Story 2 — Trifecta Indicators

Cai Lin

2025-09-28

==

Assignment: Investigating the Trifecta of Economic Indicators

Base R only: fetch from FRED API, prep monthly data, ONE plot, write PDF

Output: trifecta_econ_indicators.pdf in your working directory

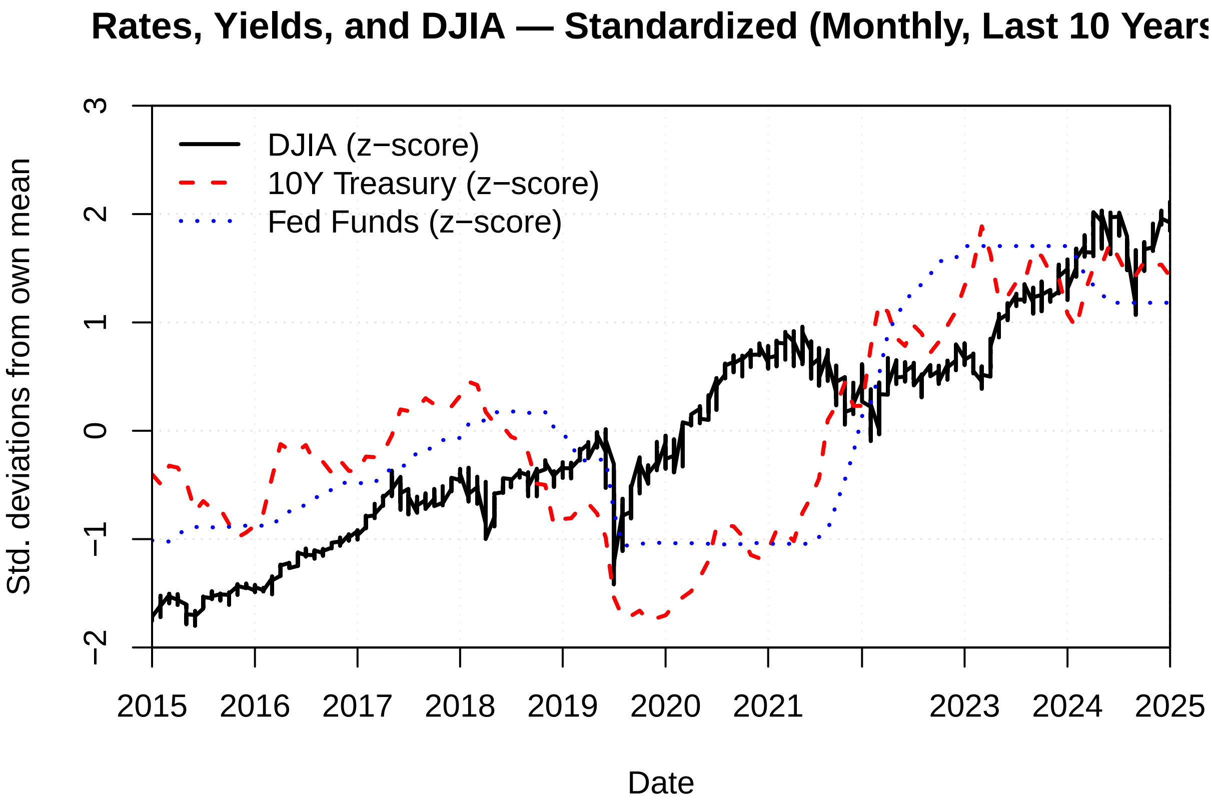
==

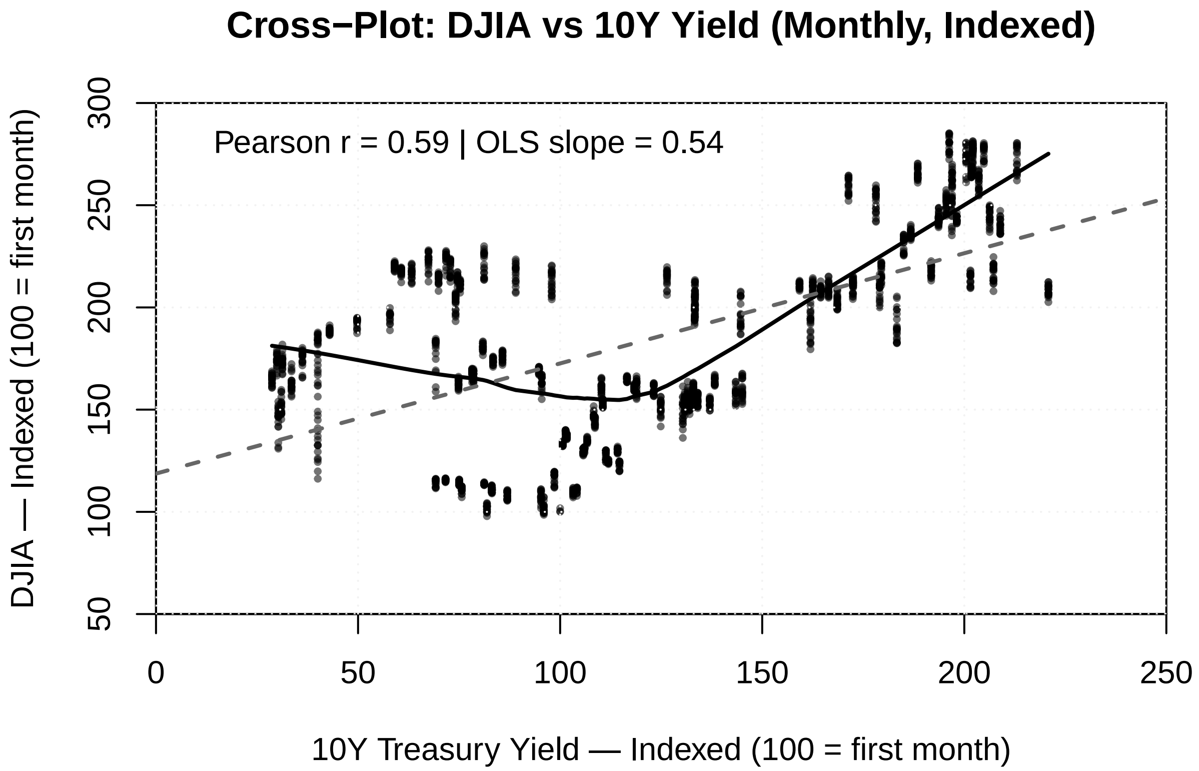
```
knitr::opts_chunk$set(echo = FALSE, message = FALSE, warning = FALSE)

# -------------- API KEY ---------------
FRED_API_KEY <- Sys.getenv("FRED_API_KEY_ENV")
# --------------------------------------

start_date <- "2000-01-01"
end_date   <- format(Sys.Date(), "%Y-%m-%d")

# ---- Helper: fredgraph monthly CSV (no API key) ----
fredgraph_monthly <- function(series_id, start_date="2000-01-01", end_date=format(Sys.Date(), "%Y-%m-%d
                              aggregation = c("avg","eop")) {
  aggregation <- match.arg(aggregation)
  url <- paste0(
    "https://fred.stlouisfed.org/graph/fredgraph.csv",
    "?id=", series_id,
    "&cosd=", start_date,
    "&coed=", end_date,
    "&frequency=m",
    "&aggregation_method=", aggregation
  )
  dest <- tempfile(fileext = ".csv")
  old <- getOption("download.file.method"); on.exit(options(download.file.method = old), add = TRUE)
  options(download.file.method = "wininet")
  try(utils::download.file(url, destfile = dest, mode = "wb", quiet = TRUE), silent = TRUE)
  if (!file.exists(dest) || is.na(file.info(dest)$size) || file.info(dest)$size < 20) {
    options(download.file.method = "libcurl")
    utils::download.file(url, destfile = dest, mode = "wb", quiet = TRUE)
  }
  z <- utils::read.csv(dest, stringsAsFactors = FALSE)
  names(z)[1:2] <- c("DATE","VALUE")
  z$DATE <- as.Date(z$DATE)
  suppressWarnings(z$VALUE <- as.numeric(z$VALUE))
  z <- z[!is.na(z$VALUE), c("DATE","VALUE")]
  data.frame(
    ym         = format(z$DATE, "%Y-%m"),
    month_date = as.Date(paste0(format(z$DATE, "%Y-%m"), "-01")),
    value      = z$VALUE
  )
}

# ---- Helper: FRED API (download → auto-decompress → read CSV) ----
fred_api_csv <- function(series_id, api_key, start_date, end_date) {
  base <- "https://api.stlouisfed.org/fred/series/observations"
  url  <- paste0(
    base,
    "?series_id=", series_id,
    "&api_key=", api_key,
    "&file_type=csv",
    "&observation_start=", start_date,
    "&observation_end=", end_date
  )

  dest <- tempfile(fileext = ".csv")
  old <- getOption("download.file.method"); on.exit(options(download.file.method = old), add = TRUE)
  options(download.file.method = "wininet")
  try(utils::download.file(url, destfile = dest, mode = "wb", quiet = TRUE), silent = TRUE)
  if (!file.exists(dest) || is.na(file.info(dest)$size) || file.info(dest)$size < 20) {
    options(download.file.method = "libcurl")
    utils::download.file(url, destfile = dest, mode = "wb", quiet = TRUE)
  }

  # Read raw; detect gzip (1F 8B) or zlib/deflate (78 9C / 78 DA)
  sz  <- file.info(dest)$size
  raw <- readBin(dest, what = "raw", n = sz)
  decomp <- NULL
  if (length(raw) >= 2 && raw[1] == as.raw(0x1F) && raw[2] == as.raw(0x8B)) {
    decomp <- memDecompress(raw, type = "gzip")
  } else if (length(raw) >= 2 && raw[1] == as.raw(0x78) && (raw[2] == as.raw(0x9C) || raw[2] == as.raw(
    decomp <- memDecompress(raw, type = "zlib")
  }

  if (!is.null(decomp)) {
    txt <- rawToChar(decomp); Encoding(txt) <- "UTF-8"
    x <- utils::read.csv(textConnection(txt), stringsAsFactors = FALSE)
  } else {
    x <- try(utils::read.csv(dest, stringsAsFactors = FALSE), silent = TRUE)
    if (inherits(x, "try-error")) x <- data.frame()
  }

  # If weird columns, try streaming via gzcon(url())
  if (!all(c("date","value") %in% names(x))) {
```

```r
    con <- NULL
    x2 <- try({
      con <- gzcon(url(url, open = "rb"))
      on.exit(try(close(con), silent = TRUE), add = TRUE)
      txt <- readLines(con, warn = FALSE)
      utils::read.csv(textConnection(txt), stringsAsFactors = FALSE)
    }, silent = TRUE)
    if (!inherits(x2, "try-error") && all(c("date","value") %in% names(x2))) x <- x2
  }

  # Final fallback (still base R): use monthly fredgraph for this series
  if (!all(c("date","value") %in% names(x))) {
    mg <- fredgraph_monthly(series_id, start_date, end_date, aggregation = "avg")
    x <- data.frame(date = mg$month_date, value = mg$value, stringsAsFactors = FALSE)
  }

  x$date <- as.Date(x$date)
  suppressWarnings(x$value <- as.numeric(x$value))
  x <- x[!is.na(x$value), c("date","value")]
  if (nrow(x) == 0) stop("No rows after parsing for ", series_id, ".")
  x
}

# ---- Daily → Monthly mean ----
toMonthlyMean <- function(df) {
  ym <- format(df$date, "%Y-%m")
  m  <- aggregate(df$value, by = list(ym = ym), FUN = function(v) mean(v, na.rm = TRUE))
  names(m)[2] <- "value"
  m$month_date <- as.Date(paste0(m$ym, "-01"))
  m[order(m$month_date), ]
}

indexer <- function(x) 100 * x / x[1]



# Quick ping to confirm the key + parser work
head(fred_api_csv("FEDFUNDS", FRED_API_KEY, "2020-01-01", "2020-01-31"))
```

```
## Warning in utils::download.file(url, destfile = dest, mode = "wb", quiet =
## TRUE): the 'wininet' method is deprecated for http:// and https:// URLs

## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
## line 1 appears to contain embedded nulls

## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
## line 2 appears to contain embedded nulls

## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
## line 4 appears to contain embedded nulls

## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
## incomplete final line found by readTableHeader on
## 'C:\Users\Kai\AppData\Local\Temp\RtmpakhsFi\file1f92849cb6762.csv'

## Warning in utils::download.file(url, destfile = dest, mode = "wb", quiet =
## TRUE): the 'wininet' method is deprecated for http:// and https:// URLs
```

```
##         date value
## 1 2020-01-01  1.55
```





### Interpretation

Over roughly 25 years, the indicators generally move in related cycles. Periods of rising policy rates tend to align with higher 10-year yields, and equity performance often slows or becomes choppier during tightening phases; conversely, easing cycles and falling long yields frequently precede or accompany rebounds in equities. The cross-plot shows a modest positive association between long rates and equity levels across the sample, though the relationship is not one-for-one and depends on growth and inflation expectations. Overall, monetary policy and the term structure provide useful context for interpreting market pricing, but other fundamentals also shape equity outcomes.