

DSS Project

Cara Spitzer

2025-11-16

#Set up

```
setwd("C:/Users/Admin/OneDrive/LSE/extra/Data Science")

#install.packages("gridExtra")

library(tidyverse)

## Warning: Paket 'tidyverse' wurde unter R Version 4.5.2 erstellt

## — 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    3.5.2      ✓ tibble     3.3.0
## ✓ lubridate  1.9.4      ✓ tidyr      1.3.1
## ✓ purrr      1.1.0
## — Conflicts —
tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()     masks stats::lag()
## ⓘ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors

library(lubridate)
library(scales)

## Warning: Paket 'scales' wurde unter R Version 4.5.2 erstellt

##
## Attache Paket: 'scales'
##
## Das folgende Objekt ist maskiert 'package:purrr':
##
##   discard
##
## Das folgende Objekt ist maskiert 'package:readr':
##
##   col_factor

library(gridExtra)

## Warning: Paket 'gridExtra' wurde unter R Version 4.5.2 erstellt
```

```
##
## Attache Paket: 'gridExtra'
##
## Das folgende Objekt ist maskiert 'package:dplyr':
##
##      combine
```

Data prep

Harmonize the data sets of each asset

```
CLF = read.csv("CL_F.csv")%>%
  rename(fed_date = FOMC_Date,
         high = High,
         low = Low,
         close = Close,
         volume = Volume,
         open = Open,
         symbol = Ticker,
         timestamp = Date)

fed_date_v = unique(CLF$fed_date)
print(fed_date_v)

## [1] "2020-01-29" "2020-03-03" "2020-03-15" "2020-04-29" "2020-06-10"
## [6] "2020-07-29" "2020-09-16" "2020-11-05" "2020-12-16" "2021-01-27"
## [11] "2021-03-17" "2021-04-28" "2021-06-16" "2021-07-28" "2021-09-22"
## [16] "2021-11-03" "2021-12-15" "2022-01-26" "2022-03-16" "2022-05-04"
## [21] "2022-06-15" "2022-07-27" "2022-09-21" "2022-11-02" "2022-12-14"
## [26] "2023-02-01" "2023-03-22" "2023-05-03" "2023-06-14" "2023-07-26"
## [31] "2023-09-20" "2023-11-01" "2023-12-13" "2024-01-31" "2024-03-20"
## [36] "2024-05-01" "2024-06-12" "2024-07-31" "2024-09-18" "2024-11-07"
## [41] "2024-12-18" "2025-01-29" "2025-03-19" "2025-05-07" "2025-06-18"
## [46] "2025-07-30" "2025-09-17"

bitcoin = read.csv("bitcoin.csv")%>%
  filter(fed_date %in% fed_date_v)

gold = read.csv("GLD.csv")%>%
  rename(fed_date = FOMC_Date,
         high = High,
         low = Low,
         close = Close,
         volume = Volume,
         open = Open,
         symbol = Ticker,
         timestamp = Date)

c_energy = read.csv("ICLN.csv")%>%
  rename(fed_date = FOMC_Date)
```

```

SPY= read.csv("SPY.csv")%>%
  rename(fed_date = announcement_date)

dollar = read.csv("UUP.csv")%>%
  rename(fed_date = FOMC_Date,
         high = High,
         low = Low,
         close = Close,
         volume = Volume,
         open = Open,
         symbol = Ticker,
         timestamp = Date)

energy = read.csv("XLE.csv")%>%
  rename(fed_date = FOMC_Date)

ZN = read.csv("ZN_F.csv")%>%
  rename(fed_date = announcement_date)

all_assets = bind_rows(
  bitcoin,
  CLF,
  gold,
  c_energy,
  SPY,
  dollar,
  energy,
  ZN
)

all_assets_clean = all_assets%>%
  select(-c("Adj.Close", "Event_Trading_Day"))%>%
  mutate(
    timestamp = as_date(timestamp),
    fed_date = as_date(fed_date)
  )

all_assets_return = all_assets_clean%>%
  group_by(symbol) %>%
  arrange(timestamp) %>%
  mutate(
    daily_return = (close - lag(close))/lag(close) * 100
  ) %>%
  ungroup()

```

Summary Statistics

Summary of asset prices (closes) and return from 2020-2025 around the FED announcements.

##Comment: Maybe not the best timeframe, because of Covid- might be different circumstances

#---price---#

```
summary_prices = all_assets_clean %>%
  group_by(symbol) %>%
  summarise(
    n_FOMC = n_distinct(fed_date),
    min_close = min(close, na.rm = TRUE),
    max_close = max(close, na.rm = TRUE),
    avg_close = mean(close, na.rm = TRUE),
    sd_close = sd(close, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  arrange(symbol)

print(summary_prices)
```

```
## # A tibble: 8 × 6
##   symbol n_FOMC min_close max_close avg_close sd_close
##   <chr>   <int>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 BTC      47    5014.   117924.  44549.  30738.
## 2 CL=F     47     12.3    121.    69.5    19.5
## 3 GLD      47     141.    340.    196.    46.8
## 4 ICLN     47      7.95    30.5    16.5     4.40
## 5 SPY      47    220.    664.    430.    102.
## 6 UUP      47     21.6    29.5    25.1     2.25
## 7 XLE      47     18.8    91.8    64.0    22.2
## 8 ZN=F     47    106.    140.    121.    11.5
```

#---Return---#

```
summary_return = all_assets_return%>%
  summarise(
    n_fed_announcements = n_distinct(fed_date),
    avg_return = mean(daily_return, na.rm = TRUE),
    sd_return = sd(daily_return, na.rm = TRUE),
    min_return = min(daily_return, na.rm = TRUE),
    max_return = max(daily_return, na.rm = TRUE),
    .groups = "drop"
  )

print(summary_return)
```

```
## # A tibble: 1 × 5
##   n_fed_announcements avg_return sd_return min_return max_return
##               <int>   <dbl>   <dbl>   <dbl>   <dbl>
## 1                 47    0.459    6.14   -38.7    93.1
```

#Time series

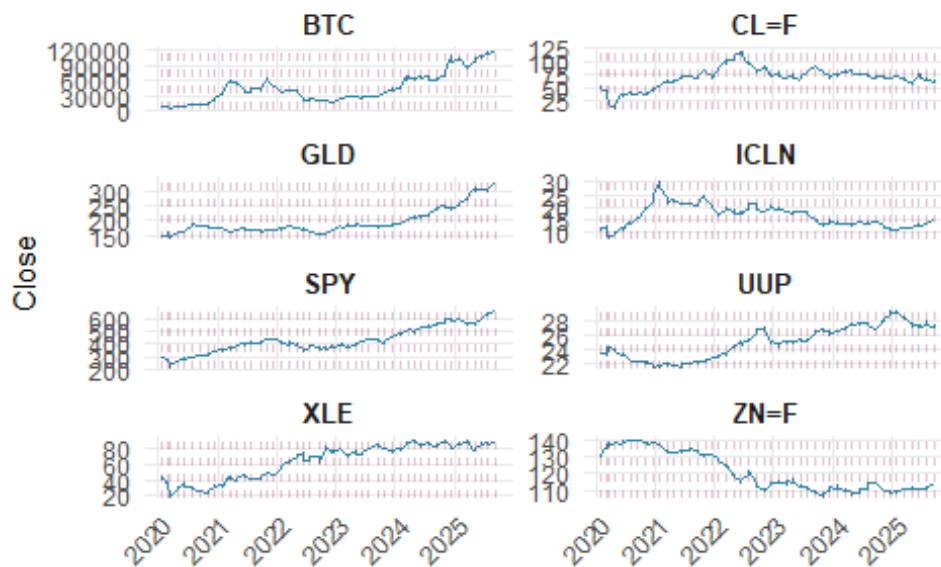
Development overtime around the fed announcements (dashed line) of prices and returns sperated by assets.

#Prices

```
plot_prices = ggplot(all_assets_clean, aes(x = timestamp, y = close)) +  
  geom_line(color = "#2E86AB", linewidth = 0.6) +  
  geom_vline(xintercept = as.Date(fed_date_v),  
            color = "#A23B72", alpha = 0.3, linetype = "dashed", linewidth =  
0.3) +  
  facet_wrap(~ symbol, scales = "free_y", ncol = 2) +  
  labs(  
    title = "Prices across asset groups (2020-2025)",  
    subtitle = "Vertical line: FOMC-Announcements",  
    x = "",  
    y = "Close",  
    caption = "Floating scales"  
  ) +  
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +  
  theme_minimal() +  
  theme(  
    plot.title = element_text(face = "bold", size = 14),  
    plot.subtitle = element_text(size = 10, color = "gray30"),  
    strip.text = element_text(face = "bold", size = 10),  
    axis.text.x = element_text(angle = 45, hjust = 1),  
    panel.grid.minor = element_blank()  
  )  
  
print(plot_prices)
```

Prices across asset groups (2020-2025)

Vertical line: FOMC-Announcements



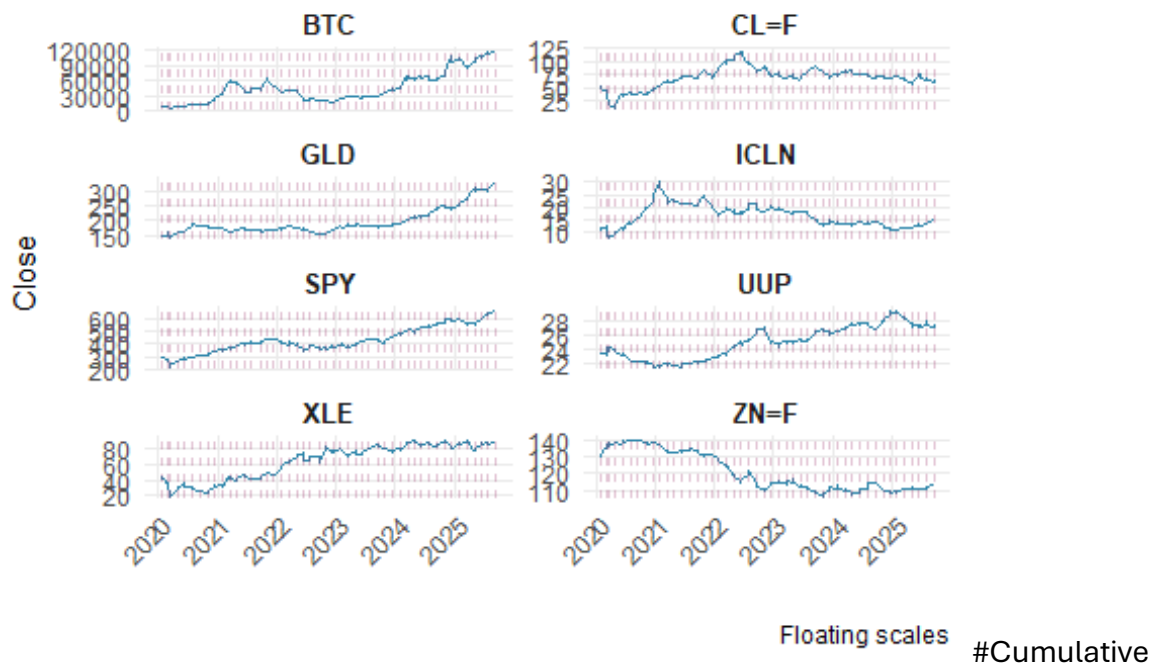
Floating scales

#Returns

```
plot_returns = ggplot(all_assets_return, aes(x = timestamp, y = close)) +
  geom_line(color = "#2E86AB", linewidth = 0.6) +
  geom_vline(xintercept = as.Date(fed_date_v),
             color = "#A23B72", alpha = 0.3, linetype = "dashed", linewidth =
0.3) +
  facet_wrap(~ symbol, scales = "free_y", ncol = 2) +
  labs(
    title = "Returns across asset groups (2020-2025)",
    subtitle = "Vertical line: FOMC-Announcements",
    x = "",
    y = "Close",
    caption = "Floating scales"
  ) +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 14),
    plot.subtitle = element_text(size = 10, color = "gray30"),
    strip.text = element_text(face = "bold", size = 10),
    axis.text.x = element_text(angle = 45, hjust = 1),
    panel.grid.minor = element_blank()
  )
print(plot_returns)
```

Returns across asset groups (2020-2025)

Vertical line: FOMC-Announcements



Returns The cumulative returns after the announcement.

##Prepare

```
all_assets_event = all_assets_return%>%
  mutate( days_from_fomc = as.numeric(timestamp -fed_date))%>%
  ungroup()
```

```
event_window = all_assets_event %>%
  filter(days_from_fomc >= 0 & days_from_fomc <= 5)
```

##cumulative returns

```
cumulative_returns = event_window %>%
  group_by(symbol, fed_date) %>%
  arrange(timestamp) %>%
  mutate(cumulative_return = cumsum(daily_return)
  ) %>%
  ungroup()
```

##avg

```
avg_cumulative = cumulative_returns %>%
  group_by(symbol, days_from_fomc) %>%
  summarise(
    avg_cum_return = mean(cumulative_return, na.rm = TRUE),
    sd_cum_return = sd(cumulative_return, na.rm = TRUE),
    n_events = n(),
```

```

    .groups = "drop"
  )

##graph

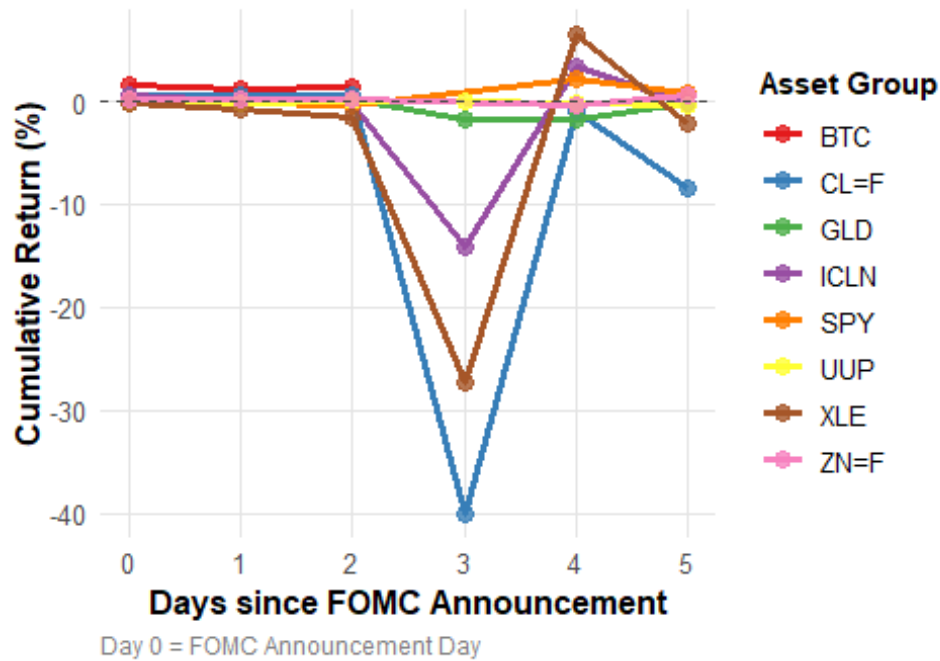
plot_cumulative_all = ggplot(avg_cumulative,
                             aes(x = days_from_fomc, y =
avg_cum_return,color = symbol, group = symbol))+
  geom_line(linewidth = 1.3) +
  geom_point(size = 3, alpha = 0.8) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "gray30", linewidth
= 0.6) +
  labs(
    title = "Cumulative Returns after FOMC Announcements",
    subtitle = "Average across all FOMC events (2020-2025)",
    x = "Days since FOMC Announcement",
    y = "Cumulative Return (%)",
    color = "Asset Group",
    caption = "Day 0 = FOMC Announcement Day"
  ) +
  scale_x_continuous(breaks = 0:5) +
  scale_color_brewer(palette = "Set1") +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(size = 11, color = "gray30"),
    axis.title = element_text(face = "bold", size = 12),
    axis.text = element_text(size = 10),
    legend.position = "right",
    legend.title = element_text(face = "bold", size = 11),
    legend.text = element_text(size = 10),
    panel.grid.minor = element_blank(),
    panel.grid.major = element_line(color = "gray90"),
    plot.caption = element_text(size = 9, color = "gray50", hjust = 0)
  )

print(plot_cumulative_all)

```


Cumulative Returns after FOMC Announce

Average across all FOMC events (2020-2025)



##Table

```
table_cumulative = avg_cumulative %>%
  select(symbol, days_from_fomc, avg_cum_return) %>%
  pivot_wider(
    names_from = days_from_fomc,
    values_from = avg_cum_return,
    names_prefix = "Day_"
  ) %>%
  arrange(symbol)

print(table_cumulative)
```

```
## # A tibble: 8 × 7
##   symbol Day_0 Day_1 Day_2 Day_3 Day_4 Day_5
##   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 BTC    1.51  1.27  1.36    NA    NA    NA
## 2 CL=F    0.573 0.660 0.529 -40.1 -0.909 -8.45
## 3 GLD    0.319 0.183 0.234 -1.78 -1.70 -0.274
## 4 ICLN    0.527 0.0962 -0.0998 -14.2  3.28 -0.233
## 5 SPY    0.145 -0.183 -0.301  NA    2.24  0.738
## 6 UUP   -0.129 -0.117 0.0621  0.0794 -0.111 -0.328
## 7 XLE   -0.132 -0.766 -1.51  -27.3  6.44 -2.16
## 8 ZN=F    0.151 0.255 0.281  NA   -0.342 0.549
```

#Volatility

The volatility before and after the announcements on average by asset group.

```

volatility_data = all_assets_event %>%
  filter(!is.na(daily_return)) %>%
  mutate(
    event_period = case_when(
      days_from_fomc >= -5 & days_from_fomc <= -1 ~ "Pre-Event",
      days_from_fomc >= 0 & days_from_fomc <= 5 ~ "Post-Event",
      TRUE ~ "Other"
    )
  )

volatility_stats = volatility_data %>%
  group_by(symbol, event_period) %>%
  summarise(
    volatility = sd(daily_return, na.rm = TRUE),
    mean_return = mean(daily_return, na.rm = TRUE),
    n_observations = n(),
    .groups = "drop"
  ) %>%
  arrange(symbol, event_period)

print(volatility_stats)

```

```

## # A tibble: 16 × 5
##   symbol event_period volatility mean_return n_observations
##   <chr>   <chr>         <dbl>      <dbl>          <int>
## 1 BTC     Post-Event      3.64       0.452           141
## 2 BTC     Pre-Event      16.7       3.47            93
## 3 CL=F    Post-Event      4.46      -0.0781         141
## 4 CL=F    Pre-Event     13.4       1.20            93
## 5 GLD     Post-Event      1.17       0.0387          141
## 6 GLD     Pre-Event      3.68       0.904           93
## 7 ICLN    Post-Event      2.66      -0.0540         141
## 8 ICLN    Pre-Event      8.24       0.780           93
## 9 SPY     Post-Event      1.74      -0.0578         140
## 10 SPY    Pre-Event      4.43       1.08            93
## 11 UUP     Post-Event      0.674      0.0262          141
## 12 UUP     Pre-Event      1.58       0.140           93
## 13 XLE     Post-Event      3.02      -0.510          141
## 14 XLE     Pre-Event     10.0       2.09            93
## 15 ZN=F    Post-Event      0.493      0.0884          140
## 16 ZN=F    Pre-Event      1.35      -0.281           93

```

##Graph

```

plot_volatility_bars <- ggplot(volatility_stats,
                               aes(x = symbol, y = volatility, fill =
event_period)) +
  geom_col(position = "dodge", width = 0.7) +
  geom_text(aes(label = sprintf("%.2f%%", volatility)),
            position = position_dodge(width = 0.7),
            vjust = -0.5, size = 3.5, fontface = "bold") +

```

```

labs(
  title = "Volatility Comparison: Pre-Event vs. Post-Event",
  subtitle = "Standard deviation of daily returns",
  x = "Asset Group",
  y = "Volatility (SD of Returns in %)",
  fill = "Period",
  caption = "Pre-Event: 5 days before FOMC | Post-Event: FOMC day + 5 days
after"
) +
scale_fill_manual(
  values = c("Pre-Event" = "#06A77D", "Post-Event" = "#D62246"),
  labels = c("Pre-Event (-5 to -1)", "Post-Event (0 to +5)")
) +
theme_minimal() +
theme(
  plot.title = element_text(face = "bold", size = 14),
  plot.subtitle = element_text(size = 11, color = "gray30"),
  axis.title = element_text(face = "bold", size = 11),
  axis.text.x = element_text(angle = 45, hjust = 1, face = "bold", size =
10),
  axis.text.y = element_text(size = 10),
  legend.position = "top",
  legend.title = element_text(face = "bold"),
  panel.grid.major.x = element_blank(),
  panel.grid.minor = element_blank()
)

print(plot_volatilityBars)

```

