

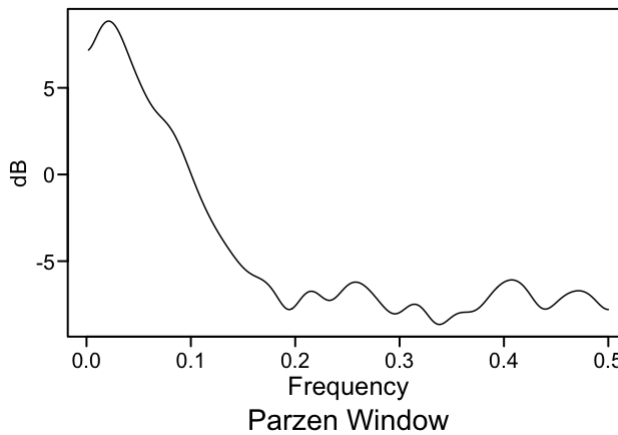
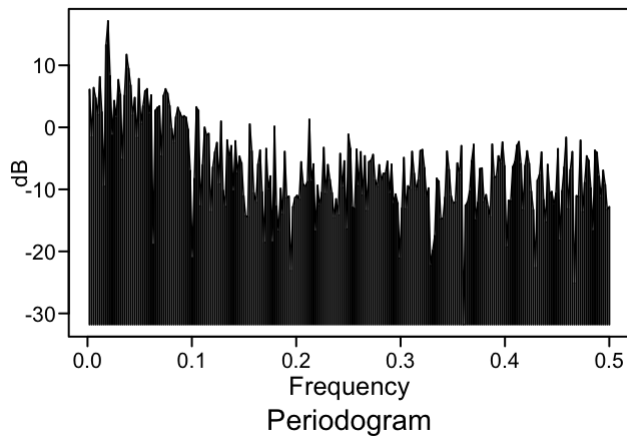
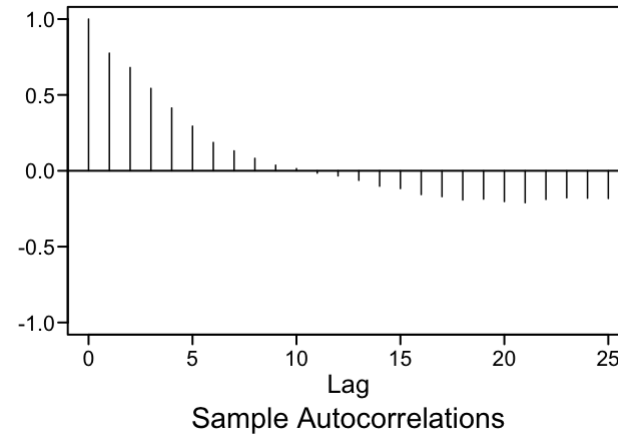
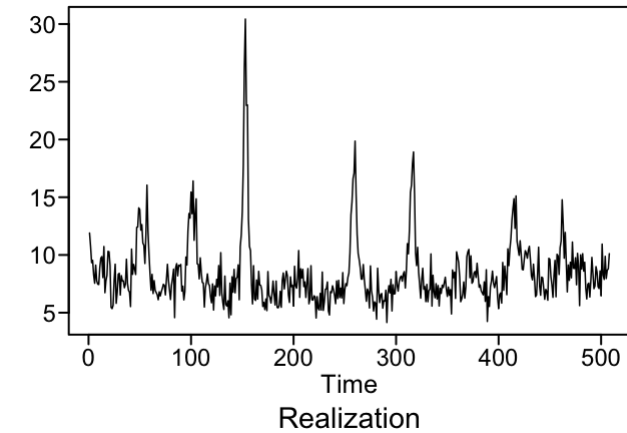
MSDS 6373 Time Series Section 403 Final Project - Tej Tenmattam

8/16/2019

The following data set is a portion of the data taken from a study (Shumway 1988) on the possible effects of pollution and temperature on weekly Respiratory mortality in Los Angeles County. The dataset is called “lap” and is available from the *astsa* package. The goal of this project is to forecast 5 weeks of respiratory mortality beyond the 508 observations we were given.

1. (5 pts) Plot the respiratory mortality data you have.

```
# Import "lap" data
data(lap)
# Convert to a time series data frame called RM
RM=data.frame(date=time(lap),Time=as.factor(seq(1,508,1)),as.matrix(lap))
# plot realization in RM along with sample autocorrelations, periodogram, and Parzen window-based spectral estimator
plotts.sample.wge(RM$rmort)
```



2. (5 pts) Comment on it's stationarity or nonstationarity.

```

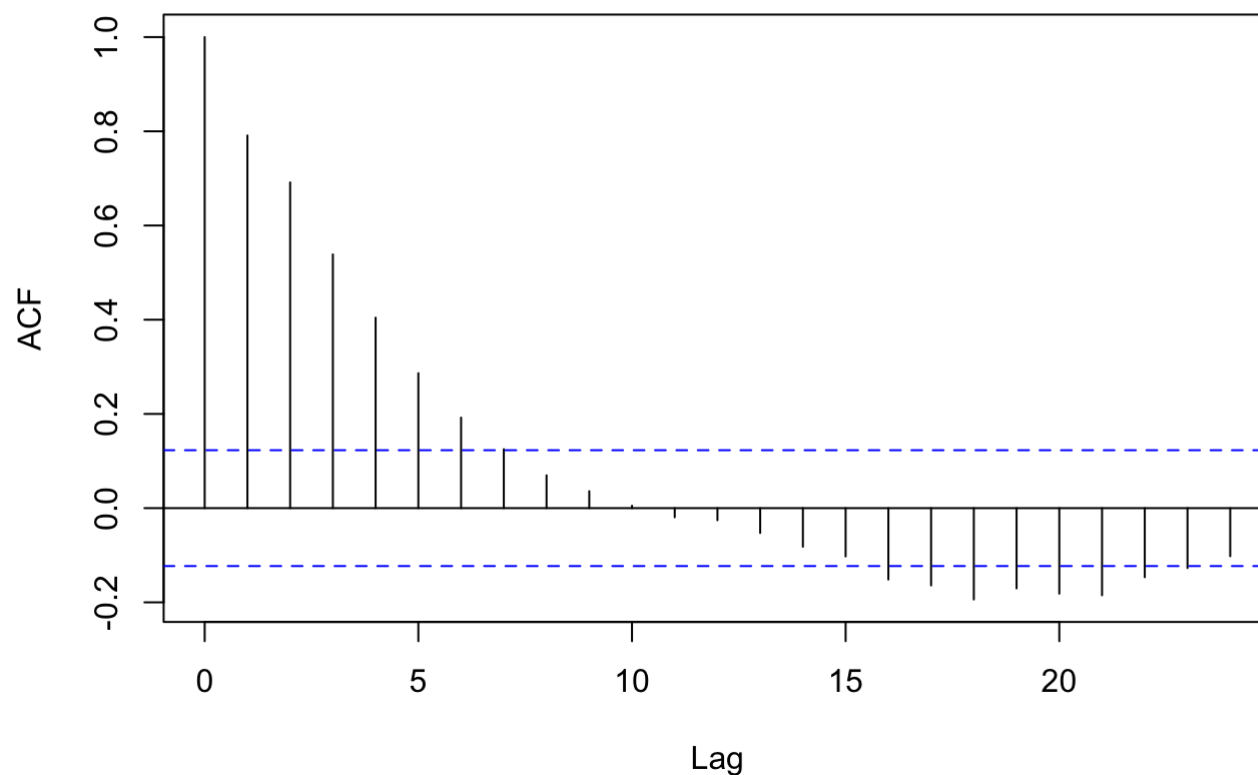
# Checking for stationarity using plots:
# Condition 1: Subpopulations of RM$mort for a given time have constant mean for all t.
#  $E[X_t] = \mu$ 
# The realization is pseudo-cyclic. The condition that mean does not depend on time is not met visually.

# Condition 2: Subpopulations of X for a given time have a constant and finite variance for all t.
#  $\text{Var}[X_t] = \sigma^2 < \infty$ 
# Variance also does not appear constant because of the pseudo-cyclic behavior and spikes.

# Condition 3: The correlation of  $X_{(t_1)}$  and  $X_{(t_2)}$  depends only on  $t_2 - t_1$ . That is, the covariance between data points is dependent only on how far apart they are, not where they are.
acf(RM$mort[1:254])

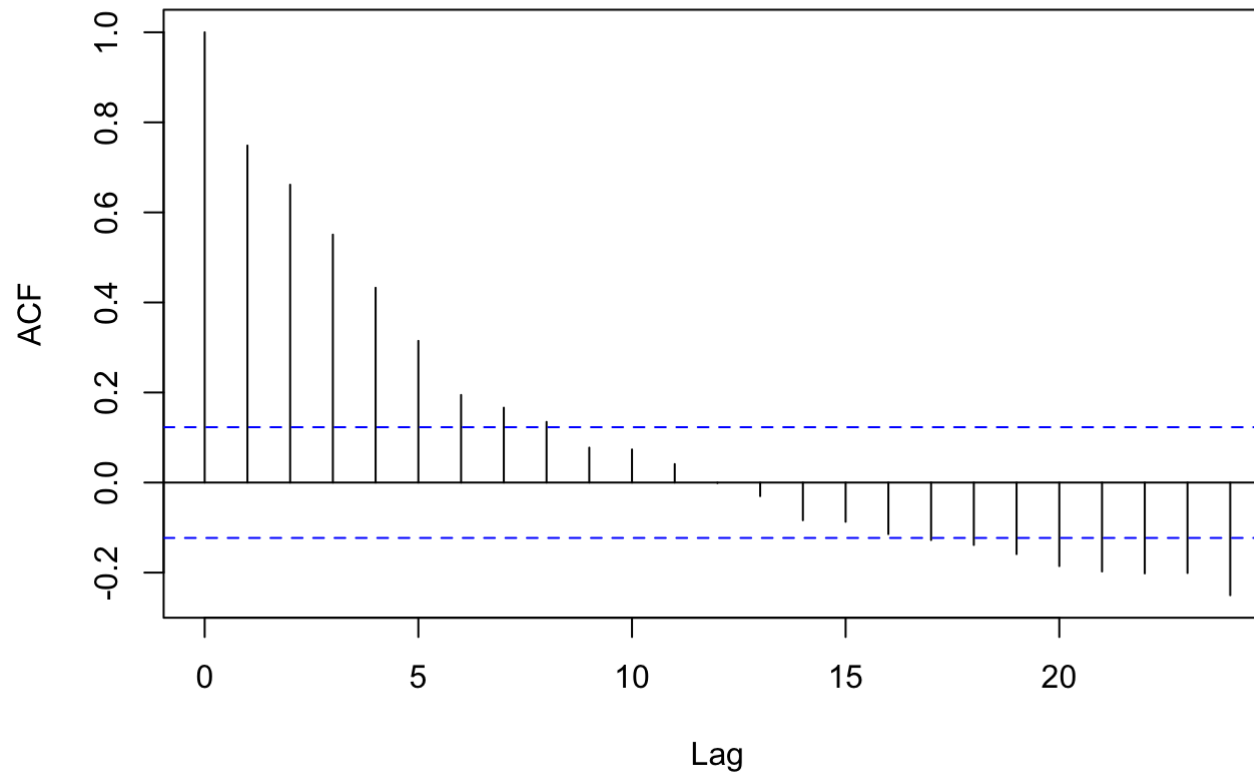
```

Series RM\$mort[1:254]



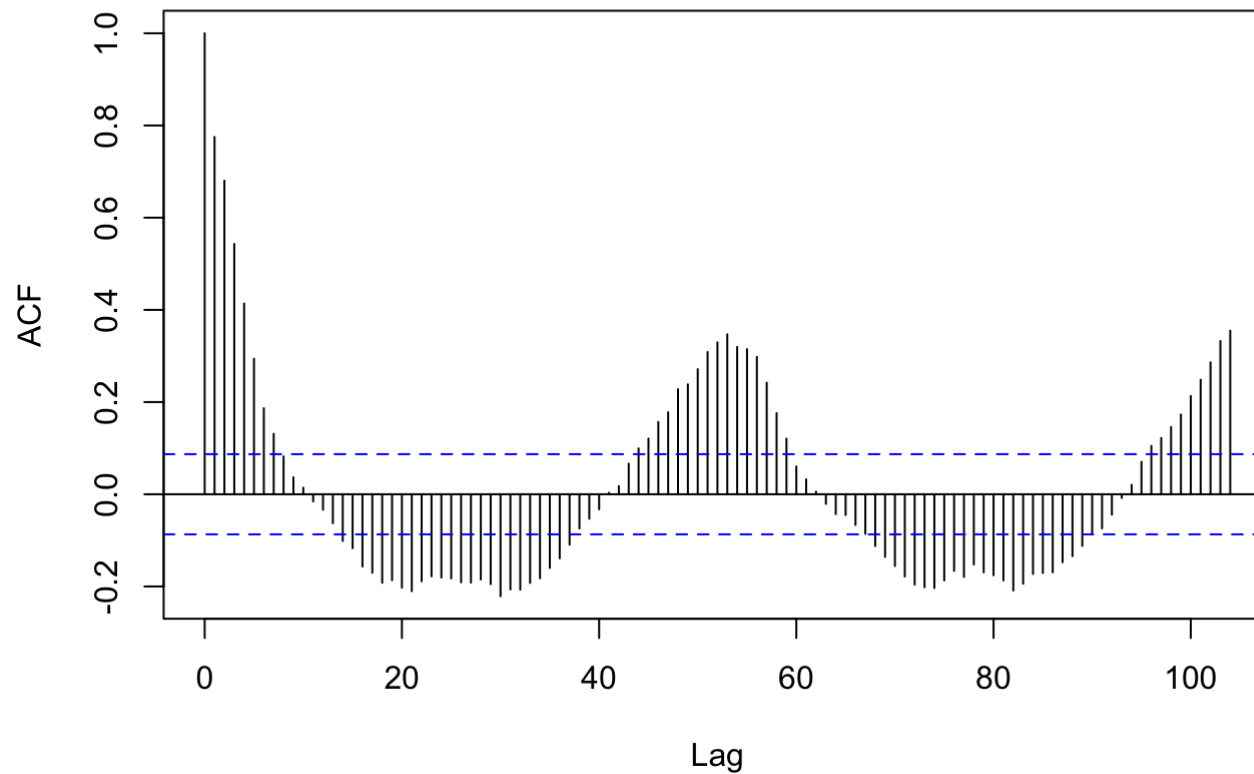
```
acf(RM$rmort[255:508])
```

Series RM\$rmort[255:508]



```
# This condition has been met as the lags do not appear to be dependent on time as shown by the ACFs  
acf(RM$rmort, lag.max = 104)
```

Series RM\$mort



Autocorrelations have a damped sinusoidal behavior with cycle length about 52, which is consistent with $f_0 = 0.019$. Spectral density has a peak at about $1/52=0.019$

To verify a season of 52, I have also checked the factor table.
`est.ar.wge(RM$mort,p=52,type='burg')`

```
##
## Coefficients of Original polynomial:
## 0.5957 0.2794 -0.0175 -0.0575 -0.0660 -0.0922 0.0714 0.0498 -0.0604 -0.0020 -0.0019 0.0262 0.0203 -0.0627 0.02
00 -0.0573 0.0258 -0.0174 0.0621 -0.0616 -0.0702 0.0770 0.0464 -0.0517 -0.0151 -0.0401 -0.0043 0.0901 0.0044 -0.1
470 0.0372 0.0112 0.0199 -0.0117 -0.0148 -0.0528 0.0359 0.0587 -0.0199 -0.0523 0.0104 -0.0622 0.0887 0.0491 -0.05
64 -0.0060 -0.0329 0.0888 -0.0191 -0.0287 0.0392 0.0631
##
## Factor                Roots                Abs Recip      System Freq
## 1-1.9685B+0.9828B^2    1.0015+-0.1204i    0.9914         0.0190
## 1-1.9055B+0.9620B^2    0.9904+-0.2422i    0.9808         0.0382
## 1-0.4788B+0.9548B^2    0.2508+-0.9922i    0.9771         0.2106
## 1+0.0121B+0.9448B^2    -0.0064+-1.0288i    0.9720         0.2510
## 1-1.7270B+0.9405B^2    0.9182+-0.4693i    0.9698         0.0752
## 1-1.8201B+0.9390B^2    0.9691+-0.3545i    0.9690         0.0558
## 1+1.6678B+0.9383B^2    -0.8887+-0.5253i    0.9686         0.4150
## 1-0.9682B              1.0329             0.9682         0.0000
## 1+1.5132B+0.9366B^2    -0.8078+-0.6443i    0.9678         0.3928
## 1+0.2458B+0.9304B^2    -0.1321+-1.0283i    0.9646         0.2703
## 1+1.9124B+0.9275B^2    -1.0309+-0.1237i    0.9631         0.4810
## 1+1.1263B+0.9270B^2    -0.6075+-0.8424i    0.9628         0.3494
## 1+1.8628B+0.9267B^2    -1.0050+-0.2626i    0.9627         0.4593
## 1+0.7655B+0.9241B^2    -0.4142+-0.9542i    0.9613         0.3152
## 1-1.4713B+0.9228B^2    0.7972+-0.6694i    0.9606         0.1112
## 1-1.2881B+0.9215B^2    0.6989+-0.7725i    0.9600         0.1330
## 1-0.8042B+0.9207B^2    0.4367+-0.9462i    0.9595         0.1812
## 1-1.6057B+0.9205B^2    0.8722+-0.5707i    0.9594         0.0922
## 1-0.2480B+0.9177B^2    0.1351+-1.0351i    0.9580         0.2293
## 1+0.4734B+0.9130B^2    -0.2593+-1.0140i    0.9555         0.2898
## 1+1.3050B+0.9123B^2    -0.7153+-0.7646i    0.9551         0.3697
## 1+1.7527B+0.8974B^2    -0.9765+-0.4009i    0.9473         0.4380
## 1-1.0703B+0.8953B^2    0.5977+-0.8716i    0.9462         0.1543
## 1+0.9243B              -1.0819            0.9243         0.5000
## 1+0.8168B+0.8295B^2    -0.4924+-0.9814i    0.9108         0.3240
## 1-0.8595B+0.7921B^2    0.5425+-0.9839i    0.8900         0.1698
## 1+1.2414B+0.5350B^2    -1.1603+-0.7232i    0.7314         0.4113
##
##
```

\$phi

```
## [1] 0.595711769 0.279372966 -0.017468086 -0.057467379 -0.065969228
## [6] -0.092166860 0.071427890 0.049781231 -0.060440649 -0.002037535
## [11] -0.001862669 0.026161324 0.020314095 -0.062684817 0.019991124
## [16] -0.057282475 0.025781147 -0.017408654 0.062089145 -0.061643115
## [21] -0.070240762 0.077021479 0.046448404 -0.051652242 -0.015094689
## [26] -0.040070989 -0.004251436 0.090093807 0.004356688 -0.147024891
## [31] 0.037162036 0.011243768 0.019932973 -0.011668413 -0.014839260
## [36] -0.052849519 0.035939440 0.058682332 -0.019913593 -0.052335710
## [41] 0.010391611 -0.062241825 0.088685982 0.049118040 -0.056406236
## [46] -0.005966067 -0.032915314 0.088804522 -0.019069564 -0.028732884
## [51] 0.039214757 0.063143407
```

##

\$res

```
## [1] 0.000000000 -0.147183571 -1.038249386 0.214478805 -0.732019077
## [6] -0.713274185 1.563618954 -0.709956414 -0.891615051 -0.210212554
## [11] 1.256559315 1.369484793 0.668097647 -1.652076343 2.058590864
## [16] -2.548835833 0.255725626 1.160688709 2.236149589 0.649851726
## [21] -0.613398240 -3.696656860 -1.224322002 0.350534362 2.298890526
## [26] 1.903669180 -3.042547250 -0.498936743 1.734949434 0.430354846
## [31] -1.269300746 0.828600457 -0.084394062 -0.179655908 -0.278098089
## [36] -0.061891174 1.933806475 -1.560437427 -1.294752971 -0.794788959
## [41] -1.441584615 4.099113833 -0.879181072 -0.459981024 -0.400349033
## [46] -0.463610647 3.617807139 1.125525271 1.853862850 1.033104009
## [51] -0.733459906 0.159625999 -0.239073776 0.224180121 -0.682196015
## [56] 2.831616458 5.353904380 -1.422382510 -2.581831735 -1.232821178
## [61] -1.295066358 1.994171849 -0.545476018 -0.073781739 -0.843663422
## [66] -0.234740939 -0.438096950 -0.102764480 -0.313810645 -0.705415005
## [71] -0.140080884 0.306515230 0.515396145 0.196586137 -1.075192090
## [76] 0.514297826 1.976085305 -0.138586211 -0.990075716 -1.777453710
## [81] 0.357803030 1.681462277 1.760301225 -3.715828751 1.328732194
## [86] 1.360797425 0.154923110 0.625746234 0.551898729 -0.562028458
## [91] -1.969687342 0.351709767 -1.356963139 0.082087325 2.013806744
## [96] -0.293474695 2.650696873 2.401067847 0.608084964 1.268775152
## [101] -0.781989786 3.105570222 -2.227056586 1.924320582 1.916812227
## [106] -3.141301098 -0.569081984 -1.045821091 1.366544665 -0.814874958
## [111] -0.571375308 -0.413700179 -1.684137721 0.574501390 1.373250466
## [116] 1.459432768 -1.211906571 0.207075458 -0.796413637 -0.974227109
## [121] 1.551896924 -1.463878664 1.046563024 -1.021596937 1.440676223
## [126] -0.494141262 1.520016674 -1.980878759 2.579032339 -1.598199755
```

```
## [131] -1.633238923 -0.191186326 1.474491609 -1.618900415 -0.257192718
## [136] -0.214274427 -2.108136546 1.340762281 -1.870234795 1.332777100
## [141] -0.721841603 0.983813963 -2.520891913 0.528100033 0.279353932
## [146] 0.470990348 -2.427319087 1.848374975 1.879648946 2.107877910
## [151] 5.030335495 9.506491779 7.589963995 -2.766100817 0.422441136
## [156] -6.272389205 -2.126953610 2.319546373 -0.006175954 -2.139492242
## [161] 2.654092532 -1.768713845 -0.986386143 1.354888071 -1.486542957
## [166] 1.961904515 -0.037522739 -0.061710944 0.600404723 -1.095792734
## [171] -0.364952064 -0.419677140 0.854896513 -0.880389969 2.600418426
## [176] -1.365125887 -1.994636264 0.043448876 1.171579115 1.020424445
## [181] -1.750065747 0.056114087 2.044598928 -1.152428508 0.178585923
## [186] -0.876821958 1.347706948 -0.789839146 1.609185915 0.154628015
## [191] -1.082615184 -0.347474462 1.159343875 -0.064516588 -1.620072675
## [196] -2.663303513 0.913707001 -0.986091452 -0.665328931 -2.657986845
## [201] 0.983094824 -2.149289240 -2.143183613 -1.703844527 0.515494543
## [206] -4.873310592 0.068526149 -0.541410963 -0.406821175 -1.126888805
## [211] 0.117438937 -2.412500860 0.951574484 1.442361698 -2.600050561
## [216] -1.360072262 2.673468304 -2.597739630 -0.852629293 -0.228245584
## [221] -0.744288524 -1.867080281 1.219287429 -1.527511680 -0.624779715
## [226] 0.470572749 -1.951371852 2.189094112 -2.354784247 -1.190813255
## [231] 0.810550571 -0.120956827 -0.052467839 -2.101937682 -1.558997040
## [236] 0.269604677 0.744761455 0.113027058 -0.575536945 -1.539645828
## [241] -1.419944949 0.787156287 -1.112109782 1.172041997 -2.569377401
## [246] -0.351702515 -0.437384629 1.250212899 -2.521056090 -0.268557968
## [251] 0.027104762 -0.520544933 0.373481869 0.388874726 0.778726424
## [256] 3.623468179 2.330515549 3.418683773 1.971220869 4.165638638
## [261] -2.183685291 -2.209824845 -0.528704736 0.654915153 -0.003682021
## [266] -0.097549590 -0.090741358 1.606301949 -0.824976389 -1.164133280
## [271] -1.670880807 1.066586767 0.624289847 -0.847619227 2.863419679
## [276] -1.194857755 -1.072852673 -0.614700271 -0.389370683 -0.260819673
## [281] -1.265329961 3.335556308 -1.113765166 -1.631608495 -0.034196899
## [286] 0.605890867 -1.509124461 0.127417913 0.172703876 0.908604265
## [291] -2.528274412 0.939396923 1.321335241 -1.056429793 0.620265411
## [296] -2.524153667 0.502620140 -0.279961383 -0.416941136 1.544489951
## [301] -1.185287207 -1.346372997 -0.095514004 -0.508928072 -0.400187509
## [306] -0.506676137 -0.590147863 -1.066052287 -1.883750891 -0.076512905
## [311] 1.390640435 -1.228040396 1.183144872 3.975580494 2.354019860
## [316] 2.930306713 3.114222532 -1.416851292 -4.119952054 -0.972030003
## [321] 2.640809837 0.286689399 -1.339223584 -0.991647912 1.207322947
## [326] 1.189679133 -1.069438639 -0.608243712 -1.226919052 1.179664610
## [331] -0.298219649 1.139483843 -0.229055783 3.072928651 -2.910186335
```



```
## [336] 0.833902535 -0.170412273 -1.114941449 1.018519978 -0.875282437
## [341] 0.710261718 -0.821533125 0.787167893 -0.278973094 0.102509730
## [346] 0.322874447 1.197389695 0.621723445 -1.772107671 -0.523360567
## [351] 0.499083305 -0.371264997 -0.075449259 -2.640410673 2.274001932
## [356] 0.135267867 0.608132634 -2.736955051 2.104975937 0.360197148
## [361] -0.604270475 -1.704140068 -2.887161955 -1.053510046 -0.661020426
## [366] 0.437430726 -2.345830864 -1.112790556 1.124336971 0.427701982
## [371] 0.274060582 -2.713271255 1.122458203 -1.981869224 0.288426723
## [376] 1.257071606 0.177088310 -1.081528083 1.099156428 -0.541146020
## [381] -1.045247763 -0.460825687 -0.395026246 -1.119563065 0.026847463
## [386] 1.175724642 -1.490835860 1.021305632 -3.655134836 0.529241550
## [391] -0.190248113 0.863828052 0.277732120 0.559400914 -0.905258713
## [396] -0.805564136 0.325530636 0.074898394 -1.881957168 -1.542102161
## [401] 0.671297662 0.495196912 -1.567871693 2.592339206 -2.111692009
## [406] 1.628738977 -0.944399085 -2.035034374 3.241633740 -0.041533639
## [411] 0.924035928 1.570966373 2.015615012 1.581968927 2.681581034
## [416] -1.373108651 2.563455526 -0.634862718 -0.794202163 0.511118483
## [421] 1.281518308 0.075023734 -1.093679375 -0.094532844 0.902180627
## [426] 1.305497551 0.819089008 0.146792711 0.623019822 1.444499518
## [431] -0.259934867 -1.420963984 0.135743793 1.722394190 0.162268466
## [436] -1.753710392 -0.369041986 1.777255470 3.392448321 -2.251902003
## [441] -1.566291039 1.104947178 1.192940809 -0.203874235 -0.932645067
## [446] 0.363001862 0.274609343 -0.545013695 3.081210542 0.611914531
## [451] -0.510550073 -1.364073700 -1.234578468 1.578578150 2.569708578
## [456] -2.311801526 1.894672169 0.386355866 -0.975709621 2.275742649
## [461] 0.912544274 3.284738330 -0.090903918 -2.588986453 0.965678068
## [466] -1.405715691 -3.127713358 1.608337336 -0.607326581 0.738858475
## [471] -0.623017922 1.974337382 -1.767531896 -0.533265048 1.416970858
## [476] -1.444564998 1.141086442 -0.015567648 -2.939266946 2.993142629
## [481] 0.551346606 0.394055322 -1.144159832 0.870852466 -2.179767518
## [486] -0.948538534 0.672936699 0.772846554 -0.775490152 0.460684998
## [491] 2.360969497 -0.241494775 -1.757918622 -0.502902000 1.697652560
## [496] 0.994400992 -1.177966972 -0.365561964 1.068565865 -1.855562820
## [501] 3.732736238 -0.561779268 -2.119708259 2.439632136 -1.058145200
## [506] -0.845020820 1.440813281 1.602628415
##
## $avar
## [1] 2.729843
##
## $aic
## [1] 1.212906
```

```
##  
## $aicc  
## [1] 2.242655  
##  
## $bic  
## [1] 1.654275
```

```
# Factor table also confirms a system frequency at 0.0190 with a complex conjugate root of "1.0015+-0.1204i" with  
absolute reciprocal of 0.9914.  
# Another test for stationarity is Dickey Fuller test where H0: model has a root of +1 and HA: the model does not  
have a root of +1  
adf.test(RM$rmort)
```

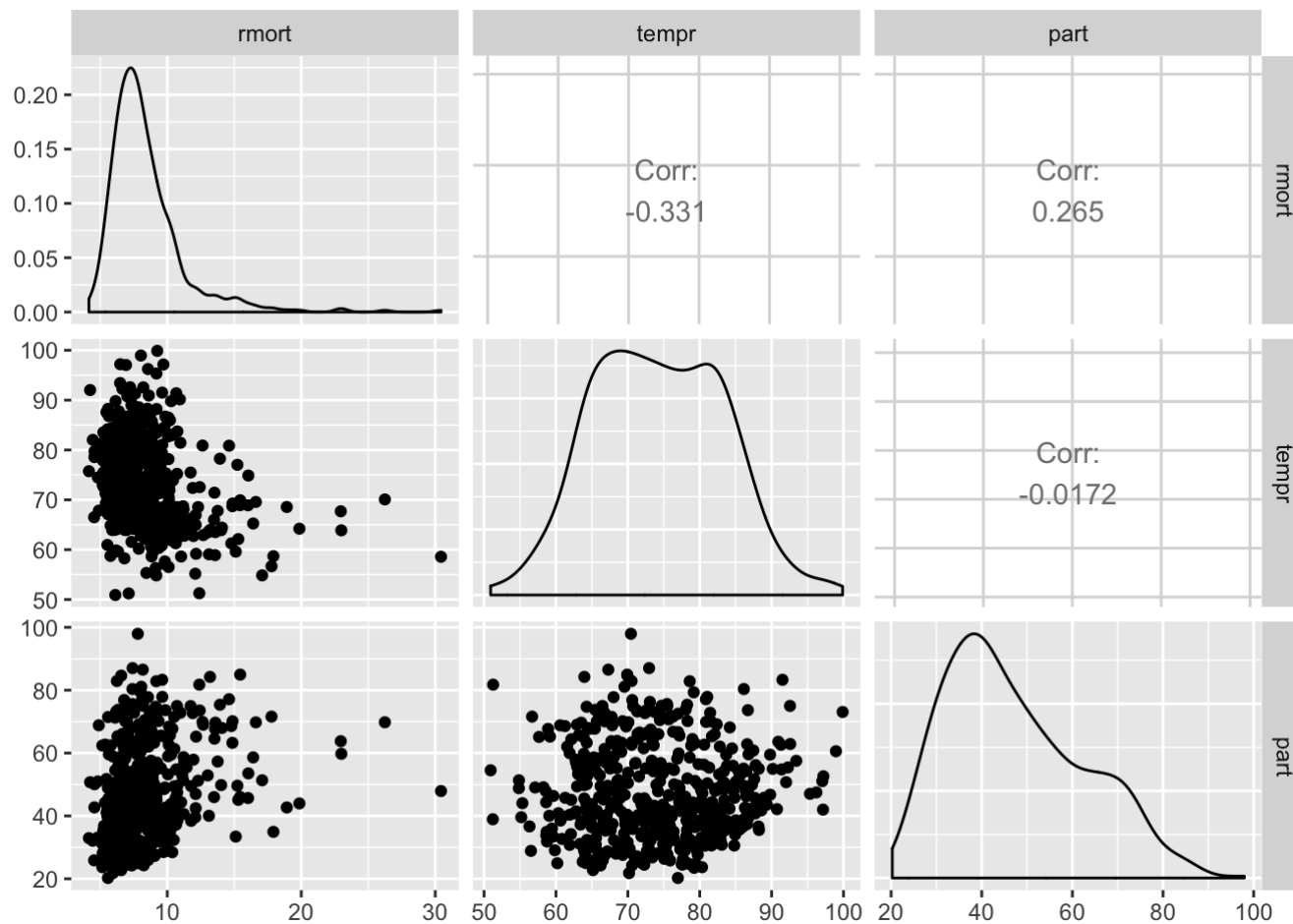
```
## Warning in adf.test(RM$rmort): p-value smaller than printed p-value
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: RM$rmort  
## Dickey-Fuller = -5.8365, Lag order = 7, p-value = 0.01  
## alternative hypothesis: stationary
```

```
# The Dickey-Fuller test of H_0: the model has a unit root, is rejected (p-value=0.01)  
# As per all the above evidence, I can say that the data is non-stationary. This is weekly data with a season of  
52.
```

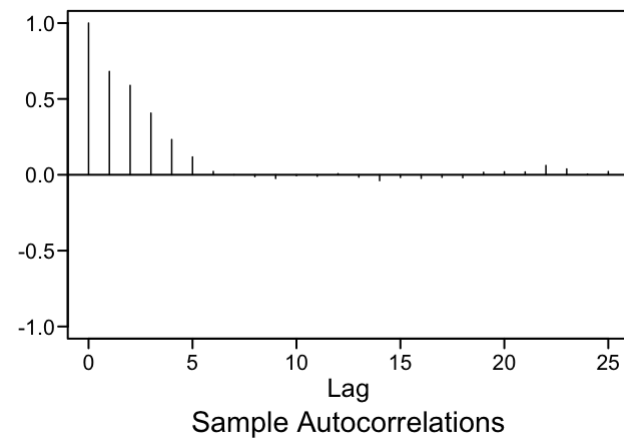
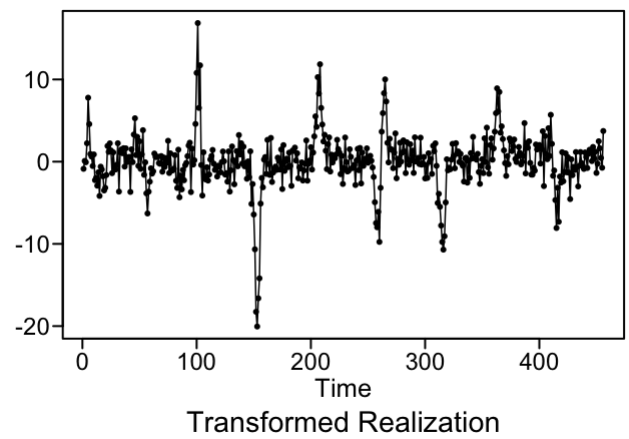
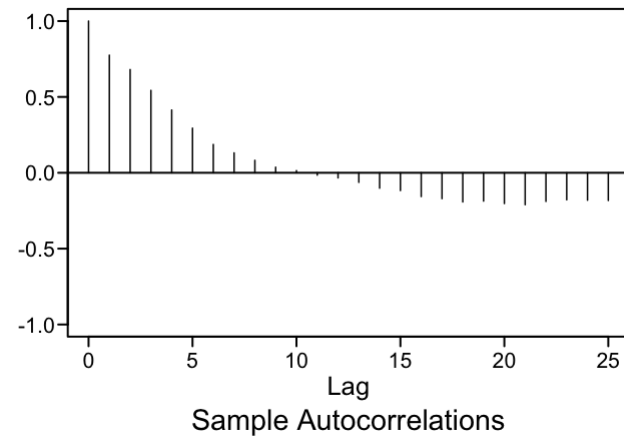
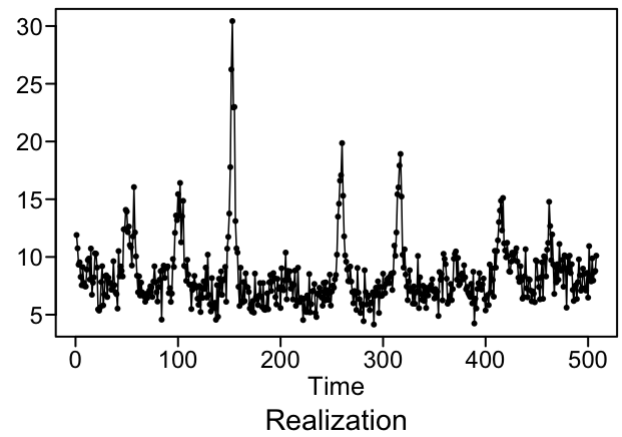
3a. (10 pts) Perform a univariate analysis using a) AR, ARMA, ARIMA, and/or ARUMA models (at least one). Also, clearly explain how you arrived at your final model.

```
# EDA:
# I have used ggpairs plot to check the correlation between the variables
RM2=RM[c(-1,-2,-3,-5,-7,-8,-9,-10,-11,-12)]
ggpairs(RM2)
```

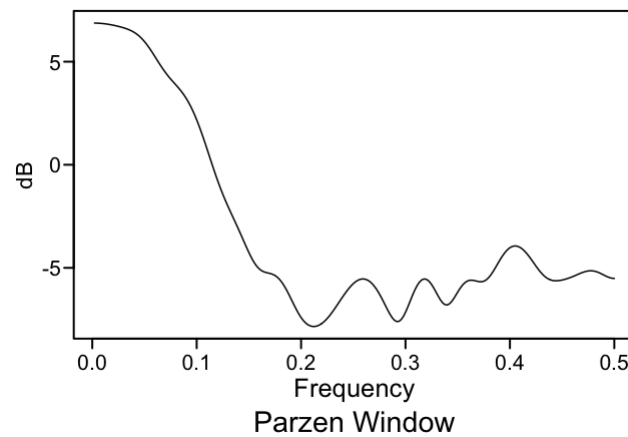
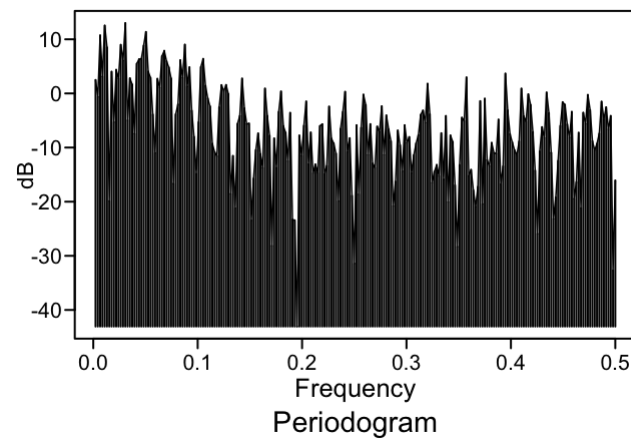
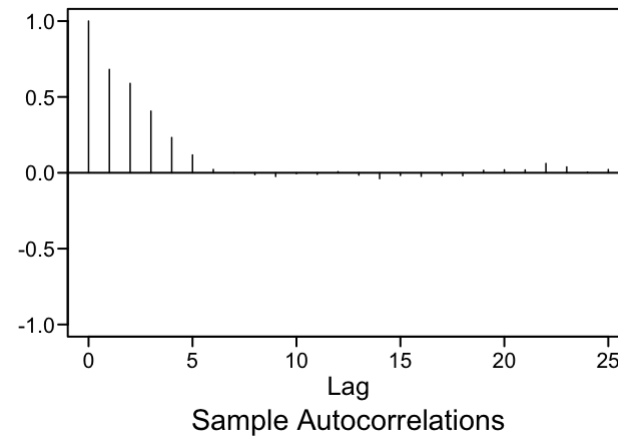
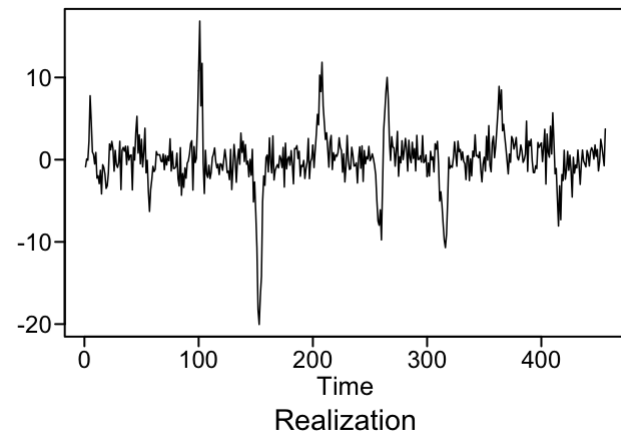


From the above plots. Particle counts and temperature are highly correlated with rmort (respiratory mortality). Whereas temperature and particles are not correlated (-0.0172) with each other. Based on this plot I can say that the predictors "temp" and "particles" are independent.

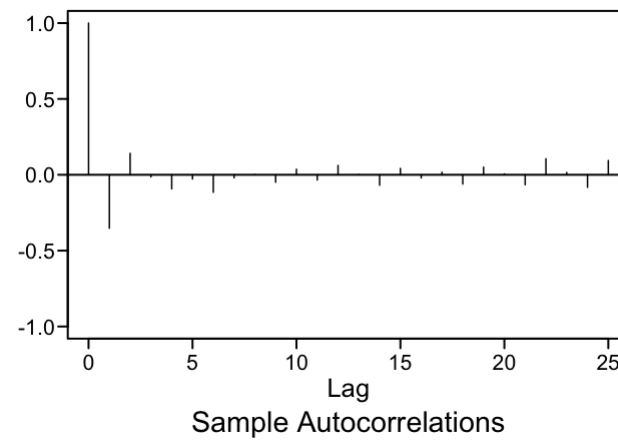
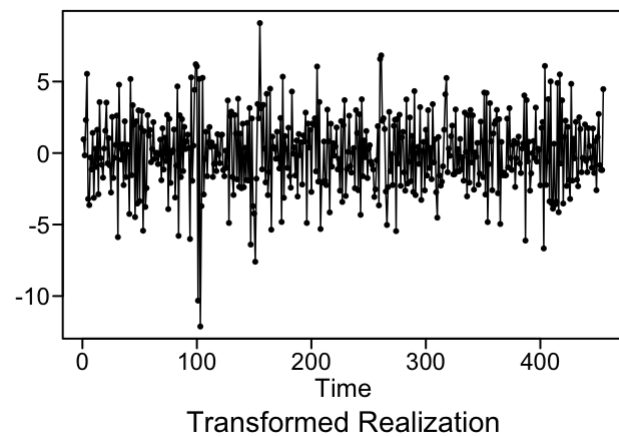
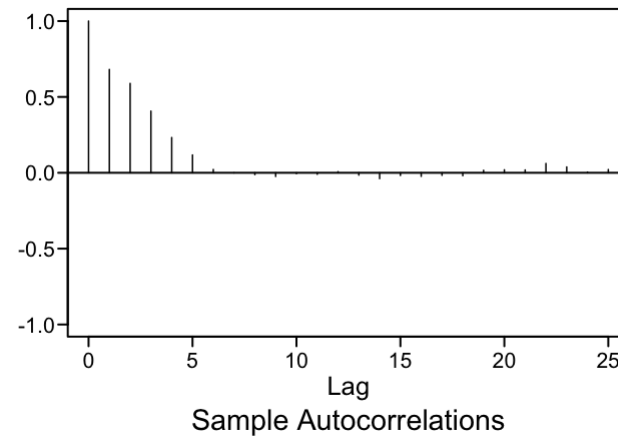
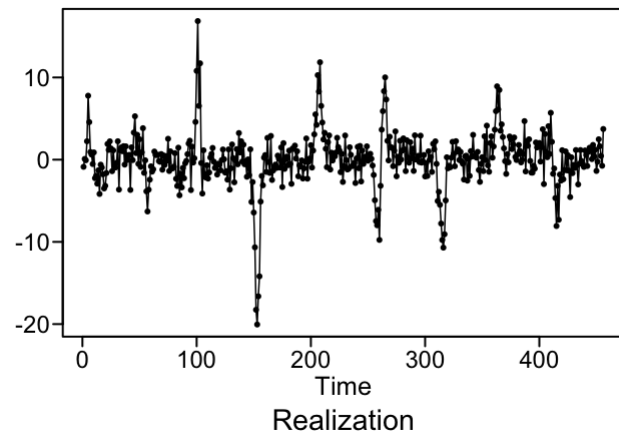
```
# Since the data is weekly, remove weekly trend
RM_rmort = artrans.wge(RM$rmort, c(rep(0,51),1))
```



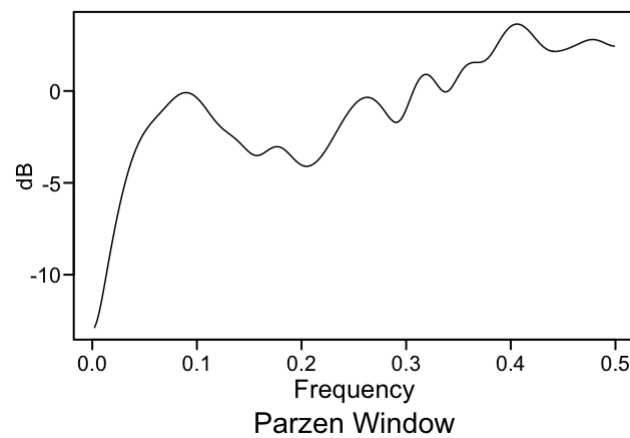
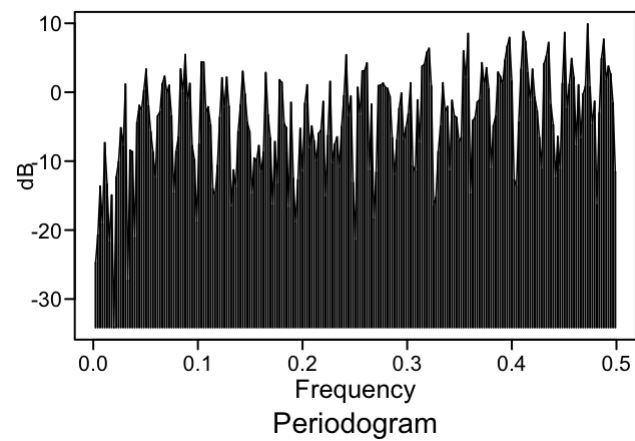
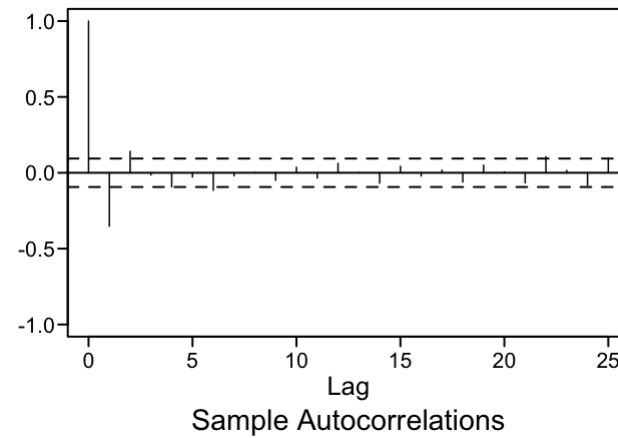
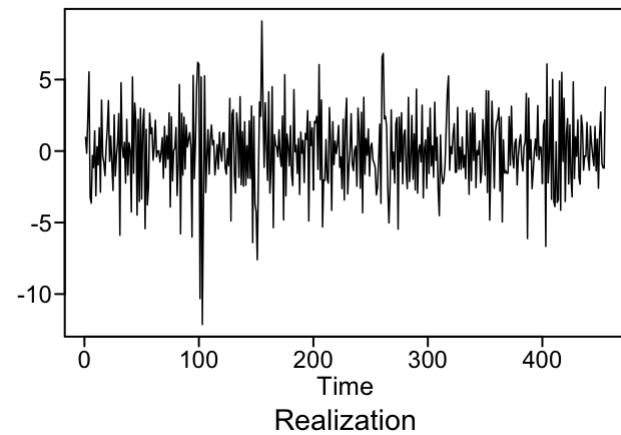
```
# Now plot the transformed data  
plots.sample.wge(RM_rmort)
```



The transformed data still has some autocorrelations, so I have decided to difference it to remove the trend.
`RM_rmort2 = artrans.wge(RM_rmort,phi.tr=1)`



```
# The transformed data now appears white.
plots.sample.wge(RM_rmort2, arlimits = TRUE)
```



```
# set a seed
set.seed(2)
# I will use aic to perform model selection
aic5.wge(RM_rmort2,type = 'aic')
```

```
## -----WORKING... PLEASE WAIT...
##
##
## Five Smallest Values of aic
```

```
##      p      q      aic
## 12     3     2  1.845795
## 16     5     0  1.846906
##  4     1     0  1.855495
## 13     4     0  1.855811
##  3     0     2  1.858729
```

```
# AIC picked ARMA (3,2)
aic.wge(RM_rmort2,type = 'bic')
```

```
## $type
## [1] "bic"
##
## $value
## [1] 1.873607
##
## $p
## [1] 1
##
## $q
## [1] 0
##
## $phi
## [1] -0.3534987
##
## $theta
## [1] 0
##
## $vara
## [1] 6.338893
```

```
# BIC picked ARMA (1,0)
# Now let's check the residuals for white noise using the Ljung-Box test
ljung.wge(RM_rmort2, p=3, q=2)
```



```
## Obs -0.3520291 0.1412502 -0.01406047 -0.09278324 -0.02787398 -0.1153653 -0.01993846 0.001433452 -0.04922364 0.03649838 -0.03414737 0.06161636 0.001979767 -0.06854318 0.04193858 -0.02093571 0.01710674 -0.06097488 0.05008689 0.005862183 -0.06560768 0.1055253 0.01517399 -0.08225973
```

```
## $test
## [1] "Ljung-Box test"
##
## $K
## [1] 24
##
## $chi.square
## [1] 97.92663
##
## $df
## [1] 19
##
## $pval
## [1] 1.268985e-12
```

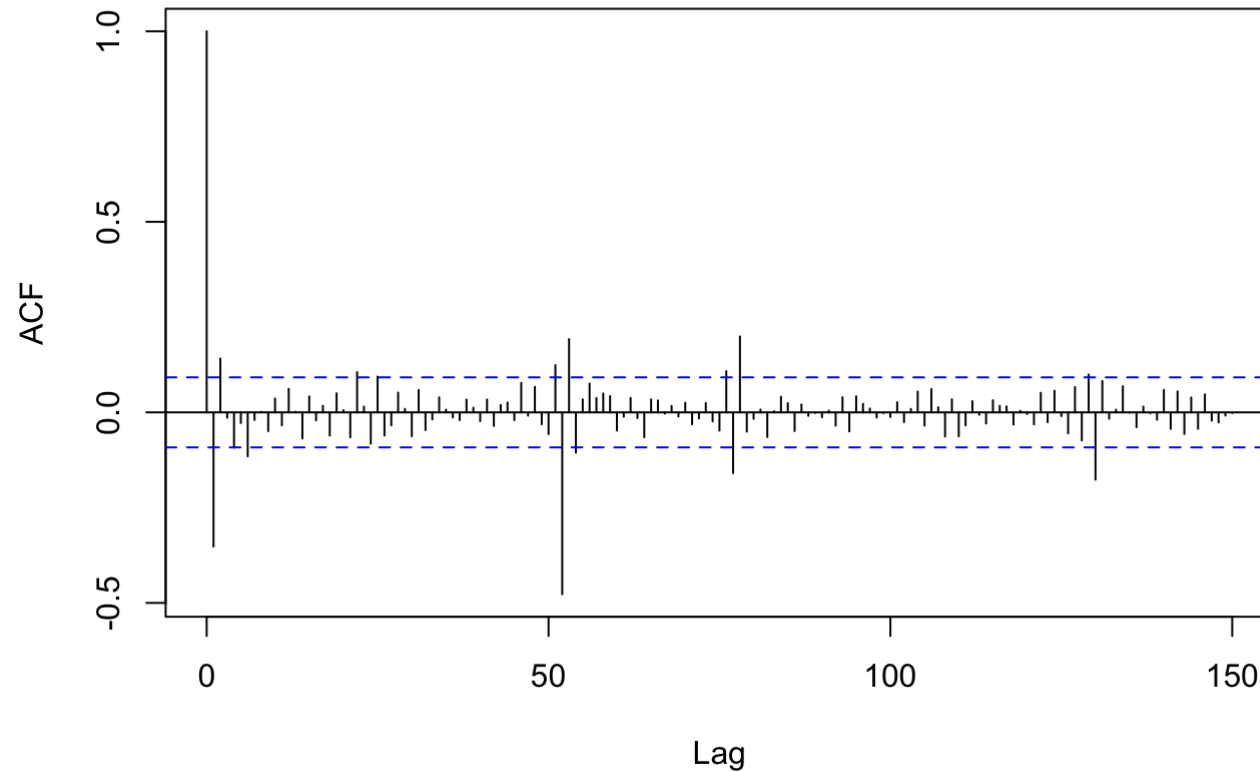
```
ljung.wge(RM_rmort2, p=3, q=2, K = 48)
```

```
## Obs -0.3520291 0.1412502 -0.01406047 -0.09278324 -0.02787398 -0.1153653 -0.01993846 0.001433452 -0.04922364 0.03649838 -0.03414737 0.06161636 0.001979767 -0.06854318 0.04193858 -0.02093571 0.01710674 -0.06097488 0.05008689 0.005862183 -0.06560768 0.1055253 0.01517399 -0.08225973 0.09391878 -0.06079859 -0.03417504 0.05192313 0.00912756 5 -0.06281365 0.05884427 -0.04656787 -0.0186081 0.03935257 0.007024073 -0.01356923 -0.02031177 0.03351866 0.01219 998 -0.02300785 0.03370222 -0.03559723 0.01945384 0.02586647 -0.02045659 0.07743952 -0.008533939 0.06711773
```

```
## $test
## [1] "Ljung-Box test"
##
## $K
## [1] 48
##
## $chi.square
## [1] 120.0932
##
## $df
## [1] 43
##
## $pval
## [1] 3.255047e-09
```

```
# For both K=24 and 48 we reject white noise with p-value less than 0.05. Based on Checks 1 and 2 the residuals from the fitted model do not seem to be white.
acf(RM_rmort2, lag.max = 150)
```

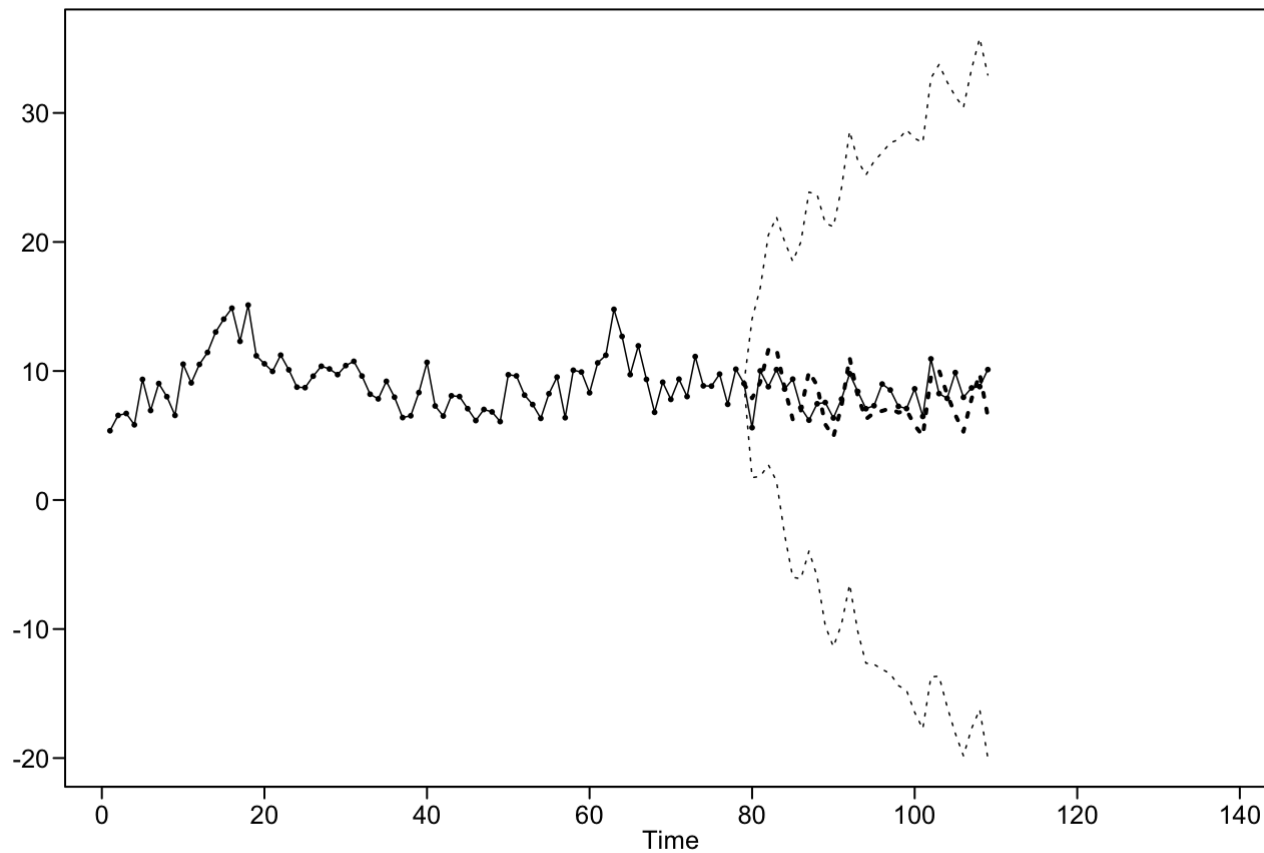
Series RM_rmort2



```
# ACF still confirms that there is still some lags. I will now proceed with forecasts.  
# Get the AIC estimates  
RM_rmort2_est_aic=est.arma.wge(RM_rmort2,p=3,q=2)
```

```
##  
## Coefficients of Original polynomial:  
## 0.3322 -0.6921 -0.3198  
##  
## Factor          Roots          Abs Recip    System Freq  
## 1-0.6775B+0.9261B^2  0.3658+-0.9726i  0.9623      0.1927  
## 1+0.3454B          -2.8954         0.3454      0.5000  
##  
##
```

```
# Now forecast the AIC model  
RM_rmort2_fore_aic=fore.aruma.wge(RM$rmort[400:508],phi=RM_rmort2_est_aic$phi,theta=RM_rmort2_est_aic$theta,n.ahead=30,s=52,d=1,lastn=T)
```

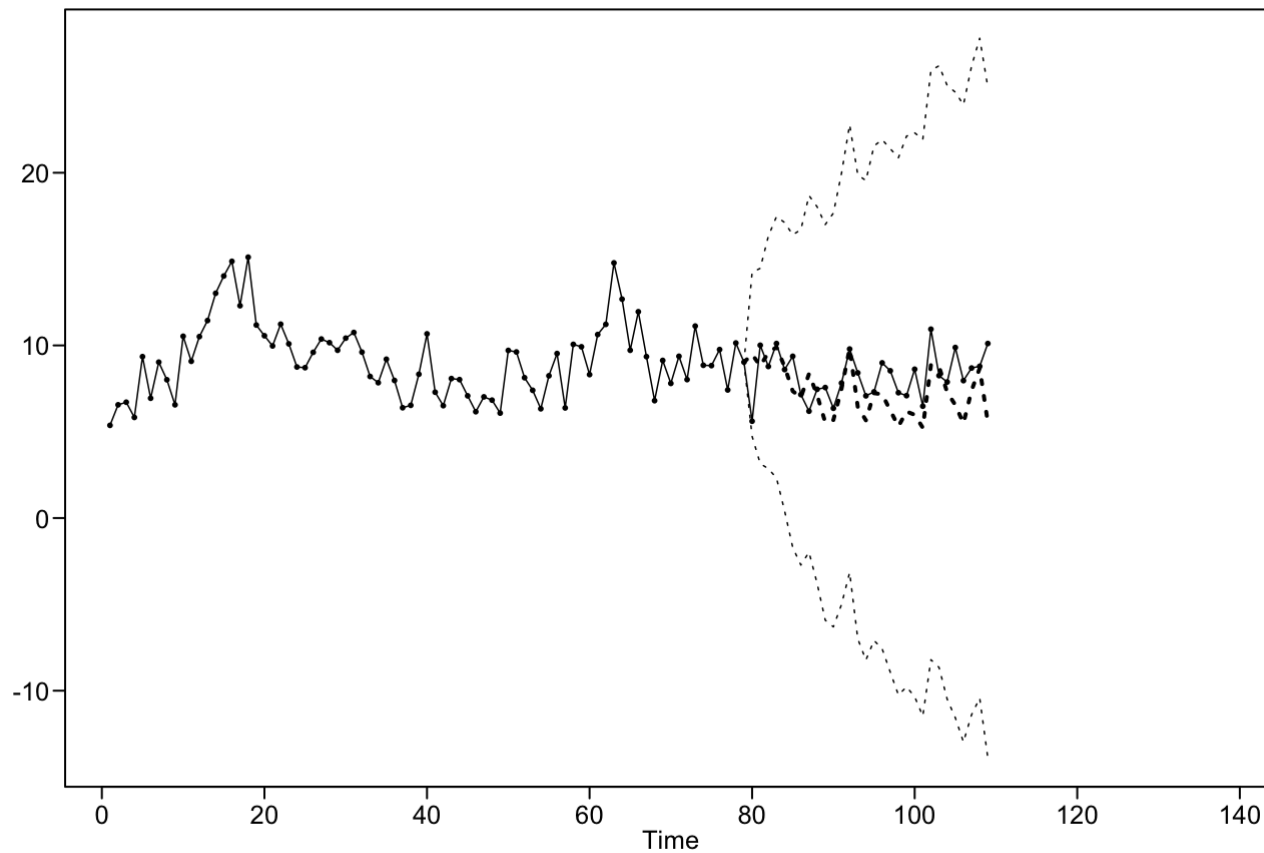


```
# Get the BIC estimates
RM_rmort2_est_bic=est.arma.wge(RM_rmort2,p=1,q=0)
```

```
##
## Coefficients of Original polynomial:
## -0.3535
##
## Factor          Roots          Abs Recip    System Freq
## 1+0.3535B       -2.8289         0.3535      0.5000
##
##
```

```
# Now forecast the BIC model
```

```
RM_rmort2_fore_bic=fore.aruma.wge(RM$rmort[400:508],phi=RM_rmort2_est_bic$phi,theta=RM_rmort2_est_bic$theta,n.ahead=30,s=52,d=1,lastn=T)
```



```
# ASE from AIC
```

```
ASE1 = mean((RM$rmort[(508-30+1):508]-RM_rmort2_fore_aic$f)^2)
```

```
ASE1
```

```
## [1] 3.516085
```

```
# ASE from BIC
ASE2 = mean((RM$rmort[(508-30+1):508]-RM_rmort2_fore_bic$f)^2)
ASE2
```

```
## [1] 3.44117
```

```
# ASE from BIC is lower (3.4411) and I will pick that model.
# (1-B)(1-B^52)(1+0.3534987B)(x_t-8.38) = a_t; sigma^a = 6.33
```

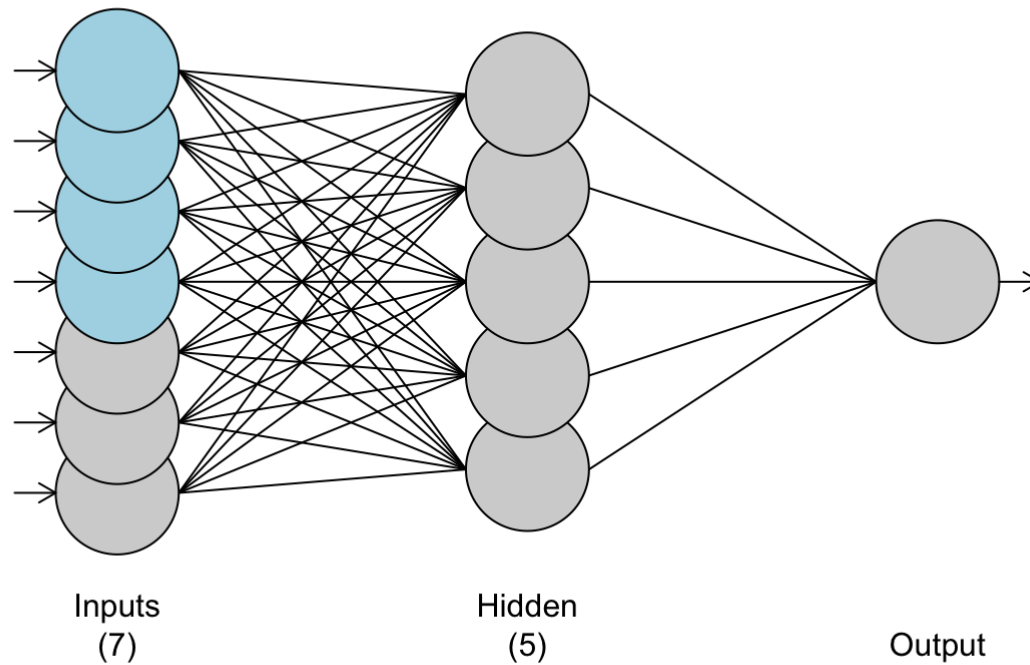
b) using a neural network based model.

```
# set a seed
set.seed(2)
# Create a new data frame with week number included
RM3=RM[c(-1,-3,-5,-7,-8,-9,-10,-11,-12)]
# Create a training set
RMsmall = RM3[1:478,]
RMsmallDF = data.frame(Week = ts(RM3$Time),temp = ts(RM3$tempr), part = ts(RM3$part))
# Fit a neural network mlp model
fit.mlp1 = mlp(ts(RMsmall$rmort),reps = 50,comb = "mean",xreg = RMsmallDF)
fit.mlp1
```

```
## MLP fit with 5 hidden nodes and 50 repetitions.
## Univariate lags: (1,2,4)
## 2 regressors included.
## - Regressor 1 lags: (1,2,4)
## - Regressor 2 lags: (2)
## Forecast combined using the mean operator.
## MSE: 1.6468.
```

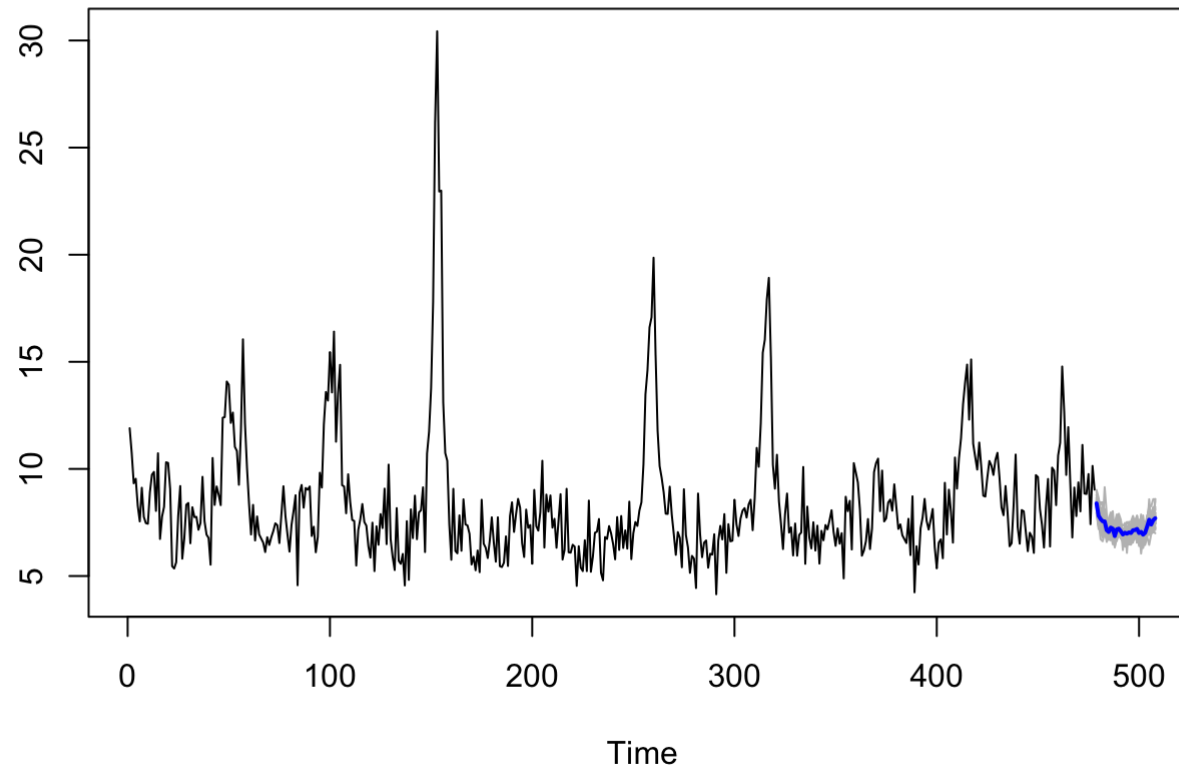
```
# Plot the model
plot(fit.mlp1)
```

MLP

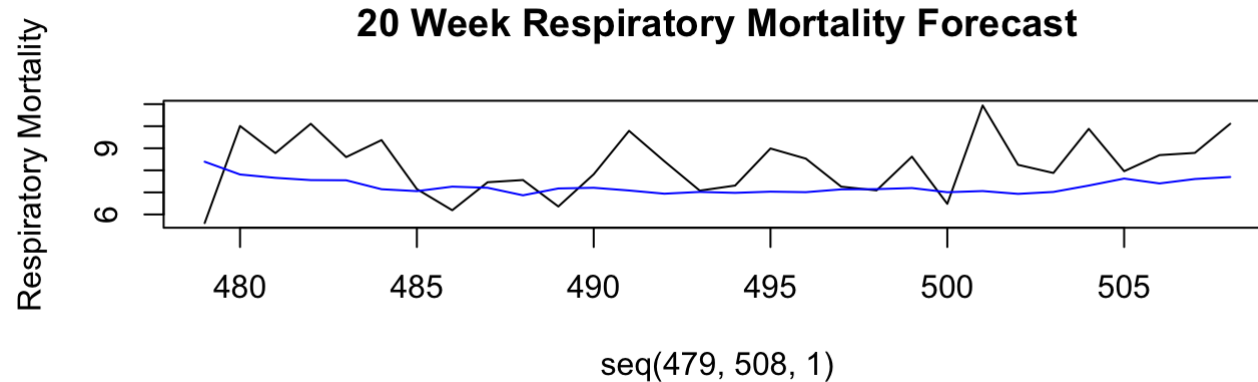
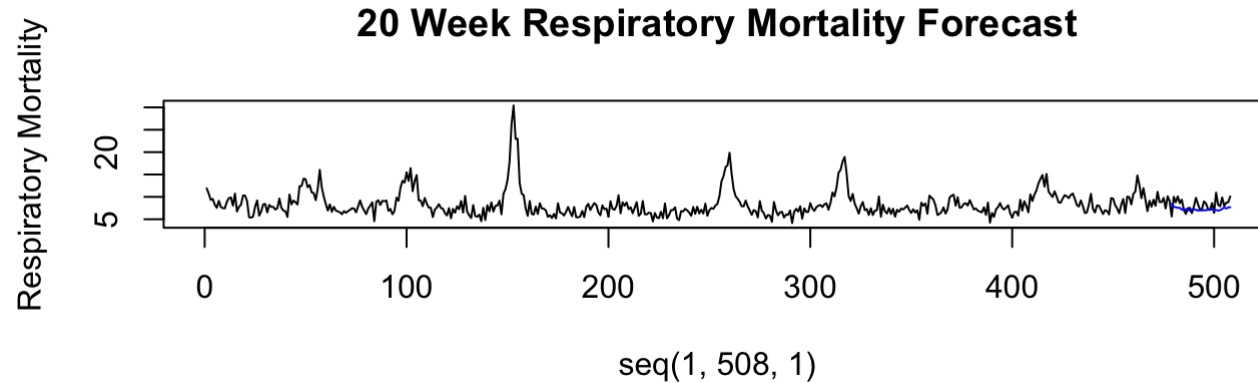


```
# Create a test set
RMDF = data.frame(Week = ts(RM$Time),temp = ts(RM$tempr), part = ts(RM$part))
# Forecast next 30 weeks
fore.mlp1 = forecast(fit.mlp1, h = 30, xreg = RMDF)
# Plot the forecasts
plot(fore.mlp1)
```


Forecasts from MLP



```
par(mfrow = c(2,1))
#Plot
plot(seq(1,508,1), RM$mort, type = "l",xlim = c(0,508), ylab = "Respiratory Mortality", main = "20 Week Respirat
ory Mortality Forecast")
lines(seq(479,508,1), fore.mlp1$mean, type = "l", col = "blue")
#Plot
plot(seq(479,508,1), RM$mort[479:508], type = "l",xlim = c(479,508), ylab = "Respiratory Mortality", main = "20
Week Respiratory Mortality Forecast")
lines(seq(479,508,1), fore.mlp1$mean, type = "l", col = "blue")
```



```
# Calculate and display ASE
ASE = mean((RM$rmort[479:508] - fore.mlp1$mean)^2)
ASE
```

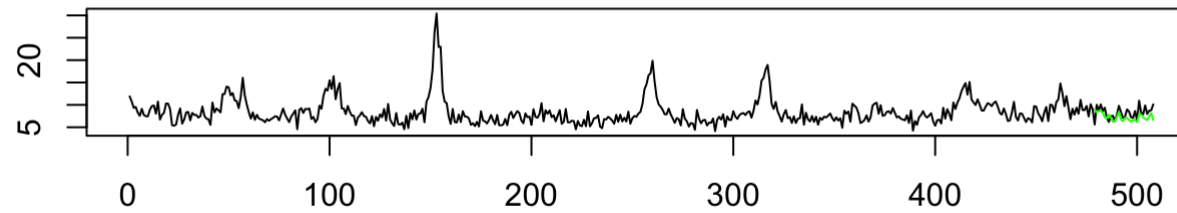
```
## [1] 2.676626447
```

```
# ASE using neural network model is 2.6766
```

c) an ensemble model with a model from (a) and (b).

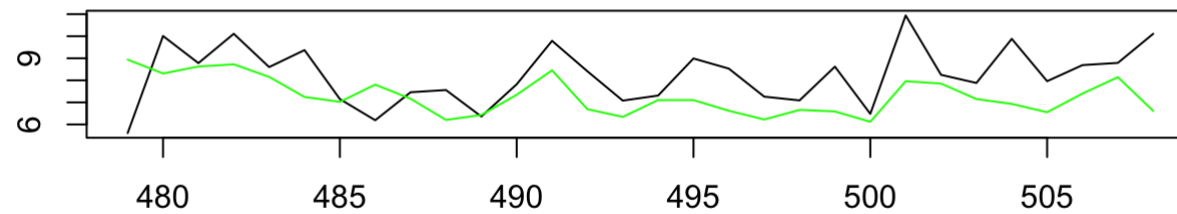
```
# set a seed
set.seed(2)
# I will now create an ensemble model from BIC and mlp
ensemble = (RM_rmort2_fore_bic$f + fore.mlp1$mean)/2
# Plot the forecasts
par(mfrow = c(2,1))
#Plot
plot(seq(1,508,1), RM$rmort, type = "l",xlim = c(0,508), ylab = "Respiratory Mortality", main = "20 Week Respirat
ory Mortality Forecast")
lines(seq(479,508,1), ensemble, type = "l", col = "green")
#Plot
plot(seq(479,508,1), RM$rmort[479:508], type = "l",xlim = c(479,508), ylab = "Respiratory Mortality", main = "20
Week Respiratory Mortality Forecast")
lines(seq(479,508,1), ensemble, type = "l", col = "green")
```

Respiratory Mortality

20 Week Respiratory Mortality Forecast

seq(1, 508, 1)

Respiratory Mortality

20 Week Respiratory Mortality Forecast

seq(479, 508, 1)

```
# Calculate the ASE of the ensemble model
ASE = mean((RM$rmort[479:508] - ensemble)^2)
ASE
```

```
## [1] 2.599823775
```

```
# ASE using ensemble model is 2.5998
```

3b. (5 pts) Compare these models and describe which univariate model you feel is the best and why.

With ARIMA Seasonal model using BIC I got an ASE of 3.4411. With neural network MLP model I got an ASE of 2.6766 and with the combined ensemble model, the ASE was 2.5998. All of these models have their ASE's very close and might perform better than one another if I try multiple seeds. Remembering George Box's quote: All models are wrong but some are useful. With seed 2, the ensemble model has the lowest ASE and I will use that model for forecasts.

4a. (10 pts) Perform a multivariate analysis using at least one model from each category:

4aa. VAR model:

```
# set a seed
set.seed(2)
# I will use VARselect to pick p
VARselect(cbind(RMsmall$rmort[1:478], RMsmall$part[1:478], RMsmall$tempr[1:478]), lag.max = 10, season = 52, type
= "both")
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      5      2      1      5
##
## $criteria
##           1           2           3           4
## AIC(n)    9.044853210    8.973981156    8.963536466    8.926781628
## HQ(n)     9.630845102    9.591365471    9.612313204    9.606950789
## SC(n)    10.534046957    10.542953139    10.612286686    10.655310084
## FPE(n)  8504.149474681  7926.977360509  7849.744104538  7571.927071045
##           5           6           7           8
## AIC(n)    8.918012533    8.945403036    8.961133296    8.975540213
## HQ(n)     9.629574117    9.688357042    9.735479726    9.781279065
## SC(n)    10.726319226    10.833487965    10.928996463    11.023181615
## FPE(n)  7511.770755690  7727.065896897  7857.003858002  7979.215915289
##           9           10
## AIC(n)    8.979486345    8.963601153
## HQ(n)     9.816617620    9.832124851
## SC(n)    11.106905984    11.170799029
## FPE(n)  8019.690916872  7902.804804141
```

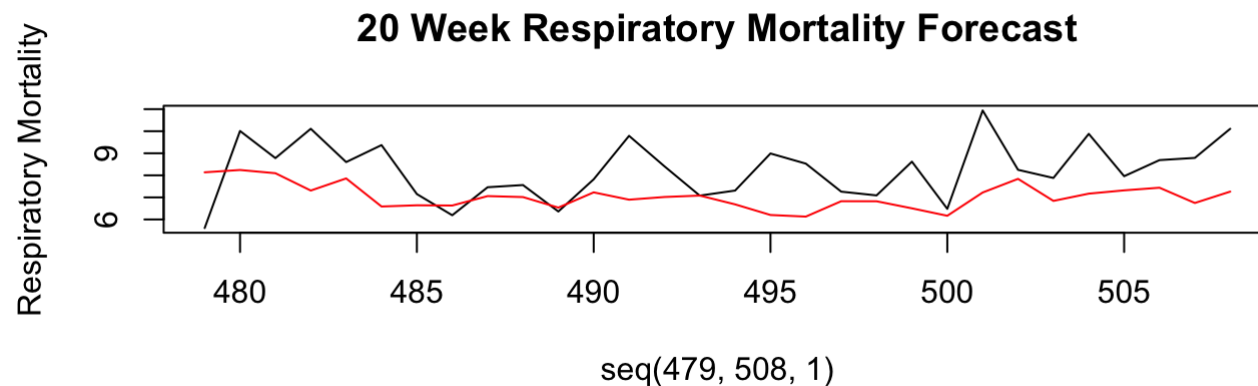
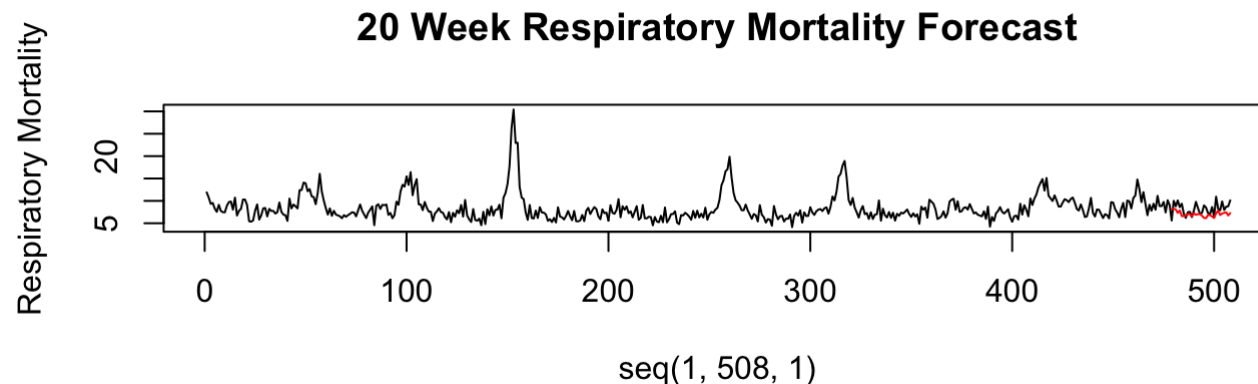
```
# VARselect picks p=5 (using AIC) and p=1 (using BIC). I will select p=1 from BIC
RMortVAR = VAR(cbind(RMsmall$rmort[1:478], RMsmall$part[1:478], RMsmall$tempr[1:478]),season = 52, type = "both",
p = 1)
```

```
## Warning in VAR(cbind(RMsmall$rmort[1:478], RMsmall$part[1:478], RMsmall$tempr[1:478]), : No column names supplied in y, using: y1, y2, y3 , instead.
```

```

# Forecast next 30 weeks
preds=predict(RMortVAR,n.ahead=30)
# Plot the forecasts
par(mfrow = c(2,1))
#Plot
plot(seq(1,508,1), RM$rmort, type = "l",xlim = c(0,508), ylab = "Respiratory Mortality", main = "20 Week Respiratory Mortality Forecast")
lines(seq(479,508,1), preds$fcst$y1[,1], type = "l", col = "red")
#Plot
plot(seq(479,508,1), RM$rmort[479:508], type = "l",xlim = c(479,508), ylab = "Respiratory Mortality", main = "20 Week Respiratory Mortality Forecast")
lines(seq(479,508,1), preds$fcst$y1[,1], type = "l", col = "red")

```



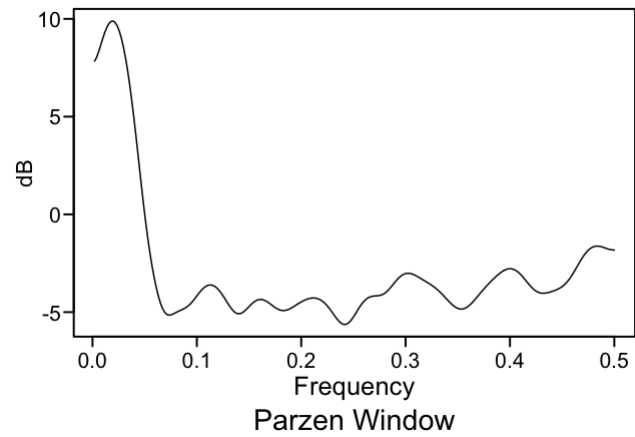
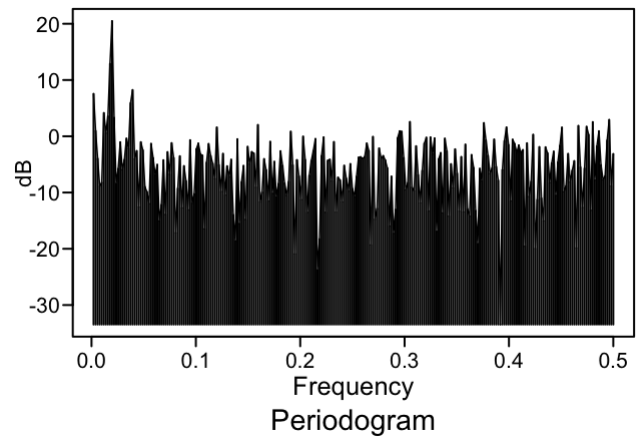
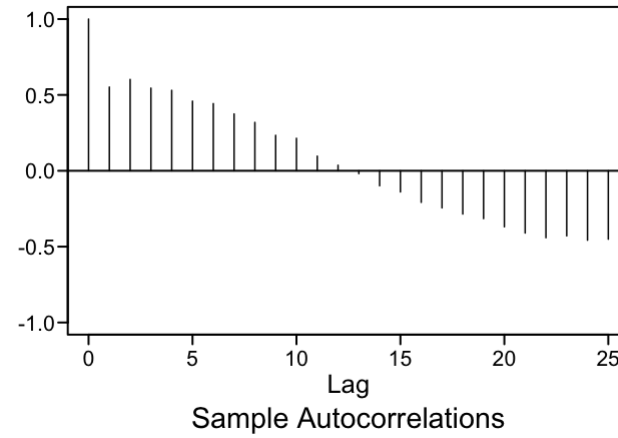
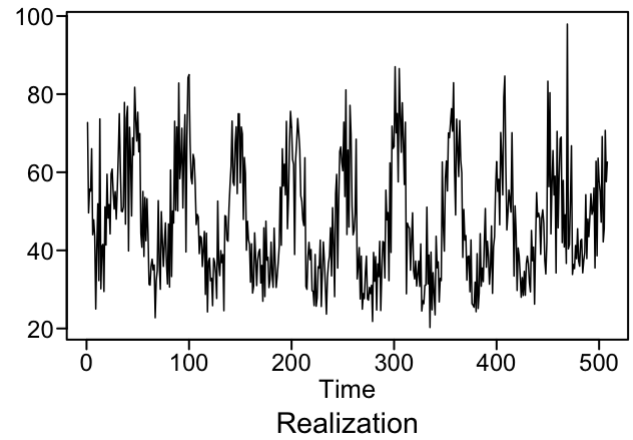
```
# Calculate and display ASE
ASE = mean((RM$rmort[479:508] - preds$fcst$y1[,1])^2)
ASE
```

```
## [1] 3.10579714
```

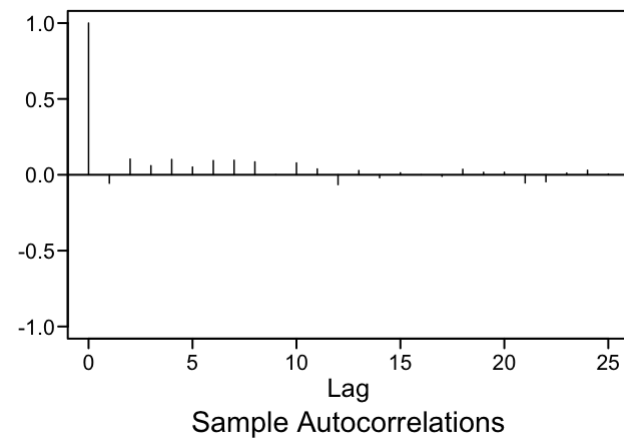
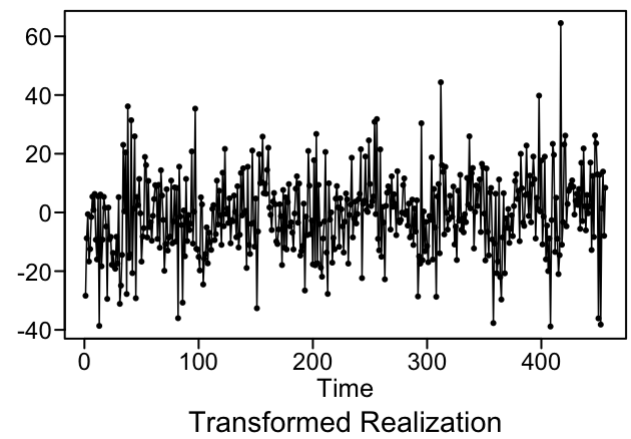
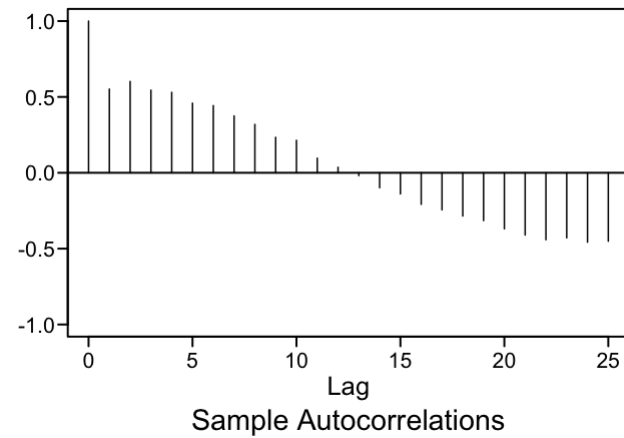
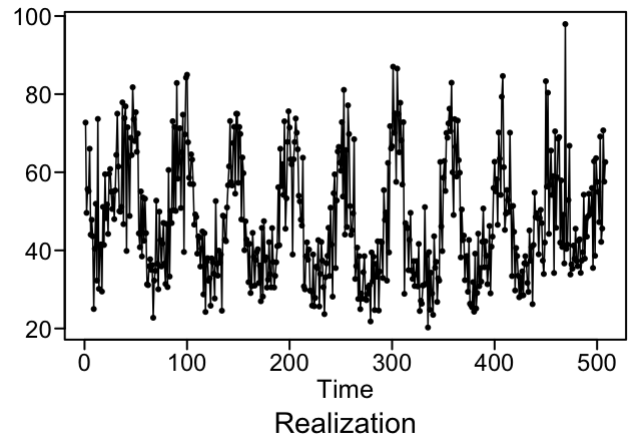
```
# ASE using VAR model is 3.1057
```

4ab. MLR with correlated errors:

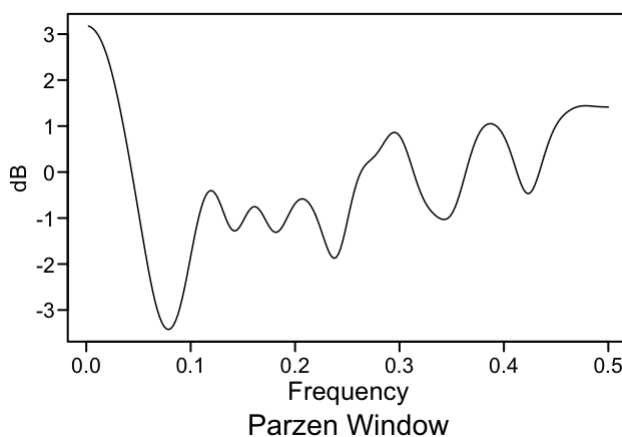
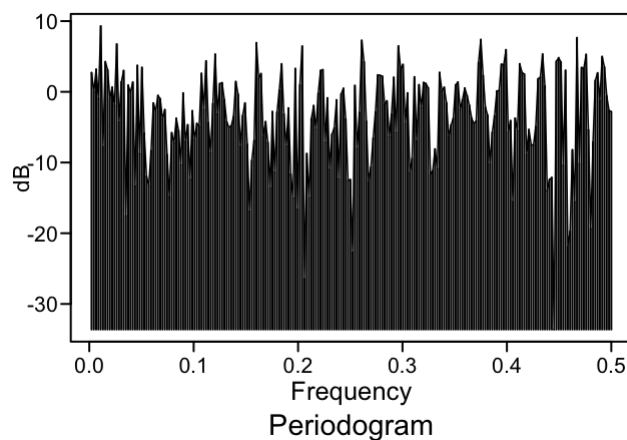
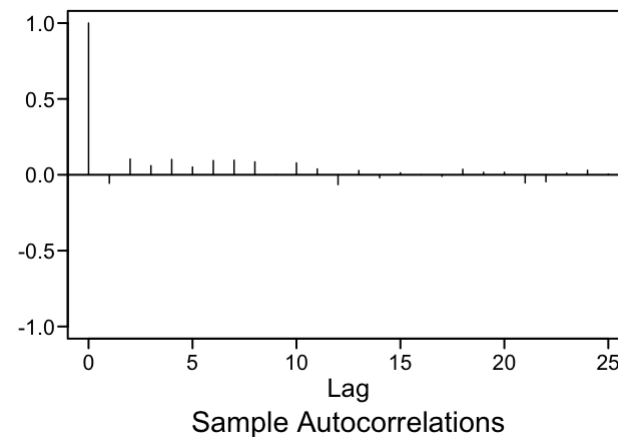
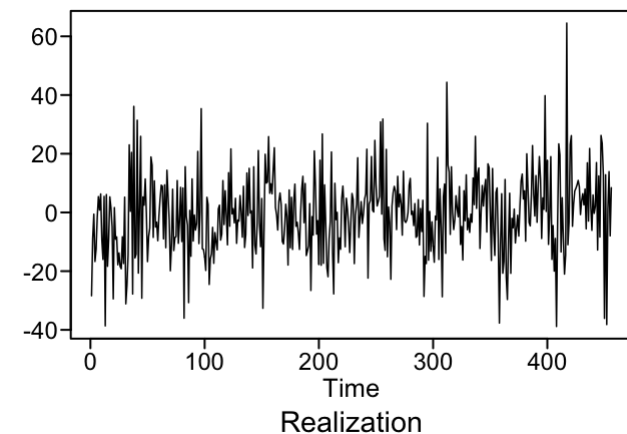
```
# set a seed
set.seed(2)
#forecast Particles
# plots show a freq near .0192 (annual)
plotts.sample.wge(RM$part)
```

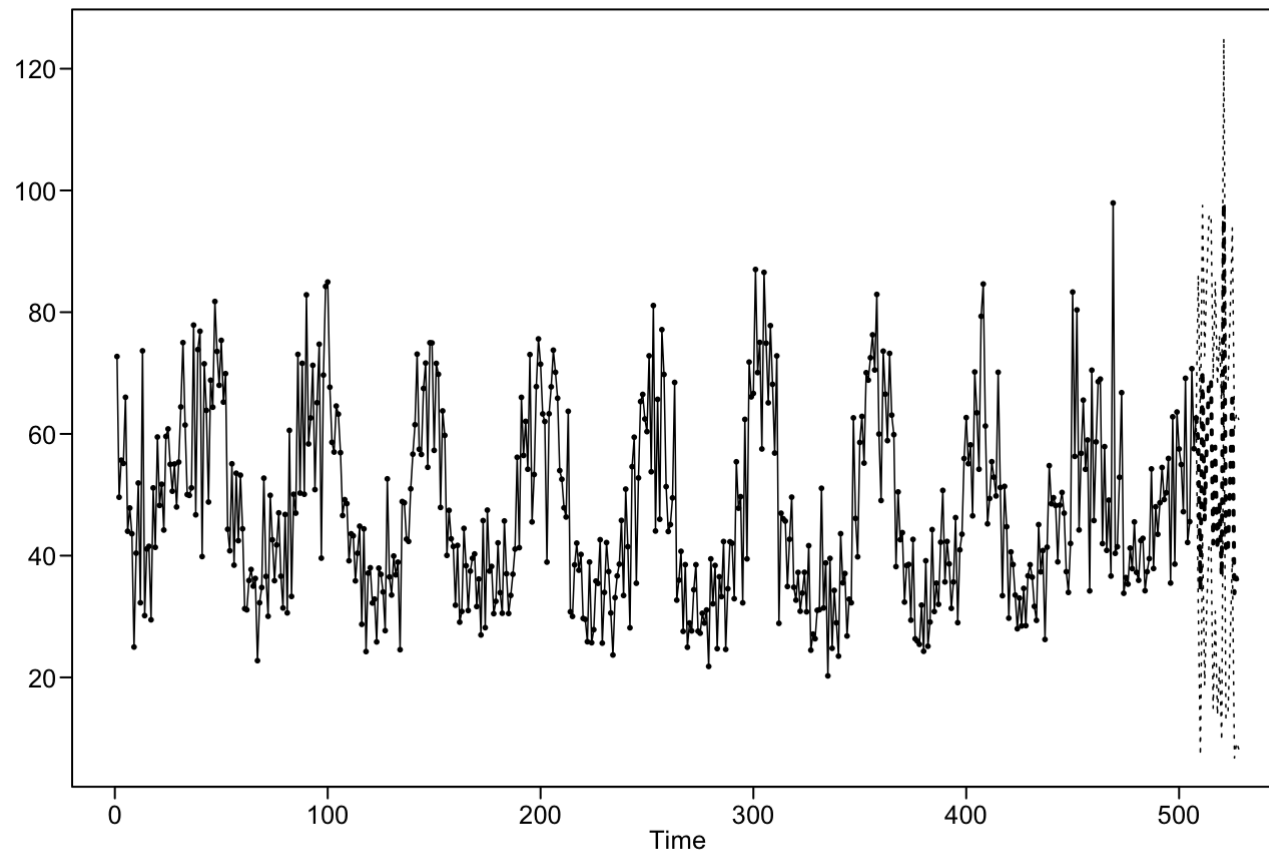
```
# Remove the seasonality
RM_52 = artrans.wge(RM$part, c(rep(0,51),1))
```



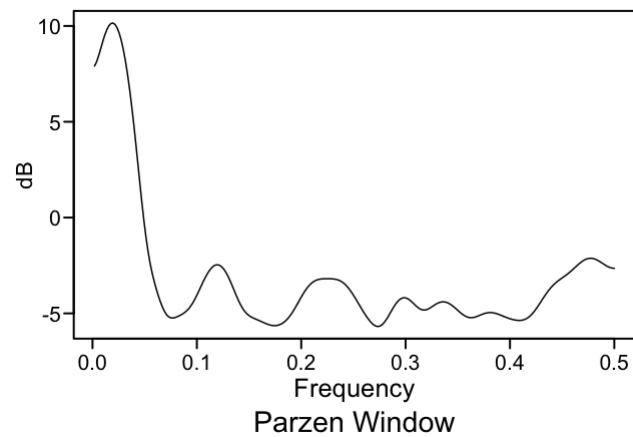
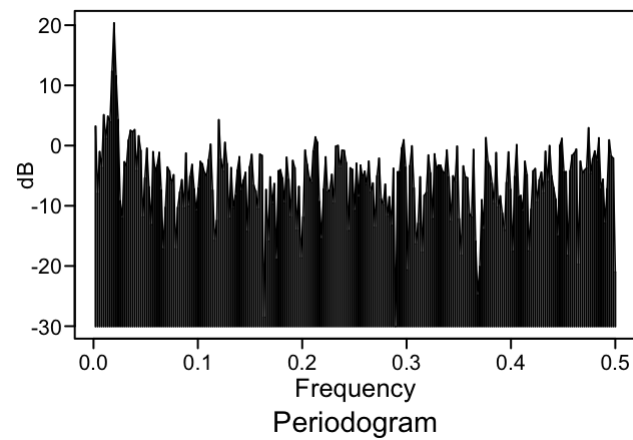
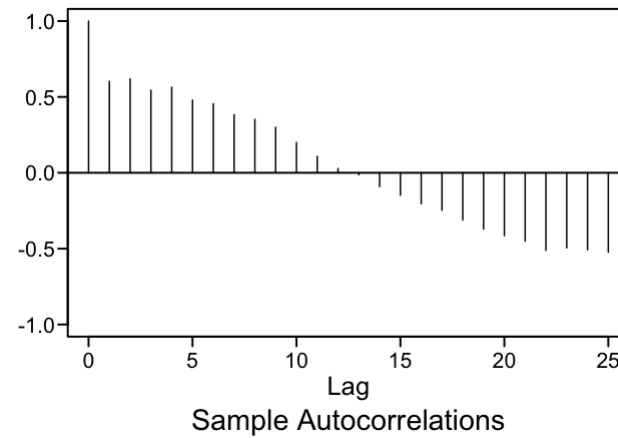
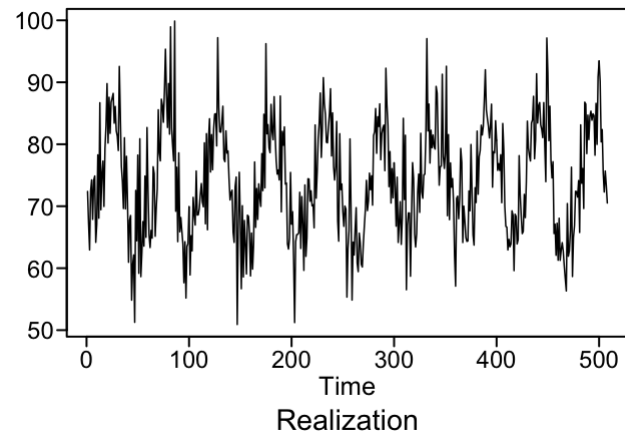
```
# plot again. Looks like there is some low freq.  
plotts.sample.wge(RM_52)
```



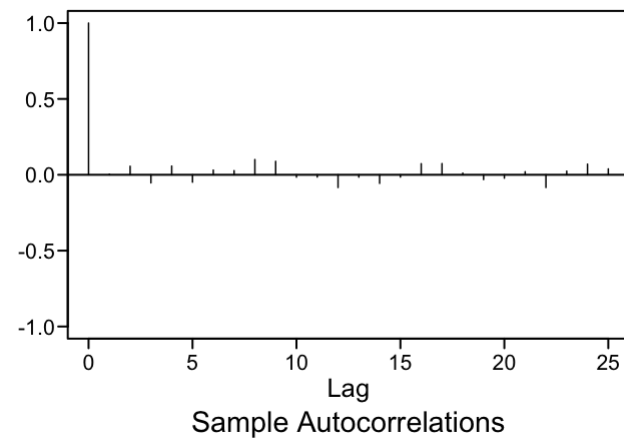
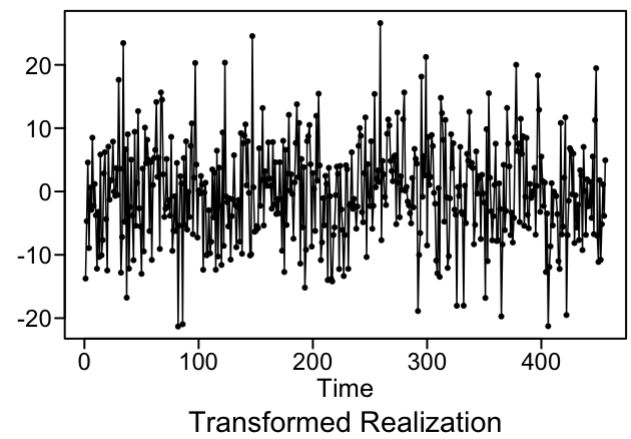
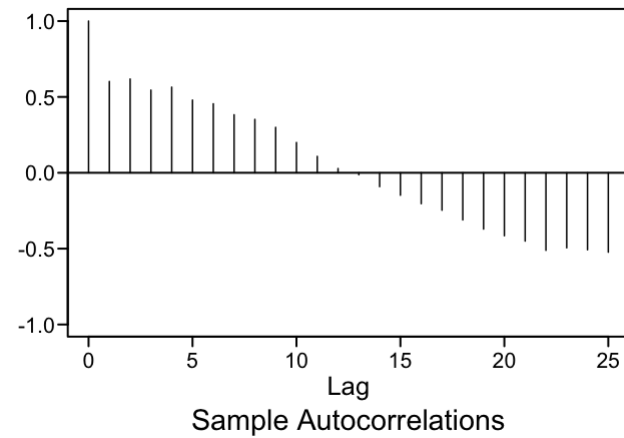
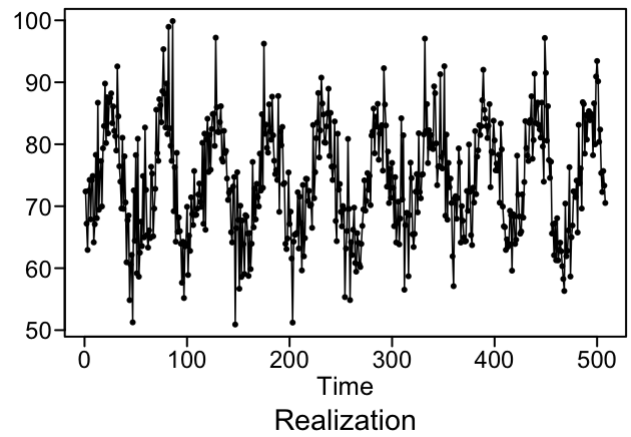
```
# aic5 picks ARMA(2,1). I will assume stationary
aic5.wge(RM_52)
# bic picks ARMA(0,0)
aic5.wge(RM_52,type = "bic")
# Now let's check the residuals for white noise using the Ljung-Box test
ljung.wge(RM_52)$pval
ljung.wge(RM_52, K = 48)$pval
# For K=48 we fail to reject white noise. Based on Checks 1 and 2 the residuals from the fitted ARMA(2,1) model seem to be white.
# Forecast the next 20 weeks
predsPart = fore.aruma.wge(RM$part,s = 52, n.ahead = 20)
```



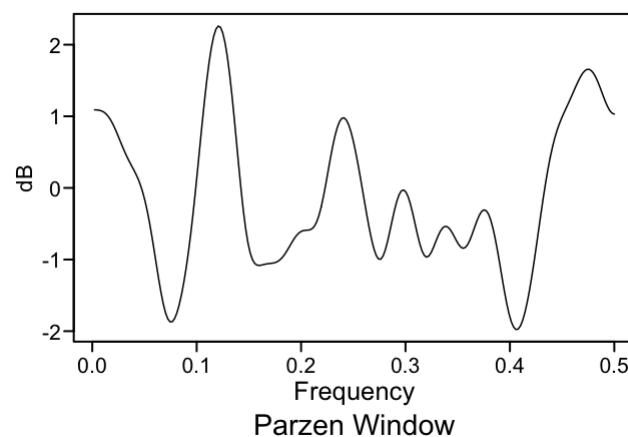
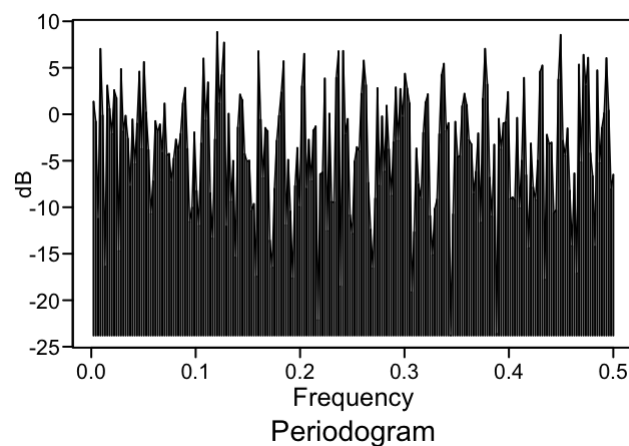
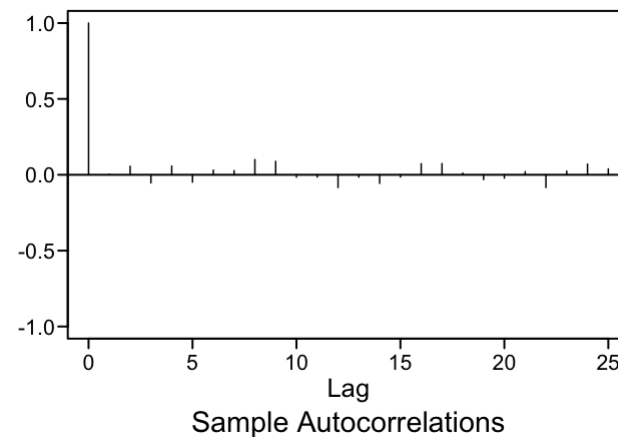
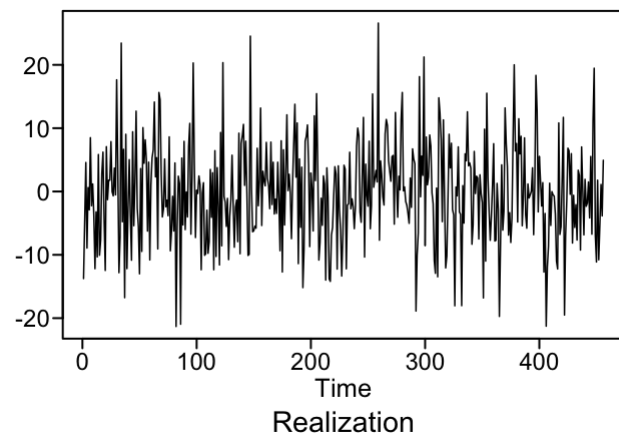
```
#forecast Temp  
# plots show a freq near .0192 (annual)  
plotts.sample.wge(RM$temp)
```



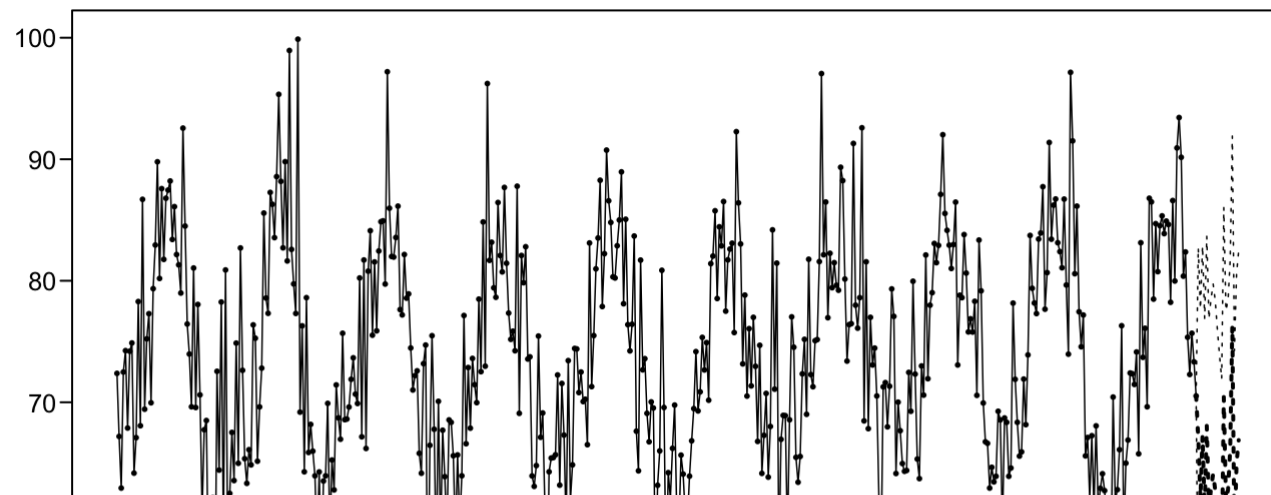
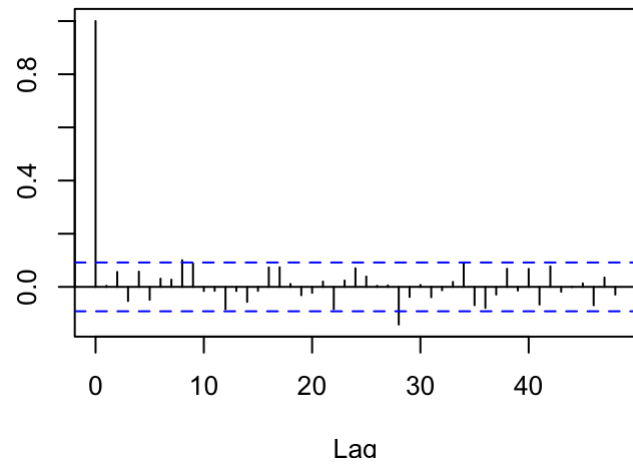
```
# Remove the seasonality
RM_52 = artrans.wge(RM$tempr, c(rep(0,51),1))
```

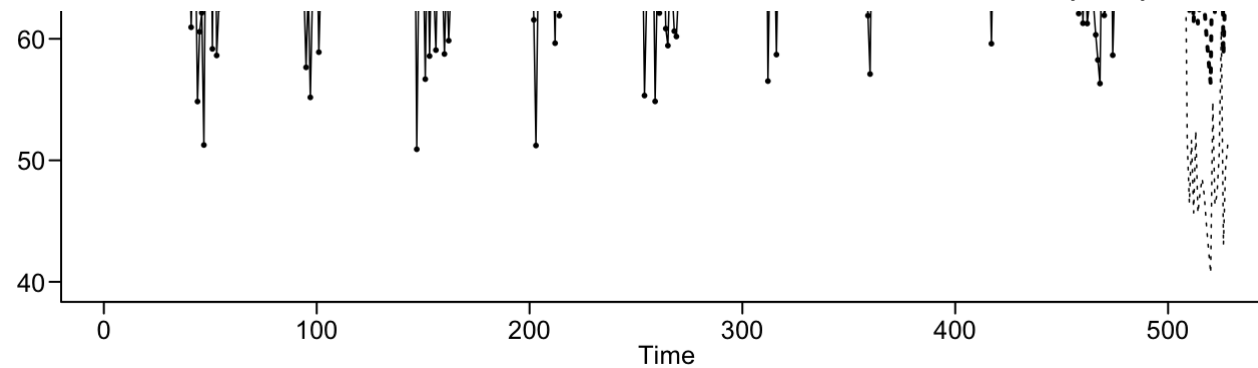


```
# plot again.  
plotts.sample.wge(RM_52)
```



```
# aic5 picks ARMA(0,0)
aic5.wge(RM_52)
# Now let's check the residuals for white noise using the Ljung-Box test
ljung.wge(RM_52)$pval
ljung.wge(RM_52, K = 48)$pval
# For both K=24 and 48 we fail to reject white noise
# Forecast the next 20 weeks
# acf looks consistent with white noise
acf(RM_52, lag.max = 48)
# Forecast the next 20 weeks
predsTemp = fore.aruma.wge(RM$tempr, s = 52, n.ahead = 20)
```

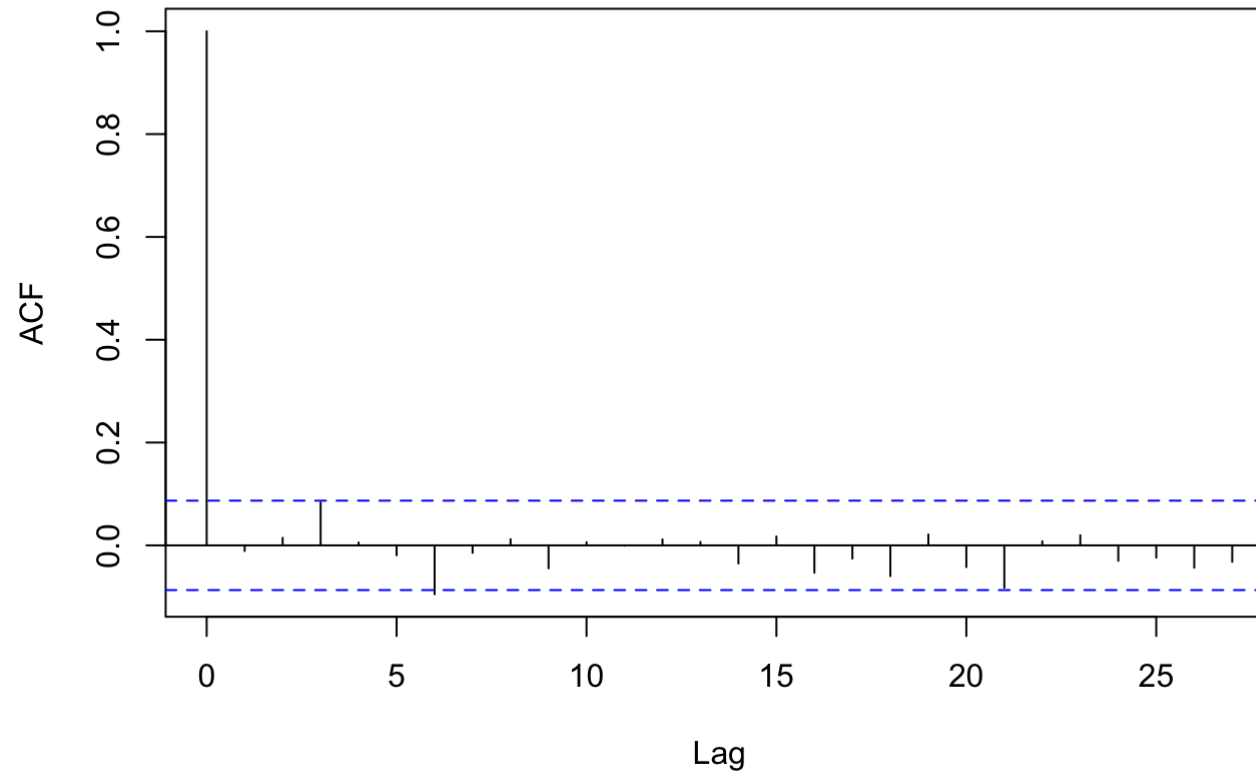





```
# Model rmort based on predicted part and temp using MLR with Cor Erros
#assuming data is loaded in dataframe RMFullDF
RMFullDF = data.frame(Week = ts(RM$Time),temp = ts(RM$tempr), part = ts(RM$part), rmort = ts(RM$rmort))
ksfit = lm(RMFullDF$rmort~RMFullDF$temp+RMFullDF$part+RMFullDF$Week)
phi = aic.wge(ksfit$residuals)
fit = arima(RMFullDF$rmort,order = c(phi$p,0,0), seasonal = list(order = c(1,0,0), period = 52), xreg = cbind(RMFullDF$temp, RMFullDF$part, RMFullDF$Week))

# Check for whiteness of residuals
acf(fit$residuals)
```

Series fit\$residuals



```
ljung.wge(fit$residuals) # pval = .6970
```

```
## Obs -0.0110382395 0.01468307651 0.08330824346 0.005673070689 -0.01931918393 -0.09503461387 -0.01460508257 0.01
207580129 -0.04468438394 0.006035703618 -0.0007268842208 0.01169383343 0.00668396178 -0.03527883264 0.01722745816
-0.05310402045 -0.02567609828 -0.05986020413 0.02116376003 -0.0422235713 -0.08236655303 0.007970467103 0.01975374
346 -0.03004364714
```

```
## $test
## [1] "Ljung-Box test"
##
## $K
## [1] 24
##
## $chi.square
## [1] 19.99433067
##
## $df
## [1] 24
##
## $pval
## [1] 0.6970985053
```

```
ljung.wge(fit$residuals, K = 48) # pval = .7992
```

```
## Obs -0.0110382395 0.01468307651 0.08330824346 0.005673070689 -0.01931918393 -0.09503461387 -0.01460508257 0.01
207580129 -0.04468438394 0.006035703618 -0.0007268842208 0.01169383343 0.00668396178 -0.03527883264 0.01722745816
-0.05310402045 -0.02567609828 -0.05986020413 0.02116376003 -0.0422235713 -0.08236655303 0.007970467103 0.01975374
346 -0.03004364714 -0.0238637395 -0.04345662204 -0.03198859521 0.03185241534 -0.008529765385 -0.1012542566 -0.009
245183577 -0.03061243148 -0.027365582 -0.02629011625 -0.01827326936 -0.03234533292 -0.01632187025 0.02382941266 -
0.001602987306 -0.03552192966 0.02951592644 -0.05078726602 0.03302163033 0.04222996461 0.001748642305 0.026237772
7 -0.01634094275 0.08755192106
```

```
## $test
## [1] "Ljung-Box test"
##
## $K
## [1] 48
##
## $chi.square
## [1] 39.64502937
##
## $df
## [1] 48
##
## $pval
## [1] 0.799203149
```

```
# For both K=24 and 48 we fail to reject white noise
```

```
#load the forecasted Part and Temp in a data frame
```

```
next20 = data.frame(temp = predsTemp$f, part = predsPart$f, Week = seq(509,528,1))
```

```
#get predictions
```

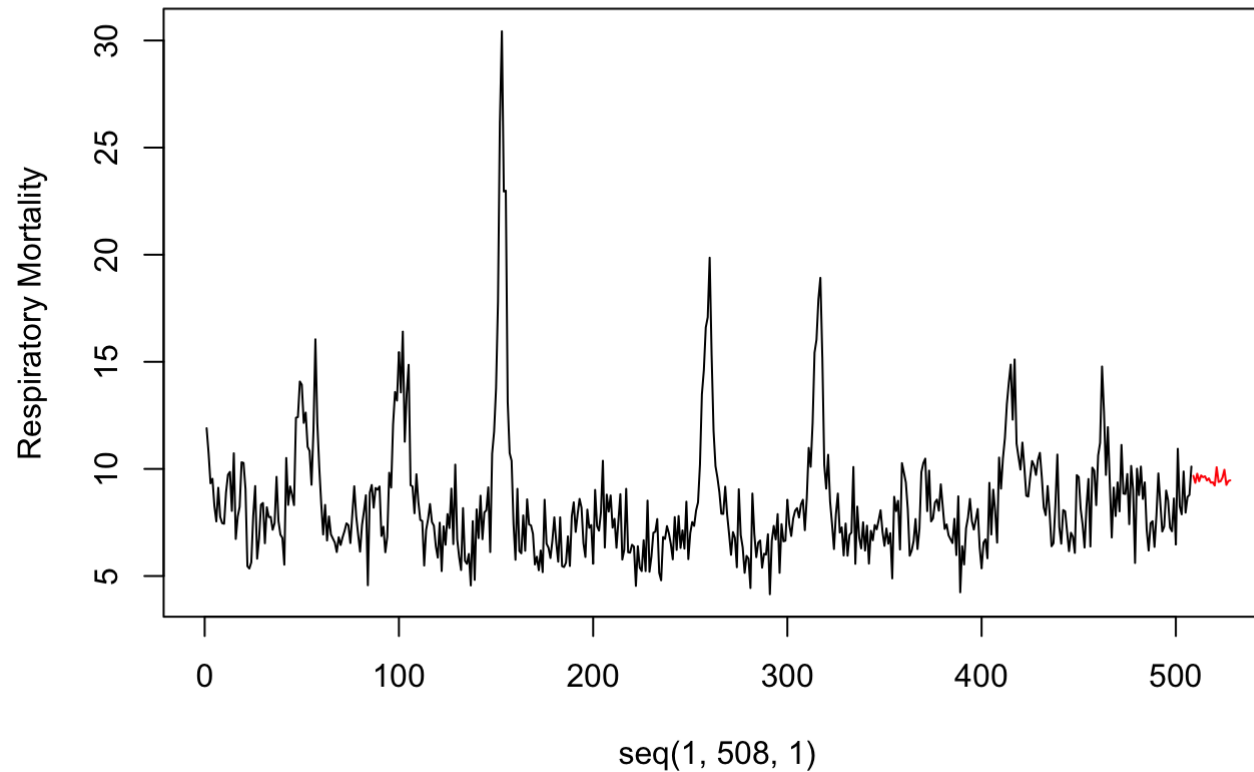
```
predsRMort = predict(fit,newxreg = next20)
```

```
#plot next 20 rmort wrt time
```

```
plot(seq(1,508,1), RM$rmort, type = "l",xlim = c(0,528), ylab = "Respiratory Mortality", main = "20 Week Respiratory Mortality Forecast")
```

```
lines(seq(509,528,1), predsRMort$pred, type = "l", col = "red")
```

20 Week Respiratory Mortality Forecast



```
#Find ASE Need to forecast last 30 of known series.
RMsmall = RMFullDF[1:478,]
attach(RMsmall)
```

```
## The following object is masked from package:astsa:
##
## part
```

```
ksfit = lm(rmort~temp+part+Week, data = RMsmall)
phi = aic.wge(ksfit$residuals)
attach(RMsmall)
```

```
## The following objects are masked from RMsall (pos = 3):  
##  
##      part, rmort, temp, Week  
##  
## The following object is masked from package:astsa:  
##  
##      part
```

```
fit = arima(RMsall$rmort, order = c(phi$p, 0, 0), seasonal = list(order = c(1, 0, 0), period = 52), xreg = cbind(RMsm  
all$temp, RMsall$part, RMsall$Week))  
  
last30 = data.frame(temp = RM$temp[479:508], part = RM$part[479:508], Week = seq(479, 508, 1))  
#get predictions  
predsRMort = predict(fit, newxreg = last30)  
  
# Calculate and display ASE  
ASE = mean((RM$rmort[479:508] - predsRMort$pred)^2)  
ASE
```

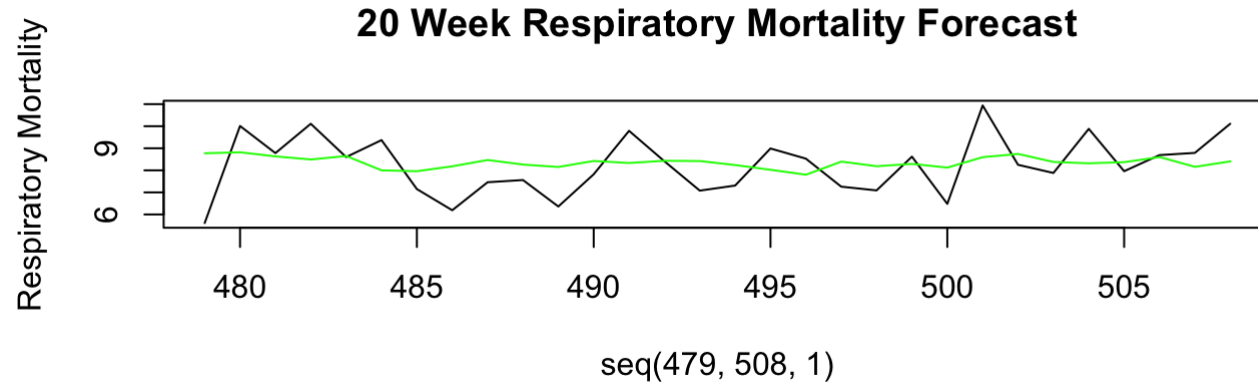
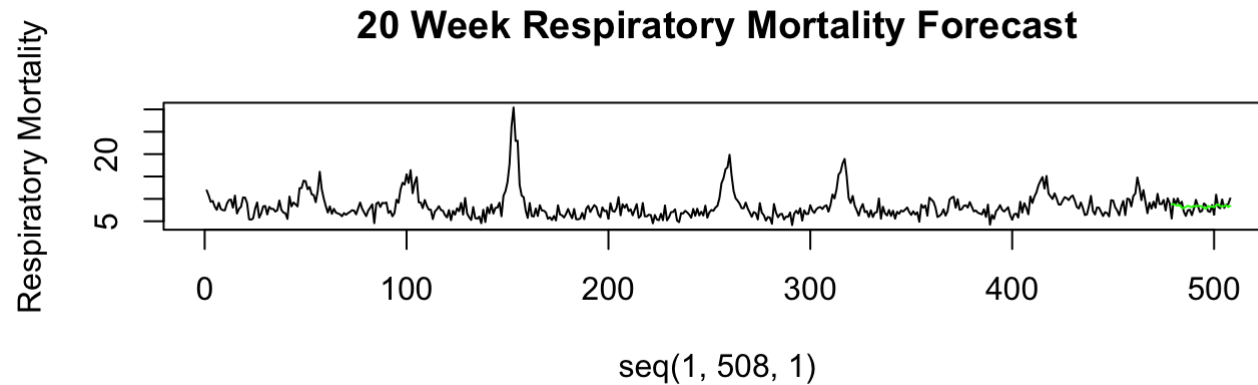
```
## [1] 3.812581738
```

```
# ASE using MLR with correlated errors is 3.8125
```

4b. (5pts) Fit and evaluate an ensemble model from the models you fit in 4a.

```
# set a seed
set.seed(2)
# I will now create an ensemble model from VAR and MLR models
ensemble2 = (preds$fcst$y1[,1] + predsRMort$pred)/2

# Plot the forecasts
par(mfrow = c(2,1))
#Plot
plot(seq(1,508,1), RM$mort, type = "l",xlim = c(0,508), ylab = "Respiratory Mortality", main = "20 Week Respiratory Mortality Forecast")
lines(seq(479,508,1), ensemble2, type = "l", col = "green")
#Plot
plot(seq(479,508,1), RM$mort[479:508], type = "l",xlim = c(479,508), ylab = "Respiratory Mortality", main = "20 Week Respiratory Mortality Forecast")
lines(seq(479,508,1), ensemble2, type = "l", col = "green")
```

```
# Calculate the ASE of the ensemble model
ASE = mean((RM$rmort[479:508] - ensemble2)^2)
ASE
```

```
## [1] 1.637225041
```

```
# ASE using ensemble model is 1.6372
```

4c. (5 pts) Compare these models and describe which multivariate model you feel is the best and why.

With VAR model I got an ASE of 3.1057. With MLR correlated errors, I got an ASE of 3.8125 and with the combined multivariate ensemble model, the ASE was 1.6372. Remembering George Box's quote: All models are wrong but some are useful. With seed 2, the ensemble model has the lowest ASE and I will use that model for forecasts.

5. (5 pts) Using the model you feel is most useful to forecasting the next 5 weeks of respiratory mortality

I have picked the ensemble multivariate model of VAR and MLR correlated errors for forecasting the next 5 weeks of respiratory mortality. The decision to use this model was because of the lowest ASE of 1.6372.

```
# Predict next 5 weeks of rmort using VAR
CMortVAR_5 = VAR(cbind(RMFullDF$rmort, RMFullDF$part, RMFullDF$temp), season = 52, type = "both", p = 1)
# Forecast next 30 weeks
pred_5 = predict(CMortVAR_5, n.ahead = 5)
pred_5$fcst$RMFullDF.rmort
```

```
##           fcst      lower      upper      CI
## [1,]  9.050455033  5.674423806 12.42648626  3.376031226
## [2,]  9.210646091  5.137254697 13.28403749  4.073391394
## [3,]  8.873076461  4.507149272 13.23900365  4.365927189
## [4,]  8.980878920  4.482409424 13.47934842  4.498469496
## [5,] 10.232032072  5.671821479 14.79224267  4.560210593
```

```
# display the five forecasts
pred_5$fcst$RMFullDF.rmort[,1]
```

```
## [1]  9.050455033  9.210646091  8.873076461  8.980878920 10.232032072
```

```
# Predit next 5 weeks of rmort using MLR
#load the forecasted Part and Temp in a data frame
next5 = data.frame(temp = predsTemp$f[1:5], part = predsPart$f[1:5], Week = seq(509,513,1))
#get predictions
predsRMort_5 = predict(fit,newxreg = next5)
# display the five forecasts
predsRMort_5$pred
```

```
## Time Series:
## Start = 479
## End = 483
## Frequency = 1
## [1] 9.425600406 9.045270109 9.545543471 9.142002861 9.445441198
```

```
# Create an ensemble of these 5 forecasts
ensemble5_forecast = (pred_5$fcst$RMFullDF.rmort[,1] + predsRMort_5$pred)/2
# Display the next 5 weeks of respiratory mortality forecasts
ensemble5_forecast
```

```
## Time Series:
## Start = 479
## End = 483
## Frequency = 1
## [1] 9.238027719 9.127958100 9.209309966 9.061440891 9.838736635
```

Honor Code:

I, Tej Tenmattam, abided by the SMU Honor Code and did not communicate about the content of this exam with anyone except for my professor Dr. Bivin Sadler.