My R Markdown Document

Modeling and Forecasting the Morocco Stock Index 20

```
# Loading libraries
# Date manipulation
suppressPackageStartupMessages(library(lubridate))

# Descriptive statistics
suppressPackageStartupMessages(library(fBasics))
```

```
## Warning: le package 'fBasics' a été compilé avec la version R 4.4.2
```

```
# Coefficient significance tests
suppressPackageStartupMessages(library(lmtest))
```

```
## Warning: le package 'lmtest' a été compilé avec la version R 4.4.2
```

```
# Unit root test
suppressPackageStartupMessages(library(urca))
```

```
## Warning: le package 'urca' a été compilé avec la version R 4.4.2
```

```
# Visualisation
suppressPackageStartupMessages(library(ggplot2))

# Return calculations
suppressPackageStartupMessages(library(quantmod))
```

```
## Warning: le package 'quantmod' a été compilé avec la version R 4.4.2

## Warning: le package 'xts' a été compilé avec la version R 4.4.2

## Warning: le package 'TTR' a été compilé avec la version R 4.4.2
```

```
suppressPackageStartupMessages(library(PerformanceAnalytics))
```

```
## Warning: le package 'PerformanceAnalytics' a été compilé avec la version R
## 4.4.2
```

```
suppressPackageStartupMessages(library(forecast))
```

```
## Warning: le package 'forecast' a été compilé avec la version R 4.4.2
```

```
# Changements structurels
suppressPackageStartupMessages(library(strucchange))
```

```
## Warning: le package 'strucchange' a été compilé avec la version R 4.4.2

## Warning: le package 'sandwich' a été compilé avec la version R 4.4.2
```

```
# Loading the 'tseries' library for ADF testing
library(tseries)
```

```
## Warning: le package 'tseries' a été compilé avec la version R 4.4.2
```

```r
library(timeSeries)
```

```
## Warning: le package 'timeSeries' a été compilé avec la version R 4.4.2
```

```
## Le chargement a nécessité le package : timeDate
```

```
##
## Attachement du package : 'timeDate'
```

```
## Les objets suivants sont masqués depuis 'package:PerformanceAnalytics':
##
##     kurtosis, skewness
```

```
##
## Attachement du package : 'timeSeries'
```

```
## L'objet suivant est masqué depuis 'package:zoo':
##
##     time<-
```

```
## Les objets suivants sont masqués depuis 'package:graphics':
##
##     lines, points
```

```r
library(xts)
library(pastecs)
```

```
## Warning: le package 'pastecs' a été compilé avec la version R 4.4.2
```

```
##
## Attachement du package : 'pastecs'
```

```
## Les objets suivants sont masqués depuis 'package:xts':
##
##     first, last
```

```r
# For MLP model
library(nnfor)
```

```
## Warning: le package 'nnfor' a été compilé avec la version R 4.4.2
```

```
## Le chargement a nécessité le package : generics
```

```
##
## Attachement du package : 'generics'
```

```
## L'objet suivant est masqué depuis 'package:sandwich':
##
##     estfun
```

```
## L'objet suivant est masqué depuis 'package:lubridate':
##
##     as.difftime
```

```
## Les objets suivants sont masqués depuis 'package:base':
##
##     as.difftime, as.factor, as.ordered, intersect, is.element, setdiff,
##     setequal, union
```

```r
library(readr)
dailyMSI20 <- read.csv("MSI20_update.csv", head = TRUE)
head(dailyMSI20)
```

```
##         Date   Open  Close   High    Low
## 1 2021-01-04 924.78 928.96 928.97 922.30
## 2 2021-01-05 925.99 919.52 926.79 919.52
## 3 2021-01-06 919.52 915.67 921.57 915.01
## 4 2021-01-07 915.67 921.52 924.79 915.67
## 5 2021-01-08 921.52 919.60 924.31 918.90
## 6 2021-01-12 919.60 923.89 923.89 916.78
```

```
# Clean data
# Convert Date column to time format
dailyMSI20$Date <- as.Date(dailyMSI20$Date)

# Check updated format
str(dailyMSI20)
```

```
## 'data.frame':    590 obs. of  5 variables:
##  $ Date : Date, format: "2021-01-04" "2021-01-05" ...
##  $ Open : num  925 926 920 916 922 ...
##  $ Close: num  929 920 916 922 920 ...
##  $ High : num  929 927 922 925 924 ...
##  $ Low  : num  922 920 915 916 919 ...
```

```
head(dailyMSI20)
```

```
##         Date   Open  Close   High    Low
## 1 2021-01-04 924.78 928.96 928.97 922.30
## 2 2021-01-05 925.99 919.52 926.79 919.52
## 3 2021-01-06 919.52 915.67 921.57 915.01
## 4 2021-01-07 915.67 921.52 924.79 915.67
## 5 2021-01-08 921.52 919.60 924.31 918.90
## 6 2021-01-12 919.60 923.89 923.89 916.78
```

```
tail(dailyMSI20)
```

```
##           Date   Open  Close   High    Low
## 585 2023-04-27 838.25 840.12 843.95 838.25
## 586 2023-04-28 840.12 848.13 848.13 840.12
## 587 2023-05-02 848.13 840.54 852.01 840.54
## 588 2023-05-03 840.54 838.55 841.44 834.71
## 589 2023-05-04 838.55 837.13 838.55 831.77
## 590 2023-05-05 837.13 835.44 840.93 833.74
```

```
dim(dailyMSI20)
```

```
## [1] 590   5
```

```
#count the total number of missing values in the dataset
sum(is.na(dailyMSI20))
```

```
## [1] 0
```

```
t(stat.desc(dailyMSI20))
```

```
##       nbr.val nbr.null nbr.na      min      max  range        sum    median
## Date      590        0      0 18631.00 19482.00 851.00 11243991.0 19059.500
## Open      590        0      0   775.38  1140.69 365.31   575589.3   982.695
## Close     590        0      0   775.38  1140.69 365.31   575563.4   982.695
## High      590        0      0   797.01  1142.56 345.55   578174.3   985.980
```

```
## Low       590        0      0   775.38  1135.86 360.48   573290.4   980.045
##              mean   SE.mean CI.mean.0.95       var   std.dev   coef.var
## Date  19057.6119 10.157295    19.948925 60870.679 246.71984 0.01294600
## Open    975.5752  3.543246     6.958935  7407.210  86.06515 0.08821991
## Close   975.5312  3.550503     6.973188  7437.583  86.24142 0.08840458
## High    979.9565  3.512745     6.899031  7280.233  85.32428 0.08706946
## Low     971.6786  3.552519     6.977147  7446.033  86.29040 0.08880549
```

```
#Prices in time series format
Close_Price <- xts(dailyMSI20[,3],order.by=as.Date(dailyMSI20[,1]))
head(Close_Price)
```

```
##              [,1]
## 2021-01-04 928.96
## 2021-01-05 919.52
## 2021-01-06 915.67
## 2021-01-07 921.52
## 2021-01-08 919.60
## 2021-01-12 923.89
```

```
tail(Close_Price)
```

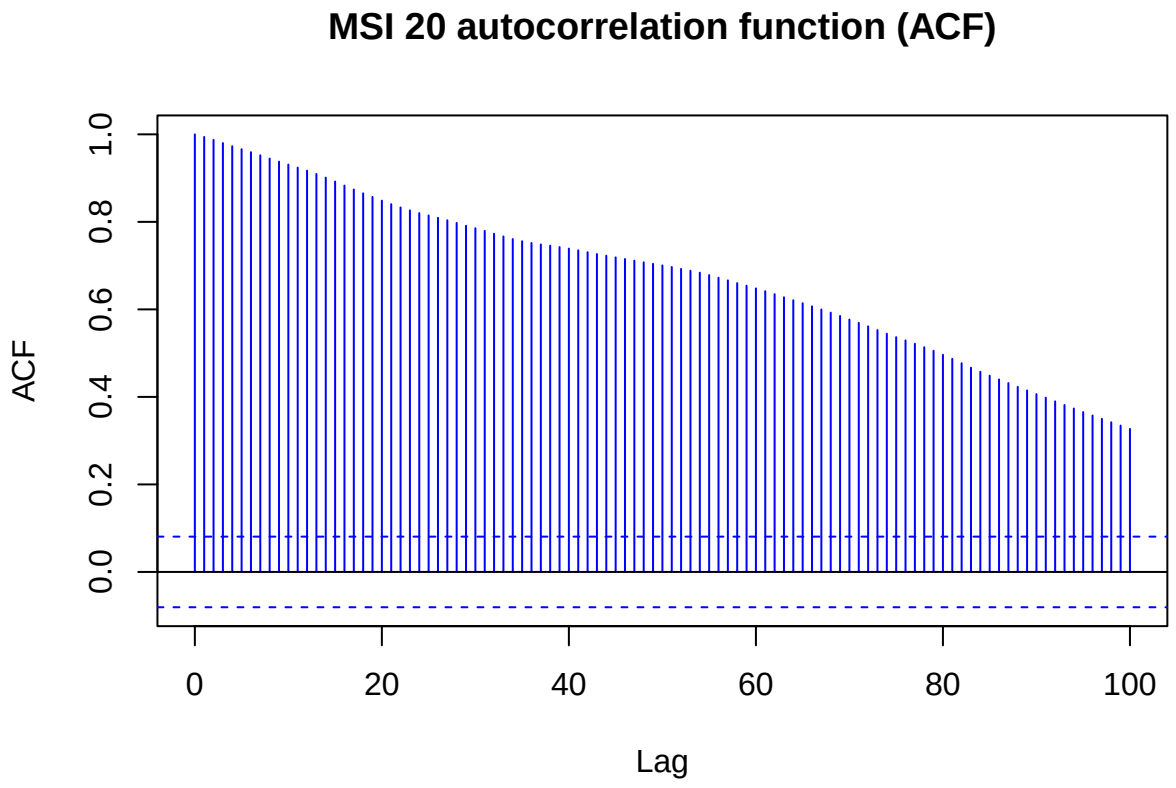
```
##              [,1]
## 2023-04-27 840.12
## 2023-04-28 848.13
## 2023-05-02 840.54
## 2023-05-03 838.55
## 2023-05-04 837.13
## 2023-05-05 835.44
```

```
autoplot(Close_Price) +
  labs(title = "MSI 20 daily closing price",
       x = "Date: January 4, 2021 to May 5, 2023",
       y = "Closing price")
```

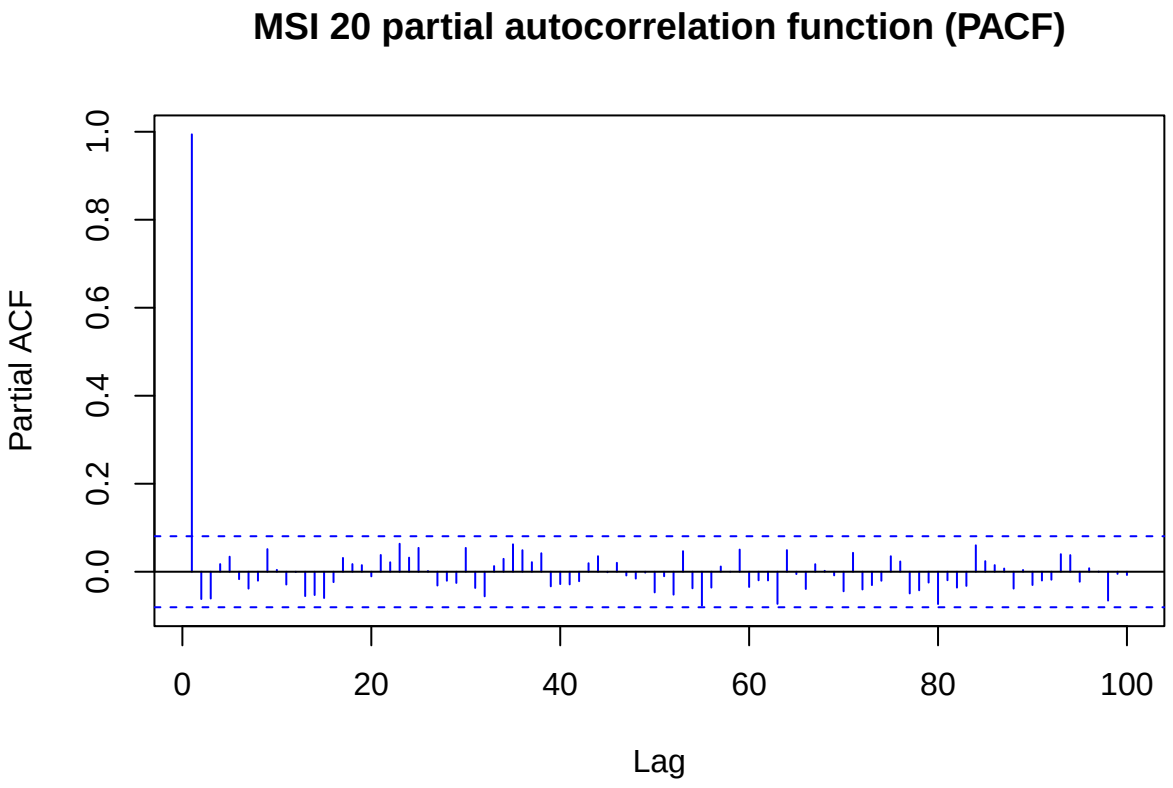
Here we see the corresponding chart, as produced by chartSeries in the quantmod package.

```
chart_Series(Close_Price, name = deparse(substitute(Daily_closing_price_of_MSI_20)),col = "blue")
```


```r
# Plot ACF
acf(Close_Price, main = "MSI 20 autocorrelation function (ACF)", col = "blue",
    lag.max = 100)
```




```
# Plot PACF
pacf(Close_Price, main = "MSI 20 partial autocorrelation function (PACF)",
     col = "blue",lag.max = 100)
```



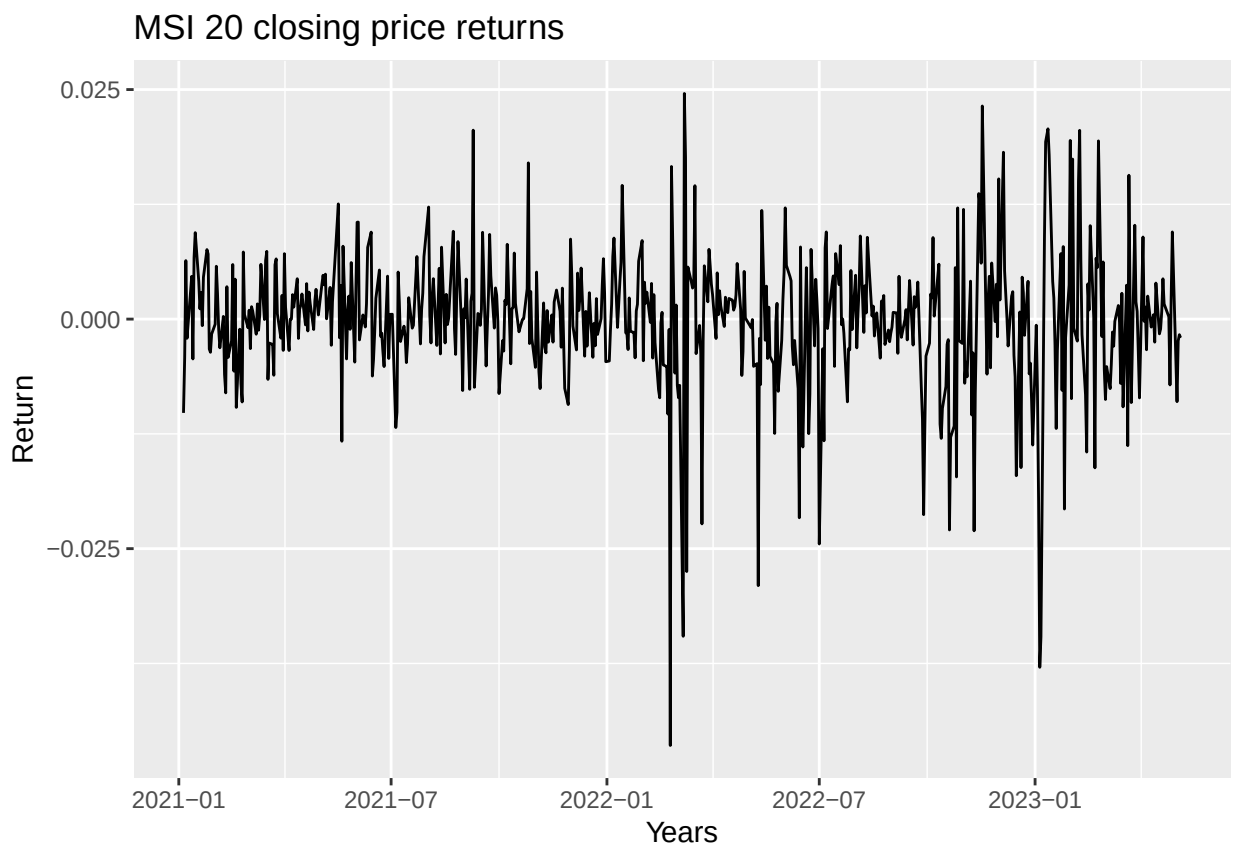

```
#Perform ADF test
adf_test <- adf.test(Close_Price)

#Print test results
print(adf_test)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  Close_Price
## Dickey-Fuller = -1.701, Lag order = 8, p-value = 0.7049
## alternative hypothesis: stationary
```

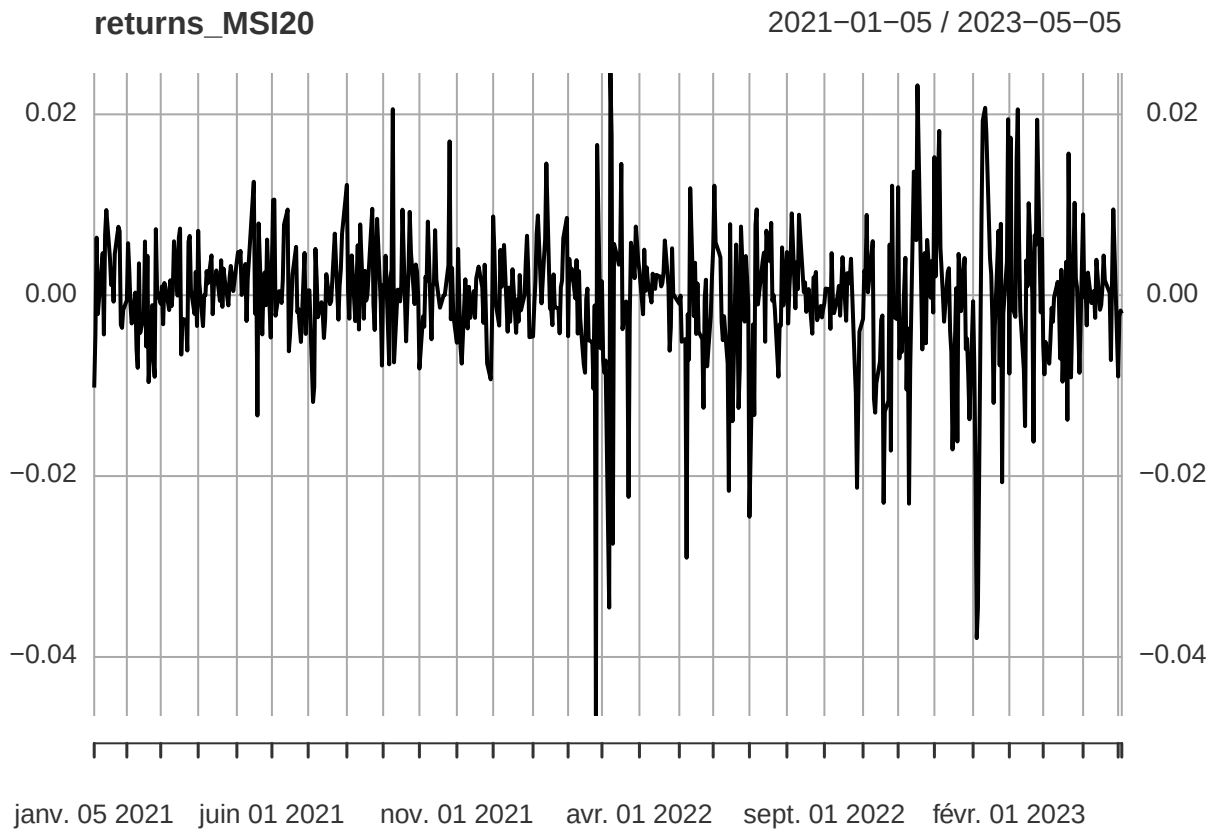
```
returns_MSI20 <- diff(log(Close_Price))[-1,]
head(returns_MSI20)
```

```
##                    [,1]
## 2021-01-05 -0.010213886
## 2021-01-06 -0.004195757
## 2021-01-07  0.006368443
## 2021-01-08 -0.002085688
## 2021-01-12  0.004654224
## 2021-01-13 -0.004338919
```

```
autoplot(returns_MSI20) +
  labs(x = "Years",
       y = "Return",
       title = "MSI 20 closing price returns")
```




```
plot(returns_MSI20)
```




```
library(gridExtra)
```

```
## Warning: le package 'gridExtra' a été compilé avec la version R 4.4.2
```

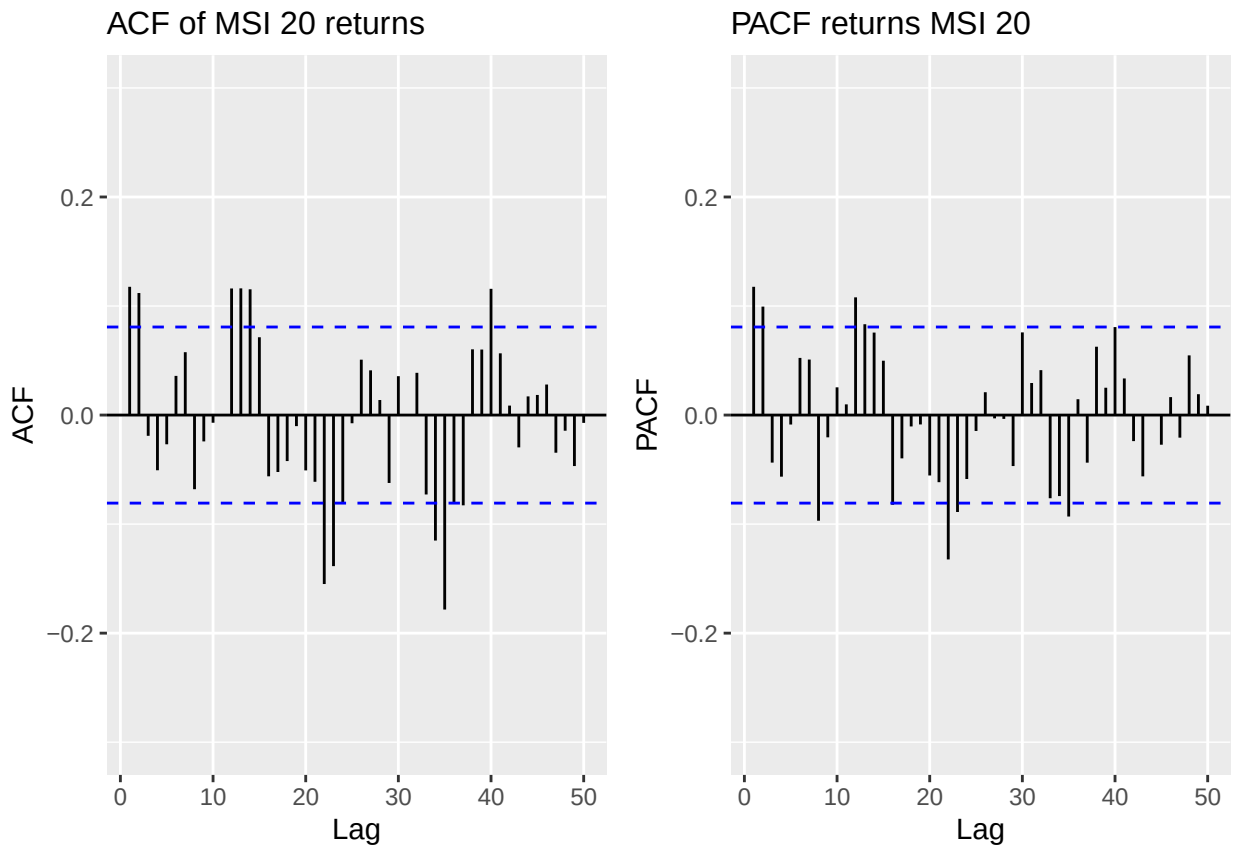
```
ACF_plot <- ggAcf(returns_MSI20, main="ACF of MSI 20 returns",lag=50, ylim=c(-0.3,0.3))+
  theme(plot.title = element_text(size = 12))
```

```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown
## parameters: `main` and `ylim`
```

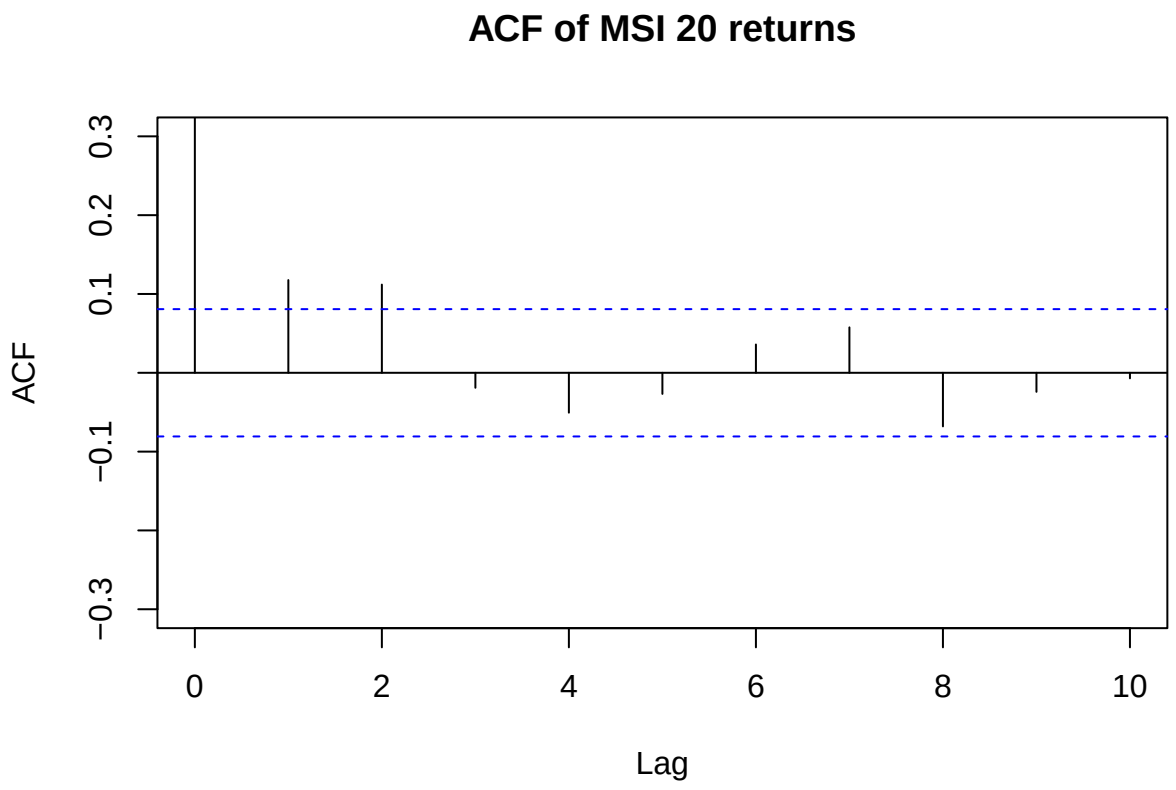
```
PACF_plot <- ggPacf(returns_MSI20, main= "PACF returns MSI 20", lag=50,ylim=c(-0.3,0.3))+
  theme(plot.title = element_text(size = 12))
```

```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown
## parameters: `main` and `ylim`
```

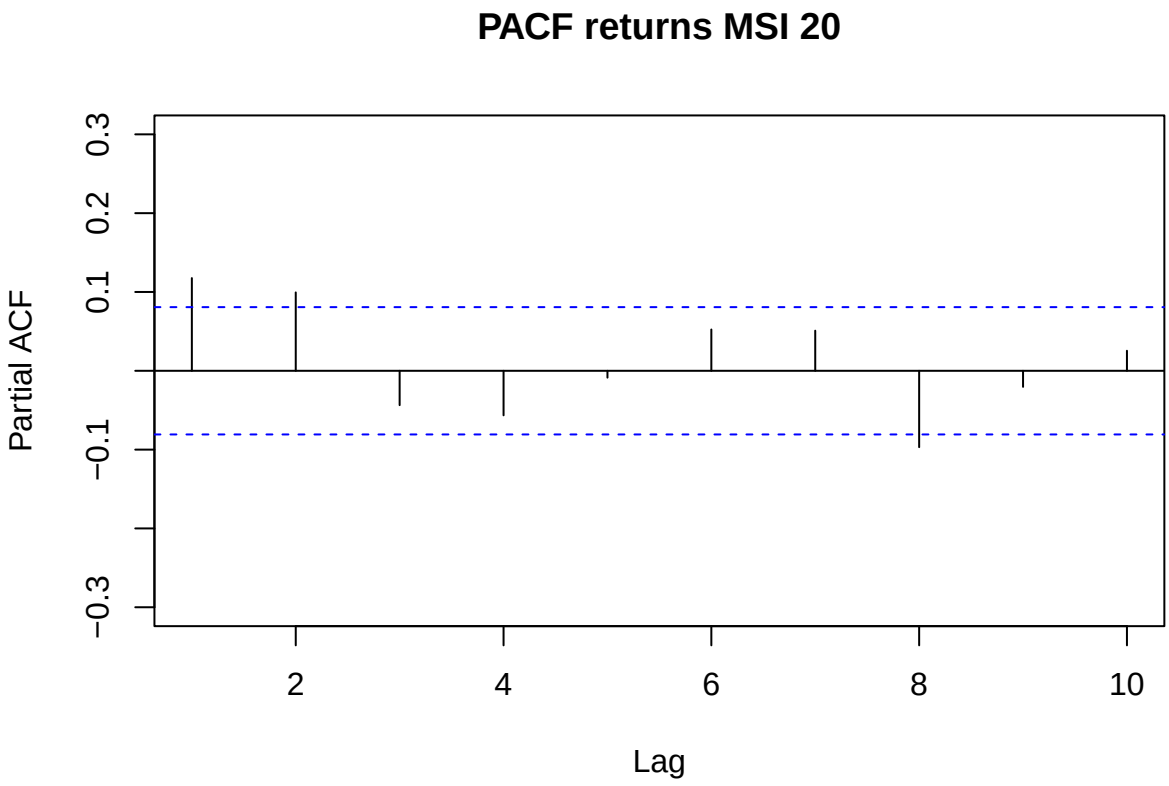
```
grid.arrange(ACF_plot, PACF_plot, nrow = 1)
```



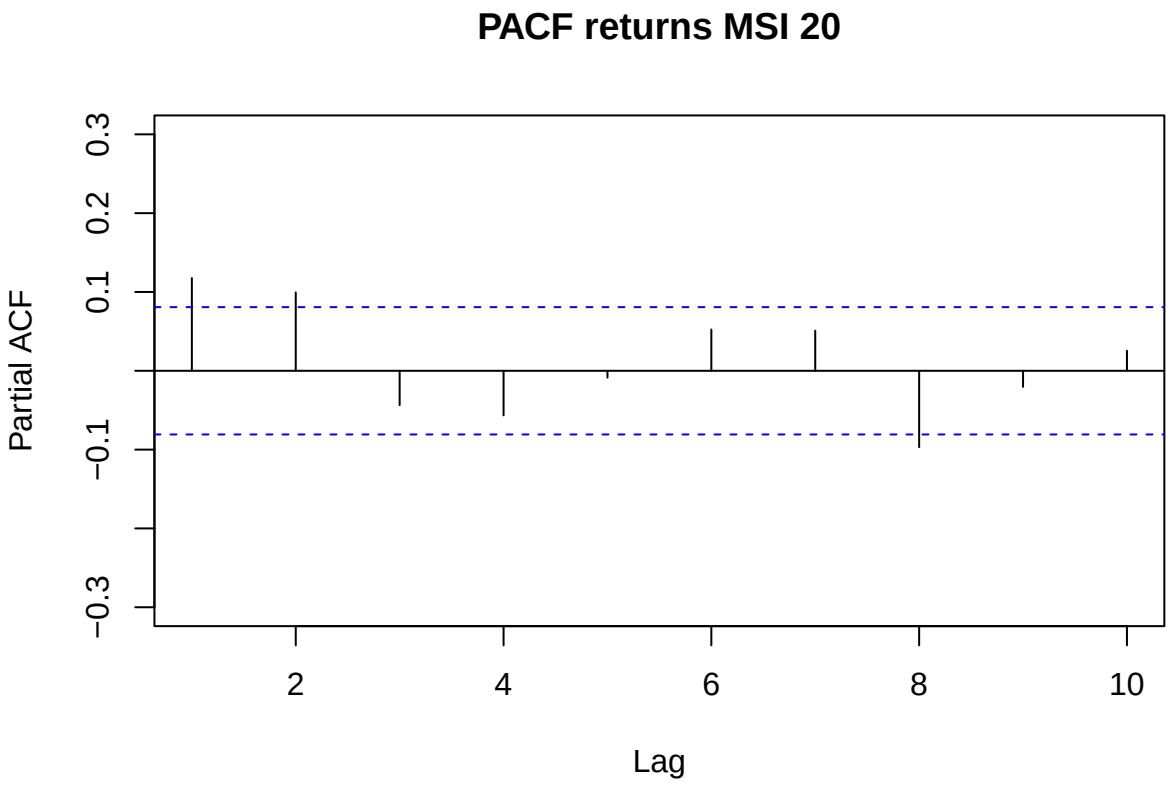

```
ACF_plot <- acf(returns_MSI20, main="ACF of MSI 20 returns",lag=10, ylim=c(-0.3,0.3))+
  theme(plot.title = element_text(size = 12),las=1)
```




```
PACF_plot <- pacf(returns_MSI20, main= "PACF returns MSI 20", lag=10,ylim=c(-0.3,0.3))+
  theme(plot.title = element_text(size = 12))
```




```
pacf(returns_MSI20, main= "PACF returns MSI 20", lag=10,ylim=c(-0.3,0.3))+
  theme(plot.title = element_text(size = 12))
```




```
## NULL
```

```
# Phillips-Perron (PP) test
pp_test <- pp.test(returns_MSI20)
```

```
## Warning in pp.test(returns_MSI20): p-value smaller than printed p-value
```

```
print(pp_test)
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  returns_MSI20
## Dickey-Fuller Z(alpha) = -531.84, Truncation lag parameter = 6, p-value
## = 0.01
## alternative hypothesis: stationary
```

```
#  Kwiatkowski-Phillips-Schmidt-Shin-Test (KPSS)
kpss_test <- kpss.test(returns_MSI20)
print(kpss_test)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  returns_MSI20
## KPSS Level = 0.36255, Truncation lag parameter = 6, p-value = 0.0933
```

```
# Augmented Dickey-Fuller (ADF) test
Adf_test <- adf.test(returns_MSI20)
```

```
## Warning in adf.test(returns_MSI20): p-value smaller than printed p-value
```

```
print(Adf_test)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  returns_MSI20
## Dickey-Fuller = -8.4448, Lag order = 8, p-value = 0.01
## alternative hypothesis: stationary
```

```
#  Augmented Dickey-Fuller (ADF) tests
adf_test <- ur.df(returns_MSI20)
summary(adf_test)
```

```
##
## ###############################################
## # 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
## -0.045280 -0.003468  0.000159  0.003296  0.028929
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -0.79472    0.05459 -14.559   <2e-16 ***
## z.diff.lag -0.09993    0.04107  -2.433   0.0153 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.007468 on 585 degrees of freedom
## Multiple R-squared:  0.4471, Adjusted R-squared:  0.4452
## F-statistic: 236.5 on 2 and 585 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -14.5593
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

```
# splitting into train and test data
train_MSI20 <- Close_Price[1:531] # 2021-01-04 --> 2023-02-09 (531 Obs.)
test_MSI20 <- Close_Price[532:590] # 2023-02-10 -->2023-05-05 (59 Obs.)
```

```
model1 <- arima(train_MSI20, order=c(1,1,1),method="CSS")
library(lmtest)
coeftest(model1)
```

```
##
```

```
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1  0.49937    0.18379  2.7171 0.006585 **
## ma1 -0.38338    0.19092 -2.0081 0.044637 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
calculate_bic <- function(arima_model, data) {
  # Calculate the number of parameters in the model
  k <- length(coef(arima_model))

  # Calculate BIC
  bic <- -2 * logLik(arima_model) + k * log(length(data))

  return(bic)
}
```

```
model1 <- arima(train_MSI20, order=c(1,1,1),method="ML")
coeftest(model1)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1  0.56434    0.20093  2.8087 0.004975 **
## ma1 -0.45181    0.21607 -2.0910 0.036524 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(model1)
```

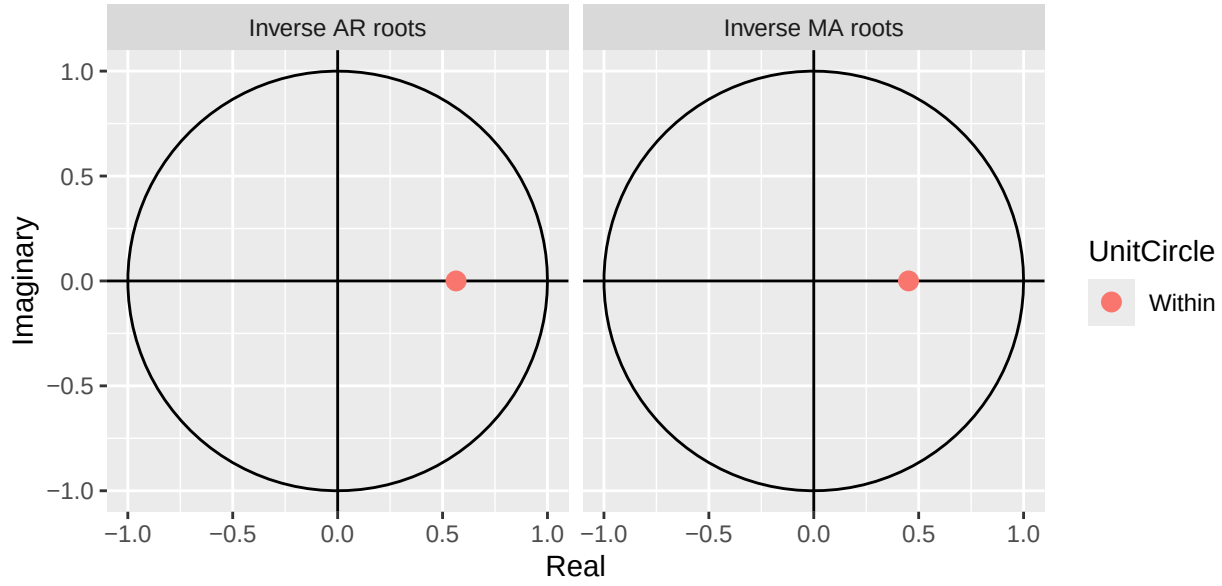
```
##
## Call:
## arima(x = train_MSI20, order = c(1, 1, 1), method = "ML")
##
## Coefficients:
##          ar1      ma1
##       0.5643  -0.4518
## s.e.  0.2009   0.2161
##
## sigma^2 estimated as 52.32:  log likelihood = -1800.74,  aic = 3607.47
##
## Training set error measures:
##                       ME     RMSE      MAE         MPE      MAPE      MASE
## Training set -0.08627774 7.226294 4.952053 -0.01168927 0.5111407 0.9948819
##                      ACF1
## Training set -0.01533004
```

```
bic_model1 <- calculate_bic(model1, train_MSI20)
print(bic_model1)
```

```
## 'log Lik.' 3614.02 (df=3)
```

```
library(knitr)
opts_knit$set(global.par = TRUE)
```

```r
par(mar=c(5,5,0,0)) #it's important to have this in a separate piece
autoplot(model1)
```




```r
model2 <- arima(train_MSI20, order=c(2,1,2),method="ML")
coeftest(model2)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ar1  0.31483    0.19427  1.6206 0.1051081
## ar2 -0.58963    0.20794 -2.8356 0.0045747 **
## ma1 -0.24151    0.16053 -1.5045 0.1324594
## ma2  0.69482    0.20822  3.3369 0.0008471 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(model2)
```

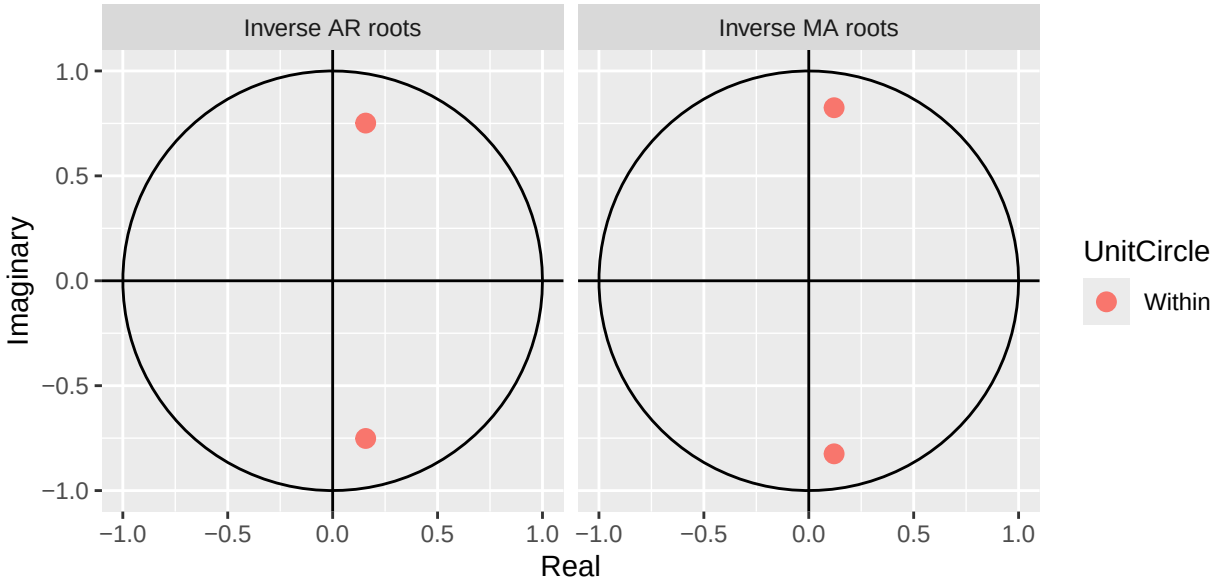
```
##
## Call:
## arima(x = train_MSI20, order = c(2, 1, 2), method = "ML")
##
## Coefficients:
##          ar1      ar2      ma1     ma2
##       0.3148  -0.5896  -0.2415  0.6948
## s.e.  0.1943   0.2079   0.1605  0.2082
```

```
##
## sigma^2 estimated as 51.68:  log likelihood = -1797.56,  aic = 3605.11
##
## Training set error measures:
##                       ME     RMSE      MAE         MPE      MAPE     MASE
## Training set -0.1011205 7.182558 4.923816 -0.01367498 0.5075745 0.989209
##                     ACF1
## Training set 0.02328722
```

```
bic_model2 <- calculate_bic(model2, train_MSI20)
print(bic_model2)
```

```
## 'log Lik.' 3620.214 (df=5)
```

```
par(mar=c(5,5,0,0))
autoplot(model2)
```




```
model3 <- arima(train_MSI20, order=c(2,1,0),method="ML")
coeftest(model3)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1 0.098045   0.043223  2.2683  0.02331 *
## ar2 0.103647   0.043439  2.3861  0.01703 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

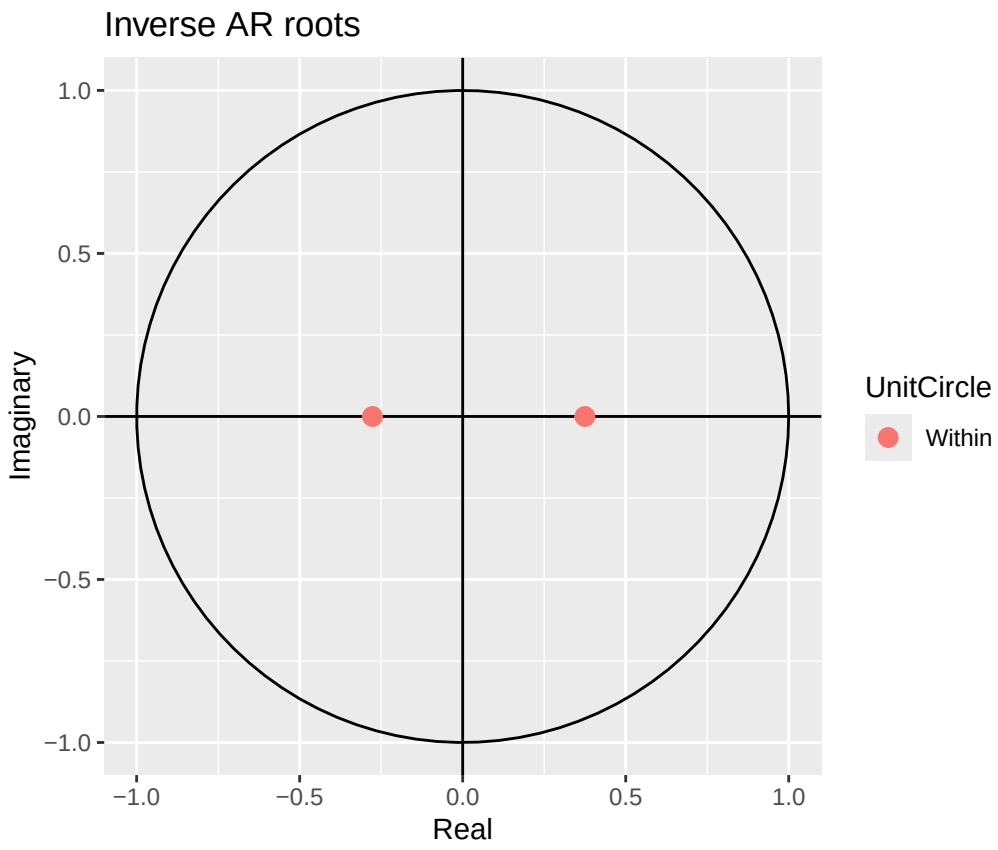
```r
summary(model3)
```

```
##
## Call:
## arima(x = train_MSI20, order = c(2, 1, 0), method = "ML")
##
## Coefficients:
##          ar1     ar2
##       0.0980  0.1036
## s.e.  0.0432  0.0434
##
## sigma^2 estimated as 52.09:  log likelihood = -1799.58,  aic = 3605.15
##
## Training set error measures:
##                       ME     RMSE      MAE         MPE      MAPE      MASE
## Training set -0.08740403 7.210452 4.929905 -0.01181901 0.5087557 0.9904324
##                      ACF1
## Training set 0.0003644483
```

```r
bic_model3 <- calculate_bic(model3, train_MSI20)
print(bic_model3)
```

```
## 'log Lik.' 3611.704 (df=3)
```

```r
par(mar=c(5,5,0,0))
autoplot(model3)
```



```r
model4 <- arima(train_MSI20, order=c(0,1,2),method="ML")
coeftest(model4)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ma1 0.095003   0.043070  2.2058  0.02740 *
## ma2 0.116780   0.044096  2.6483  0.00809 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

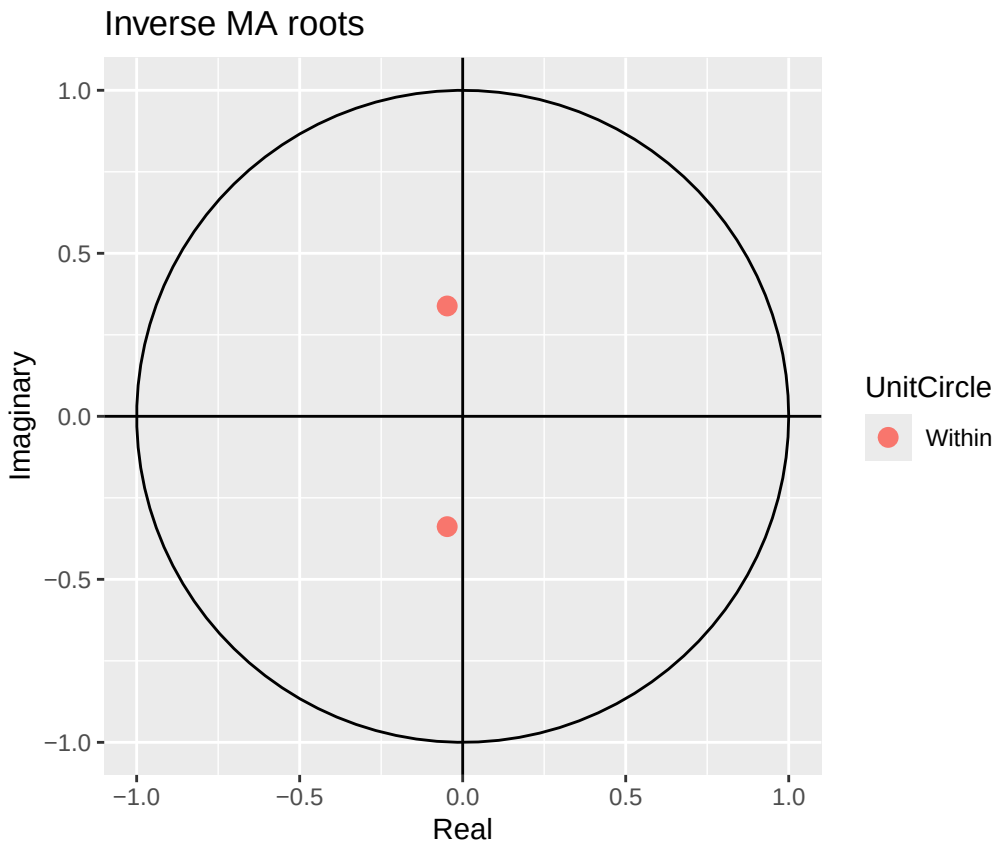
```r
summary(model4)
```

```
##
## Call:
## arima(x = train_MSI20, order = c(0, 1, 2), method = "ML")
##
## Coefficients:
##          ma1     ma2
##       0.0950  0.1168
## s.e.  0.0431  0.0441
##
## sigma^2 estimated as 52.07:  log likelihood = -1799.5,  aic = 3604.99
##
## Training set error measures:
##                       ME     RMSE      MAE         MPE      MAPE      MASE
## Training set -0.09243934 7.209352 4.926385 -0.01250443 0.5083449 0.9897251
##                      ACF1
## Training set 0.002415873
```

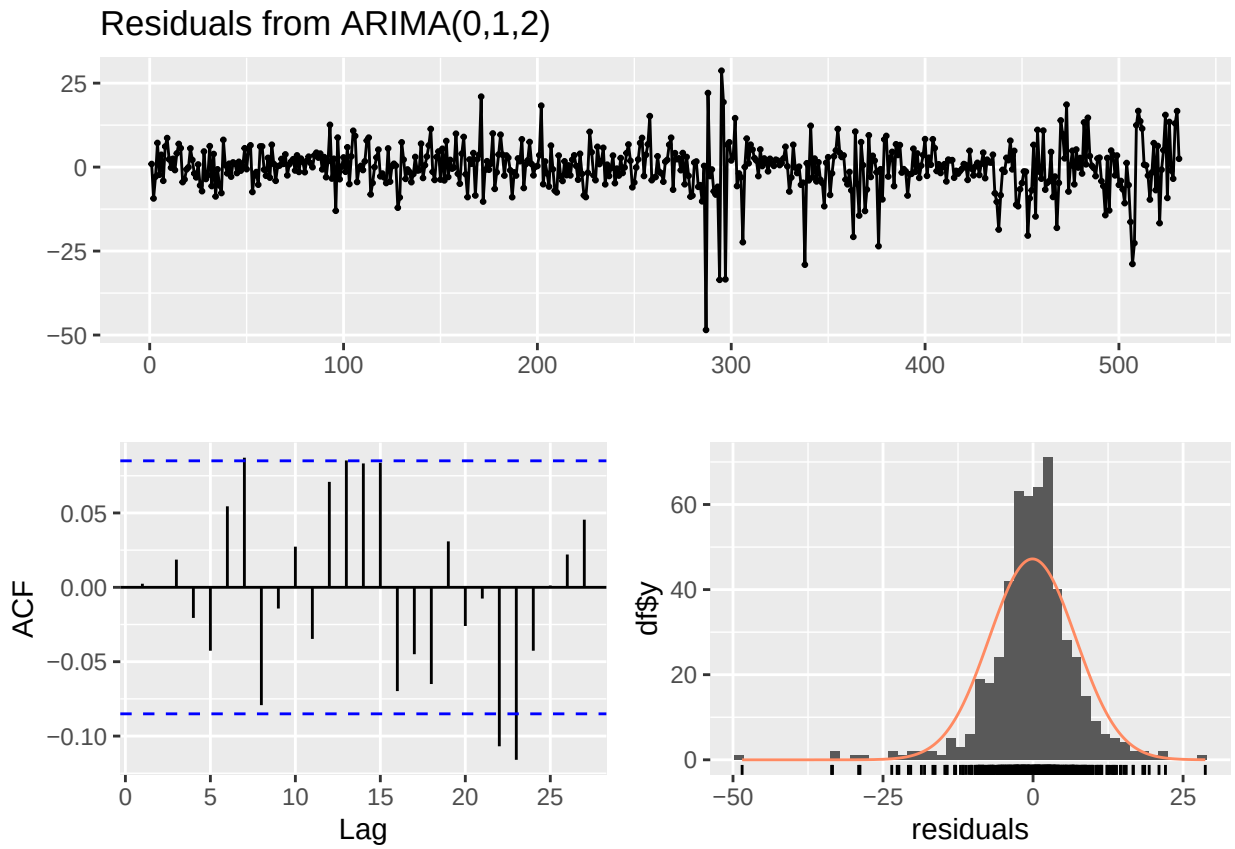
```r
bic_model4 <- calculate_bic(model4, train_MSI20)
print(bic_model4)
```

```
## 'log Lik.' 3611.543 (df=3)
```

```r
par(mar=c(5,5,0,0))
autoplot(model4)
```




```
checkresiduals(model4)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,2)
## Q* = 11.005, df = 8, p-value = 0.2014
##
## Model df: 2.   Total lags used: 10
```

```
resi.ima <- residuals(model4)
tsoutliers(resi.ima)
```

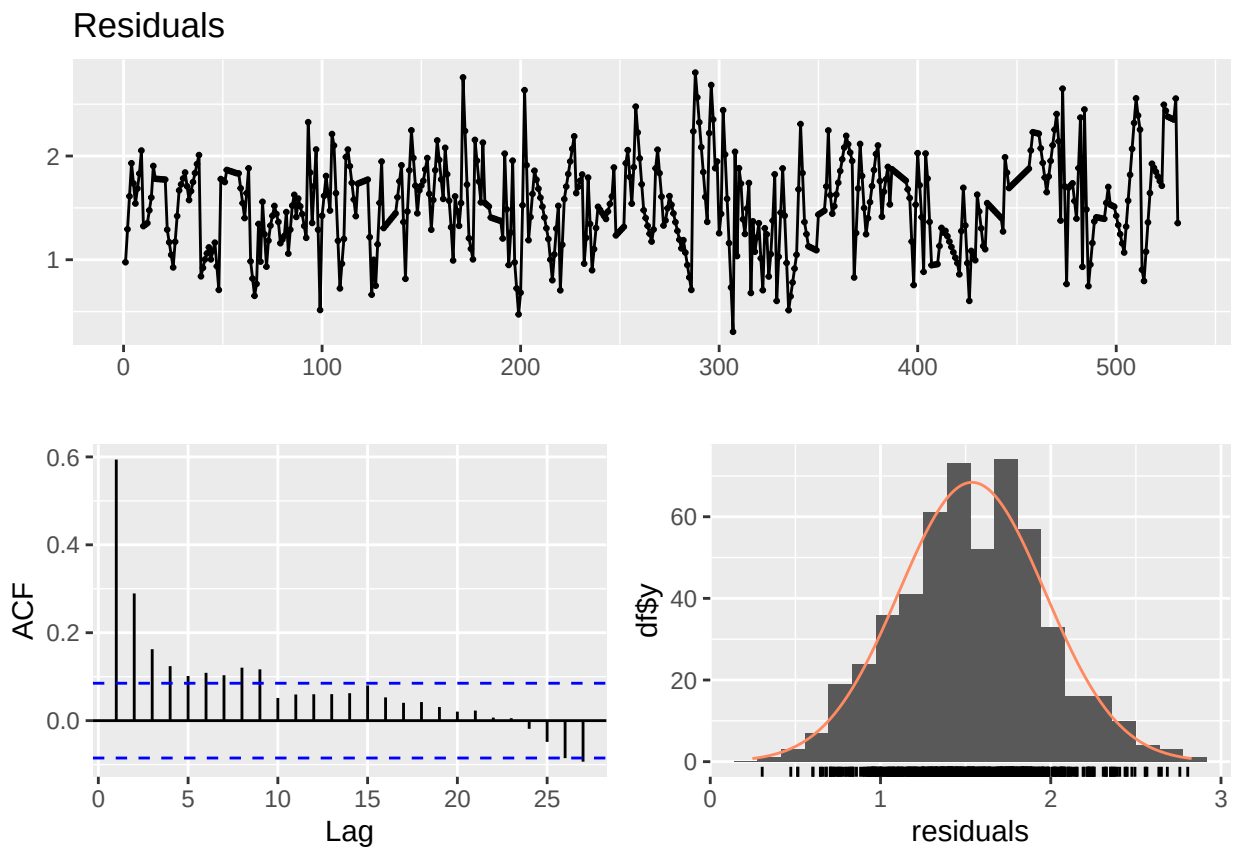
```
## $index
## [1] 287 294 295 297 338 376 507
##
## $replacements
## [1]  11.210394   2.544107  10.948620  13.014059  -1.741477  -1.093739 -19.484213
```

```
resid_ts_clean_ima = tsclean(resi.ima)
```

```
#resid_ts_clean_ima_tra = sqrt(resid_ts_clean_ima)
resid_ts_clean_ima_tra = (resid_ts_clean_ima)^(1/3)
```

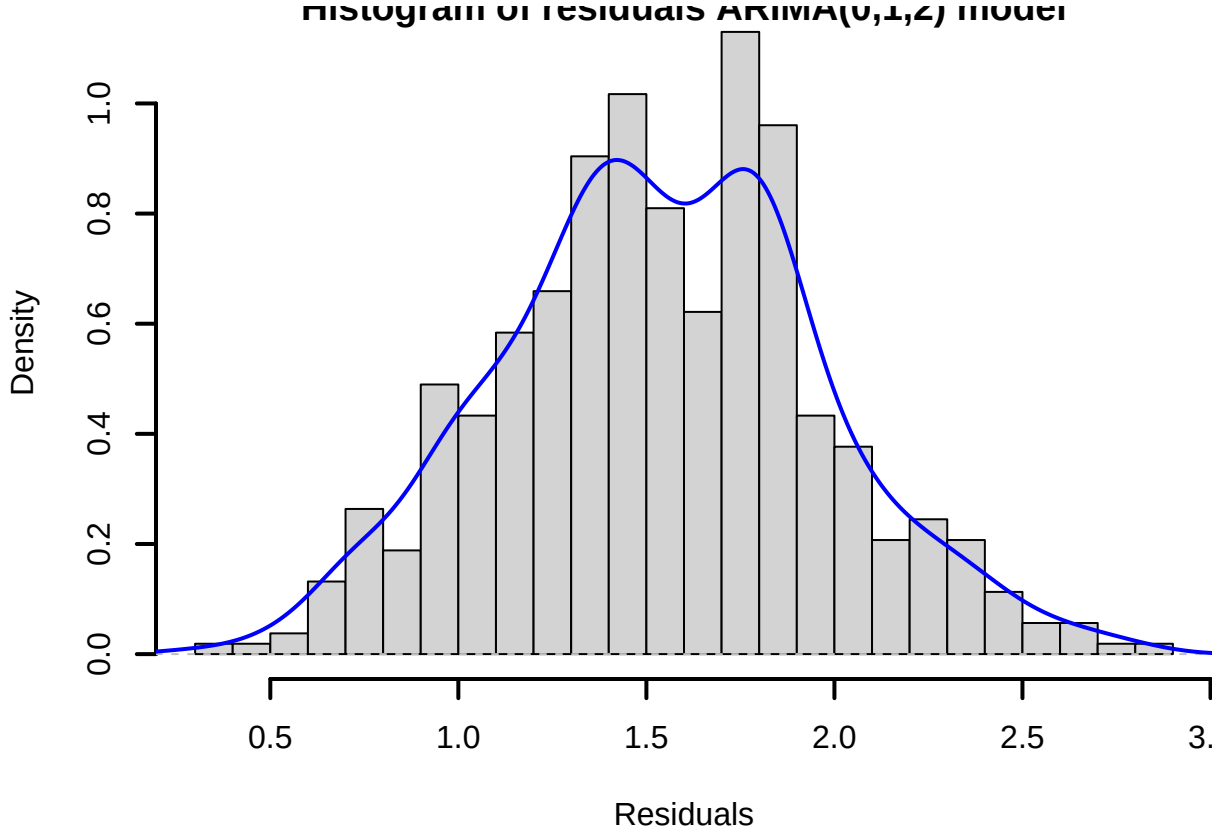
```
# Assuming resid_ts_clean_ima_tra is a time series object

# Impute missing values using linear interpolation
resid_imputed <- na.approx(resid_ts_clean_ima_tra)
```

```
# diagnostic of the series for cleaned residues
checkresiduals(resid_imputed)
```




```
##
##  Ljung-Box test
##
## data:  Residuals
## Q* = 289.95, df = 10, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 10
```

```
hist(resid_imputed, prob=TRUE, 24,main = paste("Histogram of residuals ARIMA(0,1,2) model" ),
     xlab ="Residuals",lwd=2)   # histogram
# Grid below plot
# Vertical grid
abline(v = seq(-0.02, 0.02, 0.0025),
       lty = 2, col = "gray")

# Horizontal grid
abline(h = seq(0, 80, 10),
       lty = 2, col = "gray")
lines(density(resid_imputed),type="l", col="blue", lwd=2)     # smooth it - ?density for details
```




```
library(ggpubr)
```

```
## Warning: le package 'ggpubr' a été compilé avec la version R 4.4.2
```

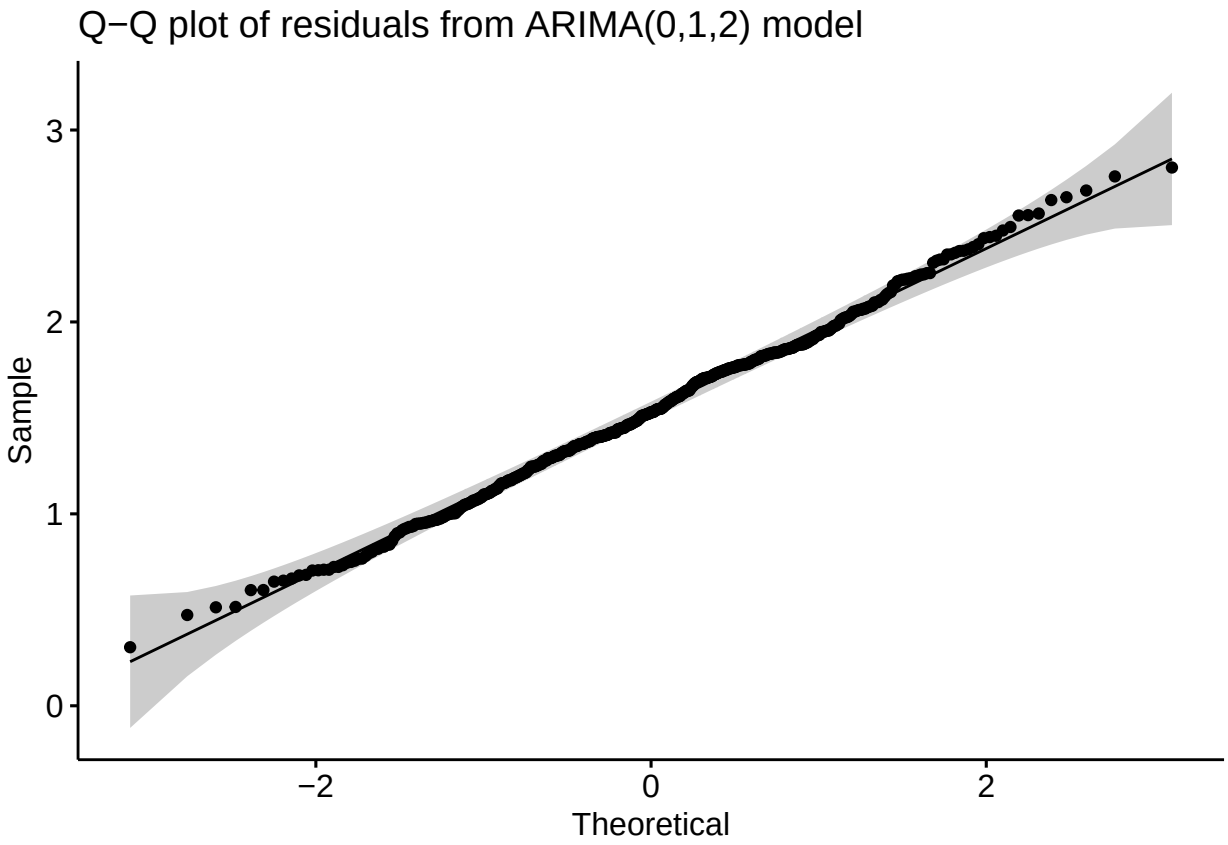
```
##
## Attachement du package : 'ggpubr'
```

```
## L'objet suivant est masqué depuis 'package:forecast':
##
##     gghistogram
```

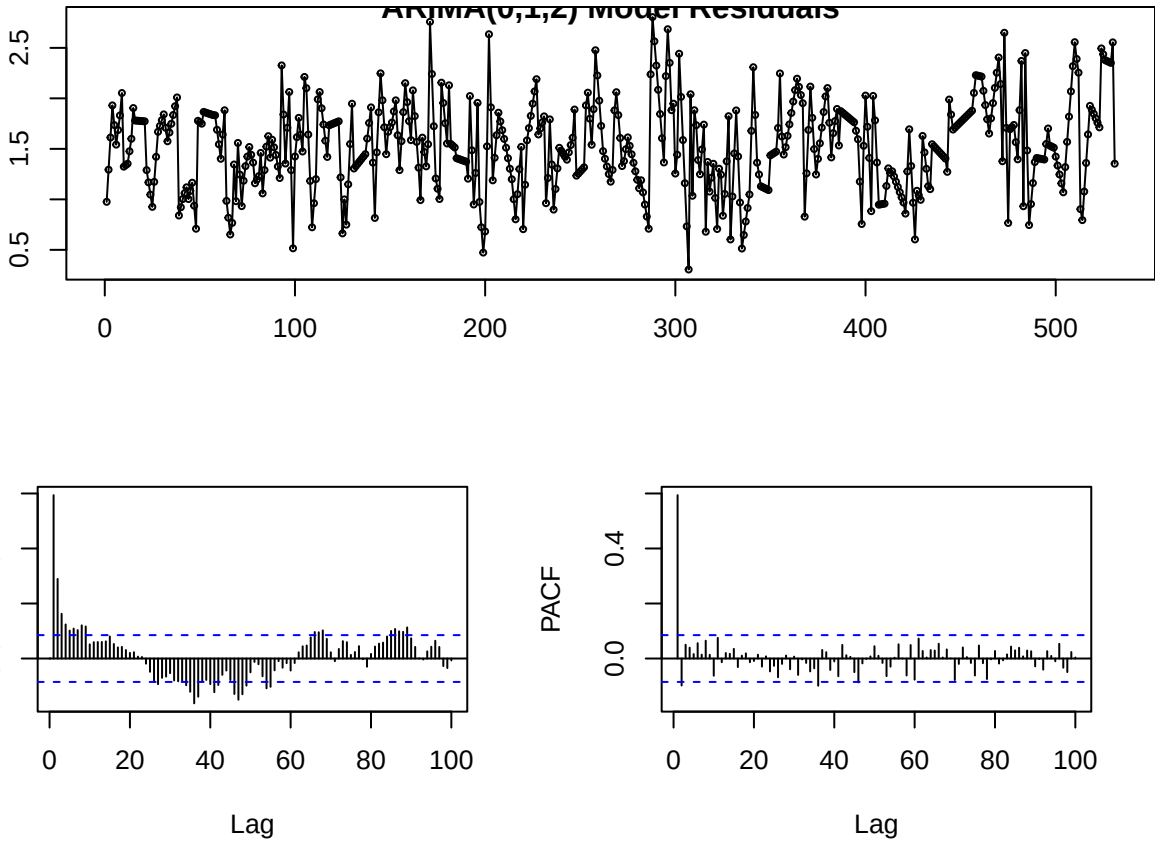
```
ggqqplot(resid_imputed,title = "Q-Q plot of residuals from ARIMA(0,1,2) model", merge = FALSE)
```

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




```
tsdisplay(resid_imputed, lag.max=100, main="ARIMA(0,1,2) Model Residuals")
```




```
#Tests auto-correlation of order greater than 1
Box.test(resid_imputed, lag = 1, type = c("Box-Pierce", "Ljung-Box"), fitdf = 0)
```

```
##
##  Box-Pierce test
##
## data:  resid_imputed
## X-squared = 187.32, df = 1, p-value < 2.2e-16
```

```
# Test of normality
# Perform the Jarque-Bera test on the imputed time series
result_test_jb <- jarque.bera.test(resid_imputed)
print(result_test_jb)
```

```
##
##  Jarque Bera Test
##
## data:  resid_imputed
## X-squared = 0.75552, df = 2, p-value = 0.6854
```

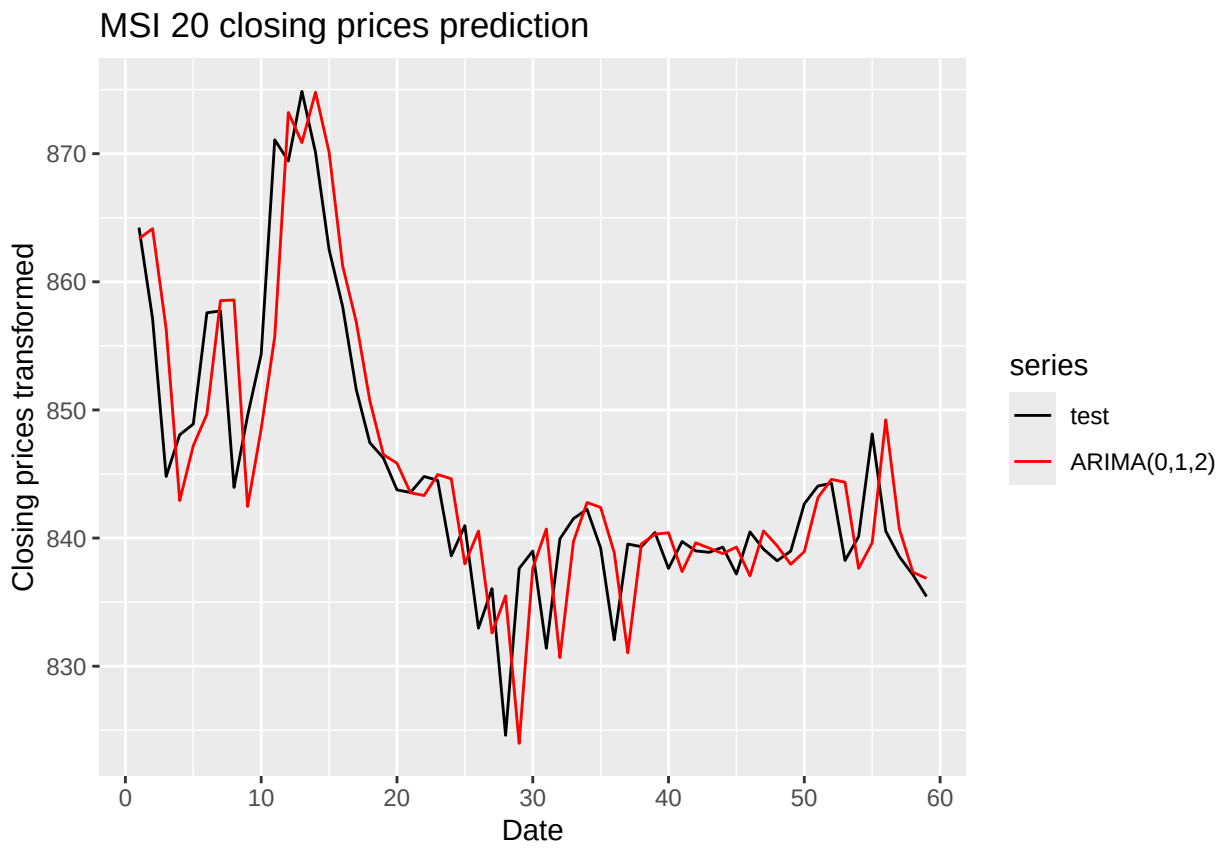
```
# Perform the Shapiro-Wilk test
resultat_test_sh <- shapiro.test(resid_imputed)

# Show test results
print(resultat_test_sh)
```

```
##
##  Shapiro-Wilk normality test
```

```
##
## data:  resid_imputed
## W = 0.99693, p-value = 0.4171
```

```
# Load package
library(nortest)
forecast_test <- Arima(test_MSI20,model=model4)$fitted
autoplot(ts(test_MSI20), series = 'test',xlab ="Date",ylab="Closing prices transformed") +
autolayer(forecast_test, series = 'ARIMA(0,1,2)',PI =FALSE) +
ggtitle("MSI 20 closing prices prediction")+
scale_colour_manual(values = c('test'='black', 'ARIMA(0,1,2)'='red'),

breaks = c('test', 'ARIMA(0,1,2)'))
```

```
## Warning in ggplot2::geom_line(ggplot2::aes(x = .data[["timeVal"]], y =
## .data[["seriesVal"]], : Ignoring unknown parameters: `PI`
```




```
## Create a time series object
## create the zoo object as before
forecast_ts <- zoo(forecast_test , dailyMSI20[532:590,1]) ### 2023-02-10
#to 2023-05-05 (59 Obs.)

head(forecast_ts)
```

```
## 2023-02-10 2023-02-13 2023-02-14 2023-02-15 2023-02-16 2023-02-17
##   863.3658   864.1510   856.3266   842.9247   847.2119   849.6599
```

```r
tail(forecast_ts)
```

```
## 2023-04-27 2023-04-28 2023-05-02 2023-05-03 2023-05-04 2023-05-05
##   837.6344   839.6428   849.2266   840.7059   837.3308   836.8592
```

```r
forecast_ts <- ts(forecast_test,
           start = c(2023, as.numeric(format(dailyMSI20[532:590,1], "%j"))))
head(forecast_ts)
```

```
## Time Series:
## Start = 2063
## End = 2068
## Frequency = 1
## [1] 863.3658 864.1510 856.3266 842.9247 847.2119 849.6599
```

```r
# merge into multivariate time series
ts.merge <- merge(test_MSI20, as.zoo(forecast_ts))

colnames(ts.merge) <- c("ts_test", "ts_forecast")
#as.ts(ts.merge)
head(ts.merge)
```

```
##            ts_test ts_forecast
## 2023-02-10  864.23    863.3658
## 2023-02-13  857.13    864.1510
## 2023-02-14  844.81    856.3266
## 2023-02-15  848.06    842.9247
## 2023-02-16  848.90    847.2119
## 2023-02-17  857.58    849.6599
```

```r
tail(ts.merge)
```

```
##            ts_test ts_forecast
## 2023-04-27  840.12    837.6344
## 2023-04-28  848.13    839.6428
## 2023-05-02  840.54    849.2266
## 2023-05-03  838.55    840.7059
## 2023-05-04  837.13    837.3308
## 2023-05-05  835.44    836.8592
```

```r
forecast_model4 <- forecast(model4, h = 59)
accuracy(forecast_model4,test_MSI20)
```

```
##                        ME      RMSE       MAE         MPE      MAPE      MASE
## Training set  -0.09243934  7.209352  4.926385 -0.01250443 0.5083449 0.9897251
## Test set     -23.28370316 25.474242 23.718416 -2.77102972 2.8208400 4.7650993
##                     ACF1
## Training set 0.002415873
## Test set              NA
```