Extra Credit Project

Marc Luiz, Nelly Shieh, Jeff Nguyen

18/09/2020

Coitegration Analysis of Different Financial Markets
University of Southern California
Marshall School of Business
FBE 543 Forecasting and Risk Analysis
Spring 2021
Directed by Professor Mohammad Safarzadeh

Topic:

Using major stock indices for the United States (S&P500), Unite Kingdom (FTSE100), Germany (DAX), and France (CAC40) show that US and European financial markets conintegrate.

Method:

Reviewing cointegration condition:

Consider 2 time series variables Y and X. We have the regression equation as follow:

$$Y_t = \beta_0 + \beta_1 X_t + \epsilon_t \tag{1}$$

The cointegration is such that if X and Y are both non-stationary variables AND ϵ is a stationary variable, then X and Y cointegrate, i.e. they "move together" in the long run.

Thus, our primary methodology is as follow:

1. Test indices for stationarity:

Test representative stock indices in the United States (S&P 500), the United Kingdom (FTSE), Germany (DAX) and France (CAC40) for stationarity.
To execute this task, for each indices, we run the Augmented Dickey-Fuller Test (A)DF, which hypothesizes that a unit root is present in an autoregressive model. The intuition is such that if a variable is stationary, it tends to a constant mean–i.e. the values oscillates/ alternate for large to small. As a result, the process is not a random walk, i.e. nonstationary.

2. Test error term for stationarity:

Should the regressed result confirm non-stationarity, check whether the error term of the regression ϵ are non-stationary variables. If they are, then the indices cointegrate.

First we check for the common issue with time series data: positive autocorrelation by running Durbin-Watson Test. If autocorrelation exists, we add the first order autoregressive term $AR(1)$ into the model and subsequently $AR(2)$ as necessary.

Then, we run (A)DF test as above to test the error term for stationarity–not having a unit root in the (A)DF test.

Data Analysis

For this model, we use 20 years of data from March 2001 to March 2021.

Downloading data:

```
library(quantmod)
```

```
## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

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

## Loading required package: TTR

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

## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```
# Set start date and end date of data
start_date <- "2001-01-01"
end_date <- "2021-03-18"

# Get data
getSymbols("^GSPC", src = "yahoo", , from = start_date, to = end_date) # S&P 500
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
## [1] "^GSPC"
```

```
getSymbols("^FTSE", src = "yahoo", , from = start_date, to = end_date) # S&P 500
```

```
## Warning: ^FTSE contains missing values. Some functions will not work if objects
## contain missing values in the middle of the series. Consider using na.omit(),
## na.approx(), na.fill(), etc to remove or replace them.
```

```
## [1] "^FTSE"
```

```
getSymbols("^GDAXI", src = "yahoo", , from = start_date, to = end_date) # S&P 500
```

```
## Warning: ^GDAXI contains missing values. Some functions will not work if objects
## contain missing values in the middle of the series. Consider using na.omit(),
## na.approx(), na.fill(), etc to remove or replace them.
```

```
## [1] "^GDAXI"
```

```
getSymbols("^FCHI", src = "yahoo", , from = start_date, to = end_date) # S&P 500
```

```
## Warning: ^FCHI contains missing values. Some functions will not work if objects
## contain missing values in the middle of the series. Consider using na.omit(),
## na.approx(), na.fill(), etc to remove or replace them.
```

```
## [1] "^FCHI"
```

```
# Adjusted Prices
adjGSPC_mo <- to.monthly(GSPC)$GSPC.Adjusted
adjFTSE_mo <- to.monthly(FTSE)$FTSE.Adjusted
```

```
## Warning in to.period(x, "months", indexAt = indexAt, name = name, ...): missing
## values removed from data
```

```
adjGDAXI_mo <- to.monthly(GDAXI)$GDAXI.Adjusted
```

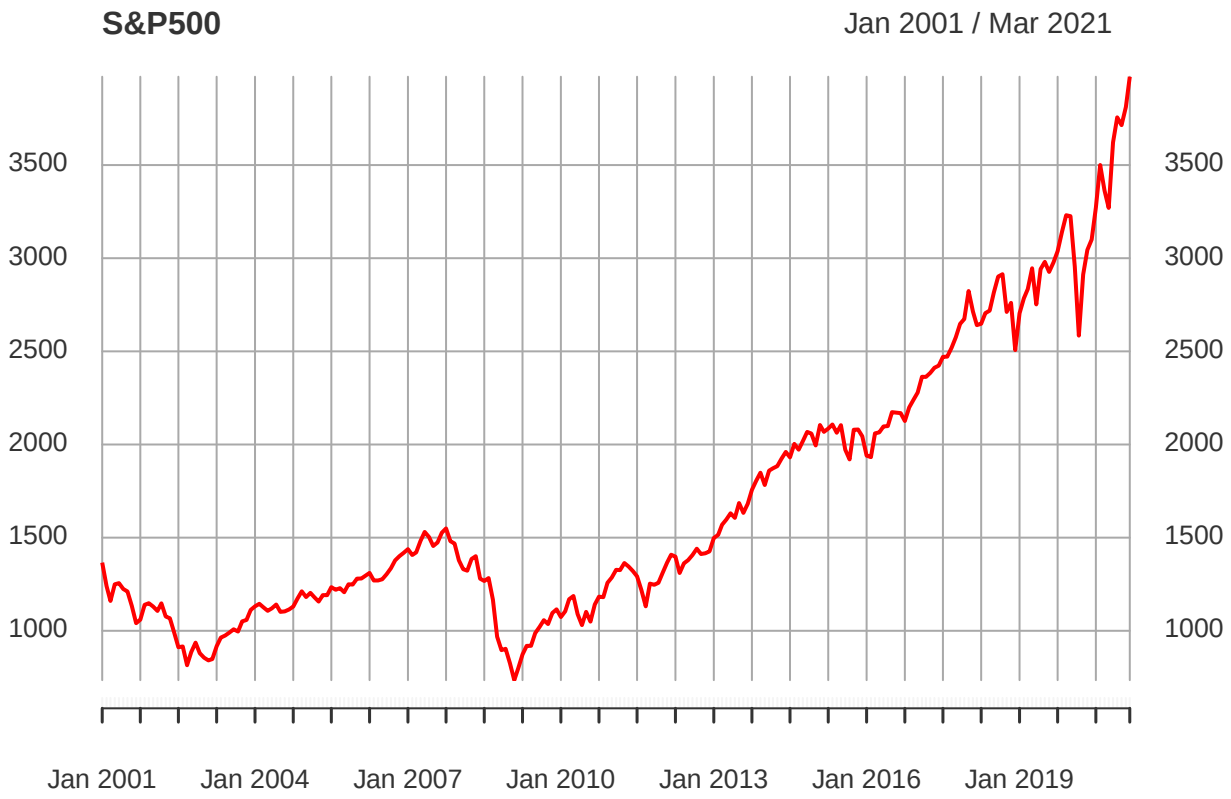
```
## Warning in to.period(x, "months", indexAt = indexAt, name = name, ...): missing
## values removed from data
```

```
adjFCHI_mo <- to.monthly(FCHI)$FCHI.Adjusted
```

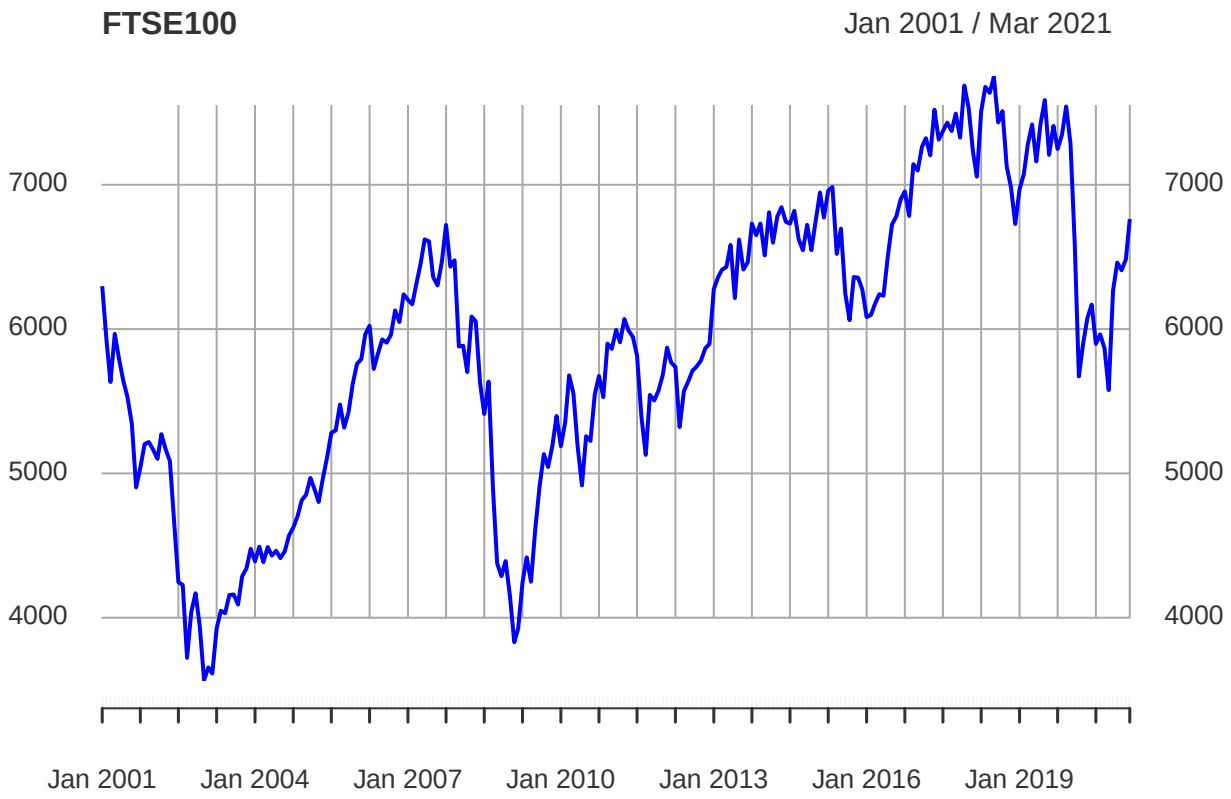
```
## Warning in to.period(x, "months", indexAt = indexAt, name = name, ...): missing
## values removed from data
```

Observing each indices:

```
plot(adjGSPC_mo, main="S&P500", col="Red")
```

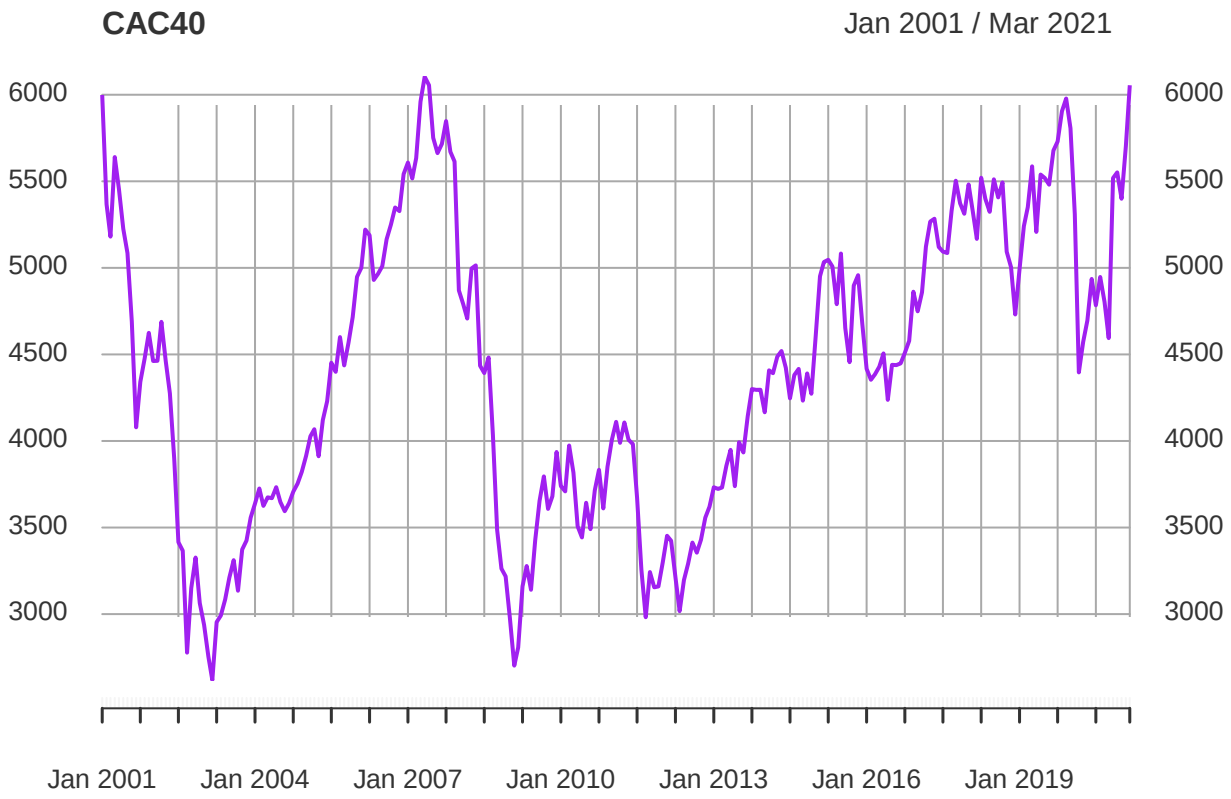



```
plot(adjFTSE_mo, main="FTSE100", col="Blue")
```




```
plot(adjGDAXI_mo, main="DAX30", col="Chocolate")
```


```
plot(adjFCHI_mo, main="CAC40", col="Purple")
```


CAC40
Jan 2001 / Mar 2021
6000
5500
5000
4500
4000
3500
3000
Jan 2001
Jan 2004
Jan 2007
Jan 2010
Jan 2013
Jan 2016
Jan 2019