

scatch

Yutian Yang

11/23/2021

```
library(tidyverse)
```

```
## -- Attaching packages --
----- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.1      v purrr   0.3.4
## v tibble  3.0.0      v dplyr    1.0.0
## v tidyr   1.1.0      v stringr  1.4.0
## v readr   1.3.1      vforcats  0.5.0
```

```
## -- Conflicts --
----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()
```

```
library(forecast)
```

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

```
covid <- readRDS("data/covid_north_complete.RData")
```

```
fit = lm(new_cases_per_million~new_tests_per_million+as.factor(continent),data = covid)
summary(fit)
```

```

## 
## Call:
## lm(formula = new_cases_per_million ~ new_tests_per_million +
##     as.factor(continent), data = covid)
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -596.12  -54.64   -26.21   18.81 1364.18 
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                4.166e+01  4.025e+00 10.351 < 2e-16 ***
## new_tests_per_million       2.059e-02  7.716e-04 26.680 < 2e-16 ***
## as.factor(continent)Asia   -2.657e+01  5.500e+00 -4.831 1.4e-06 *** 
## as.factor(continent)Europe  6.896e+01  6.715e+00 10.270 < 2e-16 *** 
## as.factor(continent)North America 5.426e+01  5.379e+00 10.088 < 2e-16 *** 
## ---                        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
##
## Residual standard error: 136.3 on 5530 degrees of freedom
## Multiple R-squared:  0.3049, Adjusted R-squared:  0.3044 
## F-statistic: 606.4 on 4 and 5530 DF,  p-value: < 2.2e-16

```

#Taking out new tests rate effect on new cases rate

```

residuals = residuals(fit)

new_covid <-
  covid %>% mutate("residuals" = residuals)

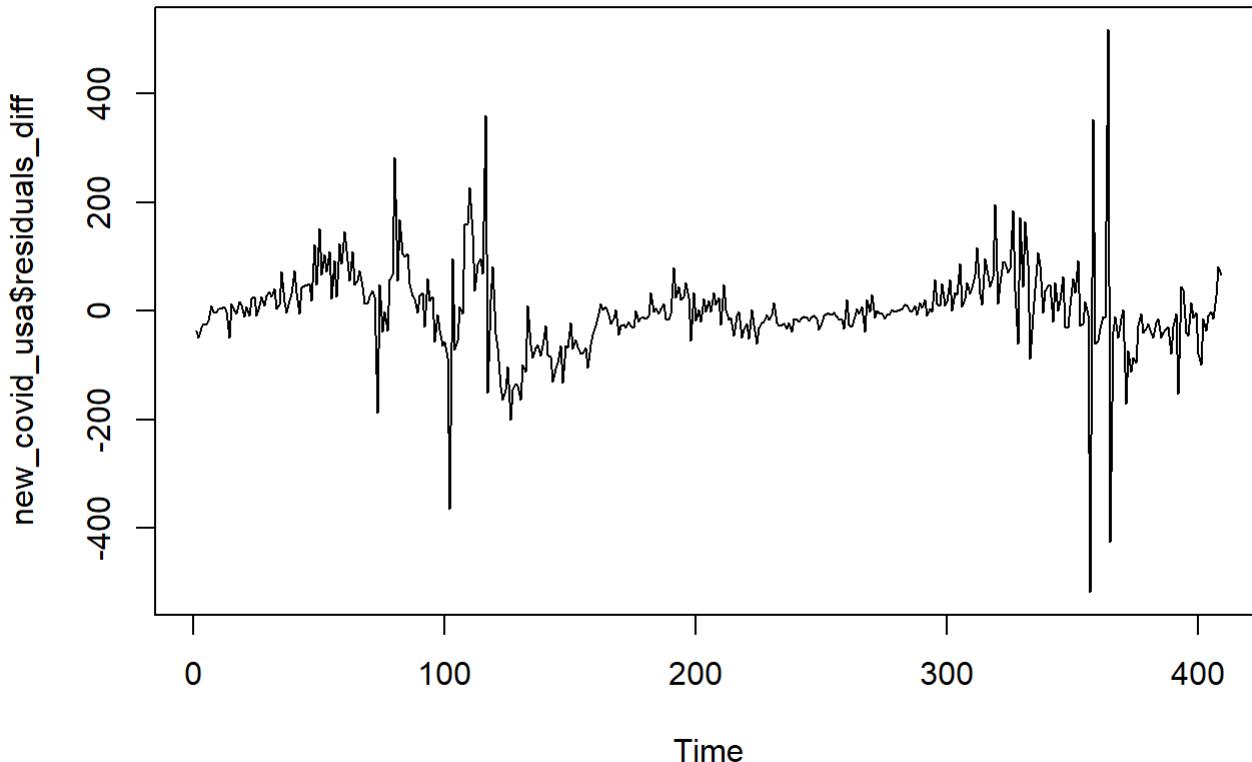
covid_usa <- new_covid %>% filter(location == "United States")

covid_canada <- new_covid %>% filter(location == "Canada")
covid_cote <- new_covid %>% filter(location == "Cote d'Ivoire")
covid_ghana <- new_covid %>% filter(location == "Ghana")
covid_guatemala <- new_covid %>% filter(location == "Guatemala")
covid_india <- new_covid %>% filter(location == "India")
covid_indonesia <- new_covid %>% filter(location == "Indonesia")
covid_italy <- new_covid %>% filter(location == "Italy")
covid_japan <- new_covid %>% filter(location == "Japan")
covid_mexico <- new_covid %>% filter(location == "Mexico")
covid_morocco <- new_covid %>% filter(location == "Morocco")
covid_portugal <- new_covid %>% filter(location == "Portugal")
covid_russia <- new_covid %>% filter(location == "Russia")
covid_saudi <- new_covid %>% filter(location == "Saudi Arabia")
covid_south_africa <- new_covid %>% filter(location == "South Africa")
covid_uk <- new_covid %>% filter(location == "United Kingdom")

```

```
new_covid_usa <-
  covid_usa %>% mutate("residuals_diff" = residuals)

#deseasonalized
new_covid_usa$residuals_diff[8:409] = new_covid_usa$residuals_diff %>% diff(7)
plot.ts(new_covid_usa$residuals_diff)
```



```
# ARIMA as a special case of the ARIMAX intercept
regression_model = lm(residuals_diff~1, data = new_covid_usa[8:379,])
summary(regression_model)
```

```
##
## Call:
## lm(formula = residuals_diff ~ 1, data = new_covid_usa[8:379,
##   ])
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -520.99 -29.14  -3.83  29.87 513.88
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.338     4.300   0.776   0.438
##
## Residual standard error: 82.93 on 371 degrees of freedom
```

```
anova(regression_model)
```

```
## Analysis of Variance Table
##
## Response: residuals_diff
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 371 2551419  6877.1
```

```
error = residuals(regression_model)
ts_error = ts(error)

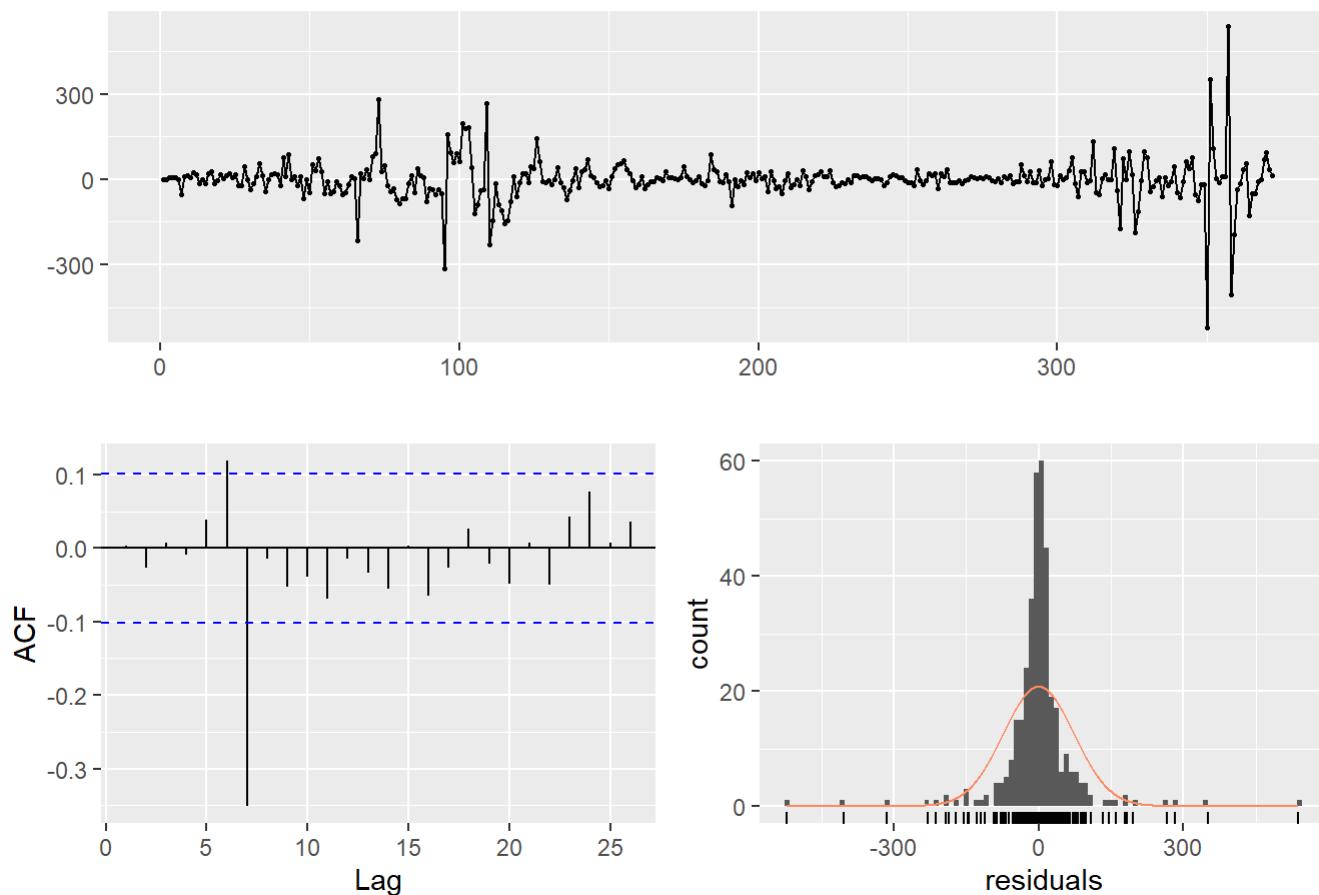
#ARIMA on error

error_model = auto.arima(ts_error,d=1)
fitted_error = fitted(error_model)
summary(error_model)
```

```
## Series: ts_error
## ARIMA(0,1,2)
##
## Coefficients:
##             ma1      ma2
##            -1.0069  0.3219
## s.e.    0.0504  0.0614
##
## sigma^2 estimated as 5451: log likelihood=-2121.92
## AIC=4249.85  AICc=4249.91  BIC=4261.59
##
## Training set error measures:
##               ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -0.2440456 73.53196 38.59503 -255.7629 428.6835 0.7869466
##                         ACF1
## Training set 0.003187813
```

```
checkresiduals(error_model1)
```

Residuals from ARIMA(0,1,2)



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(0,1,2)  
## Q* = 54.878, df = 8, p-value = 4.663e-09  
##  
## Model df: 2. Total lags used: 10
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

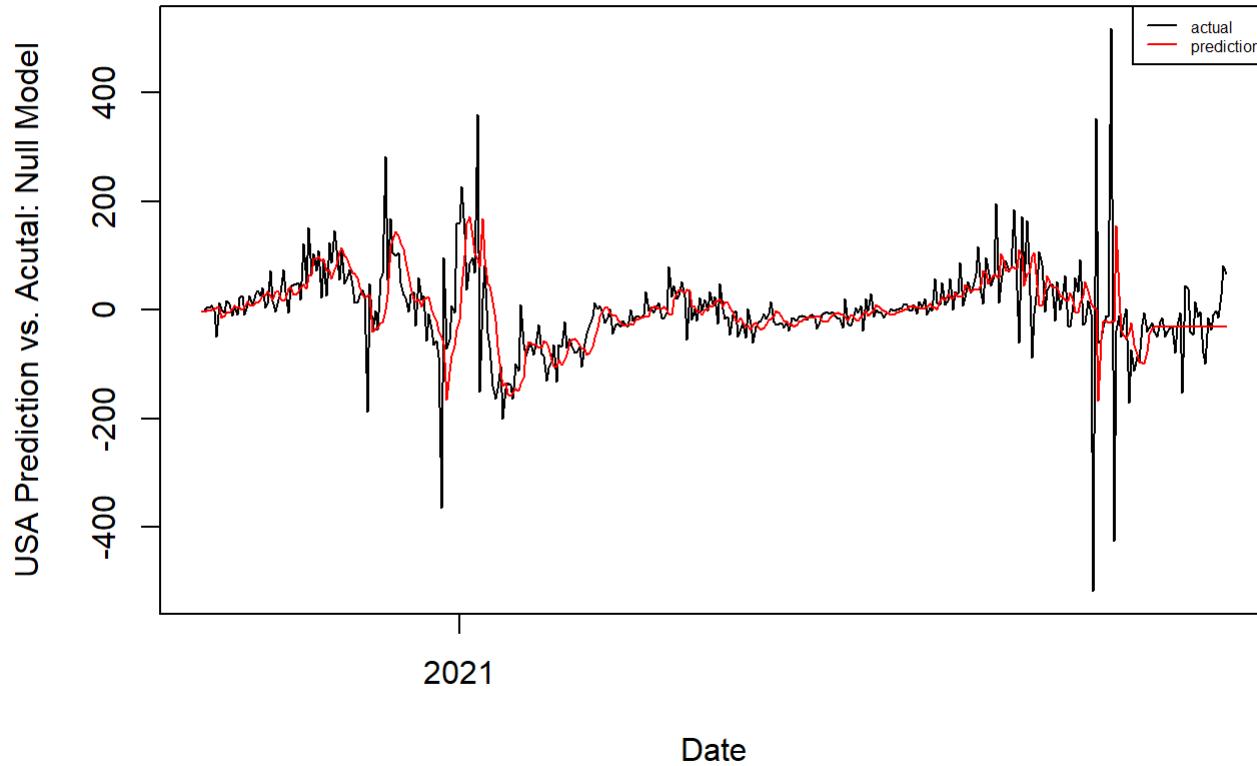
regression_prediction = predict(regression_model, newdata = new_covid_usa[380:409,])
error_forecast = predict(error_model, n.ahead = 30)$pred

final_prediction_2 = regression_prediction + error_forecast

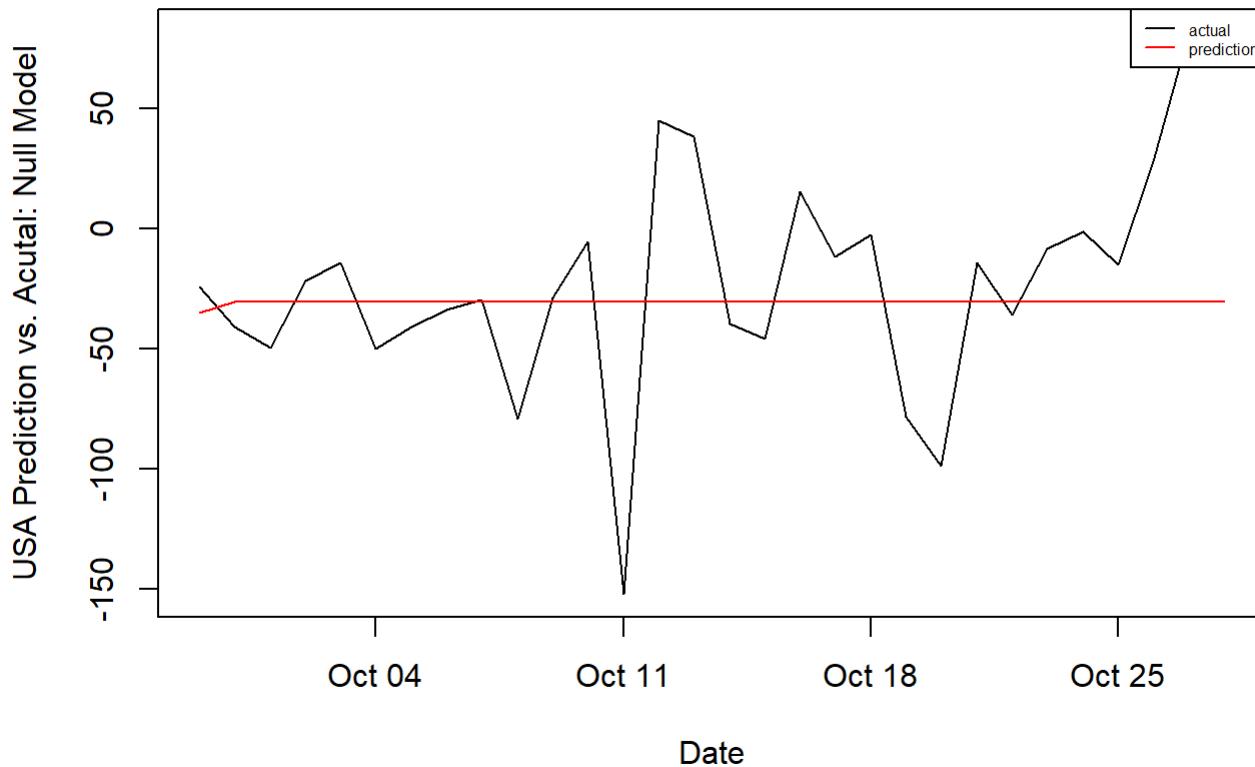
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_covid_usa$residuals_diff[8:409])
colnames(compare_table) = c("predict","actual")

plot(new_covid_usa$Day[8:409],                                     # Draw first time series
      new_covid_usa$residuals_diff[8:409],
      type = "l",
      xlab = "Date",
      ylab = "USA Prediction vs. Actual: Null Model")
lines(new_covid_usa$Day[8:409],                                     # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_covid_usa$Day[380:409],  
      new_covid_usa$residuals_diff[380:409],  
      type = "l",  
      xlab = "Date",  
      ylab = "USA Prediction vs. Acutal: Null Model")  
lines(new_covid_usa$Day[380:409],  
      final_prediction_2,  
      type = "l",  
      col = "red")  
legend(x = "topright", legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE USA Null
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 5406.949
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 2250.964
```

```
regression_model = lm(residuals_diff~as.factor(vaccination_policy), data = new_covid_usa[8:379,])
summary(regression_model)
```

```

## 
## Call:
## lm(formula = residuals_diff ~ as.factor(vaccination_policy),
##     data = new_covid_usa[8:379, ])
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -525.96  -28.58   -9.53   24.70  508.91 
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                 39.751    8.691   4.574 6.57e-06 *** 
## as.factor(vaccination_policy)1 -78.348   12.615  -6.211 1.43e-09 *** 
## as.factor(vaccination_policy)2 -41.325   16.667  -2.479  0.01361 *  
## as.factor(vaccination_policy)3 -38.878   19.724  -1.971  0.04946 *  
## as.factor(vaccination_policy)4 -61.779   21.619  -2.858  0.00451 ** 
## as.factor(vaccination_policy)5 -31.445   10.871  -2.892  0.00405 ** 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 79.18 on 366 degrees of freedom 
## Multiple R-squared:  0.1006, Adjusted R-squared:  0.08834 
## F-statistic:  8.19 on 5 and 366 DF,  p-value: 2.405e-07

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: residuals_diff
##                               Df  Sum Sq Mean Sq F value    Pr(>F)    
## as.factor(vaccination_policy) 5  256730  51346  8.1896 2.405e-07 ***
## Residuals                      366 2294689   6270 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

error = residuals(regression_model)
ts_error = ts(error)

#ARIMA on error

error_model = auto.arima(ts_error,d=1)
fitted_error = fitted(error_model)
summary(error_model)

```

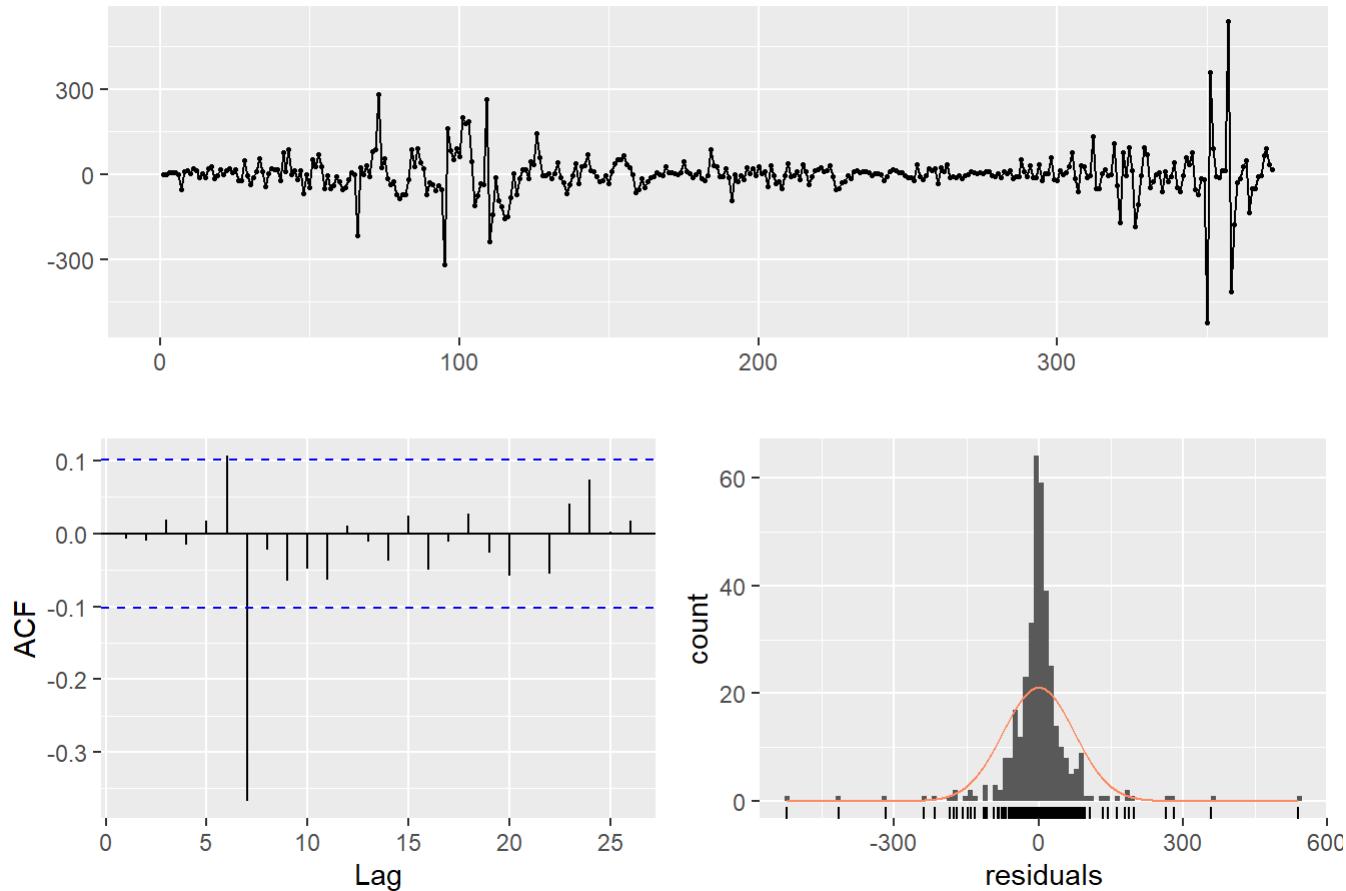
```

## Series: ts_error
## ARIMA(0,1,2)
##
## Coefficients:
##      ma1     ma2
##     -0.9865  0.2865
## s.e.   0.0489  0.0637
##
## sigma^2 estimated as 5532: log likelihood=-2124.65
## AIC=4255.3    AICc=4255.37   BIC=4267.05
##
## Training set error measures:
##           ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -0.003997561 74.07735 39.18947 47.12982 215.4583 0.7916252
##          ACF1
## Training set -0.006875177

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(0,1,2)



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(0,1,2)  
## Q* = 58.971, df = 8, p-value = 7.416e-10  
##  
## Model df: 2. Total lags used: 10
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

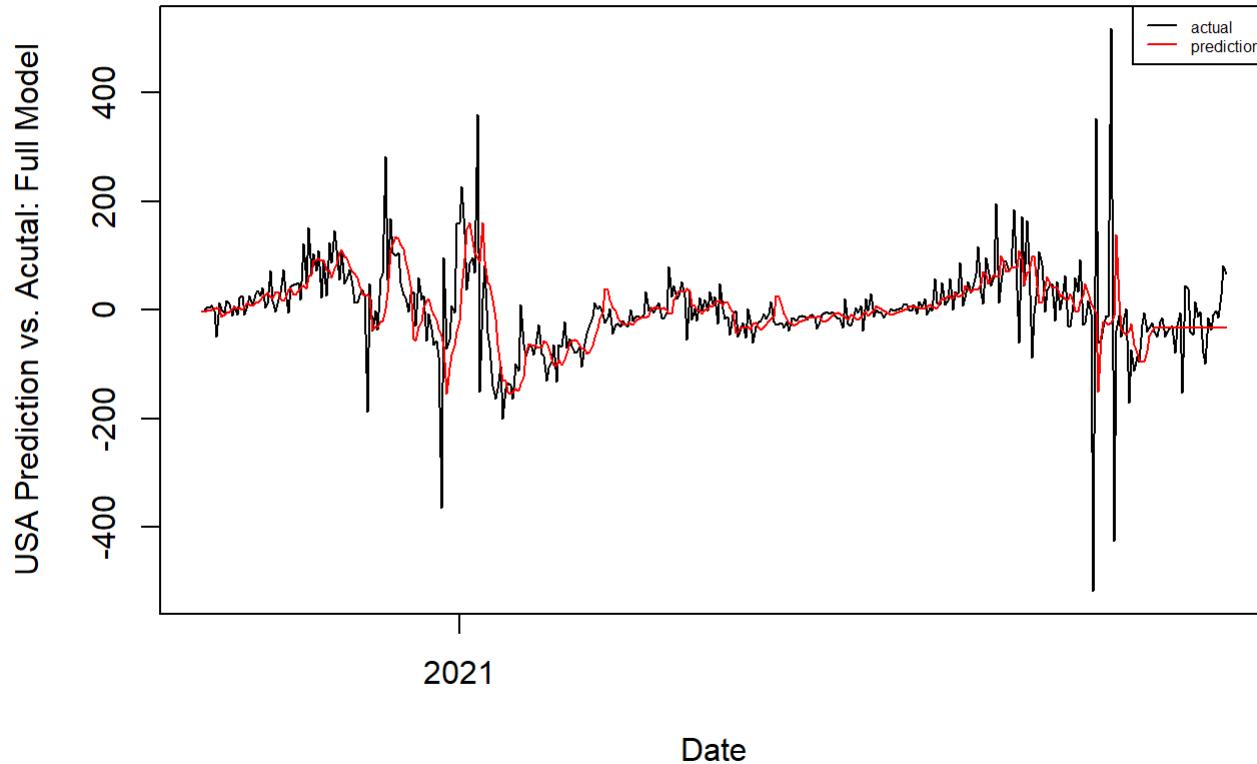
regression_prediction = predict(regression_model, newdata = new_covid_usa[380:409,])
error_forecast = predict(error_model, n.ahead = 30)$pred

final_prediction_2 = regression_prediction + error_forecast

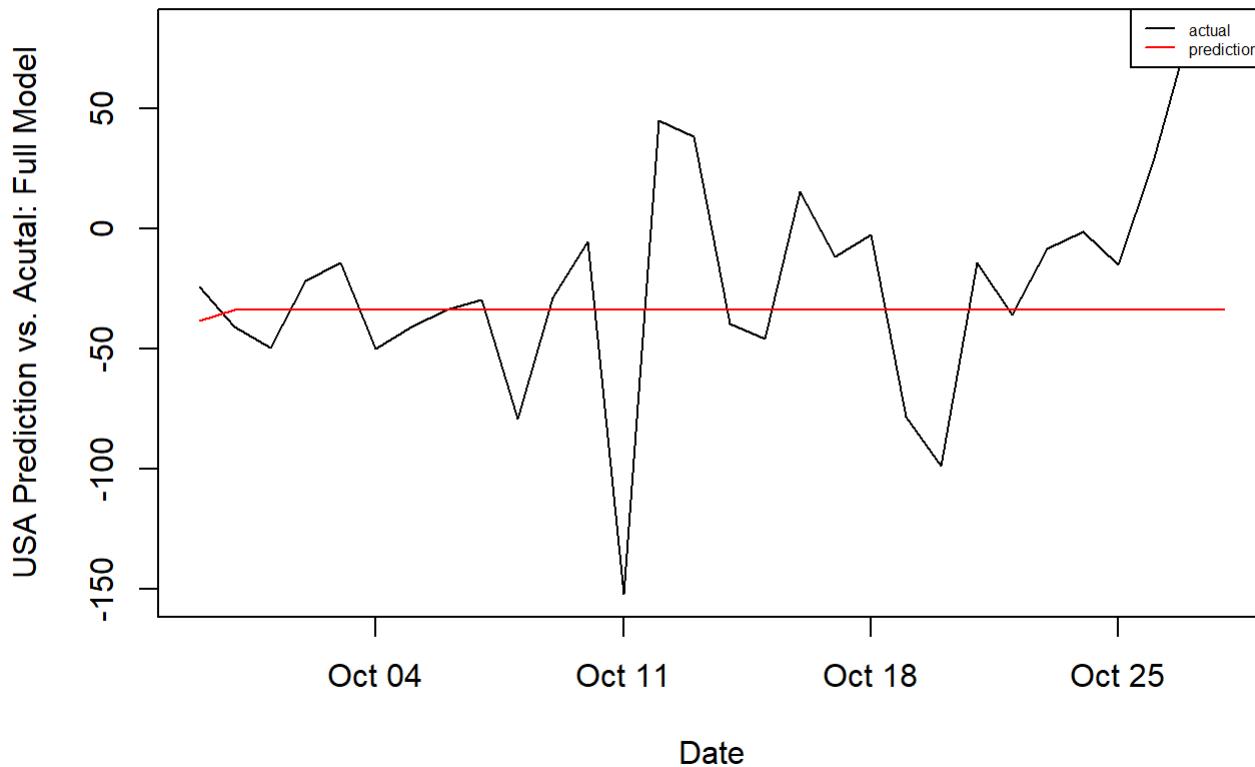
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_covid_usa$residuals_diff[8:409])
colnames(compare_table) = c("predict","actual")

plot(new_covid_usa$Day[8:409],                                     # Draw first time series
      new_covid_usa$residuals_diff[8:409],
      type = "l",
      xlab = "Date",
      ylab = "USA Prediction vs. Actual: Full Model")
lines(new_covid_usa$Day[8:409],                                     # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_covid_usa$Day[380:409],  
      new_covid_usa$residuals_diff[380:409],  
      type = "l",  
      xlab = "Date",  
      ylab = "USA Prediction vs. Acutal: Full Model")  
# Draw first time series  
lines(new_covid_usa$Day[380:409],  
      final_prediction_2,  
      type = "l",  
      col = "red")  
# Draw second time series  
legend(x = "topright", legend=c("actual", "prediction"),  
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE USA Full
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

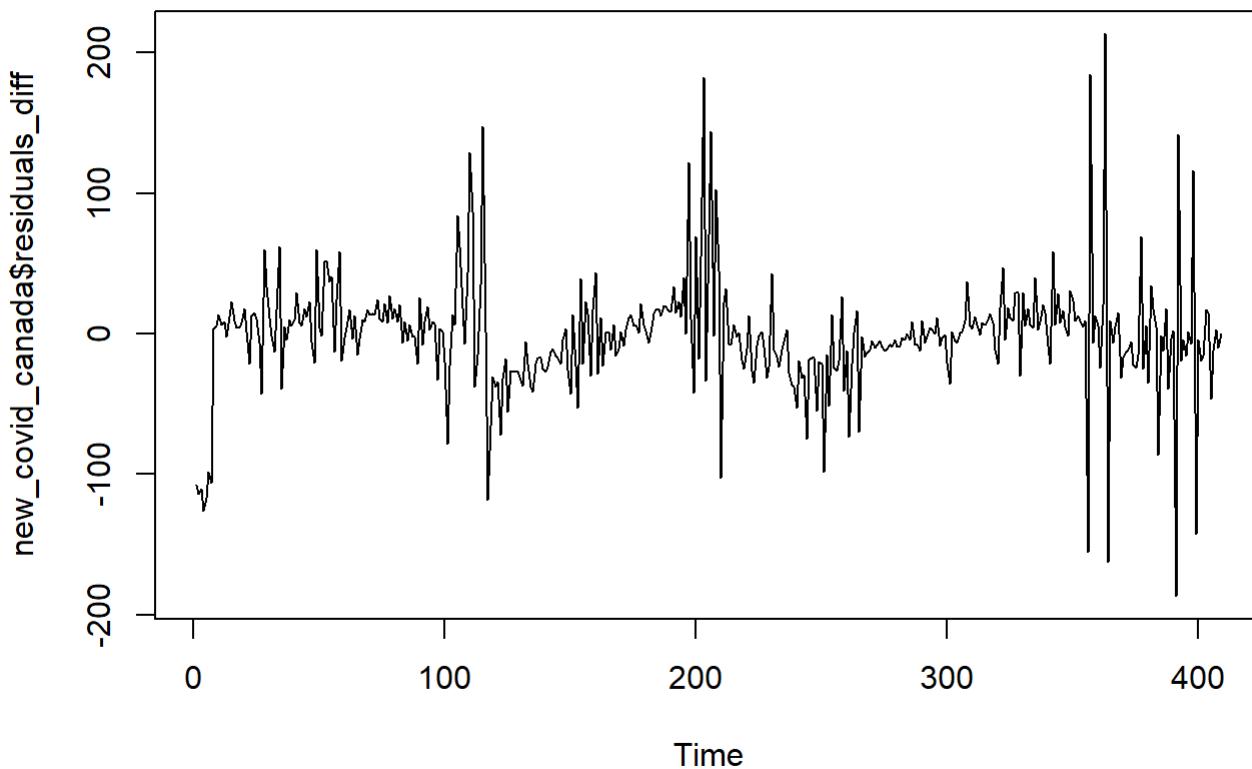
```
## [1] 5487.454
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 2318.572
```

```
new_covid_canada <-
  covid_canada %>% mutate("residuals_diff" = residuals)

#deseasonalized
new_covid_canada$residuals_diff[8:409] = new_covid_canada$residuals_diff %>% diff(7)
plot.ts(new_covid_canada$residuals_diff)
```



```
# ARIMA as a special case of the ARIMAX intercept
regression_model = lm(residuals_diff~1, data = new_covid_canada[8:379,])
summary(regression_model)
```

```
##
## Call:
## lm(formula = residuals_diff ~ 1, data = new_covid_canada[8:379,
##     ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -163.973  -17.786   -0.094  12.084  211.922
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.351     1.909    0.708    0.48
##
## Residual standard error: 36.82 on 371 degrees of freedom
```

```
anova(regression_model)
```

```
## Analysis of Variance Table
##
## Response: residuals_diff
##             Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 371 502898 1355.5
```

```
error = residuals(regression_model)
ts_error = ts(error)

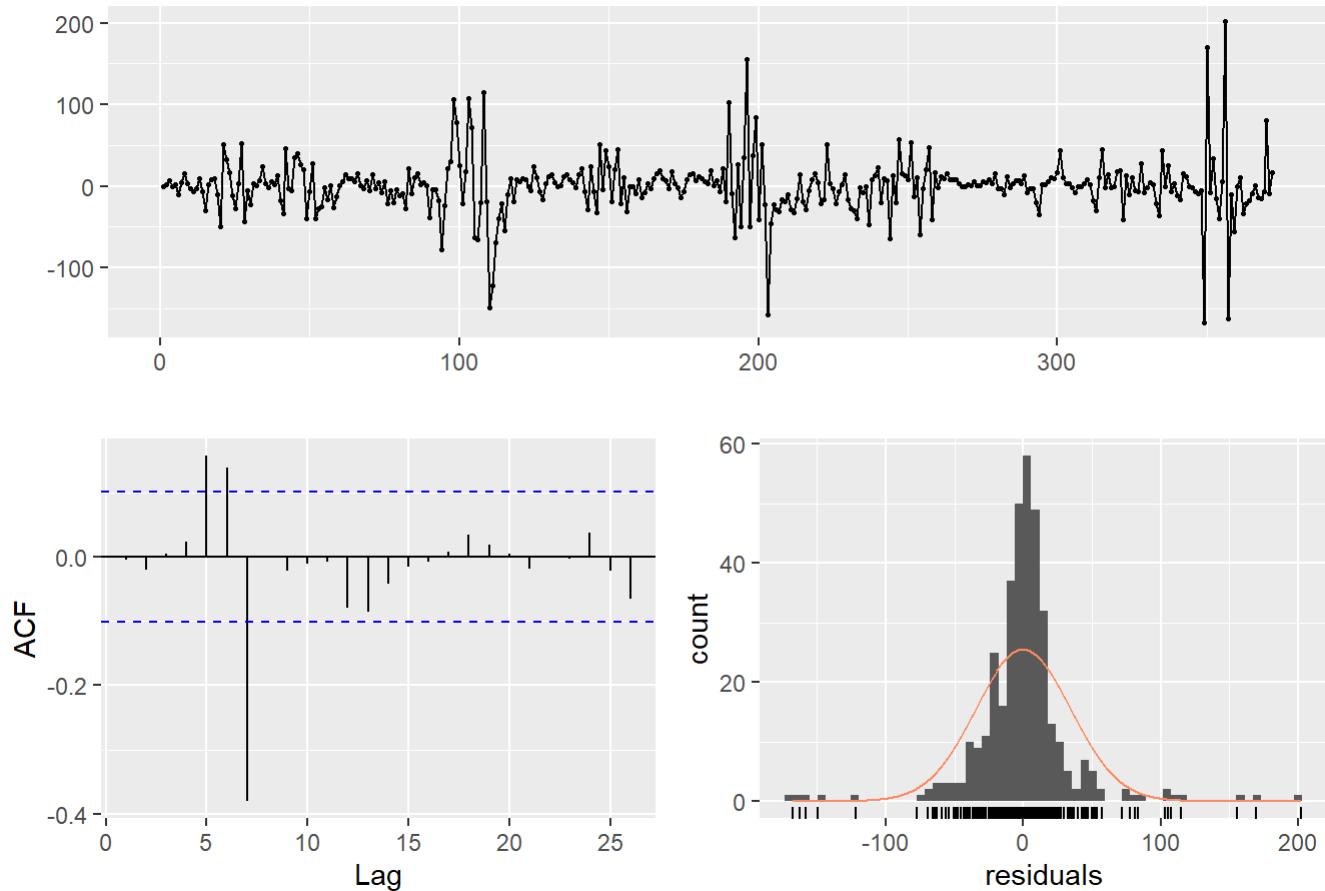
#ARIMA on error

error_model = auto.arima(ts_error,d=1)
fitted_error = fitted(error_model)
summary(error_model)
```

```
## Series: ts_error
## ARIMA(2,1,1)
##
## Coefficients:
##          ar1      ar2      ma1
##     -0.2590 -0.2054 -0.7627
## s.e.  0.0685  0.0638  0.0548
##
## sigma^2 estimated as 1221: log likelihood=-1844.12
## AIC=3696.24  AICc=3696.35  BIC=3711.91
##
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -0.09572003 34.75416 20.42632 187.058 334.9517 0.727128
##                      ACF1
## Training set -0.005289134
```

```
checkresiduals(error_model)
```

Residuals from ARIMA(2,1,1)



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,1,1)  
## Q* = 72.389, df = 7, p-value = 4.855e-13  
##  
## Model df: 3. Total lags used: 10
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

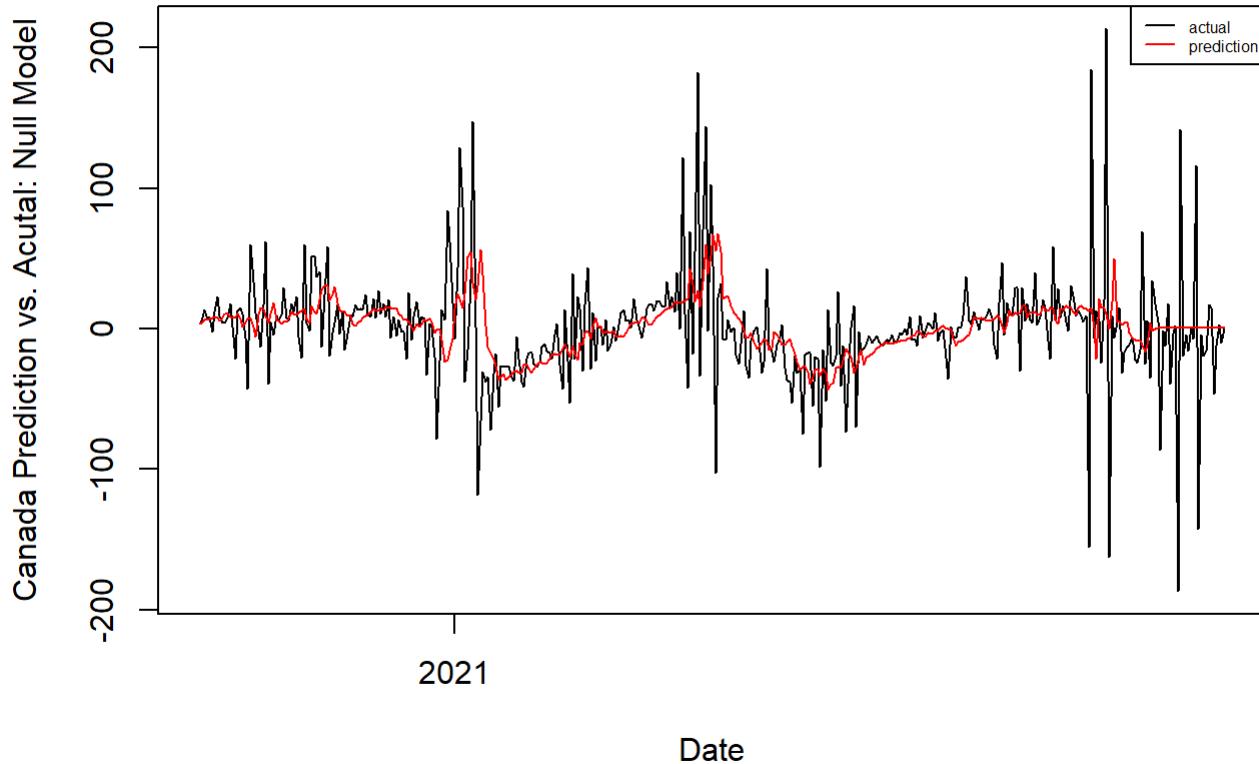
regression_prediction = predict(regression_model, newdata = new_covid_canada[380:409,])
error_forecast = predict(error_model, n.ahead = 30)$pred

final_prediction_2 = regression_prediction + error_forecast

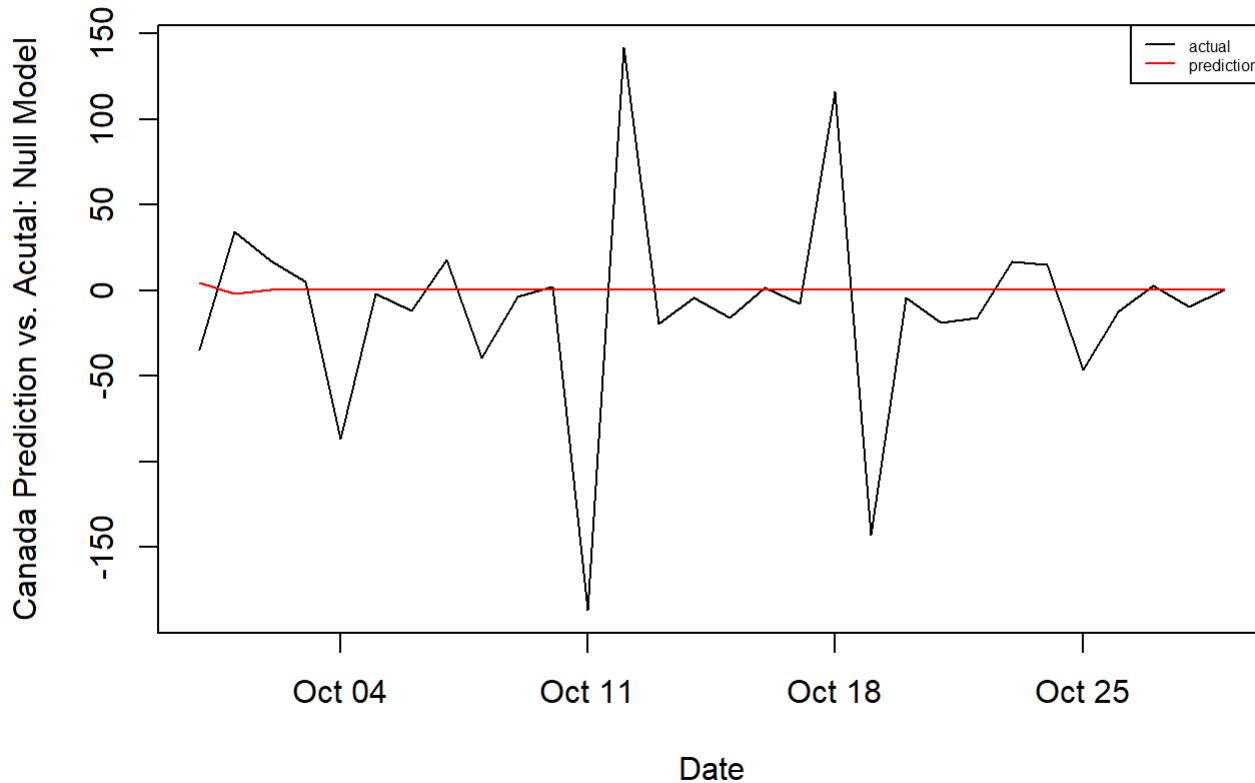
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_covid_canada$residuals_diff[8:409])
colnames(compare_table) = c("predict","actual")

plot(new_covid_canada$Day[8:409],                                     # Draw first time series
      new_covid_canada$residuals_diff[8:409],
      type = "l",
      xlab = "Date",
      ylab = "Canada Prediction vs. Actual: Null Model")
lines(new_covid_canada$Day[8:409],                                      # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_covid_canada$Day[380:409],  
      new_covid_canada$residuals_diff[380:409],  
      type = "l",  
      xlab = "Date",  
      ylab = "Canada Prediction vs. Acutal: Null Model")  
lines(new_covid_canada$Day[380:409],  
      final_prediction_2,  
      type = "l",  
      col = "red")  
legend(x = "topright", legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Canada Null
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 1207.852
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 3530.072
```

```
# ARIMA as a special case of the ARIMAX intercept
regression_model = lm(residuals_diff~as.factor(vaccination_policy), data = new_covid_canada[8:379,])
summary(regression_model)
```

```

## 
## Call:
## lm(formula = residuals_diff ~ as.factor(vaccination_policy),
##      data = new_covid_canada[8:379, ])
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -164.955  -15.516   -1.396   10.756  210.940 
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                 10.276    3.960   2.595  0.00984 **  
## as.factor(vaccination_policy)2  -11.208    7.200  -1.557  0.12039    
## as.factor(vaccination_policy)3  -10.685    5.177  -2.064  0.03973 *   
## as.factor(vaccination_policy)4  -40.518    9.380  -4.320 2.02e-05 ***  
## as.factor(vaccination_policy)5   -7.944    5.168  -1.537  0.12515    
## ---                        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 36.08 on 367 degrees of freedom
## Multiple R-squared:  0.05019,   Adjusted R-squared:  0.03984 
## F-statistic: 4.848 on 4 and 367 DF,  p-value: 0.0008039

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: residuals_diff
##                               Df Sum Sq Mean Sq F value    Pr(>F)    
## as.factor(vaccination_policy) 4  25241  6310.4  4.8485 0.0008039 ***
## Residuals                      367 477656  1301.5
## ---                        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

```

```

error = residuals(regression_model)
ts_error = ts(error)

#ARIMA on error

error_model = auto.arima(ts_error,d=1)
fitted_error = fitted(error_model)
summary(error_model)

```

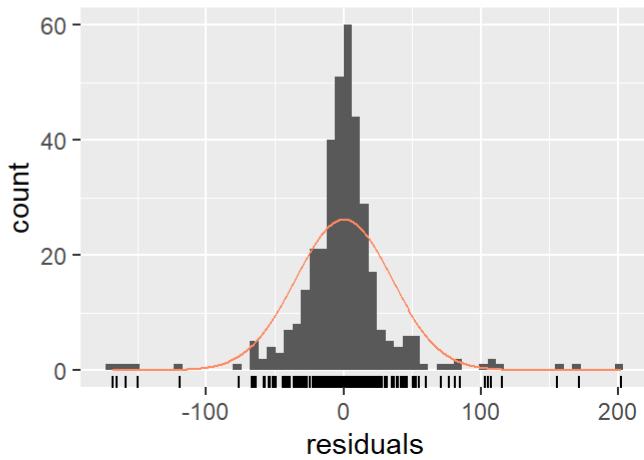
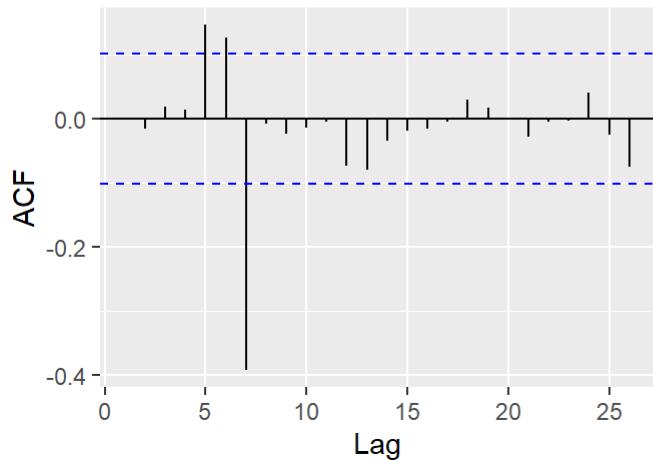
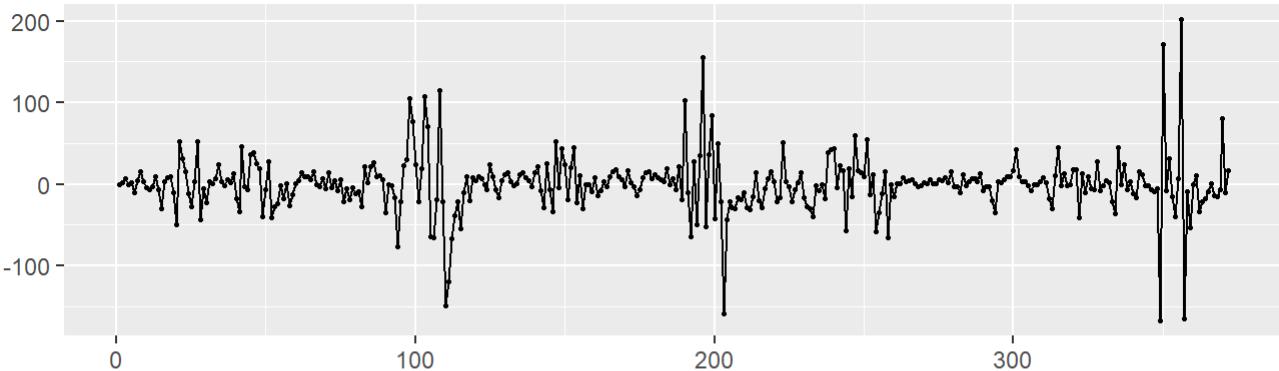
```

## Series: ts_error
## ARIMA(2,1,1)
##
## Coefficients:
##      ar1     ar2     ma1
##     -0.2461  -0.1948  -0.7641
## s.e.   0.0720   0.0665   0.0600
##
## sigma^2 estimated as 1232: log likelihood=-1845.81
## AIC=3699.62   AICc=3699.73   BIC=3715.28
##
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.03516 34.91388 20.54392 -5.806666 263.7701 0.7364794
##          ACF1
## Training set -0.001841229

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(2,1,1)



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,1,1)  
## Q* = 73.708, df = 7, p-value = 2.623e-13  
##  
## Model df: 3. Total lags used: 10
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

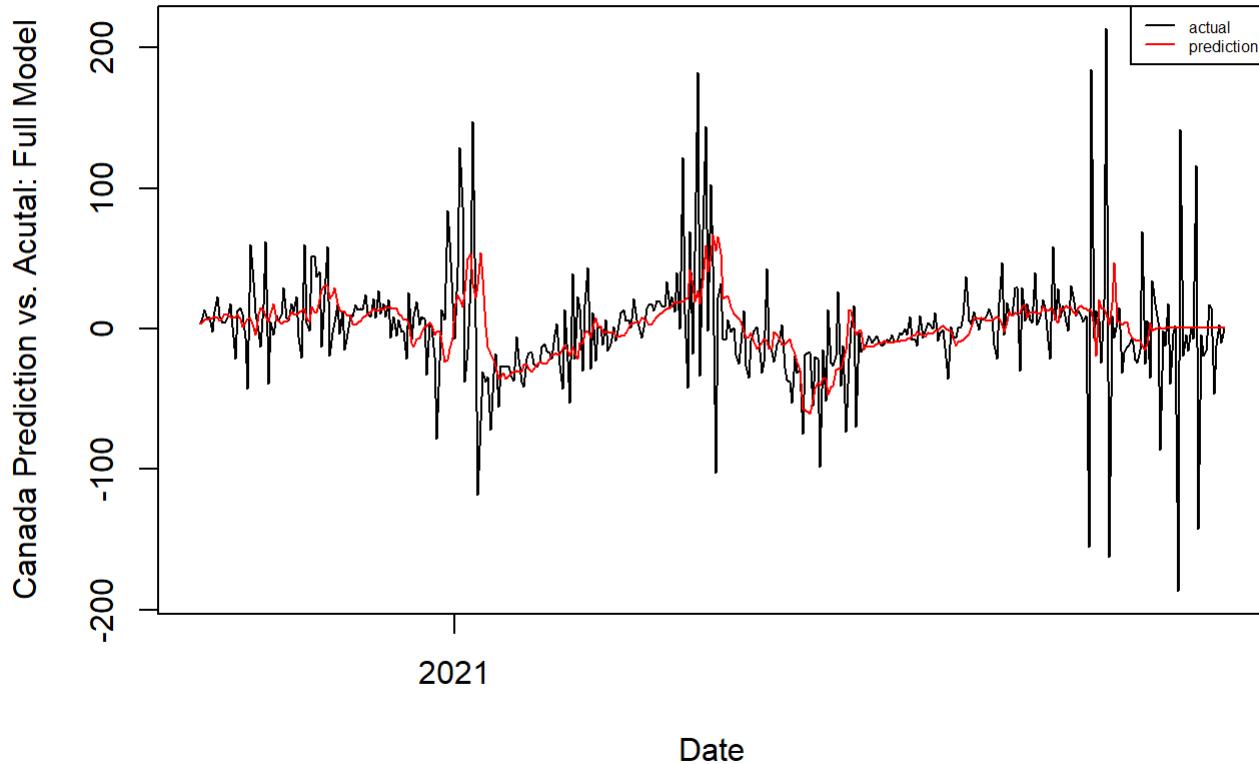
regression_prediction = predict(regression_model, newdata = new_covid_canada[380:409,])
error_forecast = predict(error_model, n.ahead = 30)$pred

final_prediction_2 = regression_prediction + error_forecast

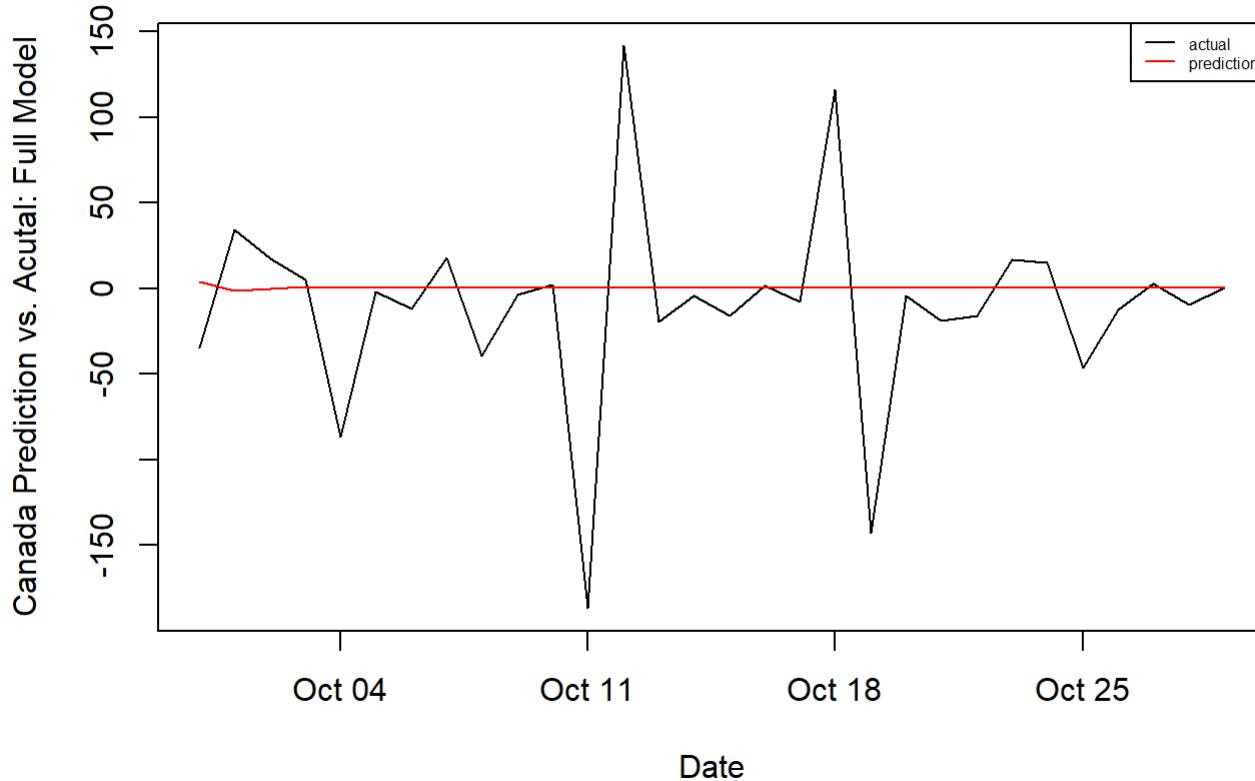
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_covid_canada$residuals_diff[8:409])
colnames(compare_table) = c("predict","actual")

plot(new_covid_canada$Day[8:409],  
      new_covid_canada$residuals_diff[8:409],  
      type = "l",  
      xlab = "Date",  
      ylab = "Canada Prediction vs. Acutal: Full Model")  
lines(new_covid_canada$Day[8:409],  
      compare_table$predict,  
      type = "l",  
      col = 'red')  
legend(x = "topright",legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_covid_canada$Day[380:409],  
      new_covid_canada$residuals_diff[380:409],  
      type = "l",  
      xlab = "Date",  
      ylab = "Canada Prediction vs. Acutal: Full Model")  
lines(new_covid_canada$Day[380:409],  
      final_prediction_2,  
      type = "l",  
      col = "red")  
legend(x = "topright", legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Canada Full
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 1218.979
```

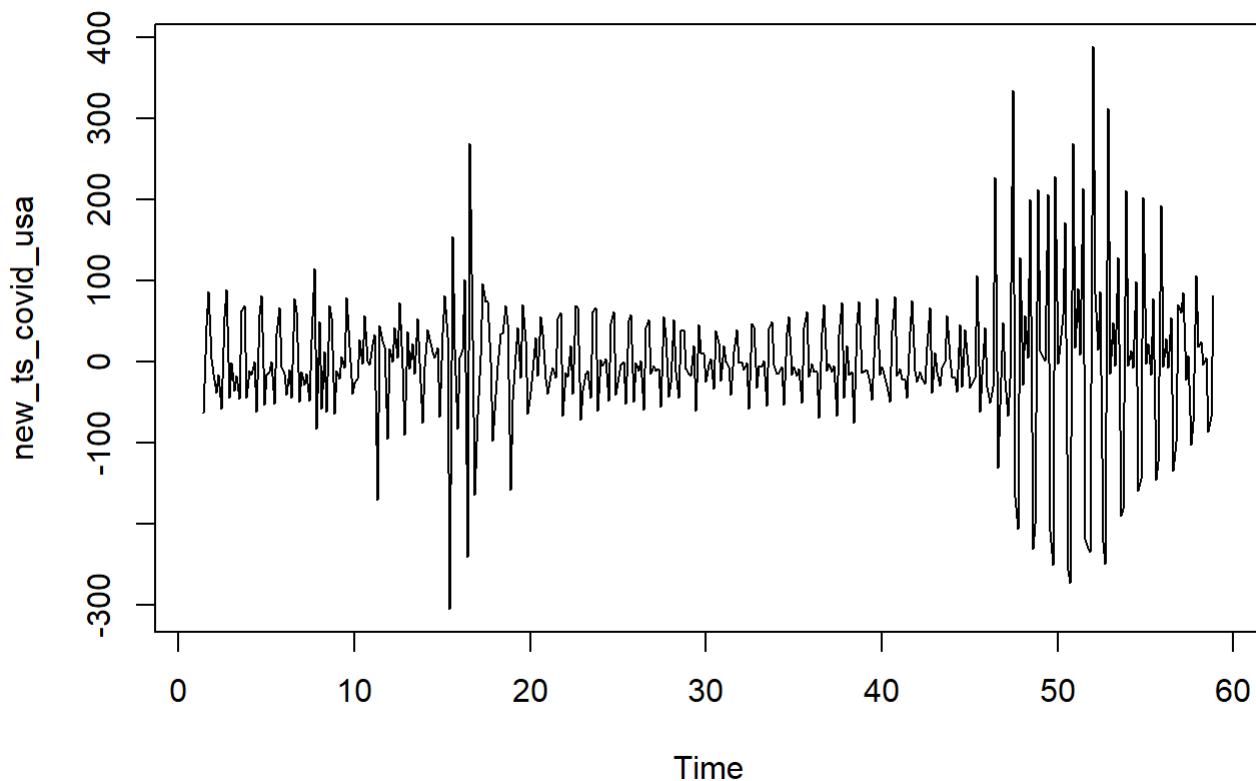
```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 3528.12
```

```
#USA : Yt - Mt - St
ts_covid_usa = ts(covid_usa$new_cases_per_million, frequency = 7)
#decompose(ts_covid_usa)

new_ts_covid_usa = ts_covid_usa - decompose(ts_covid_usa)$seasonal - decompose(ts_covid_usa)$trend

plot.ts(new_ts_covid_usa)
```



```
new_covid_usa <-
  new_covid_usa %>% mutate("new_ts_covid_usa" = new_ts_covid_usa) %>%
  na.omit()
```

```
regression_model = lm(new_ts_covid_usa~1, data = new_covid_usa[1:373,])
summary(regression_model)
```

```

## 
## Call:
## lm(formula = new_ts_covid_usa ~ 1, data = new_covid_usa[1:373,
##   ])
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -304.65 -28.81  -4.82   38.92 388.12 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -0.01713   4.26744  -0.004    0.997    
## 
## Residual standard error: 82.42 on 372 degrees of freedom

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: new_ts_covid_usa
##           Df Sum Sq Mean Sq F value Pr(>F)    
## Residuals 372 2526888 6792.7

```

```

error = residuals(regression_model)
ts_error = ts(error)

#ARIMA on error

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)

```

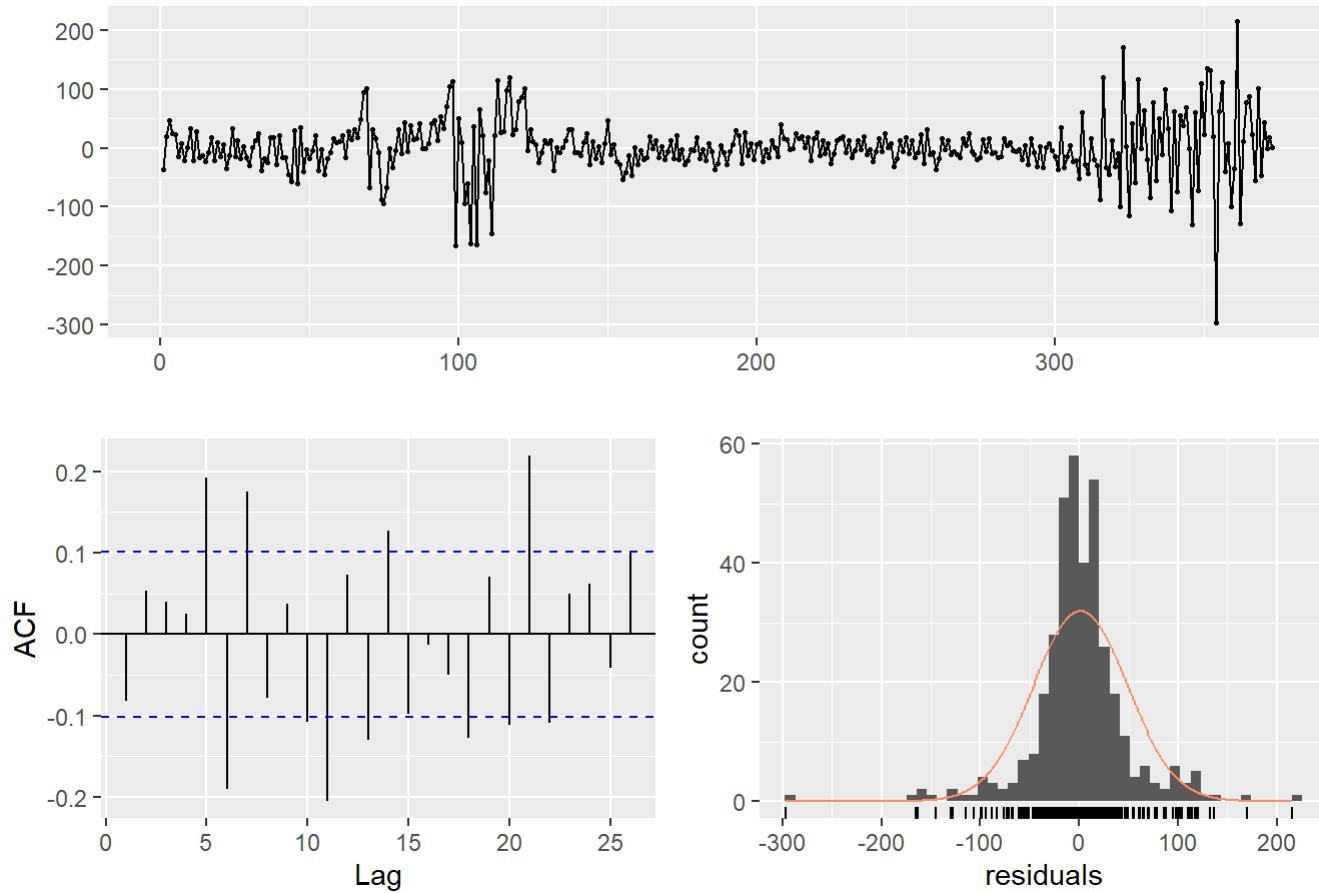
```

## Series: ts_error
## ARIMA(5,0,3) with zero mean
## 
## Coefficients:
##         ar1      ar2      ar3      ar4      ar5      ma1      ma2      ma3  
##       -0.0456  -0.8651  -0.2450  -0.3038  -0.5619  -0.7014  0.5365  -0.6413 
## s.e.   0.0536   0.0472   0.0651   0.0422   0.0466   0.0595   0.0508   0.0661 
## 
## sigma^2 estimated as 2326: log likelihood=-1973.94
## AIC=3965.89   AICc=3966.38   BIC=4001.18
## 
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE      ACF1  
## Training set 1.04331 47.71312 31.00802 -49.67664 453.6815 0.4065845 -0.08202149

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(5,0,3) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(5,0,3) with zero mean  
## Q* = 67.776, df = 3, p-value = 1.277e-14  
##  
## Model df: 8. Total lags used: 11
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

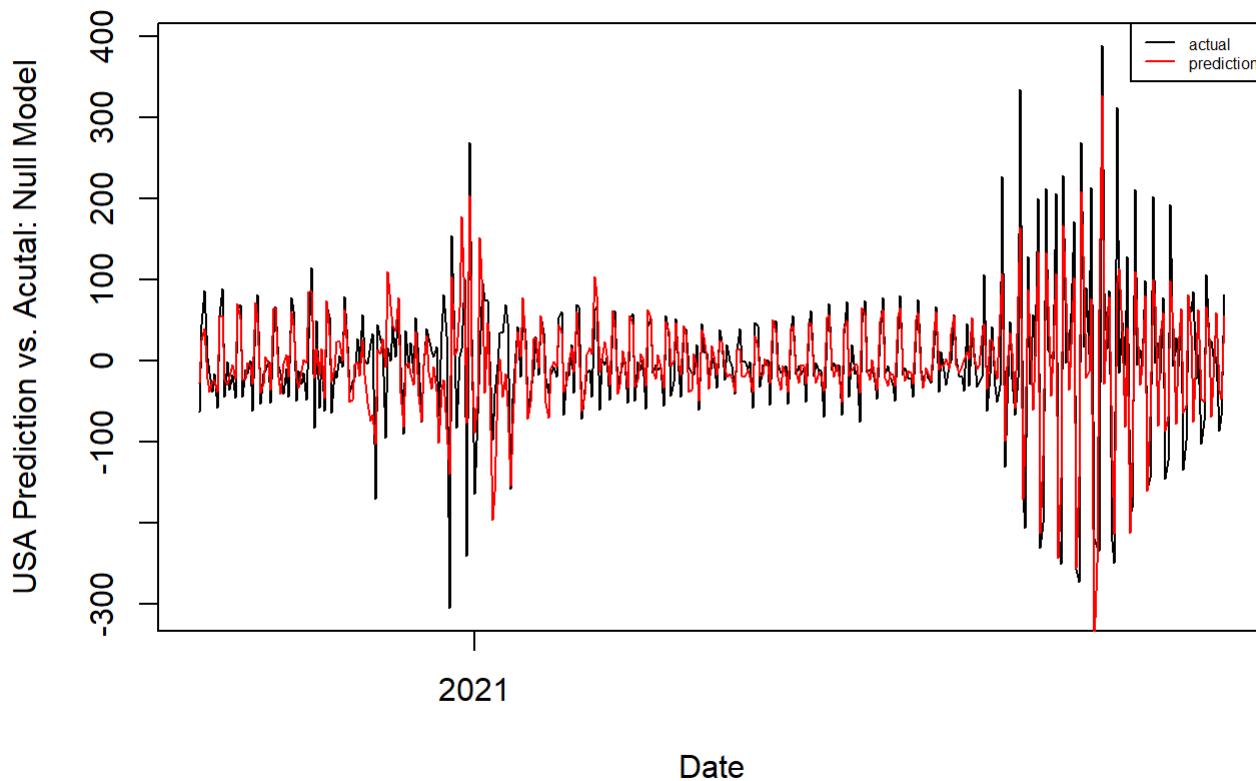
regression_prediction = predict(regression_model, newdata = new_covid_usa[374:403,])
error_forecast = predict(error_model, n.ahead = 30)$pred

final_prediction_2 = regression_prediction + error_forecast

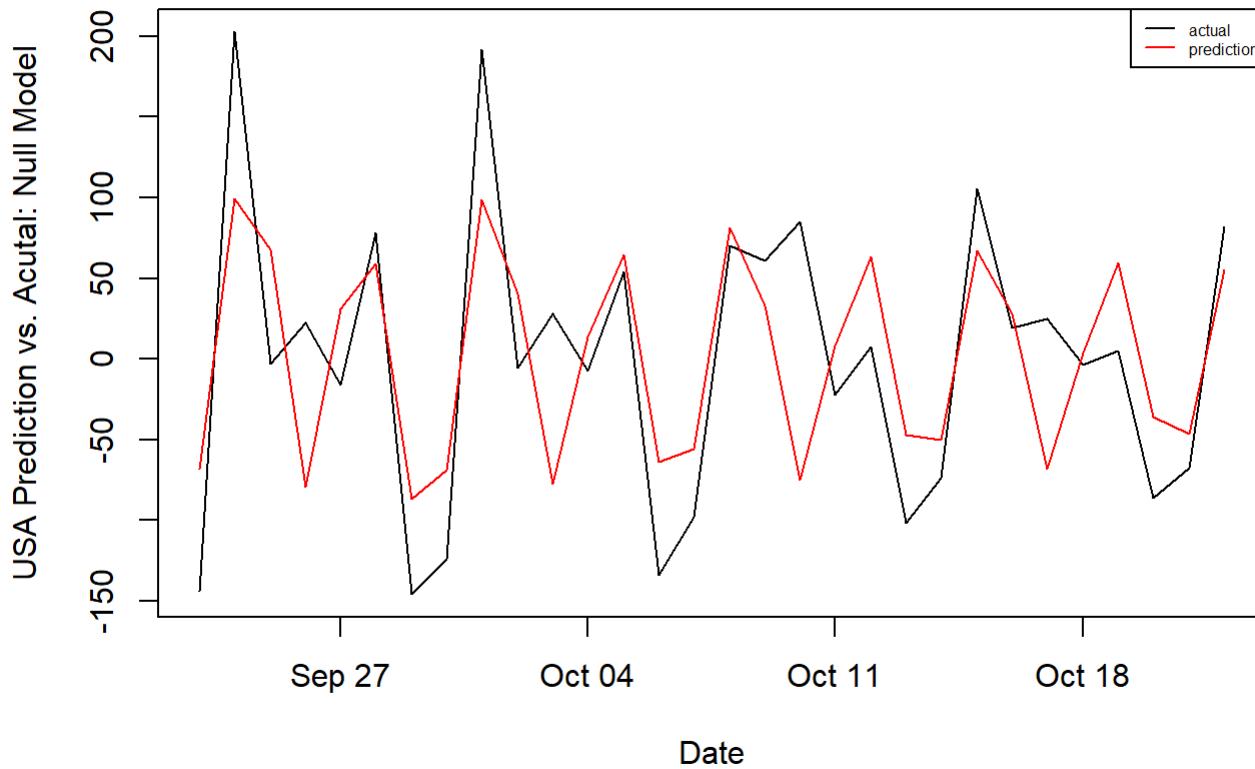
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_covid_usa$new_ts_covid_usa[1:403])
colnames(compare_table) = c("predict","actual")

plot(covid_usa$Day[1:403],                                     # Draw first time series
      new_covid_usa$new_ts_covid_usa[1:403],
      type = "l",
      xlab = "Date",
      ylab = "USA Prediction vs. Actual: Null Model")
lines(covid_usa$Day[1:403],                                     # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(covid_usa$Day[374:403],                                     # Draw first time series
     new_covid_usa$new_ts_covid_usa[374:403],
     type = "l",
     xlab = "Date",
     ylab = "USA Prediction vs. Acutal: Null Model")
lines(covid_usa$Day[374:403],                                     # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE USA Null Method2
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 2276.541
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 4041.949
```

```
regression_model = lm(new_ts_covid_usa~as.factor(vaccination_policy), data = new_covid_usa[1:373,])
summary(regression_model)
```

```

## 
## Call:
## lm(formula = new_ts_covid_usa ~ as.factor(vaccination_policy),
##     data = new_covid_usa[1:373, ])
## 
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -303.60  -28.50   -4.74   38.76  388.46 
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                 0.6476    8.8956   0.073   0.942    
## as.factor(vaccination_policy)1 -1.7141   13.0738  -0.131   0.896    
## as.factor(vaccination_policy)2  1.5087   17.3555   0.087   0.931    
## as.factor(vaccination_policy)3 -0.5040   20.5756  -0.024   0.980    
## as.factor(vaccination_policy)4 -0.7638   22.5702  -0.034   0.973    
## as.factor(vaccination_policy)5 -0.9990   11.2668  -0.089   0.929    
## 
## Residual standard error: 82.97 on 367 degrees of freedom
## Multiple R-squared:  0.0001125, Adjusted R-squared:  -0.01351 
## F-statistic: 0.008257 on 5 and 367 DF,  p-value: 1

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: new_ts_covid_usa
##                               Df  Sum Sq Mean Sq F value Pr(>F)    
## as.factor(vaccination_policy) 5    284    56.8  0.0083    1    
## Residuals                  367 2526604  6884.5  

```

```

error = residuals(regression_model)
ts_error = ts(error)

```

```
#ARIMA on error
```

```

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)

```

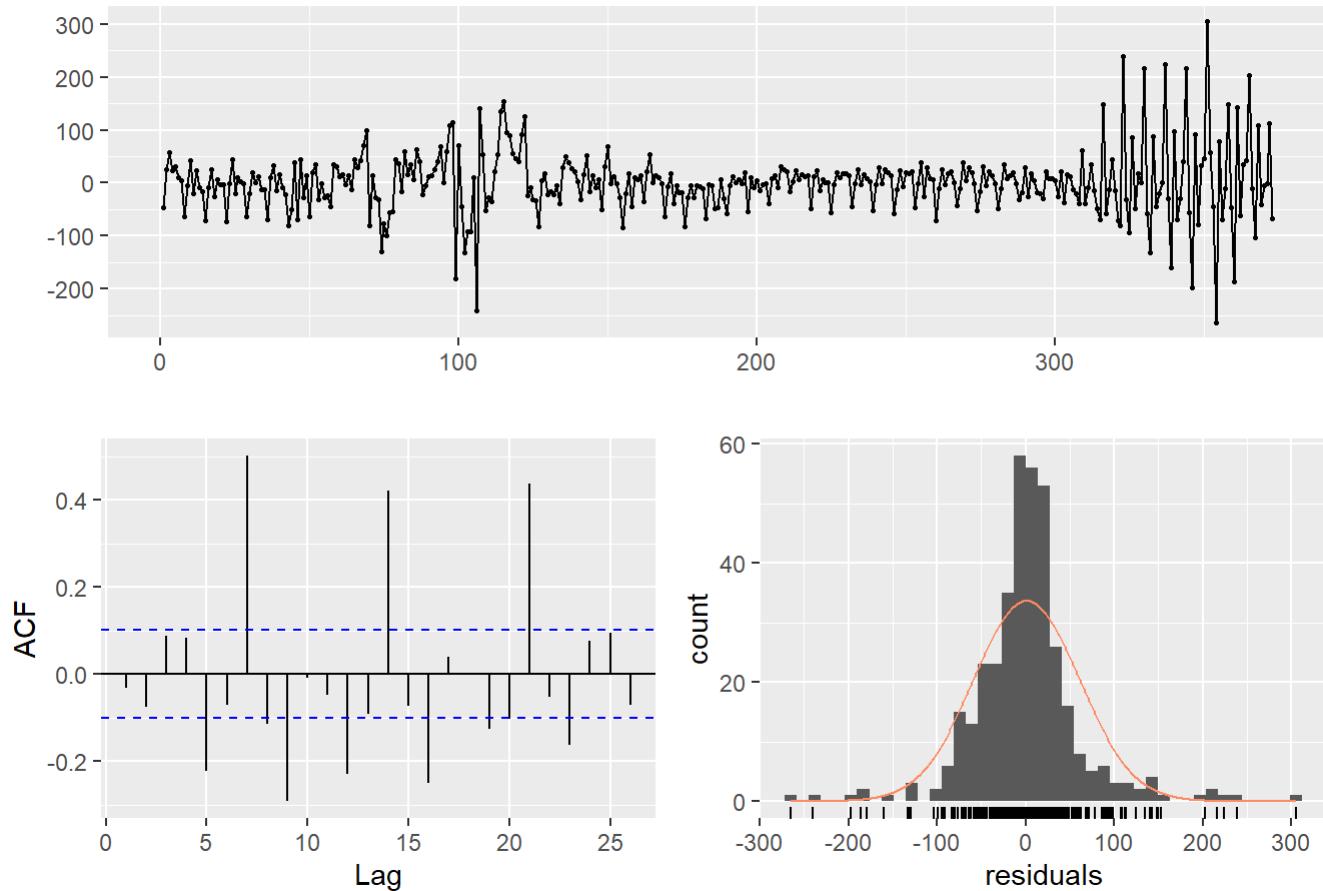
```

## Series: ts_error
## ARIMA(2,0,2) with zero mean
##
## Coefficients:
##      ar1     ar2     ma1     ma2
##      0.8430 -0.5297 -1.5722  0.6614
## s.e.  0.0632  0.0541  0.0508  0.0632
##
## sigma^2 estimated as 3624: log likelihood=-2057.01
## AIC=4124.02   AICc=4124.18   BIC=4143.63
##
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.8497896 59.87576 39.16667 191.5565 368.7459 0.5136889
##          ACF1
## Training set -0.03077804

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(2,0,2) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,0,2) with zero mean  
## Q* = 162.52, df = 6, p-value < 2.2e-16  
##  
## Model df: 4. Total lags used: 10
```

#goodness of fit

```
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

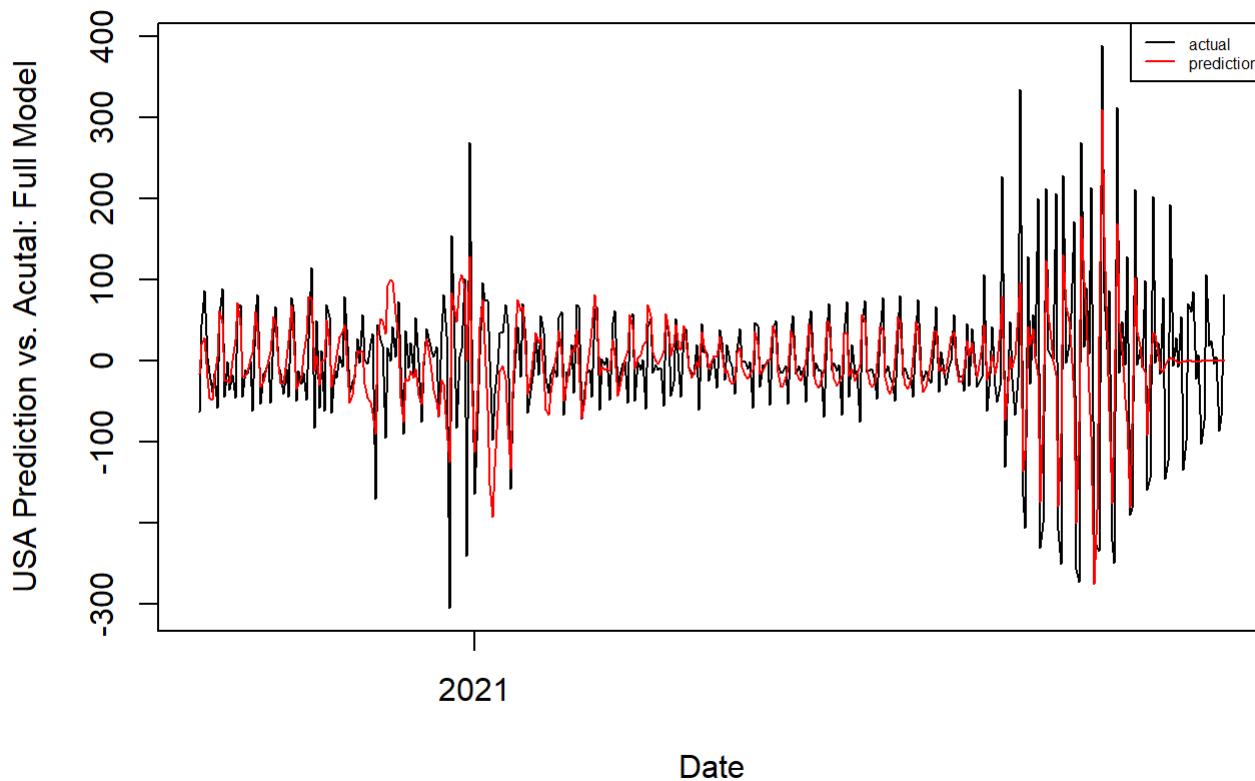
regression_prediction = predict(regression_model, newdata = new_covid_usa[374:403,])
error_forecast = predict(error_model, n.ahead = 30)$pred

final_prediction_2 = regression_prediction + error_forecast
```

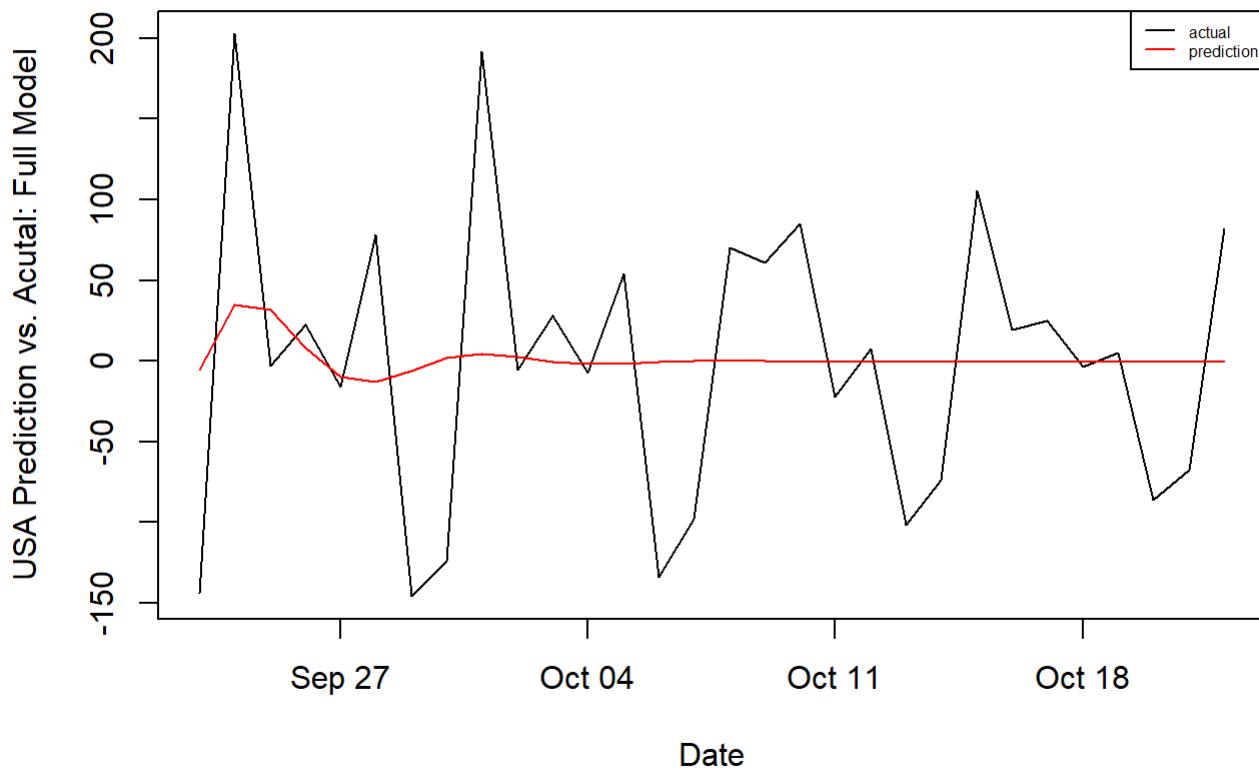
#compare prediction and actual

```
compare_table = data.frame(c(final_prediction,final_prediction_2),new_covid_usa$new_ts_covid_usa
[1:403])
colnames(compare_table) = c("predict","actual")
```

```
plot(covid_usa$Day[1:403],                                     # Draw first time series
     new_covid_usa$new_ts_covid_usa[1:403],
     type = "l",
     xlab = "Date",
     ylab = "USA Prediction vs. Actual: Full Model")
lines(covid_usa$Day[1:403],                                     # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(covid_usa$Day[374:403],                                     # Draw first time series
     new_covid_usa$new_ts_covid_usa[374:403],
     type = "l",
     xlab = "Date",
     ylab = "USA Prediction vs. Acutal: Full Model")
lines(covid_usa$Day[374:403],                                     # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE USA Full Method 2
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 3585.106
```

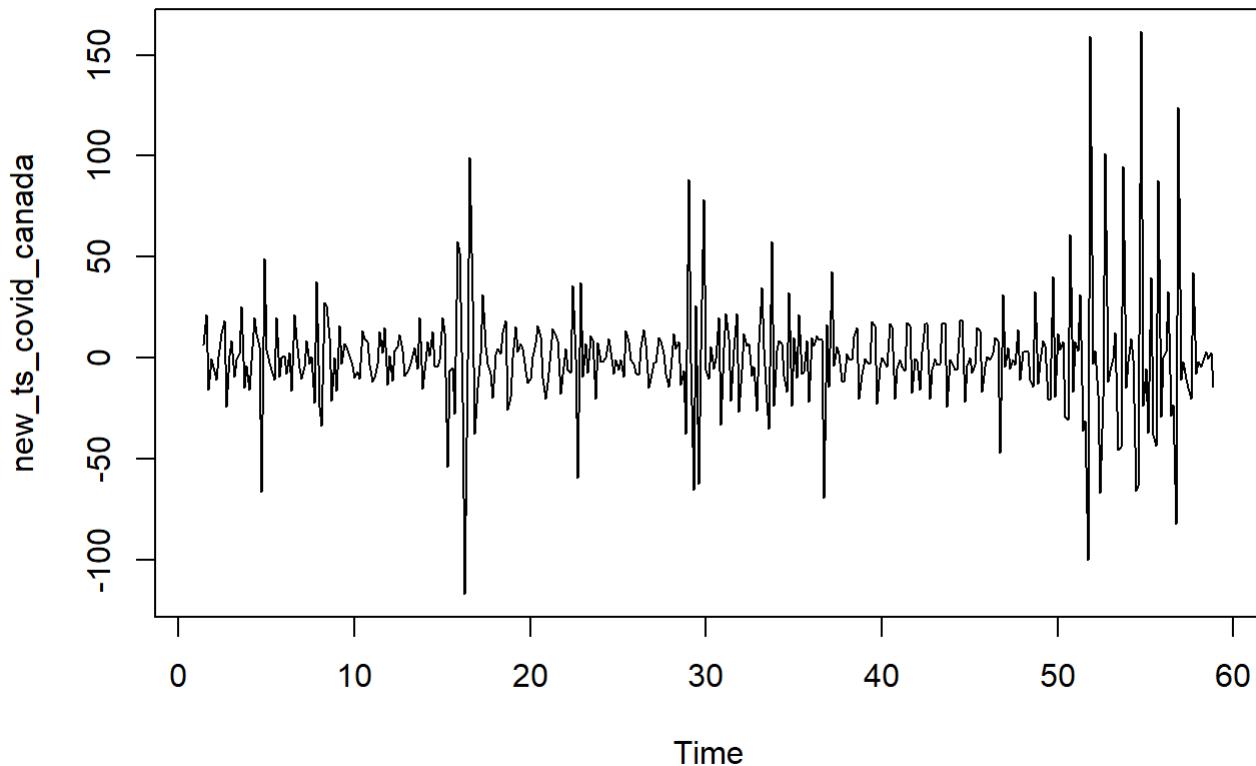
```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 7410.106
```

```
#Canada : Yt - Mt - St
ts_covid_canada = ts(covid_canada$new_cases_per_million, frequency = 7)
#decompose(ts_covid_canada)

new_ts_covid_canada = ts_covid_canada - decompose(ts_covid_canada)$seasonal - decompose(ts_covid_canada)$trend

plot.ts(new_ts_covid_canada)
```



```
new_covid_canada <-
  new_covid_canada %>% mutate("new_ts_covid_canada" = new_ts_covid_canada) %>%
  na.omit()
```

```
regression_model = lm(new_ts_covid_canada~1, data = new_covid_canada[1:373,])
summary(regression_model)
```

```
##
## Call:
## lm(formula = new_ts_covid_canada ~ 1, data = new_covid_canada[1:373,
##   ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -116.663  -8.996  -0.421   8.972 159.511 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -0.2513     1.2566    -0.2    0.842    
## 
## Residual standard error: 24.27 on 372 degrees of freedom
```

```
anova(regression_model)
```

```
## Analysis of Variance Table
##
## Response: new_ts_covid_canada
##           Df Sum Sq Mean Sq F value Pr(>F)    
## Residuals 372 219109      589
```

```
error = residuals(regression_model)
ts_error = ts(error)

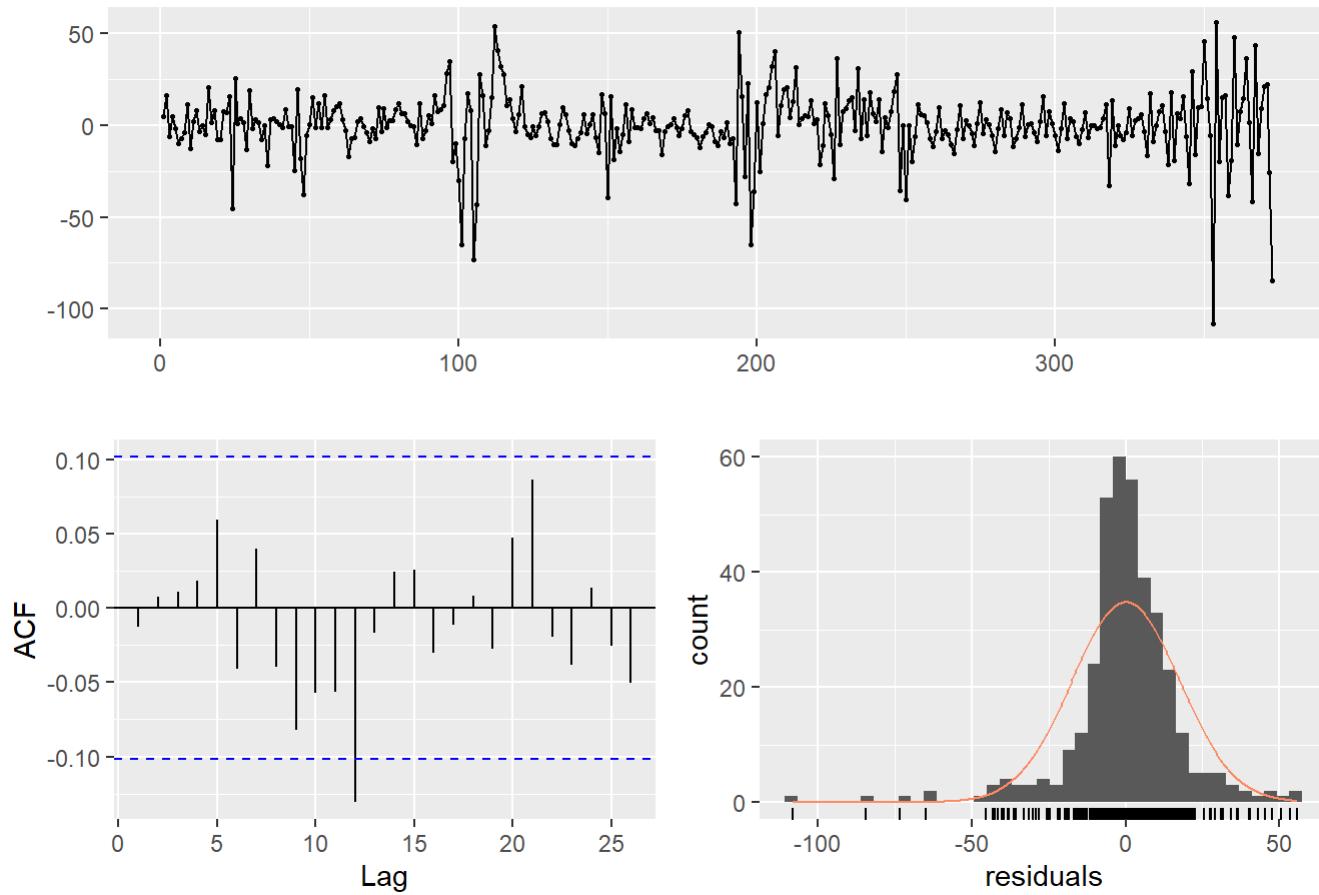
#ARIMA on error

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)
```

```
## Series: ts_error
## ARIMA(5,0,1) with non-zero mean
##
## Coefficients:
##         ar1      ar2      ar3      ar4      ar5      ma1    mean
##        -0.0057  -0.4892  -0.3502  -0.2449  -0.2394  -0.7730  0.2427
## s.e.    0.0730   0.0590   0.0656   0.0591   0.0640   0.0604  0.0897
## 
## sigma^2 estimated as 312.8: log likelihood=-1598.8
## AIC=3213.59  AICc=3213.99  BIC=3244.96
## 
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.01631147 17.51833 11.42657 -8.796033 310.0721 0.5043037
##                      ACF1
## Training set -0.01265951
```

```
checkresiduals(error_model1)
```

Residuals from ARIMA(5,0,1) with non-zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(5,0,1) with non-zero mean  
## Q* = 7.2743, df = 3, p-value = 0.06365  
##  
## Model df: 7. Total lags used: 10
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

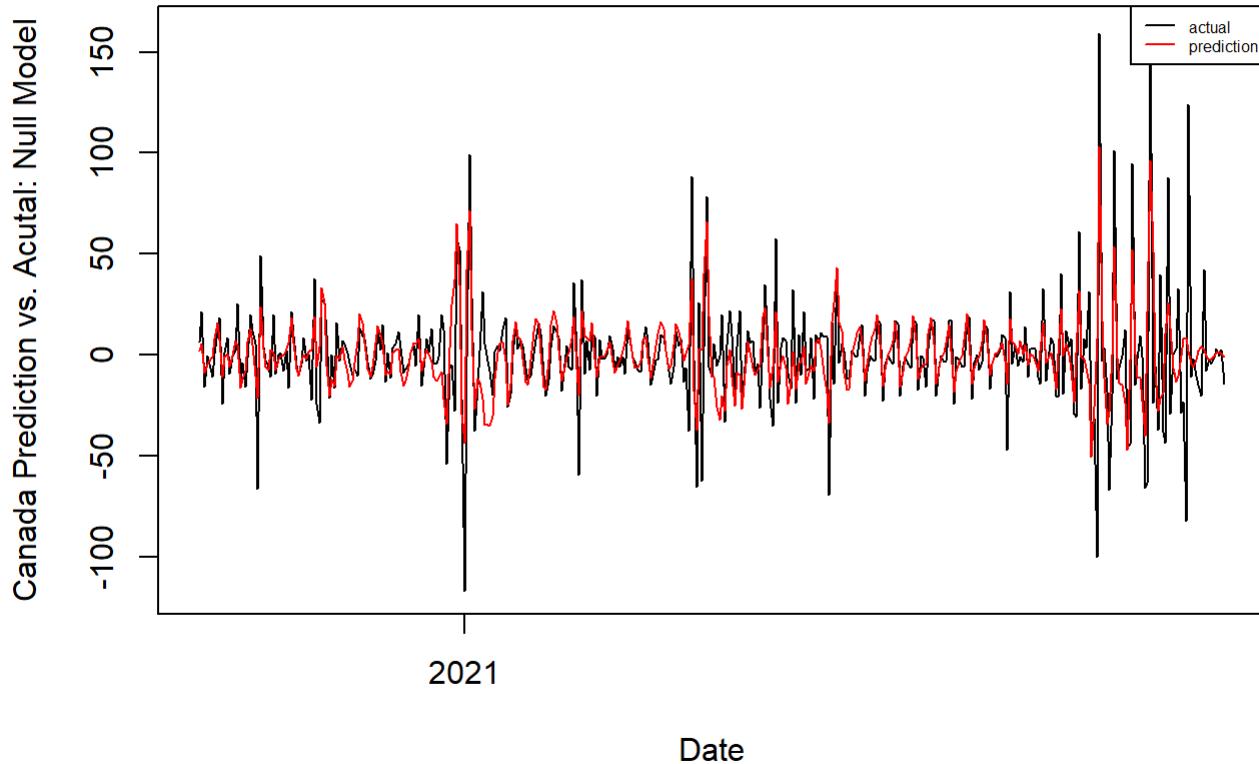
regression_prediction = predict(regression_model, newdata = new_covid_canada[374:403,])
error_forecast = predict(error_model, n.ahead = 30)$pred

final_prediction_2 = regression_prediction + error_forecast

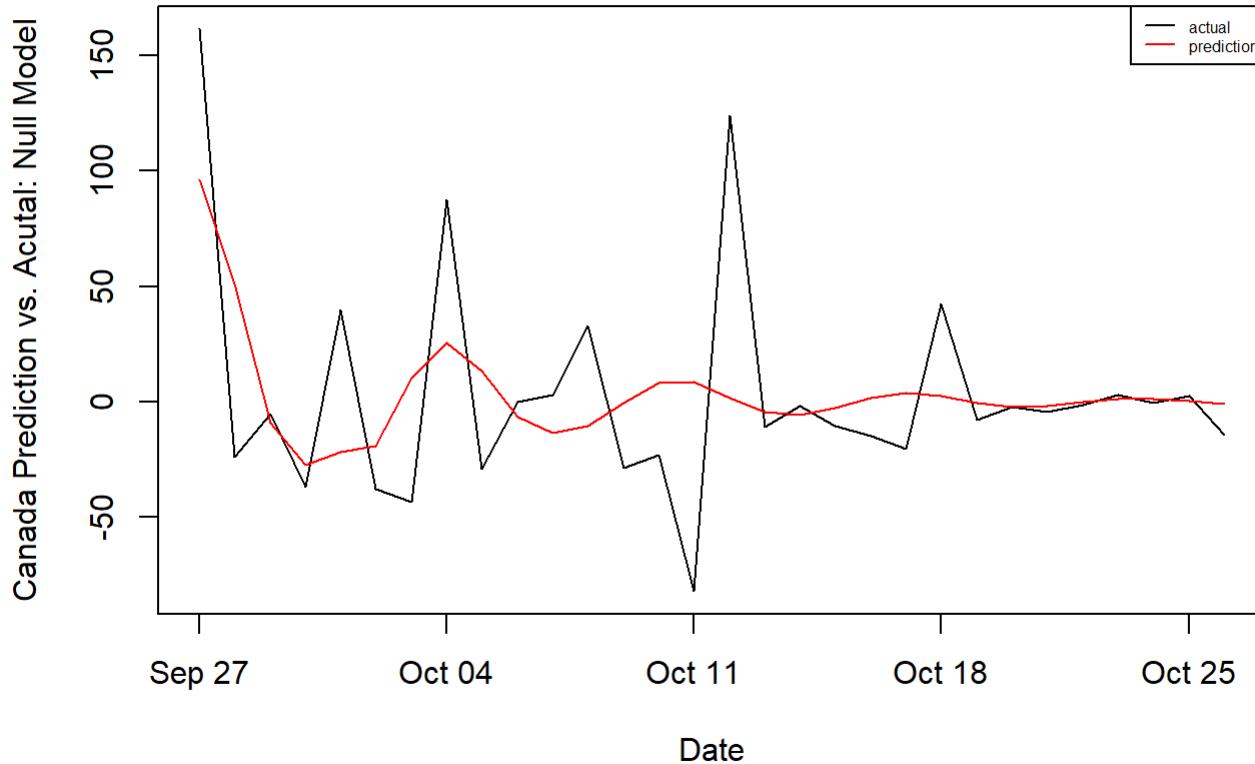
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_covid_canada$new_ts_covid_canada[1:403])
colnames(compare_table) = c("predict","actual")

plot(new_covid_canada$Day[1:403],                                     # Draw first time series
      new_covid_canada$new_ts_covid_canada[1:403],
      type = "l",
      xlab = "Date",
      ylab = "Canada Prediction vs. Actual: Null Model")
lines(new_covid_canada$Day[1:403],                                      # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_covid_canada$Day[374:403],                                     # Draw first time series
     new_covid_canada$new_ts_covid_canada[374:403],
     type = "l",
     xlab = "Date",
     ylab = "Canada Prediction vs. Acutal: Null Model")
lines(new_covid_canada$Day[374:403],                                     # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Canada Null Method 2
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 306.8921
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 1752.865
```

```
regression_model = lm(new_ts_covid_canada~as.factor(vaccination_policy), data = new_covid_canada[1:373,])
summary(regression_model)
```

```

## 
## Call:
## lm(formula = new_ts_covid_canada ~ as.factor(vaccination_policy),
##      data = new_covid_canada[1:373, ])
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -116.340  -9.528  -0.078   9.255  160.184 
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                 0.2814    2.6154   0.108   0.914    
## as.factor(vaccination_policy)2  -0.8559    4.8344  -0.177   0.860    
## as.factor(vaccination_policy)3  -0.1180    3.4536  -0.034   0.973    
## as.factor(vaccination_policy)4  -0.8518    6.3169  -0.135   0.893    
## as.factor(vaccination_policy)5  -1.2065    3.4663  -0.348   0.728    
## 
## Residual standard error: 24.4 on 368 degrees of freedom
## Multiple R-squared:  0.0004683, Adjusted R-squared:  -0.0104  
## F-statistic: 0.0431 on 4 and 368 DF,  p-value: 0.9965

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: new_ts_covid_canada
##                               Df Sum Sq Mean Sq F value Pr(>F)    
## as.factor(vaccination_policy) 4     103   25.65  0.0431 0.9965  
## Residuals                      368 219006  595.12

```

```

error = residuals(regression_model)
ts_error = ts(error)

#ARIMA on error

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)

```

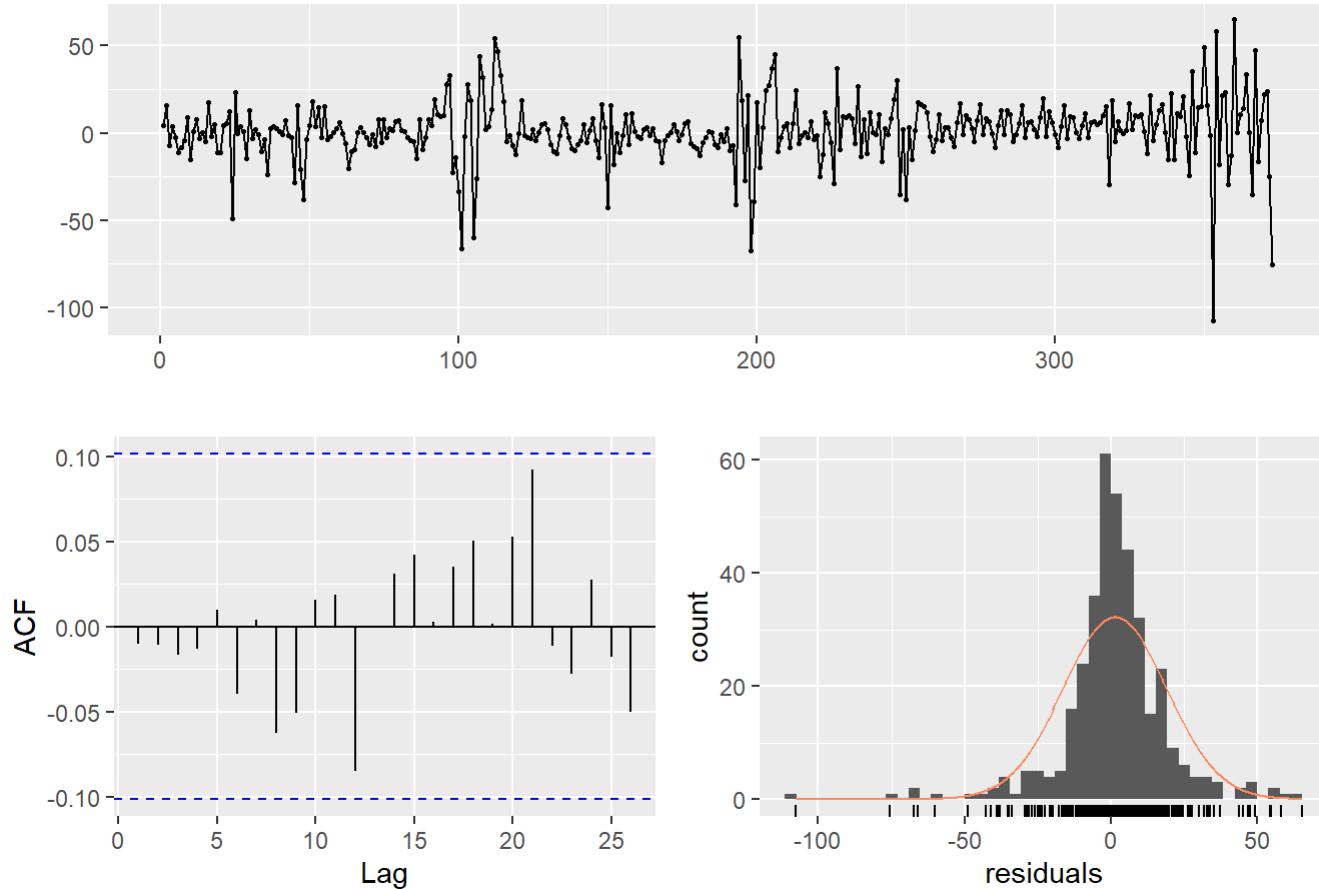
```

## Series: ts_error
## ARIMA(5,0,2) with zero mean
##
## Coefficients:
##      ar1     ar2     ar3     ar4     ar5     ma1     ma2
##      0.4319 -0.6236 -0.2139 -0.1885 -0.2052 -1.1729  0.4948
##  s.e.  0.1398  0.0638  0.0853  0.0624  0.0721  0.1338  0.1166
##
## sigma^2 estimated as 322.3: log likelihood=-1604.25
## AIC=3224.5   AICc=3224.89   BIC=3255.87
##
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 1.25696 17.78253 11.55249 189.9477 556.9271 0.5099318 -0.009867794

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(5,0,2) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(5,0,2) with zero mean  
## Q* = 3.473, df = 3, p-value = 0.3243  
##  
## Model df: 7. Total lags used: 10
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

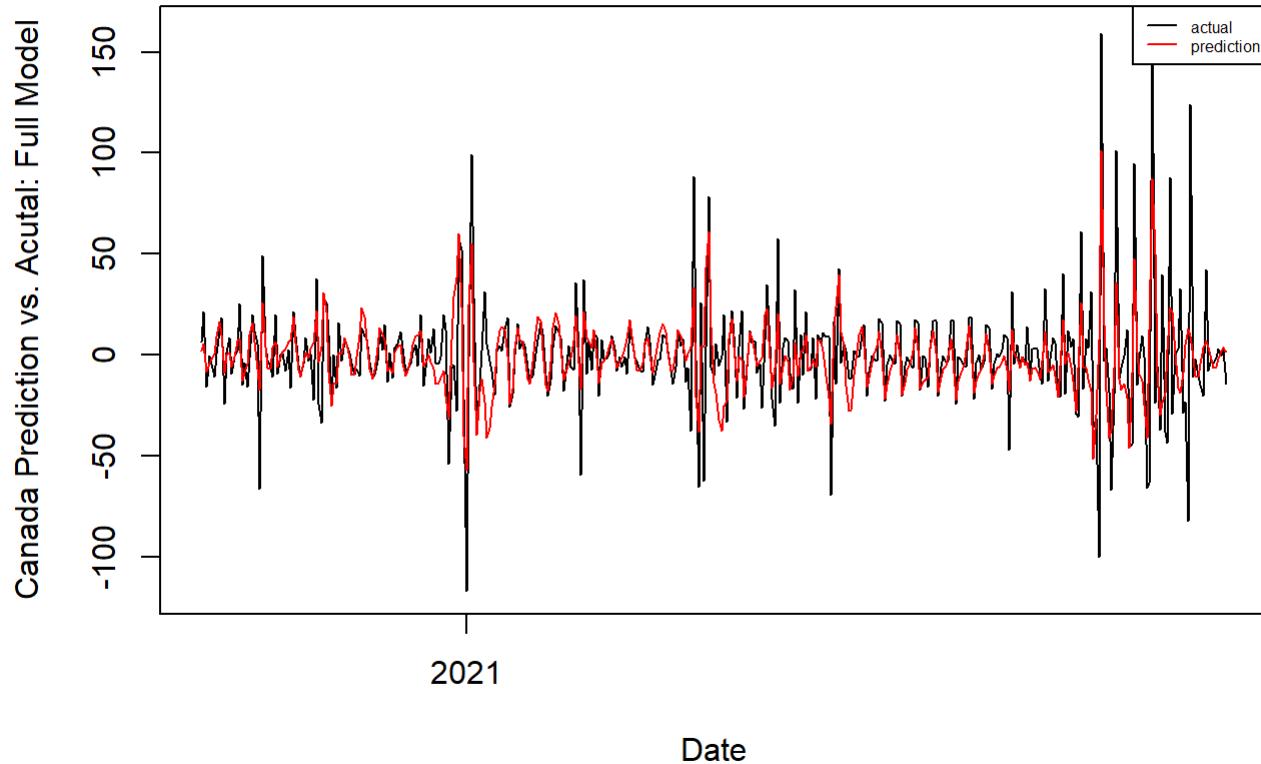
regression_prediction = predict(regression_model, newdata = new_covid_canada[374:403,])
error_forecast = predict(error_model, n.ahead = 30)$pred

final_prediction_2 = regression_prediction + error_forecast

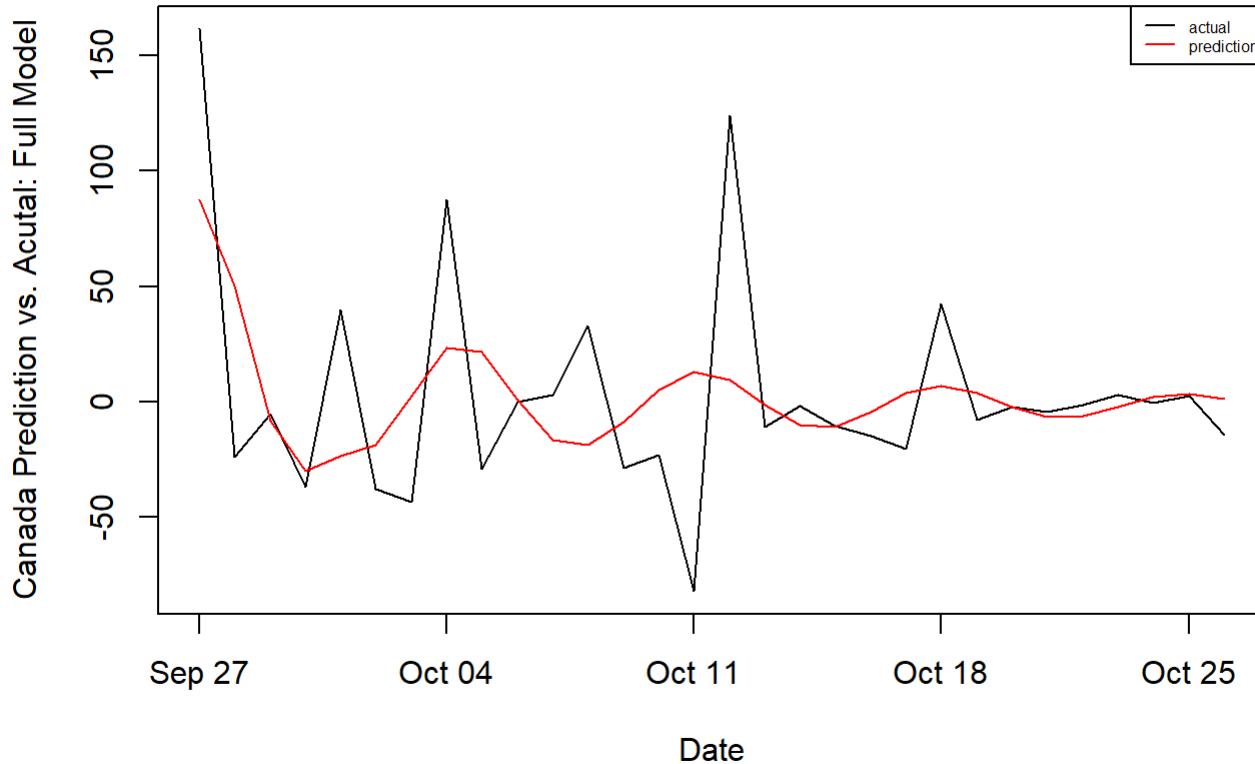
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_covid_canada$new_ts_covid_canada[1:403])
colnames(compare_table) = c("predict","actual")

plot(new_covid_canada$Day[1:403],  
      new_covid_canada$new_ts_covid_canada[1:403],  
      type = "l",  
      xlab = "Date",  
      ylab = "Canada Prediction vs. Acutal: Full Model")  
lines(new_covid_canada$Day[1:403],  
      compare_table$predict,  
      type = "l",  
      col = 'red')  
legend(x = "topright",legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_covid_canada$Day[374:403],                                     # Draw first time series
     new_covid_canada$new_ts_covid_canada[374:403],
     type = "l",
     xlab = "Date",
     ylab = "Canada Prediction vs. Acutal: Full Model")
lines(new_covid_canada$Day[374:403],                                     # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Canada Full Method 2
```

```
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 316.2185
```

```
#Testing MSE
```

```
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 1772.211
```

```
#USA example aggregate weekly data
```

```
ts_covid_usa = ts(covid_usa$residuals,frequency = 7)
ts_covid_usa_2 = ts(colSums(matrix(ts_covid_usa, nrow=7)))
```

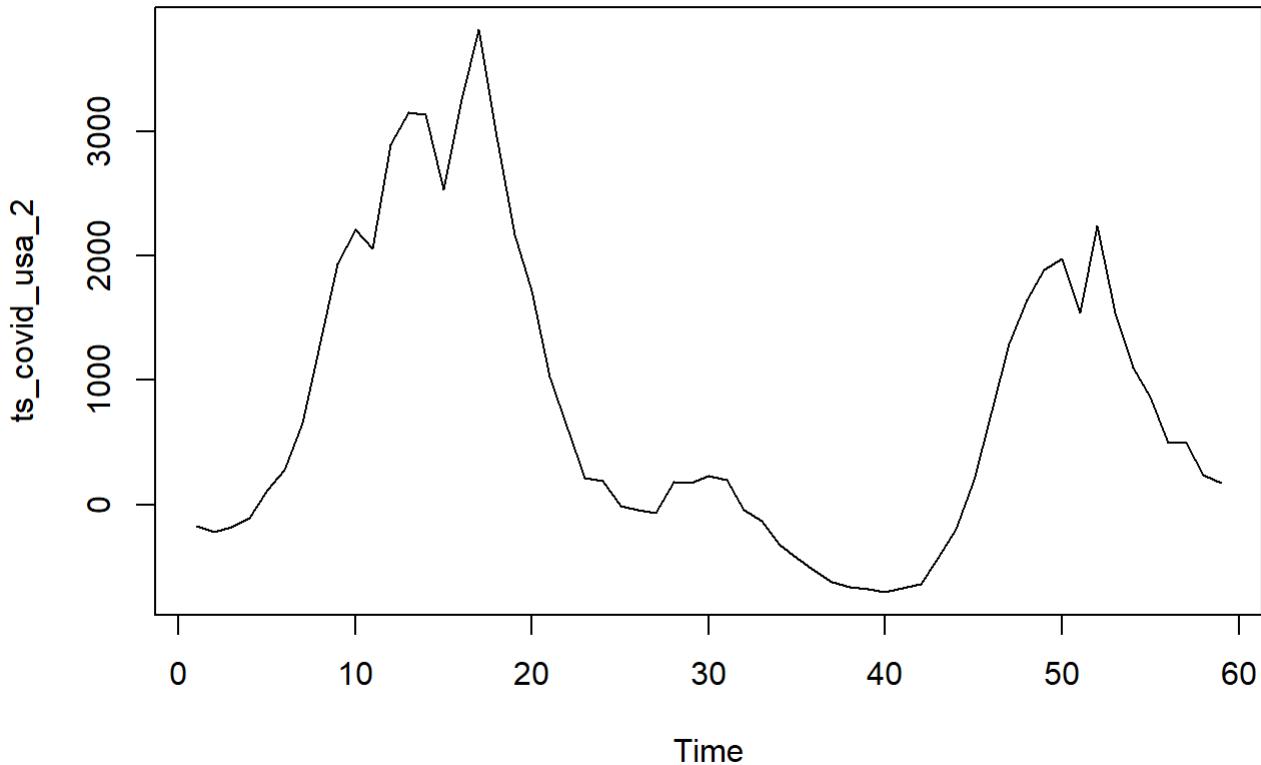
```
## Warning in matrix(ts_covid_usa, nrow = 7): data length [409] is not a sub-
## multiple or multiple of the number of rows [7]
```

```
vaccine_policy = colSums(matrix(as.numeric(covid_usa$vaccination_policy), nrow=7))

## Warning in matrix(as.numeric(covid_usa$vaccination_policy), nrow = 7): data
## length [409] is not a sub-multiple or multiple of the number of rows [7]

new_ts_covid_usa = data.frame(1:59,ts_covid_usa_2,vaccine_policy)
names(new_ts_covid_usa) = c("week","weekly_aggregated_residuals","vaccination_policy")

plot.ts(ts_covid_usa_2)
```



```
regression_model = lm(weekly_aggregated_residuals~1, data = new_ts_covid_usa[1:53,])
summary(regression_model)
```

```
##  
## Call:  
## lm(formula = weekly_aggregated_residuals ~ 1, data = new_ts_covid_usa[1:53,  
##     ])  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max  
## -1520.6  -990.7  -602.4  1074.1  3004.3  
##  
## Coefficients:  
##             Estimate Std. Error t value Pr(>|t|)  
## (Intercept)  814.6     174.1    4.679  2.1e-05 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 1268 on 52 degrees of freedom
```

```
anova(regression_model)
```

```
## Analysis of Variance Table  
##  
## Response: weekly_aggregated_residuals  
##              Df  Sum Sq Mean Sq F value Pr(>F)  
## Residuals  52 83561096 1606944
```

```
error = residuals(regression_model)  
ts_error = ts(error)
```

```
#ARIMA on error
```

```
error_model = auto.arima(ts_error)  
fitted_error = fitted(error_model)  
summary(error_model)
```

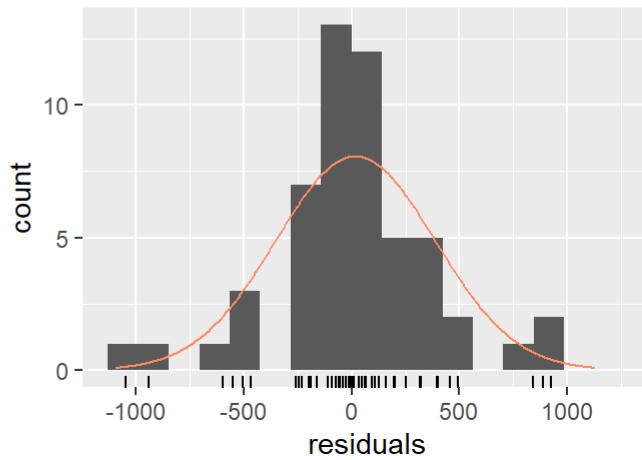
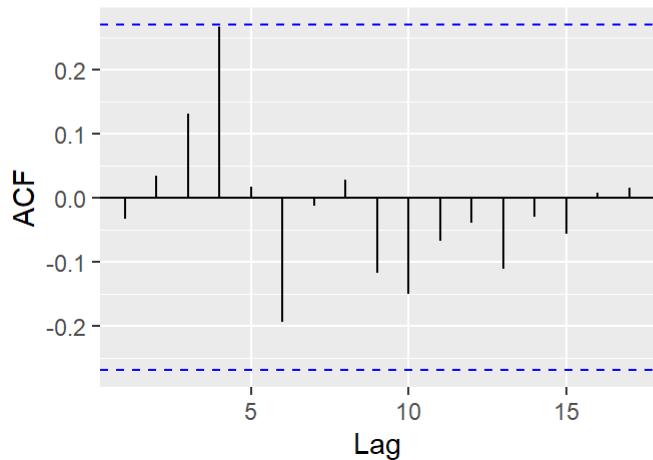
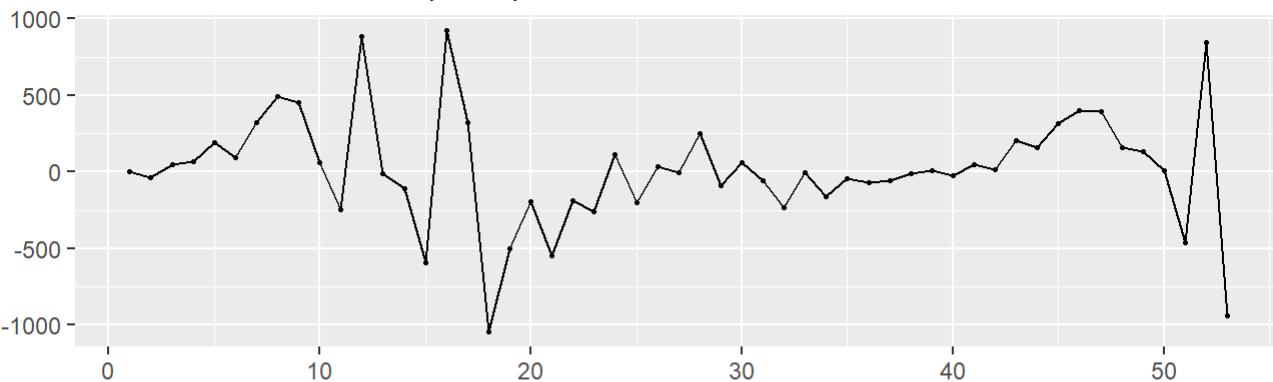
```

## Series: ts_error
## ARIMA(1,1,0)
##
## Coefficients:
##      ar1
##      0.3296
## s.e.  0.1337
##
## sigma^2 estimated as 139643:  log likelihood=-381.36
## AIC=766.71   AICc=766.96   BIC=770.61
##
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 17.33109 366.5697 247.2428 -15.98298 40.80275 0.8405701
##          ACF1
## Training set -0.03310144

```

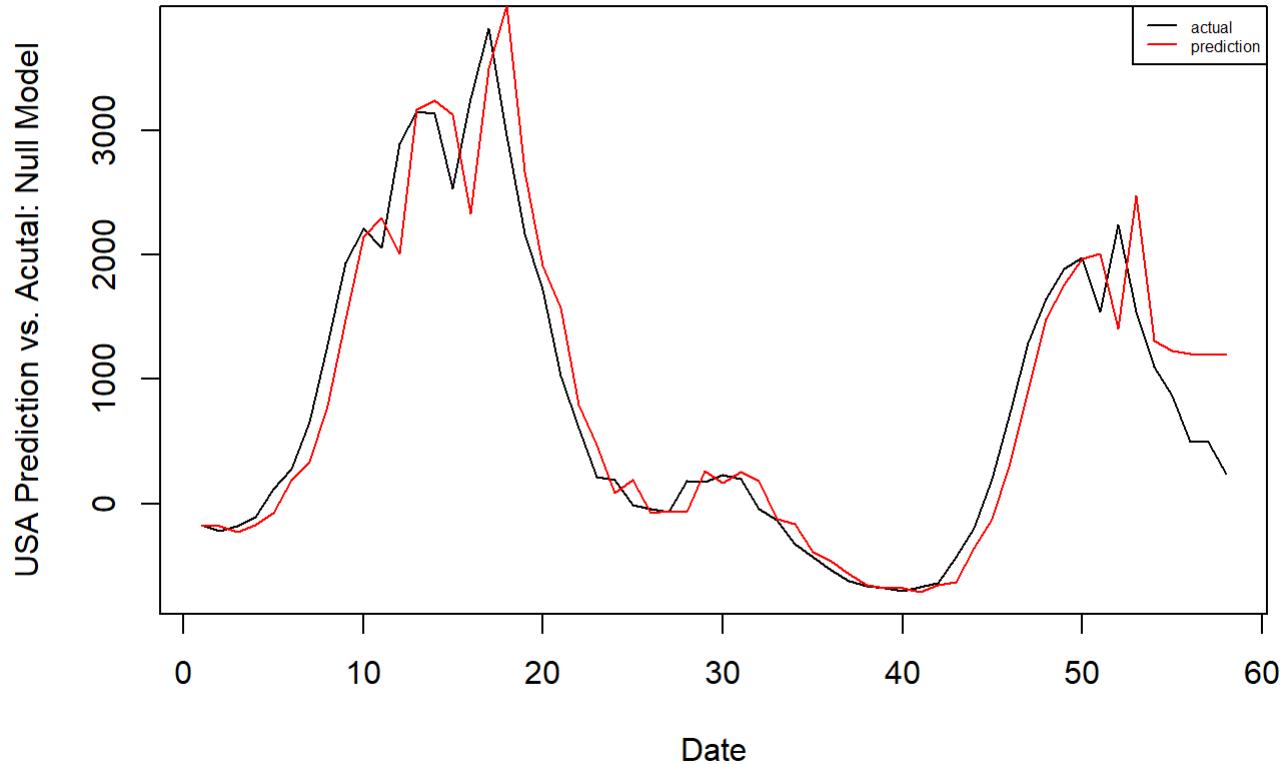
```
checkresiduals(error_model)
```

Residuals from ARIMA(1,1,0)

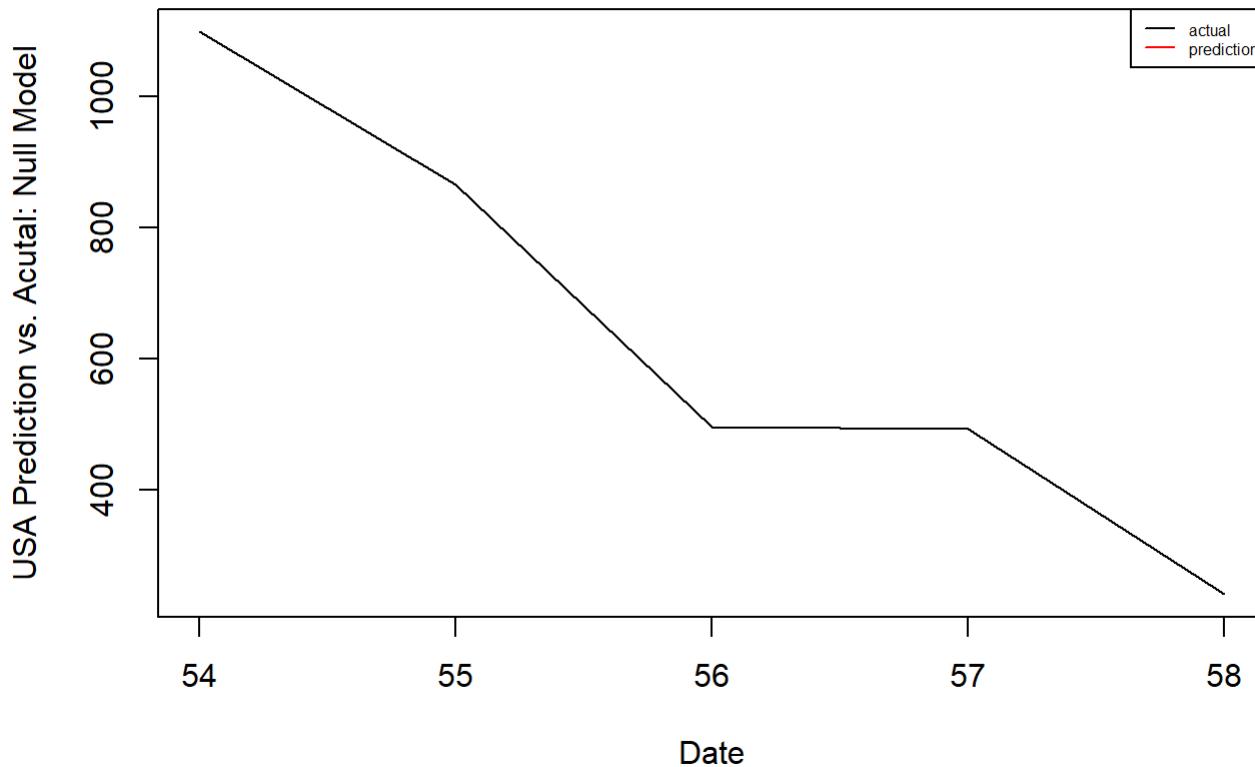


```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(1,1,0)  
## Q* = 10.165, df = 9, p-value = 0.3373  
##  
## Model df: 1. Total lags used: 10
```

```
#goodness of fit  
regression_prediction = predict(regression_model)  
  
final_prediction = regression_prediction + fitted_error  
  
regression_prediction = predict(regression_model, newdata = new_ts_covid_usa[54:58,])  
error_forecast = predict(error_model, n.ahead = 5)$pred  
  
final_prediction_2 = regression_prediction + error_forecast  
  
#compare prediction and actual  
  
compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_usa$weekly_aggregated_residuals[1:58])  
colnames(compare_table) = c("predict","actual")  
  
plot(new_ts_covid_usa$week[1:58],  
      new_ts_covid_usa$weekly_aggregated_residuals[1:58],  
      type = "l",  
      xlab = "Date",  
      ylab = "USA Prediction vs. Actual: Null Model")  
lines(new_ts_covid_usa$week[1:58],  
      compare_table$predict,  
      type = "l",  
      col = 'red')  
legend(x = "topright",legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_usa$week[54:58],                                     # Draw first time series
     new_ts_covid_usa$weekly_aggregated_residuals[54:58],
     type = "l",
     xlab = "Date",
     ylab = "USA Prediction vs. Acutal: Null Model")
lines(new_ts_covid_usa$week[54:58],                                       # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE USA Null Method 3
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 134373.4
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 416172
```

```
regression_model = lm(weekly_aggregated_residuals~vaccination_policy, data = new_ts_covid_usa[1:53,])
summary(regression_model)
```

```
##
## Call:
## lm(formula = weekly_aggregated_residuals ~ vaccination_policy,
##     data = new_ts_covid_usa[1:53, ])
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -1603.3 -945.1 -545.3 1019.4 2654.7 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 1607.61    321.97   4.993 7.33e-06 ***
## vaccination_policy -31.67     11.08  -2.857  0.00617 ** 
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
##
## Residual standard error: 1188 on 51 degrees of freedom
## Multiple R-squared:  0.138, Adjusted R-squared:  0.1211 
## F-statistic: 8.164 on 1 and 51 DF,  p-value: 0.00617
```

```
anova(regression_model)
```

```
## Analysis of Variance Table
##
## Response: weekly_aggregated_residuals
##              Df  Sum Sq Mean Sq F value Pr(>F)    
## vaccination_policy 1 11530653 11530653 8.1641 0.00617 ** 
## Residuals         51 72030443 1412362                        
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
error = residuals(regression_model)
ts_error = ts(error)
```

```
#ARIMA on error
```

```
error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)
```

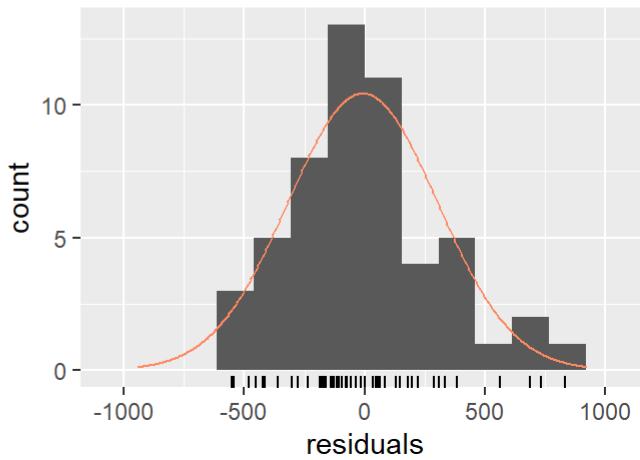
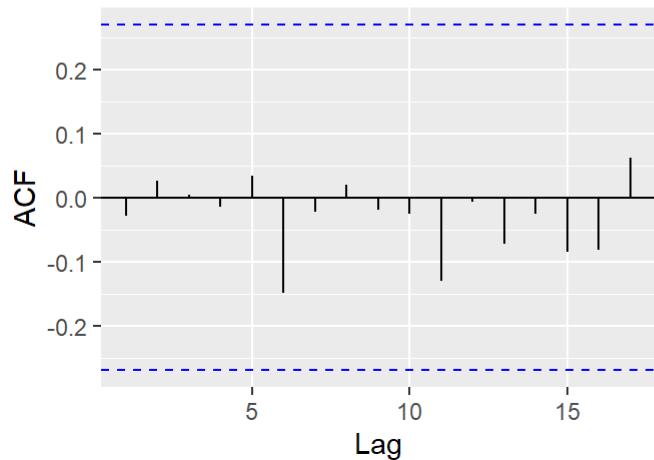
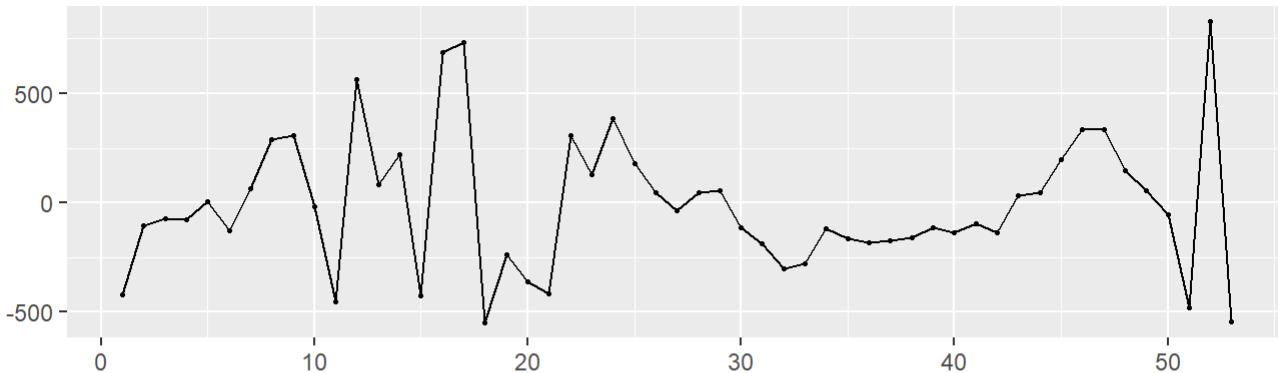
```

## Series: ts_error
## ARIMA(5,0,0) with zero mean
##
## Coefficients:
##      ar1     ar2     ar3     ar4     ar5
##      1.0398 -0.0670  0.0441  0.2094 -0.4158
##  s.e.  0.1296  0.2149  0.2323  0.2335  0.1419
##
## sigma^2 estimated as 104841:  log likelihood=-380.88
## AIC=773.76  AICc=775.58  BIC=785.58
##
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -8.562177 308.1402 237.8946 0.4607077 37.35291 0.8159201
##          ACF1
## Training set -0.02774625

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(5,0,0) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(5,0,0) with zero mean  
## Q* = 1.6604, df = 5, p-value = 0.8939  
##  
## Model df: 5. Total lags used: 10
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

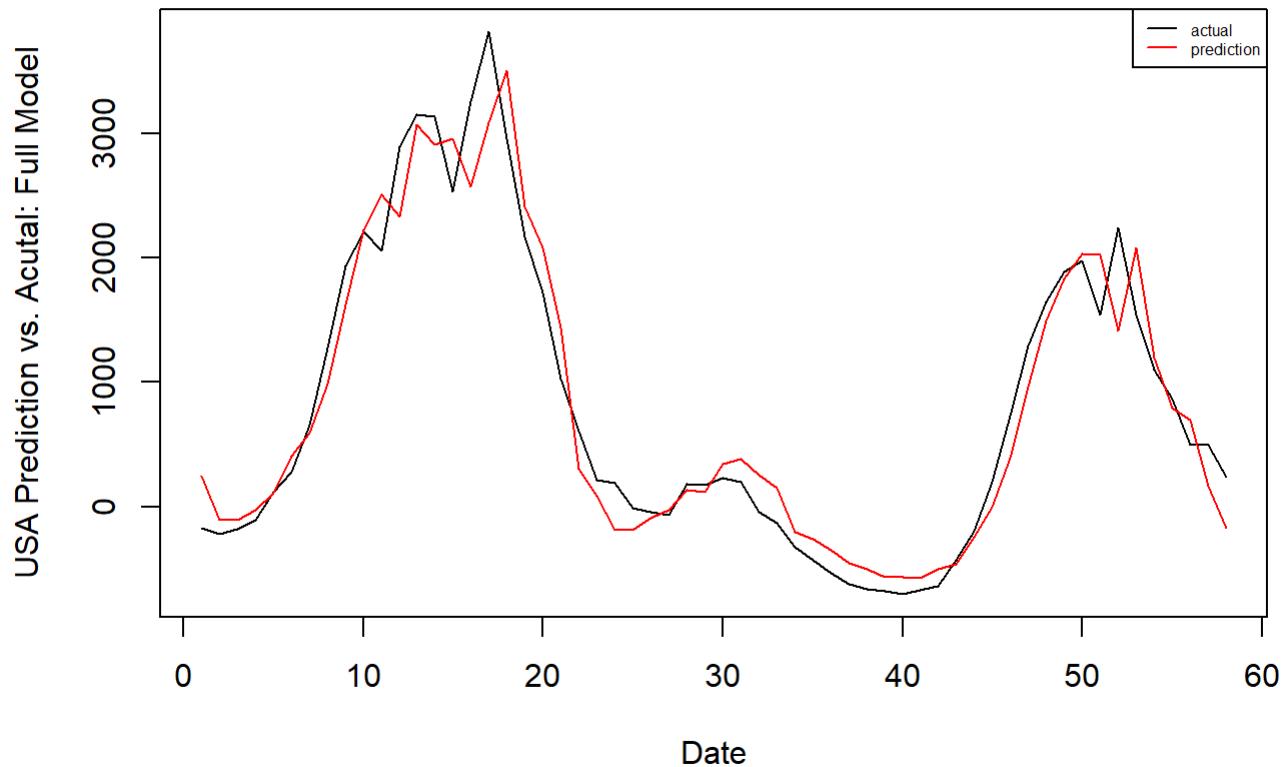
regression_prediction = predict(regression_model, newdata = new_ts_covid_usa[54:58,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

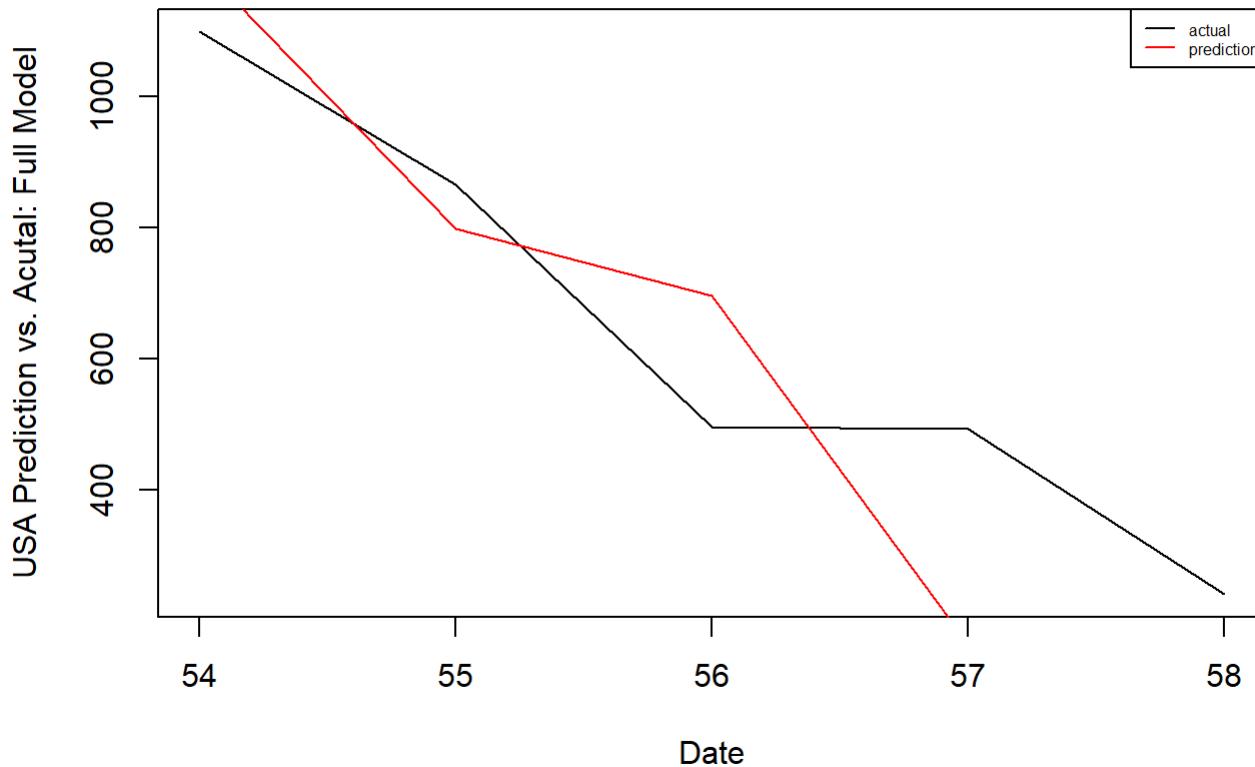
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_usa$weekly_aggregated_residuals[1:58])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_usa$week[1:58],                                     # Draw first time series
      new_ts_covid_usa$weekly_aggregated_residuals[1:58],
      type = "l",
      xlab = "Date",
      ylab = "USA Prediction vs. Actual: Full Model")
lines(new_ts_covid_usa$week[1:58],                                     # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_usa$week[54:58],                                     # Draw first time series
     new_ts_covid_usa$weekly_aggregated_residuals[54:58],
     type = "l",
     xlab = "Date",
     ylab = "USA Prediction vs. Acutal: Full Model")
lines(new_ts_covid_usa$week[54:58],                                       # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE USA Full Method
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 94950.39
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 67544.61
```

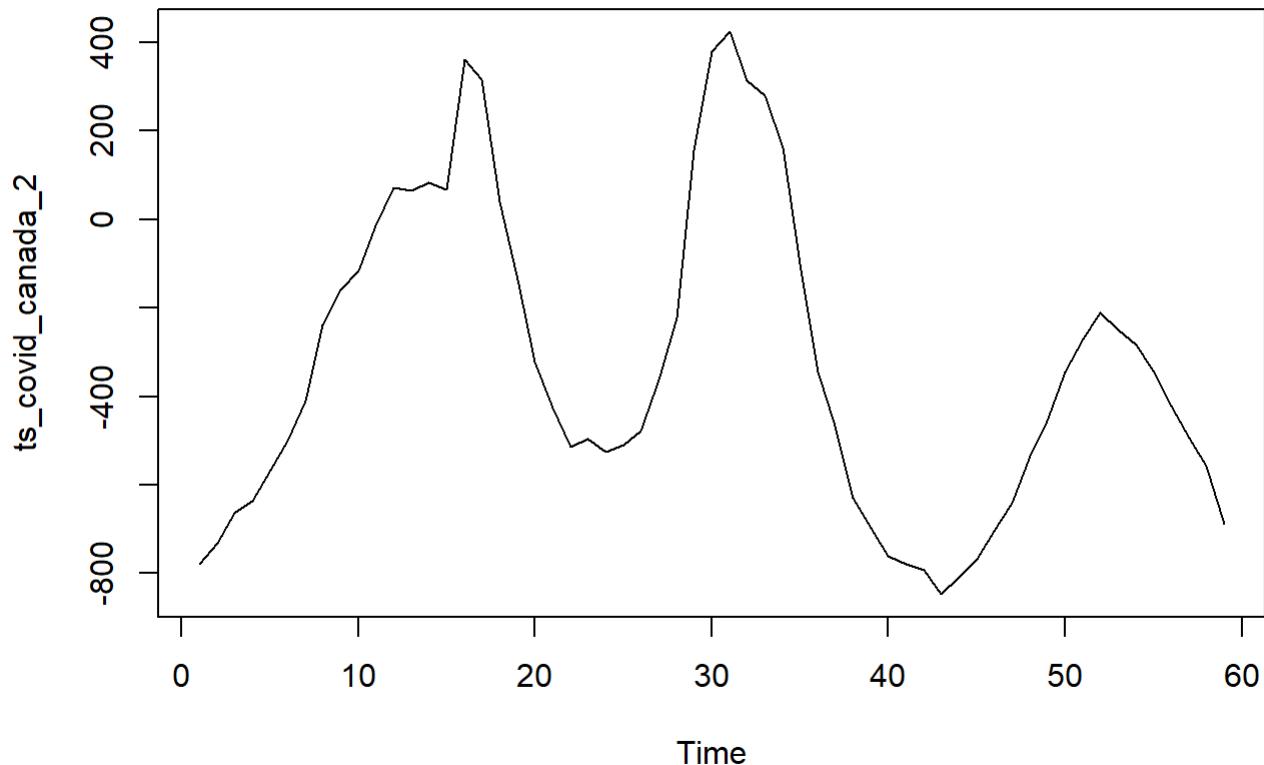
```
#Canada Example on aggregate weekly
ts_covid_canada = ts(covid_canada$residuals,frequency = 7)
ts_covid_canada_2 = ts(colSums(matrix(ts_covid_canada, nrow=7)))
```

```
## Warning in matrix(ts_covid_canada, nrow = 7): data length [409] is not a sub-
## multiple or multiple of the number of rows [7]
```

```
vaccine_policy = colSums(matrix(as.numeric(covid_canada$vaccination_policy), nrow=7))
```

```
## Warning in matrix(as.numeric(covid_canada$vaccination_policy), nrow = 7): data  
## length [409] is not a sub-multiple or multiple of the number of rows [7]
```

```
new_ts_covid_canada = data.frame(1:59,ts_covid_canada_2,vaccine_policy)  
names(new_ts_covid_canada) = c("week","weekly_aggregated_residuals","vaccination_policy")  
  
plot.ts(ts_covid_canada_2)
```



```
regression_model = lm(weekly_aggregated_residuals~1, data = new_ts_covid_canada[1:53,])  
summary(regression_model)
```

```
##  
## Call:  
## lm(formula = weekly_aggregated_residuals ~ 1, data = new_ts_covid_canada[1:53,  
##     ])  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max  
## -536.01 -317.48 -50.82 296.18 735.08  
##  
## Coefficients:  
##             Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -311.22      50.24  -6.194 9.49e-08 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 365.8 on 52 degrees of freedom
```

```
anova(regression_model)
```

```
## Analysis of Variance Table  
##  
## Response: weekly_aggregated_residuals  
##              Df  Sum Sq Mean Sq F value Pr(>F)  
## Residuals  52 6957325 133795
```

```
error = residuals(regression_model)  
ts_error = ts(error)
```

```
#ARIMA on error
```

```
error_model = auto.arima(ts_error)  
fitted_error = fitted(error_model)  
summary(error_model)
```

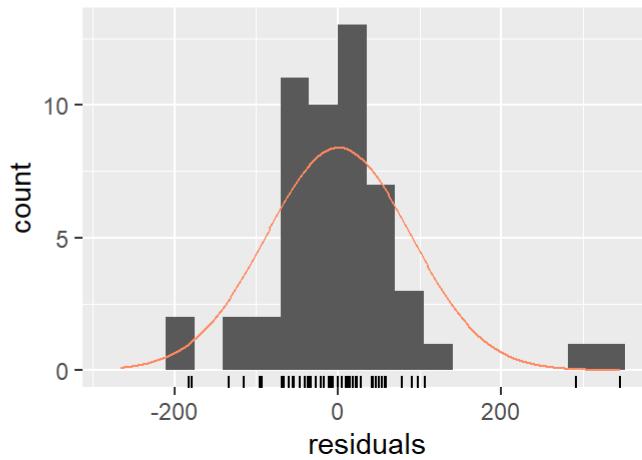
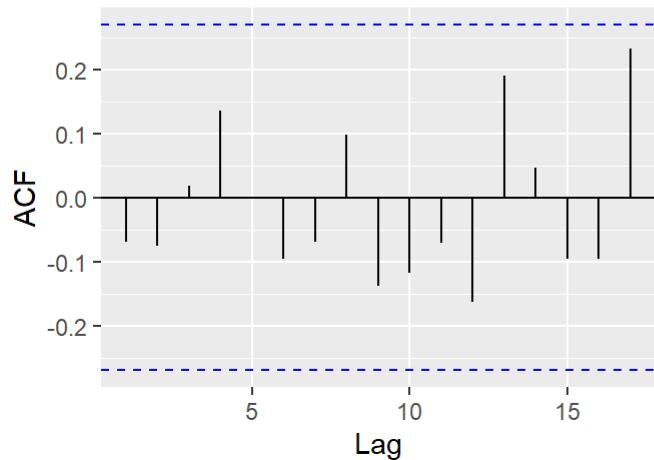
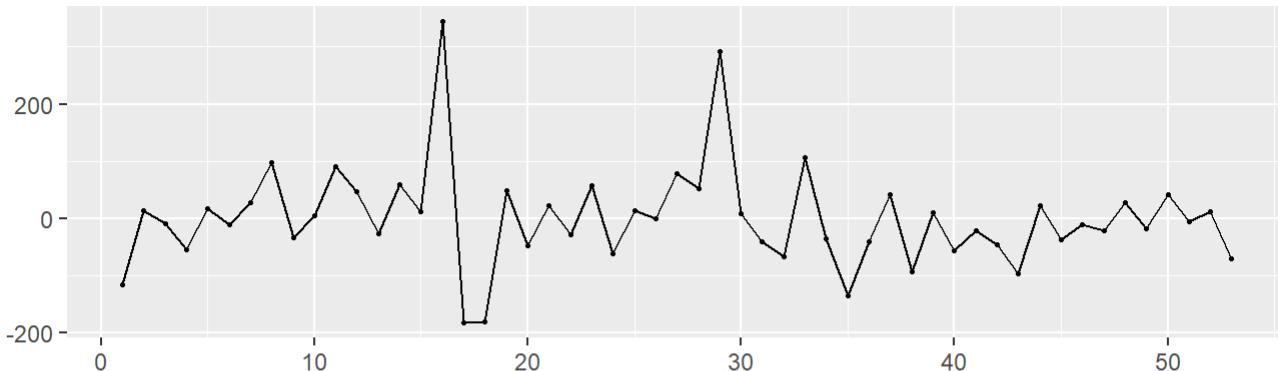
```

## Series: ts_error
## ARIMA(2,0,0) with zero mean
##
## Coefficients:
##      ar1     ar2
##      1.5896 -0.6911
##  s.e.  0.0939  0.0946
##
## sigma^2 estimated as 8012: log likelihood=-314.11
## AIC=634.22   AICc=634.71   BIC=640.13
##
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.5595646 87.80656 58.35176 8.042255 36.16421 0.6064867
##          ACF1
## Training set -0.06829179

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(2,0,0) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,0,0) with zero mean  
## Q* = 5.3701, df = 8, p-value = 0.7174  
##  
## Model df: 2. Total lags used: 10
```

#goodness of fit

```
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

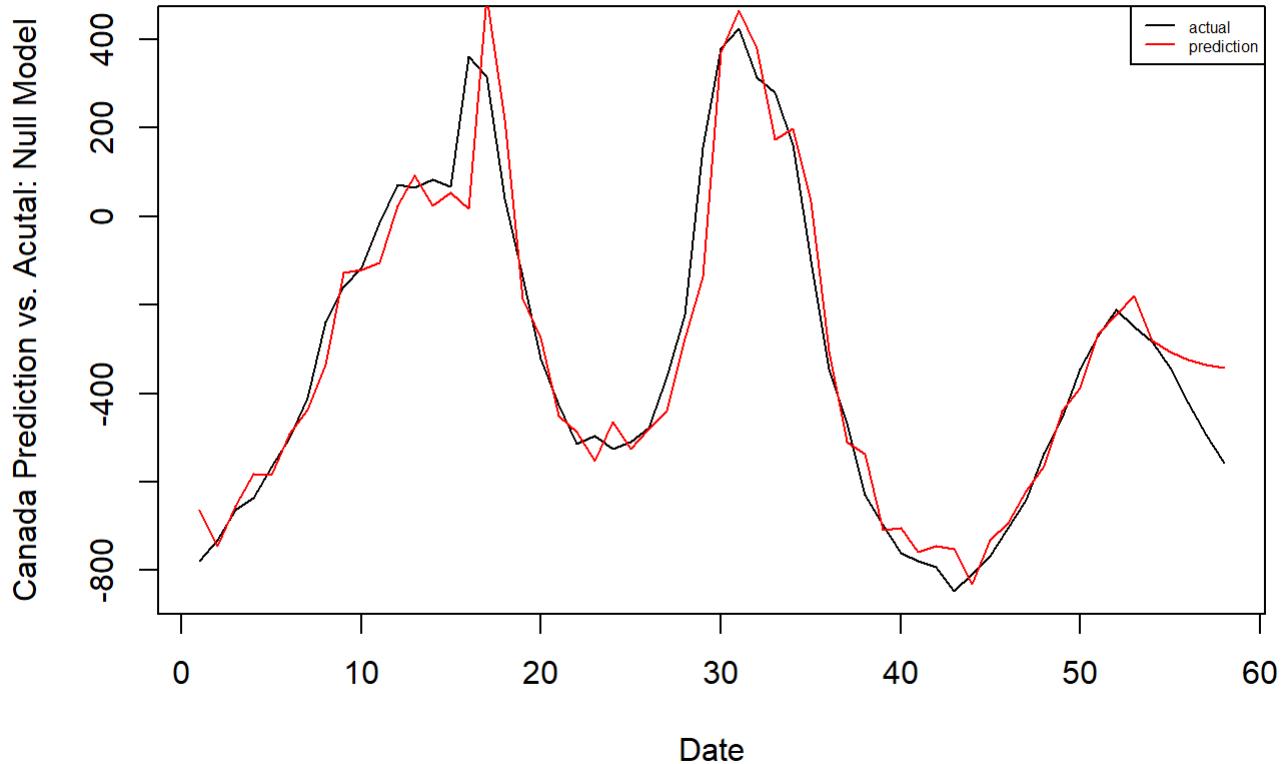
regression_prediction = predict(regression_model, newdata = new_ts_covid_canada[54:58,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast
```

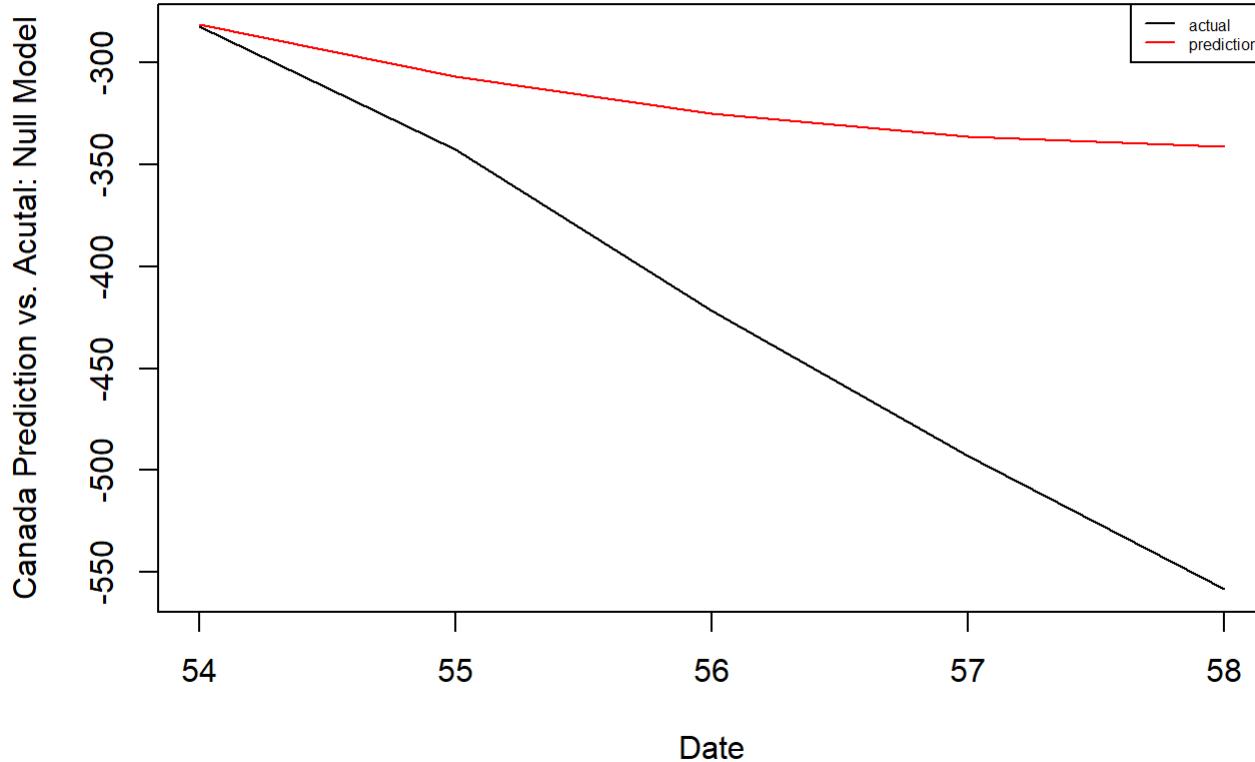
#compare prediction and actual

```
compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_canada$weekly_aggregated_residuals[1:58])
colnames(compare_table) = c("predict","actual")
```

```
plot(new_ts_covid_canada$week[1:58],  
     new_ts_covid_canada$weekly_aggregated_residuals[1:58],  
     type = "l",  
     xlab = "Date",  
     ylab = "Canada Prediction vs. Actual: Null Model")  
lines(new_ts_covid_canada$week[1:58],  
      compare_table$predict,  
      type = "l",  
      col = 'red')  
legend(x = "topright",legend=c("actual", "prediction"),  
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_canada$week[54:58],                                     # Draw first time series
     new_ts_covid_canada$weekly_aggregated_residuals[54:58],
     type = "l",
     xlab = "Date",
     ylab = "Canada Prediction vs. Acutal: Null Model")
lines(new_ts_covid_canada$week[54:58],                                     # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Canada Null method 3
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 7709.992
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 16462.48
```

```
regression_model = lm(weekly_aggregated_residuals~vaccination_policy, data = new_ts_covid_canada[1:53,])
summary(regression_model)
```

```
##  
## Call:  
## lm(formula = weekly_aggregated_residuals ~ vaccination_policy,  
##      data = new_ts_covid_canada[1:53, ])  
##  
## Residuals:  
##     Min      1Q   Median      3Q     Max  
## -613.53 -272.96 -41.36 234.26 744.55  
##  
## Coefficients:  
##                         Estimate Std. Error t value Pr(>|t|)  
## (Intercept)       -114.169    111.601 -1.023  0.3111  
## vaccination_policy -7.375      3.755 -1.964  0.0549 .  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 356.1 on 51 degrees of freedom  
## Multiple R-squared:  0.07034,   Adjusted R-squared:  0.05211  
## F-statistic: 3.859 on 1 and 51 DF,  p-value: 0.05494
```

```
anova(regression_model)
```

```
## Analysis of Variance Table  
##  
## Response: weekly_aggregated_residuals  
##                         Df  Sum Sq Mean Sq F value Pr(>F)  
## vaccination_policy  1 489403 489403  3.859 0.05494 .  
## Residuals          51 6467922 126822  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
error = residuals(regression_model)  
ts_error = ts(error)  
  
#ARIMA on error  
  
error_model = auto.arima(ts_error)  
fitted_error = fitted(error_model)  
summary(error_model)
```

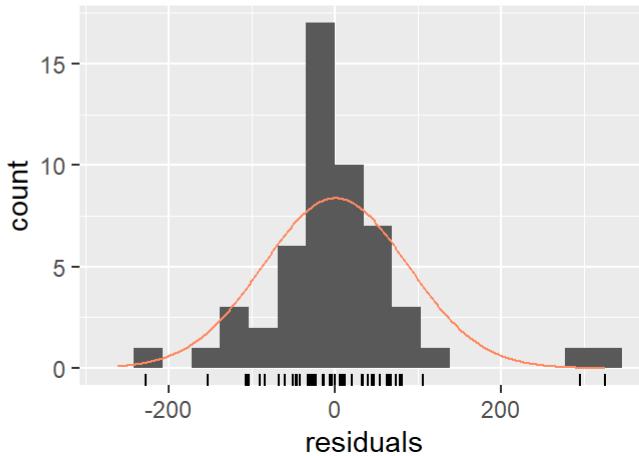
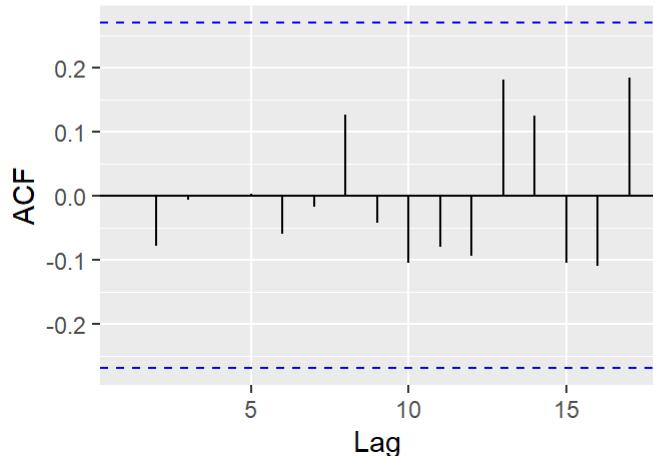
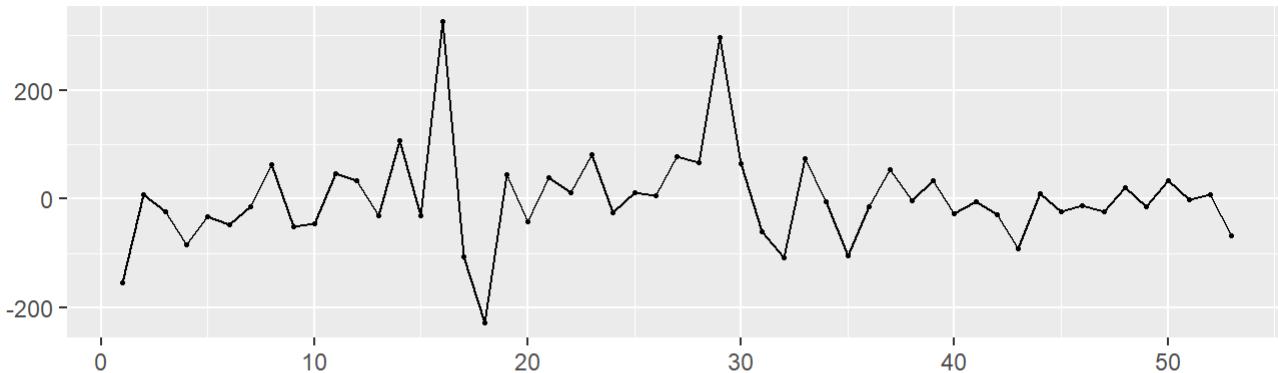
```

## Series: ts_error
## ARIMA(2,0,1) with zero mean
##
## Coefficients:
##             ar1      ar2      ma1
##            1.7816 -0.8713 -0.4343
## s.e.    0.0925  0.0884  0.1986
## 
## sigma^2 estimated as 7954:  log likelihood=-313.44
## AIC=634.88   AICc=635.71   BIC=642.76
## 
## Training set error measures:
##               ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 0.3590227 86.62385 56.88558 32.3087 95.9402 0.5988167 0.001339009

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(2,0,1) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,0,1) with zero mean  
## Q* = 2.4832, df = 7, p-value = 0.9284  
##  
## Model df: 3. Total lags used: 10
```

#goodness of fit

```
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

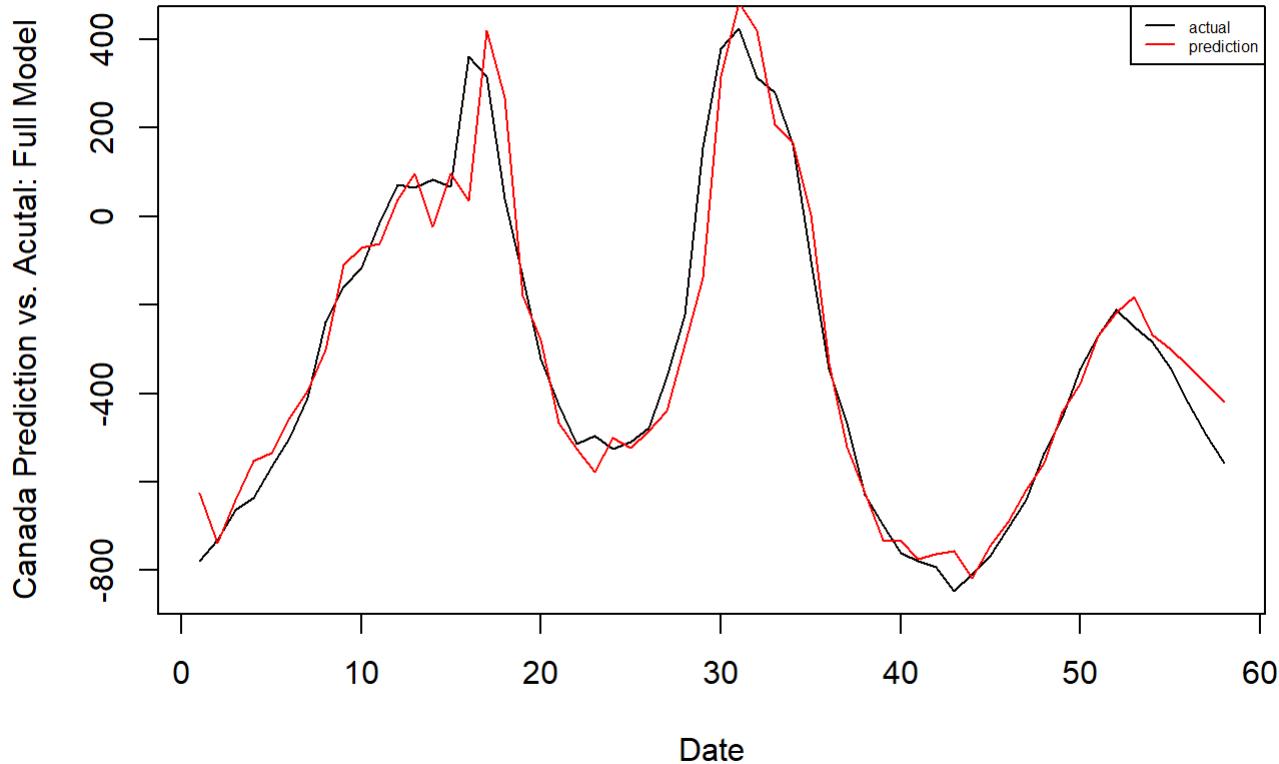
regression_prediction = predict(regression_model, newdata = new_ts_covid_canada[54:58,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast
```

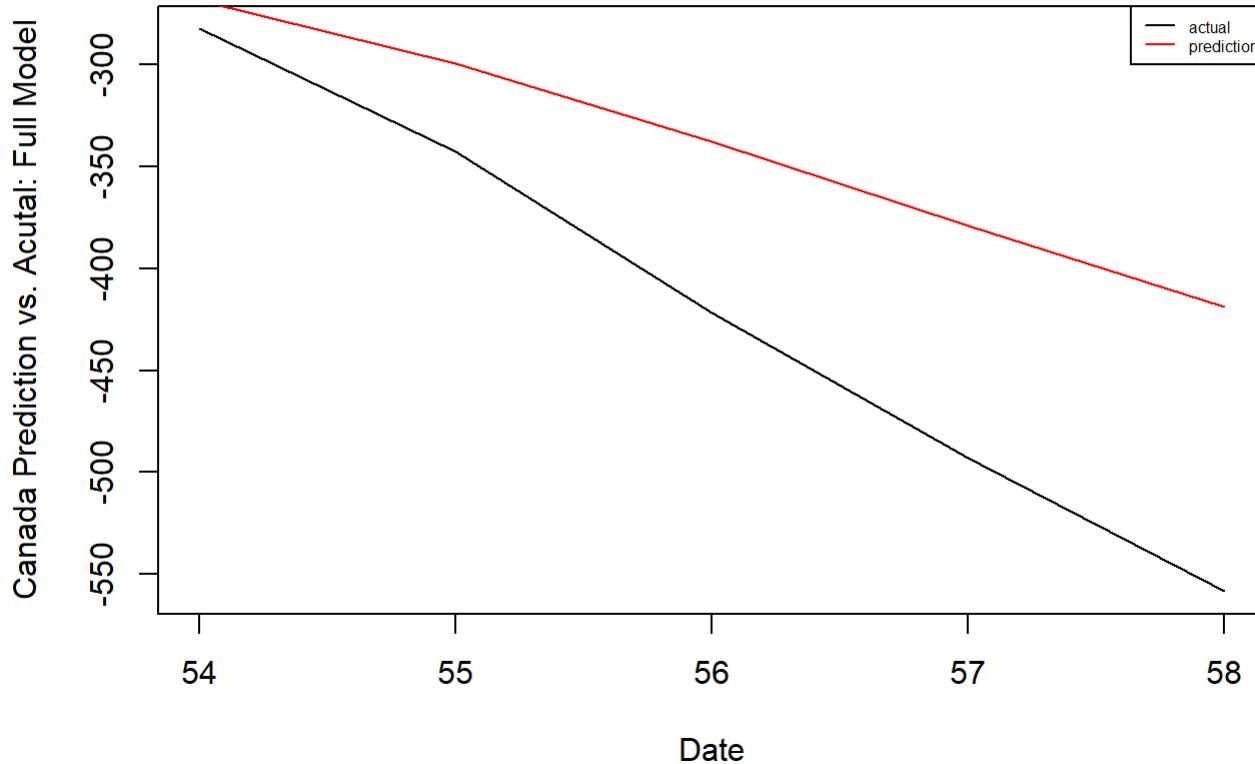
#compare prediction and actual

```
compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_canada$weekly_aggregated_residuals[1:58])
colnames(compare_table) = c("predict","actual")
```

```
plot(new_ts_covid_canada$week[1:58],  
     new_ts_covid_canada$weekly_aggregated_residuals[1:58],  
     type = "l",  
     xlab = "Date",  
     ylab = "Canada Prediction vs. Actual: Full Model")  
lines(new_ts_covid_canada$week[1:58],  
      compare_table$predict,  
      type = "l",  
      col = 'red')  
legend(x = "topright",legend=c("actual", "prediction"),  
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_canada$week[54:58],                                     # Draw first time series
     new_ts_covid_canada$weekly_aggregated_residuals[54:58],
     type = "l",
     xlab = "Date",
     ylab = "Canada Prediction vs. Acutal: Full Model")
lines(new_ts_covid_canada$week[54:58],                                     # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Canada Full Method 3
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 7503.692
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 8346.138
```

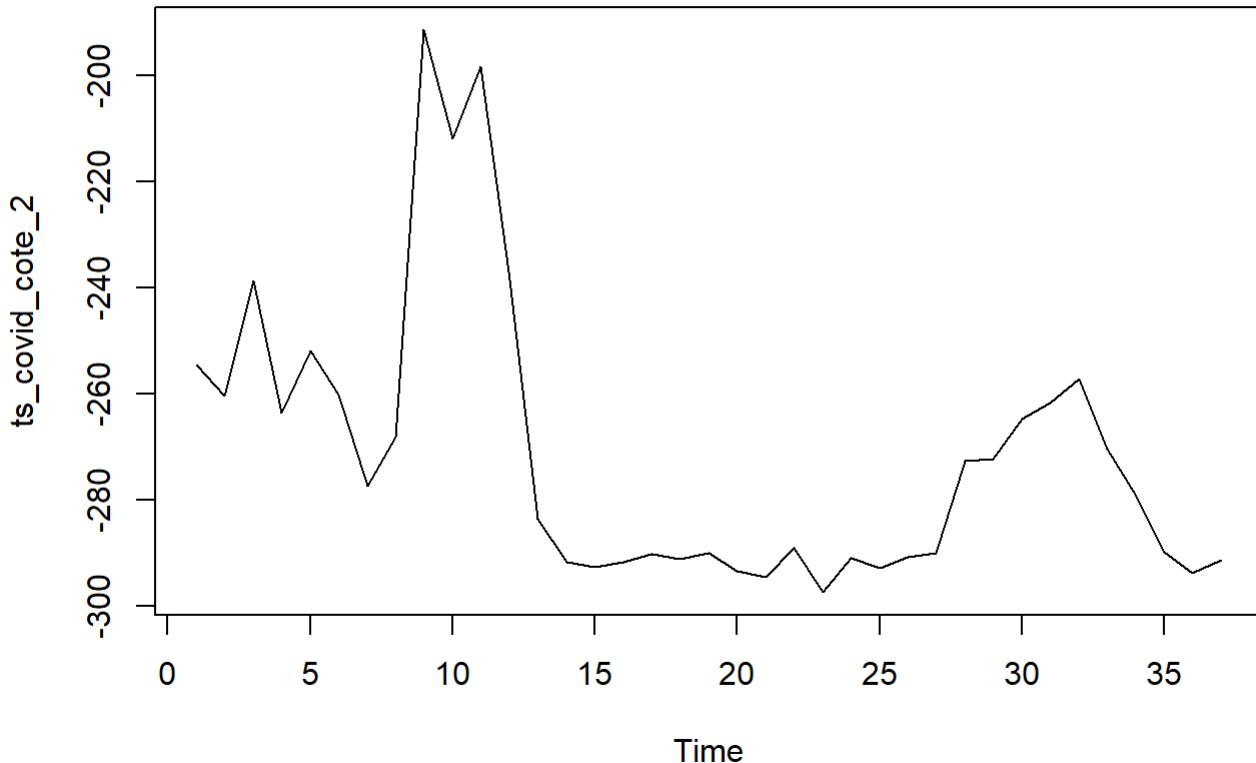
```
#Cote Example on aggregate weekly
ts_covid_cote = ts(covid_cote$residuals,frequency = 7)
ts_covid_cote_2 = ts(colSums(matrix(ts_covid_cote, nrow=7)))
```

```
## Warning in matrix(ts_covid_cote, nrow = 7): data length [258] is not a sub-
## multiple or multiple of the number of rows [7]
```

```
vaccine_policy = colSums(matrix(as.numeric(covid_cote$vaccination_policy), nrow=7))
```

```
## Warning in matrix(as.numeric(covid_cote$vaccination_policy), nrow = 7): data  
## length [258] is not a sub-multiple or multiple of the number of rows [7]
```

```
new_ts_covid_cote = data.frame(1:37,ts_covid_cote_2,vaccine_policy)  
names(new_ts_covid_cote) = c("week","weekly_aggregated_residuals","vaccination_policy")  
  
plot.ts(ts_covid_cote_2)
```



```
regression_model = lm(weekly_aggregated_residuals~1, data = new_ts_covid_cote[1:32,])  
summary(regression_model)
```

```

## 
## Call:
## lm(formula = weekly_aggregated_residuals ~ 1, data = new_ts_covid_cote[1:32,
##   ])
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -28.198 -21.796 -5.869  9.661 77.678 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -269.114     4.989  -53.94 <2e-16 ***
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 28.22 on 31 degrees of freedom

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: weekly_aggregated_residuals
##           Df Sum Sq Mean Sq F value Pr(>F)    
## Residuals 31  24690  796.44

```

```

error = residuals(regression_model)
ts_error = ts(error)

```

```

#ARIMA on error

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)

```

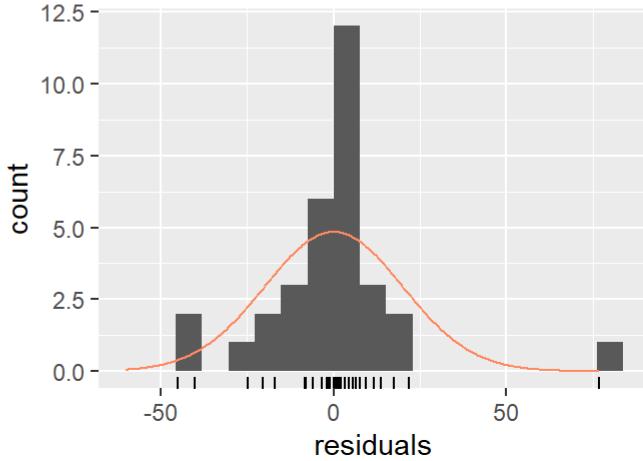
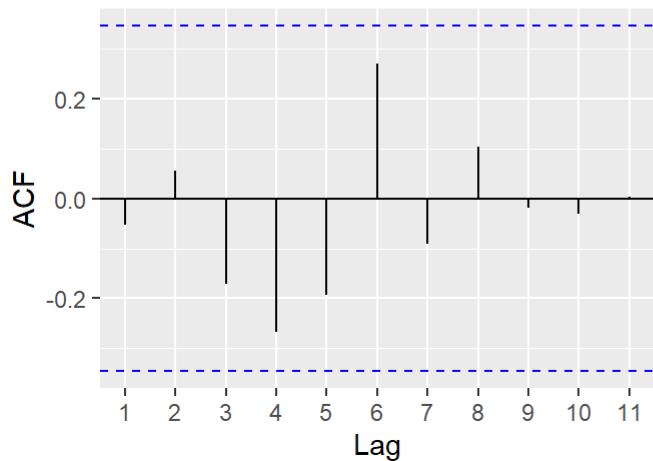
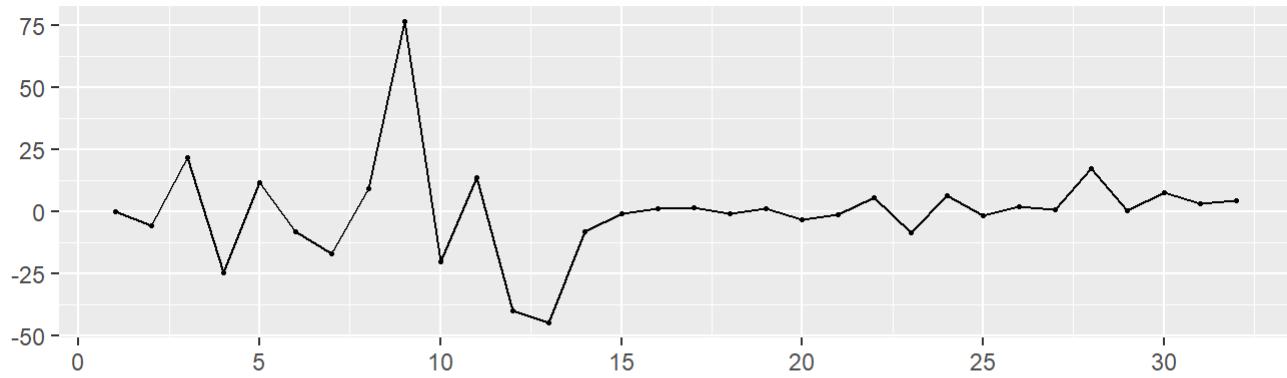
```

## Series: ts_error
## ARIMA(0,1,0)
## 
## sigma^2 estimated as 402.6:  log likelihood=-136.95
## AIC=275.91  AICc=276.05  BIC=277.34
## 
## Training set error measures:
##               ME      RMSE      MAE      MPE      MAPE      MASE      ACF1  
## Training set -0.0843873 19.74826 11.6149 22.33857 108.327 0.9687883 -0.05175717

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(0,1,0)



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(0,1,0)  
## Q* = 8.6505, df = 6, p-value = 0.1942  
##  
## Model df: 0. Total lags used: 6
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

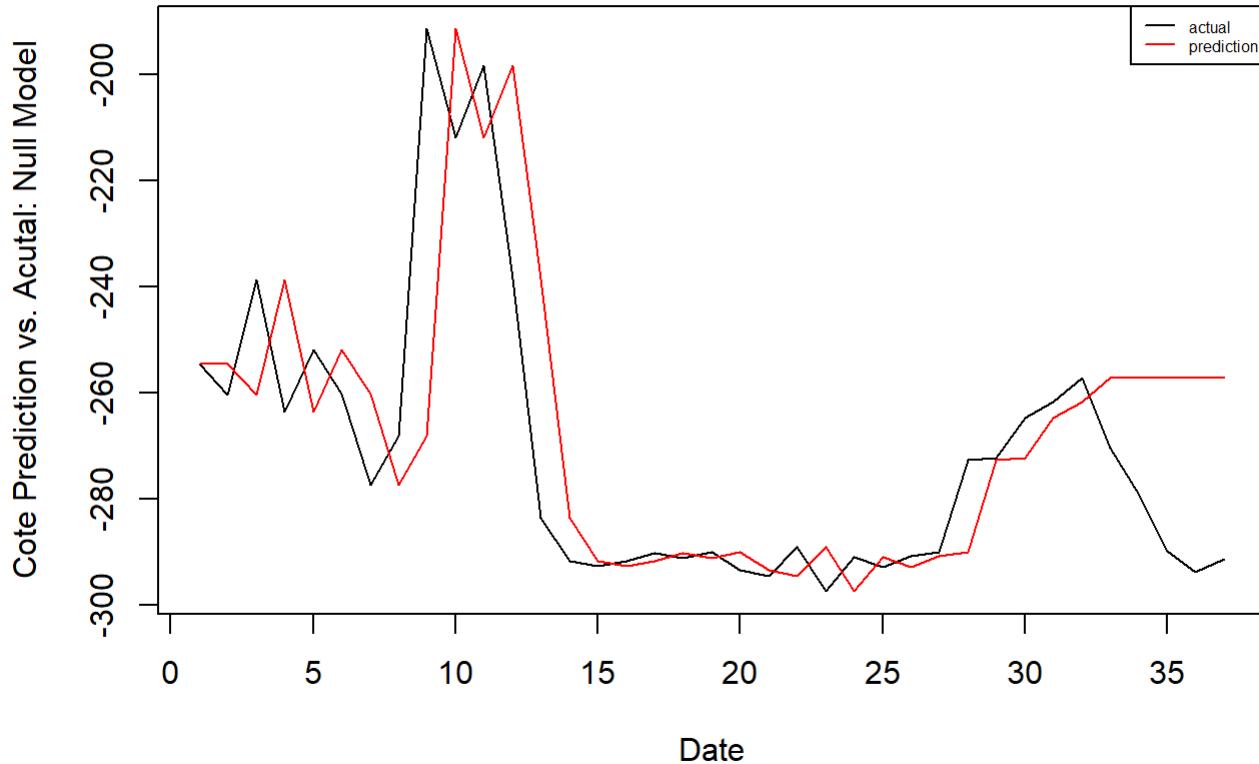
regression_prediction = predict(regression_model, newdata = new_ts_covid_cote[33:37,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

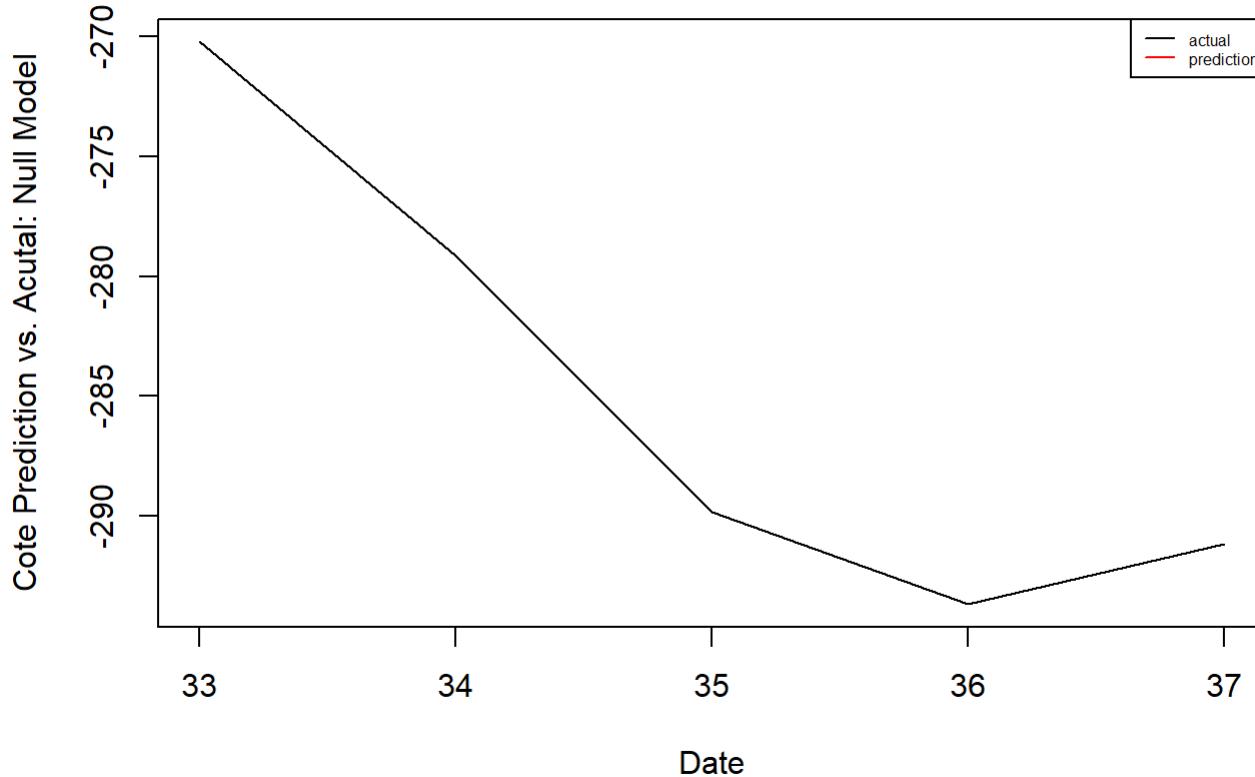
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_cote$weekly_aggregated_residuals[1:37])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_cote$week[1:37],                                     # Draw first time series
      new_ts_covid_cote$weekly_aggregated_residuals[1:37],
      type = "l",
      xlab = "Date",
      ylab = "Cote Prediction vs. Actual: Null Model")
lines(new_ts_covid_cote$week[1:37],                                       # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_cote$week[33:37],                                     # Draw first time series
     new_ts_covid_cote$weekly_aggregated_residuals[33:37],
     type = "l",
     xlab = "Date",
     ylab = "Cote Prediction vs. Acutal: Null Model")
lines(new_ts_covid_cote$week[33:37],                                     # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Cote Null
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 389.9936
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 842.3394
```

```
regression_model = lm(weekly_aggregated_residuals~vaccination_policy, data = new_ts_covid_cote[1:32,])
summary(regression_model)
```

```

## 
## Call:
## lm(formula = weekly_aggregated_residuals ~ vaccination_policy,
##     data = new_ts_covid_cote[1:32, ])
## 
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -34.846 -11.336 -8.605  9.634  65.799 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -234.6084    8.9112 -26.327 < 2e-16 ***
## vaccination_policy   -1.1313    0.2614  -4.328 0.000154 *** 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 22.51 on 30 degrees of freedom 
## Multiple R-squared:  0.3843, Adjusted R-squared:  0.3638 
## F-statistic: 18.73 on 1 and 30 DF,  p-value: 0.0001542

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: weekly_aggregated_residuals
##              Df Sum Sq Mean Sq F value    Pr(>F)    
## vaccination_policy  1  9489  9489.0 18.728 0.0001542 ***
## Residuals         30 15201   506.7 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

```

```

error = residuals(regression_model)
ts_error = ts(error)

#ARIMA on error

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)

```

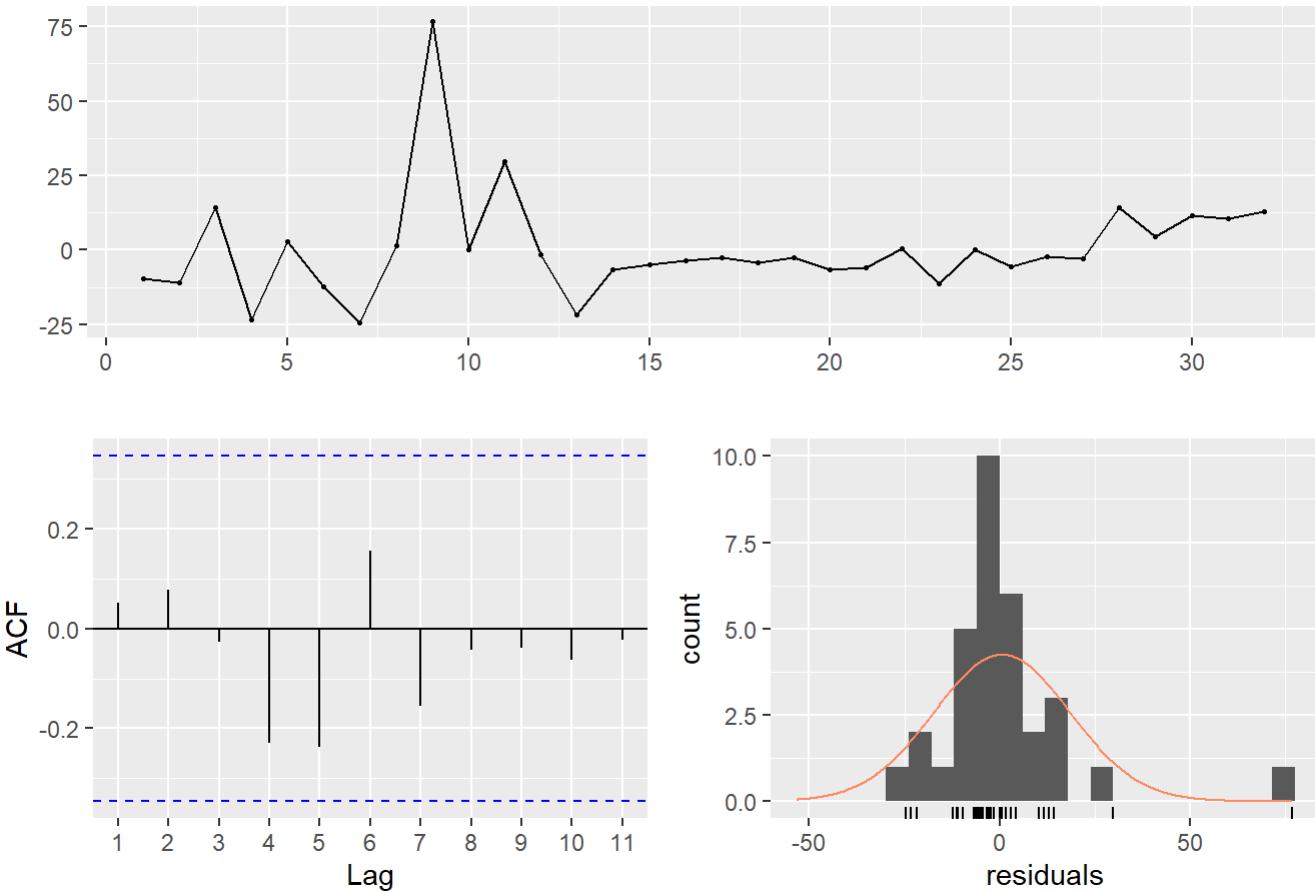
```

## Series: ts_error
## ARIMA(1,0,0) with zero mean
##
## Coefficients:
##      ar1
##      0.5835
## s.e.  0.1422
##
## sigma^2 estimated as 321.5: log likelihood=-137.47
## AIC=278.94   AICc=279.36   BIC=281.88
##
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 0.5253276 17.64688 10.69229 70.32338 73.6064 0.9169547 0.05085159

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(1,0,0) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(1,0,0) with zero mean  
## Q* = 5.6938, df = 5, p-value = 0.3372  
##  
## Model df: 1. Total lags used: 6
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

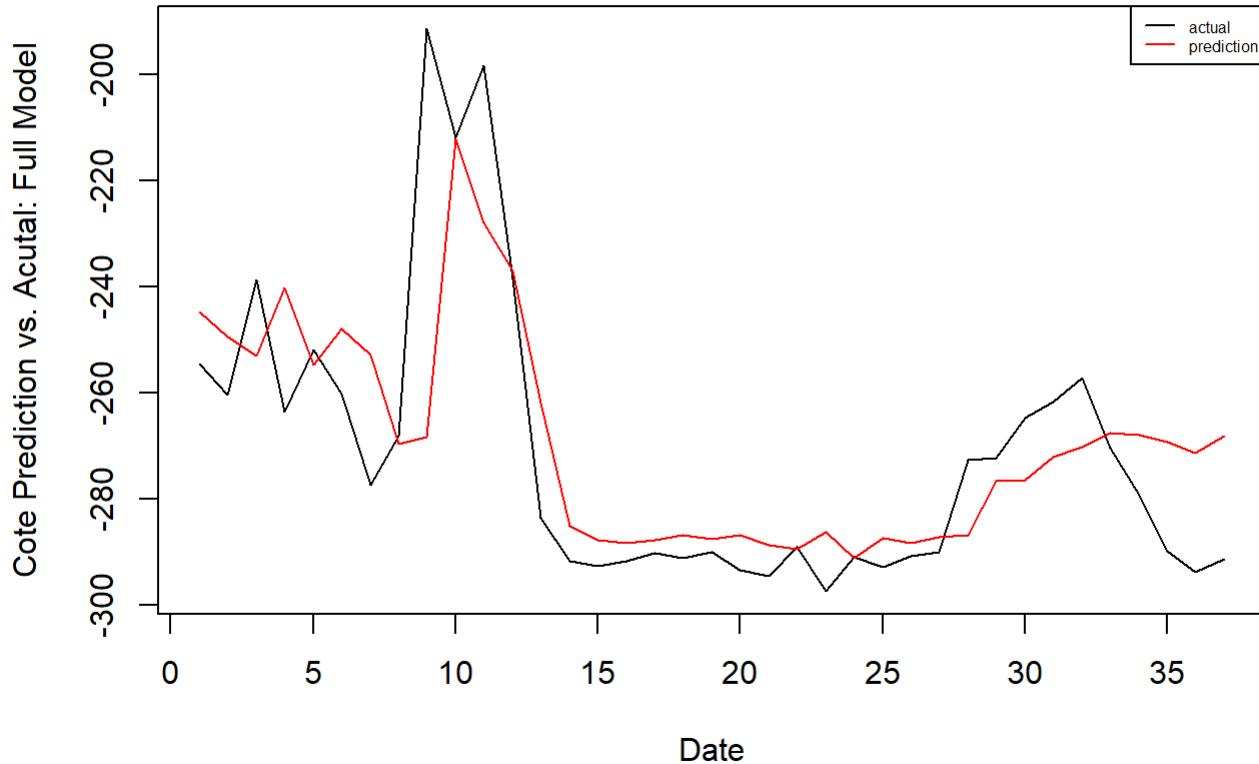
regression_prediction = predict(regression_model, newdata = new_ts_covid_cote[33:37,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

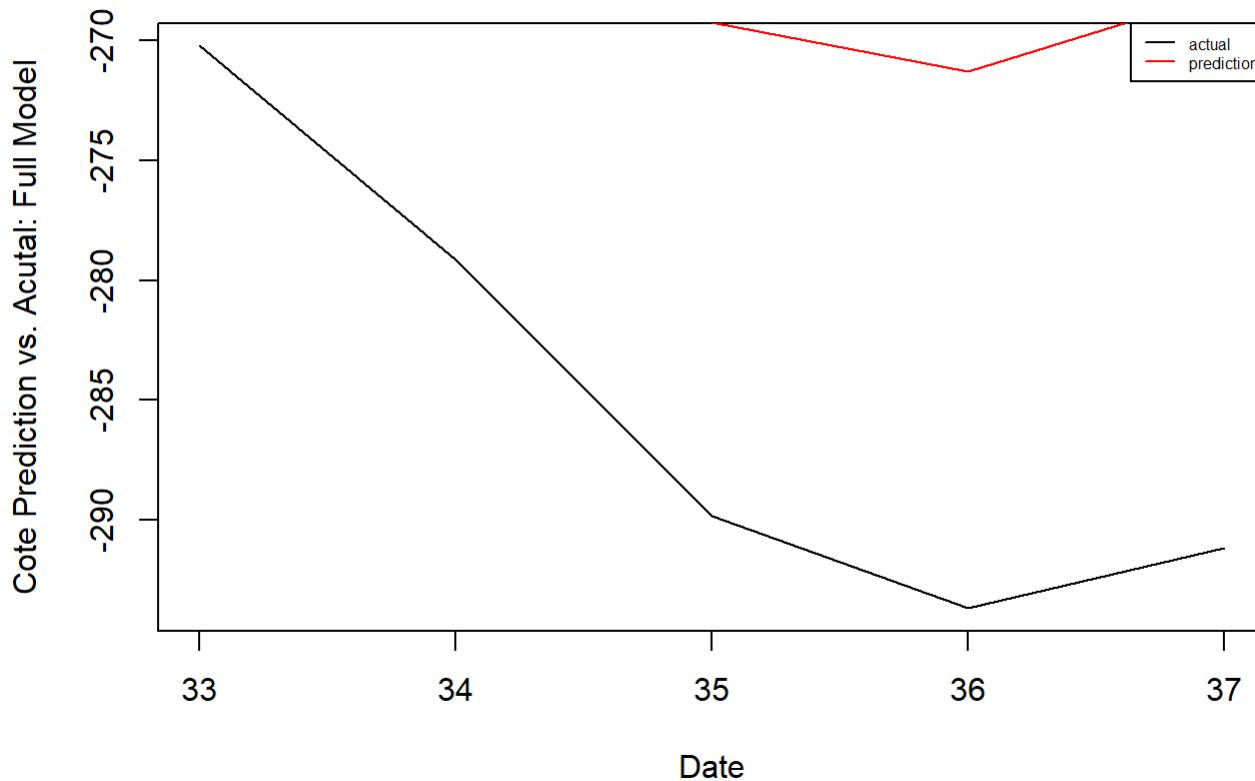
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_cote$weekly_aggregated_residuals[1:37])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_cote$week[1:37],  
      new_ts_covid_cote$weekly_aggregated_residuals[1:37],  
      type = "l",  
      xlab = "Date",  
      ylab = "Cote Prediction vs. Acutal: Full Model")  
lines(new_ts_covid_cote$week[1:37],  
      compare_table$predict,  
      type = "l",  
      col = 'red')  
legend(x = "topright",legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_cote$week[33:37],                                     # Draw first time series
     new_ts_covid_cote$weekly_aggregated_residuals[33:37],
     type = "l",
     xlab = "Date",
     ylab = "Cote Prediction vs. Acutal: Full Model")
lines(new_ts_covid_cote$week[33:37],                                       # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Cote Full
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 311.4124
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 318.1598
```

```
#Ghana Example on aggregate weekly
ts_covid_ghana = ts(covid_ghana$residuals,frequency = 7)
ts_covid_ghana_2 = ts(colSums(matrix(ts_covid_ghana, nrow=7)))
```

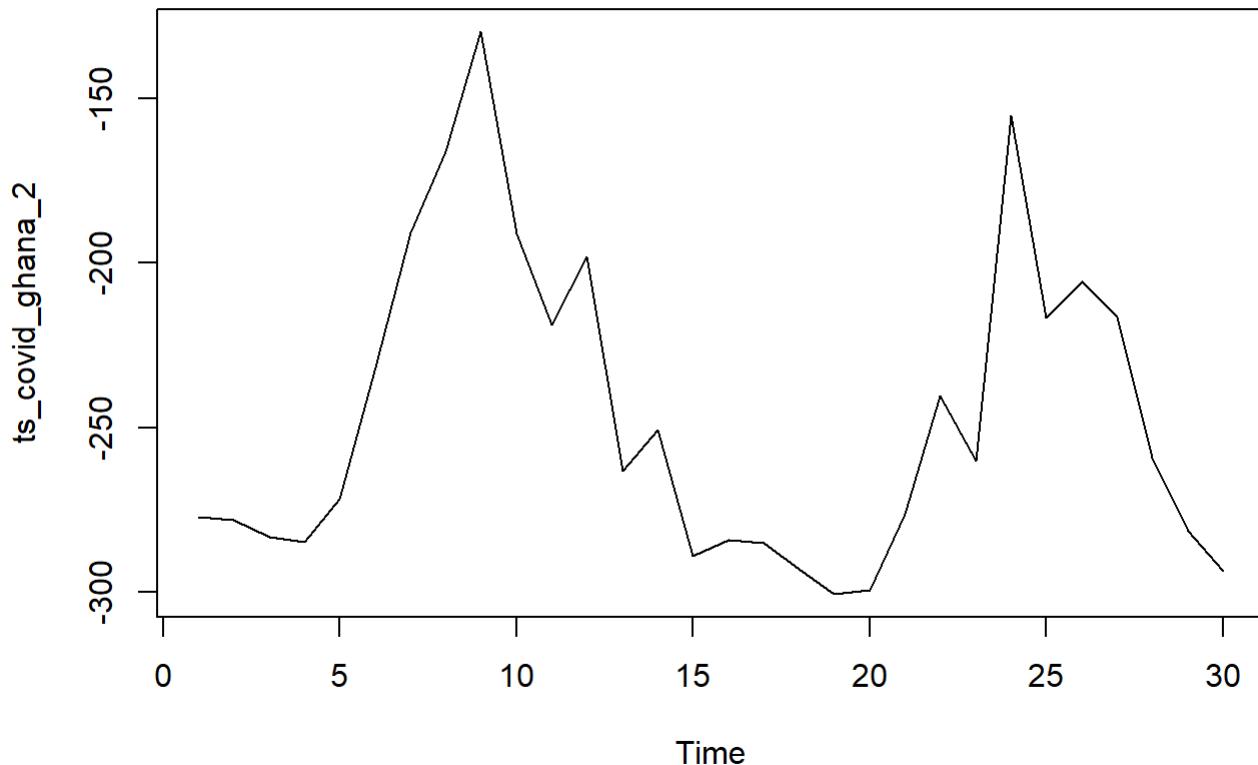
```
## Warning in matrix(ts_covid_ghana, nrow = 7): data length [208] is not a sub-
## multiple or multiple of the number of rows [7]
```

```
vaccine_policy = colSums(matrix(as.numeric(covid_ghana$vaccination_policy), nrow=7))

## Warning in matrix(as.numeric(covid_ghana$vaccination_policy), nrow = 7): data
## length [208] is not a sub-multiple or multiple of the number of rows [7]

new_ts_covid_ghana = data.frame(1:30,ts_covid_ghana_2,vaccine_policy)
names(new_ts_covid_ghana) = c("week","weekly_aggregated_residuals","vaccination_policy")

plot.ts(ts_covid_ghana_2)
```



```
regression_model = lm(weekly_aggregated_residuals~1, data = new_ts_covid_ghana[1:25,])
summary(regression_model)
```

```

## 
## Call:
## lm(formula = weekly_aggregated_residuals ~ 1, data = new_ts_covid_ghana[1:25,
##   ])
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -55.17 -38.72 -17.59  28.87 115.61 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -245.37     9.84   -24.93 <2e-16 ***
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 49.2 on 24 degrees of freedom

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: weekly_aggregated_residuals
##           Df Sum Sq Mean Sq F value Pr(>F)    
## Residuals 24  58101  2420.8

```

```

error = residuals(regression_model)
ts_error = ts(error)

```

```
#ARIMA on error
```

```

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)

```

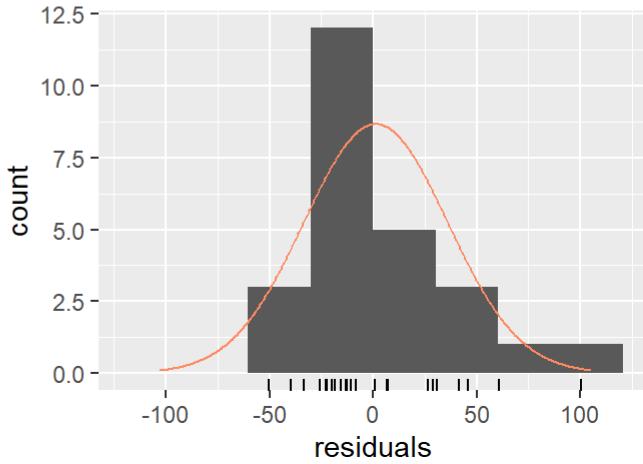
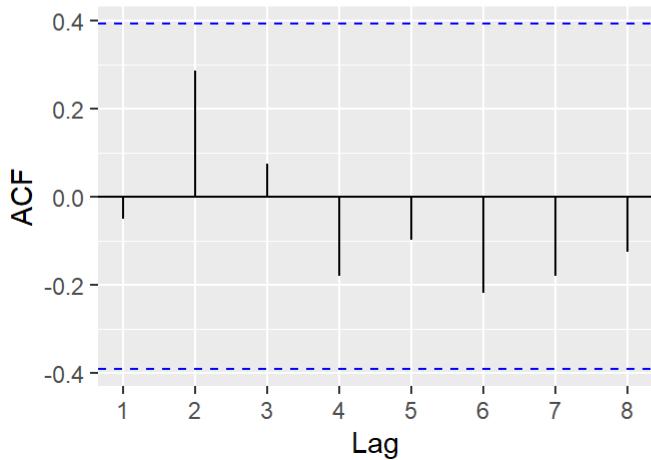
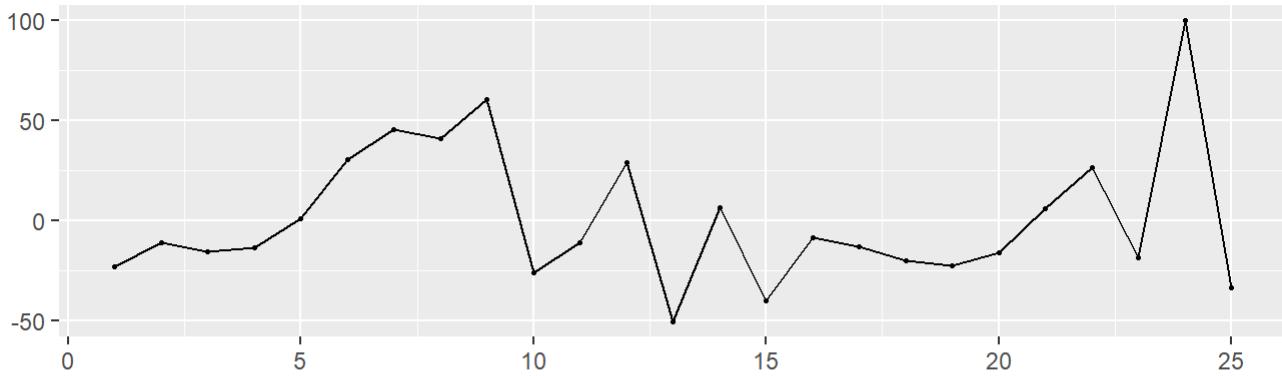
```

## Series: ts_error
## ARIMA(1,0,0) with zero mean
## 
## Coefficients:
##         ar1
##       0.6927
## s.e.  0.1361
## 
## sigma^2 estimated as 1207:  log likelihood=-123.99
## AIC=251.98  AICc=252.53  BIC=254.42
## 
## Training set error measures:
##               ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 1.153442 34.04175 26.77771 64.13638 92.9342 0.9800371 -0.05051922

```

```
checkresiduals(error_model1)
```

Residuals from ARIMA(1,0,0) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(1,0,0) with zero mean  
## Q* = 4.0093, df = 4, p-value = 0.4048  
##  
## Model df: 1. Total lags used: 5
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

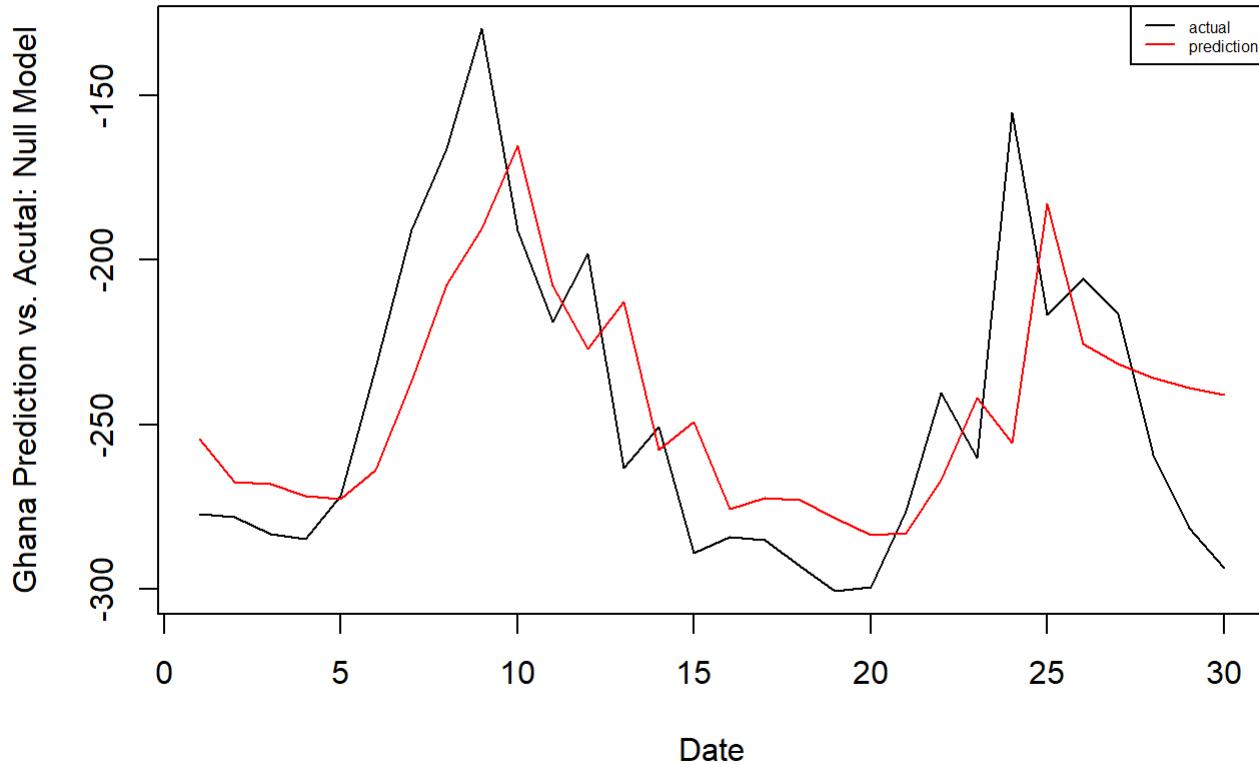
regression_prediction = predict(regression_model, newdata = new_ts_covid_ghana[26:30,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_ghana$weekly_aggregated_residuals[1:30])
colnames(compare_table) = c("predict","actual")

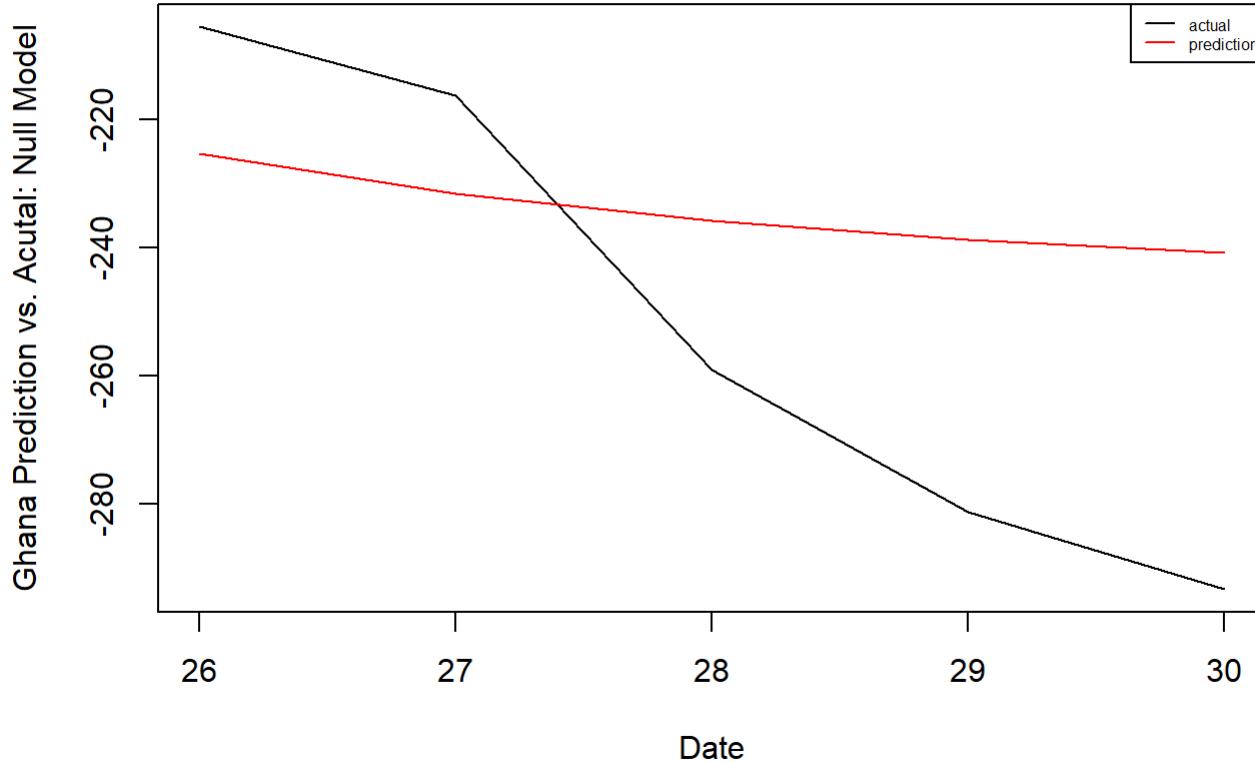
plot(new_ts_covid_ghana$week[1:30],                                     # Draw first time series
      new_ts_covid_ghana$weekly_aggregated_residuals[1:30],
      type = "l",
      xlab = "Date",
      ylab = "Ghana Prediction vs. Acutal: Null Model")
lines(new_ts_covid_ghana$week[1:30],                                      # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```

plot(new_ts_covid_ghana$week[26:30],                                     # Draw first time series
     new_ts_covid_ghana$weekly_aggregated_residuals[26:30],
     type = "l",
     xlab = "Date",
     ylab = "Ghana Prediction vs. Acutal: Null Model")
lines(new_ts_covid_ghana$week[26:30],                                       # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)

```



```
#Training MSE Ghana Null
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 1158.84
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 1150.946
```

```
regression_model = lm(weekly_aggregated_residuals~vaccination_policy, data = new_ts_covid_ghana[1:25,])
summary(regression_model)
```

```
##
## Call:
## lm(formula = weekly_aggregated_residuals ~ vaccination_policy,
##     data = new_ts_covid_ghana[1:25, ])
##
## Residuals:
##    Min      1Q Median      3Q     Max
## -56.63 -37.00 -12.07  36.99 106.98
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -216.7710   19.5713 -11.076 1.07e-10 ***
## vaccination_policy -1.6251    0.9725  -1.671    0.108
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 47.46 on 23 degrees of freedom
## Multiple R-squared:  0.1083, Adjusted R-squared:  0.06951
## F-statistic: 2.793 on 1 and 23 DF,  p-value: 0.1082
```

```
anova(regression_model)
```

```
## Analysis of Variance Table
##
## Response: weekly_aggregated_residuals
##              Df Sum Sq Mean Sq F value Pr(>F)
## vaccination_policy  1  6291  6290.9  2.7927 0.1082
## Residuals          23 51810  2252.6
```

```
error = residuals(regression_model)
ts_error = ts(error)

#ARIMA on error

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)
```

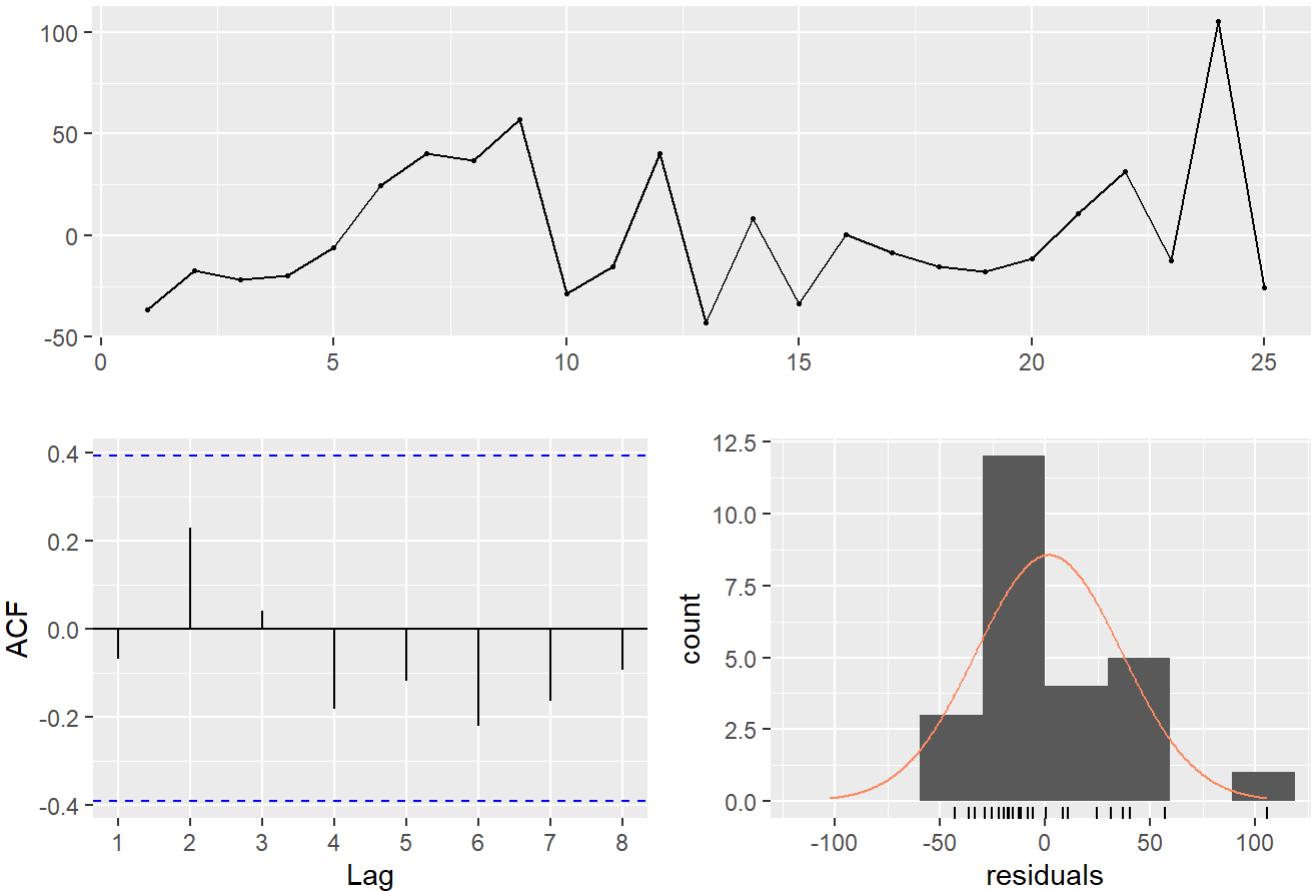
```

## Series: ts_error
## ARIMA(1,0,0) with zero mean
##
## Coefficients:
##      ar1
##      0.6670
## s.e.  0.1482
##
## sigma^2 estimated as 1204:  log likelihood=-123.93
## AIC=251.86   AICc=252.4   BIC=254.3
##
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 1.720138 34.00371 26.7172 130.3091 250.7883 0.9611488 -0.06827427

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(1,0,0) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(1,0,0) with zero mean  
## Q* = 3.2821, df = 4, p-value = 0.5118  
##  
## Model df: 1. Total lags used: 5
```

#goodness of fit

```
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

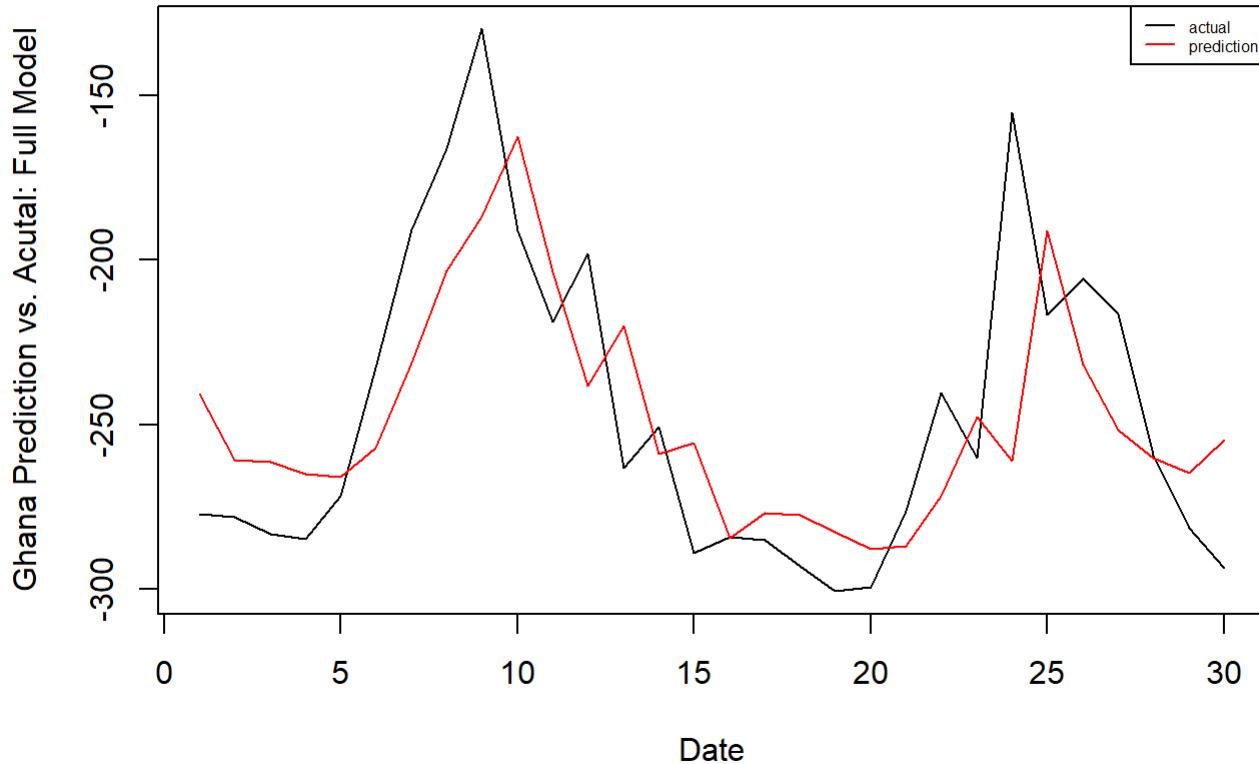
regression_prediction = predict(regression_model, newdata = new_ts_covid_ghana[26:30,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast
```

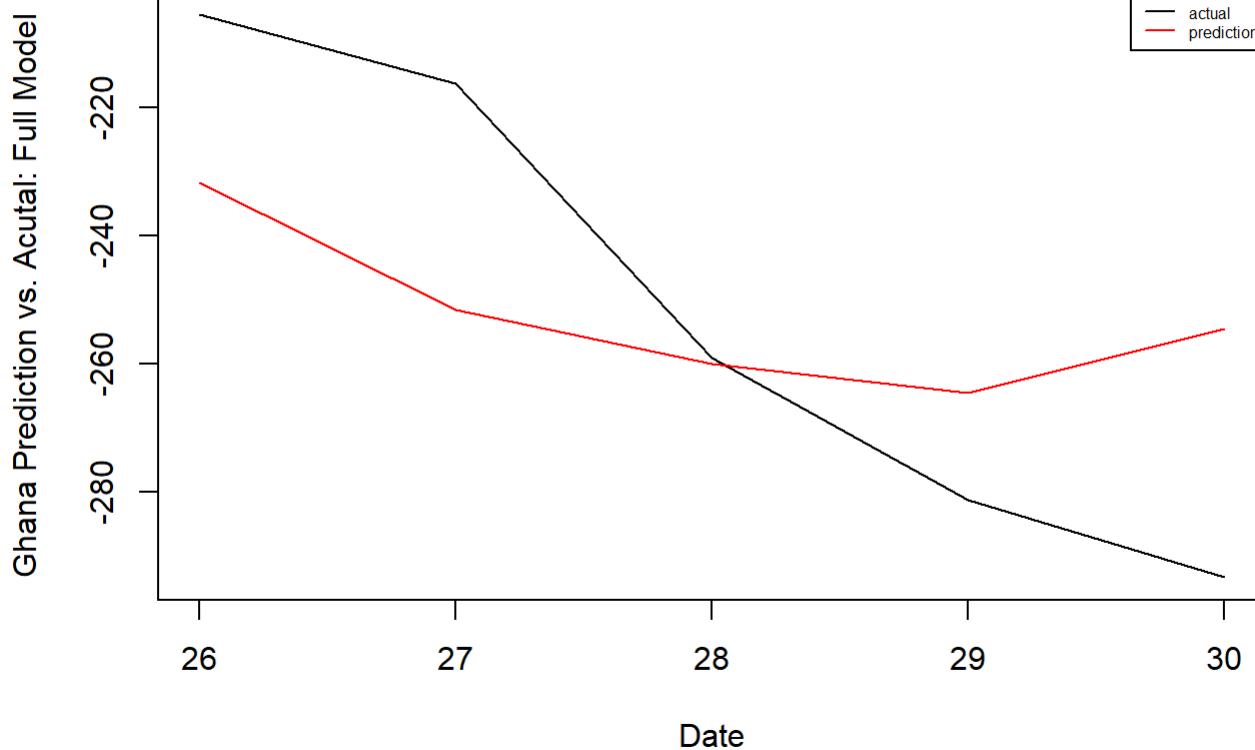
#compare prediction and actual

```
compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_ghana$weekly_aggregated_residuals[1:30])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_ghana$week[1:30],                                     # Draw first time series
      new_ts_covid_ghana$weekly_aggregated_residuals[1:30],
      type = "l",
      xlab = "Date",
      ylab = "Ghana Prediction vs. Acutal: Full Model")
lines(new_ts_covid_ghana$week[1:30],                                       # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_ghana$week[26:30],                                     # Draw first time series
     new_ts_covid_ghana$weekly_aggregated_residuals[26:30],
     type = "l",
     xlab = "Date",
     ylab = "Ghana Prediction vs. Acutal: Full Model")
lines(new_ts_covid_ghana$week[26:30],                                     # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Ghana Full
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 1156.252
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 745.3843
```

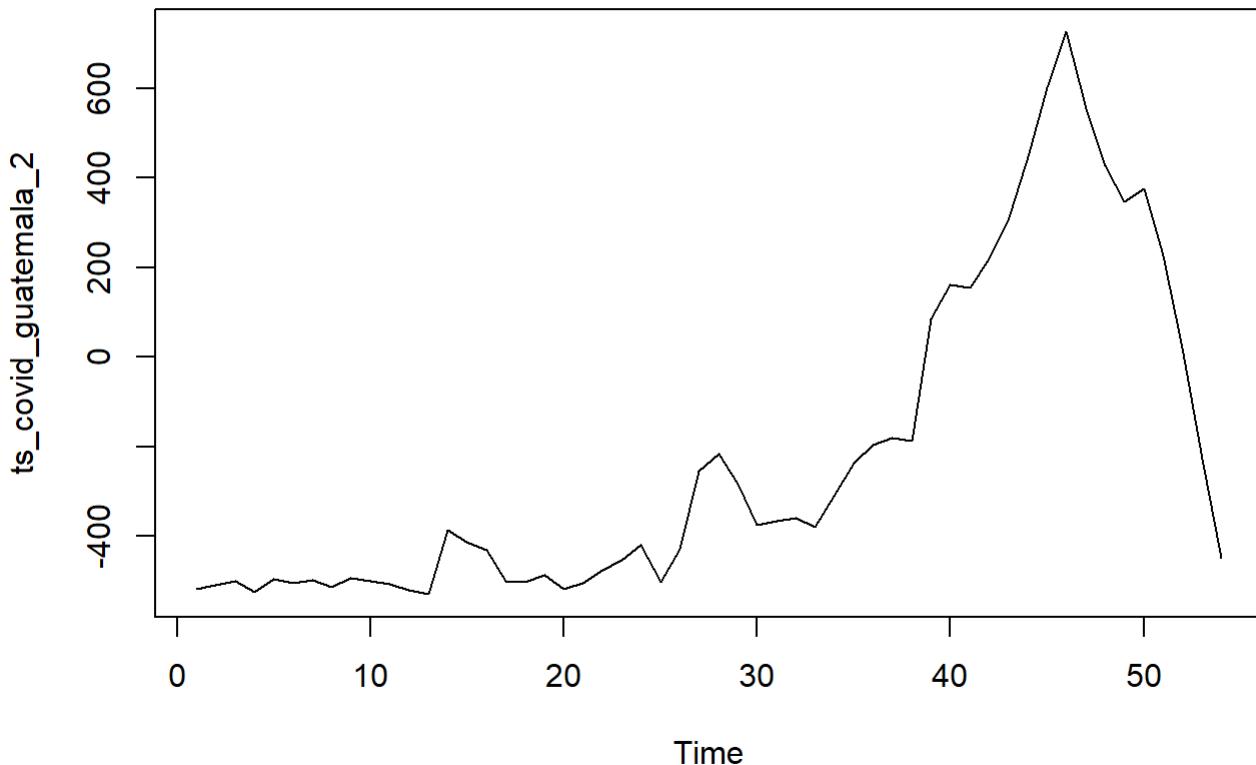
```
#Guatemala Example on aggregate weekly
ts_covid_guatemala = ts(covid_guatemala$residuals,frequency = 7)
ts_covid_guatemala_2 = ts(colSums(matrix(ts_covid_guatemala, nrow=7)))
```

```
## Warning in matrix(ts_covid_guatemala, nrow = 7): data length [374] is not a sub-
## multiple or multiple of the number of rows [7]
```

```
vaccine_policy = colSums(matrix(as.numeric(covid_guatemala$vaccination_policy), nrow=7))
```

```
## Warning in matrix(as.numeric(covid_guatemala$vaccination_policy), nrow = 7):  
## data length [374] is not a sub-multiple or multiple of the number of rows [7]
```

```
new_ts_covid_guatemala = data.frame(1:54,ts_covid_guatemala_2,vaccine_policy)  
names(new_ts_covid_guatemala) = c("week","weekly_aggregated_residuals","vaccination_policy")  
  
plot.ts(ts_covid_guatemala_2)
```



```
regression_model = lm(weekly_aggregated_residuals~1, data = new_ts_covid_guatemala[1:49,])  
summary(regression_model)
```

```

## 
## Call:
## lm(formula = weekly_aggregated_residuals ~ 1, data = new_ts_covid_guatemala[1:49,
##   ])
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -285.80 -257.25 -169.97  57.69  969.82 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -243.73     51.19  -4.762 1.81e-05 *** 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 358.3 on 48 degrees of freedom

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: weekly_aggregated_residuals
##           Df  Sum Sq Mean Sq F value Pr(>F)    
## Residuals 48  6162315 128382

```

```

error = residuals(regression_model)
ts_error = ts(error)

```

```

#ARIMA on error

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)

```

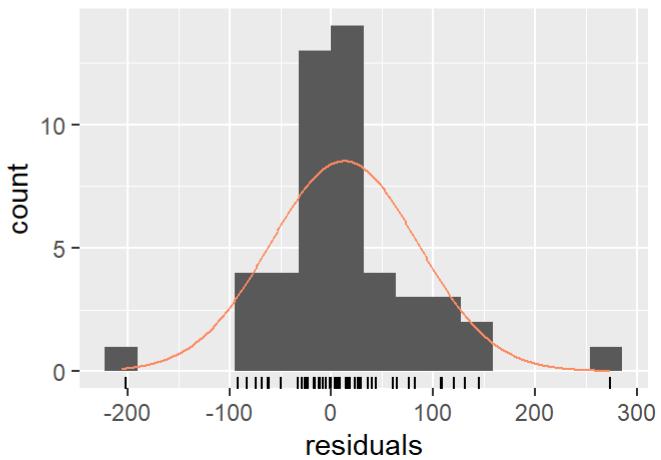
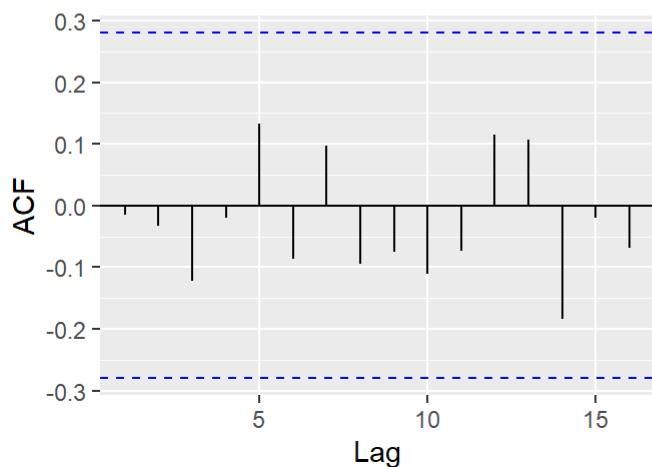
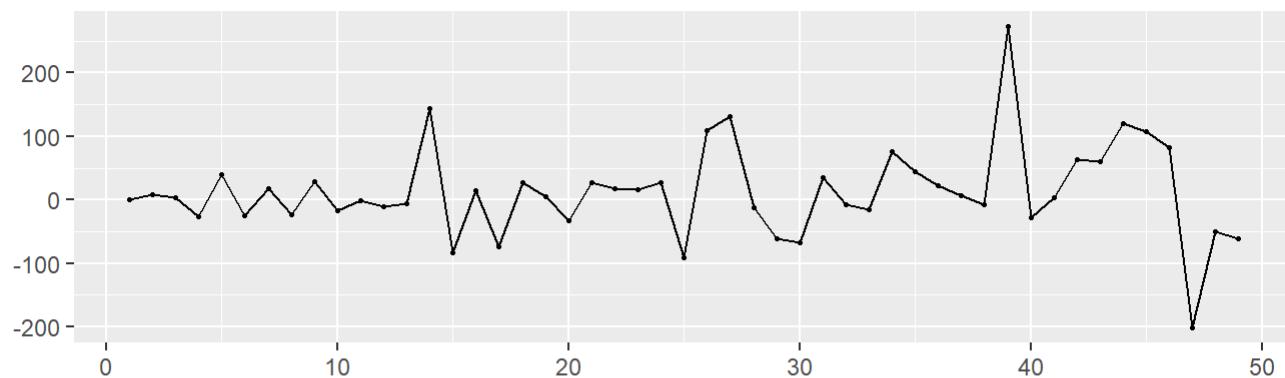
```

## Series: ts_error
## ARIMA(0,1,1)
## 
## Coefficients:
##             ma1
##             0.3868
## s.e.  0.1312
## 
## sigma^2 estimated as 5567:  log likelihood=-274.68
## AIC=553.35  AICc=553.62  BIC=557.1
## 
## Training set error measures:
##               ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 12.3586 73.07507 49.37455 -10.47867 57.09238 0.9001046 -0.01459681

```

```
checkresiduals(error_model1)
```

Residuals from ARIMA(0,1,1)



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(0,1,1)  
## Q* = 4.576, df = 9, p-value = 0.8696  
##  
## Model df: 1. Total lags used: 10
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

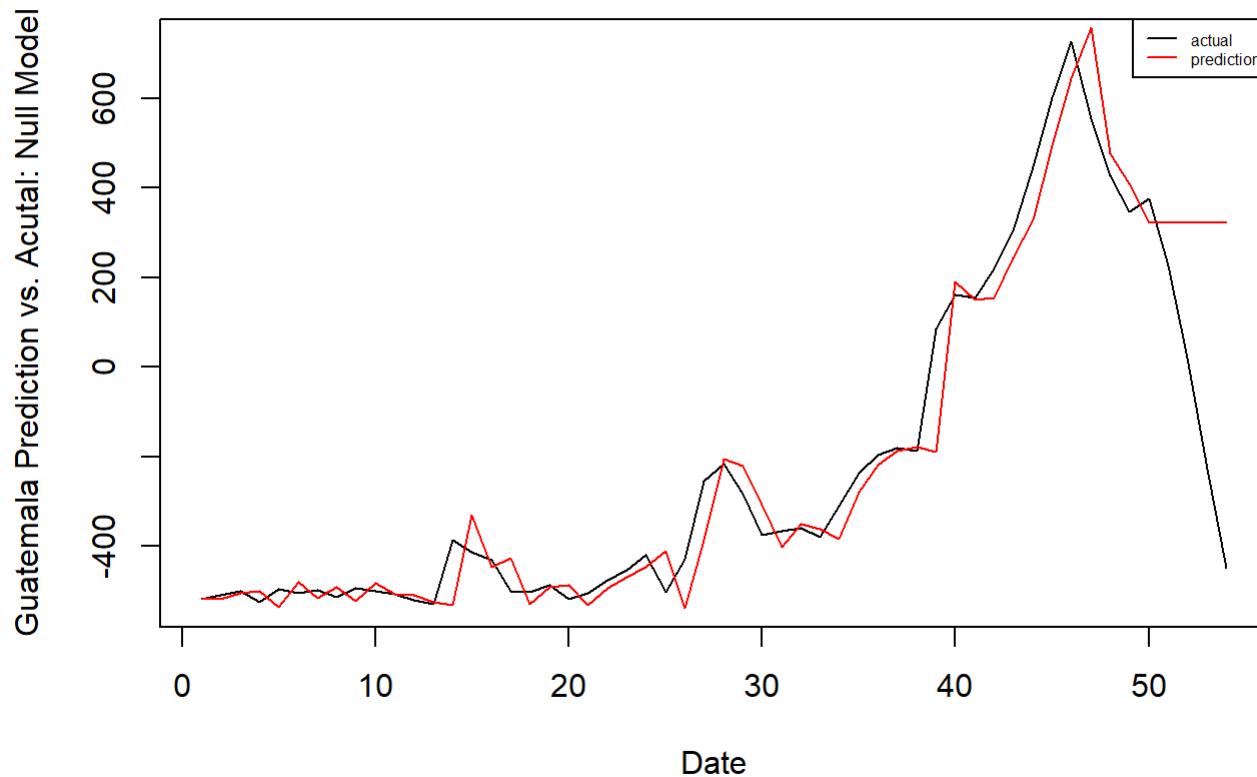
regression_prediction = predict(regression_model, newdata = new_ts_covid_guatemala[50:54,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

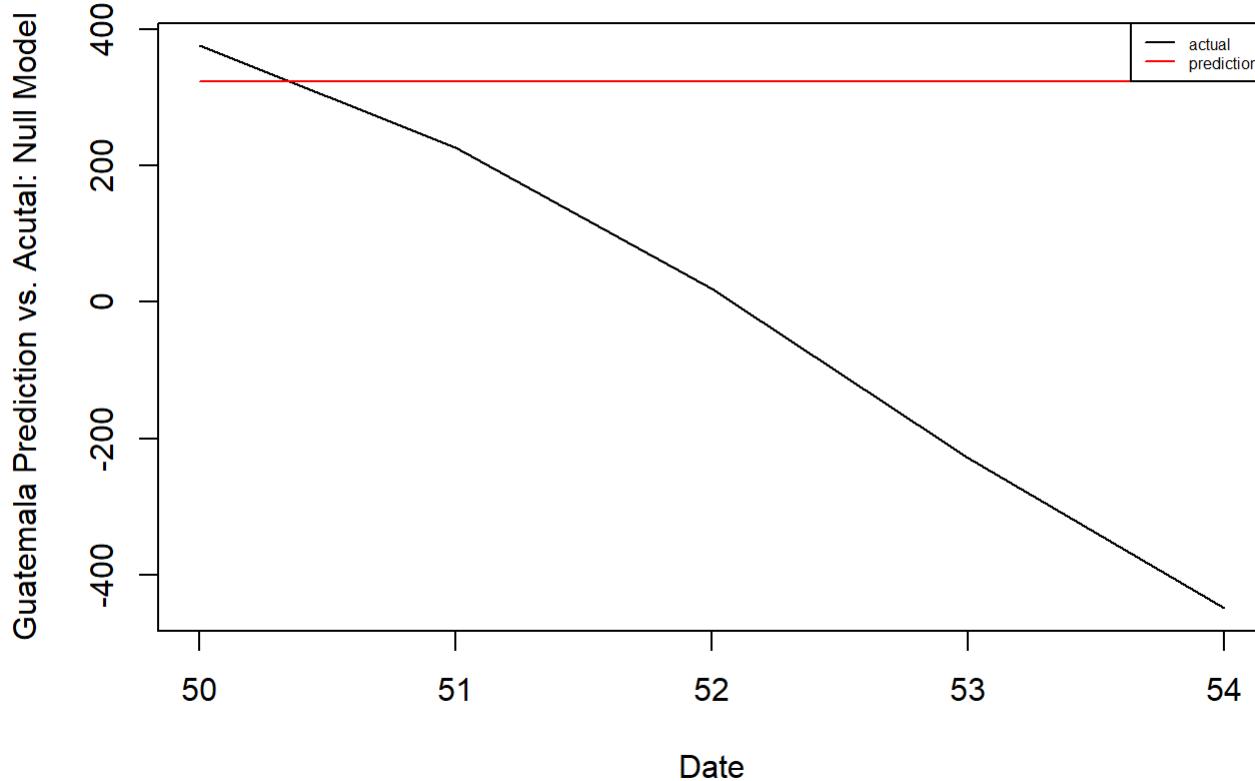
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_guatemala$weekly_aggregated_residuals[1:54])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_guatemala$week[1:54],                                     # Draw first time series
      new_ts_covid_guatemala$weekly_aggregated_residuals[1:54],
      type = "l",
      xlab = "Date",
      ylab = "Guatemala Prediction vs. Acutal: Null Model")
lines(new_ts_covid_guatemala$week[1:54],                                     # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_guatemala$week[50:54],  
     new_ts_covid_guatemala$weekly_aggregated_residuals[50:54],  
     type = "l",  
     xlab = "Date",  
     ylab = "Guatemala Prediction vs. Acutal: Null Model")  
lines(new_ts_covid_guatemala$week[50:54],  
      final_prediction_2,  
      type = "l",  
      col = "red")  
legend(x = "topright", legend=c("actual", "prediction"),  
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Guatemala Null
```

```
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 5339.966
```

```
#Testing MSE
```

```
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 200357.1
```

```
regression_model = lm(weekly_aggregated_residuals~vaccination_policy, data = new_ts_covid_guatemala[1:49,])
summary(regression_model)
```

```
##
## Call:
## lm(formula = weekly_aggregated_residuals ~ vaccination_policy,
##     data = new_ts_covid_guatemala[1:49, ])
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -513.50  -47.93   40.15   65.33  524.01 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -748.88     58.03  -12.90 < 2e-16 ***
## vaccination_policy 27.17      2.70   10.06 2.6e-13 ***
## ---    
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 203.9 on 47 degrees of freedom
## Multiple R-squared:  0.6831, Adjusted R-squared:  0.6763 
## F-statistic: 101.3 on 1 and 47 DF,  p-value: 2.603e-13
```

```
anova(regression_model)
```

```
## Analysis of Variance Table
##
## Response: weekly_aggregated_residuals
##              Df  Sum Sq Mean Sq F value    Pr(>F)    
## vaccination_policy  1 4209207 4209207 101.29 2.603e-13 ***
## Residuals         47 1953109   41556
## ---    
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
error = residuals(regression_model)
ts_error = ts(error)

#ARIMA on error

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)
```

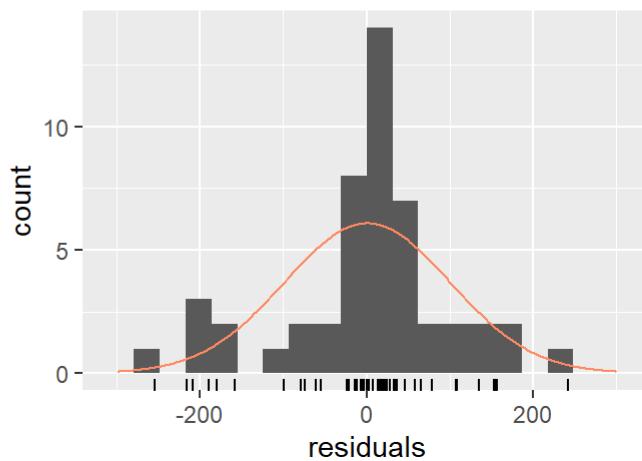
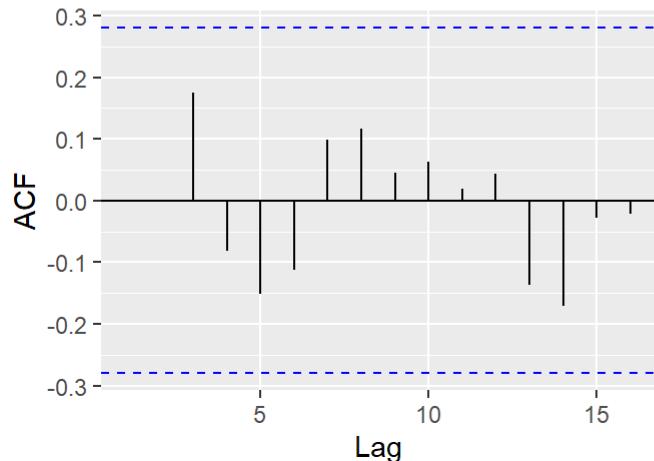
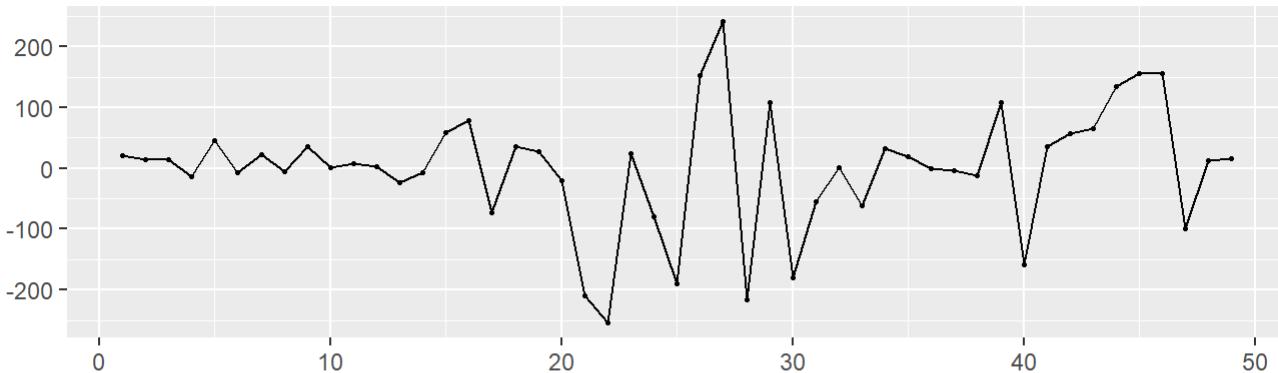
```

## Series: ts_error
## ARIMA(2,0,0) with zero mean
##
## Coefficients:
##      ar1     ar2
##      1.1471 -0.3688
## s.e.  0.1300  0.1310
##
## sigma^2 estimated as 10163: log likelihood=-295.31
## AIC=596.62   AICc=597.15   BIC=602.29
##
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.422652 98.73422 68.47015 -3.317319 118.1861 0.9044994
##          ACF1
## Training set 0.0003548907

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(2,0,0) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,0,0) with zero mean  
## Q* = 5.866, df = 8, p-value = 0.6622  
##  
## Model df: 2. Total lags used: 10
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

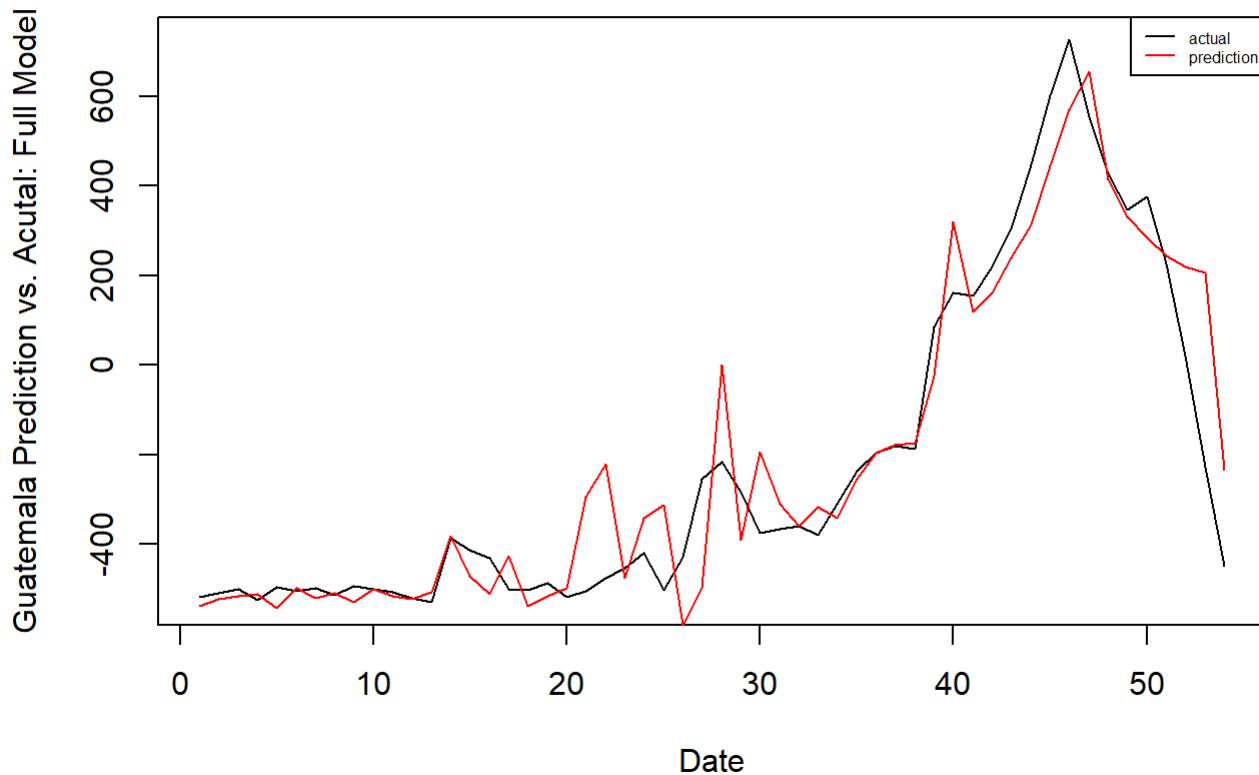
regression_prediction = predict(regression_model, newdata = new_ts_covid_guatemala[50:54,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

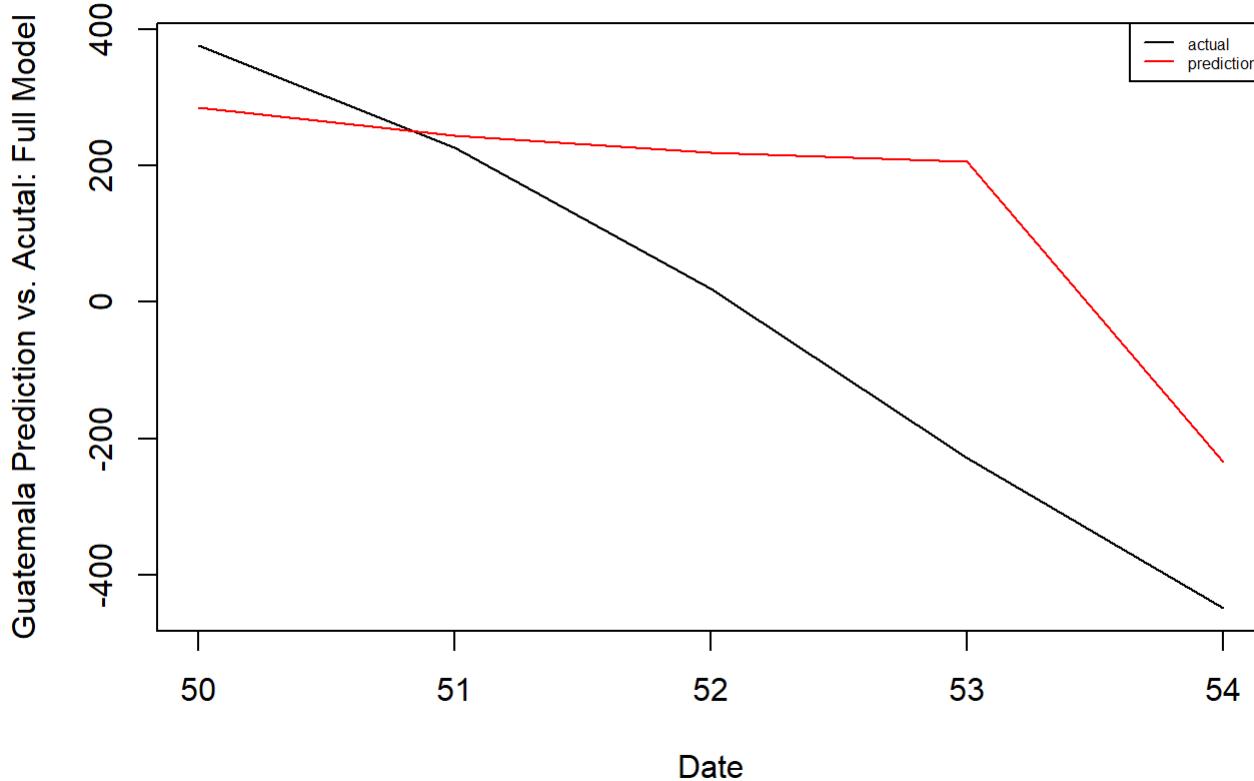
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_guatemala$weekly_aggregated_residuals[1:54])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_guatemala$week[1:54],  
      new_ts_covid_guatemala$weekly_aggregated_residuals[1:54],  
      type = "l",  
      xlab = "Date",  
      ylab = "Guatemala Prediction vs. Actual: Full Model")  
# Draw first time series
lines(new_ts_covid_guatemala$week[1:54],  
      compare_table$predict,  
      type = "l",  
      col = 'red')  
# Draw second time series
legend(x = "topright",legend=c("actual", "prediction"),  
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_guatemala$week[50:54],  
     new_ts_covid_guatemala$weekly_aggregated_residuals[50:54],  
     type = "l",  
     xlab = "Date",  
     ylab = "Guatemala Prediction vs. Acutal: Full Model")  
lines(new_ts_covid_guatemala$week[50:54],  
      final_prediction_2,  
      type = "l",  
      col = "red")  
legend(x = "topright", legend=c("actual", "prediction"),  
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Guatemala Full
```

```
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 9748.447
```

```
#Testing MSE
```

```
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 56498.24
```

```
#India Example on aggregate weekly
```

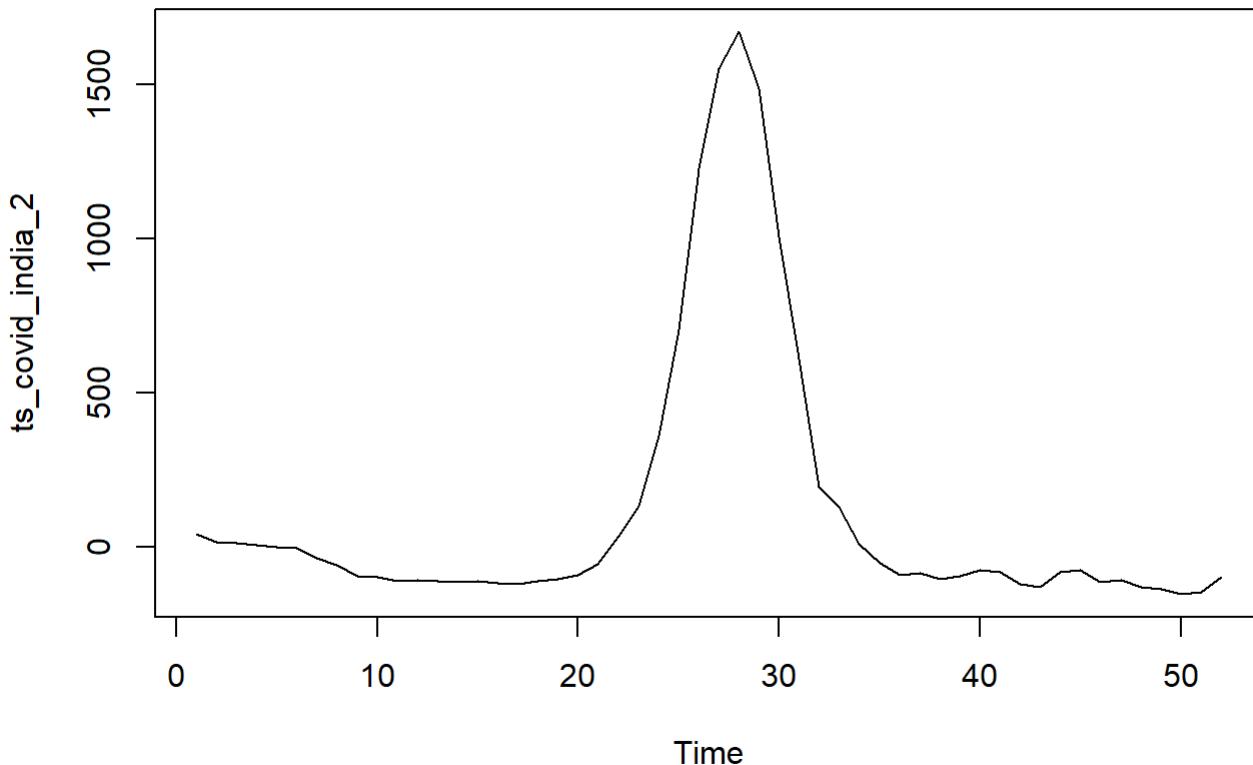
```
ts_covid_india = ts(covid_india$residuals,frequency = 7)
ts_covid_india_2 = ts(colSums(matrix(ts_covid_india, nrow=7)))
```

```
## Warning in matrix(ts_covid_india, nrow = 7): data length [362] is not a sub-
## multiple or multiple of the number of rows [7]
```

```
vaccine_policy = colSums(matrix(as.numeric(covid_india$vaccination_policy), nrow=7))
```

```
## Warning in matrix(as.numeric(covid_india$vaccination_policy), nrow = 7): data  
## length [362] is not a sub-multiple or multiple of the number of rows [7]
```

```
new_ts_covid_india = data.frame(1:52,ts_covid_india_2,vaccine_policy)  
names(new_ts_covid_india) = c("week","weekly_aggregated_residuals","vaccination_policy")  
  
plot.ts(ts_covid_india_2)
```



```
regression_model = lm(weekly_aggregated_residuals~1, data = new_ts_covid_india[1:47,])  
summary(regression_model)
```

```

## 
## Call:
## lm(formula = weekly_aggregated_residuals ~ 1, data = new_ts_covid_india[1:47,
##   ])
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -269.8 -245.3 -214.8 -100.8 1533.5 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 138.14     69.26   1.994   0.0521 .  
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 474.8 on 46 degrees of freedom

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: weekly_aggregated_residuals
##           Df  Sum Sq Mean Sq F value Pr(>F)    
## Residuals 46 10371969  225478

```

```

error = residuals(regression_model)
ts_error = ts(error)

```

```

#ARIMA on error

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)

```

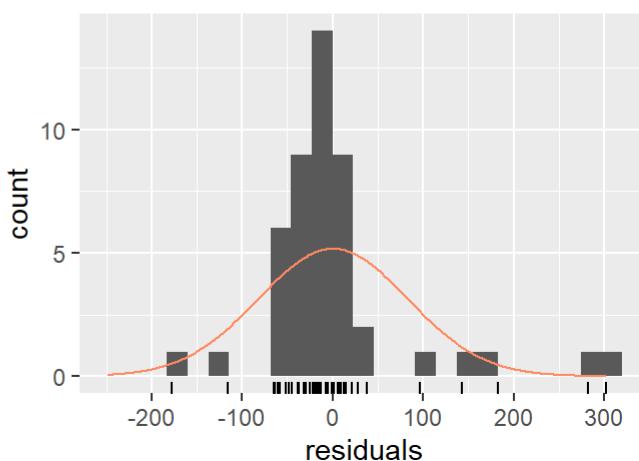
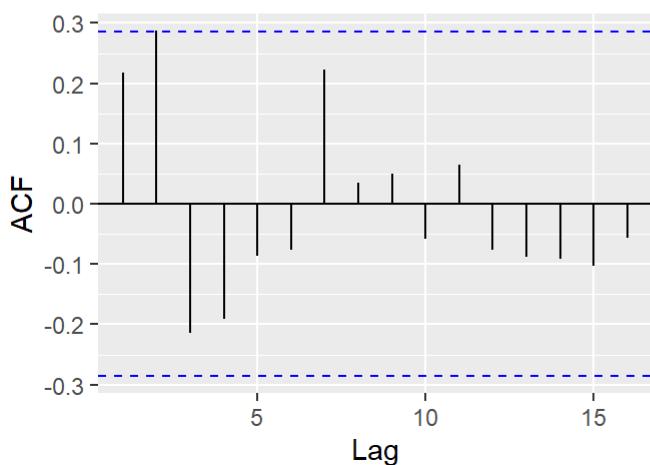
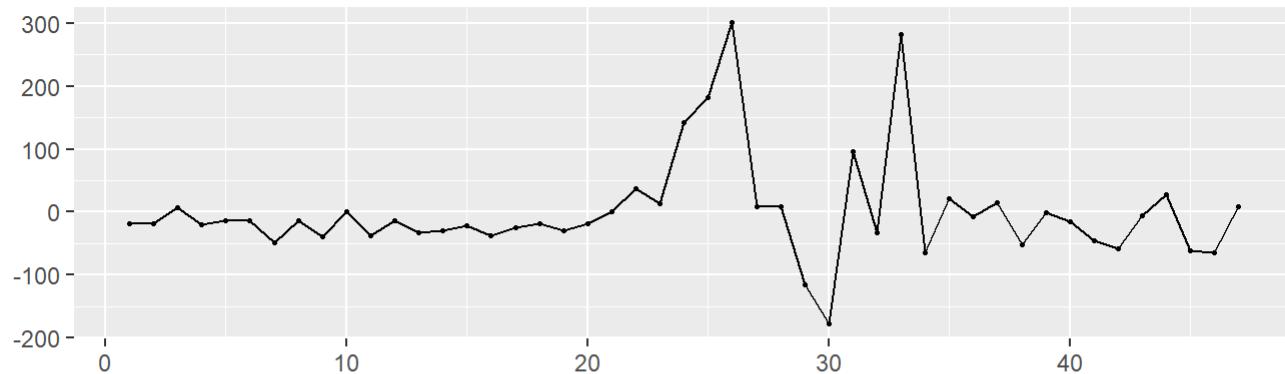
```

## Series: ts_error
## ARIMA(2,0,0) with zero mean
## 
## Coefficients:
##         ar1      ar2
##       1.719   -0.8293
## s.e.  0.070   0.0695
## 
## sigma^2 estimated as 7028: log likelihood=-276.06
## AIC=558.12   AICc=558.68   BIC=563.67
## 
## Training set error measures:
##               ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 0.1521017 82.02886 48.86054 -53.89103 77.26162 0.5707606 0.2177165

```

```
checkresiduals(error_model1)
```

Residuals from ARIMA(2,0,0) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,0,0) with zero mean  
## Q* = 14.72, df = 7, p-value = 0.03976  
##  
## Model df: 2. Total lags used: 9
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

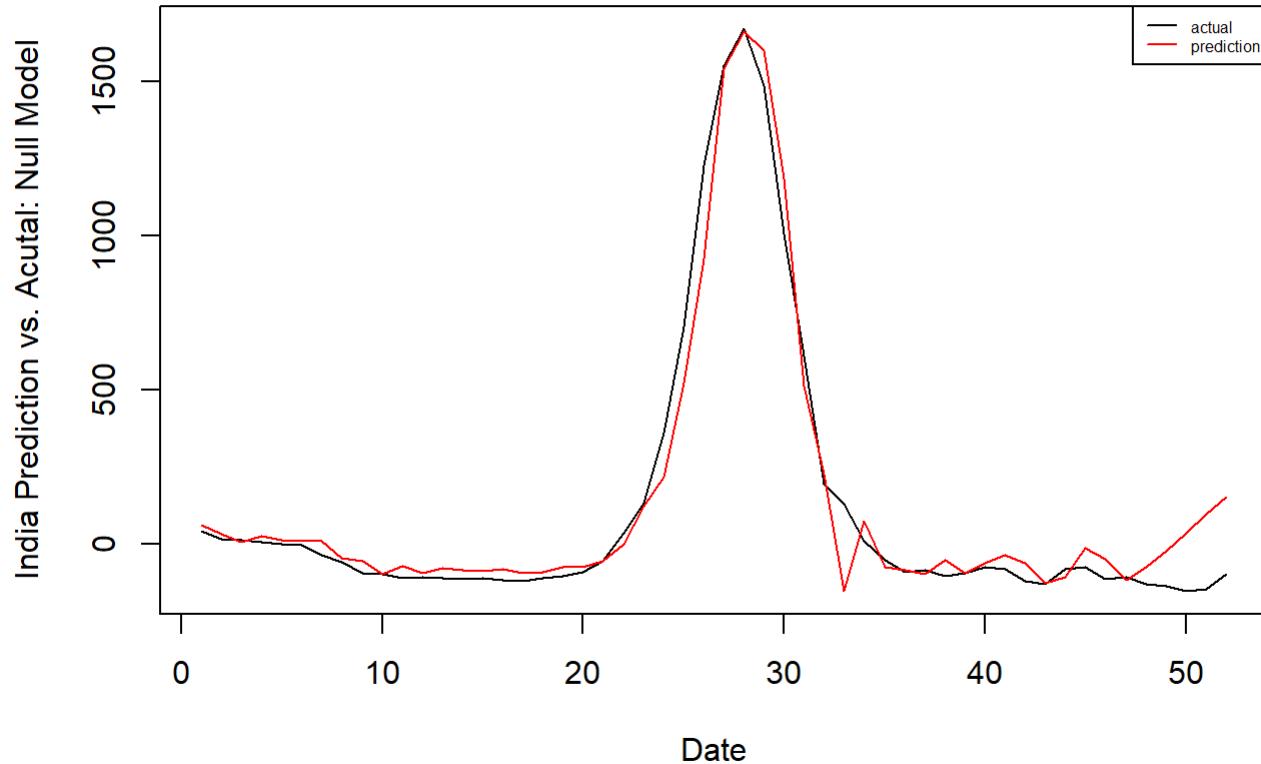
regression_prediction = predict(regression_model, newdata = new_ts_covid_india[48:52,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

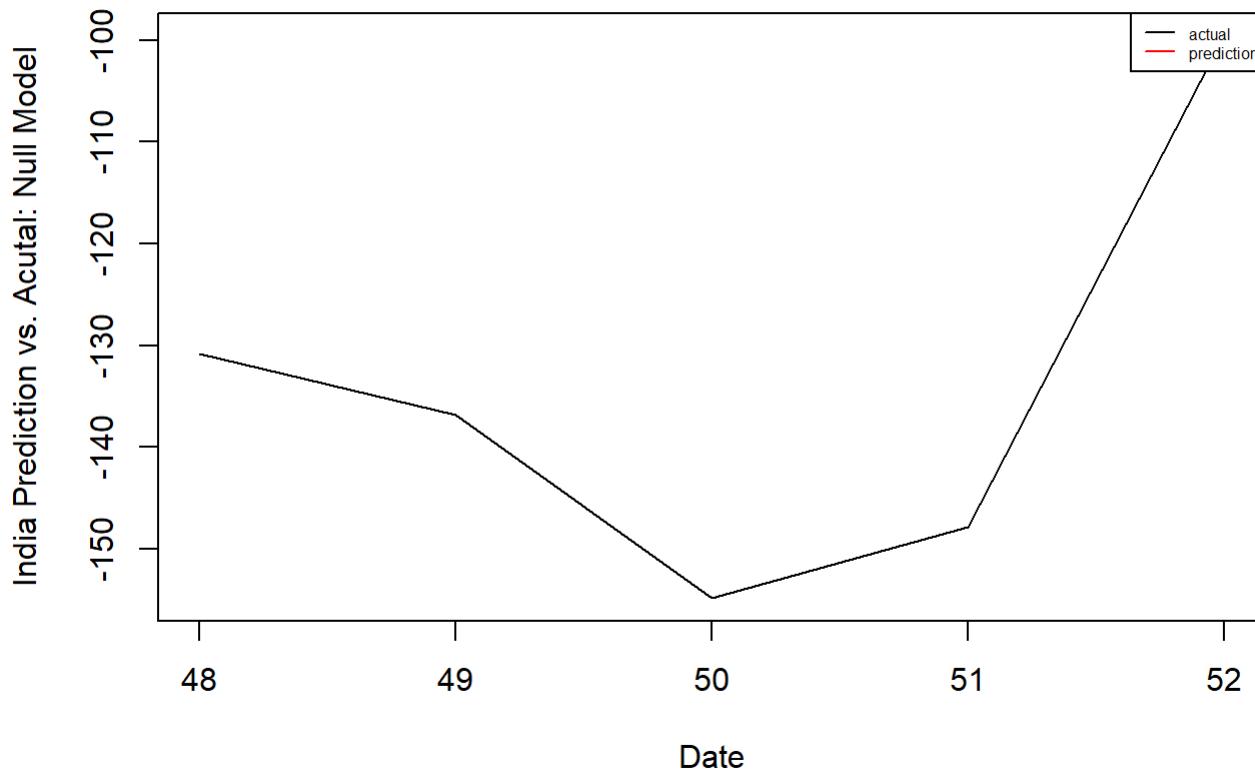
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_india$weekly_aggregated_residuals[1:52])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_india$week[1:52],                                     # Draw first time series
      new_ts_covid_india$weekly_aggregated_residuals[1:52],
      type = "l",
      xlab = "Date",
      ylab = "India Prediction vs. Acutal: Null Model")
lines(new_ts_covid_india$week[1:52],                                     # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_india$week[48:52],                                     # Draw first time series
     new_ts_covid_india$weekly_aggregated_residuals[48:52],
     type = "l",
     xlab = "Date",
     ylab = "India Prediction vs. Acutal: Null Model")
lines(new_ts_covid_india$week[48:52],                                       # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE India Null
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 6728.735
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 34547.26
```

```
regression_model = lm(weekly_aggregated_residuals~vaccination_policy, data = new_ts_covid_india[1:47,])
summary(regression_model)
```

```
##  
## Call:  
## lm(formula = weekly_aggregated_residuals ~ vaccination_policy,  
##      data = new_ts_covid_india[1:47, ])  
##  
## Residuals:  
##     Min      1Q   Median      3Q     Max  
## -457.21 -322.43 -129.39   89.27 1414.67  
##  
## Coefficients:  
##                         Estimate Std. Error t value Pr(>|t|)  
## (Intercept)       -174.42     153.56  -1.136  0.2621  
## vaccination_policy    12.32      5.46   2.257  0.0289 *  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 455 on 45 degrees of freedom  
## Multiple R-squared:  0.1017, Adjusted R-squared:  0.08173  
## F-statistic: 5.094 on 1 and 45 DF,  p-value: 0.0289
```

```
anova(regression_model)
```

```
## Analysis of Variance Table  
##  
## Response: weekly_aggregated_residuals  
##                         Df  Sum Sq Mean Sq F value Pr(>F)  
## vaccination_policy  1 1054791 1054791  5.0944 0.0289 *  
## Residuals          45 9317179  207048  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
error = residuals(regression_model)  
ts_error = ts(error)  
  
#ARIMA on error  
  
error_model = auto.arima(ts_error)  
fitted_error = fitted(error_model)  
summary(error_model)
```

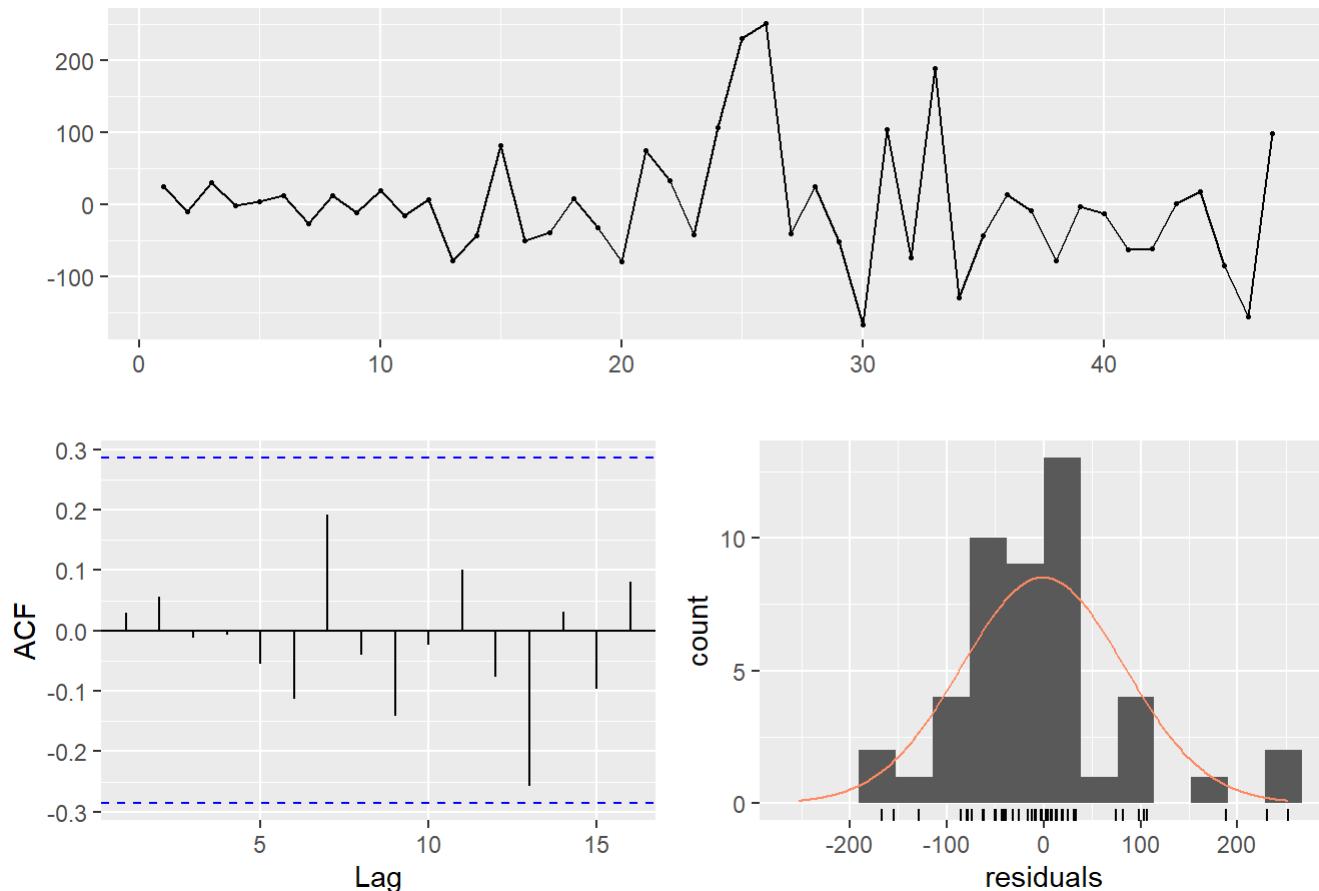
```

## Series: ts_error
## ARIMA(2,0,2) with zero mean
##
## Coefficients:
##      ar1     ar2     ma1     ma2
##      1.5191 -0.6670  0.3196  0.3334
## s.e.  0.1512  0.1451  0.2125  0.1538
##
## sigma^2 estimated as 7549: log likelihood=-276.77
## AIC=563.54   AICc=565   BIC=572.79
##
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -0.9666123 83.10409 58.46786 10.76172 36.87665 0.6467632
##          ACF1
## Training set 0.02945278

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(2,0,2) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,0,2) with zero mean  
## Q* = 4.517, df = 5, p-value = 0.4776  
##  
## Model df: 4. Total lags used: 9
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

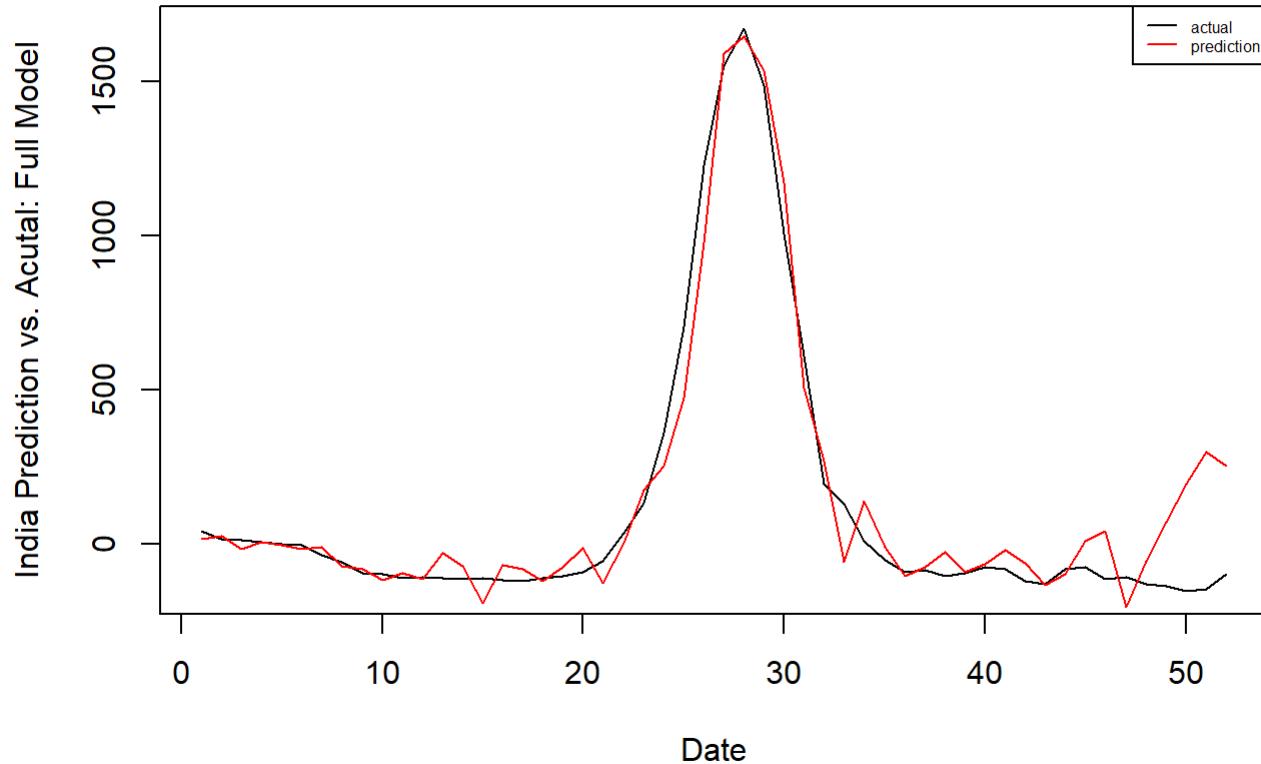
regression_prediction = predict(regression_model, newdata = new_ts_covid_india[48:52,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

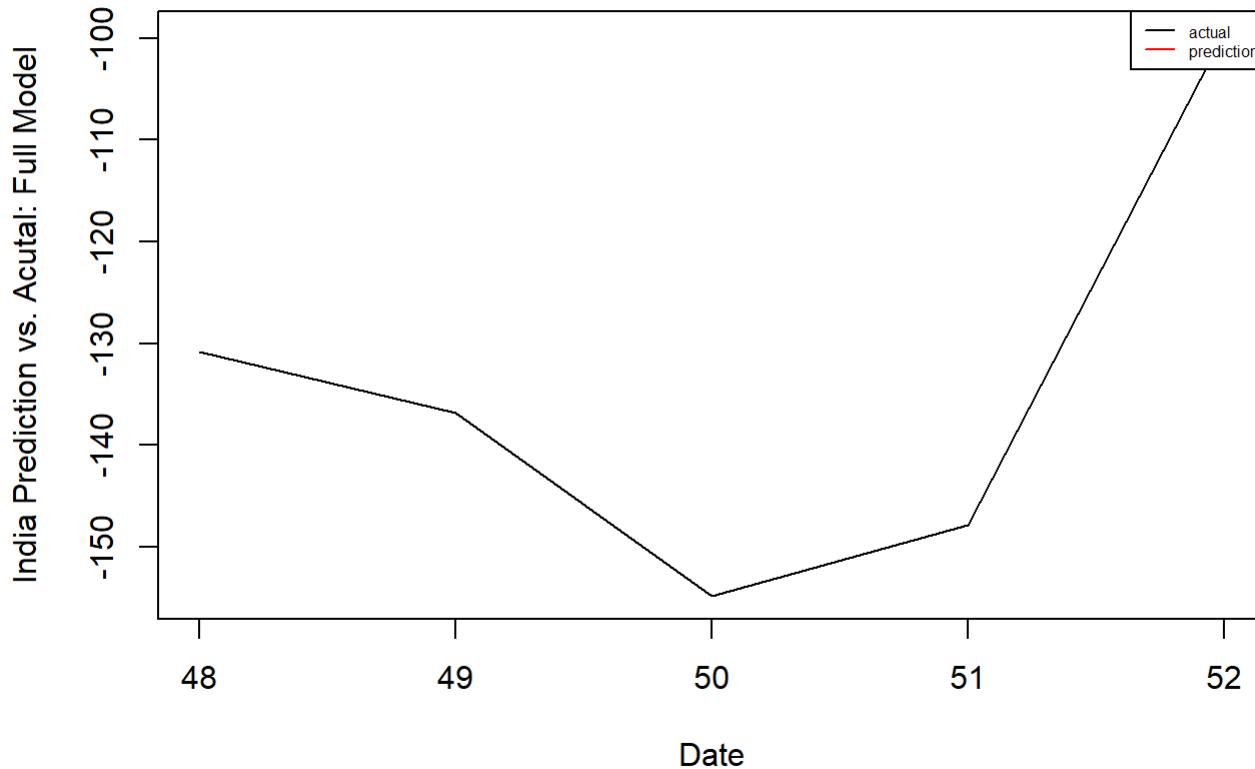
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_india$weekly_aggregated_residuals[1:52])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_india$week[1:52],  
      new_ts_covid_india$weekly_aggregated_residuals[1:52],  
      type = "l",  
      xlab = "Date",  
      ylab = "India Prediction vs. Actual: Full Model")  
lines(new_ts_covid_india$week[1:52],  
      compare_table$predict,  
      type = "l",  
      col = 'red')  
legend(x = "topright",legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_india$week[48:52],                                     # Draw first time series
     new_ts_covid_india$weekly_aggregated_residuals[48:52],
     type = "l",
     xlab = "Date",
     ylab = "India Prediction vs. Acutal: Full Model")
lines(new_ts_covid_india$week[48:52],                                      # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE India Full
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 6906.291
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 98352.42
```

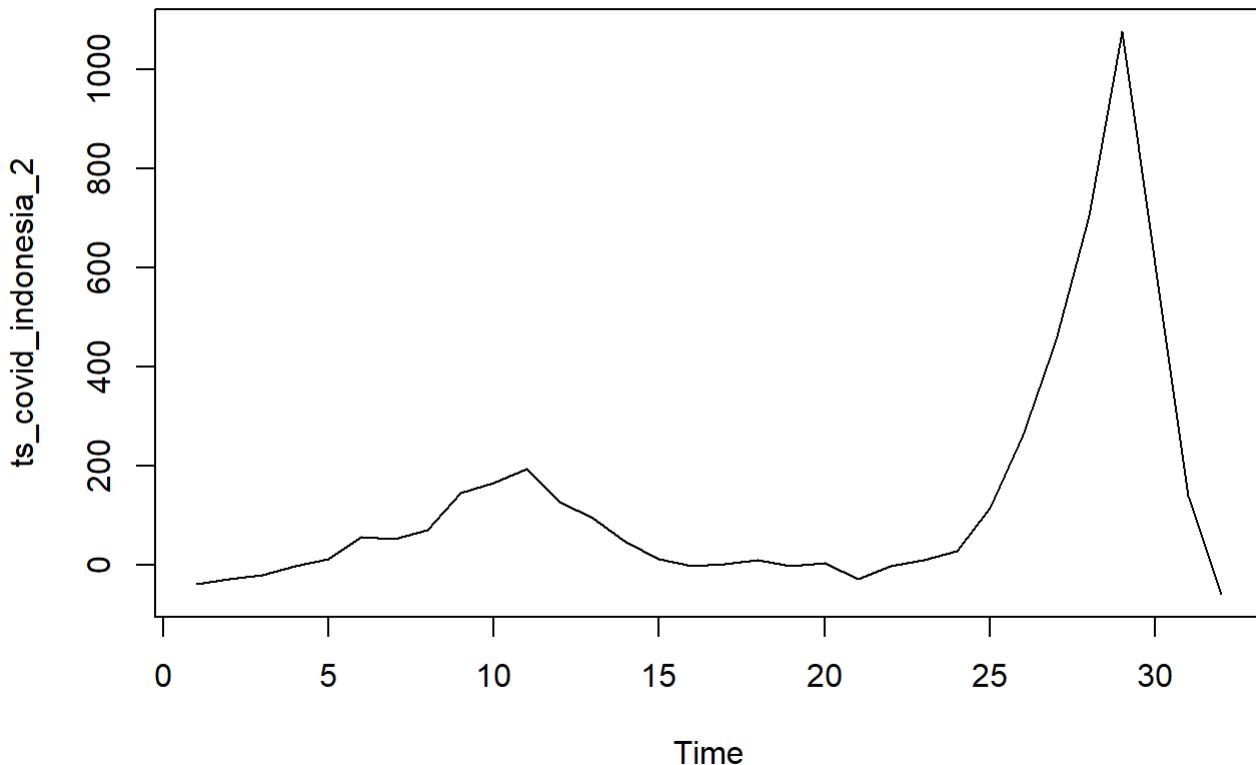
```
#Indonesia Example on aggregate weekly
ts_covid_indonesia = ts(covid_indonesia$residuals,frequency = 7)
ts_covid_indonesia_2 = ts(colSums(matrix(ts_covid_indonesia, nrow=7)))
```

```
## Warning in matrix(ts_covid_indonesia, nrow = 7): data length [219] is not a sub-
## multiple or multiple of the number of rows [7]
```

```
vaccine_policy = colSums(matrix(as.numeric(covid_indonesia$vaccination_policy), nrow=7))
```

```
## Warning in matrix(as.numeric(covid_indonesia$vaccination_policy), nrow = 7):  
## data length [219] is not a sub-multiple or multiple of the number of rows [7]
```

```
new_ts_covid_indonesia = data.frame(1:32,ts_covid_indonesia_2,vaccine_policy)  
names(new_ts_covid_indonesia) = c("week","weekly_aggregated_residuals","vaccination_policy")  
  
plot.ts(ts_covid_indonesia_2)
```



```
regression_model = lm(weekly_aggregated_residuals~1, data = new_ts_covid_indonesia[1:27,])  
summary(regression_model)
```

```

## 
## Call:
## lm(formula = weekly_aggregated_residuals ~ 1, data = new_ts_covid_indonesia[1:27,
##   ])
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -102.25 -65.98 -52.13  39.85 390.41 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  63.83     20.90   3.054  0.00515 **  
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 108.6 on 26 degrees of freedom

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: weekly_aggregated_residuals
##           Df Sum Sq Mean Sq F value Pr(>F)    
## Residuals 26 306531 11790

```

```

error = residuals(regression_model)
ts_error = ts(error)

```

```
#ARIMA on error
```

```

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)

```

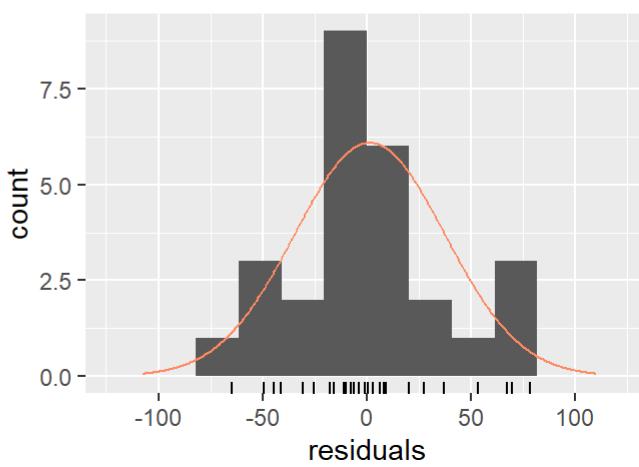
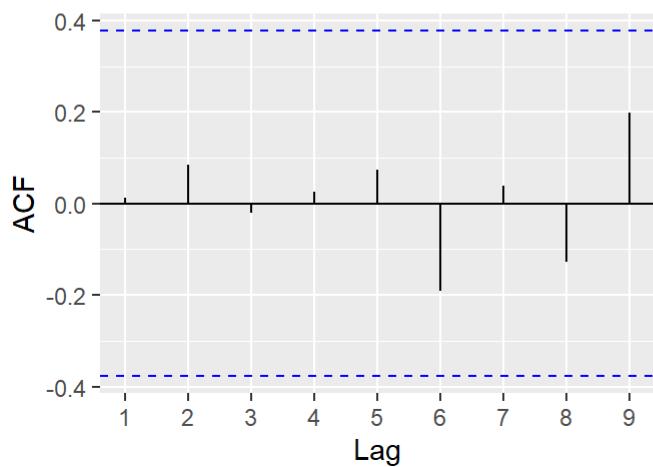
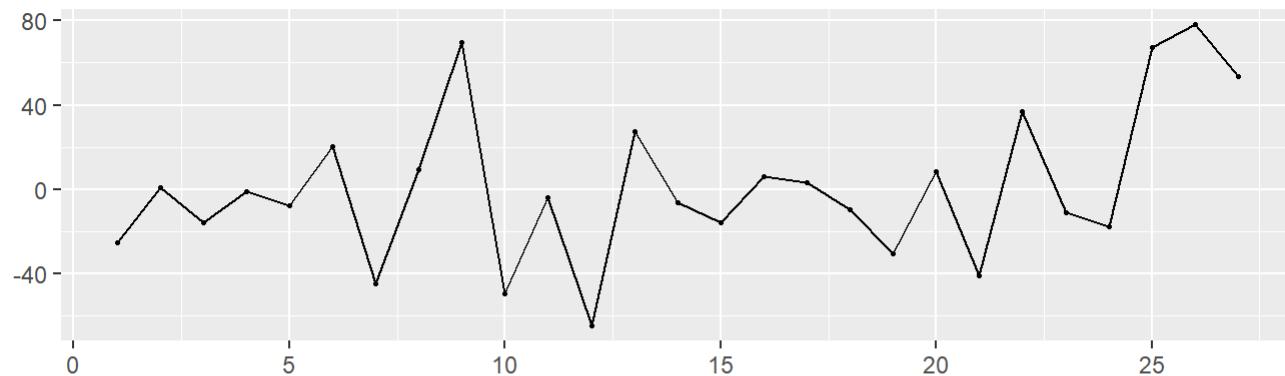
```

## Series: ts_error
## ARIMA(2,0,2) with non-zero mean
## 
## Coefficients:
##         ar1      ar2      ma1      ma2      mean
##       1.6795 -0.7798  0.1194  0.3676  73.1623
## s.e.  0.3120  0.2896  0.4215  0.3145 116.5817
## 
## sigma^2 estimated as 1553: log likelihood=-137.53
## AIC=287.06  AICc=291.26  BIC=294.83
## 
## Training set error measures:
##               ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 1.380255 35.56799 26.86736 26.92491 62.23681 0.7148083 0.01272989

```

```
checkresiduals(error_model1)
```

Residuals from ARIMA(2,0,2) with non-zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,0,2) with non-zero mean  
## Q* = 2.564, df = 3, p-value = 0.4638  
##  
## Model df: 5. Total lags used: 8
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

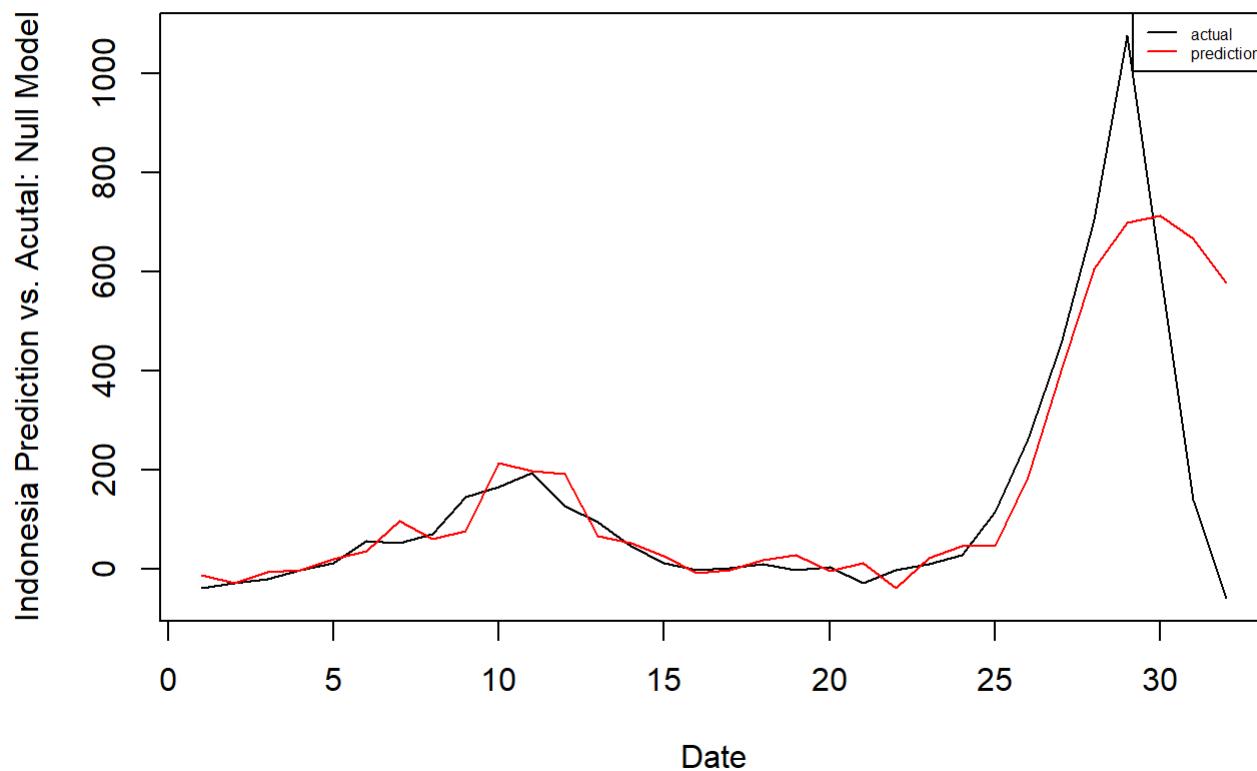
regression_prediction = predict(regression_model, newdata = new_ts_covid_indonesia[28:32,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

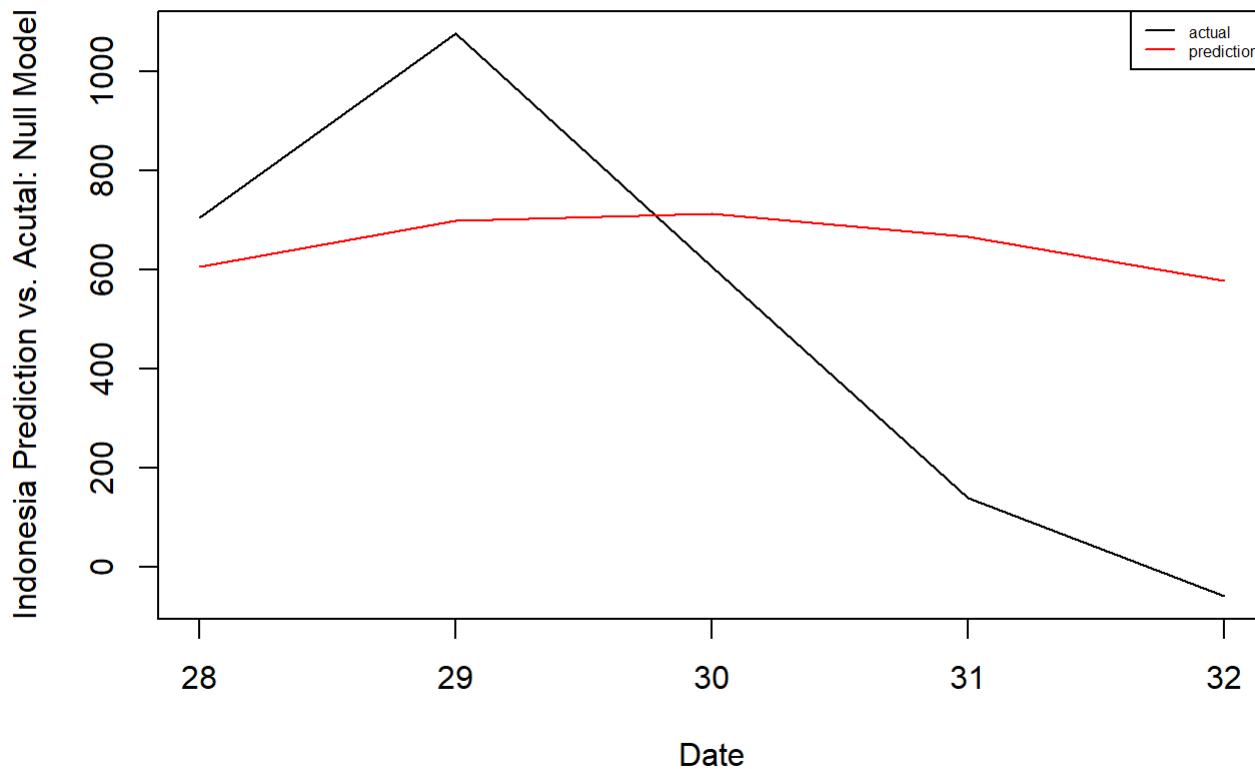
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_indonesia$weekly_aggregated_residuals[1:32])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_indonesia$week[1:32],                                     # Draw first time series
      new_ts_covid_indonesia$weekly_aggregated_residuals[1:32],
      type = "l",
      xlab = "Date",
      ylab = "Indonesia Prediction vs. Acutal: Null Model")
lines(new_ts_covid_indonesia$week[1:32],                                     # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_indonesia$week[28:32],  
      new_ts_covid_indonesia$weekly_aggregated_residuals[28:32],  
      type = "l",  
      xlab = "Date",  
      ylab = "Indonesia Prediction vs. Acutal: Null Model")  
lines(new_ts_covid_indonesia$week[28:32],  
      final_prediction_2,  
      type = "l",  
      col = "red")  
legend(x = "topright", legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Indonesia Null
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 1265.082
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 170099.7
```

```
regression_model = lm(weekly_aggregated_residuals~vaccination_policy, data = new_ts_covid_indonesia[1:27,])
summary(regression_model)
```

```
##
## Call:
## lm(formula = weekly_aggregated_residuals ~ vaccination_policy,
##     data = new_ts_covid_indonesia[1:27, ])
##
## Residuals:
##    Min      1Q   Median      3Q      Max
## -113.61  -75.15  -39.06   39.06  369.63
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -13.578     56.901  -0.239   0.813
## vaccination_policy  4.676      3.207   1.458   0.157
##
## Residual standard error: 106.3 on 25 degrees of freedom
## Multiple R-squared:  0.07835,   Adjusted R-squared:  0.04149
## F-statistic: 2.125 on 1 and 25 DF, p-value: 0.1573
```

```
anova(regression_model)
```

```
## Analysis of Variance Table
##
## Response: weekly_aggregated_residuals
##              Df Sum Sq Mean Sq F value Pr(>F)
## vaccination_policy  1  24017  24017  2.1253 0.1573
## Residuals          25 282514  11301
```

```
error = residuals(regression_model)
ts_error = ts(error)

#ARIMA on error

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)
```

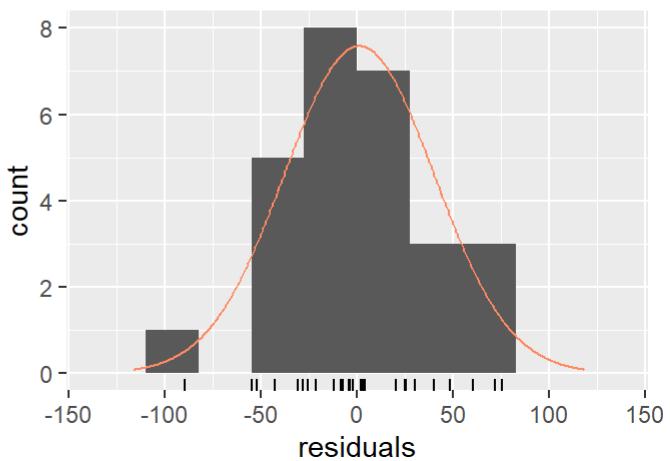
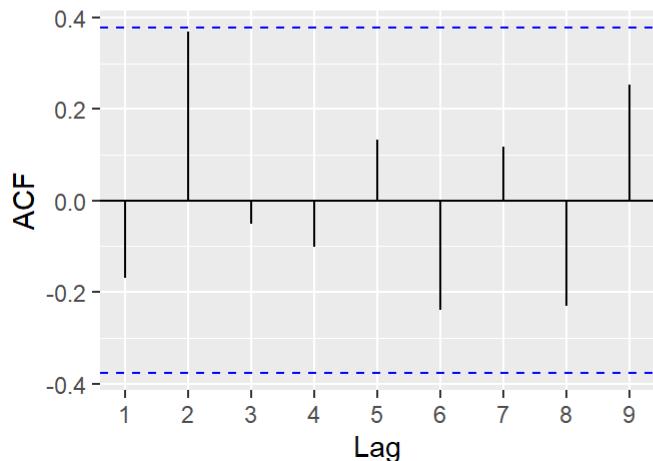
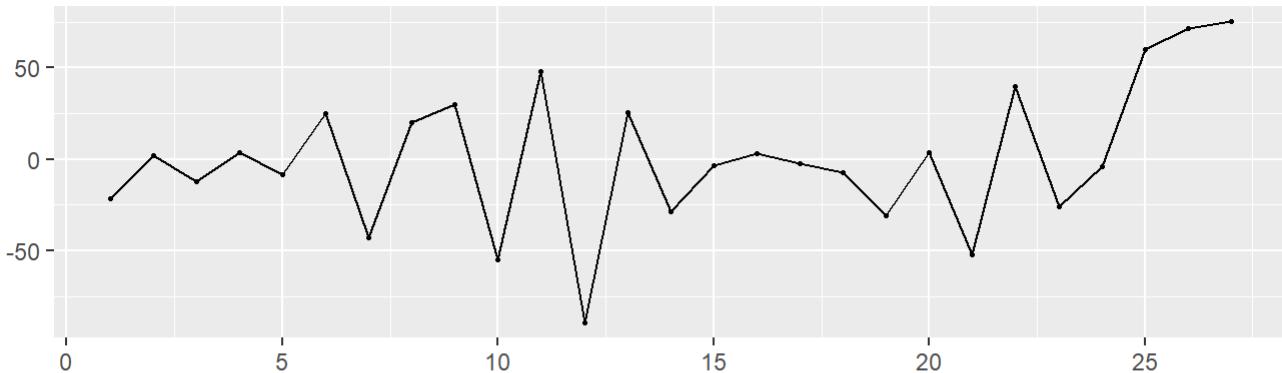
```

## Series: ts_error
## ARIMA(2,0,1) with non-zero mean
##
## Coefficients:
##             ar1      ar2      ma1      mean
##             1.7184 -0.8122  0.0725   65.3274
## s.e.    0.2010  0.1902  0.2237 100.0418
## 
## sigma^2 estimated as 1727: log likelihood=-139.12
## AIC=288.25   AICc=291.11   BIC=294.73
## 
## Training set error measures:
##               ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 0.9605484 38.35349 29.30842 23.75439 59.65693 0.8049584 -0.1692383

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(2,0,1) with non-zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,0,1) with non-zero mean  
## Q* = 8.8461, df = 3, p-value = 0.03141  
##  
## Model df: 4. Total lags used: 7
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

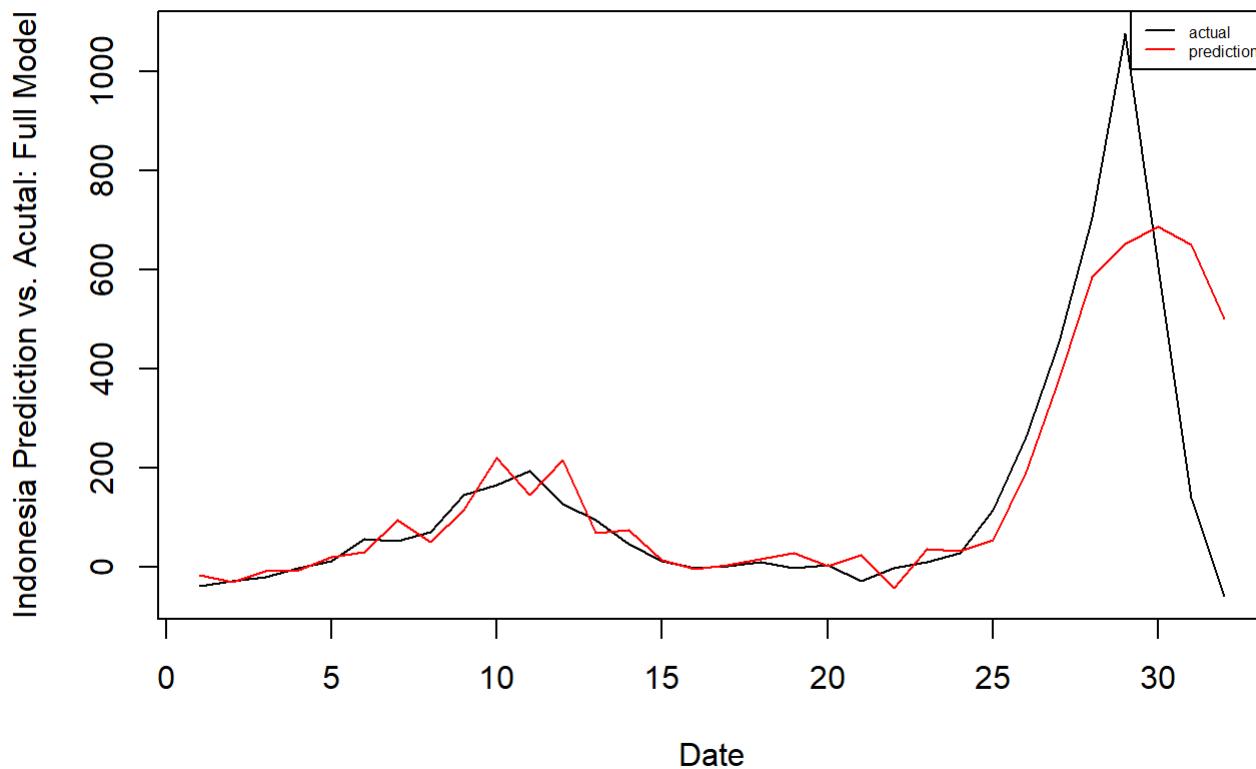
regression_prediction = predict(regression_model, newdata = new_ts_covid_indonesia[28:32,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

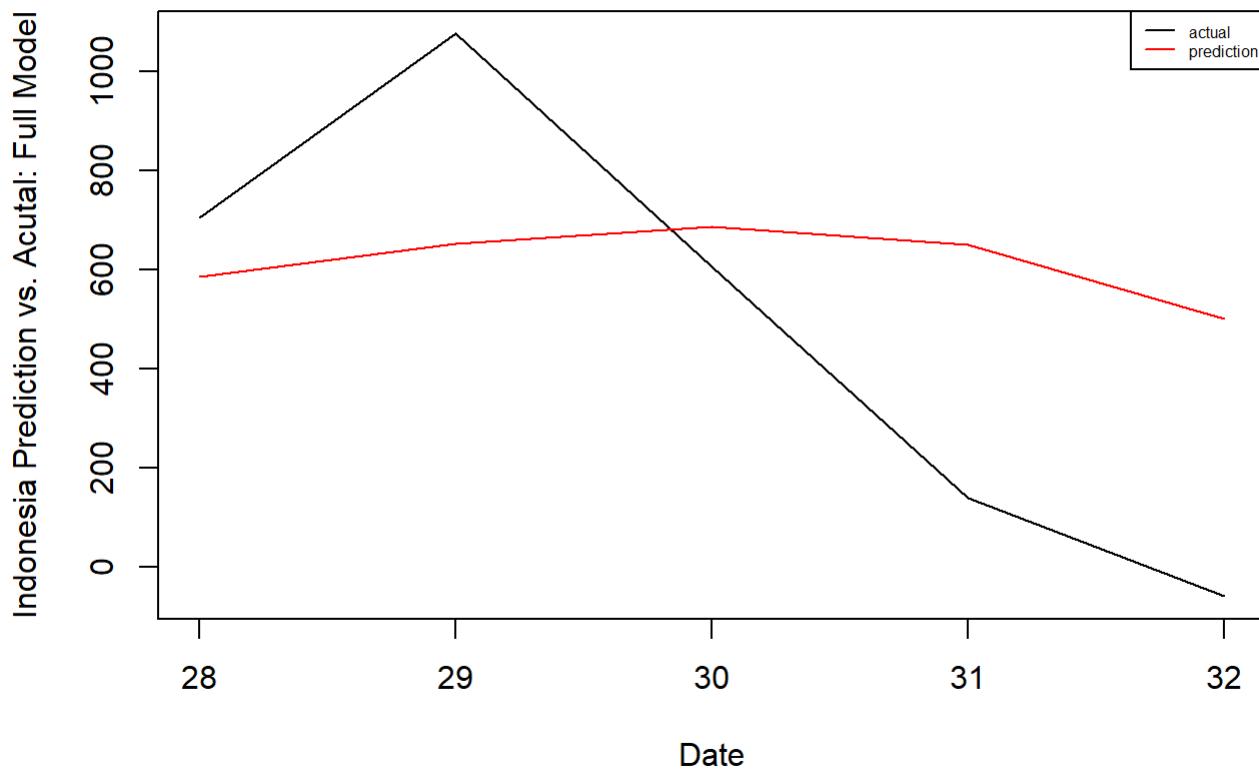
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_indonesia$weekly_aggregated_residuals[1:32])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_indonesia$week[1:32],  
      new_ts_covid_indonesia$weekly_aggregated_residuals[1:32],  
      type = "l",  
      xlab = "Date",  
      ylab = "Indonesia Prediction vs. Actual: Full Model")  
lines(new_ts_covid_indonesia$week[1:32],  
      compare_table$predict,  
      type = "l",  
      col = 'red')  
legend(x = "topright",legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_indonesia$week[28:32],  
     new_ts_covid_indonesia$weekly_aggregated_residuals[28:32],  
     type = "l",  
     xlab = "Date",  
     ylab = "Indonesia Prediction vs. Acutal: Full Model")  
lines(new_ts_covid_indonesia$week[28:32],  
      final_prediction_2,  
      type = "l",  
      col = "red")  
legend(x = "topright", legend=c("actual", "prediction"),  
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Indonesia Full
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 1470.991
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 155289.7
```

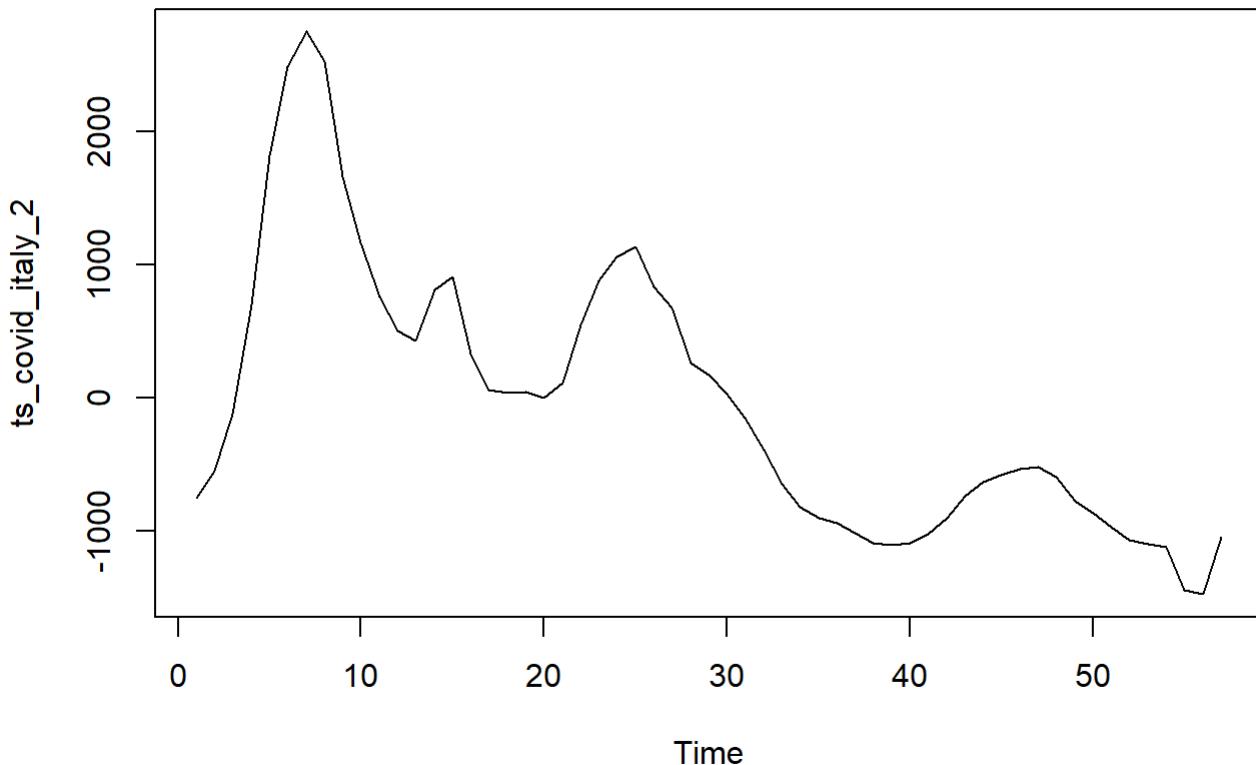
```
#Italy Example on aggregate weekly
ts_covid_italy = ts(covid_italy$residuals,frequency = 7)
ts_covid_italy_2 = ts(colSums(matrix(ts_covid_italy, nrow=7)))
```

```
## Warning in matrix(ts_covid_italy, nrow = 7): data length [395] is not a sub-
## multiple or multiple of the number of rows [7]
```

```
vaccine_policy = colSums(matrix(as.numeric(covid_italy$vaccination_policy), nrow=7))
```

```
## Warning in matrix(as.numeric(covid_italy$vaccination_policy), nrow = 7): data  
## length [395] is not a sub-multiple or multiple of the number of rows [7]
```

```
new_ts_covid_italy = data.frame(1:57,ts_covid_italy_2,vaccine_policy)  
names(new_ts_covid_italy) = c("week","weekly_aggregated_residuals","vaccination_policy")  
  
plot.ts(ts_covid_italy_2)
```



```
regression_model = lm(weekly_aggregated_residuals~1, data = new_ts_covid_italy[1:52,])  
summary(regression_model)
```

```

## 
## Call:
## lm(formula = weekly_aggregated_residuals ~ 1, data = new_ts_covid_italy[1:52,
##   ])
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -1178.8 -835.1 -61.7  636.4 2672.3 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  72.94     138.77   0.526   0.601    
## 
## Residual standard error: 1001 on 51 degrees of freedom

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: weekly_aggregated_residuals
##           Df  Sum Sq Mean Sq F value Pr(>F)    
## Residuals 51 51069523 1001363

```

```

error = residuals(regression_model)
ts_error = ts(error)

#ARIMA on error

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)

```

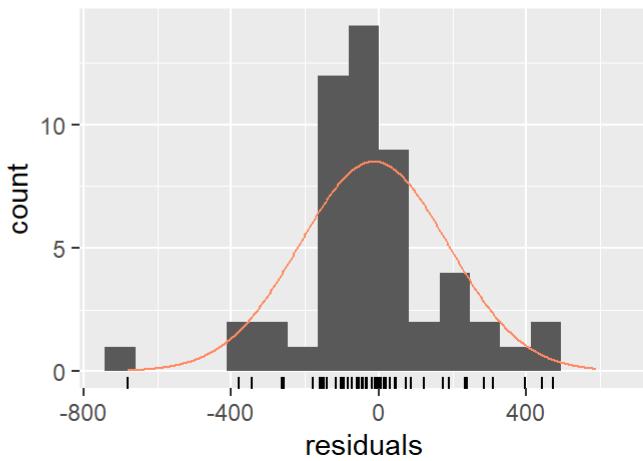
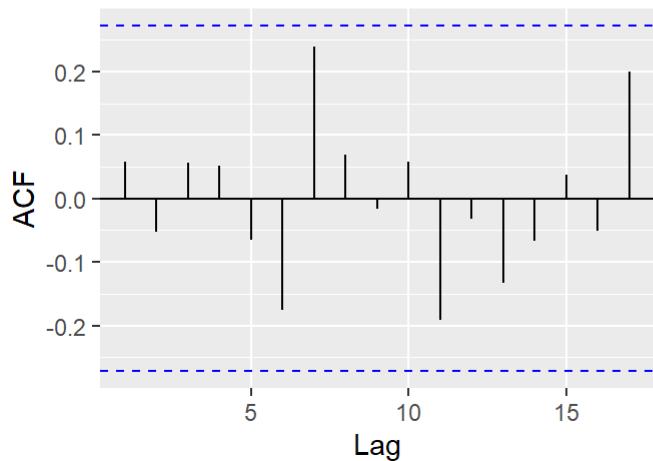
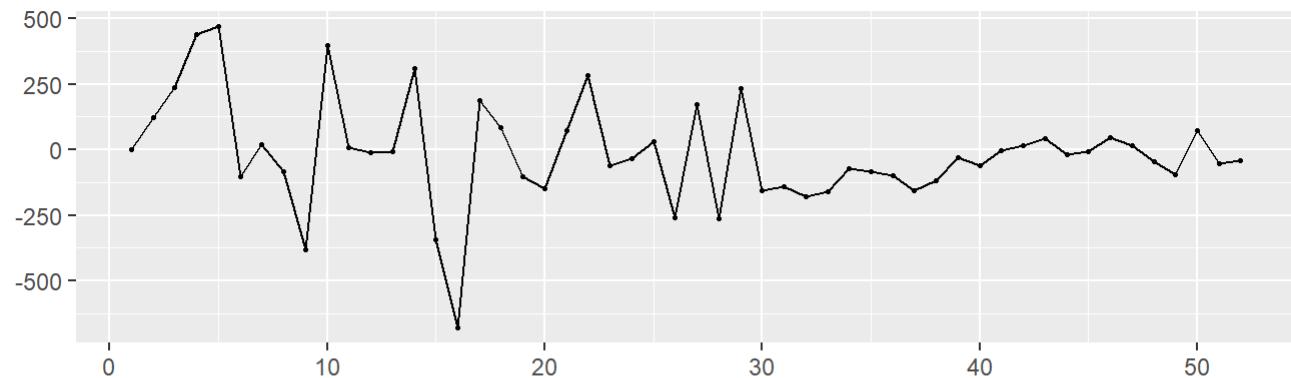
```

## Series: ts_error
## ARIMA(2,1,1)
## 
## Coefficients:
##             ar1      ar2      ma1
##            1.5740 -0.7901 -0.7701
## s.e.  0.1158  0.0932  0.1579
## 
## sigma^2 estimated as 43133:  log likelihood=-343.77
## AIC=695.53  AICc=696.4  BIC=703.26
## 
## Training set error measures:
##               ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set -14.05871 199.5375 139.8977 -13.4275 81.6519 0.6079437 0.05768143

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(2,1,1)



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,1,1)  
## Q* = 6.9628, df = 7, p-value = 0.4328  
##  
## Model df: 3. Total lags used: 10
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

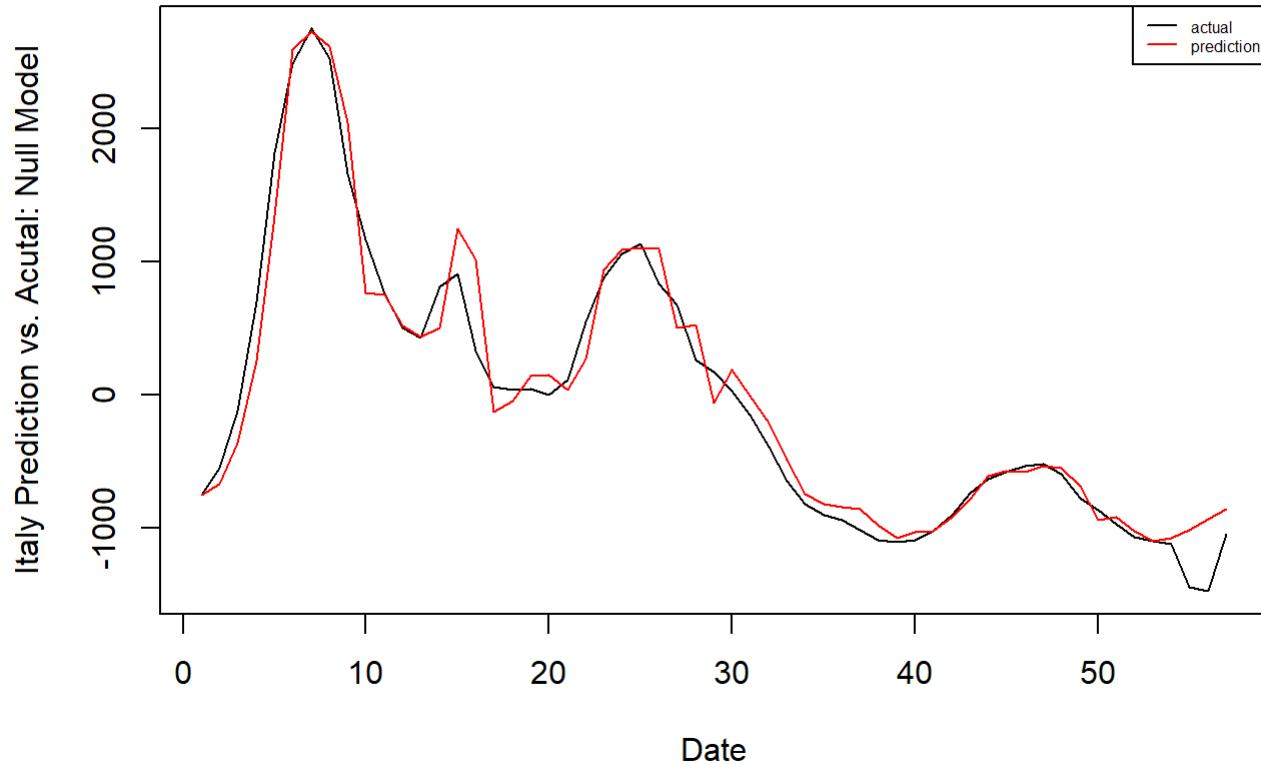
regression_prediction = predict(regression_model, newdata = new_ts_covid_italy[53:57,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

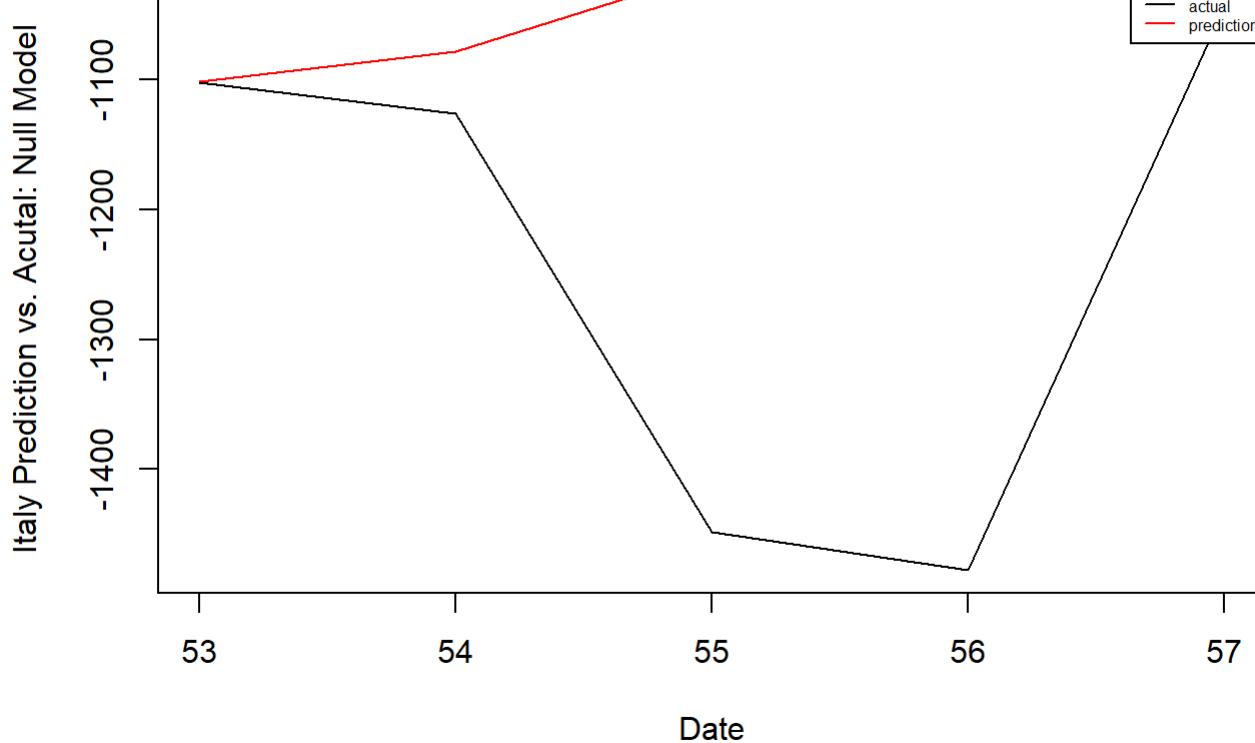
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_italy$weekly_aggregated_residuals[1:57])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_italy$week[1:57],                                     # Draw first time series
      new_ts_covid_italy$weekly_aggregated_residuals[1:57],
      type = "l",
      xlab = "Date",
      ylab = "Italy Prediction vs. Actual: Null Model")
lines(new_ts_covid_italy$week[1:57],                                      # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_italy$week[53:57],                                     # Draw first time series
     new_ts_covid_italy$weekly_aggregated_residuals[53:57],
     type = "l",
     xlab = "Date",
     ylab = "Italy Prediction vs. Acutal: Null Model")
lines(new_ts_covid_italy$week[53:57],                                       # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Italy Null
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 39815.22
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 102746.2
```

```
regression_model = lm(weekly_aggregated_residuals~vaccination_policy, data = new_ts_covid_italy[1:52,])
summary(regression_model)
```

```

## 
## Call:
## lm(formula = weekly_aggregated_residuals ~ vaccination_policy,
##     data = new_ts_covid_italy[1:52, ])
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1896.14  -332.82   -39.44  349.21 1605.13 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 1546.915   194.672   7.946 2.01e-10 ***
## vaccination_policy -58.110      6.816  -8.525 2.59e-11 ***
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 645.2 on 50 degrees of freedom
## Multiple R-squared:  0.5924, Adjusted R-squared:  0.5843 
## F-statistic: 72.68 on 1 and 50 DF,  p-value: 2.589e-11

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: weekly_aggregated_residuals
##              Df  Sum Sq Mean Sq F value Pr(>F)    
## vaccination_policy 1 30255775 30255775 72.682 2.589e-11 ***
## Residuals         50 20813748  416275                
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

error = residuals(regression_model)
ts_error = ts(error)

#ARIMA on error

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)

```

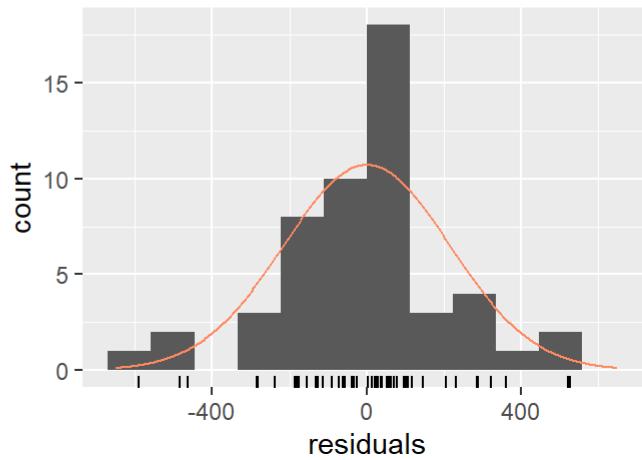
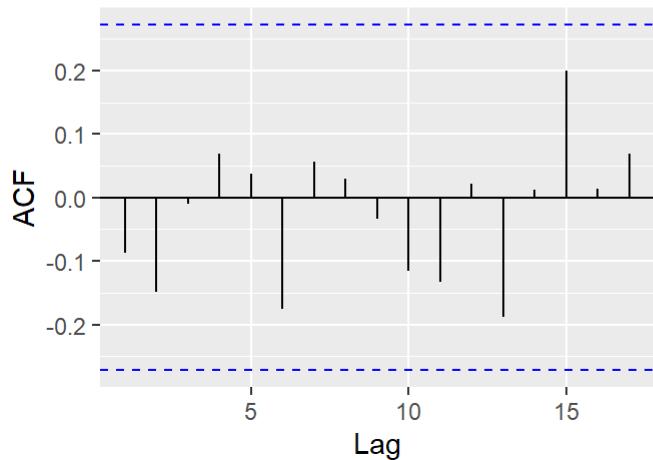
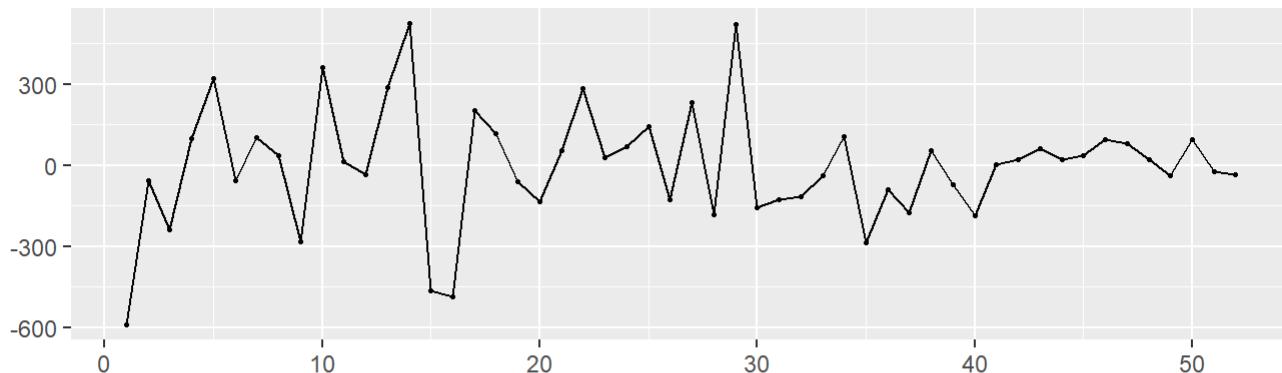
```

## Series: ts_error
## ARIMA(2,0,0) with zero mean
##
## Coefficients:
##      ar1     ar2
##      1.5276 -0.8196
## s.e.  0.0815  0.0852
##
## sigma^2 estimated as 47749: log likelihood=-354.61
## AIC=715.21   AICc=715.71   BIC=721.06
##
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -0.2749928 214.2711 155.0454 -485.8878 644.355 0.6348906
##          ACF1
## Training set -0.08708702

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(2,0,0) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,0,0) with zero mean  
## Q* = 5.1099, df = 8, p-value = 0.7458  
##  
## Model df: 2. Total lags used: 10
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

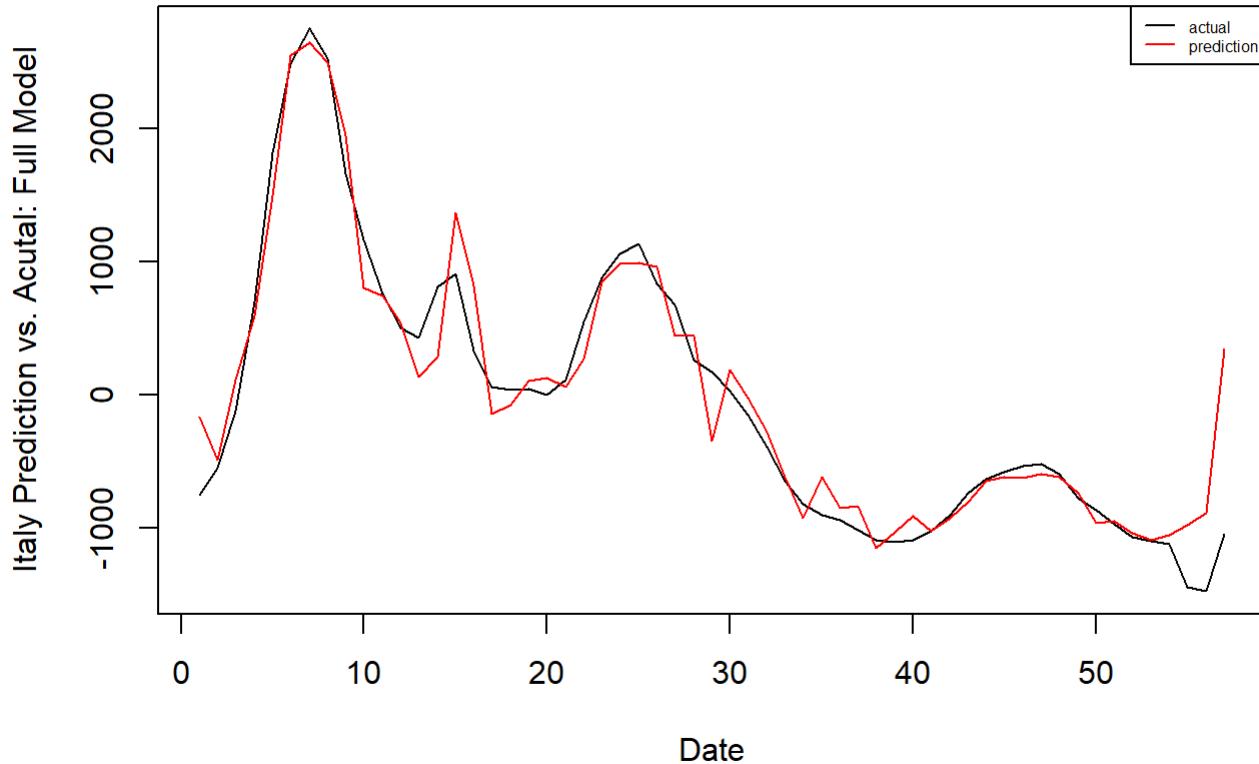
regression_prediction = predict(regression_model, newdata = new_ts_covid_italy[53:57,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

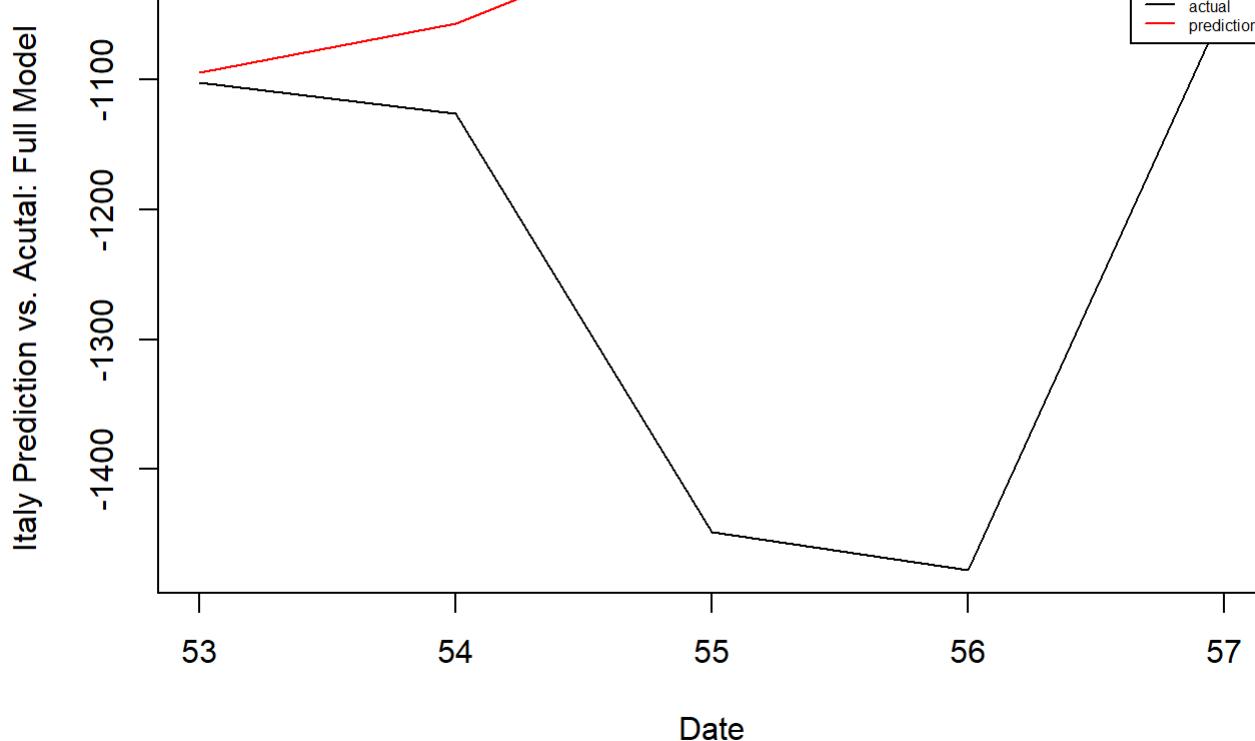
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_italy$weekly_aggregated_residuals[1:57])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_italy$week[1:57],  
      new_ts_covid_italy$weekly_aggregated_residuals[1:57],  
      type = "l",  
      xlab = "Date",  
      ylab = "Italy Prediction vs. Actual: Full Model")  
lines(new_ts_covid_italy$week[1:57],  
      compare_table$predict,  
      type = "l",  
      col = 'red')  
legend(x = "topright",legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_italy$week[53:57],                                     # Draw first time series
     new_ts_covid_italy$weekly_aggregated_residuals[53:57],
     type = "l",
     xlab = "Date",
     ylab = "Italy Prediction vs. Acutal: Full Model")
lines(new_ts_covid_italy$week[53:57],                                       # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Italy Full
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 45912.11
```

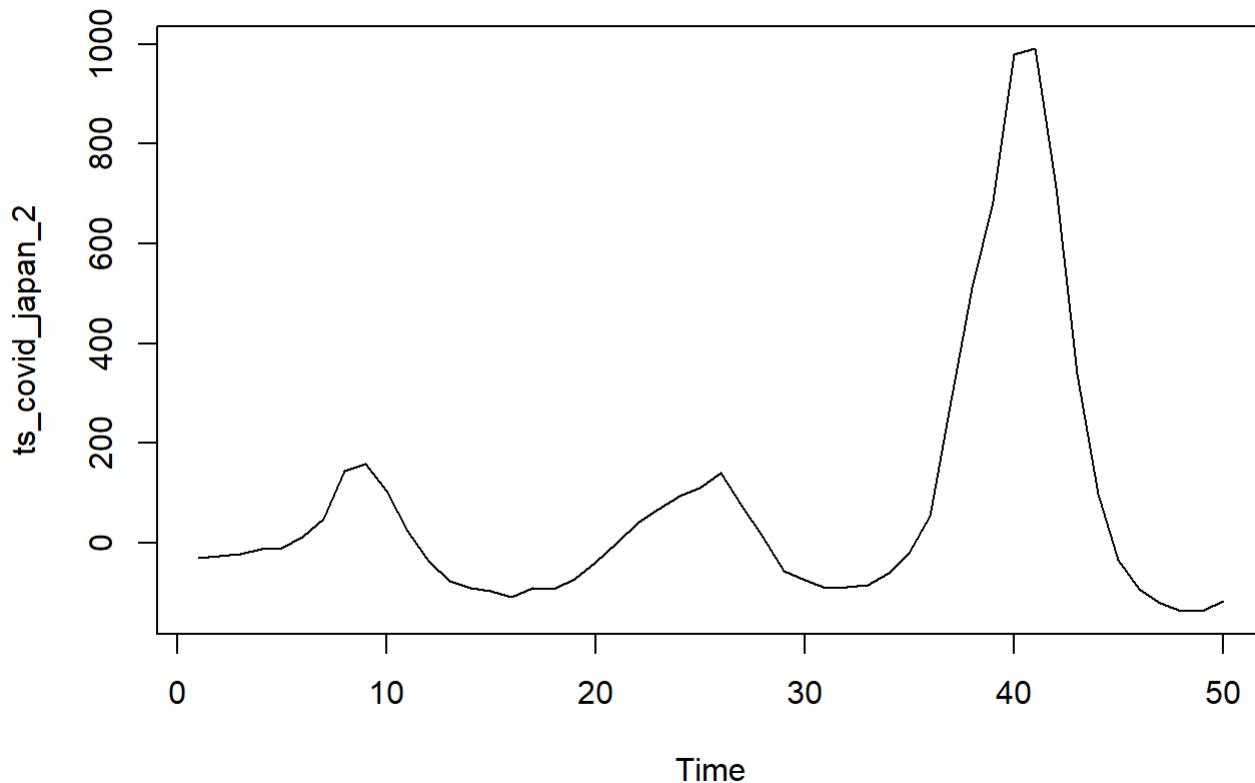
```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 500346.4
```

```
#Japan Example on aggregate weekly
ts_covid_japan = ts(covid_japan$residuals,frequency = 7)
ts_covid_japan_2 = ts(colSums(matrix(ts_covid_japan, nrow=7)))

vaccine_policy = colSums(matrix(as.numeric(covid_japan$vaccination_policy), nrow=7))
new_ts_covid_japan = data.frame(1:50,ts_covid_japan_2,vaccine_policy)
names(new_ts_covid_japan) = c("week","weekly_aggregated_residuals","vaccination_policy")

plot.ts(ts_covid_japan_2)
```



```
regression_model = lm(weekly_aggregated_residuals~1, data = new_ts_covid_japan[1:45,])
summary(regression_model)
```

```

## 
## Call:
## lm(formula = weekly_aggregated_residuals ~ 1, data = new_ts_covid_japan[1:45,
##   ])
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -206.13 -157.80 - 97.81   7.03 893.42 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 97.32      40.04   2.431   0.0192 *  
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 268.6 on 44 degrees of freedom

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: weekly_aggregated_residuals
##           Df Sum Sq Mean Sq F value Pr(>F)    
## Residuals 44 3174188   72141

```

```

error = residuals(regression_model)
ts_error = ts(error)

```

```
#ARIMA on error
```

```

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)

```

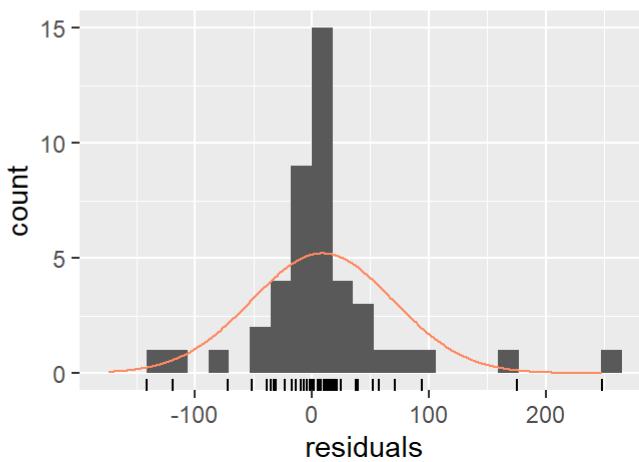
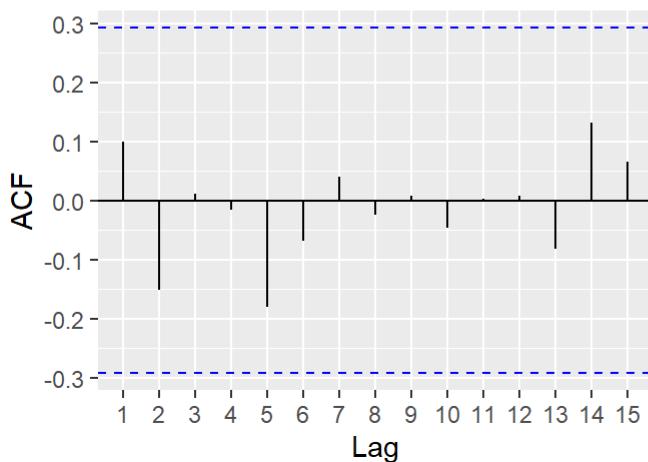
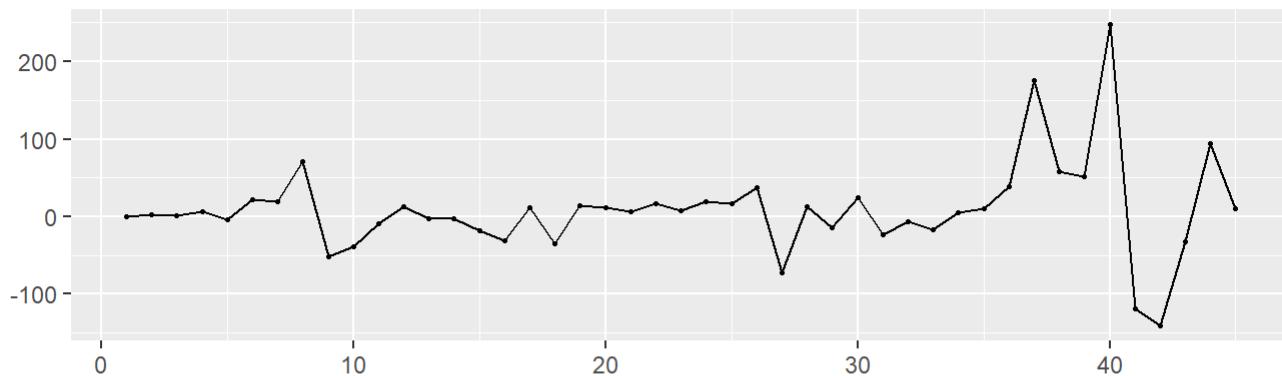
```

## Series: ts_error
## ARIMA(2,1,1)
## 
## Coefficients:
##         ar1      ar2      ma1
##       1.5850 -0.8371 -0.7656
## s.e. 0.1009  0.0792  0.1829
## 
## sigma^2 estimated as 4074: log likelihood=-244.8
## AIC=497.6   AICc=498.62   BIC=504.73
## 
## Training set error measures:
##               ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 8.695904 60.92537 36.06221 460.4996 527.1406 0.5205244 0.09901281

```

```
checkresiduals(error_model1)
```

Residuals from ARIMA(2,1,1)



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,1,1)  
## Q* = 3.7289, df = 6, p-value = 0.7133  
##  
## Model df: 3. Total lags used: 9
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

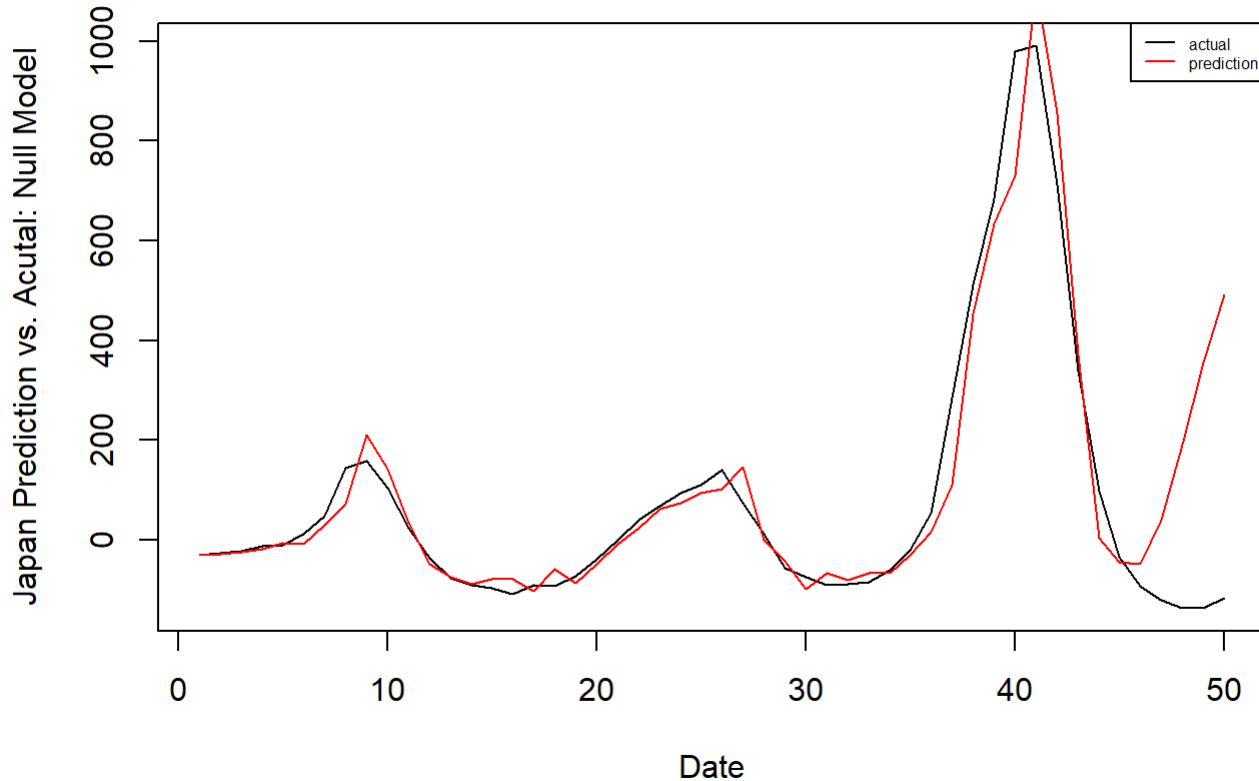
regression_prediction = predict(regression_model, newdata = new_ts_covid_japan[46:50,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

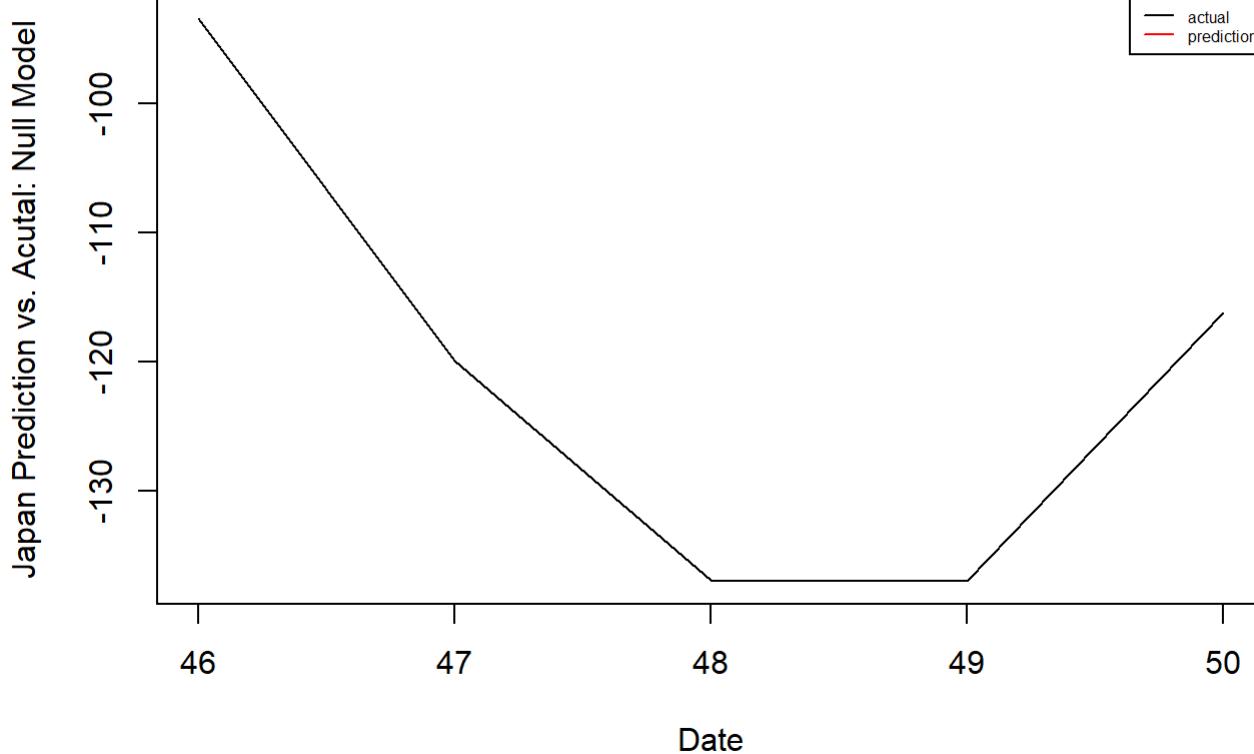
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_japan$weekly_aggregated_residuals[1:50])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_japan$week[1:50],                                     # Draw first time series
      new_ts_covid_japan$weekly_aggregated_residuals[1:50],
      type = "l",
      xlab = "Date",
      ylab = "Japan Prediction vs. Acutal: Null Model")
lines(new_ts_covid_japan$week[1:50],                                     # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_japan$week[46:50],                                     # Draw first time series
     new_ts_covid_japan$weekly_aggregated_residuals[46:50],
     type = "l",
     xlab = "Date",
     ylab = "Japan Prediction vs. Acutal: Null Model")
lines(new_ts_covid_japan$week[46:50],                                       # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Japan Null
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 3711.901
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 149276.6
```

```
regression_model = lm(weekly_aggregated_residuals~vaccination_policy, data = new_ts_covid_japan[1:45,])
summary(regression_model)
```

```
##  
## Call:  
## lm(formula = weekly_aggregated_residuals ~ vaccination_policy,  
##      data = new_ts_covid_japan[1:45, ])  
##  
## Residuals:  
##     Min      1Q   Median      3Q     Max  
## -361.80 -125.63  -29.93   57.62  737.40  
##  
## Coefficients:  
##                         Estimate Std. Error t value Pr(>|t|)  
## (Intercept)       -119.549    72.093 -1.658  0.10455  
## vaccination_policy 10.654     3.074   3.466  0.00121 **  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 240.2 on 43 degrees of freedom  
## Multiple R-squared:  0.2184, Adjusted R-squared:  0.2002  
## F-statistic: 12.01 on 1 and 43 DF,  p-value: 0.001211
```

```
anova(regression_model)
```

```
## Analysis of Variance Table  
##  
## Response: weekly_aggregated_residuals  
##                         Df  Sum Sq Mean Sq F value Pr(>F)  
## vaccination_policy  1 693110  693110 12.012 0.001211 **  
## Residuals          43 2481078   57699  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
error = residuals(regression_model)  
ts_error = ts(error)  
  
#ARIMA on error  
  
error_model = auto.arima(ts_error)  
fitted_error = fitted(error_model)  
summary(error_model)
```

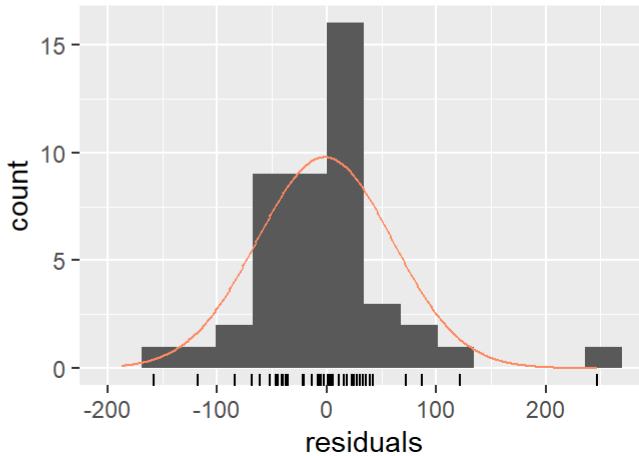
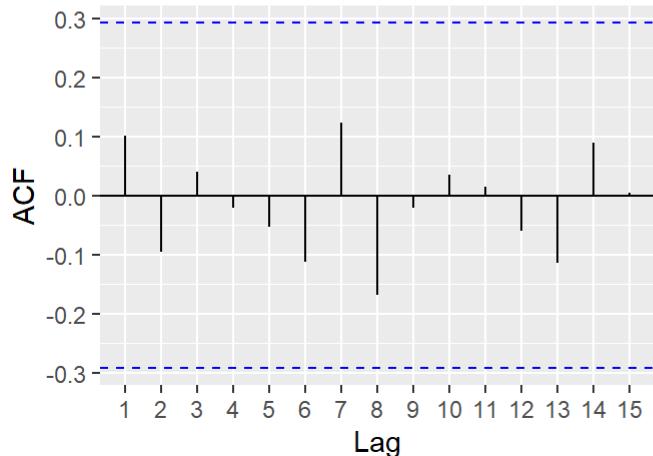
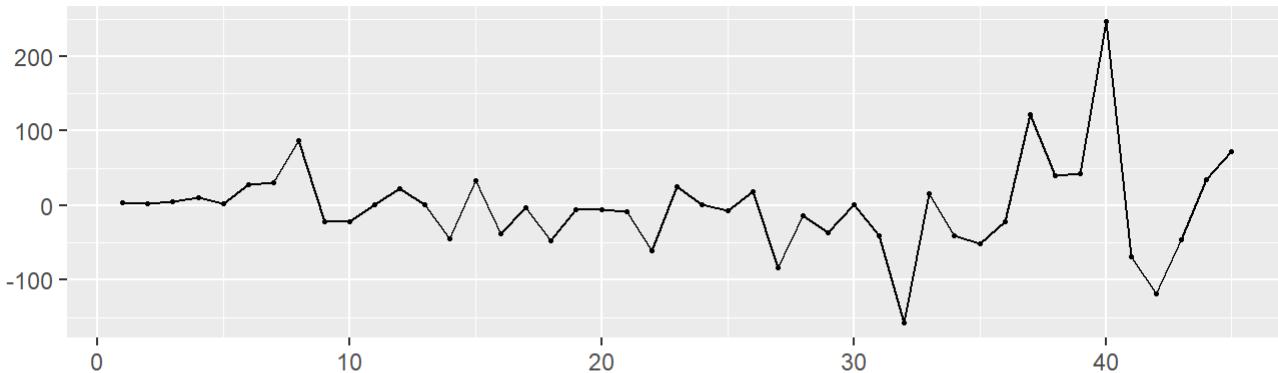
```

## Series: ts_error
## ARIMA(2,0,0) with zero mean
##
## Coefficients:
##      ar1     ar2
##      1.6108 -0.8407
## s.e.  0.0740  0.0727
##
## sigma^2 estimated as 3928: log likelihood=-250.99
## AIC=507.98   AICc=508.57   BIC=513.4
##
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set -1.9505 61.26785 39.68365 28.09882 45.71338 0.5281591 0.1006244

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(2,0,0) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,0,0) with zero mean  
## Q* = 4.3878, df = 7, p-value = 0.7342  
##  
## Model df: 2. Total lags used: 9
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

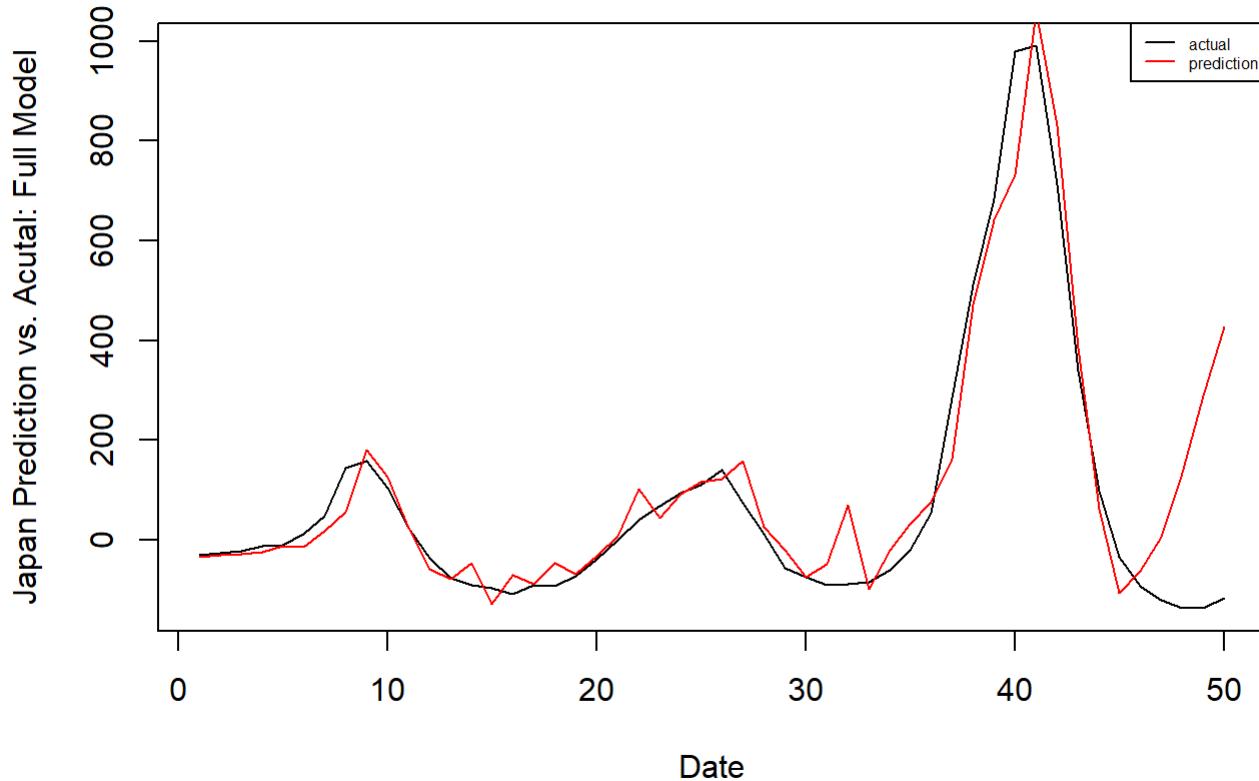
regression_prediction = predict(regression_model, newdata = new_ts_covid_japan[46:50,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

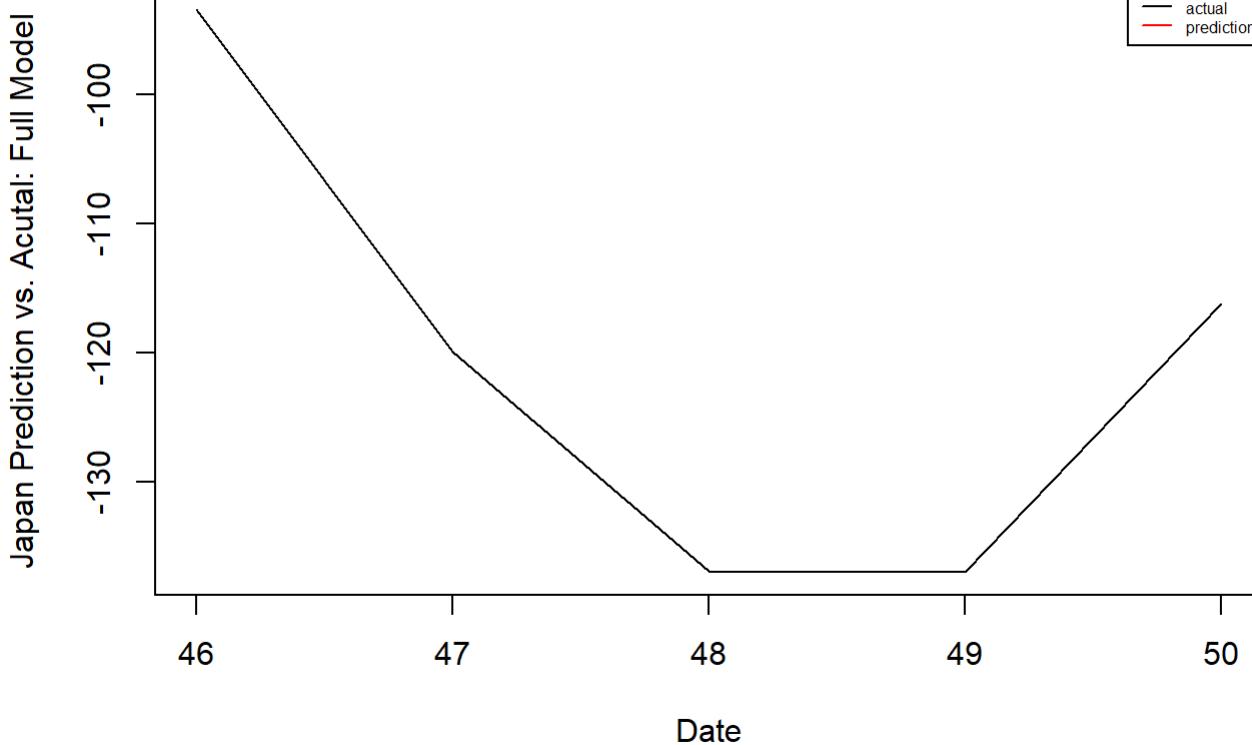
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_japan$weekly_aggregated_residuals[1:50])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_japan$week[1:50],  
      new_ts_covid_japan$weekly_aggregated_residuals[1:50],  
      type = "l",  
      xlab = "Date",  
      ylab = "Japan Prediction vs. Actual: Full Model")  
lines(new_ts_covid_japan$week[1:50],  
      compare_table$predict,  
      type = "l",  
      col = 'red')  
legend(x = "topright",legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_japan$week[46:50],  
      new_ts_covid_japan$weekly_aggregated_residuals[46:50],  
      type = "l",  
      xlab = "Date",  
      ylab = "Japan Prediction vs. Acutal: Full Model")  
lines(new_ts_covid_japan$week[46:50],  
      final_prediction_2,  
      type = "l",  
      col = "red")  
legend(x = "topright", legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Japan Full
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 3753.749
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 113307.6
```

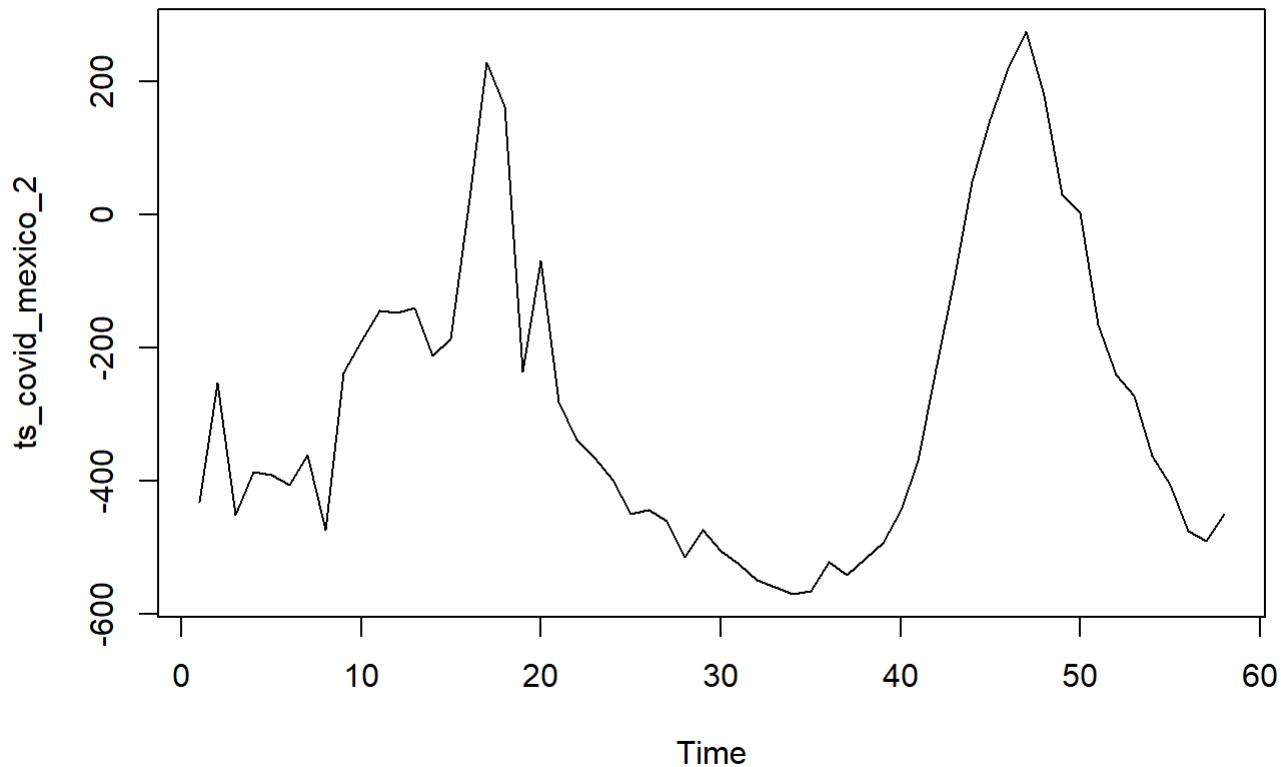
```
#Mexico Example on aggregate weekly
ts_covid_mexico = ts(covid_mexico$residuals,frequency = 7)
ts_covid_mexico_2 = ts(colSums(matrix(ts_covid_mexico, nrow=7)))
```

```
## Warning in matrix(ts_covid_mexico, nrow = 7): data length [401] is not a sub-
## multiple or multiple of the number of rows [7]
```

```
vaccine_policy = colSums(matrix(as.numeric(covid_mexico$vaccination_policy), nrow=7))
```

```
## Warning in matrix(as.numeric(covid_mexico$vaccination_policy), nrow = 7): data  
## length [401] is not a sub-multiple or multiple of the number of rows [7]
```

```
new_ts_covid_mexico = data.frame(1:58,ts_covid_mexico_2,vaccine_policy)  
names(new_ts_covid_mexico) = c("week","weekly_aggregated_residuals","vaccination_policy")  
  
plot.ts(ts_covid_mexico_2)
```



```
regression_model = lm(weekly_aggregated_residuals~1, data = new_ts_covid_mexico[1:53,])  
summary(regression_model)
```

```

## 
## Call:
## lm(formula = weekly_aggregated_residuals ~ 1, data = new_ts_covid_mexico[1:53,
##   ])
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -300.68 -190.97 -68.37 126.40 545.21 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -270.0      32.9   -8.207 6.01e-11 ***
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 239.5 on 52 degrees of freedom

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: weekly_aggregated_residuals
##           Df  Sum Sq Mean Sq F value Pr(>F)    
## Residuals 52 2983244   57370

```

```

error = residuals(regression_model)
ts_error = ts(error)

```

```
#ARIMA on error
```

```

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)

```

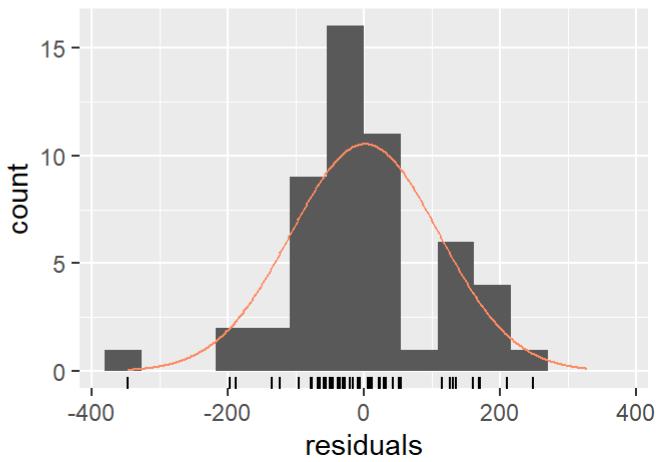
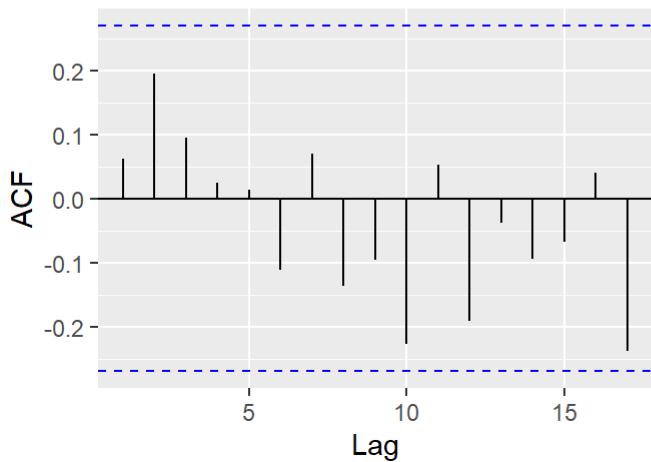
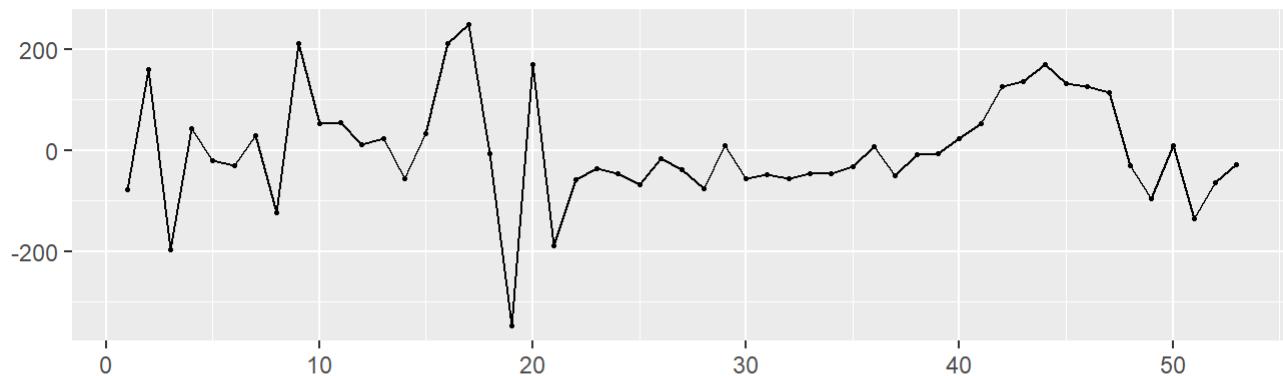
```

## Series: ts_error
## ARIMA(1,0,0) with zero mean
## 
## Coefficients:
##         ar1
##       0.8793
## s.e. 0.0589
## 
## sigma^2 estimated as 11877: log likelihood=-324.07
## AIC=652.15   AICc=652.39   BIC=656.09
## 
## Training set error measures:
##               ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 1.573802 107.9501 79.84595 86.8258 149.0733 1.00417 0.06238871

```

```
checkresiduals(error_model1)
```

Residuals from ARIMA(1,0,0) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(1,0,0) with zero mean  
## Q* = 9.3, df = 9, p-value = 0.4101  
##  
## Model df: 1. Total lags used: 10
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

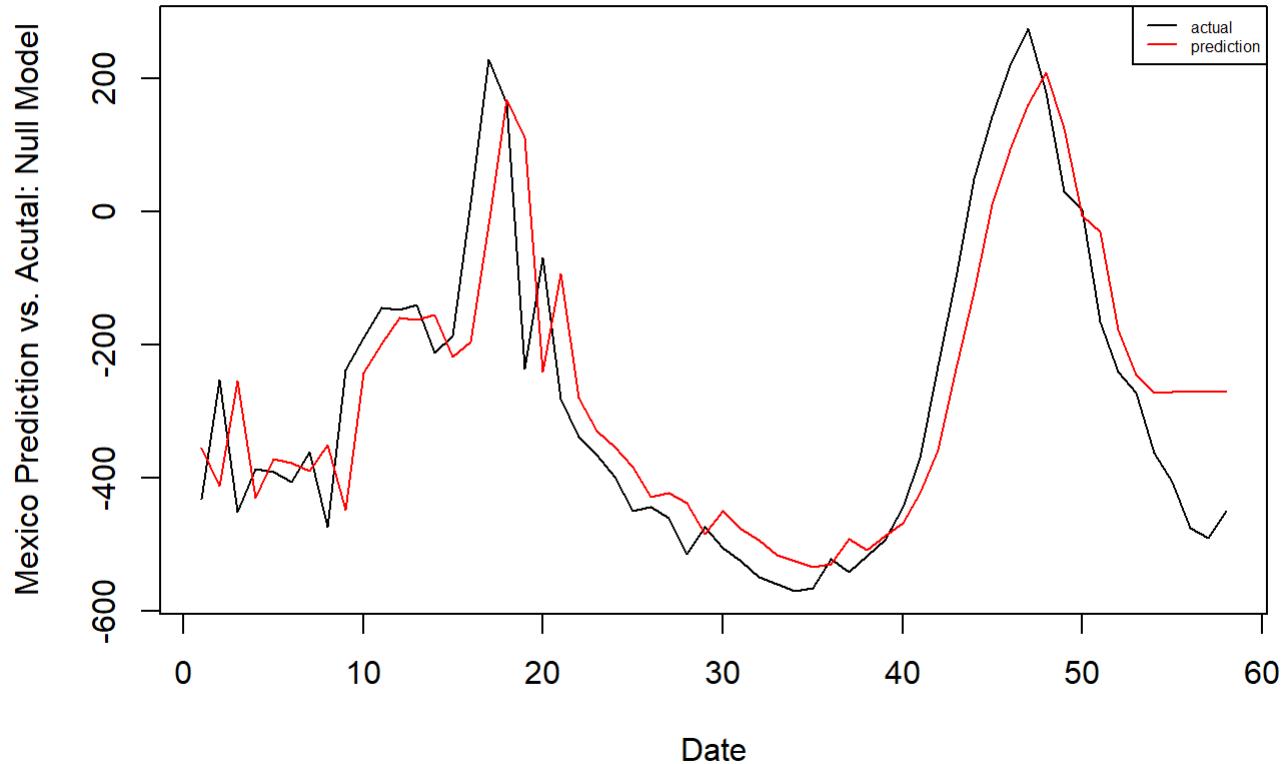
regression_prediction = predict(regression_model, newdata = new_ts_covid_mexico[54:58,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

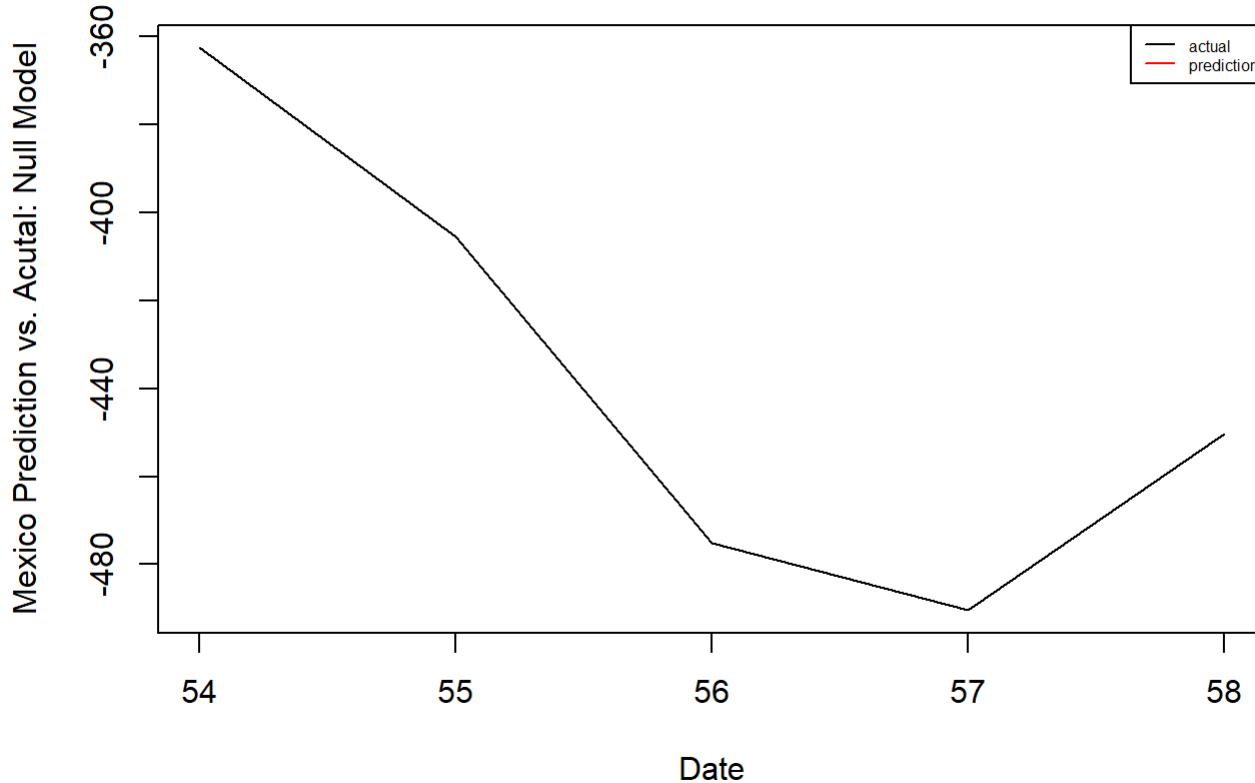
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_mexico$weekly_aggregated_residuals[1:58])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_mexico$week[1:58],                                     # Draw first time series
      new_ts_covid_mexico$weekly_aggregated_residuals[1:58],
      type = "l",
      xlab = "Date",
      ylab = "Mexico Prediction vs. Actual: Null Model")
lines(new_ts_covid_mexico$week[1:58],                                      # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_mexico$week[54:58],                                     # Draw first time series
     new_ts_covid_mexico$weekly_aggregated_residuals[54:58],
     type = "l",
     xlab = "Date",
     ylab = "Mexico Prediction vs. Acutal: Null Model")
lines(new_ts_covid_mexico$week[54:58],                                     # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Mexico Null
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 11653.22
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 29623.64
```

```
regression_model = lm(weekly_aggregated_residuals~vaccination_policy, data = new_ts_covid_mexico[1:53,])
summary(regression_model)
```

```
##  
## Call:  
## lm(formula = weekly_aggregated_residuals ~ vaccination_policy,  
##      data = new_ts_covid_mexico[1:53, ])  
##  
## Residuals:  
##     Min      1Q   Median      3Q     Max  
## -331.58 -183.66 - 38.19 165.91 545.30  
##  
## Coefficients:  
##                         Estimate Std. Error t value Pr(>|t|)  
## (Intercept)       -391.948     66.164 -5.924 2.7e-07 ***  
## vaccination_policy    5.459      2.596   2.103  0.0404 *  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 232 on 51 degrees of freedom  
## Multiple R-squared:  0.0798, Adjusted R-squared:  0.06176  
## F-statistic: 4.423 on 1 and 51 DF,  p-value: 0.04042
```

```
anova(regression_model)
```

```
## Analysis of Variance Table  
##  
## Response: weekly_aggregated_residuals  
##                         Df Sum Sq Mean Sq F value Pr(>F)  
## vaccination_policy  1 238063 238063 4.4227 0.04042 *  
## Residuals          51 2745181 53827  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
error = residuals(regression_model)  
ts_error = ts(error)  
  
#ARIMA on error  
  
error_model = auto.arima(ts_error)  
fitted_error = fitted(error_model)  
summary(error_model)
```

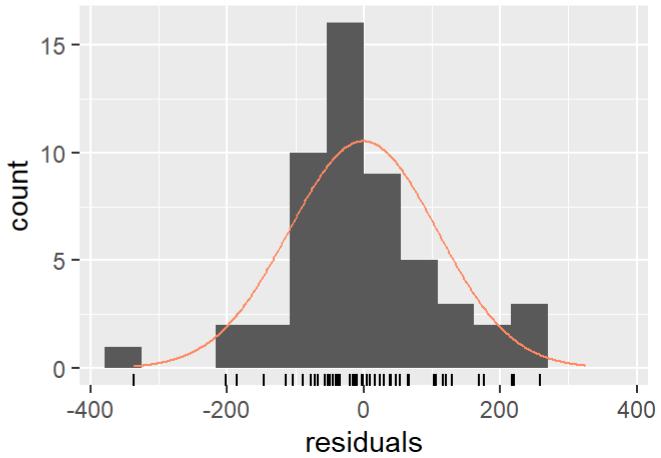
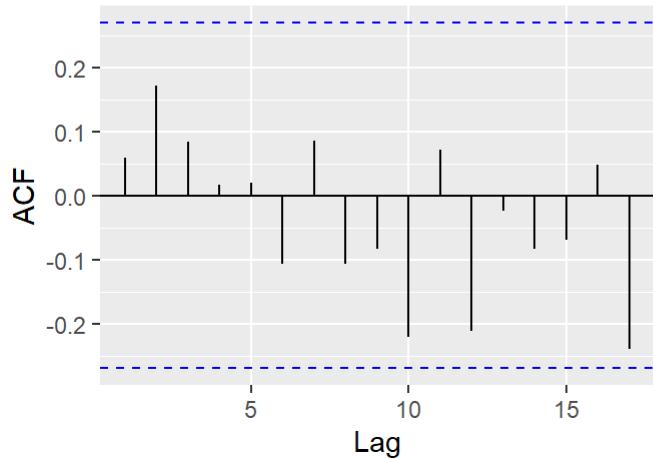
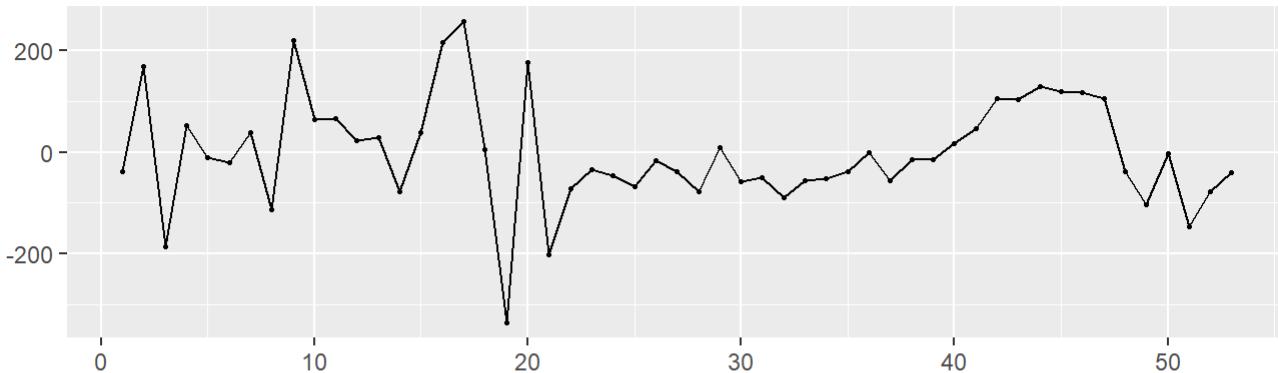
```

## Series: ts_error
## ARIMA(1,0,0) with zero mean
##
## Coefficients:
##      ar1
##      0.8690
## s.e.  0.0615
##
## sigma^2 estimated as 11821: log likelihood=-323.91
## AIC=651.82   AICc=652.06   BIC=655.76
##
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set -1.04045 107.6934 80.89451 83.91661 245.5383 1.010686 0.05919192

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(1,0,0) with zero mean



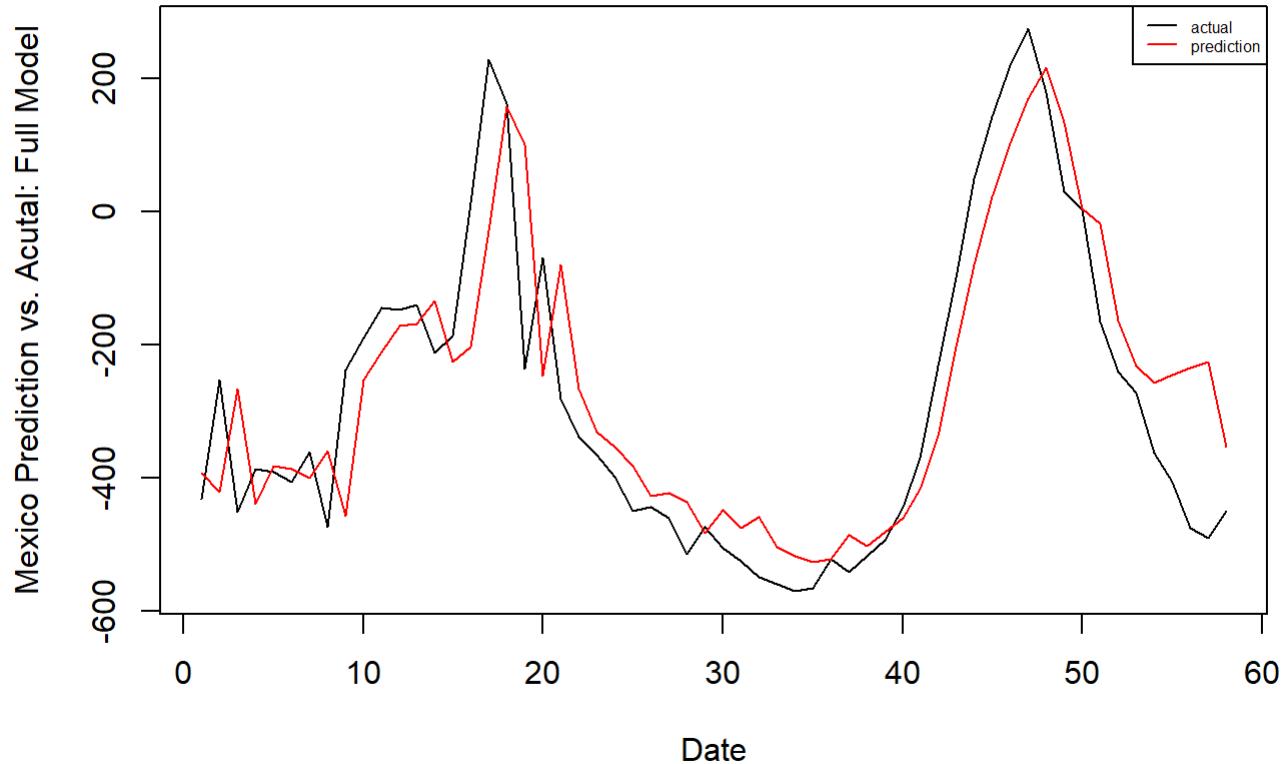
```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(1,0,0) with zero mean  
## Q* = 7.9716, df = 9, p-value = 0.537  
##  
## Model df: 1. Total lags used: 10
```

#goodness of fit

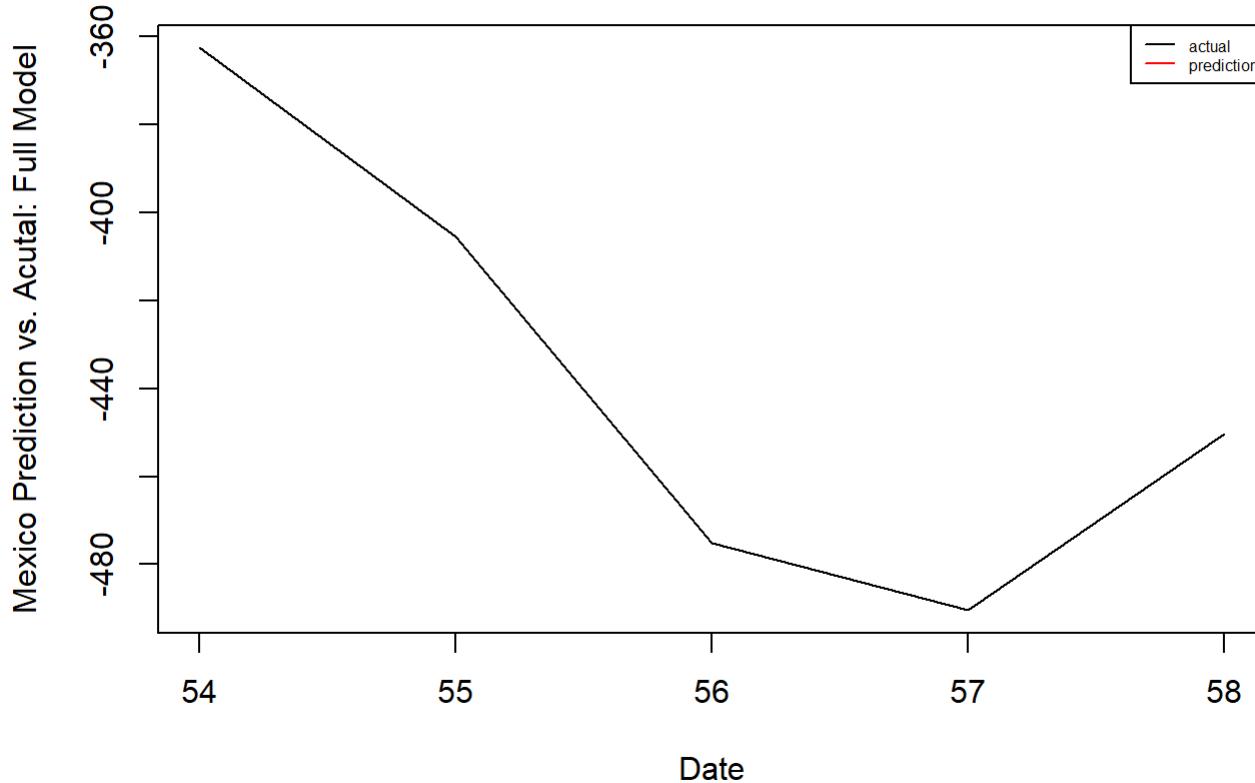
```
regression_prediction = predict(regression_model)  
  
final_prediction = regression_prediction + fitted_error  
  
regression_prediction = predict(regression_model, newdata = new_ts_covid_mexico[54:58,])  
error_forecast = predict(error_model, n.ahead = 5)$pred  
  
final_prediction_2 = regression_prediction + error_forecast
```

#compare prediction and actual

```
compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_mexico$weekly_aggregated_residuals[1:58])  
colnames(compare_table) = c("predict","actual")  
  
plot(new_ts_covid_mexico$week[1:58],  
      new_ts_covid_mexico$weekly_aggregated_residuals[1:58],  
      type = "l",  
      xlab = "Date",  
      ylab = "Mexico Prediction vs. Actual: Full Model")  
lines(new_ts_covid_mexico$week[1:58],  
      compare_table$predict,  
      type = "l",  
      col = 'red')  
legend(x = "topright",legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_mexico$week[54:58],                                     # Draw first time series
     new_ts_covid_mexico$weekly_aggregated_residuals[54:58],
     type = "l",
     xlab = "Date",
     ylab = "Mexico Prediction vs. Acutal: Full Model")
lines(new_ts_covid_mexico$week[54:58],                                     # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Mexico Full
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 11597.87
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 34939.01
```

```
#Morocco Example on aggregate weekly
ts_covid_morocco = ts(covid_morocco$residuals,frequency = 7)
ts_covid_morocco_2 = ts(colSums(matrix(ts_covid_morocco, nrow=7)))
```

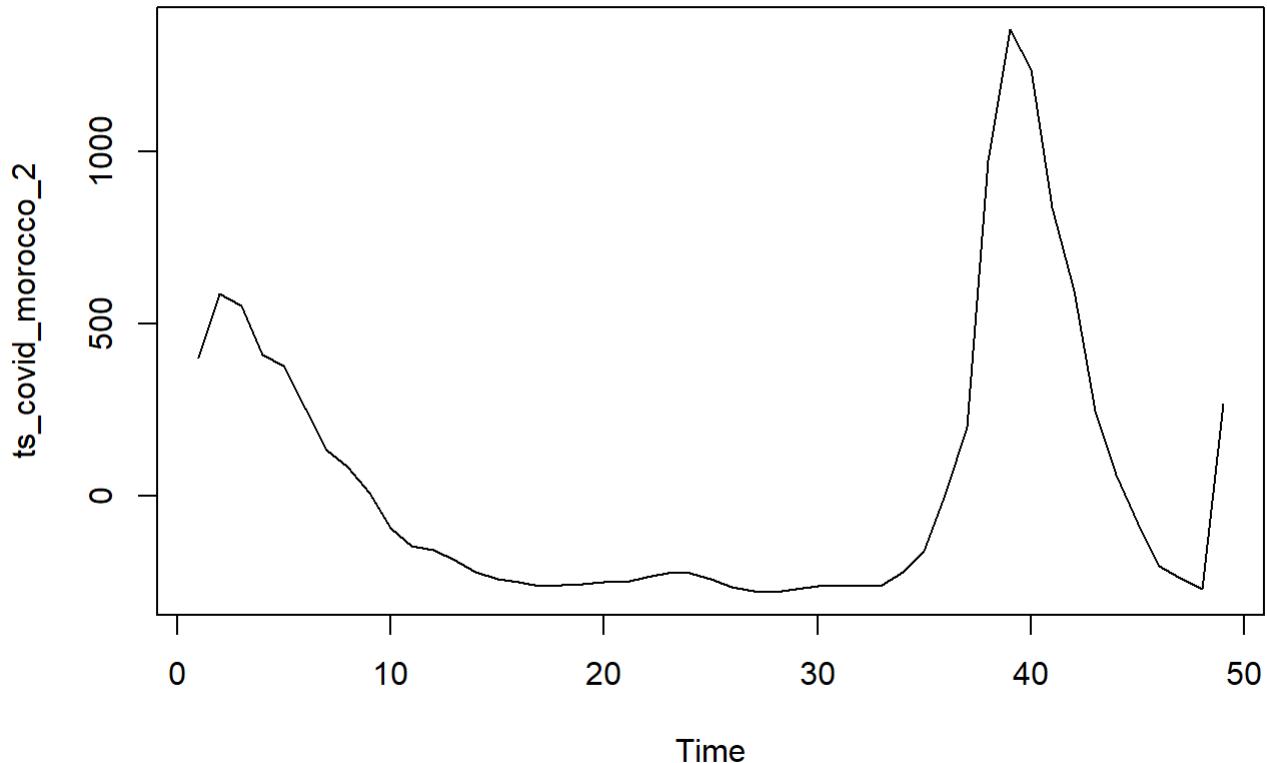
```
## Warning in matrix(ts_covid_morocco, nrow = 7): data length [337] is not a sub-
## multiple or multiple of the number of rows [7]
```

```
vaccine_policy = colSums(matrix(as.numeric(covid_morocco$vaccination_policy), nrow=7))
```

```
## Warning in matrix(as.numeric(covid_morocco$vaccination_policy), nrow = 7): data
## length [337] is not a sub-multiple or multiple of the number of rows [7]
```

```
new_ts_covid_morocco = data.frame(1:49,ts_covid_morocco_2,vaccine_policy)
names(new_ts_covid_morocco) = c("week","weekly_aggregated_residuals","vaccination_policy")

plot.ts(ts_covid_morocco_2)
```



```
regression_model = lm(weekly_aggregated_residuals~1, data = new_ts_covid_morocco[1:44,])
summary(regression_model)
```

```

## 
## Call:
## lm(formula = weekly_aggregated_residuals ~ 1, data = new_ts_covid_morocco[1:44,
##   ])
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -332.8 -305.3 -225.4  196.1 1302.0 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  52.56     65.12   0.807   0.424    
## 
## Residual standard error: 432 on 43 degrees of freedom

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: weekly_aggregated_residuals
##           Df  Sum Sq Mean Sq F value Pr(>F)    
## Residuals 43 8023558 186594

```

```

error = residuals(regression_model)
ts_error = ts(error)

#ARIMA on error

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)

```

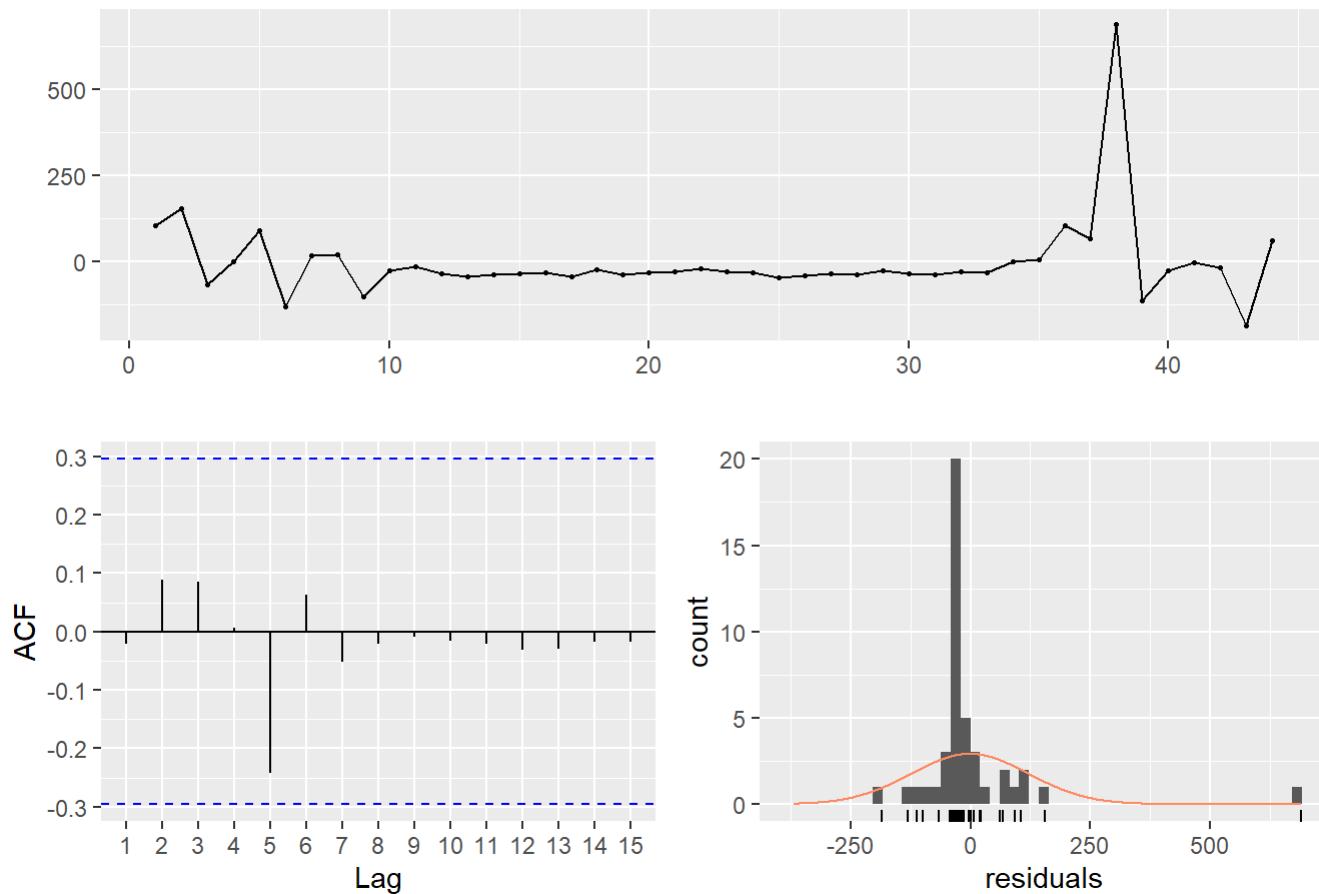
```

## Series: ts_error
## ARIMA(1,0,2) with zero mean
## 
## Coefficients:
##         ar1      ma1      ma2
##       0.7351  1.0147  0.5313
## s.e.  0.1146  0.1963  0.1701
## 
## sigma^2 estimated as 15732:  log likelihood=-275.19
## AIC=558.39  AICc=559.41  BIC=565.52
## 
## Training set error measures:
##               ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set -1.741959 121.0742 62.01379 28.1465 48.32618 0.6566521 -0.02048108

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(1,0,2) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(1,0,2) with zero mean  
## Q* = 4.1816, df = 6, p-value = 0.6521  
##  
## Model df: 3. Total lags used: 9
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

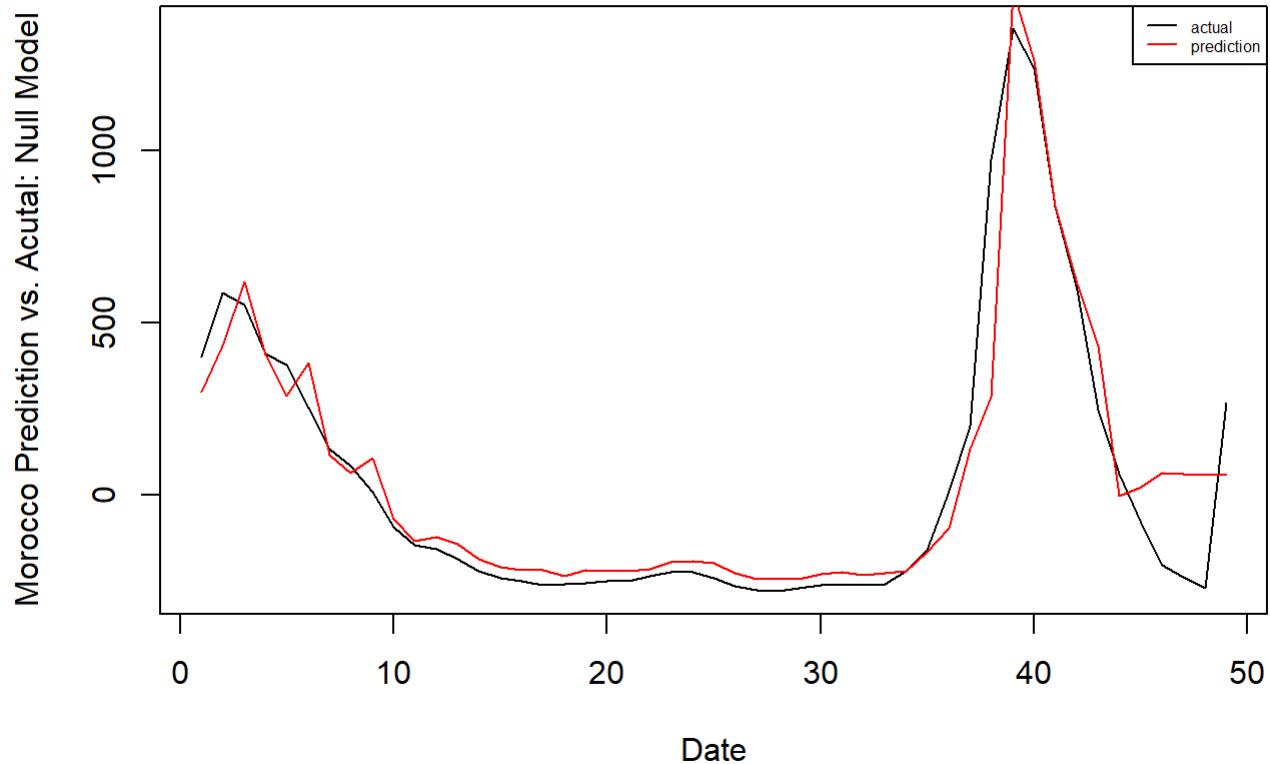
regression_prediction = predict(regression_model, newdata = new_ts_covid_morocco[45:49,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

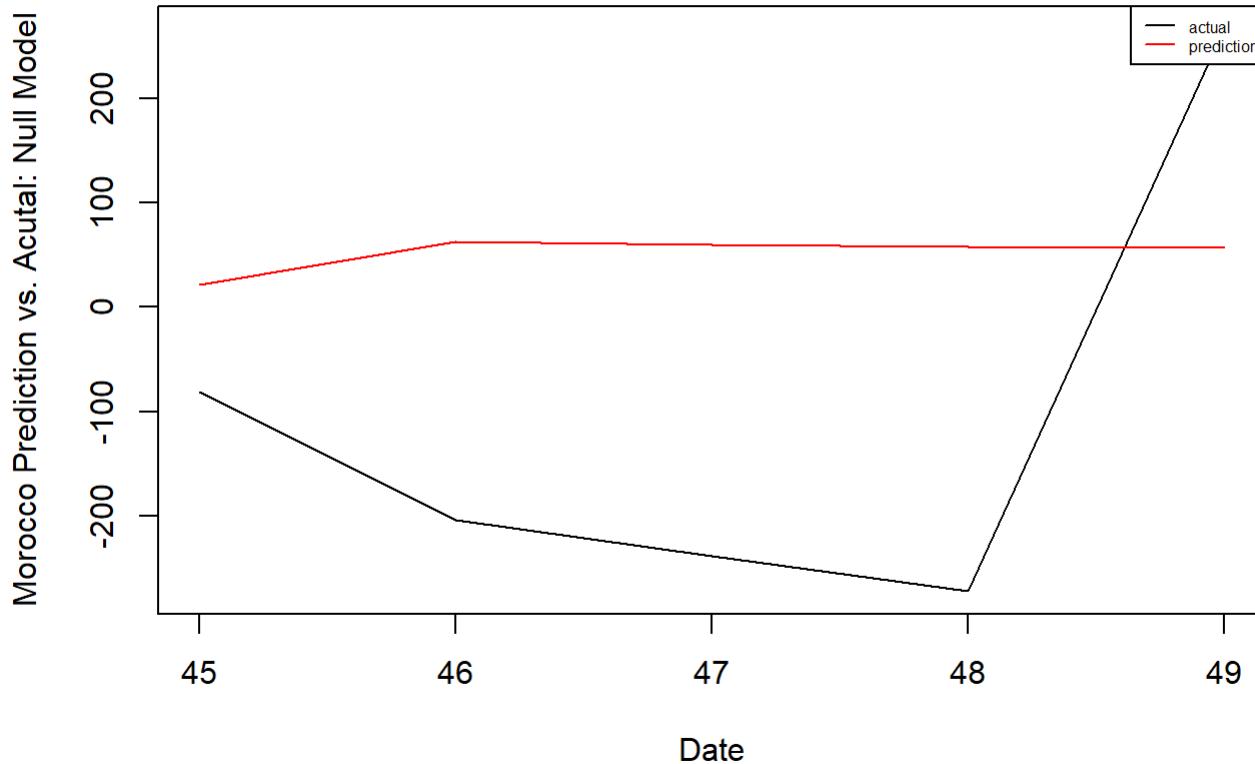
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_morocco$weekly_aggregated_residuals[1:49])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_morocco$week[1:49],                                     # Draw first time series
      new_ts_covid_morocco$weekly_aggregated_residuals[1:49],
      type = "l",
      xlab = "Date",
      ylab = "Morocco Prediction vs. Acutal: Null Model")
lines(new_ts_covid_morocco$week[1:49],                                     # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_morocco$week[45:49],  
      new_ts_covid_morocco$weekly_aggregated_residuals[45:49],  
      type = "l",  
      xlab = "Date",  
      ylab = "Morocco Prediction vs. Acutal: Null Model")  
lines(new_ts_covid_morocco$week[45:49],  
      final_prediction_2,  
      type = "l",  
      col = "red")  
legend(x = "topright", legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Morocco Null
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 14658.97
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 64730.88
```

```
regression_model = lm(weekly_aggregated_residuals~vaccination_policy, data = new_ts_covid_morocco[1:44,])
summary(regression_model)
```

```
##  
## Call:  
## lm(formula = weekly_aggregated_residuals ~ vaccination_policy,  
##      data = new_ts_covid_morocco[1:44, ])  
##  
## Residuals:  
##     Min      1Q Median      3Q     Max  
## -342.4 -306.4 -221.7  186.5 1292.4  
##  
## Coefficients:  
##                         Estimate Std. Error t value Pr(>|t|)  
## (Intercept)          28.201    154.814   0.182   0.856  
## vaccination_policy    1.213      6.973   0.174   0.863  
##  
## Residual standard error: 436.9 on 42 degrees of freedom  
## Multiple R-squared:  0.0007194, Adjusted R-squared: -0.02307  
## F-statistic: 0.03024 on 1 and 42 DF, p-value: 0.8628
```

```
anova(regression_model)
```

```
## Analysis of Variance Table  
##  
## Response: weekly_aggregated_residuals  
##                         Df Sum Sq Mean Sq F value Pr(>F)  
## vaccination_policy  1  5772   5772  0.0302 0.8628  
## Residuals          42 8017786 190900
```

```
error = residuals(regression_model)  
ts_error = ts(error)  
  
#ARIMA on error  
  
error_model = auto.arima(ts_error)  
fitted_error = fitted(error_model)  
summary(error_model)
```

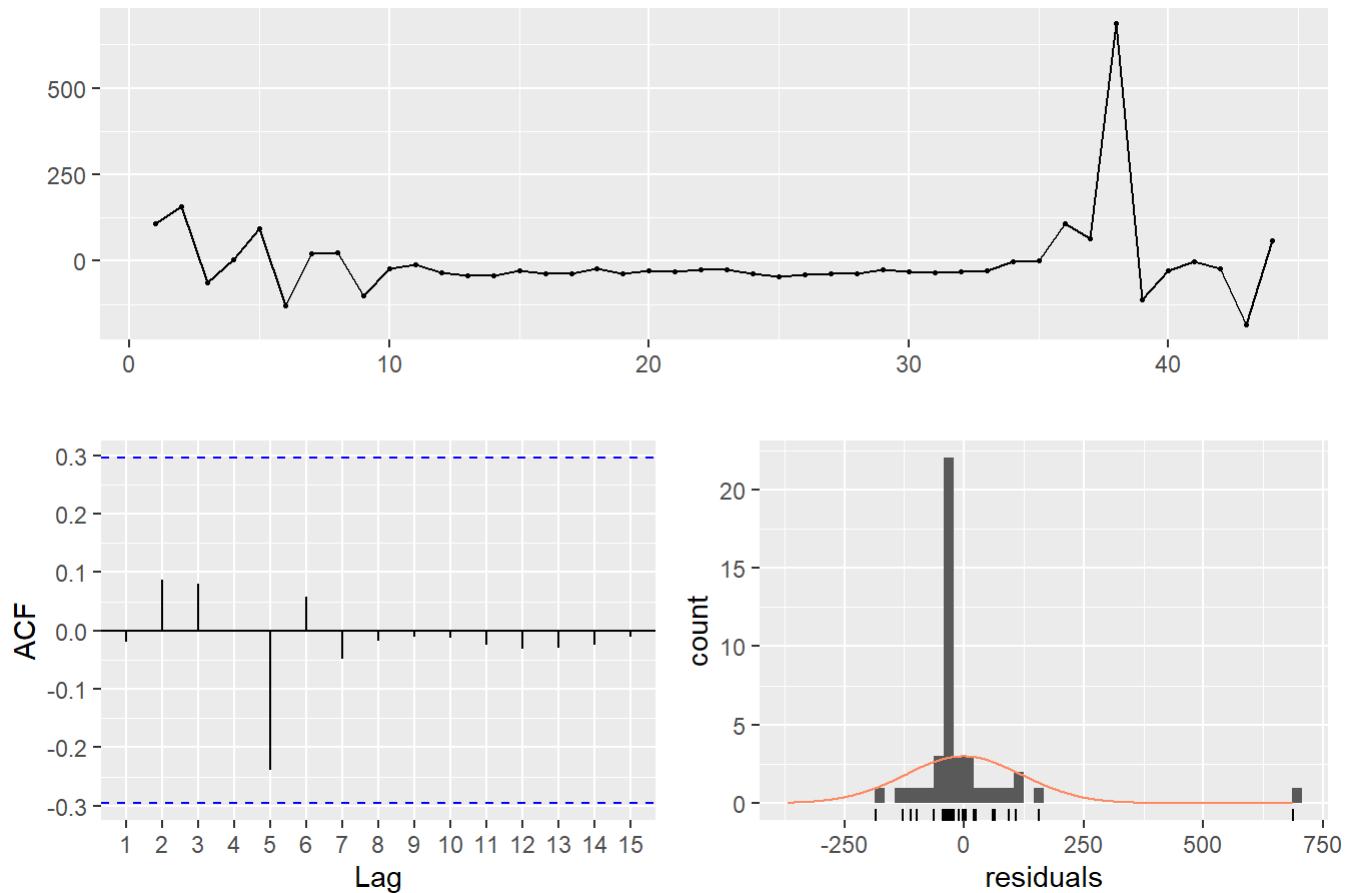
```

## Series: ts_error
## ARIMA(1,0,2) with zero mean
##
## Coefficients:
##      ar1     ma1     ma2
##      0.7347  1.0160  0.5372
## s.e.  0.1139  0.1916  0.1674
##
## sigma^2 estimated as 15679: log likelihood=-275.13
## AIC=558.26   AICc=559.28   BIC=565.39
##
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -1.961649 120.8719 62.32005 -0.8027414 41.9949 0.657148
##          ACF1
## Training set -0.01846285

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(1,0,2) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(1,0,2) with zero mean  
## Q* = 3.9724, df = 6, p-value = 0.6804  
##  
## Model df: 3. Total lags used: 9
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

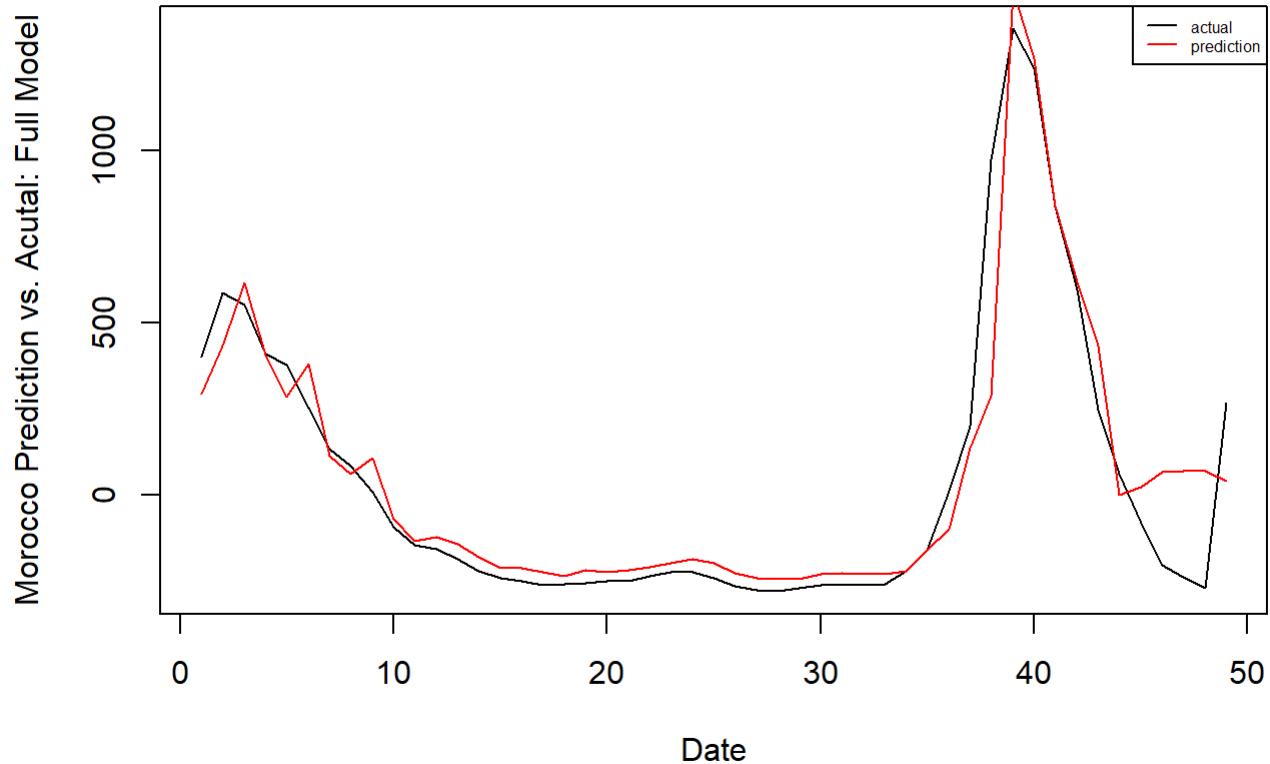
regression_prediction = predict(regression_model, newdata = new_ts_covid_morocco[45:49,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

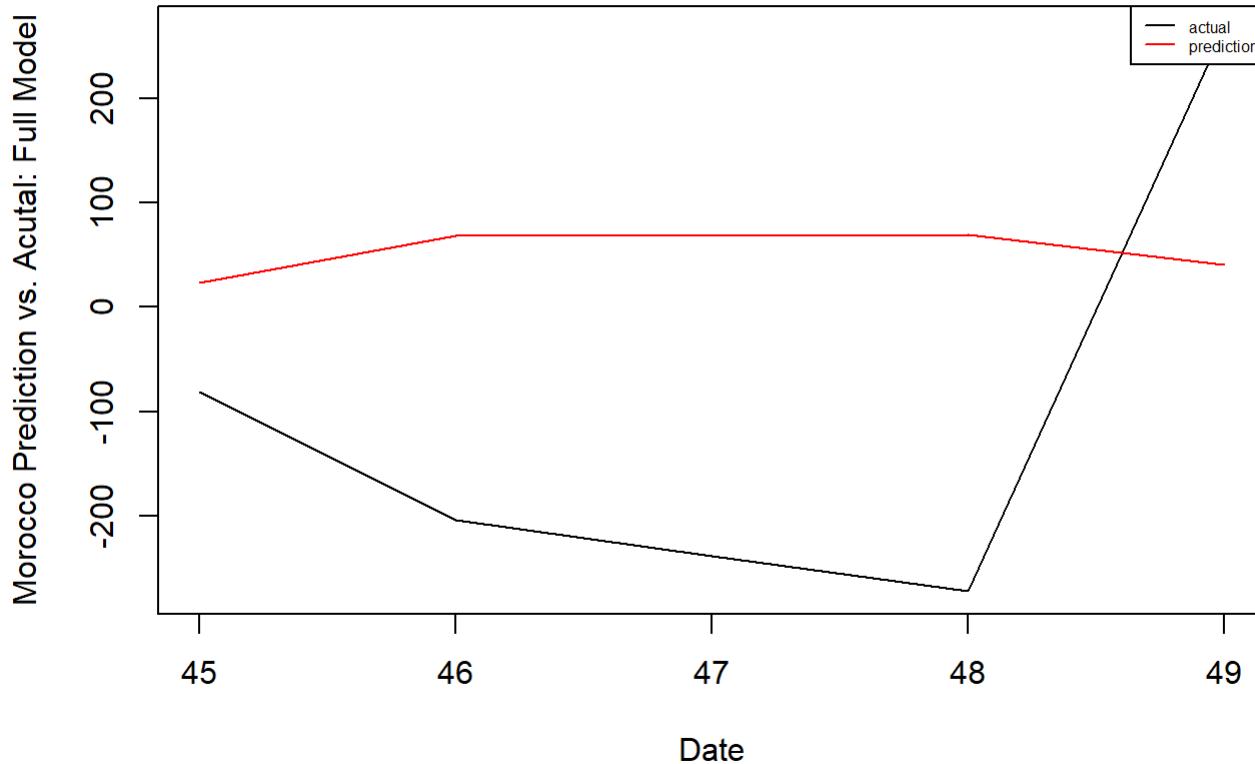
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_morocco$weekly_aggregated_residuals[1:49])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_morocco$week[1:49],                                     # Draw first time series
      new_ts_covid_morocco$weekly_aggregated_residuals[1:49],
      type = "l",
      xlab = "Date",
      ylab = "Morocco Prediction vs. Actual: Full Model")
lines(new_ts_covid_morocco$week[1:49],                                     # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_morocco$week[45:49],  
      new_ts_covid_morocco$weekly_aggregated_residuals[45:49],  
      type = "l",  
      xlab = "Date",  
      ylab = "Morocco Prediction vs. Acutal: Full Model")  
lines(new_ts_covid_morocco$week[45:49],  
      final_prediction_2,  
      type = "l",  
      col = "red")  
legend(x = "topright", legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Morocco Full
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 14610.02
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 69294.38
```

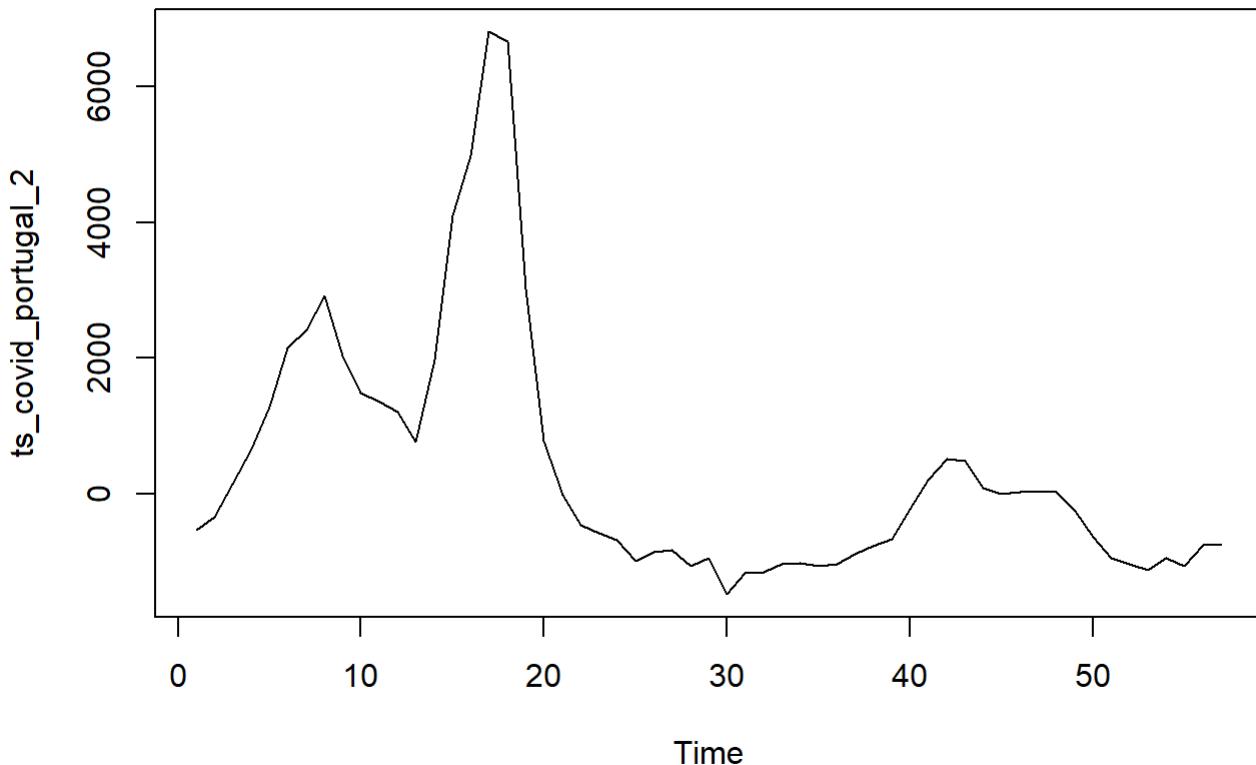
```
#Portugal Example on aggregate weekly
ts_covid_portugal = ts(covid_portugal$residuals,frequency = 7)
ts_covid_portugal_2 = ts(colSums(matrix(ts_covid_portugal, nrow=7)))
```

```
## Warning in matrix(ts_covid_portugal, nrow = 7): data length [397] is not a sub-
## multiple or multiple of the number of rows [7]
```

```
vaccine_policy = colSums(matrix(as.numeric(covid_portugal$vaccination_policy), nrow=7))
```

```
## Warning in matrix(as.numeric(covid_portugal$vaccination_policy), nrow = 7): data  
## length [397] is not a sub-multiple or multiple of the number of rows [7]
```

```
new_ts_covid_portugal = data.frame(1:57,ts_covid_portugal_2,vaccine_policy)  
names(new_ts_covid_portugal) = c("week","weekly_aggregated_residuals","vaccination_policy")  
  
plot.ts(ts_covid_portugal_2)
```



```
regression_model = lm(weekly_aggregated_residuals~1, data = new_ts_covid_portugal[1:52,])  
summary(regression_model)
```

```
##  
## Call:  
## lm(formula = weekly_aggregated_residuals ~ 1, data = new_ts_covid_portugal[1:52,  
##     ])  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max  
## -1971.0 -1353.8  -502.7   738.2  6322.0  
##  
## Coefficients:  
##             Estimate Std. Error t value Pr(>|t|)  
## (Intercept)  488.6     264.2    1.85   0.0702 .  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 1905 on 51 degrees of freedom
```

```
anova(regression_model)
```

```
## Analysis of Variance Table  
##  
## Response: weekly_aggregated_residuals  
##              Df  Sum Sq Mean Sq F value Pr(>F)  
## Residuals  51 185099032 3629393
```

```
error = residuals(regression_model)  
ts_error = ts(error)
```

```
#ARIMA on error
```

```
error_model = auto.arima(ts_error)  
fitted_error = fitted(error_model)  
summary(error_model)
```

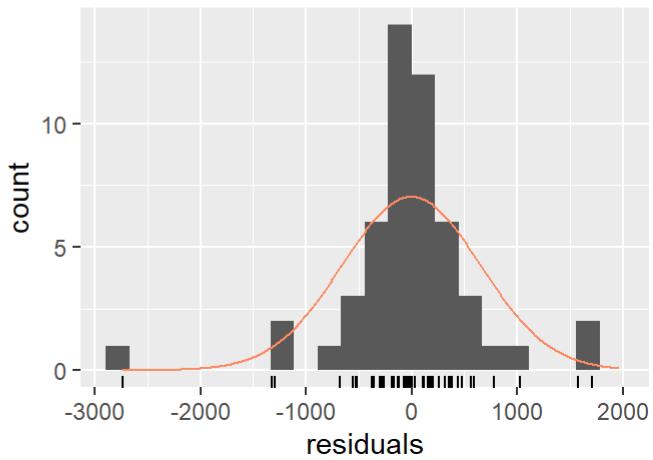
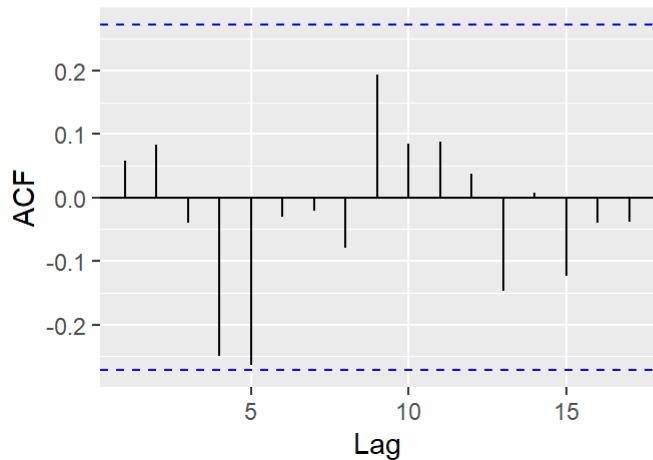
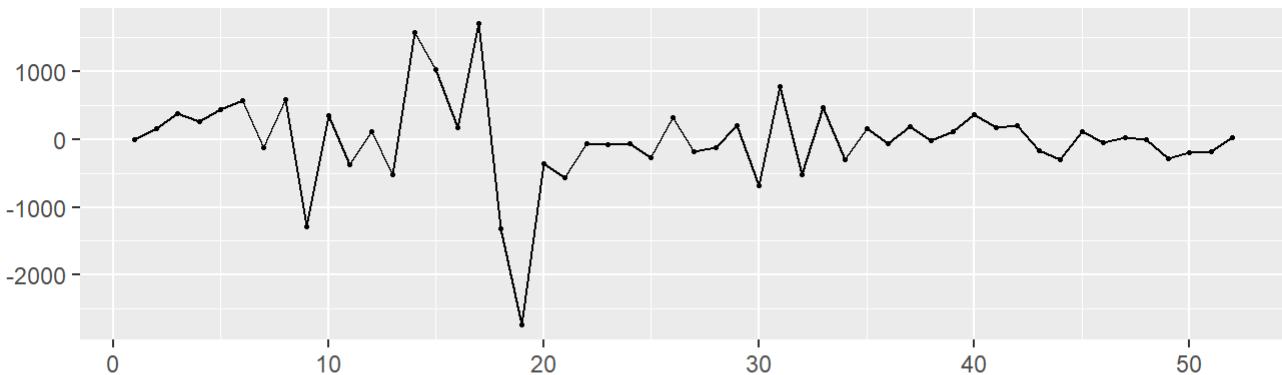
```

## Series: ts_error
## ARIMA(0,1,1)
##
## Coefficients:
##         ma1
##     0.6919
## s.e.  0.1048
##
## sigma^2 estimated as 439412:  log likelihood=-403.51
## AIC=811.03    AICc=811.28    BIC=814.89
##
## Training set error measures:
##             ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -5.843044 650.0086 409.2997 -140.7349 201.4593 0.8600816
##             ACF1
## Training set 0.05855213

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(0,1,1)



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(0,1,1)  
## Q* = 11.86, df = 9, p-value = 0.2213  
##  
## Model df: 1. Total lags used: 10
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

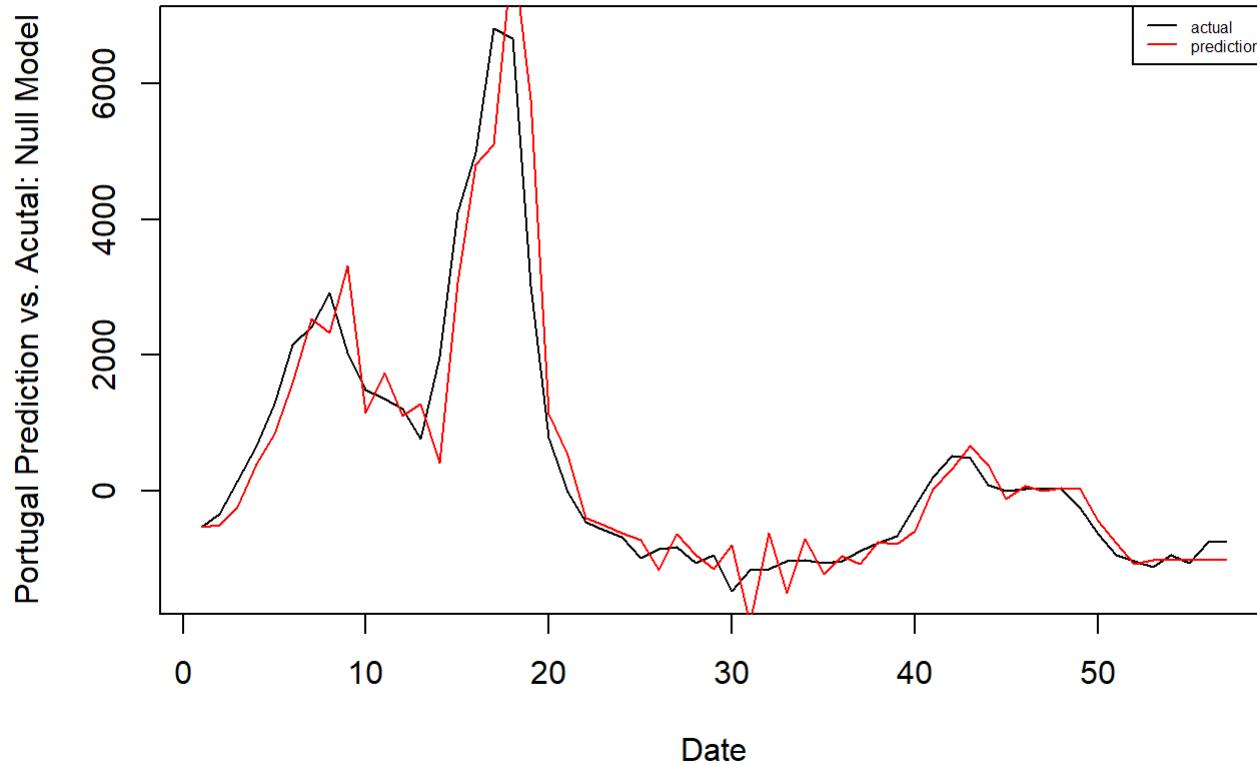
regression_prediction = predict(regression_model, newdata = new_ts_covid_portugal[53:57,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

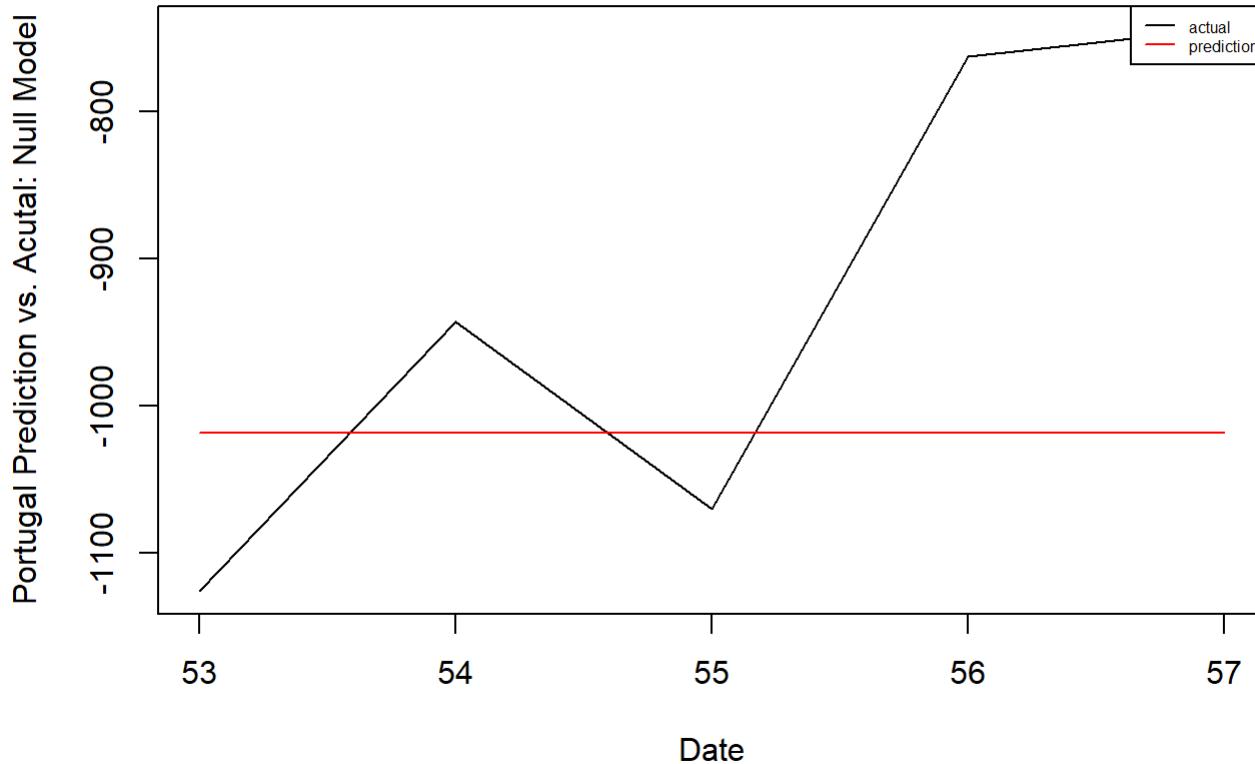
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_portugal$weekly_aggregated_residuals[1:57])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_portugal$week[1:57],                                     # Draw first time series
      new_ts_covid_portugal$weekly_aggregated_residuals[1:57],
      type = "l",
      xlab = "Date",
      ylab = "Portugal Prediction vs. Actual: Null Model")
lines(new_ts_covid_portugal$week[1:57],                                     # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_portugal$week[53:57],  
     new_ts_covid_portugal$weekly_aggregated_residuals[53:57],  
     type = "l",  
     xlab = "Date",  
     ylab = "Portugal Prediction vs. Acutal: Null Model")  
lines(new_ts_covid_portugal$week[53:57],  
      final_prediction_2,  
      type = "l",  
      col = "red")  
legend(x = "topright", legend=c("actual", "prediction"),  
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Portugal Null
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 422511.2
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 32079.86
```

```
regression_model = lm(weekly_aggregated_residuals~vaccination_policy, data = new_ts_covid_portugal[1:52,])
summary(regression_model)
```

```
##
## Call:
## lm(formula = weekly_aggregated_residuals ~ vaccination_policy,
##     data = new_ts_covid_portugal[1:52, ])
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -2381.6  -949.3  -431.5   418.4  5969.0 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 2347.42    557.37   4.212 0.000106 ***
## vaccination_policy -71.70     19.47  -3.683 0.000565 *** 
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
##
## Residual standard error: 1706 on 50 degrees of freedom
## Multiple R-squared:  0.2134, Adjusted R-squared:  0.1977 
## F-statistic: 13.57 on 1 and 50 DF,  p-value: 0.0005654
```

```
anova(regression_model)
```

```
## Analysis of Variance Table
##
## Response: weekly_aggregated_residuals
##              Df  Sum Sq Mean Sq F value Pr(>F)    
## vaccination_policy 1 39505355 39505355 13.567 0.0005654 ***
## Residuals         50 145593677 2911874
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
error = residuals(regression_model)
ts_error = ts(error)

#ARIMA on error

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)
```

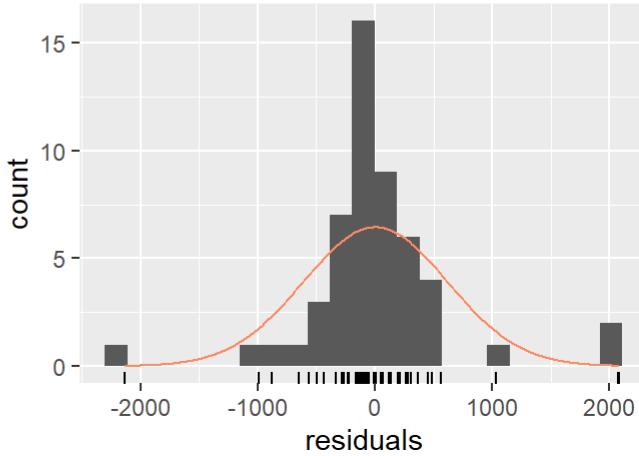
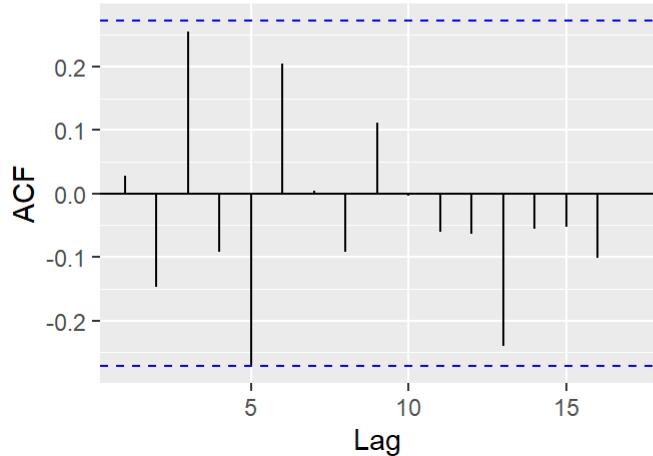
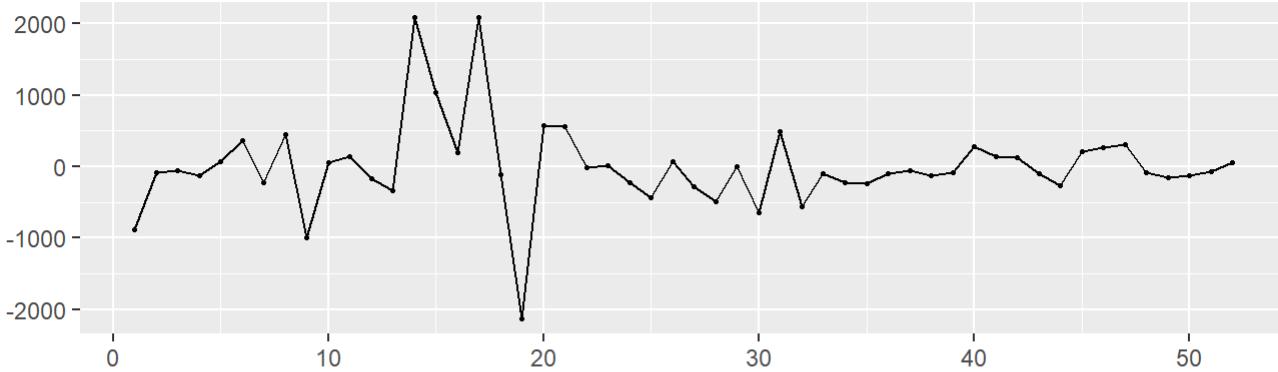
```

## Series: ts_error
## ARIMA(2,0,0) with zero mean
##
## Coefficients:
##      ar1     ar2
##      1.4320 -0.6283
##  s.e.  0.1037  0.1045
##
## sigma^2 estimated as 389234:  log likelihood=-408.68
## AIC=823.36   AICc=823.86   BIC=829.21
##
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 1.573901 611.7709 365.025 -12.30432 63.54187 0.7431862 0.02792383

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(2,0,0) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,0,0) with zero mean  
## Q* = 13.761, df = 8, p-value = 0.0882  
##  
## Model df: 2. Total lags used: 10
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

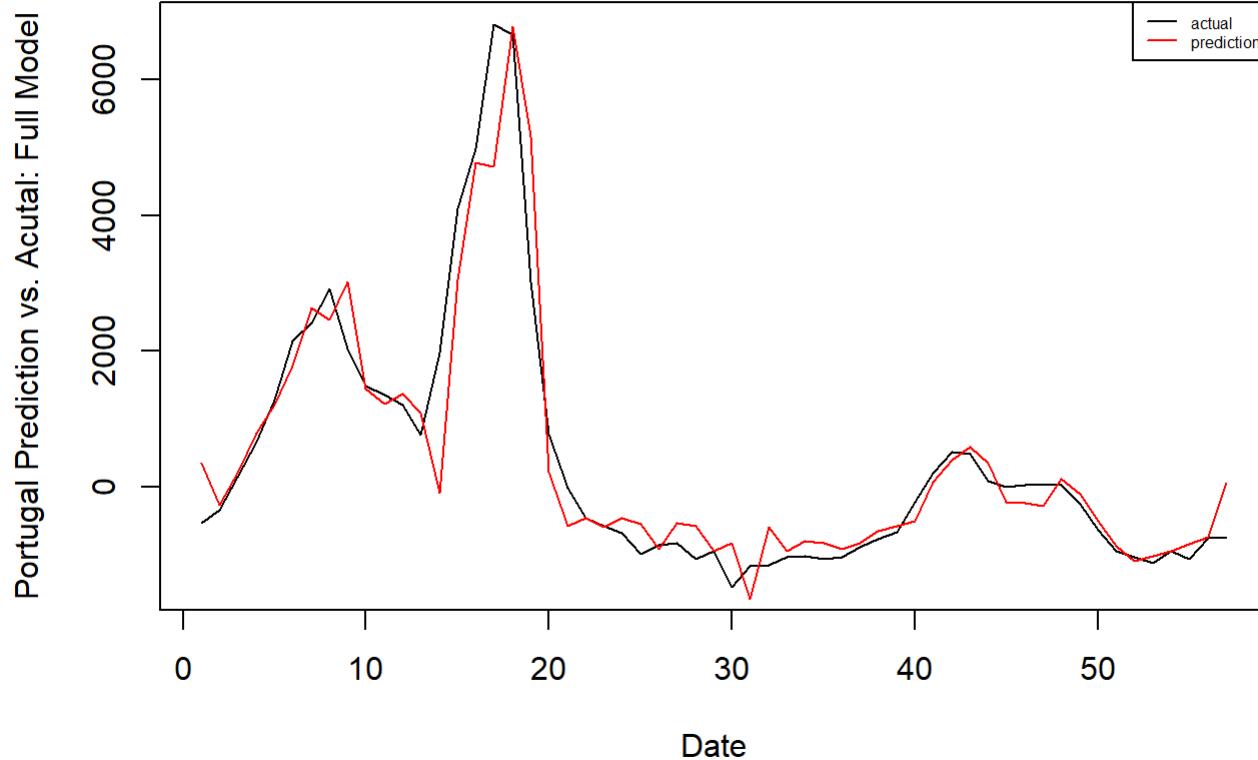
regression_prediction = predict(regression_model, newdata = new_ts_covid_portugal[53:57,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

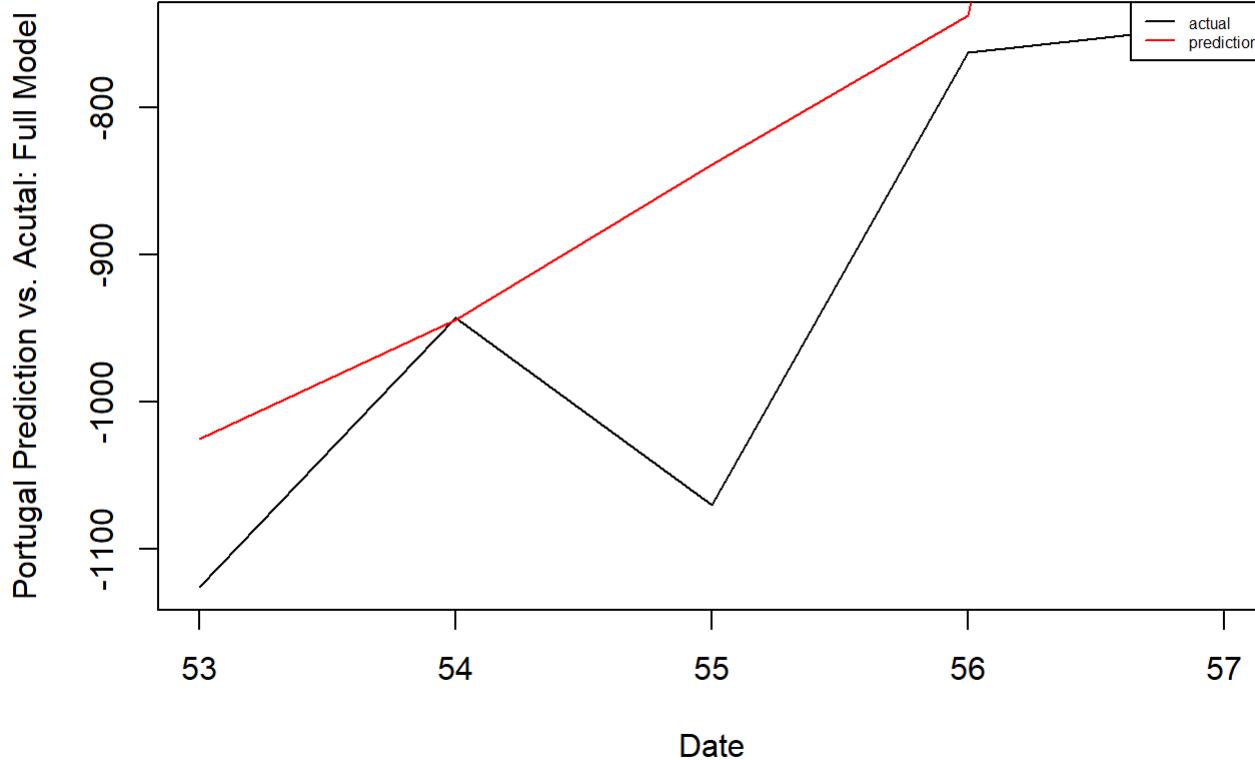
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_portugal$weekly_aggregated_residuals[1:57])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_portugal$week[1:57],  
      new_ts_covid_portugal$weekly_aggregated_residuals[1:57],  
      type = "l",  
      xlab = "Date",  
      ylab = "Portugal Prediction vs. Actual: Full Model")  
lines(new_ts_covid_portugal$week[1:57],  
      compare_table$predict,  
      type = "l",  
      col = 'red')  
legend(x = "topright",legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_portugal$week[53:57],  
      new_ts_covid_portugal$weekly_aggregated_residuals[53:57],  
      type = "l",  
      xlab = "Date",  
      ylab = "Portugal Prediction vs. Acutal: Full Model")  
# Draw first time series  
lines(new_ts_covid_portugal$week[53:57],  
      final_prediction_2,  
      type = "l",  
      col = "red")  
# Draw second time series  
legend(x = "topright", legend=c("actual", "prediction"),  
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Portugal Full
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 374263.6
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 141228.8
```

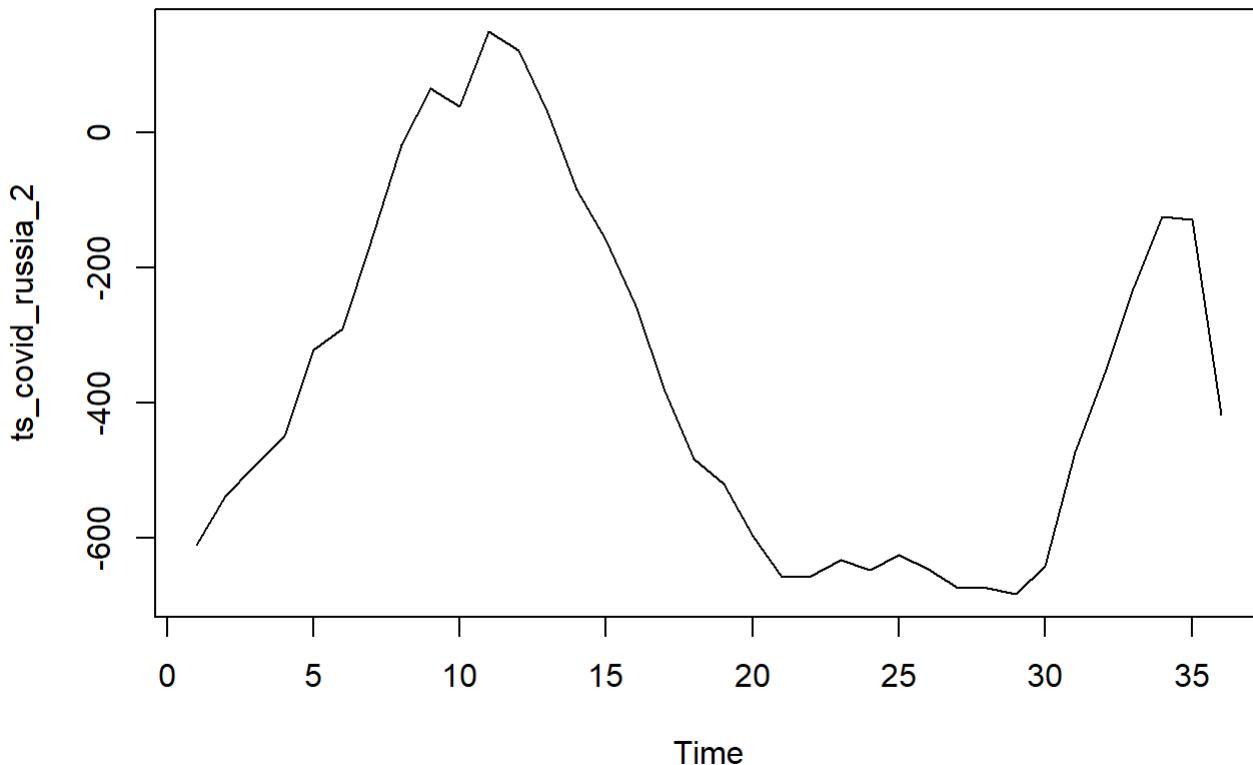
```
#Russia Example on aggregate weekly
ts_covid_russia = ts(covid_russia$residuals,frequency = 7)
ts_covid_russia_2 = ts(colSums(matrix(ts_covid_russia, nrow=7)))
```

```
## Warning in matrix(ts_covid_russia, nrow = 7): data length [248] is not a sub-
## multiple or multiple of the number of rows [7]
```

```
vaccine_policy = colSums(matrix(as.numeric(covid_russia$vaccination_policy), nrow=7))
```

```
## Warning in matrix(as.numeric(covid_russia$vaccination_policy), nrow = 7): data  
## length [248] is not a sub-multiple or multiple of the number of rows [7]
```

```
new_ts_covid_russia = data.frame(1:36,ts_covid_russia_2,vaccine_policy)  
names(new_ts_covid_russia) = c("week","weekly_aggregated_residuals","vaccination_policy")  
  
plot.ts(ts_covid_russia_2)
```



```
regression_model = lm(weekly_aggregated_residuals~1, data = new_ts_covid_russia[1:31,])  
summary(regression_model)
```

```

## 
## Call:
## lm(formula = weekly_aggregated_residuals ~ 1, data = new_ts_covid_russia[1:31,
##   ])
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -297.37 -250.87 -97.64 227.85 535.02 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -385.77     50.27  -7.674 1.47e-08 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 279.9 on 30 degrees of freedom

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: weekly_aggregated_residuals
##           Df  Sum Sq Mean Sq F value Pr(>F)    
## Residuals 30 2350014   78334

```

```

error = residuals(regression_model)
ts_error = ts(error)

```

```

#ARIMA on error

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)

```

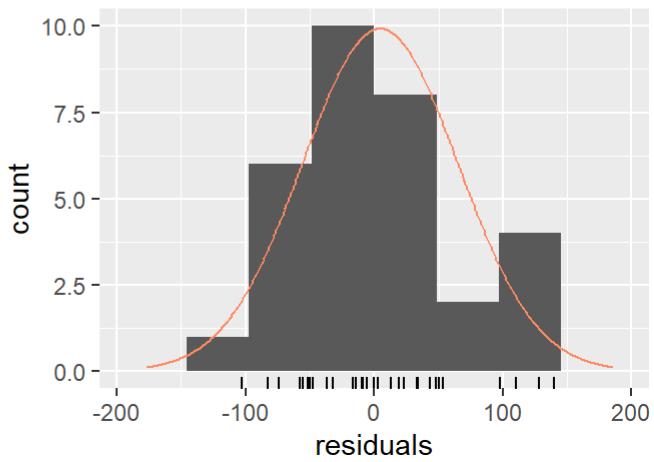
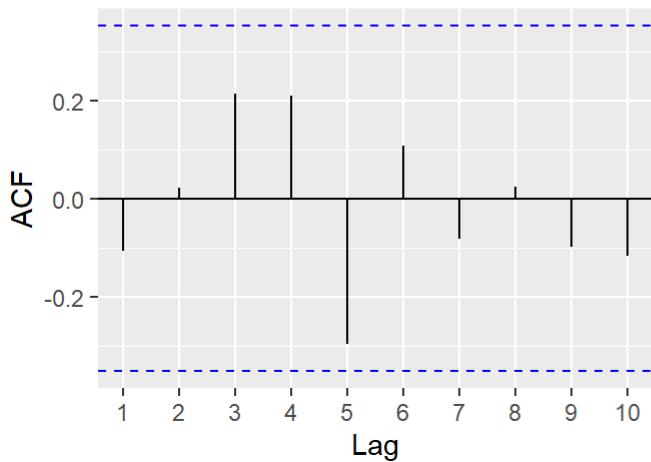
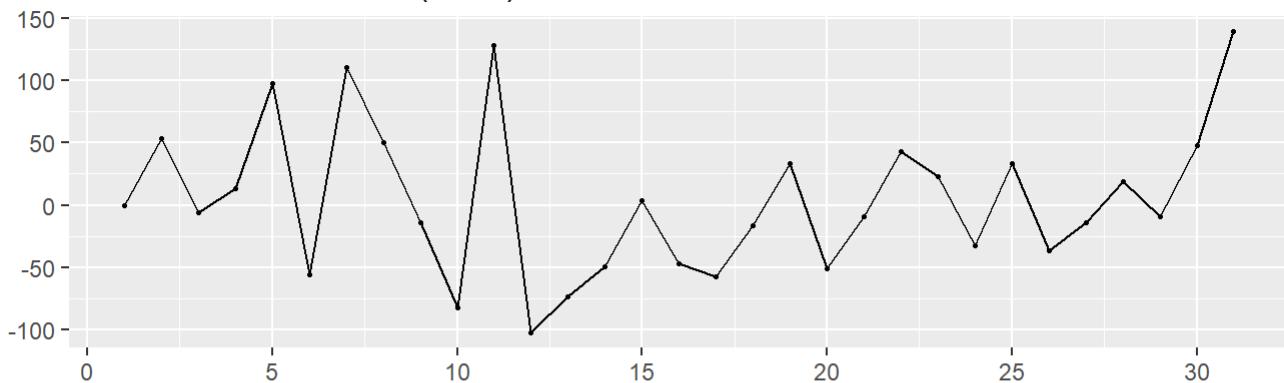
```

## Series: ts_error
## ARIMA(1,1,0)
## 
## Coefficients:
##         ar1
##       0.6854
## s.e.  0.1447
## 
## sigma^2 estimated as 3804:  log likelihood=-166.03
## AIC=336.07  AICc=336.51  BIC=338.87
## 
## Training set error measures:
##               ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 4.471692 59.65287 46.93884 -46.88967 67.77412 0.721452 -0.1063178

```

```
checkresiduals(error_model1)
```

Residuals from ARIMA(1,1,0)



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(1,1,0)  
## Q* = 7.7048, df = 5, p-value = 0.1733  
##  
## Model df: 1. Total lags used: 6
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

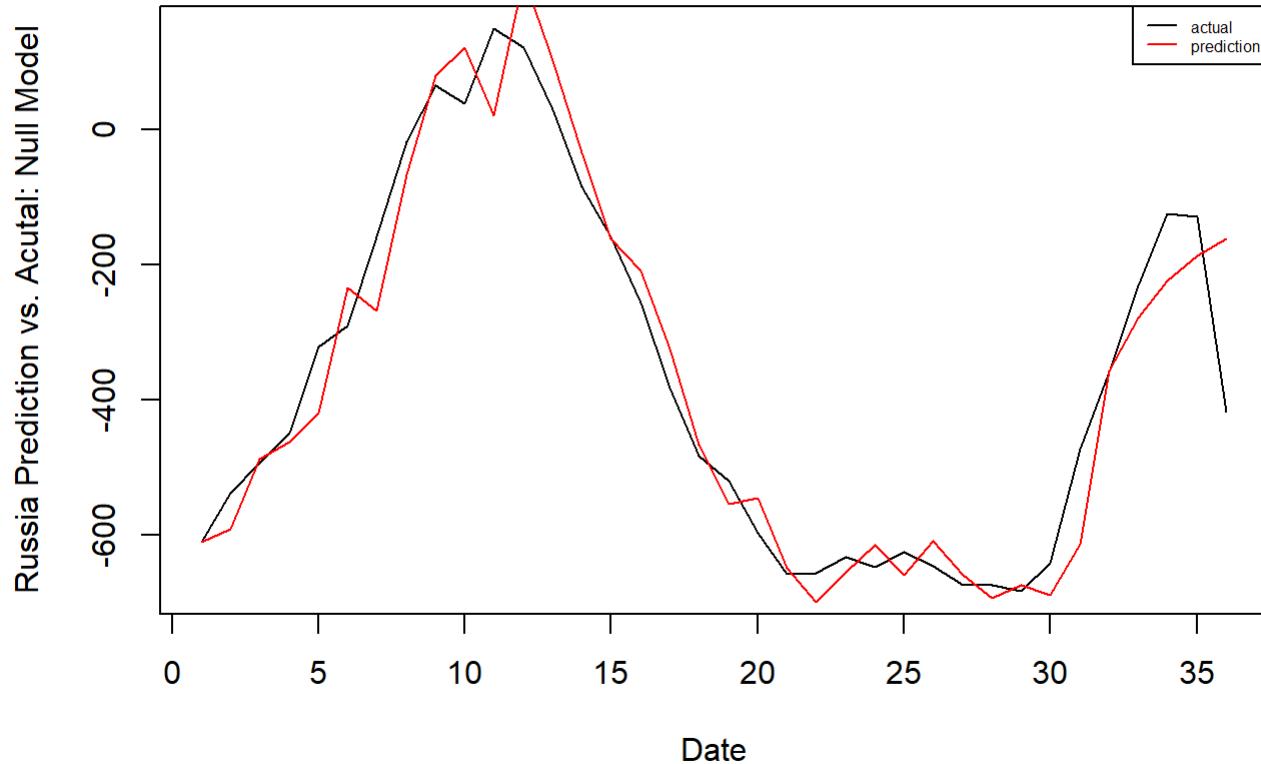
regression_prediction = predict(regression_model, newdata = new_ts_covid_russia[32:36,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

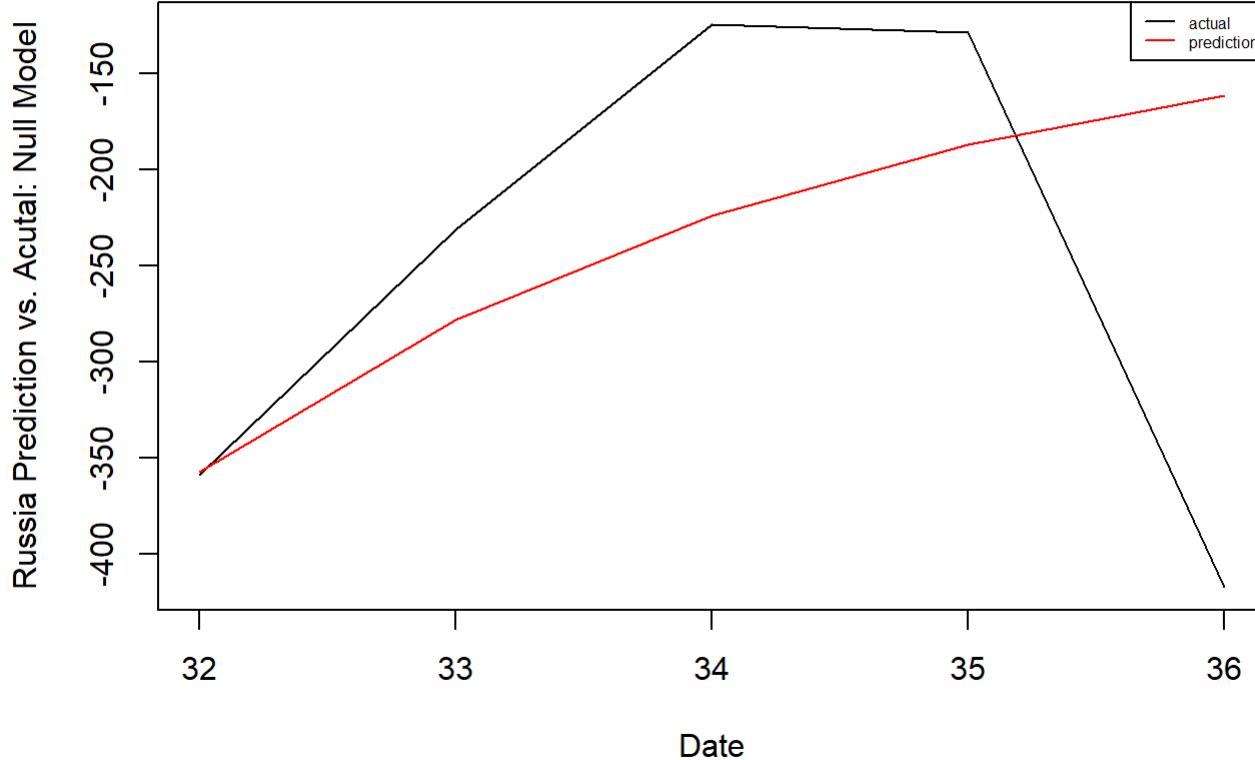
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_portugal$weekly_aggregated_residuals[1:36])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_russia$week[1:36],                                     # Draw first time series
      new_ts_covid_russia$weekly_aggregated_residuals[1:36],
      type = "l",
      xlab = "Date",
      ylab = "Russia Prediction vs. Actual: Null Model")
lines(new_ts_covid_russia$week[1:36],                                      # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_russia$week[32:36],                                     # Draw first time series
     new_ts_covid_russia$weekly_aggregated_residuals[32:36],
     type = "l",
     xlab = "Date",
     ylab = "Russia Prediction vs. Acutal: Null Model")
lines(new_ts_covid_russia$week[32:36],                                     # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Russia Null
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 6538372
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 676692.2
```

```
regression_model = lm(weekly_aggregated_residuals~vaccination_policy, data = new_ts_covid_russia[1:31,])
summary(regression_model)
```

```

## 
## Call:
## lm(formula = weekly_aggregated_residuals ~ vaccination_policy,
##     data = new_ts_covid_russia[1:31, ])
## 
## Residuals:
##    Min      1Q   Median      3Q     Max 
## -441.13 -86.04 -27.46 167.43 318.89 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -79.972    67.288 -1.188   0.244    
## vaccination_policy -12.811     2.378 -5.388 8.63e-06 *** 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 201.2 on 29 degrees of freedom 
## Multiple R-squared:  0.5003, Adjusted R-squared:  0.483 
## F-statistic: 29.03 on 1 and 29 DF,  p-value: 8.634e-06

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: weekly_aggregated_residuals
##              Df  Sum Sq Mean Sq F value    Pr(>F)    
## vaccination_policy 1 1175600 1175600 29.029 8.634e-06 *** 
## Residuals         29 1174413   40497 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

```

```

error = residuals(regression_model)
ts_error = ts(error)

#ARIMA on error

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)

```

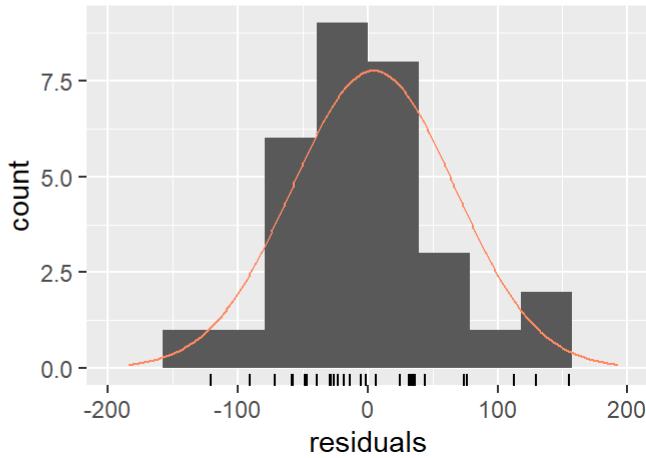
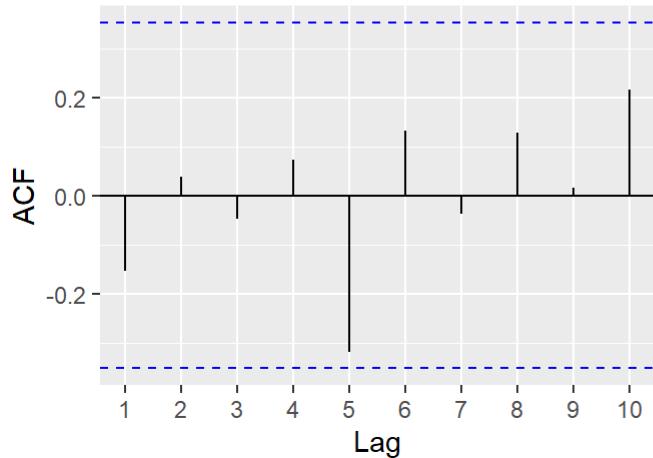
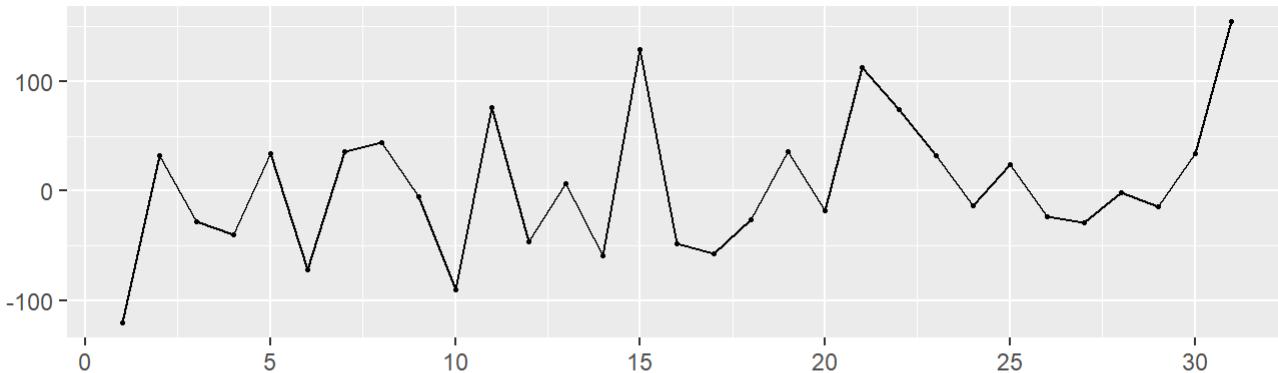
```

## Series: ts_error
## ARIMA(2,0,1) with zero mean
##
## Coefficients:
##      ar1     ar2     ma1
##      1.8772 -0.9497 -0.8122
##  s.e.  0.0532  0.0474  0.1588
##
## sigma^2 estimated as 4247:  log likelihood=-173.78
## AIC=355.56  AICc=357.1  BIC=361.3
##
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 4.415691 61.93291 49.08293 -14.78802 72.61039 0.7757536 -0.1539014

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(2,0,1) with zero mean



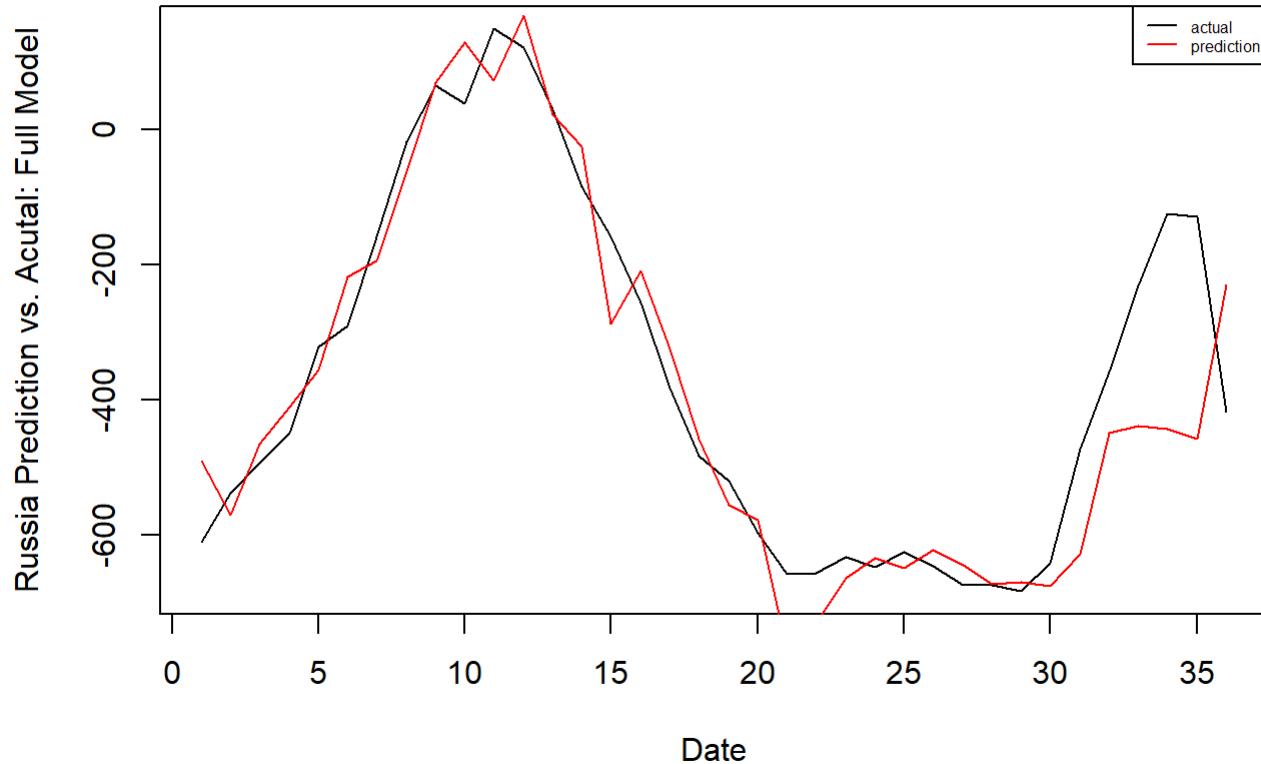
```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,0,1) with zero mean  
## Q* = 5.872, df = 3, p-value = 0.118  
##  
## Model df: 3. Total lags used: 6
```

#goodness of fit

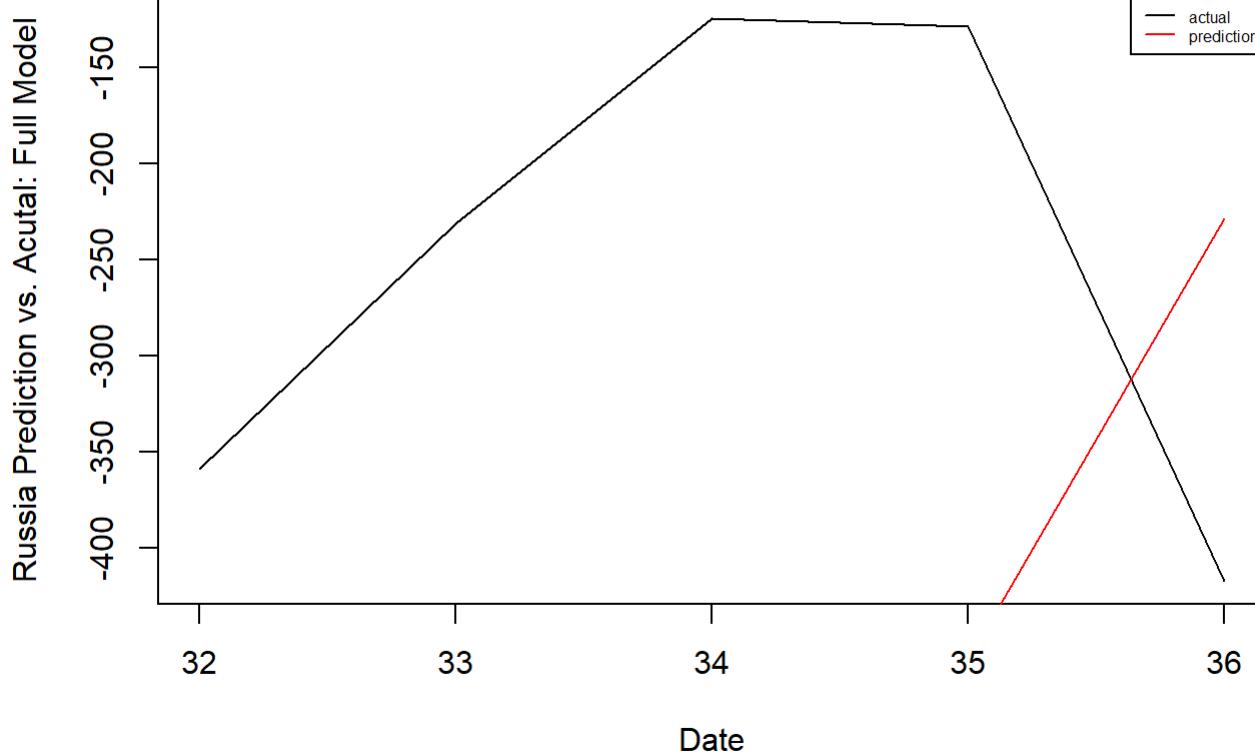
```
regression_prediction = predict(regression_model)  
  
final_prediction = regression_prediction + fitted_error  
  
regression_prediction = predict(regression_model, newdata = new_ts_covid_russia[32:36,])  
error_forecast = predict(error_model, n.ahead = 5)$pred  
  
final_prediction_2 = regression_prediction + error_forecast
```

#compare prediction and actual

```
compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_portugal$weekly_aggregated_residuals[1:36])  
colnames(compare_table) = c("predict","actual")  
  
plot(new_ts_covid_russia$week[1:36],  
      new_ts_covid_russia$weekly_aggregated_residuals[1:36],  
      type = "l",  
      xlab = "Date",  
      ylab = "Russia Prediction vs. Actual: Full Model")  
lines(new_ts_covid_russia$week[1:36],  
      compare_table$predict,  
      type = "l",  
      col = 'red')  
legend(x = "topright",legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_russia$week[32:36],                                     # Draw first time series
     new_ts_covid_russia$weekly_aggregated_residuals[32:36],
     type = "l",
     xlab = "Date",
     ylab = "Russia Prediction vs. Acutal: Full Model")
lines(new_ts_covid_russia$week[32:36],                                     # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Russia Full
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 6553271
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 441397.7
```

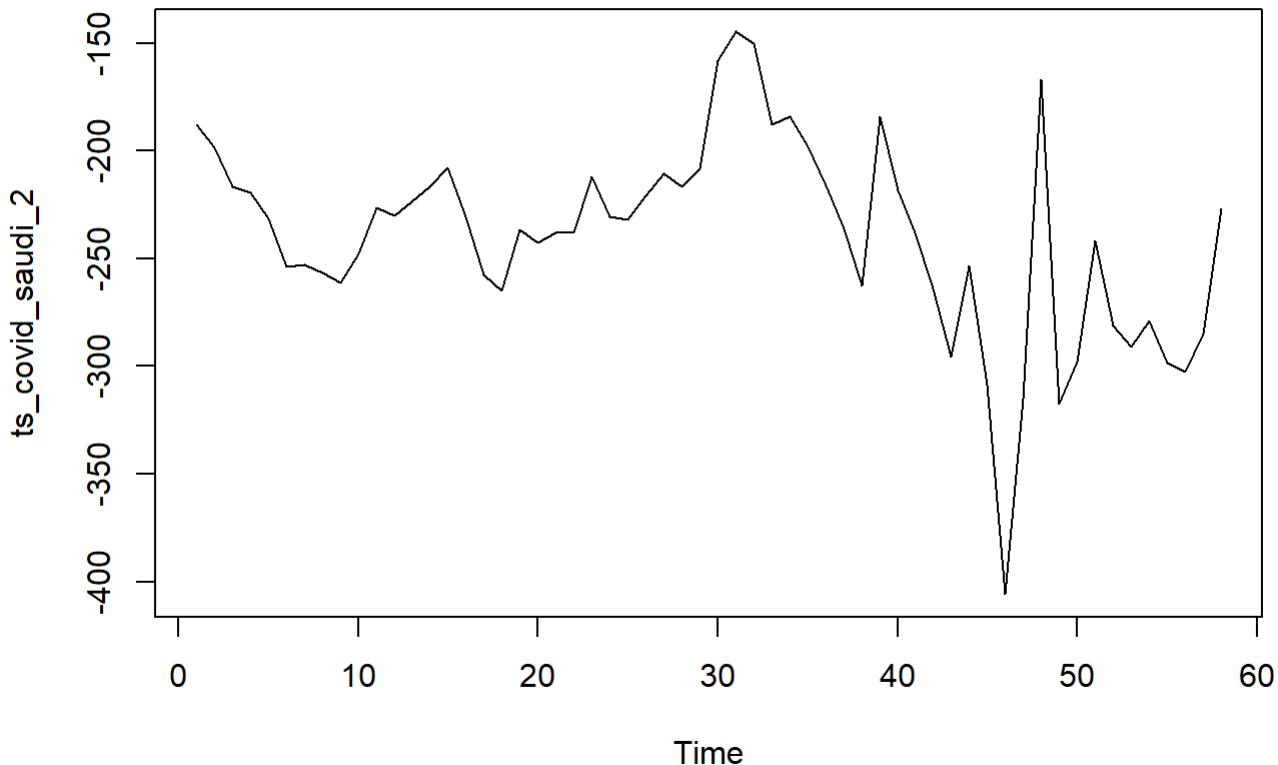
```
#Saudi Example on aggregate weekly
ts_covid_saudi = ts(covid_saudi$residuals,frequency = 7)
ts_covid_saudi_2 = ts(colSums(matrix(ts_covid_saudi, nrow=7)))
```

```
## Warning in matrix(ts_covid_saudi, nrow = 7): data length [403] is not a sub-
## multiple or multiple of the number of rows [7]
```

```
vaccine_policy = colSums(matrix(as.numeric(covid_saudi$vaccination_policy), nrow=7))
```

```
## Warning in matrix(as.numeric(covid_saudi$vaccination_policy), nrow = 7): data  
## length [403] is not a sub-multiple or multiple of the number of rows [7]
```

```
new_ts_covid_saudi = data.frame(1:58,ts_covid_saudi_2,vaccine_policy)  
names(new_ts_covid_saudi) = c("week","weekly_aggregated_residuals","vaccination_policy")  
  
plot.ts(ts_covid_saudi_2)
```



```
regression_model = lm(weekly_aggregated_residuals~1, data = new_ts_covid_saudi[1:53,])  
summary(regression_model)
```

```
##  
## Call:  
## lm(formula = weekly_aggregated_residuals ~ 1, data = new_ts_covid_saudi[1:53,  
##     ])  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max  
## -170.324  -20.928    4.183  23.546  90.863  
##  
## Coefficients:  
##             Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -235.562     6.303  -37.37 <2e-16 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 45.89 on 52 degrees of freedom
```

```
anova(regression_model)
```

```
## Analysis of Variance Table  
##  
## Response: weekly_aggregated_residuals  
##             Df Sum Sq Mean Sq F value Pr(>F)  
## Residuals  52 109506  2105.9
```

```
error = residuals(regression_model)  
ts_error = ts(error)
```

```
#ARIMA on error
```

```
error_model = auto.arima(ts_error)  
fitted_error = fitted(error_model)  
summary(error_model)
```

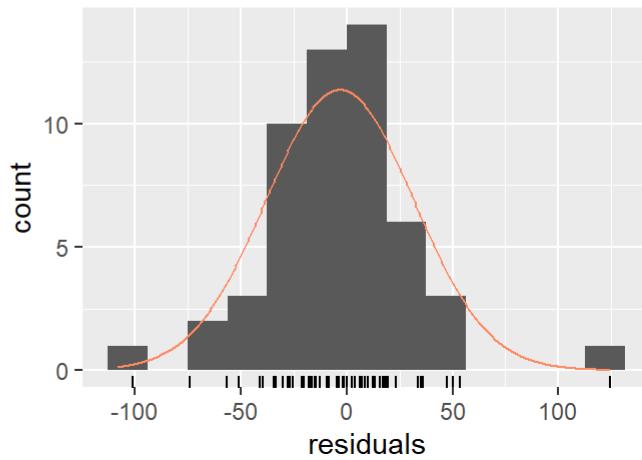
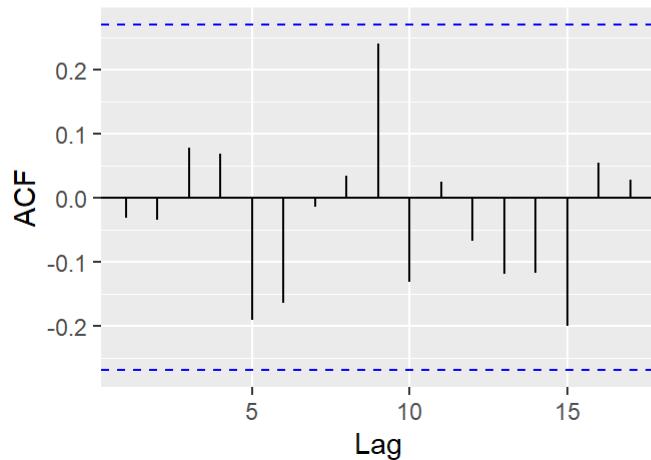
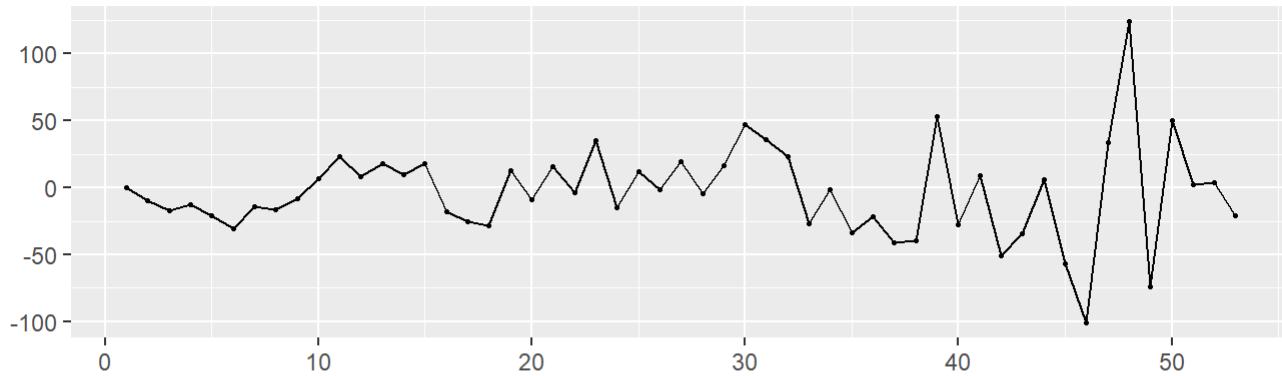
```

## Series: ts_error
## ARIMA(0,1,3)
##
## Coefficients:
##          ma1      ma2      ma3
##        -0.2765  -0.4784  0.2599
## s.e.    0.1344   0.1517  0.1322
##
## sigma^2 estimated as 1308: log likelihood=-259.19
## AIC=526.37   AICc=527.22   BIC=534.18
##
## Training set error measures:
##             ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -3.345921 34.77815 25.50902 147.1636 318.1266 0.9440369
##             ACF1
## Training set -0.03109063

```

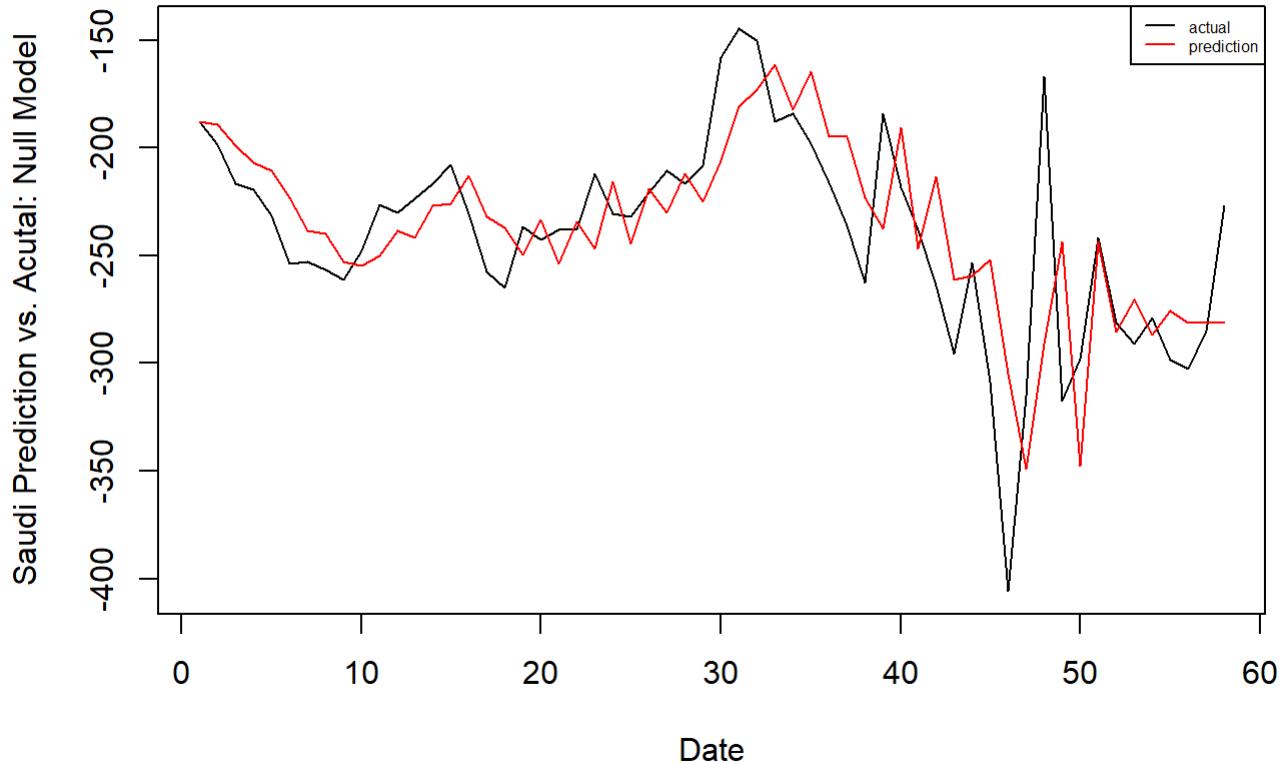
```
checkresiduals(error_model)
```

Residuals from ARIMA(0,1,3)

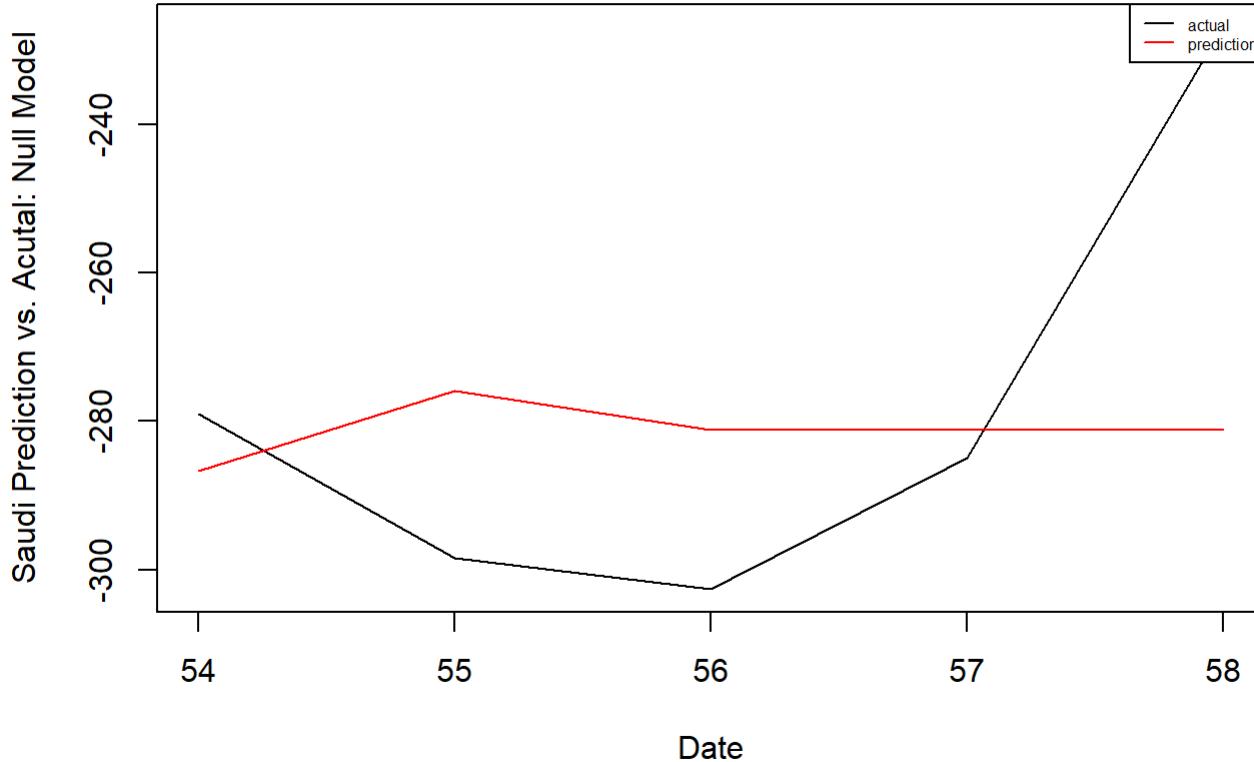


```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(0,1,3)  
## Q* = 9.7121, df = 7, p-value = 0.2055  
##  
## Model df: 3. Total lags used: 10
```

```
#goodness of fit  
regression_prediction = predict(regression_model)  
  
final_prediction = regression_prediction + fitted_error  
  
regression_prediction = predict(regression_model, newdata = new_ts_covid_saudi[54:58,])  
error_forecast = predict(error_model, n.ahead = 5)$pred  
  
final_prediction_2 = regression_prediction + error_forecast  
  
#compare prediction and actual  
  
compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_saudi$weekly_aggregated_residuals[1:58])  
colnames(compare_table) = c("predict","actual")  
  
plot(new_ts_covid_saudi$week[1:58],  
      new_ts_covid_saudi$weekly_aggregated_residuals[1:58],  
      type = "l",  
      xlab = "Date",  
      ylab = "Saudi Prediction vs. Actual: Null Model")  
lines(new_ts_covid_saudi$week[1:58],  
      compare_table$predict,  
      type = "l",  
      col = 'red')  
legend(x = "topright",legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_saudi$week[54:58],                                     # Draw first time series
     new_ts_covid_saudi$weekly_aggregated_residuals[54:58],
     type = "l",
     xlab = "Date",
     ylab = "Saudi Prediction vs. Acutal: Null Model")
lines(new_ts_covid_saudi$week[54:58],                                       # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Saudi Null
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 1209.52
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 801.4738
```

```
regression_model = lm(weekly_aggregated_residuals~vaccination_policy, data = new_ts_covid_saudi[1:53,])
summary(regression_model)
```

```

## 
## Call:
## lm(formula = weekly_aggregated_residuals ~ vaccination_policy,
##     data = new_ts_covid_saudi[1:53, ])
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -168.135  -23.097    1.735   24.298   93.051 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -229.1285   15.2670 -15.008 <2e-16 ***
## vaccination_policy -0.2053    0.4430  -0.463   0.645  
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 46.24 on 51 degrees of freedom
## Multiple R-squared:  0.004193, Adjusted R-squared:  -0.01533 
## F-statistic: 0.2147 on 1 and 51 DF,  p-value: 0.645

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: weekly_aggregated_residuals
##              Df Sum Sq Mean Sq F value Pr(>F)    
## vaccination_policy  1  459   459.15  0.2147  0.645  
## Residuals          51 109047 2138.17

```

```

error = residuals(regression_model)
ts_error = ts(error)

#ARIMA on error

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)

```

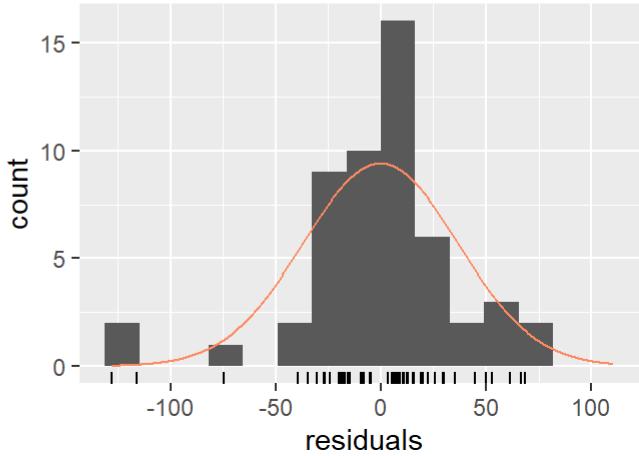
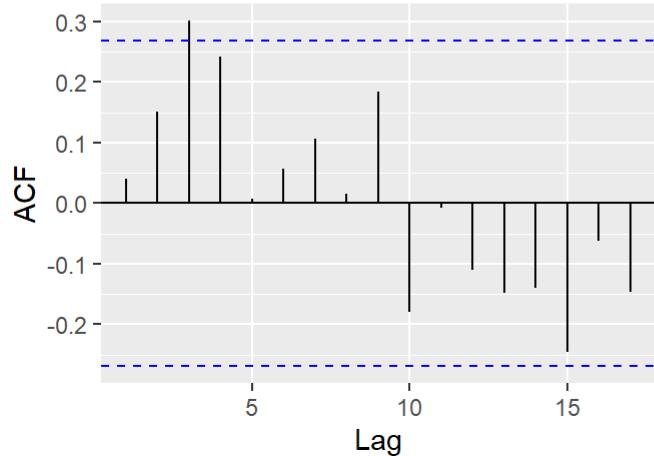
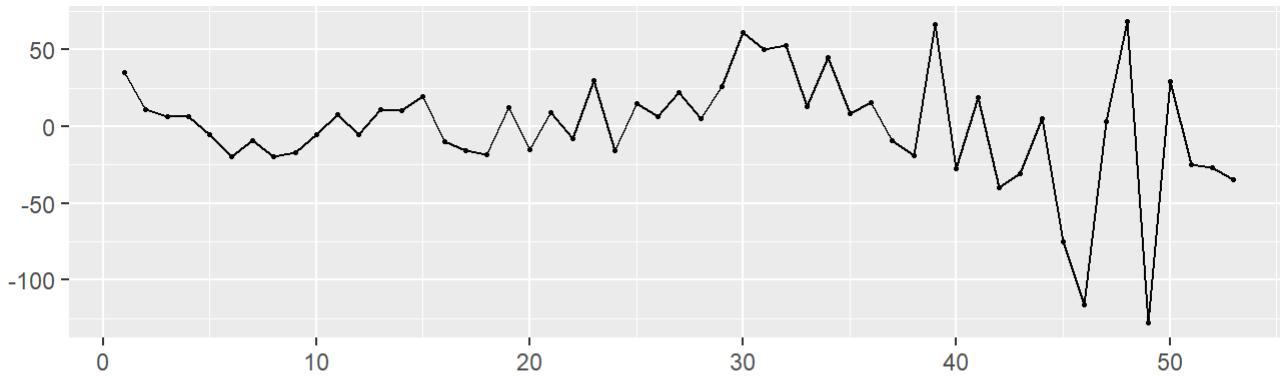
```

## Series: ts_error
## ARIMA(0,0,1) with zero mean
##
## Coefficients:
##         ma1
##     0.6997
## s.e.  0.1109
##
## sigma^2 estimated as 1359:  log likelihood=-266.22
## AIC=536.44    AICc=536.68    BIC=540.38
##
## Training set error measures:
##             ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -0.3100878 36.51426 25.75868 -448.3118 642.7089 0.9500839
##                 ACF1
## Training set 0.04060782

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(0,0,1) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(0,0,1) with zero mean  
## Q* = 15.54, df = 9, p-value = 0.07714  
##  
## Model df: 1. Total lags used: 10
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

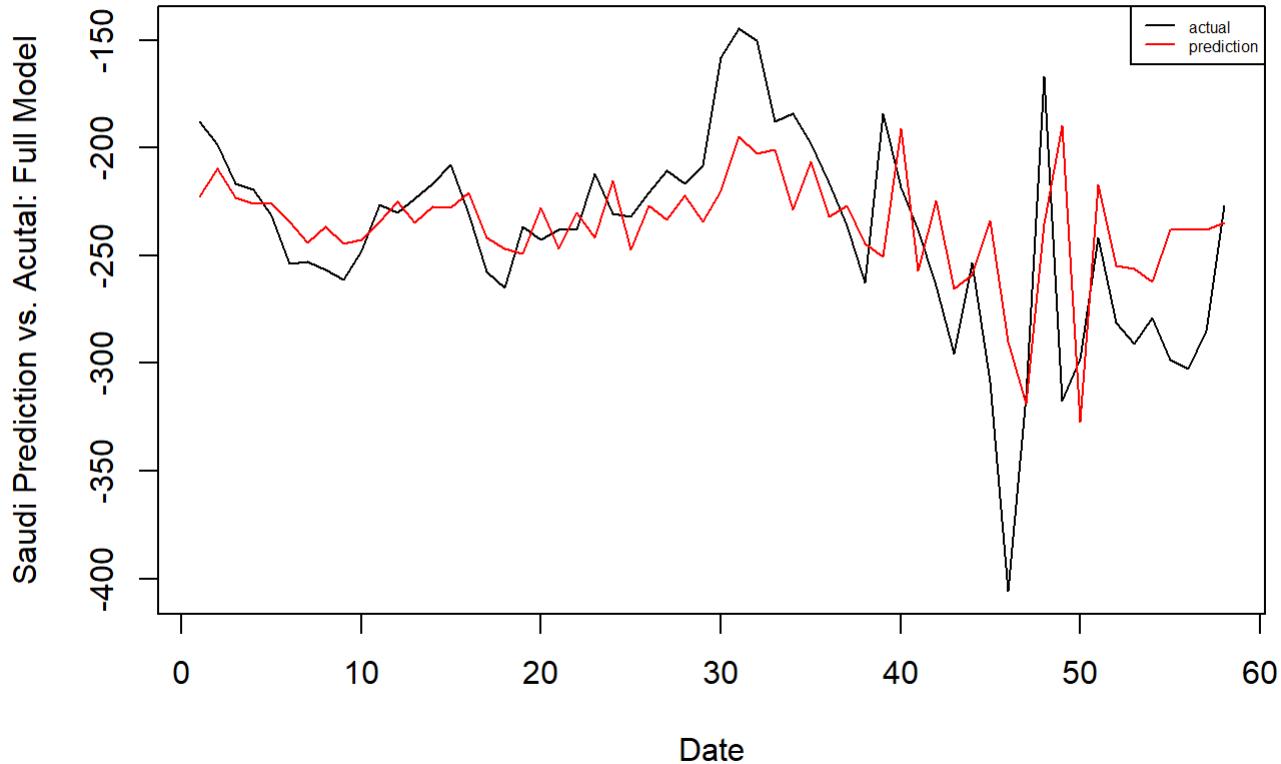
regression_prediction = predict(regression_model, newdata = new_ts_covid_saudi[54:58,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

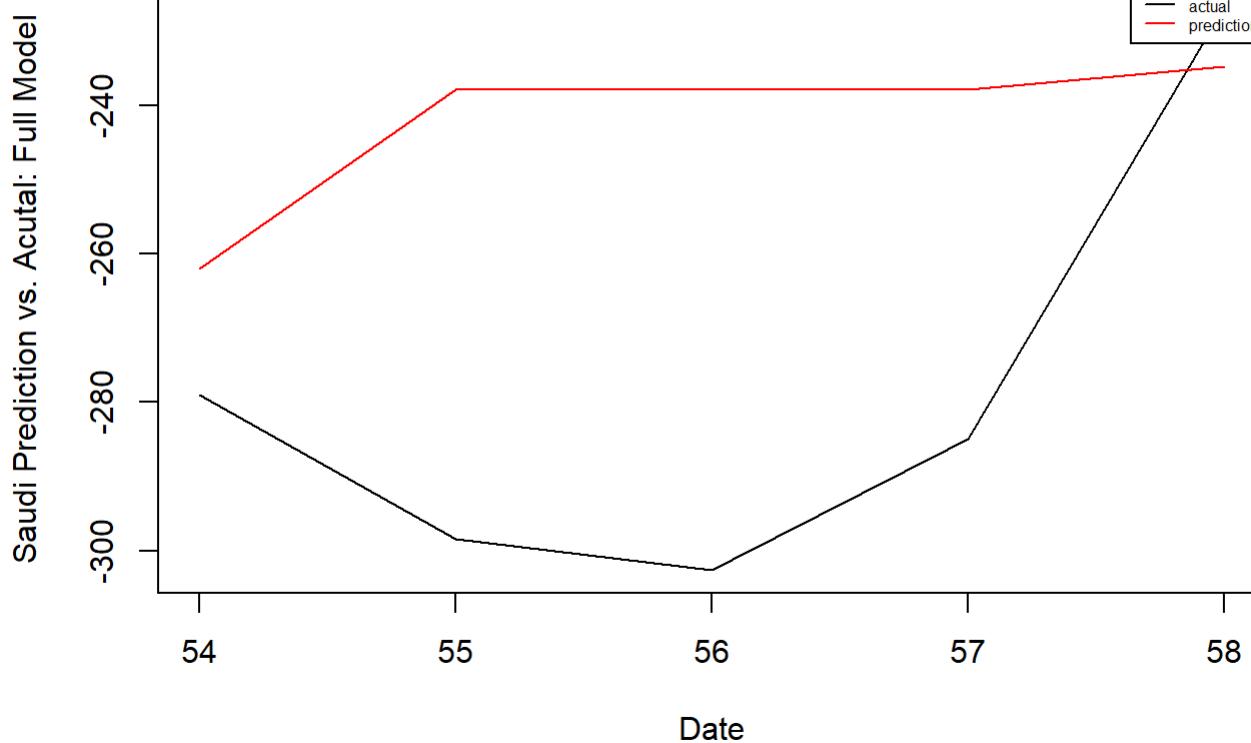
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_saudi$weekly_aggregated_residuals[1:58])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_saudi$week[1:58],  
      new_ts_covid_saudi$weekly_aggregated_residuals[1:58],  
      type = "l",  
      xlab = "Date",  
      ylab = "Saudi Prediction vs. Actual: Full Model")  
lines(new_ts_covid_saudi$week[1:58],  
      compare_table$predict,  
      type = "l",  
      col = 'red')  
legend(x = "topright",legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_saudi$week[54:58],                                     # Draw first time series
     new_ts_covid_saudi$weekly_aggregated_residuals[54:58],
     type = "l",
     xlab = "Date",
     ylab = "Saudi Prediction vs. Acutal: Full Model")
lines(new_ts_covid_saudi$week[54:58],                                       # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE Saudi Full
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 1333.291
```

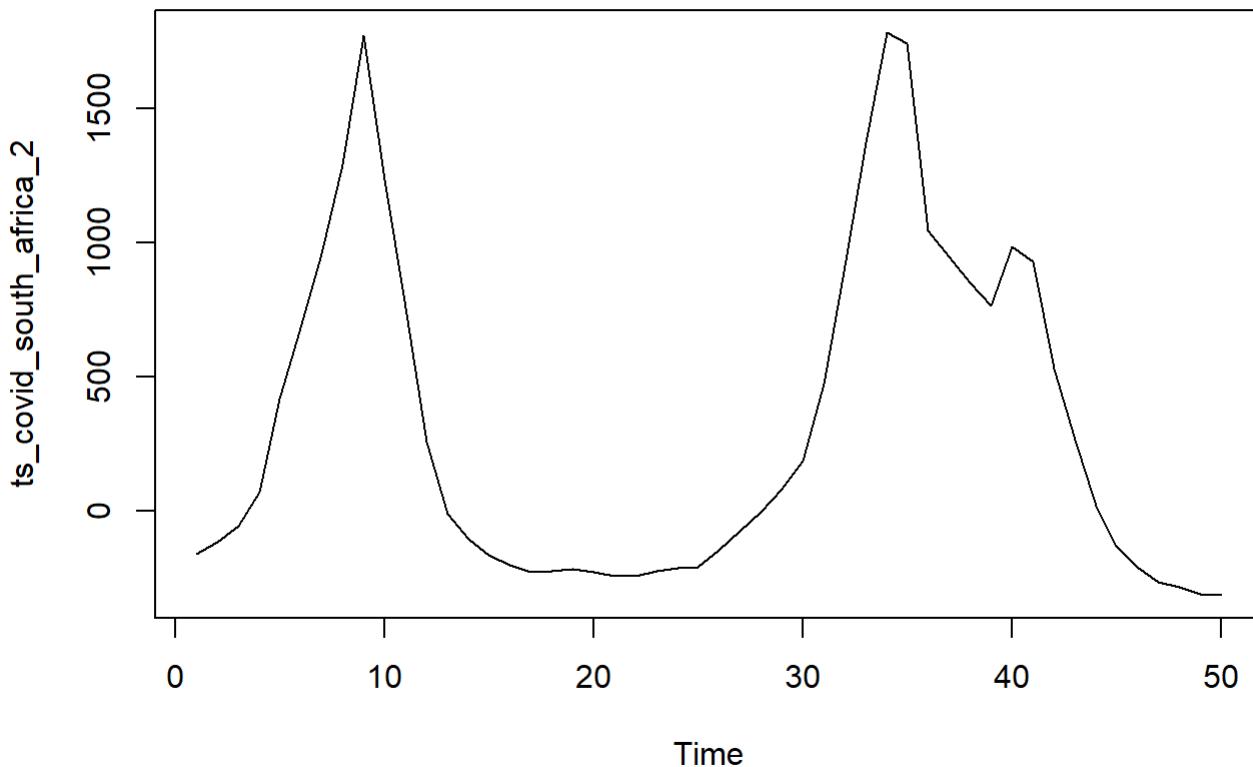
```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 2095.487
```

```
#South Africa Example on aggregate weekly
ts_covid_south_africa = ts(covid_south_africa$residuals,frequency = 7)
ts_covid_south_africa_2 = ts(colSums(matrix(ts_covid_south_africa, nrow=7)))

vaccine_policy = colSums(matrix(as.numeric(covid_south_africa$vaccination_policy), nrow=7))
new_ts_covid_south_africa = data.frame(1:50,ts_covid_south_africa_2,vaccine_policy)
names(new_ts_covid_south_africa) = c("week","weekly_aggregated_residuals","vaccination_policy")

plot.ts(ts_covid_south_africa_2)
```



```
regression_model = lm(weekly_aggregated_residuals~1, data = new_ts_covid_south_africa[1:45,])
summary(regression_model)
```

```

## 
## Call:
## lm(formula = weekly_aggregated_residuals ~ 1, data = new_ts_covid_south_africa[1:45,
##   ])
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -622.4 -539.7 -297.9  531.8 1399.1 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 382.07     93.38   4.091  0.00018 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 626.4 on 44 degrees of freedom

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: weekly_aggregated_residuals
##           Df  Sum Sq Mean Sq F value Pr(>F)    
## Residuals 44 17266207 392414

```

```

error = residuals(regression_model)
ts_error = ts(error)

```

```
#ARIMA on error
```

```

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)

```

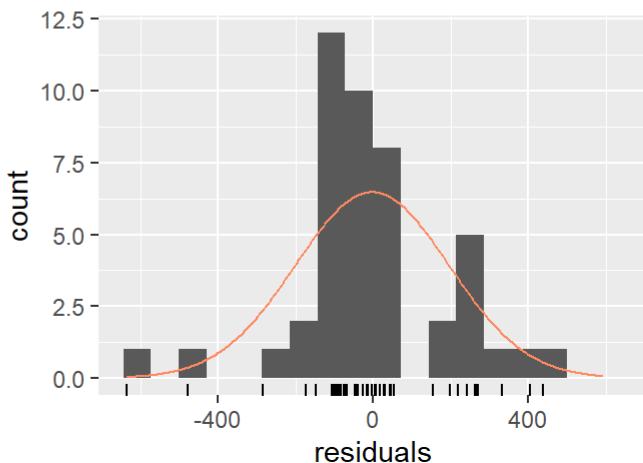
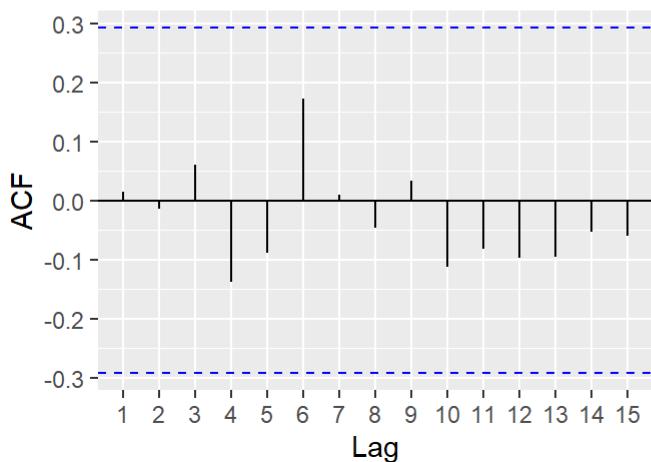
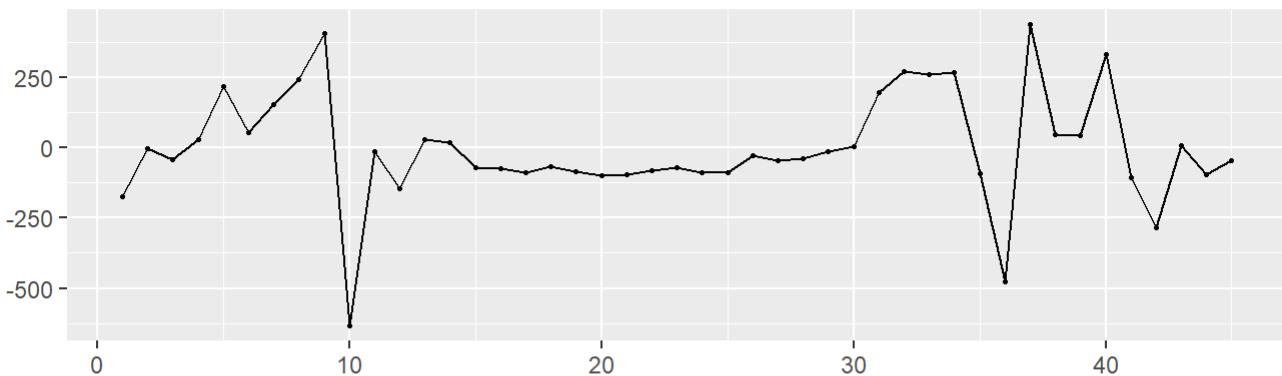
```

## Series: ts_error
## ARIMA(2,0,0) with zero mean
## 
## Coefficients:
##         ar1      ar2
##       1.4815  -0.6276
## s.e.  0.1106   0.1109
## 
## sigma^2 estimated as 40340: log likelihood=-302.83
## AIC=611.65  AICc=612.24  BIC=617.07
## 
## Training set error measures:
##               ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set -3.332915 196.3346 137.3029 25.37396 42.97551 0.7246983 0.01553128

```

```
checkresiduals(error_model1)
```

Residuals from ARIMA(2,0,0) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,0,0) with zero mean  
## Q* = 3.3935, df = 7, p-value = 0.8464  
##  
## Model df: 2. Total lags used: 9
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

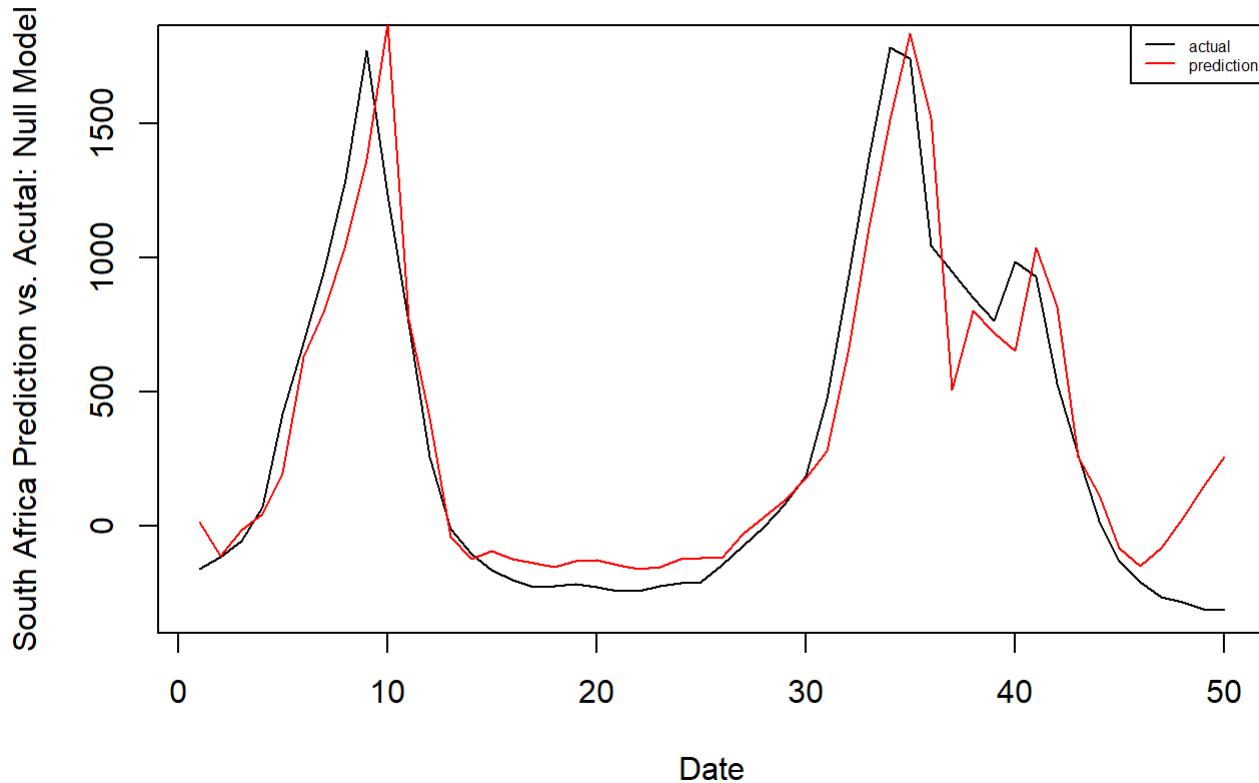
regression_prediction = predict(regression_model, newdata = new_ts_covid_south_africa[46:50,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

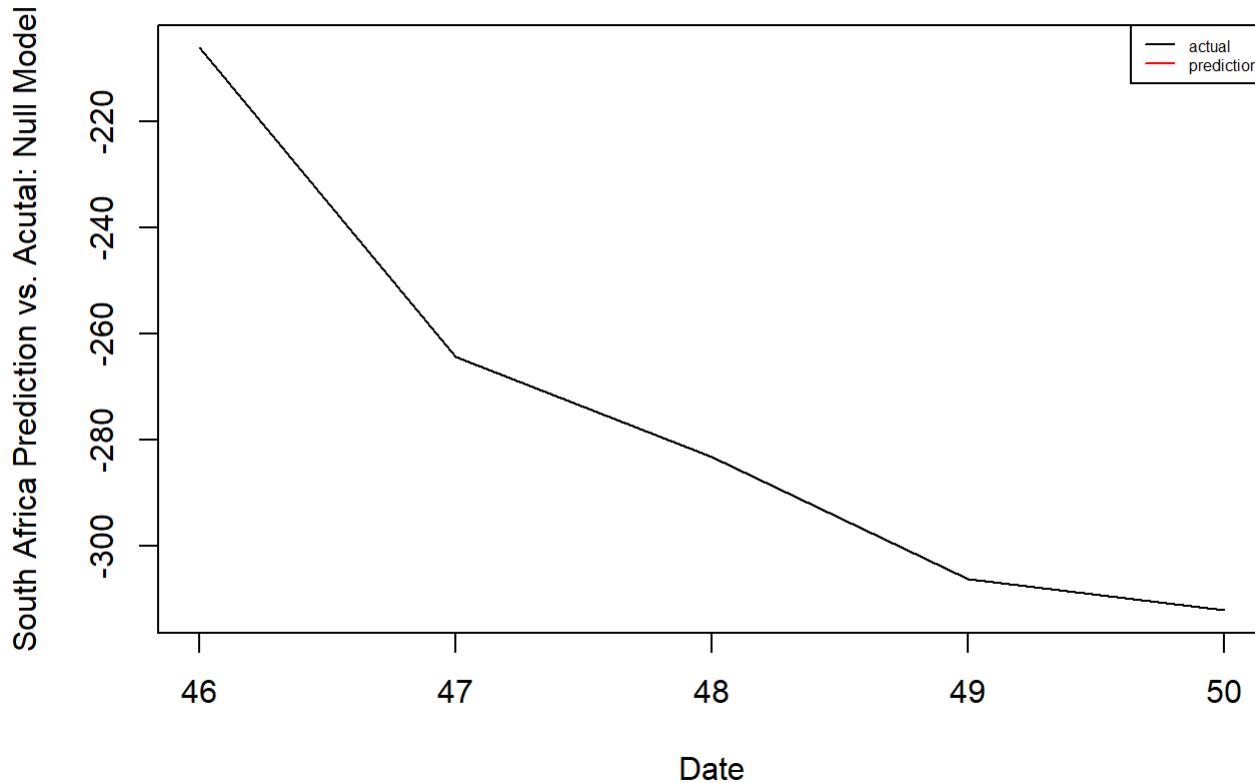
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_south_africa$weekly_aggregated_residuals[1:50])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_south_africa$week[1:50],                                     # Draw first time series
      new_ts_covid_south_africa$weekly_aggregated_residuals[1:50],
      type = "l",
      xlab = "Date",
      ylab = "South Africa Prediction vs. Actual: Null Model")
lines(new_ts_covid_south_africa$week[1:50],                                     # Draw second time serie
      s
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_south_africa$week[46:50],  
      # Draw first time serie  
      s  
      new_ts_covid_south_africa$weekly_aggregated_residuals[46:50],  
      type = "l",  
      xlab = "Date",  
      ylab = "South Africa Prediction vs. Acutal: Null Model")  
lines(new_ts_covid_south_africa$week[46:50],  
      # Draw second time serie  
      s  
      final_prediction_2,  
      type = "l",  
      col = "red")  
legend(x = "topright", legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE South Africa Null
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 38547.27
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 132918.7
```

```
regression_model = lm(weekly_aggregated_residuals~vaccination_policy, data = new_ts_covid_south_africa[1:45,])
summary(regression_model)
```

```
##
## Call:
## lm(formula = weekly_aggregated_residuals ~ vaccination_policy,
##     data = new_ts_covid_south_africa[1:45, ])
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -715.6 -540.5 -248.5  462.9 1503.0 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 205.368   185.586   1.107   0.275    
## vaccination_policy 9.088      8.255   1.101   0.277    
## 
## Residual standard error: 624.9 on 43 degrees of freedom
## Multiple R-squared:  0.02741,    Adjusted R-squared:  0.004793 
## F-statistic: 1.212 on 1 and 43 DF,  p-value: 0.2771
```

```
anova(regression_model)
```

```
## Analysis of Variance Table
##
## Response: weekly_aggregated_residuals
##              Df  Sum Sq Mean Sq F value Pr(>F)    
## vaccination_policy  1 473298 473298  1.2119 0.2771    
## Residuals          43 16792909 390533
```

```
error = residuals(regression_model)
ts_error = ts(error)

#ARIMA on error

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)
```

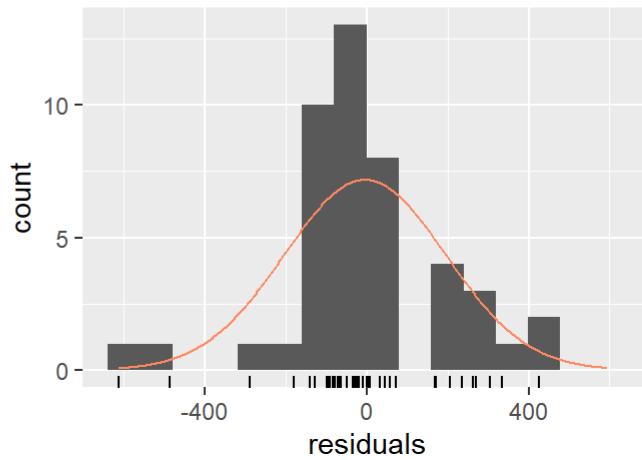
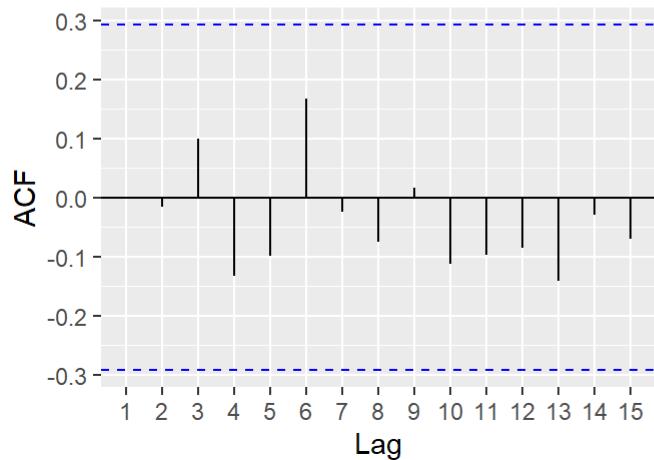
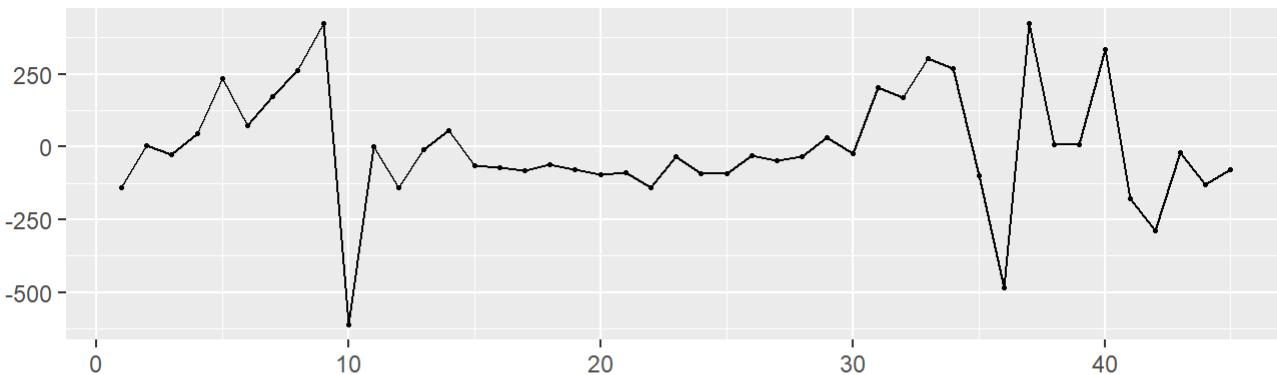
```

## Series: ts_error
## ARIMA(2,0,0) with zero mean
##
## Coefficients:
##      ar1     ar2
##      1.4750 -0.6229
## s.e.  0.1116  0.1122
##
## sigma^2 estimated as 40937: log likelihood=-303.14
## AIC=612.28   AICc=612.87   BIC=617.7
##
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -4.695147 197.7821 139.4343 41.77138 53.13291 0.7194772
##          ACF1
## Training set -0.002009171

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(2,0,0) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,0,0) with zero mean  
## Q* = 3.8106, df = 7, p-value = 0.8013  
##  
## Model df: 2. Total lags used: 9
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

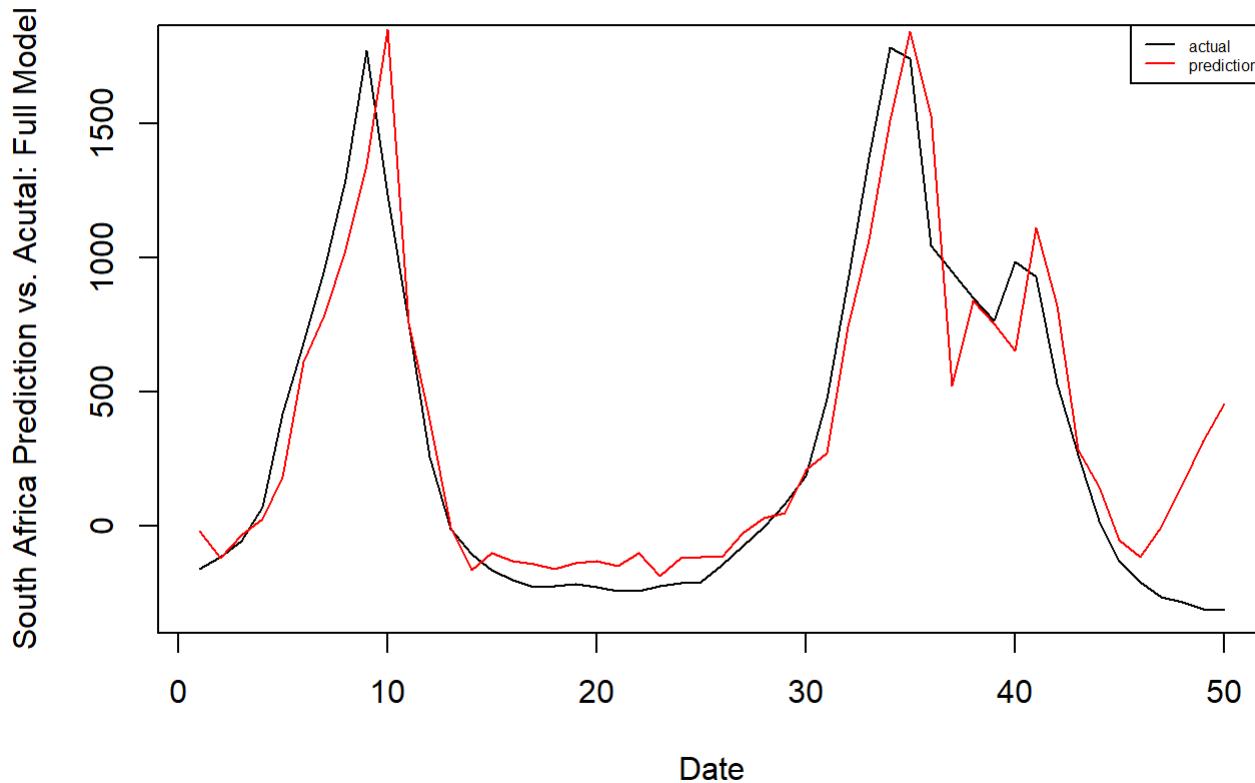
regression_prediction = predict(regression_model, newdata = new_ts_covid_south_africa[46:50,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

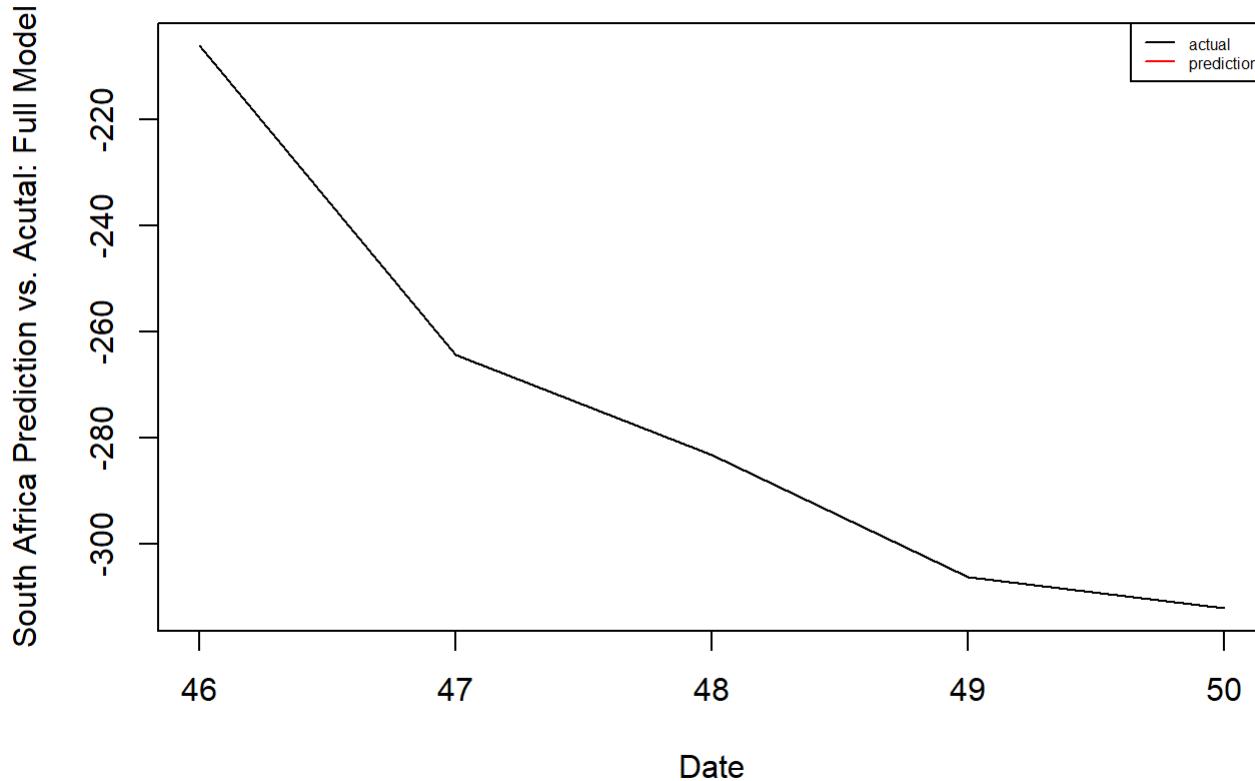
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_south_africa$weekly_aggregated_residuals[1:50])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_south_africa$week[1:50],  
      new_ts_covid_south_africa$weekly_aggregated_residuals[1:50],  
      type = "l",  
      xlab = "Date",  
      ylab = "South Africa Prediction vs. Actual: Full Model")  
lines(new_ts_covid_south_africa$week[1:50],  
      s  
      compare_table$predict,  
      type = "l",  
      col = 'red')  
legend(x = "topright",legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_south_africa$week[46:50],  
      # Draw first time serie  
      s  
      new_ts_covid_south_africa$weekly_aggregated_residuals[46:50],  
      type = "l",  
      xlab = "Date",  
      ylab = "South Africa Prediction vs. Acutal: Full Model")  
lines(new_ts_covid_south_africa$week[46:50],  
      # Draw second time serie  
      s  
      final_prediction_2,  
      type = "l",  
      col = "red")  
legend(x = "topright", legend=c("actual", "prediction"),  
      col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE South Africa Null
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 39117.78
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 249879.3
```

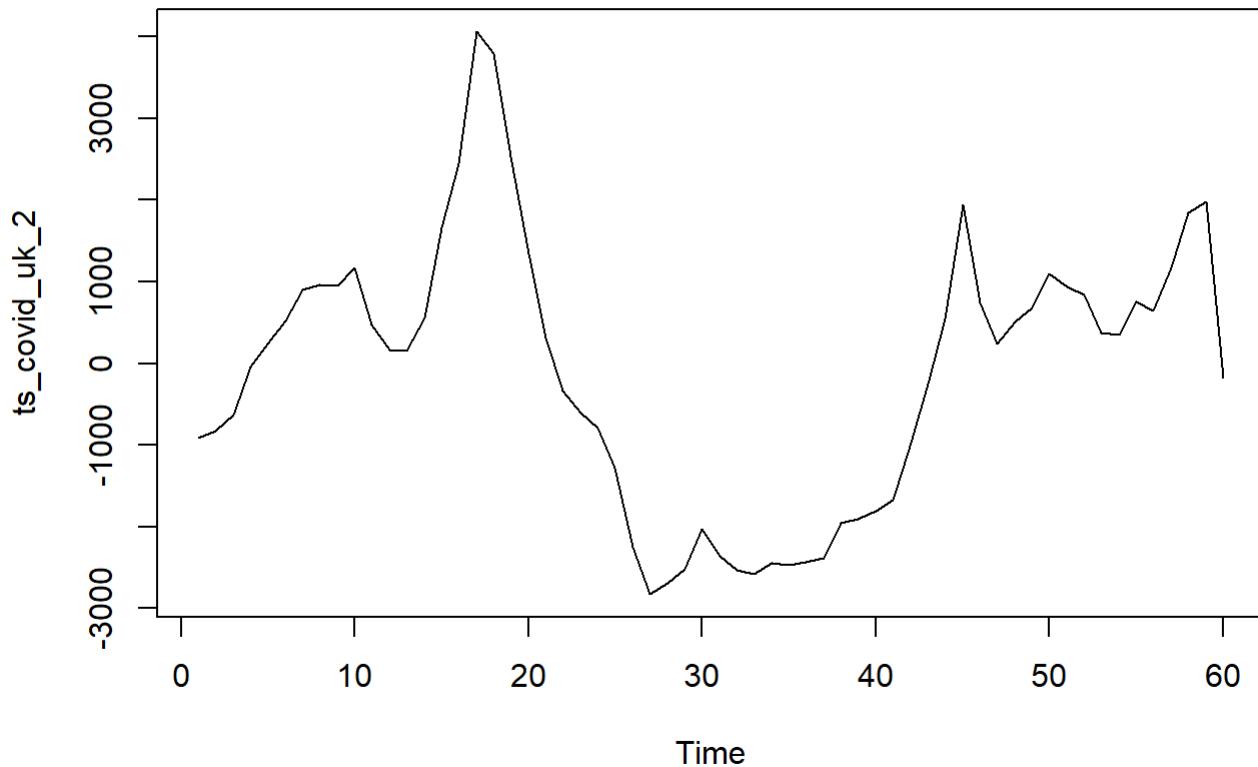
```
#United Kingdom Example on aggregate weekly
ts_covid_uk = ts(covid_uk$residuals,frequency = 7)
ts_covid_uk_2 = ts(colSums(matrix(ts_covid_uk, nrow=7)))
```

```
## Warning in matrix(ts_covid_uk, nrow = 7): data length [415] is not a sub-
## multiple or multiple of the number of rows [7]
```

```
vaccine_policy = colSums(matrix(as.numeric(covid_uk$vaccination_policy), nrow=7))
```

```
## Warning in matrix(as.numeric(covid_uk$vaccination_policy), nrow = 7): data  
## length [415] is not a sub-multiple or multiple of the number of rows [7]
```

```
new_ts_covid_uk = data.frame(1:60,ts_covid_uk_2,vaccine_policy)  
names(new_ts_covid_uk) = c("week","weekly_aggregated_residuals","vaccination_policy")  
  
plot.ts(ts_covid_uk_2)
```



```
regression_model = lm(weekly_aggregated_residuals~1, data = new_ts_covid_uk[1:55,])  
summary(regression_model)
```

```

## 
## Call:
## lm(formula = weekly_aggregated_residuals ~ 1, data = new_ts_covid_uk[1:55,
##   ])
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -2604.1 -1639.2  393.2 1022.1 4283.0 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -223.8     226.0   -0.99    0.326    
## 
## Residual standard error: 1676 on 54 degrees of freedom

```

```
anova(regression_model)
```

```

## Analysis of Variance Table
## 
## Response: weekly_aggregated_residuals
##           Df Sum Sq Mean Sq F value Pr(>F)    
## Residuals 54 151631465 2807990

```

```

error = residuals(regression_model)
ts_error = ts(error)

#ARIMA on error

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)

```

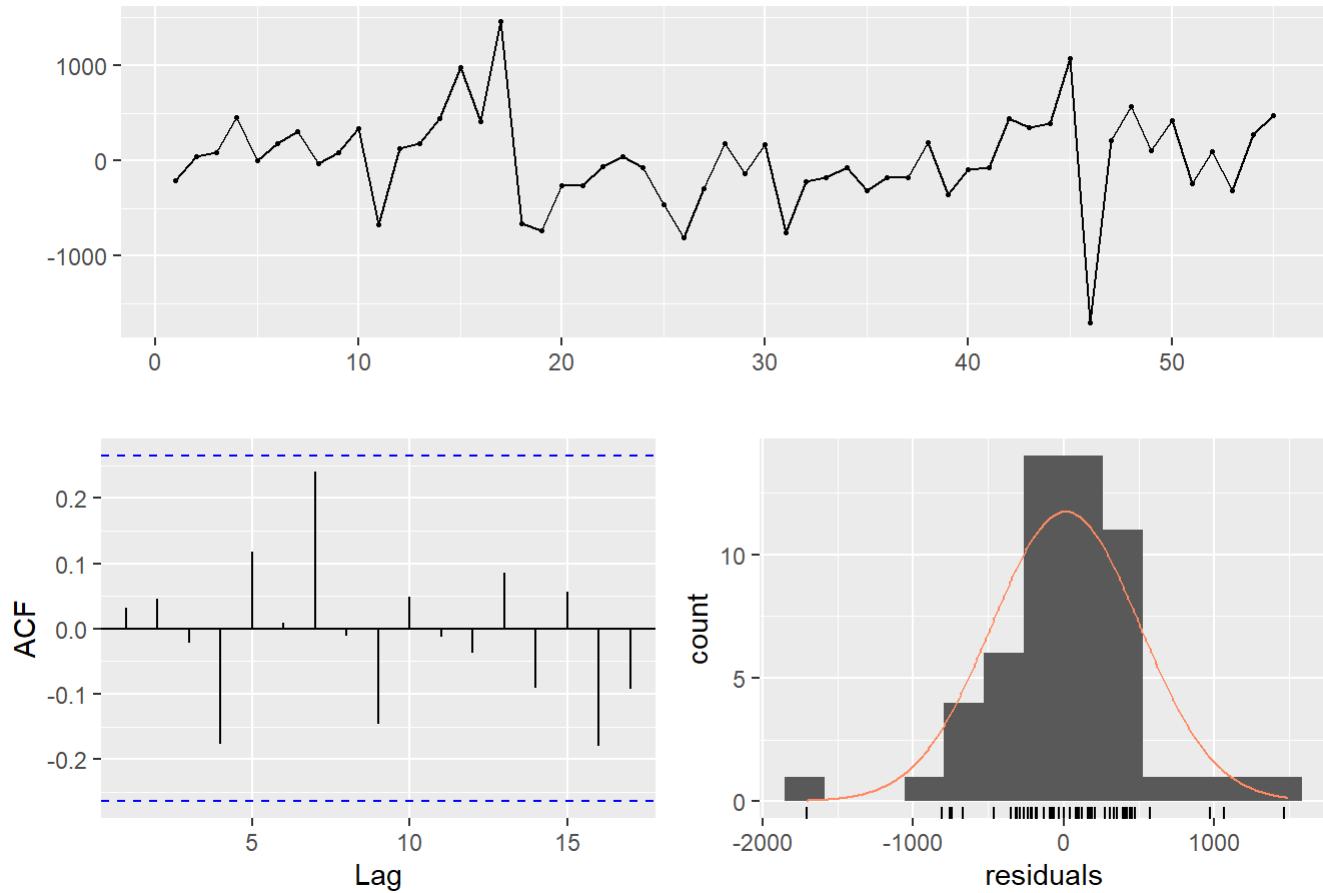
```

## Series: ts_error
## ARIMA(2,0,0) with zero mean
## 
## Coefficients:
##         ar1      ar2
##       1.4162  -0.5147
## s.e.  0.1122  0.1121
## 
## sigma^2 estimated as 248381:  log likelihood=-419.99
## AIC=845.98  AICc=846.45  BIC=852.01
## 
## Training set error measures:
##               ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 13.22947 489.233 352.619 -14.97154 60.87659 0.8096925 0.03213931

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(2,0,0) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,0,0) with zero mean  
## Q* = 8.4209, df = 8, p-value = 0.3935  
##  
## Model df: 2. Total lags used: 10
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

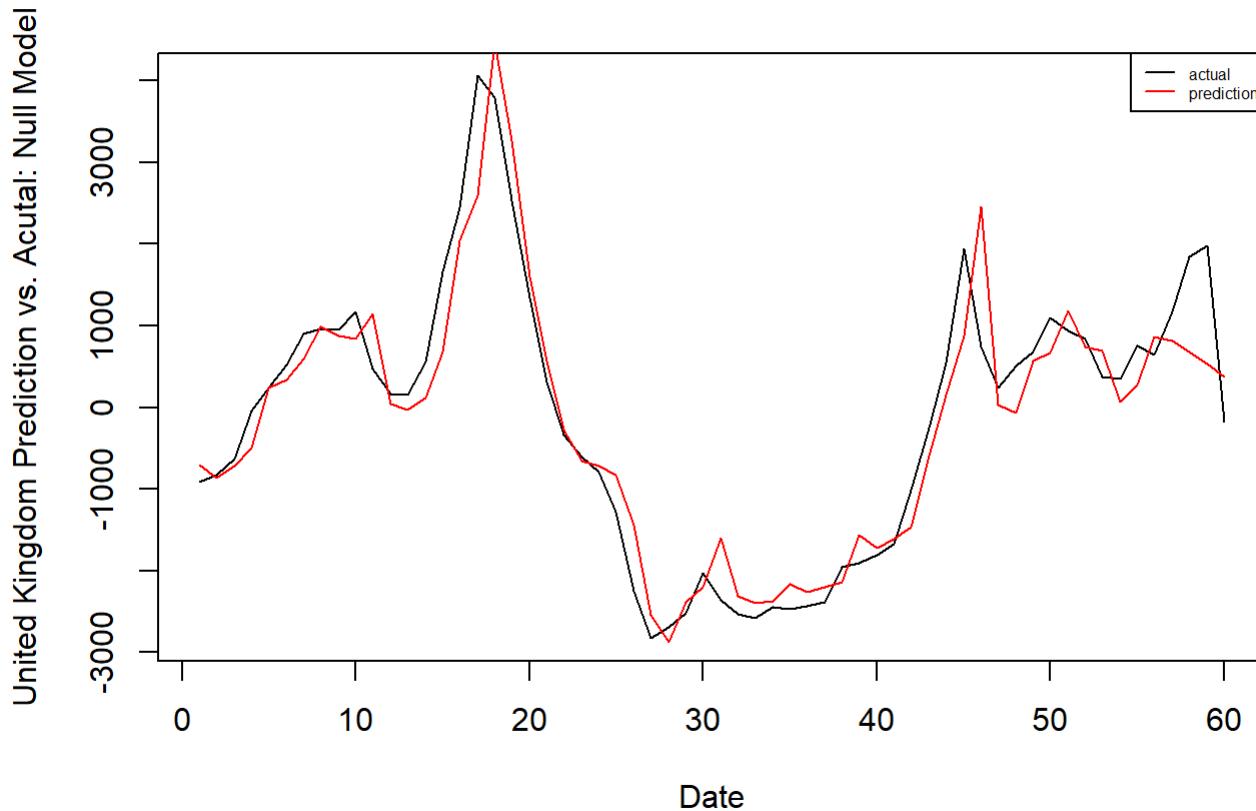
regression_prediction = predict(regression_model, newdata = new_ts_covid_uk[56:60,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

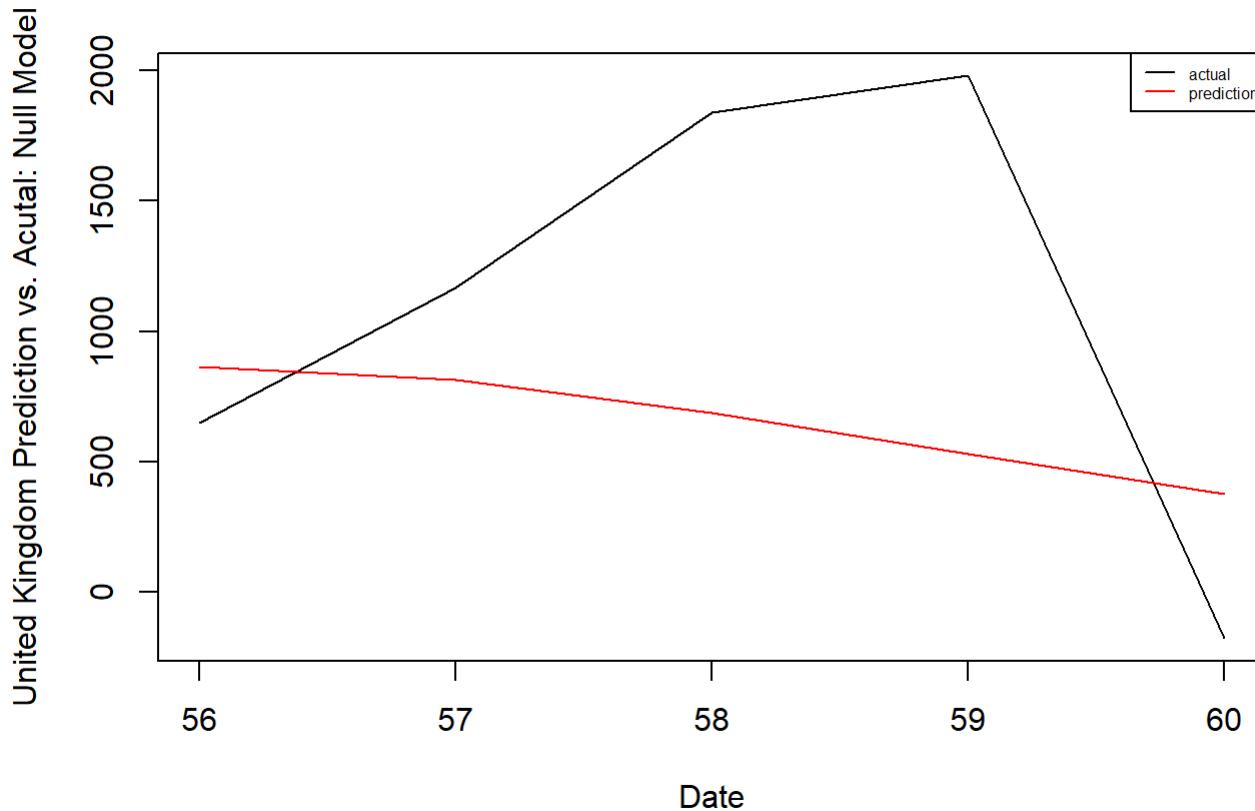
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_uk$weekly_aggregated_residuals[1:60])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_uk$week[1:60],                                     # Draw first time series
      new_ts_covid_uk$weekly_aggregated_residuals[1:60],
      type = "l",
      xlab = "Date",
      ylab = "United Kingdom Prediction vs. Actual: Null Model")
lines(new_ts_covid_uk$week[1:60],                                       # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_uk$week[56:60],                                     # Draw first time series
     new_ts_covid_uk$weekly_aggregated_residuals[56:60],
     type = "l",
     xlab = "Date",
     ylab = "United Kingdom Prediction vs. Acutal: Null Model")
lines(new_ts_covid_uk$week[56:60],                                       # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE UK Null
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 239348.9
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 781661
```

```
regression_model = lm(weekly_aggregated_residuals~vaccination_policy, data = new_ts_covid_uk[1:5,])
summary(regression_model)
```

```
##
## Call:
## lm(formula = weekly_aggregated_residuals ~ vaccination_policy,
##     data = new_ts_covid_uk[1:55, ])
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -2612.1 -1512.7   52.4  1011.9  4274.9 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  322.36    549.28   0.587   0.56    
## vaccination_policy -19.22     17.62  -1.090   0.28    
## 
## Residual standard error: 1673 on 53 degrees of freedom
## Multiple R-squared:  0.02195,    Adjusted R-squared:  0.003491 
## F-statistic: 1.189 on 1 and 53 DF,  p-value: 0.2804
```

```
anova(regression_model)
```

```
## Analysis of Variance Table
##
## Response: weekly_aggregated_residuals
##              Df  Sum Sq Mean Sq F value Pr(>F)    
## vaccination_policy 1 3327565 3327565 1.1892 0.2804  
## Residuals         53 148303900 2798187
```

```
error = residuals(regression_model)
ts_error = ts(error)

#ARIMA on error

error_model = auto.arima(ts_error)
fitted_error = fitted(error_model)
summary(error_model)
```

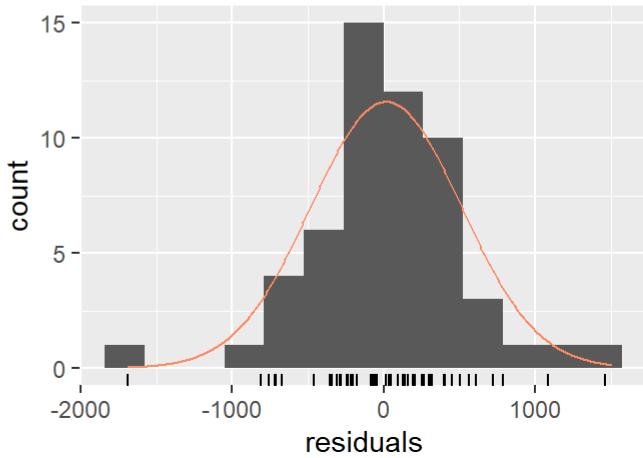
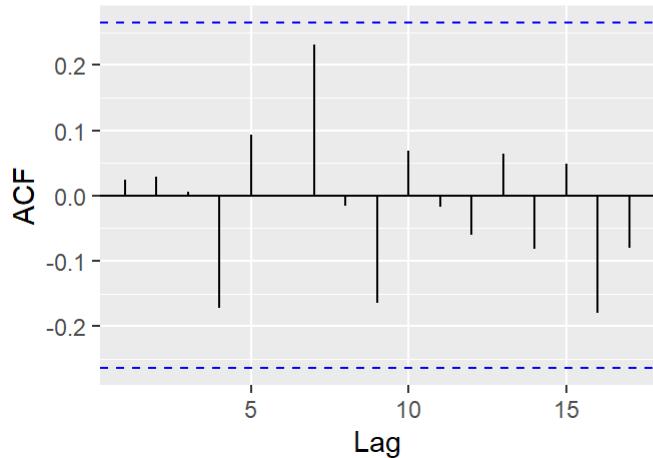
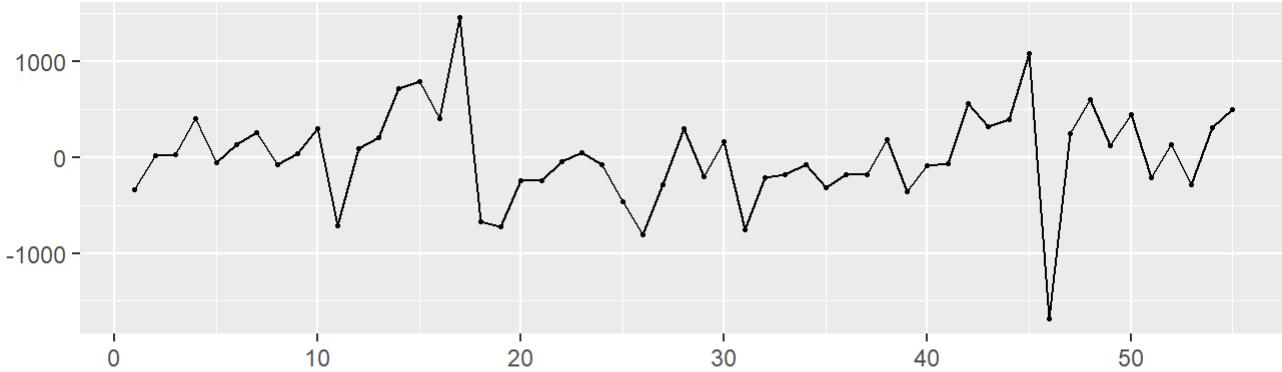
```

## Series: ts_error
## ARIMA(2,0,0) with zero mean
##
## Coefficients:
##      ar1     ar2
##      1.4251 -0.5282
## s.e.  0.1112  0.1114
##
## sigma^2 estimated as 253070: log likelihood=-420.51
## AIC=847.02   AICc=847.49   BIC=853.04
##
## Training set error measures:
##          ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 16.16142 493.8294 360.3937 5.124869 58.85776 0.8052135 0.02417055

```

```
checkresiduals(error_model)
```

Residuals from ARIMA(2,0,0) with zero mean



```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(2,0,0) with zero mean  
## Q* = 8.0939, df = 8, p-value = 0.4244  
##  
## Model df: 2. Total lags used: 10
```

```
#goodness of fit
regression_prediction = predict(regression_model)

final_prediction = regression_prediction + fitted_error

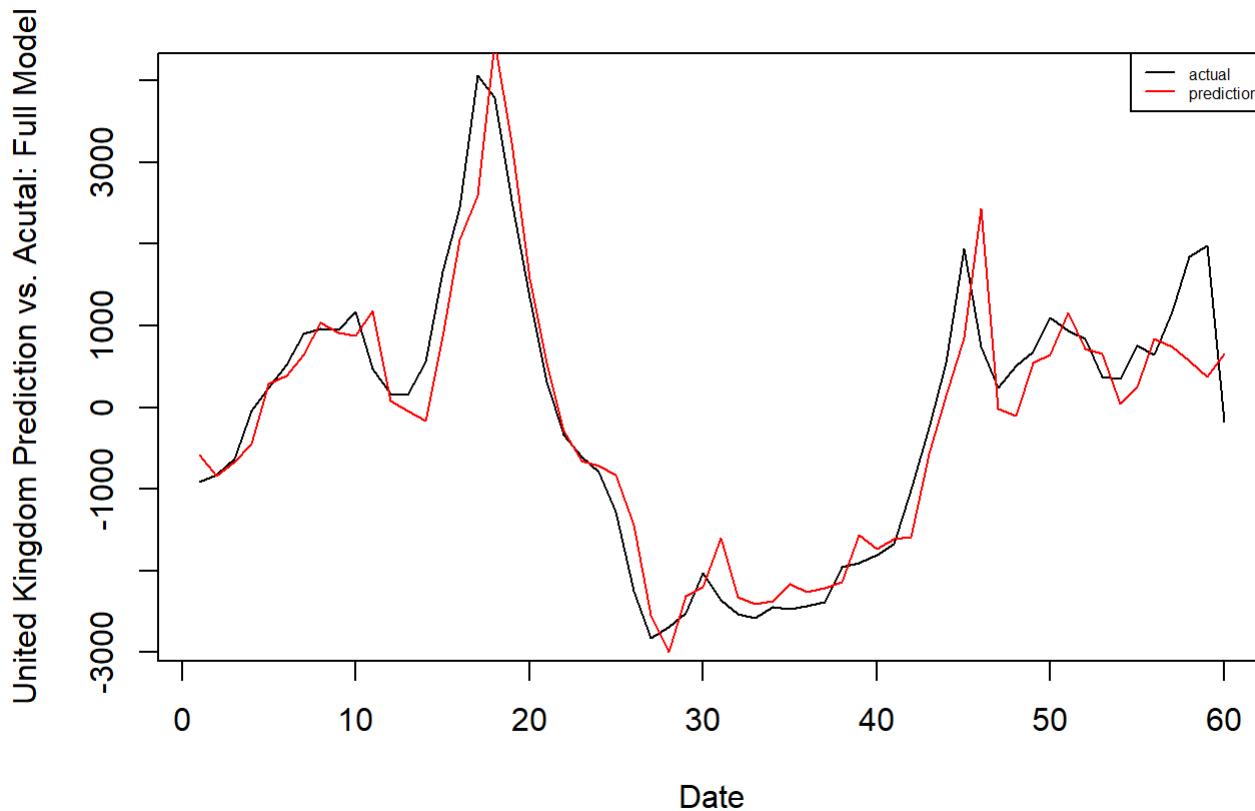
regression_prediction = predict(regression_model, newdata = new_ts_covid_uk[56:60,])
error_forecast = predict(error_model, n.ahead = 5)$pred

final_prediction_2 = regression_prediction + error_forecast

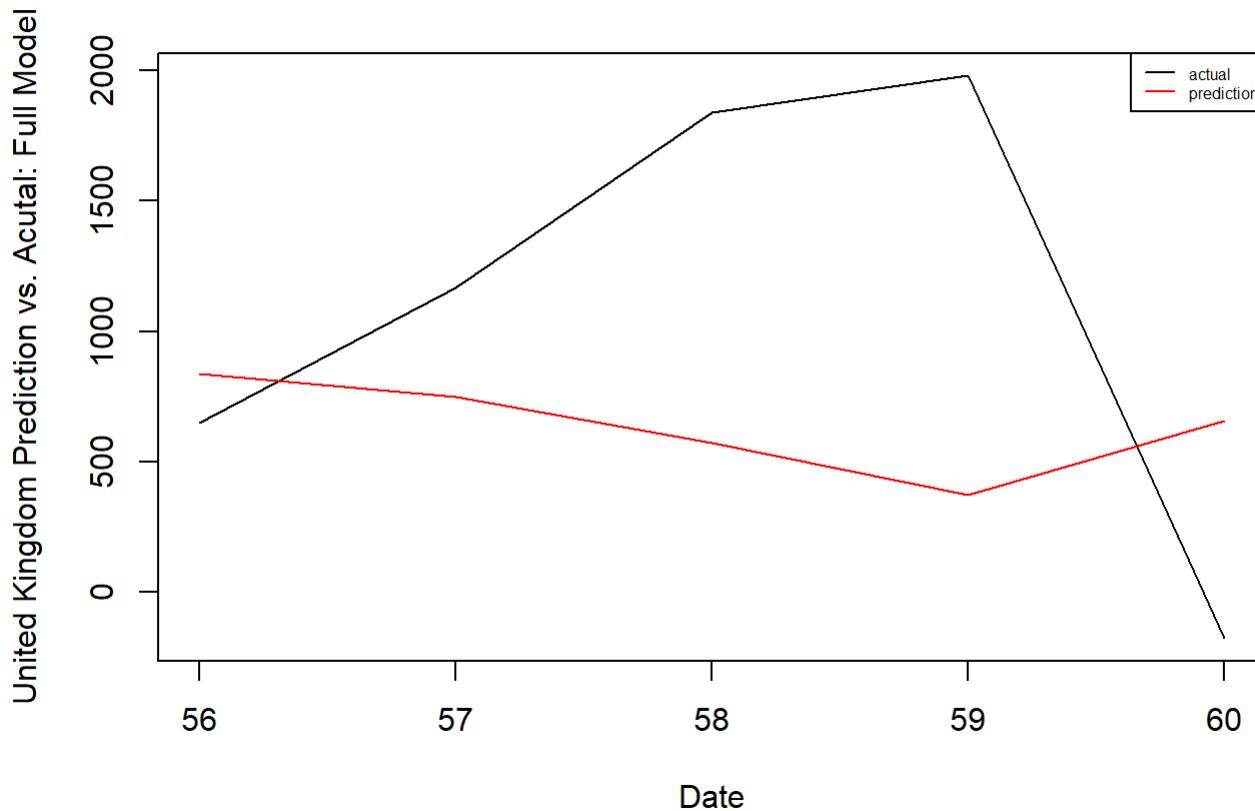
#compare prediction and actual

compare_table = data.frame(c(final_prediction,final_prediction_2),new_ts_covid_uk$weekly_aggregated_residuals[1:60])
colnames(compare_table) = c("predict","actual")

plot(new_ts_covid_uk$week[1:60],                                     # Draw first time series
      new_ts_covid_uk$weekly_aggregated_residuals[1:60],
      type = "l",
      xlab = "Date",
      ylab = "United Kingdom Prediction vs. Actual: Full Model")
lines(new_ts_covid_uk$week[1:60],                                       # Draw second time series
      compare_table$predict,
      type = "l",
      col = 'red')
legend(x = "topright",legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
plot(new_ts_covid_uk$week[56:60],                                     # Draw first time series
     new_ts_covid_uk$weekly_aggregated_residuals[56:60],
     type = "l",
     xlab = "Date",
     ylab = "United Kingdom Prediction vs. Acutal: Full Model")
lines(new_ts_covid_uk$week[56:60],                                       # Draw second time series
      final_prediction_2,
      type = "l",
      col = "red")
legend(x = "topright", legend=c("actual", "prediction"),
       col=c("black", "red"), lty=1, cex=0.5)
```



```
#Training MSE UK Full
SSE_train = sum((final_prediction - compare_table$actual[1:length(final_prediction)])^2)
MSE_train = SSE_train/length(final_prediction)
MSE_train
```

```
## [1] 243867.4
```

```
#Testing MSE
SSE_test = sum((final_prediction_2 - compare_table$actual[length(final_prediction)+1:length(final_prediction_2)])^2)
MSE_test = SSE_test/length(final_prediction_2)
MSE_test
```

```
## [1] 1021245
```