IBEX Assignment 3

Vikas Agarwal, Camille Blain-Coallier, Giulio De Felice, Nayla Fakhoury, Alejandro Koury, Federico Loguercio, Victor Vu

03/06/2019

```
rm(list = ls())

local({
    r <- getOption("repos")
    r["CRAN"] <- "http://cran.rediris.es/"
    options(repos = r)})

install.packages("psych")
```

```
## Installing package into 'C:/Users/Camille Blain-Coalli/Documents/R/win-library/3.5'
## (as 'lib' is unspecified)
```

```
## package 'psych' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Camille Blain-Coalli\AppData\Local\Temp\RtmpqmAfqM\downloaded_packages
```

```
library("psych")
```

```
## Warning: package 'psych' was built under R version 3.5.2
```

```
library("fBasics")
```

```
## Warning: package 'fBasics' was built under R version 3.5.2
```

```
## Loading required package: timeDate
```

```
## Warning: package 'timeDate' was built under R version 3.5.2
```

```
## Loading required package: timeSeries
```

```
## Warning: package 'timeSeries' was built under R version 3.5.2
```

```
##
## Attaching package: 'timeSeries'
## The following object is masked from 'package:psych':
##
##     outlier
```

```
##
## Attaching package: 'fBasics'
```

```
## The following object is masked from 'package:psych':
##
##     tr
```

```
library("forecast")
```

```
## Warning: package 'forecast' was built under R version 3.5.2
```

Data Import

No need for a split and train data considering the small size of the data.

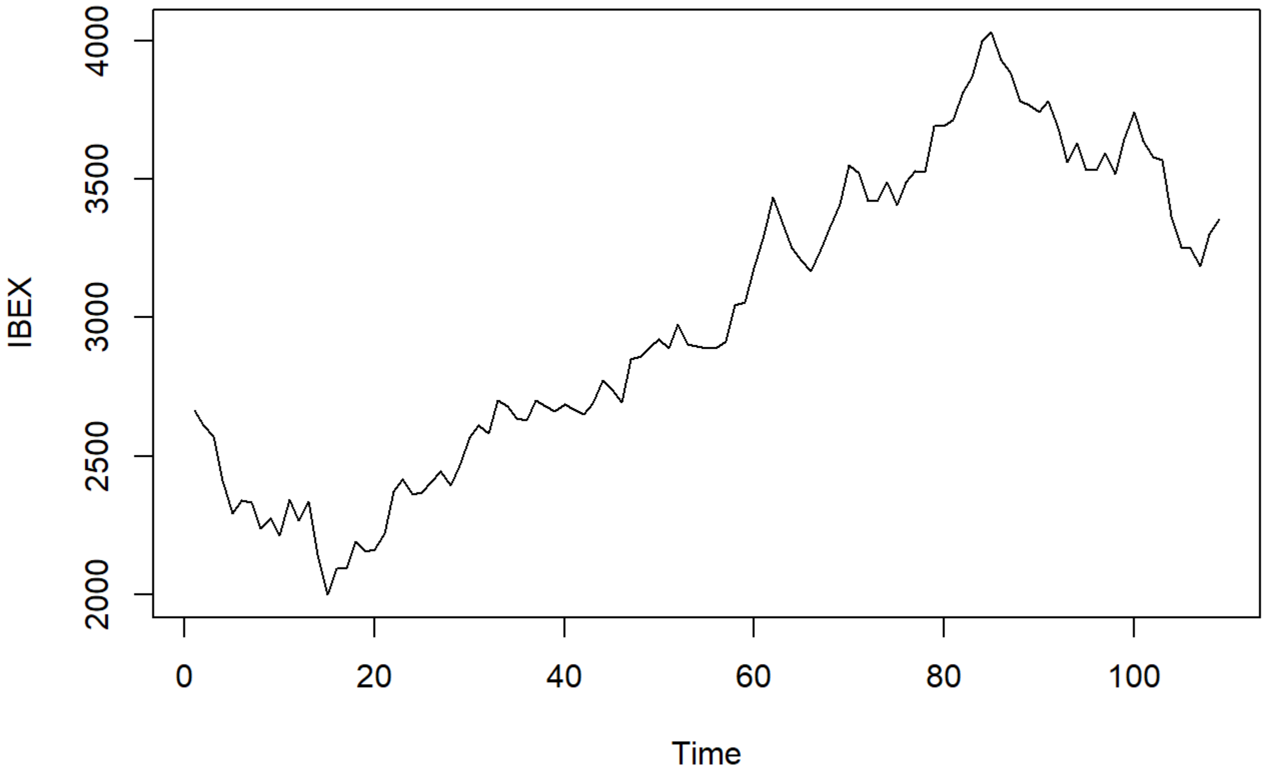
```
df_raw <- read.csv("https://gist.githubusercontent.com/f-loguercio/df301be228aff27132d0f3d6fa4ee
932/raw/6ccefaa8243e5af3e1e40717009c6ceba0b16d56/hw3_fts.csv", header = TRUE, sep = ";", dec=","
)

colnames(df_raw)<-c("Week","IBEX","EX","ST","LT")
```

Create Separate Variables

```
IBEX<-df_raw[,2]
EX<-df_raw[,3]
ST<-df_raw[,4]
LT<-df_raw[,5]
```

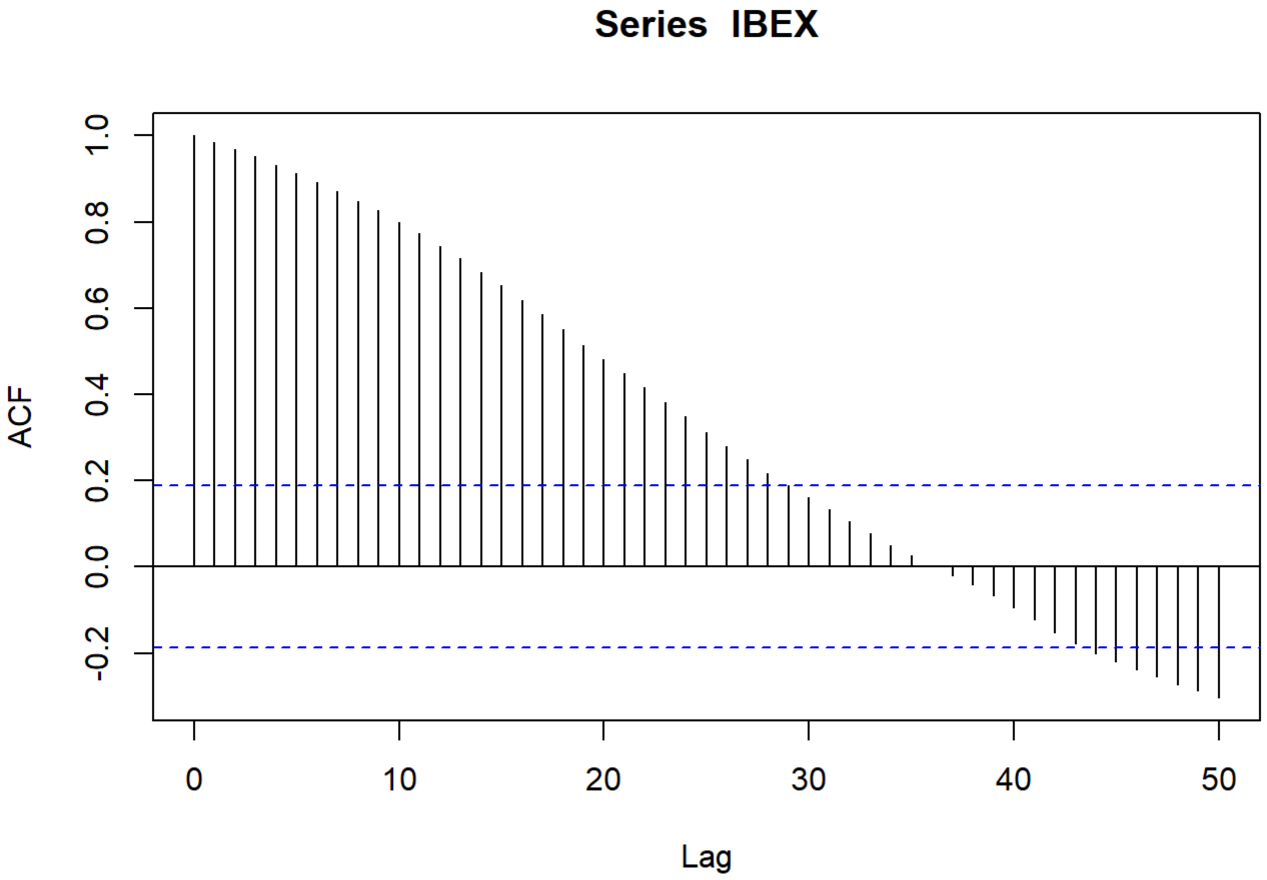
1. Find the best time series model for the variable “ibex”

```
nlags=50
ts.plot(IBEX)
```

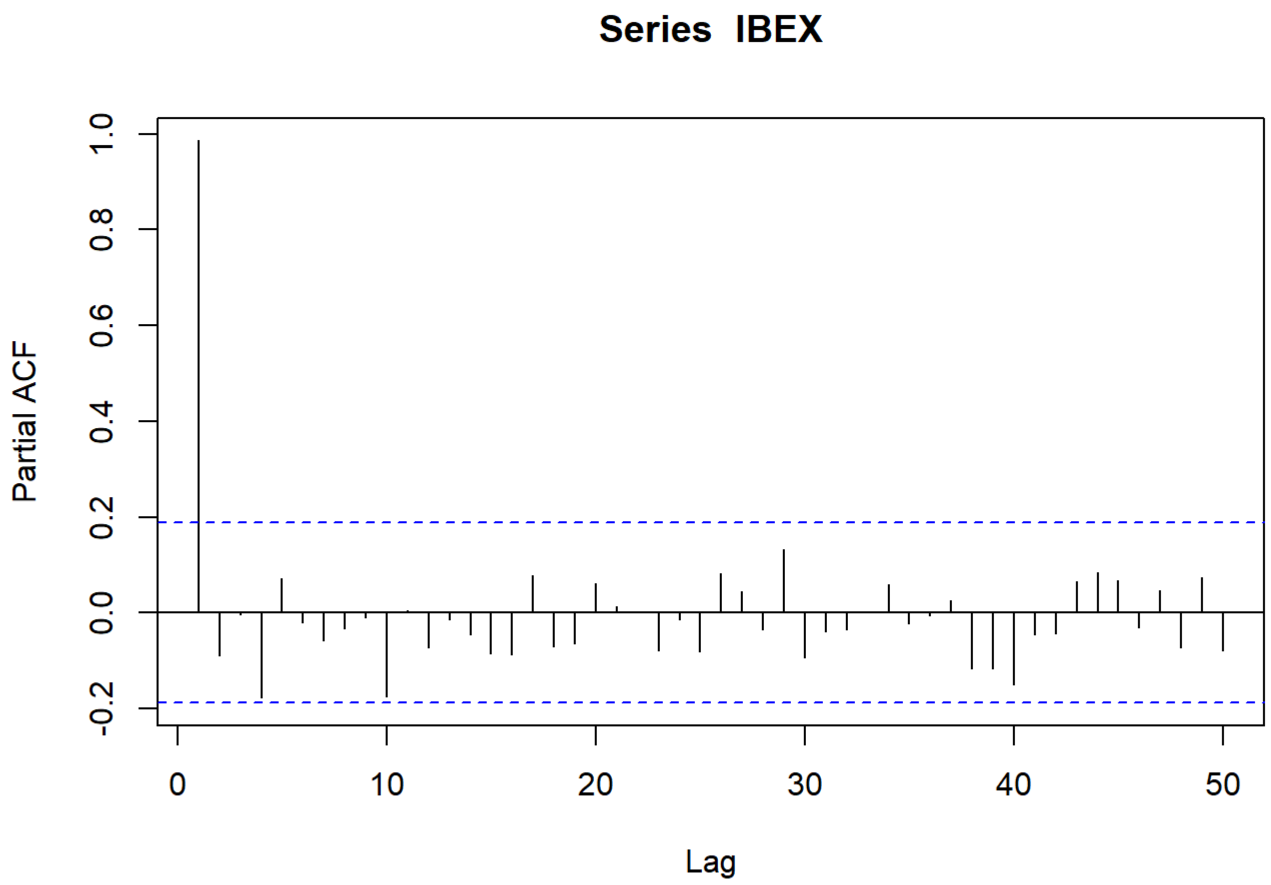


From the original dataset for the variable IBEX, we can see that the data is not stationary.

```
acf(IBEX,nlags)
```



```
pacf(IBEX,nlags)
```



In addition, there seems to be presence of cyclicality in the data, with the ACF having sinusodial shape.

```
s=52
nsdiffs(IBEX, m = s, test = c("ocsb"))
```

```
## Warning: argument m is deprecated; please set the frequency in the ts
## object.
```

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

```
ndiffs(IBEX, alpha=0.05, test=c("adf"))
```

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

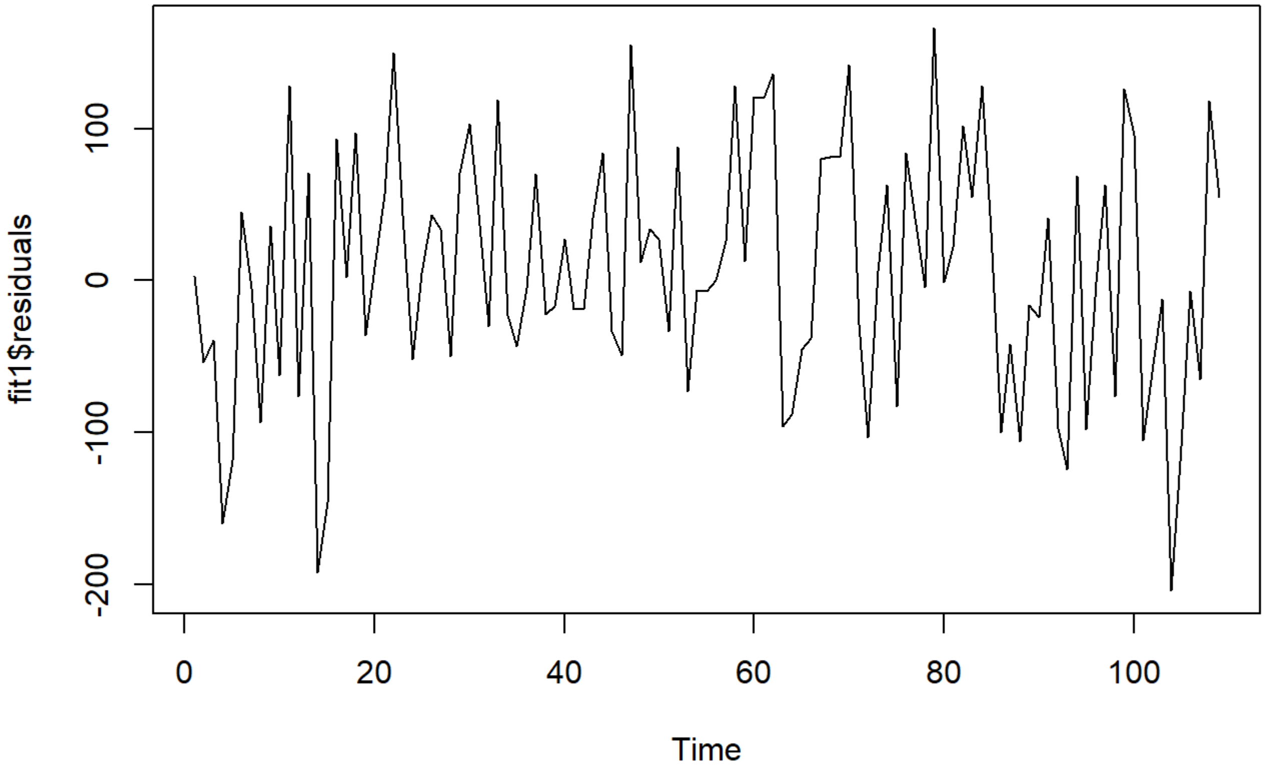
The formal test suggests that one regular difference needs to be taken. No seasonal difference is needed.

We will first take that suggested regular difference and then analyse the remaining structure in the data.

```
fit1 <- arima(IBEX,order=c(0,1,0),seasonal=list(order=c(0,0,0),period=s))
fit1
```

```
##
## Call:
## arima(x = IBEX, order = c(0, 1, 0), seasonal = list(order = c(0, 0, 0), period = s))
##
##
## sigma^2 estimated as 6467:  log likelihood = -627.07,  aic = 1256.13
```

```
plot(fit1$residuals)
```



Data seems to be stationary in the variance and in the mean. This is confirmed with the formal tests indicating no need for additional differences.

```
nsdiffs(fit1$residuals, m = s, test = c("ocsb"))
```

```
## Warning: argument m is deprecated; please set the frequency in the ts
## object.
```

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

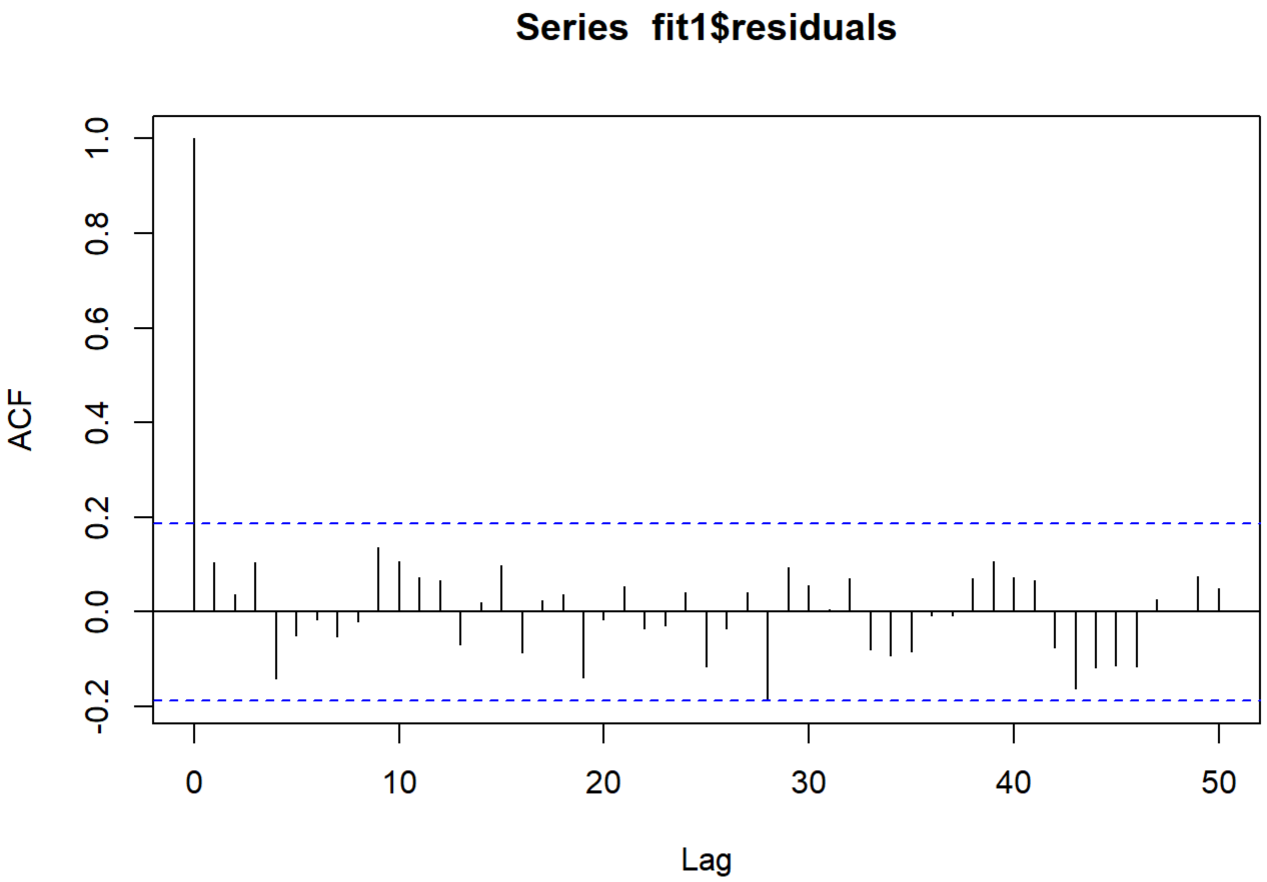
```
ndiffs(fit1$residuals, alpha=0.05, test=c("adf"))
```

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

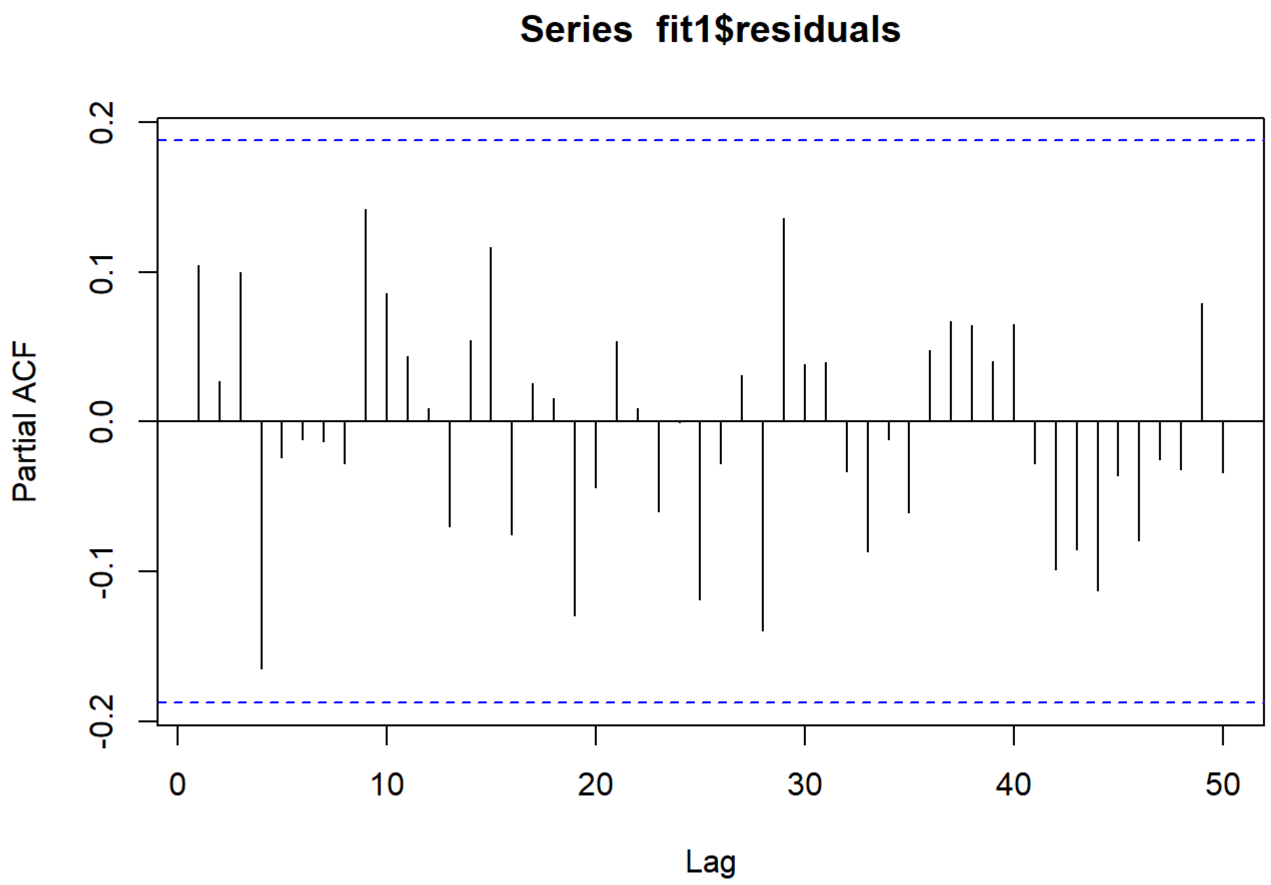
We will first proceed to estimate a model with this data and then compare the estimation performance with the rest of the following models.

Let’s analyse the residuals in terms of autocorrelation within ACF and PACF:

```
acf(fit1$residuals,nlags)
```



```
pacf(fit1$residuals,nlags)
```



There is no structure in the mean of the data, no remaining autocorrelation can be identified in the ACF and the PACF. The Box-Pierce test formally confirms that the data is White Noise.

```
Box.test(fit1$residuals, lag = 20)
```

```
##
##  Box-Pierce test
##
## data:  fit1$residuals
## X-squared = 14.384, df = 20, p-value = 0.8105
```

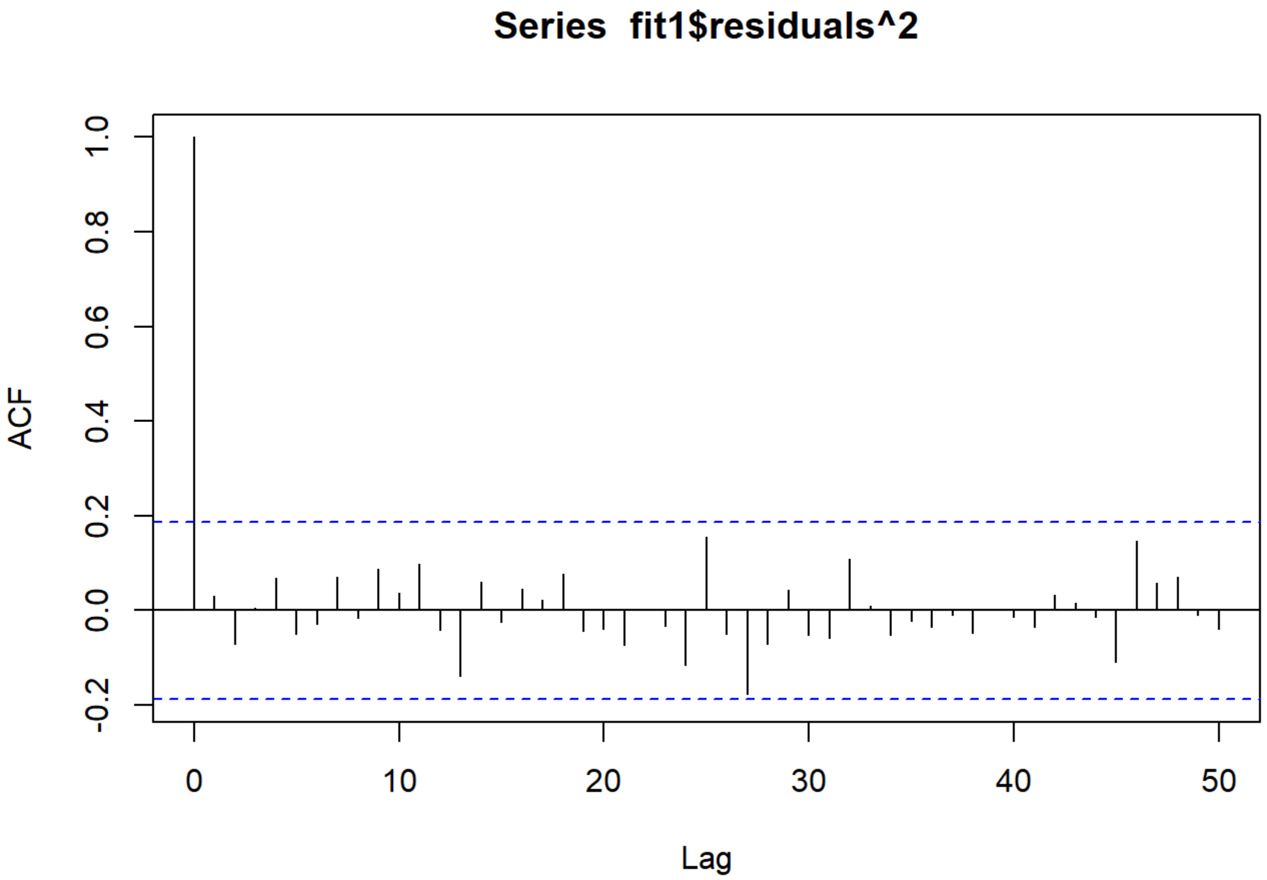
Checking for normality.

```
shapiro.test(fit1$residuals)
```

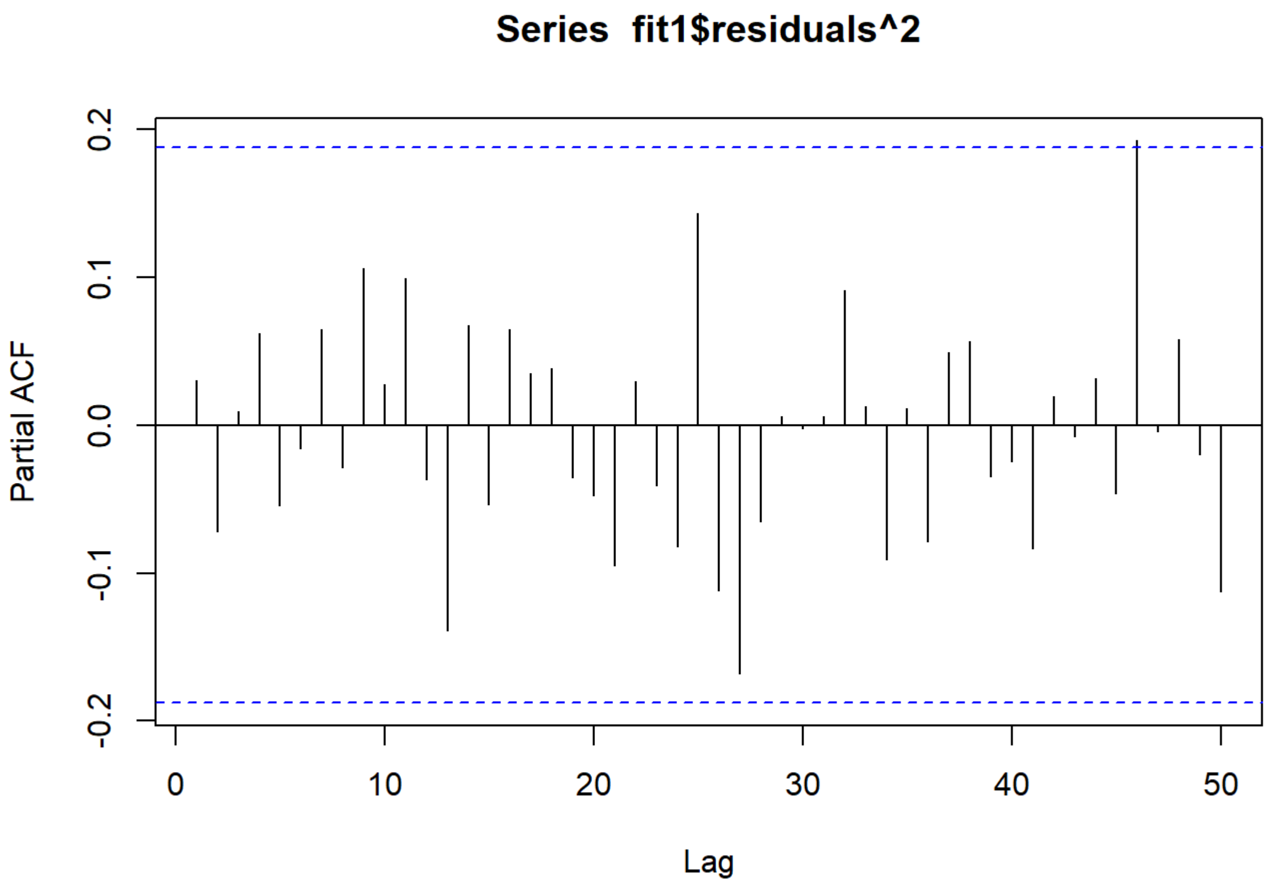
```
##
##  Shapiro-Wilk normality test
##
## data:  fit1$residuals
## W = 0.98799, p-value = 0.4433
```

Formal Shapiro Test confirms the data is normally distributed, meaning the residuals are Gaussian White Noise (GWN). Thus, we can infer the presence of Strict White Noise (SWN) as well, confirmed by the formal test (Box-Test of the squared residuals)

```
acf(fit1$residuals^2,nlags)
```



```
pacf(fit1$residuals^2,nlags)
```



```
Box.test(fit1$residuals^2, lag = 20)
```

```
##
##  Box-Pierce test
##
## data:  fit1$residuals^2
## X-squared = 8.1168, df = 20, p-value = 0.9911
```

2. Find the best regression model for the dependent variable “ibex”.

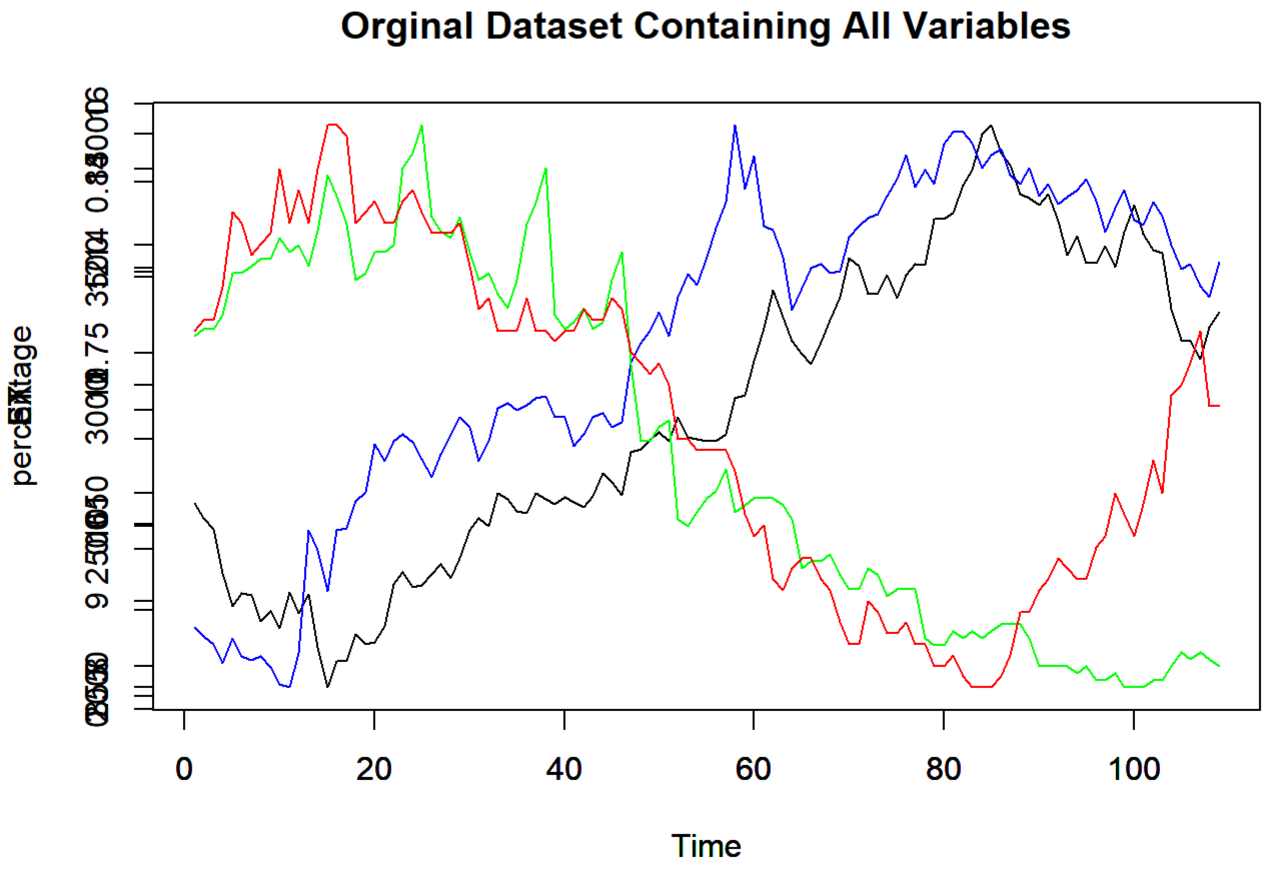
The following questions will be answered:

**A. Do we have multicollinearity with these explanatory variables?

**B. Are the residuals White Noise?

Plotting all the variables

```
par(mfrow=c(1,1))
ts.plot(IBEX,col="black",ylab="percentage",
        main = "Orginal Dataset Containing All Variables")
par(new=TRUE)
ts.plot(EX,col="blue")
par(new=TRUE)
ts.plot(ST,col="green")
par(new=TRUE)
ts.plot(LT,col="red")
```



Verifying correlation between all the variables. There seems to be some correlation between variables. However, it is important to mention that the highest correlations are between the target variable (IBEX) and the rest of the variables. The rest of the variables are less correlated to each other, confirming the absence of multicollinearity between the explanatory variables.

```
corr_all<-corr.test(df_raw[2:5])
corr_all
```

```
## Call:corr.test(x = df_raw[2:5])
## Correlation matrix
##       IBEX    EX    ST    LT
## IBEX  1.00  0.89 -0.93 -0.94
## EX    0.89  1.00 -0.84 -0.87
## ST   -0.93 -0.84  1.00  0.87
## LT   -0.94 -0.87  0.87  1.00
## Sample Size
## [1] 109
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##      IBEX EX ST LT
## IBEX    0  0  0  0
## EX      0  0  0  0
## ST      0  0  0  0
## LT      0  0  0  0
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

Fitting a regression model with all the explanatory variables.

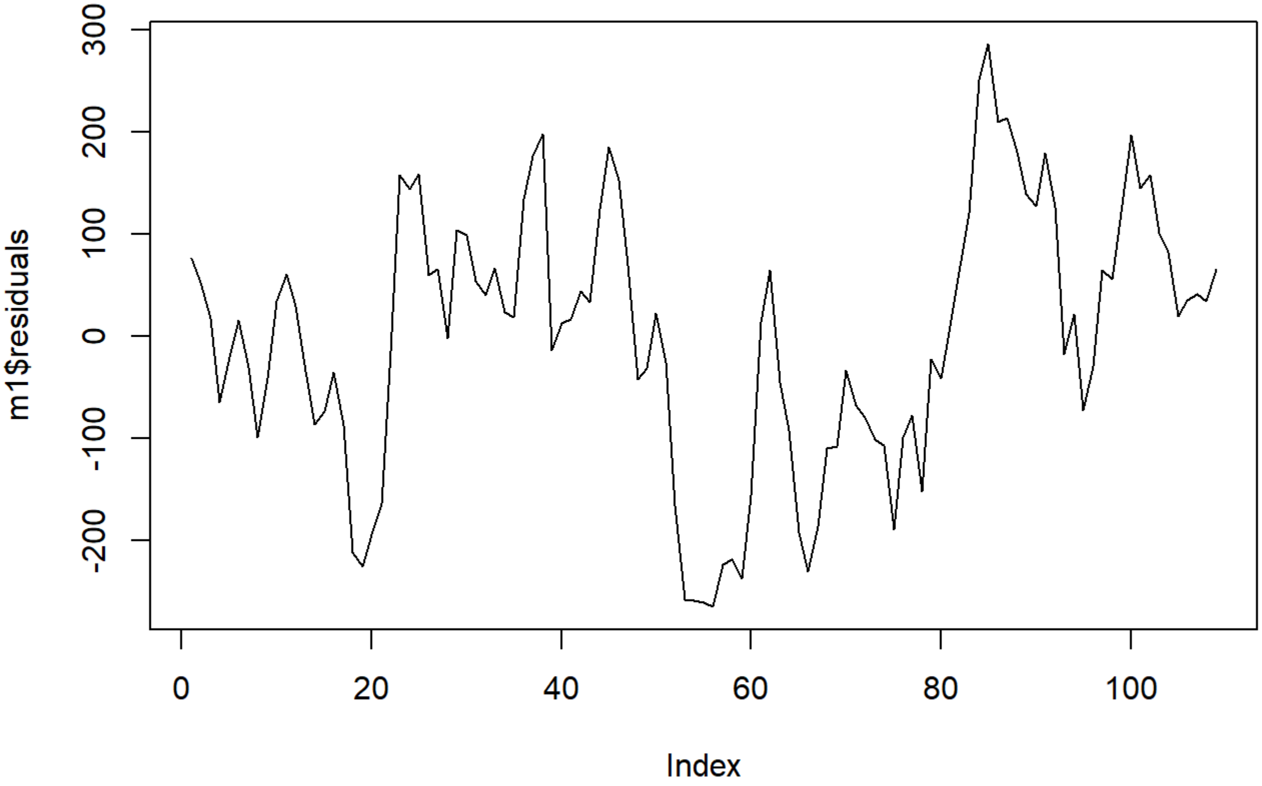
```
m1 = lm(IBEX ~ EX + ST + LT)

summary(m1)
```

```
##
## Call:
## lm(formula = IBEX ~ EX + ST + LT)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -264.47  -78.77   16.29   76.58  285.68
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5231.68     376.91  13.881  < 2e-16 ***
## EX            783.34     288.44   2.716  0.00773 **
## ST            -88.70      10.51  -8.444 1.84e-13 ***
## LT           -172.16      18.92  -9.098 6.45e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 129.3 on 105 degrees of freedom
## Multiple R-squared:  0.9471, Adjusted R-squared:  0.9455
## F-statistic: 626.1 on 3 and 105 DF,  p-value: < 2.2e-16
```

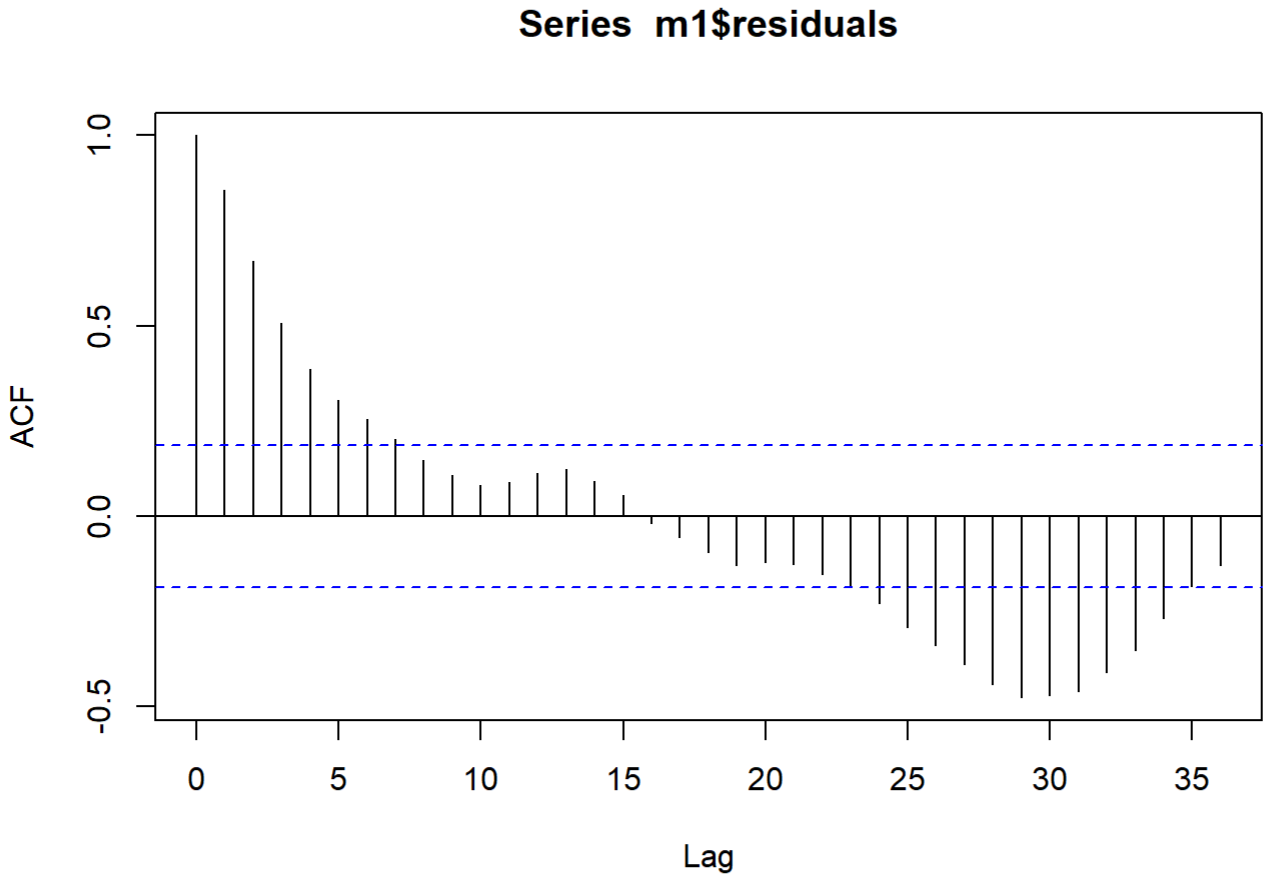
All variables are significant in estimating the model (all below p-value<0.05).

We then check if the residuals of the joint estimation are stationary.

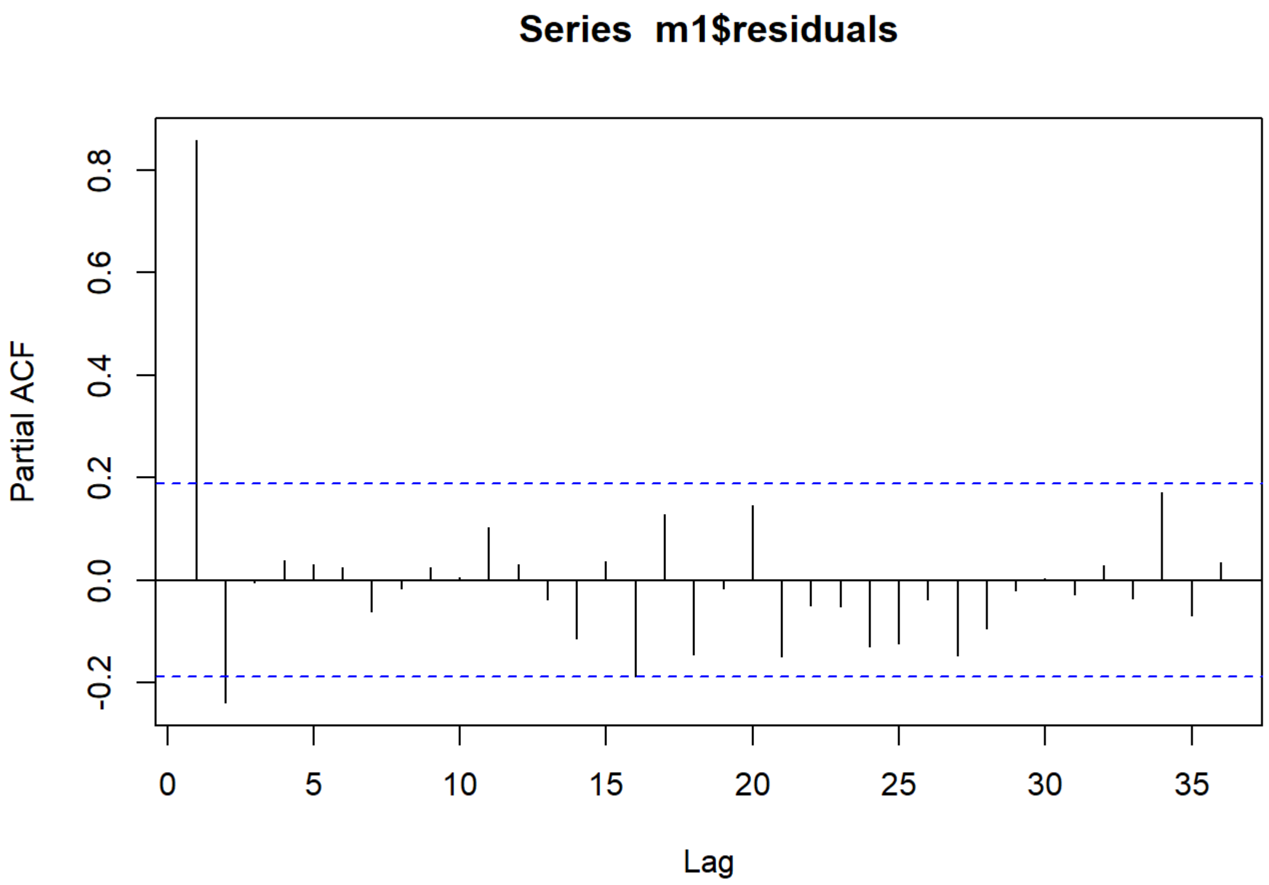
```
plot(m1$residuals,type='l')
```



```
acf(m1$residuals,lag=36)
```



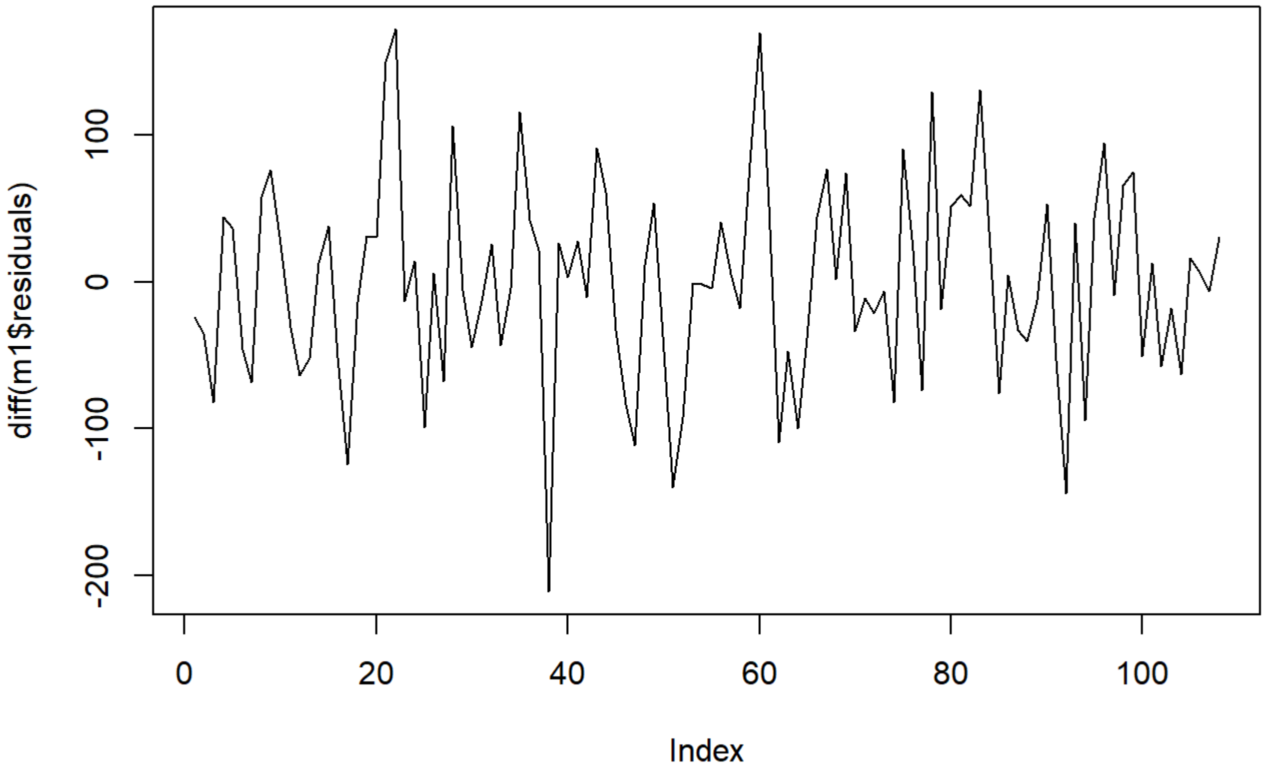
```
pacf(m1$residuals,lag=36)
```



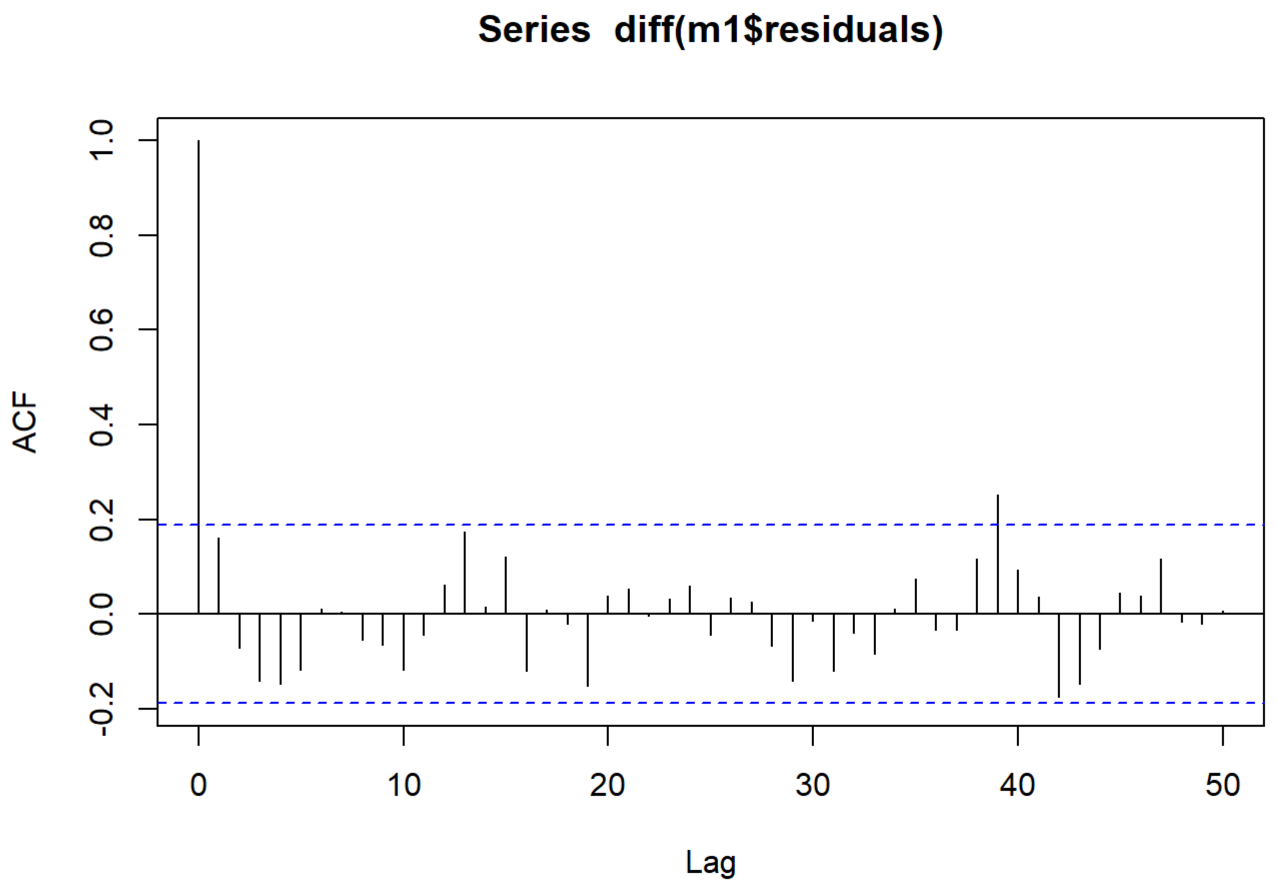
Joint estimation of the residuals is not stationary for model 1 (m1), therefore we need to take one difference for every explanatory variable.

We check the residuals for the difference of the variables to see if they are stationary.

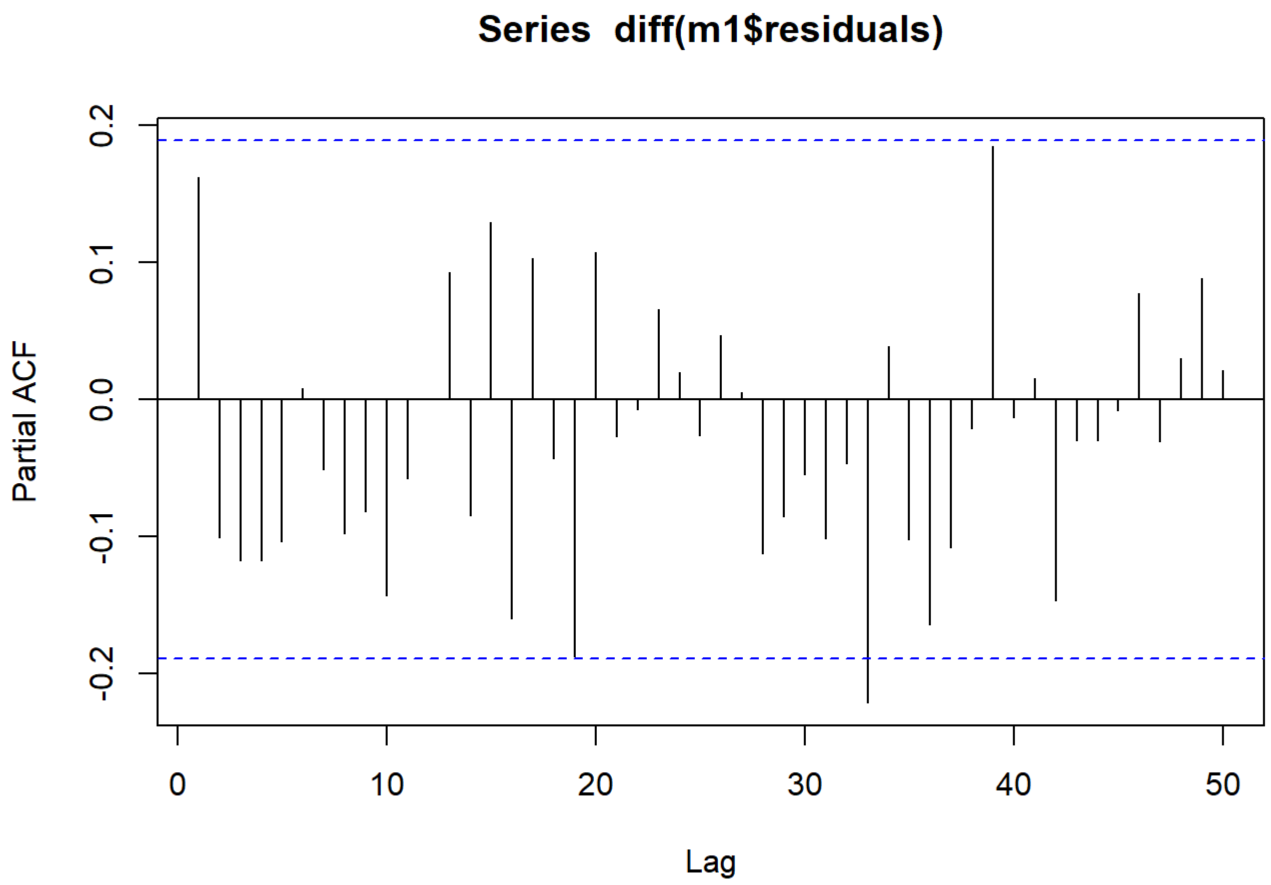
```
plot(diff(m1$residuals),type='l')
```



```
acf(diff(m1$residuals),lag=50)
```



```
pacf(diff(m1$residuals),lag=50)
```



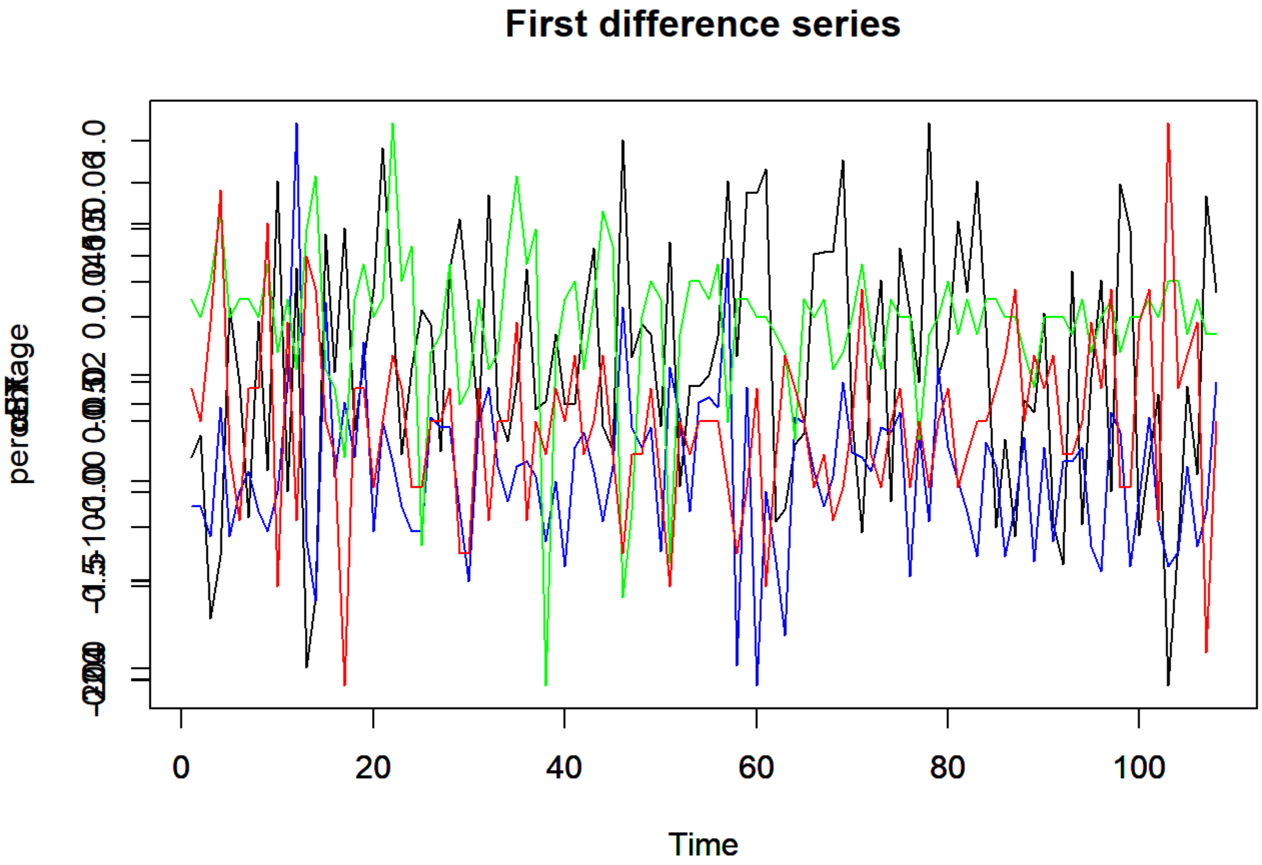
Following the difference of the model, we can see that the joint estimation has become stationary. Thus, from this, we confirm the need to take a difference for all the variables so the joint becomes stationary.

Take the differences of each variable.

```
cIBEX<-diff(IBEX)
cEX<-diff(EX)
cST<-diff(ST)
cLT<-diff(LT)
```

Now, plotting the differences for each variable - we can see that each of them are stationary.

```
ts.plot(cIBEX,col="black",ylab="percentage",
        main = "First difference series")
par(new=TRUE)
ts.plot(cEX,col="blue")
par(new=TRUE)
ts.plot(cST,col="green")
par(new=TRUE)
ts.plot(cLT,col="red")
```



Then, here is the new regression model with the differences.

```
m2 = lm(cIBEX ~ cEX + cST + cLT)
summary(m2)
```

```
##
## Call:
## lm(formula = cIBEX ~ cEX + cST + cLT)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -150.683  -34.686    2.572   36.021  167.939
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.5640     5.6745   0.628   0.5313
## cEX          767.1516   360.1982   2.130   0.0355 *
## cST           -0.9214    14.4686  -0.064   0.9493
## cLT         -200.5001    23.8518  -8.406 2.37e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 58.32 on 104 degrees of freedom
## Multiple R-squared:  0.4903, Adjusted R-squared:  0.4755
## F-statistic: 33.34 on 3 and 104 DF,  p-value: 3.511e-15
```

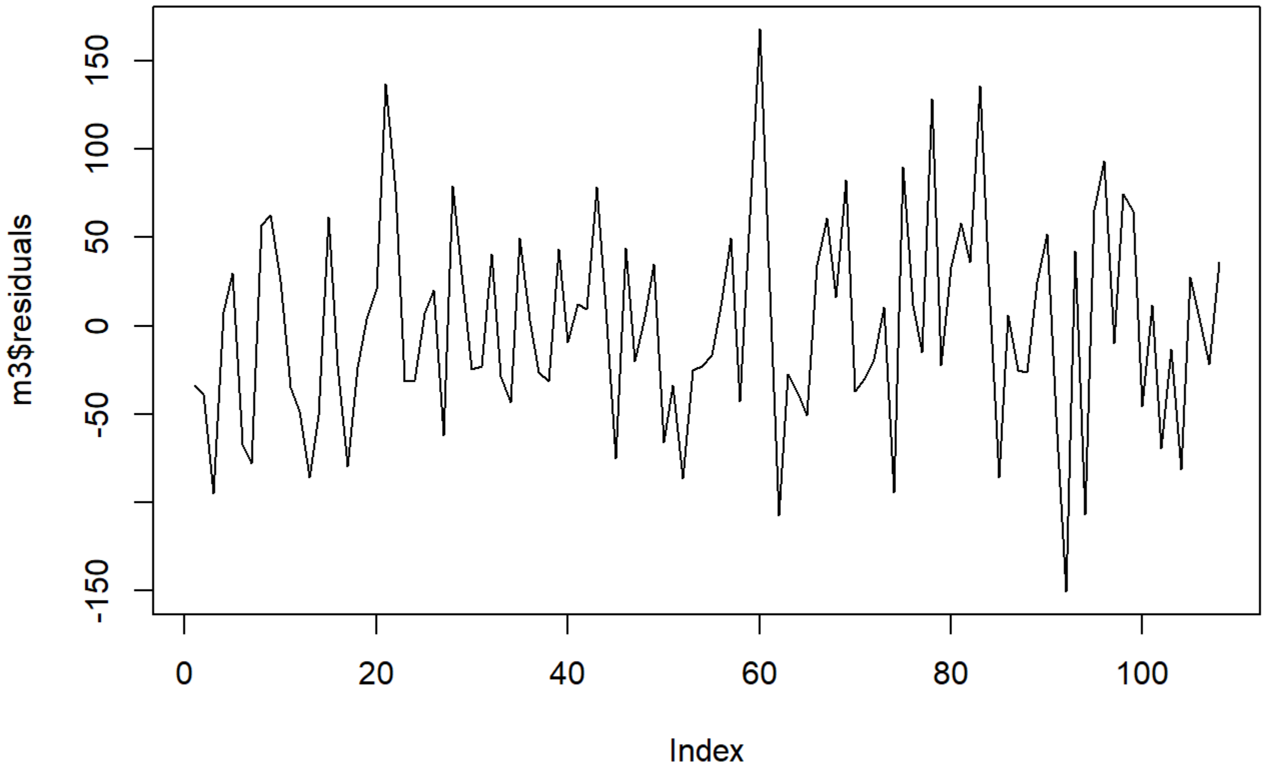
From the summary, we can see that some variables are insignificant when modelling the linear regression. Therefore, we take out insignificant variables with pvalue < 0.05. In this case, only the short-term rate (cST) is insignificant and taken out.

```
m3 = lm(cIBEX ~ cEX + cLT)
summary(m3)
```

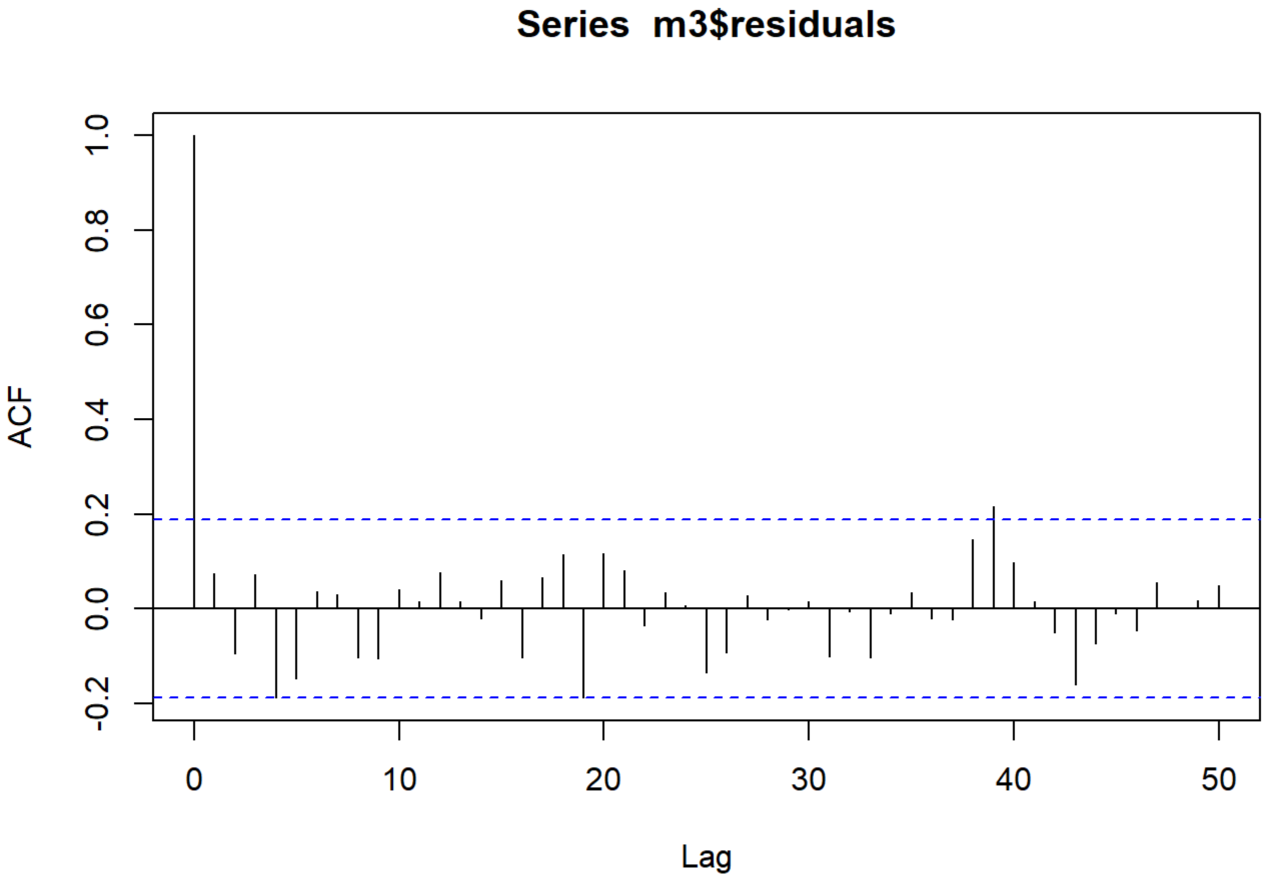
```
##
## Call:
## lm(formula = cIBEX ~ cEX + cLT)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -150.790  -34.078    2.372   35.934  168.101
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.594      5.628   0.639   0.5245
## cEX          770.234    355.234   2.168   0.0324 *
## cLT         -201.156     21.414  -9.394 1.41e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 58.04 on 105 degrees of freedom
## Multiple R-squared:  0.4902, Adjusted R-squared:  0.4805
## F-statistic: 50.49 on 2 and 105 DF,  p-value: 4.335e-16
```

We now check if the residuals of the joint estimation (new model = m3) are stationary.

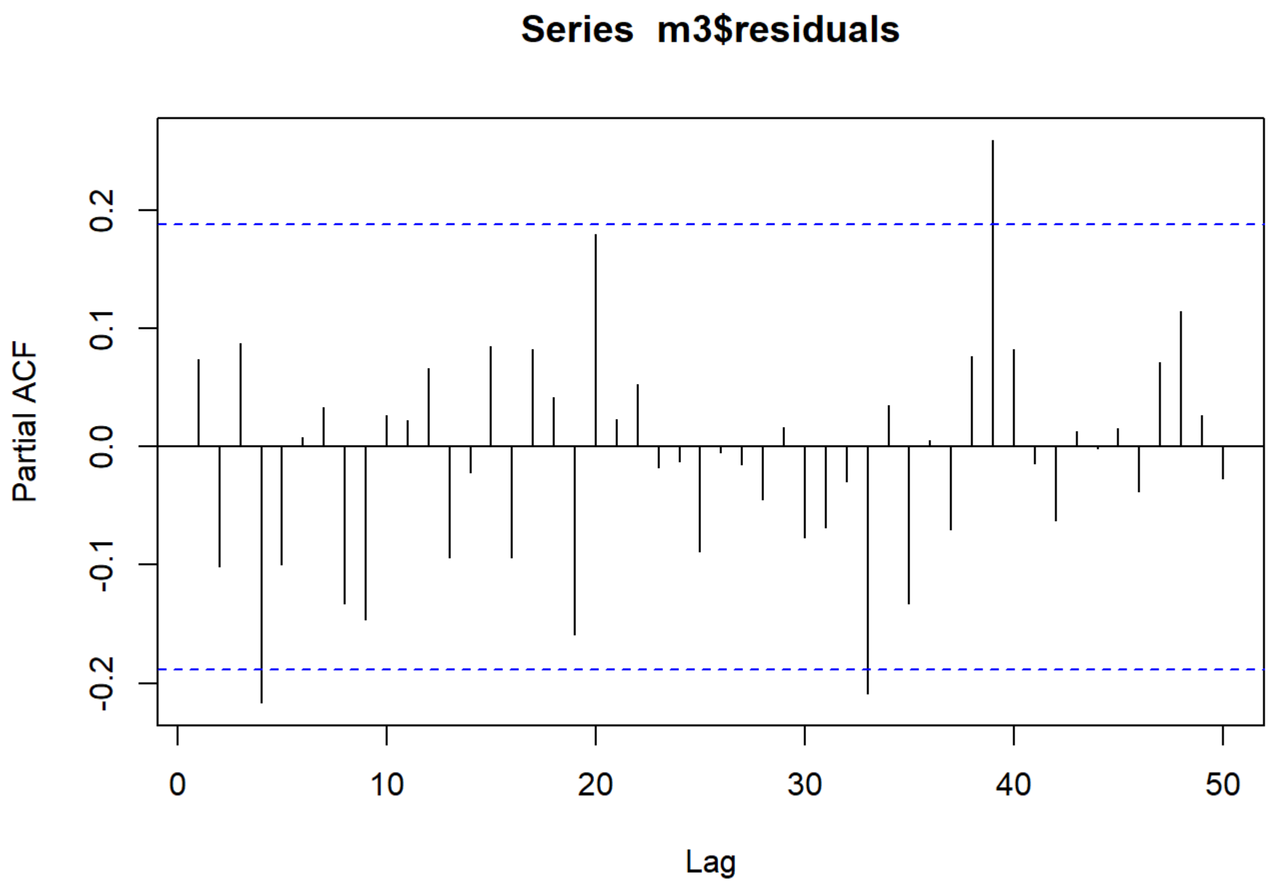
```
plot(m3$residuals,type='l')
```



```
acf(m3$residuals,lag=50)
```



```
pacf(m3$residuals,lag=50)
```



```
Box.test(m3$residuals,lag=50)
```

```
##
##  Box-Pierce test
##
## data:  m3$residuals
## X-squared = 39.932, df = 50, p-value = 0.8451
```

We can confirm that the model is stationary. In addition, when running the Box-Test, it confirms the presence of White Noise in the residuals. However, we can see that in the PACF, there is a residual out of limit, at lag 4.

In this case, we will compare both models (in question 3), one without fitting an ARIMA model (m3) and one fitting an ARIMA model (m4) for lag 4, and see which model has the best predictions and the least variance (errors).

3. Find the best regression model with time series errors for the dependent variable “ibex”

The following questions will be answered:

**A. Does this model maintain the same number of lags as the model found in question 2, and the same number of regressors as those found in question 1?

**B. Derive the final equation for the selected model

In this case, we will use the AR model at lag 4 because there is a lag at time 4 in the PACF (which was not done in question 2 for model 3).

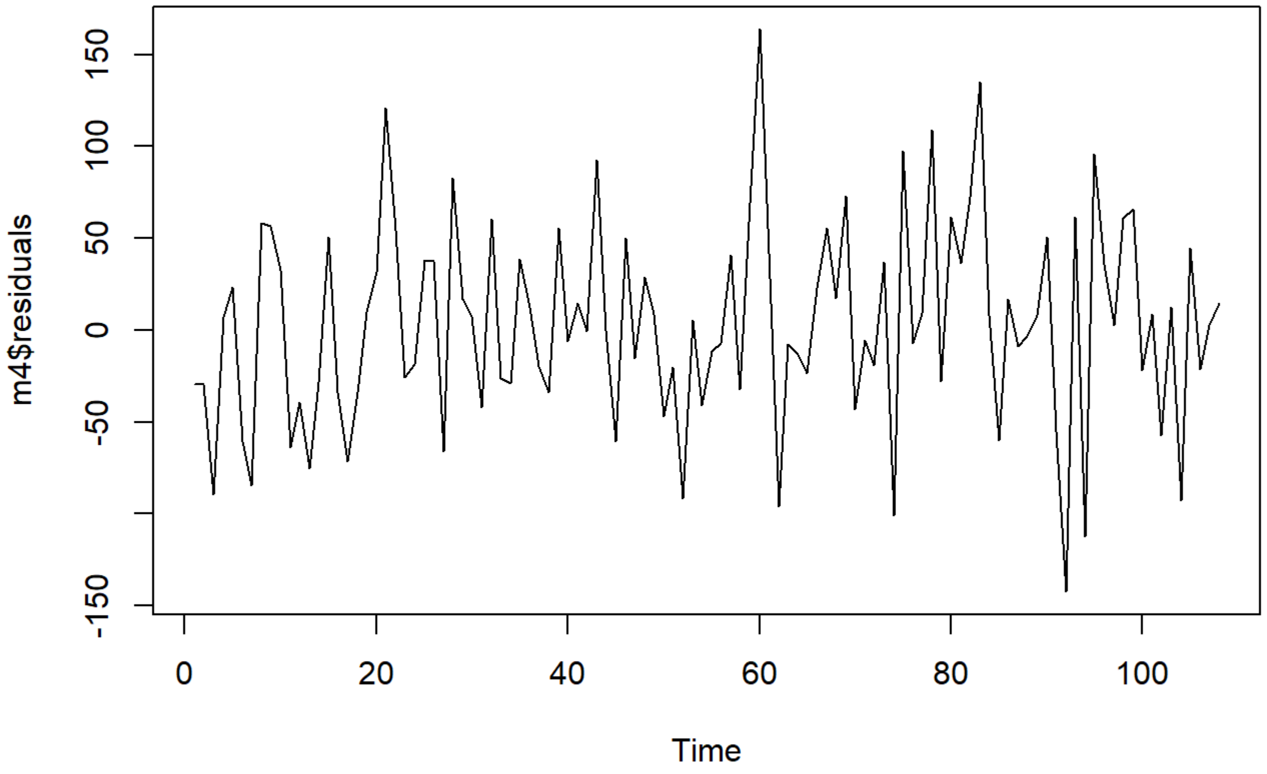
```
d_features = data.frame(cEX,cLT)

m4=arima(cIBEX,order=c(4,0,0),xreg=d_features,include.mean=F)
summary(m4)
```

```
##
## Call:
## arima(x = cIBEX, order = c(4, 0, 0), xreg = d_features, include.mean = F)
##
## Coefficients:
##          ar1      ar2     ar3      ar4        cEX        cLT
##       0.1443  -0.1283  0.1276  -0.2341  1000.6455  -185.0093
## s.e.  0.0977   0.0941  0.0958   0.0957   323.1217    20.6746
##
## sigma^2 estimated as 3033:  log likelihood = -586.33,  aic = 1186.66
##
## Training set error measures:
##                    ME     RMSE      MAE  MPE MAPE      MASE        ACF1
## Training set 3.474125 55.07672 43.35776 -Inf  Inf 0.4944346 -0.02134058
```

From the model, we can see that AR(4) is significant in our prediction model.

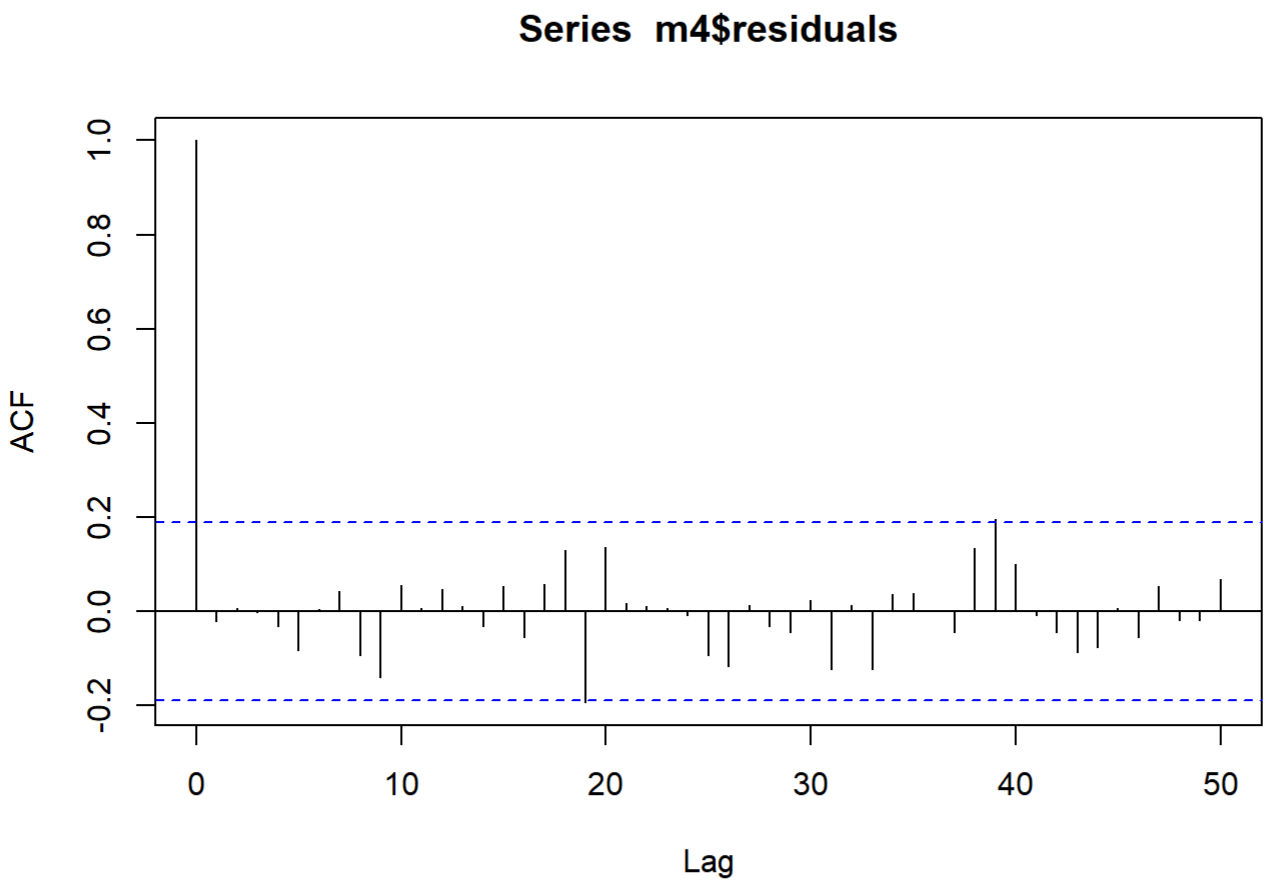
We then see if our joint estimation model is still stationary in the residuals.

```
plot(m4$residuals,type='l')
```

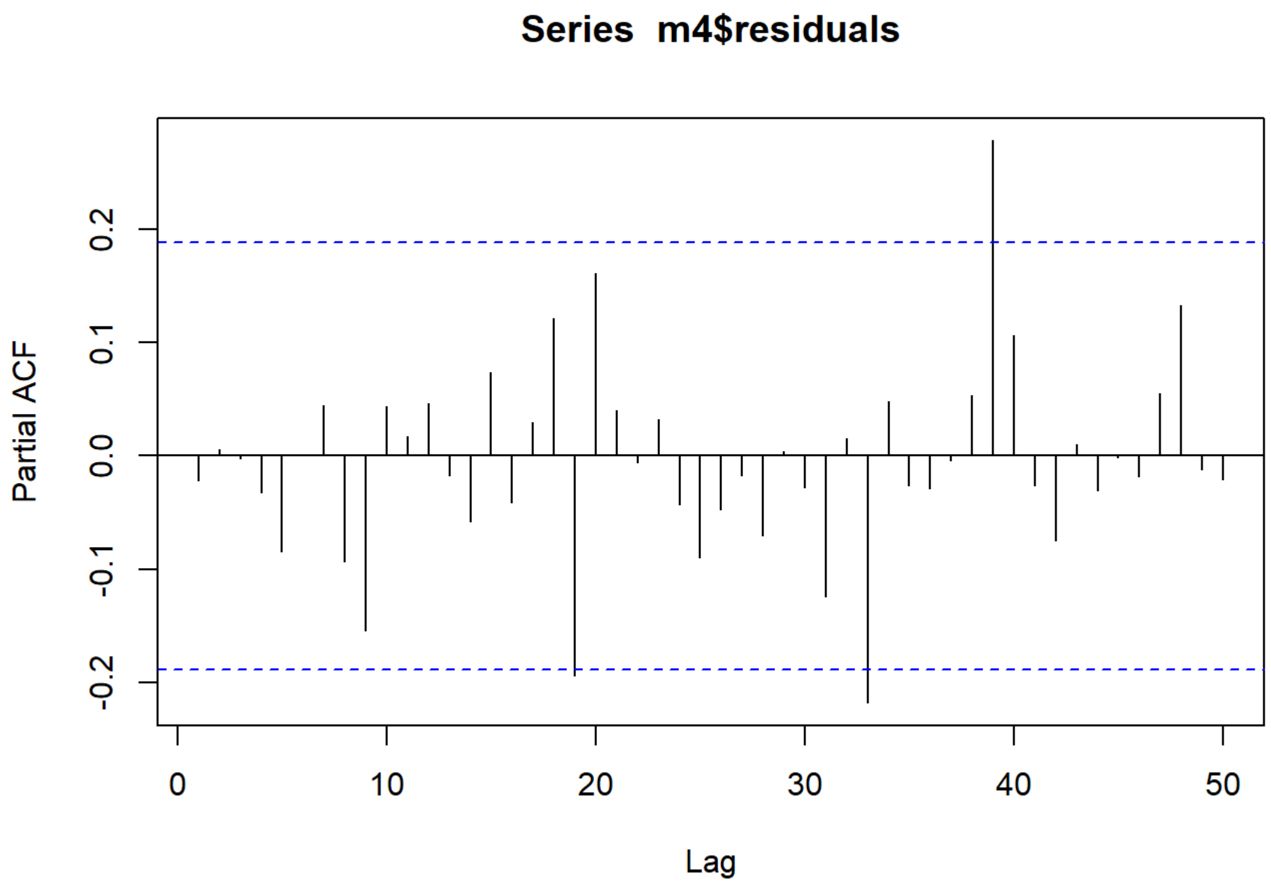


Residuals are stationary. Therefore, we want to see if the residuals are WN.

```
acf(m4$residuals,lag=50)
```



```
pacf(m4$residuals,lag=50)
```



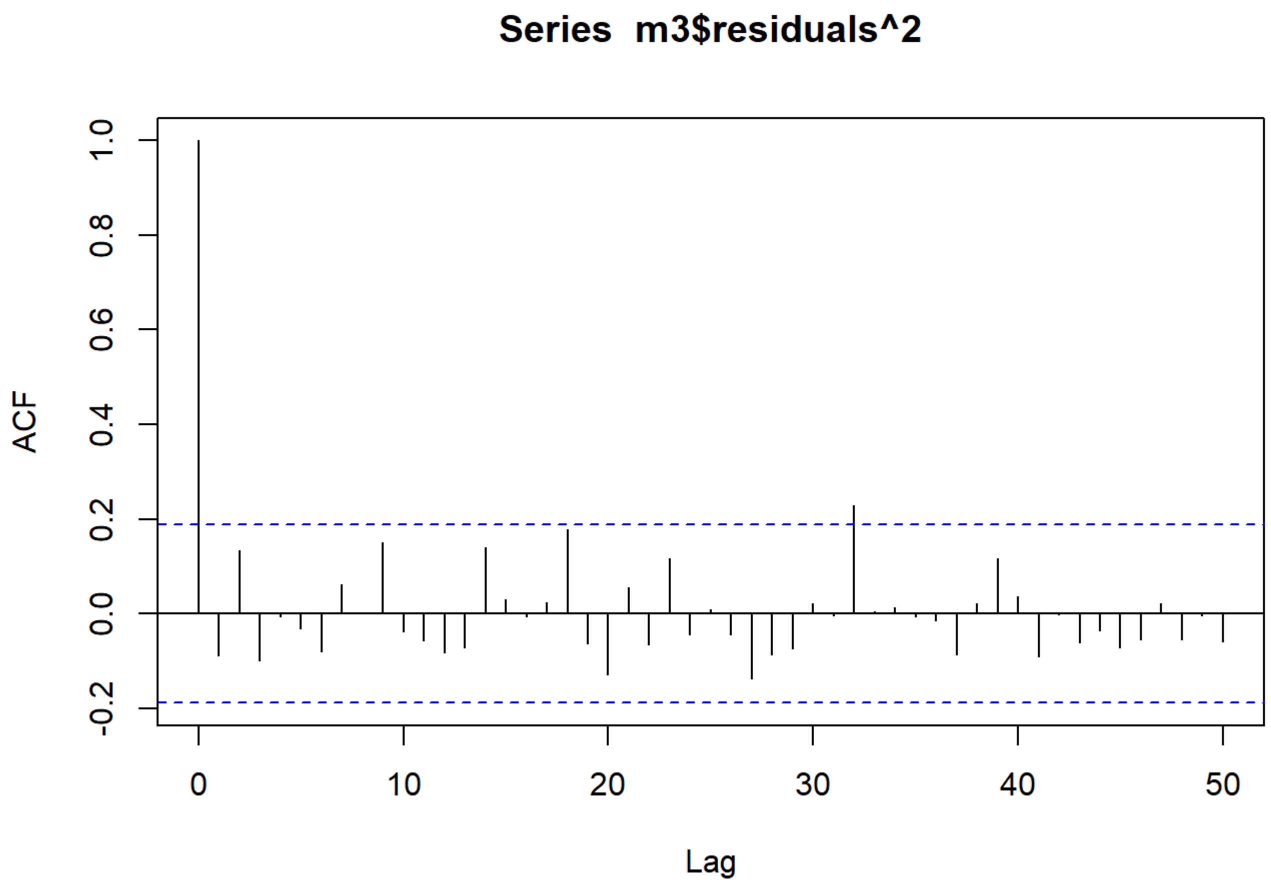
```
Box.test(m4$residuals,lag=50)
```

```
##
##  Box-Pierce test
##
## data:  m4$residuals
## X-squared = 30.397, df = 50, p-value = 0.9871
```

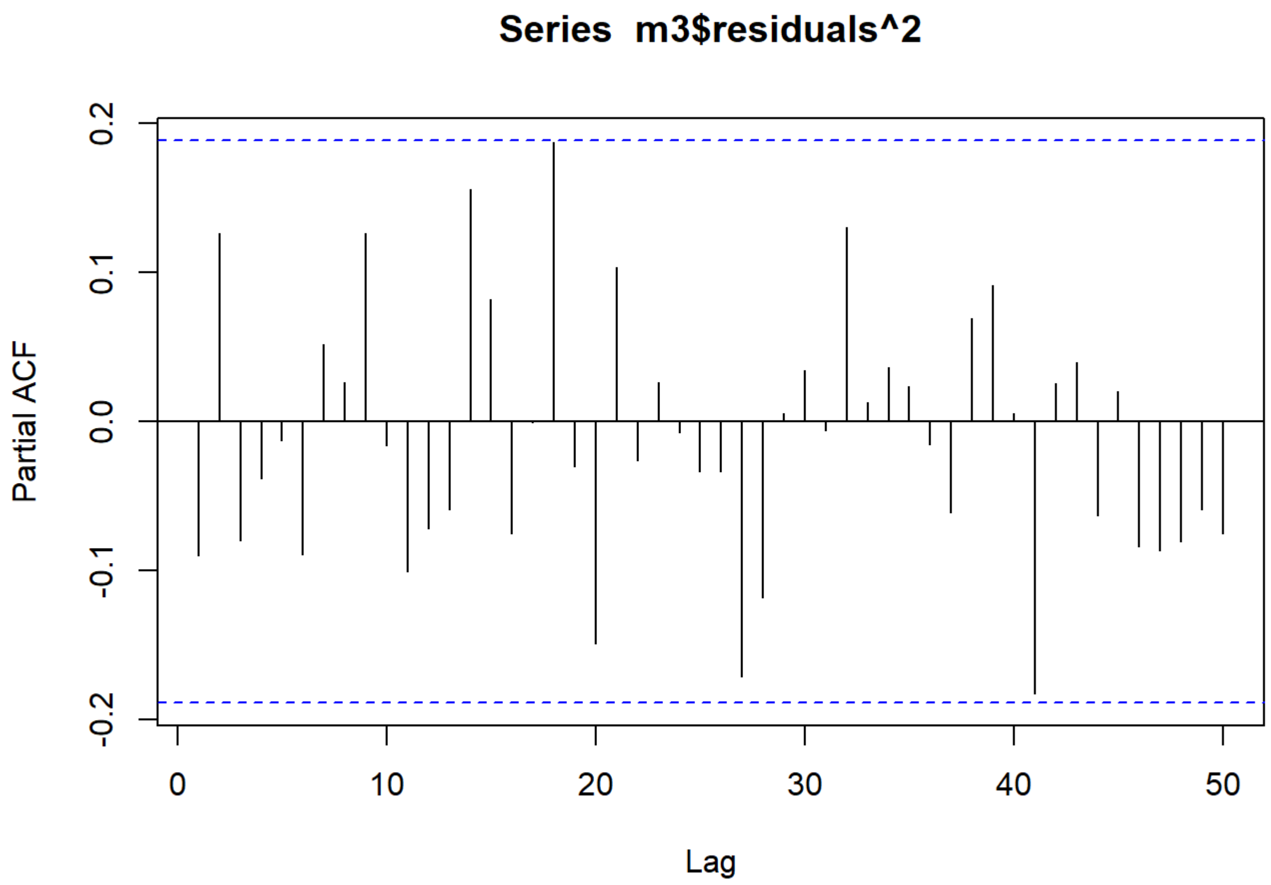
No lags seem to be out of limits in the ACF and PACF. In addtion, when running the Box-Test, it confirms the presence of White Noise.

Are the squared residuals SWN?

```
acf(m3$residuals^2,lag=50)
```



```
pacf(m3$residuals^2,lag=50)
```



```
Box.test(m3$residuals^2,lag=50)
```

```
##
##  Box-Pierce test
##
## data:  m3$residuals^2
## X-squared = 34.544, df = 50, p-value = 0.9528
```

Checking for normality.

```
shapiro.test(m4$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  m4$residuals
## W = 0.99569, p-value = 0.9846
```

There are no lags out of limit in the ACF and PACF of the squared residuals. To confirm this, we check for normality and there is presence of GWN, hence presence of SWN.

Thus, there is SWN and we cannot fit GARCH(1,1) model.

Computing a model with the original data

```
df2<-df_raw
df2$ST<-NULL
df2$IBEX<-NULL
df2$Week<-NULL

m5=arima(IBEX,order=c(4,1,0),xreg=df2,include.mean=F)
summary(m5)
```

```
##
## Call:
## arima(x = IBEX, order = c(4, 1, 0), xreg = df2, include.mean = F)
##
## Coefficients:
##          ar1      ar2     ar3      ar4         EX         LT
##       0.1444  -0.1283  0.1276  -0.2342  1000.4160  -185.0082
## s.e.  0.0977   0.0941  0.0958   0.0957   323.1235    20.6746
##
## sigma^2 estimated as 3033:  log likelihood = -586.33,  aic = 1186.66
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE    MAPE      MASE
## Training set 3.481331 54.82497 42.99878 0.06450493 1.46997 0.6565628
##                     ACF1
## Training set -0.02144263
```

Model 5 (m5) takes in consideration both the differences and the lags taken in question 1 and 2.

The final equation for the selected model (m5) is:

Y(t) = -0.2342Y(t-4) + 1000.4160EX -185.0082LT + error (where error is WN)

4. Choose among the three previous models the best one to explain variable “ibex” using the “estimate of the residual variance” as the in-sample criterion.

```
MSE_fit1<-mean(fit1$residuals^2)
MSE_fit1
```

```
## [1] 6407.634
```

```
MSE_m2<-mean(m2$residuals^2)
MSE_m2
```

```
## [1] 3275.489
```

```
MSE_m3<-mean(m3$residuals^2)
MSE_m3
```

```
## [1] 3275.617
```

```
MSE_m4<-mean(m4$residuals^2)
MSE_m4
```

```
## [1] 3033.445
```

```
MSE_m5<-mean(m5$residuals^2)
MSE_m5
```

```
## [1] 3005.777
```

Model 5 is the best model because it has the lowest mean squared error (mean residual variance) at 3005.77.

5. For the best model found in question 4, compute the one step ahead point prediction and confidence interval for the “ibex” given the values indicated in the case for all the explanatory variables.

```
y.pred<-predict(m5, n.ahead=1, newxreg=df2)

y.pred$pred   # point predictions
```

```
## Time Series:
## Start = 110
## End = 218
## Frequency = 1
##   [1] 3014.288 2990.785 2985.783 2919.276 2804.776 2812.272 2865.774
##   [8] 2849.274 2824.771 2703.762 2794.265 2758.771 2886.303 2781.794
##  [15] 2683.781 2719.796 2739.297 2903.310 2889.811 2899.322 2926.320
##  [22] 2938.325 2905.325 2881.822 2908.819 2935.817 2948.822 2959.827
##  [29] 2952.331 3020.331 3074.326 3067.830 3142.341 3145.342 3141.340
##  [36] 3088.839 3148.343 3149.344 3155.840 3137.339 3120.332 3090.333
##  [43] 3118.838 3120.839 3075.834 3097.336 3206.354 3235.859 3261.363
##  [50] 3253.867 3276.862 3392.376 3405.381 3417.880 3433.886 3450.894
##  [57] 3465.900 3547.920 3584.908 3640.918 3581.400 3671.903 3674.397
##  [64] 3606.383 3600.887 3612.892 3651.895 3665.393 3721.896 3778.906
##  [71] 3784.909 3715.908 3736.409 3784.415 3794.420 3789.925 3807.918
##  [78] 3817.922 3846.921 3869.930 3858.432 3895.434 3907.932 3892.926
##  [85] 3900.930 3885.430 3833.422 3754.417 3763.420 3710.412 3698.914
##  [92] 3649.908 3672.410 3694.913 3701.915 3633.408 3596.899 3536.902
##  [99] 3583.908 3603.902 3545.398 3484.401 3531.900 3348.385 3315.879
## [106] 3281.878 3213.370 3336.874 3356.882
```

```
mean(y.pred$pred)
```

```
## [1] 3333.57
```

```
y.pred$se     # standard errors
```

```
## Time Series:
## Start = 110
## End = 110
## Frequency = 1
## [1] 55.07672
```

In conclusion, when using model 5, we predict that next week’s (week 110) IBEX value will be at 3333.57 +/- 55.07672.