Vector Auto Regression Modelling

Problem 1: VAR Implementation

Use the data on quarterly excess stock market returns, the market Dividend / Price ratio, and the di§erence between the 10-yr Treasury yield and the Fed Funds rate in the excel spreadsheet “MktRet_DP_TermSpread.xlsx”. The interest rate data is from the FRED data depository, available online from the St. Louis Fed.

1. Plot each series. Give the sample mean, standard deviation, and First order autocor- relation of each series. From the First-order autocorrelation, calculate the half-life of each series (see ARMA notes for exact half-life formula).

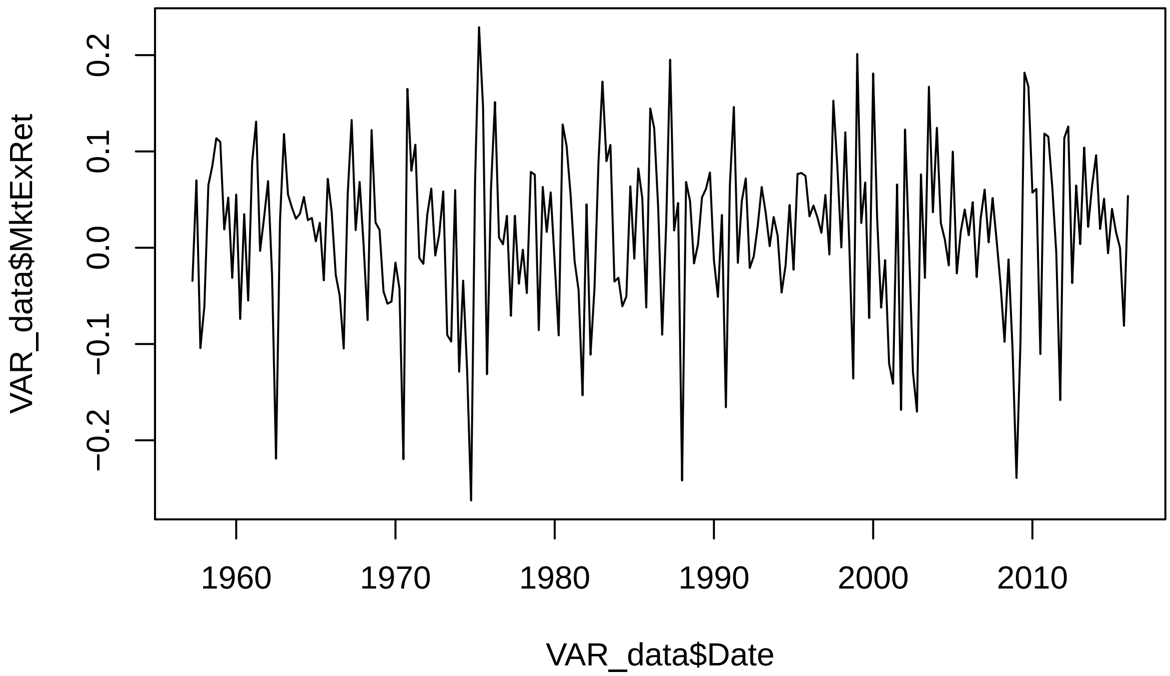
```r
library(readxl)
library(DataAnalytics)
library(dynlm)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

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

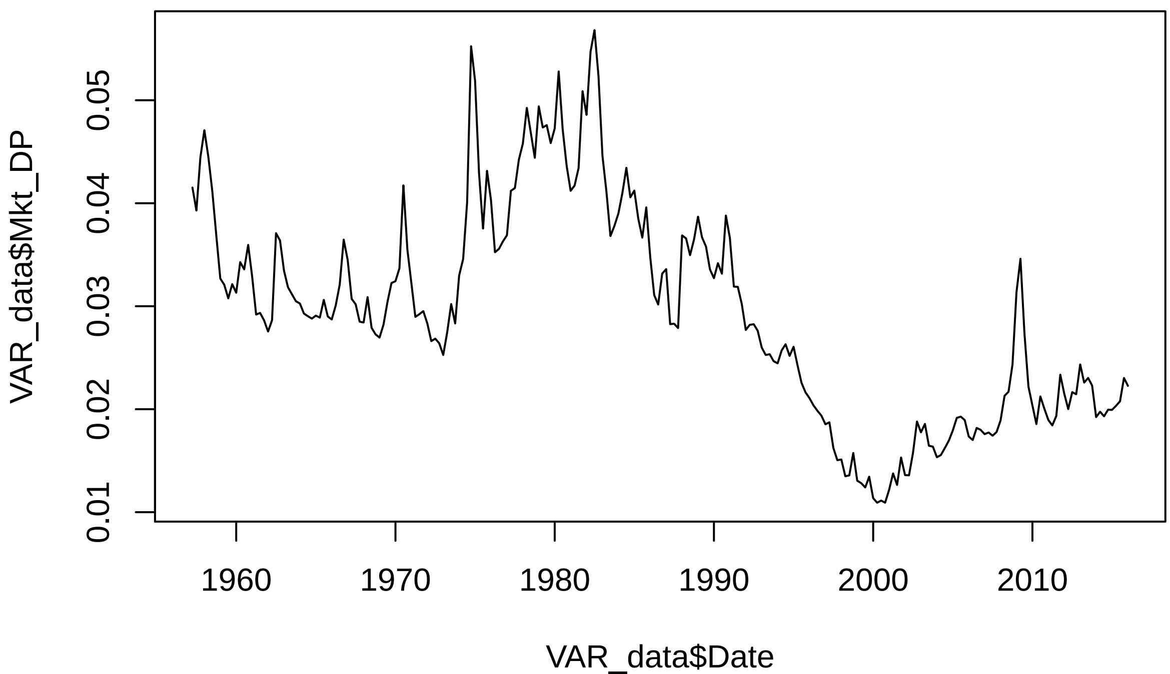
```r
library(vars)
```

```
## Loading required package: MASS

## Loading required package: strucchange

## Loading required package: sandwich

## Loading required package: urca

## Loading required package: lmtest
```

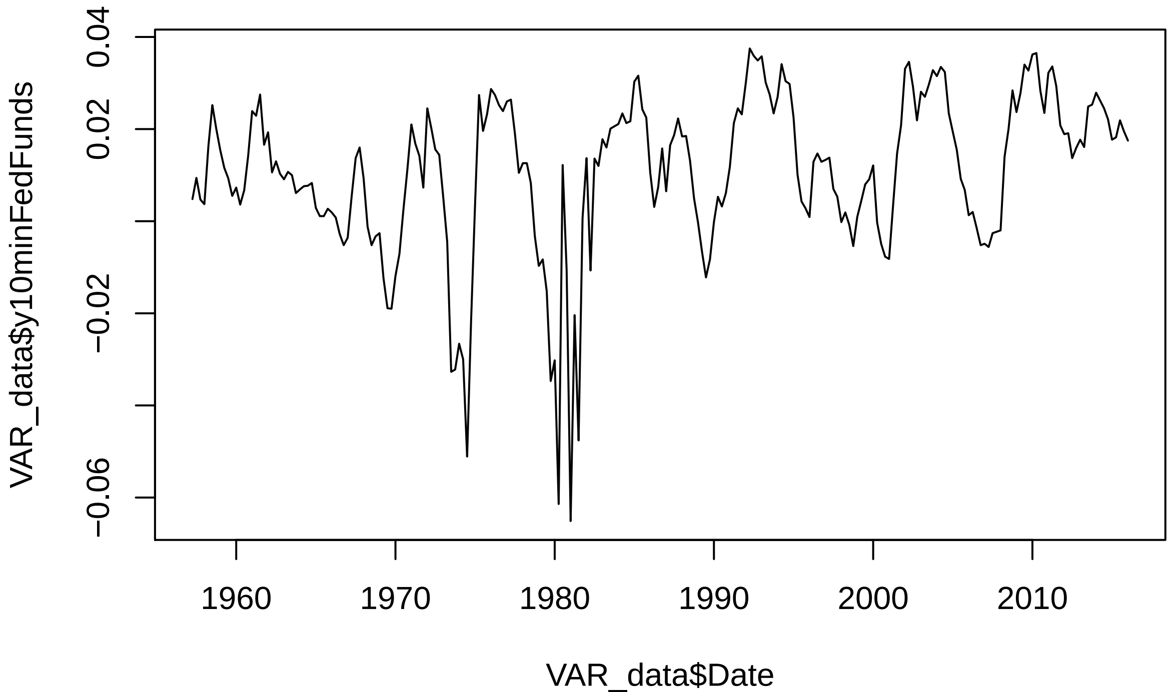
```r
VAR_data <- read_excel("~/Documents/Documents/Empirical Methods/MktRet_DP_TermSpread.xlsx")

plot(VAR_data$Date,VAR_data$MktExRet,type='l')
```



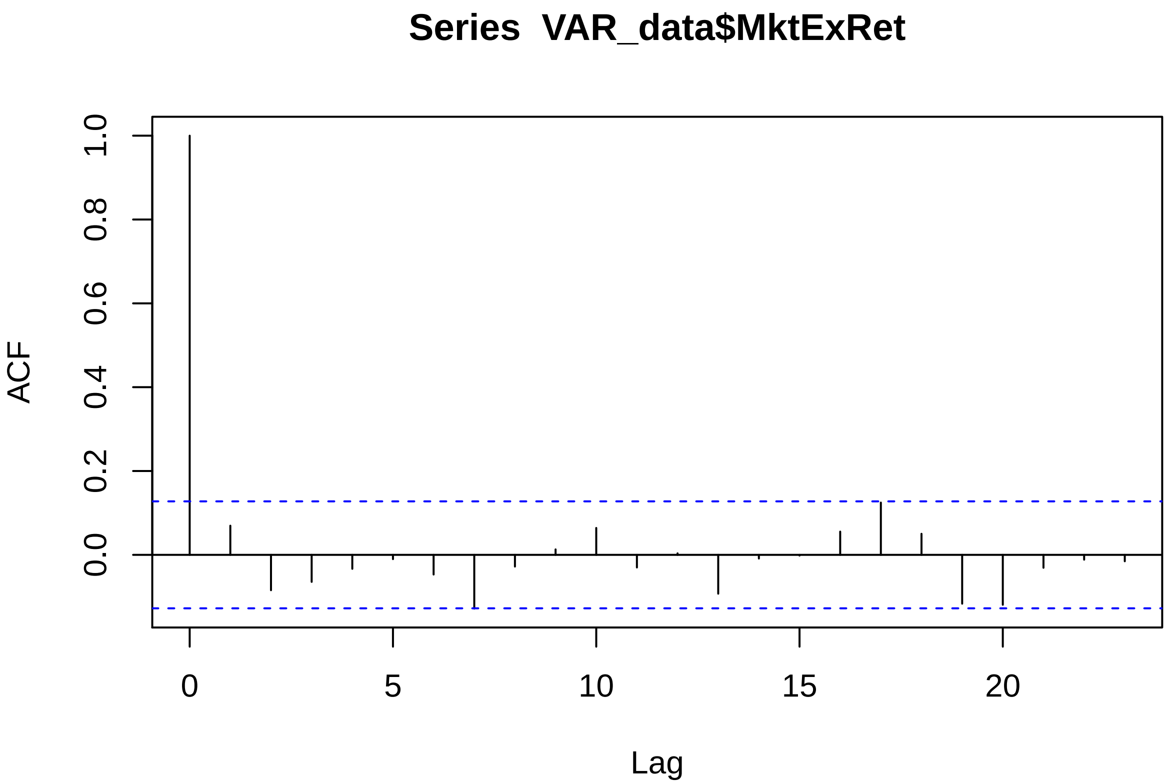

```
plot(VAR_data$Date,VAR_data$Mkt_DP,type='l')
```



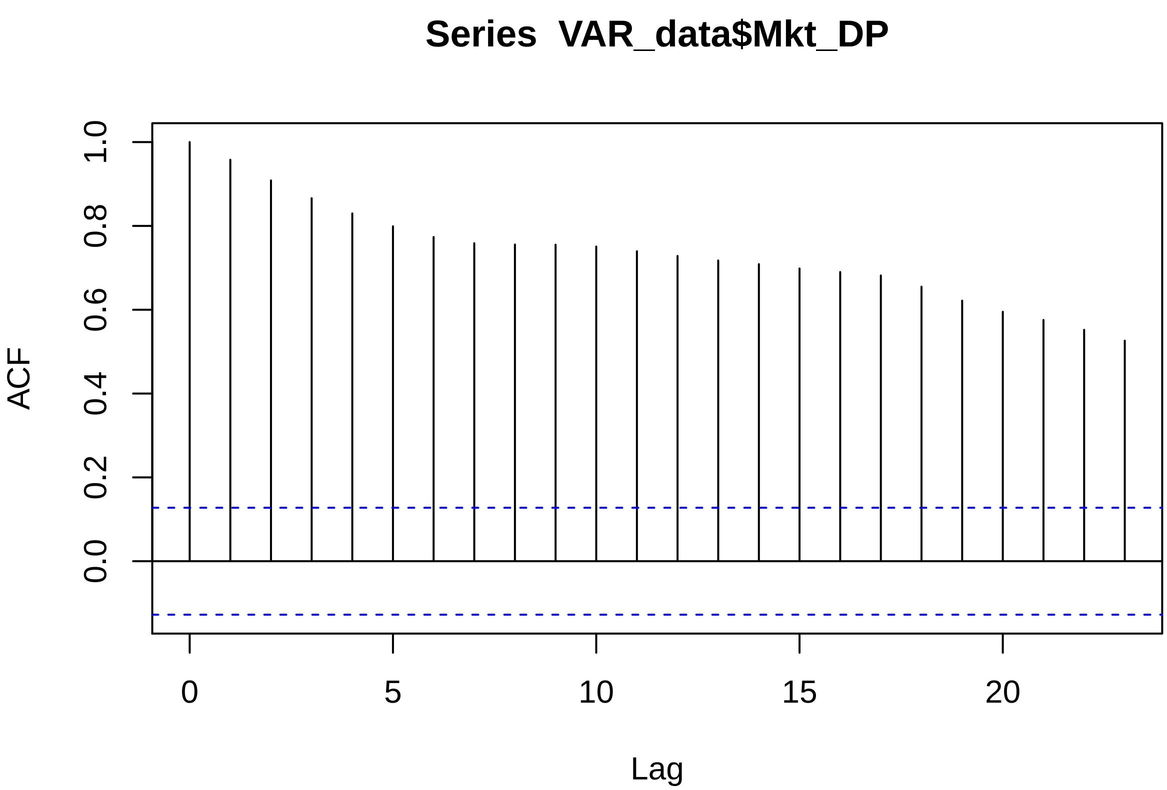

```
plot(VAR_data$Date,VAR_data$y10minFedFunds,type='l')
```



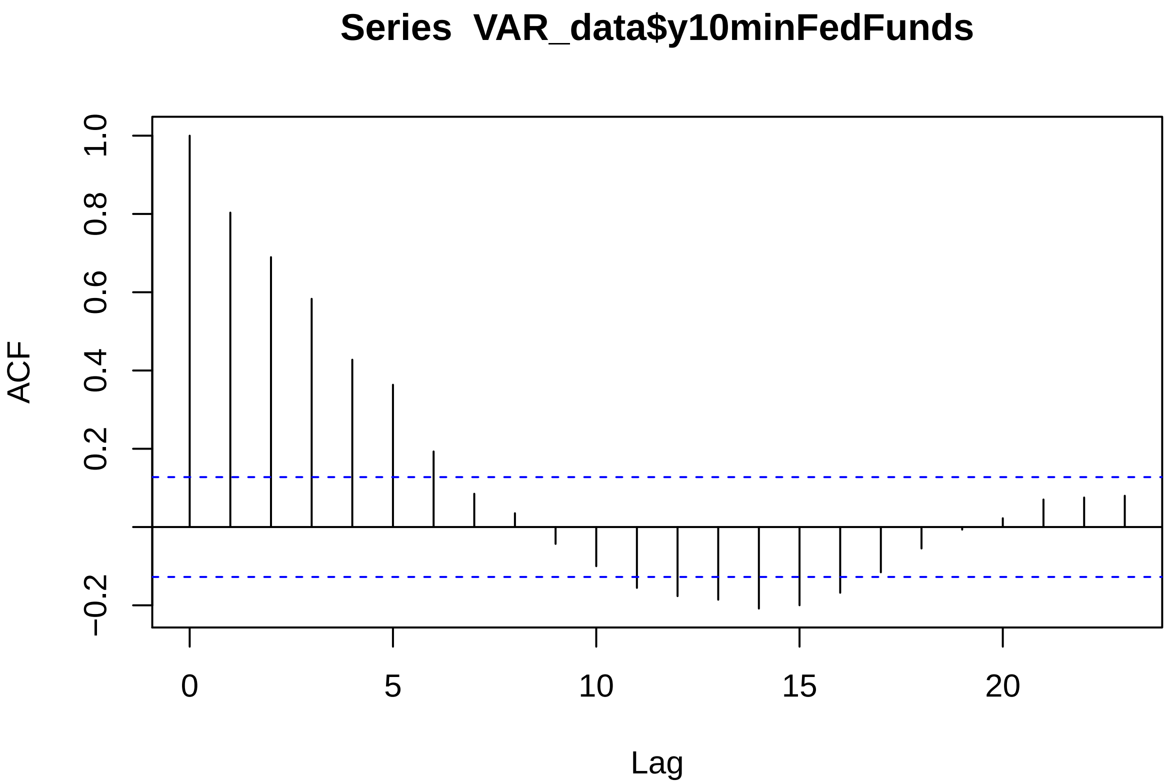

```
Mean_mkt <- mean(VAR_data$MktExRet)
Mean_dp <- mean(VAR_data$Mkt_DP)
Mean_y10ff <- mean(VAR_data$y10minFedFunds)

sd_mkt <- sd(VAR_data$MktExRet)
sd_dp <- sd(VAR_data$Mkt_DP)
sd_y10ff <- sd(VAR_data$y10minFedFunds)

ac_mkt <- acf(VAR_data$MktExRet)[1]
```




```
ac_dp <- acf(VAR_data$Mkt_DP)[1]
```




```
ac_y10ff <- acf(VAR_data$y10minFedFunds)[1]
```




```
hl_mkt <- log(0.5)/log(ac_mkt$acf)
hl_dp <- log(0.5)/log(ac_dp$acf)
hl_y10ff <- log(0.5)/log(ac_y10ff$acf)
```

2. Estimate a VAR(1). Give the coe¢ cient estimates, their White standard errors, and the R2 from each regression.

```
all_reg <- VAR(VAR_data[,2:4],p=1)

summary(all_reg)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: MktExRet, Mkt_DP, y10minFedFunds
## Deterministic variables: const
## Sample size: 235
## Log Likelihood: 2262.805
## Roots of the characteristic polynomial:
## 0.9407 0.7953 0.07442
## Call:
## VAR(y = VAR_data[, 2:4], p = 1)
##
##
## Estimation results for equation MktExRet:
## =========================================
## MktExRet = MktExRet.l1 + Mkt_DP.l1 + y10minFedFunds.l1 + const
##
##                   Estimate Std. Error t value Pr(>|t|)
## MktExRet.l1        0.04852    0.06554   0.740  0.45987
```

```
## Mkt_DP.l1          1.45092    0.54218   2.676  0.00798 **
## y10minFedFunds.l1  1.04958    0.33379   3.144  0.00188 **
## const             -0.03827    0.01805  -2.121  0.03501 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.08239 on 231 degrees of freedom
## Multiple R-Squared: 0.06072, Adjusted R-squared: 0.04852
## F-statistic: 4.977 on 3 and 231 DF,  p-value: 0.002297
##
##
## Estimation results for equation Mkt_DP:
## =======================================
## Mkt_DP = MktExRet.l1 + Mkt_DP.l1 + y10minFedFunds.l1 + const
##
##                     Estimate Std. Error t value Pr(>|t|)
## MktExRet.l1       -0.0022822  0.0021906  -1.042 0.298577
## Mkt_DP.l1          0.9402772  0.0181216  51.887  < 2e-16 ***
## y10minFedFunds.l1 -0.0388263  0.0111566  -3.480 0.000599 ***
## const              0.0021214  0.0006032   3.517 0.000526 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.002754 on 231 degrees of freedom
## Multiple R-Squared: 0.9298,  Adjusted R-squared: 0.9289
## F-statistic:  1021 on 3 and 231 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation y10minFedFunds:
## ===============================================
## y10minFedFunds = MktExRet.l1 + Mkt_DP.l1 + y10minFedFunds.l1 + const
##
##                    Estimate Std. Error t value Pr(>|t|)
## MktExRet.l1       -0.014836   0.008060  -1.841   0.0669 .
## Mkt_DP.l1          0.010477   0.066677   0.157   0.8753
## y10minFedFunds.l1  0.821641   0.041050  20.016   <2e-16 ***
## const              0.001872   0.002220   0.844   0.3998
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.01013 on 231 degrees of freedom
## Multiple R-Squared: 0.6516,  Adjusted R-squared: 0.647
## F-statistic:   144 on 3 and 231 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##                 MktExRet     Mkt_DP y10minFedFunds
## MktExRet        0.0067882 -2.058e-04      1.130e-04
## Mkt_DP         -0.0002058  7.583e-06     -4.058e-06
## y10minFedFunds  0.0001130 -4.058e-06      1.027e-04
```

```
##
## Correlation matrix of residuals:
##                MktExRet  Mkt_DP y10minFedFunds
## MktExRet         1.0000 -0.9073         0.1353
## Mkt_DP          -0.9073  1.0000        -0.1454
## y10minFedFunds   0.1353 -0.1454         1.0000
```

```
Robust_White <- coeftest(all_reg, vcov = vcovHC(all_reg, type="HC0"))
```

3. Is the VAR stationary?

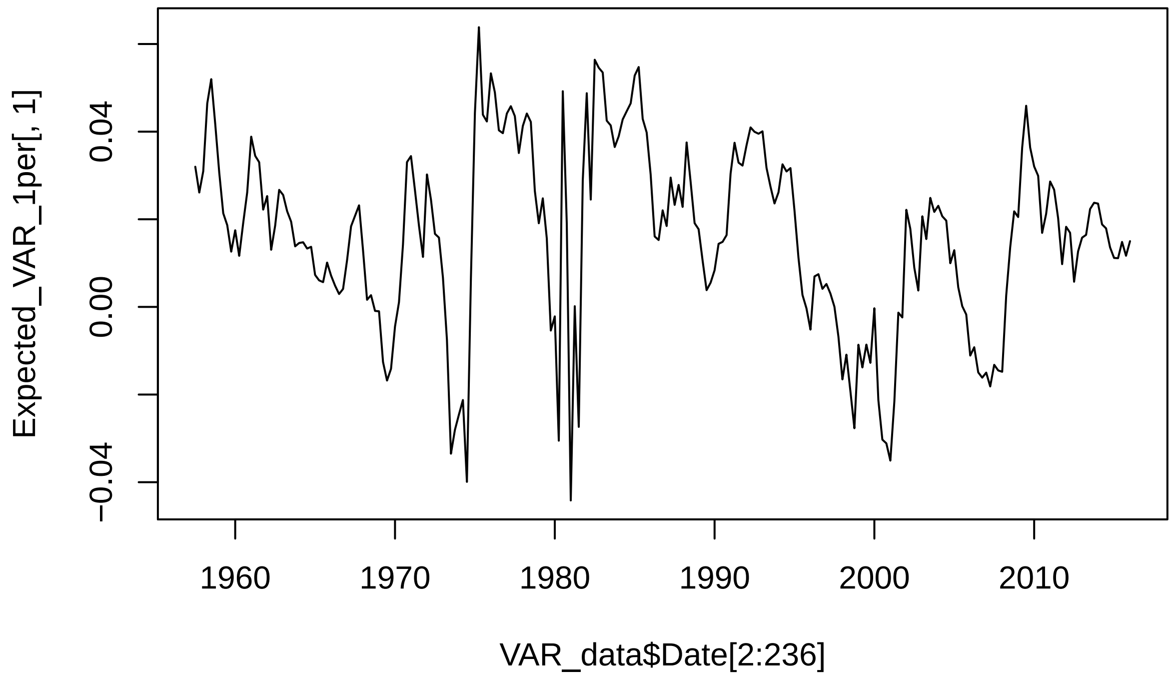
   Yes, the VAR is stationary as all auto-correlations are < 1.

4. What is the volatility of quarterly expected returns given the return forecasting regression?

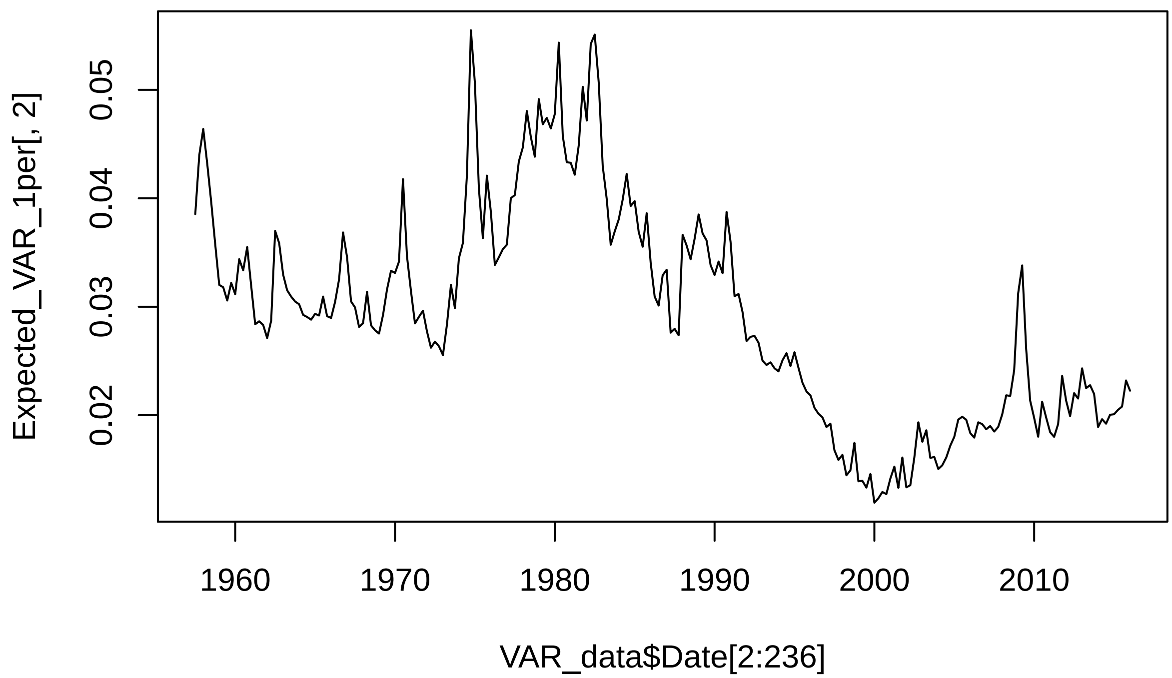
   From the summary of the VAR we see that the volatility (Regression Standard Error) is 8.24%.

5. Plot the one-quarter ahead expected return series. Plot the four quarters ahead expected return series. Plot the twenty quarters ahead expected return series. Comment on how the persistence of the term spread and the DP-ratio affects the expected return forecasts at different horizons.

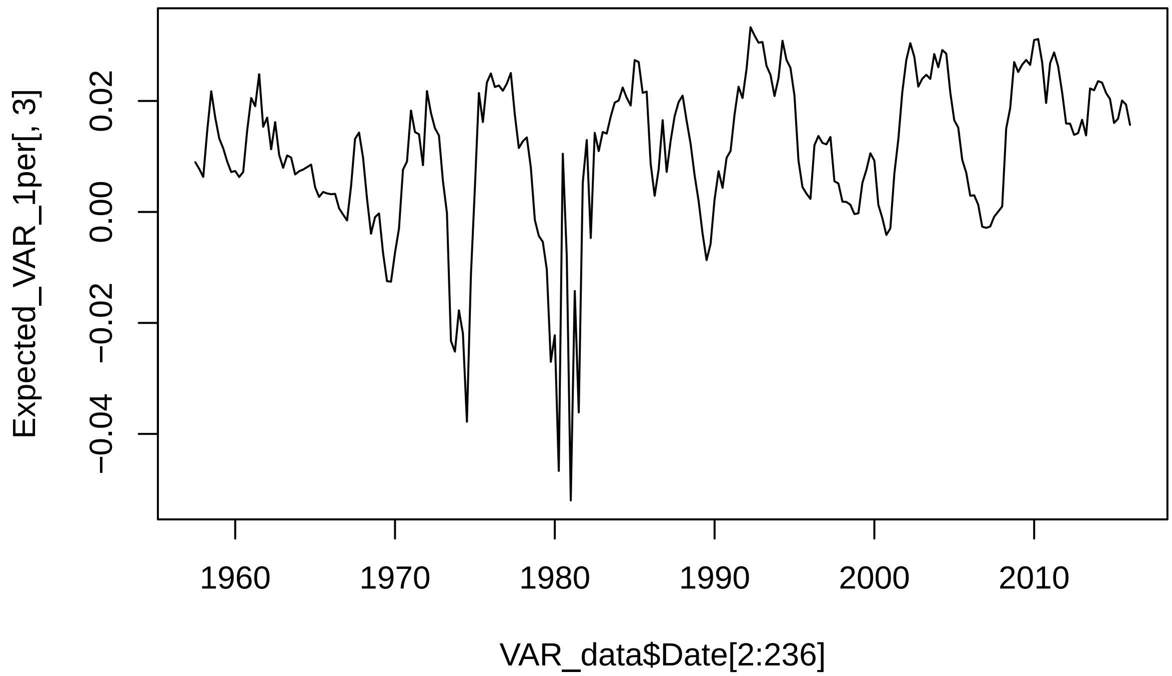
```
phi0 <- matrix(0, 3, 1)
phi1 <- matrix(0, 3, 3)
phi0[1,] <- all_reg$varresult$MktExRet$coefficients[4]
phi0[2,] <- all_reg$varresult$Mkt_DP$coefficients[4]
phi0[3,] <- all_reg$varresult$y10minFedFunds$coefficients[4]

phi1[1,] <- all_reg$varresult$MktExRet$coefficients[1:3]
phi1[2,] <- all_reg$varresult$Mkt_DP$coefficients[1:3]
phi1[3,] <- all_reg$varresult$y10minFedFunds$coefficients[1:3]

Expected_VAR <- as.matrix(VAR_data)
Expected_VAR <- Expected_VAR[2:length(VAR_data$MktExRet),2:4]

Expected_VAR_1per <- rep(phi0) +   phi1 %*% t(Expected_VAR)
Expected_VAR_1per <- t(Expected_VAR_1per)

plot(VAR_data$Date[2:236], Expected_VAR_1per[,1], type='l')
```



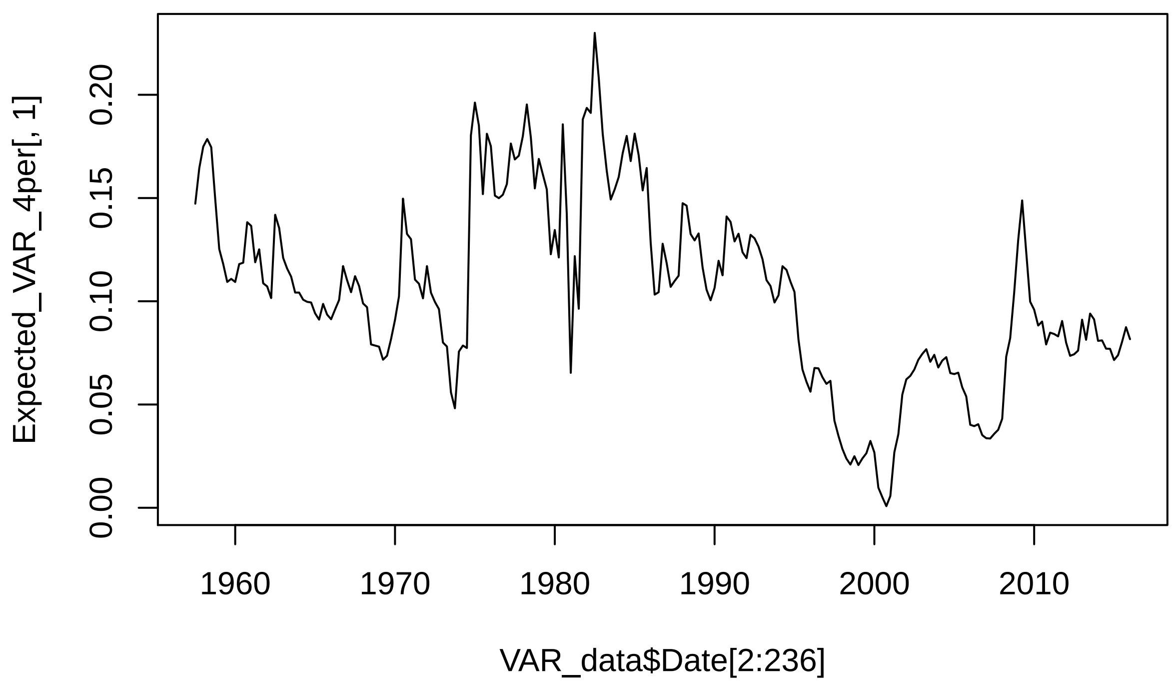

```
plot(VAR_data$Date[2:236], Expected_VAR_1per[,2], type='l')
```



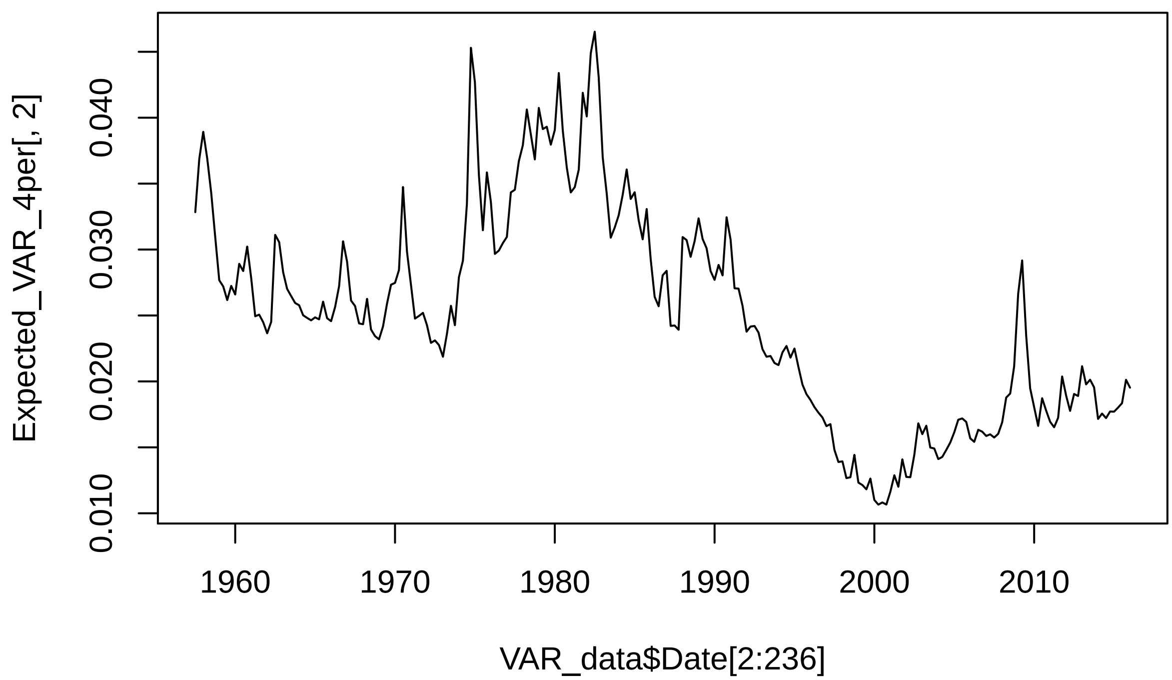

```
plot(VAR_data$Date[2:236], Expected_VAR_1per[,3], type='l')
```



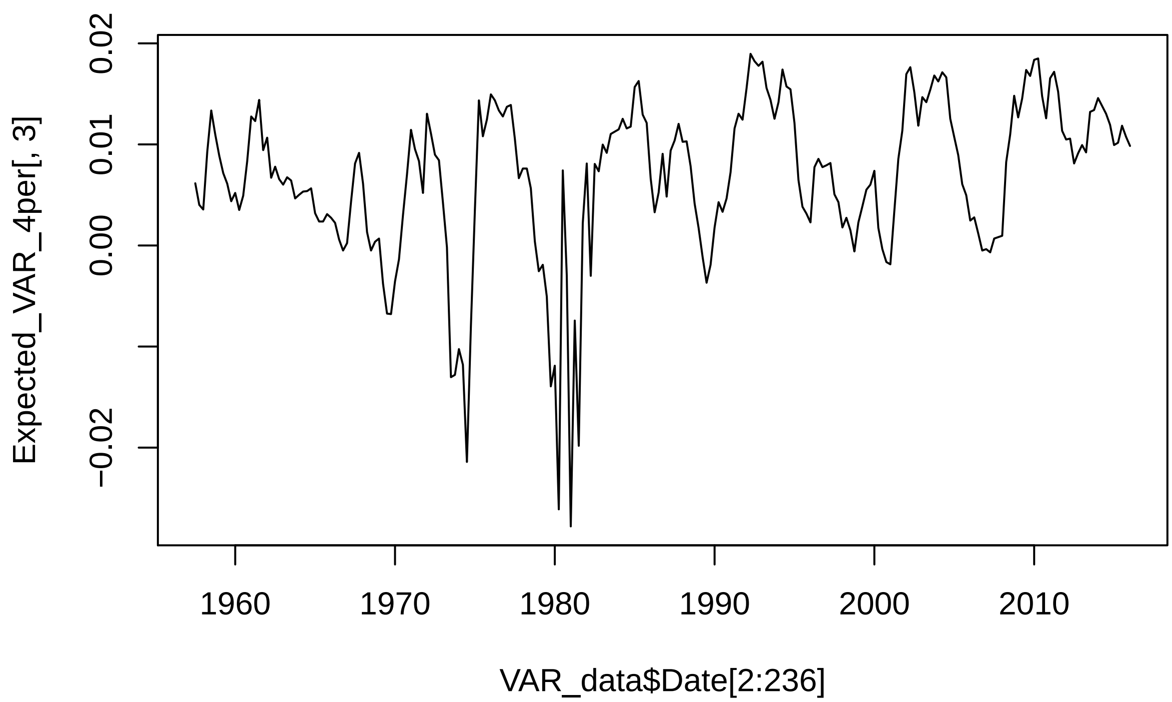

```
Expected_VAR_4per <- rep(phi0) +   phi1^4 %*% t(Expected_VAR)
Expected_VAR_4per <- t(Expected_VAR_4per)

plot(VAR_data$Date[2:236], Expected_VAR_4per[,1], type='l')
```



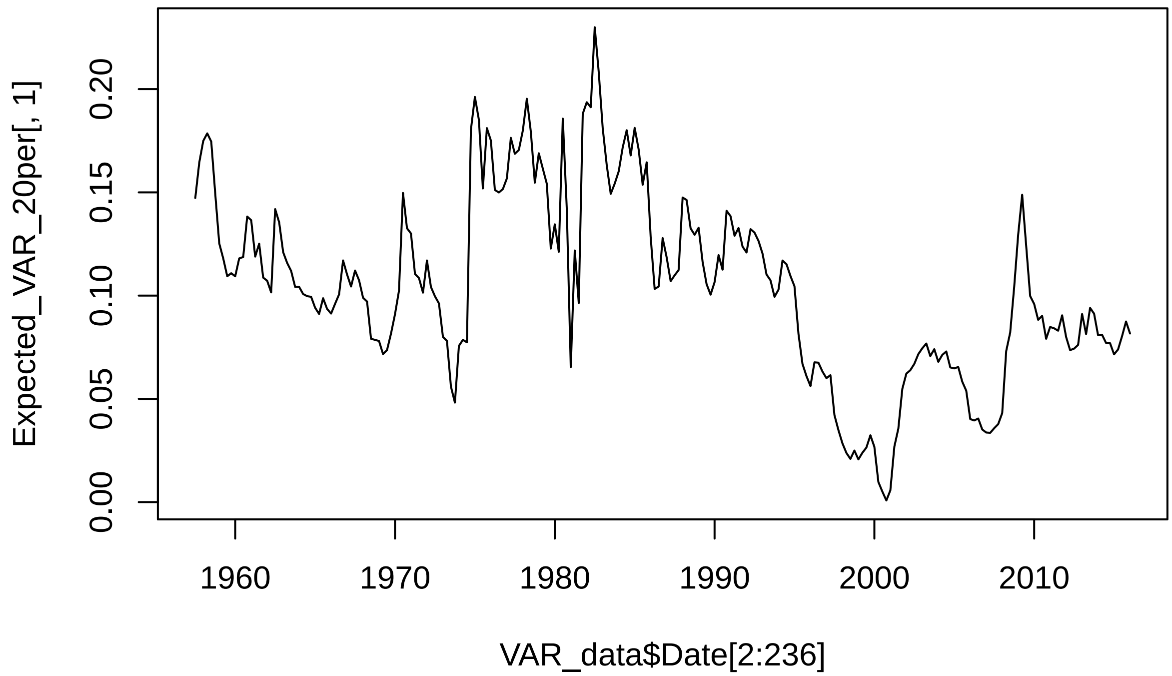

```
plot(VAR_data$Date[2:236], Expected_VAR_4per[,2], type='l')
```



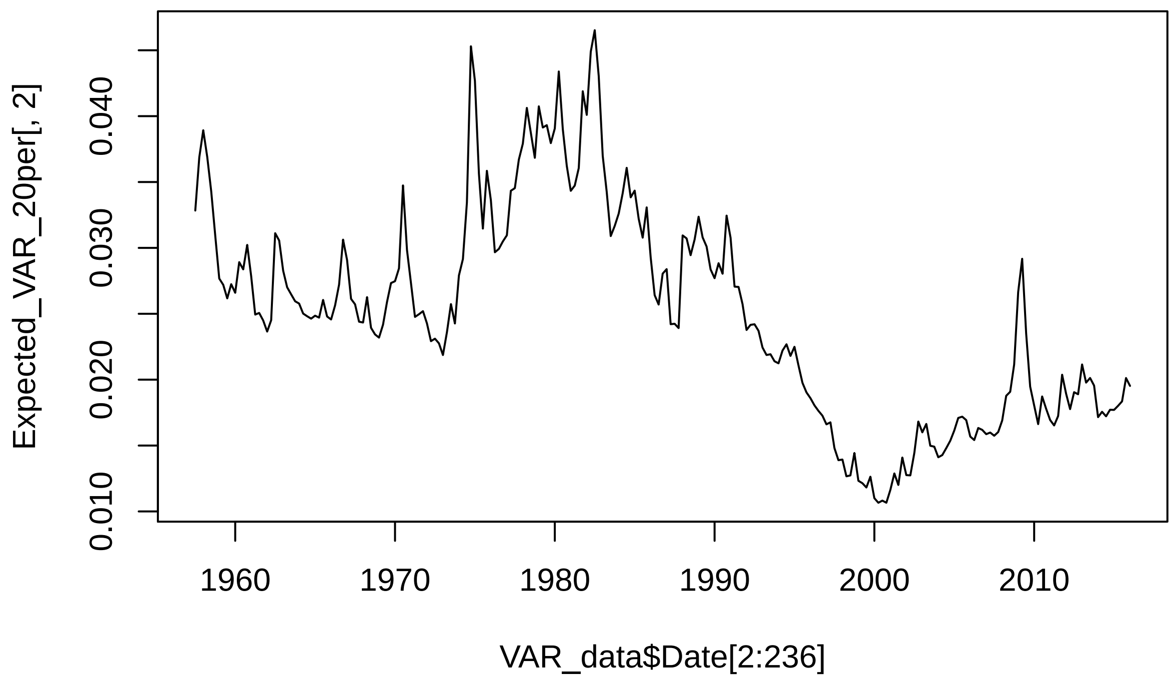

```
plot(VAR_data$Date[2:236], Expected_VAR_4per[,3], type='l')
```



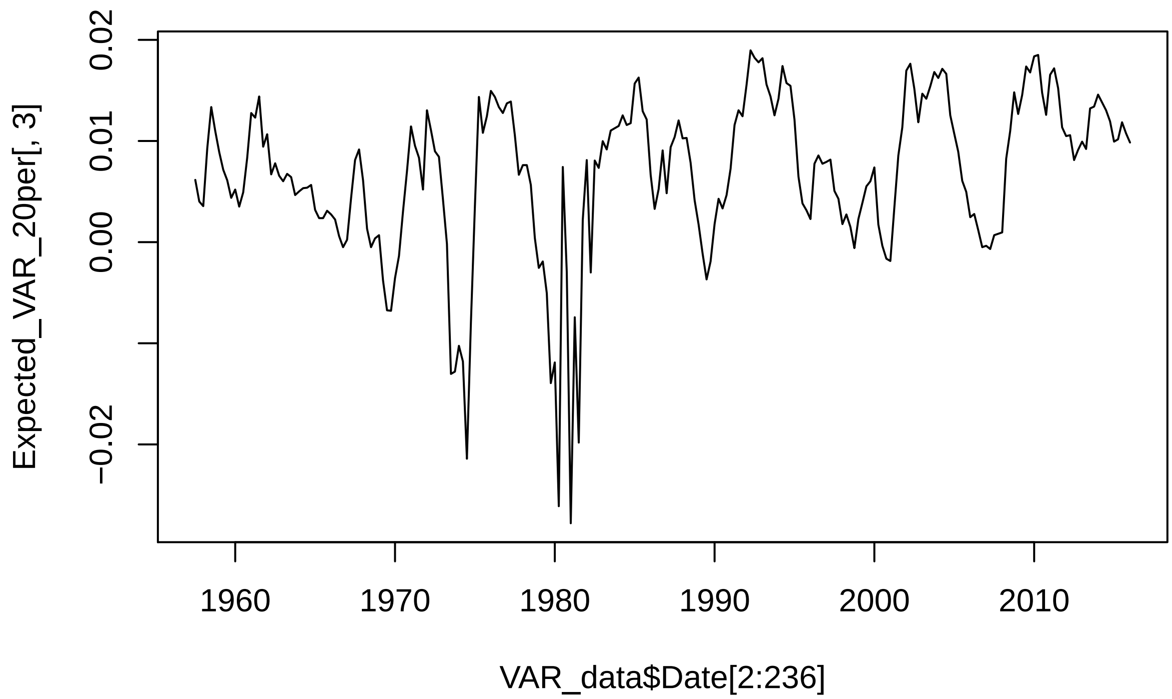

```
Expected_VAR_20per <- rep(phi0) + phi1^4 %*% t(Expected_VAR)
Expected_VAR_20per <- t(Expected_VAR_20per)

plot(VAR_data$Date[2:236], Expected_VAR_20per[,1], type='l')
```




```
plot(VAR_data$Date[2:236], Expected_VAR_20per[,2], type='l')
```

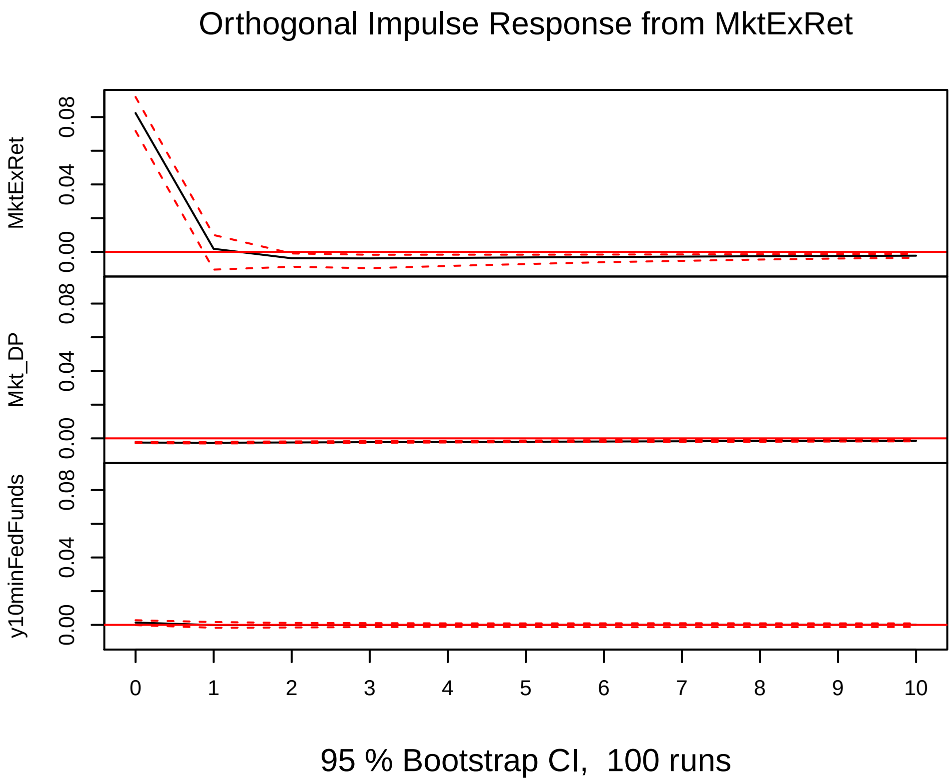


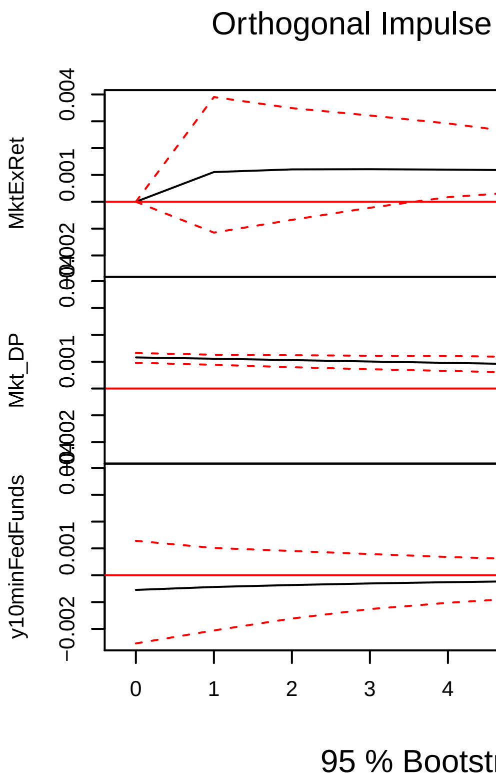

```
plot(VAR_data$Date[2:236], Expected_VAR_20per[,3], type='l')
```

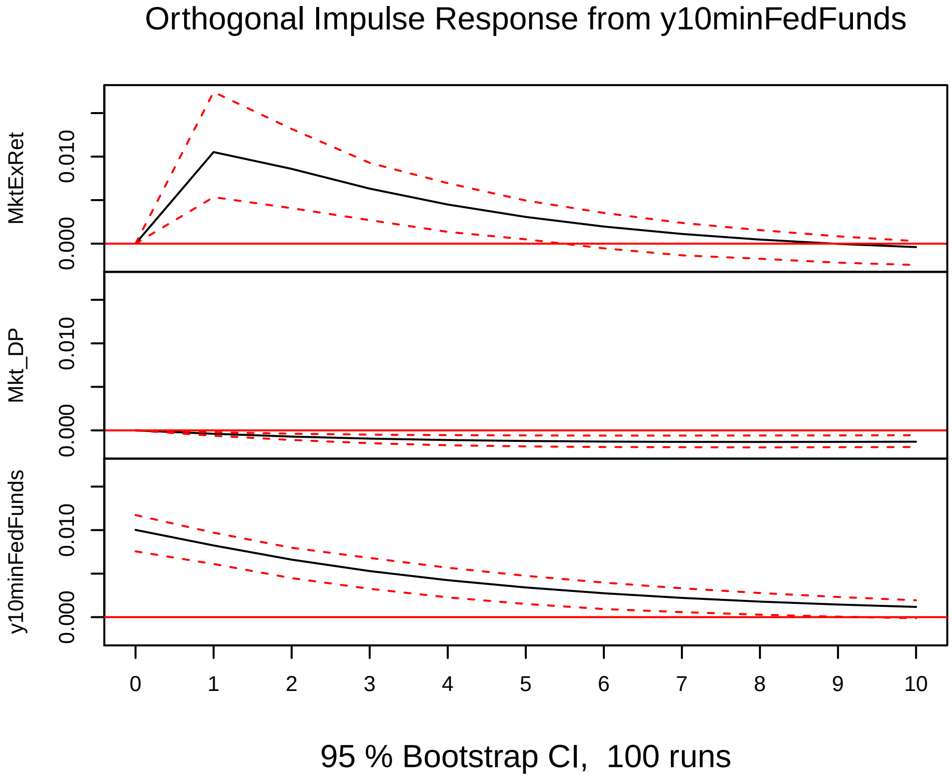


6. Plot the impulse-response function for returns from a one standard deviation positive shock from each of the three shocks in turn using 20 lags. You can do this by simulation. Start at unconditional averages for the lagged values of all the variables in the VAR (time t-1). Then set the time t shock in row 1 of the VAR to its one standard deviation value, and set all other current and future shocks to zero. Trace out the response by simulating future variables using the VAR dynamics. That ís the impulse-response for the first shock. Then go to the second shock and repeat the procedure just outlined for the first shock, but now set the second shock to its one standard deviation value and all other shocks to zero, etc. (it is best to plot the orthogonalized shock version of impulse-response (order the shocks Term Spread, D/P. and Returns), but it is also fine if you do it the simple way. The orthogonalized impulse-response math is given in the appendix).

```
Impulse_Response <- irf(all_reg)
plot(Impulse_Response)
```







7. Using 80% of the data as a training sample, report results from an out-of-sample test for predicting excess market returns where you re-estimate the model at each time t and get the prediction error for the t + 1 realizations.

```r
library("forecastSNSTS")
library("forecast")
```

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

```r
library("mosaic")
```

```
## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2

##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.

##
## Attaching package: 'mosaic'

## The following objects are masked from 'package:dplyr':
##
##     count, do, tally

## The following object is masked from 'package:Matrix':
##
##     mean

## The following object is masked from 'package:ggplot2':
##
##     stat

## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var

## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```r
library("mltools")

Trained_Reg <- lm(VAR_data$MktExRet[1:round(0.8*236)] ~ back(VAR_data$MktExRet[1:round(0.8*236)]) + bac

Out_of_Sample_Est <- 0
Out_of_Sample_Est[1:189] <- 0

for(i in 190:236){
  Out_of_Sample_Est[i] <- Trained_Reg$coefficients[1] + (Trained_Reg$coefficients[2]*VAR_data$MktExRet[
}

MSPE <- mse(preds = Out_of_Sample_Est[190:236], actuals = VAR_data$MktExRet[190:236])
MSPE
```

```
## [1] 0.006867407
```