Forecasting Homework 1

Yuhao Zhao, N17878783

September 30, 2015

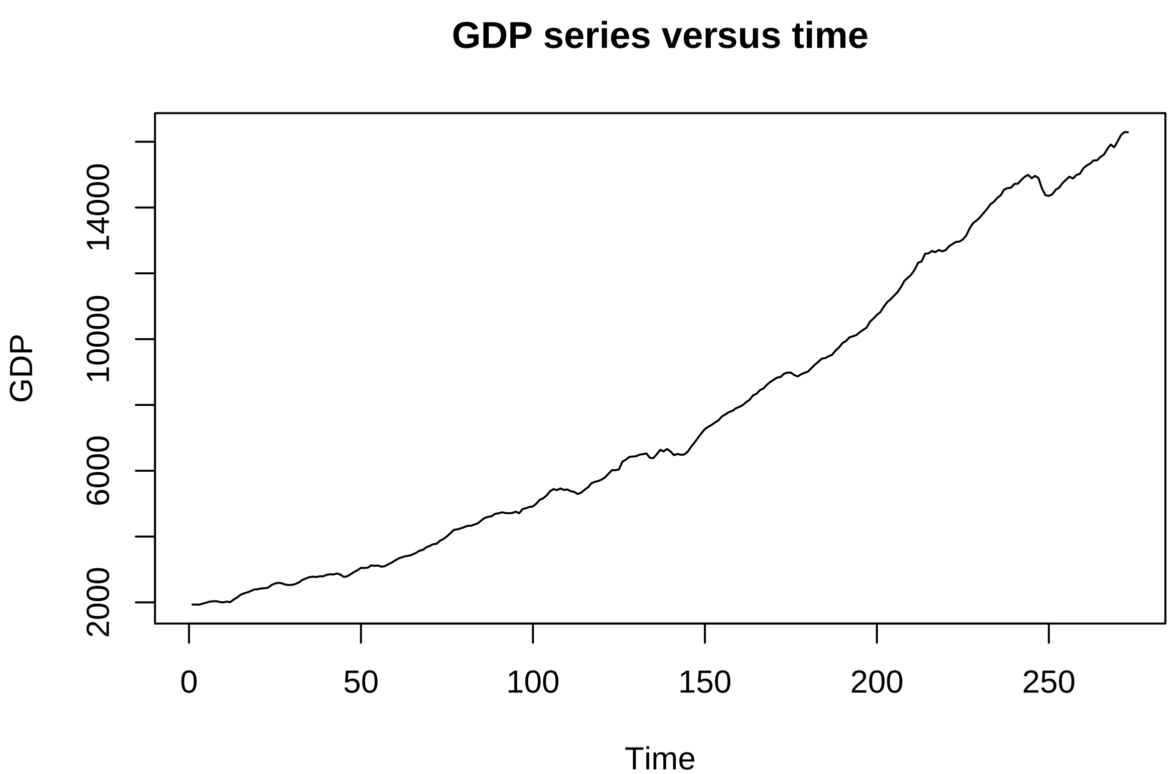
Data Import

```r
setwd("~/Desktop/Assignment/STAT-2302")
data = read.csv("http://people.stern.nyu.edu/churvich/Forecasting/Data/GDP.CSV")
attach(data)
library("itsmr")
```

Part (1)

a) GDP Plot

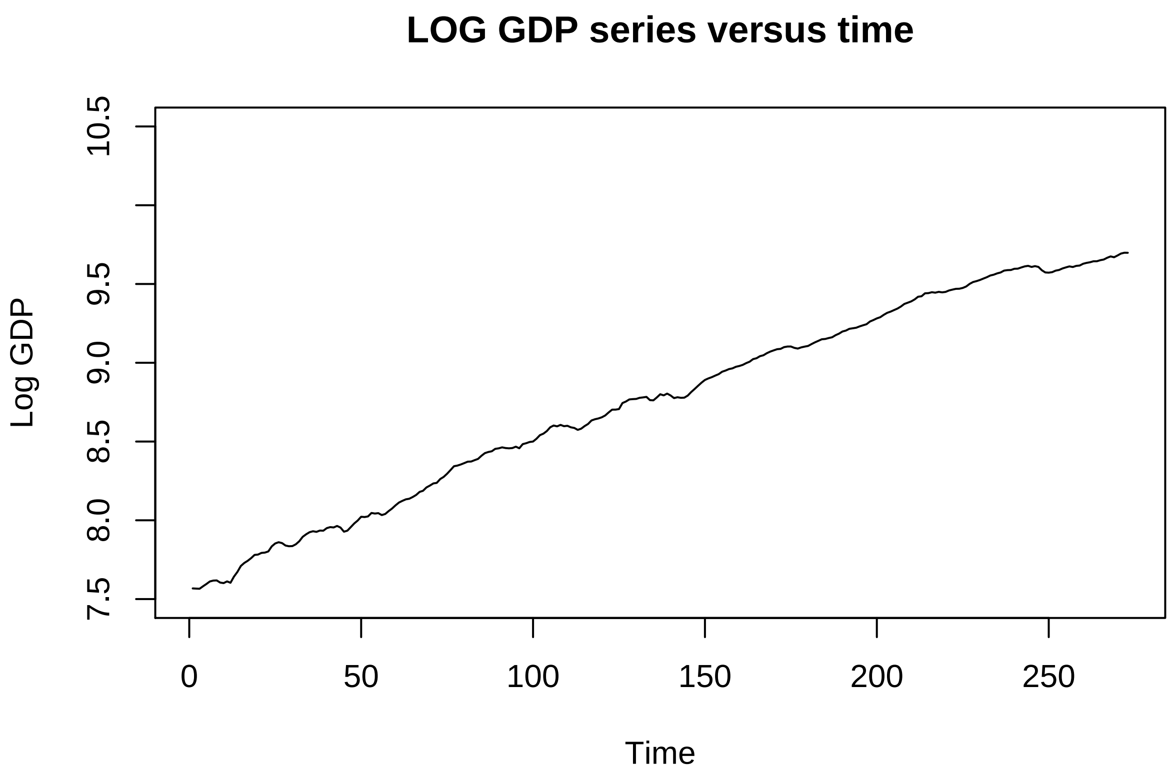
```r
ts.plot(GDP,ylab = "GDP",xlab = 'Time', main = 'GDP series versus time')
```



The GDP doesn't grow linearly over time from the time series plot. Instead, it seems to grow exponentially.

b) Log GDP Plot

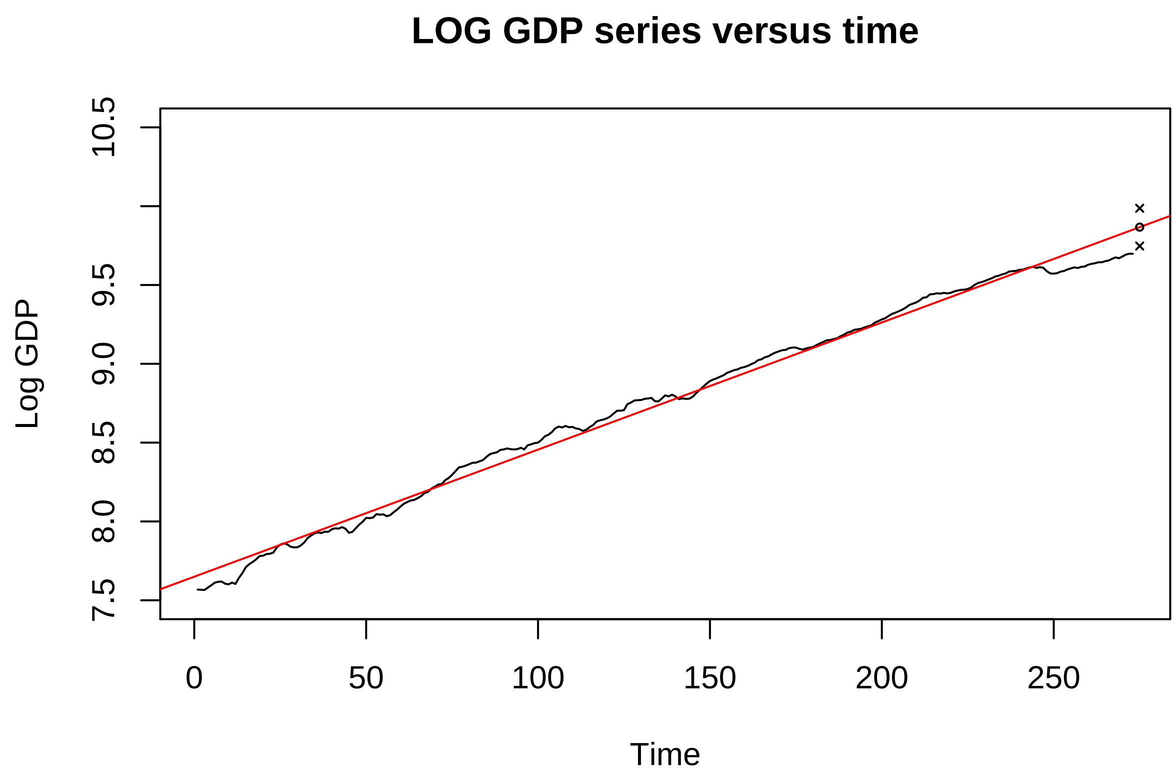
```r
ts.plot(log(GDP),ylab = "Log GDP",ylim = c(7.5,10.5),xlab = 'Time',
        main = 'LOG GDP series versus time')
```



The log GDP appear to grow linearly over time. This implies that the GDP series grow exponentially over time.

c) OLS Fit of LogGDP

```r
t = 1:length(GDP)
m2 = lm(log(GDP)~t)
ts.plot(log(GDP),ylab = "Log GDP",ylim = c(7.5,10.5),xlab = 'Time',
        main = 'LOG GDP series versus time')
abline(coef = m2$coef,col  ="red")
new = data.frame(t = 275)
new.value = predict(m2,new,interval = "predict",level = 0.95)
new.int = cbind(rep(as.numeric(new),3),as.numeric(new.value))
points(new.int[c(2,3),1],new.int[c(2,3),2],type = "b",col = "black",lty = 4,cex = 0.5,pch = 4)
points(new.int[1,1],new.int[1,2],type = "b",col = "black",lty = 4,cex = 0.5)
```




```r
res = t(new.int[,2])
colnames(res) = c("Lower Bound","Prediction","Upper Bound")
res
```

```
##      Lower Bound Prediction Upper Bound
## [1,]    9.866959   9.747103    9.986814
```

From the R output, the OLS forecast of the LogGDP for the second quarter of the year 2015 is 9.747103, and the 95% prediction inter val is [9.866959,9.986814]. If we add the prediction interval onto the plot,and compare the graph, it's clear that the prediction is not valid. The reason is that, firstly, the LogGDP of fourth quarter of 2014 is 9.698165, the prediction is supposed to be close to the LogGDP at the most recent time. However, the 95% prediction interval doesn't cover the 9.698165. Secondly, the length of the PI is too wide compared to the volatility of the time series. This mean the forecast is not valid.

d) Fit Plot

```r
summary(m2)
```

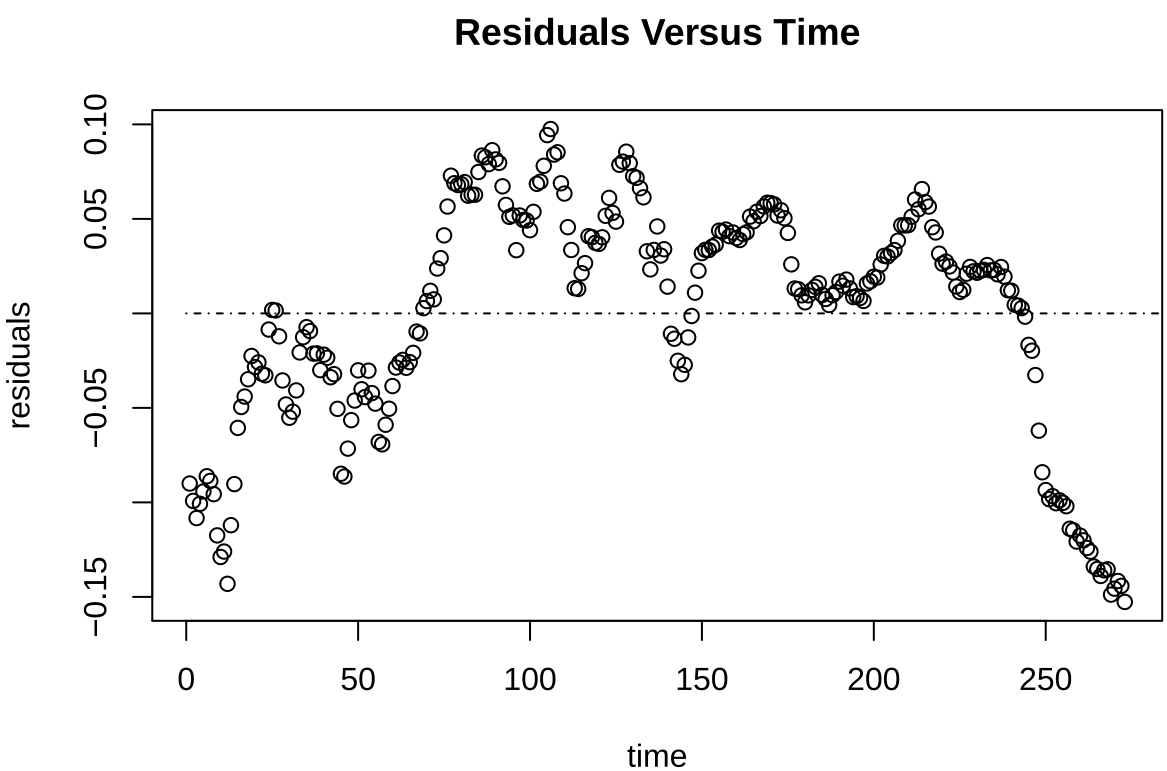
```
##
## Call:
## lm(formula = log(GDP) ~ t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.15267 -0.03215  0.01415  0.04435  0.09755
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.650e+00  7.335e-03  1042.9   <2e-16 ***
## t           8.063e-03  4.641e-05   173.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06043 on 271 degrees of freedom
## Multiple R-squared:  0.9911, Adjusted R-squared:  0.9911
## F-statistic: 3.019e+04 on 1 and 271 DF,  p-value: < 2.2e-16
```

Refer to the plot attached in c), as well as the summary table of the regression, the line fits well. The R square is 0.99 which is very high. Both coefficients are significant.

e) Resiuals v.s time plot

```r
plot(m2$residuals,main = 'Residuals Versus Time',xlab = 'time',ylab= 'residuals')
points(c(0,300),c(0,0),type = 'l',lty = 4)
```

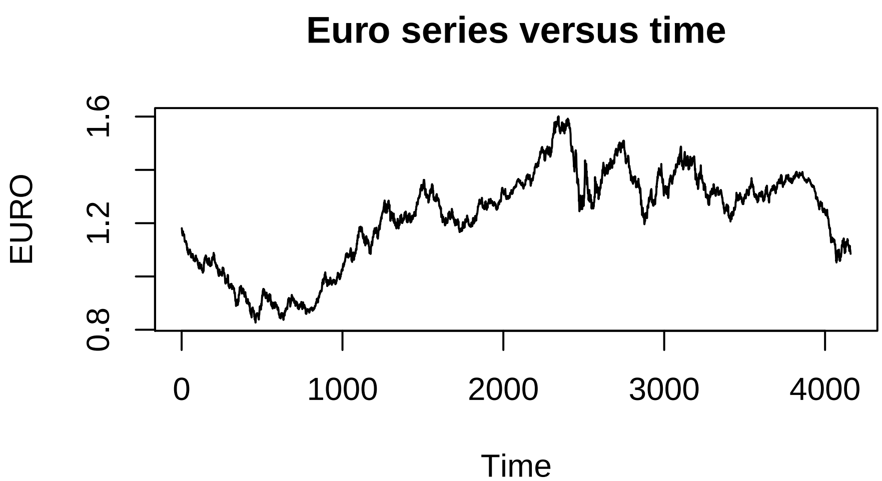


From the plot, we can observe that the residual is not normally distributed over 0. In particular, the residuals are clearly correlated and has unconstant variance. This can be a severe problem to OLS, since the validity of forecasting using OLS is satisfying the Gaussian Markov assumption, which requires uncorrelated residuals. Therefore, the occurrence of these problem will spoil the calidity of the forecast.

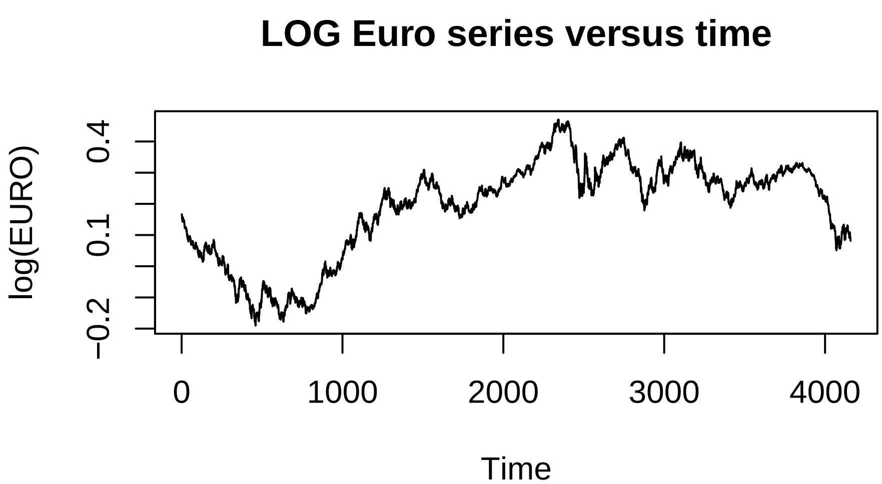
Part 2)

a) Data Plot

```
euro = read.csv("http://people.stern.nyu.edu/churvich/Forecasting/Data/EURO.CSV")
attach(euro)
ts.plot(EURO,main = 'Euro series versus time')
```




```
ts.plot(log(EURO),main = 'LOG Euro series versus time')
```



The plots show that, neither the Euro nor LogEuro are appropriate for straight-line model.

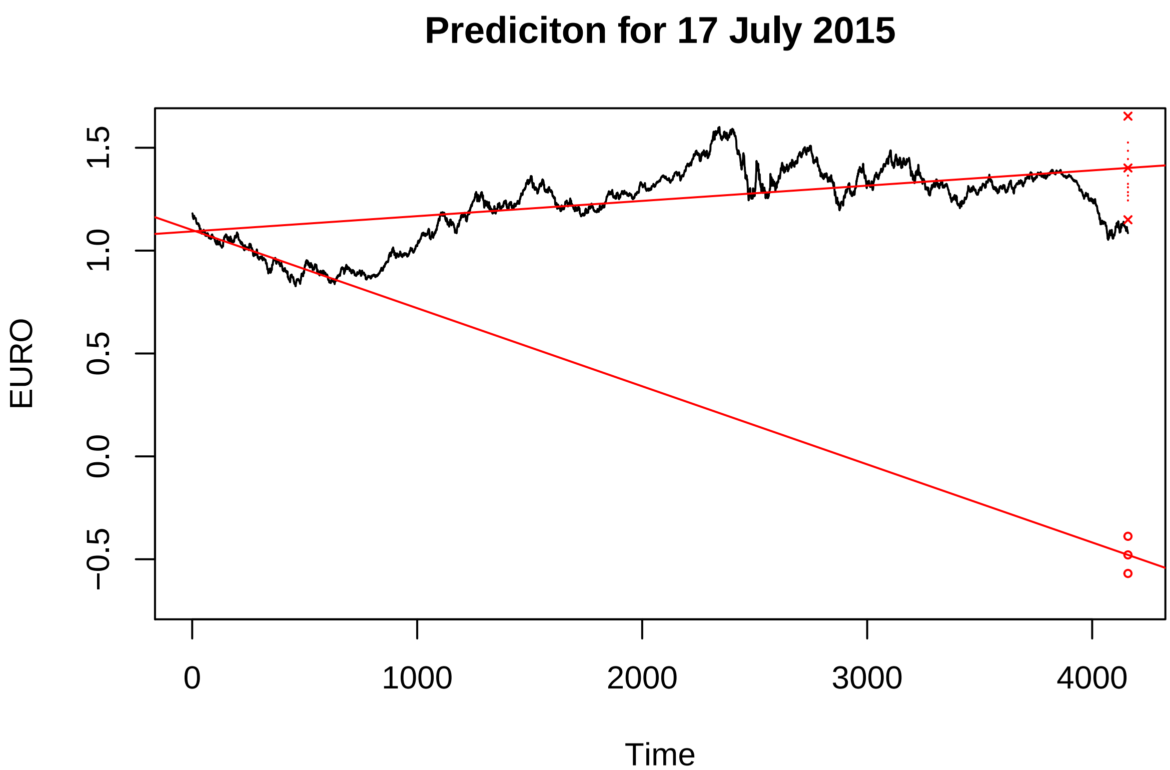
b) Predictions

```
t3 = c(1:700);m3 = lm(EURO[1:700]~t3)
t4 = c(701:4158);m4 = lm(EURO[701:4158]~ t4)

pred_3 = data.frame(t3 = 4159)
p_3 = predict(m3,pred_3,interval = "prediction")
```

```r
pred_4 = data.frame(t4 = 4159)
p_4 = predict(m4,pred_4,interval = "prediction")

ts.plot(EURO,ylim = c(-0.7,1.6),main = 'Prediciton for 17 July 2015')
abline(m3,col = 'red')
abline(m4,col = 'red')
points(rep(as.numeric(data.frame(t = 4159)),3),p_3,type = 'b',col = 'red',cex = 0.5)
points(rep(as.numeric(data.frame(t = 4159)),3),p_4,type = 'b',col = 'red'
       ,lty = 3,cex = 0.5,pch = 4)
```



From the plot, neither of the interval succeed in containing the actual values for 17 Jul 2015. In this problem, a ordinary linear regression is not appropriate because the data shows clear correlation. In Problem 1 we find out that if the data are not independent, the forecast can be extramely imprecise.