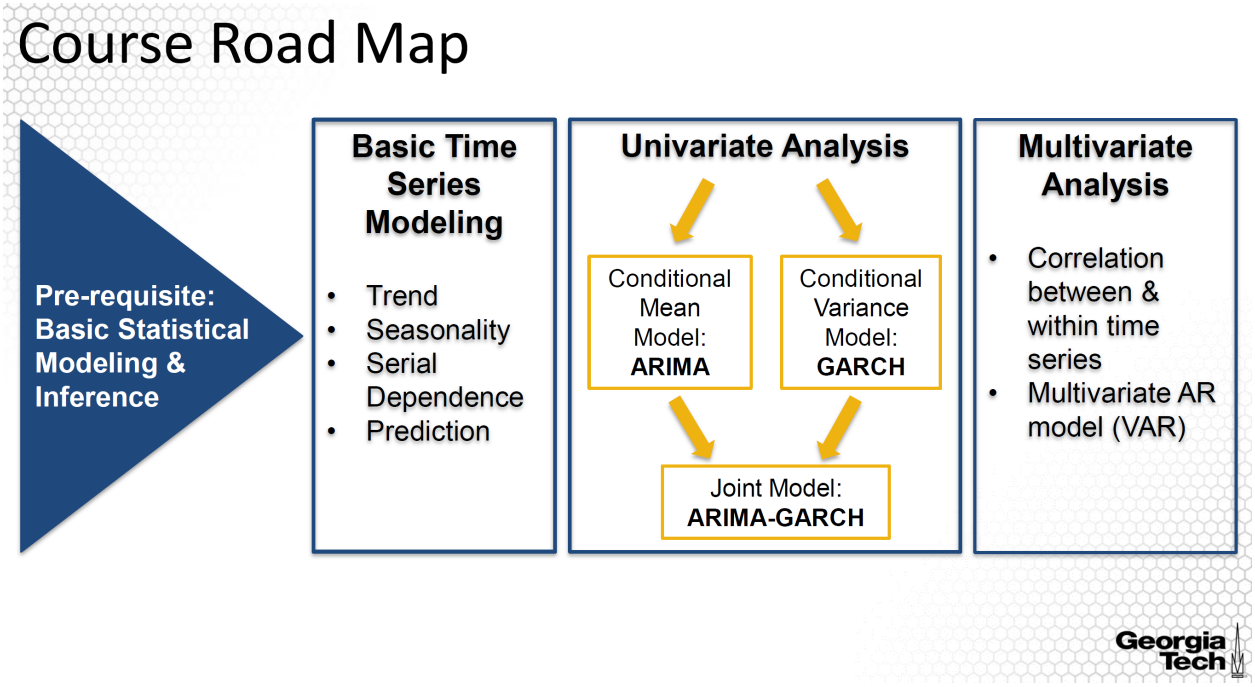
Table of Contents

Module 3: Modeling Heteroskedasticity

Module 3 Road Map

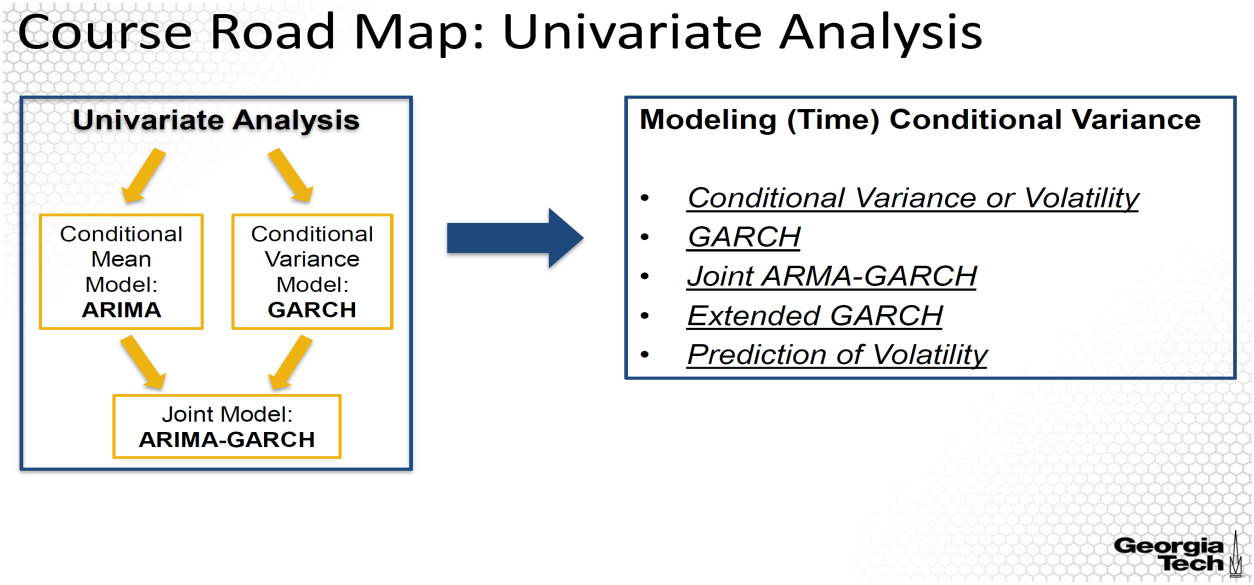
This lesson is an overview of the content covered in the second module of the Time series Course.

Course Road Map



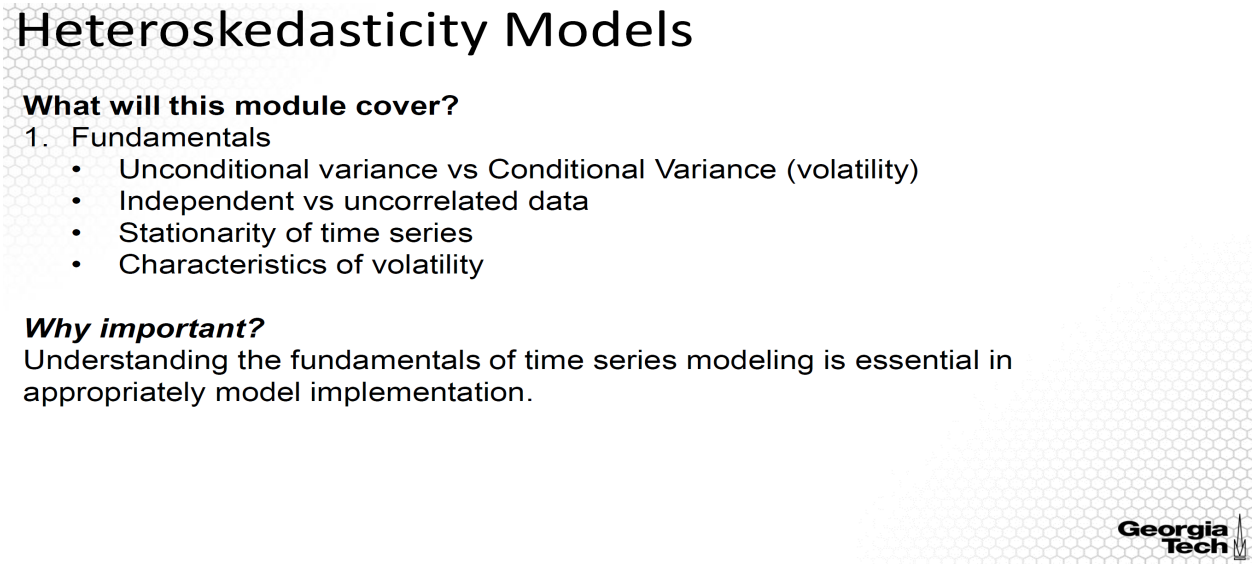
This course roadmap displays the material covered in the course starting with the pre-requisites needed for a better and deeper understanding of the different time series models introduced in this course. The course begins with basic concepts of time series analysis including trend and seasonality analysis as well as an understanding of the dependence in a time series. The rest of the course follows in presenting three main modeling approaches commonly employed in time series analysis; they are the ARMA, GARCH and VAR models.

Course Road Map: Univariate Analysis



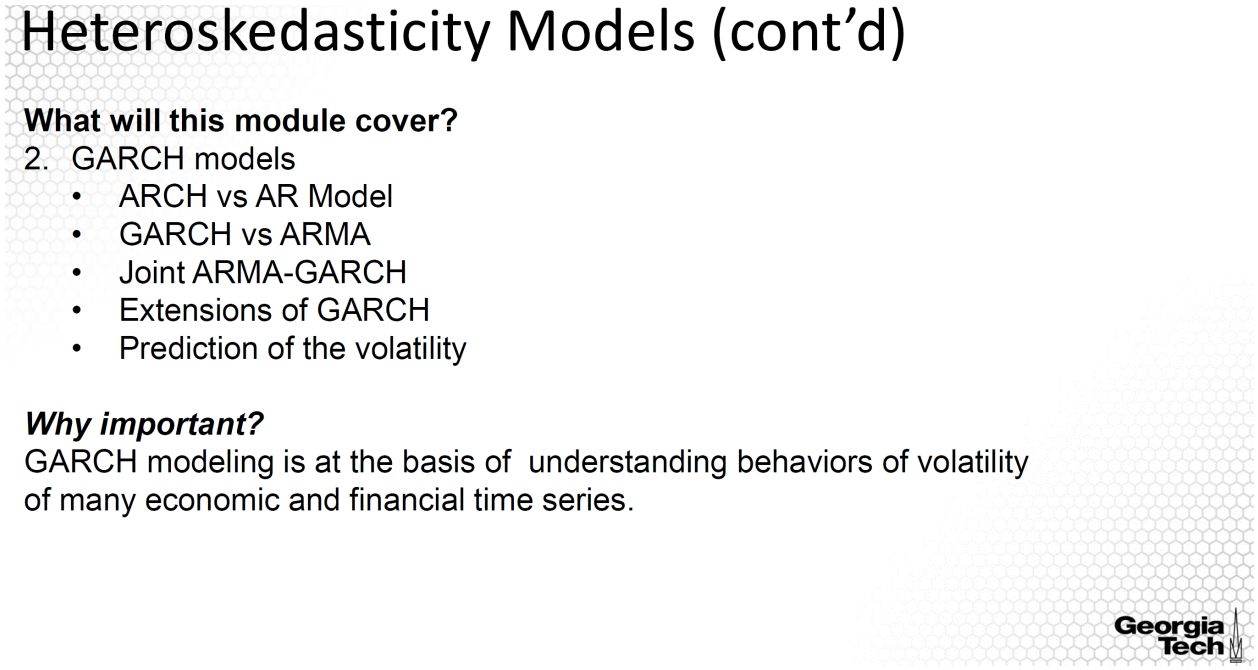
Univariate models in time series analysis focus on the conditional mean and/or the conditional variance. Models for the variance given the past data of a time series are called heteroskedasticity models.
Module 3 introduces one of the most common heteroskedasticity modeling approaches for time series analysis, specifically, the so called GARCH models; GARCH stands for Generalized Autoregressive Conditional Heteroskedasticity. GARCH applies to time series that are stationary but with a time-varying conditional variance. That is the variance given past data would vary with time; this is often called volatility.

Heteroskedasticity Models



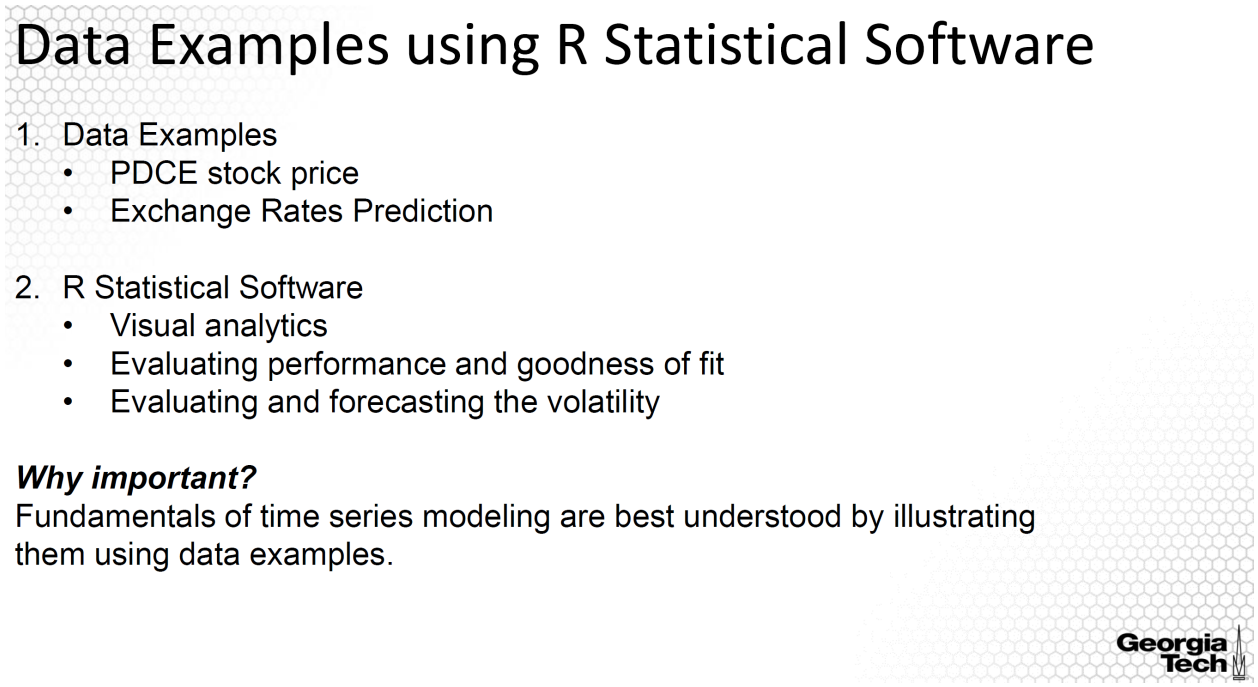
In this module, I will put the basis of understanding the difference between unconditional and conditional variance, and on the modeling of conditional variance using heteroskedasticity models. I will also once more underline the idea that uncorrelated data does not imply independence. In fact, we could have white noise with time-varying conditional variance. We also will discuss stationarity condition for such models.
Understanding the fundamentals of time series modeling is essential in appropriately model implementation.

Heteroskedasticity Models (cont'd)



We will begin with one of the simplest heteroskedasticity models, the ARCH model, which reduces to an AR model for the squared residual time series after accounting for the conditional mean. We then expand on the generalized ARCH or GARCH model, which reduces to an ARMA model for the squared residual time series. It is common to model both the mean and variance given past data of the time series using the joint ARMA-GARCH model. It is important to note that we are not only interested in predicting what would we expect the behavior of the mean to be in the future but also to predict its volatility. For many time series analysis, evaluating and predicting volatility is most relevant, for example, when interested in evaluating risk in investments. There are many extensions of GARCH accounting for behaviors of the volatility as I will illustrate in the latter lessons of this module.
We study these models because GARCH modeling is at the basis of understanding behaviors of volatility of many economic and financial time series.

Data Examples using R Statistical Software



Throughout this module we will also experiment the main concepts using two data examples. In these examples, we will analyze the stock price of the PDC Energy, a Crude oil and natural gas producer in the US; I selected this company due it the extensive volatility in price change. A second example is the analysis of exchange rates for USD-Euro, USD-Brazilian Real and USD-Chinese Yuan, there major currencies.
We will analyze these two data examples using the R statistical software. In particular, we will perform exploratory data analysis using visual analytics, we will evaluate the goodness of fit and the performance of the GARCH and ARMA-GARCH fit. We also will use these models for prediction of the mean and variance given past data.
It is important to practice with real data examples because fundamentals of time series modeling are best understood by illustrating them using data examples.

Summary:

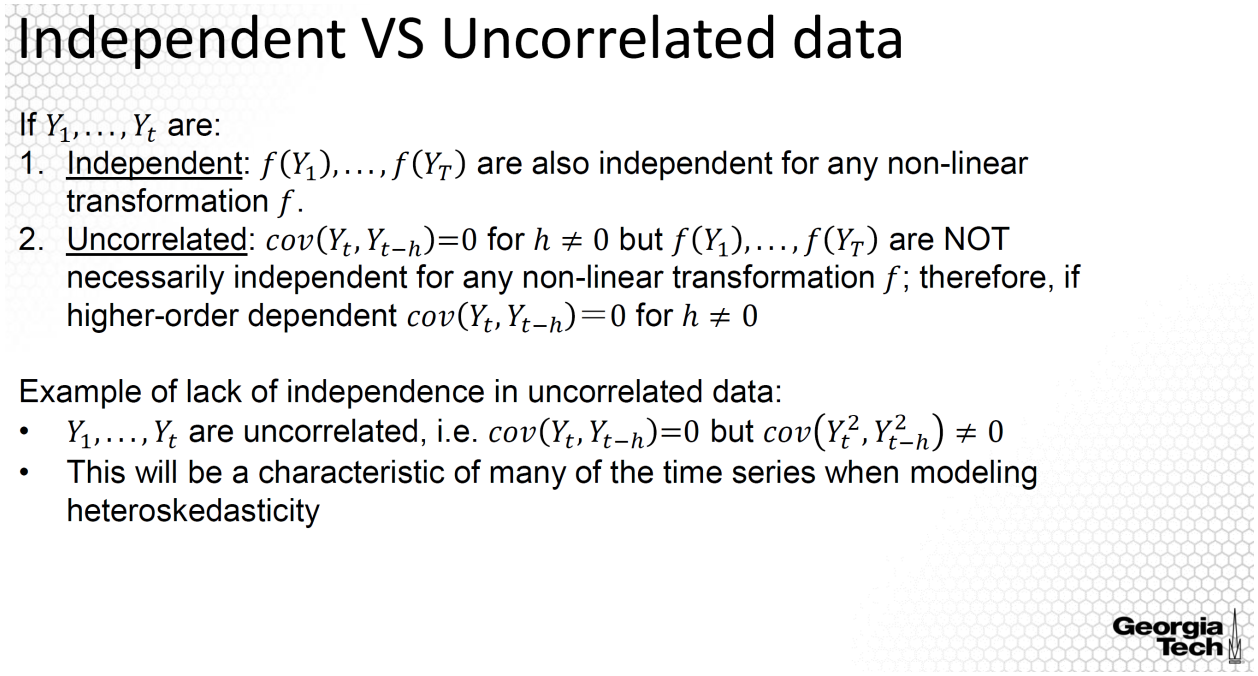
This lesson overviewed the main topics covered in Module 3. Let's now begin with the lessons overviewing the main concepts of heteroskedastic models.

3.1: Basic Models

3.1.1 Basic Concepts

In this lesson, I will introduce the basic concept of heteroskedasticity modeling in time series..

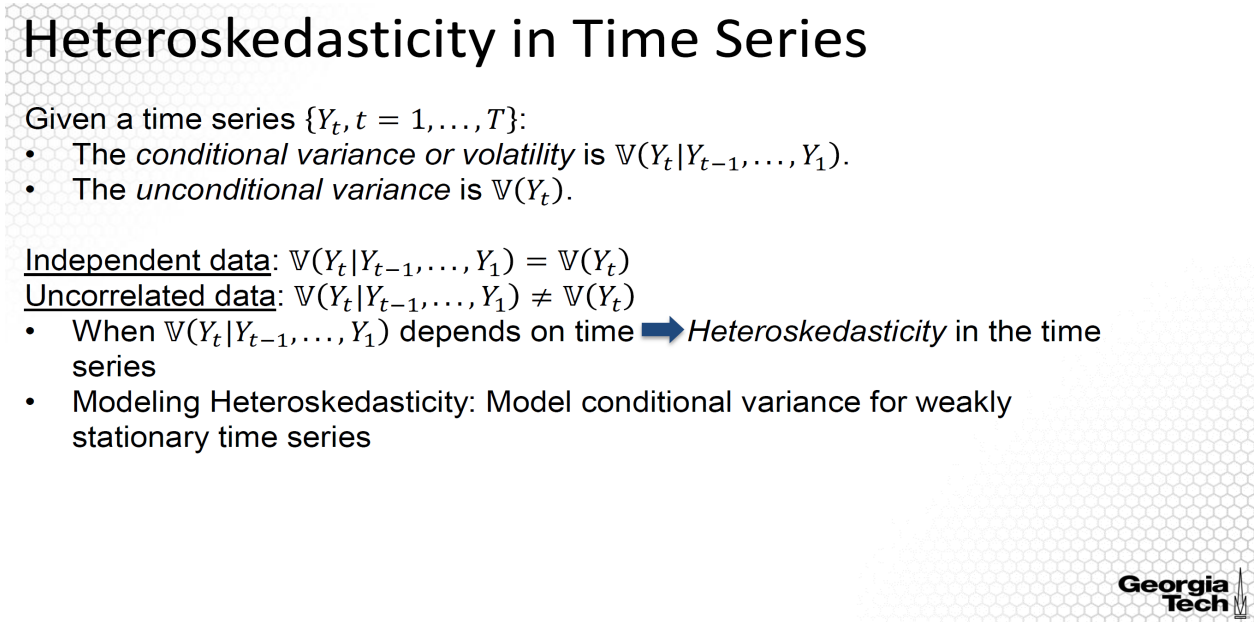
Independent VS Uncorrelated data



First, we'll learn about the difference between independent and uncorrelated data, which is at the basis of the modeling techniques introduced in this module. In the simple case, if the time series data are independent then what happens at time ‘t’ does not depend on what happened in the past. Independence implies lack of dependence or lack of correlation. On the other hand, in the case when the times series data are uncorrelated but not necessarily independent, the covariance between any two variables in a time series is zero meaning that there isn’t a linear dependence between any pairs of the variables in a time series; however, it is possible to find a transformation f() of the variables in a time series such that f of x1, f of x2 to f of xn are in fact, correlated. That is while there is no linear dependence, it is possible to have a nonlinear dependence; this distinguishes between independent and uncorrelated data.

One common transformation we'll use in this course is the power transformation, for example, square power transformation. In this example, while the covariance of Y_t and Y_t-h is zero, the covariance of the squared Y_t and squared Y_t-h is not zero. This will be a characteristic of many of the time series when modeling heteroskedasticity.

Heteroskedasticity in Time Series



Next, we'll learn about the difference between conditional and unconditional variance of a time series. The conditional variance of the time series at time ‘t’ is the variance of the time series, given the past values of the time series. In contrast, the unconditional variance of the time series at time ‘t’ is simply its variance disregarding the past data. We commonly refer to conditional variance as volatility. The difference between these two concepts can be better understood as we reflect on the difference between independent data and uncorrelated data. For example, under independence, the conditional variance is the same as the unconditional variance since the variability at time ‘t’ is independent of the past data. However, when the data are uncorrelated but not necessarily independent, then the conditional and unconditional variance are not the same. When the conditional variance depends on time, we say that we have heteroskedasticity in the time series. Note that because the conditional variance is not equal to the unconditional variance, it is possible that the unconditional variance to be constant over time, hence the time series to be weakly stationary, while displaying heteroskedasticity. In this module, we will focus on modeling heteroskedasticity for stationary time series.

Diagnosis: Heteroskedasticity in Time Series

Diagnosis: Heteroskedasticity in Time Series

Diagnostics for independent and uncorrelated data:

1. **Independent:** ACF of the time series $Y_1, \ldots, Y_t$ as well as of its non-linear transformations $f(Y_1), \ldots, f(Y_T)$ resemble the ACF of white noise.
2. **Uncorrelated but Dependent:** ACF of the time series $Y_1, \ldots, Y_t$ resemble the ACF of white noise but the ACF of at least one non-linear transformation of the time series (e.g. $Y_1^2, \ldots, Y_T^2$) does not resemble the ACF of white noise.

Georgia Tech

How can we diagnose independent and uncorrelated data? We have independence when the ACF plot of the time series, as well as transformations of the time series resemble the ACF of white noise. However, it is practically impossible to assess all possible transformations of the data. On the other hand, we have uncorrelated data but still dependent, if the ACF plot of the time series resembles that of white noise, but the ACF plot of a transformation of the time series does not. The difference between independent and uncorrelated data becomes apparent when we study the residuals of ARMA model applied to data with non-constant conditional variance. We'll see in some illustrative data examples that while the residuals are white noise, the squared residuals may not be white noise.

Modeling Heteroskedasticity

Modeling Heteroskedasticity

The class of heteroskedastic models is a generalization of the models so far:

- ARIMA models: estimation and inference on the conditional expectation: $E(Y_t|Y_{t-1}, \ldots, Y_1)$
- Heteroskedasticity models: estimation and inference on the conditional variance: $\mathbb{V}(Y_t|Y_{t-1}, \ldots, Y_1)$
- General model: $Y_t = \mathrm{f}(Y_{t-1}, \ldots, Y_1) + \sigma(Y_{t-1}, \ldots, Y_1)\ \varepsilon_t$

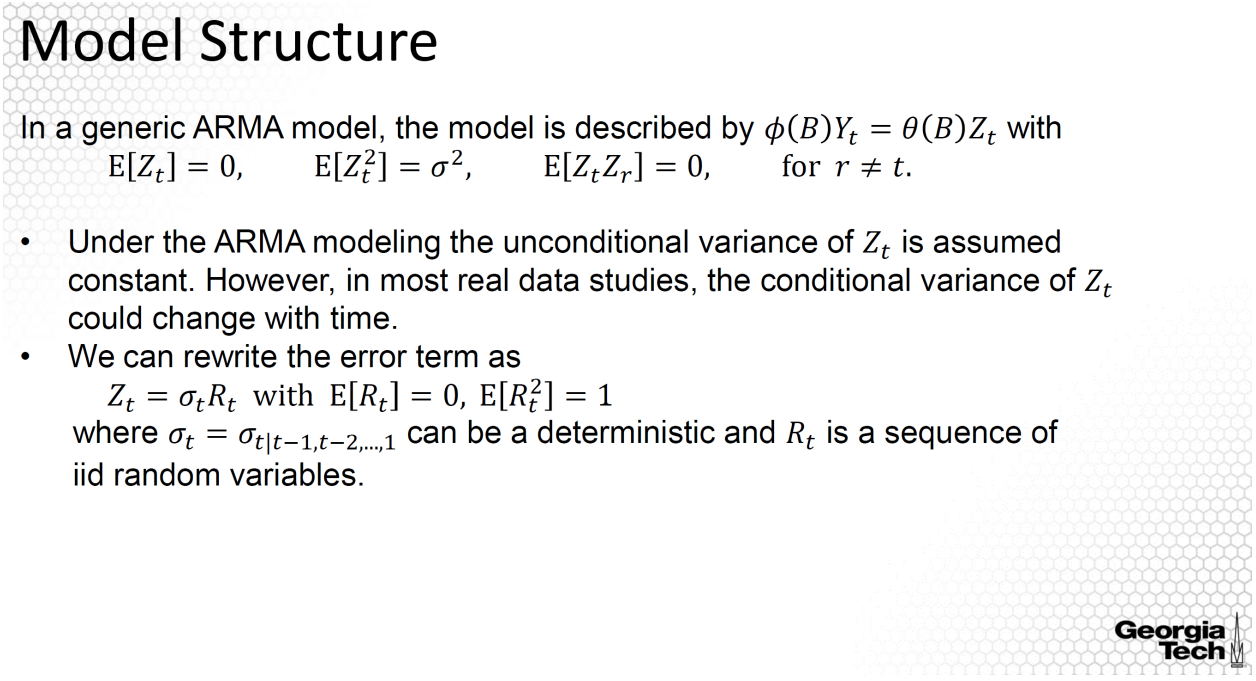
Example: Modeling stock returns or other financial indicators that display volatility.

Georgia Tech

The class of heteroskedastic models is a generalization of the models discussed in the first two modules of this course, where such models can be applied by further modeling the conditional variance. Specifically, ARMA models are used to model the conditional mean of a process given the past data. The class of heteroskedastic models hence focuses on modeling the conditional variance. More generally, we will discuss modeling approaches for the conditional expectation and conditional variance jointly.

A classic example is modeling stock returns. For example, suppose we have noticed that recently daily returns of a stock have been unusually volatile. A classic ARMA model cannot capture this type of behavior because it models the conditional expectation. We are not necessarily interested in the expectation in this example but the stock return volatility. To be even more specific, in financial data, it is very difficult to predict the mean of the price of a financial instrument such as stock price. If it were to be easy, other investors would predict the price as well and trade in such a way as to remove the advantage of the prediction. As a result due to the removal of the predictable information from the financial instrument by investors, almost all that is left is a random walk with a weak trend. The long-term growth rate reflects how to trade a financial instrument. If it has large volatility in the future, the returns need to be larger to attract investment to compensate for their risk of a loss, of a potential loss. More generally, volatility can be used as a proxy of the risk in management and investment. We will expand on this example in more detail in other lessons in this module.

Model Structure



To be more specific, for an ARMA model as provided on the slide, the residual process, Zt, is assumed to be white noise with constant variance given by sigma squared. Note however that the assumption of constant variance is for the unconditional variance. In many applications, time

series data often exhibit volatility clustering, particularly, the time series may show periods of high volatility and periods of low volatility; this would be reflected in the conditional variance. In this case, we can rewrite the error term Z_t as the product between a deterministic, but time varying component, sigma_t, and a random process R_t with mean 0 and variance 1. In the formulation provided in the previous slide, sigma_t is of interest as it models the conditional variance of the time series. Sigma t's often referred to as volatility in financial literature. In practice, the distribution of R_t often assumed to have a normal or t-distribution.

Characteristics of Conditional Variance

Characteristics of Conditional Variance

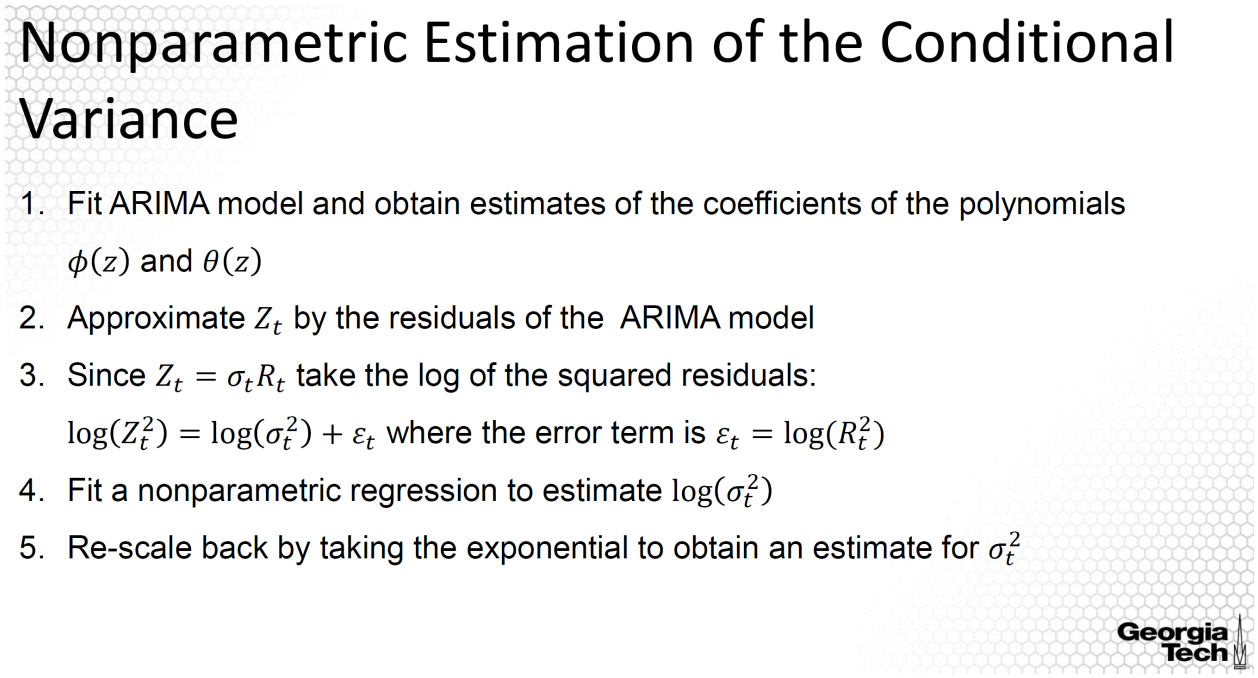
Some common characteristics of the function σ_t, commonly referred in financial literature as *volatility* are:

1. The function σ_t may be high for certain time periods and low for other periods - clusters of variability
2. The function σ_t varies with time in a continuous manner (i.e. there are no discontinuities, anomalies in the volatility)
3. The function σ_t varies within a specific range.
4. The function σ_t commonly is assumed to have the so called *leverage property*, i.e. it reacts differently to large amplitude changes from the small amplitude changes in the time process.

Georgia Tech

What are the important features of the conditional variance sigma_t? The sigma_t is a function of time and it may be high for certain time periods and low for other periods, also reflecting clusters of variability. The function sigma_t varies with time in a continuous manner. In other words, there are no discontinuities or anomalies in the volatility. The function sigma_t varies within a specific range. That means it's not infinite, it takes a finite range of values. The function sigma_t is commonly assumed to have the so-called leveraged property. In other words, it reacts differently to large amplitude changes from the small amplitude changes in the time process.

Nonparametric Estimation of the Conditional Variance

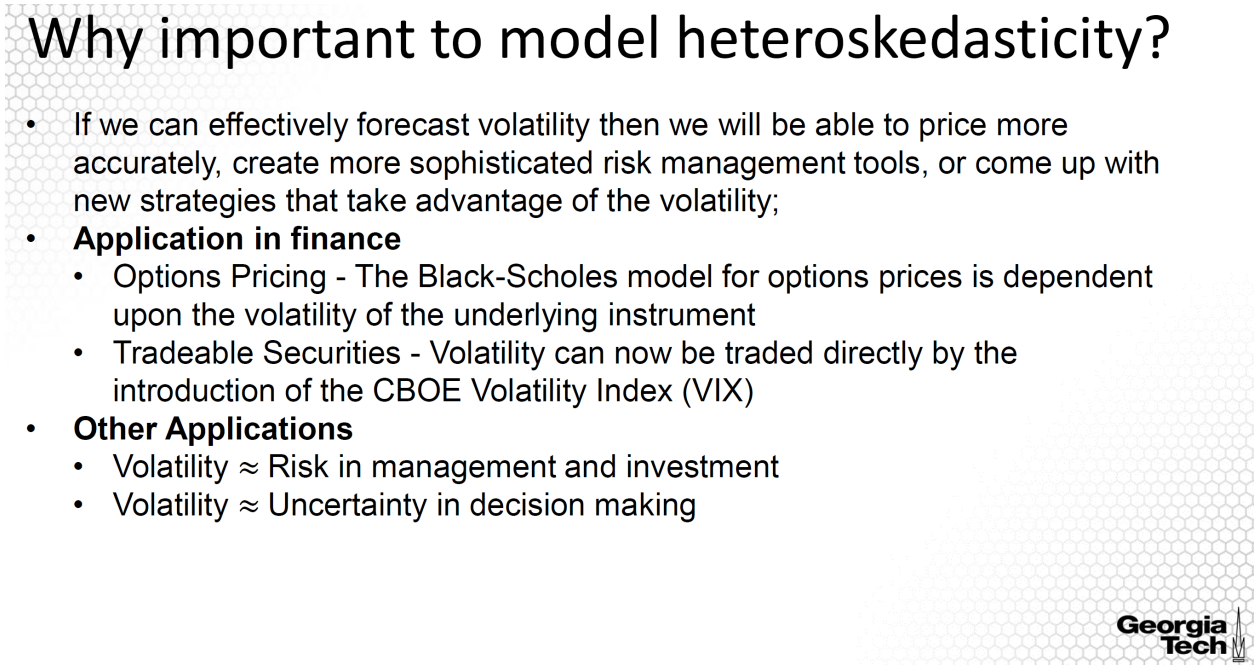


A straightforward approach to estimate the conditional variance is using nonparametric regression. Specifically, after applying a time series model for the conditional mean, for example, ARIMA model, we would approximate Z_t by the residuals of the model. Further, we can take the log of the squared residuals. Log of the squared Z_t is equal to log of the conditional variance plus an error term epsilon_t, which is log of squared Rt.

We can then use nonparametric regression on the log of squared residuals to estimate log of the conditional variance.

To obtain the estimate of the conditional variance, we can transform back by taking the exponential of the estimate of the log sigma square t. This approach can provide a good estimate of the conditional variance, if it changes smoothly over time.-For modeling more complex structures and volatility, other models can be considered as introduced in this module.

Why important to model heteroskedasticity?



Modeling heteroskedasticity has become important in the analysis of time series data, particularly in financial and economic applications. These models are especially useful when the goal of the study is to analyze and forecast volatility. If we can effectively forecast volatility, then we'll be able to price, for example, more accurately, create more sophisticated risk management tools or come up with new strategies that take advantage of the volatility.

Classic applications in finance where the estimation of volatility is needed are, for example, for option pricing, specifically in the Black-Scholes model for option pricing which is dependent upon the volatility of the underlying financial instrument. It can also be needed in tradable securities where volatility can be traded directly by the introduction of the CBOE volatility index, abbreviated VIX.

Moreover, volatility is a measure of uncertainty in any decision making. Similar principles as in investment for financial instruments apply within other settings where risk of uncertainty plays an important role in decision making.

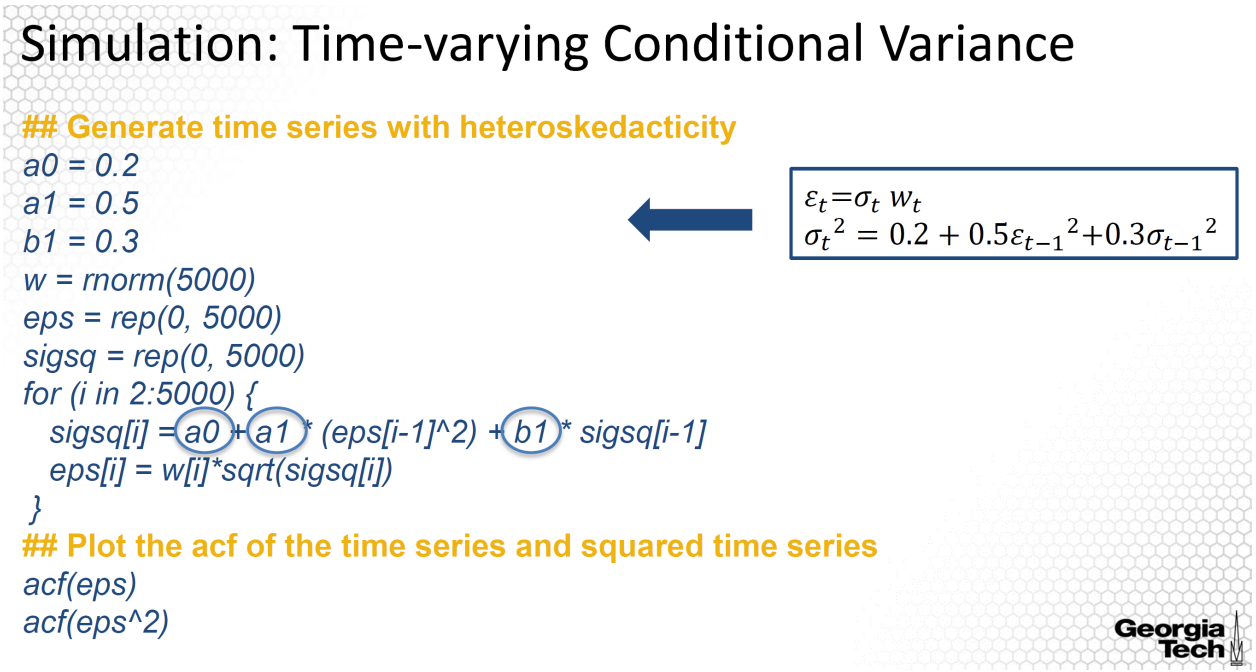
Summary:

This lesson introduces concepts related to modeling heteroskedasticity in time series. These concepts are at the basis of the modeling techniques described in this module.

3.1.2 Basic Concepts: Data Examples

In this lesson I will illustrate the concept of time-varying conditional variance or heteroskedasticity with a data example.

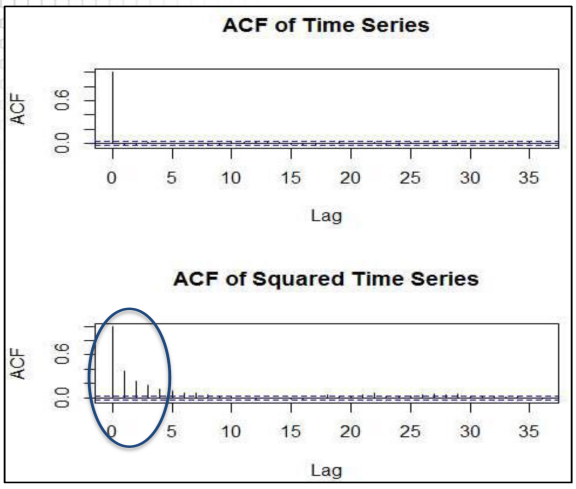
Simulation: Uncorrelated Time Series [Time-varying Conditional Variance]



Here I will illustrate how to simulate a time series with non-constant conditional variance. To simulate a time series with non-constant conditional variance in R, we need to simulate the vector ‘w’ for the random white noise, define a different vector called here ‘eps’ for the time series values, and a vector sigsq for the conditional variance. The coefficients a0 and a1 define the MA effects and the coefficient b0 defines the AR effect in the conditional variance as provided in the formula for sigma squared in the text box on the slide. The time series is simulated recursively by generating the time series value at time ‘t’ based on passed values of the volatility defined by sigsq. In the for-loop, we simulate the conditional variance or the volatility at time t as an ARMA process although the white noise is instead the squared time series eps_t-1 squared. To obtain the time series at time t, we multiply the simulated conditional variance at time ‘t’ with the white noise process w_t. The last two R commands are for plotting the ACF of the time series as well as of the squared time series.

Simulation: Time-varying Conditional Variance (cont'd)

Simulation: Time-varying Conditional Variance



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

Georgia Tech

The ACF plot of the time series is in the upper panel and the ACF plot of the squared time series is in the lower plot. While the ACF plot of the time series resemble that of white noise, the ACF plot of the square time series does not.

We see that the first three lags are outside of the confidence band, indicating serial correlation in the squared time series. This is an example of uncorrelated data, but not independent data since the squared time series is correlated data.

PDC Energy, Inc (PDCE)

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

Georgia Tech

We'll consider one data example of a financial instrument, the stock price of a company. I selected here a company with very large volatility, PDC Energy, which is a crude oil and natural

gas producer with headquarters in Denver, Colorado. Daily stock price for more than 12 years of data starting with January 2007 are considered in this data example. The reason for being interested in assessing volatility for this company is because it is highly dependent on the crude oil price, with extreme large volatility in recent years.

Financial Data Analysis

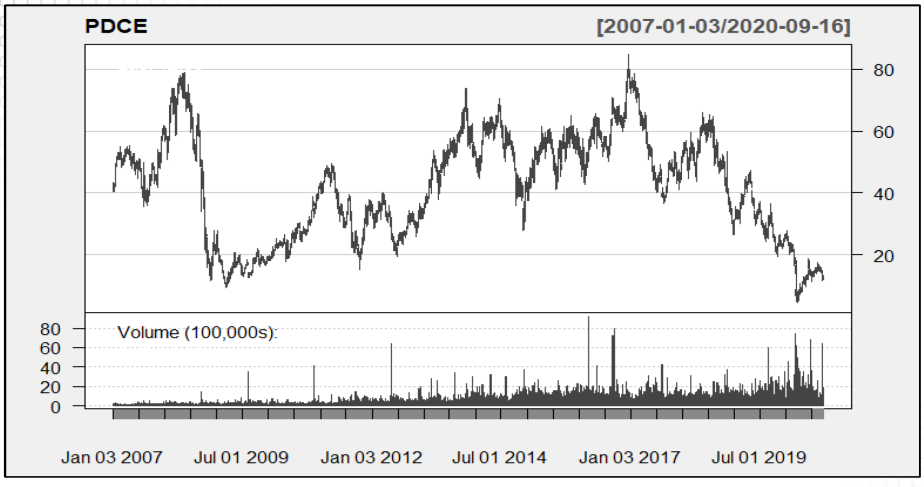
Financial Data Analysis

```
## 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="")
```

For this analysis, I'm using R functions in the quantmod package, which is designed to assist the quantitative traders in the development testing and deployment of statistically-based trading models. Using the R command getSymbols,-it's possible to load data from a variety of sources, including Yahoo Finance and Federal Reserve Bank. In this implementation, the source is Yahoo. The R command for plotting the time series is candleChart. We have seen other approaches to plot time series data. This command is yet another one! Here we are plotting the time series and its log return.

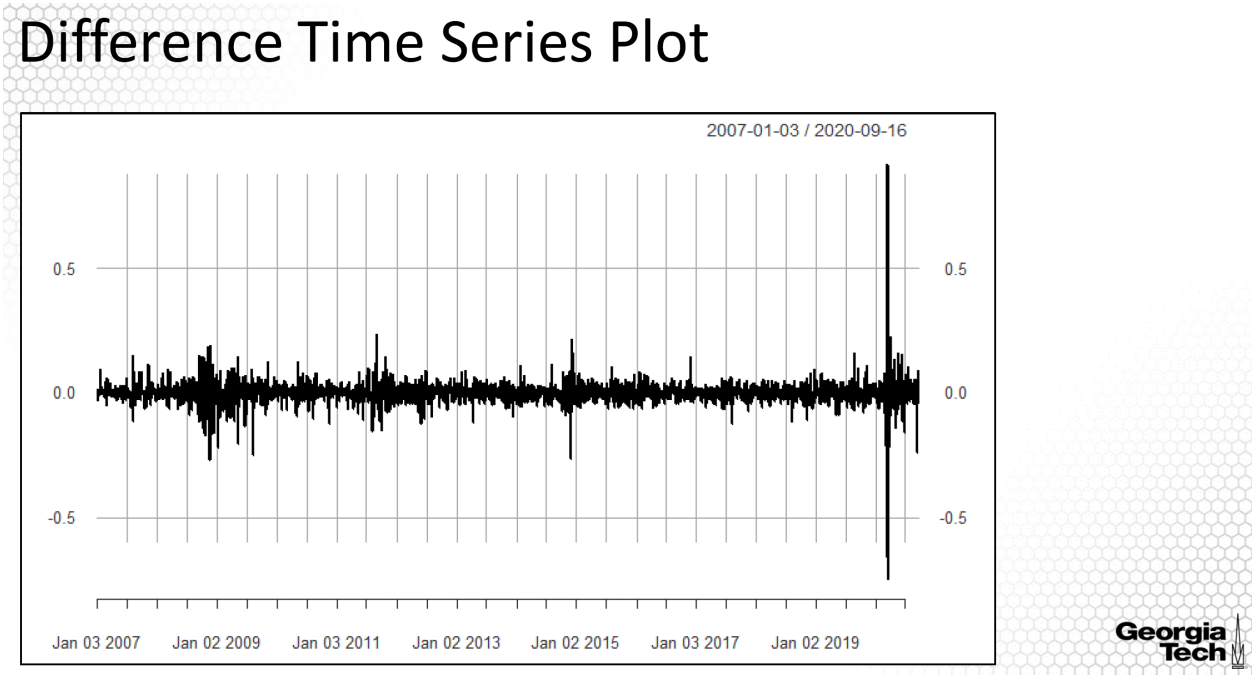
Time Series Plot: Price & Volume

Time Series Plot: Price & Volume



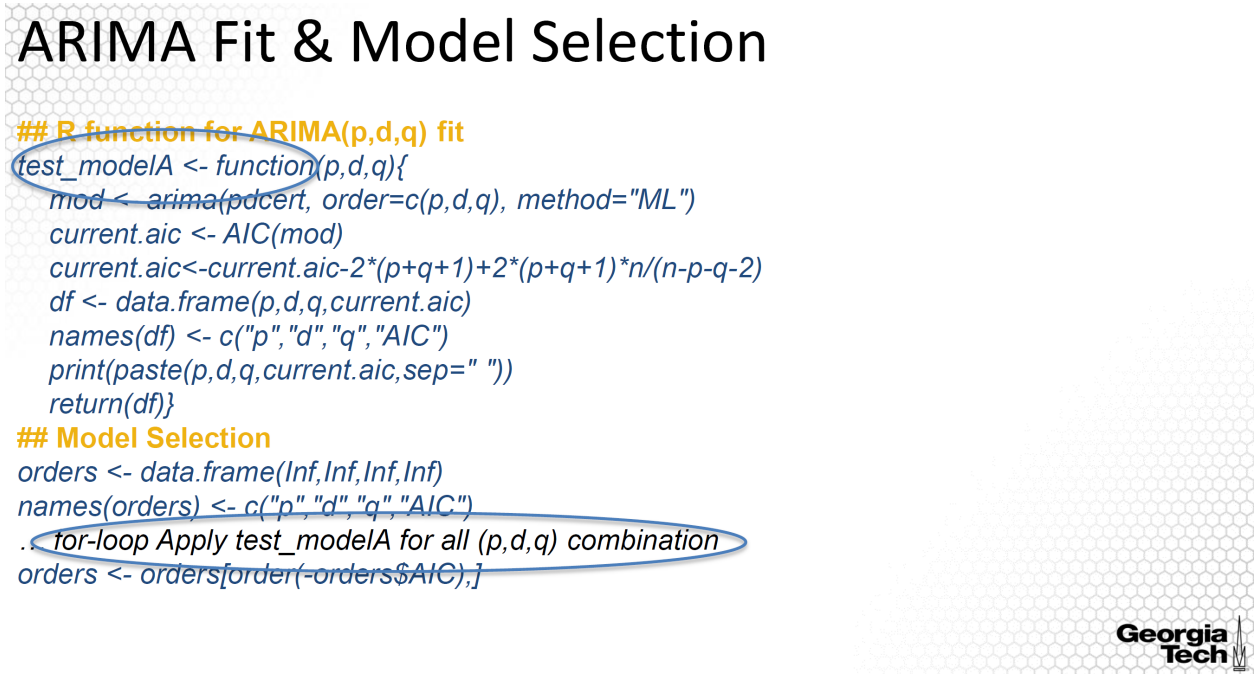
The time series plot of the close price along with the volume is on the slide. The time series is clearly non-stationary with a nonlinear trend. The price widely varied across years. It also seems that there are periods of very high volatility.

Difference Time Series Plot



A better way to visualize volatility is by plotting the return of the log price. Here's the time series of the return log price for PDCE. This time series clearly shows large volatility in the first quarter of 2008, for example, and as expected, during the covid19 crisis. This time series indicates that we have non-constant conditional variance or time-varying volatility.

ARIMA Fit & Model Selection



We'll first fit an ARMA model. For this, we'll select the order of ARIMA, including the order of ARMA and the order of the integrated part of the model, specifically, p & q for ARMA and d for the difference. The code provided here is different from the one we used in fitting ARIMA models in the previous module. Please note that the code on the slide is not complete; you can find the complete code in the accompanying R code file for this module. The reason for using the code provided here is that in some ARIMA models for log return prices, it is possible that not all combinations of p,d,q to result in convergence hence the code may terminate without running through all combinations. For this, I am providing here an R function, called '*test_modelA'* which fits an ARIMA model for the log return data given the ARIMA orders. It then outputs the orders and the AIC value. Then I applied this R function for all order combinations while making sure that the for-loop does not terminate if non-convergence. This last part is not shown on the slide. The output of this code is a data.frame called orders including information about AIC values and the corresponding orders. I also re-arranged this matrix such that the rows are ordered by AIC values.

ARIMA Fit & Model Selection (cont'd)

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

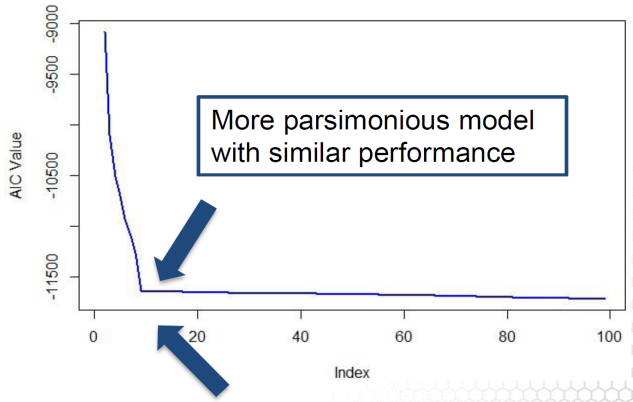
Georgia Tech

… with tail(orders) showing the last few rows, where the last row of the matrix corresponds to the orders with the minimum AIC. Below is the output showing the last rows in the orders matrix. The last row provides the orders with the lowest AIC value hence the selected orders using this approach are p=4, d=0 and q=4. On the slides in the right is the plot of the AIC values.

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



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

Georgia Tech

However, if we were to look at the AIC plot as provided on the slide, the AIC values drop significantly for the orders p=0, d=1 and q=1 then stay relatively small.
This suggests a more parsimonious model with these orders than the selected model with a similar performance in terms of AIC. Generally, we prefer more parsimonious or less complex models in forecasting.

ARIMA Fit: Residual Analysis

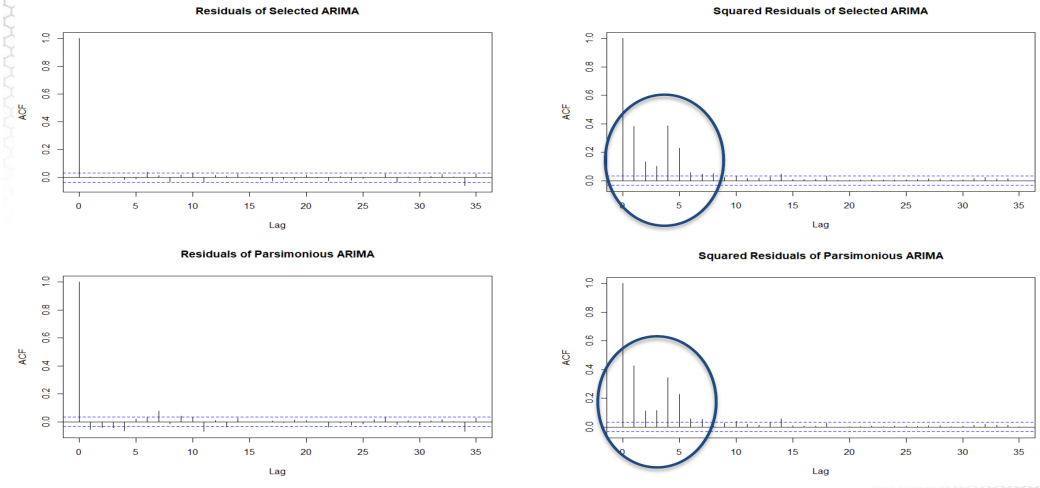
ARIMA Fit: Residual Analysis

```
## 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

Next we perform the residual analysis for the two models. Specifically, we plot the ACF of the residual time series and of the squared residuals time series as provided on the slide. We do so for both models, the selected model with p=4, d=0 and q=4 and the parsimonious model with p=0, d=1 and q=1.

ARIMA Fit: Residual Analysis

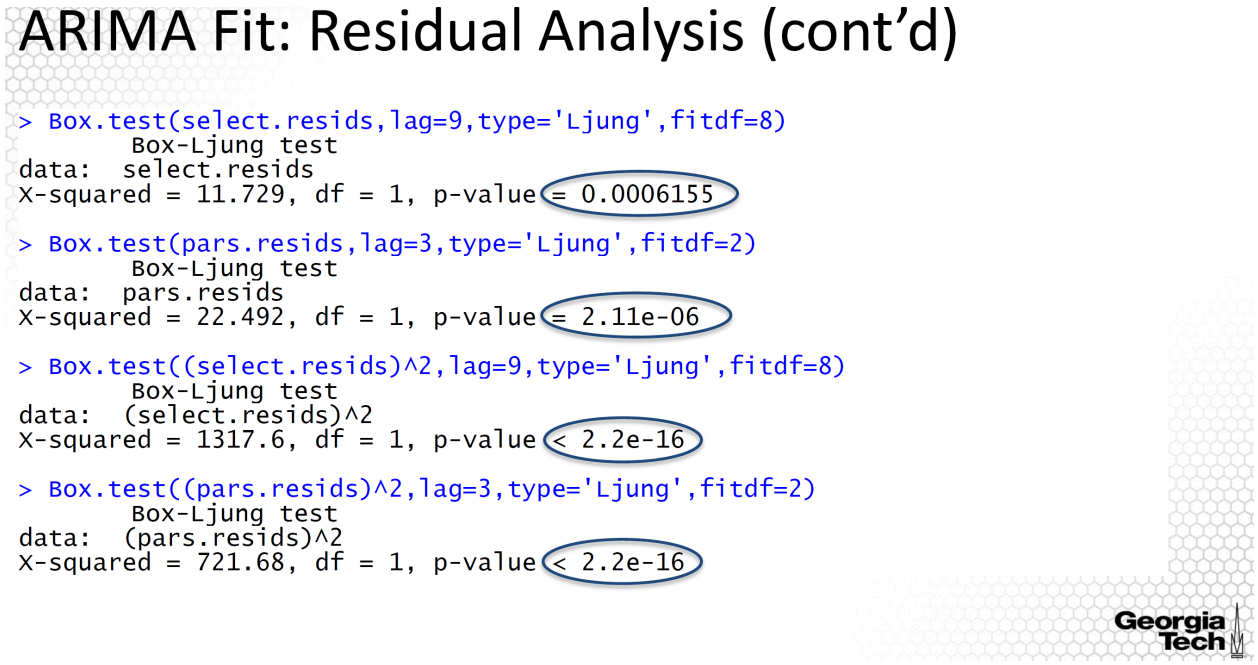


Georgia Tech

The ACF plots are here. The ACF plot of the residuals for both models looks like one for the white noise process. On the other hand, the ACF plot of the squared residuals suggests that there is a serial dependence in the squared residuals.

There are few ACF values for lags 1 to 5 of the ACF plot of the squared residuals outside of the confidence band. Thus, the residuals are linearly uncorrelated, but not independent since the transformed residuals, specifically the squared residuals, are correlated.

ARIMA Fit: Residual Analysis (cont'd)

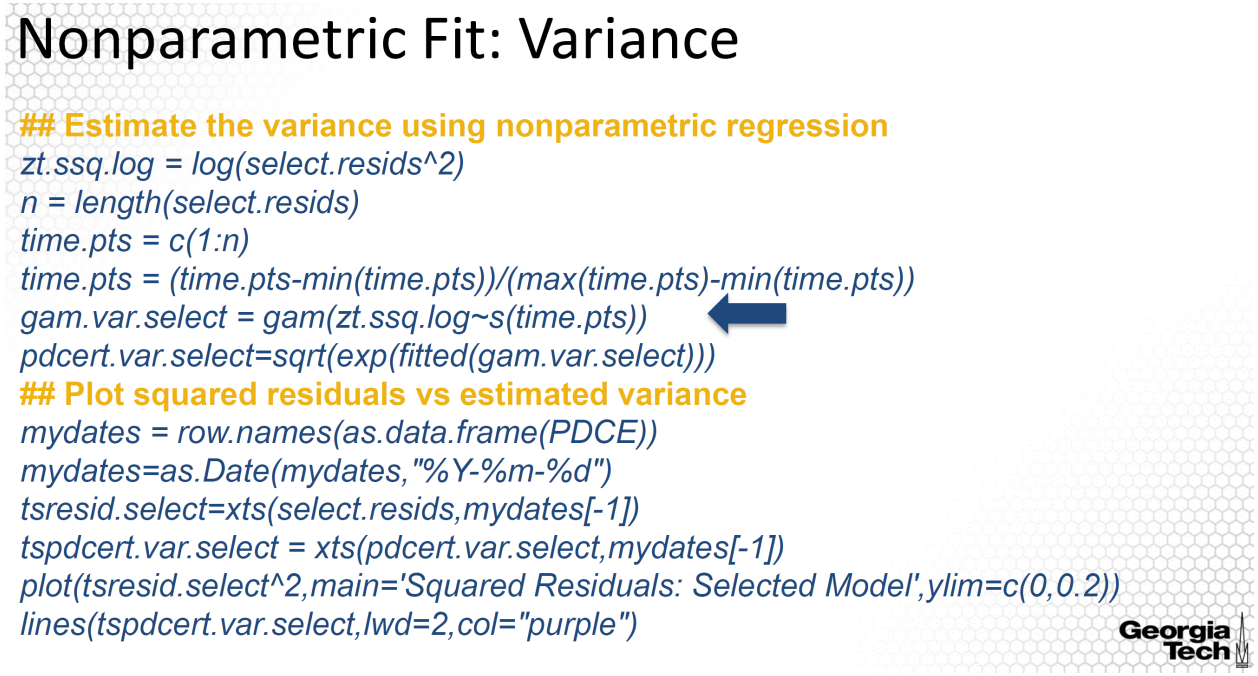


Let's test whether the residuals and square residuals are correlated using hypothesis testing. We apply here the Box-Ljung test. Note that the test is not for testing independence, but for testing whether the data are uncorrelated where the null hypothesis is that the data aren't correlated versus the alternative that the data are correlated.

The p-values for testing uncorrelated residuals are very small for both models, an indication that the residuals are correlated. However, the ACF plots from the previous slide look similarly to those of white noise.

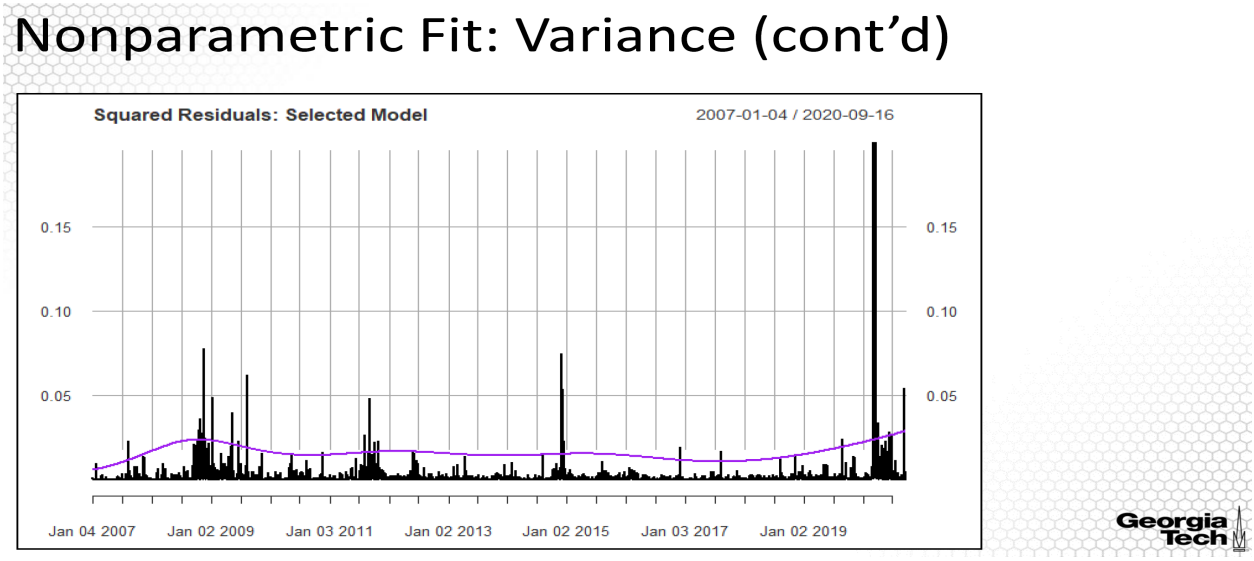
The p-values for testing uncorrelated squared residuals are also very small indicating that we reject the null hypothesis, and thus, conclude that the squared residuals are correlated. Other tests can be applied as shown in one of the lessons of Module 1.

Nonparametric Fit of Variance

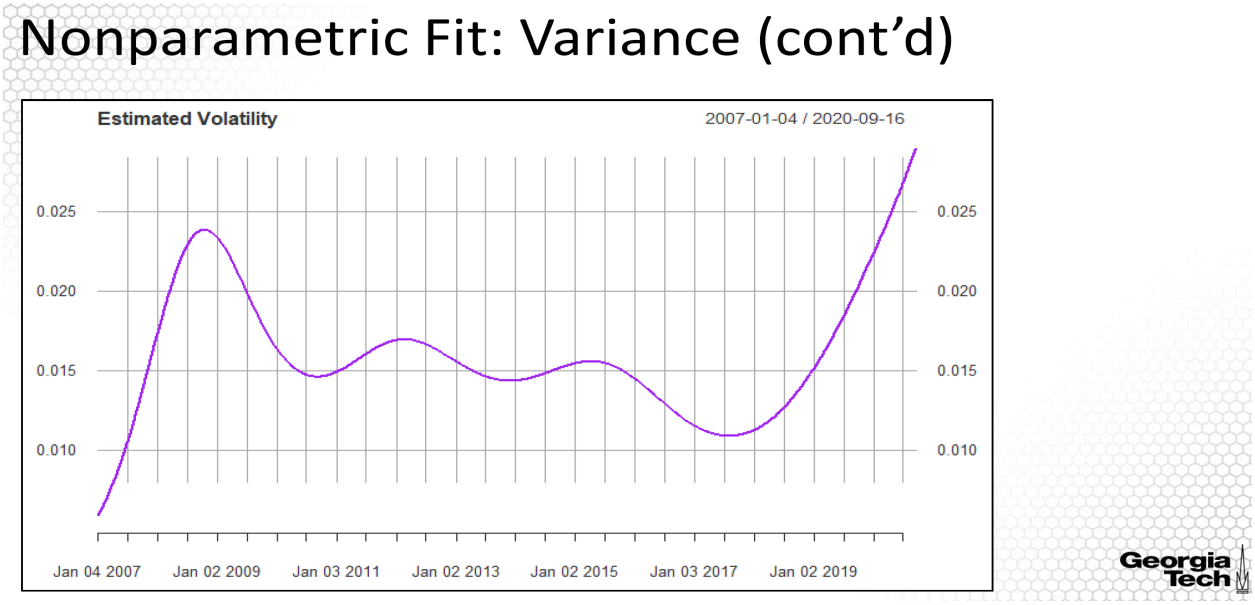


Last, we estimate the variance using nonparametric regression. For this, we take the log of squared residuals, and apply the nonparametric regression approach. We can apply the splines regression model using the gam function in the mgcv library of R. In this implementation, the response is the log of squared residuals. I am applying this approach only to the residuals from the selected ARMA model. We further compare the fitted estimated trend from this nonparametric regression to the squared residuals. For this, we have to extract the vector of dates as provided on the slide so that we will define the time series for the squared residuals and for the variance estimate.

Nonparametric Fit of Variance (cont'd)



The plot of the fitted volatility is here. The estimated variance is shown in purple. We can see that the volatility has a nonlinear trend with several periods of years of high volatility. I will note here that I truncated the y-axis scale since the covid19 crisis again has shown volatility that is multiple folds higher, than experienced in previous years. The covid19 crisis was compounded with a large drop in oil prices that have led to such large volatility.



To see the clear volatility pattern, I also plotted the estimated variance alone without the squared residuals. From this plot, it is interesting to see that the volatility still has an upward trend in these current days, the most recent days of the data.

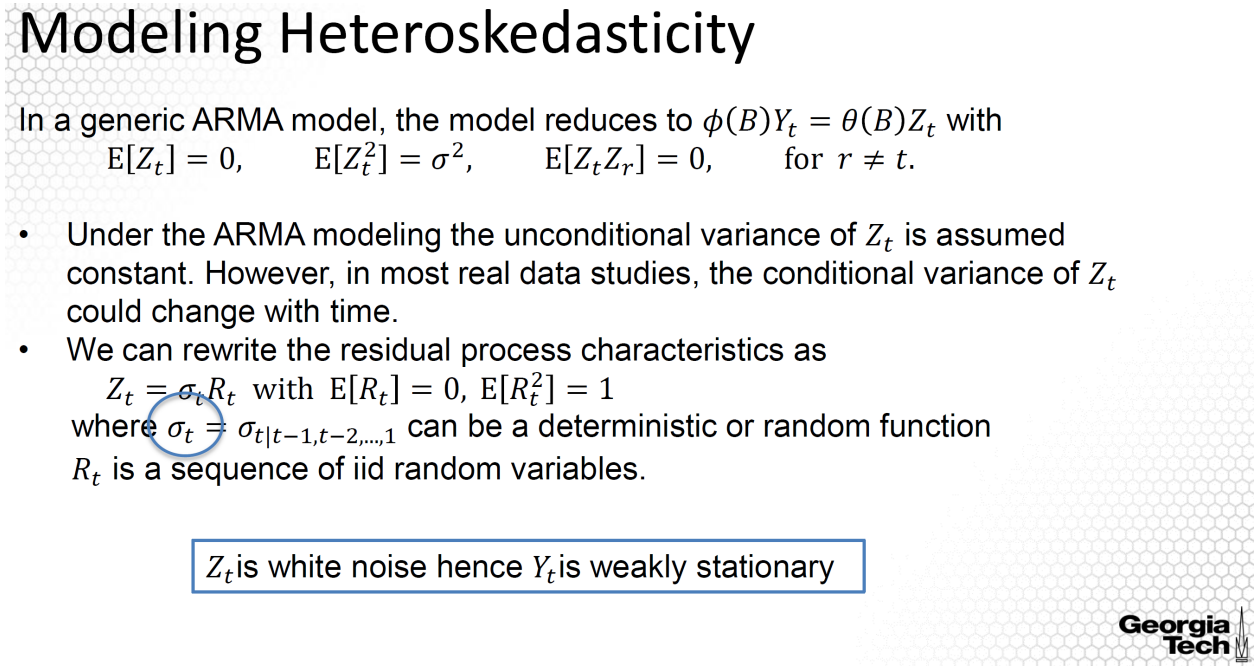
Summary:

In summary, in this lesson, I illustrated the concept of non-constant variability or heteroskedasticity with specific data examples.

3.1.3 ARCH Model

In this lesson, I'll introduce one of the simplest time series approaches to model the conditional variance, the so-called ARCH model.

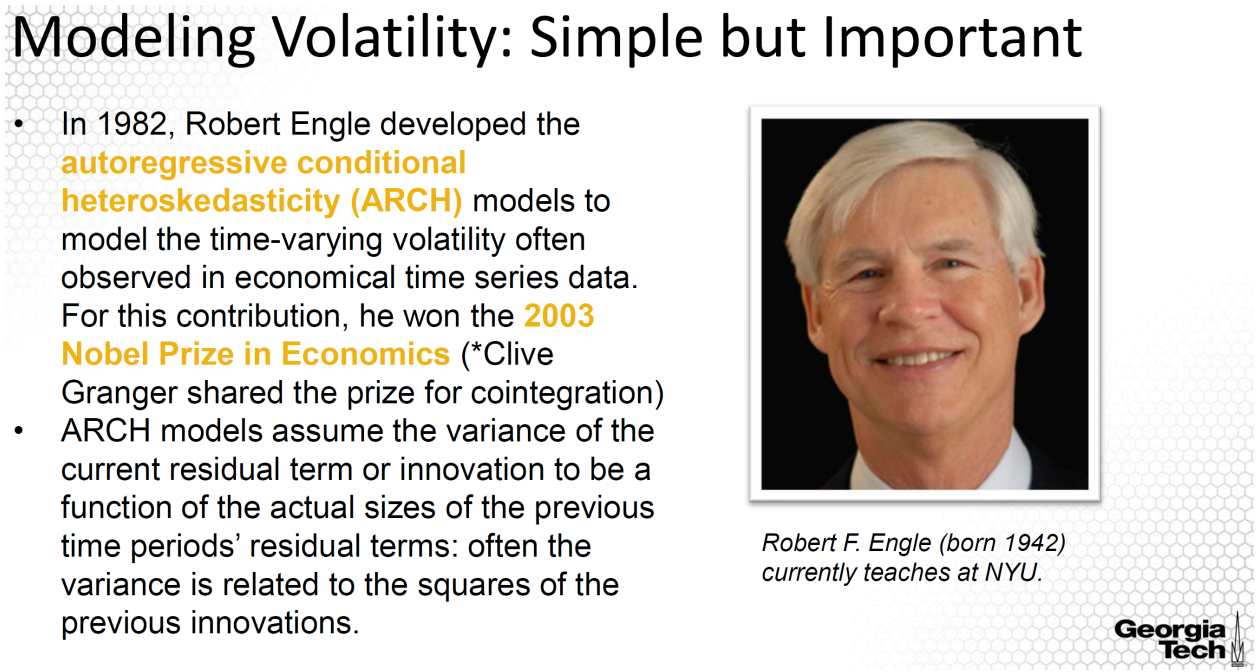
Modeling Heteroskedasticity



We begin with an ARMA model applied to a time series Yt as provided on the slide, where the residual process is Zt assumed to have constant unconditional variance given by sigma squared. However, in many applications the time series data often exhibits volatility clustering where the time series shows periods of high volatility and periods of low volatility indicating non-constant variance. In such cases, we can rewrite the residual process Zt as the product between a time varying component sigma_t and the random process Rt with mean 0 and variance 1 as provided on the slide. In this lesson, we'll focus on modeling the volatility defined by sigma_t. I will point out that the models provided in this course assume that the volatility or conditional variance sigma_t is deterministic, that is, it is simply a function of time. However, there are heteroskedasticity models that assume that sigma_t is a random process; those models are more challenging since we would need to account for uncertainty not only in the time series but also in the volatility process.

I will highlight here that in this formulation, the residual process Zt is still white noise and hence the time series Yt is assumed weekly stationary. However, as I discussed in the previous lessons, it is the conditional variance that is non-constant over time and hence it is the conditional variance of Zt that we model here.

Modeling Volatility: Simple but Important



We'll begin with one of the simplest time series models, introduced by Robert Engle in 1982. The autoregressive conditional heteroskedasticity or in short, ARCH model, is commonly used to model the time-varying volatility often observed in economical time series data. For his contribution related to ARCH family of heteroskedasticity model, Robert Engle won the 2003 Nobel Prize in Economics. ARCH models assume the variance of the current residual process or innovation to be a function of the actual sizes of the previous time period's residual terms. Often the variance is related to the square of the previous innovations or residual terms.

The ARCH Family of Models

The ARCH Family of Models

The *Autoregressive Conditional Heteroskedasticity (ARCH)* model:

If σ_t is a linear function with lagged values of the mean equation residuals, then the time-series dynamic of volatility is like an AR process.

$$Z_t^2 = \gamma_0 + \gamma_1 Z_{t-1}^2 + \ldots + \omega_t$$

AR process

$$\mathrm{E}[\omega_t] = 0, \qquad \mathrm{E}[\omega_t^2] = \lambda^2, \qquad \mathrm{E}[\omega_t \omega_r] = 0, \qquad \text{for } r \neq t.$$

Conditional on the past history $F_{t-1} = \{Z_{t-1}, Z_{t-2}, \ldots\}$ we have

$$\mathrm{E}[Z_t^2 | F_{t-1}] = \sigma_t^2 = \gamma_0 + \gamma_1 Z_{t-1}^2 + \gamma_2 Z_{t-2}^2 + \ldots$$

Reference: Robert Engle. "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation." *Econometrica*, 50, pages 987:1007 (July 1982)

Georgia Tech

The ARCH family of models assume that the volatility, or sigma_t squared, is a linear function of lagged values of the residual process Zt. Thus, the time-series dynamics of the volatility is like an AR process. Specifically, the squared residuals Zt are modeled using an AR process, where gamma 0, gamma 1, and so on, are the AR coefficients, and omega t is the error term assumed to be uncorrelated with mean 0 and variance lambda squared. The expectation of Zt squared, given past history of the residual process, is equal to the AR polynomial for Zt squared. But, the expectation of Zt squared is also equal to sigma t squared. Hence, we model the conditional variance as a linear combination of past squared residual terms; this is deterministic since it depends on past data and the AR coefficients, which are unknown and need to be estimated. In this course, A process such as the ARCH process where the conditional mean is constant but the conditional variance is not constant, is an example of uncorrelated but dependent process. The dependence of the conditional variance on the past causes the process to be dependent.

ARCH vs AR Model

ARCH vs AR Model

For example, we consider the $ARCH(1)$ model:

$$Y_t = \mu + Z_t, \qquad Z_t|F_{t-1} \sim N(0, \sigma_t^2) \text{ and } \sigma_t^2 = \gamma_0 + \gamma_1 Z_{t-1}^2$$

σ_t is a deterministic estimate of volatility. Since we do not observe volatility with certainty, the true volatility of Z_t is defined as:

$$Z_t^2 = \sigma_t^2 + \omega_t \ (Z_t = \sigma_t R_t) \text{ with } \mathrm{E}[\omega_t] = 0 \text{ and}$$
$$\mathrm{E}[\omega_t^2] = 1 \ (\mathrm{E}[R_t] = 0 \text{ and } \mathrm{E}[R_t^2] = 1)$$

Then the above equation for σ_t can be re-expressed as:

$$Z_t^2 - \omega_t = \gamma_0 + \gamma_1 Z_{t-1}^2$$
$$(1 - \lambda_1 B) Z_t^2 = \gamma_0 + \omega_t$$

This is an AR(1) in Z_t^2 where ω_t is the error between the true conditional variance σ_t^2 and actual volatility Z_t^2.

Georgia Tech

Again, in ARCH models, the conditional variance has a structure very similar to the structure of the conditional expectation in an AR model. I will explain here this concept with the simplest ARCH model: ARCH of order 1. We'll thus consider here Yt to be-a process with a mean that does not vary with time, plus the Zt process that is modeled using an ARCH(1) model. That is, the conditional variance is equal to gamma0 + gamma1 Zt-1 squared. Note again that sigma_t is assumed deterministic. However, we do not observe volatility with certainty. The true volatility of Zt is sigma_t squared, which is deterministic, plus omega t, which is random, where omega_t has mean 0 and variance 1. Those two processes, Zt and omega_t, are also assumed uncorrelated. Together this leads to the AR(1) equation for Zt squared, as provided on this slide.

ARCH Model: Interpretation

ARCH Model: Interpretation

Consider the simple equation of ARCH(1):

$Z_t^2 = \sigma_t^2 + \omega_t \ (Z_t = \sigma_t R_t)$ with $\mathrm{E}[\omega_t] = 0$ and $\mathrm{E}[\omega_t^2] = 1$ ($\mathrm{E}[R_t] = 0$ and $\mathrm{E}[R_t^2] = 1$)

The propagation effect:

- If Z_{t-1} has an unusually large absolute value, then σ_t larger than usual;
- When Z_t has a large deviation that makes σ_t^2 large, so that Z_{t+1} tends to be large, and so on
- Volatility in Z_t tends to persist throughout for a long time

Georgia Tech

This equation is crucial in understanding how ARCH processes work. If Zt-1 has an unusually large absolute value, then sigma_t is larger than usual and thus Zt is also expected to have an unusually large magnitude. This volatility propagates since when Zt has a large deviation that makes sigma_t+1 squared large, so that Zt+1 tends to be large, and so on. Similarly, if Zt -1 squared is unusually small, then sigma_t squared is small, and Zt squared is also expected to be small and so forth. Because of this behavior, unusual volatility in Zt tends to persist throughout but not forever.

Stationarity of ARCH(1)

Stationarity of ARCH(1)

The stationarity conditions are:

- $\mathrm{E}[Y_t] = \mu$
- $\mathrm{E}[Y_t^2] = \mathrm{E}[Z_t^2]$ computed as below

  $(1 - \gamma_1)\mathrm{E}[Z_t^2] = \gamma_0$

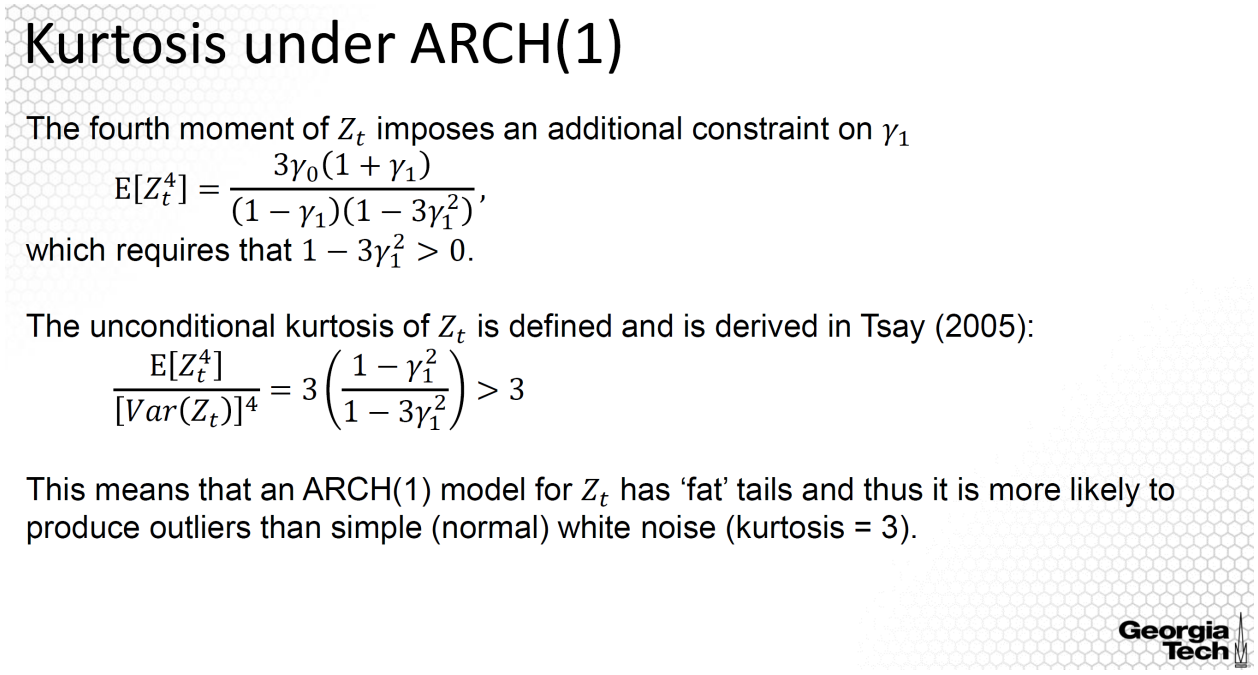
  $\mathrm{E}[Z_t^2] = \gamma_0/(1 - \gamma_1)$
- $Cov(Y_t, Y_{t-1}) = \mathrm{E}[Z_t Z_{t-1}] = 0$

The weak stationarity condition is that $\gamma_1 < 1$.

Georgia Tech

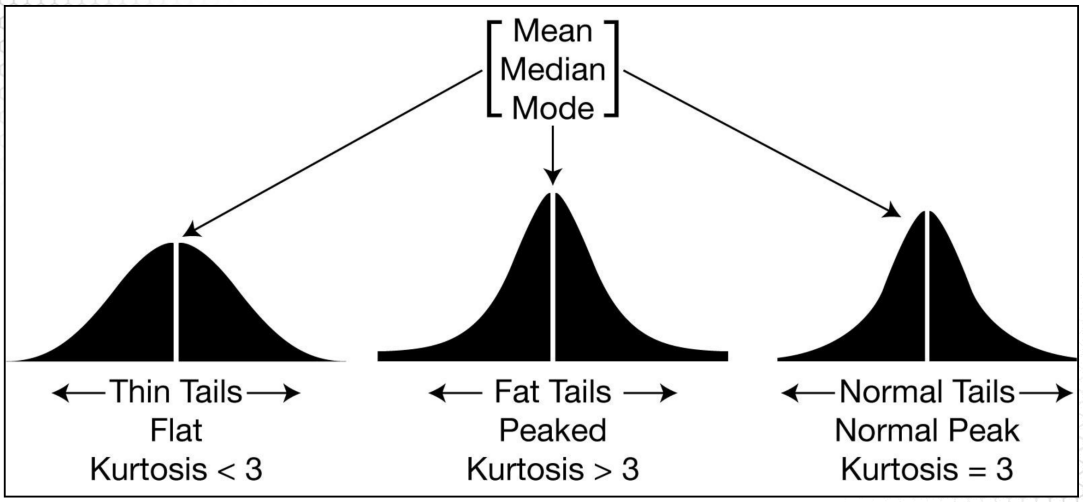
Because of the characterization of Zt squared as an AR process, we can further obtain the conditions of stationarity of the ARCH model. Using the same notation of the ARCH of order 1 process Yt, the conditions are as follows: the unconditional expectation of the time series Yt is mu, which does not dependent on time, the unconditional variance of Yt is the variance of Zt squared which can be shown to be equal to gamma0/(1-gamma1) where gamma0 and gamma1 are the AR(1) coefficients for Zt squared. We also need to have that Yt and hence Zt to be uncorrelated. We assume that the variance is finite and positive under stationarity, thus gamma 1 needs to be smaller than 1. These are the conditions under the so called weak stationarity. As I highlighted before, the ARCH process can be weakly stationary, as we saw here, under specific conditions, even when the conditional variance varies over time. This is the difference between weak and strong stationarity.

Kurtosis under ARCH(1)



In order to ensure strong stationarity, we need to impose additional constraints on gamma 1, specifically ensuring that the fourth moment of Zt is finite. The fourth moment, assuming normality, is provided on the slide. In order for this to be finite, we also need to impose the additional constraint that 3*(gamma1 squared) is smaller than 1. This is needed in the derivation of the kurtosis which is a measure of how fat the tails are. The kurtosis of the ARCH(1) model under the assumption of stationarity is larger than 3, which means that an ARCH(1) model for Zt has fat tails, and thus it is more likely to produce outliers than the simple normal white noise.

Kurtosis under ARCH(1)



Georgia Tech

The slide presents a comparison of distributions of different Kurtosis. For a distribution with Kurtosis with a coefficient larger than three, as the one on the middle, the curvature is high in the middle of the distribution and the tails are fatter than those of a normal distribution provided on the right. A distribution of data with fat tails is frequently observed in financial markets.

Examples of Joint ARCH models

Examples of Joint ARCH models

AR-ARCH models: for example, AR(1)-ARCH(1) is defined by the set of equations below

$$Y_t = \mu + \phi Y_{t-1} + Z_t$$
$$\sigma_t^2 = \gamma_0 + \gamma_1 Z_{t-1}^2$$

ARMA-ARCH models: for example, ARMA(1,1)-ARCH(2) is defined by the set of equations below

$$Y_t = \mu + \phi Y_{t-1} + Z_t + \theta Z_{t-1}$$
$$\sigma_t^2 = \gamma_0 + \gamma_1 Z_{t-1}^2 + \gamma_2 Z_{t-2}^2$$

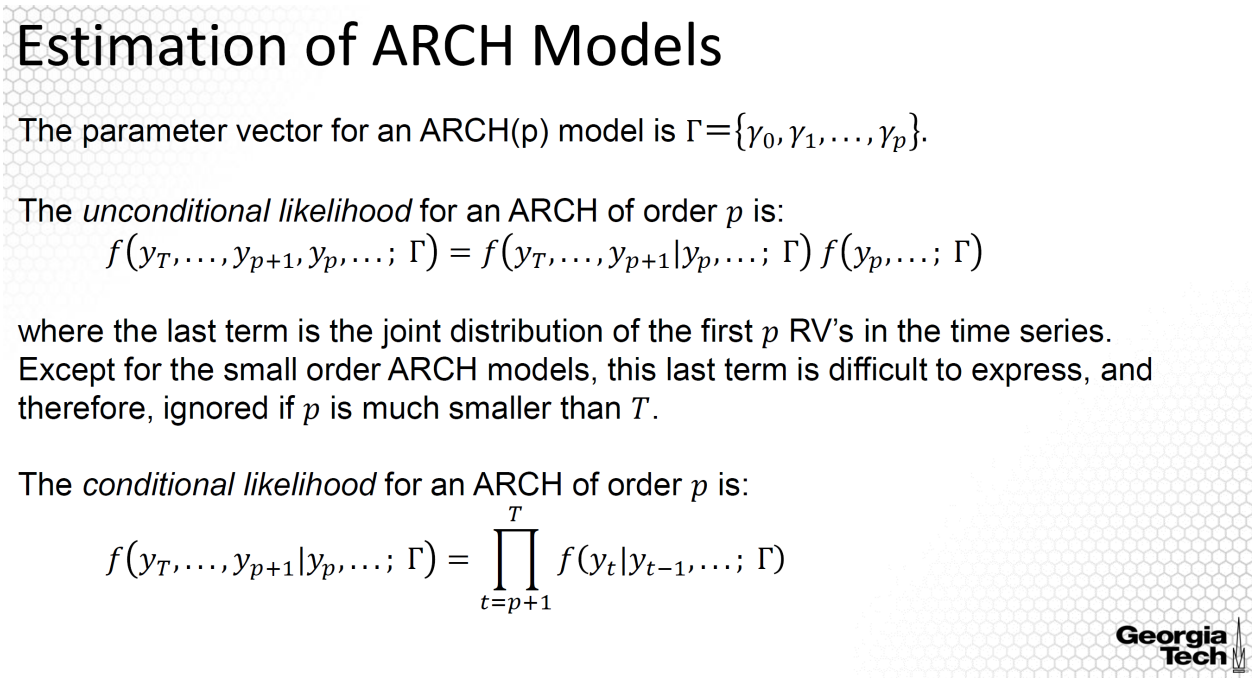
Georgia Tech

We can combine the ARCH model with an ARMA model as illustrated in the first slide of this lesson. Some specific examples are provided here. We can begin by assuming an AR(1) process for modeling Yt with mean mu and AR coefficient 5. Then the residual process Zt can

be assumed to be an ARCH (1) model. Here the condition of finite variance is gamma 0 + gamma 1 times Z t-1 squared.

A second example provided in this slide is the ARMA(1,1) and ARCH(2) joint models where Yt is modeled by an ARMA (1,1) with Zt, the residual process being modeled by an ARCH(2) model. Hence, the conditional finite variance is gamma 0 plus gamma 1 times Z_t-1 squared plus gamma 2 times Z_t-2 squared. It's important to note that when we have joint models, as provided here on this slide, we need to estimate them jointly. It's not good practice to first estimate the ARMA process, then to apply an ARCH model to the residual process. We will discuss this aspect with a data example in a different lesson.

Estimation of ARCH Models



How to estimate the ARCH coefficients, or the ARCH model? The common approach is using the maximum likelihood estimation. Because the variables in the time series Yt are not independent, then the likelihood function cannot be written as the product of the individual marginal distributions of Yt variables. Instead, it needs to be decomposed as a product of conditional distributions. I'm introducing here the concept of conditional likelihood. As discussed in the first lesson of the first module reviewing essential concepts in statistics, such as the decomposition of the joint distribution, we can decompose a joint distribution of the random variables Yt, Yt-1 up to Y1 modeled by an ARCH(p) model, using the unconditional likelihood as defined on the slide. The second term in the decomposition is a joint distribution of the first p random variables in the time series. Except for the small order ARCH models, the last term in the decomposition is difficult to express; therefore, it is commonly ignored if p is relatively small.

Thus, in fitting an ARCH(p) model, we instead maximize the conditional likelihood defined on the slide, which ignores the second term of the unconditional likelihood. The maximization of the conditional is with respect to the AR coefficients in the ARCH model. Since the estimation requires a numeric algorithm, it's only done using a statistical software such as R.

Summary:

In summary, in this lesson I introduced one of the simplest models used to model the conditional variance or heteroskedasticity in a time series, the so-called ARCH model.

3.1.4 ARCH: Data Examples

This lesson is on modeling heteroskedasticity in a time series, with a focus on a specific data example for illustrating the ARCH model.

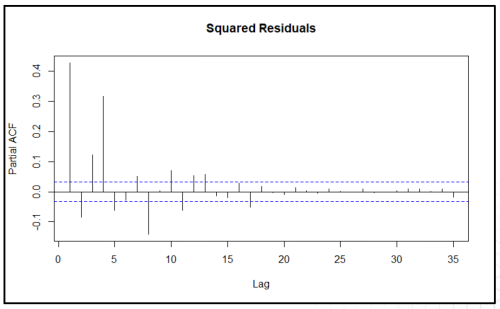
PDC Energy, Inc (PDCE)

We'll return to the data example in which we will model the stock price of a company. I selected here a company with very large volatility, PDC Energy, which is a crude oil and natural gas producer with headquarter in Denver, Colorado. Daily stock price for more than ten years of data starting with January 2007 is considered in this data example. The reason for being interested in estimating the volatility for this stock price is because it's highly dependent on the crude oil price.

ARCH Fit

ARCH Fit

```
## garch from tseries library
library(tseries)
## What order?
pacf(resids^2, ,main="Squared Residuals")
garch.fit = garch(resids, order = c(0, 8)
,trace=F)
summary(garch.fit)
## Evaluate goodness of fit
resids.fgarch = residuals(garch.fit)[-c(1:8)]
acf(resids.fgarch,main="ACF of ARCH Residuals")
acf(resids.fgarch^2,main="ACF of Squared ARCH Residuals")
```



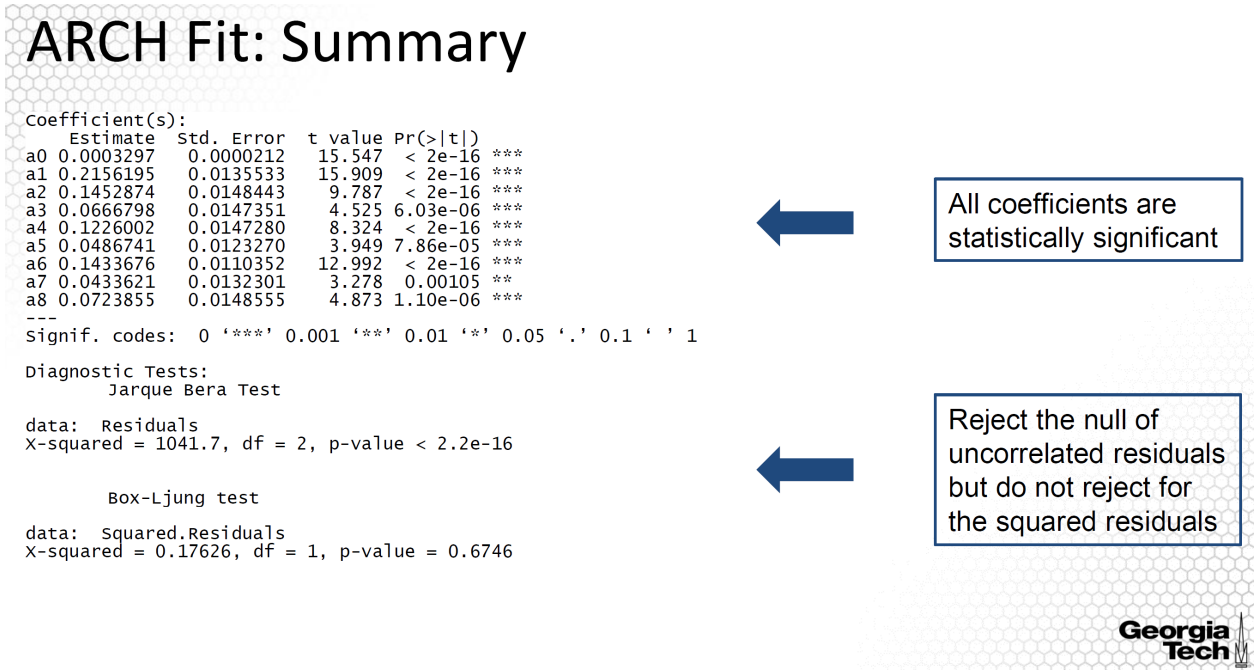
Georgia Tech

In the lecture for ARMA modeling, we learned that we can identify the order of an AR process using the PACF plot. Since ARCH is an AR process for the squared Zt process or the squared residual process, we can use the PACF plot to identify the order of the ARCH model. Thus, in this example I used the R command PACF of the squared residual process. The PACF plot is on the right. From this plot we see that the PACF is large for the first seven orders but decreases afterwards. Using this approach, we may select an order equal to seven as fitted in the command on this slide using the garch() R command. In this command, we input the process to be modeled, in this case the residual process, and the order for the ARCH model.

Please pay attention on how the ARCH order is specified in this R command. We'll see other implementations of ARCH and GARCH models where the order is specified differently.

Last, we evaluate the goodness of fit by first obtaining the residuals from the ARCH model. Note that I remove the first seven values from the residual vector since there will be NA's. This is because we fit an ARCH model of order seven. I plotted here the ACF of the residuals and of the squared residuals from the ARCH model to check whether the residuals are uncorrelated and whether there is an ARCH effect left in the residuals.

ARCH Fit: Summary

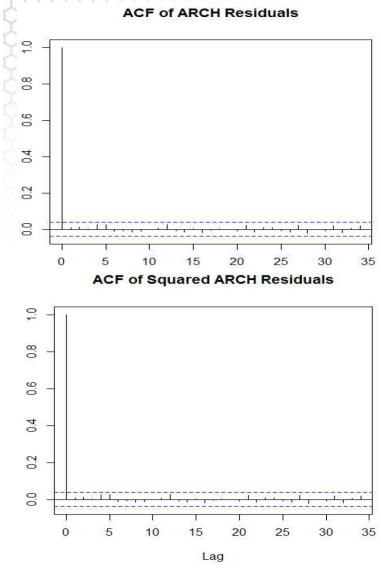


The output of the ARCH fit is here. It is similar to that of an AR fit. The output provides the estimated error coefficients, along with the inference on statistical significance of the AR coefficients.
According to this output, all coefficients are statistically significant hence, an order of seven or higher is appropriate. The output also provides the results from hypothesis testing procedures testing for uncorrelated residuals, and for an ARCH effect in the residuals.

According to these two tests, we reject the null hypothesis of uncorrelated residuals, but do not reject the null hypothesis of an ARCH effect in the residuals.

ARCH Fit: Residual Analysis

ARCH Fit: Residual Analysis



ACF of White Noise

ACF of White Noise

The ACF plots on the other hand look like those of white noise since the ACF values are within the confidence band except for the lag 0.

ARCH Fit: Different Implementations

ARCH Fit: Different Implementations

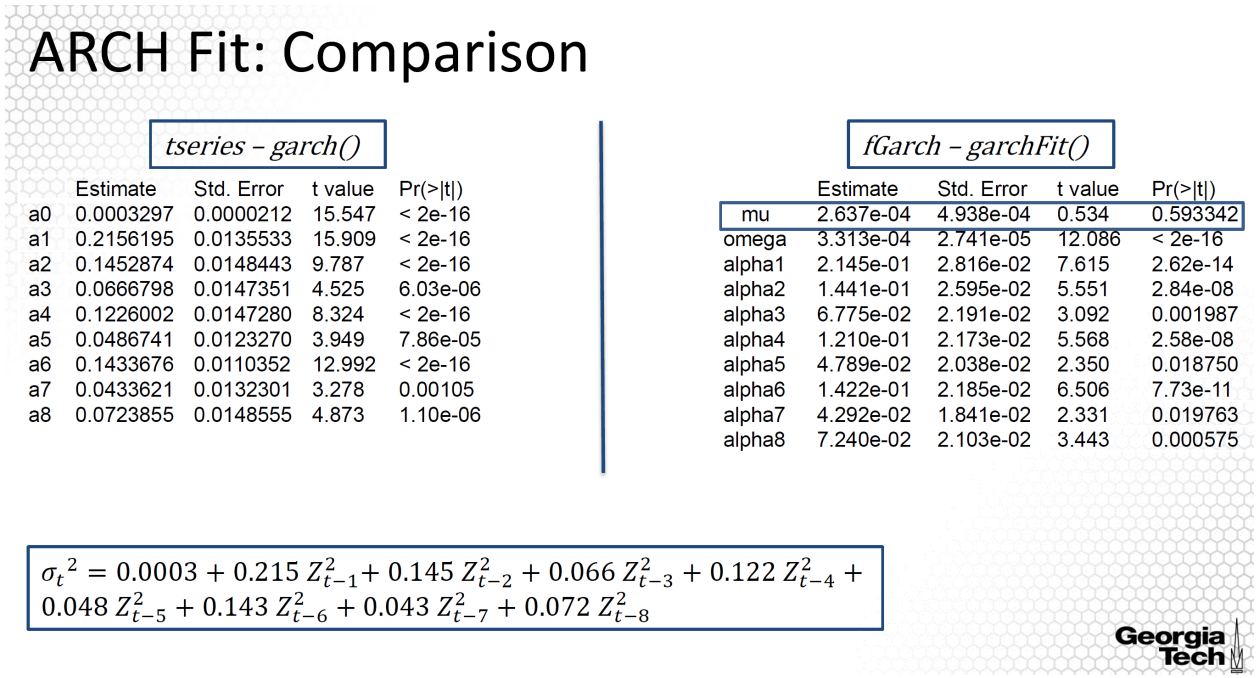
```
## garchFit from the fGarch library
library(fGarch)
## Fit ARCH on the ARMA residuals
archFit.resid = garchFit(~garch(8,0), data = resids, trace = FALSE)
summary(archFit.resid)
## Fit ARMA-ARCH
archFit.ts = garchFit(~ arma(4,4)+garch(8,0), data=pdcert[-1], trace = FALSE)
summary(archFit.ts)
```

I will note here that there are multiple R packages that can be used to fit ARCH models, and more generally GARCH models. One such implementation is using the Garch command from the 'tseries' package as demonstrated in the previous slides and a second implementation is the GarchFit() command from the fGarch package as demonstrated on this slide. The GARCH function from the tseries package is fast but does not always converge. The garchFit function from fGarch package is slower but does converge more consistently. In the implementation using the GarchFit command, we can fit the ARMA-ARCH model using the GarchFit function in two ways. One is to fit an ARMA process on a time series first, then model the residuals from the ARMA fit using an ARCH model. This is the first implementation on this slide. However, this approach does not provide efficient estimates for the ARMA-ARCH joint model, not as efficient as the one when we estimated the two models simultaneously.

We can also fit an ARCH model using the garchFit command by specifying the orders of the ARMA model through the option ARMA in this command.

The orders of the ARCH model are further specified by the Garch option. Note that for this implementation the first order is for the AR part of the model and that is specified as Garch (8,0) in this implementation.

ARCH Fit: Comparison

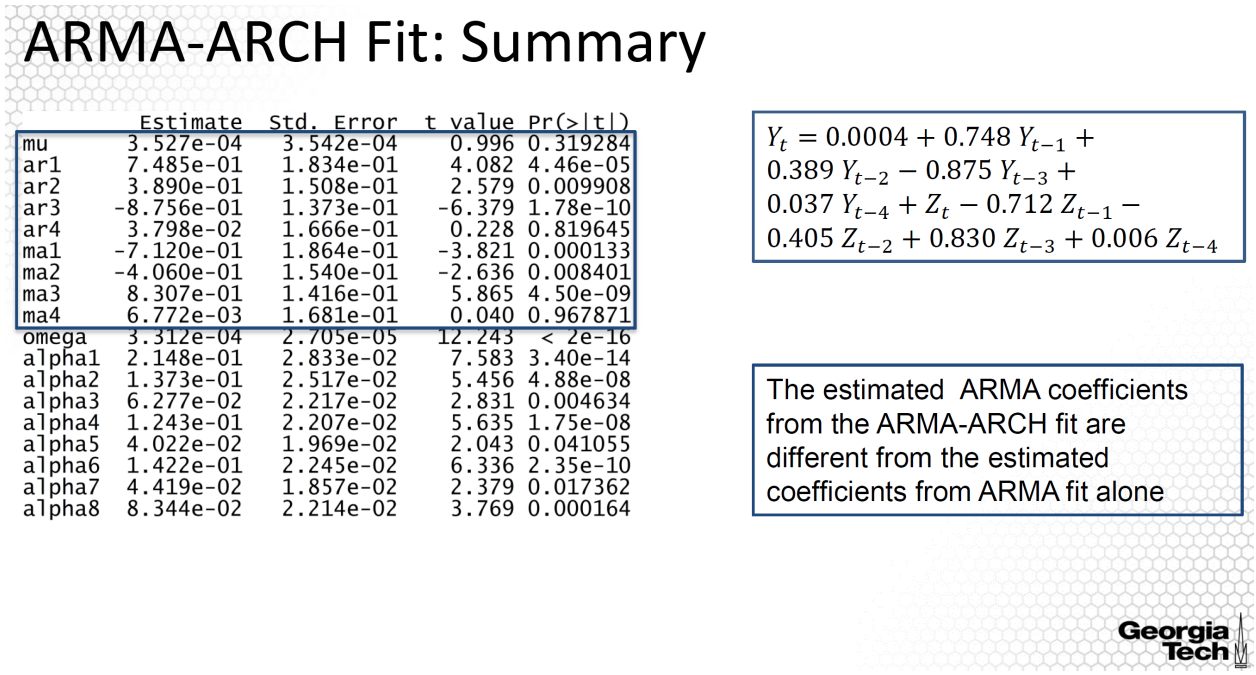


This slide compares the outputs using the two different implementations, specifically, using the garch() command on the left and the garchFit() command on the right applied to the residual process from the ARMA fit. The omega coefficient using garchFit corresponds to the a0 coefficient using garch(). The coefficients alpha1 to alpha8 correspond to the coefficients ar1 to ar8. The estimates from the two models are almost identical.

The difference between the output is that the garchFit function also provides the estimate for the mean of the fitted process, the mu parameter. In this case the estimate of the mean parameter mu is not statistically significant. This is not surprising since we applied the ARCH model on the residuals of an ARMA fit.

Based on this output the fitted ARCH model is provided here on this slide, specifically the formula of the conditional variance as a function of Zt-1 up to Zt-8 since we fit an ARCH(8) model.

ARMA-ARCH Fit: Summary



This is the output from the joint modeling of ARMA-ARCH applied to the time series of the return stock price using the garchFit command.

The first part of the output is for the ARMA model, specifically the estimated ARMA coefficients. The output provided for the ARMA-ARCH model fit is different from the output of the ARCH fit using the residual process.

ARMA-ARCH Fit: Summary

	Estimate	Std. Error	t value	Pr(>|t|)
mu	3.527e-04	3.542e-04	0.996	0.319284
ar1	7.485e-01	1.834e-01	4.082	4.46e-05
ar2	3.890e-01	1.508e-01	2.579	0.009908
ar3	-8.756e-01	1.373e-01	-6.379	1.78e-10
ar4	3.798e-02	1.666e-01	0.228	0.819645
ma1	-7.120e-01	1.864e-01	-3.821	0.000133
ma2	-4.060e-01	1.540e-01	-2.636	0.008401
ma3	8.307e-01	1.416e-01	5.865	4.50e-09
ma4	6.772e-03	1.681e-01	0.040	0.967871
omega	3.312e-04	2.705e-05	12.243	< 2e-16
alpha1	2.148e-01	2.833e-02	7.583	3.40e-14
alpha2	1.373e-01	2.517e-02	5.456	4.88e-08
alpha3	6.277e-02	2.217e-02	2.831	0.004634
alpha4	1.243e-01	2.207e-02	5.635	1.75e-08
alpha5	4.022e-02	1.969e-02	2.043	0.041055
alpha6	1.422e-01	2.245e-02	6.336	2.35e-10
alpha7	4.419e-02	1.857e-02	2.379	0.017362
alpha8	8.344e-02	2.214e-02	3.769	0.000164

$$Z_t = \sigma_t R_t$$

$$\sigma_t{}^2 = 0.0003 + 0.214\, Z_{t-1}^2 + 0.137\, Z_{t-2}^2 + 0.062\, Z_{t-3}^2 + 0.124\, Z_{t-4}^2 + 0.040\, Z_{t-5}^2 + 0.142\, Z_{t-6}^2 + 0.044\, Z_{t-7}^2 + 0.083\, Z_{t-8}^2$$

Georgia Tech

If we compare the estimates from the ARMA process that we initially estimated disregarding the ARCH portion of the model to the estimated coefficients using the joint ARMA-ARCH model, we find that the estimated ARMA coefficients are different. On the other hand, if we compare the ARCH fitted model using the ARMA-ARCH fit versus the ARCH model on the residuals, the fitted models are similar.

Summary:

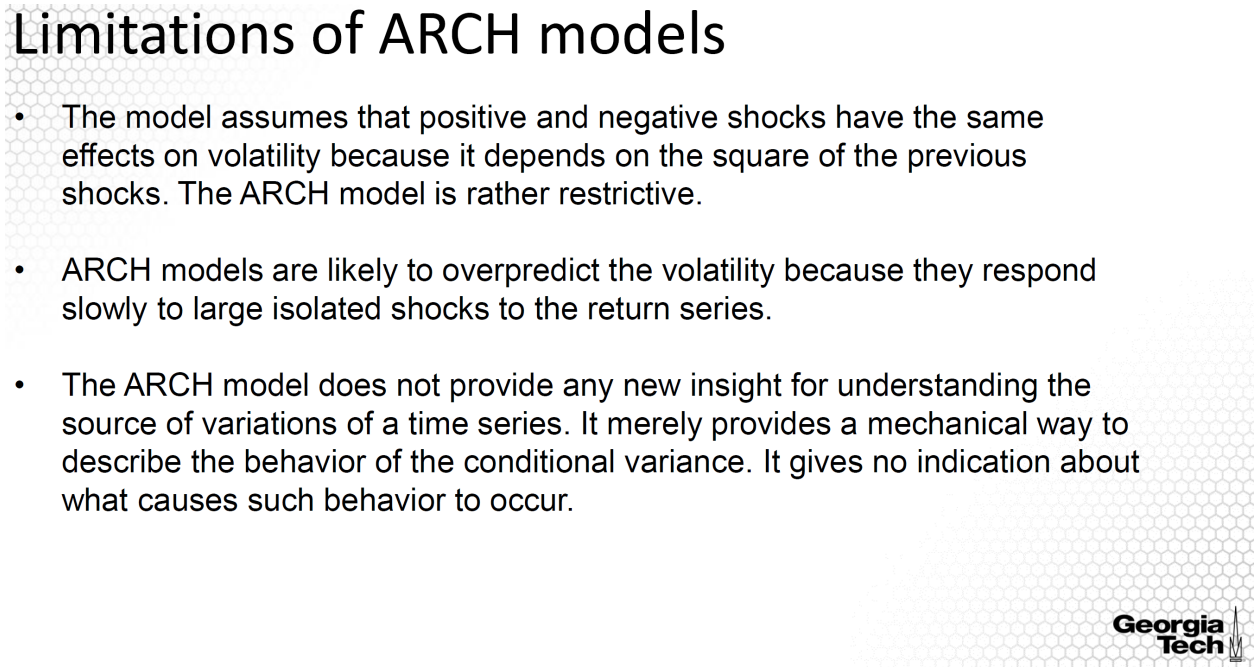
In summary, in this lesson I provided a data example to illustrate how to fit an ARCH model, with comparison across multiple implementations.

3.2: GARCH Models

3.2.1 GARCH Model

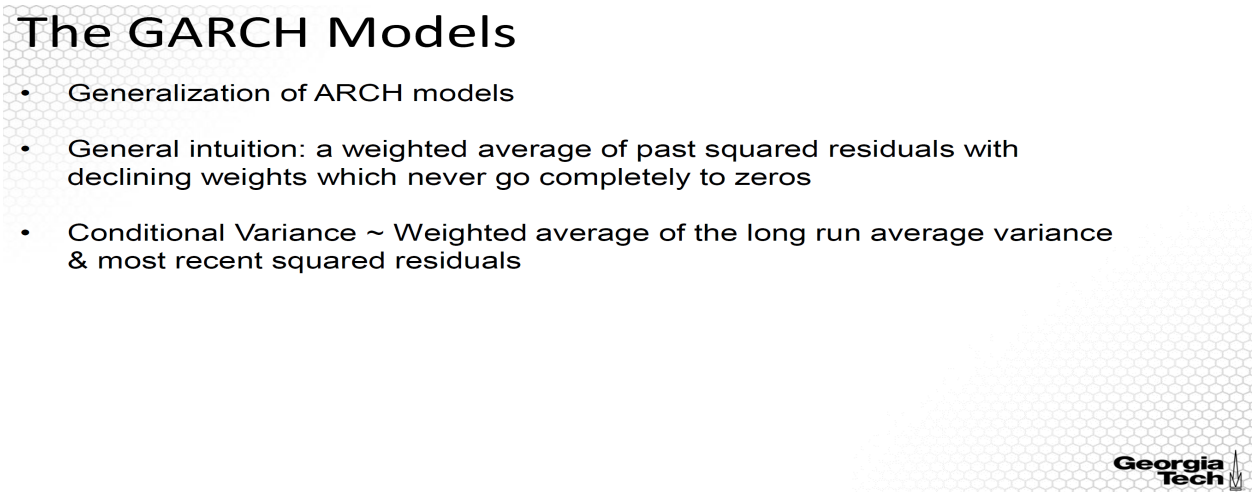
This lesson introduces a more general model for heteroskedasticity, an extension of the ARCH model, the so-called generalized ARCH model or GARCH.

Limitations of ARCH models



The ARCH model, while easy to understand and interpret, it has some limitations. The ARCH model is rather restrictive in that the model assumes that positive and negative shocks have the same effect on volatility because it depends on the square of the previous shocks. In practice, however, it is well known that many economic and financial indicators, or other measures with time dynamics respond differently to positive and negative shocks, particularly, because negative and positive shocks may be of different nature. For example, some negative shocks can be political or environmental, while some positive shocks can be driven by economic dynamics. Moreover, ARCH models are likely to overpredict the volatility because they respond slowly to large isolated shocks. When modeled with the ARCH model, large variance effects disappear quickly, certainly, no longer than the ARCH order. Given that the impact of high variance may last for a longer period of time, we would prefer a model in which the impact of large variance can be expressed for much longer than the order of the ARCH. Last, the ARCH model does not provide any new insight for understanding the source of variations of a time series. It provides a mechanical way to describe the behavior of the conditional variance. It gives no indication about what causes such behavior to happen.

The GARCH Model



Some of these limitations can be addressed with a more general heteroskedasticity models. One generalization is the GARCH model and extensions of the GARCH model. GARCH stands for Generalized autoregressive conditional heteroskedasticity or generalized ARCH. The intuition behind this model is that it is a weighted average of past squared residuals with declining weights which never go completely to zeros. Hence, it models the impact of large variance for a longer period of time than the specified orders. The most widely used GARCH specification asserts that the best predictor of the variance is a weighted average of the long run average variance and the new information for the current period consisting of the most recent squared residuals. Such an updating rule is a simple description of an adaptive or learning behavior.

The GARCH Model Formulation

The GARCH Model Formulation

The simplest of the GARCH specification is that of a $GARCH(1,1)$ model. This is defined as:

$$Y_t = \mu + Z_t, \qquad Z_t|F_{t-1} \sim N(0, \sigma_t^2)$$
$$\sigma_t^2 = \gamma_0 + \gamma_1 Z_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

More generically, there are $GARCH(m,n)$ specifications where σ_t^2 is:

$$\sigma_t^2 = \gamma_0 + \gamma_1 Z_{t-1}^2 + \ldots + \gamma_m Z_{t-m}^2 + \beta_1 \sigma_{t-1}^2 + \ldots + \beta_n \sigma_{t-n}^2$$

Reference: Timothy Bollerslev. "Generalized autoregressive conditional heteroskedasticity". Journal of Econometrics, 31, pages 307:327 (1986).

Georgia Tech

For ease of understanding, I'm presenting on this slide the simplest GARCH model, the GARCH(1,1) model, which often is sufficient to model heteroskedasticity. The conditional variance sigma_t squared is modeled using the lag squared noise process, Zt-1 squared, similar to the ARCH model, but it also includes additional lagged relationship with past variability through sigma_t-1 squared. This additional term allows for modeling the impact of the variance for longer than an order one ARCH model. More generally, a GARCH(m, n), with an AR order m and MA order n, is when the conditional variance depends linearly on the lagged squared noise process through Zt-1 squared, Zt-2 squared up to Zt-m squared, but also through the past variability sigma t-1 squared up to sigma t-n squared. Thus the interpretation of the two orders is as follows: m refers to how many autoregressive lags or ARCH terms appear in the equation while n refers to how many moving average lags are specified. This model was introduced by Bollerslev in 1986.

GARCH vs ARMA

GARCH vs ARMA

GARCH is an ARMA form of heteroscedasticity

$$\sigma_t^2 = \gamma_0 + \gamma_1 Z_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

However, this is only the deterministic part of volatility. Actual volatility Z_t^2 cannot be observed fully, and is therefore defined as:

$$Z_t^2 = \sigma_t^2 + \omega_t.$$

Then, the above equation in σ_t^2 becomes:

$$Z_t^2 = \gamma_0 + \gamma_1 Z_{t-1}^2 + \beta_1 (Z_{t-1}^2 - \omega_{t-1}) + \omega_t$$
$$= \gamma_0 + (\gamma_1 + \beta_1) Z_{t-1}^2 + \omega_t - \beta_1 \omega_{t-1}$$
$$(1 - (\gamma_1 + \beta_1)B) Z_t^2 = \gamma_0 + (1 - \beta_1 B)\omega_t$$

This is an $ARMA(1,1)$ model in the AR component of Z_t^2 and the MA component of ω_t.

Georgia Tech

In the description of the GARCH model, I referred to m and n as the AR and MA orders, respectively. But is there a direct relationship with the ARMA model other than simply the interpretation? I will describe here that the GARCH model is practically an ARMA model for the squared residuals. Specifically, if Zt is modeled by a GARCH model, then Zt squared is an ARMA process with the exception that the residual process of the corresponding model is not white noise. The volatility of the Zt squared cannot be observed fully since it depends on the lagged Zt squared time series. In the GARCH(1,1) model, it depends on Z t-1 squared.

Starting with this formula, we can replace sigma t-1 squared with Zt-1 squared minus sigma t-1 squared. Separating the terms in Zt-1 and omega, we have an ARMA(1, 1) process. Adding together the terms with Zt-1 squared leads then to an ARMA model in Zt-1 squared where the AR coefficient is gamma_1+beta_1, and the MA coefficient is -beta 1. This is an $ARMA(1, 1)$ model with the AR component of Z_t^2 and the MA component of ω_t.

Stationarity of GARCH(1,1)

Stationarity of GARCH(1,1)

The stationarity conditions are satisfied:

- $\mathrm{E}[Y_t] = \mu$
- $\mathbb{V}(Y_t) = \mathrm{E}[Z_t^2]$ computed as below

  $\mathrm{E}[Z_t^2] = \gamma_0 + (\gamma_1 + \beta_1)\mathrm{E}[Z_{t-1}^2] + \mathrm{E}[\omega_t - \beta_1\omega_{t-1}]$

  $(1 - \gamma_1 - \beta_1)\mathrm{E}[Z_t^2] = \gamma_0$

  $\sigma^2 = \gamma_0/(1 - \gamma_1 - \beta_1)$

The stationarity conditions are that $(\gamma_1 + \beta_1) < 1$ and $0 \leq \gamma_1, \beta_1 \leq 1$.

- $Cov(Y_t, Y_{t-1}) = \mathrm{E}[Z_t Z_{t-1}] = 0.$

Georgia Tech

Let's return to the concept of stationarity. Can a GARCH model be stationary? Recall that a GARCH model is for the conditional variance hence it is possible to have a white noise process or more generally a stationary process with conditional variance varying with time. The stationarity is evaluated in the unconditional first and second moments of the process as I will illustrate with a simple example. In this example, Yt is the sum between the mean, mu, and the GARCH process of orders 1 and 1. Yt is stationary if its expectation, mu is constant over time, the variance of Yt, which is equal to the expectation of the Zt squared, is also constant over time. The last condition is that the covariance between Yt and Yt-1 is 0. That is, Yt and Yt-1 are uncorrelated. Note that this does not mean they are independent.

The only condition that varies with time involves the variance of the process since it depends on the expectation of past Z_t, specifically, squared Z_t-1 for the GARCH(1,1) process. From the derivation involving the variance of the process, we derive a condition in the marginal or unconditional variance of Yt and the GARCH coefficients, gamma 0, gamma 1 and beta 1. Specifically, the unconditional variance is equal to gamma 0 divided by 1- gamma 1- beta 1.

For this variance to be finite and positive as required for stationarity Yt, we need to have that 1- gamma 1- beta 1 to be positive or the sum of the two coefficients to be smaller than 1. We also need to restrict that the two coefficients are positive. These conditions ensure that the variance of the the process Y_t is constant and finite, which in turn it ensures that the process is weakly stationary. Last, I'll point out that this condition of stationarity will be different for other GARCH model.

Characteristics of GARCH(1,1)

Characteristics of GARCH(1,1)

1. Weak stationarity under $(\gamma_1 + \beta_1) < 1$ and $0 \leq \gamma_1, \beta_1 \leq 1$
2. Model volatility/variability clusters
3. It has heavy tails: if $1 - 2\beta_1^2 - (\gamma_1 + \beta_1)^2 > 0$ then the corresponding kurtosis is

$$\frac{\mathrm{E}[Z_t^4]}{\left[\mathrm{E}[Z_t^2]\right]^2} = \frac{3(1 - (\gamma_1 + \beta_1)^2)}{1 - 2\beta_1^2 - (\gamma_1 + \beta_1)^2} > 3$$

Georgia Tech

These are some important characteristics of a GARCH(1,1) model. The first condition was derived in the previous slide, and it guarantees stationarity. The second characteristic is that GARCH models capture volatility that comes in clusters. That is, periods of high volatility, followed by periods of low volatility. Last, the GARCH model has heavy tails measured by the kurtosis, that is, the kurtosis is larger than 3. A heavy tail distribution means that GARCH can capture outlying periods of large volatility.

Examples of Joint GARCH models

Examples of Joint GARCH models

- AR-GARCH models: for example, AR(1)-GARCH(1,1) is defined by the set of equations below

$$Y_t = \mu + \phi Y_{t-1} + Z_t$$
$$\sigma_t^2 = \gamma_0 + \gamma_1 Z_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

- ARMA-GARCH models: for example, ARMA(1,1)-GARCH(1,2) is defined by the set of equations below

$$Y_t = \mu + \phi Y_{t-1} + Z_t + \theta Z_{t-1}$$
$$\sigma_t^2 = \gamma_0 + \gamma_1 Z_{t-1}^2 + \beta_1 \sigma_{t-1}^2 + \beta_2 \sigma_{t-2}^2$$

Georgia Tech

We can expand the GARCH model by considering ARMA-GARCH joint models. For an AR(1)-GARCH(1,1) model, the model equations are on the slide. Yt is modeled by an AR(1) model with AR coefficient phi whereas the noise, Zt, is modeled by a GARCH(1,1) model which means that the variance of Zt, sigma t squared, follows the formula of a GARCH(1,1) model. Another example provided on this slide is the ARMA (1,1) with GARCH (1,2), joint model for illustration of such joint models.

MLE for GARCH models

MLE for GARCH models

The parameter vector for a GARCH(m, n) model is $\Gamma=\{\gamma_0, \gamma_1, \ldots, \gamma_m, \beta_1, \ldots, \beta_n\}$.
Assumptions Under normality assumptions, the conditional likelihood the GARCH model becomes:

$$f(y_t|y_{t-1}, \ldots; \Gamma) = \frac{1}{\sqrt{2\pi\sigma_t^2}} exp\left[-\frac{1}{2}\left(\frac{y_t^2}{\sigma_t^2}\right)\right]$$

$$= \frac{1}{\sqrt{2\pi(\gamma_0 + \gamma_1 y_{t-1}^2 + \gamma_2 y_{t-2}^2 + \ldots + \beta_1 \sigma_{t-1}^2 + \ldots)}}$$

$$exp\left[-\frac{1}{2}\left(\frac{y_t^2}{\gamma_0 + \gamma_1 y_{t-1}^2 + \gamma_2 y_{t-2}^2 + \ldots + \beta_1 \sigma_{t-1}^2 + \ldots}\right)\right]$$

and the unconditional distribution is a combination of the conditional density and the joint density of $Y_{\max(m,n)}, \ldots, Y_1$.

Georgia Tech

The estimation approach of the parameters of GARCH models is the maximum likelihood estimation or in short MLE. The parameters to be estimated are the coefficients gamma_0, gamma_1, up to gamma_m and the coefficients beta_1, up to beta_n, for a GARCH model of orders m and n. If we assume normality, the conditional likelihood function of Yt given the past history up to time t, can be expressed as a normal distribution with mean 0 if the mean of Yt is 0 and variance sigma_t squared. The probability density function is as on this slide. If we replace sigma_t squared, the conditional distribution is now a function of the GARCH coefficients. As we discussed in the first lesson of the first lecture of this course, we commonly estimate such models by using the conditional likelihood, which will be the product of such conditional distributions. As you would expect, maximizing the resulting likelihood function is very complicated, requiring numerical algorithms. This is why when we estimate GARCH models in R, it can take a minute or two to get the fitted model. There are also different numeric algorithms that can be used. Some can be slower, but reaching convergence, some are faster but not guaranteeing convergence.

Summary: To summarize, this lesson has provided an introduction of the generalized ARCH model, including insights on the interpretation of the model and reviewing one simple GARCH model, the GARCH with order 1 and 1. In the next lesson, I will illustrate this model with a data example.

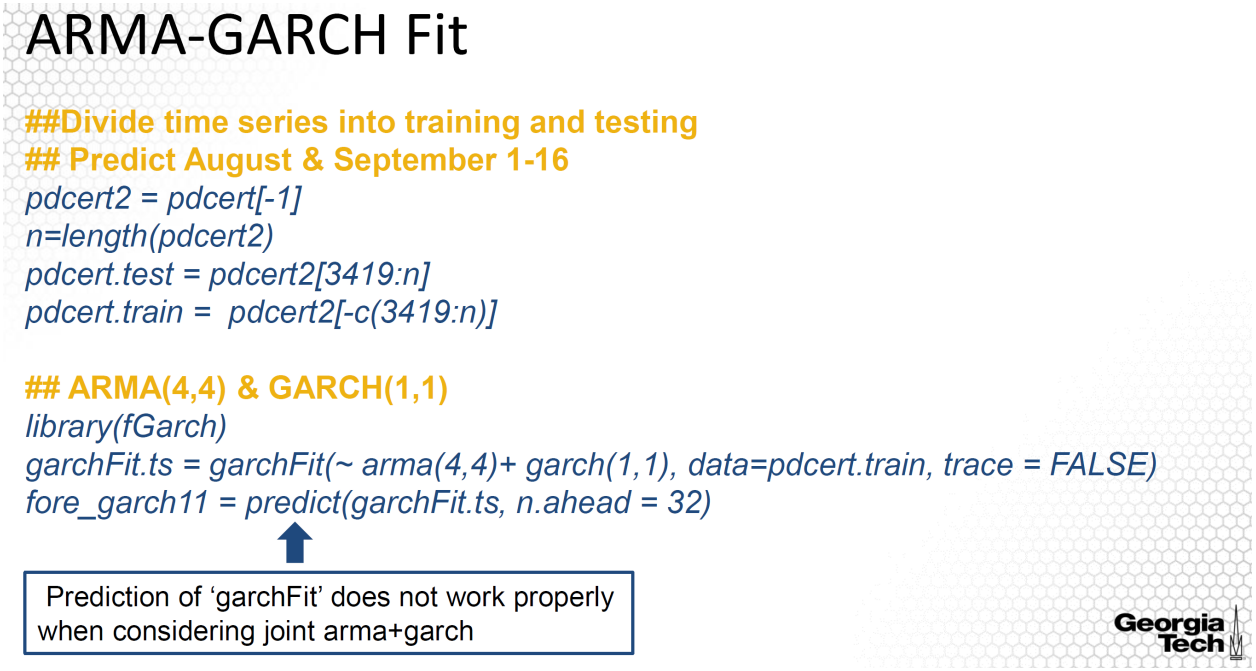
3.2.2 GARCH Model: Data Examples

This lesson provides a data example for illustrating the implementation of the GARCH model along with interpretation and prediction.

PDC Energy, Inc (PDCE)

We'll return to the data example in which we model the stock price of PDC Energy, a company that is a producer of crude oil and natural gas in the United States. Daily stock prices for more than 13 years of data starting with January 2007 are considered in this data example.

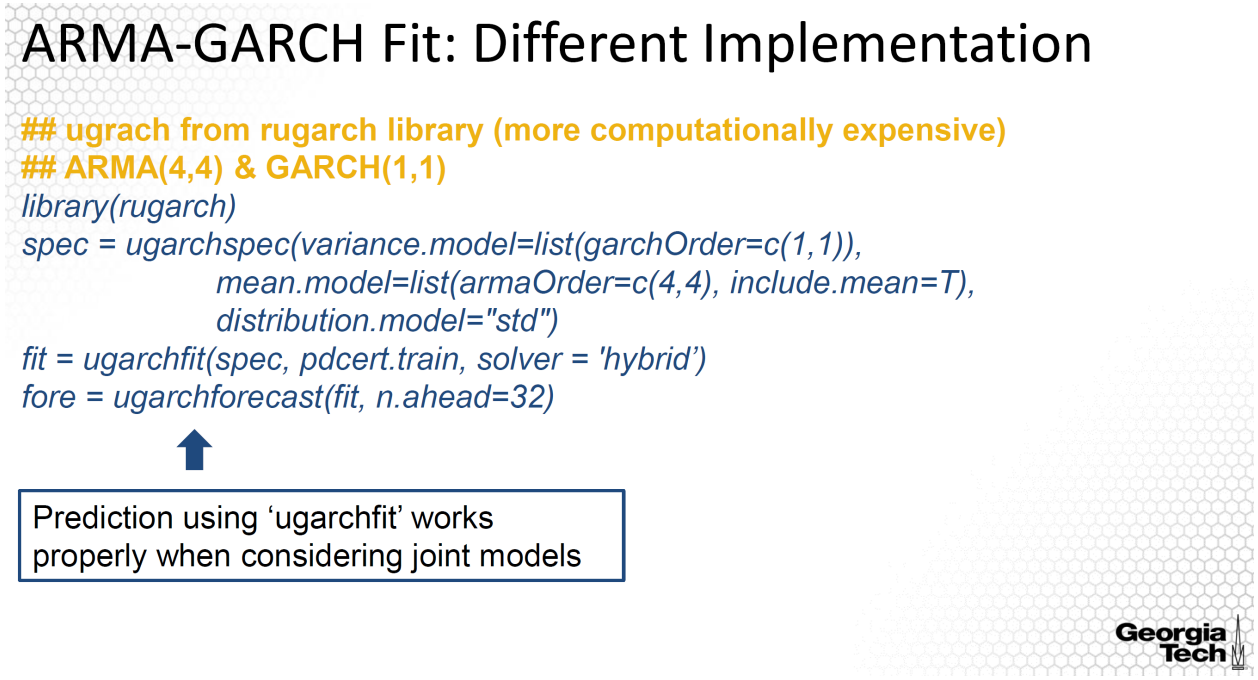
ARMA-GARCH Fit



We will begin with a simple implementation of an ARMA-GARCH joint model. I will apply a similar R command as we did for fitting the ARCH model, specifically the garchFit() command. This command allows fitting a joint ARMA-GARCH model as specified in the command on the slide. In this example, I used the same orders for the ARMA model that we fit previously on the log return for PDCE stock price, specifically, orders 4 and 4. Again this will model the conditional mean only. I also used a GARCH(1,1) to model the condition variance.

While the model fit using this implementation works fine, this R command will often fail to provide predictions when we have an ARMA-GARCH joint model.

ARMA-GARCH Fit: Different Implementation



Since prediction is an important aspect in modeling time series, I will introduce yet another implementation, which is a bit more challenging to implement and it is more computationally expensive, but overall has better performance since it generally converges for any combination of orders. In this implementation, I first define the model's specifications through the *ugarchspec()* R command. The model's specifications are similar to those from the model fitted in the previous slide, particularly, the ARMA orders are 4 and 4, the GARCH orders are 1 and 1. We consider the mean to be nonzero, through the option include.mean=T and we assume the t-distribution instead of normal distribution through the specification of the option *distribution.model*. Then the model is fitted using the ugarchfit with the formulation specified by the specifications provided in the *ugarchspec*. All R commands used in this slide are available in *rugarch* library. This library is the most recent implementation of the ARMA-GARCH model. It provides many more modeling options for modeling the conditional mean and the conditional variance jointly, as I will show in the rest of the lessons of this module.

Last, the predictions for both the conditional mean and the conditional variance work for this implementation as I will illustrate later in this lesson.

ARMA+GARCH Order Selection

ARMA+GARCH Order Selection

ARMA & GARCH – simultaneous fit and model selection for efficient estimators of the model parameters.

⇒ ARMA(p,q)+GARCH(m,n) – computationally expensive to search over all combinations of (p,q)x(m,n) orders, hence apply a heuristic algorithm

- Step 1: Apply ARMA to model the time series and select orders ⇒ selected initial orders (p_0, q_0)
- Step 2: Apply ARMA(p_0, q_0)+GARCH(m,n) with varying m & n orders (consider only small values) ⇒ selected initial orders (m_0, n_0)
- Step 3: Apply ARMA(p,q)+GARCH(m_0, n_0) with varying p & q orders ⇒ selected initial orders (p_1, q_1)
- Step 4: Apply ARMA(p_1, q_1)+GARCH(m,n) with varying m & n orders (consider only small values) ⇒ selected initial orders (m_1, n_1)

Georgia Tech

So far, we fit an ARMA model first to obtain the orders, then we considered the GARCH model applied to the residuals. However, when fitting ARMA-GARCH jointly, we should determine both the ARMA and the GARCH orders jointly. If the process is indeed well approximated by an ARMA-GARCH model, then considering the ARMA and its order selection while neglecting the GARCH for modeling the conditional variance, as we did in the first implementation, will lead to inefficient estimators. Similarly, when considering the fit and order selection of the conditional variance model, we should not neglect the model for the conditional mean. This is because neither the conditional mean model nor the conditional variance model can be estimated consistently if taken separately. Joint estimation will typically be more efficient, and this is why it is preferred. Unfortunately, the task of jointly determining the ARMA and GARCH orders is computationally challenging. It will require fitting a very large number of models, where each model requires extensive computational time. For example, if we assume orders from 0 to 5 for both orders of the ARMA model for the conditional mean, and orders from 0 to 3 for both orders of the GARCH for the conditional variance, we would have to fit 576 models. If each model requires 1 minute per run, this means about 10 hours of computational time. Alternatively, we can select the orders for ARMA and GARCH iteratively.

Specifically, at Step 1, we apply ARMA to model the time series and select orders resulting in initial orders p0 and q0 for the ARMA model of the conditional mean.

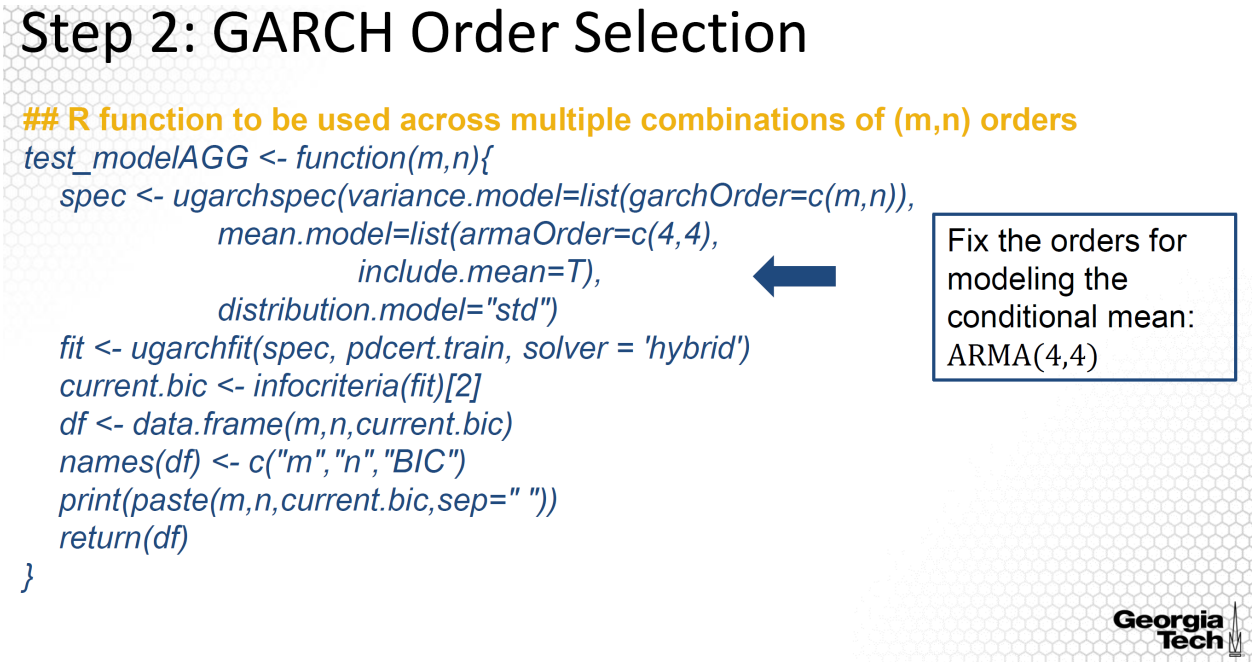
At Step 2,apply an ARMA-GARCH model with the ARMA orders set fixed to those selected in Step 1, and select the orders of the GARCH model only, giving us the initial GARCH orders m0 and n0.

At Step 3, we update the orders of ARMA by fixing the GARCH orders to the initial values m0 and no; we thus vary the orders p and q, giving us the updated orders p1 and q1.

Finally, at Step 4, we update the orders of GARCH by fixing the ARMA orders to the updated values p1 and q1; we thus vary the orders m and n, giving us the updated orders m1 and n1.

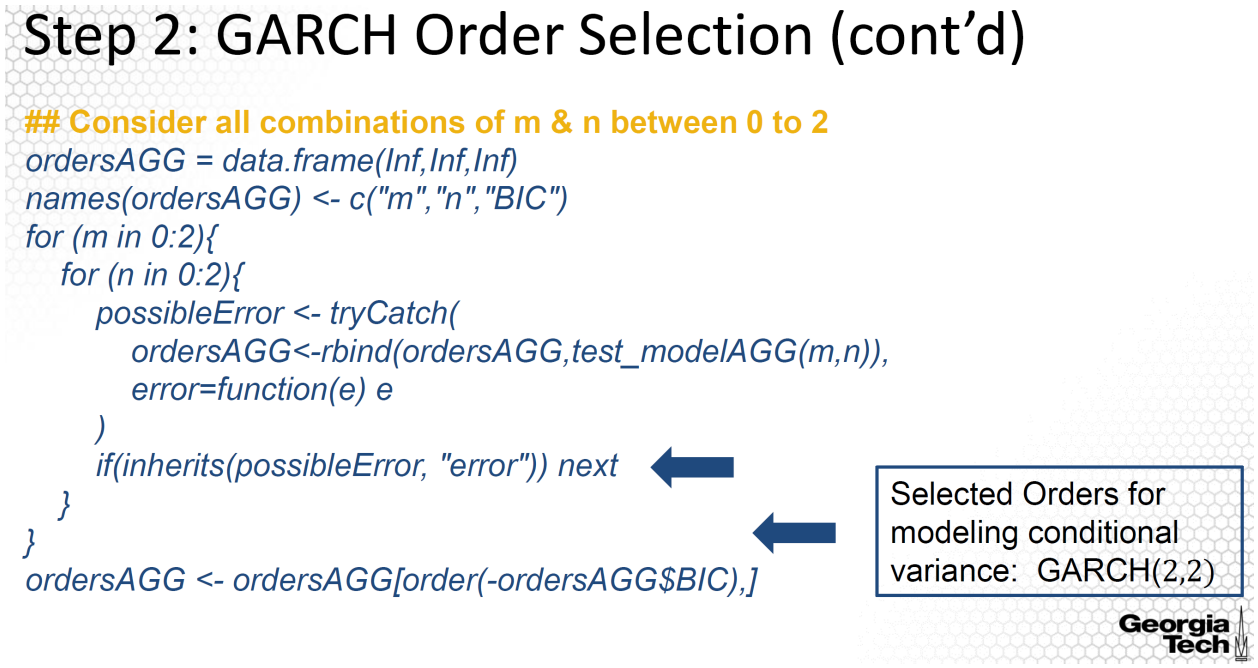
We can repeat the updating of the ARMA and GARCH orders by adding another round of order selection for the two models, but generally the model selected with one update of the orders is a good enough model in terms of fit and complexity.

Step 2: GARCH Order Selection



Here, we consider modeling the return price using a joint ARMA-GARCH model. At Step 2 of order selection, we fix the ARMA orders to be the selected orders (4,4) using the AIC approach from the previous lesson, then we'll select the orders of the GARCH model. This R function fits the ARMA-GARCH model with ARMA(4,4) and GARCH(m,n) with m and n as the input. The function outputs the BIC value for the input m and n orders. Note that here I am using BIC instead of AIC to select the orders. This is because BIC prefers smaller, less complex models. Selecting lesser complex models here is important because we fit an ARMA-GARCH joint model which is already very complex.

Step 2: GARCH Order Selection (cont'd)



Then I used this function to apply it for a range of m and n values taking values 0, 1, and 2 hence a maximum order of 2. To select the order, we loop over all combinations of orders m and n. Note that we consider somewhat smaller orders for GARCH than we used for ARMA modeling, since a small GARCH model is commonly sufficient in modeling the volatility or conditional variance. Note that it is still possible that the model fit not to converge for some combination of orders. We thus skip these models as provided in the code.

In this example, the selected GARCH orders are m equal to 2 and n equal to 2.

Step 3: ARMA Order Selection Update

Step 3: ARMA Order Selection Update

```
## R function to be used across multiple combinations of (p,q) orders
test_modelAGA <- function(p,q){
   spec = ugarchspec(variance.model=list(garchOrder=c(2,2)),
                     mean.model=list(armaOrder=c(p,q),
                                     include.mean=T),
                     distribution.model="std")
   fit = ugarchfit(spec, pdcert.train, solver = 'hybrid')
   current.bic = infocriteria(fit)[2]
   df = data.frame(p,q,current.bic)
   names(df) <- c("p","q","BIC")
   print(paste(p,q,current.bic,sep=" "))
   return(df)
}
```

Fix the orders for modeling the conditional variance: GARCH(2,2)

Georgia Tech

At the next step, we apply a similar code as for Step 1, except that now we fix the GARCH orders to the orders selected in Step 2, specifically the GARCH orders are m equal to 2 and n equal to 2. Then perform the fit given the p & q orders of ARMA.

Step 3: ARMA Order Selection Update (cont'd)

Step 3: ARMA Order Selection Update (cont'd)

```
## Update the ARMA order
ordersAGA = data.frame(Inf,Inf,Inf)
names(ordersAGA) <- c("p","q","BIC")
for (p in 0:4){
   for (q in 0:4){
      possibleError <- tryCatch(
         ordersAGA<-rbind(ordersAGA,test_modelAGA(p,q)),
         error=function(e) e
      )
      if(inherits(possibleError, "error")) next
   }
}
ordersAGA <- ordersAGA[order(-ordersAGA$BIC),]
tail(ordersAGA)
```

Selected Orders for modeling the conditional mean: ARMA(0,0)

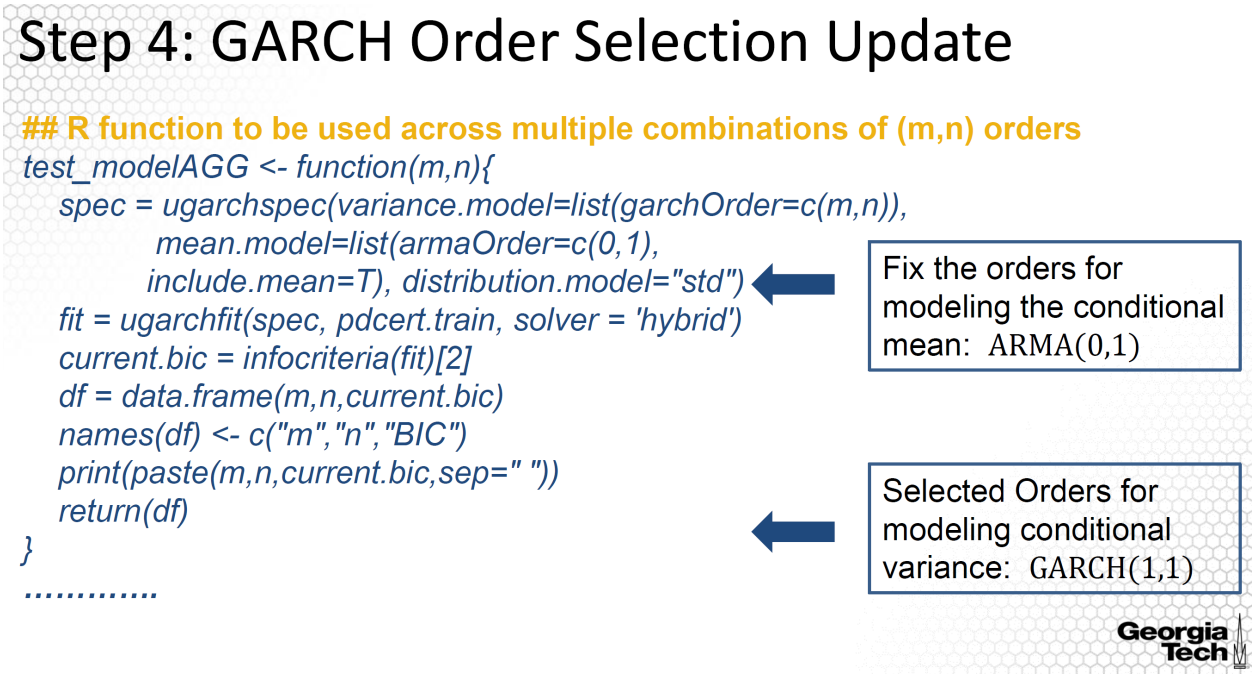
Choose: AR=0, MA=1 instead

Georgia Tech

Next we loop over all combinations of p & q orders for the ARMA model of the conditional mean and apply the R function provided in the previous slide. Similar to Step 2, it is possible that the model fit not to converge for some combination of orders. We thus skip these models as provided in the code.

The selected orders for modeling the conditional mean based on this approach are AR order or p = 0 and MA order or q=0. Since this is the trivial model, we choose: AR=0, MA=1 instead.

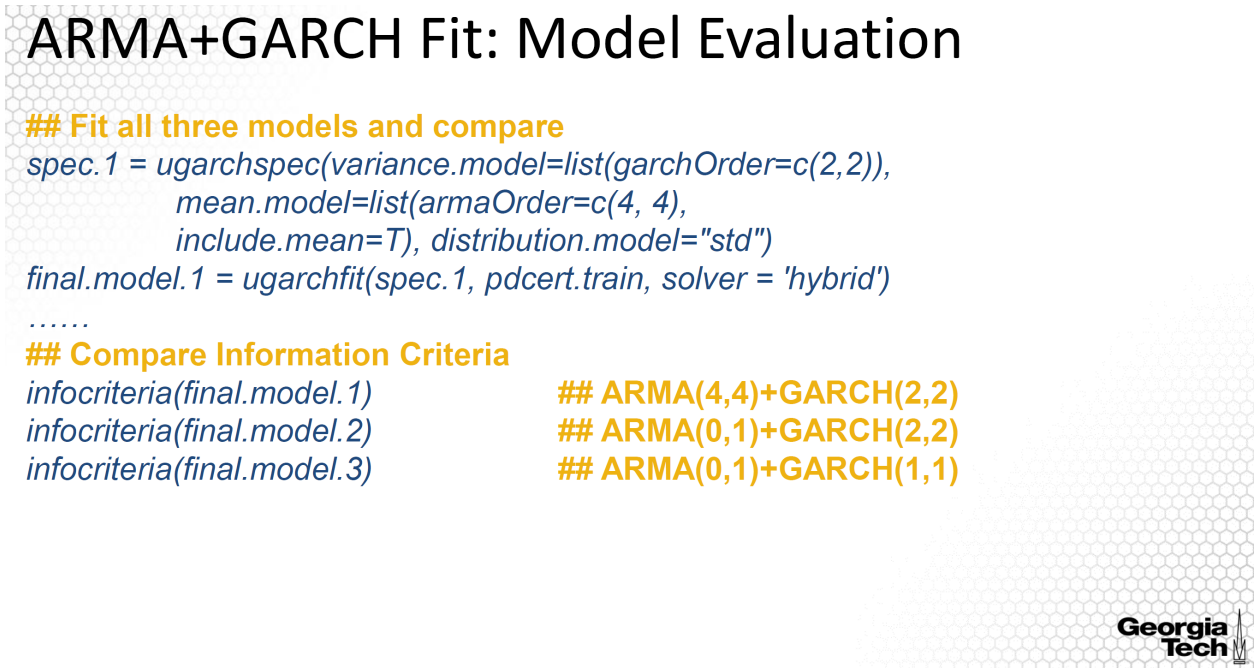
Step 4: GARCH Order Selection Update



This step is similar to Step 2 except that now we use the updated ARMA model orders as provided in Step 3.

The selected orders for modeling the conditional variance are GARCH(1,1).

ARMA+GARCH Fit: Model Evaluation



Next we will evaluate three models, with the selected orders as given at the steps provided above. From the order selection analysis provided in the previous steps, we saw that as we refined or updated the orders, with the selected models in the later steps being more parsimonious; for example, the first model after Step 2 is an ARMA(4,4)-GARCH(2.2). The second model after performing Step 3 is ARMA(0,1)-GARCH(2.2). The third model after Step 4 is ARMA(0,1)-GARCH(1.1). A parsimonious model is one that does not have a large number of parameters to estimate, that is one with small orders for the ARMA-GARCH models. Here we are applying the infocriteria() command in R to compare the three models based on multiple model evaluation criteria, including AIC and BIC.

ARMA+GARCH Fit: Model Evaluation (cont'd)

ARMA+GARCH Fit: Model Evaluation (cont'd)

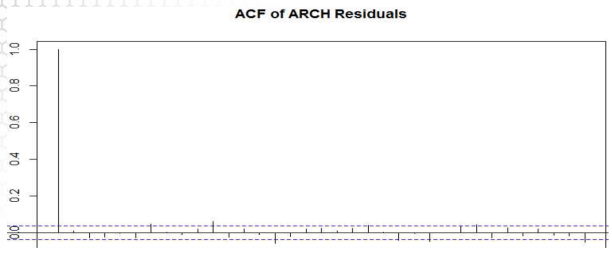
```
> ## compare Information Criteria
> infocri teria(final.model.1)

Akaike          -4.060253
Bayes           -4.033321
Shibata         -4.060291
Hannan-Quinn    -4.050629
> infocriteria(final.model.2)

Akaike          -4.055557
Bayes           -4.041193
Shibata         -4.055568
Hannan-Quinn    -4.050424
> infocriteria(final.model.3)

Akaike          -4.056774
Bayes           -4.046001
Shibata         -4.056780
Hannan-Quinn    -4.052925
```

All models perform similarly: choose least complex model

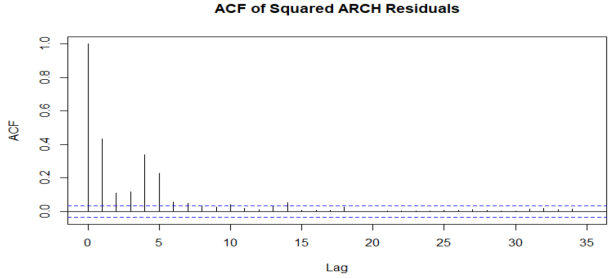
The output is here. The first two values are for the Akaike information criteria, or AIC, and for the Bayes information criteria, or BIC. In this example the values across the four criteria are very similar for the three models. Since we prefer less complex models, thus we would chose the last model, where we have an ARMA with orders 0, 1 and a GARCH with orders 1 and 1.

Residual Analysis

Residual Analysis



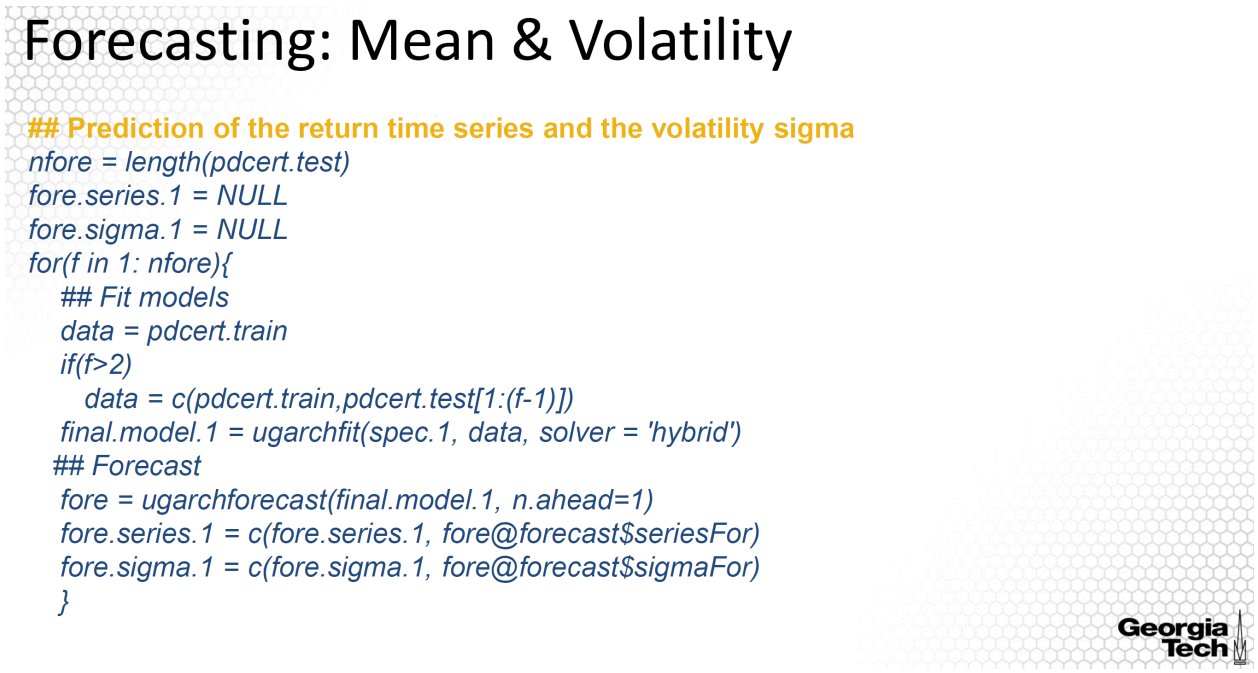
White Noise



Not White Noise

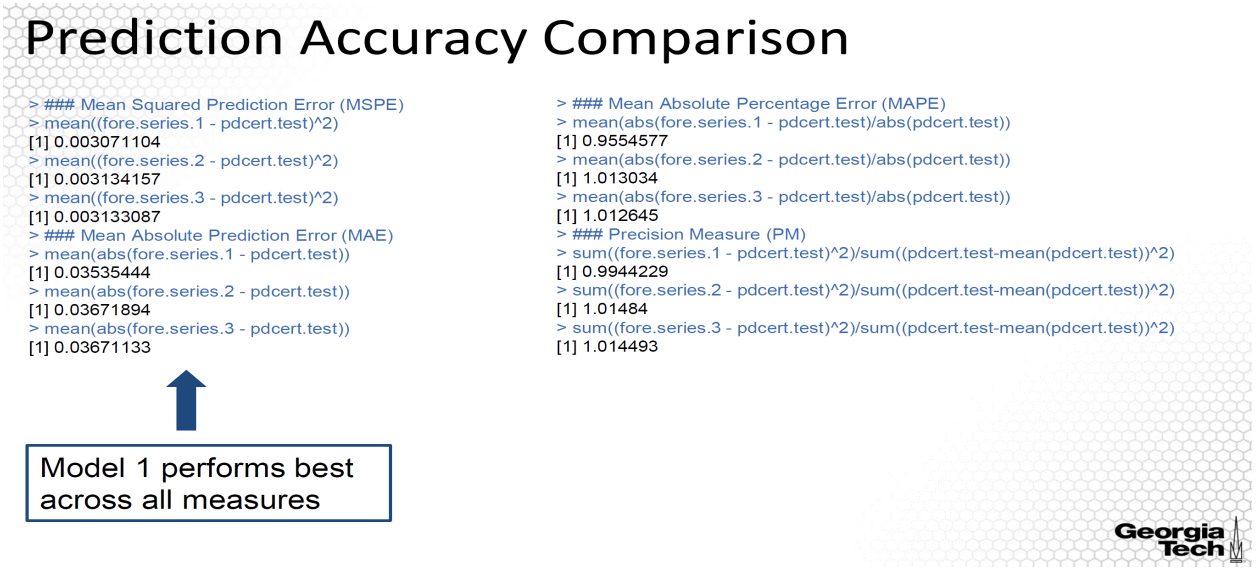
These are the ACF plots for the residuals and the squared residuals of the selected (third) model. Based on these plots, the ACF of the residuals resembles the ACF of white noise. However, this is not the case for the squared residuals suggesting that a higher order GARCH model may be considered.

Forecasting: Mean & Volatility



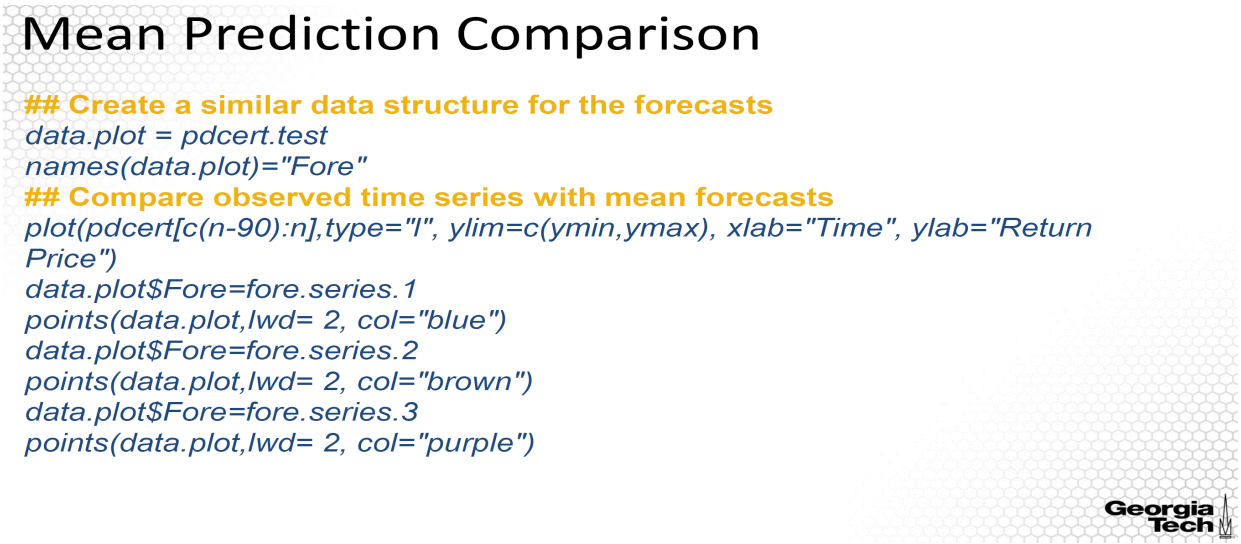
The R code on this slide is for obtaining the predictions for the conditional mean and for the conditional variance or so-called volatility. The prediction is performed one lag ahead on a rolling basis. The code on the slide is for the first model, but I applied the code to obtain the predictions using the other two models also. The output consists of forecasts for the return price for the conditional mean and the conditional variance.

Prediction Accuracy Comparison



These are the prediction accuracy measures for the conditional mean based on the three models. The three models perform similarly across all prediction measures, however, model 1 performs best across all measures. The precision measure is close to 1, meaning that the variability in the predictions is similar to the variability in the observed data over the prediction period.

Mean Prediction Comparison



This is the R code for comparing the predictions to the observed data using a visual display.

From the resulting plot, we see that the predictions for the conditional mean are close to the zero line, not capturing the variations we see in the observed data. This might be because in

this example the volatility predominates over the conditional mean. That is, the variations we see are not due to the conditional mean but due to the variability.

Variance Prediction Comparison

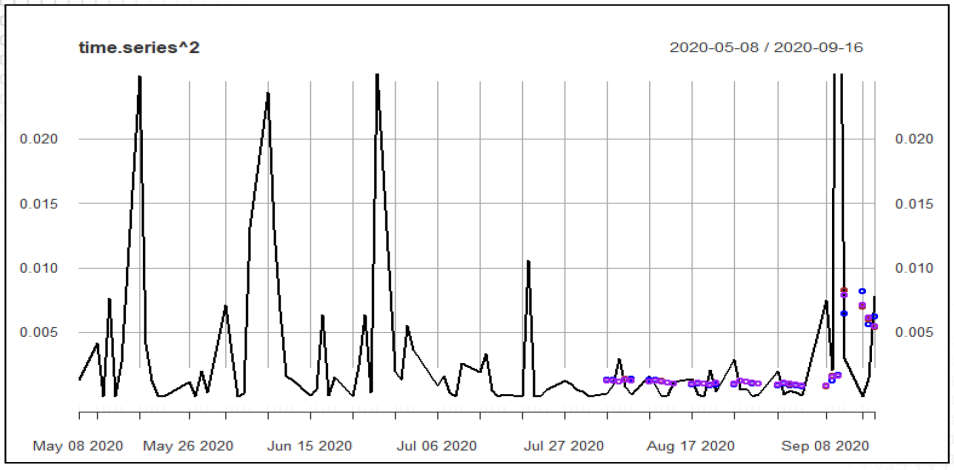
Variance Prediction Comparison

```
## Compare squared observed time series with variance forecasts
plot(time.series^2,type="l", ylim=c(ymin,ymax), xlab="Time", ylab="Return Price")
data.plot$Fore=fore.sigma.1^2
points(data.plot,lwd= 2, col="blue")
data.plot$Fore=fore.sigma.2^2
points(data.plot,lwd= 2, col="brown")
data.plot$Fore=fore.sigma.3^2
points(data.plot,lwd= 2, col="purple")
```

Georgia Tech

In this code, I'm contrasting the squared time series with the predictions for the conditional variance.

Variance Prediction Comparison



Georgia Tech

In the resulting plot, the predictions capture some of the volatility. However, all models underpredict the conditional variance or the volatility while still capturing the large volatility during the covid19 crisis.

Summary:

This lesson illustrated the ARMA-GARCH modeling with the time series of the stock price return of a crude oil and natural gas producer.

3.2.3 Other GARCH Models

In this lesson I will conclude the introduction of the GARCH modeling with other extensions of more advanced heteroskedasticity models, most being extensions of the GARCH model.

Other Heteroskedasticity Models: EGARCH

Other Heteroskedasticity Models: EGARCH

Standardized residuals from GARCH models still display some presence of fat tails. This led to the following models:

- R_t is modelled as having a non-Gaussian distribution (e.g. t-distribution)
- EGARCH (exponential GARCH)
  1. It models $\log(\sigma_t^2)$ instead of σ_t^2.
  2. It allows the variance to have asymmetric behavior to reflect any asymmetries in the data. This model uses $|Z_{t-i}|$ along with Z_{t-i} to capture the asymmetries in the returns data in the heteroskedasticity model.

Reference: Daniel B. Nelson (1991). "Conditional heteroskedasticity in asset returns: A new approach." Econometrica, 59, pages 347-370

A first extension of the GARCH was introduced because of one of the limitations of the simple version of GARCH introduced in the previous lessons. Particularly, the residuals from GARCH models often display fat tails, suggesting that a distribution with fatter tails than the normal distribution should be considered, for example, the t-distribution. We have seen this illustrated in the modeling using GARCH for the return of the PDCE stock price when we used the t-distribution. In 1991, Daniel Nelson took this idea further by proposing a model with a transformation of the conditional variance, specifically the log transformation. Thus, according to this model, called also EGARCH, or exponential GARCH model, the log of sigma_t squared is modeled using a GARCH formulation. This transformation dealt also with asymmetric tails, since the model allows the variance to have asymmetric behavior to reflect asymmetries in the volatility. This addresses one other limitation of GARCH, the assumption that the impact of positive and negative shocks is similar.

Other Heteroskedasticity Models: APARCH

Other Heteroskedasticity Models: APARCH

- In some time series, large negative observations appear to increase volatility more than do positive observations of the same magnitude. This is called the *leverage effect*.
- GARCH models cannot model the leverage effect because they model σ_t^2 as a function of past values of Z_t^2-- whether positive or negative is not taken into account
- The solution is to replace the square function with a flexible class of nonnegative functions that include asymmetric functions.
- APARCH(p; q) model for the conditional standard deviation

$$\sigma_t^\delta = \gamma_0 + \sum_{i=1}^{p} \alpha_i (|Z_{t-i}| - \gamma_i Z_{t-i})^\delta + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^\delta$$

- For δ=2 and $\gamma_1 = \cdots = \gamma_p = 0$ it becomes GARCH

Another model which allows for asymmetry in the impact of negative and positive shocks on the volatility is APARCH. The intuition of this model is that for some time series, large negative observations appear to increase volatility, more than do positive observations of the same magnitude. This is called the leverage effect. The solution is to replace the square function applied to the Z_t's with a flexible class of non-negative functions, including asymmetric functions. The APARCH model is an ARCH model formulation for the conditional standard deviation rather than variance. Specifically, it models the power of the standard deviation. If the power denoted by delta is equal to two and the gamma coefficients in the APARCH are zero then we have the GARCH model. Thus the GARCH model is a particular case of APARCH.

Other Heteroskedasticity Models

Other Heteroskedasticity Models

- IGARCH (Integrated GARCH). These are models where the variance is nonstationary. It has the form of the GARCH model, where the GARCH parameters add to one. For example, an IGARCH(1,1) model will have the form:

$$\sigma_t^2 = \gamma_0 + \gamma_1 Z_{t-1}^2 + (1 - \gamma_1)\sigma_{t-1}^2$$

- Threshold GARCH model of Zakoian (1994) or GJR model of Glosten, Jagannathan, and Runkle (1993).
- Conditional heteroscedastic ARMA (CHARMA) model of Tsay (1987).
- Random coefficient autoregressive (RCA) model of Nicholls and Quinn (1982).
- Stochastic volatility (SV) models of Melino and Turnbull (1990), Harvey, Ruiz and Shephard (1994), and Jacquier, Polson and Rossi (1994).

Another extension of GARCH is the Integrated GARCH which applies when the conditional variance is nonstationary. The difference between GARCH and IGARCH is that the GARCH parameters add to one. The most popular is IGARCH with orders 1 and 1 as presented on the slide, where the coefficients for Zt-1 squared is gamma_1 and the coefficient for the lagged variance sigma_t-1 squared is 1 – gamma_1, hence the sum of the two coefficients is 1. But EGARCH, APARCH and IGARCH are just three of the many other extensions. The rest of the slide points to many other references where such extensions have been introduced.

Summary: This lesson concludes the introduction of main concepts on modeling heteroskedasticity using GARCH models. Next I will introduce a case study.