STAD57 A3

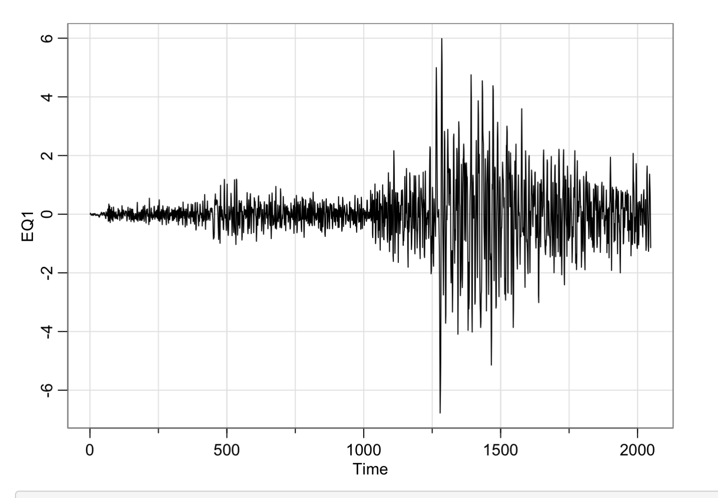
2022-12-02

```
library(astsa)
attach(eqexp)
```

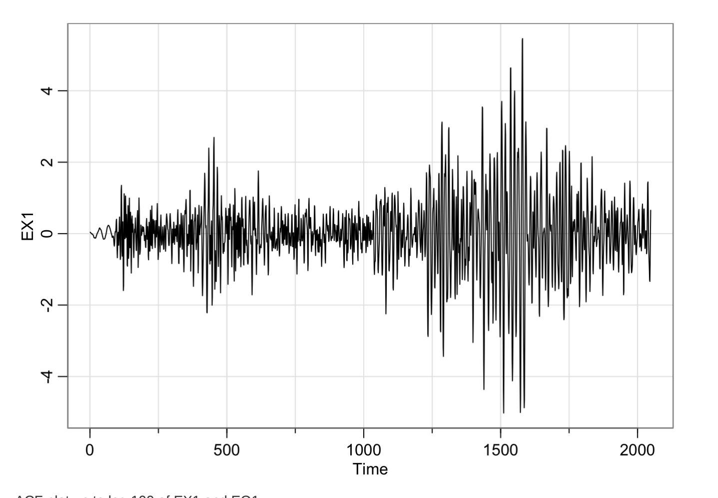
```
## The following object is masked from package:astsa:
##
##     EQ5
```

Q1 a) Time Series plot of EX1 and EQ1

```
tsplot(EQ1)
```

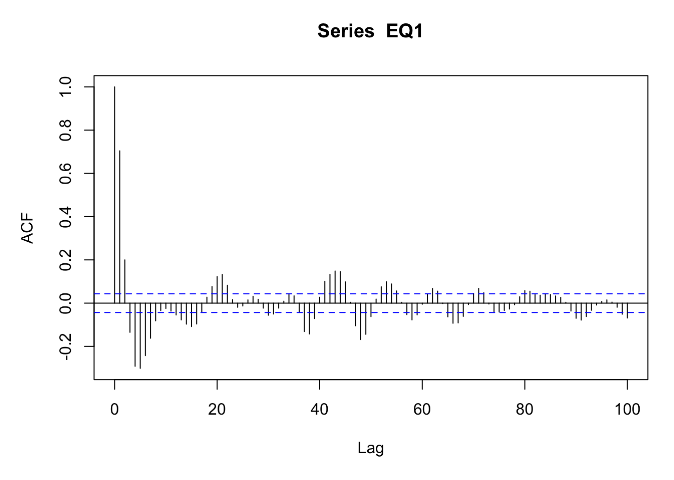


```
tsplot(EX1)
```

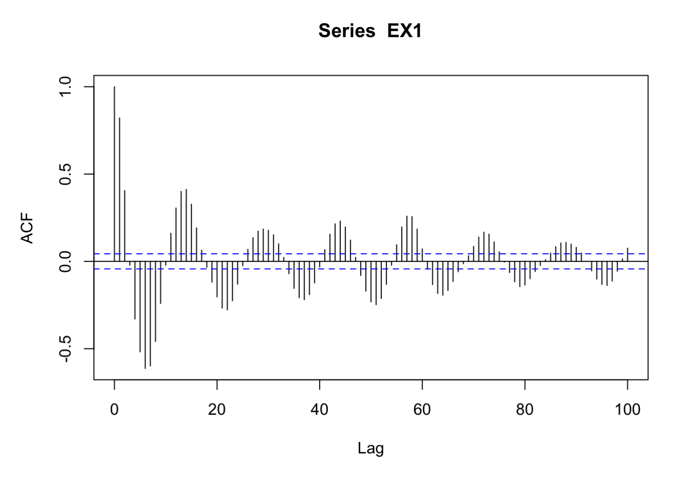


ACF plot up to lag 100 of EX1 and EQ1

```
acf(EQ1, lag.max = 100)
```

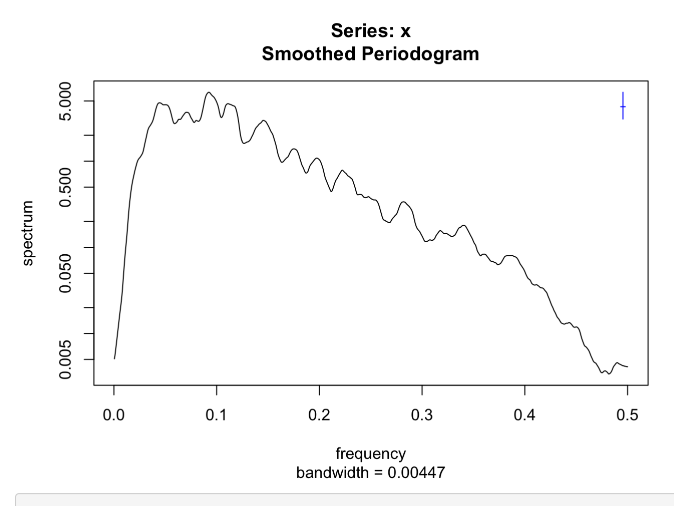



```
acf(EX1, lag.max = 100)
```

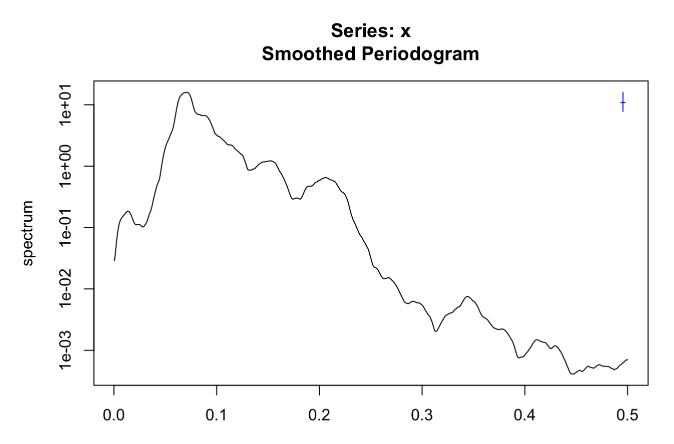


Spectral density estimation of EX1 and EQ1

```
spectrum(EQ1, spans = c(30,10) )
```




```
spectrum(EX1, spans = c(30,10) )
```




```
#Earthquake signal decreases in the same pattern through the frequency
#Explosion signal decrease rapidly when frequency is between 0.1 to 0.3
```

Q1 b)

```
# Combine all series
EXEQ = cbind( EX1, EX2, EX3, EX4, EX5, EX6, EX7, EX8,
             EQ1, EQ2, EQ3, EQ4, EQ5, EQ6, EQ7, EQ8 )

# create matrix to hold spectral density values
spectra = matrix(0, nrow = 16, ncol = 30)

# estimate spectra at 30 frequencies using AR(30) model for all series
for(i in 1:16 ){
  spectra[i,] = arima.sim(list(EXEQ[,i]),n = 30)
}
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.3.6      ✔ purrr   0.3.4
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.1      ✔ stringr 1.4.0
## ✔ readr   2.1.2      ✔ forcats 0.5.2
## ── Conflicts ────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
# create data frame with spectra and series type
spectra = tibble(spectra) %>%
  mutate( type = colnames(EXEQ) %>% str_sub(1,2) )

# Perform Linear Discriminant Analysis and plot histograms of the most
# discriminating linear combinations of the spectra for Explosions and Earthquakes
library(MASS)
```

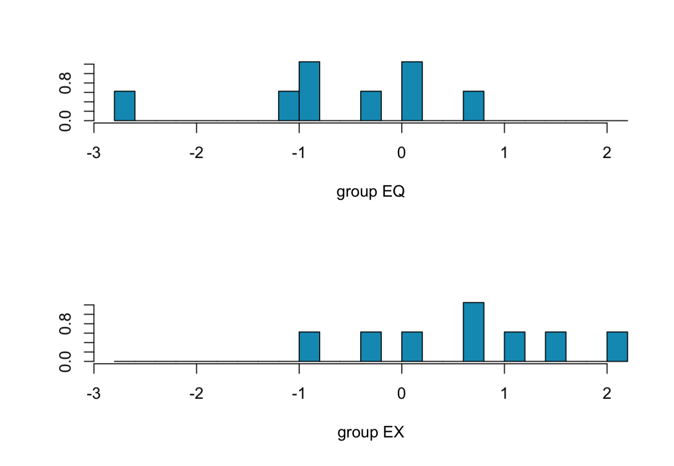
```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
lda_spec = MASS::lda( type ~ . , data = spectra )
```

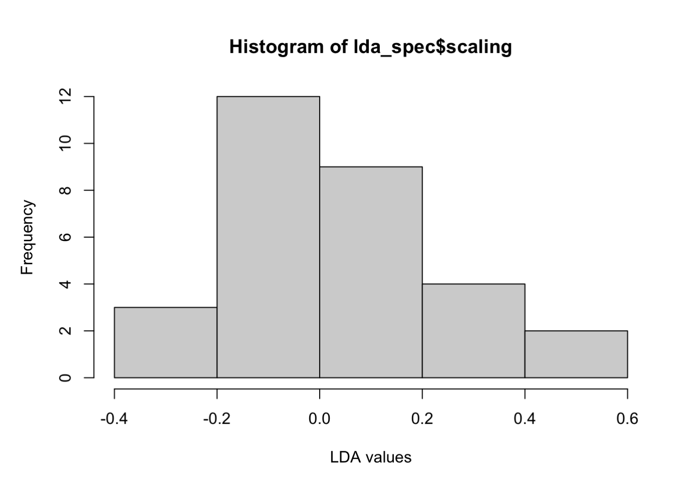
```
## Warning in lda.default(x, grouping, ...): variables are collinear
```

Historgram of LDA values

```
plot(lda_spec)
```




```
hist(lda_spec$scaling, xlab = "LDA values")
```



I would prefer to classify signals using AR coefficient analysis because everything is shown in two graphs. Doing Spectrum analysis is sophiscating, because we need to plot each graph, In this case a total of 16 graphs.

Q2

```
load("A3.RData")
```

Split the data

```
X_train = window(X, start = c(2002,1), end = c(2015,12))
X_test = window(X, start = c(2016,1), end = c(2019,12))
```

Q2 a)

```
fit = forecast::auto.arima(X_train[,'emp'])
```

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

```
#Report the fitted model
fit
```

```
## Series: X_train[, "emp"]
## ARIMA(0,0,0)(1,1,2)[12]
##
## Coefficients:
##          sar1     sma1     sma2
##       -0.0501  -0.5409  -0.1017
## s.e.   0.9659   0.9596   0.6062
##
## sigma^2 = 7.353e-06:  log likelihood = 699
## AIC=-1390   AICc=-1389.73   BIC=-1377.8
```

```
#Create 1 to 24 step ahead predictions
library(forecast)
```

```
##
## Attaching package: 'forecast'
```

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

```
fore <- forecast(fit, h = 24)
fore
```

```
##          Point Forecast         Lo 80         Hi 80        Lo 95         Hi 95
## Jan 2016  -0.0139441312 -0.0174191511 -1.046911e-02 -0.019258715 -0.0086295469
## Feb 2016   0.0031379548 -0.0003370651  6.612975e-03 -0.002176630  0.0084525391
## Mar 2016   0.0002175112 -0.0032575087  3.692531e-03 -0.005097073  0.0055320955
## Apr 2016   0.0075725897  0.0040975698  1.104761e-02  0.002258005  0.0128871740
## May 2016   0.0211424021  0.0176673822  2.461742e-02  0.015827818  0.0264569863
## Jun 2016   0.0115546164  0.0080795965  1.502964e-02  0.006240032  0.0168692007
## Jul 2016  -0.0033973512 -0.0068723711  7.766871e-05 -0.008711935  0.0019172331
## Aug 2016  -0.0005216693 -0.0039966892  2.953351e-03 -0.005836254  0.0047929150
## Sep 2016  -0.0099066488 -0.0133816687 -6.431629e-03 -0.015221233 -0.0045920645
## Oct 2016   0.0009088390 -0.0025661809  4.383859e-03 -0.004405745  0.0062234233
## Nov 2016  -0.0040156374 -0.0074906573 -5.406175e-04 -0.009330222  0.0012989469
## Dec 2016  -0.0048465345 -0.0083215544 -1.371515e-03 -0.010161119  0.0004680498
## Jan 2017  -0.0140451579 -0.0177996971 -1.029062e-02 -0.019787230 -0.0083030858
## Feb 2017   0.0032315110 -0.0005230282  6.986050e-03 -0.002510561  0.0089735832
## Mar 2017   0.0003418926 -0.0034126465  4.096432e-03 -0.005400180  0.0060839648
## Apr 2017   0.0075684620  0.0038139228  1.132300e-02  0.001826390  0.0133105342
## May 2017   0.0210251756  0.0172706364  2.477971e-02  0.015283103  0.0267672478
## Jun 2017   0.0115873603  0.0078328211  1.534190e-02  0.005845288  0.0173294325
## Jul 2017  -0.0034132765 -0.0071678157  3.412627e-04 -0.009155349  0.0023287957
## Aug 2017  -0.0005595592 -0.0043140984  3.194980e-03 -0.006301631  0.0051825129
## Sep 2017  -0.0098001593 -0.0135546984 -6.045620e-03 -0.015542231 -0.0040580871
## Oct 2017   0.0008471844 -0.0029073548  4.601724e-03 -0.004894888  0.0065892565
## Nov 2017  -0.0038644105 -0.0076189497 -1.098713e-04 -0.009606483  0.0018776617
## Dec 2017  -0.0048605556 -0.0086150948 -1.106016e-03 -0.010602628  0.0008815166
```

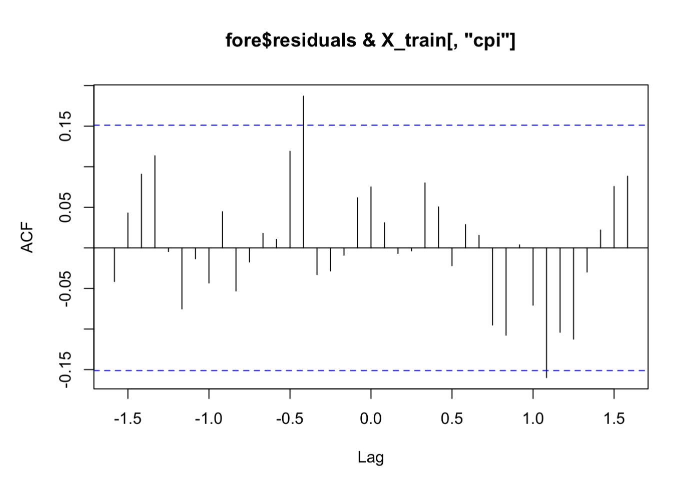
Calculate mean square error of predictions versus the actual Employment

```
mean((X_test[,'emp']-fore$mean)^2)
```

```
## [1] 7.415863e-06
```

Q2 b)

```
ccf(fore$residuals,X_train[,'cpi'])
```



No, most of the data in 95% confidence interval

Q2 c)

```
library(vars)
```

```
## Loading required package: strucchange
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

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

```
## Loading required package: sandwich
```

```
##
## Attaching package: 'strucchange'
```

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

```
## Loading required package: urca
```

```
## Loading required package: lmtest
```

```
vars::VARselect(X_train,lag.max=24)
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     12     12     12     12
##
## $criteria
##                    1             2             3             4             5
## AIC(n) -2.070175e+01 -2.070787e+01 -2.092545e+01 -2.098865e+01 -2.095894e+01
## HQ(n)  -2.065147e+01 -2.062406e+01 -2.080813e+01 -2.083781e+01 -2.077457e+01
## SC(n)  -2.057801e+01 -2.050163e+01 -2.063672e+01 -2.061743e+01 -2.050522e+01
## FPE(n)  1.021762e-09  1.015576e-09  8.170657e-10  7.671583e-10  7.905092e-10
##                    6             7             8             9            10
## AIC(n) -2.100433e+01 -2.114021e+01 -2.128446e+01 -2.145942e+01 -2.150587e+01
## HQ(n)  -2.078644e+01 -2.088880e+01 -2.099953e+01 -2.114097e+01 -2.115389e+01
## SC(n)  -2.046812e+01 -2.052150e+01 -2.058326e+01 -2.067572e+01 -2.063967e+01
## FPE(n)  7.557230e-10  6.600591e-10  5.717892e-10  4.804368e-10  4.591342e-10
##                   11            12            13            14            15
## AIC(n) -2.163149e+01 -2.283199e+01 -2.280390e+01 -2.280563e+01 -2.277383e+01
## HQ(n)  -2.124600e+01 -2.241297e+01 -2.235137e+01 -2.231958e+01 -2.225425e+01
## SC(n)  -2.068280e+01 -2.180080e+01 -2.169022e+01 -2.160946e+01 -2.149516e+01
## FPE(n)  4.054698e-10  1.222584e-10  1.259767e-10  1.260339e-10  1.304355e-10
##                   16            17            18            19            20
## AIC(n) -2.276365e+01 -2.272030e+01 -2.268764e+01 -2.272871e+01 -2.270190e+01
## HQ(n)  -2.221055e+01 -2.213367e+01 -2.206749e+01 -2.207505e+01 -2.201472e+01
## SC(n)  -2.140248e+01 -2.127664e+01 -2.116148e+01 -2.112007e+01 -2.101076e+01
## FPE(n)  1.321512e-10  1.384596e-10  1.435872e-10  1.383830e-10  1.428055e-10
##                   21            22            23            24
## AIC(n) -2.269157e+01 -2.267628e+01 -2.265139e+01 -2.271290e+01
## HQ(n)  -2.197086e+01 -2.192205e+01 -2.186364e+01 -2.189163e+01
## SC(n)  -2.091793e+01 -2.082015e+01 -2.071276e+01 -2.069178e+01
## FPE(n)  1.450362e-10  1.481157e-10  1.528109e-10  1.446964e-10
```

```
#AIC = 12
```

Report the fitted model

```
newfit <- vars::VAR(X_train,p=12) %>% vars::restrict()
newfit
```

```
##
## VAR Estimation Results:
## =======================
##
## Estimated coefficients for equation emp:
## ========================================
## Call:
## emp = emp.l12
##
##   emp.l12
## 0.9370206
##
##
## Estimated coefficients for equation cpi:
## ========================================
## Call:
## cpi = cpi.l1 + emp.l3 + emp.l9 + emp.l11 + const
##
##       cpi.l1       emp.l3       emp.l9      emp.l11        const
## 0.1730817429 0.0773155827 0.1465263754 0.1342325394 0.0008171901
```

Create 1-to-24-step-ahead predictions

```
newfore <- forecast(newfit, h = 24)
newfore
```

```
## emp
##          Point Forecast         Lo 80         Hi 80         Lo 95         Hi 95
## Jan 2016  -0.0121943456 -0.0161354090 -0.0082532822 -0.0182216820 -0.0061670093
## Feb 2016   0.0021132719 -0.0018277915  0.0060543353 -0.0039140644  0.0081406083
## Mar 2016  -0.0005856463 -0.0045267097  0.0033554171 -0.0066129826  0.0054416900
## Apr 2016   0.0066835843  0.0027425209  0.0106246477  0.0006562479  0.0127109206
## May 2016   0.0204744964  0.0165334330  0.0244155598  0.0144471601  0.0265018328
## Jun 2016   0.0106527957  0.0067117323  0.0145938591  0.0046254594  0.0166801321
## Jul 2016  -0.0029676954 -0.0069087588  0.0009733680 -0.0089950318  0.0030596409
## Aug 2016  -0.0004454480 -0.0043865114  0.0034956154 -0.0064727844  0.0055818883
## Sep 2016  -0.0097229354 -0.0136639988 -0.0057818720 -0.0157502718 -0.0036955991
## Oct 2016   0.0014907595 -0.0024503039  0.0054318229 -0.0045365768  0.0075180959
## Nov 2016  -0.0050467595 -0.0089878229 -0.0011056961 -0.0110740959  0.0009805768
## Dec 2016  -0.0046198318 -0.0085608952 -0.0006787684 -0.0106471681  0.0014075046
## Jan 2017  -0.0114263533 -0.0168272030 -0.0060255035 -0.0196862402 -0.0031664663
## Feb 2017   0.0019801794 -0.0034206704  0.0073810292 -0.0062797076  0.0102400663
## Mar 2017  -0.0005487627 -0.0059496124  0.0048520871 -0.0088086496  0.0077111243
## Apr 2017   0.0062626563  0.0008618065  0.0116635060 -0.0019972307  0.0145225432
## May 2017   0.0191850253  0.0137841755  0.0245858751  0.0109251383  0.0274449122
## Jun 2017   0.0099818892  0.0045810394  0.0153827390  0.0017220023  0.0182417762
## Jul 2017  -0.0027807918 -0.0081816416  0.0026200580 -0.0110406788  0.0054790951
## Aug 2017  -0.0004173940 -0.0058182438  0.0049834558 -0.0086772809  0.0078424930
## Sep 2017  -0.0091105910 -0.0145114407 -0.0037097412 -0.0173704779 -0.0008507040
## Oct 2017   0.0013968724 -0.0040039774  0.0067977222 -0.0068630146  0.0096567594
## Nov 2017  -0.0047289177 -0.0101297675  0.0006719320 -0.0129888047  0.0035309692
## Dec 2017  -0.0043288776 -0.0097297274  0.0010719722 -0.0125887646  0.0039310093
##
## cpi
##          Point Forecast         Lo 80       Hi 80        Lo 95       Hi 95
## Jan 2016   0.0014690755 -0.0025620499 0.005500201 -0.004695999 0.007634150
## Feb 2016   0.0037728393 -0.0003182213 0.007863900 -0.002483898 0.010029577
## Mar 2016   0.0037122903 -0.0003805523 0.007805133 -0.002547172 0.009971753
## Apr 2016   0.0029859006 -0.0011185593 0.007090360 -0.003291329 0.009263130
## May 2016   0.0029537893 -0.0011510181 0.007058597 -0.003323972 0.009231550
## Jun 2016  -0.0006624003 -0.0047672181 0.003442418 -0.006940177 0.005615377
## Jul 2016   0.0013885906 -0.0027162275 0.005493409 -0.004889187 0.007666368
## Aug 2016   0.0004584861 -0.0036463320 0.004563304 -0.005819291 0.006736264
## Sep 2016   0.0012113059 -0.0028935122 0.005316124 -0.005066472 0.007489084
## Oct 2016  -0.0017123689 -0.0058576087 0.002432871 -0.008051966 0.004627228
## Nov 2016   0.0001342081 -0.0040122365 0.004280653 -0.006207232 0.006475648
## Dec 2016  -0.0016340059 -0.0058162860 0.002548274 -0.008030251 0.004762240
## Jan 2017   0.0019126237 -0.0022707252 0.006095973 -0.004485256 0.008310504
## Feb 2017   0.0036794782 -0.0005039027 0.007862859 -0.002718451 0.010077407
## Mar 2017   0.0035549257 -0.0006284562 0.007738308 -0.002843005 0.009952856
## Apr 2017   0.0028625457 -0.0013306015 0.007055693 -0.003550319 0.009275411
## May 2017   0.0028304252 -0.0013630142 0.007023865 -0.003582887 0.009243737
## Jun 2017  -0.0005583706 -0.0047518187 0.003635077 -0.006971696 0.005854955
## Jul 2017   0.0013633893 -0.0028300591 0.005556838 -0.005049937 0.007776715
## Aug 2017   0.0004918516 -0.0037015967 0.004685300 -0.005921474 0.006905178
## Sep 2017   0.0011972575 -0.0029961909 0.005390706 -0.005216068 0.007610583
## Oct 2017  -0.0015422865 -0.0057705022 0.002685929 -0.008008784 0.004924211
## Nov 2017   0.0001879942 -0.0040412586 0.004417247 -0.006280090 0.006656078
## Dec 2017  -0.0014688588 -0.0057289800 0.002791263 -0.007984152 0.005046434
```

Calculate the MSE of the predictions versus the actual Employment data

```
mean((newfore$forecast$emp$mean-X_test[,'emp'])^2)
```

```
## [1] 7.359114e-06
```

This is really close to the univariate data from part a. (almost the same)

Q2 d)

```
GCT <- vars::causality(newfit)
```

```
## Warning in vars::causality(newfit):
## Argument 'cause' has not been specified;
## using first variable in 'x$y' (emp) as cause variable.
```

```
GCT
```

```
## $Granger
##
##  Granger causality H0: emp do not Granger-cause cpi
##
## data:  VAR object newfit
## F-Test = 2.4124, df1 = 12, df2 = 262, p-value = 0.005578
##
##
## $Instant
##
##  H0: No instantaneous causality between: emp and cpi
##
## data:  VAR object newfit
## Chi-squared = 3.5654, df = 1, p-value = 0.059
```

```
#do not Granger-causes employment
#p-value is 0.005578
```

Yes the test result coincide with my MSE comparison