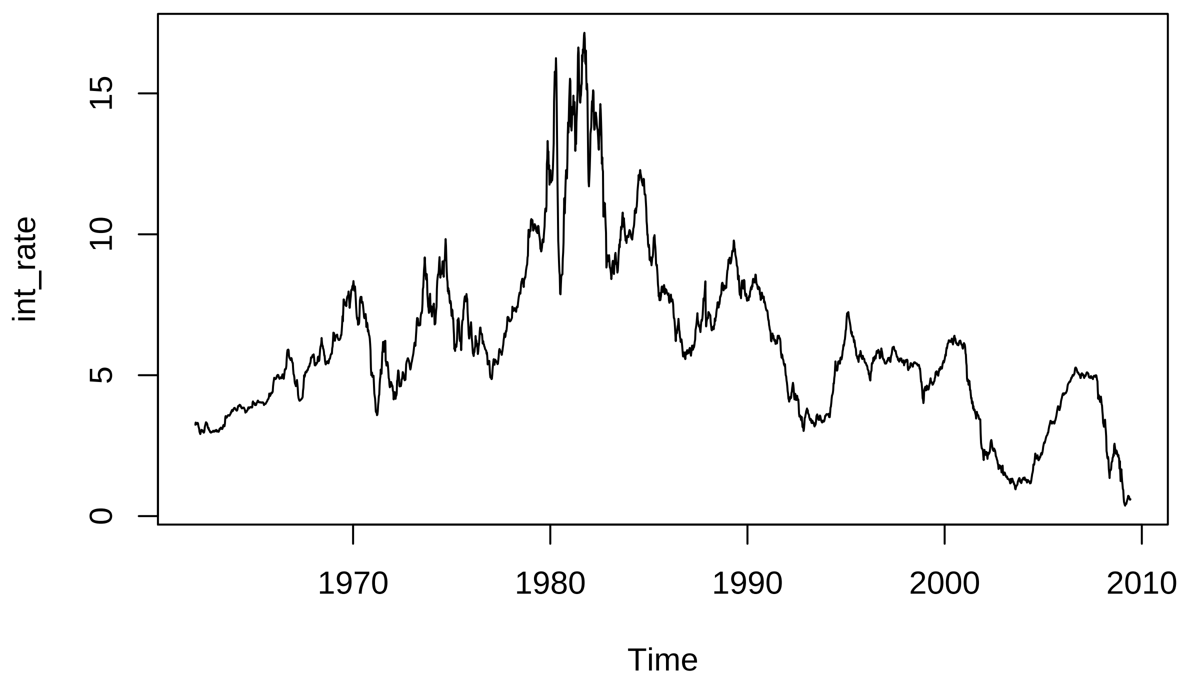
Homework3

Austin Lee

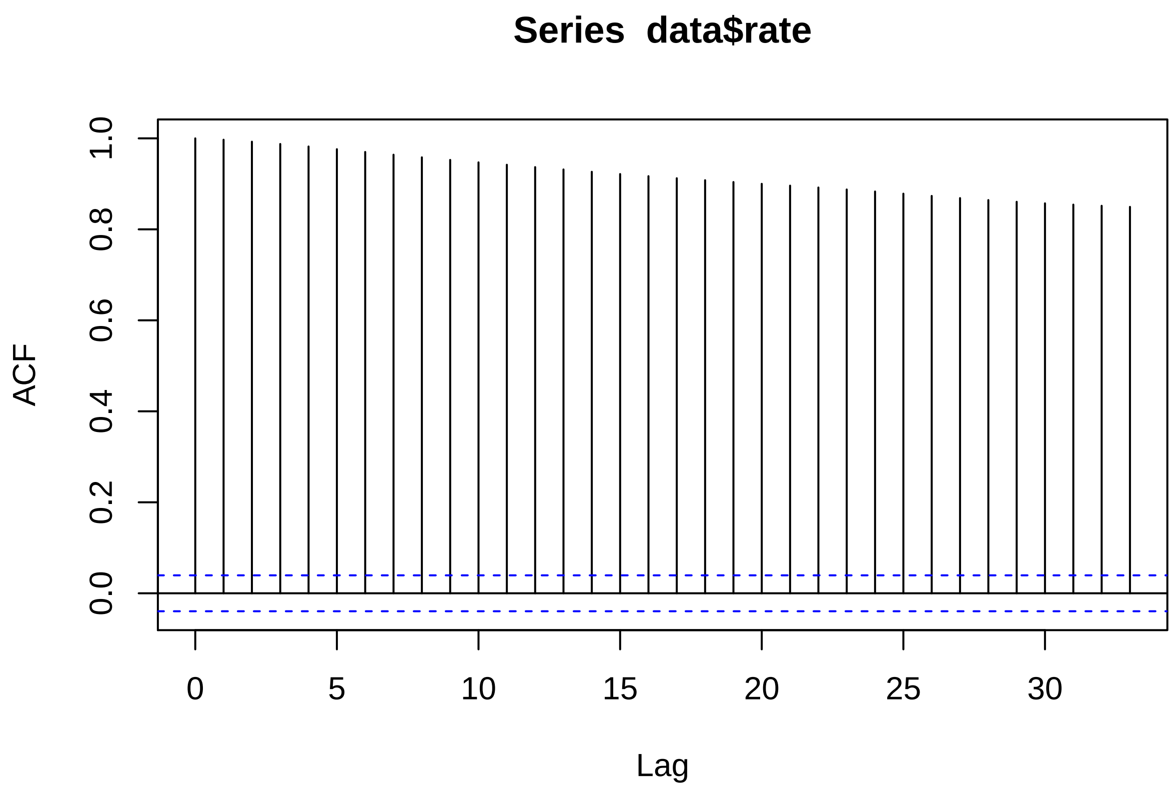
February 21, 2019

1 Part A

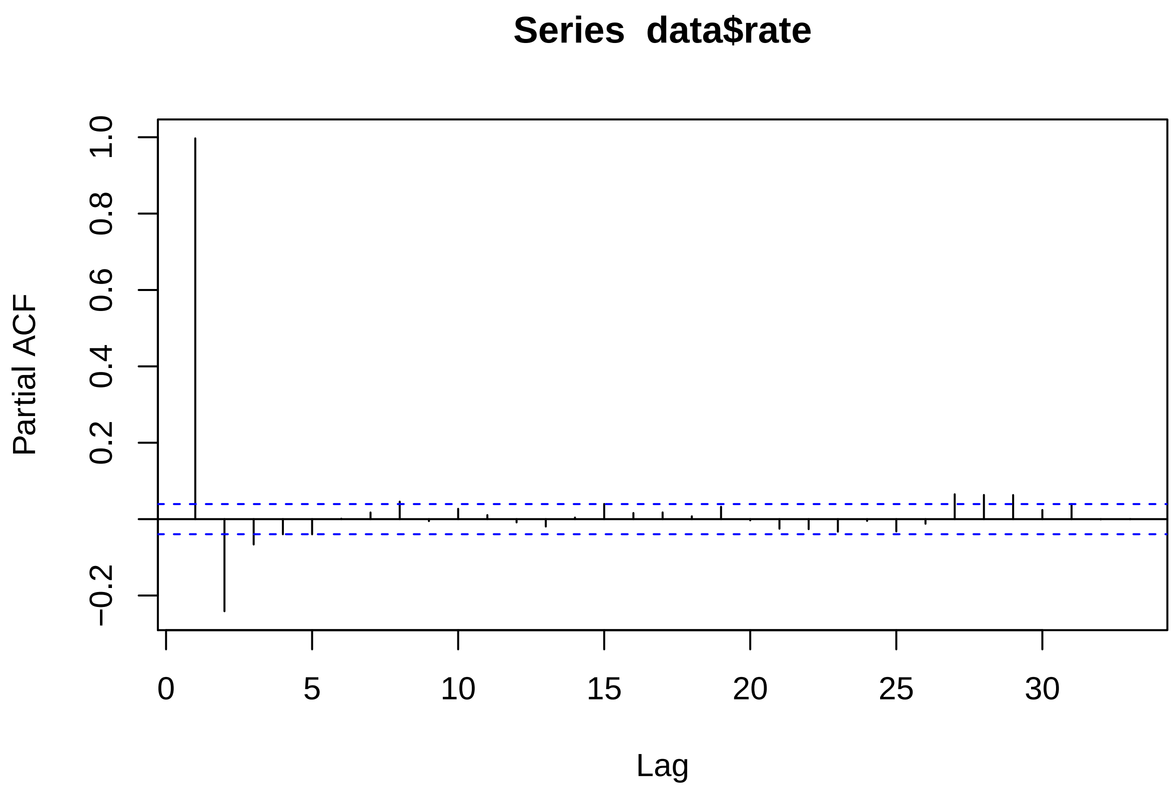
```r
#C:\Users\Austin\Downloads\w-gs1yr.txt

data <-read.table(file = 'c:\\Users\\Austin\\Downloads\\w-gs1yr.txt', header = TRUE)
int_rate <- ts(data$rate, start = 1962, freq = 52)
plot(int_rate)
```




```r
acf(data$rate)
```




```r
pacf(data$rate)
```



By looking at the PACF, we can see that there are two spikes that exist on the first and second lag, indicating that this may be an AR(2). We also see a declining trend with the ACF, which supports the AR(2)

1 Part B

```
modarma1 <- (arima(int_rate,order = c(2,0,0)))
modarma2 <- (arima(int_rate,order = c(1,0,0)))
modarma3 <- (arima(int_rate,order = c(1,0,1)))

(modarma1)
```

```
##
## Call:
## arima(x = int_rate, order = c(2, 0, 0))
##
## Coefficients:
##          ar1      ar2  intercept
##       1.3434  -0.3458     6.0621
## s.e.  0.0189   0.0189     1.3612
##
## sigma^2 estimated as 0.03151:  log likelihood = 761.22,  aic = -1514.44
```

```
(modarma2)
```

```
##
## Call:
## arima(x = int_rate, order = c(1, 0, 0))
```

```
##
## Coefficients:
##          ar1  intercept
##       0.9985     6.0944
## s.e.  0.0014     2.3646
##
## sigma^2 estimated as 0.03579:  log likelihood = 604.11,  aic = -1202.23
```

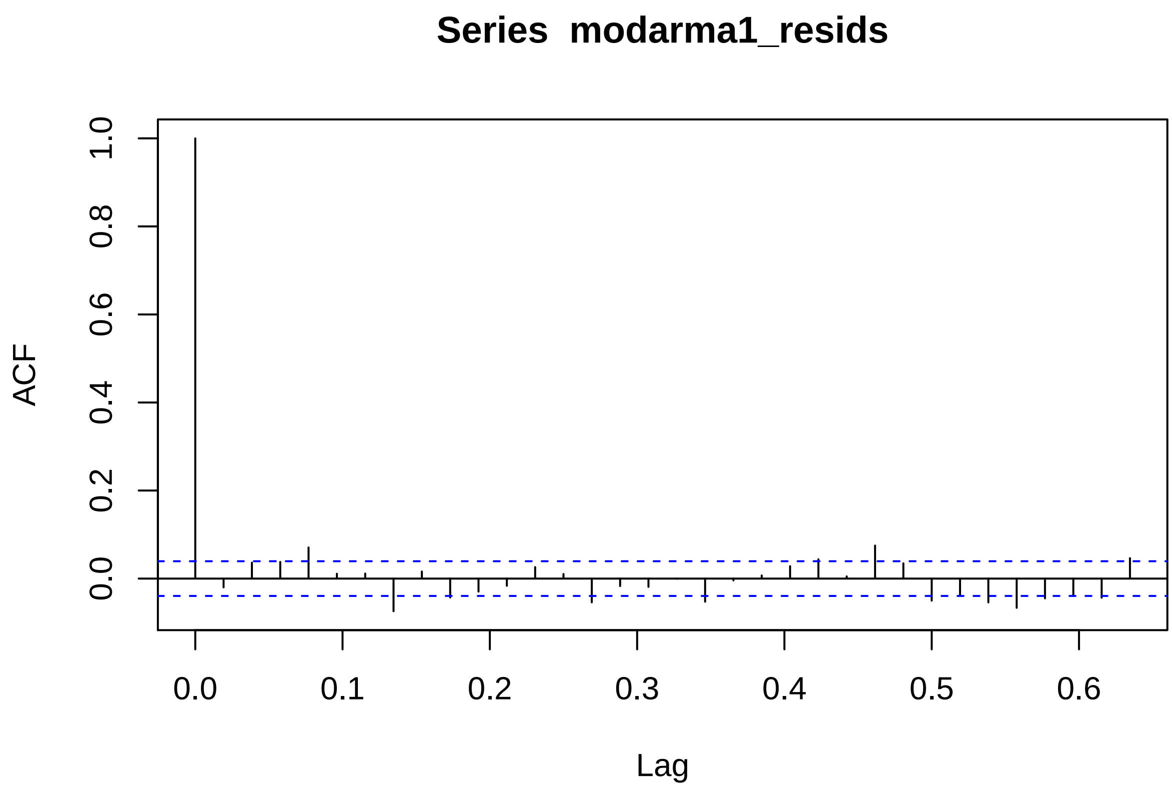
```
(modarma3)
```

```
##
## Call:
## arima(x = int_rate, order = c(1, 0, 1))
##
## Coefficients:
##          ar1     ma1  intercept
##       0.9974  0.2956     6.0927
## s.e.  0.0016  0.0172     1.6807
##
## sigma^2 estimated as 0.0322:  log likelihood = 734.69,  aic = -1461.38
```

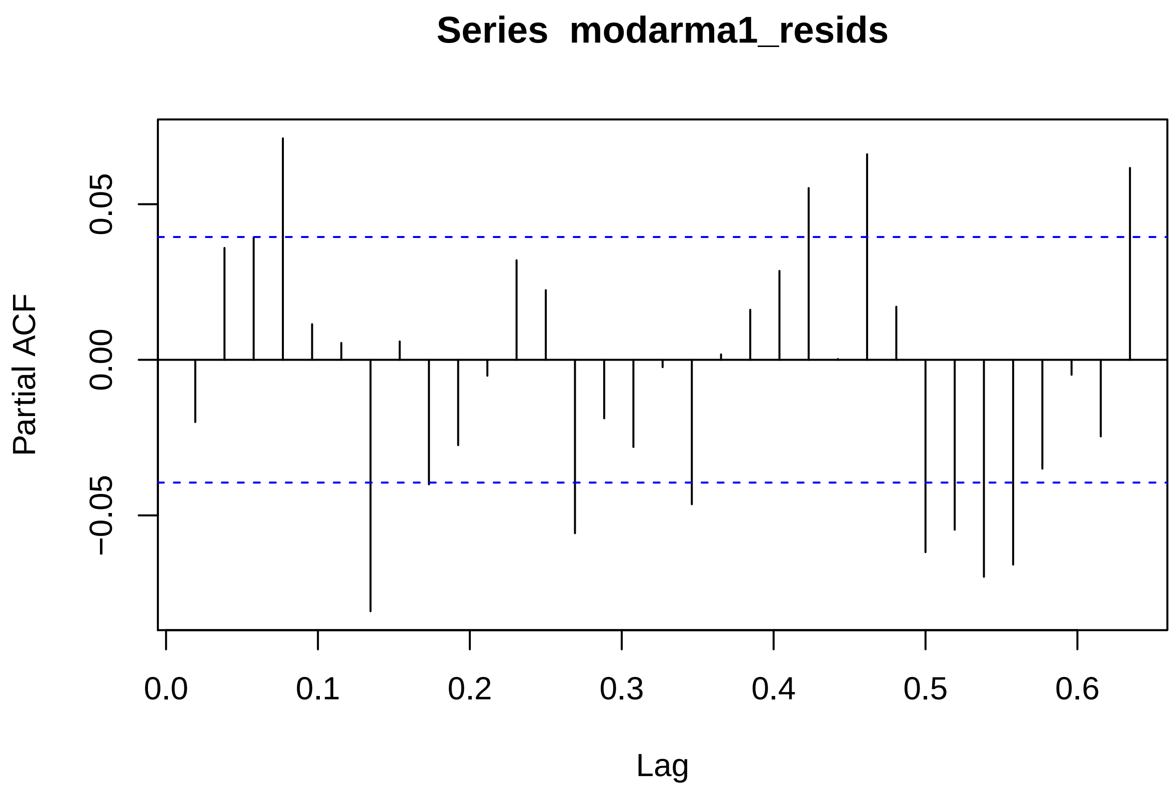
It was previously discussed that AR(2) would be a good fit according to the ACF and PACF. The AR(2) has the best AIC, and lowest variance estimated at .03151, so we will use that model.

1 Part C

```
modarma1_resids <- modarma1$residuals

acf(modarma1_resids)
```




```r
pacf(modarma1_resids)
```



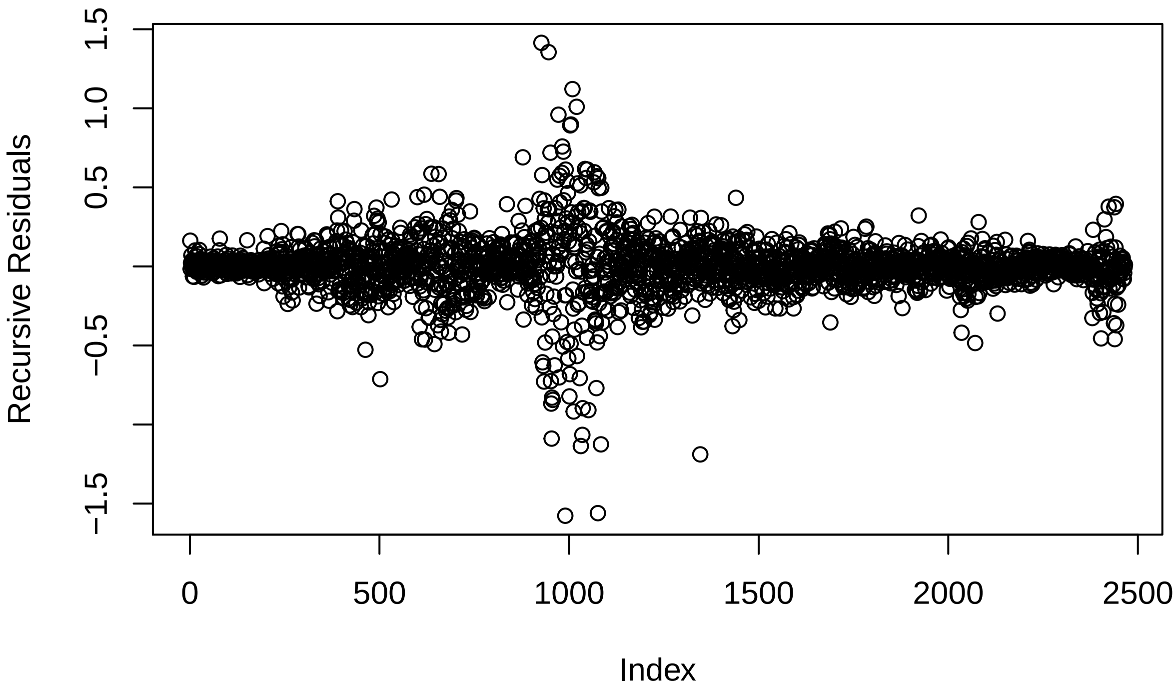
By looking at the PACF, we see that there still exists some cyclical trends within our data that our model did not capture. However, we can see that our model nearly eliminated the residuals to white noise.

1 Part D

```
library("strucchange")
```

```
## Warning: package 'strucchange' was built under R version 3.5.2

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 3.5.2

##
## Attaching package: 'zoo'

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

## Loading required package: sandwich

## Warning: package 'sandwich' was built under R version 3.5.2
```

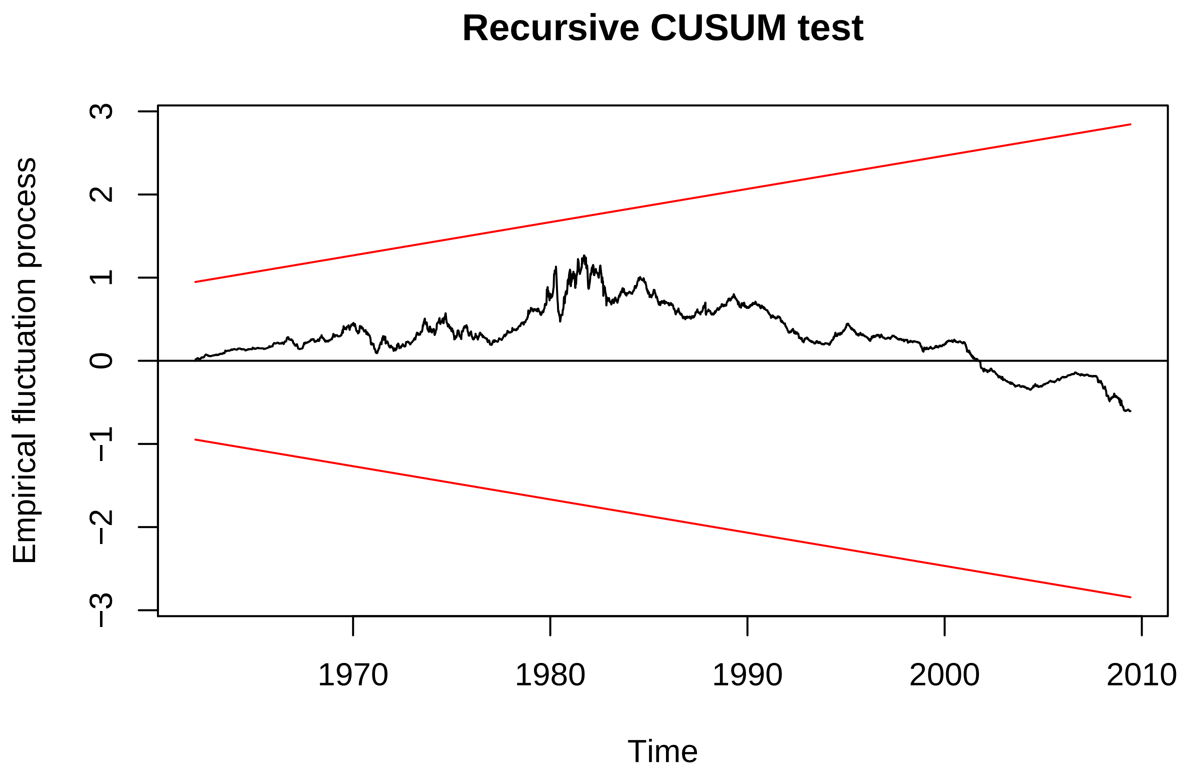
```
recursive_resids <- recresid(modarma1_resids~1)
plot(recursive_resids, ylab= "Recursive Residuals")
```



There seems to be a structural break at index 1000. Other than that, our residuals are mostly centered at 0.

1 Part E

```
plot(efp(modarma1_resids~1, type = "Rec-CUSUM"))
```



Although there was a large variance at Index 1000 for the recursive residual plot, we see that our model did not have any structural breaks.

1 Part F

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.5.2
```

```
modarma4 <- auto.arima(int_rate)
summary(modarma4)
```

```
## Series: int_rate
## ARIMA(1,1,2)
##
## Coefficients:
##          ar1      ma1      ma2
##       0.6284  -0.3065  -0.0527
## s.e.  0.0642   0.0675   0.0299
##
## sigma^2 estimated as 0.03143:  log likelihood=768.32
## AIC=-1528.65   AICc=-1528.63   BIC=-1505.41
##
## Training set error measures:
##                         ME      RMSE       MAE         MPE     MAPE
## Training set -0.0006210918 0.1771539 0.1046884 -0.05084758 1.820023
##                   MASE          ACF1
```

```
## Training set 0.07539674 -0.0002601686
```

```r
summary(modarma1)
```

```
##
## Call:
## arima(x = int_rate, order = c(2, 0, 0))
##
## Coefficients:
##          ar1      ar2  intercept
##       1.3434  -0.3458     6.0621
## s.e.  0.0189   0.0189     1.3612
##
## sigma^2 estimated as 0.03151:  log likelihood = 761.22,  aic = -1514.44
##
## Training set error measures:
##                          ME      RMSE       MAE        MPE     MAPE
## Training set -0.0006784007 0.1775171 0.1050281 -0.1586663 1.832503
##                   MASE        ACF1
## Training set 0.9398335 -0.02001622
```

```r
BIC(modarma1)
```

```
## [1] -1491.193
```

According to the auto.arima, it yields a better AIC and BIC. If we plotted the CUSUM of the modarma4, we would see that there would be a less noticable structural break in the plot, possibly indicating that ARIMA(1,1,2) is better than AR(2).

1 Part G

```r
library(forecast)
forecast(modarma1, h = 24)
```

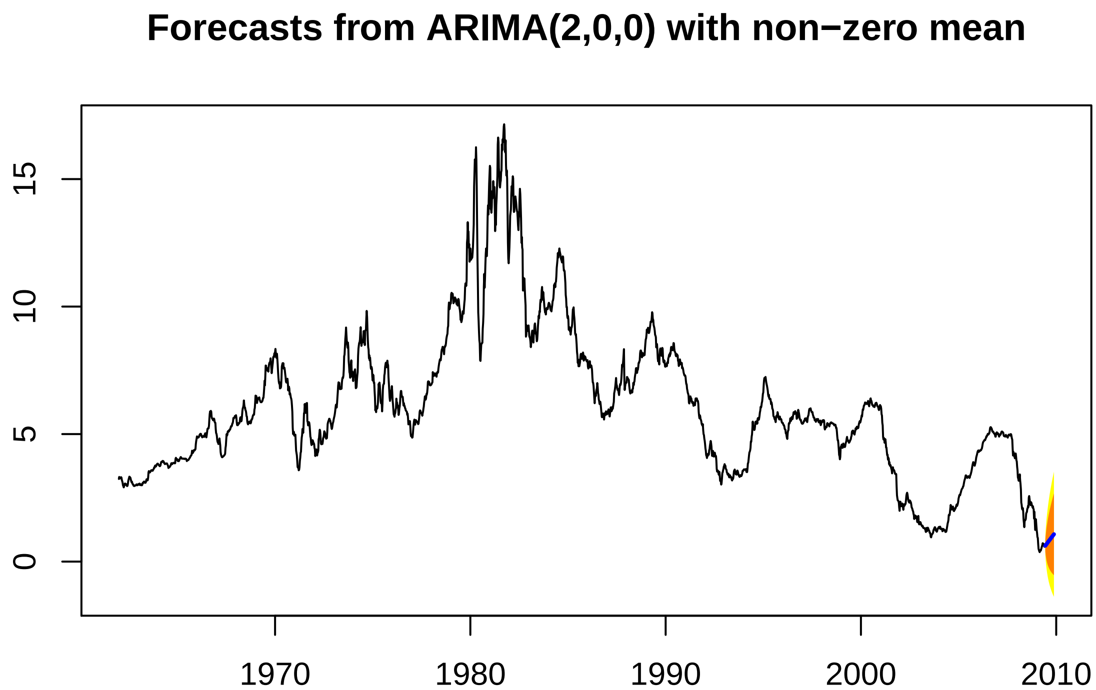
```
##          Point Forecast        Lo 80     Hi 80       Lo 95     Hi 95
## 2009.442      0.6203267  0.392829412 0.8478241  0.27239960 0.9682539
## 2009.462      0.6407166  0.259724137 1.0217090  0.05803893 1.2233942
## 2009.481      0.6610782  0.155806720 1.1663496 -0.11166782 1.4338241
## 2009.500      0.6813800  0.072282416 1.2904776 -0.25015436 1.6129143
## 2009.519      0.7016113  0.002982824 1.4002398 -0.36684876 1.7700714
## 2009.538      0.7217686 -0.056092530 1.4996297 -0.46786733 1.9114045
## 2009.558      0.7418507 -0.107529788 1.5912313 -0.55716468 2.0408662
## 2009.577      0.7618576 -0.153045079 1.6767604 -0.63736532 2.1610806
## 2009.596      0.7817894 -0.193818829 1.7573976 -0.71027462 2.2738534
## 2009.615      0.8016462 -0.230693615 1.8339860 -0.77718131 2.3804737
## 2009.635      0.8214284 -0.264291844 1.9071486 -0.83903740 2.4818941
## 2009.654      0.8411361 -0.295087359 1.9773596 -0.89656774 2.5788400
## 2009.673      0.8607698 -0.323450353 2.0449899 -0.95033863 2.6718782
## 2009.692      0.8803296 -0.349676468 2.1103357 -1.00080236 2.7614616
## 2009.712      0.8998159 -0.374006253 2.1736380 -1.04832697 2.8479588
## 2009.731      0.9192289 -0.396638582 2.2350964 -1.09321675 2.9316745
## 2009.750      0.9385689 -0.417740142 2.2948779 -1.13572677 3.0128645
## 2009.769      0.9578361 -0.437452314 2.3531246 -1.17607341 3.0917457
## 2009.788      0.9770310 -0.455896276 2.4099582 -1.21444213 3.1685040
## 2009.808      0.9961536 -0.473176844 2.4654840 -1.25099339 3.2433005
```

```
## 2009.827      1.0152043 -0.489385426 2.5197940 -1.28586711 3.3162757
## 2009.846      1.0341834 -0.504602321 2.5729691 -1.31918626 3.3875530
## 2009.865      1.0530911 -0.518898534 2.6250807 -1.35105956 3.4572417
## 2009.885      1.0719277 -0.532337224 2.6761926 -1.38158376 3.5254391
```

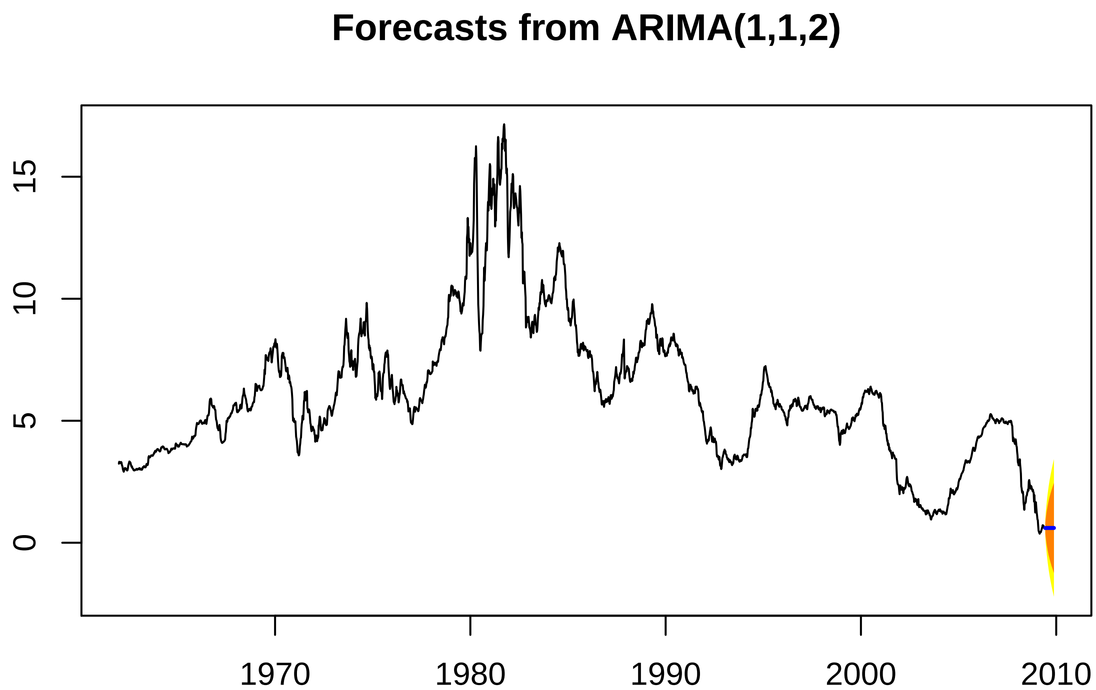
```r
forecast(modarma4, h = 24)
```

```
##          Point Forecast       Lo 80     Hi 80       Lo 95     Hi 95
## 2009.442      0.6038364  0.37662026 0.8310525  0.25633931 0.9513334
## 2009.462      0.6048191  0.22819773 0.9814404  0.02882644 1.1808117
## 2009.481      0.6054366  0.10181682 1.1090563 -0.16478337 1.3756565
## 2009.500      0.6058246 -0.01074696 1.2223962 -0.33714025 1.5487895
## 2009.519      0.6060685 -0.11255286 1.3246898 -0.49296803 1.7051050
## 2009.538      0.6062217 -0.20554821 1.4179917 -0.63527326 1.8477167
## 2009.558      0.6063180 -0.29120884 1.5038449 -0.76633086 1.9789669
## 2009.577      0.6063785 -0.37070891 1.5834660 -0.88794775 2.1007048
## 2009.596      0.6064165 -0.44499495 1.6578280 -1.00157858 2.2144117
## 2009.615      0.6064404 -0.51483360 1.7277145 -1.10840023 2.3212811
## 2009.635      0.6064555 -0.58084800 1.7937589 -1.20936848 2.4222794
## 2009.654      0.6064649 -0.64354683 1.8564766 -1.30526306 2.5181929
## 2009.673      0.6064708 -0.70334764 1.9162893 -1.39672365 2.6096653
## 2009.692      0.6064746 -0.76059532 1.9735444 -1.48427839 2.6972275
## 2009.712      0.6064769 -0.81557675 2.0285305 -1.56836648 2.7813203
## 2009.731      0.6064784 -0.86853233 2.0814891 -1.64935582 2.8623126
## 2009.750      0.6064793 -0.91966500 2.1326236 -1.72755699 2.9405156
## 2009.769      0.6064799 -0.96914740 2.1821071 -1.80323409 3.0161938
## 2009.788      0.6064802 -1.01712746 2.2300879 -1.87661345 3.0895739
## 2009.808      0.6064805 -1.06373284 2.2766938 -1.94789036 3.1608513
## 2009.827      0.6064806 -1.10907448 2.3220357 -2.01723448 3.2301957
## 2009.846      0.6064807 -1.15324937 2.3662108 -2.08479420 3.2977556
## 2009.865      0.6064808 -1.19634287 2.4093044 -2.15070006 3.3636616
## 2009.885      0.6064808 -1.23843051 2.4513921 -2.21506756 3.4280291
```

```r
plot(forecast(modarma1, h = 24), shadecols = "oldstyle" )
```



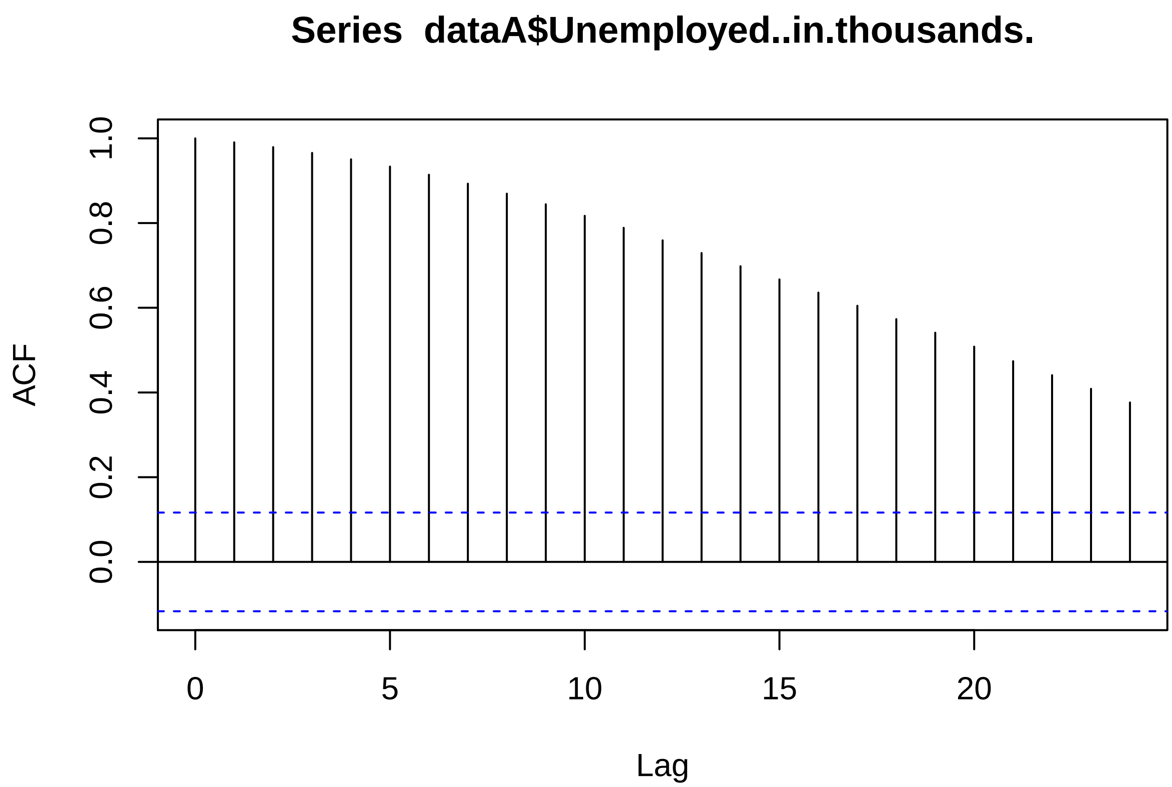

```r
plot(forecast(modarma4, h = 24), shadecols = "oldstyle")
```



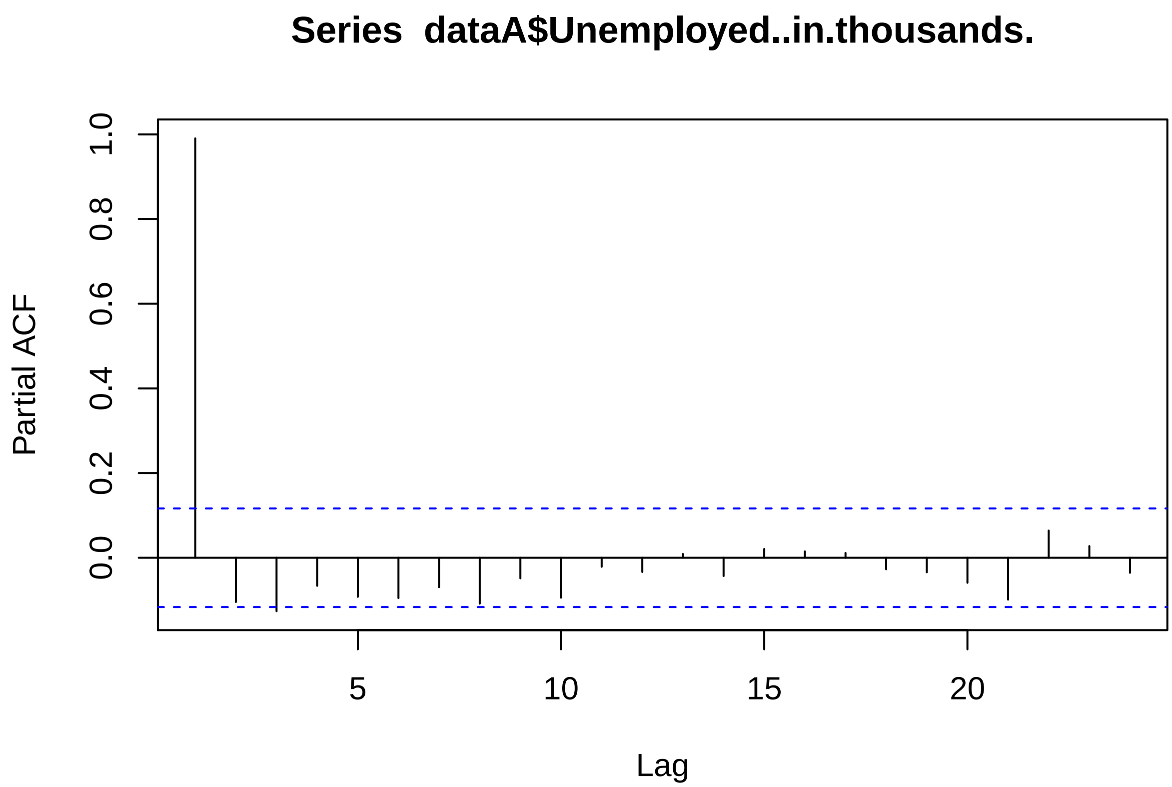
It appears that my model trends upwards while the R's fitted models reversts around .606.

Problem 2

```
dataA<-read.table(file ='C:\\Users\\Austin\\Documents\\R\\Exercise7.2.csv'
                  , header = T, sep = ',')
acf(dataA$Unemployed..in.thousands.)
```



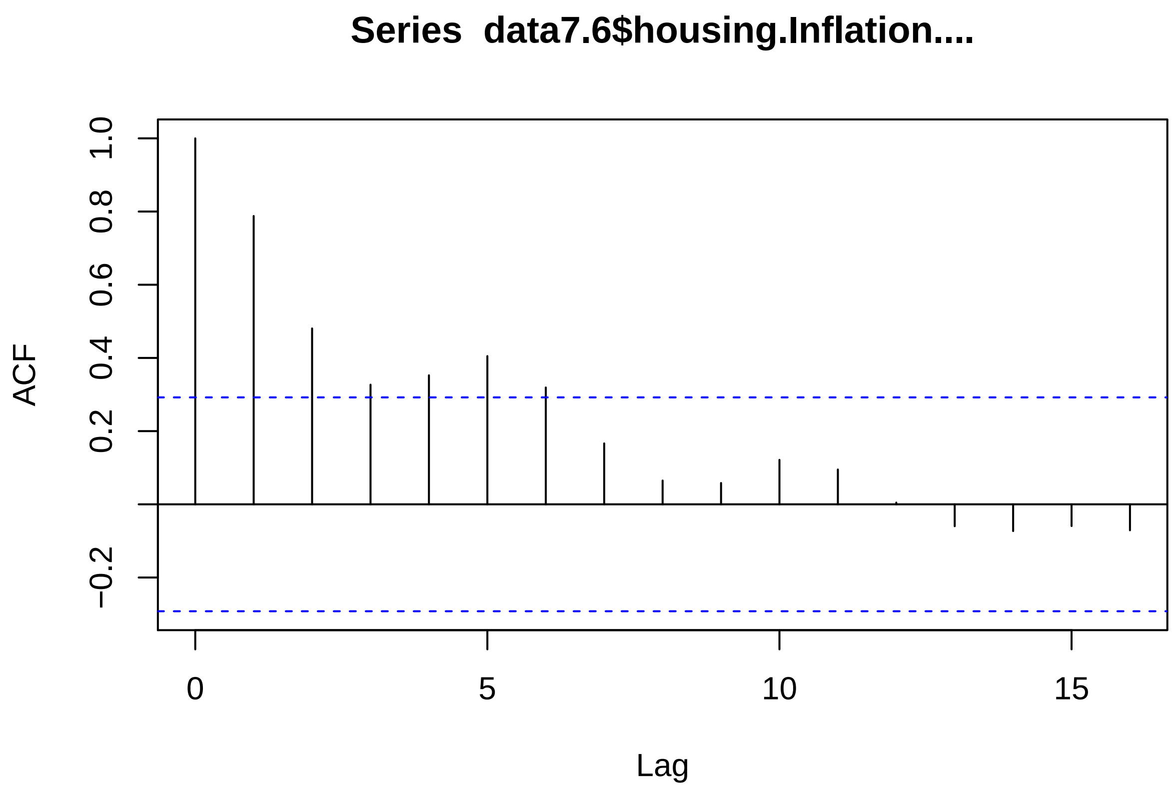

```r
pacf(dataA$Unemployed..in.thousands.)
```



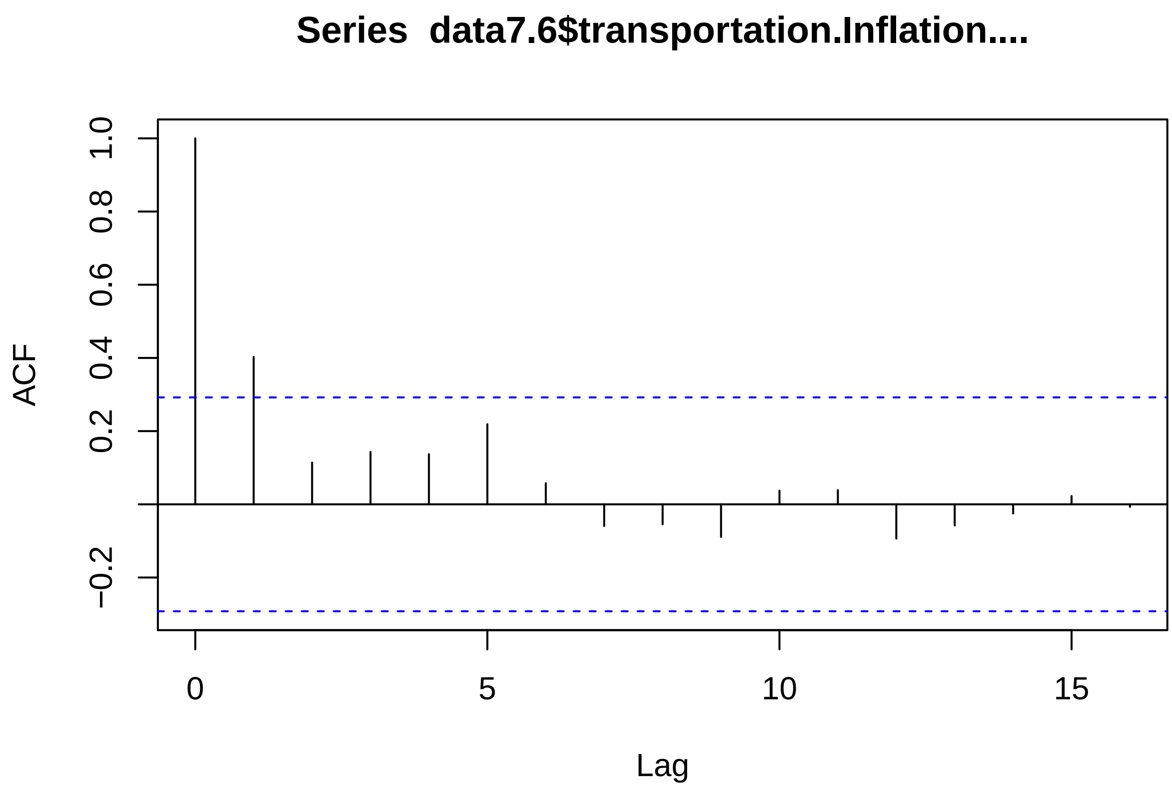
Based on the ACF and PACF alone, it appears that an AR(1) would be a good fit to explain the time dependence within the model. The autocorrelation coefficients decline overtime, indicating high persistence, which would be consistent with AR(1).

Problem 3

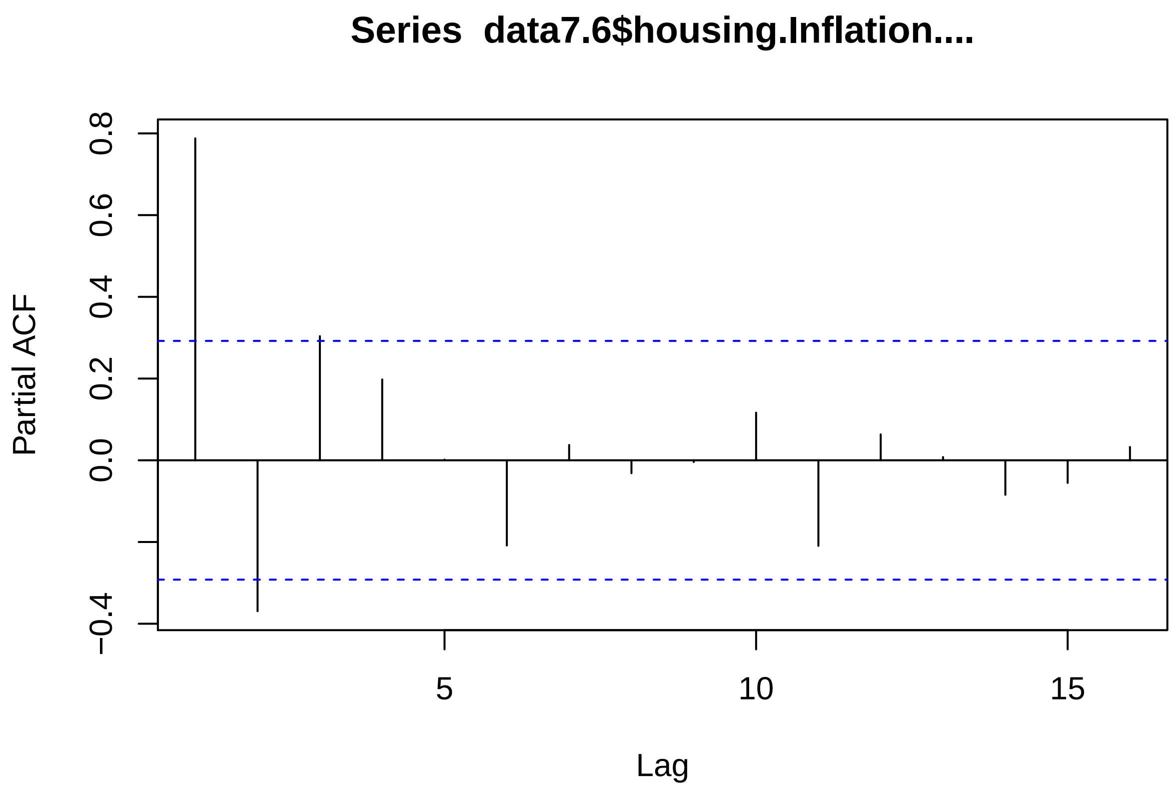
```
library(forecast)
data7.6 <- read.table(file = 'C:\\Users\\Austin\\Documents\\R\\Exercise7.6.csv',
                      header = T, sep = ',')

acf(data7.6$housing.Inflation....,na.action = na.pass)
```



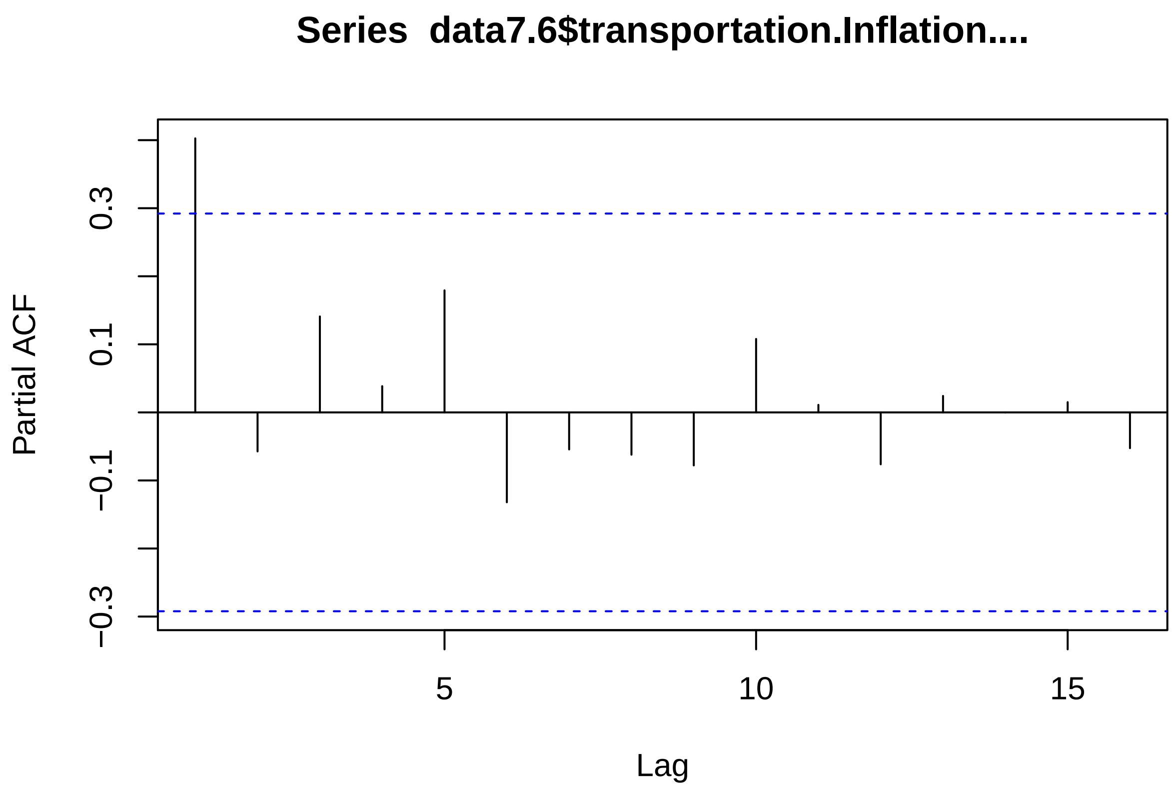

```
acf(data7.6$transportation.Inflation...., na.action = na.pass)
```




```
pacf(data7.6$housing.Inflation....,na.action = na.pass)
```




```
pacf(data7.6$transportation.Inflation...., na.action = na.pass)
```




```r
auto.arima(data7.6$housing.Inflation....)
```

```
## Series: data7.6$housing.Inflation....
## ARIMA(2,1,2)
##
## Coefficients:
##          ar1      ar2      ma1     ma2
##       0.6976  -0.9291  -0.4953  0.6634
## s.e.  0.1066   0.0824   0.1919  0.1816
##
## sigma^2 estimated as 3.115:  log likelihood=-84.06
## AIC=178.12   AICc=179.74   BIC=186.93
```

```r
auto.arima(data7.6$transportation.Inflation....)
```

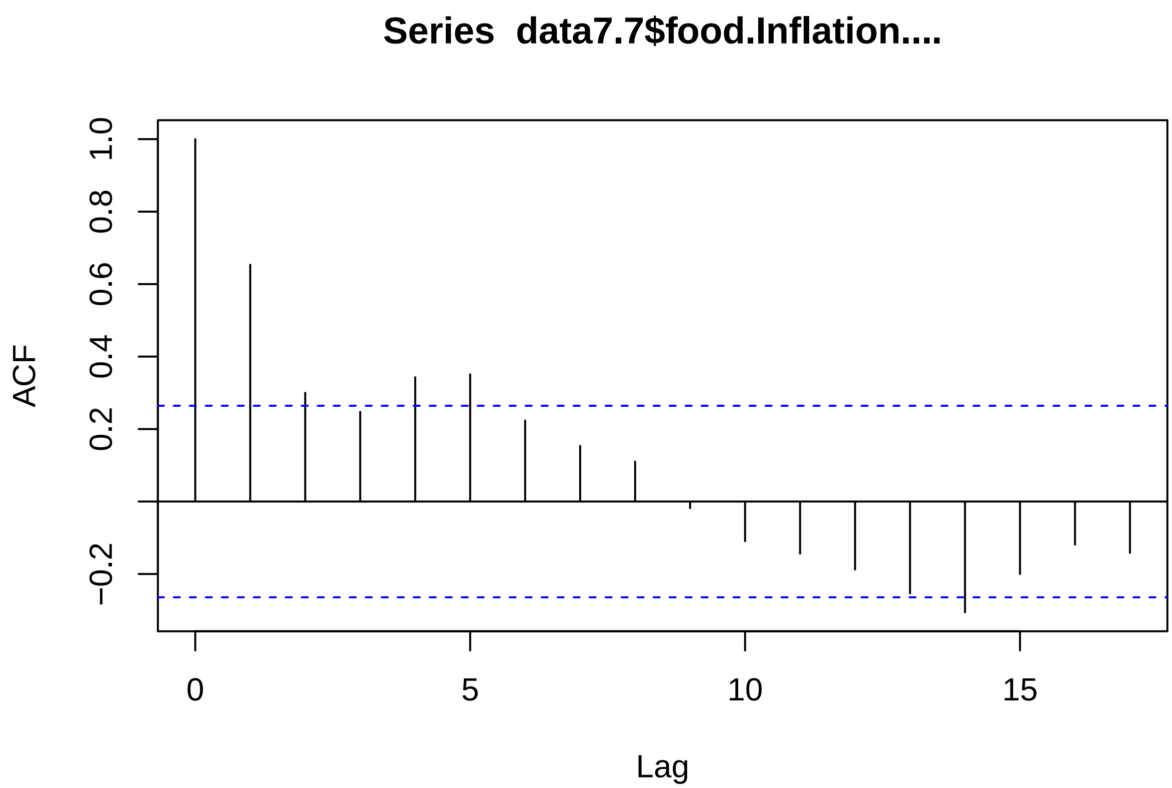
```
## Series: data7.6$transportation.Inflation....
## ARIMA(0,0,1) with non-zero mean
##
## Coefficients:
##          ma1    mean
##       0.4336  4.3965
## s.e.  0.1442  0.8751
##
## sigma^2 estimated as 17.4:  log likelihood=-124.36
## AIC=254.72   AICc=255.32   BIC=260.07
```

For the housing inflation, its possible that an AR(2) would be viable because there are two spikes at the PACF at 1 and 2. According to the auto.arima function, which fits the best arima model based on AIC and

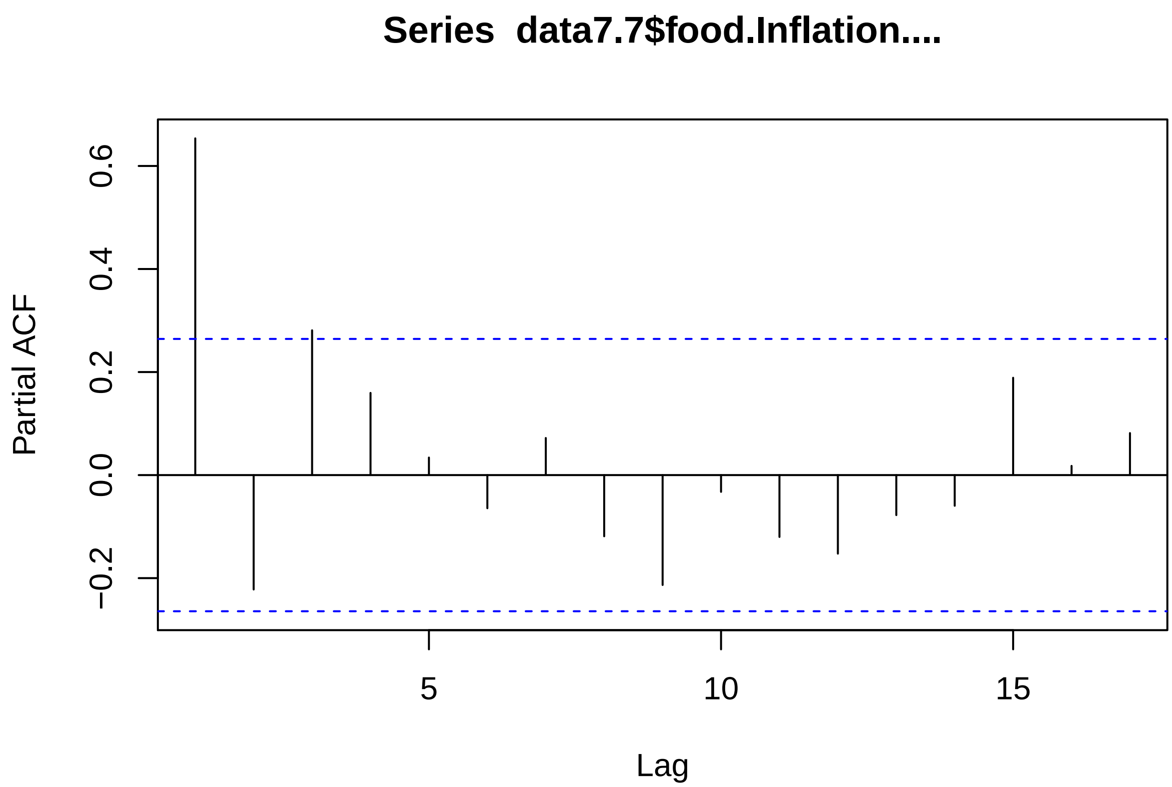
BIC, the housing inflation is not property modeled by just AR(2). We can see that a ARIMA(2,1,2) is a better fit. According to the auto.arima function, a MA(1) seems to be a better fit and a AR(2) does not seem viable; there aren't any statistically significant spikes at lag 2.

Problem 4

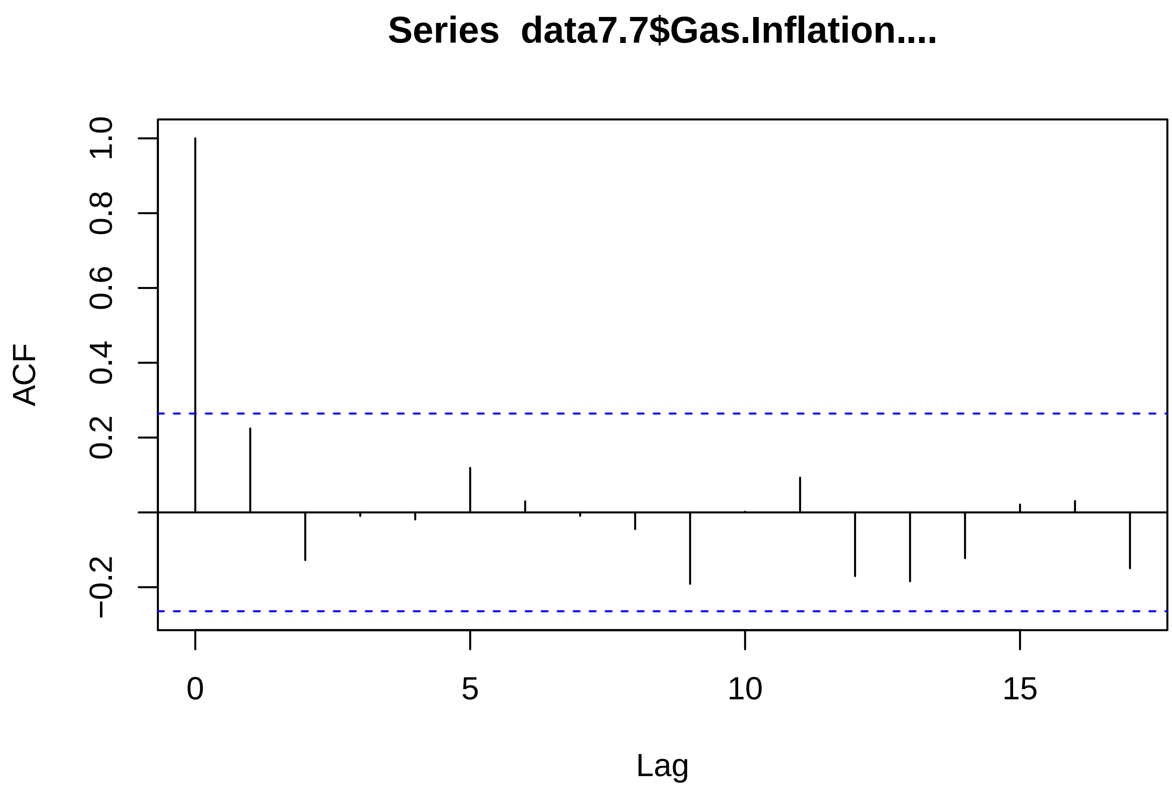
```
data7.7 <- read.table(file = 'C:\\Users\\Austin\\Documents\\R\\Exercise7.7.csv',
                      header = T, sep = ',')
ts_food<-ts(data7.7$food.Inflation...., start = 1968, freq=1)
ts_gas <-ts(data7.7$Gas.Inflation...., start = 1968, freq =1)
acf(data7.7$food.Inflation...., na.action = na.pass)
```



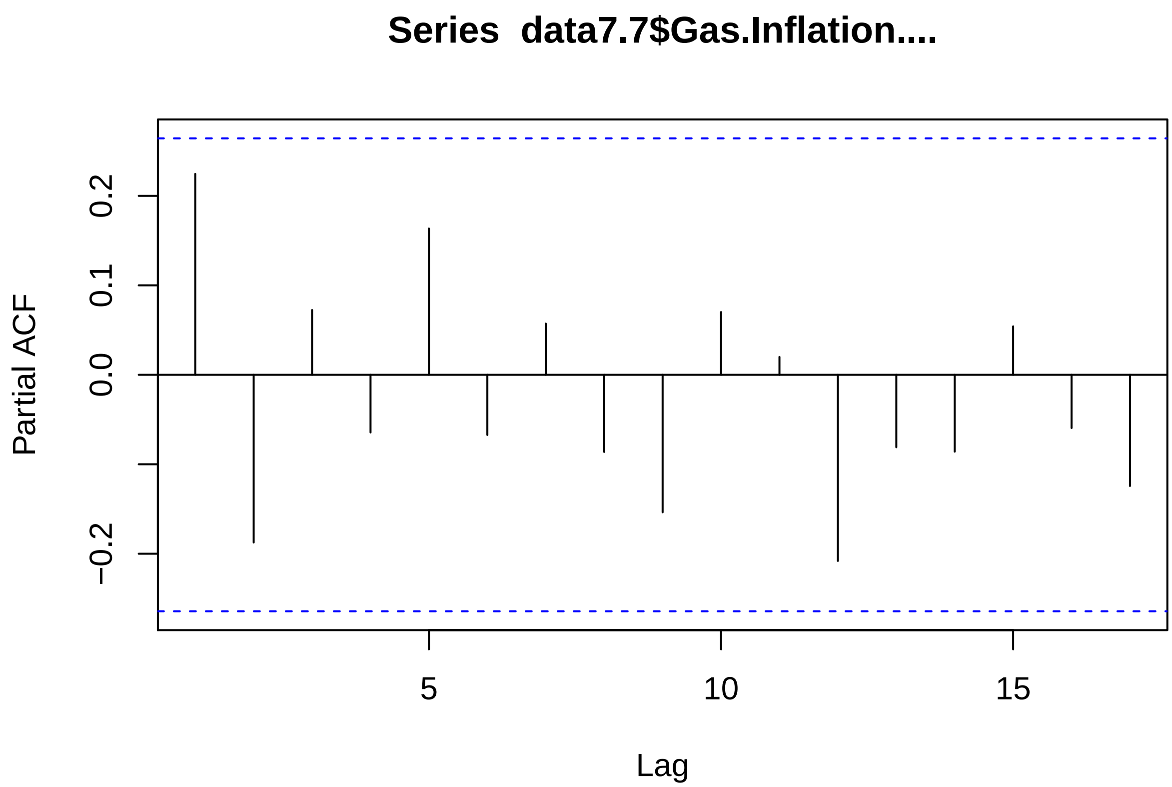

```
pacf(data7.7$food.Inflation...., na.action = na.pass)
```




```
acf(data7.7$Gas.Inflation...., na.action = na.pass)
```




```r
pacf(data7.7$Gas.Inflation....,na.action = na.pass)
```




```r
#food inflation ARMA(1,1)
#gas inflation arima(1,0,0)
best_fit_food<-Arima(ts_food,order = c(1,0,1))
best_fit_gas<-Arima(ts_gas, order = c(1,0,0))



food_predict1 <- predict(best_fit_food, n.ahead =3)
print(paste("Point H-1, H-2, H-3 for Food:",food_predict1[1]))
```

```
## [1] "Point H-1, H-2, H-3 for Food: c(4.59718438613192, 4.24449428096977, 4.11368625244479)"
```

```r
print(paste("Forecast Error at H-1, H-2, H-3 for Food", food_predict1[2]))
```

```
## [1] "Forecast Error at H-1, H-2, H-3 for Food c(2.29942031908443, 3.10430682879447, 3.19922004736924
```

```r
food_predict1$se[3:3]*food_predict1$se[3:3]
```

```
## [1] 10.23501
```

```r
uncertainty_holder = vector()
for (i in 1:3){
  uncertainty_holder[i]=food_predict1$se[i]*food_predict1$se[i]
}
print(paste("Uncertainty H-1,2 and 3 for Food", uncertainty_holder))
```

```
## [1] "Uncertainty H-1,2 and 3 for Food 5.28733380381834"
## [2] "Uncertainty H-1,2 and 3 for Food 9.63672088730001"
## [3] "Uncertainty H-1,2 and 3 for Food 10.2350089114892"
```

```r
gas_predict1 <- predict(best_fit_gas, n.ahead=3)
print(paste("Point H-1, 2, and 3 for Gas:",gas_predict1[1]))
```

```
## [1] "Point H-1, 2, and 3 for Gas: c(10.4823385837972, 6.7431946932554, 5.86741366164741)"
```

```r
print(paste("Forecast Error at H-1, H-2, H-3 for Gas:",gas_predict1[2]))
```

```
## [1] "Forecast Error at H-1, H-2, H-3 for Gas: c(12.1839004219495, 12.5136359631604, 12.5314738402831
```

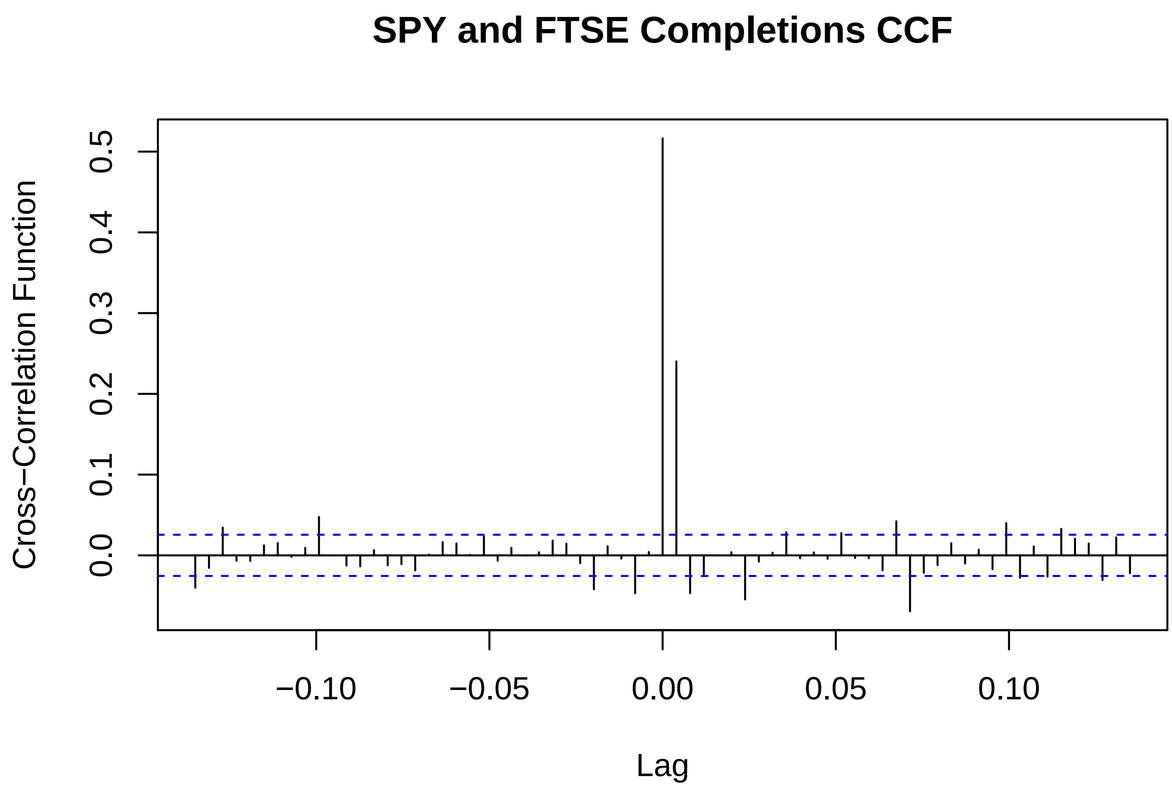
```r
uncertainty_holder1 = vector()
for (i in 1:3){
  uncertainty_holder1[i]=gas_predict1$se[i]*gas_predict1$se[i]
}
print(paste("Uncertainty H-1, 2 and 3 for Gas", uncertainty_holder1))
```

```
## [1] "Uncertainty H-1, 2 and 3 for Gas 148.447429491981"
## [2] "Uncertainty H-1, 2 and 3 for Gas 156.591085018502"
## [3] "Uncertainty H-1, 2 and 3 for Gas 157.037836609699"
```

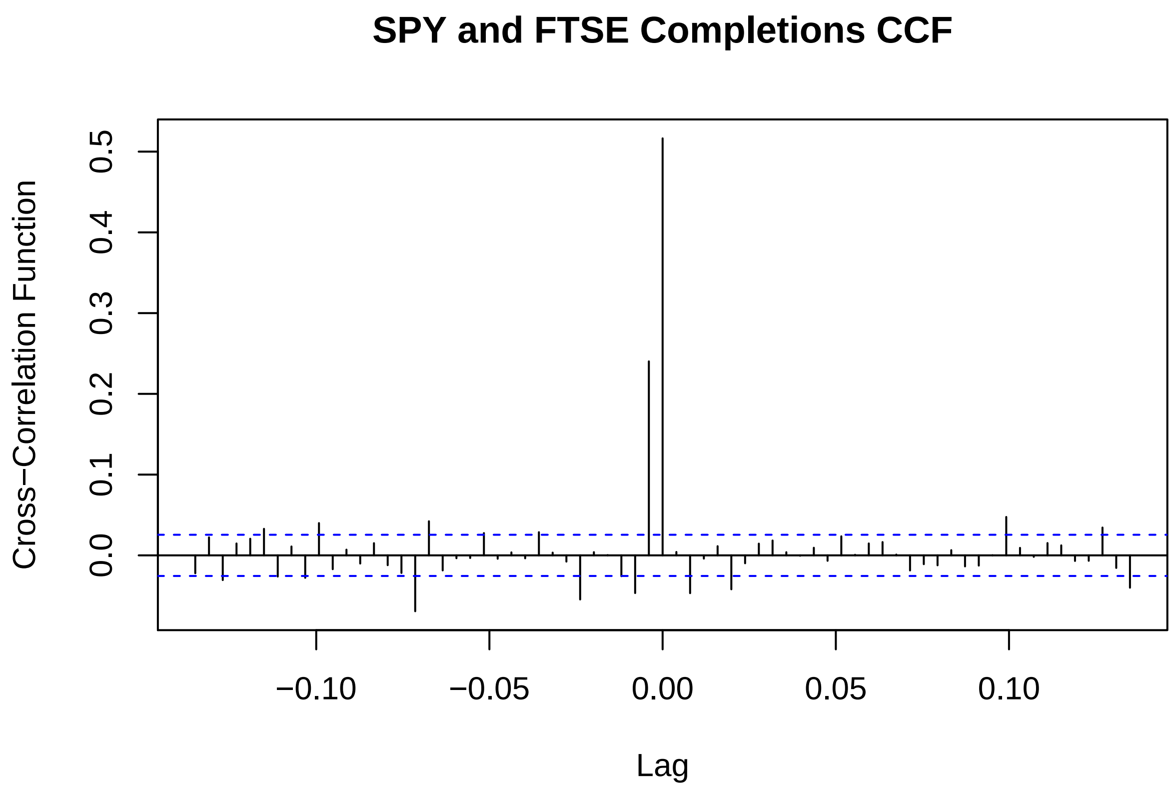
It is clear that there is far greater uncertainty for gas inflation than there is for food. This is due to the uncertainty coefficient being much larger than the uncertainty coefficient for food.

Problem 5

```r
data_SPYFTSE <- read.table(file = 'C:\\Users\\Austin\\Documents\\R\\Exercise8.7.csv',
                           header = T, sep = ',')

ts_returns_SPY <- ts(data_SPYFTSE$R_SP500_OPEN...., start = 1990+(1/252), freq = 252)
ts_returns_FTSE <- ts(data_SPYFTSE$R_FTSE_OPEN...., start = 1990+(1/252), freq = 252)

ccf(ts_returns_FTSE, ts_returns_SPY, ylab = "Cross-Correlation Function",
    main = "SPY and FTSE Completions CCF", na.action = na.pass)
```




```r
ccf( ts_returns_SPY,ts_returns_FTSE, ylab = "Cross-Correlation Function",
    main = "SPY and FTSE Completions CCF", na.action = na.pass)
```




```r
#Look at CCF()
```

The ccf function computes the sample cross correlation of SPY and FTSE returns, and identifies lags in the FTSE that might be helpful for predicting the SPY. We see that the ccf plot is skewed left, indicating that there is potentially some type of casual relationship within the first few hours of trading.

```r
library(vars)
```

```
## Warning: package 'vars' was built under R version 3.5.2
## Loading required package: MASS
## Loading required package: urca
## Warning: package 'urca' was built under R version 3.5.2
## Loading required package: lmtest
## Warning: package 'lmtest' was built under R version 3.5.2
```

```r
library('tseries')
```

```
## Warning: package 'tseries' was built under R version 3.5.2
```

```r
combined_SPY_FTSE = na.remove(cbind(ts_returns_SPY,ts_returns_FTSE))
tot_combined_SPY_FTSE<-data.frame(combined_SPY_FTSE)
VARselect(tot_combined_SPY_FTSE)
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      8      7      3      8
```

```
##
## $criteria
##                1         2         3         4         5         6
## AIC(n) 0.1772849 0.1679881 0.1612826 0.1607612 0.1581608 0.1569128
## HQ(n)  0.1798678 0.1722930 0.1673094 0.1685100 0.1676315 0.1681055
## SC(n)  0.1846791 0.1803119 0.1785359 0.1829440 0.1852731 0.1889546
## FPE(n) 1.1939712 1.1829225 1.1750170 1.1744045 1.1713546 1.1698937
##                7         8         9        10
## AIC(n) 0.1536027 0.1530978 0.1543607 0.1547620
## HQ(n)  0.1665173 0.1677343 0.1707192 0.1728424
## SC(n)  0.1905740 0.1949986 0.2011911 0.2065218
## FPE(n) 1.1660276 1.1654390 1.1669118 1.1673802
```

```
var_model <- VAR(tot_combined_SPY_FTSE, p = 3)
```

We will chose VAR(p=3)because the because BIC has the lowest value and ignore AIC because it overparameterizes the models.

```
summary(var_model)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: ts_returns_SPY, ts_returns_FTSE
## Deterministic variables: const
## Sample size: 5349
## Log Likelihood: -15596.258
## Roots of the characteristic polynomial:
## 0.477 0.4235 0.4235 0.2566 0.2566 0.08491
## Call:
## VAR(y = tot_combined_SPY_FTSE, p = 3)
##
##
## Estimation results for equation ts_returns_SPY:
## ===============================================
## ts_returns_SPY = ts_returns_SPY.l1 + ts_returns_FTSE.l1 + ts_returns_SPY.l2 + ts_returns_FTSE.l2 + ts
##
##                     Estimate Std. Error t value Pr(>|t|)
## ts_returns_SPY.l1  -0.056376   0.016442  -3.429 0.000611 ***
## ts_returns_FTSE.l1  0.033280   0.016718   1.991 0.046572 *
## ts_returns_SPY.l2  -0.054705   0.017715  -3.088 0.002025 **
## ts_returns_FTSE.l2 -0.007859   0.017259  -0.455 0.648866
## ts_returns_SPY.l3  -0.002407   0.017224  -0.140 0.888857
## ts_returns_FTSE.l3 -0.004978   0.015988  -0.311 0.755543
## const               0.025614   0.015771   1.624 0.104421
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1.152 on 5342 degrees of freedom
## Multiple R-Squared: 0.004841,    Adjusted R-squared: 0.003723
## F-statistic: 4.331 on 6 and 5342 DF,  p-value: 0.0002293
##
##
## Estimation results for equation ts_returns_FTSE:
```

```
## =================================================
## ts_returns_FTSE = ts_returns_SPY.l1 + ts_returns_FTSE.l1 + ts_returns_SPY.l2 + ts_returns_FTSE.l2 + t
##
##                     Estimate Std. Error t value Pr(>|t|)
## ts_returns_SPY.l1   0.372476   0.016121  23.104  < 2e-16 ***
## ts_returns_FTSE.l1 -0.257130   0.016392 -15.686  < 2e-16 ***
## ts_returns_SPY.l2   0.070639   0.017370   4.067 4.84e-05 ***
## ts_returns_FTSE.l2 -0.093963   0.016922  -5.553 2.95e-08 ***
## ts_returns_SPY.l3   0.052880   0.016888   3.131  0.00175 **
## ts_returns_FTSE.l3 -0.089301   0.015676  -5.697 1.29e-08 ***
## const              -0.002131   0.015464  -0.138  0.89038
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1.13 on 5342 degrees of freedom
## Multiple R-Squared: 0.09925, Adjusted R-squared: 0.09824
## F-statistic: 98.11 on 6 and 5342 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##                 ts_returns_SPY ts_returns_FTSE
## ts_returns_SPY           1.328           0.724
## ts_returns_FTSE          0.724           1.277
##
## Correlation matrix of residuals:
##                 ts_returns_SPY ts_returns_FTSE
## ts_returns_SPY           1.000           0.556
## ts_returns_FTSE          0.556           1.000
```

When forecasting SPY, it appears that after lag1, FTSE has statistically insignificant coefficients that render the model. On the otherhand, when forecasting FTSE returns, all of the coefficients are statistically significant, meaning that it is possible that SPY and FTSE returns are a good predicter of FTSE returns.

```r
library(lmtest)
grangertest(ts_returns_SPY~ts_returns_FTSE, order = 3)
```

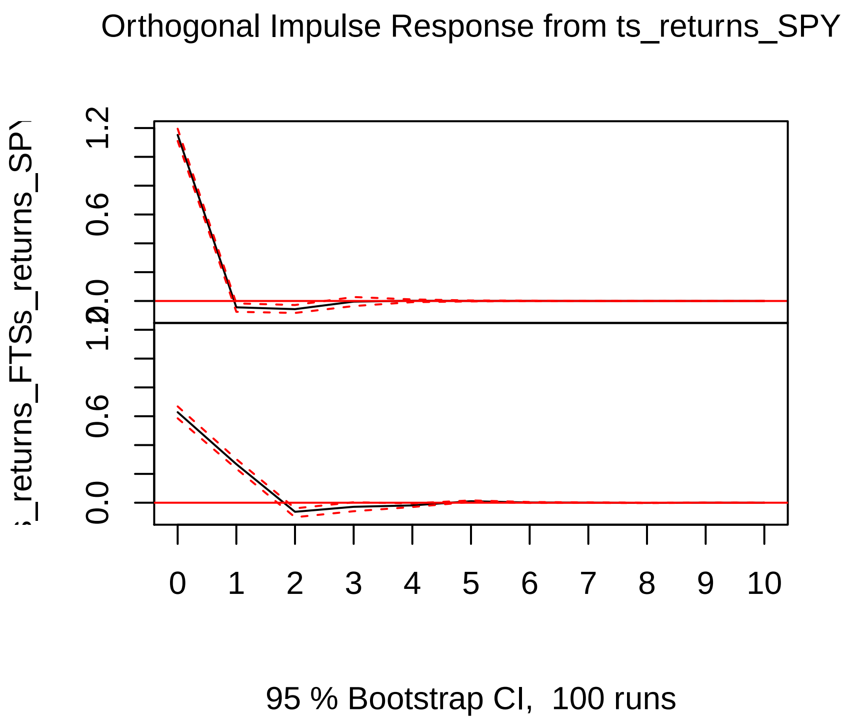
```
## Granger causality test
##
## Model 1: ts_returns_SPY ~ Lags(ts_returns_SPY, 1:3) + Lags(ts_returns_FTSE, 1:3)
## Model 2: ts_returns_SPY ~ Lags(ts_returns_SPY, 1:3)
##   Res.Df Df      F Pr(>F)
## 1   5342
## 2   5345 -3 1.6741 0.1703
```

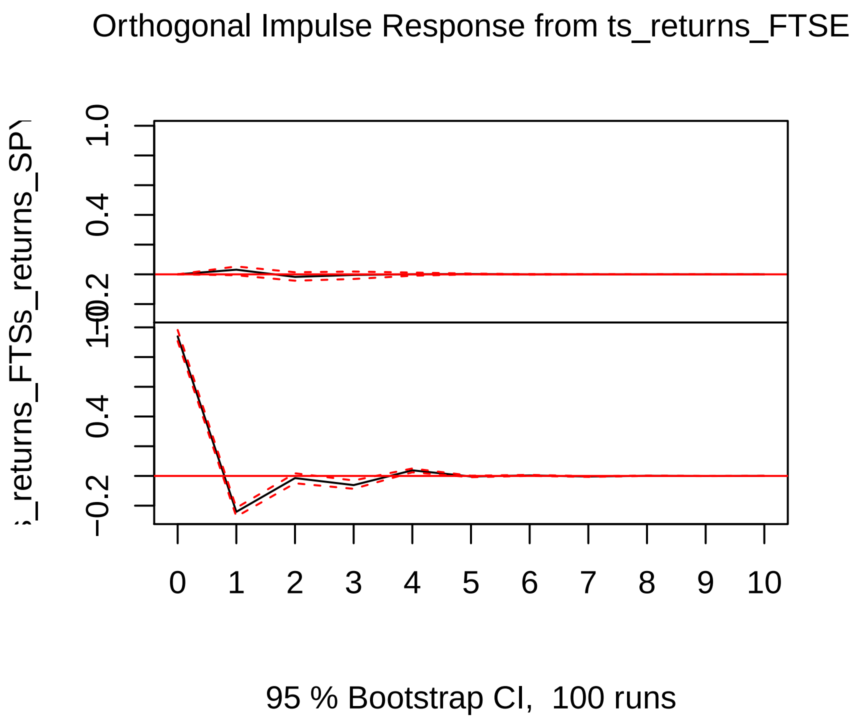
```r
grangertest(ts_returns_FTSE~ts_returns_SPY, order = 3)# SPY can predict FTSE
```

```
## Granger causality test
##
## Model 1: ts_returns_FTSE ~ Lags(ts_returns_FTSE, 1:3) + Lags(ts_returns_SPY, 1:3)
## Model 2: ts_returns_FTSE ~ Lags(ts_returns_FTSE, 1:3)
##   Res.Df Df      F    Pr(>F)
## 1   5342
## 2   5345 -3 179.81 < 2.2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
plot(irf(var_model))
```





The granger preditive test reinforces the idea that SPY returns may help predict FTSE returns. When looking at the IRF, we see the impact of returns from FTSE on SPY, we see that its insignificant. However, when we look at SPY for FTSE, we see that it has high predictive capabilities up until, the second lag.