.

Ch 10 : GARCH model

Contents

1 Conditional Mean and Variance **2**
1.1 Conditional Mean and Variance 3
1.2 Conditional Mean of Financial Return 12
1.3 ARCH 15

2 GARCH **18**
2.1 GARCH 19

3 ARMA-GARCH **37**
3.1 ARMA - GARCH model 38
3.2 Structural Stability 45
3.3 Cheking Conditional Distribution 46

April 28, 2017

1

Conditional Mean and Variance

[ToC]

1.1 Conditional Mean and Variance

[ToC]

- Suppose Y_t is AR(1) series observation
- (unconditional) Mean

$$E(Y_t) = 0$$

- (unconditional) Variance

$$V(Y_t) = \gamma(0) = (1 + \phi_1^2)\sigma^2$$

Conditional Mean

- Conditonal Mean:

$$E\Big(Y_t \;\Big|\; \text{all variables realized by yesterday.}\Big)$$

- Conditonal mean of AR(1): $Y_t = \phi Y_{t-1} + e_t$

$$\begin{aligned} & E\Big(Y_t \;\Big|\; Y_{t-1}, Y_{t-2}, \ldots, e_{t-1}, e_{t-2}, \ldots\Big) \\ &\quad = E\Big(\phi Y_{t-1} + e_t \;\Big|\; Y_{t-1}, Y_{t-2}, \ldots, e_{t-1}, e_{t-2}, \ldots\Big) \\ &\quad = \phi Y_{t-1} + E(e_t) \;=\; \phi Y_{t-1}. \end{aligned}$$

- Conditonal variance

$$
\begin{aligned}
V\Big(Y_t \ \Big|\ Y_{t-1}, Y_{t-2}, \ldots, e_{t-1}, e_{t-2}, \ldots\Big) & \\
&= V\Big(\phi Y_{t-1} + e_t \ \Big|\ Y_{t-1}, Y_{t-2}, \ldots, e_{t-1}, e_{t-2}, \ldots\Big) \\
&= V(e_t) \ = \ \sigma^2
\end{aligned}
$$

```
n=200
Y <- arima.sim(n = n, list(ar = c(0.8) ))

v.AR1 = (1+.8^2)*1

plot(Y, type="o", main="AR(1)", xlim=c(0,n*1.1))
abline(h=0)
abline(h=c(1,-1)*1.96*v.AR1, col="red")

lines(n+1, .8*Y[n], type="p", col="blue")
lines(c(n+1,n+1), .8*Y[n]+c(1.96,-1.96), type="p", col="red")
```

- Suppose Y_t is MA(1) series observation
- (unconditional) Mean

$$E(Y_t) = 0$$

- (unconditional) Variance

$$V(Y_t) = \gamma(0) = (1 + \theta_1^2)\sigma^2$$

Conditonal mean of MA(1): $Y_t = e_t + \theta_1 e_{t-1}$

$$E\Big(Y_t \;\Big|\; Y_{t-1}, Y_{t-2}, \ldots, e_{t-1}, e_{t-2}, \ldots\Big)$$

$$= \quad E\Big(e_t + \theta_1 e_{t-1} \;\Big|\; Y_{t-1}, Y_{t-2}, \ldots, e_{t-1}, e_{t-2}, \ldots\Big)$$

$$= \quad E(e_t) + \theta_1 e_{t-1} \quad = \quad \theta_1 e_{t-1}$$

Note that e_{t-1} is not observable.

Conditonal variance of MA(1): $Y_t = e_t + \theta_1 e_{t-1}$

$$\begin{aligned} Var\Big(Y_t \;\Big|\; Y_{t-1}, Y_{t-2}, \ldots, e_{t-1}, e_{t-2}, \ldots\Big) \\ &= Var\Big(e_t + \theta_1 e_{t-1} \;\Big|\; Y_{t-1}, Y_{t-2}, \ldots, e_{t-1}, e_{t-2}, \ldots\Big) \\ &= Var(e_t) \;=\; \sigma^2. \end{aligned}$$

AR(1)

$$
\begin{array}{lcl}
& \text{Uncond'l} & \text{cond'l} \\
E(Y_t) = & 0, & E(Y_t|\omega_{t-1}) = \phi_1 Y_{t-1}, \\
Var(Y_t) = & (1+\phi_1^2)\sigma^2 & Var(Y_t|\omega_{t-1}) = \sigma^2
\end{array}
$$

MA(1)

$$
\begin{array}{lcl}
& \text{Uncond'l} & \text{cond'l} \\
E(Y_t) = & 0, & E(Y_t|\omega_{t-1}) = \theta_1 e_{t-1}, \\
Var(Y_t) = & (1+\theta_1^2)\sigma^2 & Var(Y_t|\omega_{t-1}) = \sigma^2
\end{array}
$$

For ARMA(p,q) model, conditional mean changes, but conditional variance is constant.

```
x  <- arima.sim(n = 50, list(ar = c(0.8) ))
plot(x, type="o", main="AR(1)"); abline(h=0)

x  <- arima.sim(n = 50, list(ma = c(0.5) ))
plot(x, type="o", main="MA(1)"); abline(h=0)
```

1.2 Conditional Mean of Financial Return

[ToC]

$$
\begin{aligned}
X_t &= \text{Stock Price (observation)} \\
Y_t &= \ln(X_t) - \ln(X_{t-1}) \qquad : \text{ log return}
\end{aligned}
$$

```
library(quantmod)
source("http://gozips.uakron.edu/~nmimoto/477/TS_R-90.txt")

getSymbols("AAPL") #- download from Yahoo!
X <- as.ts(AAPL$AAPL.Adjusted[200:600])

plot(X, type="o")
plot(diff(X), type="o")

Randomness.tests(diff(log(X)))
```

Stylized Facts about Financial Return

- Uncorrelated
- Squares are correlated
- Clustering
- Asymmetry
- Heavy Tailed unconditional and conditional distribution

Heteroschedasticity

Dont confuse the conditional heteroscedasticity with (unconditonal) heteroscedasticity:

```
D <- read.csv("http://gozips.uakron.edu/~nmimoto/pages/datasets/Gas.csv")

D1 <- ts(D[,2], start=c(1956, 1), freq=12)

plot(D1, type='o')
```

1.3 ARCH

[ToC]

- Engle (1985) AutoRegressive Conditionally Heteroscedastic Model

$$
\begin{aligned}
Y_t &= \sigma_t e_t \qquad e_t \sim_{iid} \mathcal{N}(0,1) \\
\sigma_t^2 &= \omega + \alpha Y_{t-1}^2
\end{aligned}
$$

- Mean

$$E(Y_t) = \sigma_t E(e_t) = 0$$

- Variance

$$V(Y_t) = V(\sigma_t)V(e_t) =$$

Conditonal Mean and Variance of ARCH

- Conditional mean

$$E[Y_t \Big| Y_{t-1}, e_{t-1}, \ldots] \quad = \quad \sigma_t E[e_t] = 0$$

- Conditional variance

$$V[Y_t \Big| Y_{t-1}, e_{t-1}, \ldots] \quad = \quad \sigma_t^2 V[e_t] = \sigma_t^2$$

ARCH is uncorrelated

- Y_t is uncorrelated, so it will pass the Ljung-Box test.
- But Y_t^2 is correlated, so it will NOT pass the McLeod-Li test.

2

GARCH

[ToC]

2.1 GARCH

[ToC]

GARCH(1,1) model

$$
\begin{aligned}
Y_t &= \sigma_t e_t \qquad e_t \sim_{iid} \mathcal{N}(0,1) \\
\sigma_t^2 &= \omega + \alpha Y_{t-1}^2 + \beta \sigma_{t-1}^2
\end{aligned}
$$

Conditional Mean and Variance of GARCH

Conditional Mean: 0

Conditional Variance : σ_t^2

Example: Daily SPY

Daily Price of SP500 ETF (SPY) from Jan 02 2000 to Dec 31 2014

```
D <- read.csv("http://gozips.uakron.edu/~nmimoto/pages/datasets/SPY.csv", header=T)
source("http://gozips.uakron.edu/~nmimoto/477/TS_R-90.txt")

head(D)
X <- ts(D[,7], start=1)
plot(X)

X1 <- diff(log(X))
plot(X1)

acf(X1)
acf(X1^2)
```

```
library(fGarch)
source("http://gozips.uakron.edu/~nmimoto/477/TS_R-90.txt") #- load Randomness.tests

Fit1 <-  garchFit(~ garch(1,1), data=X1, cond.dist="norm",  include.mean = FALSE, trace = FALSE)

Fit1@fit$par           #-- estimated parameters
Fit1@residuals         #-- this is not the garch residuals!!!! same as X1
Fit1@sigma.t           #-- estimated sig_t
Fit1@residuals/Fit1@sigma.t       #-- this is the (standardized) GARCH residuals


print(Fit1@fit$ics)   #-- AIC and BIC are here

layout(matrix(1:2, 2, 1, byrow=T))
plot(X1)
plot(Fit1@sigma.t, type="l")   #- estimated Sigma_t


layout(1)
plot(X1)
lines(1.96*Fit1@sigma.t, type="l", col="red")

res1  <- X1/Fit1@sigma.t   #-- this is the residuals
Randomness.tests(res1)
```

Representing σ_t^2 using Y_t^2

$$
\begin{aligned}
\sigma_t^2 &= \omega + \alpha Y_{t-1}^2 + \beta \sigma_{t-1}^2 \\
&= \omega + \alpha Y_{t-1}^2 + \beta \Big(\omega + \alpha Y_{t-2}^2 + \beta \sigma_{t-2}^2 \Big) \\
&= \omega + \beta\omega + \alpha Y_{t-1}^2 + \beta\alpha Y_{t-2}^2 + \beta^2 \sigma_{t-2}^2 \\
&= \omega + \beta\omega + \alpha Y_{t-1}^2 + \beta\alpha Y_{t-2}^2 + \beta^2 \Big(\omega + \alpha Y_{t-3}^2 + \beta \sigma_{t-3}^2 \Big) \\
&= \omega + \beta\omega + \beta^2\omega + \alpha Y_{t-1}^2 + \beta\alpha Y_{t-2}^2 + \beta^2\alpha Y_{t-3}^2 + \beta^3 \sigma_{t-3}^2
\end{aligned}
$$

continuing, we get

$$
\begin{aligned}
&= \omega(1+\beta+\beta^2+\cdots) + \alpha \sum_{i=0}^{k} \beta^i Y_{t-1-i}^2 + \beta^{k+1}\sigma_{t-1-k}^2 \\
&= \frac{\omega}{1-\beta} + \alpha \sum_{i=0}^{\infty} \beta^i Y_{t-1-i}^2
\end{aligned}
$$

$$
\sigma_t^2 = \frac{\omega}{1-\beta} + \alpha \sum_{i=0}^{\infty} \beta^i Y_{t-1-i}^2
$$

We will use the truncated and estimated version of this,

$$
\hat{\sigma}_t^2 = \frac{\hat{\omega}}{1-\hat{\beta}} + \hat{\alpha} \sum_{i=0}^{t-1} \hat{\beta}^i Y_{t-1-i}^2
$$

Residuals of GARCH process

$$Y_t = \sigma_t e_t$$

Using observation $\{Y_1, \ldots, Y_n\}$, the residuals are

$$\hat{e}_t = Y_t / \hat{\sigma}_t$$

Quasi-Maximum Likelihood Estimator

Importatnce of Conditional Distribution

$$
\begin{aligned}
Y_t &= \sigma_t e_t \\
\sigma_t^2 &= \omega + \alpha Y_{t-1}^2 + \beta \sigma_{t-1}^2
\end{aligned}
$$

Conditonal distribution = Distribution of e_t

Guessing correct distribution for e_t is very important in GARCH parameter estimation.

Built-in distributions

```
x <- seq(-10,10, .1)

#- Standardized t-distribution
plot(x, dstd(x, mean = 0, sd = 1, nu = 5), type="l" )


#- Generalized Error Distribution
plot(x, dged(x, mean = 0, sd = 1, nu = 2), type="l" )

#- Skewed-Standardized t-distribution
plot( x, dsstd(x, mean = 0, sd = 1, nu = 5, xi = 1.5), type="l" )
lines(x, dstd(x, mean = 0, sd = 1, nu = 5), col="red" )

#- Skewed-Generalized Error Distribution
plot(x, dsged(x, mean = 0, sd = 1, nu = 5, xi = 1.5) , type="l" )
lines(x, dged(x, mean = 0, sd = 1, nu = 2), type="l" )
```

Example: Simulation Study

True parameter (.024, .1, .8)

1. True cond'l dist: Normal Estimated using: Normal
2. True cond'l dist: std(5) Estimated using: Normal
3. True cond'l dist: sged(skew=.7, shape=1.45) Estimated using: Normal

```
MSE1/MSE1   #- Relative MSE
[1] 1 1 1

MSE2/MSE1
[1] 1.584734 3.396001 1.831015

MSE3/MSE1
[1] 1.198293 1.613116 1.355613
```

Stationarity Condition for GARCH(1,1)

GARCH(1,1) is weakly stationary if

$$E(\log(\beta + \alpha e_t^2)) < 0$$

This is satisfied if

$$\alpha + \beta < 1.$$

```
#--- See how closely true sigma is estimated and predicted (Using simulated GARCH)---

library(quantmod)
library(fGarch)
source("http://gozips.uakron.edu/~nmimoto/477/TS_R-90.txt")

    theta = c(.5, .7, .2)
    spec1 <- garchSpec(model = list(omega=theta[1], alpha=theta[2], beta=theta[3]),  cond.dist="norm")

    #- Generate GARCH
    X      <- garchSim(spec1, n = 1000, extended=T)   #extended=T:  now  Y has 3 columns: garch sigma eps
    X      <- as.xts(X)
    X      <- xts(X, order.by=as.Date(index(X)))

    Y         <- X[,1]    #- simulated GARCh
    sig.true <- X[,2]    #- this is true sigma


    #- Estimate GARCH
    out1  <-  garchFit(~ garch(1,1), data=Y, cond.dist="norm", include.mean = FALSE, trace = FALSE)
    sig.estim <- xts(out1@sigma.t, order.by=index(Y))
```

```
#- compare true sigma vs estimated sigma
cbind(sig.true, sig.estim)

plot(sig.true, main="sig.true vs sig.estim")
lines(sig.estim, col="red")


#- Compute the sigma for tomorrow
w <- out1@fit$par[1]
a <- out1@fit$par[2]
b <- out1@fit$par[3]
sig.pred <- xts( sqrt( w + a*Y[1000]^2 + b*out1@sigma.t[1000]^2 ), order.by=index(Y[1000])+1)

#- This does the same thing
sig.pred2 <- predict(out1)
sig.pred2
sig.pred2[1,3]
```

```
#- Rolling 1-day prediction of sig.t
sig.pred1 <- numeric(0)
date.stp <- numeric(0)
for (i in 1:750){
    T <- Y[(1:250)+(i-1)]

    n <- length(T)
    out1  <-  garchFit(~ garch(1,1), data=T, cond.dist="norm", include.mean = FALSE, trace = FALSE)
    sig.pred1[i] <- predict(out1)[1,3]   #- sig.t prediction
    date.stp[i]  <- index(T[n])+1
}

sig.pred <- xts(sig.pred1, order.by=as.Date(date.stp))


#- Plot and compare estimated vs rolling predicted
plot(sig.true, main="sigma.t:  True vs Estim vs Pred")
lines(sig.estim, col="red")
lines(sig.pred, col="blue")
```

```
#--- See how close estimated sigma and rolling predicted sigma (Using real data)---

    getSymbols("^GSPC")     #- download S\&P500 index from Yahoo
    X <- Ad(GSPC)

    plot(X)

    layout(matrix(1:2,2,1))
    plot(X)
    plot(diff(X))

    auro.arima(X)



    Y <- diff(log(X["2013::2016"]))[-1]    #- we model this with GARCH (this time we don't know the true sigma.t)
    length(Y)

    #- Estimate GARCH
    out1  <-  garchFit(~ garch(1,1), data=Y, cond.dist="norm", include.mean = FALSE, trace = FALSE)
    sig.estim <- xts(out1@sigma.t, order.by=index(Y))
```

```
#- Rolling 1-day prediction of sig.t
sig.pred1 <- numeric(0)
date.stp  <- numeric(0)

for (i in 1:750){
    T <- Y[(1:250)+(i-1)]
    n <- length(T)
    out1  <-  garchFit(~ garch(1,1), data=T, cond.dist="norm", include.mean = FALSE, trace = FALSE)
    sig.pred1[i] <- predict(out1)[1,3]   #- sig.t prediction
    date.stp[i]  <- index(Y(1:250+i)
}

sig.pred <- xts(sig.pred1, order.by=as.Date(date.stp))

#- Compare estimated sigma vs predicted sigma -
plot(sig.estim)
lines(sig.pred, col="blue")

#- Compare daily CI from two sigmas
plot(Y["2016"], type="h")
lines( 1.65*sig.estim, col="red")
lines(-1.65*sig.estim, col="red")
lines( 1.65*sig.pred, col="blue")
lines(-1.65*sig.pred, col="blue")
```

3

ARMA-GARCH

[ToC]

3.1 ARMA - GARCH model

[ToC]

Y_t is ARMA with error ϵ_t. ϵ_t is GARCH.

$$
\begin{aligned}
\Phi(B)Y_t &= \Theta(B)\epsilon_t \\
\epsilon_t &= \sigma_t e_t \qquad e_t \sim_{iid} \mathcal{N}(0,1) \\
\sigma_t^2 &= \omega + \alpha\epsilon_{t-1}^2 + \beta\sigma_{t-1}^2
\end{aligned}
$$

```
#--- ARMA-GARCH estimation example ---

    D <- read.csv("http://gozips.uakron.edu/~nmimoto/pages/datasets/SPY.csv", header=T)
    source("http://gozips.uakron.edu/~nmimoto/477/TS_R-90.txt")
    library(xts)

    head(D)
    X <- xts(D[,7], order.by=as.Date(D[,"Date"], format="%m/%d/%Y"))    #- X is a xts object
    plot(X)

    X1 <- diff(log(X))[-1]
    plot(X1)

    acf(X1)
    acf(X1^2)

    Randomness.tests(X1)
```

```
library(fGarch)

#- Estimae parameters of ARMA and GARCH at the same time
  Fit1 <-  garchFit(~ arma(1,1) + garch(1,1), data=X1, cond.dist="norm",  include.mean = FALSE, trace = FALSE)
  Fit1

  Fit1@fit$par           #-- estimated parameters
  Fit1@residuals         #-- this is ARCH residuals. (before GARCH)
  print(Fit1@fit$ics)    #-- AIC and BIC are here
  Randomness.tests(res1)
  plot(Fit1)             #-- choose option 13 for residual qq plot

  sig.t <- xts(Fit1@sigma.t,                order.by=index(X1))  #-- estimated sig_t fo GARCH
  res1  <- xts(Fit1@residuals/Fit1@sigma.t, order.by=index(X1))   #-- this is the (standardized) ARMA-GARCH residuals

#- plot log-return and sigma_t -
  layout(matrix(1:2, 2, 1, byrow=T))
  plot(X1, main="SPY daily log-return")
  plot(sig.t, type="l", main="estimated sigma_t")   #- estimated Sigma_t

#- plot log-return overlay with sigma_t -
  layout(1)
  plot(X1)
  lines( 1.96*sig.t, type="l", col="red")
  lines(-1.96*sig.t, type="l", col="red")
```

```
#--- h-step ahead prediction from ARMA-GARCH ---

  h = 15    #- predict 15 days ahead

  X.pred  <- xts(predict(Fit1, n.ahead=h)[,1], order.by=index(X1)[length(X1)]+(1:h))
  SD.pred <- xts(predict(Fit1, n.ahead=h)[,3], order.by=index(X1)[length(X1)]+(1:h))

#- plot log-return overlay with sigma_t -

  plot(rbind(X1["2014"], X.pred))
  lines(X.pred, col="red")
  lines( 1.96*SD.pred, type="l", col="red")
  lines(-1.96*SD.pred, type="l", col="red")
```

```
#--- Rolling 1-step prediction from ARMA-GARCH
Y <- X1

Y.pred1 <- sig.pred1 <- date.stp <- numeric(0)
for (i in 1:1000){
    T <- Y[(1:250)+(i-1)]
    Fit1 <-  garchFit(~ arma(1,1) + garch(1,1), data=T, cond.dist="norm", include.mean = FALSE, trace = FALSE)
    Y.pred1[i]   <- predict(Fit1, n.ahead=1)[,1]
    sig.pred1[i] <- predict(Fit1, n.ahead=1)[,3]
    date.stp[i]  <- index(Y[250+i])
}
Y.pred   <- xts(Y.pred1,   order.by=as.Date(date.stp))
sig.pred <- xts(sig.pred1, order.by=as.Date(date.stp))


#- Compare predicted vs actual -
plot(Y["2000::2004"]);   lines(Y.pred, col="red");   lines( 1.96*sig.pred, col="red");   lines(-1.96*sig.pred, col="red")


#- Actual vs Predicted
head(  cbind(Y[index(Y.pred)], Y.pred)  )

plot(Y[index(Y.pred)] - Y.pred)

mean( sign(Y[index(Y.pred)])==sign(Y.pred) )  #- Can Y.pred guess up/down ?
mean( sign(Y[index(Y.pred)])==1 )             #- num of days SPY went up
```

3.2 Structural Stability

[ToC]

- When a long time series is fitted, you must ask if entire series can be fitted with a single model with same parameter.
- Sometime breaking up a series into couple of parts will allow you to fit better.

3.3 Cheking Conditional Distribution

[ToC]

Kolmogolov-Smirnov Test

In iid setting, K-S test can be used to test hypothesis $X_i \sim F$, and

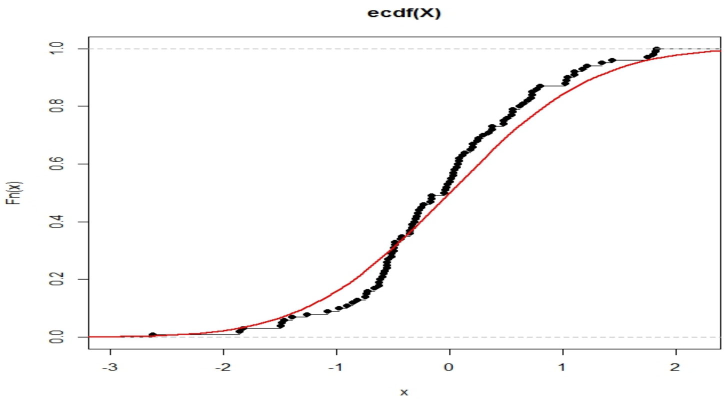
$$H_0 : F = F_0 \quad vs \quad H_A : F \neq F_0 \qquad \text{(F_0 is completely specified)}$$

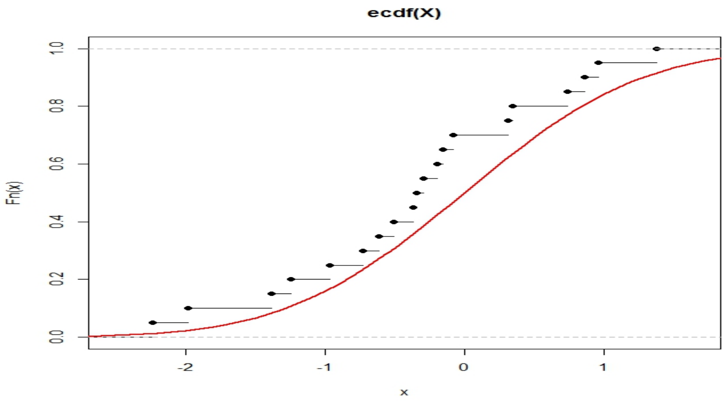
K-S test uses Empirical Distibution (EDF, ECDF),

$$\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^{n} I(x_i < x)$$

and calculate the test statistic

$$KS = \sup_x |\hat{F}_n(x) - F(x)|$$





Where is $(\frac{i}{n}, X_{(i)})$ and $(\frac{i-1}{n}, X_{(i)})$?

$$
\begin{aligned}
D_n &= \sup_x |\hat{F}_n(x) - F_0(x)| \\
&= \max_i \left(\left| \frac{i}{n} - F_0(x_{(i)}) \right|, \quad \left| \frac{i-1}{n} - F_0(x_{(i)}) \right| \right)
\end{aligned}
$$

If X_i are indeed from F_0,

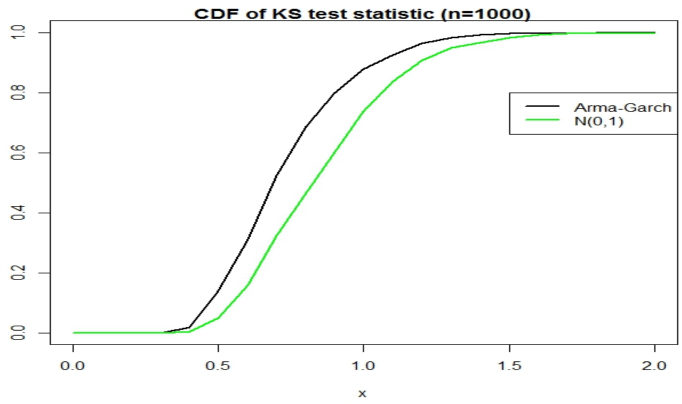
$$D_n \quad =_d \quad \max_i \left(\left| \frac{i}{n} - U_{(i)} \right|, \quad \left| \frac{i-1}{n} - U_{(i)} \right| \right)$$

with $U_i \sim_{iid} Unif(0, 1)$.

- No matter what F is, distribution of the test statistic KS under the null is known.
- KS test with completely specified F_0 is a distribution-free test.
- When F_0 contains nuisance parameter, Distribution of D_n under the null depends on F. KS test becomes only asymptotically distribution free test. For finite n, null distribution must be computed by Monte Carlo simulation.

K-S test in Time Series Setting

- In time sries, we want to use K-S test for checking distribution of innovations. (errors).
- Since we assume e_t are i.i.d. noise, if we can observe e_t, K-S test can be directly applicable.
- However, in Time Series analysis e_t are not observable, and only residuals $\hat{e}_t$ are available.
- Can we still use K-S test on $\hat{e}_t$?



CDF of KS test statistic (n=1000)
Arma-Garch
N(0,1)
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5
2.0
x