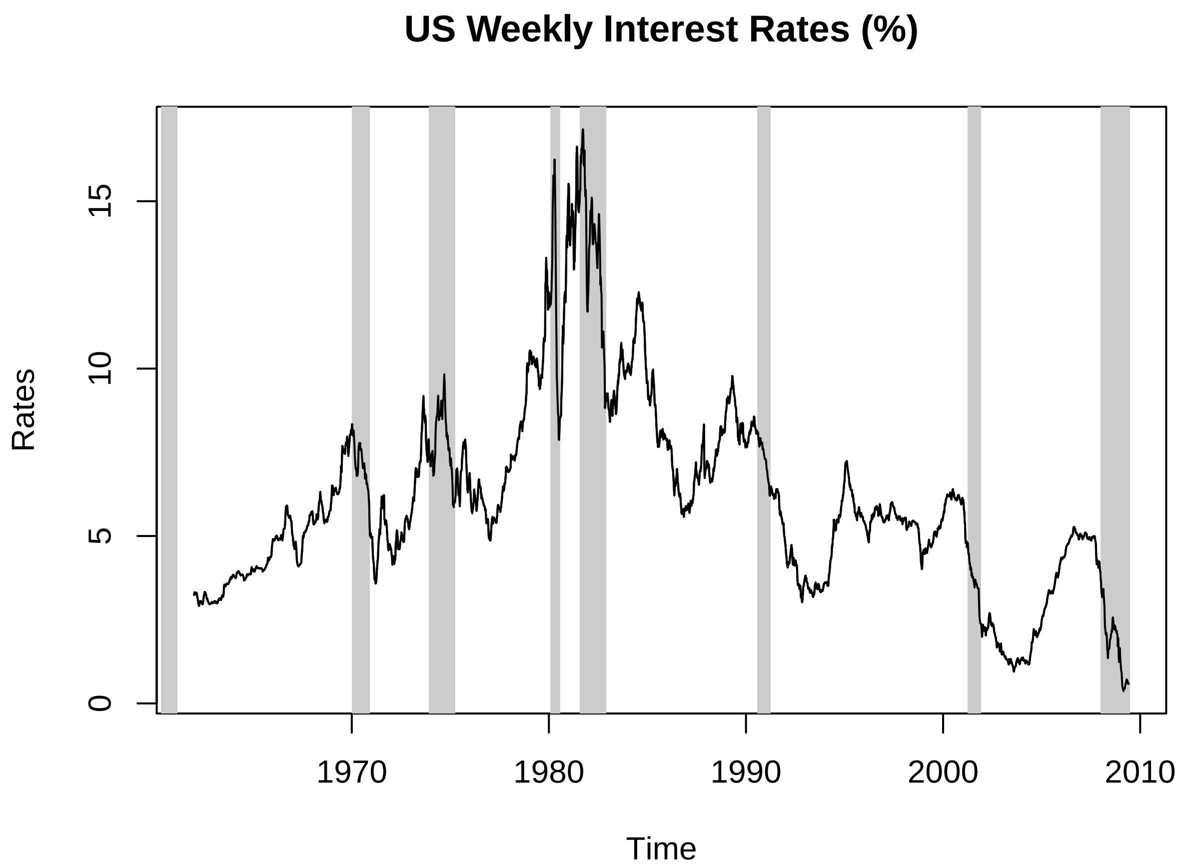
Econ 144: Homework #4

Problem 1.

1a. Show a plot of the data, along with the respective ACF and PACF functions.

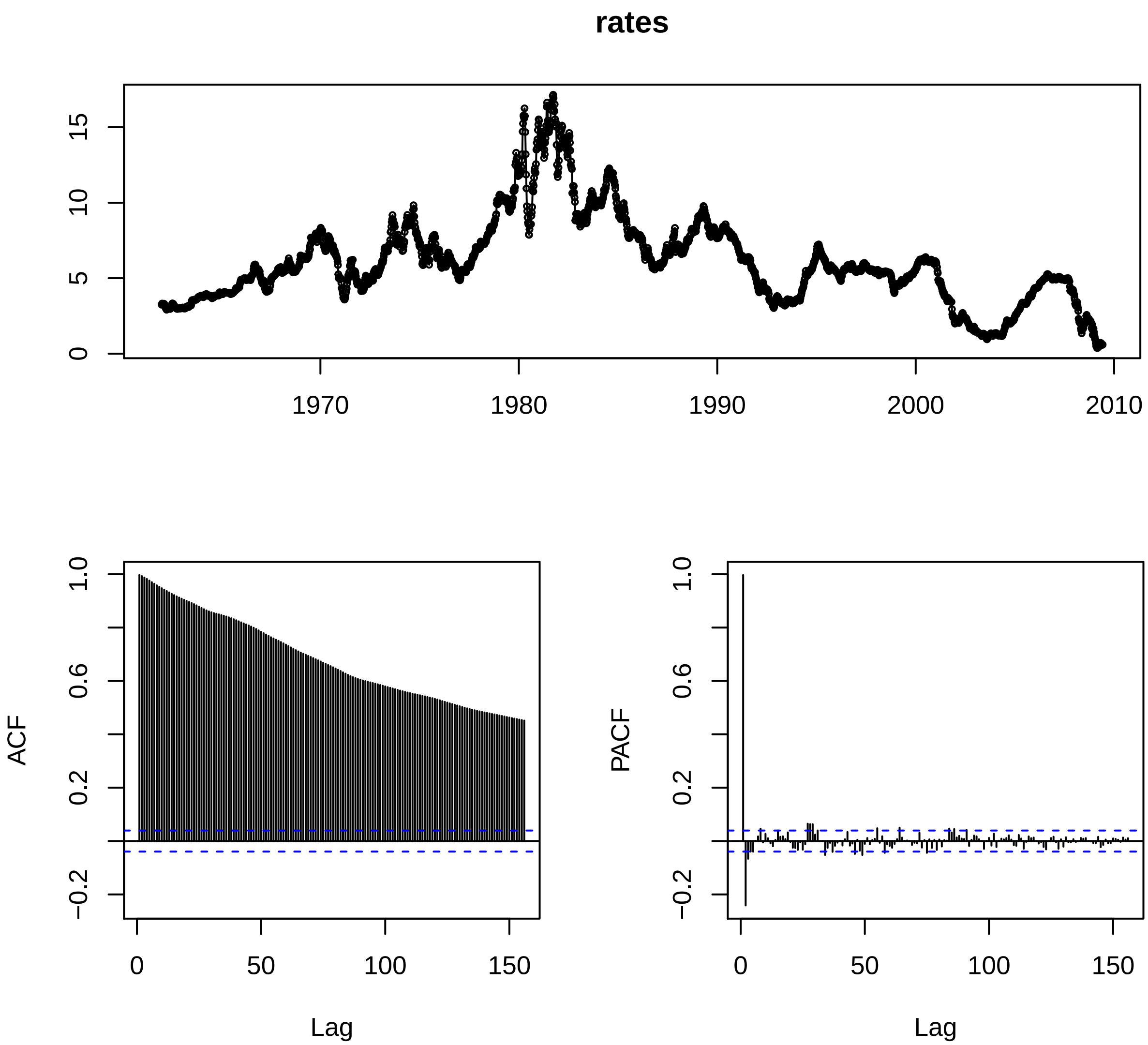
There appears to be some degree of persistence in the data; interest rates appear to stall at various levels for some time before dropping or jumping up to a new level. Additionally, changes in interest rates appear to precede recessions.

```
quartz()
plot(t, rates, main = "US Weekly Interest Rates (%)", ylab = "Rates",
    xlab = "Time", type = "l")
nberShade()
lines(rates)
```

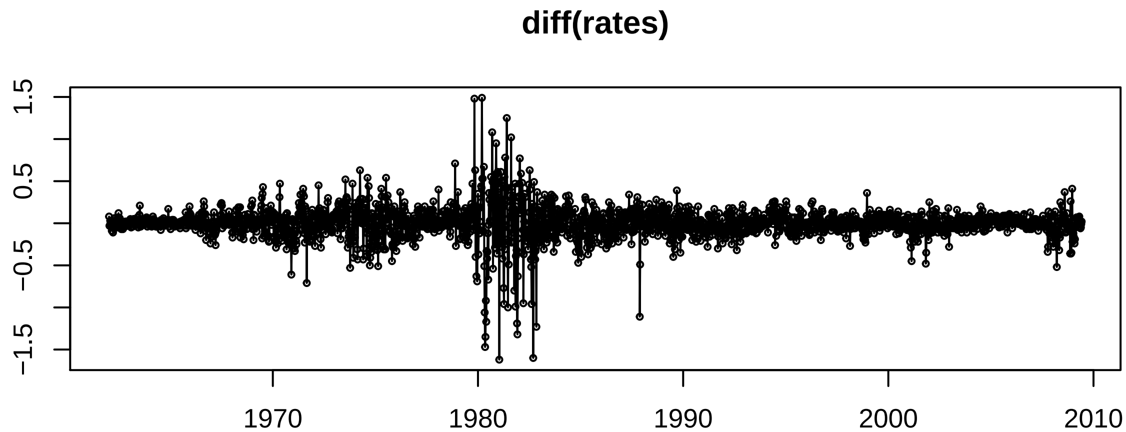


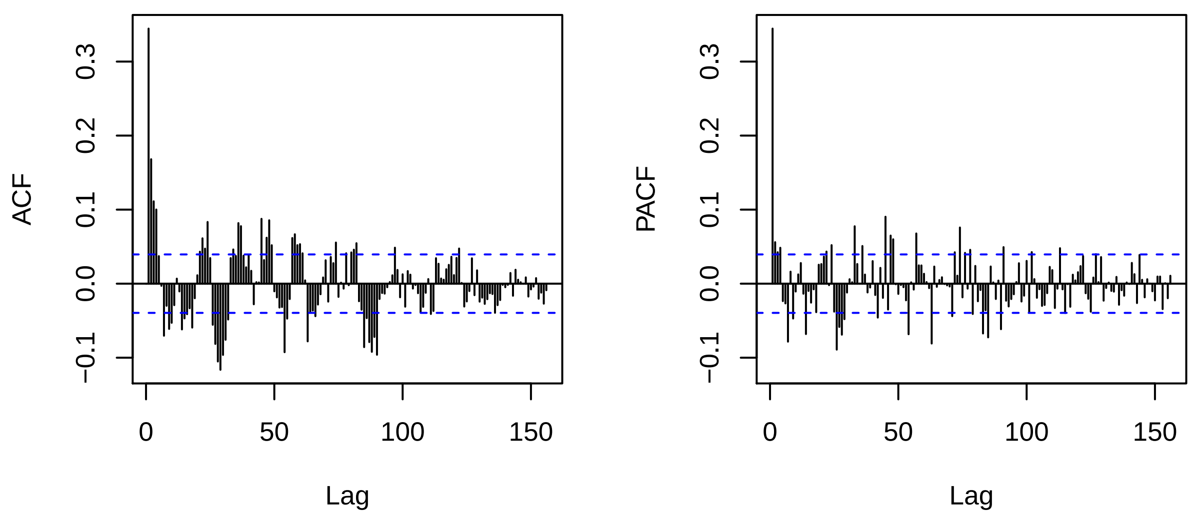
With the ACF's and PACF's, we observe 2 significant spikes in PACF and a dampening effect in the ACF; however, the spikes appear close to 1, which may indicate that the series is not covariance stationary, and that we will need to take the first difference to successfully implement an ARMA model.

```
tsdisplay(rates)
```




```
# Looking at the first difference:
tsdisplay(diff(rates))
```





We see 1 significant spike in the PACF, and the ACF dampening a lot quicker in the first difference. This could indicate an AR(1) process, though there appears to be some more complex seasonal components as well.

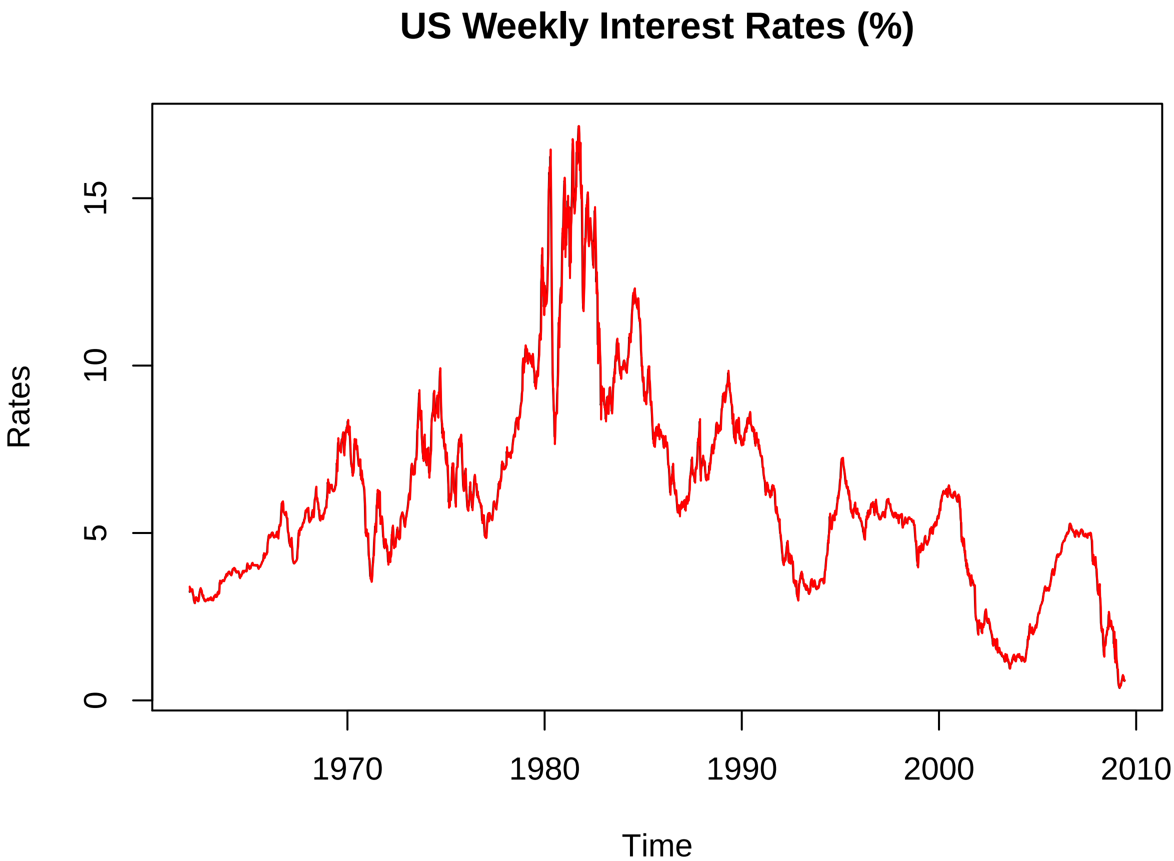
1b. ARMA(p,q)

We try 3 different fits: AR(2), MA(10), and an AR(1) with an integrated factor of 1 (ARIMA(1,1,0)). The AR(2) was selected from looking at the ACF/PACF plots above, while the ARIMA(1,1,0) was selected from the first differenced ACF/PACF plots. MA(10) was chosen to see whether or not a moving average of large degree could also provide a relatively good fit, as opposed to a lower order AR model.

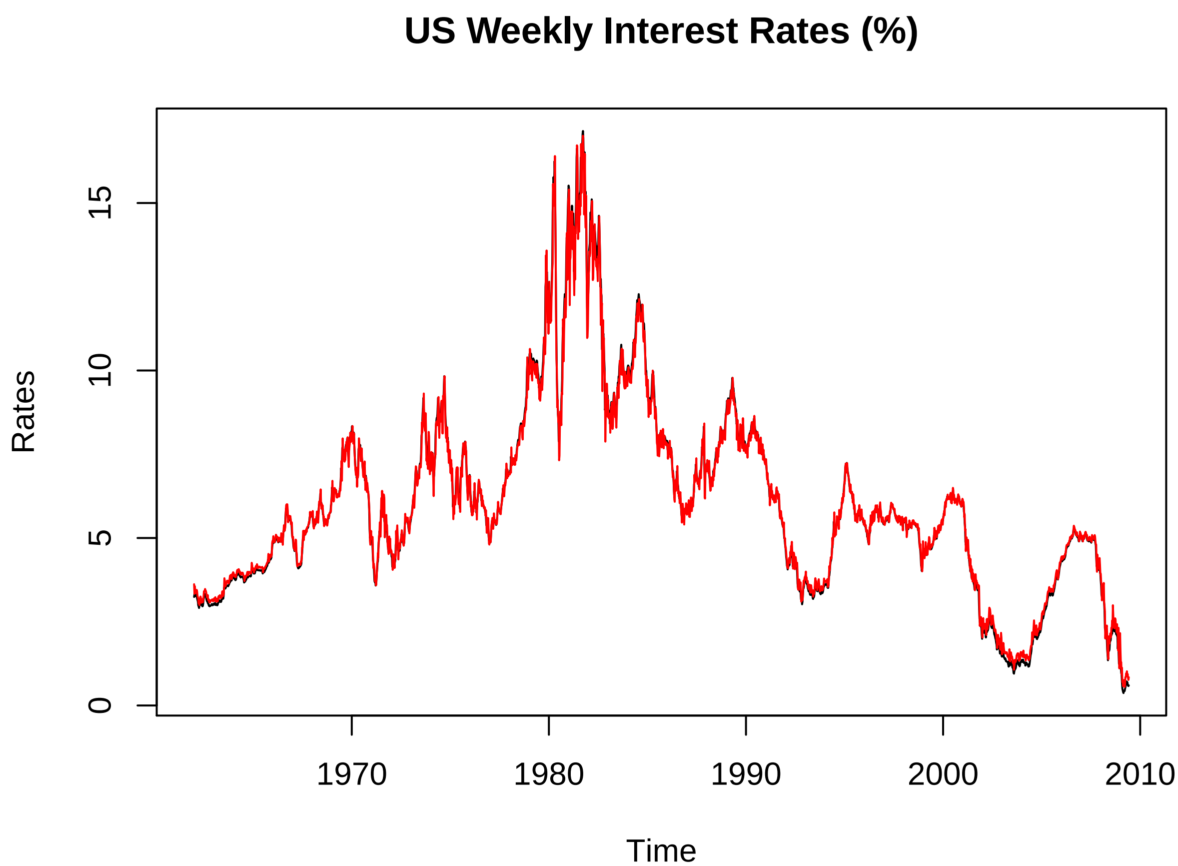
AIC/BIC converge to select the 3rd model (ARIMA(1,1,0)).

```
fit1 <- Arima(rates, order = c(2, 0, 0))
quartz()
plot(t, rates, main = "US Weekly Interest Rates (%)", ylab = "Rates",
```

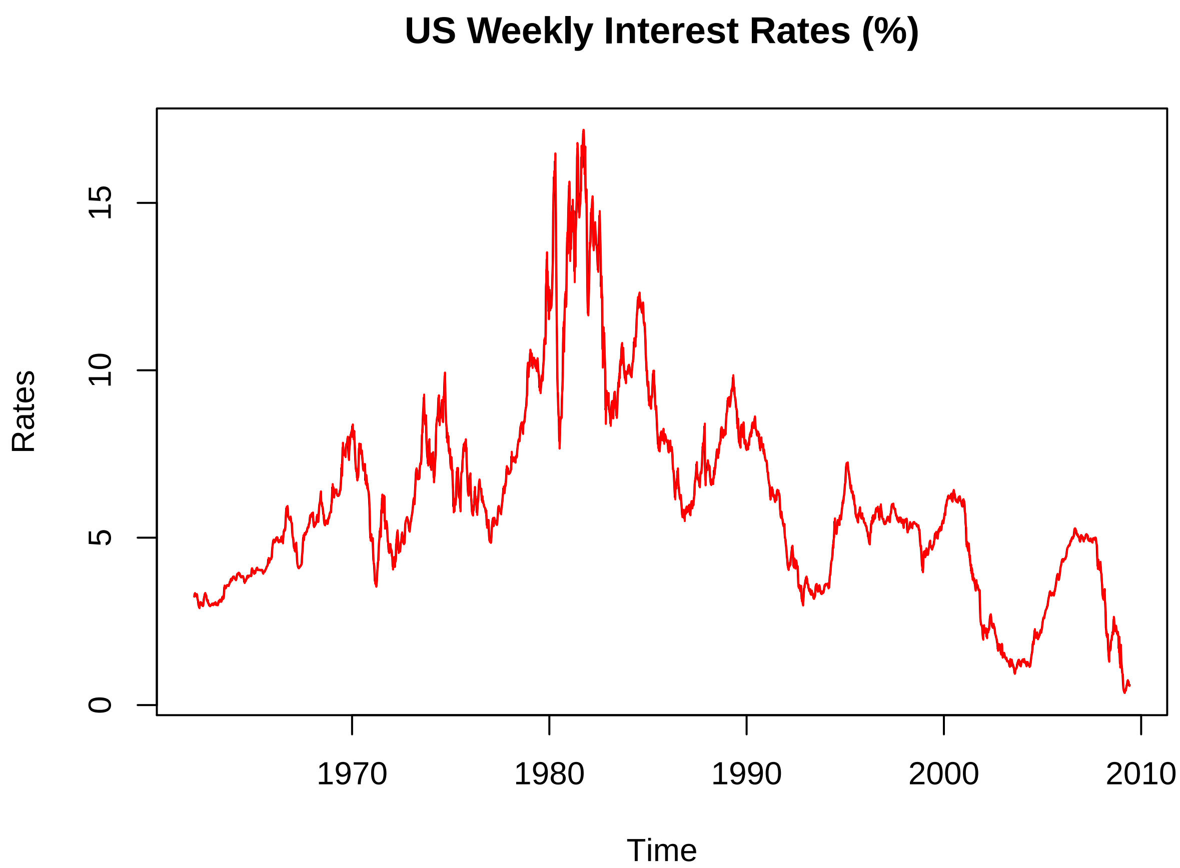
```
    xlab = "Time", type = "l")
lines(fit1$fitted, col = "red")
```




```
fit2 <- Arima(rates, order = c(0, 0, 10))
quartz()
plot(t, rates, main = "US Weekly Interest Rates (%)", ylab = "Rates",
    xlab = "Time", type = "l")
lines(fit2$fitted, col = "red")
```




```
fit3 <- Arima(rates, order = c(1, 1, 0))
quartz()
plot(t, rates, main = "US Weekly Interest Rates (%)", ylab = "Rates",
    xlab = "Time", type = "l")
lines(fit3$fitted, col = "red")
```




```
AIC(fit1, fit2, fit3)
```

```
##      df         AIC
## fit1  4 -1514.43626
## fit2 12    28.91915
## fit3  2 -1519.33508
```

```
BIC(fit1, fit2, fit3)
```

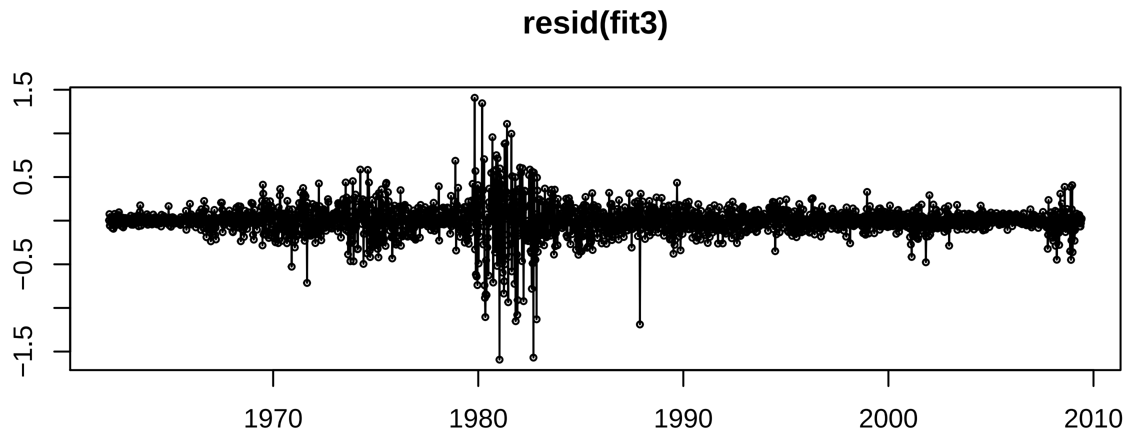
```
##      df         BIC
## fit1  4 -1491.19322
## fit2 12    98.64825
## fit3  2 -1507.71437
```

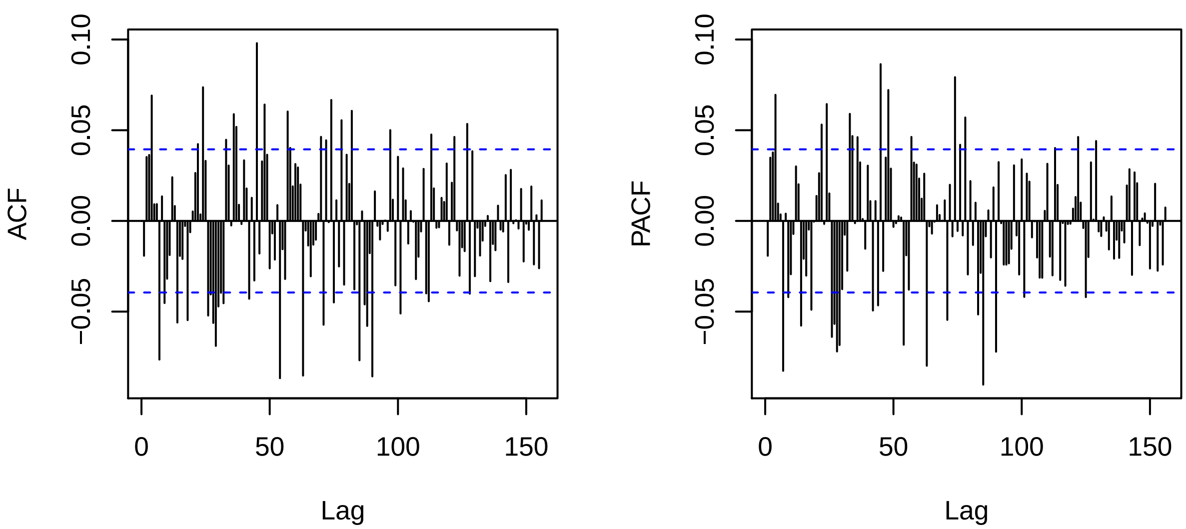
All the plots and the corresponding fits appear to be pretty good, at least qualitatively. We see that AIC/BIC pick the model that is consistent with our hypothesis for the model parameters based on looking at the ACF/PACF plots.

1c. Residuals

There appears to still be some structure left in the residuals, though the spikes are pretty small; we would have to conduct tests to check for the significance of the spikes.

```
tsdisplay(resid(fit3))
```

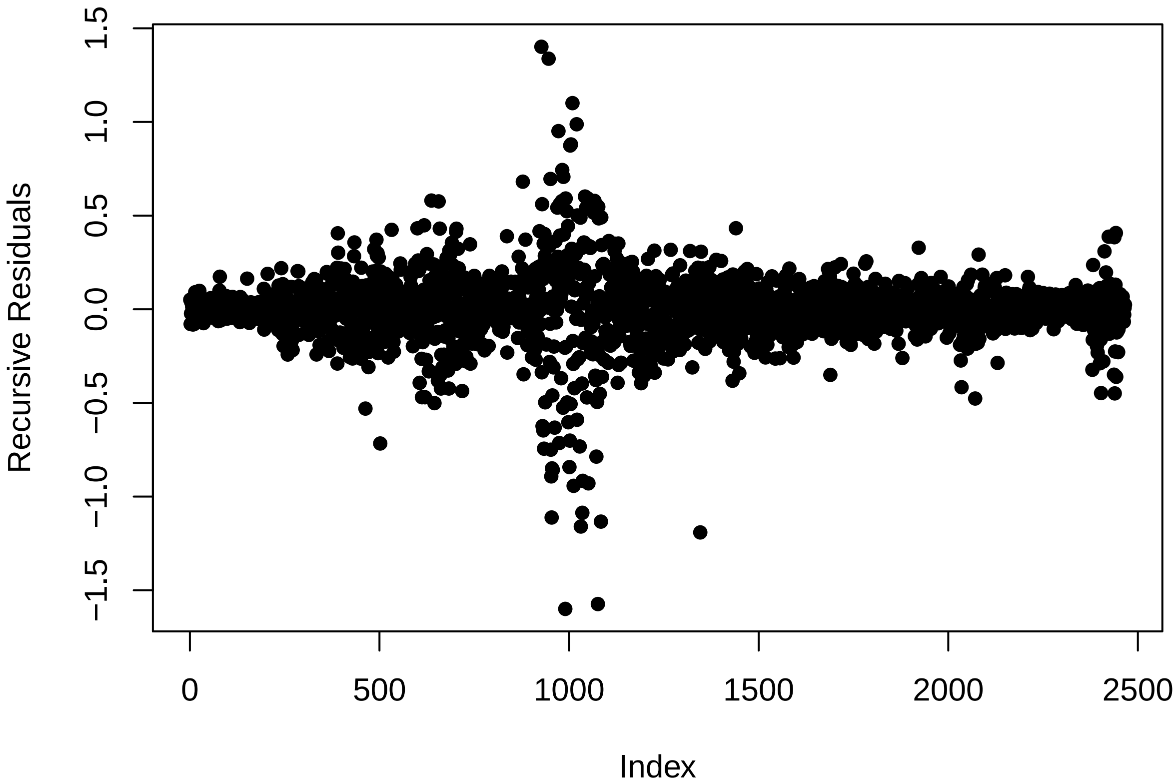




1d. Recursive residuals

We see that the model has some struggles forecasting in the 1980's and around 2008. This corresponds to recessions and periods of economic instability. We may need to add in a dummy variable to account for these time periods.

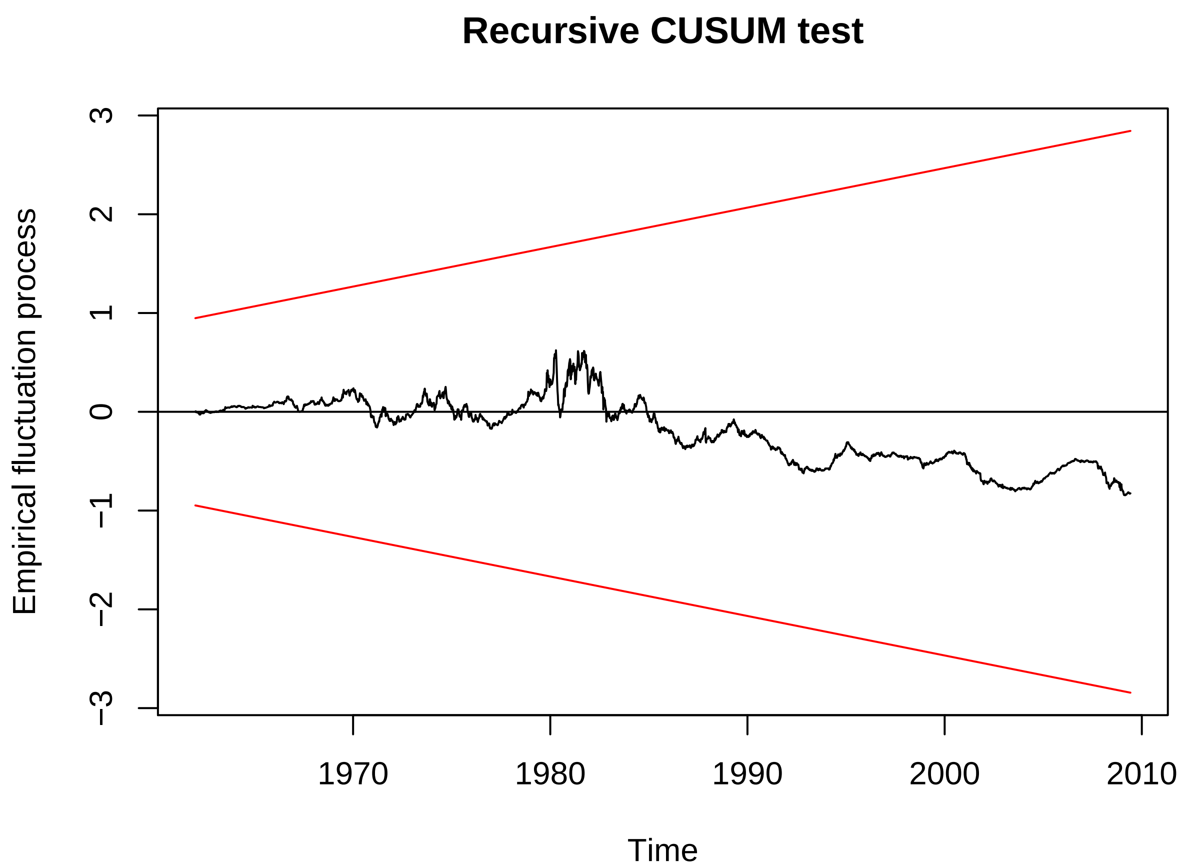
```
y1 <- recresid(fit3$res ~ 1)
quartz()
plot(y1, pch = 16, ylab = "Recursive Residuals")
```



1e. CUSUM

The model appears to be structurally fine; everything is within the two bands, which may indicate that the potential structural breaks we observed in part (d) are not technically statistically significant.

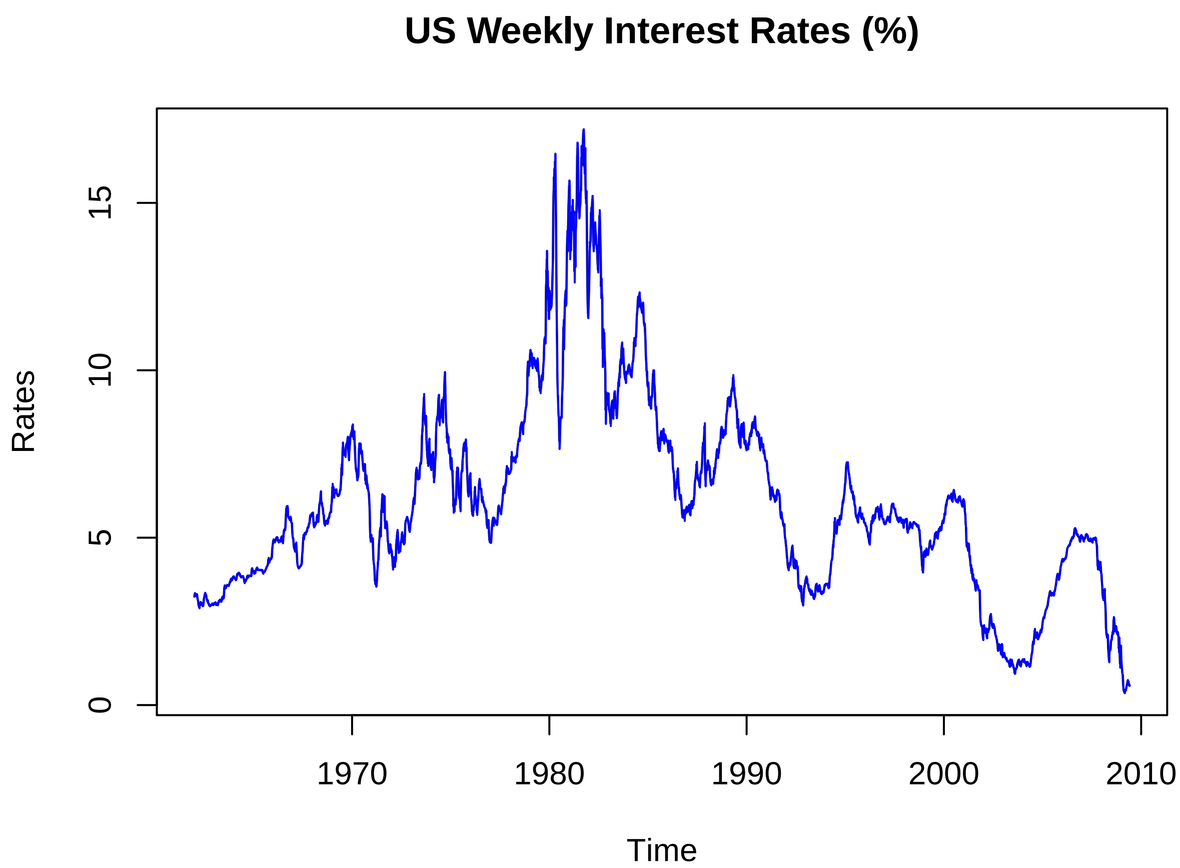
```
quartz()
plot(efp(fit3$res ~ 1, type = "Rec-CUSUM"))
```



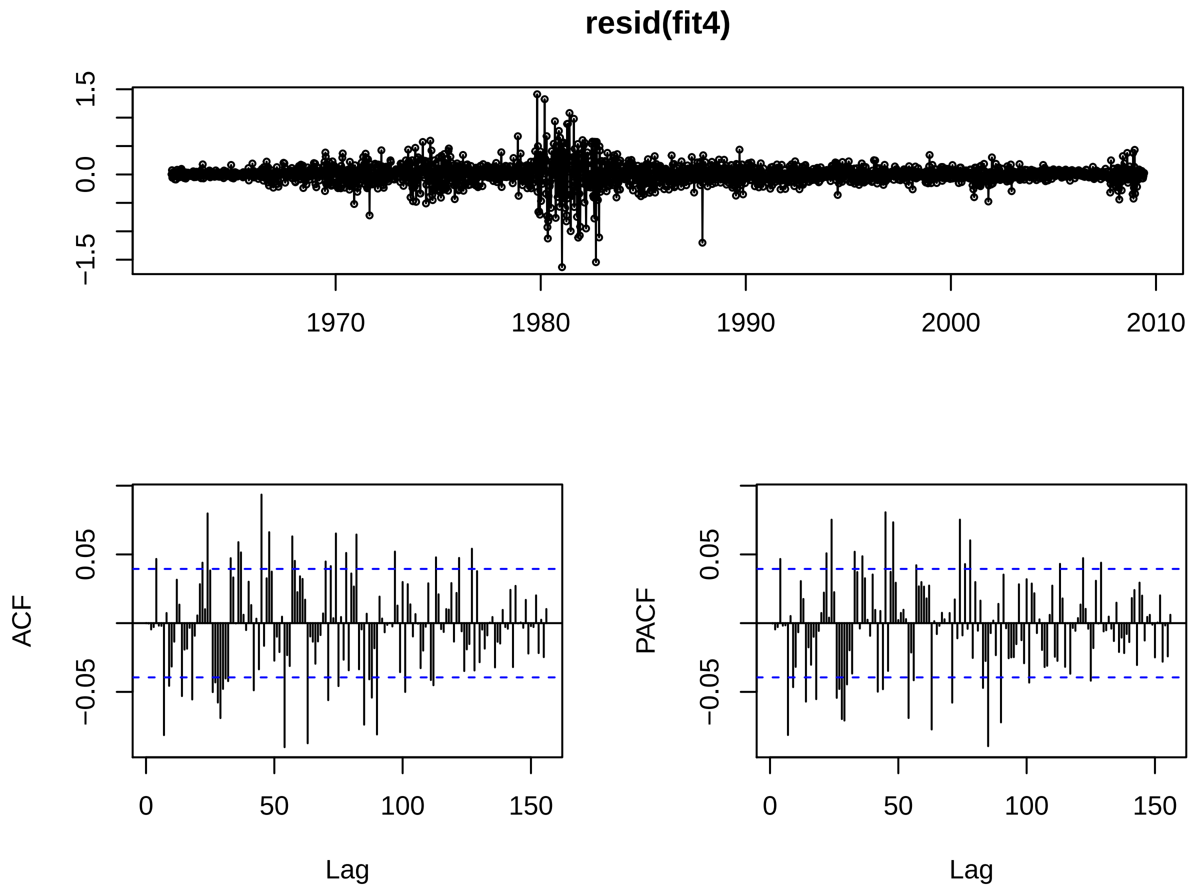
1f. Best fit model from R

Using `auto.arima`, we find that the 'best fit' is an ARIMA(1,1,2) model. The results are pretty similar to those found with the model implemented earlier. Once again, we observe that the recursive residuals show that the model has a tough time accounting for the dynamics around the 1980's and 2008.

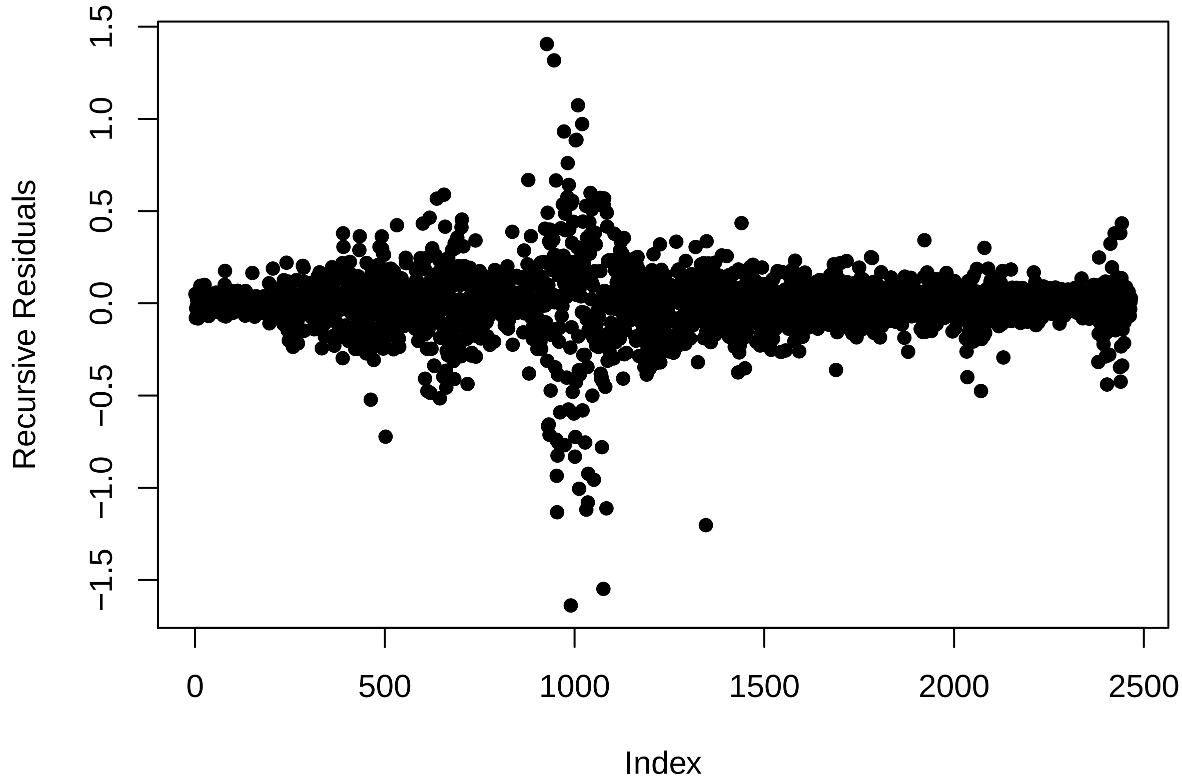
```
fit4 <- auto.arima(rates)

quartz()
plot(t, rates, main = "US Weekly Interest Rates (%)", ylab = "Rates",
    xlab = "Time", type = "l")
lines(fit4$fitted, col = "blue")
```



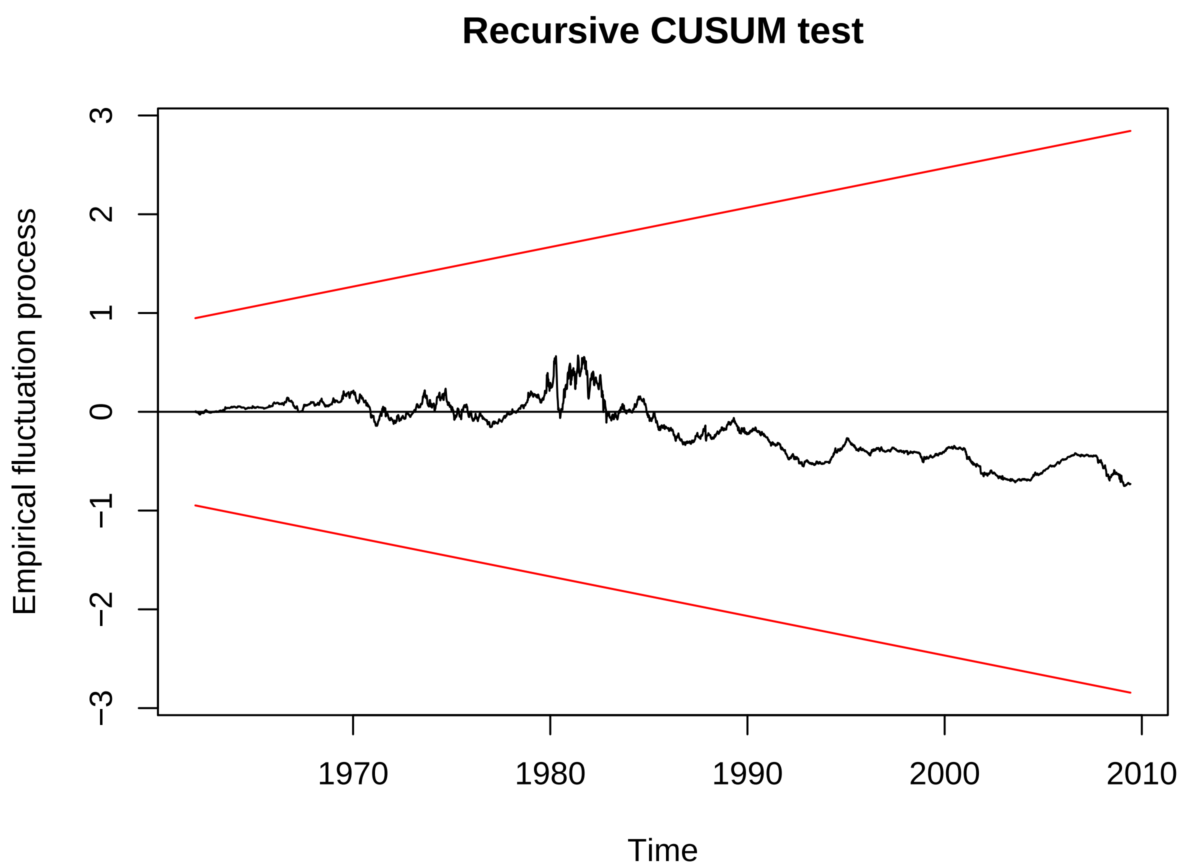

```
quartz()
tsdisplay(resid(fit4))
```




```
y2 <- recresid(fit4$res ~ 1)
quartz()
plot(y2, pch = 16, ylab = "Recursive Residuals")
```



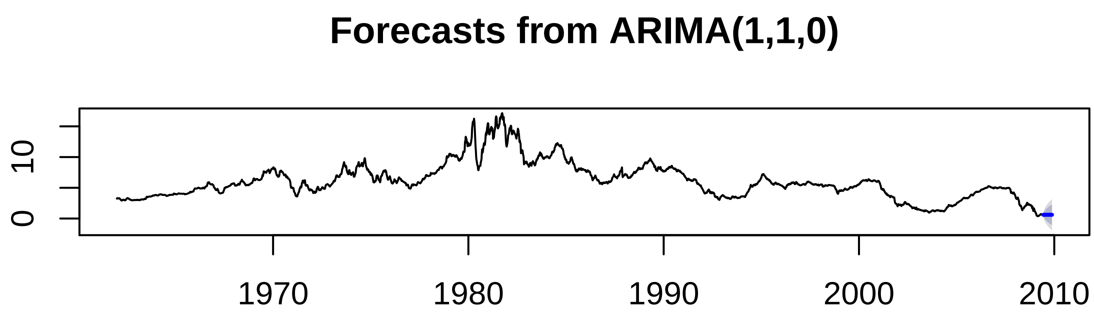

```
quartz()
plot(efp(fit4$res ~ 1, type = "Rec-CUSUM"))
```

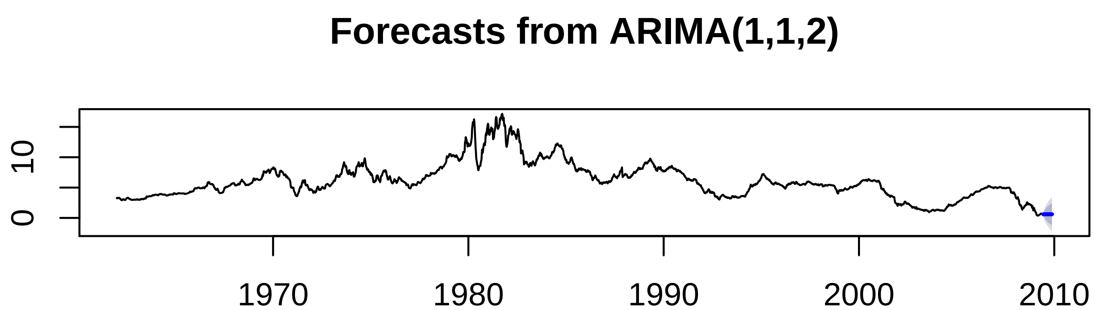


1g. Forecast 24 steps ahead.

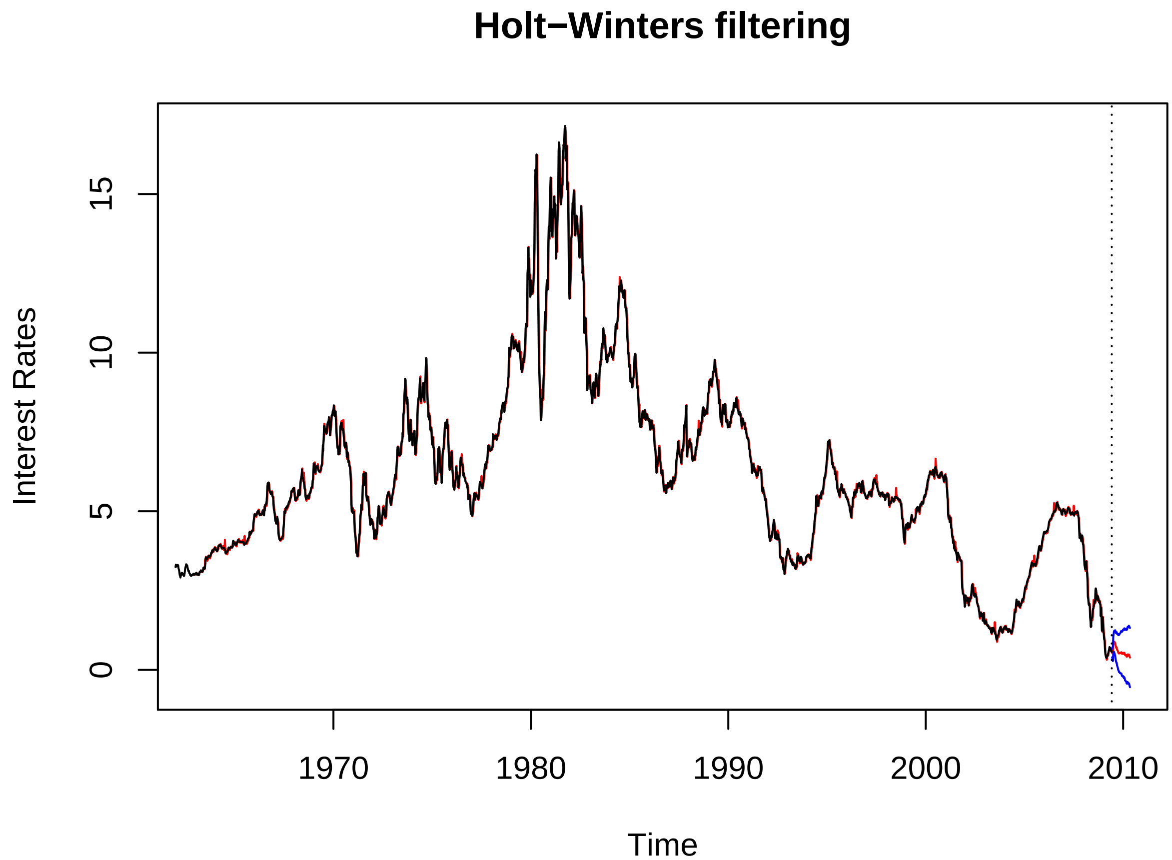
The two forecasts produced by ARIMA(1,1,0) and ARIMA(1,1,2) (`auto.arima`'s fit) both appear very similar; however, they indicate that the interest rate levels will remain flat/constant for the next 24 steps. This does not seem consistent with economic theory. Running a Holt-Winters forecast on the data, it appears that the model is able to capture more of the dynamics within interest rates. It shows that interest rates will decline for the next few periods; this is consistent with what we now know occurred during 2010, in which the Federal Reserve lowered interest rates in an attempt to boost the economy.

```
quartz()
par(mfrow = c(2, 1))
plot(forecast(fit3, h = 24))
plot(forecast(fit4, h = 24))
```



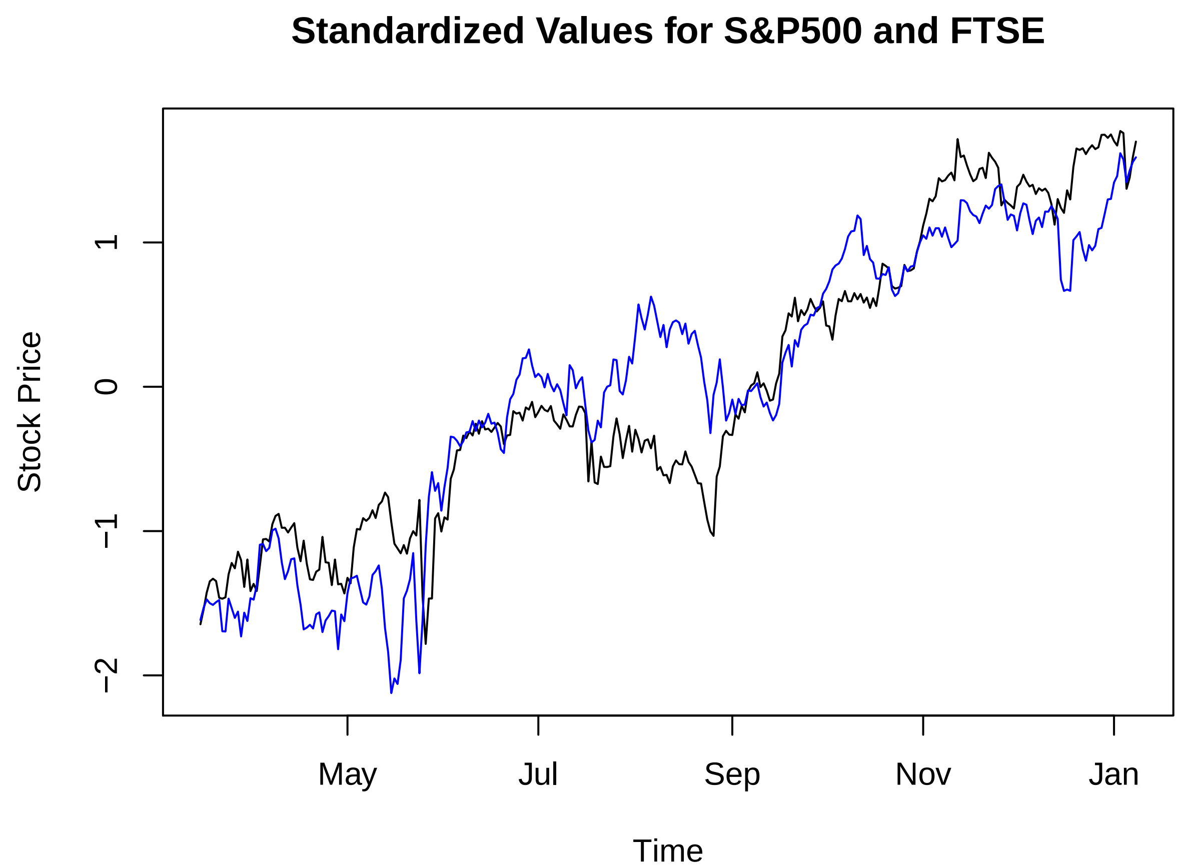



```
r.hw <- HoltWinters(rates)
quartz()
r.hw.pred <- predict(r.hw, 48, prediction.interval = TRUE, level = 0.5)
plot(r.hw, r.hw.pred, ylab = "Interest Rates")
```



Problem 2.

We begin by looking at a plot of the two price series (FTSE and SP500). The FTSE series is lined in blue, while the black line denotes the S&P 500.



Looking at a plot of the data, we see that the two indices mimic one another in fluctuations. It looks as though the SP500 (the black line) may come after the FTSE (blue line), but we will need to run a Granger Causality test to determine whether this qualitative observation is true.

We compute the returns approximately:

We choose the order for our Granger Causality test using the `VARselect()` function. An order of 7 is chosen.

```
VARselect(data.frame(ret_gspc, ret_ftse))
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      4      2      2      4
##
## $criteria
##                    1             2             3             4
## AIC(n) -1.987512e+01 -1.992770e+01 -1.993532e+01 -1.995431e+01
## HQ(n)  -1.984462e+01 -1.987687e+01 -1.986415e+01 -1.986280e+01
## SC(n)  -1.979900e+01 -1.980084e+01 -1.975770e+01 -1.972595e+01
## FPE(n)  2.335309e-09  2.215702e-09  2.198923e-09  2.157606e-09
##                    5             6             7             8
## AIC(n) -1.992891e+01 -1.993696e+01 -1.993896e+01 -1.992004e+01
## HQ(n)  -1.981708e+01 -1.980479e+01 -1.978646e+01 -1.974720e+01
## SC(n)  -1.964981e+01 -1.960711e+01 -1.955836e+01 -1.948870e+01
```

```
## FPE(n)  2.213169e-09  2.195540e-09  2.191289e-09  2.233339e-09
##                    9            10
## AIC(n) -1.989860e+01 -1.990325e+01
## HQ(n)  -1.970543e+01 -1.968975e+01
## SC(n)  -1.941651e+01 -1.937042e+01
## FPE(n)  2.281987e-09  2.271690e-09
```

```
# Picks order 7
grangertest(ret_gspc ~ ret_ftse, order = 4)
```

```
## Granger causality test
##
## Model 1: ret_gspc ~ Lags(ret_gspc, 1:4) + Lags(ret_ftse, 1:4)
## Model 2: ret_gspc ~ Lags(ret_gspc, 1:4)
##   Res.Df Df      F    Pr(>F)
## 1    286
## 2    290 -4 9.2542 4.802e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
grangertest(ret_ftse ~ ret_gspc, order = 4)
```

```
## Granger causality test
##
## Model 1: ret_ftse ~ Lags(ret_ftse, 1:4) + Lags(ret_gspc, 1:4)
## Model 2: ret_ftse ~ Lags(ret_ftse, 1:4)
##   Res.Df Df      F Pr(>F)
## 1    286
## 2    290 -4 0.9411 0.4405
```

We find that the Granger Test yields the S&P 500 is Granger caused by the FTSE, but that the converse is not true.

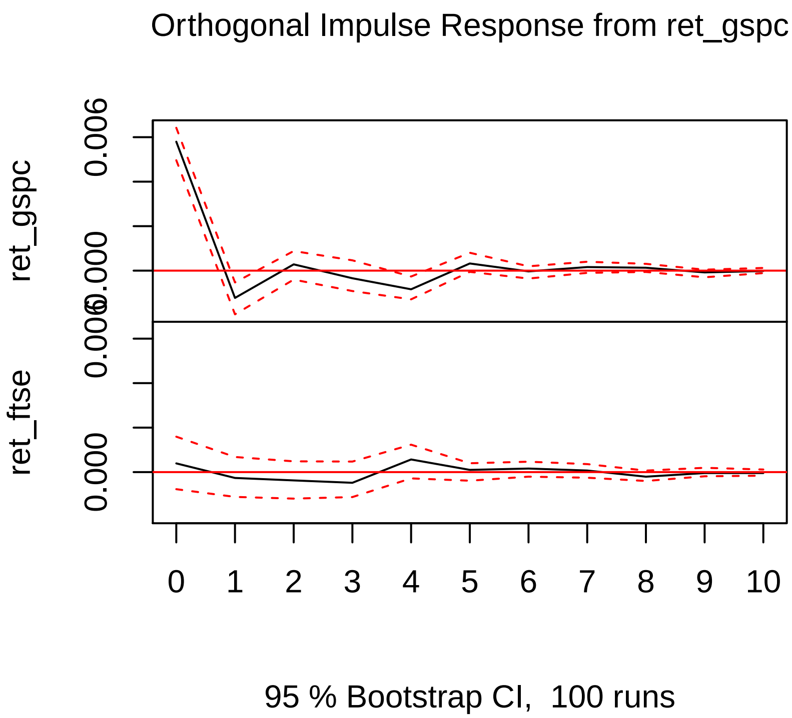
We fit a VAR model between the two series:

```
model <- VAR(data.frame(ret_gspc, ret_ftse), 4)
# summary(model)
```

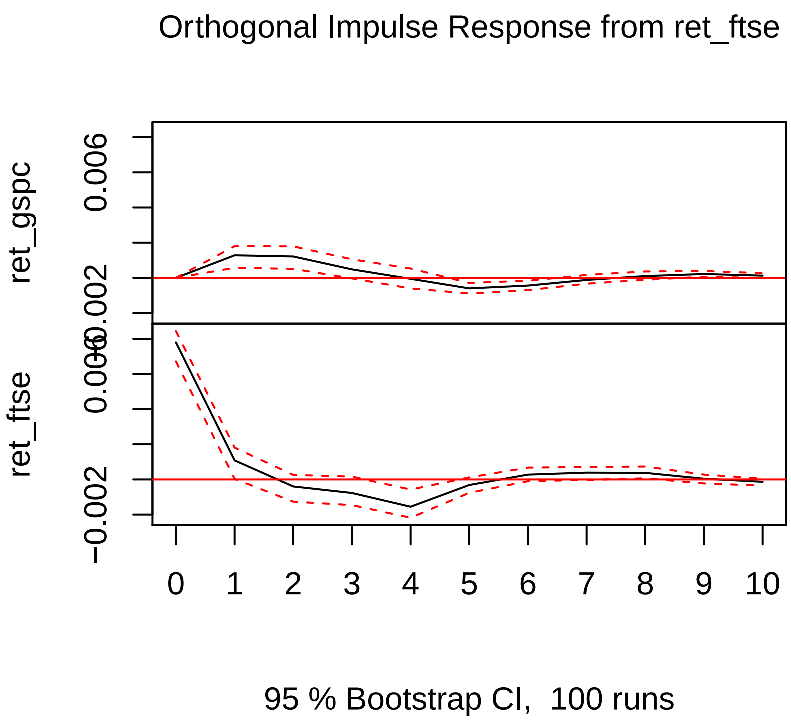
(Note: We suppress summary output for readability.)

What is interesting is that if we look at the impulse response functions, we can examine the estimated responses of one series to a unit shock to the other. We find that when a unit shock to the S&P 500 returns occurs, not much effect occurs in the FTSE. However, the FTSE returns shows a longer persistence in response for two periods.

```
plot(irf(model), plot.type='multiple', names='ret_gspc')
```




```
plot(irf(model), plot.type='multiple', names='ret_ftse')
```



Therefore, we have reason to believe that lagged values of the FTSE may help explain the S&P 500, but the converse is not necessarily true.

Problem 3.

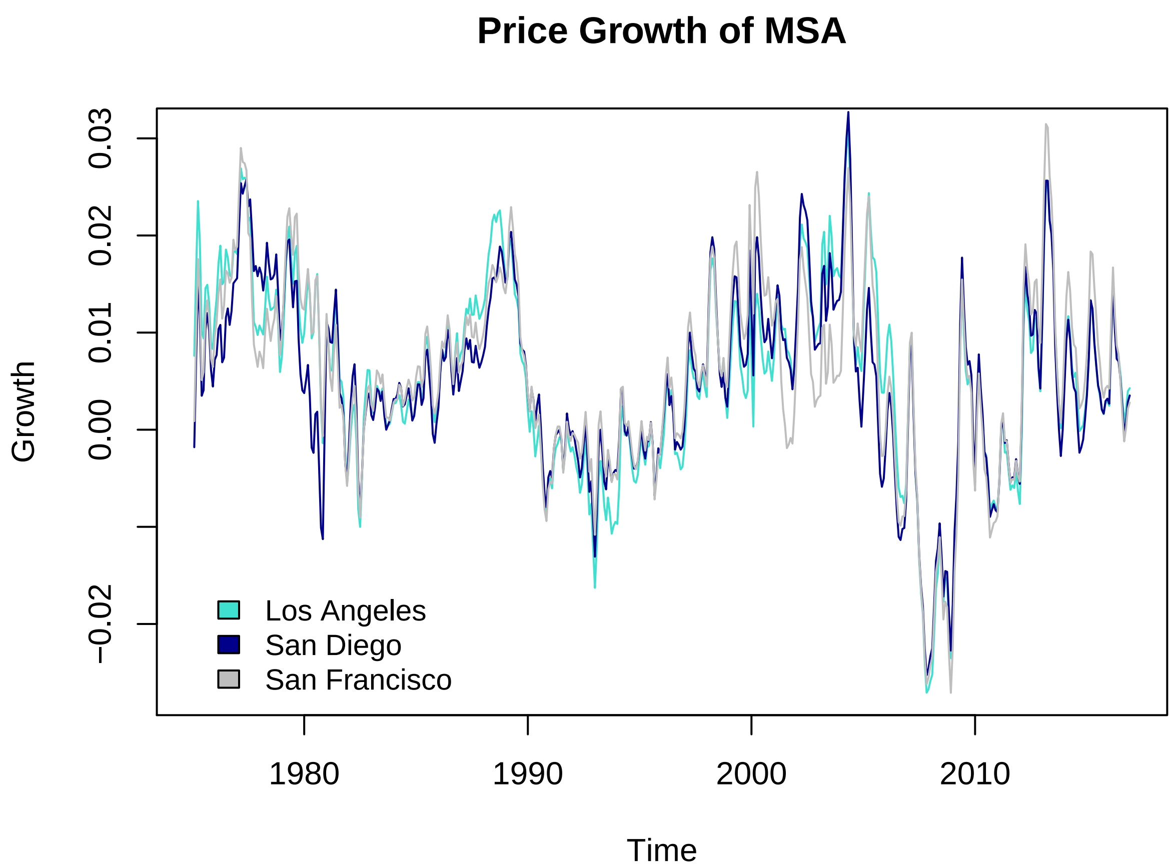
This particular question asks us essentially set up a bivariate VAR model and a trivariate VAR model, and then compare one step ahead forecasts between these two VAR models, and a univariate model.

We are examining the effect of different cities' housing price growths on other cities. The cities used are Los Angeles, San Diego, and San Francisco. (Note: the analysis provided in this portion will vary depending on the cities you have chosen.)

We begin by splitting our data into an estimation and a prediction set. While it is normal practice to split the data into 2/3rd estimation and 1/3rd prediction, via the textbook's suggestion and to cut down on computation time, we simply leave 20 observations in the prediction sample.

```
#Note:
#MSA1 refers to LA
#MSA2 refers to SD
#MSA3 refers to SF
#Data is monthly
msa1<-ts(msa$LosAngeles, start=1975, freq=12)
msa2<-ts(msa$SanDiego, start=1975, freq=12)
msa3<-ts(msa$SanFrancisco, start=1975, freq=12)
t<-seq(1975, by=1/12, length=length(msa1))

#Growth given by percentage change

msa1.growth<-(msa1/timeSeries::lag(msa1,-1))-1
msa2.growth<-(msa2/timeSeries::lag(msa2,-1))-1
msa3.growth<-(msa3/timeSeries::lag(msa3,-1))-1

sample.size<-20
msa1.est<-msa1.growth[1:(length(msa1.growth)-sample.size)]
msa2.est<-msa2.growth[1:(length(msa2.growth)-sample.size)]
msa3.est<-msa3.growth[1:(length(msa3.growth)-sample.size)]

msa1.pred<-msa1.growth[-(1:(length(msa1.growth)-sample.size))]
msa2.pred<-msa2.growth[-(1:(length(msa2.growth)-sample.size))]
msa3.pred<-msa3.growth[-(1:(length(msa3.growth)-sample.size))]
```

We begin by visualizing our series:



Intuitively we would expect the Los Angeles and San Diego price growth series to be related, as they are more geographically similar. Therefore, for the two-variable VAR model, we use Los Angeles and San Diego. We add in San Francisco into the three-variable VAR model.

```
VARselect(data.frame(msa1.est, msa2.est))
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     10     10      4     10
##
## $criteria
##                    1             2             3             4
## AIC(n) -2.602084e+01 -2.714830e+01 -2.726428e+01 -2.734804e+01
## HQ(n)  -2.600009e+01 -2.711371e+01 -2.721586e+01 -2.728578e+01
## SC(n)  -2.596808e+01 -2.706037e+01 -2.714118e+01 -2.718976e+01
## FPE(n)  5.003711e-12  1.620485e-12  1.443034e-12  1.327098e-12
##                    5             6             7             8
## AIC(n) -2.737768e+01 -2.736649e+01 -2.743362e+01 -2.743087e+01
## HQ(n)  -2.730160e+01 -2.727657e+01 -2.732987e+01 -2.731328e+01
## SC(n)  -2.718424e+01 -2.713787e+01 -2.716983e+01 -2.713190e+01
## FPE(n)  1.288341e-12  1.302854e-12  1.218282e-12  1.221668e-12
##                    9            10
## AIC(n) -2.747198e+01 -2.748691e+01
## HQ(n)  -2.734056e+01 -2.734165e+01
```

```
## SC(n)  -2.713785e+01 -2.711760e+01
## FPE(n)  1.172485e-12  1.155150e-12
```

```r
# Lag of 4
var1 <- VAR(data.frame(msa1.est, msa2.est), 4)
# summary(var1)
```

We find from running a Granger causality test that using a lag order of 4, San Diego housing price growth does not Granger cause Los Angeles housing price growth, but that Los Angeles housing price growth does Granger cause San Diego's housing price growth. In other words, we may use lagged values of Los Angele's housing price growth to help explain contemperaneous observations of San Diego's housing price growth, but the converse is not true.

```r
grangertest(msa1.est ~ msa2.est, order = 4)  #Insignificant
```

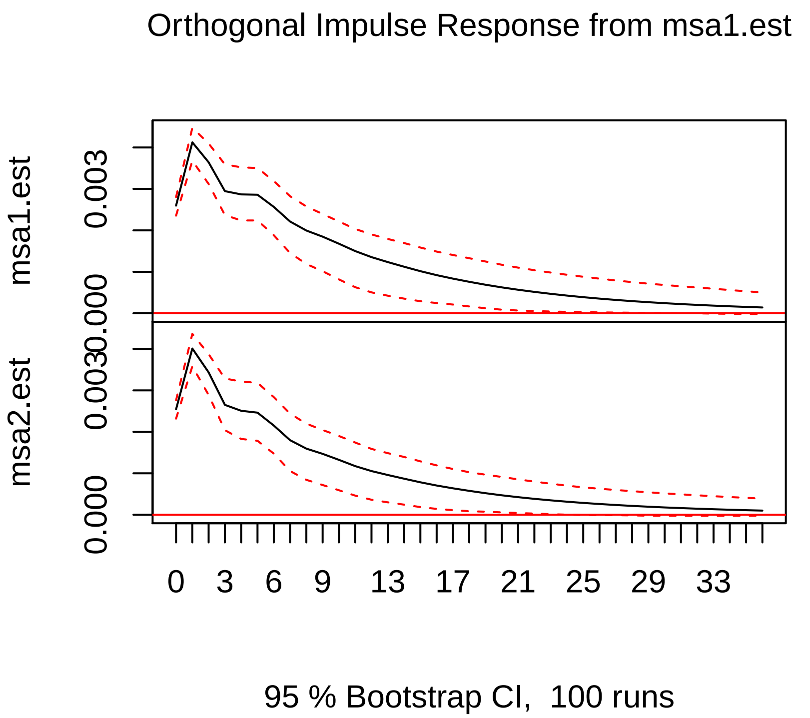
```
## Granger causality test
##
## Model 1: msa1.est ~ Lags(msa1.est, 1:4) + Lags(msa2.est, 1:4)
## Model 2: msa1.est ~ Lags(msa1.est, 1:4)
##   Res.Df Df      F Pr(>F)
## 1    470
## 2    474 -4 1.2202 0.3014
```

```r
grangertest(msa2.est ~ msa1.est, order = 4)  #Significant
```

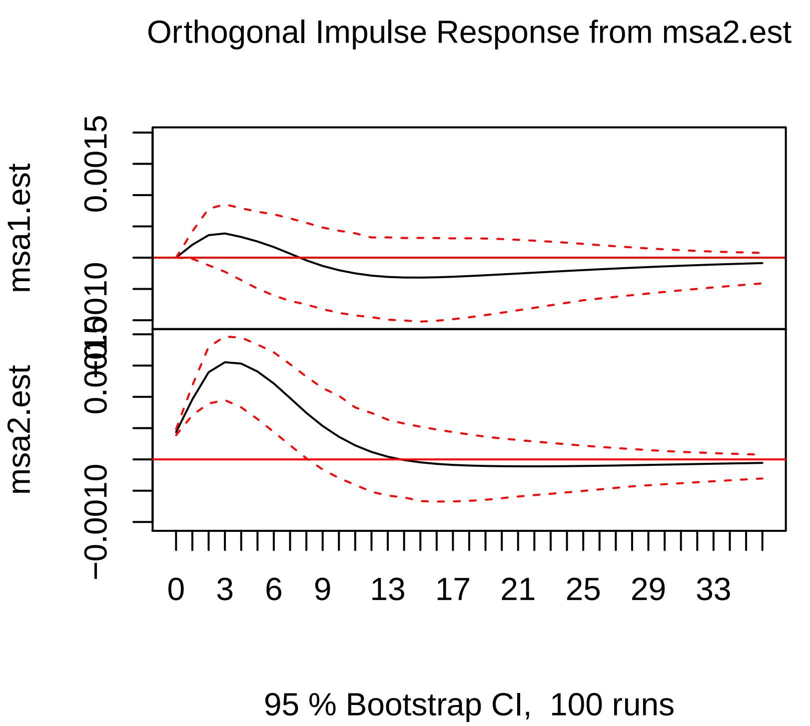
```
## Granger causality test
##
## Model 1: msa2.est ~ Lags(msa2.est, 1:4) + Lags(msa1.est, 1:4)
## Model 2: msa2.est ~ Lags(msa2.est, 1:4)
##   Res.Df Df      F   Pr(>F)
## 1    470
## 2    474 -4 4.5608 0.00127 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

A look at the impulse response functions confirms this.

```r
irf1 <- irf(var1, n.ahead = 36)
plot(irf1, plot.type = "multiple", names = "msa1.est")
```




```
plot(irf1, plot.type = "multiple", names = "msa2.est")
```



We expand our VAR model to include San Francisco's data now.

```
VARselect(data.frame(msa1.est, msa2.est, msa3.est))
```

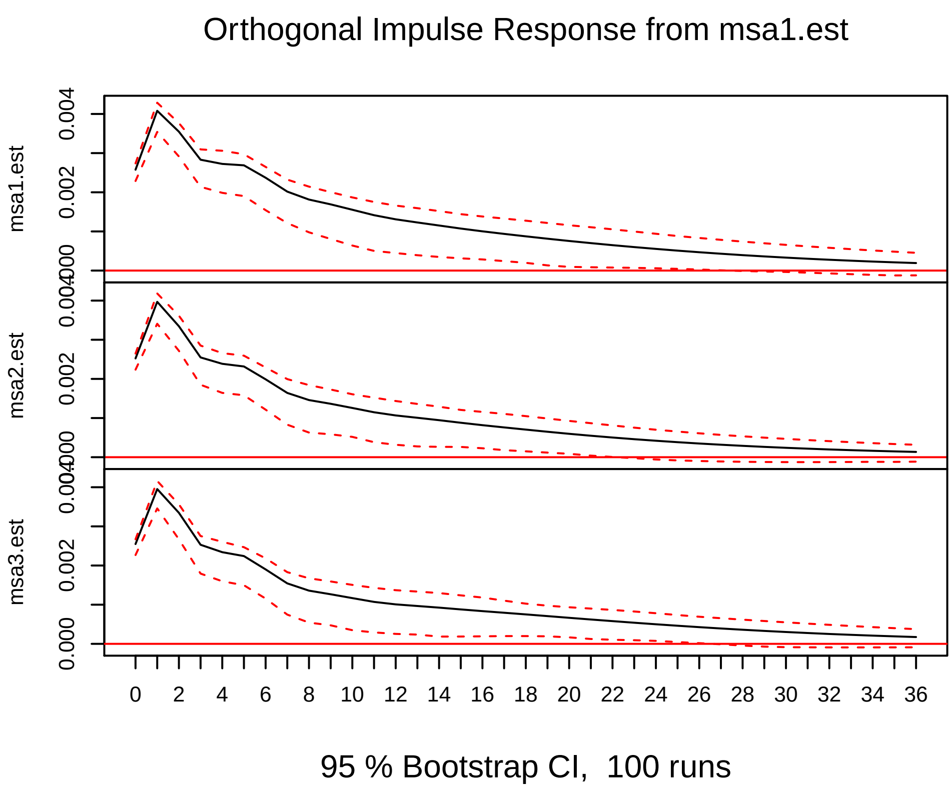
```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
```

```
##      10       5       4      10
##
## $criteria
##                     1             2             3             4
## AIC(n) -4.068894e+01 -4.250243e+01 -4.270452e+01 -4.284400e+01
## HQ(n)  -4.064744e+01 -4.242980e+01 -4.260077e+01 -4.270912e+01
## SC(n)  -4.058343e+01 -4.231777e+01 -4.244073e+01 -4.250107e+01
## FPE(n)  2.133132e-18  3.478831e-19  2.842307e-19  2.472331e-19
##                     5             6             7             8
## AIC(n) -4.289976e+01 -4.287552e+01 -4.293454e+01 -4.291771e+01
## HQ(n)  -4.273375e+01 -4.267838e+01 -4.270628e+01 -4.265832e+01
## SC(n)  -4.247769e+01 -4.237432e+01 -4.235420e+01 -4.225823e+01
## FPE(n)  2.338328e-19  2.395829e-19  2.258666e-19  2.297235e-19
##                     9            10
## AIC(n) -4.297174e+01 -4.299000e+01
## HQ(n)  -4.268123e+01 -4.266836e+01
## SC(n)  -4.223313e+01 -4.217225e+01
## FPE(n)  2.176660e-19  2.137602e-19
```

```
tri_var <- VAR(data.frame(msa1.est, msa2.est, msa3.est), 4)
# summary(tri_var)
```

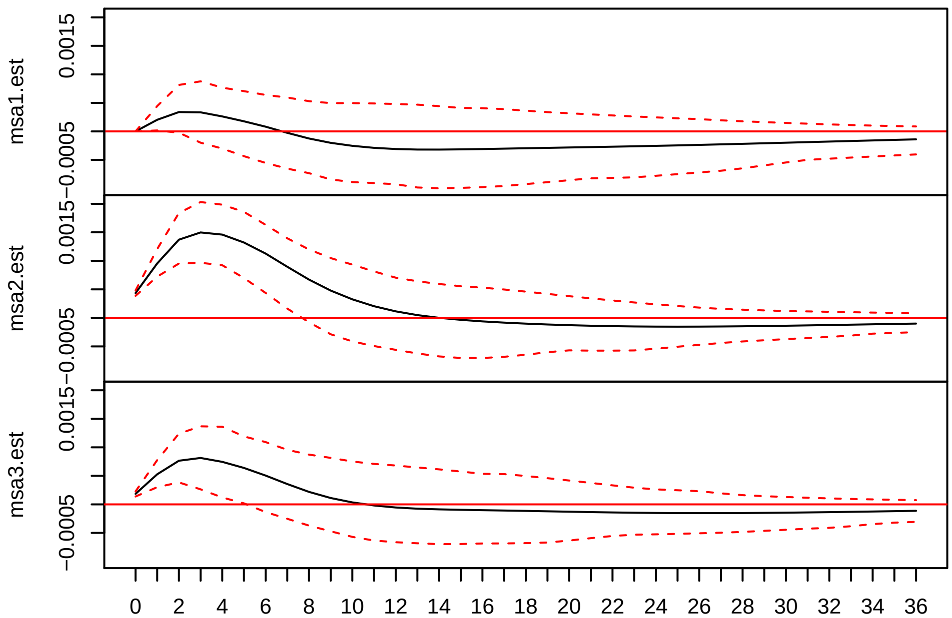
The IRF plots are displayed below:

```
tri_irf <- irf(tri_var, n.ahead = 36)
plot(tri_irf, plot.type = "multiple", names = "msa1.est")
```



```r
plot(tri_irf, plot.type = "multiple", names = "msa2.est")
```

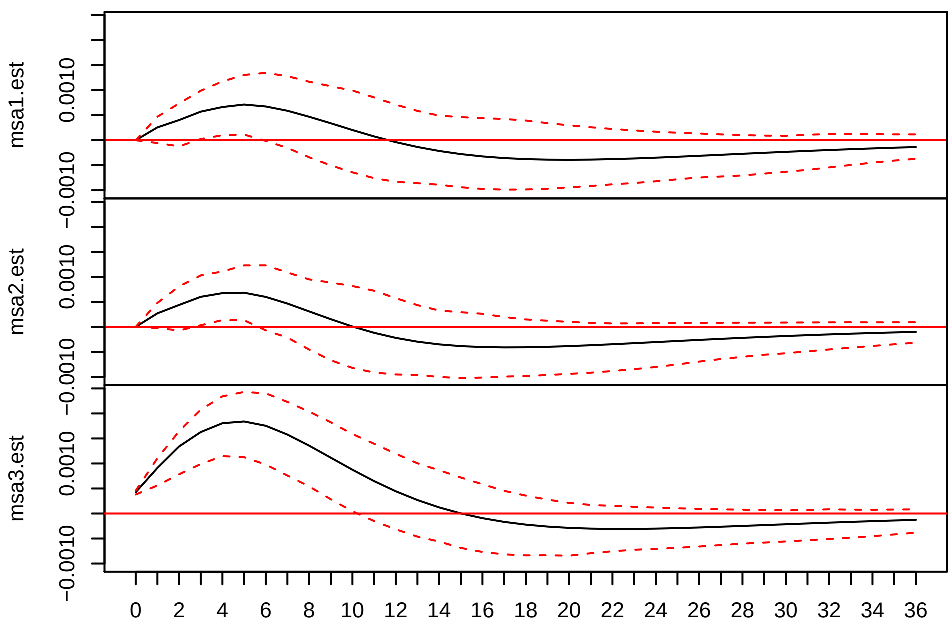
Orthogonal Impulse Response from msa2.est



95 % Bootstrap CI, 100 runs

```r
plot(tri_irf, plot.type = "multiple", names = "msa3.est")
```

Orthogonal Impulse Response from msa3.est



95 % Bootstrap CI, 100 runs

We see that unit shocks to each series leads to some response from the other series. What is odd is that the effect of the shock appears to be occurring at the same time in all three series (at least qualitatively).

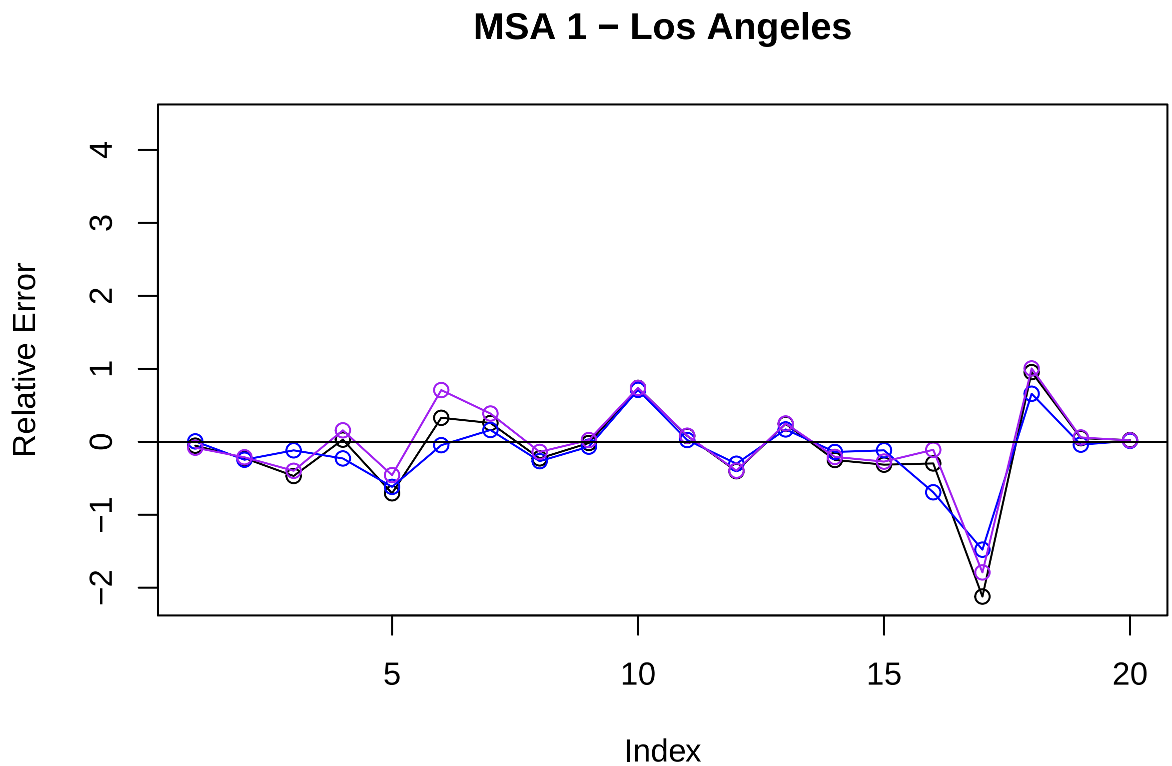
We now backtest the performance of the original, two-variable VAR model, the three-variable VAR model, and a univariate ARIMA model in forecasting 1 step ahead.

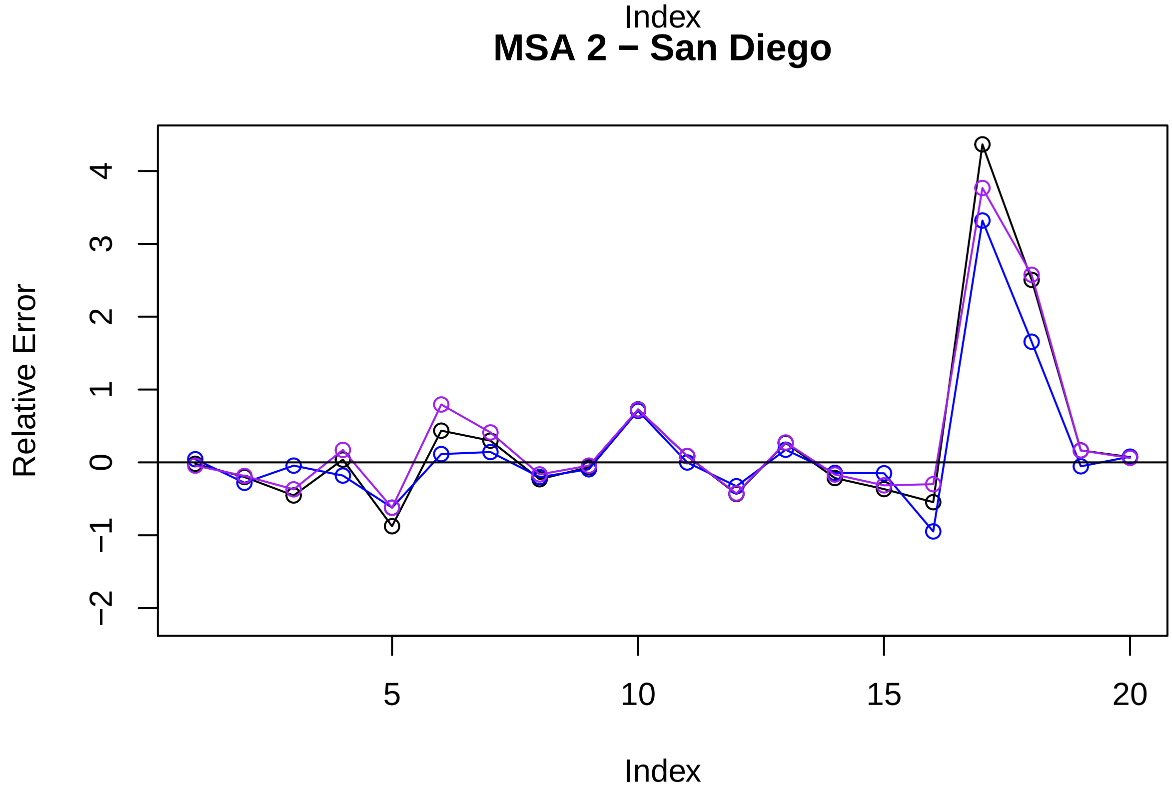
The backtesting sequences are displayed below:

```
# For MSA1/MSA2 only:
var.predict = predict(var1, n.ahead = 1)
err1 <- msa1.pred[1] - var.predict$fcst$msa1.est[, 1]
rel_err1 <- err1/msa1.pred[1]
err2 <- msa2.pred[1] - var.predict$fcst$msa2.est[, 1]
rel_err2 <- err2/msa2.pred[1]
# VAR backtesting:
for (i in 1:19) {
    est1 <- append(msa1.est, msa1.pred[1:i])
    est2 <- append(msa2.est, msa2.pred[1:i])
    # We keep lag of 4 fixed
    model <- VAR(data.frame(est1, est2), 4)
    pred <- predict(model, n.ahead = 1)
    error1 <- msa1.pred[i + 1] - pred$fcst$est1[, 1]
    err1 <- append(err1, error1)
    rel_err1 <- append(rel_err1, error1/msa1.pred[i + 1])

    error2 <- msa2.pred[i + 1] - pred$fcst$est2[, 1]
    err2 <- append(err2, error2)
    rel_err2 <- append(rel_err2, error2/msa2.pred[i + 1])
}

# ARIMA:
arima.msa1 <- predict(auto.arima(msa1.est), 1)
arima_err1 <- msa1.pred[1] - arima.msa1$pred
arima.msa2 <- predict(auto.arima(msa2.est), 1)
arima_err2 <- msa1.pred[1] - arima.msa2$pred

arima_rel_err1 <- arima_err1/msa1.pred[1]
arima_rel_err2 <- arima_err2/msa2.pred[1]

for (i in 1:19) {
    est1 <- append(msa1.est, msa1.pred[1:i])
    est2 <- append(msa2.est, msa2.pred[1:i])

    arima.msa1 <- data.frame(forecast(auto.arima(est1), 1))[,
        1]
    arima.msa2 <- data.frame(forecast(auto.arima(est2), 1))[,
        1]
```

```
    error1 <- msa1.pred[i + 1] - arima.msa1
    arima_err1 <- append(arima_err1, error1)
    arima_rel_err1 <- append(arima_rel_err1, error1/msa1.pred[i +
        1])

    error2 <- msa2.pred[i + 1] - arima.msa2
    arima_err2 <- append(arima_err2, error2)
    arima_rel_err2 <- append(arima_rel_err2, error2/msa2.pred[i +
        1])
}


var.predict2 = predict(tri_var, n.ahead = 1)
tri_err1 <- msa1.pred[1] - var.predict2$fcst$msa1.est[, 1]
tri_rel_err1 <- tri_err1/msa1.pred[1]

tri_err2 <- msa2.pred[1] - var.predict2$fcst$msa2.est[, 1]
tri_rel_err2 <- tri_err2/msa2.pred[1]

tri_err3 <- msa3.pred[1] - var.predict2$fcst$msa3.est[, 1]
tri_rel_err3 <- tri_err3/msa3.pred[1]

# VAR backtesting:
for (i in 1:19) {
    tri_est1 <- append(msa1.est, msa1.pred[1:i])
    tri_est2 <- append(msa2.est, msa2.pred[1:i])
    tri_est3 <- append(msa3.est, msa3.pred[1:i])

    # We keep lag of 4 fixed
    model <- VAR(data.frame(tri_est1, tri_est2, tri_est3), 4)
    pred <- predict(model, n.ahead = 1)
    error1 <- msa1.pred[i + 1] - pred$fcst$tri_est1[, 1]
    tri_err1 <- append(tri_err1, error1)
    tri_rel_err1 <- append(tri_rel_err1, error1/msa1.pred[i +
        1])

    error2 <- msa2.pred[i + 1] - pred$fcst$tri_est2[, 1]
    tri_err2 <- append(tri_err2, error2)
    tri_rel_err2 <- append(tri_rel_err2, error2/msa2.pred[i +
        1])

    error3 <- msa3.pred[i + 1] - pred$fcst$tri_est3[, 1]
    tri_err3 <- append(tri_err3, error3)
    tri_rel_err3 <- append(tri_rel_err3, error3/msa3.pred[i +
        1])
}
```

Because we only implement a two-variable VAR model for Los Angeles and San Diego's housing price growth series, we compare the performance of the third model's capability in forecasting only the series corresponding to Los Angeles and San Diego.





To look at the average absolute relative errors:

```
mean(abs(rel_err1))

## [1] 0.3894558

mean(abs(arima_err1))

## [1] 0.001190507

mean(abs(tri_rel_err1))

## [1] 0.3747893

mean(abs(rel_err2))

## [1] 0.618561

mean(abs(arima_err2))

## [1] 0.00122006

mean(abs(tri_rel_err2))

## [1] 0.5835785
```

We find that in general, the ARIMA model actually performs best. We do see a slight reduction in relative error when adding in a third variable into our VAR model. However, the difference is marginal, so we would have to test the statistical significance of the improvement in model performance with the inclusion of a new variable and all of its subsequent lags.

R-Code:

```
rm(list = ls(all = TRUE))
#------------------------------------------------------------------------
# Load Libraries
library(lattice)
library(foreign)
library(MASS)
library(car)
require(stats)
require(stats4)
library(KernSmooth)
library(fastICA)
library(cluster)
library(leaps)
library(mgcv)
library(rpart)
library(pan)
library(mgcv)
library(DAAG)
library(TTR)
```

```r
library(tis)
require("datasets")
require(graphics)
library(forecast)
# install.packages('astsa') require(astsa)
library(xtable)
# New libraries added:
library(stats)
library(TSA)
library(timeSeries)
library(fUnitRoots)
library(fBasics)
library(tseries)
library(timsac)
library(TTR)
library(fpp)
library(strucchange)
# library(MSBVAR)
library(vars)
library(lmtest)
library(dlnm)
library(dplyr)
library(mice)
#---------------------------------------------------------------------
# Question 1. (50%) The file w-gs1yr.txt contains the U.S.
# weekly interest rates (in percentages) from January 5,
# 1962, to April 10, 2009. For this assignment you will fit
# an appropriate ARMA(p,q) model and make a 24-steps-ahead
# forecast.

# Setting up the data:
data = read.table("w-gs1yr.txt", header = TRUE)
rates <- ts(data$rate, start = 1962, deltat = 1/52, freq = 52)
t <- seq(1962, length = length(rates), by = 1/52)

# Part (a) Show a plot of the data, along with the respective
# ACF and PACF functions. Discuss the plots. Data Plot
quartz()
plot(t, rates, main = "US Weekly Interest Rates (%)", ylab = "Rates",
    xlab = "Time", type = "l")
nberShade()
lines(rates)

# ACF/PACF We observe 2 significant spikes in PACF and a
# dampening effect in the ACF; however, the spikes appear
# close to 1, which may indicate that the series is not
# covariance stationary, and that we will need to take the
```

```
# first difference to successfully implement an ARMA model.
quartz()
tsdisplay(rates)

# Looking at the first difference:
quartz()
tsdisplay(diff(rates))

# We see 1 significant spike in the PACF, and the ACF
# dampening a lot quicker in the first difference. This could
# indicate an AR(1) process, though there appears to be some
# more complex seasonal components as well.

# Part (b) Based on your discussion in (a), fit 3 different
# ARMA(p,q) models, and show the fits over the original data.
# Discuss your results, and select one model as your
# preferred choice. Fit 1: ARMA(2, 0)
fit1 <- Arima(rates, order = c(2, 0, 0))
quartz()
plot(t, rates, main = "US Weekly Interest Rates (%)", ylab = "Rates",
    xlab = "Time", type = "l")
lines(fitted(fit1), col = "red")

# Fit 2: ARMA(0, 10)
fit2 <- Arima(rates, order = c(0, 0, 10))
quartz()
plot(t, rates, main = "US Weekly Interest Rates (%)", ylab = "Rates",
    xlab = "Time", type = "l")
lines(fitted(fit2), col = "red")

# After differencing: Fit 3: ARMA(1, 0)
fit3 <- Arima(rates, order = c(1, 1, 0))
quartz()
plot(t, rates, main = "US Weekly Interest Rates (%)", ylab = "Rates",
    xlab = "Time", type = "l")
lines(fitted(fit3), col = "red")

AIC(fit1, fit2, fit3)
BIC(fit1, fit2, fit3)
# AIC/BIC pick the 3rd fit

# Part (c) Now that you fit an ARMA(p,q) model to the data,
# plot the ACF and PACF of the residuals from your preferred
# fit model, and discuss your results.
quartz()
par(mfrow = c(2, 1))
acf(resid(fit3))
```

```r
pacf(resid(fit3))
# There appears to still be some structure left in the
# residuals, though the spikes are pretty small; we would
# have to conduct tests to check for the significance of the
# spikes.

# Part (d) Compute and plot the recursive residuals from your
# best fit model. Interpret the plot.
y1 <- recresid(fit3$res ~ 1)
quartz()
plot(y1, pch = 16, ylab = "Recursive Residuals")
# ??

# Part (e) Compute and plot the CUSUM from your best fit
# model. Interpret the plot.
quartz()
plot(efp(fit3$res ~ 1, type = "Rec-CUSUM"))
# Does not show any structural breaks!

# Part (f) Compute the best fit model according to ‘R’ and
# compare it against your model. Discuss these results.
fit4 <- auto.arima(rates)
# auto.arima returns the following: ARIMA(1,1,2)
quartz()
plot(t, rates, main = "US Weekly Interest Rates (%)", ylab = "Rates",
    xlab = "Time", type = "l")
lines(fit4$fitted, col = "blue")

quartz()
par(mfrow = c(2, 1))
acf(resid(fit4))
pacf(resid(fit4))

y2 <- recresid(fit4$res ~ 1)
quartz()
plot(y2, pch = 16, ylab = "Recursive Residuals")

quartz()
plot(efp(fit4$res ~ 1, type = "Rec-CUSUM"))

# Part (g) Using your best fit model as well as ‘R’s’ best
# fit model, compute the respective 24-steps-ahead forecast,
# and compare your results.
quartz()
par(mfrow = c(2, 1))
plot(forecast(fit3, h = 24))
plot(forecast(fit4, h = 24))
```

```r
# They are basically identical. However, it doesn't appear
# quite correct.

r.hw <- HoltWinters(rates)
quartz()
r.hw.pred <- predict(r.hw, 48, prediction.interval = TRUE, level = 0.5)
plot(r.hw, r.hw.pred, ylab = "Interest Rates")
#-----------------------------------------------------------------------
# Question 2 When the US stock market opens in New York, the
# European markets have already been in session for several
# hours. Does the activity in European markets have
# predictive content for the US market? Download the British
# stock index (FTSE) and the SP500 index at the daily
# frequency. Examine whether FTSE returns can help to
# forecast SP500 returns.
#-----------------------------------------------------------------------
# SP500
gspc <- as.data.frame(get.hist.quote("^GSPC", start = "1984-01-03",
    quote = "AdjClose", compression = "d", retclass = "ts"))
gspc_tis <- tis(gspc$AdjClose, start = "1984-01-03", freq = 1)
gspc_interp <- as.data.frame(interpNA(gspc_tis, method = "linear"))
names(gspc_interp) <- c("x")
s_ts <- ts(gspc_interp$x, start = 1984 + (3/365), deltat = 1/365)
t <- seq(1984 + (3/365), by = 1/365, length = length(s_ts))

ftse <- as.data.frame(get.hist.quote("^FTSE", start = "1984-01-03",
    quote = "AdjClose", compression = "d", retclass = "ts"))
ftse_tis <- tis(ftse$AdjClose, start = "1984-01-03", freq = 1)
ftse_interp <- as.data.frame(interpNA(ftse_tis, method = "linear"))
names(ftse_interp) <- c("x")
f_ts <- ts(ftse_interp$x, start = 1984 + (3/365), deltat = 1/365)

# Standardize:
z_sp500 <- (s_ts - mean(s_ts))/sd(s_ts)
z_ftse <- (f_ts - mean(f_ts))/sd(f_ts)

plot(t, z_sp500, ylim = c(0, max(f_ts)), ylab = "Stock Price",
    xlab = "Time", type = "l")
lines(t, z_ftse, col = "blue")

# Get Returns:
gspc <- stockPortfolio::getReturns("^GSPC", start = "1984-01-03",
    freq = "day")
ftse <- stockPortfolio::getReturns("^FTSE", start = "1984-01-03",
    freq = "day")

df_gspc <- data.frame(dates = as.Date(rownames(gspc$R)), gspc = gspc$R,
```

```r
    stringsAsFactors = FALSE)
df_ftse <- data.frame(dates = as.Date(rownames(ftse$R)), ftse = ftse$R,
    stringsAsFactors = FALSE)

data <- left_join(df_gspc, df_ftse, by = "dates")
data_imputed <- mice(data[, -1], method = "pmm")
data_all <- cbind(data[, 1], complete(data_imputed))

ret_gspc <- ts(rev(data_all[, 2]))
ret_ftse <- ts(rev(data_all[, 3]))

VARselect(data.frame(ret_gspc, ret_ftse))

# Granger test: Picks order 7
grangertest(ret_gspc ~ ret_ftse, order = 7)
grangertest(ret_ftse ~ ret_gspc, order = 7)

# VAR
model <- VAR(data.frame(ret_gspc, ret_ftse), 7)
summary(model)

plot(irf(model))
#---------------------------------------------------------------------
# Problem 3 (11.9)
msa <- read.csv("~/Documents/Econ 144 (S17)/Homework 4 Solutions/msa.csv",
    stringsAsFactors = FALSE, header = TRUE)
# Note: MSA1 refers to Denver, Colorado MSA2 refers to Los
# Angeles, CA Data is monthly
msa1 <- ts(msa$LosAngeles, start = 1975, freq = 12)
msa2 <- ts(msa$SanDiego, start = 1975, freq = 12)
msa3 <- ts(msa$SanFrancisco, start = 1975, freq = 12)
t <- seq(1975, by = 1/12, length = length(msa1))

# Growth given by percentage change

msa1.growth <- (msa1/timeSeries::lag(msa1, -1)) - 1
msa2.growth <- (msa2/timeSeries::lag(msa2, -1)) - 1
msa3.growth <- (msa3/timeSeries::lag(msa3, -1)) - 1

sample.size <- 20
msa1.est <- msa1.growth[1:(length(msa1.growth) - sample.size)]
msa2.est <- msa2.growth[1:(length(msa2.growth) - sample.size)]
msa3.est <- msa3.growth[1:(length(msa3.growth) - sample.size)]

msa1.pred <- msa1.growth[-(1:(length(msa1.growth) - sample.size))]
msa2.pred <- msa2.growth[-(1:(length(msa2.growth) - sample.size))]
msa3.pred <- msa3.growth[-(1:(length(msa3.growth) - sample.size))]
```

```r
#-----------------------------------------------------------------------
# We begin by visualizing our series:
plot(msa1.growth, type = "l", col = "turquoise", ylab = "Growth",
    main = "Price Growth of MSA")
lines(msa2.growth, col = "darkblue")
lines(msa3.growth, col = "gray")
legend(1975, -0.015, c("Los Angeles", "San Diego", "San Francisco"),
    fill = c("turquoise", "darkblue", "gray"))

# Intuitively we would expect the Los Angeles and San Diego
# price growth series to be related, as they are more
# geographically similar. Therefore, we begin by fitting a
# VAR model between these two series:
VARselect(data.frame(msa1.est, msa2.est))
# Lag of 4
var1 <- VAR(data.frame(msa1.est, msa2.est), 4)
summary(var1)

grangertest(msa1.est ~ msa2.est, order = 4)  #Insignificant
grangertest(msa2.est ~ msa1.est, order = 4)  #Significant

irf1 <- irf(var1, n.ahead = 36)
plot(irf1)


#-----------------------------------------------------------------------
# Part 4: Construct the 1-step ahead forecast. Compare the
# 1-step ahead forecasts from the VAR model with those that
# you would obtain from the best univariate model that you
# could find for each series.

# For MSA1/MSA2 only:
var.predict = predict(var1, n.ahead = 1)
err1 <- msa1.pred[1] - var.predict$fcst$msa1.est[, 1]
rel_err1 <- err1/msa1.pred[1]
err2 <- msa2.pred[1] - var.predict$fcst$msa2.est[, 1]
rel_err2 <- err2/msa2.pred[1]
# VAR backtesting:
for (i in 1:19) {
    est1 <- append(msa1.est, msa1.pred[1:i])
    est2 <- append(msa2.est, msa2.pred[1:i])
    # We keep lag of 4 fixed
    model <- VAR(data.frame(est1, est2), 4)
    pred <- predict(model, n.ahead = 1)
    error1 <- msa1.pred[i + 1] - pred$fcst$est1[, 1]
    err1 <- append(err1, error1)
    rel_err1 <- append(rel_err1, error1/msa1.pred[i + 1])
```

```
    error2 <- msa2.pred[i + 1] - pred$fcst$est2[, 1]
    err2 <- append(err2, error2)
    rel_err2 <- append(rel_err2, error2/msa2.pred[i + 1])
}

# ARIMA:
arima.msa1 <- predict(auto.arima(msa1.est), 1)
arima_err1 <- msa1.pred[1] - arima.msa1$pred
arima.msa2 <- predict(auto.arima(msa2.est), 1)
arima_err2 <- msa1.pred[1] - arima.msa2$pred


arima_rel_err1 <- arima_err1/msa1.pred[1]
arima_rel_err2 <- arima_err2/msa2.pred[1]

for (i in 1:19) {
    est1 <- append(msa1.est, msa1.pred[1:i])
    est2 <- append(msa2.est, msa2.pred[1:i])

    arima.msa1 <- predict(auto.arima(est1), 1)
    arima.msa2 <- predict(auto.arima(est2), 1)

    error1 <- msa1.pred[i + 1] - arima.msa1$pred
    arima_err1 <- append(arima_err1, error1)
    arima_rel_err1 <- append(arima_rel_err1, error1/msa1.pred[i +
        1])

    error2 <- msa2.pred[i + 1] - arima.msa2$pred
    arima_err2 <- append(arima_err2, error2)
    arima_rel_err2 <- append(arima_rel_err2, error2/msa2.pred[i +
        1])
}

#-----------------------------------------------------------------------
# Expanding to 3 variables:
VARselect(data.frame(msa1.est, msa2.est, msa3.est))
tri_var <- VAR(data.frame(msa1.est, msa2.est, msa3.est), 4)
summary(tri_var)

tri_irf <- irf(tri_var, n.ahead = 36)
plot(tri_irf)

var.predict2 = predict(tri_var, n.ahead = 1)
tri_err1 <- msa1.pred[1] - var.predict2$fcst$msa1.est[, 1]
tri_rel_err1 <- tri_err1/msa1.pred[1]

tri_err2 <- msa2.pred[1] - var.predict2$fcst$msa2.est[, 1]
```

```
tri_rel_err2 <- tri_err2/msa2.pred[1]

tri_err3 <- msa3.pred[1] - var.predict2$fcst$msa3.est[, 1]
tri_rel_err3 <- tri_err3/msa3.pred[1]

# VAR backtesting:
for (i in 1:19) {
    tri_est1 <- append(msa1.est, msa1.pred[1:i])
    tri_est2 <- append(msa2.est, msa2.pred[1:i])
    tri_est3 <- append(msa3.est, msa3.pred[1:i])

    # We keep lag of 4 fixed
    model <- VAR(data.frame(tri_est1, tri_est2, tri_est3), 4)
    pred <- predict(model, n.ahead = 1)
    error1 <- msa1.pred[i + 1] - pred$fcst$tri_est1[, 1]
    tri_err1 <- append(tri_err1, error1)
    tri_rel_err1 <- append(tri_rel_err1, error1/msa1.pred[i +
        1])

    error2 <- msa2.pred[i + 1] - pred$fcst$tri_est2[, 1]
    tri_err2 <- append(tri_err2, error2)
    tri_rel_err2 <- append(tri_rel_err2, error2/msa2.pred[i +
        1])

    error3 <- msa3.pred[i + 1] - pred$fcst$tri_est3[, 1]
    tri_err3 <- append(tri_err3, error3)
    tri_rel_err3 <- append(tri_rel_err3, error3/msa3.pred[i +
        1])
}


par(mfrow = c(1, 2))
plot(rel_err1, type = "o", main = "MSA 1 - Los Angeles", ylim = range(rel_err1,
    rel_err2), ylab = "Relative Error")
lines(arima_rel_err1, type = "o", col = "blue")
lines(tri_rel_err1, type = "o", col = "purple")
abline(h = 0)

plot(rel_err2, type = "o", main = "MSA 2 - San Diego", ylim = range(rel_err1,
    rel_err2), ylab = "Relative Error")
lines(arima_rel_err2, type = "o", col = "blue")
lines(tri_rel_err2, type = "o", col = "purple")
abline(h = 0)

mean(abs(rel_err1))
mean(abs(arima_err1))
mean(abs(tri_rel_err1))
```

```
mean(abs(rel_err2))
mean(abs(arima_err2))
mean(abs(tri_rel_err2))
```