

OSHA Budget vs Inflation

2024-10-19

Setup

Libraries

```
# Loading Excel data
```

```
library(readxl)
```

```
# Data Cleaning
```

```
library(tidyverse) # For data cleaning
```

```
## — Attaching core tidyverse packages — tidyverse 2.0.0 —
```

```
## ✓ dplyr      1.1.4      ✓ readr      2.1.5
```

```
## ✓ forcats   1.0.0      ✓ stringr    1.5.1
```

```
## ✓ ggplot2   3.5.1      ✓ tibble     3.2.1
```

```
## ✓ lubridate 1.9.3      ✓ tidyr      1.3.1
```

```
## ✓ purrr     1.0.2
```

```
## — Conflicts — tidyverse_conflicts() —
```

```
## ✗ dplyr::filter() masks stats::filter()
```

```
## ✗ dplyr::lag()     masks stats::lag()
```

```
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be  
come errors
```

```
library(lubridate) # Working with dates
```

```
# FRED API
```

```
library(fredr) # API package for FRED
```

```
# Custom FRED functions
```

```
source("FRED Custom Functions.R")
```

```
# Visualization
```

```
library(ggthemes)
```

Cleaning

OSHA Budget Data

Setup

```
# This was manually-extracted from the DoL operating budget reports
# https://www.dol.gov/general/aboutdol#budget
original <- read_xlsx('OSHA OB Data.xlsx')
```

Cleaning

```
# Some of the early operating budgets contained overlapping years
# Will resolve through seniority--the most recent report is presumed to be the most accurate
govt <- original

authoritative <- data.frame() %>% as_tibble()

for (year in unique(govt$Year)){

  year_rows <- filter(govt, Year == year) %>%
    mutate(
      Source = gsub('FY', '', Source),
      Source = as.numeric(Source)
    ) %>%
    filter(Source == max(.$Source))

  authoritative <- bind_rows(authoritative, year_rows)

  # Rows are sorted such that
}

# Cleanup
govt <- authoritative
rm(authoritative, year_rows, year)
```

```
# Rename columns to be more friendly
govt <- govt %>%
  transmute(
    year = Year,
    budget = `Budget (Total)`
  )
```

```
# No need to have such large units: convert to millions of USD.
govt <- govt %>%
  mutate(
    budget = budget/1e6
  )
```

```
# Necessary for future merging
govt <- govt %>%
  mutate(date = as.Date(paste(year, "01", "01", sep = "-")))
```

```
# This is for later. We will want to pull FRED data as early as the year before  
# the earliest year we have budget data for, as taking the derivative requires  
# an additional year
```

```
govt_min_year <- min(govt$date)
```

```
govt <- govt %>%  
  select(year, date, everything())
```

```
# Remove all objects that don't need to be in the environment in future scripts  
rm(original)
```

FRED Inflation Data

Setup

```
# Create a FRED key here: https://fred.stlouisfed.org/docs/api/api_key.html  
# FRED key  
fredr_set_key("Insert key here!")
```

```
# Interested in a variety of inflation rates

# https://fred.stlouisfed.org/series/CPIAUCSL
# https://fred.stlouisfed.org/series/CORESTICKM159SFRBATL
# https://fred.stlouisfed.org/series/PCEPI
# https://fred.stlouisfed.org/series/PCEPILFE
# https://fred.stlouisfed.org/series/CWUR0000SA0
# https://fred.stlouisfed.org/series/A191RI1Q225SBEA
# https://fred.stlouisfed.org/series/USAGDPDEFQISMEI
# https://fred.stlouisfed.org/series/FPCPITOTLZGUSA
# https://fred.stlouisfed.org/series/MEDCPIM158SFRBCLE
# https://fred.stlouisfed.org/series/PCETRM12M159SFRBDAL
# https://fred.stlouisfed.org/series/CRESTKCPIXSLTRM159SFRBATL
# https://fred.stlouisfed.org/series/CUUR0000SASL2RS

fred_rates <- c(
  "PCEPI",
  "PCEPILFE",
  "CPIAUCSL",
  "CWUR0000SA0",
  "USAGDPDEFQISMEI",
  "CORESTICKM159SFRBATL",
  "A191RI1Q225SBEA",
  "FPCPITOTLZGUSA",
  "MEDCPIM158SFRBCLE",
  "PCETRM12M159SFRBDAL",
  "CRESTKCPIXSLTRM159SFRBATL",
  "CUUR0000SASL2RS"
)

fred_startdate <- govt_min_year - years(1) # Subtract by one, in order to take the derivative
later
```

Cleaning

```
for (i in 1:length(fred_rates)){

  ### Load Data ###
  new <- fredr(
    series_id = fred_rates[i],
    observation_start = fred_startdate
  )

  new_series_id <- unique(new$series_id)

  ### Process Data ###

  # Check if yearly
  if (check_yearly(new) == FALSE){

    # Convert to yearly
    new <- date_to_year(new)

  }

  # Check if derivative
  if (check_derivative(new) == TRUE){

    new$value <- new$value*1/100 # WARNING: Assumes that all rates are like 2.00, and not .02
    new$value <- rate_to_indexed(new)

  }

  # Rename value column
  colnames(new)[colnames(new) == "value"] <- new_series_id

  ### Compile ###

  # Check if this is the first iteration
  if (i == 1){

    rates <- new

  }

  if (i != 1){

    new <- new[, c("date",new_series_id)]
    rates <- left_join(rates,new, by = "date")

  }

}
```

```
rates <- rates %>%  
  select(year, date, everything())
```

```
# For uniformity  
rates <- rates %>%  
  mutate(year = as.numeric(year))
```

```
rm(new,i,new_series_id)
```

Final Cleaning

Functions

```
# Function so that parameters can be easily changed  
  
inflate_multiple_series <- function(series_data,inflator,result, initial_date,fred_rates){  
  
  initial_value <- series_data$budget[series_data$date == initial_date]  
  
  # Calculate  
  for (i in fred_rates){  
  
    # Designate which column will be modified this iteration  
    colnames(inflator)[colnames(inflator) == i] <- "current_iteration"  
  
    result$current_iteration <- inflate(  
      initial_value = initial_value,  
      inflator = inflator$current_iteration/inflator$current_iteration[inflator$date == initial_date]  
    )  
  
    # Restore the column's previous name  
    colnames(result)[colnames(result) == "current_iteration"] <- i  
    colnames(inflator)[colnames(inflator) == "current_iteration"] <- i  
  }  
  
  return(result)  
  
}
```

```
calculate_series_mean <- function(series){  
  
  result <- series %>%  
    mutate(mean = rowMeans(select(., -c(year,date,budget)), na.rm = TRUE))  
  
  return(result)  
}
```

```
reformat_panel <- function(series){  
  
  # Useful for visualizations  
  panel <- series %>%  
    pivot_longer(  
      .,  
      cols = names(select(., -c(year,date))),  
      names_to = "series",  
      values_to = "value"  
    )  
  
  panel_rates <- panel %>%  
    filter(series != "budget" & series != "mean")  
  
  return_list <- list("panel" = panel, "panel_rates" = panel_rates)  
  
  return(return_list)  
}
```

```
# Now one big function, which does each of these steps  
apply_functions <- function(series){  
  # Input is the output of inflate_multiple_series!  
  
  result <- calculate_series_mean(series)  
  
  result <- reformat_panel(result)  
  
  return(result)  
}
```

Data Merges

```
merge_budget_rates <- function(rates,budget){  
  
  result <- left_join(  
    rates,  
    budget,  
    by = "date"  
  ) %>%  
    select(year,date,budget,everything())  
  
  return(result)  
  
}  
  
merged <- merge_budget_rates(  
  rates = rates,  
  budget = govt %>% select(-c("year"))  
)
```

Visualization


```
# Make Data

initial_date <- as.Date("2010-01-01")

result <- apply_functions(
  inflate_multiple_series(
    series_data = govt,
    inflator = rates,
    result = merged %>% select(year,date,budget),
    initial_date = initial_date,
    fred_rates = fred_rates
  )
)

panel <- result[[1]]
panel_rates <- result[[2]]

# Parameters
parameter_linewidth = 1
parameter_graph_startyear = as.Date("2008-01-01")

osha_budget_plot <- ggplot(data = panel, mapping = aes(x = date))+

### LINES

#### OSHA BUDGET
geom_line(
  data = filter(panel, series == "budget" & date >= parameter_graph_startyear),
  mapping = aes(y = value, color = "Actual Budget"),
  linewidth = parameter_linewidth
)+
geom_point(
  data = filter(panel, series == "budget" & date >= parameter_graph_startyear),
  mapping = aes(y = value),
  color = "darkblue"
)+

#### INFLATION RATES
geom_line(data = filter(panel_rates, date >= initial_date),
  mapping = aes(y = value, group = series,
    color = "2010 Inflation-Adjusted Budget Estimates"),
  alpha = .1,
  linewidth = parameter_linewidth)+

### LABELS
labs(
  title = "OSHA's Budget and Inflation",
  subtitle = "Since the 2010 budget freeze, OSHA's budget has not kept up with inflation",
  x = "Year",
  y = "US Dollars (Millions)",
  caption = "Inflation data source: FRED ||| OSHA budget data source: Department of Labor B
```

udget Reports (2009-2024)

Note: Multiple inflation rates were used, to give a range of estimates.

Note: Only discretionary spending was counted, not mandatory spending. This lowers the FY2020 budget.

Note: Whether the budget appears to have kept up with inflation depends on the chosen year. 2010 was

chosen because it was the start of the decade-long budget freeze, in which inflation shrank the budget.

OSHA's first FY, 1974, was not chosen, because inflation adjustments are less accurate over time."

)+

LEGEND

```
scale_color_manual(name="", # No need for Legend name
                    breaks=c("2010 Inflation-Adjusted Budget Estimates", "Actual Budget"),
                    values=c("2010 Inflation-Adjusted Budget Estimates"='black', "Actual Budget"='steelblue'))+
```

AXES

```
scale_x_date(
  breaks = scales::pretty_breaks(n = 10),
  date_labels = paste0("FY ", "%y")
)+
```

THEMES

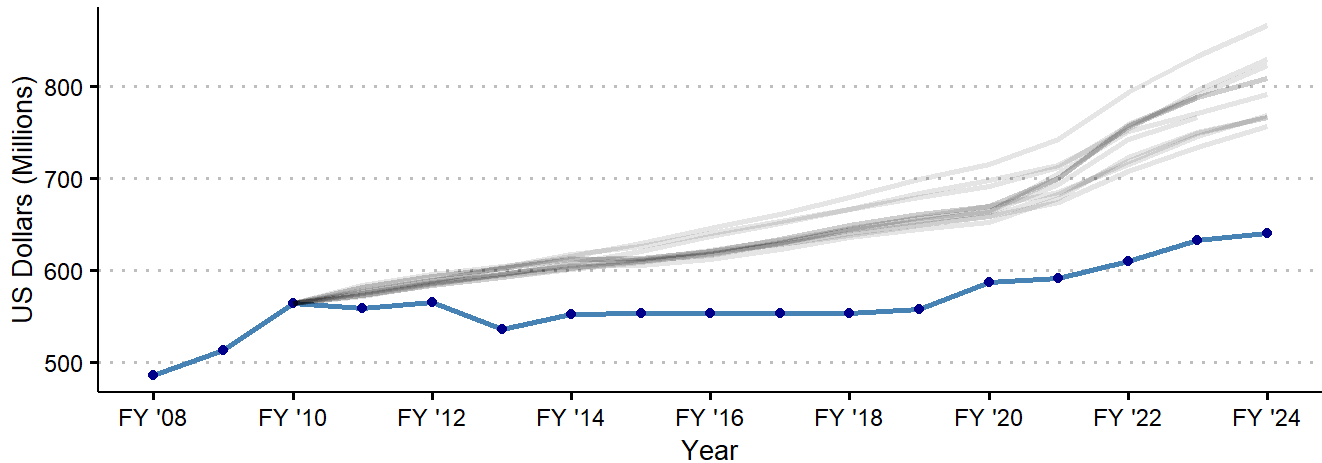
```
theme_clean()+
theme(
  plot.caption = element_text(hjust = 0, face = "italic"),
  legend.position = "bottom"
)
```

osha_budget_plot

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

OSHA's Budget and Inflation

Since the 2010 budget freeze, OSHA's budget has not kept up with inflation



— 2010 Inflation-Adjusted Budget Estimates — Actual Budget

Inflation data source: FRED ||| OSHA budget data source: Department of Labor Budget Reports (2009-2024)

Note: Multiple inflation rates were used, to give a range of estimates.

Note: Only discretionary spending was counted, not mandatory spending. This lowers the FY2020 budget.

Note: Whether the budget appears to have kept up with inflation depends on the chosen year. 2010 was chosen because it was the start of the decade-long budget freeze, in which inflation shrank the budget. OSHA's first FY, 1974, was not chosen, because inflation adjustments are less accurate over time.

```
# This is a diagnostic plot, for added transparency as to which inflation rates estimate what
inflation_breakdown_plot <-
  ggplot(
    data = filter(panel_rates, date >= parameter_graph_startyear),
    aes(x = date, y = value, color = series)
  ) +

  ### LINES
  geom_point() +
  geom_line() +

  ### LABELS
  labs(
    title = "OSHA Budget Inflation Estimates Breakdown",
    subtitle = "The estimate varies, depending on the inflation rate chosen",
    x = "Year",
    y = "US Dollars (Millions)",
    color = "FRED Code",
    caption = "Inflation data source: FRED ||| OSHA budget data source: Department of Labor B
udget Reports (2009-2024)
"

  ) +

  ### AXES
  scale_x_date(breaks = scales::pretty_breaks(n = 10),
    date_labels = paste0("FY '", "%y")) +

  ### THEMES
  theme_clean() +
  theme(plot.caption = element_text(hjust = 0, face = "italic"),
    legend.position = "right")

inflation_breakdown_plot
```

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

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

