Time Series Midterm 2020 Take Home

Chad Madding

February 26, 2020

Take-Home Portion:

Due 11:59pm CST Saturday February 29. Please Submit to 2DS in addition to Emailing it to Dr. Sadler.

Question about the realization.

1. Do you think the data come from a stationary process? Defend your thoughts using the 3 conditions of stationarity. Provide acf plots for condition 3.

```
#load the midterm data
midterm2020 = read.csv("midterm2020.csv",header = TRUE)
#Convert to a Time Series
midterm2020 = ts(midterm2020$x)
```

Description

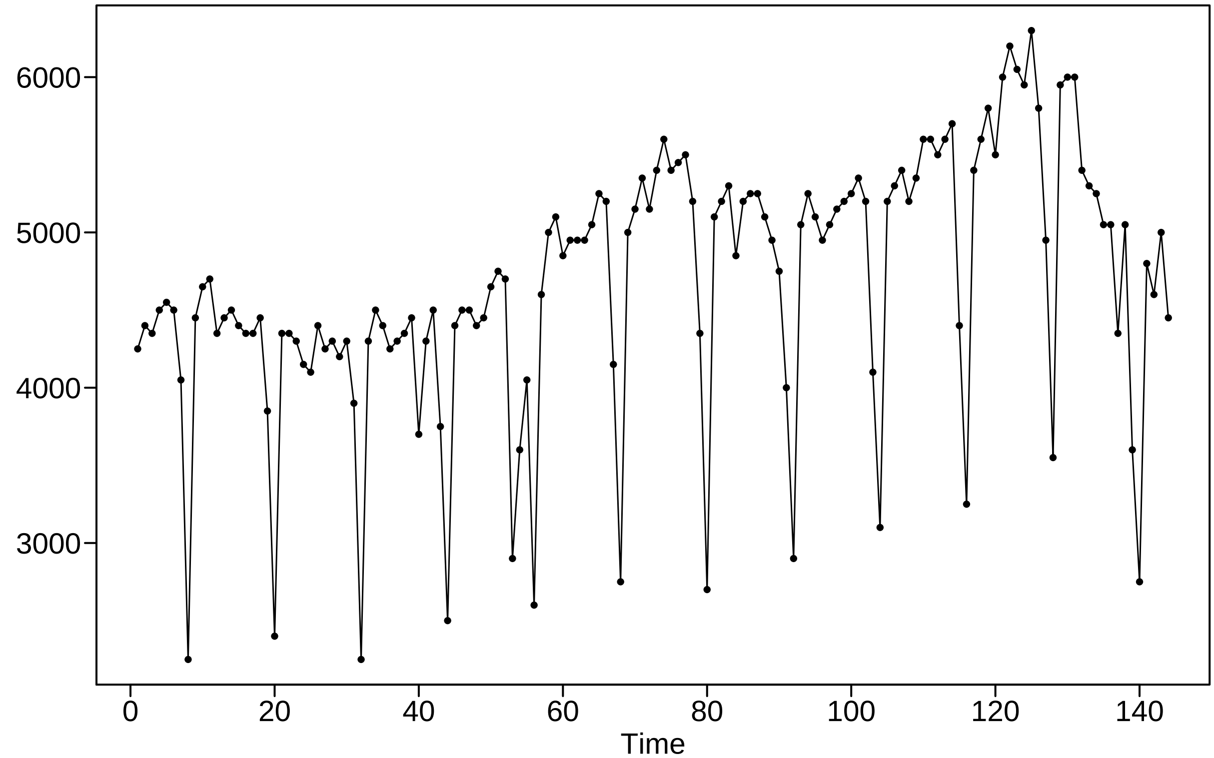
Midterm2020 is a dataset in the form of a CSV provided for the take-home portion of the Spring 2020 Midterm in MSDS 6373 Time Series.

Additional Realization

We do not know the origin of the Midterm2020 dataset other than it was provided for one portion of the midterm. Therefore, we cannot obtain additional realizations.

Condition 1 - Subpopulations of Xt have the same mean for each t. Restated, the mean does not depend on time (t).

```
#Visualize the midterm data
plotts.wge(midterm2020)
```



Realization

The Midterm2020 dataset appears to oscillate with some seasonality. The series trends up slightly, then declines during the last unit, which does appear to represent a year. It seems the series cannot be stationary because there is a level of dependency between the mean and time.

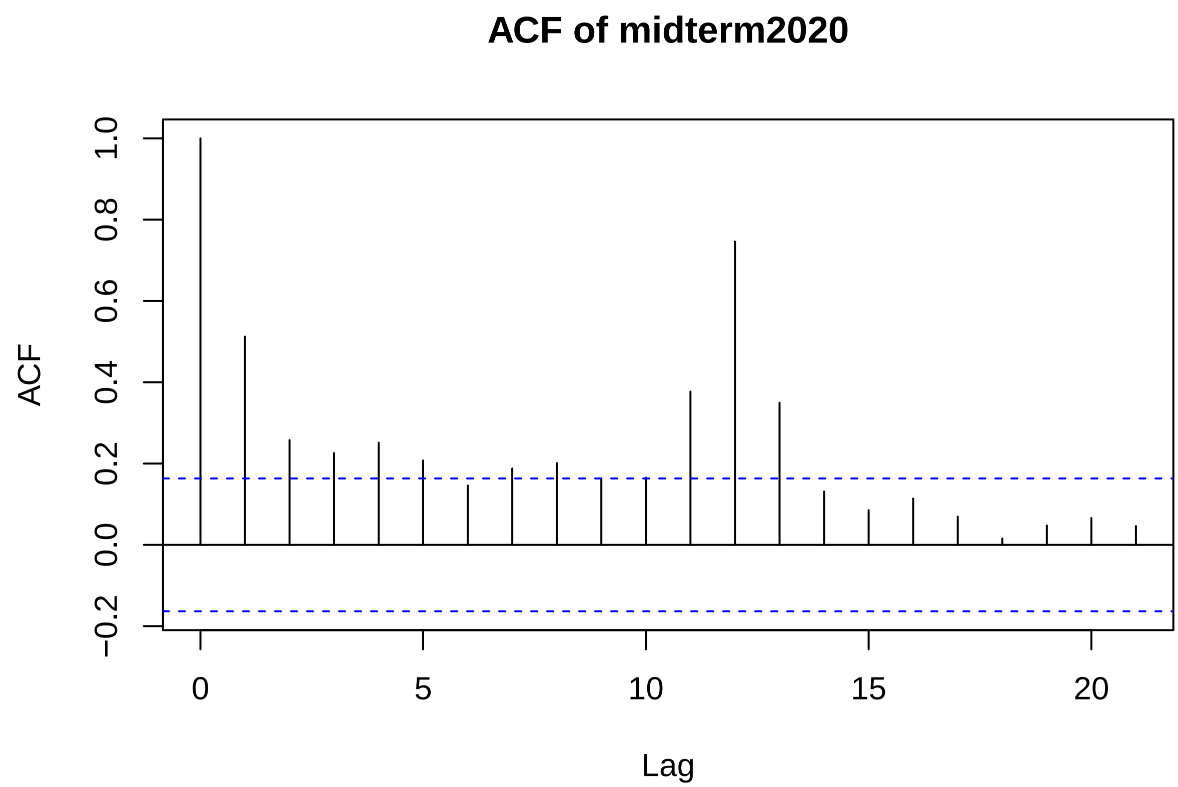
Condition 2 - Subpopulations of X for a given time have a finite and constant variance for all t. Restated, the variance does not depend on time.

We can not accurately assess the homoscedasticity of the Midterm2020 dataset since the data is dependent on time (and therefore not stationary). Nonetheless, variance seems to be greater earlier in the measurements and smaller later in the measurements.

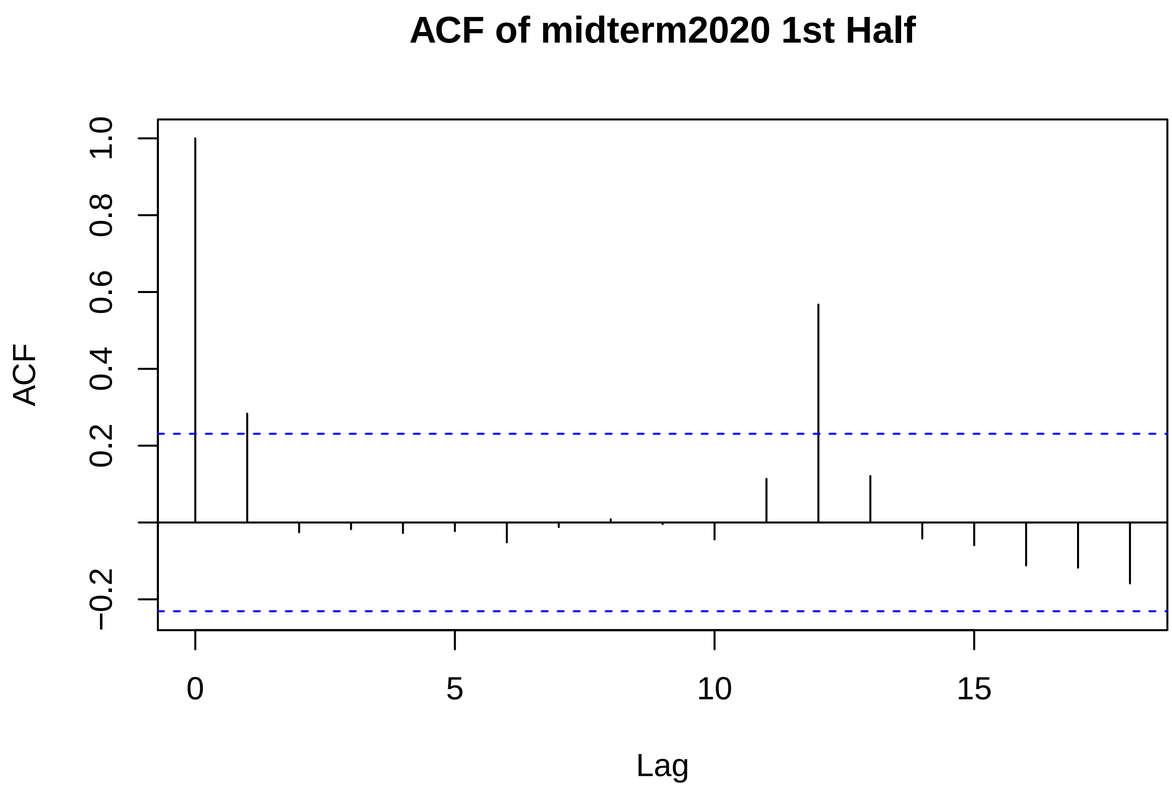
Condition 3 - The correlation between Xt_1 and Xt_2 depends only on $t_1 - t_2$. That is, the correlation between data points depends only on how far apart they are in time, not where they are in time.

Based on the first ACF chart (ACF of midterm2020), there appears to be a strong seasonal component represented in the sinusoidal degradation. Autocorrelation cycles are almost identical across similar volumes of lags, evenly spaced.

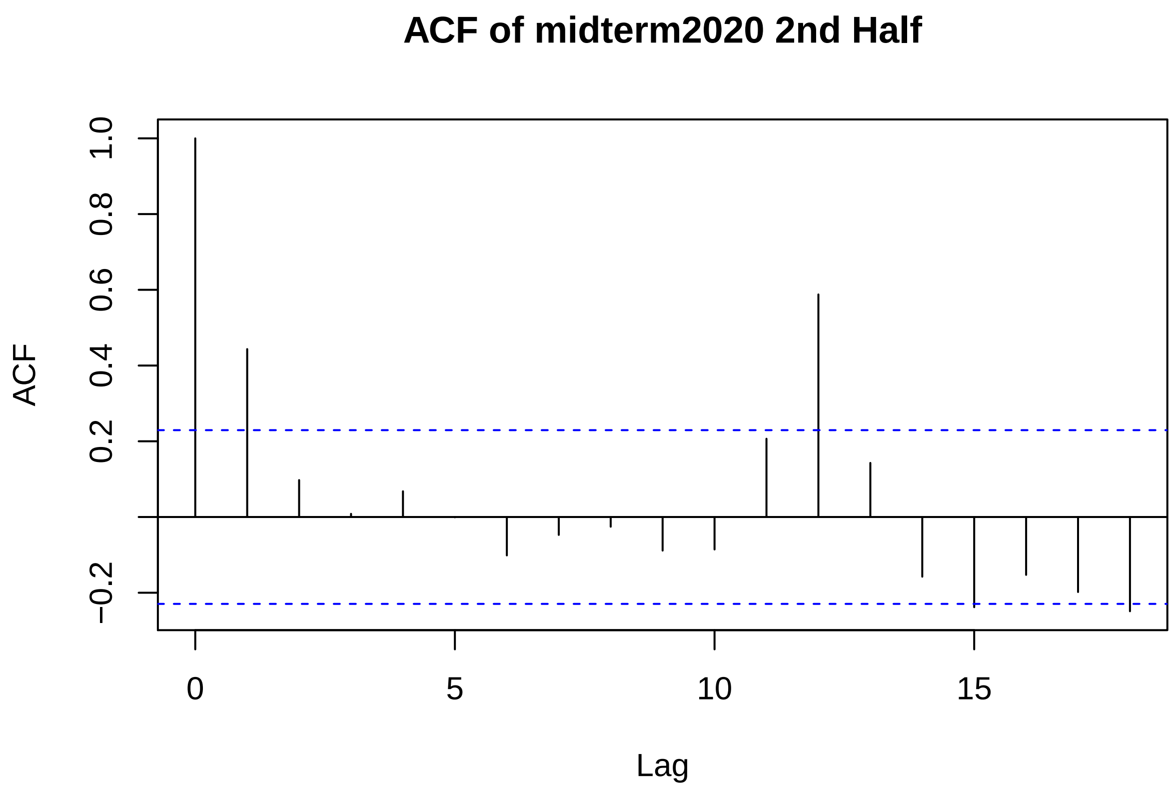
```
acf(midterm2020, main="ACF of midterm2020")
```




```r
acf(midterm2020[1:72],plot=TRUE, main="ACF of midterm2020 1st Half")
```




```r
acf(midterm2020[72:144],plot=TRUE, main="ACF of midterm2020 2nd Half")
```



In analyzing the ACFs of the first and second halves of the series, the autocorrelations seem to mirror themselves (when comparing the first half to the second half). Therefore, the data seems dependent on position in time, not just on the distance between each pair of points.

Conclusion

Because the three conditions of a stationary time series cannot be confirmed, we must conclude that this is not a stationary time series and that there is a dependency on time driving the position of each successive data point.

The Models:

Consider these two models of the data in the realization in Midterm2020.csv:

- Model 1:
$$(1 - B^{12})(1 - 0.5380B - 0.0606B^2 - 0.1923B^3)X_t = a_t$$
- Model 2:
$$(1 - 1.0507B + 0.0756B^2)X_t = (1 - 0.5927B - 0.2751B^2)a_t$$

Questions about Model 1:

2. Write this model in GLP form up to 4 terms.

```
psi.weights.wge(phi = c(0.5380, 0.0606, 0.1923), lag.max = 4)
```

```
## [1] 0.5380000 0.3500440 0.4132265 0.3469859
```

$$(1 - B^{12})Xt = at + 0.538a_{t-1} + 0.35a_{t-2} + 0.413a_{t-3} + 0.347a_{t-4}$$

Questions about Model 2:

3. Is Model 2 Invertible? Provide evidence for or against.

Model 2 is invertible, in that both absolute reciptricals are less than 1.

```
factor.wge(phi = c(1.0507, -0.0756))
```

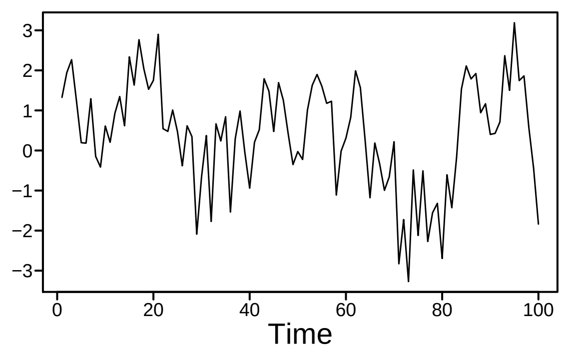
```
##
## Coefficients of Original polynomial:
## 1.0507 -0.0756
##
## Factor                 Roots                Abs Recip    System Freq
## 1-0.9730B              1.0277               0.9730       0.0000
## 1-0.0777B              12.8704               0.0777       0.0000
##
##
```

Questions for each model:

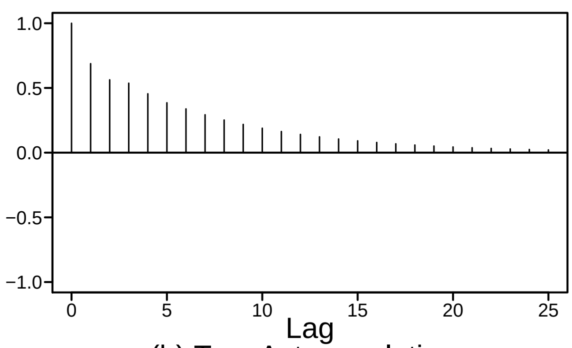
4. Provide acfs and spectral densities for each model.

Here are the ACF and spectral density for Model 1.

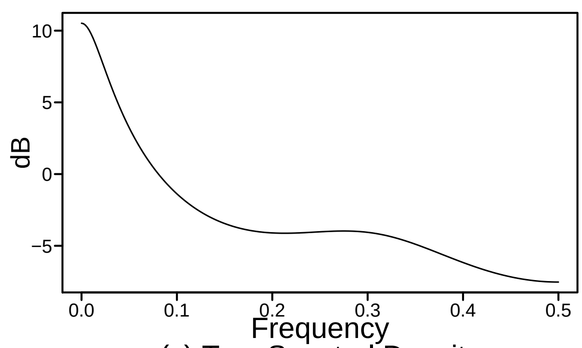
```
#Model 1
#plotts.sample.wge(x = midterm2020, phi = c(0.5380, 0.0606, 0.1923),s=12)
plotts.true.wge(phi = c(0.5380, 0.0606, 0.1923))
```



(a) Realization



(b) True Autocorrelations



(c) True Spectral Density

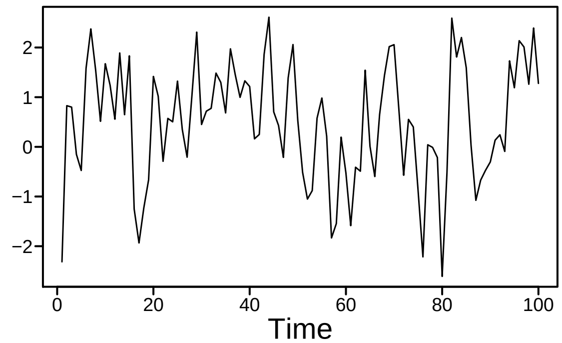
```
## $data
## Time Series:
```

```
## Start = 1
## End = 100
## Frequency = 1
##   [1]  1.32582015  1.94135390  2.26655905  1.24908855  0.19427654
##   [6]  0.18318599  1.29232914 -0.14777546 -0.41406707  0.61009566
##  [11]  0.20446199  0.93070338  1.34637161  0.61720455  2.33794333
##  [16]  1.63282438  2.76485706  2.04508760  1.52811781  1.75064132
##  [21]  2.90012812  0.54334223  0.47639304  1.01050654  0.46747803
##  [26] -0.38158051  0.61595404  0.34632118 -2.08941941 -0.65682467
##  [31]  0.37198264 -1.77129082  0.66482991  0.23831321  0.84157446
##  [36] -1.53552345  0.29038000  0.98485033 -0.04964372 -0.93987005
##  [41]  0.20602904  0.51842147  1.79099539  1.47933287  0.47497420
##  [46]  1.69438550  1.24823330  0.41994400 -0.35173060 -0.02741721
##  [51] -0.22416617  1.00650409  1.62662297  1.89874085  1.60281192
##  [56]  1.17694567  1.22891413 -1.11217019 -0.01567978  0.30910918
##  [61]  0.82743878  1.98868790  1.56704783  0.24472622 -1.18043489
##  [66]  0.18636304 -0.32753125 -0.99478855 -0.66142649  0.21842737
##  [71] -2.82840211 -1.72526004 -3.27274192 -0.48791689 -2.12143750
##  [76] -0.50803395 -2.27209179 -1.55547900 -1.31914240 -2.69713417
##  [81] -0.60691889 -1.42931437 -0.16728111  1.53323983  2.11072682
##  [86]  1.78672213  1.92083026  0.94234259  1.16672918  0.40083836
##  [91]  0.42724439  0.71288326  2.36597911  1.50126178  3.18937522
##  [96]  1.74444972  1.86266321  0.58135393 -0.44232578 -1.83781175
##
## $aut1
##  [1] 1.00000000 0.68795822 0.56301589 0.53689282 0.45526147 0.38573433
##  [7] 0.33835840 0.29295910 0.25229323 0.21855340 0.18920673 0.16355355
## [13] 0.14148555 0.12241503 0.10588466 0.09159197 0.07923350 0.06853972
## [19] 0.05928905 0.05128762 0.04436584 0.03837814 0.03319862 0.02871812
## [25] 0.02484230 0.02148957
##
## $acv
##  [1] 2.03039409 1.39682631 1.14314414 1.09010401 0.92436019 0.78319270
##  [7] 0.68700090 0.59482243 0.51225468 0.44374953 0.38416423 0.33207815
## [13] 0.28727143 0.24855075 0.21498758 0.18596779 0.16087523 0.13916263
## [19] 0.12038014 0.10413408 0.09008014 0.07792274 0.06740628 0.05830911
## [25] 0.05043966 0.04363230
##
## $spec
##   [1] 10.51711593 10.48503551 10.39022902 10.23679623 10.03094670
##   [6]  9.78024275  9.49281538  9.17669332  8.83931891  8.48726176
##  [11]  8.12609987  7.76042107  7.39389869  7.02940422  6.66913174
##  [16]  6.31471826  5.96735221  5.62786648  5.29681604  4.97454107
##  [21]  4.66121757  4.35689749  4.06154018  3.77503691  3.49722987
##  [26]  3.22792684  2.96691234  2.71395624  2.46882021  2.23126259
##  [31]  2.00104200  1.77792003  1.56166314  1.35204400  1.14884247
##  [36]  0.95184610  0.76085054  0.57565966  0.39608554  0.22194838
##  [41]  0.05307636 -0.11069467 -0.26952137 -0.42355323 -0.57293292
##  [46] -0.71779651 -0.85827387 -0.99448891 -1.12655991 -1.25459984
##  [51] -1.37871659 -1.49901326 -1.61558846 -1.72853648 -1.83794763
##  [56] -1.94390838 -2.04650162 -2.14580684 -2.24190035 -2.33485547
##  [61] -2.42474266 -2.51162974 -2.59558200 -2.67666241 -2.75493169
##  [66] -2.83044852 -2.90326961 -2.97344986 -3.04104249 -3.10609909
##  [71] -3.16866980 -3.22880337 -3.28654728 -3.34194783 -3.39505023
```

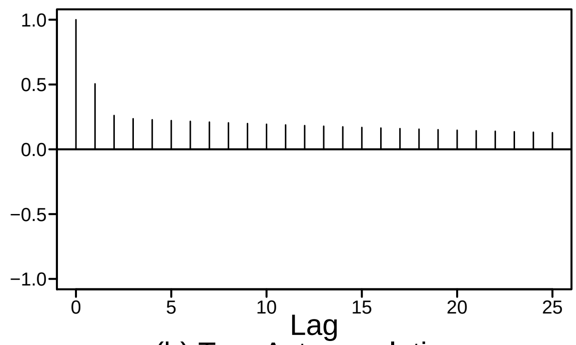
```
##  [76] -3.44589869 -3.49453653 -3.54100620 -3.58534944 -3.62760731
##  [81] -3.66782027 -3.70602827 -3.74227082 -3.77658705 -3.80901579
##  [86] -3.83959563 -3.86836501 -3.89536226 -3.92062566 -3.94419354
##  [91] -3.96610431 -3.98639652 -4.00510896 -4.02228068 -4.03795106
##  [96] -4.05215988 -4.06494736 -4.07635421 -4.08642174 -4.09519183
## [101] -4.10270703 -4.10901063 -4.11414664 -4.11815991 -4.12109611
## [106] -4.12300183 -4.12392457 -4.12391278 -4.12301592 -4.12128448
## [111] -4.11876996 -4.11552495 -4.11160309 -4.10705910 -4.10194878
## [116] -4.09632902 -4.09025773 -4.08379389 -4.07699746 -4.06992938
## [121] -4.06265151 -4.05522657 -4.04771807 -4.04019025 -4.03270796
## [126] -4.02533660 -4.01814197 -4.01119018 -4.00454750 -3.99828022
## [131] -3.99245451 -3.98713621 -3.98239073 -3.97828279 -3.97487628
## [136] -3.97223404 -3.97041763 -3.96948720 -3.96950117 -3.97051608
## [141] -3.97258639 -3.97576419 -3.98009906 -3.98563781 -3.99242434
## [146] -4.00049938 -4.00990037 -4.02066125 -4.03281233 -4.04638016
## [151] -4.06138738 -4.07785267 -4.09579061 -4.11521167 -4.13612218
## [156] -4.15852425 -4.18241588 -4.20779090 -4.23463907 -4.26294615
## [161] -4.29269399 -4.32386065 -4.35642055 -4.39034458 -4.42560031
## [166] -4.46215213 -4.49996148 -4.53898704 -4.57918492 -4.62050889
## [171] -4.66291060 -4.70633981 -4.75074459 -4.79607155 -4.84226607
## [176] -4.88927251 -4.93703439 -4.98549461 -5.03459568 -5.08427982
## [181] -5.13448922 -5.18516615 -5.23625315 -5.28769313 -5.33942955
## [186] -5.39140651 -5.44356889 -5.49586239 -5.54823372 -5.60063057
## [191] -5.65300177 -5.70529727 -5.75746828 -5.80946723 -5.86124784
## [196] -5.91276517 -5.96397560 -6.01483686 -6.06530805 -6.11534960
## [201] -6.16492334 -6.21399242 -6.26252133 -6.31047589 -6.35782322
## [206] -6.40453174 -6.45057112 -6.49591225 -6.54052725 -6.58438941
## [211] -6.62747319 -6.66975413 -6.71120890 -6.75181519 -6.79155176
## [216] -6.83039831 -6.86833554 -6.90534506 -6.94140940 -6.97651192
## [221] -7.01063684 -7.04376919 -7.07589477 -7.10700013 -7.13707253
## [226] -7.16609994 -7.19407100 -7.22097497 -7.24680176 -7.27154186
## [231] -7.29518634 -7.31772682 -7.33915545 -7.35946490 -7.37864835
## [236] -7.39669943 -7.41361227 -7.42938141 -7.44400186 -7.45746903
## [241] -7.46977875 -7.48092724 -7.49091112 -7.49972739 -7.50737342
## [246] -7.51384693 -7.51914602 -7.52326914 -7.52621509 -7.52798299
## [251] -7.52857235
```

Here are the ACF and spectral density for Model 2.

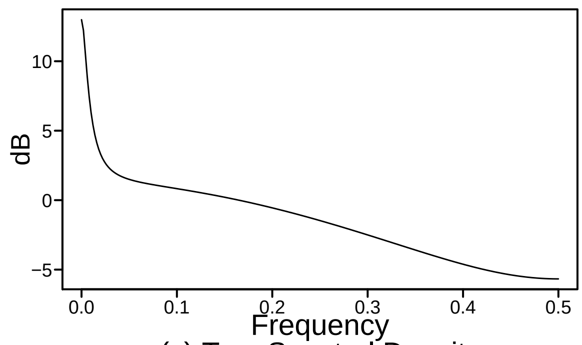
```r
#Model 2
plotts.true.wge(phi = c(1.0507,-0.0756), theta = c(0.5927, 0.2751))
```



(a) Realization



(b) True Autocorrelations



(c) True Spectral Density

```
## $data
## Time Series:
## Start = 1
## End = 100
## Frequency = 1
##   [1] -2.31499195  0.82864722  0.79995063 -0.15086354 -0.47576425
##   [6]  1.58109053  2.37422795  1.54157289  0.51445029  1.67343966
##  [11]  1.23937954  0.55684367  1.88998679  0.64785836  1.83162487
##  [16] -1.25186797 -1.93455223 -1.23239558 -0.66254500  1.41699604
##  [21]  1.01742792 -0.29058832  0.57244160  0.50406341  1.32302315
##  [26]  0.35032622 -0.20784030  1.03468078  2.30973034  0.45005902
##  [31]  0.71823399  0.77558686  1.48468309  1.29756303  0.68317008
##  [36]  1.97277363  1.45407204  0.99641391  1.32933138  1.21451329
##  [41]  0.16150477  0.24955906  1.86127091  2.61004795  0.70346805
##  [46]  0.42846860 -0.21174682  1.38833015  2.05935476  0.52981503
##  [51] -0.51347181 -1.05162781 -0.88203483  0.57725561  0.98300619
##  [56]  0.21386154 -1.83379159 -1.54582770  0.19493221 -0.52977129
##  [61] -1.58613349 -0.41066080 -0.49037984  1.54268258  0.01177153
##  [66] -0.59783171  0.65212667  1.43443316  2.01725251  2.05718912
##  [71]  0.75276159 -0.57169910  0.55257859  0.39957691 -0.88782401
##  [76] -2.21531389  0.04148955 -0.01036405 -0.21486710 -2.60862896
##  [81] -0.51051895  2.59140093  1.80887521  2.20101910  1.59196101
##  [86]  0.03122317 -1.07690229 -0.67017866 -0.47380813 -0.30380852
##  [91]  0.13394000  0.24320489 -0.09232447  1.73088321  1.18984892
##  [96]  2.13730674  2.01067577  1.26100705  2.39225886  1.27721298
##
```

```
## $aut1
##  [1] 1.0000000 0.5051247 0.2609183 0.2359594 0.2281971 0.2219282 0.2159283
##  [8] 0.2100981 0.2044259 0.1989068 0.1935368 0.1883118 0.1832278 0.1782811
## [15] 0.1734679 0.1687847 0.1642279 0.1597941 0.1554801 0.1512825 0.1471982
## [22] 0.1432242 0.1393575 0.1355951 0.1319344 0.1283725
##
## $acv
##  [1] 1.4164624 0.7154902 0.3695810 0.3342277 0.3232327 0.3143530 0.3058543
##  [8] 0.2975960 0.2895615 0.2817440 0.2741376 0.2667365 0.2595353 0.2525284
## [15] 0.2457108 0.2390771 0.2326226 0.2263424 0.2202316 0.2142859 0.2085007
## [22] 0.2028717 0.1973946 0.1920654 0.1868801 0.1818348
##
## $spec
##   [1] 12.98859173 12.21742294 10.56639098  8.87763016  7.44795386
##   [6]  6.30353392  5.40081444  4.68883562  4.12406739  3.67232814
##  [11]  3.30754894  3.01003787  2.76493417  2.56098476  2.38961900
##  [16]  2.24426277  2.11983394  2.01237184  1.91876447  1.83654689
##  [21]  1.76375126  1.69879462  1.64039429  1.58750362  1.53926291
##  [26]  1.49496148  1.45400832  1.41590917  1.38024838  1.34667467
##  [31]  1.31488967  1.28463882  1.25570397  1.22789741  1.20105699
##  [36]  1.17504209  1.14973038  1.12501506  1.10080260  1.07701084
##  [41]  1.05356741  1.03040841  1.00747723  0.98472366  0.96210301
##  [46]  0.93957547  0.91710546  0.89466116  0.87221405  0.84973850
##  [51]  0.82721150  0.80461232  0.78192228  0.75912450  0.73620376
##  [56]  0.71314629  0.68993964  0.66657252  0.64303473  0.61931700
##  [61]  0.59541098  0.57130905  0.54700434  0.52249063  0.49776227
##  [66]  0.47281419  0.44764177  0.42224087  0.39660777  0.37073910
##  [71]  0.34463188  0.31828342  0.29169134  0.26485354  0.23776816
##  [76]  0.21043360  0.18284844  0.15501151  0.12692180  0.09857847
##  [81]  0.06998086  0.04112847  0.01202092 -0.01734201 -0.04696043
##  [86] -0.07683432 -0.10696355 -0.13734790 -0.16798701 -0.19888046
##  [91] -0.23002772 -0.26142819 -0.29308116 -0.32498587 -0.35714147
##  [96] -0.38954703 -0.42220156 -0.45510399 -0.48825320 -0.52164799
## [101] -0.55528711 -0.58916924 -0.62329301 -0.65765697 -0.69225963
## [106] -0.72709945 -0.76217480 -0.79748404 -0.83302543 -0.86879720
## [111] -0.90479751 -0.94102449 -0.97747618 -1.01415059 -1.05104565
## [116] -1.08815926 -1.12548923 -1.16303335 -1.20078931 -1.23875477
## [121] -1.27692732 -1.31530447 -1.35388370 -1.39266240 -1.43163789
## [126] -1.47080744 -1.51016824 -1.54971742 -1.58945202 -1.62936902
## [131] -1.66946531 -1.70973771 -1.75018297 -1.79079774 -1.83157859
## [136] -1.87252200 -1.91362437 -1.95488201 -1.99629113 -2.03784783
## [141] -2.07954813 -2.12138795 -2.16336309 -2.20546924 -2.24770200
## [146] -2.29005685 -2.33252914 -2.37511411 -2.41780687 -2.46060243
## [151] -2.50349565 -2.54648124 -2.58955381 -2.63270782 -2.67593758
## [156] -2.71923725 -2.76260086 -2.80602228 -2.84949521 -2.89301321
## [161] -2.93656968 -2.98015783 -3.02377072 -3.06740123 -3.11104208
## [166] -3.15468577 -3.19832467 -3.24195093 -3.28555651 -3.32913320
## [171] -3.37267258 -3.41616603 -3.45960474 -3.50297969 -3.54628165
## [176] -3.58950119 -3.63262867 -3.67565423 -3.71856781 -3.76135913
## [181] -3.80401768 -3.84653276 -3.88889342 -3.93108852 -3.97310669
## [186] -4.01493634 -4.05656566 -4.09798264 -4.13917504 -4.18013041
## [191] -4.22083609 -4.26127922 -4.30144674 -4.34132536 -4.38090163
## [196] -4.42016188 -4.45909228 -4.49767881 -4.53590727 -4.57376330
## [201] -4.61123239 -4.64829987 -4.68495094 -4.72117067 -4.75694402
```

```
## [206] -4.79225581 -4.82709081 -4.86143367 -4.89526900 -4.92858135
## [211] -4.96135523 -4.99357510 -5.02522547 -5.05629081 -5.08675564
## [216] -5.11660453 -5.14582211 -5.17439310 -5.20230231 -5.22953470
## [221] -5.25607536 -5.28190956 -5.30702275 -5.33140061 -5.35502903
## [226] -5.37789417 -5.39998249 -5.42128072 -5.44177595 -5.46145558
## [231] -5.48030741 -5.49831963 -5.51548084 -5.53178008 -5.54720684
## [236] -5.56175111 -5.57540334 -5.58815454 -5.59999623 -5.61092050
## [241] -5.62091998 -5.62998792 -5.63811814 -5.64530511 -5.65154388
## [246] -5.65683016 -5.66116030 -5.66453131 -5.66694085 -5.66838725
## [251] -5.66886949
```

5. Provide a factor table for each model.

Here is the factor table for Model 1.

```r
#Model 1 factor table
factor.wge(phi = c(0.5380, 0.0606, 0.1923))
```

```
##
## Coefficients of Original polynomial:
## 0.5380 0.0606 0.1923
##
## Factor                 Roots                Abs Recip    System Freq
## 1-0.8650B              1.1560               0.8650       0.0000
## 1+0.3270B+0.2223B^2   -0.7356+-1.9893i      0.4715       0.3064
##
##
```

Here is the factor table for Model 2.

```r
#Model 2 factor table
factor.wge(phi = c(1.0507,-0.0756))
```

```
##
## Coefficients of Original polynomial:
## 1.0507 -0.0756
##
## Factor                 Roots                Abs Recip    System Freq
## 1-0.9730B              1.0277               0.9730       0.0000
## 1-0.0777B              12.8704               0.0777       0.0000
##
##
```

6. Calculate the ASE for the last 12 months of the data set. (This will be only 1 ASE per model.).

```r
#Get the length of the dataset
lengthMT2020=length(midterm2020)
#Model 1 Forcast
model1Q6f = fcst(aruma, midterm2020, s = 12, phi = c(0.5380, 0.0606, 0.1923), n.ahead = 12, lastn = T, p
#Model 1 ASE
model1Q6_ase = ase(midterm2020, model1Q6f)

#Model 2 Forcast
model2Q6f = fcst(arma, midterm2020, phi = c(1.0507,-0.0756), theta = c(0.5927,0.2751), n.ahead = 12, las
#Model 2 ASE
model2Q6_ase = ase(midterm2020, model2Q6f)
```

The ASE for Model 1 for the last 12 months of the data set: 1301829

```
model1Q6_ase
```

```
## [1] 1301829
```

The ASE for Model 2 for the last 12 months of the data set: 836289

```
model2Q6_ase
```

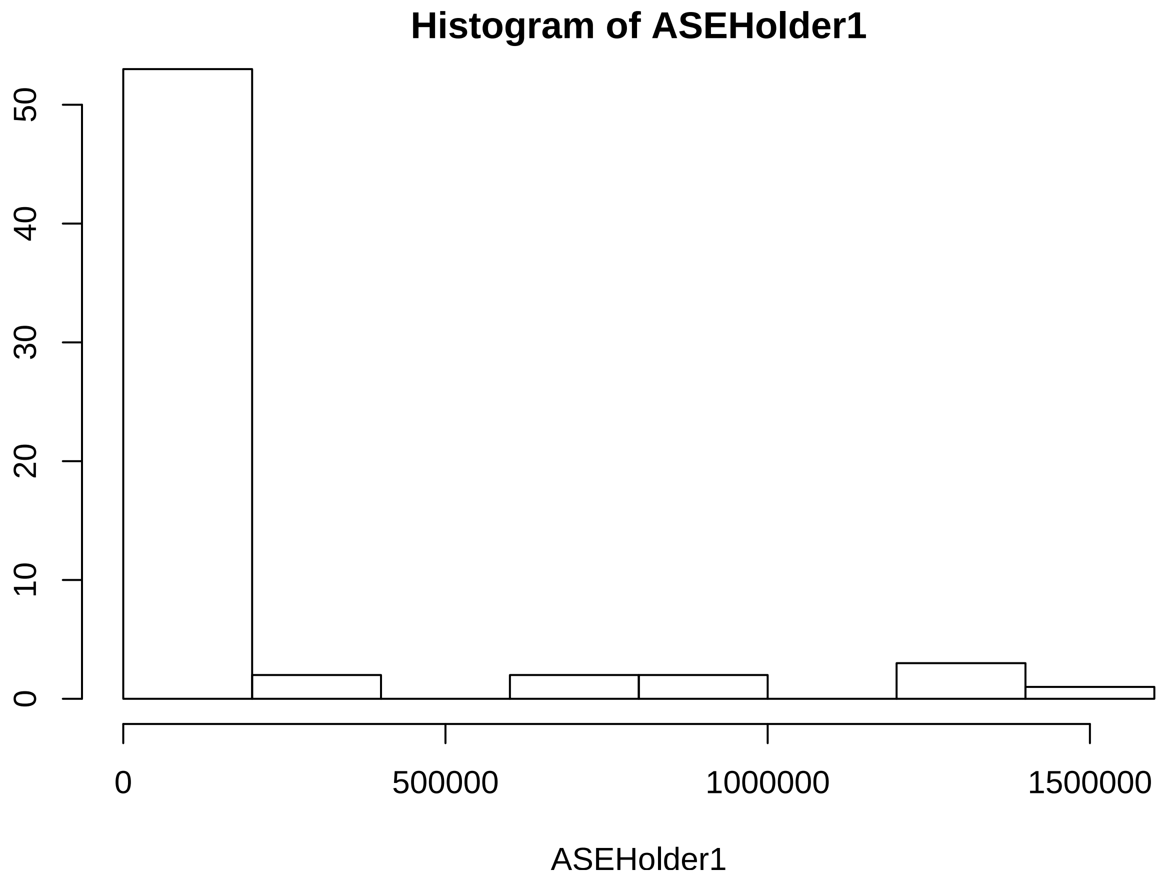
```
## [1] 836289.1
```

7. Calculate at least 10 ASEs across the data set and find their average (the rolling window ASE).

```
#Model 1
phis1 = c(0.5380, 0.0606, 0.1923)
thetas1 = 0
s1  = 12
d1  = 0

trainingSize = 70
horizon = 12
ASEHolder1 = numeric()
dataLength=length(midterm2020)
i=0
for( i in 1:(dataLength-(trainingSize + horizon) + 1))
{

  forecasts1 = fore.aruma.wge(midterm2020[i:(i+(trainingSize-1))],phi = phis1, theta = thetas1, s = s1,

  ASE1 = mean((midterm2020[(trainingSize+i):(trainingSize+ i + (horizon) - 1)] - forecasts1$f)^2)

  ASEHolder1[i] = ASE1

}

ASEHolder1
```

```
##  [1]   62151.67   49493.80   51547.38   39834.38   57418.98   63021.43
##  [7]   81029.90   96712.73   90059.87  112142.56   91342.26  103495.29
## [13]   98359.05   91144.05   56163.90   53471.08   39212.14   44274.56
## [19]   36140.13   57318.58   87993.35   91163.85   44907.78   48249.58
## [25]   45475.88   62613.31   52417.91   74309.88   90632.58  105686.83
## [31]   78196.15   42693.73   48163.76   60902.04   58609.89   65562.07
## [37]   79966.91   59950.80   59143.63   73644.86   75917.25   81205.85
## [43]   98479.48  133801.64  101860.29  125121.80  141870.24  163328.35
## [49]  150228.36  117176.41  114385.08  105394.04  186167.90  283486.74
## [55]  363189.07  737157.60  733698.95  958978.04  984575.34 1218370.87
## [61] 1388842.84 1423357.87 1301829.30
```

```
hist(ASEHolder1)
```



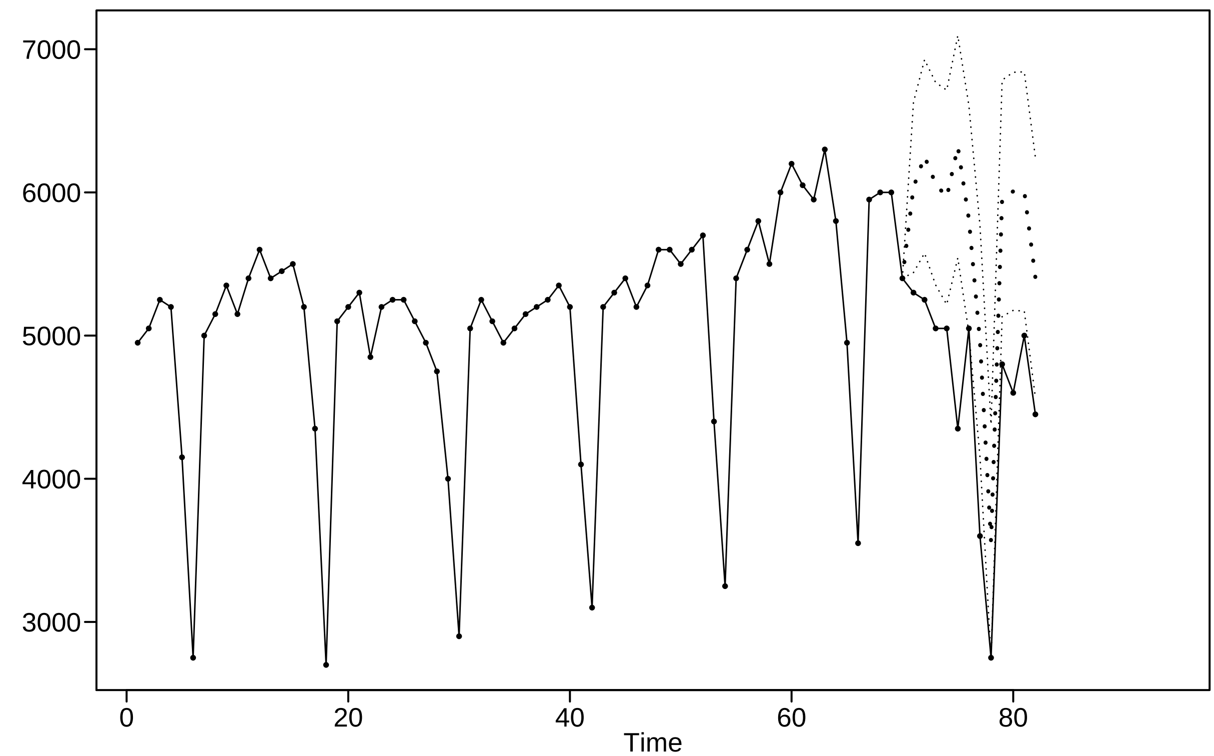

```
summary(ASEHolder1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   36140   58014   87993  217350  121149 1423358
```

```
WindowedASE1 = mean(ASEHolder1)
WindowedASE1
```

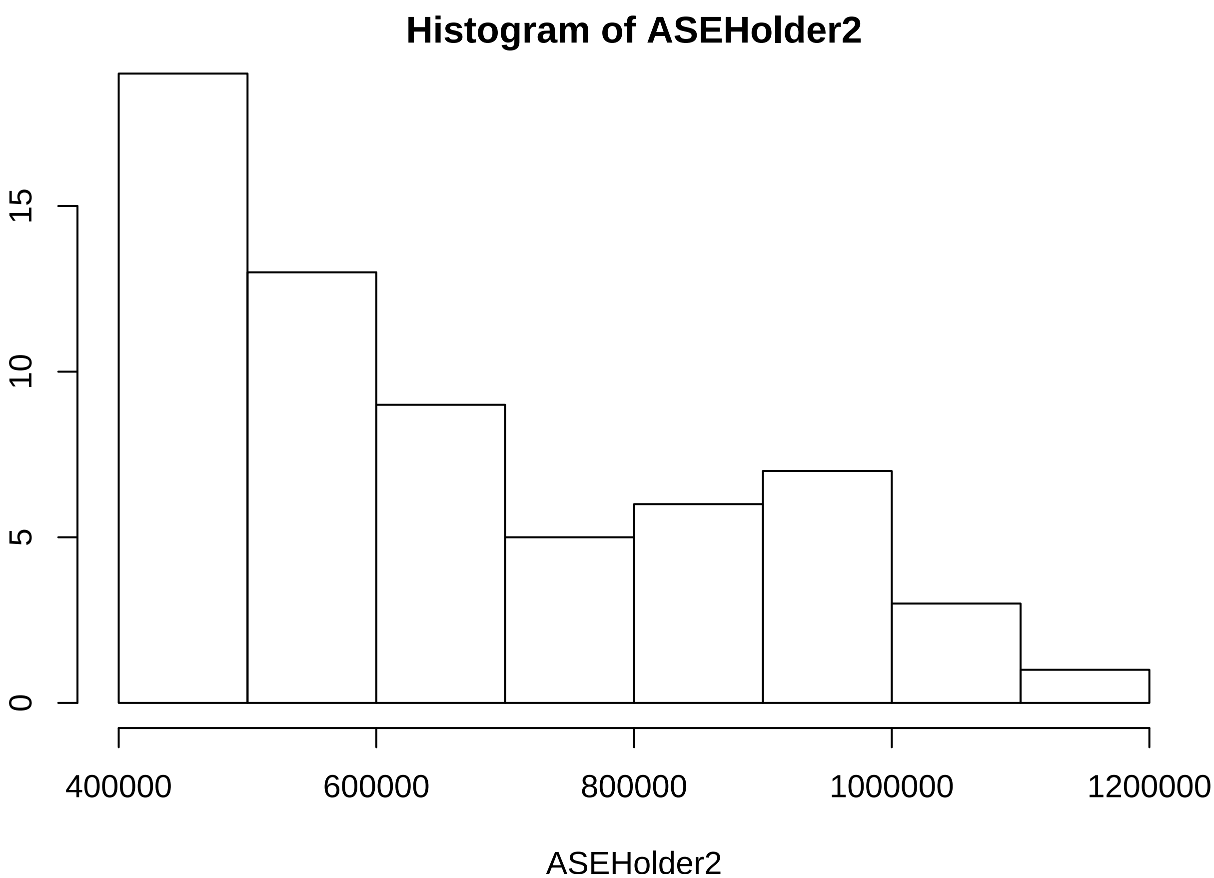
```
## [1] 217349.9
```

```
# Visualization

i = length(ASEHolder1)
fs1 = fore.aruma.wge(midterm2020[i:(i+(trainingSize+horizon)-1)],phi = phis1, theta = thetas1, s = s1, 
```




```
ASE1 = mean((midterm2020[(i+trainingSize):(i+(trainingSize+horizon)-1)] - fs1$f )^2)
```

```
#Model 2
phis2 = c(1.0507,-0.0756)
thetas2 = c(0.5927, 0.2751)
s2  = 0
d2  = 0

ASEHolder2 = numeric()
i=0
for( i in 1:(dataLength-(trainingSize + horizon) + 1))
{

  forecasts2 = fore.arma.wge(midterm2020[i:(i+(trainingSize-1))],phi = phis2, theta = thetas2, n.ahead =

  ASE2 = mean((midterm2020[(trainingSize+i):(trainingSize+ i + (horizon) - 1)] - forecasts2$f)^2)

  ASEHolder2[i] = ASE2

}

#ASEHolder2 = ASEHolder
hist(ASEHolder2)
```



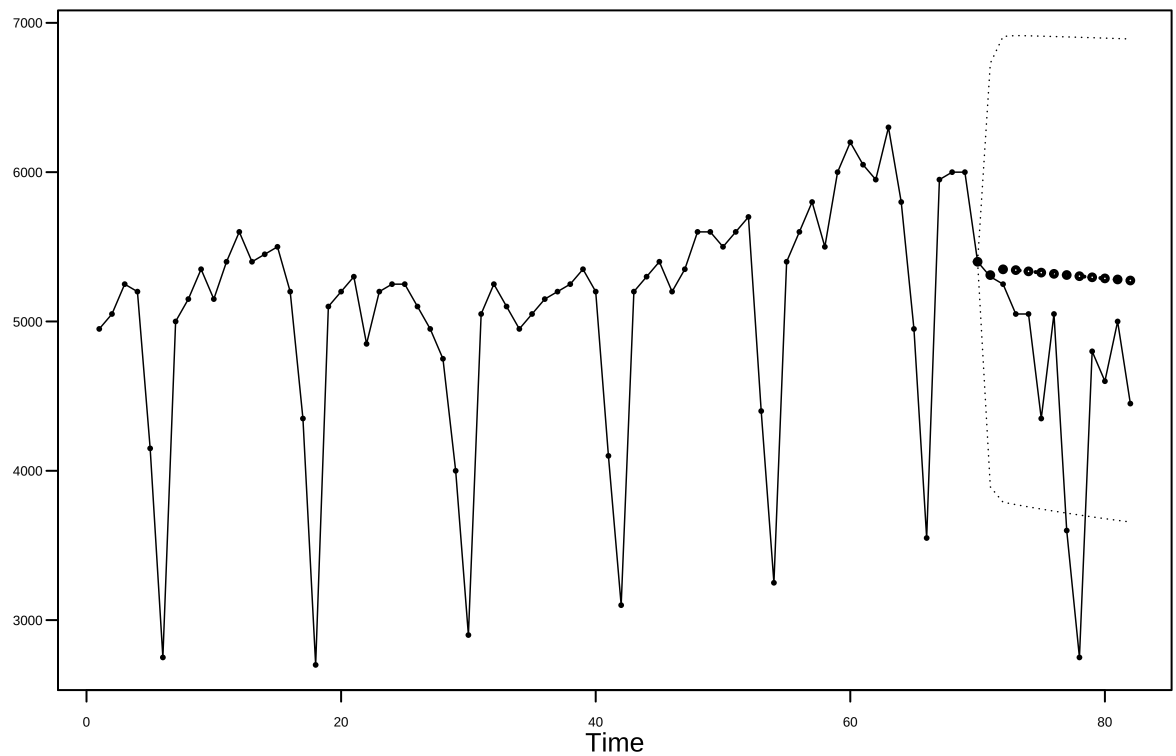

```r
WindowedASE2 = mean(ASEHolder2)

summary(ASEHolder2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  412399  485372  588553  654834  814259 1126142
```

```r
WindowedASE2
```

```
## [1] 654834.2
```

```r
# Visualization

i = length(ASEHolder2)
fs2 = fore.arma.wge(midterm2020[i:(i+(trainingSize+horizon)-1)],phi = phis2, theta = thetas2, n.ahead =
```




```
ASE2 = mean((midterm2020[(i+trainingSize):(i+(trainingSize+horizon)-1)] - fs2$f )^2)
```

8. Compare the single ASE to the rolling window ASE. Are they roughly the same, is one significantly larger? Does it provide evidence as to which model is more useful?

Now, let's compare the single ASE from question 6 to the "windowed" results in question 7. For Model one, we have the original ASE at 1301829 and the windowed at 217350. The original ASE from the second model is 836289 and the windowed model 2 ASE is 654834. Both models ASE's improved significantly, but when the original ASE's were calculated, model 2 had the better ASE. After applying a rolling window to both models, the first model outperformed the second one. Applying an average to ASE's taken in windowed sections provide evidence that model one is more useful.

```
#Original model 1 ase
model1Q6_ase
```

```
## [1] 1301829
```

```
#Windowed model 1 ase
WindowedASE1
```

```
## [1] 217349.9
```

```
#Original model 2 ase
model2Q6_ase
```

```
## [1] 836289.1
```

```
#Windowed model 2 ase
WindowedASE2
```

```
## [1] 654834.2
```

Final Question:

9. Given your analysis, which model do you feel is more useful in making 12-month forecasts?

Given all the information above, we feel model one will outperform the second one over time. The ACF obtained on the first model using the rolling window method proved to be better than the second model.

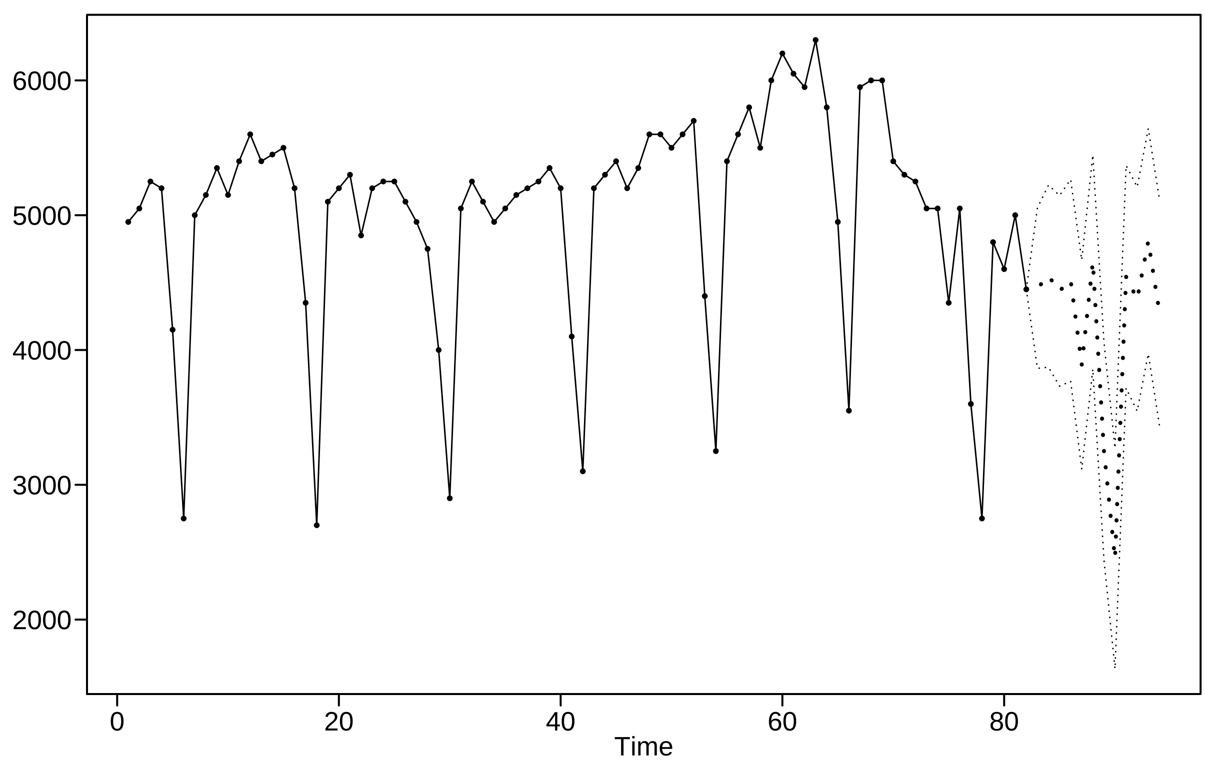
BONUS (up to 3 points):

Create an interesting, descriptive and useful plot to visualize the forecasts that the rolling window ASE was based on. This would help the analyst diagnose why the ASE is large or small and/or where it is fitting relatively well and relatively poorly. In addition, it may add confidence to the client that the model is performing adequately.

To help visualize how much better model one could perform over model two, we have provided a few plots of both models. Each chart shows a forecast for the coming year.

With the results collected from the rolling window ASE model, one will outperform the second model. The Forecast Model One plot below shows a more natural trend seen over the past several years.

Forecast Model One




```
## $f
##  [1] 4459.080 4547.715 4438.526 4516.760 3891.012 4653.164 3256.145
##  [8] 2452.694 4542.900 4377.540 4807.565 4283.548
```

```
##
## $ll
##  [1] 3864.068 3872.057 3731.493 3768.194 3114.497 3859.097 2448.199
##  [8] 1634.358 3717.084 3546.138 3971.982 3444.863
##
## $ul
##  [1] 5054.092 5223.374 5145.559 5265.325 4667.527 5447.230 4064.091
##  [8] 3271.031 5368.716 5208.943 5643.147 5122.233
##
## $resid
##  [1]     0.000     0.000     0.000     0.000     0.000     0.000     0.000
##  [8]     0.000     0.000     0.000     0.000     0.000     0.000     0.000
## [15]     0.000  -245.275   107.930  -205.675   114.780   -39.230   -73.345
## [22]  -295.360   -45.185  -214.605   108.110  -209.630  -285.305  -104.045
## [29]    -7.265   521.335   -49.855   132.085  -262.330   214.185  -201.295
## [36]    13.100    -6.340   211.805   341.560   235.325  -195.185    42.010
## [43]   -50.195   -62.050   225.550    56.725   137.705   215.760    91.645
## [50]   -50.160     4.725   273.430   -32.225   -89.775     4.970   125.620
## [57]   197.635    28.160   406.670   155.200    30.120    46.545   315.250
## [64]  -390.405   367.245  -136.570   336.040   -19.845  -106.220  -337.605
## [71]  -735.240  -605.800  -427.250  -169.820 -1222.515   545.940  -655.260
## [78]   346.735  -493.565  -473.215   -23.270  -106.015
##
## $wnv
## [1] 92159.46
##
## $se
##  [1] 303.5778 344.7237 360.7310 381.9210 396.1812 405.1358 412.2174
##  [8] 417.5184 421.3347 424.1848 426.3178 427.9005
##
## $psi
##  [1] 0.5380000 0.3500440 0.4132265 0.3469859 0.2790334 0.2506108 0.2184634
##  [8] 0.1863784 0.1617029 0.1403012 0.1211218 1.1047613
##
## $ptot
## [1] 15
##
## $phitot
##  [1]  0.5380  0.0606  0.1923  0.0000  0.0000  0.0000  0.0000  0.0000
##  [9]  0.0000  0.0000  0.0000  1.0000 -0.5380 -0.0606 -0.1923
```

Model two is trending to the mean. This type of data has too much seasonality in it for model two, and the ASE seemed to point that out.

Forecast Model Two