Time Series Analysis in Finance

Impact of Red Sea Crisis on Global Shipping and Stock Markets

Jiahua Duojie, Thiam Mouhamadou

Lucerne University of Applied Sciences and Arts

May 9, 2025

Contents

1 Introduction

The Red Sea is one of the most important maritime trade routes in the world, with approximately 12% of global trade passing through this vital waterway. Since late 2023, Houthi rebel attacks on commercial vessels in response to the Israel-Hamas conflict have severely disrupted shipping operations in this region. Many shipping companies have been forced to reroute vessels around the Cape of Good Hope, adding significant time and costs to global supply chains.

This project aims to analyze the impact of the Red Sea crisis on global shipping and stock markets using time series methods. By comparing shipping-related financial instruments with broader market indices, we seek to quantify the specific effects of this geopolitical crisis on the shipping industry and related financial markets.

The Red Sea crisis represents a unique natural experiment to examine how geopolitical events affect specific industries differently from broader market movements. This analysis has implications for risk management, portfolio diversification, and understanding the economic impact of regional conflicts on global trade.

2 Data Collection and Preprocessing

```r
# Define tickers and dates
shipping_tickers <- c("ZIM", "MAERSK-B.CO", "HLAG.DE", "DAC") # Shipping companies
index_tickers <- c("SPY", "XLE")                              # Market indices
oil_tickers <- c("USO")                                       # Oil ETF

start_date <- "2023-01-01"
end_date <- "2025-04-30"
crisis_start <- "2023-11-19"  # Date of first Houthi attack

# Download data with error handling
shipping_data <- download_with_error_handling(shipping_tickers, start_date, end_date)
index_data <- download_with_error_handling(index_tickers, start_date, end_date)
oil_data <- download_with_error_handling(oil_tickers, start_date, end_date)

# Process data
data <- process_financial_data(shipping_data, index_data, oil_data, crisis_start)

# Display summary statistics
stats_table <- calculate_descriptive_stats(data$returns)
kable(stats_table, caption = "Summary Statistics of Daily Returns (%)",
      digits = 2, booktabs = TRUE)
```

Table 1: Summary Statistics of Daily Returns (%)

	Mean	SD	Min	Max	Skewness	Kurtosis
ZIM	0	0.04	-0.18	0.17	0.02	1.90
MAERSK.B.CO	0	0.03	-0.19	0.08	-1.14	6.92
HLAG.DE	0	0.03	-0.17	0.15	-0.46	4.56
DAC	0	0.02	-0.08	0.08	0.14	2.66
SPY	0	0.01	-0.06	0.10	0.72	18.62
XLE	0	0.01	-0.10	0.07	-0.88	6.92
USO	0	0.02	-0.07	0.06	-0.28	0.67

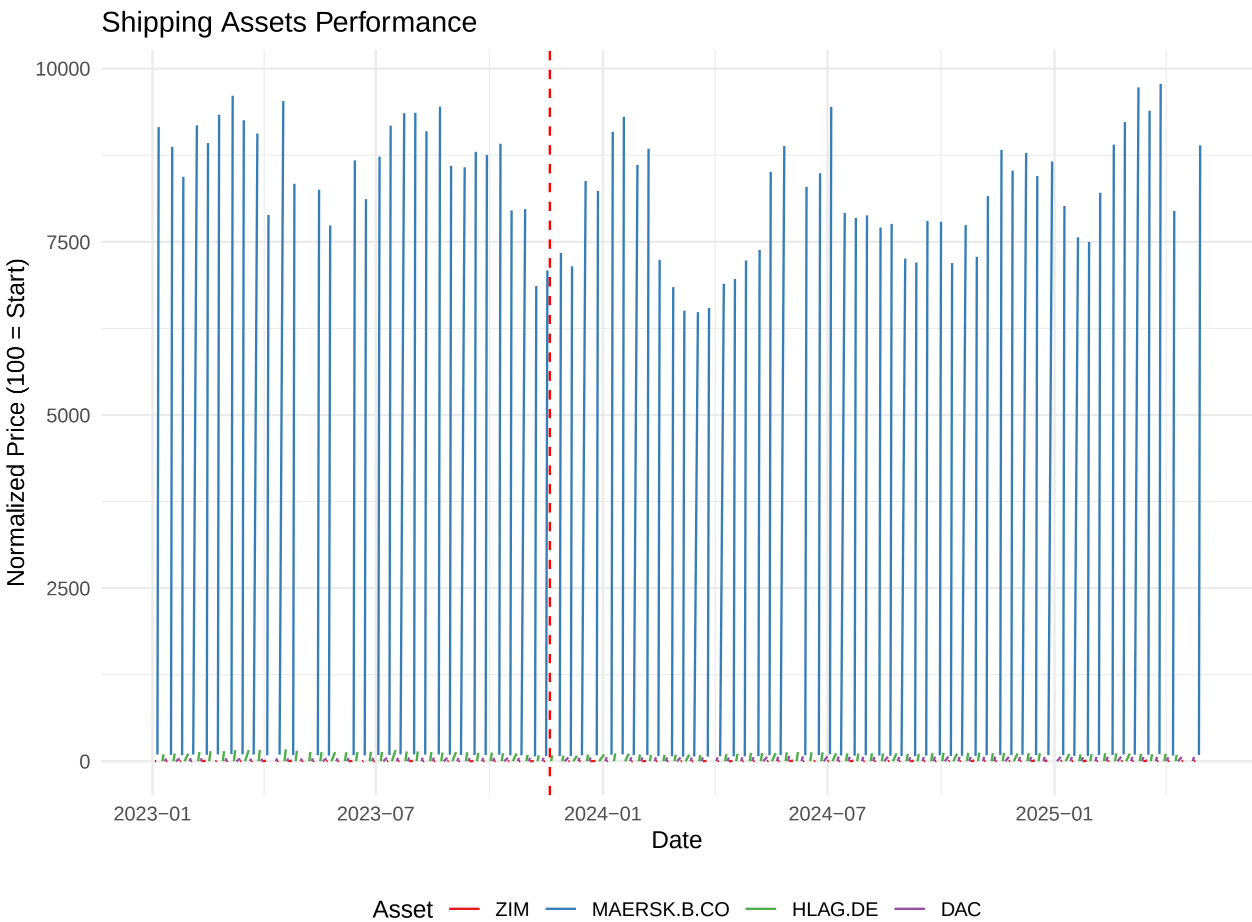
3 Descriptive Analysis

```r
# Compare pre-crisis and post-crisis periods
period_comparison <- compare_periods(data$returns, data$crisis_start)
kable(period_comparison, caption = "Pre-Crisis vs Crisis Period Statistics",
      digits = 2, booktabs = TRUE)
```

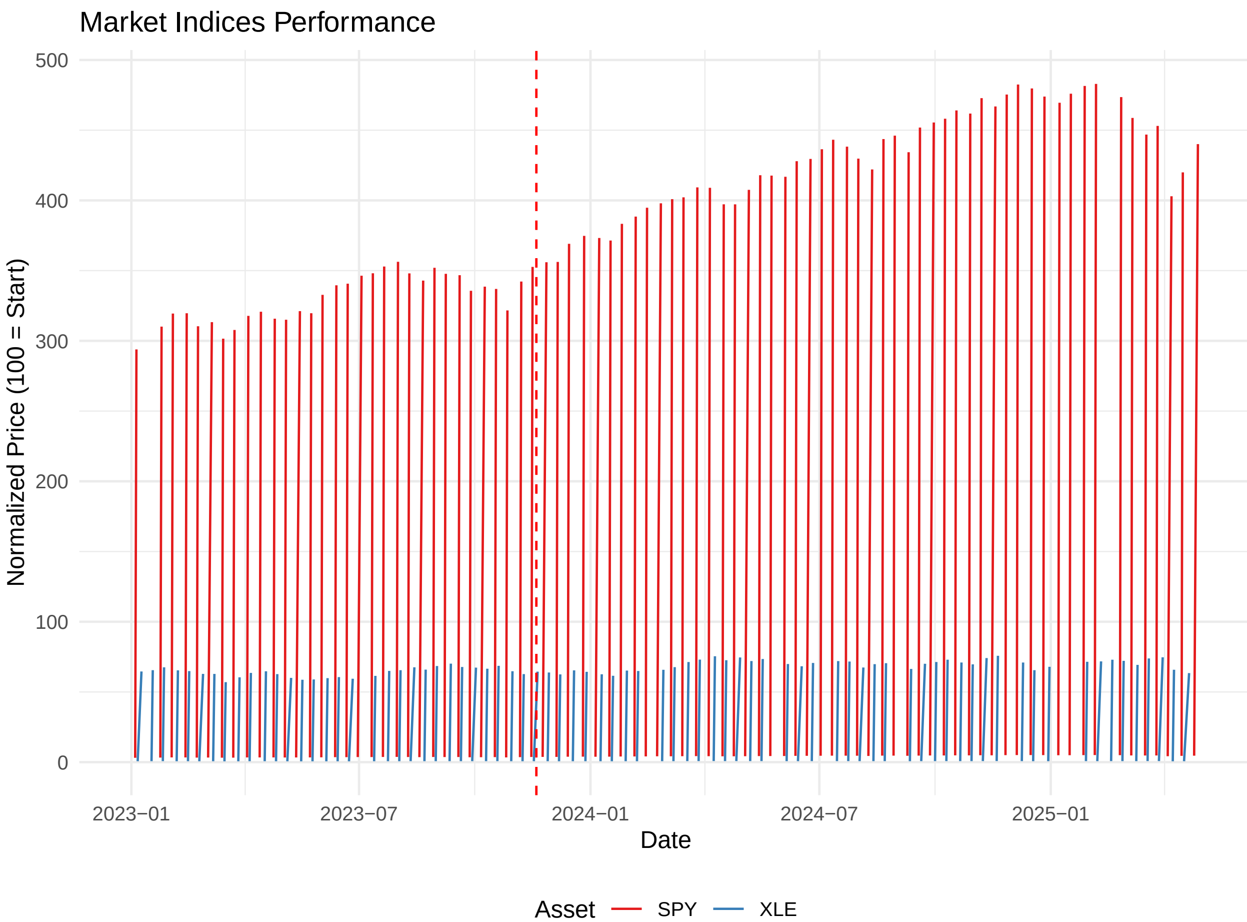
Table 2: Pre-Crisis vs Crisis Period Statistics

Asset	Pre_Crisis_Mean	Crisis_Mean	Pre_Crisis_SD	Crisis_SD	Pre_Crisis_Min	Crisis_Min	Pre_Crisis_Max	Crisis_Max	Pre_Crisis_Skew	Crisis_Skew	Pre_Crisis_Kurt	Crisis_Kurt	Mean_Change	SD_Change
ZIM	0	0	0.04	0.05	-0.11	-0.18	0.14	0.17	0.39	-0.13	1.04	1.66	0.01	0.01
MAERSK.B.CO	0	0	0.02	0.03	-0.19	-0.16	0.05	0.08	-2.25	-0.67	14.25	3.82	0.00	0.00
HLAG.DE	0	0	0.03	0.03	-0.11	-0.17	0.11	0.15	-0.24	-0.58	1.89	5.77	0.00	0.00
DAC	0	0	0.02	0.02	-0.05	-0.08	0.06	0.08	0.47	-0.03	1.50	3.18	0.00	0.00
SPY	0	0	0.01	0.01	-0.02	-0.06	0.02	0.10	-0.01	0.90	-0.49	21.09	0.00	0.00
XLE	0	0	0.01	0.01	-0.06	-0.10	0.04	0.07	-0.04	-1.43	0.57	10.93	0.00	0.00
USO	0	0	0.02	0.02	-0.06	-0.07	0.06	0.06	-0.26	-0.31	0.14	1.07	0.00	0.00

```r
# Plot price data
shipping_plot <- plot_prices(data$normalized_prices, names(shipping_data),
                             data$crisis_start, "Shipping Assets Performance")
print(shipping_plot)
```




```r
market_plot <- plot_prices(data$normalized_prices, names(index_data),
                           data$crisis_start, "Market Indices Performance")
print(market_plot)
```



4 Stationarity and Correlation Analysis

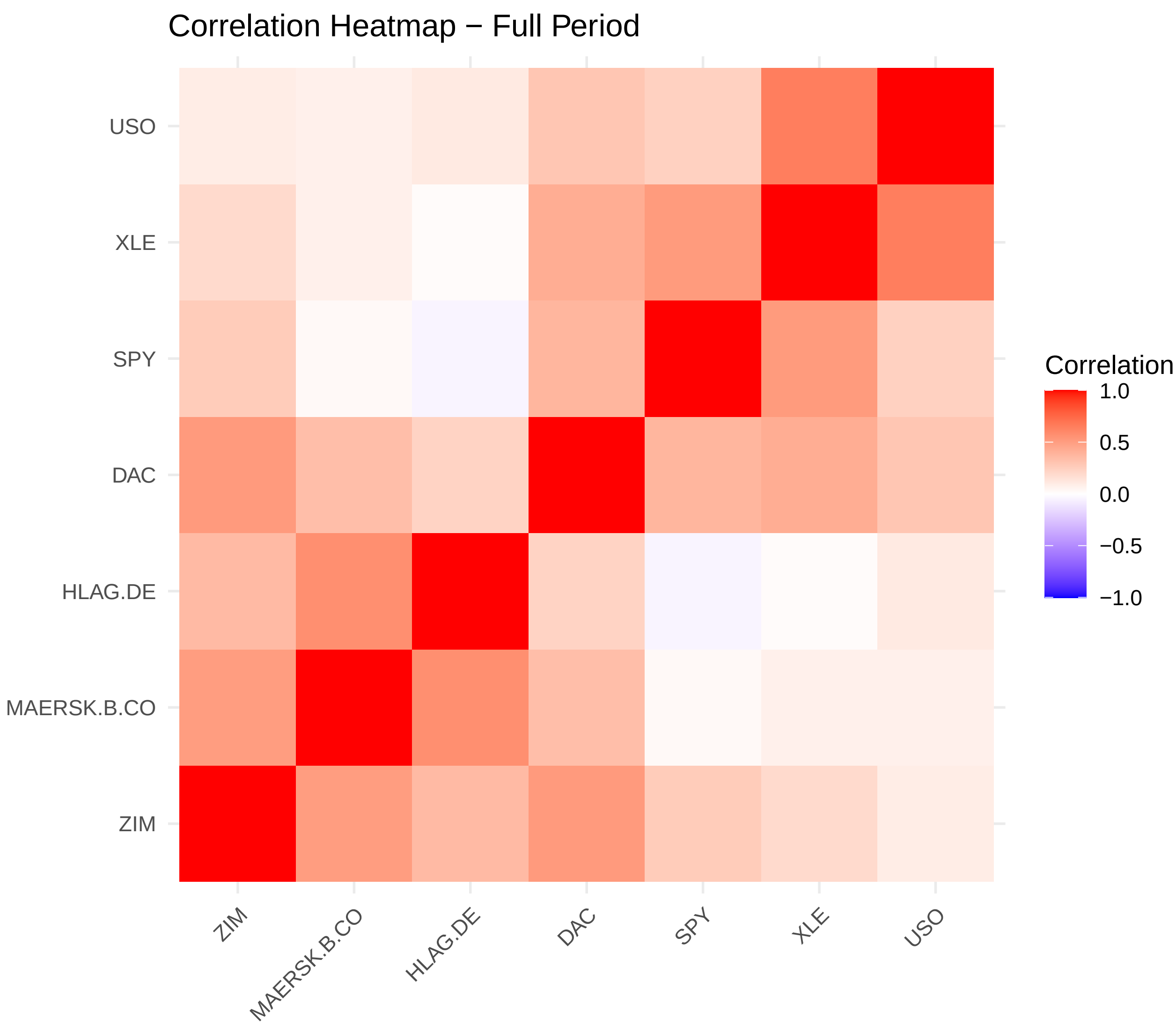
```r
# Test stationarity
stationarity_results <- enhanced_stationarity_tests(data$returns)
kable(stationarity_results, caption = "Stationarity Test Results",
      digits = 4, booktabs = TRUE)
```

Table 3: Stationarity Test Results

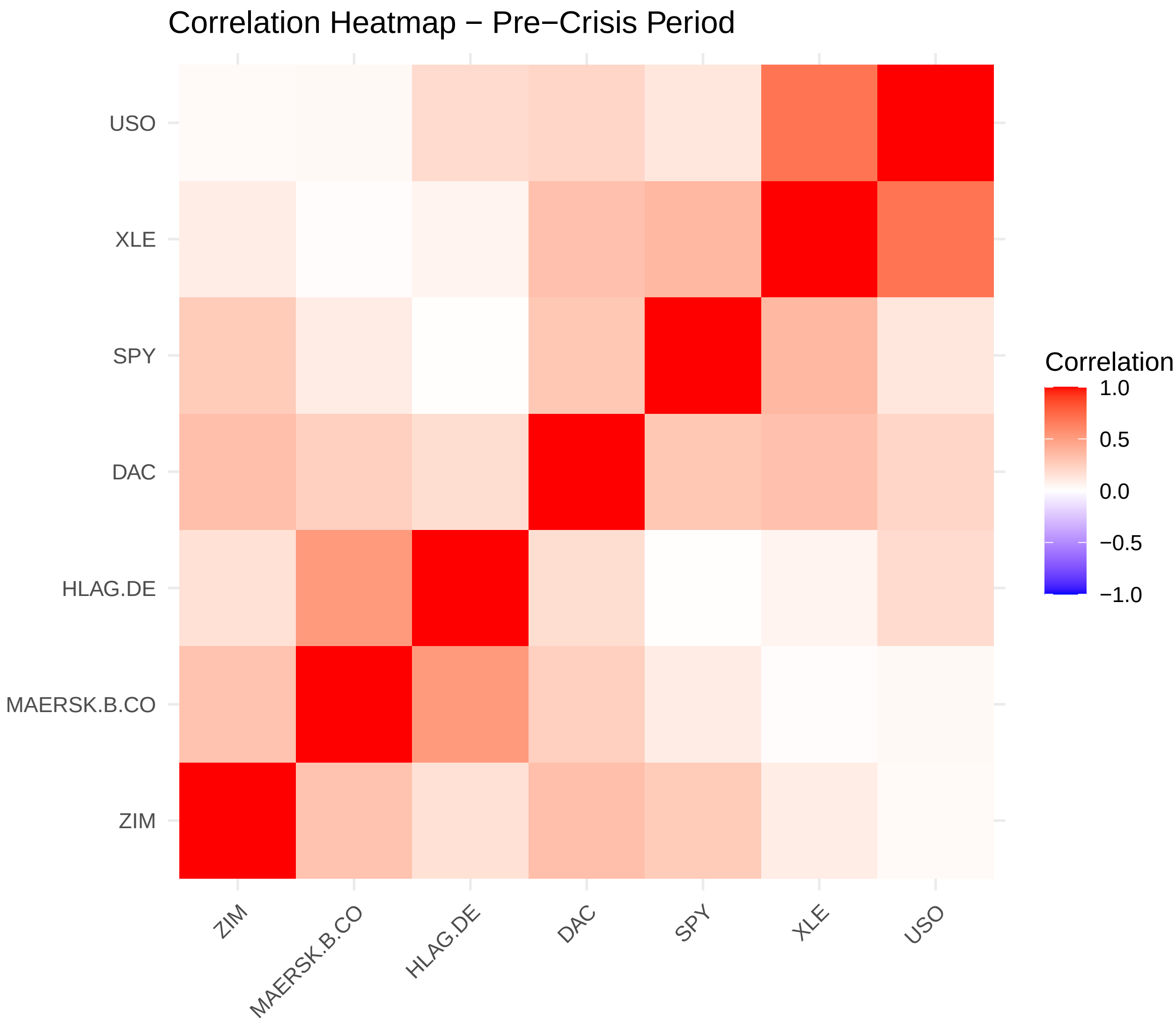
	Asset	ADF_Stat	ADF_pval	KPSS_Stat	KPSS_pval	PP_Stat	PP_pval	ADF_Interpretation	KPSS_Interpretation	PP_Interpretation	Overall_Conclusion
Dickey-Fuller	ZIM	-7.2724	0.01	0.0624	0.1	-489.5847	0.01	Stationary	Stationary	Stationary	Stationary
Dickey-Fuller1	MAERSK.B.CO	-7.8737	0.01	0.0777	0.1	-513.1787	0.01	Stationary	Stationary	Stationary	Stationary
Dickey-Fuller2	HLAG.DE	-7.5316	0.01	0.0489	0.1	-494.0972	0.01	Stationary	Stationary	Stationary	Stationary
Dickey-Fuller3	DAC	-7.8048	0.01	0.0889	0.1	-527.0444	0.01	Stationary	Stationary	Stationary	Stationary
Dickey-Fuller4	SPY	-7.4683	0.01	0.1709	0.1	-542.6688	0.01	Stationary	Stationary	Stationary	Stationary
Dickey-Fuller5	XLE	-8.3656	0.01	0.1046	0.1	-500.2477	0.01	Stationary	Stationary	Stationary	Stationary

	Asset	ADF_Stat	ADF_pval	KPSS_Stat	KPSS_pval	PP_Stat	PP_pval	ADF_Interpretation	KPSS_Interpretation	PP_Interpretation	Overall_Conclusion
Dickey-Fuller6	USO	-9.2320	0.01	0.1449	0.1	-503.7522	0.01	Stationary	Stationary	Stationary	Stationary

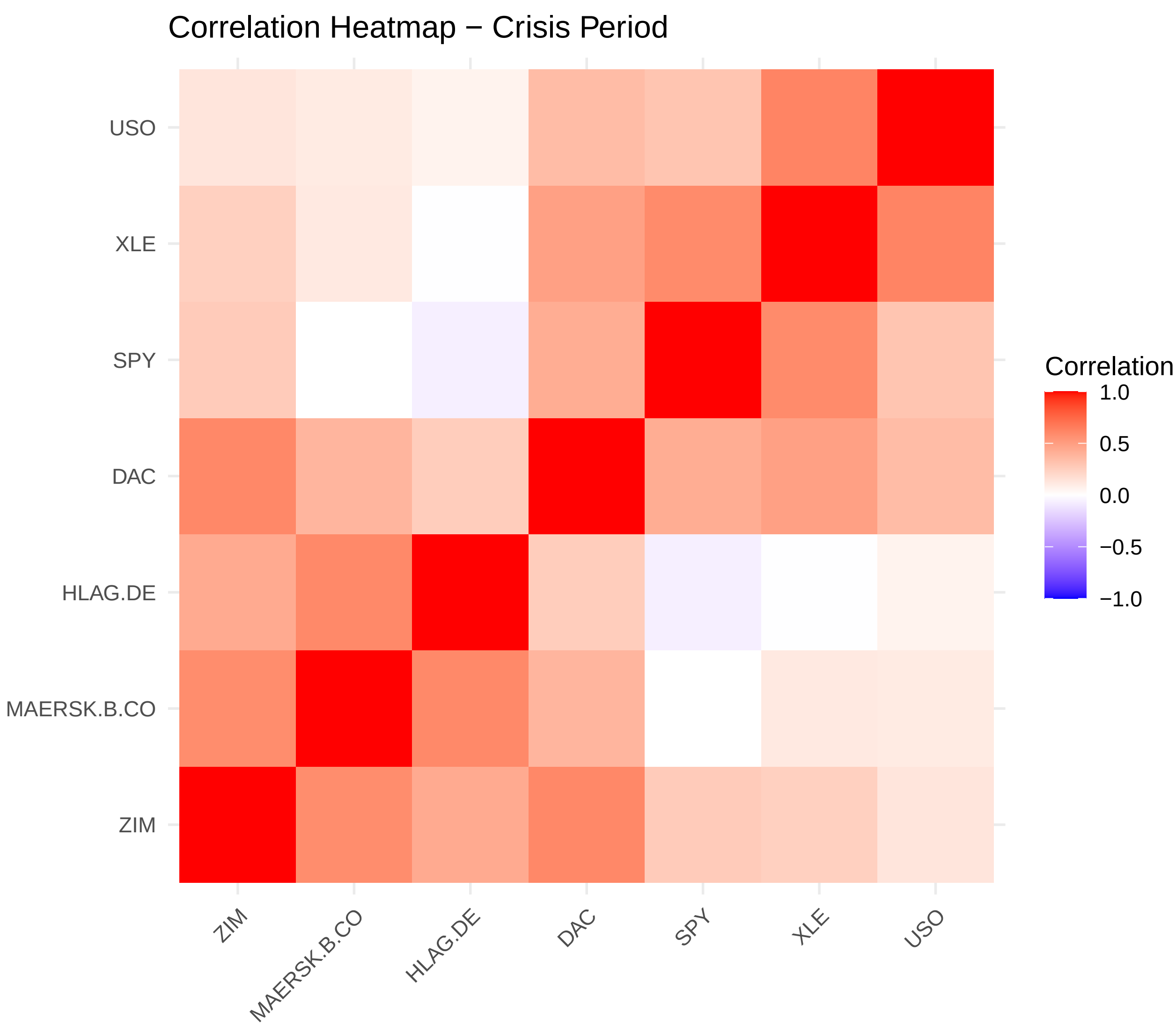
```r
# Correlation analysis
all_corr <- create_correlation_heatmap(data$returns)
pre_crisis_corr <- create_correlation_heatmap(data$returns, "pre_crisis", data$crisis_start)
crisis_corr <- create_correlation_heatmap(data$returns, "crisis", data$crisis_start)

print(all_corr)
```



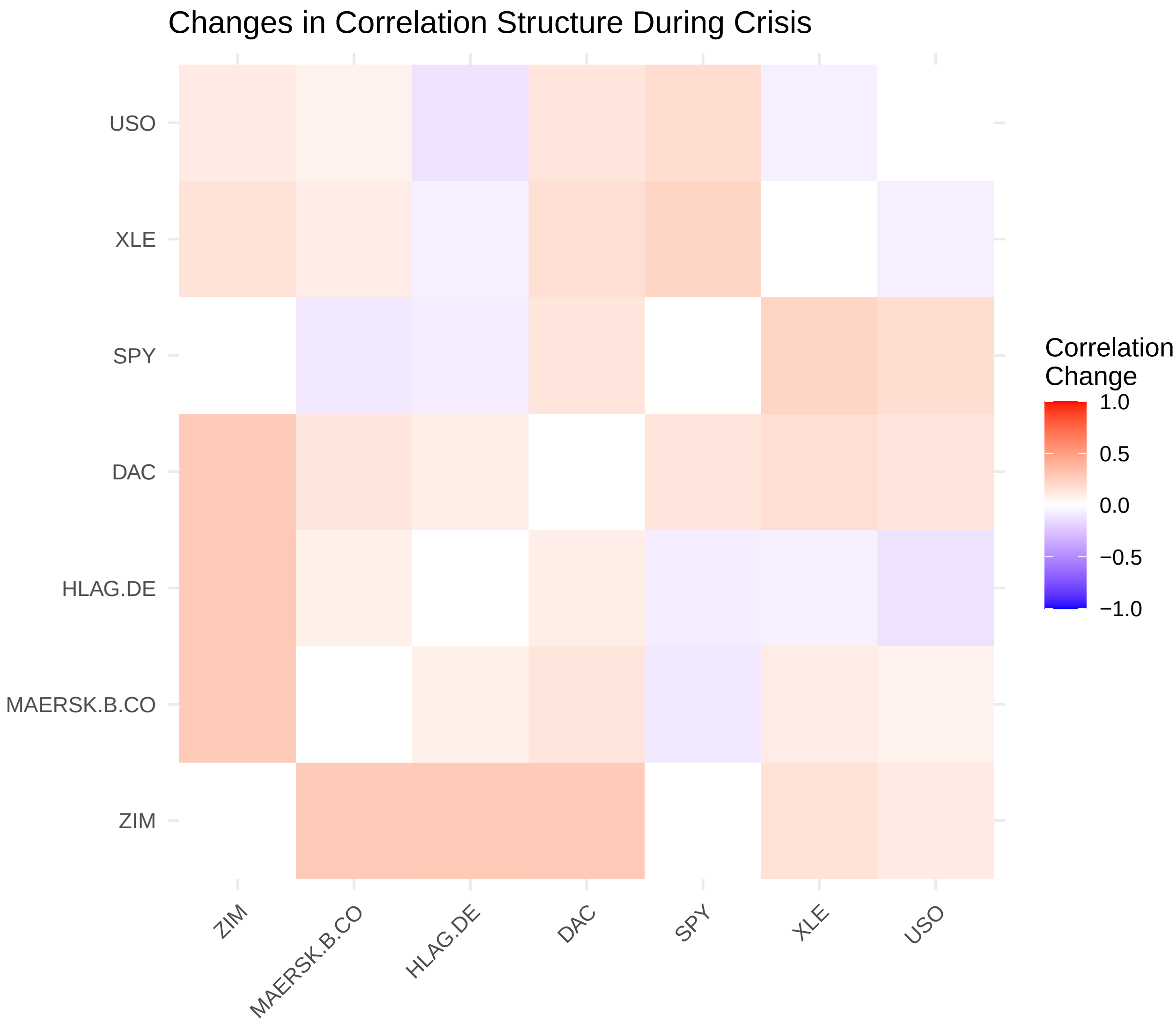

```r
print(pre_crisis_corr)
```




```r
print(crisis_corr)
```




```r
# Analyze correlation changes
cor_change <- analyze_correlation_changes(data$returns, data$crisis_start)
print(cor_change$plot)
```



5 Event Analysis

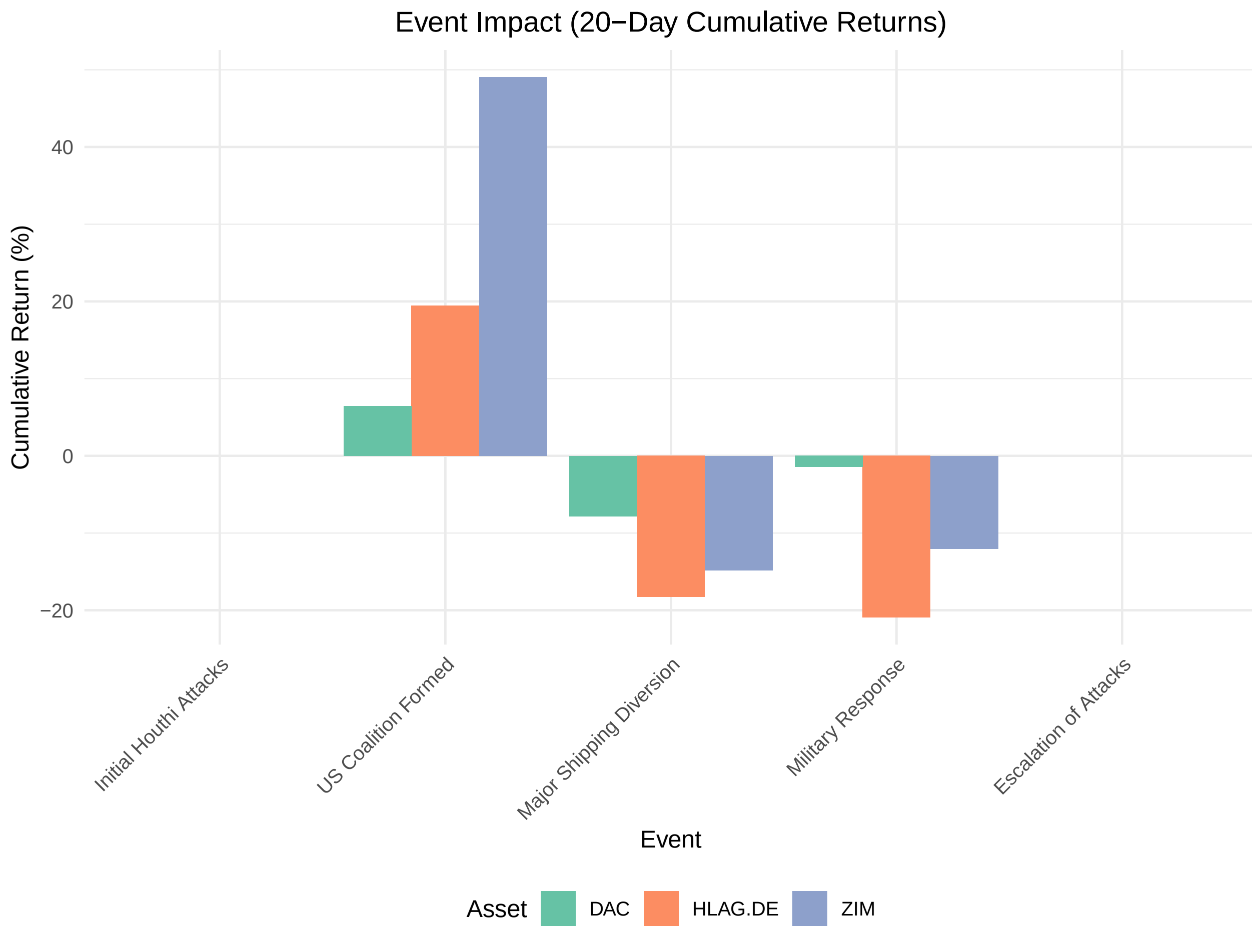
```
# Define events
# Define major Red Sea crisis events
events <- data.frame(
  Event = c("Initial Houthi Attacks",
            "US Coalition Formed",
            "Major Shipping Diversion",
            "Military Response",
            "Escalation of Attacks"),
  Date = as.Date(c("2023-11-19",
                  "2023-12-18",
                  "2024-01-05",
                  "2024-01-12",
                  "2024-02-19"))
)

# Calculate event returns
```

```r
market_index <- NULL
if (any(grepl("SPY_returns", colnames(data$returns)))) {
  market_index <- "SPY_returns"
}

event_returns <- calculate_event_returns(data$returns, events, market_index)

# Display event returns
if (!is.null(event_returns)) {
  kable(event_returns, caption = "20-Day Cumulative Returns (%) After Key Events",
        digits = 2, booktabs = TRUE)
}
```

Table 4: 20-Day Cumulative Returns (%) After Key Events

Event	Date	ZIM_cum_return	MAERSK.B.CO_cum_return	HLAG.DE_cum_return	DAC_cum_return	SPY_cum_return	XLE_cum_return	USO_cum_return
Initial Houthi Attacks	2023-11-19	NA	NA	NA	NA	NA	NA	NA
US Coalition Formed	2023-12-18	49.06	11.02	19.44	6.43	2.46	-2.30	3.08
Major Shipping Diversion	2024-01-05	-14.82	-10.43	-18.26	-7.83	5.79	1.10	0.82
Military Response	2024-01-12	-12.04	-20.73	-20.95	-1.45	3.95	3.25	7.72
Escalation of Attacks	2024-02-19	NA	NA	NA	NA	NA	NA	NA

```r
# Plot event impact
impact_plot <- plot_event_impact(event_returns, data$shipping_cols)
print(impact_plot)
```



6 Market Beta Analysis

```r
# Calculate rolling betas
if ("SPY_returns" %in% colnames(data$returns)) {
  rolling_betas <- calculate_rolling_beta(data$returns, "SPY_returns")

  # Plot betas
  beta_plot <- plot_rolling_betas(rolling_betas, data$crisis_start)
  print(beta_plot)

  # Analyze beta changes
  beta_changes <- analyze_beta_changes(rolling_betas, data$crisis_start)
  kable(beta_changes, caption = "Beta Changes Analysis",
        digits = 4, booktabs = TRUE)
}
```

7 Value at Risk Analysis

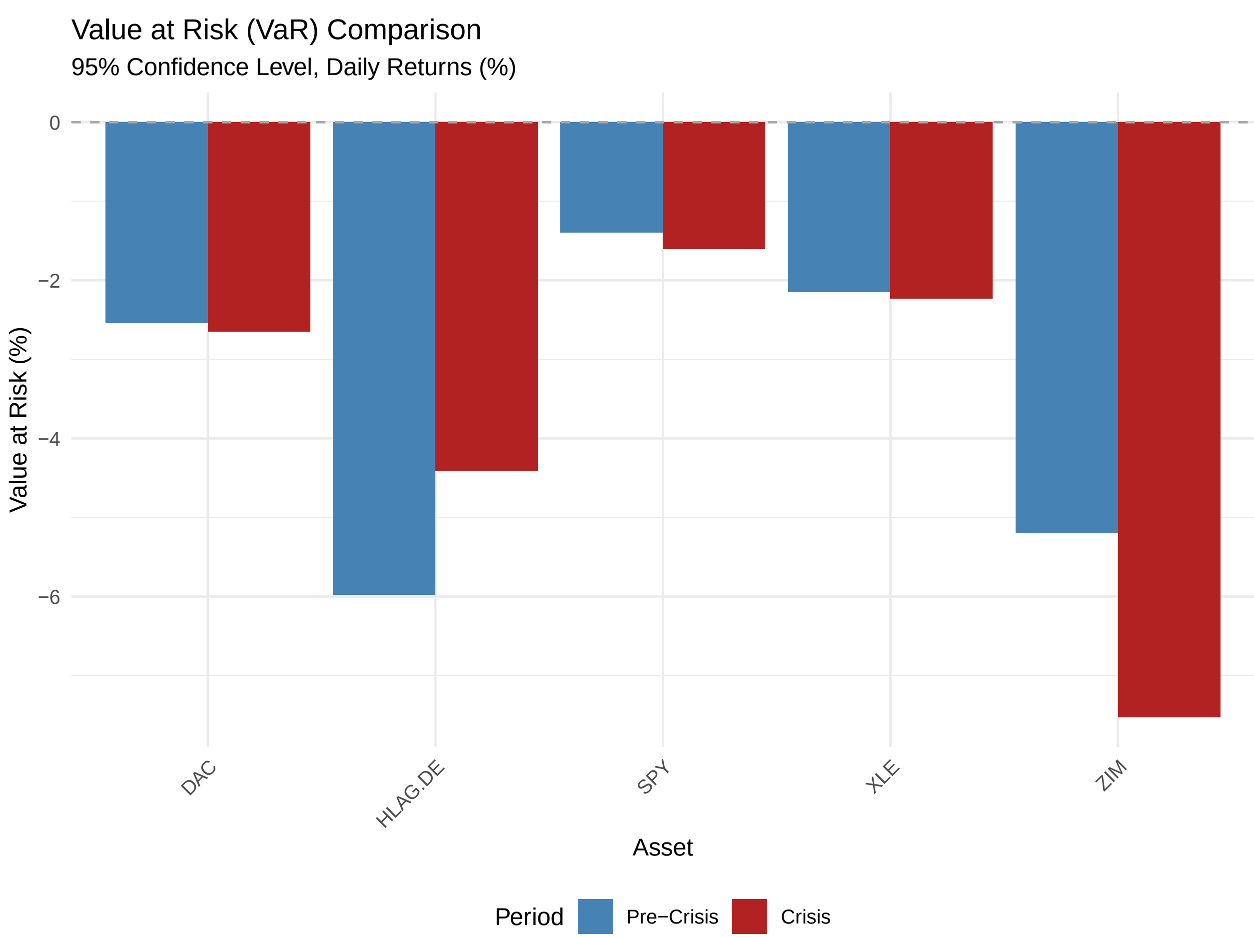
```r
# Calculate VaR
var_results <- compare_var_periods(data$returns,
                                   c(data$shipping_cols, data$market_cols),
                                   data$crisis_start)
```

```r
# Display VaR results
kable(var_results, caption = "Value at Risk Analysis",
      digits = 2, booktabs = TRUE)
```

Table 5: Value at Risk Analysis

Asset	Pre_Crisis_VaR	Crisis_VaR	VaR_Change	VaR_Pct_Change
ZIM	-5.20	-7.53	-2.33	-44.80
HLAG.DE	-5.98	-4.41	1.57	26.27
DAC	-2.54	-2.65	-0.11	-4.27
SPY	-1.39	-1.61	-0.21	-15.11
XLE	-2.15	-2.23	-0.08	-3.86

```r
# Plot VaR comparison
var_plot <- plot_var_comparison(var_results)
print(var_plot)
```



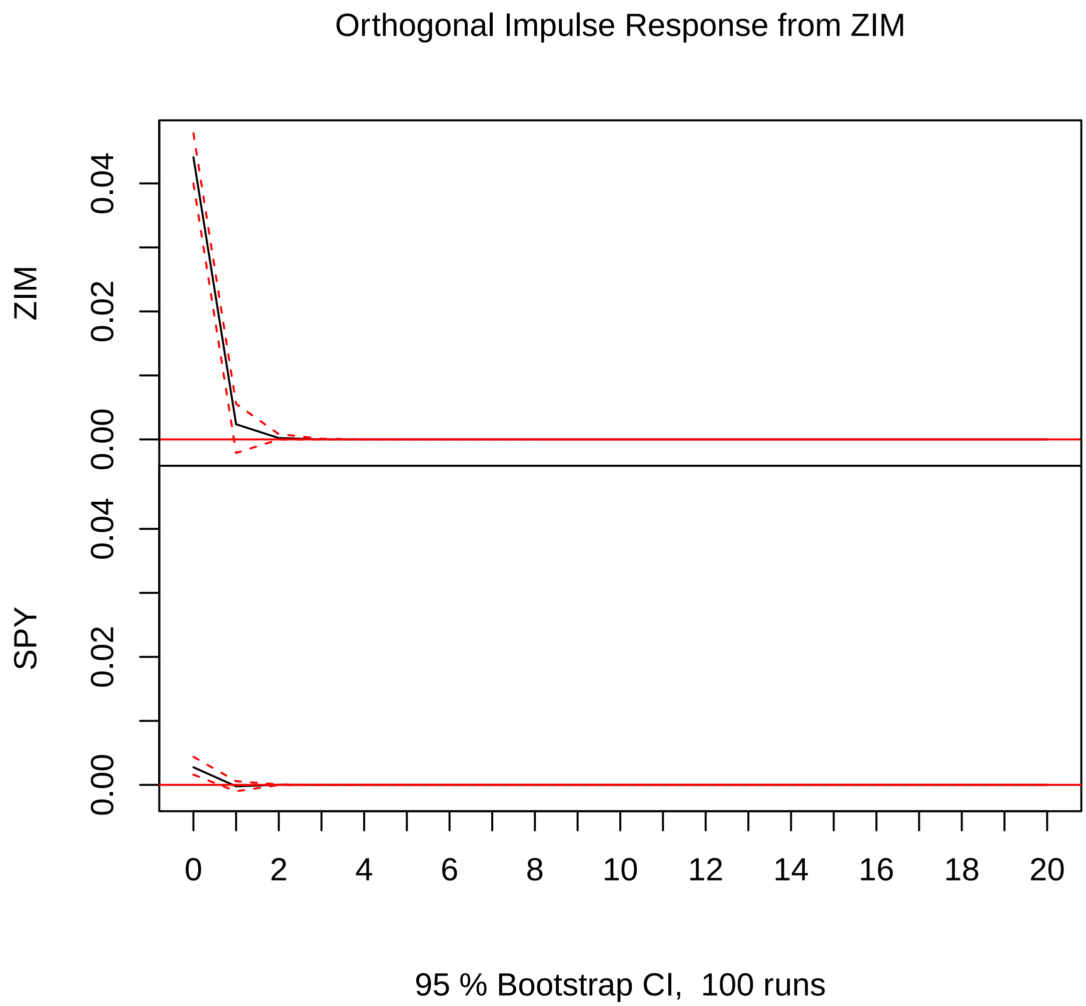

```r
# Analyze VaR changes
var_changes <- analyze_var_changes(var_results)
kable(var_changes$summary, caption = "VaR Changes Analysis",
      digits = 2, booktabs = TRUE)
```

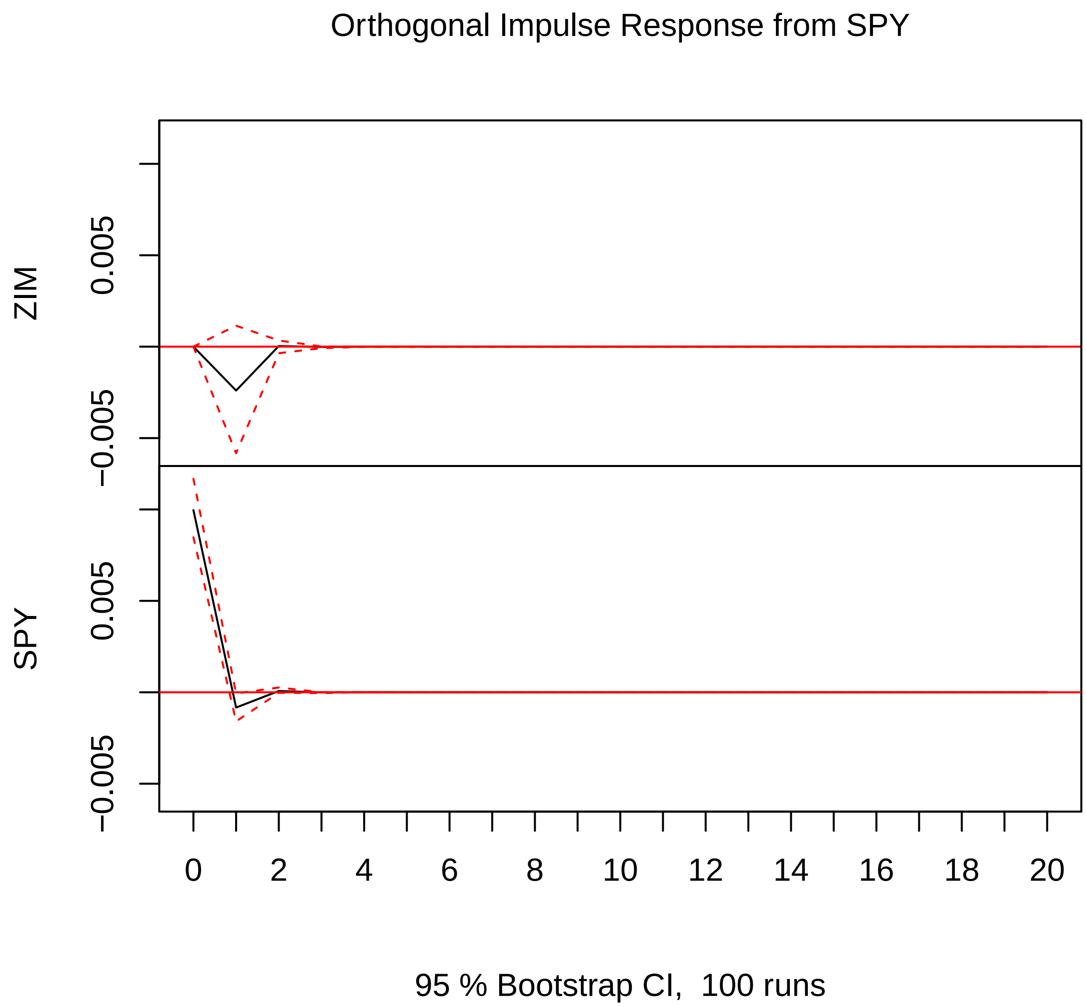
Table 6: VaR Changes Analysis

Mean_VaR_Change	Median_VaR_Change	Max_VaR_Worsening	Max_VaR_Improvement	Mean_Pct_Change	Assets_With_Increased_Risk	Total_Assets	Pct_Assets_With_Increased_Risk
-0.23	-0.11	-2.33	1.57	-8.35	4	5	80

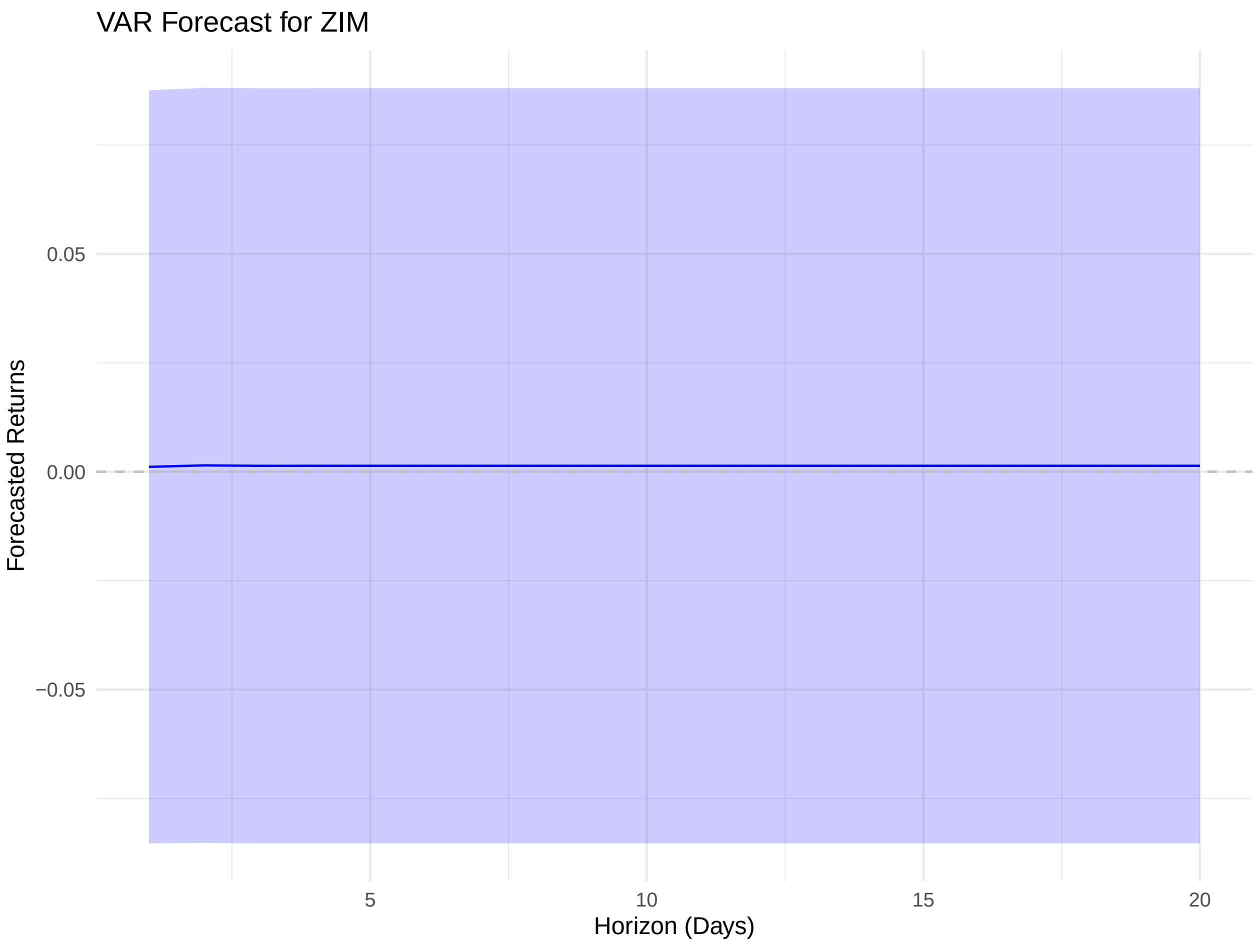
8 Forecasting

```r
# Prepare VAR data
if (length(data$shipping_cols) > 0 && length(data$market_cols) > 0) {
  # Select columns for VAR
  var_cols <- c(data$shipping_cols[1], data$market_cols[1])
  var_data <- data$returns[, var_cols]

  # Run VAR analysis
  var_results <- run_var_analysis(var_data)

  # Generate forecasts
  forecasts <- generate_var_forecasts(var_results$model)
  print(forecasts$plots)

  # ARIMA forecasts for first shipping stock
  if (length(names(shipping_data)) > 0) {
    ticker <- names(shipping_data)[1]
    arima_forecast <- forecast_with_arima(data$prices[, ticker], ticker)
    print(arima_forecast$plot)
  }

  # Validate forecasts
  error_metrics <- validate_forecasts(var_data, var_results$model)
  kable(error_metrics, caption = "Forecast Error Metrics",
        digits = 4, booktabs = TRUE)
}
```


Orthogonal Impulse Response from ZIM
ZIM
SPY
0.00
0.02
0.04
0.00
0.02
0.04
0
2
4
6
8
10
12
14
16
18
20
95 % Bootstrap CI, 100 runs


Orthogonal Impulse Response from SPY
ZIM
SPY
0.005
−0.005
0.005
−0.005
0
2
4
6
8
10
12
14
16
18
20
95 % Bootstrap CI, 100 runs

```
## $ZIM
```


VAR Forecast for ZIM
0.05
0.00
−0.05
Forecasted Returns
5
10
15
20
Horizon (Days)

```
##
## $SPY
```

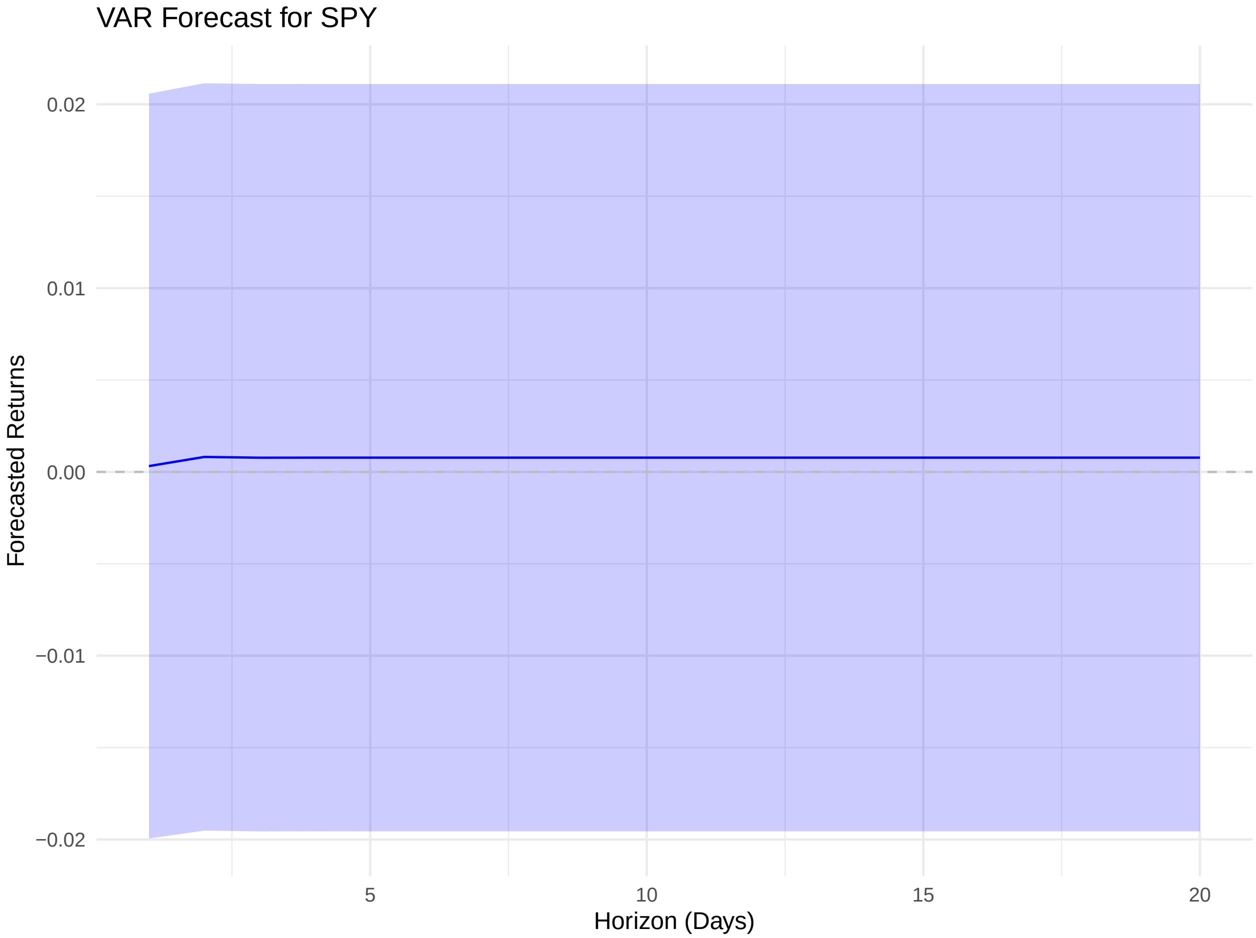

VAR Forecast for SPY
0.02
0.01
0.00
−0.01
−0.02
Forecasted Returns
5
10
15
20
Horizon (Days)

Table 7: Forecast Error Metrics

Asset	MAE	RMSE	MAPE	Theil_U
ZIM	0.0517	0.0683	100.0234	1.0006
SPY	0.0207	0.0322	103.2825	1.0016

9 Hypothesis Testing

```
# Test volatility changes
vol_test <- test_volatility_changes(data$returns, data$shipping_cols, data$crisis_start)
kable(vol_test, caption = "Volatility Change Tests",
      digits = 4, booktabs = TRUE)
```

Table 8: Volatility Change Tests

	Asset	Pre_Crisis_Vol	Crisis_Vol	Change_Pct	F_Stat	P_Value	Significant
F	ZIM	0.0352	0.0488	38.7619	1.9255	0.0000	TRUE
F1	HLAG.DE	0.0316	0.0340	7.6948	1.1598	0.2491	FALSE
F2	DAC	0.0162	0.0171	5.4496	1.1120	0.4101	FALSE

```r
# Test beta changes if available
if (exists("rolling_betas")) {
  beta_test <- test_beta_changes(rolling_betas, data$crisis_start)
  kable(beta_test, caption = "Beta Change Tests",
        digits = 4, booktabs = TRUE)
}

# Test event significance
if (!is.null(event_returns)) {
  event_test <- test_event_significance(event_returns, data$shipping_cols)
  kable(event_test, caption = "Event Impact Tests",
        digits = 4, booktabs = TRUE)
}
```

Table 9: Event Impact Tests

	Asset	Mean_Event_Return	T_Statistic	P_Value	Significant
t	ZIM	7.3990	0.3549	0.7566	FALSE
t1	HLAG.DE	-6.5906	-0.5055	0.6634	FALSE
t2	DAC	-0.9518	-0.2308	0.8389	FALSE

```r
# Test VaR changes
if (exists("var_changes")) {
  var_test <- test_var_changes(var_changes)
  kable(var_test, caption = "VaR Change Tests",
        digits = 4, booktabs = TRUE)
}
```

Table 10: VaR Change Tests

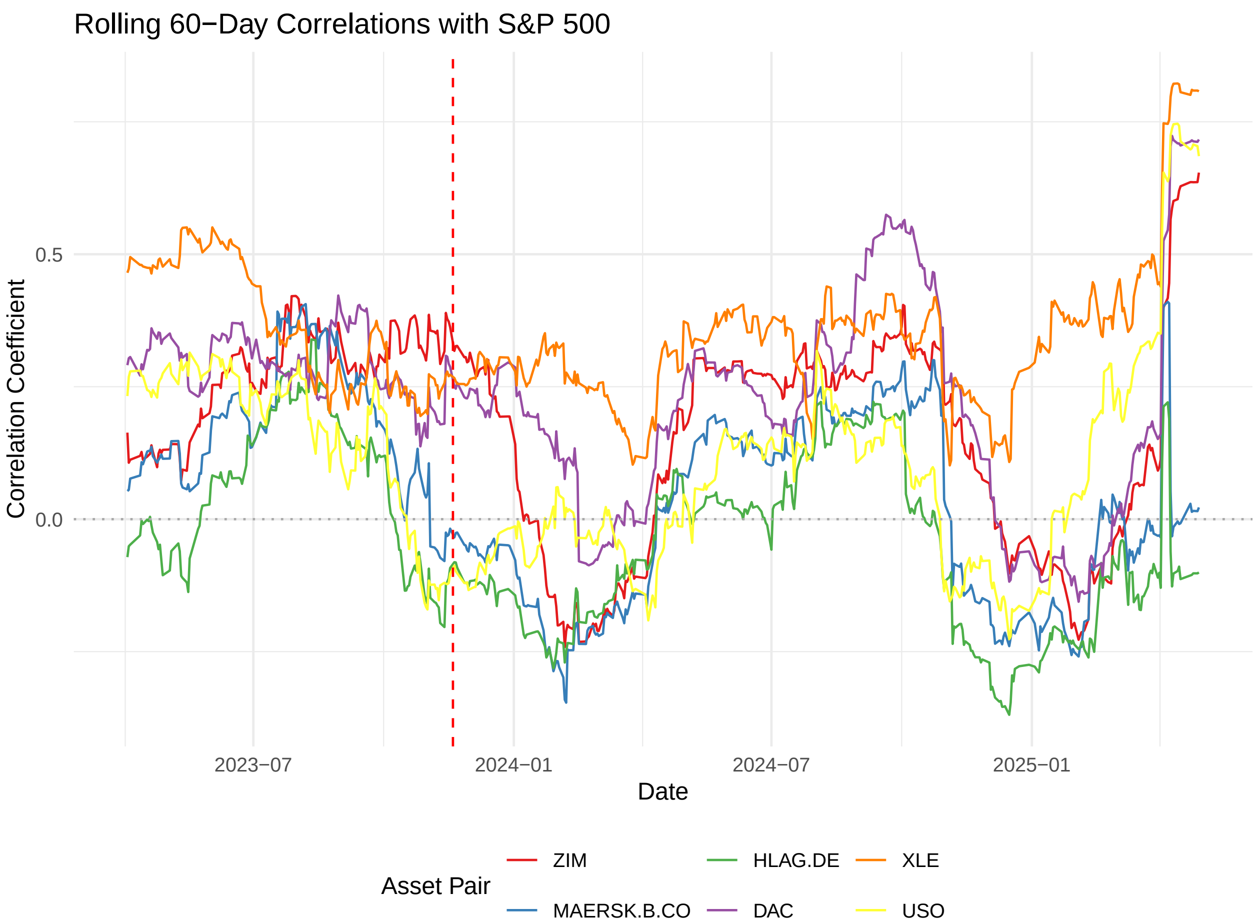
	Test	Statistic	P_Value	Significant
t	T-test for VaR Change	2.5	0.1296	FALSE

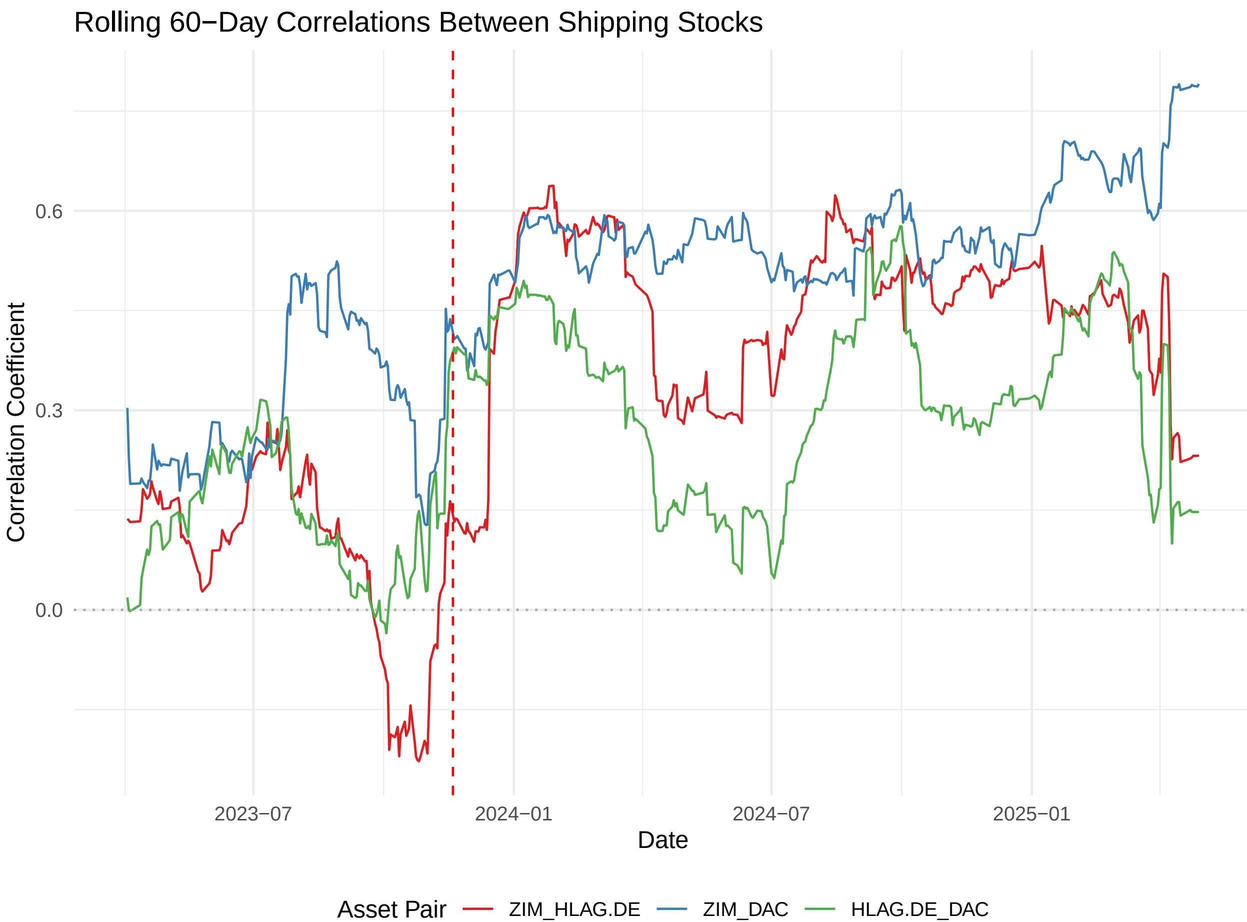
10 Enhanced Analysis

10.1 Rolling Correlations Analysis

```r
# Calculate rolling correlations between shipping and market
if ("SPY" %in% colnames(data$returns)) {
  # Calculate rolling correlations with market
  rolling_cors <- calculate_rolling_correlations(data$returns, window_size = 60, market_index = "SPY")

  # Plot rolling correlations
  rolling_cors_plot <- plot_rolling_correlations(rolling_cors, data$crisis_start,
                                                "Rolling 60-Day Correlations with S&P 500")
  print(rolling_cors_plot)

  # Calculate shipping correlations among themselves
  if (length(data$shipping_cols) > 1) {
    shipping_cors <- calculate_rolling_correlations(data$returns[, data$shipping_cols], window_size = 6
    shipping_cors_plot <- plot_rolling_correlations(shipping_cors, data$crisis_start,
```

```r
                                              "Rolling 60-Day Correlations Between Shipping Stocks")
    print(shipping_cors_plot)
  }
}
```





10.2 GARCH Volatility Modeling

```
# Check if rugarch is installed
if (!requireNamespace("rugarch", quietly = TRUE)) {
  # Only run this if you want to install the package
  # install.packages("rugarch")
  cat("Package 'rugarch' is required for GARCH analysis but not available.\n")
} else {
  # Load the library
  library(rugarch)

  # Run GARCH analysis on shipping stocks
  garch_models <- list()
  garch_vols <- list()

  # Analyze selected assets
  assets_to_analyze <- c(data$shipping_cols[1], data$market_cols[1]) # First shipping stock and market

  for (asset in assets_to_analyze) {
    if (asset %in% colnames(data$returns)) {
      # Run GARCH model
      garch_result <- run_garch_analysis(data$returns[, asset], asset)
      garch_models[[asset]] <- garch_result
```

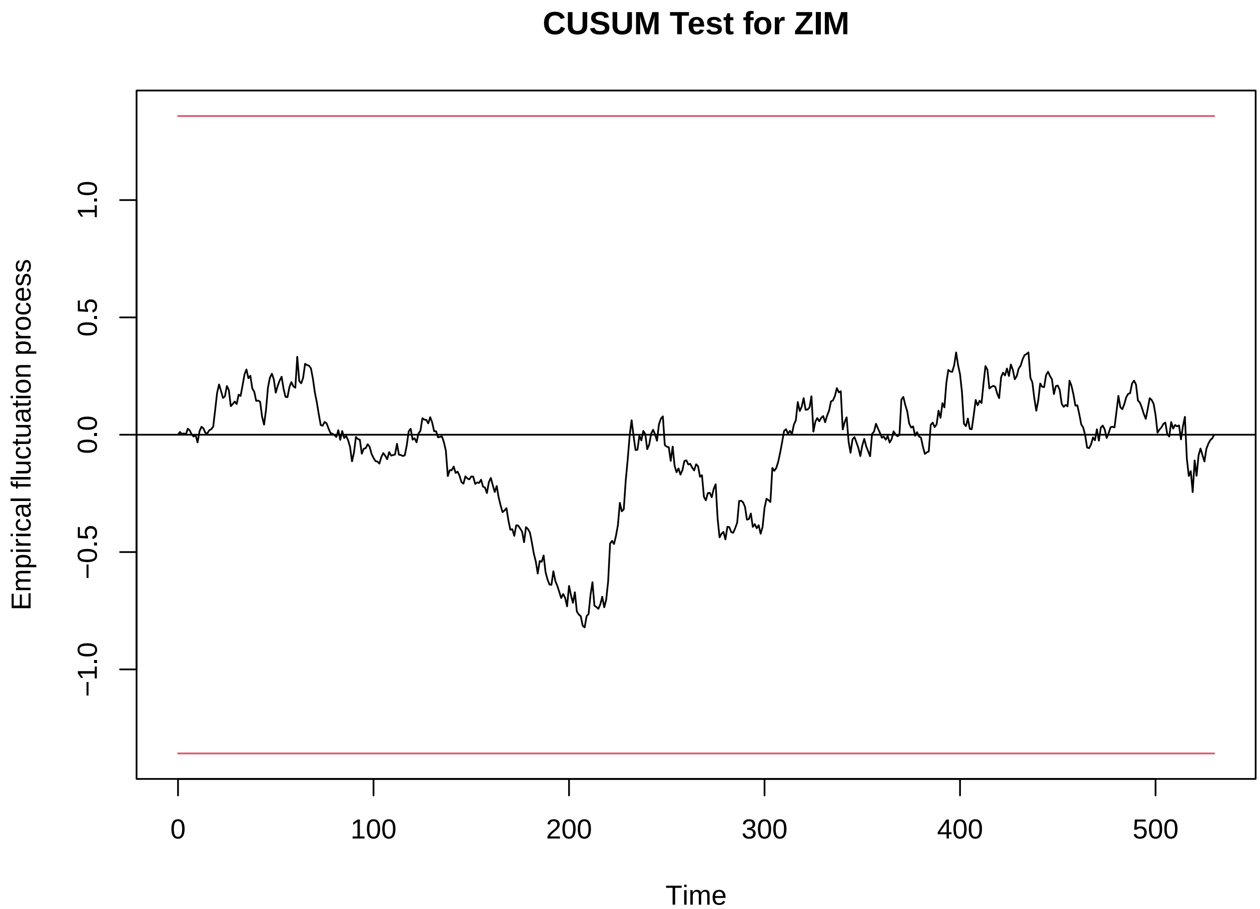
```r
      # Extract volatility
      if (!is.null(garch_result$model)) {
        vol <- extract_garch_volatility(garch_result, data$returns[, asset])
        garch_vols[[asset]] <- vol

        # Display GARCH parameters
        cat("GARCH Model Parameters for", asset, ":\n")
        print(rugarch::coef(garch_result$model))

        # Plot model diagnostics
        plot(garch_result$model, which = 2) # Conditional SD plot
        plot(garch_result$model, which = 8) # News Impact Curve
      }
    }
  }

  # Plot conditional volatilities
  if (length(garch_vols) > 0) {
    garch_vol_plot <- plot_garch_volatility(garch_vols, data$crisis_start)
    print(garch_vol_plot)

    # Compare pre/post-crisis volatility persistence
    persistence_results <- data.frame(
      Asset = character(),
      Pre_Crisis_Persistence = numeric(),
      Crisis_Persistence = numeric(),
      Persistence_Change = numeric(),
      stringsAsFactors = FALSE
    )

    for (asset in names(garch_models)) {
      if (!is.null(garch_models[[asset]]$model)) {
        model_coef <- rugarch::coef(garch_models[[asset]]$model)
        persistence <- sum(model_coef[grepl("alpha|beta", names(model_coef))])

        persistence_results <- rbind(persistence_results, data.frame(
          Asset = asset,
          GARCH_Persistence = persistence,
          Half_Life_Days = log(0.5) / log(persistence),
          stringsAsFactors = FALSE
        ))
      }
    }

    kable(persistence_results, caption = "GARCH Volatility Persistence",
          digits = 4, booktabs = TRUE)
  }
}
```

```
## Package 'rugarch' is required for GARCH analysis but not available.
```

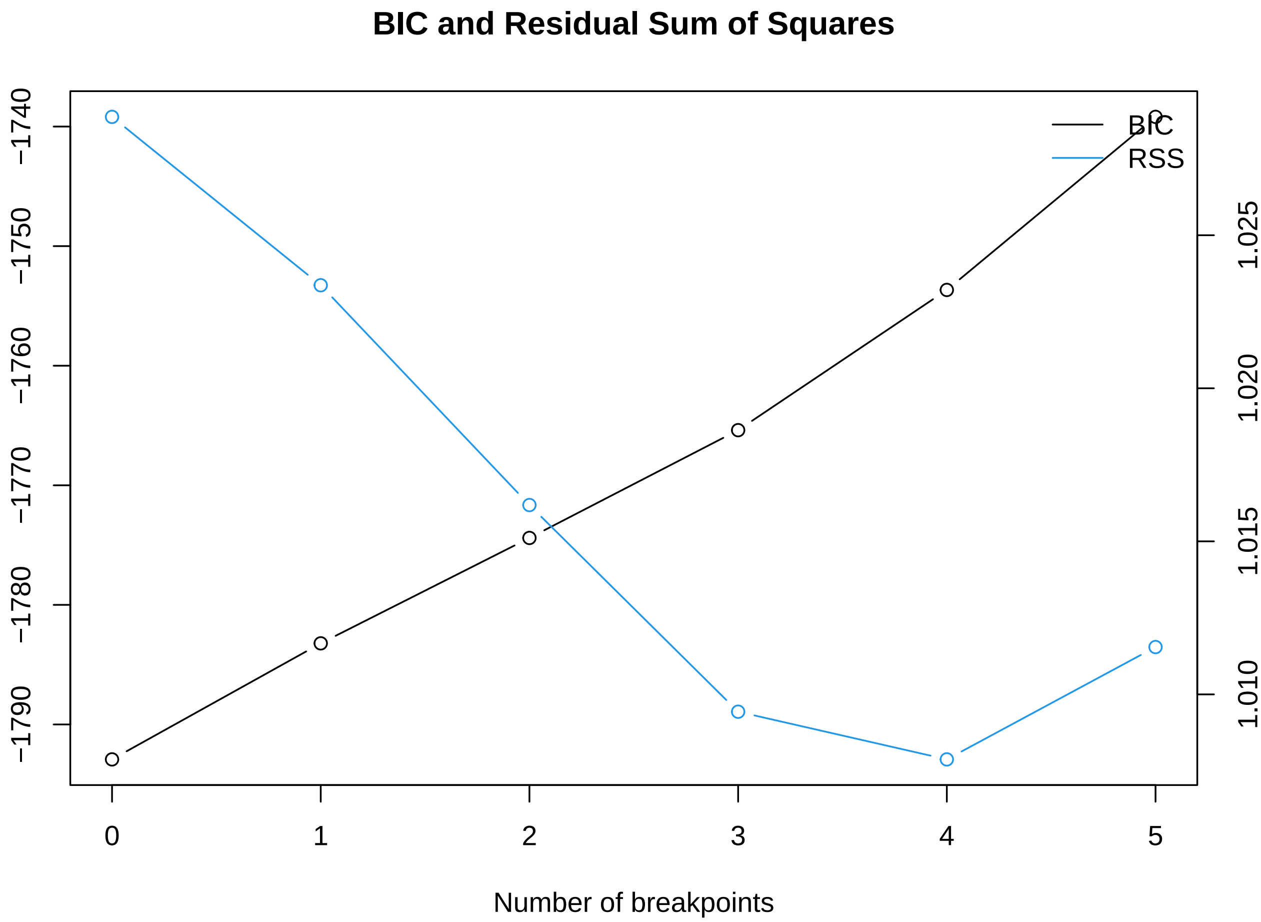
10.3 Structural Break Testing

```r
# Check if strucchange is installed
if (!requireNamespace("strucchange", quietly = TRUE)) {
  # Only run this if you want to install the package
  # install.packages("strucchange")
  cat("Package 'strucchange' is required for structural break testing but not available.\n")
} else {
  # Load the library
  library(strucchange)

  # Test for structural breaks in shipping stocks
  break_results <- list()
  for (asset in data$shipping_cols) {
    if (asset %in% colnames(data$returns)) {
      break_test <- test_structural_breaks(data$returns[, asset], data$crisis_start)
      break_results[[asset]] <- break_test

      # Plot CUSUM test
      if (!is.null(break_test$cusum)) {
        plot(break_test$cusum, main = paste("CUSUM Test for", asset))

        # Report Chow test results
        if (!is.null(break_test$chow)) {
          cat("\nChow Test for", asset, "at Red Sea Crisis Date:\n")
          print(break_test$chow)
        }

        # Plot breakpoints
        if (!is.null(break_test$breakpoints)) {
          # Plot with breakpoints
          breakdates <- breakpoints(break_test$breakpoints)
          plot(break_test$breakpoints)

          # Print estimated break dates
          cat("\nEstimated Break Dates for", asset, ":\n")
          print(breakdates)
        }
      }
    }
  }

  # Summarize structural break results
  break_summary <- data.frame(
    Asset = character(),
    Crisis_Date_Is_Break = logical(),
    Chow_Statistic = numeric(),
    Chow_P_Value = numeric(),
    Detected_Breaks = character(),
    stringsAsFactors = FALSE
  )

  for (asset in names(break_results)) {
    result <- break_results[[asset]]
```

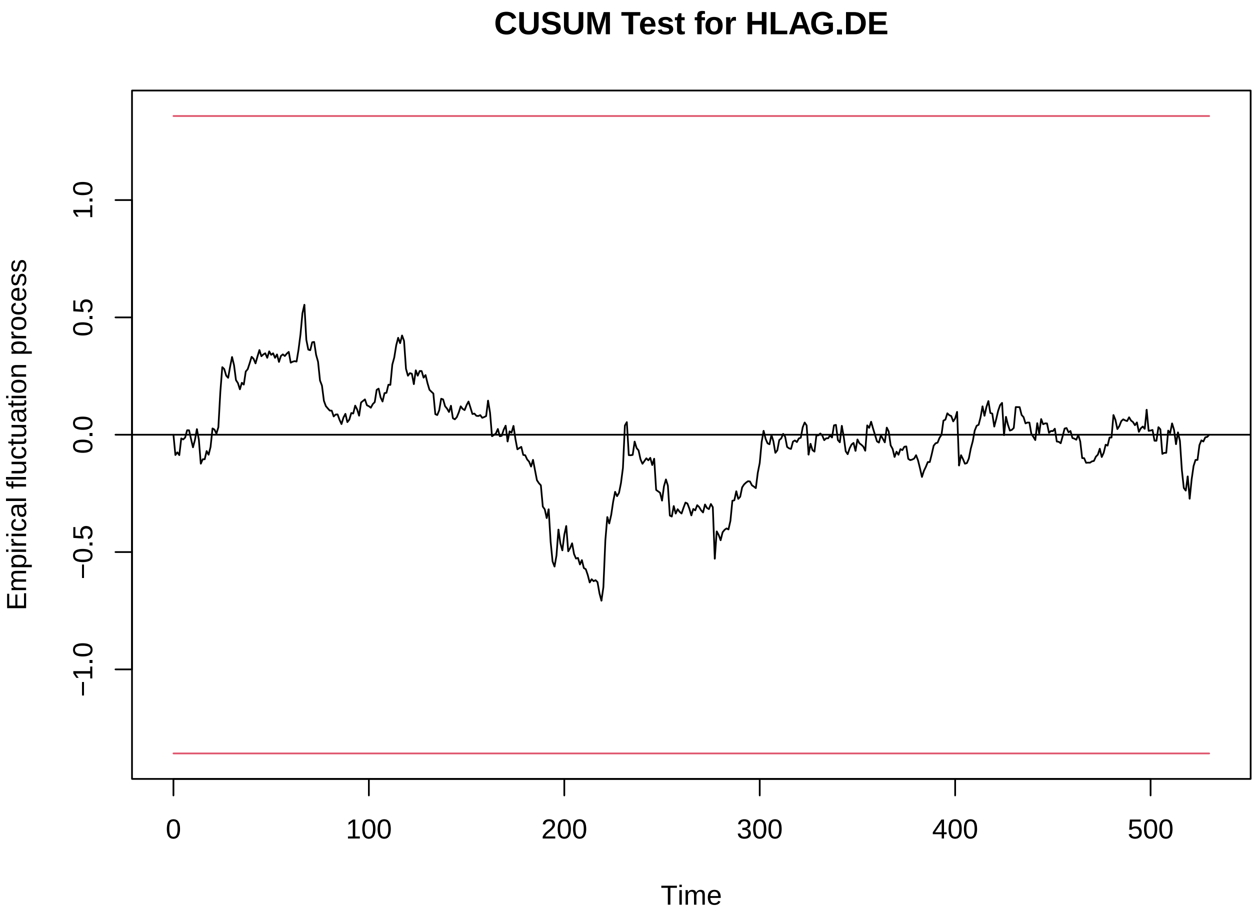
```r
    # Check if Chow test available
    chow_stat <- NA
    chow_pval <- NA
    is_break <- FALSE
    if (!is.null(result$chow)) {
      chow_stat <- result$chow$statistic
      chow_pval <- result$chow$p.value
      is_break <- chow_pval < 0.05
    }

    # Get detected breaks
    break_dates <- "None detected"
    if (!is.null(result$breakpoints) && !is.null(result$breakpoints$breakpoints)) {
      dates <- index(data$returns)[result$breakpoints$breakpoints]
      if (length(dates) > 0) {
        break_dates <- paste(as.character(dates), collapse = ", ")
      }
    }

    # Add to summary
    break_summary <- rbind(break_summary, data.frame(
      Asset = asset,
      Crisis_Date_Is_Break = is_break,
      Chow_Statistic = chow_stat,
      Chow_P_Value = chow_pval,
      Detected_Breaks = break_dates,
      stringsAsFactors = FALSE
    ))
  }

  kable(break_summary, caption = "Structural Break Test Results",
        digits = 4, booktabs = TRUE)
}
```



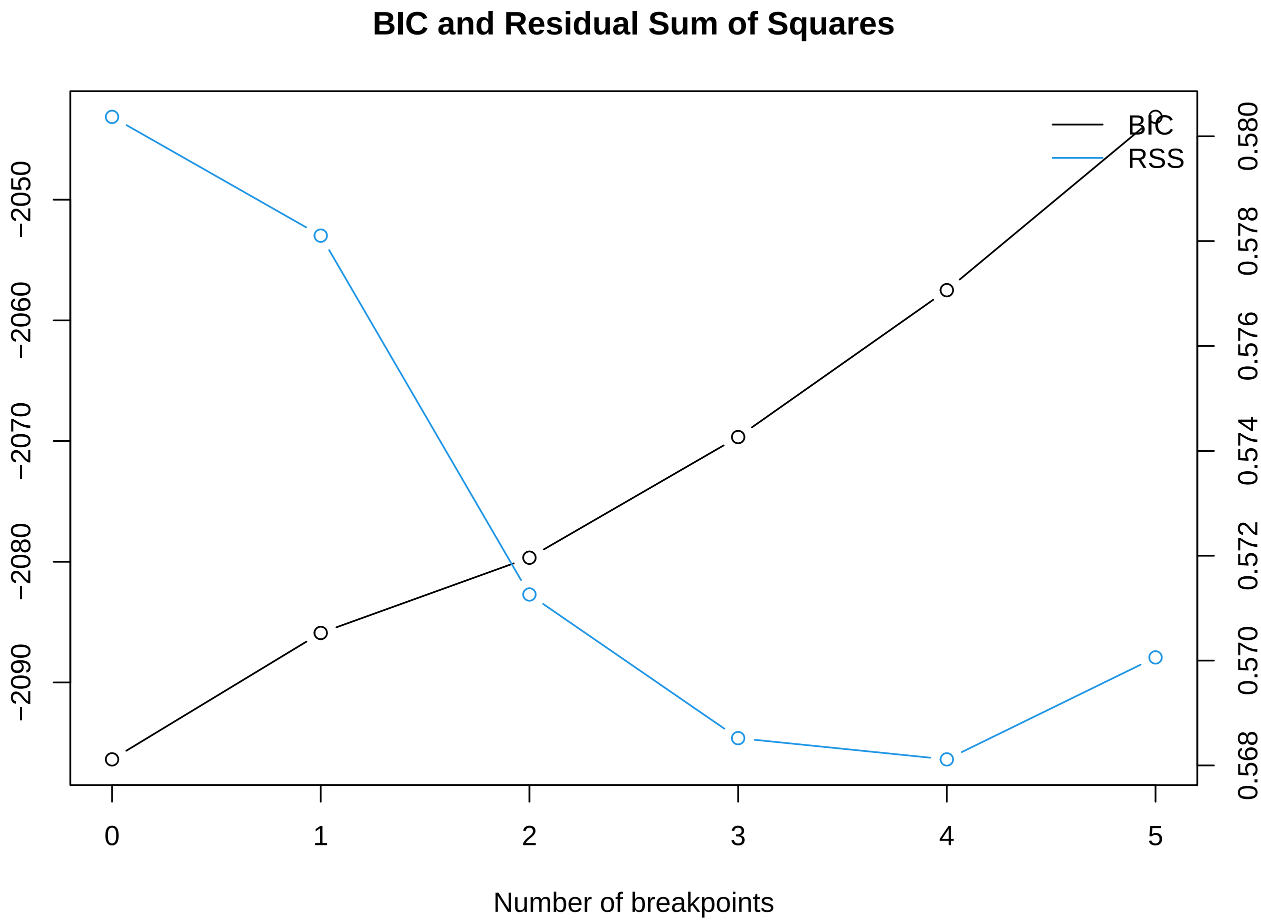

```
##
## Chow Test for ZIM at Red Sea Crisis Date:
##
##  Chow test
##
## data:  returns_ts ~ 1
## F = 2.4044, p-value = 0.1216
```



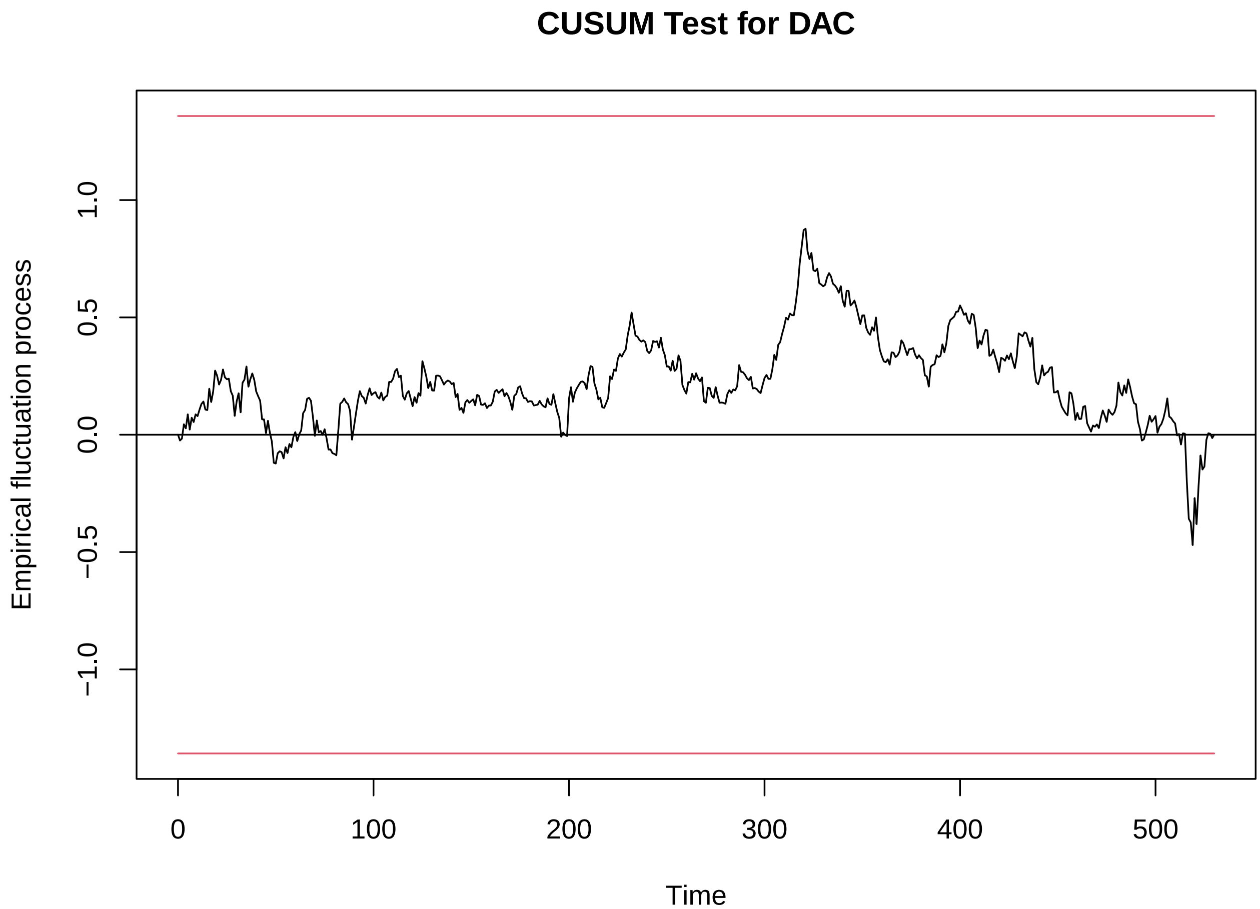

```
##
## Estimated Break Dates for ZIM :
##
##   Optimal 1-segment partition:
##
## Call:
## breakpoints.breakpointsfull(obj = break_test$breakpoints)
##
## Breakpoints at observation number:
## NA
##
## Corresponding to breakdates:
## NA
```



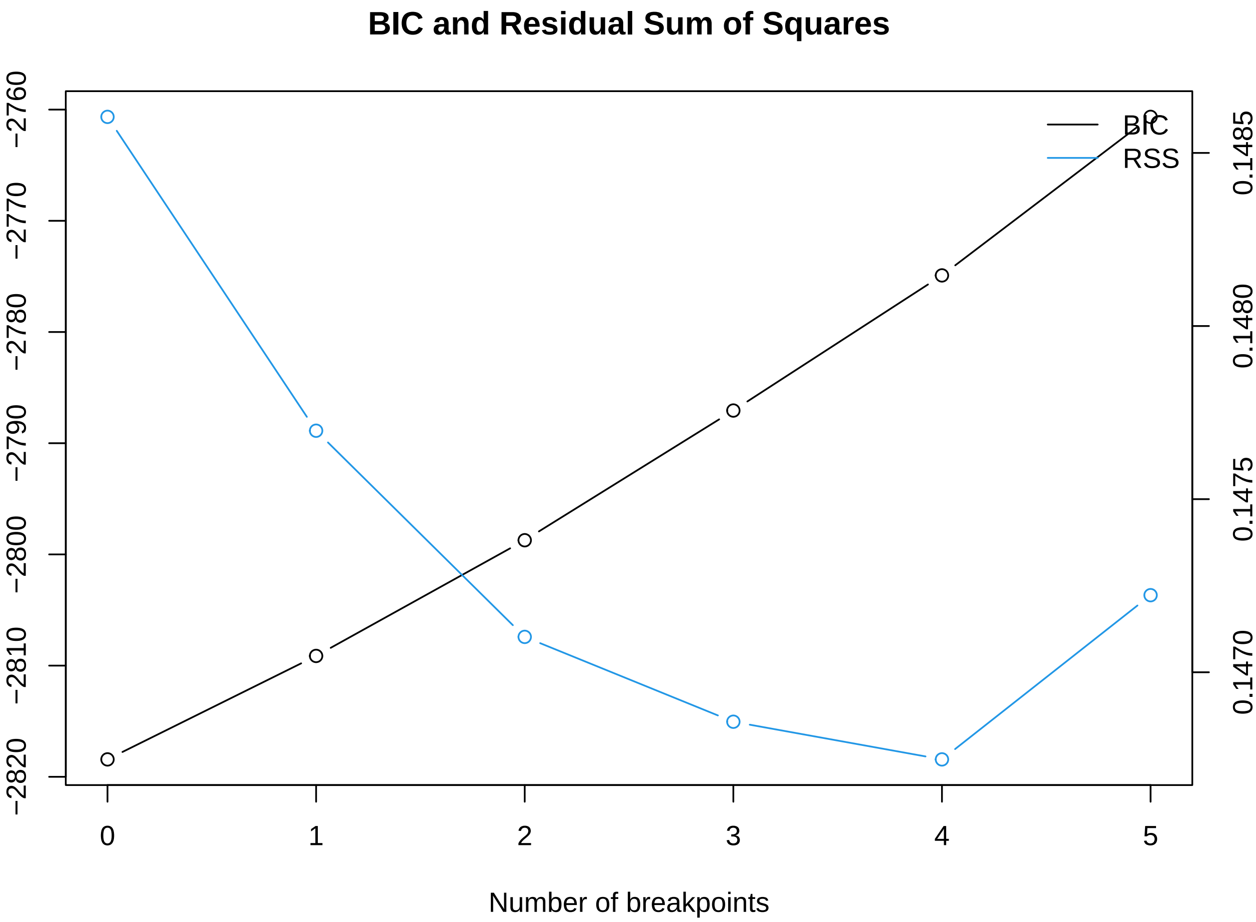

```
##
## Chow Test for HLAG.DE at Red Sea Crisis Date:
##
##  Chow test
##
## data:  returns_ts ~ 1
## F = 0.90345, p-value = 0.3423
```




```
##
## Estimated Break Dates for HLAG.DE :
##
##   Optimal 1-segment partition:
##
## Call:
## breakpoints.breakpointsfull(obj = break_test$breakpoints)
##
## Breakpoints at observation number:
## NA
##
## Corresponding to breakdates:
## NA
```




```
##
## Chow Test for DAC at Red Sea Crisis Date:
##
##  Chow test
##
## data:  returns_ts ~ 1
## F = 0.1651, p-value = 0.6847
```




```
##
## Estimated Break Dates for DAC :
##
##   Optimal 1-segment partition:
##
## Call:
## breakpoints.breakpointsfull(obj = break_test$breakpoints)
##
## Breakpoints at observation number:
## NA
##
## Corresponding to breakdates:
## NA
```

Table 11: Structural Break Test Results

	Asset	Crisis	[illegible]	Breaks
F	ZIM	FALSE	2014-12-16	NA, NA

Asset [illegible] Breaks	
F1 HL[illegible] 0.3423	NA, NA

Asset	Crisis_Shock	Chow_Stat	Chow_P_Value	Breaks
F2	DAX	FALSE	0.6551847	NA, NA

10.4 Economic Significance Assessment

```r
# Analyze economic significance of the crisis
econ_results <- analyze_economic_significance(data$prices, data$returns,
                                              c(data$shipping_cols, data$market_cols),
                                              data$crisis_start)

# Display price change results
kable(econ_results$price_changes, caption = "Economic Impact: Price Changes",
      digits = 2, booktabs = TRUE)
```

Table 12: Economic Impact: Price Changes

Asset	Pre_Crisis_Price	Post_Crisis_Price	Price_Change_Pct	Annualized_Return_Pre	Annualized_Return_Post	Return_Difference
HLAG.DE	107.07	130.45	21.83	-15.79	14.35	30.14

```r
# Display risk-adjusted returns
kable(econ_results$risk_adjusted, caption = "Economic Impact: Risk-Adjusted Returns",
      digits = 4, booktabs = TRUE)
```

Table 13: Economic Impact: Risk-Adjusted Returns

Asset	Pre_Crisis_Sharpe	Crisis_Sharpe	Sharpe_Change
ZIM	-0.9477	1.0882	2.0359
HLAG.DE	-0.8820	0.5522	1.4343
DAC	1.0516	0.4826	-0.5690
SPY	1.5266	0.8012	-0.7255
XLE	0.3014	-0.2184	-0.5199

```r
# Calculate implied cost increases (for shipping companies)
shipping_indices <- data$shipping_cols
if (length(shipping_indices) > 0) {
  # Calculate weighted average impact
  avg_price_change <- mean(econ_results$price_changes$Price_Change_Pct[
    econ_results$price_changes$Asset %in% shipping_indices], na.rm = TRUE)

  # Calculate economic impact estimates
  impact_estimates <- data.frame(
    Metric = c("Avg. Stock Price Change (%)",
               "Est. Shipping Rate Change (%)",
               "Est. Annual Impact ($B)",
               "Est. Trip Extension (days)",
               "Est. Added Fuel Costs (%)"),
    Value = c(avg_price_change,
              avg_price_change * 1.5, # Estimated relationship between stock prices and shipping rates
              20 * (avg_price_change * 1.5) / 100, # Global container shipping ~$20B annually
              8, # Cape of Good Hope adds ~8 days
              30), # Fuel cost increase due to longer route
    stringsAsFactors = FALSE
  )

  kable(impact_estimates, caption = "Estimated Red Sea Crisis Economic Impact",
        digits = 2, booktabs = TRUE)
}
```

Table 14: Estimated Red Sea Crisis Economic Impact

Metric	Value
Avg. Stock Price Change (%)	21.83
Est. Shipping Rate Change (%)	32.74
Est. Annual Impact ($B)	6.55
Est. Trip Extension (days)	8.00
Est. Added Fuel Costs (%)	30.00