DS-6331 Time Series Project- Covid-19 Death Rate and the Influence of Vaccine and other variables

Tai Chowdhury and Simerpreet Reddy

4/09/2021

### Objective:

Analyze Covid-19 Death Rate(per day) and the influence of other factors- New cases per day, Vaccination Rates, number of hospitalizations and number of ICU patients per day. We will perform the following taks: 1) Performing Univariate analysis on ‘New Deaths per million’. 2) Build ARIMA model. 3) Build the following models: We will first build models using only ‘New Deaths per million’ and ‘New Vaccinations per million’ and then will analyze all other variables as well. We will also analyse lagged variables. Models to be built: i)Multivariate analysis ii) VAR models iii) MLP models iv) ensemble models using any of the above 2 models

### Data Set:

1. All Covid: Data Set with all data and related variables since the beginning of the pandemic.
2. Post Vaccine: Daily data since the first vaccine got administered in the US(12/21/20) through 03/26/2021 (96 observations). This is the data set we are bulding the models on.

### Data Set description:

96 observations with the following attributes: Date, New Cases per million, New Deaths per million, New ICU patients per million, New Hospitalizations per million, New Vaccinations per million

### Source:

Coronavirus (COVID-19) Vaccinations - Statistics and Research - Our World in Data. It is updated daily and includes data on confirmed cases, deaths, hospitalizations, testing, and vaccinations as well as other variables of potential interest.

#Include necessary libraries  
library(nnfor)  
library(forecast)  
library(vars)  
library(tswge)  
  
All\_Covid <- read.csv(file.choose(), header=TRUE)  
nrow(All\_Covid)

## [1] 429

Post\_Vaccine <- read.csv(file.choose(), header=TRUE)  
nrow(Post\_Vaccine)

## [1] 96

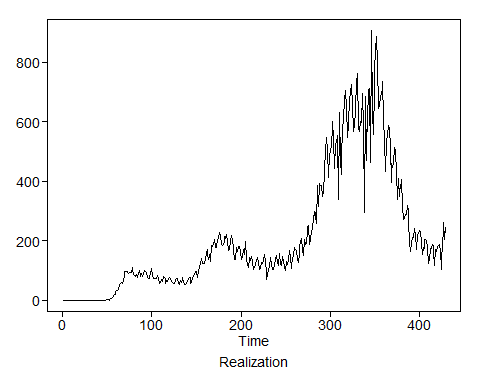
str(Post\_Vaccine)

## 'data.frame': 96 obs. of 7 variables:  
## $ date : Factor w/ 96 levels "1/1/2021","1/10/2021",..: 32 33 34 35 36 37 38 39 40 41 ...  
## $ location : Factor w/ 1 level "United States": 1 1 1 1 1 1 1 1 1 1 ...  
## $ new\_cases\_per\_million : num 602 598 694 587 295 ...  
## $ new\_deaths\_per\_million : num 5.85 10.25 10.33 8.79 4.23 ...  
## $ icu\_patients\_per\_million : num 79.9 80.8 81.4 80.7 80.3 ...  
## $ hosp\_patients\_per\_million : num 346 355 356 351 342 ...  
## $ new\_vaccinations\_smoothed\_per\_million: int 173 381 450 571 644 692 632 646 705 763 ...

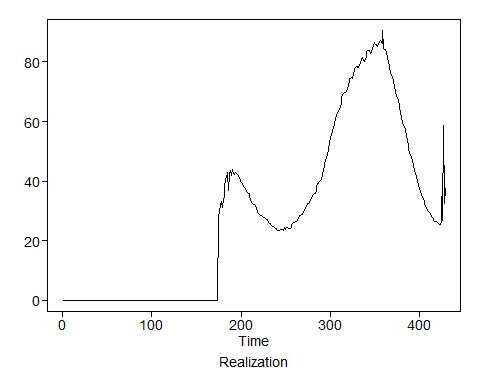
#Rolling Window Function def.n  
Rolling\_Window\_ASE = function(series, trainingSize, horizon, s, d, phis, thetas)  
{  
# trainingSize = 70  
# horizon = 12  
ASEHolder = numeric()  
# s = 10  
# d = 0  
# phis = phis  
# thetas = thetas  
for( i in 1:(length(series)-(trainingSize + horizon) + 1))  
{  
 forecasts = fore.aruma.wge(series[i:(i+(trainingSize-1))],phi = phis, theta = thetas, s = s, d = d,n.ahead = horizon)  
 ASE = mean((series[(trainingSize+i):(trainingSize+ i + (horizon) - 1)] - forecasts$f)^2)  
 ASEHolder[i] = ASE  
}  
ASEHolder  
hist(ASEHolder)  
WindowedASE = mean(ASEHolder)  
print(horizon)  
print(trainingSize)  
print("The Summary Statistics for the Rolling Window ASE Are:")  
print(summary(ASEHolder))  
print(paste("The Rolling Window ASE is: ",WindowedASE))  
return(WindowedASE)  
}

### EDA

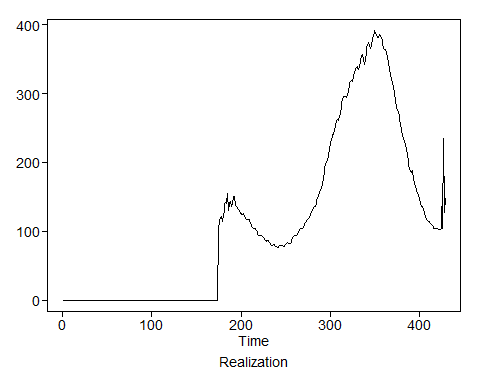
#Plotting time series for all variables since the beginning of the pandemic  
plotts.wge(All\_Covid$new\_cases\_per\_million)



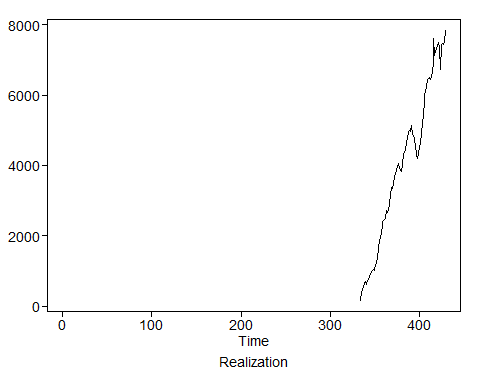
plotts.wge(All\_Covid$icu\_patients\_per\_million)



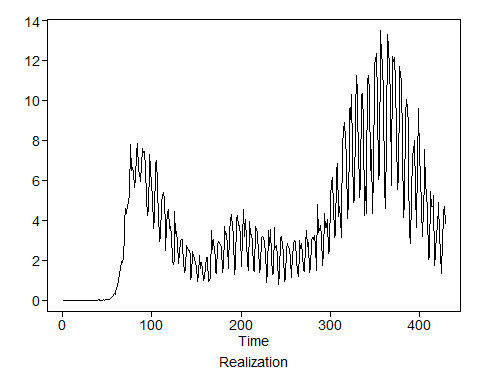
plotts.wge(All\_Covid$hosp\_patients\_per\_million)



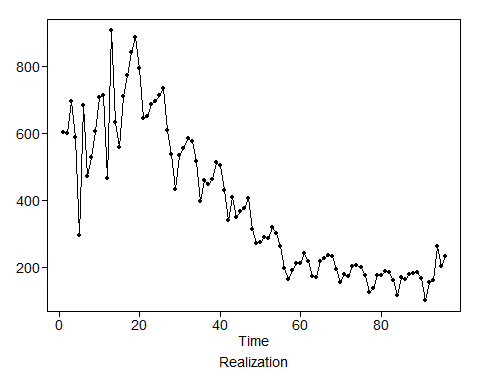
plotts.wge(All\_Covid$new\_vaccinations\_smoothed\_per\_million)



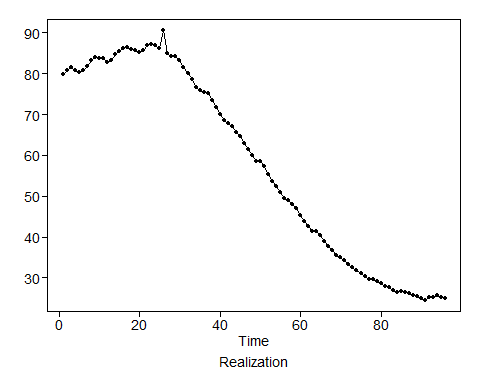
plotts.wge(All\_Covid$new\_deaths\_per\_million)



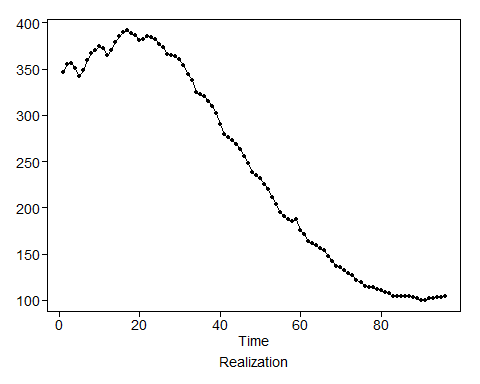
# Plotting time series for all variables Post Vaccine  
plotts.wge(Post\_Vaccine$new\_cases\_per\_million)



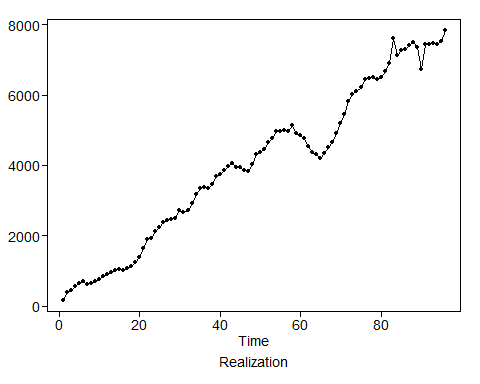
plotts.wge(Post\_Vaccine$icu\_patients\_per\_million)



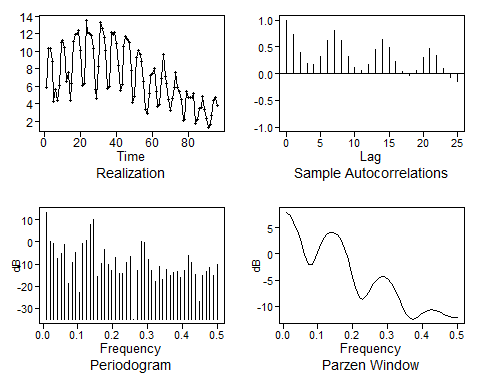
plotts.wge(Post\_Vaccine$hosp\_patients\_per\_million)



plotts.wge(Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million)



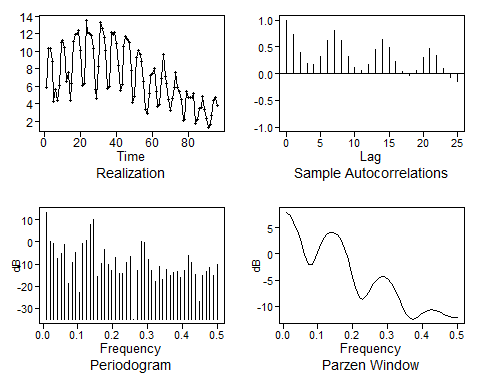
plotts.sample.wge(Post\_Vaccine$new\_deaths\_per\_million)



## $autplt  
## [1] 1.00000000 0.73924679 0.39319861 0.19136535 0.16605707  
## [6] 0.31555373 0.61873853 0.80447891 0.62269036 0.31663455  
## [11] 0.11686075 0.05243779 0.17584745 0.45067084 0.63267417  
## [16] 0.48665593 0.22107219 0.03480056 -0.04182223 0.05443786  
## [21] 0.30371212 0.46345231 0.33586596 0.09769245 -0.06443383  
## [26] -0.14549428  
##   
## $freq  
## [1] 0.01041667 0.02083333 0.03125000 0.04166667 0.05208333 0.06250000  
## [7] 0.07291667 0.08333333 0.09375000 0.10416667 0.11458333 0.12500000  
## [13] 0.13541667 0.14583333 0.15625000 0.16666667 0.17708333 0.18750000  
## [19] 0.19791667 0.20833333 0.21875000 0.22916667 0.23958333 0.25000000  
## [25] 0.26041667 0.27083333 0.28125000 0.29166667 0.30208333 0.31250000  
## [31] 0.32291667 0.33333333 0.34375000 0.35416667 0.36458333 0.37500000  
## [37] 0.38541667 0.39583333 0.40625000 0.41666667 0.42708333 0.43750000  
## [43] 0.44791667 0.45833333 0.46875000 0.47916667 0.48958333 0.50000000  
##   
## $db  
## [1] 13.38358616 -0.04328896 -0.75174151 -7.39894318 -5.36827217  
## [6] -1.37811985 -18.65942222 -9.16593121 -4.81889180 -22.75305069  
## [11] -0.77211585 0.54572886 7.95571565 9.93941684 -15.56200379  
## [16] -9.72457754 -3.38560102 -10.07724696 -12.72833848 -7.01784205  
## [21] -14.08874582 -14.14874488 -9.46702448 -6.70844920 -34.85089971  
## [26] -12.74161900 0.28877836 -0.24148581 -8.06338124 -12.89052420  
## [31] -18.01705539 -10.88015895 -17.18122820 -12.52831066 -14.98346501  
## [36] -13.80789350 -13.39160766 -15.87723676 -12.76930129 -5.93323504  
## [41] -9.12450411 -14.86457983 -26.89294306 -15.26595997 -13.47613264  
## [46] -11.75981173 -15.29994077 -10.07454685  
##   
## $dbz  
## [1] 7.9788031 7.2672461 6.0743600 4.3973430 2.2678901  
## [6] -0.1151091 -1.9929271 -2.0984330 -0.6209184 1.1639919  
## [11] 2.6360468 3.6397567 4.1442404 4.1419169 3.6253326  
## [16] 2.5826368 1.0007549 -1.1173108 -3.6908808 -6.3778929  
## [21] -8.3117954 -8.6721389 -7.8674247 -6.7118290 -5.6323643  
## [26] -4.8234984 -4.3885502 -4.3801163 -4.8170740 -5.6923301  
## [31] -6.9669555 -8.5427426 -10.2087892 -11.6078159 -12.3780204  
## [36] -12.4590336 -12.1037429 -11.5984253 -11.1315678 -10.8111772  
## [41] -10.6914415 -10.7802035 -11.0384540 -11.3843171 -11.7155324  
## [46] -11.9537425 -12.0792430 -12.1155303

### ARIMA Model 1 (with seasonal component, s = 7)

#ARIMA Models as this data is assumed non-stationary  
# Step 1 Check if data is white noise  
plotts.sample.wge(Post\_Vaccine$new\_deaths\_per\_million)

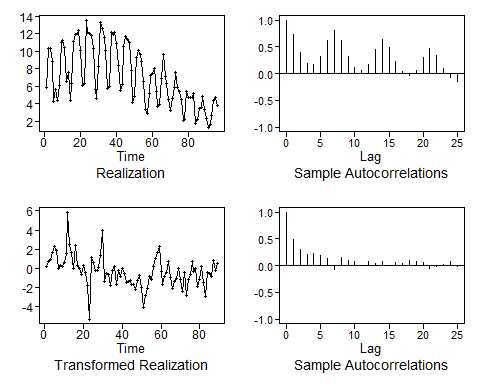
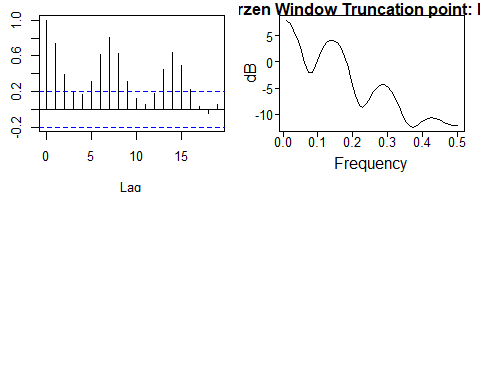


## $autplt  
## [1] 1.00000000 0.73924679 0.39319861 0.19136535 0.16605707  
## [6] 0.31555373 0.61873853 0.80447891 0.62269036 0.31663455  
## [11] 0.11686075 0.05243779 0.17584745 0.45067084 0.63267417  
## [16] 0.48665593 0.22107219 0.03480056 -0.04182223 0.05443786  
## [21] 0.30371212 0.46345231 0.33586596 0.09769245 -0.06443383  
## [26] -0.14549428  
##   
## $freq  
## [1] 0.01041667 0.02083333 0.03125000 0.04166667 0.05208333 0.06250000  
## [7] 0.07291667 0.08333333 0.09375000 0.10416667 0.11458333 0.12500000  
## [13] 0.13541667 0.14583333 0.15625000 0.16666667 0.17708333 0.18750000  
## [19] 0.19791667 0.20833333 0.21875000 0.22916667 0.23958333 0.25000000  
## [25] 0.26041667 0.27083333 0.28125000 0.29166667 0.30208333 0.31250000  
## [31] 0.32291667 0.33333333 0.34375000 0.35416667 0.36458333 0.37500000  
## [37] 0.38541667 0.39583333 0.40625000 0.41666667 0.42708333 0.43750000  
## [43] 0.44791667 0.45833333 0.46875000 0.47916667 0.48958333 0.50000000  
##   
## $db  
## [1] 13.38358616 -0.04328896 -0.75174151 -7.39894318 -5.36827217  
## [6] -1.37811985 -18.65942222 -9.16593121 -4.81889180 -22.75305069  
## [11] -0.77211585 0.54572886 7.95571565 9.93941684 -15.56200379  
## [16] -9.72457754 -3.38560102 -10.07724696 -12.72833848 -7.01784205  
## [21] -14.08874582 -14.14874488 -9.46702448 -6.70844920 -34.85089971  
## [26] -12.74161900 0.28877836 -0.24148581 -8.06338124 -12.89052420  
## [31] -18.01705539 -10.88015895 -17.18122820 -12.52831066 -14.98346501  
## [36] -13.80789350 -13.39160766 -15.87723676 -12.76930129 -5.93323504  
## [41] -9.12450411 -14.86457983 -26.89294306 -15.26595997 -13.47613264  
## [46] -11.75981173 -15.29994077 -10.07454685  
##   
## $dbz  
## [1] 7.9788031 7.2672461 6.0743600 4.3973430 2.2678901  
## [6] -0.1151091 -1.9929271 -2.0984330 -0.6209184 1.1639919  
## [11] 2.6360468 3.6397567 4.1442404 4.1419169 3.6253326  
## [16] 2.5826368 1.0007549 -1.1173108 -3.6908808 -6.3778929  
## [21] -8.3117954 -8.6721389 -7.8674247 -6.7118290 -5.6323643  
## [26] -4.8234984 -4.3885502 -4.3801163 -4.8170740 -5.6923301  
## [31] -6.9669555 -8.5427426 -10.2087892 -11.6078159 -12.3780204  
## [36] -12.4590336 -12.1037429 -11.5984253 -11.1315678 -10.8111772  
## [41] -10.6914415 -10.7802035 -11.0384540 -11.3843171 -11.7155324  
## [46] -11.9537425 -12.0792430 -12.1155303

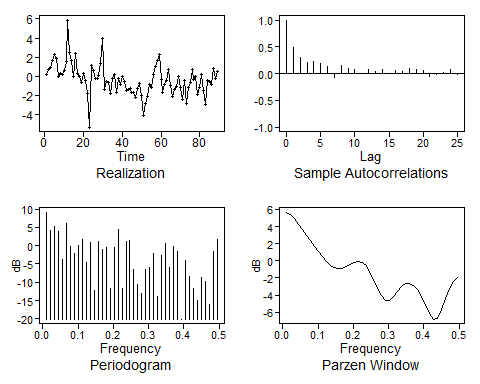
acf(Post\_Vaccine$new\_deaths\_per\_million)   
parzen.wge(Post\_Vaccine$new\_deaths\_per\_million)

## $freq  
## [1] 0.01041667 0.02083333 0.03125000 0.04166667 0.05208333 0.06250000  
## [7] 0.07291667 0.08333333 0.09375000 0.10416667 0.11458333 0.12500000  
## [13] 0.13541667 0.14583333 0.15625000 0.16666667 0.17708333 0.18750000  
## [19] 0.19791667 0.20833333 0.21875000 0.22916667 0.23958333 0.25000000  
## [25] 0.26041667 0.27083333 0.28125000 0.29166667 0.30208333 0.31250000  
## [31] 0.32291667 0.33333333 0.34375000 0.35416667 0.36458333 0.37500000  
## [37] 0.38541667 0.39583333 0.40625000 0.41666667 0.42708333 0.43750000  
## [43] 0.44791667 0.45833333 0.46875000 0.47916667 0.48958333 0.50000000  
##   
## $pzgram  
## [1] 7.9788031 7.2672461 6.0743600 4.3973430 2.2678901  
## [6] -0.1151091 -1.9929271 -2.0984330 -0.6209184 1.1639919  
## [11] 2.6360468 3.6397567 4.1442404 4.1419169 3.6253326  
## [16] 2.5826368 1.0007549 -1.1173108 -3.6908808 -6.3778929  
## [21] -8.3117954 -8.6721389 -7.8674247 -6.7118290 -5.6323643  
## [26] -4.8234984 -4.3885502 -4.3801163 -4.8170740 -5.6923301  
## [31] -6.9669555 -8.5427426 -10.2087892 -11.6078159 -12.3780204  
## [36] -12.4590336 -12.1037429 -11.5984253 -11.1315678 -10.8111772  
## [41] -10.6914415 -10.7802035 -11.0384540 -11.3843171 -11.7155324  
## [46] -11.9537425 -12.0792430 -12.1155303

# This data set is not white noise, Autocorrelation are correlated. Damped sinusoidal indicating seasonality.  
  
#Bulding Models  
#ARIMA Model 1 Only Seasonality   
  
#Take out the s=7 (1-B^7) term from the model  
pv\_s7 <- artrans.wge(Post\_Vaccine$new\_deaths\_per\_million,phi.tr = c(rep(0,6),1)) #Looks statinary



#Plot the timeseries without the seasonal component  
plotts.sample.wge(pv\_s7)



## $autplt  
## [1] 1.000000000 0.492180488 0.299766772 0.213089243 0.229396186  
## [6] 0.183677332 0.141909332 -0.067354143 0.147042664 0.091131712  
## [11] 0.081067140 0.011537174 0.078199694 0.047485936 0.075193179  
## [16] 0.006739226 0.055923563 0.032423785 0.100260130 0.071144592  
## [21] 0.059250540 -0.056189562 -0.016383601 0.017998032 0.080795779  
## [26] -0.019187692  
##   
## $freq  
## [1] 0.01123596 0.02247191 0.03370787 0.04494382 0.05617978 0.06741573  
## [7] 0.07865169 0.08988764 0.10112360 0.11235955 0.12359551 0.13483146  
## [13] 0.14606742 0.15730337 0.16853933 0.17977528 0.19101124 0.20224719  
## [19] 0.21348315 0.22471910 0.23595506 0.24719101 0.25842697 0.26966292  
## [25] 0.28089888 0.29213483 0.30337079 0.31460674 0.32584270 0.33707865  
## [31] 0.34831461 0.35955056 0.37078652 0.38202247 0.39325843 0.40449438  
## [37] 0.41573034 0.42696629 0.43820225 0.44943820 0.46067416 0.47191011  
## [43] 0.48314607 0.49438202  
##   
## $db  
## [1] 9.16405197 4.07488662 5.37430756 3.96627628 -3.85152335  
## [6] 6.19637279 -0.13694563 -2.21294200 0.05704151 1.68546786  
## [11] -4.53904475 0.96760308 -12.22322189 1.10724572 -1.14304363  
## [16] -0.44439686 -11.75326374 -0.58249618 4.34938832 -11.66188244  
## [21] 1.09207923 1.34202972 -6.56859978 -10.56045389 -12.94669805  
## [26] -6.45976335 -6.04322323 -2.25645380 -13.80650261 -2.60128366  
## [31] 0.57694076 -6.01416586 -0.34343245 -1.55988204 -20.09144758  
## [36] -3.94552598 -8.40304597 -11.77125917 -14.88780164 -8.68176633  
## [41] -9.82368199 -16.16512670 -1.54036469 1.65378534  
##   
## $dbz  
## [1] 5.67582676 5.36873305 4.89287908 4.29804713 3.64187226  
## [6] 2.97429944 2.32347514 1.69417607 1.08141145 0.48923046  
## [11] -0.05733847 -0.51246906 -0.82169108 -0.94386270 -0.87603907  
## [16] -0.66624378 -0.40246773 -0.18576173 -0.10648527 -0.23305597  
## [21] -0.60924722 -1.25112607 -2.13402073 -3.15998188 -4.11233134  
## [26] -4.67768251 -4.65985654 -4.18731419 -3.56438998 -3.02980270  
## [31] -2.70606390 -2.64369362 -2.86026276 -3.35815200 -4.12431621  
## [36] -5.10511538 -6.13343801 -6.82365751 -6.69662320 -5.73289342  
## [41] -4.43283409 -3.25053835 -2.40281570 -1.96756367

#Overfitting  
est.ar.wge(Post\_Vaccine$new\_deaths\_per\_million, p=10, type='burg')

##   
## Coefficients of Original polynomial:   
## 0.5536 -0.0193 0.0857 -0.0049 -0.0853 0.2286 0.5595 -0.1781 -0.1493 -0.0437   
##   
## Factor Roots Abs Recip System Freq   
## 1-1.2402B+0.9784B^2 0.6338+-0.7877i 0.9892 0.1422  
## 1-0.9788B 1.0217 0.9788 0.0000  
## 1+0.4129B+0.9012B^2 -0.2291+-1.0282i 0.9493 0.2849  
## 1+1.4865B+0.6800B^2 -1.0930+-0.5253i 0.8246 0.4287  
## 1-0.7062B 1.4161 0.7062 0.0000  
## 1+0.4722B+0.1054B^2 -2.2401+-2.1143i 0.3246 0.3796  
##   
##

## $phi  
## [1] 0.553589364 -0.019316268 0.085735201 -0.004904562 -0.085261659  
## [6] 0.228646043 0.559491742 -0.178106108 -0.149342180 -0.043678239  
##   
## $res  
## [1] -0.38095255 1.27335697 -0.24728887 0.76000753 -1.77349638  
## [6] 1.63853732 -0.81103872 0.04805977 1.41958216 0.05784013  
## [11] 1.73449922 -0.06842955 1.74204304 -1.76003728 0.06261579  
## [16] 0.88054517 0.02248746 1.96681789 3.67320919 0.63766902  
## [21] -0.80325048 -1.09238408 3.13700420 -1.25158741 0.02176969  
## [26] -0.45087442 0.75250787 -0.71883394 -2.15462603 -2.23615266  
## [31] 3.09208879 0.62409106 -0.14823268 0.17647564 -0.40724456  
## [36] 1.42482035 3.33549621 -2.44949107 -0.12957686 -0.69072969  
## [41] -0.69496304 0.33385322 -0.05183209 -0.01171220 -0.01472176  
## [46] -0.23107782 0.60435853 -0.46737714 -1.28157399 -0.68339391  
## [51] 0.39101605 -0.88225262 -0.78416953 -0.80903892 -0.05143122  
## [56] -0.33335934 -1.57569241 -2.04379326 -0.71187965 -0.48344840  
## [61] 0.65961699 -0.61054584 0.61908603 0.76929206 1.31790215  
## [66] 1.61734261 -2.07262104 -1.11191683 -0.68789290 -0.01779155  
## [71] 0.93743785 -1.30113059 -0.74879773 -1.13695101 0.09418679  
## [76] 0.42199793 -1.45174988 -1.42245830 0.49450707 -1.99699263  
## [81] -0.02449981 -0.06989499 1.28828350 -1.15273942 -0.17404870  
## [86] -1.31088245 -0.77159348 0.89529843 -1.90516378 -1.43581293  
## [91] -0.01879617 -0.09983680 -0.25856235 0.67511943 -0.22147557  
## [96] -0.01569065  
##   
## $avar  
## [1] 1.491563  
##   
## $aic  
## [1] 0.6289909  
##   
## $aicc  
## [1] 1.688981  
##   
## $bic  
## [1] 0.9228225

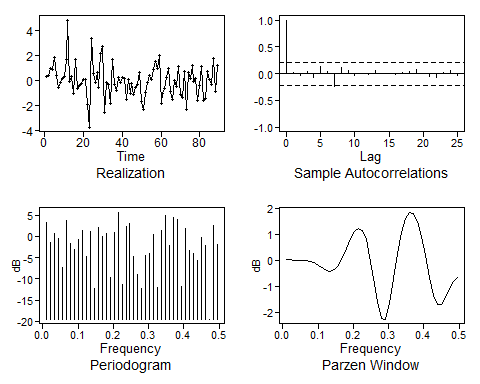
#aic with defaults  
aic.wge(pv\_s7) #=p=4, q=1

## $type  
## [1] "aic"  
##   
## $value  
## [1] 0.6759946  
##   
## $p  
## [1] 4  
##   
## $q  
## [1] 1  
##   
## $phi  
## [1] -0.42786774 0.43299184 0.04853331 0.17183275  
##   
## $theta  
## [1] -0.9471465  
##   
## $vara  
## [1] 1.718004

arma\_s7\_1 <- est.arma.wge(pv\_s7, p=4, q=1)

##   
## Coefficients of Original polynomial:   
## -0.4279 0.4330 0.0485 0.1718   
##   
## Factor Roots Abs Recip System Freq   
## 1+0.9915B -1.0086 0.9915 0.5000  
## 1-0.7219B 1.3853 0.7219 0.0000  
## 1+0.1582B+0.2401B^2 -0.3296+-2.0141i 0.4900 0.2758  
##   
##

#residuals  
plotts.sample.wge(arma\_s7\_1$res,arlimits = T) #Residuals not white noise



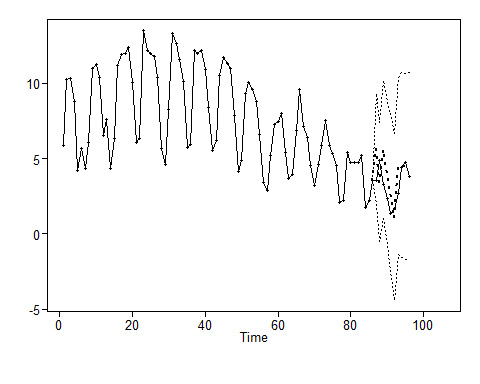
## $autplt  
## [1] 1.0000000000 0.0133111056 -0.0377762716 0.0360261071 -0.0750652450  
## [6] 0.1296215473 0.0497015005 -0.2483066169 0.1217109200 0.0610226774  
## [11] -0.0337191131 0.0016071293 -0.0238589853 0.0646591851 -0.0016082686  
## [16] 0.0002678429 -0.0237093208 0.0219909167 0.0372003324 0.0857732595  
## [21] 0.0019404344 -0.0518530464 -0.0721966394 0.0313406865 0.0566506492  
## [26] -0.0380707823  
##   
## $freq  
## [1] 0.01123596 0.02247191 0.03370787 0.04494382 0.05617978 0.06741573  
## [7] 0.07865169 0.08988764 0.10112360 0.11235955 0.12359551 0.13483146  
## [13] 0.14606742 0.15730337 0.16853933 0.17977528 0.19101124 0.20224719  
## [19] 0.21348315 0.22471910 0.23595506 0.24719101 0.25842697 0.26966292  
## [25] 0.28089888 0.29213483 0.30337079 0.31460674 0.32584270 0.33707865  
## [31] 0.34831461 0.35955056 0.37078652 0.38202247 0.39325843 0.40449438  
## [37] 0.41573034 0.42696629 0.43820225 0.44943820 0.46067416 0.47191011  
## [43] 0.48314607 0.49438202  
##   
## $db  
## [1] 3.16511473 -1.52117846 0.64692449 -0.42240332 -7.43896959  
## [6] 3.73127425 -1.71255287 -3.13335850 -0.74676501 1.41282669  
## [11] -4.74213806 1.20451364 -12.30440175 1.99070209 -0.01414299  
## [16] 0.65654244 -9.72270621 0.89312591 5.69470180 -11.41954841  
## [21] 2.44309035 2.94764754 -4.69994213 -9.08761188 -12.26381132  
## [26] -4.50006621 -4.15229073 0.41484313 -12.00120718 1.31805897  
## [31] 4.85348391 -2.22004364 4.53074670 3.89109318 -11.89694144  
## [36] 1.74051351 -3.25965569 -3.99805437 -5.73697207 -0.33334461  
## [41] -2.11257995 -19.51413473 2.56746569 -1.87102100  
##   
## $dbz  
## [1] 0.04499331 0.02173661 -0.00177432 -0.01443642 -0.01619729  
## [6] -0.01945961 -0.04199865 -0.09648074 -0.18266454 -0.28492406  
## [11] -0.37425178 -0.41296073 -0.36216059 -0.19424906 0.09049337  
## [16] 0.45291842 0.82062094 1.10754632 1.23390852 1.13775273  
## [21] 0.78053106 0.15570159 -0.68676039 -1.58783571 -2.23124357  
## [26] -2.26562309 -1.64757544 -0.68923500 0.27677939 1.06459762  
## [31] 1.59757356 1.84930699 1.81389587 1.49817318 0.92799066  
## [36] 0.16662946 -0.65873078 -1.35000360 -1.71125386 -1.69171536  
## [41] -1.42196858 -1.08681671 -0.81576820 -0.67009418

#Check bic  
aic5.wge(pv\_s7, p=0:13, q=0:3, type="bic") #BIC picks ARMA(1,0) model. Looking at the data, this does not look appropriate.

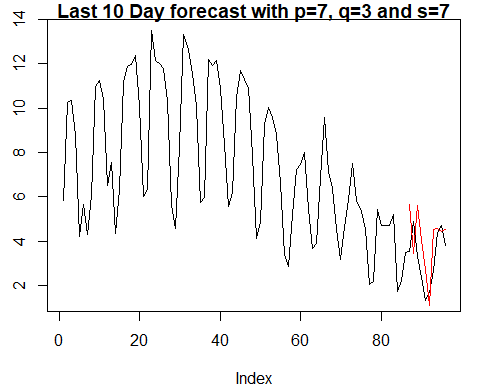
## ---------WORKING... PLEASE WAIT...   
##   
##   
## Error in aic calculation at 1 3   
## Error in aic calculation at 2 3   
## Error in aic calculation at 4 3   
## Error in aic calculation at 10 2   
## Five Smallest Values of bic

## p q bic  
## 5 1 0 0.7485879  
## 6 1 1 0.7902914  
## 9 2 0 0.7929518  
## 3 0 2 0.8172374  
## 2 0 1 0.8214123

#arma\_s7\_d1 <- est.arma.wge(pv\_s7\_d1, p=7, q=3)  
for\_aruma2\_s7 = fore.aruma.wge(Post\_Vaccine$new\_deaths\_per\_million,s=7, phi = arma\_s7\_1$phi,n.ahead = 10, lastn = T)



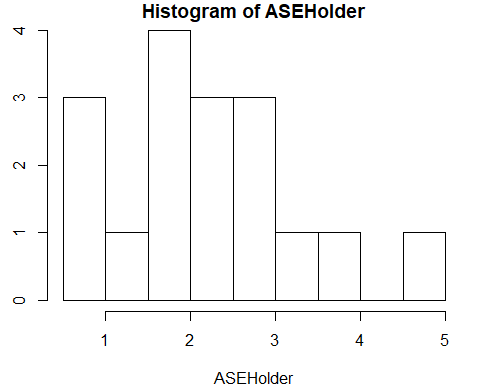
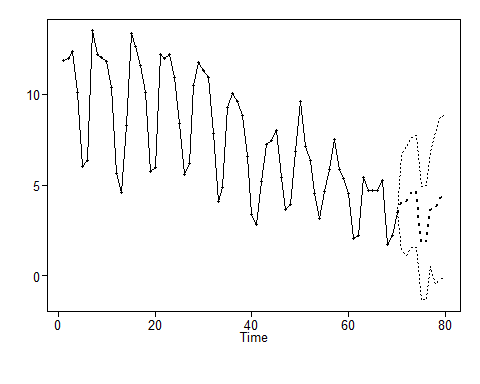
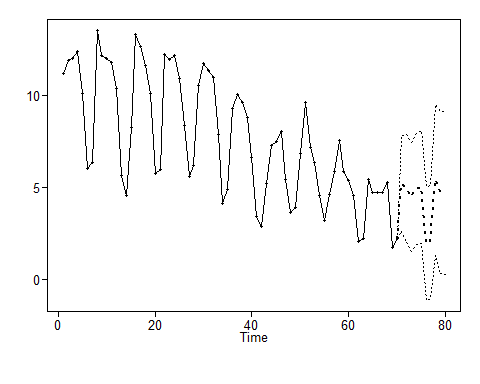
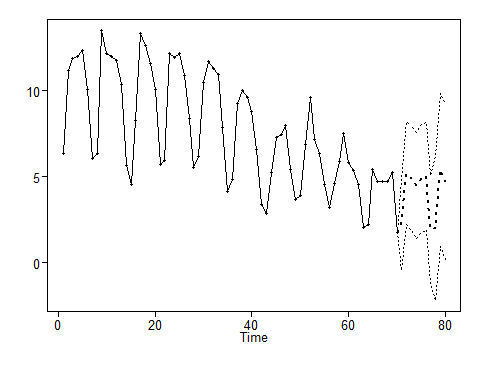
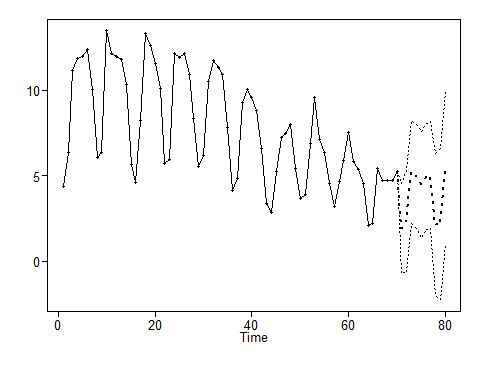
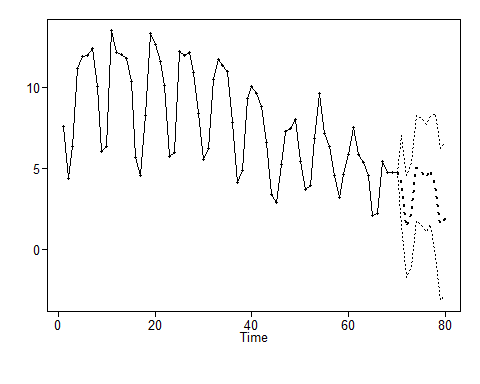
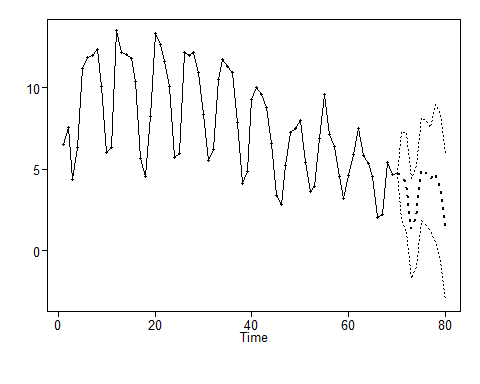
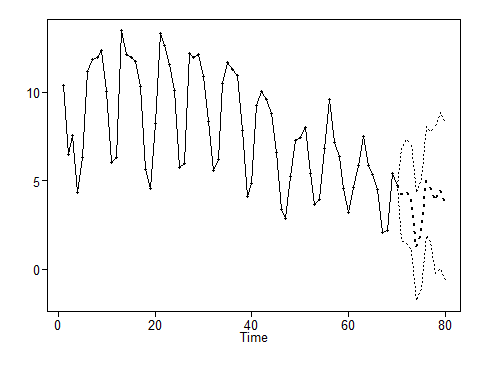
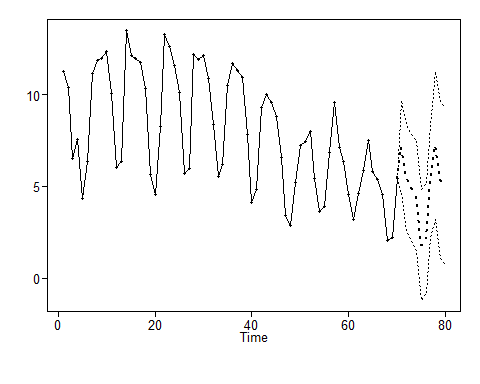
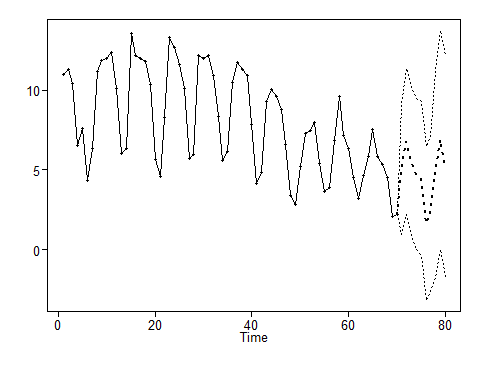
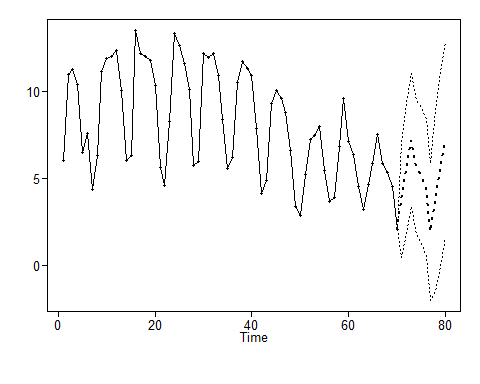
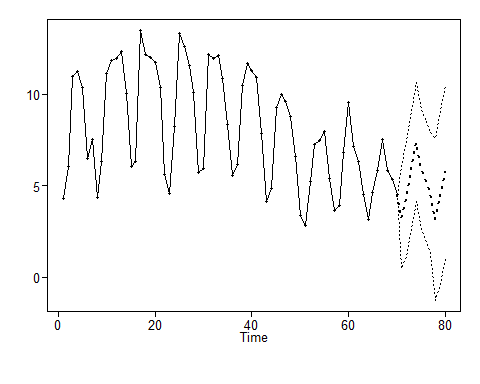
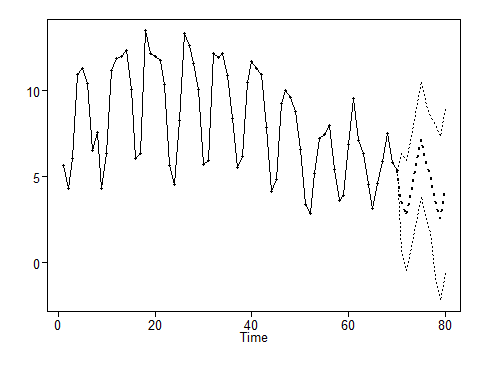
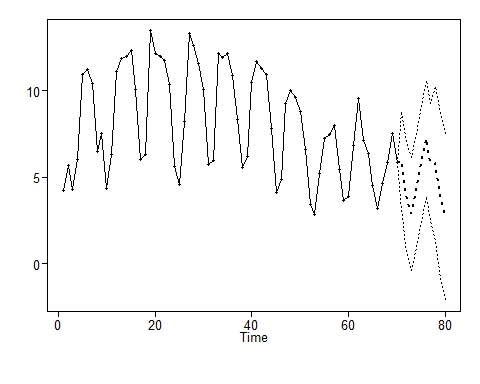
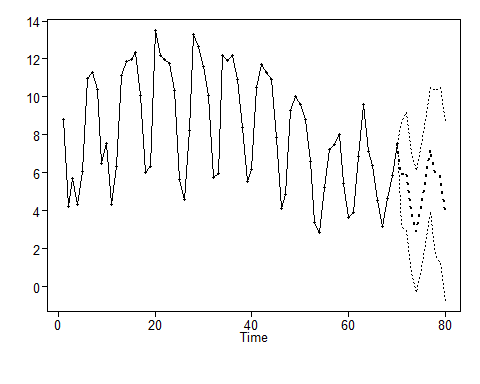
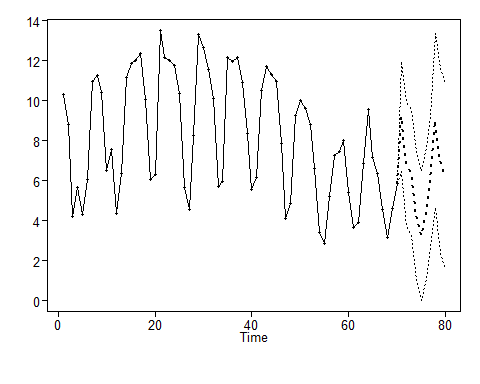
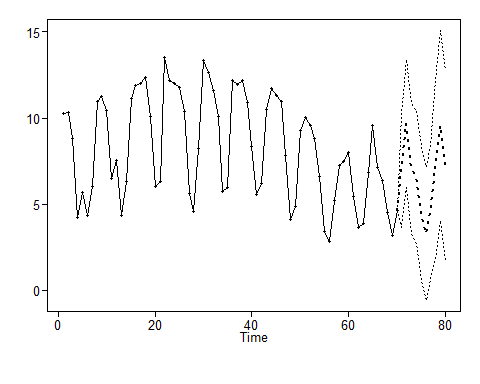
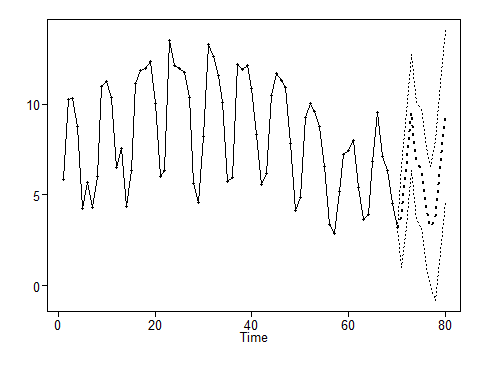
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,96), main = "Last 10 Day forecast with p=7, q=3 and s=7")  
lines(seq(87,96,1),for\_aruma2\_s7$f, col = "red")



#ASE  
arma\_s7\_ase = mean((for\_aruma2\_s7$f - Post\_Vaccine$new\_deaths\_per\_million[(96-10+1):96])^2)  
arma\_s7\_ase

## [1] 2.076175

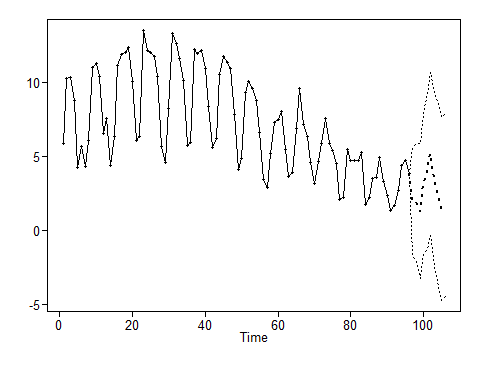
#Rolling\_Window\_ASE  
Rolling\_Window\_ASE(Post\_Vaccine$new\_deaths\_per\_million, trainingSize = 70, horizon = 10, d = 0, phis = arma\_s7\_1$phi,  
 s= 7, thetas = arma\_s7\_1$theta)



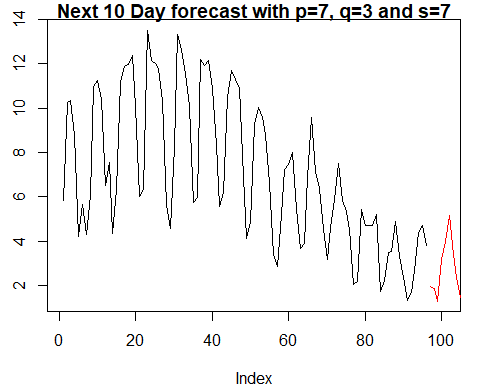
## [1] 10  
## [1] 70  
## [1] "The Summary Statistics for the Rolling Window ASE Are:"  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.9702 1.6788 2.2482 2.2539 2.7086 4.8301   
## [1] "The Rolling Window ASE is: 2.2538629137224"

## [1] 2.253863

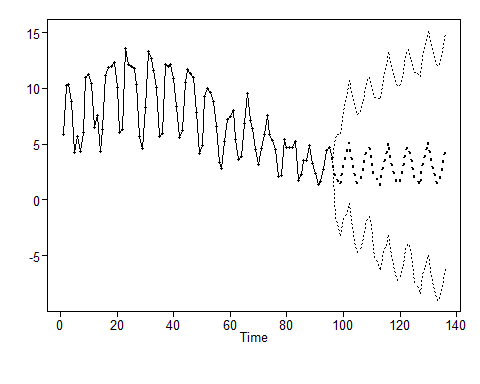
#Forecasting  
#Next 10  
for\_aruma2\_s7\_f10 = fore.aruma.wge(Post\_Vaccine$new\_deaths\_per\_million,s=7, phi = arma\_s7\_1$phi,n.ahead = 10)



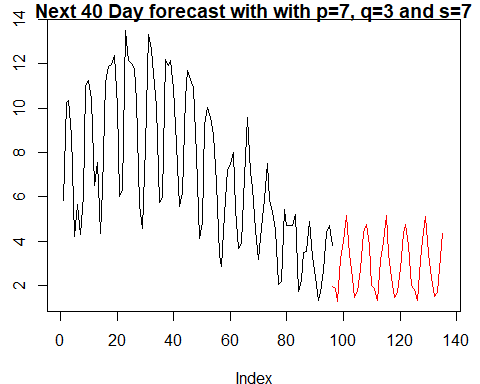
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,101), main = "Next 10 Day forecast with p=7, q=3 and s=7")  
lines(seq(97,106,1),for\_aruma2\_s7\_f10$f, col = "red")



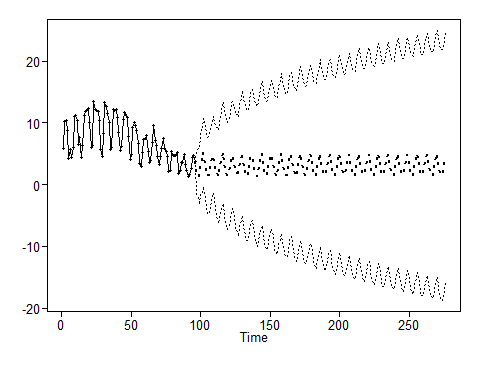
#Next 40  
for\_aruma2\_s7\_f40 = fore.aruma.wge(Post\_Vaccine$new\_deaths\_per\_million,s=7, phi = arma\_s7\_1$phi,n.ahead = 40)



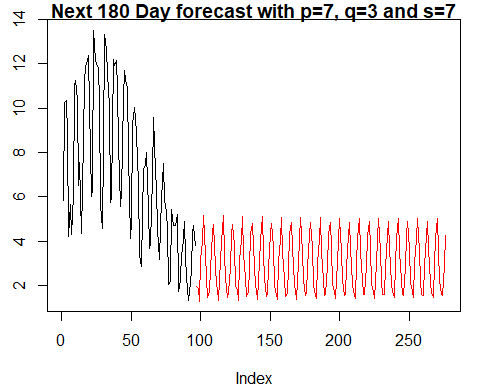
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,136), main = "Next 40 Day forecast with with p=7, q=3 and s=7")  
lines(seq(96,135,1),for\_aruma2\_s7\_f40$f, col = "red")



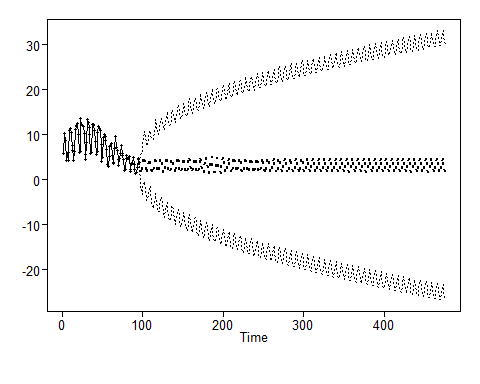
#Next 180  
for\_aruma2\_s7\_f180 = fore.aruma.wge(Post\_Vaccine$new\_deaths\_per\_million,s=7, phi = arma\_s7\_1$phi,n.ahead = 180)



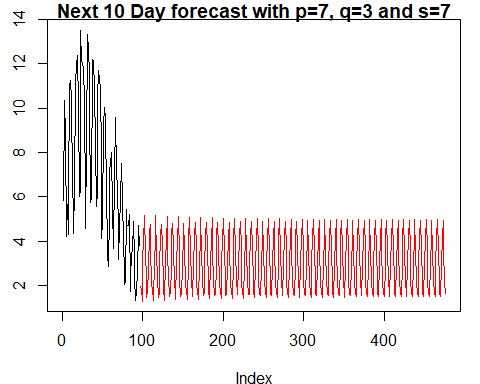
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,276), main = "Next 180 Day forecast with p=7, q=3 and s=7")  
lines(seq(97,276,1),for\_aruma2\_s7\_f180$f, col = "red")



#Next 380  
for\_aruma2\_s7\_f380 = fore.aruma.wge(Post\_Vaccine$new\_deaths\_per\_million,s=7, phi = arma\_s7\_1$phi,n.ahead = 380)



plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,476), main = "Next 10 Day forecast with p=7, q=3 and s=7")  
lines(seq(97,476,1),for\_aruma2\_s7\_f380$f, col = "red")



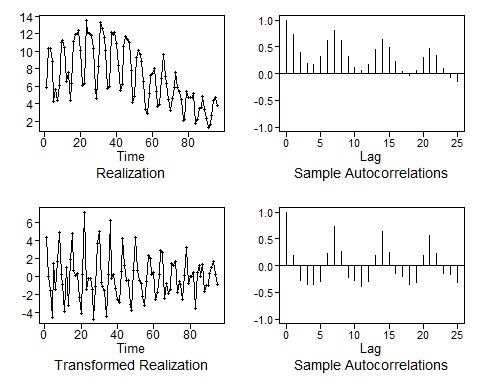
### ARMA MODEL 2 - (with (1-B) component)

# overfitting to check non stationary portion of the data  
est.ar.wge(Post\_Vaccine$new\_deaths\_per\_million, p=8, type='burg') #has one 1-0.9868B

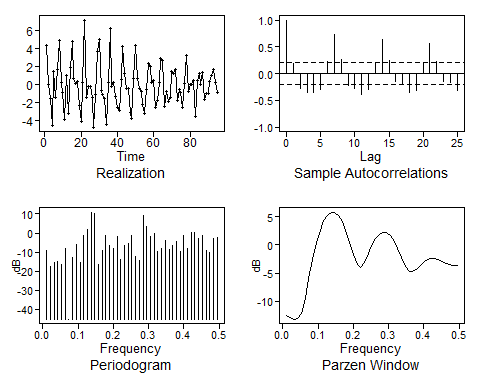
##   
## Coefficients of Original polynomial:   
## 0.6105 -0.1117 0.0222 -0.0007 -0.0816 0.2254 0.5762 -0.2837   
##   
## Factor Roots Abs Recip System Freq   
## 1-0.9868B 1.0133 0.9868 0.0000  
## 1-1.2341B+0.9673B^2 0.6379+-0.7917i 0.9835 0.1421  
## 1+0.4502B+0.9015B^2 -0.2497+-1.0232i 0.9495 0.2881  
## 1+1.5912B+0.7652B^2 -1.0398+-0.4752i 0.8747 0.4318  
## 1-0.4309B 2.3205 0.4309 0.0000  
##   
##

## $phi  
## [1] 0.6105124076 -0.1117396292 0.0222221729 -0.0007395317 -0.0815648629  
## [6] 0.2254343105 0.5762355978 -0.2837410524  
##   
## $res  
## [1] -0.395967551 1.412382837 -0.262528353 0.642458488 -1.769929618  
## [6] 1.486845251 -0.920722449 0.133877695 1.467078233 0.168383118  
## [11] 1.692432954 -0.009618396 1.652627698 -1.415288592 0.268812894  
## [16] 1.004441629 0.174783230 1.882634588 3.676864986 0.322574139  
## [21] -0.174359237 -0.768964700 3.488453151 -1.190383726 0.139261616  
## [26] -0.298916532 0.802287438 -0.876023197 -2.264964790 -2.201857114  
## [31] 3.440926848 -0.247540763 -0.387604666 0.313236396 -0.393871247  
## [36] 1.219944926 3.233965987 -2.484785325 0.501737648 -0.657136371  
## [41] -0.832051410 0.320646089 -0.397447960 -0.171959374 0.293088150  
## [46] -0.505689198 0.545988447 -0.606554288 -1.365936926 -0.726750912  
## [51] 0.294926337 -0.921699580 -0.920083197 -0.976694497 -0.178883129  
## [56] -0.645250289 -1.874015679 -2.045693216 -0.580525947 -0.873063802  
## [61] 0.265574157 -0.918324125 0.459088551 0.540648571 1.228638870  
## [66] 1.716508445 -1.939398452 -0.796768668 -0.422414205 -0.226119625  
## [71] 0.766816503 -1.418711372 -0.583548431 -0.971105108 -0.198890363  
## [76] 0.283228350 -1.647584629 -1.442203539 0.613180527 -2.257846760  
## [81] 0.028389017 -0.345533947 1.090004103 -1.289275466 -0.170318214  
## [86] -1.261766943 -0.533543898 0.729455680 -2.082129334 -1.393772694  
## [91] 0.123863314 -0.542962997 -0.343373755 0.726750325 -0.379203270  
## [96] 0.134596076  
##   
## $avar  
## [1] 1.538098  
##   
## $aic  
## [1] 0.6180465  
##   
## $aicc  
## [1] 1.665841  
##   
## $bic  
## [1] 0.8584541

# 1st difference  
pv\_dif1 <- artrans.wge(Post\_Vaccine$new\_deaths\_per\_million, phi.tr=1)#dif data does not look like white noise.



plotts.sample.wge(pv\_dif1, arlimits = T)

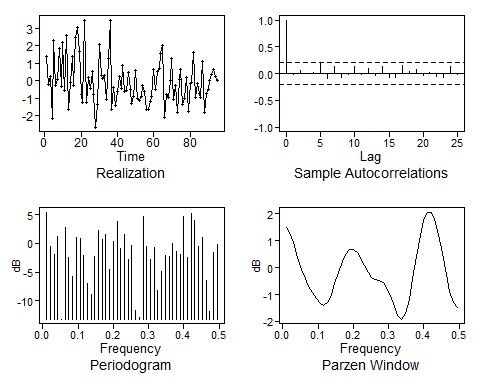


## $autplt  
## [1] 1.0000000 0.1829721 -0.2849749 -0.3608558 -0.3599959 -0.3005161  
## [7] 0.2334081 0.7310677 0.2649303 -0.2164156 -0.2681506 -0.3857824  
## [13] -0.2979713 0.1962294 0.6450992 0.2456240 -0.1543115 -0.2107155  
## [19] -0.3587187 -0.3068705 0.1814334 0.5695782 0.2365243 -0.1421584  
## [25] -0.1699776 -0.3195886  
##   
## $freq  
## [1] 0.01052632 0.02105263 0.03157895 0.04210526 0.05263158 0.06315789  
## [7] 0.07368421 0.08421053 0.09473684 0.10526316 0.11578947 0.12631579  
## [13] 0.13684211 0.14736842 0.15789474 0.16842105 0.17894737 0.18947368  
## [19] 0.20000000 0.21052632 0.22105263 0.23157895 0.24210526 0.25263158  
## [25] 0.26315789 0.27368421 0.28421053 0.29473684 0.30526316 0.31578947  
## [31] 0.32631579 0.33684211 0.34736842 0.35789474 0.36842105 0.37894737  
## [37] 0.38947368 0.40000000 0.41052632 0.42105263 0.43157895 0.44210526  
## [43] 0.45263158 0.46315789 0.47368421 0.48421053 0.49473684  
##   
## $db  
## [1] -9.1447854 -17.5152409 -15.2217035 -14.9483634 -16.3616096  
## [6] -7.9817964 -45.0710116 -13.0450700 -6.1762515 -15.6690427  
## [11] -1.3854730 1.6535434 10.8920771 10.2193492 -16.4890836  
## [16] -9.4007269 -1.2487324 -6.7457546 -8.3066540 -1.7381442  
## [21] -13.7238194 -6.7195505 -5.6034071 -1.4702385 -12.2003715  
## [26] -14.1900449 8.9418766 3.1645957 -1.9220632 -0.4066820  
## [31] -9.5157392 -7.9162448 -4.0282890 -8.4591908 -6.3743177  
## [36] -4.3450179 -9.4047613 -1.5259365 -8.2762607 0.4892745  
## [41] 0.4036087 -2.9976560 -1.4914511 -8.9984662 -10.3793638  
## [46] -2.9386257 -2.5320579  
##   
## $dbz  
## [1] -12.4451922 -12.8227637 -13.1370990 -13.0916798 -12.1167990  
## [6] -9.5301209 -5.9587015 -2.5061310 0.3773855 2.6192123  
## [11] 4.2460509 5.2962917 5.7991779 5.7716621 5.2202531  
## [16] 4.1464253 2.5608484 0.5253171 -1.7255003 -3.5449330  
## [21] -4.0139922 -3.0898128 -1.5882803 -0.1054644 1.0983031  
## [26] 1.9119495 2.2865782 2.2034478 1.6620878 0.6820996  
## [31] -0.6762250 -2.2593908 -3.7425763 -4.6649628 -4.7856215  
## [36] -4.3306474 -3.6751986 -3.0656378 -2.6239360 -2.4039716  
## [41] -2.4140284 -2.6181534 -2.9358211 -3.2582865 -3.4930791  
## [46] -3.6120759 -3.6501624

dif1\_aic <- aic.wge(pv\_dif1) #p=5, q=2  
arima\_1\_diff1 <- est.arma.wge(pv\_dif1, p=5, q=2)

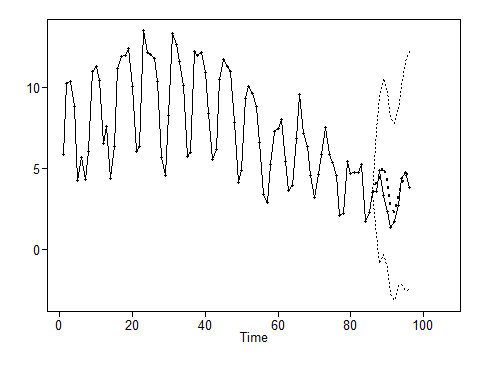
##   
## Coefficients of Original polynomial:   
## 0.2025 -0.6998 -0.2596 -0.3568 -0.5102   
##   
## Factor Roots Abs Recip System Freq   
## 1-1.2498B+0.9790B^2 0.6383+-0.7836i 0.9895 0.1412  
## 1+0.3766B+0.7771B^2 -0.2423+-1.1082i 0.8815 0.2843  
## 1+0.6706B -1.4911 0.6706 0.5000  
##   
##

#Examine model residuals  
plotts.sample.wge(arima\_1\_diff1$res, arlimits = T) #Not white noise, model needs improvement

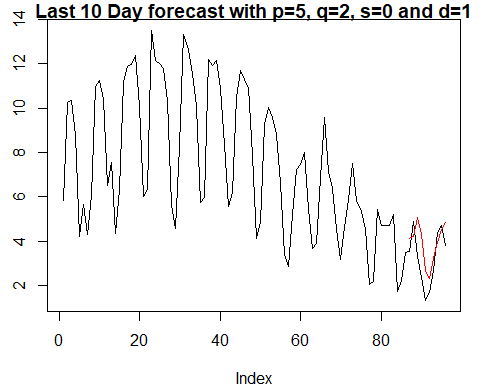


## $autplt  
## [1] 1.000000000 -0.016366806 0.054503248 0.002332935 0.025854277  
## [6] 0.211628198 -0.084437349 0.128636457 -0.074199295 -0.016021224  
## [11] 0.142565771 -0.023295876 0.095217566 -0.037238769 0.135927595  
## [16] -0.077909711 -0.069864550 0.144421426 0.049922851 0.077595812  
## [21] -0.025247490 0.021675670 -0.056095901 -0.078544644 0.131544968  
## [26] -0.006282457  
##   
## $freq  
## [1] 0.01052632 0.02105263 0.03157895 0.04210526 0.05263158 0.06315789  
## [7] 0.07368421 0.08421053 0.09473684 0.10526316 0.11578947 0.12631579  
## [13] 0.13684211 0.14736842 0.15789474 0.16842105 0.17894737 0.18947368  
## [19] 0.20000000 0.21052632 0.22105263 0.23157895 0.24210526 0.25263158  
## [25] 0.26315789 0.27368421 0.28421053 0.29473684 0.30526316 0.31578947  
## [31] 0.32631579 0.33684211 0.34736842 0.35789474 0.36842105 0.37894737  
## [37] 0.38947368 0.40000000 0.41052632 0.42105263 0.43157895 0.44210526  
## [43] 0.45263158 0.46315789 0.47368421 0.48421053 0.49473684  
##   
## $db  
## [1] 5.40611344 -0.49803418 -1.99280291 1.24806517 -13.16482918  
## [6] 2.73873122 -2.50766171 -5.75026381 0.98753922 0.76260924  
## [11] -2.03933164 -7.05586782 -8.80392032 -2.22635319 2.15348220  
## [16] 0.71192840 1.61447550 -4.50276392 0.39830853 3.70851584  
## [21] -0.85886176 1.51362400 -2.79414454 -0.45141704 -11.65743526  
## [26] -12.90335034 4.70365429 -0.53994101 -2.87455108 -0.76125682  
## [31] -8.16728522 -4.95700989 -2.14838280 -2.28333603 -0.09824085  
## [36] -1.36737049 -1.90344899 4.60318677 -2.37414887 5.11286181  
## [41] 4.02378696 -0.52134385 1.05860385 -6.49212578 -11.77147755  
## [46] -1.60313804 -0.15740303  
##   
## $dbz  
## [1] 1.49396923 1.22061557 0.82056493 0.36867844 -0.05692298  
## [6] -0.40760236 -0.68388135 -0.91896019 -1.13685527 -1.31783882  
## [11] -1.39651923 -1.30237900 -1.01979945 -0.60932980 -0.16881942  
## [16] 0.21709169 0.49768339 0.65109649 0.67265326 0.56889674  
## [21] 0.36085787 0.09215278 -0.17062685 -0.36057095 -0.45304991  
## [26] -0.49067354 -0.55886590 -0.73353367 -1.03926746 -1.42785331  
## [31] -1.77527779 -1.91350576 -1.71381094 -1.17196461 -0.40539628  
## [36] 0.42356923 1.17080225 1.73104415 2.03652701 2.05185713  
## [41] 1.77295286 1.23311007 0.51508516 -0.24360069 -0.88215138  
## [46] -1.29468920 -1.47906613

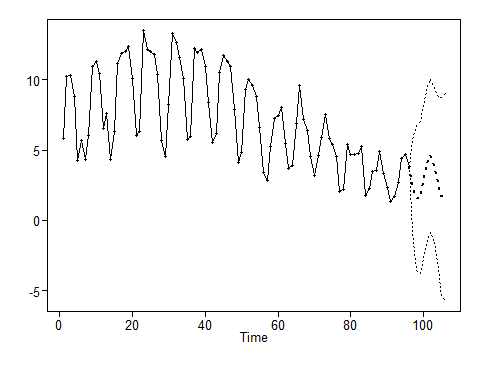
#forecast using diff-1  
#Last 10  
for\_aruma1\_diff1 = fore.aruma.wge(Post\_Vaccine$new\_deaths\_per\_million,d=1, phi = arima\_1\_diff1$phi,n.ahead = 10, lastn = T)



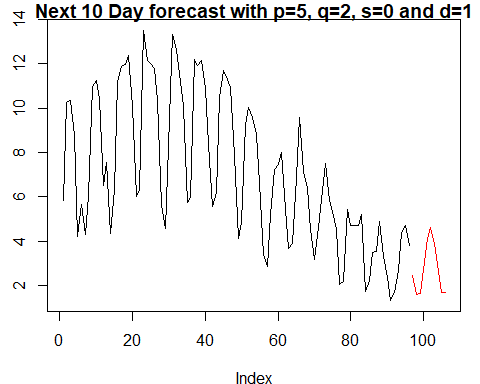
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,96), main = "Last 10 Day forecast with p=5, q=2, s=0 and d=1")  
lines(seq(87,96,1),for\_aruma1\_diff1$f, col = "red")



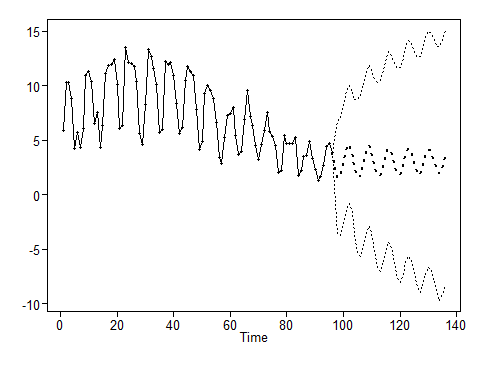
#Forecast future 10  
for\_aruma1\_diff\_f10 = fore.aruma.wge(Post\_Vaccine$new\_deaths\_per\_million,d=1, phi = arima\_1\_diff1$phi,n.ahead = 10)



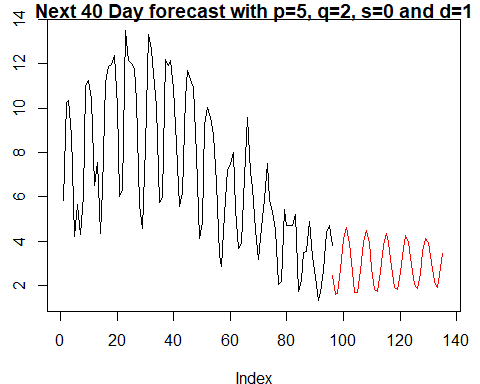
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,106), main = "Next 10 Day forecast with p=5, q=2, s=0 and d=1")  
lines(seq(97,106,1),for\_aruma1\_diff\_f10$f, col = "red")



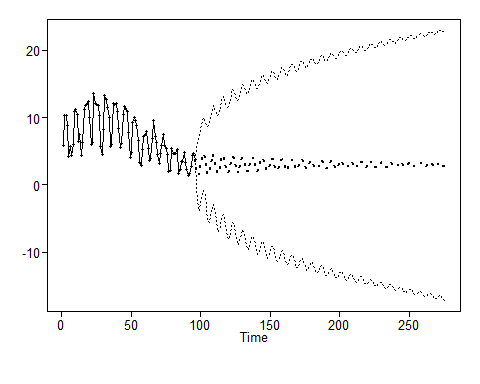
#Forecast future 40  
for\_aruma1\_diff\_f40 = fore.aruma.wge(Post\_Vaccine$new\_deaths\_per\_million,d=1, phi = arima\_1\_diff1$phi,n.ahead = 40)



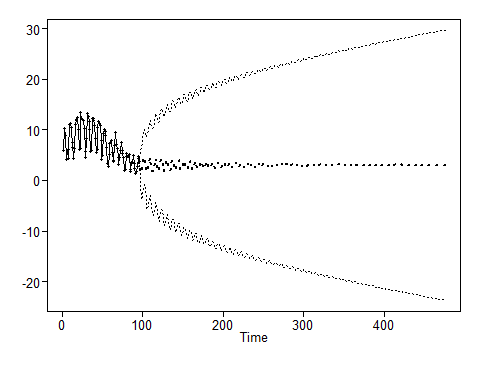
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,136), main = "Next 40 Day forecast with p=5, q=2, s=0 and d=1")  
lines(seq(96,135,1),for\_aruma1\_diff\_f40$f, col = "red")



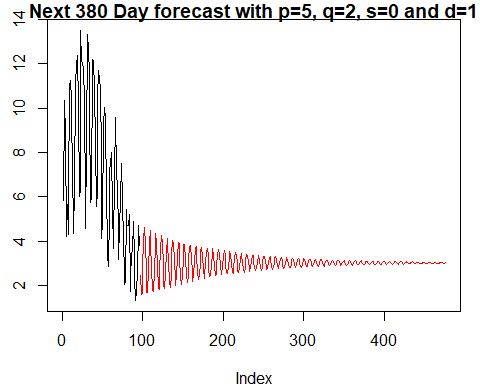
#Forecast future 180  
for\_aruma1\_diff\_f180 = fore.aruma.wge(Post\_Vaccine$new\_deaths\_per\_million,d=1, phi = arima\_1\_diff1$phi,n.ahead = 180)



#Forecast future 380  
for\_aruma1\_diff\_f380 = fore.aruma.wge(Post\_Vaccine$new\_deaths\_per\_million,d=1, phi = arima\_1\_diff1$phi,n.ahead = 380)



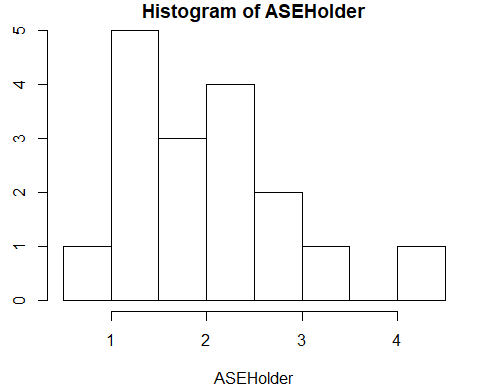
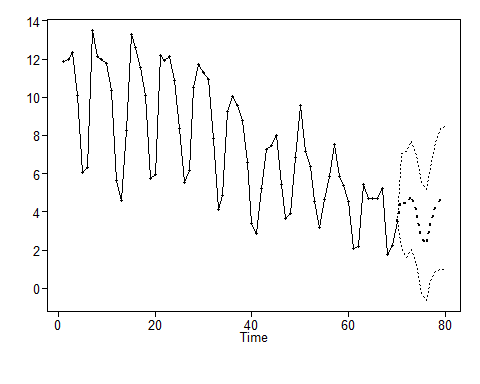
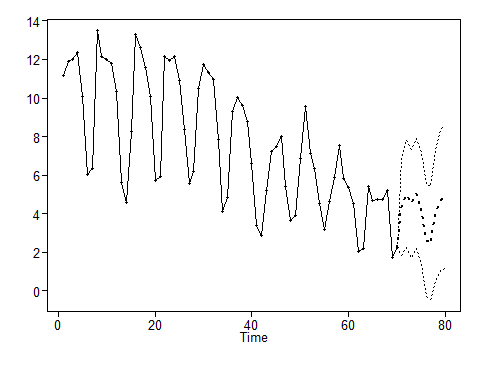
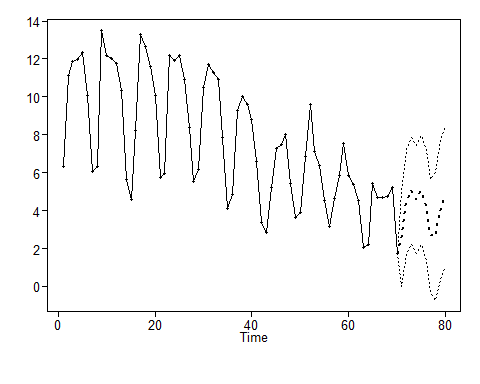
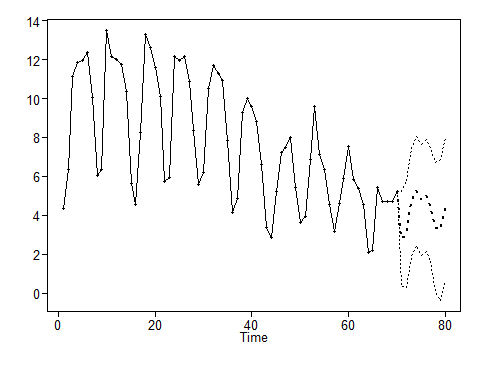
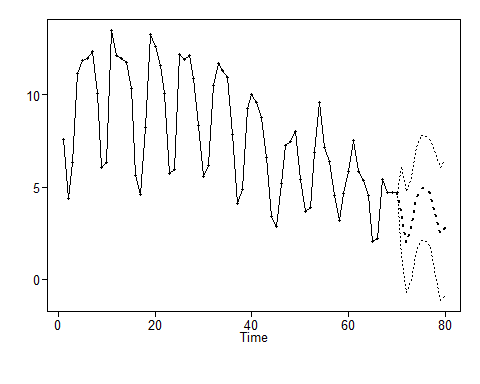
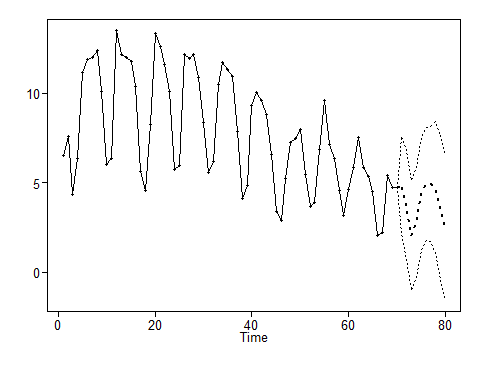
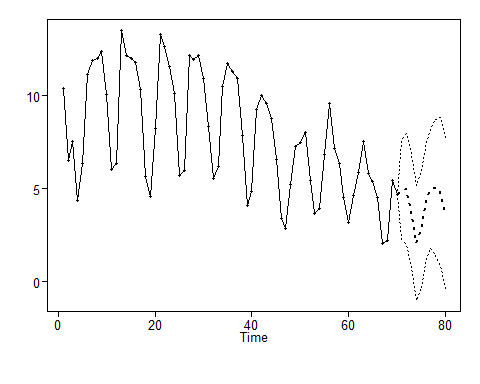
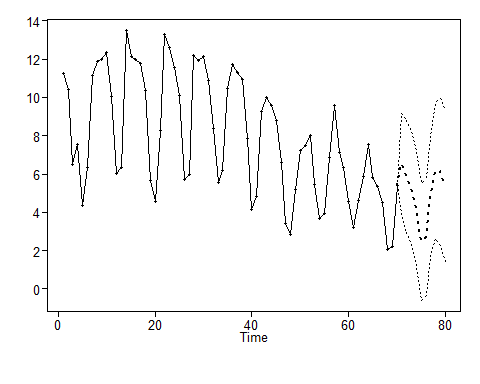
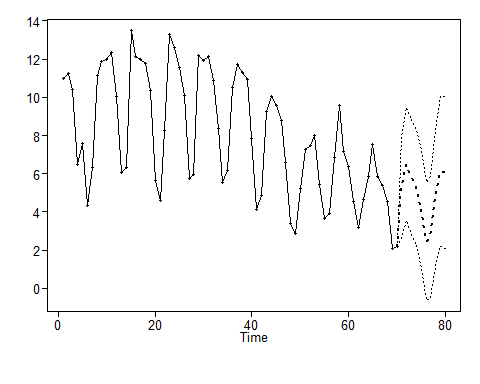
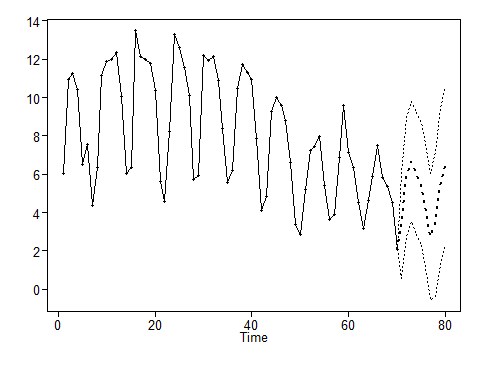
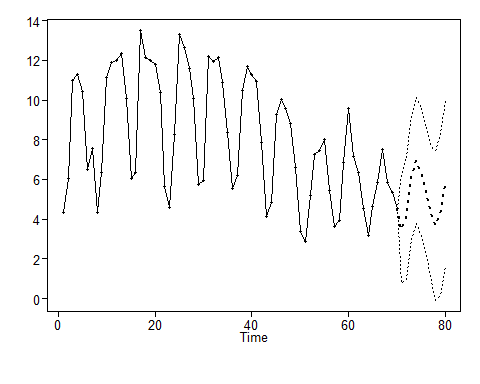
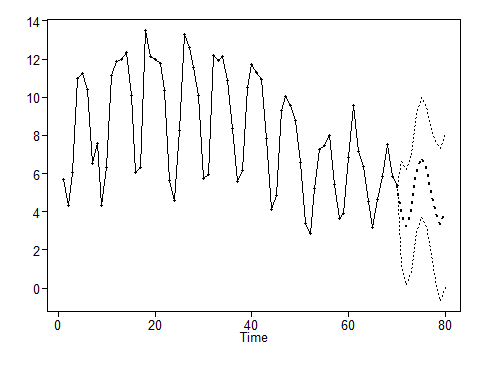
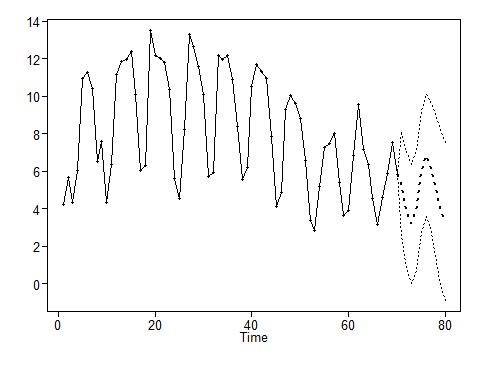
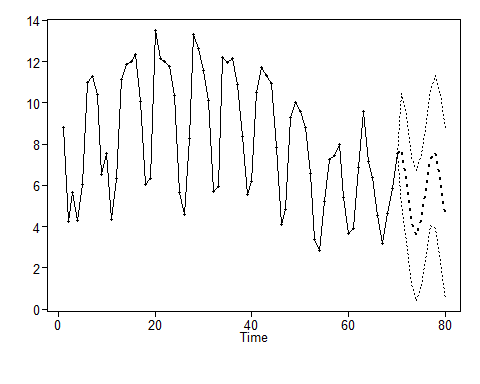
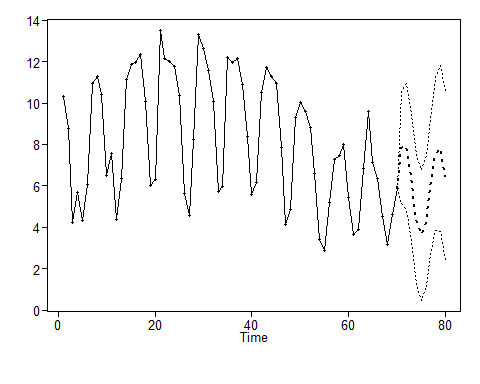
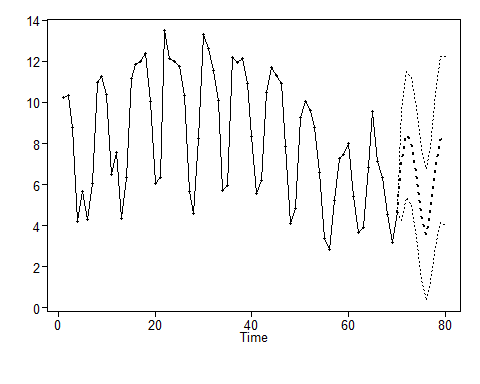
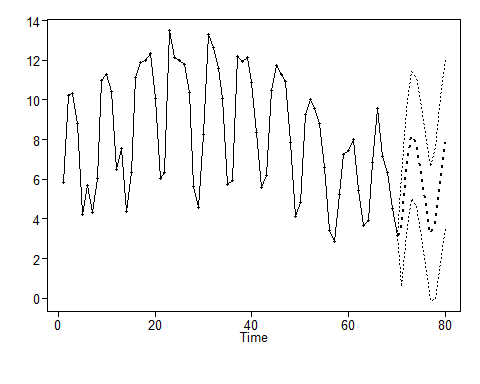
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,476), main = "Next 380 Day forecast with p=5, q=2, s=0 and d=1")  
lines(seq(97,476,1),for\_aruma1\_diff\_f380$f, col = "red")



#ASE  
arima1\_diff1\_ase = mean((for\_aruma1\_diff1$f - Post\_Vaccine$new\_deaths\_per\_million[(96-10+1):96])^2)  
arima1\_diff1\_ase

## [1] 1.159396

Rolling\_Window\_ASE(Post\_Vaccine$new\_deaths\_per\_million, trainingSize = 70, horizon = 10, d = 1, phis = arima\_1\_diff1$phi,  
 s= 0, thetas = arima\_1\_diff1$theta)

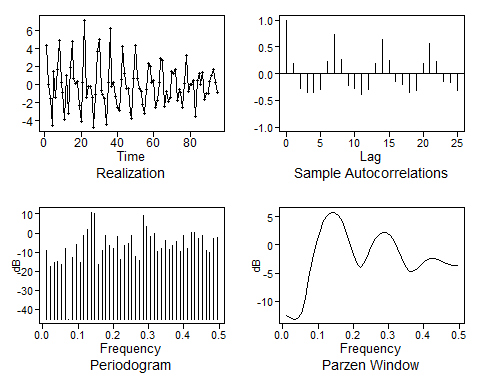


## [1] 10  
## [1] 70  
## [1] "The Summary Statistics for the Rolling Window ASE Are:"  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.9537 1.3391 1.9846 2.0191 2.3092 4.2075   
## [1] "The Rolling Window ASE is: 2.01912770779326"

## [1] 2.019128

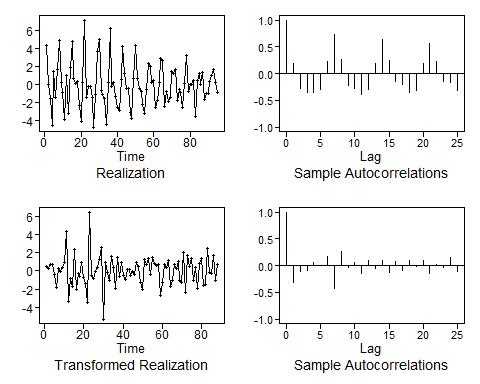
### ARIMA - (with both seasonal(s = 7) and trend(1-B) components)

#Seasonality plus 1-B  
plotts.sample.wge(pv\_dif1)

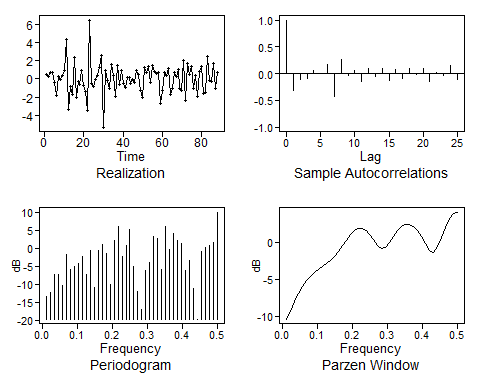


## $autplt  
## [1] 1.0000000 0.1829721 -0.2849749 -0.3608558 -0.3599959 -0.3005161  
## [7] 0.2334081 0.7310677 0.2649303 -0.2164156 -0.2681506 -0.3857824  
## [13] -0.2979713 0.1962294 0.6450992 0.2456240 -0.1543115 -0.2107155  
## [19] -0.3587187 -0.3068705 0.1814334 0.5695782 0.2365243 -0.1421584  
## [25] -0.1699776 -0.3195886  
##   
## $freq  
## [1] 0.01052632 0.02105263 0.03157895 0.04210526 0.05263158 0.06315789  
## [7] 0.07368421 0.08421053 0.09473684 0.10526316 0.11578947 0.12631579  
## [13] 0.13684211 0.14736842 0.15789474 0.16842105 0.17894737 0.18947368  
## [19] 0.20000000 0.21052632 0.22105263 0.23157895 0.24210526 0.25263158  
## [25] 0.26315789 0.27368421 0.28421053 0.29473684 0.30526316 0.31578947  
## [31] 0.32631579 0.33684211 0.34736842 0.35789474 0.36842105 0.37894737  
## [37] 0.38947368 0.40000000 0.41052632 0.42105263 0.43157895 0.44210526  
## [43] 0.45263158 0.46315789 0.47368421 0.48421053 0.49473684  
##   
## $db  
## [1] -9.1447854 -17.5152409 -15.2217035 -14.9483634 -16.3616096  
## [6] -7.9817964 -45.0710116 -13.0450700 -6.1762515 -15.6690427  
## [11] -1.3854730 1.6535434 10.8920771 10.2193492 -16.4890836  
## [16] -9.4007269 -1.2487324 -6.7457546 -8.3066540 -1.7381442  
## [21] -13.7238194 -6.7195505 -5.6034071 -1.4702385 -12.2003715  
## [26] -14.1900449 8.9418766 3.1645957 -1.9220632 -0.4066820  
## [31] -9.5157392 -7.9162448 -4.0282890 -8.4591908 -6.3743177  
## [36] -4.3450179 -9.4047613 -1.5259365 -8.2762607 0.4892745  
## [41] 0.4036087 -2.9976560 -1.4914511 -8.9984662 -10.3793638  
## [46] -2.9386257 -2.5320579  
##   
## $dbz  
## [1] -12.4451922 -12.8227637 -13.1370990 -13.0916798 -12.1167990  
## [6] -9.5301209 -5.9587015 -2.5061310 0.3773855 2.6192123  
## [11] 4.2460509 5.2962917 5.7991779 5.7716621 5.2202531  
## [16] 4.1464253 2.5608484 0.5253171 -1.7255003 -3.5449330  
## [21] -4.0139922 -3.0898128 -1.5882803 -0.1054644 1.0983031  
## [26] 1.9119495 2.2865782 2.2034478 1.6620878 0.6820996  
## [31] -0.6762250 -2.2593908 -3.7425763 -4.6649628 -4.7856215  
## [36] -4.3306474 -3.6751986 -3.0656378 -2.6239360 -2.4039716  
## [41] -2.4140284 -2.6181534 -2.9358211 -3.2582865 -3.4930791  
## [46] -3.6120759 -3.6501624

pv\_s7\_d1 <- artrans.wge(pv\_dif1,phi.tr = c(rep(0,6),1)) #Looks statinary



plotts.sample.wge(pv\_s7\_d1)



## $autplt  
## [1] 1.000000000 -0.310489297 -0.108797691 -0.097201773 0.058890926  
## [6] -0.003543395 0.178997215 -0.424863312 0.261685568 -0.041392776  
## [11] 0.059461184 -0.151476480 0.105238964 -0.059635158 0.103862480  
## [16] -0.119913933 0.082902182 -0.098411931 0.104543127 -0.023775132  
## [21] 0.099846159 -0.146811040 0.014406847 -0.039855995 0.158195165  
## [26] -0.114622681  
##   
## $freq  
## [1] 0.01136364 0.02272727 0.03409091 0.04545455 0.05681818 0.06818182  
## [7] 0.07954545 0.09090909 0.10227273 0.11363636 0.12500000 0.13636364  
## [13] 0.14772727 0.15909091 0.17045455 0.18181818 0.19318182 0.20454545  
## [19] 0.21590909 0.22727273 0.23863636 0.25000000 0.26136364 0.27272727  
## [25] 0.28409091 0.29545455 0.30681818 0.31818182 0.32954545 0.34090909  
## [31] 0.35227273 0.36363636 0.37500000 0.38636364 0.39772727 0.40909091  
## [37] 0.42045455 0.43181818 0.44318182 0.45454545 0.46590909 0.47727273  
## [43] 0.48863636 0.50000000  
##   
## $db  
## [1] -13.3604568 -12.2220301 -7.4012596 -7.2847222 -10.4496042  
## [6] -1.7456939 -5.9337457 -5.0909825 -4.1490614 -2.2346615  
## [11] -7.2088406 -0.6040349 -11.0293421 -0.5624124 0.9898893  
## [16] -1.3787451 -10.1832260 2.1951342 5.9785213 -2.2660619  
## [21] 0.9335691 5.1498756 -5.1276850 -12.0322733 -17.0711550  
## [26] -6.2845356 -3.8616592 3.3654951 2.7925788 -6.0014150  
## [31] 6.1073378 -0.3369475 4.2486121 2.2993108 1.3249688  
## [36] -6.2279128 -3.4789517 -11.0473726 -19.7086141 -0.9642477  
## [41] 0.1445992 0.7589488 1.5658848 10.0682368  
##   
## $dbz  
## [1] -10.40935994 -9.32761767 -8.07608475 -6.92408967 -5.96208710  
## [6] -5.19509494 -4.59385150 -4.11545638 -3.71217855 -3.33591517  
## [11] -2.93984038 -2.47939218 -1.91866082 -1.24620501 -0.49130192  
## [16] 0.27659188 0.96805325 1.49769540 1.79833578 1.82548337  
## [21] 1.56004216 1.01881612 0.28320967 -0.45299537 -0.87735351  
## [26] -0.73664301 -0.09406032 0.75022898 1.52951710 2.10029961  
## [31] 2.40321242 2.41845112 2.14261281 1.58209921 0.76411634  
## [36] -0.21638297 -1.08870944 -1.34844939 -0.65772975 0.65740598  
## [41] 2.02938041 3.12138550 3.80286978 4.03293524

#Overfitting  
est.ar.wge(Post\_Vaccine$new\_deaths\_per\_million, p=8, type='burg')

##   
## Coefficients of Original polynomial:   
## 0.6105 -0.1117 0.0222 -0.0007 -0.0816 0.2254 0.5762 -0.2837   
##   
## Factor Roots Abs Recip System Freq   
## 1-0.9868B 1.0133 0.9868 0.0000  
## 1-1.2341B+0.9673B^2 0.6379+-0.7917i 0.9835 0.1421  
## 1+0.4502B+0.9015B^2 -0.2497+-1.0232i 0.9495 0.2881  
## 1+1.5912B+0.7652B^2 -1.0398+-0.4752i 0.8747 0.4318  
## 1-0.4309B 2.3205 0.4309 0.0000  
##   
##

## $phi  
## [1] 0.6105124076 -0.1117396292 0.0222221729 -0.0007395317 -0.0815648629  
## [6] 0.2254343105 0.5762355978 -0.2837410524  
##   
## $res  
## [1] -0.395967551 1.412382837 -0.262528353 0.642458488 -1.769929618  
## [6] 1.486845251 -0.920722449 0.133877695 1.467078233 0.168383118  
## [11] 1.692432954 -0.009618396 1.652627698 -1.415288592 0.268812894  
## [16] 1.004441629 0.174783230 1.882634588 3.676864986 0.322574139  
## [21] -0.174359237 -0.768964700 3.488453151 -1.190383726 0.139261616  
## [26] -0.298916532 0.802287438 -0.876023197 -2.264964790 -2.201857114  
## [31] 3.440926848 -0.247540763 -0.387604666 0.313236396 -0.393871247  
## [36] 1.219944926 3.233965987 -2.484785325 0.501737648 -0.657136371  
## [41] -0.832051410 0.320646089 -0.397447960 -0.171959374 0.293088150  
## [46] -0.505689198 0.545988447 -0.606554288 -1.365936926 -0.726750912  
## [51] 0.294926337 -0.921699580 -0.920083197 -0.976694497 -0.178883129  
## [56] -0.645250289 -1.874015679 -2.045693216 -0.580525947 -0.873063802  
## [61] 0.265574157 -0.918324125 0.459088551 0.540648571 1.228638870  
## [66] 1.716508445 -1.939398452 -0.796768668 -0.422414205 -0.226119625  
## [71] 0.766816503 -1.418711372 -0.583548431 -0.971105108 -0.198890363  
## [76] 0.283228350 -1.647584629 -1.442203539 0.613180527 -2.257846760  
## [81] 0.028389017 -0.345533947 1.090004103 -1.289275466 -0.170318214  
## [86] -1.261766943 -0.533543898 0.729455680 -2.082129334 -1.393772694  
## [91] 0.123863314 -0.542962997 -0.343373755 0.726750325 -0.379203270  
## [96] 0.134596076  
##   
## $avar  
## [1] 1.538098  
##   
## $aic  
## [1] 0.6180465  
##   
## $aicc  
## [1] 1.665841  
##   
## $bic  
## [1] 0.8584541

#aic with defaults  
aic.wge(pv\_s7\_d1) #=p=0 q=2

## $type  
## [1] "aic"  
##   
## $value  
## [1] 0.7419945  
##   
## $p  
## [1] 0  
##   
## $q  
## [1] 2  
##   
## $phi  
## [1] 0  
##   
## $theta  
## [1] 0.5978693 0.2840983  
##   
## $vara  
## [1] 1.961702

#Check bic  
aic5.wge(pv\_s7\_d1, p=0:13, q=0:3, type="bic") #BIC picks MA(2) model. Looking at the data, this does not look appropriate.

## ---------WORKING... PLEASE WAIT...   
##   
##   
## Error in aic calculation at 3 2   
## Error in aic calculation at 3 3   
## Five Smallest Values of bic

## p q bic  
## 3 0 2 0.8264491  
## 2 0 1 0.8410476  
## 4 0 3 0.8789084  
## 6 1 1 0.8793909  
## 29 7 0 0.9121151

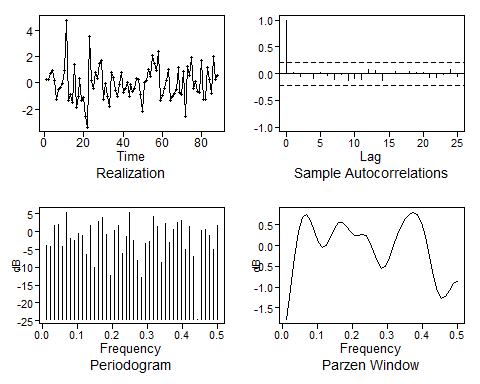
#aic with bigger p and q range  
aic.wge(pv\_s7\_d1, p=0:13, q=0:3) #=p=7 q=0

## $type  
## [1] "aic"  
##   
## $value  
## [1] 0.6869027  
##   
## $p  
## [1] 7  
##   
## $q  
## [1] 0  
##   
## $phi  
## [1] -0.40091849 -0.37704006 -0.32101509 -0.22790505 -0.17732419 -0.03340181  
## [7] -0.38601519  
##   
## $theta  
## [1] 0  
##   
## $vara  
## [1] 1.657126

arima\_s7\_d1 <- est.ar.wge(pv\_s7\_d1, p=7, type='burg')

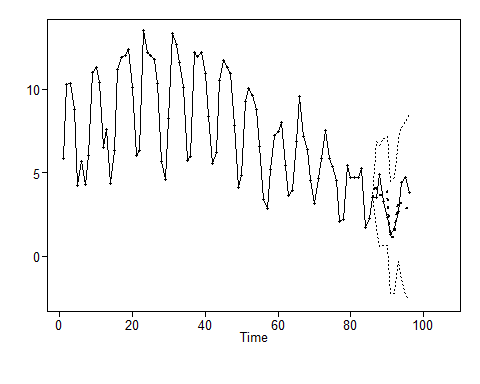
##   
## Coefficients of Original polynomial:   
## -0.4043 -0.3805 -0.3330 -0.2273 -0.1792 -0.0318 -0.3981   
##   
## Factor Roots Abs Recip System Freq   
## 1+0.9256B -1.0804 0.9256 0.5000  
## 1+1.1234B+0.8137B^2 -0.6903+-0.8674i 0.9021 0.3570  
## 1-0.3342B+0.7895B^2 0.2116+-1.1054i 0.8885 0.2199  
## 1-1.3105B+0.6694B^2 0.9788+-0.7319i 0.8182 0.1022  
##   
##

#residuals  
plotts.sample.wge(arima\_s7\_d1$res, arlimits = T) #Much better model residuals

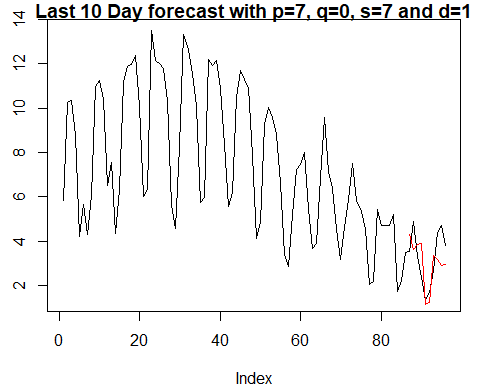


## $autplt  
## [1] 1.000000000 0.019981407 -0.044545692 0.007099925 -0.083089823  
## [6] 0.023080402 -0.027706775 -0.109170850 -0.018925376 -0.126356402  
## [11] -0.092552967 -0.114069069 0.082793873 0.062516232 -0.124033594  
## [16] 0.008950829 0.041389898 0.007846784 0.038008294 0.030155077  
## [21] 0.027149544 -0.062078547 -0.073221877 -0.012868532 0.081410078  
## [26] -0.046902137  
##   
## $freq  
## [1] 0.01136364 0.02272727 0.03409091 0.04545455 0.05681818 0.06818182  
## [7] 0.07954545 0.09090909 0.10227273 0.11363636 0.12500000 0.13636364  
## [13] 0.14772727 0.15909091 0.17045455 0.18181818 0.19318182 0.20454545  
## [19] 0.21590909 0.22727273 0.23863636 0.25000000 0.26136364 0.27272727  
## [25] 0.28409091 0.29545455 0.30681818 0.31818182 0.32954545 0.34090909  
## [31] 0.35227273 0.36363636 0.37500000 0.38636364 0.39772727 0.40909091  
## [37] 0.42045455 0.43181818 0.44318182 0.45454545 0.46590909 0.47727273  
## [43] 0.48863636 0.50000000  
##   
## $db  
## [1] -3.9515446 -4.2113149 1.6461729 1.9041508 -4.1162478  
## [6] 5.3562119 -1.9262027 -2.5646471 -0.5698288 -1.0870752  
## [11] -6.4815411 1.8317864 -10.1318551 2.9763482 4.0479796  
## [16] -0.8112941 -12.2578872 0.3713146 1.6279814 -6.2174057  
## [21] -1.3637767 5.4815943 -2.5610656 -7.9519876 -12.8394052  
## [26] -3.4424914 -2.6091713 4.2250037 1.4172525 -8.7457783  
## [31] 2.2450812 -2.9041104 0.6079985 2.6193152 3.2860079  
## [36] -4.8926547 1.5509569 -6.9836817 -24.5516785 0.2680124  
## [41] 0.6193338 -1.0847123 -4.9570825 1.6174128  
##   
## $dbz  
## [1] -1.772213069 -1.056868847 -0.278226542 0.328301022 0.669868529  
## [6] 0.742143947 0.596192940 0.328688985 0.070812262 -0.052615868  
## [11] 0.009380827 0.202567318 0.414060526 0.545282004 0.553450055  
## [16] 0.459072068 0.330792819 0.245775260 0.235163742 0.258006205  
## [21] 0.233569870 0.101463605 -0.133996286 -0.391235396 -0.543798021  
## [26] -0.503069101 -0.287685712 0.004070721 0.278920572 0.495272553  
## [31] 0.653487118 0.761116457 0.805073115 0.749266078 0.555232274  
## [36] 0.210021686 -0.250655220 -0.732578349 -1.104849684 -1.268001564  
## [41] -1.224346646 -1.070402317 -0.923816720 -0.865810634

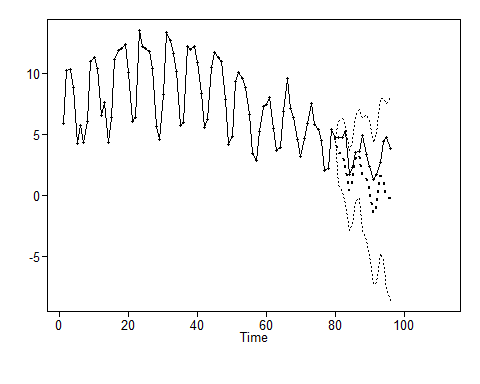
#Forecast last 10  
for\_aruma2\_s7d1 = fore.aruma.wge(Post\_Vaccine$new\_deaths\_per\_million,d=1,s=7, phi = arima\_s7\_d1$phi,n.ahead = 10, lastn = T)



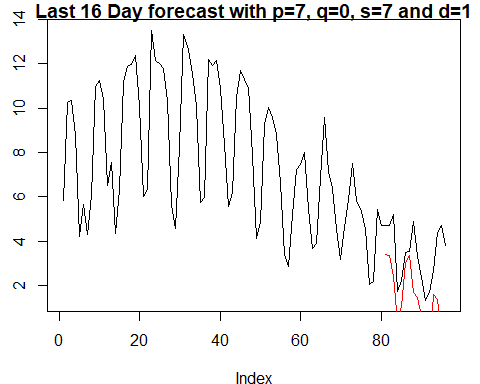
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,96), main = "Last 10 Day forecast with p=7, q=0, s=7 and d=1")  
lines(seq(87,96,1),for\_aruma2\_s7d1$f, col = "red")



#Forecast last 16  
for\_aruma2\_s7d1\_16 = fore.aruma.wge(Post\_Vaccine$new\_deaths\_per\_million,d=1,s=7, phi = arima\_s7\_d1$phi,n.ahead = 16, lastn = T)



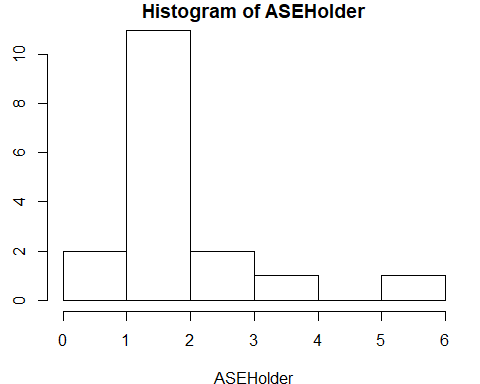
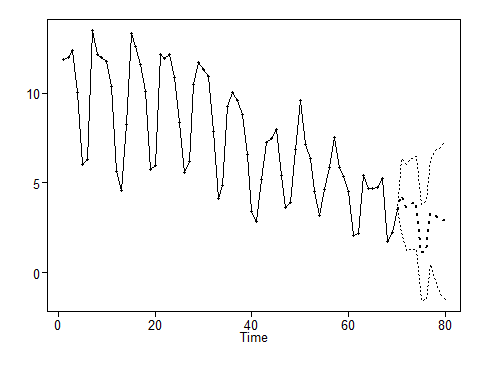
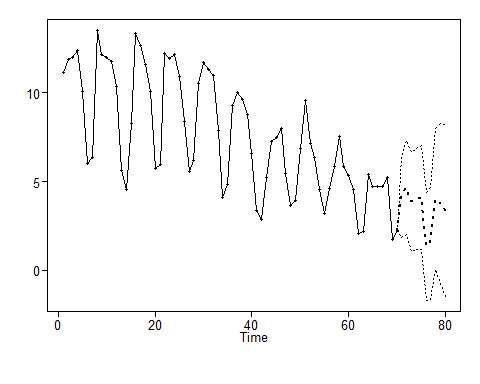
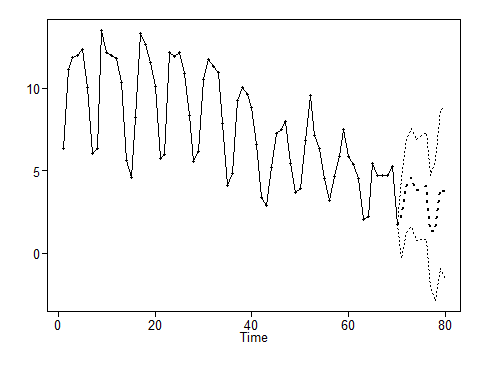
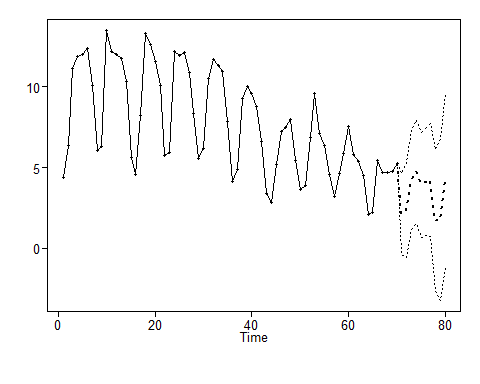
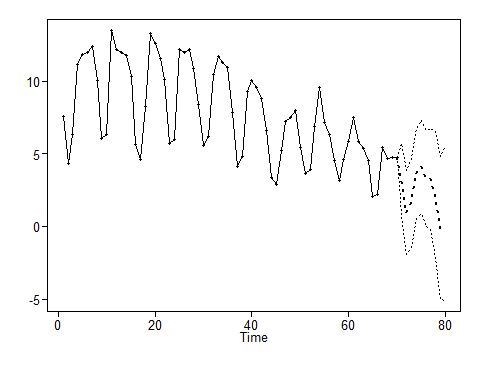
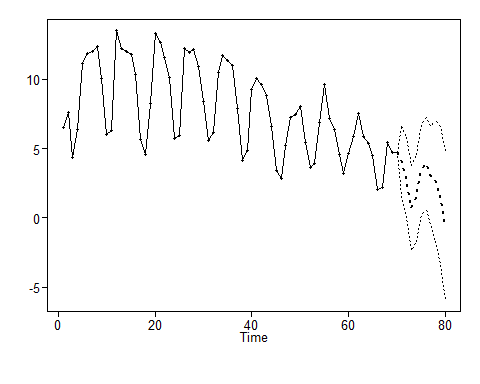
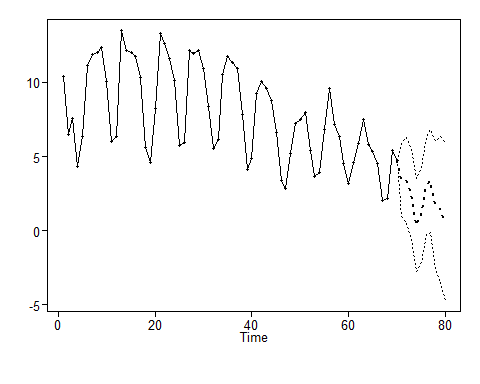
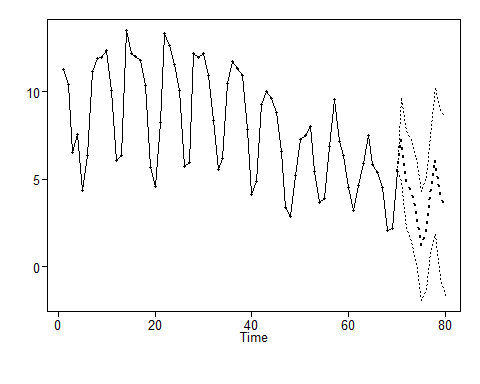
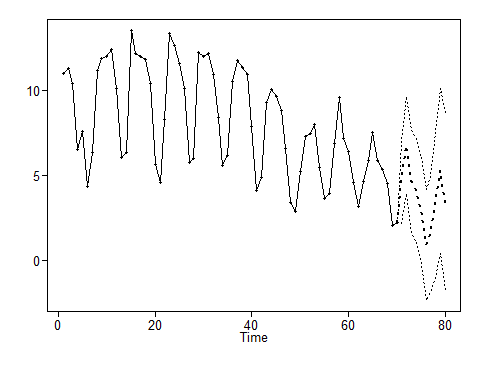
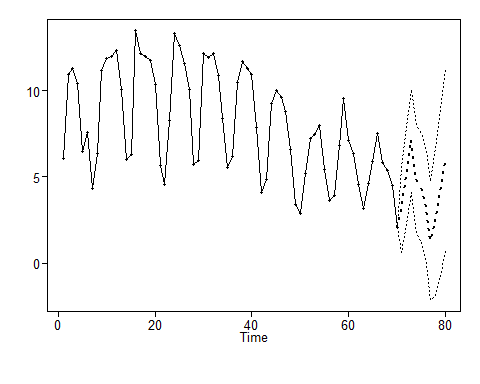
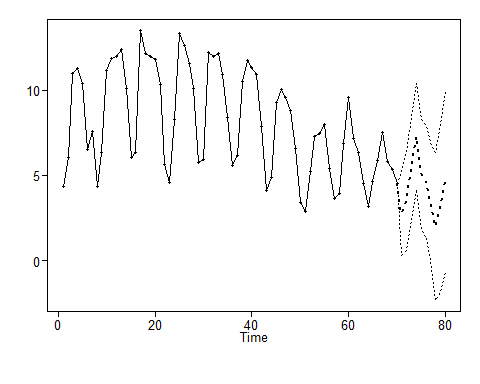
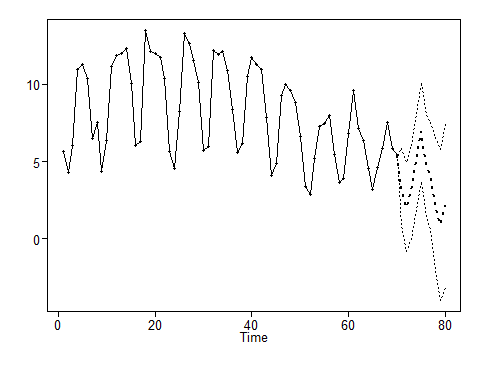
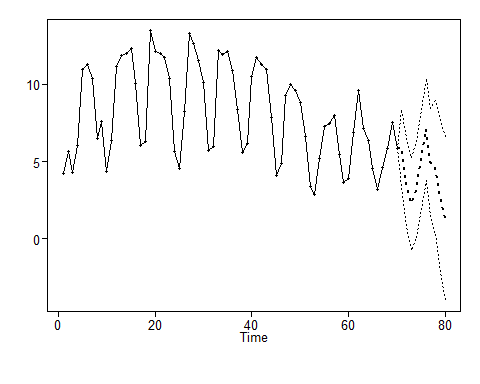
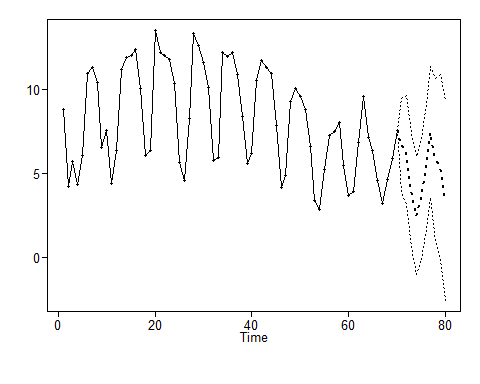
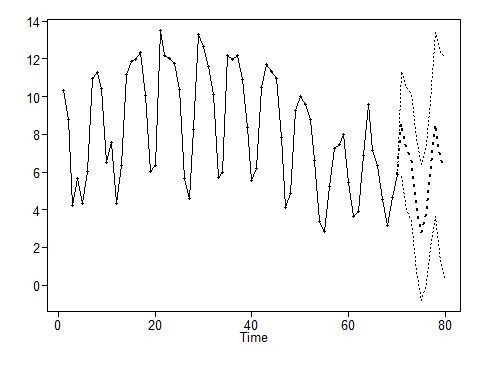
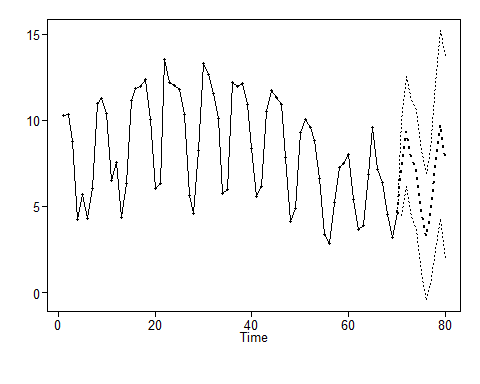
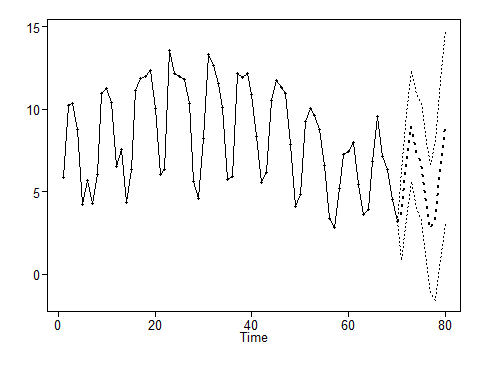
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,96), main = "Last 16 Day forecast with p=7, q=0, s=7 and d=1")  
lines(seq(81,96,1),for\_aruma2\_s7d1\_16$f, col = "red")



#ASE  
arma\_s7\_d1\_ase = mean((for\_aruma2\_s7d1\_16$f - Post\_Vaccine$new\_deaths\_per\_million[(96-16+1):96])^2)  
arma\_s7\_d1\_ase

## [1] 6.068007

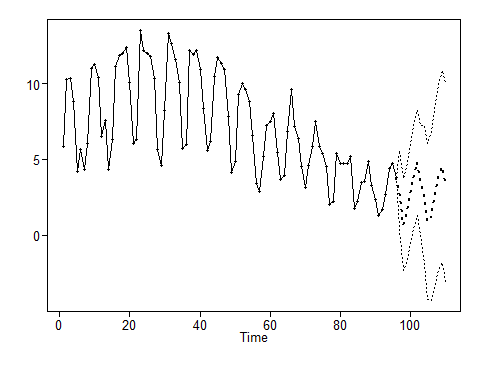
#Rolling\_Window\_ASE  
Rolling\_Window\_ASE(Post\_Vaccine$new\_deaths\_per\_million, trainingSize = 70, horizon = 10, d = 1, phis = arima\_s7\_d1$phi,  
 s= 7, theta = 0)



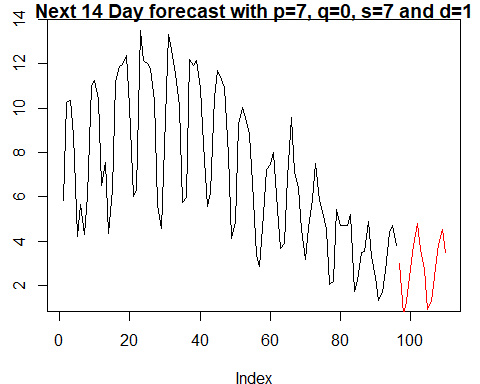
## [1] 10  
## [1] 70  
## [1] "The Summary Statistics for the Rolling Window ASE Are:"  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.805 1.201 1.616 1.924 1.877 5.670   
## [1] "The Rolling Window ASE is: 1.92382631816256"

## [1] 1.923826

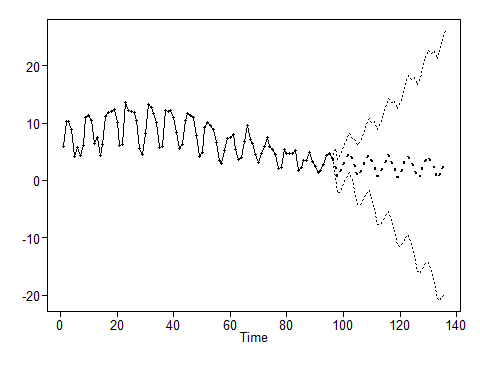
#Future forecasts  
for\_aruma2\_s7d1\_f14 = fore.aruma.wge(Post\_Vaccine$new\_deaths\_per\_million,d=1, s=7, phi = arima\_s7\_d1$phi,n.ahead = 14)



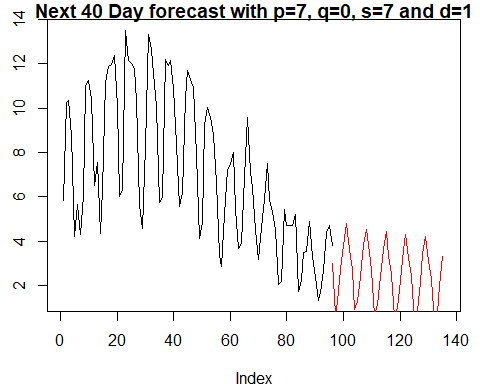
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,110), main = "Next 14 Day forecast with p=7, q=0, s=7 and d=1")  
lines(seq(97,110,1),for\_aruma2\_s7d1\_f14$f, col = "red")



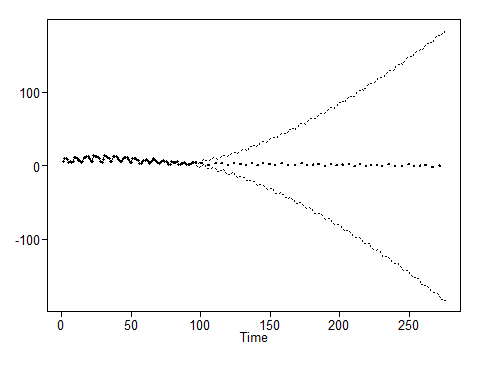
for\_aruma2\_s7d1\_f40 = fore.aruma.wge(Post\_Vaccine$new\_deaths\_per\_million,d=1,s=7, phi = arima\_s7\_d1$phi,n.ahead = 40)



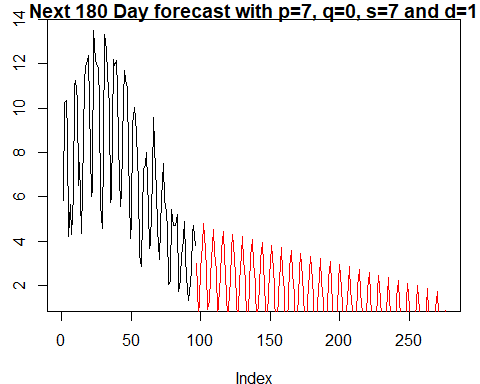
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,136), main = "Next 40 Day forecast with p=7, q=0, s=7 and d=1")  
lines(seq(96,135,1),for\_aruma2\_s7d1\_f40$f, col = "red")



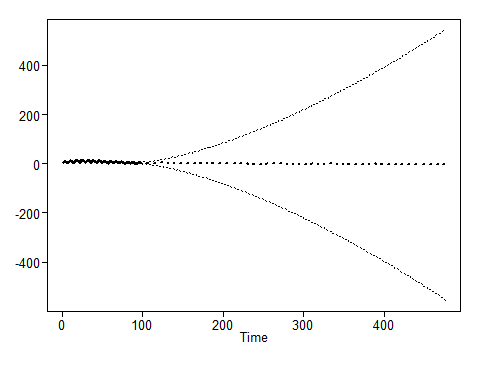
for\_aruma2\_s7d1\_f180 = fore.aruma.wge(Post\_Vaccine$new\_deaths\_per\_million,d=1,s=7, phi = arima\_s7\_d1$phi,n.ahead = 180)



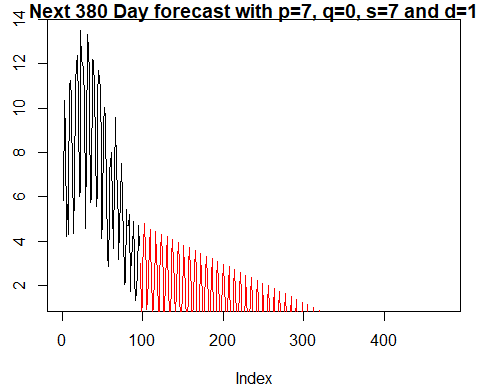
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,276), main = "Next 180 Day forecast with p=7, q=0, s=7 and d=1")  
lines(seq(97,276,1),for\_aruma2\_s7d1\_f180$f, col = "red")



for\_aruma2\_s7d1\_f380 = fore.aruma.wge(Post\_Vaccine$new\_deaths\_per\_million,d=1,s=7, phi = arima\_s7\_d1$phi,n.ahead = 380)



plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,476), main = "Next 380 Day forecast with p=7, q=0, s=7 and d=1")  
lines(seq(97,476,1),for\_aruma2\_s7d1\_f380$f, col = "red")



### Multivariate - model with only vaccine variable

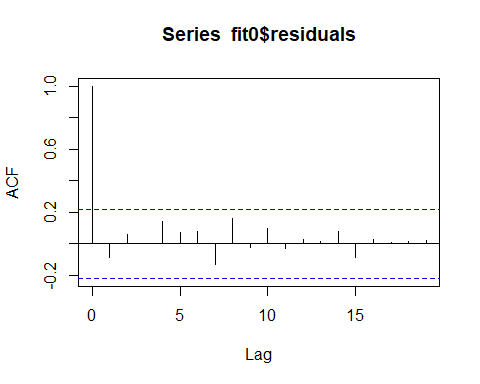
# Considering Only new\_vaccinations\_smoothed\_per\_million  
PVsmall = Post\_Vaccine[1:80,]  
mv\_fit0 <- lm( new\_deaths\_per\_million~ new\_vaccinations\_smoothed\_per\_million ,data = PVsmall)  
aic.wge(mv\_fit0$residuals, p=0:8,q=0) #AIC picks 8,0 ; aic =0.8019881

## $type  
## [1] "aic"  
##   
## $value  
## [1] 0.6664726  
##   
## $p  
## [1] 8  
##   
## $q  
## [1] 0  
##   
## $phi  
## [1] 0.59563682 -0.12304724 0.04186753 -0.03825942 -0.06340488 0.18244759  
## [7] 0.60000280 -0.28433640  
##   
## $theta  
## [1] 0  
##   
## $vara  
## [1] 1.554995

fit0 = arima(PVsmall$new\_deaths\_per\_million, order=c(8,0,0), xreg = PVsmall[,c(7)] )  
fit0 #AIC 295.99; Only new\_vaccinations\_smoothed\_per\_million looks significant

##   
## Call:  
## arima(x = PVsmall$new\_deaths\_per\_million, order = c(8, 0, 0), xreg = PVsmall[,   
## c(7)])  
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ar5 ar6 ar7 ar8  
## 0.5969 -0.1228 0.0419 -0.0381 -0.0633 0.1824 0.6005 -0.2859  
## s.e. 0.1102 0.1081 0.1071 0.1115 0.1122 0.1132 0.1114 0.1137  
## intercept PVsmall[, c(7)]  
## 9.0735 -7e-04  
## s.e. 1.6740 3e-04  
##   
## sigma^2 estimated as 1.662: log likelihood = -136.99, aic = 295.99

acf(fit0$residuals) #appear to be white noise



#ljung test for white noise of residuals  
ltest = ljung.wge(fit0$residuals) #null hypothesis = white noise, alternate- not white noise. pval = 0.975. Here we FTR null hypothesis, so this is white noise

## Obs -0.08933317 0.05728943 -0.0001381661 0.1447606 0.07306859 0.07705902 -0.1333722 0.1628458 -0.02067391 0.09979599 -0.0281532 0.02821667 0.01406468 0.07994413 -0.08846367 0.02845776 0.008774531 0.01521167 0.01981078 0.04623145 0.03766596 -0.09266563 -0.02844871 0.1111617

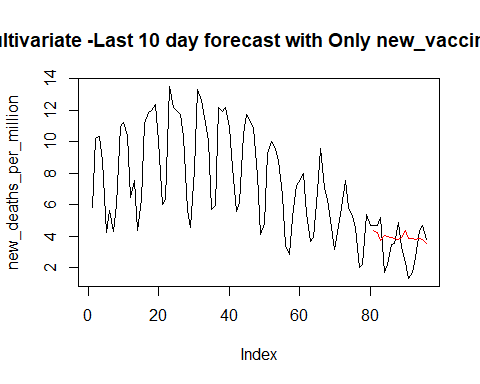
ltest

## $test  
## [1] "Ljung-Box test"  
##   
## $K  
## [1] 24  
##   
## $chi.square  
## [1] 13.38595  
##   
## $df  
## [1] 24  
##   
## $pval  
## [1] 0.9593268

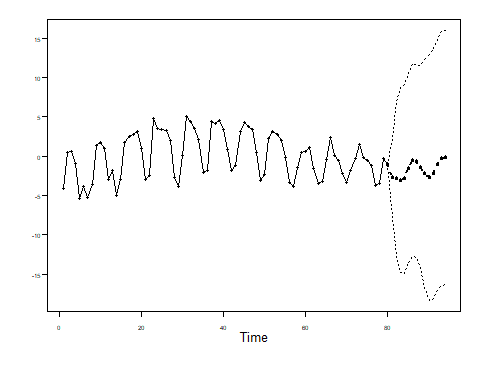
preds\_mv0 = predict(fit0, newxreg = (Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million[81:96] ))  
ASE\_mv0 = mean((Post\_Vaccine$new\_deaths\_per\_million[81:96] - preds\_mv0$pred)^2)  
ASE\_mv0 #1.896324

## [1] 1.896324

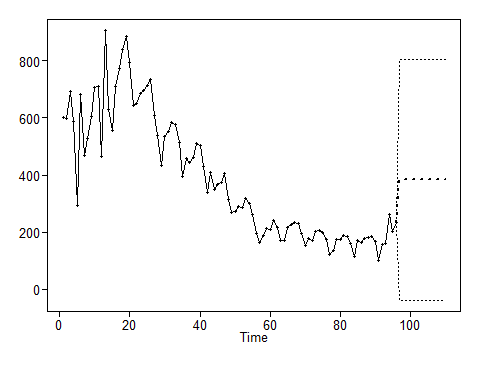
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,96), ylab = "new\_deaths\_per\_million", main = "Multivariate -Last 10 day forecast with Only new\_vaccinations")  
lines(seq(81,96,1),preds\_mv0$pred, col = "red") # Even thought the ASE is low, plot doesn't capture the trend very well.



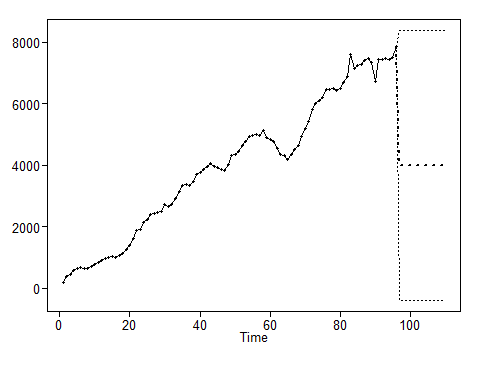
phi = aic.wge(mv\_fit0$residuals)  
m1\_resids\_0 = fore.arma.wge(mv\_fit0$residuals, phi = phi$phi, n.ahead = 14)



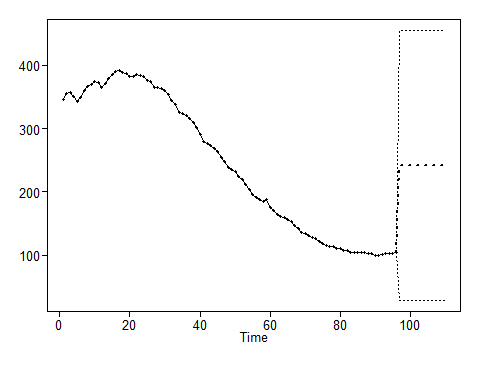
#Predicting future forecasts - 14  
Pred\_nc\_10\_m1 = fore.aruma.wge(Post\_Vaccine$new\_cases\_per\_million, n.ahead = 14)



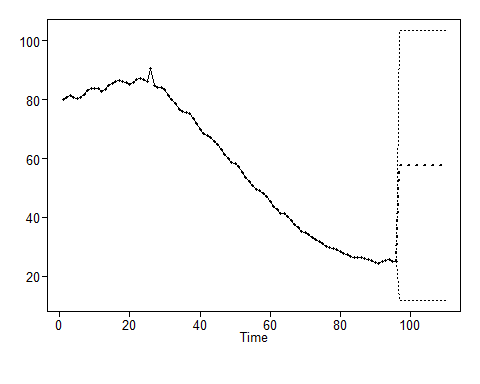
Pred\_nv\_10\_m1 = fore.aruma.wge(Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million, n.ahead = 14)



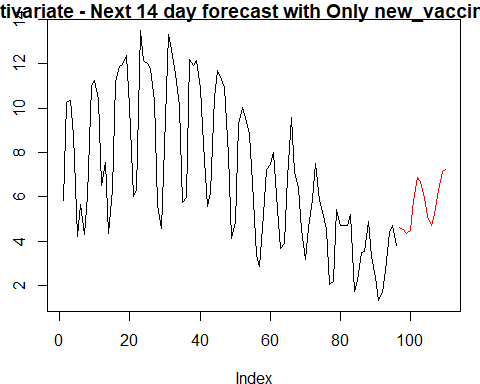
Pred\_hs\_10\_m1 = fore.aruma.wge(Post\_Vaccine$hosp\_patients\_per\_million, n.ahead = 14)



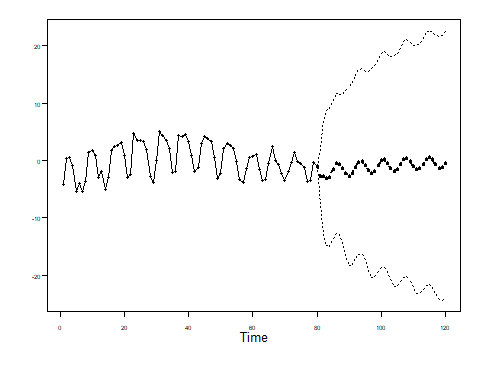
Pred\_ic\_10\_m1 = fore.aruma.wge(Post\_Vaccine$icu\_patients\_per\_million, n.ahead = 14)



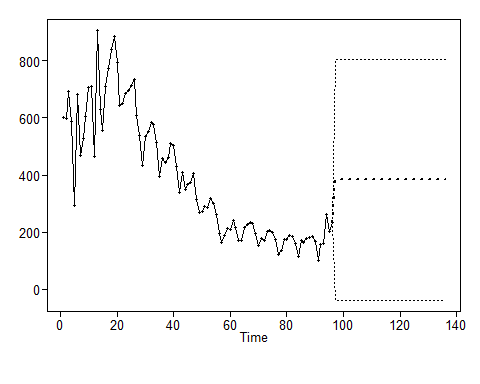
next10\_m1 = data.frame(new\_cases\_per\_million=Pred\_nc\_10\_m1$f, new\_vaccinations\_smoothed\_per\_million=Pred\_nv\_10\_m1$f,hosp\_patients\_per\_million=Pred\_hs\_10\_m1$f,icu\_patients\_per\_million=Pred\_ic\_10\_m1$f)  
#get predictions  
preds\_m1\_10 = predict(mv\_fit0, newdata = next10\_m1)  
Preds\_m1\_10\_final = preds\_m1\_10 + m1\_resids\_0$f  
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,110), ylab = "new\_deaths\_per\_million", main = "Multivariate - Next 14 day forecast with Only new\_vaccinations")  
lines(seq(97,110,1),Preds\_m1\_10\_final, col = "red")



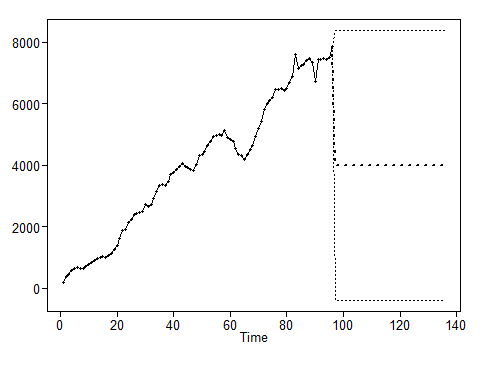
#Predicting future forecasts - 40  
m1\_resids\_40 = fore.arma.wge(mv\_fit0$residuals, phi = phi$phi, n.ahead = 40)



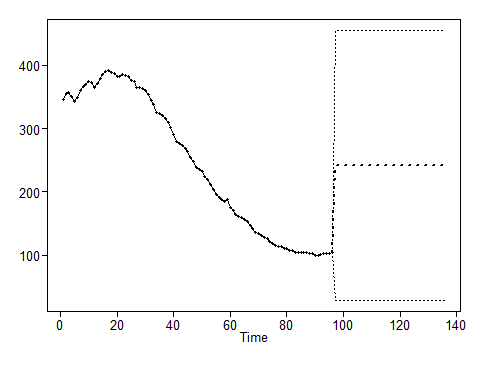
Pred\_nc\_40\_m1 = fore.aruma.wge(Post\_Vaccine$new\_cases\_per\_million, n.ahead = 40)



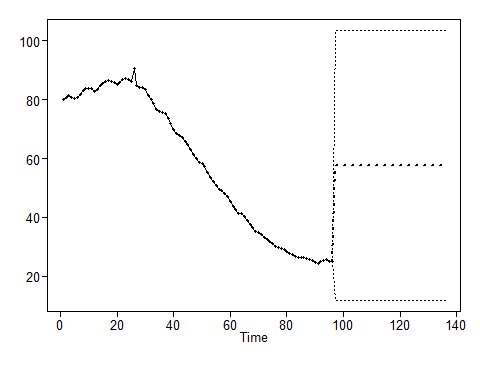
Pred\_nv\_40\_m1 = fore.aruma.wge(Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million, n.ahead = 40)



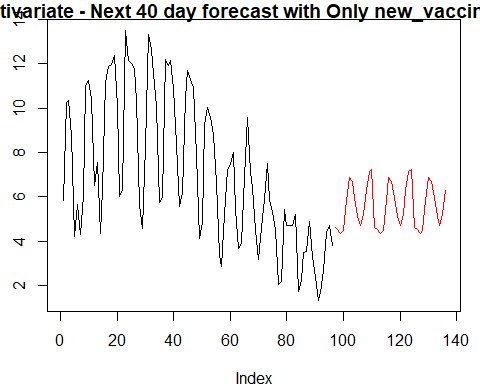
Pred\_hs\_40\_m1 = fore.aruma.wge(Post\_Vaccine$hosp\_patients\_per\_million, n.ahead = 40)



Pred\_ic\_40\_m1 = fore.aruma.wge(Post\_Vaccine$icu\_patients\_per\_million, n.ahead = 40)



next40\_m1 = data.frame(new\_cases\_per\_million=Pred\_nc\_40\_m1$f, new\_vaccinations\_smoothed\_per\_million=Pred\_nv\_40\_m1$f,hosp\_patients\_per\_million=Pred\_hs\_40\_m1$f,icu\_patients\_per\_million=Pred\_ic\_40\_m1$f)  
#get predictions  
preds\_m1\_40 = predict(mv\_fit0, newdata = next40\_m1)  
Preds\_m1\_40\_final = preds\_m1\_40 + m1\_resids\_0$f  
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,136), ylab = "new\_deaths\_per\_million", main = "Multivariate - Next 40 day forecast with Only new\_vaccinations")  
lines(seq(97,136,1),Preds\_m1\_40\_final, col = "red")



### Multivariate - model with all explanatory variables

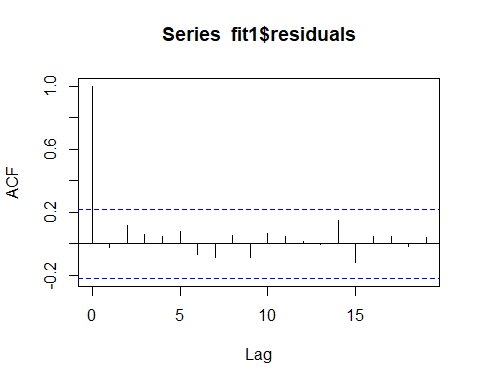
#Multivariate - Model 1 - No trend and no lags  
#PVsmall = Post\_Vaccine[1:80,]  
mv\_fit1 <- lm( new\_deaths\_per\_million~ new\_cases\_per\_million+ icu\_patients\_per\_million+hosp\_patients\_per\_million+new\_vaccinations\_smoothed\_per\_million ,data = PVsmall)  
aic.wge(mv\_fit1$residuals, p=0:8,q=0) #AIC picks 7,0 ; aic =0.8019881

## $type  
## [1] "aic"  
##   
## $value  
## [1] 0.8019881  
##   
## $p  
## [1] 7  
##   
## $q  
## [1] 0  
##   
## $phi  
## [1] 0.229948469 -0.036463718 -0.170439101 -0.006486376 -0.079572531  
## [6] 0.087570618 0.489722948  
##   
## $theta  
## [1] 0  
##   
## $vara  
## [1] 1.825745

fit1 = arima(PVsmall$new\_deaths\_per\_million, order=c(7,0,0), xreg = PVsmall[,c(3,5,6,7)] )  
fit1 #AIC 281.45; Only new\_cases\_per\_million looks significant

##   
## Call:  
## arima(x = PVsmall$new\_deaths\_per\_million, order = c(7, 0, 0), xreg = PVsmall[,   
## c(3, 5, 6, 7)])  
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ar5 ar6 ar7 intercept  
## 0.4278 -0.1509 -0.0288 0.0719 -0.1452 0.2669 0.4721 1.8616  
## s.e. 0.1051 0.1161 0.1189 0.1162 0.1119 0.1099 0.1022 2.7545  
## new\_cases\_per\_million icu\_patients\_per\_million  
## 0.0080 0.1453  
## s.e. 0.0018 0.1258  
## hosp\_patients\_per\_million new\_vaccinations\_smoothed\_per\_million  
## -0.0274 -1e-04  
## s.e. 0.0292 3e-04  
##   
## sigma^2 estimated as 1.329: log likelihood = -127.72, aic = 281.45

acf(fit1$residuals) #appear to be white noise



#ljung test for white noise of residuals  
ltest = ljung.wge(fit1$residuals) #null hypothesis = white noise, alternate- not white noise. pval = 0.975. Here we FTR null hypothesis, so this is white noise

## Obs -0.0225653 0.1170221 0.06159769 0.04841578 0.08116654 -0.06688804 -0.08781792 0.05318628 -0.08540389 0.06803597 0.04596115 0.01316819 -0.004960854 0.1465708 -0.1169755 0.04841578 0.04412049 -0.01465749 0.04264349 -0.05848696 0.04144058 -0.09182495 -0.07552927 0.08273832

ltest

## $test  
## [1] "Ljung-Box test"  
##   
## $K  
## [1] 24  
##   
## $chi.square  
## [1] 12.12541  
##   
## $df  
## [1] 24  
##   
## $pval  
## [1] 0.978458

#Forecast model 1  
preds\_mv1 = predict(fit1, newxreg = cbind(Post\_Vaccine$new\_cases\_per\_million[81:96], Post\_Vaccine$icu\_patients\_per\_million[81:96],Post\_Vaccine$hosp\_patients\_per\_million[81:96],Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million[81:96]))  
ASE\_mv1 = mean((Post\_Vaccine$new\_deaths\_per\_million[81:96] - preds\_mv1$pred)^2)  
ASE\_mv1 #2.59

## [1] 2.573935

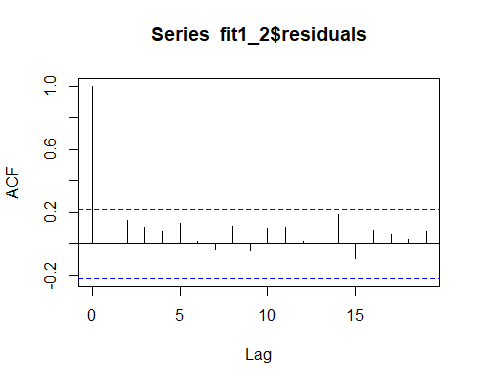
#Multivariate - No trend and no lags with only new\_cases\_per\_million  
mv\_fit1\_2 <- lm( new\_deaths\_per\_million~ new\_cases\_per\_million ,data = PVsmall)  
aic.wge(mv\_fit1\_2$residuals, p=0:8,q=0) #AIC picks 7,0 ; aic =0.471746

## $type  
## [1] "aic"  
##   
## $value  
## [1] 0.471746  
##   
## $p  
## [1] 7  
##   
## $q  
## [1] 0  
##   
## $phi  
## [1] 0.44078828 -0.17531875 0.01392374 0.02483295 -0.10761464 0.25213304  
## [7] 0.46623873  
##   
## $theta  
## [1] 0  
##   
## $vara  
## [1] 1.312254

fit1\_2 = arima(PVsmall$new\_deaths\_per\_million, order=c(7,0,0), xreg = PVsmall[,c(3)] )  
fit1\_2 #AIC 281.45; Only new\_cases\_per\_million looks significant

##   
## Call:  
## arima(x = PVsmall$new\_deaths\_per\_million, order = c(7, 0, 0), xreg = PVsmall[,   
## c(3)])  
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ar5 ar6 ar7 intercept  
## 0.4524 -0.1862 0.0192 0.0254 -0.1162 0.2612 0.4571 3.3906  
## s.e. 0.1035 0.1132 0.1135 0.1144 0.1134 0.1113 0.1015 1.2187  
## PVsmall[, c(3)]  
## 0.0081  
## s.e. 0.0014  
##   
## sigma^2 estimated as 1.365: log likelihood = -128.71, aic = 277.43

acf(fit1\_2$residuals) #appear to be white noise



#ljung test for white noise of residuals  
ltest = ljung.wge(fit1\_2$residuals) #null hypothesis = white noise, alternate- not white noise. pval = 0.975. Here we FTR null hypothesis, so this is white noise

## Obs 0.0003100592 0.1499663 0.1034955 0.07902539 0.1275278 0.01686719 -0.03732423 0.1109621 -0.04114831 0.09704431 0.1055418 0.016953 0.0003275814 0.184288 -0.09100432 0.08281432 0.06161375 0.02780213 0.07697265 -0.03159458 0.05110265 -0.07440334 -0.07181274 0.07211165

ltest

## $test  
## [1] "Ljung-Box test"  
##   
## $K  
## [1] 24  
##   
## $chi.square  
## [1] 16.42048  
##   
## $df  
## [1] 24  
##   
## $pval  
## [1] 0.8723045

#Forecast model 1\_2  
preds\_mv1\_2 = predict(fit1\_2, newxreg = cbind(Post\_Vaccine$new\_cases\_per\_million[81:96] ))  
ASE\_mv1\_2 = mean((Post\_Vaccine$new\_deaths\_per\_million[81:96] - preds\_mv1\_2$pred)^2)  
ASE\_mv1\_2 #3.736719

## [1] 3.736719

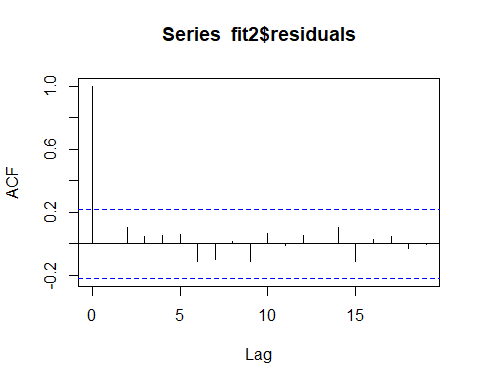
#Model 1 - No trend and no lags has the best ASE so far.  
  
  
#Multivariate - Model 2 - Trend but no lags  
t = seq(1:96)  
mv\_fit2 <- lm( new\_deaths\_per\_million~ t[1:80]+new\_cases\_per\_million+ icu\_patients\_per\_million+hosp\_patients\_per\_million+new\_vaccinations\_smoothed\_per\_million ,data = PVsmall)  
aic.wge(mv\_fit2$residuals, p=0:8,q=0) #AIC picks 7,0, aic = 0.6252096

## $type  
## [1] "aic"  
##   
## $value  
## [1] 0.6252096  
##   
## $p  
## [1] 7  
##   
## $q  
## [1] 0  
##   
## $phi  
## [1] 0.170524799 -0.044142741 -0.190096497 -0.008118636 -0.109335267  
## [6] 0.075185106 0.527296313  
##   
## $theta  
## [1] 0  
##   
## $vara  
## [1] 1.529911

fit2 = arima(PVsmall$new\_deaths\_per\_million, order=c(7,0,0), xreg = cbind(t[1:80],PVsmall[,c(3,5,6,7)]) )  
fit2 #AIC aic = 274.31; time, hosp\_patients\_per\_million looks insignificant,

##   
## Call:  
## arima(x = PVsmall$new\_deaths\_per\_million, order = c(7, 0, 0), xreg = cbind(t[1:80],   
## PVsmall[, c(3, 5, 6, 7)]))  
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ar5 ar6 ar7 intercept  
## 0.3321 -0.1440 -0.1179 0.0659 -0.2102 0.2152 0.4698 -7.3622  
## s.e. 0.1137 0.1106 0.1278 0.1090 0.1180 0.1127 0.1110 3.6489  
## t[1:80] new\_cases\_per\_million icu\_patients\_per\_million  
## 0.1466 0.0075 0.3596  
## s.e. 0.0422 0.0018 0.1615  
## hosp\_patients\_per\_million new\_vaccinations\_smoothed\_per\_million  
## -0.0538 -8e-04  
## s.e. 0.0326 4e-04  
##   
## sigma^2 estimated as 1.197: log likelihood = -123.16, aic = 274.31

acf(fit2$residuals) #appear to be white noise



#ljung test for white noise of residuals  
ltest2 = ljung.wge(fit2$residuals) #null hypothesis = white noise, alternate- not white noise. pval = 0.975. Here we FTR null hypothesis, so this is white noise

## Obs 0.004095523 0.100914 0.04429023 0.05263636 0.05838233 -0.1117387 -0.09824692 0.01441274 -0.1121791 0.06319369 -0.01318881 0.05060121 0.002351999 0.103157 -0.1144501 0.0260979 0.04860438 -0.03123468 -0.004483388 -0.09553879 0.0130966 -0.1264479 -0.1618971 -0.003617753

ltest2

## $test  
## [1] "Ljung-Box test"  
##   
## $K  
## [1] 24  
##   
## $chi.square  
## [1] 14.0008  
##   
## $df  
## [1] 24  
##   
## $pval  
## [1] 0.9466322

#Forecast model 2  
preds\_mv2 = predict(fit2, newxreg = cbind(t[81:96], Post\_Vaccine$new\_cases\_per\_million[81:96], Post\_Vaccine$icu\_patients\_per\_million[81:96],Post\_Vaccine$hosp\_patients\_per\_million[81:96],Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million[81:96]))  
ASE\_mv2 = mean((Post\_Vaccine$new\_deaths\_per\_million[81:96] - preds\_mv2$pred)^2)  
ASE\_mv2 #5.407704

## [1] 5.271115

#Take hosp\_patients\_per\_million out  
mv\_fit2\_1 <- lm( new\_deaths\_per\_million~ t[1:80]+new\_cases\_per\_million+ icu\_patients\_per\_million +new\_vaccinations\_smoothed\_per\_million ,data = PVsmall)  
aic.wge(mv\_fit2\_1$residuals, p=0:8,q=0) #AIC picks 7,0, aic = 0.6189045

## $type  
## [1] "aic"  
##   
## $value  
## [1] 0.6189045  
##   
## $p  
## [1] 7  
##   
## $q  
## [1] 0  
##   
## $phi  
## [1] 0.39867963 -0.13735621 -0.03605856 0.02013120 -0.10836136 0.21409378  
## [7] 0.45756199  
##   
## $theta  
## [1] 0  
##   
## $vara  
## [1] 1.520295

fit2\_1 = arima(PVsmall$new\_deaths\_per\_million, order=c(7,0,0), xreg = cbind(t[1:80],PVsmall[,c(3,5,7)]) )  
fit2\_1 #AIC aic = 274.31; time, new\_vaccinations\_smoothed\_per\_million looks insignificant

##   
## Call:  
## arima(x = PVsmall$new\_deaths\_per\_million, order = c(7, 0, 0), xreg = cbind(t[1:80],   
## PVsmall[, c(3, 5, 7)]))  
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ar5 ar6 ar7 intercept  
## 0.4028 -0.1789 -0.0162 0.0318 -0.1562 0.2458 0.4735 -5.5820  
## s.e. 0.1004 0.1079 0.1096 0.1103 0.1102 0.1072 0.0997 3.6068  
## t[1:80] new\_cases\_per\_million icu\_patients\_per\_million  
## 0.1274 0.0064 0.1128  
## s.e. 0.0453 0.0016 0.0427  
## new\_vaccinations\_smoothed\_per\_million  
## -6e-04  
## s.e. 4e-04  
##   
## sigma^2 estimated as 1.238: log likelihood = -124.7, aic = 275.4

#Forecast model 2\_1  
preds\_mv2\_1 = predict(fit2\_1, newxreg = cbind(t[81:96], Post\_Vaccine$new\_cases\_per\_million[81:96], Post\_Vaccine$icu\_patients\_per\_million[81:96],Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million[81:96] ))  
ASE\_mv2\_1 = mean((Post\_Vaccine$new\_deaths\_per\_million[81:96] - preds\_mv2\_1$pred)^2)  
ASE\_mv2\_1 #6.364181

## [1] 6.315825

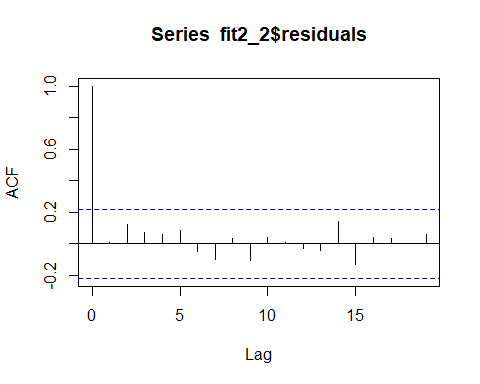
#Take new\_vaccinations\_smoothed\_per\_million out  
mv\_fit2\_2 <- lm( new\_deaths\_per\_million~ t[1:80]+new\_cases\_per\_million+ icu\_patients\_per\_million ,data = PVsmall)  
aic.wge(mv\_fit2\_2$residuals, p=0:8,q=0) #AIC picks 7,0, aic = 0.5544063

## $type  
## [1] "aic"  
##   
## $value  
## [1] 0.5544063  
##   
## $p  
## [1] 7  
##   
## $q  
## [1] 0  
##   
## $phi  
## [1] 0.39279569 -0.13942168 -0.03341932 0.02502220 -0.11431402 0.21775082  
## [7] 0.47087041  
##   
## $theta  
## [1] 0  
##   
## $vara  
## [1] 1.425334

fit2\_2 = arima(PVsmall$new\_deaths\_per\_million, order=c(7,0,0), xreg = cbind(t[1:80],PVsmall[,c(3,5)]))  
fit2\_2 #AIC aic = 275.97; time, AIC increased a little bit

##   
## Call:  
## arima(x = PVsmall$new\_deaths\_per\_million, order = c(7, 0, 0), xreg = cbind(t[1:80],   
## PVsmall[, c(3, 5)]))  
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ar5 ar6 ar7 intercept  
## 0.4102 -0.1847 -0.0235 0.0327 -0.1625 0.2481 0.4553 -5.4533  
## s.e. 0.1014 0.1100 0.1111 0.1116 0.1122 0.1084 0.1020 3.6641  
## t[1:80] new\_cases\_per\_million icu\_patients\_per\_million  
## 0.0794 0.0068 0.1072  
## s.e. 0.0342 0.0016 0.0430  
##   
## sigma^2 estimated as 1.283: log likelihood = -125.98, aic = 275.97

acf(fit2\_2$residuals) #appear to be white noise



#ljung test for white noise of residuals  
ltest2\_2 = ljung.wge(fit2\_2$residuals) #null hypothesis = white noise, alternate- not white noise. pval = 0.975. Here we FTR null hypothesis, so this is white noise

## Obs 0.007809401 0.1246527 0.06976255 0.05935119 0.08727446 -0.04774794 -0.1018511 0.03612639 -0.1029307 0.03813898 0.01007962 -0.03007361 -0.04276539 0.1407171 -0.1332761 0.04274424 0.03431625 0.00186642 0.05645683 -0.05586972 0.03362173 -0.1008829 -0.10878 0.04155017

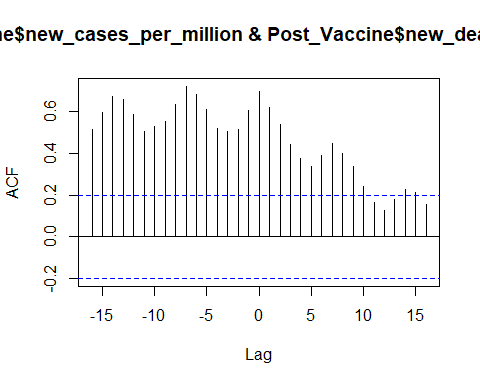
ltest2\_2

## $test  
## [1] "Ljung-Box test"  
##   
## $K  
## [1] 24  
##   
## $chi.square  
## [1] 12.93714  
##   
## $df  
## [1] 24  
##   
## $pval  
## [1] 0.9671452

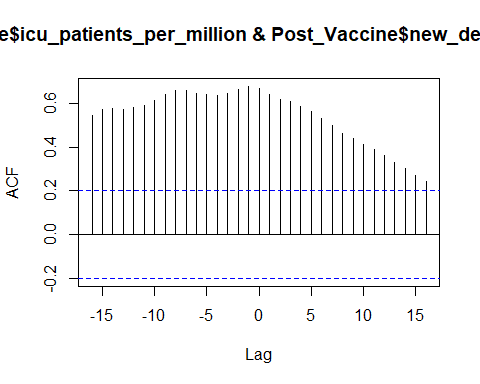
#Forecast model 2\_1  
preds\_mv2\_2 = predict(fit2\_2, newxreg = cbind(t[81:96], Post\_Vaccine$new\_cases\_per\_million[81:96], Post\_Vaccine$icu\_patients\_per\_million[81:96]))  
ASE\_mv2\_2 = mean((Post\_Vaccine$new\_deaths\_per\_million[81:96] - preds\_mv2\_2$pred)^2)  
ASE\_mv2\_2 #6.53061 No improvememnt

## [1] 6.477088

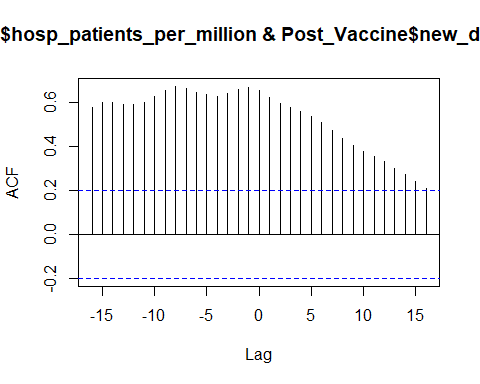
#Model 1 - No trend and no lags has the best ASE so far.  
#Multivariate - Model 2 - Trend and lags  
#Lagged variables - all  
ccf(Post\_Vaccine$new\_cases\_per\_million,Post\_Vaccine$new\_deaths\_per\_million) #7



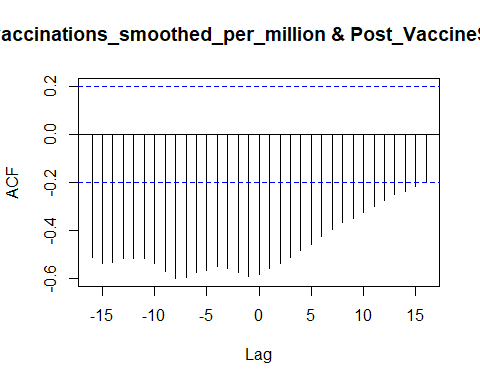
ccf(Post\_Vaccine$icu\_patients\_per\_million,Post\_Vaccine$new\_deaths\_per\_million) #1



ccf(Post\_Vaccine$hosp\_patients\_per\_million,Post\_Vaccine$new\_deaths\_per\_million) #1



ccf(Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million,Post\_Vaccine$new\_deaths\_per\_million) #8



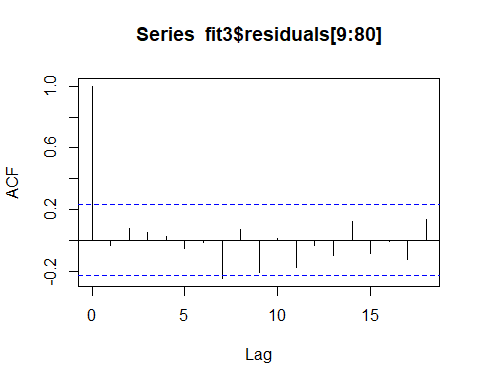
Post\_Vaccine$new\_cases\_per\_million\_7 <- dplyr::lag(Post\_Vaccine$new\_cases\_per\_million,7)  
Post\_Vaccine$icu\_patients\_per\_million\_1 <- dplyr::lag(Post\_Vaccine$icu\_patients\_per\_million,1)  
Post\_Vaccine$hosp\_patients\_per\_million\_1 <- dplyr::lag(Post\_Vaccine$hosp\_patients\_per\_million,1)  
Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million\_8 <- dplyr::lag(Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million,8)  
PVsmall = Post\_Vaccine[1:80,]  
mv\_fit3 <- lm( new\_deaths\_per\_million~ t[1:80]+new\_cases\_per\_million\_7+ icu\_patients\_per\_million\_1+hosp\_patients\_per\_million\_1+new\_vaccinations\_smoothed\_per\_million\_8 ,data = PVsmall)  
aic.wge(mv\_fit3$residuals, p=0:8,q=0) #AIC picks 8,0, aic = 0.8735

## $type  
## [1] "aic"  
##   
## $value  
## [1] 1.022664  
##   
## $p  
## [1] 7  
##   
## $q  
## [1] 0  
##   
## $phi  
## [1] 0.35216207 -0.22011773 -0.10286256 -0.01584649 -0.13393431 0.07807821  
## [7] 0.36792006  
##   
## $theta  
## [1] 0  
##   
## $vara  
## [1] 2.226523

fit3 = arima(PVsmall$new\_deaths\_per\_million, order=c(8,0,0), xreg = cbind(PVsmall[,c(8,9,10,11)]) )  
fit3 #AIC aic = 263.57; time, icu\_patients\_per\_million\_1 and hosp\_patients\_per\_million\_1 looks sig,

##   
## Call:  
## arima(x = PVsmall$new\_deaths\_per\_million, order = c(8, 0, 0), xreg = cbind(PVsmall[,   
## c(8, 9, 10, 11)]))  
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ar5 ar6 ar7 ar8  
## 0.4851 -0.3189 0.0056 -0.182 -0.1394 0.1128 0.3720 -0.1546  
## s.e. 0.1288 0.1281 0.1419 0.165 0.1552 0.1488 0.1366 0.1411  
## intercept new\_cases\_per\_million\_7 icu\_patients\_per\_million\_1  
## -2.6125 0.0027 0.1264  
## s.e. 2.8790 0.0019 0.1417  
## hosp\_patients\_per\_million\_1 new\_vaccinations\_smoothed\_per\_million\_8  
## -0.0025 6e-04  
## s.e. 0.0343 5e-04  
##   
## sigma^2 estimated as 1.529: log likelihood = -119.22, aic = 266.44

acf(fit3$residuals[9:80]) # does not appear to be white noise



ltest3 = ljung.wge(fit3$residuals)

## Obs -0.03322311 0.07742046 0.05274098 0.022839 -0.05545107 -0.01306201 -0.2477912 0.07068801 -0.2125809 0.01236821 -0.1756755 -0.03174781 -0.1018219 0.1229846 -0.08584995 -0.007642846 -0.1225899 0.1357601 -0.07492321 0.05195774 0.006702245 -0.05569772 -0.1121167 0.1215022

ltest3 #FTR.There is not enough evidence to suggest that the residuals are serailly correlated. null hypothesis = white noise, alternate- not white noise. pval = 0.975. Here we FTR null hypothesis, so this is white noise

## $test  
## [1] "Ljung-Box test"  
##   
## $K  
## [1] 24  
##   
## $chi.square  
## [1] 25.56972  
##   
## $df  
## [1] 24  
##   
## $pval  
## [1] 0.3753519

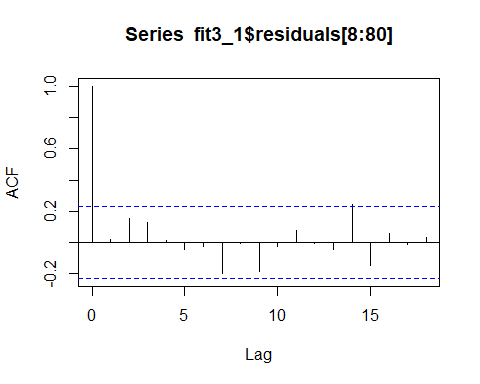
#Lagged variables - icu\_patients\_per\_million\_1 and hosp\_patients\_per\_million\_1 - Best aic so far with white noise residuals.  
mv\_fit3\_1 <- lm( new\_deaths\_per\_million~ t[1:80]+new\_cases\_per\_million+ icu\_patients\_per\_million\_1+hosp\_patients\_per\_million\_1+new\_vaccinations\_smoothed\_per\_million ,data = PVsmall)  
aic.wge(mv\_fit3\_1$residuals, p=0:8,q=0) #AIC picks 7,0, aic = 0.5299

## $type  
## [1] "aic"  
##   
## $value  
## [1] 0.5668668  
##   
## $p  
## [1] 7  
##   
## $q  
## [1] 0  
##   
## $phi  
## [1] 0.26447143 -0.15486846 0.02252074 -0.19753293 -0.01938498 0.10438991  
## [7] 0.48000247  
##   
## $theta  
## [1] 0  
##   
## $vara  
## [1] 1.439557

fit3\_1 = arima(PVsmall$new\_deaths\_per\_million, order=c(7,0,0), xreg = cbind(t[1:80],PVsmall[,c(3,9,10,7)]) )  
fit3\_1 #AIC= time, icu\_patients\_per\_million\_1 and hosp\_patients\_per\_million\_1 looks sig,

##   
## Call:  
## arima(x = PVsmall$new\_deaths\_per\_million, order = c(7, 0, 0), xreg = cbind(t[1:80],   
## PVsmall[, c(3, 9, 10, 7)]))  
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ar5 ar6 ar7 intercept  
## 0.3869 -0.2750 0.0835 -0.2183 -0.0425 0.1394 0.4185 -8.2722  
## s.e. 0.1080 0.1217 0.1174 0.1471 0.1175 0.1203 0.1156 2.9848  
## t[1:80] new\_cases\_per\_million icu\_patients\_per\_million\_1  
## 0.1486 0.0076 0.4062  
## s.e. 0.0386 0.0017 0.1486  
## hosp\_patients\_per\_million\_1 new\_vaccinations\_smoothed\_per\_million  
## -0.0621 -8e-04  
## s.e. 0.0312 4e-04  
##   
## sigma^2 estimated as 1.156: log likelihood = -119.99, aic = 267.98

acf(fit3\_1$residuals[8:80]) # appears to be white noise



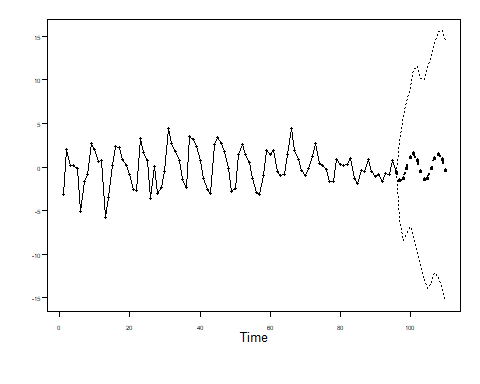
ltest3\_1 = ljung.wge(fit3\_1$residuals) #FTR. There is not enough evidence to suggest that the residuals are serailly correlated. null hypothesis = white noise, alternate- not white noise. pval = 0.5670672 Here we FTR null hypothesis, so this is white noise

## Obs 0.04050703 0.155121 0.1127794 -0.008856396 -0.04319614 -0.04597227 -0.1619064 -0.01852375 -0.1749616 -0.04760524 0.05436933 -0.03846552 -0.07485584 0.2106473 -0.1425802 0.05778509 0.007091063 0.03233046 -0.0101994 -0.04799799 -0.01637473 -0.05474732 -0.1851901 -0.07755122

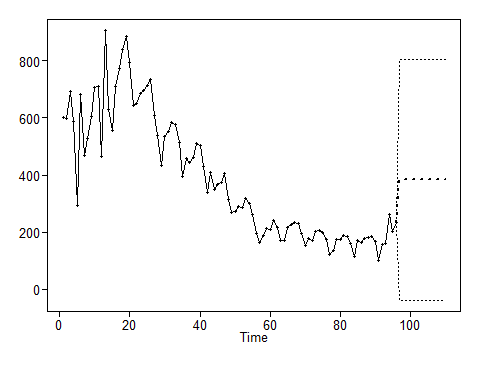
#Forecast model 2\_1  
preds\_mv3\_1 = predict(fit3\_1, newxreg = cbind(t[81:96], Post\_Vaccine$new\_cases\_per\_million[81:96], Post\_Vaccine$icu\_patients\_per\_million\_1[81:96], Post\_Vaccine$hosp\_patients\_per\_million\_1[81:96] , Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million[81:96]))  
ASE\_mv3\_1 = mean((Post\_Vaccine$new\_deaths\_per\_million[81:96] - preds\_mv3\_1$pred)^2)  
ASE\_mv3\_1 #4.948167

## [1] 4.948167

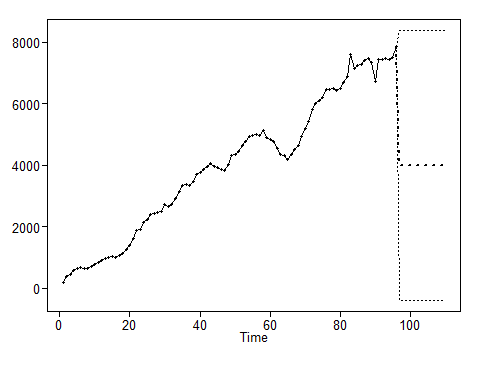
#As Model 1 (no trend, no lag) has the best ASE. We will forecast future values using this model.  
m1\_fit = lm(new\_deaths\_per\_million~new\_cases\_per\_million+new\_vaccinations\_smoothed\_per\_million+hosp\_patients\_per\_million+icu\_patients\_per\_million, data = Post\_Vaccine)  
phi = aic.wge(m1\_fit$residuals)  
m1\_resids\_14 = fore.arma.wge(m1\_fit$residuals, phi = phi$phi, n.ahead = 14)



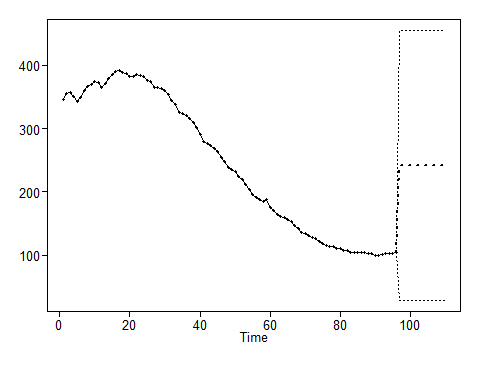
#Predicting future forecasts - 14  
Pred\_nc\_10\_m1 = fore.aruma.wge(Post\_Vaccine$new\_cases\_per\_million, n.ahead = 14)



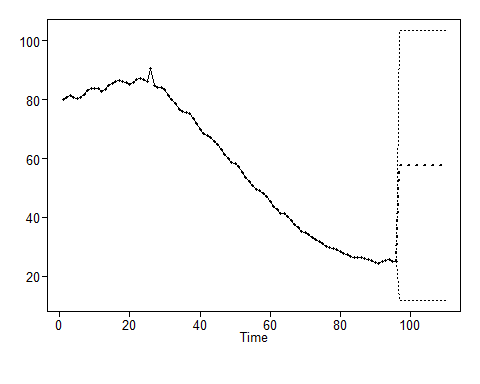
Pred\_nv\_10\_m1 = fore.aruma.wge(Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million, n.ahead = 14)



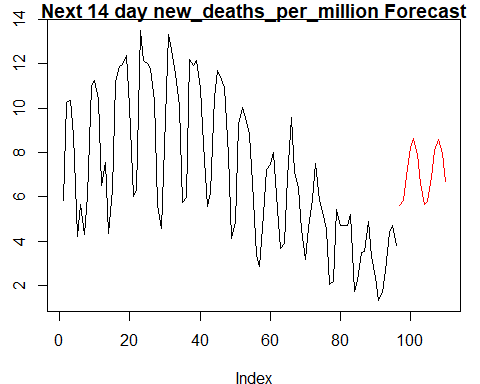
Pred\_hs\_10\_m1 = fore.aruma.wge(Post\_Vaccine$hosp\_patients\_per\_million, n.ahead = 14)



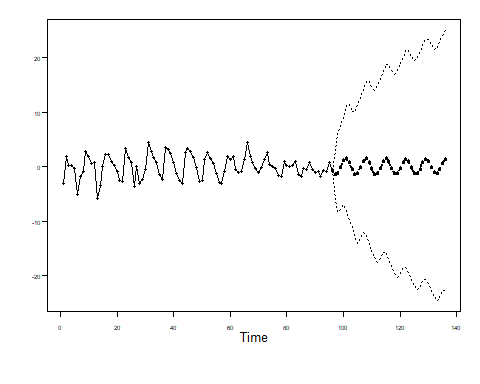
Pred\_ic\_10\_m1 = fore.aruma.wge(Post\_Vaccine$icu\_patients\_per\_million, n.ahead = 14)



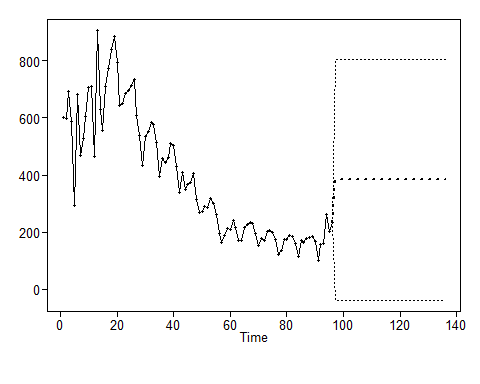
next10\_m1 = data.frame(new\_cases\_per\_million=Pred\_nc\_10\_m1$f, new\_vaccinations\_smoothed\_per\_million=Pred\_nv\_10\_m1$f,hosp\_patients\_per\_million=Pred\_hs\_10\_m1$f,icu\_patients\_per\_million=Pred\_ic\_10\_m1$f)  
  
#get predictions  
preds\_m1\_10 = predict(m1\_fit, newdata = next10\_m1)  
Preds\_m1\_10\_final = preds\_m1\_10 + m1\_resids\_14$f  
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,110), ylab = "new\_deaths\_per\_million", main = "Next 14 day new\_deaths\_per\_million Forecast")  
lines(seq(97,110,1),Preds\_m1\_10\_final, col = "red")



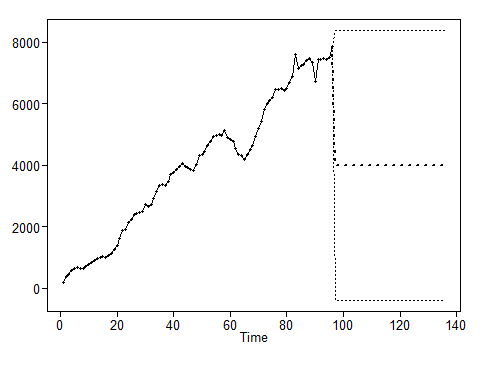
#Predicting future forecasts - 40  
m1\_resids\_40 = fore.arma.wge(m1\_fit$residuals, phi = phi$phi, n.ahead = 40)



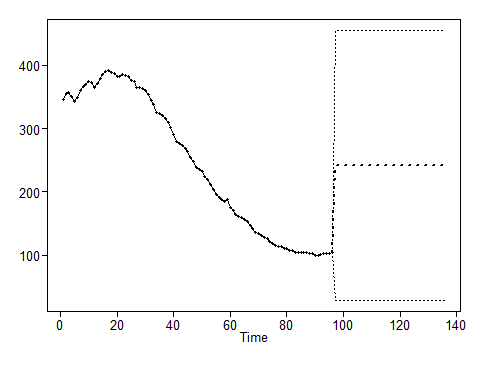
Pred\_nc\_40\_m1 = fore.aruma.wge(Post\_Vaccine$new\_cases\_per\_million, n.ahead = 40)



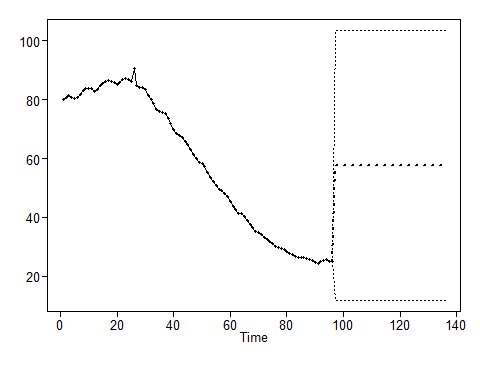
Pred\_nv\_40\_m1 = fore.aruma.wge(Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million, n.ahead = 40)



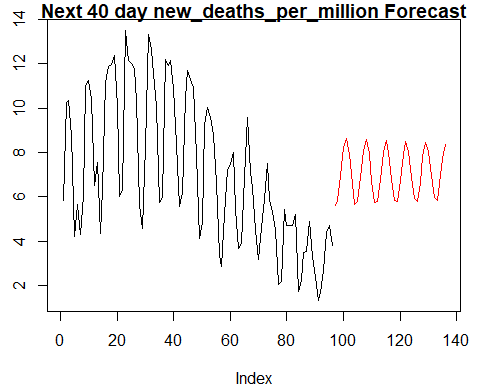
Pred\_hs\_40\_m1 = fore.aruma.wge(Post\_Vaccine$hosp\_patients\_per\_million, n.ahead = 40)



Pred\_ic\_40\_m1 = fore.aruma.wge(Post\_Vaccine$icu\_patients\_per\_million, n.ahead = 40)

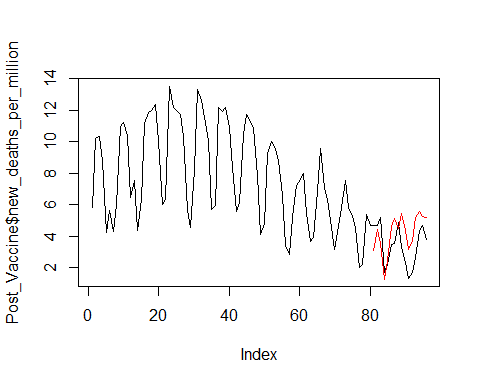


next40\_m1 = data.frame(new\_cases\_per\_million=Pred\_nc\_40\_m1$f, new\_vaccinations\_smoothed\_per\_million=Pred\_nv\_40\_m1$f,hosp\_patients\_per\_million=Pred\_hs\_40\_m1$f,icu\_patients\_per\_million=Pred\_ic\_40\_m1$f)  
  
#get predictions  
preds\_m1\_40 = predict(m1\_fit, newdata = next40\_m1)  
Preds\_m1\_40\_final = preds\_m1\_40 + m1\_resids\_40$f  
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,136), ylab = "new\_deaths\_per\_million", main = "Next 40 day new\_deaths\_per\_million Forecast")  
lines(seq(97,136,1),Preds\_m1\_40\_final, col = "red")



### VAR MODEL 1 with no lagged variables

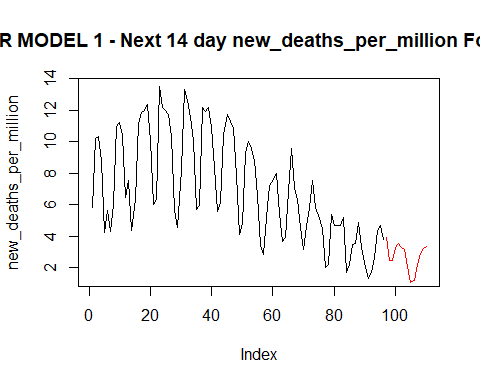
PVsmall = Post\_Vaccine[1:80,]  
VAR\_PV = VAR(cbind(PVsmall$new\_deaths\_per\_million,PVsmall$new\_cases\_per\_million,PVsmall$icu\_patients\_per\_million,PVsmall$hosp\_patients\_per\_million,PVsmall$new\_vaccinations\_smoothed\_per\_million),lag.max = 8, type = "both")  
pred = predict(VAR\_PV,n.ahead = 16)  
plot(Post\_Vaccine$new\_deaths\_per\_million, type = "l")  
lines(seq(81,96,1),pred$fcst$y1[,1],col = "red")



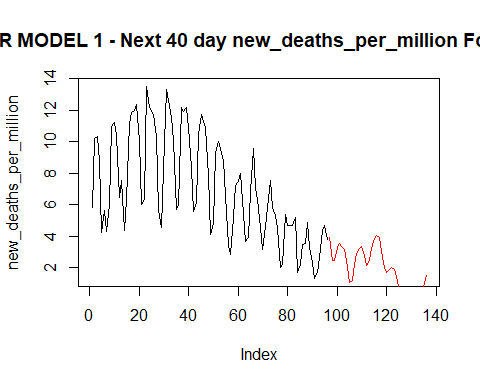
ASE = mean((Post\_Vaccine$new\_deaths\_per\_million[81:96] - pred$fcst$y1[1:16])^2)  
ASE

## [1] 2.237209

pred\_14 = predict(VAR\_PV,n.ahead = 30)  
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,110),ylab = "new\_deaths\_per\_million", main = "VAR MODEL 1 - Next 14 day new\_deaths\_per\_million Forecast")  
lines(seq(97,110),tail(pred\_14$fcst$y1[,1],14), col = "red")

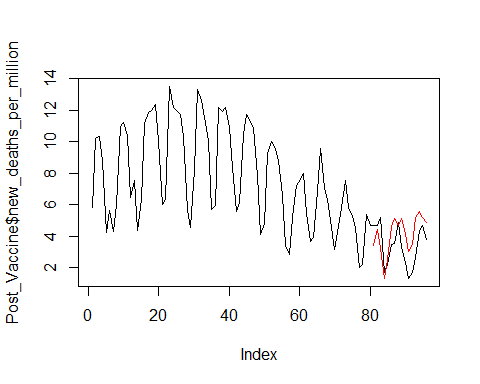


pred\_40 = predict(VAR\_PV,n.ahead = 56)  
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,136), ylab = "new\_deaths\_per\_million", main = "VAR MODEL 1 - Next 40 day new\_deaths\_per\_million Forecast")  
lines(seq(97,136,1),tail(pred\_40$fcst$y1[,1],40), col = "red")



### VAR MODEL 2 - with lagged variables (Hospital and ICU)

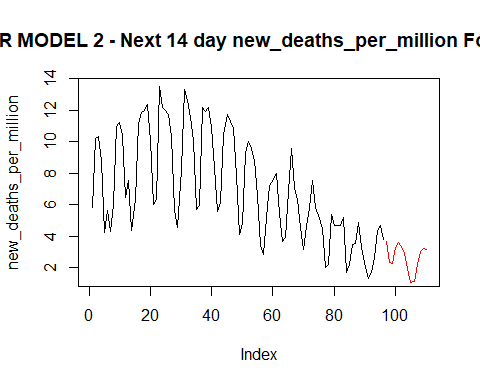
#####lagged icu\_1, hosp\_1######  
VAR\_PV = VAR(cbind(PVsmall$new\_deaths\_per\_million[2:80],PVsmall$new\_cases\_per\_million[2:80],PVsmall$icu\_patients\_per\_million\_1[2:80],PVsmall$hosp\_patients\_per\_million\_1[2:80],PVsmall$new\_vaccinations\_smoothed\_per\_million[2:80]),lag.max = 8, type = "both")  
pred = predict(VAR\_PV,n.ahead = 16)  
plot(Post\_Vaccine$new\_deaths\_per\_million, type = "l")  
lines(seq(81,96,1),pred$fcst$y1[,1],col = "red")



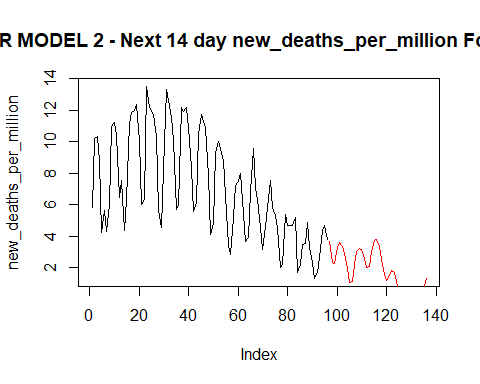
ASE = mean((Post\_Vaccine$new\_deaths\_per\_million[81:96] - pred$fcst$y1[1:16])^2)  
ASE

## [1] 1.985431

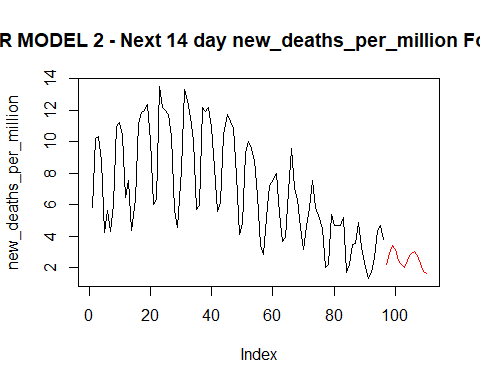
pred\_14 = predict(VAR\_PV,n.ahead = 30)  
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,110),ylab = "new\_deaths\_per\_million", main = "VAR MODEL 2 - Next 14 day new\_deaths\_per\_million Forecast")  
lines(seq(97,110,1),tail(pred\_14$fcst$y1[,1],14), col = "red")



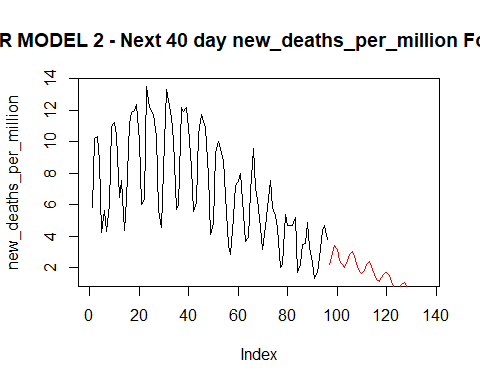
pred\_40 = predict(VAR\_PV,n.ahead = 56)  
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,136),ylab = "new\_deaths\_per\_million", main = "VAR MODEL 2 - Next 14 day new\_deaths\_per\_million Forecast")  
lines(seq(97,136,1),tail(pred\_40$fcst$y1[,1],40), col = "red")



#Model entire data for future forecasting  
VAR\_PV\_full = VAR(cbind(Post\_Vaccine$new\_deaths\_per\_million[2:96],Post\_Vaccine$new\_cases\_per\_million[2:96],Post\_Vaccine$icu\_patients\_per\_million\_1[2:96],Post\_Vaccine$hosp\_patients\_per\_million\_1[2:96],Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million[2:96]),lag.max = 8, type = "both")  
pred\_14 = predict(VAR\_PV\_full,n.ahead = 30)  
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,110),ylab = "new\_deaths\_per\_million", main = "VAR MODEL 2 - Next 14 day new\_deaths\_per\_million Forecast")  
lines(seq(97,110,1),tail(pred\_14$fcst$y1[,1],14), col = "red")



pred\_40 = predict(VAR\_PV\_full,n.ahead = 56)   
plot(as.numeric(Post\_Vaccine$new\_deaths\_per\_million), type = "l", xlim = c(1,136),ylab = "new\_deaths\_per\_million", main = "VAR MODEL 2 - Next 40 day new\_deaths\_per\_million Forecast")  
lines(seq(97,136,1),tail(pred\_40$fcst$y1[,1],40), col = "red")



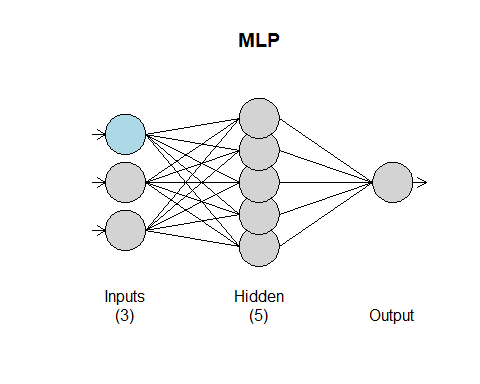
## We will proceed with Model 2 - with lagged variables ##

### MLP With Vaccine Variable Only

#Build the model  
Vacc\_Train = Post\_Vaccine[1:80,]  
Vacc\_Test = Post\_Vaccine[81:96,]  
Vacc\_xreg = data.frame(Vaccine = ts(Vacc\_Train$new\_vaccinations\_smoothed\_per\_million))  
set.seed(2)  
Vacc\_fit.mlp1 = mlp(ts(Vacc\_Train$new\_deaths\_per\_million) ,reps = 50,comb = "median", xreg = Vacc\_xreg )   
Vacc\_fit.mlp1 #mse 1.168

## MLP fit with 5 hidden nodes and 50 repetitions.  
## Univariate lags: (1,2)  
## 1 regressor included.  
## - Regressor 1 lags: (2)  
## Forecast combined using the median operator.  
## MSE: 1.1966.

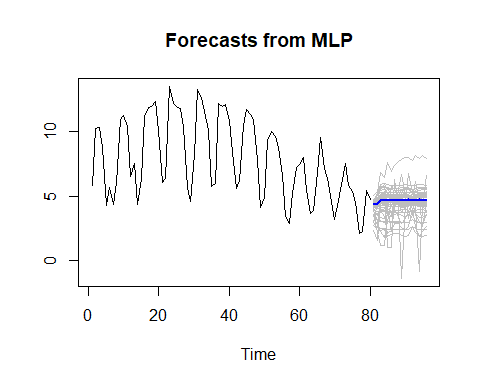
plot(Vacc\_fit.mlp1)



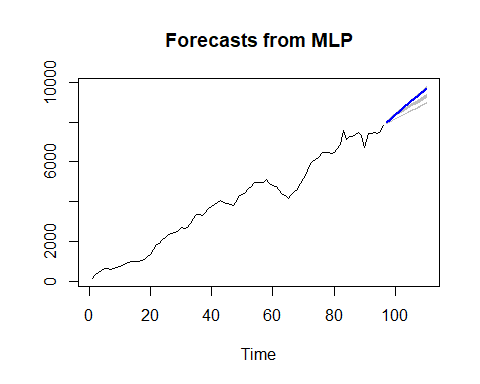
fore\_vacc\_df\_exp = data.frame(Vaccine = ts(Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million))  
Vacc\_fore.mlp1 = forecast(Vacc\_fit.mlp1, h = 16, xreg = fore\_vacc\_df\_exp)   
MLP1\_ASE = mean((Vacc\_Test$new\_deaths\_per\_million- Vacc\_fore.mlp1$mean)^2)  
MLP1\_ASE

## [1] 3.108692

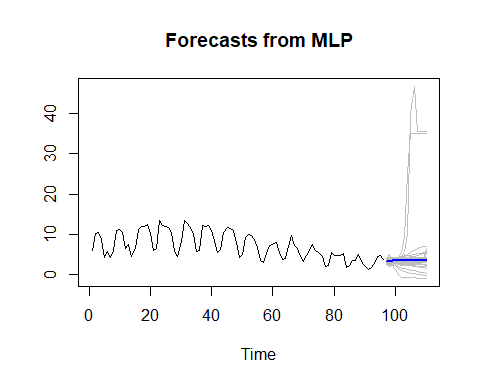
plot(Vacc\_fore.mlp1)



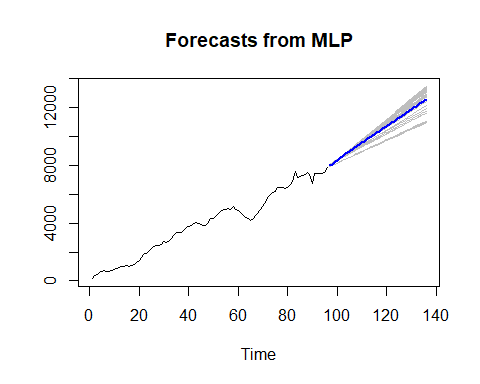
#New Vacc - 14 day forecast on this explanatory variable  
fit.mlp.NV = mlp(ts(Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million),reps = 50,comb = "median")  
fore.mlp2.14.NV = forecast(fit.mlp.NV, h = 14)  
plot(fore.mlp2.14.NV)



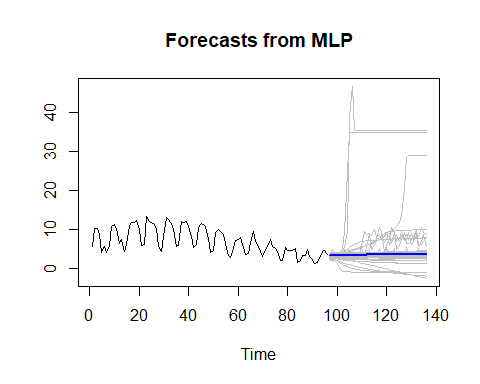
PV\_DF\_fore = data.frame(new\_vaccinations\_smoothed\_per\_million =ts(c(Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million,fore.mlp2.14.NV$mean)))  
fit.mlp = mlp(ts(Post\_Vaccine$new\_deaths\_per\_million),reps = 50,comb = "median",xreg = PV\_DF\_fore)  
fore.mlp = forecast(fit.mlp, h = 14, xreg = PV\_DF\_fore)  
plot(fore.mlp)



#New Vacc - 40 day forecast on this explanatory variable  
fit.mlp.NV = mlp(ts(Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million),reps = 50,comb = "median")  
fore.mlp2.40.NV = forecast(fit.mlp.NV, h = 40)  
plot(fore.mlp2.40.NV)

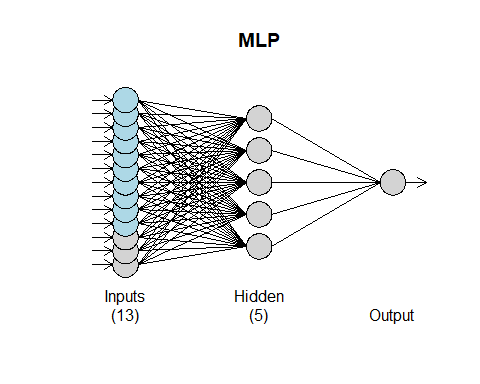


PV\_DF\_fore = data.frame(new\_vaccinations\_smoothed\_per\_million =ts(c(Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million,fore.mlp2.40.NV$mean)))  
fore.vacc\_40 = forecast(fit.mlp, h = 40, xreg = PV\_DF\_fore)  
plot(fore.vacc\_40)



### MLP With all explanatory variables

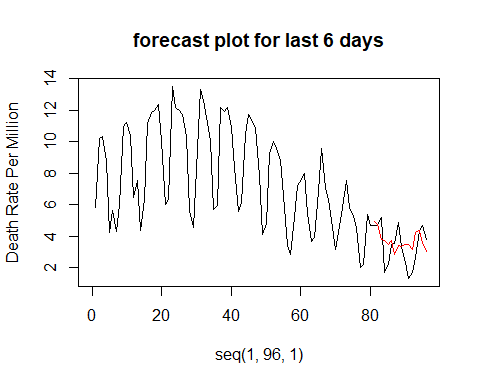
VC\_train = Post\_Vaccine[1:80,]  
VC\_test = Post\_Vaccine[81:96,]  
all\_exp\_xreg = data.frame(Vaccine = ts(t = ts(seq(1:80)),VC\_train$new\_vaccinations\_smoothed\_per\_million), NewCase = ts(VC\_train$new\_cases\_per\_million), Hosp = ts(VC\_train$hosp\_patients\_per\_million), ICU = ts(VC\_train$icu\_patients\_per\_million))  
set.seed(2)  
allexp\_fit\_mlp2 = mlp(ts(VC\_train$new\_deaths\_per\_million), reps = 50, comb = "median", xreg = all\_exp\_xreg )  
plot(allexp\_fit\_mlp2)



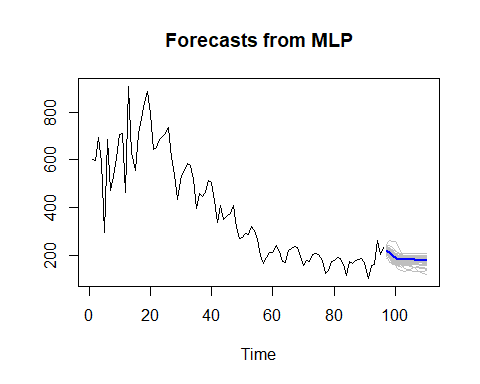
fore\_all\_exp\_df = data.frame(Vaccine = ts(Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million), NewCase = ts(Post\_Vaccine$new\_cases\_per\_million), Hosp = ts(Post\_Vaccine$hosp\_patients\_per\_million), ICU = ts(Post\_Vaccine$icu\_patients\_per\_million))  
allexp\_fore.m2 = forecast(allexp\_fit\_mlp2, h = 16, xreg = fore\_all\_exp\_df)   
MLP2\_ASE = mean((VC\_test$new\_deaths\_per\_million- allexp\_fore.m2$mean)^2)  
MLP2\_ASE #1.43

## [1] 1.427807

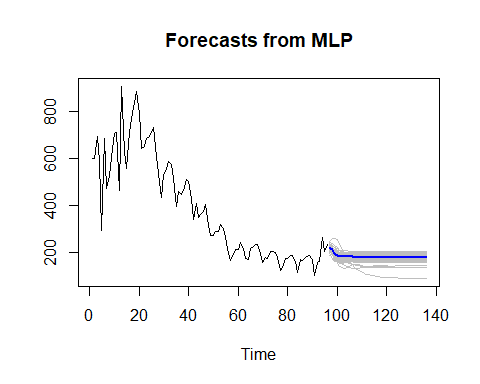
#Forecast - fitting on last 16 days  
plot(seq(1,96,1), Post\_Vaccine$new\_deaths\_per\_million, type = "l",xlim = c(0,96), ylab = "Death Rate Per Million", main = "forecast plot for last 6 days")  
lines(seq(81,96), allexp\_fore.m2$mean, type = "l", col = "red")



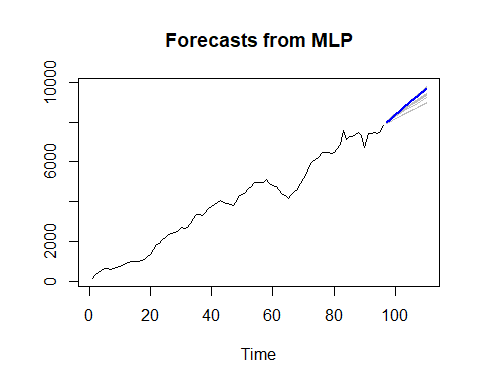
### Forecast on future short term (14 days) and long term (40 days) ###  
#New Case Forecast  
fit.mlp.NC = mlp(ts(Post\_Vaccine$new\_cases\_per\_million),reps = 50,comb = "median")  
fore.mlp2.14.NC = forecast(fit.mlp.NC, h = 14)  
fore.mlp2.40.NC = forecast(fit.mlp.NC, h = 40)  
plot(fore.mlp2.14.NC) #for short term



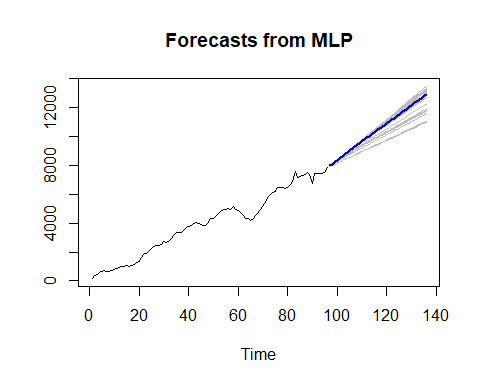
plot(fore.mlp2.40.NC) #for long term



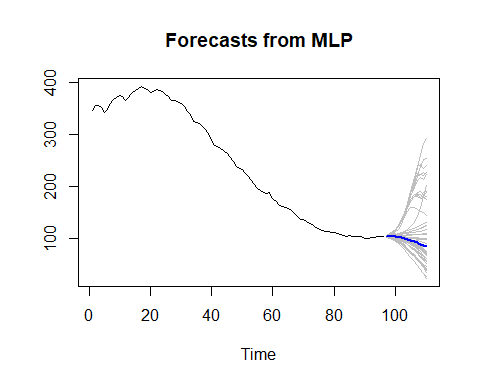
#New Vacc  
fit.mlp.NV = mlp(ts(Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million),reps = 50,comb = "median")  
fore.mlp2.14.NV = forecast(fit.mlp.NV, h = 14)  
fore.mlp2.40.NV = forecast(fit.mlp.NV, h = 40)  
plot(fore.mlp2.14.NV) #for short term



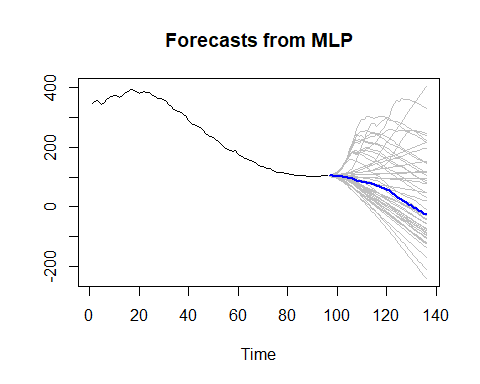
plot(fore.mlp2.40.NV) #for long term



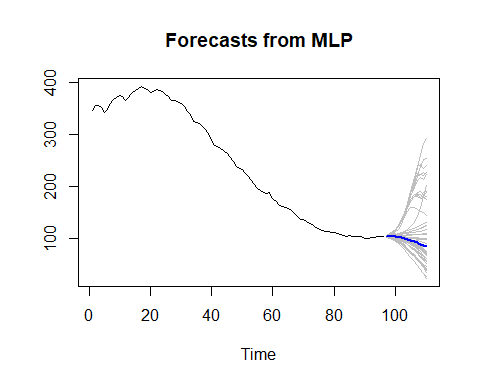
#New Hos  
fit.mlp.HS = mlp(ts(Post\_Vaccine$hosp\_patients\_per\_million),reps = 50,comb = "median")  
fore.mlp2.14.HS = forecast(fit.mlp.HS, h = 14)  
fore.mlp2.40.HS = forecast(fit.mlp.HS, h = 40)  
plot(fore.mlp2.14.HS) #for short term



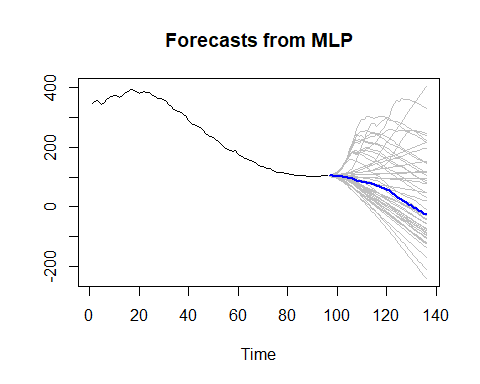
plot(fore.mlp2.40.HS) #for long term



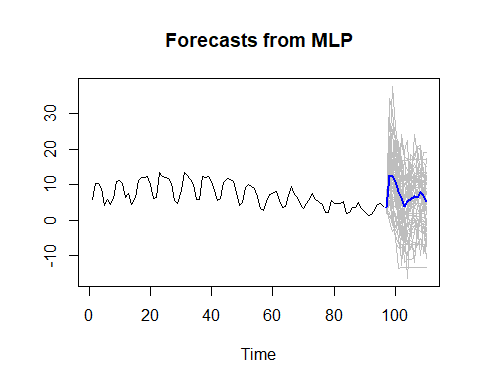
#New ICU  
fit.mlp.IC = mlp(ts(Post\_Vaccine$icu\_patients\_per\_million),reps = 50,comb = "median")  
fore.mlp2.14.IC = forecast(fit.mlp.HS, h = 14)  
fore.mlp2.40.IC = forecast(fit.mlp.HS, h = 40)  
plot(fore.mlp2.14.IC) #for short term



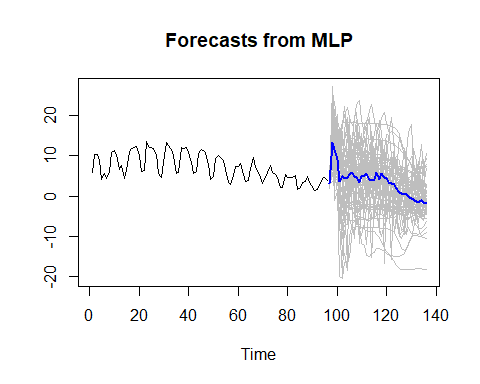
plot(fore.mlp2.40.IC) #for long term



#Forecast 14 days  
PV\_DF\_fore1 = data.frame(t = ts(seq(1:110)),new\_cases\_per\_million = ts(c(Post\_Vaccine$new\_cases\_per\_million,fore.mlp2.14.NC$mean)), icu\_patients\_per\_million = ts(c(Post\_Vaccine$icu\_patients\_per\_million,fore.mlp2.14.IC$mean)),  
 hosp\_patients\_per\_million = ts(c(Post\_Vaccine$hosp\_patients\_per\_million,fore.mlp2.14.HS$mean)), new\_vaccinations\_smoothed\_per\_million =ts(c(Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million,fore.mlp2.14.NV$mean)))  
fit.mlp2\_14 = mlp(ts(Post\_Vaccine$new\_deaths\_per\_million),reps = 50,comb = "median",xreg = PV\_DF\_fore1)  
fore.mlp2\_14 = forecast(fit.mlp2\_14, h = 14, xreg = PV\_DF\_fore1)  
plot(fore.mlp2\_14)

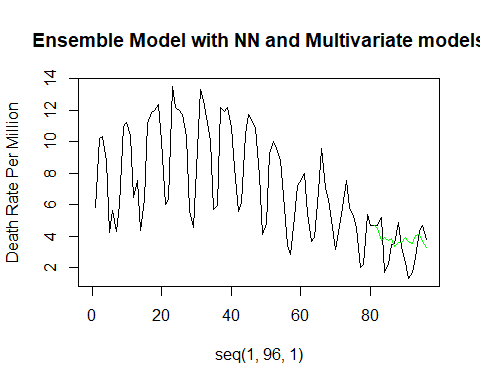


#Forecast 40 days  
PV\_DF\_fore = data.frame(t = ts(seq(1:136)),new\_cases\_per\_million = ts(c(Post\_Vaccine$new\_cases\_per\_million,fore.mlp2.40.NC$mean)), icu\_patients\_per\_million = ts(c(Post\_Vaccine$icu\_patients\_per\_million,fore.mlp2.40.IC$mean)),  
 hosp\_patients\_per\_million = ts(c(Post\_Vaccine$hosp\_patients\_per\_million,fore.mlp2.40.HS$mean)), new\_vaccinations\_smoothed\_per\_million =ts(c(Post\_Vaccine$new\_vaccinations\_smoothed\_per\_million,fore.mlp2.40.NV$mean)))  
fit.mlp2\_40 = mlp(ts(Post\_Vaccine$new\_deaths\_per\_million),reps = 50,comb = "median",xreg = PV\_DF\_fore)  
fore.mlp2\_40 = forecast(fit.mlp2\_40, h = 40, xreg = PV\_DF\_fore)  
plot(fore.mlp2\_40)



### Ensemble Model with NN (all variables) Model and VAR 2

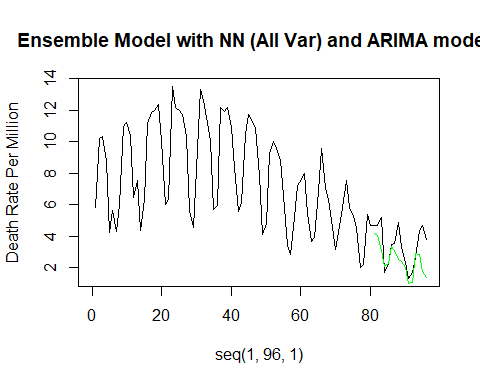
#preds\_mv0$pred : multivariate with only vaccine  
#allexp\_fore.m2$mean: mlp with all variables  
ensemble = (preds\_mv0$pred + allexp\_fore.m2$mean)/2  
  
#Plot  
plot(seq(1,96,1), Post\_Vaccine$new\_deaths\_per\_million, type = "l",xlim = c(0,96), ylab = "Death Rate Per Million", main = "Ensemble Model with NN and Multivariate models")  
lines(seq(81,96,1), ensemble, type = "l", col = "green")



ASE\_en = mean((Post\_Vaccine$new\_deaths\_per\_million[81:96] - ensemble)^2)  
ASE\_en #1.590614

## [1] 1.590614

#for\_aruma2\_s7d1\_16: Univariate 16 last n  
#allexp\_fore.m2$mean: mlp with all variables  
  
#Univariate Model 3 and MLP with all variables  
ensemble2 = (for\_aruma2\_s7d1\_16$f + allexp\_fore.m2$mean)/2  
  
#Plot  
plot(seq(1,96,1), Post\_Vaccine$new\_deaths\_per\_million, type = "l",xlim = c(0,96), ylab = "Death Rate Per Million", main = "Ensemble Model with NN (All Var) and ARIMA models")  
lines(seq(81,96,1), ensemble2, type = "l", col = "green")



ASE\_en2 = mean((Post\_Vaccine$new\_deaths\_per\_million[81:96] - ensemble2)^2)  
ASE\_en2 #1.836393

## [1] 1.836393