MSDS_6373_Final

Jeffrey Lancon

August 17, 2019

Time Series Final Take Home Question

50 pts . This analysis question uses the same data as the cardiac mortality data that we used in class (Unit 12 and 13). It is from the astsa package and is a dataset called lap. Your overall goal is to forecast 5 weeks of respiratory mortality beyond the 508 observations you have. Please complete an analysis that covers the following requirements:

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

2. (5 pts) Comment on it's stationarity or nonstationarity. 3a. (10 pts)Perform a univariate analysis using
   a) AR,ARMA, ARIMA, and/or ARUMA models (at least one)
      i. clearly explain how you arrived at your final model
   b) using a neural network based model
   c) an ensemble model with a model from (a) and (b). 3b. (5 pts)Compare these models and describe which univariate model you feel is the best and why. 4a. (10 pts) Perform a multivariate analysis using at least one model from each category:
   a. VAR or MLR with correlated errors

i. clearly explain how you arrived at your final model
   b. mlp Be sure and use forecasted values of the predictors where appropriate.
      4b. (5pts) Fit and evaluate an ensemble model from the models you fit in 4a.
      4c. (5 pts) Compare these models and describe which multivariate model you feel is the best and why.

5. (5 pts)Using the model you feel is most useful to forecasting the next 5 weeks of respiratory mortality (Beyond the 508 observation you have.)

Please submit an R Markdown file and a knit (pdf or Word) version of your analysis. This knit document should contain well commented code and clear descriptions / explanations of your thoughts and finding. Please include the code in the knit document (even though this may make it much longer.)

```
#Exploratory Data Analysis
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.1
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

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

```
library(astsa)
library(tswge)
library(ggplot2)
library(nnfor)
```

```
## Loading required package: forecast
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

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

## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff

##
## Attaching package: 'forecast'

## The following object is masked from 'package:astsa':
##
##     gas
```

Loading Dataset

Package(astsa) - Dataset 'lap': https://cran.r-project.org/web/packages/astsa/astsa.pdf

```
# Load Dataset
data("lap")
RM = lap
RM[1:5,]
```

```
##       tmort rmort  cmort tempr    rh    co  so2   no2 hycarb   o3  part
## [1,] 183.63 11.90  97.85 72.38 29.20 11.51 3.37  9.64  45.79 6.69 72.72
## [2,] 191.05 10.75 104.64 67.19 67.51  8.92 2.59 10.05  43.90 6.83 49.60
## [3,] 180.09  9.33  94.36 62.94 61.42  9.48 3.29  7.80  32.18 4.98 55.68
## [4,] 184.67  9.54  98.05 72.49 58.99 10.28 3.04 13.39  40.43 9.25 55.16
## [5,] 173.60  8.27  95.85 74.25 34.80 10.57 3.39 11.90  48.53 9.15 66.02
```

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

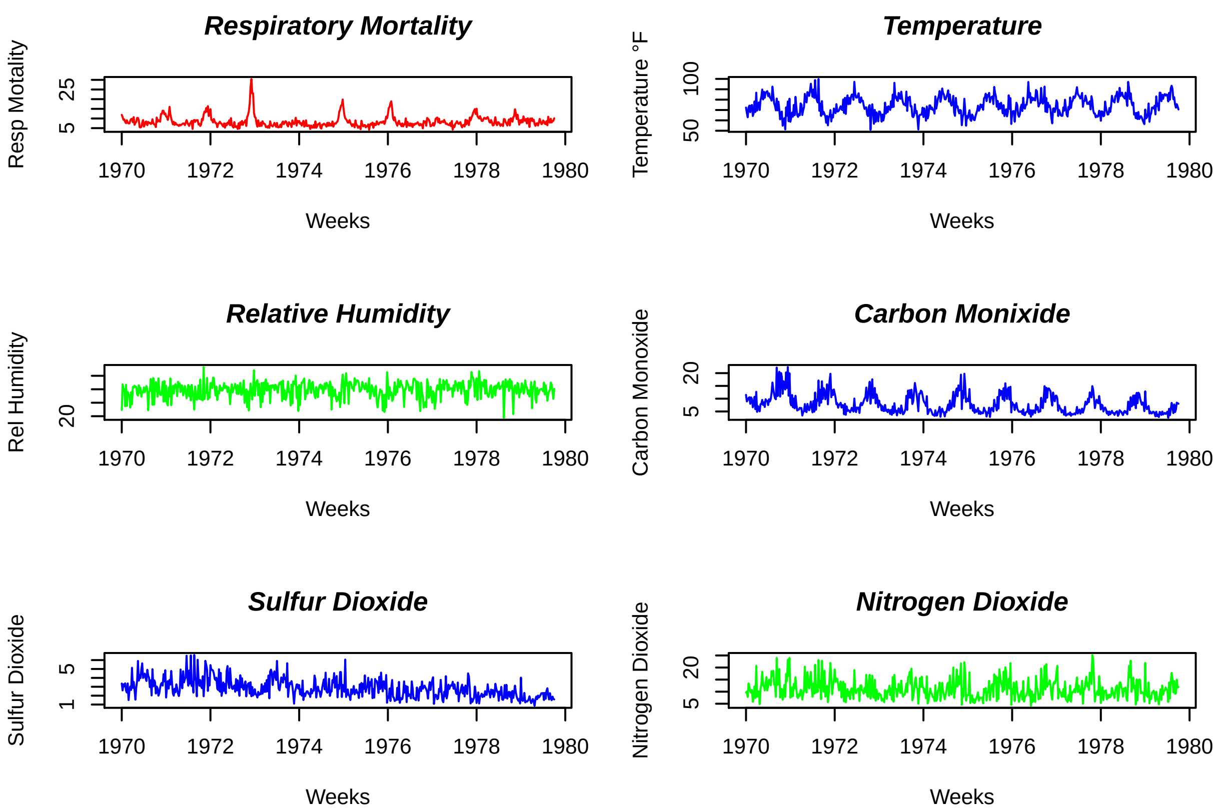
Plotting Realization of Dataset; 11 variables, 508 observations, Data Gathered on a weekly basis.

Variables:

Respiratory Mortality(Target), Temperature, Relative Humidity, Carbon Monoxide,Sulfur Dioxide,Nitrogen Dioxide

```
# Plottting original Data
par(mfrow=c(3,2))
plot(RM[,2],type='l',main="Respiratory Mortality",font.main=4,
     col='red',
     xlab="Weeks",
     ylab="Resp Motality")
plot(RM[,4],type='l',main="Temperature",font.main=4,
     col='Blue',
     xlab="Weeks",
     ylab="Temperature Â°F")
plot(RM[,5],type='l',main="Relative Humidity",font.main=4,
     col='green',
     xlab="Weeks",
```

```
     ylab="Rel Humidity")
plot(RM[,6],type='l',main="Carbon Monixide",font.main=4,
     col='blue',
     xlab="Weeks",
     ylab="Carbon Monoxide")
plot(RM[,7],type='l',main="Sulfur Dioxide",font.main=4,
     col='blue',
     xlab="Weeks",
     ylab="Sulfur Dioxide")
plot(RM[,8],type='l',main="Nitrogen Dioxide",font.main=4,
     col='green',
     xlab="Weeks",
     ylab="Nitrogen Dioxide")
```

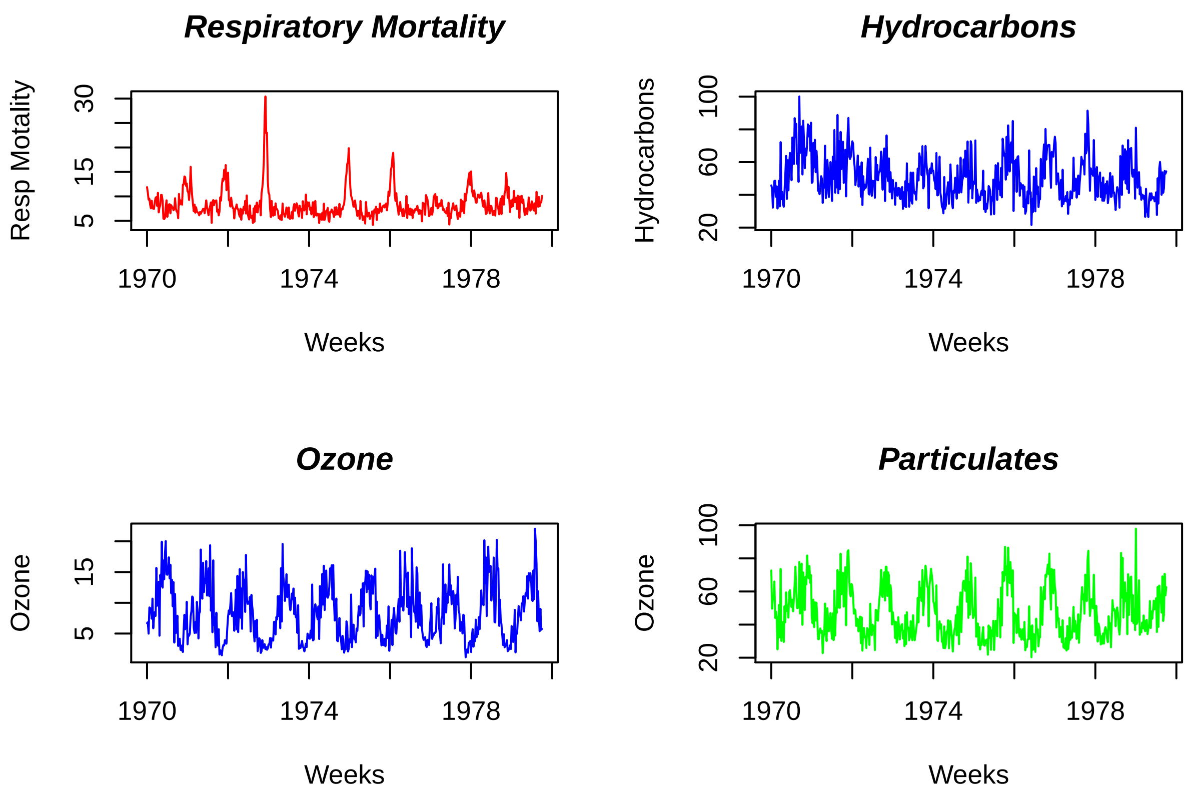


Remaining Variables:

Respiratory Mortality(Target), Hydrocarbons, Ozone, Particulates

```
par(mfrow=c(2,2))
plot(RM[,2],type='l',main="Respiratory Mortality",font.main=4,
     col='red',
     xlab="Weeks",
     ylab="Resp Motality")
plot(RM[,9],type='l',main="Hydrocarbons",font.main=4,
     col='blue',
     xlab="Weeks",
```

```r
      ylab="Hydrocarbons")
plot(RM[,10],type='l',main="Ozone",font.main=4,
      col='blue',
      xlab="Weeks",
      ylab="Ozone")
plot(RM[,11],type='l',main="Particulates",font.main=4,
      col='green',
      xlab="Weeks",
      ylab="Ozone")
```



Plotting of Repiratory Variable;

Plotting Respiratory Mortality (RM) Realization, ACF, and Spectral Density

RM Data appears to have a yearly frequency (52 segments), evident from the ACF and Spectral Density plots.

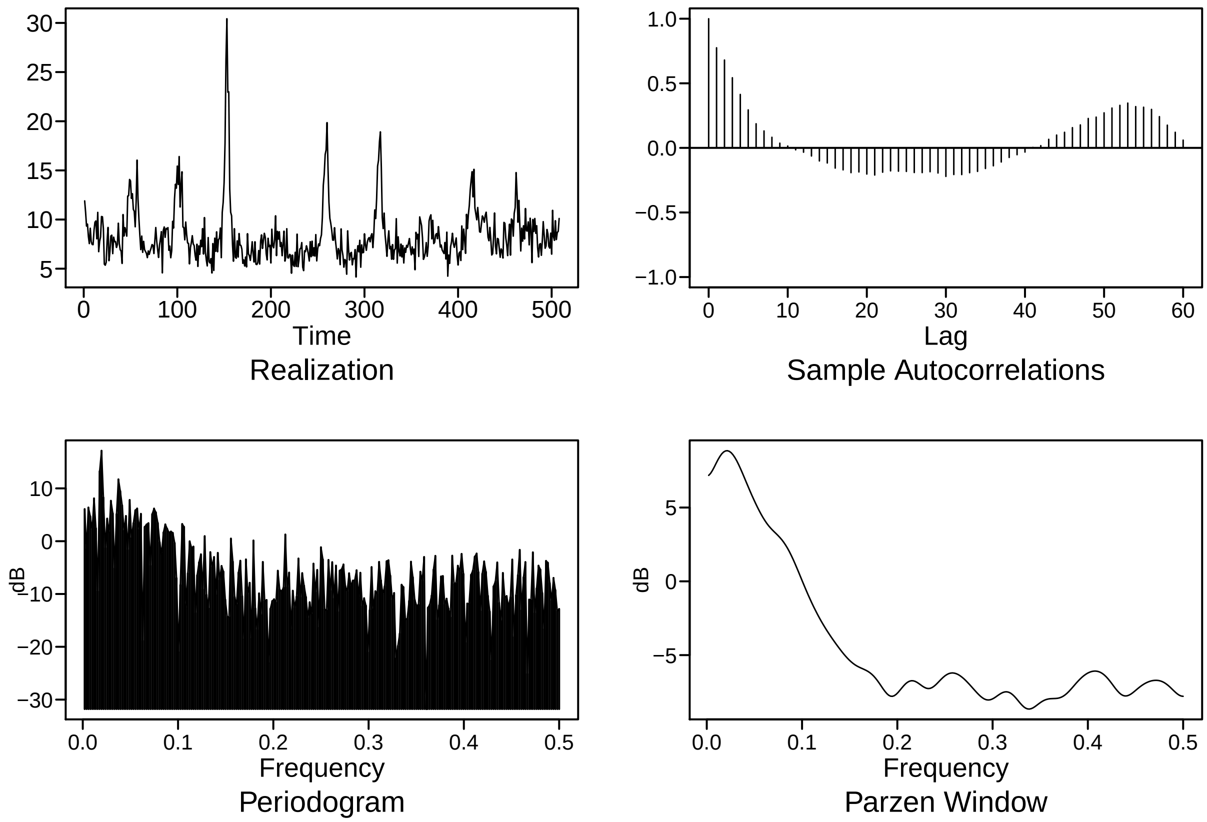
2. (5 pts) Comment on it's stationarity or nonstationarity.

RM data appears to be nonstationary.

- Constant mean and variance are suspect with this realization
- There appears to be a seasonal component within the data as well

RM data also shows some peaks that do not follow a set pattern.

```r
RMplot = plotts.sample.wge(RM[,2],lag.max = 60)
```

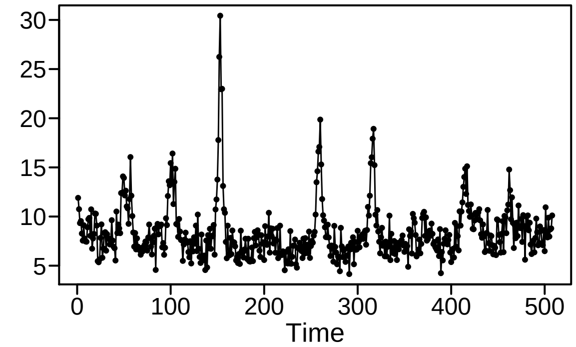


3a.a) Perform a univariate analysis using; AR,ARMA, ARIMA, and/or ARUMA models (at least one)

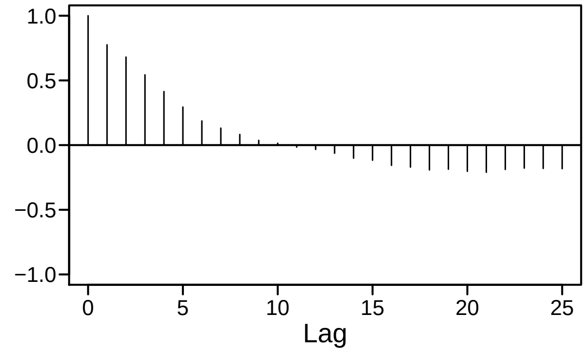
i. clearly explain how you arrived at your final model

Since RM data appears to have a seasonal component, We first take the difference of the data with a seasonal factor of 52 (1 year)

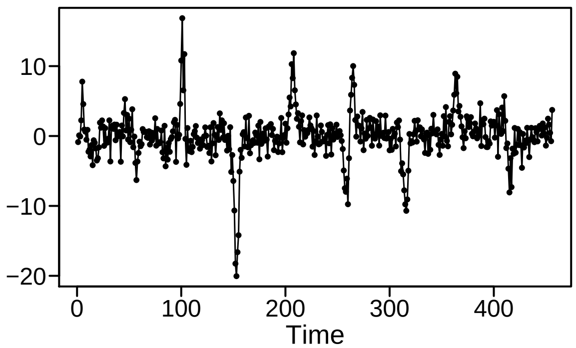
We will then plot the seasonally transformed data RM_52 and inspect it.



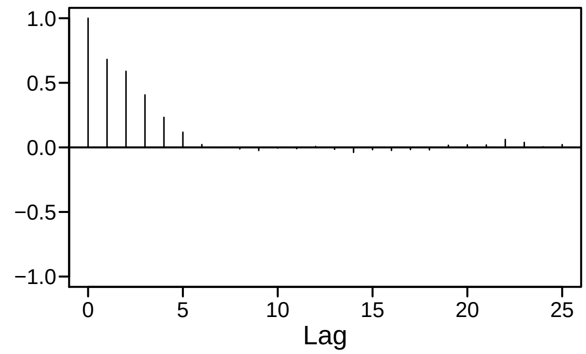
Realization



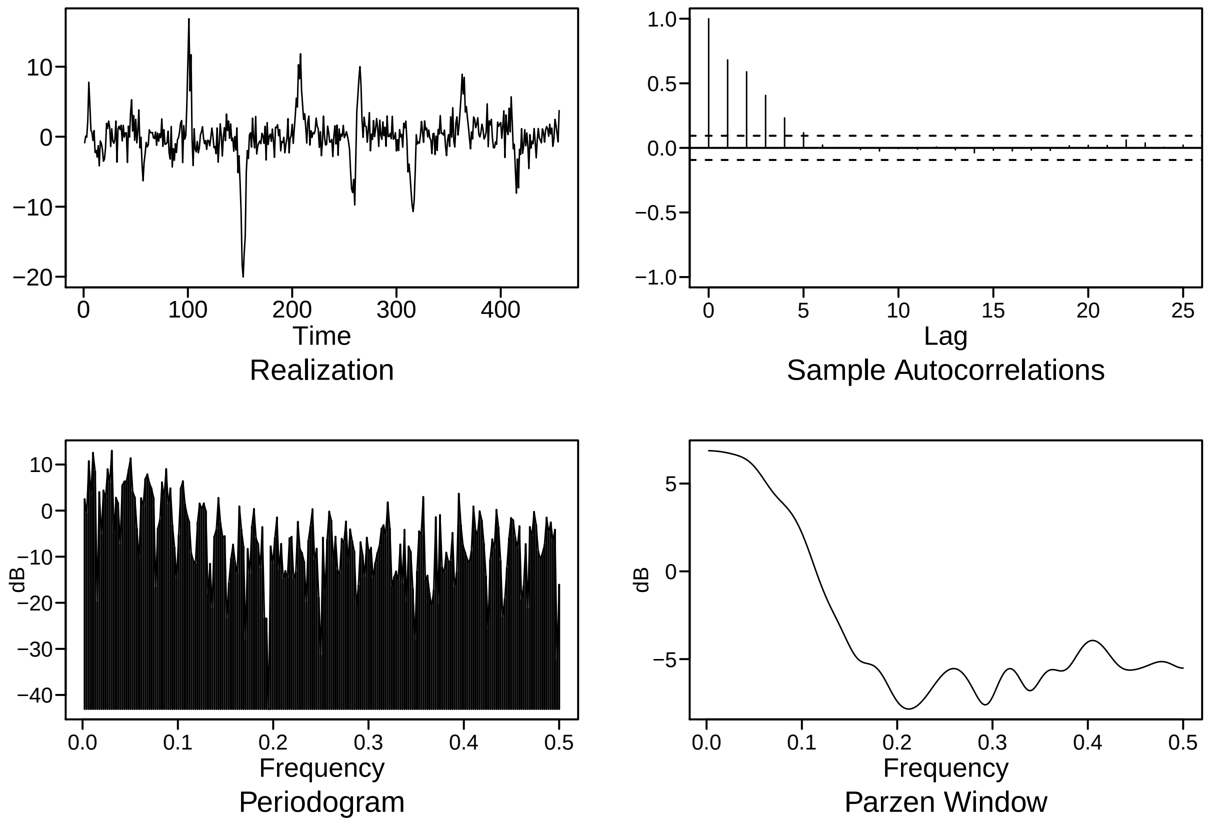
Sample Autocorrelations



Transformed Realization



Sample Autocorrelations




```
## $autplt
##  [1]  1.0000000000  0.6810568463  0.5890416208  0.4064426989  0.2323226891
##  [6]  0.1174714212  0.0217395074 -0.0005415251 -0.0120764810 -0.0241252441
## [11] -0.0056709759 -0.0097017187  0.0072606585 -0.0153320411 -0.0385652385
## [16] -0.0175204465 -0.0234651663 -0.0165722205 -0.0192526412  0.0162258012
## [21]  0.0197005729  0.0187527524  0.0612789218  0.0384650762  0.0045356733
## [26]  0.0213662127
##
## $freq
##   [1] 0.002192982 0.004385965 0.006578947 0.008771930 0.010964912
##   [6] 0.013157895 0.015350877 0.017543860 0.019736842 0.021929825
##  [11] 0.024122807 0.026315789 0.028508772 0.030701754 0.032894737
##  [16] 0.035087719 0.037280702 0.039473684 0.041666667 0.043859649
##  [21] 0.046052632 0.048245614 0.050438596 0.052631579 0.054824561
##  [26] 0.057017544 0.059210526 0.061403509 0.063596491 0.065789474
##  [31] 0.067982456 0.070175439 0.072368421 0.074561404 0.076754386
##  [36] 0.078947368 0.081140351 0.083333333 0.085526316 0.087719298
##  [41] 0.089912281 0.092105263 0.094298246 0.096491228 0.098684211
##  [46] 0.100877193 0.103070175 0.105263158 0.107456140 0.109649123
##  [51] 0.111842105 0.114035088 0.116228070 0.118421053 0.120614035
##  [56] 0.122807018 0.125000000 0.127192982 0.129385965 0.131578947
##  [61] 0.133771930 0.135964912 0.138157895 0.140350877 0.142543860
##  [66] 0.144736842 0.146929825 0.149122807 0.151315789 0.153508772
##  [71] 0.155701754 0.157894737 0.160087719 0.162280702 0.164473684
##  [76] 0.166666667 0.168859649 0.171052632 0.173245614 0.175438596
##  [81] 0.177631579 0.179824561 0.182017544 0.184210526 0.186403509
```

```
##  [86] 0.188596491 0.190789474 0.192982456 0.195175439 0.197368421
##  [91] 0.199561404 0.201754386 0.203947368 0.206140351 0.208333333
##  [96] 0.210526316 0.212719298 0.214912281 0.217105263 0.219298246
## [101] 0.221491228 0.223684211 0.225877193 0.228070175 0.230263158
## [106] 0.232456140 0.234649123 0.236842105 0.239035088 0.241228070
## [111] 0.243421053 0.245614035 0.247807018 0.250000000 0.252192982
## [116] 0.254385965 0.256578947 0.258771930 0.260964912 0.263157895
## [121] 0.265350877 0.267543860 0.269736842 0.271929825 0.274122807
## [126] 0.276315789 0.278508772 0.280701754 0.282894737 0.285087719
## [131] 0.287280702 0.289473684 0.291666667 0.293859649 0.296052632
## [136] 0.298245614 0.300438596 0.302631579 0.304824561 0.307017544
## [141] 0.309210526 0.311403509 0.313596491 0.315789474 0.317982456
## [146] 0.320175439 0.322368421 0.324561404 0.326754386 0.328947368
## [151] 0.331140351 0.333333333 0.335526316 0.337719298 0.339912281
## [156] 0.342105263 0.344298246 0.346491228 0.348684211 0.350877193
## [161] 0.353070175 0.355263158 0.357456140 0.359649123 0.361842105
## [166] 0.364035088 0.366228070 0.368421053 0.370614035 0.372807018
## [171] 0.375000000 0.377192982 0.379385965 0.381578947 0.383771930
## [176] 0.385964912 0.388157895 0.390350877 0.392543860 0.394736842
## [181] 0.396929825 0.399122807 0.401315789 0.403508772 0.405701754
## [186] 0.407894737 0.410087719 0.412280702 0.414473684 0.416666667
## [191] 0.418859649 0.421052632 0.423245614 0.425438596 0.427631579
## [196] 0.429824561 0.432017544 0.434210526 0.436403509 0.438596491
## [201] 0.440789474 0.442982456 0.445175439 0.447368421 0.449561404
## [206] 0.451754386 0.453947368 0.456140351 0.458333333 0.460526316
## [211] 0.462719298 0.464912281 0.467105263 0.469298246 0.471491228
## [216] 0.473684211 0.475877193 0.478070175 0.480263158 0.482456140
## [221] 0.484649123 0.486842105 0.489035088 0.491228070 0.493421053
## [226] 0.495614035 0.497807018 0.500000000
## 
## $db
##   [1]   2.4874137  -0.6526187  10.7325977   3.1102937  12.5437887
##   [6]   8.4960230 -19.9011776   4.0162700  -5.3233692   4.3732408
##  [11]   2.7446155   8.9892856   6.0545974  12.9982409  -4.9790994
##  [16]   2.8059221   1.6319722  -7.4521691   5.4898126   6.3227793
##  [21]   6.3960293   8.8198396  11.3633094   4.0417067   2.8866346
##  [26]  -3.9033177 -11.0008360   2.7099300   1.0614104   6.8794807
##  [31]   7.8774425   5.9479310   4.8040730   2.7498405 -16.6763456
##  [36]  -3.8678808  -2.0328618   6.1419198   2.8844994   9.0184141
##  [41]   1.2984728   4.8410246  -3.1912046  -8.0316088 -14.8339949
##  [46]  -5.3283119   4.8789200   6.3886967   1.4098012  -0.9249240
##  [51]  -2.4122362  -9.2800023 -10.8436786 -11.6122578  -2.5600458
##  [56]   1.5453843   0.4039441   1.5718416  -0.1096929 -18.5646219
##  [61] -11.5359715 -21.2164254  -5.6174267  -4.1551956   2.7533757
##  [66]  -2.7389437  -5.6601037  -5.5494316 -23.4537270 -15.7330545
##  [71] -10.4716714  -7.3779618 -11.1839557 -13.7537928   0.9049521
##  [76]  -4.1435073  -7.8090259 -28.1484686  -8.3028637 -13.7301034
##  [81]  -3.4222220   0.3696649  -6.0053657  -7.2495777 -12.5318009
##  [86]  -3.5773831 -23.3110406 -23.4322862 -43.0580045  -7.7684863
##  [91] -11.5188073  -5.9578461  -1.4821151 -13.0625244  -7.2057770
##  [96] -14.7466161 -13.1055252 -14.7003693  -6.0140590  -5.6875110
## [101] -14.9020996 -13.3830492  -2.4217417  -8.3270170  -9.2187195
## [106] -11.1058583 -19.8939615  -6.5511029  -3.4315093   0.3073733
## [111] -10.8144500  -8.2641780 -18.8995310 -31.4269764  -5.8679644
```

```
## [116] -18.6554718  -6.3474786  -0.2156250  -2.1344884 -11.3638939
## [121]  -5.6662777 -12.9321423 -13.7692230  -6.1111542  -7.1226993
## [126]  -2.3647829 -11.3586223  -4.0525657  -6.6400049  -8.8417431
## [131] -20.8161976 -16.2936991  -6.8414528  -9.5832191 -14.3872539
## [136]  -5.8666557  -9.1608140  -8.0452210 -14.7047015 -11.4122615
## [141]  -9.2329809  -7.7143604  -3.9487838  -3.1515767  -5.4128210
## [146]   1.7970242  -3.8652648 -16.5156516 -14.4588350 -13.1924375
## [151] -15.2816562  -7.3414295 -15.8803000  -4.1545198 -20.0841313
## [156]  -7.7585334  -8.9618023 -16.9750966 -28.3234004 -13.2264609
## [161]  -4.4703010  -5.1415411   2.9684223 -15.2327781 -14.1006352
## [166] -17.9195181 -20.9126818 -17.2409495  -1.4180333 -20.4427974
## [171]  -0.9559059 -12.3188402 -13.4496552  -9.1120941 -11.0030055
## [176] -11.1109662  -4.8278439 -16.7794380 -13.4666806   3.6700460
## [181]  -3.1244469  -7.5197690  -9.0488244 -10.5061188 -11.3898097
## [186]  -8.8703374   0.8983446  -4.2111649  -5.2113087  -0.1451262
## [191]  -2.1262820  -7.1481026 -14.2563484 -25.9330041 -10.6971040
## [196]  -6.2180645  -8.2484220   0.1801723  -3.6348001 -10.9580755
## [201] -23.1734268 -19.2067903 -12.4619913  -6.0061093  -1.6100186
## [206]  -2.0817451  -5.7224354  -8.1462539  -3.3873209 -19.4702043
## [211] -16.3980518  -7.2524086 -21.1869239  -3.5577590  -5.6351334
## [216]  -0.2603340  -3.0923954  -8.8231862 -10.5049167  -9.2868497
## [221]  -7.4295740  -1.5118876  -4.7855602  -2.5544178  -6.6315745
## [226]  -4.1461385 -32.7008668 -16.0814443
##
## $dbz
##   [1]  6.8726925  6.8692507  6.8633919  6.8549728  6.8438635  6.8299950
##   [7]  6.8133934  6.7941882  6.7725866  6.7488135  6.7230210  6.6951828
##  [13]  6.6649907  6.6317741  6.5944618  6.5515989  6.5014219  6.4419889
##  [19]  6.3713496  6.2877404  6.1897834  6.0766730  5.9483310  5.8055174
##  [25]  5.6498783  5.4839172  5.3108729  5.1344956  4.9587227  4.7872773
##  [31]  4.6232351  4.4686245  4.3241358  4.1889963  4.0610378  3.9369398
##  [37]  3.8125941  3.6835200  3.5452640  3.3937369  3.2254656  3.0377591
##  [43]  2.8288018  2.5976923  2.3444466  2.0699786  1.7760675  1.4653097
##  [49]  1.1410504  0.8072812  0.4684853  0.1294157 -0.2052028 -0.5310396
##  [55] -0.8445589 -1.1433967 -1.4266422 -1.6949504 -1.9504440 -2.1963988
##  [61] -2.4367530 -2.6755093 -2.9161034 -3.1608092 -3.4102369 -3.6629747
##  [67] -3.9154373 -4.1619942 -4.3954585 -4.6079827 -4.7923101 -4.9431865
##  [73] -5.0585941 -5.1404394 -5.1944661 -5.2294250 -5.2557735 -5.2842737
##  [79] -5.3247861 -5.3854046 -5.4719401 -5.5876797 -5.7333362 -5.9071234
##  [85] -6.1049345 -6.3206320 -6.5464807 -6.7737498 -6.9934748 -7.1973072
##  [91] -7.3783147 -7.5315602 -7.6543182 -7.7458896 -7.8070954 -7.8396274
##  [97] -7.8454485 -7.8263855 -7.7839719 -7.7195092 -7.6342675 -7.5297196
## [103] -7.4077261 -7.2706203 -7.1211871 -6.9625638 -6.7981060 -6.6312586
## [109] -6.4654598 -6.3040828 -6.1504087 -6.0076139 -5.8787565 -5.7667512
## [115] -5.6743284 -5.6039819 -5.5579112 -5.5379668 -5.5456032 -5.5818401
## [121] -5.6472261 -5.7417899 -5.8649607 -6.0154296 -6.1909238 -6.3878717
## [127] -6.6009511 -6.8225632 -7.0423430 -7.2469358 -7.4203850 -7.5455105
## [133] -7.6064535 -7.5920275 -7.4988312 -7.3327913 -7.1083598 -6.8457296
## [139] -6.5672934 -6.2945741 -6.0462487 -5.8372531 -5.6786197 -5.5776617
## [145] -5.5382080 -5.5607134 -5.6421615 -5.7757598 -5.9505182 -6.1509318
## [151] -6.3571666 -6.5462819 -6.6949410 -6.7835150 -6.8005032 -6.7454325
## [157] -6.6287474 -6.4687268 -6.2869525 -6.1041523 -5.9374483 -5.7990782
## [163] -5.6961410 -5.6308656 -5.6010669 -5.6006586 -5.6202516 -5.6479606
## [169] -5.6705556 -5.6750159 -5.6503497 -5.5893224 -5.4896058 -5.3539705
```

```
## [175] -5.1894741 -5.0059651 -4.8144020 -4.6254213 -4.4483731 -4.2908371
## [181] -4.1585107 -4.0553254 -3.9836709 -3.9446406 -3.9382498 -3.9636014
## [187] -4.0189931 -4.1019710 -4.2093442 -4.3371807 -4.4808166 -4.6349098
## [193] -4.7935773 -4.9506418 -5.0999978 -5.2360714 -5.3543094 -5.4515999
## [199] -5.5265233 -5.5793575 -5.6118286 -5.6266614 -5.6270371 -5.6160751
## [205] -5.5964380 -5.5701156 -5.5384034 -5.5020514 -5.4615319 -5.4173594
## [211] -5.3703888 -5.3220263 -5.2743062 -5.2298203 -5.1915108 -5.1623675
## [217] -5.1450758 -5.1416636 -5.1531883 -5.1794922 -5.2190541 -5.2689624
## [223] -5.3250359 -5.3821189 -5.4345584 -5.4768394 -5.5043030 -5.5138246
```

Reviewing the yearly transformed data, there still appears to be some structure to the dataset.

The next step is to determine what structure still remains in the dataset.

We will use aic5.wge() to determine the order of the ARMA models. We will use both 'bic' and 'aic'

```
#Univariate Response
aic5.wge(RM_52, p=0:10,q=0:3, type = 'bic')  # Picked a p=4:q=0
```

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

```
##       p    q        bic
## 11    2    2   1.770165
## 7     1    2   1.770671
## 17    4    0   1.774443
## 8     1    3   1.775273
## 15    3    2   1.783526
```

```
aic5.wge(RM_52, p=0:15,q=0:3, type = 'aic')  # Picked a p=4:q=0
```

```
## ---------WORKING... PLEASE WAIT...
##
##
## Error in aic calculation at 12 3
## Five Smallest Values of  aic
```

```
##       p    q        aic
## 51   12    2   1.723094
## 11    2    2   1.724963
## 55   13    2   1.725641
## 27    6    2   1.726538
## 17    4    0   1.729240
```

Reviewing the results of the aic5 function, for both the BIC and AIC, we have chosen to model an ARMA(4,0)

To find the estimated parameters for the transformed Respiratory Data, we use the est.arma.wge() function from tswge.

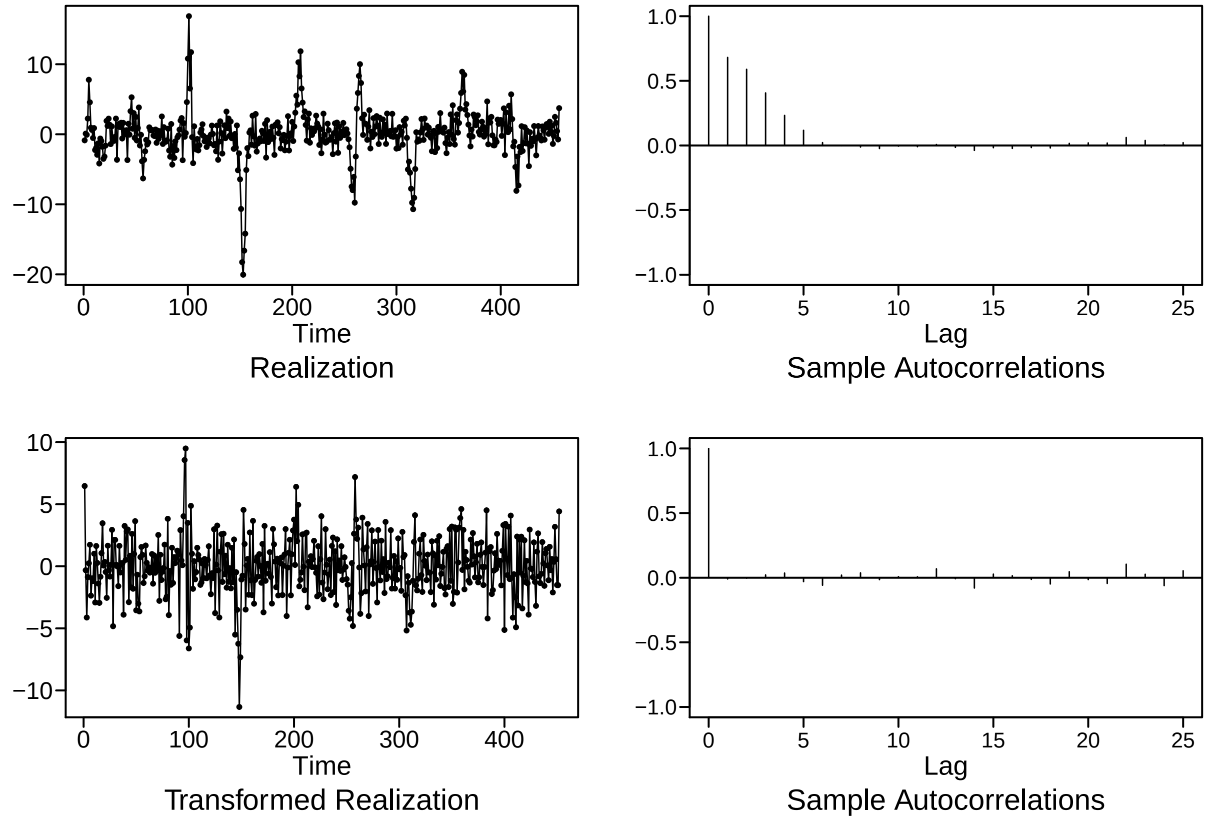
```
est = est.arma.wge(RM_52, p = 4, q = 0)
```

```
##
## Coefficients of Original polynomial:
## 0.5300 0.3445 -0.0265 -0.1680
##
## Factor                 Roots                Abs Recip    System Freq
## 1-1.3895B+0.5368B^2    1.2942+-0.4334i      0.7327       0.0514
## 1+0.8595B+0.3129B^2   -1.3732+-1.1445i      0.5594       0.3894
```

```
##
##
```

We will then transform the data once again, using the phi's determined in the previous step. We will call this realization RM_52_AR4_MA0.

```r
RM_52_AR4_MA0 = artrans.wge(RM_52,phi = est$phi)
```



The transformed realization RM_52_AR4_MA0 does appear to have no remaining structure. We will test to see if there is structure remaining using ljung.wge() function, using both K=24 & 48. H0:

```r
ljung.wge(RM_52_AR4_MA0)
```

```
## Obs -0.009742107 -0.00303301 0.02113635 0.03595356 -0.03079552 -0.05801608 0.01998781 0.03700051 -0.0

## $test
## [1] "Ljung-Box test"
##
## $K
## [1] 24
##
## $chi.square
## [1] 20.10188
##
## $df
## [1] 24
##
## $pval
## [1] 0.6909678
```

```r
ljung.wge(RM_52_AR4_MA0,K=48)
```

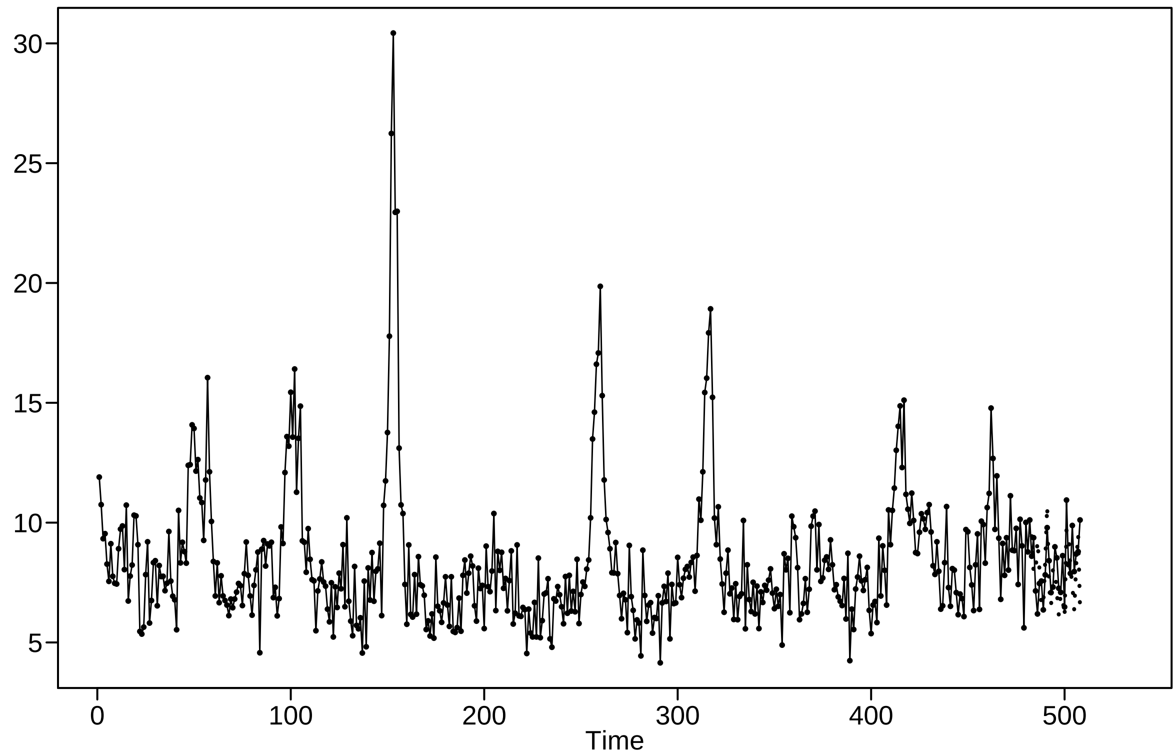
```
## Obs -0.009742107 -0.00303301 0.02113635 0.03595356 -0.03079552 -0.05801608 0.01998781 0.03700051 -0.0
```

```
## $test
## [1] "Ljung-Box test"
##
## $K
## [1] 48
##
## $chi.square
## [1] 33.4218
##
## $df
## [1] 48
##
## $pval
## [1] 0.9454828
```

At 95% confidence (alpha=0.05), We fail to reject the null hypothesis of residual data is white noise (b=0), K=24; p-value 0.691 and K=48; p-value 0.945 There appears to be no structure remaining in the transformed data

To compare models, we will go ahead and estimate the last 26 observations of the Respiratory Mortality Data and use ASE as the metric to determine the preferred model.

```r
RM_52_AR4_MA0.fore=fore.aruma.wge(RM[,2],phi = est$phi, s = 52, n.ahead = 26,lastn = TRUE,limits = FALSE
```



```
RM_52_AR4_MA0.fore$f
```

```
##  [1]  9.462481  7.896135  7.920604  9.249472  8.056831  6.501972  6.604407
##  [8]  8.397396 10.713796  7.315646  6.524392  8.083979  8.019030  7.076166
## [15]  6.155111  7.015445  6.826166  6.077173  9.708123  9.618898  8.129488
## [22]  7.399874  6.330101  8.240209  9.530235  6.380215
```

The ASE for the ARMA(4,0) with seasonal=52 model is:

```
ASE_ARMA = mean((RM[,2][483:508] - RM_52_AR4_MA0.fore$f)^2)
print(paste0('ASE for Univariate ARUMA(4,0,0) s=52 Model: ',ASE_ARMA))
```

```
## [1] "ASE for Univariate ARUMA(4,0,0) s=52 Model: 2.01090443427486"
```

Neural Network - Univariate

3a.b) Perform a univariate analysis using a neural network based model

We will compare our Univariate ARMA results, using ASE, and compare that with a Neural Network model.

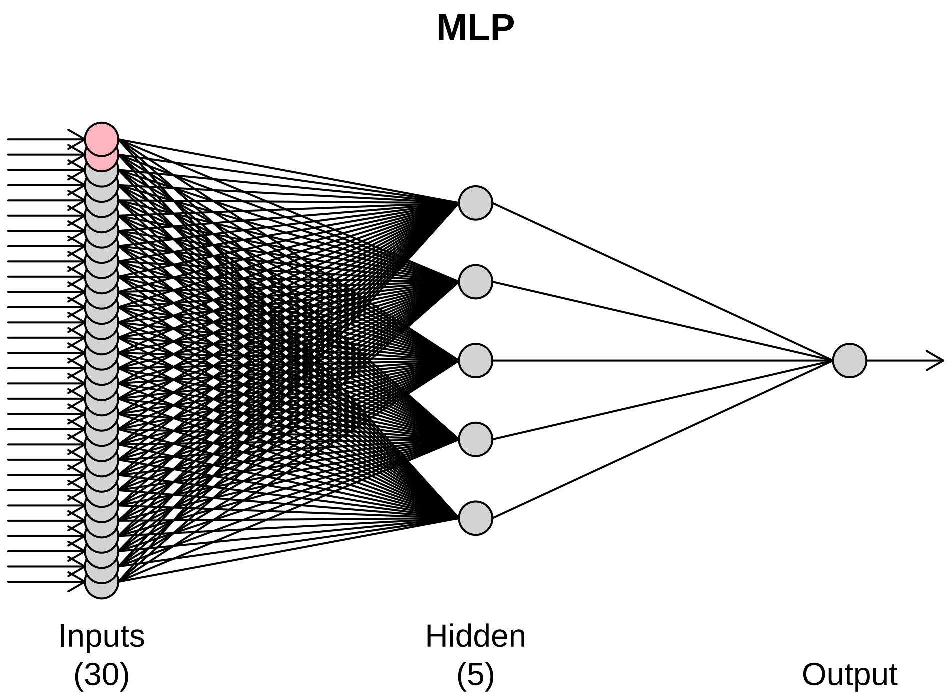
For the NN model, we will train the model on first 482 observations and predict last 26 observation, and determining the ASE for the NN models

```
set.seed(244)
RM_small = window(RM, start = c(1970, 1),end = c(1979,14))
RM.fit.mlp= mlp(RM_small[,2], reps=50,comb='mean')
RM.fit.mlp
```

```
## MLP fit with 5 hidden nodes and 50 repetitions.
## Series modelled in differences: D1.
## Univariate lags: (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,24,27,30,42,47,50)
## Deterministic seasonal dummies included.
## Forecast combined using the mean operator.
## MSE: 0.3131.
```

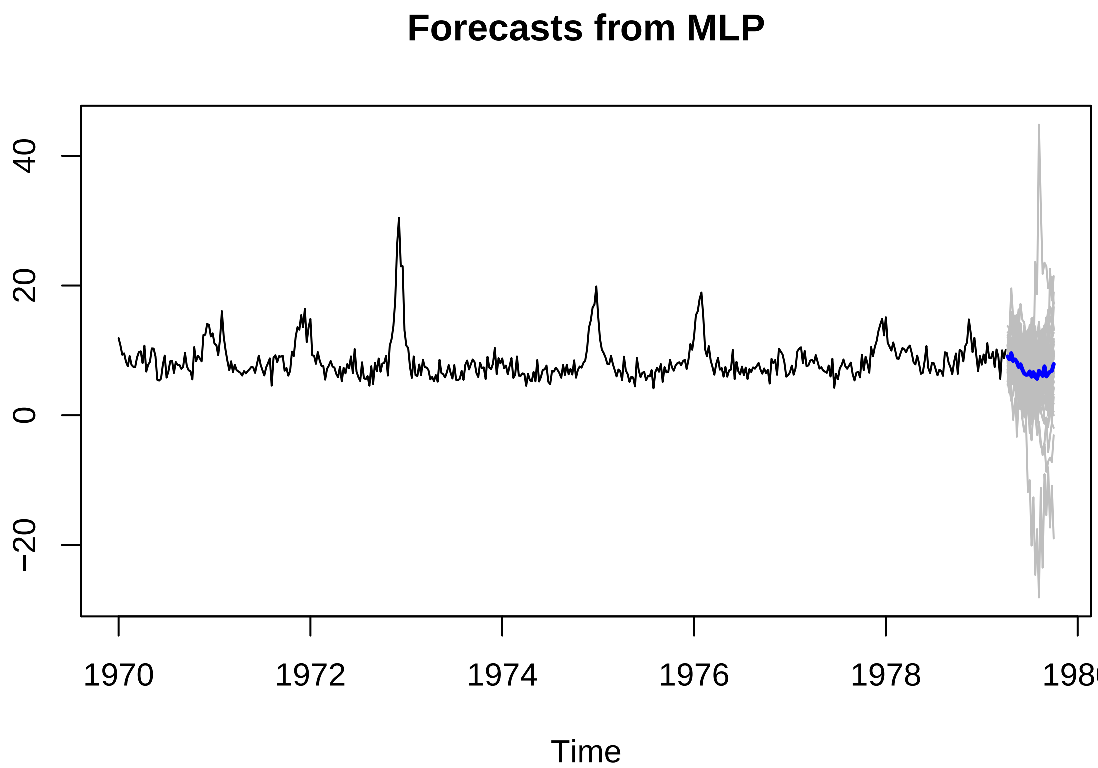
Visualize the Neural Network

```
plot(RM.fit.mlp)
```



Create a forecast for 26 segments ahead, using the Neural Network Model

```
RM.fore.mlp = forecast(RM.fit.mlp,h=26)
plot(RM.fore.mlp)
```



The ASE for the univariate NN Model is:

```
ASE_MLP = mean((RM[,2][483:508] - RM.fore.mlp$mean)^2)
print(paste0('ASE for Univariate Neural Network Model: ',ASE_MLP))
```

```
## [1] "ASE for Univariate Neural Network Model: 4.02296268739379"
```

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

Create an Ensemble method. We will create an Ensemble method utilizing the ARMA(4,0) and Neural Network Models by taking the average of the two predictions.

We will judge the model's fit using ASE as well by predicting the observations on the same 26 observations.

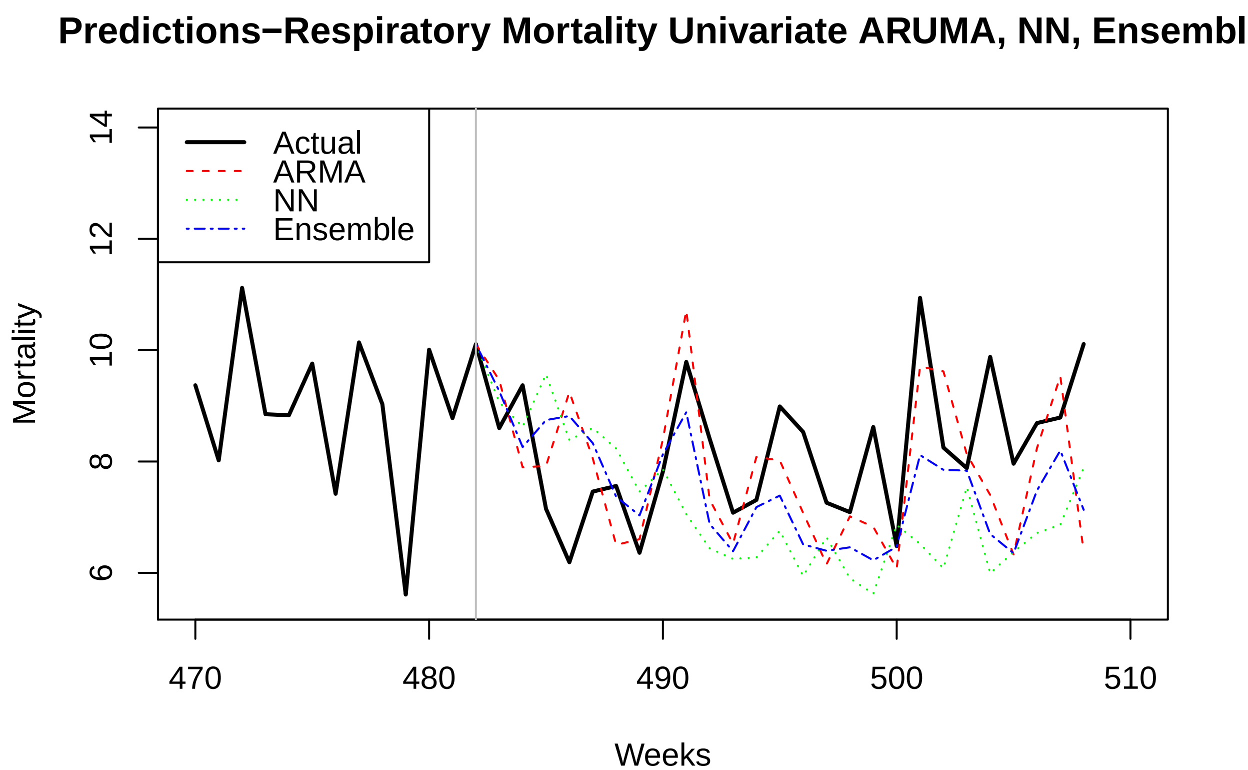
```
ensemble_uni = (RM_52_AR4_MA0.fore$f + RM.fore.mlp$mean)/2

ASE_ens = mean((RM[,2][483:508] - ensemble_uni)^2)
print(paste0('ASE for Univariate Ensemble Model: ',ASE_ens))
```

```
## [1] "ASE for Univariate Ensemble Model: 2.35343389321449"
```

Forecast Plots - ARMA, NN, Ensemble - Univariate

```
plot(seq(470,508,1),RM[470:508,2], type = "l",
     main='Predictions-Respiratory Mortality Univariate ARUMA, NN, Ensemble'
     ,xlab="Weeks"
     ,ylab="Mortality",
     xlim=c(470, 510),
     ylim=c(5.5,14),lwd=2)
```

```r
points(seq(482,508,1),c(RM[482,2],RM_52_AR4_MA0.fore$f),type='l',
       col = 'red',lwd=1,lty=2)
points(seq(482,508,1),c(RM[482,2],RM.fore.mlp$mean),type='l',
       col = 'green',pch = 17,lwd=1,lty=3)
points(seq(482,508,1),c(RM[482,2],ensemble_uni),type='l',
       col = 'blue',pch = 17,lwd=1,lty=6)
abline(v=482, col="grey",lwd=1)
legend('topleft', legend=c("Actual","ARMA", "NN","Ensemble"),
        col=c("black","red", "green","blue"), lty=c(1,2,3,6), lwd=c(2,1,1,1),y.intersp=0.75)
```



With the above plot, we are able to clearly see how the 2 univariate models (ARUMA,NN) and the Ensemble methods compare to each other:

The ARUMA(4,0,0), with a seasonality of s=52 preformed the best. The predictions tend to match the actual observation values well. The up-and-down patterns sometimes become mirrored from the actual data.

The univariate Neural Network model tends to underestimate the mortality figures across the board.

The ensemble model, since it is the average of the 2 models, and given the performance of the other models, it would be expected that it would be better than the NN but not better than the ARUMA with Seasonality.

Multivariate Analysis

4a.a) Perform a multivariate analysis using at least one model from: VAR or MLR with correlated errors

i. clearly explain how you arrived at your final model

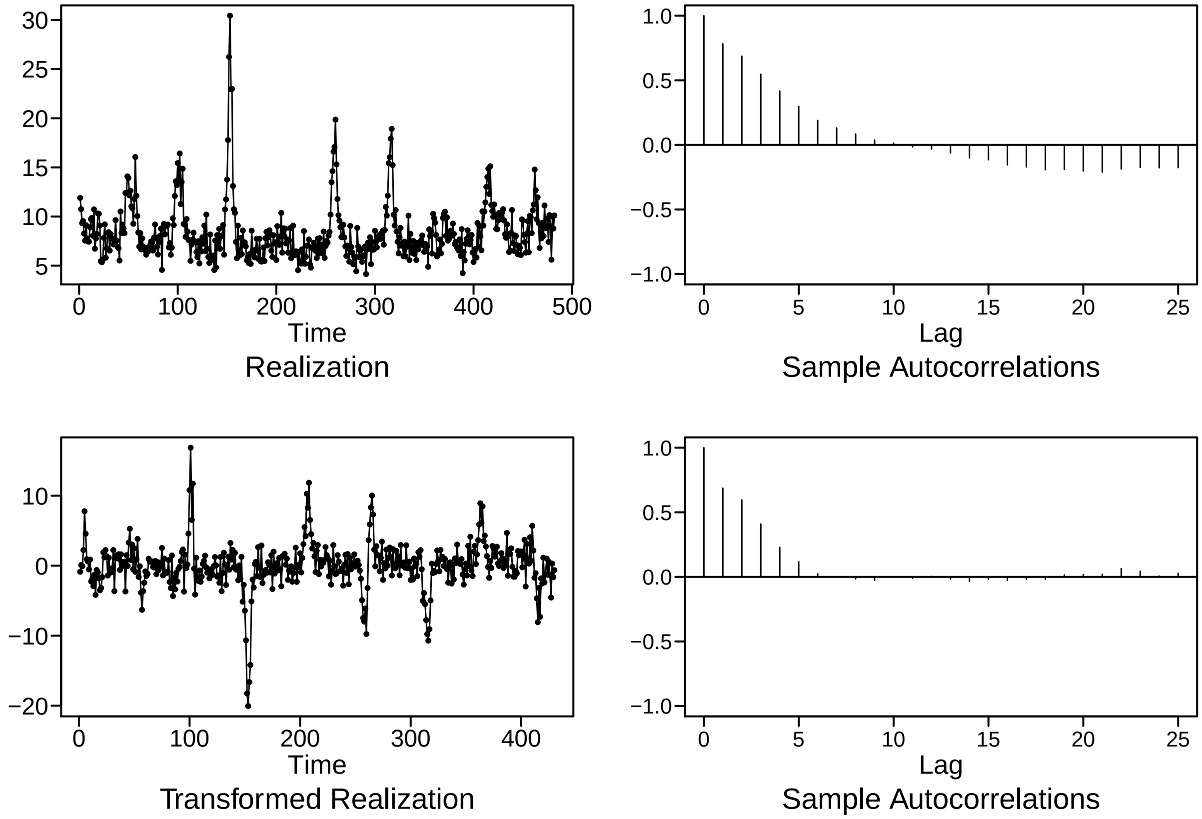
Below, we will take the lessens learned from the Univariate model and utilize the seasonality component of the data s=52. We will begin by performing artrans on all the explanitory variables as well as the target variable Respiratory Mortality.

We will also truncate the dataset to only include the first 482 observations. The last 26 will be used for prediction purposes, and model comparision, using ASE values.

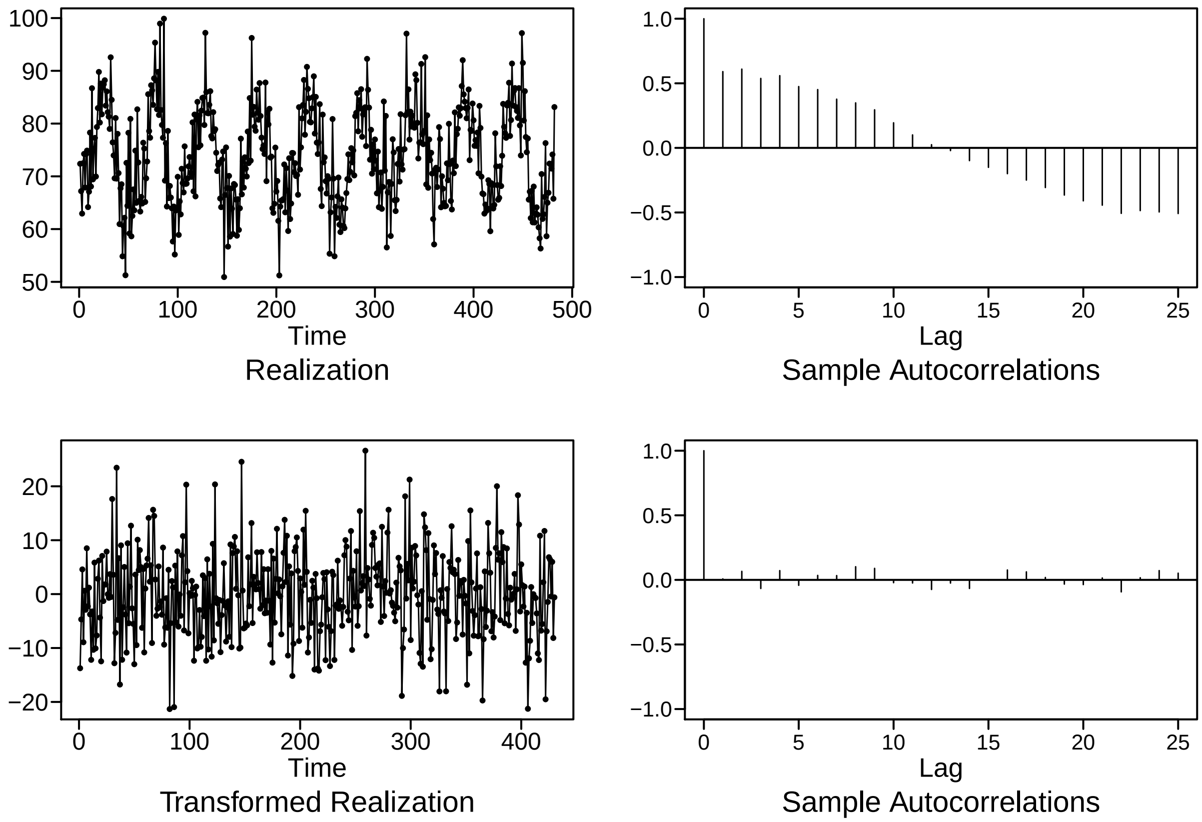
```
############ VAR MODELS ##########################

#VAR Model 1 Forecasts Seasonally Differenced Data

#Difference all series to make them stationary (assumptoin of VAR)
# Doesn't have to be white... just stationary
library(vars)
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: strucchange

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

##
## Attaching package: 'strucchange'

## The following object is masked from 'package:stringr':
##
##     boundary

## Loading required package: urca

## Loading required package: lmtest
```

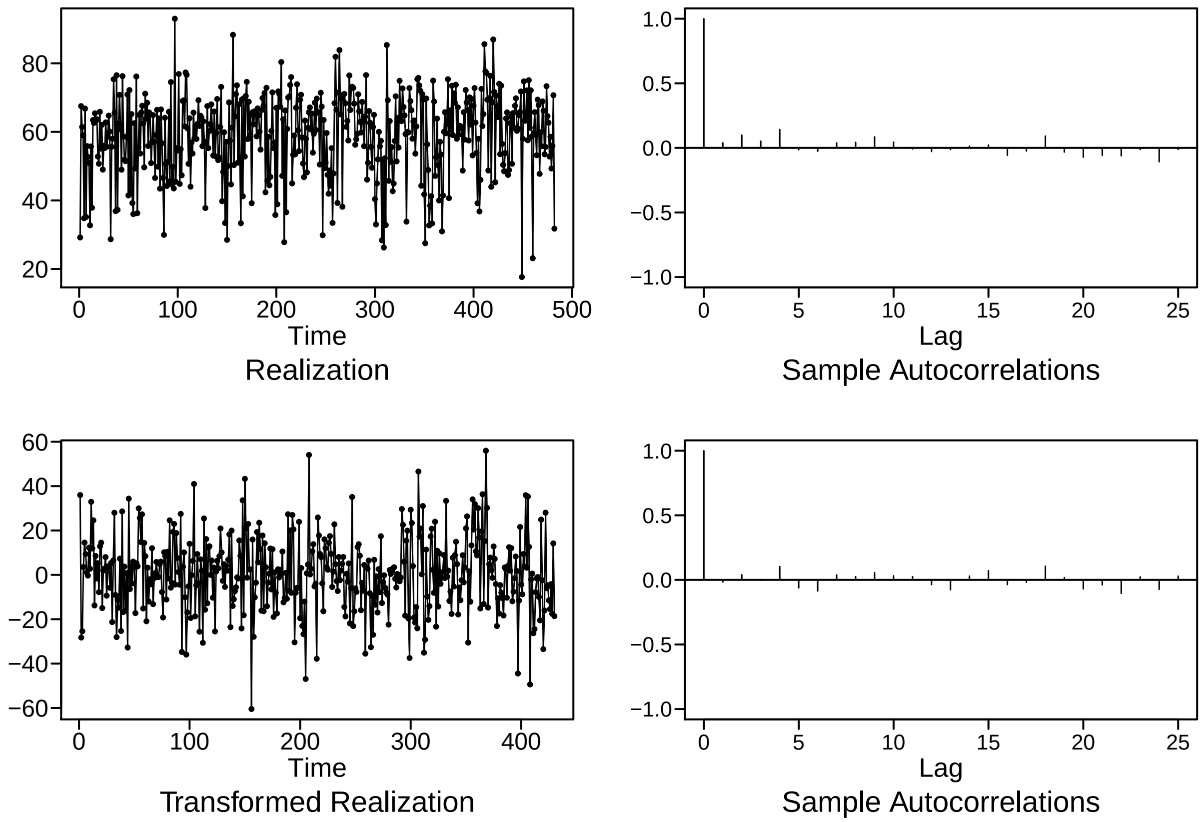
```
RM_small = window(RM, start = c(1970, 1),end = c(1979,14))
RMS_52 = artrans.wge(RM_small[,2], c(rep(0,51),1))
```



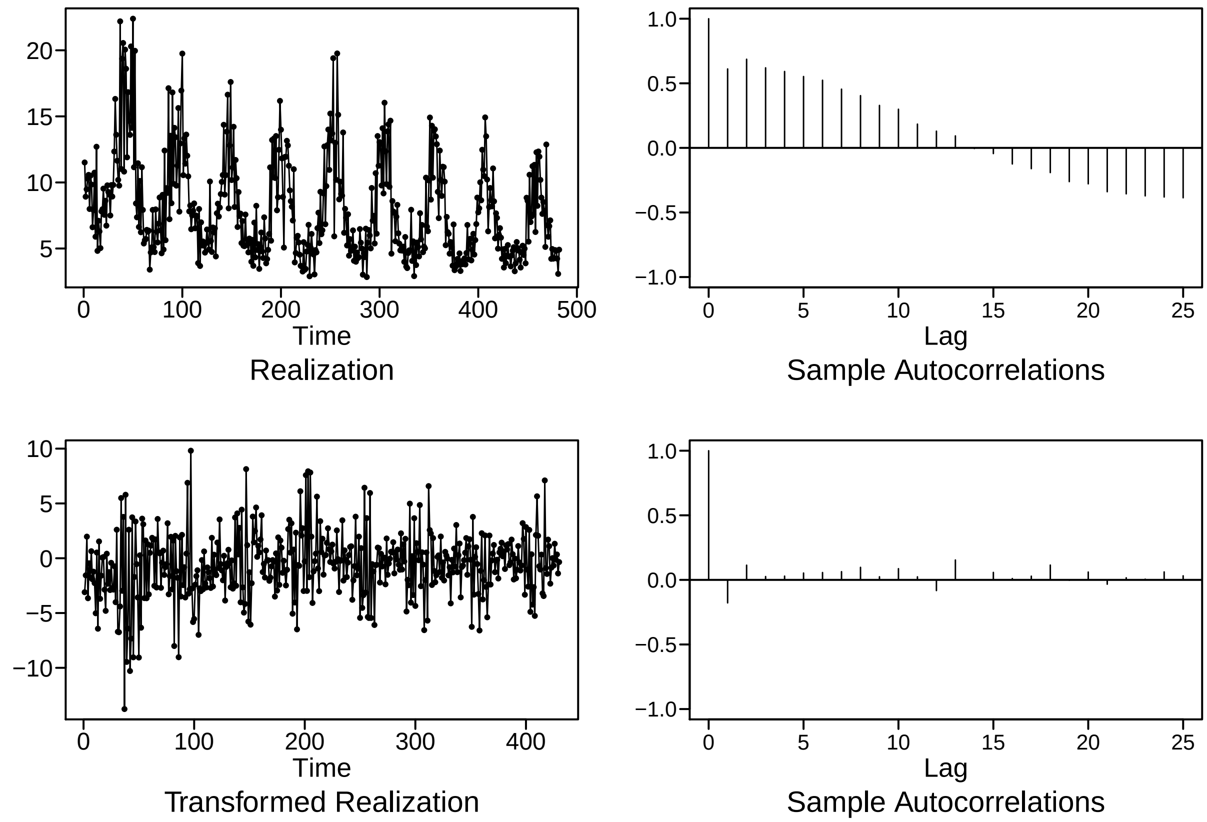

```
RMS_52_tmp = artrans.wge(RM_small[,4], c(rep(0,51),1))
```



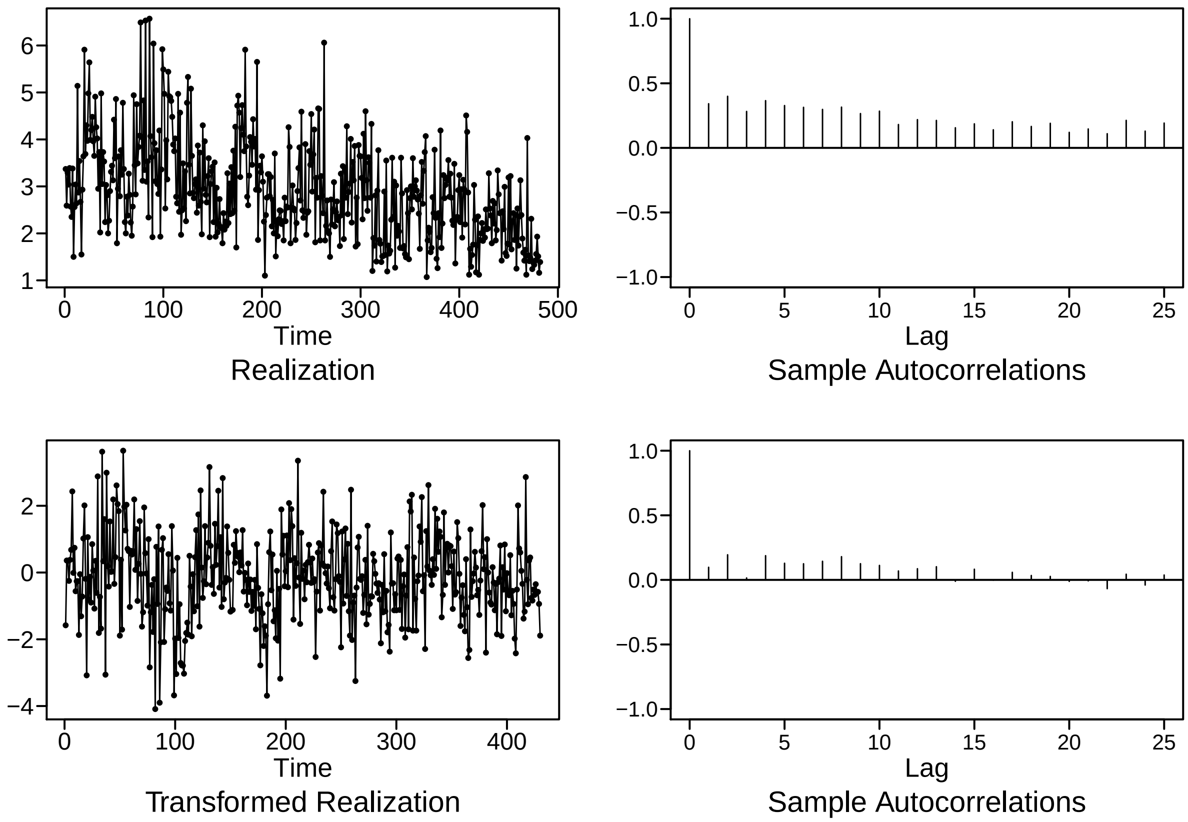

```
RMS_52_rh = artrans.wge(RM_small[,5], c(rep(0,51),1))
```



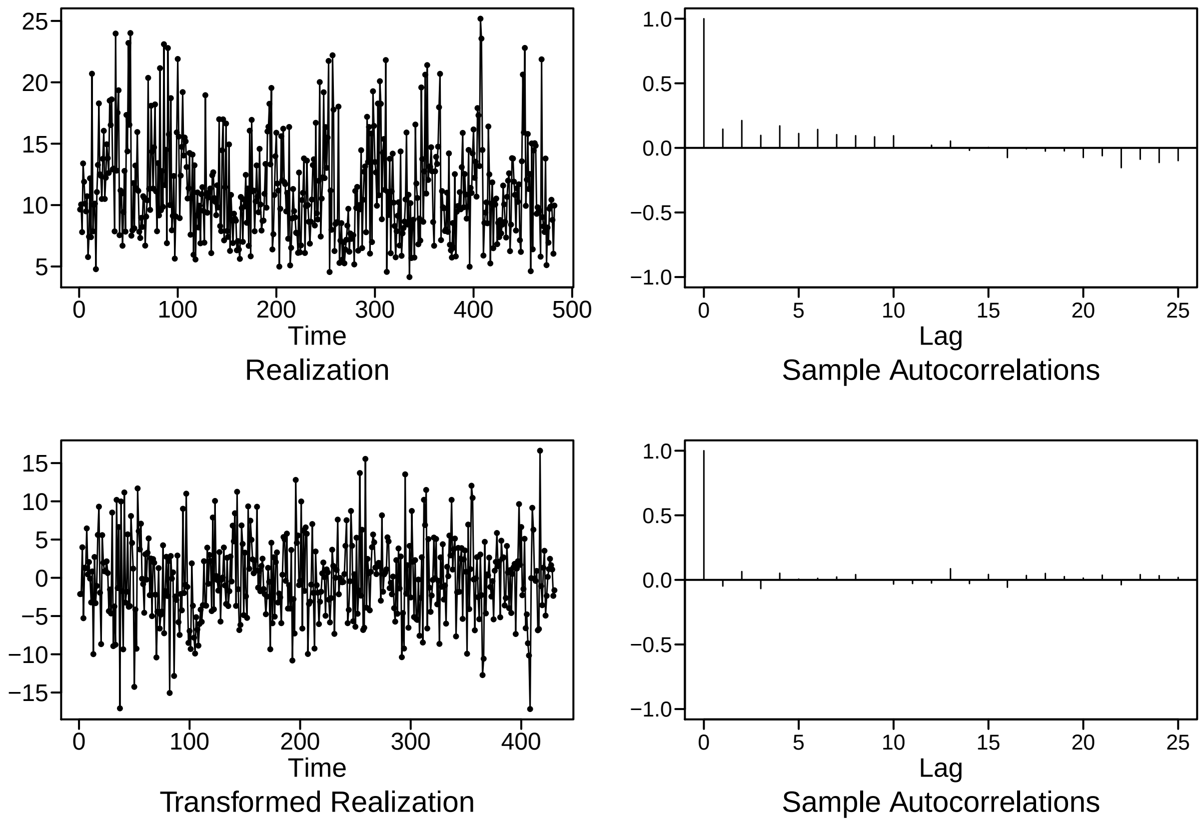

```
RMS_52_co = artrans.wge(RM_small[,6], c(rep(0,51),1))
```



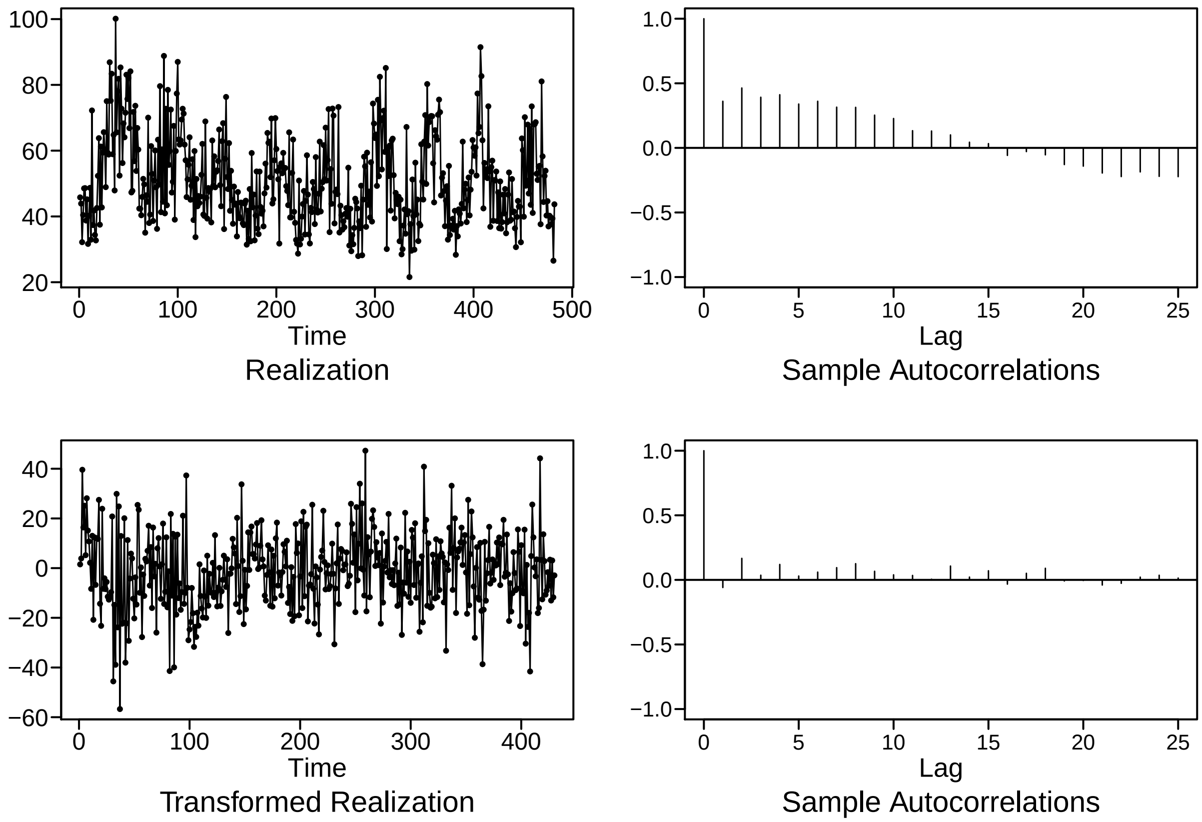

```
RMS_52_so2 = artrans.wge(RM_small[,7], c(rep(0,51),1))
```



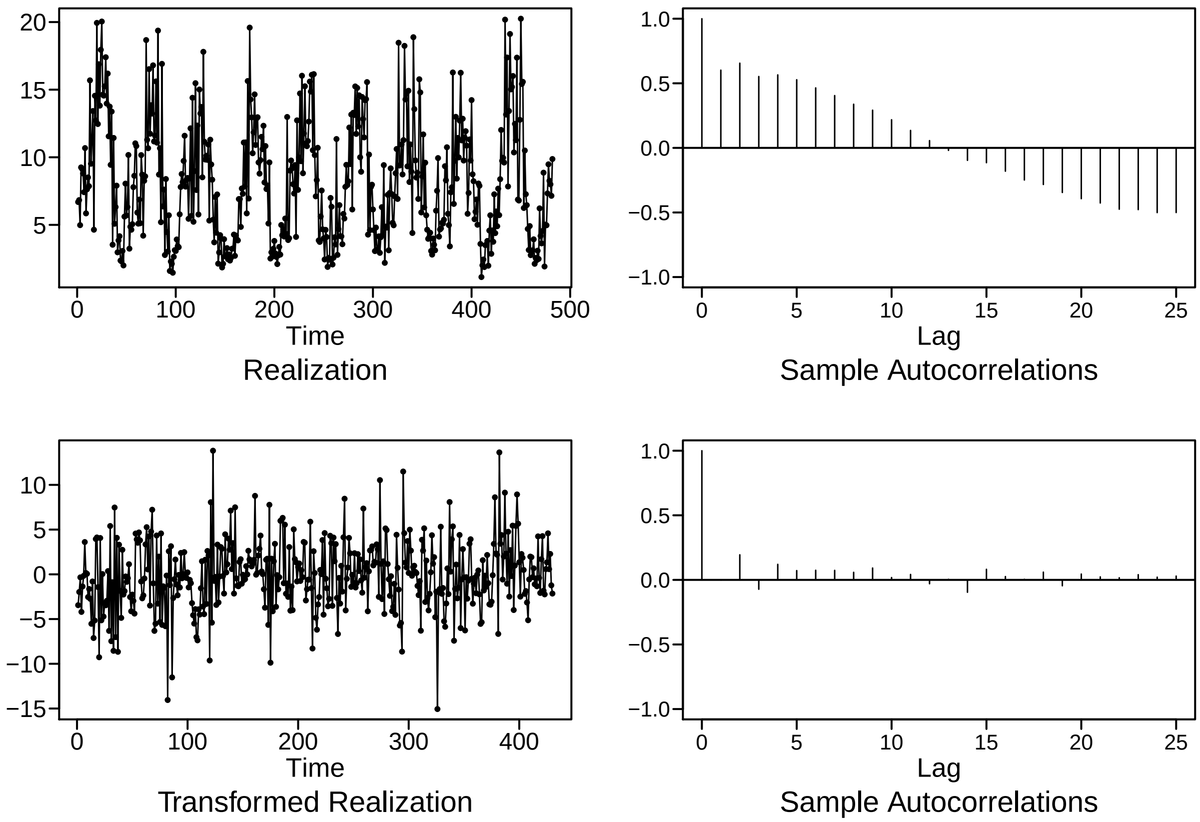

```
RMS_52_no2 = artrans.wge(RM_small[,8], c(rep(0,51),1))
```



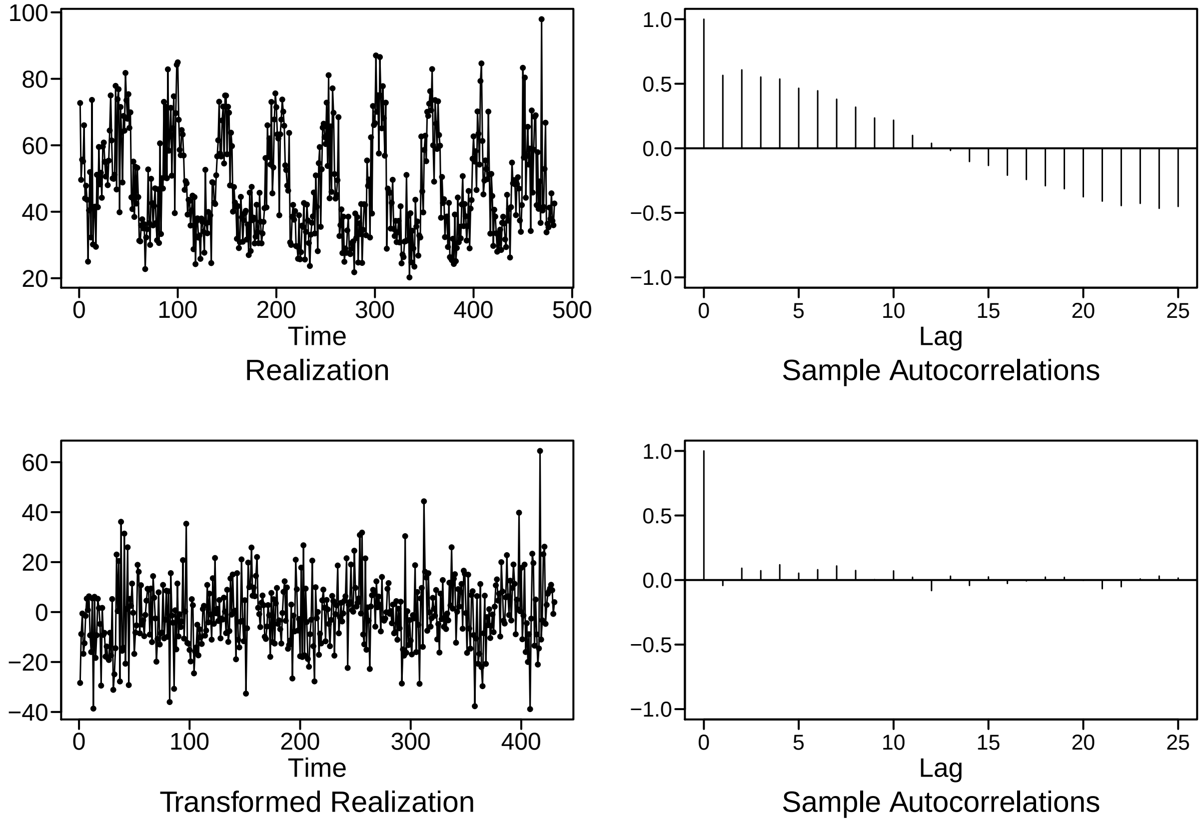

```
RMS_52_hyd = artrans.wge(RM_small[,9], c(rep(0,51),1))
```




```
RMS_52_o3 = artrans.wge(RM_small[,10], c(rep(0,51),1))
```




```
RMS_52_part = artrans.wge(RM_small[,11], c(rep(0,51),1))
```



We will use VARSelect, using multiple combinations of target and explanatory variable to determine the best combination After trying dozens of model Target & Explanatory combinations,using ASE as the metric, the model with the best ASE score ended up utilizing 1 Target and 2 Explanatory (Temperature, Particulates). The model also contains trend and constant components.

This is ironic because these are the same factors that were used for the cardiac mortality study.

```r
#VARSelect on Differenced Data chooses 1
VARselect(cbind(RMS_52,RMS_52_tmp, RMS_52_part),lag.max = 20, type = "both")
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      5      4      1      5
##
## $criteria
##                  1           2           3           4           5
## AIC(n)    10.67982    10.62480    10.59657    10.53707    10.52312
## HQ(n)     10.73795    10.71781    10.72445    10.69984    10.72077
## SC(n)     10.82675    10.85989    10.91982    10.94848    11.02269
## FPE(n) 43469.69436 41143.12751 39998.84605 37689.98480 37170.31442
##                  6           7           8           9          10
## AIC(n)    10.56355    10.55949    10.58130    10.56579    10.59180
## HQ(n)     10.79607    10.82689    10.88358    10.90294    10.96384
## SC(n)     11.15128    11.23538    11.34535    11.41800    11.53217
## FPE(n) 38707.32978 38555.02483 39411.47073 38812.43656 39845.11671
##                 11          12          13          14          15
## AIC(n)    10.60759    10.62356    10.64050    10.65189    10.67600
```

```
## HQ(n)      11.01451    11.06535    11.11717    11.16344    11.22243
## SC(n)      11.63612    11.74025    11.84534    11.94490    12.05717
## FPE(n) 40491.38654 41157.49726 41877.73179 42377.82013 43435.67768
##                 16          17          18          19          20
## AIC(n)     10.67910    10.69173    10.71388    10.70076    10.73001
## HQ(n)      11.26040    11.30791    11.36494    11.38670    11.45083
## SC(n)      12.14843    12.24922    12.35952    12.43456    12.55198
## FPE(n) 43597.75687 44183.06452 45208.18026 44658.27400 46028.95137
```

```
    # AIC(n)  HQ(n)   SC(n) FPE(n)
    # 5      4       1       5
```

The VARSelect function suggest we use either a p=5, using AIC or a p=1 using BIC. After a little additional playing around with parameters, the Model with the lowest ASE value utilized a p=2. For the analysis, we will be using p=2.

We predict the differences using

```
RMortDiffVAR = VAR(cbind(RMS_52,RMS_52_tmp, RMS_52_part),type = "both",p = 2)
preds=predict(RMortDiffVAR,n.ahead=26)
#print(paste0('VAR Difference Prediction:',preds$fcst))
```

We have predicted differences .... calculate actual respiratory mortalities

```
startingPoints = RM[,2][483:508]
RMortForcasts = preds$fcst$RMS_52[,1:3] + startingPoints
```

We will judge the VAR model's fit using ASE, by predicting the last 26 observations.

```
ASE_VAR = mean((RM[,2][483:508] - RMortForcasts[,1])^2)
print(paste0('ASE for Multivariate VAR Model: ',ASE_VAR))
```

```
## [1] "ASE for Multivariate VAR Model: 0.0672277414738173"
```

4a.b) Perform a multivariate analysis using a multivariate MLP model

We will be developing a multivariate neural network model to predict the respiratory mortality rate. For this model, we will be using all (9) explanitory variables in the model. Basically trowing the 'kitchen-sink' at the analysis.

We will also truncate the dataset to only include the first 482 observations. The last 26 will be used for prediction purposes, and model comparision, using ASE values.

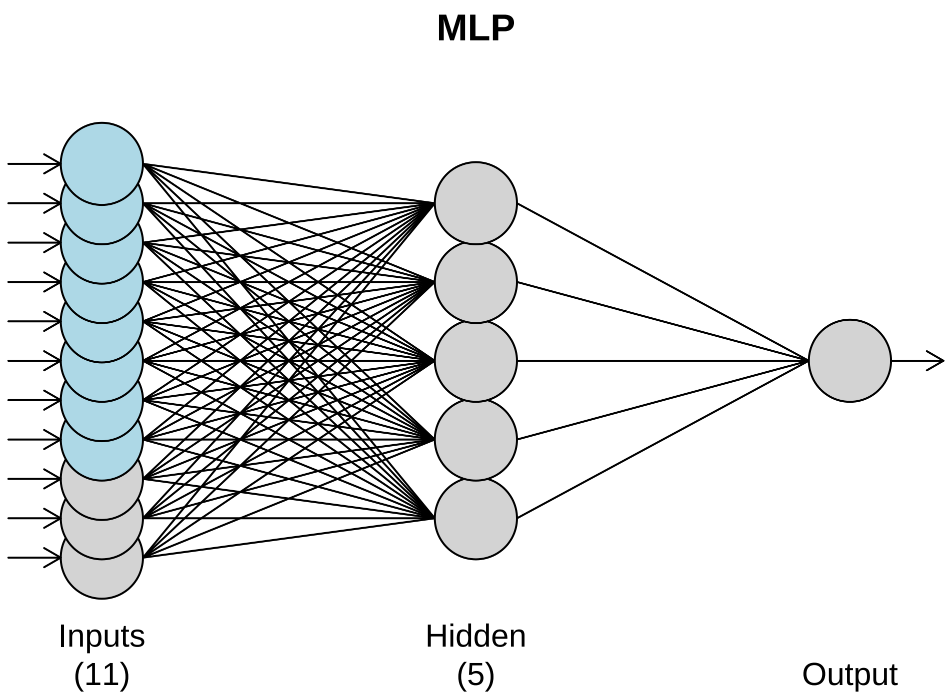
```
set.seed(254)
RM_small = window(RM, start = c(1970, 1),end = c(1979,14))

RM_smallDF = data.frame(week=seq(1,482),RM_small[,c(4:11)])
fit.mlp = mlp(ts(RM_small[,2]),reps = 50,comb = "mean",xreg = RM_smallDF)
fit.mlp
```

```
## MLP fit with 5 hidden nodes and 50 repetitions.
## Univariate lags: (1,2,4)
## 5 regressors included.
## - Regressor 1 lags: (2,3,4)
## - Regressor 2 lags: (2)
## - Regressor 3 lags: (2)
## - Regressor 4 lags: (1,3)
## - Regressor 5 lags: (3)
## Forecast combined using the mean operator.
```

```
## MSE: 1.2984.
```

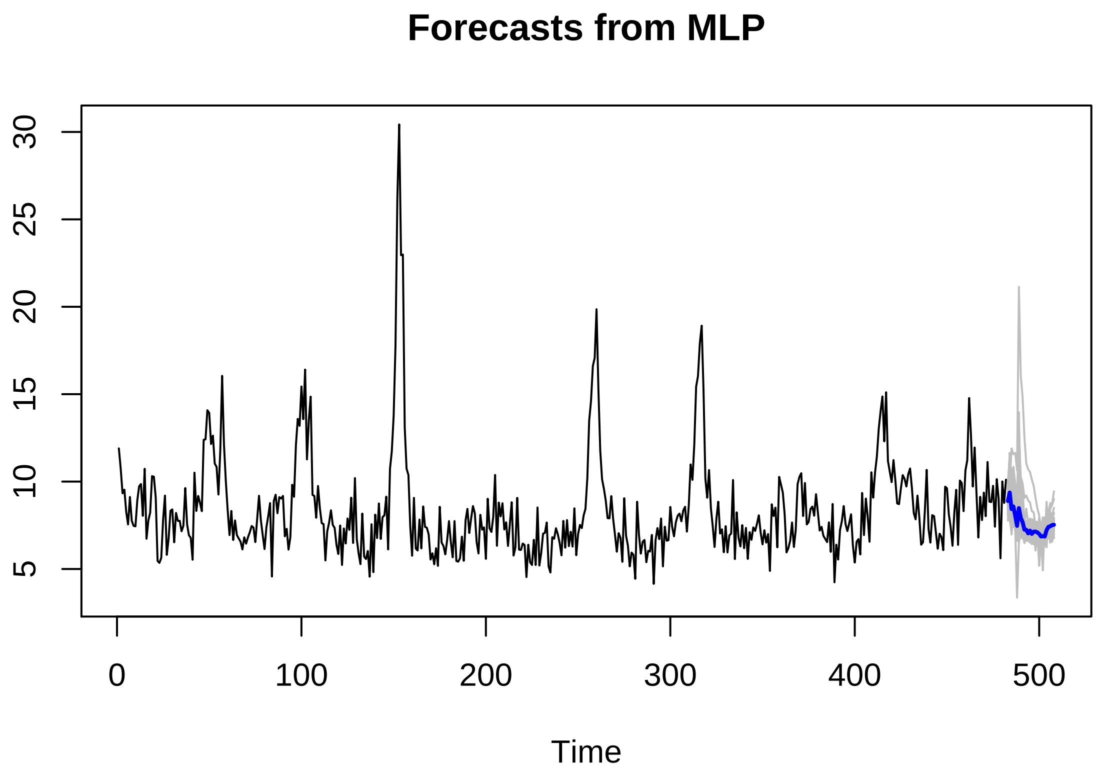
Visualize the Multivariate Neural Network

```r
plot(fit.mlp)
```



Create a forecast for 26 segments ahead, using the Multivariate Neural Network Model and plot the results

```r
RMDF = data.frame(week=seq(1,508),RM[,c(4:11)])
fore.xreg.mlp = forecast(fit.mlp, h = 26, xreg = RMDF)
plot(fore.xreg.mlp)
```



The ASE for the Miltivariate NN Model is:

```
ASE.xreg.mlp = mean((RM[,2][483:508] - fore.xreg.mlp$mean)^2)
print(paste0('ASE for Multivariate NN Model: ',ASE.xreg.mlp))
```

```
## [1] "ASE for Multivariate NN Model: 2.42771896429294"
```

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

We will create an Ensemble method utilizing the mulitvariate VAR and Neural Network Models by taking the average of the two predictions.

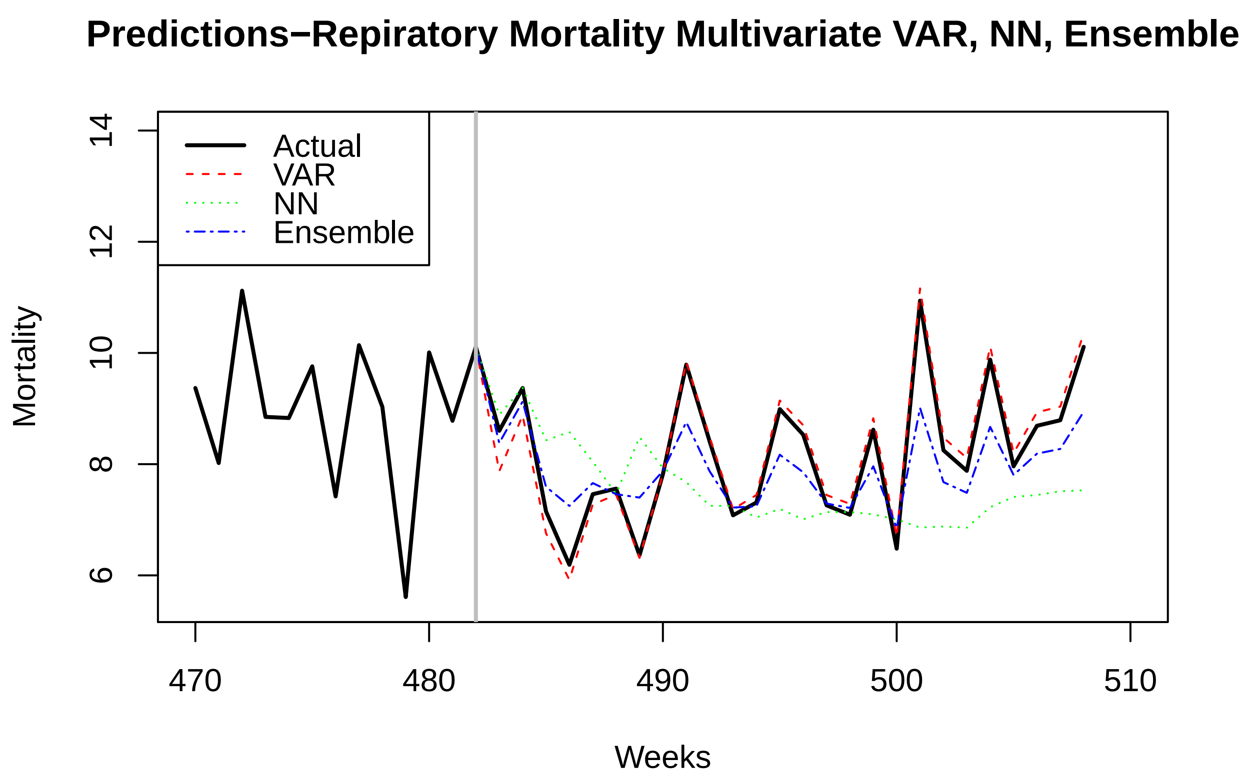
We will judge the model's fit using ASE as well by predicting the observations on the same 26 observations.

```
ensemble_multi = (RMortForcasts[,1] + fore.xreg.mlp$mean)/2

ASE_ens_MV = mean((RM[,2][483:508] - ensemble_multi)^2)
print(paste0('ASE for Multivariate Ensemble Model: ',ASE_ens_MV))
```

```
## [1] "ASE for Multivariate Ensemble Model: 0.508647975973648"
```

```
plot(seq(470,508,1),RM[470:508,2], type = "l",
     main='Predictions-Repiratory Mortality Multivariate VAR, NN, Ensemble'
     ,xlab="Weeks"
     ,ylab="Mortality",
     xlim=c(470, 510),
     ylim=c(5.5,14),lwd=2)
points(seq(482,508,1),c(RM[482,2],RMortForcasts[,1]),type='l',
       col = 'red',lwd=1,lty=2)
```

```
points(seq(482,508,1),c(RM[482,2],fore.xreg.mlp$mean),type='l',
       col = 'green',pch = 17,lwd=1,lty=3)
points(seq(482,508,1),c(RM[482,2],ensemble_multi),type='l',
       col = 'blue',pch = 17,lwd=1,lty=6)
abline(v=482, col="grey",lwd=2)
legend('topleft', legend=c("Actual","VAR", "NN","Ensemble"),
       col=c("black","red", "green","blue"), lty=c(1,2,3,6), lwd=c(2,1,1,1),y.intersp=0.75)
```



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

With the above plot, we are able to clearly see how the 2 Multivariate models (VAR,NN) and the Ensemble methods compare to each other:

The VAR model preformed the best. The predictions tend to match the actual observation values very well. It also had an extremely small ASE of 0.067. This was by far, the best forecasting model we explored.

The Multivariate Neural Network model tends to underestimate the mortality figures across the board, with exception of the initial predictions. The predicted response (variablity of observations) appears to be much more muted than the actual observations

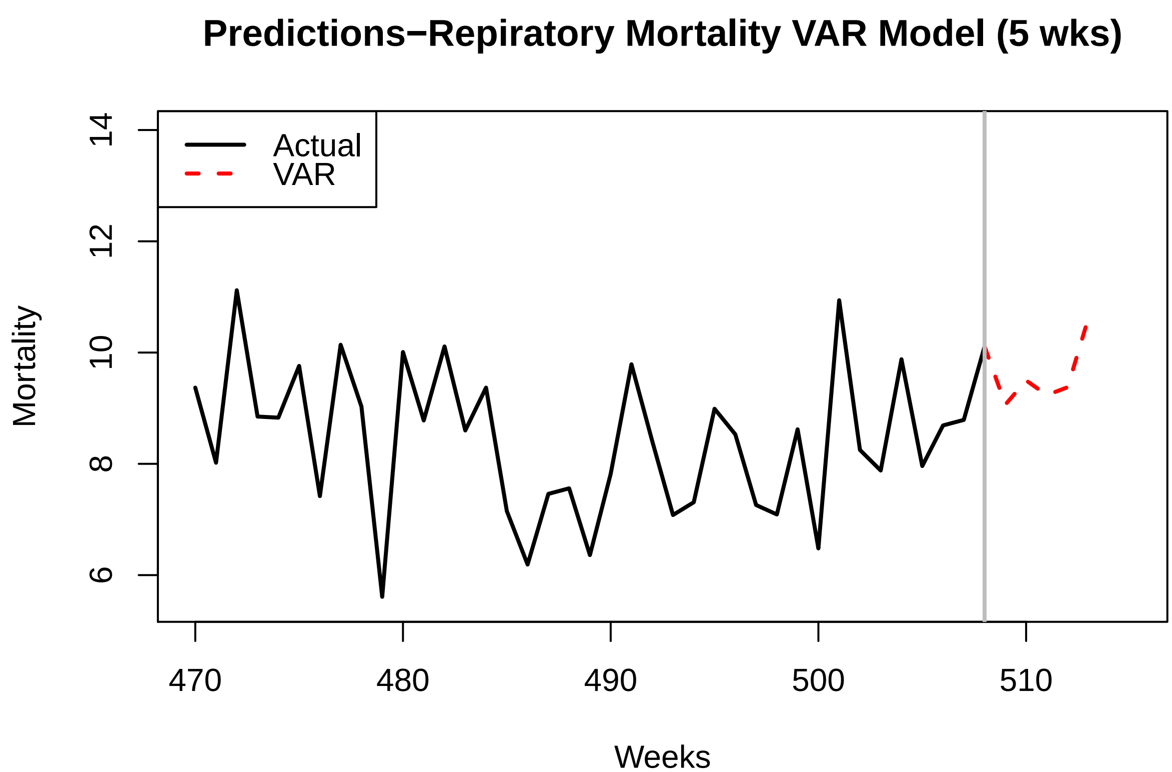
The ensemble model, since it is the average of the 2 models, and given the performance of the other models, it would be expected that it would be better than the NN but not better than the VAR.

```
#RM_52 = artrans.wge(RM[,2], c(rep(0,51),1))
#RM_52_tmp = artrans.wge(RM[,4], c(rep(0,51),1))
#RM_52_part = artrans.wge(RM[,11], c(rep(0,51),1))
```

```r
#RMortDiffVAR_full = VAR(cbind(RM_52,RM_52_tmp, RM_52_part),type = "both",p = 2)
#preds_full=predict(RMortDiffVAR_full,n.ahead=5)

#VAR with p = 2
RMortVAR_full = VAR(cbind(RM[,2], RM[,4], RM[,11]),season = 52, type = "both",p = 2)
preds=predict(RMortVAR_full,n.ahead=5)
print(paste0('VAR Model Predictions:',preds$fcst$RM...2.[,1]))
```

```
## [1] "VAR Model Predictions:9.06431285371794"
## [2] "VAR Model Predictions:9.49942500977947"
## [3] "VAR Model Predictions:9.2351468729523"
## [4] "VAR Model Predictions:9.37759032408864"
## [5] "VAR Model Predictions:10.6236598169289"
```

```r
plot(seq(470,508,1),RM[470:508,2], type = "l",
     main='Predictions-Repiratory Mortality VAR Model (5 wks)'
     ,xlab="Weeks"
     ,ylab="Mortality",
     xlim=c(470, 515),
     ylim=c(5.5,14),lwd=2)
points(seq(508,513,1),c(RM[508,2],preds$fcst$RM...2.[,1]),type='l',
       col = 'red',lwd=2,lty=2)
abline(v=508, col="grey",lwd=2)
legend('topleft', legend=c("Actual","VAR"),
       col=c("black","red"), lty=c(1,2), lwd=c(2,2),y.intersp=0.75)
```



I, Jeffrey Lancon, abided by the SMU Honor Code and did not communicate about the content of this exam with anyone except for Bivin Sadler.