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
```

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

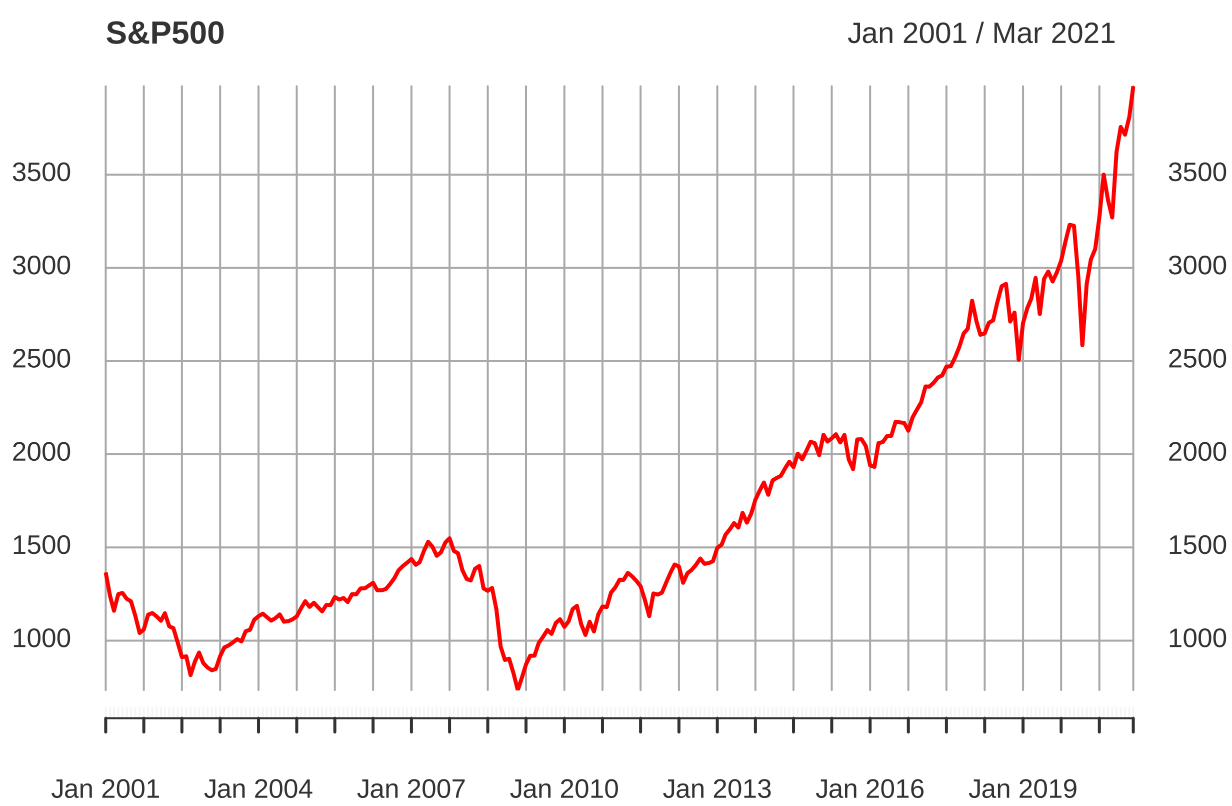
```
## Warning in to.period(x, "months", indexAt = indexAt, name = name, ...): missing
## values removed from data
```

```r
# Merge xts object
globalIndices <- merge.xts(adjGSPC_mo)
```

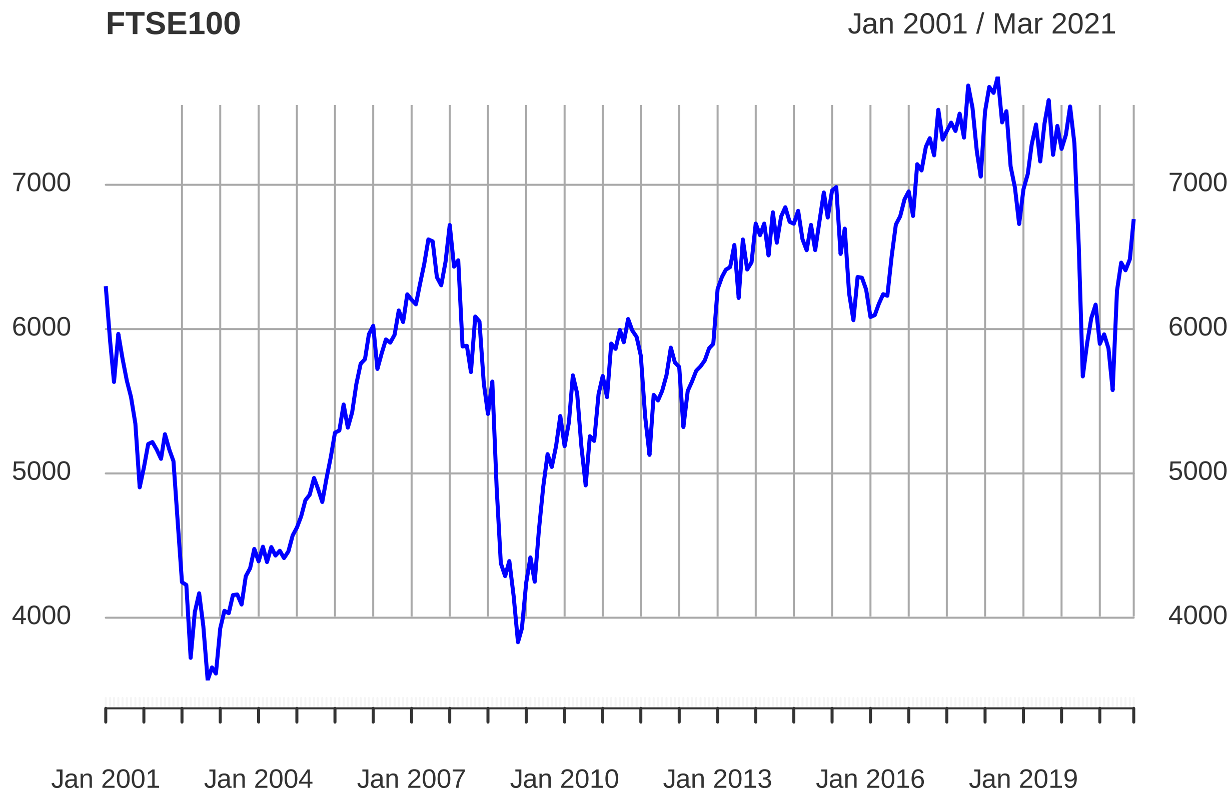
Data Observation:

Observing each indices:

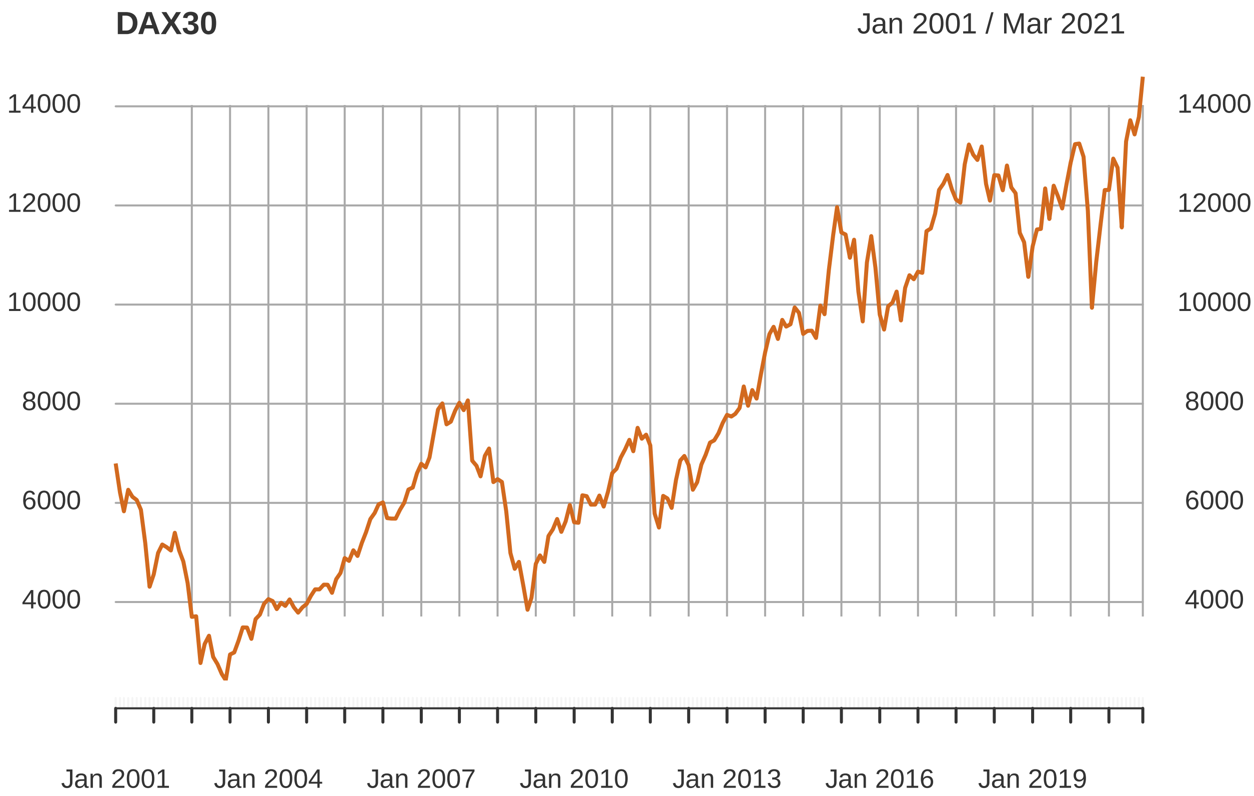
```r
plot(adjGSPC_mo, main="S&P500", col="Red")
```



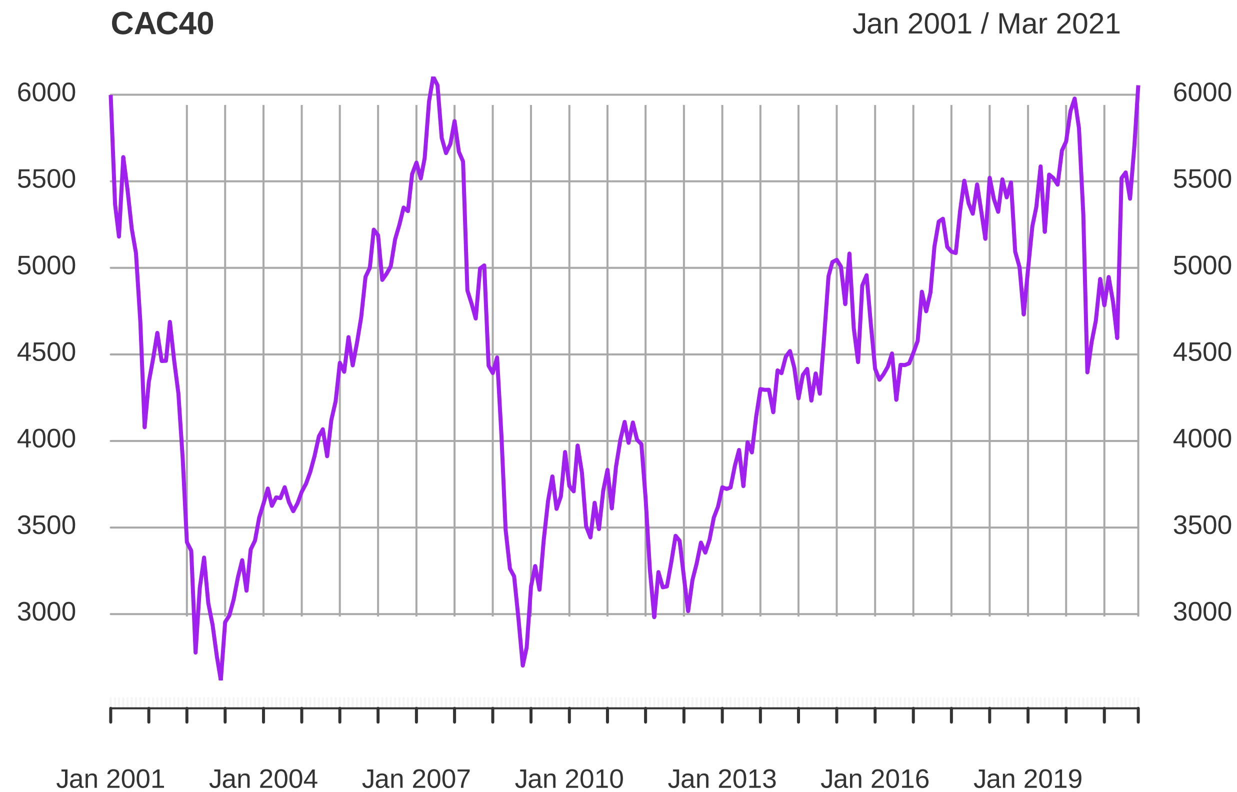

```r
plot(adjFTSE_mo, main="FTSE100", col="Blue")
```




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




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



Remarks
1. Seasonal Trend:
All display similar seasonal trend with up and down spikes in price months after months.
2. Cyclical Trend:
All are affected by business cycle with similar peaks and troughs. The S&P 500 has less prominent troughs compared to those of other indices.
3. Auto-correlation:
All behave similar in this regard where both rises for some times when they rise and vice versa.
4. Randomness:
Their prices are unpredictable via inspection.
5. Time:
All follows similar time trend.
6. Structural Break:
We can see several structural breaks conincided with the 2001 Tech Bust and the 2007 Great Financial Crisis.
7. Outliers:
All have multiple outliers.

In summary, we can observe clear trend and non-constant drift from all indices.

Testing for Stationarity for indices

Per methodology, we run Augmented Dickey Fuller Test for each indices. Recall that the null hypothesis for Dickey-Fuller Test is that a unit root is present in our autoregressive model, meaning the variable is a non-stationary variable.

S&P 500

```
library(aTSA)
```

```
##
## Attaching package: 'aTSA'

## The following object is masked from 'package:graphics':
##
##     identify
```

```
adf.test(adjGSPC_mo)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag  ADF p.value
## [1,]   0 3.47    0.99
## [2,]   1 3.47    0.99
## [3,]   2 4.06    0.99
## [4,]   3 3.83    0.99
## [5,]   4 3.83    0.99
## Type 2: with drift no trend
##      lag  ADF p.value
## [1,]   0 3.80    0.99
## [2,]   1 3.73    0.99
## [3,]   2 4.32    0.99
## [4,]   3 4.25    0.99
## [5,]   4 4.33    0.99
## Type 3: with drift and trend
##      lag  ADF p.value
## [1,]   0 2.71    0.99
## [2,]   1 2.95    0.99
## [3,]   2 3.46    0.99
## [4,]   3 3.35    0.99
## [5,]   4 3.45    0.99
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

We can observe $p - value = .99 > .05$. Thus, we fail to reject the null hypothesis. In other words, S&P 500 monthly adjusted closing price has a unit root and therefore, is a non-stationary variable.

FTSE 100

```
adf.test(adjFTSE_mo)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
```

```
##
## Type 1: no drift no trend
##      lag   ADF p.value
## [1,]   0 0.440   0.770
## [2,]   1 0.551   0.802
## [3,]   2 0.640   0.828
## [4,]   3 0.538   0.799
## [5,]   4 0.572   0.808
## Type 2: with drift no trend
##      lag  ADF p.value
## [1,]   0 1.84    0.99
## [2,]   1 1.87    0.99
## [3,]   2 1.92    0.99
## [4,]   3 1.86    0.99
## [5,]   4 1.79    0.99
## Type 3: with drift and trend
##      lag  ADF p.value
## [1,]   0 2.08    0.99
## [2,]   1 2.36    0.99
## [3,]   2 2.62    0.99
## [4,]   3 2.39    0.99
## [5,]   4 2.44    0.99
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

We can observe $p - value > .05$. Thus, we fail to reject the null hypothesis. In other words, FTSE100 monthly adjusted closing price has a unit root and therefore, is a non-stationary variable.

DAX 30

```
adf.test(adjGDAXI_mo)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag  ADF p.value
## [1,]   0 1.86   0.984
## [2,]   1 1.83   0.983
## [3,]   2 2.05   0.990
## [4,]   3 2.00   0.989
## [5,]   4 2.04   0.990
## Type 2: with drift no trend
##      lag  ADF p.value
## [1,]   0 2.16    0.99
## [2,]   1 2.01    0.99
## [3,]   2 2.19    0.99
## [4,]   3 2.23    0.99
## [5,]   4 2.23    0.99
## Type 3: with drift and trend
##      lag  ADF p.value
## [1,]   0 2.03    0.99
```

```
## [2,]   1 2.19    0.99
## [3,]   2 2.56    0.99
## [4,]   3 2.44    0.99
## [5,]   4 2.60    0.99
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

We can observe $p - value = .99 > .05$. Thus, we fail to reject the null hypothesis. In other words, DAX 30 monthly adjusted closing price has a unit root and therefore, is a non-stationary variable.

CAC 40

```
adf.test(adjFCHI_mo)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag   ADF p.value
## [1,]   0 0.413   0.763
## [2,]   1 0.611   0.820
## [3,]   2 0.673   0.837
## [4,]   3 0.493   0.786
## [5,]   4 0.563   0.806
## Type 2: with drift no trend
##      lag  ADF p.value
## [1,]   0 2.12    0.99
## [2,]   1 2.20    0.99
## [3,]   2 2.42    0.99
## [4,]   3 2.09    0.99
## [5,]   4 2.15    0.99
## Type 3: with drift and trend
##      lag  ADF p.value
## [1,]   0 1.61    0.99
## [2,]   1 1.87    0.99
## [3,]   2 2.08    0.99
## [4,]   3 1.66    0.99
## [5,]   4 1.78    0.99
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

We can observe $p - value > .05$. Thus, we fail to reject the null hypothesis. In other words, CAC 40 monthly adjusted closing price has a unit root and therefore, is a non-stationary variable.

Remarks

Through (A)DF tests, we can observe that the indices adjusted monthly closing prices are non-stationary variables.

Testing for stationarity for error term

Regression model

Shit's not working. FFS. Will need to run on residual instead of adjGSPC_mo

```
model1 <- lm(adjGSPC_mo ~ adjFTSE_mo + adjGDAXI_mo + adjFCHI_mo, data=globalIndices)
model1
```

```
##
## Call:
## lm(formula = adjGSPC_mo ~ adjFTSE_mo + adjGDAXI_mo + adjFCHI_mo,
##     data = globalIndices)
##
## Coefficients:
## (Intercept)   adjFTSE_mo  adjGDAXI_mo   adjFCHI_mo
##    928.0168      -0.3887       0.3024       0.1580
```

Checking for positive autocorrelation

We first run Durbin-Watson Test to check for positive autocorrelation.

```
library(car)
```

```
## Loading required package: carData
```

```
durbinWatsonTest(model1)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.9886016             0       0
##  Alternative hypothesis: rho != 0
```

We can observe $p - value < .05$ from the DW Test, implying autocorrelation of the residuals in this model. We will attempt to remedy by adding $AR(1)$.

Fitting $ARIMA(1,0,0)$ or $AR(1)$