6 Month Forecast of the Global Economic Policy Uncertainty Index

Oscar Olaya

12/29/2022

Loading data and libraries

```
load('C:\\Users\\oscar\\Desktop\\STAT 631\\Final2022.Rdata')
source('C:\\Users\\oscar\\Desktop\\STAT 631\\GARCH_RFunctions.R')
```

```
## Warning: package 'rugarch' was built under R version 4.1.3
```

```
## Loading required package: parallel
```

```
##
## Attaching package: 'rugarch'
```

```
## The following object is masked from 'package:stats':
##
##     sigma
```

```
source('C:\\Users\\oscar\\Desktop\\STAT 631\\GARCH_plotFunctions.R')
library(quantmod); library(rugarch);library(forecast);library(urca)
```

```
## Warning: package 'quantmod' was built under R version 4.1.3
```

```
## Loading required package: xts
```

```
## Warning: package 'xts' was built under R version 4.1.2
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.1.3
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: TTR
```

```
## Warning: package 'TTR' was built under R version 4.1.2
```

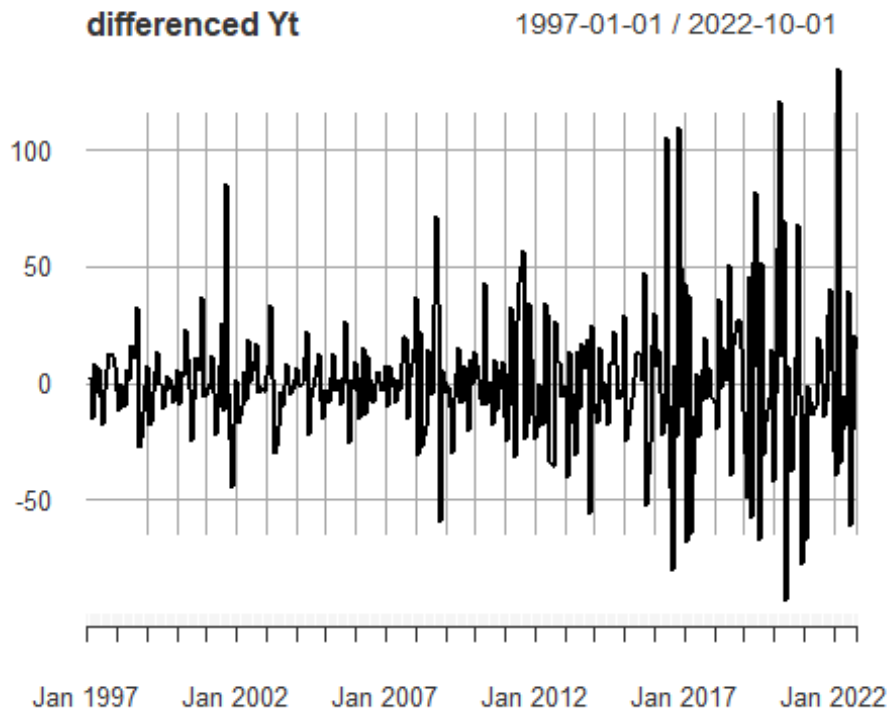
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Warning: package 'forecast' was built under R version 4.1.3

## Warning: package 'urca' was built under R version 4.1.2

cat("Starting at:");

## Starting at:

head(Yt,2)

##             GEPUPPP
## 1997-01-01 80.19395
## 1997-02-01 82.37771

cat("Ending at:");

## Ending at:

tail(Yt,2)

##             GEPUPPP
## 2022-09-01 281.1915
## 2022-10-01 295.8545
```

Plots

- From the plots we can observe that Yt has an upward trend and it seems to be a non stationary random walk process. After taking the difference we can observe that the series has a constant mean, but its variance is not constant. This suggests a log transformation.

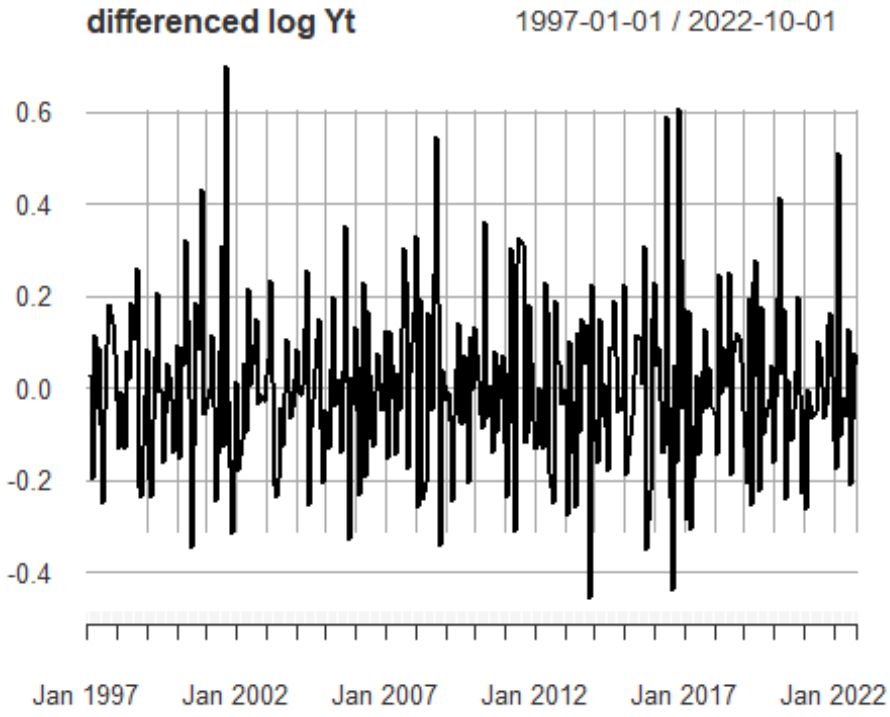
```
plot(Yt, yaxis.right = F);plot(diff(Yt), yaxis.right = F, main = "differenced Yt")
```



Plots after log tranforming the data

- It seems to be stationary after log transforming Yt and taking the difference

```
Xt = log(Yt);dXt = diff(Xt)[-1,];
plot(Xt, yaxis.right = F, main = "log Yt");plot(diff(Xt), yaxis.right = F,
                                                    main = "differenced log Yt
")
```



Checking if Xt is a unit root process (ADF unit root test on Xt)

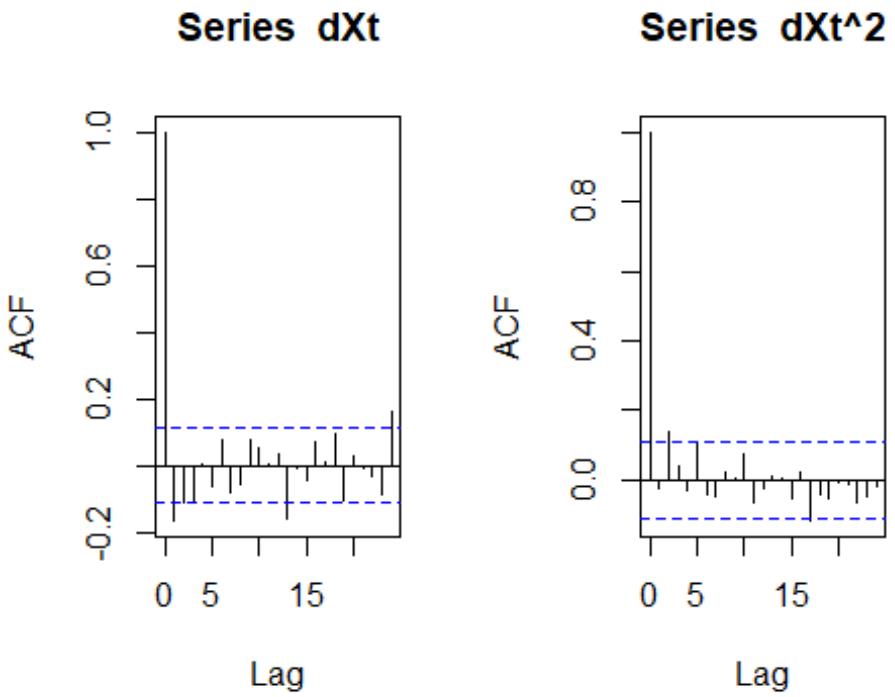
- The ADF test with lag parameter p = 3 fails to reject the null of a unit root process, confirming that Xt is a unit root process.

```
n = dim(Xt)[1]; p.max = round(12*(n/100)^.25)
adf = ur.df(Xt,type = "drift", lags = p.max, selectlags = "AIC" );
cat("ADF test with maximum lag", p.max, "and lag selected by AIC ", adf@testr
eg$df[1]-2,
       "\nTest Statistic = ", adf@teststat[1],"\nCritical values");

## ADF test with maximum lag 16 and lag selected by AIC  3
## Test Statistic =  -1.667552
## Critical values

adf@cval[1,];

##  1pct  5pct 10pct
## -3.44 -2.87 -2.57

cat("\nReject or not Reject the null:");

##
## Reject or not Reject the null:

adf@teststat[1] < adf@cval[1,]

##  1pct  5pct 10pct
## FALSE FALSE FALSE
```

Checking if Xt has GARCH effects (Ljung-Box test)

- We conclude that Xt is of ARIMA(p, 1, q) class without GARCH effects

```
par(mfrow = c(1,2))
acf(dXt);acf(dXt^2);
```

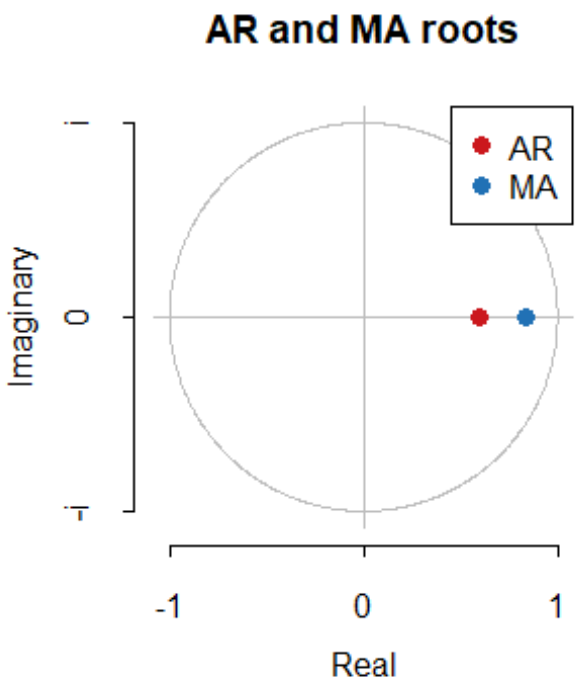



```
test1 = sapply(4:8,function(u) Box.test(dXt,u, type = "L")[c(1,3)])
test2 = sapply(4:8,function(u) Box.test(dXt^2,u, type = "L")[c(1,3)])
colnames(test1) = colnames(test2) = paste("df=",4:8)
cat("Ljung-Box test of dXt");

## Ljung-Box test of dXt

test1

##           df= 4       df= 5       df= 6       df= 7       df= 8
## statistic 15.79286    17.04312    19.08239    21.0343     21.98611
## p.value   0.003310087 0.004418738 0.004026731 0.003719712 0.004941655

cat("Ljung-Box test of squared dXt");

## Ljung-Box test of squared dXt

test2

##           df= 4     df= 5      df= 6      df= 7     df= 8
## statistic 7.083376  10.6498    11.20885   11.88893  12.05322
## p.value   0.1315473 0.05878292 0.08213212 0.1042712 0.1488452
```

Selecting an adequate model

- Both AIC and BIC selected the same model, ARIMA(1,1,1). Both coefficients are significant, the AR and MA roots are not close numerically or graphically. The selected model does not have parameter redundancy.

```
aic.X = auto.arima(Xt, d = 1, ic = "aic")
aic.X

## Series: Xt
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1      ma1
##       0.592  -0.8374
## s.e.  0.108   0.0762
##
## sigma^2 = 0.02795:  log likelihood = 115.12
## AIC=-224.24   AICc=-224.16   BIC=-213.04

plot_Roots(coef(aic.X))
```



Checking the residuals of ARIMA(1,1,1)

- Its residuals are not correlated

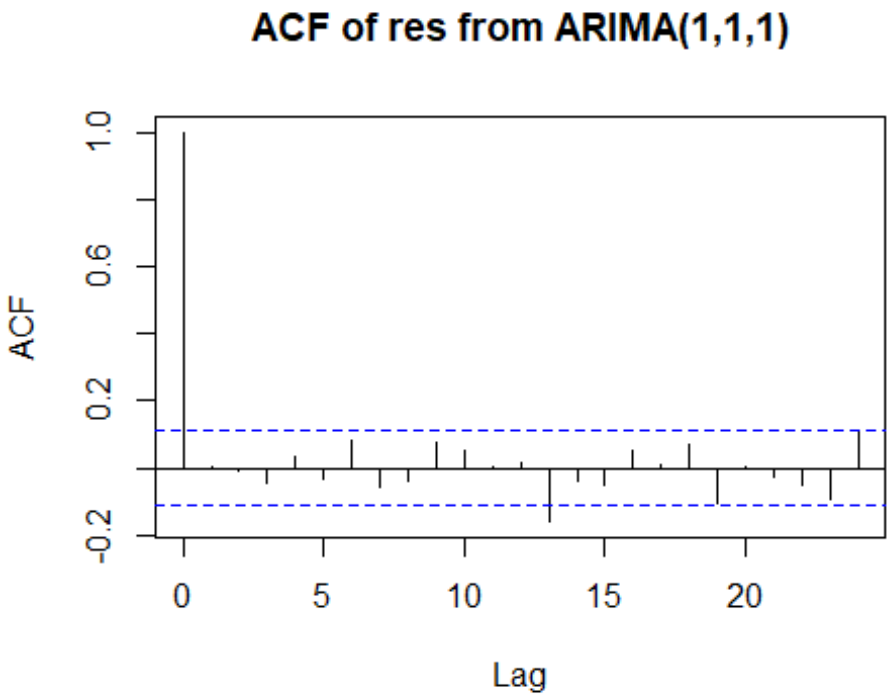
```
res = aic.X$residuals
restest = sapply(4:8,function(u) Box.test(res,u + 2, type = "L", 2)[c(1,2,3)]
)
colnames(restest) = paste("lags =",(4:8) + 2);
restest

##           lags = 6  lags = 7  lags = 8  lags = 9  lags = 10
## statistic 3.29694   4.381852  4.893757  6.638368  7.535605
```

```
## parameter 4         5         6         7         8
## p.value   0.5094172 0.4958451 0.5575102 0.4674778 0.4800942
```

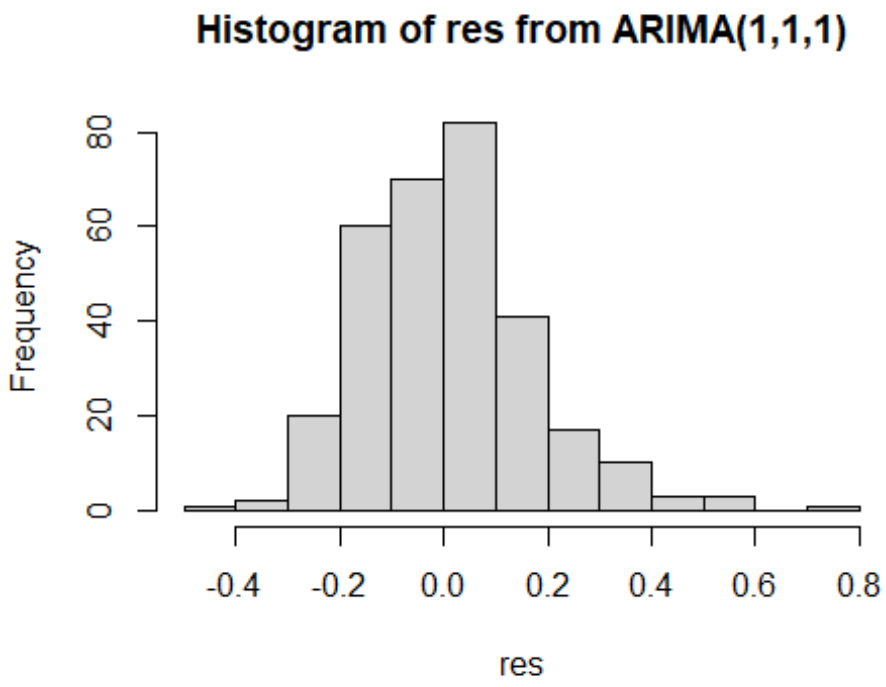
Finding a suitable distribution for the errors (white noise) in the selected model

- Both AIC and BIC select Johnson's SU distribution for the white noise of ARIMA(1,1,1) for Xt
- Plot with the residuals from ARIMA(1,1,1) shows the selected distribution fits well. The final model for Xt = log Yt is ARIMA(1,1,1) with i.i.d. white noise from Johnson's SU distribution.

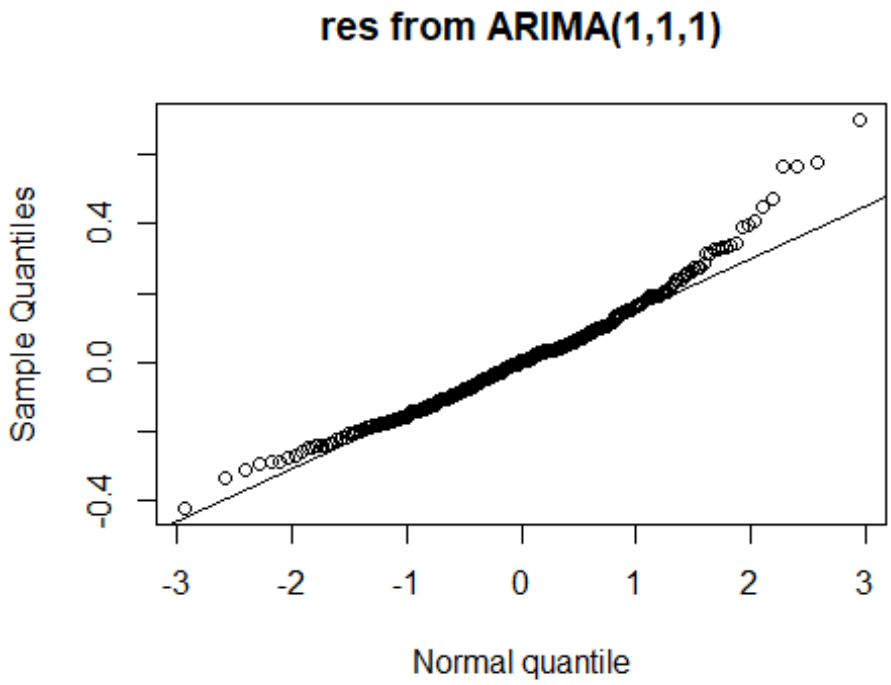
```
res = resid(aic.X)

acf(res, main = "ACF of res from ARIMA(1,1,1)")
```




```
hist(res, main = "Histogram of res from ARIMA(1,1,1)")
```




```
qqnorm(res, main = "res from ARIMA(1,1,1)", xlab = "Normal quantile")
qqline(res)
```



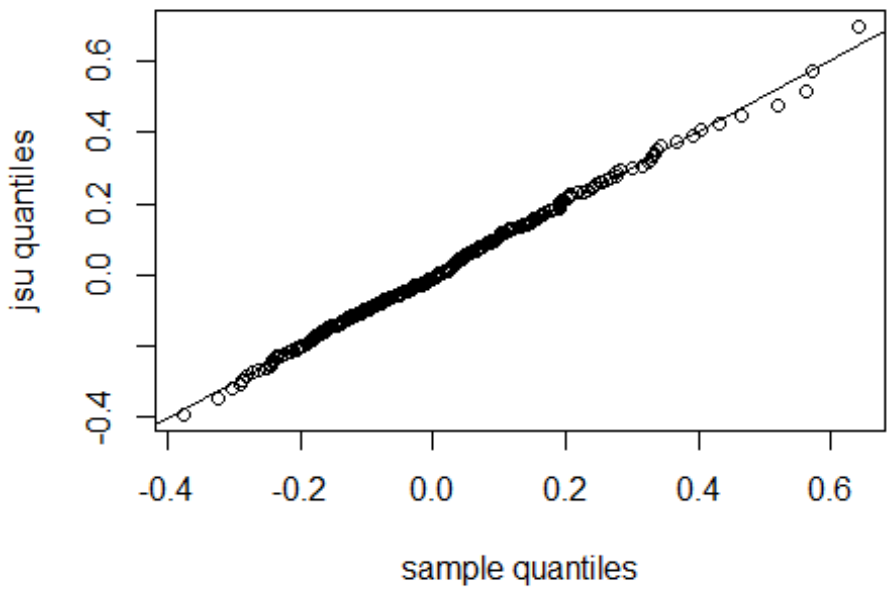
```
dists = c("snorm", "snorm", "sstd","sged", "jsu", "nig")
AIC = BIC = Inf; aic.dist = bic.dist = c()
for(i in 1:length(dists)){
   out = fitdist(dists[i],res)
   aic = 2*tail(out$val,1) + 2*length(out$par)
   bic = 2*tail(out$val,1) + log(n)*length(out$par)
   if(aic < AIC) {AIC = aic; aic.dist = list(dist = dists[i], coef = out$par)
}
   if(bic < BIC) {BIC = bic; bic.dist = list(dist = dists[i], coef = out$par)
}
   }
```

```
## Warning in .safefunx(tmpv, .solnp_fun, .env, ...):
## solnp-->warning: Inf detected in function call...check your function

## Warning in .safefunx(tmpv, .solnp_fun, .env, ...):
## solnp-->warning: Inf detected in function call...check your function
```

```
cat("AIC selects", paste(aic.dist$dist), "\t\tBIC selects", paste(bic.dist$di
st));
```

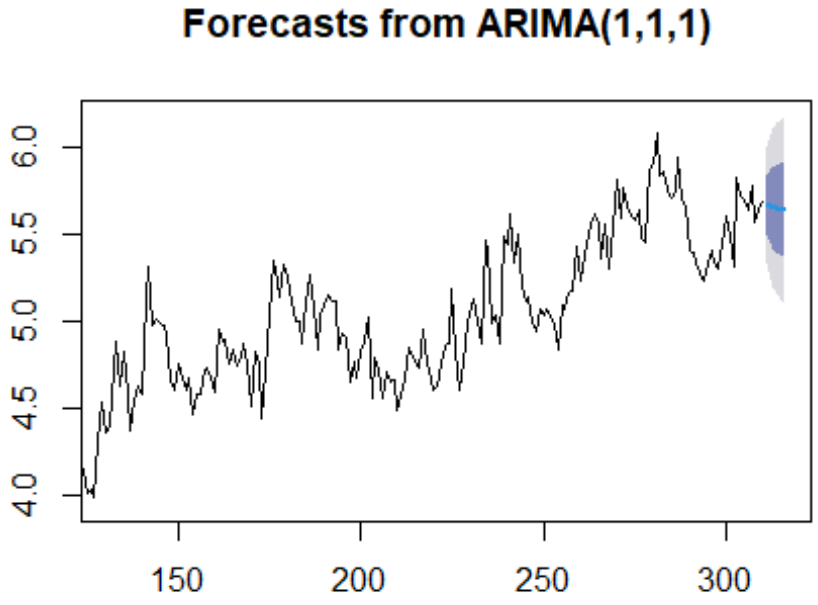
```
## AIC selects jsu      BIC selects jsu
```

```
dist = aic.dist$dist; coef = aic.dist$coef
ps = ((1:n)-1/2)/n
xs = quantile(res,ps)
ys = qdist(dist,ps,mu = coef[1],sigma = coef[2], skew = coef[3], shape = coef
[4])

plot(xs,ys, xlab = "sample quantiles", ylab = "jsu quantiles" )
abline(lsfit(xs,ys)$coef)
```



6 month forecast with 95% prediction intervals

```
plot(forecast(aic.X, h = 6, level = c(68,95)), xlim = c(131,(n+6)))
```



```r
qs = qdist(dist,c(.025, .0975), sigma = coef[2], skew = coef[3], shape = coef
[4])
pred = predict(aic.X, n.ahead = 6)
pi = sapply(1:2, function(u) pred$pred + qs[u]*pred$se)
pi = cbind(pred$pred, pi)
colnames(pi) = c("point pred", "low95", "hi95")
cat("1-6 step ahead for Xt");pi

## 1-6 step ahead for Xt

## Time Series:
## Start = 311
## End = 316
## Frequency = 1
##     point pred    low95     hi95
## 311   5.669070 5.622809 5.636606
## 312   5.656758 5.598805 5.616089
## 313   5.649469 5.585026 5.604245
## 314   5.645154 5.576314 5.596845
## 315   5.642599 5.570375 5.591915
## 316   5.641087 5.566022 5.588409

cat("1-6 step ahead for Yt");exp(pi)

## 1-6 step ahead for Yt

## Time Series:
## Start = 311
## End = 316
## Frequency = 1
##     point pred    low95     hi95
## 311   289.7650 276.6656 280.5092
## 312   286.2192 270.1034 274.8125
## 313   284.1405 266.4071 271.5769
## 314   282.9170 264.0964 269.5746
## 315   282.1952 262.5326 268.2489
## 316   281.7687 261.3922 267.3101
```