11225075

oanh

2025-02-10

library(Metrics)

## Warning: package 'Metrics' was built under R version 4.3.3

library(tseries)

## Warning: package 'tseries' was built under R version 4.3.3

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

library(urca)

## Warning: package 'urca' was built under R version 4.3.3

library(Hmisc)

## Warning: package 'Hmisc' was built under R version 4.3.3

##   
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':  
##   
## format.pval, units

library(forecast)

## Warning: package 'forecast' was built under R version 4.3.3

##   
## Attaching package: 'forecast'

## The following object is masked from 'package:Metrics':  
##   
## accuracy

library(car)

## Warning: package 'car' was built under R version 4.3.2

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.3.3

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.3.3

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.3.3

## Warning: package 'tibble' was built under R version 4.3.2

## Warning: package 'tidyr' was built under R version 4.3.2

## Warning: package 'readr' was built under R version 4.3.2

## Warning: package 'purrr' was built under R version 4.3.2

## Warning: package 'dplyr' was built under R version 4.3.3

## Warning: package 'stringr' was built under R version 4.3.2

## Warning: package 'forcats' was built under R version 4.3.2

## Warning: package 'lubridate' was built under R version 4.3.3

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ lubridate 1.9.3 ✔ tibble 3.2.1  
## ✔ purrr 1.0.2 ✔ tidyr 1.3.0

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ dplyr::recode() masks car::recode()  
## ✖ purrr::some() masks car::some()  
## ✖ dplyr::src() masks Hmisc::src()  
## ✖ dplyr::summarize() masks Hmisc::summarize()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(cowplot)

## Warning: package 'cowplot' was built under R version 4.3.3

##   
## Attaching package: 'cowplot'  
##   
## The following object is masked from 'package:lubridate':  
##   
## stamp

library(readxl)

## Warning: package 'readxl' was built under R version 4.3.3

data <- read\_excel("D:/Môn học kì 6/Timeseries/CTG doanh thu.xlsx")  
names(data)=c("Doanhthu")  
View(data)  
attach(data)

## 1. Định dạng số liệu thời gian tạo biến time và biến mùa vụ

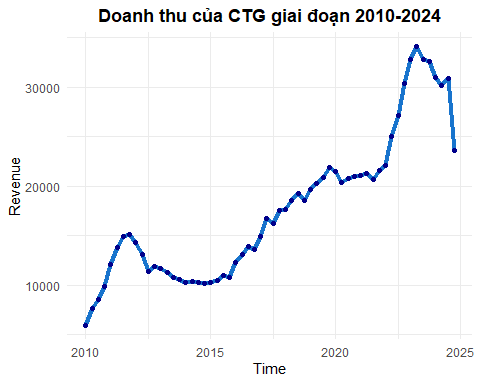
# Định dạng số liệu thời gian  
DT <- ts(data$Doanhthu, start=c(2010,1), frequency = 4)  
# Tạo biến xu thế thời gian  
t <- seq\_along(DT)   
summary(t)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 15.75 30.50 30.50 45.25 60.00

# Đồ thị   
library(ggplot2)  
library(dplyr)  
dt\_plot <- data.frame(Time = time(DT), Revenue = DT)  
dt\_plot %>% ggplot(aes(x=Time,y=Revenue)) +  
 geom\_line(col="dodgerblue3", size=1.5) +  
 geom\_point(col="darkblue") +   
 theme\_minimal() +  
 labs(title="Doanh thu của CTG giai đoạn 2010-2024") +  
 theme(plot.title = element\_text(hjust = 0.5, face = "bold"))

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

## Don't know how to automatically pick scale for object of type <ts>. Defaulting  
## to continuous.  
## Don't know how to automatically pick scale for object of type <ts>. Defaulting  
## to continuous.



## 2. Các mô hình dự báo ( 2010-2024)

### Chia dữ liệu thành traning set

DT\_train <- ts(Doanhthu[1:56], start=c(2010,1), frequency = 4)  
t\_train <- seq\_along(DT\_train)  
summary(t\_train)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 14.75 28.50 28.50 42.25 56.00

# 1. Mô hình lin - lin

reg1 <- lm(DT\_train ~ t\_train)  
summary(reg1)

##   
## Call:  
## lm(formula = DT\_train ~ t\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4449.1 -2316.3 -674.4 1172.1 7616.8   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6166.60 856.46 7.2 1.95e-09 \*\*\*  
## t\_train 376.54 26.14 14.4 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3162 on 54 degrees of freedom  
## Multiple R-squared: 0.7935, Adjusted R-squared: 0.7897   
## F-statistic: 207.5 on 1 and 54 DF, p-value: < 2.2e-16

rmse(DT\_train, fitted(reg1))

## [1] 3104.764

mape(DT\_train, fitted(reg1))

## [1] 0.1618684

# 2. Mô hình lin-log

reg2 <- lm(DT\_train ~ log(t\_train))  
summary(reg2)

##   
## Call:  
## lm(formula = DT\_train ~ log(t\_train))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -6705.4 -3744.3 -233.1 2529.0 12115.6   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -336.8 2297.5 -0.147 0.884   
## log(t\_train) 5599.8 717.2 7.808 2.01e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4768 on 54 degrees of freedom  
## Multiple R-squared: 0.5303, Adjusted R-squared: 0.5216   
## F-statistic: 60.97 on 1 and 54 DF, p-value: 2.014e-10

rmse(DT\_train, fitted(reg2))

## [1] 4682.524

mape(DT\_train, fitted(reg2))

## [1] 0.2550433

# 3. Mô hình log - lin

reg3 <- lm(log(DT\_train) ~ t\_train)  
summary(reg3)

##   
## Call:  
## lm(formula = log(DT\_train) ~ t\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.36664 -0.11896 -0.00374 0.07454 0.42426   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 9.028276 0.047356 190.65 <2e-16 \*\*\*  
## t\_train 0.022050 0.001445 15.26 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1748 on 54 degrees of freedom  
## Multiple R-squared: 0.8117, Adjusted R-squared: 0.8082   
## F-statistic: 232.7 on 1 and 54 DF, p-value: < 2.2e-16

rmse(DT\_train, exp(fitted(reg3)))

## [1] 2549.862

mape(DT\_train, exp(fitted(reg3)))

## [1] 0.1326691

# 4. Mô hình log - log

reg4 <- lm(log(DT\_train) ~ log(t\_train))  
summary(reg4)

##   
## Call:  
## lm(formula = log(DT\_train) ~ log(t\_train))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.40942 -0.17466 0.03677 0.12649 0.45567   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 8.55845 0.11608 73.729 < 2e-16 \*\*\*  
## log(t\_train) 0.35684 0.03624 9.848 1.17e-13 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2409 on 54 degrees of freedom  
## Multiple R-squared: 0.6423, Adjusted R-squared: 0.6357   
## F-statistic: 96.97 on 1 and 54 DF, p-value: 1.175e-13

rmse(DT\_train, exp(fitted(reg4)))

## [1] 4279.057

mape(DT\_train, exp(fitted(reg4)))

## [1] 0.1933827

# 5. Hồi quy với biến giả mùa vụ

s1 <- c(rep(c(1,0,0,0), 14))   
s2 <- c(rep(c(0,1,0,0), 14))  
s3 <- c(rep(c(0,0,1,0), 14))  
s4 <- c(rep(c(0,0,0,1), 14))  
  
reg5 <- lm(DT\_train ~ s2 + s3 + s4)  
summary(reg5)

##   
## Call:  
## lm(formula = DT\_train ~ s2 + s3 + s4)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -10270 -5734 -1564 3779 17264   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 16175.7 1890.2 8.558 1.69e-11 \*\*\*  
## s2 676.9 2673.1 0.253 0.801   
## s3 858.7 2673.1 0.321 0.749   
## s4 1353.6 2673.1 0.506 0.615   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 7072 on 52 degrees of freedom  
## Multiple R-squared: 0.005039, Adjusted R-squared: -0.05236   
## F-statistic: 0.08779 on 3 and 52 DF, p-value: 0.9664

rmse(DT\_train, fitted(reg5))

## [1] 6815.061

mape(DT\_train, fitted(reg5))

## [1] 0.3807118

# 6. Xu thế tuyến tính + Mùa vụ dạng cộng

reg6 <- lm(DT\_train ~ t\_train + s2 + s3 + s4)  
summary(reg6)

##   
## Call:  
## lm(formula = DT\_train ~ t\_train + s2 + s3 + s4)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4517.0 -2310.2 -687.2 1175.5 7479.9   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6015.32 1133.25 5.308 2.43e-06 \*\*\*  
## t\_train 376.31 26.94 13.966 < 2e-16 \*\*\*  
## s2 300.57 1229.13 0.245 0.808   
## s3 106.03 1230.02 0.086 0.932   
## s4 224.68 1231.49 0.182 0.856   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3251 on 51 degrees of freedom  
## Multiple R-squared: 0.7938, Adjusted R-squared: 0.7776   
## F-statistic: 49.08 on 4 and 51 DF, p-value: < 2.2e-16

rmse(DT\_train, fitted(reg6))

## [1] 3102.654

mape(DT\_train, fitted(reg6))

## [1] 0.1614211

#7. Xu thế tuyến tính + Mùa vụ dạng nhân

reg7 <- lm(DT\_train ~ t\_train + I(t\_train\*s2) + I(t\_train\*s3)+ I(t\_train\*s4))  
summary(reg7)

##   
## Call:  
## lm(formula = DT\_train ~ t\_train + I(t\_train \* s2) + I(t\_train \*   
## s3) + I(t\_train \* s4))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4565.8 -2206.3 -681.5 1135.4 7419.3   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6174.995 880.663 7.012 5.21e-09 \*\*\*  
## t\_train 368.514 36.612 10.065 1.03e-13 \*\*\*  
## I(t\_train \* s2) 11.527 38.534 0.299 0.766   
## I(t\_train \* s3) 6.296 38.052 0.165 0.869   
## I(t\_train \* s4) 12.536 37.604 0.333 0.740   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3249 on 51 degrees of freedom  
## Multiple R-squared: 0.7941, Adjusted R-squared: 0.7779   
## F-statistic: 49.16 on 4 and 51 DF, p-value: < 2.2e-16

rmse(DT\_train, fitted(reg7))

## [1] 3100.61

mape(DT\_train, fitted(reg7))

## [1] 0.1617478

# 8. Xu thế không tuyến tính + Mùa vụ dạng cộng

reg8 <- lm(log(DT\_train) ~ t\_train + s2 + s3 + s4)  
summary(reg8)

##   
## Call:  
## lm(formula = log(DT\_train) ~ t\_train + s2 + s3 + s4)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.35180 -0.12553 -0.01323 0.07221 0.42302   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 9.013454 0.062607 143.970 <2e-16 \*\*\*  
## t\_train 0.022024 0.001489 14.796 <2e-16 \*\*\*  
## s2 0.025128 0.067904 0.370 0.713   
## s3 0.016236 0.067953 0.239 0.812   
## s4 0.020847 0.068034 0.306 0.761   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1796 on 51 degrees of freedom  
## Multiple R-squared: 0.8123, Adjusted R-squared: 0.7975   
## F-statistic: 55.16 on 4 and 51 DF, p-value: < 2.2e-16

rmse(DT\_train, exp(fitted(reg8)))

## [1] 2548.503

mape(DT\_train, exp(fitted(reg8)))

## [1] 0.1327919

# 9. Xu thế không tuyến tính + Mùa vụ dạng nhân

reg9 <- lm(log(DT\_train) ~ t\_train + I(t\_train\*s2) + I(t\_train\*s3)+ I(t\_train\*s4))  
summary(reg9)

##   
## Call:  
## lm(formula = log(DT\_train) ~ t\_train + I(t\_train \* s2) + I(t\_train \*   
## s3) + I(t\_train \* s4))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.36652 -0.12061 -0.01139 0.07504 0.42467   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 9.0283424 0.0487387 185.240 < 2e-16 \*\*\*  
## t\_train 0.0218593 0.0020262 10.788 9.11e-15 \*\*\*  
## I(t\_train \* s2) 0.0004276 0.0021326 0.200 0.842   
## I(t\_train \* s3) 0.0001216 0.0021059 0.058 0.954   
## I(t\_train \* s4) 0.0001985 0.0020811 0.095 0.924   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1798 on 51 degrees of freedom  
## Multiple R-squared: 0.8118, Adjusted R-squared: 0.7971   
## F-statistic: 55.01 on 4 and 51 DF, p-value: < 2.2e-16

rmse(DT\_train, exp(fitted(reg9)))

## [1] 2546.72

mape(DT\_train, exp(fitted(reg9)))

## [1] 0.132852

# 10. Holt-Winter có mùa vụ dạng cộng

hw.dt.train.a <- HoltWinters(DT\_train, seasonal = "a") #a là "additive"  
hw.dt.train.a

## Holt-Winters exponential smoothing with trend and additive seasonal component.  
##   
## Call:  
## HoltWinters(x = DT\_train, seasonal = "a")  
##   
## Smoothing parameters:  
## alpha: 1  
## beta : 0.4931384  
## gamma: 0.9361004  
##   
## Coefficients:  
## [,1]  
## a 33027.0349  
## b 261.3418  
## s1 192.8182  
## s2 446.6110  
## s3 -249.6323  
## s4 -389.7969

rmse(DT\_train, fitted(hw.dt.train.a)[,1])

## [1] 1171.123

mape(DT\_train, fitted(hw.dt.train.a)[,1])

## [1] 0.05679071

# 11. Holt-Winter có mùa vụ dạng nhân

hw.dt.train.m <- HoltWinters(DT\_train, seasonal = "m") # m là "multiplicative"  
hw.dt.train.m

## Holt-Winters exponential smoothing with trend and multiplicative seasonal component.  
##   
## Call:  
## HoltWinters(x = DT\_train, seasonal = "m")  
##   
## Smoothing parameters:  
## alpha: 0.9682685  
## beta : 0.3398843  
## gamma: 1  
##   
## Coefficients:  
## [,1]  
## a 3.371011e+04  
## b 8.651010e+02  
## s1 1.005512e+00  
## s2 1.032101e+00  
## s3 9.898563e-01  
## s4 9.681736e-01

rmse(DT\_train, fitted(hw.dt.train.m)[,1])

## [1] 1297.908

mape(DT\_train, fitted(hw.dt.train.m)[,1])

## [1] 0.06035494

# Holt - Winter có mùa vụ dạng cộng trên toàn dữ liệu 2010 - 2024

hw.dt.all.a <- HoltWinters(DT, seasonal = "a") # m là "multiplicative"  
hw.dt.all.a

## Holt-Winters exponential smoothing with trend and additive seasonal component.  
##   
## Call:  
## HoltWinters(x = DT, seasonal = "a")  
##   
## Smoothing parameters:  
## alpha: 0.8274308  
## beta : 0.6407054  
## gamma: 0.2811853  
##   
## Coefficients:  
## [,1]  
## a 25021.695243  
## b -3591.966386  
## s1 -147.053677  
## s2 286.336730  
## s3 -5.422925  
## s4 -601.847448

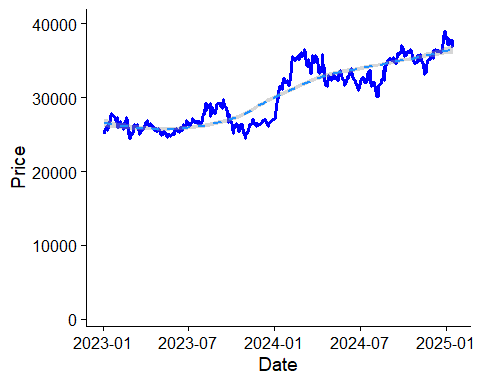
### Chuỗi giá và log-return

library(readxl)  
Chuỗi\_giá\_cổ\_phiếu <- read\_excel("D:/Môn học kì 6/Timeseries/Chuỗi giá cổ phiếu.xlsx")  
attach(Chuỗi\_giá\_cổ\_phiếu)

price <- ts(Price[1:499], start=1, frequency=1)  
logreturn <- ts(`Log-return(%)`[1:499], start=1, frequency=1)

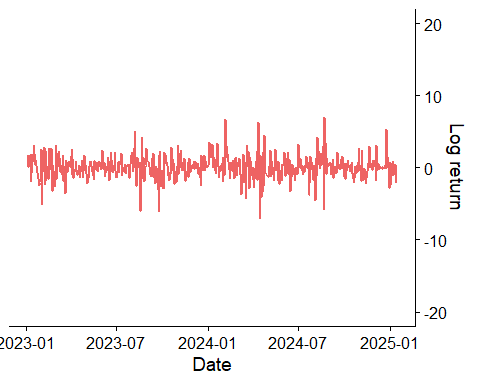
plot1 <- Chuỗi\_giá\_cổ\_phiếu %>% ggplot(aes(x=Date,y=ts(Price))) +  
 geom\_line(color="blue1", size=1.2) +  
 geom\_smooth(linetype="dashed", color="#1E90FF")+  
 labs(x="Date", y="Price") +  
 theme(plot.title = element\_text(hjust = 0.5, face = "bold")) +  
 scale\_y\_continuous(position="left", limits=c(1000,40000)) +  
 theme\_cowplot()  
plot1

## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'



plot2 <- Chuỗi\_giá\_cổ\_phiếu %>% ggplot(aes(x=Date, y=ts(`Log-return(%)`))) +  
 geom\_line(color="indianred2", size = 0.8) +  
 scale\_y\_continuous(position="right", limits=c(-20,20)) +  
 labs(x="Date", y="Log return") +  
 theme\_cowplot()  
plot2

## Warning: Removed 1 row containing missing values or values outside the scale range  
## (`geom\_line()`).



# 1. Kiểm định ADF cho chuỗi giá

summary(ur.df(Chuỗi\_giá\_cổ\_phiếu$Price,type="none",lag=0))

##   
## ###############################################   
## # Augmented Dickey-Fuller Test Unit Root Test #   
## ###############################################   
##   
## Test regression none   
##   
##   
## Call:  
## lm(formula = z.diff ~ z.lag.1 - 1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2472.46 -222.43 8.22 229.30 2329.81   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## z.lag.1 0.0006257 0.0007280 0.86 0.39  
##   
## Residual standard error: 501.9 on 507 degrees of freedom  
## Multiple R-squared: 0.001455, Adjusted R-squared: -0.0005145   
## F-statistic: 0.7388 on 1 and 507 DF, p-value: 0.3905  
##   
##   
## Value of test-statistic is: 0.8595   
##   
## Critical values for test statistics:   
## 1pct 5pct 10pct  
## tau1 -2.58 -1.95 -1.62

summary(ur.df(Chuỗi\_giá\_cổ\_phiếu$Price,type="drift",lag=0))

##   
## ###############################################   
## # Augmented Dickey-Fuller Test Unit Root Test #   
## ###############################################   
##   
## Test regression drift   
##   
##   
## Call:  
## lm(formula = z.diff ~ z.lag.1 + 1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2436.34 -230.42 -1.87 245.79 2339.86   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 221.718060 164.562064 1.347 0.178  
## z.lag.1 -0.006556 0.005380 -1.219 0.224  
##   
## Residual standard error: 501.5 on 506 degrees of freedom  
## Multiple R-squared: 0.002926, Adjusted R-squared: 0.0009559   
## F-statistic: 1.485 on 1 and 506 DF, p-value: 0.2235  
##   
##   
## Value of test-statistic is: -1.2186 1.2776   
##   
## Critical values for test statistics:   
## 1pct 5pct 10pct  
## tau2 -3.43 -2.86 -2.57  
## phi1 6.43 4.59 3.78

summary(ur.df(Chuỗi\_giá\_cổ\_phiếu$Price,type="trend",lag = 0))

##   
## ###############################################   
## # Augmented Dickey-Fuller Test Unit Root Test #   
## ###############################################   
##   
## Test regression trend   
##   
##   
## Call:  
## lm(formula = z.diff ~ z.lag.1 + 1 + tt)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2336.75 -243.18 -3.29 242.35 2258.35   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 840.86120 279.50925 3.008 0.00276 \*\*  
## z.lag.1 -0.03444 0.01152 -2.989 0.00294 \*\*  
## tt 0.88761 0.32498 2.731 0.00653 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 498.3 on 505 degrees of freedom  
## Multiple R-squared: 0.01744, Adjusted R-squared: 0.01355   
## F-statistic: 4.482 on 2 and 505 DF, p-value: 0.01176  
##   
##   
## Value of test-statistic is: -2.9885 3.3493 4.482   
##   
## Critical values for test statistics:   
## 1pct 5pct 10pct  
## tau3 -3.96 -3.41 -3.12  
## phi2 6.09 4.68 4.03  
## phi3 8.27 6.25 5.34

# 2. Kiểm định ADF cho chuỗi log-return

Data1<- Chuỗi\_giá\_cổ\_phiếu[!is.na(Chuỗi\_giá\_cổ\_phiếu$`Log-return(%)`), ]  
summary(ur.df(Data1$`Log-return(%)`,type="trend"))

##   
## ###############################################   
## # Augmented Dickey-Fuller Test Unit Root Test #   
## ###############################################   
##   
## Test regression trend   
##   
##   
## Call:  
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -6.3122 -0.8364 -0.0296 0.7932 6.4269   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.179e-02 1.428e-01 0.363 0.7170   
## z.lag.1 -1.042e+00 6.742e-02 -15.457 <2e-16 \*\*\*  
## tt 8.955e-05 4.867e-04 0.184 0.8541   
## z.diff.lag -9.228e-02 4.448e-02 -2.075 0.0385 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.599 on 502 degrees of freedom  
## Multiple R-squared: 0.5777, Adjusted R-squared: 0.5751   
## F-statistic: 228.9 on 3 and 502 DF, p-value: < 2.2e-16  
##   
##   
## Value of test-statistic is: -15.4569 79.6413 119.4588   
##   
## Critical values for test statistics:   
## 1pct 5pct 10pct  
## tau3 -3.96 -3.41 -3.12  
## phi2 6.09 4.68 4.03  
## phi3 8.27 6.25 5.34

summary(ur.df(Data1$`Log-return(%)`,type="drift"))

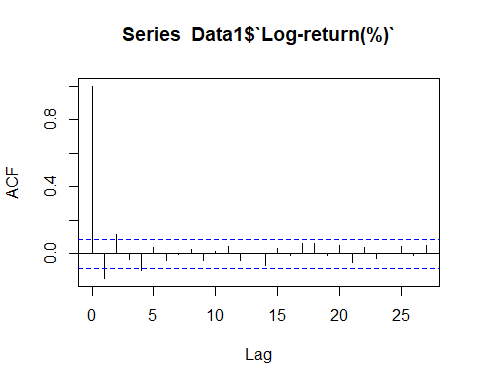
##   
## ###############################################   
## # Augmented Dickey-Fuller Test Unit Root Test #   
## ###############################################   
##   
## Test regression drift   
##   
##   
## Call:  
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -6.3072 -0.8399 -0.0338 0.7934 6.4326   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.07457 0.07120 1.047 0.2955   
## z.lag.1 -1.04198 0.06735 -15.471 <2e-16 \*\*\*  
## z.diff.lag -0.09237 0.04444 -2.079 0.0381 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.597 on 503 degrees of freedom  
## Multiple R-squared: 0.5776, Adjusted R-squared: 0.5759   
## F-statistic: 343.9 on 2 and 503 DF, p-value: < 2.2e-16  
##   
##   
## Value of test-statistic is: -15.4707 119.6749   
##   
## Critical values for test statistics:   
## 1pct 5pct 10pct  
## tau2 -3.43 -2.86 -2.57  
## phi1 6.43 4.59 3.78

summary(ur.df(Data1$`Log-return(%)`,type="none"))

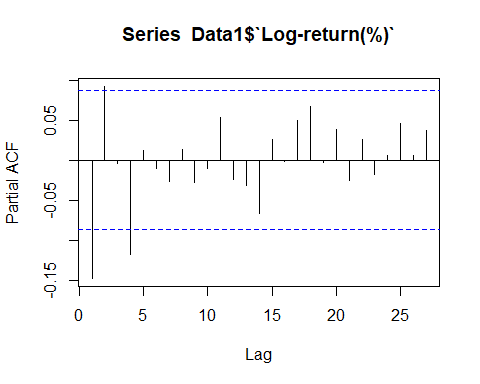
##   
## ###############################################   
## # Augmented Dickey-Fuller Test Unit Root Test #   
## ###############################################   
##   
## Test regression none   
##   
##   
## Call:  
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -6.2487 -0.7637 0.0437 0.8668 6.5115   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## z.lag.1 -1.03684 0.06718 -15.434 <2e-16 \*\*\*  
## z.diff.lag -0.09504 0.04437 -2.142 0.0327 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.597 on 504 degrees of freedom  
## Multiple R-squared: 0.5767, Adjusted R-squared: 0.575   
## F-statistic: 343.3 on 2 and 504 DF, p-value: < 2.2e-16  
##   
##   
## Value of test-statistic is: -15.434   
##   
## Critical values for test statistics:   
## 1pct 5pct 10pct  
## tau1 -2.58 -1.95 -1.62

# 3. Xem ACF, PACF của chuỗi log return

acf(Data1$`Log-return(%)`)



pacf(Data1$`Log-return(%)`)



Đề xuất mô hình ARIMA(4,0,3) , ARIMA(2,0,2)

# 3.1 Ước lượng mô hình ARIMA

#ARIMA(0,0,4)  
reg.loisuat.arima004=Arima(Data1$`Log-return(%)`,order=c(0,0,4),include.constant = TRUE)  
summary(reg.loisuat.arima004)

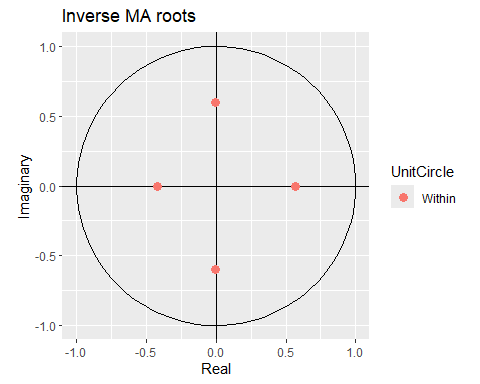
## Series: Data1$`Log-return(%)`   
## ARIMA(0,0,4) with non-zero mean   
##   
## Coefficients:  
## ma1 ma2 ma3 ma4 mean  
## -0.1331 0.1161 -0.0554 -0.0848 0.0748  
## s.e. 0.0444 0.0451 0.0470 0.0451 0.0592  
##   
## sigma^2 = 2.526: log likelihood = -953.74  
## AIC=1919.49 AICc=1919.65 BIC=1944.87  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 0.0009138504 1.581583 1.138357 NaN Inf 0.6507336 -0.001300926

#ARIMA(2,0,2)  
reg.loisuat.arima202=Arima(Data1$`Log-return(%)`,order=c(2,0,2),include.constant = TRUE)  
summary(reg.loisuat.arima202)

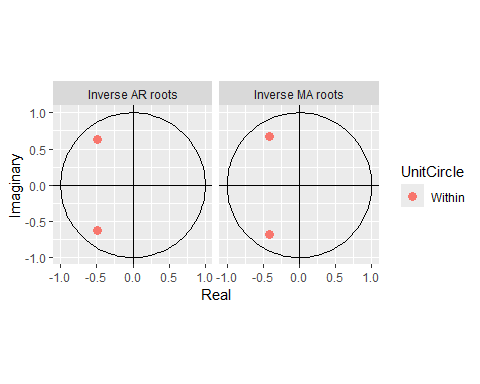
## Series: Data1$`Log-return(%)`   
## ARIMA(2,0,2) with non-zero mean   
##   
## Coefficients:  
## ar1 ar2 ma1 ma2 mean  
## -0.9717 -0.6360 0.8397 0.6326 0.0742  
## s.e. 0.2236 0.1519 0.2173 0.1351 0.0665  
##   
## sigma^2 = 2.526: log likelihood = -953.73  
## AIC=1919.46 AICc=1919.63 BIC=1944.84  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 0.0007397522 1.581522 1.132645 NaN Inf 0.647468 0.003649015

# 3.2 Xem tính dừng qua nghiệm nghịch đảo

autoplot(reg.loisuat.arima004)

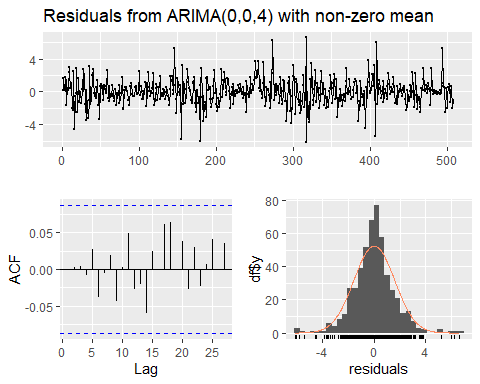


autoplot(reg.loisuat.arima202)



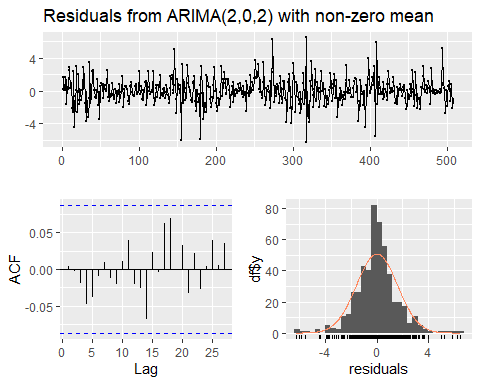
# 3.3 Kiểm định tính nhiễu trắng của phần dư

checkresiduals(reg.loisuat.arima004)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,0,4) with non-zero mean  
## Q\* = 2.3683, df = 6, p-value = 0.8829  
##   
## Model df: 4. Total lags used: 10

checkresiduals(reg.loisuat.arima202)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,0,2) with non-zero mean  
## Q\* = 2.6144, df = 6, p-value = 0.8555  
##   
## Model df: 4. Total lags used: 10

# 3.4 Dự báo log-return

logf.arima004<-forecast(reg.loisuat.arima004,h=10)  
logf.arima004

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 509 -0.041910254 -2.078839 1.995019 -3.157124 3.073304  
## 510 0.000713725 -2.054188 2.055615 -3.141987 3.143414  
## 511 0.301760509 -1.766714 2.370235 -2.861698 3.465219  
## 512 0.165895185 -1.905658 2.237449 -3.002272 3.334063  
## 513 0.074791805 -2.003946 2.153530 -3.104364 3.253947  
## 514 0.074791805 -2.003946 2.153530 -3.104364 3.253947  
## 515 0.074791805 -2.003946 2.153530 -3.104364 3.253947  
## 516 0.074791805 -2.003946 2.153530 -3.104364 3.253947  
## 517 0.074791805 -2.003946 2.153530 -3.104364 3.253947  
## 518 0.074791805 -2.003946 2.153530 -3.104364 3.253947

logf.arima202<-forecast(reg.loisuat.arima202,h=10)  
logf.arima202

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 509 -0.23477296 -2.271623 1.802077 -3.349867 2.880321  
## 510 0.09884051 -1.955677 2.153358 -3.043273 3.240954  
## 511 0.24684109 -1.823371 2.317054 -2.919276 3.412958  
## 512 -0.10913797 -2.180754 1.962478 -3.277401 3.059125  
## 513 0.14263356 -1.930839 2.216106 -3.028469 3.313737  
## 514 0.12438749 -1.953394 2.202169 -3.053305 3.302080  
## 515 -0.01800405 -2.097108 2.061100 -3.197719 3.161711  
## 516 0.13195853 -1.947186 2.211103 -3.047819 3.311736  
## 517 0.07680113 -2.003203 2.156805 -3.104291 3.257893  
## 518 0.03502357 -2.045577 2.115624 -3.146981 3.217028

log\_real <- ts(Chuỗi\_giá\_cổ\_phiếu$`Log-return(%)`, start=501, end=510, frequency=1)

# ARIMA(0,0,4)  
rmse(log\_real, logf.arima004$mean)

## [1] 2.214479

mape(log\_real, logf.arima004$mean)

## [1] 1.030265

# ARIMA(2,0,2)  
rmse(log\_real, logf.arima202$mean)

## [1] 2.183948

mape(log\_real, logf.arima202$mean)

## [1] 1.15397

# dự báo giá cổ phiếu 10 phiên đầu năm 2025

logf004 <- data.frame(logf.arima004$mean)  
logf202 <- data.frame(logf.arima202$mean)  
  
# ARIMA(0,0,4)  
price\_f1 <- exp(logf004[1,1]/100)\*Chuỗi\_giá\_cổ\_phiếu$Price[501] # Giá phiên 1   
  
k <- nrow(logf004)  
price\_f <- rep(0,k)  
# Tạo vòng lặp để tính giá   
for(i in 2:k){  
 price\_f[1] <- price\_f1  
 price\_f[i] <- exp(logf004[i,1]/100)\*price\_f[i-1]  
}  
price\_f # Giá 10 phiên

## [1] 37184.41 37184.68 37297.06 37358.98 37386.93 37414.91 37442.90 37470.91  
## [9] 37498.95 37527.01

# sai số ARIMA(0,0,4)  
close\_real <- ts(Chuỗi\_giá\_cổ\_phiếu$Price[500:509], start=501, end=510, frequency=1)  
rmse(close\_real, price\_f)

## [1] 445.8601

mape(close\_real, price\_f)

## [1] 0.008440755

# ARIMA(2,0,2)  
gia\_f1 <- exp(logf202[1,1]/100)\*Chuỗi\_giá\_cổ\_phiếu$Price[501] # Giá phiên 1   
  
h <- nrow(logf202)  
gia\_f <- rep(0,h)  
# Tạo vòng lặp để tính giá   
for(i in 2:h){  
 gia\_f[1] <- gia\_f1  
 gia\_f[i] <- exp(logf202[i,1]/100)\*gia\_f[i-1]  
}  
gia\_f # Giá 10 phiên

## [1] 37112.77 37149.47 37241.28 37200.66 37253.76 37300.13 37293.41 37342.65  
## [9] 37371.35 37384.44

# sai số ARIMA(2,0,2)  
rmse(close\_real, gia\_f)

## [1] 458.239

mape(close\_real, gia\_f)

## [1] 0.009581344

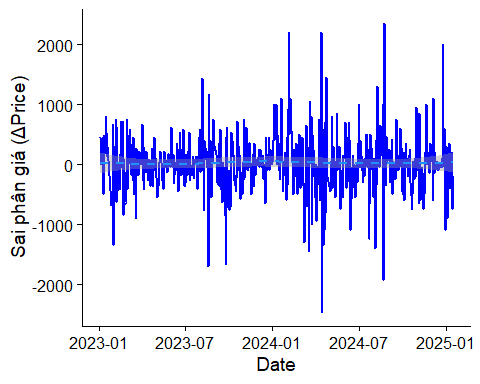
## Với sai phân của chuỗi price

Chuỗi\_giá\_cổ\_phiếu <- Chuỗi\_giá\_cổ\_phiếu %>%  
 mutate(dPrice = c(NA, diff(Price))) # thêm giá trị NA đầu tiên để giữ cùng chiều dài  
plot3 <- Chuỗi\_giá\_cổ\_phiếu %>%   
 ggplot(aes(x = Date, y = dPrice)) +  
 geom\_line(color = "blue1", size = 0.8) +  
 geom\_smooth(linetype = "dashed", color = "#1E90FF") +  
 labs(x = "Date", y = "Sai phân giá (ΔPrice)") +  
 theme(plot.title = element\_text(hjust = 0.5, face = "bold")) +  
 theme\_cowplot()  
plot3

## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'

## Warning: Removed 1 row containing non-finite outside the scale range  
## (`stat\_smooth()`).

## Warning: Removed 1 row containing missing values or values outside the scale range  
## (`geom\_line()`).



## KD ADF có xu thế  
summary(ur.df(diff(price), type="trend"))

##   
## ###############################################   
## # Augmented Dickey-Fuller Test Unit Root Test #   
## ###############################################   
##   
## Test regression trend   
##   
##   
## Call:  
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2171.33 -242.49 -10.36 235.18 2229.00   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.83225 44.81712 0.152 0.8789   
## z.lag.1 -1.05430 0.06863 -15.363 <2e-16 \*\*\*  
## tt 0.07759 0.15592 0.498 0.6190   
## z.diff.lag -0.08866 0.04511 -1.965 0.0499 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 496.8 on 492 degrees of freedom  
## Multiple R-squared: 0.5791, Adjusted R-squared: 0.5765   
## F-statistic: 225.6 on 3 and 492 DF, p-value: < 2.2e-16  
##   
##   
## Value of test-statistic is: -15.3625 78.6799 118.0128   
##   
## Critical values for test statistics:   
## 1pct 5pct 10pct  
## tau3 -3.98 -3.42 -3.13  
## phi2 6.15 4.71 4.05  
## phi3 8.34 6.30 5.36

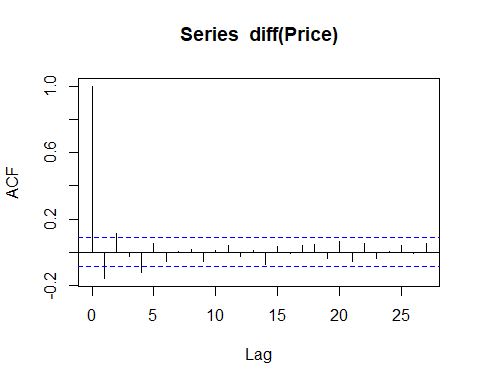
## KD ADF chỉ có hệ số chặn  
summary(ur.df(diff(price),type = "drift"))

##   
## ###############################################   
## # Augmented Dickey-Fuller Test Unit Root Test #   
## ###############################################   
##   
## Test regression drift   
##   
##   
## Call:  
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2167.7 -244.3 -12.4 241.7 2234.6   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 26.15377 22.36841 1.169 0.243   
## z.lag.1 -1.05290 0.06852 -15.367 <2e-16 \*\*\*  
## z.diff.lag -0.08933 0.04506 -1.983 0.048 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 496.4 on 493 degrees of freedom  
## Multiple R-squared: 0.5789, Adjusted R-squared: 0.5772   
## F-statistic: 338.9 on 2 and 493 DF, p-value: < 2.2e-16  
##   
##   
## Value of test-statistic is: -15.3668 118.0762   
##   
## Critical values for test statistics:   
## 1pct 5pct 10pct  
## tau2 -3.44 -2.87 -2.57  
## phi1 6.47 4.61 3.79

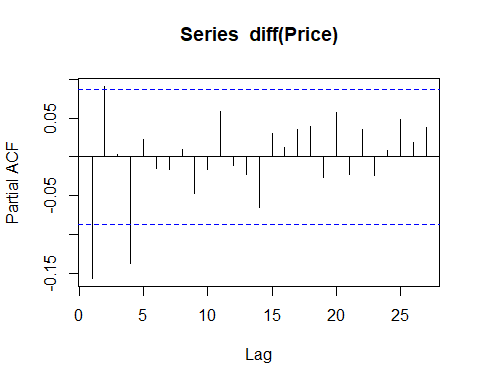
## KD ADF none  
summary(ur.df(diff(price),type = "none"))

##   
## ###############################################   
## # Augmented Dickey-Fuller Test Unit Root Test #   
## ###############################################   
##   
## Test regression none   
##   
##   
## Call:  
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2149.04 -217.81 14.23 268.30 2262.58   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## z.lag.1 -1.04619 0.06830 -15.317 <2e-16 \*\*\*  
## z.diff.lag -0.09272 0.04498 -2.061 0.0398 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 496.6 on 494 degrees of freedom  
## Multiple R-squared: 0.5777, Adjusted R-squared: 0.576   
## F-statistic: 337.9 on 2 and 494 DF, p-value: < 2.2e-16  
##   
##   
## Value of test-statistic is: -15.317   
##   
## Critical values for test statistics:   
## 1pct 5pct 10pct  
## tau1 -2.58 -1.95 -1.62

acf(diff(Price))



pacf(diff(Price))

 Đề xuất mô hình ARIMA(1,1,1,), ARIMA(4,1,4), ARIMA(1,1,4), ARIMA(4,1,1)

reg.price.arima111 <- Arima(price, order = c(1,1,1), include.constant = TRUE)  
summary(reg.price.arima111)

## Series: price   
## ARIMA(1,1,1) with drift   
##   
## Coefficients:  
## ar1 ma1 drift  
## -0.4373 0.2821 25.7488  
## s.e. 0.1511 0.1579 19.7918  
##   
## sigma^2 = 246566: log likelihood = -3796.57  
## AIC=7601.15 AICc=7601.23 BIC=7617.99  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.1257538 494.5601 345.4791 -0.01811645 1.139784 0.9987497  
## ACF1  
## Training set 0.01198832

reg.price.arima414 <- Arima(price, order = c(4,1,4), include.constant = TRUE)  
summary(reg.price.arima414)

## Series: price   
## ARIMA(4,1,4) with drift   
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ma1 ma2 ma3 ma4  
## -0.1089 0.2357 0.303 -0.1958 -0.0276 -0.1328 -0.3163 0.1018  
## s.e. 0.5227 0.3143 0.286 0.3014 0.5254 0.2891 0.2716 0.3045  
## drift  
## 25.3971  
## s.e. 17.9130  
##   
## sigma^2 = 243312: log likelihood = -3790.28  
## AIC=7600.57 AICc=7601.02 BIC=7642.68  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.4094675 488.2993 345.7504 -0.01813961 1.140686 0.9995337  
## ACF1  
## Training set -0.0004927134

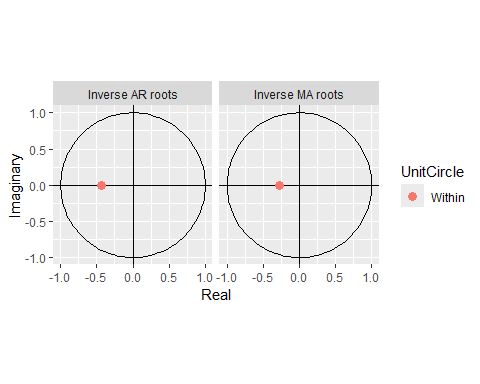
reg.price.arima114 <- Arima(price, order = c(1,1,4), include.constant = TRUE)  
summary(reg.price.arima114)

## Series: price   
## ARIMA(1,1,4) with drift   
##   
## Coefficients:  
## ar1 ma1 ma2 ma3 ma4 drift  
## -0.5325 0.3926 0.0428 0.0078 -0.1282 25.6306  
## s.e. 0.2943 0.2939 0.0597 0.0592 0.0432 18.8464  
##   
## sigma^2 = 242828: log likelihood = -3791.29  
## AIC=7596.59 AICc=7596.81 BIC=7626.06  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.305152 489.3074 345.8863 -0.0179134 1.141311 0.9999267  
## ACF1  
## Training set 0.0006555241

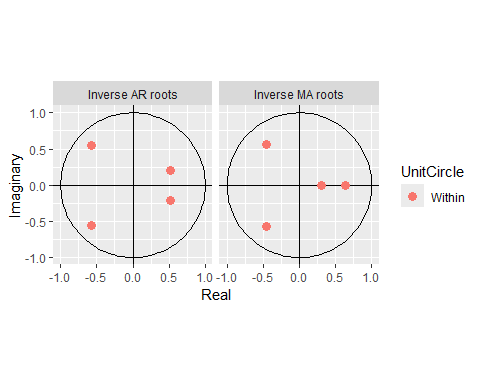
reg.price.arima411 <- Arima(price, order = c(4,1,1), include.constant = TRUE)  
summary(reg.price.arima411)

## Series: price   
## ARIMA(4,1,1) with drift   
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ma1 drift  
## -0.3368 0.0746 -0.0020 -0.1380 0.1981 25.6190  
## s.e. 0.2785 0.0616 0.0537 0.0452 0.2797 18.7501  
##   
## sigma^2 = 242399: log likelihood = -3790.86  
## AIC=7595.72 AICc=7595.95 BIC=7625.19  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.3123405 488.8751 346.6869 -0.01792226 1.143833 1.002241  
## ACF1  
## Training set 0.0005228244

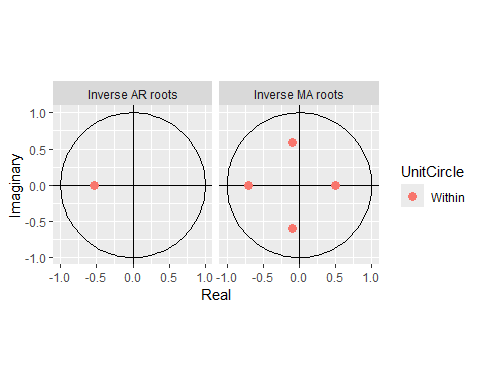
autoplot(reg.price.arima111)



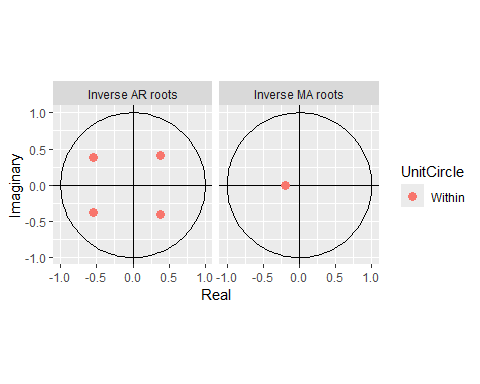
autoplot(reg.price.arima414)



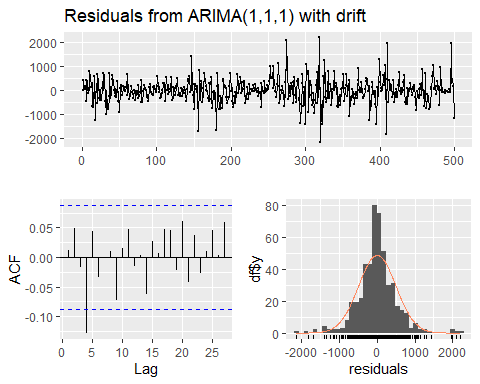
autoplot(reg.price.arima114)



autoplot(reg.price.arima411)

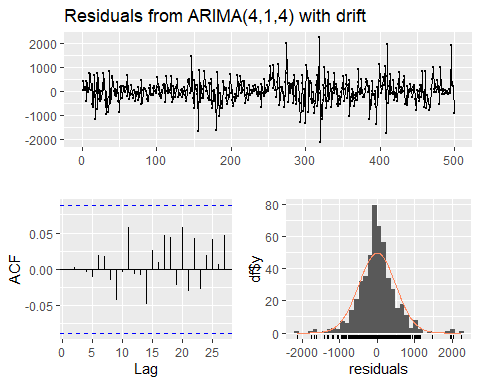


checkresiduals(reg.price.arima111)



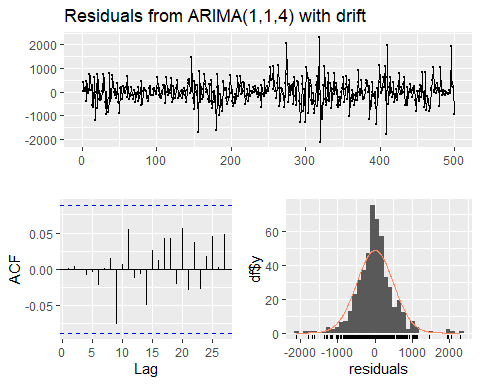
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,1) with drift  
## Q\* = 14.053, df = 8, p-value = 0.08038  
##   
## Model df: 2. Total lags used: 10

checkresiduals(reg.price.arima414)



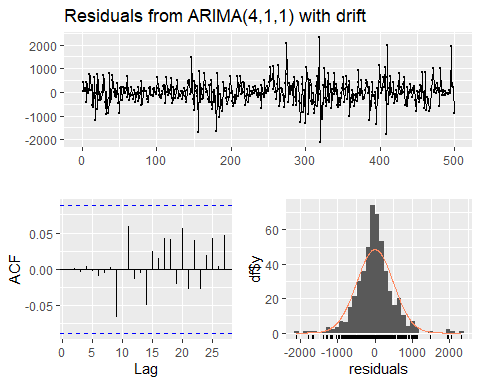
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(4,1,4) with drift  
## Q\* = 3.1894, df = 3, p-value = 0.3633  
##   
## Model df: 8. Total lags used: 11

checkresiduals(reg.price.arima114)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,4) with drift  
## Q\* = 3.4321, df = 5, p-value = 0.6337  
##   
## Model df: 5. Total lags used: 10

checkresiduals(reg.price.arima411)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(4,1,1) with drift  
## Q\* = 2.3508, df = 5, p-value = 0.7988  
##   
## Model df: 5. Total lags used: 10

pricef.arima111 <- forecast(reg.price.arima111,h=10)  
pricef.arima111

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 500 37994.58 37358.22 38630.94 37021.35 38967.80  
## 501 37946.49 37113.45 38779.54 36672.47 39220.52  
## 502 38004.53 36989.02 39020.04 36451.44 39557.62  
## 503 38016.16 36855.56 39176.76 36241.17 39791.14  
## 504 38048.08 36755.00 39341.17 36070.48 40025.69  
## 505 38071.13 36659.38 39482.88 35912.05 40230.22  
## 506 38098.06 36576.28 39619.84 35770.71 40425.42  
## 507 38123.29 36499.17 39747.42 35639.41 40607.18  
## 508 38149.27 36428.77 39869.76 35518.00 40780.54  
## 509 38174.92 36363.21 39986.62 35404.15 40945.68

pricef.arima414 <- forecast(reg.price.arima414,h=10)  
pricef.arima414

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 500 38050.10 37417.95 38682.25 37083.31 39016.89  
## 501 37795.28 36960.05 38630.51 36517.90 39072.65  
## 502 37859.41 36818.98 38899.84 36268.21 39450.61  
## 503 38009.32 36816.40 39202.23 36184.91 39833.72  
## 504 37901.38 36600.23 39202.52 35911.45 39891.30  
## 505 38037.26 36621.79 39452.72 35872.49 40202.03  
## 506 38049.33 36544.00 39554.67 35747.12 40351.55  
## 507 38037.44 36446.78 39628.10 35604.74 40470.14  
## 508 38123.35 36447.62 39799.08 35560.54 40686.16  
## 509 38107.69 36357.50 39857.89 35431.00 40784.39

pricef.arima114 <- forecast(reg.price.arima114,h=10)  
pricef.arima114

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 500 38038.90 37407.38 38670.42 37073.08 39004.72  
## 501 37862.72 37029.71 38695.73 36588.74 39136.70  
## 502 37963.83 36927.06 39000.60 36378.23 39549.43  
## 503 38068.60 36879.29 39257.91 36249.71 39887.49  
## 504 38052.09 36754.02 39350.16 36066.87 40037.32  
## 505 38100.16 36689.03 39511.29 35942.02 40258.30  
## 506 38113.84 36604.45 39623.24 35805.42 40422.27  
## 507 38145.84 36541.01 39750.67 35691.46 40600.21  
## 508 38168.08 36474.79 39861.36 35578.42 40757.74  
## 509 38195.51 36417.35 39973.67 35476.05 40914.97

pricef.arima411 <- forecast(reg.price.arima411,h=10)  
pricef.arima411

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 500 38032.77 37401.81 38663.73 37067.80 38997.74  
## 501 37825.25 36992.54 38657.97 36551.73 39098.78  
## 502 37936.86 36898.71 38975.01 36349.15 39524.58  
## 503 38071.03 36878.72 39263.33 36247.56 39894.50  
## 504 38038.39 36739.01 39337.76 36051.17 40025.61  
## 505 38123.72 36712.95 39534.49 35966.13 40281.31  
## 506 38112.80 36608.32 39617.27 35811.90 40413.69  
## 507 38140.32 36542.23 39738.40 35696.25 40584.38  
## 508 38170.49 36483.31 39857.67 35590.17 40750.81  
## 509 38186.55 36415.96 39957.14 35478.67 40894.43

close\_real <- ts(Chuỗi\_giá\_cổ\_phiếu$Price[500:509], start=500, end=509, frequency=1)  
# ARIMA(1,1,1)  
rmse(close\_real, pricef.arima111$mean)

## [1] 778.0506

mape(close\_real, pricef.arima111$mean)

## [1] 0.01824928

# ARIMA(4,1,4)  
rmse(close\_real, pricef.arima414$mean)

## [1] 718.0937

mape(close\_real, pricef.arima414$mean)

## [1] 0.01619924

# ARIMA(1,1,4)  
rmse(close\_real, pricef.arima114$mean)

## [1] 781.7677

mape(close\_real, pricef.arima114$mean)

## [1] 0.01823542

# ARIMA(4,1,1)  
rmse(close\_real, pricef.arima411$mean)

## [1] 776.3611

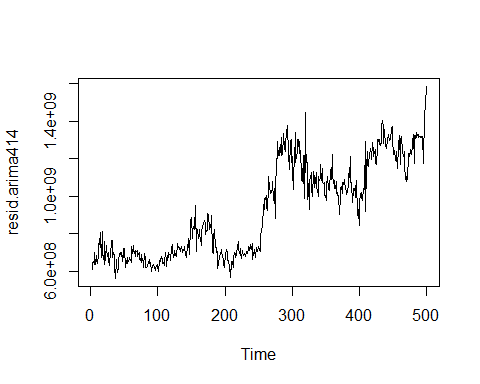
mape(close\_real, pricef.arima411$mean)

## [1] 0.01807621

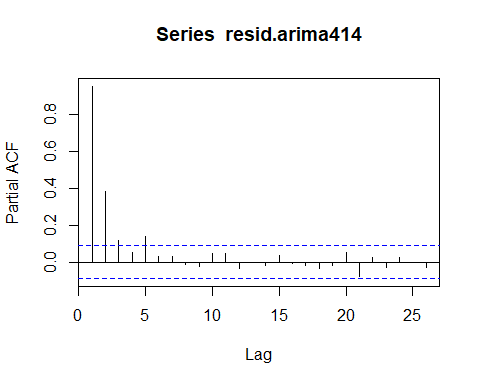
### Mô hình ARCH - GARCH

# Mô hình ARIMA(4,1,4) là mô hình tốt nhất cho dự báo chuỗi giá=> ARMA(4,4)

# lấy sai phân  
dprice <- diff(price)  
# lấy phần dư  
dprice1 <- dprice - fitted(reg.price.arima414)  
# bình phương phần dư  
resid.arima414 <- dprice1^2  
ts.plot(resid.arima414)



pacf(resid.arima414)



library(fGarch)

## Warning: package 'fGarch' was built under R version 4.3.3

## NOTE: Packages 'fBasics', 'timeDate', and 'timeSeries' are no longer  
## attached to the search() path when 'fGarch' is attached.  
##   
## If needed attach them yourself in your R script by e.g.,  
## require("timeSeries")

g1 <- garchFit(~arma(4,4)+garch(1,1), data=dprice, trace=FALSE)  
summary(g1)

##   
## Title:  
## GARCH Modelling   
##   
## Call:  
## garchFit(formula = ~arma(4, 4) + garch(1, 1), data = dprice,   
## trace = FALSE)   
##   
## Mean and Variance Equation:  
## data ~ arma(4, 4) + garch(1, 1)  
## <environment: 0x0000015c593f6760>  
## [data = dprice]  
##   
## Conditional Distribution:  
## norm   
##   
## Coefficient(s):  
## mu ar1 ar2 ar3 ar4 ma1   
## 3.89882 0.72687 0.45488 0.30010 -0.65223 -0.87398   
## ma2 ma3 ma4 omega alpha1 beta1   
## -0.28268 -0.36752 0.65112 13245.19175 0.07921 0.86841   
##   
## Std. Errors:  
## based on Hessian   
##   
## Error Analysis:  
## Estimate Std. Error t value Pr(>|t|)   
## mu 3.899e+00 2.677e+00 1.457 0.145221   
## ar1 7.269e-01 2.111e-01 3.443 0.000575 \*\*\*  
## ar2 4.549e-01 3.386e-01 1.343 0.179140   
## ar3 3.001e-01 1.904e-01 1.576 0.115053   
## ar4 -6.522e-01 1.157e-01 -5.638 1.72e-08 \*\*\*  
## ma1 -8.740e-01 2.137e-01 -4.091 4.30e-05 \*\*\*  
## ma2 -2.827e-01 3.835e-01 -0.737 0.461082   
## ma3 -3.675e-01 2.344e-01 -1.568 0.116987   
## ma4 6.511e-01 1.092e-01 5.960 2.52e-09 \*\*\*  
## omega 1.325e+04 9.007e+03 1.471 0.141416   
## alpha1 7.921e-02 2.743e-02 2.887 0.003886 \*\*   
## beta1 8.684e-01 5.734e-02 15.146 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Log Likelihood:  
## -3769.103 normalized: -7.56848   
##   
## Description:  
## Tue Apr 15 03:16:42 2025 by user: ADMIN   
##   
##   
## Standardised Residuals Tests:  
## Statistic p-Value  
## Jarque-Bera Test R Chi^2 259.1141267 0.000000e+00  
## Shapiro-Wilk Test R W 0.9553868 3.987250e-11  
## Ljung-Box Test R Q(10) 3.4434584 9.689834e-01  
## Ljung-Box Test R Q(15) 6.0549943 9.788091e-01  
## Ljung-Box Test R Q(20) 8.9885886 9.830391e-01  
## Ljung-Box Test R^2 Q(10) 5.6500937 8.437545e-01  
## Ljung-Box Test R^2 Q(15) 7.7974562 9.316458e-01  
## Ljung-Box Test R^2 Q(20) 12.1227622 9.117858e-01  
## LM Arch Test R TR^2 7.9604247 7.882154e-01  
##   
## Information Criterion Statistics:  
## AIC BIC SIC HQIC   
## 15.18515 15.28661 15.18403 15.22497

library(rugarch)

## Warning: package 'rugarch' was built under R version 4.3.3

## Loading required package: parallel

##   
## Attaching package: 'rugarch'

## The following object is masked from 'package:purrr':  
##   
## reduce

## The following object is masked from 'package:stats':  
##   
## sigma

model <- ugarchspec(variance.model=list(model="iGARCH",garchOrder=c(1,1)),  
 mean.model=list(armaOrder=c(4,4)))  
model\_fit <- ugarchfit(model,dprice)  
model\_fit

##   
## \*---------------------------------\*  
## \* GARCH Model Fit \*  
## \*---------------------------------\*  
##   
## Conditional Variance Dynamics   
## -----------------------------------  
## GARCH Model : iGARCH(1,1)  
## Mean Model : ARFIMA(4,0,4)  
## Distribution : norm   
##   
## Optimal Parameters  
## ------------------------------------  
## Estimate Std. Error t value Pr(>|t|)  
## mu 14.452644 1.2783e+01 1.1306 0.258211  
## ar1 0.939734 1.1012e-01 8.5340 0.000000  
## ar2 0.278417 1.8370e-01 1.5156 0.129614  
## ar3 0.298532 5.8178e-02 5.1313 0.000000  
## ar4 -0.609129 7.5129e-02 -8.1077 0.000000  
## ma1 -1.115279 1.7410e-03 -640.6908 0.000000  
## ma2 -0.057670 9.9400e-04 -58.0041 0.000000  
## ma3 -0.388533 1.5260e-03 -254.5975 0.000000  
## ma4 0.638064 1.0850e-03 587.9081 0.000000  
## omega 3561.817431 1.4445e+03 2.4657 0.013675  
## alpha1 0.099351 2.0563e-02 4.8314 0.000001  
## beta1 0.900649 NA NA NA  
##   
## Robust Standard Errors:  
## Estimate Std. Error t value Pr(>|t|)  
## mu 14.452644 7.0522e+01 0.20494 0.83762  
## ar1 0.939734 8.5226e-01 1.10264 0.27019  
## ar2 0.278417 1.4573e+00 0.19105 0.84849  
## ar3 0.298532 1.9678e-01 1.51709 0.12924  
## ar4 -0.609129 6.7171e-01 -0.90684 0.36449  
## ma1 -1.115279 1.4438e-02 -77.24563 0.00000  
## ma2 -0.057670 1.1162e-02 -5.16641 0.00000  
## ma3 -0.388533 1.2465e-02 -31.17077 0.00000  
## ma4 0.638064 8.4680e-03 75.35205 0.00000  
## omega 3561.817431 4.6445e+03 0.76689 0.44315  
## alpha1 0.099351 7.8484e-02 1.26587 0.20556  
## beta1 0.900649 NA NA NA  
##   
## LogLikelihood : -3767.966   
##   
## Information Criteria  
## ------------------------------------  
##   
## Akaike 15.177  
## Bayes 15.270  
## Shibata 15.176  
## Hannan-Quinn 15.213  
##   
## Weighted Ljung-Box Test on Standardized Residuals  
## ------------------------------------  
## statistic p-value  
## Lag[1] 1.738 0.1874  
## Lag[2\*(p+q)+(p+q)-1][23] 7.716 1.0000  
## Lag[4\*(p+q)+(p+q)-1][39] 12.439 0.9953  
## d.o.f=8  
## H0 : No serial correlation  
##   
## Weighted Ljung-Box Test on Standardized Squared Residuals  
## ------------------------------------  
## statistic p-value  
## Lag[1] 0.0191 0.8901  
## Lag[2\*(p+q)+(p+q)-1][5] 1.5556 0.7262  
## Lag[4\*(p+q)+(p+q)-1][9] 3.3561 0.6990  
## d.o.f=2  
##   
## Weighted ARCH LM Tests  
## ------------------------------------  
## Statistic Shape Scale P-Value  
## ARCH Lag[3] 0.0005118 0.500 2.000 0.9820  
## ARCH Lag[5] 1.6207510 1.440 1.667 0.5611  
## ARCH Lag[7] 2.4727812 2.315 1.543 0.6180  
##   
## Nyblom stability test  
## ------------------------------------  
## Joint Statistic: 1.6207  
## Individual Statistics:   
## mu 0.08175  
## ar1 0.05848  
## ar2 0.10140  
## ar3 0.14578  
## ar4 0.16700  
## ma1 0.04140  
## ma2 0.05482  
## ma3 0.07265  
## ma4 0.09345  
## omega 0.14728  
## alpha1 0.18498  
##   
## Asymptotic Critical Values (10% 5% 1%)  
## Joint Statistic: 2.49 2.75 3.27  
## Individual Statistic: 0.35 0.47 0.75  
##   
## Sign Bias Test  
## ------------------------------------  
## t-value prob sig  
## Sign Bias 0.08615 0.9314   
## Negative Sign Bias 0.46511 0.6421   
## Positive Sign Bias 0.04063 0.9676   
## Joint Effect 0.27347 0.9649   
##   
##   
## Adjusted Pearson Goodness-of-Fit Test:  
## ------------------------------------  
## group statistic p-value(g-1)  
## 1 20 43.85 0.0009914  
## 2 30 49.11 0.0112470  
## 3 40 61.20 0.0130967  
## 4 50 69.67 0.0276890  
##   
##   
## Elapsed time : 0.6130161

f2025 <- ugarchforecast(model\_fit, n.ahead = 10)

## Warning in `setfixed<-`(`\*tmp\*`, value = as.list(pars)): Unrecognized Parameter  
## in Fixed Values: beta1...Ignored

f2025

##   
## \*------------------------------------\*  
## \* GARCH Model Forecast \*  
## \*------------------------------------\*  
## Model: iGARCH  
## Horizon: 10  
## Roll Steps: 0  
## Out of Sample: 0  
##   
## 0-roll forecast [T0=499-01-01]:  
## Series Sigma  
## T+1 399.018 650.7  
## T+2 -47.928 653.4  
## T+3 146.110 656.1  
## T+4 181.513 658.8  
## T+5 -44.772 661.5  
## T+6 82.611 664.2  
## T+7 31.691 666.9  
## T+8 -69.813 669.6  
## T+9 -3.512 672.2  
## T+10 -62.262 674.9