

```

### pre-processing
petro <- read.csv("petro.csv", header = F)

library(fBasics)
basicStats(petro)
petro <- ts(petro, frequency = 12)
ts.plot(petro)
acf(petro, lag = 100, main = "ACF")
pacf(petro, lag = 100)
d <- as.matrix(petro)
plot(d[2:328], d[1:327], xlab = "x(t-1)", ylab = "x(t)") #linearity test
lines(d[2:328], d[2:328])

#####decompose#####
petro.decom <- decompose(petro)
plot(petro.decom)

###stationary test###
library(tseries)
adf.test(petro)
pp.test(petro)
kpss.test(petro)
spectrum(petro)

#####difference&test#####
petrodiff <- diff(petro)
plot(petrodiff)
spectrum(petrodiff)

adf.test(petrodiff)
pp.test(petrodiff)
kpss.test(petrodiff)

Box.test(petrodiff, lag = 6, type = "Ljung-Box")
Box.test(petrodiff, lag = 12, type = "Ljung-Box")

acf(petrodiff, lag = 100)
pacf(petrodiff, lag = 100)

#####arima#####
library(forecast)
library(zoo)
auto.arima(petrodiff)

```

```

aic.t <- rep(NA, 9)
for (i in 1:3){
  for (j in 1:3){
    aic.t[3*(i-1)+j] <- arima(petrodiff, order = c(i,0,j), include.mean = T)$aic
  }
}
aic.t

```

```

library(TSA)
eacf(petrodiff)
res <- residuals(auto.arima(petrodiff))
plot(res)
Box.test(res, lag = 6)
Box.test(res, lag = 12)
tsdiag(auto.arima(petrodiff))
jarque.bera.test(res)
qqnorm(res)
qqline(res)

```

```

####arch-test####
res2 <- res^2
Box.test(res2, lag = 12, type = "Ljung-Box")
acf(res2)
pacf(res2)
library(FinTS)
ArchTest(res)
library(TSA)
McLeod.Li.test(y = res)

```

```

####garch(1,0) model####
library(fGarch)
library(timeSeries)
petro.garch1 <- garchFit(~garch(1,0), data = res, trace = F)
summary(petro.garch1)

```

```

####garch(1,1) model####
petro.garch21 <- garchFit(~garch(1,1), data = res, trace = F)
summary(petro.garch21)#####alpha+beta>1

```

```

#####igarch#####
library(rugarch)
library(parallel)
specinorm <- ugarchspec(variance.model = list(model = "iGARCH",
                                              garchOrder = c(1, 1), submodel = "GARCH"),

```

```
mean.model = list(armaOrder = c(2, 1)), distribution.model = "norm")
petro.igarch1 <- ugarchfit(data = petrodifff, spec = specinorm)
```

```
specistd <- ugarchspec(variance.model = list(model = "iGARCH",
      garchOrder = c(1, 1), submodel = "GARCH"),
      mean.model = list(armaOrder = c(2, 1)), distribution.model = "std")
petro.igarch2 <- ugarchfit(data = petrodifff, spec = specistd)
```

```
speciged <- ugarchspec(variance.model = list(model = "iGARCH",
      garchOrder = c(1, 1), submodel = "GARCH"),
      mean.model = list(armaOrder = c(2, 1)), distribution.model = "ged")
petro.igarch3 <- ugarchfit(data = petrodifff, spec = speciged)
```

```
#####egarch#####
```

```
specenorm <- ugarchspec(variance.model = list(model = "eGARCH",
      garchOrder = c(1, 1), submodel = "GARCH"),
      mean.model = list(armaOrder = c(2, 1)), distribution.model = "norm")
petro.egarch1 <- ugarchfit(data = petrodifff, spec = specenorm, solver = "nlminb")
```

```
specestd <- ugarchspec(variance.model = list(model = "eGARCH",
      garchOrder = c(1, 1), submodel="GARCH"),
      mean.model = list(armaOrder = c(2,1)), distribution.model = "std")
petro.egarch2 <- ugarchfit(data = petrodifff,spec = specestd,solver = "nlminb")
```

```
speceged <- ugarchspec(variance.model = list(model = "eGARCH",
      garchOrder = c(1, 1), submodel = "GARCH"),
      mean.model = list(armaOrder = c(2, 1)), distribution.model = "ged")
petro.egarch3 <- ugarchfit(data = petrodifff, spec = speceged, solver = "nlminb")
```

```
#####sgarch#####
```

```
specsnorm <- ugarchspec(variance.model = list(model = "sGARCH",
      garchOrder = c(1,1), submodel = "GARCH"),
      mean.model = list(armaOrder = c(2, 1)), distribution.model = "norm")
petro.sgarch1 <- ugarchfit(data = petrodifff, spec = specsnorm)
```

```
specsstd <- ugarchspec(variance.model = list(model = "sGARCH",
      garchOrder = c(1, 1), submodel = "GARCH"),
      mean.model = list(armaOrder = c(2, 1)), distribution.model = "std")
petro.sgarch2 <- ugarchfit(data = petrodifff, spec = specsstd)
```

```
specsged <- ugarchspec(variance.model = list(model = "sGARCH",
      garchOrder = c(1, 1), submodel = "GARCH"),
```

```
mean.model = list(armaOrder = c(2, 1)), distribution.model = "ged")
petro.sgarch3 <- ugarchfit(data = petrodifff, spec = specsGED)
```

```
#####gjrgarch#####
```

```
specgjrnorm <- ugarchspec(variance.model = list(model = "gjrGARCH",
      garchOrder = c(1, 1), submodel = "GARCH"),
      mean.model = list(armaOrder = c(2, 1)), distribution.model = "norm")
petro.gjrgarch1 <- ugarchfit(data = petrodifff, spec = specgjrnorm)
```

```
specgjrstdd <- ugarchspec(variance.model = list(model = "gjrGARCH", garchOrder = c(1,1),
      submodel = "GARCH"),
      mean.model = list(armaOrder = c(2, 1)), distribution.model = "std")
petro.gjrgarch2 <- ugarchfit(data = petrodifff, spec = specgjrstdd)
```

```
specgjrged <- ugarchspec(variance.model = list(model = "gjrGARCH",
      garchOrder = c(1, 1), submodel = "GARCH"),
      mean.model = list(armaOrder = c(2, 1)), distribution.model = "ged")
petro.gjrgarch3 <- ugarchfit(data = petrodifff, spec = specgjrged)
```

```
#####aparch#####
```

```
specapnorm <-
ugarchspec(variance.model=list(model="apARCH",garchOrder=c(1,1),submodel="GARCH"),
      mean.model=list(armaOrder=c(2,2)),distribution.model = "norm")
petro.apgarch1<-ugarchfit(data=res,spec=specapnorm)
```

```
specapstd<-
ugarchspec(variance.model=list(model="apARCH",garchOrder=c(1,1),submodel="GARCH"),
      mean.model=list(armaOrder=c(2,1)),distribution.model = "std")
petro.apgarch2<-ugarchfit(data=petrodifff,spec=specapstd)
```

```
specapged<-
ugarchspec(variance.model=list(model="apARCH",garchOrder=c(1,1),submodel="GARCH"),
      mean.model=list(armaOrder=c(2,1)),distribution.model = "ged")
petro.apgarch3<-ugarchfit(data=petrodifff,spec=specapged)
```

```
#####modeling of garch residual#####
```

```
egarchres2 <- ts(residuals(petro.egarch3,standard = T))[, 1]
```

```
plot(egarchres2)
```

```

jarque.bera.test(egarchres2)
qqnorm(egarchres2)
qqline(egarchres2)

des.egarchres2 <- density.default(x = egarchres2)
plot(des.egarchres2,col = 'blue')
x <- seq(min(egarchres2),max(egarchres2),0.01)
lines(x, dged(x, mean(egarchres2), sd(egarchres2), nu = 1.561563),
      lty = 1, col = "red")
legend("topleft", inset = .05, c("residuals", "ged"),
      lty = c(1, 1), col = c("blue", "red"))
mean(egarchres2);sd(egarchres2)

#####forecast#####
library(tseries)
library(urca)
library(fracdiff)
library(forecast)
library(rugarch)
library(FinTS)
library(fGarch)
library(xts)
#####HoltWinters#####
petroHW1=HoltWinters(petro,seasonal='additive')
plot(petroHW1)
legend("topleft", c("Observed", "Fitted"),
      lty = c(1, 1),col = c("black", "red"))
plot(forecast(petroHW1,h = 5))
HWresi1 <- forecast(petroHW1)$residuals
plot(HWresi1)
par(mfrow = c(2,1))
acf(HWresi1,lag = 100)
pacf(HWresi1,lag = 100)
Box.test(HWresi1,type = 'Ljung-Box', lag = 12)
m1 <- garchM(HWresi1)
names(m1)
resi1 <- m1$residuals
ts.plot(resi1)
ArchTest(resi1)
#####additive is better
petroHW2 <- HoltWinters(petro, seasonal = 'multiplicative')
plot(petroHW2)
legend("topleft", c("Observed", "Fitted"),
      lty = c(1, 1), col = c("black", "red"))

```

```

plot(forecast(petroHW2,h = 5))
HWresi2 <- forecast(petroHW2)$residuals
plot(HWresi2)
par(mfrow = c(2, 1))
acf(HWresi2,lag = 100)
pacf(HWresi2,lag = 100)
Box.test(HWresi2, type = 'Ljung-Box', lag=12)

```

#####ARIMA+EGARCH FORECAST#####

#####in-sample#####

```

data <- petro[1:322]
data1 <- ts(data, frequency=12)
realresi <- egarchres2[323:327]
resi <- ts(egarchres2)
g.pred <- ugarchforecast(petro.egarch3, n.ahead = 5)
petro.prev <- c(-0.3525, -0.0415, 0.4694, 0.5512, 0.5260)
petro.sigma <- c(10.119, 10.056, 9.995, 9.936, 9.878)
petro.pred <- ts(petro.prev, start = 323, end = 327)
petro.se <- ts(petro.sigma)
ts.plot(resi)
lines(petro.pred, col = 'red')
q <- qged(0.975)
sup <- petro.prev + q * petro.sigma
inf <- petro.prev - q * petro.sigma
sup.ts <- ts(sup, start = 323, end = 327)
inf.ts=ts(inf, start = 323, end = 327)
ts.plot(window(resi, start = 1, end = 327))
lines(sup.ts,col = 'blue')
lines(inf.ts,col = 'blue')
lines(petro.pred, col = 'red')

```

#####out-of-sample#####

```

resi <- ts(egarchres2)
g.pred <- ugarchforecast(petro.egarch3, n.ahead = 5)
petro.prev <- c(-0.3525, -0.0415, 0.4694, 0.5512, 0.5260)
petro.sigma <- c(10.119, 10.056, 9.995, 9.936, 9.878)
petro.pred <- ts(petro.prev, start = 328, end = 332)
petro.se <- ts(petro.sigma)
ts.plot(resi)
lines(petro.pred,col = 'red')
q <- qged(0.975)
sup <- petro.prev + q * petro.sigma
inf <- petro.prev - q * petro.sigma
sup.ts <- ts(sup, start = 328, end = 332)

```

```

inf.ts <- ts(inf, start = 328, end = 332)
ts.plot(window(resi, start = 1, end = 327))
lines(sup.ts, col='blue')
lines(inf.ts, col='blue')
lines(petro.pred, col='red')

```

```
#####dynamic#####
```

```
#####in-sample#####
```

```

petro1 <- ts(petro)
petro.ts <- ts(diff(petro))
petro.mod <- arima0(petro.ts, order = c(2,0,1), include.mean = F)
tsdiag(petro.mod)
q <- qged(0.975)

petro.prev.dyn <- ts(start = 323, end = 327)
petro.se.dyn <- ts(start = 323, end = 327)
for (i in 323:327){
  petro.mod.temp <- arima0(window(petro.ts, start = 1, end = i),
    order = c(2,0,1), include.mean = F)
  petro.prev.dyn[i-322] <- predict(petro.mod.temp, n.ahead = 1)$pred
  petro.se.dyn[i-322] <- predict(petro.mod.temp, n.ahead = 1)$se
}

```

```

par(mfrow = c(1,1))
plot(window(petro.ts, start = 200, end = 327))
lines(petro.prev.dyn, col = "red")
lines(petro.prev.dyn + q * petro.se.dyn, col = "blue")
lines(petro.prev.dyn - q * petro.se.dyn, col = "blue")
#####forecast#####

```

```

petro.prev.dyn
petro.ts[323:327]
f1 <- petro.prev.dyn
pred <- numeric()
pred[324] <- f1[1] + petro[323]
for (i in 325:328){
  pred[i] <- pred[i-1] + f1[i-324]
}
estimate <- pred[324:328]
real <- petro[324:328]
percent <- abs(estimate - real) / real
plot(petro1, type = 'l')

```

```

lines(ts(estimate, start = 324, end = 328), type = 'l', col = 'red')

#####out-of-sample#####
petro.prev.dyn <- ts(start = 327, end = 331)
petro.se.dyn <- ts(start = 327, end = 331)
petro.new <- petro.ts
q <- qged(0.975)

for (i in 327:331){
  petro.mod.temp <- arima0(window(petro.new, start = 1, end = i),
    order = c(2,0,1), include.mean = F)
  petro.prev.dyn[i-326] <- predict(petro.mod.temp, n.ahead = 1)$pred
  petro.se.dyn[i-326] <- predict(petro.mod.temp, n.ahead = 1)$se
  matrixa <- na.omit(as.matrix(petro.prev.dyn))
  matrixb <- as.matrix(petro.new)
  matrixc <- rbind(matrixb, matrixa)
  petro.new <- ts(matrixc)
  petro.prev.dyn <- ts(matrixa)
}
petro.prev.dyn <- ts(petro.prev.dyn, start = 327, end = 331)
petro.se.dyn <- ts(petro.se.dyn, start = 327, end = 331)

par(mfrow = c(1,1))
plot(window(petro.ts, start = 200, end = 327))
lines(petro.prev.dyn, col = "red")
lines(petro.prev.dyn + q * petro.se.dyn, col = "blue")
lines(petro.prev.dyn - q * petro.se.dyn, col = "blue")

#####dynamic forecast#####
petro.prev.dyn
petro.ts[323:327]
f2 <- petro.prev.dyn
pred <- numeric()
pred[328] <- f2[1] + petro[327]
for (i in 329:332){
  pred[i] <- pred[i-1] + f2[i-328]
}
estimate <- pred[328:332]

f3 <- petro.prev.dyn + q * petro.se.dyn
sup <- numeric()
sup[328] <- f3[1] + petro[327]
for (i in 329:332){

```



```

    sup[i] <- sup[i-1] + f3[i-328]
  }
  super <- sup[328:332]

f4 <- petro.prev.dyn - q * petro.se.dyn
inf <- numeric()
inf[328] <- f4[1] + petro[327]
for (i in 329:332){
  inf[i] <- inf[i-1] + f4[i-328]
}
infer <- inf[328:332]

plot(petro1, type = 'l')
lines(ts(estimate, start = 328, end = 332), type = 'l', col = 'red')
#lines(ts(super,start=328,end=332),type='l',col='blue')
#lines(ts(infer,start=328,end=332),type='l',col='blue')

###Neural network###
#####nn prediction#####
petro <- Index
data.petro <- petro[5:132]
data.lag <- cbind(petro[4:131], petro[3:130], petro[2:129], petro[1:128])
library(nnet)
bp.petro <- nnet(data.lag,data.petro, size = 15,
                 decay = 0.01, maxit = 1000, linout = TRUE, trace = FALSE)
bp.petro

sse <- sum((data.petro - predict(bp.petro,data.lag))^2)
sse
erro.aver <- sum((data.petro - predict(bp.petro, data.lag)) / data.petro) / 320
erro.aver
plot(data.petro, type = "l", lwd = 1)
lines(predict(bp.petro, data.lag), col = 2)

##### in-sample prediction error#####
data.predict.inner <- cbind(petro[323:327], petro[322:326],
                           petro[321:325], petro[320:324])
predict.inner <- predict(bp.petro, data.predict.inner)
erro.inner <- (petro[324:328] - predict.inner) / petro[324:328]
erro.inner

plot(data1$V1, type = "l", lwd = 2)
t <- seq(324, 328)
lines(t, predict.inner, col = 3, lwd = 2)

```

```
#####out-of-sample prediction#####
pred <- cbind(139.2929, 134.5844, 131.0425)
predict(bp.petro,newdata = pred)
result <- c(124.6808, 130.9538, 130.6553, 137.5247, 138.3707)
plot(as.vector(petro), type = "l", lwd = 2)
t2 <- seq(329,333)
lines(t2, result, col=2, lwd=2)
```

```
#####relaiton btw gold price and petro#####
data1 <- read.csv("petro.csv", header = FALSE)
petro <- ts(data1$V1[1:326], frequency = 12, start = c(1989,1))
petro1 <- na.omit(decompose(petro)$random)
data2 <- read.table("gold.txt")
gold <- ts(data2$V2, frequency = 12, start = c(1989,1))
par(mfrow = c(2,1))
ts.plot(petro, main = "imported petroleum pice index")
ts.plot(gold, main = "gold price")
```

```
###stationary test###
library(tseries)
lnpetro <- log(petro)
lngold <- log(gold)
adf.test(lnpetro.d)
pp.test(lnpetro.d)
kpss.test(lnpetro.d)
lnpetro.d <- diff(lnpetro)
lngold.d <- diff(lngold)
ts.plot(lngold.d)
```

```
###co-movement###
library(zoo)
library(forecast)
library(lmtest)
reg1 <- lm(lnpetro ~ lngold)
summary(reg1)
dwtest(reg1)
adf.test(reg1$residuals)
reg2 <- lm(lnpetro[3:326] ~ lngold[3:326] + lngold[2:325]
          + lnpetro[2:325] + lnpetro[1:324])
summary(reg2)
dwtest(reg2)
```

```
ts.plot(reg2$residuals)
adf.test(reg2$residuals)
pp.test(reg2$residuals)
kpss.test(reg2$residuals)
```

```
reg3 <- lm(lnpetro.d[4:326] ~ lngold.d[4:326] + lnpetro.d[3:325] + reg2$residuals[1:323]-1)
summary(reg3)
dwtest(reg3)
adf.test(reg3$residuals)
pp.test(reg3$residuals)
kpss.test(reg3$residuals)
###grangertest##
grangertest(lnpetro, lngold, order = 1)
grangertest(lngold, lnpetro, order = 1)
```