Time Series Analysis

Modeling Heteroskedasticity

Nicoleta Serban, Ph.D.

Professor

Stewart School of Industrial and Systems Engineering

Basic Concepts: Data Examples

Georgia Tech

About This Lesson

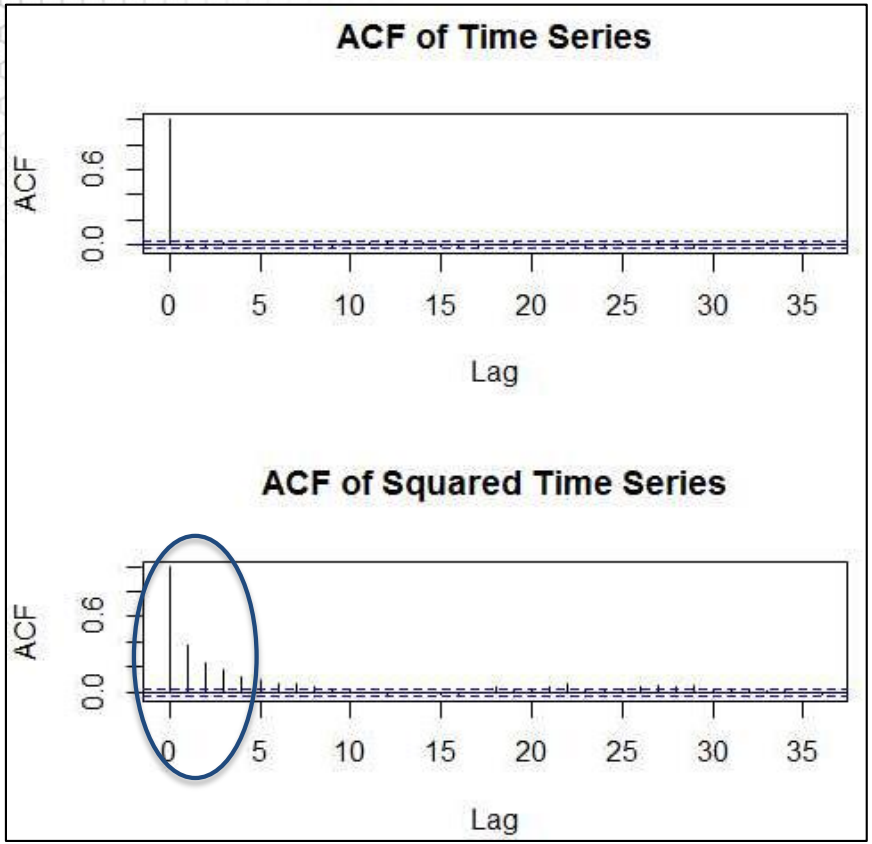
Simulation: Time-varying Conditional Variance

```
## Generate time series with heteroskedacticity
a0 = 0.2
a1 = 0.5
b1 = 0.3
w = rnorm(5000)
eps = rep(0, 5000)
sigsq = rep(0, 5000)
for (i in 2:5000) {
   sigsq[i] = a0 + a1 * (eps[i-1]^2) + b1 * sigsq[i-1]
   eps[i] = w[i]*sqrt(sigsq[i])
 }
## Plot the acf of the time series and squared time series
acf(eps)
acf(eps^2)
```

$$\varepsilon_t = \sigma_t \, w_t$$
$$\sigma_t{}^2 = 0.2 + 0.5\varepsilon_{t-1}{}^2 + 0.3\sigma_{t-1}{}^2$$

Georgia Tech

Simulation: Time-varying Conditional Variance



$$\varepsilon_t = \sigma_t \, w_t$$
$$\sigma_t{}^2 = 0.2 + 0.5\varepsilon_{t-1}{}^2 + 0.3\sigma_{t-1}{}^2$$

PDC Energy, Inc (PDCE)

Summary:

- Crude oil and natural gas producer with headquartered in Denver, Colorado
- PDC's portfolio is comprised of the Wattenberg Field in Colorado, the Delaware Basin in West Texas and the Utica Shale in Ohio

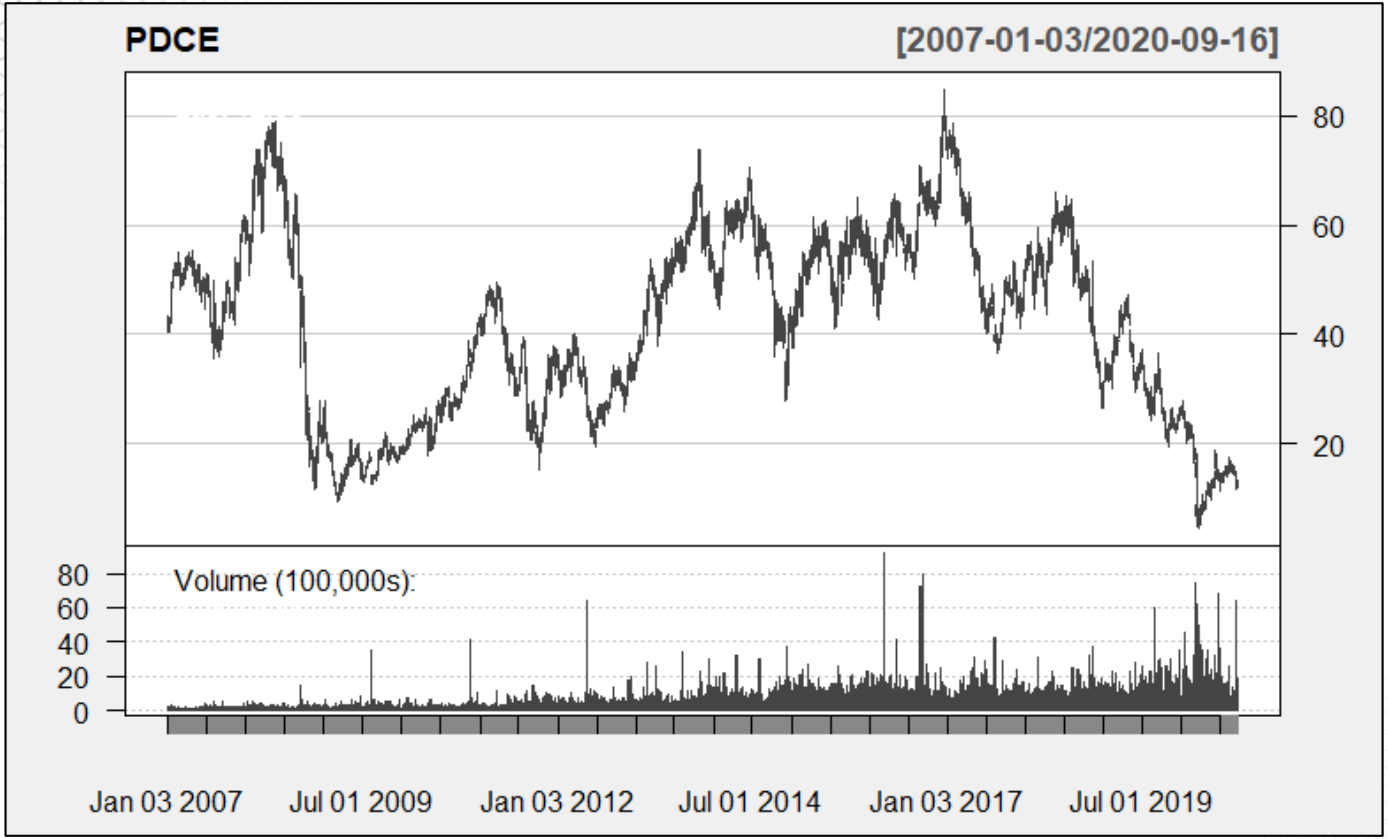
Time Series Data:

- Daily stock price for more than 12 years of data starting with January 2007
- Largely dependent on the crude oil price

Financial Data Analysis

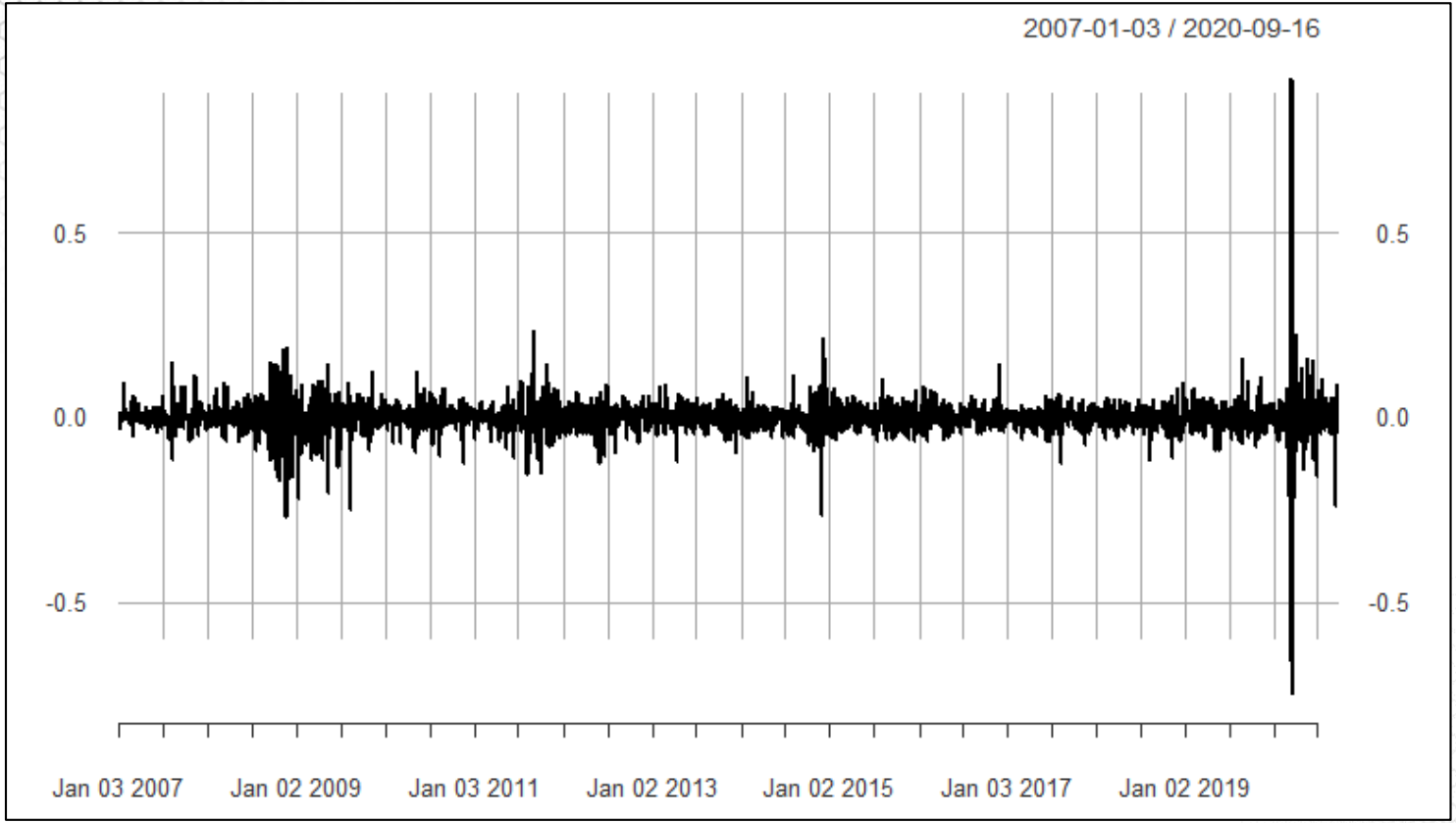
```
## Financial Data Analysis
library(quantmod)
## Get the daily trading data for PDCE
getSymbols("PDCE",src="yahoo")
## Time Series Plot
candleChart(PDCE, theme="white")
## Log returns of the close price
pdcert = diff(log(Cl(PDCE)))
plot(pdcert,main="")
```

Time Series Plot: Price & Volume



Georgia Tech

Difference Time Series Plot



ARIMA Fit & Model Selection

```
## R function for ARIMA(p,d,q) fit
test_modelA <- function(p,d,q){
   mod <- arima(pdcert, order=c(p,d,q), method="ML")
   current.aic <- AIC(mod)
   current.aic<-current.aic-2*(p+q+1)+2*(p+q+1)*n/(n-p-q-2)
   df <- data.frame(p,d,q,current.aic)
   names(df) <- c("p","d","q","AIC")
   print(paste(p,d,q,current.aic,sep=" "))
   return(df)}
## Model Selection
orders <- data.frame(Inf,Inf,Inf,Inf)
names(orders) <- c("p","d","q","AIC")
… for-loop Apply test_modelA for all (p,d,q) combination
orders <- orders[order(-orders$AIC),]
```

Georgia Tech

ARIMA Fit & Model Selection

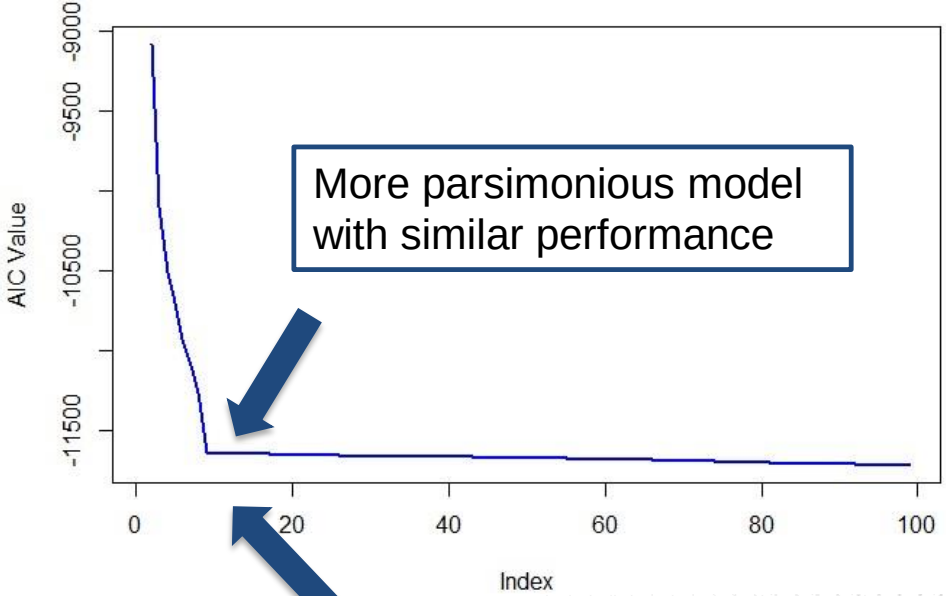
```
## R function for ARIMA(p,d,q) fit
test_modelA <- function(p,d,q){
   mod <- arima(pdcert, order=c(p,d,q), method="ML")
   current.aic <- AIC(mod)
   current.aic<-current.aic-2*(p+q+1)+2*(p+q+1)*n/(n-p-q-2)
   df <- data.frame(p,d,q,current.aic)
   names(df) <- c("p","d","q","AIC")
   print(paste(p,d,q,current.aic,sep=" "))
   return(df)}
## Model Selection
orders <- data.frame(Inf,Inf,Inf,Inf)
names(orders) <- c("p","d","q","AIC")
… for-loop Apply test_modelA for all (p,d,q) combination
orders <- orders[order(-orders$AIC),]
```

Output

ARIMA Fit & Model Selection (cont'd)

```
> tail(orders)
    p  d  q        AIC
90  6  0  4 -11718.17
64  4  0  6 -11718.19
63  4  0  5 -11720.65
76  5  0  4 -11720.76
92  6  0  6 -11722.42
62  4  0  4 -11722.51
```

Selected Order: p=4, d=0, q=4



More parsimonious model with similar performance

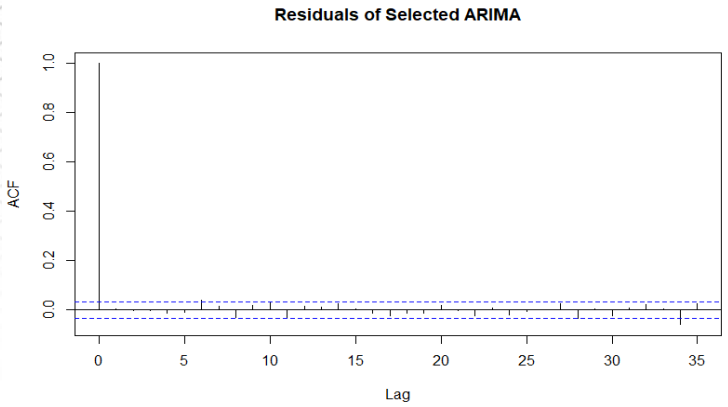
Selected Order: p=0, d=1, q=1

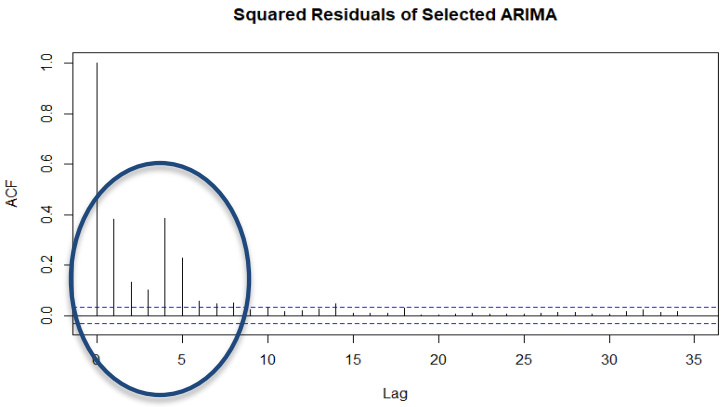
ARIMA Fit: Residual Analysis

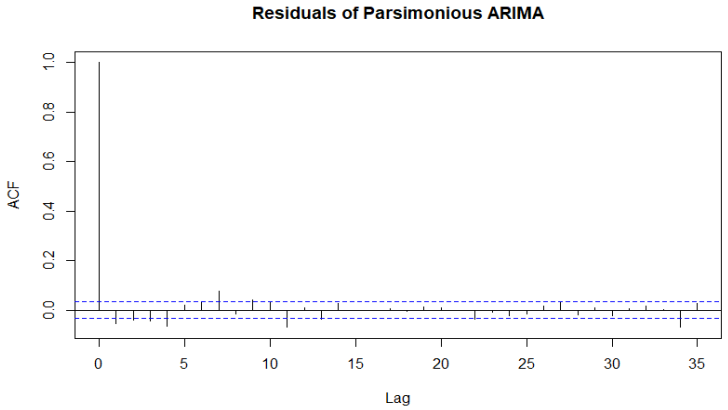
```
## ARIMA(p,d,q) fit for selected and parsimonious models
select.arima <- arima(pdcert, order=c(4,0,4))
pars.arima <- arima(pdcert, order=c(0,1,1))

## Residual Analysis
select.resids <- resid(select.arima)[-1]
pars.resids <- resid(pars.arima)[-1]
acf(select.resids,main="Residuals of Selected ARIMA")
acf(select.resids^2,main="Squared Residuals of Selected ARIMA")
acf(pars.resids,main="Residuals of Parsimonious ARIMA")
acf(pars.resids^2,main="Squared Residuals of Parsimonious ARIMA")
```

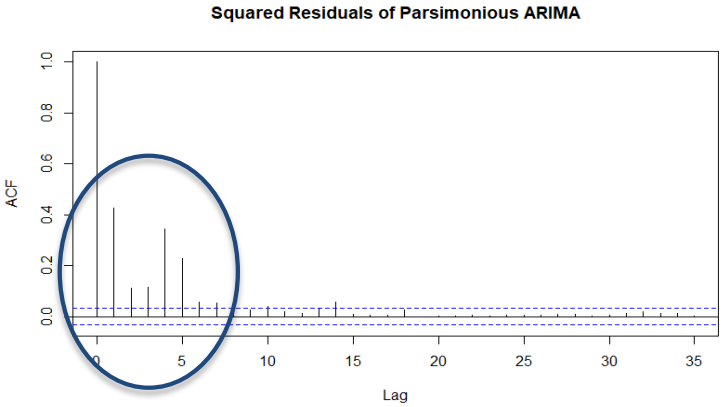
Georgia Tech

ARIMA Fit: Residual Analysis









ARIMA Fit: Residual Analysis (cont'd)

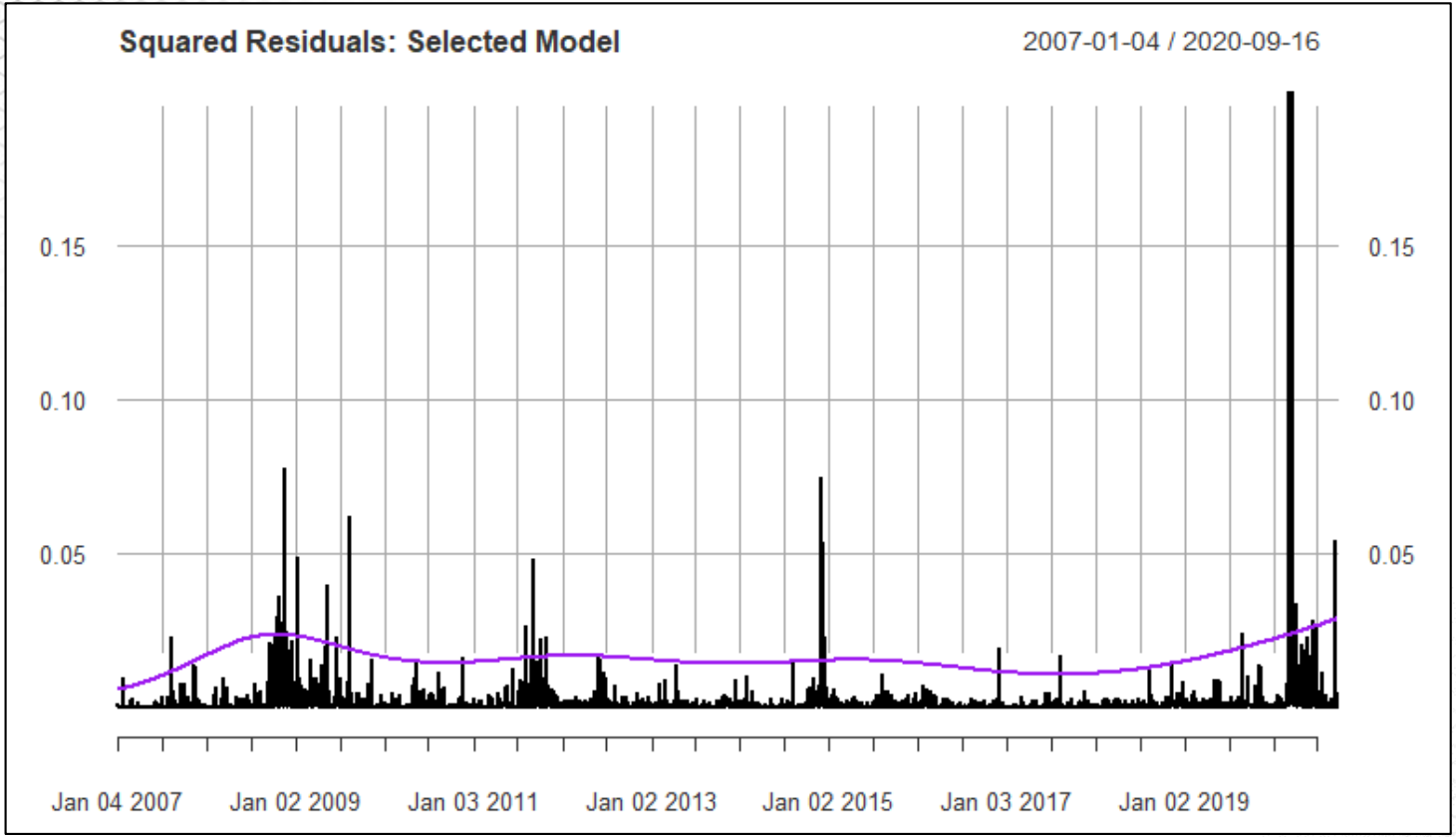
```
> Box.test(select.resids,lag=9,type='Ljung',fitdf=8)
        Box-Ljung test
data:  select.resids
X-squared = 11.729, df = 1, p-value = 0.0006155

> Box.test(pars.resids,lag=3,type='Ljung',fitdf=2)
        Box-Ljung test
data:  pars.resids
X-squared = 22.492, df = 1, p-value = 2.11e-06

> Box.test((select.resids)^2,lag=9,type='Ljung',fitdf=8)
        Box-Ljung test
data:  (select.resids)^2
X-squared = 1317.6, df = 1, p-value < 2.2e-16

> Box.test((pars.resids)^2,lag=3,type='Ljung',fitdf=2)
        Box-Ljung test
data:  (pars.resids)^2
X-squared = 721.68, df = 1, p-value < 2.2e-16
```

Georgia Tech

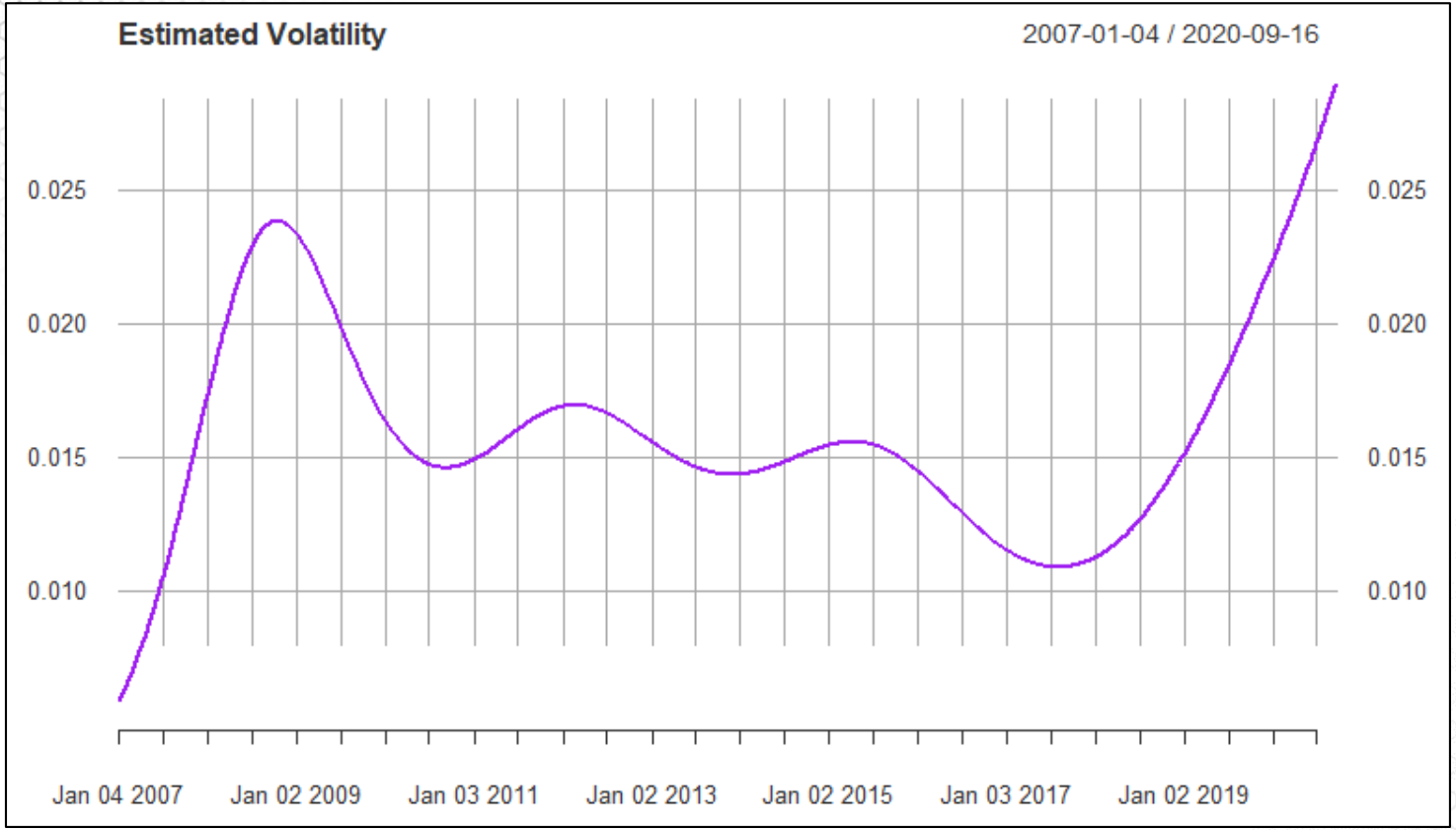
Nonparametric Fit: Variance

```
## Estimate the variance using nonparametric regression
zt.ssq.log = log(select.resids^2)
n = length(select.resids)
time.pts = c(1:n)
time.pts = (time.pts-min(time.pts))/(max(time.pts)-min(time.pts))
gam.var.select = gam(zt.ssq.log~s(time.pts))
pdcert.var.select=sqrt(exp(fitted(gam.var.select)))
## Plot squared residuals vs estimated variance
mydates = row.names(as.data.frame(PDCE))
mydates=as.Date(mydates,"%Y-%m-%d")
tsresid.select=xts(select.resids,mydates[-1])
tspdcert.var.select = xts(pdcert.var.select,mydates[-1])
plot(tsresid.select^2,main='Squared Residuals: Selected Model',ylim=c(0,0.2))
lines(tspdcert.var.select,lwd=2,col="purple")
```

Nonparametric Fit: Variance (cont'd)



Nonparametric Fit: Variance (cont'd)



Summary