MA415 Final Project Workfile

Lok Tin Kevin Chan

Changepoint Detection in Financial Timeseries

Introduction

Change points are abrupt variations in time series data. Such abrupt changes may represent transitions that occur between states. Detection of change points is useful in modelling and prediction of time series and is found in application areas such as medical condition monitoring, climate change detection, speech and image analysis, and human activity analysis.

Methodology

Changepoint detection is the name given to the problem of estimating the point(s) at which the statistical properties of a time-series observations change. In this project, I focus on the changes in mean and variance. The penalty function is chosen as SIC (Schwarz Information Criterion); and we employ the PELT (Pruned Exact Linear Time) searching algorithm for the change points proposed by Killick et al. (2012).

Changepoint Detections

```r
library(readxl)
library(tidyverse)
library(changepoint)
library(ggfortify)
library(imputeTS)
library(lubridate)
library(padr)
library(grid)
library(gridExtra)
library(plotly)
library(devtools)
library(stringr)
library(moments)
source("./Scripts/Clean data function.R")
source("./Scripts/summary statistics.R")

#Read data from in excel files(already removed bloomberg commands)
Weekdays <- read_excel("./misc/weekdays.xlsx")
News <- read_excel("./misc/news.xlsx")
Brent <- read_excel("./data/Brent oil price data.xlsx")
sp500 <- read_excel("./data/SP500 Data.xlsx")
gold <- read_excel("./data/Gold price data.xlsx")
bond <- read_excel("./data/tbill data.xlsx")
vix <- read_excel("./data/vix data.xlsx")
usdeur <- read_excel("./data/usdeur data.xlsx")

#Clean the data & calculate the log return
```

```r
Brent <- clean.function(Brent)
sp500 <- clean.function(sp500)
gold <- clean.function(gold)
bond <- clean.function(bond)
vix <- clean.function(vix)
usdeur <- clean.function(usdeur)
```

```r
#Using Changepoint package to detect change opt amount point dates using PELT method

Brent.Changepoint <- (cpt.meanvar(Brent$log_return,
                                  method = "PELT",
                                  test.stat = "Normal"))


SP500.Changepoint <- cpt.meanvar(sp500$log_return,
                                 method = "PELT",
                                 test.stat = "Normal")


gold.Changepoint <- cpt.meanvar(gold$log_return,
                                method = "PELT",
                                test.stat = "Normal")


vix.Changepoint <- cpt.meanvar(vix$log_return,
                               method = "PELT",
                               test.stat = "Normal")


usdeur.Changepoint <- cpt.meanvar(usdeur$log_return,
                                  method = "PELT",
                                  test.stat = "Normal")

B.Date <-Brent$Date[cpts(Brent.Changepoint)]
sp.Date <- sp500$Date[cpts(SP500.Changepoint)]
g.Date <- gold$Date[cpts(gold.Changepoint)]
u.Date <- usdeur$Date[cpts(usdeur.Changepoint)]
v.Date <- vix$Date[cpts(vix.Changepoint)]
```

```r
#Aggregate the data into one file and tidy the data

Agg_Price <- Brent[1:2] %>%
                rename(Brent = Price) %>%
                right_join(sp500[1:2]) %>%
                rename(SP500 = Price) %>%
                right_join(gold[1:2]) %>%
                rename(Gold = Price) %>%
                right_join(vix[1:2]) %>%
                rename(VIX = Price) %>%
                right_join(usdeur[1:2]) %>%
                rename(USD_EUR_Exchange = Price)
```

```
## Joining, by = "Date"
## Joining, by = "Date"
## Joining, by = "Date"
## Joining, by = "Date"
```

```
Agg.tidy <- Agg_Price %>%
                gather(key = "Index", value = "Price", -Date)
```

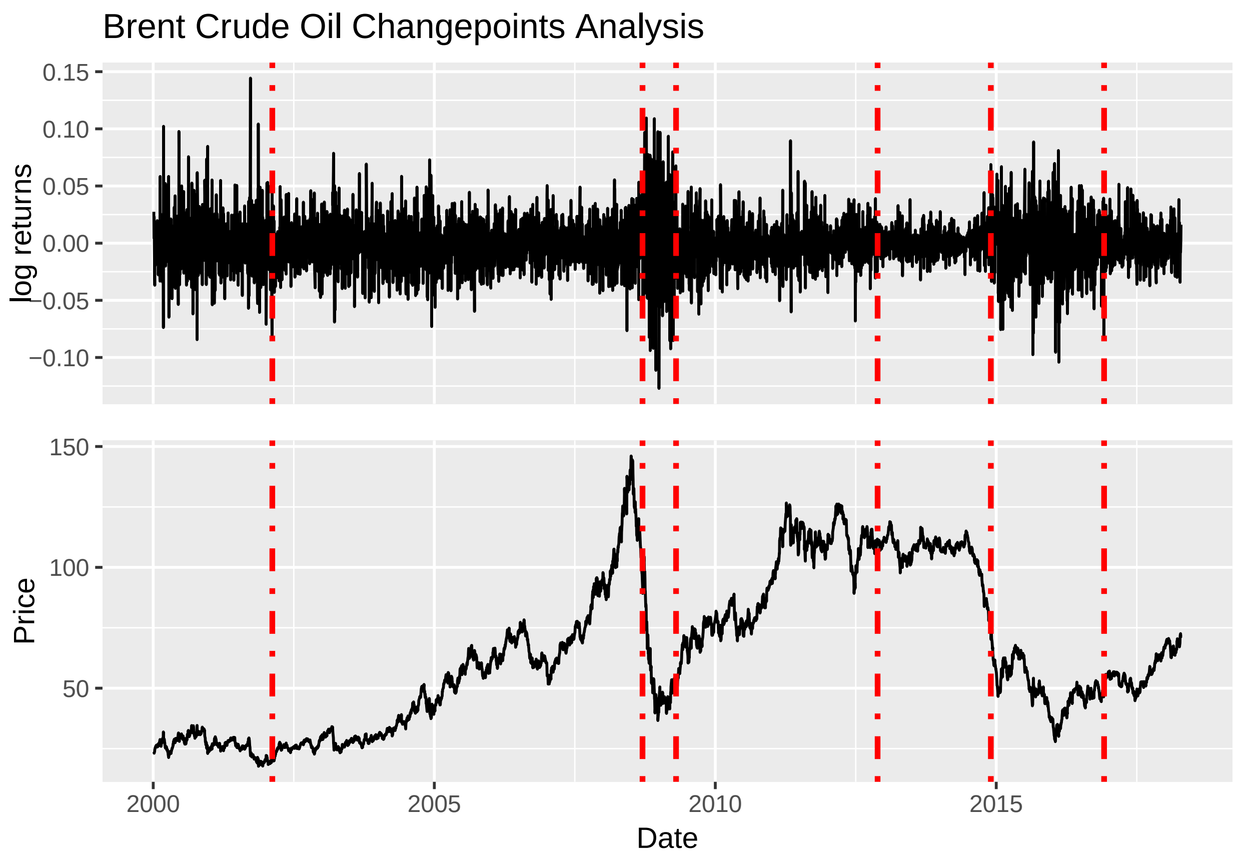
Brent Crude Oil Changepoint Detection

```
#Create Brent Changepoint Detection Plot

Brent.Plot1 <- ggplot(Brent, aes(as.Date(Date),log_return))+
  geom_line()+
  geom_vline(xintercept = as.Date(B.Date) , color = "red", linetype = "dotdash", size = 1)+
  ylab("log returns")+
  ggtitle("Brent Crude Oil Changepoints Analysis")+
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())

Brent.Plot2 <- ggplot(Brent, aes(Date, Price))+  geom_line()+
  geom_vline(xintercept = as.numeric(B.Date), color = "red", linetype = "dotdash", size = 1)+
  xlab("Date")+
  ylab("Price")

grid.newpage()
  grid.draw(rbind(ggplotGrob(Brent.Plot1),ggplotGrob(Brent.Plot2), size = "last"))
```



```
as.tibble(News[which(as.Date(News$Date) %in% as.Date(B.Date)),])
```

```
## # A tibble: 6 x 2
##   Date                News
##   <dttm>              <chr>
## 1 2002-02-13 00:00:00 JetBlues files plans for 125 million offering
## 2 2008-09-15 00:00:00 Financial Crisis 08
## 3 2009-04-21 00:00:00 Smart Money Moves for Young Investors
## 4 2012-11-21 00:00:00 A preliminary gauge of China's manufacturing activi~
## 5 2014-11-27 00:00:00 Oil Seen in New Era as OPEC Won’t Yield to U.S. Sha~
## 6 2016-12-02 00:00:00 America’s Job Market Gets Tighter, as Wage Drop See~
```

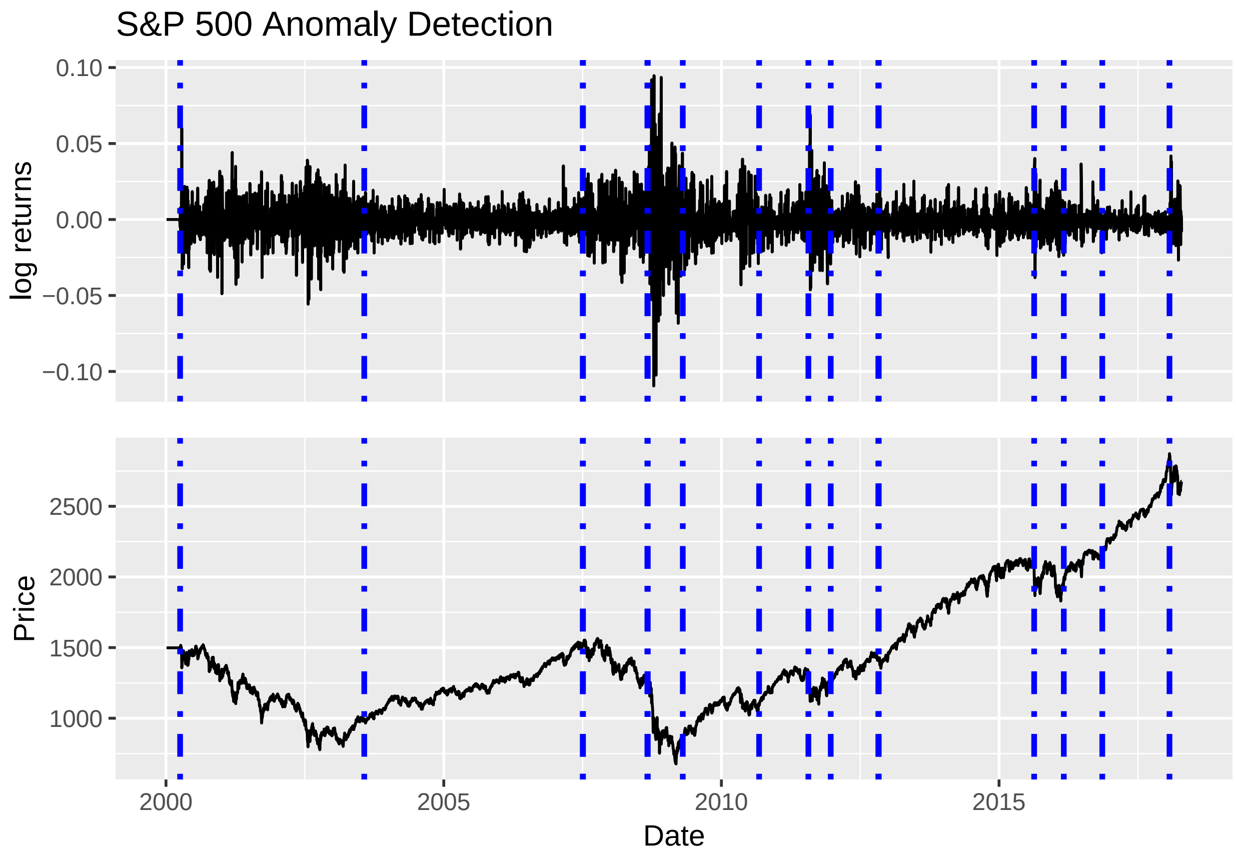
S&P 500 Changepoint detection

```
#Create S&P 500 changepoint detection

sp500.Plot1 <- ggplot(sp500, aes(Date,log_return))+
  geom_line()+
  geom_vline(xintercept = as.numeric(sp.Date), color = "Blue", linetype = "dotdash", size = 1)+
  ylab("log returns")+
  ggtitle("S&P 500 Anomaly Detection")+
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```

```
sp500.Plot2 <- ggplot(sp500, aes(Date, Price))+  geom_line()+
  geom_vline(xintercept = as.numeric(sp.Date), color = "Blue", linetype = "dotdash", size = 1)+
  xlab("Date")+
  ylab("Price")


grid.newpage()
  grid.draw(rbind(ggplotGrob(sp500.Plot1),ggplotGrob(sp500.Plot2), size = "last"))
```



```
as.tibble(News[which(as.Date(News$Date) %in% as.Date(sp.Date)),])
```

```
## # A tibble: 16 x 2
##    Date                News
##    <dttm>              <chr>
##  1 2003-07-28 00:00:00 The breakdown of Microsoft Corp. antitrust suit se~
##  2 2007-07-04 00:00:00 Fed Lowers rate to 4.75%
##  3 2007-07-06 00:00:00 In London's Financial World, Carbon Trading is the~
##  4 2007-07-06 00:00:00 Subprime Mortgage Crsis
##  5 2008-09-01 00:00:00 Financial Crisis 08
##  6 2008-09-04 00:00:00 Financial Crisis 08
##  7 2009-04-22 00:00:00 IMF puts Bank Losses from Global Financial Crisis
##  8 2010-09-06 00:00:00 NBER declares recession ended in June 2009
##  9 2011-07-27 00:00:00  Stocks retreated deep into negative territory on ~
## 10 2011-12-21 00:00:00 Ex-Fannie CEO Mudd leaves Fortress after SEC charg~
## 11 2012-10-29 00:00:00 U.S. markets to remain closed Due to Hurricane San~
## 12 2012-11-01 00:00:00 U.S. markets to remain closed Tuesday Hurricane Sa~
## 13 2015-08-20 00:00:00 Dow plummets 358 points to close below 17,000 for ~
## 14 2015-08-20 00:00:00 Stocks Fall Most in 4 Years as China Dread Sinks G~
## 15 2016-11-10 00:00:00 Dow Record Comes in Market Radically Polarized by ~
## 16 2018-01-26 00:00:00 Dow pops 224 points, stocks notch record close on ~
```

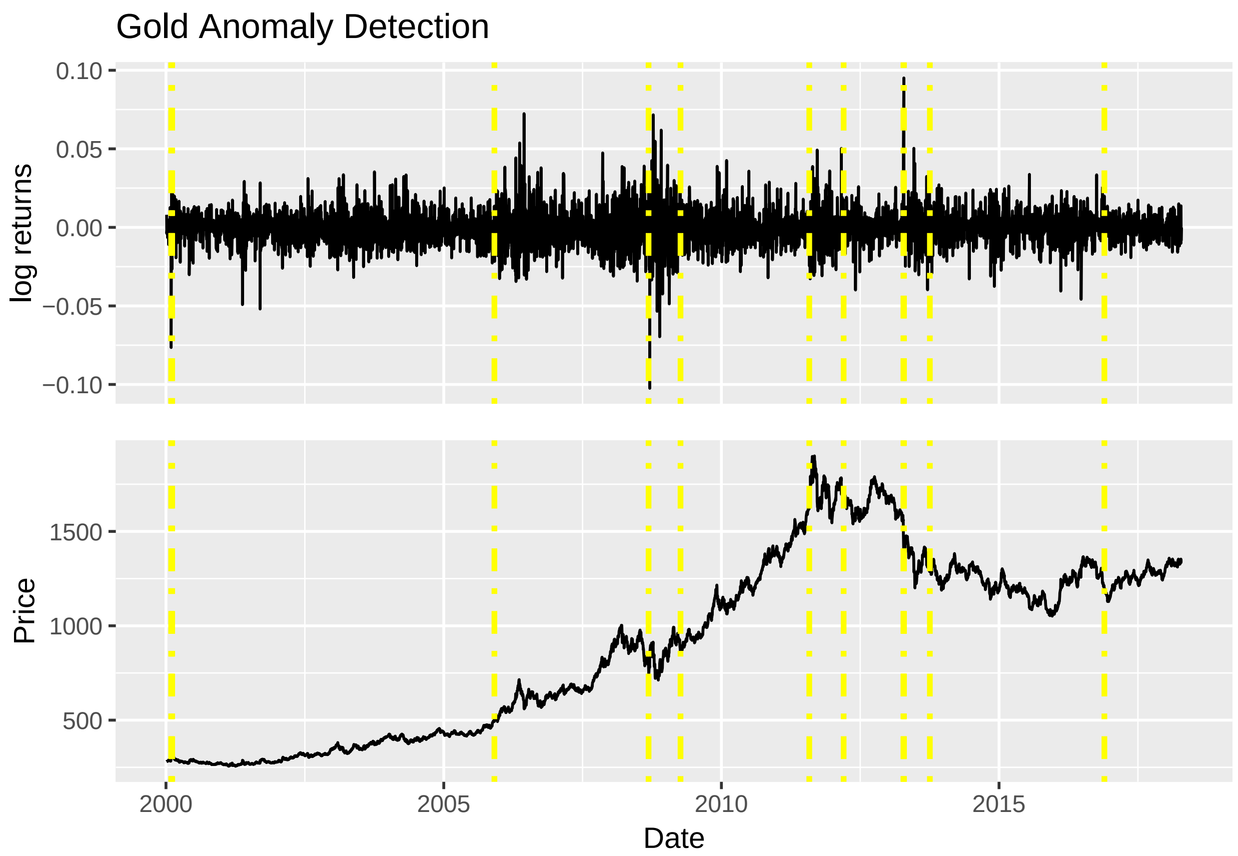
Gold Changepoint detectio

```r
#Gold changepoint detection plot

gold.Plot1 <- ggplot(gold, aes(Date,log_return))+
  geom_line()+
  geom_vline(xintercept = as.numeric(g.Date), color = "Yellow", linetype = "dotdash", size = 1)+
  ylab("log returns")+
  ggtitle("Gold Anomaly Detection")+
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())

gold.Plot2 <- ggplot(gold, aes(Date, Price))+  geom_line()+
  geom_vline(xintercept = as.numeric(g.Date), color = "Yellow", linetype = "dotdash", size = 1)+
  xlab("Date")+
  ylab("Price")


grid.newpage()
  grid.draw(rbind(ggplotGrob(gold.Plot1),ggplotGrob(gold.Plot2), size = "last"))
```

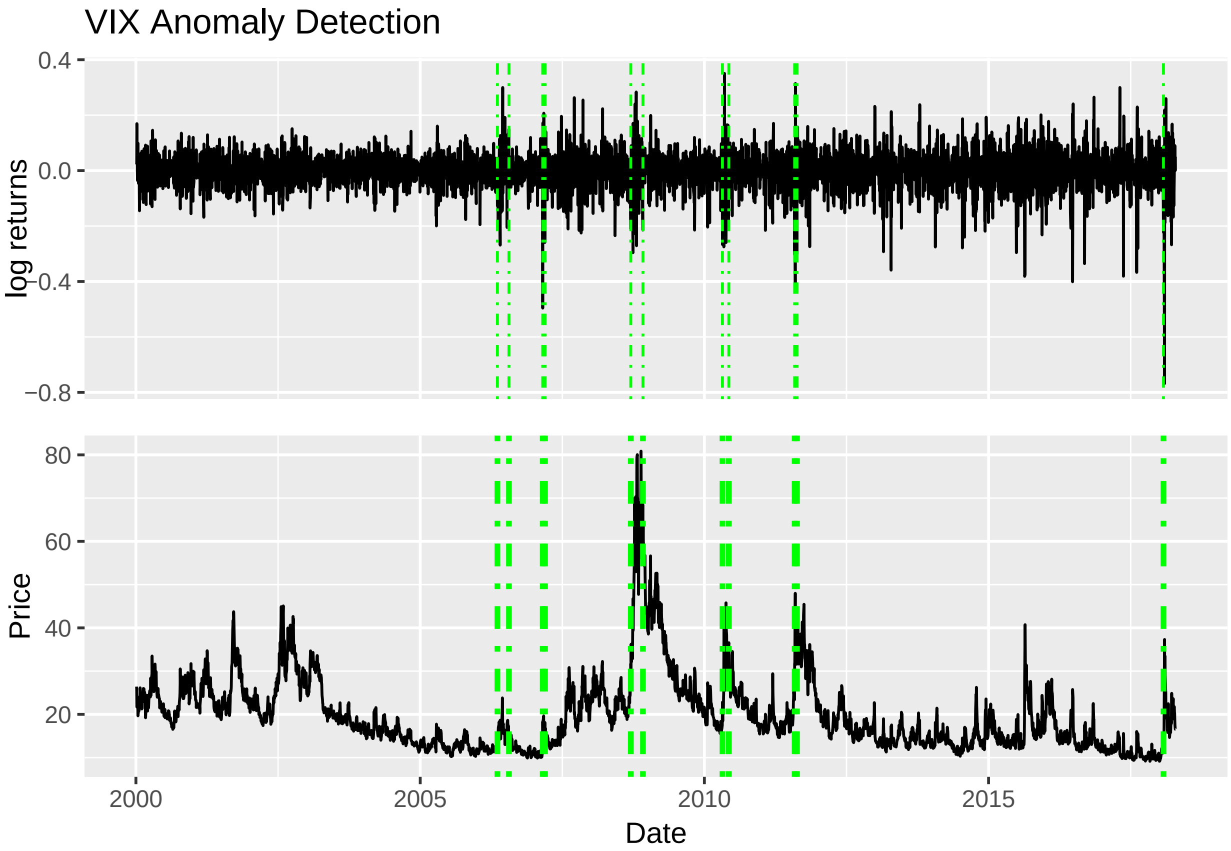


```
#Vix changepoint detection plot

vix.Plot1 <- ggplot(vix, aes(Date,log_return))+
  geom_line()+
  geom_vline(xintercept = as.numeric(v.Date), color = "Green", linetype = "dotdash")+
  ylab("log returns")+
  ggtitle("VIX Anomaly Detection")+
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())

vix.Plot2 <- ggplot(vix, aes(Date, Price))+  geom_line()+
  geom_vline(xintercept = as.numeric(v.Date), color = "Green", linetype = "dotdash", size = 1)+
  xlab("Date")+
  ylab("Price")


grid.newpage()
  grid.draw(rbind(ggplotGrob(vix.Plot1),ggplotGrob(vix.Plot2), size = "last"))
```

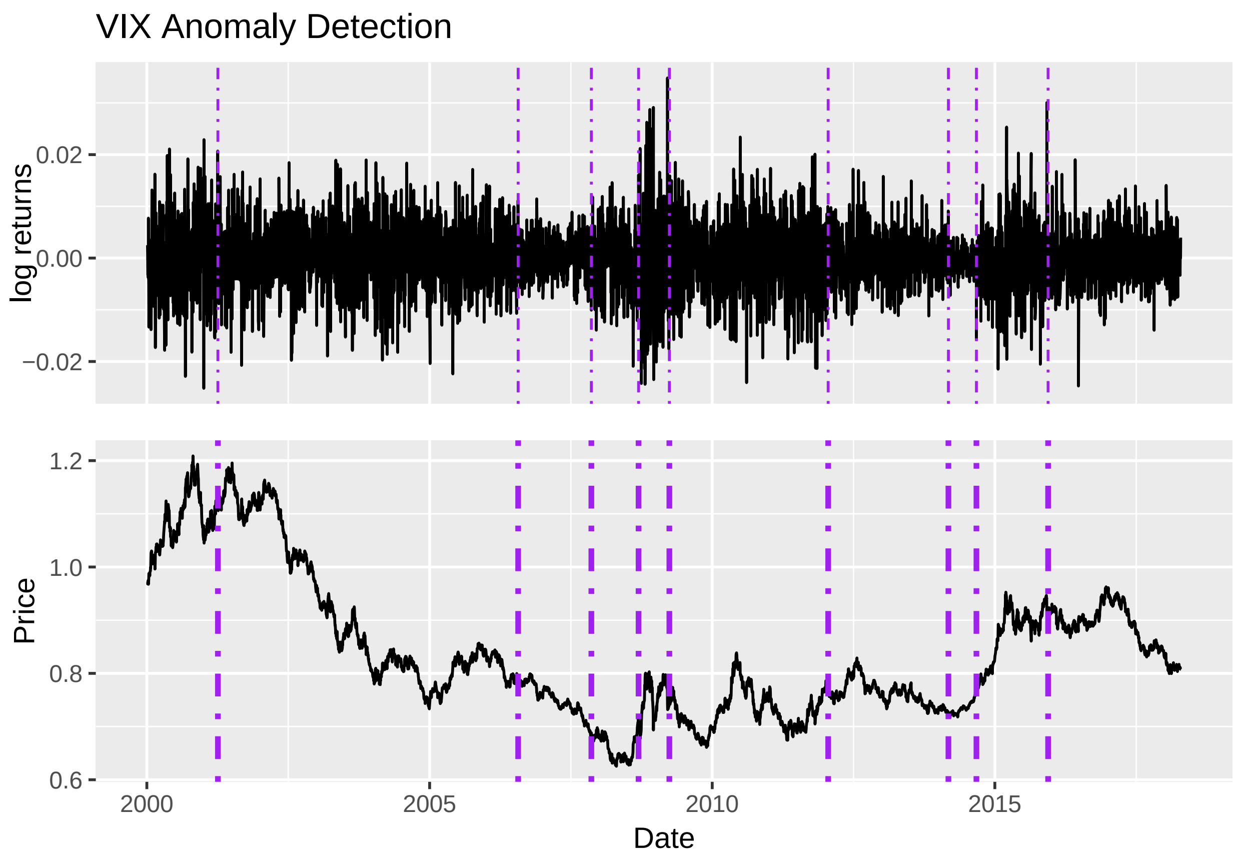


```
#USD EUR Changepoint detection plot

usdeur.Plot1 <- ggplot(usdeur, aes(Date,log_return))+
  geom_line()+
  geom_vline(xintercept = as.numeric(u.Date), color = "purple", linetype = "dotdash")+
  ylab("log returns")+
  ggtitle("VIX Anomaly Detection")+
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())

usdeur.Plot2 <- ggplot(usdeur, aes(Date, Price))+  geom_line()+
  geom_vline(xintercept = as.numeric(u.Date), color = "purple", linetype = "dotdash", size = 1)+
  xlab("Date")+
  ylab("Price")


grid.newpage()
  grid.draw(rbind(ggplotGrob(usdeur.Plot1),ggplotGrob(usdeur.Plot2), size = "last"))
```



Summary Statistics

```
#Create the summary statistics (mean, var, skewness, kewtosis)


Brent.Ss <-sum.stat(Brent$log_return)
sp500.Ss <- sum.stat(sp500$log_return)
gold.ss <- sum.stat(gold$log_return)
vix.ss <- sum.stat(vix$log_return)
usdeur.ss <- sum.stat(usdeur$log_return)


summary_statistics <-  cbind(Brent.Ss, sp500.Ss, gold.ss,vix.ss,usdeur.ss)

colnames(summary_statistics) <- c("Brent","S&P 500","Gold","VIX","USD EUR")
summary_statistics
```

```
##                      Brent       S&P 500          Gold           VIX
## mean          -0.0002280822 -0.0001234565 -0.0003322091  0.0001040445
## standard.dev   0.0217333462  0.0117760712  0.0110633981  0.0668356623
## skewness       0.1461533153  0.1944713406  0.2249452202 -0.9330738027
## kurtosis       5.9671290597 12.3504584312  9.1585885351 10.7406895414
##                    USD EUR
## mean          3.920053e-05
## standard.dev  6.285766e-03
## skewness     3.840434e-02
## kurtosis     4.479254e+00
```

Correlation Table

```
# Create the correlation table

    tmp<- cbind(Brent$log_return,sp500$log_return,gold$log_return,vix$log_return,usdeur$log_return)
    colnames(tmp) <- c("Brent","S&P 500","Gold","VIX","USD EUR")
    corr.table <- round(cor(tmp),2)
    corr.table
```

```
##         Brent S&P 500  Gold   VIX USD EUR
## Brent    1.00    0.21  0.22 -0.17   -0.17
## S&P 500  0.21    1.00 -0.01 -0.73   -0.11
## Gold     0.22   -0.01  1.00 -0.01   -0.39
## VIX     -0.17   -0.73 -0.01  1.00    0.08
## USD EUR -0.17   -0.11 -0.39  0.08    1.00
```

Writing clean and tidy data to csv for Shiny App

```
#Writing Result tables to csv files for Shiny Application

# write.csv(Brent, file = "Brent.csv")
# write.csv(sp500, file = "sp500.csv")
# write.csv(gold, file = "gold.csv")
# write.csv(vix, file = "vix.csv")
# write.csv(bond, file = "bond.csv")
# write.csv(usdeur, file = "usdeur.csv")
# write.csv(Agg_Price, file = "Agg_Price.csv")
# write.csv(Agg.tidy, file = "Agg_tidy.csv")
# write.csv(summary_statistics, file = "summary.csv")
# write.csv(corr.table, file = "corr_table.csv")
```