Risk-Management-R

October 26, 2023

[1]:
```
options(warn = -1)
```

0.1 Setting the random seed and picking the starting year

[42]:
```
set.seed(903948185)
random_year_with_seed <- sample(1980:2001, 1)
random_year_with_seed
```

2000

[3]:
```
# Load the CSV file into an R dataframe
dsf <- read.csv("dsf_jan_2000_dec_2010.csv")

# Display the first few rows for a quick check
head(dsf)
```

A data.frame: 6 × 5

	X <int>	PERMNO <int>	date <chr>	RET <dbl>	vwretd <dbl>
1	0	10056	2000-01-03	-0.028846	-0.006803
2	1	10056	2000-01-04	-0.029703	-0.039652
3	2	10056	2000-01-05	-0.005102	-0.000935
4	3	10056	2000-01-06	-0.015385	-0.007391
5	4	10056	2000-01-07	0.000000	0.032516
6	5	10056	2000-01-10	0.046875	0.018608

[4]:
```
# Check for NAs in each column of dsf and print the count
na_counts <- sapply(dsf, function(x) sum(is.na(x)))
print(na_counts)
```

```
     X PERMNO   date    RET vwretd
     0      0      0      0      0
```

[5]:
```
# Convert the Date column in dsf into a proper date format
dsf$date <- as.Date(dsf$date, format="%Y-%m-%d")

# Display the first few rows for verification
head(dsf)
```

A data.frame: 6 × 5

	X	PERMNO	date	RET	vwretd
	<int>	<int>	<date>	<dbl>	<dbl>
1	0	10056	2000-01-03	-0.028846	-0.006803
2	1	10056	2000-01-04	-0.029703	-0.039652
3	2	10056	2000-01-05	-0.005102	-0.000935
4	3	10056	2000-01-06	-0.015385	-0.007391
5	4	10056	2000-01-07	0.000000	0.032516
6	5	10056	2000-01-10	0.046875	0.018608

0.2 Reading the Rf from Ken French's Dataset

[6]:
```
# Load the CSV file into an R dataframe
french_data <- read.csv("F-F_Research_Data_Factors_daily.csv")

# Display the first few rows for a quick check
head(french_data)
```

A data.frame: 6 × 5

	Date	Mkt.RF	SMB	HML	RF
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	19260701	0.10	-0.25	-0.27	0.009
2	19260702	0.45	-0.33	-0.06	0.009
3	19260706	0.17	0.30	-0.39	0.009
4	19260707	0.09	-0.58	0.02	0.009
5	19260708	0.21	-0.38	0.19	0.009
6	19260709	-0.71	0.43	0.57	0.009

[7]:
```
# Convert the Date column into a proper date format
french_data$Date <- as.Date(as.character(french_data$Date), format="%Y%m%d")

# Display the first few rows for verification
head(french_data)
```

A data.frame: 6 × 5

	Date	Mkt.RF	SMB	HML	RF
	<date>	<dbl>	<dbl>	<dbl>	<dbl>
1	1926-07-01	0.10	-0.25	-0.27	0.009
2	1926-07-02	0.45	-0.33	-0.06	0.009
3	1926-07-06	0.17	0.30	-0.39	0.009
4	1926-07-07	0.09	-0.58	0.02	0.009
5	1926-07-08	0.21	-0.38	0.19	0.009
6	1926-07-09	-0.71	0.43	0.57	0.009

[8]:
```
# Check the class of the date column in dsf
dsf_date_format <- class(dsf$date)

# Check the class of the Date column in french_data
french_date_format <- class(french_data$Date)

# Print the results
```

```
cat("Date format in dsf:", dsf_date_format, "\n")
cat("Date format in french_data:", french_date_format, "\n")

# Check if they are the same format
if (dsf_date_format == french_date_format) {
  cat("The date formats in dsf and french_data are the same.\n")
} else {
  cat("The date formats in dsf and french_data are different.\n")
}
```

```
Date format in dsf: Date
Date format in french_data: Date
The date formats in dsf and french_data are the same.
```

[9]:
```
# Inner join
merged_data_inner <- merge(dsf, french_data, by.x = "date", by.y = "Date")

# Left join
merged_data_left <- merge(dsf, french_data, by.x = "date", by.y = "Date", all.x␣
↪= TRUE)

# Compare the number of rows in the two merged datasets
num_rows_inner <- nrow(merged_data_inner)
num_rows_left <- nrow(merged_data_left)

cat("Number of rows in inner join:", num_rows_inner, "\n")
cat("Number of rows in left join:", num_rows_left, "\n")

# Check if they are the same
if (num_rows_inner == num_rows_left) {
  cat("The inner and left joins produced the same number of rows.\n")
} else {
  cat("The inner and left joins produced different numbers of rows.\n")
}
```

```
Number of rows in inner join: 392735
Number of rows in left join: 392735
The inner and left joins produced the same number of rows.
```

[10]:
```
# Merge dsf and french_data on the date columns
merged_data <- merge(dsf, french_data, by.x = "date", by.y = "Date")

# Display the first few rows for verification
head(merged_data)
```

A data.frame: 6 × 9

	date	X	PERMNO	RET	vwretd	Mkt.RF	SMB	HML
	<date>	<int>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	2000-01-03	0	10056	-0.028846	-0.006803	-0.71	0.62	-1.41
2	2000-01-03	70923	52695	-0.039216	-0.006803	-0.71	0.62	-1.41
3	2000-01-03	91341	60354	0.148649	-0.006803	-0.71	0.62	-1.41
4	2000-01-03	162550	77222	-0.035088	-0.006803	-0.71	0.62	-1.41
5	2000-01-03	52332	41823	0.112500	-0.006803	-0.71	0.62	-1.41
6	2000-01-03	279332	85207	-0.016667	-0.006803	-0.71	0.62	-1.41

[11]:
```
merged_data$RET <- as.numeric(as.character(merged_data$RET))
# Display the first few rows for verification
head(merged_data)
```

A data.frame: 6 × 9

	date	X	PERMNO	RET	vwretd	Mkt.RF	SMB	HML
	<date>	<int>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	2000-01-03	0	10056	-0.028846	-0.006803	-0.71	0.62	-1.41
2	2000-01-03	70923	52695	-0.039216	-0.006803	-0.71	0.62	-1.41
3	2000-01-03	91341	60354	0.148649	-0.006803	-0.71	0.62	-1.41
4	2000-01-03	162550	77222	-0.035088	-0.006803	-0.71	0.62	-1.41
5	2000-01-03	52332	41823	0.112500	-0.006803	-0.71	0.62	-1.41
6	2000-01-03	279332	85207	-0.016667	-0.006803	-0.71	0.62	-1.41

0.3 Semi Beta, Downside Beta, Co-skewness and Tail Risk

[12]:
```
# Compute the signed intra-period returns for each firm
merged_data$r_plus <- pmax(merged_data$RET, 0)
merged_data$r_minus <- pmin(merged_data$RET, 0)

# Compute the signed intra-period returns for the market
merged_data$f_plus <- pmax(merged_data$Mkt.RF, 0)
merged_data$f_minus <- pmin(merged_data$Mkt.RF, 0)

# Display the first few rows to verify
head(merged_data)
```

A data.frame: 6 × 13

	date	X	PERMNO	RET	vwretd	Mkt.RF	SMB	HML
	<date>	<int>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	2000-01-03	0	10056	-0.028846	-0.006803	-0.71	0.62	-1.41
2	2000-01-03	70923	52695	-0.039216	-0.006803	-0.71	0.62	-1.41
3	2000-01-03	91341	60354	0.148649	-0.006803	-0.71	0.62	-1.41
4	2000-01-03	162550	77222	-0.035088	-0.006803	-0.71	0.62	-1.41
5	2000-01-03	52332	41823	0.112500	-0.006803	-0.71	0.62	-1.41
6	2000-01-03	279332	85207	-0.016667	-0.006803	-0.71	0.62	-1.41

[13]:
```
# Convert the columns to numeric
merged_data$r_plus <- as.numeric(as.character(merged_data$r_plus))
merged_data$r_minus <- as.numeric(as.character(merged_data$r_minus))
```

```
# Compute the semi-betas again
f_squared_sum <- sum(merged_data$Mkt.RF^2)
merged_data$beta_N <- (merged_data$r_minus * merged_data$f_minus) /
 ↪f_squared_sum
merged_data$beta_P <- (merged_data$r_plus * merged_data$f_plus) / f_squared_sum
merged_data$beta_M_minus <- -(merged_data$r_plus * merged_data$f_minus) /
 ↪f_squared_sum
merged_data$beta_M_plus <- -(merged_data$r_minus * merged_data$f_plus) /
 ↪f_squared_sum

# Display the first few rows to verify
head(merged_data)
```

A data.frame: 6 × 17

	date	X	PERMNO	RET	vwretd	Mkt.RF	SMB	HML
	<date>	<int>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	2000-01-03	0	10056	-0.028846	-0.006803	-0.71	0.62	-1.41
2	2000-01-03	70923	52695	-0.039216	-0.006803	-0.71	0.62	-1.41
3	2000-01-03	91341	60354	0.148649	-0.006803	-0.71	0.62	-1.41
4	2000-01-03	162550	77222	-0.035088	-0.006803	-0.71	0.62	-1.41
5	2000-01-03	52332	41823	0.112500	-0.006803	-0.71	0.62	-1.41
6	2000-01-03	279332	85207	-0.016667	-0.006803	-0.71	0.62	-1.41

[14]:
```
# Compute beta
merged_data$beta <- (merged_data$RET * merged_data$Mkt.RF) / f_squared_sum

# Compute downbeta
f_minus_squared_sum <- sum(merged_data$f_minus^2)
merged_data$downbeta <- (merged_data$RET * merged_data$f_minus) /
 ↪f_minus_squared_sum

# Compute upbeta
f_plus_squared_sum <- sum(merged_data$f_plus^2)
merged_data$upbeta <- (merged_data$RET * merged_data$f_plus) /
 ↪f_plus_squared_sum

# Display the first few rows to verify
head(merged_data)
```

A data.frame: 6 × 20

	date	X	PERMNO	RET	vwretd	Mkt.RF	SMB	HML
	<date>	<int>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	2000-01-03	0	10056	-0.028846	-0.006803	-0.71	0.62	-1.41
2	2000-01-03	70923	52695	-0.039216	-0.006803	-0.71	0.62	-1.41
3	2000-01-03	91341	60354	0.148649	-0.006803	-0.71	0.62	-1.41
4	2000-01-03	162550	77222	-0.035088	-0.006803	-0.71	0.62	-1.41
5	2000-01-03	52332	41823	0.112500	-0.006803	-0.71	0.62	-1.41
6	2000-01-03	279332	85207	-0.016667	-0.006803	-0.71	0.62	-1.41

[15]:

```r
# 1. Compute the products
merged_data$r_times_f2 <- merged_data$RET * merged_data$Mkt.RF^2
merged_data$r_times_f3 <- merged_data$RET * merged_data$Mkt.RF^3

# 2. Sum these products for each i (using aggregate function)
sum_r_times_f2 <- aggregate(r_times_f2 ~ PERMNO, data = merged_data, sum)
sum_r_times_f3 <- aggregate(r_times_f3 ~ PERMNO, data = merged_data, sum)

# 3. Compute the denominator parts
r_squared_sum <- aggregate(RET^2 ~ PERMNO, data = merged_data, sum)
f_squared_sum <- sum(merged_data$Mkt.RF^2) / nrow(merged_data)

denom_coskew <- sqrt(r_squared_sum$RET^2 * f_squared_sum)
denom_cokurt <- sqrt(r_squared_sum$RET^2 * f_squared_sum^(3/2))

# 4. Combine to get the final coskewness and cokurtosis values
coskewness <- sum_r_times_f2$r_times_f2 / (nrow(merged_data) * denom_coskew)
cokurtosis <- sum_r_times_f3$r_times_f3 / (nrow(merged_data) * denom_cokurt)

# Combine the results
results <- data.frame(PERMNO = sum_r_times_f2$PERMNO, coskewness, cokurtosis)

# Display the results
head(results)
```

A data.frame: 6 × 3

	PERMNO <int>	coskewness <dbl>	cokurtosis <dbl>
1	10056	5.457246e-06	6.445211e-05
2	10100	8.475469e-06	3.481581e-05
3	10307	-5.431912e-06	1.287718e-05
4	10456	4.000440e-05	4.419144e-04
5	10517	-1.739406e-05	2.948554e-04
6	10745	-7.817798e-06	6.299136e-06

0.4 Data Distribution of Co-Skewness and Co-Kurtosis

Co-skewness:

- The average co-skewness value is approximately 4×10-64×10-6, which is very close to zero. This indicates that, on average, the stocks in this sample have a very weak positive skew with respect to the market.
- The standard deviation is 4.5×10-54.5×10-5, indicating that the values are generally close to the mean.
- Some stocks exhibit significant positive or negative co-skewness, as seen from the max and min values.
- 30 stocks have been identified as outliers for co-skewness. These are stocks whose co-skewness values deviate significantly from the majority of stocks in the sample.
- The top stocks with the highest positive co-skewness may represent higher potential returns during market upswings. Conversely, those with negative values might perform better during

market downturns.

Co-kurtosis:

- The average co-kurtosis is approximately 4.08×10-44.08×10-4, suggesting that the returns of the stocks in this sample have tails that are similar to the market's.
- The standard deviation is 7.41×10-47.41×10-4, indicating a bit more variability around the mean compared to co-skewness.
- The max value is significantly higher than the 75th percentile, suggesting a few stocks have very high co-kurtosis values.
- 29 stocks have been identified as outliers for co-kurtosis. These stocks have co-kurtosis values that significantly deviate from the majority.
- High positive co-kurtosis indicates that the stock may experience extreme returns (either up or down) more frequently than the market. Conversely, negative co-kurtosis suggests less frequent extreme returns compared to the market.

0.5 Value-at-Risk (VaR) and Expected Shortfall (ES)

[16]:
```r
# Load the CSV file into an R dataframe
dsf <- read.csv("dsf_jan_2000_dec_2010.csv")
# Convert the Date column in dsf into a proper date format
dsf$date <- as.Date(dsf$date, format="%Y-%m-%d")
#Convert the RET column in dsf to numeric
dsf$RET <- as.numeric(as.character(dsf$RET))
head(dsf)
```

A data.frame: 6 × 5

	X <int>	PERMNO <int>	date <date>	RET <dbl>	vwretd <dbl>
1	0	10056	2000-01-03	-0.028846	-0.006803
2	1	10056	2000-01-04	-0.029703	-0.039652
3	2	10056	2000-01-05	-0.005102	-0.000935
4	3	10056	2000-01-06	-0.015385	-0.007391
5	4	10056	2000-01-07	0.000000	0.032516
6	5	10056	2000-01-10	0.046875	0.018608

0.6 Var and ES Computation part -1

[17]:
```r
# 1. Portfolio Construction: You've already invested $1,000,000 in each stock.

# 2. Calculate Daily Portfolio Returns
dsf$weight <- 1/250  # Equal weight for each stock since you invest $1,000,000␣
↪in each
dsf$weighted_return <- dsf$RET * dsf$weight
portfolio_returns <- aggregate(weighted_return ~ date, data = dsf,␣
↪sum)$weighted_return

# 3. Compute VaR and ES
# VaR at 5% level
```

```
VaR <- quantile(portfolio_returns, probs = 0.05)

# Expected Shortfall (ES)
ES <- mean(portfolio_returns[portfolio_returns < VaR])

# $VaR for the portfolio
dollar_VaR <- 250000000 * VaR  # Total portfolio value is $250,000,000

list(VaR = VaR, dollar_VaR = dollar_VaR, ES = ES)
```

$VaR 5\%: -0.0101899004

$dollar_VaR 5\%: -2547475.1

$ES -0.0154725723741007

0.7 Var and ES Computation part -2

[18]:
```
# Load the CSV file into an R dataframe
dsf_2001_2011 <- read.csv("dsf_jan_2001_dec_2011.csv")
# Convert the Date column in dsf into a proper date format
dsf_2001_2011$date <- as.Date(dsf_2001_2011$date, format="%Y-%m-%d")
#Convert the RET column in dsf to numeric
dsf_2001_2011$RET <- as.numeric(as.character(dsf_2001_2011$RET))
```

[19]:
```
na_counts <- sapply(dsf_2001_2011, function(x) sum(is.na(x)))
print(na_counts)
```

```
     X PERMNO    date     RET vwretd
     0      0       0       0      0
```

[20]:
```
# Remove rows with NA in the RET column
dsf_2001_2011_clean <- dsf_2001_2011[!is.na(dsf_2001_2011$RET), ]
```

[21]:
```
na_counts <- sapply(dsf_2001_2011_clean, function(x) sum(is.na(x)))
print(na_counts)
```

```
     X PERMNO    date     RET vwretd
     0      0       0       0      0
```

[22]:
```
# Compute portfolio returns for each day
dsf_2001_2011_clean$portfolio_return <- with(dsf_2001_2011_clean, (RET *
 ↪1000000) / 250000000)

# Compute VaR, $VaR, and ES for the 2001-2011 period

# VaR at 5% level
VaR_5_percent_2001_2011 <- quantile(dsf_2001_2011_clean$portfolio_return, 0.05)
```

```r
# Expected Shortfall at 5% level
ES_5_percent_2001_2011 <- mean(subset(dsf_2001_2011_clean$portfolio_return,
  dsf_2001_2011_clean$portfolio_return < VaR_5_percent_2001_2011))

# $VaR at 5% level
dollar_VaR_2001_2011 <- VaR_5_percent_2001_2011 * 250000000

list(VaR_5_percent_2001_2011 = VaR_5_percent_2001_2011,
     dollar_VaR_2001_2011 = dollar_VaR_2001_2011,
     ES_5_percent_2001_2011 = ES_5_percent_2001_2011)
```

$VaR__5__percent__2001__2011 5\%: -0.00020712

$dollar__VaR__2001__2011 5\%: -51780

$ES__5__percent__2001__2011 -0.000362794579221643

0.7.1 Analysis of Risk Measures:

Historical Distribution of Returns (2000-2010 vs. 2001-2011): The risk measures, specifically VaR and Expected Shortfall (ES), are influenced by the historical distribution of returns. This distribution reflects the past market conditions, economic events, and other financial indicators during a given period.

Value at Risk (VaR):

- 2000-2010: The one-day 5% VaR is -0.0101899004, implying that there is a 5% chance that the portfolio will experience a loss of 1.0189% or more on any given day.
- 2001-2011: The one-day 5% VaR is -0.00020712, suggesting a potential loss of only 0.0207% or more with a 5% probability on any given day.

The VaR in the second period (2001-2011) is significantly lower than in the first period (2000-2010), indicating that the portfolio's risk profile in terms of potential losses has reduced over time.

Dollar Value at Risk ($VaR):

- 2000-2010: The Dollar Value of VaR is -2,547,475.1, meaning there's a 5% chance the portfolio will lose more than 2.54 million in a single day.
- 2001-2011: The Dollar Value of VaR is -51,780, indicating a potential loss of 51,780 or more with a 5% probability on any given day.

The difference in dollar VaR between the two periods is substantial, further emphasizing the decreased risk in the latter period.

Expected Shortfall (ES):

- 2000-2010: The ES is -0.0154725723741007, or -1.5473%. This implies that, given the portfolio experiences losses exceeding the VaR, the expected loss is 1.5473%.
- 2001-2011: The ES is -0.000362794579221643, or -0.0363%, suggesting that if losses exceed VaR, the expected loss would be only 0.0363%.

The ES for the 2001-2011 period is also significantly lower than the 2000-2010 period, reinforcing the idea of diminished risk in the latter decade.

0.7.2 Conclusion:

From 2000-2010, the risk measures indicate a more volatile and uncertain market condition compared to 2001-2011. The substantial reduction in VaR, $VaR, and ES in the 2001-2011 period highlights a more stable financial environment, fewer extreme market events, or perhaps a more diversified or conservative investment approach.

This difference in risk profiles underscores the importance of regularly updating risk assessments and being aware of how changing market conditions, economic events, and other factors can impact the risk landscape.

0.8 Volatility modeling, JPMC Risk Model

[23]:
```r
# Load the CSV file into an R dataframe
dsf <- read.csv("dsf_jan_2000_dec_2010.csv")
# Convert the Date column in dsf into a proper date format
dsf$date <- as.Date(dsf$date, format="%Y-%m-%d")
#Convert the RET column in dsf to numeric
dsf$RET <- as.numeric(as.character(dsf$RET))
```

[24]:
```r
# Constants
lambda <- 0.94

# 1. Compute annualized variance for each firm as initial values
initial_variances <- aggregate(RET^2 ~ PERMNO, data = dsf, function(x) mean(x)␣
 ↪* 252)

# Rename the columns for clarity
colnames(initial_variances) <- c("PERMNO", "Initial_Variance")

# 2. Use the Risk Metrics model to forecast future variances

compute_risk_metrics_variance <- function(returns, initial_variance) {
    n <- length(returns)
    variances <- numeric(n)
    variances[1] <- initial_variance

    for (i in 2:n) {
        variances[i] <- lambda * variances[i-1] + (1 - lambda) * returns[i-1]^2
    }

    return(variances)
}
```

```
# Compute and store Risk Metrics variances for all firms
all_variances <- list()

for (permno in unique(dsf$PERMNO)) {
    firm_returns <- dsf$RET[dsf$PERMNO == permno]
    init_var <- initial_variances$Initial_Variance[initial_variances$PERMNO ==␣
 ↪permno]
    all_variances[[as.character(permno)]] <-␣
 ↪compute_risk_metrics_variance(firm_returns, init_var)
}

# The list all_variances now contains Risk Metrics variances for all firms.
```

[25]:
```
# Load the CSV file into an R dataframe
dsf_2001_2011 <- read.csv("dsf_jan_2001_dec_2011.csv")
# Convert the Date column in dsf into a proper date format
dsf_2001_2011$date <- as.Date(dsf_2001_2011$date, format="%Y-%m-%d")
#Convert the RET column in dsf to numeric
dsf_2001_2011$RET <- as.numeric(as.character(dsf_2001_2011$RET))
dsf_2001_2011 <- dsf_2001_2011[!is.na(dsf_2001_2011$RET),]
```

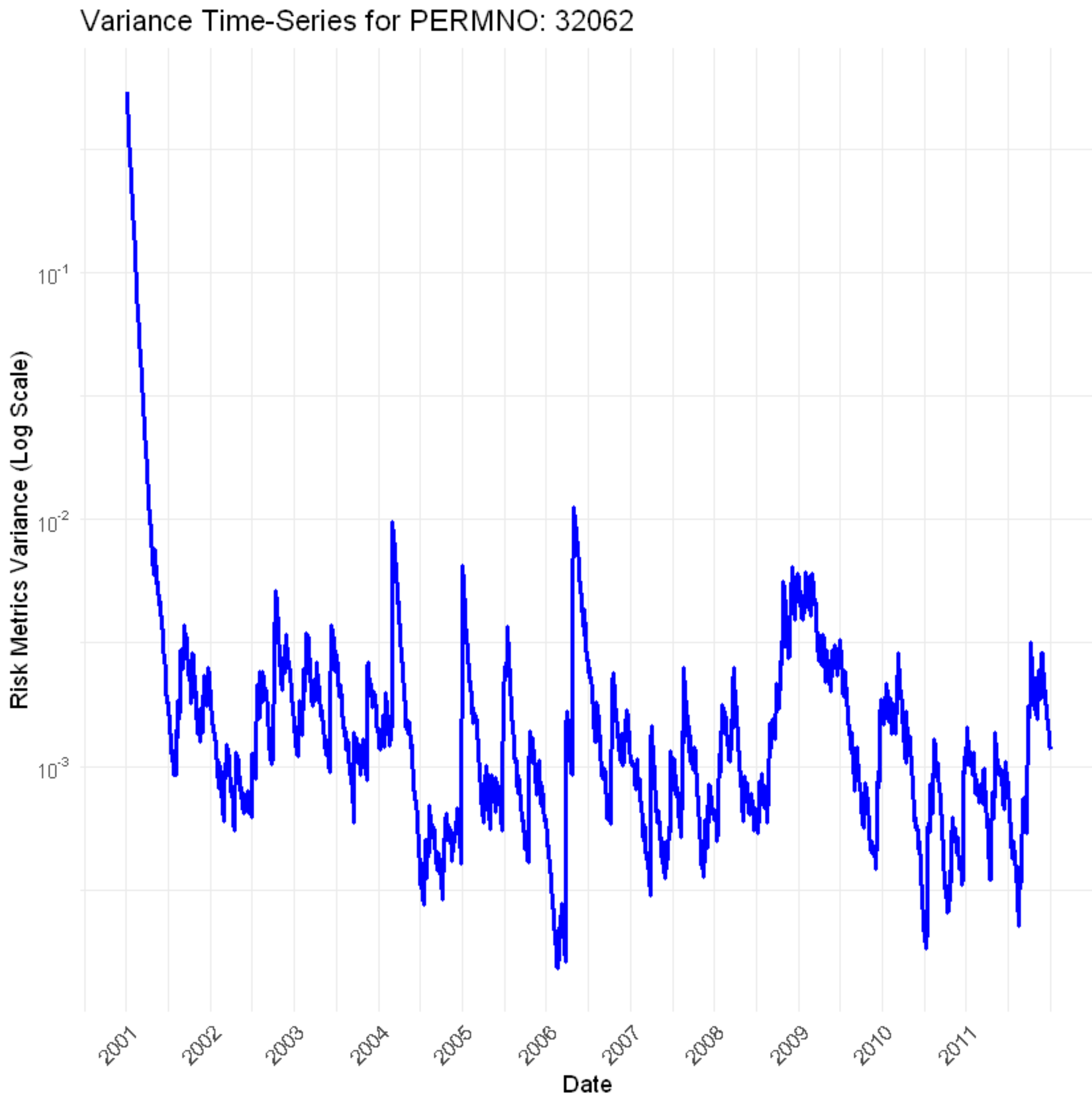
[26]:
```
na_counts <- sapply(dsf_2001_2011, function(x) sum(is.na(x)))
print(na_counts)
```

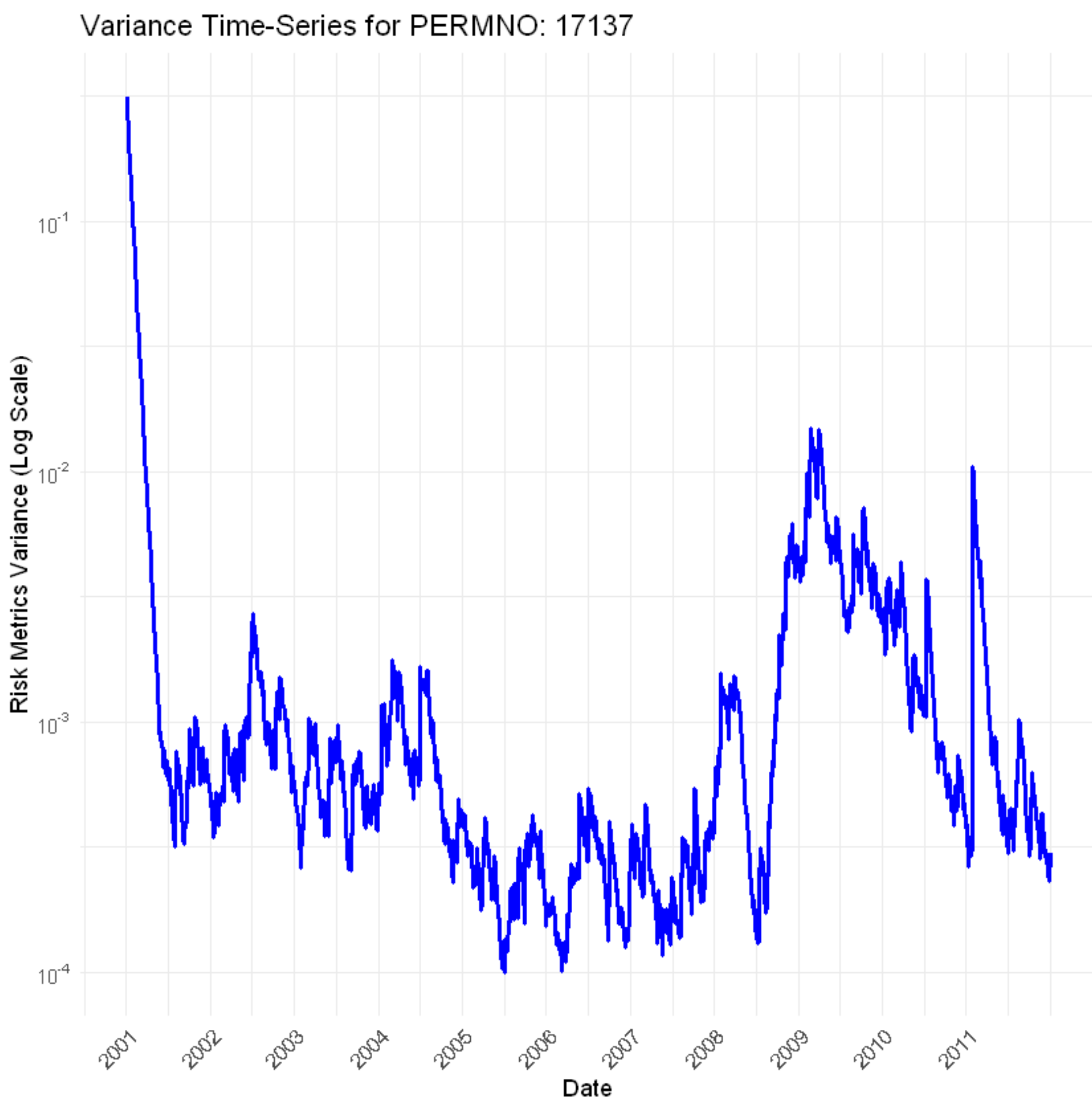
```
     X PERMNO   date    RET vwretd 
     0      0      0      0      0 
```

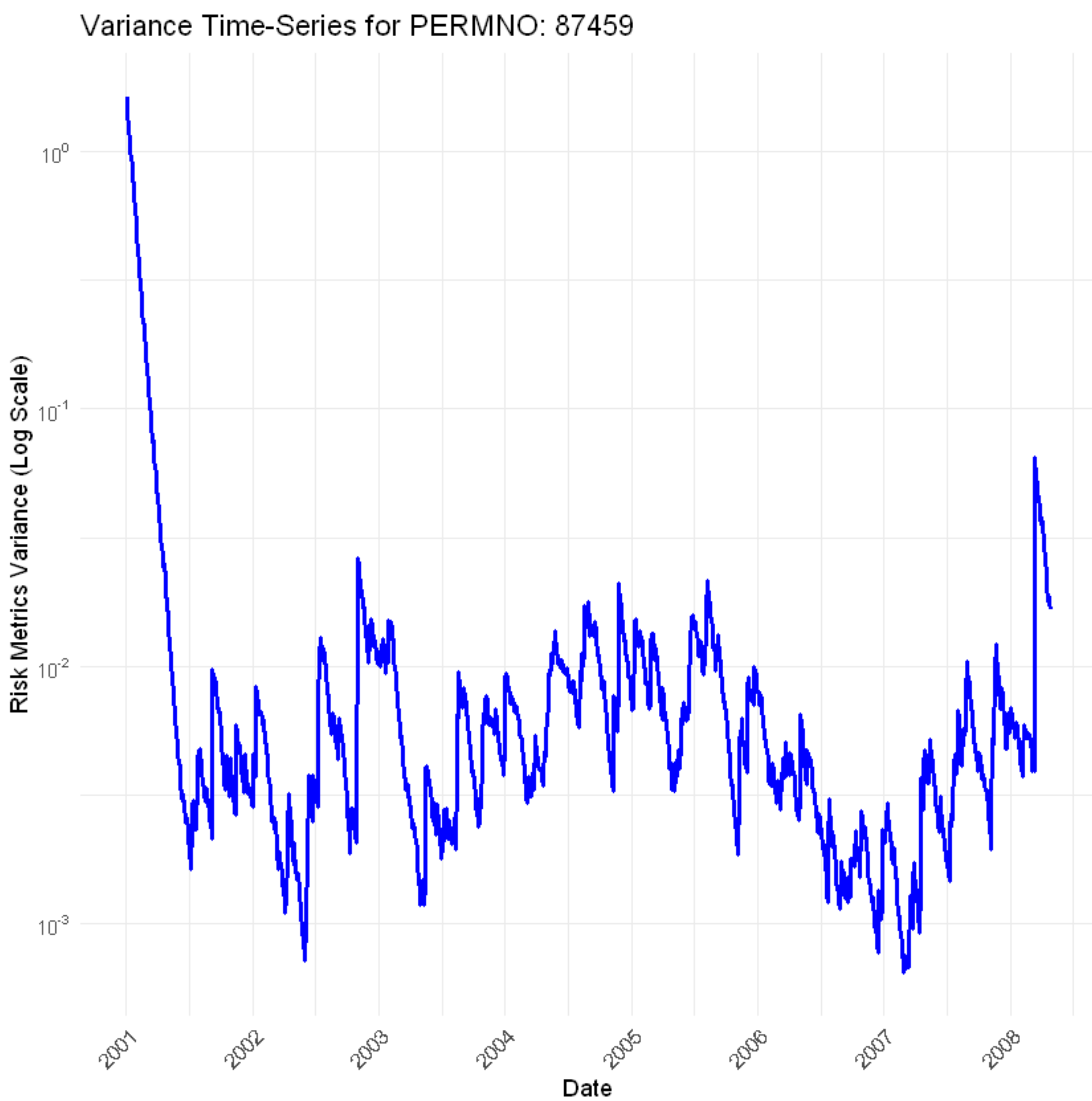
[27]:
```
# Constants
lambda <- 0.94

# Risk Metrics Variance Calculation Function
compute_risk_metrics_variance <- function(returns, initial_variance) {
    n <- length(returns)
    variances <- numeric(n)
    variances[1] <- initial_variance
    for (i in 2:n) {
        variances[i] <- lambda * variances[i-1] + (1 - lambda) * returns[i-1]^2
    }
    return(variances)
}

# 1. Randomly select 5 unique firms from the dataset
set.seed(903948185)
selected_firms <- sample(dsf_2001_2011$PERMNO), 5)

# Print the selected firms for clarity
cat("Selected unique firms (PERMNO):", selected_firms, "\n")
```

```r
# 2. For each of these firms, compute Risk Metrics variances for January 2001␣
↪to December 2011
variances_2001_2011 <- list()

for (permno in selected_firms) {
    # Extract returns and dates for the firm
    firm_data_2001_2011 <- subset(dsf_2001_2011, PERMNO == permno)

    # Get initial variance from the previous computation
    init_var <- initial_variances$Initial_Variance[initial_variances$PERMNO ==␣
↪permno]

    # Compute Risk Metrics variances for 2001-2011
    computed_variances <-␣
↪compute_risk_metrics_variance(firm_data_2001_2011$RET, init_var)
    variances_2001_2011[[as.character(permno)]] <- data.frame(Date =␣
↪firm_data_2001_2011$date, Variance = computed_variances)
}

# 3. Plot the time-series of variances for each of the selected firms with a␣
↪logarithmic scale
library(ggplot2)

date_breaks <- as.Date(paste0(2001:2011, "-01-01"))

for (permno in selected_firms) {
    df_plot <- variances_2001_2011[[as.character(permno)]]

    p <- ggplot(df_plot, aes(x = Date, y = Variance)) +
            geom_line(color = "blue", size = 1) +
            scale_y_log10(breaks = scales::trans_breaks("log10", function(x)␣
↪10^x),
                          labels = scales::trans_format("log10", scales::
↪math_format(10^.x))) +
            scale_x_date(breaks = date_breaks, date_labels = "%Y") +
            ggtitle(paste0("Variance Time-Series for PERMNO: ", permno)) +
            ylab("Risk Metrics Variance (Log Scale)") +
            xlab("Date") +
            theme_minimal() +
            theme(axis.text.x = element_text(angle = 45, hjust = 1))
    print(p)
}
```

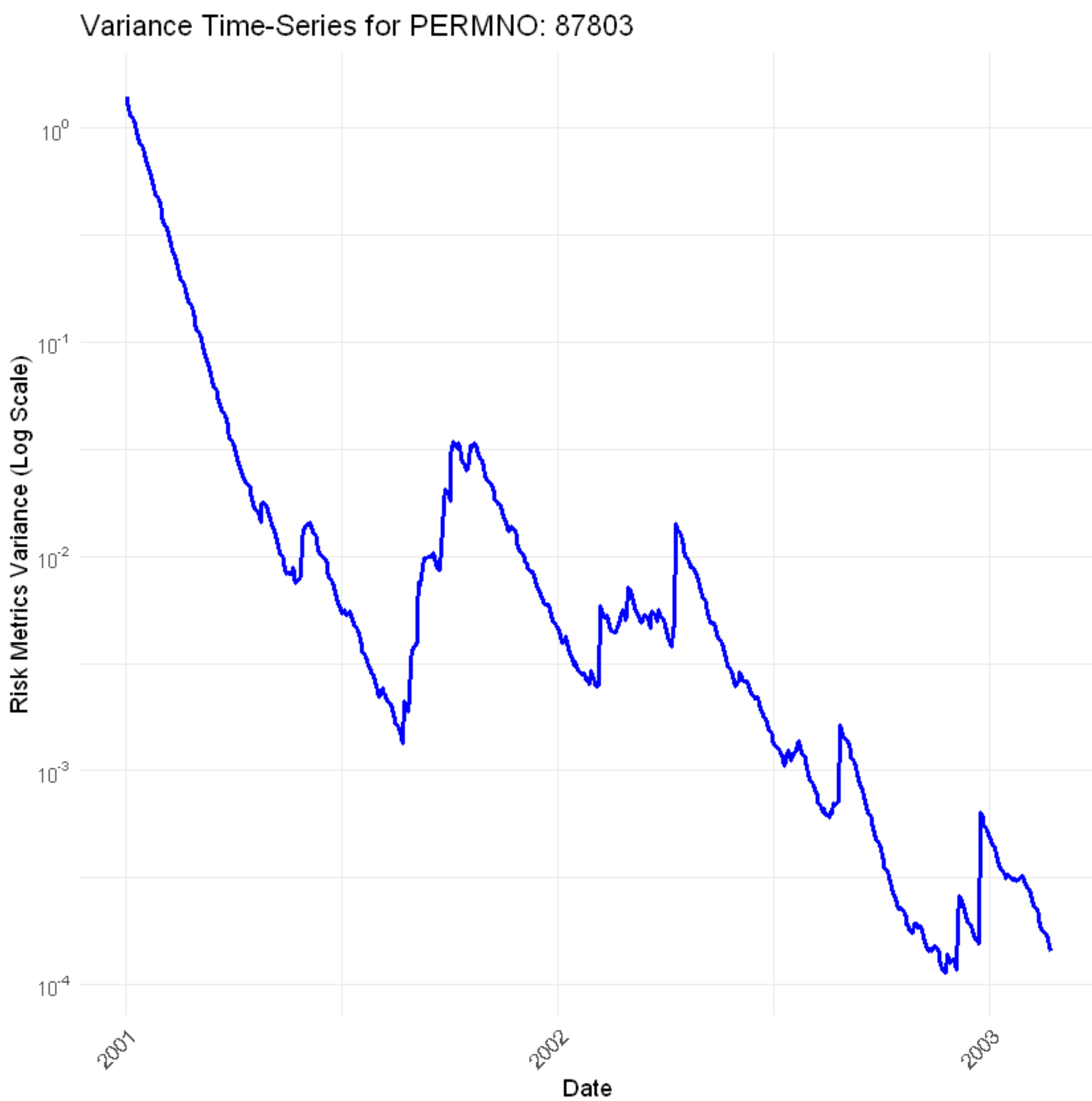
```
Selected unique firms (PERMNO): 32062 17137 87459 87803 88403
```


Variance Time-Series for PERMNO: 32062
Risk Metrics Variance (Log Scale)
10^-1
10^-2
10^-3
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
Date

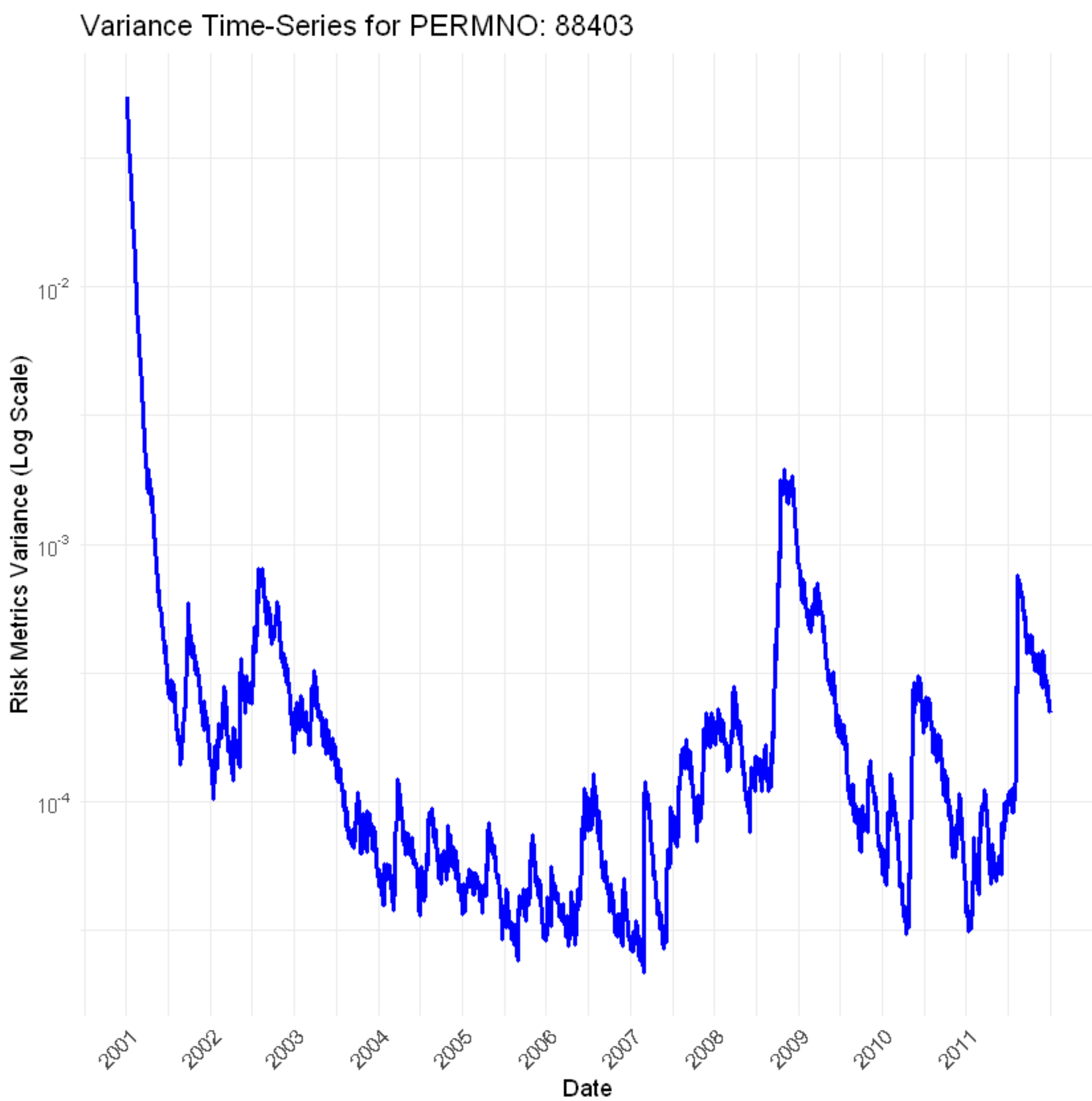

Variance Time-Series for PERMNO: 17137
Risk Metrics Variance (Log Scale)
10^{-1}
10^{-2}
10^{-3}
10^{-4}
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
Date



Variance Time-Series for PERMNO: 87459
Risk Metrics Variance (Log Scale)
10^{0}
10^{-1}
10^{-2}
10^{-3}
2001
2002
2003
2004
2005
2006
2007
2008
Date



Variance Time-Series for PERMNO: 87803
Risk Metrics Variance (Log Scale)
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
2001
2002
2003
Date



[28]:
```r
# 1. Read the file
dsf_data <- read.csv("dsf_jan_2000_dec_2010.csv")

# 2. Convert the date column to a date type
dsf_data$date <- as.Date(dsf_data$date, format="%Y-%m-%d")

# 3. Convert the RET column to numeric
dsf_data$RET <- as.numeric(as.character(dsf_data$RET))

# 4. Check for NAs in each column
na_counts <- sapply(dsf_data, function(x) sum(is.na(x)))
print(na_counts)
```

```
X PERMNO    date     RET vwretd
0      0       0       0      0
```

[29]:
```
total_rows <- nrow(dsf_data)
cat("Total number of rows in dsf_data:", total_rows, "\n")
```

```
Total number of rows in dsf_data: 392735
```

[30]:
```
library(rugarch)

# Function to estimate GARCH(1,1) parameters for a given firm's returns
fit_garch <- function(returns) {
    spec <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder =␣
 ↪c(1, 1)),
                       mean.model = list(armaOrder = c(0, 0), include.mean =␣
 ↪TRUE),
                       distribution.model = "norm")
    fit <- ugarchfit(spec, data = returns, solver = "hybrid")
    coefs <- coef(fit)
    return(coefs[c("alpha1", "beta1")])
}

# Estimate GARCH(1,1) parameters for each of the 250 firms
garch_params <- list()

for (permno in unique(dsf$PERMNO)) {
    firm_returns <- dsf$RET[dsf$PERMNO == permno]
    garch_params[[as.character(permno)]] <- fit_garch(firm_returns)
}

# garch_params now contains estimated alpha and beta values for each of the 250␣
 ↪firms
```

```
Loading required package: parallel


Attaching package: 'rugarch'


The following object is masked from 'package:stats':

    sigma
```

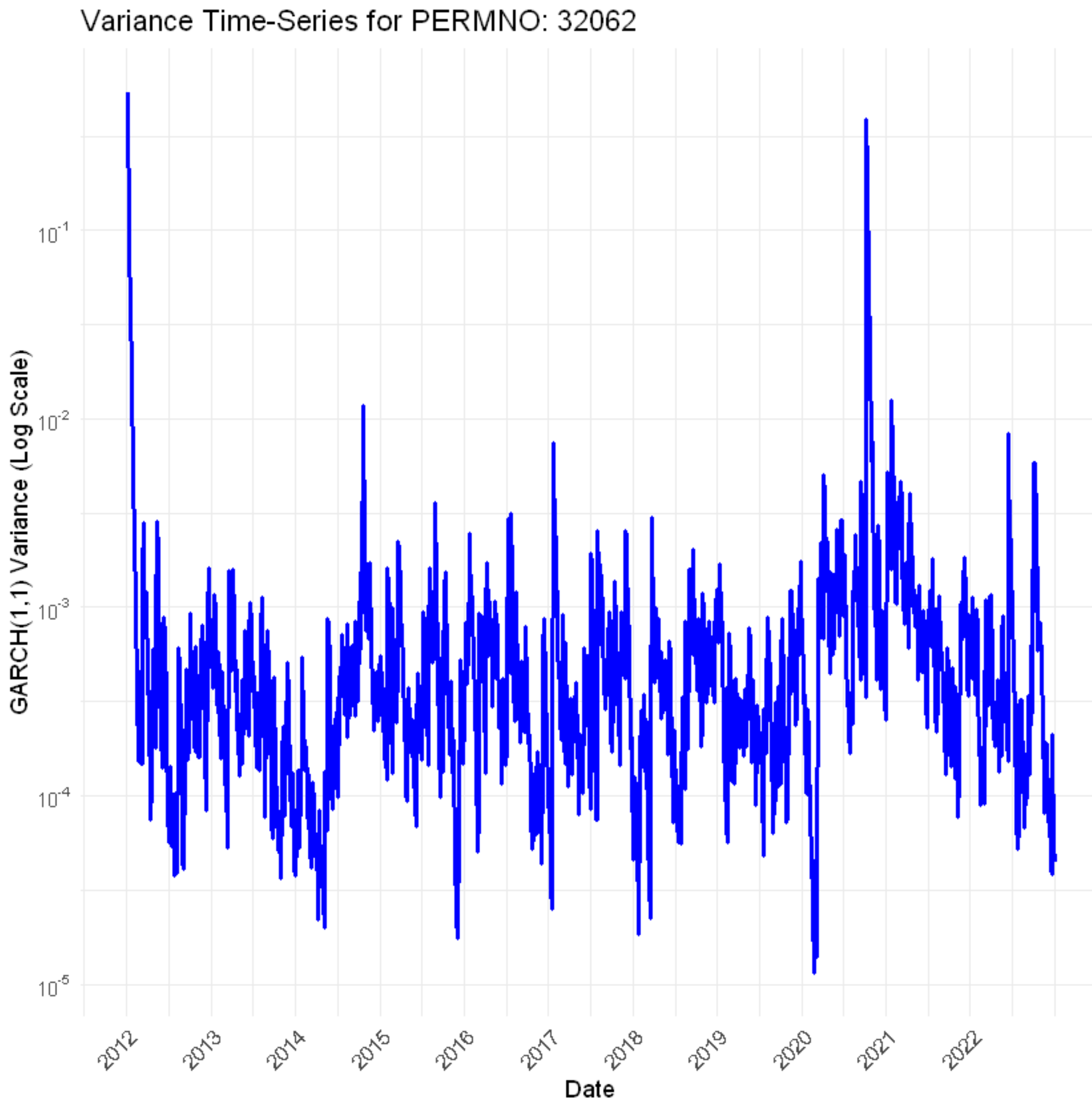
[31]:
```
head(garch_params)
```

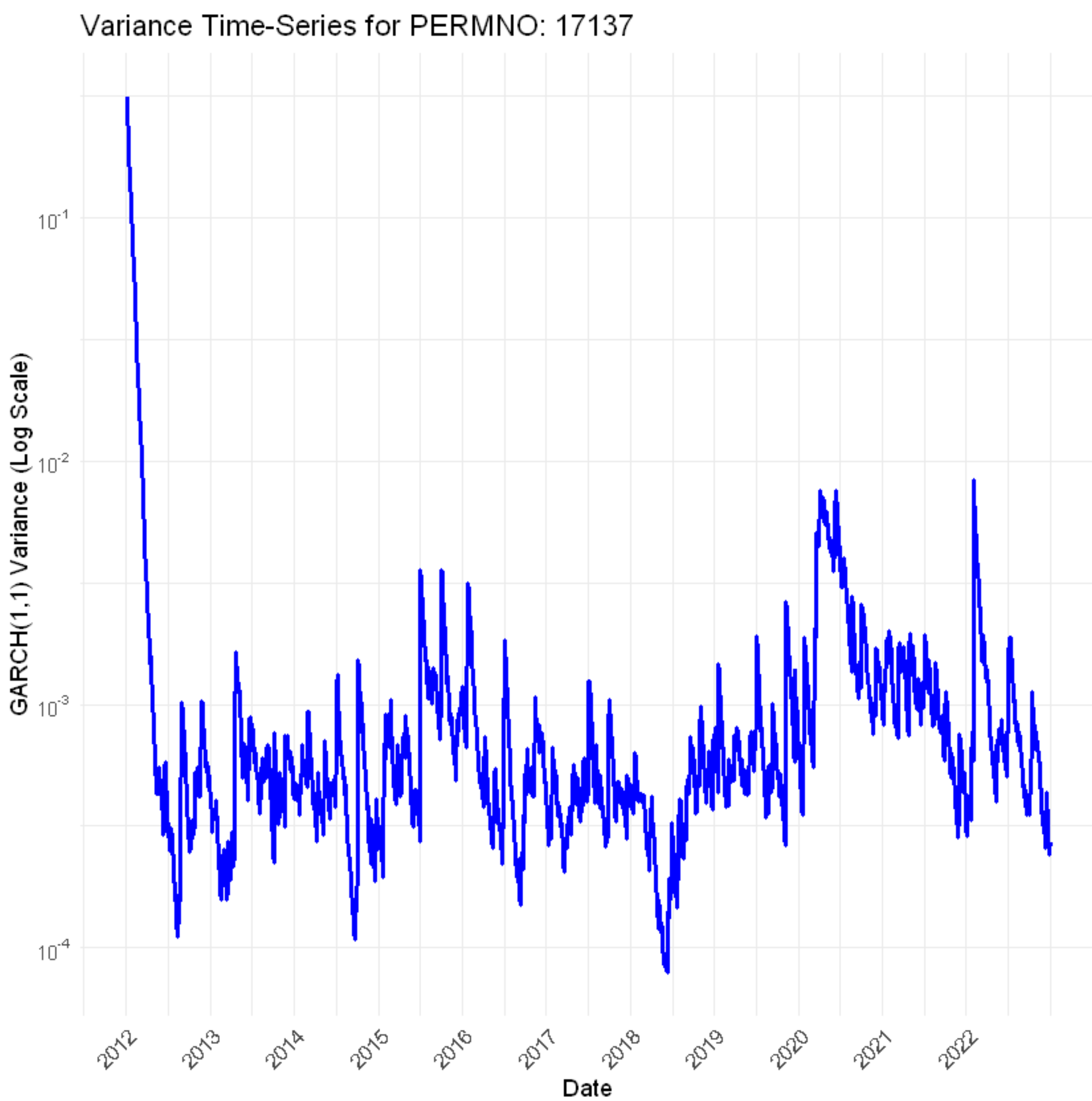
$‘10056‘	**alpha1**	0.329841608460946	**beta1**	0.669158379163588
$‘10100‘	**alpha1**	0.151555744029344	**beta1**	0.847444236834842

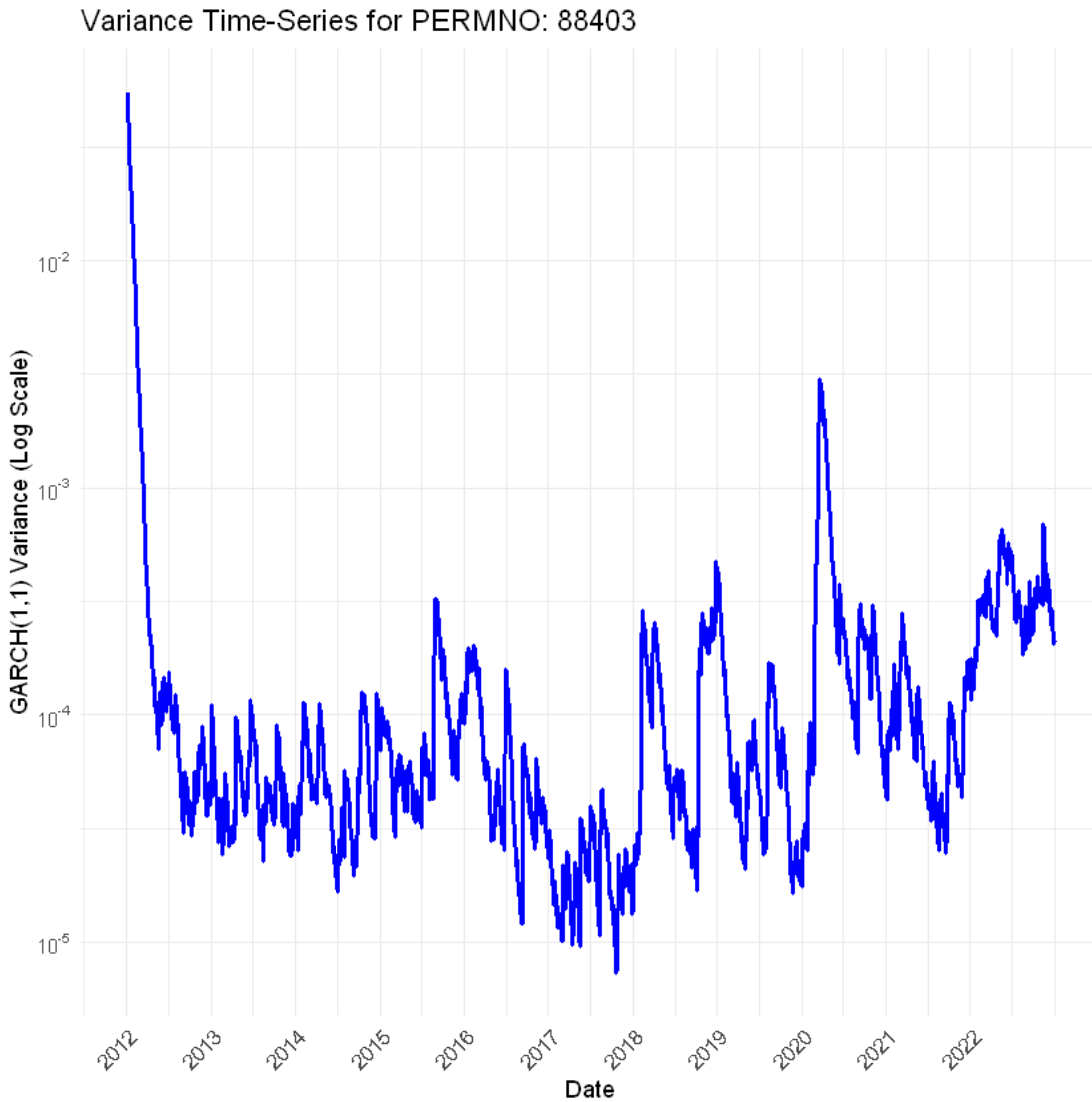
$`10307`	**alpha1**	0.0534610185152604 **beta1**	0.944819041035655
$`10456`	**alpha1**	0.409191687551794 **beta1**	3.57560068104885e-09
$`10517`	**alpha1**	0.182854163381449 **beta1**	0.768460250829255
$`10745`	**alpha1**	0.164913306539782 **beta1**	0.834086683075598

[41]:
```
# Define the function to compute GARCH(1,1) variances over time
compute_garch_variance <- function(returns, alpha, beta, initial_variance) {
    n <- length(returns)
    variances <- numeric(n)
    variances[1] <- initial_variance

    for (i in 2:n) {
        variances[i] <- alpha * returns[i-1]^2 + beta * variances[i-1]
    }

    return(variances)
}

# Load the CSV file for 2012-2022 into an R dataframe
dsf_2012_2022 <- read.csv("dsf_jan_2012_dec_2022.csv")
# Convert the Date column in dsf into a proper date format
dsf_2012_2022$date <- as.Date(dsf_2012_2022$date, format="%Y-%m-%d")
# Convert the RET column to numeric
dsf_2012_2022$RET <- as.numeric(as.character(dsf_2012_2022$RET))

# Explicitly specify the selected firms from Q4
selected_firms <- c(32062, 17137, 88403)


# Compute GARCH(1,1) variances for the period "January 2012 to December 2022"␣
↪using the estimated parameters
variances_2012_2022 <- list()



for (permno in selected_firms) {
    # Extract returns for the firm for the period 2012-2022
    firm_returns_2012_2022 <- dsf_2012_2022$RET[dsf_2012_2022$PERMNO == permno]
    # Get initial variance from the previous computation
    init_var <- initial_variances$Initial_Variance[initial_variances$PERMNO ==␣
↪permno]
    # Use GARCH(1,1) to forecast variances for 2012-2022
    alpha <- garch_params[[as.character(permno)]]["alpha1"]
    beta <- garch_params[[as.character(permno)]]["beta1"]
    variances_2012_2022[[as.character(permno)]] <-␣
↪compute_garch_variance(firm_returns_2012_2022, alpha, beta, init_var)
```

```
}

# Plot the time-series of variances for each of the selected firms with a␣
↪logarithmic scale
library(ggplot2)

date_breaks <- as.Date(paste0(2012:2022, "-01-01"))

for (permno in selected_firms) {
    # Extract the relevant date range for the firm
    relevant_dates <- dsf_2012_2022$date[dsf_2012_2022$PERMNO == permno]

    # Convert the variance vector to a dataframe
    df_plot <- data.frame(Date = relevant_dates,
                          Variance = variances_2012_2022[[as.
↪character(permno)]])

    p <- ggplot(df_plot, aes(x = Date, y = Variance)) +
            geom_line(color = "blue", size = 1) +
            scale_y_log10(breaks = scales::trans_breaks("log10", function(x)␣
↪10^x),
                          labels = scales::trans_format("log10", scales::
↪math_format(10^.x))) +
            scale_x_date(breaks = date_breaks, date_labels = "%Y") +
            ggtitle(paste0("Variance Time-Series for PERMNO: ", permno)) +
            ylab("GARCH(1,1) Variance (Log Scale)") +
            xlab("Date") +
            theme_minimal() +
            theme(axis.text.x = element_text(angle = 45, hjust = 1))
    print(p)
}
```


Variance Time-Series for PERMNO: 32062
GARCH(1,1) Variance (Log Scale)
10^-1
10^-2
10^-3
10^-4
10^-5
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
Date


Variance Time-Series for PERMNO: 17137
GARCH(1,1) Variance (Log Scale)
10⁻¹
10⁻²
10⁻³
10⁻⁴
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
Date



Comparison and Analysis:

1. Volatility Over Time:
    - In the plots from the first decade (2000-2010), the volatilities are generally smoother with occasional spikes. This indicates periods of financial calm interspersed with short-term shocks or market events.
    - The plots from the second decade (2012-2022), generated using GARCH, show more pronounced variance. This suggests more frequent market events, shocks, or higher market uncertainty during this period.
2. Increased Variance in the Second Decade:
    - The GARCH model often captures the clustering of volatility, which means that high volatility days tend to be followed by high volatility days and vice versa for low volatility.

This "volatility clustering" feature of financial returns is a central reason why models like GARCH are popular.
- The higher variance in the second decade might be indicative of significant economic events or changes during this period. For example, we know from history that the 2010s saw significant geopolitical tensions, trade wars, and other macroeconomic events that might have caused increased market uncertainty.

3. Economic Implications:
   - Financial Markets: Increased volatility often implies greater uncertainty in the financial markets. This can be due to a variety of reasons - from macroeconomic factors, policy changes, geopolitical events, to shifts in investor sentiment.
   - Investment: For investors, higher volatility usually means higher risk. Portfolios might need rebalancing to adapt to the changing risk profile of assets.
   - Trading: For traders, especially those involved in short-term trading, increased volatility might mean more opportunities due to price fluctuations. However, it also means more risk.
   - Economic Indicators: High volatility in stock returns can sometimes be indicative of broader economic issues. For instance, it might indicate problems with corporate profitability, changes in regulations, or concerns about economic growth.

4. Missing Data for Two Firms in the Second Decade:
   - It's important to note that data was missing for two firms in the 2012-2022 dataset. The absence of these firms might skew the comparison slightly. The reasons for missing data can vary - the firms might have gone out of business, been acquired, or the data might just be unavailable.
   - It would be interesting to investigate why data for these firms is missing, as that in itself could provide insights into the economic and corporate landscape of the second decade.

5. Comparison of Models:
   - While the RiskMetrics model provided a more general approach to volatility, GARCH(1,1) provides a more adaptable model where volatility is influenced by past returns (through the alpha parameter) and past volatilities (through the beta parameter). This adaptability is likely why the GARCH model picked up more pronounced variances in the second decade.

Conclusion: The increased volatility in the 2012-2022 period, as captured by the GARCH model, suggests a decade of higher market uncertainty compared to 2000-2010. While models provide a quantitative measure, understanding the qualitative economic, geopolitical, and corporate events of these periods can provide a comprehensive picture of the evolving risk landscape.