COVID-19 Excess Mortality Analysis

Code

Background

Course: BUS4469 - Competing with Analytics - Group Project

Topic: COVID-19 Excess Mortality Analysis

Description: What was the impact of COVID-19 on causes of death in Ontario and our recommendations for the future

Submission: December 8th, 2021

Project Team: 9 (Adam Miller, Anna Larkin, Dalton McPhaden, Justin Voronoff, Nir Oyberman)

Introduction

The team will be investigating excess mortality in Ontario during the 2020-2021 COVID-19 pandemic. We will utilize a Statistics Canada dataset of weekly causes of mortality. We hope to understand how many additional deaths COVID-19 has caused directly and indirectly, and how it affected mortality due to other causes. Analyzing this data could provide interesting implications regarding the severity of COVID-19. For example, we may find that a strained healthcare system dealing with the pandemic resulted in higher deaths in other causes, or perhaps lockdowns due to COVID-19 decreased deaths due to accidents. This pandemic has been the greatest, most damaging public health crisis to impact people globally in the 21st century. By understanding how COVID-19 affected mortality, we can get a better idea of the true impact of the pandemic.

Questions we would like to understand:

- How many people were expected to die in the province of Ontario during the COVID-19 time period?
- How many people actually died in the province of Ontario during the COVID-19 time period?
- How many extra people died during the COVID-19 time period? (i.e. actual deaths compared to forecast for that period)
- How many of these deaths are due to the COVID-19 virus itself?
- How many of these deaths are due to other causes; did COVID-19 affect mortality by other causes?
- How effective were restrictions at lowering excess deaths during the COVID-19 time period?

We will be training and testing various models on pre-COVID deaths. Once we find the model with the lowest error, we will re-train the model with the entire pre-COVID dataset. This model will forecast expected deaths assuming COVID did not occur. We can then compare actual deaths to expected deaths to understand excess mortality as a result of COVID-19, directly or indirectly, and the impact of COVID-19 on other causes of death.

There are three objectives in this project:

- Impact of COVID-19 on total mortality
- Impact of COVID-19 on non-COVID causes of death
- Evaluate efficacy of lockdowns

Loading Data

Load Libraries

Hide

```
# time series forecasting
library(forecast)

# neural Networks
library(nnfor)

# errors
library(Metrics)

# time series manipulation
library(xts)

#plotting tools with expanded functionality
library(ggplot2)
```

Data Source: Statistics Canada - Provisionaly Weekly Death Counts by Cause (https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1310081001&cubeTimeFrame.startDaily=2010-01-09&cubeTimeFrame.endDaily=2021-06-30&referencePeriods=20100109%2C20210630)

Load Data and Observe

Hide

```
DeathData = read.csv("data/ontario_covid_2010_2021.csv", fileEncoding = "UTF-8-BOM")
head(DeathData)
```

	Date <chr>	Total..all.causes.of.death <int>	Malignant.neoplasms <int>	Diseases.of.heart <int>
1	2010-01-09	1905	500	380
2	2010-01-16	1860	495	365
3	2010-01-23	1840	545	360
4	2010-01-30	1705	500	355
5	2010-02-06	1720	490	340
6	2010-02-13	1810	510	400

6 rows | 1-5 of 9 columns

This dataset displays weekly death counts in Ontario, organized by cause of death. The first column ‘Date’ is the first day of each week and ranges from 2010/01/09 to 2021/06/26 (YY/MM/DD), giving us 599 data points. The original dataset had values for all provinces, but had significant amounts of missing data for provinces other than Ontario. For this reason, we preprocessed the data to remove provinces other than Ontario. As we would expect from StatsCan data, there are no missing values, null values, or values that are extreme outliers. As such, no further data cleaning is required. The data range goes back to 2010. As we will show below, there was a large spike in deaths in 2018, likely due to a particularly bad North American flu season. (https://www.cdc.gov/flu/about/season/flu-season-2017-2018.htm). We selected data back to 2010 so that models will not overemphasize 2018, which had an unusually large number of deaths.

Hide

```
summary(DeathData)
```

```
     Date           Total..all.causes.of.death Malignant.neoplasms Diseases.of.heart Cer
ebrovascular.diseases
 Length:599         Min.   :1510               Min.   :425.0       Min.   :260.0     Mi
n.   : 60.00
 Class :character   1st Qu.:1745               1st Qu.:520.0       1st Qu.:340.0     1st
Qu.: 90.00
 Mode  :character   Median :1890               Median :545.0       Median :360.0     Med
ian : 95.00
                    Mean   :1913               Mean   :543.7       Mean   :366.2     Mea
n   : 96.94
                    3rd Qu.:2045               3rd Qu.:565.0       3rd Qu.:390.0     3rd
Qu.:105.00
                    Max.   :2645               Max.   :635.0       Max.   :510.0     Ma
x.   :135.00
 Chronic.lower.respiratory.diseases Accidents..unintentional.injuries.    COVID.19
Information.unavailable
 Min.   : 45.0                      Min.   : 55.0                      Min.   :  0.00
Min.   :  0.000
 1st Qu.: 65.0                      1st Qu.: 85.0                      1st Qu.:  0.00
1st Qu.:  0.000
 Median : 75.0                      Median : 95.0                      Median :  0.00
Median :  0.000
 Mean   : 78.4                      Mean   : 98.9                      Mean   : 15.98
Mean   :  6.269
 3rd Qu.: 85.0                      3rd Qu.:110.0                      3rd Qu.:  0.00
3rd Qu.:  0.000
 Max.   :140.0                      Max.   :180.0                      Max.   :490.00
Max.   :355.000
```

The above summary shows the descriptive statistics of the “Total..all.causes.of.death” and all other causes of death included in this dataset.

To create the time series, we will have to create year and week columns.

Data Clean Up: Add Year, Month and Day Columns

Hide

```
# number of weeks
numData = dim(DeathData)[1]

DateSplit = c(0,0,0)

for (i in 1:numData){
  DateSplit = unlist(strsplit(DeathData$Date[i],"-"))

  DeathData$Year[i] = as.numeric(DateSplit[1])
  DeathData$Month[i] = as.numeric(DateSplit[2])
  DeathData$Day[i] = as.numeric(DateSplit[3])
  DeathData$Total_named_causes_of_death[i] = sum(DeathData[i,3:8])
  DeathData$Other[i] = DeathData$Total..all.causes.of.death[i] - DeathData$Total_named_c
auses_of_death[i] # adds column which captures all unclassified deaths
}

DeathData$Year = sapply(DeathData$Year, as.numeric)
DeathData$Day = sapply(DeathData$Day, as.numeric)
startYear = DeathData$Year[1]

# display new columns
head(DeathData)
```

	Date <chr>	Total..all.causes.of.death <int>	Malignant.neoplasms <int>	Diseases.of.heart <int>
1	2010-01-09	1905	500	380
2	2010-01-16	1860	495	365
3	2010-01-23	1840	545	360
4	2010-01-30	1705	500	355
5	2010-02-06	1720	490	340
6	2010-02-13	1810	510	400

6 rows | 1-5 of 14 columns

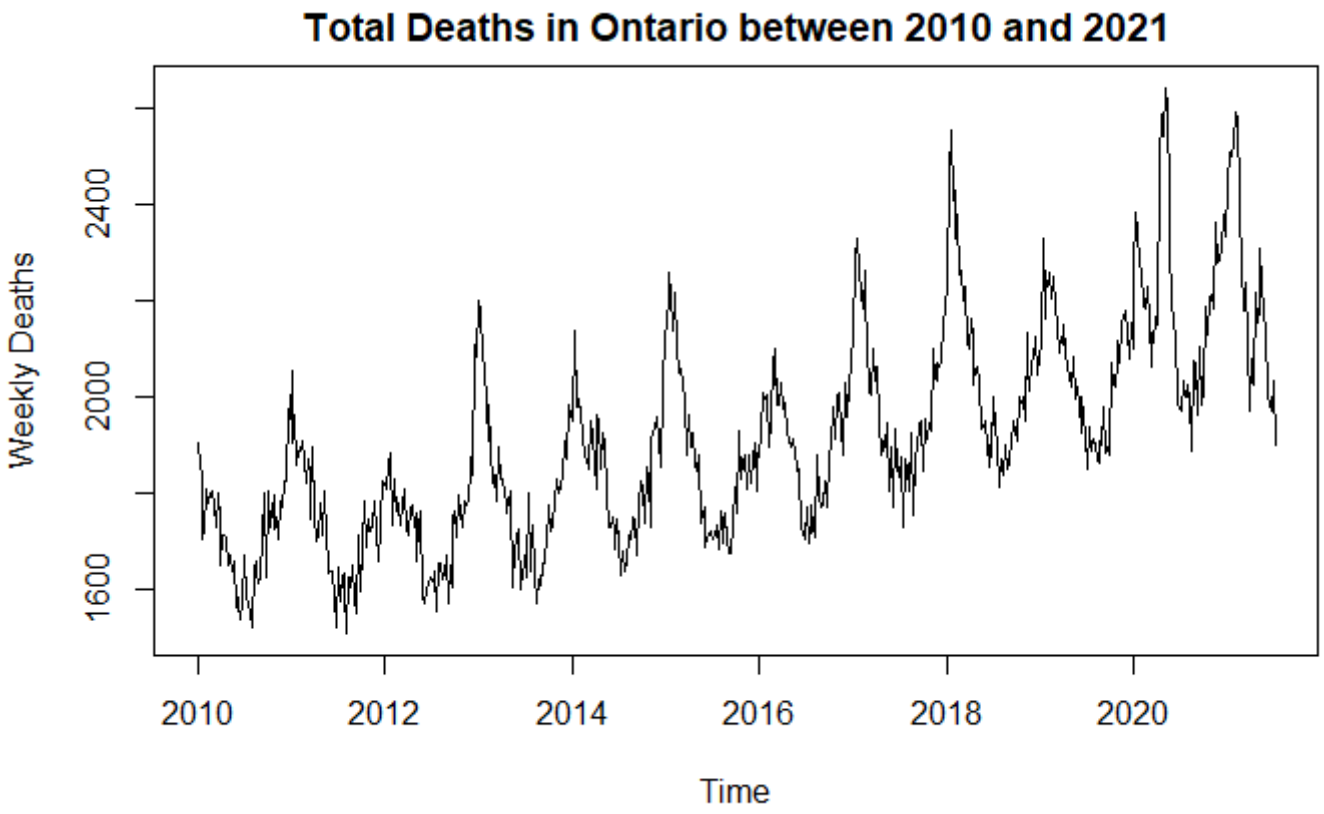
The dataset now has new columns for Year, Month and Day. Additionally, since the selected causes of death do not add up to the total deaths, a column titled “Other” was added, representing the difference between total deaths and deaths belonging to one of the causes included in the dataset.

Initial Observations

Create a Time Series of Total..all.causes.of.death

Hide

```
DeathData.ts <- ts(DeathData$Total..all.causes.of.death, start=c(startYear), frequency=52)
plot(DeathData.ts, main = "Total Deaths in Ontario between 2010 and 2021", ylab = "Weekly Deaths")
```



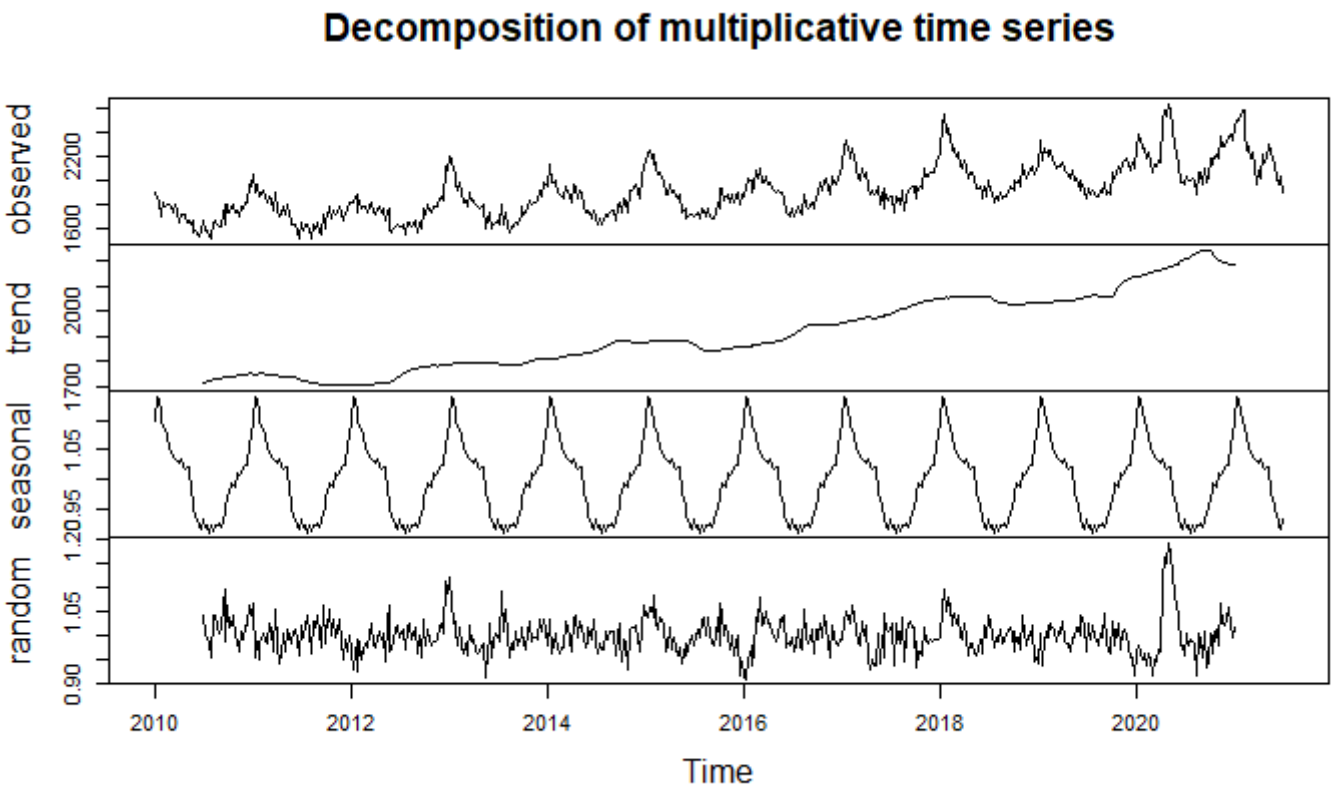
Observations:

- We can observe that the total deaths in Ontario displays seasonality and an increasing trend over time
- This points us to models such as Holts-Winter, neural networks, ARIMA and other potential non-course models such as TBATS because they account for trend and seasonality
- The relative seasonal amplitudes seem to increase, indicating that this is a multiplicative time series
- There is a spike in total deaths around January 2018, likely due to the unusually bad flu season
- The pattern changes from 2020 onwards. It is difficult to tell if it increases or decreases but there is a definite shift in the pattern due to COVID-19

View Decomposed Data

Hide

```
DeathData.ts.decomposed = decompose(DeathData.ts, type = "multiplicative")
plot(DeathData.ts.decomposed)
```



Observations:

- There is an increasing trend over time, which makes sense as the population increases over time and thus mortality should too
- There is seasonality evident in the data. This will be observed further in the non-COVID data to ensure the impact of the pandemic is removed
- The randomness is relatively constant around 1, therefore the multiplicative model is appropriate. There is a spike in randomness in 2020 which is the impact of COVID-19

Objective 1: What was the impact of COVID-19 on “Total..all.causes.of.death”:

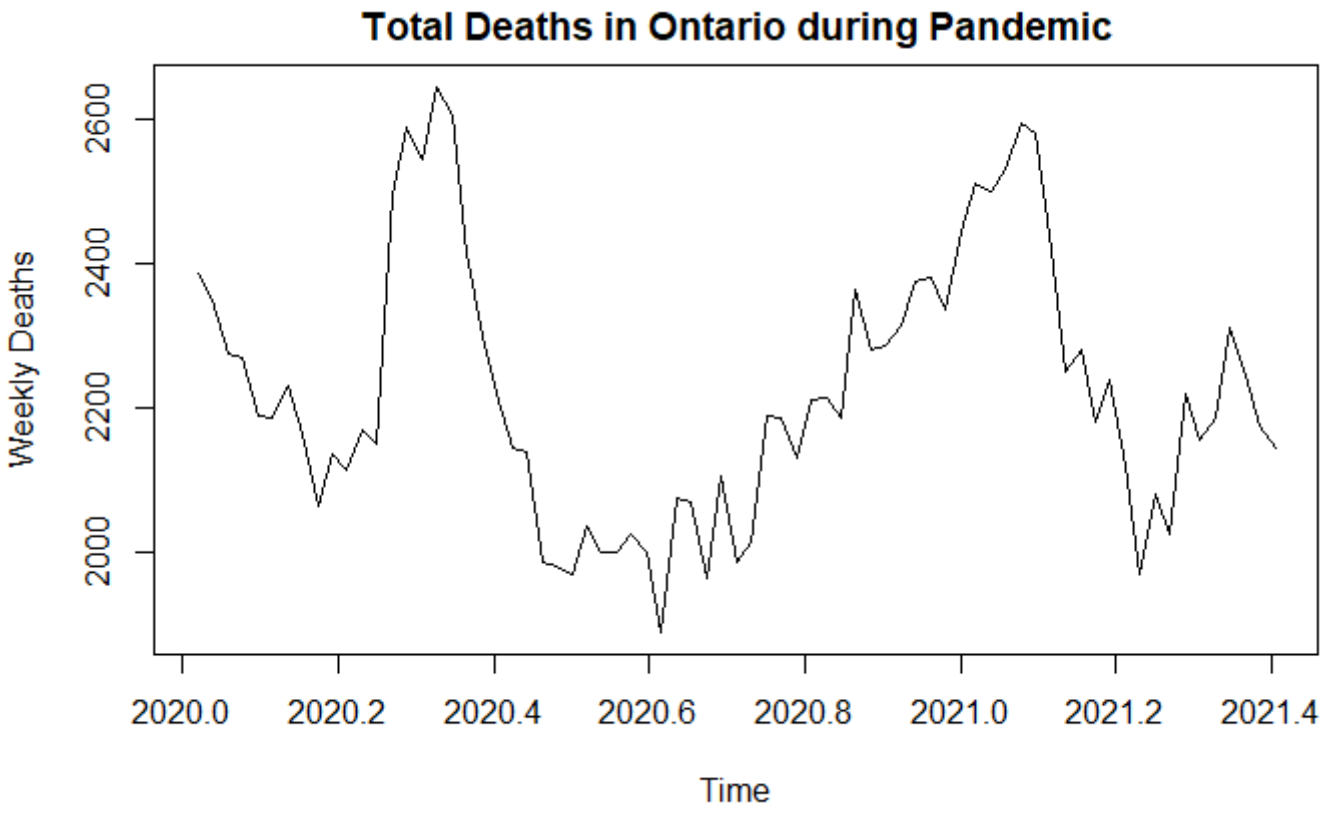
Process to answer Objective 1:

1. Split into pre-COVID and COVID data, and perform exploratory analysis
2. Test various models using pre-COVID data to find the lowest error
3. Rebuild best model and forecast what mortality should have been if COVID did not happen
4. Compare expected deaths without COVID to actual deaths during COVID to determine excess mortality

1) Subset Data into Pre-COVID and COVID data; exploratory analysis

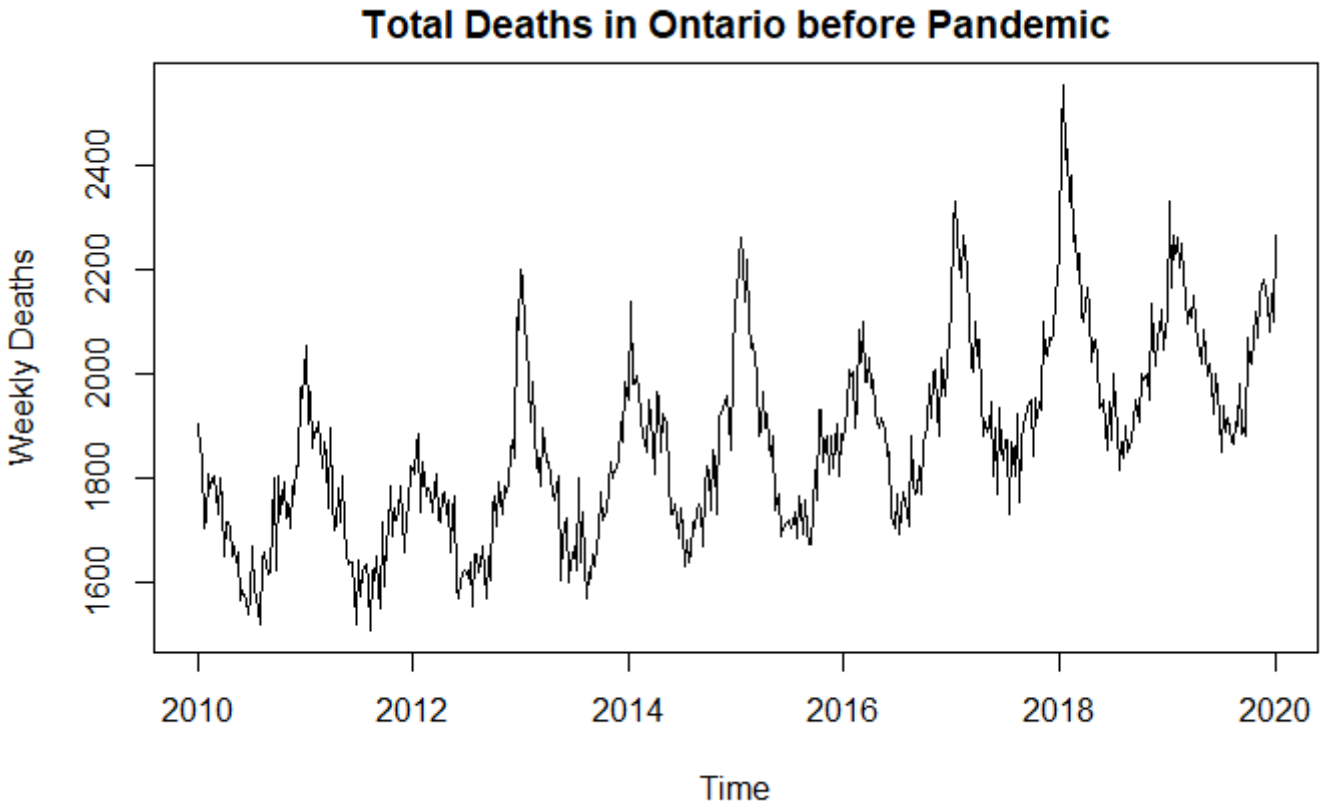
Hide

```
# define data points
numDataPoints = length(DeathData.ts)
start2020 = numDataPoints-(52+26)

# subset
DeathData.pre2020 = subset(DeathData.ts, start = 1, end = start2020)
DeathData.2020_2021 = subset(DeathData.ts, start = start2020+1 , end = numDataPoints-5)
#COVID set is from 2020 onwards
DeathData.predict.csv = subset(DeathData.ts, start = numDataPoints-5+1, end = numDataPoi
nts)
# write the last 2020-2021 subset to csv; this is the data that we will compare our pred
ictions to to determine excess deaths
write.csv(DeathData.predict.csv, 'data/predict.csv')

# plot subset data
plot(DeathData.2020_2021, main = "Total Deaths in Ontario during Pandemic", ylab = "Week
ly Deaths")
```



Hide

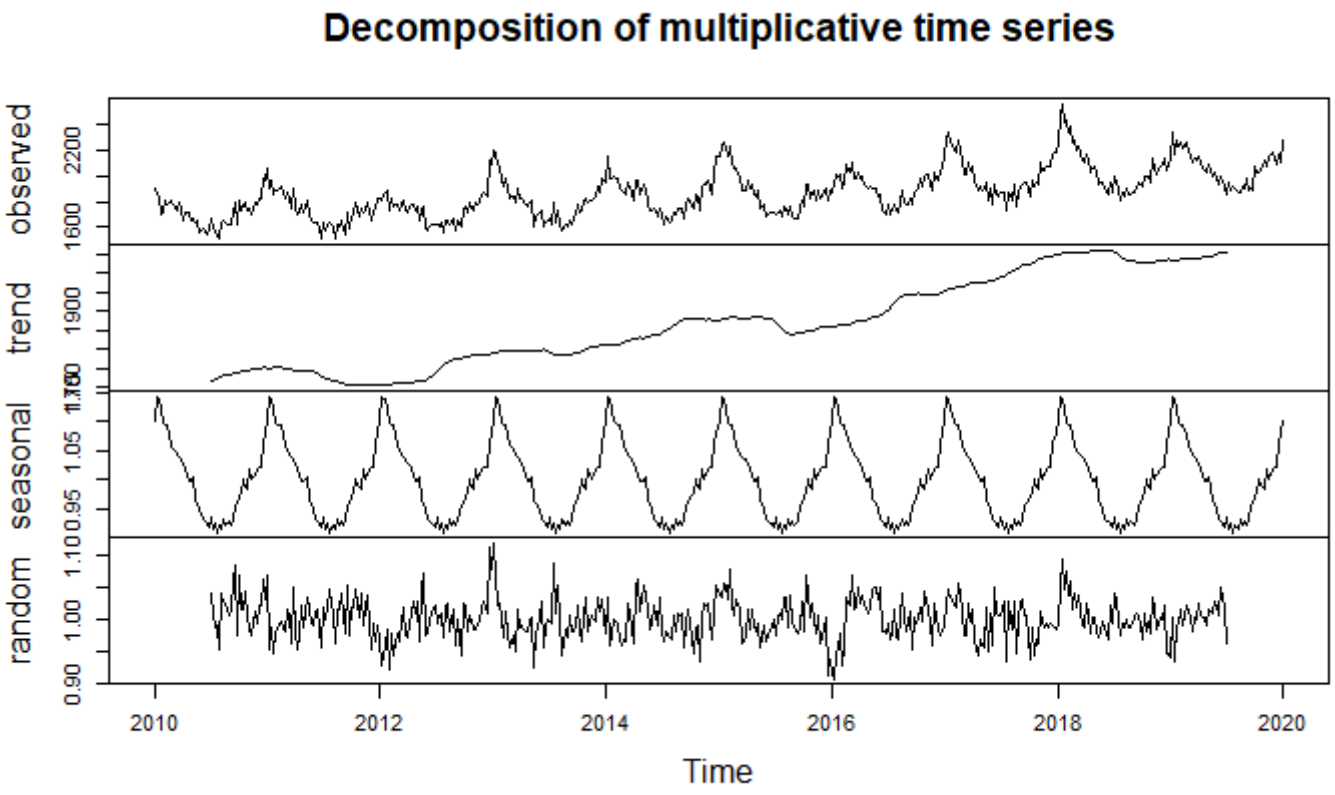
```
plot(DeathData.pre2020, main = "Total Deaths in Ontario before Pandemic", ylab = "Weekly
Deaths")
```



1.1.1) Decomposition of Pre-COVID

Hide

```
DeathData.pre2020.decomposed = decompose(DeathData.pre2020, type = "multiplicative")
plot(DeathData.pre2020.decomposed)
```



Hide

```
order(DeathData.pre2020.decomposed$figure)
```

```
 [1] 30 32 28 26 36 34 31 25 29 35 24 37 23 33 27 22 38 39 21 20 40 41 44 43 18 17 42 46
19 16 47 45 48 50 49 15 14 13 12
[40] 11 10  9 51  8 52  7  6  1  5  4  3  2
```

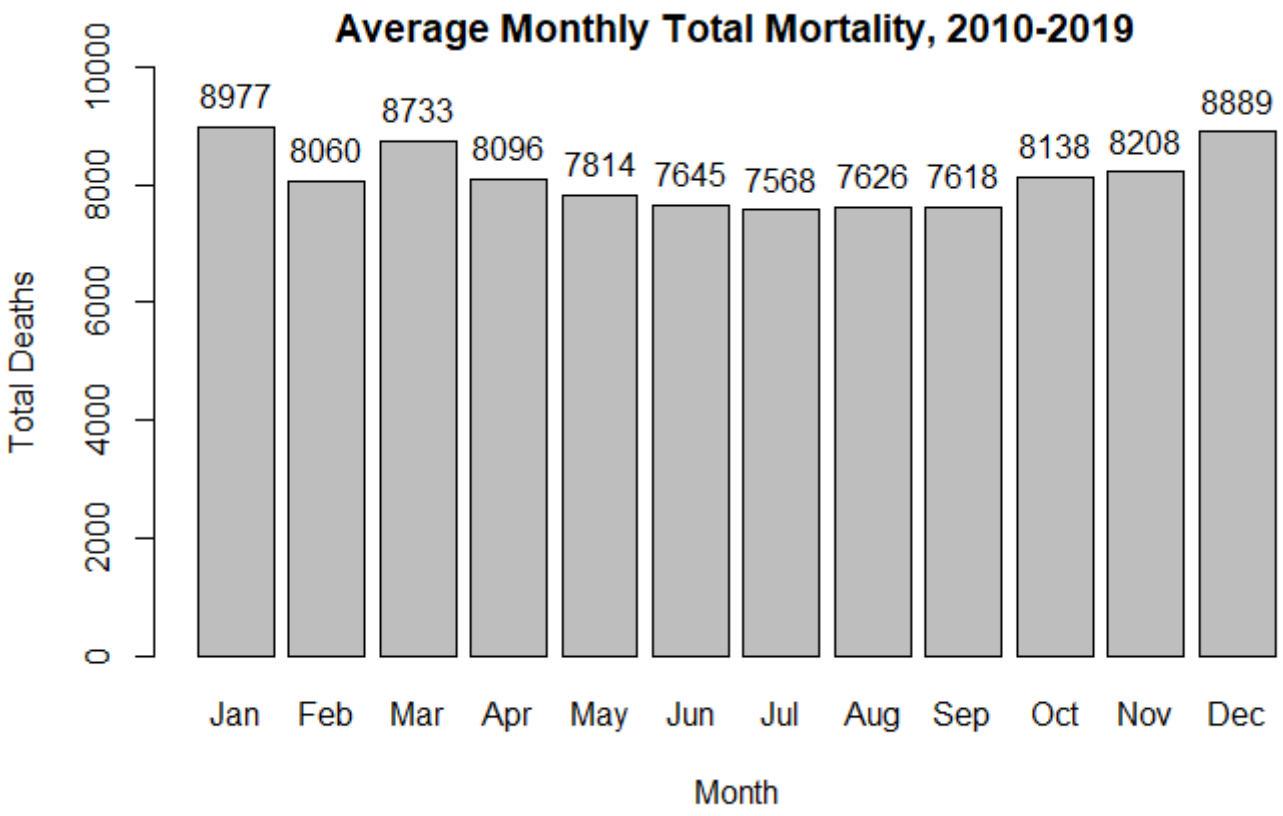
Observations of the non-COVID Total Deaths Data:

- The seasonality is ordered from lowest to highest, therefore the most deaths in Ontario occur from around weeks 1 - 7 (January - March) and least occur around weeks 25 - 35 (June - August)
- In the United States, death rates are generally 8-12% higher in winter months than in summer months due to various factors related to colder temperatures. (https://www.epa.gov/climate-indicators/climate-change-indicators-cold-related-deaths) It is unsurprising to see a similar pattern in Canada; we will explore below differences in mortality between seasons. We can also visually represent this in the section below by aggregating weekly data into months, and finding the average for that month over the years.

1.1.2) Monthly Total Deaths

Hide

```
# retrieve the dates that we need
dates = DeathData$Date[1:length(DeathData.pre2020)]

# convert the time series object to an xts for manipulation
DeathData.pre2020.xts = xts(x = DeathData.pre2020,
                            order.by = as.POSIXct(dates))

# aggregate months into a year/month matrix
monthMatrix = matrix(0, 10, 12)
for(i in 1:10){
  year = paste('20', as.character(i+9), sep = '')
  for(j in 1:12){
    month = as.character(j)
    index = paste(year, month, sep='-')
    monthMatrix[i,j] = sum(DeathData.pre2020.xts[index])
  }
}

# find the average of each month
monthlyAverage = matrix(0, 0, 12)
for(i in 1:12){
  # note: we divide by 10 because there are 10 years (2010 - 2019)
  # therefore there are 10 months worth of deaths
  # dividing by 10 gives average deaths in that month in this timeframe
  monthlyAverage[i] = sum(monthMatrix[,i]) / 10
}

# generate months for x-axis
months = seq(1, 12, 1)
months = month.abb[months]

# plot
plot =  barplot(monthlyAverage,
                main = "Average Monthly Total Mortality, 2010-2019",
                ylab = "Total Deaths",
                xlab = "Month",
                names.arg = months,
                ylim = c(0, 10000))

# data labels
text(plot, monthlyAverage, round(monthlyAverage, 0), cex = 1, pos = 3)
```



Validating what we observed from the decomposition, we can see that over the years, typically there are more deaths in the winter months and fewer in the summer months. To see how much higher deaths are in the winter than the summer:

Hide

```
DeathsOrdered = order(monthlyAverage)
print(DeathsOrdered)
```

```
 [1]  7  9  8  6  5  2  4 10 11  3 12  1
```

Hide

```
# this ranks deaths by month. Deaths are highest in January and lowest in July. By how m
uch is January higher than July?
HighestAverage = monthlyAverage[DeathsOrdered[12]]
LowestAverage = monthlyAverage[DeathsOrdered[1]]
SeasonalDiff = HighestAverage/LowestAverage-1
print(SeasonalDiff)
```

```
[1] 0.1861003
```

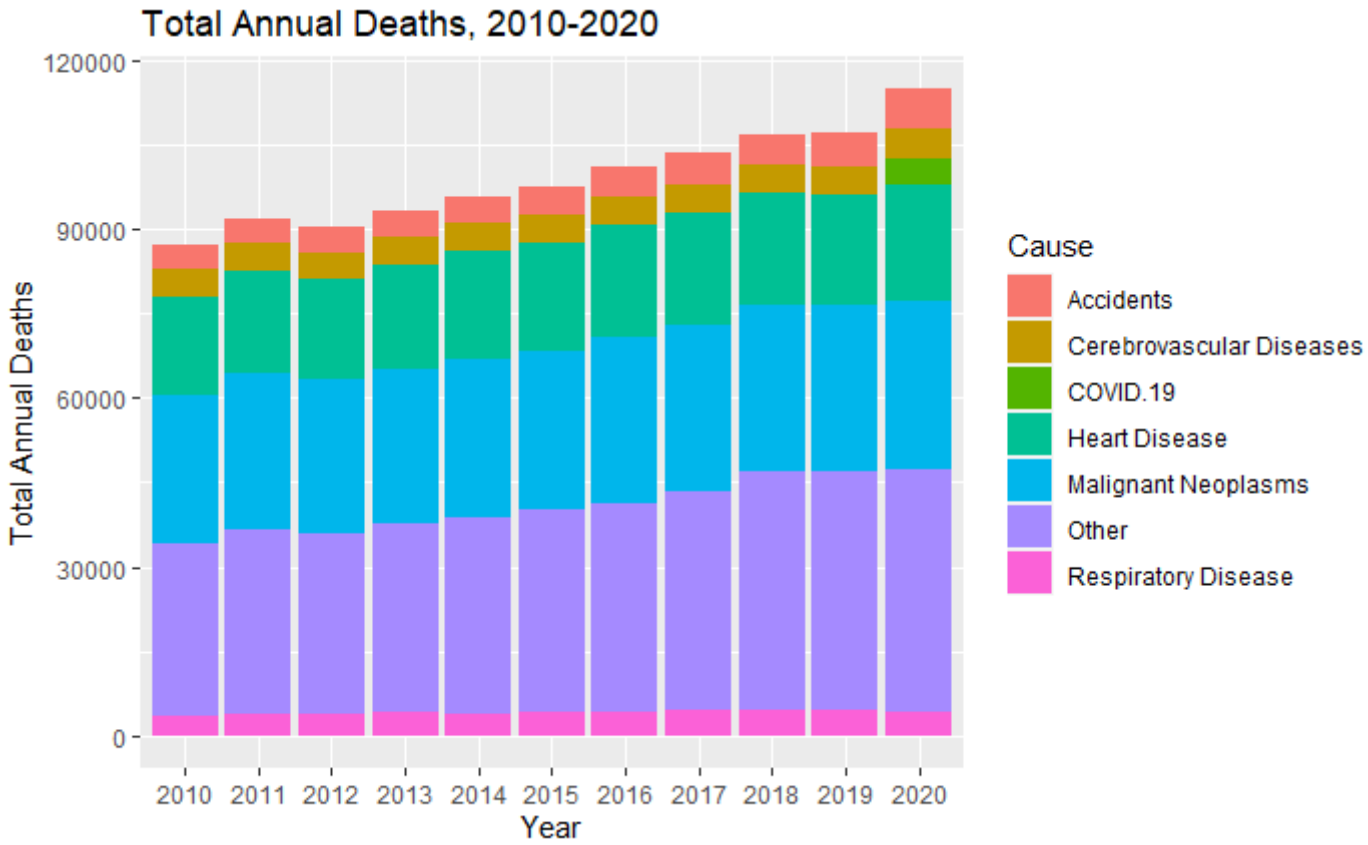
Deaths are higher in the winter by 18.6%, significantly higher than the 8-12% figure cited by the U.S. EPA report discussed above. This is likely due to Canada having an overall colder climate during the winter, with more parts of the country experiencing winter conditions. This increases the magnitude of the seasonality.

1.1.3) Total Annual Deaths by Cause

Hide

```
# create Empty Data Frame:
DeathsPerYear = data.frame(
  "Year" = c(2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020),
  "Malignant Neoplasms" = c(0),
  "Heart Disease" = c(0),
  "Cerebrovascular Diseases" = c(0),
  "Respiratory Disease" = c(0),
  "Accidents" = c(0),
  "COVID.19" = c(0),
  "Other" = c(0))


# shrink dataset from Deaths/Week into Deaths/Year:
index = 1
baseyear = 2010
neoplasmTotal <- heartTotal <- vascularTotal <- respiratoryTotal <- accidentsTotal <- co
vidTotal <- otherTotal <- 0

# repeat from 2010 to 2020
for (i in 1:11){
  # sum deaths in a given year
  while (DeathData$Year[index] == baseyear){
    neoplasmTotal = neoplasmTotal + DeathData$Malignant.neoplasms[index]
    heartTotal = heartTotal + DeathData$Diseases.of.heart[index]
    vascularTotal = vascularTotal + DeathData$Cerebrovascular.diseases[index]
    respiratoryTotal = respiratoryTotal + DeathData$Chronic.lower.respiratory.diseases[i
ndex]
    accidentsTotal = accidentsTotal + DeathData$Accidents..unintentional.injuries.[inde
x]
    covidTotal = covidTotal + DeathData$COVID.19[index]
    otherTotal = otherTotal + DeathData$Other[index]

    index = index + 1
  }

  # fill in the empty data frame
  DeathsPerYear[i, 2] = neoplasmTotal
  DeathsPerYear[i, 3] = heartTotal
  DeathsPerYear[i, 4] = vascularTotal
  DeathsPerYear[i, 5] = respiratoryTotal
  DeathsPerYear[i, 6] = accidentsTotal
  DeathsPerYear[i, 7] = covidTotal
  DeathsPerYear[i, 8] = otherTotal
  neoplasmTotal <- heartTotal <- vascularTotal <- respiratoryTotal <- accidentsTotal <-
 covidTotal <- otherTotal <- 0
  baseyear = baseyear + 1
}

# re-arrange data for plotting
causesNum = 7
year = c(
  rep("2010", causesNum),
```

```
  rep("2011", causesNum),
  rep("2012", causesNum),
  rep("2013", causesNum),
  rep("2014", causesNum),
  rep("2015", causesNum),
  rep("2016", causesNum),
  rep("2017", causesNum),
  rep("2018", causesNum),
  rep("2019", causesNum),
  rep("2020" , causesNum))

condition = rep(c(
    "Malignant Neoplasms",
    "Heart Disease",
    "Cerebrovascular Diseases",
    "Respiratory Disease",
    "Accidents",
    "COVID.19",
    "Other"
  ), 11)

value = data.frame(77)
index = 1
for (i in 1:11) {
  for (j in 1:causesNum) {
    value[index] = DeathsPerYear[i, j + 1]
    index = index + 1
  }
}

value <- as.data.frame(t(value))

# create dataframe
dea = data.frame("Year" = year, "Cause" = condition, "Deaths" = value)

# plot
ggplot(data = dea, aes(x = Year, y = V1, fill = Cause)) +
       geom_col() +
       ylab("Total Annual Deaths") +
       ggtitle("Total Annual Deaths, 2010-2020")
```



Although there was a clear increase in deaths in 2020 due to COVID-19, COVID-19 is actually a relatively small contributor to the total annual deaths, with Accidents, Malignant Neoplasms, Heart Disease, and “other” causes of death being clearly more. This shows that we must investigate the impact of COVID-19 on all causes of death, as the second-order impacts of COVID and policy responses to the pandemic may have influenced other causes of death.

2) Model Selection

2.1) Split non-COVID data into Training and Testing time series

Hide

```
training.percent = .75

nTrain = round(length(DeathData.pre2020)*training.percent)
nTest = round(length(DeathData.pre2020)-nTrain)

train.ts = subset(DeathData.pre2020, start = 1, end = nTrain)
testing.ts = subset(DeathData.pre2020, start = nTrain + 1, end = nTrain + nTest)

# check
originalLength = length(DeathData.pre2020)
originalLength == (nTrain + nTest)
```

```
[1] TRUE
```

Pass: originalLength == nTrain + nTest

2.2) Try different models

We will test the following three models for the lowest error, then re-train the best model with the entire dataset.

Models:

- 2.2.1 - Feed Forward Neural Network
- 2.2.2 - ARIMA
- 2.2.3 - TBATS

We chose not to do Drift, Mean, or (S)Naive, and other models which do not capture trend/seasonality, as they will not appropriately handle our data, which has trend and seasonality. Holt-Winters doesn't suffice as the frequency of our data is greater than 48 seasonal patterns. We rejected linear regression as it does not handle multiplicative seasonality.

2.2.1 Feed Forward Neural Network

Model Description: A neural network is a collection of connected nodes in which the connections between nodes do not form a cycle. Neural networks are trained by modifying the weights of the connections, resulting in a model that can approximate complicated functions. Neural networks are effective because unlike other models they make no assumptions about the trend. The following implementation uses the elm() function in the 'nnfor' package, which fits Extreme Learning Machine (ELM) neural networks for time series forecasting.

Train the Neural Network and Forecast

Hide

```
defaultW <- getOption("warn")
options(warn = -1)

DeathData.ts.pre2020.training.elm = elm(train.ts)
summary(DeathData.ts.pre2020.training.elm)
```

```
            Length Class  Mode
net          14    nn     list
hd            1    -none- numeric
W.in          0    -none- NULL
W            20    -none- list
b            20    -none- numeric
W.dct        20    -none- list
lags         25    -none- numeric
xreg.lags     0    -none- NULL
difforder     1    -none- numeric
sdummy        1    -none- logical
ff.det        1    -none- numeric
det.type      1    -none- character
y           391    ts     numeric
minmax        4    -none- list
xreg.minmax   0    -none- NULL
comb          1    -none- character
type          1    -none- character
direct        1    -none- logical
fitted      341    ts     numeric
MSE           1    -none- numeric
```

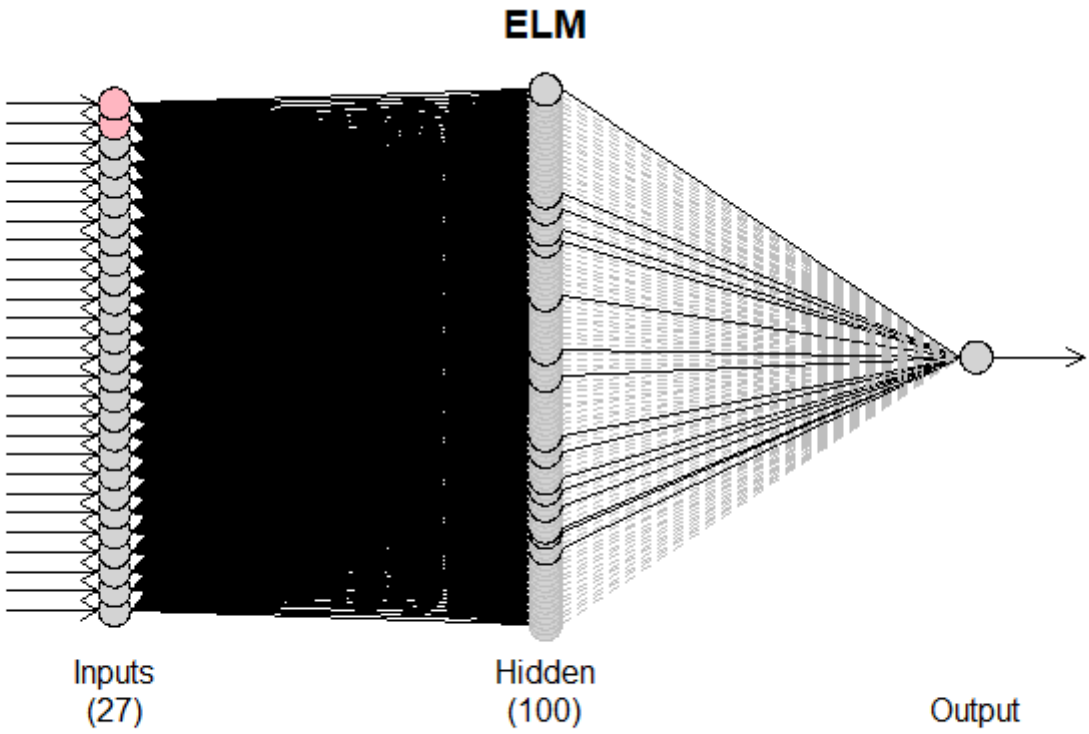
Hide

```
DeathData.ts.pre2020.training.elm.fcast = forecast(DeathData.ts.pre2020.training.elm, h
= nTest)
options(warn = defaultW)
```

Plot the NN and Display Statistics

Hide

```
plot(DeathData.ts.pre2020.training.elm)
```



Hide

```
print(DeathData.ts.pre2020.training.elm)
```

```
ELM fit with 100 hidden nodes and 20 repetitions.
Series modelled in differences: D1.
Univariate lags: (1,2,3,4,5,6,7,8,9,10,11,12,14,15,16,17,18,19,21,24,26,39,41,48,49)
Deterministic seasonal dummies included.
Forecast combined using the median operator.
Output weight estimation using: lasso.
MSE: 3963.7809.
```

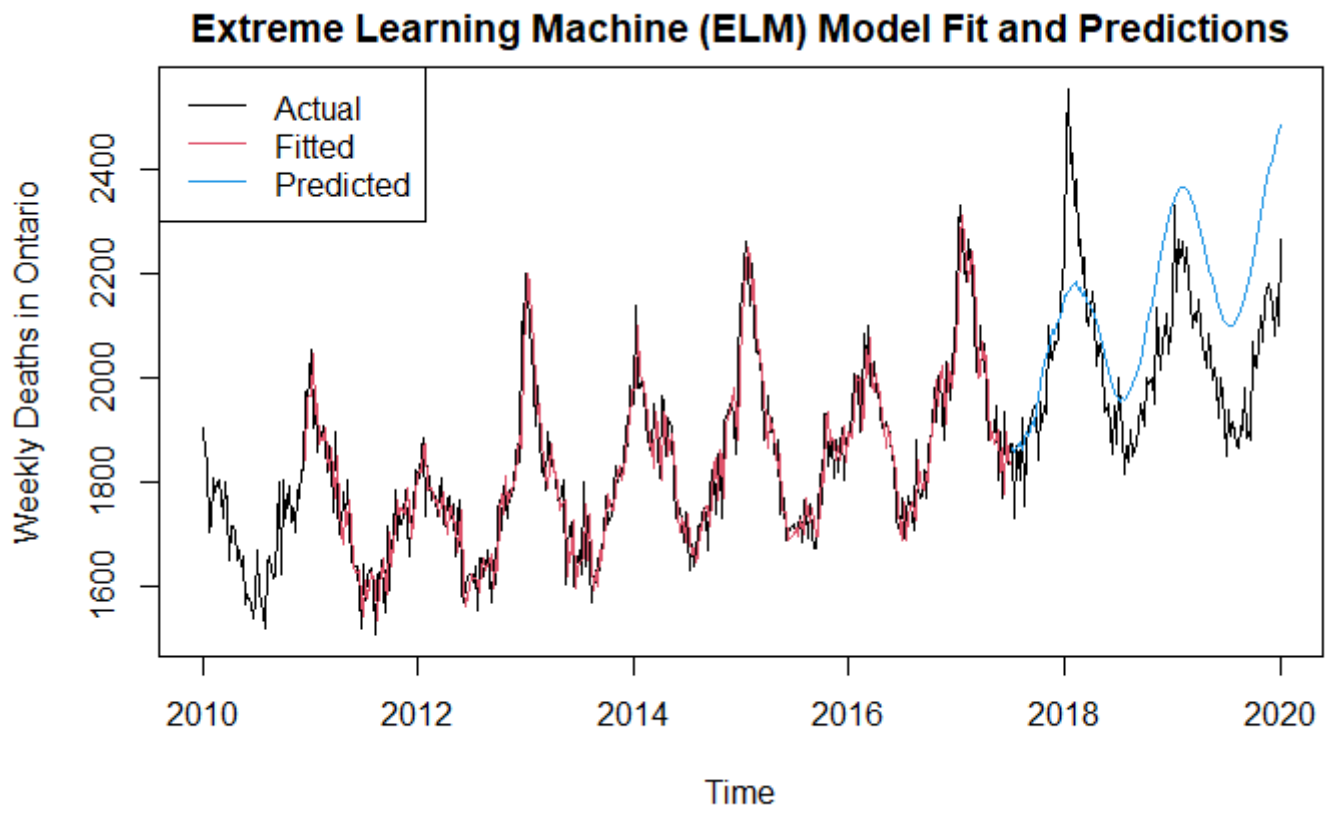
Display Model Fit and Predictions

Hide

```
plot(DeathData.pre2020, main = "Extreme Learning Machine (ELM) Model Fit and Prediction
s", ylab = "Weekly Deaths in Ontario")
lines(DeathData.ts.pre2020.training.elm.fcast$fitted, col = 2)
```

Hide

```
lines(DeathData.ts.pre2020.training.elm.fcast$mean, col = 4)
legend("topleft", lty = 1, col = c(1,2,4), legend = c("Actual", "Fitted", "Predicted"))
```



NN RMSE

Hide

```
elm.rmse = rmse(testing.ts[1:length(testing.ts)], DeathData.ts.pre2020.training.elm.fcast$mean)
elm.rmse
```

```
[1] 166.8074
```

NN MAPE

Hide

```
elm.mape = mape(testing.ts[1:length(testing.ts)], DeathData.ts.pre2020.training.elm.fcast$mean)
elm.mape
```

```
[1] 0.06928313
```

2.2.2 ARIMA

Model Description: Autoregressive integrated moving average (ARIMA) is the most used approach for time series data and is composed of three parts: 1. auto regressive part which is a regression model that includes lag (past observations). 2. a differencing part that makes the data stationary by using a constant mean and variance over time, and 3. a moving average part that takes the average of the last N observations. Seasonality can be

incorporated into ARIMA by using SARIMA, which adds additional parameters for seasonality. The following implementation uses auto.arima() which automates the process of finding the parameters that generate the least error.

Train the ARIMA model

Hide

```
# create the model and view summary
arima.model = auto.arima(train.ts)
```

Test the Model

Hide

```
# print the summary
summary(arima.model)
```

```
Series: train.ts
ARIMA(1,0,2)(2,1,0)[52] with drift

Coefficients:
         ar1      ma1     ma2     sar1     sar2   drift
      0.8668  -0.5398  0.0294  -0.6803  -0.1848  0.6953
s.e.  0.0435   0.0695  0.0645   0.0571   0.0651  0.1473

sigma^2 estimated as 4458:  log likelihood=-1914.68
AIC=3843.36   AICc=3843.69   BIC=3870.14

Training set error measures:
                    ME     RMSE      MAE         MPE    MAPE      MASE         ACF1
Training set -0.456487 61.61973 44.87435 -0.1260478 2.45313 0.5028894 0.0004267557
```

Hide

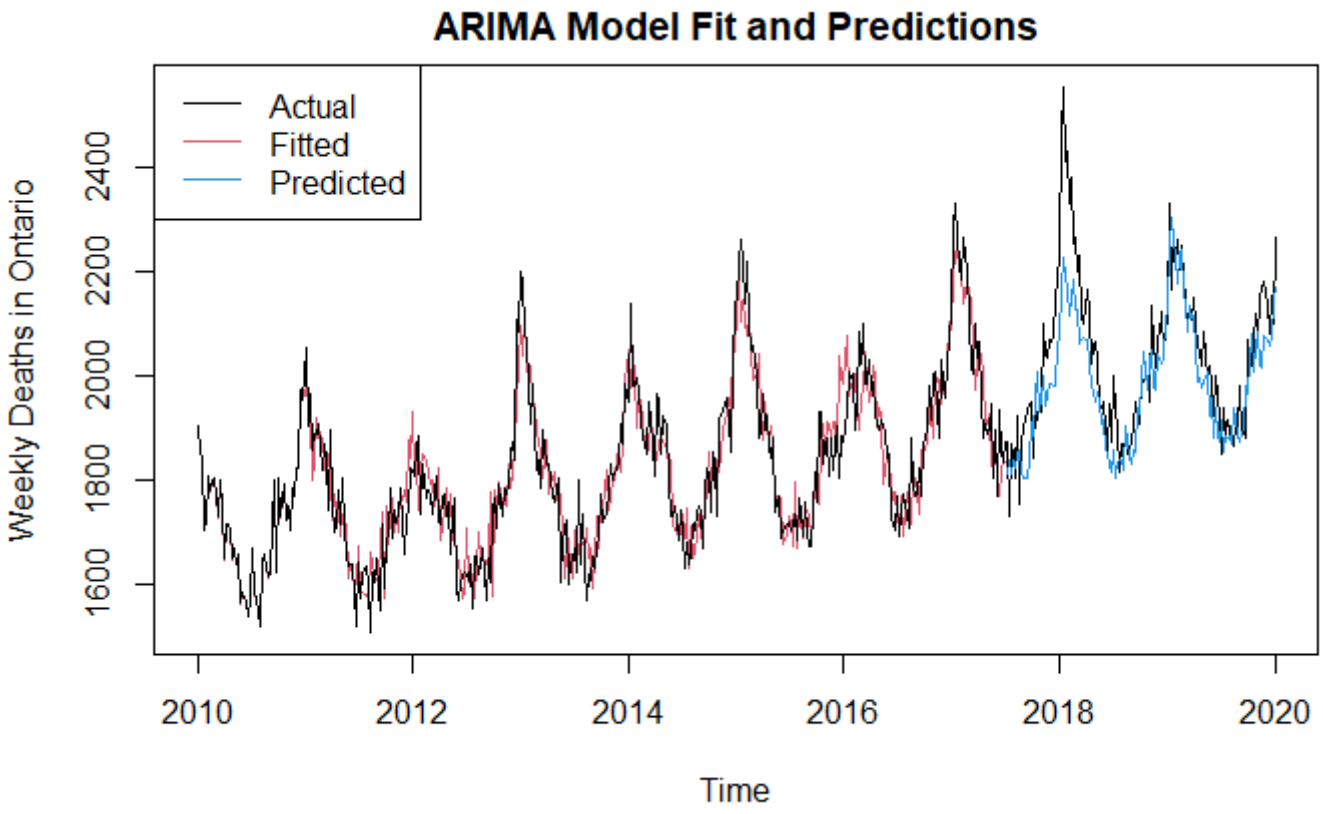
```
# create the forecast
arima.fcast = forecast(arima.model, h = nTest)

# plot
plot(DeathData.pre2020, col = 1, main = "ARIMA Model Fit and Predictions", ylab = "Weekl
y Deaths in Ontario")
lines(arima.fcast$fitted, col = 2)
```

Hide

```
lines(DeathData.pre2020, col = 1)
lines(arima.fcast$mean, col = 4)
```

Hide

```
legend("topleft", lty = 1, col = c(1,2,4), legend = c("Actual", "Fitted", "Predicted"))
```



Hide

```
# calculate errors
arima.rmse = rmse(testing.ts[1:length(testing.ts)], arima.fcast$mean)
arima.mape = mape(testing.ts[1:length(testing.ts)], arima.fcast$mean)

# display errors
arima.rmse
```

```
[1] 90.50631
```

Hide

```
arima.mape
```

```
[1] 0.0323805
```

Observations:

- The constants from the auto.arima are as follows: ARIMA(1,0,2)(2,1,0)[52] with drift
- The display of ARIMA constants for both trend and seasonality confirms our above assumption that there is clear trend and seasonality in this data

2.2.3 TBATS

Model Description: TBATS is an advanced time series forecasting method that consists of the following features: trigonometric seasonality, box-cox transformation, ARMA errors, trend and seasonal components. The following implementation uses the tbats() function from the 'forecast' library, as described by De Llevra, Hyndman & Snyder in 2011 (https://www.rdocumentation.org/packages/forecast/versions/8.15/topics/tbats). Since we were unable to implement Holts-winter, TBATS is an effective substitute as it incorporates exponential smoothing.

Hide

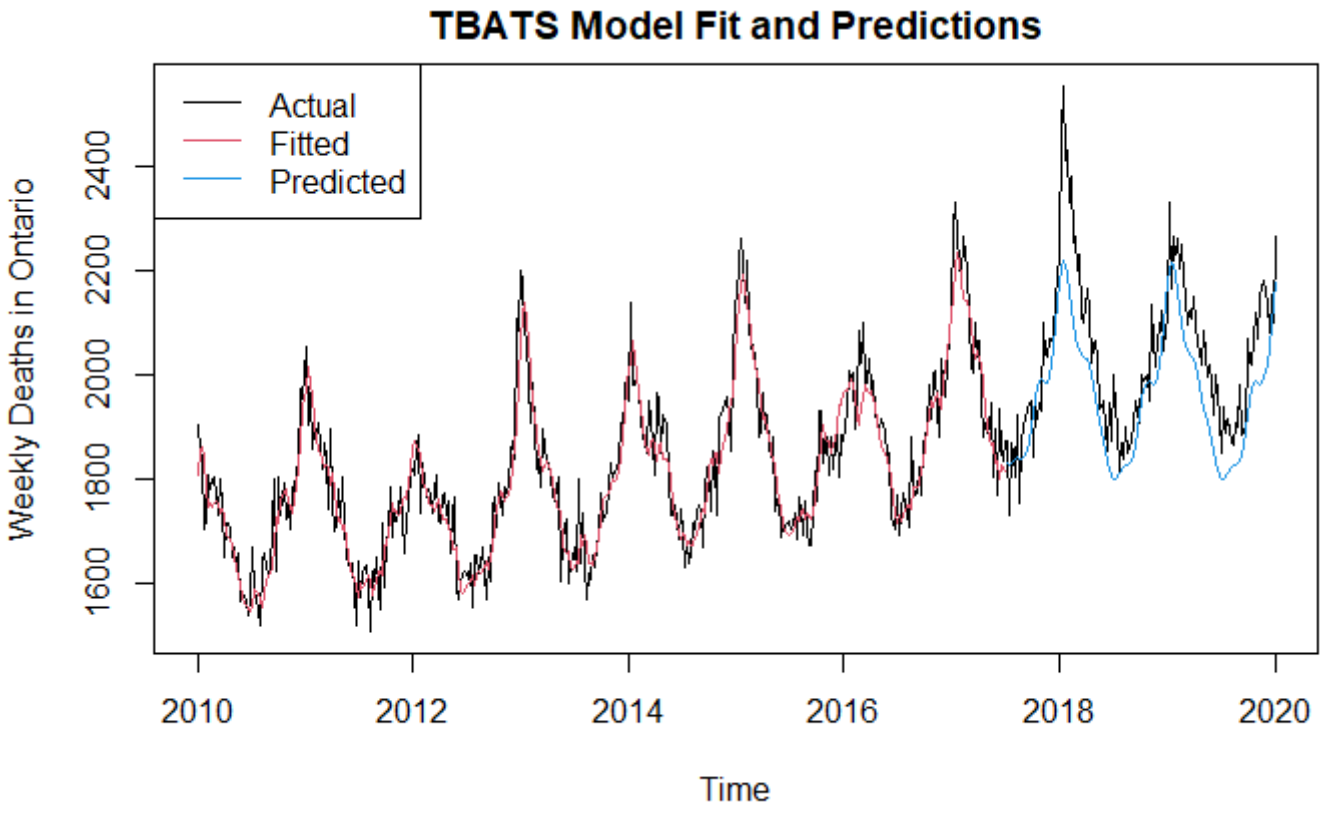
```
# train and forecast TBATS
trainTBATS <- tbats(train.ts)
trainTBATS.fcast <- forecast(trainTBATS, h = nTest)
```

Hide

```
# display TBATS Model Fit and Predictions
plot(DeathData.pre2020, main = "TBATS Model Fit and Predictions", ylab = "Weekly Deaths
 in Ontario")
lines(trainTBATS.fcast$fitted, col = 2)
```

Hide

```
lines(trainTBATS.fcast$mean, col = 4)
legend("topleft", lty = 1, col = c(1,2,4), legend = c("Actual", "Fitted", "Predicted"))
```



Hide

```
# calculate Errors
tbats.rmse = rmse(testing.ts[1:length(testing.ts)], trainTBATS.fcast$mean)
tbats.mape = mape(testing.ts[1:length(testing.ts)], trainTBATS.fcast$mean)

# display Errors
tbats.rmse
```

```
[1] 101.9873
```

Hide

```
tbats.mape
```

```
[1] 0.04059625
```

2.3 Select the best model

We create a table that displays the MAPE and RMSE of each model that we have tested.

Hide

```
errors = matrix(c(elm.mape,elm.rmse,arima.mape,arima.rmse,tbats.mape,tbats.rmse),ncol =
2,byrow = TRUE)
colnames(errors) = c("MAPE","RMSE")
rownames(errors) = c("ELM","ARIMA","TBATS")
print(errors)
```

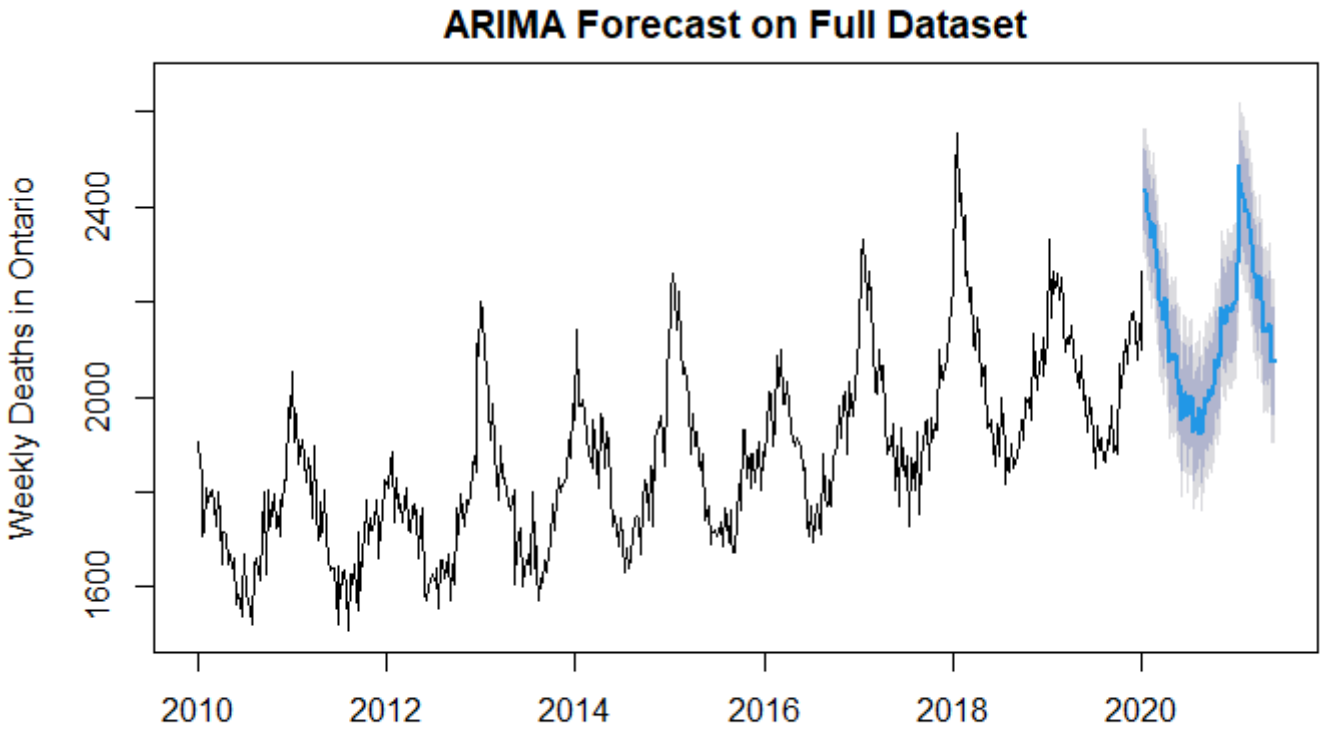
```
            MAPE      RMSE
ELM   0.06928313 166.80741
ARIMA 0.03238050  90.50631
TBATS 0.04059625 101.98727
```

All models have MAPE below ~6% indicating that the models selected for testing are appropriate for the data. ARIMA has lowest MAPE and lowest RMSE and is selected as our final model.

3) Rebuild model with full dataset, and use it to predict excess deaths.

Hide

```
# re-create the ARIMA model with the entirety of the 2020 data
actualArima <- auto.arima(DeathData.pre2020)
actualArima.fcast <- forecast(actualArima, h = length(DeathData.2020_2021))
plot(actualArima.fcast, main = "ARIMA Forecast on Full Dataset", ylab = "Weekly Deaths i
n Ontario")
```



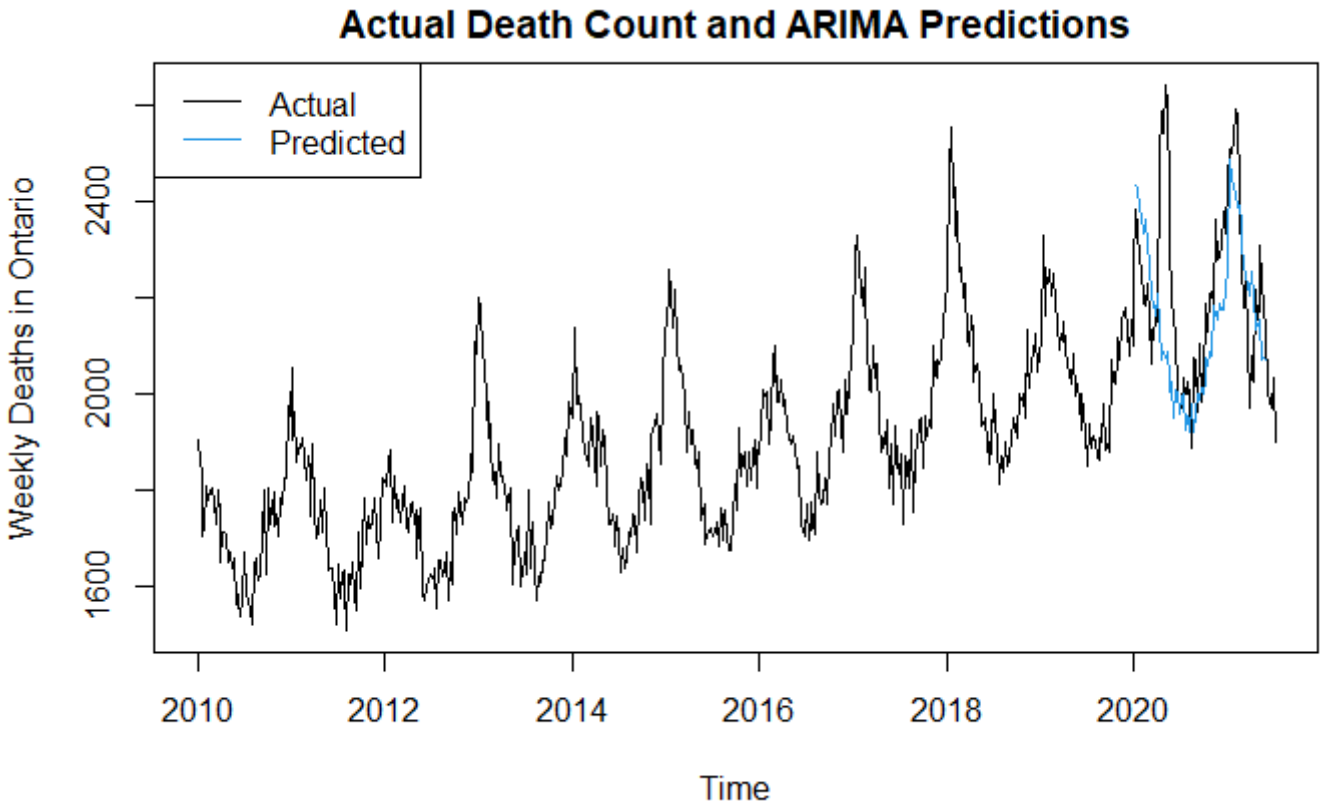
4) Compare predicted deaths to actual deaths during COVID

Hide

```
plot(DeathData.ts, main = "Actual Death Count and ARIMA Predictions", ylab = "Weekly Dea
ths in Ontario")
lines(actualArima.fcast$mean, col=4)
```

Hide

```
legend("topleft", lty=1, col=c(1,4), legend = c("Actual", "Predicted"))
```



The ARIMA prediction does not align with actual deaths during the COVID time period. This does not indicate that our prediction is poor, rather, it demonstrates the impact of COVID-19 on mortality.

Hide

```
PredictedDeaths = sum(actualArima.fcast$mean)
paste("Predicted Deaths:", round(PredictedDeaths, digits=0))
```

```
[1] "Predicted Deaths: 157496"
```

Hide

```
ActualDeaths = sum(DeathData.2020_2021)
paste("Actual Deaths:", ActualDeaths)
```

```
[1] "Actual Deaths: 162495"
```

Hide

```
ExcessDeaths = ActualDeaths - PredictedDeaths
paste("Excess Deaths:", round(ExcessDeaths, digits=0))
```

```
[1] "Excess Deaths: 4999"
```

Hide

```
COVIDDeaths = sum(DeathData$COVID.19)
paste("COVID Deaths:", round(COVIDDeaths, digits=0))
```

```
[1] "COVID Deaths: 9575"
```

Our model forecasts that approximately 167,629 deaths were expected, while 172,390 actually occurred. The COVID-19 pandemic resulted in approximately 4,761 more deaths than expected. There have been 9,575 COVID-19 deaths, which indicates other causes of death decreased.

We now know that COVID-19 decreased the amount of other death causes in Ontario over 2020 and 2021. This could be for a number of reasons, for example, less people are going outside, more people wearing masks, misdiagnosis of other causes, etc. For further analysis, we need to observe the impact of COVID-19 on each respective cause of death.

Objective 2: What was the impact of COVID-19 on other causes of death

Review summary of causes of death

Hide

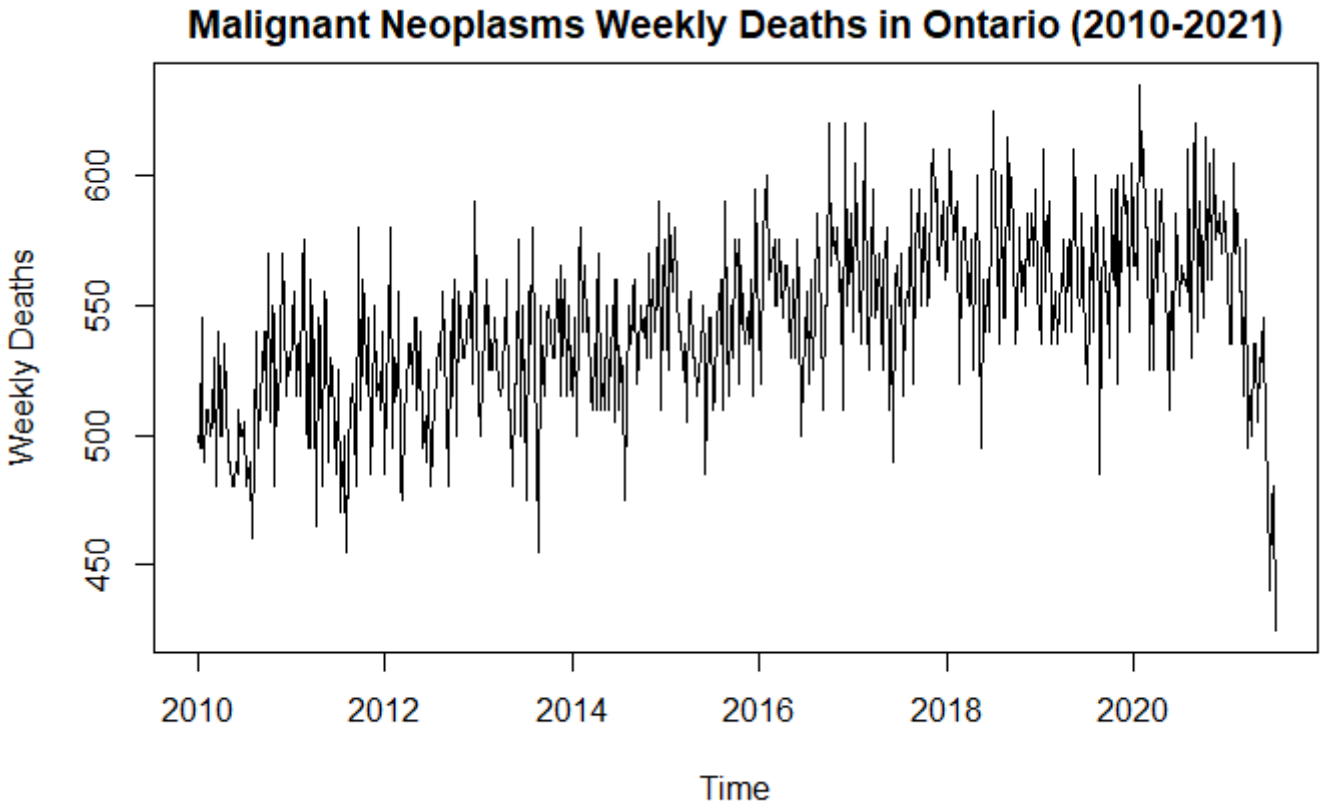
```
summary(DeathData)
```

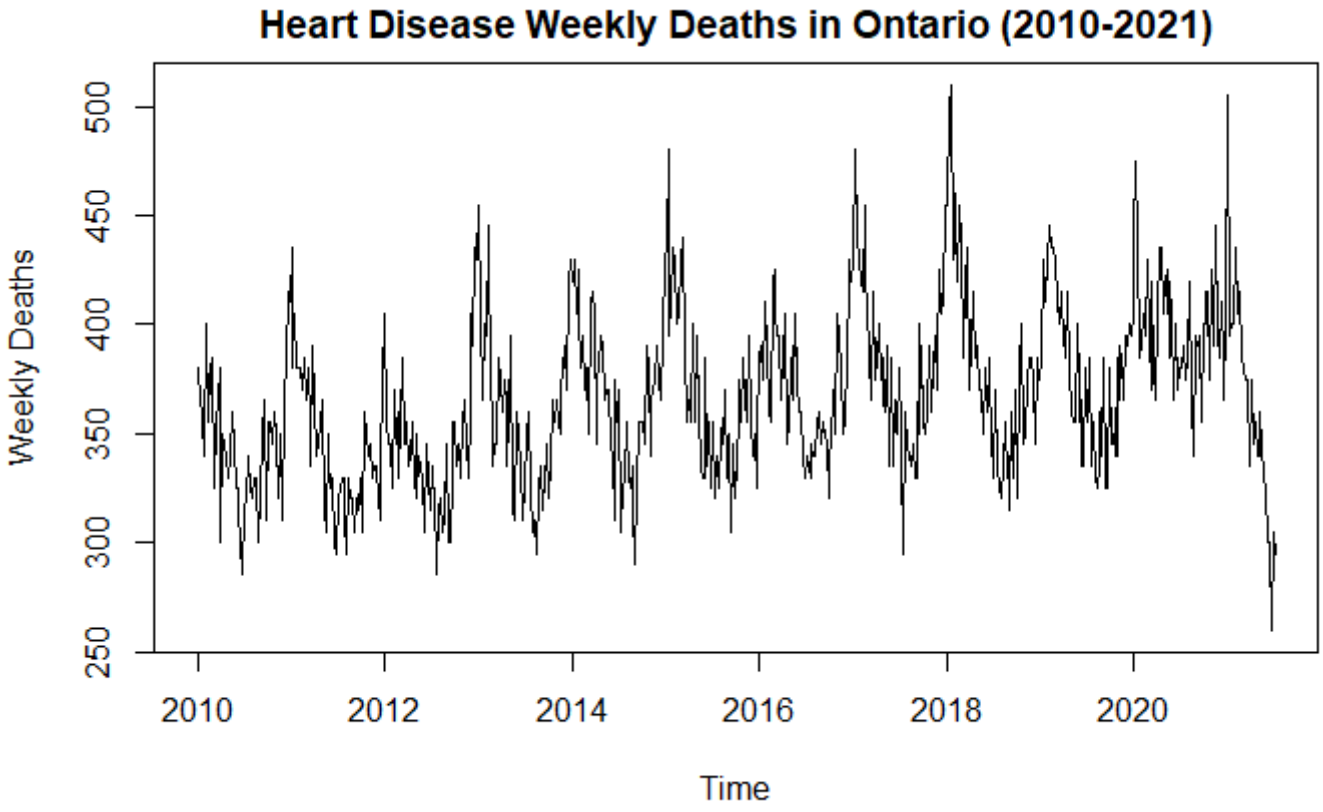
```
     Date           Total..all.causes.of.death Malignant.neoplasms Diseases.of.heart Cer
ebrovascular.diseases
 Length:599         Min.   :1510               Min.   :425.0       Min.   :260.0     Mi
n.   : 60.00
 Class :character   1st Qu.:1745               1st Qu.:520.0       1st Qu.:340.0     1st
Qu.: 90.00
 Mode  :character   Median :1890               Median :545.0       Median :360.0     Med
ian : 95.00
                    Mean   :1913               Mean   :543.7       Mean   :366.2     Mea
n   : 96.94
                    3rd Qu.:2045               3rd Qu.:565.0       3rd Qu.:390.0     3rd
Qu.:105.00
                    Max.   :2645               Max.   :635.0       Max.   :510.0     Ma
x.   :135.00
 Chronic.lower.respiratory.diseases Accidents..unintentional.injuries.    COVID.19
Information.unavailable
 Min.   : 45.0                      Min.   : 55.0                       Min.   :  0.00
Min.   :  0.000
 1st Qu.: 65.0                      1st Qu.: 85.0                       1st Qu.:  0.00
1st Qu.:  0.000
 Median : 75.0                      Median : 95.0                       Median :  0.00
Median :  0.000
 Mean   : 78.4                      Mean   : 98.9                       Mean   : 15.98
Mean   :  6.269
 3rd Qu.: 85.0                      3rd Qu.:110.0                       3rd Qu.:  0.00
3rd Qu.:  0.000
 Max.   :140.0                      Max.   :180.0                       Max.   :490.00
Max.   :355.000
      Year          Month             Day        Total_named_causes_of_death     Other
 Min.   :2010   Min.   : 1.000   Min.   : 1.00   Min.   : 935                Min.   : 50
0.0
 1st Qu.:2012   1st Qu.: 3.000   1st Qu.: 8.00   1st Qu.:1130                1st Qu.: 61
5.0
 Median :2015   Median : 6.000   Median :16.00   Median :1185                Median : 70
0.0
 Mean   :2015   Mean   : 6.404   Mean   :15.76   Mean   :1200                Mean   : 71
3.3
 3rd Qu.:2018   3rd Qu.: 9.000   3rd Qu.:23.00   3rd Qu.:1255                3rd Qu.: 79
5.0
 Max.   :2021   Max.   :12.000   Max.   :31.00   Max.   :1800                Max.   :109
0.0
```

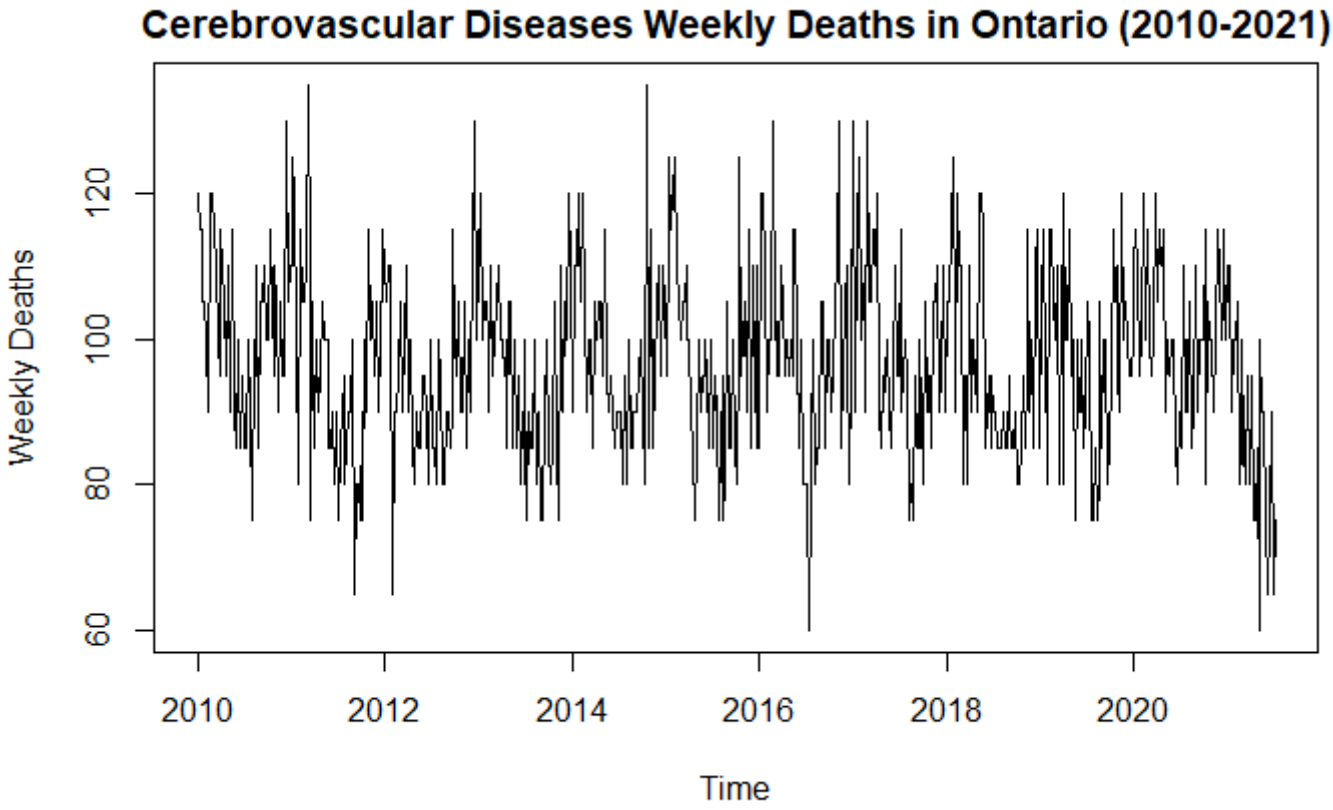
Create time series for all death causes:

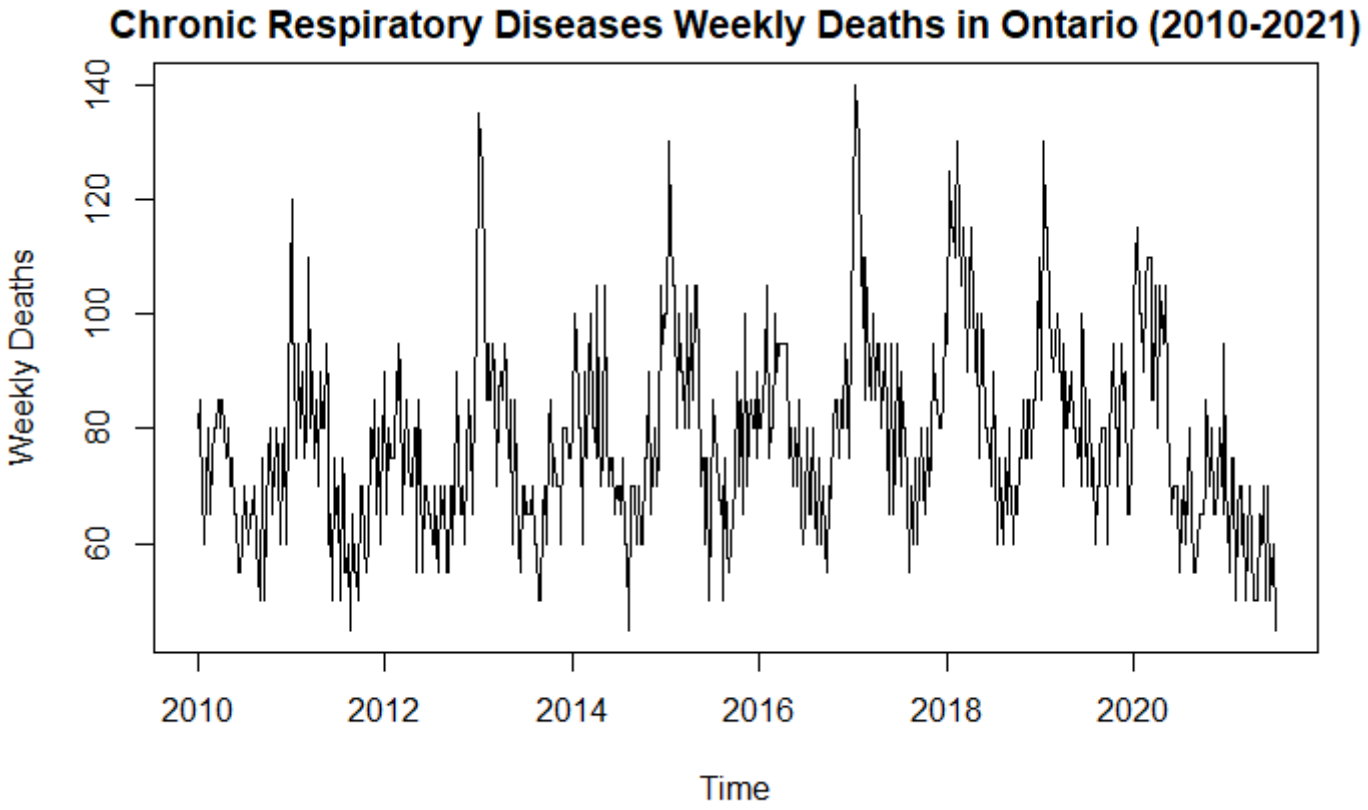
Hide

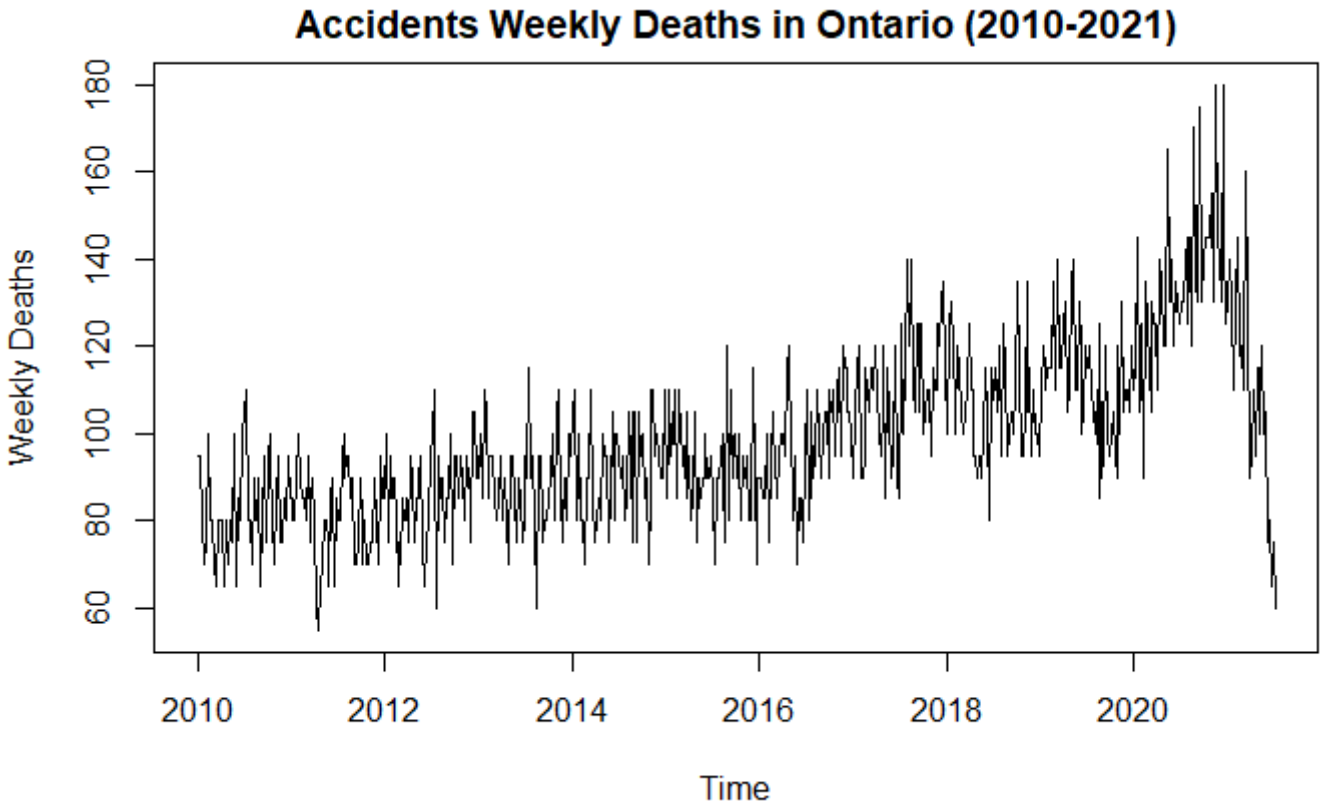
```
Malignant.neoplasms.ts = ts(DeathData$Malignant.neoplasms[1:numDataPoints],
                            start = startYear,
                            frequency = 52)

Diseases.of.heart.ts = ts(DeathData$Diseases.of.heart[1:numDataPoints],
                          start = startYear,
                          frequency = 52)

Cerebrovascular.diseases.ts = ts(DeathData$Cerebrovascular.diseases[1:numDataPoints],
                                 start = startYear,
                                 frequency = 52)

resp.ts = ts(DeathData$Chronic.lower.respiratory.diseases[1:numDataPoints],
             start = startYear,
             frequency = 52)

Accidents.ts = ts(DeathData$Accidents..unintentional.injuries.[1:numDataPoints],
                  start = startYear,
                  frequency = 52)

noInfo.ts = ts(DeathData$Information.unavailable[1:numDataPoints],
               start = startYear,
               frequency = 52)

other.ts = ts(DeathData$Other[1:numDataPoints],
              start = startYear,
              frequency = 52)

causes.ts = list(
            list("Malignant Neoplasms", Malignant.neoplasms.ts),
            list("Heart Disease", Diseases.of.heart.ts),
            list("Cerebrovascular Diseases",Cerebrovascular.diseases.ts),
            list("Chronic Respiratory Diseases",resp.ts),
            list("Accidents",Accidents.ts),
            list("No Information",noInfo.ts),
            list("Other",other.ts)
)

numCauses = 7

# plot the time series
for (i in 1:numCauses){
  plot(causes.ts[[i]][[2]], main = paste(causes.ts[[i]][[1]], "Weekly Deaths in Ontario
 (2010-2021)"), ylab = "Weekly Deaths")
}
```

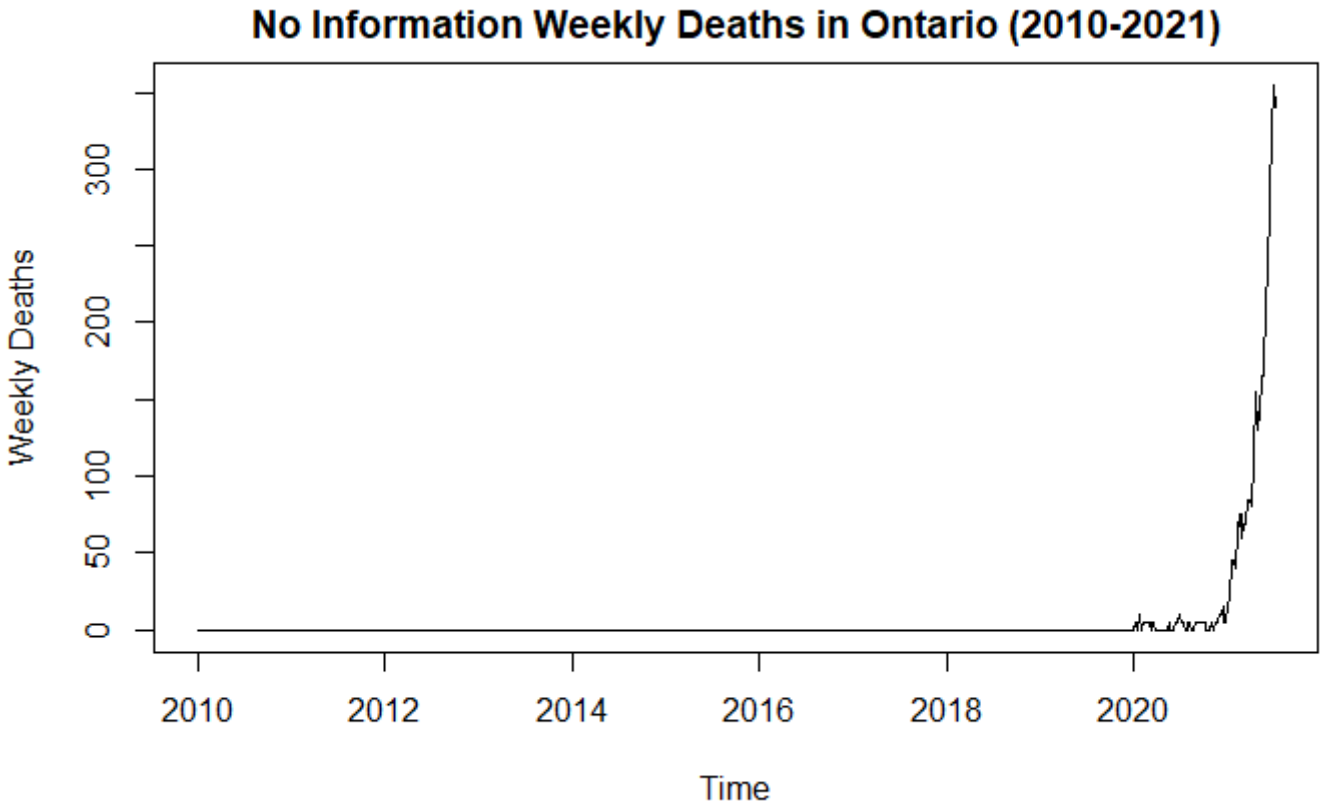

Malignant Neoplasms Weekly Deaths in Ontario (2010-2021)
Weekly Deaths
450
500
550
600
2010
2012
2014
2016
2018
2020
Time

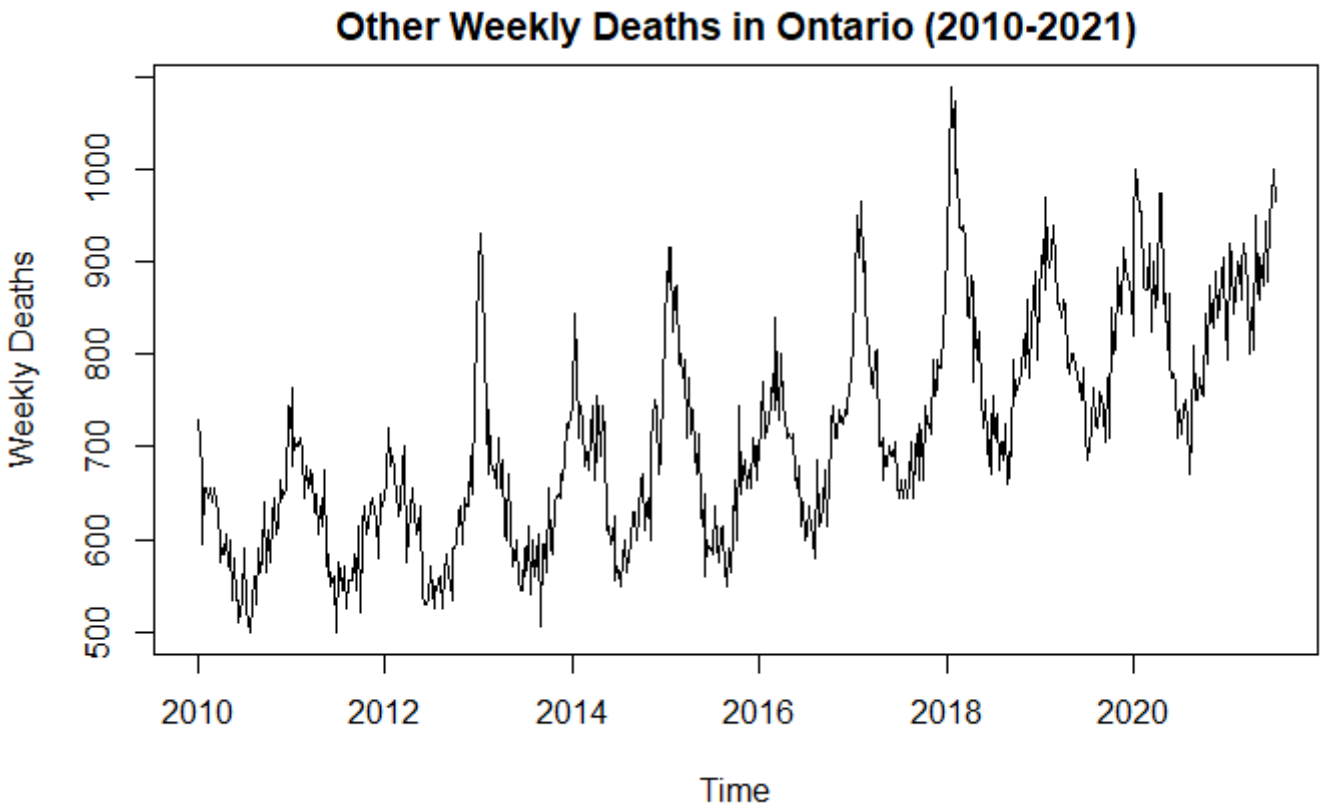

Heart Disease Weekly Deaths in Ontario (2010-2021)
Weekly Deaths
250
300
350
400
450
500
2010
2012
2014
2016
2018
2020
Time


Cerebrovascular Diseases Weekly Deaths in Ontario (2010-2021)
Weekly Deaths
60
80
100
120
2010
2012
2014
2016
2018
2020
Time


Chronic Respiratory Diseases Weekly Deaths in Ontario (2010-2021)
Weekly Deaths
60
80
100
120
140
2010
2012
2014
2016
2018
2020
Time


Accidents Weekly Deaths in Ontario (2010-2021)
Weekly Deaths
60
80
100
120
140
160
180
2010
2012
2014
2016
2018
2020
Time

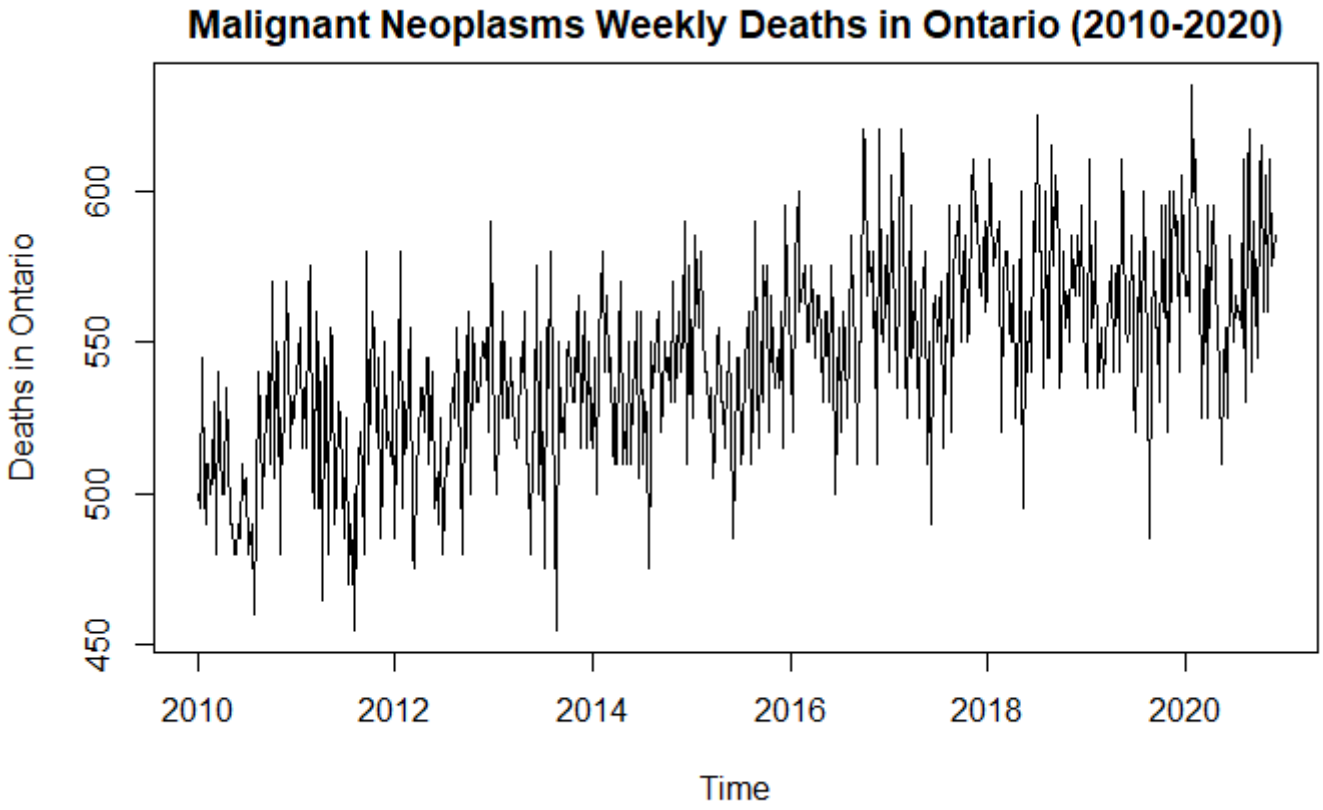

No Information Weekly Deaths in Ontario (2010-2021)
Weekly Deaths
0
50
100
200
300
2010
2012
2014
2016
2018
2020
Time

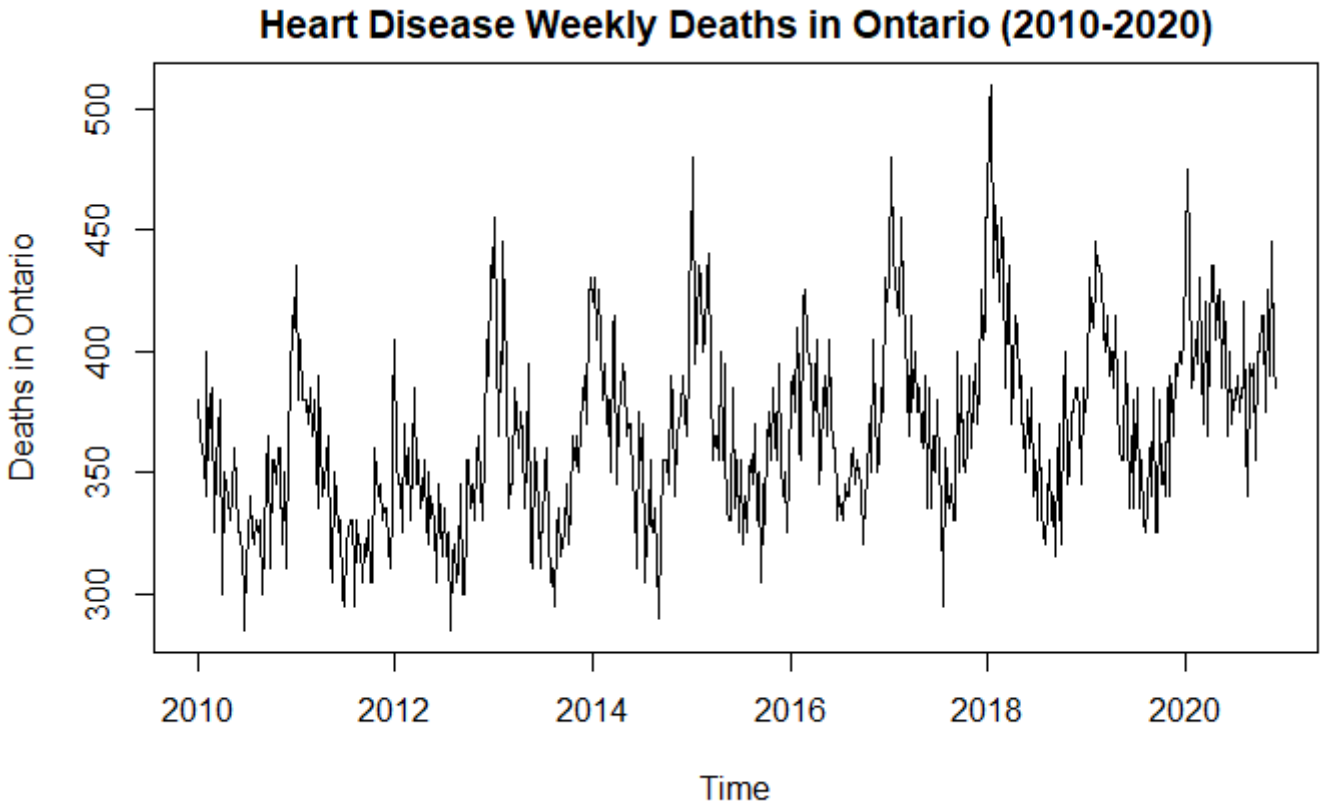


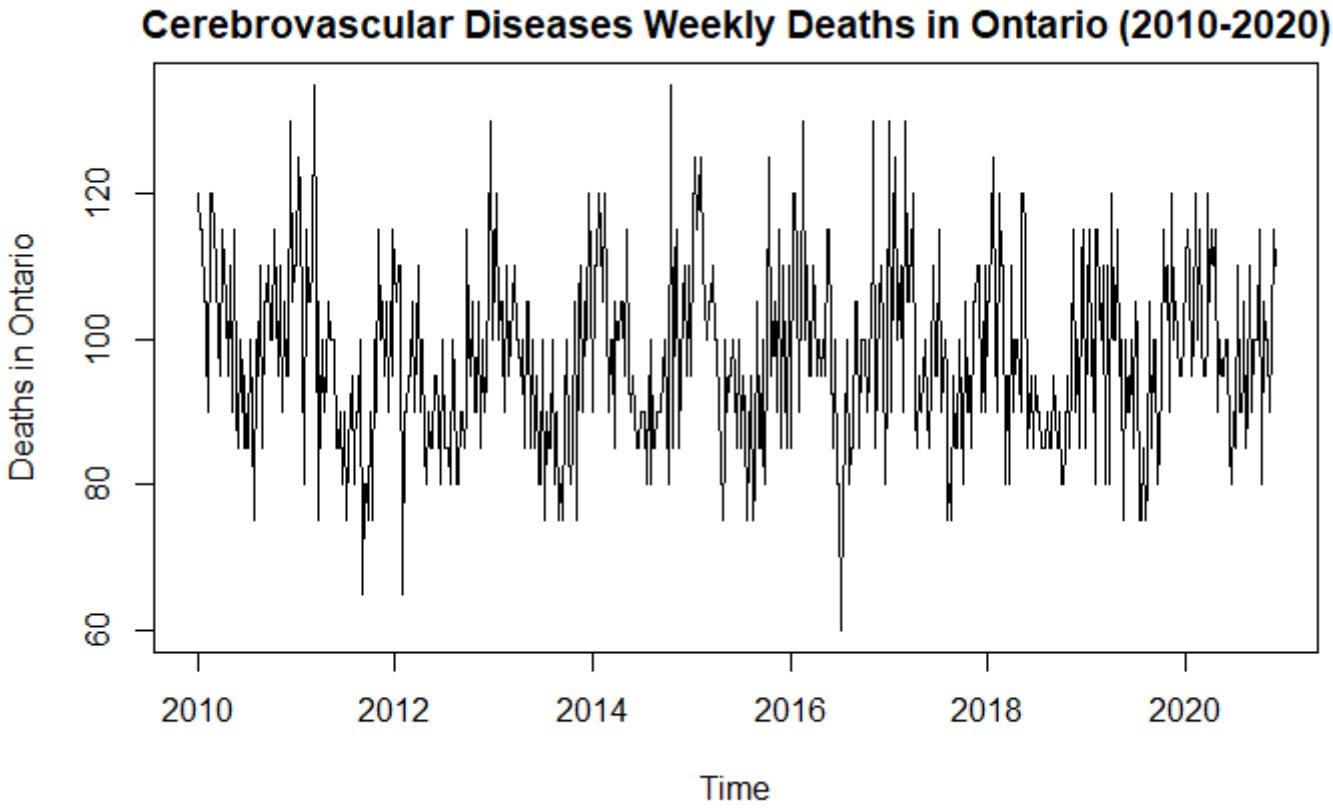
In 2021, there is a clear increase in cause “no information”. This makes sense; the more recent a death, the less likely that a coroner’s report has been completed. Therefore, for a fair analysis, we will restrict these causes of death to 2020.

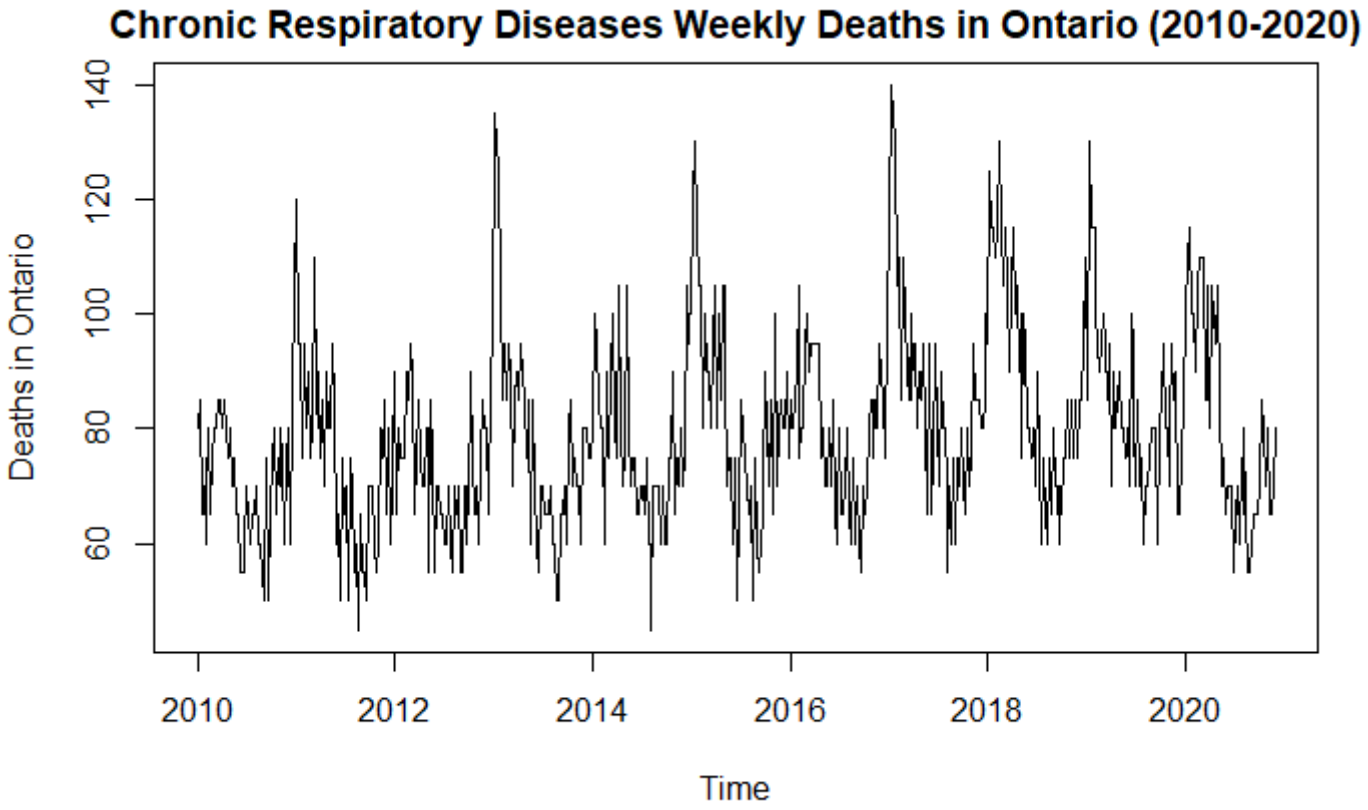
Hide

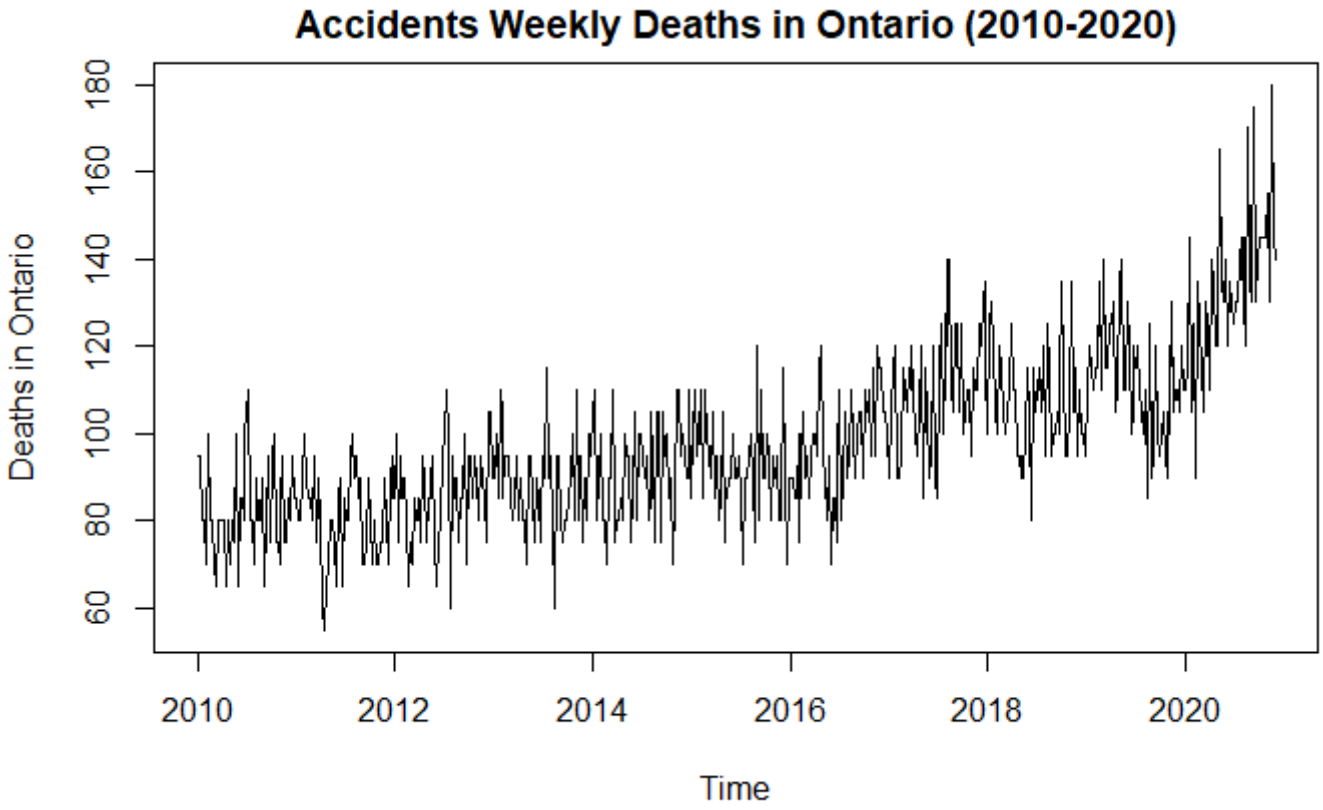
```
Malignant.neoplasms.ts = ts(DeathData$Malignant.neoplasms[1:(numDataPoints-26-5)],
                            start = startYear,
                            frequency = 52)

Diseases.of.heart.ts = ts(DeathData$Diseases.of.heart[1:(numDataPoints-26-5)],
                          start = startYear,
                          frequency = 52)

Cerebrovascular.diseases.ts = ts(DeathData$Cerebrovascular.diseases[1:(numDataPoints-26-
5)],
                                 start = startYear,
                                 frequency = 52)

resp.ts = ts(DeathData$Chronic.lower.respiratory.diseases[1:(numDataPoints-26-5)],
             start = startYear,
             frequency = 52
)

Accidents.ts = ts(DeathData$Accidents..unintentional.injuries.[1:(numDataPoints-26-5)],
               start = startYear,
               frequency = 52
)

#no information doesn't have enough data to be decomposed, and is therefore excluded

other.ts = ts(DeathData$Other[1:(numDataPoints-26-5)],
              start = startYear,
              frequency = 52)

causes.ts = list(
            list("Malignant Neoplasms", Malignant.neoplasms.ts),
            list("Heart Disease", Diseases.of.heart.ts),
            list("Cerebrovascular Diseases",Cerebrovascular.diseases.ts),
            list("Chronic Respiratory Diseases",resp.ts),
            list("Accidents",Accidents.ts),
            list("Other",other.ts)
)

numCauses = 6

# plot the time series
for (i in 1:numCauses){
  plot(causes.ts[[i]][[2]], main = paste(causes.ts[[i]][[1]], "Weekly Deaths in Ontario
 (2010-2020)"), ylab = "Deaths in Ontario")
}
```

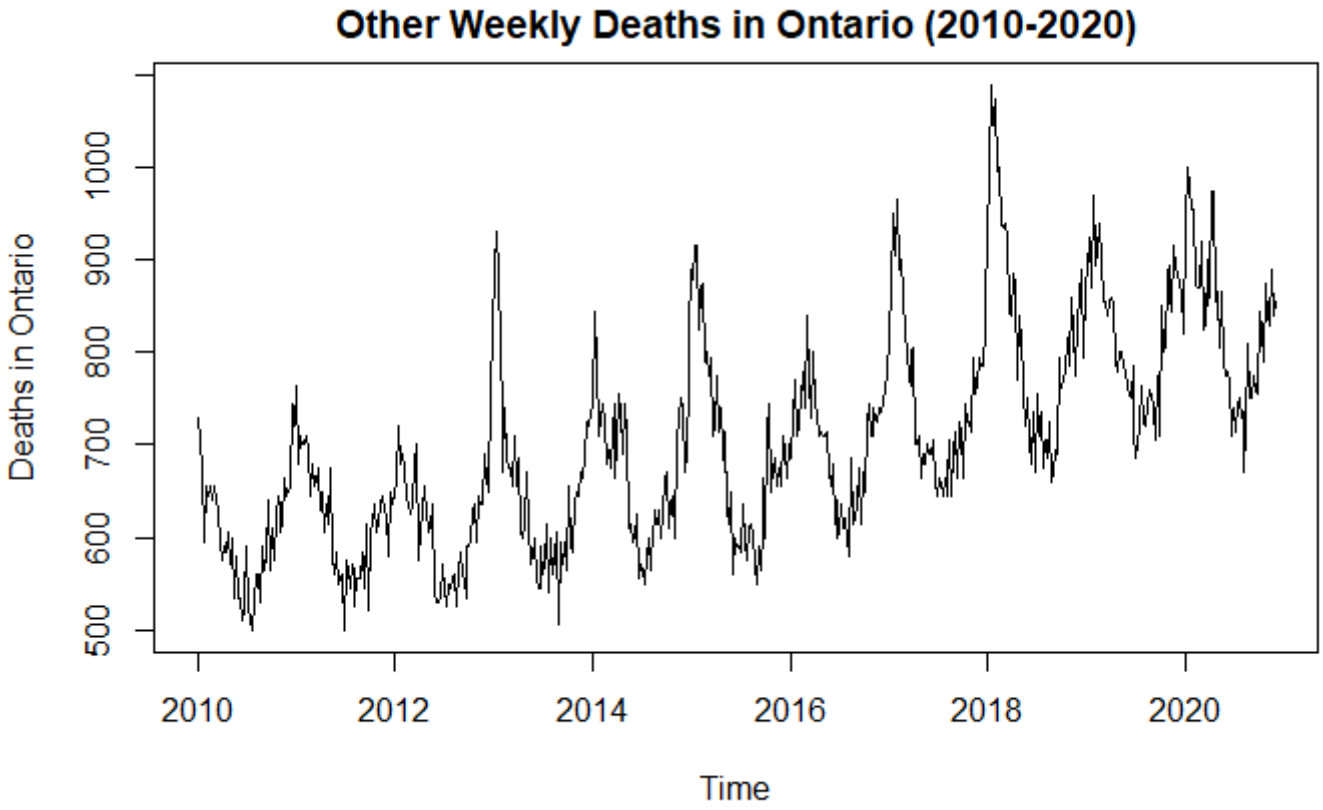

Malignant Neoplasms Weekly Deaths in Ontario (2010-2020)
Deaths in Ontario
450
500
550
600
2010
2012
2014
2016
2018
2020
Time


Heart Disease Weekly Deaths in Ontario (2010-2020)
Deaths in Ontario
300
350
400
450
500
2010
2012
2014
2016
2018
2020
Time


Cerebrovascular Diseases Weekly Deaths in Ontario (2010-2020)
Deaths in Ontario
60
80
100
120
2010
2012
2014
2016
2018
2020
Time


Chronic Respiratory Diseases Weekly Deaths in Ontario (2010-2020)
Deaths in Ontario
60
80
100
120
140
2010
2012
2014
2016
2018
2020
Time





All causes of death seem to have trend and seasonality, except for Accidents, which seems to have trend but no seasonality, and cerebrovascular deseases, which don't appear to have trend (this may indicate that their prevalence is decreasing at the rate of population increase). Interestingly, there is a clear trend and seasonality in the "Other" cause of death.

For confirmation, we will decompose each time series

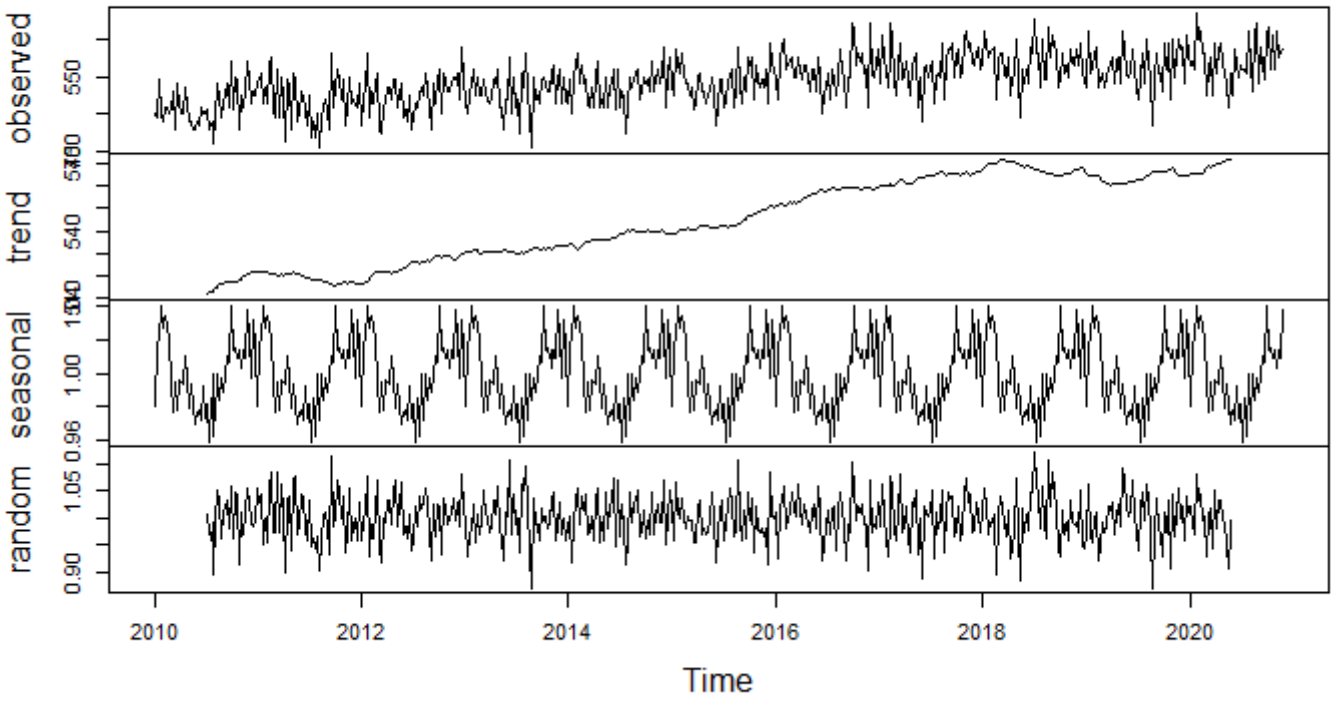
Hide

```
# helper function
customTitleDecompPlot = function (x, title = "", ...)
{
  xx <- x$x
  if (is.null(xx))
    xx <- with(x, if (type == "additive")
      random + trend + seasonal
      else
        random * trend * seasonal)
  plot(
    cbind(
      observed = xx,
      trend = x$trend,
      seasonal = x$seasonal,
      random = x$random
    ),
    main = title,
    ...
  )
}
```

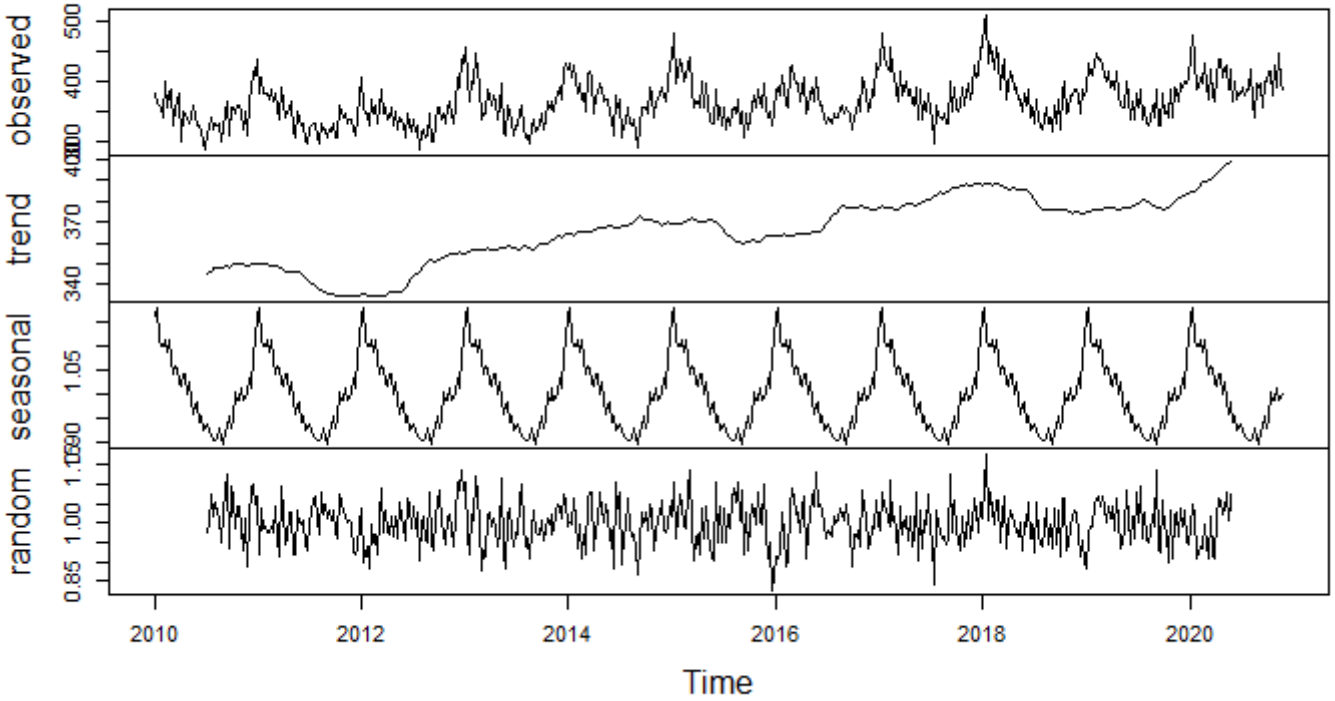
Hide

```
# plot the time series
decomposed = list()
for (i in 1:numCauses){
  decomposedCause = decompose(causes.ts[[i]][[2]], type = "multiplicative")
  decomposed[[i]]=decomposedCause
  customTitleDecompPlot(decomposed[[i]],causes.ts[[i]][[1]])
}
```

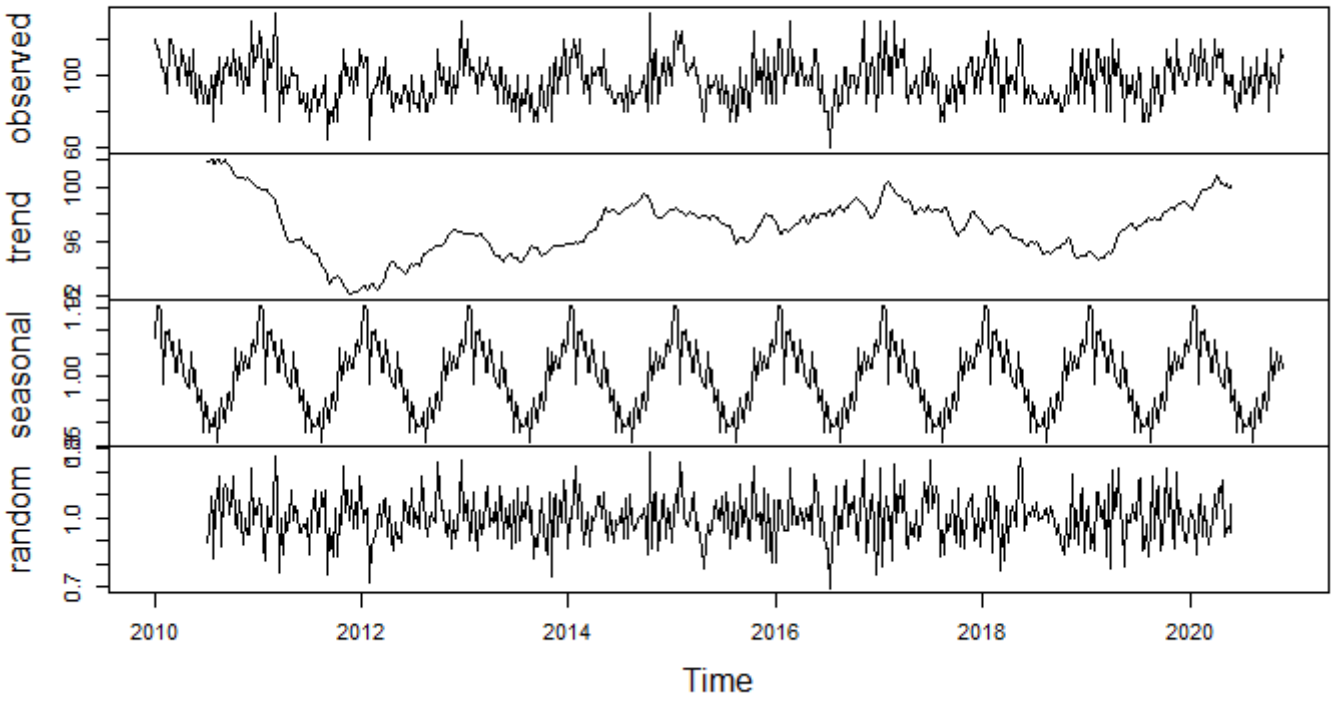
Malignant Neoplasms



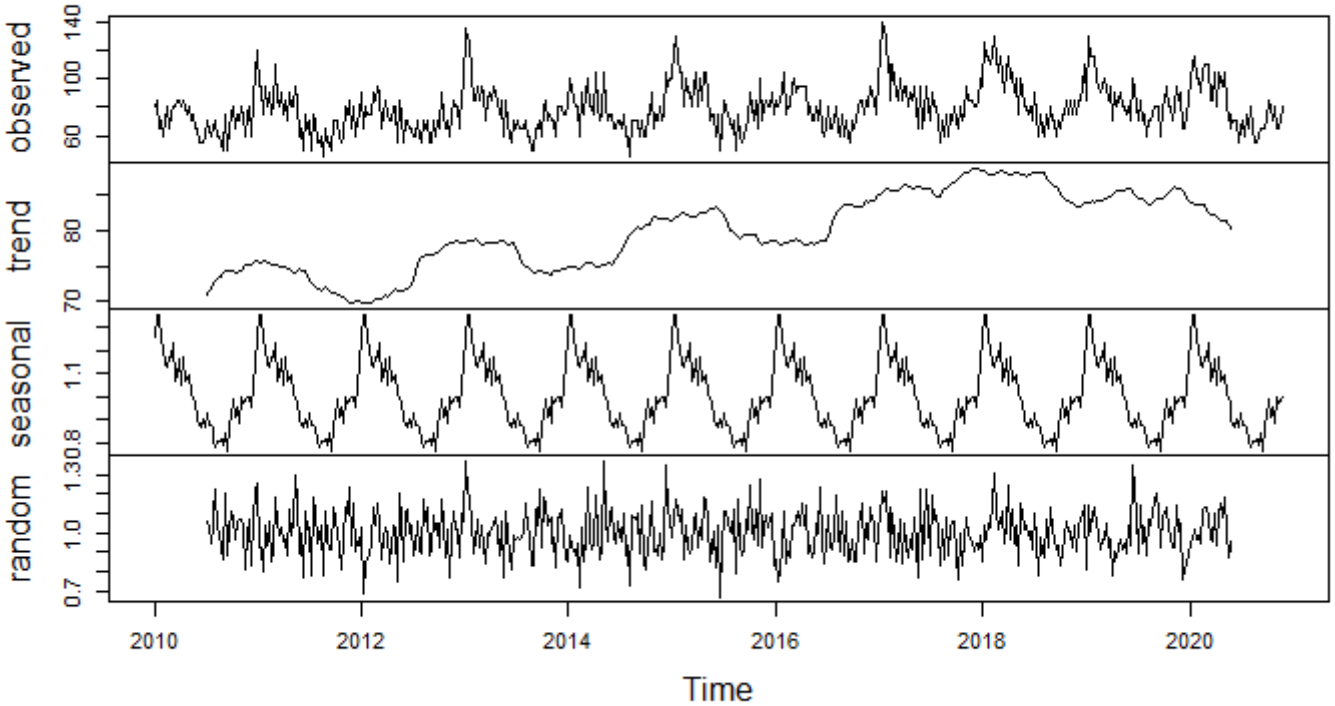
Heart Disease

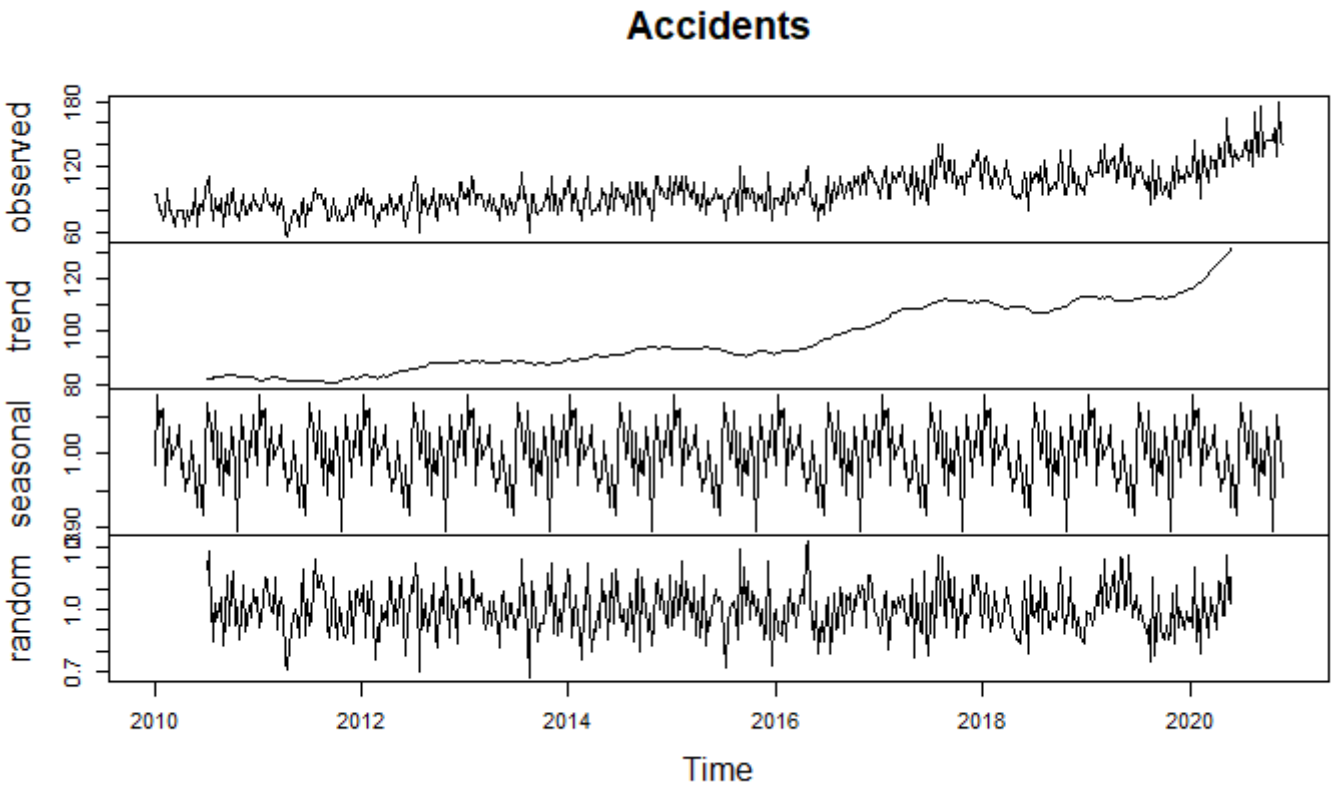


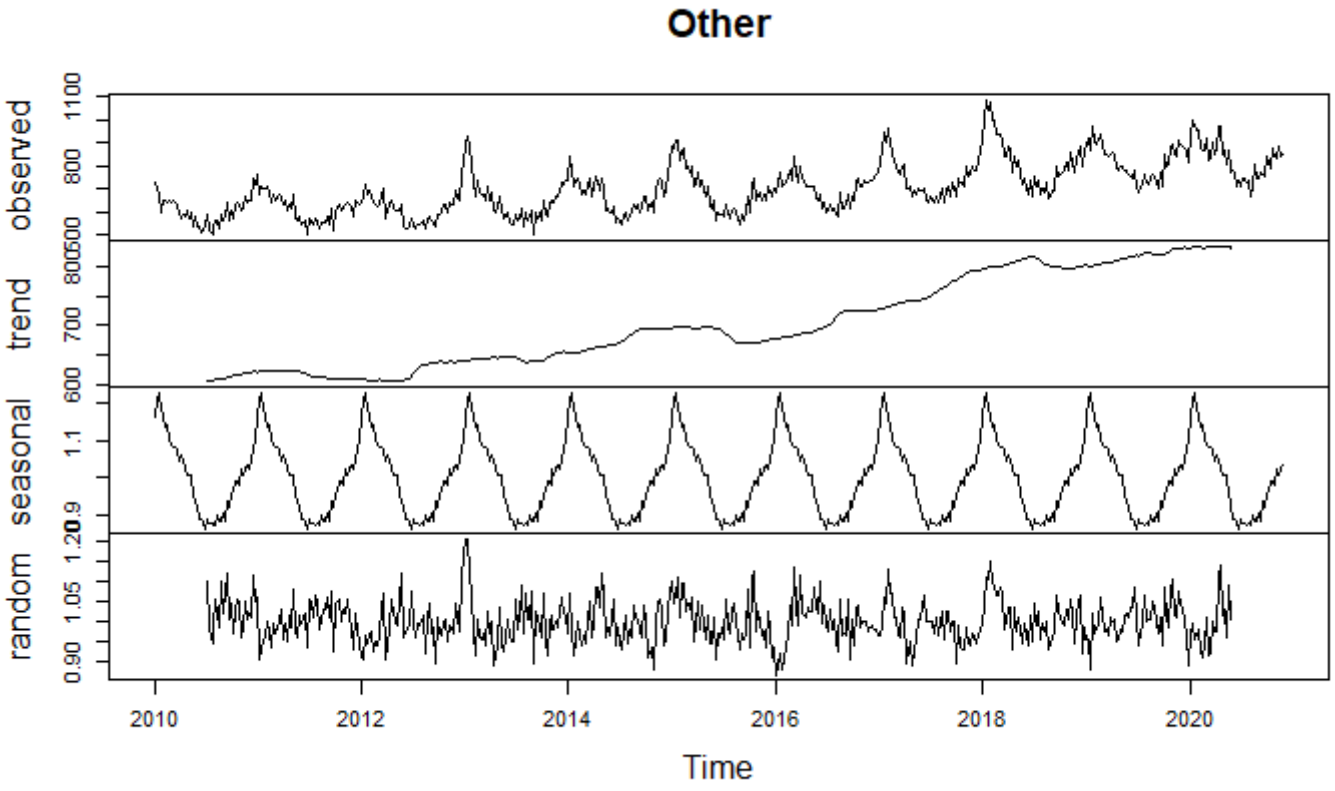
Cerebrovascular Diseases



Chronic Respiratory Diseases







Implications:

- Some causes of death did not see a significant difference when COVID started. For example, malignant neoplasms did not see significant changes in their trendlines when COVID began

- Some causes of death, especially those that are due to acute episodes that must be treated immediately, increased, likely because people were not getting needed treatment due to hospital strain or fear of contracting COVID at the hospital. For example, Heart disease and accidents, both of which are likely to result in death from sudden acute episodes (physical trauma, heart attacks), saw increases in deaths
- Chronic respiratory deaths saw a marked decrease. This was likely due to COVID outcompeting other respiratory viruses, as well as people with existing respiratory conditions dying of COVID due to their weakened respiratory systems
- A final, interesting implication is the trendline in “other” deaths became flat with the onset of COVID. Considering that generally the trend of weekly deaths should be increasing as population increases, this may indicate that there is another cause of death that was decreasing, but that is not visible, in the “other” category. This may be acute respiratory diseases such as the flu, which were clearly less prevalent during the COVID-19 pandemic

Objective 3: Evaluate Efficacy of Lockdowns

In this section, we evaluate the effectiveness of lockdowns by comparing lockdown stringency in Ontario to deaths, and analyzing potential correlation.

Create excess deaths data frame to use with stringency index data

Hide

```
DeathData.COVID = DeathData[c(522:(599-5)),c(10:11)]    # take only COVID data
rownames(DeathData.COVID) <- 1:73                    # re-index row names
DeathData.COVID$Actual <- DeathData.2020_2021
DeathData.COVID$Predicted <- actualArima.fcast$mean
DeathData.COVID$Excess <- DeathData.2020_2021 - actualArima.fcast$mean
colnames(DeathData.COVID) <- c("Year","Month","Actual Deaths","Predicted Deaths","Excess Deaths")
DeathData.COVID
```

	Year <dbl>	Month <dbl>	Actual Deaths <int>	Predicted Deaths <dbl>	Excess Deaths <dbl>
1	2020	1	2385	2435.396	-50.395693
2	2020	1	2350	2431.162	-81.161871
3	2020	1	2275	2390.894	-115.893784
4	2020	1	2270	2373.437	-103.437163
5	2020	2	2190	2336.339	-146.338855
6	2020	2	2185	2363.662	-178.661560
7	2020	2	2230	2313.134	-83.134363
8	2020	2	2165	2270.707	-105.707337
9	2020	2	2065	2204.479	-139.478951
10	2020	3	2135	2196.371	-61.370725

1-10 of 73 rows Previous **1** 2 3 4 5 6 … 8 Next

Stringency index is a a measurement created by the Centre of Excellence on the Canadian Federation used to "measure the relative stringency of policy responses" to the COVID-19 pandemic. The index captures 13 measures by imposed by governments, including travel and gathering restrictions, mask mandates, amongst others.

Data Source: Centre of Excellence on the Canadian Federation - COVID-19 Stringency Index (https://centre.irpp.org/data/covid-19-provincial-policies/)

Load stringency index data and observe

Hide

```
StringencyData = read.csv("data/ontario_stringency_index_2020_2021.csv", fileEncoding =
"UTF-8-BOM") # get data
StringencyData.weekly = StringencyData[seq(1,nrow(StringencyData),7),] # select weekly d
ata

# create lagged stringencyIndex data
# lag by three weeks as 18.5 is the average number of days between first symptoms of COV
ID and death
# source: https://www.drugs.com/medical-answers/covid-19-symptoms-progress-death-353626
4/
laggedStringencyData.weekly <- StringencyData.weekly[c(rep(NA,3), 1:(nrow(StringencyDat
a.weekly)-3)),]
StringencyData.weekly$`Stringency Index -3 Weeks` <- laggedStringencyData.weekly$stringe
ncyIndex # add lagged data to df

StringencyData.weekly = StringencyData.weekly[c(1:73),]       # only take data with corr
esponding COVID data
rownames(StringencyData.weekly) <- 1:73                       # re-index row names

tail(StringencyData.weekly)
```

	date	stringencyIndex	Stringency Index -3 Weeks
	<chr>	<dbl>	<dbl>
68	2021-04-17	58.9	47.1
69	2021-04-24	73.2	54.2
70	2021-05-01	73.2	58.9
71	2021-05-08	73.2	58.9
72	2021-05-15	73.2	73.2
73	2021-05-22	73.2	73.2

6 rows

Add stringency index to the data frame

Hide

```
DeathData.COVID$`Stringency Index` <- StringencyData.weekly$stringencyIndex
DeathData.COVID$`Stringency Index -3 Weeks` <- StringencyData.weekly$`Stringency Index -
3 Weeks`
colnames(DeathData.COVID) <- c("Year","Month","Actual Deaths","Predicted Deaths","Excess
Deaths","Stringency Index","Stringency Index -3 Weeks")

DeathData.COVID = DeathData.COVID[c(14:73),]          # take data from the beginning of
lockdowns
rownames(DeathData.COVID) <- 1:60                     # re-index row names

tail(DeathData.COVID)
```

	Y...	Mo...	Actual Deaths	Predicted Deaths	Excess Deaths	Stringency Index
	<dbl>	<dbl>	<int>	<dbl>	<dbl>	<dbl>
55	2021	4	2155	2147.235	7.764682	58.9
56	2021	4	2185	2135.352	49.647802	73.2
57	2021	5	2310	2154.541	155.459015	73.2
58	2021	5	2245	2073.980	171.019512	73.2
59	2021	5	2175	2074.545	100.455385	73.2
60	2021	5	2145	2073.167	71.833340	73.2

6 rows | 1-7 of 7 columns

Hide

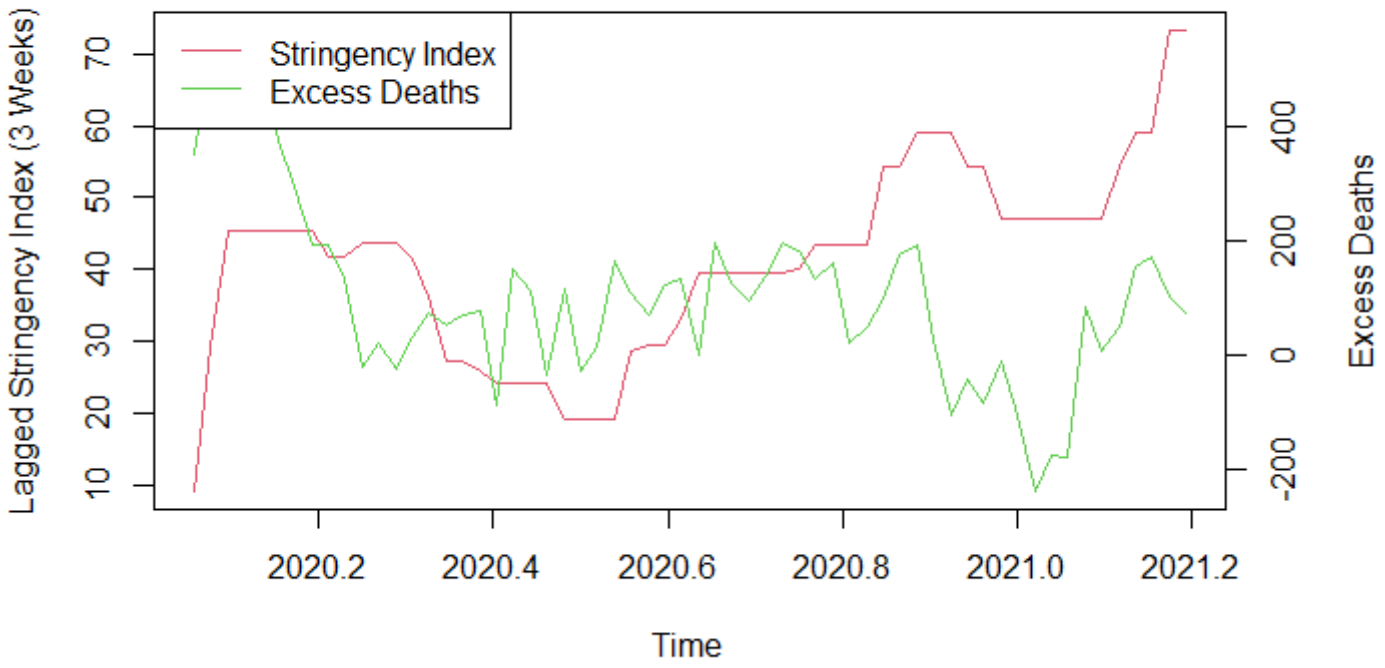
```
startYear = DeathData.COVID$Year[1]
startMonth = DeathData.COVID$Month[1]

ExcessDeaths.ts <- ts(DeathData.COVID$`Excess Deaths`, start=c(startYear,startMonth), fr
equency=52)
StringencyIndex.ts <- ts(DeathData.COVID$`Stringency Index -3 Weeks`, start=c(startYear,
startMonth), frequency=52)

par(mar = c(5, 4, 4, 4) + 0.3)                                          # additio
nal space for second y-axis
plot(StringencyIndex.ts, col = 2,ylab = "Lagged Stringency Index (3 Weeks)")   # create
 first plot
par(new = TRUE)                                                         # add new
plot
```

Hide

```
plot(ExcessDeaths.ts, col = 3,                                          # create
 second plot without axes
     axes = FALSE, xlab = "", ylab = "")
axis(side = 4, at = pretty(range(ExcessDeaths.ts)))                     # add sec
ond axis
```

Hide

```
mtext("Excess Deaths", side = 4, line = 3)                                    # add sec
ond axis label
legend("topleft", lty = 1, col = c(2,3), legend = c("Stringency Index", "Excess Deaths"
))
```

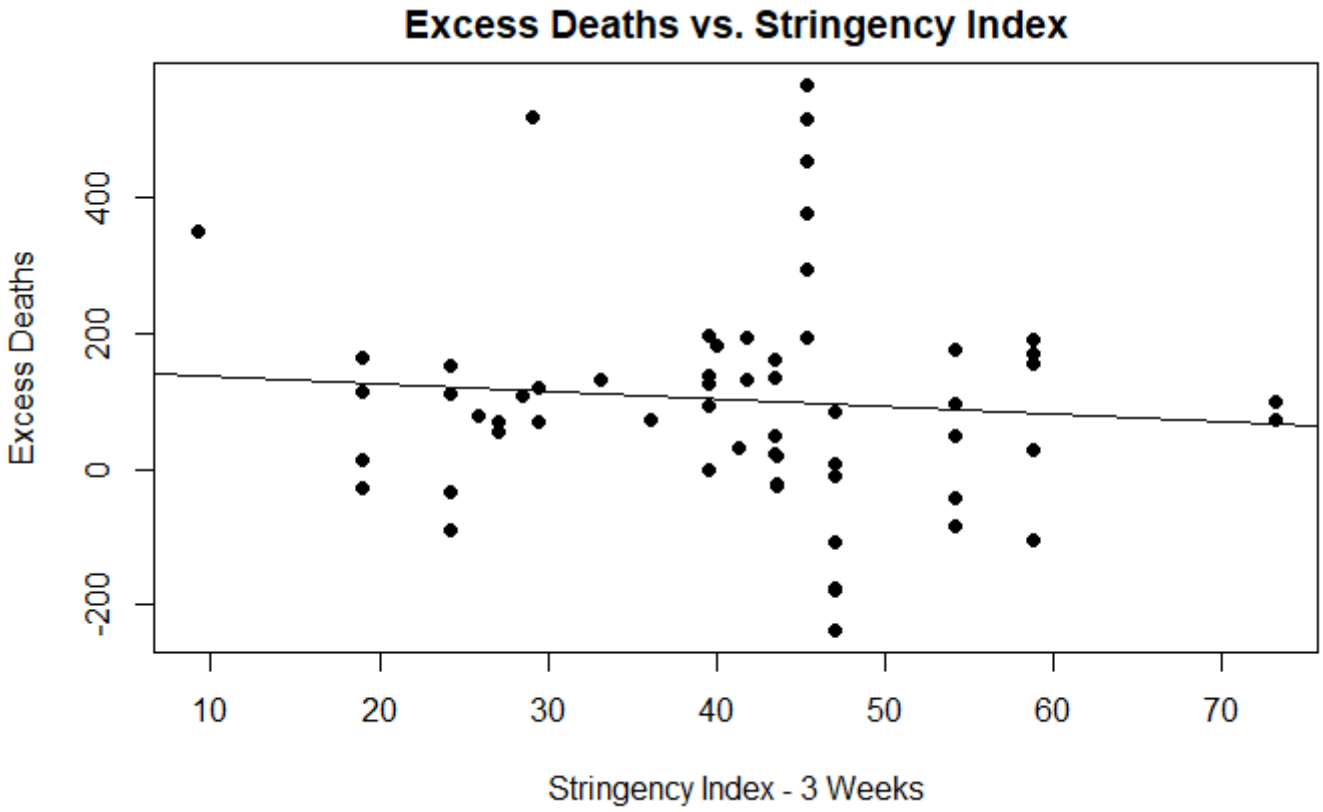


Visually, it appears that there is a negative correlation between the stringency of government measures and excess deaths.

Create correlation between stringency index and excess deaths

Hide

```
# plot correlation between variables
plot(DeathData.COVID$`Stringency Index -3 Weeks`,DeathData.COVID$`Excess Deaths`,
     pch=16,
     main="Excess Deaths vs. Stringency Index",
     xlab="Stringency Index - 3 Weeks",
     ylab="Excess Deaths")

# fit line
abline(lm(DeathData.COVID$`Excess Deaths` ~ DeathData.COVID$`Stringency Index -3 Weeks
`))
```



Observations:

- Scatter plot shows negative line of best fit but it does not appear to be a strong correlation.

Hide

```
cor.test(DeathData.COVID$`Excess Deaths`,DeathData.COVID$`Stringency Index -3 Weeks`, me
thod="pearson") # get correlation between variables
```

```
    Pearson's product-moment correlation

data:  DeathData.COVID$`Excess Deaths` and DeathData.COVID$`Stringency Index -3 Weeks`
t = -0.7033, df = 58, p-value = 0.4847
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.3379895  0.1658402
sample estimates:
        cor
-0.09195691
```

Observations:

P-value of correlation: ~.5

- As the p-value of the correlation is greater than 0.05, we can conclude that the relationship between excess deaths and stringency index is not statistically significant

Correlation coefficient: -0.09

- There is a negative correlation between the two variables meaning a higher stringency index results in less excess deaths
- The correlation coefficient can be anywhere between -1 and 1 and therefore, a correlation coefficient of -0.09 shows a weak relationship

In conclusion, stringency index and excess deaths on their own are not a good measure of lockdown effectiveness. However, the limited dataset, absence of other variables that significantly impact excess deaths, makes it difficult to determine the true relationship between the variables.

Conclusion

In conclusion, COVID-19 has caused a significant impact on mortality in the province of Ontario, including other causes of death. Notably, excess mortality was lower than the number of COVID deaths, indicating that the virus directly affected other causes of death. Analysis investigating the correlation between stringency of government response to the pandemic and excess deaths was inconclusive. Preliminary findings indicated a negative correlation, though the result was statistically insignificant. In the future, as data becomes more readily availble post-pandemic, the team would like to re-visit this analysis as a next step.

Implications for the long term:

- Although the data for lockdown efficacy was inconclusive due to a high p-value, it still showed a negative correlation between excess deaths and lockdowns. With more data as the pandemic continues, this may be able to provide strong evidence either for or against the effectiveness of lockdowns.
- The decompositions in objective 2 show the importance of ensuring there is sufficient hospital capacity, and the importance of ensuring that people continue to seek medical care, even during periods of lockdown. This can inform public health decision making going forward
- Lastly, the decrease in deaths from other (inc. flu) and chronic respiratory diseases may show that measures like social distancing and masking are effective in slowing the spread of non-COVID respiratory viruses; this can be used as evidence to encourage their localized use in high-risk settings in the future.