VAR_Stocks_Model

kkitonga and ewayagi

9/10/2022

1.Install packages:ONLY if you have not installed

```
#install.packages('vars')
#install.packages('tseries')
#install.packages('psych')
```

2.Load libraries

```
library(vars)                    #VAR model
library(tseries)                 #ADF test
library(psych)                   #Neat Descriptives
```

3.Load data

```
#1.Loading dataset
data(EuStockMarkets)

#2.Creating dataframe:Indexing Last200 observations
df = na.omit(data.frame
             (ts(EuStockMarkets[1600:1800,1:4])))
```

4.Creating dataframe based on returns

```
#1.Get returns of stock market prices
DAX_re <- diff(ts(df$DAX))
SMI_re <- diff(ts(df$SMI))
CAC_re <- diff(ts(df$CAC))
FTSE_re <- diff(ts(df$FTSE))

#2.Bind them into a dataframe
df_re <-cbind(DAX_re,SMI_re,CAC_re,FTSE_re)

#3.Renaming columns in df_re
colnames(df_re) <- c('DAX','SMI','CAC','FTSE')
```

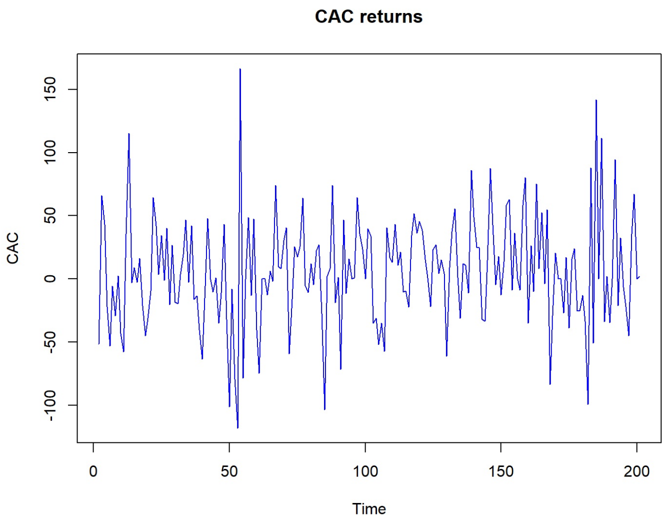
5.Exploratory data annalysis

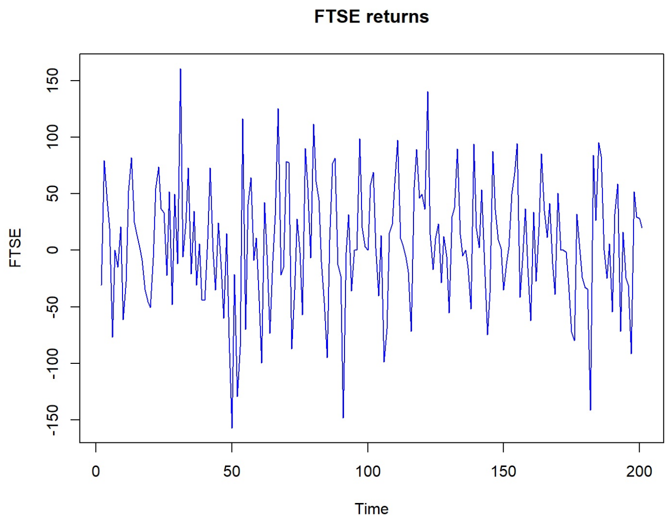
```
#i.Create descriptives function
descrip <-function(df){
                   p1<-print('first five observations')
                   first<-head(df)
                   p2<-print('last five observations')
                   tail<-tail(df)
                   p3<-print('five point summary')
                   five<-describe(df,fast=TRUE)
                   mylist <- list(p1,first,p2,
                                  tail,p3,five)
                   return(mylist)
}

#ii.run function on dataframe
descrip(df_re)
```

6.Visualization

```
#i.Plots:DAX,SMI returns

par(mfrow=c(1,2))
plot(DAX_re,type='l',col="blue",
     ylab='DAX',
     main='DAX returns ')

plot(SMI_re,type='l',col="blue",
     ylab='SMI',
     main='SMI returns')
```


```
par(mfrow=c(1,1))

#i.plots:CAC,FTSE returns
par(mfrow=c(1,2))
plot(CAC_re,type='l',col="blue",
     ylab='CAC',
     main='CAC returns')

plot(FTSE_re,type='l',col="blue",
     ylab='FTSE',
     main='FTSE returns')
```



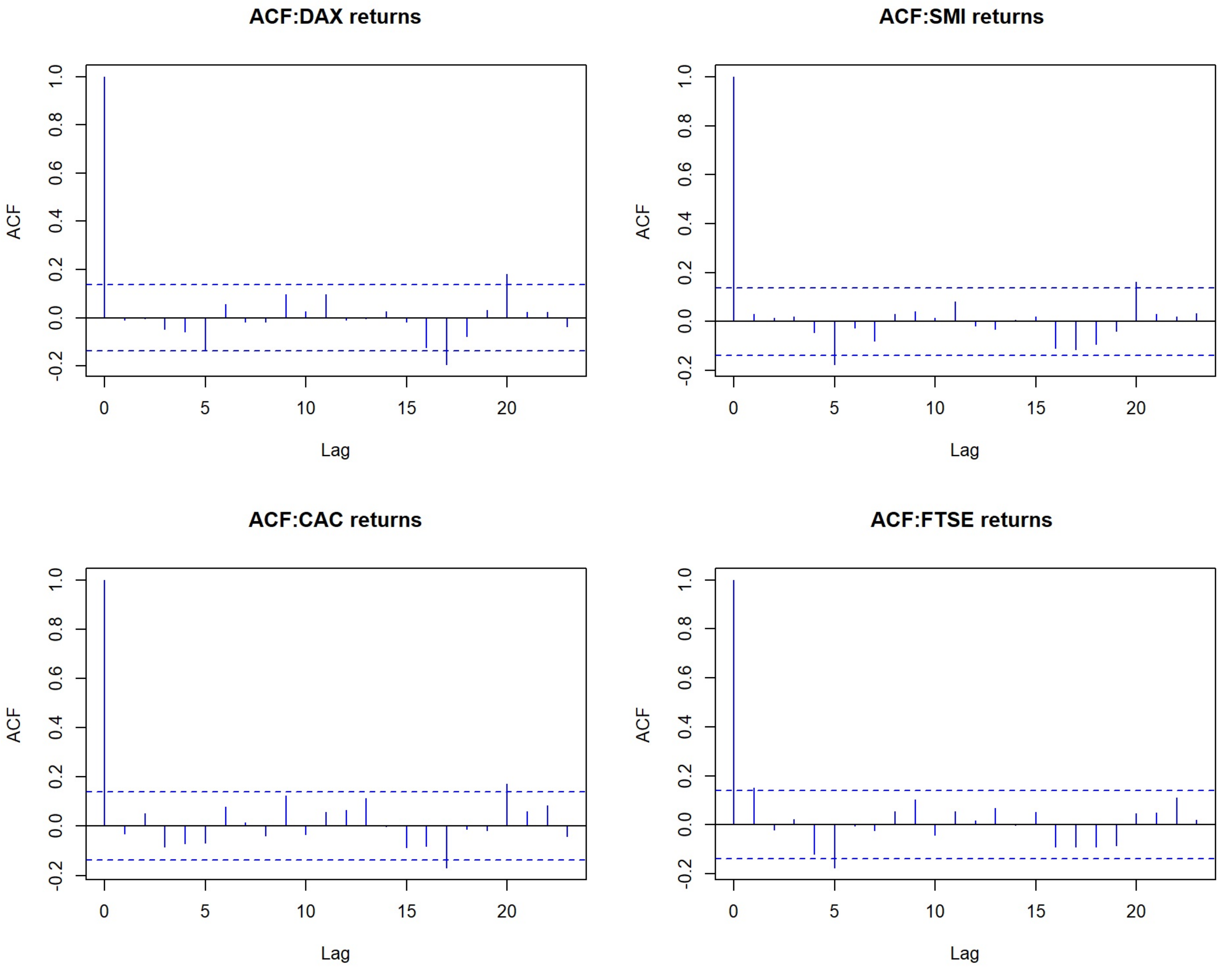



```
par(mfrow=c(1,1))
```

7.Stationarity

i.ACF plots

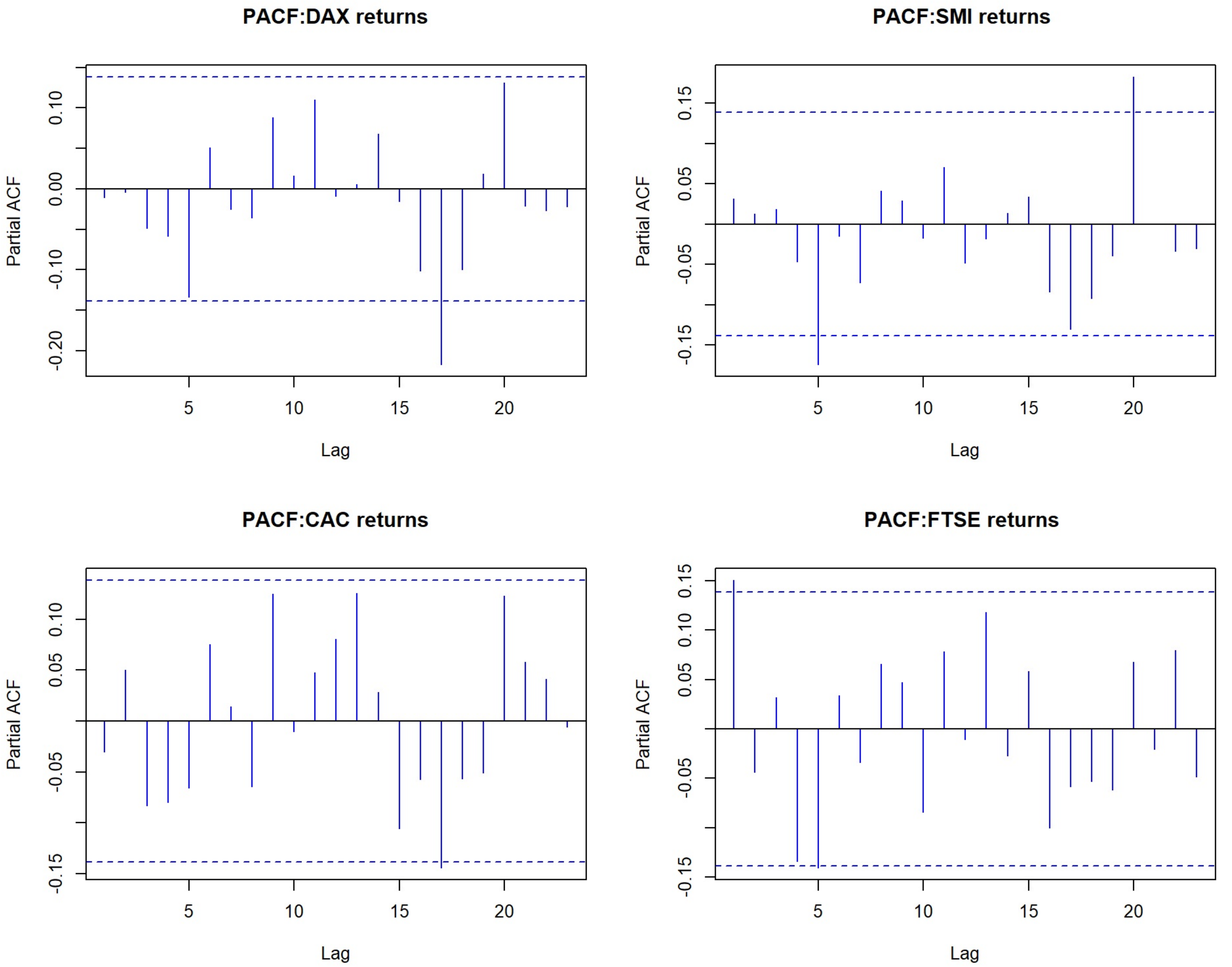
```
#i.Visual inspection :ACF PLOTS
par(mfrow=c(2,2))
acf(DAX_re,main='ACF:DAX returns',col='blue')
acf(SMI_re,main='ACF:SMI returns ',col='blue')
acf(CAC_re,main='ACF:CAC returns',col='blue')
acf(FTSE_re,main='ACF:FTSE returns ',col='blue')
```




```
par(mfrow=c(1,1))
```

ii.PACF plots

```
par(mfrow=c(2,2))
pacf(DAX_re,main='PACF:DAX returns',col='blue')
pacf(SMI_re,main='PACF:SMI returns ',col='blue')
pacf(CAC_re,main='PACF:CAC returns',col='blue')
pacf(FTSE_re,main='PACF:FTSE returns ',col='blue')
```




```
par(mfrow=c(2,2))
```

iii.Statistical test

```
#ho:time series is non-stationary
#h1:time series is stationary

adf.test(SMI_re,k=10)                   #stationary
```

```
## Warning in adf.test(SMI_re, k = 10): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  SMI_re
## Dickey-Fuller = -4.1854, Lag order = 10, p-value = 0.01
## alternative hypothesis: stationary
```

```
adf.test(DAX_re,k=10)                   #stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  DAX_re
## Dickey-Fuller = -3.7667, Lag order = 10, p-value = 0.02188
## alternative hypothesis: stationary
```

```
adf.test(SMI_re,k=10)                   #stationary
```

```
## Warning in adf.test(SMI_re, k = 10): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  SMI_re
## Dickey-Fuller = -4.1854, Lag order = 10, p-value = 0.01
## alternative hypothesis: stationary
```

```
adf.test(DAX_re,k=10)                   #stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  DAX_re
## Dickey-Fuller = -3.7667, Lag order = 10, p-value = 0.02188
## alternative hypothesis: stationary
```

8.Lag selection

```
optimal <-VARselect(df_re, type="const")
optimal$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      1      1      1      1
```

9.VAR Model

```
#i.VAR model
model <-VAR(df_re, p = 1, type = c("const"),
    season = NULL, exogen = NULL, lag.max = NULL,
    ic = c("AIC"))
#ii.VAR model results
summary(model)
```

10:DIAGNOSTICS

i.Autocorrelation

```
autocorr <- serial.test(model,lags.pt = 16,
                      type = c("PT.asymptotic"))
autocorr
```

ii.Heteroskedasticity

```
#ho:presence of heteroskedasticity
#h1:heteroskedasticity

hetesk <- arch.test(model,lags.multi=5,
                    multivariate.only=TRUE)
hetesk
```

```
##
##  ARCH (multivariate)
##
## data:  Residuals of VAR object model
## Chi-squared = 602.74, df = 500, p-value = 0.001069
```

iii.normality

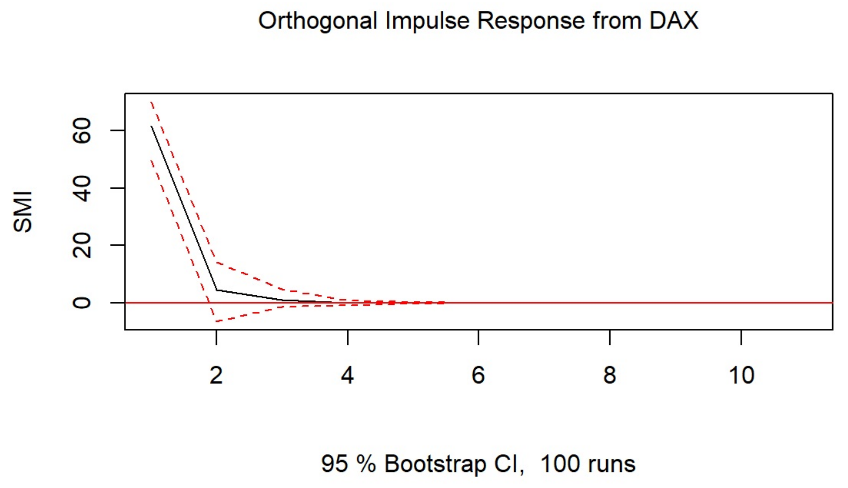
```
#ho:residuals do not follow normal distribution
#h1:residual follow normal distribution

normality <- normality.test(model,
                            multivariate.only=TRUE)
normality
```

11.Granger causality

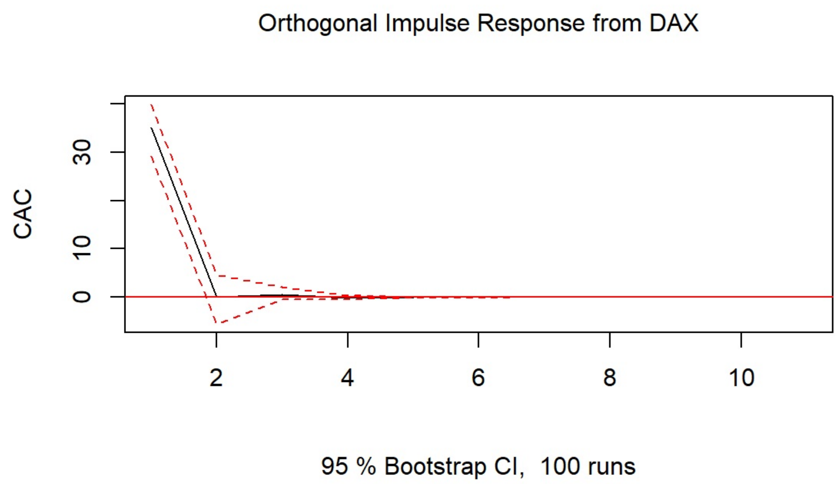
```
#verdict:p-value <0.05 reject null
#Conclude:There is instantaneous causality


#i.If DAX granger causes SMI,CAC,FTSE
DAX_granger <- causality(model,cause='DAX')
DAX_granger

#ii.If SMI granger causes DAX,CAC,FTSE
#verdict:p-value <0.05 reject null
DAX_granger <- causality(model,cause='SMI')
DAX_granger

#iii.If CAC granger causes SMI,FTSE,DAX
#verdict:p-value <0.05 reject null
DAX_granger <- causality(model,cause='CAC')
DAX_granger

#iv.If FTSE granger causes SMI,CAC,DAX
#verdict:p-value <0.05 reject null
DAX_granger <- causality(model,cause='FTSE')
DAX_granger
```

12.IMPULSE RESPONSE FUNCTIONS:Case of DAX variable

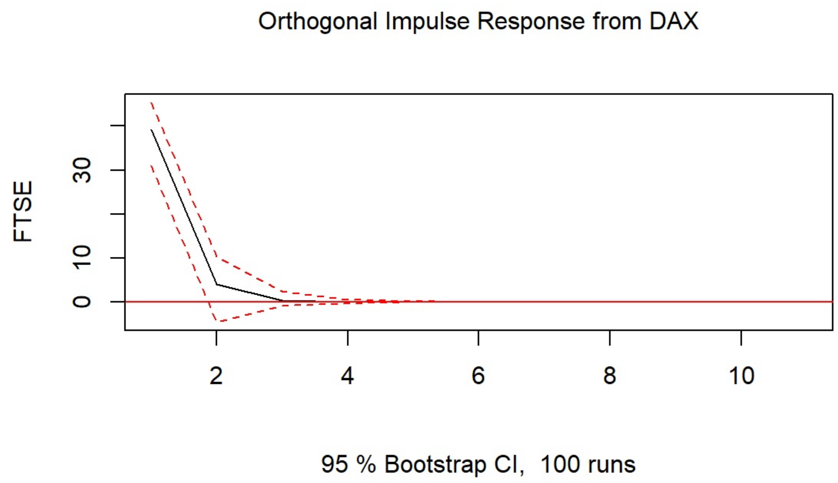
```
#i.IRF:Impact of a schock to DAX on other variables
DAX_SMI <- irf(model,impulse='DAX',response='SMI',n.ahead=10)
DAX_CAC <- irf(model,impulse='DAX',response='CAC',n.ahead=10)
DAX_FTSE <- irf(model,impulse='DAX',response='FTSE',n.ahead=10)

#ii.Plots of irf
plot(DAX_SMI)
```




```
plot(DAX_CAC)
```

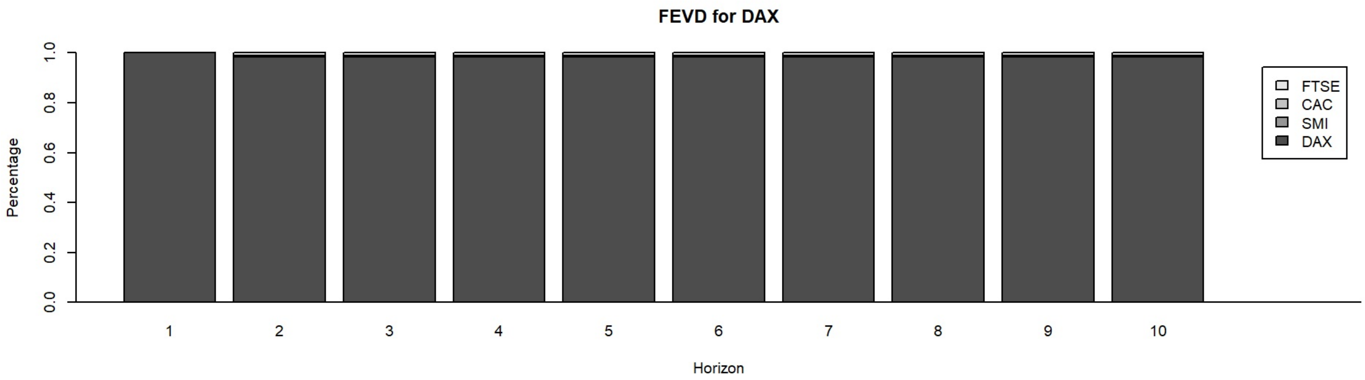


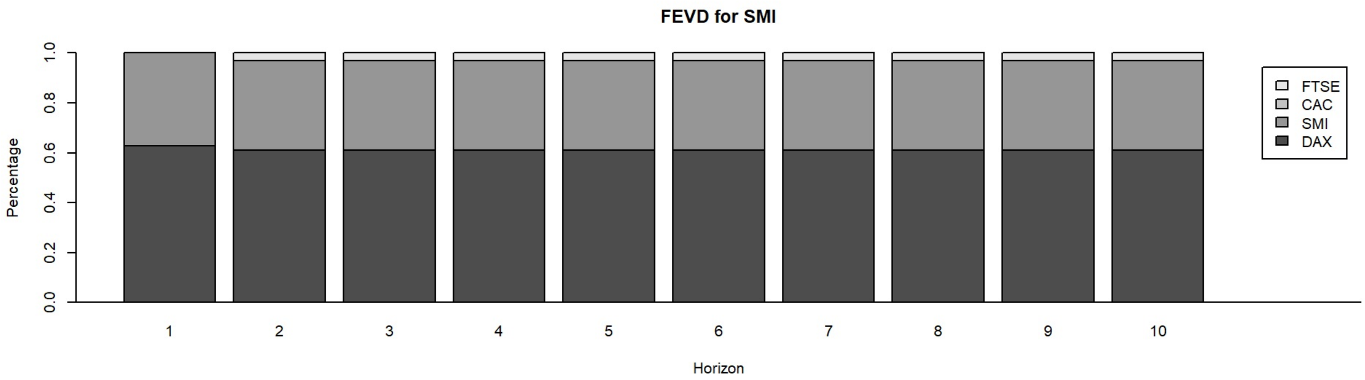

```
plot(DAX_FTSE)
```

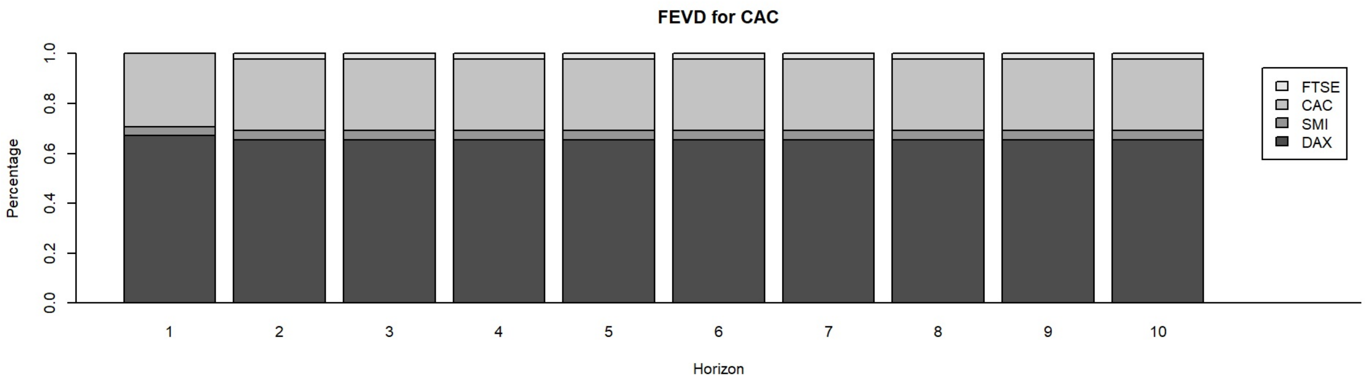


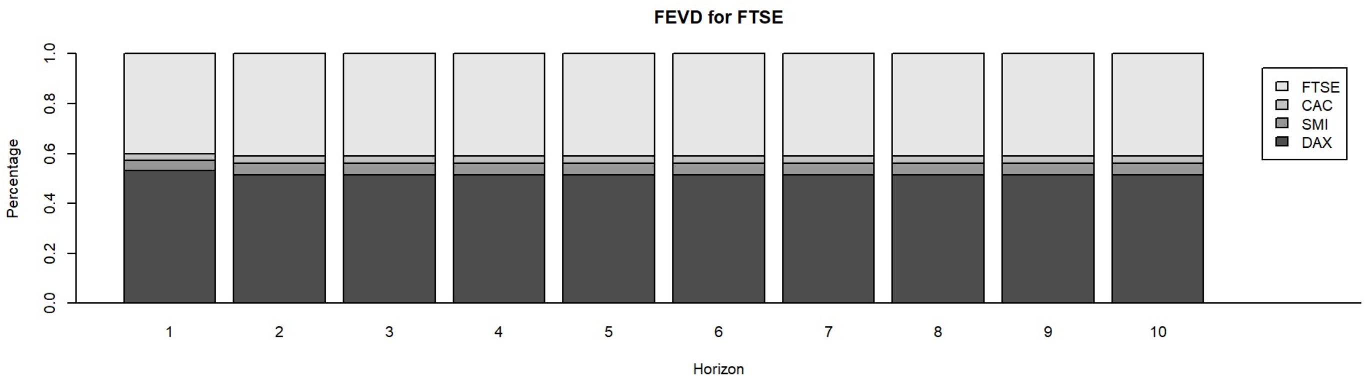
13.VARIANCE DECOMPOSITION

```
var_fevd <- fevd(model,n.ahead=10)
plot(var_fevd)
```



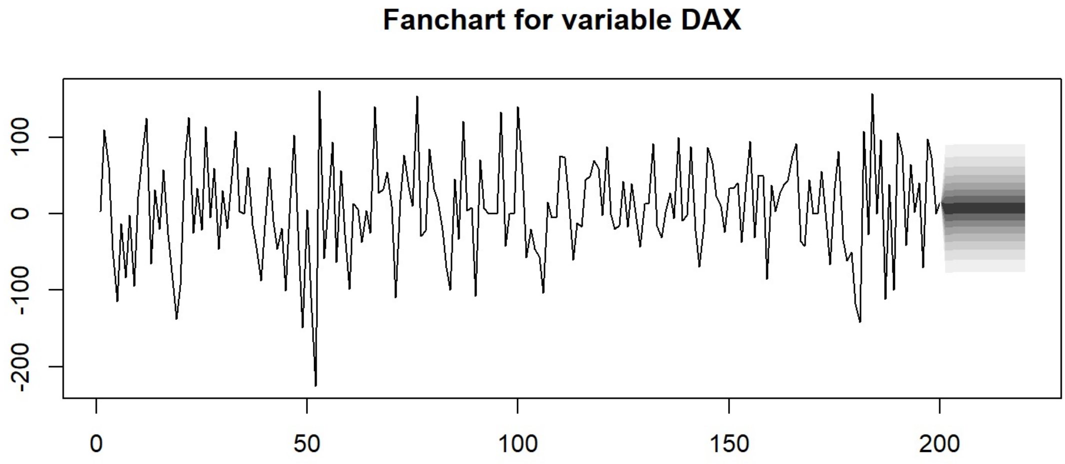




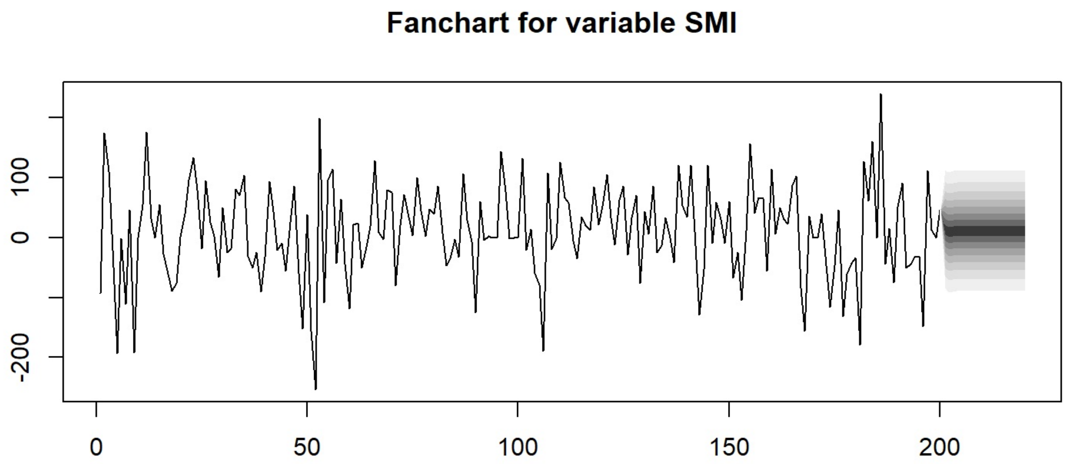


14.FORECASTING

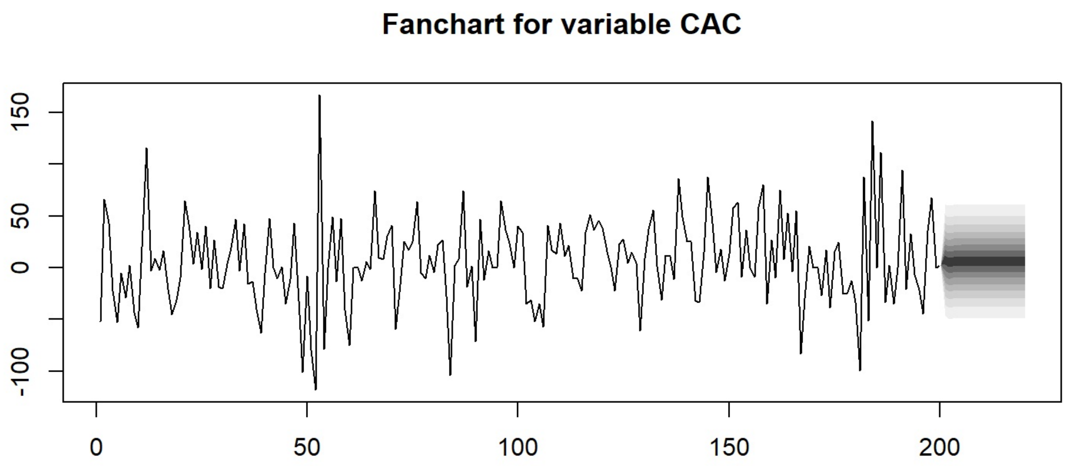
```
#i.forecast for each variable
forecast <- predict(model,n.ahead=20,ci=0.95)

#ii.fancharts for each
fanchart(forecast,names='DAX')
```



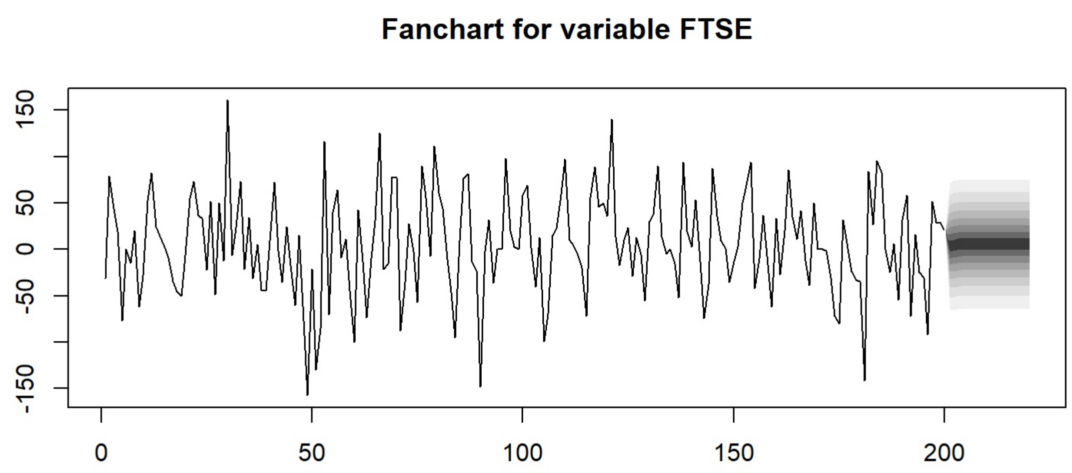

```
fanchart(forecast,names='SMI')
```




```
fanchart(forecast,names='CAC')
```




```
fanchart(forecast,names='FTSE')
```


Fanchart for variable FTSE
150
50
0
-50
-150
0
50
100
150
200