Project 2 - Team 12

2025-03-14

Project 2 - Team 12

Needed packages

```
library(dplyr)
library(zoo)
library(ggplot2)
library(gridExtra)
library(frenchdata)
library(lubridate)
library(tidyr)
library(stringr)
library(zoo)
library(gt)
library(openxlsx)
library(quantmod)
library(xts)
library(TTR)
library(gridExtra)
library(tidyverse)
library(tidyquant)
library(slider)
library(broom)
```

Exercise 6: Estimate the rolling Equity Beta

To determine the Equity Beta, we use the Russel3000 Index, which includes Alnylam Pharmaceuticals, as market portfolio estimate. We use the following rolling formula to determine the Equity Beta (coming from the CAPM), which measures the sensitivity of a stock's returns relative to the market portfolio:

$$\beta_t = \frac{\text{Cov}(R_{e,t}, R_{m,t})}{\text{Var}(R_{m,t})}$$

where:

- β_t is the equity beta at time t,
- $\text{Cov}(R_e, R_m)$ is the covariance between the return on the asset R_e and the return of the market portfolio R_m,
- $\text{Var}(R_m)$ is the variance of the market return R_m.

Download and clean the data:

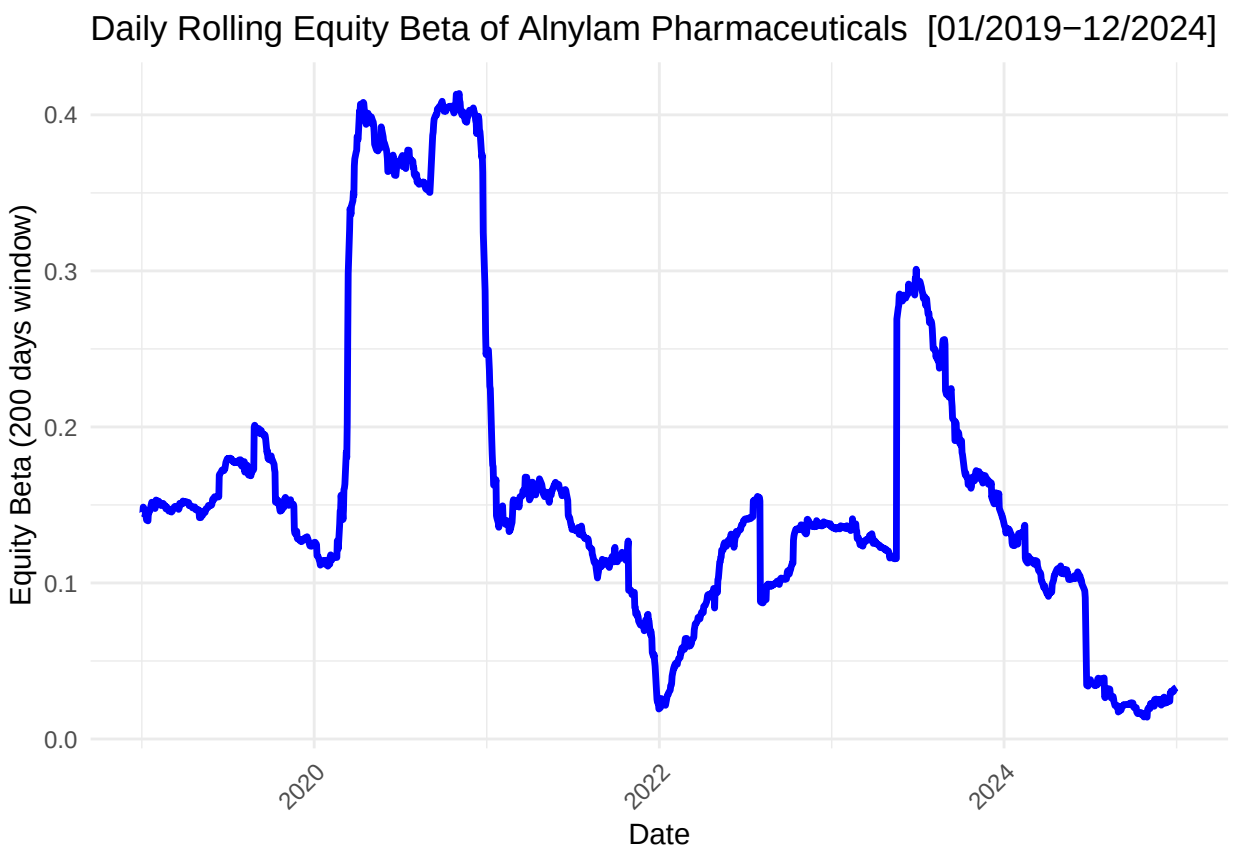
```
ALNY_PX <- read.xlsx("Daily_Data_Case_1.xlsx", sheet = 2)
ALNY_PX$Date <- as.Date(ALNY_PX$Date,origin = "1899-12-30")
```

```r
RUS_3_PX <- read.xlsx("Daily_Data_Case_1.xlsx", sheet = 1)
RUS_3_PX$Date <- as.Date(RUS_3_PX$Date,origin = "1899-12-30")

# Observe NA values

ALNY_NA <- sum(is.na(ALNY_PX$Last.Price))
RUS_3_NA <- sum(is.na(RUS_3_PX$Last.Price))

# One NA in ALNY_PX --> Lin. Interpolate

na <- which(is.na(ALNY_PX$Last.Price))

ALNY_PX$Last.Price[na] <- (ALNY_PX$Last.Price[na-1] + ALNY_PX$Last.Price[na+1]) * 0.5


# Combine the data frames to match the dates

Prices <- inner_join(ALNY_PX , RUS_3_PX , by = "Date")
colnames(Prices) <- c("Date", "ALNY", "RUS3")

# Create xts for quantmod
Prices_xts <- xts(Prices[,-1], order.by = as.Date(Prices$Date))
```

Equity Beta with daily returns

```r
ALNY_ret_d <- periodReturn(Prices_xts$ALNY, period = "daily", type = "arithmetic")
ALNY_ret_d <- data.frame(Date = index(ALNY_ret_d), coredata(ALNY_ret_d))

RUS3_ret_d <- periodReturn(Prices_xts$RUS3, period = "daily", type = "arithmetic")
RUS3_ret_d <- data.frame(Date = index(RUS3_ret_d), coredata(RUS3_ret_d))

Returns_daily <- inner_join(RUS3_ret_d  , ALNY_ret_d  , by = "Date")[-1,]
colnames(Returns_daily) <- c("Date", "ALNY", "RUS3")

start_date <- which(Returns_daily$Date == as.Date("2019-01-02")) - 200
Returns_daily <- Returns_daily[-(1:(start_date)),]

# Market Return Variance

Returns_daily$RUS3_var <- rollapply(Returns_daily$RUS3, width = 200, FUN = var, align = "right", fill =

Returns_daily$rolling_cov <- runCov(
  x = Returns_daily$ALNY,
  y = Returns_daily$RUS3,
  n = 200
)

Returns_daily <- Returns_daily %>%
  mutate(
    E_Beta_daily = rolling_cov / RUS3_var
  )
```

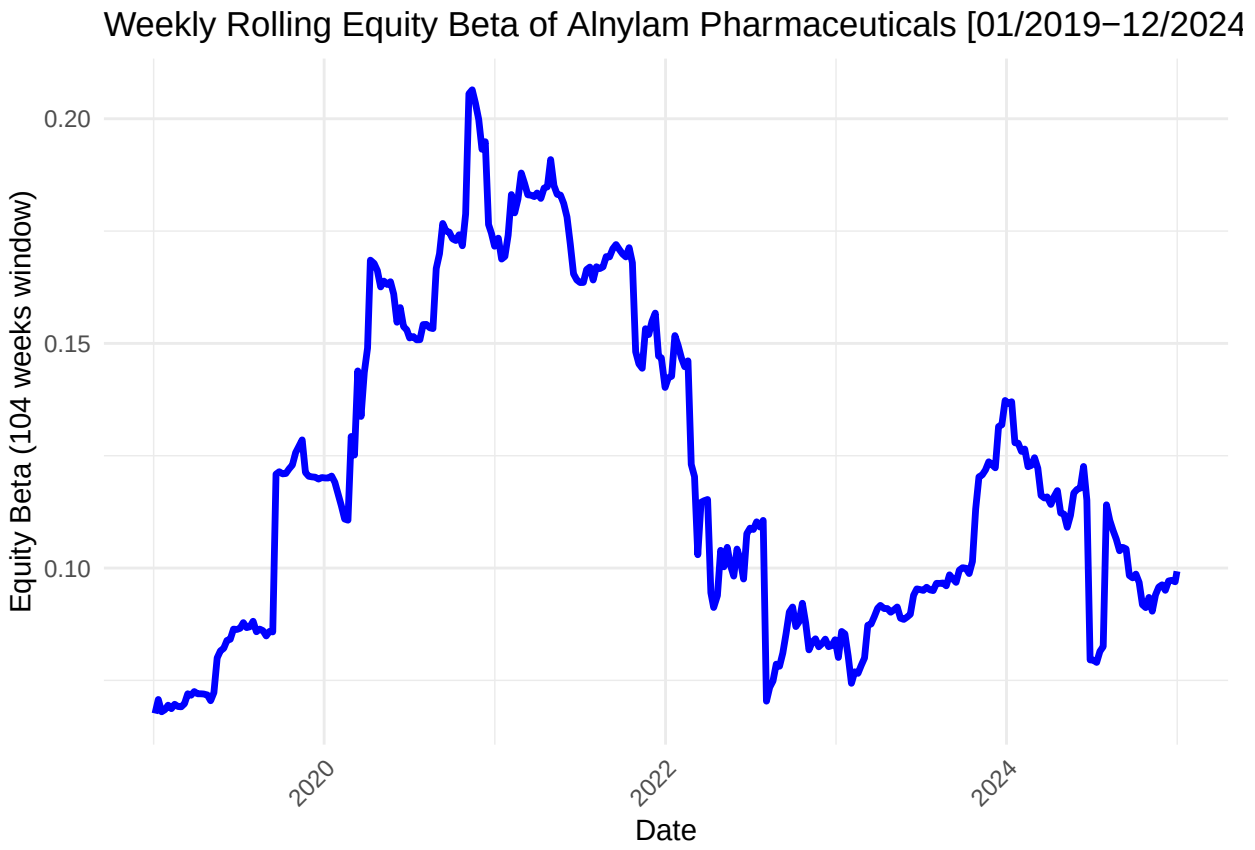
```r
# Remove na rows
Returns_daily <- na.omit(Returns_daily)

ggplot(Returns_daily, aes(x = Date, y = E_Beta_daily)) +
  geom_line(linewidth = 1.2,color = "blue") +
  labs(title = "Daily Rolling Equity Beta of Alnylam Pharmaceuticals  [01/2019-12/2024]",
       x = "Date",
       y = "Equity Beta (200 days window)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Equity Beta for weekly returns

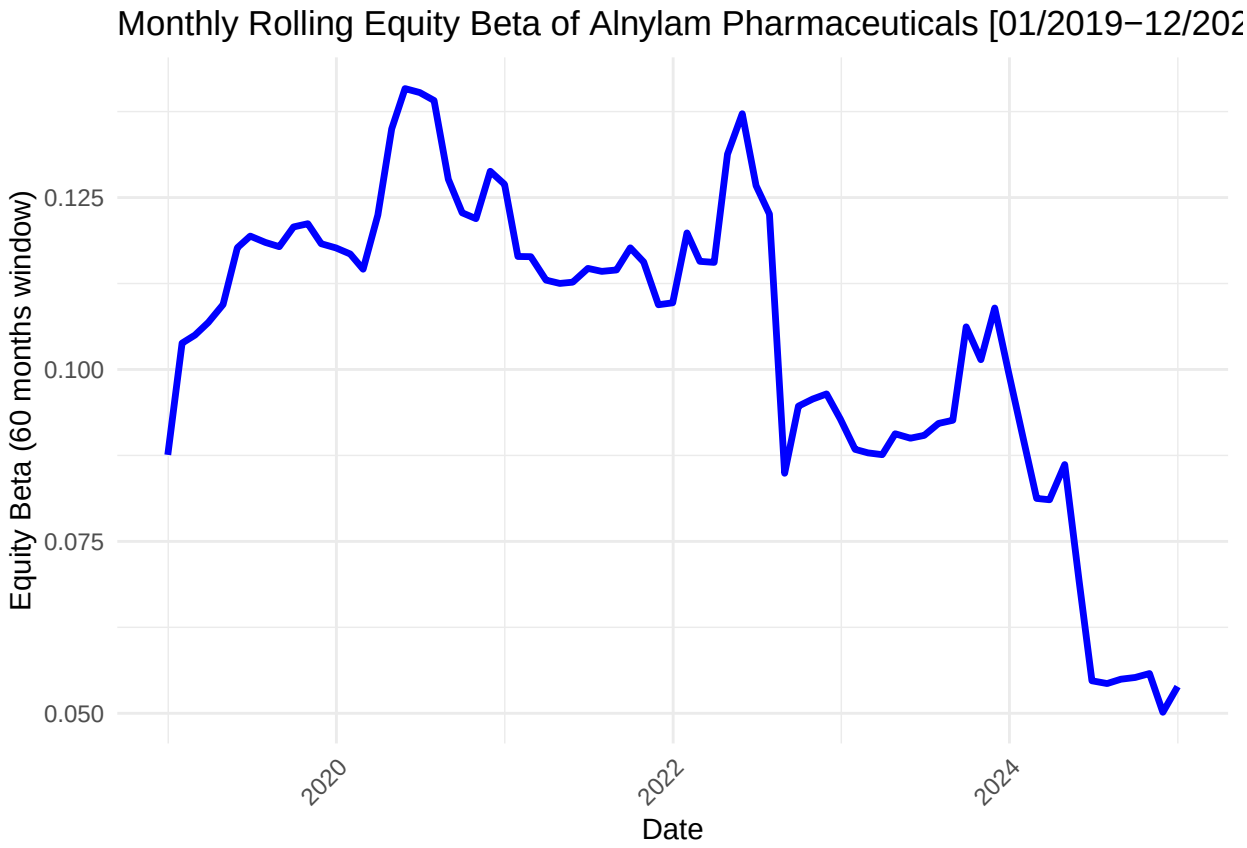
```r
ALNY_ret_w <- periodReturn(Prices_xts$ALNY, period = "weekly", type = "arithmetic")
ALNY_ret_w <- data.frame(Date = index(ALNY_ret_w), coredata(ALNY_ret_w))

RUS3_ret_w <- periodReturn(Prices_xts$RUS3, period = "weekly", type = "arithmetic")
RUS3_ret_w <- data.frame(Date = index(RUS3_ret_w), coredata(RUS3_ret_w))

Returns_weekly <- inner_join(RUS3_ret_w  , ALNY_ret_w  , by = "Date")[-1,]
colnames(Returns_weekly) <- c("Date", "ALNY", "RUS3")

start_date <- which(Returns_weekly$Date == as.Date("2019-01-04")) - 104
Returns_weekly <- Returns_weekly[-(1:(start_date)),]

# Market Return Variance
```

```r
Returns_weekly$RUS3_var <- rollapply(Returns_weekly$RUS3, width = 104, FUN = var, align = "right", fill

Returns_weekly$rolling_cov <- runCov(
  x = Returns_weekly$ALNY,
  y = Returns_weekly$RUS3,
  n = 104
)

Returns_weekly <- Returns_weekly %>%
  mutate(
    E_Beta_weekly = rolling_cov / RUS3_var
  )

# Remove na rows
Returns_weekly <- na.omit(Returns_weekly)


ggplot(Returns_weekly, aes(x = Date, y = E_Beta_weekly)) +
  geom_line(linewidth = 1.2,color = "blue") +
  labs(title = "Weekly Rolling Equity Beta of Alnylam Pharmaceuticals [01/2019-12/2024]",
       x = "Date",
       y = "Equity Beta (104 weeks window)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Equity Beta for monthly returns

```
ALNY_ret_m <- periodReturn(Prices_xts$ALNY, period = "monthly", type = "arithmetic")
ALNY_ret_m <- data.frame(Date = index(ALNY_ret_m), coredata(ALNY_ret_m))

RUS3_ret_m <- periodReturn(Prices_xts$RUS3, period = "monthly", type = "arithmetic")
RUS3_ret_m <- data.frame(Date = index(RUS3_ret_m), coredata(RUS3_ret_m))

Returns_monthly <- inner_join(RUS3_ret_m  , ALNY_ret_m  , by = "Date")[-1,]
colnames(Returns_monthly) <- c("Date", "ALNY", "RUS3")

start_date <- which(Returns_monthly$Date == as.Date("2018-12-31")) - 60
Returns_monthly <- Returns_monthly[-(1:(start_date)),]

# Market Return Variance

Returns_monthly$RUS3_var <- rollapply(Returns_monthly$RUS3, width = 60, FUN = var, align = "right", fil

Returns_monthly$rolling_cov <- runCov(
  x = Returns_monthly$ALNY,
  y = Returns_monthly$RUS3,
  n = 60
)

Returns_monthly <- Returns_monthly %>%
  mutate(
    E_Beta_monthly = rolling_cov / RUS3_var
  )

# Remove na rows
Returns_monthly <- na.omit(Returns_monthly)


ggplot(Returns_monthly, aes(x = Date, y = E_Beta_monthly)) +
  geom_line(linewidth = 1.2,color = "blue") +
  labs(title = "Monthly Rolling Equity Beta of Alnylam Pharmaceuticals [01/2019-12/2024]",
       x = "Date",
       y = "Equity Beta (60 months window)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

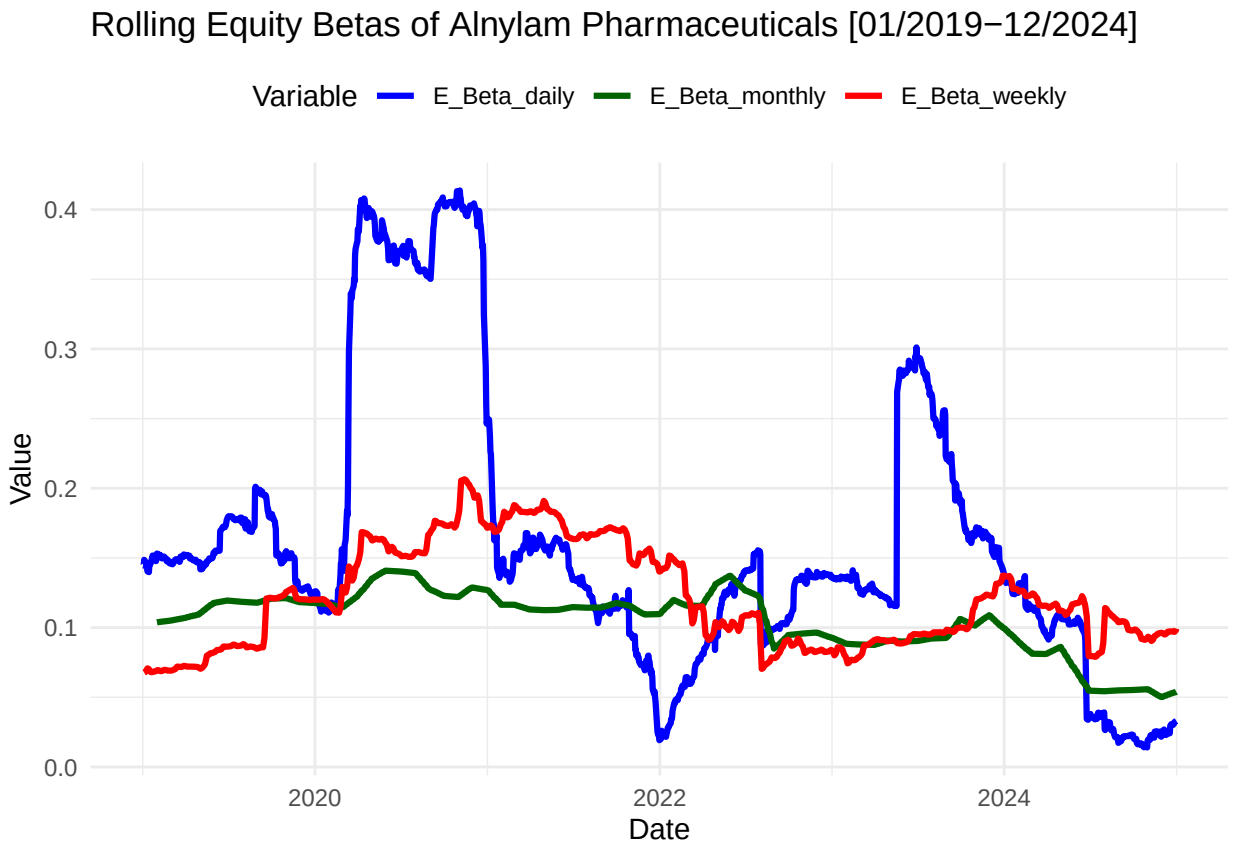


Plot every Rolling Method

```
result1 <- Returns_daily[,c(1,6)] %>%
  full_join(Returns_weekly[,c(1,6)], by = "Date") %>%
  full_join(Returns_monthly[,c(1,6)], by = "Date")

# Interpolate missing values for each variable
result1_filled <- result1 %>%
  mutate(across(-Date, ~ zoo::na.approx(.x, na.rm = FALSE)))

# Reshape data to long format
result1_long <- result1_filled %>%
  pivot_longer(cols = -Date, names_to = "variable", values_to = "value")

# Plot using ggplot2
# Plot with separate lines for each variable
ggplot(result1_long, aes(x = Date, y = value, color = variable, group = variable)) +
  geom_line(size = 1.1, na.rm = TRUE) +  # Ignore NAs for drawing lines
  labs(title = "Rolling Equity Betas of Alnylam Pharmaceuticals [01/2019-12/2024]",
       x = "Date",
       y = "Value",
       color = "Variable") +
  theme_minimal() +
  theme(legend.position = "top")+
  scale_color_manual(values = c(E_Beta_daily = "blue",E_Beta_weekly  = "red",E_Beta_monthly  = "darkgre
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
```

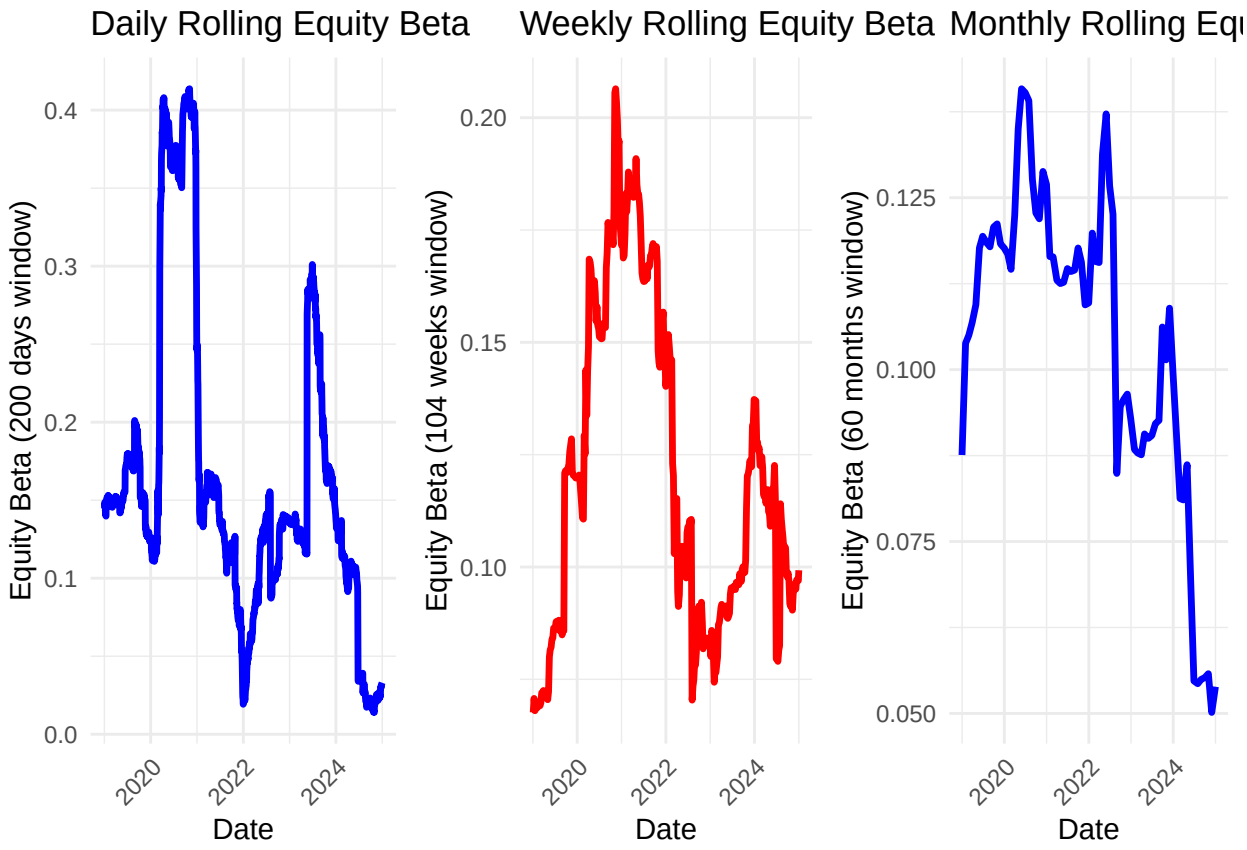
```
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Additional Plots

```
# Rolling Plots Side by Side
plot_daily <- ggplot(Returns_daily, aes(x = Date, y = E_Beta_daily)) +
  geom_line(linewidth = 1.2, color = "blue") +
  labs(title = "Daily Rolling Equity Beta",
       x = "Date",
       y = "Equity Beta (200 days window)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
plot_weekly <- ggplot(Returns_weekly, aes(x = Date, y = E_Beta_weekly)) +
  geom_line(linewidth = 1.2, color = "red") +
  labs(title = "Weekly Rolling Equity Beta",
       x = "Date",
       y = "Equity Beta (104 weeks window)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
plot_monthly <- ggplot(Returns_monthly, aes(x = Date, y = E_Beta_monthly)) +
  geom_line(linewidth = 1.2,color = "blue") +
  labs(title = "Monthly Rolling Equity Beta",
       x = "Date",
       y = "Equity Beta (60 months window)") +
```

```
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

grid.arrange(plot_daily, plot_weekly, plot_monthly, ncol = 3)
```




```
# Table with Averages and Confidence Interval

E_Beta_d <- Returns_daily$E_Beta_daily
E_Beta_w <- Returns_weekly$E_Beta_weekly
E_Beta_m <- Returns_monthly$E_Beta_monthly

# Calculate mean and 95% confidence interval for each series (ASSUMING NORMALITY)
mean_d <- mean(E_Beta_d)
ci_d <- t.test(E_Beta_d, conf.level = 0.95)$conf.int
mean_w <- mean(E_Beta_w)
ci_w <- t.test(E_Beta_w, conf.level = 0.95)$conf.int
mean_m <- mean(E_Beta_m)
ci_m <- t.test(E_Beta_m, conf.level = 0.95)$conf.int

# Create a data frame with the results
results <- data.frame(
  Series = c("Daily", "Weekly", "Monthly"),
  Mean = round(c(mean_d, mean_w, mean_m),4),
  CI_Lower = round(c(ci_d[1], ci_w[1], ci_m[1]),4),
  CI_Upper = round(c(ci_d[2], ci_w[2], ci_m[2]),4)
)
```

Average Equity Betas and Confidence Intervals for Alnylam Pharmaceuticals [01/2019-12/2024]

Rolling Type	Avgerage Equity Beta	CI Lower Limit	CI Upper Limit
Daily	0.1623	0.1572	0.1675
Weekly	0.1209	0.1168	0.1249
Monthly	0.1044	0.0992	0.1097

```
colnames(results) <- c("Rolling Type","Avgerage Equity Beta", "CI Lower Limit", "CI Upper Limit")

results %>%
  gt() %>%
  tab_style(
    style = cell_borders(sides = "right", color = "grey80", weight = px(2)),
    locations = cells_body(columns = everything())
  ) %>%
  tab_style(
    style = cell_borders(sides = "right", color = "grey80", weight = px(2)),
    locations = cells_column_labels(columns = everything())
  ) %>%
  tab_header(title = "Average Equity Betas and Confidence Intervals for Alnylam Pharmaceuticals [01/201
```

Exercise 9

1. Get Fama/French Weekly Data

```
FF_weekly <- download_french_data("Fama/French 3 Factors [Weekly]")$subsets[, 2]$data[[1]] %>%
  mutate(date = ymd(date)) %>%
  filter(between(date, ymd("2017-01-01"), ymd("2024-12-31")))
```

```
## New names:
## * `` -> `...1`
```

2. Get ALNY Prices and Weekly Returns

```
prices_alyn_weekly <- tq_get("ALNY", from = "2017-01-01", to = "2024-12-31") %>%
  tq_transmute(select = adjusted,
               mutate_fun = periodReturn,
               period = "weekly",
               type = "log",
               col_rename = "return_weekly") %>%
  filter(between(date, ymd("2017-01-01"), ymd("2024-12-31")))
```

3. Join FF Factors and Stock Returns

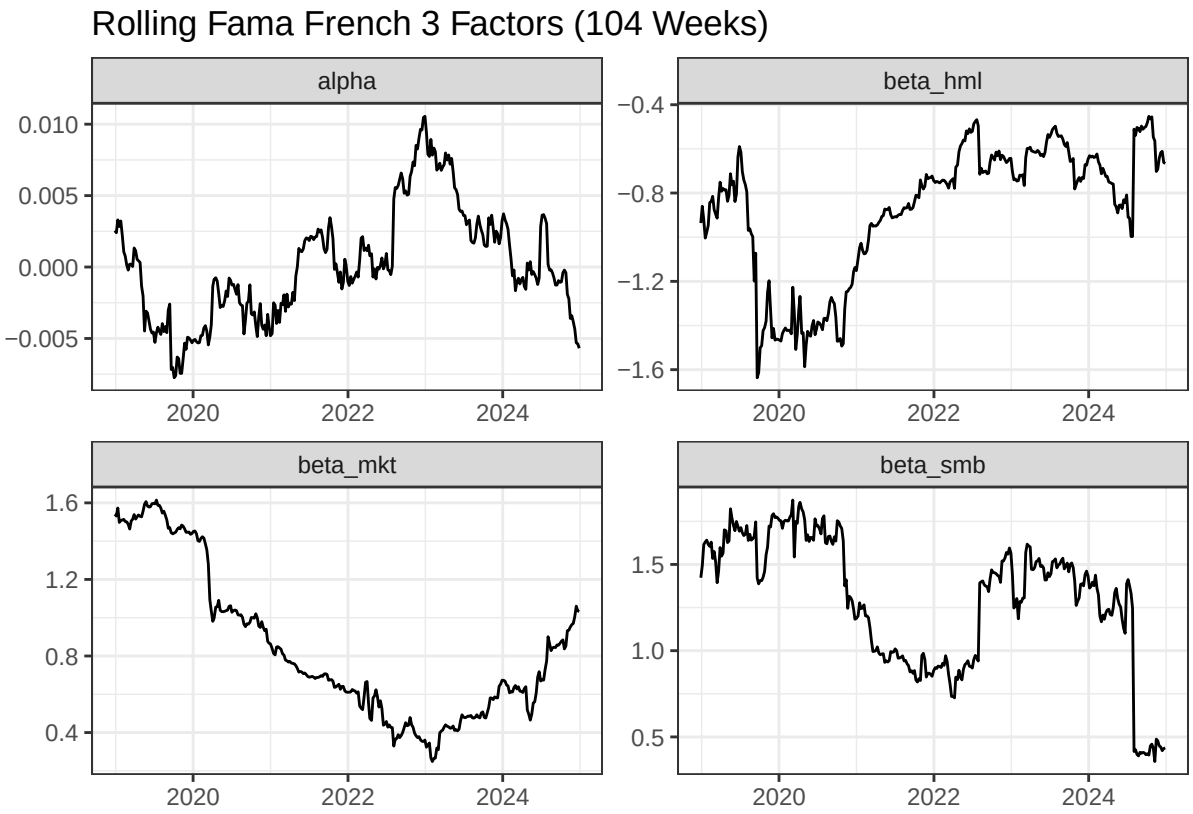
```
ff_smci <- FF_weekly %>%
  mutate(across(c(`Mkt-RF`, SMB, HML, RF), ~ .x / 100)) %>%  # convert to decimals
  left_join(prices_alyn_weekly, by = "date") %>%
  mutate(excess_return = return_weekly - RF)
```

4. Rolling Regression - 104 Weeks Window

```
rolling_betas <- slide_dfr(
  .x = ff_smci,
  .f = ~ {
    # Only keep rows with no NAs in required columns
    window_data <- .x %>% drop_na(excess_return, `Mkt-RF`, SMB, HML)
     # Check if we have at least 104 rows after removing NAs
    if (nrow(window_data) < 104) return(tibble())

    fit <- lm(excess_return ~ `Mkt-RF` + SMB + HML, data = window_data)
    tibble(
      date = max(window_data$date),
      alpha = coef(fit)[1],
      beta_mkt = coef(fit)[2],
      beta_smb = coef(fit)[3],
      beta_hml = coef(fit)[4]
    )
  },
  .before = 103,
  .complete = TRUE
)
```

5. Plot Rolling Betas

```
rolling_betas %>%
  pivot_longer(-date) %>%
  ggplot(aes(x = date, y = value)) +
  geom_line() +
  facet_wrap(~name, scales = "free") +
  labs(x = "", y = "", title = "Rolling Fama French 3 Factors (104 Weeks)") +
  theme_bw()
```



Exercise 10

```
rolling_comparison <- FF_weekly %>%
  left_join(prices_alyn_weekly, by = "date") %>%
  mutate(across(c("Mkt-RF", SMB, HML), ~ .x / 100)) %>%
  slide(.x = ., .f = identity, .before = 103) %>%
  keep(~ nrow(.x) == 104) %>%
  map_dfr(~ {
    # Remove NAs before regression
    data_clean <- drop_na(.x, return_weekly, `Mkt-RF`, SMB, HML)

    # Skip if not enough rows
    if (nrow(data_clean) < 104) return(tibble())

    # Run both regressions
    ff_model <- lm(return_weekly ~ `Mkt-RF` + SMB + HML, data = data_clean)
    capm_model <- lm(return_weekly ~ `Mkt-RF`, data = data_clean)

    # Tidy results with confidence intervals
    ff_tidy <- tidy(ff_model, conf.int = TRUE, conf.level = 0.95) %>% mutate(model = "FF")
    capm_tidy <- tidy(capm_model, conf.int = TRUE, conf.level = 0.95) %>% mutate(model = "CAPM")

    # Combine results and tag with date
    bind_rows(ff_tidy, capm_tidy) %>%
      mutate(date = max(data_clean$date))
  })
```

Plotting

```
rolling_comparison %>%
  ggplot(aes(x = date, y = estimate, color = model, fill = model)) +
  scale_color_manual(values = c("orange", "blue")) +
  scale_fill_manual(values = c("orange", "blue")) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high, color = NULL),  alpha = 0.3) +
  geom_line(size = 1.1) +
  geom_hline(yintercept = 0, color = "darkred", size = 1.1, linetype = "dashed") +
  facet_wrap(~term, scales = "free_y") +
  labs(x = "", y = "", title = "Comparison of Rolling CAPM and Fama French Coefficients") +
  theme_classic() +
  theme(legend.title = element_blank())
```