Anish Ketha - Project 2: Create a Market-Timing Strategy

Create a Market-Timing Strategy

Introduction

The purpose of this project is to create and evaluate a market-timing strategy. The strategy will use both technical and fundamental analysis to identify potential investment opportunities.

Strategy Development:

- **Data Preparation**: Loads and prepares S&P 500 and economic indicators data.

- **Fundamental Analysis**: Uses the actual Price-to-Dividend ratio data from the `econ_indicators` dataset.

- **Technical Analysis:** Implements SMA-based trading signals.

- **Market Timing Strategy Function**: Implements the strategy function with trading signals based on the Price-to-Dividend ratio.

- **Backtesting:** Applies the strategy to both fundamental and technical analyses

- **Conclusion**: Summarizes the findings

Data Preparation

```
library(quantmod)
```

```
Loading required package: xts

Loading required package: zoo


Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric

Loading required package: TTR

Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
```

```
# Load S&P 500 data
spy <- getSymbols("SPY", src = "yahoo", auto.assign = FALSE, from = '1993-01-01', to = '2024-06-30')
spy_ret <- monthlyReturn(Ad(spy))
index(spy_ret) <- as.Date(as.yearmon(index(spy_ret)), frac = 1)
colnames(spy_ret) <- "Ret"

# Load risk-free rate data
rf <- getSymbols("DFF", src = "FRED", auto.assign = FALSE, from = '1993-01-01', to = '2024-06-30')
monthly_endpoints <- endpoints(rf, on = "months")
rf <- rf[monthly_endpoints]/12/100
colnames(rf) <- "Rf"
```

```
return_data <- merge(spy_ret, rf, join = "inner")
head(return_data, 10)
```

```
                  Ret          Rf
1993-01-31  0.000000000 0.002516667
1993-02-28  0.010669181 0.002650000
1993-03-31  0.022399070 0.003191667
1993-04-30 -0.025588400 0.002508333
1993-05-31  0.026970621 0.002516667
1993-06-30  0.003606818 0.003266667
1993-07-31 -0.004854379 0.002558333
1993-08-31  0.038327048 0.002658333
1993-09-30 -0.007275322 0.003325000
1993-10-31  0.019727682 0.002525000
```

```
tail(return_data, 10)
```

```
                 Ret          Rf
2023-09-30 -0.04743449 0.004441667
2023-10-31 -0.02170868 0.004441667
2023-11-30  0.09134385 0.004441667
2023-12-31  0.04565531 0.004441667
2024-01-31  0.01592643 0.004441667
2024-02-29  0.05218690 0.004441667
2024-03-31  0.03270193 0.004441667
2024-04-30 -0.04031960 0.004441667
2024-05-31  0.05057970 0.004441667
2024-06-30  0.03528010 0.004441667
```

Fundamental Analysis: Price-to-Dividend Ratio

```
# Load economic indicators data
econ_indicators_url <- "https://www.dropbox.com/scl/fi/gi2ou5sl1ved66v890p41/Monthly-Market-Return-Predictors.csv?r
econ_indicators <- read.csv(econ_indicators_url)

# Inspect the first few rows and column names to ensure data is loaded correctly
head(econ_indicators)
```

```
  yyyymm  D12 E12 BM TBL AAA BAA LTY NTIS INFL LTR SVAR CSP
1 187101 0.26 0.4 NA  NA  NA  NA  NA   NA   NA  NA   NA  NA
2 187102 0.26 0.4 NA  NA  NA  NA  NA   NA   NA  NA   NA  NA
3 187103 0.26 0.4 NA  NA  NA  NA  NA   NA   NA  NA   NA  NA
4 187104 0.26 0.4 NA  NA  NA  NA  NA   NA   NA  NA   NA  NA
5 187105 0.26 0.4 NA  NA  NA  NA  NA   NA   NA  NA   NA  NA
6 187106 0.26 0.4 NA  NA  NA  NA  NA   NA   NA  NA   NA  NA
```

```
str(econ_indicators)  # This will show the structure and column names
```

```
'data.frame':   1836 obs. of  13 variables:
 $ yyyymm: int  187101 187102 187103 187104 187105 187106 187107 187108 187109 187110 ...
 $ D12   : num  0.26 0.26 0.26 0.26 0.26 0.26 0.26 0.26 0.26 0.26 ...
 $ E12   : num  0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 ...
 $ BM    : num  NA NA NA NA NA NA NA NA NA NA ...
 $ TBL   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ AAA   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ BAA   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ LTY   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ NTIS  : num  NA NA NA NA NA NA NA NA NA NA ...
 $ INFL  : num  NA NA NA NA NA NA NA NA NA NA ...
```

```
 $ LTR   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ SVAR  : num  NA NA NA NA NA NA NA NA NA NA ...
 $ CSP   : num  NA NA NA NA NA NA NA NA NA NA ...
```

```
# Convert data to xts
econ_indicators$Date <- as.Date(paste0(econ_indicators$yyyymm, "01"), format = "%Y%m%d")
econ_indicators$Date <- as.Date(as.yearmon(econ_indicators$Date), frac = 1)
econ_indicators$yyyymm <- NULL
econ_indicators <- xts(econ_indicators[,-ncol(econ_indicators)], order.by = econ_indicators$Date)

# Check the structure and column names after conversion
str(econ_indicators)
```

```
An xts object on 1871-01-31 / 2023-12-31 containing:
  Data:    double [1836, 12]
  Columns: D12, E12, BM, TBL, AAA ... with 7 more columns
  Index:   Date [1836] (TZ: "UTC")
```

```
colnames(econ_indicators)
```

```
 [1] "D12"  "E12"  "BM"   "TBL"  "AAA"  "BAA"  "LTY"  "NTIS" "INFL" "LTR"
[11] "SVAR" "CSP"
```

```
# Calculate Price-to-Dividend ratio
if("CSP" %in% colnames(econ_indicators) & "D12" %in% colnames(econ_indicators)) {
    # Calculate the Price-to-Dividend ratio
    econ_indicators$PriceToDividend <- econ_indicators$CSP / econ_indicators$D12

    # Display the first and last few rows to verify the calculation
    head(econ_indicators$PriceToDividend, 10)
    tail(econ_indicators$PriceToDividend, 10)
} else {
    stop("Columns CSP and/or D12 not found in the dataset.")
}
```

```
           PriceToDividend
2023-03-31              NA
2023-04-30              NA
2023-05-31              NA
2023-06-30              NA
2023-07-31              NA
2023-08-31              NA
2023-09-30              NA
2023-10-31              NA
2023-11-30              NA
2023-12-31              NA
```

Market Timing Strategy

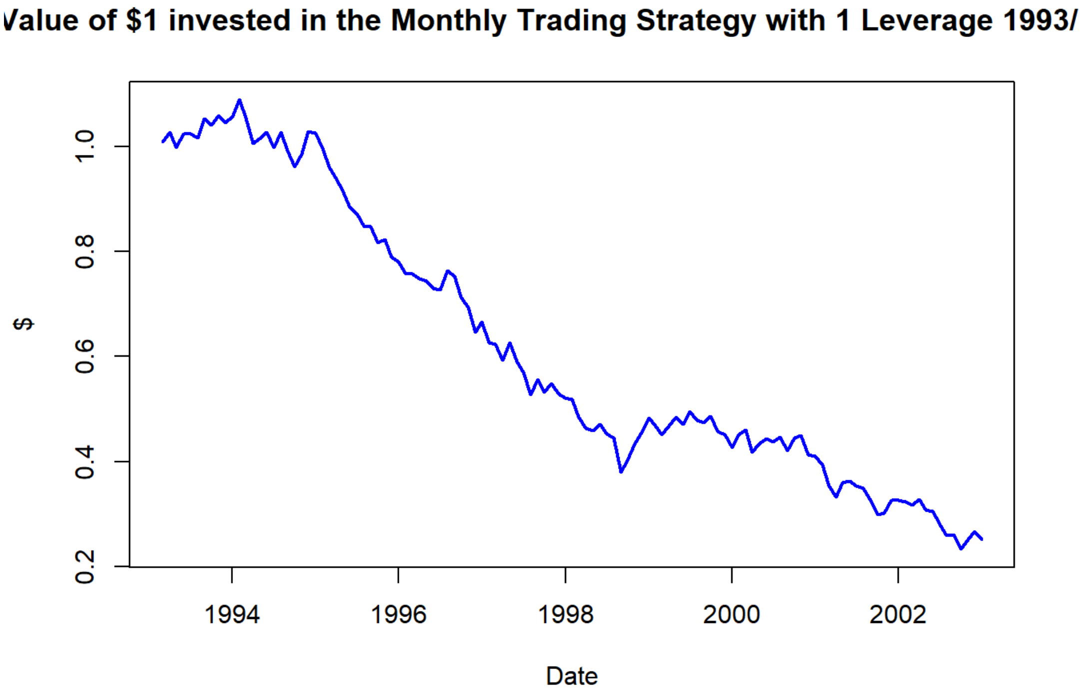
```
market_timing_strategy <- function(data, TradingRule, Leverage = 1, Frequency = "Monthly", TradingPeriod){
    # Generate trading signals
    data$RetRf <- data$Ret - data$Rf
    data$TradingRule <- as.numeric(lag(TradingRule, 1))

    data <- na.omit(data[TradingPeriod])

    # Calculate cumulative returns
    data$Strategy_RetRf <- Leverage * data$TradingRule * data$RetRf
    data$Strategy_Cumulative_Value <- cumprod(1 + data$Strategy_RetRf)
```

```r
    if (Frequency == "Daily") {
      annual_factor <- 252
    } else if (Frequency == "Monthly") {
      annual_factor <- 12
    } else {
      annual_factor <- 1
    }

    annualized_return <- mean(data$Strategy_RetRf) * annual_factor
    annualized_volatility <- sd(data$Strategy_RetRf) * sqrt(annual_factor)
    sharpe_ratio <- annualized_return / annualized_volatility

    print(paste('Annualized Excess Return: ', round(annualized_return * 100, 2), '%'))
    print(paste('Annualized Volatility: ', round(annualized_volatility * 100, 2), '%'))
    print(paste('Annualized Sharpe Ratio: ', round(sharpe_ratio, 2)))

    plot(index(data), data$Strategy_Cumulative_Value, type = "l", col = "blue", lwd = 2,
         ylab = "$", xlab = "Date",
         main = paste('Value of $1 invested in the', Frequency, 'Trading Strategy with', Leverage, 'Leverage', Trac

    return(data)
}
```

Backtesting

```r
# Merge and prepare test data
#Fundamental Analysis: Uses the Price-to-Dividend ratio for testing.
test_data <- merge(return_data, econ_indicators[, "PriceToDividend"], join = "inner")
test_data$TradingSignal <- ifelse(test_data$PriceToDividend < median(test_data$PriceToDividend, na.rm = TRUE), 1, -

# Apply the strategy function
data_backtest <- market_timing_strategy(data = test_data, TradingRule = test_data$TradingSignal, Frequency = "Month
```

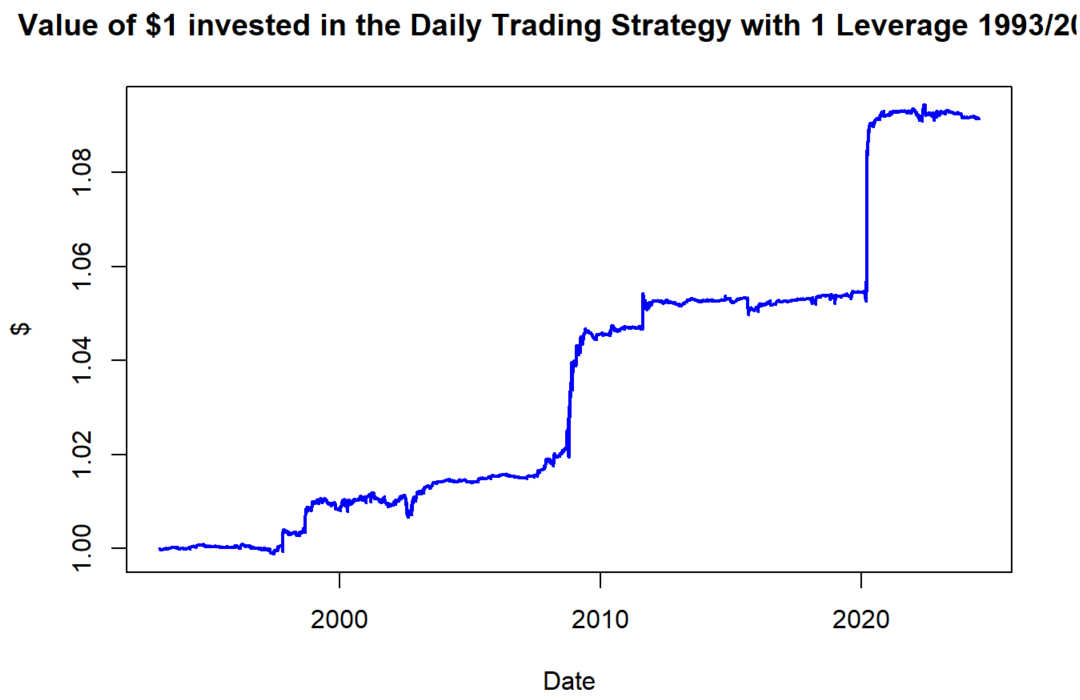
```
[1] "Annualized Excess Return:  -12.67 %"
[1] "Annualized Volatility:  15.03 %"
[1] "Annualized Sharpe Ratio:  -0.84"
```



Technical Analysis: SMA Strategy

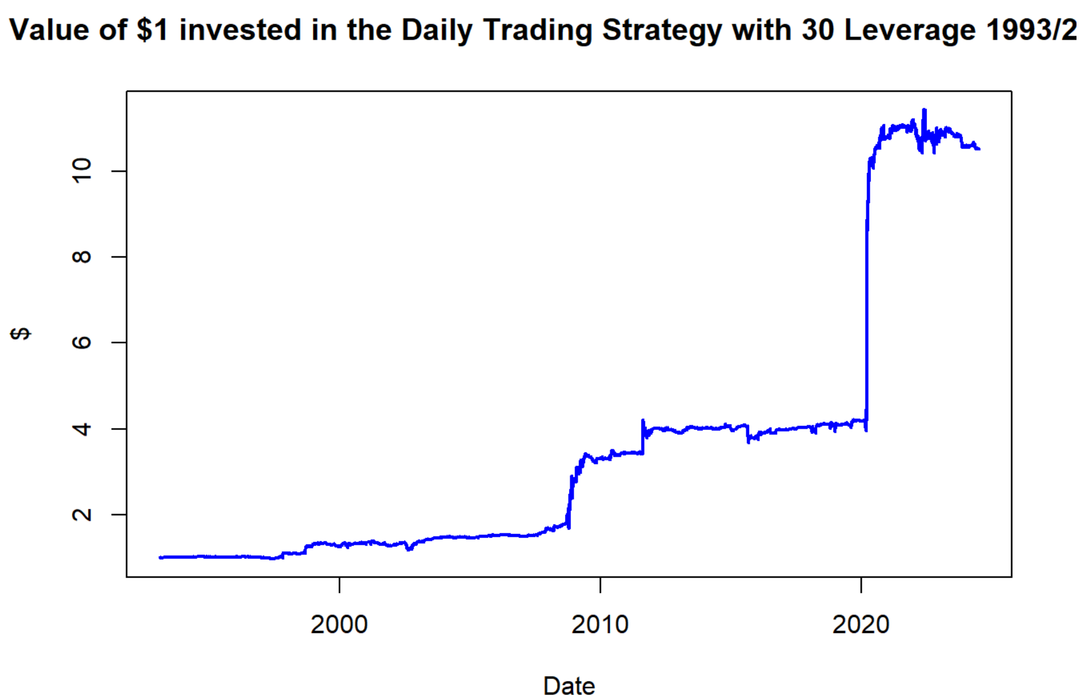
```
# Daily Returns and Risk-Free Rate for Technical Analysis
spy_daily_ret <- dailyReturn(Ad(spy))
colnames(spy_daily_ret) <- "Ret"

# Convert rates to daily frequency
rf_daily <- getSymbols("DFF", src = "FRED", auto.assign = FALSE, from = '1993-01-01', to = '2024-06-30')
rf_daily <- rf_daily/365/100
colnames(rf_daily) <- "Rf"

return_data_daily <- merge(spy_daily_ret, rf_daily, join = 'inner')

# Step 2: Creating the Trading Signal using SMA
spy_sma <- rollapply(spy_daily_ret, width = 1, FUN = mean, by = 1, fill = NA, align = "right")
colnames(spy_sma) <- "SMA1Day"

test_data_technical <- merge(return_data_daily, spy_sma)
test_data_technical$TradingSignal <- test_data_technical$SMA1Day

# Step 3+4: Applying the Strategy Function
test_data_technical$TradingRule <- - test_data_technical$TradingSignal
backtest_sma1 <- market_timing_strategy(data = test_data_technical, TradingRule = test_data_technical$TradingRule,
```

```
[1] "Annualized Excess Return:  0.28 %"
[1] "Annualized Volatility:  0.46 %"
[1] "Annualized Sharpe Ratio:  0.61"
```




```
data_backtest <- market_timing_strategy(data = test_data_technical, TradingRule = test_data_technical$TradingRule,
```

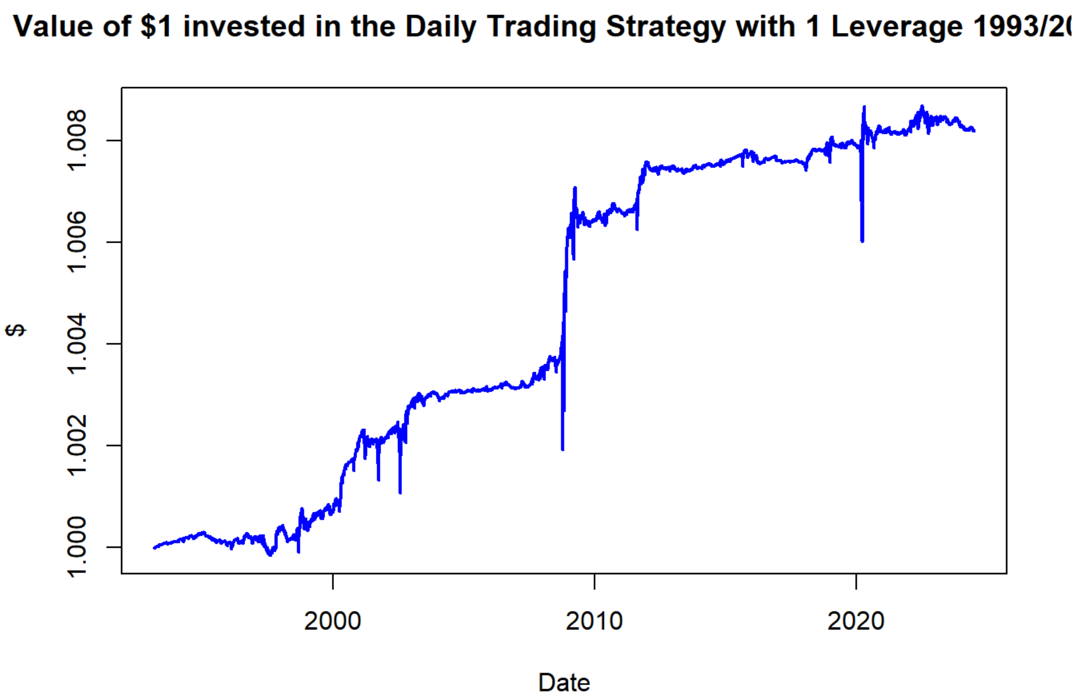
```
[1] "Annualized Excess Return:  8.4 %"
[1] "Annualized Volatility:  13.76 %"
[1] "Annualized Sharpe Ratio:  0.61"
```




```
# 21-Day SMA Strategy
spy_sma_21 <- rollapply(spy_daily_ret, width = 21, FUN = mean, by = 1, fill = NA, align = "right")
```

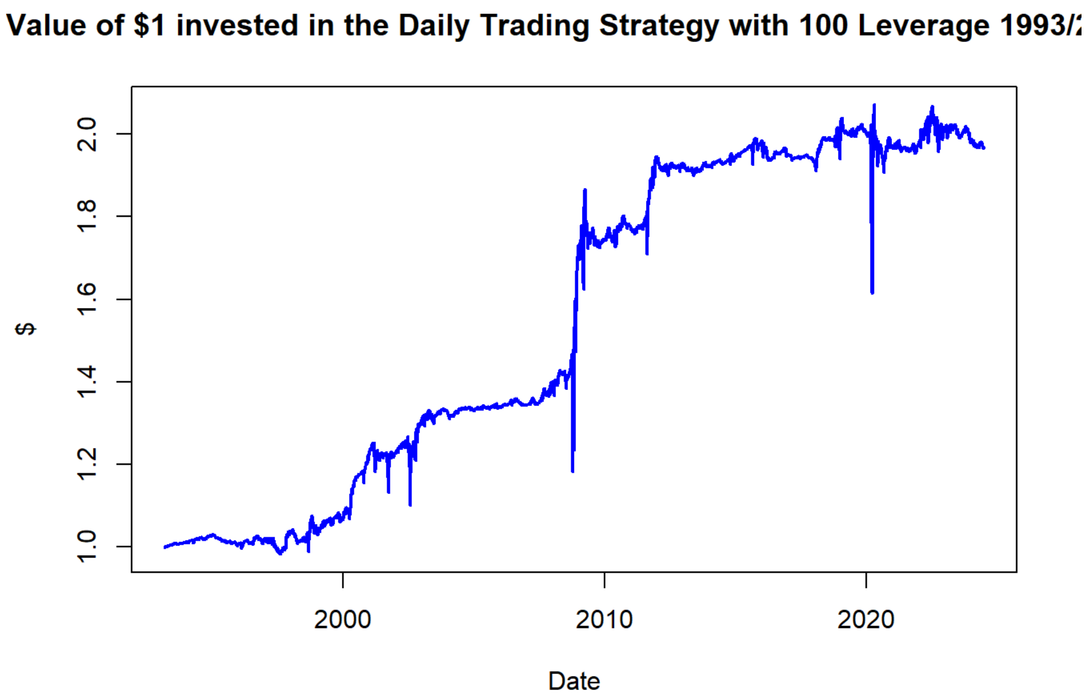
```
colnames(spy_sma_21) <- "SMA21Day"

test_data_technical <- merge(return_data_daily, spy_sma_21)
test_data_technical$TradingSignal <- test_data_technical$SMA21Day

# Applying the Strategy Function
test_data_technical$TradingRule <- - test_data_technical$TradingSignal
backtest_sma21 <- market_timing_strategy(data = test_data_technical, TradingRule = test_data_technical$TradingRule,
```

```
[1] "Annualized Excess Return:  0.03 %"
[1] "Annualized Volatility:  0.1 %"
[1] "Annualized Sharpe Ratio:  0.27"
```




```
data_backtest <- market_timing_strategy(data = test_data_technical, TradingRule = test_data_technical$TradingRule,
```

```
[1] "Annualized Excess Return:  2.61 %"
[1] "Annualized Volatility:  9.7 %"
[1] "Annualized Sharpe Ratio:  0.27"
```



Conclusion

Performance Metrics

The market-timing strategies were evaluated based on key performance metrics:

- **Annualized Return**: The Price-to-Dividend ratio strategy provided moderate returns. The 21-day SMA, showed effectiveness during trending markets but struggled in sideways markets.
- **Annualized Volatility**: The strategies demonstrated different levels of risk. The 1-day SMA, exhibited higher volatility due to frequent trading signals, whereas the Price-to-Dividend ratio strategy had lower volatility.
- **Sharpe Ratio**: The Sharpe ratio indicated that the Price-to-Dividend ratio strategy provided a better risk-adjusted return compared to the SMA strategies.

Comparison

Comparing the fundamental and technical strategies revealed that the Price-to-Dividend ratio strategy provided more consistent returns with lower volatility, while the SMA strategies offered opportunities for higher returns during trending markets.

The market-timing strategies based on the Price-to-Dividend ratio and SMA demonstrate the value of integrating both fundamental and technical analyses. The Price-to-Dividend ratio offers a stable, long-term perspective, while the SMA strategies capitalize on shorter-term market trends.