BUAN 6530 Business forecasting and predictive analytic

Mark Drummond

Fall 2020 - Final Projet

```r
library(fpp2)
```

```
## Loading required package: ggplot2

## Loading required package: forecast

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

## Loading required package: fma

## Loading required package: expsmooth
```

```r
# Display Working Directory and System Time
getwd()
```

```
## [1] "C:/Mark/Grad School/Classes/BUAN 6530 Forcasting/Rcode"
```

```r
Sys.time()
```

```
## [1] "2020-12-13 07:41:23 EST"
```

```r
#You may run the code by deleting "#" below the 1st time if you cannot knit a pdf file
#tinytex::install_tinytex()
```

```r
#install.packages("tseries")
#install.packages("quantmod")
#install.packages("ggplot2")
#install.packages("xlsx")
#library(tseries)
#library(quantmod)
#library(ggplot2)
#library(forecast)
library(fma)
library(Mcomp)
library(smooth)
```

```
## Loading required package: greybox

## Package "greybox", v0.6.2 loaded.
## Did you know that you can use your own loss function in alm()? This is regulated with 'loss' paramete

## This is package "smooth", v2.6.0
```

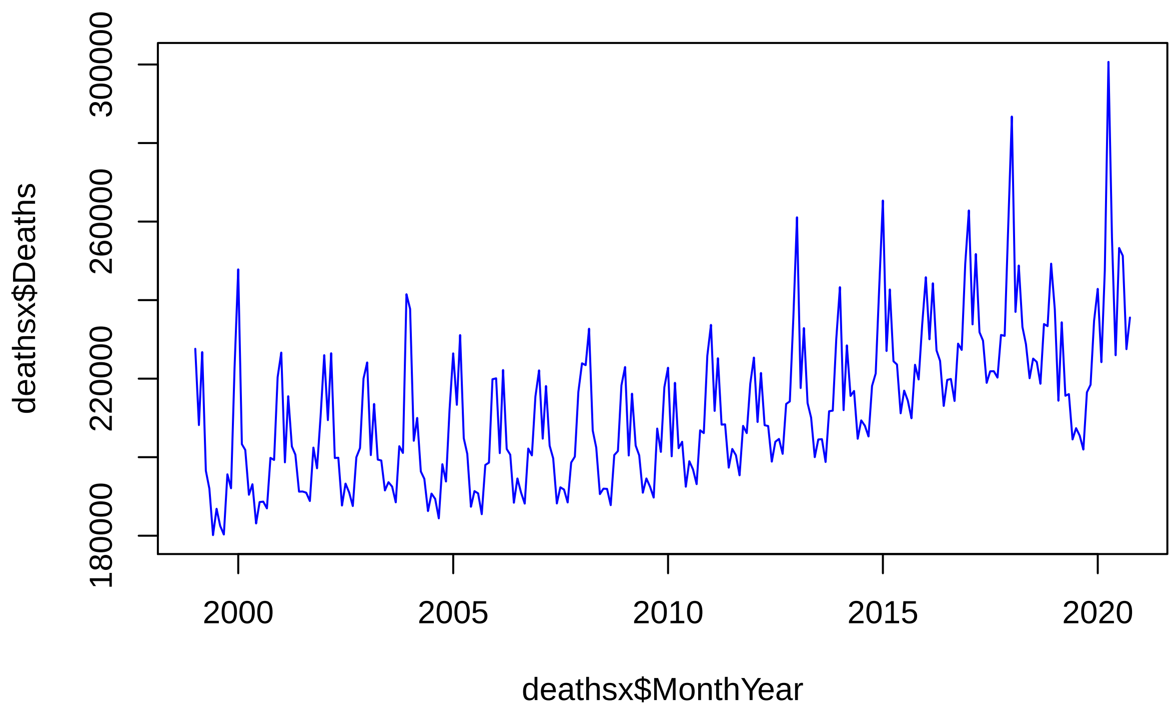
Load data from File

```
# a
# Monthly total Deaths from 1/1/1999 to 10/1/2020
deathsx <- readxl::read_excel("Data Sets/ALL Deaths 1999 2020.xlsx")
```

```
# b
head(deathsx)
```

```
## # A tibble: 6 x 3
##   MonthYear           Deaths `COV-Deaths`
##   <dttm>               <dbl>        <dbl>
## 1 1999-01-01 00:00:00 227604           NA
## 2 1999-02-01 00:00:00 208174           NA
## 3 1999-03-01 00:00:00 226765           NA
## 4 1999-04-01 00:00:00 196544           NA
## 5 1999-05-01 00:00:00 191982           NA
## 6 1999-06-01 00:00:00 180153           NA
```

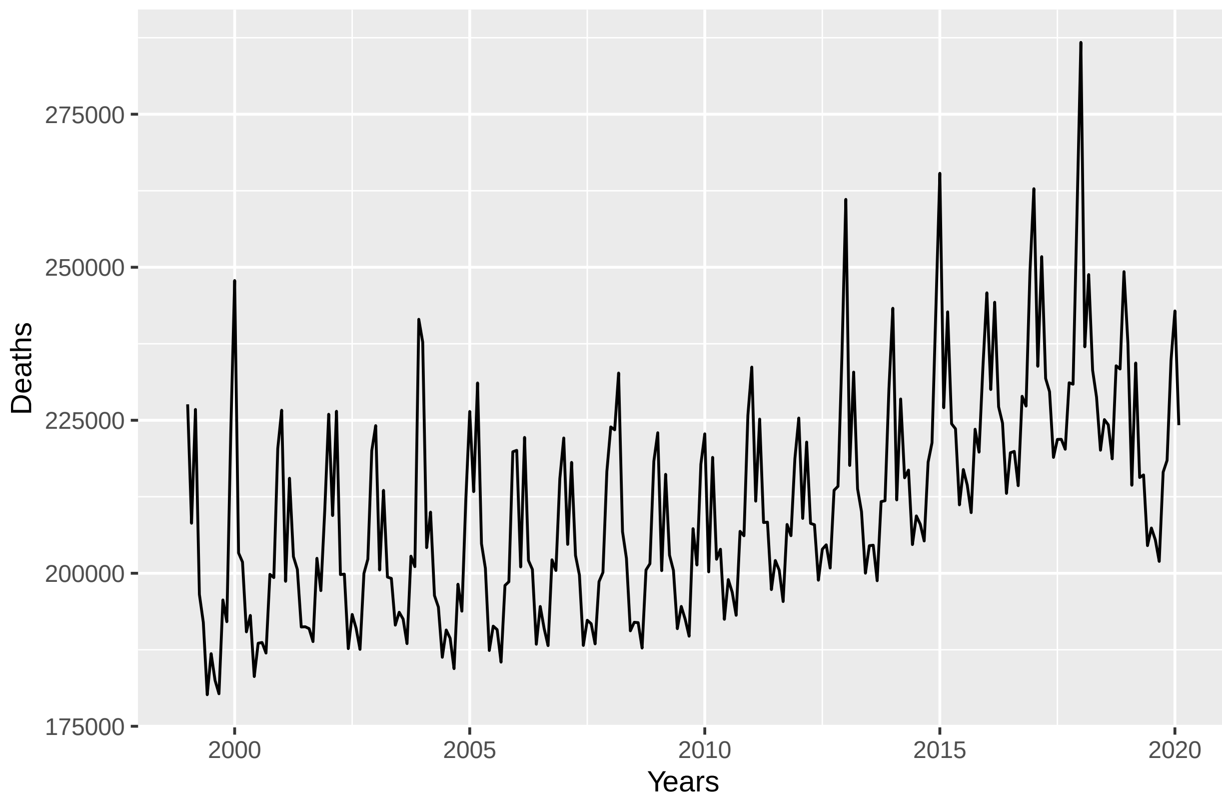
```
plot(deathsx$MonthYear, deathsx$Deaths, col="blue", type = "l", lwd=1)
```



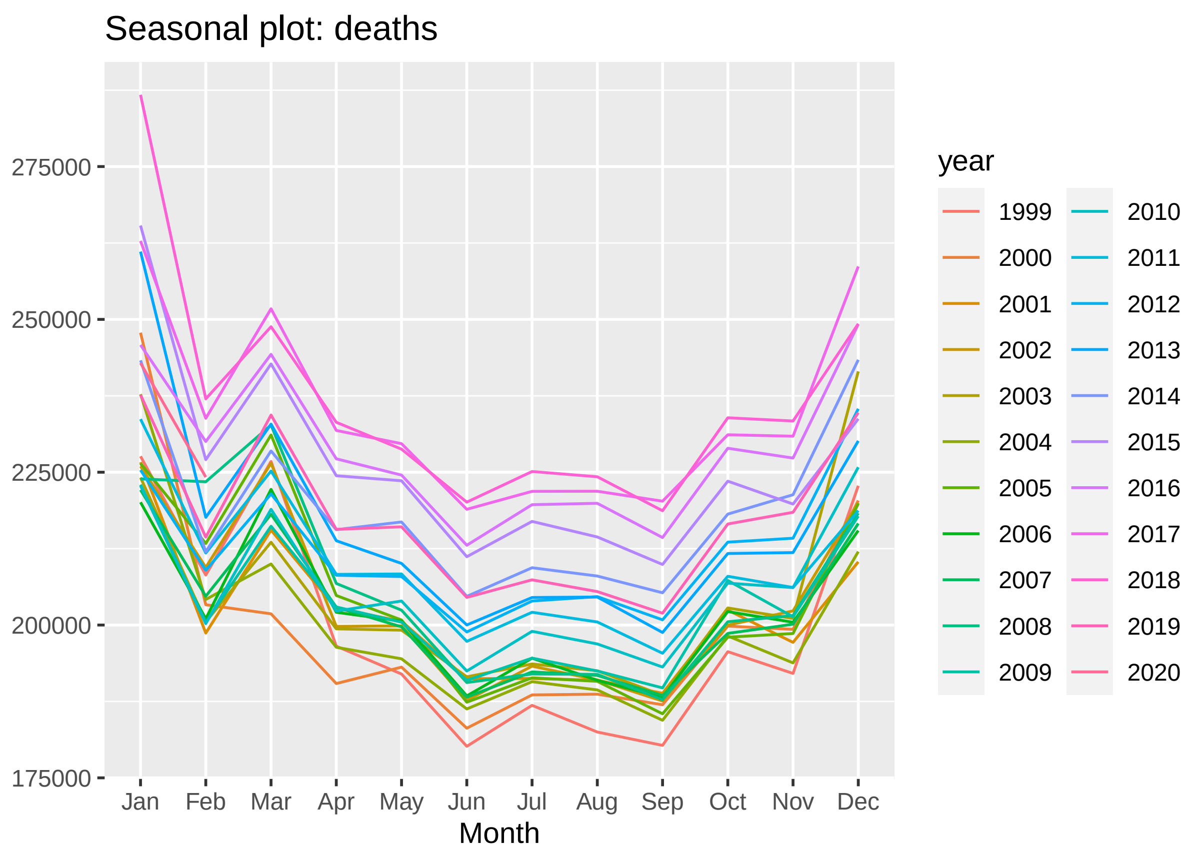

```r
# Convert the data into time series
# limit to before any Covid Deaths were reported
death_ts <- ts(deathsx[,-1], start=1999,end=c(2020,2),  frequency=12)
#death_ts <- ts(deathsx[,-1], frequency=12)
#  Name death_ts[, "Deaths"] as 'death'
deaths <- death_ts[, 'Deaths']
```

```r
#head(deaths)
#View(deaths)
```

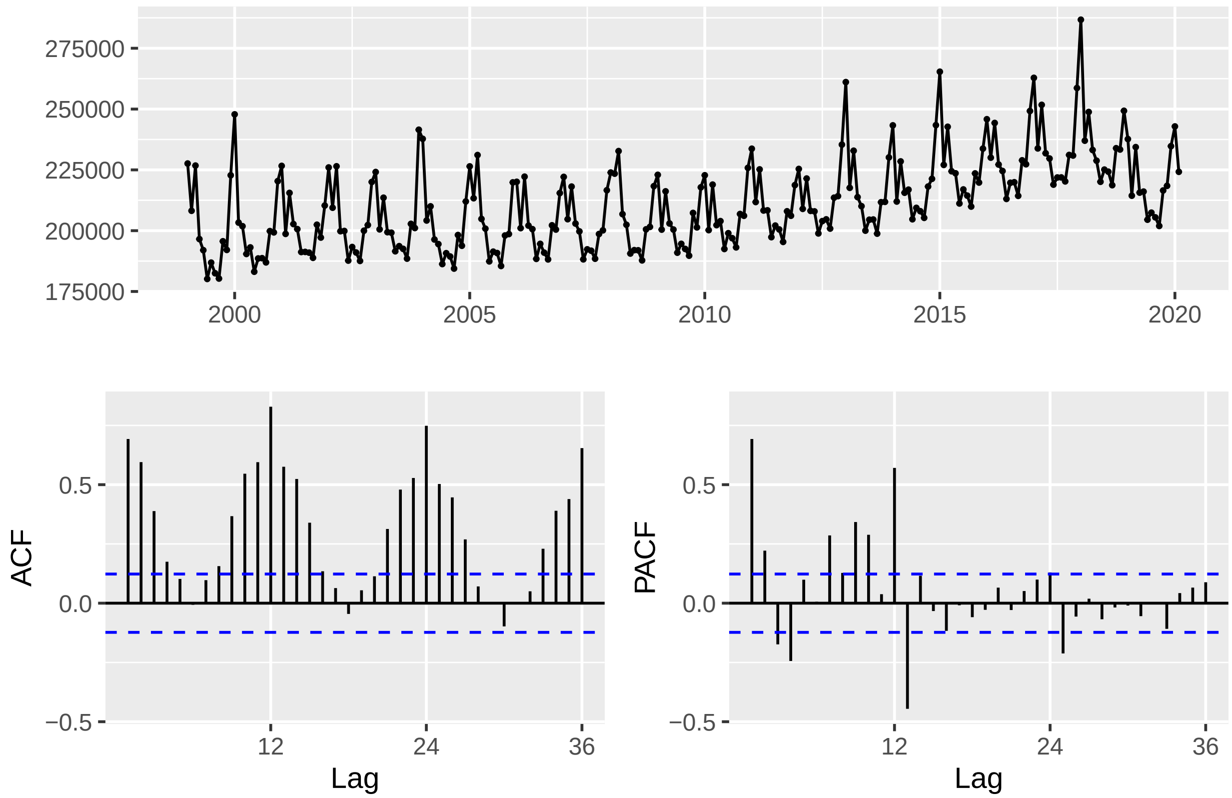
```r
# Plot 'death' and identify  any seasonal pattern and stationarity
autoplot(deaths, xlab = 'Years', ylab = "Deaths")
```




```
# look at the seasonal plot
ggseasonplot(deaths)
```




```
# stationarity,ACF and PACF
ggtsdisplay(deaths)
```




```r
# Add additional commentary here
# The PACF plots show that there are significant lags at 1 and 13.
# The AFC plot there is a geometric decay at each Lag.
# This would indicate that a seasonal AR model would be used
```

```r
# Get Box Cov
BoxCox.lambda(deaths)
```

```
## [1] 0.250792
```

```r
# For Non Seasonal
ndiffs(BoxCox(deaths,lambda=0.250792))
```

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

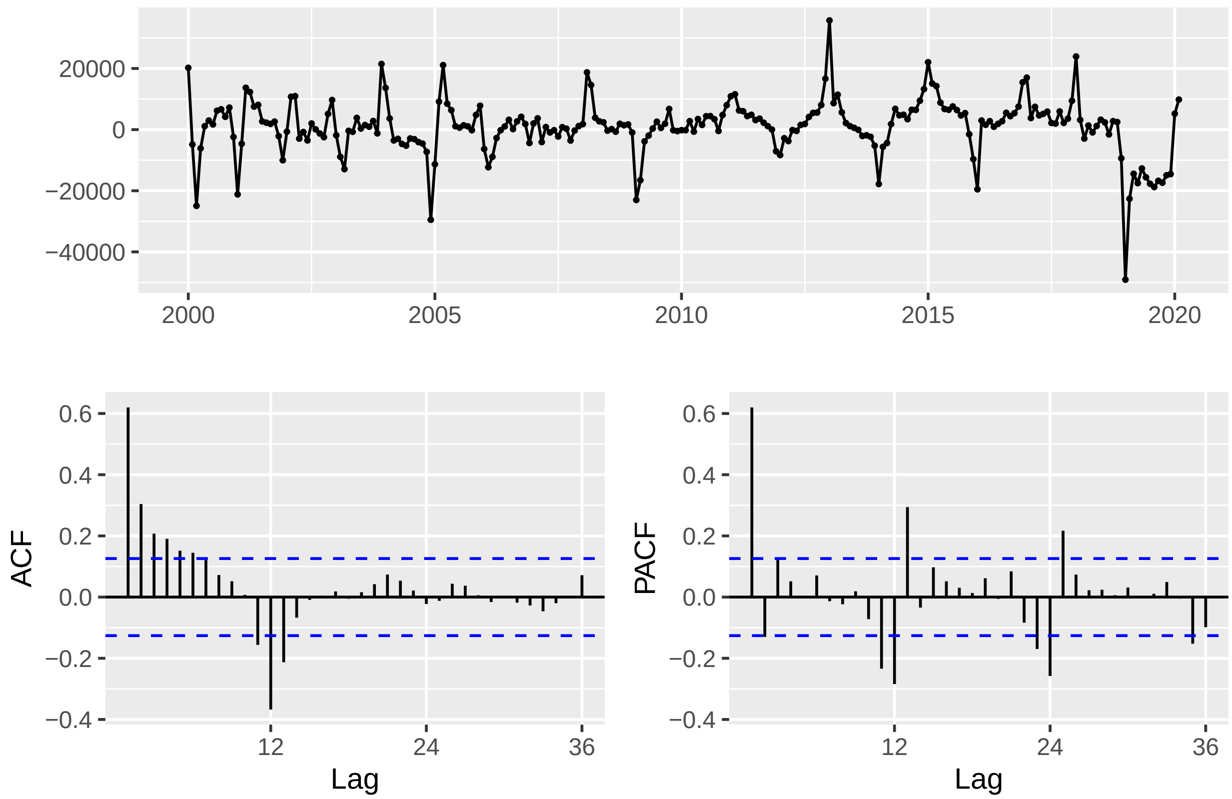
```r
# = 1 so the model needs a non-Seasonal differencing of 1
```

```r
# For Seasonal
# Gives me d for the Arima Model
# Difference once based on ndiff
nsdiffs(diff(BoxCox(deaths,lambda=0.250792),lag=12))
```

```
## [1] 0
```

```r
# = 0 so no seasonality needed
```

```r
ggtsdisplay(diff(deaths,lag=12))
```




```r
# Looks stationary now
# The AFC plot there is a geometric decay

# The PACF plots show that there are significant lags at 1 then minor ones at 12,13,24,25.
# This would indicate that a AR model with some seasonality should be used
```

```r
#   Comments on the seasonality, stationarity, and possible forecasting strategies
#   Appears to have yearly seasonality and a trend so you should take 1 difference
#   with the 1st difference then
#   Looks to be following a ARIMA(1,d,q) since
#   The spikes on in the ACF and PACF drops significantly after lag 12
#   and most of the other spikes are with the blue lines in the ACF plot
```

Section 2. Testing diffrent models

Section 2.1

Simple Moving Averages (SMA)

```r
# a
dea_sma1 <- sma(deaths,lambda=0.250792)
dea_sma2 <- sma(deaths,order=12,lambda=0.250792)
```

```r
#
# Summarize the sma models results and explain the parameters of the optimal model
accuracy(dea_sma1$fitted,deaths)
```

```
##                 ME     RMSE      MAE        MPE     MAPE      ACF1 Theil's U
## Test set 54.89173 13252.08 10389.01 -0.2061187 4.811705 0.1724512 0.9581079
```

```r
accuracy(dea_sma2$fitted,deaths)
```

```
##                ME     RMSE     MAE        MPE     MAPE      ACF1 Theil's U
## Test set 550.763 14255.52 10843.2 -0.1536632 5.036141 0.5046706  1.050097
```

```r
summary(dea_sma1)
```

```
## Time elapsed: 2.34 seconds
## Model estimated: SMA(2)
## Initial values were produced using backcasting.
##
## Loss function type: MSE; Loss function value: 175617507.125
## Error standard deviation: 13304.56
## Sample size: 254
## Number of estimated parameters: 2
## Number of degrees of freedom: 252
## Information criteria:
##      AIC     AICc      BIC     BICc
## 5546.711 5546.759 5553.785 5553.918
```

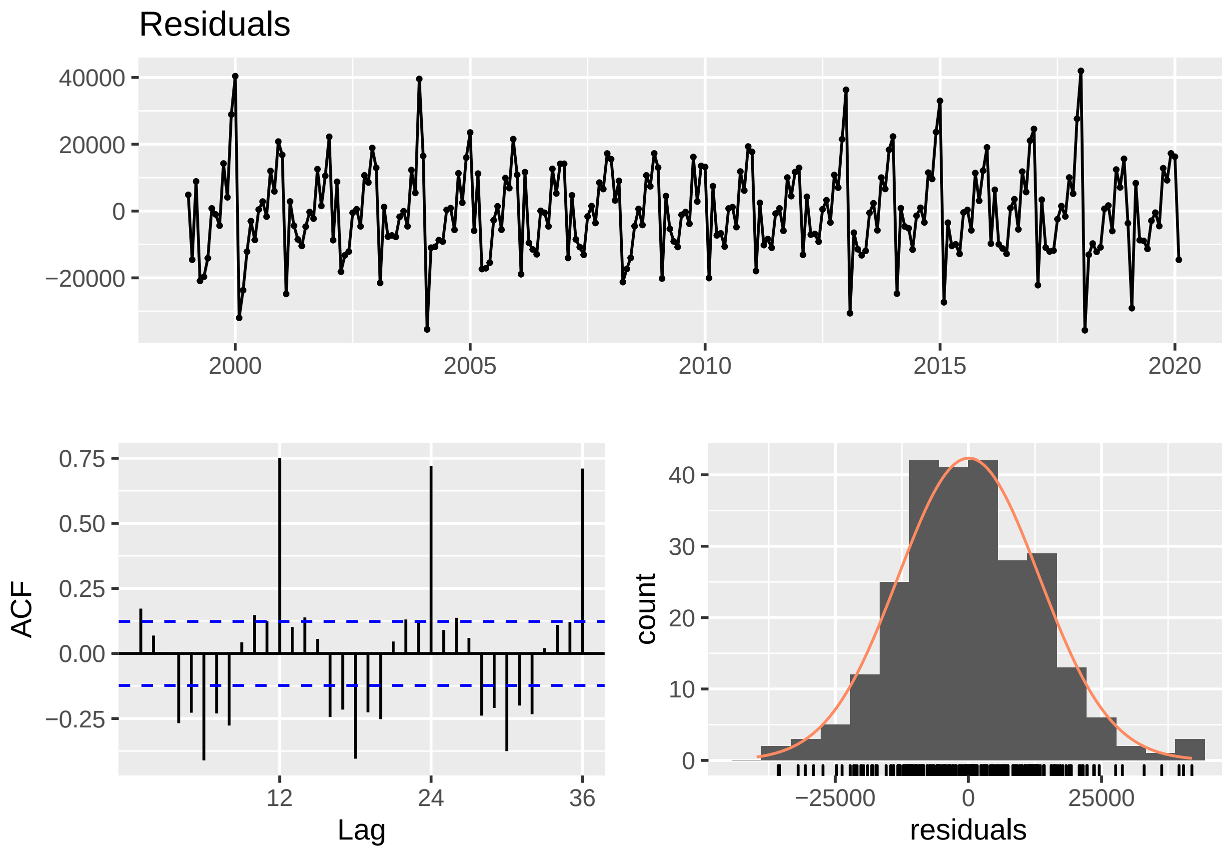
```r
summary(dea_sma2)
```

```
## Time elapsed: 0 seconds
## Model estimated: SMA(12)
## Initial values were produced using backcasting.
##
## Loss function type: MSE; Loss function value: 203219985.2919
## Error standard deviation: 14311.98
## Sample size: 254
```

```
## Number of estimated parameters: 2
## Number of degrees of freedom: 252
## Information criteria:
##      AIC     AICc      BIC     BICc
## 5583.790 5583.838 5590.864 5590.997
```

```
# MODEL1 HAVE THE lowest RSME 13252.08   vs 14255.52
checkresiduals(dea_sma1)
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```




```
# STILL SOME SPIKES outside the blue lines
```

```
###  Exponential Smoothing  (SES)
#
dea_ses1 <- ses(deaths, lambda=0.250792)
dea_ses2 <- ses(deaths, lambda=0.250792,alpha=0.1)
```

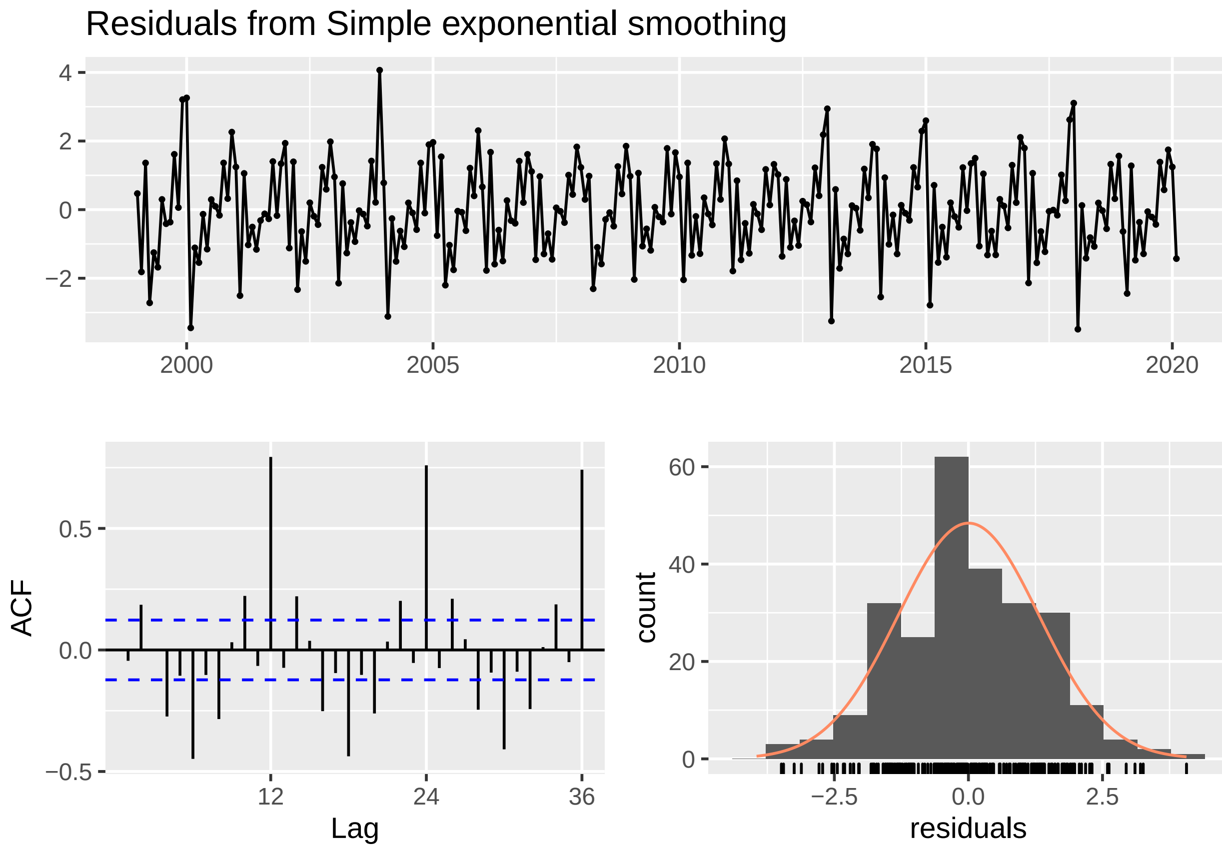
```
#
accuracy(dea_ses1$fitted,deaths)
```

```
##                 ME     RMSE      MAE        MPE     MAPE        ACF1 Theil's U
## Test set 112.4124 13246.03 10347.25 -0.1806992 4.794393 -0.05852106 0.9572697
```

```r
#accuracy(dea_ses2$fitted,deaths)

summary(dea_ses1)
```

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = deaths, lambda = 0.250792)
##
##   Box-Cox transformation: lambda= 0.2508
##
##   Smoothing parameters:
##     alpha = 0.7227
##
##   Initial states:
##     l = 83.4911
##
##   sigma:  1.3204
##
##      AIC     AICc      BIC
## 1551.678 1551.774 1562.290
##
## Error measures:
##                    ME     RMSE      MAE        MPE     MAPE     MASE
## Training set 112.4124 13246.03 10347.25 -0.1806992 4.794393 1.682574
##                     ACF1
## Training set -0.05852106
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Mar 2020       228275.1 211270.8 246284.7 202663.2 256238.6
## Apr 2020       228275.1 207435.8 250644.8 196999.2 263131.0
## May 2020       228275.1 204260.1 254345.6 192341.5 269017.3
## Jun 2020       228275.1 201500.4 257630.5 188318.1 274269.4
## Jul 2020       228275.1 199034.0 260622.1 184741.6 279074.9
## Aug 2020       228275.1 196788.6 263392.7 181501.8 283543.9
## Sep 2020       228275.1 194717.7 265988.9 178527.3 287747.7
## Oct 2020       228275.1 192788.8 268443.1 175768.9 291735.6
## Nov 2020       228275.1 190978.6 270778.7 173190.8 295543.5
## Dec 2020       228275.1 189269.4 273013.2 170766.1 299198.0
```

```r
#print('###################')
#summary(dea_ses2)
  # BEST MODEL IS dea_ses1 with a RMSE 13246.03  vs 14611.42
checkresiduals(dea_ses1)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Simple exponential smoothing
## Q* = 577.76, df = 22, p-value < 2.2e-16
##
## Model df: 2.   Total lags used: 24
```

```
# g
# SES 1 is the best model with a RMSE = 13978.23
#dea$ses_fit <- dea_ses1$fitted
#head(dea)
```

Section 2.2 - Simple Linear Regression

Time Series with TREND

```r
#print('#################### With TREND ######################')
# a
deaths_tslm1 <- tslm(data=death_ts,deaths ~ trend )
```

```r
#
summary(deaths_tslm1)
```

```
##
## Call:
## tslm(formula = deaths ~ trend, data = death_ts)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -25686  -9978  -3289   7822  61826
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 193943.49    1878.58  103.24   <2e-16 ***
## trend          135.26      12.77   10.59   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14930 on 252 degrees of freedom
## Multiple R-squared:  0.308,  Adjusted R-squared:  0.3052
## F-statistic: 112.1 on 1 and 252 DF,  p-value: < 2.2e-16
```

```r
accuracy(deaths_tslm1)
```

```
##                          ME     RMSE     MAE        MPE     MAPE     MASE
## Training set 2.522308e-12 14866.79 11635.3 -0.4655319 5.424406 1.892023
##                   ACF1
## Training set 0.5496212
```

```r
# RMSE 14866.79  Trend t-value is 11.72 is above 2 and so a good for the model
# Adjusted R-squared: 0.3052
# R-squared:  0.3456
```

Time Series with TREND & SEASON

```r
#print('#################### With TREND & SEASON #####################')
##
deaths_tslm2 <- tslm(data=death_ts,Deaths ~ trend + season )
```

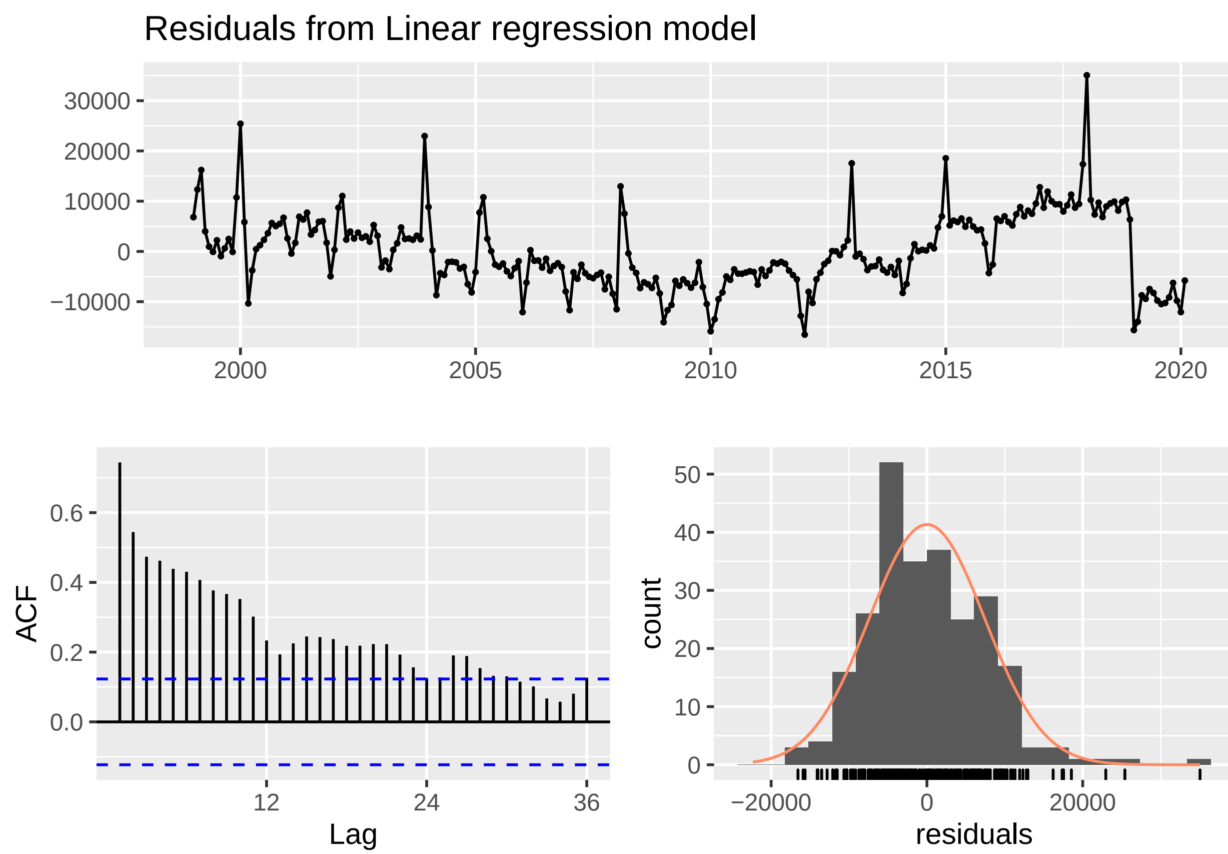
```r
#
summary(deaths_tslm2)
```

```
##
## Call:
## tslm(formula = Deaths ~ trend + season, data = death_ts)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -16557  -4833  -1401   5232  35064
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 220652.975   1826.045 120.837  < 2e-16 ***
## trend          135.491      6.533  20.739  < 2e-16 ***
## season2     -25072.536   2300.447 -10.899  < 2e-16 ***
## season3     -10497.640   2327.810  -4.510 1.01e-05 ***
## season4     -28635.941   2327.745 -12.302  < 2e-16 ***
## season5     -30300.527   2327.699 -13.017  < 2e-16 ***
## season6     -41217.589   2327.672 -17.708  < 2e-16 ***
## season7     -36952.461   2327.663 -15.875  < 2e-16 ***
## season8     -38301.142   2327.672 -16.455  < 2e-16 ***
## season9     -42191.681   2327.699 -18.126  < 2e-16 ***
## season10    -28847.315   2327.745 -12.393  < 2e-16 ***
## season11    -29960.139   2327.810 -12.871  < 2e-16 ***
## season12    -10245.677   2327.892  -4.401 1.62e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7630 on 241 degrees of freedom
## Multiple R-squared:  0.8271, Adjusted R-squared:  0.8185
## F-statistic: 96.05 on 12 and 241 DF,  p-value: < 2.2e-16
```

```r
accuracy(deaths_tslm2)
```

```
##                         ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -2.306466e-13 7431.877 5898.497 -0.1081362 2.713877 0.9591587
##                   ACF1
## Training set 0.7439228
```

```r
# Check the model
checkresiduals(deaths_tslm2)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 24
##
## data:  Residuals from Linear regression model
## LM test = 152.46, df = 24, p-value < 2.2e-16
```

```
# RMSE 9222.548   All of the seasons have t-values of above 2 which is good
# Adjusted R-squared:  0.758
# R-squared:  0.8271
```

```
#
# With TREND and Season has the lowest Adjusted R-squared: 0.758, RMSE = 9222.548
# And of the seasons have t-values of above 2
#
#dea$tslm2_fit <-deaths_tslm2$fitted.values

#head(dea) # to see if it works
```

Section 2.3 - Holt, HW and ETS

Holt's method

```
# a - HOLT
#print('################### HOLT ######################')
deaths_holt <- holt(deaths,lambda=0.250792)
```

```
# b
summary(deaths_holt)
```

```
##
## Forecast method: Holt's method
##
## Model Information:
## Holt's method
##
## Call:
##  holt(y = deaths, lambda = 0.250792)
##
##   Box-Cox transformation: lambda= 0.2508
##
##   Smoothing parameters:
##     alpha = 0.7231
##     beta  = 1e-04
##
##   Initial states:
##     l = 83.6484
##     b = 0.0018
##
##   sigma:  1.3258
##
##      AIC     AICc      BIC
## 1555.727 1555.969 1573.414
##
## Error measures:
##                    ME     RMSE      MAE        MPE    MAPE     MASE        ACF1
## Training set 82.57739 13247.62 10343.98 -0.1945082 4.79347 1.682042 -0.05828272
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Mar 2020       228297.2 211224.1 246383.8 202583.2 256382.0
## Apr 2020       228315.4 207386.6 250787.9 196907.7 263334.2
## May 2020       228333.6 204211.3 254530.0 192242.9 269276.4
## Jun 2020       228351.8 201453.6 257854.8 188214.9 274582.5
## Jul 2020       228370.1 198990.0 260885.7 184635.2 279440.8
## Aug 2020       228388.3 196748.2 263695.0 181393.2 283962.0
```

```
## Sep 2020       228406.5 194681.2 266329.7 178417.3 288217.8
## Oct 2020       228424.8 192756.6 268822.3 175658.0 292257.7
## Nov 2020       228443.0 190951.0 271196.2 173079.4 296117.5
## Dec 2020       228461.2 189246.5 273469.1 170654.4 299824.2
```

```r
accuracy(deaths_holt)
```

```
##                    ME     RMSE      MAE        MPE    MAPE     MASE        ACF1
## Training set 82.57739 13247.62 10343.98 -0.1945082 4.79347 1.682042 -0.05828272
```

```r
# RMSE 13247.62
```

Holt-Winters' additive method

```r
# c  HW
#print('################### HW ######################')
deaths_hw <- hw(deaths,seasonal="additive")
```

```r
# d
summary(deaths_hw)
```

```
##
## Forecast method: Holt-Winters' additive method
##
## Model Information:
## Holt-Winters' additive method
##
## Call:
##  hw(y = deaths, seasonal = "additive")
##
##   Smoothing parameters:
##     alpha = 0.6436
##     beta  = 1e-04
##     gamma = 1e-04
##
##   Initial states:
##     l = 201563.2597
##     b = 131.9233
##     s = 16961.27 -2964.197 -1753.395 -15121.55 -11388.55 -10112.58
##            -14226.83 -3417.434 -1941.378 16163.56 1404.808 26396.27
##
##   sigma:  5357.609
##
##      AIC     AICc      BIC
## 5785.783 5788.377 5845.918
##
## Error measures:
```

```
##                     ME     RMSE      MAE         MPE     MAPE      MASE
## Training set -85.06235 5186.121 3177.732 -0.07532331 1.457408 0.5167332
##                   ACF1
## Training set 0.1187685
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Mar 2020       237179.1 230313.1 244045.2 226678.4 247679.8
## Apr 2020       219204.6 211039.0 227370.2 206716.4 231692.8
## May 2020       217858.2 208572.9 227143.5 203657.6 232058.8
## Jun 2020       207178.7 196894.6 217462.8 191450.5 222906.9
## Jul 2020       211423.3 200228.9 222617.7 194302.9 228543.7
## Aug 2020       210276.8 198240.5 222313.2 191868.9 228684.8
## Sep 2020       206673.3 193850.0 219496.6 187061.8 226284.9
## Oct 2020       220171.4 206606.5 233736.4 199425.7 240917.2
## Nov 2020       219090.7 204822.4 233359.0 197269.3 240912.2
## Dec 2020       239145.4 224206.7 254084.2 216298.6 261992.3
## Jan 2021       248711.4 233130.8 264292.0 224882.9 272539.9
## Feb 2021       223849.0 207651.8 240046.2 199077.5 248620.5
## Mar 2021       238736.3 221944.7 255527.8 213055.8 264416.8
## Apr 2021       220761.7 203396.2 238127.3 194203.4 247320.1
## May 2021       219415.4 201494.0 237336.7 192007.0 246823.8
## Jun 2021       208735.9 190275.2 227196.5 180502.7 236969.0
## Jul 2021       212980.5 193995.7 231965.2 183945.8 242015.1
## Aug 2021       211834.0 192339.1 231328.9 182019.1 241648.9
## Sep 2021       208230.5 188238.3 228222.7 177655.0 238805.9
## Oct 2021       221728.6 201251.0 242206.2 190410.8 253046.4
## Nov 2021       220647.9 199696.0 241599.8 188604.7 252691.0
## Dec 2021       240702.6 219286.8 262118.4 207949.9 273455.2
## Jan 2022       250268.6 228398.5 272138.6 216821.2 283715.9
## Feb 2022       225406.1 203091.0 247721.3 191278.1 259534.2
```

```r
accuracy(deaths_hw)
```

```
##                     ME     RMSE      MAE         MPE     MAPE      MASE
## Training set -85.06235 5186.121 3177.732 -0.07532331 1.457408 0.5167332
##                   ACF1
## Training set 0.1187685
```

```r
# RMSE  5186.121
```

Holt-Winters' multiplicative method

```r
# multiplicative
deaths_hw <- hw(deaths,seasonal="multiplicative")
```

```r
# d
summary(deaths_hw)
```

```
##
## Forecast method: Holt-Winters' multiplicative method
##
## Model Information:
## Holt-Winters' multiplicative method
##
## Call:
##  hw(y = deaths, seasonal = "multiplicative")
##
##   Smoothing parameters:
##     alpha = 0.2535
##     beta  = 0.0148
##     gamma = 1e-04
##
##   Initial states:
##     l = 201673.2024
##     b = 146.0079
##     s = 1.0765 0.9829 0.997 0.927 0.9539 0.956
##            0.931 0.9875 0.9877 1.069 1.0049 1.1267
##
##   sigma:  0.0257
##
##      AIC     AICc      BIC
## 5791.249 5793.842 5851.383
##
## Error measures:
##                     ME     RMSE      MAE        MPE    MAPE      MASE      ACF1
## Training set -146.2943 5535.074 3626.803 -0.1202464 1.67671 0.5897569 0.4091426
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Mar 2020       232041.4 224396.2 239686.6 220349.1 243733.7
## Apr 2020       213908.0 206609.9 221206.2 202746.4 225069.7
## May 2020       213392.3 205842.0 220942.6 201845.1 224939.5
## Jun 2020       200735.0 193360.5 208109.4 189456.7 212013.2
## Jul 2020       205650.1 197798.1 213502.1 193641.6 217658.7
## Aug 2020       204732.8 196602.5 212863.1 192298.5 217167.0
## Sep 2020       198506.3 190302.7 206709.9 185960.0 211052.6
## Oct 2020       213008.3 203844.1 222172.5 198992.9 227023.8
## Nov 2020       209539.3 200152.3 218926.3 195183.2 223895.5
## Dec 2020       228958.7 218277.6 239639.7 212623.4 245293.9
## Jan 2021       239083.2 227469.2 250697.1 221321.2 256845.2
## Feb 2021       212746.8 201987.0 223506.6 196291.1 229202.5
## Mar 2021       225806.6 213918.9 237694.4 207625.9 243987.4
## Apr 2021       208147.6 196744.8 219550.5 190708.5 225586.7
## May 2021       207632.8 195800.8 219464.9 189537.3 225728.4
## Jun 2021       195304.9 183732.6 206877.3 177606.5 213003.3
## Jul 2021       200074.5 187753.4 212395.7 181230.9 218918.2
## Aug 2021       199169.5 186427.7 211911.3 179682.6 218656.4
## Sep 2021       193100.0 180273.0 205926.9 173482.9 212717.0
```

```
## Oct 2021       207193.8 192910.6 221477.1 185349.5 229038.2
## Nov 2021       203806.5 189233.4 218379.6 181518.9 226094.1
## Dec 2021       222680.2 206173.1 239187.3 197434.8 247925.6
## Jan 2022       232512.1 214653.1 250371.1 205199.0 259825.1
## Feb 2022       206886.1 190429.6 223342.6 181718.1 232054.1
```

```r
accuracy(deaths_hw)
```

```
##                     ME     RMSE      MAE        MPE    MAPE      MASE      ACF1
## Training set -146.2943 5535.074 3626.803 -0.1202464 1.67671 0.5897569 0.4091426
```

```r
# RMSE  5535.074
```

Exponential smoothing (ETS)

```r
# e ets
#print('################### ETS ######################')
deaths_ets <- ets(deaths)
```

```r
# f
summary(deaths_ets)
```

```
## ETS(M,Ad,M)
##
## Call:
##  ets(y = deaths)
##
##   Smoothing parameters:
##     alpha = 0.7944
##     beta  = 0.0017
##     gamma = 1e-04
##     phi   = 0.9783
##
##   Initial states:
##     l = 200112.9901
##     b = 108.4069
##     s = 1.0816 0.9894 0.9944 0.9297 0.9446 0.9497
##            0.9309 0.982 0.9886 1.0747 1.009 1.1255
##
##   sigma:  0.0233
##
##      AIC     AICc      BIC
## 5741.032 5743.942 5804.704
##
## Training set error measures:
##                    ME    RMSE      MAE         MPE MAPE      MASE       ACF1
## Training set 87.78155 5037.35 3074.928 0.003415265 1.41 0.5000162 0.04556511
```

```r
accuracy(deaths_ets)
```

```
##                    ME    RMSE      MAE         MPE MAPE      MASE       ACF1
## Training set 87.78155 5037.35 3074.928 0.003415265 1.41 0.5000162 0.04556511
```

```r
# RMSE 5037.35
```

```r
# g
# ETS had the lowest RSM value of 5037.35
#dea$ets_fit <- dea_ets$fitted
#head(dea) # to see if it worked
```

Section 2.4 - ARIMA MODELS

```r
BoxCox.lambda(deaths)
```

```
## [1] 0.250792
```

```r
# For Non Seasonal
ndiffs(BoxCox(deaths,lambda=0.250792))
```

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

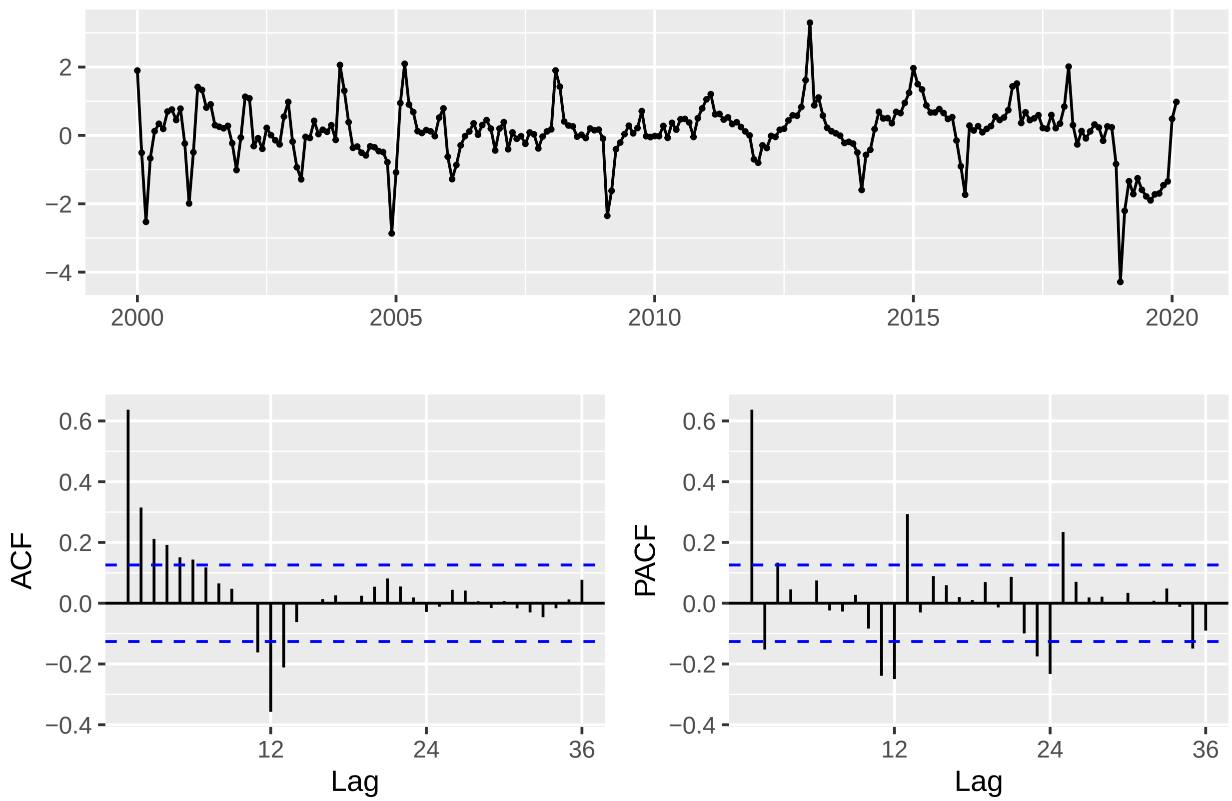
```r
# Gives me d for the Arima Model
# = 1 so the model needs a non-Seasonal differencing of 1
```

```r
# For Seasonal
# Gives me d for the Arima Model
nsdiffs(diff(BoxCox(deaths,lambda=0.250792),lag=12))
```

```
## [1] 0
```

```r
# = 0 so do not need to Seasonal differencing  of 1 after one Diff indicated in ndiff
```

```r
#
#   The PACF plots show that there are significant lags at 1 then minor ones at 12,13,24,25.
#   The AFC plot there is a geometric decay at lag 12. This would indicate that a seasonal AR model sho

ggtsdisplay(diff(BoxCox(deaths,lambda=0.250792),lag = 12))
```



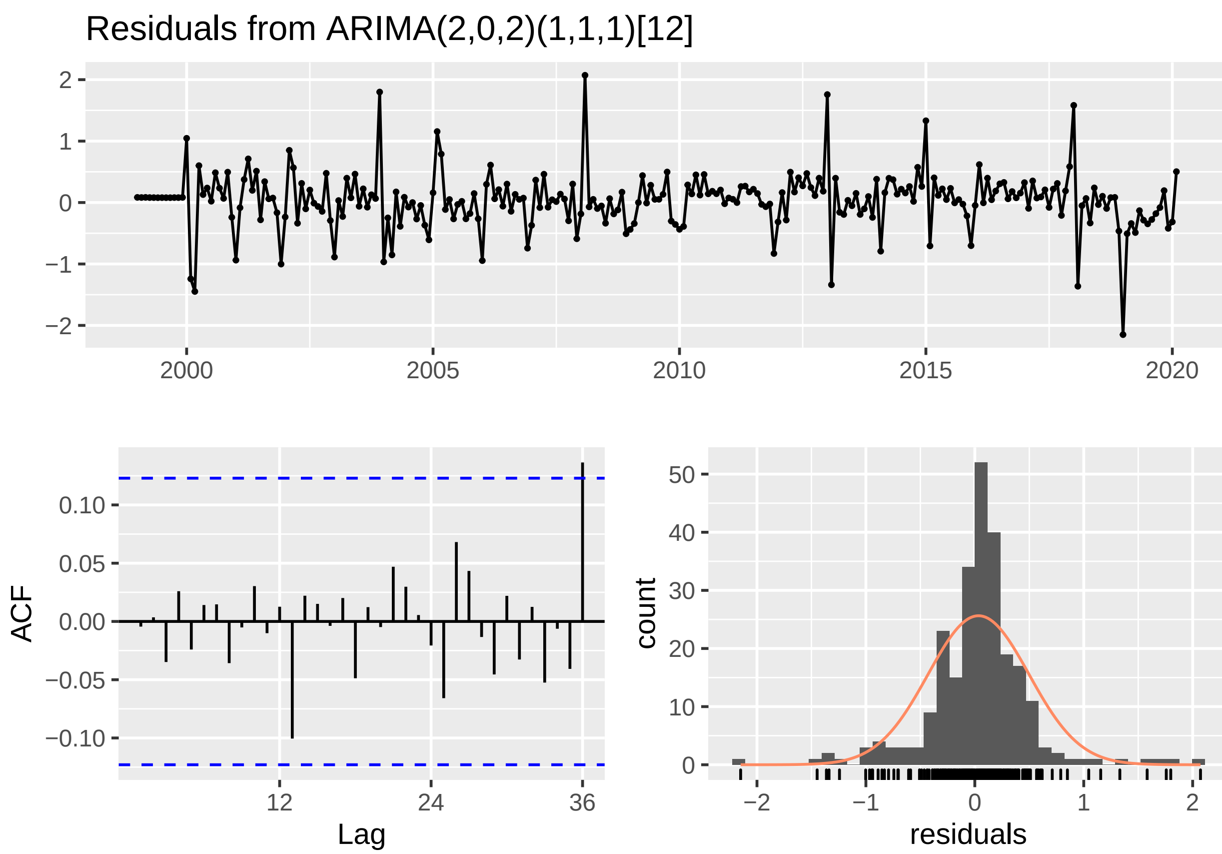
ARIMA Manual

```
# Manual ARIMA

#dea_arima1 <- Arima(deaths, order = c(2,0,0),seasonal=c(2,1,1),lambda=0.250792)
# AICc = 386.65

#dea_arima1 <- Arima(deaths, order = c(1,0,1),seasonal=c(2,1,1),lambda=0.250792)
# AICc = 385.7

#dea_arima1 <- Arima(deaths, order = c(2,0,2),seasonal=c(2,1,0),lambda=0.250792)
# AICc = 382.31

#dea_arima1 <- Arima(deaths, order = c(2,0,1),seasonal=c(1,1,1),lambda=0.250792)
# AICc = 373.47

#dea_arima1 <- Arima(deaths, order = c(2,0,2),seasonal=c(2,1,1),lambda=0.250792)
# AICc = 369.15

#dea_arima1 <- Arima(deaths, order = c(1,0,2),seasonal=c(2,1,2),lambda=0.250792)
# AICc = 369.14
```

```r
dea_arima1 <- Arima(deaths, order = c(2,0,2),seasonal=c(1,1,1),lambda=0.250792)
# Best Model LOWEST AICc
# AICC = 368.60
# RMSE = 4787.379

summary(dea_arima1)
```

```
## Series: deaths
## ARIMA(2,0,2)(1,1,1)[12]
## Box Cox transformation: lambda= 0.250792
##
## Coefficients:
##          ar1      ar2      ma1      ma2     sar1     sma1
##       1.2821  -0.2844  -0.5224  -0.2648  -0.0347  -0.9612
## s.e.  0.1434   0.1412   0.1359   0.0861   0.0801   0.1561
##
## sigma^2 estimated as 0.2317:  log likelihood=-177.06
## AIC=368.12   AICc=368.6   BIC=392.54
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 324.3337 4787.379 3062.717 0.1377605 1.402881 0.4980306
##                     ACF1
## Training set -0.01653005
```

```r
checkresiduals(dea_arima1)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,2)(1,1,1)[12]
## Q* = 6.1479, df = 18, p-value = 0.9956
##
## Model df: 6.   Total lags used: 24
```

AUTO ARIMA

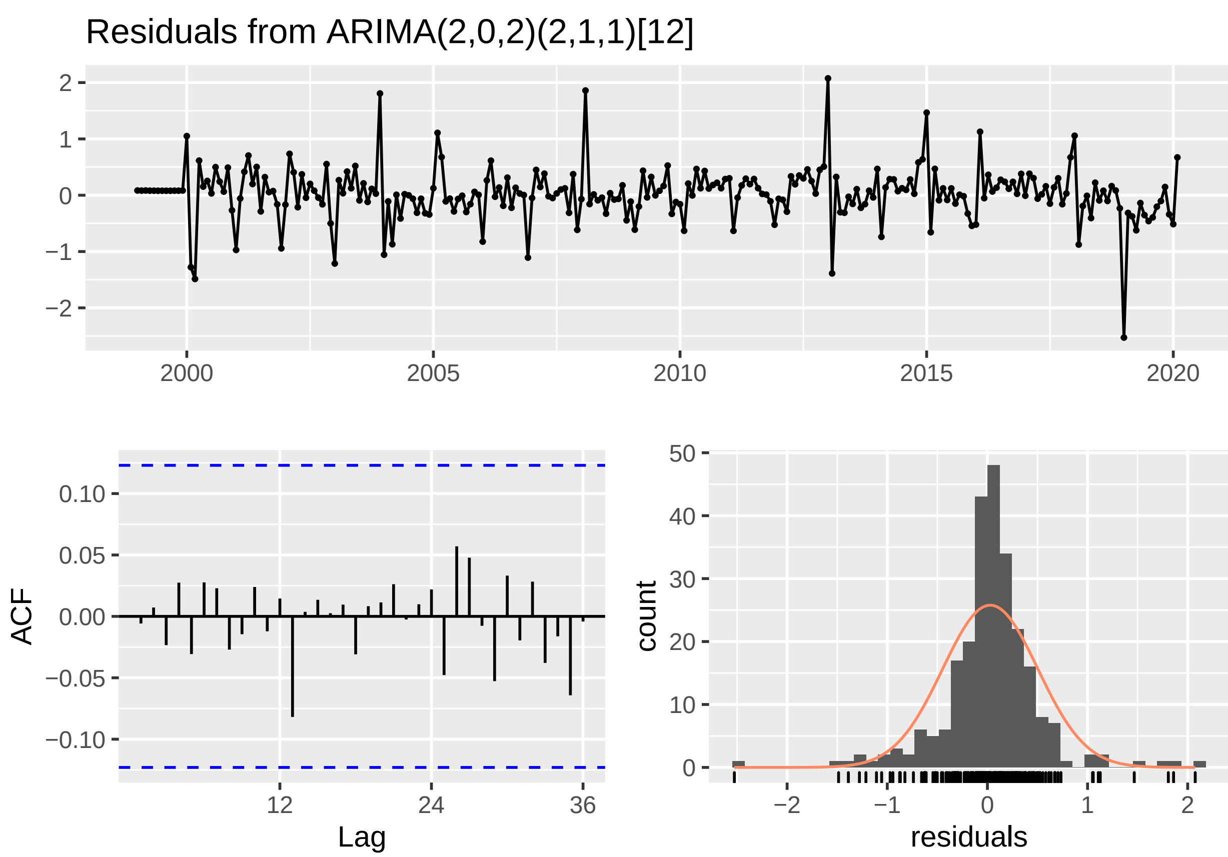
```r
# AUTO ARIMA
aarima <- auto.arima(deaths , lambda=0.250792)
summary(aarima)
```

```
## Series: deaths
## ARIMA(2,0,2)(2,1,1)[12]
## Box Cox transformation: lambda= 0.250792
##
## Coefficients:
##          ar1      ar2      ma1      ma2     sar1     sar2     sma1
##       1.2483  -0.2637  -0.4965  -0.2534  -0.3545  -0.3005  -0.5552
```

```
## s.e.  0.1595   0.1534   0.1530   0.0869   0.1433   0.1151   0.1553
##
## sigma^2 estimated as 0.2449:  log likelihood=-176.27
## AIC=368.53   AICc=369.15   BIC=396.44
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 269.4836 4914.212 3090.992 0.1097452 1.415898 0.5026283
##                     ACF1
## Training set -0.01262414
```

```
# AICc = 369.15
# RMSE = 4914.212
# Not better that the manual
```

```
# f The auto Arima Model had a RMSE value of 1.587109
#   The Arima Model of order = c(0,1,0),seasonal=c(1,1,0) has a RMSE = 49.51
#   They match

# Check the
checkresiduals(aarima)
```



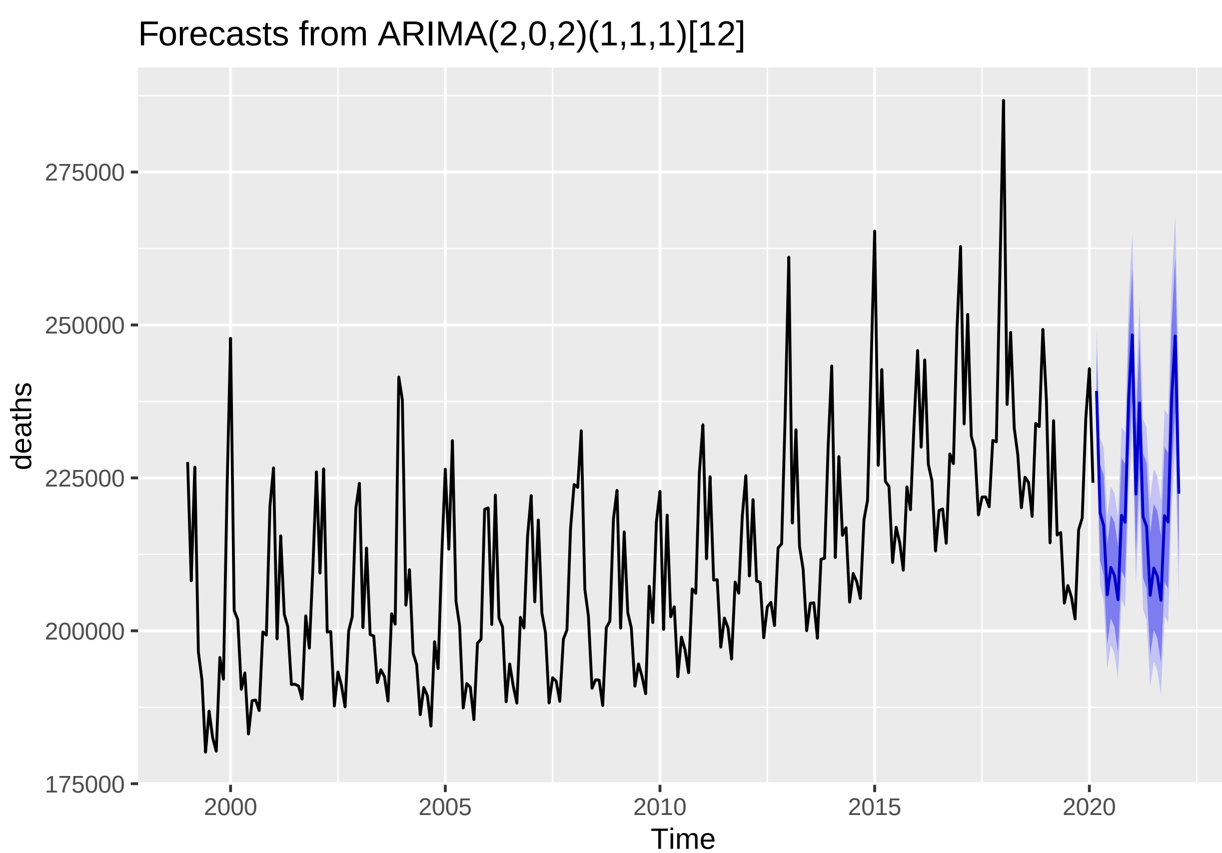

```
##
##  Ljung-Box test
##
```

```
## data:  Residuals from ARIMA(2,0,2)(2,1,1)[12]
## Q* = 4.0076, df = 17, p-value = 0.9995
##
## Model df: 7.   Total lags used: 24
```

Section 3 - Forecast Best Model

ARIMA

```r
# g
# Manual Arima model has the lowest RSM value of 44.3173
#deaths$arima1_fit <- dea_arima1$fitted

#head(deaths) # to see if it worked
#head(dea_arima1$fitted)
autoplot(forecast(dea_arima1))
```



```r
forecast(dea_arima1)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Mar 2020       239217.3 232624.5 245949.1 229190.1 249569.7
## Apr 2020       219264.2 211536.0 227202.1 207528.5 231490.2
## May 2020       217177.8 209088.4 225499.4 204898.6 230000.1
## Jun 2020       205863.5 197865.7 214101.0 193727.4 218560.4
## Jul 2020       210367.7 202051.2 218938.1 197749.7 223579.5
## Aug 2020       209067.8 200618.7 217780.7 196250.9 222501.7
## Sep 2020       205076.6 196588.0 213836.8 192202.4 218586.2
## Oct 2020       218893.1 209813.4 228263.9 205122.8 233344.5
## Nov 2020       217751.1 208548.8 227254.4 203797.0 232409.1
## Dec 2020       238127.4 228119.4 248460.8 222950.8 254065.0
## Jan 2021       248385.0 237889.6 259223.6 232470.0 265102.6
## Feb 2021       222333.6 212530.0 232472.2 207473.3 237977.4
## Mar 2021       237238.3 226767.3 248067.4 221366.5 253947.7
## Apr 2021       218660.5 208657.5 229018.4 203503.2 234648.1
## May 2021       217010.9 206921.0 227465.0 201724.3 233149.5
## Jun 2021       205788.3 195961.2 215980.0 190903.7 221525.8
## Jul 2021       210262.7 200143.5 220760.5 194936.8 226474.1
## Aug 2021       208948.7 198749.5 219535.1 193503.8 225299.2
## Sep 2021       204975.5 194799.8 215544.4 189568.9 221301.7
## Oct 2021       218820.3 208004.1 230052.4 202443.3 236170.3
## Nov 2021       217784.9 206884.1 229110.4 201281.8 235281.4
## Dec 2021       238019.8 226235.2 250258.3 220176.7 256924.8
## Jan 2022       248202.5 235910.3 260968.3 229591.0 267921.9
## Feb 2022       222408.3 210974.6 234300.0 205103.3 240784.7
```

```r
forecast(dea_arima1$fitted)
```

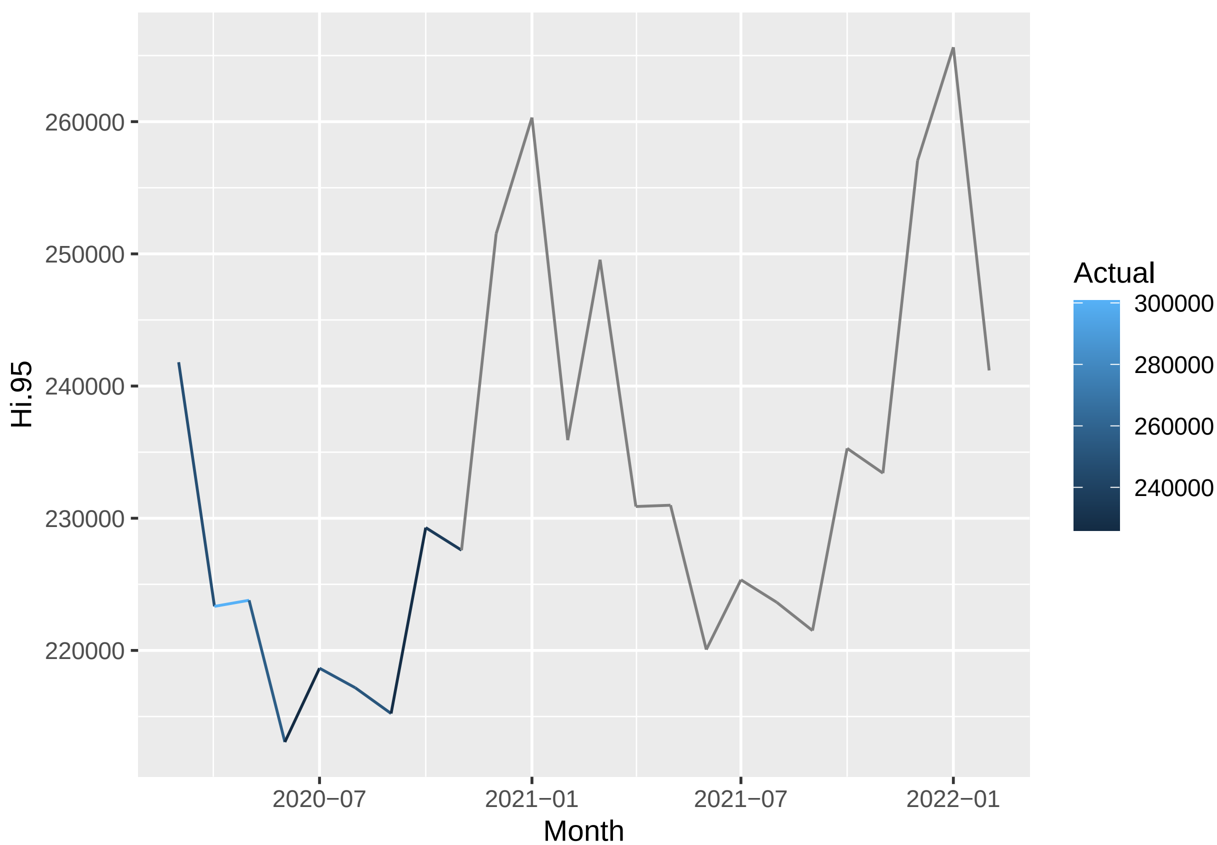
```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Mar 2020       233472.1 228025.5 238918.7 225142.2 241802.0
## Apr 2020       214578.2 208854.2 220302.3 205824.1 223332.4
## May 2020       214220.7 207960.4 220480.9 204646.4 223794.9
## Jun 2020       203025.5 196457.6 209593.4 192980.7 213070.2
## Jul 2020       207835.8 200761.7 214909.8 197017.0 218654.5
## Aug 2020       205762.3 198300.1 213224.4 194349.9 217174.7
## Sep 2020       203270.0 195452.2 211087.8 191313.8 215226.2
## Oct 2020       216494.2 208128.6 224859.7 203700.1 229288.2
## Nov 2020       214243.5 205520.8 222966.2 200903.3 227583.7
## Dec 2020       237157.3 227765.4 246549.2 222793.6 251520.9
## Jan 2021       245154.2 235241.5 255067.0 229994.0 260314.5
## Feb 2021       220573.4 210546.6 230600.3 205238.7 235908.2
## Mar 2021       233472.1 222955.0 243989.2 217387.6 249556.6
## Apr 2021       214578.2 203914.5 225241.9 198269.5 230887.0
## May 2021       214220.7 203259.5 225181.9 197457.0 230984.4
## Jun 2021       203025.5 191885.4 214165.5 185988.2 220062.7
## Jul 2021       207835.8 196389.7 219281.8 190330.5 225341.0
## Aug 2021       205762.3 194072.0 217452.5 187883.6 223640.9
## Sep 2021       203270.0 191349.3 215190.7 185038.9 221501.1
## Oct 2021       216494.2 204207.0 228781.3 197702.6 235285.8
## Nov 2021       214243.5 201710.2 226776.8 195075.4 233411.6
## Dec 2021       237157.3 224149.0 250165.5 217262.9 257051.6
```

```
## Jan 2022       245154.2 231764.9 258543.6 224677.0 265631.5
## Feb 2022       220573.4 207099.1 234047.8 199966.2 241180.6
```

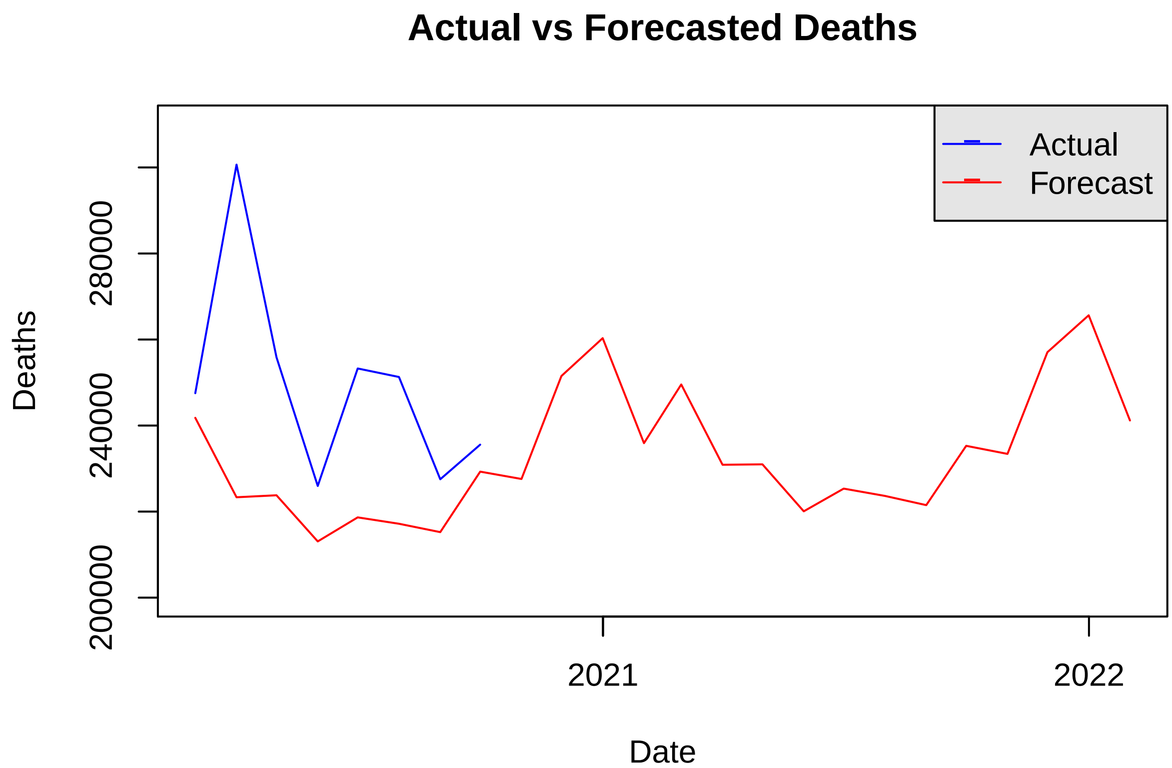
```r
#deathForAct <- readxl::read_excel("Data Sets/ALL Deaths 1999 2020.xlsx")
deathForAct <- readxl::read_excel("Data Sets/FinalProjet Forcast Vs Actual-Mark Drummond.xlsx")
```

```r
# DEATHS with ACTUALS
death_tsAA <- ts(deathsx[,-1], frequency=12)
deathsAAA <- death_tsAA[, 'Deaths']
#View(deathsAAA)
```

```r
ggplot(data = deathForAct, aes(x=Month, y=Hi.95)) + geom_line(aes(colour=Actual))
```




```r
#Plot Actuals
plot(deathForAct$Month, deathForAct$Actual ,col="blue", type="l"  ,xlab="Date", ylab="Deaths", main="Ac
points(deathForAct$Month, deathForAct$Hi.95, col="red", type="l")
legend("topright", c("Actual", "Forecast"), col = c("blue", "red"),
       text.col = "black", lty = c(1, 1), pch = c("-","-"),
       merge = TRUE, bg = "gray90")
```




```r
# Write out forcast into 'examreview-your-name.csv' file
write.csv(forecast(dea_arima1$fitted),"Data Sets/FinalProjetForecast-Mark Drummond.csv")
```