

HW1

Erik Andersen

2024-04-05

```
# Here setup
here::i_am("HW1/code/HW1.Rmd")

## here() starts at /Users/erikandersen/Documents/Classes/FIN-592

# Load packages
pacman::p_load(haven, tidyverse, magrittr, ggplot2, gridExtra, stargazer, kableExtra)

# Load data
brets_df = read_dta(here::here("HW1", "data", "brets.dta"))
tretts_df = read_dta(here::here("HW1", "data", "tretts.dta"))
```

To begin, I will define some helper functions so that I don't have to repeat the same code over and over. This first one first calculates the variance for each firm, then aggregates it over the event window. Finally I calculate the aggregate variance and standard error.

```
# Function to Calculate SE for each firm
firm_var = function(df){
  # Define iterator
  firms = unique(unlist((df["permno"])))

  lapply(firms, function(x){
    # Get the abnormal returns in the estimation period for each firm
    epsilon = df |>
      # Estimation window is 3 months before 0 date
      filter(rd <= -61) |>
      filter(permno == {x}) |>
      select(abnorm) |>
      drop_na() |>
      as.matrix() # Make a matrix for matrix algebra in next step

    # Calculate the variance. We want to keep the variances rather than calculate SE because we need to s
    sigma2 = t(epsilon) %*% epsilon / (nrow(epsilon) - 2)
    return(sigma2)
  }) |> unlist()
}

# Now calculate the SE for each firm within the range of interest
cum_var = function(df, from, to, firm_var){
  # Define iterator
  firms = unique(unlist((df["permno"])))

  var = lapply(firms, function(x){
    # Calculate the length of the series for each firm
```

```

N = df |>
  filter(permno == {x}) |>
  drop_na() |>
  nrow()

# Make diagonal variance matrix
V = diag(firm_var[which(unique(df$permno) == x)], nrow = N)

# Create vector of 1's and 0 to indicate the event days. Since the matrix is symmetric, we don't
gamma = c(rep(1, length(from:to)), rep(0, N - length(from:to)))

# Calculate firm variance within the window
firm_cum_var = t(gamma) %*% V %*% gamma

return(firm_cum_var)
}) |> unlist()

# Now calculate variance aggregating over firms
agg_var = sum(var) / length(var)^2
agg_se = sqrt(agg_var)

return(agg_se)
}

```

This function calculates the cumulative returns for each firm then aggregates over firms at each date in the event window.

```

# Calculate the cumulative abnormal returns
cumulative_returns = function(df, from, to){
  # Note: event_window should be a two long vector with the first value the lower bound of the event window
  # Define iterator
  firms = unique(unlist((df["permno"])))

  # Define event window for each firm and get cumulative returns over the window
  car_firms = sapply(firms, function(x){
    df |>
      # Select firms
      filter(permno == {x}) |>
      # Add a column of 1's and 0's where 1's indicate value is in the event window
      mutate(event = if_else(rd >= from & rd <= to, 1, 0)) |>
      filter(event == 1) |>
      drop_na() |>
      select(abnorm) |>
      cumsum()
  })

  # Remove vector names
  sapply(1:length(car_firms), function(i){
    names(car_firms[[i]]) <- NULL
  })

  # Unlist
  car_firms = unlist(car_firms)
}

```

```

# Aggregate over firms
car = sapply(seq(1, length(from:to)), function(i){
  # Select returns for each date across all firms
  if(length(from:to) == 1)
    date = car_firms
  else
    date = car_firms[names(car_firms) == paste("abnorm", i, sep = "")]

  # Aggregate cumulative abnormal returns at each date in the event window
  sum(date) / length(date)
})

return(car)
}

```

This is the crucial function. This calls the previous helper functions and gives us our test statistic and p value.

```

# Calculate the test statistic and return t stat and p value
test_stat = function(df, from, to){
  # Calculate SE
  se = cum_var(df, from, to, firm_var = firm_var(df))

  # Calculate cumulative returns. We just need the final one for the test stat
  car = cumulative_returns(df, from, to)
  car = car[length(car)]

  # Calculate test stat which is car/se
  j = car / se

  # Calculate p value
  p = 2*(1 - pnorm(abs(j)))

  return(list(`Cumulative Abnormal Return` = car, `Standard Error` = se, `T-stat` = j, `P Value` = p))
}

```

This one plots the CAR and standard error over a 30 period window around the event.

```

# Plot the CAR
plot_CAR = function(df){

  se = cum_var(df, -30,30, firm_var(df))

  # Calculate cumulative returns
  ret = cumulative_returns(df, -30, 30)

  # Make into a dataframe to plot
  cum = tibble(abnorm = ret, date = -30:30, se = se)

  cum |>
    drop_na() |>
    ggplot(aes(x = date, y = abnorm)) +
    geom_point() +
    geom_ribbon(aes(ymin = abnorm - 2*se, ymax = abnorm + 2*se), alpha = 0.3) +
    geom_vline(xintercept = 0, linewidth = 1.5, alpha = 0.5) +

```

```

scale_x_continuous(breaks = seq(-30,30, by = 10)) +
cowplot::theme_cowplot() +
labs(x = "Month Relative to Merger", y = "Cumulative Abnormal Return")
}

```

This calculates the mean returns for each firm. I don't make it a function since I only do it once.

```

mean_returns = function(df, error = 0){
  # for convenience define a vector of all the firm ids
  firms = unique(unlist((df["permno"])))

  # Calculate mean returns
  returns = lapply(firms, function(x){
    df |>
      filter(rd <= -61) |>
      filter(permno == {x}) |>
      reframe(permno = permno, mean_return = mean(ret, na.rm = T)) |>
      unique()
  }) |> bind_rows()

  df = df |>
    left_join(returns) |>
    mutate(abnorm = ret - mean_return)
}

```

This is another helper function to calculate the CAPM model for a given data set.

```

# Estimate CAPM returns model for target firms
CAPM_returns = function(df, error = 0){
  # Define iterator
  firms = unique(unlist((df["permno"])))

  returns = lapply(firms, function(x) {
    # Calculate beta for each firm
    df |>
      filter(rd <= -61) |>
      #drop_na() |>
      filter(permno == {x}) %>%
      reframe(permno = permno, constant = coef(lm(ret ~ vwret, .))[1], beta = coef(lm(ret ~ vwret, .))
      unique()
  }) |> bind_rows()

  # Merge capm coefficients back
  df = left_join(df, returns)

  # Get predictions and abnormal returns
  # For latter use this if else statement lets me calculate the abnormal returns for either the point e
  df = df |>
    mutate(capm = ifelse(error == 0, vwret * beta + constant,
                          ifelse(error == 1, constant + vwret * (beta + se), constant + vwret * (beta
                          abnorm = ret - capm)

  return(df)
}

```

Question 1: Short Horizon Target

Now we have defined helper functions, actually calculating the relevant statistics is simple. Below I calculate them for the mean model and plot the graph. Soon I will do the CAPM model and print the result.

```
# Calculate mean returns
tretts_mean = mean_returns(tretts_df)

## Joining with `by = join_by(permmo)`
# Calculate the relevant statistics for each event window
means = sapply(c(0,1,2,5), function(x){
  test_stat(tretts_mean, 0-x, 0+x)
})

colnames(means) = c(0,1,2,5)

means_plot = plot_CAR(tretts_mean) + ggtitle("Mean Returns Model")
```

Now we have that we can calculate the CAPM returns for the target firms.

```
# Add capm predictions to data set
tretts_capm = CAPM_returns(tretts_df)

## Joining with `by = join_by(permmo)`

Finally we can calculate the relevant statistics and graph it.

# CAPM results for 0, 1, 2, and 5 day windows
capm = sapply(c(0,1,2,5), function(x){
  test_stat(tretts_capm, 0-x, 0+x)
})

colnames(capm) = c(0,1,2,5)

# Plot
capm_plot = plot_CAR(tretts_capm) + ggtitle("CAPM Model")
```

a) Here are the relevant statistics from the two models

```
one_table = kbl(
  cbind(means, capm),
  format = "latex",
  longtable = F,
  booktabs = T,
  caption = "Target Firms 30 Day Event Window",
  digits = 2
) |>
  add_header_above(c(" ", "Mean Returns" = 4, "CAPM" = 4)) |>
  kable_styling(latex_options = c("striped", "scale_down", "repeat_header", "hold_position"))
```

The first thing to notice about the two models is that they produce virtually identical results. If you squint at the graphs, you will notice that the CAPM has slightly higher abnormal abnormal cumulative returns, but it is basically unnoticeable. The difference is slightly more noticeable by looking at the T-statistics in the table. The slightly higher CARs for the CAPM mean the T-stats are consistently 5-10 points higher than the means model. However these T-statistics are all so enormous that its not relevant at all. The smallest T-statistic I produced was 65 which is comically high. Clearly whatever model you use here mergers had an effect on the stock price of target firms.

Question 2: Short Horizon Bidders

```
# Estimate the capm for the bidding firms
brets_capm = CAPM_returns(brets_df)

## Joining with `by = join_by(permno)`
# CAPM results for 0, 1, 2, and 5 day windows
capm_B = sapply(c(0,1,2,5), function(x){
  test_stat(brets_capm, 0-x, 0+x)
})

# Plot results
capm_B_plot = plot_CAR(brets_capm) + ggtitle("CAPM Model for Bidding Firms")
```

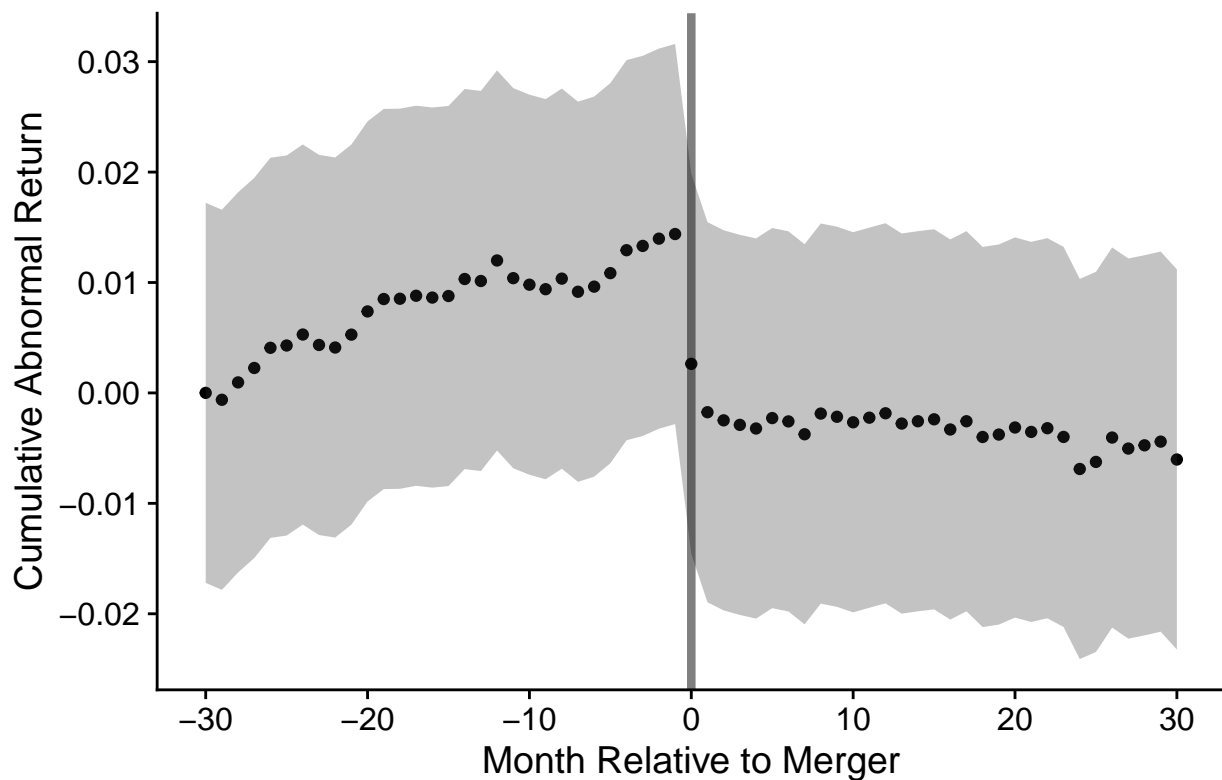
a) Here are the tables of the results

```
two_table = kbl(
  capm_B,
  format = "latex",
  longtable = F,
  booktabs = T,
  caption = "Bidding Firms 30 Day Event Window. CAPM Model",
  digits = 2
) |>
  add_header_above(c(" ", "CAPM" = 4)) |>
  kable_styling(latex_options = c("striped", "scale_down", "repeat_header", "hold_position"))
```

b) Here is the graph

capm_B_plot

CAPM Model for Bidding Firms



c)

Question 3: Long Horizon Bidders

*# Calculate abnormal returns for three CAR series: regular, and plus or minus one se
I have to make three different data frames because of how my functions work and I don't want to chang*

```
long_capm_point_df = CAPM_returns(brets_df, error = 0)
```

```
## Joining with `by = join_by(permno)`
```

```
long_capm_plus_df = CAPM_returns(brets_df, error = 1)
```

```
## Joining with `by = join_by(permno)`
```

```
long_capm_minus_df = CAPM_returns(brets_df, error = -1)
```

```
## Joining with `by = join_by(permno)`
```

Calculate average of beta standard errors

```
beta_se = long_capm_point_df$se |> unique() |> mean()
```

a)

CAPM results for 0, 1, 2, and 5 day windows for point estimate

```
capm_point = sapply(c(0,1,2,5), function(x){
```

```

  test_stat(long_capm_point_df, 0-x, 0+x)
})
colnames(capm_point) = c(0,1,2,5)

capm_plus = sapply(c(0,1,2,5), function(x){
  test_stat(long_capm_plus_df, 0-x, 0+x)
})
colnames(capm_plus) = c(0,1,2,5)

capm_minus = sapply(c(0,1,2,5), function(x){
  test_stat(long_capm_minus_df, 0-x, 0+x)
})
colnames(capm_minus) = c(0,1,2,5)

```

b) Here is a table of the relevant stats

```

three_table = kbl(
  cbind(capm_minus, capm_point, capm_plus),
  format = 'latex',
  longtable = F,
  booktabs = T,
  caption = "Target Firms 30 Day Event Window",
  digits = 2
) |>
  add_header_above(c(" ", "Minus" = 4, "Point" = 4, "Plus" = 4)) |>
  kable_styling(latex_options = c("striped", "scale_down", "repeat_header", "hold_position"))

```

c) Now we'll plot the three different series on the same plot.

```

# Calculate cumulative returns
ret_point = cumulative_returns(long_capm_point_df, -30, 30)
ret_plus = cumulative_returns(long_capm_plus_df, -30, 30)
ret_minus = cumulative_returns(long_capm_minus_df, -30, 30)

# Make into a dataframe to plot
cum = tibble(
  abnorm_point = ret_point,
  abnorm_plus = ret_plus,
  abnorm_minus = ret_minus,
  date = -30:30
)

long_plot1 = cum |>
  pivot_longer(cols = starts_with("abnorm")) |>
  drop_na() |>
  ggplot(aes(x = date, y = value, color = name)) +
  geom_point() +
  geom_line() +
  #geom_vline(xintercept = 0,
  #           #linewidth = 1.5,
  #           #alpha = 0.5) +
  scale_x_continuous(breaks = seq(-30, 30, by = 10)) +

```



```

scale_color_discrete(
  limits = c("abnorm_plus", "abnorm_point", "abnorm_minus"),
  labels = c("+1 SE", "Point", "-1 SE")
) +
cowplot::theme_cowplot() +
labs(x = "Month Relative to Split", y = "Cumulative Abnormal Return", color = NULL)

```

d) Now we'll make the same plot but into the future

```

# Calculate cumulative returns for the relevant period
ret_point = cumulative_returns(long_capm_point_df,-2, 250)
ret_plus = cumulative_returns(long_capm_plus_df,-2, 250)
ret_minus = cumulative_returns(long_capm_minus_df,-2, 250)

# Make into one data frame
cum = tibble(
  abnorm_point = ret_point,
  abnorm_plus = ret_plus,
  abnorm_minus = ret_minus,
  date = -2:250
)

long_plot2 = cum |>
  pivot_longer(cols = starts_with("abnorm")) |>
  drop_na() |>
  ggplot(aes(x = date, y = value, color = name)) +
  geom_point() +
  geom_line() +
  #geom_vline(xintercept = 0,
    #linewidth = 1.5,
    #alpha = 0.5) +
  scale_x_continuous(breaks = seq(-2, 250, by = 50)) +
  scale_color_discrete(
    limits = c("abnorm_plus", "abnorm_point", "abnorm_minus"),
    labels = c("-1 SE", "Point", "+1 SE")
  ) +
cowplot::theme_cowplot() +
labs(x = "Month Relative to Split", y = "Cumulative Abnormal Return", color = NULL)

```