Time Series Covid Project

Reagan Meagher and Megan Riley

#Modeling Covid19 Time Series

Covid19 is a worldwide pandemic that will likely define 2020. In the United States, currently over four million people have been infected and over 150,000 have died as a result of Covid19. As the pandemic continues, limiting infections, serious harm, and death is a primary concern for all involved.

As this is a novel illness, we know relatively little, but understanding how Covid19 is spreading and judging the severity of an outbreak can be approximated with the data we have available. In this report we aim to build effective time series models to forecast future Covid19 cases using the techniques we have learned from this Time Series course. ##*Goal One: Data Collection*

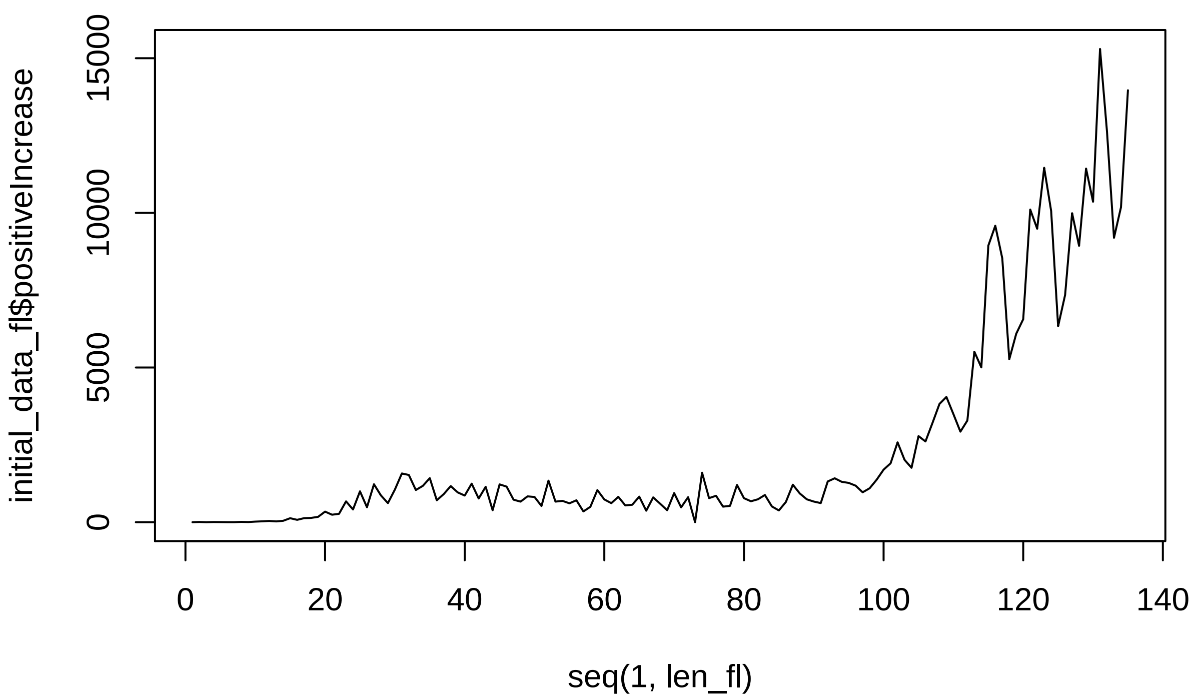
The data source we are using is sourced from The Covid Tracking Project.

```
initial_data_fl <- read.csv(file="https://raw.githubusercontent.com/megnn/TimeSeries_Covid/master/covid
initial_data_us <- read.csv(file="https://raw.githubusercontent.com/megnn/TimeSeries_Covid/master/covid

initial_data_fl = initial_data_fl[order(nrow(initial_data_fl):1),]
initial_data_us = initial_data_us[order(nrow(initial_data_us):1),]

len_fl = dim(initial_data_fl)[1]
len_us = dim(initial_data_us)[1]
```

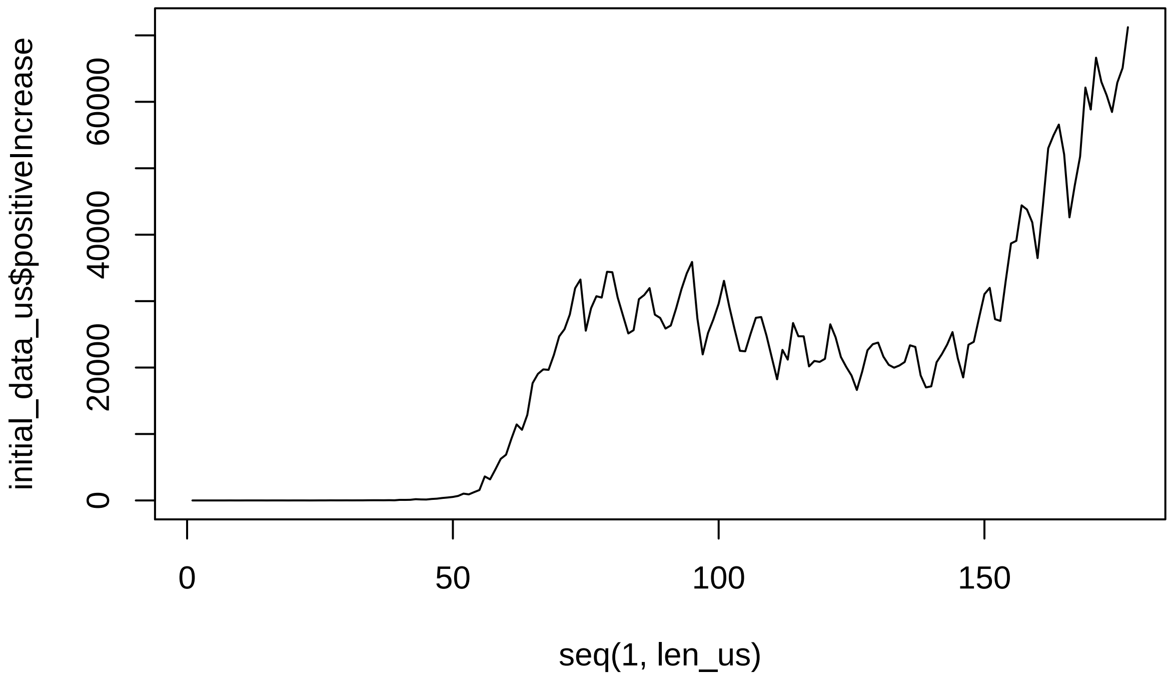
Insert exploration of cases, deaths, hospitalizations etc.

Plotting the daily positive cases in Florida and US.

```
plot(x = seq(1,len_fl), y = initial_data_fl$positiveIncrease, type = "l")
```




```
plot(x = seq(1,len_us), y = initial_data_us$positiveIncrease, type = "l")
```

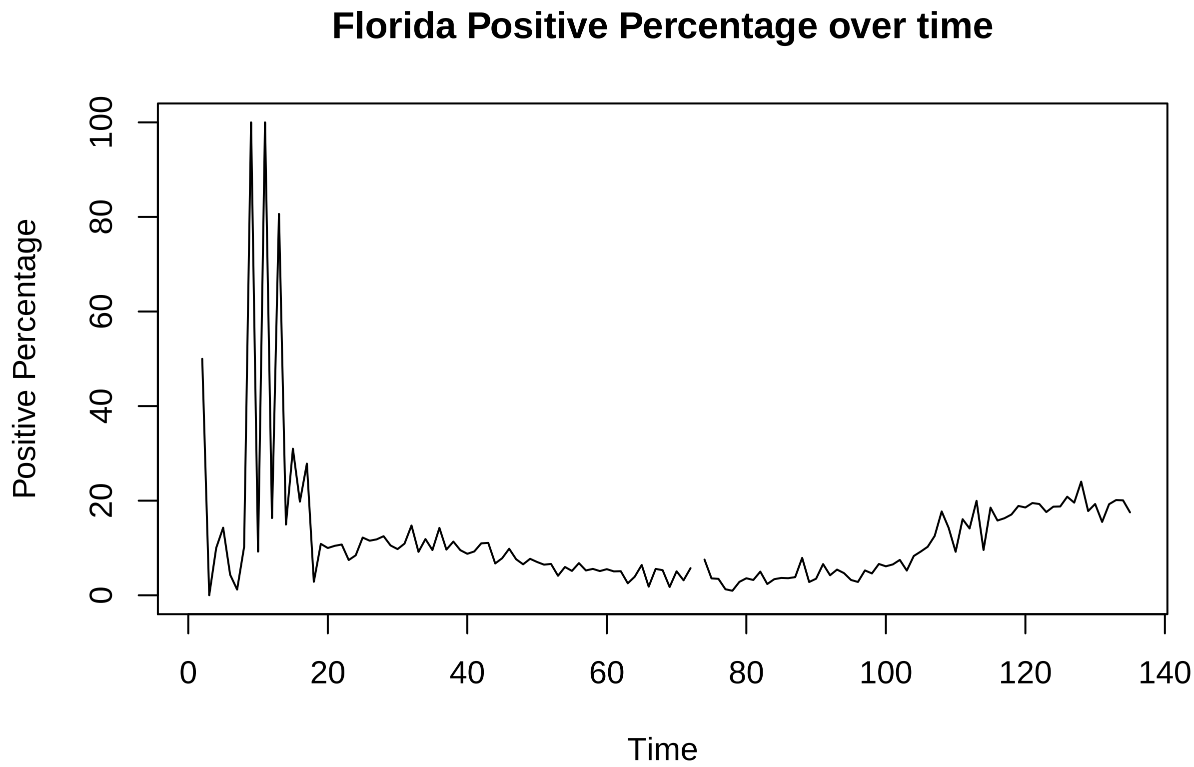


##Positive Percentage

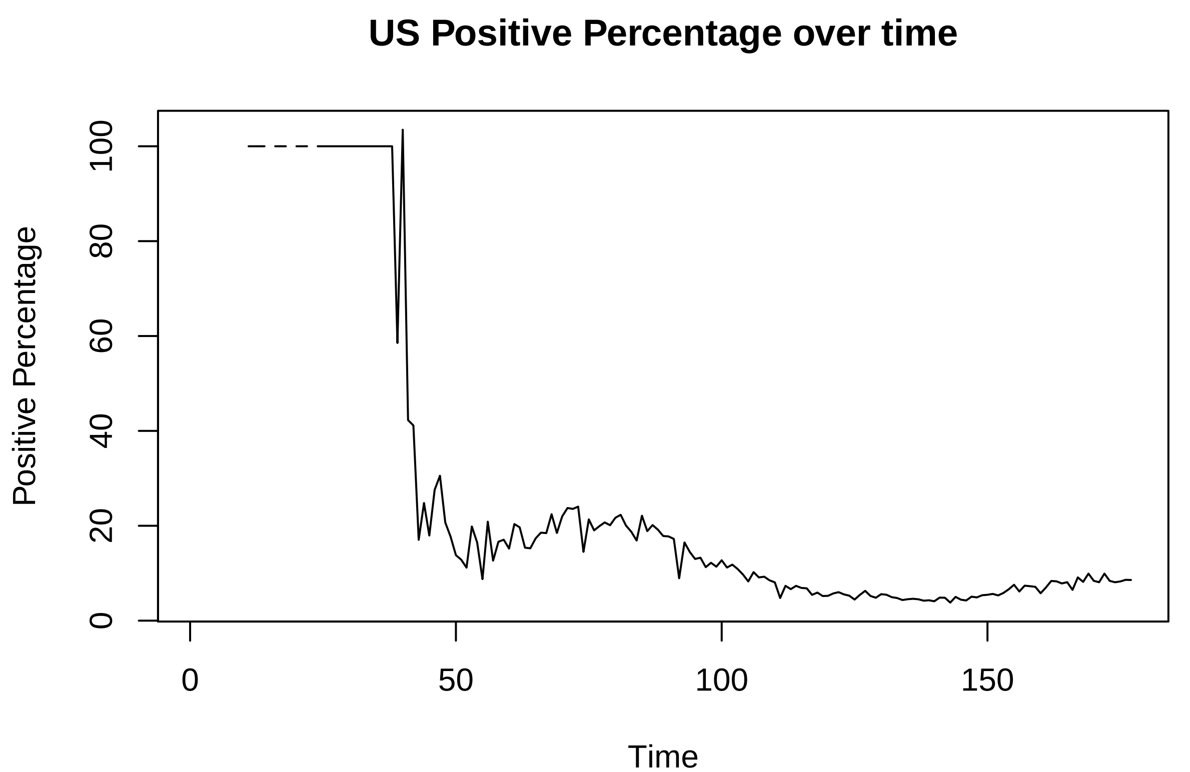
Positive Percentage is a statistic that calculates daily positive tests as a percentage of daily overall tests returned. We calculated this column and added it to our data below followed by some visual exploration of the statistic itself.

Overall we see a clear instance of high and often 100% positive test rates early on in the first days and weeks of the pandemic spread. We understand this as a result of the fact that Covid19 spread fast and we had more community spread than anticipated early on without the testing available. It is abundantly clear that when we have extremely high percent positive rates near 100% we can expect true positive case numbers at the time to be under represented. But without better epidemlogical understanding we can't make judgement calls on true case numbers when percent positives rise from 5% to 10% as we see begin to happen somewhat in recent days in Florida.

```
for (i in 1:nrow(initial_data_fl)) {
  n <- round((initial_data_fl$positiveIncrease / initial_data_fl$totalTestResultsIncrease) * 100, digit
  initial_data_fl$positive_percentage <- n
}

for (i in 1:nrow(initial_data_us)) {
  n <- round((initial_data_us$positiveIncrease / initial_data_us$totalTestResultsIncrease) * 100, digit
  initial_data_us$positive_percentage <- n
}


#Percent Positive Exploration
plot(x = seq(1:len_fl), y = initial_data_fl$positive_percentage, type = "l", main = "Florida Positive Pe
```




```
plot(x = seq(1:len_us), y = initial_data_us$positive_percentage, type = "l", main = "US Positive Percen
```

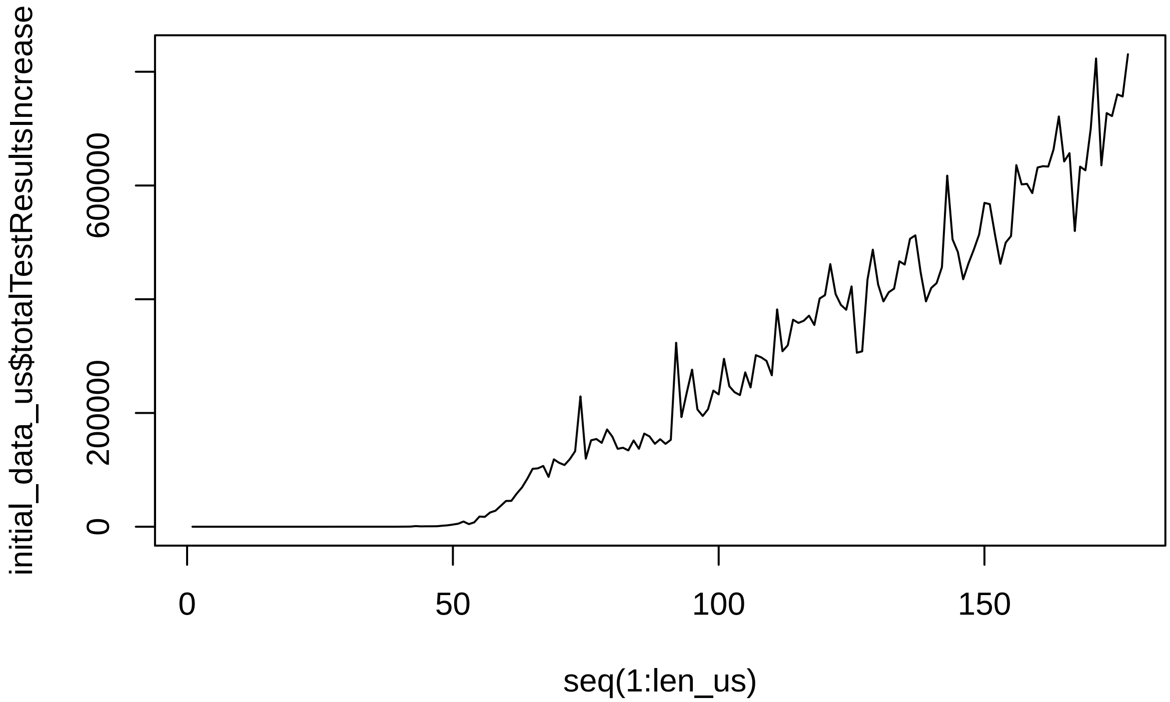


Positive Percentage as a metric is a measure of two main things, how many tests are we administering and how many positives are we receiving. If tests are skyrocketing while positive cases are increasing, we would see a stable or even diminishing line which could indicate not a pandemic under control but simply better testing resources but could be interpreted as a pandemic managed.

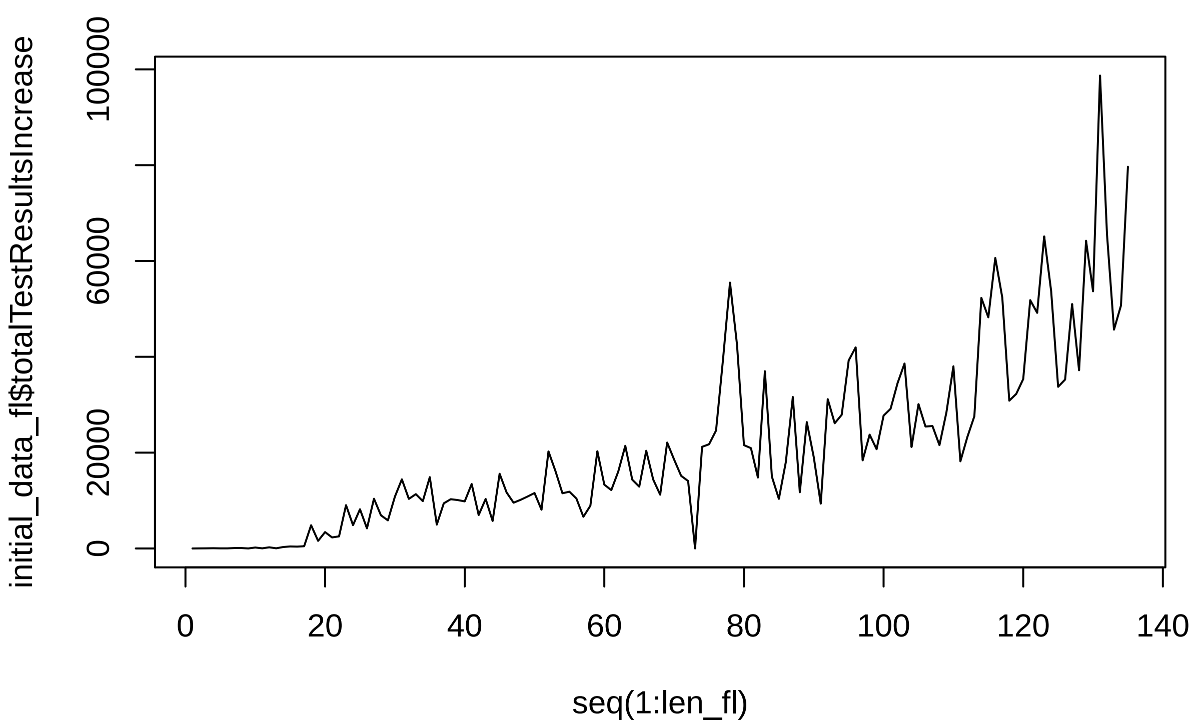
Keeping tests increasing to continue to keep percent positives level is a good indication we have leveled up our resources to continue to diagnose the pandemic at the same level, but if we need to scale up our testing to keep the same positive percentage, there is more covid spread.

However, an increasing positive percentage is a good indicator that our testing resources may not be up to actually up to tracking the current stage of the pandemic.

```
plot(x = seq(1:len_us), y = initial_data_us$totalTestResultsIncrease, type = "l")
```




```
plot(x = seq(1:len_fl), y = initial_data_fl$totalTestResultsIncrease, type = "l")
```



####Data Preperation

In order to model new case numbers by day we set up dataframes with only our date and positive increase amount per day.

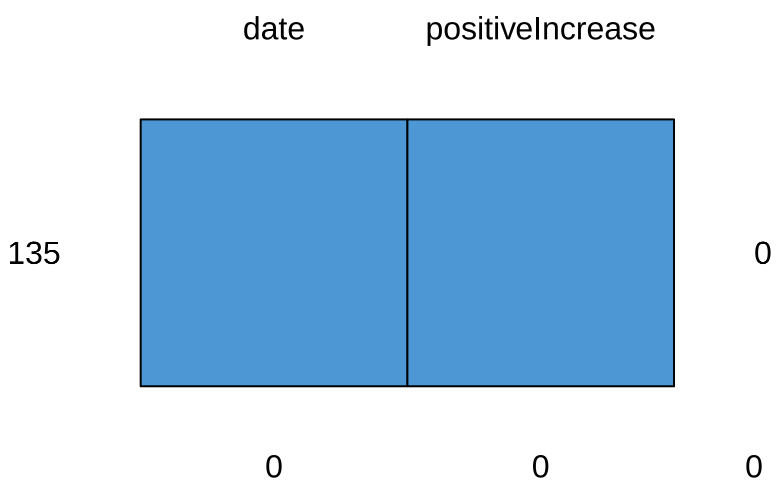
```
newcases_fl <- dplyr::select(initial_data_fl, c("date", "positiveIncrease"))
newcases_us <- dplyr::select(initial_data_us, c("date", "positiveIncrease"))
```

####Checking for NAs

We can see with the missing value analysis below that we have no NAs present in our new case data.

```
#Checking for NAs
md.pattern(newcases_fl)
```

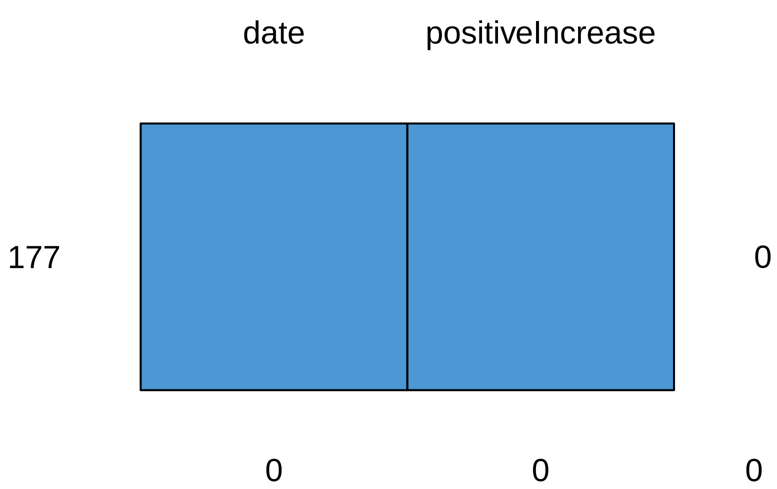
```
##  /\     /\
## {  `---'  }
## {  O   O  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
```




```
##     date positiveIncrease
## 135    1                1 0
##        0                0 0
```

```
# No NAs present


#Checking for NAs
md.pattern(newcases_us)
```

```
##  /\     /\
## {  `---'  }
## {  O   O  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
```

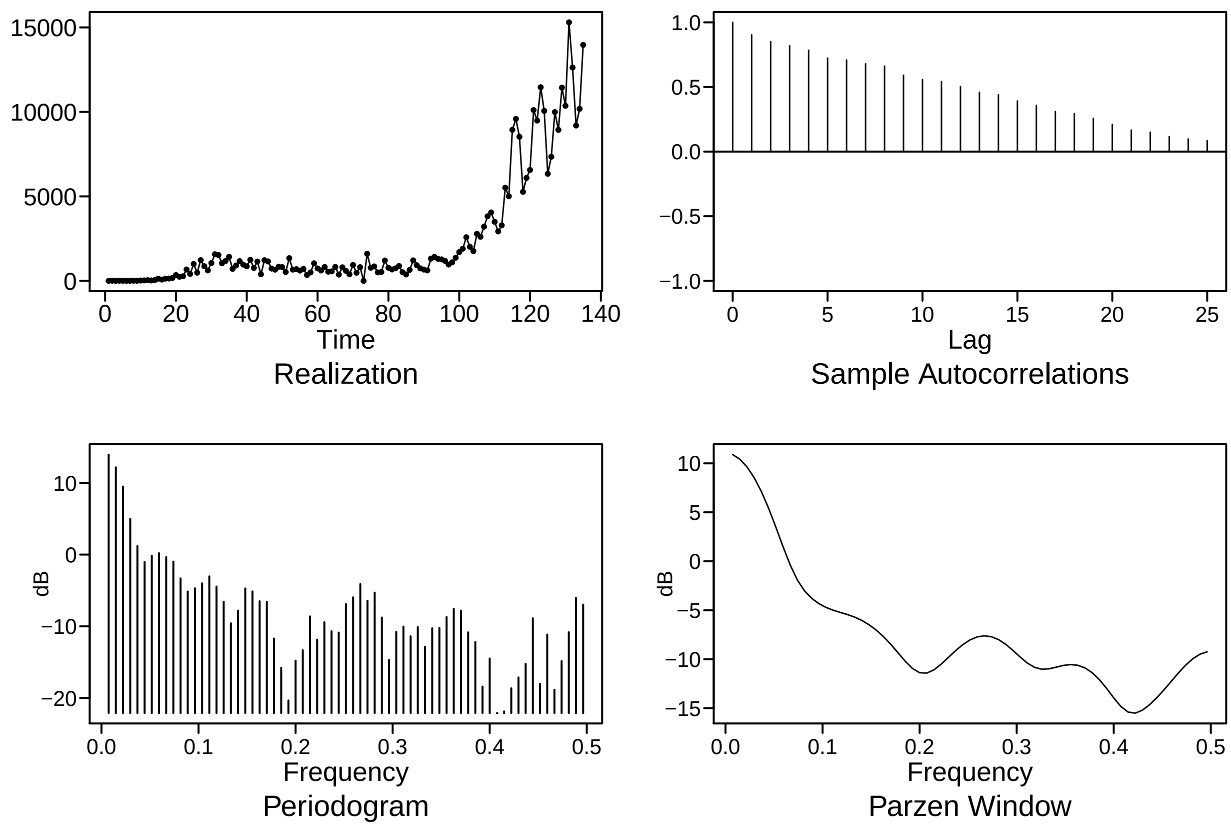



```
##      date positiveIncrease
## 177     1                1 0
##         0                0 0
```

```
#No NAs present
```

###**Florida Daily Cases:**

```
plotts.sample.wge(newcases_fl$positiveIncrease)
```

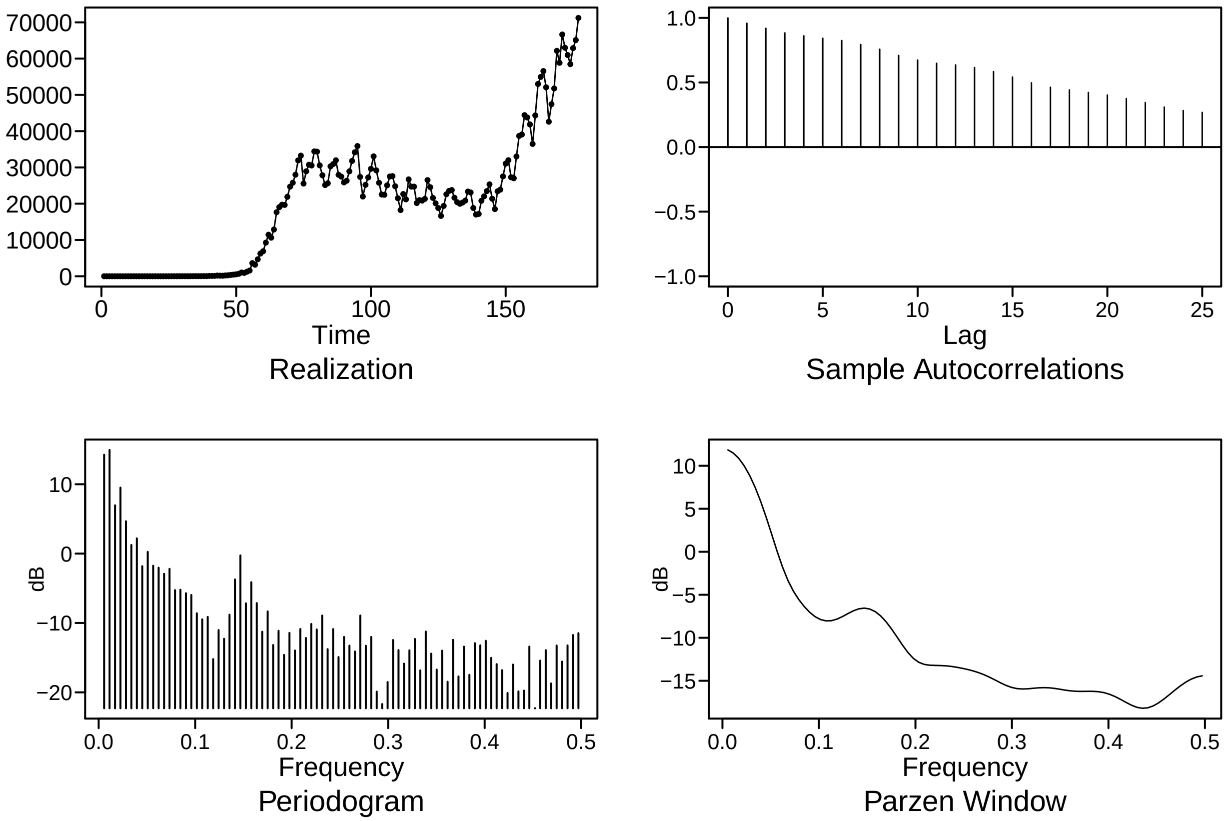



```
## $autplt
##  [1] 1.00000000 0.90351878 0.85112908 0.81870330 0.78426726 0.72368910
##  [7] 0.70892248 0.68040351 0.66167307 0.59052464 0.55736019 0.54019962
## [13] 0.50346164 0.45903473 0.44018864 0.39175866 0.35750776 0.31032227
```

```
## [19] 0.29436806 0.25707296 0.21075421 0.16689463 0.15038715 0.11609674
## [25] 0.09827282 0.08510643
##
## $freq
##  [1] 0.007407407 0.014814815 0.022222222 0.029629630 0.037037037
##  [6] 0.044444444 0.051851852 0.059259259 0.066666667 0.074074074
## [11] 0.081481481 0.088888889 0.096296296 0.103703704 0.111111111
## [16] 0.118518519 0.125925926 0.133333333 0.140740741 0.148148148
## [21] 0.155555556 0.162962963 0.170370370 0.177777778 0.185185185
## [26] 0.192592593 0.200000000 0.207407407 0.214814815 0.222222222
## [31] 0.229629630 0.237037037 0.244444444 0.251851852 0.259259259
## [36] 0.266666667 0.274074074 0.281481481 0.288888889 0.296296296
## [41] 0.303703704 0.311111111 0.318518519 0.325925926 0.333333333
## [46] 0.340740741 0.348148148 0.355555556 0.362962963 0.370370370
## [51] 0.377777778 0.385185185 0.392592593 0.400000000 0.407407407
## [56] 0.414814815 0.422222222 0.429629630 0.437037037 0.444444444
## [61] 0.451851852 0.459259259 0.466666667 0.474074074 0.481481481
## [66] 0.488888889 0.496296296
##
## $db
##  [1]  13.9443470  12.1738932   9.4960087   5.0020029   1.2010108
##  [6]  -1.0130101  -0.1506086   0.2039391  -0.3463438  -0.9726125
## [11]  -3.3208756  -5.1446096  -4.6985377  -4.0076175  -3.0331597
## [16]  -4.4568722  -6.5691185  -9.5735201  -7.8085456  -4.7203532
## [21]  -5.1362477  -6.5033145  -6.5841302 -11.7008911 -15.7768594
## [26] -20.3454088 -14.7914168 -13.3311995  -8.6134078 -11.8506844
## [31]  -9.4294479 -10.6796170 -10.8717772  -6.8765678  -5.9591468
## [36]  -4.1063148  -6.4365181  -5.2994271  -8.7718362 -14.6806581
## [41] -10.7718204 -10.0417172 -11.3897236 -10.1069047 -12.8544662
## [46] -10.2709895 -10.2140102  -8.6959976  -7.5565160  -7.8110494
## [51] -10.8377249 -12.1847651 -18.4078010 -14.4929054 -22.1053741
## [56] -21.8671829 -18.6457477 -17.1312143 -15.2219426  -8.8827236
## [61] -18.0198672 -11.1478112 -18.8429668 -14.8396876 -10.8187051
## [66]  -6.0417196  -6.9676002
##
## $dbz
##  [1]  10.8952916  10.4208269   9.6290499   8.5210982   7.1053841
##  [6]   5.4084900   3.4951614   1.4961088  -0.3815372  -1.9206026
## [11]  -3.0320075  -3.7937247  -4.3311799  -4.7225622  -5.0088115
## [16]  -5.2333565  -5.4517814  -5.7160206  -6.0608486  -6.5041876
## [21]  -7.0550944  -7.7189474  -8.4927699  -9.3490346 -10.2124871
## [26] -10.9466178 -11.3848655 -11.4222535 -11.0879684 -10.5084533
## [31]  -9.8212142  -9.1329494  -8.5201783  -8.0386115  -7.7282189
## [36]  -7.6148214  -7.7102817  -8.0113599  -8.4961143  -9.1170132
## [41]  -9.7927830 -10.4087161 -10.8436066 -11.0271353 -10.9840189
## [46] -10.8151075 -10.6393942 -10.5538767 -10.6249324 -10.8939499
## [51] -11.3823560 -12.0894051 -12.9791522 -13.9554605 -14.8397014
## [56] -15.4053463 -15.5118058 -15.2066656 -14.6399148 -13.9276436
## [61] -13.1243083 -12.2660250 -11.4056316 -10.6130044  -9.9570803
## [66]  -9.4915816  -9.2507536
```

###**US Daily Cases:**

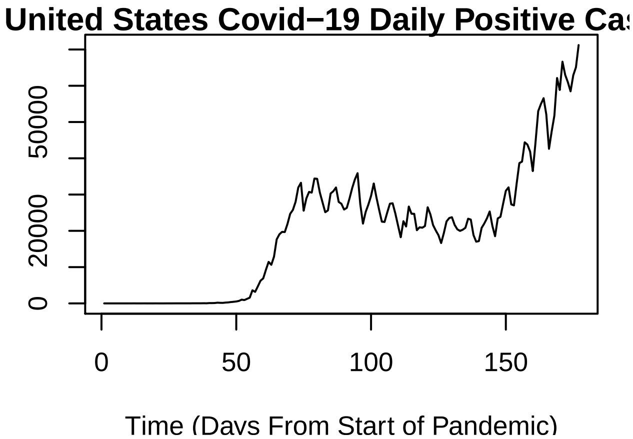
```r
plotts.sample.wge(newcases_us$positiveIncrease)
```




```
## $autplt
##  [1] 1.0000000 0.9588275 0.9195777 0.8839579 0.8613438 0.8422725 0.8250617
##  [8] 0.7941208 0.7575713 0.7092840 0.6736316 0.6484889 0.6360652 0.6158459
## [15] 0.5843954 0.5425539 0.4973999 0.4629802 0.4433251 0.4227875 0.4027823
## [22] 0.3756549 0.3451029 0.3099138 0.2830722 0.2684966
##
## $freq
##  [1] 0.005649718 0.011299435 0.016949153 0.022598870 0.028248588
##  [6] 0.033898305 0.039548023 0.045197740 0.050847458 0.056497175
## [11] 0.062146893 0.067796610 0.073446328 0.079096045 0.084745763
## [16] 0.090395480 0.096045198 0.101694915 0.107344633 0.112994350
## [21] 0.118644068 0.124293785 0.129943503 0.135593220 0.141242938
## [26] 0.146892655 0.152542373 0.158192090 0.163841808 0.169491525
## [31] 0.175141243 0.180790960 0.186440678 0.192090395 0.197740113
## [36] 0.203389831 0.209039548 0.214689266 0.220338983 0.225988701
## [41] 0.231638418 0.237288136 0.242937853 0.248587571 0.254237288
## [46] 0.259887006 0.265536723 0.271186441 0.276836158 0.282485876
## [51] 0.288135593 0.293785311 0.299435028 0.305084746 0.310734463
## [56] 0.316384181 0.322033898 0.327683616 0.333333333 0.338983051
## [61] 0.344632768 0.350282486 0.355932203 0.361581921 0.367231638
## [66] 0.372881356 0.378531073 0.384180791 0.389830508 0.395480226
## [71] 0.401129944 0.406779661 0.412429379 0.418079096 0.423728814
## [76] 0.429378531 0.435028249 0.440677966 0.446327684 0.451977401
## [81] 0.457627119 0.463276836 0.468926554 0.474576271 0.480225989
## [86] 0.485875706 0.491525424 0.497175141
```

```
##
## $db
##  [1]  14.2491777  14.9471733   6.9591395   9.5150572   4.6664320
##  [6]   1.2452886   2.2001263  -1.8153526   0.2487892  -1.7451730
## [11]  -2.0281764  -2.9144478  -2.1941621  -5.2629156  -5.2055932
## [16]  -5.7078490  -5.9605659  -8.5976595  -9.4776476  -9.1217112
## [21] -15.2098111 -11.0130098 -12.2778403  -8.7998267  -3.7259575
## [26]  -0.2641817  -7.1652992  -4.1204548  -7.1212961 -11.2647439
## [31]  -8.3267561 -13.1787605 -11.1231072 -14.6051895 -11.4235887
## [36] -13.9657569 -10.8725668 -12.1561887 -10.1420832 -10.9336635
## [41]  -8.9248527 -13.7687929 -10.8808798 -14.9080849 -12.0130570
## [46] -13.2631016 -14.1075759  -8.9262800 -13.2798822 -12.0143070
## [51] -19.8939725 -21.6888817 -18.5067252 -12.4549799 -13.9023779
## [56] -15.8533153 -13.9405227 -12.2972153 -16.8346161 -11.2349117
## [61] -14.4207233 -16.7182236 -13.9828581 -18.4659902 -12.4266337
## [66] -17.7015821 -13.4087263 -17.4766913 -12.9127916 -13.2269329
## [71] -12.5587063 -15.0282294 -15.9243220 -16.7978132 -20.0879876
## [76] -15.9938423 -19.8581093 -19.7477287 -13.4007437 -22.2946687
## [81] -15.4248492 -13.9078952 -18.7173113 -13.2457797 -15.5641975
## [86] -13.2318891 -11.7321059 -11.4612747
##
## $dbz
##  [1]  11.8458790  11.4754641  10.8560827   9.9854360   8.8621551
##  [6]   7.4886835   5.8768395   4.0578211   2.0976111   0.1122841
## [11]  -1.7390488  -3.3133638  -4.5663762  -5.5582448  -6.3701775
## [16]  -7.0372735  -7.5489517  -7.8837835  -8.0344230  -8.0096686
## [21]  -7.8311721  -7.5384172  -7.1927256  -6.8685401  -6.6368757
## [26]  -6.5528464  -6.6515867  -6.9494978  -7.4465624  -8.1271327
## [31]  -8.9581871  -9.8854455 -10.8302689 -11.6947752 -12.3846543
## [36] -12.8467322 -13.0936848 -13.1902507 -13.2146456 -13.2278985
## [41] -13.2637087 -13.3324027 -13.4305242 -13.5511321 -13.6916541
## [46] -13.8569119 -14.0567183 -14.2998416 -14.5873019 -14.9074910
## [51] -15.2348171 -15.5333097 -15.7659711 -15.9078358 -15.9565105
## [56] -15.9334175 -15.8749606 -15.8195333 -15.7968128 -15.8217940
## [61] -15.8930162 -15.9942147 -16.0995791 -16.1827504 -16.2275006
## [66] -16.2352671 -16.2251647 -16.2268128 -16.2706537 -16.3802168
## [71] -16.5676425 -16.8313863 -17.1546884 -17.5045102 -17.8324754
## [76] -18.0809193 -18.1961576 -18.1453945 -17.9272792 -17.5687318
## [81] -17.1120696 -16.6026318 -16.0823776 -15.5882458 -15.1521143
## [86] -14.8005859 -14.5543805 -14.4276807
```

```r
plot(x = seq(1,len_us), y = newcases_us$positiveIncrease, type = "l", main = 'United States Covid-19 Da
```

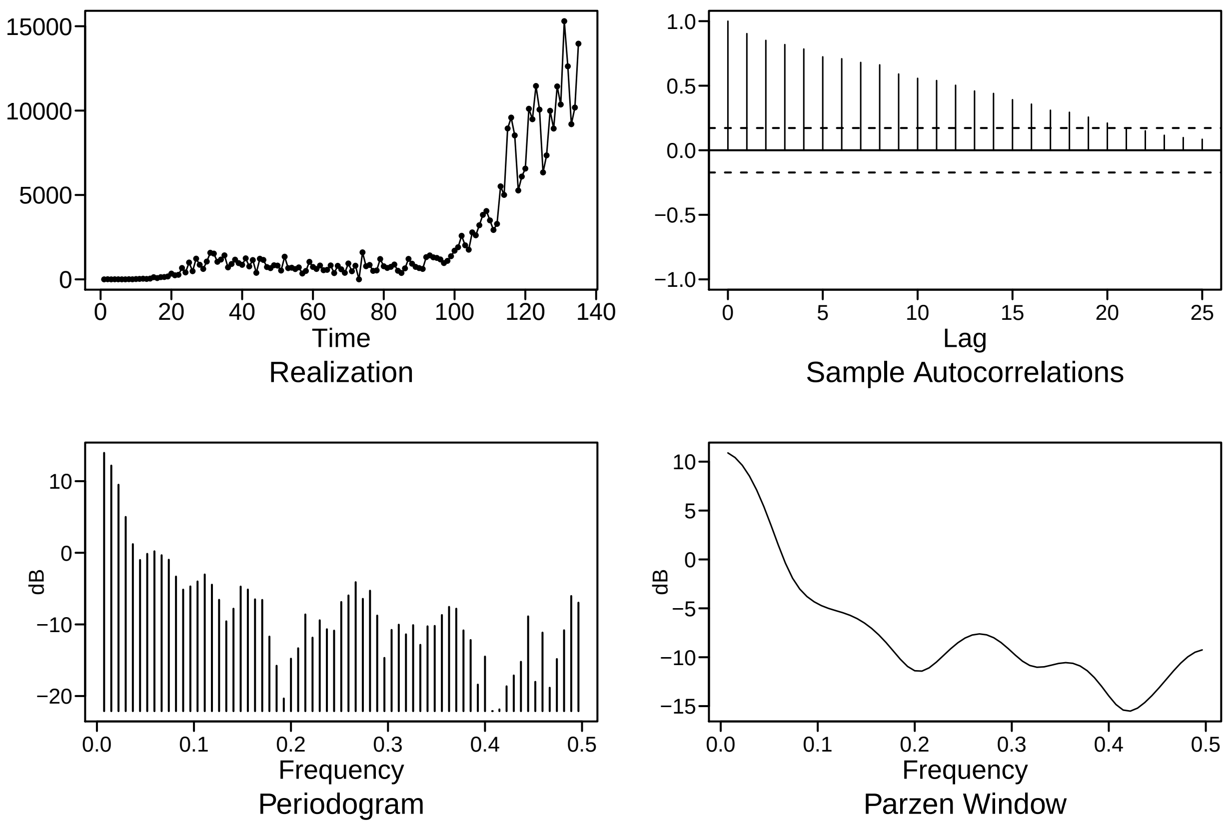


##*Goal Two: Univariate Analysis*

###Model Building for Cases in Florida

Stationarity vs non-stationarity - concerns about the data

```
plotts.sample.wge(newcases_fl$positiveIncrease, arlimits = TRUE)
```




```
## $autplt
##  [1] 1.00000000 0.90351878 0.85112908 0.81870330 0.78426726 0.72368910
##  [7] 0.70892248 0.68040351 0.66167307 0.59052464 0.55736019 0.54019962
## [13] 0.50346164 0.45903473 0.44018864 0.39175866 0.35750776 0.31032227
## [19] 0.29436806 0.25707296 0.21075421 0.16689463 0.15038715 0.11609674
## [25] 0.09827282 0.08510643
##
```

```
## $freq
##  [1] 0.007407407 0.014814815 0.022222222 0.029629630 0.037037037
##  [6] 0.044444444 0.051851852 0.059259259 0.066666667 0.074074074
## [11] 0.081481481 0.088888889 0.096296296 0.103703704 0.111111111
## [16] 0.118518519 0.125925926 0.133333333 0.140740741 0.148148148
## [21] 0.155555556 0.162962963 0.170370370 0.177777778 0.185185185
## [26] 0.192592593 0.200000000 0.207407407 0.214814815 0.222222222
## [31] 0.229629630 0.237037037 0.244444444 0.251851852 0.259259259
## [36] 0.266666667 0.274074074 0.281481481 0.288888889 0.296296296
## [41] 0.303703704 0.311111111 0.318518519 0.325925926 0.333333333
## [46] 0.340740741 0.348148148 0.355555556 0.362962963 0.370370370
## [51] 0.377777778 0.385185185 0.392592593 0.400000000 0.407407407
## [56] 0.414814815 0.422222222 0.429629630 0.437037037 0.444444444
## [61] 0.451851852 0.459259259 0.466666667 0.474074074 0.481481481
## [66] 0.488888889 0.496296296
##
## $db
##  [1]  13.9443470  12.1738932   9.4960087   5.0020029   1.2010108
##  [6]  -1.0130101  -0.1506086   0.2039391  -0.3463438  -0.9726125
## [11]  -3.3208756  -5.1446096  -4.6985377  -4.0076175  -3.0331597
## [16]  -4.4568722  -6.5691185  -9.5735201  -7.8085456  -4.7203532
## [21]  -5.1362477  -6.5033145  -6.5841302 -11.7008911 -15.7768594
## [26] -20.3454088 -14.7914168 -13.3311995  -8.6134078 -11.8506844
## [31]  -9.4294479 -10.6796170 -10.8717772  -6.8765678  -5.9591468
## [36]  -4.1063148  -6.4365181  -5.2994271  -8.7718362 -14.6806581
## [41] -10.7718204 -10.0417172 -11.3897236 -10.1069047 -12.8544662
## [46] -10.2709895 -10.2140102  -8.6959976  -7.5565160  -7.8110494
## [51] -10.8377249 -12.1847651 -18.4078010 -14.4929054 -22.1053741
## [56] -21.8671829 -18.6457477 -17.1312143 -15.2219426  -8.8827236
## [61] -18.0198672 -11.1478112 -18.8429668 -14.8396876 -10.8187051
## [66]  -6.0417196  -6.9676002
##
## $dbz
##  [1]  10.8952916  10.4208269   9.6290499   8.5210982   7.1053841
##  [6]   5.4084900   3.4951614   1.4961088  -0.3815372  -1.9206026
## [11]  -3.0320075  -3.7937247  -4.3311799  -4.7225622  -5.0088115
## [16]  -5.2333565  -5.4517814  -5.7160206  -6.0608486  -6.5041876
## [21]  -7.0550944  -7.7189474  -8.4927699  -9.3490346 -10.2124871
## [26] -10.9466178 -11.3848655 -11.4222535 -11.0879684 -10.5084533
## [31]  -9.8212142  -9.1329494  -8.5201783  -8.0386115  -7.7282189
## [36]  -7.6148214  -7.7102817  -8.0113599  -8.4961143  -9.1170132
## [41]  -9.7927830 -10.4087161 -10.8436066 -11.0271353 -10.9840189
## [46] -10.8151075 -10.6393942 -10.5538767 -10.6249324 -10.8939499
## [51] -11.3823560 -12.0894051 -12.9791522 -13.9554605 -14.8397014
## [56] -15.4053463 -15.5118058 -15.2066656 -14.6399148 -13.9276436
## [61] -13.1243083 -12.2660250 -11.4056316 -10.6130044  -9.9570803
## [66]  -9.4915816  -9.2507536
```

Model IDing of stationary models

Model Building

####Florida Cases - ARMA(2,1) Model

To model Florida cases we start with a base model. We can see that the most favored model by BIC is an AR(1). So we begin by building that model. Our AR(1) has a phi of .975, quite close to the unit circle, which we expect to model strongly wandering behavior.

This model has an AIC estimate of 14.01.

In order to estimate an average ASE, we are running this model over segments of our data with 54 iterations. In each case training with at least seventy data points and predicting on twelve. Overall this produces an average ASE of 6,353,070.

Below our ASE estimates we forecast short term and long term forecasts and both follow AR(1) behavior of data dampening towards our mean.

```r
aic5.wge(newcases_fl$positiveIncrease)
```

```
## ---------WORKING... PLEASE WAIT...
##
##
## Error in aic calculation at 1 1
## Error in aic calculation at 1 2
## Error in aic calculation at 2 0
## Error in aic calculation at 2 2
## Error in aic calculation at 3 0
## Error in aic calculation at 3 1
## Error in aic calculation at 3 2
## Error in aic calculation at 4 0
## Error in aic calculation at 4 1
## Error in aic calculation at 4 2
## Error in aic calculation at 5 0
## Error in aic calculation at 5 1
## Error in aic calculation at 5 2
## Five Smallest Values of  aic
```

```
##      p    q        aic
## 8    2    1   13.97724
## 4    1    0   14.01245
## 3    0    2   14.80561
## 2    0    1   15.43958
## 1    0    0   16.27428
```

```r
aic5.wge(newcases_fl$positiveIncrease, type = 'bic')
```

```
## ---------WORKING... PLEASE WAIT...
##
##
## Error in aic calculation at 1 1
## Error in aic calculation at 1 2
## Error in aic calculation at 2 0
## Error in aic calculation at 2 2
## Error in aic calculation at 3 0
## Error in aic calculation at 3 1
## Error in aic calculation at 3 2
## Error in aic calculation at 4 0
## Error in aic calculation at 4 1
## Error in aic calculation at 4 2
## Error in aic calculation at 5 0
## Error in aic calculation at 5 1
## Error in aic calculation at 5 2
## Five Smallest Values of  bic
```

```
##      p    q        bic
## 4    1    0   14.05549
## 8    2    1   14.06332
## 3    0    2   14.87017
## 2    0    1   15.48262
## 1    0    0   16.29580
```

```r
fl_arma_21 = est.arma.wge(newcases_fl$positiveIncrease, p = 2, q = 1)
```

```
##
## Coefficients of Original polynomial:
## -0.0070 0.9634
##
## Factor                 Roots                Abs Recip    System Freq
## 1+0.9850B             -1.0152               0.9850       0.5000
## 1-0.9780B              1.0225               0.9780       0.0000
##
##
```

```r
fl_arma_21
```

```
## $phi
## [1] -0.006958978  0.963353207
##
## $theta
## [1] -0.9221269
##
## $res
##   [1]   -96.980030   -31.446941   -72.949698   -36.471073   -65.330728
##   [6]   -45.549146   -61.877488   -36.893978   -64.876273   -32.814665
##  [11]   -48.465346   -35.457746   -70.039076   -29.808364    29.719937
##  [16]   -95.862182    -6.379709   -30.423757   -26.217494   131.433577
##  [21]  -145.518665   -24.636130   363.431958  -282.548857   511.065905
##  [26]  -478.293879   607.985877  -254.397575  -426.424493   517.214198
##  [31]   409.019162    41.458684  -606.888552   167.792148   172.553403
##  [36]  -673.223311    61.876216   332.217433  -312.910066   -69.985375
##  [41]   286.821460  -421.258504   235.449816  -663.078588   632.056247
##  [46]   101.816955  -638.099434    47.615189    -6.640751    84.271350
##  [51]  -455.440684   879.447680  -742.453664   -16.552237  -112.568365
##  [56]    50.300458  -384.098389    69.551687   541.043082  -337.470012
##  [61]  -170.652071   170.581555  -304.057150   -43.831702   246.202377
##  [66]  -494.646068   362.982017  -193.534920  -306.001397   550.666659
##  [71]  -496.086986   260.276677  -797.827951  1456.312778  -656.760412
##  [76]  -179.301368  -177.240334  -230.768626   834.865558  -595.157039
##  [81]   -31.663234   -75.656780   200.691549  -484.823999  -119.172558
##  [86]   171.186415   591.567341  -339.205324  -211.339228  -128.000624
##  [91]   -74.240039   645.199359   136.823849  -182.026824   -22.061377
##  [96]  -149.991125  -212.932603    60.319997   290.408665   281.915507
## [101]   231.100730   643.361538  -493.594169  -361.225326  1084.212942
## [106]  -165.986553   595.215148   679.105163   257.905827  -499.577450
## [111]  -591.625121   383.962847  2259.036748  -308.342849  3850.117969
## [116]  1174.313729 -1202.465388 -2901.551155   485.845142   982.375685
## [121]  3277.089110   111.972557  1580.239818  -560.737458 -4217.026051
## [126]  1487.359042  2462.791359  -446.245537  2181.741834  -281.838325
## [131]  4515.972026 -1516.162835 -4161.355810  1818.911795  3399.515909
```

```
##
## $avar
## [1] 1107900
##
## $aic
## [1] 13.97724
##
## $aicc
## [1] 14.9955
##
## $bic
## [1] 14.06332
##
## $se.phi
## [1] 0.0009161559 0.0008599616
##
## $se.theta
## [1] 0.001601026
```

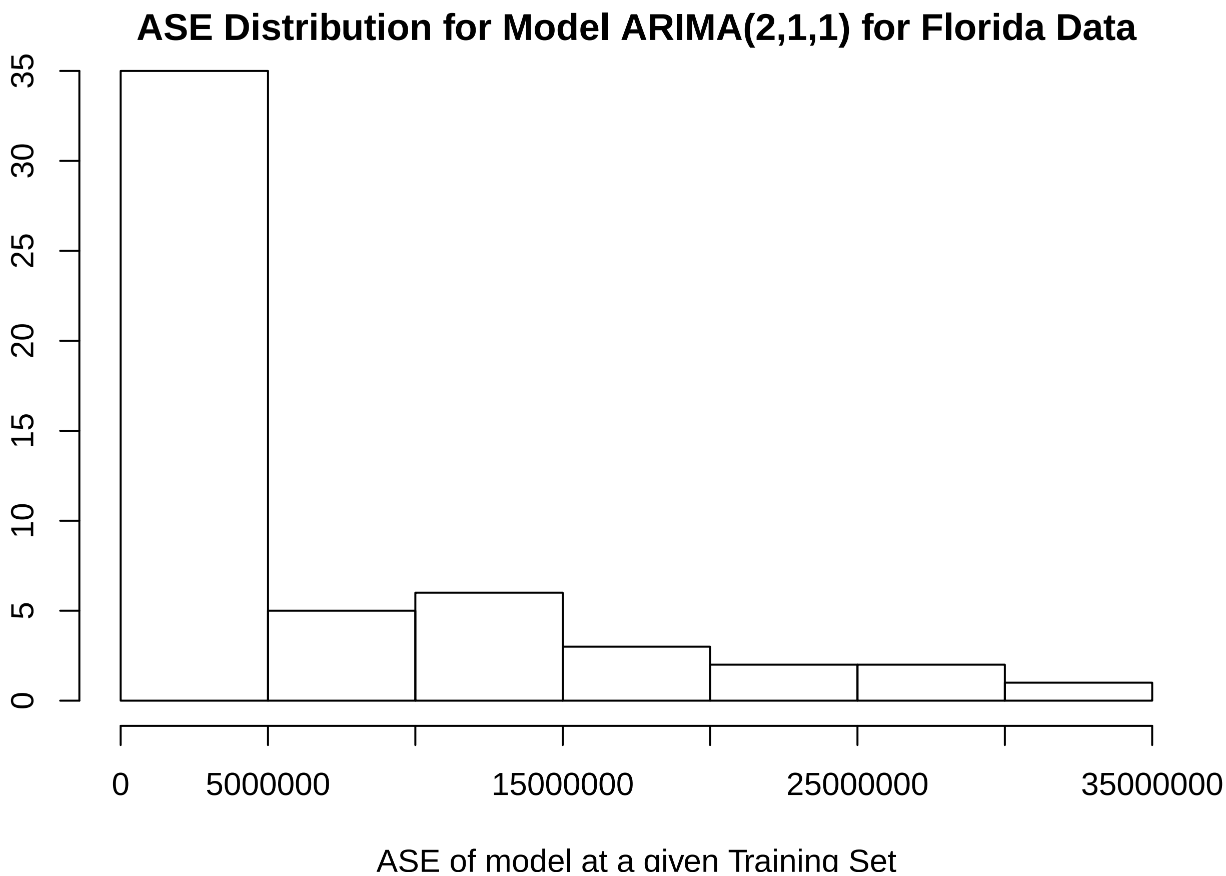
```
fl_arma_21$aic
```

```
## [1] 13.97724
```

```
trainingSize = 70
horizon = 12
ASEHolder = numeric()

for( i in 1:(135-(trainingSize + horizon) + 1))
{

  forecasts = fore.aruma.wge(newcases_fl$positiveIncrease[i:(i+(trainingSize-1))],phi = fl_arma_21$phi,

  ASE = mean((newcases_fl$positiveIncrease[(trainingSize+i):(trainingSize+ i + (horizon) - 1)] - forecas
  ASEHolder[i] = ASE

}

ASEHolder
```

```
##  [1]   167627.68   206397.80   137941.85   555602.91   521952.39
##  [6]   102153.75    87478.32   108624.36   183158.65   270619.99
## [11]    71089.92   115747.03   149317.00   133958.81   312340.25
## [16]   400954.76   247387.67    93284.93   158738.58   355742.35
## [21]   730532.31   931261.68   316047.95   439025.90   684184.64
## [26]  1021243.81  1771168.82  2899014.83  3188439.68  2658403.85
## [31]  2249266.64  2963186.04  2424907.70  7652197.48 14157083.00
## [36] 12795601.42 14103207.41 12339977.68 10590941.21 13337047.71
## [41] 19029966.10 29257992.40 30450023.78 15124656.10 17509801.44
## [46]  4594811.59  4654248.02  6537591.44 20214870.57 25595997.81
## [51] 23181259.41  8136658.95  8319256.74  8111410.44
```

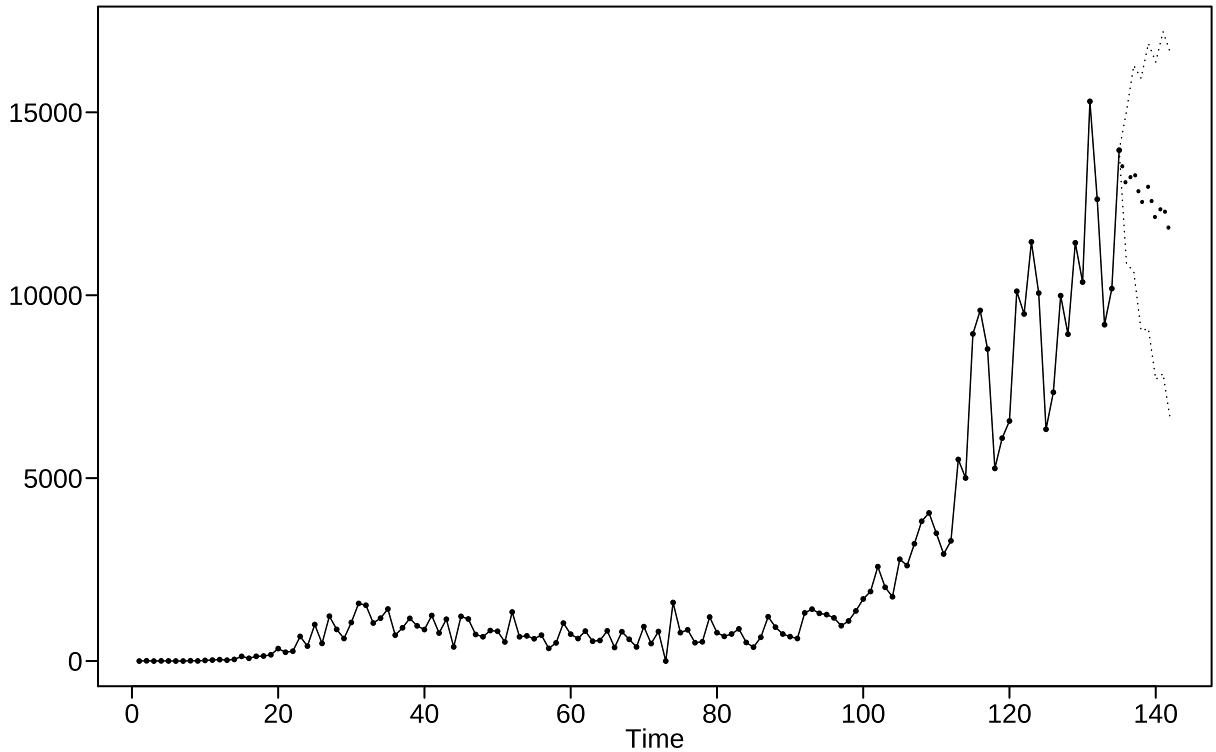
```
#Distribution of ASEs on Two Week Periods
hist(ASEHolder, xlab = "ASE of model at a given Training Set",  main = "ASE Distribution for Model ARIM
```



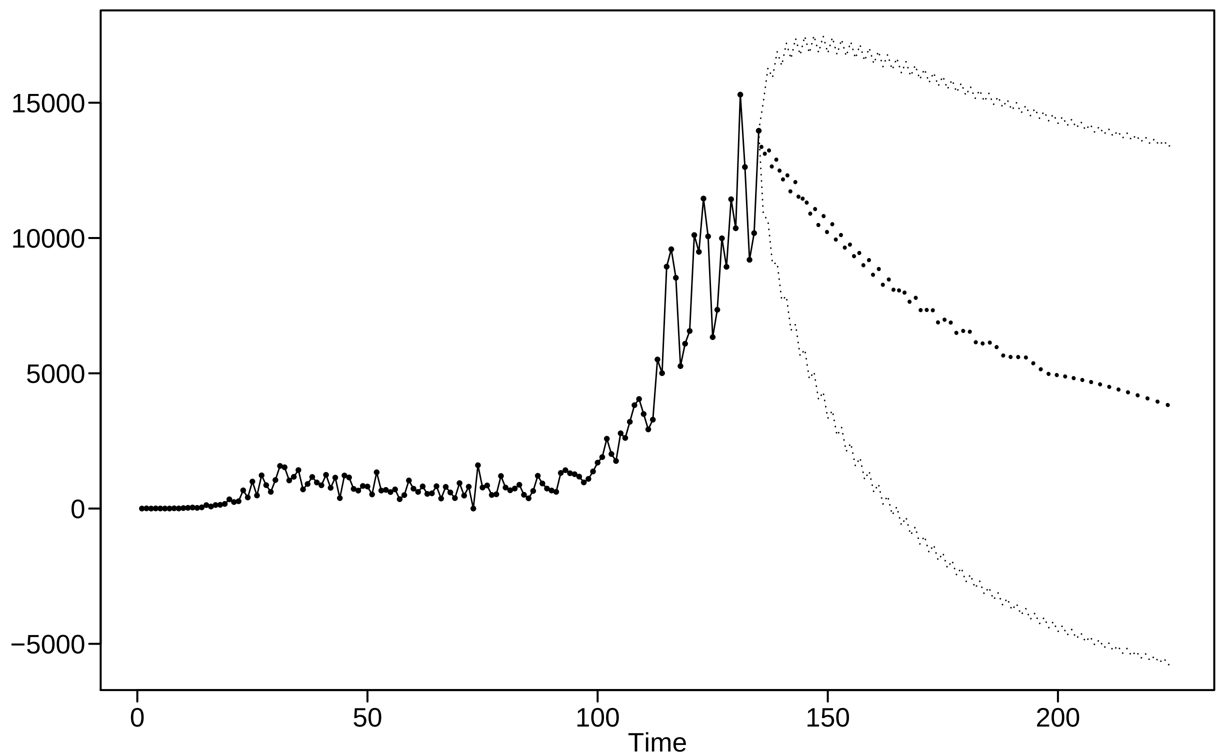

```r
#Mean ASE
WindowedASE = mean(ASEHolder)
WindowedASE
```

```
## [1] 6154656
```

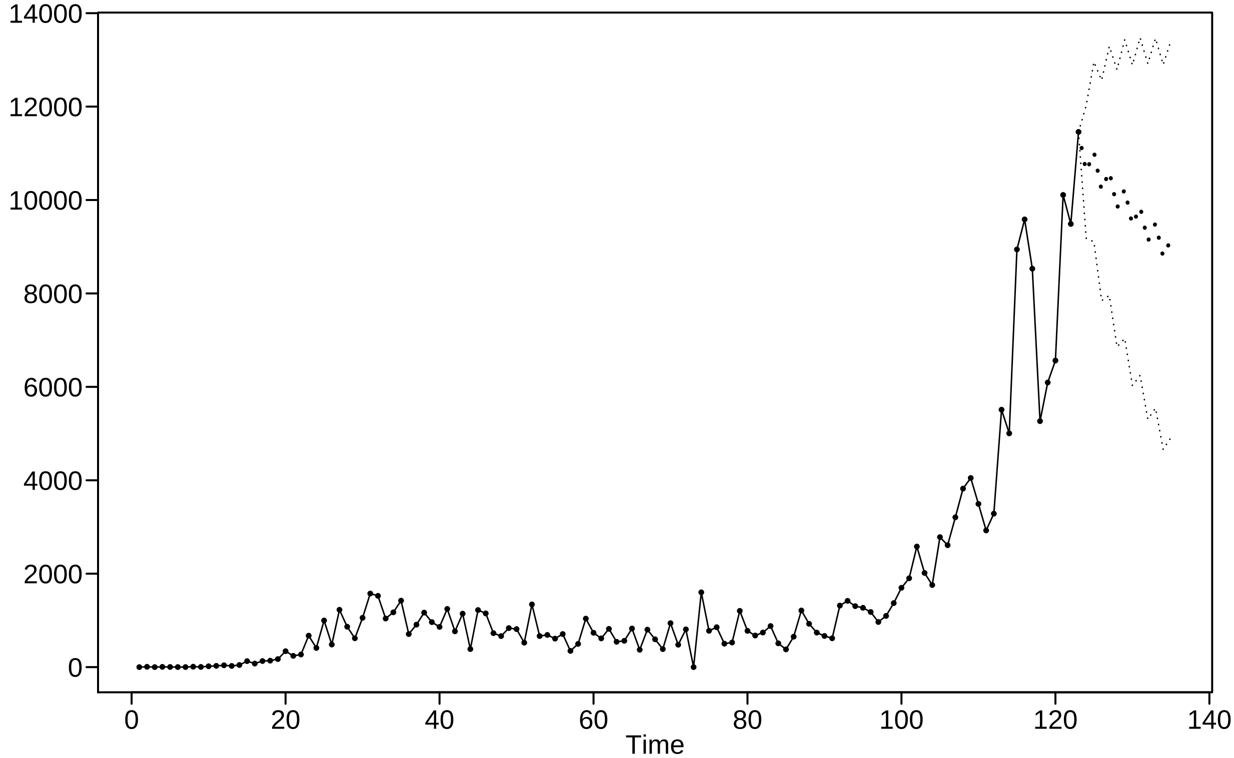
```r
##6154656



short_fl_arma = fore.aruma.wge(newcases_fl$positiveIncrease,phi = fl_arma_21$phi,theta = fl_arma_21$the
```



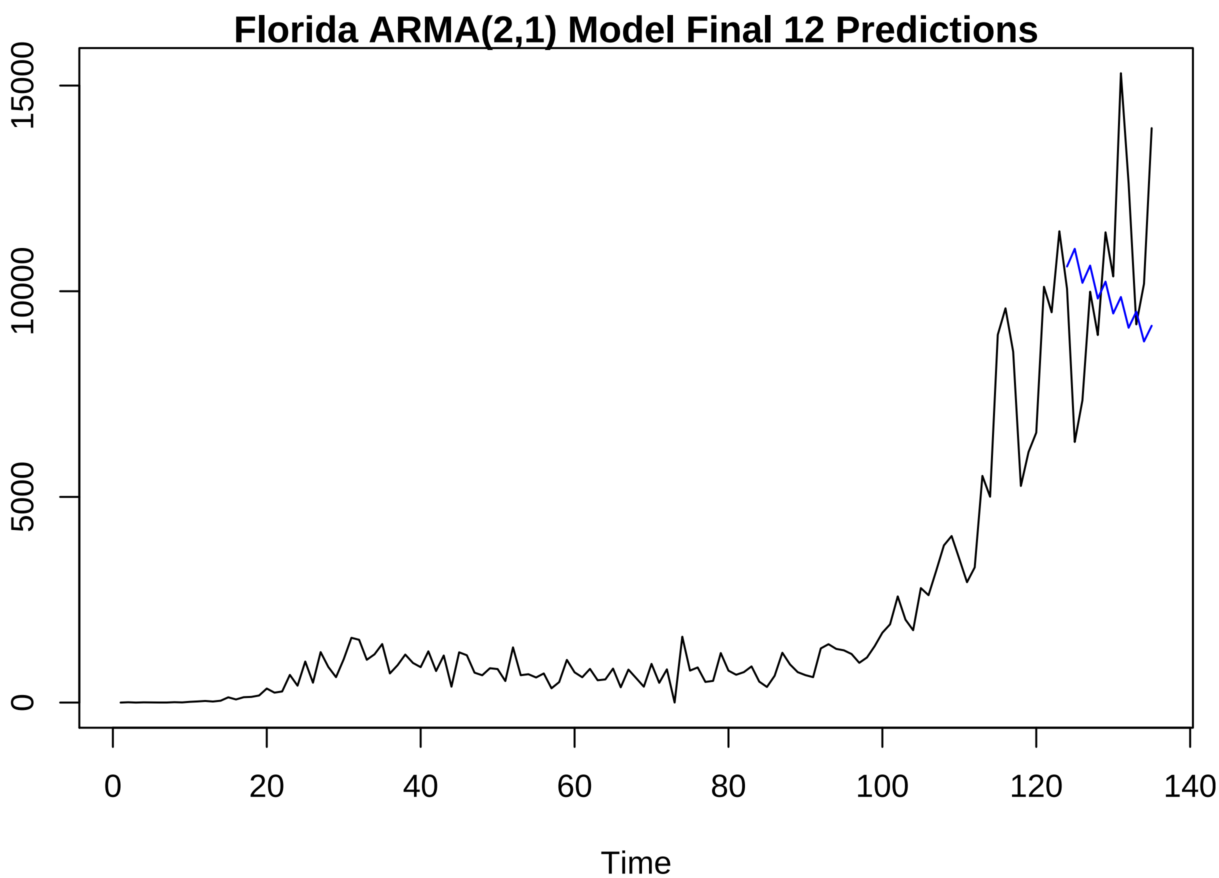

```
long_fl_arma = fore.aruma.wge(newcases_fl$positiveIncrease,phi = fl_arma_21$phi,theta = fl_arma_21$theta
```




```
final_pred = fore.aruma.wge(newcases_fl$positiveIncrease[1:123],phi = fl_arma_21$phi,theta = fl_arma_21$
```




```r
final_pred_df = data.frame(t = seq(124:135), final_pred$f)

plot(newcases_fl$positiveIncrease, type = "l", ylab = "Count of New Cases", xlab = "Time", main = "Flori
lines(ts(final_pred$f, start = 124, end = 135),  col = "blue")
```



MLP/RNN

####MLP Model for Florida Cases

```
trainingSize = 70
horizon = 12
ASEHolder = numeric()


for( i in 1:(135-(trainingSize + horizon) + 1))
{
  mlp.fit = mlp(ts(newcases_fl$positiveIncrease[1:trainingSize+i]), hd = 5, comb = "median")
  forecasts = forecast(mlp.fit,h = horizon)

  ASE = mean((newcases_fl$positiveIncrease[(trainingSize+i):(trainingSize+ i + (horizon) - 1)] -forecas
  print(c(i,ASE, "from",trainingSize+i,"to",(trainingSize+ i + (horizon) - 1)))
  ASEHolder[i] = ASE

}
```

```
## [1] "1"                 "126342.918469499" "from"
## [4] "71"                "to"               "82"
## [1] "2"                 "203427.338797191" "from"
## [4] "72"                "to"               "83"
## [1] "3"               "202306.26064129"  "from"             "73"
## [5] "to"              "84"
## [1] "4"                 "284046.527450543" "from"
## [4] "74"                "to"               "85"
```

```
## [1] "5"                "55677.4369127345" "from"
## [4] "75"               "to"               "86"
## [1] "6"                "81273.5077027466" "from"
## [4] "76"               "to"               "87"
## [1] "7"               "90196.751990875" "from"             "77"
## [5] "to"              "88"
## [1] "8"                "78371.4603359285" "from"
## [4] "78"               "to"               "89"
## [1] "9"                "97000.7560000339" "from"
## [4] "79"               "to"               "90"
## [1] "10"              "71035.427530876" "from"             "80"
## [5] "to"              "91"
## [1] "11"              "84238.334167012" "from"             "81"
## [5] "to"              "92"
## [1] "12"              "99151.343347452" "from"             "82"
## [5] "to"              "93"
## [1] "13"               "114862.564501097" "from"
## [4] "83"               "to"               "94"
## [1] "14"             "171924.6631314" "from"            "84"
## [5] "to"             "95"
## [1] "15"               "201490.449369577" "from"
## [4] "85"               "to"               "96"
## [1] "16"               "369038.059195622" "from"
## [4] "86"               "to"               "97"
## [1] "17"               "140020.693855445" "from"
## [4] "87"               "to"               "98"
## [1] "18"               "91332.7838252649" "from"
## [4] "88"               "to"               "99"
## [1] "19"               "141286.182739811" "from"
## [4] "89"               "to"               "100"
## [1] "20"               "286719.609782263" "from"
## [4] "90"               "to"               "101"
## [1] "21"               "1087389.27241766" "from"
## [4] "91"               "to"               "102"
## [1] "22"               "621874.074114186" "from"
## [4] "92"               "to"               "103"
## [1] "23"               "649123.617906168" "from"
## [4] "93"               "to"               "104"
## [1] "24"              "895489.11536653" "from"             "94"
## [5] "to"              "105"
## [1] "25"               "900578.947847859" "from"
## [4] "95"               "to"               "106"
## [1] "26"               "818374.168052311" "from"
## [4] "96"               "to"               "107"
## [1] "27"               "1884188.16659032" "from"
## [4] "97"               "to"               "108"
## [1] "28"              "2494000.3554607" "from"             "98"
## [5] "to"              "109"
## [1] "29"               "2990531.25669644" "from"
## [4] "99"               "to"               "110"
## [1] "30"               "2825937.90648963" "from"
## [4] "100"              "to"               "111"
## [1] "31"               "1927998.92499964" "from"
## [4] "101"              "to"               "112"
```

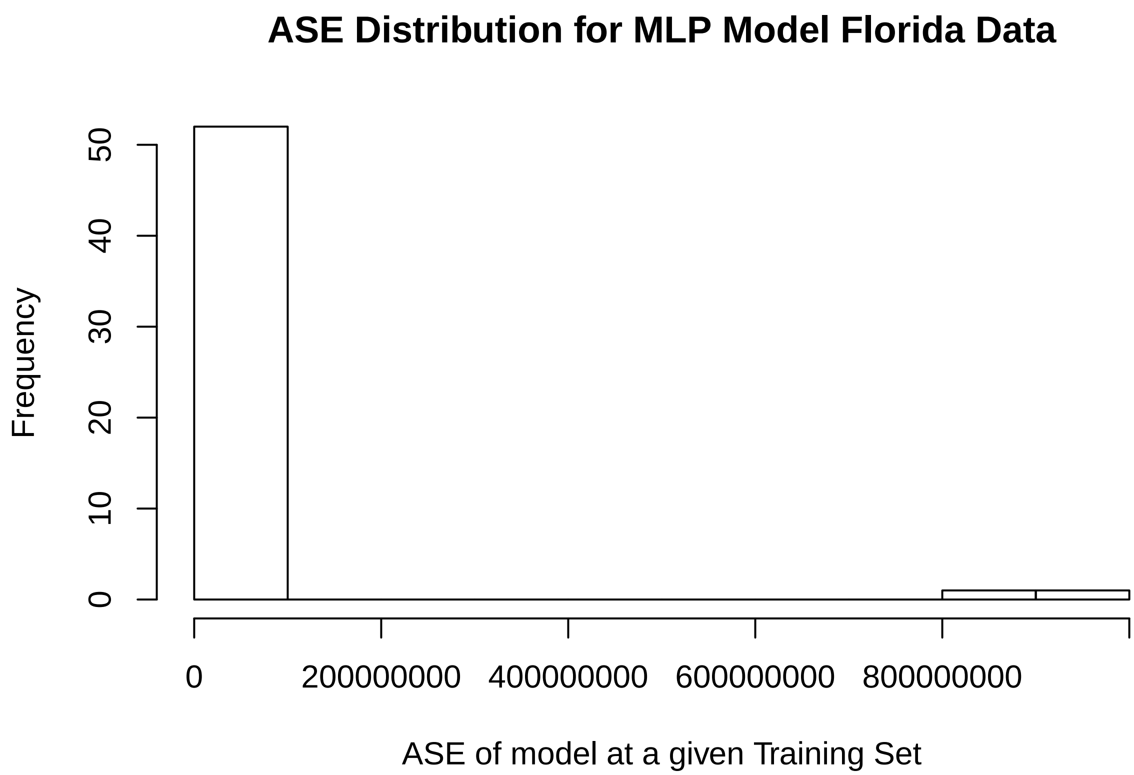
```
## [1] "32"               "1581107.68348046" "from"
## [4] "102"              "to"               "113"
## [1] "33"               "2923306.99821124" "from"
## [4] "103"              "to"               "114"
## [1] "34"               "6650372.26000452" "from"
## [4] "104"              "to"               "115"
## [1] "35"               "10285205.7251092" "from"
## [4] "105"              "to"               "116"
## [1] "36"               "10846027.0083773" "from"
## [4] "106"              "to"               "117"
## [1] "37"              "11261836.342726" "from"            "107"
## [5] "to"              "118"
## [1] "38"               "7276072.70437445" "from"
## [4] "108"              "to"               "119"
## [1] "39"               "6691664.73783155" "from"
## [4] "109"              "to"               "120"
## [1] "40"               "11968707.1910894" "from"
## [4] "110"              "to"               "121"
## [1] "41"               "17487960.6676839" "from"
## [4] "111"              "to"               "122"
## [1] "42"               "21735678.9638669" "from"
## [4] "112"              "to"               "123"
## [1] "43"               "13814964.8826825" "from"
## [4] "113"              "to"               "124"
## [1] "44"               "14154489.5491293" "from"
## [4] "114"              "to"               "125"
## [1] "45"               "871794356.846582" "from"
## [4] "115"              "to"               "126"
## [1] "46"               "950390431.376566" "from"
## [4] "116"              "to"               "127"
## [1] "47"               "6105963.60477521" "from"
## [4] "117"              "to"               "128"
## [1] "48"               "30610449.9836053" "from"
## [4] "118"              "to"               "129"
## [1] "49"               "12212584.3431474" "from"
## [4] "119"              "to"               "130"
## [1] "50"               "15882466.8424715" "from"
## [4] "120"              "to"               "131"
## [1] "51"               "16610681.7019103" "from"
## [4] "121"              "to"               "132"
## [1] "52"               "16718582.1168852" "from"
## [4] "122"              "to"               "133"
## [1] "53"               "6450903.68375201" "from"
## [4] "123"              "to"               "134"
## [1] "54"               "6215708.90216226" "from"
## [4] "124"              "to"               "135"
```

```
ASEHolder
```

```
##  [1]    126342.92    203427.34    202306.26    284046.53     55677.44
##  [6]     81273.51     90196.75     78371.46     97000.76     71035.43
## [11]     84238.33     99151.34    114862.56    171924.66    201490.45
## [16]    369038.06    140020.69     91332.78    141286.18    286719.61
## [21]   1087389.27    621874.07    649123.62    895489.12    900578.95
## [26]    818374.17   1884188.17   2494000.36   2990531.26   2825937.91
```

```
## [31]   1927998.92   1581107.68   2923307.00   6650372.26  10285205.73
## [36]  10846027.01  11261836.34   7276072.70   6691664.74  11968707.19
## [41]  17487960.67  21735678.96  13814964.88  14154489.55 871794356.85
## [46] 950390431.38   6105963.60  30610449.98  12212584.34  15882466.84
## [51]  16610681.70  16718582.12   6450903.68   6215708.90
```

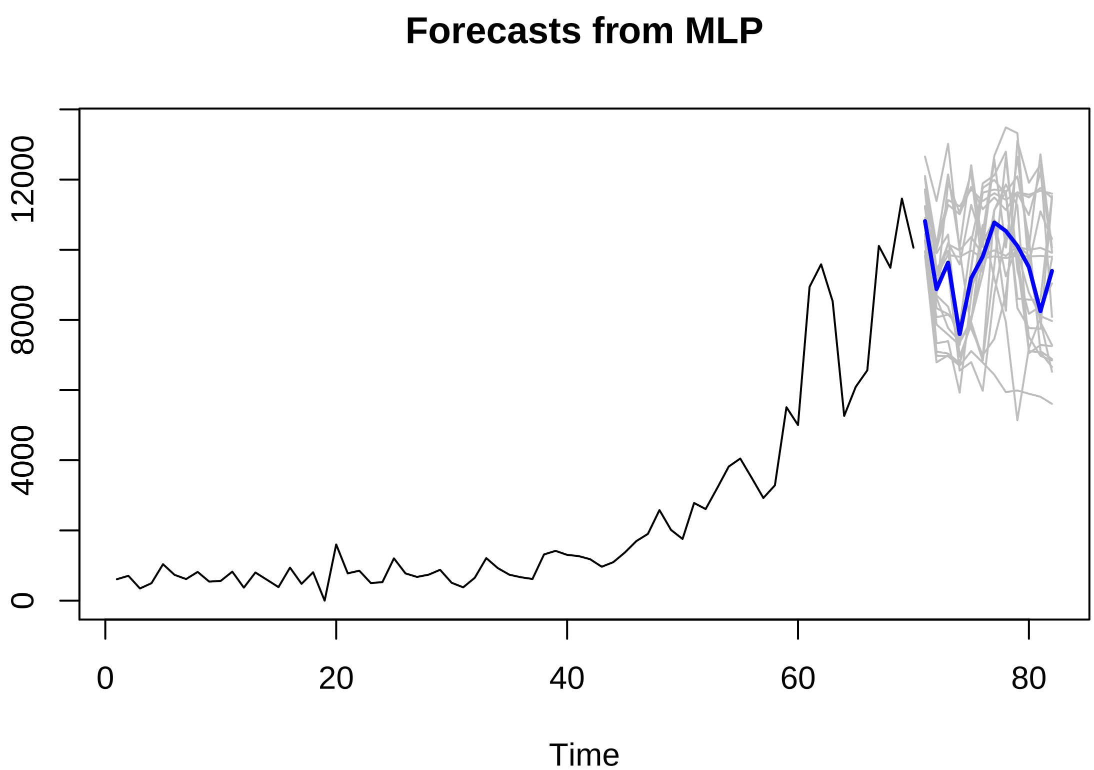
```r
#Distribution of ASEs on Two Week Periods
hist(ASEHolder, xlab = "ASE of model at a given Training Set",  main = "ASE Distribution for MLP Model F
```



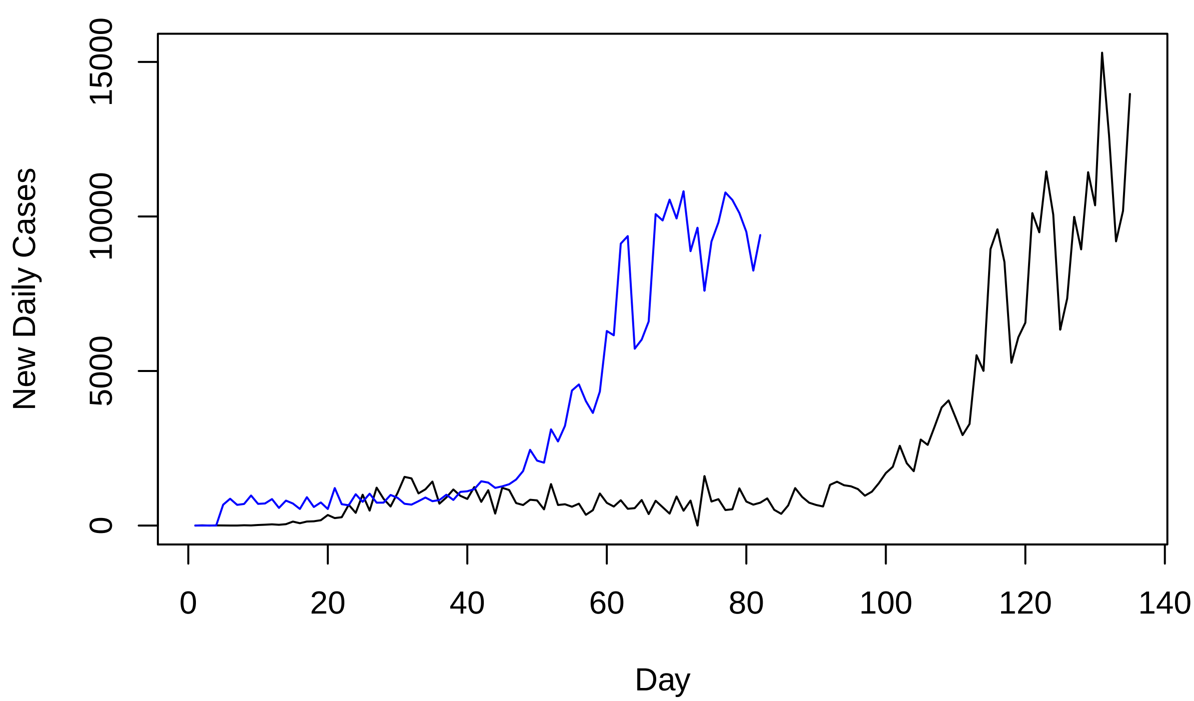

```r
#Mean ASE
WindowedASE = mean(ASEHolder)
WindowedASE
```

```
## [1] 38699162
```

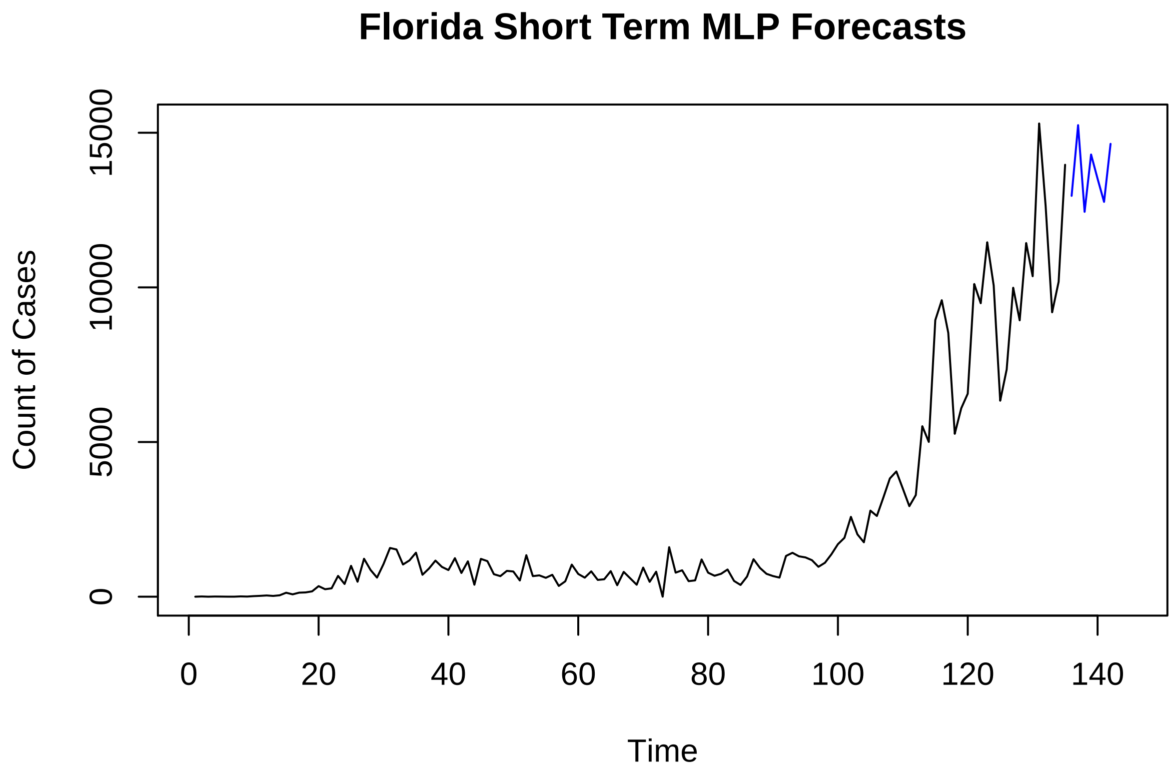
```r
plot(forecasts)
```



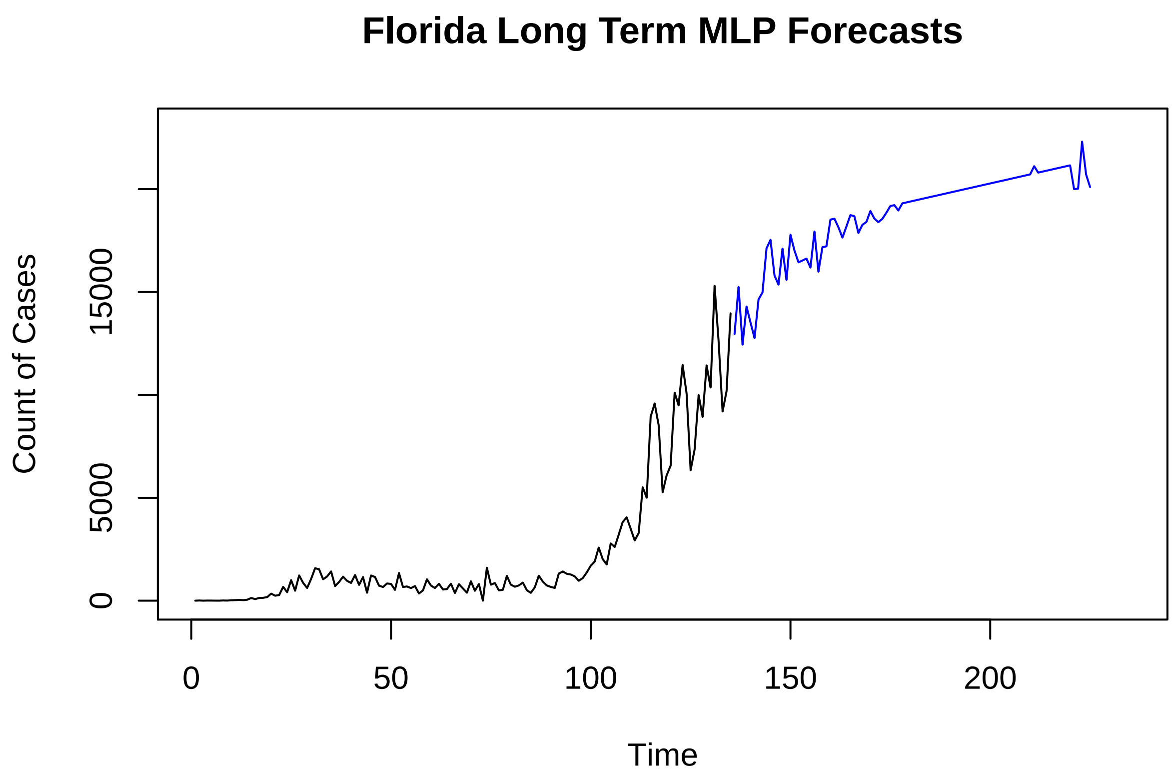

```r
#Final Forecasts with data known
mlp.fit_fl_final = mlp(ts(newcases_fl$positiveIncrease[1:123]), hd = 5, comb = "median")
forecasts_fl_mlp = forecast(mlp.fit,h = 12)

all_f = c(rep(1,4), forecasts$fitted, forecasts$mean)
plot(newcases_fl$positiveIncrease, type = "l", ylab = "New Daily Cases", xlab = "Day", main = "")
lines(all_f, col = "blue")
```



```
#Future Forecasts

mlp.fit_fl_future =mlp(ts(newcases_fl$positiveIncrease), hd = 5, comb = "median")
short_fl_mlp = forecast(mlp.fit_fl_future,h = 7)

long_fl_mlp = forecast(mlp.fit_fl_future,h = 90)

plot(newcases_fl$positiveIncrease, type = "l", xlim = c(1,145), main = "Florida Short Term MLP Forecasts
lines(short_fl_mlp$mean, col = "blue")
```




```
plot(newcases_fl$positiveIncrease, type = "l", xlim = c(1,235), ylim = c(0,23000), main = "Florida Long
lines(long_fl_mlp$mean, col = "blue")
```



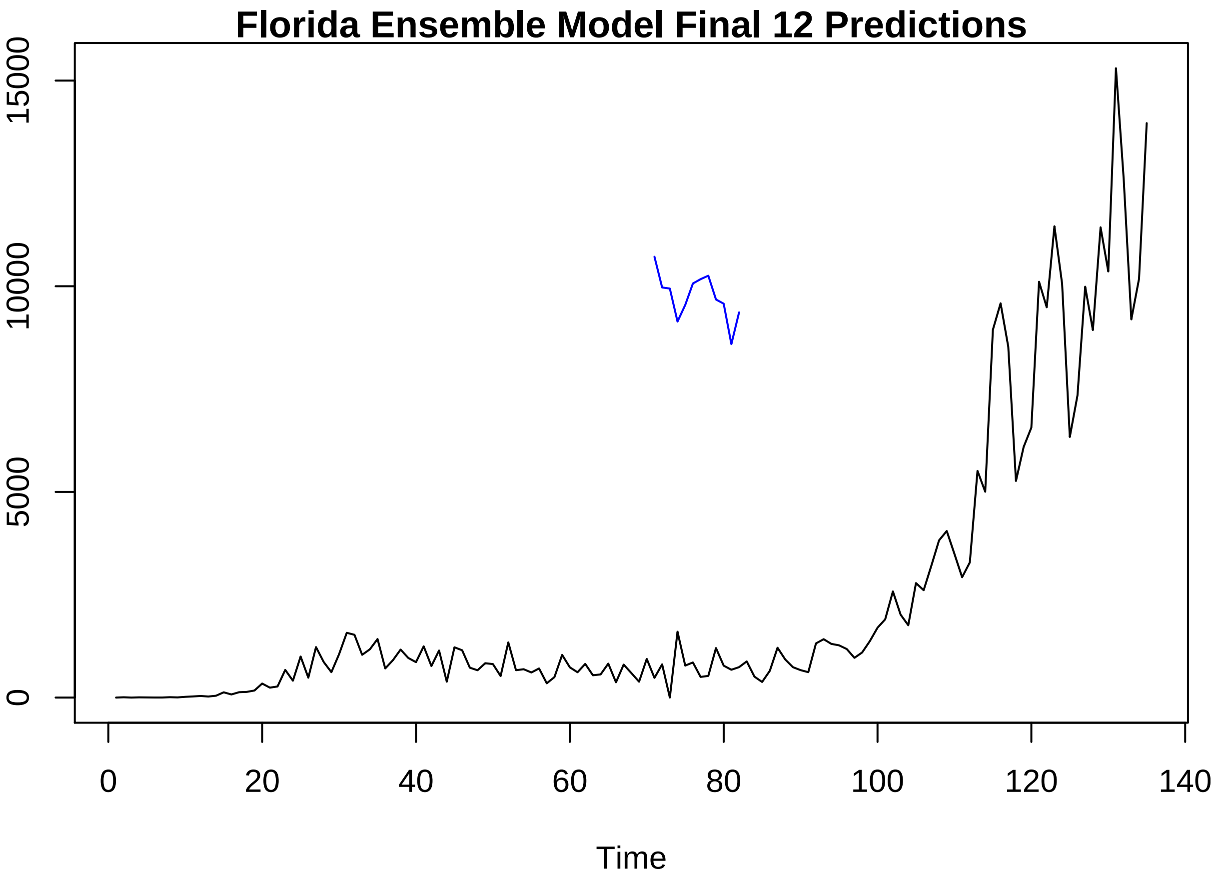

```r
mlp.fit_fl_full = mlp(ts(newcases_fl$positiveIncrease), hd = 5)
forecasts_fl_short = forecast(mlp.fit,h = 7)
forecasts_fl_long = forecast(mlp.fit,h = 90)
```

####Florida Cases Ensemble Model

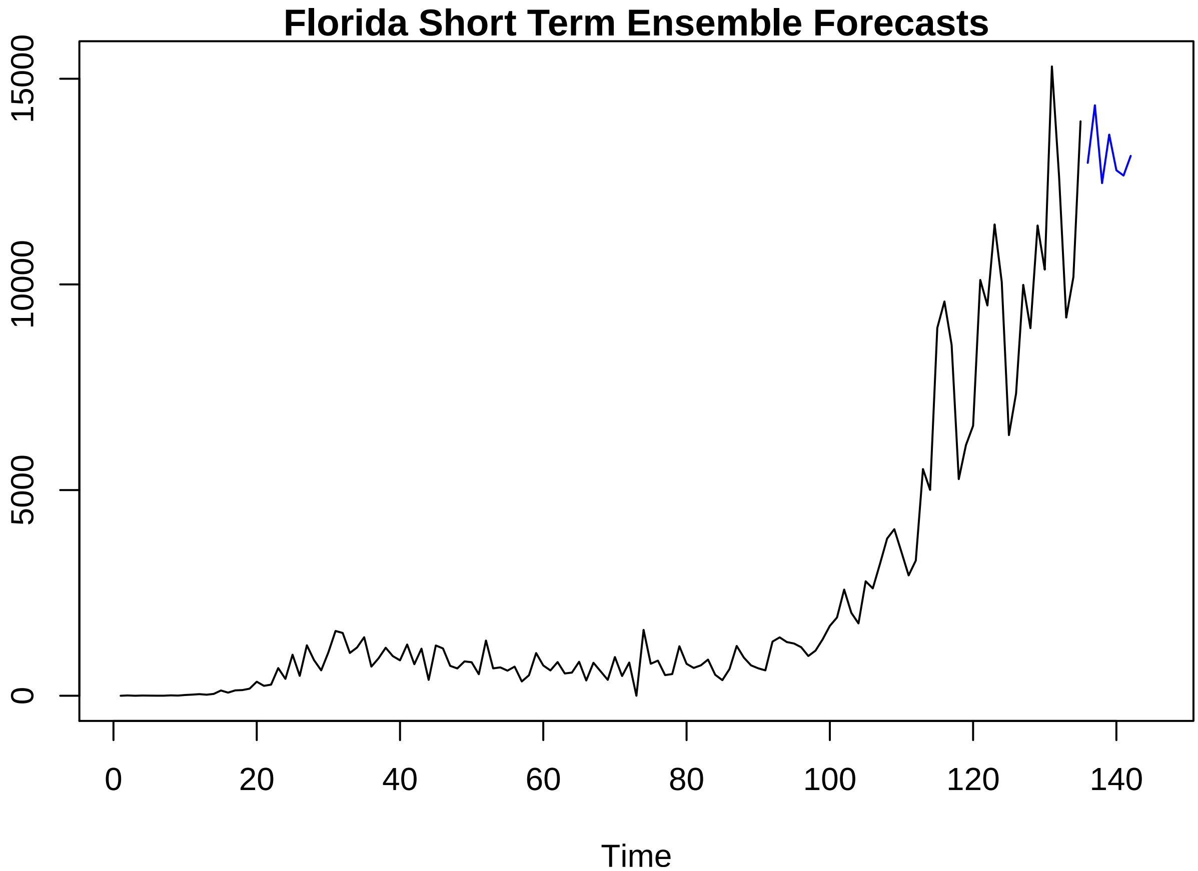
```r
#ASE fits for ensemble
mlp.fit_fl_final = mlp(ts(newcases_fl$positiveIncrease[1:123]), hd = 5, comb = "median")
forecasts_fl_mlp = forecast(mlp.fit,h = 12)

forecasts_fl_arma = fore.aruma.wge(newcases_fl$positiveIncrease[i:(i+(trainingSize-1))],phi = fl_arma_2

ensemble_fl_fore = (forecasts_fl_mlp$mean + forecasts_fl_arma$f) / 2

ensemble_ASE = mean((newcases_fl$positiveIncrease[124:135] - ensemble_fl_fore)^2)
ensemble_ASE
```

```
## [1] 6777125
```

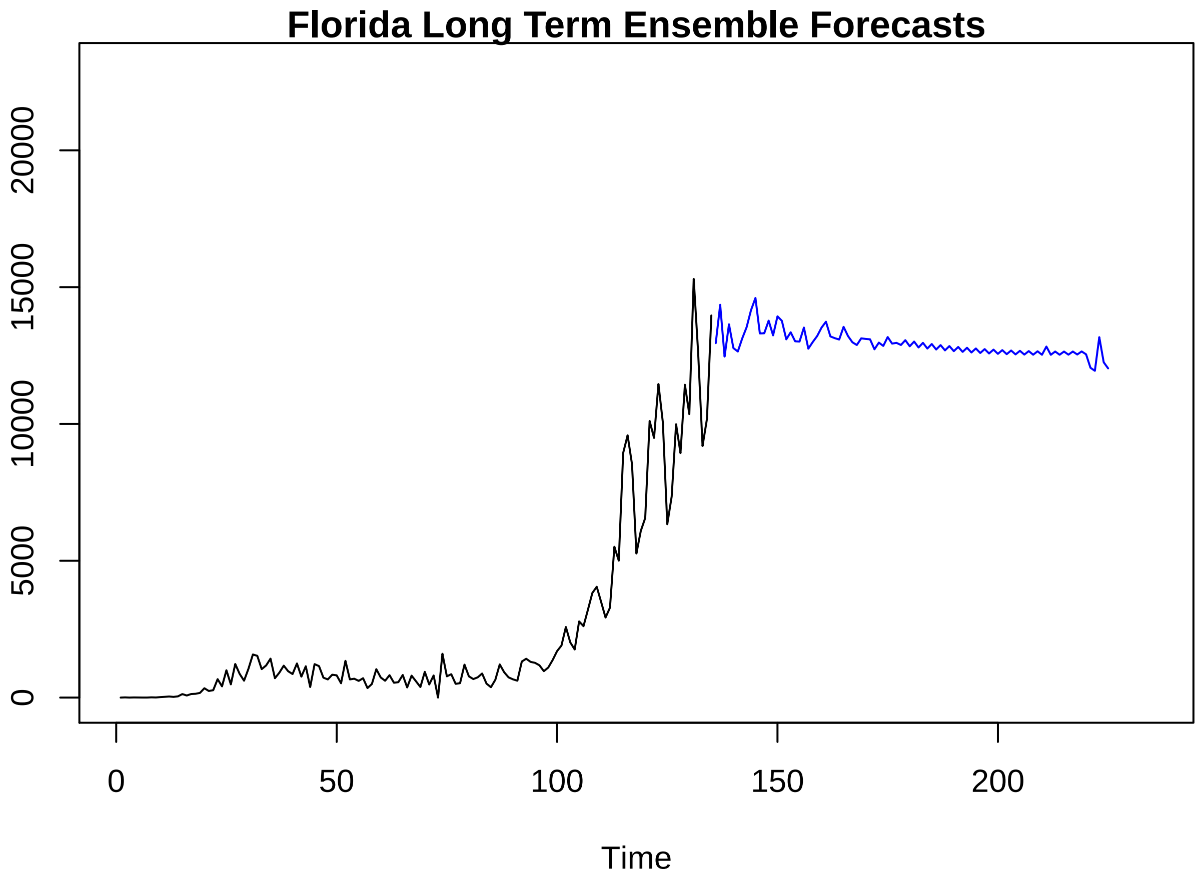
```r
#8.4 Mill

plot(newcases_fl$positiveIncrease, type = "l", ylab = "Count of New Cases", xlab = "Time", main = "Flor
lines(ensemble_fl_fore, col = "blue")
```




```r
#future forecasting

short_fl_ensemble = (short_fl_mlp$mean + short_fl_arma$f)/2
long_fl_ensemble = (long_fl_mlp$mean + long_fl_arma$f)/2



plot(newcases_fl$positiveIncrease, type = "l", xlim = c(1,145), main = "Florida Short Term Ensemble Fore
lines(short_fl_ensemble, col = "blue")
```




```
plot(newcases_fl$positiveIncrease, type = "l", xlim = c(1,235), ylim = c(0,23000), main = "Florida Long
lines(long_fl_ensemble, col = "blue")
```



Comparing and Assessing Models

###Model Building for Cases United States

Stationarity vs non-stationarity - concerns about the data

Model IDing of stationary modls

####US Cases ARMA(1,2)

```
aic5.wge(newcases_us$positiveIncrease)
```

```
## ---------WORKING... PLEASE WAIT...
##
##
## Error in aic calculation at 1 1
## Error in aic calculation at 2 0
## Error in aic calculation at 2 1
## Error in aic calculation at 2 2
## Error in aic calculation at 3 0
## Error in aic calculation at 3 1
## Error in aic calculation at 3 2
## Error in aic calculation at 4 0
## Error in aic calculation at 4 1
## Error in aic calculation at 4 2
## Error in aic calculation at 5 0
## Error in aic calculation at 5 1
## Error in aic calculation at 5 2
## Five Smallest Values of  aic
```

```
##      p    q        aic
## 6    1    2   15.95786
## 4    1    0   15.96008
## 3    0    2   17.57501
## 2    0    1   18.44652
## 1    0    0   19.55796
```

```r
aic5.wge(newcases_us$positiveIncrease, type = 'bic')
```

```
## ---------WORKING... PLEASE WAIT...
##
##
## Error in aic calculation at 1 1
## Error in aic calculation at 2 0
## Error in aic calculation at 2 1
## Error in aic calculation at 2 2
## Error in aic calculation at 3 0
## Error in aic calculation at 3 1
## Error in aic calculation at 3 2
## Error in aic calculation at 4 0
## Error in aic calculation at 4 1
## Error in aic calculation at 4 2
## Error in aic calculation at 5 0
## Error in aic calculation at 5 1
## Error in aic calculation at 5 2
## Five Smallest Values of  bic
```

```
##      p    q        bic
## 4    1    0   15.99597
## 6    1    2   16.02964
## 3    0    2   17.62884
## 2    0    1   18.48240
## 1    0    0   19.57590
```

```r
us_arma_12 = est.arma.wge(newcases_us$positiveIncrease, p = 1, q = 2)
```

```
##
## Coefficients of Original polynomial:
## 0.9919
##
## Factor                 Roots                Abs Recip    System Freq
## 1-0.9919B              1.0081               0.9919       0.0000
##
##
```

```r
us_arma_12
```

```
## $phi
## [1] 0.9919401
##
## $theta
## [1] -0.18789321 -0.06864411
##
## $res
##    [1]  -210.30561  -105.19842  -127.43409  -130.47115  -128.37416
##    [6]  -128.55969  -128.66878  -127.63555  -129.81414  -128.48378
##   [11]  -127.58420  -127.83648  -129.84277  -128.45655  -129.57929
```

```
##  [16]  -125.47155  -131.14212  -128.37475  -129.50547  -125.49104
##  [21]  -131.14352  -128.37315  -129.50567  -125.49111  -125.14350
##  [26]  -129.45215  -120.66644  -135.95697  -126.73543  -125.41043
##  [31]  -131.26822  -126.27465  -125.79469  -119.20347  -125.39426
##  [36]  -131.64320  -130.06023  -114.94482  -146.78879   -60.97207
##  [41]  -146.38665  -111.28506   -42.87213  -178.43476  -135.93365
##  [46]   -43.71046   -95.31929   -38.57411   -70.88751   -56.11484
##  [51]    13.02075   188.26586  -299.64272   225.11586   144.71745
##  [56]  1852.36521  -932.49698  1411.97729  1258.72068   185.34902
##  [61]  2146.62614  1679.93024 -1342.41623  2301.96958  4384.74267
##  [66]   386.78843   298.19385  -154.24311  2247.29209  2400.18728
##  [71]   511.15148  2004.28971  3605.29519   587.84261 -7953.56392
##  [76]  4885.36167  1499.63077  -734.02229  4004.44029  -654.24496
##  [81] -3826.72528 -1868.39333 -2034.47714  1049.45065  4654.41605
##  [86]  -239.01058   854.00880 -4049.06688   270.91090 -1309.20594
##  [91]   735.28271  2581.31849  2411.90518  1850.19746  1327.59857
##  [96] -8769.73135 -3785.31636  4553.73687  1472.72772  1856.53884
## [101]  3064.16341 -4469.38568 -2714.97901 -2381.03492   579.64339
## [106]  2680.83232  1930.81212  -371.88073 -2796.80904 -2716.64248
## [111] -2553.89407  5078.68403 -2227.91628  5591.16475 -2839.00643
## [116]   178.15218 -4331.14236  1623.50212  -139.18146   387.14321
## [121]  5141.00412 -2898.48024 -2737.02150  -759.39272  -976.02781
## [126] -1943.61036  3160.58072  2750.16971   207.83099    40.11459
## [131] -2114.92169  -816.60334  -131.48326   410.17550   461.03180
## [136]  2404.54211  -702.86684 -4291.39907  -966.35329   614.62840
## [141]  3555.64162   524.76130  1118.26545  1645.44863 -4311.32250
## [146] -2155.25520  5613.45629  -441.53835  3382.40793  2974.00485
## [151]   245.61226 -4864.06610   688.33391  6264.67302  4543.16860
## [156]  -745.51228  5330.49330 -1391.99789 -1839.11166 -4767.16587
## [161]  9030.29307  7484.31739   188.46366  1351.13580 -4456.45609
## [166] -8486.20002  6910.13848  3840.80225  9440.58925 -5009.21144
## [171]  8414.72935 -4499.70477 -1414.96658 -1608.42153  5122.92381
## [176]  1720.00550  5811.27282
## 
## $avar
## [1] 8142984
## 
## $aic
## [1] 15.95786
## 
## $aicc
## [1] 16.97115
## 
## $bic
## [1] 16.02964
## 
## $se.phi
## [1] 0.00008963278
## 
## $se.theta
## [1] 0.008017396 0.009972818
```

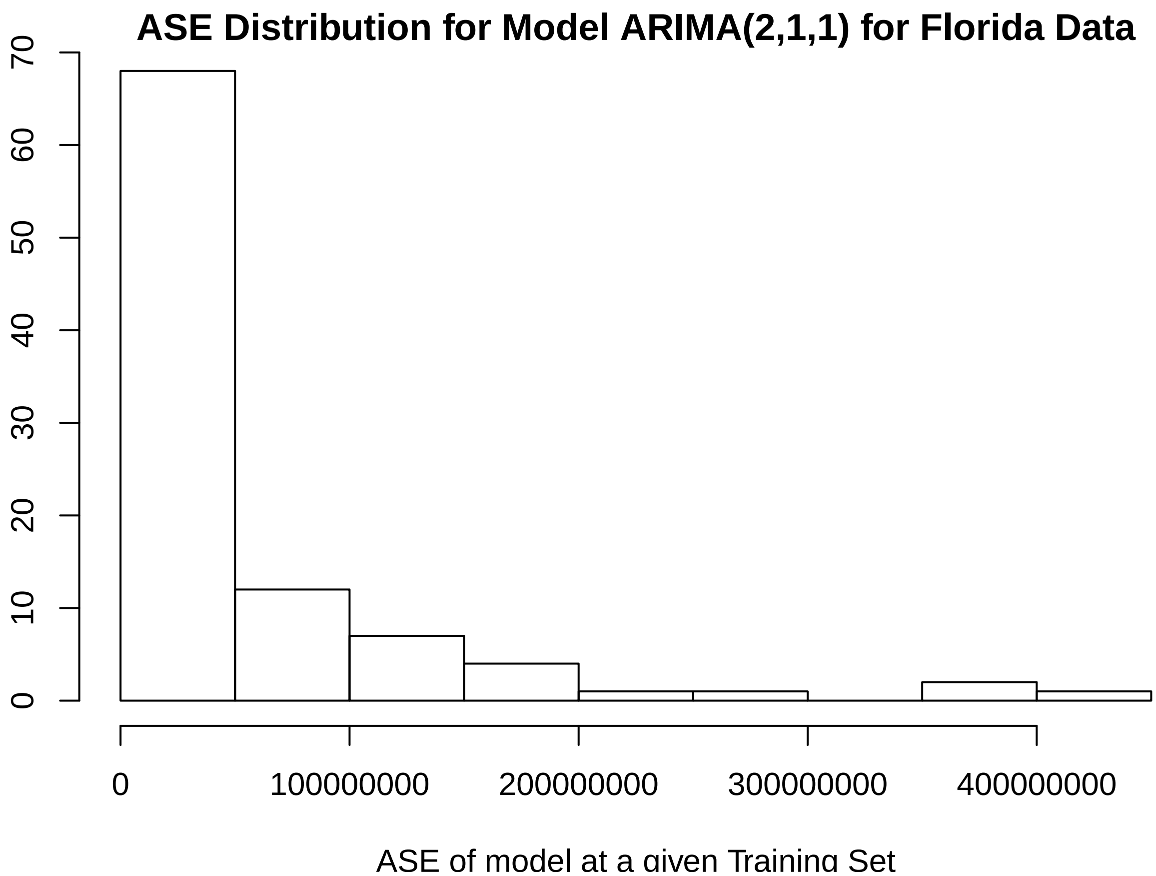
```r
us_arma_12$aic
```

```
## [1] 15.95786
```

```r
trainingSize = 70
horizon = 12
ASEHolder = numeric()

for( i in 1:(177-(trainingSize + horizon) + 1))
{

  forecasts = fore.aruma.wge(newcases_us$positiveIncrease[i:(i+(trainingSize-1))],phi = us_arma_12$phi,

  ASE = mean((newcases_us$positiveIncrease[(trainingSize+i):(trainingSize+ i + (horizon) - 1)] - foreca
  ASEHolder[i] = ASE

}

ASEHolder
```

```
##  [1]  43725973  35579734  15816563  12379608  16077405  63231777   9750780
##  [8]   7422003   7827994  34240666  28525306   6458571  16896630  48641664
## [15]  34877228  15300586  16212745  21221178  19457996  22136822  33803340
## [22]  26312425  16649648  31616011  58839891  87361504  13603088  51265548
## [29]  10098585  16291054  32135503  88968495  21834753   7680905  13124772
## [36]  10367473  12061241  30181559  28890359   8390464  11498325  35163223
## [43]   5970811   9211896  41114819  15001253  16565415  11439564   7133678
## [50]   7371227   7024688  47162686  15221181   3704723   5158051  10417502
## [57]  28158001   5428329  12670962  14728236  15172694   4829085   6396597
## [64]   6776219   6560676   6840061  15007220  12262542  35490861  65797401
## [71]  63948556  24692884  30161102  40875386  45953055 147521940 259676867
## [78] 135521118 159048021 109843644 102674520 144975958 355419374 383704325
## [85] 176877267  88357915 117944780  72816056 113223793 210599000 446366735
## [92] 160337015  55348712  62897740  66463336 160673404
```

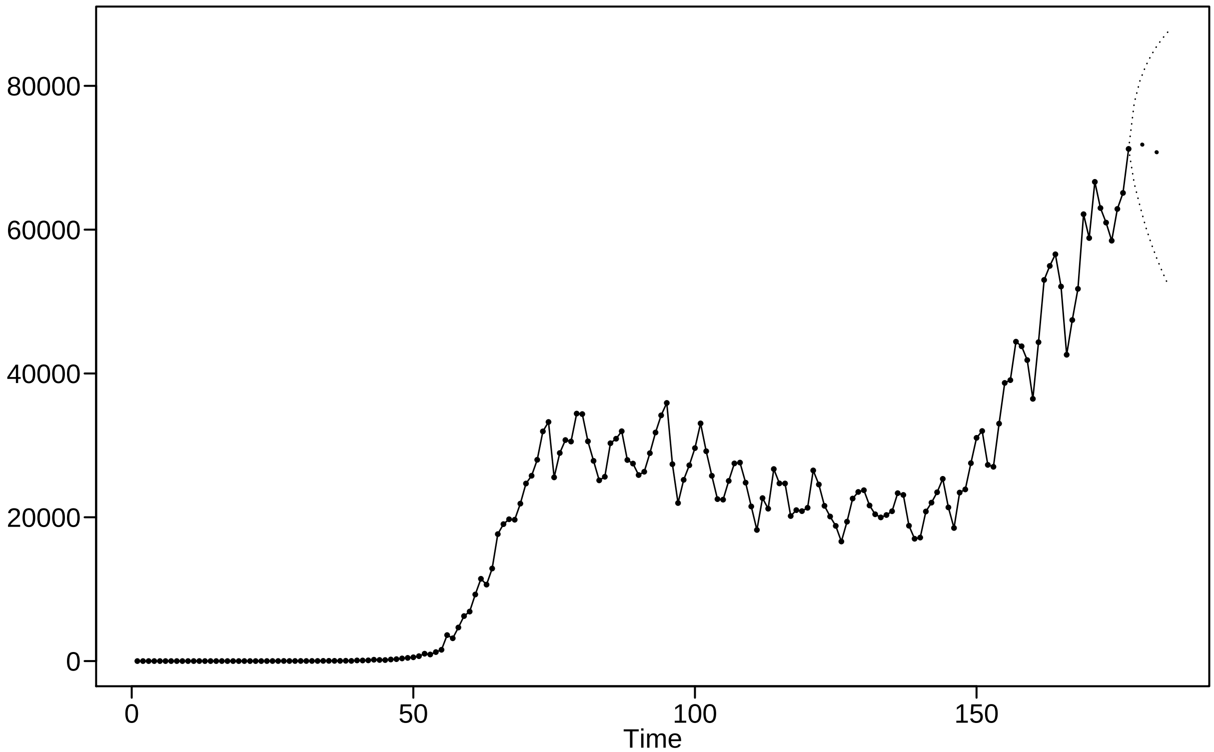
```r
#Distribution of ASEs on Two Week Periods
hist(ASEHolder, xlab = "ASE of model at a given Training Set",  main = "ASE Distribution for Model ARIM
```



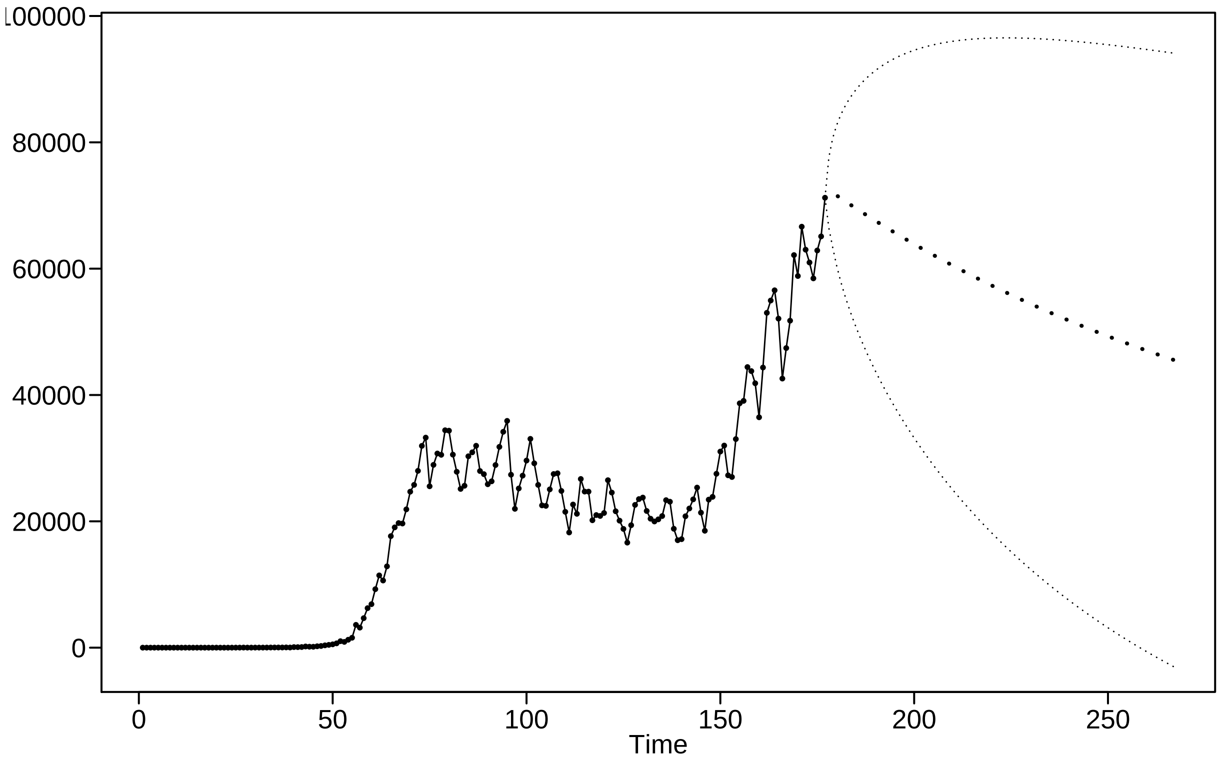

```
#Mean ASE
WindowedASE = mean(ASEHolder)
WindowedASE
```

```
## [1] 55171440
```

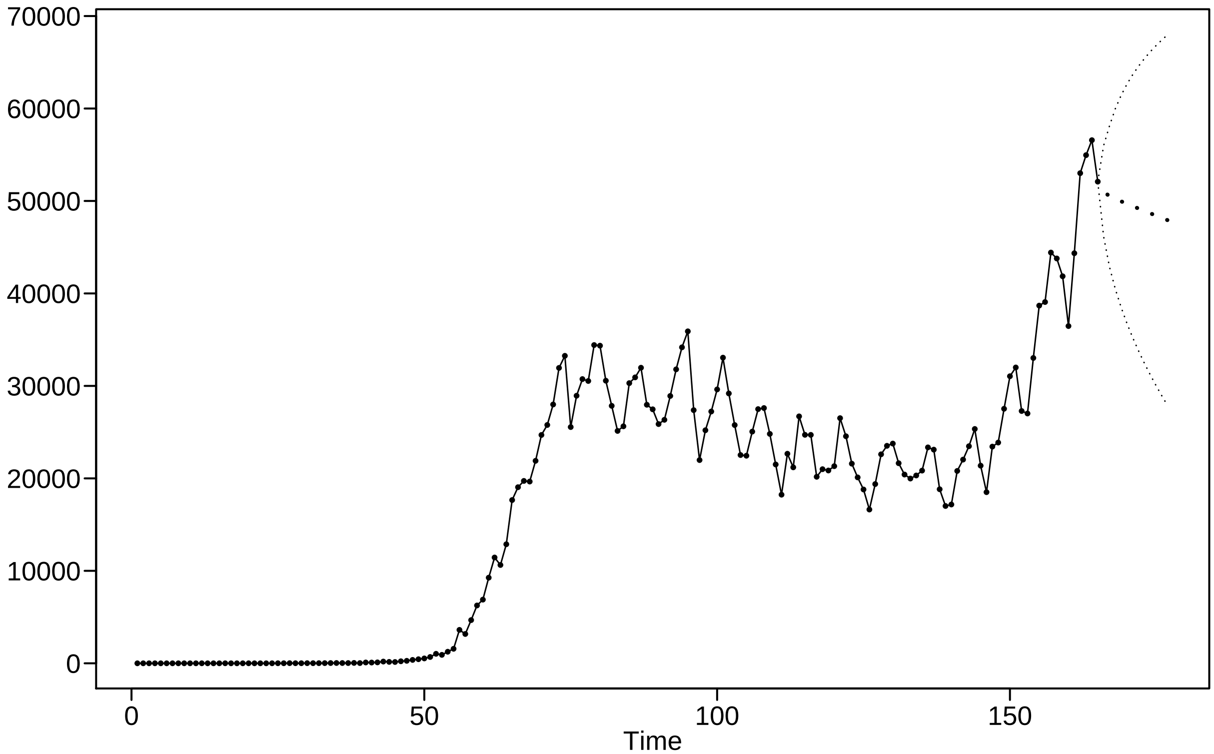
```
#55171440


short_us_ar1 = fore.aruma.wge(newcases_us$positiveIncrease,phi = us_arma_12$phi,theta = us_arma_12$theta
```



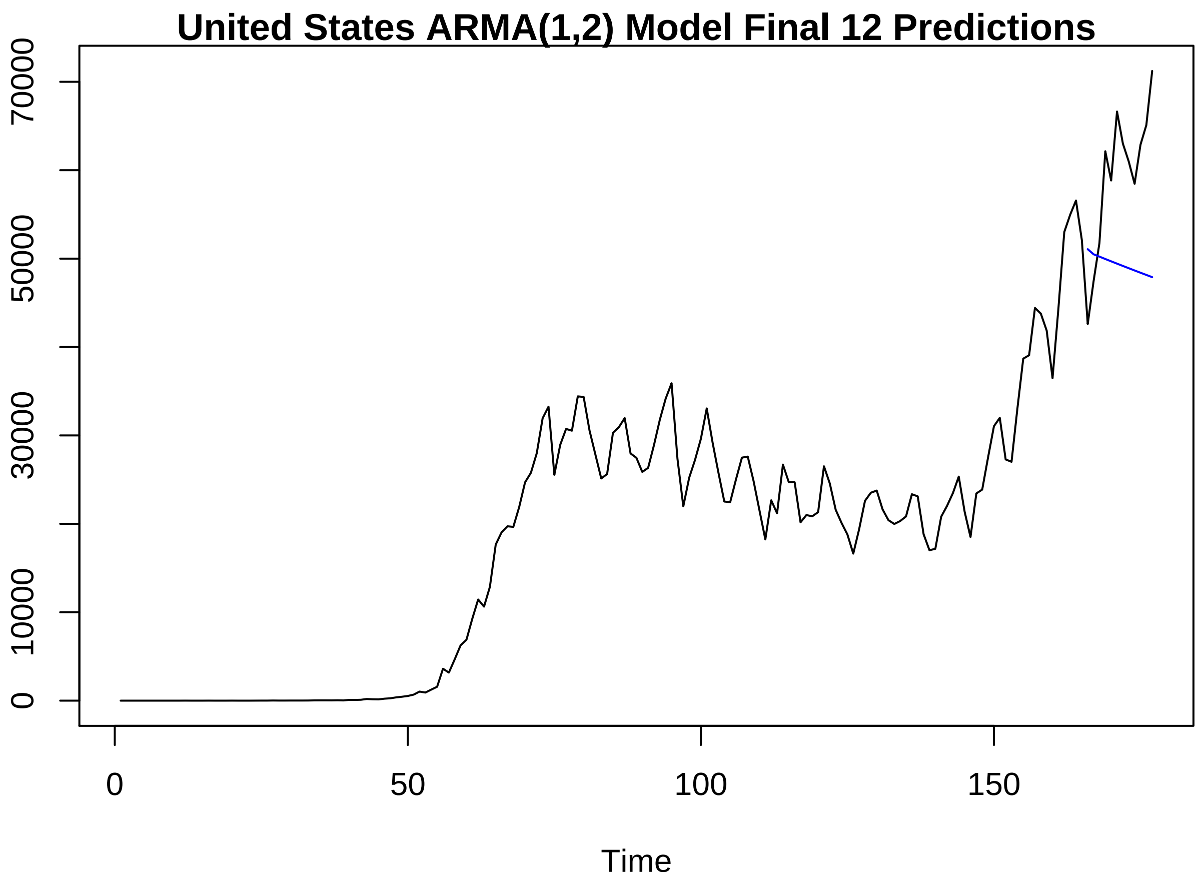

```
long_us_ar1 = fore.aruma.wge(newcases_us$positiveIncrease,phi = us_arma_12$phi,theta = us_arma_12$theta
```




```
final_pred = fore.aruma.wge(newcases_us$positiveIncrease[1:165],phi = us_arma_12$phi,theta = us_arma_12
```




```
final_pred_df = data.frame(t = seq(166:177), final_pred$f)

plot(newcases_us$positiveIncrease, type = "l", ylab = "Count of New Cases", xlab = "Time", main = "Unit
lines(ts(final_pred$f, start = 166, end = 177),  col = "blue")
```

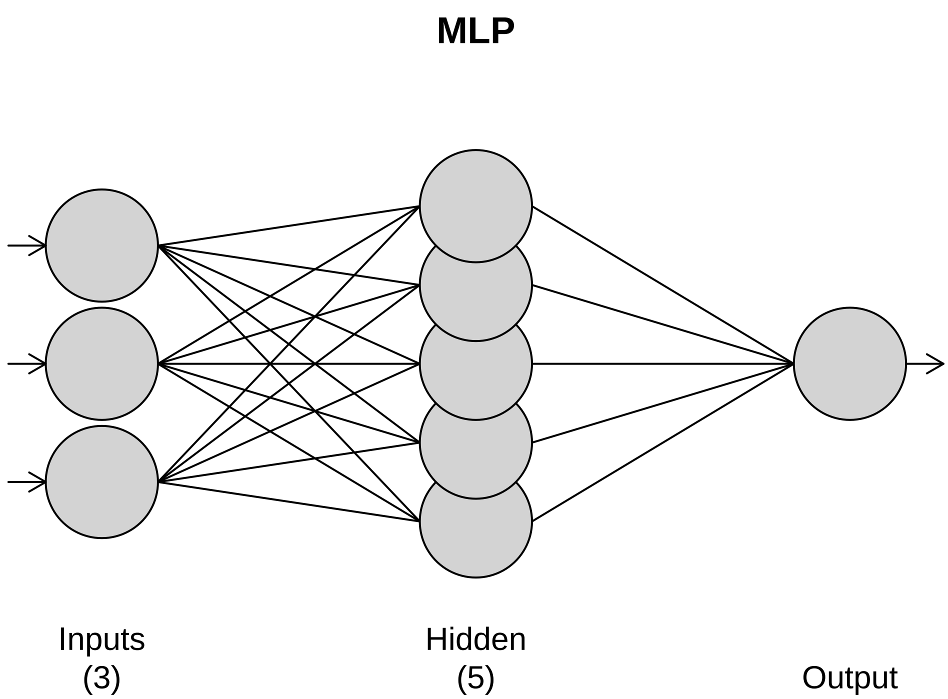


MLP/RNN

```
mlp.fit = mlp(ts(newcases_us$positiveIncrease), hd.auto.type = "cv")
mlp.fit
```

```
## MLP fit with 5 hidden nodes and 20 repetitions.
## Univariate lags: (1,3,4)
## Forecast combined using the median operator.
## MSE: 6949524.6167.
```

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




```
#best option is 20 reps and 5 hds, lags at 1,3, and 4
```

####US Cases MLP

```
trainingSize = 70
horizon = 12
ASEHolder = numeric()


for( i in 1:(177-(trainingSize + horizon) + 1))
{
  mlp.fit = mlp(ts(newcases_us$positiveIncrease[1:trainingSize+i]), hd = 5, reps = 20, lags = c(1,3,4),
  forecasts = forecast(mlp.fit,h = horizon)

  ASE = mean((newcases_us$positiveIncrease[(trainingSize+i):(trainingSize+ i + (horizon) - 1)] -forecas
  print(c(i,ASE, "from",trainingSize+i,"to",(trainingSize+ i + (horizon) - 1)))
  ASEHolder[i] = ASE

}
```

```
## [1] "1"                "148684579.1512" "from"           "71"
## [5] "to"               "82"
## [1] "2"                "214438895.7799" "from"           "72"
## [5] "to"               "83"
## [1] "3"                  "269945717.405886" "from"
## [4] "73"                 "to"               "84"
## [1] "4"                  "363428672.958542" "from"
## [4] "74"                 "to"               "85"
## [1] "5"                  "12761435.9265899" "from"
## [4] "75"                 "to"               "86"
## [1] "6"                  "13080099.6497793" "from"
## [4] "76"                 "to"               "87"
## [1] "7"                  "16037375.4254946" "from"
```

```
## [4] "77"                  "to"                 "88"
## [1] "8"                   "10973377.8165902" "from"
## [4] "78"                  "to"                 "89"
## [1] "9"                   "165952624.945672" "from"
## [4] "79"                  "to"                 "90"
## [1] "10"                  "120440087.161928" "from"
## [4] "80"                  "to"                 "91"
## [1] "11"                  "39372210.3245278" "from"
## [4] "81"                  "to"                 "92"
## [1] "12"                  "9595668.24932095" "from"
## [4] "82"                  "to"                 "93"
## [1] "13"                  "6310542.28196121" "from"
## [4] "83"                  "to"                 "94"
## [1] "14"                  "9717303.45122604" "from"
## [4] "84"                  "to"                 "95"
## [1] "15"                  "9113039.51131458" "from"
## [4] "85"                  "to"                 "96"
## [1] "16"                  "14711395.0557575" "from"
## [4] "86"                  "to"                 "97"
## [1] "17"                 "18110955.573725" "from"             "87"
## [5] "to"                 "98"
## [1] "18"                  "15293432.7909139" "from"
## [4] "88"                  "to"                 "99"
## [1] "19"                  "14457644.3427429" "from"
## [4] "89"                  "to"                 "100"
## [1] "20"                  "14761840.7838651" "from"
## [4] "90"                  "to"                 "101"
## [1] "21"                  "14839267.7799039" "from"
## [4] "91"                  "to"                 "102"
## [1] "22"                  "15166270.7436933" "from"
## [4] "92"                  "to"                 "103"
## [1] "23"                  "18893015.6708325" "from"
## [4] "93"                  "to"                 "104"
## [1] "24"                  "24400552.1823697" "from"
## [4] "94"                  "to"                 "105"
## [1] "25"                  "31952114.0873817" "from"
## [4] "95"                  "to"                 "106"
## [1] "26"                  "20050858.1704846" "from"
## [4] "96"                  "to"                 "107"
## [1] "27"                  "12992203.2378673" "from"
## [4] "97"                  "to"                 "108"
## [1] "28"                  "14792072.6331184" "from"
## [4] "98"                  "to"                 "109"
## [1] "29"                  "19718664.6327716" "from"
## [4] "99"                  "to"                 "110"
## [1] "30"                  "30736958.7317149" "from"
## [4] "100"                 "to"                 "111"
## [1] "31"                  "38557763.6401645" "from"
## [4] "101"                 "to"                 "112"
## [1] "32"                  "40500347.2477024" "from"
## [4] "102"                 "to"                 "113"
## [1] "33"                 "34095531.558537" "from"             "103"
## [5] "to"                 "114"
## [1] "34"                  "26875496.6965107" "from"
```

```
## [4] "104"               "to"               "115"
## [1] "35"                "26222515.8657507" "from"
## [4] "105"               "to"               "116"
## [1] "36"                "36214379.0689974" "from"
## [4] "106"               "to"               "117"
## [1] "37"               "38819534.290802" "from"             "107"
## [5] "to"               "118"
## [1] "38"                "47535326.9485762" "from"
## [4] "108"               "to"               "119"
## [1] "39"                "44807002.9845167" "from"
## [4] "109"               "to"               "120"
## [1] "40"                "34116417.8525435" "from"
## [4] "110"               "to"               "121"
## [1] "41"                "24183505.3225474" "from"
## [4] "111"               "to"               "122"
## [1] "42"              "30455909.02851" "from"            "112"
## [5] "to"              "123"
## [1] "43"                "27832231.7369717" "from"
## [4] "113"               "to"               "124"
## [1] "44"                "39801622.0175296" "from"
## [4] "114"               "to"               "125"
## [1] "45"                "45996131.5879191" "from"
## [4] "115"               "to"               "126"
## [1] "46"                "48340571.0152524" "from"
## [4] "116"               "to"               "127"
## [1] "47"                "40148432.4063388" "from"
## [4] "117"               "to"               "128"
## [1] "48"                "37113316.4274717" "from"
## [4] "118"               "to"               "129"
## [1] "49"                "35937836.4500677" "from"
## [4] "119"               "to"               "130"
## [1] "50"                "32310994.2258286" "from"
## [4] "120"               "to"               "131"
## [1] "51"                "43545157.6006801" "from"
## [4] "121"               "to"               "132"
## [1] "52"                "40139421.4942934" "from"
## [4] "122"               "to"               "133"
## [1] "53"                "31556547.3405823" "from"
## [4] "123"               "to"               "134"
## [1] "54"                "33250683.6845621" "from"
## [4] "124"               "to"               "135"
## [1] "55"                "23604472.8947035" "from"
## [4] "125"               "to"               "136"
## [1] "56"               "5180172.8222758" "from"             "126"
## [5] "to"               "137"
## [1] "57"                "3690984.54296625" "from"
## [4] "127"               "to"               "138"
## [1] "58"               "11742970.811547" "from"             "128"
## [5] "to"               "139"
## [1] "59"                "31555140.0087619" "from"
## [4] "129"               "to"               "140"
## [1] "60"                "8894954.66802387" "from"
## [4] "130"               "to"               "141"
## [1] "61"                "6243084.45275458" "from"
```

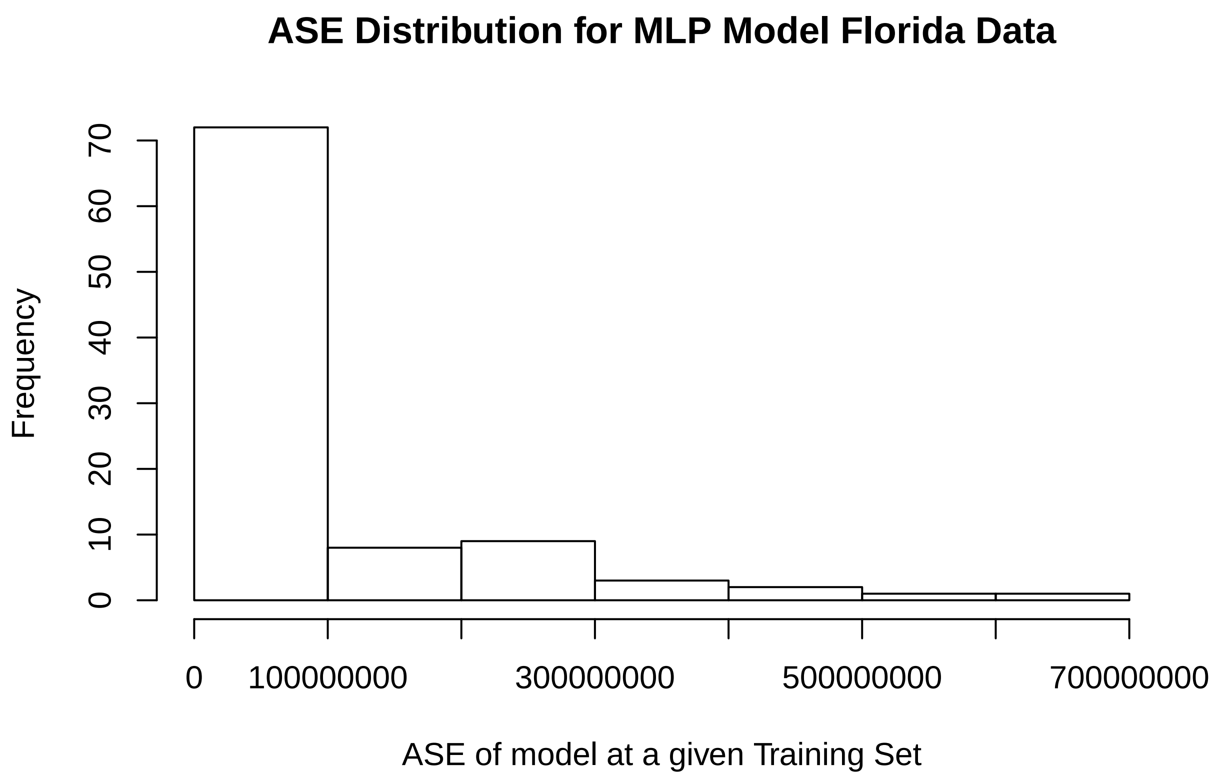
```
## [4] "131"               "to"               "142"
## [1] "62"                "4880754.84722899" "from"
## [4] "132"               "to"               "143"
## [1] "63"                "5970237.49930091" "from"
## [4] "133"               "to"               "144"
## [1] "64"                "4462873.86177125" "from"
## [4] "134"               "to"               "145"
## [1] "65"                "8529068.38446921" "from"
## [4] "135"               "to"               "146"
## [1] "66"                "6869559.14076634" "from"
## [4] "136"               "to"               "147"
## [1] "67"                "7312023.09646942" "from"
## [4] "137"               "to"               "148"
## [1] "68"                "8554380.06354165" "from"
## [4] "138"               "to"               "149"
## [1] "69"                "13369883.1542844" "from"
## [4] "139"               "to"               "150"
## [1] "70"                "18149869.2262759" "from"
## [4] "140"               "to"               "151"
## [1] "71"                "20869655.1971004" "from"
## [4] "141"               "to"               "152"
## [1] "72"                "24481295.4377758" "from"
## [4] "142"               "to"               "153"
## [1] "73"                "36486680.3359107" "from"
## [4] "143"               "to"               "154"
## [1] "74"                "64017646.4555764" "from"
## [4] "144"               "to"               "155"
## [1] "75"                "87562878.5253474" "from"
## [4] "145"               "to"               "156"
## [1] "76"               "127538416.01977" "from"             "146"
## [5] "to"               "157"
## [1] "77"                "173919643.291979" "from"
## [4] "147"               "to"               "158"
## [1] "78"                "213445851.434686" "from"
## [4] "148"               "to"               "159"
## [1] "79"                "229802100.601849" "from"
## [4] "149"               "to"               "160"
## [1] "80"                "110606493.410821" "from"
## [4] "150"               "to"               "161"
## [1] "81"                "226449504.694279" "from"
## [4] "151"               "to"               "162"
## [1] "82"                "388152328.790382" "from"
## [4] "152"               "to"               "163"
## [1] "83"                "298234763.228073" "from"
## [4] "153"               "to"               "164"
## [1] "84"                "469149470.640469" "from"
## [4] "154"               "to"               "165"
## [1] "85"                "423586020.234415" "from"
## [4] "155"               "to"               "166"
## [1] "86"                "532200061.242697" "from"
## [4] "156"               "to"               "167"
## [1] "87"                "209861712.809406" "from"
## [4] "157"               "to"               "168"
## [1] "88"                "257580171.331275" "from"
```

```
## [4] "158"               "to"                "169"
## [1] "89"                "346010708.616653" "from"
## [4] "159"               "to"                "170"
## [1] "90"                "678257846.106531" "from"
## [4] "160"               "to"                "171"
## [1] "91"                "172519192.667737" "from"
## [4] "161"               "to"                "172"
## [1] "92"                "73121693.4929402" "from"
## [4] "162"               "to"                "173"
## [1] "93"                "119717957.490213" "from"
## [4] "163"               "to"                "174"
## [1] "94"                "45662604.1699962" "from"
## [4] "164"               "to"                "175"
## [1] "95"                "72934748.2375532" "from"
## [4] "165"               "to"                "176"
## [1] "96"                "216177428.295245" "from"
## [4] "166"               "to"                "177"
```

```
ASEHolder
```

```
##  [1] 148684579 214438896 269945717 363428673  12761436  13080100  16037375
##  [8]  10973378 165952625 120440087  39372210   9595668   6310542   9717303
## [15]   9113040  14711395  18110956  15293433  14457644  14761841  14839268
## [22]  15166271  18893016  24400552  31952114  20050858  12992203  14792073
## [29]  19718665  30736959  38557764  40500347  34095532  26875497  26222516
## [36]  36214379  38819534  47535327  44807003  34116418  24183505  30455909
## [43]  27832232  39801622  45996132  48340571  40148432  37113316  35937836
## [50]  32310994  43545158  40139421  31556547  33250684  23604473   5180173
## [57]   3690985  11742971  31555140   8894955   6243084   4880755   5970237
## [64]   4462874   8529068   6869559   7312023   8554380  13369883  18149869
## [71]  20869655  24481295  36486680  64017646  87562879 127538416 173919643
## [78] 213445851 229802101 110606493 226449505 388152329 298234763 469149471
## [85] 423586020 532200061 209861713 257580171 346010709 678257846 172519193
## [92]  73121693 119717957  45662604  72934748 216177428
```

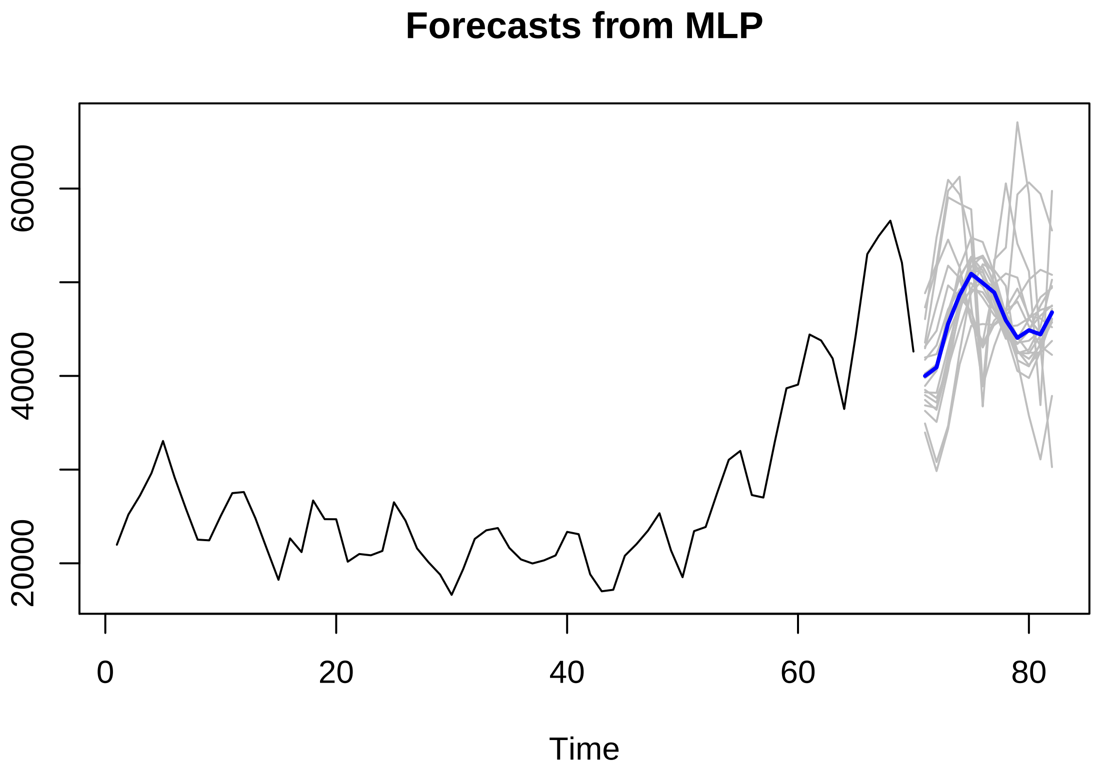
```
#Distribution of ASEs on Two Week Periods
hist(ASEHolder, xlab = "ASE of model at a given Training Set",  main = "ASE Distribution for MLP Model 
```




```r
#Mean ASE
WindowedASE = mean(ASEHolder)
WindowedASE
```

```
## [1] 87046280
```

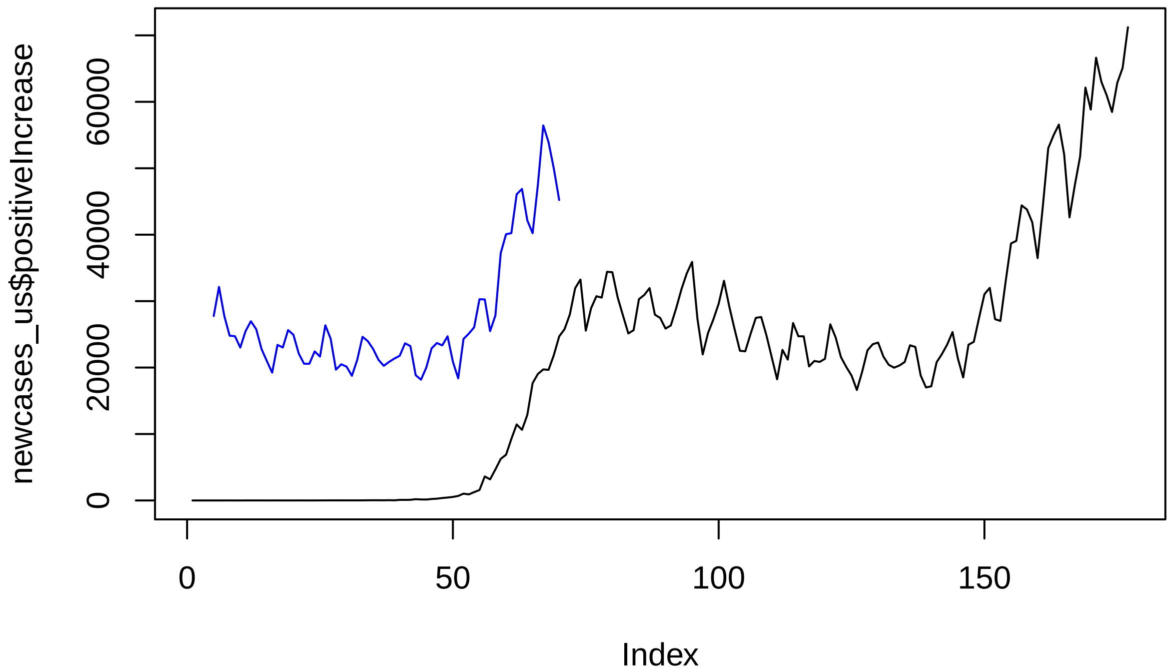
```r
#228 mill

plot(forecasts)
```



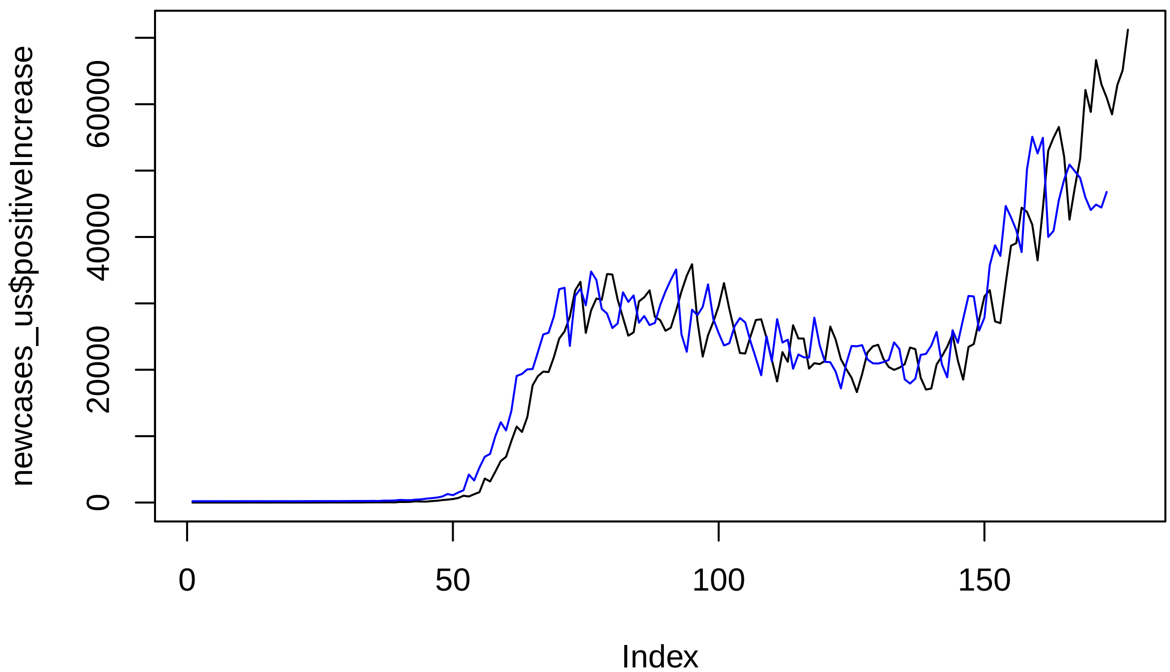

```
#Actual Forecasting on last segment of data

mlp.fit = mlp(ts(newcases_us$positiveIncrease[1:165]), hd = 5, comb = "median")
forecasts_us_mlp = forecast(mlp.fit,h = 12)

ASE = mean((newcases_us$positiveIncrease[166:177] -forecasts$mean)^2)
ASE
```

```
## [1] 216177428
```

```
#53,843,551
plot(newcases_us$positiveIncrease, type = "l")
lines(forecasts$fitted, col = "blue")
```

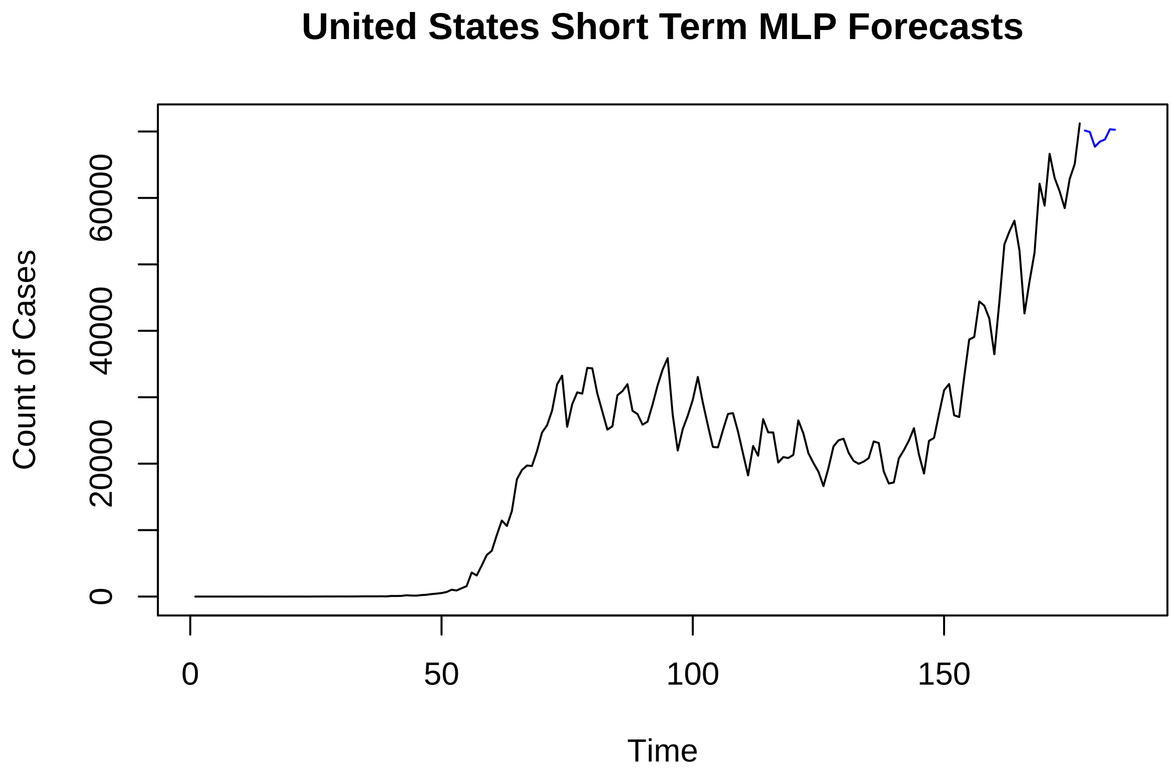



```r
all_f = c(forecasts_us_mlp$fitted, forecasts$mean)
plot(newcases_us$positiveIncrease, type = "l")
lines(all_f, col = "blue")
```

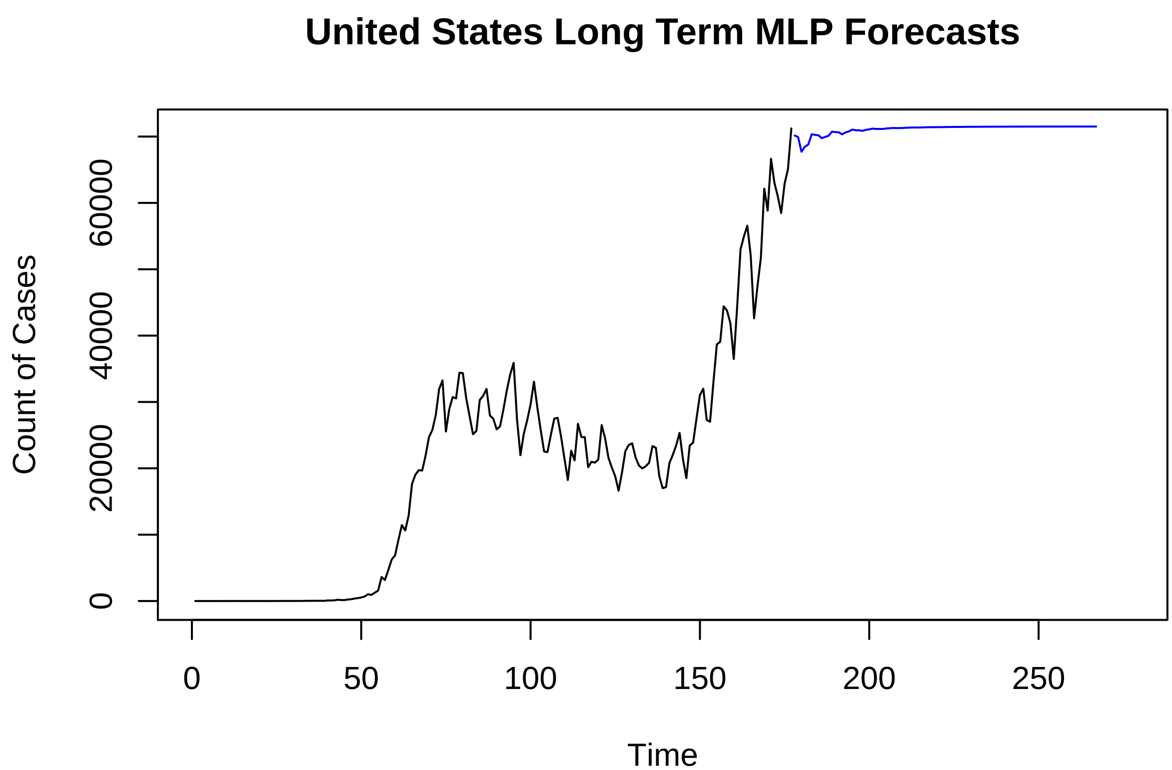



```r
#Future Predictions

mlp.fit_us_future =mlp(ts(newcases_us$positiveIncrease), hd = 5, comb = "median")
short_us_mlp = forecast(mlp.fit_us_future,h = 7)

long_us_mlp = forecast(mlp.fit_us_future,h = 90)
```

```
plot(newcases_us$positiveIncrease, type = "l", xlim = c(1,187), main = "United States Short Term MLP Fo
lines(short_us_mlp$mean, col = "blue")
```

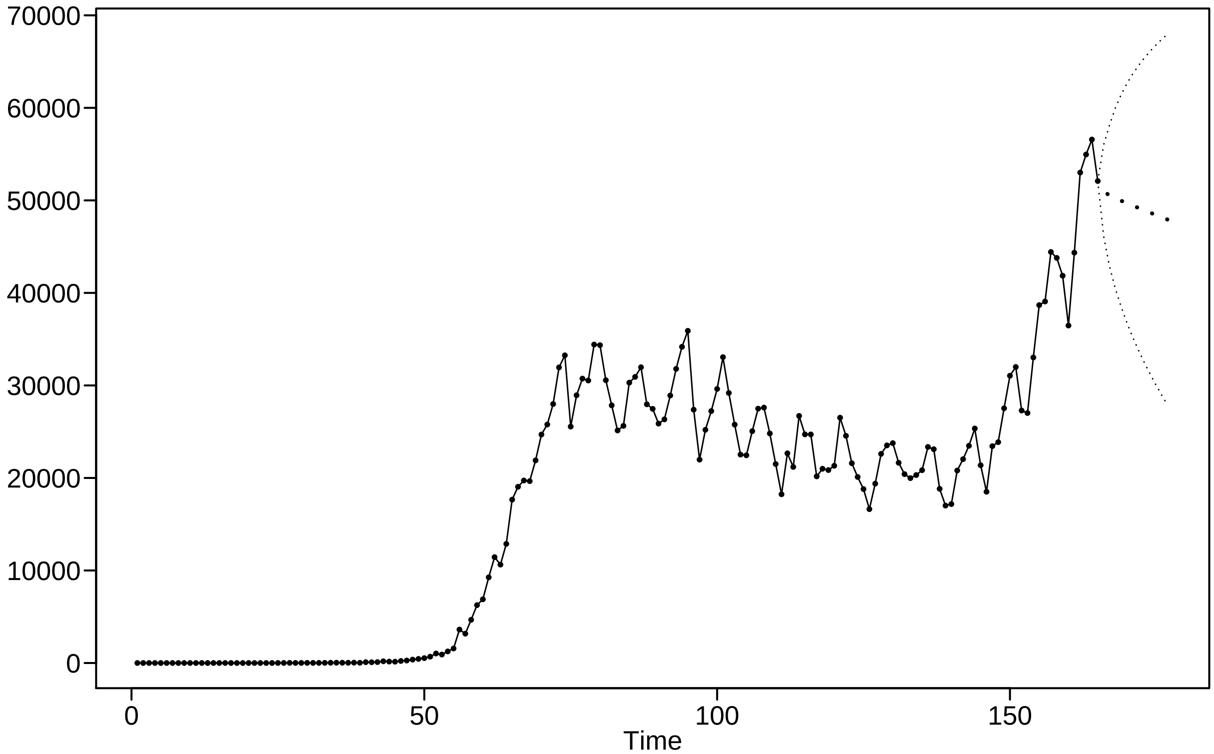



```
plot(newcases_us$positiveIncrease, type = "l", xlim = c(1,277), main = "United States Long Term MLP Fore
lines(long_us_mlp$mean, col = "blue")
```



####US Ensemble

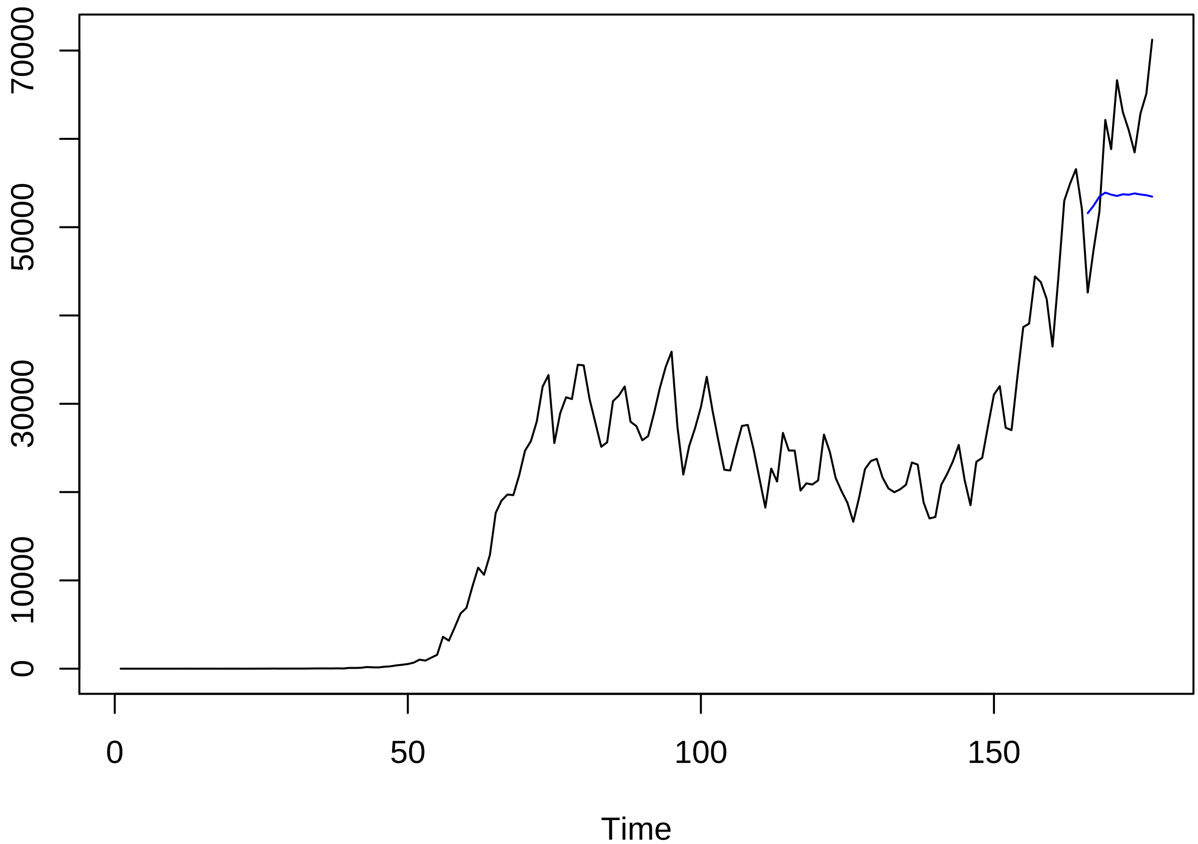
```r
#ASE fits for ensemble
#mlp.fit_us_final = mlp(ts(newcases_us$positiveIncrease[1:165]), hd = 5, comb = "median")
#forecasts_us_mlp = forecast(mlp.fit_us_final,h = 12)


forecasts_arma_us = fore.aruma.wge(newcases_us$positiveIncrease[1:165],phi = us_arma_12$phi,theta = us_a
```



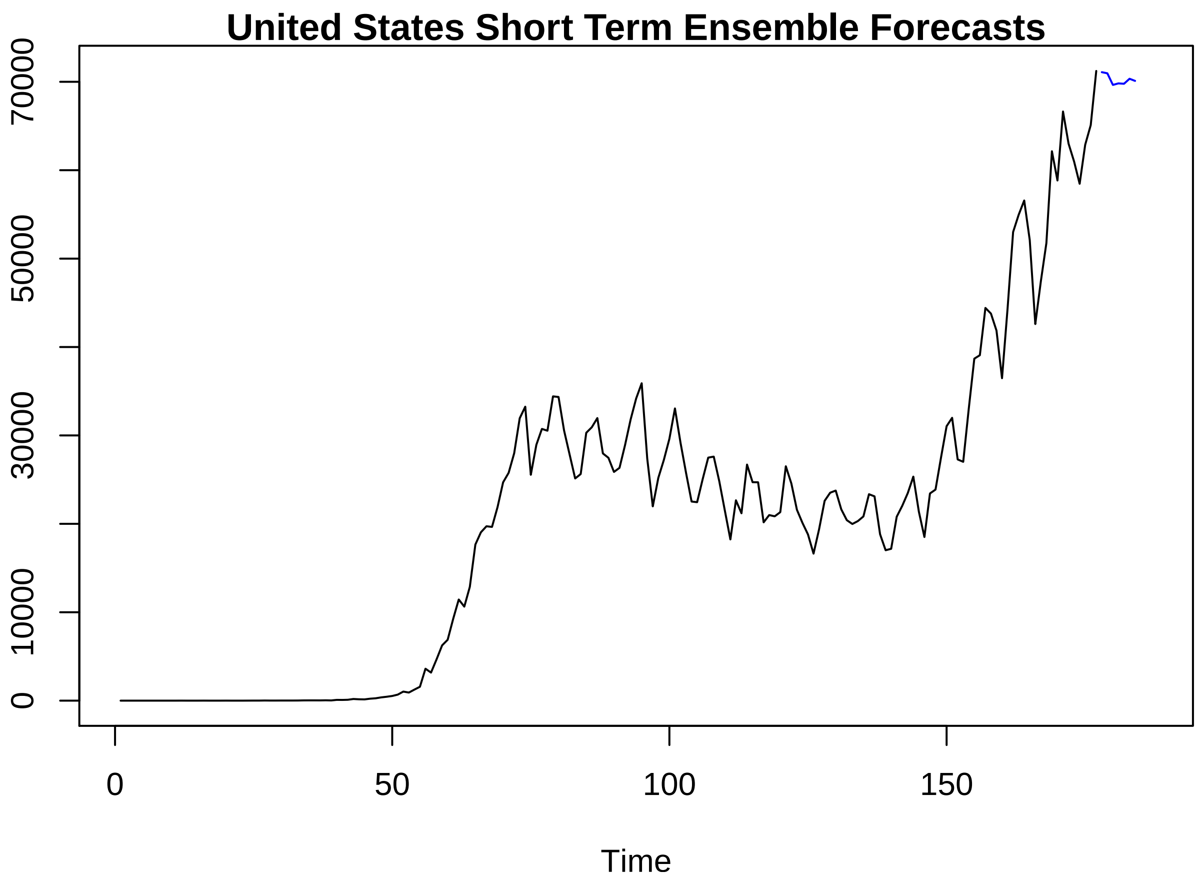

```r
ensemble_fore = (forecasts_us_mlp$mean + forecasts_arma_us$f) / 2

ensemble_ASE = mean((newcases_us$positiveIncrease[166:177] -ensemble_fore)^2)
ensemble_ASE
```

```
## [1] 88969399
```

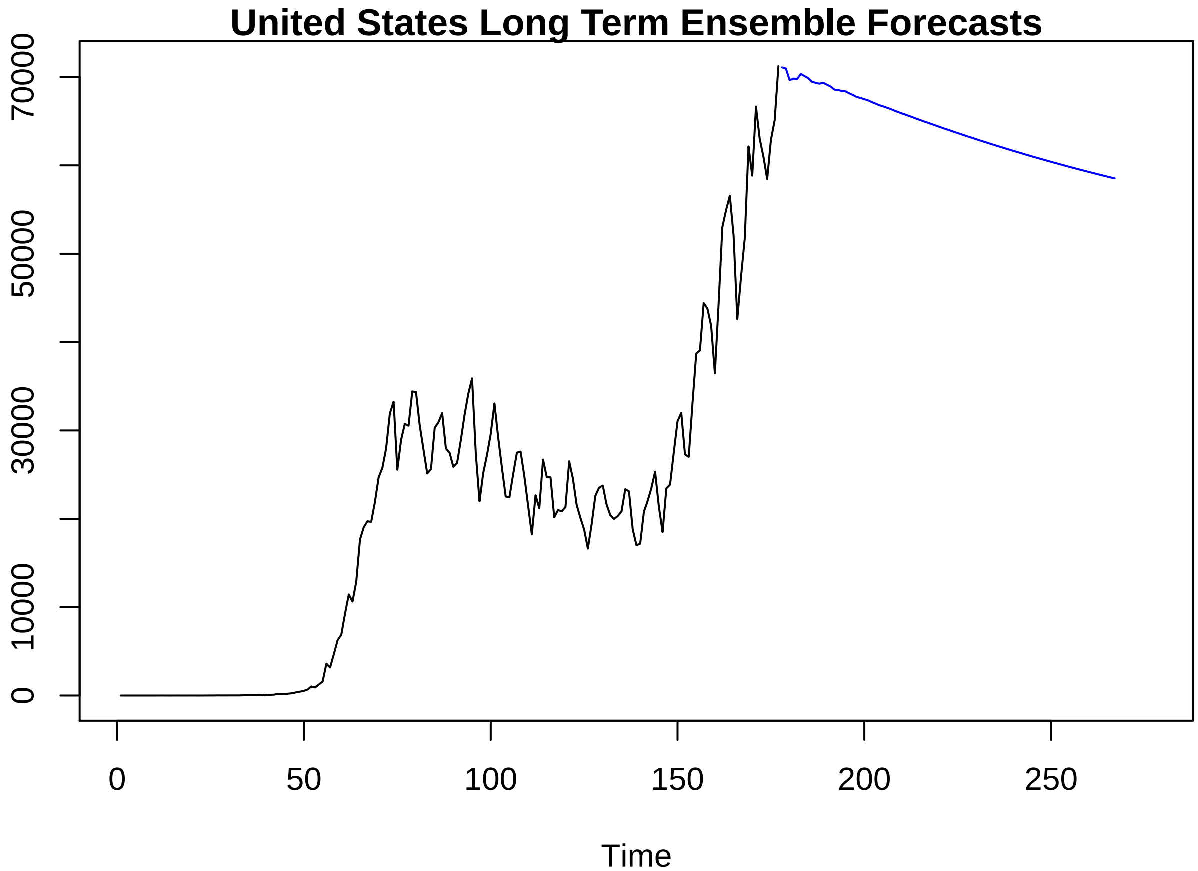
```r
#114 mill

plot(newcases_us$positiveIncrease, type = "l", ylab = "Count of New Cases", xlab = "Time")
lines(ensemble_fore, col = "blue")
```




```
#Forecasting ahead
short_ensemble_us = (short_us_mlp$mean +short_us_ar1$f)/2

long_ensemble_us = (long_us_mlp$mean +long_us_ar1$f)/2


plot(newcases_us$positiveIncrease, type = "l", xlim = c(1,187), main = "United States Short Term Ensembl
lines(short_ensemble_us, col = "blue")
```



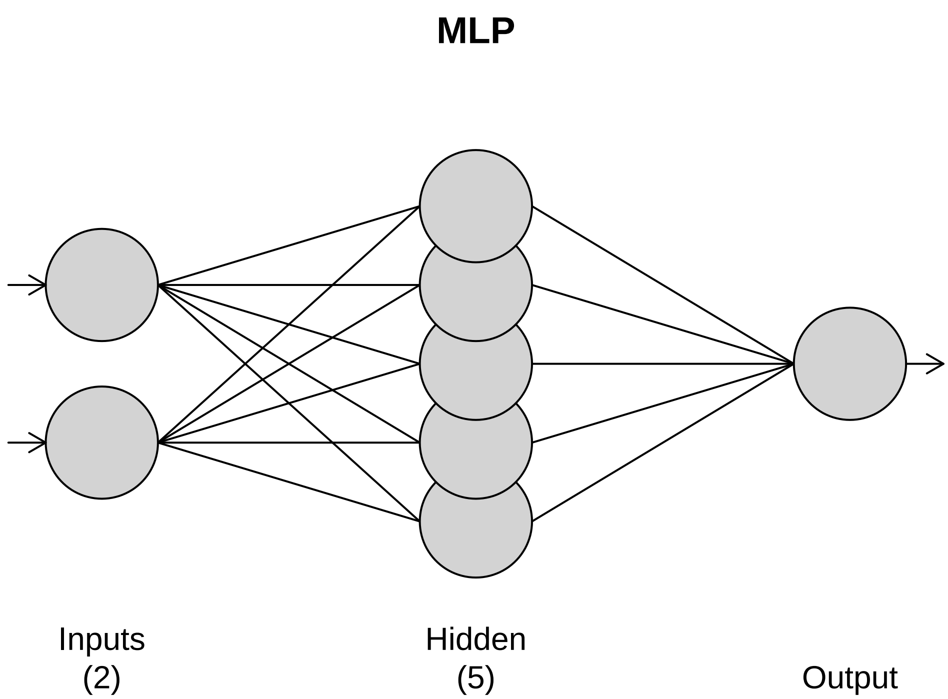

```r
plot(newcases_us$positiveIncrease, type = "l", xlim = c(1,277), main = "United States Long Term Ensembl
lines(long_ensemble_us, col = "blue")
```



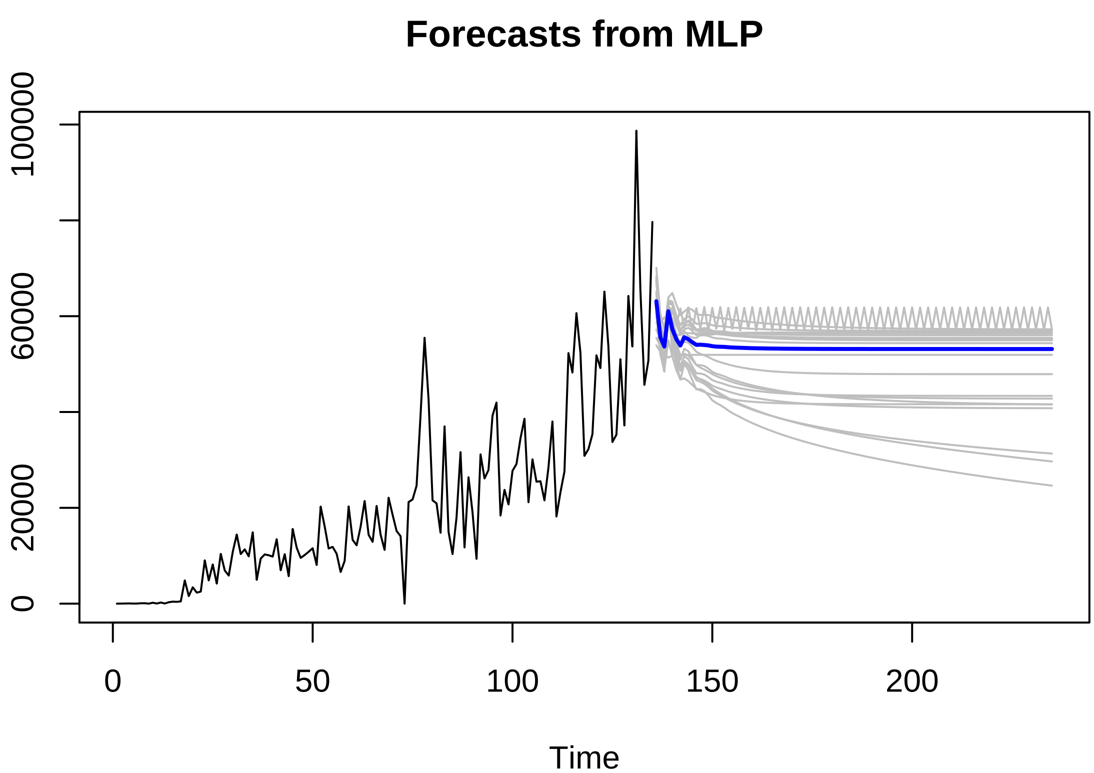
Model Building

##Goal Three: Multivariate Analysis

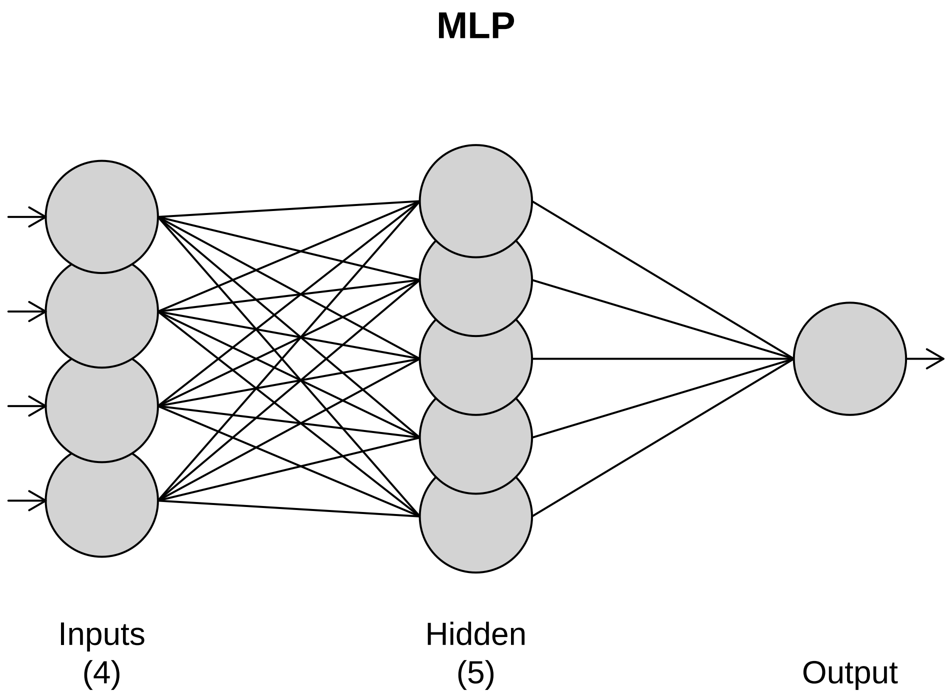
```
newcases_fl_multi =  initial_data_fl %>% dplyr::select(positiveIncrease, totalTestResultsIncrease, hospi

#Forecast beyond data for Florida

#Forecast future variables
fit.mlp.1 = mlp(ts(newcases_fl_multi$totalTestResultsIncrease),reps = 20, comb = "median")
plot(fit.mlp.1)
```



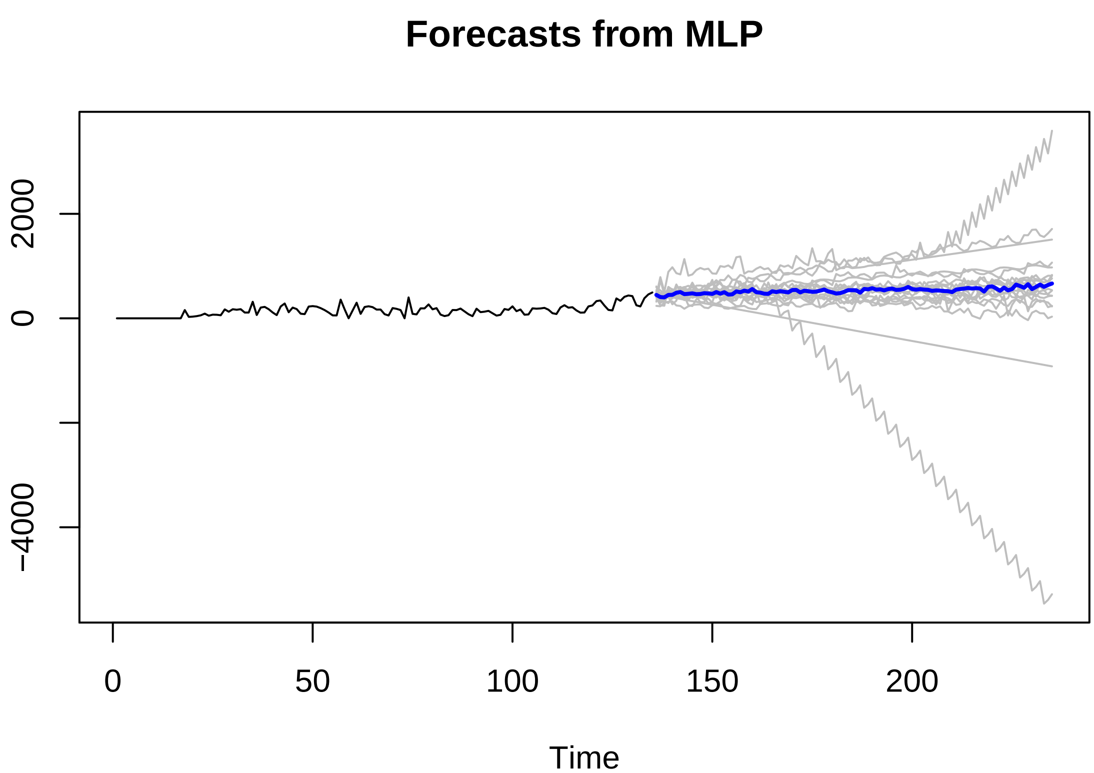

```r
fore.mlp.1 = forecast(fit.mlp.1, h = 100)
plot(fore.mlp.1)
```



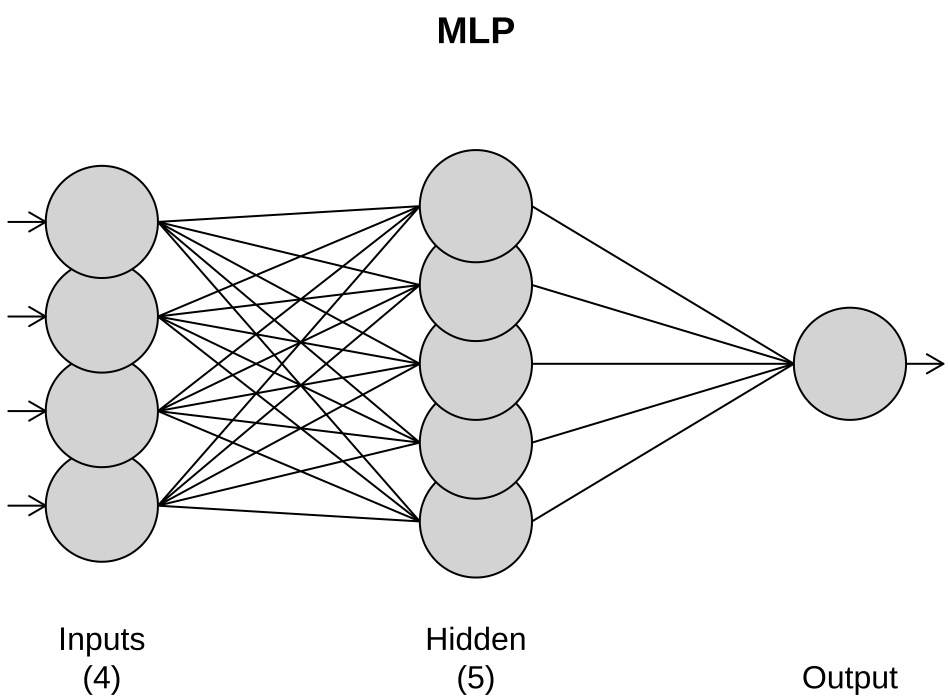

```r
fit.mlp.2 = mlp(ts(newcases_fl_multi$hospitalizedIncrease),reps = 20, comb = "median")
plot(fit.mlp.2)
```



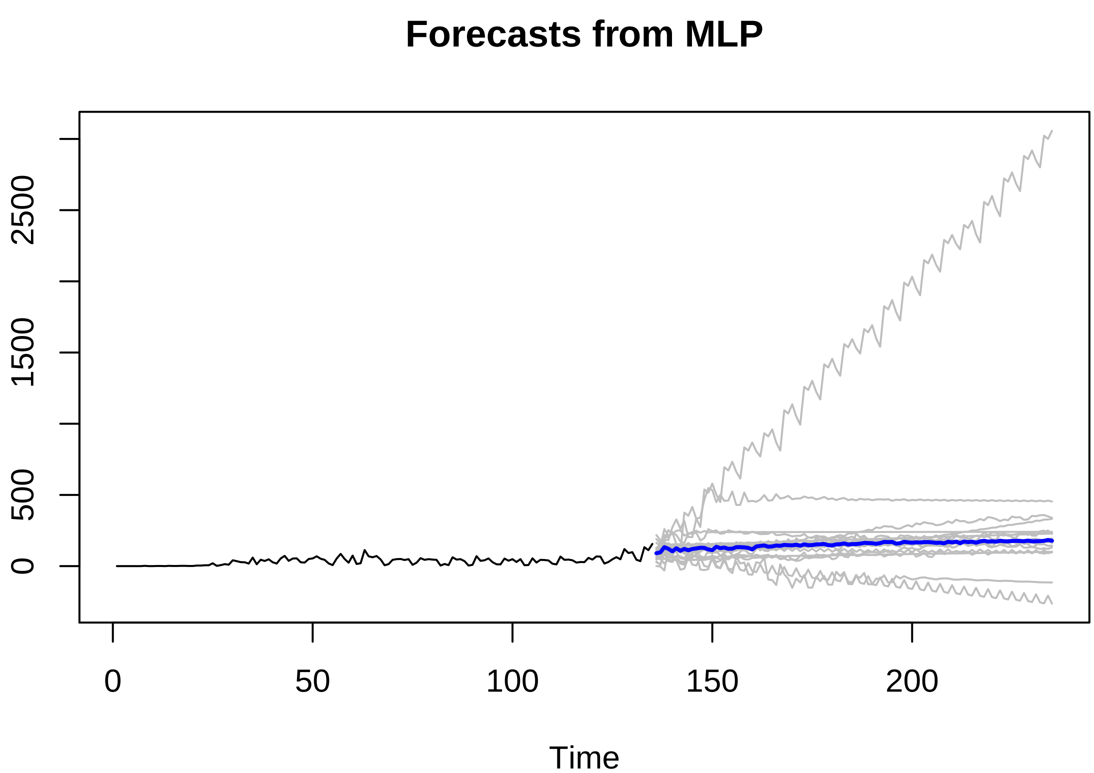

```r
fore.mlp.2 = forecast(fit.mlp.2, h = 100)
plot(fore.mlp.2)
```




```r
fit.mlp.3 = mlp(ts(newcases_fl_multi$deathIncrease),reps = 20, comb = "median")
plot(fit.mlp.3)
```




```
fore.mlp.3 = forecast(fit.mlp.3, h = 100)
plot(fore.mlp.3)
```




```
#package them up in data frame.
newvar_fore_fl = data.frame(totalTestResultsIncrease = ts(c(newcases_fl_multi$totalTestResultsIncrease,f

#Data has 100 instances beyond current data
dim(newvar_fore_fl)
```

```
## [1] 235   3
```

###Multivariate Model Building for Florida Cases

####Florida MLR Model

```r
fit = lm(positiveIncrease~totalTestResultsIncrease + hospitalizedIncrease, data = newcases_fl_multi)
summary(fit)
```

```
##
## Call:
## lm(formula = positiveIncrease ~ totalTestResultsIncrease + hospitalizedIncrease,
##     data = newcases_fl_multi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6794.8 -1047.8     1.2  1357.6  3712.4
##
## Coefficients:
##                             Estimate   Std. Error t value
## (Intercept)              -1358.579010   250.514502  -5.423
## totalTestResultsIncrease     0.137289     0.009719  14.126
## hospitalizedIncrease         5.588560     1.626045   3.437
##                                         Pr(>|t|)
## (Intercept)                           0.00000027 ***
## totalTestResultsIncrease < 0.0000000000000002 ***
## hospitalizedIncrease                    0.000787 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1687 on 132 degrees of freedom
## Multiple R-squared:  0.7585, Adjusted R-squared:  0.7548
## F-statistic: 207.3 on 2 and 132 DF,  p-value: < 0.00000000000000022
```

```r
est_tests = mean(tail(newcases_fl_multi$totalTestResultsIncrease))
est_hospital= mean(tail(newcases_fl_multi$hospitalizedIncrease))
newdata = data.frame(totalTestResultsIncrease = rep(est_tests,12), hospitalizedIncrease = rep(est_hospi

preds = predict(fit, newdata = newdata)



aic5.wge(fit$residuals)#picks 1,1
```

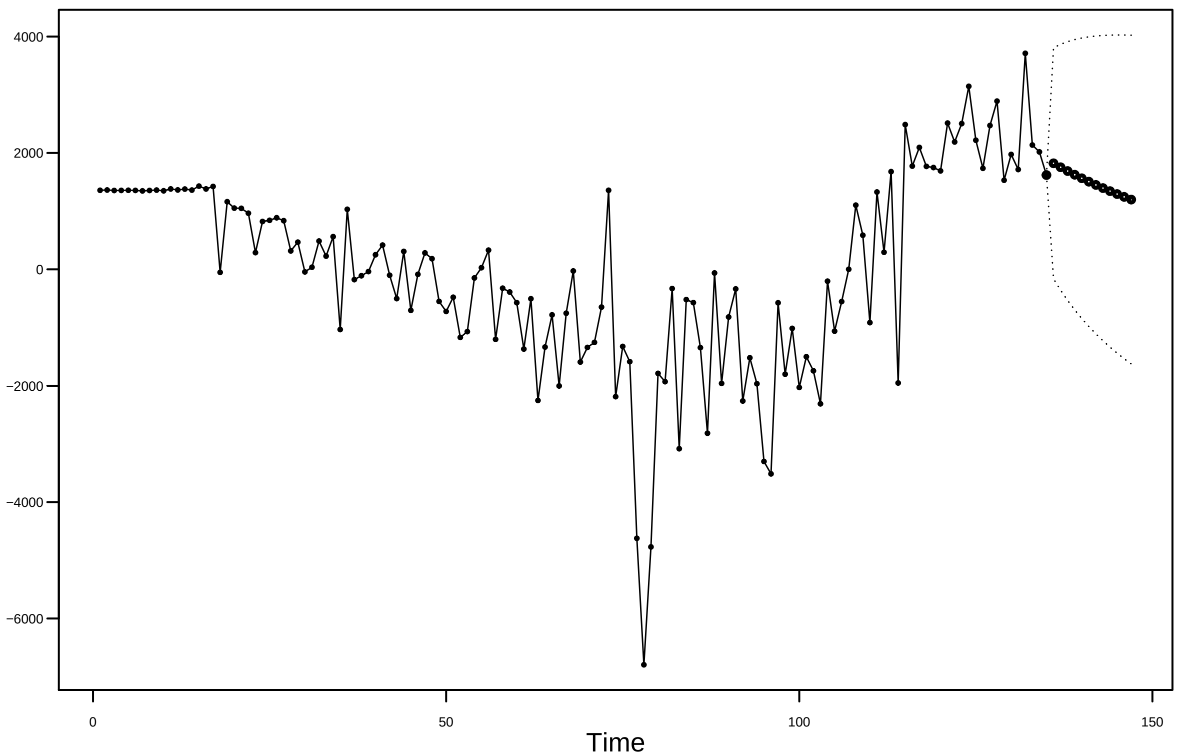
```
## ---------WORKING... PLEASE WAIT...
##
##
## Five Smallest Values of  aic
```

```
##       p    q        aic
## 5     1    1   13.87980
## 8     2    1   13.89429
## 6     1    2   13.89497
## 11    3    1   13.90310
## 9     2    2   13.90630
```

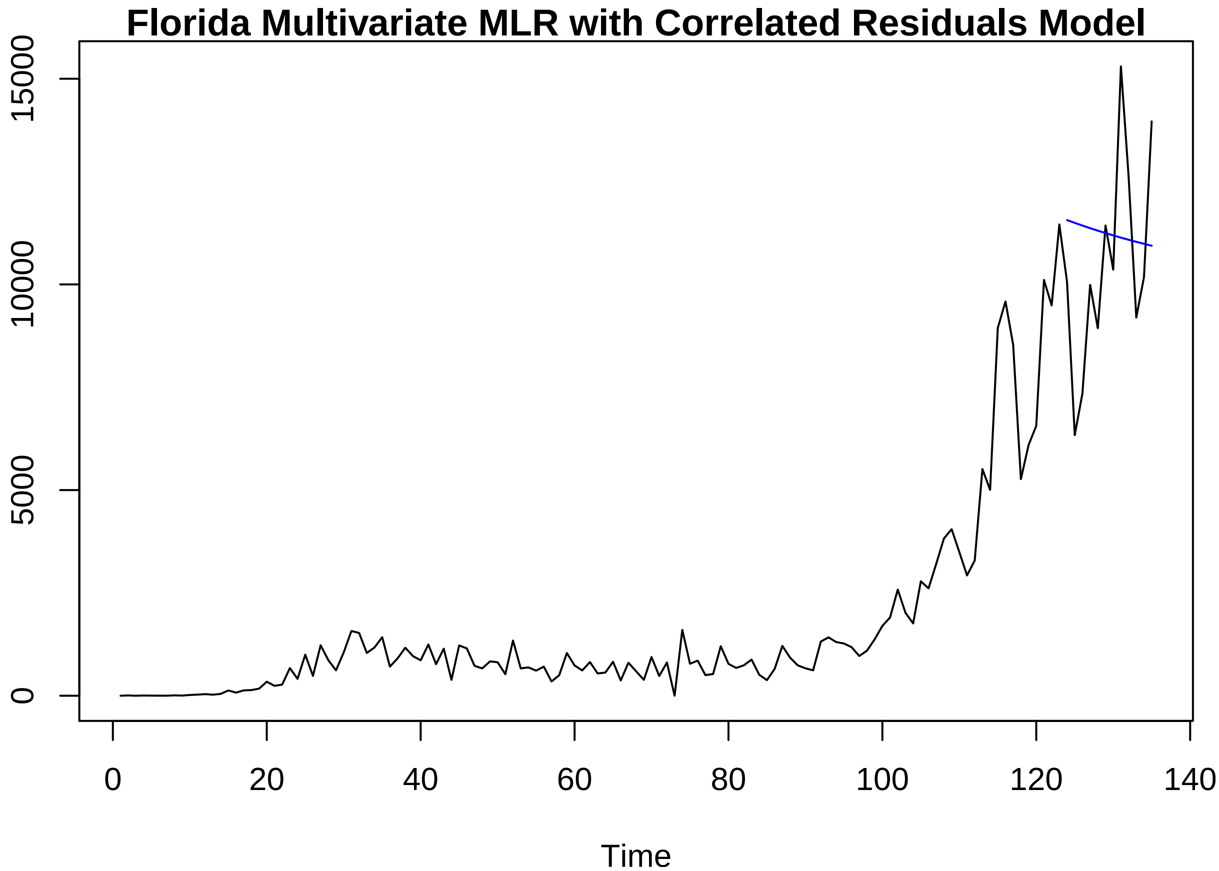
```r
est1 = est.arma.wge(fit$residuals, p = 1, q = 1)
```

```
##
## Coefficients of Original polynomial:
## 0.9626
##
## Factor                 Roots                Abs Recip    System Freq
## 1-0.9626B              1.0388               0.9626       0.0000
##
##
```

```r
forecasts = fore.arma.wge(fit$residuals,phi = est1$phi,theta = est1$theta, lastn = FALSE,n.ahead = 12)
```



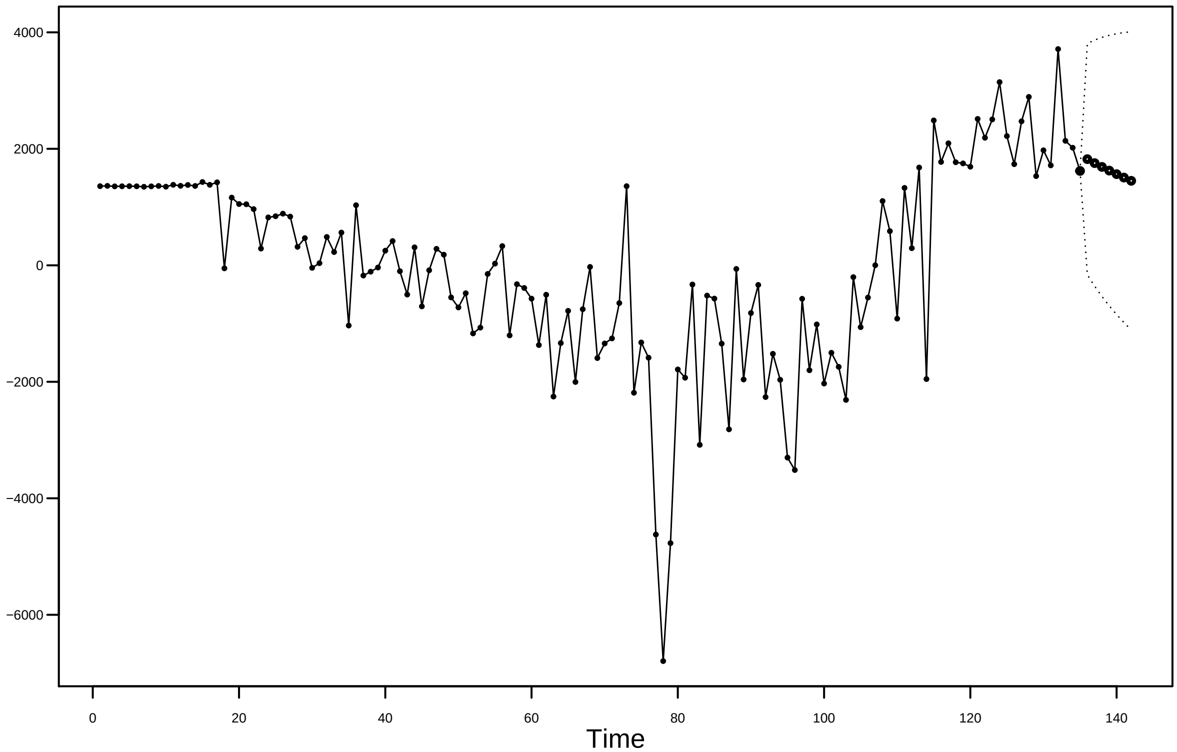

```r
FinalPredictions_fl_MLR  = preds + forecasts$f

plot(newcases_fl$positiveIncrease, type = "l", main ="Florida Multivariate MLR with Correlated Residuals
lines(ts(FinalPredictions_fl_MLR, start = 124), col = "blue")
```



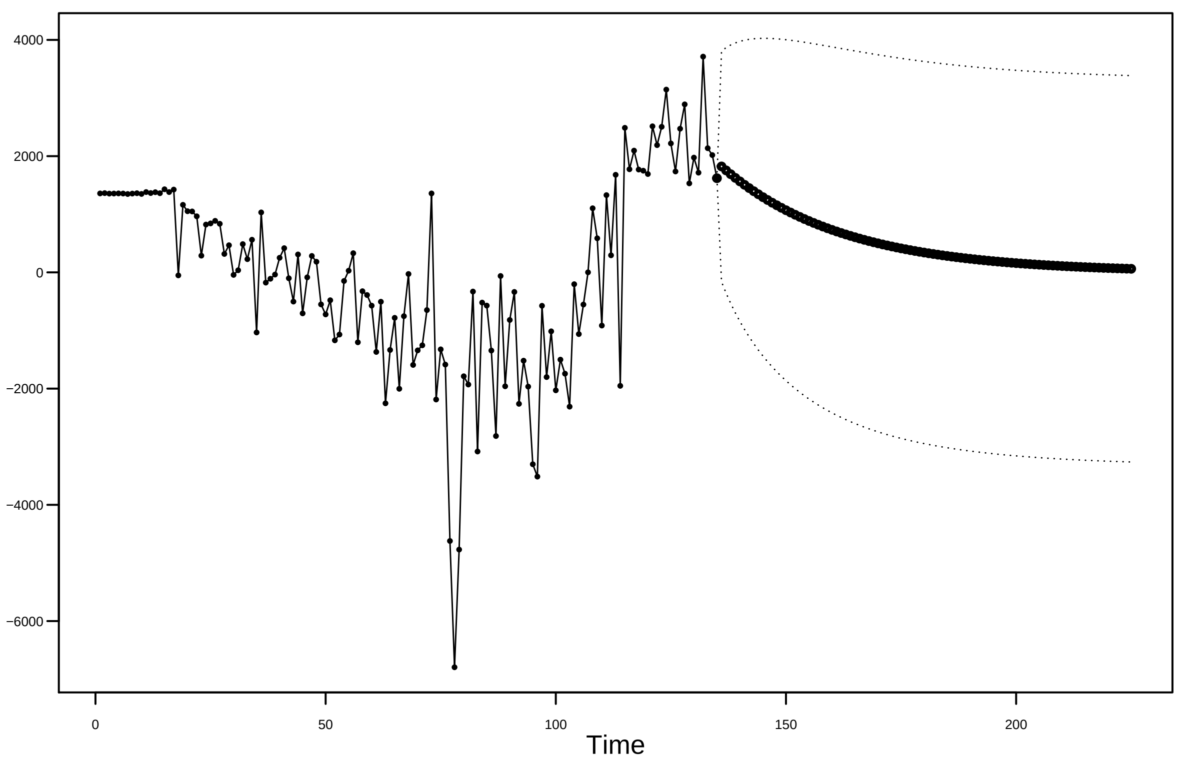

```
ASE = mean((newcases_fl_multi$positiveIncrease[124:135] - FinalPredictions_fl_MLR)^2)
ASE
```

```
## [1] 7223923
```

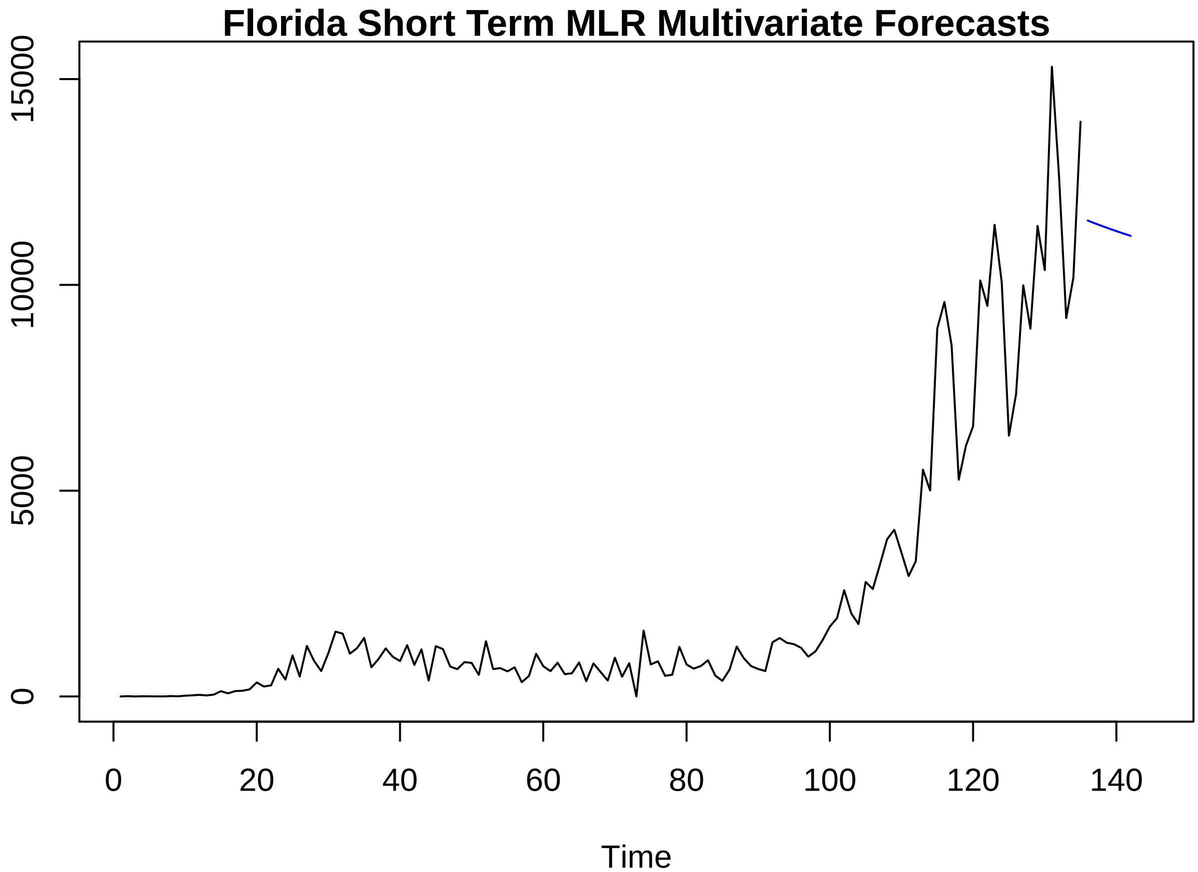
```
#7223923



#Forecasting Ahead

shortdata = data.frame(totalTestResultsIncrease = rep(est_tests,7), hospitalizedIncrease = rep(est_hosp
longdata =data.frame(totalTestResultsIncrease = rep(est_tests,90), hospitalizedIncrease = rep(est_hospi


fit = lm(positiveIncrease~totalTestResultsIncrease + hospitalizedIncrease, data = newcases_fl_multi)
#short
preds = predict(fit, newdata = shortdata)
forecasts = fore.arma.wge(fit$residuals,phi = est1$phi,theta = est1$theta, lastn = FALSE,n.ahead = 7)
```



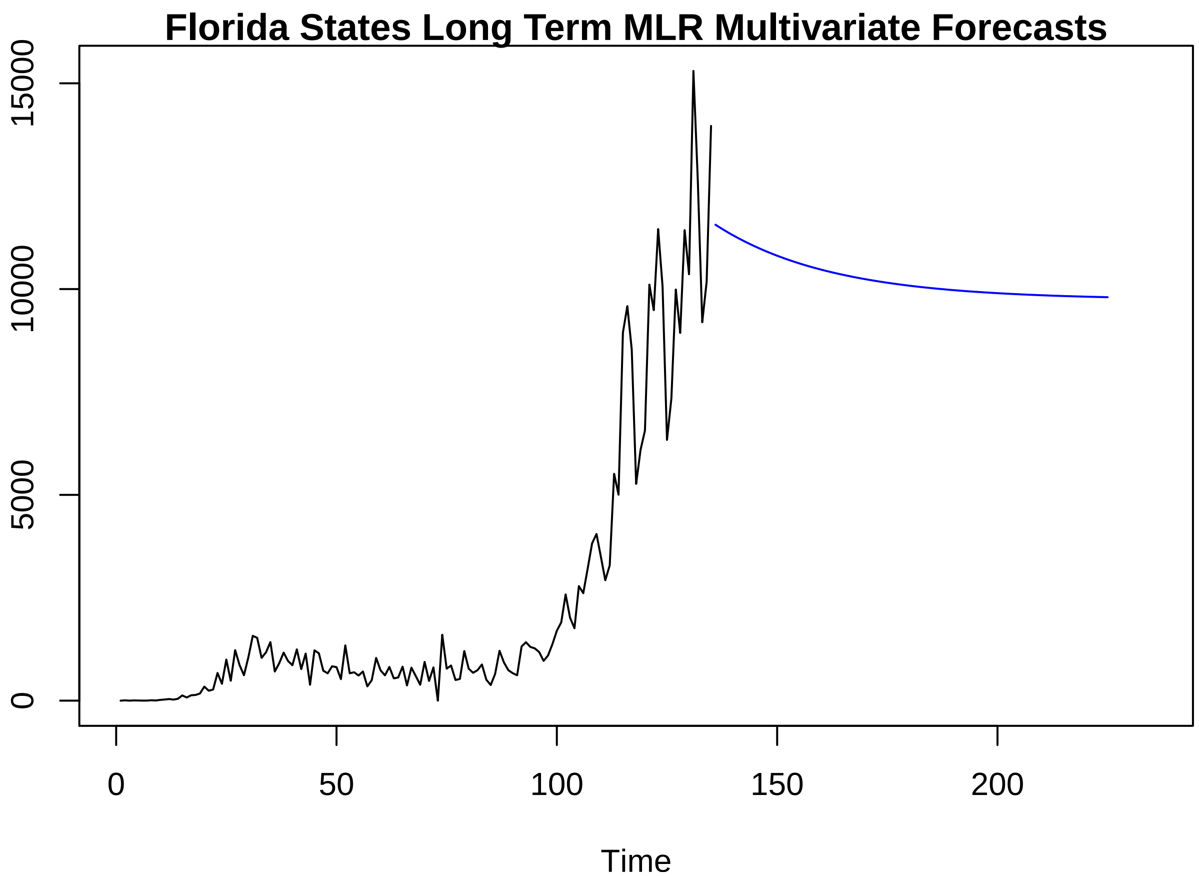

```r
short_fl_mlr_m  = preds + forecasts$f

#long
preds = predict(fit, newdata = longdata)
forecasts = fore.arma.wge(fit$residuals,phi = est1$phi,theta = est1$theta, lastn = FALSE,n.ahead = 90)
```



```
long_fl_mlr_m  = preds + forecasts$f


plot(newcases_fl$positiveIncrease, type = "l", xlim = c(1,145), main = "Florida Short Term MLR Multivari
lines(ts(short_fl_mlr_m, start = 136), col = "blue")
```




```
plot(newcases_fl$positiveIncrease, type = "l", xlim = c(1,235), main = "Florida States Long Term MLR Mul
lines(ts(long_fl_mlr_m,start = 136), col = "blue")
```



####Florida Multivariate MLP Cases Model

```
newcases_fl_multi =  initial_data_fl %>% dplyr::select(positiveIncrease, totalTestResultsIncrease, hospi
newcases_fl_var = cbind(ts(newcases_fl_multi$totalTestResultsIncrease),ts(newcases_fl_multi$hospitalize

trainingSize = 70
horizon = 12
ASEHolder = numeric()

#Out of bounds if it goes for 54 runs, this ASE will be slightly less wide than the others. But the win
for( i in 1:(135-(trainingSize + horizon) ))
{
  mlp.fit = mlp(ts(newcases_fl_multi$positiveIncrease[1:trainingSize+i]), hd = 5, comb = "median", xreg
  forecasts = forecast(mlp.fit,h = horizon, xreg = newcases_fl_var[1:(trainingSize + i + 12),])

  ASE = mean((newcases_fl_multi$positiveIncrease[(trainingSize+i):(trainingSize+ i + (horizon) - 1)] -fo
  print(c(i,ASE, "from",trainingSize+i,"to",(trainingSize+ i + (horizon) - 1)))
  ASEHolder[i] = ASE

}
```

```
## [1] "1"                "310510.472249368" "from"
## [4] "71"               "to"               "82"
## [1] "2"                "255356.440134687" "from"
## [4] "72"               "to"               "83"
## [1] "3"                "418754.170251985" "from"
## [4] "73"               "to"               "84"
```

```
## [1] "4"                "162703.95832777" "from"             "74"
## [5] "to"               "85"
## [1] "5"                 "156106.265897493" "from"
## [4] "75"                "to"               "86"
## [1] "6"                 "182183.636203729" "from"
## [4] "76"                "to"               "87"
## [1] "7"                 "268254.956821843" "from"
## [4] "77"                "to"               "88"
## [1] "8"                 "1971628.63218859" "from"
## [4] "78"                "to"               "89"
## [1] "9"                 "571315.958261751" "from"
## [4] "79"                "to"               "90"
## [1] "10"                "81641.0722765731" "from"
## [4] "80"                "to"               "91"
## [1] "11"                "120810.973678179" "from"
## [4] "81"                "to"               "92"
## [1] "12"              "115532.6633156" "from"           "82"
## [5] "to"              "93"
## [1] "13"                "196103.120999801" "from"
## [4] "83"                "to"               "94"
## [1] "14"                "234409.012975293" "from"
## [4] "84"                "to"               "95"
## [1] "15"                "251622.578799367" "from"
## [4] "85"                "to"               "96"
## [1] "16"                "392996.391924179" "from"
## [4] "86"                "to"               "97"
## [1] "17"                "277705.385502035" "from"
## [4] "87"                "to"               "98"
## [1] "18"                "207603.384232895" "from"
## [4] "88"                "to"               "99"
## [1] "19"                "311793.482552047" "from"
## [4] "89"                "to"               "100"
## [1] "20"                "395001.014921577" "from"
## [4] "90"                "to"               "101"
## [1] "21"                "570862.335298966" "from"
## [4] "91"                "to"               "102"
## [1] "22"                "670760.164481223" "from"
## [4] "92"                "to"               "103"
## [1] "23"                "610677.041481182" "from"
## [4] "93"                "to"               "104"
## [1] "24"                "1088788.41882768" "from"
## [4] "94"                "to"               "105"
## [1] "25"                "1175791.48823579" "from"
## [4] "95"                "to"               "106"
## [1] "26"               "1323996.5019858" "from"             "96"
## [5] "to"               "107"
## [1] "27"                "2076256.36519727" "from"
## [4] "97"                "to"               "108"
## [1] "28"                "3096102.18937257" "from"
## [4] "98"                "to"               "109"
## [1] "29"               "3059463.9328674" "from"             "99"
## [5] "to"               "110"
## [1] "30"                "3235170.22479128" "from"
## [4] "100"               "to"               "111"
```

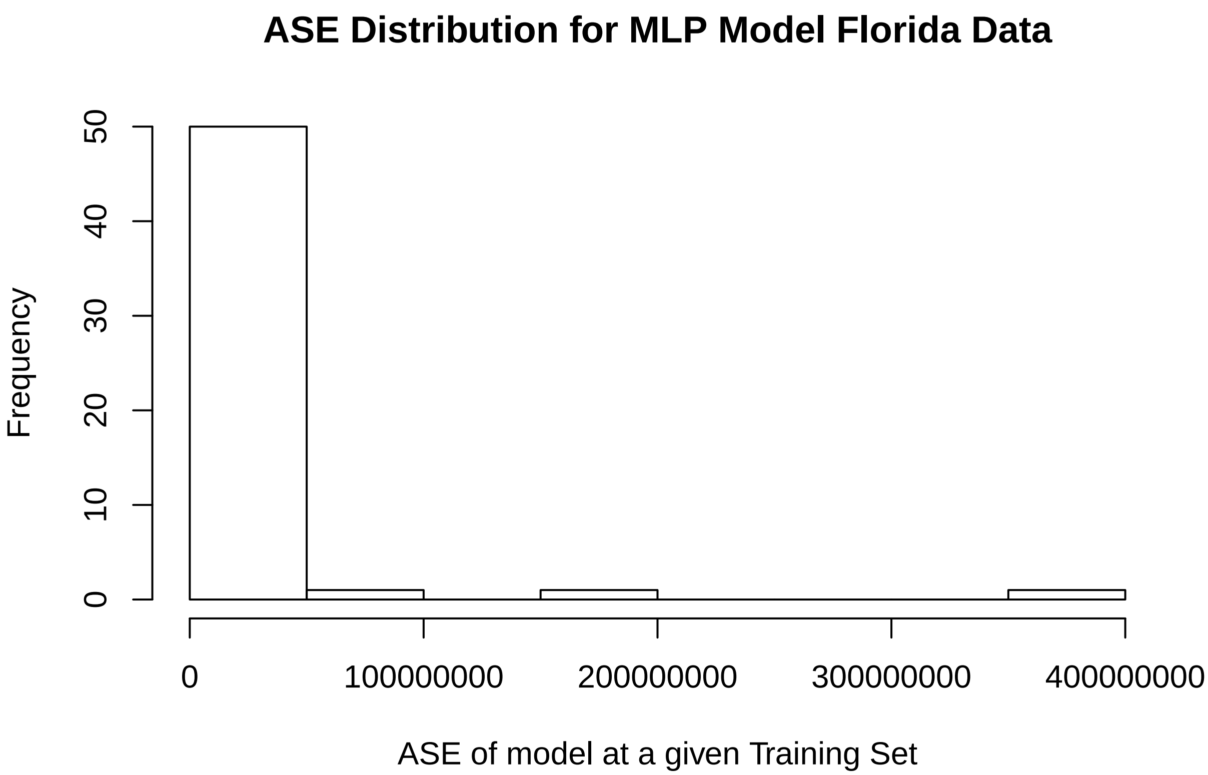
```
## [1] "31"               "2451420.26767279" "from"
## [4] "101"              "to"               "112"
## [1] "32"               "2012216.48886983" "from"
## [4] "102"              "to"               "113"
## [1] "33"             "2680232.375671" "from"           "103"
## [5] "to"             "114"
## [1] "34"               "6988470.88778208" "from"
## [4] "104"              "to"               "115"
## [1] "35"              "10765625.327423" "from"            "105"
## [5] "to"              "116"
## [1] "36"               "14503618.8310169" "from"
## [4] "106"              "to"               "117"
## [1] "37"               "10738796.4070242" "from"
## [4] "107"              "to"               "118"
## [1] "38"             "9212192.891214" "from"           "108"
## [5] "to"             "119"
## [1] "39"              "8492665.3411105" "from"            "109"
## [5] "to"              "120"
## [1] "40"               "9712764.24192737" "from"
## [4] "110"              "to"               "121"
## [1] "41"              "14278302.972415" "from"            "111"
## [5] "to"              "122"
## [1] "42"               "23598609.6743631" "from"
## [4] "112"              "to"               "123"
## [1] "43"               "11088235.2099952" "from"
## [4] "113"              "to"               "124"
## [1] "44"               "10850732.7252368" "from"
## [4] "114"              "to"               "125"
## [1] "45"               "154416666.919561" "from"
## [4] "115"              "to"               "126"
## [1] "46"               "359247367.361375" "from"
## [4] "116"              "to"               "127"
## [1] "47"              "29945753.014179" "from"            "117"
## [5] "to"              "128"
## [1] "48"              "50527842.939306" "from"            "118"
## [5] "to"              "129"
## [1] "49"               "14625637.7939288" "from"
## [4] "119"              "to"               "130"
## [1] "50"              "19854320.659604" "from"            "120"
## [5] "to"              "131"
## [1] "51"               "16457266.7269943" "from"
## [4] "121"              "to"               "132"
## [1] "52"               "15749915.6376319" "from"
## [4] "122"              "to"               "133"
## [1] "53"               "7136611.46972023" "from"
## [4] "123"              "to"               "134"
```

```
ASEHolder
```

```
##  [1]    310510.47    255356.44    418754.17    162703.96    156106.27
##  [6]    182183.64    268254.96   1971628.63    571315.96     81641.07
## [11]    120810.97    115532.66    196103.12    234409.01    251622.58
## [16]    392996.39    277705.39    207603.38    311793.48    395001.01
## [21]    570862.34    670760.16    610677.04   1088788.42   1175791.49
## [26]   1323996.50   2076256.37   3096102.19   3059463.93   3235170.22
```

```
## [31]   2451420.27   2012216.49   2680232.38   6988470.89  10765625.33
## [36]  14503618.83  10738796.41   9212192.89   8492665.34   9712764.24
## [41]  14278302.97  23598609.67  11088235.21  10850732.73 154416666.92
## [46] 359247367.36  29945753.01  50527842.94  14625637.79  19854320.66
## [51]  16457266.73  15749915.64   7136611.47
```

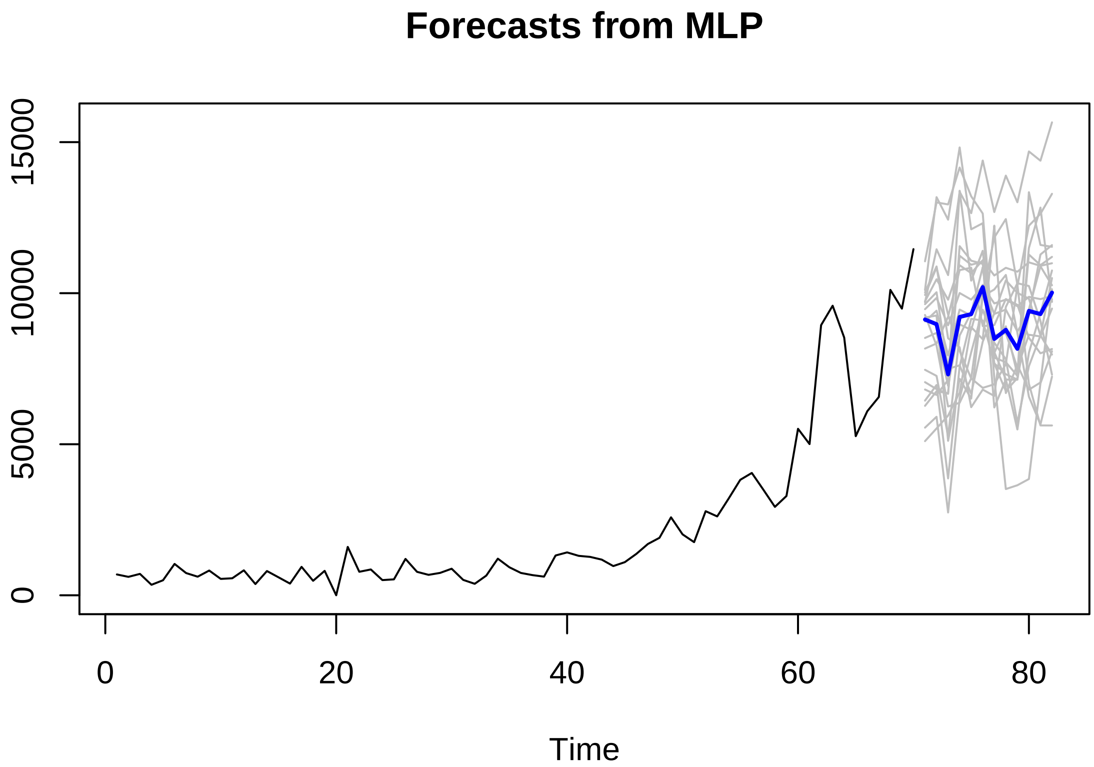
```r
#Distribution of ASEs on Two Week Periods
hist(ASEHolder, xlab = "ASE of model at a given Training Set",  main = "ASE Distribution for MLP Model
```



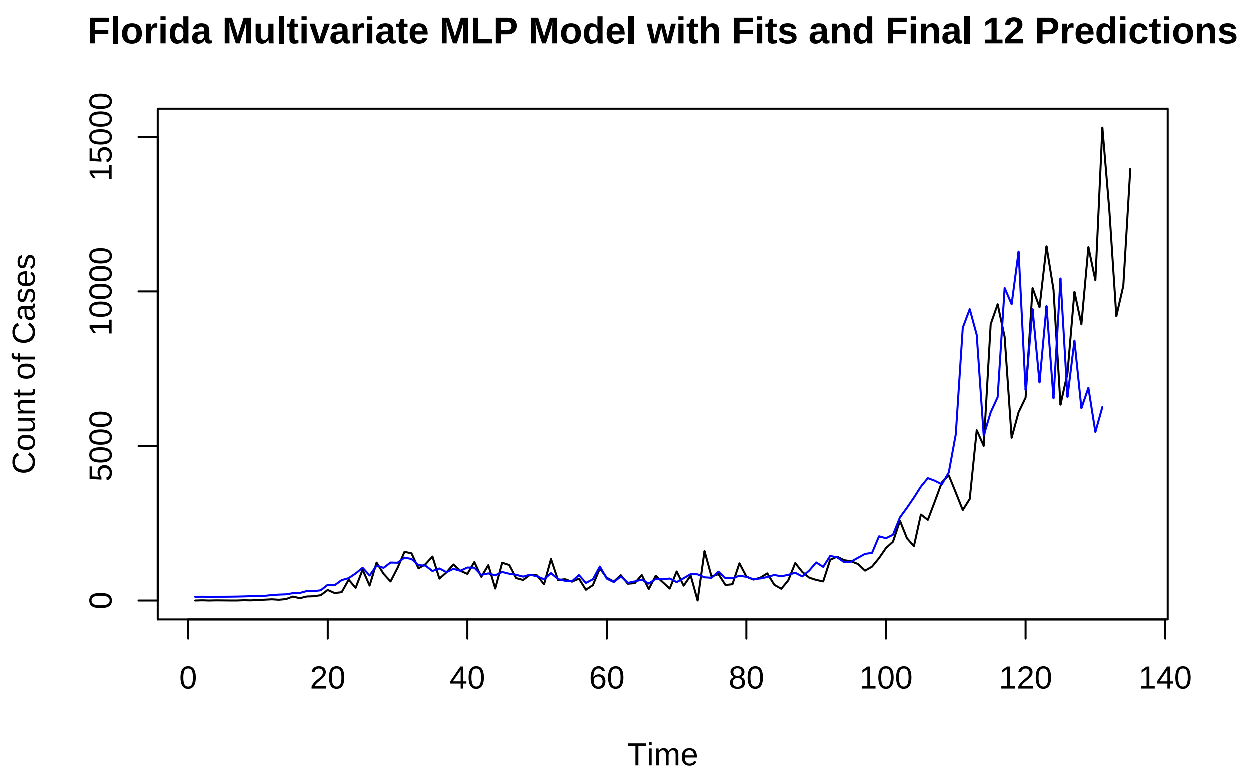

```r
#Mean ASE
WindowedASE = mean(ASEHolder)
WindowedASE
```

```
## [1] 15643871
```

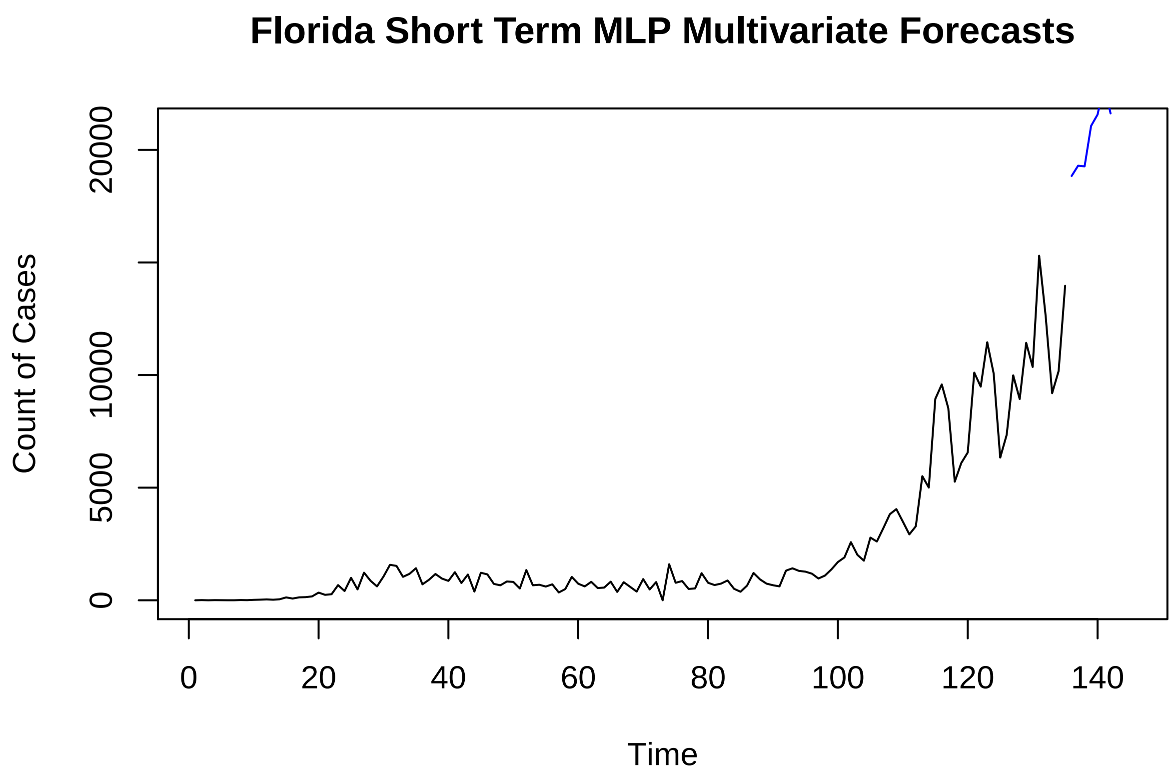
```r
#18757436 - 18 mill

plot(forecasts)
```



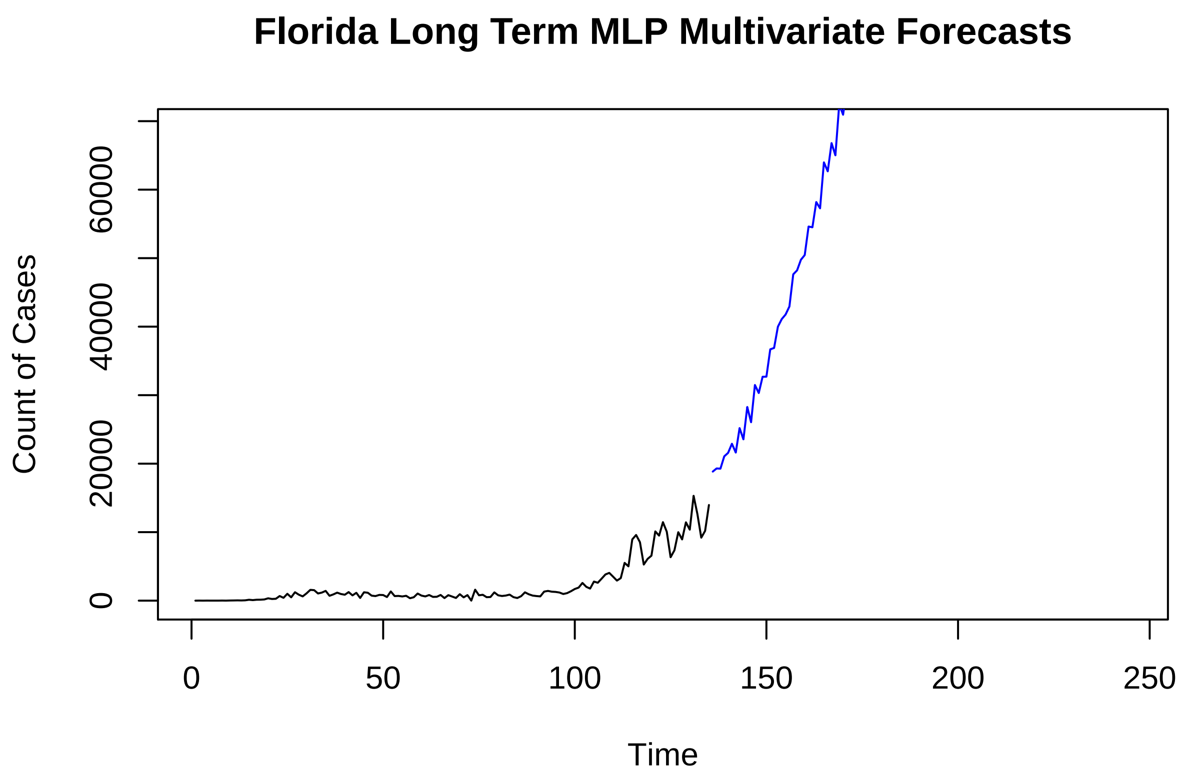

```
#Final Forecasts with data known
mlp.fit = mlp(ts(newcases_fl_multi$positiveIncrease[1:123]), hd = 5, comb = "median", xreg =newcases_fl_
forecasts = forecast(mlp.fit,h = 12, xreg = newcases_fl_var[1:135,])
fl_multi_mlp_fore = forecasts$mean

all_f = c(forecasts$fitted, forecasts$mean)
plot(newcases_fl_multi$positiveIncrease, type = "l", main = "Florida Multivariate MLP Model with Fits ar
lines(all_f, col = "blue")
```




```
#Forecast beyond data
mlp.fit = mlp(ts(newcases_fl_multi$positiveIncrease), hd = 5, comb = "median", xreg = newvar_fore_fl[1:
short_fl_mlp_m = forecast(mlp.fit,h = 7, xreg = newvar_fore_fl[1:145,])
long_fl_mlp_m = forecast(mlp.fit,h = 90, xreg = newvar_fore_fl[1:225,])

plot(newcases_fl$positiveIncrease, type = "l", xlim = c(1,145),ylim = c(0,21000), main = "Florida Short
lines(short_fl_mlp_m$mean, col = "blue")
```

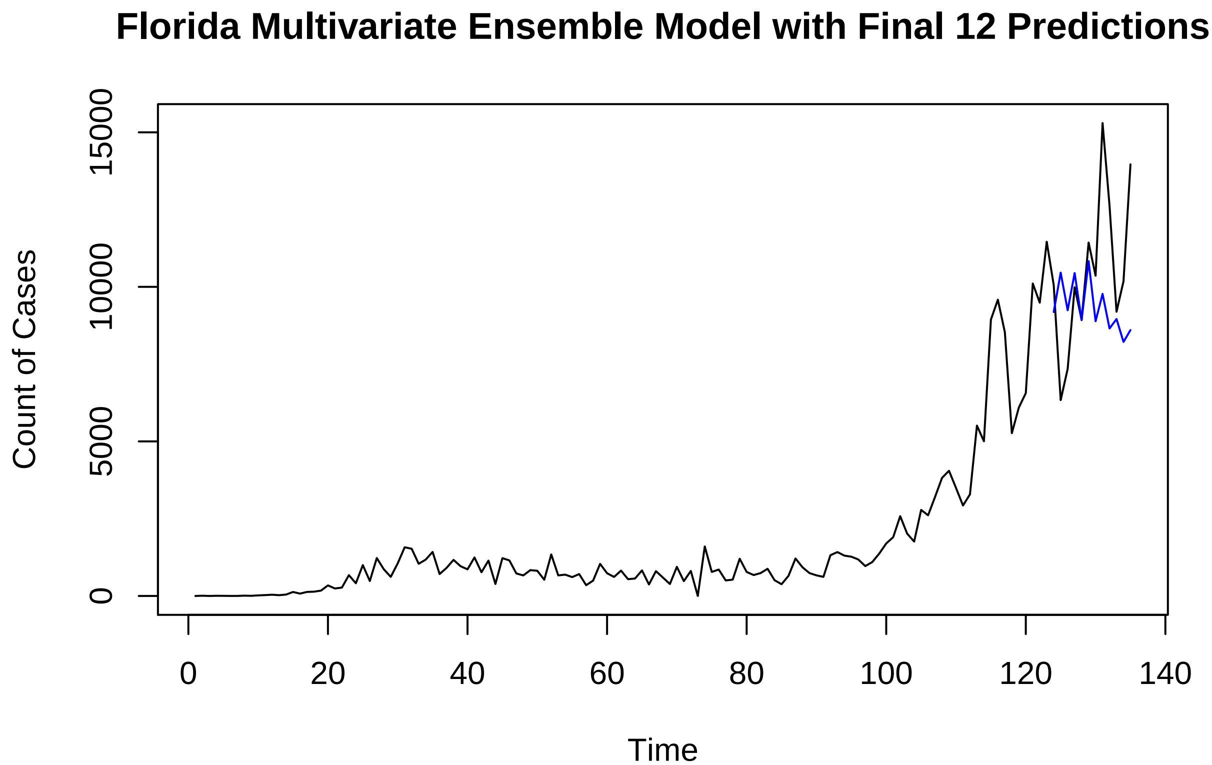



```r
plot(newcases_fl$positiveIncrease, type = "l", xlim = c(1,245),ylim = c(0,69000), main = "Florida Long 
lines(long_fl_mlp_m$mean, col = "blue")
```



####Florida Multivariate Ensemble Model

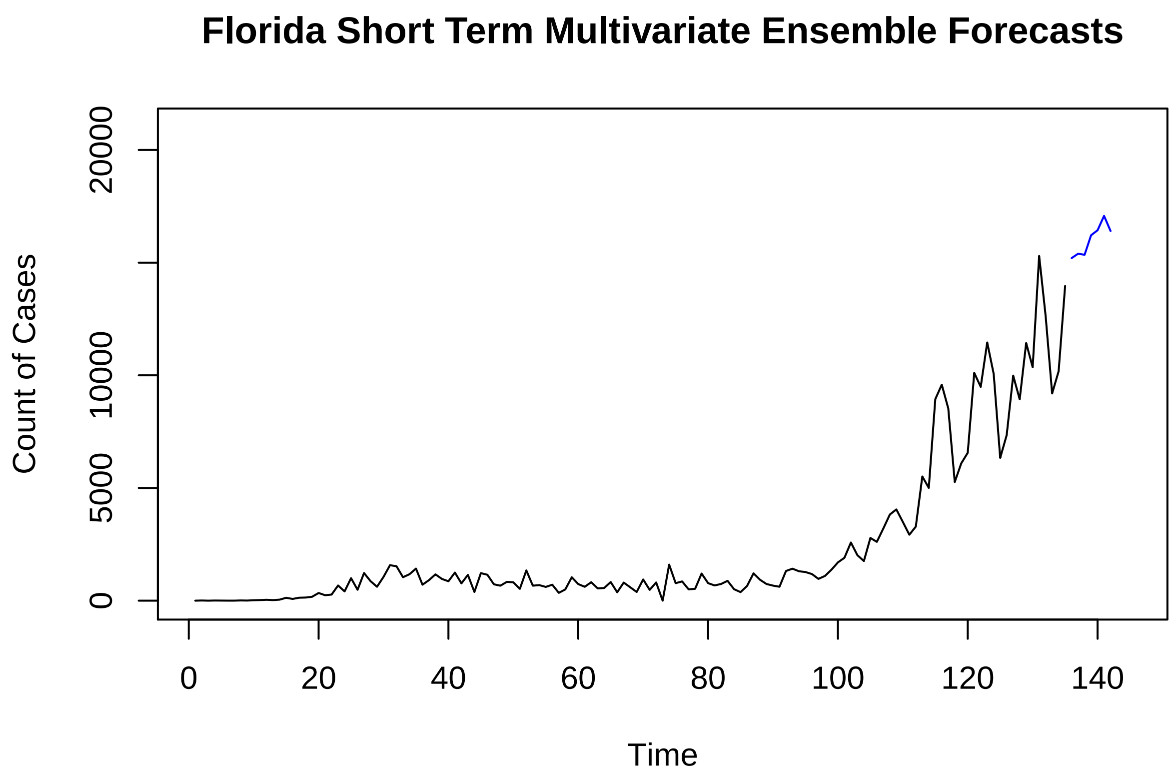
```r
ensemble_fore = (fl_multi_mlp_fore + FinalPredictions_fl_MLR)/2

plot(newcases_fl_multi$positiveIncrease, type = "l", main = "Florida Multivariate Ensemble Model with Fi
lines(ensemble_fore, col = "blue")
```



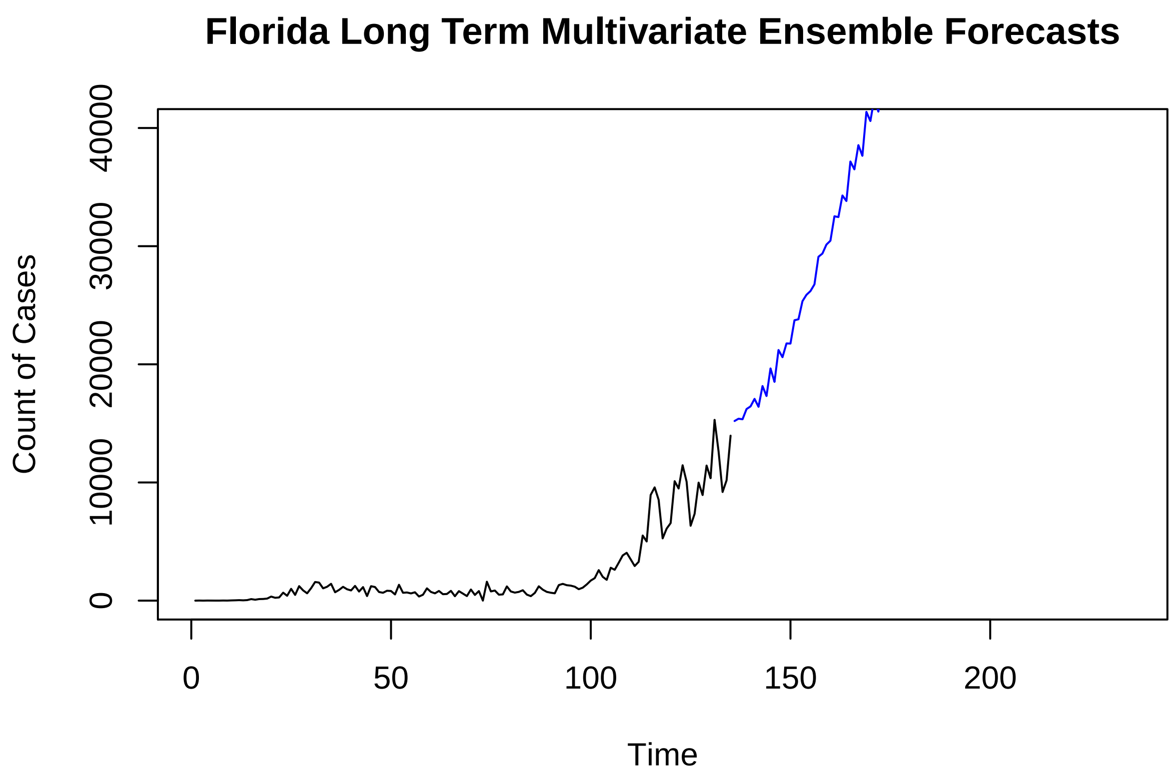

```r
ASE_fl_multi = mean((newcases_fl_multi$positiveIncrease[124:135] -ensemble_fore)^2)
ASE_fl_multi
```

```
## [1] 8591775
```

```r
#ASE of 8,427,522

#future

#long_fl_mlp_m
#short_fl_mlp_m


ensemble_fl_fore_short = ( short_fl_mlp_m$mean+ short_fl_mlr_m)/2
ensemble_fl_fore_long = (long_fl_mlp_m$mean + long_fl_mlr_m)/2

plot(newcases_fl$positiveIncrease, type = "l", xlim = c(1,145),ylim = c(0,21000), main = "Florida Short
lines(ensemble_fl_fore_short, col = "blue")
```




```
plot(newcases_fl$positiveIncrease, type = "l", xlim = c(1,235), ylim = c(0,40000),main = "Florida Long T
lines(ensemble_fl_fore_long, col = "blue")
```

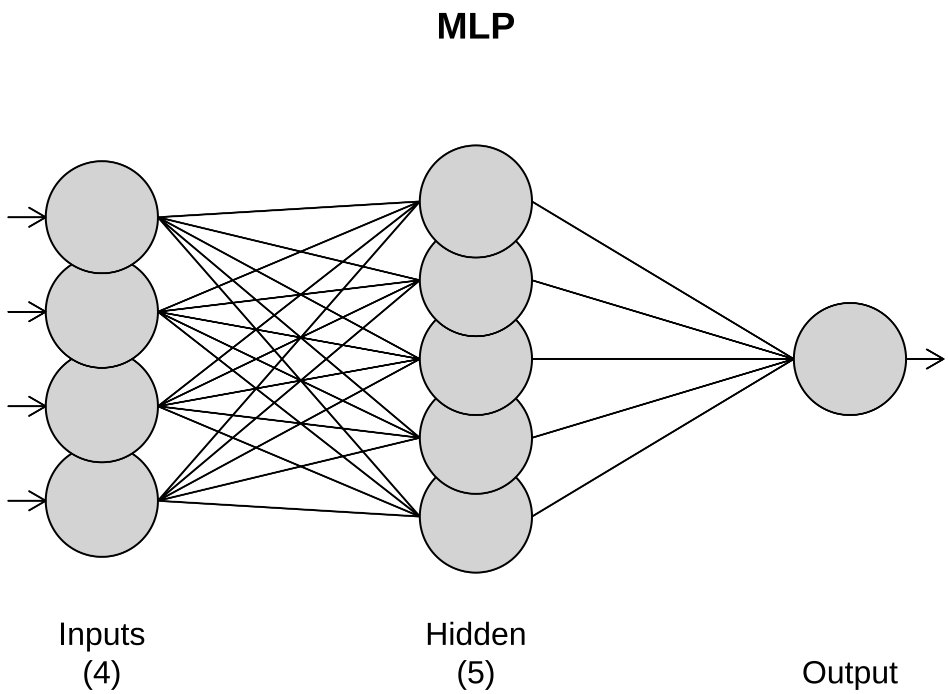


Compare Multivariate Models

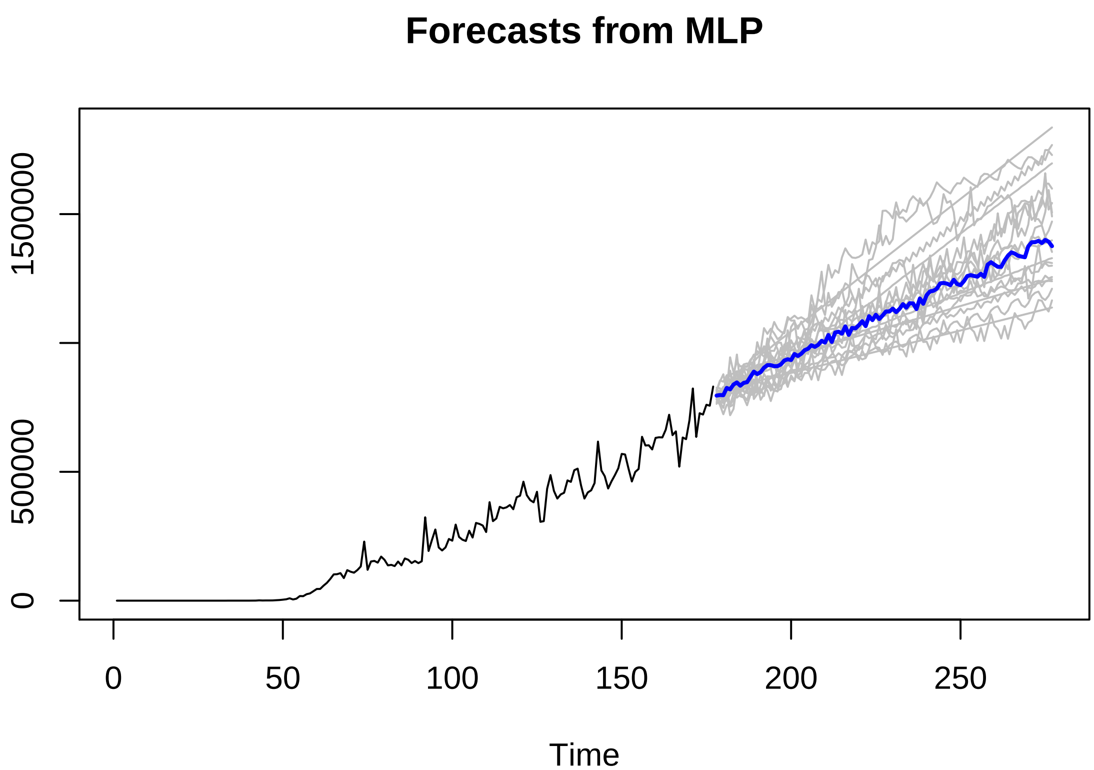
###Multivariate US Models

```r
#Forecast variables
```

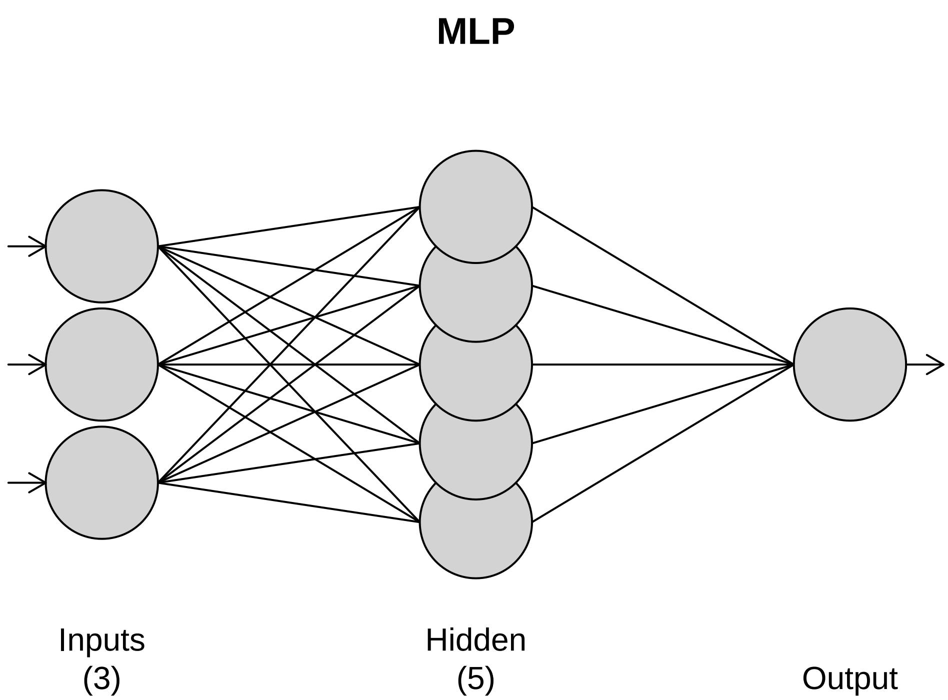
```r
newcases_us_multi =  initial_data_us %>% dplyr::select(positiveIncrease, totalTestResultsIncrease, hospi

#Forecast Future


#Forecast future variables
fit.mlp.1 = mlp(ts(newcases_us_multi$totalTestResultsIncrease),reps = 20, comb = "median")
plot(fit.mlp.1)
```



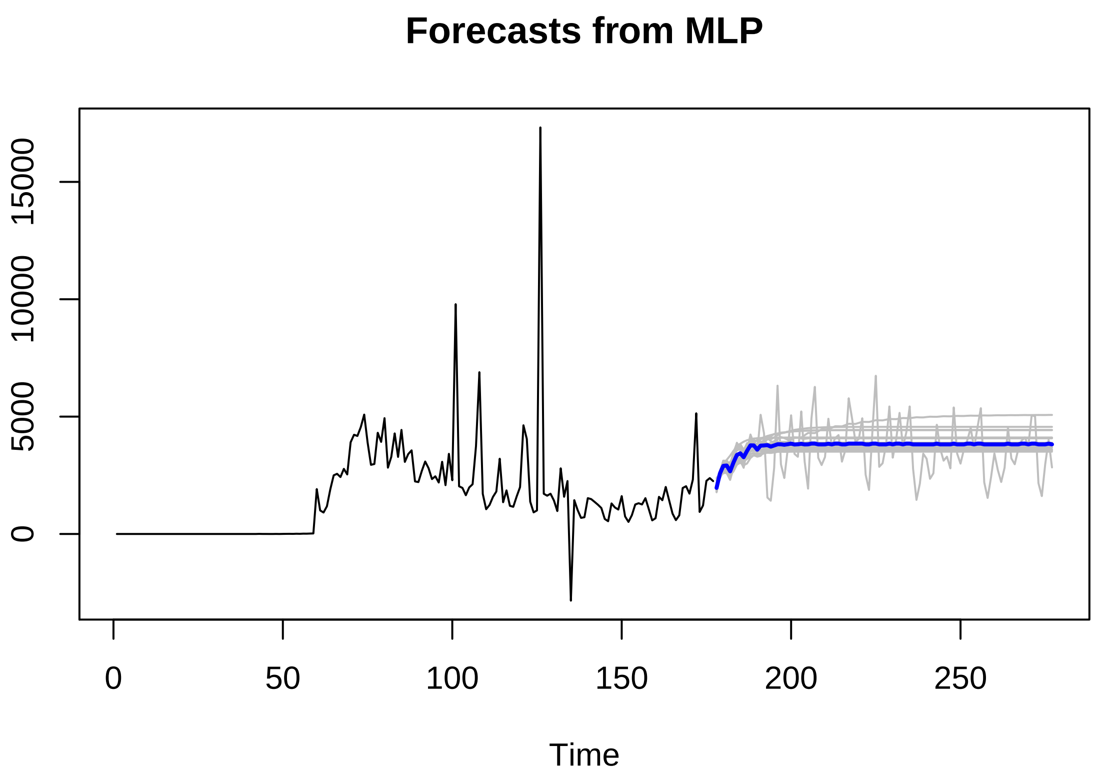

```r
fore.mlp.1 = forecast(fit.mlp.1, h = 100)
plot(fore.mlp.1)
```



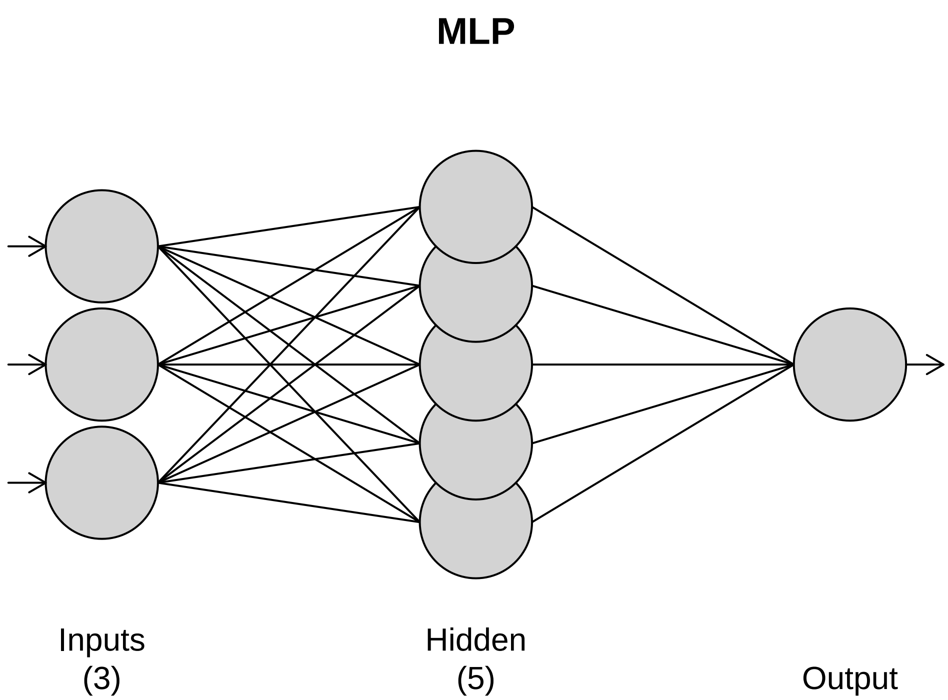

```
fit.mlp.2 = mlp(ts(newcases_us_multi$hospitalizedIncrease),reps = 20, comb = "median")
plot(fit.mlp.2)
```



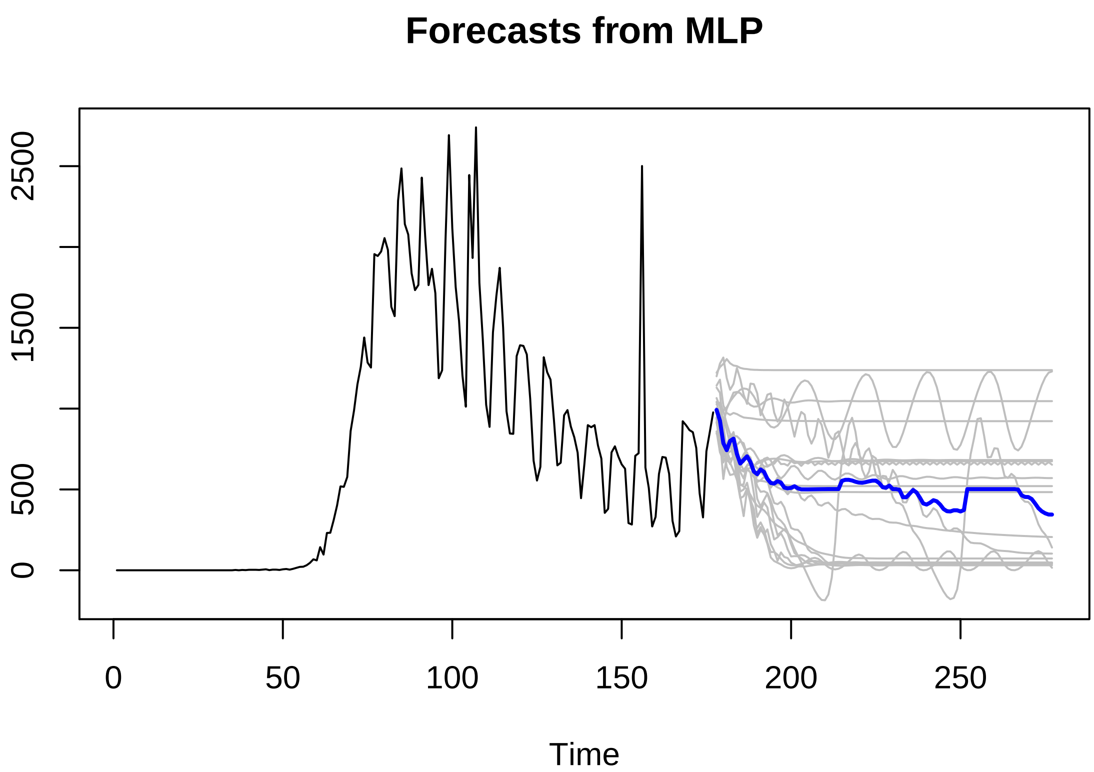

```
fore.mlp.2 = forecast(fit.mlp.2, h = 100)
plot(fore.mlp.2)
```




```r
fit.mlp.3 = mlp(ts(newcases_us_multi$deathIncrease),reps = 20, comb = "median")
plot(fit.mlp.3)
```




```r
fore.mlp.3 = forecast(fit.mlp.3, h = 100)
plot(fore.mlp.3)
```




```r
#package them up in data frame.
newvar_fore_us = data.frame(totalTestResultsIncrease = ts(c(newcases_us_multi$totalTestResultsIncrease,i

dim(newvar_fore_us)
```

```
## [1] 277   3
```

####US MLR with Correlated Errors Model

```r
fit = lm(positiveIncrease~totalTestResultsIncrease + hospitalizedIncrease, data = newcases_us_multi[1:1
summary(fit)
```

```
##
## Call:
## lm(formula = positiveIncrease ~ totalTestResultsIncrease + hospitalizedIncrease,
##     data = newcases_us_multi[1:165, ])
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -42222  -3101  -2878   5316  14897
##
## Coefficients:
##                             Estimate  Std. Error t value
## (Intercept)              3101.227341  869.662497   3.566
## totalTestResultsIncrease    0.049977    0.002809  17.793
## hospitalizedIncrease        2.336046    0.290613   8.038
##                                      Pr(>|t|)
## (Intercept)                          0.000477 ***
## totalTestResultsIncrease < 0.0000000000000002 ***
## hospitalizedIncrease        0.00000000000018 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7349 on 162 degrees of freedom
## Multiple R-squared:  0.743,  Adjusted R-squared:  0.7399
## F-statistic: 234.2 on 2 and 162 DF,  p-value: < 0.000000000000000022
```

```
est_tests = mean(tail(newcases_us_multi$totalTestResultsIncrease))
est_hospital= mean(tail(newcases_us_multi$hospitalizedIncrease))
newdata = data.frame(totalTestResultsIncrease = rep(est_tests,12), hospitalizedIncrease = rep(est_hospi

preds = predict(fit, newdata = newdata)



aic5.wge(fit$residuals)#picks 3,2 with full data
```

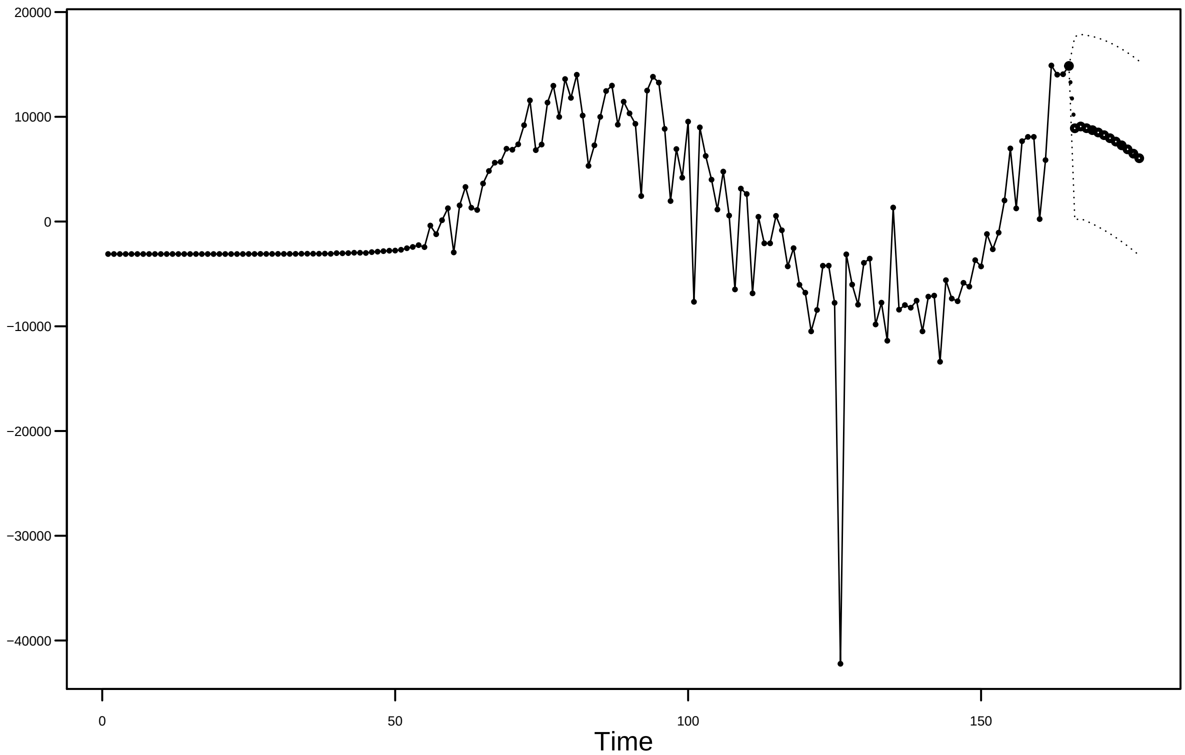
```
## ---------WORKING... PLEASE WAIT...
##
##
## Five Smallest Values of  aic
```

```
##       p    q        aic
## 5     1    1   16.83224
## 6     1    2   16.83708
## 11    3    1   16.83910
## 8     2    1   16.83932
## 13    4    0   16.83996
```

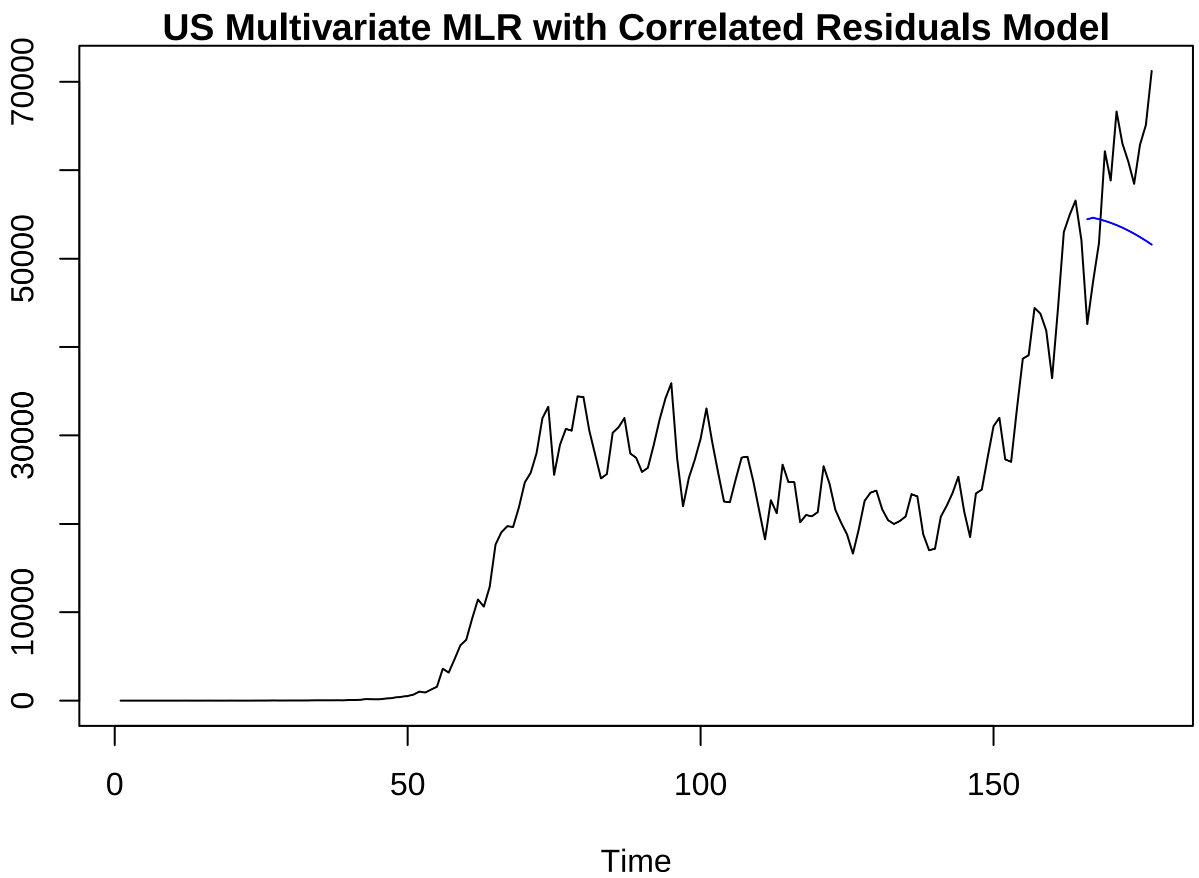
```
est1 = est.arma.wge(fit$residuals, p = 3, q = 2)
```

```
##
## Coefficients of Original polynomial:
## 1.9444 -0.9016 -0.0473
##
## Factor                 Roots                Abs Recip    System Freq
## 1-1.9919B+0.9961B^2    0.9998+-0.0652i      0.9981       0.0104
## 1+0.0474B             -21.0791               0.0474       0.5000
##
##
```

```
forecasts = fore.arma.wge(fit$residuals,phi = est1$phi,theta = est1$theta, lastn = FALSE,n.ahead = 12)
```



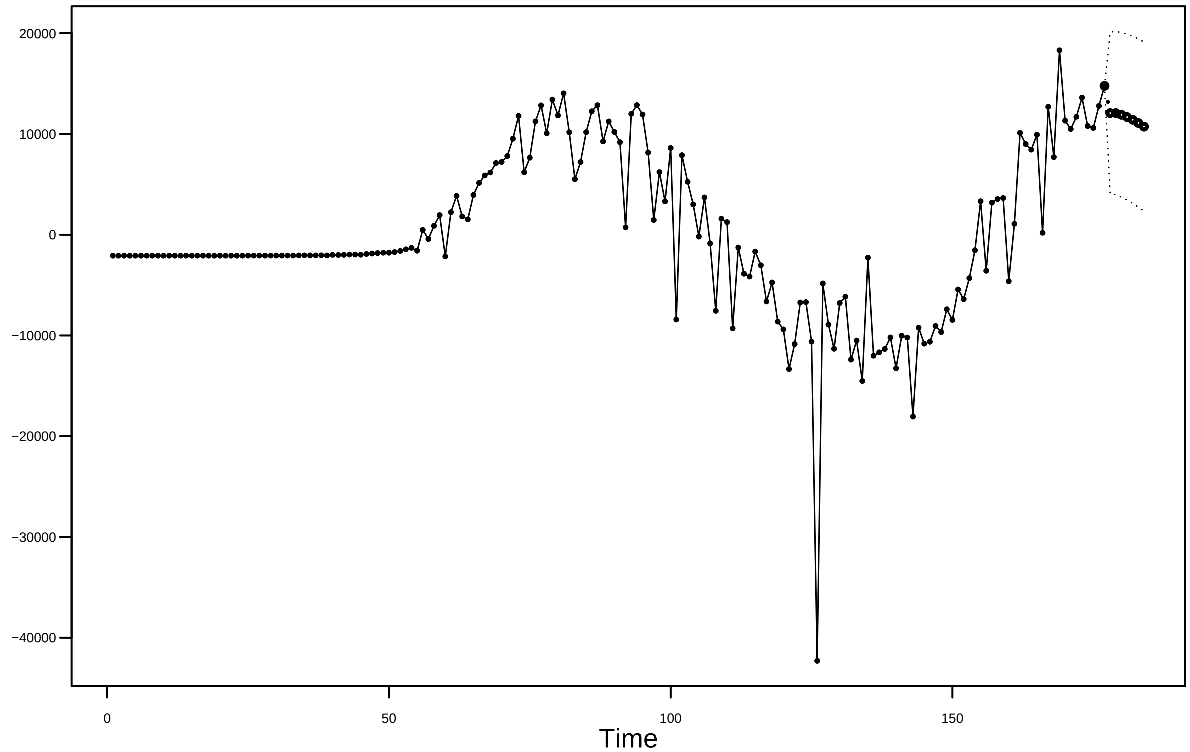

```r
FinalPredictions_us_MLR  = preds + forecasts$f

plot(newcases_us$positiveIncrease, type = "l", main ="US Multivariate MLR with Correlated Residuals Mode
lines(ts(FinalPredictions_us_MLR, start = 166), col = "blue")
```



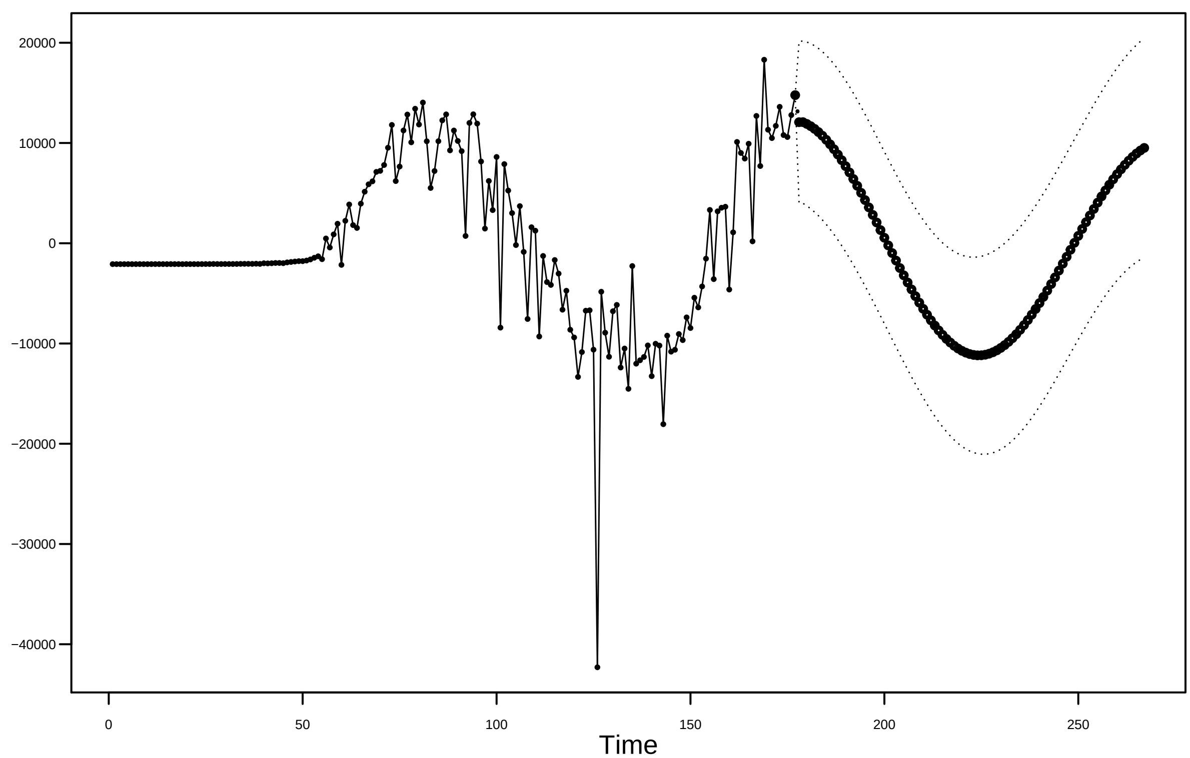

```
ASE = mean((newcases_us_multi$positiveIncrease[166:177] - FinalPredictions_us_MLR)^2)
ASE
```

```
## [1] 108181241
```

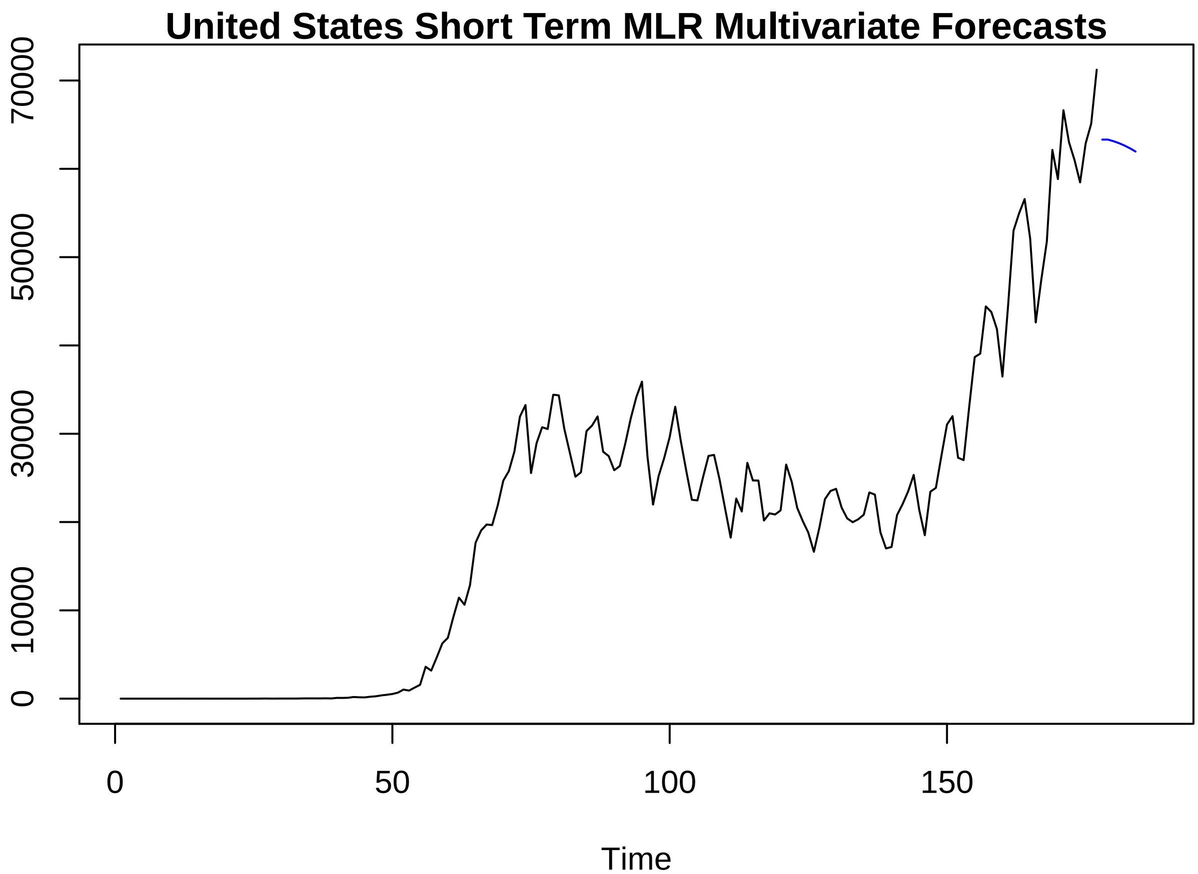
```
#108181241

#Forecasting Ahead

shortdata = data.frame(totalTestResultsIncrease = rep(est_tests,7), hospitalizedIncrease = rep(est_hospi
longdata =data.frame(totalTestResultsIncrease = rep(est_tests,90), hospitalizedIncrease = rep(est_hospit


fit = lm(positiveIncrease~totalTestResultsIncrease + hospitalizedIncrease, data = newcases_us_multi)
#short
preds = predict(fit, newdata = shortdata)
forecasts = fore.arma.wge(fit$residuals,phi = est1$phi,theta = est1$theta, lastn = FALSE,n.ahead = 7)
```



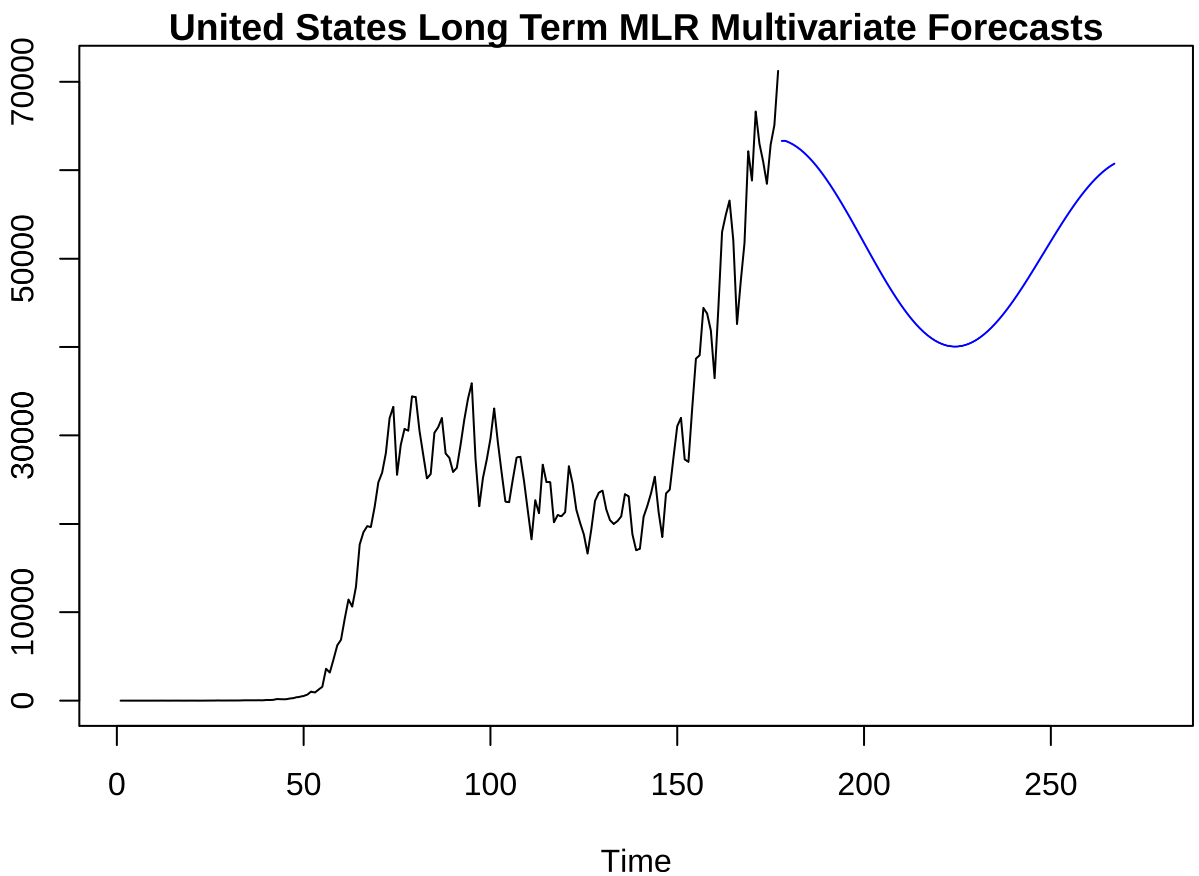
```
short_us_mlr_m  = preds + forecasts$f

#long
preds = predict(fit, newdata = longdata)
forecasts = fore.arma.wge(fit$residuals,phi = est1$phi,theta = est1$theta, lastn = FALSE,n.ahead = 90)
```



```
long_us_mlr_m  = preds + forecasts$f


plot(newcases_us$positiveIncrease, type = "l", xlim = c(1,187), main = "United States Short Term MLR Mul
lines(ts(short_us_mlr_m, start = 178), col = "blue")
```




```
plot(newcases_us$positiveIncrease, type = "l", xlim = c(1,277), main = "United States Long Term MLR Mult
lines(ts(long_us_mlr_m,start = 178), col = "blue")
```



####US MLP/RNN Model

```
trainingSize = 70
horizon = 12
ASEHolder = numeric()

for( i in 1:(177-(trainingSize + horizon) + 1))
{
  mlp.fit = mlp(ts(newcases_us_multi$positiveIncrease[1:trainingSize+i]), hd = 5, comb = "median", xreg
  forecasts = forecast(mlp.fit,h = horizon, xreg = newvar_fore_us[1:(trainingSize + i + 13),])

  ASE = mean((newcases_us_multi$positiveIncrease[(trainingSize+i):(trainingSize+ i + (horizon) - 1)] -fo
  print(c(i,ASE, "from",trainingSize+i,"to",(trainingSize+ i + (horizon) - 1)))
  ASEHolder[i] = ASE

}
```

```
## [1] "1"                  "48281868.4402562" "from"
## [4] "71"                 "to"               "82"
## [1] "2"                  "326753763.770601" "from"
## [4] "72"                 "to"               "83"
## [1] "3"                  "1175894190.22742" "from"
## [4] "73"                 "to"               "84"
## [1] "4"                  "415557230.924249" "from"
## [4] "74"                 "to"               "85"
## [1] "5"                  "1330294937.39939" "from"
## [4] "75"                 "to"               "86"
```

```
## [1] "6"                 "19994942.4751049" "from"
## [4] "76"                "to"               "87"
## [1] "7"                 "159352305.609591" "from"
## [4] "77"                "to"               "88"
## [1] "8"                 "104926770.295105" "from"
## [4] "78"                "to"               "89"
## [1] "9"                 "241304356.295009" "from"
## [4] "79"                "to"               "90"
## [1] "10"                "197317591.246749" "from"
## [4] "80"                "to"               "91"
## [1] "11"                "147570805.912179" "from"
## [4] "81"                "to"               "92"
## [1] "12"                "65750564.8438156" "from"
## [4] "82"                "to"               "93"
## [1] "13"                "31138756.9806382" "from"
## [4] "83"                "to"               "94"
## [1] "14"                "30800145.9575522" "from"
## [4] "84"                "to"               "95"
## [1] "15"                "22750284.9088245" "from"
## [4] "85"                "to"               "96"
## [1] "16"               "23516102.357341" "from"             "86"
## [5] "to"               "97"
## [1] "17"                "24616623.0796707" "from"
## [4] "87"                "to"               "98"
## [1] "18"                "38619269.5347066" "from"
## [4] "88"                "to"               "99"
## [1] "19"                "40731790.1367565" "from"
## [4] "89"                "to"               "100"
## [1] "20"                "33297607.8765436" "from"
## [4] "90"                "to"               "101"
## [1] "21"                "23974839.8136231" "from"
## [4] "91"                "to"               "102"
## [1] "22"                "12763200.9446105" "from"
## [4] "92"                "to"               "103"
## [1] "23"                "14145308.8082222" "from"
## [4] "93"                "to"               "104"
## [1] "24"                "29810070.5483037" "from"
## [4] "94"                "to"               "105"
## [1] "25"                "34327506.9817523" "from"
## [4] "95"                "to"               "106"
## [1] "26"               "37774183.563801" "from"             "96"
## [5] "to"               "107"
## [1] "27"                "24605995.4291746" "from"
## [4] "97"                "to"               "108"
## [1] "28"                "27882389.1204092" "from"
## [4] "98"                "to"               "109"
## [1] "29"                "27880819.5079144" "from"
## [4] "99"                "to"               "110"
## [1] "30"                "38385551.3797368" "from"
## [4] "100"               "to"               "111"
## [1] "31"                "51448881.6740668" "from"
## [4] "101"               "to"               "112"
## [1] "32"                "68629880.0463857" "from"
## [4] "102"               "to"               "113"
```

```
## [1] "33"                "57403880.9251903" "from"
## [4] "103"               "to"               "114"
## [1] "34"                "54207876.1828415" "from"
## [4] "104"               "to"               "115"
## [1] "35"                "46517097.7684612" "from"
## [4] "105"               "to"               "116"
## [1] "36"                "53798933.9551493" "from"
## [4] "106"               "to"               "117"
## [1] "37"                "53362717.5542341" "from"
## [4] "107"               "to"               "118"
## [1] "38"                "63237646.3039525" "from"
## [4] "108"               "to"               "119"
## [1] "39"                "70480626.9012871" "from"
## [4] "109"               "to"               "120"
## [1] "40"               "65084324.732072" "from"            "110"
## [5] "to"               "121"
## [1] "41"                "48303072.5059537" "from"
## [4] "111"               "to"               "122"
## [1] "42"                "54688784.8309347" "from"
## [4] "112"               "to"               "123"
## [1] "43"                "52946908.2876032" "from"
## [4] "113"               "to"               "124"
## [1] "44"                "68111083.4311676" "from"
## [4] "114"               "to"               "125"
## [1] "45"                "82538637.1398303" "from"
## [4] "115"               "to"               "126"
## [1] "46"                "98746178.9617348" "from"
## [4] "116"               "to"               "127"
## [1] "47"                "94207755.1010443" "from"
## [4] "117"               "to"               "128"
## [1] "48"                "86229390.4141236" "from"
## [4] "118"               "to"               "129"
## [1] "49"                "81455639.2630436" "from"
## [4] "119"               "to"               "130"
## [1] "50"                "68607223.3810632" "from"
## [4] "120"               "to"               "131"
## [1] "51"                "81307279.1988982" "from"
## [4] "121"               "to"               "132"
## [1] "52"                "91799483.4559777" "from"
## [4] "122"               "to"               "133"
## [1] "53"                "96676320.1256956" "from"
## [4] "123"               "to"               "134"
## [1] "54"                "92877759.0719886" "from"
## [4] "124"               "to"               "135"
## [1] "55"               "88432439.246575" "from"            "125"
## [5] "to"               "136"
## [1] "56"                "69805839.4181132" "from"
## [4] "126"               "to"               "137"
## [1] "57"                "21866758.2004989" "from"
## [4] "127"               "to"               "138"
## [1] "58"                "46445701.5487145" "from"
## [4] "128"               "to"               "139"
## [1] "59"                "95090958.5958462" "from"
## [4] "129"               "to"               "140"
```

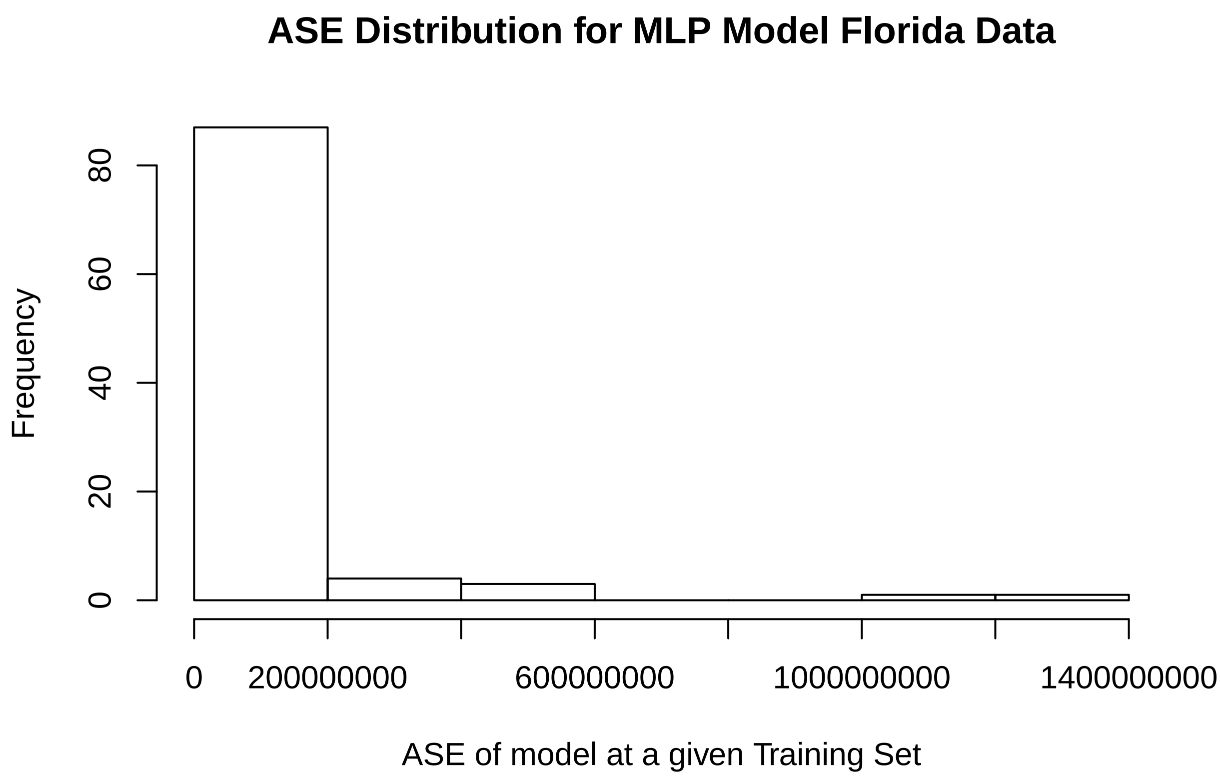
```
## [1] "60"              "104718554.4906" "from"           "130"
## [5] "to"              "141"
## [1] "61"                "106929898.652011" "from"
## [4] "131"               "to"               "142"
## [1] "62"               "96028515.596435" "from"            "132"
## [5] "to"               "143"
## [1] "63"                "95717431.8787642" "from"
## [4] "133"               "to"               "144"
## [1] "64"                "72187131.3141527" "from"
## [4] "134"               "to"               "145"
## [1] "65"                "89934327.2084688" "from"
## [4] "135"               "to"               "146"
## [1] "66"                "98274218.7229663" "from"
## [4] "136"               "to"               "147"
## [1] "67"               "99368483.470545" "from"            "137"
## [5] "to"               "148"
## [1] "68"                "99943764.5908397" "from"
## [4] "138"               "to"               "149"
## [1] "69"                "68009038.2567152" "from"
## [4] "139"               "to"               "150"
## [1] "70"                "50483846.1585762" "from"
## [4] "140"               "to"               "151"
## [1] "71"                "78097624.3137918" "from"
## [4] "141"               "to"               "152"
## [1] "72"                "79882666.9196373" "from"
## [4] "142"               "to"               "153"
## [1] "73"                "69883232.3082423" "from"
## [4] "143"               "to"               "154"
## [1] "74"                "82487074.4355033" "from"
## [4] "144"               "to"               "155"
## [1] "75"                "48349346.0760678" "from"
## [4] "145"               "to"               "156"
## [1] "76"                "52837305.4280329" "from"
## [4] "146"               "to"               "157"
## [1] "77"                "57101814.7166769" "from"
## [4] "147"               "to"               "158"
## [1] "78"                "72738958.9890754" "from"
## [4] "148"               "to"               "159"
## [1] "79"               "76795780.416462" "from"            "149"
## [5] "to"               "160"
## [1] "80"               "54652986.785473" "from"            "150"
## [5] "to"               "161"
## [1] "81"               "69715219.777838" "from"            "151"
## [5] "to"               "162"
## [1] "82"                "96192495.5188195" "from"
## [4] "152"               "to"               "163"
## [1] "83"                "81096725.3503932" "from"
## [4] "153"               "to"               "164"
## [1] "84"                "191113182.364886" "from"
## [4] "154"               "to"               "165"
## [1] "85"                "166756395.054215" "from"
## [4] "155"               "to"               "166"
## [1] "86"              "163395472.4025" "from"           "156"
## [5] "to"              "167"
```

```
## [1] "87"                "210440385.297531" "from"
## [4] "157"               "to"               "168"
## [1] "88"                "189917125.003633" "from"
## [4] "158"               "to"               "169"
## [1] "89"                "321095138.880825" "from"
## [4] "159"               "to"               "170"
## [1] "90"                "445890017.750138" "from"
## [4] "160"               "to"               "171"
## [1] "91"                "518254204.260824" "from"
## [4] "161"               "to"               "172"
## [1] "92"                "139471722.660578" "from"
## [4] "162"               "to"               "173"
## [1] "93"                "141635128.978134" "from"
## [4] "163"               "to"               "174"
## [1] "94"                "58256676.8433485" "from"
## [4] "164"               "to"               "175"
## [1] "95"              "76689829.728139" "from"            "165"
## [5] "to"              "176"
## [1] "96"              "69822133.674243" "from"            "166"
## [5] "to"              "177"
```

```r
ASEHolder
```

```
##  [1]   48281868  326753764 1175894190  415557231 1330294937   19994942
##  [7]  159352306  104926770  241304356  197317591  147570806   65750565
## [13]   31138757   30800146   22750285   23516102   24616623   38619270
## [19]   40731790   33297608   23974840   12763201   14145309   29810071
## [25]   34327507   37774184   24605995   27882389   27880820   38385551
## [31]   51448882   68629880   57403881   54207876   46517098   53798934
## [37]   53362718   63237646   70480627   65084325   48303073   54688785
## [43]   52946908   68111083   82538637   98746179   94207755   86229390
## [49]   81455639   68607223   81307279   91799483   96676320   92877759
## [55]   88432439   69805839   21866758   46445702   95090959  104718554
## [61]  106929899   96028516   95717432   72187131   89934327   98274219
## [67]   99368483   99943765   68009038   50483846   78097624   79882667
## [73]   69883232   82487074   48349346   52837305   57101815   72738959
## [79]   76795780   54652987   69715220   96192496   81096725  191113182
## [85]  166756395  163395472  210440385  189917125  321095139  445890018
## [91]  518254204  139471723  141635129   58256677   76689830   69822134
```

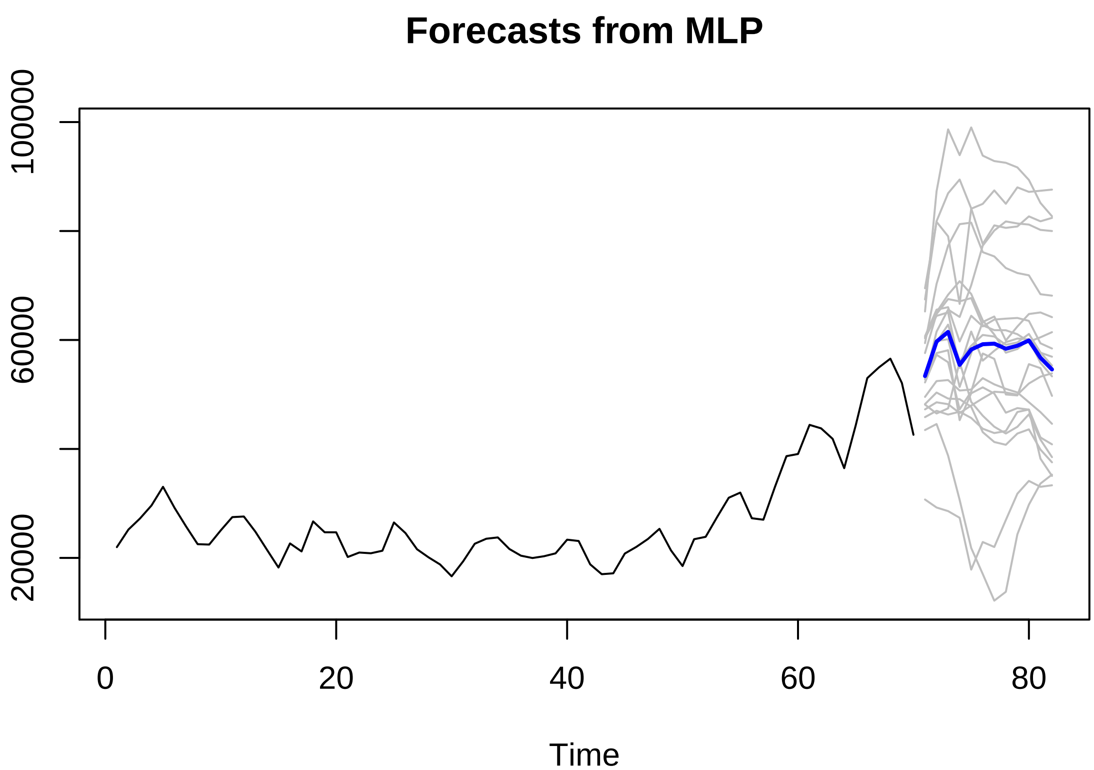
```r
#Distribution of ASEs on Two Week Periods
hist(ASEHolder, xlab = "ASE of model at a given Training Set",  main = "ASE Distribution for MLP Model I
```



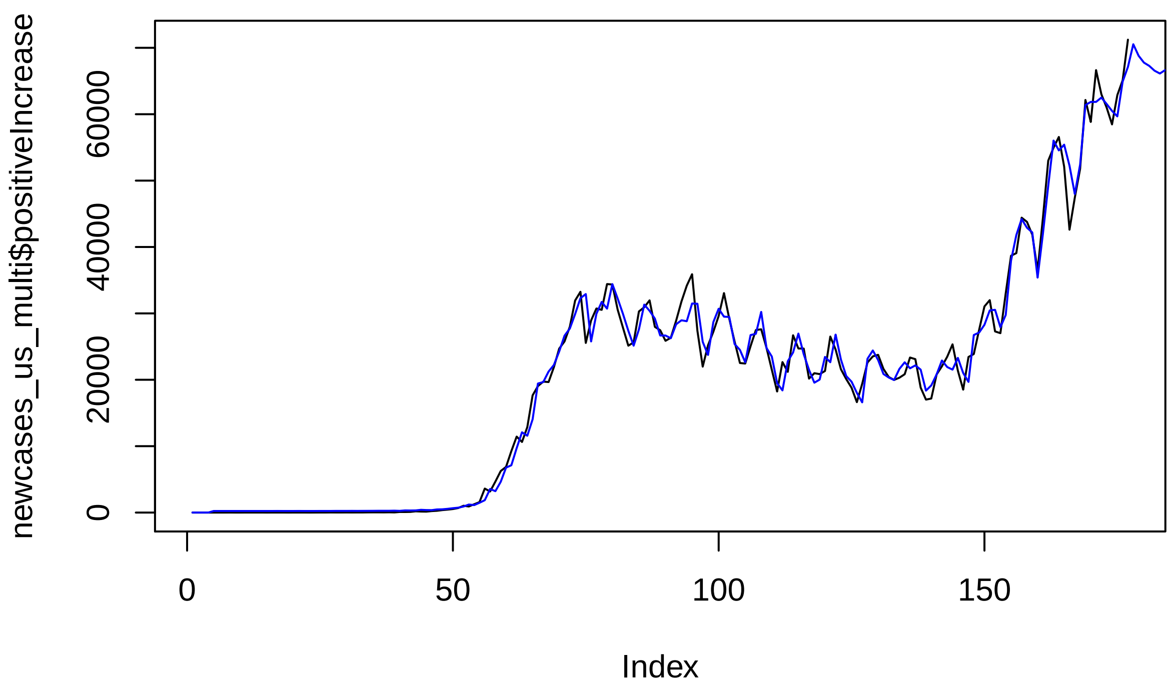

```
#Mean ASE
WindowedASE = mean(ASEHolder)
WindowedASE
```

```
## [1] 118213466
```

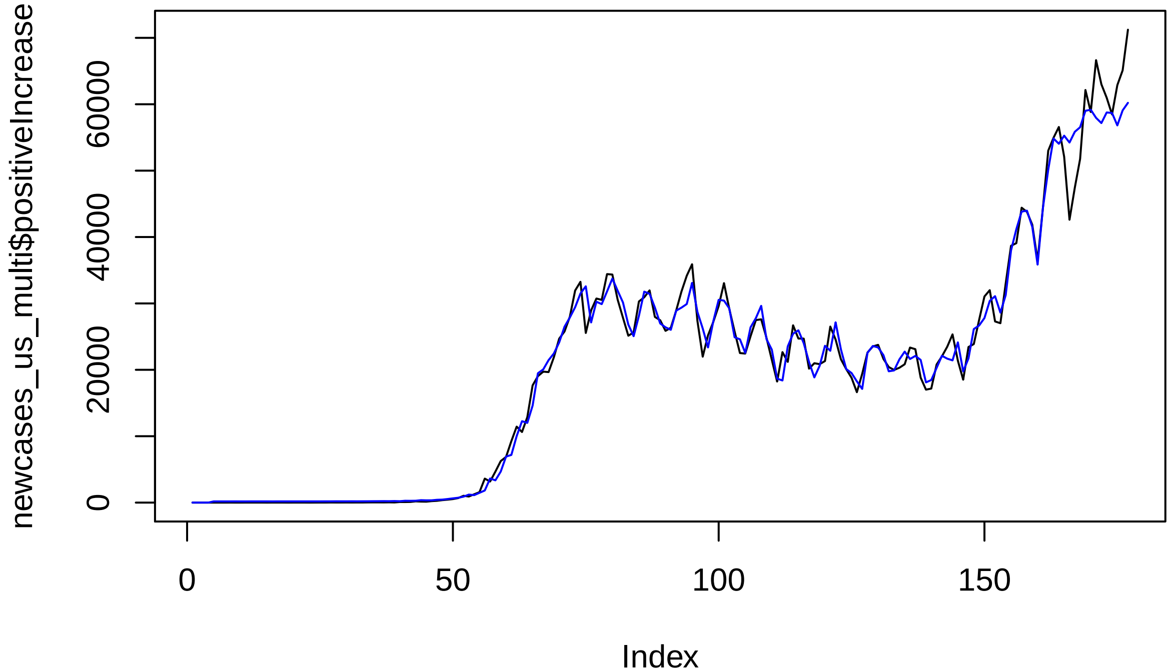
```
#97494363

plot(forecasts)
```




```r
#Final Forecasts with data known
mlp.fit = mlp(ts(newcases_us_multi$positiveIncrease[1:177]), hd = 5, comb = "median", xreg = newvar_fore
forecasts_us_mlp = forecast(mlp.fit,h = 12, xreg = newvar_fore_us[1:190,])

all_f = c(rep(1,4),forecasts_us_mlp$fitted, forecasts_us_mlp$mean)
plot(newcases_us_multi$positiveIncrease, type = "l")
lines(all_f, col = "blue")
```



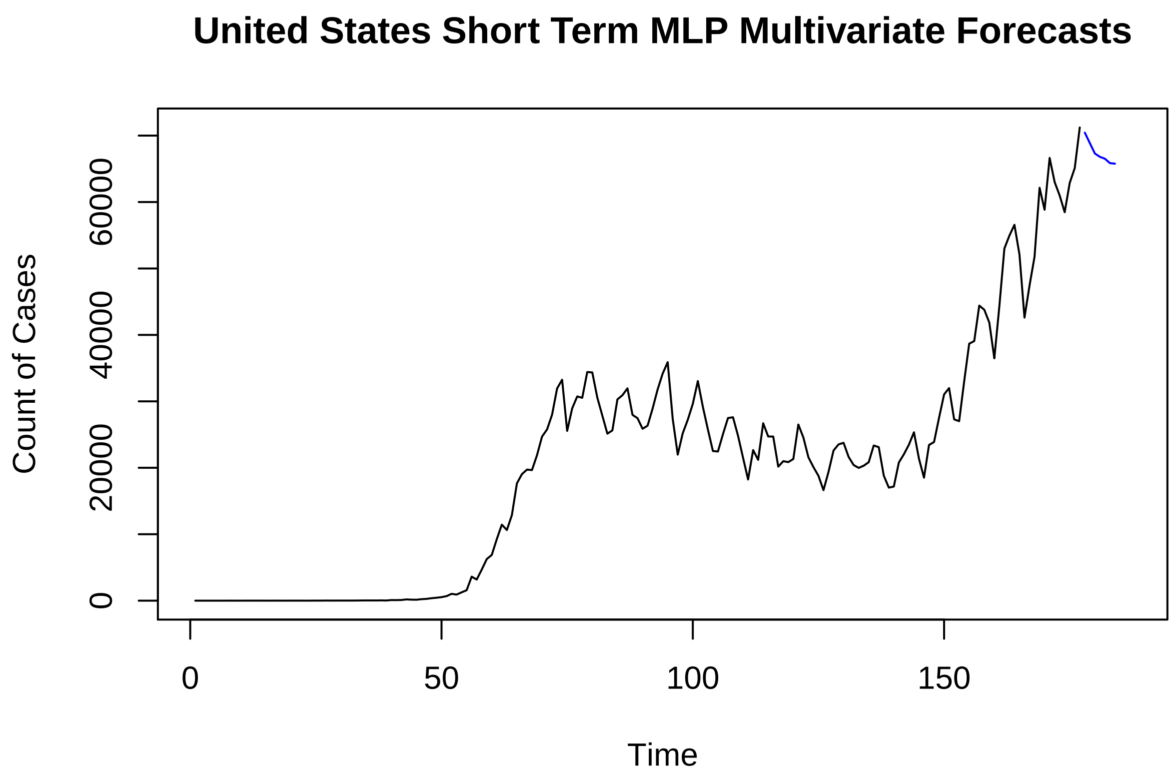
```
#final 12 forecasts

mlp.fit = mlp(ts(newcases_us_multi$positiveIncrease[1:165]), hd = 5, comb = "median", xreg = newvar_fore
forecasts_us_mlp = forecast(mlp.fit,h = 12, xreg = newvar_fore_us[1:177,])

all_f = c(rep(1,4),forecasts_us_mlp$fitted, forecasts_us_mlp$mean)
plot(newcases_us_multi$positiveIncrease, type = "l")
lines(all_f, col = "blue")
```



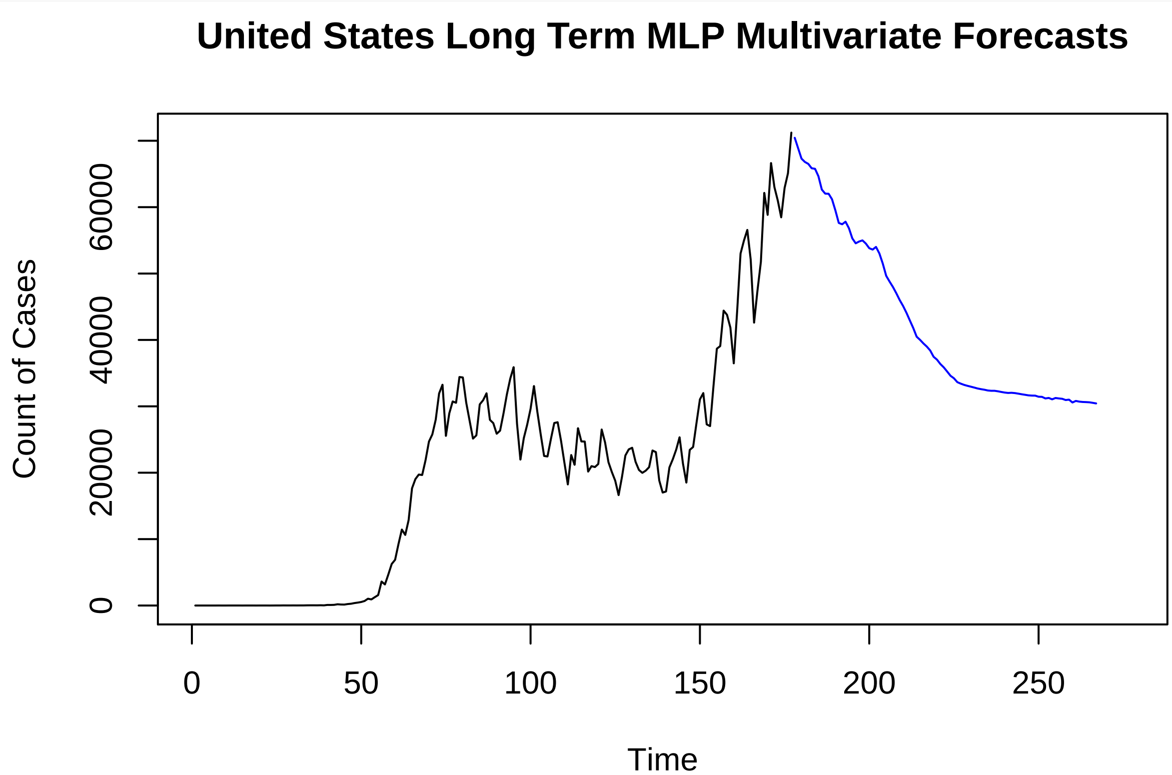

```
ASE_final12 = mean((newcases_us_multi$positiveIncrease[166:177] -forecasts_us_mlp$mean)^2)
ASE_final12
```

```
## [1] 45669535
```

```
#45799110


#Future Forecasts
mlp.fit = mlp(ts(newcases_us_multi$positiveIncrease), hd = 5, comb = "median", xreg = newvar_fore_us[1:
short_us_mlp_m = forecast(mlp.fit,h = 7, xreg = newvar_fore_us[1:187,])
long_us_mlp_m = forecast(mlp.fit,h = 90, xreg = newvar_fore_us[1:267,])

plot(newcases_us$positiveIncrease, type = "l", xlim = c(1,187), main = "United States Short Term MLP Mul
lines(short_us_mlp_m$mean, col = "blue")
```

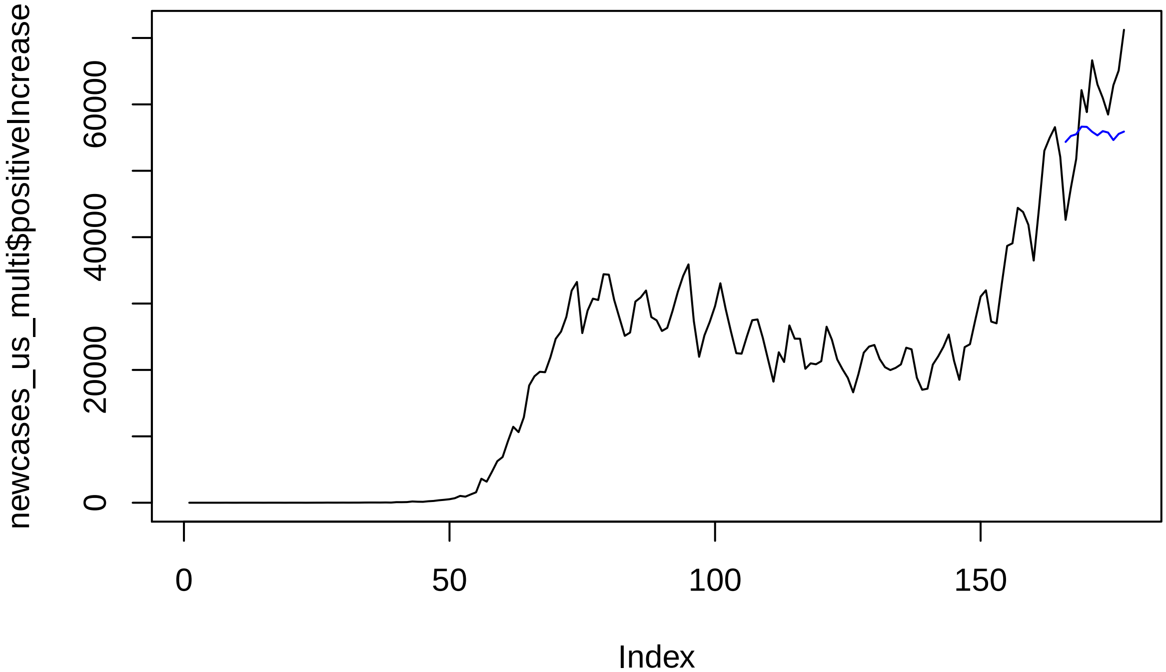



```
plot(newcases_us$positiveIncrease, type = "l", xlim = c(1,277), main = "United States Long Term MLP Mul
lines(long_us_mlp_m$mean, col = "blue")
```



####US Ensemble

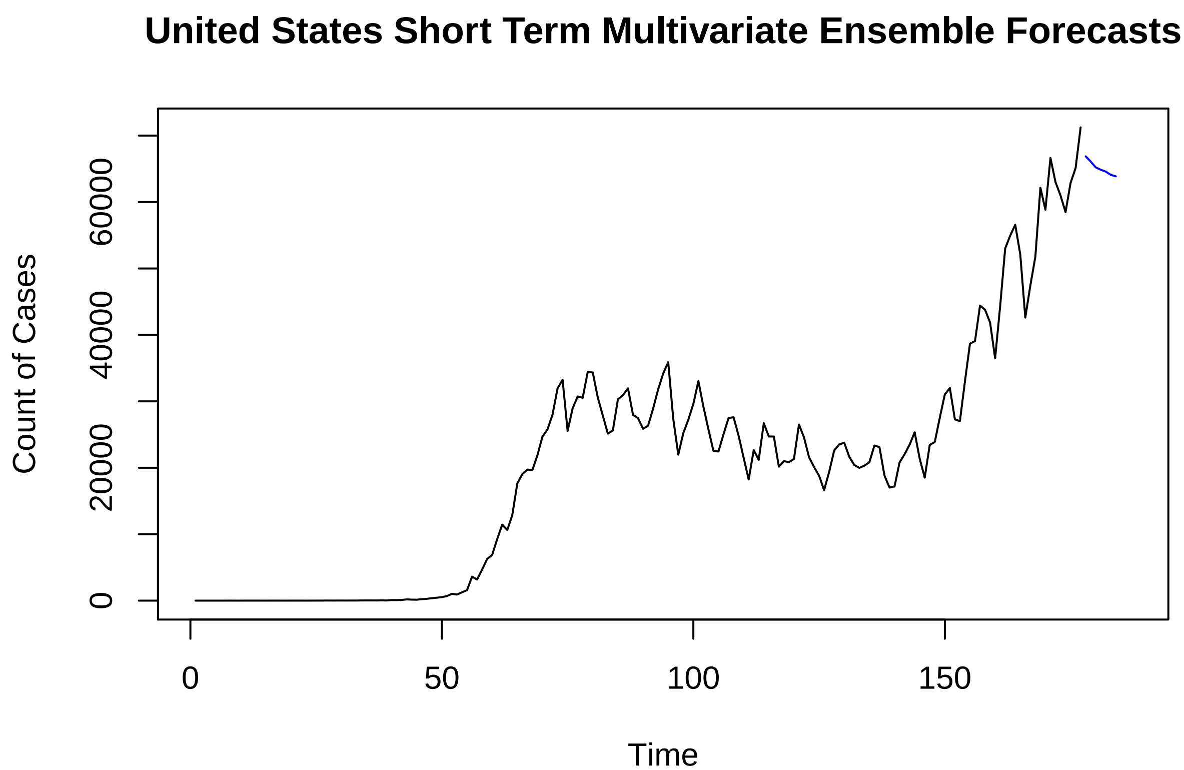
```r
ensemble_us_fore = (forecasts_us_mlp$mean + FinalPredictions_us_MLR)/2
plot(newcases_us_multi$positiveIncrease, type = "l")
lines(ensemble_us_fore, col = "blue")
```



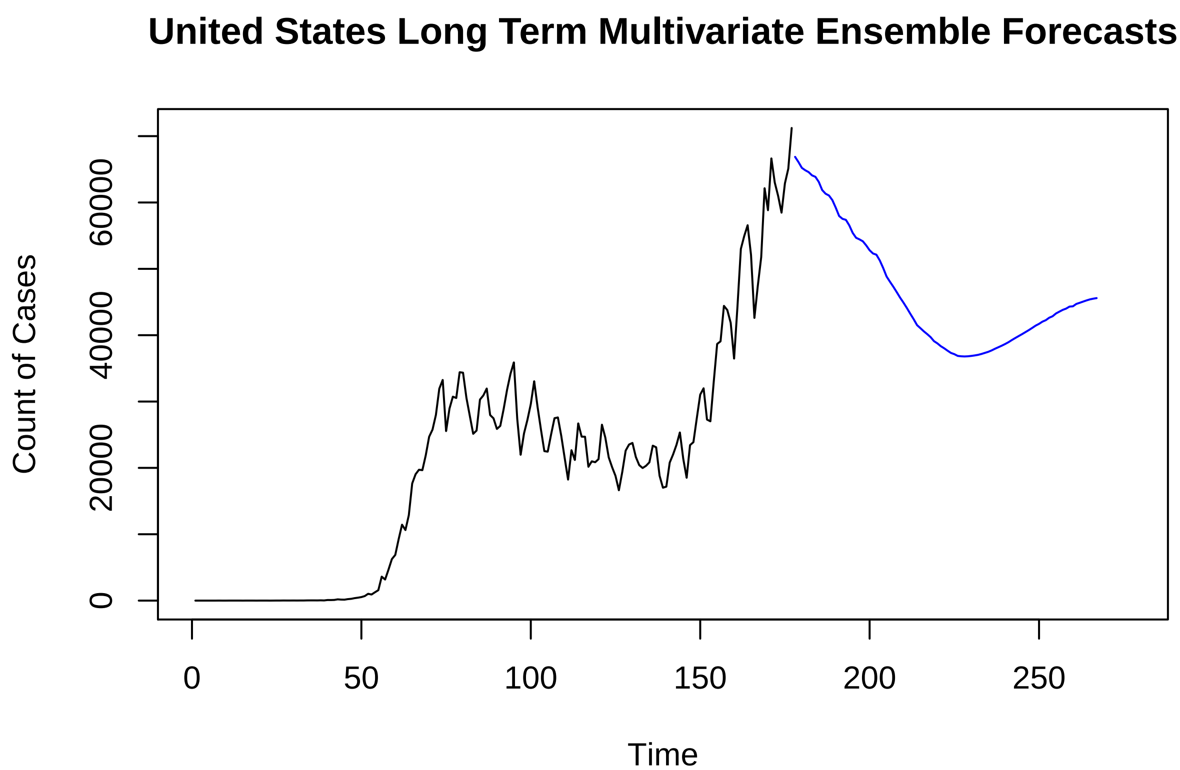

```r
#Final 12 ASE

ASE_final12 = mean((newcases_us_multi$positiveIncrease[166:177] -ensemble_us_fore)^2)
ASE_final12
```

```
## [1] 70797373
```

```r
#70596024

#Forecasting
#short_us_mlr_m
#short_us_mlp_m

#long_us_mlr_m
#long_us_mlp_m

ensemble_us_fore_short = ( short_us_mlp_m$mean+ short_us_mlr_m)/2
ensemble_us_fore_long = (long_us_mlp_m$mean + long_us_mlr_m)/2

plot(newcases_us$positiveIncrease, type = "l", xlim = c(1,187), main = "United States Short Term Multiva
lines(ensemble_us_fore_short, col = "blue")
```




```
plot(newcases_us$positiveIncrease, type = "l", xlim = c(1,277), main = "United States Long Term Multivar
lines(ensemble_us_fore_long, col = "blue")
```



##Forecasting with new data

```r
#final_us_data = read.csv()
#final_fl_data = read.csv()
```