Multinomial GARCH

Dheer Avashia

2024-12-12

Data

```r
fiveyearsap <- read.csv("C:/Users/dheer/OneDrive/Desktop/all_stocks_5yr.csv") # Import Dataset
fiveyearsap <- fiveyearsap %>%
  mutate(date = ymd(date)) %>% mutate(quarter = floor_date(date, "quarter"))
```

Returns

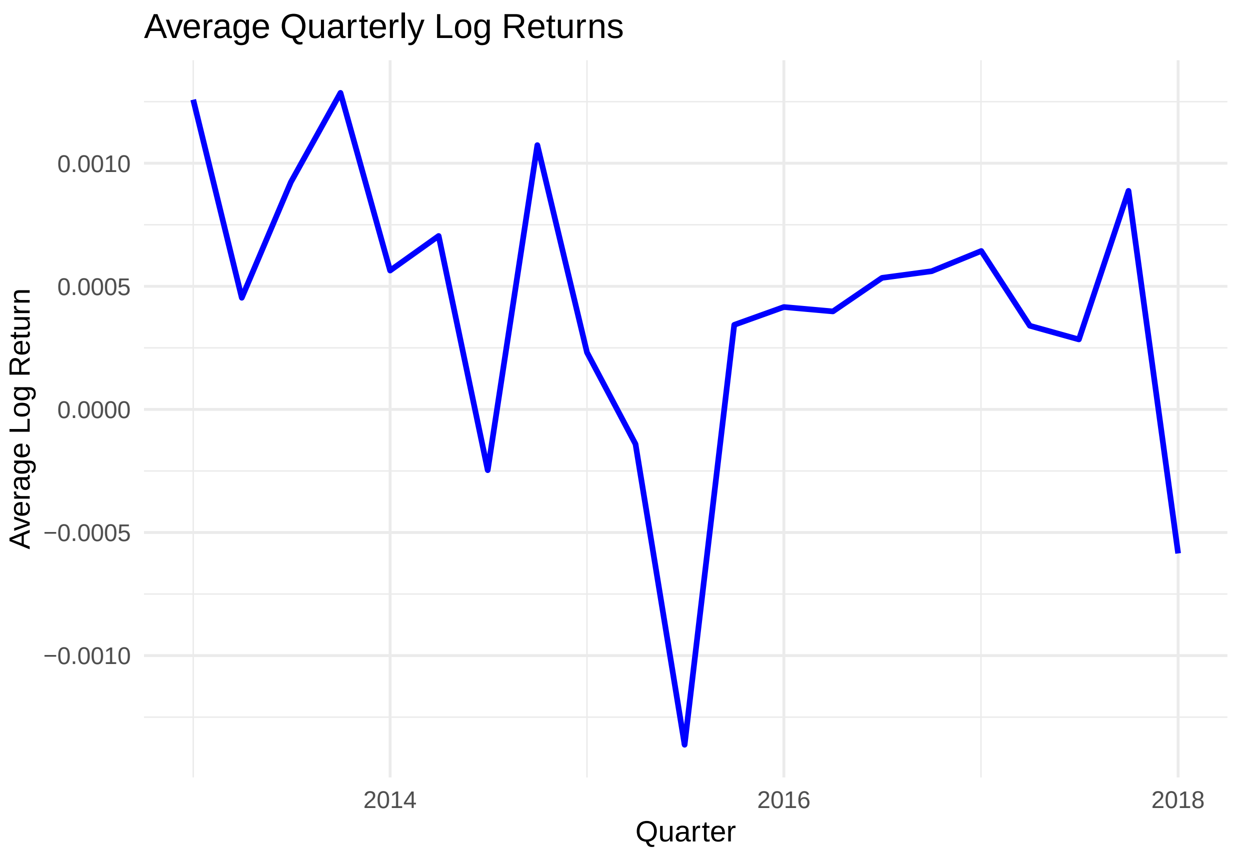
```r
fiveyearsap<- fiveyearsap %>% arrange(Name, date)
fiveyearsap <- fiveyearsap %>%
  group_by(Name) %>%
  mutate(log_return = log(lead(open)) - log(open)) %>% filter(!is.na(log_return)) %>% ungroup()
average_returns <- fiveyearsap %>%
  group_by(date) %>%
  summarize(avg_log_return = mean(log_return, na.rm = TRUE)) %>% ungroup()
```

Top ten by average volume

```r
avg_volume <- fiveyearsap %>%
  group_by(Name) %>%
  summarise(Avg_Volume = mean(volume, na.rm = TRUE)) %>%
  arrange(desc(Avg_Volume))
# Calculate average trading volume for each stock
avg_volume <- fiveyearsap %>%
  group_by(Name) %>%
  summarise(Avg_Volume = mean(volume, na.rm = TRUE)) %>%
  arrange(desc(Avg_Volume))

# Select the top 10 most traded stocks by volume
top10_stocks <- avg_volume %>% slice_head(n = 10)

# Filter the main dataset for only the top 10 stocks
fiveyearsap_top10 <- fiveyearsap %>%
  filter(Name %in% top10_stocks$Name)

# Arrange data by stock name and date
fiveyearsap_top10 <- fiveyearsap_top10 %>% arrange(Name, date)

# Calculate log returns for the top 10 stocks
fiveyearsap_top10 <- fiveyearsap_top10 %>%
  group_by(Name) %>%
  mutate(log_return = log(lead(open)) - log(open)) %>%
  filter(!is.na(log_return)) %>%
  ungroup()
# Create summary statistics for the top 10 stocks
summary_top10 <- fiveyearsap_top10 %>%
  group_by(Name) %>%
  summarise(
    Avg_Open = round(mean(open, na.rm = TRUE), 2),
    Avg_Close = round(mean(close, na.rm = TRUE), 2),
    Mean_High = round(mean(high, na.rm = TRUE), 2),
    Mean_Low = round(mean(low, na.rm = TRUE), 2),
  )
```

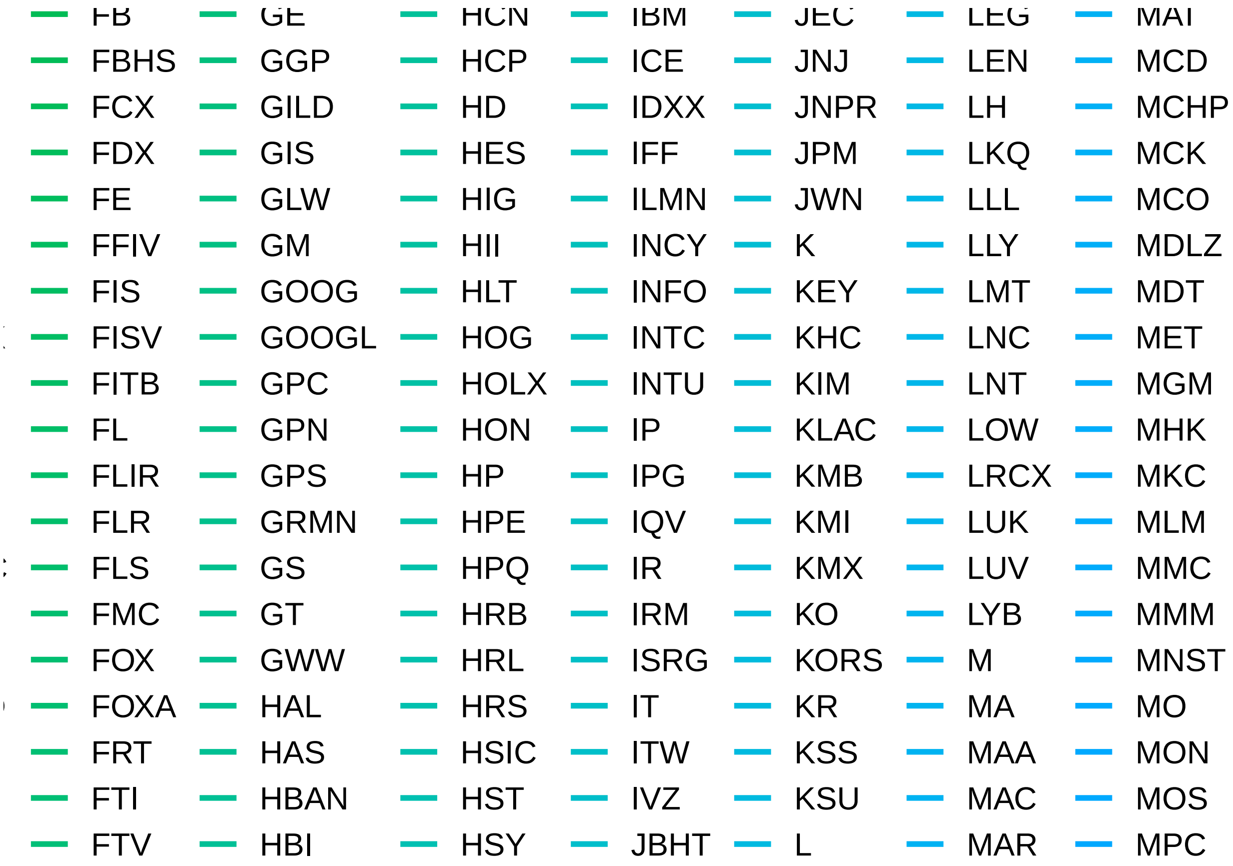
```r
# Calculate average returns per quarter
quarterly_returns <- fiveyearsap %>%
  group_by(quarter) %>%
  summarize(avg_log_return = mean(log_return, na.rm = TRUE))
ggplot(quarterly_returns, aes(x = quarter, y = avg_log_return)) +
  geom_line(color = "blue", size = 1) +
  scale_y_continuous(labels = number_format(accuracy = 0.0001)) + # Adjust decimal places
  labs(
    title = "Average Quarterly Log Returns",
    x = "Quarter",
    y = "Average Log Return"
  ) +
  theme_minimal()
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

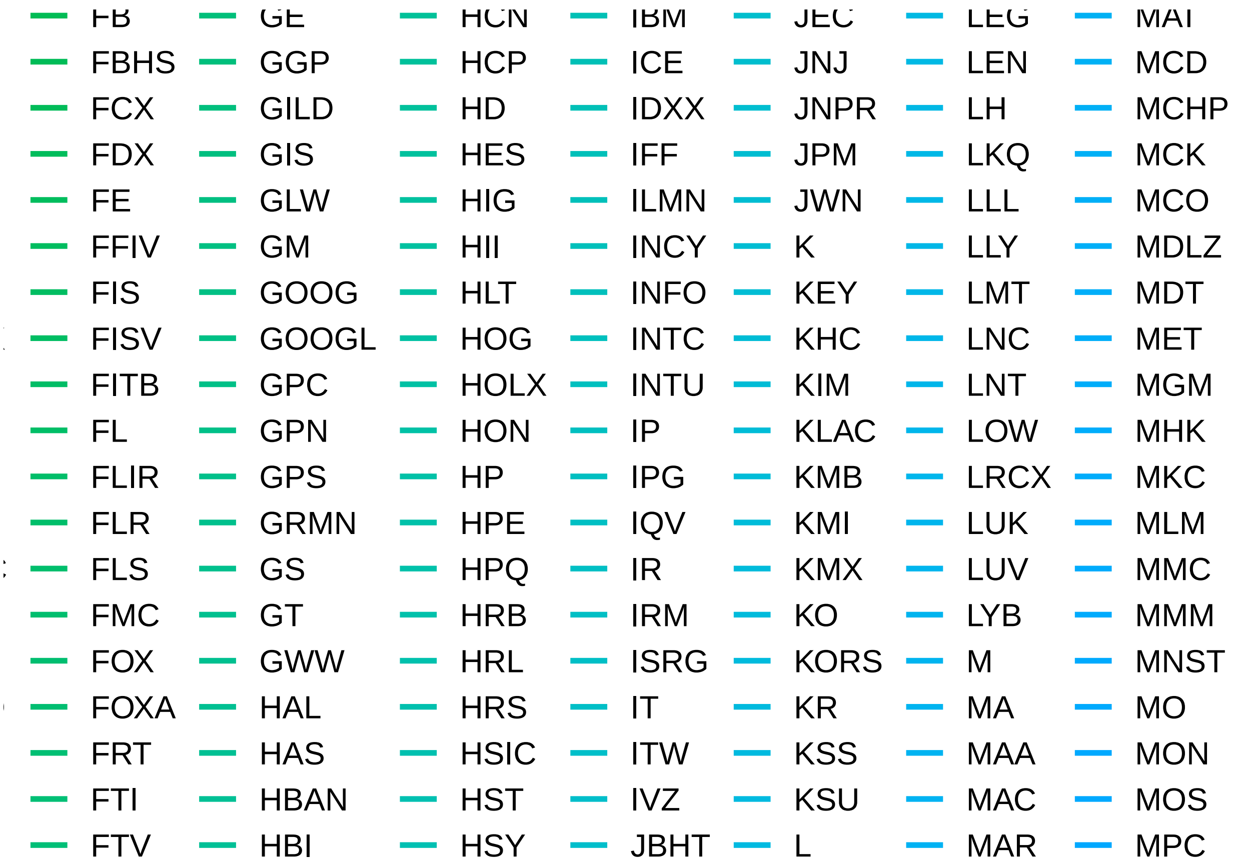


Top Volatile Stocks

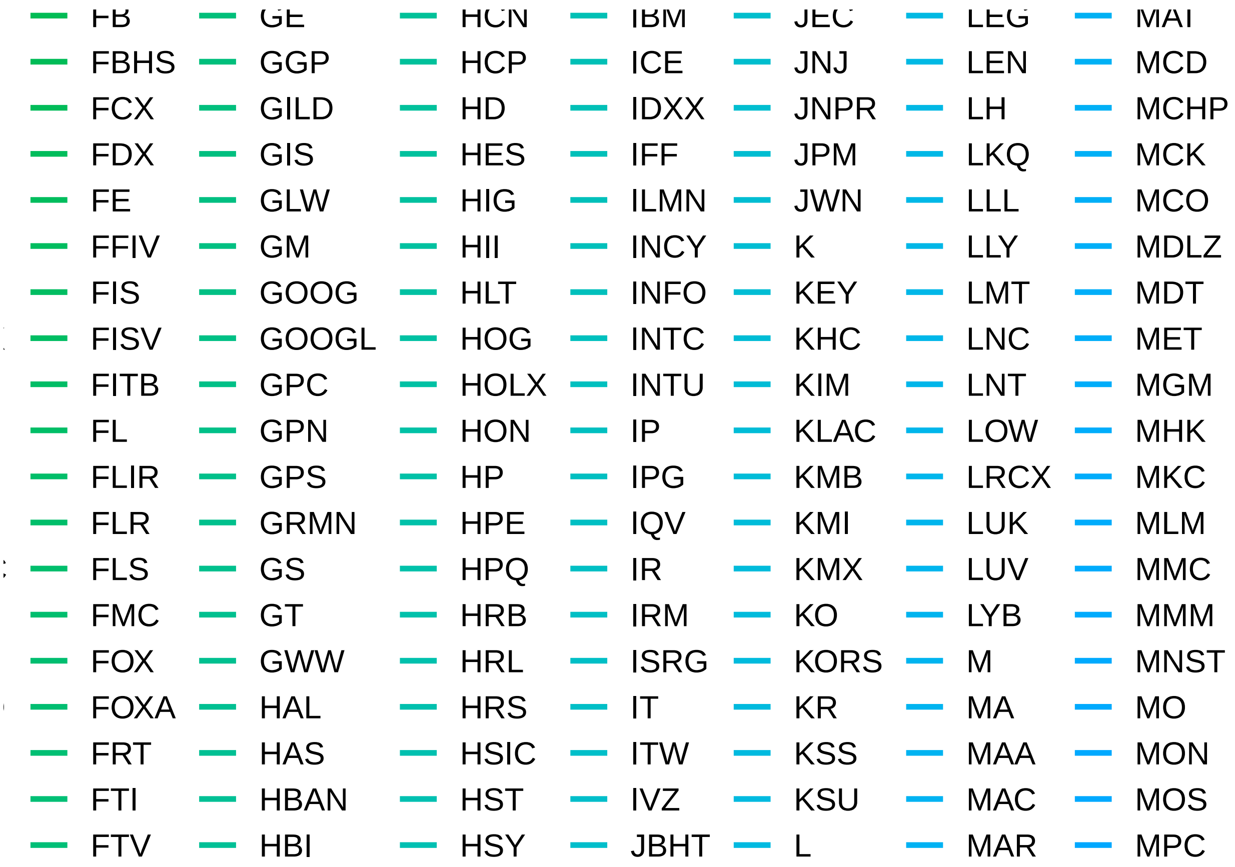
```
volatile_stocks <- fiveyearsap %>%
  group_by(Name) %>%
  summarize(volatility = sd(log_return, na.rm = TRUE)) %>%
  arrange(desc(volatility)) %>%
  pull(Name)

selected_data <- fiveyearsap %>% filter(Name %in% volatile_stocks)
ggplot(selected_data, aes(x = date, y = log_return, color = Name)) +
  geom_line(size = 1) +
  labs(
    title = "Daily Returns of 3 most Volatile S&P 500 Stocks (by sd of returns)",
    x = "Date",
    y = "Daily Log Return",
    color = "Stock"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.text = element_text(size = 12),
    axis.title = element_text(size = 14),
    legend.text = element_text(size = 12),
    legend.title = element_text(size = 14)
  )
```



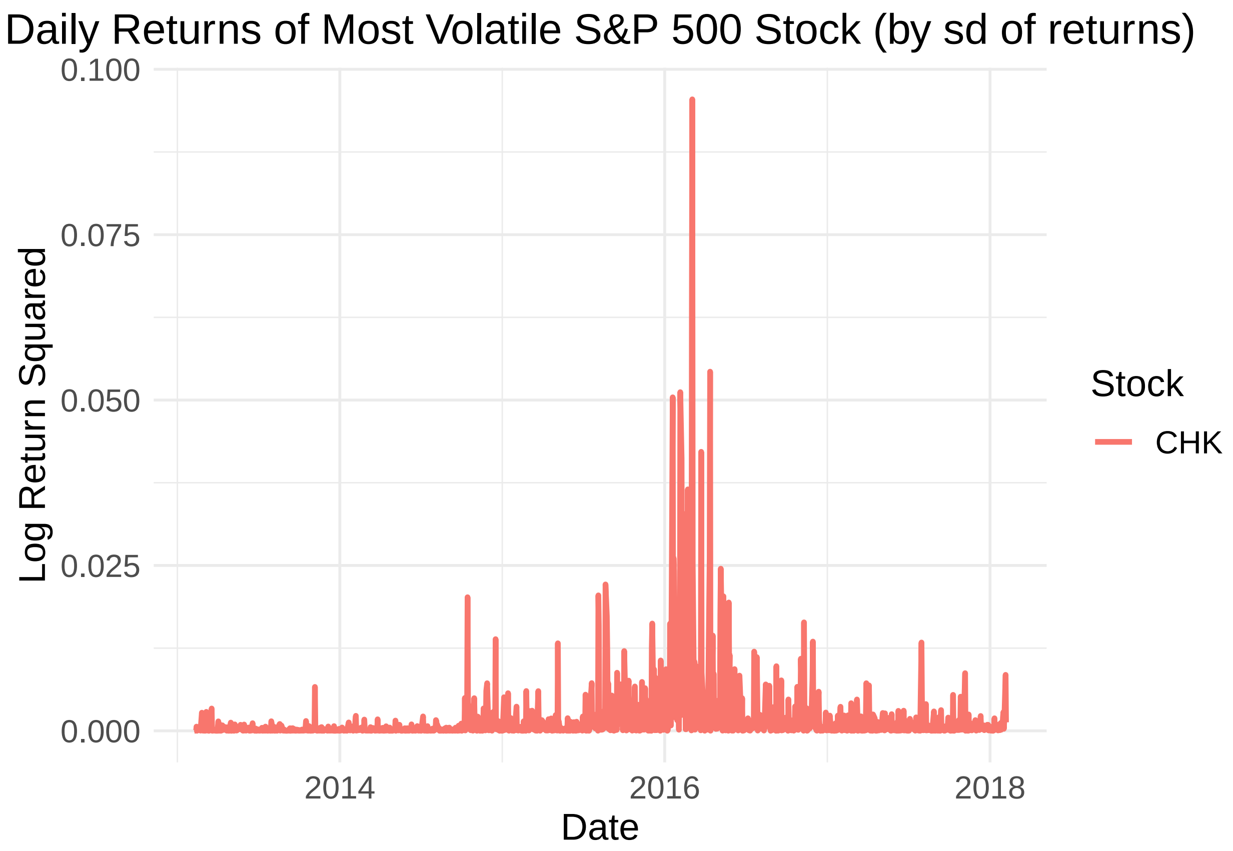
```r
ggplot(selected_data, aes(x = date, y = log_return^2, color = Name)) +
  geom_line(size = 1) +
  labs(
    title = "Daily Returns of 3 most Volatile S&P 500 Stocks (by sd of returns)",
    x = "Date",
    y = "Log Return Squared",
    color = "Stock"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.text = element_text(size = 12),
    axis.title = element_text(size = 14),
    legend.text = element_text(size = 12),
    legend.title = element_text(size = 14)
  )
```



```r
ggplot(selected_data, aes(x = date, y = log_return^2, color = Name)) +
  geom_line(size = 1) +
  labs(
    title = "Daily Returns of 3 most Volatile S&P 500 Stocks (by sd of returns)",
    x = "Date",
    y = "Log Return Squared",
    color = "Stock"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.text = element_text(size = 12),
    axis.title = element_text(size = 14),
    legend.text = element_text(size = 12),
    legend.title = element_text(size = 14)
  )
```



```r
stock_data <- selected_data %>% filter(Name == "CHK")

ggplot(stock_data, aes(x = date, y = log_return^2, color = Name)) +
  geom_line(size = 1) +
  labs(
    title = "Daily Returns of Most Volatile S&P 500 Stock (by sd of returns)",
    x = "Date",
    y = "Log Return Squared",
    color = "Stock"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.text = element_text(size = 12),
    axis.title = element_text(size = 14),
    legend.text = element_text(size = 12),
    legend.title = element_text(size = 14)
  )
```



```r
average_returns <- fiveyearsap %>%
  group_by(date) %>%
  summarise(avg_return = mean(log_return, na.rm = TRUE)) %>%
  ungroup()

lm_fit <- lm(avg_return ~ 1, data = average_returns)
residuals_squared <- residuals(lm_fit)^2
stock_data$lagged_squared <- lag(residuals_squared, 1)
arch_lm <- lm(residuals_squared ~ lagged_squared, data = stock_data, na.action = na.exclude)
summary(arch_lm)
```

```
##
## Call:
## lm(formula = residuals_squared ~ lagged_squared, data = stock_data,
##     na.action = na.exclude)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0003358 -0.0000444 -0.0000354 -0.0000041  0.0054024
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.246e-05  5.828e-06   7.285 5.68e-13 ***
## lagged_squared 3.250e-01  2.682e-02  12.121  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.000198 on 1255 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.1048, Adjusted R-squared:  0.1041
## F-statistic: 146.9 on 1 and 1255 DF,  p-value: < 2.2e-16
```

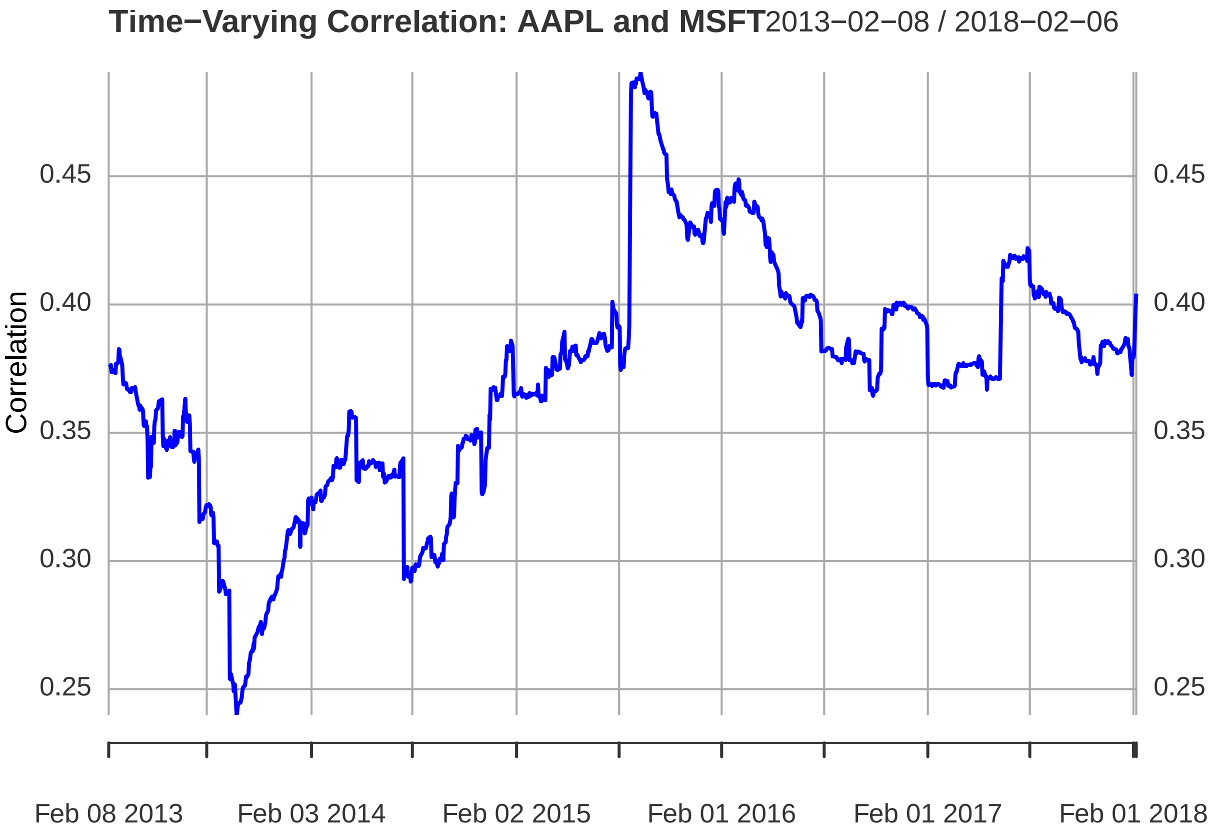
```
stock_data <- selected_data %>% filter(Name == "CHK")

lm_fit <- lm(log_return ~ 1, data = stock_data)
residuals_squared <- residuals(lm_fit)^2
stock_data$lagged_squared <- lag(residuals_squared, 1)
arch_lm <- lm(residuals_squared ~ lagged_squared, data = stock_data, na.action = na.exclude)
```

```
stock_data <- selected_data %>% filter(Name == "CHK")
lm_fit <- lm(log_return ~ 1, data = stock_data)
residuals_squared <- residuals(lm_fit)^2
stock_data$lagged_squared <- lag(residuals_squared, 1)
spec <- ugarchspec(variance.model = list(model = "sGARCH"),
                   mean.model = list(armaOrder = c(0, 0)))
garch_fit <- ugarchfit(spec, data = stock_data$log_return, solver = "hybrid")
coefficients <- coef(garch_fit)  # Coefficients
standard_errors <- garch_fit@fit$se.coef  # Standard errors
z_values <- coefficients / standard_errors  # Z-values
p_values <- 2 * (1 - pnorm(abs(z_values)))  # P-values
garch_results <- data.frame(
  Parameter = names(coefficients),
  Coefficient = round(coefficients, 4),
  `Std. Error` = round(standard_errors, 4),
  `z-Value` = round(z_values, 4),
  `p-Value` = round(p_values, 4)
)

spec <- ugarchspec(variance.model = list(model = "sGARCH"),
                   mean.model = list(armaOrder = c(0, 0)))
garch_fit <- ugarchfit(spec, data = average_returns$avg_return, solver = "hybrid")
coefficients <- coef(garch_fit)  # Coefficients
standard_errors <- garch_fit@fit$se.coef  # Standard errors
z_values <- coefficients / standard_errors  # Z-values
p_values <- 2 * (1 - pnorm(abs(z_values)))  # P-values
garch_results <- data.frame(
  Parameter = names(coefficients),
  Coefficient = round(coefficients, 4),
  `Std. Error` = round(standard_errors, 4),
  `z-Value` = round(z_values, 4),
  `p-Value` = round(p_values, 4)
)
```

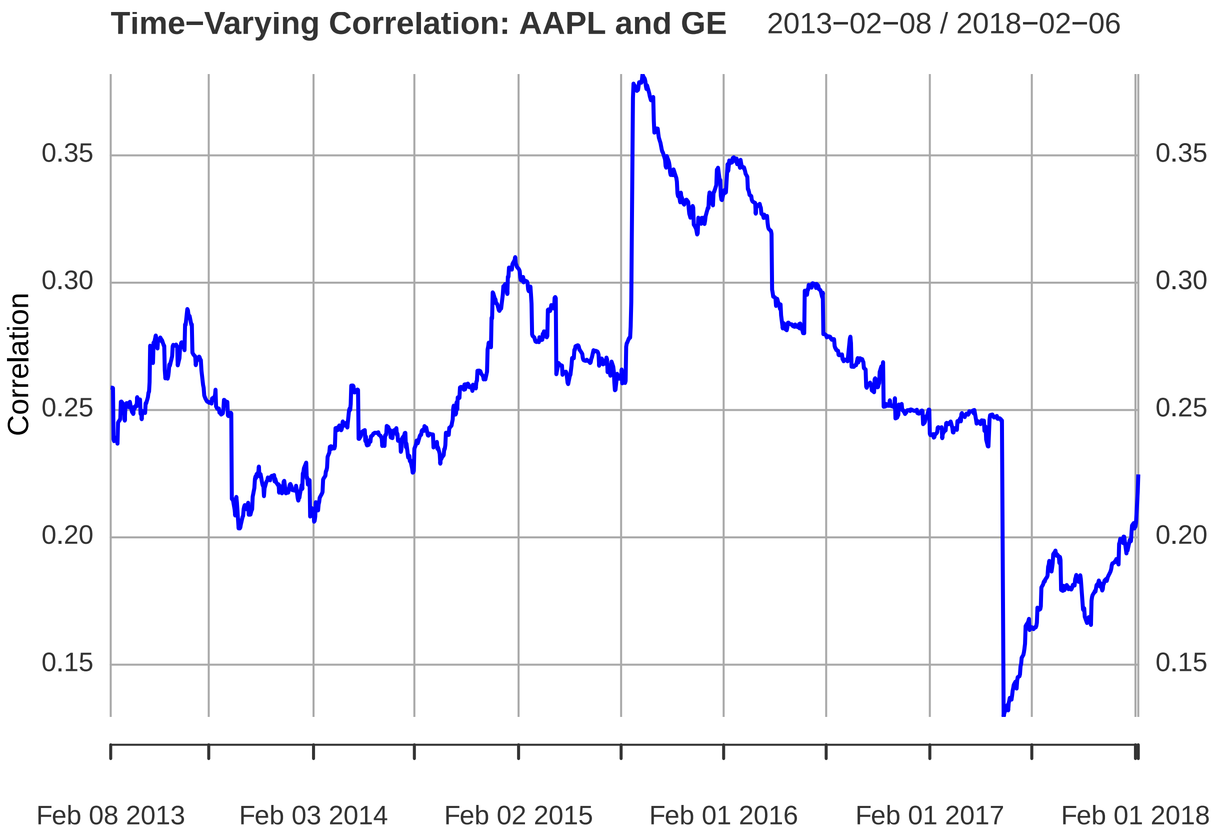
```
selected_stocks <- c("BAC", "AAPL", "GE", "F", "FB", "MSFT", "AMD", "MU", "INTC", "CSCO")
top10 <- fiveyearsap %>%
  filter(Name %in% selected_stocks)
returns_wide <- top10 %>%
  dplyr::select(date, Name, log_return) %>%
  pivot_wider(names_from = Name, values_from = log_return)
returns_matrix <- as.matrix(returns_wide[, -1])
colnames(returns_matrix) <- names(returns_wide)[-1]
uspec <- multispec(replicate(ncol(returns_matrix), ugarchspec(mean.model = list(armaOrder = c(1, 0)))))
multifit_results <- multifit(uspec, returns_matrix)
dcc_spec <- dccspec(uspec = uspec, dccOrder = c(1, 1), distribution = "mvnorm")
dcc_fit <- dccfit(dcc_spec, data = returns_matrix, fit.control = list(eval.se = TRUE), fit = multifit_r
coefficients <- coef(dcc_fit)
cov_matrices <- rcov(dcc_fit)
cor_matrices <- rcor(dcc_fit)
summary(dcc_fit)
```

```
## Length   Class    Mode
##      1 DCCfit      S4
```

```
cor_aapl_msft <- cor_matrices["AAPL", "MSFT", ]
cor_aapl_GE <- cor_matrices["AAPL", "GE", ]
cor_xts <- xts(cor_aapl_msft, order.by = returns_wide$date)
plot(cor_xts, main = "Time-Varying Correlation: AAPL and MSFT", ylab = "Correlation", col = "blue", typ
```



```
cor_xts <- xts(cor_aapl_GE, order.by = returns_wide$date)
plot(cor_xts, main = "Time-Varying Correlation: AAPL and GE", ylab = "Correlation", col = "blue", type
```



```
dcc_forecast <- dccforecast(dcc_fit, n.ahead = 10)
dcc_forecast
```

```
##
## *---------------------------------*
## *       DCC GARCH Forecast        *
## *---------------------------------*
##
## Distribution         :  mvnorm
## Model                :  DCC(1,1)
## Horizon              :  10
## Roll Steps           :  0
## -----------------------------------
##
## 0-roll forecast:
##
## First 2 Correlation Forecasts
## , , 1
##
##         [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]  [,10]
##  [1,] 1.0000 0.2578 0.3398 0.3249 0.3193 0.4113 0.2314 0.3185 0.4154 0.3359
##  [2,] 0.2578 1.0000 0.2476 0.2241 0.2213 0.2487 0.1853 0.2286 0.2817 0.3718
##  [3,] 0.3398 0.2476 1.0000 0.4232 0.4696 0.2552 0.3658 0.3055 0.3294 0.2613
##  [4,] 0.3249 0.2241 0.4232 1.0000 0.3916 0.2875 0.3360 0.4151 0.4173 0.2824
##  [5,] 0.3193 0.2213 0.4696 0.3916 1.0000 0.2362 0.3839 0.2884 0.3139 0.2729
##  [6,] 0.4113 0.2487 0.2552 0.2875 0.2362 1.0000 0.1814 0.2480 0.4065 0.3313
##  [7,] 0.2314 0.1853 0.3658 0.3360 0.3839 0.1814 1.0000 0.1961 0.2383 0.1782
##  [8,] 0.3185 0.2286 0.3055 0.4151 0.2884 0.2480 0.1961 1.0000 0.4810 0.2941
##  [9,] 0.4154 0.2817 0.3294 0.4173 0.3139 0.4065 0.2383 0.4810 1.0000 0.2965
## [10,] 0.3359 0.3718 0.2613 0.2824 0.2729 0.3313 0.1782 0.2941 0.2965 1.0000
##
## , , 2
##
##         [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]  [,10]
##  [1,] 1.0000 0.2574 0.3396 0.3249 0.3192 0.4111 0.2317 0.3188 0.4150 0.3355
##  [2,] 0.2574 1.0000 0.2477 0.2239 0.2213 0.2482 0.1852 0.2290 0.2812 0.3714
##  [3,] 0.3396 0.2477 1.0000 0.4228 0.4695 0.2552 0.3664 0.3057 0.3292 0.2617
##  [4,] 0.3249 0.2239 0.4228 1.0000 0.3914 0.2873 0.3362 0.4153 0.4172 0.2827
##  [5,] 0.3192 0.2213 0.4695 0.3914 1.0000 0.2362 0.3841 0.2889 0.3138 0.2732
##  [6,] 0.4111 0.2482 0.2552 0.2873 0.2362 1.0000 0.1817 0.2483 0.4061 0.3308
##  [7,] 0.2317 0.1852 0.3664 0.3362 0.3841 0.1817 1.0000 0.1972 0.2387 0.1787
##  [8,] 0.3188 0.2290 0.3057 0.4153 0.2889 0.2483 0.1972 1.0000 0.4809 0.2947
##  [9,] 0.4150 0.2812 0.3292 0.4172 0.3138 0.4061 0.2387 0.4809 1.0000 0.2963
## [10,] 0.3355 0.3714 0.2617 0.2827 0.2732 0.3308 0.1787 0.2947 0.2963 1.0000
##
## . . . . . . . . . .
## . . . . . . . . . .
##
## Last 2 Correlation Forecasts
## , , 1
##
##         [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]  [,10]
##  [1,] 1.0000 0.2549 0.3385 0.3251 0.3188 0.4100 0.2333 0.3208 0.4124 0.3330
##  [2,] 0.2549 1.0000 0.2482 0.2228 0.2209 0.2444 0.1844 0.2316 0.2776 0.3691
##  [3,] 0.3385 0.2482 1.0000 0.4198 0.4686 0.2553 0.3701 0.3070 0.3280 0.2645
##  [4,] 0.3251 0.2228 0.4198 1.0000 0.3904 0.2864 0.3379 0.4168 0.4164 0.2842
##  [5,] 0.3188 0.2209 0.4686 0.3904 1.0000 0.2359 0.3857 0.2922 0.3129 0.2746
##  [6,] 0.4100 0.2444 0.2553 0.2864 0.2359 1.0000 0.1837 0.2501 0.4031 0.3271
##  [7,] 0.2333 0.1844 0.3701 0.3379 0.3857 0.1837 1.0000 0.2044 0.2421 0.1816
##  [8,] 0.3208 0.2316 0.3070 0.4168 0.2922 0.2501 0.2044 1.0000 0.4803 0.2993
##  [9,] 0.4124 0.2776 0.3280 0.4164 0.3129 0.4031 0.2421 0.4803 1.0000 0.2952
## [10,] 0.3330 0.3691 0.2645 0.2842 0.2746 0.3271 0.1816 0.2993 0.2952 1.0000
##
## , , 2
##
##         [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]  [,10]
##  [1,] 1.0000 0.2545 0.3384 0.3252 0.3187 0.4098 0.2335 0.3211 0.4120 0.3326
##  [2,] 0.2545 1.0000 0.2482 0.2226 0.2209 0.2439 0.1842 0.2320 0.2771 0.3688
##  [3,] 0.3384 0.2482 1.0000 0.4193 0.4684 0.2553 0.3707 0.3072 0.3279 0.2649
##  [4,] 0.3252 0.2226 0.4193 1.0000 0.3903 0.2863 0.3381 0.4170 0.4163 0.2844
##  [5,] 0.3187 0.2209 0.4684 0.3903 1.0000 0.2359 0.3860 0.2926 0.3127 0.2748
##  [6,] 0.4098 0.2439 0.2553 0.2863 0.2359 1.0000 0.1840 0.2503 0.4027 0.3266
##  [7,] 0.2335 0.1842 0.3707 0.3381 0.3860 0.1840 1.0000 0.2054 0.2425 0.1820
##  [8,] 0.3211 0.2320 0.3072 0.4170 0.2926 0.2503 0.2054 1.0000 0.4802 0.3000
##  [9,] 0.4120 0.2771 0.3279 0.4163 0.3127 0.4027 0.2425 0.4802 1.0000 0.2950
## [10,] 0.3326 0.3688 0.2649 0.2844 0.2748 0.3266 0.1820 0.3000 0.2950 1.0000
```

```r
forecasted_cor <- dcc_forecast@mforecast$R
```