ISYE 6402 Homework 4

Background

For this data analysis, you will again analyze the currency exchange data but to a greater extent and including two different currencies for comparison. File *DailyCurrencyData.csv* contains the *daily* exchange rate of EUR/USD and GBP/USD from January 1999 through December 31st 2020. File *MonthlyCurrencyData.csv* contains the *monthly* exchange rate of EUR/USD and GBP/USD for the same time period. Similarly to homework 2, we will aggregate the daily data into weekly data. We will compare our analysis using ARMA modeling on both weekly and monthly data for the two currencies.

```
library(zoo)
library(lubridate)
library(mgcv)
```

Instructions on reading the data

To read the data in `R`, save the file in your working directory (make sure you have changed the directory if different from the R working directory) and read the data using the `R` function `read.csv()`

```
fpath <- ""

daily <- read.csv("DailyCurrencyData-2.csv", head = TRUE)
monthly <- read.csv("MonthlyCurrencyData-1.csv", head = TRUE)

daily$Date <- as.Date(daily$Date, "%m/%d/%Y")
monthly$Date <- as.Date(paste0(monthly$Date, "-01"), "%Y-%m-%d")
colnames(monthly) <- colnames(daily)
```

Question 1. Weekly vs Monthly Exploratory Data Analysis (20 points)

1a. Based on your intuition, when would you use weekly vs monthly time series data?

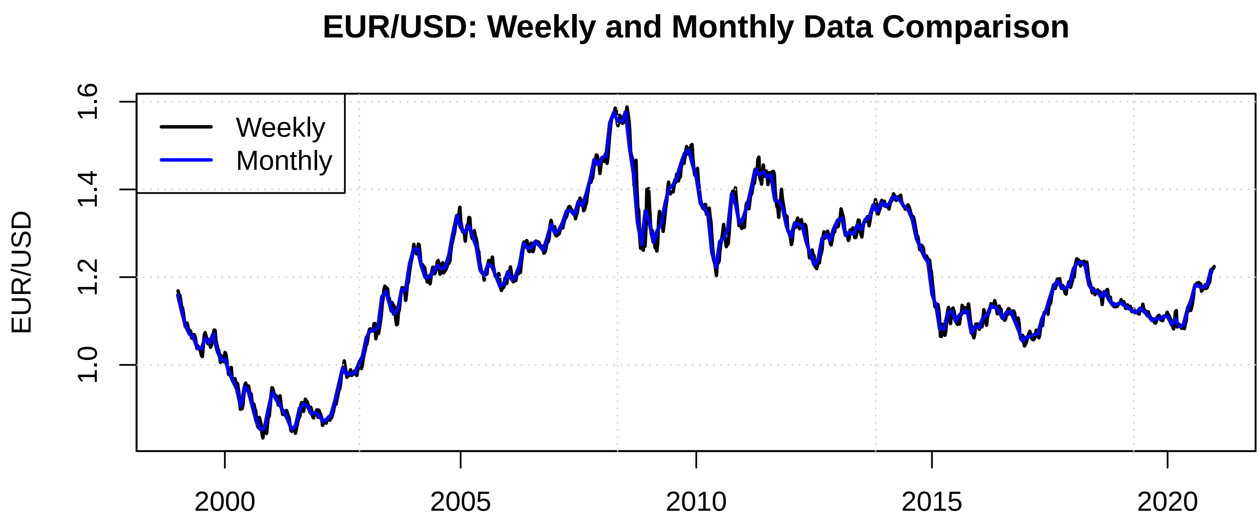
Response

When we have both the weekly and monthly data available, the weekly time series would be used for a more granular analysis, specifically, when the short term changes are of more relevance. In cases where the data fluctuates quickly (high volatility), we would prefer to gauge the variance in these observations. The disadvantage of using the weekly data versus monthly data would be introducing more variability than needed when less granular analysis (e.g. monthly predictions) is of interest.

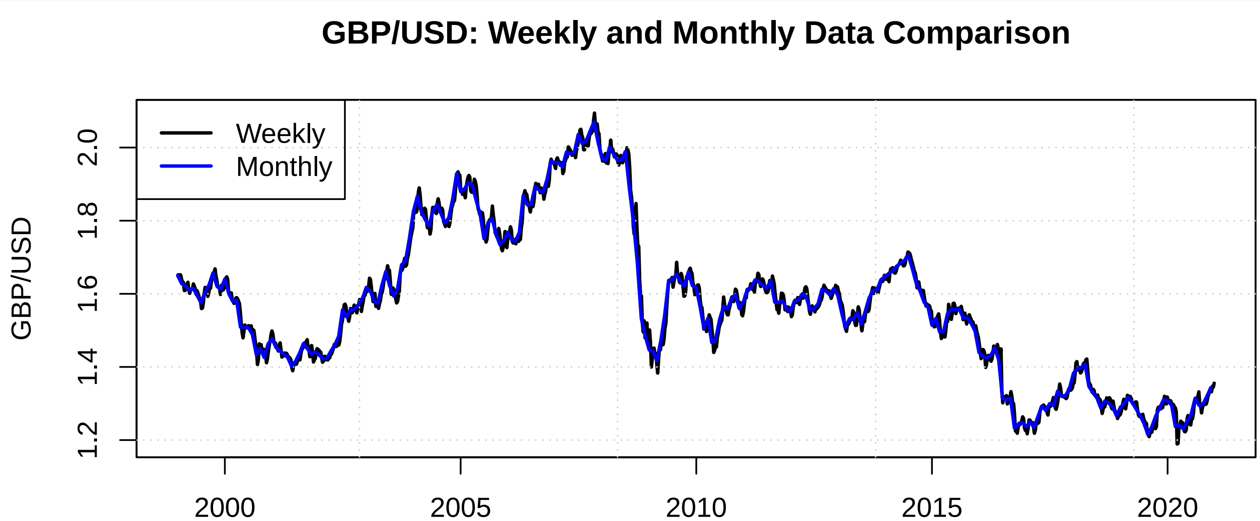
1b. Plot the time series plots for both currency exchange rates comparing weekly vs monthly data. How do the weekly and monthly time series data compare? How do the time series for the two currencies compare?

```
daily <- na.locf(daily)

weekly <- daily
weekly$Date <- floor_date(weekly$Date, "week")
```

```r
weekly <- aggregate(weekly[, 2:3], by = list(weekly$Date), FUN = mean)
colnames(weekly)[1] <- "Date"

daily <- zoo(daily[-1], daily$Date)
weekly <- zoo(weekly[-1], weekly$Date)
monthly <- zoo(monthly[-1], monthly$Date)

plot(weekly$EUR, xlab = "", ylab = "EUR/USD", col = "black", lwd = 2,
     main = "EUR/USD: Weekly and Monthly Data Comparison")
grid()
lines(monthly$EUR, col = "blue", lwd = 2)
legend("topleft", c("Weekly", "Monthly"), col = c("black", "blue"), lwd = 2)
```




```r
plot(weekly$GBP, xlab = "", ylab = "GBP/USD", col = "black", lwd = 2,
     main = "GBP/USD: Weekly and Monthly Data Comparison")
grid()
lines(monthly$GBP, col = "blue", lwd = 2)
legend("topleft", c("Weekly", "Monthly"), col = c("black", "blue"), lwd = 2)
```



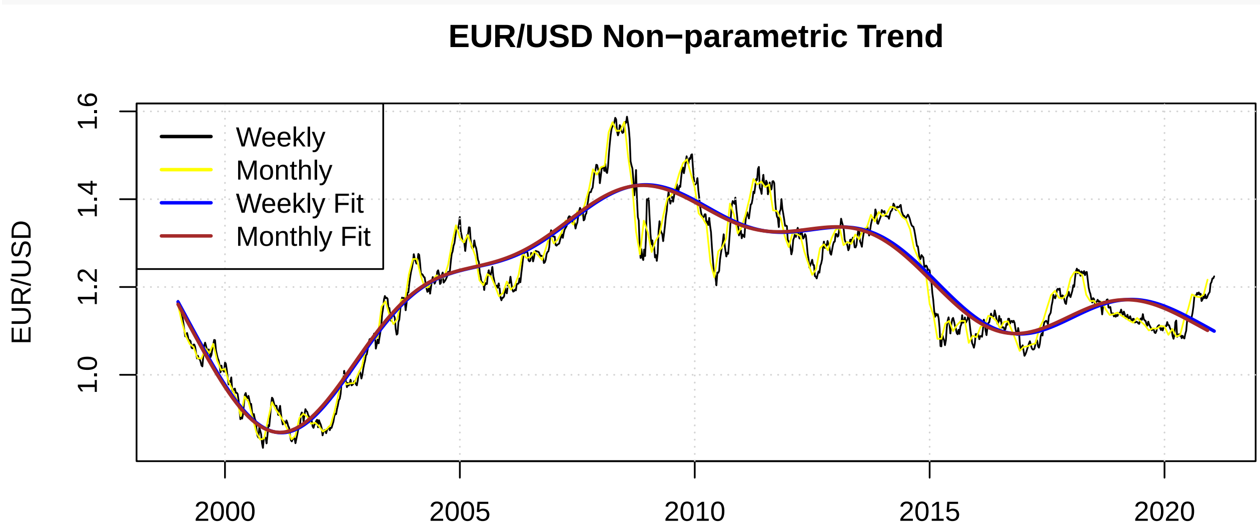
Response: Weekly vs Montly Time Series data comparison

The time series data consist of a little over 20 years for EUR and GBP, both on a weekly and monthly basis.

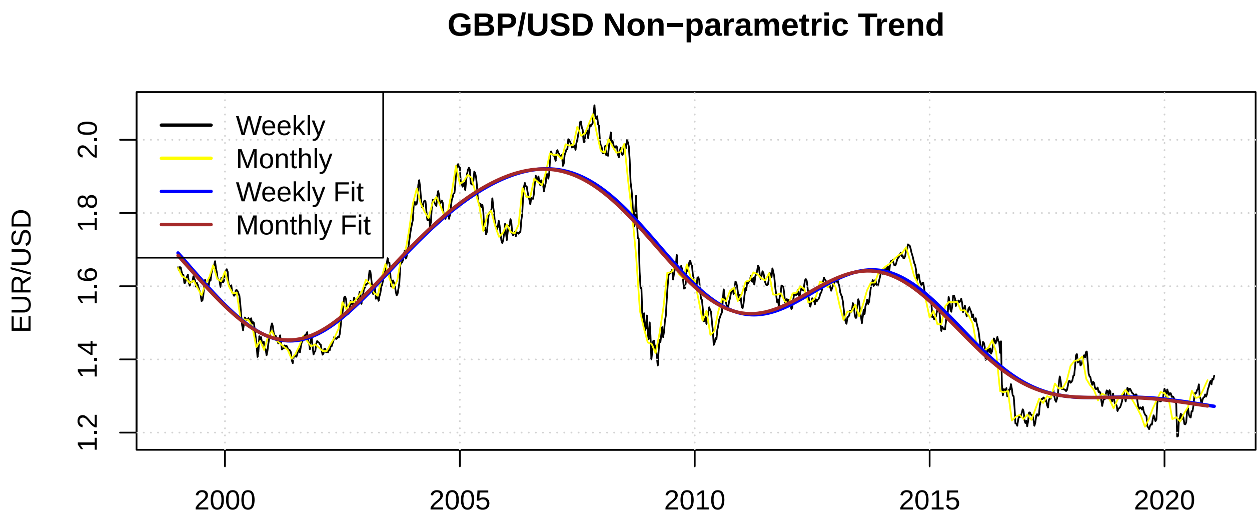
We can clearly see that the monthly data is just a smoothed version of the weekly data. For Euro currency, there is no clear seasonality in the time series, but the trend is increasing at first and then reverses over time. Similarly, for GBP currency, there is no clear seasonality in the time series but there is an increasing trend for a few years until 2008-09, then followed by a drastic drop, continuing with a decreasing the trend.

1c. Fit a non-parametric trend using splines regression to both the weekly and monthly time series data for both currencies. Overlay the fitted trends for each of the currency separately. How do the trends compare when comparing those fitted using the weekly and monthly data? How do the trends for the two currencies compare?

```
ts_weekly <- ts(weekly, start = 1999, frequency = 52)
ts_monthly <- ts(monthly, start = 1999, frequency = 12)

points_weekly <- 1:nrow(ts_weekly)
points_weekly <- (points_weekly - min(points_weekly)) / max(points_weekly)
points_monthly <- 1:nrow(ts_monthly)
points_monthly <- (points_monthly - min(points_monthly)) / max(points_monthly)

gam.w.eur <- gam(ts_weekly[, "EUR"] ~ s(points_weekly))
gam.w.gbp <- gam(ts_weekly[, "GBP"] ~ s(points_weekly))
gam.m.eur <- gam(ts_monthly[, "EUR"] ~ s(points_monthly))
gam.m.gbp <- gam(ts_monthly[, "GBP"] ~ s(points_monthly))

gam.w.eur.fit <- ts(fitted(gam.w.eur), start = 1999, frequency = 52)
gam.w.gbp.fit <- ts(fitted(gam.w.gbp), start = 1999, frequency = 52)
gam.m.eur.fit <- ts(fitted(gam.m.eur), start = 1999, frequency = 12)
gam.m.gbp.fit <- ts(fitted(gam.m.gbp), start = 1999, frequency = 12)

# Is there a trend for EUR?

ts.plot(ts_weekly[, "EUR"], xlab = "", ylab = "EUR/USD",
        main = "EUR/USD Non-parametric Trend")
grid()
lines(ts_monthly[, "EUR"], col = "yellow")
lines(gam.w.eur.fit, lwd = 2, col = "blue")
lines(gam.m.eur.fit, lwd = 2, col = "brown")
legend("topleft", legend = c("Weekly", "Monthly", "Weekly Fit", "Monthly Fit"),
       col = c("black", "yellow", "blue", "brown"), lwd = 2)
```



```r
# Is there a trend for GBP?

ts.plot(ts_weekly[, "GBP"], xlab = "", ylab = "EUR/USD",
        main = "GBP/USD Non-parametric Trend")
grid()
lines(ts_monthly[, "GBP"], col = "yellow")
lines(gam.w.gbp.fit, lwd = 2, col = "blue")
lines(gam.m.gbp.fit, lwd = 2, col = "brown")
legend("topleft", legend = c("Weekly", "Monthly", "Weekly Fit", "Monthly Fit"),
       col = c("black", "yellow", "blue", "brown"), lwd = 2)
```

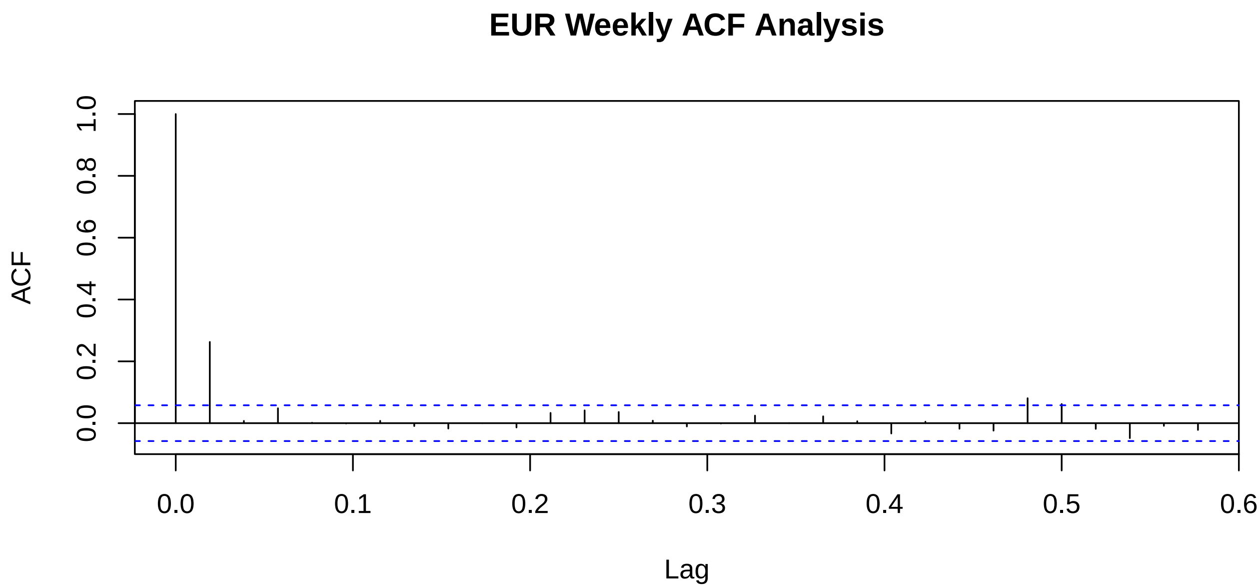


Response: Comparing Trend Estimation using weekly vs Monthly Data

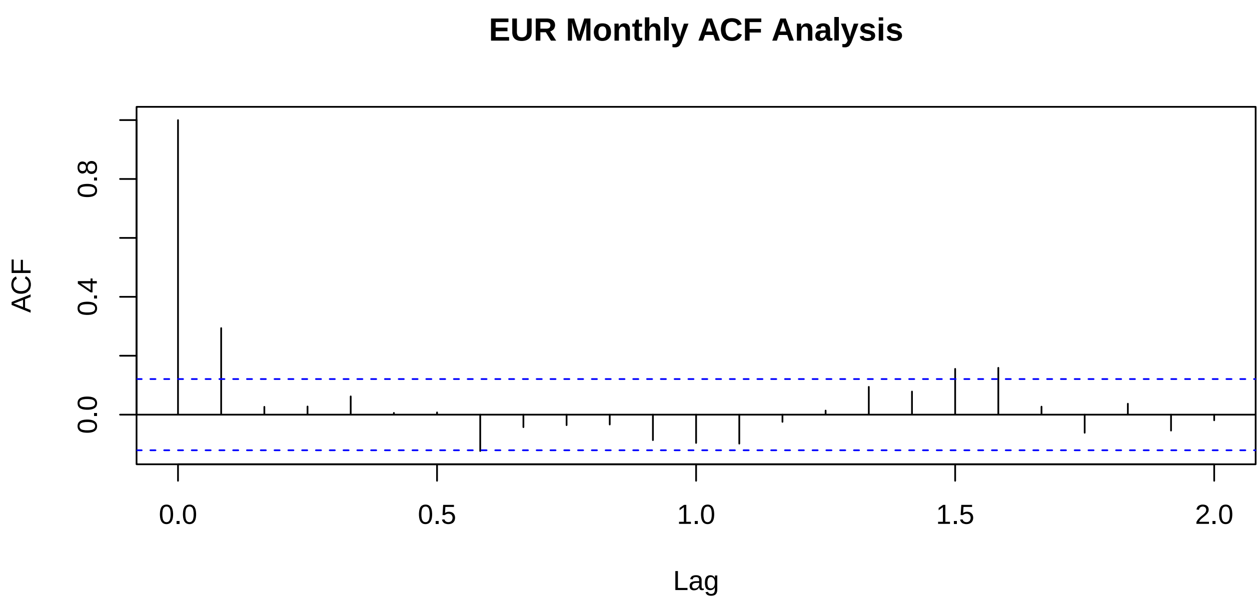
The trend estimated using weekly and monthly data are almost identical, hence the trend estimation primarily captures the overall pattern regardless whether the data are less or more granular.

1d. Take the 1st order difference of the time series weekly vs monthly data. Plot the ACF plots and compare. How do the difference time series for weekly and monthly data compare in terms of stationarity? How do the difference time series for the two currencies compare in terms of serial dependence and stationarity?

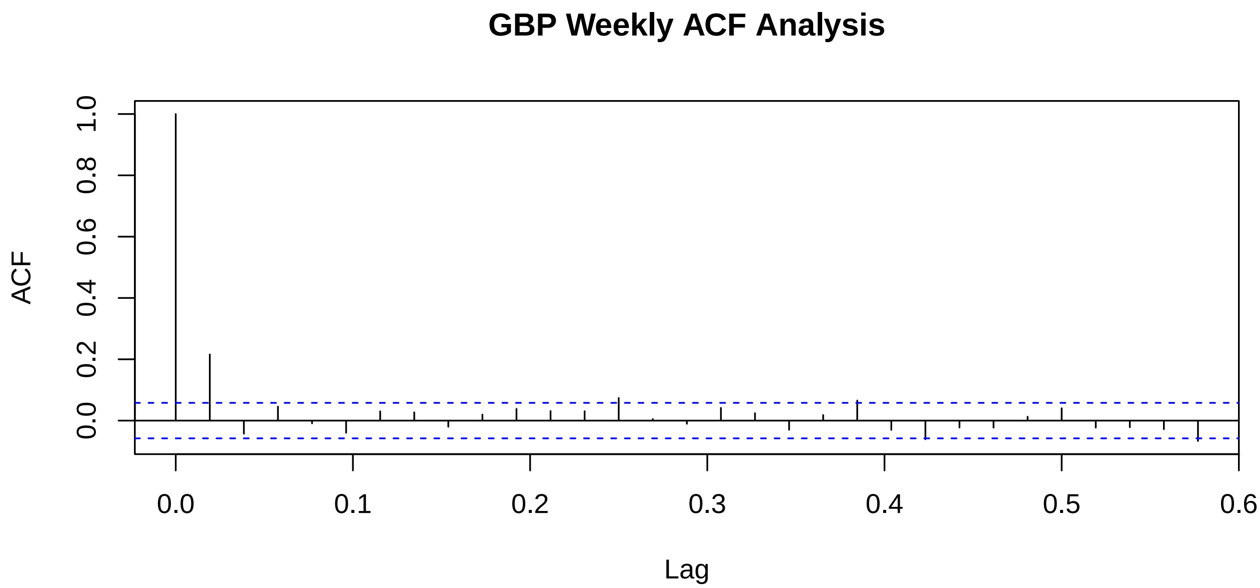
```r
diff_ts_weekly <- diff(ts_weekly)
diff_ts_monthly <- diff(ts_monthly)

acf(diff_ts_weekly[, "EUR"], na.action = na.pass,
    main = "EUR Weekly ACF Analysis")
```



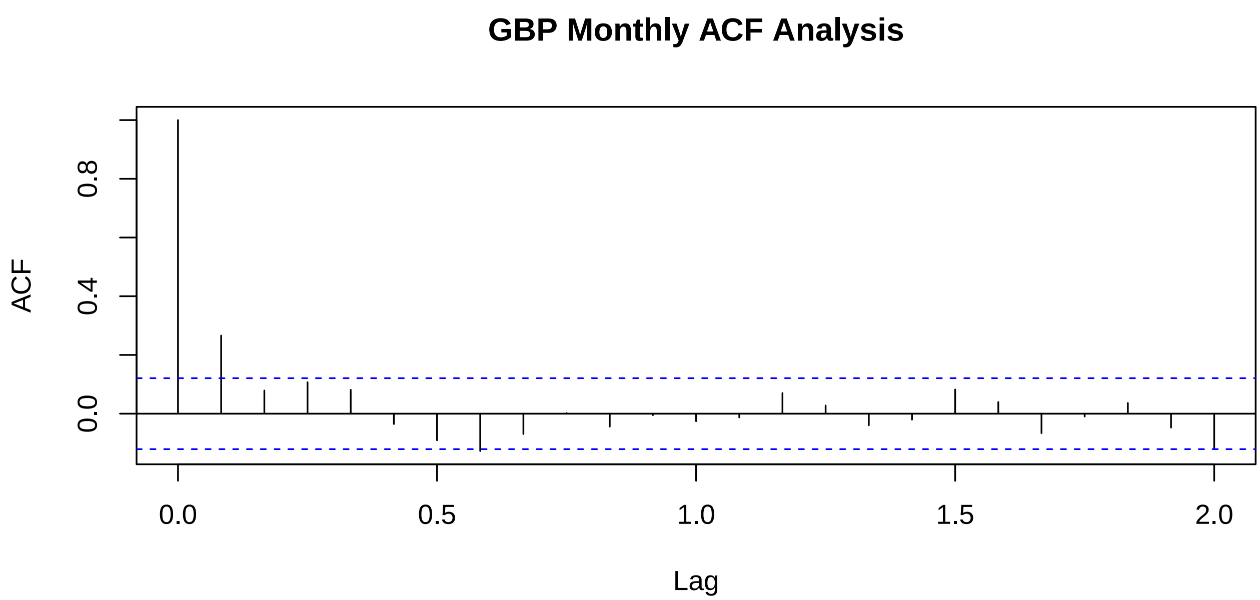

```r
acf(diff_ts_monthly[, "EUR"], na.action = na.pass,
    main = "EUR Monthly ACF Analysis")
```




```r
acf(diff_ts_weekly[, "GBP"], na.action = na.pass,
    main = "GBP Weekly ACF Analysis")
```




```
acf(diff_ts_monthly[, "GBP"], na.action = na.pass,
    main = "GBP Monthly ACF Analysis")
```



Response: Exploratory Analysis of 1st Order Difference Data

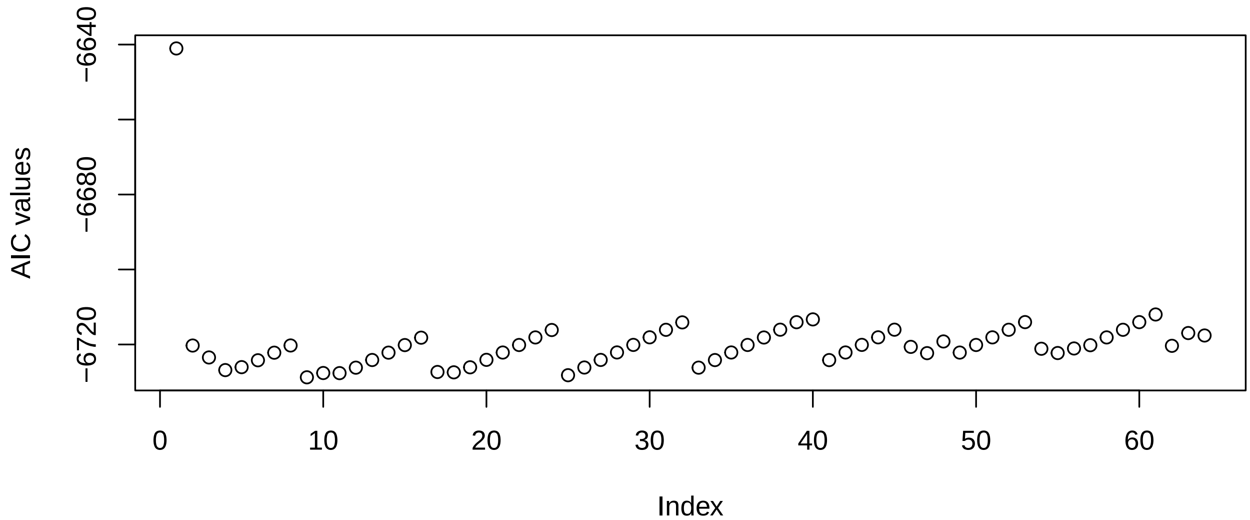
From the time series charts and ACF plots of the weekly against the monthly data for both EUR and GBP, we have two conclusions: 1. The differenced time series of the weekly data looks quite close to white noise, and 2. For EUR, the stationarity of the monthly differenced data is not as good as that of the weekly differenced data. For GBP, however, the significance of later lags seems to be insignificant in case of monthly data but that is not the case with weekly data.

Question 2. ARIMA Fitting and Forecasting: Weekly Data Analysis (23 points)

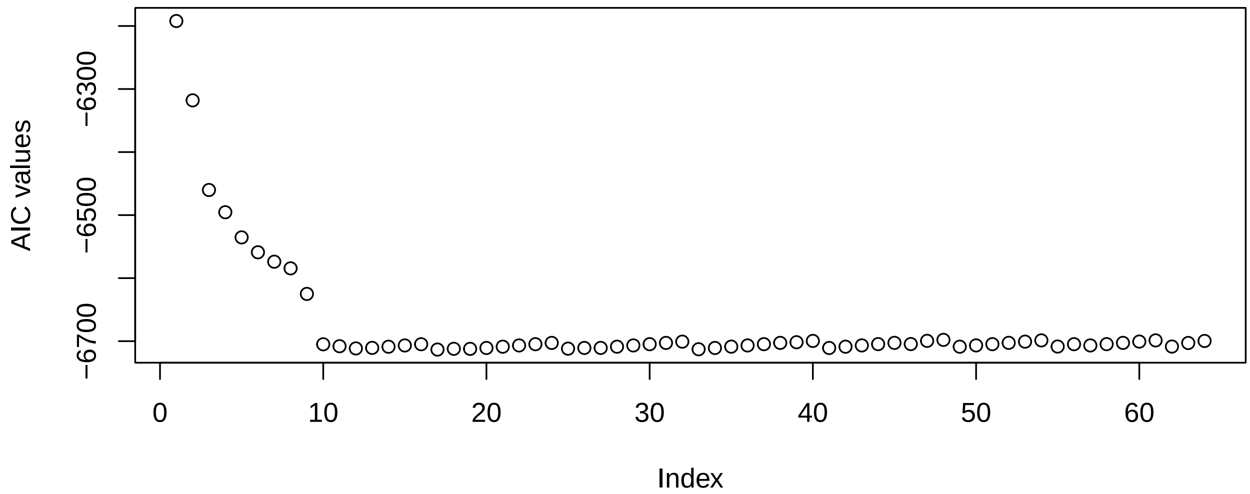
2a. Divide the data into training and testing data set, where the training data exclude the last eight weeks of data (November and December 2020) with the testing data including the last eight weeks of data. For both currency exchange rates and using the training datasets, use the iterative model to fit an ARIMA(p,d,q) model with max AR and MA orders of 8, and a differencing order of 1 or 2. Display the summary of the final model fit. Compare statistical significance of the coefficients. Would a lower order model be suggested based on the statistical significance of the coefficients?

Analyzing weekly data with ARIMA model fitting

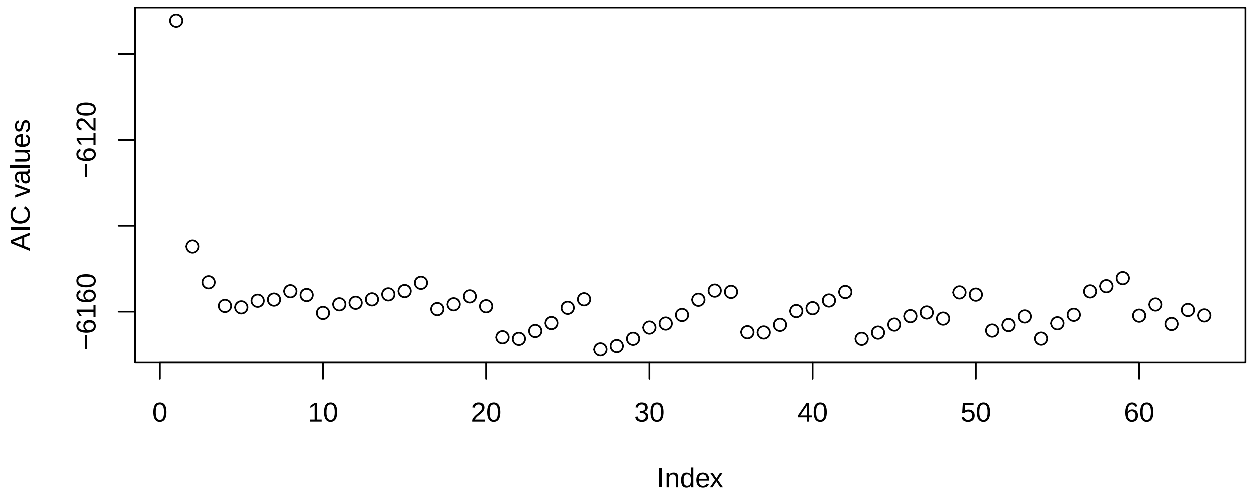
```r
num <- 8

# set up the training and testing data

eur.w <- ts_weekly[, "EUR"]
gbp.w <- ts_weekly[, "GBP"]

eur.w.train <- eur.w[1:(length(eur.w) - num)]
eur.w.test <- eur.w[(length(eur.w) - num + 1):length(eur.w)]

gbp.w.train <- gbp.w[1:(length(gbp.w) - num)]
gbp.w.test <- gbp.w[(length(gbp.w) - num + 1):length(gbp.w)]

## EUR Currency Analysis -----

## 1. Differencing Order = 1

n <- length(eur.w.train)
norder <- num

p <- 1:norder - 1
q <- 1:norder - 1
aic <- matrix(0, norder, norder)
for (i in 1:norder) {
  for (j in 1:norder) {
    modij = arima(eur.w.train,
                  order = c(p[i], 1, q[j]),
                  method = 'ML')
    aic[i, j] = modij$aic - 2 * (p[i] + q[j] + 1) + 2 * (p[i] + q[j] + 1) *
      n / (n - p[i] - q[j] - 2)
  }
}

# Extract the "best" one according to AIC
aicv <- as.vector(aic)
plot(aicv, ylab = "AIC values")
```



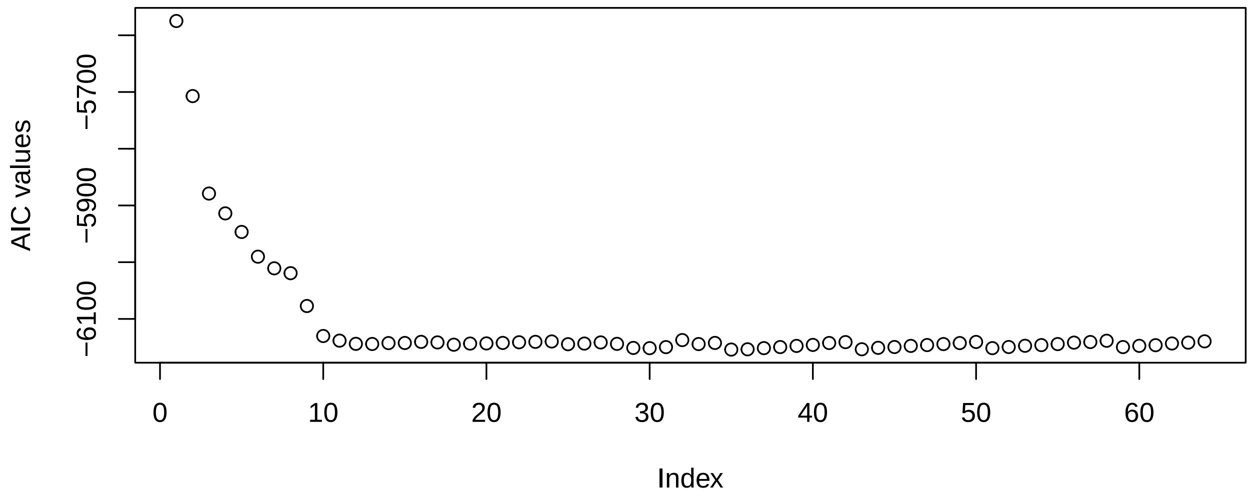

```
indexp <- rep(c(1:norder), norder)
indexq <- rep(c(1:norder), each = norder)
indexaic <- which(aicv == min(aicv))
porder_1 <- indexp[indexaic] - 1
qorder_1 <- indexq[indexaic] - 1

# Fit the "best" arima model
final_model <- arima(eur.w.train,
                     order = c(porder_1, 1, qorder_1),
                     method = "ML")

# 2. Differencing Order = 2
for (i in 1:norder) {
  for (j in 1:norder) {
    modij = arima(eur.w.train,
                  order = c(p[i], 2, q[j]),
                  method = 'ML')
    aic[i, j] = modij$aic - 2 * (p[i] + q[j] + 1) + 2 * (p[i] + q[j] + 1) *
      n / (n - p[i] - q[j] - 2)
  }
}

# Extract the "best" one according to AIC
aicv <- as.vector(aic)
plot(aicv, ylab = "AIC values")
```




```
indexp <- rep(c(1:norder), norder)
indexq <- rep(c(1:norder), each = norder)
indexaic <- which(aicv == min(aicv))
porder_2 <- indexp[indexaic] - 1
qorder_2 <- indexq[indexaic] - 1

# Fit the "best" arima model
final_model2 <- arima(eur.w.train,
                      order = c(porder_2, 2, qorder_2),
                      method = "ML")

## GBP Currency Analysis -----

## 1. Differencing Order = 1
for (i in 1:norder) {
  for (j in 1:norder) {
    modij = stats::arima(gbp.w.train,
                         order = c(p[i], 1, q[j]),
                         method = 'ML')
    aic[i, j] = modij$aic - 2 * (p[i] + q[j] + 1) + 2 * (p[i] + q[j] + 1) *
      n / (n - p[i] - q[j] - 2)

  }
}

# Extract the "best" one according to AIC
aicv <- as.vector(aic)
plot(aicv, ylab = "AIC values")
```




```
indexp <- rep(c(1:norder), norder)
indexq <- rep(c(1:norder), each = norder)
indexaic <- which(aicv == min(aicv))
porder_3 <- indexp[indexaic] - 1
qorder_3 <- indexq[indexaic] - 1

# Fit the "best" arima model
final_model_gb <- arima(gbp.w.train,
                        order = c(porder_3, 1, qorder_3),
                        method = "ML")

## 2. Differencing order = 2
for (i in 1:norder) {
  for (j in 1:norder) {
    modij = stats::arima(gbp.w.train,
                         order = c(p[i], 2, q[j]),
                         method = 'ML')
    aic[i, j] = modij$aic - 2 * (p[i] + q[j] + 1) + 2 * (p[i] + q[j] + 1) *
      n / (n - p[i] - q[j] - 2)

  }
}

# Extract the "best" one according to AIC
aicv <- as.vector(aic)
plot(aicv, ylab = "AIC values")
```




```
indexp <- rep(c(1:norder), norder)
indexq <- rep(c(1:norder), each = norder)
indexaic <- which(aicv == min(aicv))
porder_4 <- indexp[indexaic] - 1
qorder_4 <- indexq[indexaic] - 1

# Fit the "best" arima model
final_model_gb2 <- arima(gbp.w.train,
                         order = c(porder_4, 2, qorder_4),
                         method = "ML")


### AIC Summary ----

print("EUR Model Selection:")
```

```
## [1] "EUR Model Selection:"
```

```
print(paste0("ARIMA: (", porder_1, ", 1, ", qorder_1,
             ") with AICc = ", round(final_model$aic, 3)))
```

```
## [1] "ARIMA: (0, 1, 1) with AICc = -6728.767"
```

```
print(paste0("ARIMA: (", porder_2, ", 2, ", qorder_2,
             ") with AICc = ", round(final_model2$aic, 3)))
```

```
## [1] "ARIMA: (0, 2, 2) with AICc = -6713.333"
```

```
print("GBP Model Selection:")
```

```
## [1] "GBP Model Selection:"
```

```
print(paste0("ARIMA: (", porder_3, ", 1, ", qorder_3,
             ") with AICc = ", round(final_model_gb$aic, 3)))
```

```
## [1] "ARIMA: (2, 1, 3) with AICc = -6168.858"
```

```
print(paste0("ARIMA: (", porder_4, ", 2, ", qorder_4,
             ") with AICc = ", round(final_model_gb2$aic, 3)))
```

```
## [1] "ARIMA: (2, 2, 4) with AICc = -6154.124"
```

Response: Analysis of the ARIMA Fit for the Weekly and Monthly Data

The selected models for the weekly data are as follows:

- EUR Model Selection: ARIMA(0,1,1) with AICc = -6728.767 vs ARIMA(0,2,2) with AICc = -6713.333, which are quite similar in the AICc value although the ARIMA(0,1,1) model has a smaller AICc value and is less complex.
- GDP Model Selection: ARIMA(2,1,3) with AICc = -6168.858 vs ARIMA(2,2,4) with AICc = -6154.124, with the ARIMA(2,1,3) having a lower AICc value hence selected.

In order to evaluate whether a reduced order for AR or MA will result in a less complex model but with similar explanatory power, we can evaluate the p-value of the z-test for statistical significance of the coefficients corresponding to the highest AR or MA order. Below, I am providing a simple pvalue function that can be used to compute the p-values of the z-tests for all coefficients.

```
## p-value function for the z-test taking as input the test statistic
pvalue.coef <- function(tv){
  2*(1-pnorm(abs(tv)))
}

## compute the test statistics
tv.eur.weekly <-as.numeric(final_model$coef)/as.numeric(sqrt(diag(final_model$var.coef)))
tv.gdp.weekly <-as.numeric(final_model_gb$coef)/as.numeric(sqrt(diag(final_model_gb$var.coef)))

## Apply the pvalue.coef function
pvalues.eur.weekly <- sapply(tv.eur.weekly, pvalue.coef)
pvalues.gdp.weekly <- sapply(tv.gdp.weekly, pvalue.coef)

print("EUR T Values:")
```

```
## [1] "EUR T Values:"
```

```
print(tv.eur.weekly)
```

```
## [1] 9.951388
```

```
print("GBP T Values:")
```

```
## [1] "GBP T Values:"
```

```
print(tv.gdp.weekly)
```

```
## [1] -26.004397 -46.136240  23.287146  38.104968   9.195049
```

```
print("EUR P Values:")
```

```
## [1] "EUR P Values:"
```

```
print(pvalues.eur.weekly)
```

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

```
print("GBP P Values:")
```

```
## [1] "GBP P Values:"
```

```
print(pvalues.gdp.weekly)
```

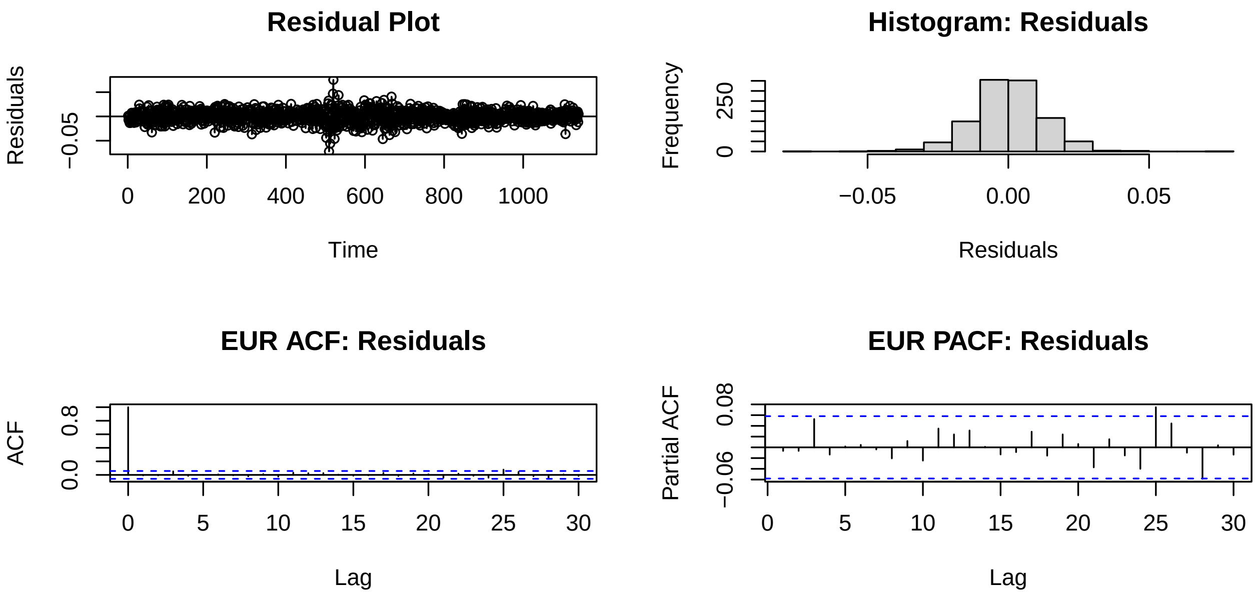
```
## [1] 0 0 0 0 0
```

Response: Statistical Significance

All the coefficients are statistically significant since their p-values are approximately zero. Hence the higher order coefficients should be included in the models.

2b. Evaluate the model residuals using the ACF and PACF plots, the residual plot and residuals' histogram as well as hypothesis testing for serial correlation for the selected models in (2a) for the two currencies. Does the model fit the time series data? Compare the model fit for the two currency exchange rates.

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

# EUR Residuals
resids <- resid(final_model)
plot(resids,
     ylab = 'Residuals',
     type = 'o',
     main = "Residual Plot")
abline(h = 0)
hist(resids, xlab = 'Residuals', main = 'Histogram: Residuals')
acf(resids, main = "EUR ACF: Residuals")
pacf(resids, main = "EUR PACF: Residuals")
```




```
Box.test(
  final_model$resid,
  lag = (porder_1 + qorder_1 + 1),
  type = "Box-Pierce",
  fitdf = (porder_1 + qorder_1)
)
```

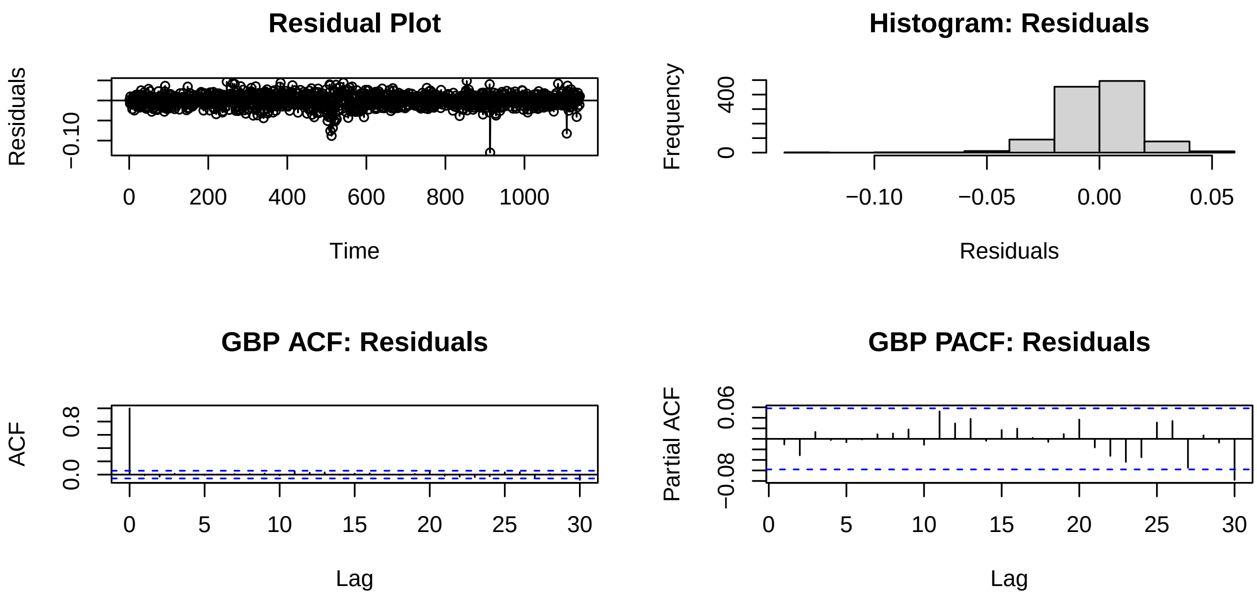
```
##
##  Box-Pierce test
##
## data:  final_model$resid
## X-squared = 0.098854, df = 1, p-value = 0.7532
```

```
Box.test(
  final_model$resid,
  lag = (porder_1 + qorder_1 + 1),
  type = "Ljung-Box",
```

```
  fitdf = (porder_1 + qorder_1)
)
```

```
##
##  Box-Ljung test
##
## data:  final_model$resid
## X-squared = 0.099157, df = 1, p-value = 0.7528
```

```
# GBP Residuals
resids <- resid(final_model_gb)
plot(resids,
     ylab = 'Residuals',
     type = 'o',
     main = "Residual Plot")
abline(h = 0)
hist(resids, xlab = 'Residuals', main = 'Histogram: Residuals')
acf(resids, main = "GBP ACF: Residuals")
pacf(resids, main = "GBP PACF: Residuals")
```




```
Box.test(
  final_model_gb$resid,
  lag = (porder_3 + qorder_3 + 1),
  type = "Box-Pierce",
  fitdf = (porder_3 + qorder_3)
)
```

```
##
##  Box-Pierce test
##
## data:  final_model_gb$resid
## X-squared = 1.5121, df = 1, p-value = 0.2188
```

```
Box.test(
  final_model_gb$resid,
  lag = (porder_3 + qorder_3 + 1),
  type = "Ljung-Box",
```

```
  fitdf = (porder_3 + qorder_3)
)
```

```
##
##  Box-Ljung test
##
## data:  final_model_gb$resid
## X-squared = 1.5177, df = 1, p-value = 0.218
```

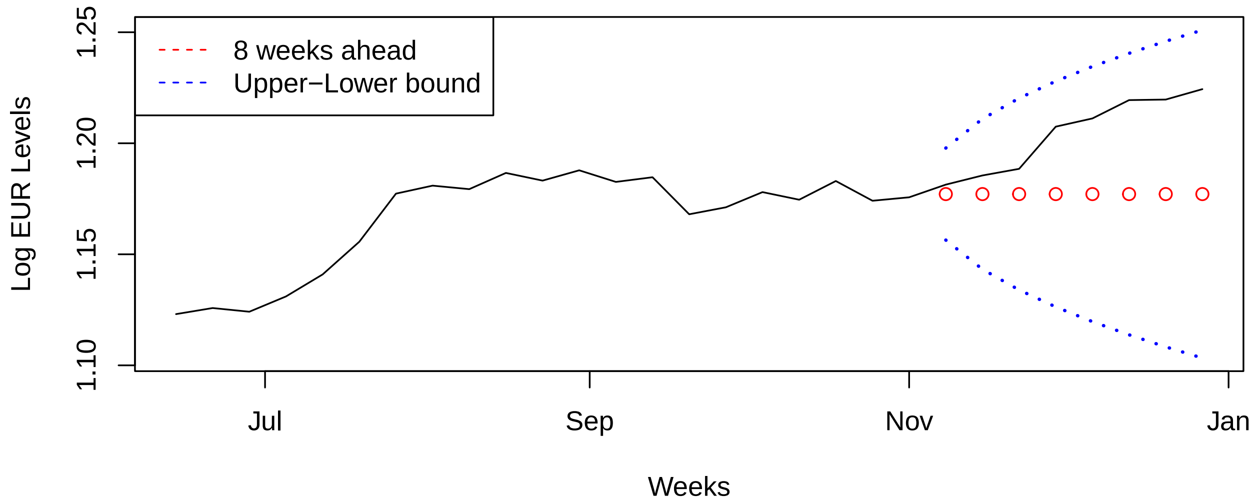
Response: Residual Analysis

The fit for the EUR data seems to be somewhat better fitting as assessed using the residual plots. The ACF and PACF plots resemble those of white noise.

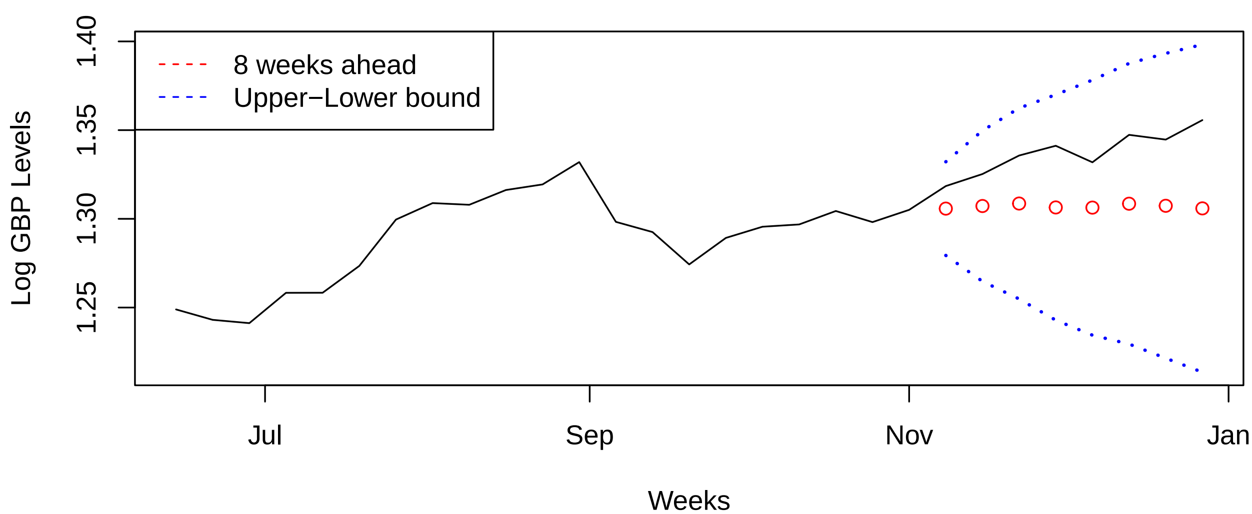
Serial correlation testing results: The null hypothesis is that the residual process consists of uncorrelated variables, which is supported since the p-values are larger than 0.05. In fact the p-values are larger than the significance level 0.1 and hence we would reject at this level also.

**2c.* For each currency exchange, apply the model identified in (2a) and forecast the last eight weeks of data. Plot the predicted data to compare the predicted values to the actual observed ones. Include 90% confidence intervals for the forecasts in the corresponding plots.

```
n <- length(eur.w)
n_fit <- length(eur.w.train)
n_forward <- n - n_fit

# EUR Data
outpred <- predict(final_model, n.ahead = n_forward)
# 90% confidence interval
ubound <- outpred$pred + 1.645 * outpred$se
lbound <- outpred$pred - 1.645 * outpred$se
ymin <- min(lbound)
ymax <- max(ubound)
dates.diff <- index(weekly)
par(mfrow = c(1, 1))
n <- length(eur.w)
plot((dates.diff)[(n - n_forward - 20):n], eur.w[(n - n_forward - 20):n],
     type = "l", ylim = c(ymin, ymax), xlab = "Weeks", ylab = "Log EUR Levels")
points((dates.diff)[(n_fit + 1):n], outpred$pred, col = "red")
lines((dates.diff)[(n_fit + 1):n], ubound,
      lty = 3, lwd = 2, col = "blue")
lines((dates.diff)[(n_fit + 1):n], lbound,
      lty = 3, lwd = 2, col = "blue")
legend('topleft',
  legend = c("8 weeks ahead ", "Upper-Lower bound"),
  lty = 2, col = c("red", "blue"))
```




```r
# GBP Data
outpred <- predict(final_model_gb, n.ahead = n_forward)
# 90% confidence interval
ubound <- outpred$pred + 1.645 * outpred$se
lbound <- outpred$pred - 1.645 * outpred$se
ymin <- min(lbound)
ymax <- max(ubound)
dates.diff <- index(weekly)
par(mfrow = c(1, 1))
n <- length(eur.w)
plot((dates.diff)[(n - n_forward - 20):n], gbp.w[(n - n_forward - 20):n],
     type = "l", ylim = c(ymin, ymax), xlab = "Weeks", ylab = "Log GBP Levels")
points((dates.diff)[(n_fit + 1):n], outpred$pred, col = "red")
lines((dates.diff)[(n_fit + 1):n], ubound,
      lty = 3, lwd = 2, col = "blue")
lines((dates.diff)[(n_fit + 1):n], lbound,
      lty = 3, lwd = 2, col = "blue")
legend('topleft',
  legend = c("8 weeks ahead ", "Upper-Lower bound"),
  lty = 2, col = c("red", "blue"))
```



2d. Calculate Mean Absolute Percentage Error (MAPE) and Precision Measure (PM). How many observations are within the prediction bands? Compare the accuracy of the predictions for the two time series using these two measures.

```
#Computing Accuracy measures

#EUR Data
outpred <- predict(final_model, n.ahead = n_forward)
ubound <- outpred$pred + 1.645 * outpred$se  #confidence interval
lbound <- outpred$pred - 1.645 * outpred$se
consump_true <- as.vector(eur.w[(n_fit + 1):n])
consump_pred <- outpred$pred

print("EUR Stats Summary --------")
```

```
## [1] "EUR Stats Summary --------"
```

```
print("MAPE:")
```

```
## [1] "MAPE:"
```

```
print(mean(abs(consump_pred - consump_true) / consump_true))
```

```
## [1] 0.02271062
```

```
print("PM:")
```

```
## [1] "PM:"
```

```
print(sum((consump_pred - consump_true) ^ 2) / sum((consump_true - mean(consump_true)) ^ 2))
```

```
## [1] 3.960071
```

```
print("Does the observed data fall outside the prediction intervals?")
```

```
## [1] "Does the observed data fall outside the prediction intervals?"
```

```
print(sum(consump_true < lbound) & sum(consump_true > ubound))
```

```
## [1] FALSE
```

```
#GBP Data
outpred <- predict(final_model_gb, n.ahead = n_forward)
ubound <- outpred$pred + 1.645 * outpred$se  #confidence interval
lbound <- outpred$pred - 1.645 * outpred$se
consump_true <- as.vector(gbp.w[(n_fit + 1):n])
consump_pred <- outpred$pred

print("GBP Stats Summary --------")
```

```
## [1] "GBP Stats Summary --------"
```

```
print("MAPE:")
```

```
## [1] "MAPE:"
```

```
print(mean(abs(consump_pred - consump_true) / consump_true))
```

```
## [1] 0.02272969
```

```
print("PM:")
```

```
## [1] "PM:"
```

```r
print(sum((consump_pred - consump_true) ^ 2) / sum((consump_true - mean(consump_true)) ^ 2))
```

```
## [1] 8.136184
```

```r
print("Does the observed data fall outside the prediction intervals?")
```

```
## [1] "Does the observed data fall outside the prediction intervals?"
```

```r
print(sum(consump_true < lbound) & sum(consump_true > ubound))
```

```
## [1] FALSE
```

Response: Prediction Accuracy

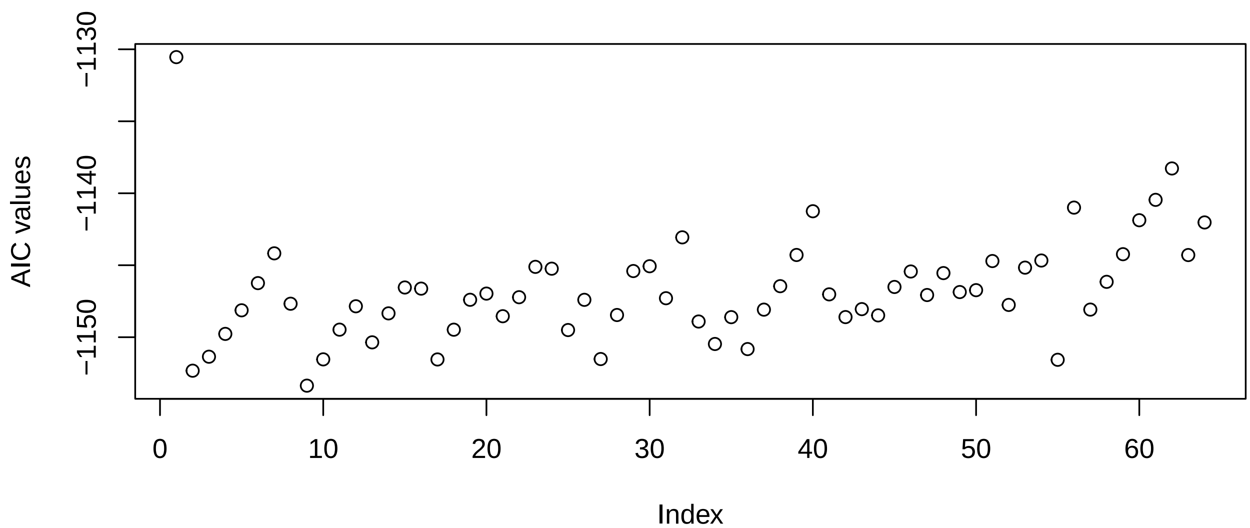
All the observations are within the prediction bands for both the models. However, the model for EUR seems to be a better fit since both the MAPE and the PM measures are lower. However, both models do not seem to provide good predictions since the variability in the predictions is much higher than the variability in the data (i.e. very high PM values).

Question 3. ARIMA Fitting: Monthly Data Analysis (17 points)

3a. Divide the data into training and testing data set, where the training data exclude the last two months of data (November and December 2020) with the testing data including the last two months. For both currency exchange rates and using the training datasets, use the iterative model to fit an ARIMA(p,d,q) model with max AR and MA orders of 8, and a differencing order of 1 or 2. Display the summary of the final model fit. Compare statistical significance of the coefficients. Compare the order selection from using monthly versus weekly data for each of the two currencies.

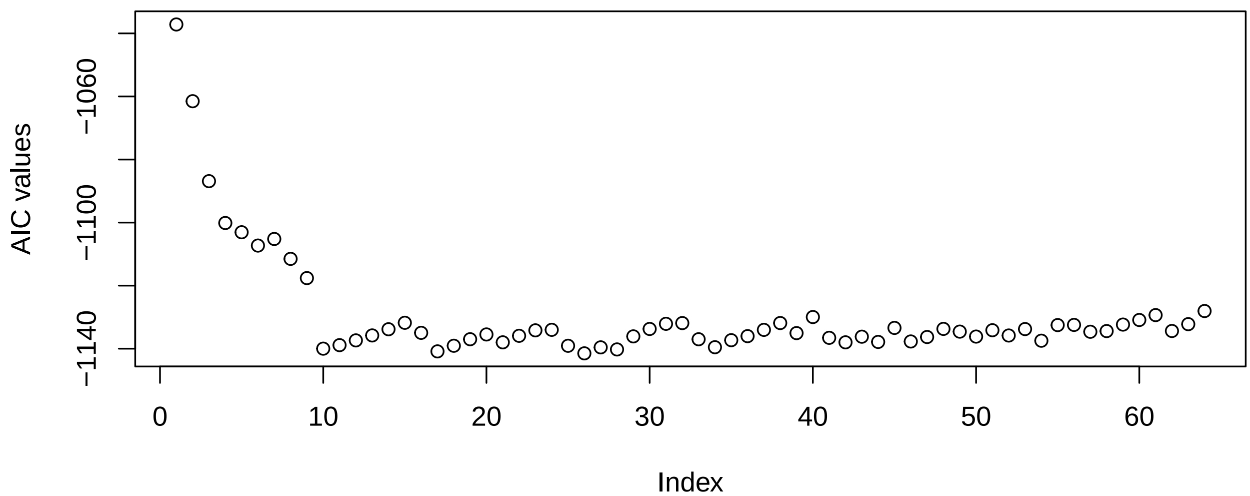
```r
num <- 2

# set up the training and testing data

eur.m <- ts_monthly[, "EUR"]
gbp.m <- ts_monthly[, "GBP"]

eur.m.train <- eur.m[1:(length(eur.m) - num)]
eur.m.test <- eur.m[(length(eur.m) - num + 1):length(eur.m)]

gbp.m.train <- gbp.m[1:(length(gbp.m) - num)]
gbp.m.test <- gbp.m[(length(gbp.m) - num + 1):length(gbp.m)]


## EUR Currency Analysis -----

## 1. Differencing Order = 1

n <- length(eur.m.train)
norder <- 8

p <- 1:norder - 1
q <- 1:norder - 1
aic <- matrix(0, norder, norder)
for (i in 1:norder) {
  for (j in 1:norder) {
    modij = arima(eur.m.train,
                  order = c(p[i], 1, q[j]),
```

```r
                  method = 'ML')
    aic[i, j] = modij$aic - 2 * (p[i] + q[j] + 1) + 2 * (p[i] + q[j] + 1) *
      n / (n - p[i] - q[j] - 2)

  }
}

# Extract the "best" one according to AIC
aicv <- as.vector(aic)
plot(aicv, ylab = "AIC values")
```



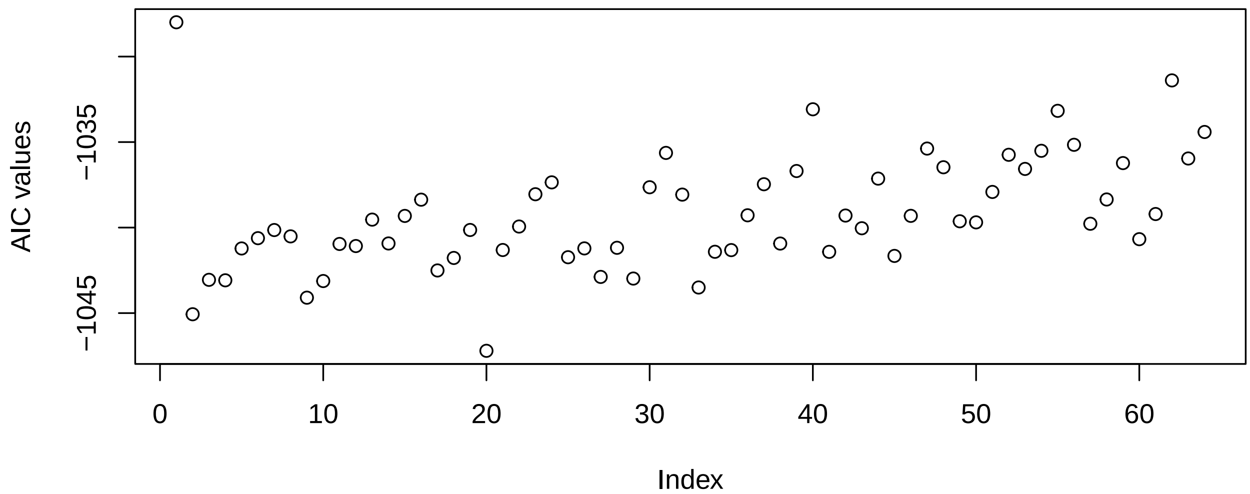

```r
indexp <- rep(c(1:norder), norder)
indexq <- rep(c(1:norder), each = norder)
indexaic <- which(aicv == min(aicv))
porder_5 <- indexp[indexaic] - 1
qorder_5 <- indexq[indexaic] - 1

# Fit the "best" arima model
m_final_model <-
  arima(eur.m.train,
        order = c(porder_5, 1, qorder_5),
        method = "ML")

# 2. Differencing Order = 2
n = length(eur.m.train)
norder = 8
p = c(1:norder) - 1
q = c(1:norder) - 1
aic = matrix(0, norder, norder)
for (i in 1:norder) {
  for (j in 1:norder) {
    modij = arima(eur.m.train,
                  order = c(p[i], 2, q[j]),
                  method = 'ML')
    aic[i, j] = modij$aic - 2 * (p[i] + q[j] + 1) + 2 * (p[i] + q[j] + 1) *
      n / (n - p[i] - q[j] - 2)
```

```
  }
}

# Extract the "best" one according to AIC
aicv <- as.vector(aic)
plot(aicv, ylab = "AIC values")
```

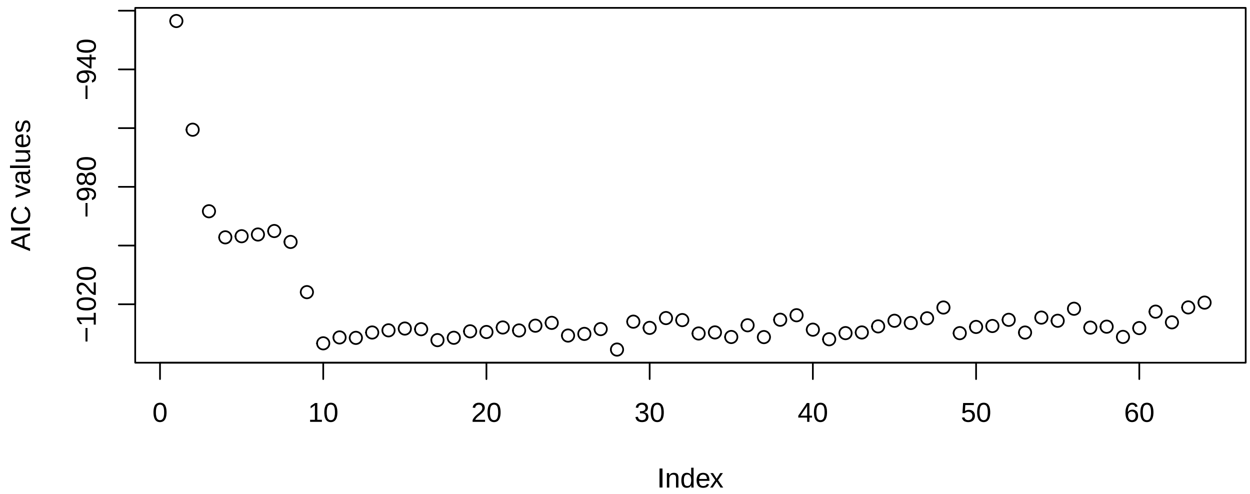



```
indexp <- rep(c(1:norder), norder)
indexq <- rep(c(1:norder), each = norder)
indexaic <- which(aicv == min(aicv))
porder_6 <- indexp[indexaic] - 1
qorder_6 <- indexq[indexaic] - 1
# Fit the "best" arima model
m_final_model2 <- arima(eur.m.train,
                        order = c(porder_6, 2, qorder_6),
                        method = "ML")


## GBP Currency Analysis -----

## 1. Differencing Order = 1

n <- length(eur.m.train)
norder <- 8

p <- 1:norder - 1
q <- 1:norder - 1
aic <- matrix(0, norder, norder)
for (i in 1:norder) {
  for (j in 1:norder) {
    modij <- arima(gbp.m.train,
                   order = c(p[i], 1, q[j]),
                   method = 'ML')
    aic[i, j] <-
      modij$aic - 2 * (p[i] + q[j] + 1) + 2 * (p[i] + q[j] + 1) * n / (n - p[i] - q[j] - 2)
  }
}
```

```r
# Extract the "best" one according to AIC
aicv <- as.vector(aic)
plot(aicv, ylab = "AIC values")
```




```r
indexp <- rep(c(1:norder), norder)
indexq <- rep(c(1:norder), each = norder)
indexaic <- which(aicv == min(aicv))
porder_7 <- indexp[indexaic] - 1
qorder_7 <- indexq[indexaic] - 1

# Fit the "best" arima model
m_final_model_gb <- arima(gbp.m.train,
                          order = c(porder_7, 1, qorder_7),
                          method = "ML")

# Differencing Order = 2
n <- length(eur.m.train)
norder <- 8

p <- 1:norder - 1
q <- 1:norder - 1
aic <- matrix(0, norder, norder)
for (i in 1:norder) {
  for (j in 1:norder) {
    modij <- arima(gbp.m.train,
                   order = c(p[i], 2, q[j]),
                   method = 'ML')
    aic[i, j] <-
      modij$aic - 2 * (p[i] + q[j] + 1) + 2 * (p[i] + q[j] + 1) * n / (n - p[i] -
                                                                       q[j] - 2)

  }
}

# Extract the "best" one according to AIC
aicv <- as.vector(aic)
plot(aicv, ylab = "AIC values")
```




```
indexp <- rep(c(1:norder), norder)
indexq <- rep(c(1:norder), each = norder)
indexaic <- which(aicv == min(aicv))
porder_8 <- indexp[indexaic] - 1
qorder_8 <- indexq[indexaic] - 1

# Fit the "best" arima model
m_final_model_gb2 <- arima(gbp.m.train,
                           order = c(porder_8, 2, qorder_8),
                           method = "ML")

### AIC Summary ----

print("EUR Model Selection:")
```

```
## [1] "EUR Model Selection:"
```

```
print(paste0("ARIMA: (", porder_5, ", 1, ", qorder_5,
             ") with AICc = ", round(m_final_model$aic, 3)))
```

```
## [1] "ARIMA: (0, 1, 1) with AICc = -1153.413"
```

```
print(paste0("ARIMA: (", porder_6, ", 2, ", qorder_6,
             ") with AICc = ", round(m_final_model2$aic, 3)))
```

```
## [1] "ARIMA: (1, 2, 3) with AICc = -1141.715"
```

```
print("GBP Model Selection:")
```

```
## [1] "GBP Model Selection:"
```

```
print(paste0("ARIMA: (", porder_7, ", 1, ", qorder_7,
             ") with AICc = ", round(m_final_model_gb$aic, 3)))
```

```
## [1] "ARIMA: (3, 1, 2) with AICc = -1047.534"
```

```
print(paste0("ARIMA: (", porder_8, ", 2, ", qorder_8,
             ") with AICc = ", round(m_final_model_gb2$aic, 3)))
```

```
## [1] "ARIMA: (3, 2, 3) with AICc = -1035.878"
```

```r
## compute the test statistics
tv.eur.monthly <-
  as.numeric(m_final_model$coef) / as.numeric(sqrt(diag(m_final_model$var.coef)))
tv.gdp.monthly <-
  as.numeric(m_final_model_gb$coef) / as.numeric(sqrt(diag(m_final_model_gb$var.coef)))

## Apply the pvalue.coef function
pvalues.eur.monthly <- sapply(tv.eur.monthly, pvalue.coef)
pvalues.gdp.monthly <- sapply(tv.gdp.monthly, pvalue.coef)

print("EUR T Values:")
```

```
## [1] "EUR T Values:"
```

```r
print(tv.eur.monthly)
```

```
## [1] 5.432135
```

```r
print("GBP T Values:")
```

```
## [1] "GBP T Values:"
```

```r
print(tv.gdp.monthly)
```

```
## [1]  16.495871 -23.990616   4.959424 -52.637611  59.992604
```

```r
print("EUR P Values:")
```

```
## [1] "EUR P Values:"
```

```r
print(pvalues.eur.monthly)
```

```
## [1] 5.568368e-08
```

```r
print("GBP P Values:")
```

```
## [1] "GBP P Values:"
```

```r
print(pvalues.gdp.monthly)
```

```
## [1] 0.00000e+00 0.00000e+00 7.07024e-07 0.00000e+00 0.00000e+00
```

Response: Analysis of the ARIMA Fit for the Weekly and Monthly Data

The selected models for the weekly data are as follows:

- EUR Model Selection: ARIMA(0,1,1) with AICc = -1153.413 vs ARIMA(1,2,3) with AICc = -1141.715, where the less complex model ARIMA(0,1,1) model has a smaller AICc value.
- GDP Model Selection: ARIMA(3,1,2) with AICc = -1047.534 vs ARIMA(3,2,3) with AICc = -1035.878, with the ARIMA(3,1,2) having a lower AICc value hence selected.

In order to evaluate whether a reduced order for AR or MA will result in a less complex model but with similar explanatory power, we can evaluate the p-value of the z-test for statistical significance of the coefficients corresponding to the highest AR or MA order. Applying a similar procedure as in previous question, we find that all p-values are approximately zero, suggesting that the models selected should not be reduced further.

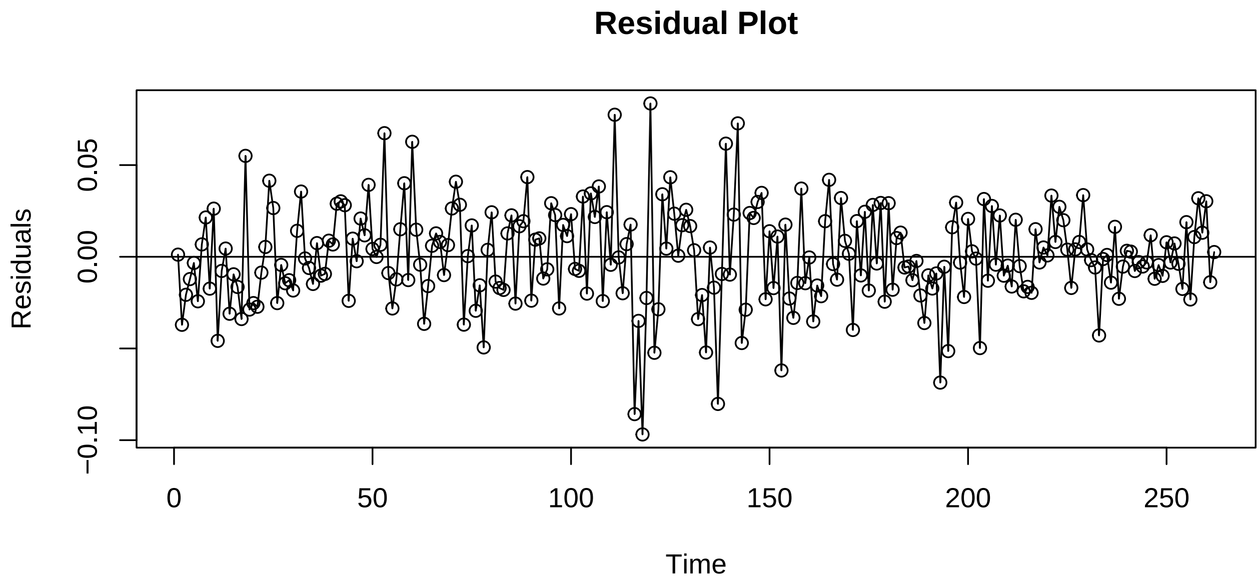
Response: Monthly vs Weekly Data

- EUR: Monthly model ARIMA(0,1,1) vs Weekly model ARIMA(0,1,1)
- GDP Monthly model ARIMA(3,1,2) vs Weekly model ARIMA(2,1,3)

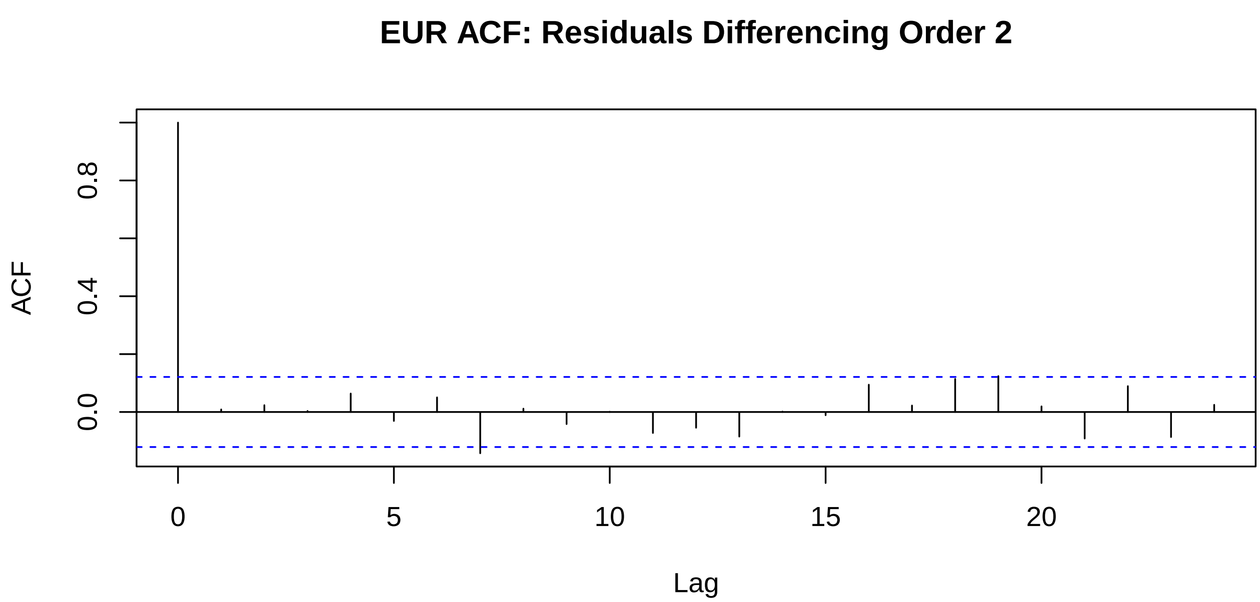
Hence the models are the same for EUR currency but slightly different for GDP currency although of similar complexity.

Below we are performing the residual analysis for the two selected models although it was not required as part of the analysis of thr weekly data model! Please don't drop points if not provided.

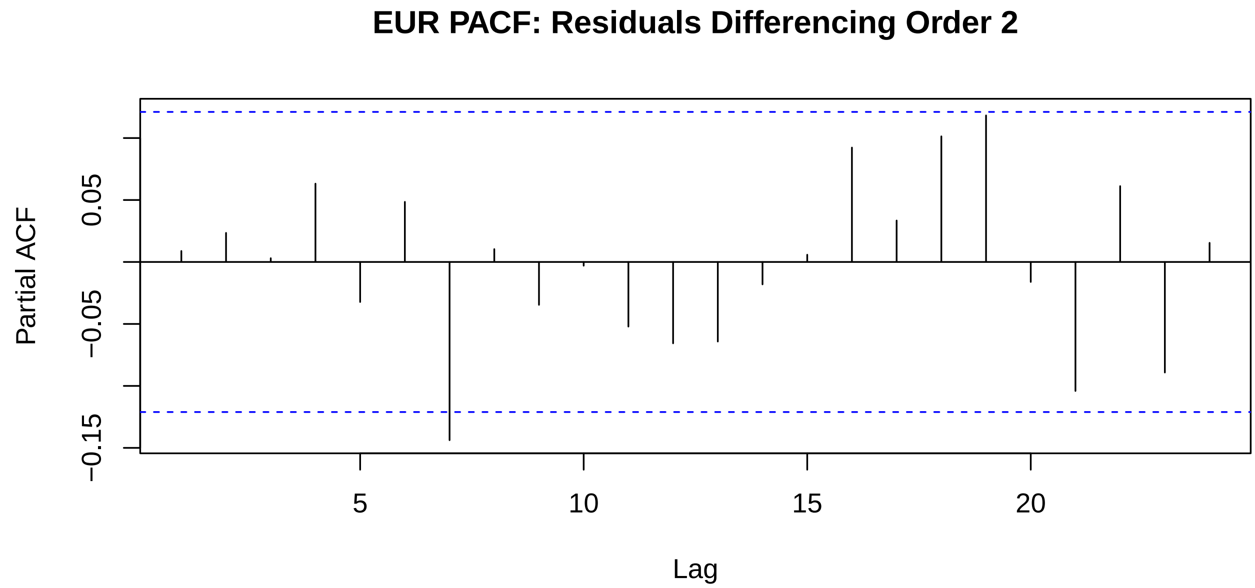
```
resids <- resid(m_final_model)
plot(resids,
     ylab = 'Residuals',
     type = 'o',
     main = "Residual Plot")
abline(h = 0)
```



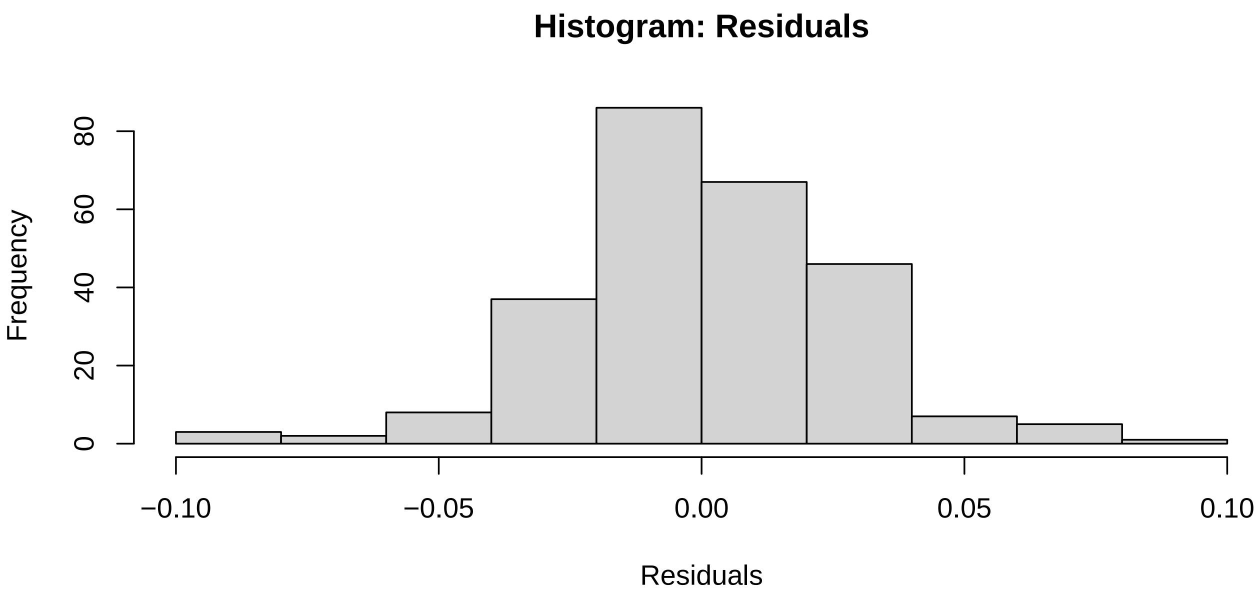

```
acf(resids, main = "EUR ACF: Residuals Differencing Order 2")
```



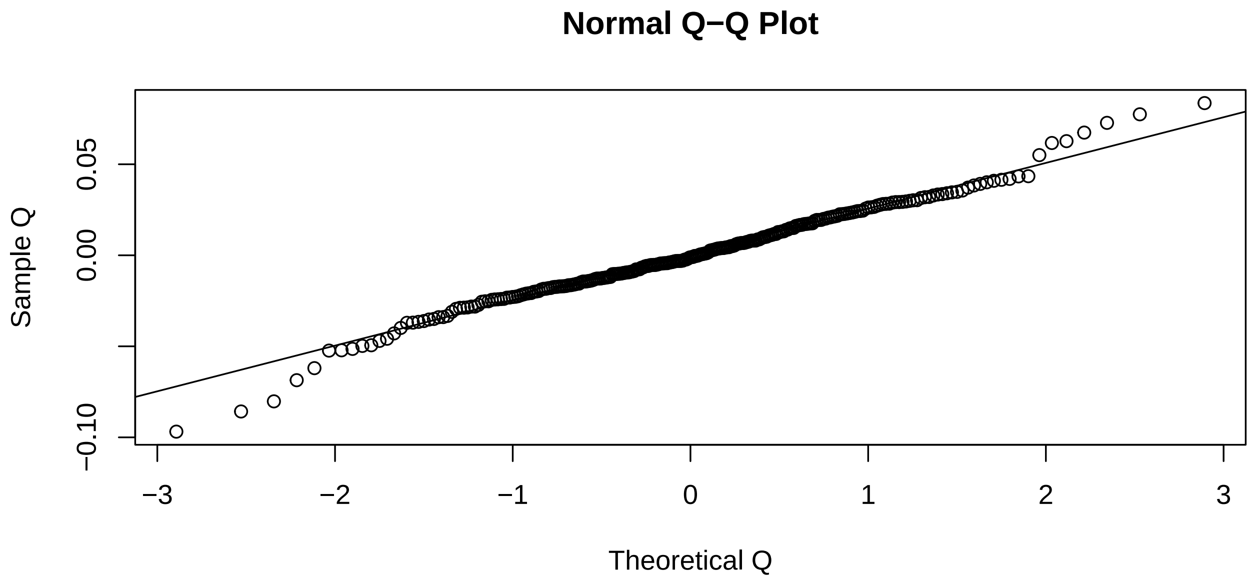
```r
pacf(resids, main = "EUR PACF: Residuals Differencing Order 2")
```




```r
hist(resids, xlab = 'Residuals', main = 'Histogram: Residuals')
```




```r
qqnorm(resids, ylab = "Sample Q", xlab = "Theoretical Q")
qqline(resids)
```



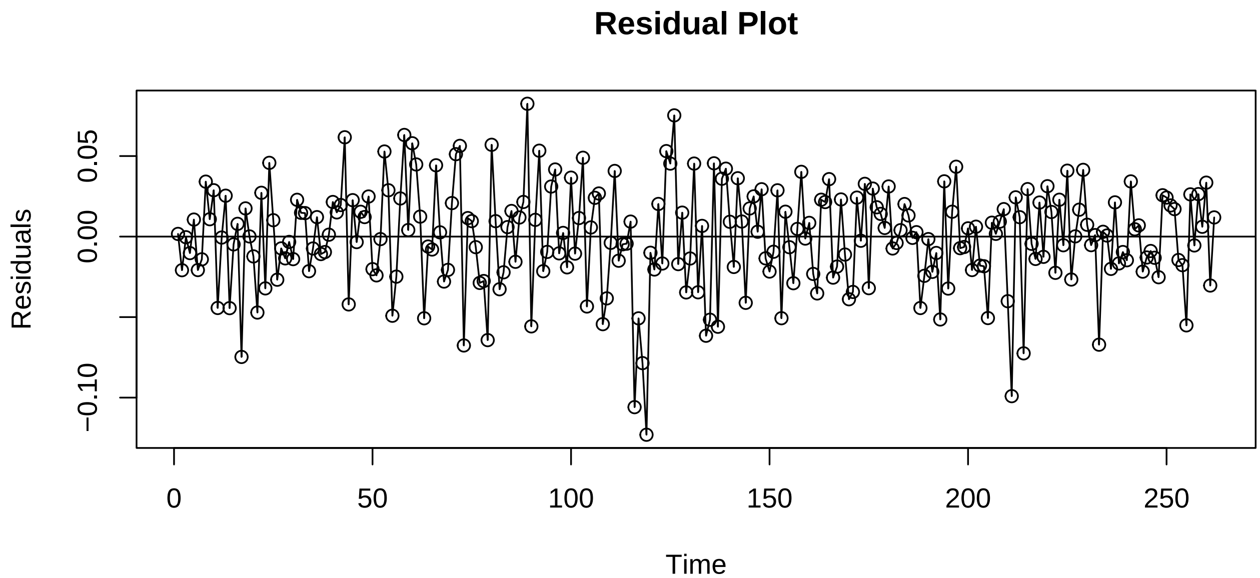

```
Box.test(
  m_final_model$resid,
  lag = (porder_5 + qorder_5 + 1),
  type = "Box-Pierce",
  fitdf = (porder_5 + qorder_5)
)
```

```
##
##  Box-Pierce test
##
## data:  m_final_model$resid
## X-squared = 0.16616, df = 1, p-value = 0.6836
```

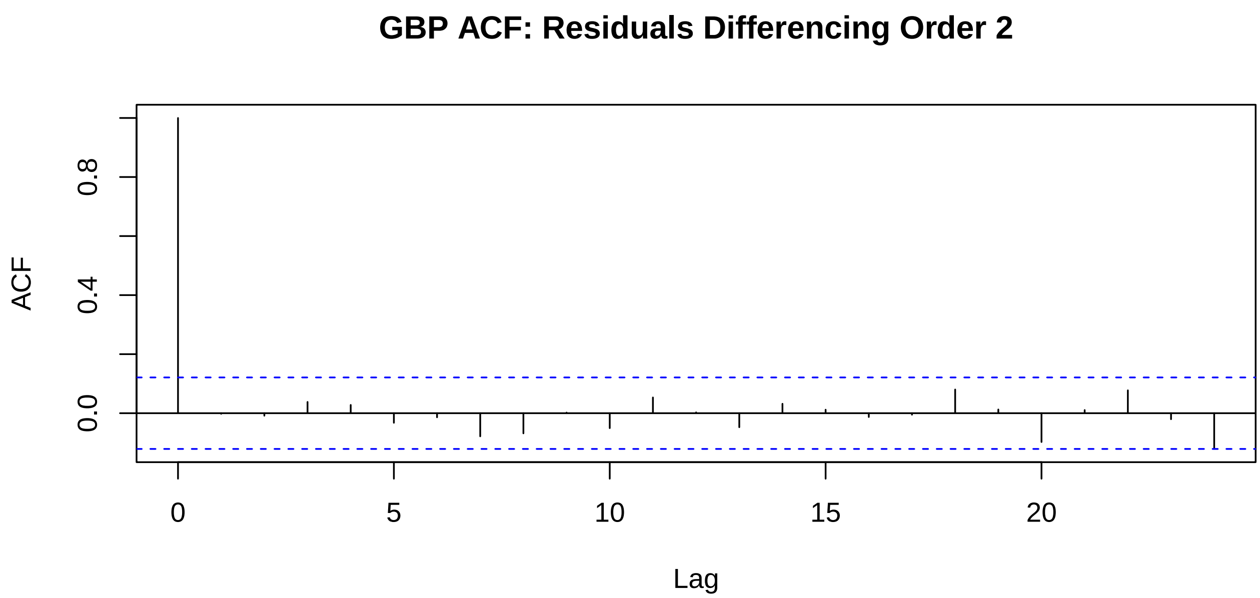
```
Box.test(
  m_final_model$resid,
  lag = (porder_5 + qorder_5 + 1),
  type = "Ljung-Box",
  fitdf = (porder_5 + qorder_5)
)
```

```
##
##  Box-Ljung test
##
## data:  m_final_model$resid
## X-squared = 0.16863, df = 1, p-value = 0.6813
```

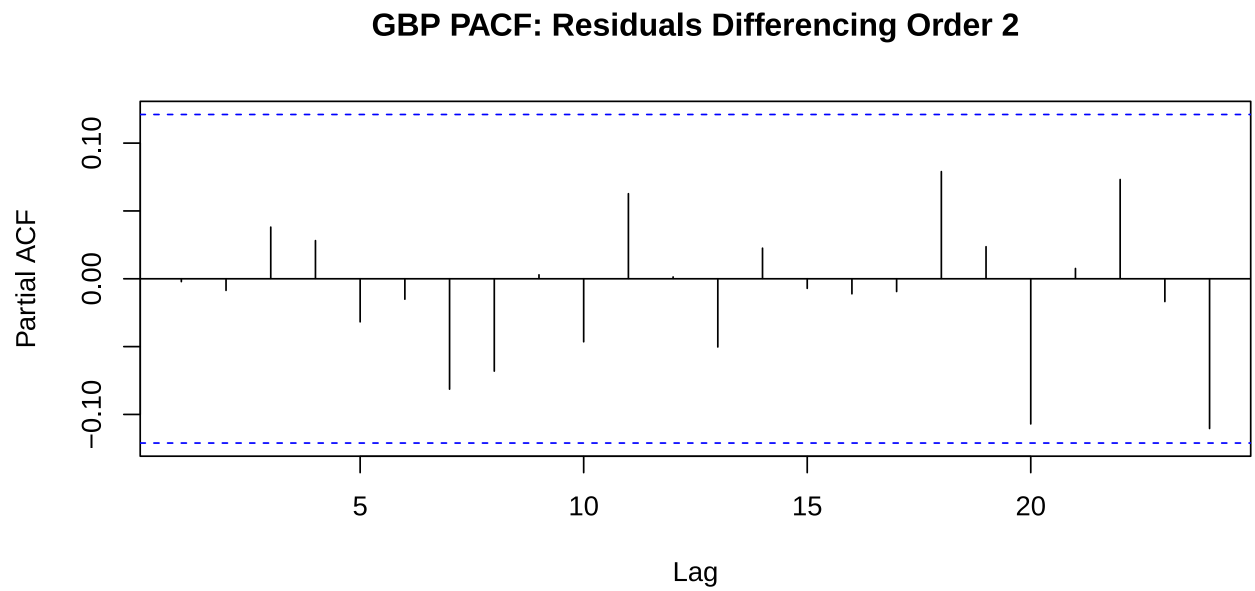
```
resids <- resid(m_final_model_gb)
plot(resids,
     ylab = 'Residuals',
     type = 'o',
     main = "Residual Plot")
abline(h = 0)
```



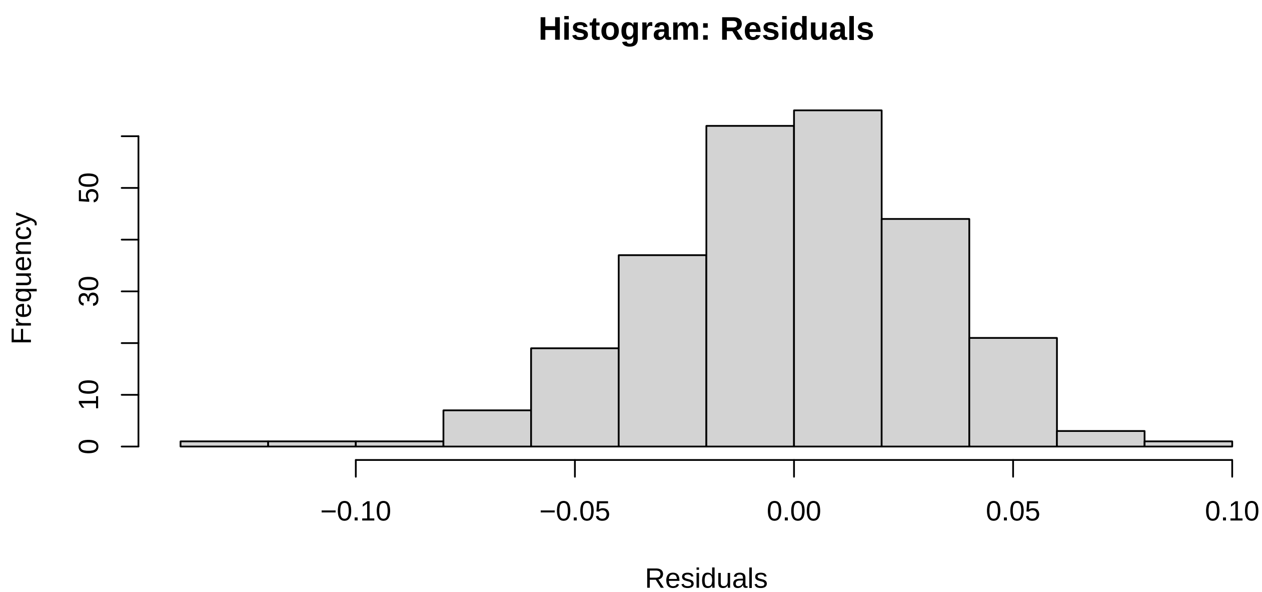

```
acf(resids, main = "GBP ACF: Residuals Differencing Order 2")
```



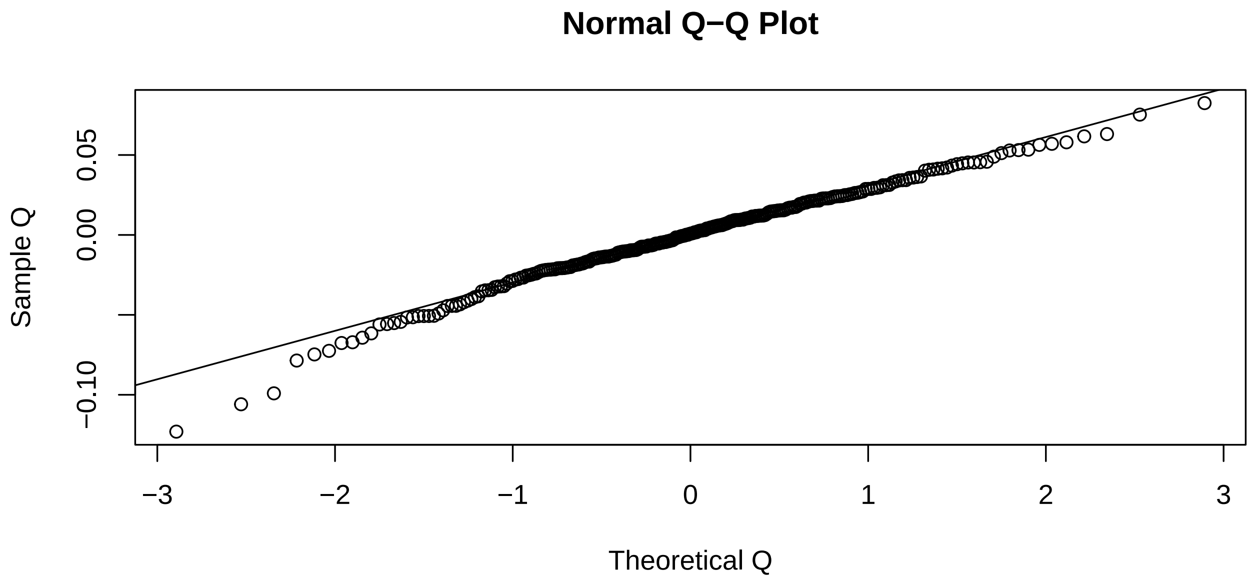

```
pacf(resids, main = "GBP PACF: Residuals Differencing Order 2")
```




```r
hist(resids, xlab = 'Residuals', main = 'Histogram: Residuals')
```




```r
qqnorm(resids, ylab = "Sample Q", xlab = "Theoretical Q")
qqline(resids)
```




```
Box.test(
  m_final_model_gb$resid,
  lag = (porder_7 + qorder_7 + 1),
  type = "Box-Pierce",
  fitdf = (porder_7 + qorder_7)
)
```

```
##
##  Box-Pierce test
##
## data:  m_final_model_gb$resid
## X-squared = 0.93255, df = 1, p-value = 0.3342
```

```
Box.test(
  m_final_model_gb$resid,
  lag = (porder_7 + qorder_7 + 1),
  type = "Ljung-Box",
  fitdf = (porder_7 + qorder_7)
)
```

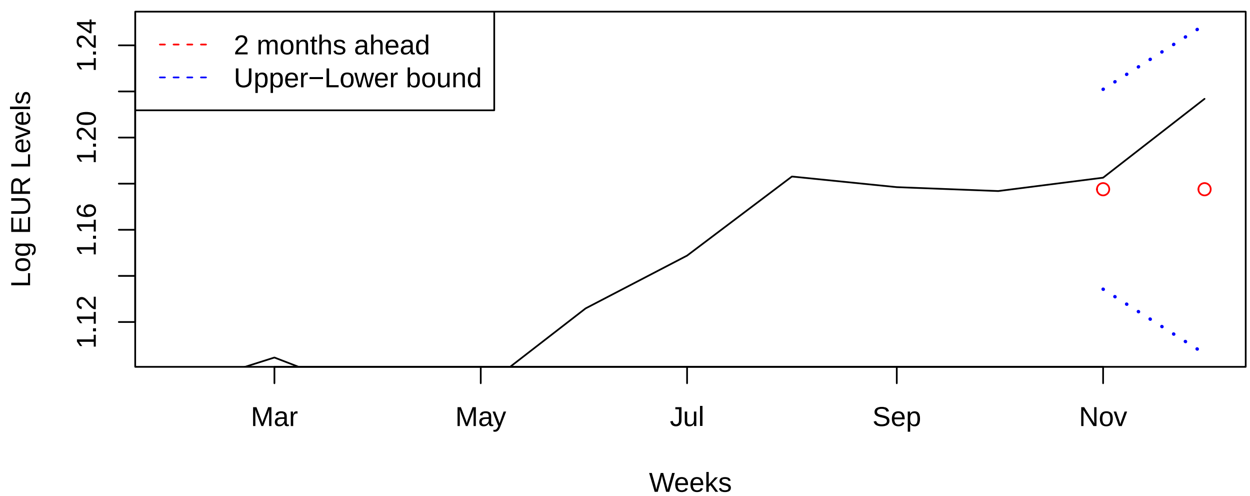
```
##
##  Box-Ljung test
##
## data:  m_final_model_gb$resid
## X-squared = 0.95408, df = 1, p-value = 0.3287
```

Residual analysis

The fit for the EUR residuals data seems to be somewhat better fitting from the residual analysis. The null hypothesis is that the residual process consists of uncorrelated variables, which is not rejected since the the p-values are large. Hence it palusbile that the residuals to follow the distribution assumptions hence the models are a good fit.

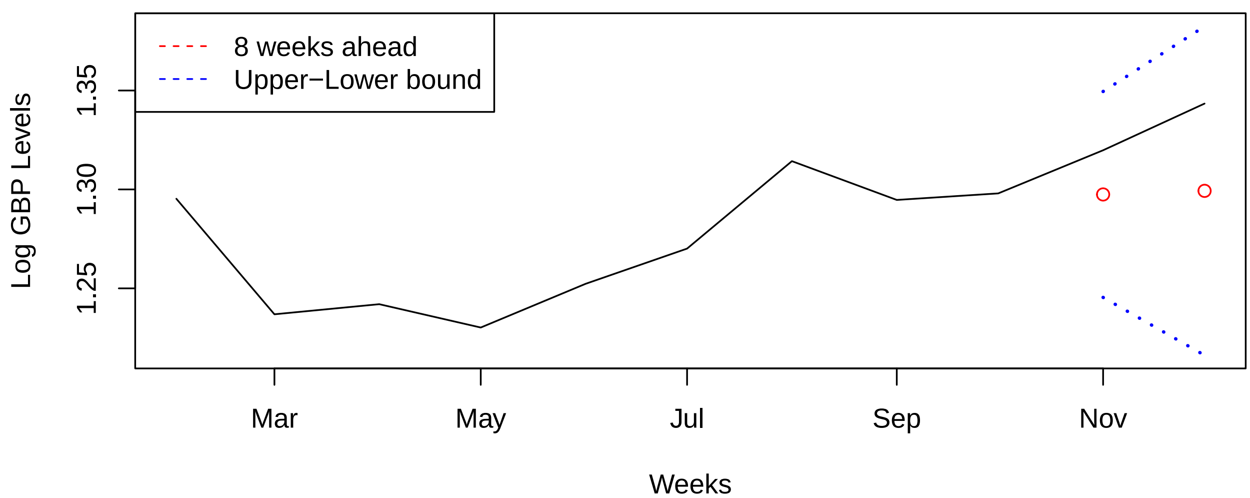
3c. For each currency exchange, apply the model identified in (3a) and forecast the last two months of data. Plot the predicted data to compare the predicted values to the actual observed ones. Include 90% confidence intervals for the forecasts in the corresponding plots.

```r
n <- length(eur.m)
n_fit <- length(eur.m.train)
n_forward <- n - n_fit

outpred <- predict(m_final_model, n.ahead = n_forward)
ubound <- outpred$pred + 1.645 * outpred$se  #confidence interval
lbound <- outpred$pred - 1.645 * outpred$se
ymin <- min(lbound)
ymax <- max(ubound)
dates.diff <- index(monthly)
n <- length(eur.m)
plot((dates.diff)[(n - n_forward - 8):n],
     eur.m[(n - n_forward - 8):n],
     type = "l",
     ylim = c(ymin, ymax),
     xlab = "Weeks",
     ylab = "Log EUR Levels"
)
points((dates.diff)[(n_fit + 1):n], outpred$pred, col = "red")
lines((dates.diff)[(n_fit + 1):n],
      ubound,
      lty = 3,
      lwd = 2,
      col = "blue")
lines((dates.diff)[(n_fit + 1):n],
      lbound,
      lty = 3,
      lwd = 2,
      col = "blue")
legend(
  'topleft',
  legend = c("2 months ahead ", "Upper-Lower bound"),
  lty = 2,
  col = c("red", "blue")
)
```




```r
# GBP Data
outpred <- predict(m_final_model_gb, n.ahead = n_forward)
```

```r
ubound <- outpred$pred + 1.645 * outpred$se  #confidence interval
lbound <- outpred$pred - 1.645 * outpred$se
ymin <- min(lbound)
ymax <- max(ubound)
dates.diff <- index(monthly)
n <- length(gbp.m)
plot((dates.diff)[(n - n_forward - 8):n],
     gbp.m[(n - n_forward - 8):n],
     type = "l",
     ylim = c(ymin, ymax),
     xlab = "Weeks",
     ylab = "Log GBP Levels"
)
points((dates.diff)[(n_fit + 1):n], outpred$pred, col = "red")
lines((dates.diff)[(n_fit + 1):n],
      ubound,
      lty = 3,
      lwd = 2,
      col = "blue")
lines((dates.diff)[(n_fit + 1):n],
      lbound,
      lty = 3,
      lwd = 2,
      col = "blue")
legend(
  'topleft',
  legend = c("8 weeks ahead ", "Upper-Lower bound"),
  lty = 2,
  col = c("red", "blue")
)
```



3d. Calculate Mean Absolute Percentage Error (MAPE) and Precision Measure (PM). How many observations are within the prediction bands? Compare the accuracy of the predictions for the two time series using these two measures.

```r
#EUR Data
outpred <- predict(m_final_model, n.ahead = n_forward)
ubound <- outpred$pred + 1.645 * outpred$se  #confidence interval
```

```r
lbound <- outpred$pred - 1.645 * outpred$se
consump_true <- as.vector(eur.m[(n_fit + 1):n])
consump_pred <- outpred$pred

print("EUR Stats Summary --------")
```

```
## [1] "EUR Stats Summary --------"
```

```r
print("MAPE:")
```

```
## [1] "MAPE:"
```

```r
print(mean(abs(consump_pred - consump_true) / consump_true))
```

```
## [1] 0.01823317
```

```r
print("PM:")
```

```
## [1] "PM:"
```

```r
print(sum((consump_pred - consump_true) ^ 2) / sum((consump_true - mean(consump_true)) ^ 2))
```

```
## [1] 2.672352
```

```r
print("Does the observed data fall outside the prediction intervals?")
```

```
## [1] "Does the observed data fall outside the prediction intervals?"
```

```r
print(sum(consump_true < lbound) & sum(consump_true > ubound))
```

```
## [1] FALSE
```

```r
#GBP Data
outpred <- predict(m_final_model_gb, n.ahead = n_forward)
ubound <- outpred$pred + 1.645 * outpred$se  #confidence interval
lbound <- outpred$pred - 1.645 * outpred$se
consump_true <- as.vector(gbp.m[(n_fit + 1):n])
consump_pred <- outpred$pred

print("GBP Stats Summary --------")
```

```
## [1] "GBP Stats Summary --------"
```

```r
print("MAPE:")
```

```
## [1] "MAPE:"
```

```r
print(mean(abs(consump_pred - consump_true) / consump_true))
```

```
## [1] 0.0248779
```

```r
print("PM:")
```

```
## [1] "PM:"
```

```r
print(sum((consump_pred - consump_true) ^ 2) / sum((consump_true - mean(consump_true)) ^ 2))
```

```
## [1] 8.77781
```

```r
print("Does the observed data fall outside the prediction intervals?")
```

```
## [1] "Does the observed data fall outside the prediction intervals?"
```

```
print(sum(consump_true < lbound) & sum(consump_true > ubound))
```

```
## [1] FALSE
```

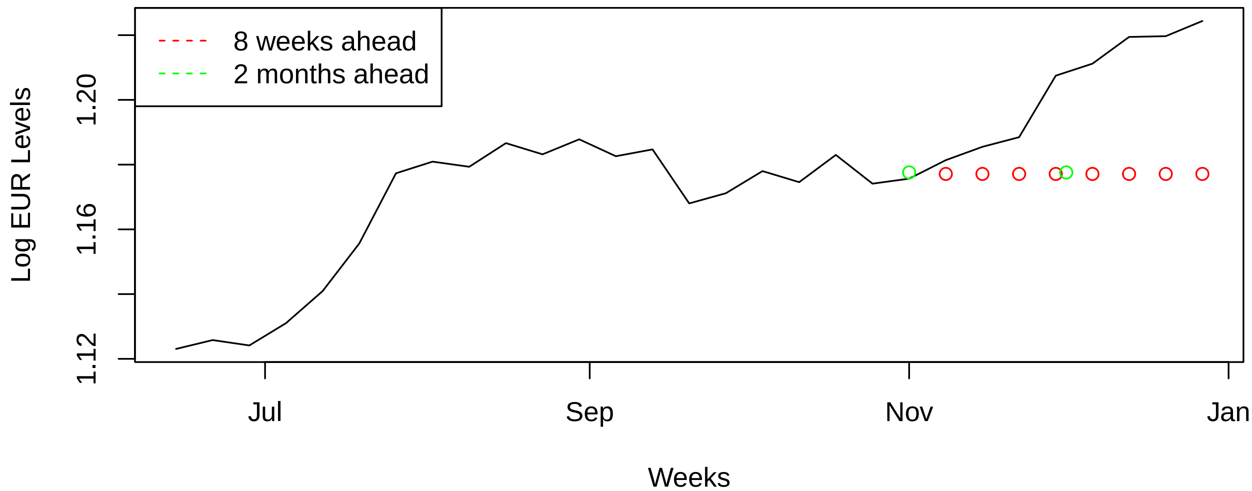
Response: Predictions

All the observations are withing the prediction bands for both the models. However, the model for EUR seems to be a better fit based on the the MAPE and PM measures, but both models do not seem to provide good predictions due to a somewhat high PM.

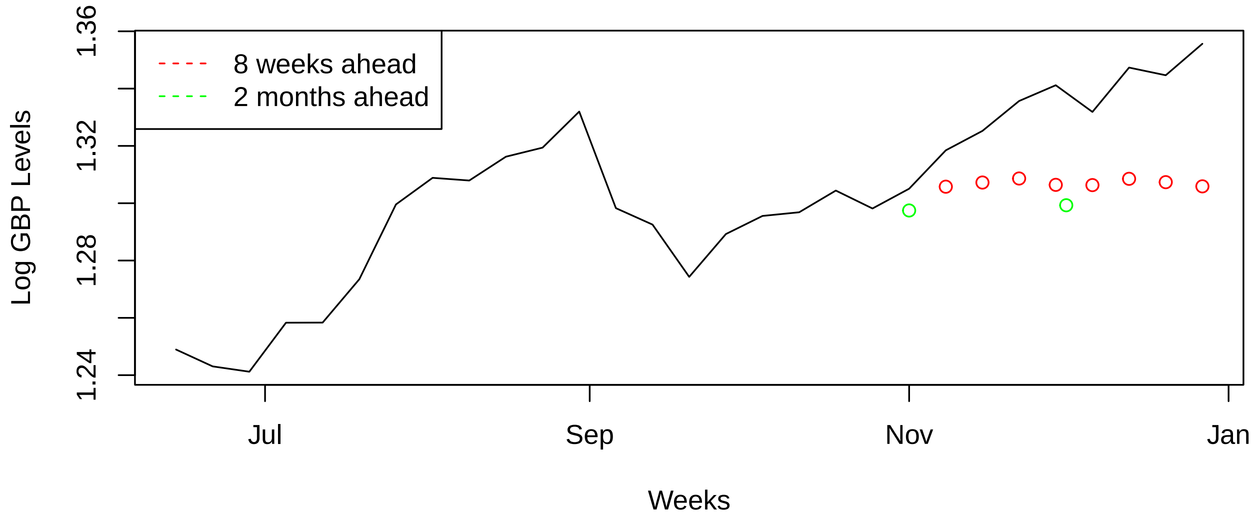
Question 4. Weekly vs Monthly Forecasting (5 points)

Compare the forecasts based on the weekly versus monthly data. Overlay the forecast into one single plot for each of the two currency exchange rates. What can you say about using weekly versus monthly data?

```
#EUR
n <- length(eur.w)
n_fit <- length(eur.w.train)
n_forward <- n - n_fit
outpred <- predict(final_model, n.ahead = n_forward)
ubound <- outpred$pred + 1.645 * outpred$se  #confidence interval
lbound <- outpred$pred - 1.645 * outpred$se
ymin <- min(lbound)
ymax <- max(ubound)
dates.diff <- index(weekly)

#monthly
mn <- length(eur.m)
mn_fit <- length(eur.m.train)
mn_forward <- mn - mn_fit
moutpred <- predict(m_final_model, n.ahead = mn_forward)
mdates.diff <- index(monthly)
plot((dates.diff)[(n - n_forward - 20):n],
     eur.w[(n - n_forward - 20):n],
     type = "l",
     xlab = "Weeks",
     ylab = "Log EUR Levels")
points((dates.diff)[(n_fit + 1):n], outpred$pred, col = "red")
points((mdates.diff)[(mn_fit + 1):mn], moutpred$pred, col = "green")
legend(
  'topleft',
  legend = c("8 weeks ahead ", "2 months ahead"),
  lty = 2,
  col = c("red", "green")
)
```




```r
# GBP
n <- length(gbp.w)
n_fit <- length(gbp.w.train)
n_forward <- n - n_fit
outpred <- predict(final_model_gb, n.ahead = n_forward)
ubound <- outpred$pred + 1.645 * outpred$se  #confidence interval
lbound <- outpred$pred - 1.645 * outpred$se
dates.diff <- index(weekly)
#monthly
mn <- length(gbp.m)
mn_fit <- length(gbp.m.train)
mn_forward <- mn - mn_fit
moutpred <- predict(m_final_model_gb, n.ahead = mn_forward)
mubound <- moutpred$pred + 1.645 * moutpred$se  #confidence interval
mlbound <- moutpred$pred - 1.645 * moutpred$se
mdates.diff <- index(monthly)
plot((dates.diff)[(n - n_forward - 20):n],
     gbp.w[(n - n_forward - 20):n],
     type = "l",
     xlab = "Weeks",
     ylab = "Log GBP Levels")
points((dates.diff)[(n_fit + 1):n], outpred$pred, col = "red")
points((mdates.diff)[(mn_fit + 1):mn], moutpred$pred, col = "green")
legend(
  'topleft',
  legend = c("8 weeks ahead ", "2 months ahead"),
  lty = 2,
  col = c("red", "green")
)
```



Response: Prediction Comparison

For the EUR currency, the predictions provided by monthly and weekly data overlap, hence providing similar predictions. For the GDP, the predictions are similar but the monthly predictions slightly under-predict.

Question 5. Reflection on ARIMA (5 points)

Considering your understanding of the ARIMA model in general as well as what your understanding of the behavior of the currency exchange data based on the completion of the above questions, how would you personally regard the effectiveness of ARIMA modelling? Where would it be appropriate to use it for forecasting and where would you recommend against? What are some specific points of caution one would need to consider when considering using it?

Response: Take Home Points

ARIMA can be a useful tool to analyze time series which can be detrended by differencing. The ARIMA fit the two time series appropriately for both currencies. While the models are good fit, the predictions are poorly capturing the increasing trend. This suggests that a model that would first fit and predict the trend would perform better than the ARIMA model. We also learn that for such time series, we only can perform short term predictions.

The (simple) ARIMA model has other limitations, for example, doesn't include external factors influencing the variation in the time series, which can be accounted in the extension of ARIMA, called ARIMAX.

Note: The response to this question will vary from one student to another!