Financial Econometrics Analysis on S&P-500 Returns (Jan 1980 - Dec 2008) and US Quarterly GDP Growth Rate (Q1 1955 - Q4 2004)

Zhiyu Chen
Imperial College London
CID: 02517659

02-12-2023

Contents

Loading R packages

```
library(readxl)
library(stats)
library(sandwich)
library(lmtest)
library(rugarch)
library(fGarch)
library(dplyr)
library(forecast)
library(fUnitRoots)
```

Dataset 1 - SP500WeekDays

This dataset contains the daily simple returns of the S&P 500 composite index from January 1980 to December 2008. The index returns include dividend distributions. The data file is S&P500WeekDays which has 9 columns. The columns are (year, month, day, SP, M, T, W, H, F), where M, T, W, H, F denotes indicator variables for Monday to Friday, respectively.

Multiple Linear Regression Model on Weekday Effects

Use a regression model to study the effects of trading days on the index returns. The fitted model is Multiple Linear Regression model because thee are more than one independent variable (in this case, M, T, W, H, F are all independent variables.)

So the model should be: SP Return= 0+ 1 · Monday+ 2 · Tuesday+ 3 · Wednesday+ 4 · Thursday+ 5 · Friday+

where 0 is the intercept, 1 to 5 are the coefficients for the weekday indicators, and is the error term.

```
data <- read_excel("SP500WeekDays.xlsx")
data <- na.omit(data)

# The dependent variable: S&P 500 daily returns
y <- data$sp

# The independent variable: indicators for the weekdays
X <- data[, c('M', 'T', 'W', 'R', 'F')]

# Fit the linear regression model
model <- lm(y ~ M + T + W + R + F, data = data)

# This shows the summary of the model
summary(model)
```

```
##
## Call:
## lm(formula = y ~ M + T + W + R + F, data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.204627 -0.005214  0.000142  0.005379  0.115842
##
```

```
## Coefficients: (1 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0003290  0.0002923   1.125    0.260
## M           -0.0003711  0.0004184  -0.887    0.375
## T            0.0004114  0.0004107   1.002    0.317
## W            0.0003108  0.0004106   0.757    0.449
## R           -0.0002646  0.0004126  -0.641    0.521
## F                   NA         NA      NA       NA
##
## Residual standard error: 0.01117 on 7314 degrees of freedom
## Multiple R-squared:  0.0007543,  Adjusted R-squared:  0.0002078
## F-statistic:  1.38 on 4 and 7314 DF,  p-value: 0.238
```

As the sum of dummy from Monday to Friday equals one, it results in the NA of one of the five variables. To refit the model, We just need to omit one day among the five weekdays. In this occasion, we omit F (Friday).

```
# Refit the model
model_refit <- lm(y ~ M + T + W + R, data = data)
summary(model_refit)
```

```
##
## Call:
## lm(formula = y ~ M + T + W + R, data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.204627 -0.005214  0.000142  0.005379  0.115842
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0003290  0.0002923   1.125    0.260
## M           -0.0003711  0.0004184  -0.887    0.375
## T            0.0004114  0.0004107   1.002    0.317
## W            0.0003108  0.0004106   0.757    0.449
## R           -0.0002646  0.0004126  -0.641    0.521
##
## Residual standard error: 0.01117 on 7314 degrees of freedom
## Multiple R-squared:  0.0007543,  Adjusted R-squared:  0.0002078
## F-statistic:  1.38 on 4 and 7314 DF,  p-value: 0.238
```

```
# Extract the p-values
p_values <- coef(summary(model_refit))[, "Pr(>|t|)"]
p_values
```

```
## (Intercept)           M           T           W           R
##   0.2604232   0.3751917   0.3165380   0.4491202   0.5213436
```

The re-fitted model is:

$$sp = -0.0003290 + (-0.0003711) \cdot M + 0.0004114 \cdot T + 0.0003108 \cdot W + (-0.0002646) \cdot R$$

The p-values for the coefficients of Monday, Tuesday, Wednesday, and Thursday are 0.375, 0.317, 0.449, 0.521. None of them are statistically significant at a 5% significance level (p-value < 0.05). Thus the the weekday effects are **NOT significant** in the returns at the 5% level.

Newey West Estimator Analysis on Weekday Effects

The Newey West Estimator provides consistent standard errors for coefficient estimates in the presence of heteroskedasticity and autocorrelation.The estimator adjusts the covariance matrix of the coefficient estimates to account for these issues, thus improving the reliability of hypothesis testing.

$$\text{Var}(\hat{\beta})_{NW} = (X'X)^{-1} \left(\sum_{t=1}^{T} \epsilon_t^2 X_t X_t' + \sum_{l=1}^{L} w_l \sum_{t=l+1}^{T} \epsilon_t \epsilon_{t-l} (X_t X_{t-l}' + X_{t-l} X_t') \right) (X'X)^{-1}$$

where:

- $\text{Var}(\hat{\beta})_{NW}$ is the Newey-West adjusted covariance matrix.
- $X'X$ is the product of the matrix of independent variables and its transpose.
- ϵ_t is the residual at time t.
- L is the chosen lag length.
- w_l are the weights assigned to the lagged terms.

Use the Newey West estimator of the covariance matrix to obtain the t-ratio of regression estimates.

```
# Calculate Newey-West standard errors
nw_se <- NeweyWest(model_refit)

#Use coeftest to get t-ratio with Newey-West standard errors
t_ratio <- coeftest(model, vcov = nw_se)
t_ratio
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value Pr(>|t|)
## (Intercept)  0.00032896  0.00026007  1.2649   0.2060
## M           -0.00037109  0.00042757 -0.8679   0.3855
## T            0.00041140  0.00038543  1.0674   0.2858
## W            0.00031078  0.00038643  0.8042   0.4213
## R           -0.00026463  0.00038645 -0.6848   0.4935
```

T-ratio (t-statistic) in regression analysis is the ratio of the estimated coefficient to its standard error. It's used to test the null hypothesis that the coefficient is equal to zero. Typically, a t-ratio greater than +1.96 or less than -1.96 is considered statistically significant at the 5% level.

In this occasion, t-value for Monday is -0.8679, for Tuesday is 1.0674, for Wednesday is 0.8042 and for Thursday if -0.6848. No t-value satisfies greater than +1.96 or less than -1.96, so the Newey West estimator **does NOT change** the conclusion of weekday effect, that weekday effects are not significant in the returns at the 5% level.

The ARCH(1) and GARCH(1,1) Models for the log Returns

The Autoregressive Conditional Heteroskedasticity (ARCH), is a statistical model used to describe and predict time series data, particularly the volatility of financial time series.

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2$$

The Generalized Autoregressive Conditional Heteroskedasticity (GARCH), is an extension of the ARCH model. It's also widely used in financial econometrics to model time series data, particularly for capturing the volatility (time-varying variance) of financial returns.

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

where:

- σ_t^2 is the conditional variance at time t.
- α_0 is a constant term, representing the long-run average variance.
- α_1 is the coefficient for the lagged squared error term, representing the impact of short-term shocks on current volatility.
- ϵ_{t-1}^2 is the squared error term (or residual) from the previous time period.
- β_1 is the coefficient for the lagged conditional variance, indicating the persistence of volatility over time.
- σ_{t-1}^2 is the conditional variance from the previous time period.

Firstly, fit the ARCH(1) and GARCH (1,1) for the SP_500 log returns.

For ARCH(1), set up a standard GARCH model which, with garchOrder "c(1,0)", becomes an ARCH(1) model as the GARCH term is set to 0.

For GARCH(1,1), set up a standard GARCH model which, with garchOrder "c(1,1)".

```
# Calculate the log returns
log_returns <- log(1 + data$sp)
log_returns <- na.omit(log_returns)

# Fit an ARCH(1) model
arch <- garchFit(~ garch(1, 0), data = log_returns, trace = FALSE)
summary(arch)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 0), data = log_returns, trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ garch(1, 0)
## <environment: 0x000001680cc02bc0>
##  [data = log_returns]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu       omega      alpha1
## 0.00042144  0.00008727  0.28996725
##
## Std. Errors:
##  based on Hessian
##
```

```
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     4.214e-04   1.163e-04    3.624  0.00029 ***
## omega  8.727e-05   1.857e-06   46.995  < 2e-16 ***
## alpha1 2.900e-01   2.005e-02   14.466  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  22980.96    normalized:  3.139904
##
## Description:
##  Mon Dec  4 13:59:36 2023 by user: 16920
##
##
## Standardised Residuals Tests:
##                                   Statistic      p-Value
##  Jarque-Bera Test   R    Chi^2  23719.17731 0.000000e+00
##  Shapiro-Wilk Test  R    W              NA           NA
##  Ljung-Box Test     R    Q(10)     20.66391 2.356297e-02
##  Ljung-Box Test     R    Q(15)     46.16440 5.006218e-05
##  Ljung-Box Test     R    Q(20)     55.91719 2.991524e-05
##  Ljung-Box Test     R^2  Q(10)   1530.18436 0.000000e+00
##  Ljung-Box Test     R^2  Q(15)   1914.76129 0.000000e+00
##  Ljung-Box Test     R^2  Q(20)   2178.13622 0.000000e+00
##  LM Arch Test       R    TR^2     954.76193 0.000000e+00
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -6.278989 -6.276161 -6.278989 -6.278016
```

```r
# Fit a GARCH(1,1) model
garch <- garchFit(~ garch(1, 1), data = log_returns, trace = FALSE)
summary(garch)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = log_returns, trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x000001680c775150>
##  [data = log_returns]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu       omega      alpha1       beta1
## 5.2147e-04  1.1980e-06  7.3985e-02  9.1768e-01
##
```

```
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     5.215e-04   9.590e-05    5.438 5.39e-08 ***
## omega  1.198e-06   2.152e-07    5.568 2.58e-08 ***
## alpha1 7.398e-02   5.821e-03   12.709  < 2e-16 ***
## beta1  9.177e-01   6.719e-03  136.571  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  23850.16    normalized:  3.258664
##
## Description:
##  Mon Dec  4 13:59:38 2023 by user: 16920
##
##
## Standardised Residuals Tests:
##                                   Statistic    p-Value
##  Jarque-Bera Test   R    Chi^2  7591.007440 0.00000000
##  Shapiro-Wilk Test  R    W              NA         NA
##  Ljung-Box Test     R    Q(10)    20.368768 0.02595150
##  Ljung-Box Test     R    Q(15)    32.298450 0.00586186
##  Ljung-Box Test     R    Q(20)    35.321833 0.01845569
##  Ljung-Box Test     R^2  Q(10)     4.382925 0.92842378
##  Ljung-Box Test     R^2  Q(15)     6.352922 0.97319035
##  Ljung-Box Test     R^2  Q(20)     9.225290 0.98014789
##  LM Arch Test       R    TR^2      5.384850 0.94387509
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -6.516235 -6.512465 -6.516236 -6.514939
```

For the ARCH(1), the p-value for coefficient mu(mean) is 0.00029, omega(variance constant) is less than 2e-16 and alpha1(ARCH term) is less than 2e-16. These three p-values are extremely small and thus, have a significant impact on the model. **The three coefficients for ARCH(1) are highly statistically significant.**

	value	p-value
mu	4.214e-04	0.00029
omega	8.727e-05	$<$ 2e-16
alpha1	2.900e-01	$<$ 2e-16

For the GARCH(1,1), the p-value for coefficient mu(mean) is 5.39e-08, omega(variance constant) is 2.58e-08, alpha1(short-term GARCH term) is less than 2e-16 and beta1(long-term GARCH term) is less than 2e-16. These four p-values are extremely small and thus, also have a significant impact on the model. **The four coefficients for GARCH(1,1) are highly statistically significant.**

	value	p-value
mu	5.215e-04	5.39e-08
omega	1.198e-06	2.58e-08
alpha1	7.398e-02	< 2e-16
beta1	9.177e-01	< 2e-16

```
# Extract coefficients for ARCH(1) model and
# Compute the unconditional variance
omega_arch <- coef(arch)["omega"]
alpha_arch <- coef(arch)["alpha1"]
uncond_var_arch <- omega_arch / (1 - alpha_arch)

# Extract the coefficients for GARCH(1,1) model and
# Compute the unconditional variance
omega_garch <- coef(garch)["omega"]
alpha_garch <- coef(garch)["alpha1"]
beta_garch <- coef(garch)["beta1"]
uncond_var_garch <- omega_garch / (1 - alpha_garch - beta_garch)

# Print the unconditional variance for the two models
round(uncond_var_arch,6)
```

```
##    omega
## 0.000123
```

```
round(uncond_var_garch,6)
```

```
##    omega
## 0.000144
```

The unconditional variance measures the average level of variance (or volatility) that can be expected over a long period.

$$\text{Unconditional Variance} = \frac{\alpha_0}{1 - \alpha_1 - \beta_1}$$

Where:

- α_0 represents the constant term in the GARCH(1,1) model.
- α_1 is the coefficient for the lagged error term, e_{t-1}^2.
- β_1 is the coefficient for the lagged variance term, σ_{t-1}^2.

The unconditional variance given by ARCH(1) is 0.000123, and by GARCH(1,1) is 0.000144.

The high significance of alpha1 and beta1 in GARCH(1,1) indicates that SP500 log returns are greatly affected by both short-term shocks and long-term volatility.

The higher unconditional variance from the GARCH(1,1) compared to the ARCH(1) indicates that GARCH model estimates a higher long-term average volatility for the SP-500 log return from January 1980 to December 2008. The GARCH(1.1) captures a higher level of volatility might be due to its structure that accounts for both volatility clustering and mean reversion in volatility, while ARCH(1) is simpler and only focusing on the immediate past volatility, which may not fully capture the persistence in volatility that GARCH(1,1) can.

Dataset 2 - USMacro_Quarterly

This dataset contains quarterly data on two macroeconomic series for the United States:

1. RealGDP: The quarterly values of Real GDP for the United States, expressed in billions of chained (2000) dollars. The data is seasonally adjusted at an annual rate..
2. TBillRate: The quarterly values of the rate on 3-month Treasury Bills. The values are quarterly averages of daily rates, expressed in percentage points at an annual rate.

The logarithm of real GDP: Y(t) = ln[GDP(t)] The quarterly growth rate of GDP: ΔY(t)

Sample period 1955:1 - 2004:4 is used.

Estimation of the Mean of ΔY(t)

```
# Read the dataset
data_US <- read_excel("USMacro_Quarterly.xls")

# Convert "Date" to a year-quarter format
data_US$Date <- as.yearqtr(data_US$Date, format = "%Y:%q")

# Calculate the logarithm of Real GDP
# Calculate the quarterly growth rate of GDP (ΔY(t))
data_US <- data_US %>%
  mutate(Log_RealGDP = log(RealGDP)) %>%
  mutate(GDP_Growth_Rate = Log_RealGDP - lag(Log_RealGDP))

# Filter for the sample period from 1955:1 to 2004:4
start_period <- as.yearqtr("1955 Q1", format = "%Y Q%q")
end_period <- as.yearqtr("2004 Q4", format = "%Y Q%q")
sample_data <- data_US %>%
  filter(Date >= start_period & Date <= end_period)

# Compute the mean of the GDP Growth Rate (na.rm=TRUE to ignore NA values)
mean_gdp_growth_rate <- mean(sample_data$GDP_Growth_Rate, na.rm = TRUE)
mean_gdp_growth_rate
```

```
## [1] 0.008258661
```

The quarter mean of GDP growth rate is **0.00826**

The Mean Growth Rate in Percentage Points at Annual Rate

To express the mean growth rate in percentage points at an annual rate, multiply the quarterly mean growth rate by 400. (Quarterly to annualy: x4; Decimal form to percentage form: x100)

```
# Get the mean growth rate in percentage points at annual rate
annual_mean_gdp_growth_rate <- mean_gdp_growth_rate * 400
annual_mean_gdp_growth_rate
```

```
## [1] 3.303464
```

The annual mean of GDP growth rate is **3.30%**

Estimation of the Standard Deviation of $\Delta Y(t)$

The result is in percentage points at an annual rate.

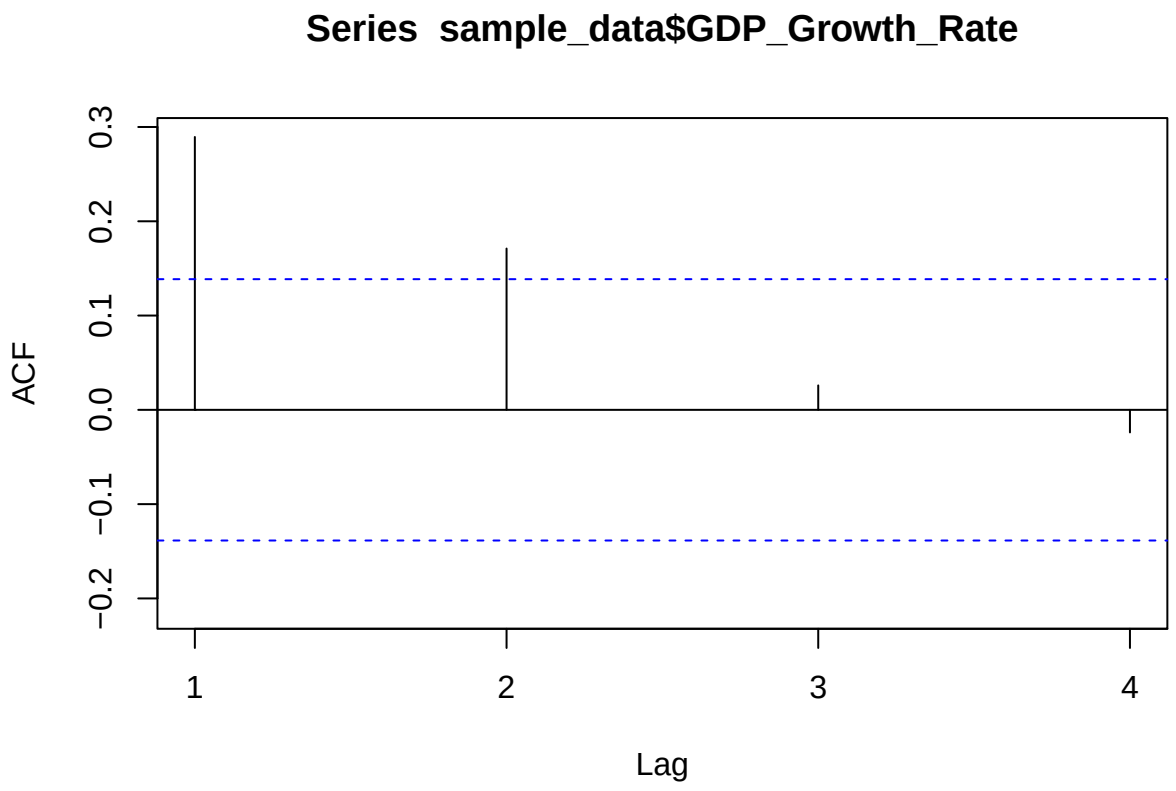
```
# Compute the standard deviation of the GDP Growth Rate (ΔY(t))
std_dev_gdp_growth_rate <- sd(sample_data$GDP_Growth_Rate, na.rm = TRUE)

# Convert the result in % points at annual rate
annual_std_dev_gdp_growth_rate <- std_dev_gdp_growth_rate * 400
annual_std_dev_gdp_growth_rate
```

```
## [1] 3.682321
```

The annual standard deviation of GDP growth rate is **3.68%**

Estimation of the First Four Autocorrelations of $\Delta Y(t)$

```
# The autocorrelations of ΔY(t)
autocorrelations <- Acf(sample_data$GDP_Growth_Rate, lag.max = 4, plot = T)
```




```
# Find the first four autocorrelations
autocorr_4 <- autocorrelations$acf[2:5]
autocorr_4
```

```
## [1]  0.28940872  0.17108983  0.02594531 -0.02396060
```

After omitting the autocorrelation of exactly itself(Lag=0 and the ACF value must be 1), **the fist four autocorrelations are: 0.2894(Lag=1), 0.1711(Lag=2), 0.0230(Lag=3) and -0.0240(Lag=4).** Autocorrelations do not have units as they are statistical measures that quantify the degree of correlation between a time series and the lagged versions of itself. These coefficients are expressed as values between -1 and +1, regardless of the units of the original data.

AR(1) Model Estimation for ΔY(t)

The Autoregression model of order 1 (AR(1) model), is a basic yet widely used time series model that explains a variable in terms of its own previous value. The AR(1) model is characterized by a single lagged term of the variable.

$$Y_t = \mu + \phi Y_{t-1} + \epsilon_t$$

where:

- Y_t is the value of the time series at time t.
- μ is the constant term or intercept.
- ϕ is the autoregressive coefficient for the first lag of the series.
- Y_{t-1} is the value of the series at the previous time step.
- ϵ_t is the error term, representing random fluctuations that cannot be explained by the model.

Estimate an AR(1) model for ΔY(t). The Arima() function is used to fit an AR(1) model, specified by 'order = c(1,0,0)'

```
# Fit the AR(1)
ar1_model <- arima(sample_data$GDP_Growth_Rate, order=c(1,0,0))
ar1_model
```

```
##
## Call:
## arima(x = sample_data$GDP_Growth_Rate, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.2951     0.0083
## s.e.  0.0682     0.0009
##
## sigma^2 estimated as 7.709e-05:  log likelihood = 663.23,  aic = -1320.45
```

```
summary(ar1_model)
```

```
##
## Call:
## arima(x = sample_data$GDP_Growth_Rate, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.2951     0.0083
```

```
## s.e.  0.0682     0.0009
##
## sigma^2 estimated as 7.709e-05:  log likelihood = 663.23,  aic = -1320.45
##
## Training set error measures:
##                         ME        RMSE         MAE       MPE     MAPE      MASE
## Training set -3.423424e-05 0.008779809 0.006513686 -37.92503 277.7068 0.7980798
##                     ACF1
## Training set -0.03398181
```

```
# Extract the estimated AR(1) coefficient
ar1_coefficient <- ar1_model$coef[1]
ar1_coefficient
```

```
##       ar1
## 0.2950893
```

The estimated AR(1) coefficient is 0.2951. To determine whether the estimated AR(1) coefficient is statistically significantly different from zero, calculate the t-statistic and the corresponding p-value for this coefficient in the output of AR(1) model.

```
# Extract standard error from the model
ar1_se <- sqrt(diag(vcov(ar1_model)))[1]

# Calculate the t-statistic
t_statistic_ar1 <- ar1_coefficient / ar1_se
t_statistic_ar1
```

```
##      ar1
## 4.325886
```

```
# Calculate the degrees of freedom for the t-distribution
df <- length(sample_data$GDP_Growth_Rate) - ar1_model$arma[1] - 1

# Calculate the two-tailed p-value
p_value_ar1 <- 2 * pt(-abs(t_statistic_ar1), df)
p_value_ar1
```

```
##          ar1
## 2.407503e-05
```

For the AR(1) coefficient, the absolute value of t-statistic is 4.3260 (larger than 2) and the p-value is 0.000024 (less than 0.05), so it is commonly considered statistically significant at the 5% level (95% confidence interval), indicating that **AR(1) coefficient is significantly different from zero.**

AR(2) Model Estimation for $\Delta Y(t)$

The Autoregressive model of order 2 (AR(2) model), is a time series model where the current value of the series is explained by its own two previous values. This model is useful when the data shows evidence of being influenced by the last two periods.

$$Y_t = \mu + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \epsilon_t$$

where:

- Y_t is the value of the time series at time t.
- μ is the constant term or intercept.
- ϕ_1 and ϕ_2 are the autoregressive coefficients for the first and second lags of the series.
- Y_{t-1} and Y_{t-2} are the values of the series at the previous two time steps.
- ϵ_t is the error term, representing random fluctuations that cannot be explained by the model.

Estimate an AR(2) model for ΔY(t). The Arima() function is used to fit an AR(2) model, specified by 'order = c(2,0,0)'

```
# Fit the AR(2)
ar2_model <- arima(sample_data$GDP_Growth_Rate, order=c(2,0,0))
ar2_model
```

```
##
## Call:
## arima(x = sample_data$GDP_Growth_Rate, order = c(2, 0, 0))
##
## Coefficients:
##          ar1     ar2  intercept
##       0.2669  0.0979     0.0083
## s.e.  0.0709  0.0709     0.0010
##
## sigma^2 estimated as 7.635e-05:  log likelihood = 664.18,  aic = -1320.35
```

```
# Extract the estimated coefficients for the two lags
ar2_coefficient <- ar2_model$coef[2]
ar2_coefficient
```

```
##        ar2
## 0.09786695
```

The estimated AR(2) coefficient is 0.0979. To determine whether it is statistically significantly different from zero, calculate the t-statistic and p-value

```
# Extract standard error from the model
ar2_se <- sqrt(diag(vcov(ar2_model)))[2]

# Calculate the t-statistic
t_statistic_ar2 <- ar2_coefficient / ar2_se
t_statistic_ar2
```

```
##      ar2
## 1.380725
```

```r
# Calculate the degrees of freedom for the t-distribution
df <- length(sample_data$GDP_Growth_Rate) - ar2_model$arma[1] - 1

# Calculate the two-tailed p-value
p_value_ar2 <- 2 * pt(-abs(t_statistic_ar2), df)
p_value_ar2
```

```
##       ar2
## 0.1689281
```

For the AR(2) coefficient, the absolute value of t-statistic is 1.3807 (not larger than 2) and the p-value is 0.1689 (not less than 0.05), so it is commonly considered NOT statistically significant at the 5% level (95% confidence interval), indicating that **AR(2) coefficient is NOT significantly different from zero.**

Based on the calculation, AR(1) coefficient is statistically significant at 95% confidence interval while AR(2) coefficient is not. So AR(1) model is more justified than AR(2) model for ΔY(t). The lack of statistical significance of the AR(2) coefficient suggests that adding the second lag does not provide additional explanatory power to the model that is statistically meaningful.

AR(3) Model Estimation for ΔY(t)

Estimate an AR(3) model for ΔY(t). The Arima() function is used to fit an AR(3) model, specified by 'order = c(3,0,0)'

```r
# Fit the AR(3)
ar3_model <- arima(sample_data$GDP_Growth_Rate, order=c(3,0,0))
ar3_model
```

```
##
## Call:
## arima(x = sample_data$GDP_Growth_Rate, order = c(3, 0, 0))
##
## Coefficients:
##          ar1     ar2     ar3  intercept
##       0.2718  0.1111  -0.051     0.0083
## s.e.  0.0711  0.0731   0.071     0.0009
##
## sigma^2 estimated as 7.615e-05:  log likelihood = 664.43,  aic = -1318.87
```

AR(4) Model Estimation for ΔY(t)

Estimate an AR(4) model for ΔY(t). The Arima() function is used to fit an AR(4) model, specified by 'order = c(4,0,0)'

```r
# Fit the AR(4)
ar4_model <- arima(sample_data$GDP_Growth_Rate, order=c(4,0,0))
ar4_model
```

```
##
## Call:
## arima(x = sample_data$GDP_Growth_Rate, order = c(4, 0, 0))
```

```
##
## Coefficients:
##          ar1     ar2      ar3      ar4  intercept
##       0.2699  0.1155  -0.0403  -0.0403     0.0083
## s.e.  0.0711  0.0735   0.0733   0.0709     0.0009
##
## sigma^2 estimated as 7.603e-05:  log likelihood = 664.59,  aic = -1317.19
```

AR(1)-AR(4) Bayesian Information Criterion (BIC) Model Selection Methodology

The Bayesian Information Criterion (BIC), is a criterion for model selection among a finite set of models. It is based on the likelihood function and is used extensively in statistical model fitting.

$$\text{BIC} = \ln(n)k - 2\ln(\hat{L})$$

where:

- n is the number of observations.
- k is the number of parameters in the model.
- ln is the natural logarithm.
- $\hat{L}$ is the maximized value of the likelihood function of the model.

In model selection, the model with the lowest BIC is generally preferred. The lower BIC suggests either a better fit, fewer parameters, or both.

```
# Extract BIC values
bic_values <- c(BIC(ar1_model),
                BIC(ar2_model),
                BIC(ar3_model),
                BIC(ar4_model))
bic_values
```

```
## [1] -1310.559 -1307.157 -1302.374 -1297.398
```

```
# Determine the optimal number of lags
optimal_lags_bic <- which.min(bic_values)
optimal_lags_bic
```

```
## [1] 1
```

As AR(1) has the lowest BIC values, **the optimal number of lags in the AR model according to the BIC criterion is 1.**

AR(1)-AR(4) Akaike Information Criterion (AIC) Model Selection Methodology

The Akaike Information Criterion (AIC), is used to compare different models and select the one that best explains the data while avoiding overfitting. It balances the model's complexity against its goodness of fit.

$$\text{AIC} = 2k - 2\ln(\hat{L})$$

where:

- k is the number of parameters in the model.
- $\ln(\hat{L})$ is the natural logarithm of the maximized likelihood function of the model.

In model selection, the model with the lowest AIC is generally preferred. The lowest AIC value among a set of models indicates the model that best balances the fit to the data and the complexity of the model.

```
# Extract BIC values
aic_values <- c(AIC(ar1_model),
                AIC(ar2_model),
                AIC(ar3_model),
                AIC(ar4_model))
aic_values
```

```
## [1] -1320.454 -1320.350 -1318.866 -1317.188
```

```
# Determine the optimal number of lags
optimal_lags_aic <- which.min(aic_values)
optimal_lags_aic
```

```
## [1] 1
```

As AR(1) has the lowest AIC values, **the optimal number of lags in the AR model according to the AIC criterion is also 1.**

The Augmented Dickey-Fuller (ADF) Test for ΔY(t)

The Augmented Dickey-Fuller (ADF), tests for a unit root in the time series. This test can help determine whether a time series is stationary or not, which is a critical aspect of many time series analyses, including AR modeling.

$$\Delta Y_t = \alpha + \beta t + \gamma Y_{t-1} + \delta_1 \Delta Y_{t-1} + \delta_2 \Delta Y_{t-2} + \cdots + \delta_p \Delta Y_{t-p} + \epsilon_t$$

where:

- ΔY_t is the first difference of the series at time t.
- α is the constant term.
- βt is the coefficient of the time trend.
- γ is the coefficient on the lagged level of the series. The null hypothesis of the ADF test is that this coefficient is zero (indicating a unit root).
- $\delta_1, \delta_2, \ldots, \delta_p$ are the coefficients on the lagged differences of the series.
- ϵ_t is the error term.

In this scenario the ΔY(t) is expected to be stationary around a deterministic trend. In addition, choose lags = 1 because from the previous BIC and AIC model selection, AR(1) performs the best.

```
# Applying the ADF test
adfTest(sample_data$GDP_Growth_Rate,lags = 1,type = c("c"))
```

```
## Warning in adfTest(sample_data$GDP_Growth_Rate, lags = 1, type = c("c")):
## p-value smaller than printed p-value
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 1
##   STATISTIC:
##     Dickey-Fuller: -7.6633
##   P VALUE:
##     0.01
##
## Description:
##  Mon Dec  4 13:59:38 2023 by user: 16920
```

The null hypothesis of ADF is that there is a unit root (implying non-stationarity). The outcome shows that the p-value of the ADF test is smaller than 0.01 (less than 0.05), suggesting rejecting the null hypothesis of a unit autoregression root, indicating that **$\Delta Y(t)$ is stationary.**

The ARCH(1) and GARCH(1,1) Models for $\Delta Y(t)$

For ARCH(1), set up a standard GARCH model which, with garchOrder "c(1,0)", becomes an ARCH(1) model as the GARCH term is set to 0.

For GARCH(1,1), set up a standard GARCH model which, with garchOrder "c(1,1)".

```r
# Fit an ARCH(1) model
arch_GDP <- garchFit(~ garch(1, 0),
                     data = sample_data$GDP_Growth_Rate, trace = FALSE)
summary(arch_GDP)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 0), data = sample_data$GDP_Growth_Rate,
##     trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ garch(1, 0)
## <environment: 0x0000016816011300>
##  [data = sample_data$GDP_Growth_Rate]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu       omega      alpha1
## 8.6215e-03  6.4033e-05  2.6822e-01
##
## Std. Errors:
##  based on Hessian
```

```
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     8.622e-03   6.594e-04   13.074  < 2e-16 ***
## omega  6.403e-05   9.283e-06    6.898 5.27e-12 ***
## alpha1 2.682e-01   1.317e-01    2.036   0.0418 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  657.9403    normalized:  3.289701
##
## Description:
##  Mon Dec  4 13:59:38 2023 by user: 16920
##
##
## Standardised Residuals Tests:
##                                 Statistic      p-Value
##  Jarque-Bera Test   R    Chi^2  10.7160704 0.0047101516
##  Shapiro-Wilk Test  R    W       0.9816597 0.0102821714
##  Ljung-Box Test     R    Q(10)  27.9010459 0.0018722797
##  Ljung-Box Test     R    Q(15)  32.8326395 0.0049505149
##  Ljung-Box Test     R    Q(20)  37.0559547 0.0115225492
##  Ljung-Box Test     R^2  Q(10)  29.6996913 0.0009590588
##  Ljung-Box Test     R^2  Q(15)  41.6396138 0.0002553096
##  Ljung-Box Test     R^2  Q(20)  54.0581890 0.0000567096
##  LM Arch Test       R    TR^2   28.1287779 0.0052969876
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -6.549403 -6.499928 -6.549844 -6.529381
```

```r
# Fit a GARCH(1,1) model
garch_GDP <- garchFit(~ garch(1, 1),
                      data = sample_data$GDP_Growth_Rate, trace = FALSE)
summary(garch_GDP)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = sample_data$GDP_Growth_Rate,
##     trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x0000016813964878>
##  [data = sample_data$GDP_Growth_Rate]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
```

```
##         mu       omega      alpha1       beta1
## 8.9695e-03  2.2462e-06  2.1697e-01  7.7440e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     8.969e-03   6.569e-04   13.655   <2e-16 ***
## omega  2.246e-06   1.982e-06    1.133   0.2571
## alpha1 2.170e-01   9.845e-02    2.204   0.0275 *
## beta1  7.744e-01   8.658e-02    8.944   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  670.0669    normalized:  3.350334
##
## Description:
##  Mon Dec  4 13:59:38 2023 by user: 16920
##
##
## Standardised Residuals Tests:
##                                  Statistic     p-Value
##  Jarque-Bera Test   R    Chi^2   6.9966166 0.030248512
##  Shapiro-Wilk Test  R    W       0.9896694 0.159812275
##  Ljung-Box Test     R    Q(10)  24.0988404 0.007342316
##  Ljung-Box Test     R    Q(15)  32.4788629 0.005537591
##  Ljung-Box Test     R    Q(20)  35.8055885 0.016210295
##  Ljung-Box Test     R^2  Q(10)  12.3293104 0.263621351
##  Ljung-Box Test     R^2  Q(15)  14.6494440 0.476950543
##  Ljung-Box Test     R^2  Q(20)  21.8080780 0.351015030
##  LM Arch Test       R    TR^2   12.6852259 0.392328799
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -6.660669 -6.594702 -6.661448 -6.633973
```

	value	p-value
mu	8.622e-03	< 2e-16
omega	6.403e-05	5.27e-12
alpha1	2.682e-01	0.0418

For the ARCH(1), the p-value for coefficient mu(mean) is less than 2e-16, omega(variance constant) is 5.27e-12 and alpha1(ARCH term) is 0.0418. The p-values for mu and omega are extremely small, suggesting that coefficient mu and omega have a significant impact on the model. The p-value for alpha1 is 0.04 (slightly less than 0.05), means that coefficient alpha1 has impact on the model at 95% confidence interval. **In summary, the three coefficients for ARCH(1) are statistically significant.**

	value	p-value
mu	8.969e-03	< 2e-16

	value	p-value
omega	2.246e-06	0.2571
alpha1	2.170e-01	0.0275
beta1	7.744e-01	<2e-16

For the GARCH(1,1), the p-value for coefficient mu(mean) is less than 2e-16, omega(variance constant) is 0.2571, alpha1(short-term GARCH term) is 0.0275 and beta1(long-term GARCH term) is less than 2e-16. The p-value for mu is extremely small, suggesting that mu has a significant impact on the model. For omega is 0.2571 (not less than 0.05), suggesting that coefficient omega have NO statically significant impact on the model at 5% significance level. The p-value for alpha1 is 0.0275 (less than 0.05), suggesting that coefficient alpha has impact on the model at 95% confidence interval. The p-value of beta1 is less than 2e-16 which is extremely small, suggesting that beta1 has a significant impact on the model. **In summary, three coefficients mu alpha1 and beta1 for GARCH(1,1) are statistically significant, while coefficient omega is not.**

```
# Extract coefficients for ARCH(1) model and
# Compute the unconditional variance
omega_arch_GDP <- coef(arch_GDP)["omega"]
alpha_arch_GDP <- coef(arch_GDP)["alpha1"]
uncond_var_arch_GDP <- omega_arch_GDP / (1 - alpha_arch_GDP)

# Extract the coefficients for GARCH(1,1) model and
# Compute the unconditional variance
omega_garch_GDP <- coef(garch_GDP)["omega"]
alpha_garch_GDP <- coef(garch_GDP)["alpha1"]
beta_garch_GDP <- coef(garch_GDP)["beta1"]
uncond_var_garch_GDP <- omega_garch_GDP /
  (1 - alpha_garch_GDP - beta_garch_GDP)

# Print the unconditional variance for the two models
round(uncond_var_arch_GDP,6)
```

```
##   omega
## 8.8e-05
```

```
round(uncond_var_garch_GDP,6)
```

```
##   omega
## 0.00026
```

The unconditional variance measures the average level of variance (or volatility) that can be expected over a long period. **The unconditional variance given by ARCH(1) is 0.00008, and by GARCH(1,1) is 0.00026.**

The significance of alpha1 and beta1 in GARCH(1,1) indicates that the quarterly GDP growth rate of US is affected by both short-term shocks and long-term volatility. However, the effect of long-term volatility (represent by beta1) is way more than short-term shocks (represent by alpha1).

The higher unconditional variance from the GARCH(1,1) compared to the ARCH(1) indicates that GARCH model estimates a higher long-term average volatility for the quarterly growth rate of US GDP from 1st quarter 1955 to 4th quarter 2004. The GARCH(1.1) captures a higher level of volatility might be due to

its structure that accounts for both volatility clustering and mean reversion in volatility, while ARCH(1) is simpler and only focusing on the immediate past volatility, which may not fully capture the persistence in volatility that GARCH(1,1) can.