ARIMA for Stock Prediction

2024-03-21

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggthemes)
library(quantmod)
```

```
## Loading required package: xts
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
##
## ######################### Warning from 'xts' package ##########################
## #                                                                             #
## # The dplyr lag() function breaks how base R's lag() function is supposed to  #
## # work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or       #
## # source() into this session won't work correctly.                            #
## #                                                                             #
## # Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #
## # conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop           #
## # dplyr from breaking base R's lag() function.                                #
## #                                                                             #
## # Code in packages is not affected. It's protected by R's namespace mechanism #
## # Set ‘options(xts.warn_dplyr_breaks_lag = FALSE)‘ to suppress this warning.  #
## #                                                                             #
## ###############################################################################
##
## Attaching package: 'xts'
##
```

```
## The following objects are masked from ’package:dplyr’:
##
##     first, last
##
## Loading required package: TTR
## Registered S3 method overwritten by ’quantmod’:
##   method            from
##   as.zoo.data.frame zoo
```

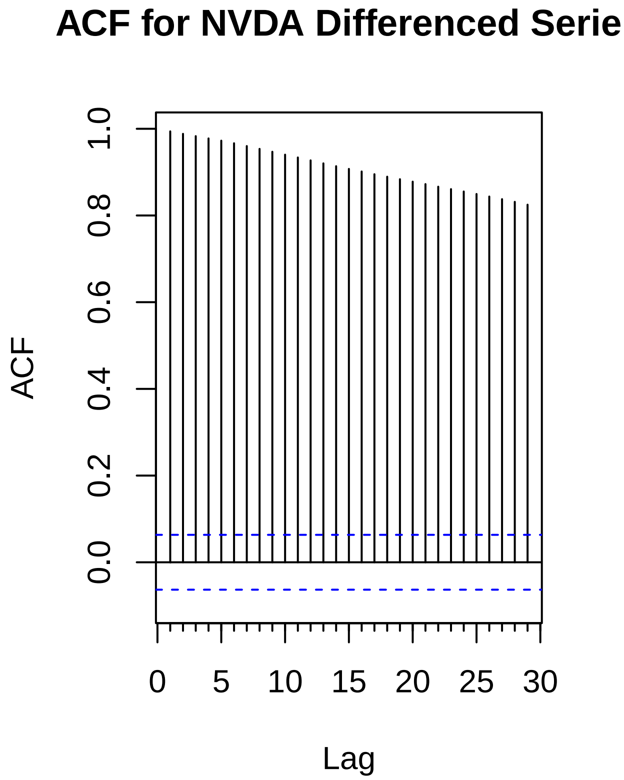
```r
library(tidyquant)
```

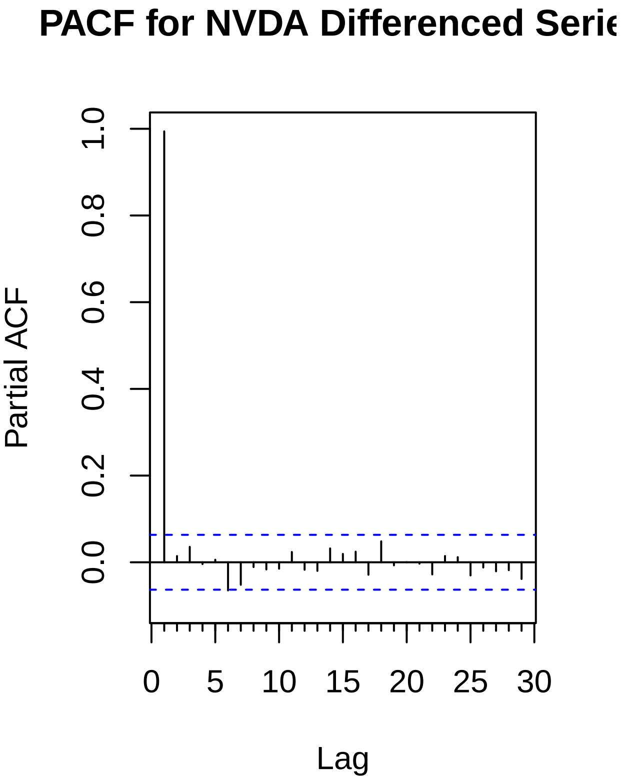
```
## Loading required package: PerformanceAnalytics
##
## Attaching package: ’PerformanceAnalytics’
##
## The following object is masked from ’package:graphics’:
##
##     legend
```

```r
library(tseries)
library(timeSeries)
```

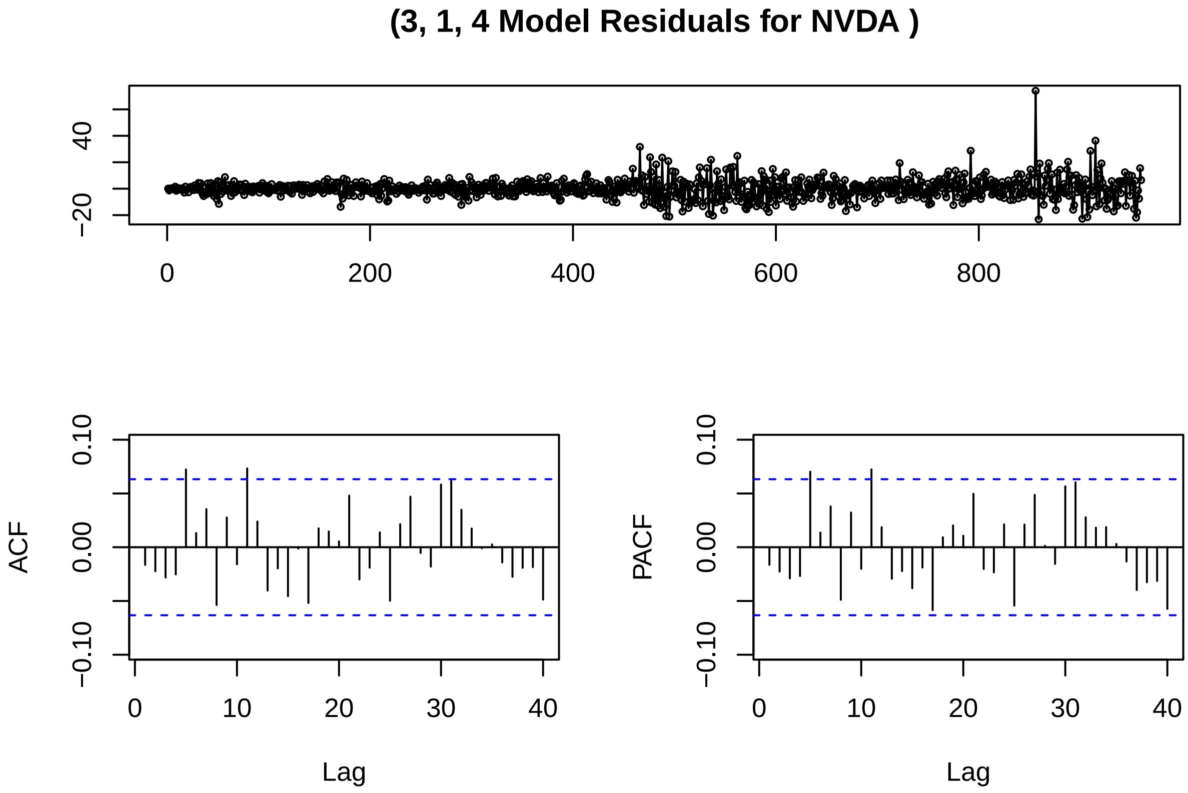
```
## Loading required package: timeDate
##
## Attaching package: ’timeDate’
##
## The following objects are masked from ’package:PerformanceAnalytics’:
##
##     kurtosis, skewness
##
##
## Attaching package: ’timeSeries’
##
## The following object is masked from ’package:zoo’:
##
##     time<-
```

```r
library(forecast)
library(xts)

analyze_stock <- function(ticker) {
  # get symbol from yahoo finance
  getSymbols(ticker, from = "2020-01-01", to = "2023-10-25")
  stock_data <- get(ticker)

  stock_Close <- stock_data[,4]

  plot(stock_Close)

  # graph the ACF (Q) and PACF (P) for custom arimas
  par(mfrow = c(1, 2))
  Acf(stock_Close, main = paste("ACF for", ticker, "Differenced Series"))
  Pacf(stock_Close, main = paste("PACF for", ticker, "Differenced Series"))

  # ADF test for p-value
  print(adf.test(stock_Close)) # p-value should be below 0.01
  auto.arima(stock_Close, seasonal = FALSE)

  FitA = auto.arima(stock_Close, seasonal = FALSE)
  tsdisplay(residuals(FitA), lag.max = 40,
            main = paste("(3, 1, 4 Model Residuals for", ticker, ")"))

  FitB = arima(stock_Close, order = c(1, 2, 4))
  tsdisplay(residuals(FitB), lag.max = 40,
            main = paste("(1, 2, 4 Model Residuals for", ticker, ")"))

  FitC = arima(stock_Close, order = c(5, 1, 4))
  tsdisplay(residuals(FitC), lag.max = 40,
            main = paste("(5, 1, 4 Model Residuals for", ticker, ")"))

  FitD = arima(stock_Close, order = c(1, 1, 1))
  tsdisplay(residuals(FitD), lag.max = 40,
            main = paste("(1, 1, 1 Model Residuals for", ticker, ")"))

  # plot forecasts
  par(mfrow = c(2,2))
  term <- 100

  fcast1 <- forecast(FitA, h = term)
  plot(fcast1)

  fcast2 <- forecast(FitB, h = term)
  plot(fcast2)

  fcast3 <- forecast(FitC, h = term)
  plot(fcast3)

  fcast4 <- forecast(FitD, h = term)
  plot(fcast4)

  # accuracy
  print(accuracy(fcast1))
  print(accuracy(fcast2))
  print(accuracy(fcast3))
  print(accuracy(fcast4))
}

# Example usage
analyze_stock("NVDA")
```

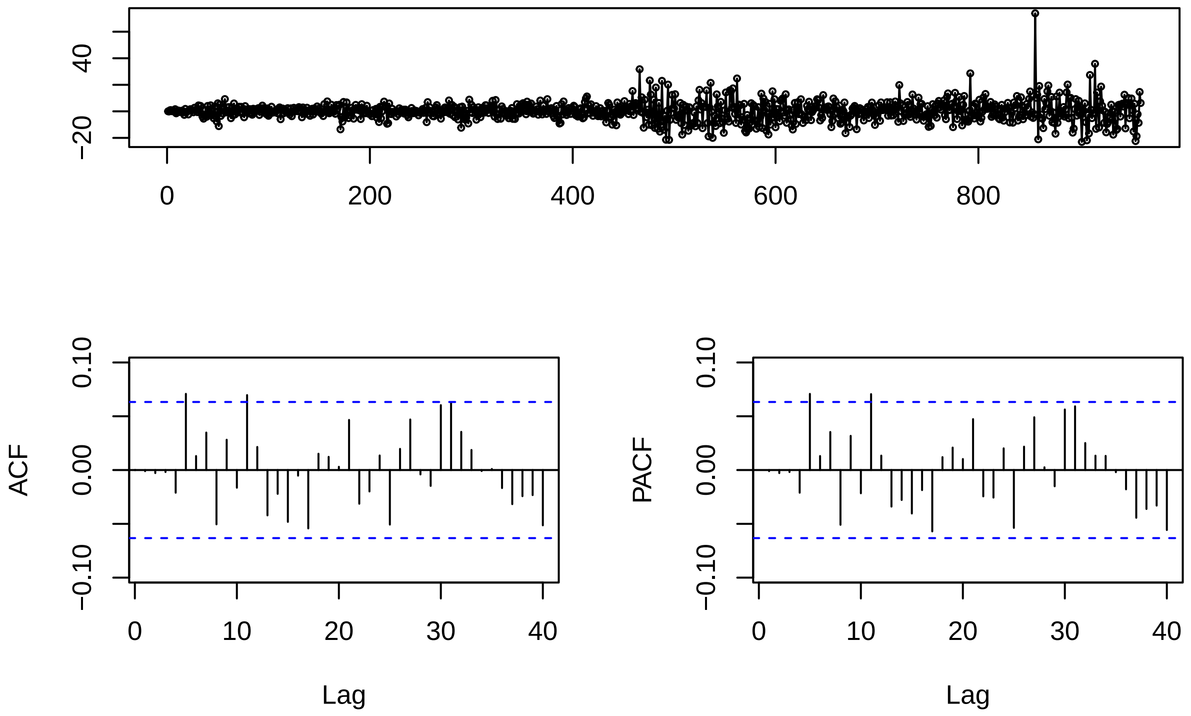

ACF for NVDA Differenced Serie


PACF for NVDA Differenced Serie

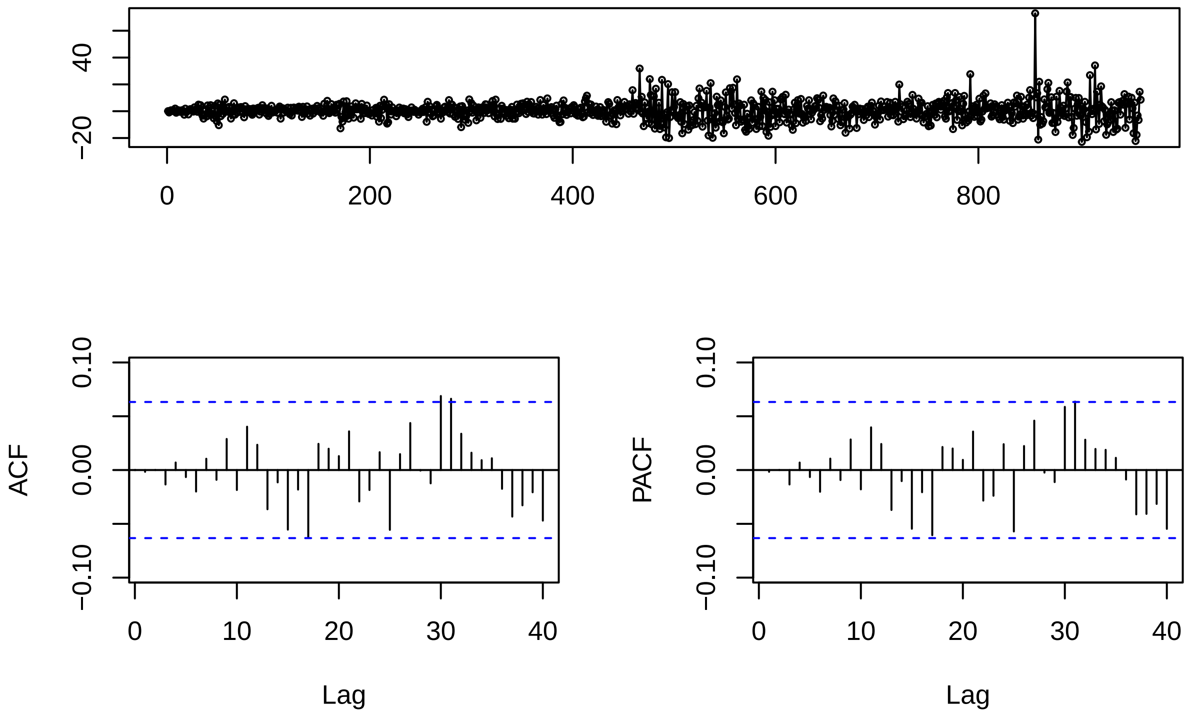
```
##
##  Augmented Dickey-Fuller Test
##
## data:  stock_Close
## Dickey-Fuller = -1.3806, Lag order = 9, p-value = 0.8405
## alternative hypothesis: stationary
```

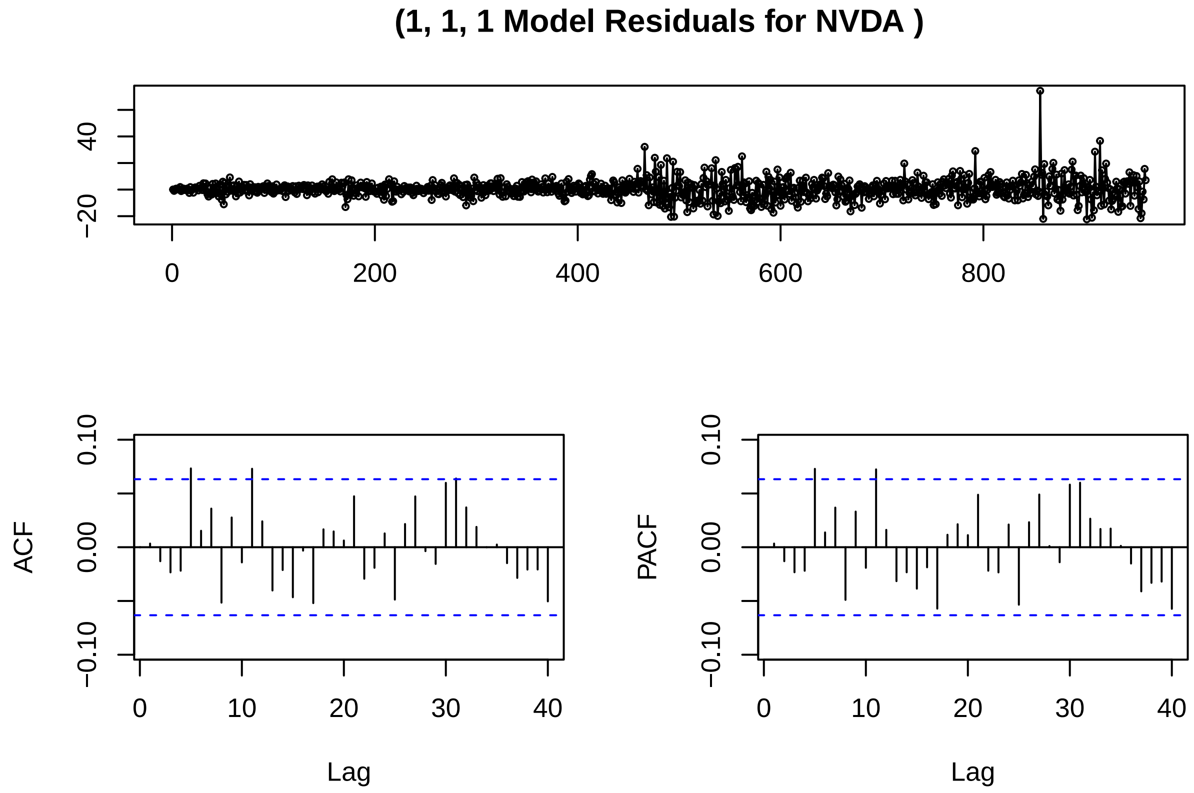

(3, 1, 4 Model Residuals for NVDA)
40
−20
0
200
400
600
800
ACF
0.10
0.00
−0.10
0
10
20
30
40
Lag
PACF
0.10
0.00
−0.10
0
10
20
30
40
Lag

(1, 2, 4 Model Residuals for NVDA)

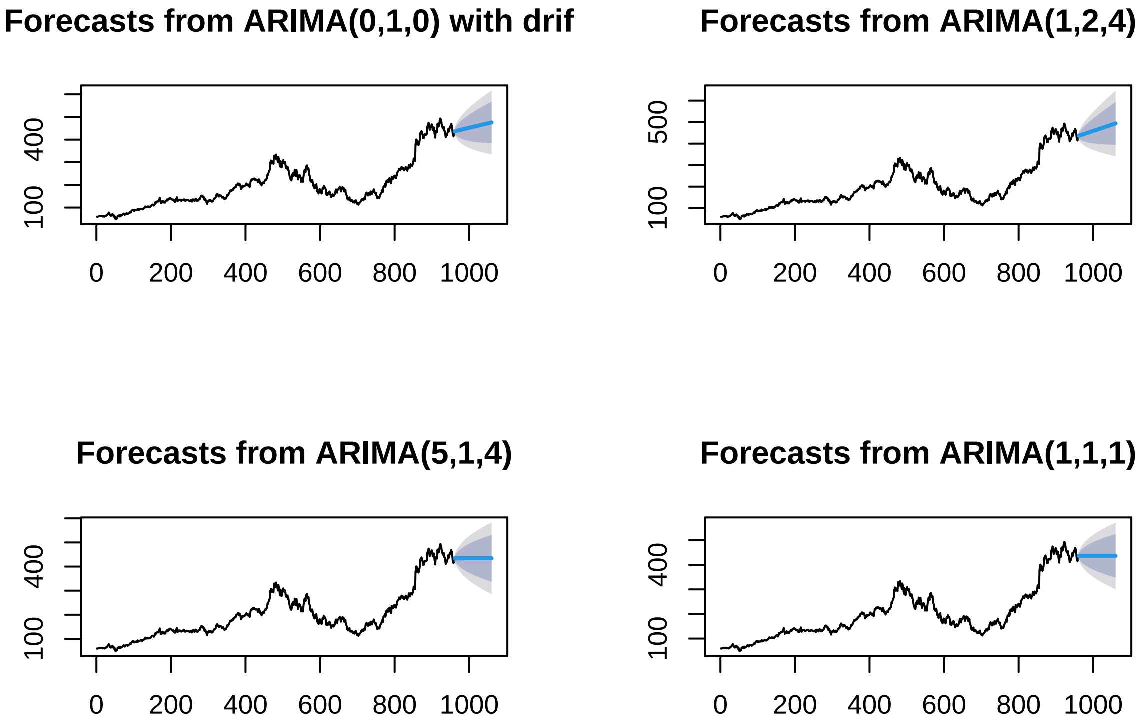


(5, 1, 4 Model Residuals for NVDA)





(1, 1, 1 Model Residuals for NVDA)



```
##                         ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 6.206741e-05 7.172768 4.837663 -0.1104002 2.537233 0.9950521
##                      ACF1
## Training set -0.01639904
##                      ME     RMSE      MAE         MPE     MAPE      MASE
## Training set 0.1111764 7.173812 4.847163 0.001102923 2.539908 0.9970062
##                       ACF1
## Training set -0.001019897
##                      ME     RMSE      MAE       MPE    MAPE      MASE
## Training set 0.3651989 7.136801 4.853468 0.1381096 2.54447 0.9983031
##                       ACF1
## Training set -0.001648946
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.4080743 7.181527 4.858071 0.1538375 2.547597 0.9992499
##                      ACF1
## Training set 0.003407262
```