ECN190 Homework 5

Computer Problems

Kevin Chen (914861432) John Mayhew (914807483)

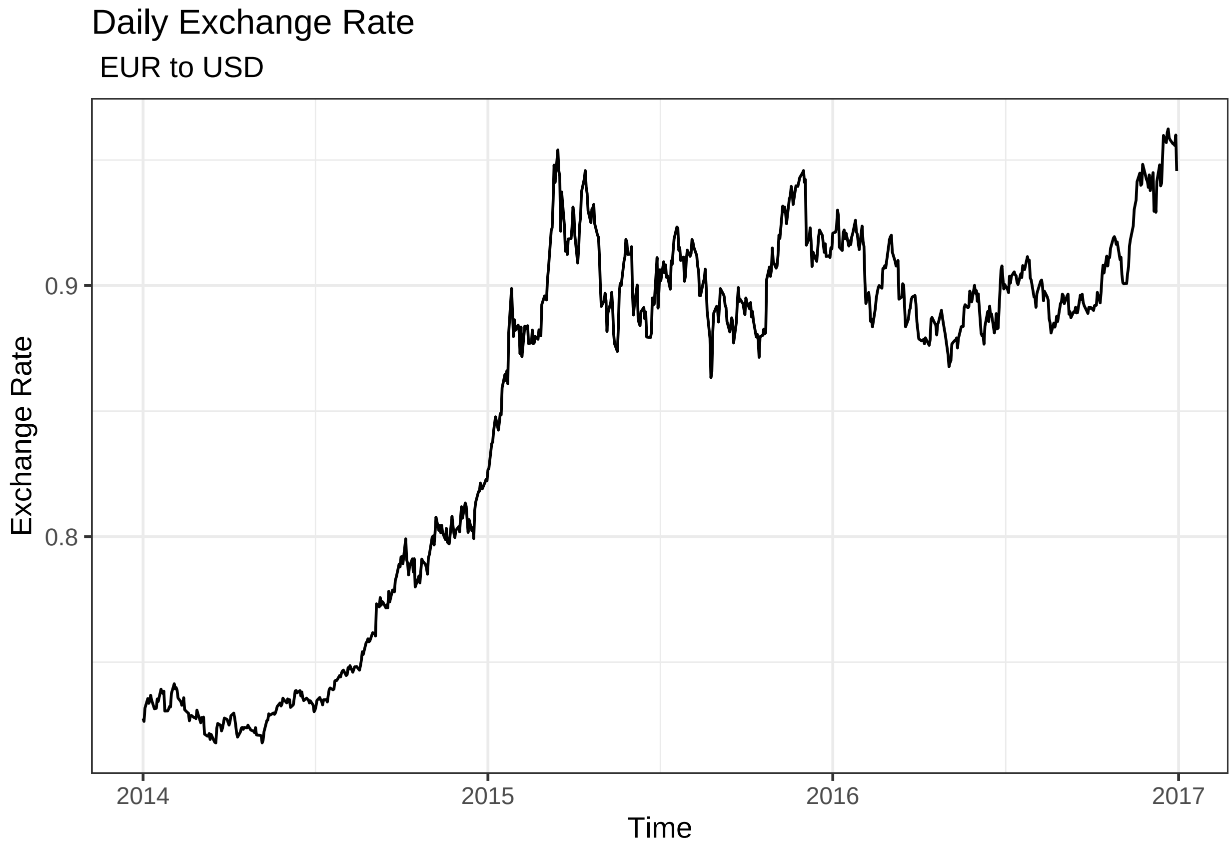
5/14/2020

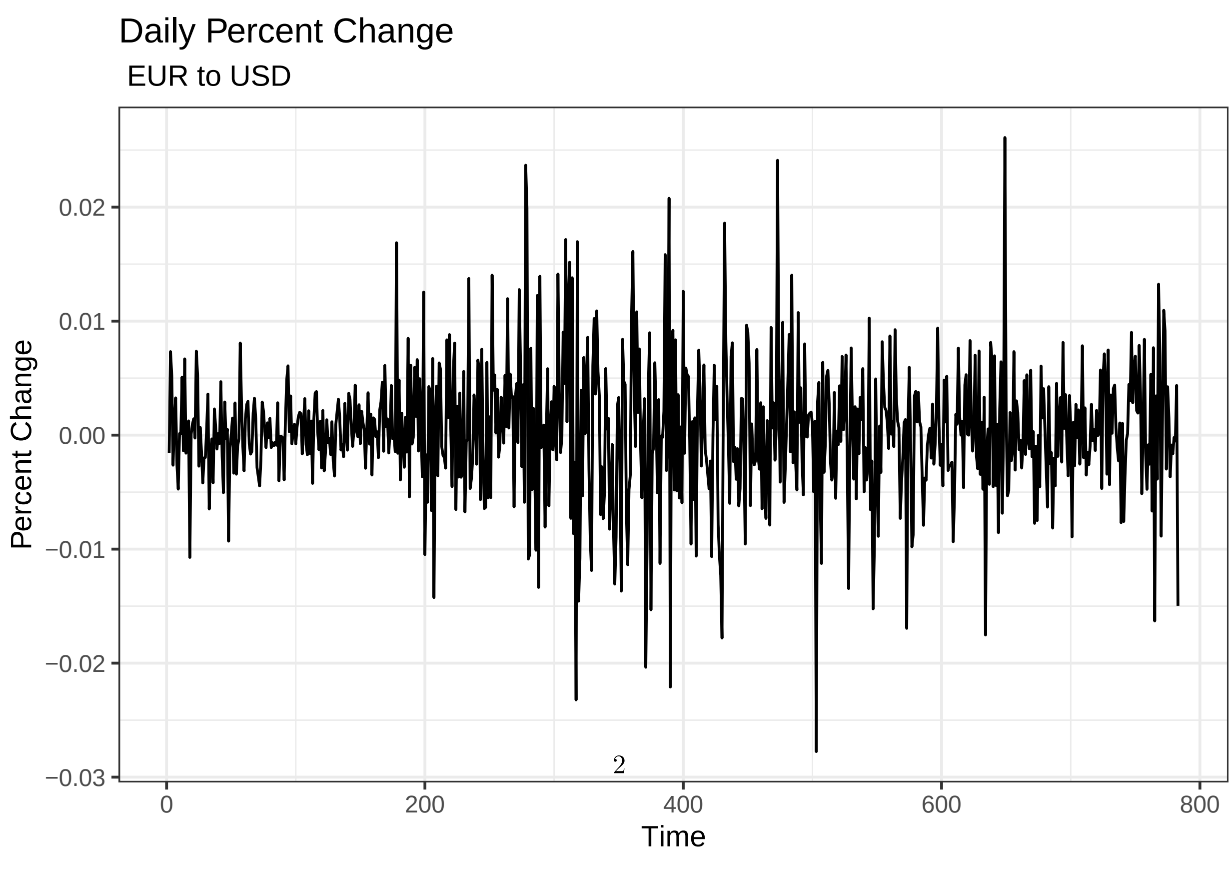
1. What is the exchange rate in the first trading day of 2014 (2014-01-01)? Explain the exchange rate number you obtained.

```
##            EUR=X.Close
## 2014-01-01     0.72754
```

The exchange rate in the first trading day of 2014 is 0.72754. This means that the closing exchange rate for January 1, 2014 was 1 U.S. dollar to 0.7275 euros.

2. First plot out the daily exchange rate series. Then generate a time series that represents the percentage change in daily closing rate. Plot out the time series. What happened to the Euro/US exchange rate in the second half of 2014 and first half of 2015?





The exchange rate in the second half of 2014 and first half of 2015 increased. You can see from both plots that the exchange rate and the percent change of the exchange rate increased during that period. In the percent change plot, the perent change was between -10% and 10% prior to July 1st, 2014, then the percent changed increased to over -20% and 20% change in days after the second half of 2014

3. Feed the series to the "auto.arima" command using the default information criteria. What model does your output suggest? What if you change the information criteria to BIC?

```
## Series: close$percentchange
## ARIMA(0,0,1) with non-zero mean
##
## Coefficients:
##           ma1   mean
##       -0.0519  4e-04
## s.e.   0.0356  2e-04
##
## sigma^2 estimated as 3.427e-05:  log likelihood=2911.3
## AIC=-5816.6   AICc=-5816.57   BIC=-5802.61
```

The auto.arima function is predicting that the series is MA(1).

```
## Series: close$percentchange
## ARIMA(0,0,0) with zero mean
##
## sigma^2 estimated as 3.44e-05:  log likelihood=2908.83
## AIC=-5815.66   AICc=-5815.65   BIC=-5810.99
```

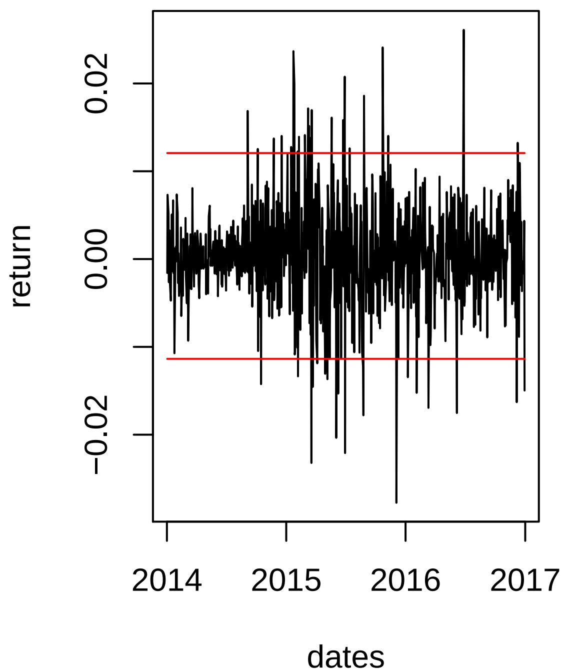
If the information criteria is changed to BIC, then an ARMA(0, 0) model is selected by the auto.arima function.

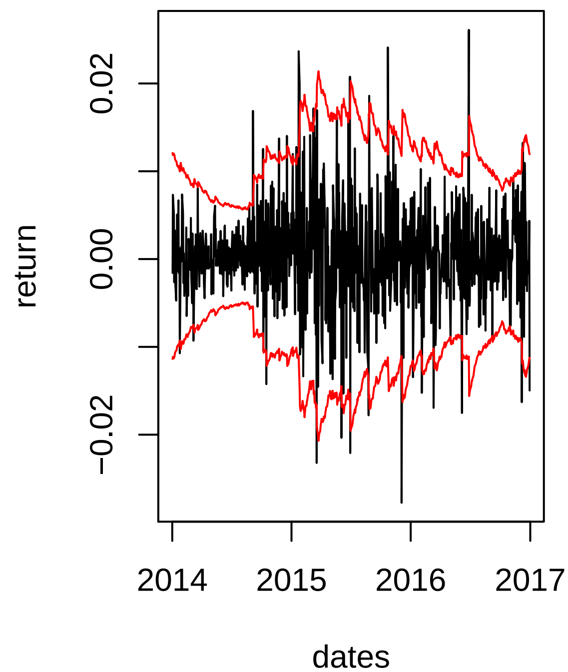
4. Fit your time series of % changes in daily exchange rate with an ARIMA(0,0,0) model with GARCH(1,1) error. Write down your fitted model for the volatility (variance of error term) at time period t.

$\mu = 3.1493473 \times 10^{-4}$, $\omega = 1.6178001 \times 10^{-7}$, $\alpha_1 = 0.0485819$, $\beta_1 = 0.9497417$

$\sigma_t^2 = 1.6178001 \times 10^{-7} + 0.0485819 u_{t-1}^2 + 0.9497417 \sigma_{t-1}^2$

5. Plot out the daily returns against the 95% in-sample forecast interval.





6. Obtain the 95% forecast interval for the first trading date of 2017.

95% Forecast Interval for Jan 1st, 2017 = [-0.0125931851647651, 0.0132230546286391]

Appendix

```
library(quantmod)
library(tidyverse)
library(forecast)
library(anytime)
library(fGarch)

HW5CP = new.env()
getSymbols("EUR=X", env = HW5CP, src="yahoo", from = "2014-01-01", to = "2016-12-31")

#1
EUR <- HW5CP[["EUR=X"]]
close <- EUR$`EUR=X.Close`
close[1,1]

#2
close <- EUR$`EUR=X.Close`
close$t = 1:nrow(close);close = close[,c(2,1)]
close = as.data.frame(close)
close$date = rownames(close)
colnames(close) = c("t", "exr", "date")
close$date = anydate(close$date)
close %>% ggplot(aes(x = date, y = exr)) + geom_line() + theme_bw() +
  ggtitle("Daily Exchange Rate", subtitle = " EUR to USD") + scale_x_date("Time") +
  scale_y_continuous("Exchange Rate")

close$percentchange = (close$exr - Lag(close$exr, 1))/Lag(close$exr, 1)
close[2:nrow(close),] %>% ggplot(aes(x = t, y = percentchange)) + geom_line() +
  theme_bw() + ggtitle("Daily Percent Change", subtitle = " EUR to USD")+
  scale_x_continuous("Time") + scale_y_continuous("Percent Change")

#3
auto.arima(close$percentchange)
auto.arima(close$percentchange, ic = "bic")

#4
results <- garchFit(formula = ~ garch(1, 1), data = close$percentchange[-1], trace = F)
mu = coef(results)[1]
omega = coef(results)[2]
alpha1 = coef(results)[3]
beta1 = coef(results)[4]

#5
return = close$percentchange[-1]
dates = close$date[-1]
par(mfrow = c(1,2))
volatility_longterm<-rep(sd(return),length(return))
plot(dates,return,type="l")
lines(dates,mean(return)+2*volatility_longterm,col="red")
lines(dates,mean(return)-2*volatility_longterm,col="red")
volatility_GARCH <- volatility(results, type = "sigma")
plot(dates,return,type="l")
```

```r
lines(dates,mean(return)+2*volatility_GARCH,col="red")
lines(dates,mean(return)-2*volatility_GARCH,col="red")
par(mfrow = c(1,1))

#6
predict(results, level=95, interval=TRUE, plot = T, n.ahead = 1)[4:5]
lower = predict(results, level=95, interval=TRUE, plot = T, n.ahead = 1)[4]
upper = predict(results, level=95, interval=TRUE, plot = T, n.ahead = 1)[5]
cat("\\newpage")
```