lisa

Michael Hernandez

2025-10-17

```
library(readxl)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.4.3
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
library(scales)
```

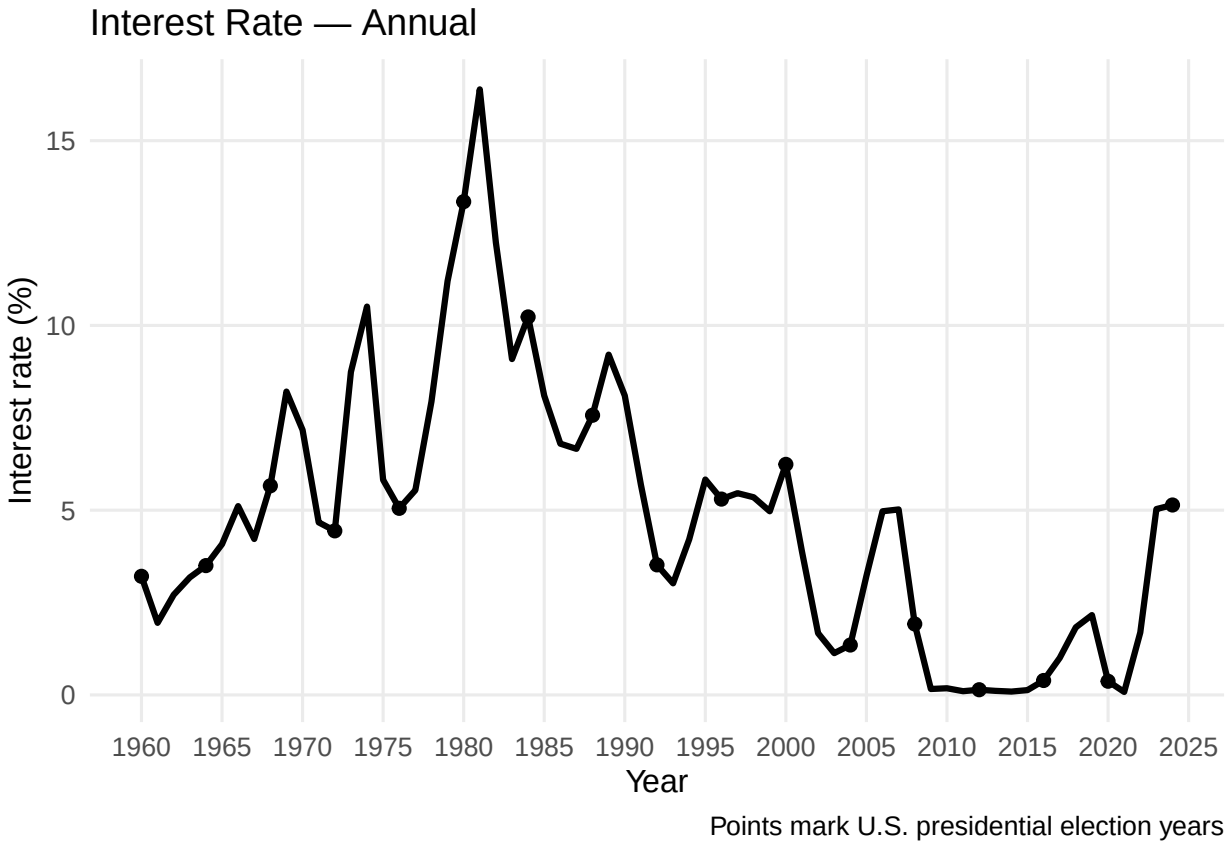
```
## Warning: package 'scales' was built under R version 4.4.3
```

```
# ==== Minimal, robust: read -> clean -> plot ====

xlsx_path <- "InterestRates__InflationData_andGraphs 1960-2024.xlsx"
```

```r
# --- 1) Read file. If R created placeholder names like "...1", make row 1 the header.
raw_try <- read_excel(xlsx_path, col_names = TRUE)
if (all(grepl("^\\.\\.\\.", names(raw_try)))) {
  raw0 <- read_excel(xlsx_path, col_names = FALSE)
  hdr  <- as.character(unlist(raw0[1, ]))
  hdr[is.na(hdr) | hdr == ""] <- paste0("col", which(is.na(hdr) | hdr == ""))
  names(raw0) <- hdr
  dat <- raw0[-1, , drop = FALSE]
} else {
  dat <- raw_try
}

# --- 2) Use FIRST THREE COLUMNS only (Year, Interest, Inflation).
stopifnot(ncol(dat) >= 2)
cn <- names(dat)
col1 <- cn[1]                       # Year-like
col2 <- cn[2]                       # Interest
col3 <- if (ncol(dat) >= 3) cn[3] else NULL  # Inflation optional

# --- 3) Helpers
to_year <- function(x){
  if (inherits(x, "Date")) return(lubridate::year(x))
  if (is.numeric(x)) {
    # already year? keep; else Excel serial
    out <- ifelse(x >= 1800 & x <= 2100, x, lubridate::year(as.Date(x, origin = "1899-12-30")))
    return(as.integer(out))
  }
  x <- as.character(x)
  y <- suppressWarnings(lubridate::year(lubridate::mdy(x)))
  y[is.na(y)] <- suppressWarnings(lubridate::year(lubridate::ymd(x[is.na(y)])))
  as.integer(y)
}

to_num <- function(x){
  if (is.numeric(x)) return(as.numeric(x))
  x <- as.character(x)
  x <- trimws(x)
  x <- gsub(",", "", x, fixed = TRUE)     # remove thousands commas
  x <- gsub("%", "", x, fixed = TRUE)     # remove percent sign
  x <- gsub("[^0-9eE+\\-\\.]", "", x)     # keep digits, dot, minus, exponent
  suppressWarnings(as.numeric(x))
}

# --- 4) Build tidy data. If monthly rows exist, average to yearly.
year_raw <- dat[[col1]]
ir_raw   <- dat[[col2]]
infl_raw <- if (!is.null(col3)) dat[[col3]] else rep(NA_character_, nrow(dat))

df <- tibble::tibble(
  year          = to_year(year_raw),
  interest_rate = to_num(ir_raw),
  inflation     = to_num(infl_raw)
) %>%
```

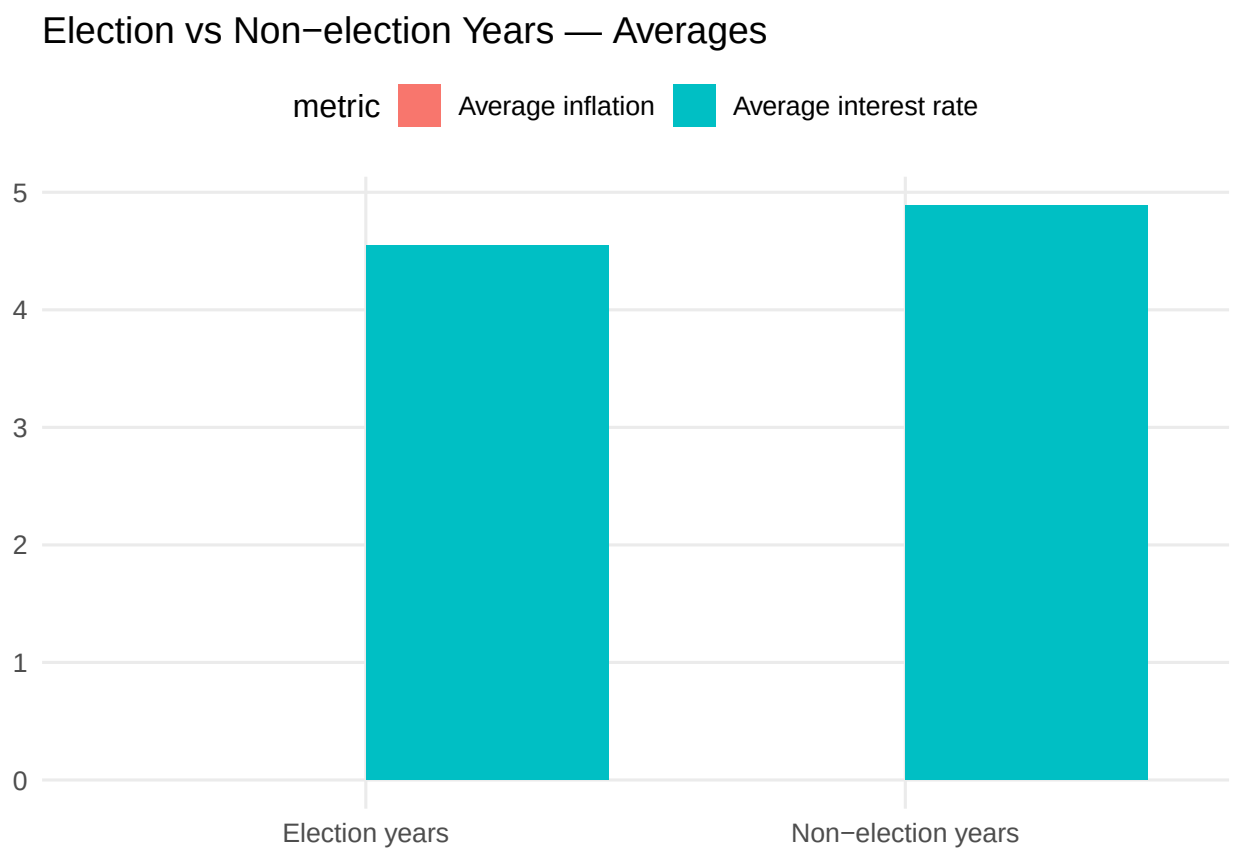
```r
  filter(!is.na(year)) %>%
  group_by(year) %>%
  summarise(
    interest_rate = if (all(is.na(interest_rate))) NA_real_ else mean(interest_rate, na.rm = TRUE),
    inflation     = if (all(is.na(inflation)))     NA_real_ else mean(inflation,     na.rm = TRUE),
    .groups = "drop"
  ) %>%
  arrange(year)

has_infl <- any(!is.na(df$inflation))

# --- 5) Mark election vs non-election
election_years <- seq(1960, 2024, by = 4)
df <- df %>%
  mutate(
    election_year = year %in% election_years,
    period = if_else(election_year, "Election years", "Non-election years")
  )



# --- 7) Plots
if (has_infl) {
  # Dual-axis with election points
  sf <- max(df$interest_rate, na.rm = TRUE) / max(df$inflation, na.rm = TRUE)
  p_all <- ggplot(df, aes(x = year)) +
    geom_line(aes(y = interest_rate, color = "Interest rate"), linewidth = 1.1) +
    geom_line(aes(y = inflation * sf,  color = "Inflation"), linewidth = 1.1, linetype = 2) +
    geom_point(data = subset(df, election_year),
               aes(y = interest_rate, color = "Interest rate"), size = 2) +
    scale_y_continuous(
      name = "Interest rate (%)",
      sec.axis = sec_axis(~ . / sf, name = "Inflation (%)")
    ) +
    scale_x_continuous(breaks = pretty(df$year, n = 16)) +
    scale_color_manual(NULL, values = c("Interest rate" = "#2C7FB8", "Inflation" = "#D95F0E")) +
    labs(title = "Interest Rate vs Inflation - Annual",
         caption = "Points mark U.S. presidential election years", x = "Year") +
    theme_minimal(base_size = 12) +
    theme(legend.position = "top", panel.grid.minor = element_blank())
} else {
  # Only interest rate available
  p_all <- ggplot(df, aes(x = year, y = interest_rate)) +
    geom_line(linewidth = 1.1) +
    geom_point(data = subset(df, election_year), size = 2) +
    scale_x_continuous(breaks = pretty(df$year, n = 16)) +
    labs(title = "Interest Rate - Annual",
         y = "Interest rate (%)", x = "Year",
         caption = "Points mark U.S. presidential election years") +
    theme_minimal(base_size = 12) +
    theme(panel.grid.minor = element_blank())
}
```

```r
# Story chart: average levels in election vs non-election years
story <- df %>%
  group_by(period) %>%
  summarise(
    `Average interest rate` = mean(interest_rate, na.rm = TRUE),
    `Average inflation`     = if (has_infl) mean(inflation, na.rm = TRUE) else NA_real_,
    .groups = "drop"
  ) %>%
  tidyr::pivot_longer(-period, names_to = "metric", values_to = "value")

p_story <- ggplot(story, aes(period, value, fill = metric)) +
  geom_col(position = "dodge") +
  labs(title = "Election vs Non-election Years - Averages", x = NULL, y = NULL) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top", panel.grid.minor = element_blank())

print(p_all)
```




```r
print(p_story)
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_col()`).
```

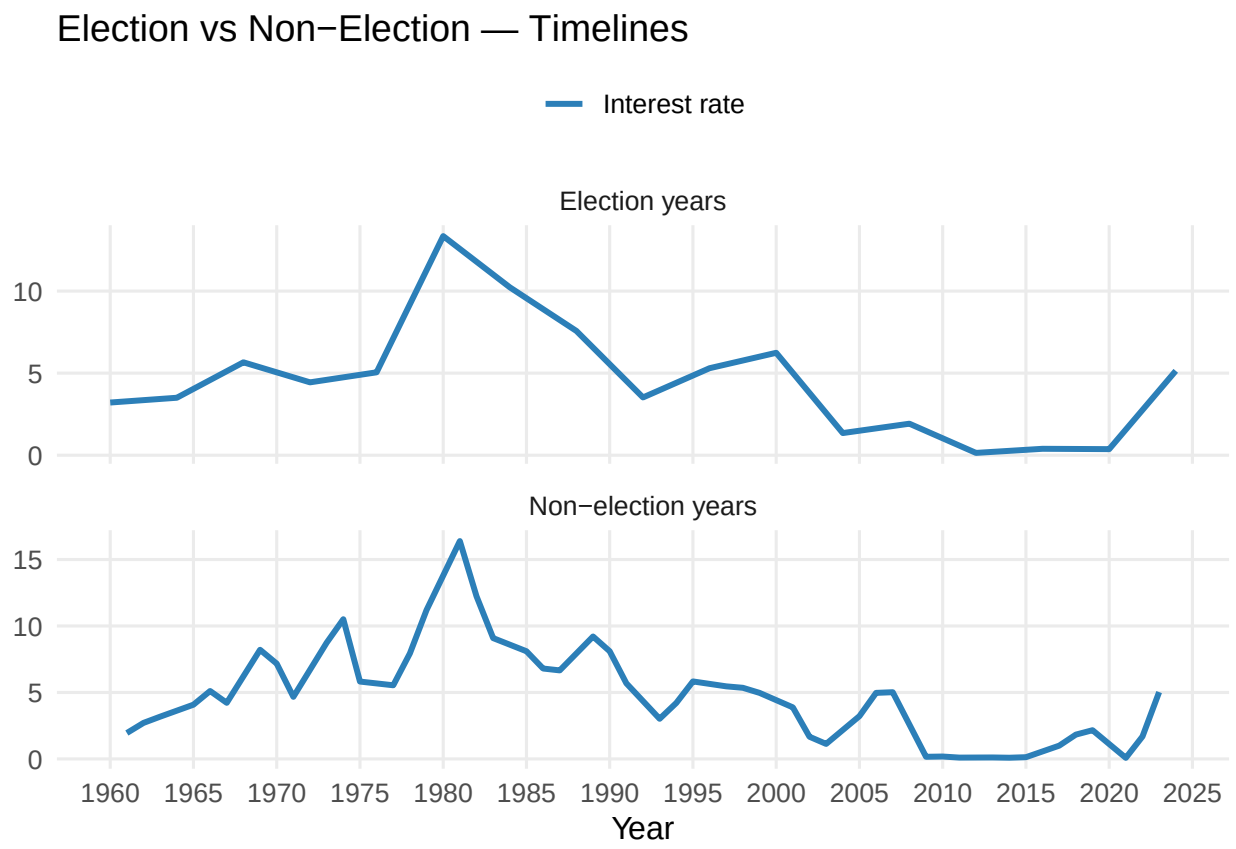



```
# Save PNGs
ggsave("Plot_AllYears.png",  p_all,   width = 12, height = 7, dpi = 300)
ggsave("Plot_Story.png",     p_story, width = 9,  height = 6, dpi = 300)
```

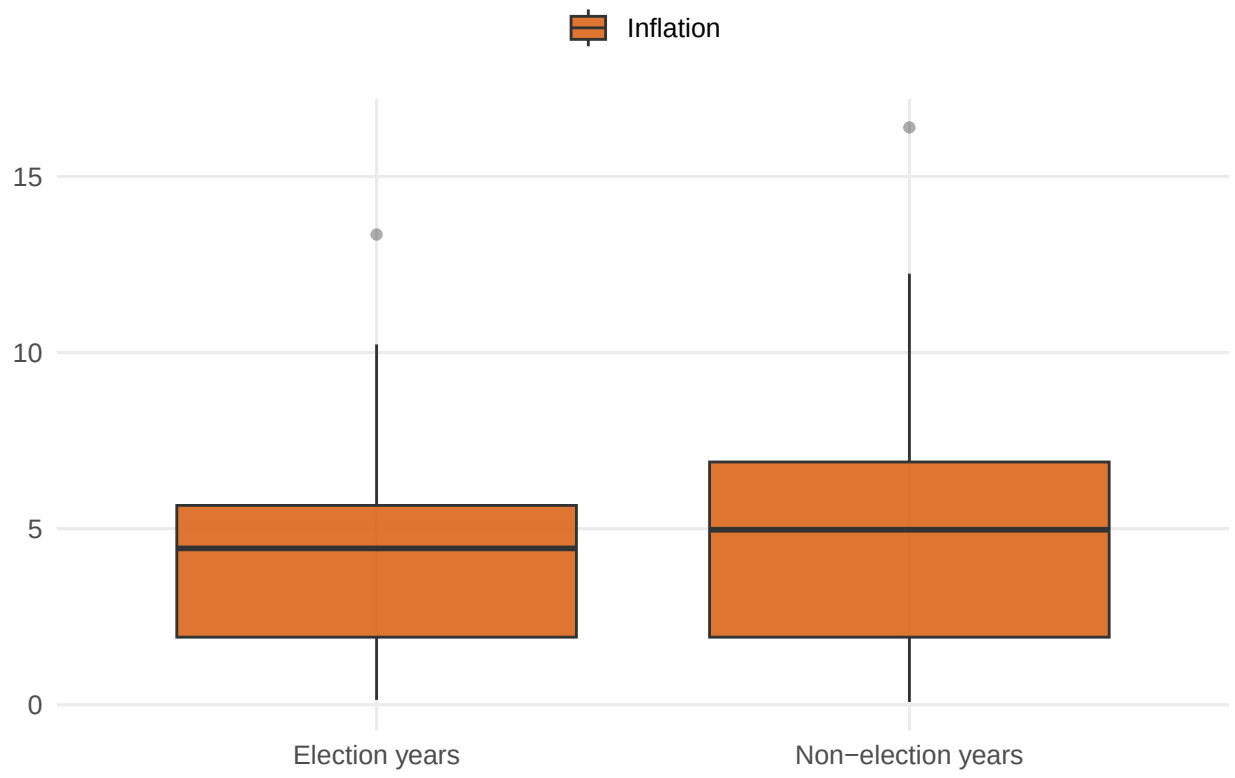
```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_col()`).
```

```
# ========= Election vs Non-Election: comparisons =========
library(ggplot2)
library(dplyr)
library(scales)

# 1) Side-by-side timeline (facet) to compare patterns
p_facet <- ggplot(df, aes(x = year)) +
  geom_line(aes(y = interest_rate, color = "Interest rate"), linewidth = 1.05) +
  { if (any(!is.na(df$inflation)))
      geom_line(aes(y = inflation, color = "Inflation"), linewidth = 1.05, linetype = 2)
    else NULL } +
  facet_wrap(~ period, ncol = 1, scales = "free_y") +
  scale_color_manual(NULL, values = c("Interest rate" = "#2C7FB8", "Inflation" = "#D95F0E")) +
  scale_x_continuous(breaks = pretty(df$year, n = 12)) +
  labs(title = "Election vs Non-Election - Timelines",
       x = "Year", y = NULL) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top", panel.grid.minor = element_blank())
```

```r
# 2) Distribution comparison (boxplots) - interest & inflation by period
long_vals <- tidyr::pivot_longer(
  df, cols = c(interest_rate, inflation),
  names_to = "metric", values_to = "value"
) %>% filter(!is.na(value))

p_boxes <- ggplot(long_vals, aes(x = period, y = value, fill = metric)) +
  geom_boxplot(alpha = 0.85, outlier.alpha = 0.4) +
  scale_fill_manual(NULL, labels = c("Inflation", "Interest rate"),
                    values = c("#D95F0E", "#2C7FB8")) +
  labs(title = "Election vs Non-Election - Distributions",
       x = NULL, y = NULL) +
  theme_minimal(base_size = 12) +
  theme(legend.position = "top", panel.grid.minor = element_blank())

# 3) Scatter with trend lines + correlations (if inflation exists)
if (any(!is.na(df$inflation))) {
  # correlations per period (shown in subtitle)
  cors <- df %>%
    group_by(period) %>%
    summarise(r = cor(interest_rate, inflation, use = "complete.obs")) %>%
    mutate(lab = paste0(period, ": r=", round(r, 2))) %>%
    pull(lab) %>%
    paste(collapse = "   |   ")

  p_scatter <- ggplot(df, aes(x = inflation, y = interest_rate, color = period)) +
    geom_point(alpha = 0.85, size = 2) +
    geom_smooth(method = "lm", se = FALSE, linewidth = 1.05) +
    scale_color_manual(NULL, values = c("Election years" = "#6A3D9A",
                                        "Non-election years" = "#1B9E77")) +
    labs(title = "Interest vs Inflation - Election vs Non-Election",
         subtitle = cors,
         x = "Inflation (%)", y = "Interest rate (%)") +
    theme_minimal(base_size = 12) +
    theme(legend.position = "top", panel.grid.minor = element_blank())
}

# Show
print(p_facet)
```




```
print(p_boxes)
```




```
if (exists("p_scatter")) print(p_scatter)

# Save
ggsave("EVNE_Timelines.png", p_facet,  width = 11, height = 7, dpi = 300)
ggsave("EVNE_Distributions.png", p_boxes, width = 9,  height = 6, dpi = 300)
if (exists("p_scatter")) ggsave("EVNE_Scatter.png", p_scatter, width = 9, height = 6, dpi = 300)
```