FA542 - Midterm Exam

I pledge my honor that I have abided by the Stevens Honor System.

Sid Bhatia

2023-10-26

Problem 1 (20pt)

Let p_t be the the log price of an asset at time t. Assume that the log price follows the model:

$$p_t = 0.001 + p_{t-1} + a_t,\ a_t \sim \mathcal{N}(\mu = 0, \sigma^2 = 0.16)$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes normal distribution with mean μ and variance σ^2.

Assume further that $p_{200} = 4.551$.

a. *Compute the 95% interval forecast for p_{201} at the forecast origin $t = 200$.*

For $t = 201$:

$$p_{201} = 0.001 + 4.551 + a_{201}$$

If we take the conditional expectation of this, we have end up with the following:

$$\mathbb{E}[p_{201}|\mathcal{F}] = 4.552 + \mathbb{E}[a_{201}] = 4.552$$

.

The variance of p_{201} is the variance of a_{201}, therefore the standard deviation is $\sigma = \sqrt{0.16} = 0.4$.

As such, the 95% CI is:

$$p_{201} \pm 1.96 * \sigma$$

```
set.seed(100)

# Establish coefficients for AR(1) model.
phi_0 <- 0.001
phi_1 <- 1
a_t <- 0.16

p_200 <- 4.551

# Compute conditional expectation forecast.
p_201 <- phi_0 + p_200

p_201
```

```
## [1] 4.552
```

```
# Compute forecast incorporating white noise.
forecast_1_step <- phi_0 + p_200 + rnorm(1, mean = 0, sd = sqrt(a_t))
forecast_1_step
```

```
## [1] 4.351123
```

```
# Compute z-score for \alpha = 0.05
z_score <- qnorm(0.975)

# Create 95% for forecast.
lower_bound <- p_201 - (z_score * sqrt(a_t))
upper_bound <- p_201 + (z_score * sqrt(a_t))

confidence_interval <- c(lower_bound, upper_bound)

confidence_interval
```

```
## [1] 3.768014 5.335986
```

b. *Compute the 2-step ahead point forecast and its standard error for p_{202} at the forecast origin $t = 200$.*

$$p_{202} = 0.001 + p_{201} + a_t = 0.001 + 4.552 + a_{202}$$

.

If we take the conditional expectation of this, we have end up with the following:

$$\mathbb{E}[p_{202}|\mathcal{F}] = 4.553 + \mathbb{E}[a_{202}] = 4.553$$

.

The standard deviation of the forecast error at time $n + m$ is:

$$SE(x^n_{n+m} - x_{n+m}) = \sqrt{\hat{\sigma}^2_w \sum_{j=0}^{m-1} \phi_j^2}$$

When forecasting $m = 1$ time past the end of the series, the SE of the forecast is:

$$SE(x^n_{n+1} - x_{n+1}) = \sqrt{\hat{\sigma}^2_w(1)}$$

When forecasting $m = 2$ time past the end of the series, the SE of the forecast is:

$$SE(x^n_{n+2} - x_{n+2}) = \sqrt{\hat{\sigma}^2_w(1 + \phi_1^2)}$$

In this case, $\phi_1 = 1$.

Therefore, the SE of the forecast is:

$$\sqrt{0.16 * (1 + 1)} = \sqrt{0.32} = 0.5656854$$

.

```r
# Compute conditional expectation forecast.
p_202 <- phi_0 + p_201
p_202
```

```
## [1] 4.553
```

```r
# Compute forecast incorporating white noise.
forecast_2_step <- phi_0 + p_201 + rnorm(1, mean = 0, sd = sqrt(a_t))
forecast_2_step
```

```
## [1] 4.605612
```

```r
# Compute forecast step error.
forecast_2_step_SE <- sqrt(0.16 * 2)
forecast_2_step_SE
```

```
## [1] 0.5656854
```

c. *What is the 100-step ahead forecast for p_{300} at the forecast origin $t = 200$?*

$$p_{300} = p_{200} + 100 * phi_0 + a_t = 4.551 + 0.1 + a_t$$

If we take the conditional expectation of this, we have end up with the following:

$$\mathbb{E}[p_{300}|\mathcal{F}] = 4.651 + \mathbb{E}[a_{202}] = 4.651$$

.

```r
# Compute forecast incorporating white noise.
forecast_100_step <- 100 * phi_0 + p_200 + rnorm(1, mean = 0, sd = sqrt(a_t))
forecast_100_step
```

```
## [1] 4.619433
```

Problem 2 (20pt)

Suppose that the quarterly growth rates r_t of an economy follows the model:

$$r_t = 0.006 + 0.168r_{t-1} + 0.338r_{t-2} - 0.189r_{t-3} + a_t,\ a_t \sim \mathcal{N}(\mu = 0, \sigma^2 = 0.0016)$$

a. *What is the expected growth rate of r_t?*

$$\mathbb{E}[r_t] = \frac{\phi_0}{1 - \phi_1 - \phi_2 - \phi_3}$$

```r
# Establish coefficents of AR(3) model.
phi_0 <- 0.006
phi_1 <- 0.168
phi_2 <- 0.338
phi_3 <- -0.189

# Calculated expected growth or \mu.
E_rt <- phi_0 / (1 - phi_1 - phi_2 - phi_3)

E_rt
```

```
## [1] 0.008784773
```

```
E_rt * 100
```

```
## [1] 0.8784773
```

As such, the expected growth rate of r_t is 0.878%.

b. *Does the model imply existence of business cycles? Why?*

$$\phi(B) = 1 - 0.168B - 0.338B^2 + 0.189B^3$$

where B is the backshift or lag operator.

```
# Coefficients of the polynomial in ascending order
coefficients <- c(1, -0.168, -0.338, 0.189)

# Compute the roots
roots <- polyroot(coefficients)

print(roots)
```

```
## [1]  1.608257+1.057496i -1.428154+0.000000i  1.608257-1.057496i
```

Since the roots of the charateristic polynommial of the AR model are :

$$1.608257 \pm 1.057496i, 1.428154$$

This implies that are are oscillatory or cyclical behavior in the series because of the complex roots.

The magnitude or modulus of a complex number is $\sqrt{a^2 + b^2}$. In this case, it is the following:

```
# Compute modulus of complex root.
a <- 1.608257
b <- 1.057496

sqrt(a^2 + b^2)
```

```
## [1] 1.924783
```

It is approximately 1.92, and since it is greater than 1, it indicates that the business cycles will not only persist but also grow over time.

c. *What is the average length of business cycles of the economy, if any?*

In order to find the average length of the business cycle introduced by a complex root, we can examine the argument (or angle) of the complex number.

For a complex number $a + bi$, the angle θ in radians is calculated as:

$$\theta = \arctan(b, a)$$

The length of the business cycle in discrete time periods (like quarters) can be determined as:

$$\text{Length} = \frac{2\pi}{\theta}$$

```r
# Calculate angle in radians.
theta <- atan2(b, a)

# Calculate the length of the business cycle in quarters.
cycle_length <- 2 * pi / theta

cycle_length
```

```
## [1] 10.80219
```

```r
# Convert quarters to years.
cycle_length / 4
```

```
## [1] 2.700547
```

As such, the average length of business cycles of the economy is 10.8 quarters or 2.7 years.

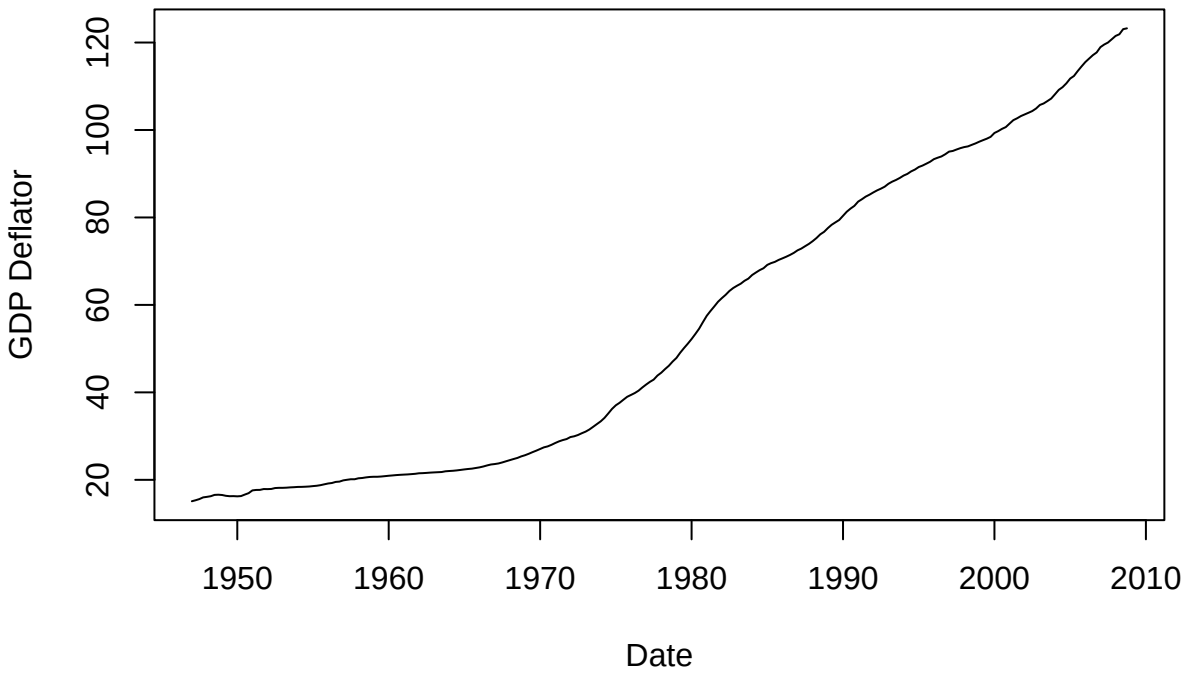
Problem 3 (20pt)

The quarterly gross domestic product implicit price deflator is often used as a measure of inflation. The file **q-gdpdef.txt** contains the data for the United States from the first quarter of 1947 to the last quarter of 2008. Data format is year, month, day, and deflator. The data are seasonally adjusted and equal to 100 for year 2000.

```r
# Establish the directory for data.
data_directory <- "C:/Users/sbhatia2/My Drive/University/Academics/Semester V/FA542 - Time Series with

# Load in the dataset.
gdp_deflator <- read.table(paste(data_directory, 'q-gdpdef.txt', sep=""), header = T)

# Convert the year, month, and day into a Date object
gdp_deflator$date <- as.Date(with(gdp_deflator, paste(year, mom, day, sep="-")), format="%Y-%m-%d")

# Move 'date' column to the first position and remove nonessential columns.
gdp_deflator <- gdp_deflator[, c("date", "gdpdef")]

head(gdp_deflator)
```

```
##         date gdpdef
## 1 1947-01-01 15.105
## 2 1947-04-01 15.329
## 3 1947-07-01 15.597
## 4 1947-10-01 15.989
## 5 1948-01-01 16.111
## 6 1948-04-01 16.254
```

```r
# Plot the data
plot(gdp_deflator$date, gdp_deflator$gdpdef, type="l", xlab="Date", ylab="GDP Deflator")
```



a. *Build an ARIMA model for the series and check the validity of the fitted model.*

```
library(tseries)
```

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

```
adf.test(gdp_deflator$gdpdef)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  gdp_deflator$gdpdef
## Dickey-Fuller = -2.6022, Lag order = 6, p-value = 0.3223
## alternative hypothesis: stationary
```

Since the p-value is greater than 0.05, we cannot reject the null hypothesis that the data is nonstationary at the 5% significance level. As such, we must difference it.

```
adf.test(diff(gdp_deflator$gdpdef))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(gdp_deflator$gdpdef)
## Dickey-Fuller = -3.0652, Lag order = 6, p-value = 0.1275
## alternative hypothesis: stationary
```

```r
adf.test(diff(diff(gdp_deflator$gdpdef)))
```

```
## Warning in adf.test(diff(diff(gdp_deflator$gdpdef))): p-value smaller than
## printed p-value
```

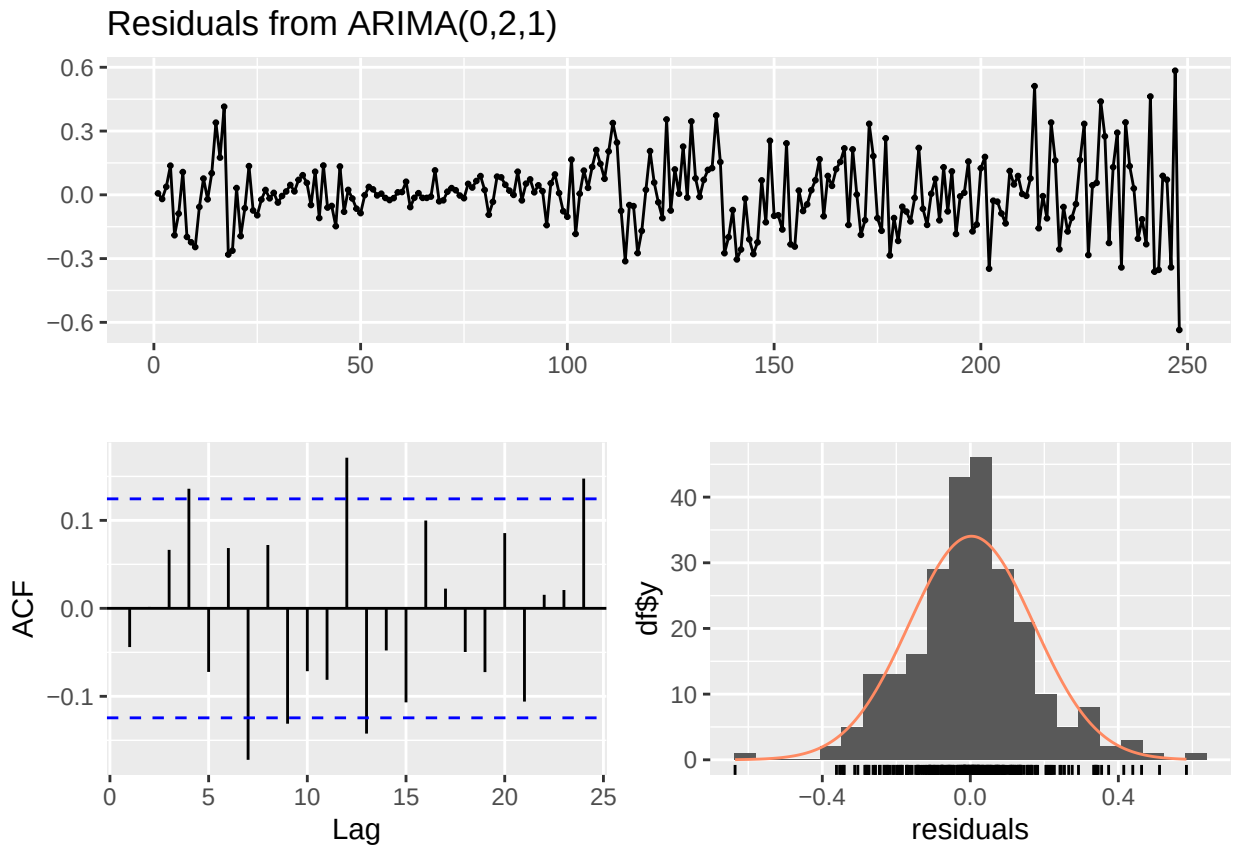
```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(diff(gdp_deflator$gdpdef))
## Dickey-Fuller = -6.9412, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

After differencing twice, we can reject the null hypothesis of a unit root, meaning the data is stationary.

```r
library(forecast)

# Create ARIMA model based on data.
arima_model <- auto.arima(gdp_deflator$gdpdef)

summary(arima_model)
```

```
## Series: gdp_deflator$gdpdef
## ARIMA(0,2,1)
##
## Coefficients:
##           ma1
##       -0.5862
## s.e.   0.0486
##
## sigma^2 = 0.02873:  log likelihood = 87.86
## AIC=-171.72   AICc=-171.67   BIC=-164.7
##
## Training set error measures:
##                       ME      RMSE       MAE        MPE     MAPE      MASE
## Training set 0.003095975 0.1684768 0.1243017 0.01220345 0.285456 0.2817907
##                     ACF1
## Training set -0.04402516
```

```r
# Check the residuals to validate the model.
checkresiduals(arima_model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,2,1)
## Q* = 23.592, df = 9, p-value = 0.004995
##
## Model df: 1.   Total lags used: 10
```

```
# Conduct the Ljung-Box test on the residuals
ljung_box_result <- Box.test(arima_model$residuals, lag=log(length(arima_model$residuals)))

ljung_box_result
```

```
##
##  Box-Pierce test
##
## data:  arima_model$residuals
## X-squared = 7.4554, df = 5.5134, p-value = 0.2343
```

The fitted model is ARIMA$(p = 0, d = 2, q = 1)$, or a MA(2) model that is differenced twice.

Since the p-value for the Ljung-Box test is greater than 0.05, we fail to reject the null hypothesis that the residuals have significant autocorrelations up to the specified lag. As such, the residuals seem to behave like white noise, validating the model.

b. *Use the fitted model to predict the inflation for each quarter of 2009.*

```r
forecast_values <- forecast(arima_model, h = 4)
forecast_values
```

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 249       123.8046 123.5874 124.0219 123.4724 124.1369
## 250       124.3653 123.9891 124.7415 123.7900 124.9406
## 251       124.9259 124.3790 125.4729 124.0895 125.7624
## 252       125.4866 124.7543 126.2188 124.3667 126.6065
```

```r
forecast_values$mean
```

```
## Time Series:
## Start = 249
## End = 252
## Frequency = 1
## [1] 123.8046 124.3653 124.9259 125.4866
```

Problem 4 (20pt)

You can use the quantmod package in R to obtain financial data.

a. *Download daily price data for January 1, 2018 through October 23, 2023 of McDonald's Corp (MCD) stock from Yahoo Finance.*

```r
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 ######################################
## # We noticed you have dplyr installed. The dplyr lag() function breaks how    #
## # base R's lag() function is supposed to work, which breaks lag(my_xts).      #
## #                                                                              #
## # If you call library(dplyr) later in this session, then calls to lag(my_xts) #
## # that you enter or source() into this session won't work correctly.           #
## #                                                                              #
## # All package code is unaffected because it is protected by the R namespace   #
## # mechanism.                                                                   #
## #                                                                              #
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning.  #
## #                                                                              #
## # You can 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.                                 #
## ###################################### WARNING ######################################
```

```
## Loading required package: TTR
```

```r
# Retrieve the data.
start_date <- '2018-01-01'
end_date <- '2023-10-23'

getSymbols("MCD", from = start_date, to = end_date)
```

```
## [1] "MCD"
```

```r
mcd_adj_price <- MCD$MCD.Adjusted

head(mcd_adj_price)
```

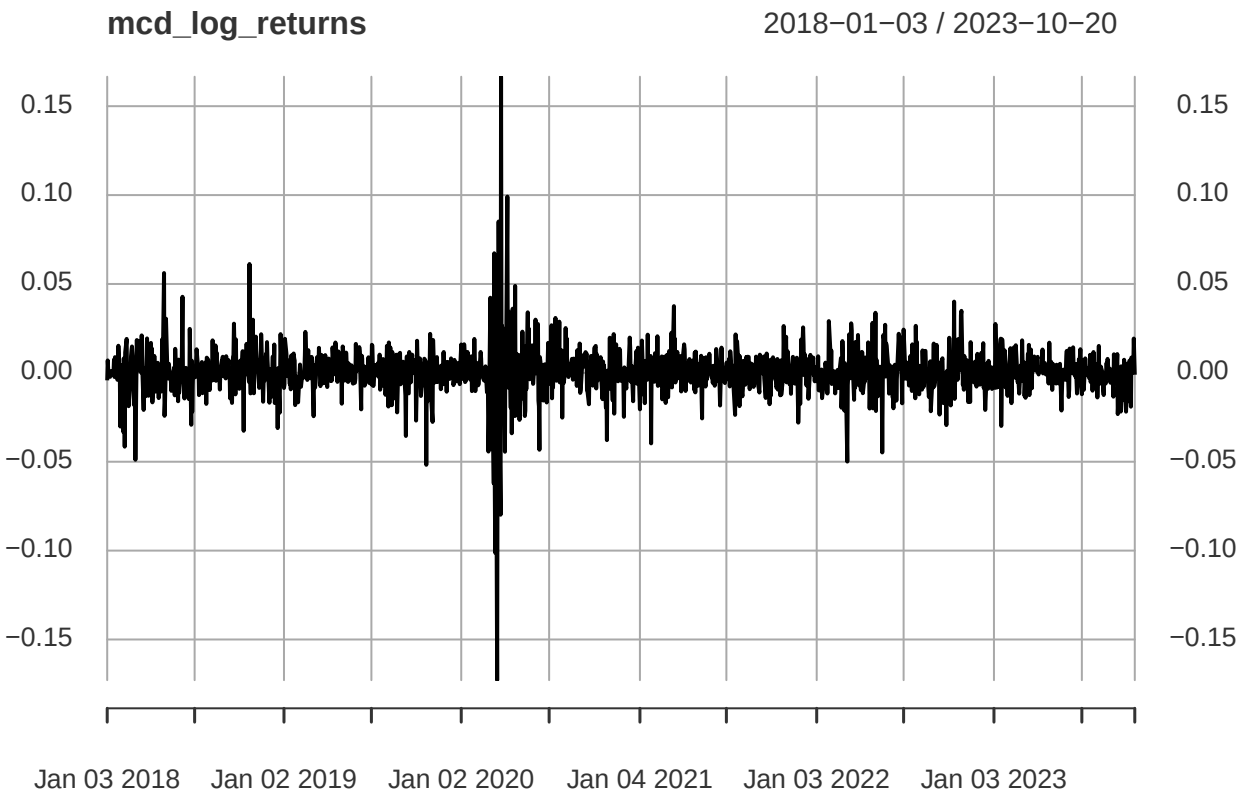
```
##            MCD.Adjusted
## 2018-01-02     151.3904
## 2018-01-03     150.7524
## 2018-01-04     151.8099
## 2018-01-05     152.1158
## 2018-01-08     152.0109
## 2018-01-09     151.6701
```

b. *Build a time series model for this data.*

```r
# Compute daily log returns.
mcd_log_returns <- diff(log(mcd_adj_price))

# Drop NA values (first value will be NA due to differencing).
mcd_log_returns <- na.omit(mcd_log_returns)

head(mcd_log_returns)
```

```
##             MCD.Adjusted
## 2018-01-03 -0.0042232795
## 2018-01-04  0.0069904941
## 2018-01-05  0.0020129462
## 2018-01-08 -0.0006896708
## 2018-01-09 -0.0022448968
## 2018-01-10 -0.0001732571
```

```r
plot(mcd_log_returns)
```




```
adf.test(mcd_log_returns)
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  mcd_log_returns
## Dickey-Fuller = -10.699, Lag order = 11, p-value = 0.01
## alternative hypothesis: stationary
```

```
mcd_model <- auto.arima(mcd_log_returns)
summary(mcd_model)
```

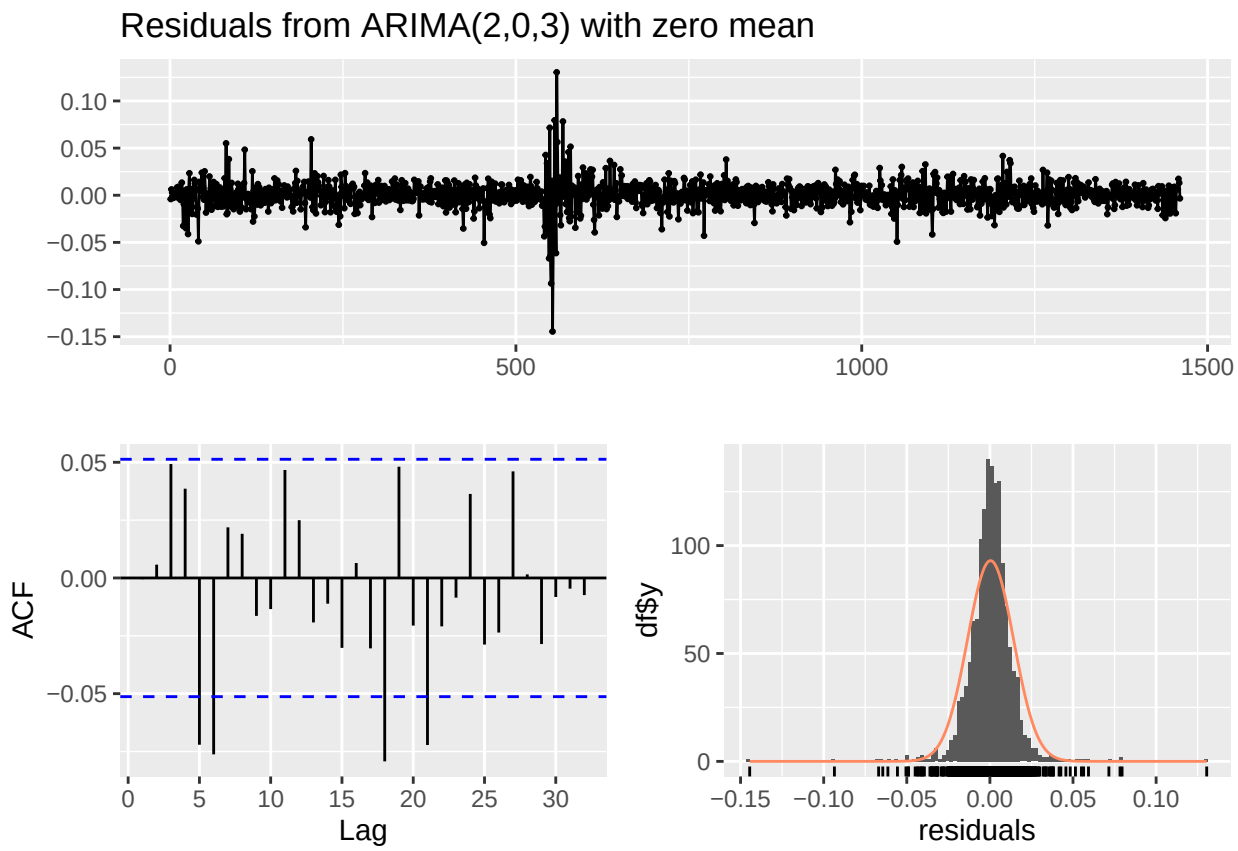
```
## Series: mcd_log_returns
## ARIMA(2,0,3) with zero mean
##
## Coefficients:
##           ar1      ar2     ma1     ma2     ma3
##       -1.6821  -0.8291  1.5986  0.7221  0.0186
## s.e.   0.0430   0.0395  0.0503  0.0576  0.0296
##
## sigma^2 = 0.0001934:  log likelihood = 4172.54
## AIC=-8333.09   AICc=-8333.03   BIC=-8301.37
##
## Training set error measures:
##                        ME       RMSE         MAE MPE MAPE      MASE
## Training set 0.0003836691 0.01388462 0.009093757 NaN  Inf 0.6732226
```

```
##                       ACF1
## Training set -0.0005232102
```

As such, for McDonald's daily log returns, the best model is an ARIMA($p = 2, d = 0, q = 3$) or an ARMA($p = 2, q = 3$).

c. *Evaluate its performance. Justify your choices.*

```
checkresiduals(mcd_model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,3) with zero mean
## Q* = 23.834, df = 5, p-value = 0.0002336
##
## Model df: 5.   Total lags used: 10
```

```
ljung_box_result <- Box.test(mcd_model$residuals, lag=log(length(mcd_model$residuals)))
ljung_box_result
```

```
##
##  Box-Pierce test
##
## data:  mcd_model$residuals
## X-squared = 22.529, df = 7.2862, p-value = 0.002513
```

As a result of the Ljung-Box test, we see that there is significant autocorrelation left in the residuals at the given lags. We must reevaluated our ARIMA model.

```r
# Create a function to perform grid search to find the best ARIMA based on AIC.
grid_search <- function(ts)
{
  best_model <- NULL
  best_aic <- Inf
  best_order <- c(0, 0, 0)

  for (p in 0:10)
  {
    for (d in 0:0)
    {
      for (q in 0:10)
      {
        model <- arima(ts, order = c(p, d, q))
        aic <- AIC(model)

        if (aic < best_aic)
        {
          best_model <- model
          best_aic <- aic
          best_order <- c(p, d, q)
        }
      }
    }
  }

  # Create a result data frame with the best AIC and formatted best order.
  result_df <- data.frame(Best_AIC = best_aic, Best_Order = paste0("(", paste(best_order, collapse = ",
  return(result_df)
}

grid_search(mcd_log_returns)
```

```
## Warning in arima(ts, order = c(p, d, q)): possible convergence problem: optim
## gave code = 1

## Warning in arima(ts, order = c(p, d, q)): possible convergence problem: optim
## gave code = 1

## Warning in arima(ts, order = c(p, d, q)): possible convergence problem: optim
## gave code = 1

## Warning in arima(ts, order = c(p, d, q)): possible convergence problem: optim
## gave code = 1

## Warning in arima(ts, order = c(p, d, q)): possible convergence problem: optim
## gave code = 1

## Warning in arima(ts, order = c(p, d, q)): possible convergence problem: optim
## gave code = 1

## Warning in arima(ts, order = c(p, d, q)): possible convergence problem: optim
## gave code = 1
```

```
## Warning in arima(ts, order = c(p, d, q)): possible convergence problem: optim
## gave code = 1

## Warning in arima(ts, order = c(p, d, q)): possible convergence problem: optim
## gave code = 1

## Warning in arima(ts, order = c(p, d, q)): possible convergence problem: optim
## gave code = 1

## Warning in arima(ts, order = c(p, d, q)): possible convergence problem: optim
## gave code = 1

## Warning in arima(ts, order = c(p, d, q)): possible convergence problem: optim
## gave code = 1

## Warning in arima(ts, order = c(p, d, q)): possible convergence problem: optim
## gave code = 1

## Warning in arima(ts, order = c(p, d, q)): possible convergence problem: optim
## gave code = 1

## Warning in arima(ts, order = c(p, d, q)): possible convergence problem: optim
## gave code = 1

##    Best_AIC Best_Order
## 1 -8362.122    (6,0,5)
```

```
mcd_model2 <- arima(mcd_log_returns, order = c(p = 6, d = 0, q = 5))

summary(mcd_model2)
```

```
##
## Call:
## arima(x = mcd_log_returns, order = c(p = 6, d = 0, q = 5))
##
## Coefficients:
##           ar1      ar2     ar3     ar4     ar5      ar6     ma1      ma2
##       -0.5258  -0.0067  0.2026  0.7185  0.6098  -0.1209  0.4313  -0.0029
## s.e.   0.0792   0.0535  0.0543  0.0403  0.0814   0.0376  0.0771   0.0470
##           ma3      ma4      ma5  intercept
##       -0.1339  -0.7509  -0.5436      5e-04
## s.e.   0.0510   0.0469   0.0820      0e+00
##
## sigma^2 estimated as 0.0001868:  log likelihood = 4194.06,  aic = -8362.12
##
## Training set error measures:
##                         ME       RMSE         MAE MPE MAPE      MASE
## Training set -4.092031e-05 0.01366707 0.009043517 NaN  Inf 0.6695032
##                      ACF1
## Training set -0.001512636
```

```
Box.test(mcd_model2$residuals, lag = log(length(mcd_model$residuals)))
```

```
##
##  Box-Pierce test
##
```

```
## data:  mcd_model2$residuals
## X-squared = 3.0209, df = 7.2862, p-value = 0.8998
```

As such, after creating a model which optimizes for the lowest AIC, we found one that satisifies the Ljung-Box test, implying that the residuals act as white noise.

Therefore, our new model is ARIMA($p = 6, d = 0, q = 5$).

```
# Determine the split point.
split_index <- round(length(mcd_log_returns) * 0.7)

# Create training and test sets.
train_data <- mcd_log_returns[1:split_index]
test_data <- mcd_log_returns[(split_index+1):length(mcd_log_returns)]

length(train_data)
```

```
## [1] 1022
```

```
length(test_data)
```

```
## [1] 438
```

```
forecast_results <- forecast(mcd_model2, h = length(test_data))

# Calculate accuracy metrics.
accuracy_metrics <- accuracy(forecast_results, test_data)
accuracy_metrics
```

```
##                         ME       RMSE         MAE  MPE MAPE      MASE
## Training set -4.092031e-05 0.01366707 0.009043517  NaN  Inf 0.6695032
## Test set     -5.871857e-04 0.01109517 0.008320230 -Inf  Inf 0.6159574
##                      ACF1
## Training set -0.001512636
## Test set               NA
```

```
summary(mcd_model2)
```

```
##
## Call:
## arima(x = mcd_log_returns, order = c(p = 6, d = 0, q = 5))
##
## Coefficients:
##           ar1      ar2     ar3     ar4     ar5      ar6     ma1      ma2
##       -0.5258  -0.0067  0.2026  0.7185  0.6098  -0.1209  0.4313  -0.0029
## s.e.   0.0792   0.0535  0.0543  0.0403  0.0814   0.0376  0.0771   0.0470
##           ma3      ma4      ma5  intercept
##       -0.1339  -0.7509  -0.5436      5e-04
## s.e.   0.0510   0.0469   0.0820      0e+00
##
## sigma^2 estimated as 0.0001868:  log likelihood = 4194.06,  aic = -8362.12
##
## Training set error measures:
##                         ME       RMSE         MAE MPE MAPE      MASE
## Training set -4.092031e-05 0.01366707 0.009043517 NaN  Inf 0.6695032
##                      ACF1
## Training set -0.001512636
```

As we can see the RMSE and MAE are very low for the respective forecast results on the training and testing data, indicating a strong model.

Specifically, the RMSE and MAE are lower for the testing set (0.01366699 vs. 0.01109467 and 0.009042408 vs. 0.008319826, respectively), which is very good.

Problem 5 (20pt)

Consider the monthly U.S. unemployment rates from January 1947 to March 2016. Due to strong serial dependence, we analyze the differenced series $x_t = r_t - r_{t-1}$ where r_t is the seasonally adjusted unemployment rate. Answer the following questions, using the R output listed below the questions. Note: A fitted ARIMA model should include residual variance.

a. *The auto.arima command in R specifies an ARIMA$(2,0,2)$ model for x_t. The fitted model is referred to as m1 in the output. Write down the fitted model.*

The **m1** model is an ARMA$(p = 2, q = 2)$. As such, this is the fitted model:

$$x_t = 1.6546x_{t-1} - 0.7753x_{t-2} + a_t - 1.6288a_{t-1} + 0.8440a_{t-2}$$

.

b. Model checking shows two large outliers. An ARIMA(2,0,2) model with two outliers are then specified, **m3**. Write down the fitted model.

The **m3** model is an ARMA$(p = 2, q = 2)$ with two external regressors $i22_t$ and $i21_t$. As such, this is the fitted model:

$$x_t = 1.6901x_{t-1} - 0.7909x_{t-2} + a_t - 1.6128a_{t-1} + 0.8014a_{t-2} - 1.5302 \times i22_t + 1.1472 \times i21_t$$

c. *Model checking shows some serial correlations at lags 12 and 24. A seasonal model is then employed and called m4. Write down the fitted model.*

The **m4** model is an ARIMA model with non-seasonal and seasonal components with a period of 12. As such, this is the fitted model:

$$x_t = 1.2357x_{t-1} - 0.3608x_{t-2} + a_t - 1.2354a_{t-1} + 0.5151a_{t-1} + 0.5542x_{t-12} - 0.8220a_{t-12}$$

d. *The outliers remain in the seasonal model. Therefore, a refined model is used and called m5. Write down the fitted model.*

The **m5** model is an ARIMA model with non-seasonal and seasonal components with a period of 12 with external regressors (outliers). As such, this is the fitted model:

$$x_t = 1.5743x_{t-1} - 0.6591x_{t-2} + a_t - 1.4869a_{t-1} + 0.6720a_{t-2} + 0.5488x_{t-12} - 0.8208a_{t-12} - 1.4762 \times i22_t + 1.1441 \times i21_t$$

e. *Based on the model checking statistics provided, are there serial correlations in the residuals of model m5? Why?*

Since the Ljung-Box test resulted in a p-value equal to 0.2674 which is greater than 0.05, we fail to reject the null hypothesis that the residuals exhibit no autocorrelation up to lag 24, implying that there are no serial correlations in the residuals of model **m5** at the 5% significance level.

f. *Among models m1, m3, m4, and m5, which model is preferred under the in-sample fit? Why?*

Since model **m1** had an AIC of -336.71, **m3** had an AIC of -439.72, **m4** had an AIC of -396.43, and **m5** had an AIC of -510.4, **m5** is preferred under the in-sample fit due to the lowest AIC.

g. *If root mean squares of forecast errors are used in out-of-sample prediction, which model is preferred? Why?*

Since model **m1** had a RMSE of 0.1621524, **m3** had a RMSE of 0.163189, **m4** had a RMSE of 0.1499355, and **m5** had a RMSE of 0.1493882, **m5** is preferred since it has the lowest RMSE.

h. *If mean absolute forecast errors are used in out-of-sample comparison, which model is preferred? Why?*

Since model **m1** had a MAE of 0.1242145, **m3** had a MAE of 0.1237345, **m4** had a MAE of 0.1164277, and **m5** had a MAE of 0.1164356, **m4** is preferred since it has the lowest MAE.

i. *Consider models m1 and m3 State the impact of outliers on in-sample fitting.*

Model **m1** is an $(ARIMA)(p = 2, d = 0, q = 2)$ model without taking into account outliers. It has an AIC of -336.71.

Model **m3** is an ARIMA$(p = 2, d = 0, q = 2)$ model with taking into account significant outliers at lags 21 and 22. It has an AIC of -439.72.

By incorporating outliers in the **m3** model, the in-sample fit has been enhanced, as evidenced by its reduced AIC value. This suggests that when the model addresses these extreme observations, it more accurately represents the inherent patterns in the data, minimizing the disproportionate impact of these outliers on the model's estimates.

j. *Again, consider models m1 and m3. State the impact of outliers on out-of-sample predictions.*

Model **m1** had a RMSE of 0.1621524 and model **m3** had a RMSE of 0.163189. As such, inclusion of outliers in the **m3** model leads to a slight increase in the RMSE, indicating that the model with outliers (**m3**) might not generalize as well to new data as the **m1** model does.

However, the MAE is marginally reduced in the **m3** model (0.1237345 vs. 0.1242145), suggesting that on average, the magnitude of the forecast errors might be slightly smaller.

Overall, the impact of outliers in the m3 model on out-of-sample predictions seems mixed, offering both potential benefits and drawbacks.