0.008643   Mean   :  105786   Mean   :  740.5   Mean   :  38.06
##  3rd Qu.:0.000000   3rd Qu.:   68104   3rd Qu.:  287.0   3rd Qu.:   9.00
##  Max.   :1.000000   Max.   :10039107   Max.   :89490.0   Max.   :6995.00
##                                        NA's   :1         NA's   :1
```

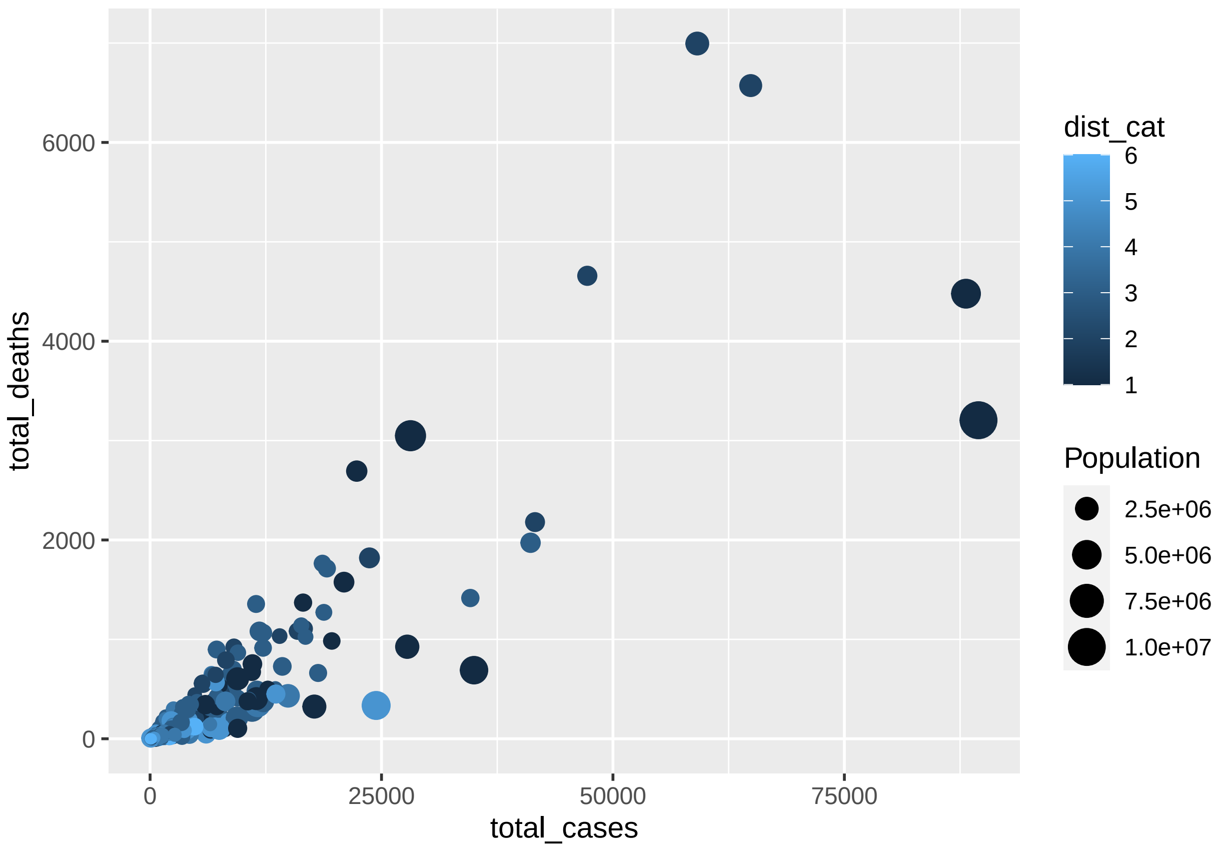
Basic EDA plots show histograms for each numeric variable in the data. There is a clear relationship between cases and deaths, as expected, and counties further from population centers do appear to have fewer cases and deaths. The relationship between population and cases and deaths is not as clear.

```
plot_num(data)
```



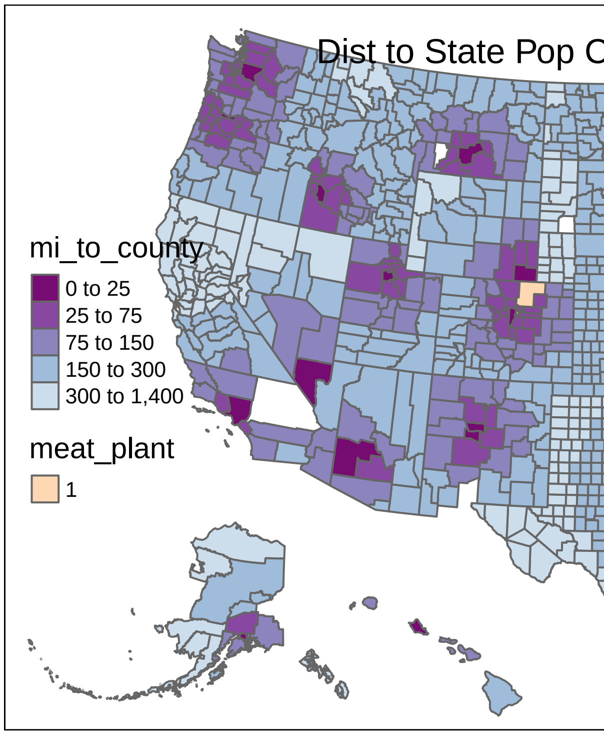
**EDA plots**

```
ggplot(data, aes(x = total_cases, y = total_deaths, color = dist_cat, size = Population)) +
  geom_point()
```

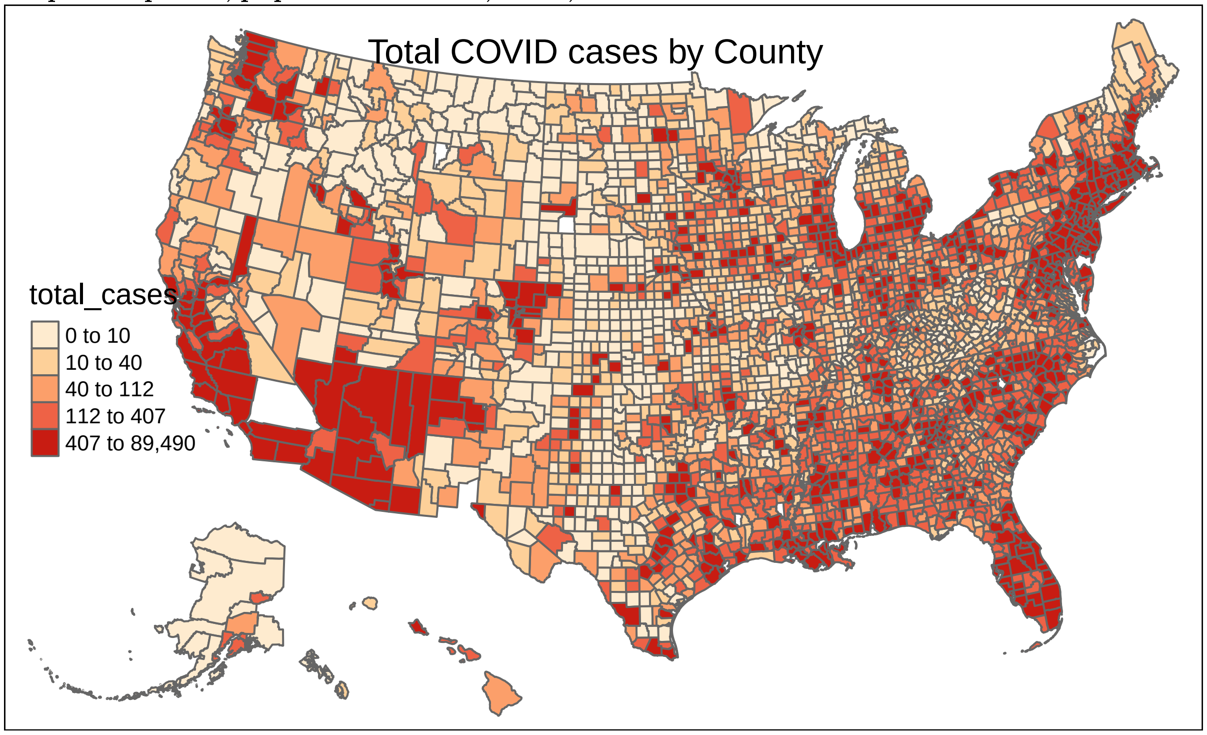


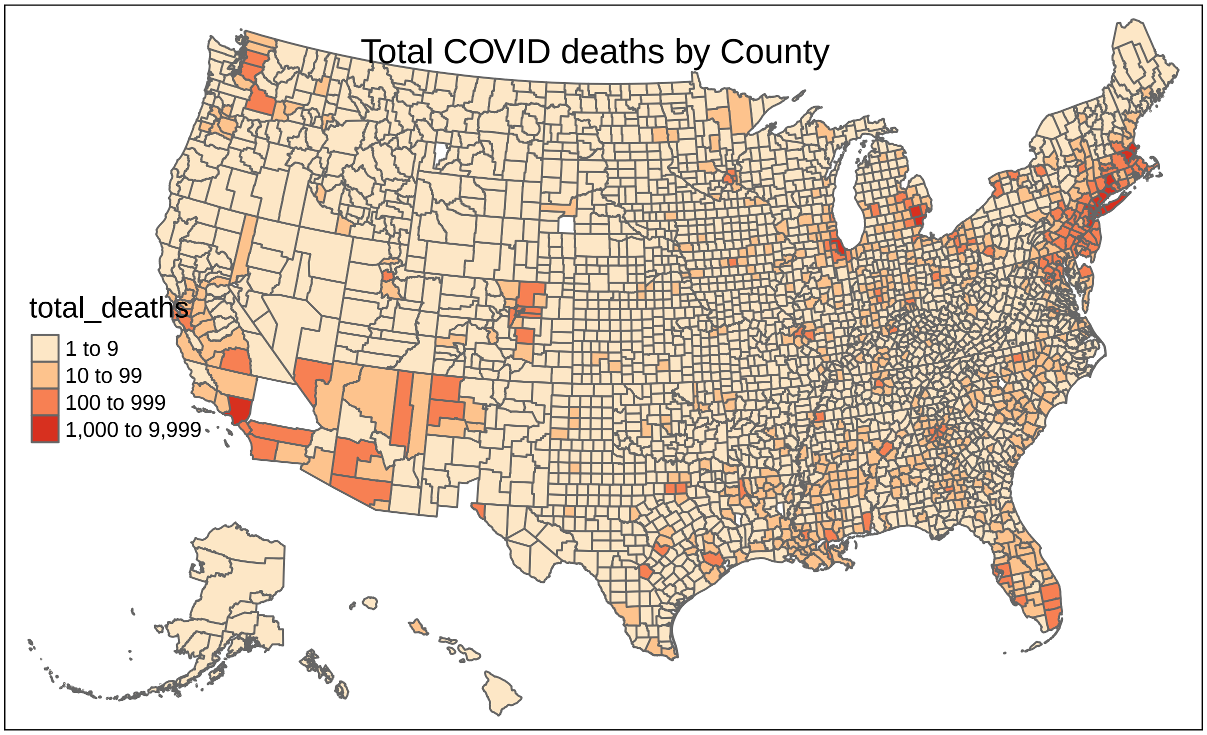
Since my data pertains to geographic locations, I decided to also view my variables using two of the mapping packages in R. First, I used urbnmapr's county level data set to add a geometry column to my data. Then, I used the tmap package to map my spatial data. I found this tutorial especially helpful when exploring tmap.

The maps easily displayed where the population center was in each state and where meat plants were located. They also gave a clear picture of total cases across the US, showing concentrations along the coasts, the southwest and the southeast. Most of the red areas appear to be around cities, as expected. The death count follows the same pattern but is much lower nationwide.

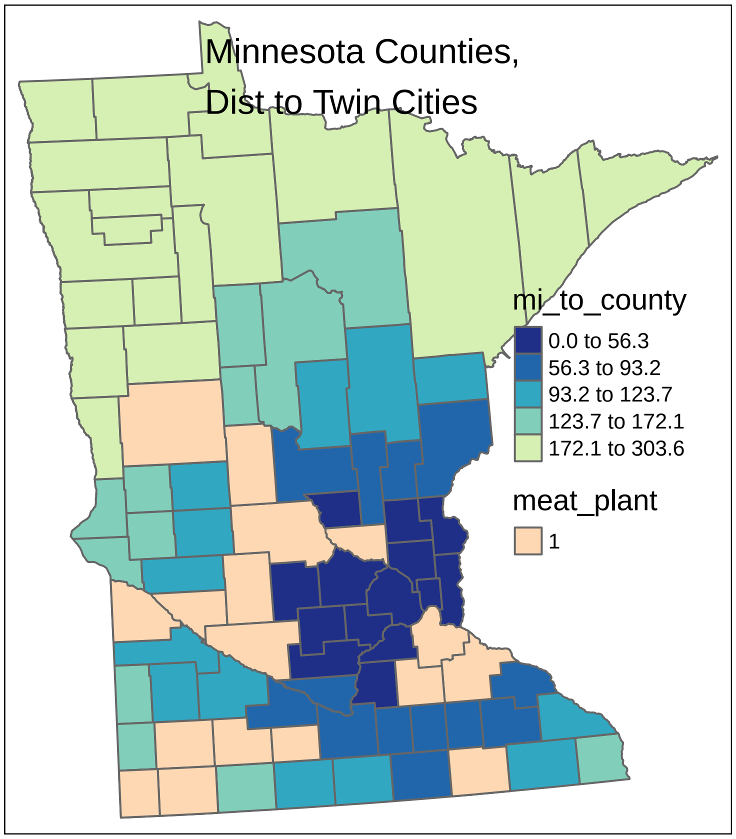


**Map meat plants, population centers, cases, and deaths for US**

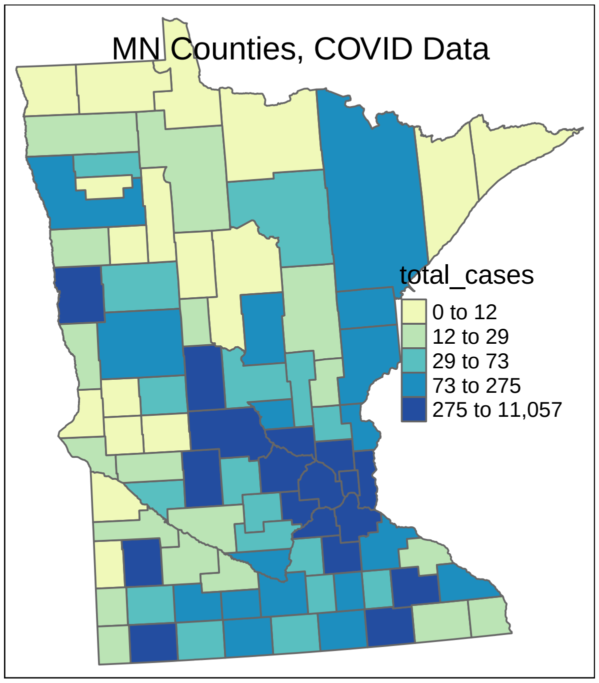


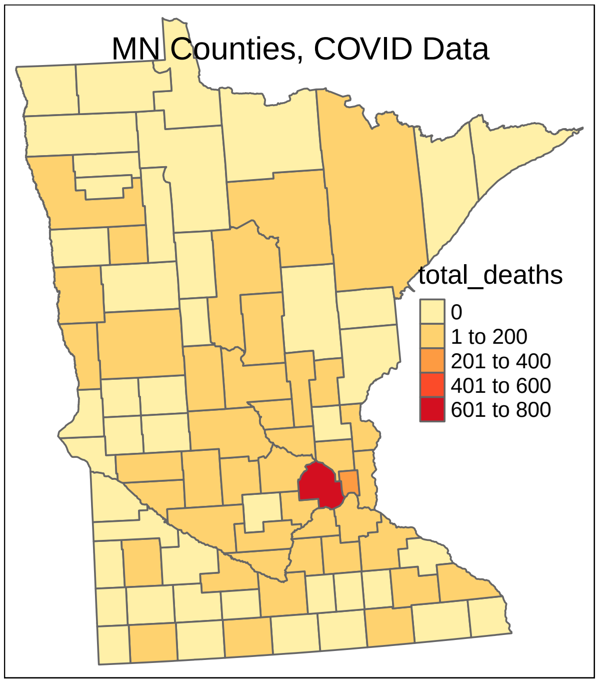


I also mapped the counties of Minnesota in order to view counties with the presence of meat packing plants. As can be seen from the map, these counties do have a higher concentration of COVID cases than their surrounding counties although this variable does not appear to have an impact on COVID deaths in the state.



**Map same information for MN counties**





## Models and Results

---

### Building the Univariate Models

Exploratory data analysis seemed to show that constants like population, distance from a population center, or presence of a meat plant might have an impact on COVID cases or deaths. To look at this more closely, I built a multiple linear regression model for the counties in Minnesota. To do this, I had to make some adjustments to my data in order to change the wide form data sets cases0624 and deaths0624 into long form so that each county and each date had its own line. An added benefit of linear regression (rather than time series) is that this type of long form data means I have 9657 observations (87 counties times 111 days of records). In regular time series analysis, I would be restricted to 111 observations.

The linear models for both cases and deaths were quite poor with relatively low adjusted R-squared values (cases = 0.4367, deaths = 0.4512). I did use k-fold cross validation to attempt to improve the models but there was no appreciable difference. Of course, adding deaths to the cases model both increased its accuracy and made it harder to use it to forecast cases since deaths are impacted by cases, so I settled for the mediocre models that did not include the highly correlated variable (deaths for cases model, cases for deaths model). Ultimately, these models convinced me that my constants were not enough on their own to accurately predict cases or deaths.

```
# predict cases using linear regression
mod1 <- lm(cases ~ dist_cat + meat_plant + Population + date, data = long_data)
summary(mod1)$adj.r.squared # mediocre adjusted R-squared (0.4367), all variables are significant excep
```

### Multiple Linear Regression for MN data

```
## [1] 0.4366753
```

```
selectedMod <- step(mod1) # no improvement to AIC to exclude any variable
```

```
## Start:  AIC=119892
## cases ~ dist_cat + meat_plant + Population + date
##
##               Df  Sum of Sq        RSS    AIC
## <none>                      2324723809 119892
## - dist_cat     1    2669550 2327393358 119901
## - meat_plant   1   11183082 2335906890 119936
## - date       110  178425096 2503148905 120386
## - Population   1 1230499321 3555223130 123992
```

```
# predict deaths using linear regression
mod1d <- lm(deaths ~ dist_cat + meat_plant + Population + date, data = long_data)
summary(mod1d)$adj.r.squared # slightly higher adj R-squared (0.4512), all variables are significant ex
```

```
## [1] 0.4511967
```

### Choosing the Best Univariate Time Series Model

NOTE: There are many excellent resources online regarding time series analysis in R, but I found the following to be most useful:
* A Little Book of R for Time Series
* Forecasting: Principles and Practice
* DataCamp: Time Series Analysis in R
* This handy flowchart

The bulk of my project focused on time series analysis and forecasting. Beginning with the full US data, I started with univariate time series analysis and focused on finding the best model to predict just one variable (cases or deaths). Using the autocorrelation function, I investigated trend and seasonality for both US cases and deaths. Neither time series showed seasonality, but that is probably because the time series is still too short. I believe we will see seasonality further down the road. I was able to determine that both cases and deaths need to be differenced at least once to become stationary.

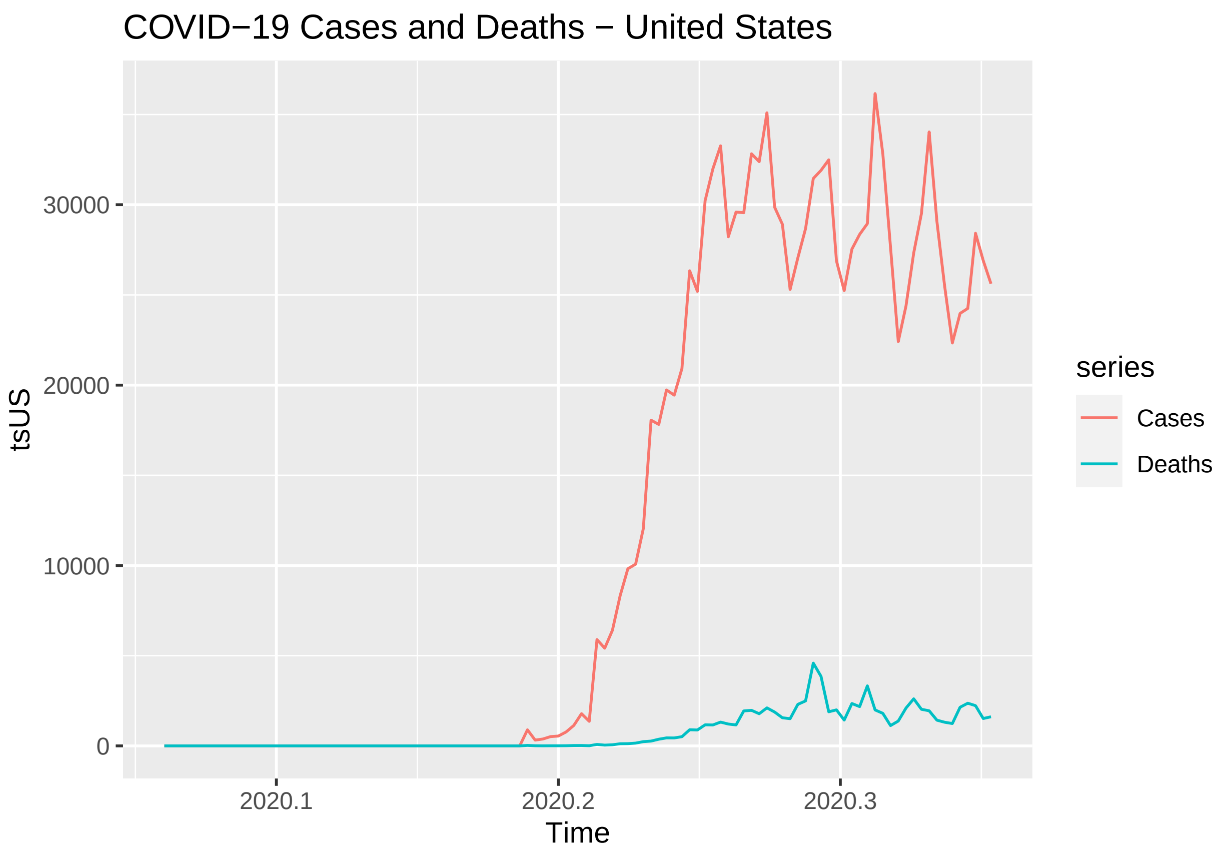
After building a simple exponential smoothing model and several different holt trend models (differenced, damped, differenced and damped), I used the information criteria (AIC, AICc, BIC) to determine that ARIMA models provided the best fit for the COVID data. Once I made the decision to use the auto.arima function, it was relatively easy to apply the same basic code to multiple countries, states, and counties. Keep in mind that all of these ARIMA models were built using the Kaggle data which only extended to May 9, 2020. I also used the xreg argument for ARIMA models that allows you to add an external regressor (so I could model US cases by referring back to US deaths), but this model actually performed worse than the basic ARIMA model and it was more difficult to use for forecasting.

The auto.arima function identified the best ARIMA model within certain parameters, but I noticed that in many cases, the residuals of the models were still significant (p value below 0.05). This meant that the model did not completely explain the trend or movement in the data and something was left behind in the residuals. When the checkresiduals call provided a p-value less than 0.05, I manually manipulated the ARIMA p (number of time lags), d (degree of differencing), or q (order of the moving average model) until I could find a better model. In the fit_casesUS model below, the auto.arima function recommended an ARIMA(3, 1, 0) model but I found that ARIMA(6, 1, 1) was much better.

### ARIMA models for US cases and deaths

| | County | Province_State | Country_Region | Population | Date | Cases | Deaths |
|---|---|---|---|---|---|---|---|
| 103 | | | US | 324141489 | 2020-05-04 | 22335 | 1240 |
| 104 | | | US | 324141489 | 2020-05-05 | 23976 | 2142 |
| 105 | | | US | 324141489 | 2020-05-06 | 24251 | 2367 |
| 106 | | | US | 324141489 | 2020-05-07 | 28420 | 2231 |
| 107 | | | US | 324141489 | 2020-05-08 | 26906 | 1518 |
| 108 | | | US | 324141489 | 2020-05-09 | 25620 | 1615 |

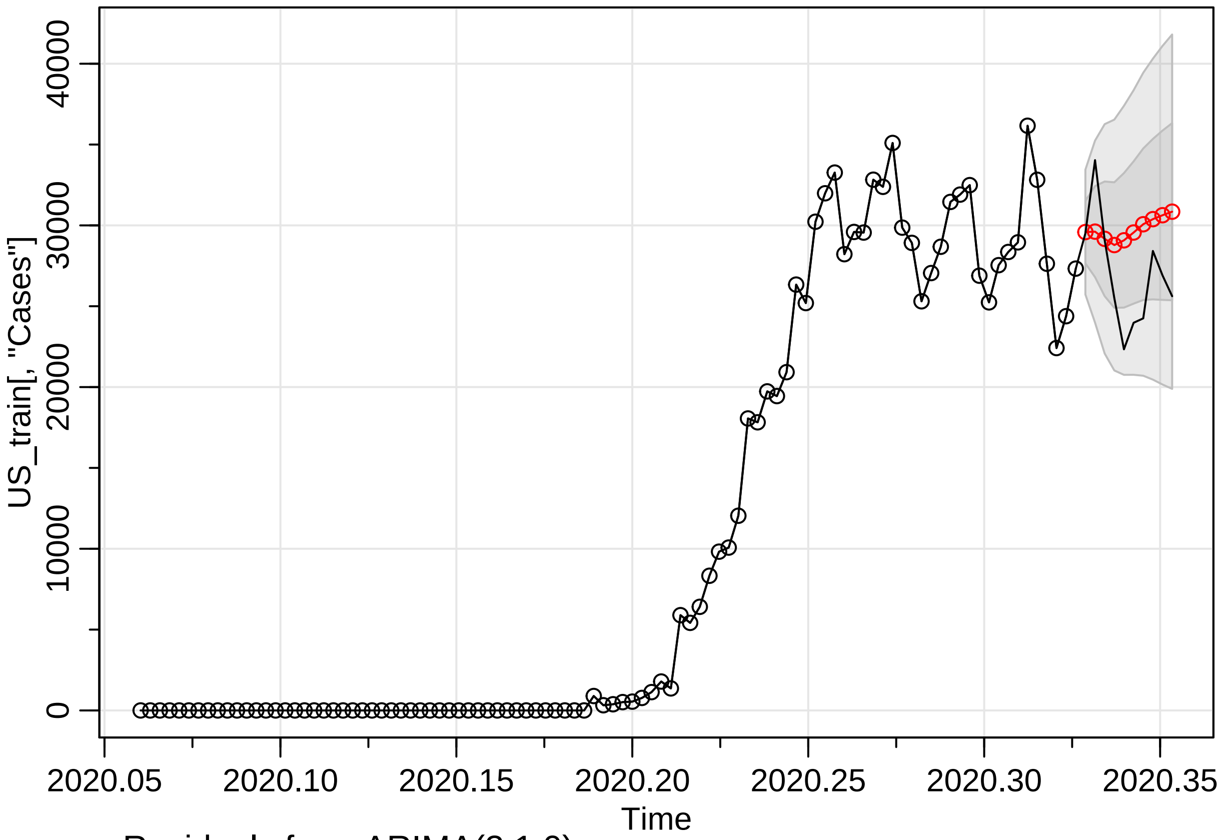
```
##  Time-Series [1:108, 1:2] from 2020 to 2020: 0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:2] "Cases" "Deaths"
```

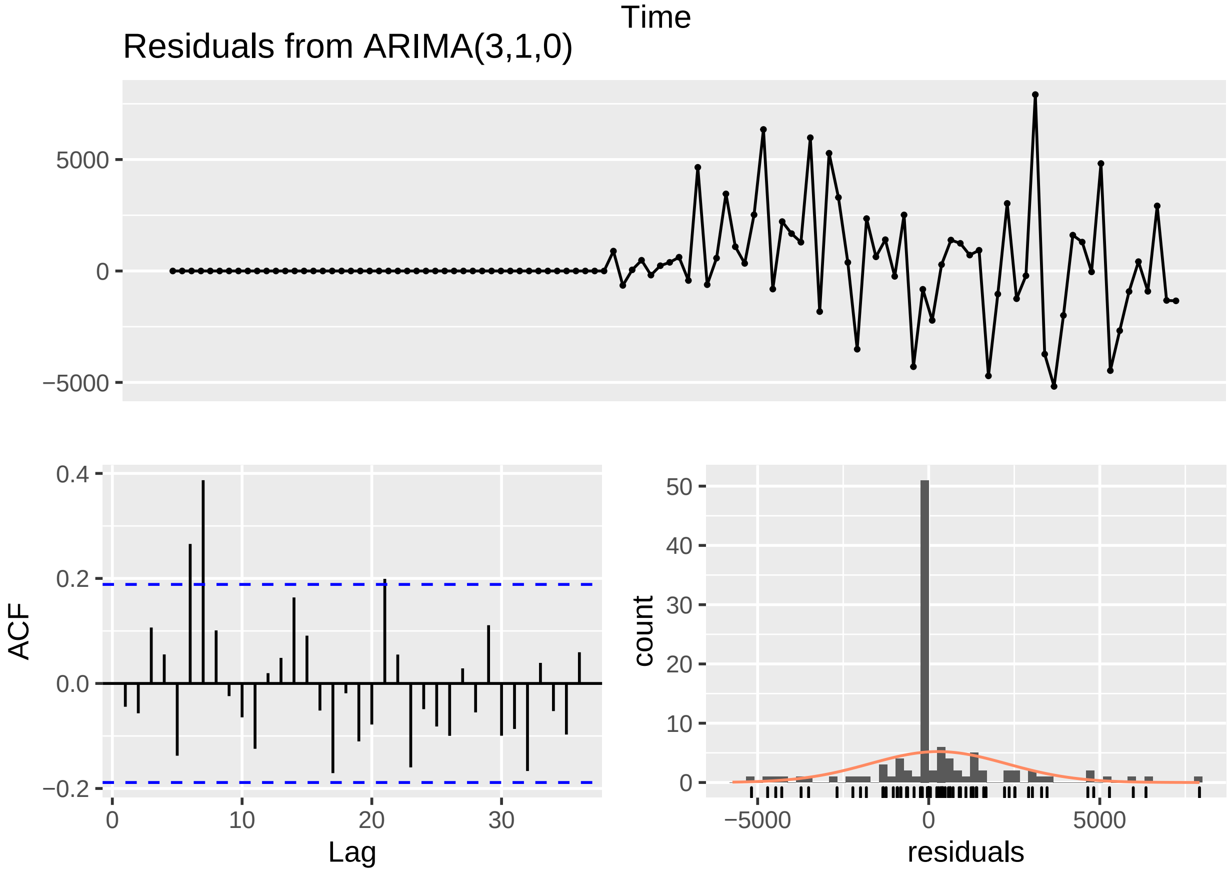


```
## Series: tsUS[, "Cases"]
## ARIMA(3,1,0)
##
## Coefficients:
##          ar1     ar2      ar3
##       0.0880  0.0681  -0.3518
## s.e.  0.0899  0.0899   0.0909
##
## sigma^2 estimated as 4212835:  log likelihood=-966.58
## AIC=1941.16   AICc=1941.55   BIC=1951.85
```

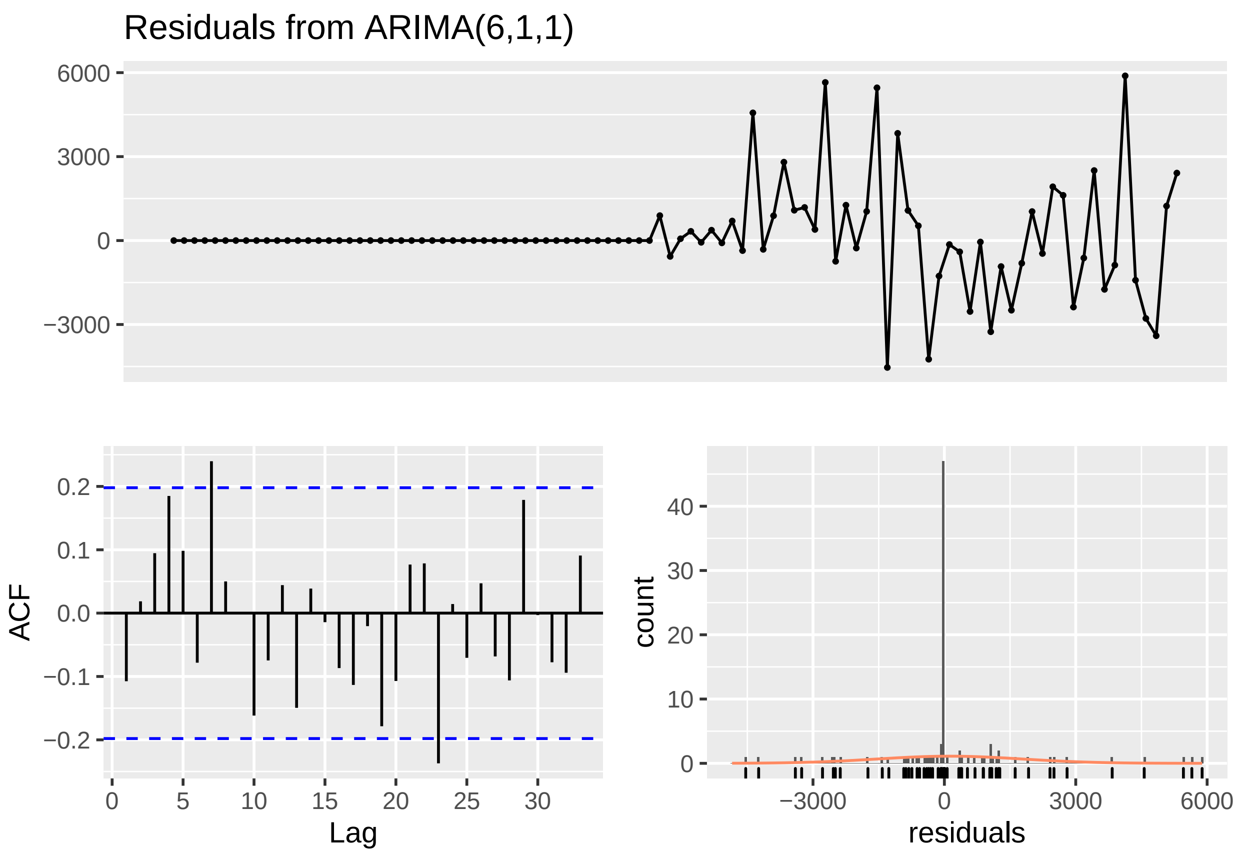
```
##                    ME     RMSE      MAE      MPE     MAPE MASE        ACF1
## Training set 276.3756 2014.151 1116.973 4.111018 18.50812  NaN -0.04432901
```

```
## $pred
## Time Series:
## Start = c(2020, 121)
## End = c(2020, 130)
## Frequency = 365
##  [1] 29586.74 29614.77 29167.82 28784.49 29075.90 29559.83 30072.01 30385.70
##  [9] 30628.82 30848.13
##
## $se
## Time Series:
## Start = c(2020, 121)
## End = c(2020, 130)
## Frequency = 365
##  [1] 1930.752 2814.316 3547.051 3878.738 4159.866 4399.658 4684.584 4962.140
##  [9] 5235.474 5479.805
```





```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,1,0)
## Q* = 51.198, df = 19, p-value = 8.715e-05
##
## Model df: 3.   Total lags used: 22
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(6,1,1)
## Q* = 28.265, df = 13, p-value = 0.008309
##
## Model df: 7.   Total lags used: 20
```

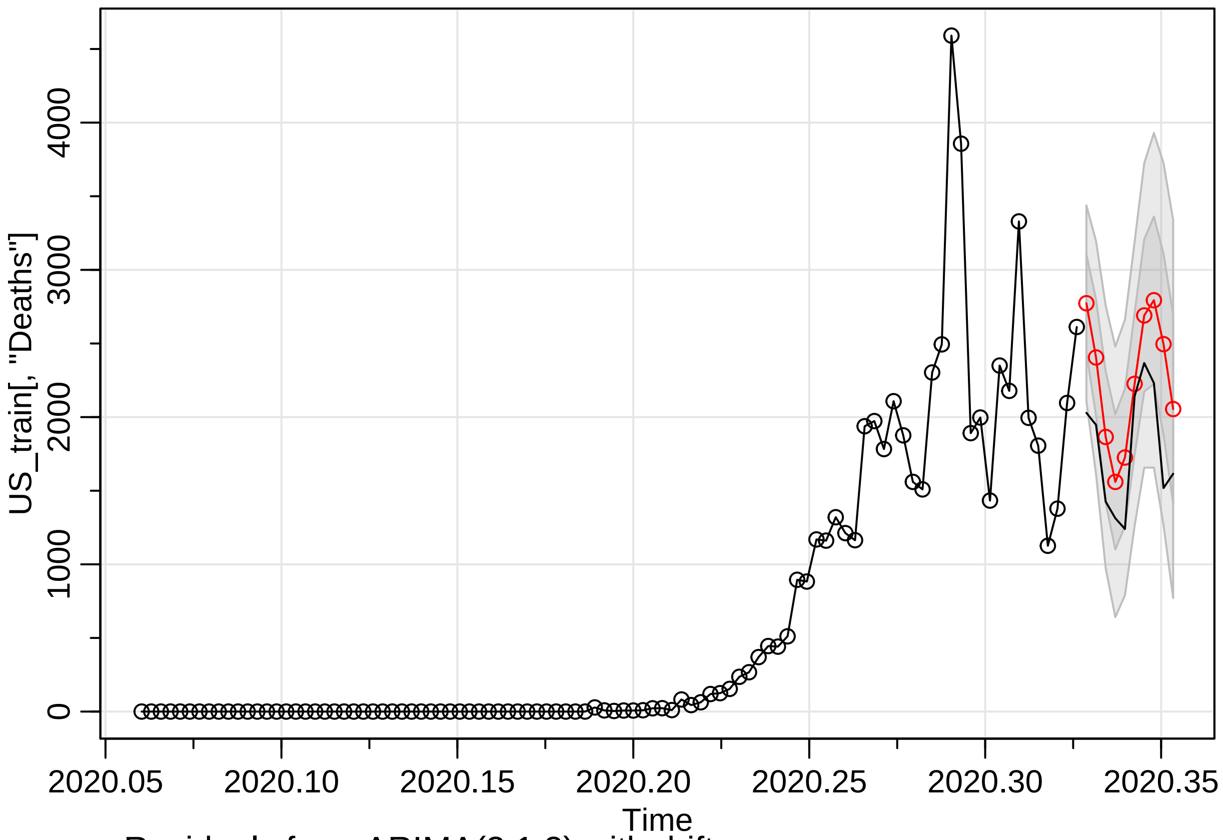
```
##
## Call:
## arima(x = US_train[, "Cases"], order = c(6, 1, 1))
##
## Coefficients:
##          ar1      ar2      ar3     ar4      ar5     ar6      ma1
##       0.5801  -0.0050  -0.2200  0.0935  -0.1474  0.5265  -0.5837
## s.e.  0.1051   0.1023   0.1007  0.1075   0.1051  0.0934   0.0853
##
## sigma^2 estimated as 2758995:  log likelihood = -858.42,  aic = 1732.85
```

```
##                     ME     RMSE      MAE      MPE     MAPE      MASE       ACF1
## Training set 122.2251 1652.526 872.3213 3.833422 17.88521 0.7964005 -0.1074746
```

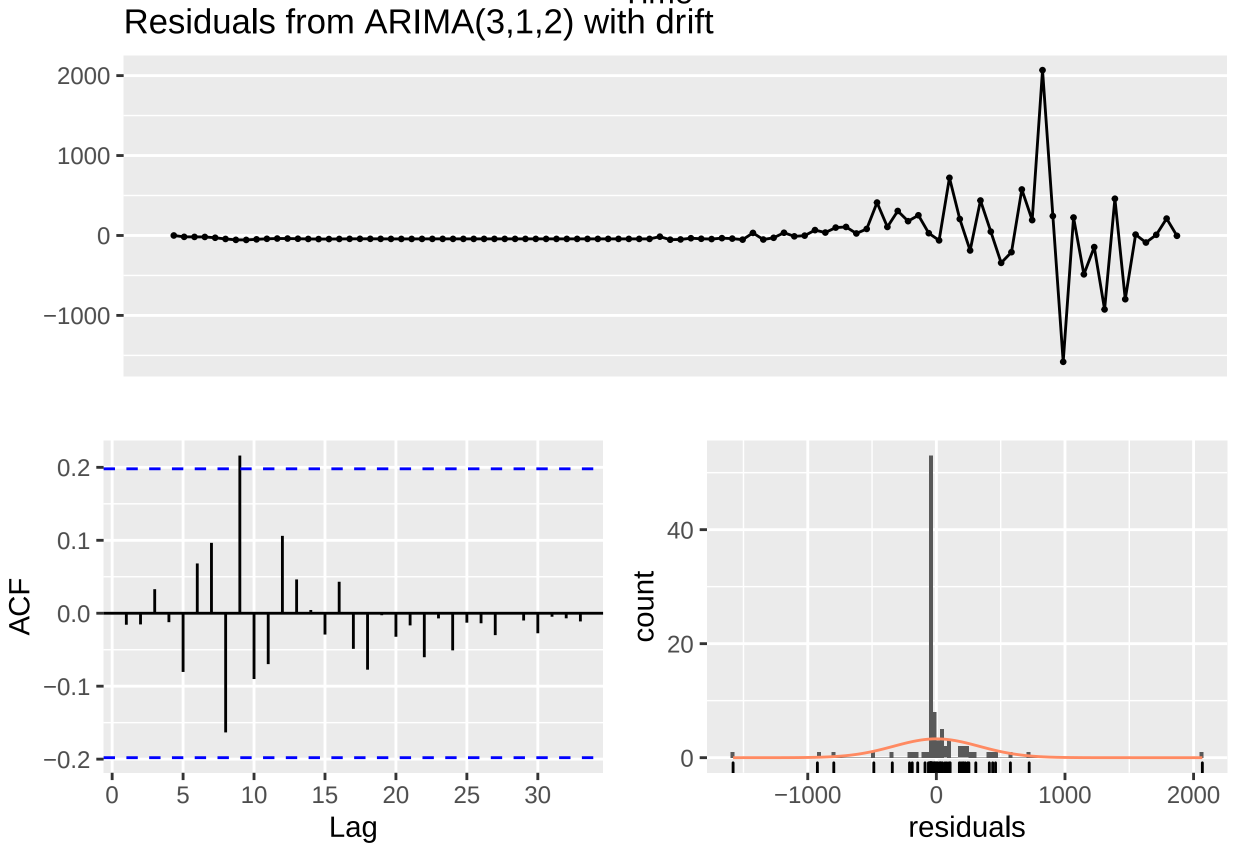
```
## Series: US_train[, "Deaths"]
## ARIMA(3,1,2) with drift
##
## Coefficients:
##          ar1      ar2      ar3      ma1     ma2    drift
##       0.7643  -0.4116  -0.4170  -1.1216  0.6612  21.6652
## s.e.  0.1434   0.1374   0.1087   0.1409  0.1204  17.2991
##
## sigma^2 estimated as 117938:  log likelihood=-701.95
## AIC=1417.89   AICc=1419.15   BIC=1435.91
```

```
##                       ME     RMSE      MAE  MPE MAPE MASE        ACF1
## Training set -0.07935582 330.9289 146.4972 -Inf  Inf  NaN -0.01576251
```

```
## $pred
## Time Series:
## Start = c(2020, 121)
## End = c(2020, 130)
## Frequency = 365
##  [1] 2773.241 2404.990 1865.051 1559.777 1725.328 2225.731 2690.403 2793.599
##  [9] 2495.591 2054.635
##
## $se
## Time Series:
## Start = c(2020, 121)
## End = c(2020, 130)
## Frequency = 365
##  [1] 332.6304 395.4061 445.8335 459.2506 467.9447 482.3156 516.8986 568.2344
##  [9] 614.7692 641.7556
```



US_train[, "Deaths"]
0
1000
2000
3000
4000
2020.05
2020.10
2020.15
2020.20
2020.25
2020.30
2020.35
Time



Residuals from ARIMA(3,1,2) with drift
2000
1000
0
−1000
ACF
0.2
0.1
0.0
−0.1
−0.2
0
5
10
15
20
25
30
Lag
count
40
20
0
−1000
0
1000
2000
residuals

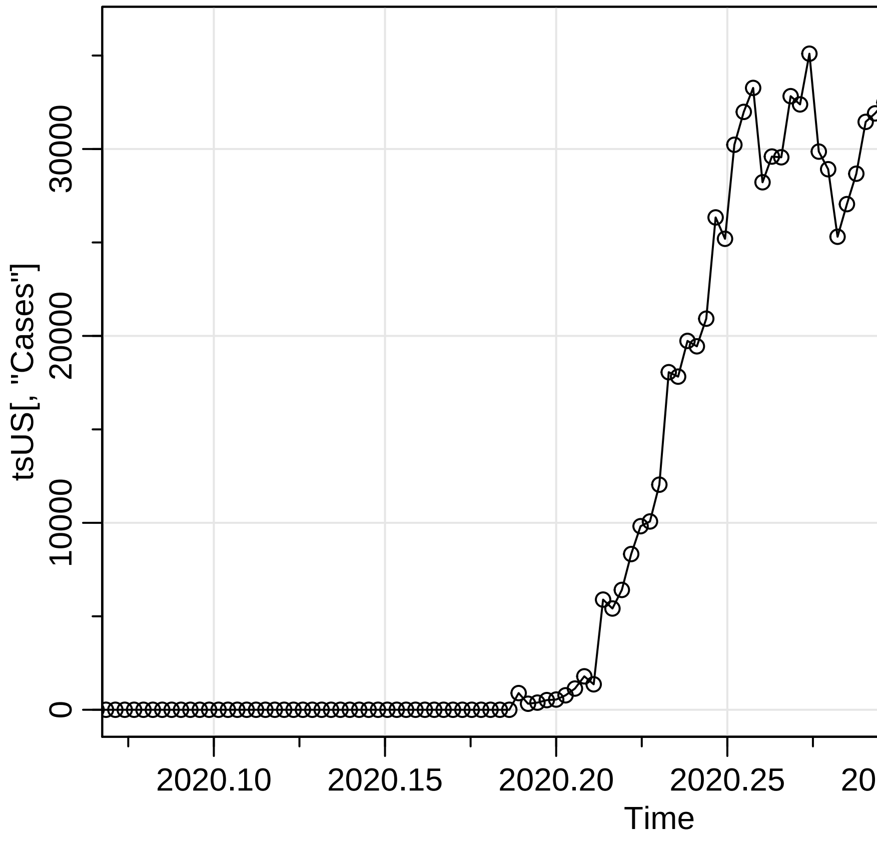
```
##
```

```
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,1,2) with drift
## Q* = 14.874, df = 14, p-value = 0.3868
##
## Model df: 6.   Total lags used: 20
```

### Using the Optimal Univariate Model to Forecast

The models above were used to forecast 10 days of cases and deaths for the US. Because these predictions were made for May 10th through May 19th, I was able to use actual data to check the accuracy of the forecasts. Using the RMSE, I was pleased to find that both models were relatively accurate, although the cases model was impressive with an RMSE of 0.10. The deaths model had an RMSE of 0.53, so this model was less accurate. Note that my ARIMA models are only accurate for short stretches of time because each new prediction is heavily influenced by the most recent days' data. In this way, my ARIMA models are not very stable because they must be updated frequently to remain accurate.

```r
# Use best models to forecast further ahead for US cases and deaths
fc_10_US <- sarima.for(tsUS[, "Cases"], n.ahead = 10, 6, 1, 1)
```



### ARIMA forecasts for US cases and deaths

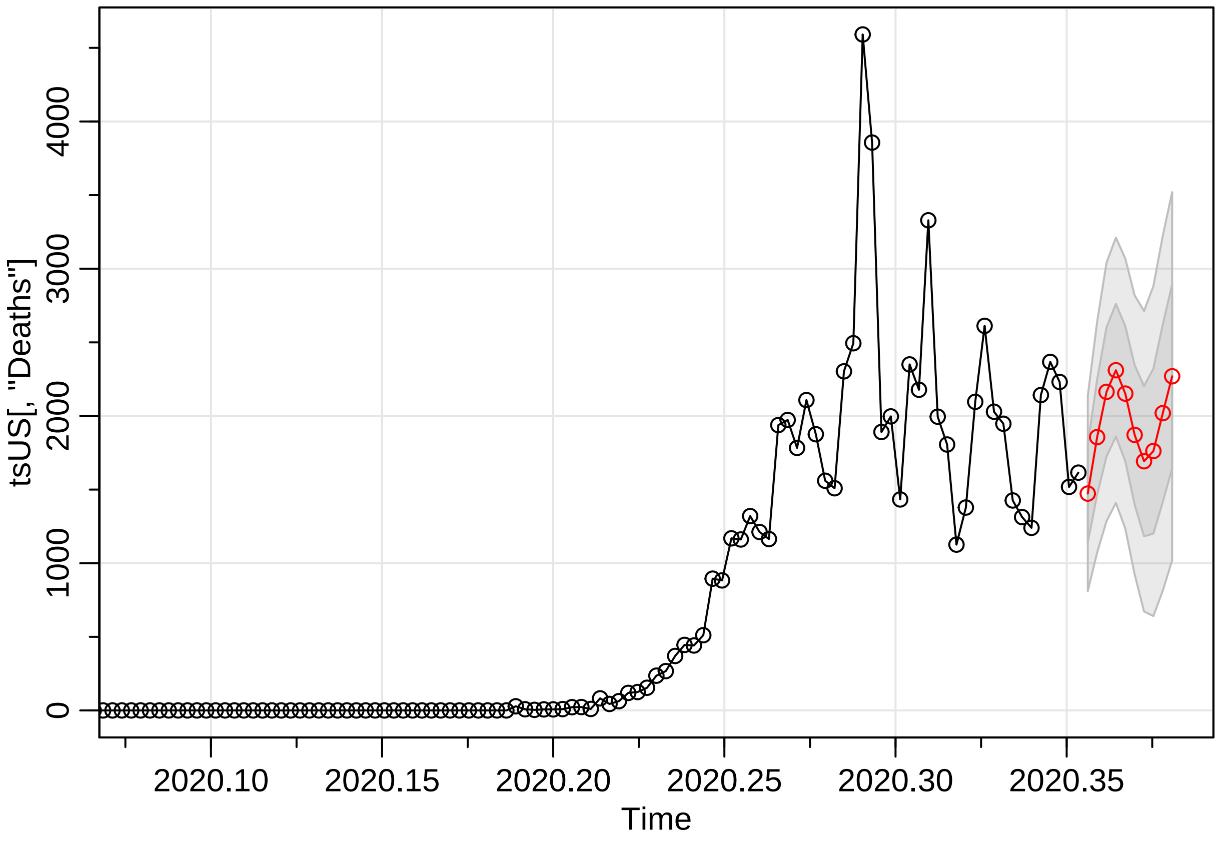
```r
fc_10_US$pred
```

```
## Time Series:
## Start = c(2020, 131)
## End = c(2020, 140)
## Frequency = 365
##  [1] 22253.63 21815.94 21274.50 23907.89 24571.55 24898.39 22885.37 21726.62
##  [9] 20432.77 21416.68
```

```
actual_US <- c(19710, 18618, 21693, 20832, 27368, 25050, 24994, 18937, 21551, 20260) # actual US cases
RMSE(fc_10_US$pred, actual_US)/mean(actual_US) # 0.10 VERY GOOD
```

```
## [1] 0.101064
```

```
fcd_10_US <- sarima.for(tsUS[, "Deaths"], n.ahead = 10, 3, 1, 2)
```



```
fcd_10_US$pred
```

```
## Time Series:
## Start = c(2020, 131)
## End = c(2020, 140)
## Frequency = 365
##  [1] 1472.774 1856.263 2163.776 2310.510 2151.882 1871.445 1692.389 1761.825
##  [9] 2019.192 2269.593
```

```
actual_USd <- c(731, 1156, 1694, 1743, 1779, 1632, 1224, 808, 785, 1574)
RMSE(fcd_10_US$pred, actual_USd)/mean(actual_USd) # 0.53 - higher than cases but still strong
```

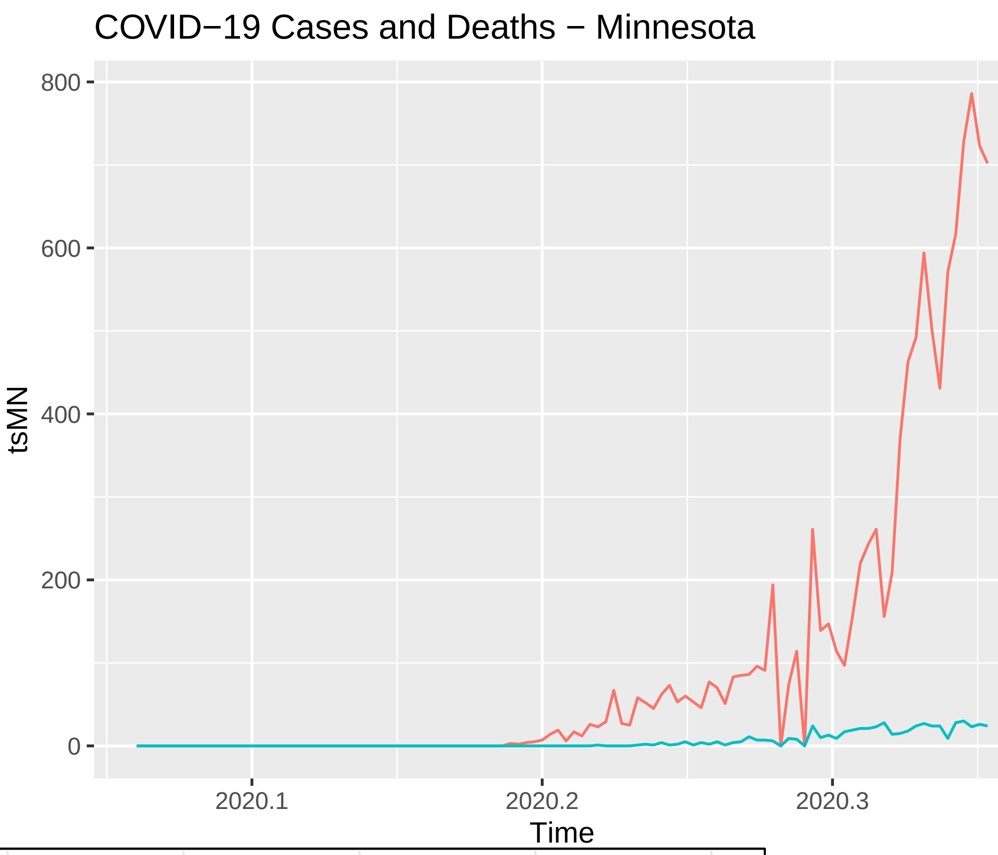
```
## [1] 0.5341616
```

```
# models show cases declining while deaths are steady
```

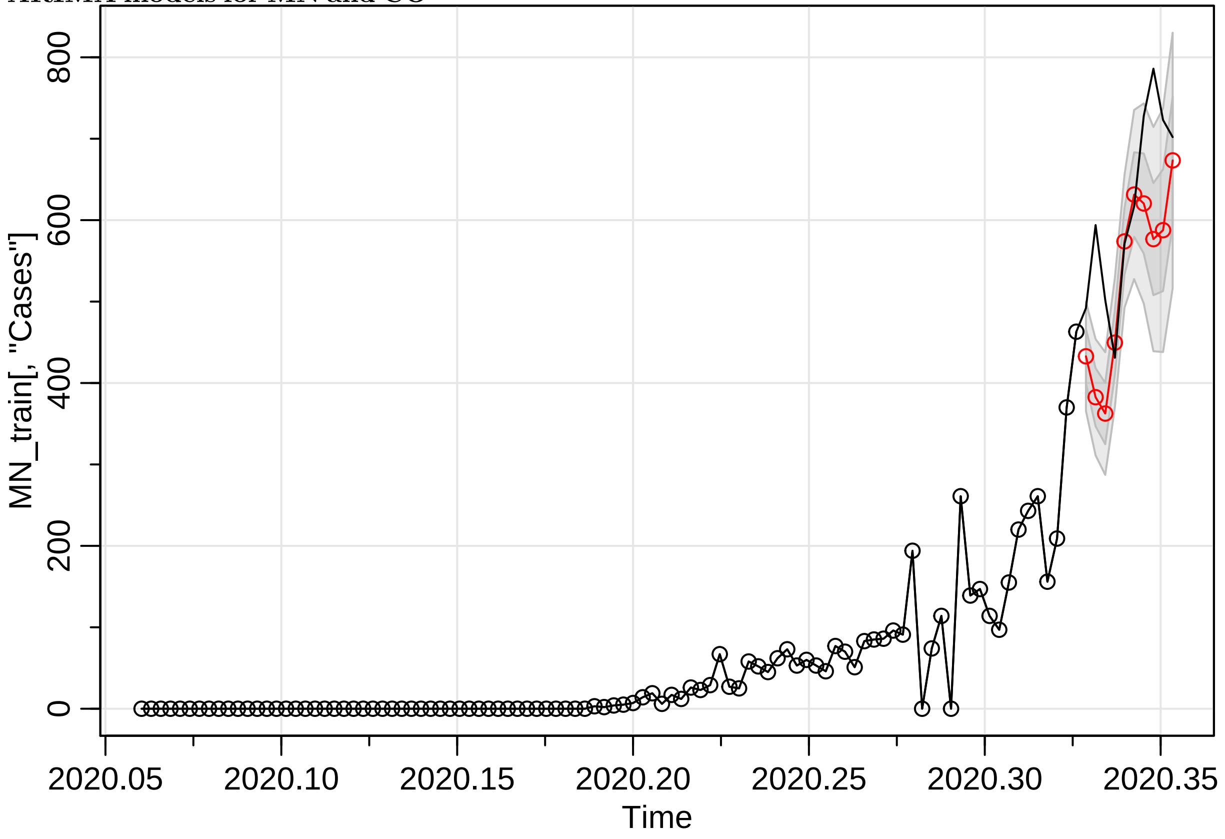
### ARIMA Models for States and Counties

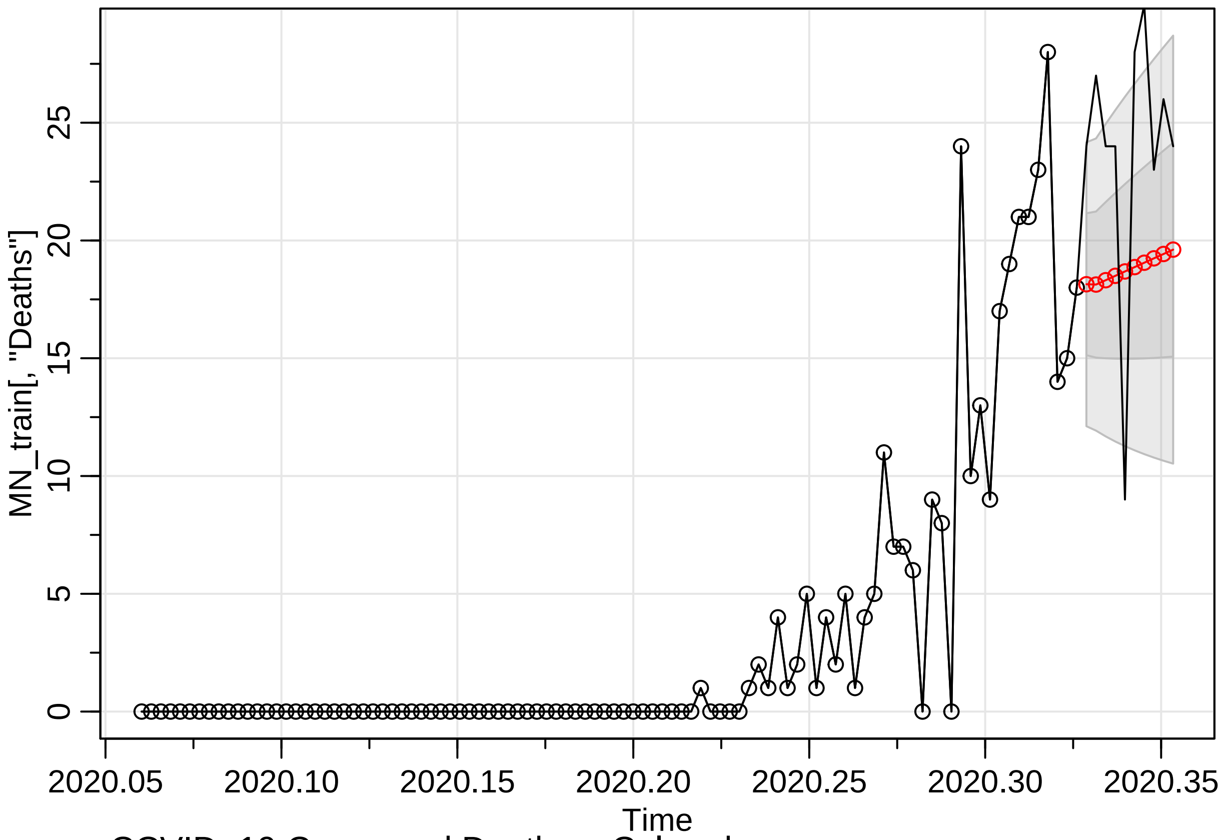
Using the same basic logic (and code), I created ARIMA models for seven states and 34 counties. These models can be found in my github code, but select states and counties are shown below. The state models are still quite strong but the ARIMA models are weaker at the county level, probably because the case and death counts are lower.

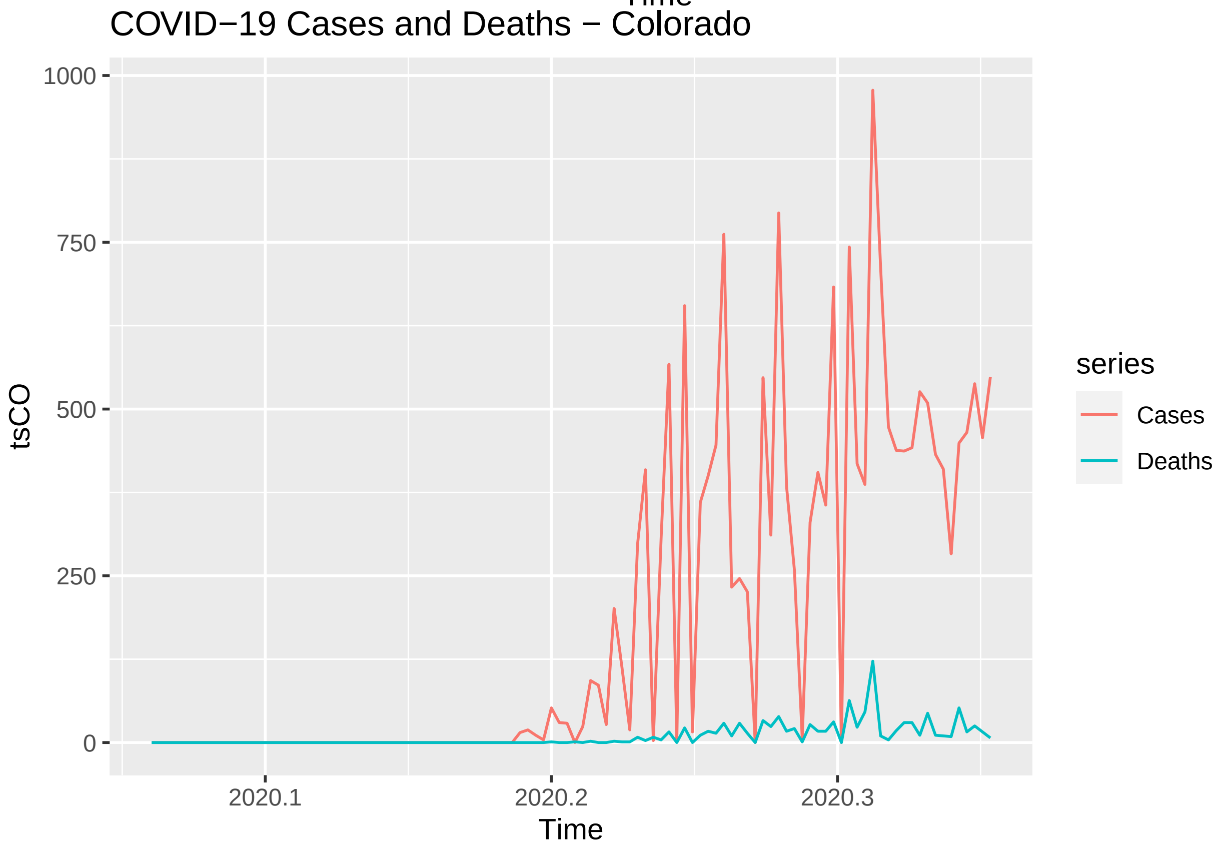
For the models below, CO had a cases RMSE of 0.32 and a deaths RMSE of 0.93. MN had a cases RMSE of 0.19 and a deaths RMSE of 0.31.

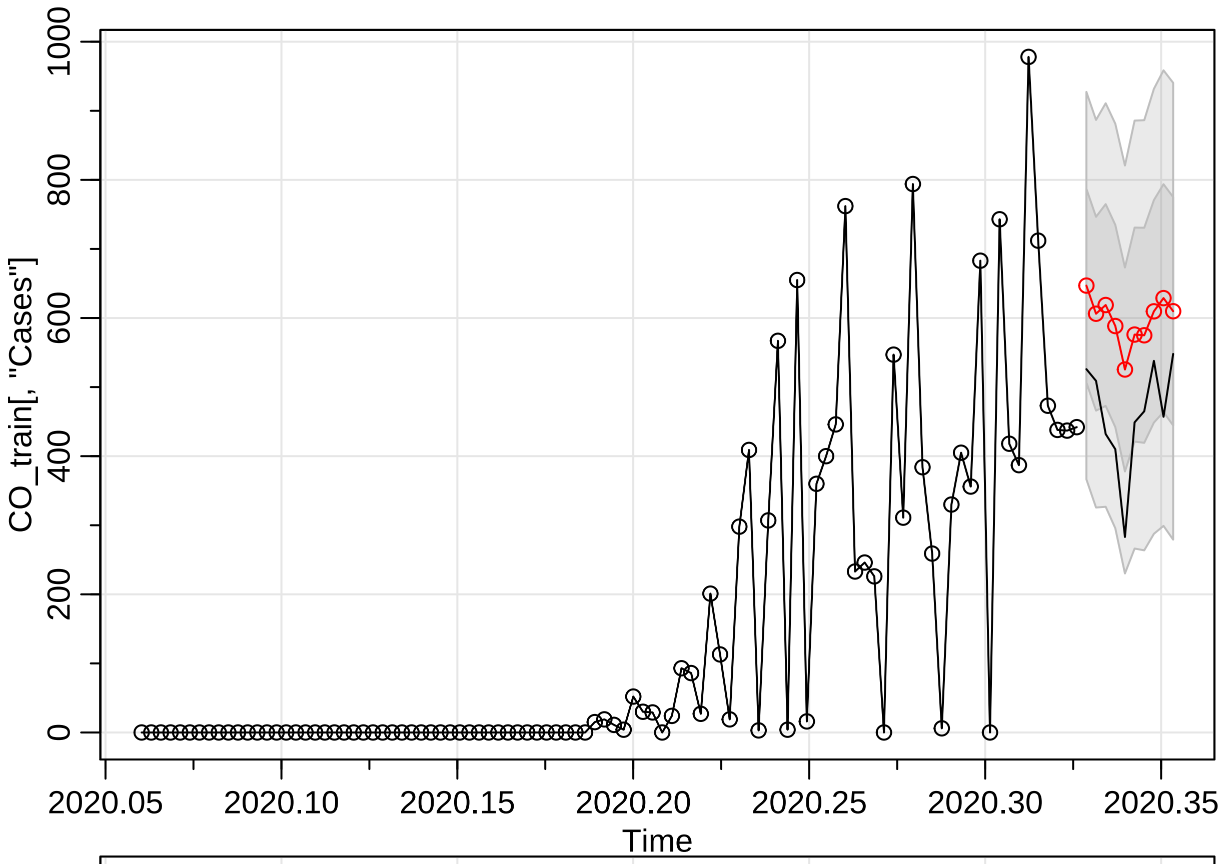


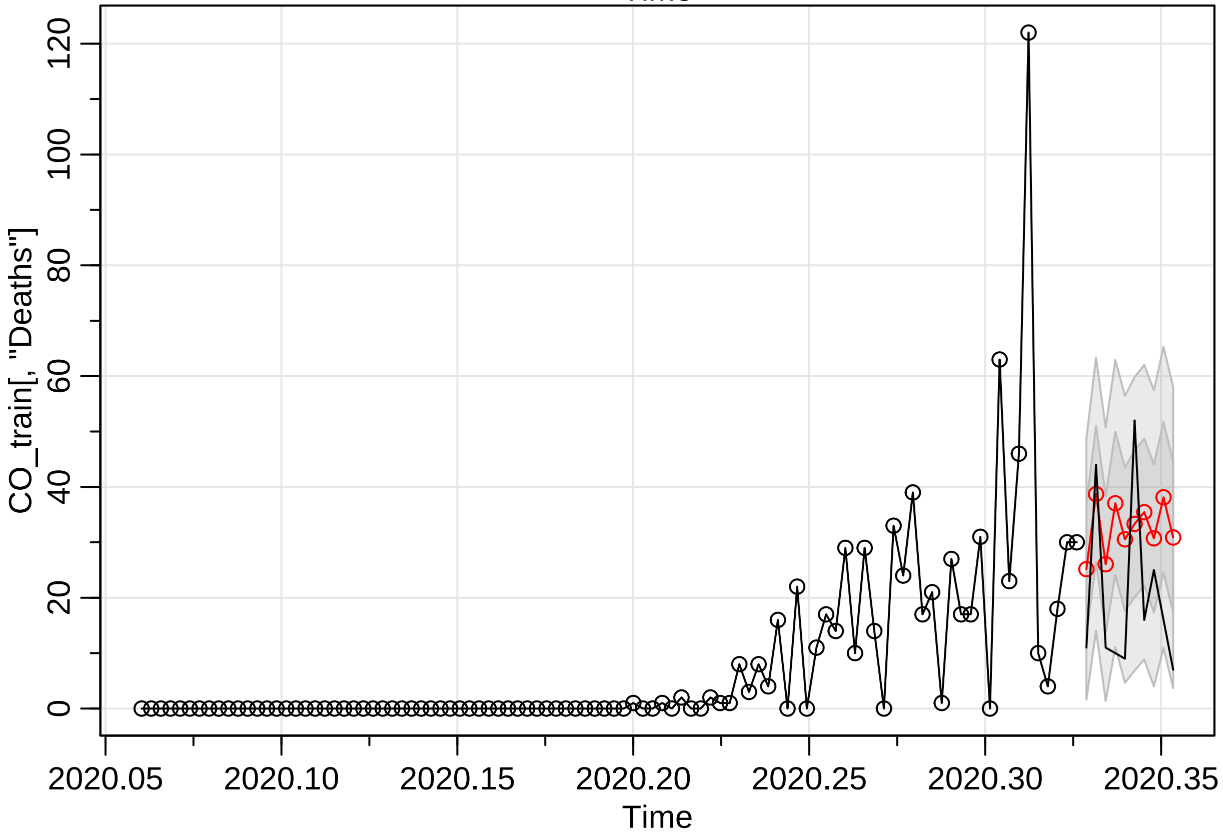
**ARIMA models for MN and CO**



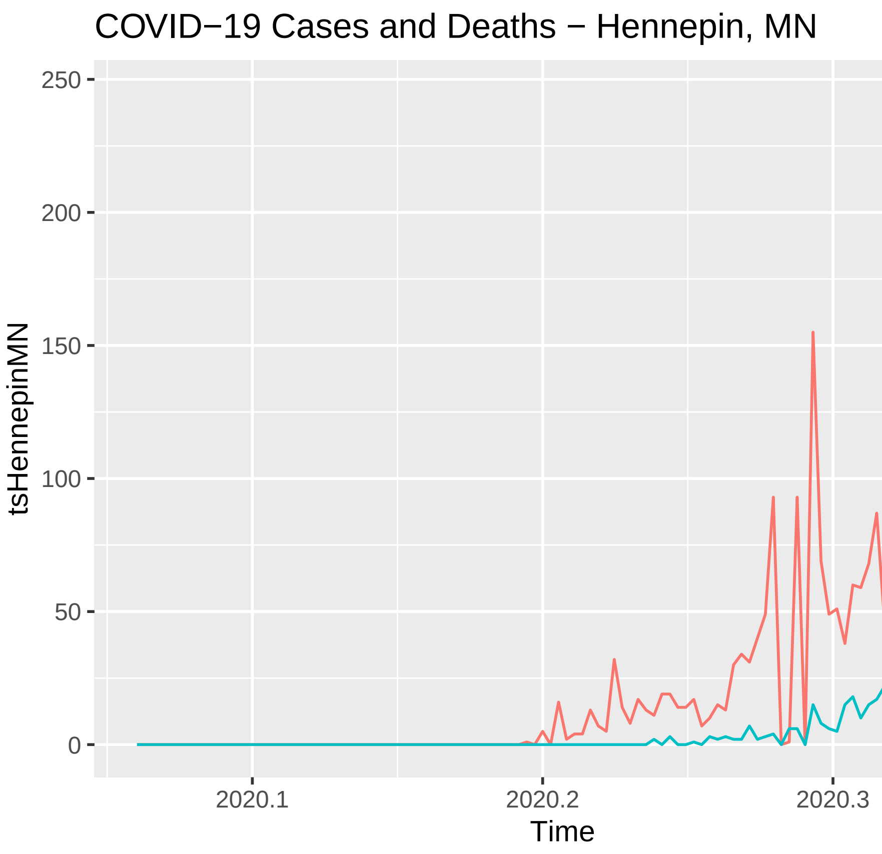

MN_train[, "Deaths"]
0
5
10
15
20
25
2020.05
2020.10
2020.15
2020.20
2020.25
2020.30
2020.35
Time


COVID−19 Cases and Deaths − Colorado
tsCO
0
250
500
750
1000
2020.1
2020.2
2020.3
Time
series
Cases
Deaths



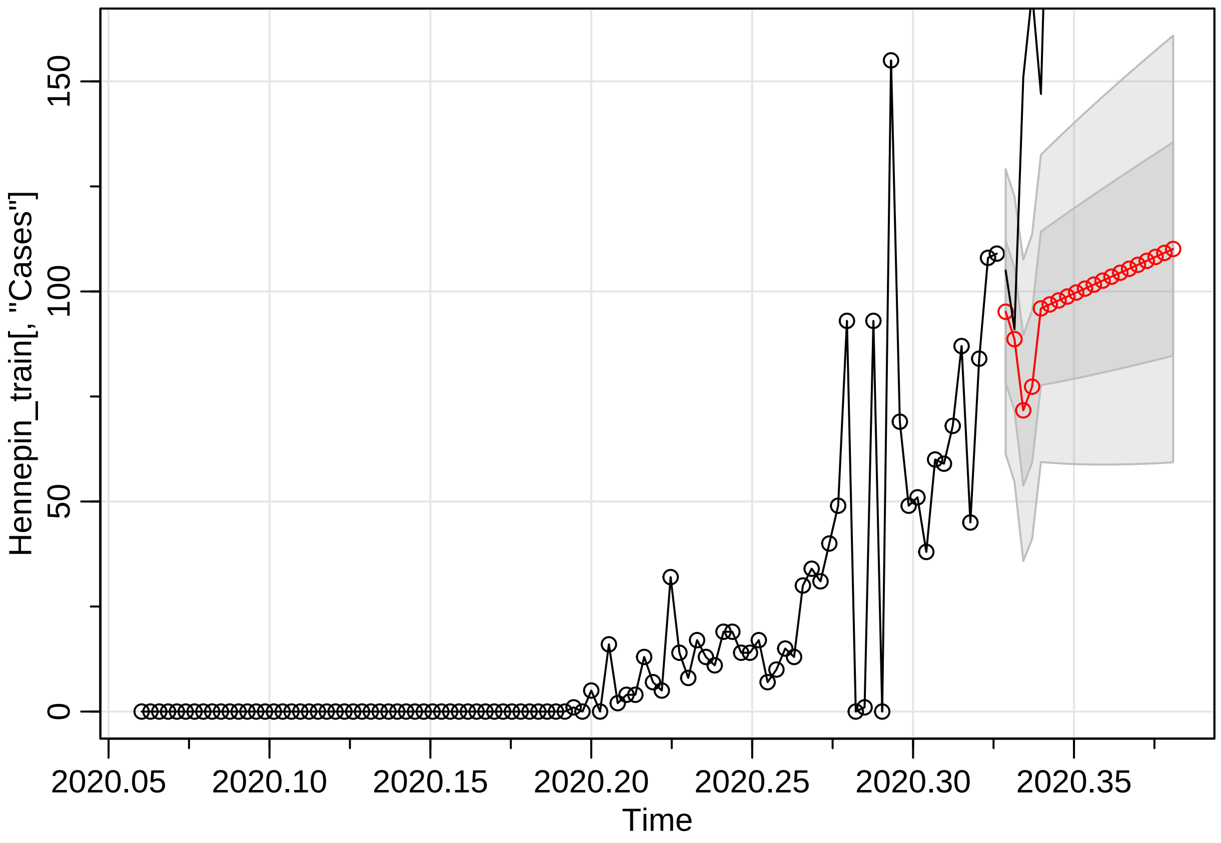


For the model below, Hennepin county had a cases RMSE of 0.87 and a deaths RMSE of 0.59. Kandiyohi county had a cases RMSE of 0.98 and a deaths RMSE of 1.02. Stearns county had a cases RMSE of 0.48 and a deaths RMSE that could not be calculated because deaths are so low in this county that the arima function identified a white noise model.

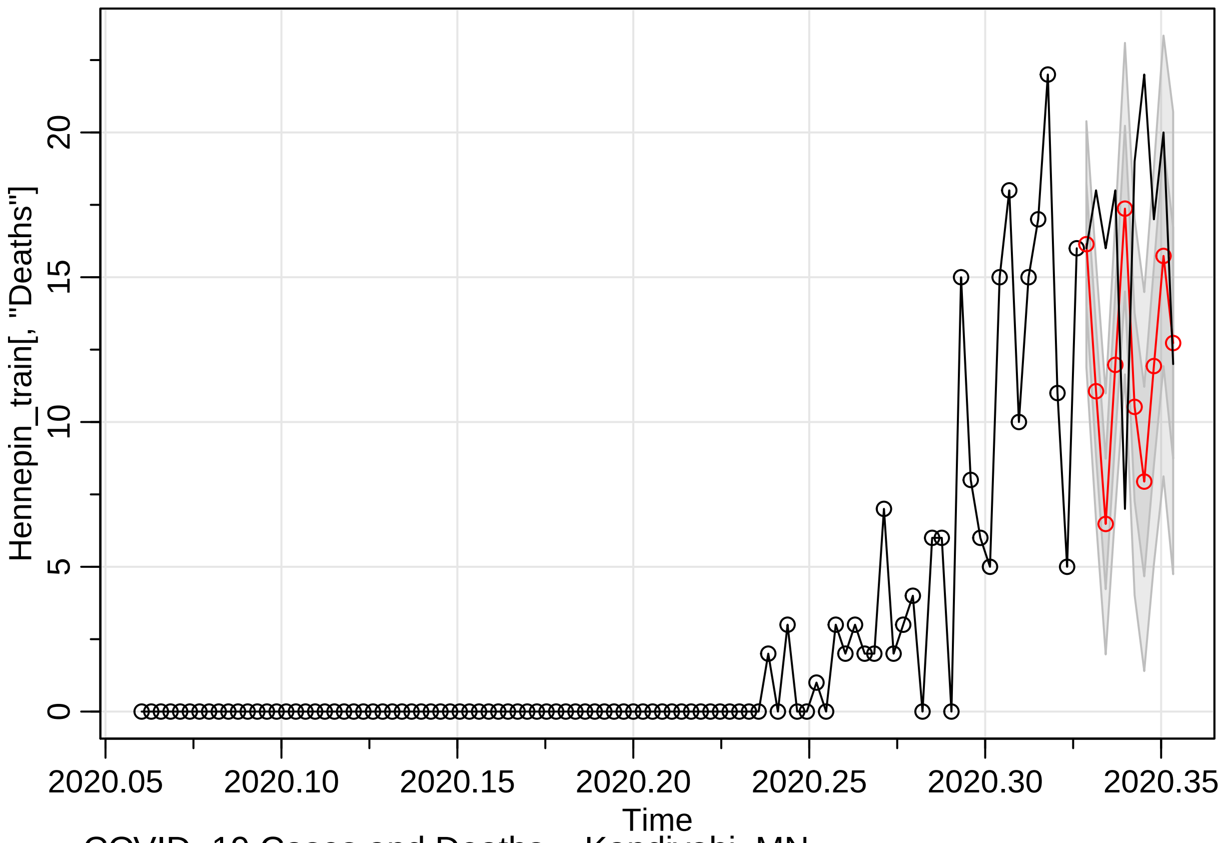


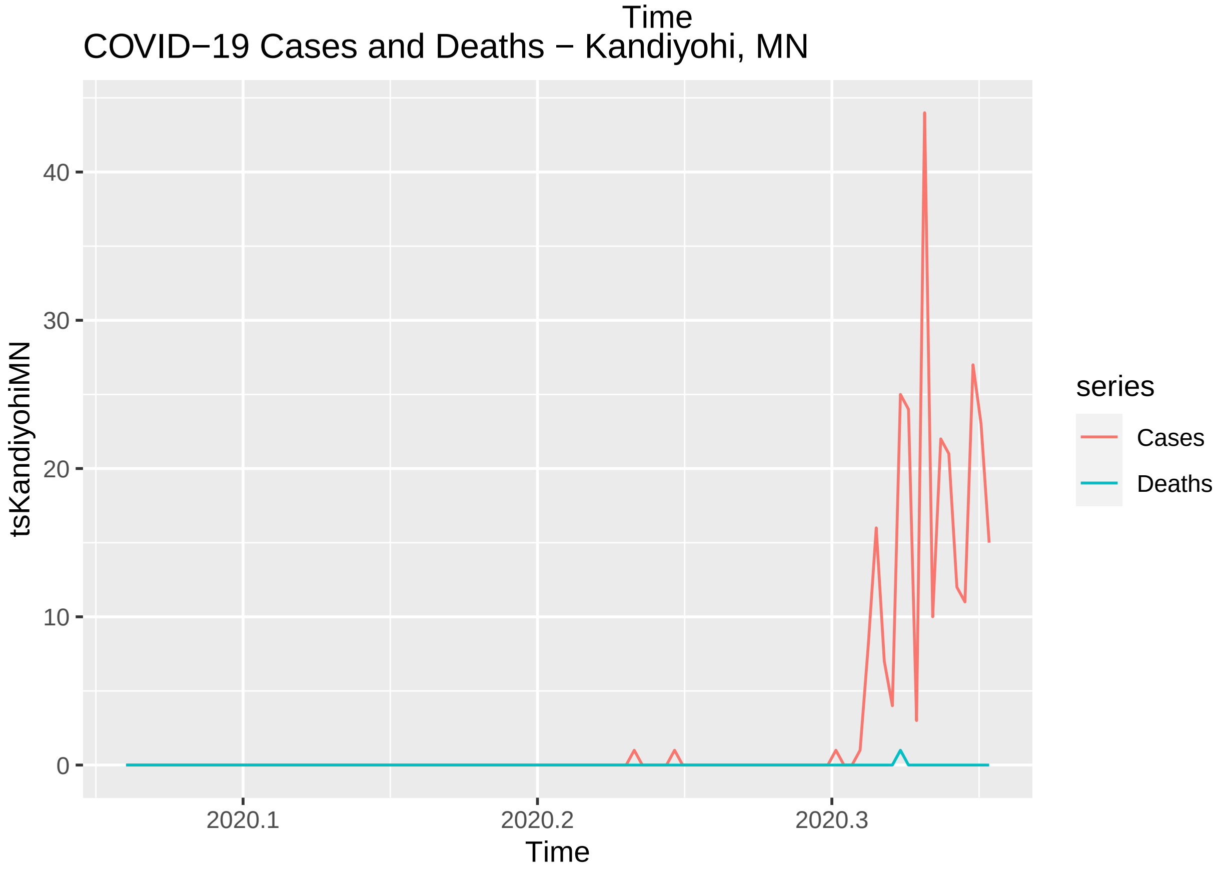
**ARIMA models for 3 Minnesota counties**

```
## Series: Hennepin_train[, "Cases"]
## ARIMA(0,1,5) with drift
##
## Coefficients:
##           ma1     ma2      ma3      ma4     ma5   drift
##       -0.9458  0.2858  -0.1809  -0.2953  0.4047  0.9433
## s.e.   0.1015  0.1330   0.1488   0.1287  0.0945  0.4671
##
## sigma^2 estimated as 306.4:  log likelihood=-413.3
## AIC=840.6   AICc=841.86   BIC=858.62
```

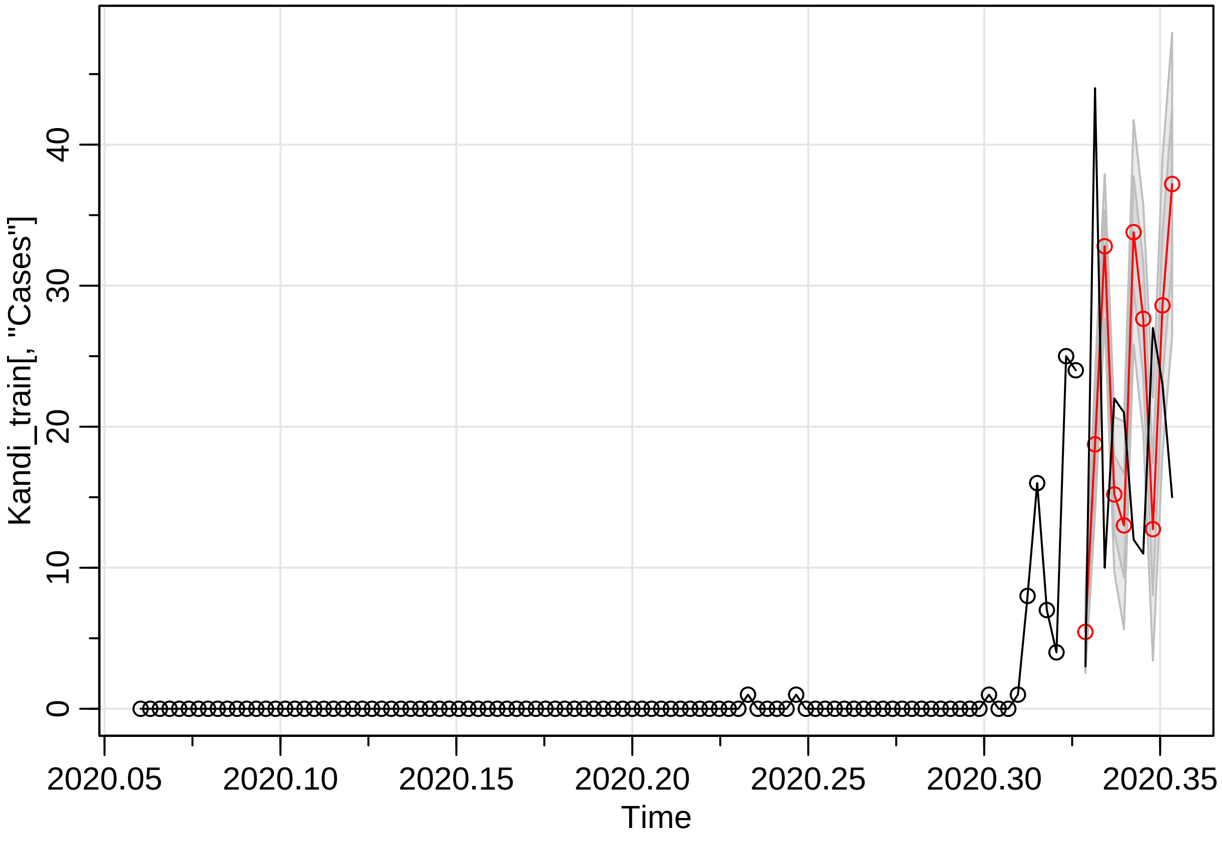


```
## Series: Hennepin_train[, "Deaths"]
## ARIMA(5,1,0)
##
## Coefficients:
##           ar1      ar2      ar3     ar4     ar5
##       -0.6724  -0.5864  -0.1379  0.2074  0.4959
## s.e.   0.0901   0.1153   0.1411  0.1308  0.1000
##
## sigma^2 estimated as 4.796:  log likelihood=-212.61
## AIC=437.22   AICc=438.16   BIC=452.67
```

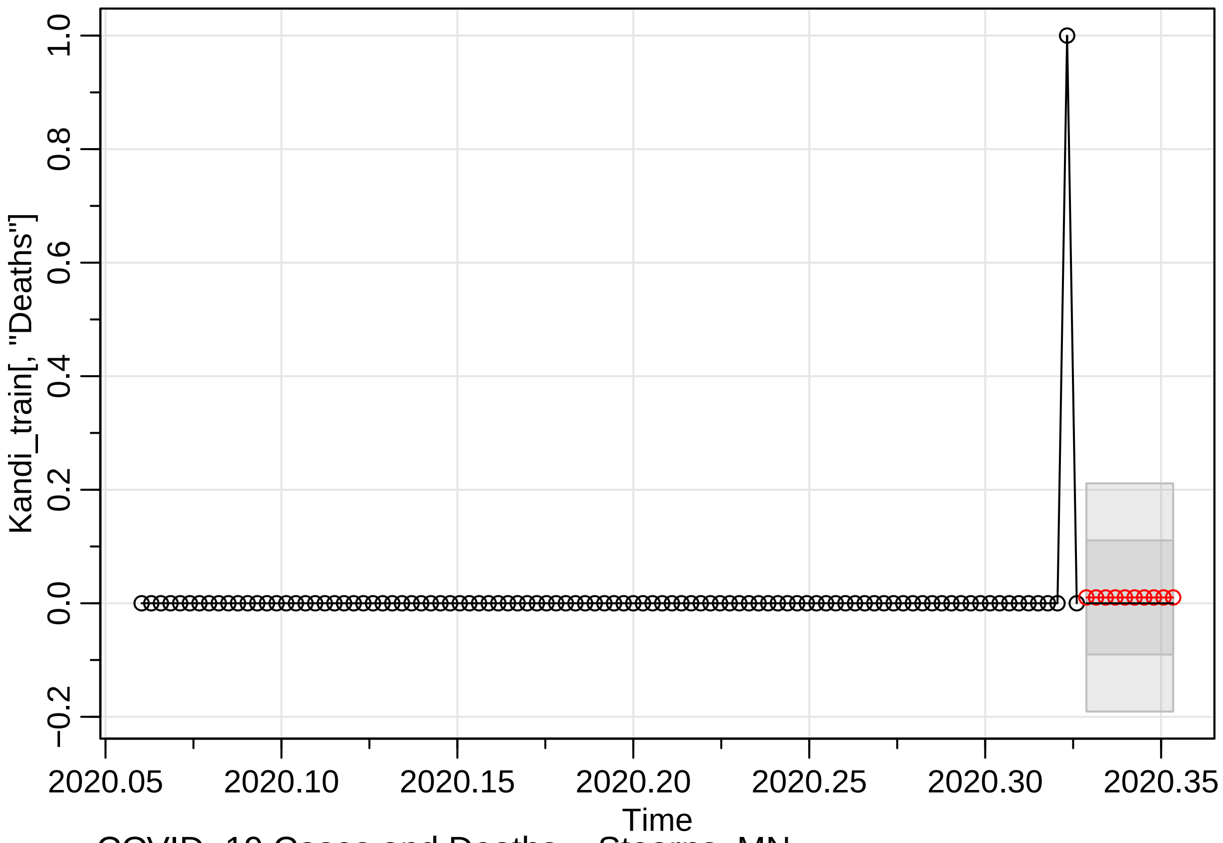


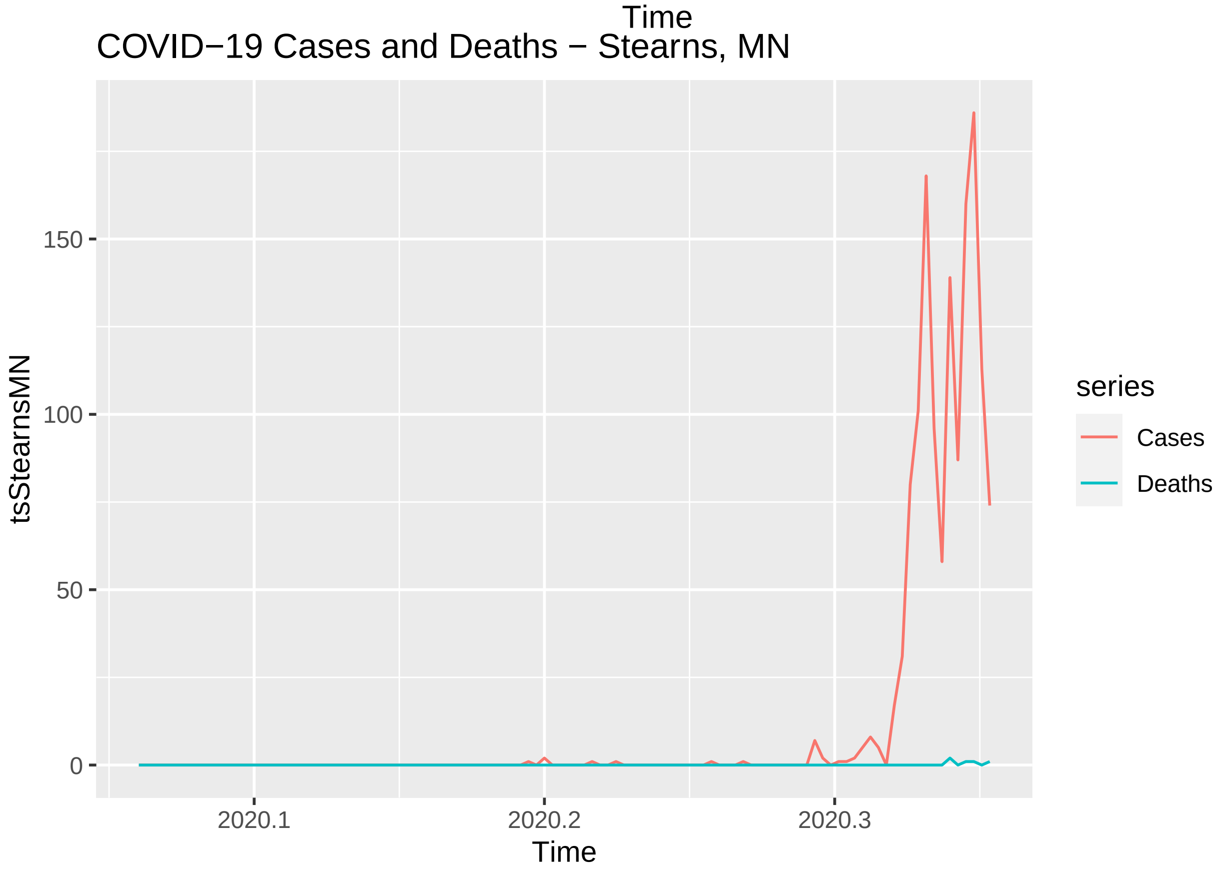


```
## Series: Kandi_train[, "Cases"]
## ARIMA(3,1,2) with drift
##
## Coefficients:
##          ar1      ar2     ar3      ma1      ma2   drift
##       0.4391  -0.4908  0.9314  -0.0534  -0.6719  0.3069
## s.e.  0.0925   0.0807  0.0672   0.1050   0.0961  0.3415
##
## sigma^2 estimated as 2.261:  log likelihood=-178.11
## AIC=370.22   AICc=371.47   BIC=388.24
```

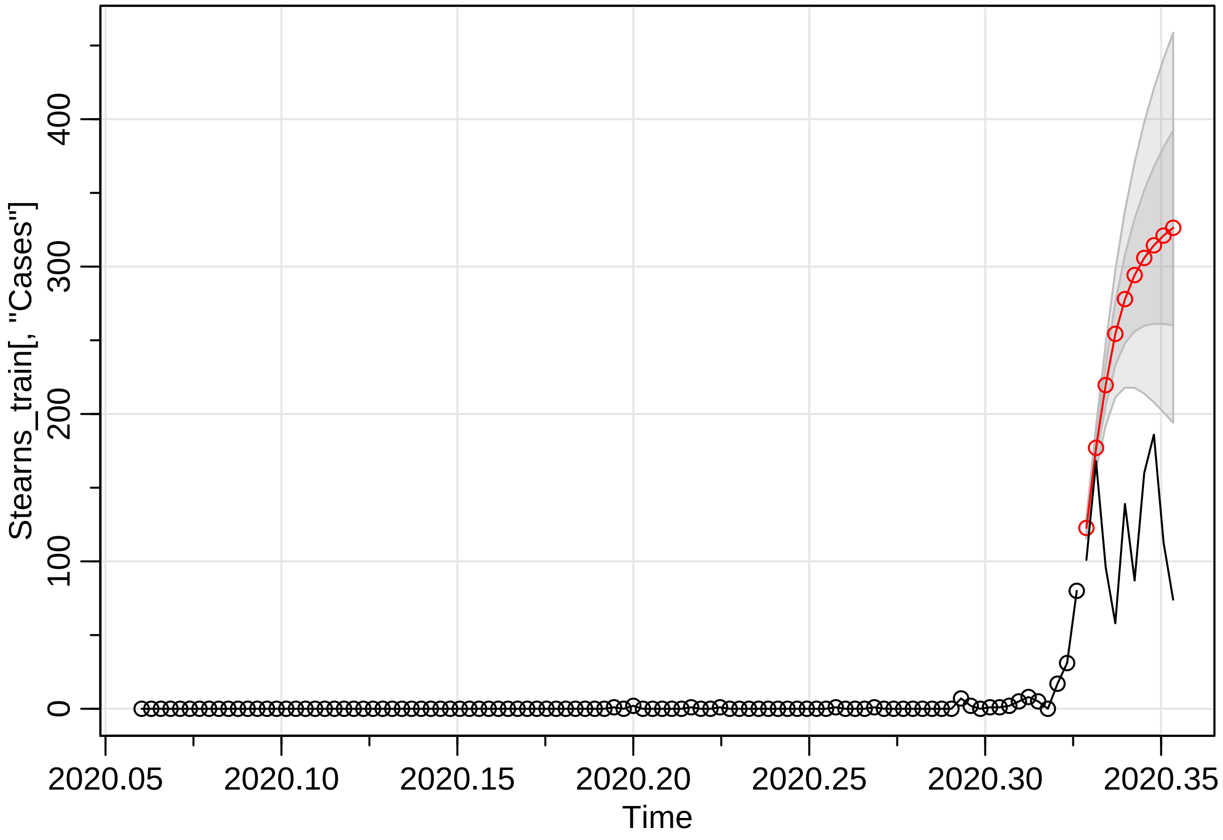


```
## Series: Kandi_train[, "Deaths"]
## ARIMA(0,0,0) with zero mean
##
## sigma^2 estimated as 0.0102:  log likelihood=85.61
## AIC=-169.21   AICc=-169.17   BIC=-166.63
```





```
## Series: Stearns_train[, "Cases"]
## ARIMA(1,1,4)
##
## Coefficients:
##          ar1     ma1     ma2     ma3     ma4
##       0.6769  0.0898  1.0063  0.3391  0.3454
## s.e.  0.3125  0.2474  0.1614  0.1686  0.2276
##
## sigma^2 estimated as 13.89:  log likelihood=-265.36
## AIC=542.72   AICc=543.66   BIC=558.17
```



```
## Series: Stearns_train[, "Deaths"]
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
## intercept
##         0
##
## sigma^2 estimated as 0:  log likelihood=Inf
## AIC=-Inf   AICc=-Inf   BIC=-Inf
```

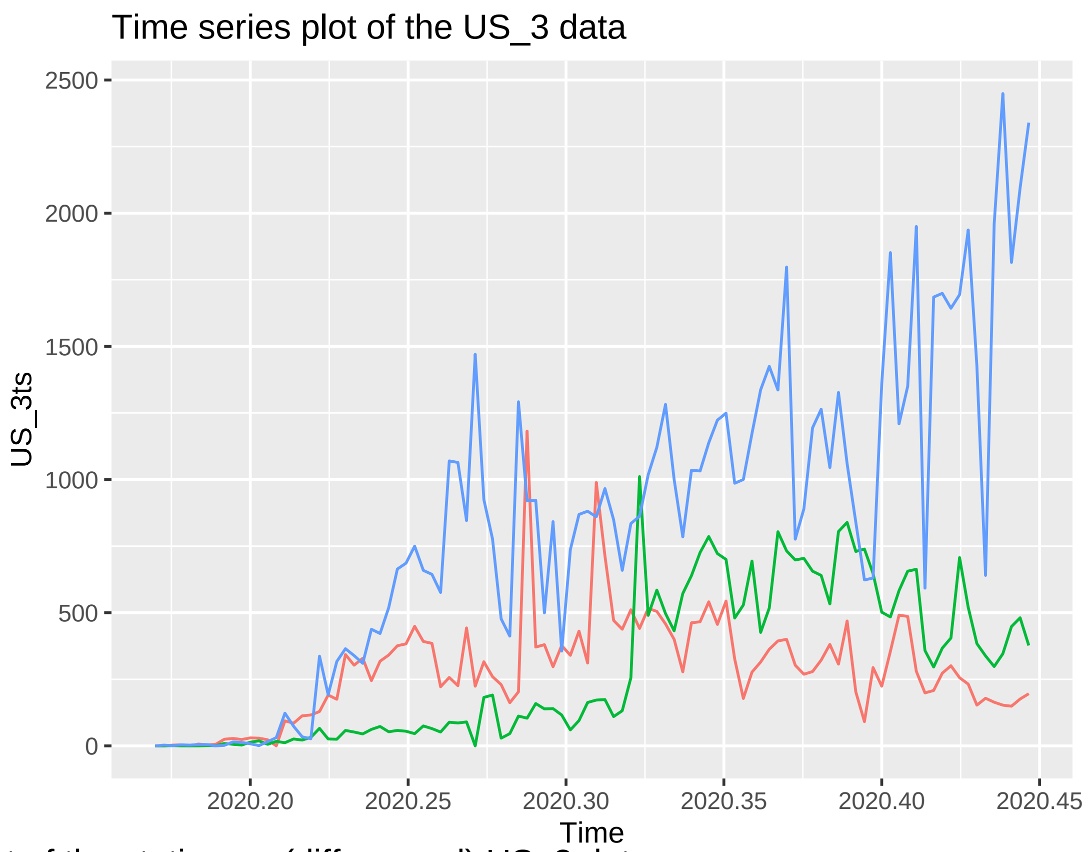
### Building Multivariate Models

NOTE: The following provide detailed information regarding Vector Autoregression and Multivariate Time Series Analysis:
* Introduction to Time Series Analysis and Forecasting in R
* CRAN Documentation for vars package
* An Introduction to Vector Autoregression (VAR)

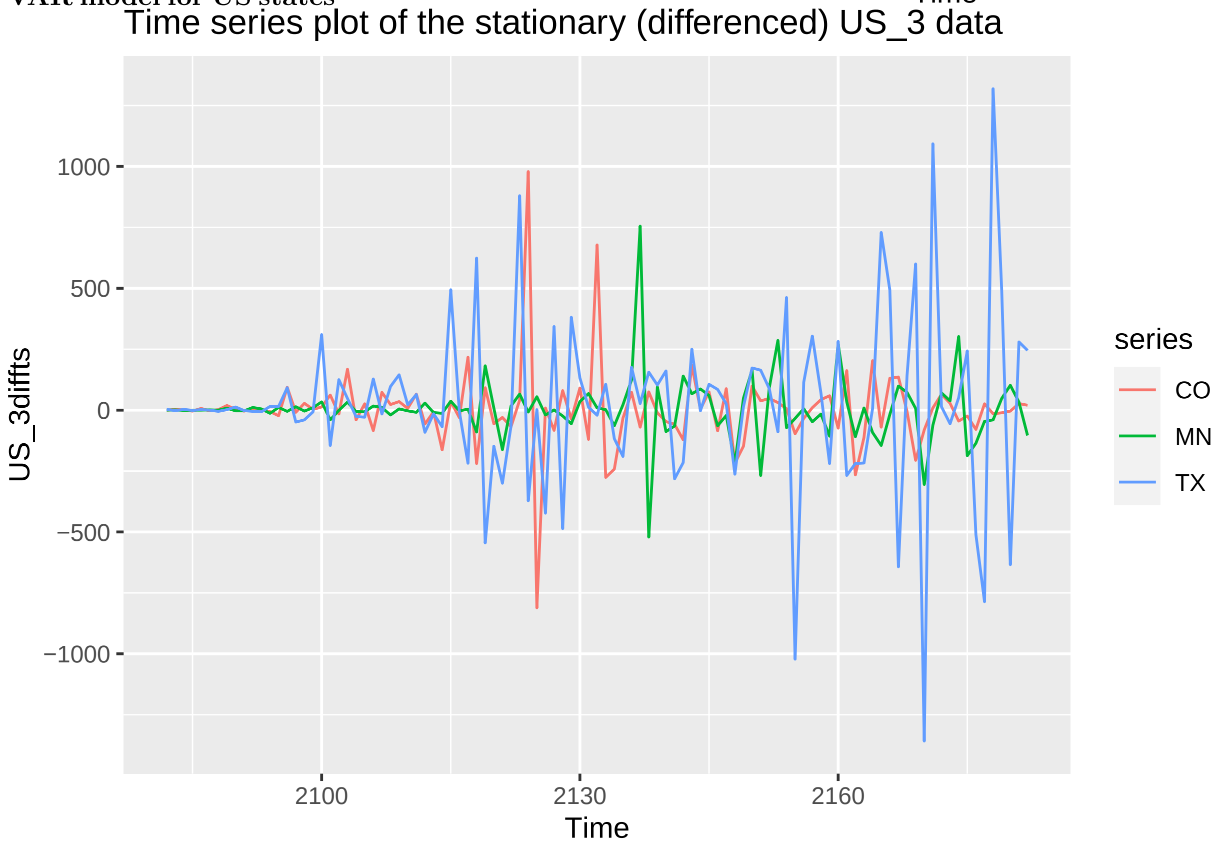
ARIMA models are useful when considering time series data that is made up of just one variable or unidirectional relationships where the forecast variable is influenced by predictor variables (exogenous variables or external regressors) but not vice versa. But there are many times in real world data when all variables affect each other. For example, COVID deaths might be influenced by the number of COVID cases, but it is unlikely that COVID cases can be predicted by deaths since there is a natural, biological order where cases comes first. COVID cases in one county, however, might very well both impact and be impacted by COVID cases in another county. Vector Autoregression models are used for this very purpose because they treat all variables symmetrically. In VAR models, all variables are treated as endogenous. These models can be very powerful but they are also difficult to interpret, so they are not used as frequently as univariate time series models.

The first set of VAR models that I built attempted to forecast cases and deaths for seven US states. The model assumed that each state's cases/deaths influenced each other state's cases/deaths. However, I found that my model was most accurate when I limited the number of variables or states. In the following code, I built two VAR models (cases and deaths) for only three states (CO, MN, TX).

Note that the cases model below had relatively high adjusted R-squared values (0.84 to 0.93), but there were some issues with autocorrelation as shown by the Portmanteau Test. The deaths model was just okay with adjusted R-squared values between 0.50 and 0.64 and autocorrelation was still present. The vars package did, however, have some interesting plots that allowed me to view impulse responses (if one state has sudden spikes, how will that impact other states) and forecast error variance decomposition (how much does each state affect each other state). In these models, each of the states is influenced by each of the others although CO cases might be slightly more independent than MN or TX.

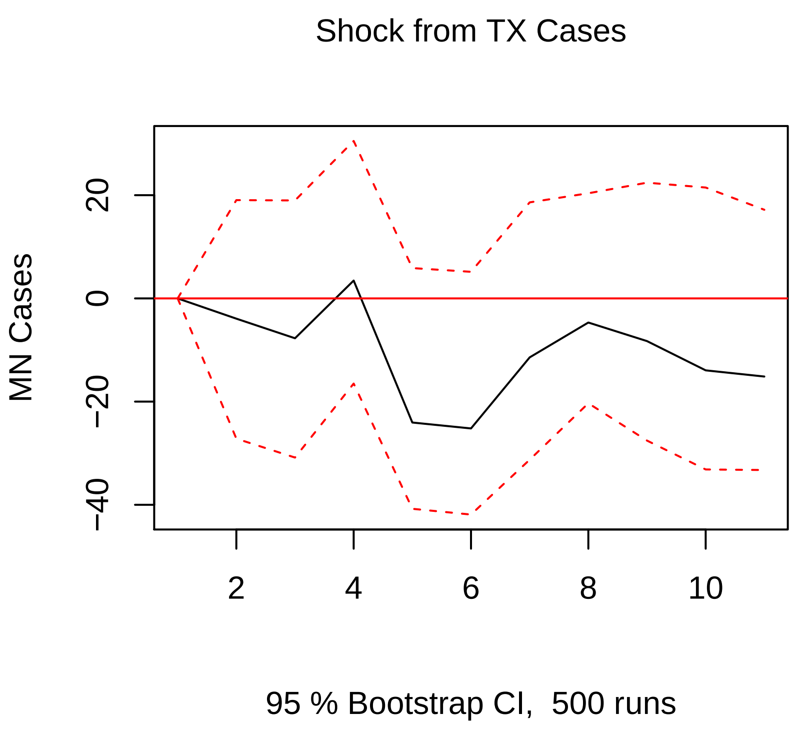


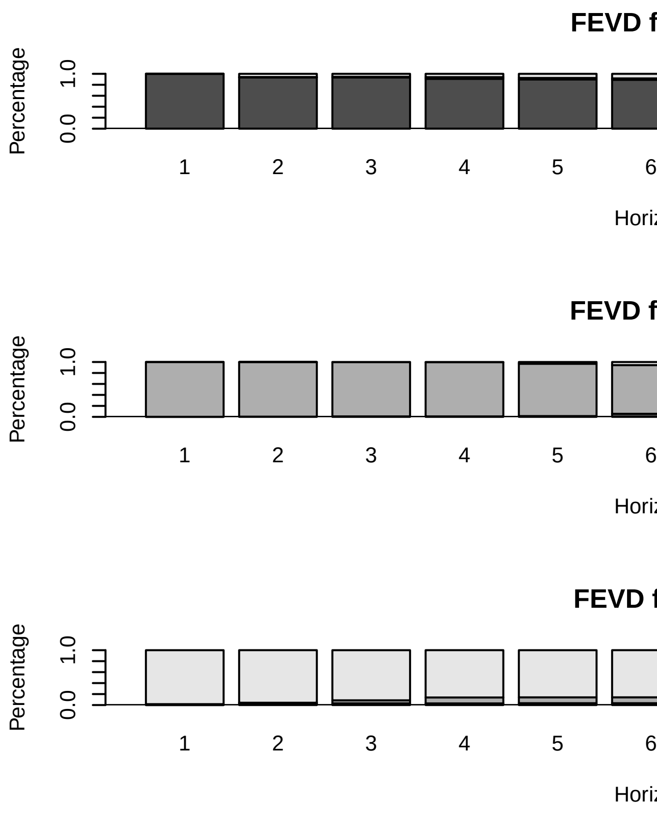
**VAR model for US states**

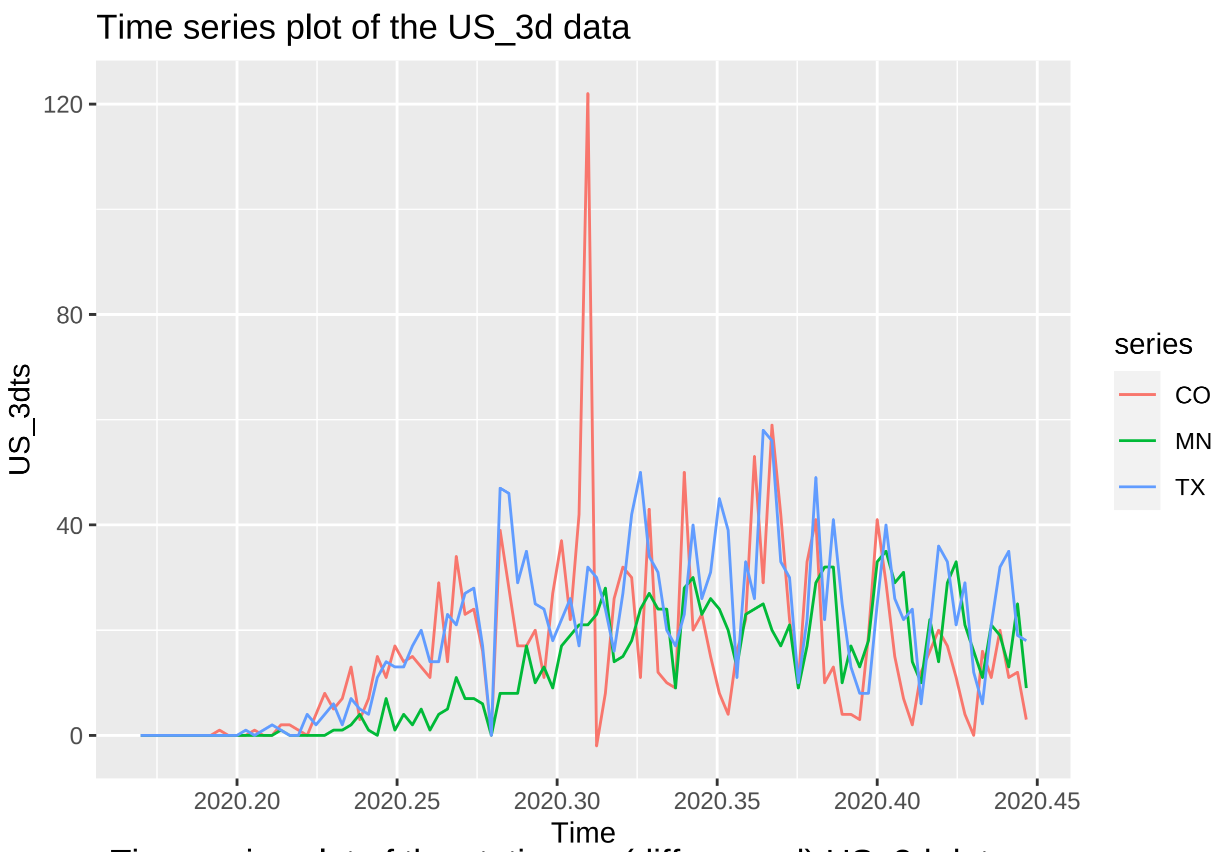


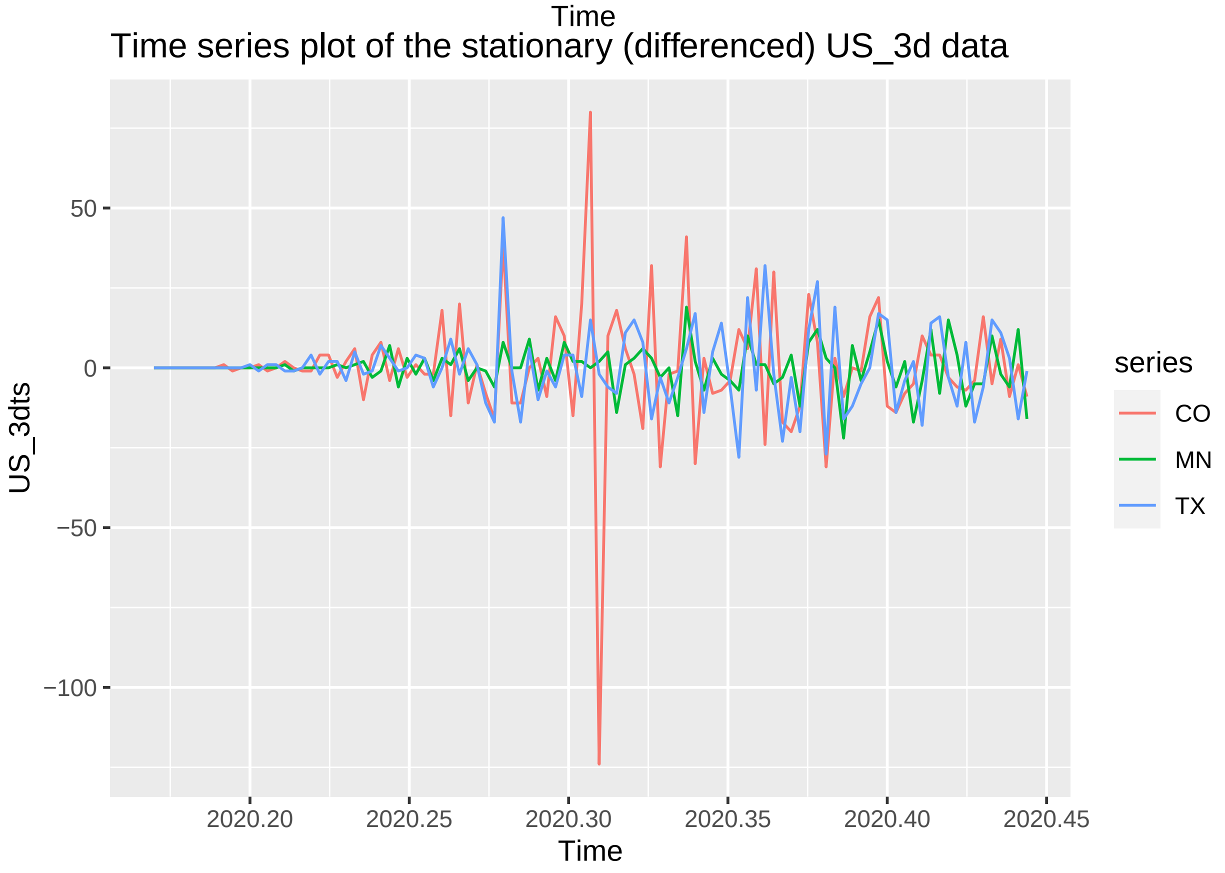
```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      7      3      1      7
##
## $criteria
##                   1            2            3            4            5
## AIC(n) 3.182822e+01 3.175276e+01 3.161973e+01 3.158999e+01 3.156188e+01
## HQ(n)  3.192779e+01 3.195189e+01 3.191843e+01 3.198827e+01 3.205972e+01
## SC(n)  3.207492e+01 3.224615e+01 3.235982e+01 3.257678e+01 3.279536e+01
## FPE(n) 6.650451e+13 6.170047e+13 5.408676e+13 5.263879e+13 5.140155e+13
##                   6            7            8            9           10
## AIC(n) 3.152479e+01 3.144900e+01 3.147180e+01 3.152030e+01 3.166426e+01
## HQ(n)  3.212221e+01 3.214598e+01 3.226835e+01 3.241642e+01 3.265994e+01
## SC(n)  3.300497e+01 3.317588e+01 3.344538e+01 3.374057e+01 3.413122e+01
## FPE(n) 4.985436e+13 4.664366e+13 4.831840e+13 5.155341e+13 6.078682e+13
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object var.US_3
## Chi-squared = 107.53, df = 81, p-value = 0.026
```



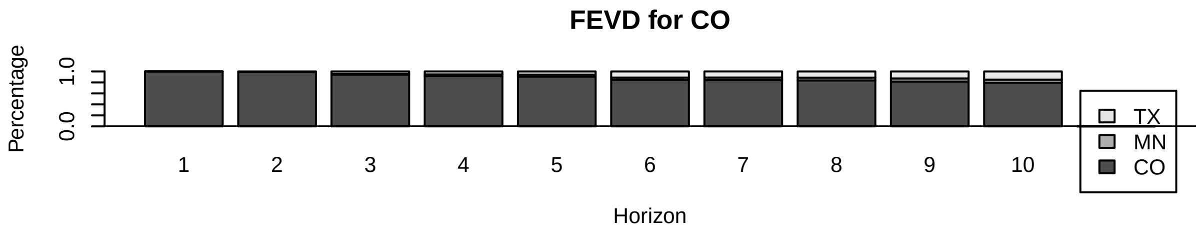


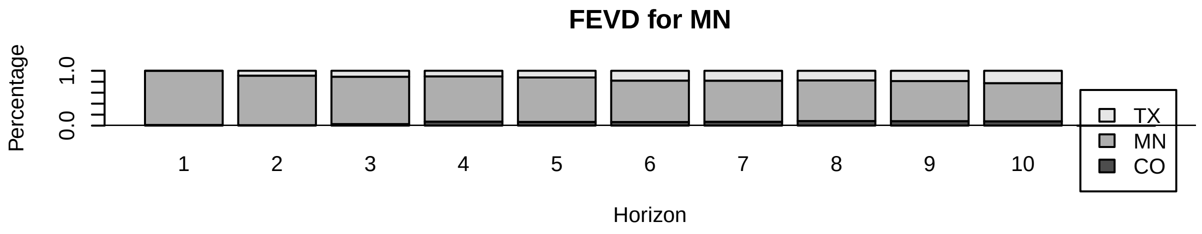

Time series plot of the US_3d data
US_3dts
Time
series
CO
MN
TX
120
80
40
0
2020.20
2020.25
2020.30
2020.35
2020.40
2020.45

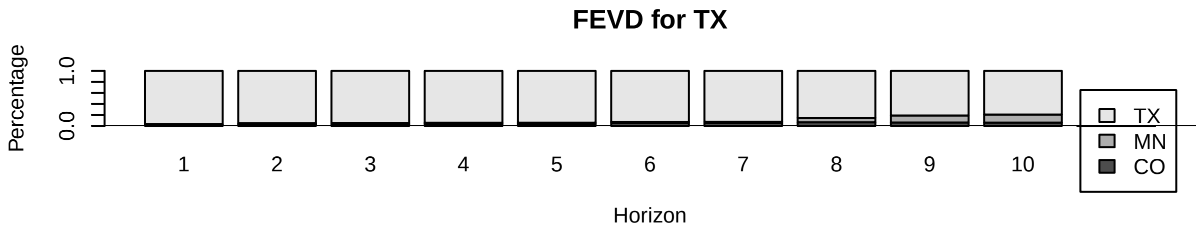

Time series plot of the stationary (differenced) US_3d data
US_3dts
Time
series
CO
MN
TX
50
0
−50
−100
2020.20
2020.25
2020.30
2020.35
2020.40
2020.45

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     10      4      2     10
##
## $criteria
##                   1            2            3            4            5
## AIC(n) 1.443372e+01 1.402610e+01 1.409547e+01 1.381925e+01 1.383510e+01
## HQ(n)  1.453391e+01 1.422646e+01 1.439603e+01 1.421999e+01 1.433602e+01
## SC(n)  1.468205e+01 1.452275e+01 1.484045e+01 1.481256e+01 1.507674e+01
## FPE(n) 1.855743e+06 1.235110e+06 1.325659e+06 1.008409e+06 1.029113e+06
##                   6            7            8            9           10
## AIC(n) 1.381228e+01 1.380511e+01 1.382273e+01     13.54496     13.24906
## HQ(n)  1.441339e+01 1.450640e+01 1.462421e+01     14.44662     14.25091
## SC(n)  1.530224e+01 1.554340e+01 1.580934e+01     15.77990     15.73233
## FPE(n) 1.012702e+06 1.015108e+06 1.046573e+06 806247.88521 612797.68863
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object var.US_3d
## Chi-squared = 105.3, df = 54, p-value = 3.692e-05
```

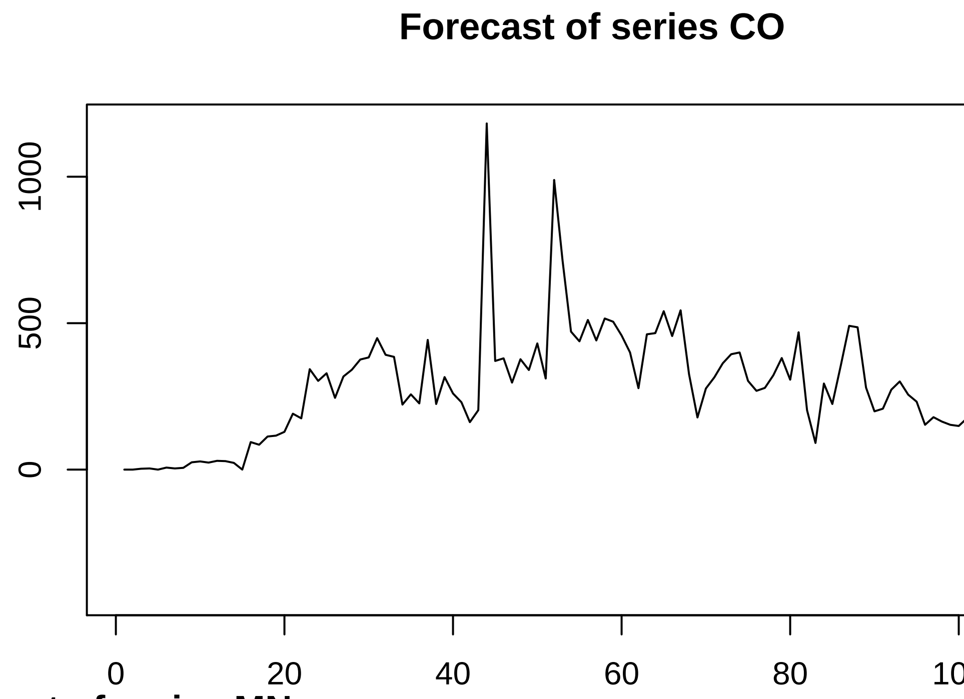




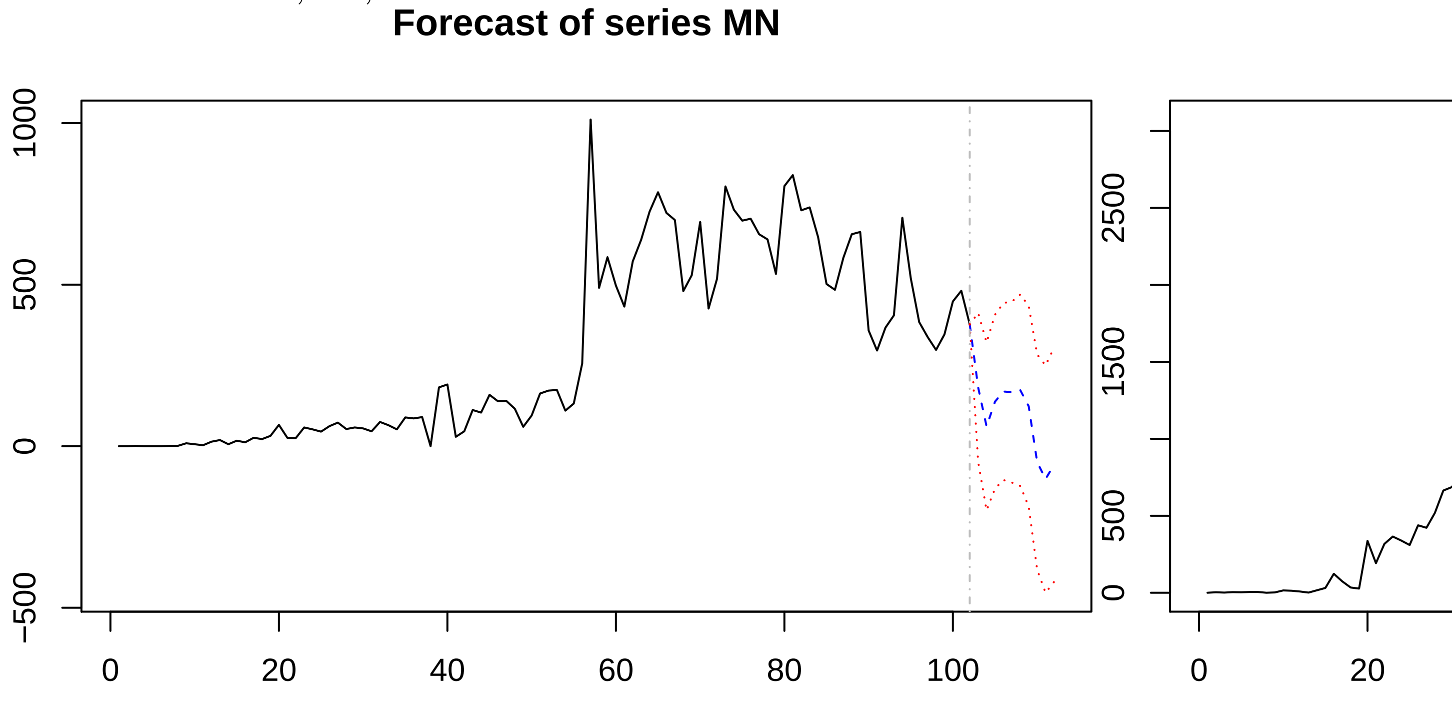


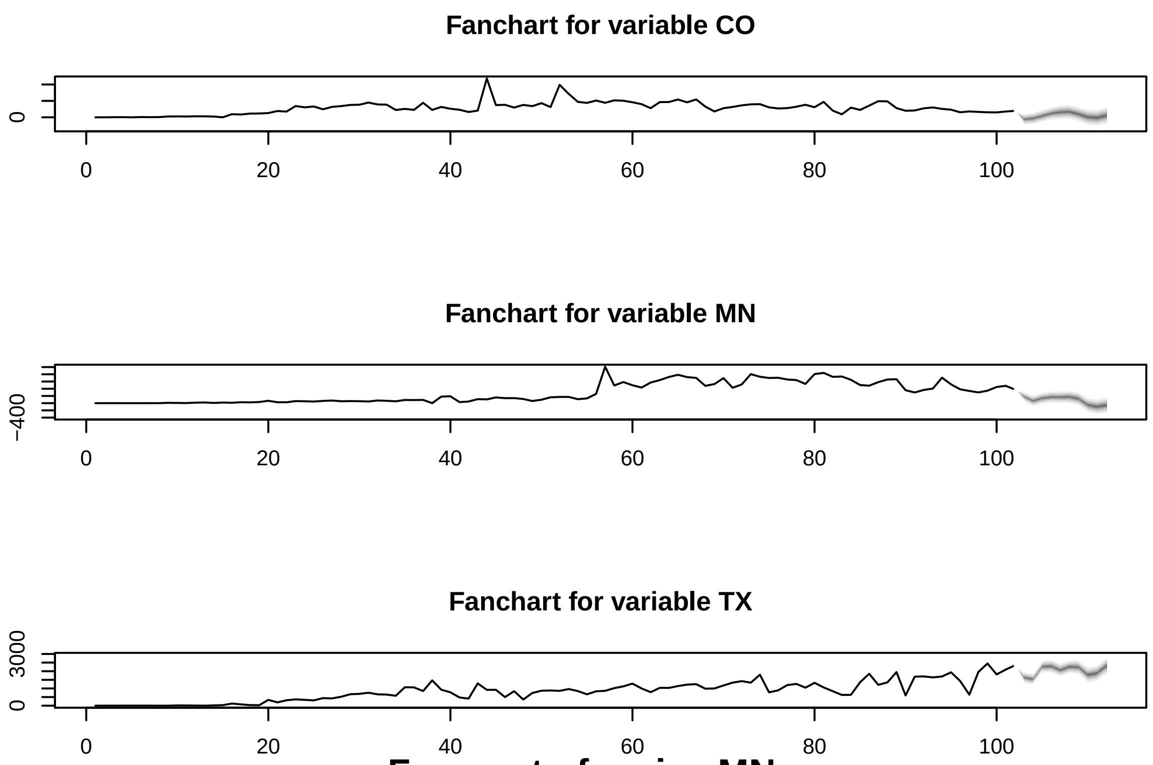
### Forecasting with Multivariate Models

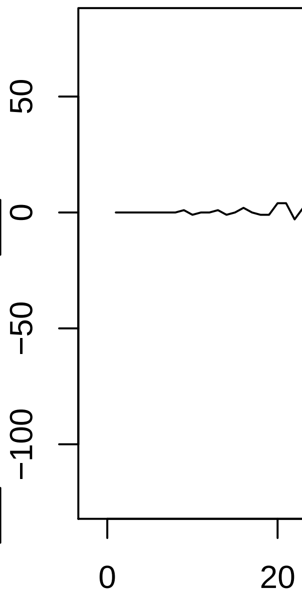
The forecasts for cases and deaths for each of the three identified states (CO, MN, TX) appeared to be relatively accurate with RMSE score between 0.39 and 0.55 for cases and 0.64 and 2.46 for deaths. Obviously, the VAR model predicted cases with more precision than it did deaths. In comparison, the ARIMA models for MN and CO had lower RMSE scores for both cases and deaths. ARIMA is also the simpler model, so it is probably the preferred choice for the state level data. VAR models, on the other hand, may be more effective at predicting longer term or if different combinations of states/counties were investigated. VAR models are also far more insightful when comparing one county to another.



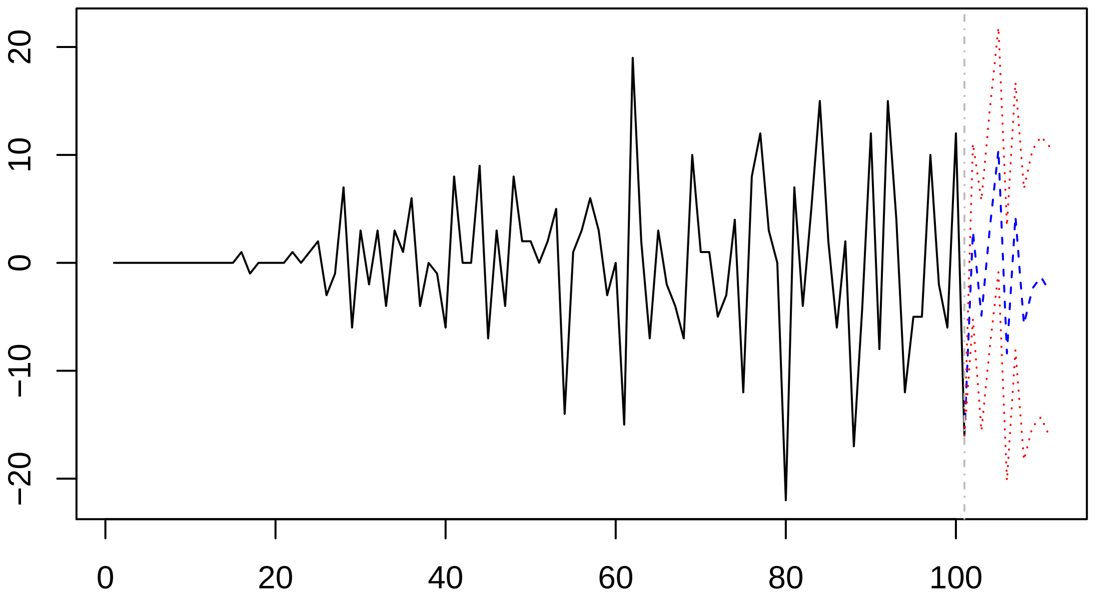
**VAR forecasts for CO, MN, and TX**

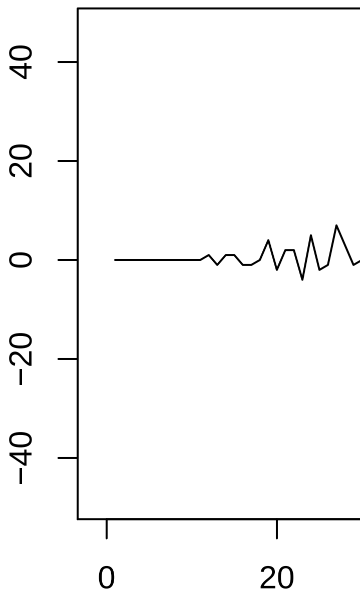


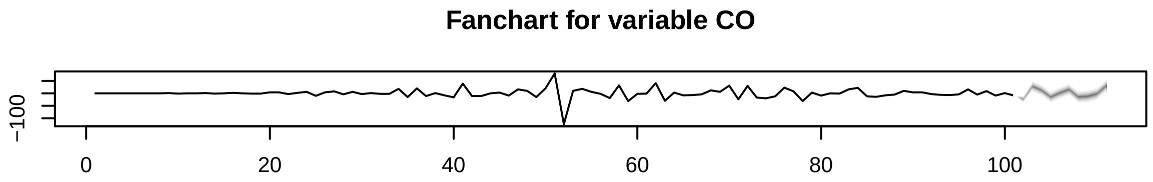


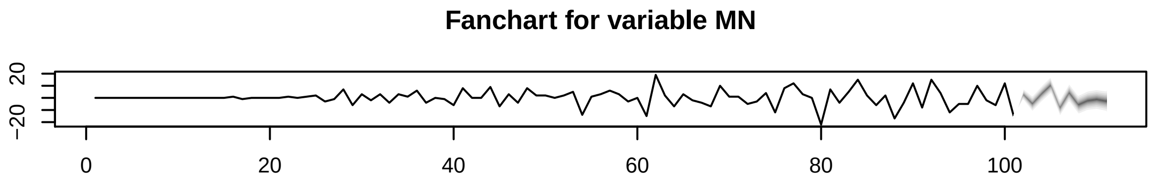


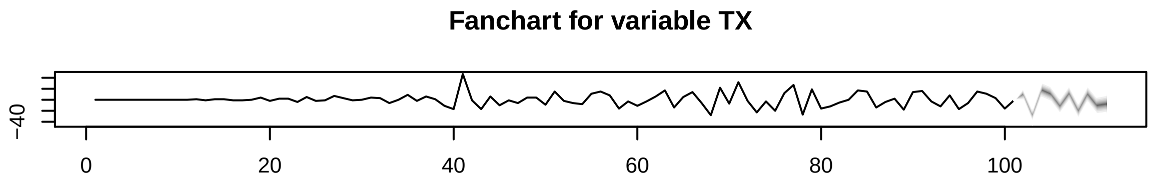
Forecast of series MN











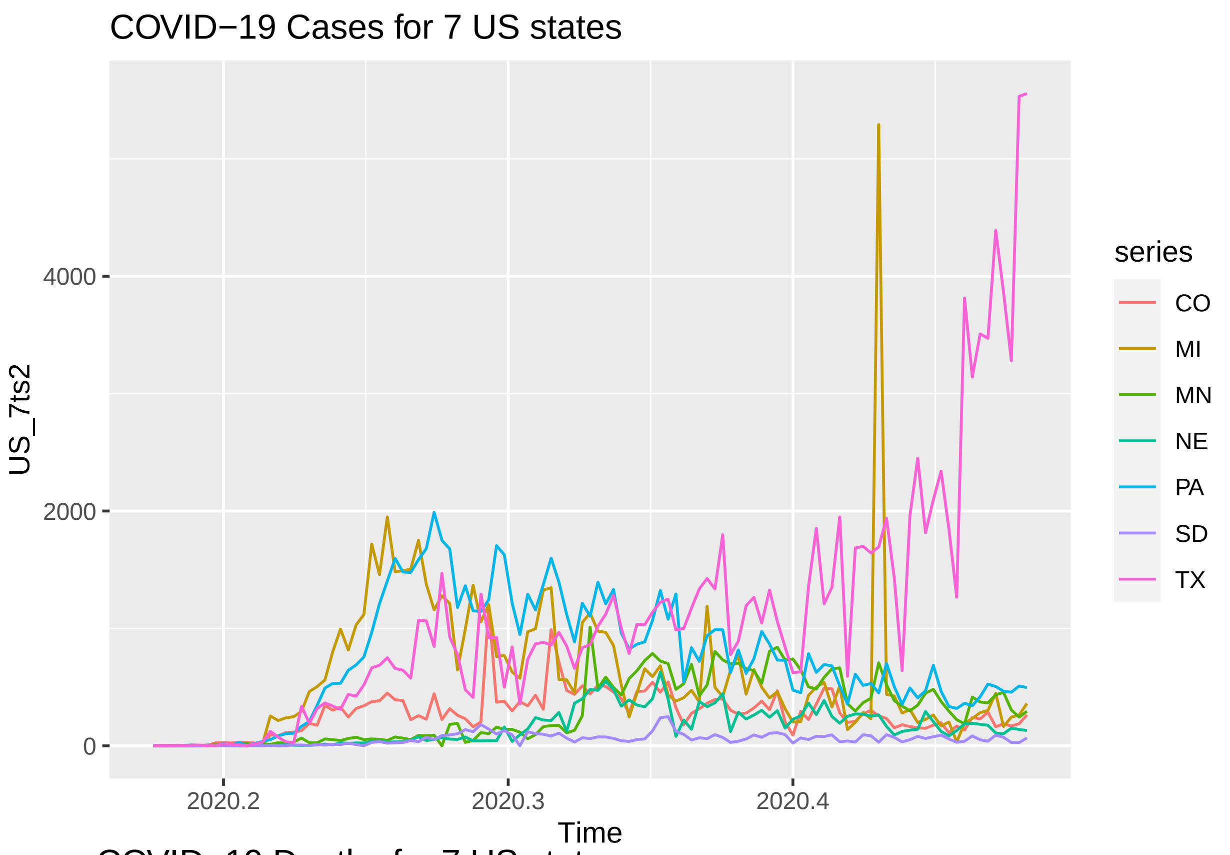
## Conclusions

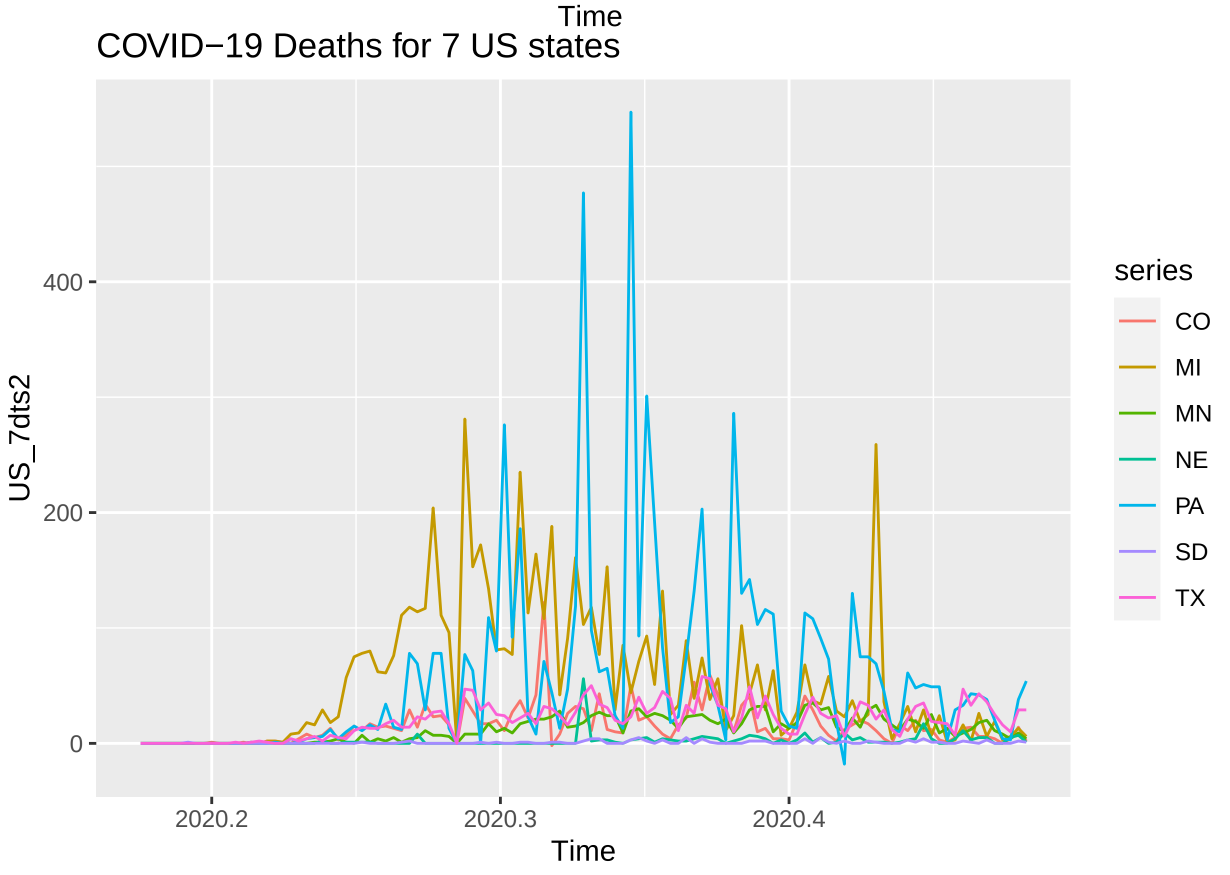
---

Is time series analysis in R an effective method for forecasting COVID cases and deaths? Ultimately, the answer depends on how far out we need to forecast and how accurate the models need to be. Based on my research and experimentation, ARIMA and VAR models can predict cases and deaths with a fair amount of accuracy for short periods of time in the future. They are also more accurate for time series with higher case and death counts, like the United States or state-level data. On the other hand, they are less accurate at the county level, probably because of the lower counts of cases and deaths and the limited number of records. Over time, these models will become more effective.

But why take my word for it? The following predictions for cases and deaths from 06.25.20 through 07.04.20 were made using ARIMA forecasting. I encourage you to compare these predictions to actual data and decide for yourself!

NOTE: to calculate your own normalized RMSE, create a vector for the actual cases or deaths data and use the following: > RMSE(fit$pred,actual)/mean(actual)


COVID−19 Cases for 7 US states
US_7ts2
4000
2000
0
2020.2
2020.3
2020.4
Time
series
CO
MI
MN
NE
PA
SD
TX


COVID−19 Deaths for 7 US states
US_7dts2
400
200
0
2020.2
2020.3
2020.4
Time
series
CO
MI
MN
NE
PA
SD
TX

```
## Time Series:
```

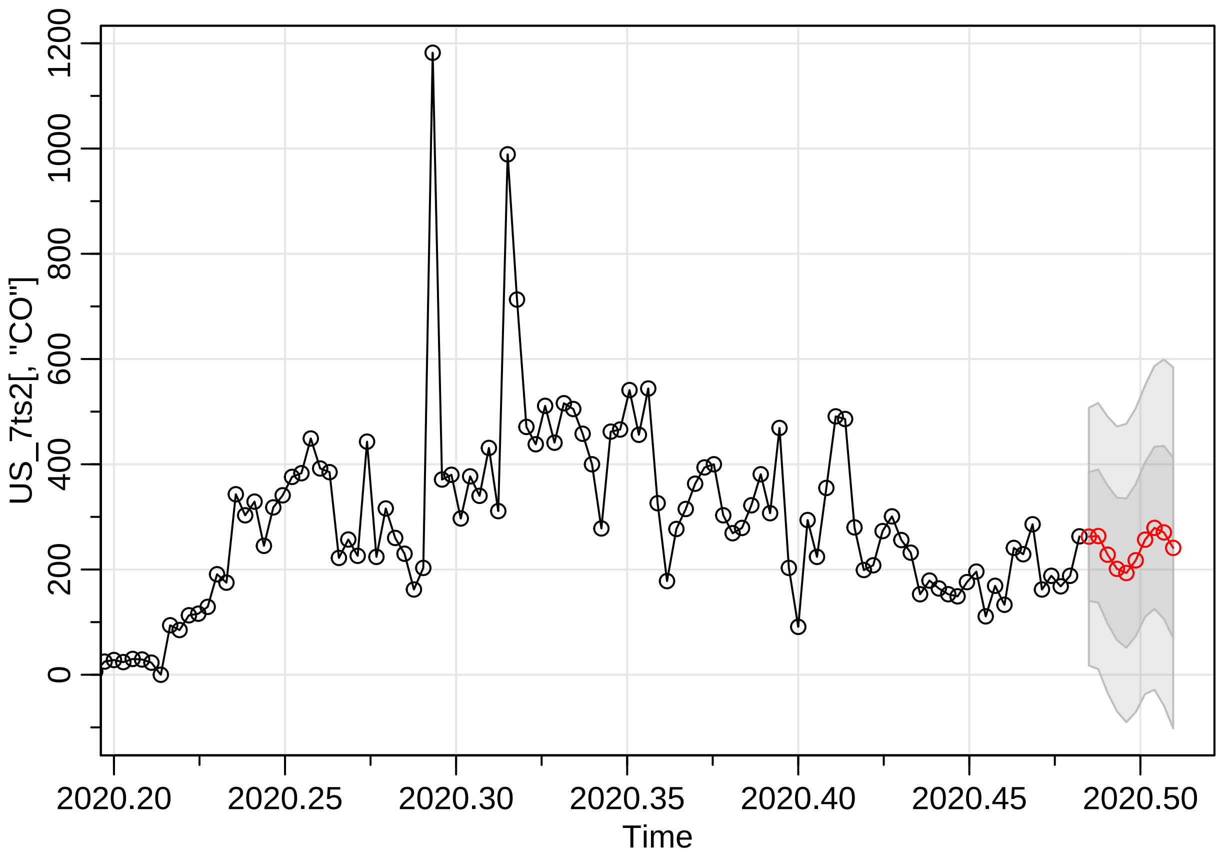


Figure 1: CO Cases

```
## Start = c(2020, 178)
## End = c(2020, 187)
## Frequency = 365
##  [1] 262.5499 263.5916 228.1819 201.2255 193.2711 217.5006 256.7161 279.1208
##  [9] 270.5932 241.0802
```

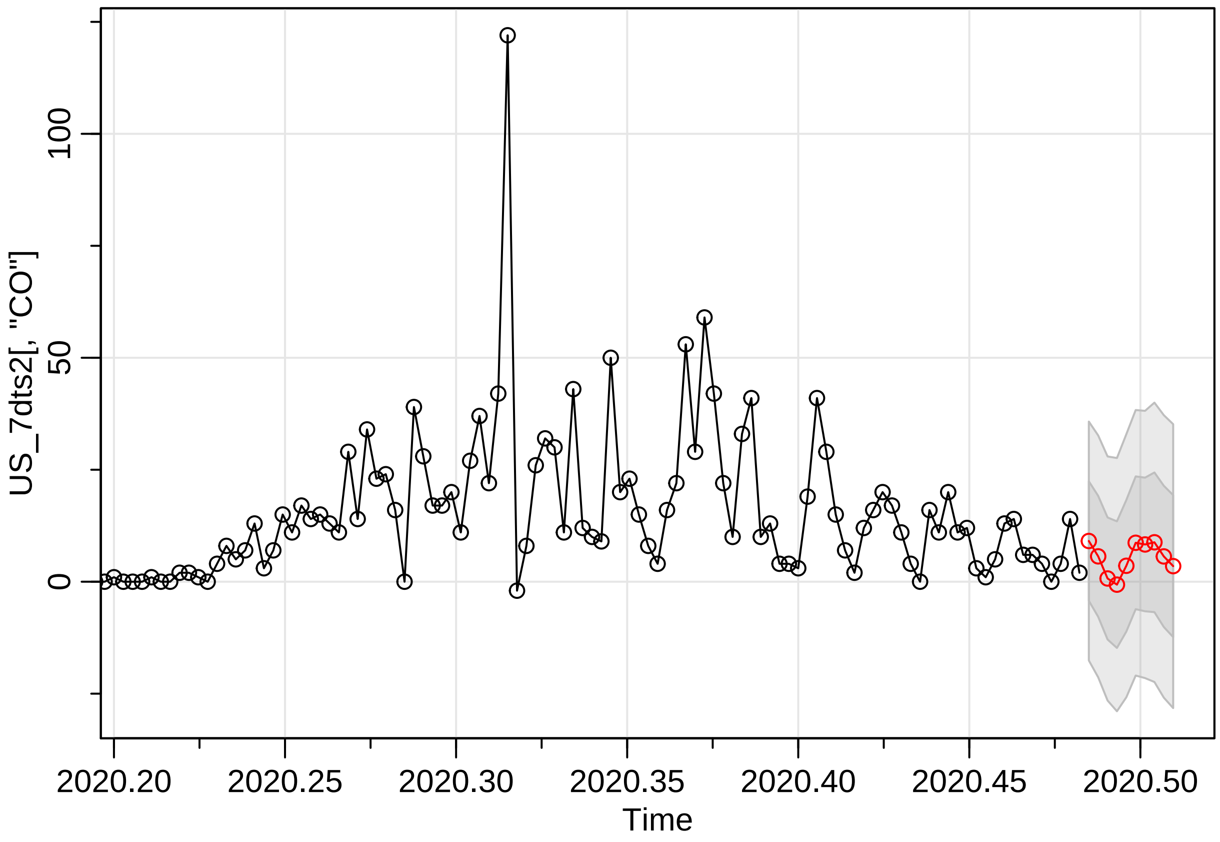


Figure 2: CO Deaths

```
## Time Series:
## Start = c(2020, 178)
## End = c(2020, 187)
## Frequency = 365
##  [1]  9.0875560  5.6590338  0.7176019 -0.6552971  3.5564503  8.6917856
##  [7]  8.3167922  8.7985093  5.6477377  3.4760598
```

```
## Time Series:
## Start = c(2020, 178)
## End = c(2020, 187)
## Frequency = 365
##  [1] 355.6022 370.2079 383.6680 396.0721 407.5032 418.0377 427.7457 436.6922
##  [9] 444.9369 452.5348
```

```
## Time Series:
## Start = c(2020, 178)
## End = c(2020, 187)
## Frequency = 365
##  [1] 15.82806 14.64134 16.89526 14.52507 14.65370 14.78232 14.91095 15.03958
##  [9] 15.16821 15.29684
```

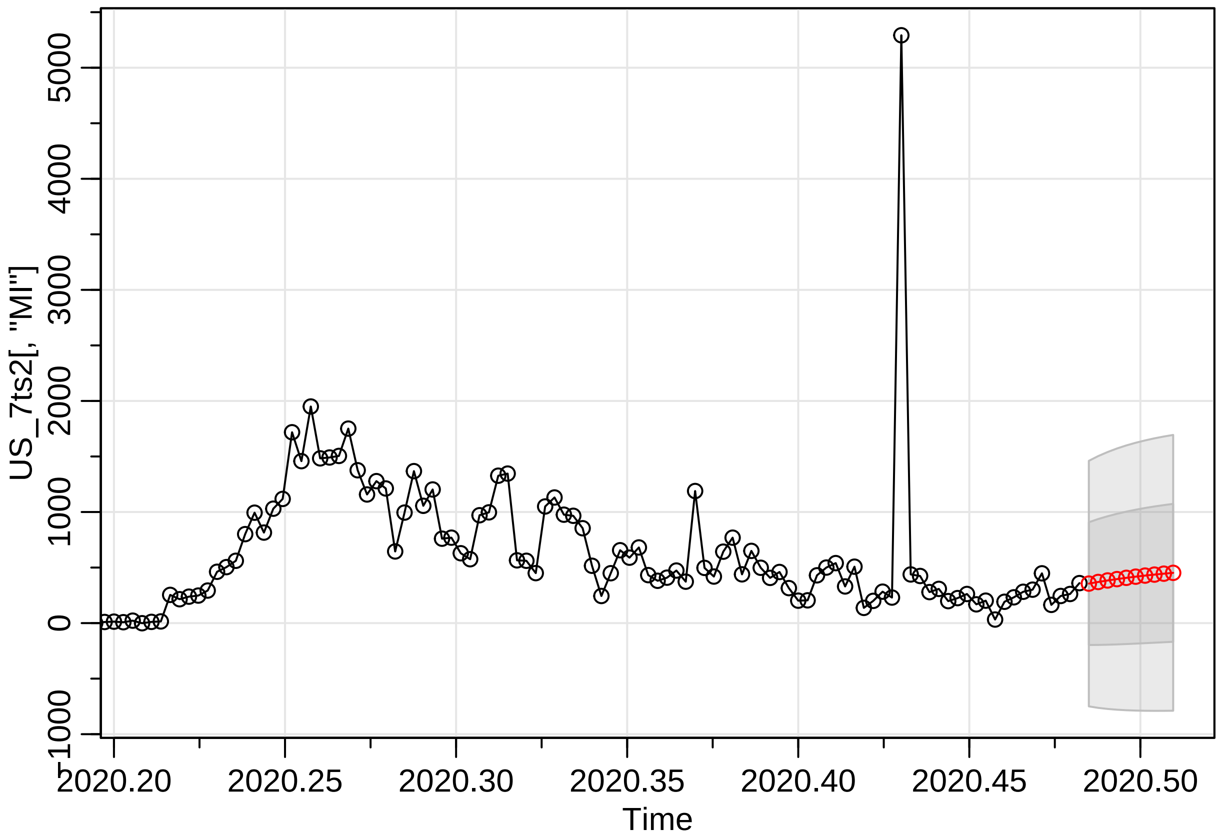


Figure 3: MI Cases

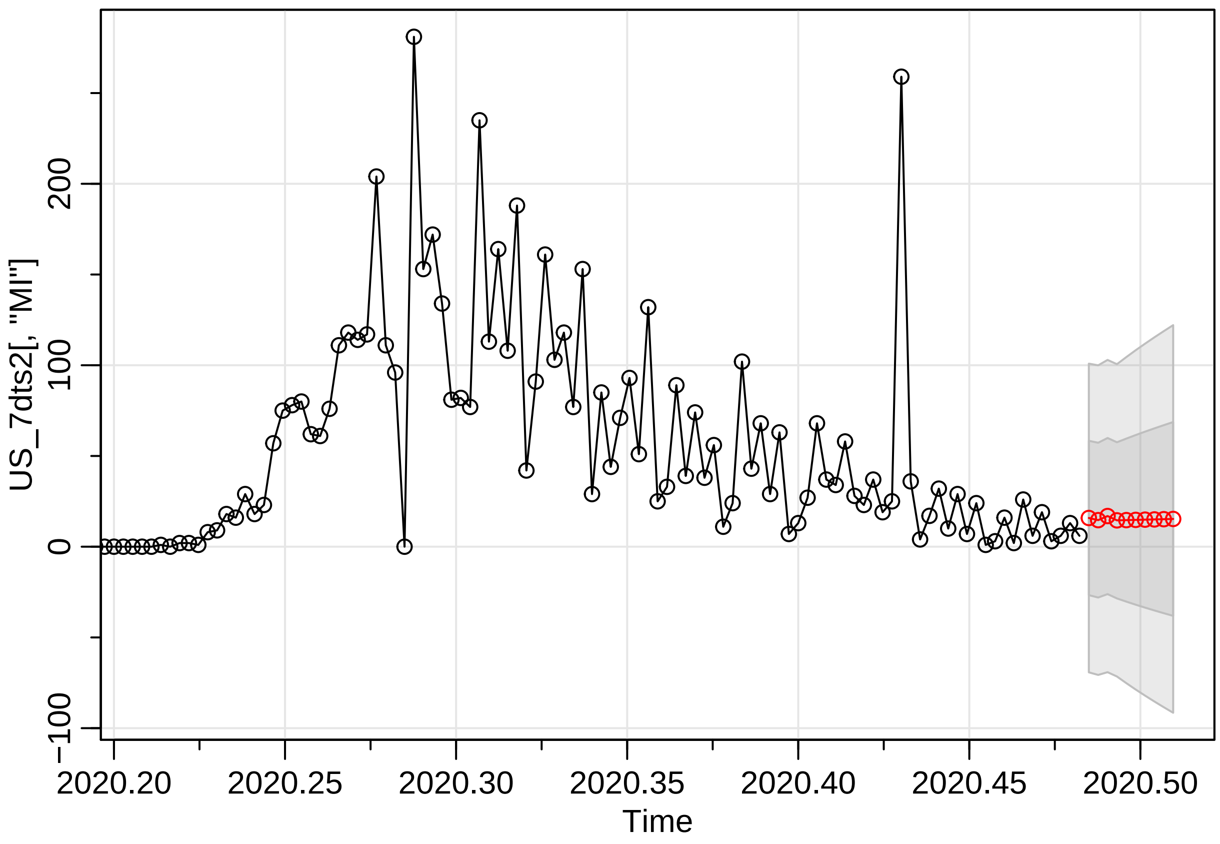


Figure 4: MI Deaths

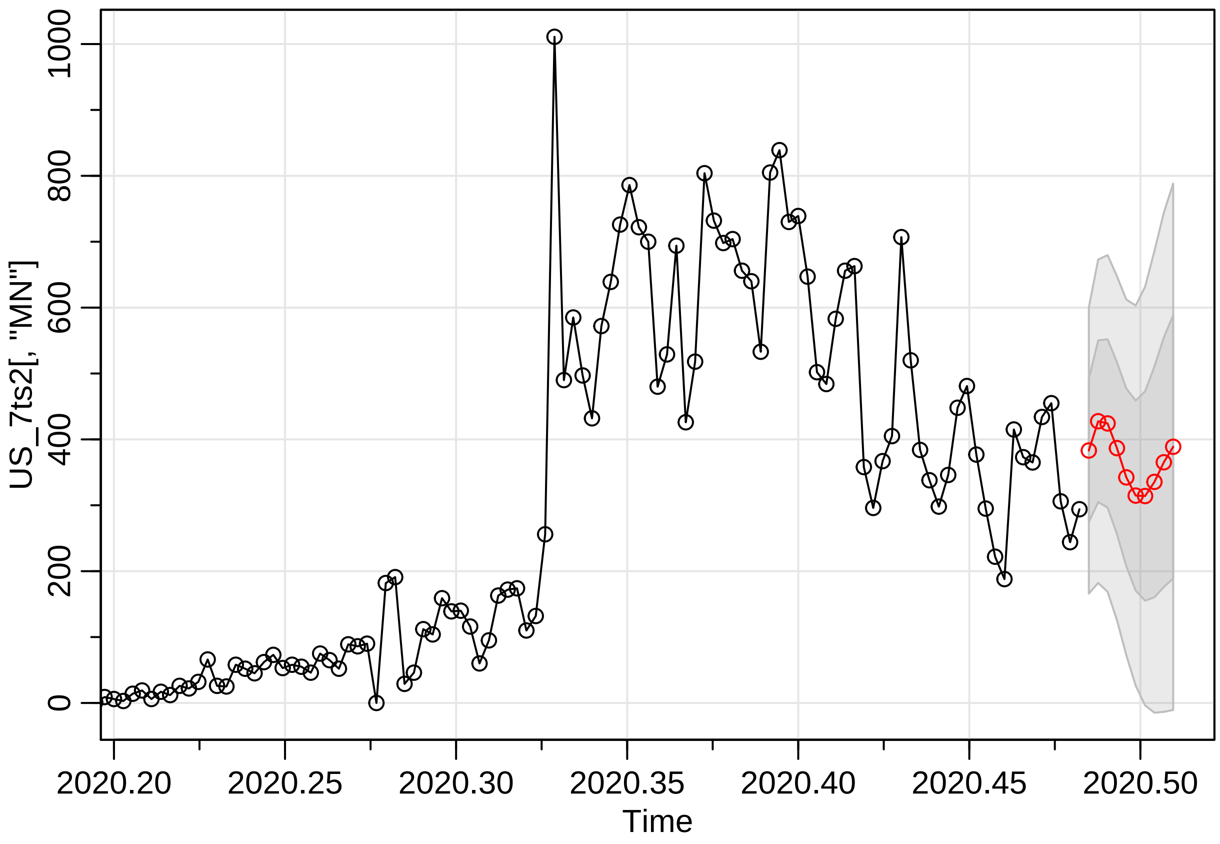


Figure 5: MN Cases

```
## Time Series:
## Start = c(2020, 178)
## End = c(2020, 187)
## Frequency = 365
##  [1] 383.1432 427.5698 424.2359 386.7671 342.3544 314.6995 314.0190 335.5214
##  [9] 365.2605 388.8272
```

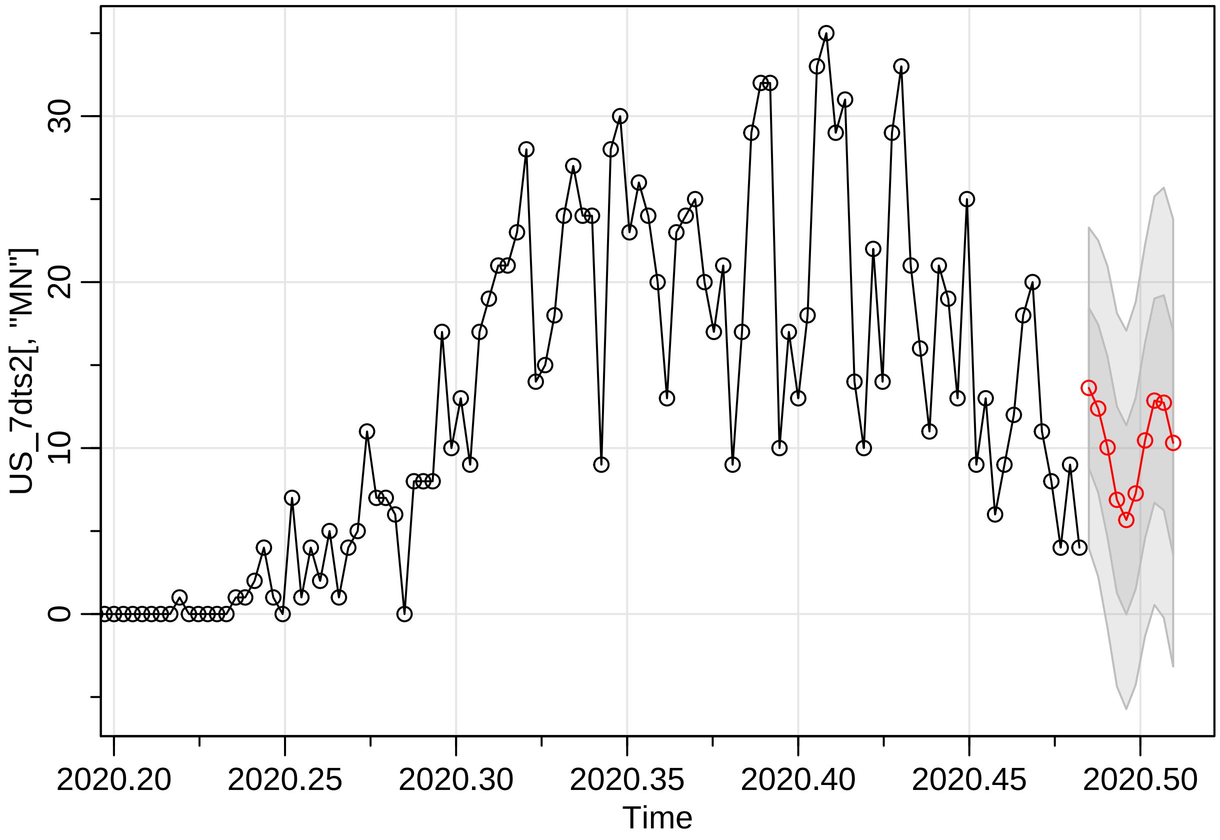


Figure 6: MN Deaths

```
## Time Series:
## Start = c(2020, 178)
## End = c(2020, 187)
## Frequency = 365
##  [1] 13.625486 12.382113 10.039815  6.882380  5.668594  7.265950 10.461590
##  [8] 12.859009 12.733426 10.311381
```

```
## Time Series:
## Start = c(2020, 178)
## End = c(2020, 187)
## Frequency = 365
##  [1] 142.8903 140.9784 116.9753 113.7361 139.2713 132.1004 126.4372 144.0084
##  [9] 145.3233 131.4189
```

```
## Time Series:
## Start = c(2020, 178)
## End = c(2020, 187)
## Frequency = 365
##  [1] 4.784640 4.828680 4.872721 4.916761 4.960801 5.004842 5.048882 5.092923
```

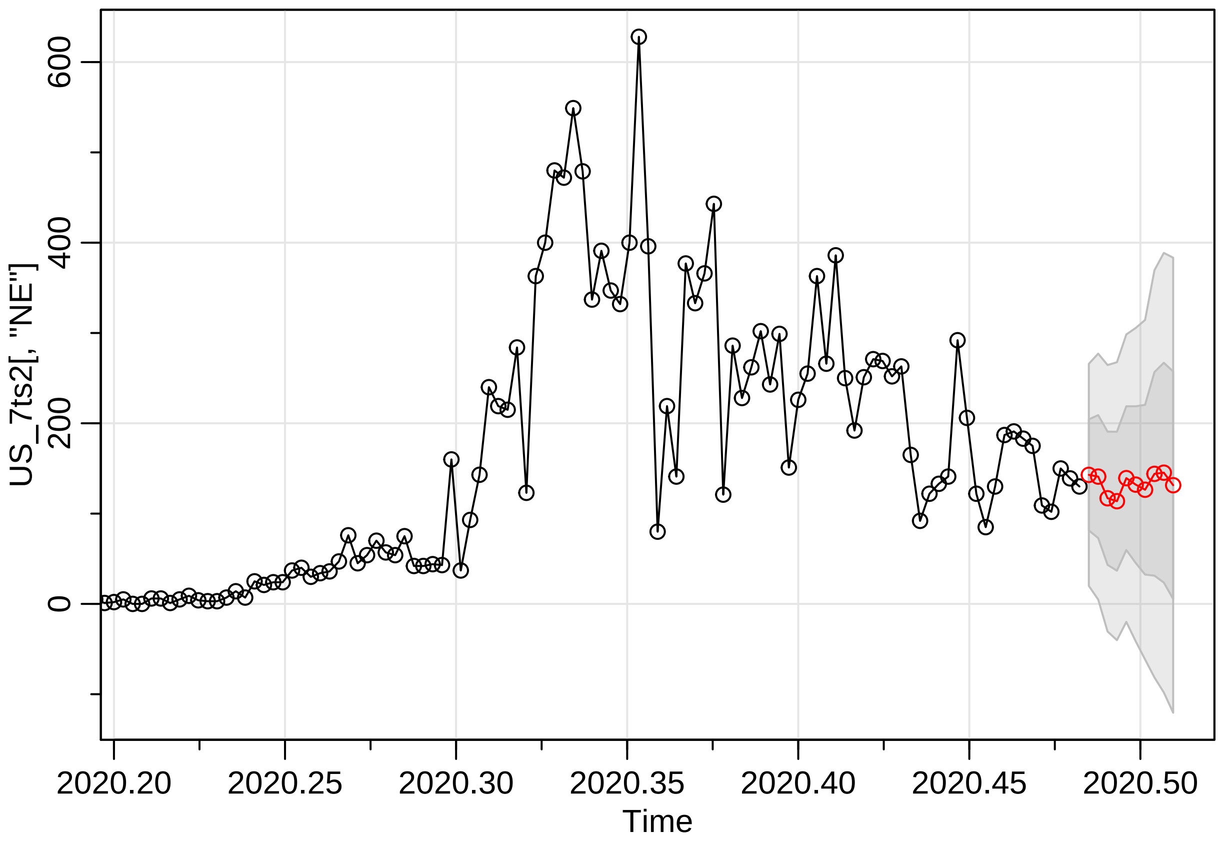


Figure 7: NE Cases

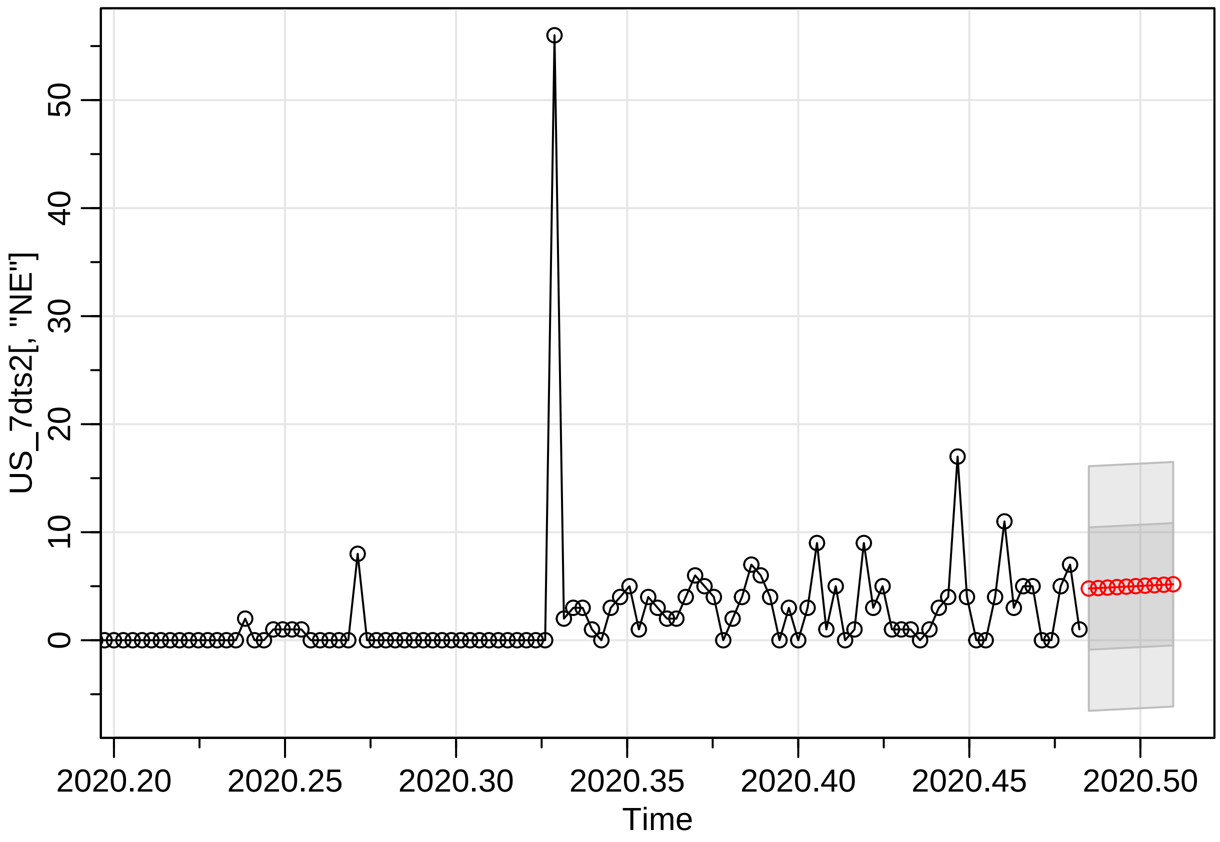


Figure 8: NE Deaths

```
##  [9] 5.136963 5.181004
```

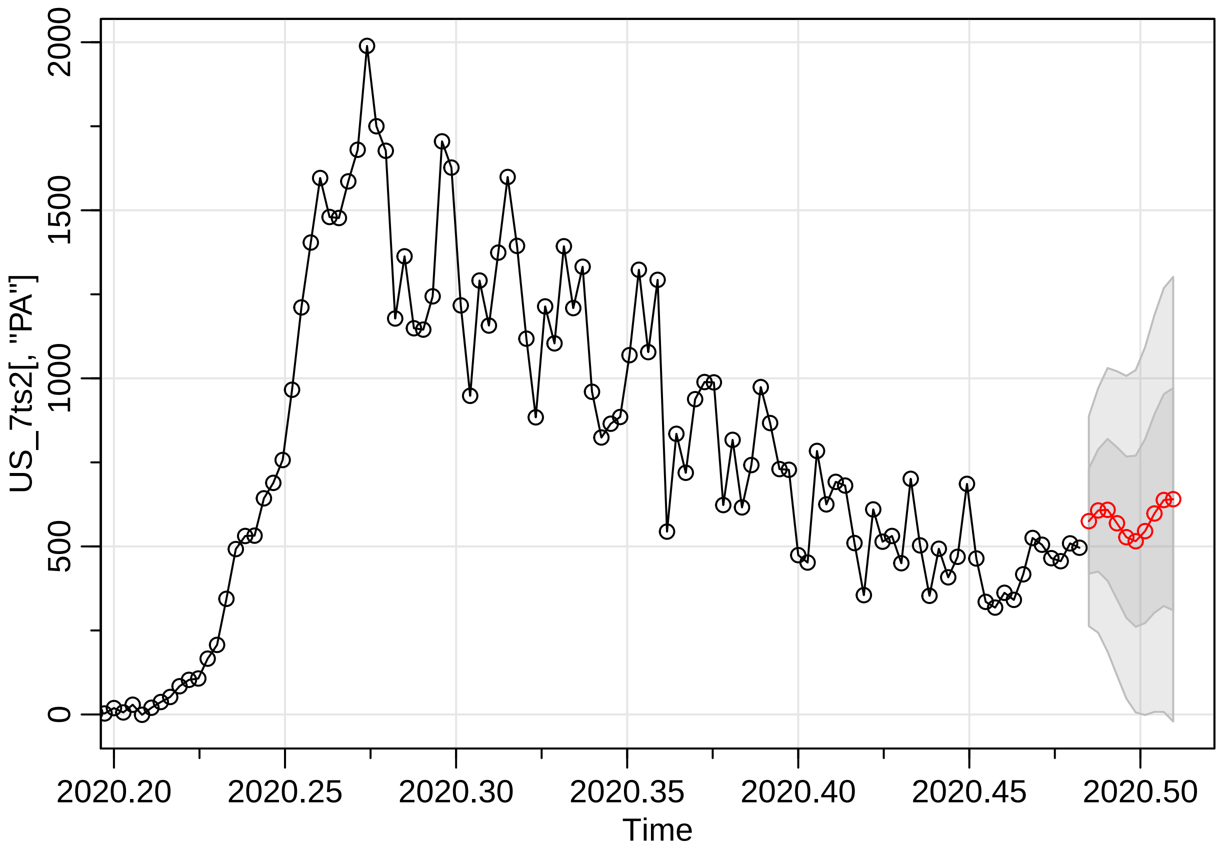


Figure 9: PA Cases

```
## Time Series:
## Start = c(2020, 178)
## End = c(2020, 187)
## Frequency = 365
##  [1] 575.1170 606.9556 608.8401 568.9204 527.3963 515.3532 545.4947 597.9499
##  [9] 637.9604 640.4923
```

```
## Time Series:
## Start = c(2020, 178)
## End = c(2020, 187)
## Frequency = 365
##  [1] 38.94207 33.03394 24.82736 23.17334 22.00762 21.19500 20.63770 20.26503
##  [9] 20.02587 19.88326
```

```
## Time Series:
## Start = c(2020, 178)
## End = c(2020, 187)
## Frequency = 365
##  [1] 53.02280 58.45148 70.16992 63.85358 50.66747 58.36565 59.13038 57.32042
##  [9] 61.32302 65.16218
```

```
## Time Series:
## Start = c(2020, 178)
## End = c(2020, 187)
```

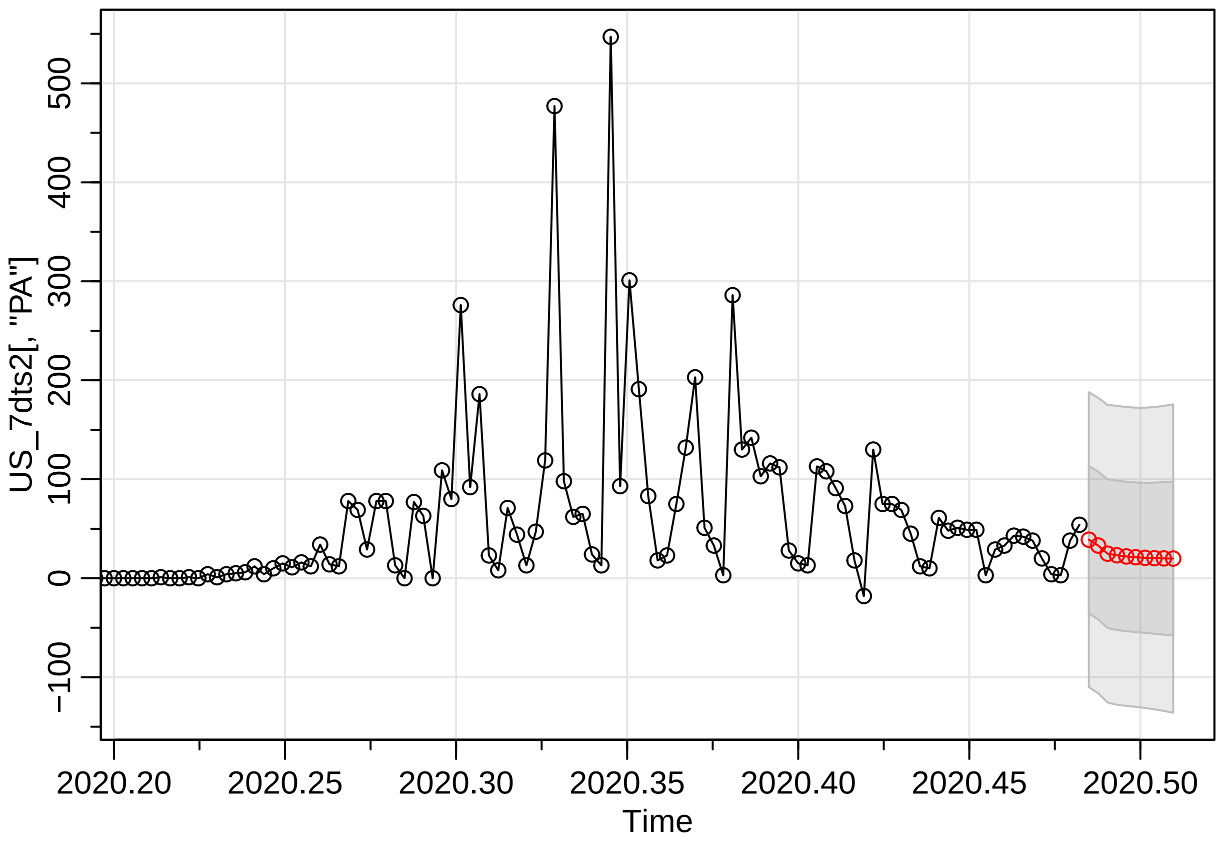


Figure 10: PA Deaths

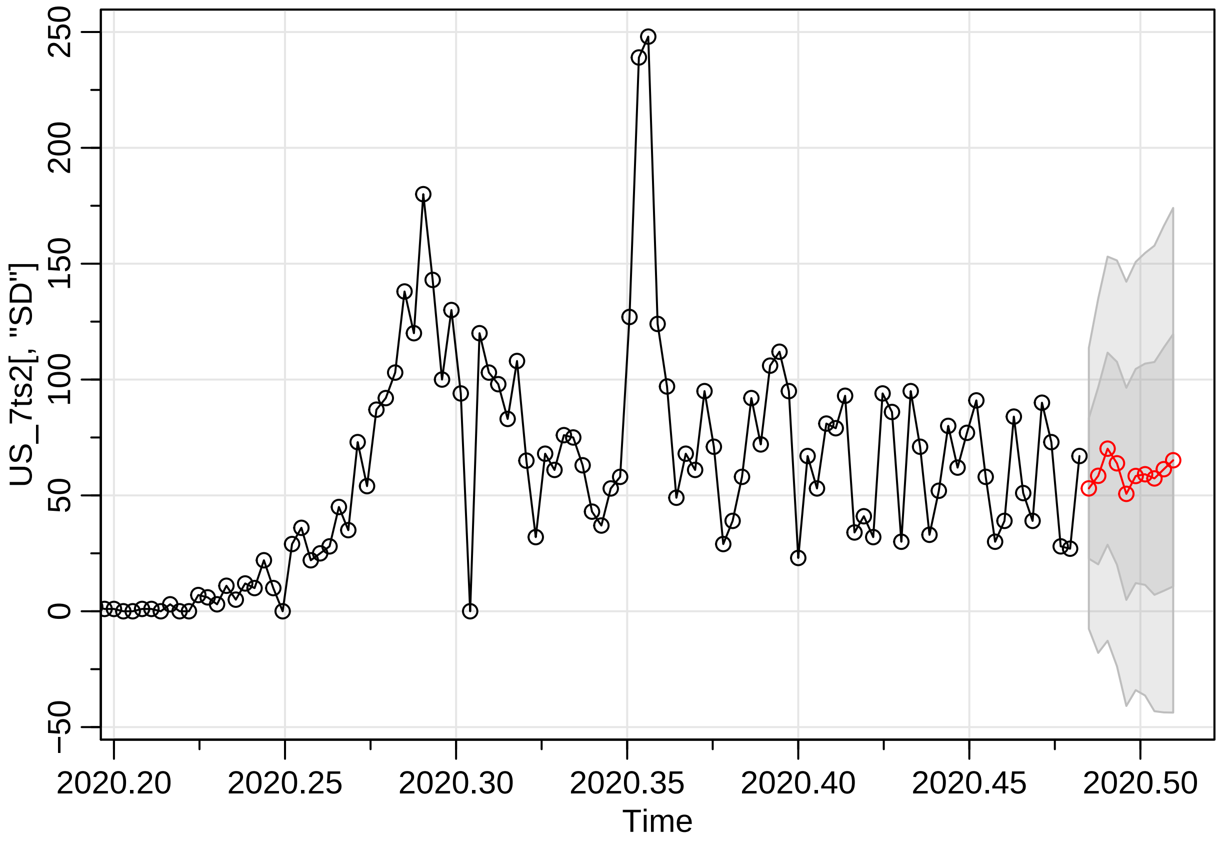


Figure 11: SD Cases

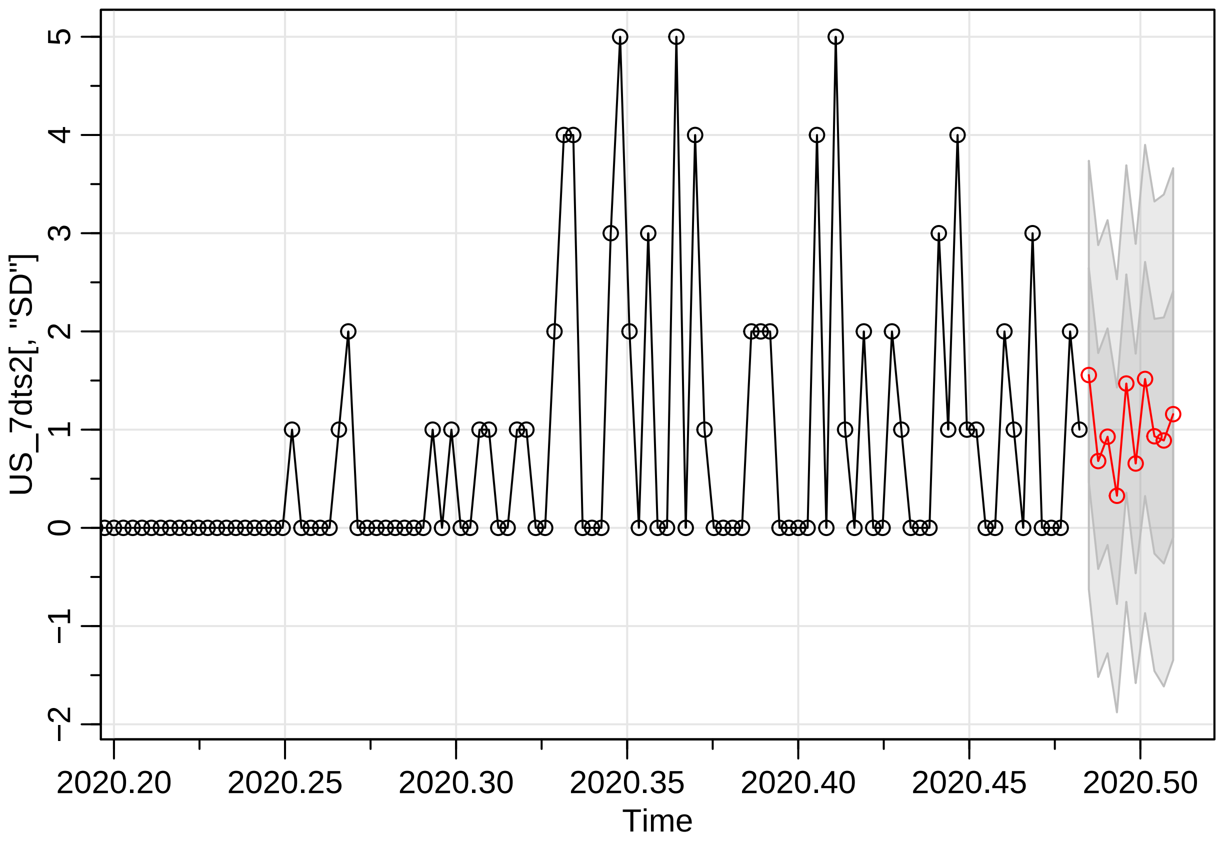


Figure 12: SD Deaths

```
## Frequency = 365
##  [1] 1.5559155 0.6800646 0.9282528 0.3264336 1.4694583 0.6550545 1.5161230
##  [8] 0.9325457 0.8895027 1.1576926
```

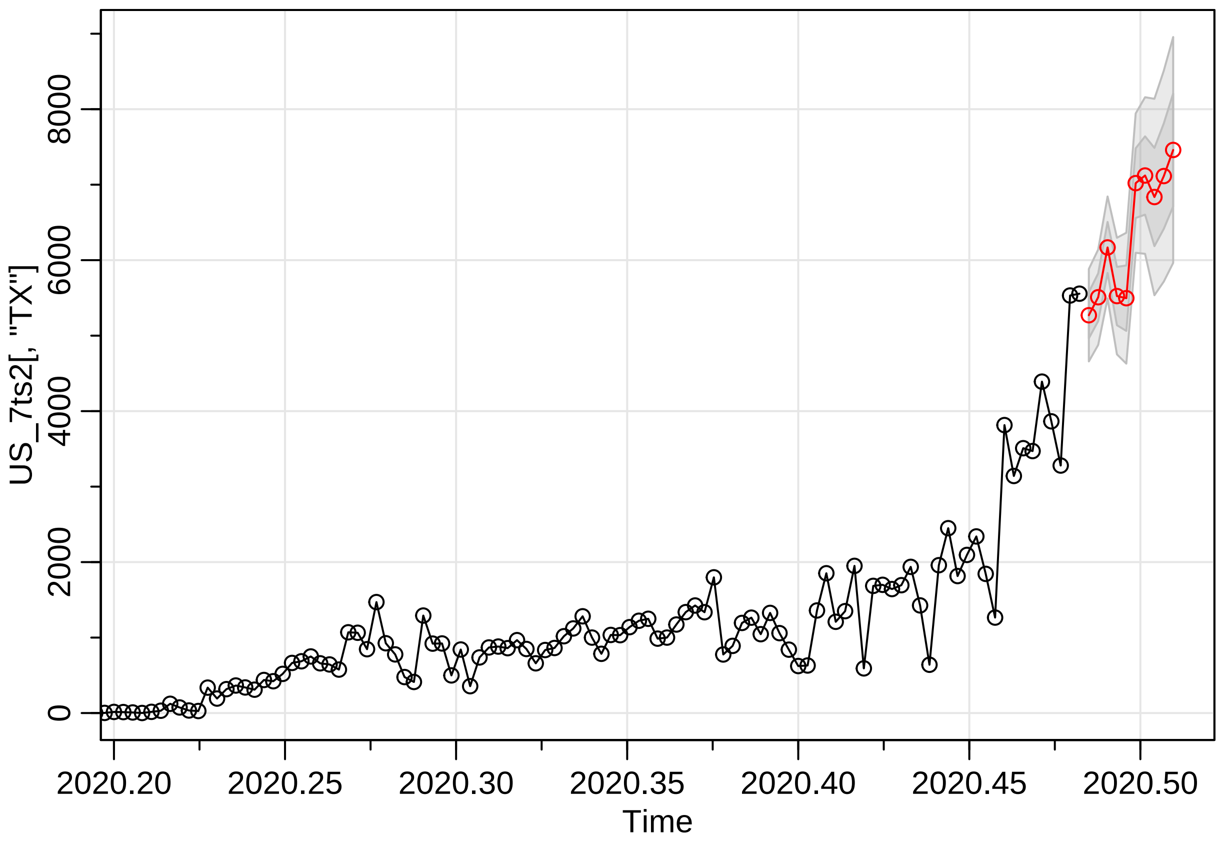


Figure 13: TX Cases

```
## Time Series:
## Start = c(2020, 178)
## End = c(2020, 187)
## Frequency = 365
##  [1] 5270.443 5509.166 6168.777 5524.268 5496.354 7020.696 7121.484 6836.454
##  [9] 7114.152 7460.074
```

```
## Time Series:
## Start = c(2020, 178)
## End = c(2020, 187)
## Frequency = 365
##  [1] 34.64500 34.88198 18.54188 12.20248 16.04625 25.87062 35.13658 33.77441
##  [9] 28.63990 20.06938
```

## References

https://cran.r-project.org/web/packages/tmap/tmap.pdf https://cran.r-project.org/web/packages/vars/vars.pdf https://data.census.gov/cedsci/table?g=0400000US27.050000&layer=VT_2018_040_00_PY_D1&y=2018&tid=ACSST5Y2018.S1603&t=Language%20Spoken%20at%20Home&vintage=2018&hidePreview=false&cid=S1601_C01_001E https://data.nber.org/data/county-

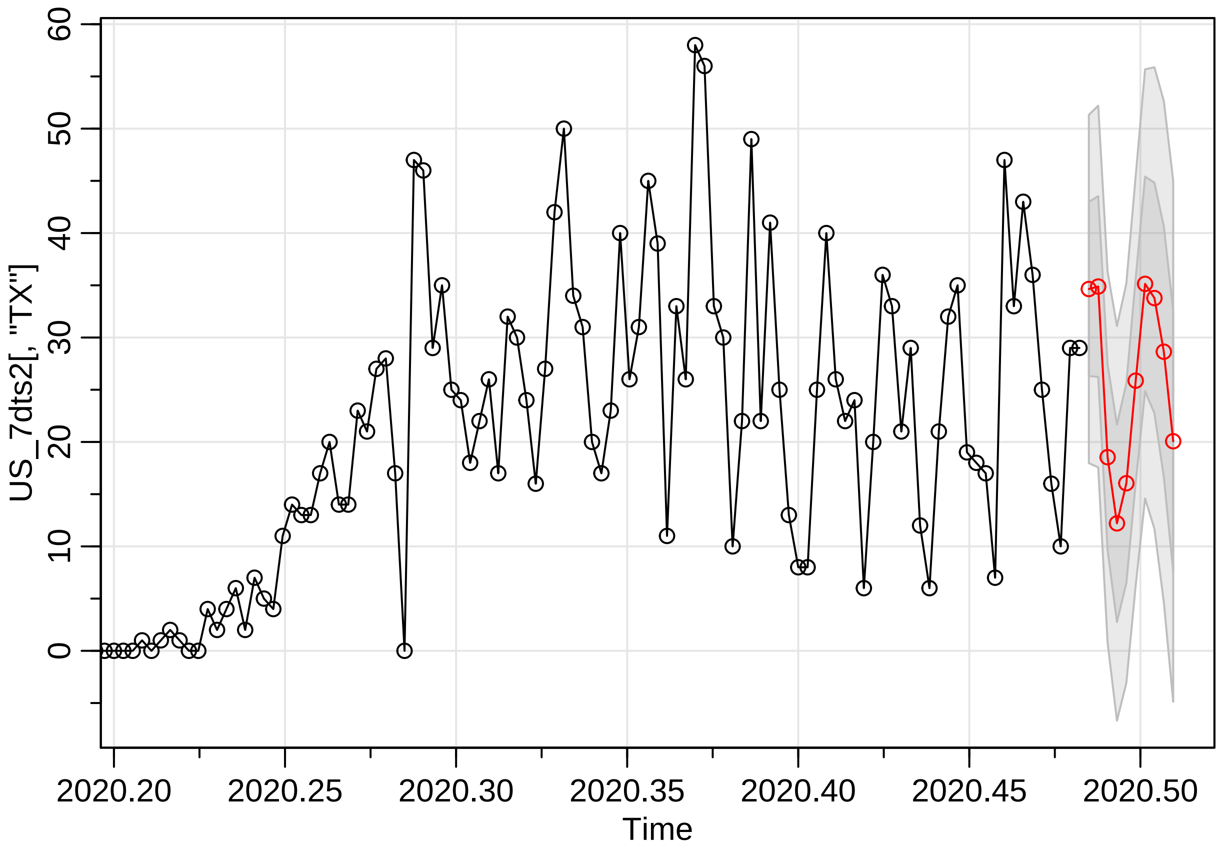


Figure 14: TX Deaths

distance-database.html https://datascienceplus.com/time-series-analysis-using-arima-model-in-r/ https://earth.esa.int/documents/10174/1573054/Factors_that_have_an_influence_on_time_series.pdf https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average https://www.ewg.org/news-and-analysis/2020/05/ewg-map-counties-meatpacking-plants-report-twice-national-average-rate http://www.healthdata.org/us-county-profiles https://www.itl.nist.gov/div898/handbook/pmc/section4/pm c451.htm https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge/tasks?taskId=558 https://www.kaggle.com/c/covid19-global-forecasting-week-5/data https://www.kaggle.com/imd evskp/corona-virus-report https://kevinkotze.github.io/ts-7-tut/ https://machinelearningmaste ry.com/time-series-forecasting/ https://www.medrxiv.org/content/10.1101/2020.04.17.20069237v1 https://www.medrxiv.org/content/10.1101/2020.04.17.20069237v1.full.pdf https://orca-mwe.cf.ac.uk /62788/1/'Horses%20for%20Courses'%20in%20demand%20forecasting.pdf https://otexts.com/fpp2/ http://past.rinfinance.com/agenda/2013/talk/RueyTsay.pdf http://people.duke.edu/~rnau/411flow.gif https://www.r-econometrics.com/timeseries/varintro/ https://rstudio-pubs-static.s3.amazonaw s.com/274358_9fbc895fea2b443aaa60ad1a75c75687.html https://simplemaps.com/data/us-cities https://www.statista.com/statistics/1103185/cumulative-coronavirus-covid19-cases-number-us-by-day/ https://www.statmethods.net/advstats/timeseries.html https://www.statmethods.net/stats/regression.html https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781783552078/1/ch01lvl1 sec08/multivariate-time-series-analysis https://thefern.org/2020/04/mapping-covid-19-in-meat-and-food-processing-plants/ https://towardsdatascience.com/top-5-r-resources-on-covid-19-coronavirus-1d4c8df6d85f https://www.tutorialspoint.com/r/r_time_series_analysis.htm https://usafacts.org/visualizations/co ronavirus-covid-19-spread-map/ https://www.usnews.com/news/healthiest-communities/articles/2020-05-01/cdc-nearly-5-000-meat-plant-workers-infected-by-coronavirus https://www.wired.com/story/why-meatpacking-plants-have-become-covid-19-hot-spots/ http://zevross.com/blog/2018/10/02/creating-beautiful-demographic-maps-in-r-with-the-tidycensus-and-tmap-packages/