FORECASTING TIME SERIES HOMEWORK 3

TEAM G

Vratul Kapur
Paul Jacques-Mignault
Sheena Miles
Ashley O'Mahony
Stavros Tsentemeidis
Karl Westphal
Irune Maury Arrue

LOAD DATA

To begin with, we load the IBEX dataset, which consists of weekly data, in our working directory. At this stage split the data into **train** – **test**, in order to validate our models based on actual predictions, on unseen data.

```
y<-data[,2][1:95]        # TRAINING SET
y_test<-data[,2][96:107]  # TEST SET
```

CHECKING STATIONARITY

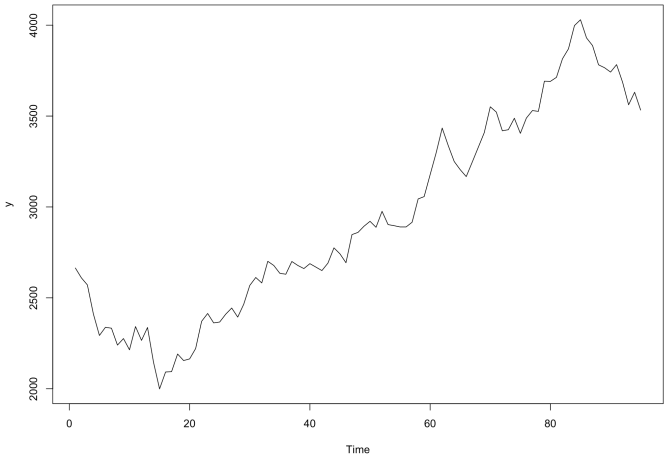
By plotting our timeseries, it is noticed that our data is not stationary both in the **mean** (not constant) as well as, in the **variance** (not constant either). In order to verify our observation, we will use the formal test. Also, as our data is about weeks, we think that there might be seasonality, so we also use the formal test for seasonal differences. In order to do so we set **s = 52** as again our data is weekly.

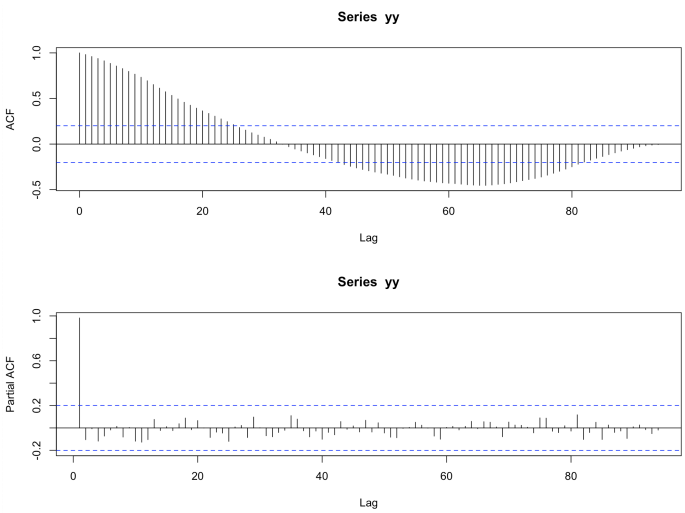
```
s=52
nsdiffs(y,m=s,test=c("ocsb"))      # seasonal differences
ndiffs(y, alpha=0.05, test=c("adf")) # regular differences
```

Find the best time series model for the variable "IBEX".

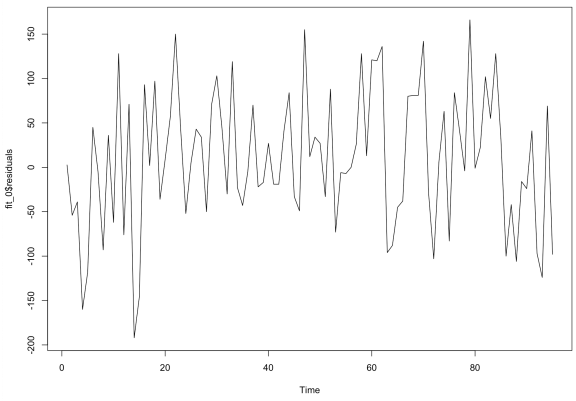
Based on the above tests we see that our model should take into account **1 regular and 0 seasonal difference**, in order to deal with the lack of stationarity in our data. Moving on with our analysis, we need to plot the **PACF** and the **ACF** graphs to detect spikes. In order to do so, we define the number of lags to be equally to 104, which corresponds to 2 years as our data is quarterly once more.

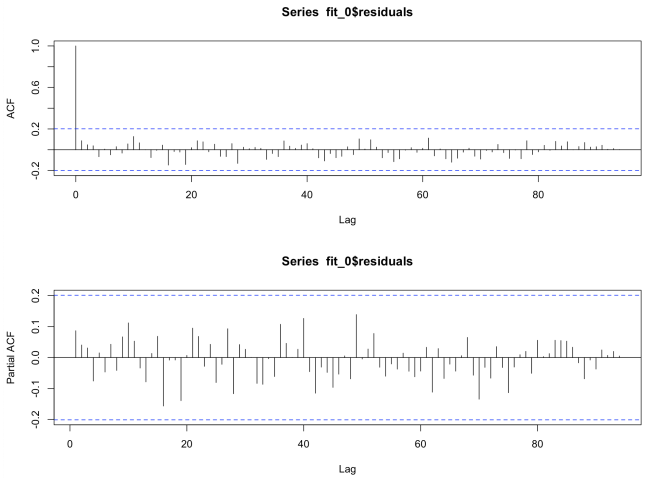
```
ts.plot(y)
nlags=40
par(mfrow=c(2,1))
acf(yy,nlags)
pacf(yy,nlags)
```





Based on the ACF and PACF we verify the fact that our data is **not stationary** as the ACF decays slowly. Furthermore, we notice that there is a **cyclic pattern** in our data. Looking at the PACF we can see that only lag 1 is out of limits. Let's now take the first difference in our data in order to observe the behavior after the transformation.

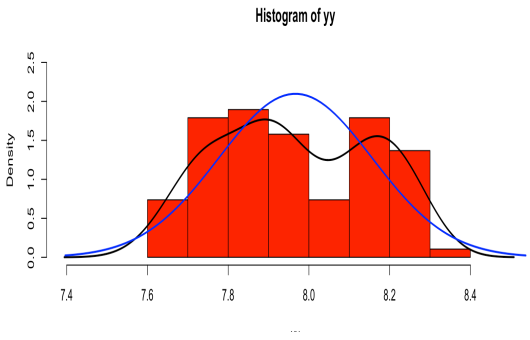





```
fit_0<-arima(yy,order=c(0,1,0))
fit_0

ts.plot(fit_0$residuals)

par(mfrow=c(2,1))
acf(fit_0$residuals,nlags)
pacf(fit_0$residuals,nlags)

ndiffs(fit_0$residuals, alpha=0.05, test=c("adf")) # regular differences?

Box.test(fit_0$residuals,lag=20)
shapiro.test(fit_0$residuals)

hist(yy,prob=T,ylim=c(0,2.5),xlim=c(mean(yy)-
3*sd(yy),mean(yy)+3*sd(yy)),col="red")
lines(density(yy),lwd=2)
mu<-mean(yy)
sigma<-sd(yy)
x<-seq(mu-3*sigma,mu+3*sigma,length=100)
yy2<-dnorm(x,mu,sigma)
lines(x,yy2,lwd=2,col="blue")
```

```
        Box-Pierce test

data:  fit_0$residuals
X-squared = 9.0493, df = 20, p-value = 0.9823
```

```
        Shapiro-Wilk normality test

data:  fit_0$residuals
W = 0.9899, p-value = 0.6899
```



After taking the first difference of our series, we can see that stationarity has been achieved both in the mean and the variance. Furthermore, we observe that our residuals are WN by both the plots, in which every lag is within bounce, but also from the Box-Ljung test. Lastly, our residuals are also normal, as the Shapiro Test gives us a p value bigger that 0.05. By combining **normality** with **WN** we conclude that our residuals are also **GWN**, which leads to the inference of **SWN**.

As the data is now clearly White Noise (WN), with **no linear** or **non -linear relationship**, it is decided that the best model to fit our data is a Random Walk, ARIMA with (0,1,0) parameters. We have deducted this as the data is WN and we only conducted one transformation of the original data. `fit_0<-arima(yy,order=c(0,1,0))`

Find the best regression model for the dependent variable "IBEX".

First we decide to use our whole dataset as a trainset, so we assign that to a variable to be more clear during the implementation.

```
model_0 <-lm(IBEX~.-Week, data=trainset)

lm_stats_0 <- summary(model_0)
lm_stats_0
```

```
Call:
lm(formula = IBEX ~ . - Week, data = trainset)

Residuals:
    Min      1Q  Median      3Q     Max
-263.83  -79.16   16.07   76.15  285.04

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      5240.62     377.33  13.889  < 2e-16 ***
Exchange_rate     779.20     288.52   2.701  0.00807 **
Short_term_rate   -88.01      10.54  -8.351 2.95e-13 ***
Long_term_rate...  -173.46      19.05  -9.106 6.19e-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:  0.9456
F-statistic: 626.6 on 3 and 105 DF,  p-value: < 2.2e-16
```

After that we created our baseline model with all the variables. The results that we were given is an **Adjusted R-Squared of 0.9456** which is pretty good. However, we have not tested yet our variables for *multicollinearity*. In order to do so, we are going to use the **VIF (Variance Inflation Factor)** method.
As a general rule, if VIF is *larger than 5*, then multi collinearity is assumed to be high. As a result, each time we are going to calculate the VIF values, remove the biggest one, re-do the model until all the explanatory variables have a VIF below 5. Handle the above procedure with a WHILE loop.

```
all_vifs <- car::vif(model_0)
signif_all <- names(all_vifs)
while(any(all_vifs > 5)){
 var_with_max_vif <- names(which(all_vifs == max(all_vifs))) # get the variable with max vif
 signif_all <- signif_all[!(signif_all) %in% var_with_max_vif] # remove this variable
 myForm <- as.formula(paste("IBEX~ ", paste (signif_all, collapse=" + "), sep="")) # design the new formula
 selectedMod <- lm(myForm, data=trainset) # re-build model with new formula
 all_vifs <- car::vif(selectedMod)
}
print(all_vifs)
```

As a resultsof the previous method, we conclude that the variable Long_term_rate has to be removed from our model. As a result, we have the following new model with its summary.

```
Exchange_rate Short_term_rate
     3.450565        3.450565
```

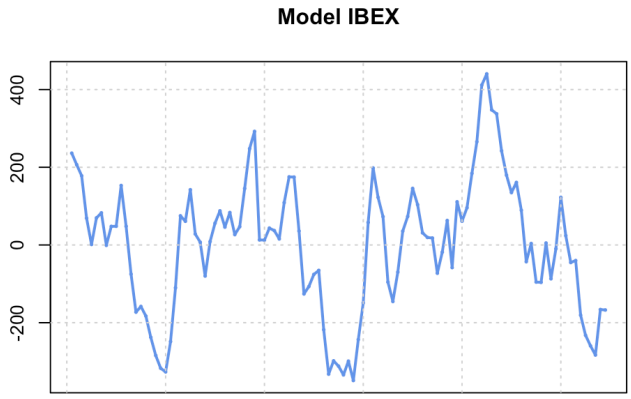
```
model_1 <-lm(IBEX~.-Week-Long_term_rate..., data=trainset)

lm_stats_1 <- summary(model_1)
lm_stats_1
```

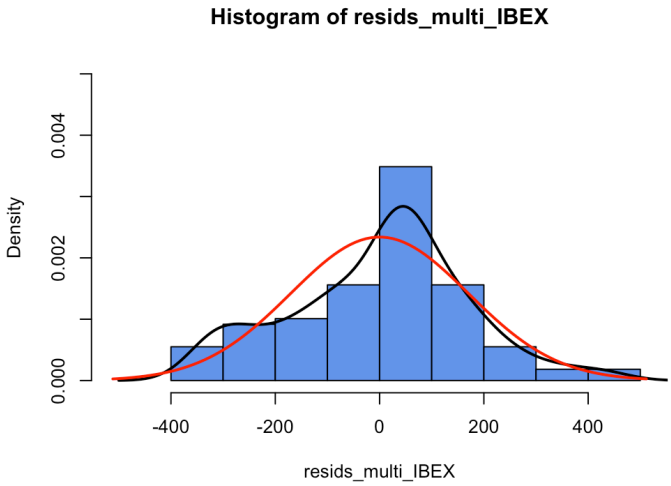
```
Call:
lm(formula = IBEX ~ . - Week - Long_term_rate..., data = trainset)

Residuals:
    Min      1Q  Median      3Q     Max
-348.84  -96.22   23.82   95.71  440.33

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     2914.49     369.78   7.882 3.03e-12 ***
Exchange_rate   2124.58     329.97   6.439 3.60e-09 ***
Short_term_rate  -137.02      12.06 -11.358  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 172.1 on 106 degrees of freedom
Multiple R-squared:  0.9053,    Adjusted R-squared:  0.9035
F-statistic: 506.8 on 2 and 106 DF,  p-value: < 2.2e-16
```

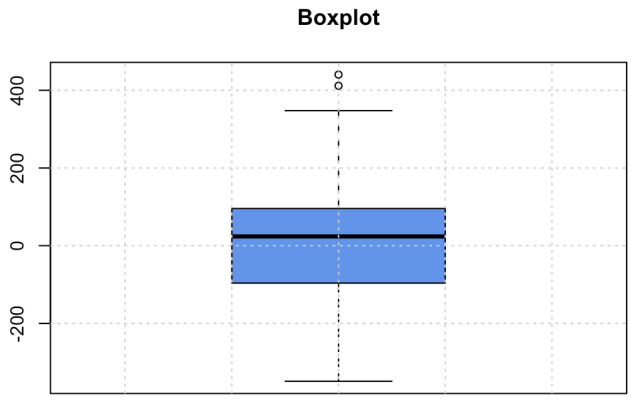
Summarizing our final regression model for the IBEX, we observe a lower Adjusted R-Squared error (now at 0.9035 compared to the previous 0.946), but also a slightly higher Residual Standard Error, from
129.3 to 172.1.

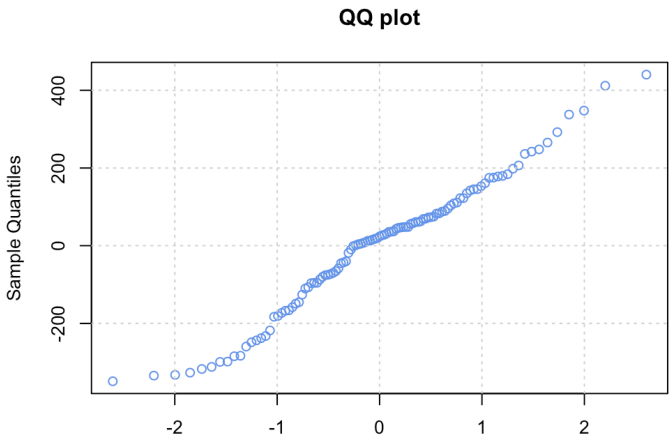
After treating multicollinearity, it is time to observe the behavior of our residuals.

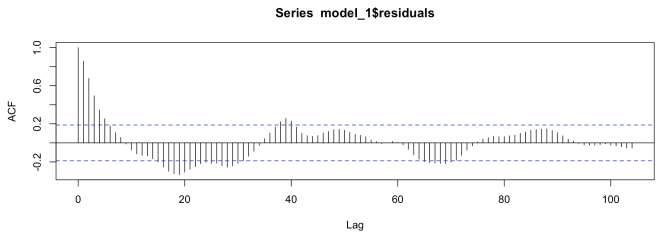
```
resids_multi_IBEX <- model_1$residuals
par(mfrow=c(2,2))
plot(resids_multi_IBEX, type='o', xlab='',ylab='',xaxt='n',lwd=2,pch=19,cex=0.1, main='Model IBEX', col='cornflowerblue'); grid()
hist(resids_multi_IBEX,prob=T,ylim=c(0,0.005),xlim=c(mean(resids_multi_IBEX)-3*sd(resids_multi_IBEX),mean(resids_multi_IBEX)+3*sd(resids_multi_IBEX)),col="cornflowerblue")
lines(density(resids_multi_IBEX),lwd=2)
mu<-mean(resids_multi_IBEX)
sigma<-sd(resids_multi_IBEX)
x<-seq(mu-3*sigma,mu+3*sigma,length=100)
yy2<-dnorm(x,mu,sigma)
lines(x,yy2,lwd=2,col="red")
boxplot(resids_multi_IBEX,main='Boxplot', col='cornflowerblue'); grid()
qqnorm(resids_multi_IBEX, col='cornflowerblue', main='QQ plot',xlab=' '); grid()
dev.off()

par(mfrow=c(2,1))
acf(model_1$residuals,nlags)
pacf(model_1$residuals,nlags)

ndiffs(model_1$residuals, alpha=0.05, test=c("adf")) # regular differences?

Box.test(model_1$residuals,lag=20)
shapiro.test(model_1$residuals)
```

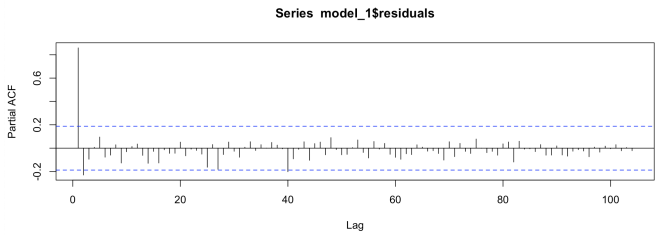













```
        Box-Pierce test

data:  model_1$residuals
X-squared = 246.09, df = 20, p-value < 2.2e-16
```

```
        Shapiro-Wilk normality test

data:  model_1$residuals
W = 0.97851, p-value = 0.07509
```

Based on the above graphs, we can notice that our **residuals** for the **regression model_1** are **stationary** and this can be verified by the ADF test. Also, they are not WN and the Box-Ljung test gives us a p-value lower than 0.05. Furthermore, by looking at the histogram as well as the Shapiro Test, we can conclude to normality.

model_1 <-lm(IBEX~.-Week-Long_term_rate..., data=trainset)

As a result, now that we have stationary residuals and not WN, we should move on to capture the structure in our residuals. In order to do so we move on to the 3 Question which is the combination of model 1 and 2.

Find the best regression model with the time series errors for the dependent variable "IBEX".

```
xreg_matrix <- cbind(trainset$Exchange_rate, trainset$Short_term_rate)
colnames(xreg_matrix) <- c('Exchange_rate','Short_term_rate')
model_3_1=arima(trainset$IBEX,order=c(0,1,0),xreg=xreg_matrix,include.mean=F)
summary(model_3_1)
```

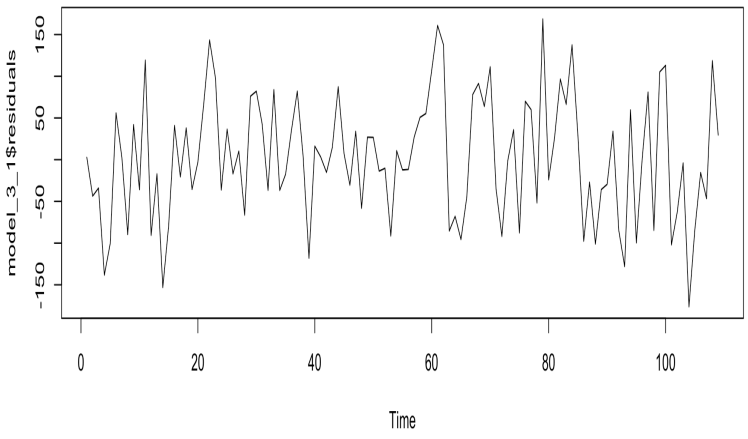
As in the first question we ended up that only the difference is enough for our dataset as it is already WN, we still have the same number of lags. Also, regarding the model from question 2, we started with 3 regressors but after dealing with multicollinearity, we ended up with 2 which are the ones we use at this model as well. Time to check the residuals of the combined model as well.

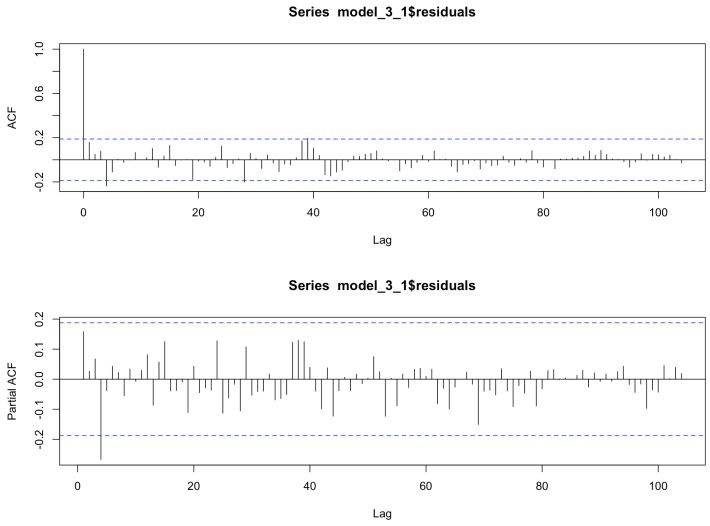
```
ts.plot(model_3_1$residuals)
par(mfrow=c(2,1))
acf(model_3_1$residuals,nlags)
pacf(model_3_1$residuals,nlags)
ndiffs(model_3_1$residuals, alpha=0.05, test=c("adf"))
Box.test(model_3_1$residuals,lag=20)
shapiro.test(model_3_1$residuals)
```

```
Call:
arima(x = trainset$IBEX, order = c(0, 1, 0), xreg = xreg_matrix, include.mean = F)

Coefficients:
      Exchange_rate  Short_term_rate
          998.6043         -53.7952
s.e.      454.4784          16.5553

sigma^2 estimated as 5506:  log likelihood = -618.37,  aic = 1242.75

Training set error measures:
                    ME     RMSE      MAE        MPE     MAPE      MASE      ACF1
Training set 2.112091 73.85862 60.39358 0.02844485 2.034576 0.9221698 0.1581153
```

```
        Box-Pierce test

data:  model_3_1$residuals
X-squared = 19.038, df = 20, p-value = 0.5194
```

```
        Shapiro-Wilk normality test

data:  model_3_1$residuals
W = 0.99051, p-value = 0.6495
```





After checking our residuals, we can still see that there is some structure in the residuals. In order to deal, with the lags out of bounce we start by introducing an MA(4) model by looking at the PACF.

```
model_3_2a=arima(trainset$IBEX,order=c(0,1,4),xreg=xreg_matrix,include.mean=F)
summary(model_3_2a)
```

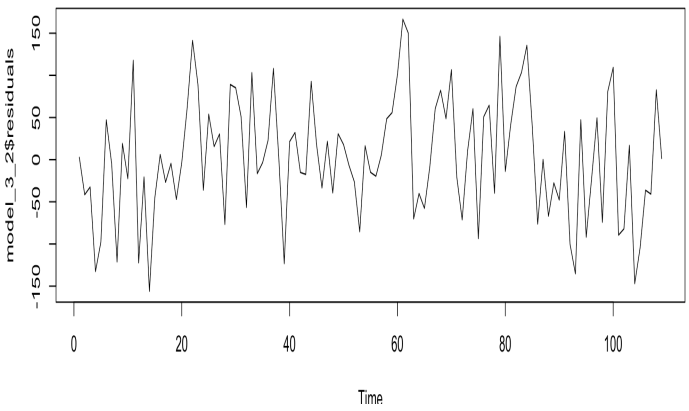
Retuning our model by removing the insignificant lags. In this case we only keep number 4.

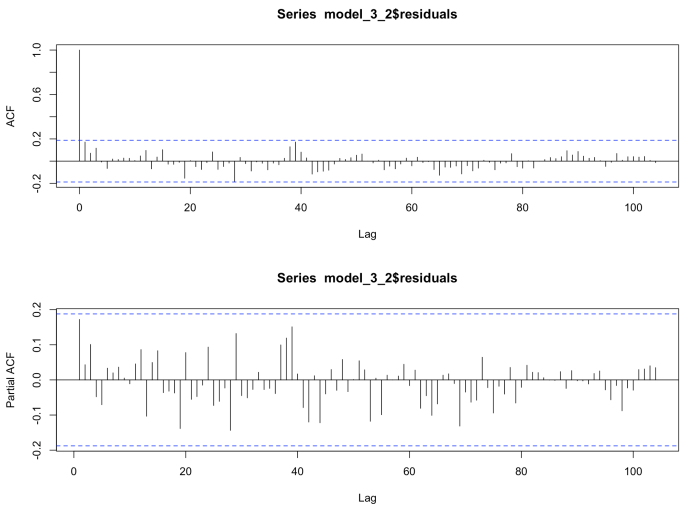
```
model_3_2 <- arima(trainset$IBEX,order=c(0,1,4),fixed=c(0,0,0,NA,NA,NA),xreg=xreg_matrix,include.mean=F)
summary(model_3_2)
```

Now that all our variables are significant let's move on checking the residuals of our model.

```
ts.plot(model_3_2$residuals)

par(mfrow=c(2,1))
acf(model_3_2$residuals,nlags)
pacf(model_3_2$residuals,nlags)

ndiffs(model_3_2$residuals, alpha=0.05, test=c("adf"))

Box.test(model_3_2$residuals,lag=20)
shapiro.test(model_3_2$residuals)
```

```
        Box-Pierce test

data:  model_3_2$residuals
X-squared = 11.853, df = 20, p-value = 0.921

        Shapiro-Wilk normality test

data:  model_3_2$residuals
W = 0.99136, p-value = 0.7233
```





Based on the above mentioned plots and tests. We now can conclude that our final combined model's residuals, are **WN** and normal, so **GWN**. Furthermore, below is mentioned the corresponding equation to the model.

$$Y\_t = -0.2473 * \varepsilon\_(t-4) + 1103.6853 * \text{〖Ex\_Rte〗}\_(t-1) - 56.4886 * \text{〖ShortT\_Rte〗}\_(t-1) + \varepsilon\_t$$

By taking the other approach, we can introduce an AR(4) model to our combined model.

```
model_3_10=arima(trainset$IBEX,order=c(4,1,0),xreg=xreg_matrix,include.mean=F)
summary(model_3_10)
```

```
Call:
arima(x = trainset$IBEX, order = c(4, 1, 0), xreg = xreg_matrix, include.mean = F)

Coefficients:
         ar1     ar2     ar3      ar4  Exchange_rate  Short_term_rate
      0.1735  0.0129  0.1113  -0.2795      1168.2528         -53.0591
s.e.  0.0921  0.0960  0.0967   0.0945       408.5806          14.5488

sigma^2 estimated as 4924:  log likelihood = -612.53,  aic = 1239.05

Training set error measures:
                    ME     RMSE      MAE        MPE     MAPE      MASE         ACF1
Training set 1.783294 69.84712 56.40653 0.01889181 1.919415 0.8612902 -0.009989404
```

As not all our variables are significant we retune the model. In this case we keep only AR(4)

```
model_3_20=arima(trainset$IBEX,order=c(4,1,0),fixed=c(0,0,0,NA,NA,NA),xreg=xreg_matrix,include.mean=F)
summary(model_3_20)
```

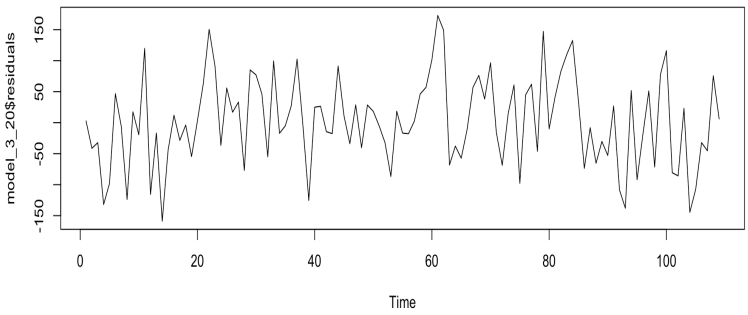
```
Call:
arima(x = trainset$IBEX, order = c(4, 1, 0), xreg = xreg_matrix, include.mean = F
    fixed = c(0, 0, 0, NA, NA, NA))

Coefficients:
      ar1  ar2  ar3      ar4  Exchange_rate  Short_term_rate
        0    0    0  -0.2507      1133.4407         -56.1656
s.e.    0    0    0   0.0958       424.0478          15.3000

sigma^2 estimated as 5168:  log likelihood = -615.09,  aic = 1238.19

Training set error measures:
                   ME    RMSE     MAE        MPE     MAPE      MASE      ACF1
Training set 2.07331 71.562 57.6307 0.01554332 1.951825 0.8799824 0.1706147
```

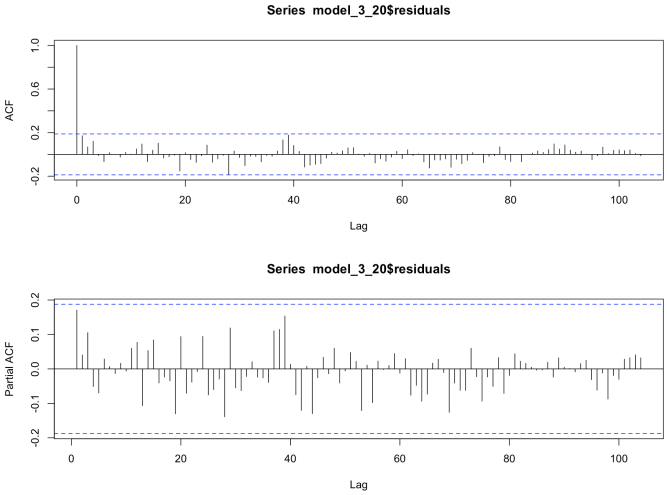
Now that all our variables are significant let's move on checking the residuals of our model.

```
ts.plot(model_3_20$residuals)
par(mfrow=c(2,1))
acf(model_3_20$residuals,nlags)
pacf(model_3_20$residuals,nlags)
ndiffs(model_3_20$residuals, alpha=0.05, test=c("adf"))
Box.test(model_3_20$residuals,lag=20)
shapiro.test(model_3_20$residuals)
```

```
        Box-Pierce test

data:  model_3_20$residuals
X-squared = 11.67, df = 20, p-value = 0.927
```

```
        Shapiro-Wilk normality test

data:  model_3_20$residuals
W = 0.99367, p-value = 0.9009
```





Based on the above mentioned plots and tests. We now can conclude that our final combined model's residuals, are **WN** and normal, so **GWN**. Furthermore, below is mentioned the corresponding **equation** to the model.

$$\#\ Y\_t = -\ 0.2507 * Y\_(t-4) + 1133.4407 * 〖Ex\_Rte〗\_(t-1) - 56.1656 * 〖ShortT\_Rte〗\_(t-1) + \varepsilon\_t$$

To sum up and in order to answer the 2 questions given at the description:

- This model does not maintain the same ARIMA structure found in question number 1, as in order to capture some structure in the residuals we end up with an AR(4) and MA(4), but only with these specific lags.
- This model maintains the same number of regressors found in question number 2

Choose among the four previous models the best one to explain variable "IBEX" using the "estimate of the residual variance" as the in-sample criterion.

In order to choose the best model to explain the IBEX variable, we need to look at the **estimate of the residual variance** of each model. In the picture below, the output of the according functions R clearly indicate Model 3.2 as the one with the lowest Residual Variance.

model_3_2 <- arima(trainset$IBEX,order=c(0,1,4),fixed=c(0,0,0,NA,NA,NA),xreg=xreg_matrix,include.mean=F)

MODEL	RESIDUAL VARIANCE	FORMULA
fit_1	6113.543	fit_0<-arima(yy,order=c(0,1,0))
model_1	29618.41	model_1 <-lm(IBEX~.-Week-Long_term_rate..., data=trainset)
model_3_2	5167.959	model_3_2<-arima(trainset$IBEX, order=c(0,1,4),fixed=c(0,0,0,NA,NA,NA),xreg=xreg_matrix,include.mean=F)
model_3_20	5168.47	model_3_20<-arima(trainset$IBEX, order=c(4,1,0),fixed=c(0,0,0,NA,NA,NA),xreg=xreg_matrix,include.mean=F)

```
fit_0$sigma2
summary(model_1)
# (172.1*172.1) = 29618.41
model_3_2$sigma2
model_3_20$sigma2
```

```
> fit_0$sigma2
[1] 6113.543
```

```
> model_3_2$sigma2
[1] 5167.959
> model_3_20$sigma2
[1] 5168.47
```

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

Based on the code below, by using the **forecast**() function, we manage to get at the same output both the point prediction **one step ahead**, but also the respective **confidence intervals** for an **80%** and **95%** level of confidence.

```
testnew <- cbind(0.781,7.6)
colnames(testnew) <- c('Exchange_rate','Short_term_rate')

model_pred<-predict(model_3_2,n.ahead = 1,newxreg = testnew)
model_pred$pred
model_pred$se
ts.plot(model_pred$pred)
forecast(model_3_2, h=1,xreg = testnew)
```

```
    Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
110       3364.276 3272.147 3456.405 3223.377 3505.175
```