

Lab No. 2 — Nonlinear Time Series Models for SGPE

Advanced Time Series Econometrics Labs

Ping Wu (ping.wu@strath.ac.uk (mailto:ping.wu@strath.ac.uk))

Outline of today's lab

- The Threshold autoregressive (TAR) model
- The Markov switching model

Packages we will use in this lab

- dplyr (<https://cran.r-project.org/web/packages/dplyr/dplyr.pdf>): A Grammar of Data Manipulation. Fast, consistent tool for working with data frame like objects, both in memory and out of memory.
- MSwM (<https://cran.r-project.org/web/packages/MSwM/MSwM.pdf>): Fitting Markov Switching Models. Includes estimation, inference and diagnostics for Univariate Autoregressive Markov Switching Models for Linear and Generalized Models.

```
# Important packages described above
install.packages("dplyr", repos = "https://cran.rstudio.com/",
  dependencies = TRUE)
install.packages("MSwM", repos = "https://cran.rstudio.com/",
  dependencies = TRUE)
```

The next step is to make sure that you can access the routines in this package by making use of the *library* command, which would need to be run regardless of the machine that you are using.

```
# Important packages described above
library(dplyr)
library(MSwM)
```

Part 0: Import data into R

In this lab, we will use the monthly data set for the SP500 from 1959 January to 2016 August.

The first thing that we do is clear all variables from the current environment and close all the plots. This is performed with the following commands:

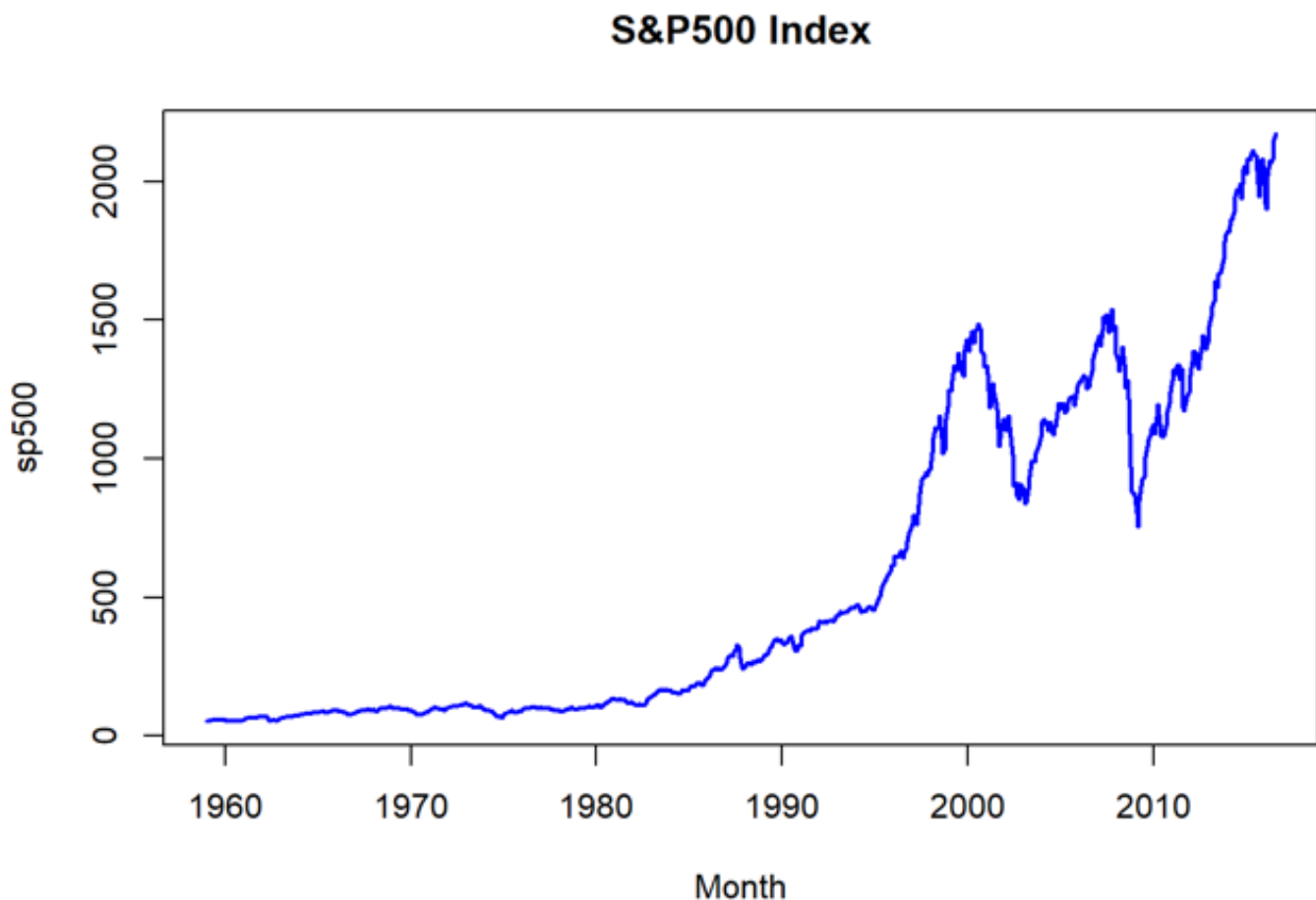
```
rm(list = ls())
graphics.off()
options(scipen = 9) # Avoid scientific notation
```

To import your downloaded data into R, you can:

```
# Allocate the variables to the object "sp500"
if("SP500.csv" %in% list.files()){
  sp500 <- read.table("SP500.csv", sep="," ,header=T)
}else{ # file.choose() allows to choose the file interactively
  sp500 <- read.table(file.choose(), sep="," ,header=T)
}
# Specify "sp500" as time series object
sp500 <- ts(sp500[, "SP500"], end = c(2016, 8), frequency = 12)
```

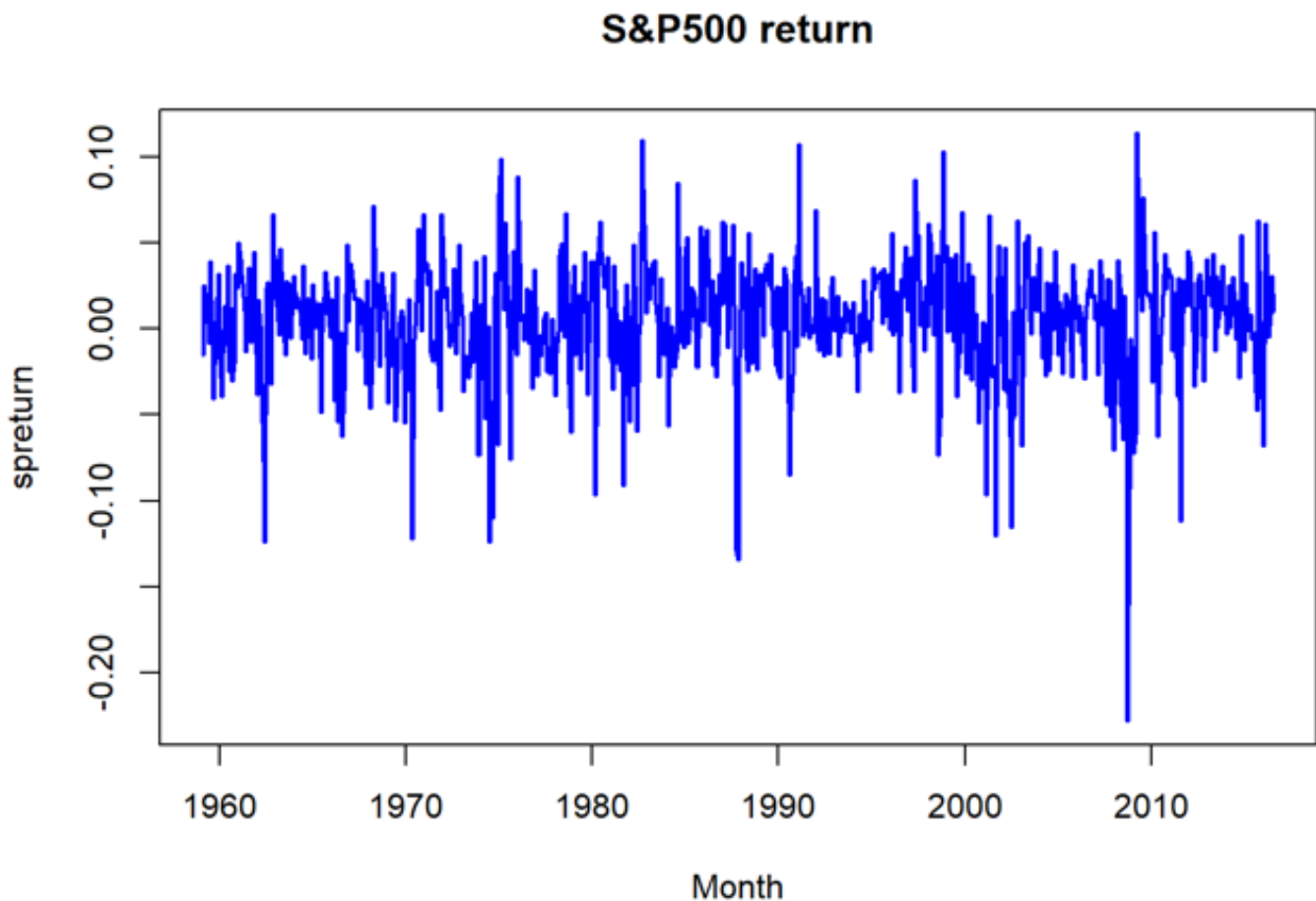
To make sure that the data has been imported correctly, we inspect a plot of the data:

```
ts.plot(sp500,
  main = "S&P500 Index",      # Title of plot
  xlab = "Month",              # Label of x-axis
  col = c("blue"),            # Define colours
  lty = "solid",               # Define line types
  lwd = 2)                     # Define line width)
```



You will see the index goes up over time (non-stationary), you might want to take the growth rate of the index. The growth rate can be interpreted as stock return and can be approximated by log difference

```
spreturn <- diff(log(sp500))  # taking the difference will lose the first data
# Specify "spreturn" as a new time series object
spreturn <- ts(spreturn, end = c(2016, 8), frequency = 12)
ts.plot(spreturn,
        main = "S&P500 return",      # Title of plot
        xlab = "Month",              # Label of x-axis
        col = c("blue"),            # Define colours
        lty = "solid",              # Define line types
        lwd = 2)                    # Define line width)
```



Part I: The Threshold autoregressive (TAR) model

The AR(1) model uses only the first lag of the dependent variable (spreturn) as explanatory variable:

$$y_t = c + \rho y_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, V).$$

The following code can estimate the AR(1) model. It will print the estimate of intercept, coefficient, and error variance. Can you match the estimate to the parameters in the above equation?

```
arl_reg <- ar.ols(spreturn, order.max = 1,
                 demean = F,
                 intercept = T)
arl_reg
```

```
##
## Call:
## ar.ols(x = spreturn, order.max = 1, demean = F, intercept = T)
##
## Coefficients:
##      1
## 0.2435
##
## Intercept: 0.004043 (0.00133)
##
## Order selected 1  sigma^2 estimated as  0.001194
```

To compute the information criteria

```
T <- length(spreturn)           # Number of observations
residuals <- ar1_reg$resid[3:T] # Get the residuals
sigma_sq <- var(residuals)       # Compute the estimated variance of the residuals
# Now compute the log likelihood
loglik <- -T / 2 * log(2 * pi) - T / 2 * log(sigma_sq) - sum(residuals^2) / (2 * sigma_sq)
n.para <- 2
r.aic <- (-2 * (loglik)) + 2 * (sum(n.para))
r.bic <- (-2 * (loglik)) + (log(length(spreturn))) * (n.para)
cat("LL", loglik)
```

```
## LL 1345.772
```

```
cat("AIC", r.aic)
```

```
## AIC -2687.545
```

```
cat("BIC", r.bic)
```

```
## BIC -2678.468
```

A homoskedastic TAR model is defined as

$$y_t = \begin{cases} \rho_1 y_{t-1} + \varepsilon_t, & \varepsilon_t \sim \mathcal{N}(0, \sigma^2), & \text{if } z_t \leq \tau \\ \rho_2 y_{t-1} + \varepsilon_t, & \varepsilon_t \sim \mathcal{N}(0, \sigma^2), & \text{if } z_t > \tau \end{cases}$$

First create dummy variable

```
data <- data.frame(spreturn) # create a data frame with the column spreturn
data$SP_thresh1 <- 0         # Initialize SP_thresh1 as 0
# Replace SP_thresh1 with spreturn where spreturn > 0
data$SP_thresh1[data$spreturn > 0] <- data$spreturn[data$spreturn > 0]
```

Do the regression

```
model = ts.intersect(data$spreturn, lagly=stats::lag(data$spreturn,-1), lag2y=data$
SP_thresh1[1:T-1])
homotar = lm(model[,1]~model[,2]+model[,3])
summary(homotar)
```

```
##
## Call:
## lm(formula = model[, 1] ~ model[, 2] + model[, 3])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.220103 -0.016582  0.002407  0.021191  0.123608
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.003210   0.002009   1.598 0.110582
## model[, 2]   0.215871   0.062139   3.474 0.000545 ***
## model[, 3]   0.061122   0.110373   0.554 0.579916
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03462 on 687 degrees of freedom
## Multiple R-squared:  0.05975,    Adjusted R-squared:  0.05701
## F-statistic: 21.83 on 2 and 687 DF,  p-value: 0.00000000006443
```

If I do not give you the information criteria, can you make a decision on Is there evidence of threshold nonlinearity in this variable? If so, which parameters are changing across regimes? The key variable is the dummy variable which is the third column in *model*. From the p-value (0.5799), we can say that this dummy variable is not significantly different from zero, which means there is no break in mean.

To compute the information criteria

```
T <- length(spreturn)           # Number of observations
residuals <- homotar$residuals  # Get the residuals
sigma_sq <- var(residuals)      # Compute the estimated variance of the residuals
# Now compute the log likelihood
loglik <- -T / 2 * log(2 * pi) - T / 2 * log(sigma_sq) - sum(residuals^2) / (2 * sigma_sq)
n.para <- 2
r.aic <- (-2 * (loglik)) + 2 * (sum(n.para))
r.bic <- (-2 * (loglik)) + (log(length(spreturn))) * (n.para)
cat("LL", loglik)
```

```
## LL 1345.671
```

```
cat("AIC", r.aic)
```

```
## AIC -2687.342
```

```
cat("BIC", r.bic)
```

```
## BIC -2678.265
```

Does your conclusion change based on information criteria?

A heterokedastic TAR model is defined as

$$y_t = \begin{cases} \rho_1 y_{t-1} + \varepsilon_{1t}, & \varepsilon_{1t} \sim \mathcal{N}(0, \sigma_1^2), & \text{if } z_t \leq \tau \\ \rho_2 y_{t-1} + \varepsilon_{2t}, & \varepsilon_{2t} \sim \mathcal{N}(0, \sigma_2^2), & \text{if } z_t > \tau \end{cases}$$

The first subset of the data

```
less <- model[,3] == 0
modell <- model[less,]
Tless <- length(model[less,1])
heter1 = lm(modell[,1]~modell[,2])
summary(heter1)
```

```
##
## Call:
## lm(formula = modell[, 1] ~ modell[, 2])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.220708 -0.022148  0.002602  0.024317  0.123770
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.007803   0.003343   2.334 0.020339 *
## modell[, 2]  0.293062   0.083394   3.514 0.000516 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04089 on 272 degrees of freedom
## Multiple R-squared:  0.04343,    Adjusted R-squared:  0.03991
## F-statistic: 12.35 on 1 and 272 DF,  p-value: 0.0005165
```

The second subset of the data

```
more <- model[,3] > 0
model2 <- model[more,]
Tmore <- length(model[more,1])
heter2 = lm(model2[,1]~model2[,2])
summary(heter2)
```

```
##
## Call:
## lm(formula = model2[, 1] ~ model2[, 2])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.123175 -0.013858  0.002551  0.018188  0.109932
##
## Coefficients:
##              Estimate Std. Error t value    Pr(>|t|)
## (Intercept) -0.001457   0.002437  -0.598      0.55
## model2[, 2]  0.390452   0.073717   5.297 0.000000192 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02957 on 414 degrees of freedom
## Multiple R-squared:  0.06346,    Adjusted R-squared:  0.0612
## F-statistic: 28.05 on 1 and 414 DF,  p-value: 0.000000192
```

To get the overall information criteria, you need to take the sum of the two

```
# The first subset
residuals <- heter1$residuals      # Get the residuals
sigma_sq <- var(residuals)         # Compute the estimated variance of the residuals
loglik1 <- -Tless / 2 * log(2 * pi) - Tless / 2 * log(sigma_sq) - sum(residuals^2) / (2 * sigma_sq)
n.para <- 2
r.aic1 <- (-2 * (loglik1)) + 2 * (sum(n.para))
r.bic1 <- (-2 * (loglik1)) + (log(Tless)) * (n.para)
# The second subset
residuals <- heter2$residuals      # Get the residuals
sigma_sq <- var(residuals)         # Compute the estimated variance of the residuals
loglik2 <- -Tmore / 2 * log(2 * pi) - Tmore / 2 * log(sigma_sq) - sum(residuals^2) / (2 * sigma_sq)
n.para <- 2
r.aic2 <- (-2 * (loglik2)) + 2 * (sum(n.para))
r.bic2 <- (-2 * (loglik2)) + (log(Tmore)) * (n.para)
cat("LL", loglik1+loglik2)
```

```
## LL 1363.596
```

```
cat("AIC", r.aic1+r.aic2)
```

```
## AIC -2719.192
```

```
cat("BIC", r.bic1+r.bic2)
```

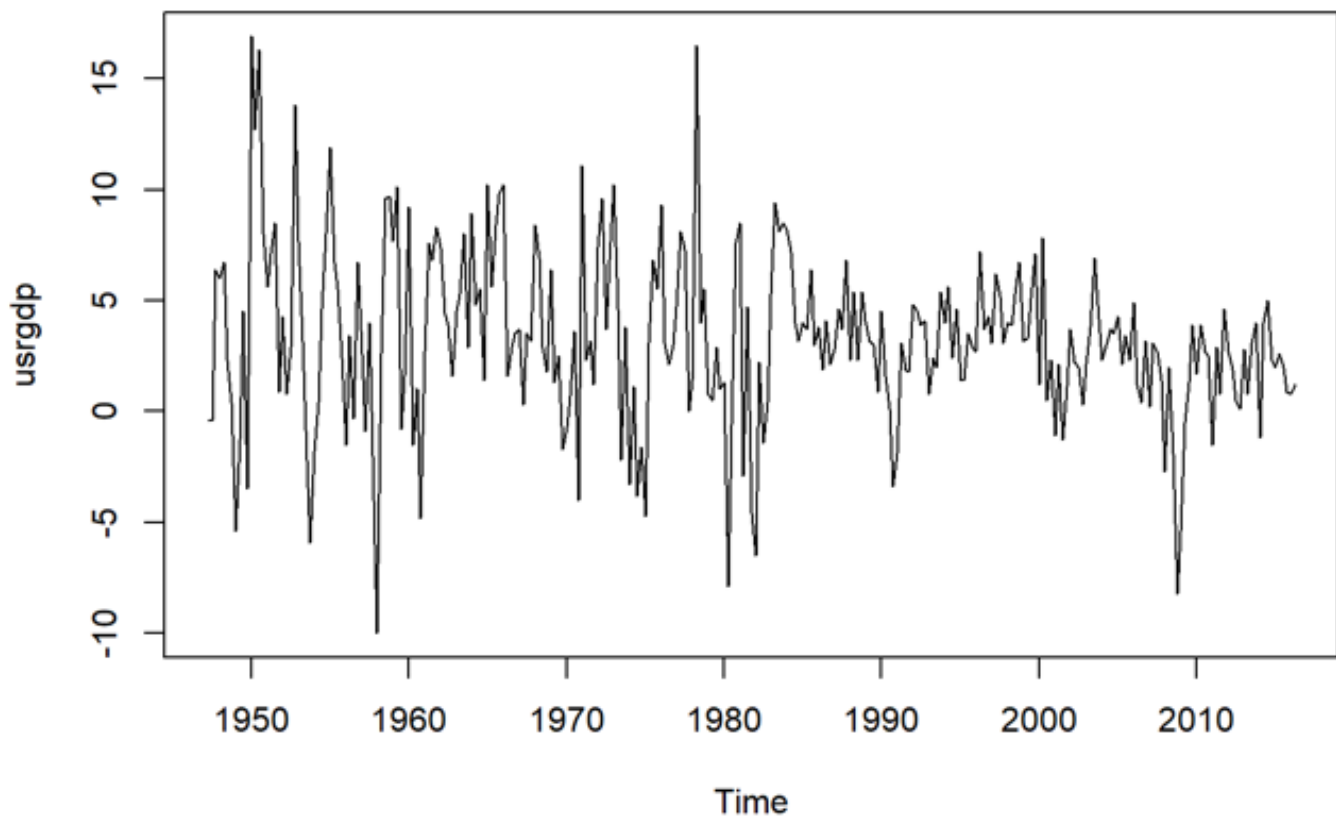
```
## BIC -2703.904
```

Can you tell me which model is best: AR(1), homoskedastic TAR, or heterokedastic TAR?

Part II: The Markov switching model (MS model)

First import data into R. The series we will use for MS is the real GDP growth rate:

```
if("US_rgdg_change.csv" %in% list.files()){
  usrgdp <- read.table("US_rgdg_change.csv", sep=",", header=T)
}else{ # file.choose() allows to choose the file interactively
  usrgdp <- read.table(file.choose(), sep=",", header=T)
}
usrgdp <- ts(usrgdp[, "RGDP_CH"], end = c(2016, 2), frequency = 4)
plot(usrgdp)
```



Next, we fit the data using an Autoregressive Markov Switching Model

```
usrgdp.lags <- embed(usrgdp, 3) # Create two lags of the variables
usrgdp.lags <- as.data.frame(usrgdp.lags)
colnames(usrgdp.lags) <- c("RGDP", "RGDP11", "RGDP12")
mod=lm(RGDP ~ RGDP11 + RGDP12, data = usrgdp.lags)
summary(mod)
```



```
##
## Call:
## lm(formula = RGDP ~ RGDP11 + RGDP12, data = usrgdp.lags)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.9011  -1.9416  -0.0848   2.0494  15.7837
##
## Coefficients:
##              Estimate Std. Error t value    Pr(>|t|)
## (Intercept)   1.85529     0.31020   5.981 0.00000000696 ***
## RGDP11         0.33443     0.06032   5.544 0.00000006982 ***
## RGDP12         0.09589     0.06027   1.591      0.113
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.639 on 272 degrees of freedom
## Multiple R-squared:  0.1452, Adjusted R-squared:  0.139
## F-statistic: 23.11 on 2 and 272 DF,  p-value: 0.0000000005383
```

```
mod.mswm = msmFit(mod,
                  k=2,                                # Number of regimes
                  sw=c(TRUE,TRUE,TRUE,TRUE),          # Intercept, 2 AR coeff, error varia
nce
                  control=list(parallel=FALSE))
summary(mod.mswm)
```

```

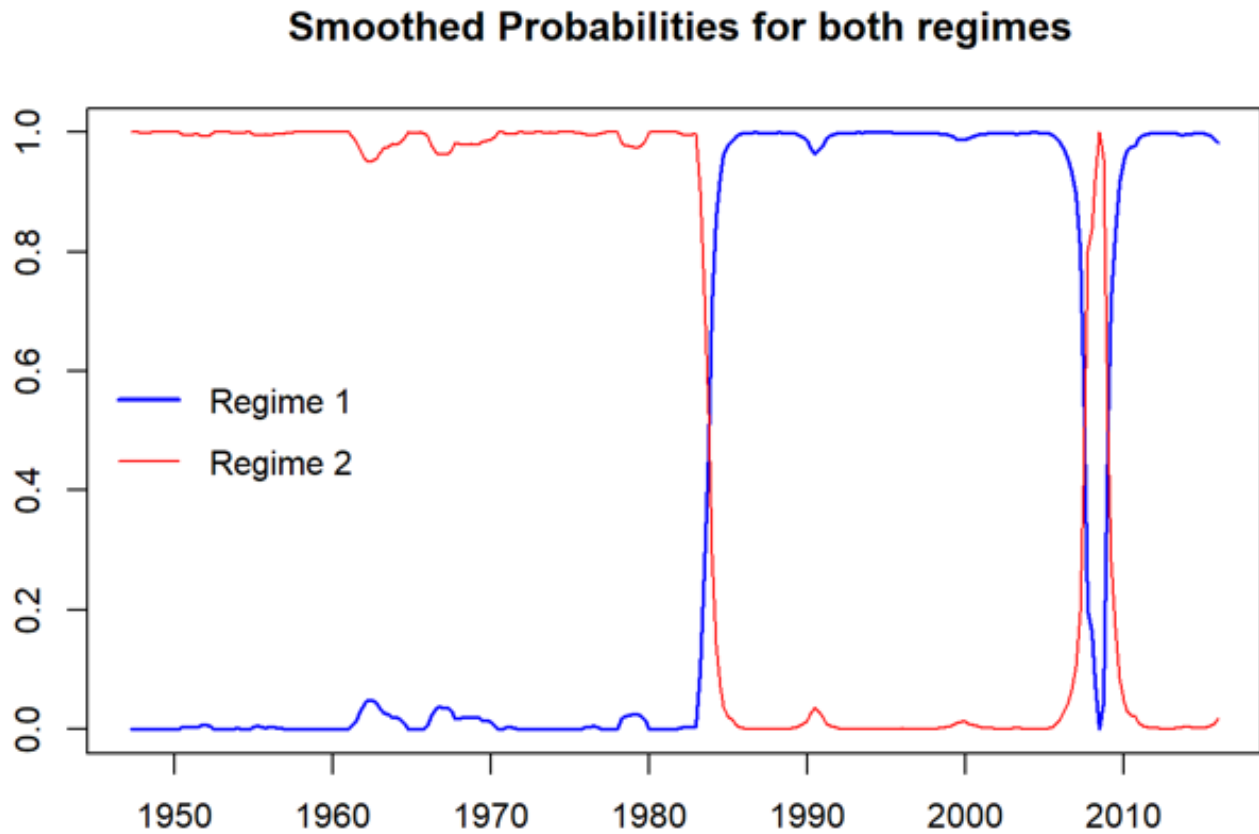
## Markov Switching Model
##
## Call: msmFit(object = mod, k = 2, sw = c(TRUE, TRUE, TRUE, TRUE), control = lis
t(parallel = FALSE))
##
##           AIC      BIC    logLik
##    1430.389 1485.79 -709.1944
##
## Coefficients:
##
## Regime 1
## -----
##           Estimate Std. Error t value    Pr(>|t|)
## (Intercept)(S)    1.6375     0.3687  4.4413 0.000008942 ***
## RGDP11(S)         0.1766     0.0952  1.8550   0.063596 .
## RGDP12(S)         0.2652     0.0836  3.1722   0.001513 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.877974
## Multiple R-squared: 0.1695
##
## Standardized Residuals:
##           Min           Q1           Med           Q3           Max
## -5.378119921370 -0.284832683269  0.000001464612  0.241296422173  4.311914007317
##
## Regime 2
## -----
##           Estimate Std. Error t value    Pr(>|t|)
## (Intercept)(S)    2.0385     0.4891  4.1679 0.00003074 ***
## RGDP11(S)         0.3582     0.0815  4.3951 0.00001107 ***
## RGDP12(S)         0.0620     0.0823  0.7533   0.4513
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.546805
## Multiple R-squared: 0.1462
##
## Standardized Residuals:
##           Min           Q1           Med           Q3           Max
## -10.85339729 -0.64562220 -0.01862763  0.60747036 15.83650327
##
## Transition probabilities:
##           Regime 1    Regime 2
## Regime 1 0.98152491 0.008556231
## Regime 2 0.01847509 0.991443769

```

```

probr1 <- ts(mod.mswm@Fit@smoProb, end = c(2016, 1), frequency = 4)
plot.ts(probr1[,1], main = " Smoothed Probabilities for both regimes", col = "blue", xlab = "", ylab = "", lwd = 1.5)
lines(probr1[,2], col = "red")
legend("left", legend = c("Regime 1", "Regime 2"),
      lwd = c(2, 1), col = c("blue", "red"), bty = "n", y.intersp = 1.5)

```



How would you interpret the two regimes? (low volatility and high volatility?)