S&P 500 Sector Performance

Alex Alvarez

title: “ S&P 500 Sector Performance” author: “Alex Alvarez” format: pdf editor: visual

---

Overview

This project analyzes the performance of **S&P 500 sectors** using sector **ETFs** (like XLK for tech, XLE for energy, etc.) from **2015 to July 2025**.
Data is pulled from Yahoo Finance using R packages `quantmod` and `PerformanceAnalytics`.

We'll explore:

- Monthly returns
- Cumulative performance
- Rolling volatility

Tools Used:

- `quantmod`: To download stock data
- `PerformanceAnalytics`: For return metrics & charts
- `tidyverse`: For data wrangling and visualization

Overview

This project analyzes the performance of S&P 500 sectors using sector ETFs (like XLK for tech, XLE for energy, etc.) from 2015 to July 2025. Data is pulled from Yahoo Finance using R packages `quantmod` and `PerformanceAnalytics`.

We'll explore: - Monthly returns - Cumulative performance - Rolling volatility

---

```
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
```

```
library(PerformanceAnalytics)
```

```

Attaching package: 'PerformanceAnalytics'
```

```
The following object is masked from 'package:graphics':

    legend
```

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.2     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.4
```

```
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::first()  masks xts::first()
x dplyr::lag()    masks stats::lag()
x dplyr::last()   masks xts::last()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

```
#create your clean prices object - a time-series dataset with all 11 sector ETF prices from 2

tickers <- c("XLK", "XLE", "XLF", "XLV", "XLY", "XLU", "XLI", "XLB", "XLRE", "XLC", "XLP")

getSymbols(tickers, src = "yahoo", from = "2015-01-01", to = Sys.Date())
```

```
 [1] "XLK"  "XLE"  "XLF"  "XLV"  "XLY"  "XLU"  "XLI"  "XLB"  "XLRE" "XLC"
[11] "XLP"
```

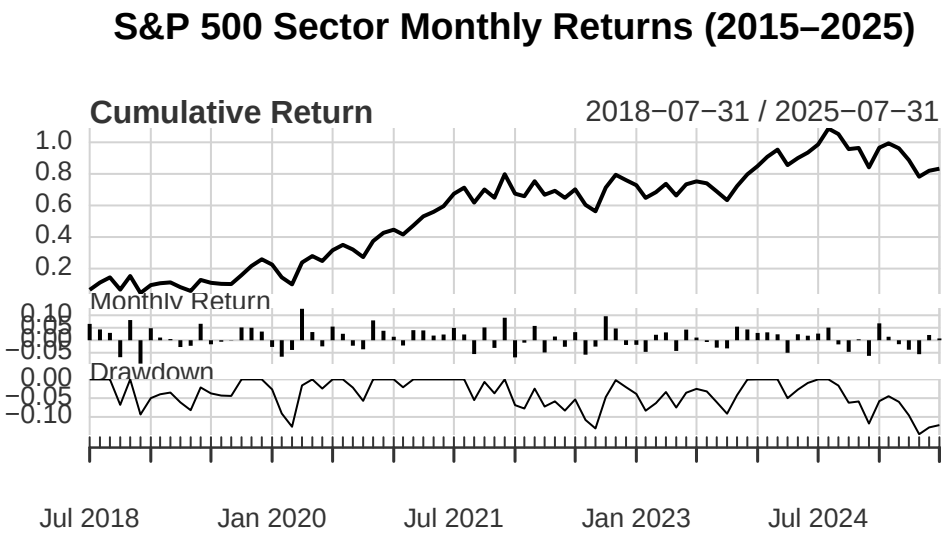
```
prices <- map(tickers, ~ Ad(get(.x))) %>% reduce(merge)
colnames(prices) <- tickers
```

```
# Calculate Mothly Returns
# Convert to monthly prices and extract closing prices only
monthly_prices <- to.monthly(prices, indexAt = "lastof", drop.time = TRUE)[, grep("\\.Close"
```

```
Warning in to.period(x, "months", indexAt = indexAt, name = name, ...): missing
values removed from data
Warning in to.period(x, "months", indexAt = indexAt, name = name, ...): missing
values removed from data
```

```r
# Calculate monthly returns
monthly_returns <- monthly_prices %>%
  ROC(type = "discrete") %>%
  na.omit()

# Plot performance summary
charts.PerformanceSummary(monthly_returns, main = "S&P 500 Sector Monthly Returns (2015-2025)")
```



The chart above compares monthly returns and cumulative performance of 11 major S&P 500 sector ETFs from 2015 to July 2025. On the cumulative returns plot, sectors like Technology (XLK) and Consumer Discretionary (XLY) show the highest long-term gains, while Energy (XLE) lags behind with more volatility. The monthly returns highlight periods of major swings, such as the COVID crash in 2020. In the drawdown chart, we observe that defensive sectors like Utilities (XLU) and Staples (XLP) experience smaller drops during market downturns, reinforcing their low-risk profile.

```r
# Clean Comparison with 5 Sectors (Optional for Simplicity)
tickers <- c("XLK", "XLY", "XLE", "XLP", "XLV")

getSymbols(tickers, src = "yahoo", from = "2015-01-01", to = Sys.Date())
```

```
[1] "XLK" "XLY" "XLE" "XLP" "XLV"
```

```
prices <- map(tickers, ~ Ad(get(.x))) %>% reduce(merge)
colnames(prices) <- tickers
```

```
# Apply to.monthly and extract Close for each ticker individually
monthly_prices_list <- map(tickers, ~ {
  to.monthly(get(.x), indexAt = "lastof", drop.time = TRUE)[, 4]  # 4 = Close
})

# Combine into one xts object
monthly_prices <- reduce(monthly_prices_list, merge)
colnames(monthly_prices) <- tickers

# Calculate monthly returns
monthly_returns <- ROC(monthly_prices, type = "discrete") %>% na.omit()
```

```
colnames(table.Stats(monthly_returns))
```

```
[1] "XLK" "XLY" "XLE" "XLP" "XLV"
```

```
summary_stats <- table.Stats(monthly_returns)
round(summary_stats, 4)
```

```
                     XLK      XLY      XLE      XLP      XLV
Observations    126.0000 126.0000 126.0000 126.0000 126.0000
NAs               0.0000   0.0000   0.0000   0.0000   0.0000
Minimum          -0.1218  -0.1532  -0.3581  -0.0974  -0.0976
Quartile 1       -0.0155  -0.0174  -0.0341  -0.0147  -0.0250
Median            0.0184   0.0110   0.0095   0.0069   0.0108
Arithmetic Mean   0.0166   0.0109   0.0048   0.0049   0.0062
Geometric Mean    0.0150   0.0092   0.0012   0.0042   0.0054
Quartile 3        0.0561   0.0361   0.0325   0.0241   0.0328
Maximum           0.1374   0.1888   0.3076   0.0970   0.1259
SE Mean           0.0050   0.0052   0.0076   0.0033   0.0037
LCL Mean (0.95)   0.0067   0.0006  -0.0103  -0.0016  -0.0011
UCL Mean (0.95)   0.0264   0.0212   0.0200   0.0113   0.0136
Variance          0.0031   0.0034   0.0073   0.0013   0.0017
Stdev             0.0557   0.0583   0.0857   0.0365   0.0416
Skewness         -0.2230   0.1807   0.1372  -0.1866  -0.0636
Kurtosis         -0.3741   0.9550   3.7257   0.2102  -0.2309
```

```r
suppressWarnings(
  charts.PerformanceSummary(
    monthly_returns,
    main = "S&P 500 Sector Monthly Returns (2015-2025)"
  )
)
```