

# practicum\_1

Kieran Douglas

## 3) Least Squares Estimates of

### A)

```
# Load data and packages
library(tidyquant)
```

Registered S3 method overwritten by 'quantmod':

```
method          from
as.zoo.data.frame zoo
```

```
-- Attaching core tidyquant packages ----- tidyquant 1.0.11 --
v PerformanceAnalytics 2.0.8      v TTR              0.24.4
v quantmod              0.4.28    v xts               0.14.1
-- Conflicts ----- tidyquant_conflicts() --
x zoo::as.Date()                masks base::as.Date()
x zoo::as.Date.numeric()        masks base::as.Date.numeric()
x PerformanceAnalytics::legend() masks graphics::legend()
x quantmod::summary()           masks base::summary()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.5
v forcats    1.0.0      v stringr    1.5.1
v ggplot2    3.5.1      v tibble     3.2.1
v lubridate  1.9.4      v tidyr      1.3.1
v purrr      1.0.2
-- Conflicts ----- tidyverse_conflicts() --
```

```
x dplyr::filter() masks stats::filter()
x dplyr::first()  masks xts::first()
x dplyr::lag()    masks stats::lag()
x dplyr::last()   masks xts::last()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
library(data.table)
```

Attaching package: 'data.table'

The following objects are masked from 'package:lubridate':

hour, isoweek, mday, minute, month, quarter, second, wday, week,  
yday, year

The following objects are masked from 'package:dplyr':

between, first, last

The following object is masked from 'package:purrr':

transpose

The following objects are masked from 'package:xts':

first, last

The following objects are masked from 'package:zoo':

yearmon, yearqtr

```
library(car)
```

Loading required package: carData

Attaching package: 'car'

The following object is masked from 'package:dplyr':

recode

The following object is masked from 'package:purrr':

some

```
stock <- tq_get(
  c("XOM", "BAC", "^GSPC", "^IRX"),
  get = "stock.prices",
  from = "2010-01-01",
  to = "2019-12-31",
  naa.rm = TRUE
)

# group data by date
stock_clean <- stock %>%
  select(date, symbol, adjusted, volume, open, high, low, close) %>%
  pivot_wider(
    names_from = symbol,
    values_from = c(adjusted, volume, open, high, low, close)
  ) %>%
  na.omit()

# My expectation is that Bank of America will be a riskier asset than Exxon since banks would
# We are using US treasury yield as the risk free rate, s&p500 index as the market rate, EXXON
# calculate returns
stock_clean$irx_return <- (stock_clean$`close`^IRX` - lag(stock_clean$`close`^IRX`)) / lag(stock_clean$`close`^IRX`)
stock_clean$xom_return <- (stock_clean$close_XOM - lag(stock_clean$close_XOM)) / lag(stock_clean$close_XOM)
stock_clean$bac_return <- (stock_clean$close_BAC - lag(stock_clean$close_BAC)) / lag(stock_clean$close_BAC)
stock_clean$gspc_return <- (stock_clean$`close`^GSPC` - lag(stock_clean$`close`^GSPC`)) / lag(stock_clean$`close`^GSPC`)

#Divide your sample in two, the first half spanning January 2010-December 2014, and the second half
from2010to2014 <- stock_clean %>%
  filter(date >= as.Date("2010-01-01") & date <= as.Date("2014-12-31"))

from2015to2019 <- stock_clean %>%
  filter(date>= as.Date("2015-01-01") & date<= as.Date("2019-12-31"))
```

**B)**

```
# Estimating the CAPM model
# start by calculating market-riskfree dif and firm-riskfree dif for xom
from2010to2014$excessreturn_xom <- from2010to2014$xom_return-from2010to2014$irx_return
```

```

from2010to2014$mrpremium <- from2010to2014$gspc_return-from2010to2014$irx_return

from2015to2019$excessreturn_xom <- from2015to2019$xom_return-from2015to2019$irx_return
from2015to2019$mrpremium <- from2015to2019$gspc_return-from2015to2019$irx_return

# run xom model for each year bundle
xom_model1 <- lm(data = from2010to2014, excessreturn_xom~mrpremium)
summary(xom_model1)

```

Call:

```
lm(formula = excessreturn_xom ~ mrpremium, data = from2010to2014)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-0.038727	-0.003741	0.000118	0.004001	0.031858

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.0002530	0.0001966	-1.287	0.198
mrpremium	0.9996397	0.0003828	2611.634	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.006914 on 1254 degrees of freedom

(1 observation deleted due to missingness)

Multiple R-squared: 0.9998, Adjusted R-squared: 0.9998

F-statistic: 6.821e+06 on 1 and 1254 DF, p-value: < 2.2e-16

```

xom_model2 <- lm(data = from2015to2019, excessreturn_xom~mrpremium)
summary(xom_model2)

```

Call:

```
lm(formula = excessreturn_xom ~ mrpremium, data = from2015to2019)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-0.042968	-0.005099	-0.000308	0.005190	0.047702

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.0005414	0.0002638	-2.053	0.0403 *
mrpremium	1.0001004	0.0005262	1900.636	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0093 on 1254 degrees of freedom

Multiple R-squared: 0.9997, Adjusted R-squared: 0.9997

F-statistic: 3.612e+06 on 1 and 1254 DF, p-value: < 2.2e-16

```
# now by calculating market-riskfree dif and firm-riskfree dif for bac
from2010to2014$excessreturn_bac <- from2010to2014$bac_return-from2010to2014$irx_return
from2010to2014$mrpremium <- from2010to2014$gspc_return-from2010to2014$irx_return

from2015to2019$excessreturn_bac <- from2015to2019$bac_return-from2015to2019$irx_return
from2015to2019$mrpremium <- from2015to2019$gspc_return-from2015to2019$irx_return

bac_model1 <- lm(data = from2010to2014, excessreturn_bac~mrpremium)
summary(bac_model1)
```

Call:

lm(formula = excessreturn\_bac ~ mrpremium, data = from2010to2014)

Residuals:

	Min	1Q	Median	3Q	Max
	-0.135995	-0.008491	-0.000793	0.007794	0.120903

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-7.878e-05	5.149e-04	-0.153	0.878
mrpremium	1.001e+00	1.002e-03	998.475	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01811 on 1254 degrees of freedom

(1 observation deleted due to missingness)

Multiple R-squared: 0.9987, Adjusted R-squared: 0.9987

F-statistic: 9.97e+05 on 1 and 1254 DF, p-value: < 2.2e-16

```

bac_model2 <- lm(data = from2015to2019, excessreturn_bac~mrpremium)
summary(bac_model2)

```

Call:

```
lm(formula = excessreturn_bac ~ mrpremium, data = from2015to2019)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.056393	-0.006984	-0.000624	0.006376	0.071586

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.0002645	0.0003413	0.775	0.439
mrpremium	0.9997472	0.0006809	1468.311	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01203 on 1254 degrees of freedom

Multiple R-squared: 0.9994, Adjusted R-squared: 0.9994

F-statistic: 2.156e+06 on 1 and 1254 DF, p-value: < 2.2e-16

```
# It seems like all of my regressions came back with beta coefficients close to what would be
```

## C)

It kind of seems like some of the major dips in premium for the Bank of America happen right around election cycles. I see that pre 2012 and pre 2016 there are some of the larger dips, but they are few and far between.

```

ggplot(data = from2010to2014, mapping = aes(x = date, y = excessreturn_bac, color = mrpremium)) +
  geom_point() +
  geom_smooth(method = lm) +
  theme_light() +
  labs(title = "Risk Premium Over Time For The Bank of America", x = "Date", y = "Risk Premium")

```

```
`geom_smooth()` using formula = 'y ~ x'
```

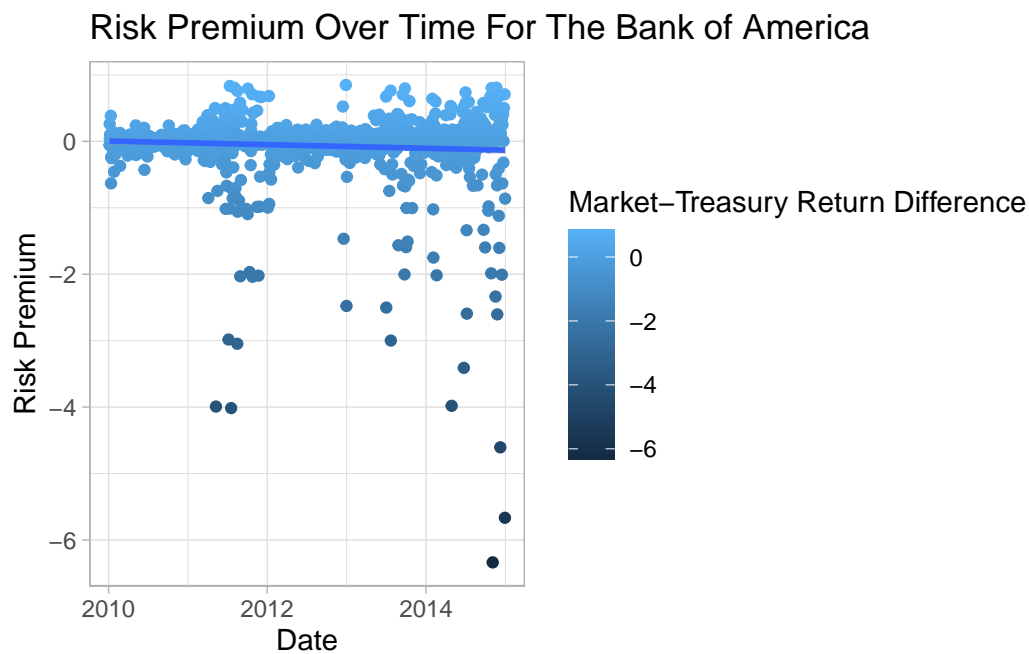
```
Warning: Removed 1 row containing non-finite outside the scale range
(`stat_smooth()`).
```

Warning: The following aesthetics were dropped during statistical transformation:  
colour.

i This can happen when ggplot fails to infer the correct grouping structure in the data.

i Did you forget to specify a `group` aesthetic or to convert a numerical variable into a factor?

Warning: Removed 1 row containing missing values or values outside the scale range (`geom\_point()`).



**D)E)**

Hypothesis testing! In each of these hypothesis tests there is insufficient evidence to conclude that for D) the intercept differs from 0 and for E) the premium coefficient differs from 1. This is to be expected when firms have returns that are in line with market trends.

```
# hpoothesis test that a=0  
linearHypothesis(bac_model1, "(Intercept) = 0")
```

Linear hypothesis test:

(Intercept) = 0

Model 1: restricted model

Model 2: excessreturn\_bac ~ mrpremium

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	1255	0.41107				
2	1254	0.41107	1	7.6731e-06	0.0234	0.8784

```
linearHypothesis(xom_model1, "(Intercept) = 0")
```

Linear hypothesis test:

(Intercept) = 0

Model 1: restricted model

Model 2: excessreturn\_xom ~ mrpremium

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	1255	0.060020				
2	1254	0.059941	1	7.9118e-05	1.6552	0.1985

# the f stat being effectively 0 with a relatively high p value indicates that we cannot reject the null hypothesis. I think that a rejection out the null in this case would probably invalidate the results, as it would imply that the excess return is not zero.

# Using the li

```
linearHypothesis(bac_model1, "mrpremium = 1")
```

Linear hypothesis test:

mrpremium = 1

Model 1: restricted model

Model 2: excessreturn\_bac ~ mrpremium

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	1255	0.41129				
2	1254	0.41107	1	0.00022808	0.6958	0.4044



```
linearHypothesis(xom_model1, "mrpremium = 1")
```

Linear hypothesis test:  
mrpremium = 1

Model 1: restricted model

Model 2: excessreturn\_xom ~ mrpremium

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	1255	0.059983				
2	1254	0.059941	1	4.236e-05	0.8862	0.3467

```
# from this I find that the null hypothesis can only be rejected for the more volatile industries
# For the bank I dont find this surprising since it was shown to not follow market trends. For
```

## F)

The output I received showing proportional market risk makes lots of sense and confirms that the stocks I chose are normal market movers. The market risk proportion is close to 1 for both indicating that almost all of the risk and excess returns from each company can be explained by the more systematic market risk. The market movements explain the volatility.

```
# bank of america prop risk
beta <- coef(bac_model1)
market_var <- var(from2010to2014$mrpremium, na.rm = TRUE)

asset_var_bac <- var(from2010to2014$excessreturn_bac, na.rm = TRUE)

# Proportion of risk attributable to market
proportion_market_risk_bac <- (beta^2 * market_var) / asset_var_bac
print(proportion_market_risk_bac)
```

(Intercept)	mrpremium
6.188243e-09	9.987437e-01

```
# exxon prop risk
beta <- coef(xom_model1)
market_var <- var(from2010to2014$mrpremium, na.rm = TRUE)
```

```
asset_var_xom <- var(from2010to2014$excessreturn_xom, na.rm = TRUE)

# Proportion of risk attributable to market
proportion_market_risk_xom <- (beta^2 * market_var) / asset_var_xom
print(proportion_market_risk_xom)
```

```
(Intercept)    mrpremium
6.402889e-08 9.998162e-01
```

**G)**

In my sample, large estimates of  $\beta$  definitely correspond with higher  $R^2$  values. This makes sense given what we have learned.

#### 4) Is January Different?

**A)**

The market risk premium would be unaffected in this case because there would be no change in compensation for risk to speak of since everything is scaling up. Adding or subtracting the same amount from both of them would leave their difference unchanged. In the case that the risk free and market are affected by the same amount, using CAPM wouldn't make sense because the excess return relation would remain unchanged as stated before. It would probably make more sense to interpret January is different as something that applies to only riskier assets for it to be reasonable to test with the CAPM framework. This is due to the fact that CAPM prices excess returns while in this scenario, the risk premium would be no different than other times of the year, fuzzing any real effect.

**B)**

In this case I think it is clear that the market risk premium would be higher by  $j_m$  because it is specified that only the risk free is not being affected. If  $a$  and  $\beta$  were constant we would have  $r'_p = r_f + \beta(r_m - r_f) + \beta_{jm} = r_p + \beta_{jm}$  which absent  $\beta_{jm}$  is just  $r_f + \beta(r_m - r_f) = r_p$  which is just the same equation as in the typical non-January months. I think this would indicate that unable to individually identify  $j_m$  CAPM wouldn't allow you to distinguish January from the rest of the months.

**C)D)**

```

# download monthly price data for the companies of interest: Apple, Exxon, and Chase (3 diff
library(ggthemes)
library(tidyquant)
library(tidyverse)
library(car)

# Define tickers and date range
tickers <- c("AAPL", "BAC", "XOM")
start_date <- "2010-01-01"
end_date <- "2019-12-31"

# Download monthly prices for each ticker
prices_monthly <- tq_get(
  tickers,
  from = start_date,
  to = end_date,
  get = "stock.prices",
  periodicity = "monthly"
)

prices_monthly <- prices_monthly %>%
  mutate(jan_dummy = ifelse(month(date) == 1, 1, 0)) %>%
  group_by(symbol) %>%
  arrange(date) %>%
  mutate(rp = (adjusted / lag(adjusted)) - 1) %>%
  ungroup()

# Get S&P 500 data and calculate market returns
sp500 <- tq_get("^GSPC", from = start_date, to = end_date, get = "stock.prices", periodicity
  arrange(date) %>%
  mutate(rm = (adjusted / lag(adjusted)) - 1)

# Merge stock and market returns
prices_monthly_joined <- prices_monthly %>%
  left_join(sp500 %>% select(date, rm), by = "date")

# Filter out rows with NA in rp or rm before running regressions
run_model <- function(symbol) {
  data <- prices_monthly_joined %>%
    filter(symbol == symbol, !is.na(rp), !is.na(rm))
  model <- lm(rp ~ rm + jan_dummy, data = data)
  summary(model)
}

```

```
}

# Run and print summaries for each company
cat("AAPL:\n")
```

AAPL:

```
print(run_model("AAPL"))
```

Call:

```
lm(formula = rp ~ rm + jan_dummy, data = data)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-0.207901	-0.034834	-0.004002	0.038635	0.236591

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.002307	0.003450	0.669	0.504
rm	1.206637	0.090195	13.378	<2e-16 ***
jan_dummy	-0.009112	0.012181	-0.748	0.455

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.06076 on 354 degrees of freedom

Multiple R-squared: 0.3358, Adjusted R-squared: 0.332

F-statistic: 89.49 on 2 and 354 DF, p-value: < 2.2e-16

```
cat("\nXOM:\n")
```

XOM:

```
print(run_model("XOM"))
```

Call:

```
lm(formula = rp ~ rm + jan_dummy, data = data)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-0.207901	-0.034834	-0.004002	0.038635	0.236591

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.002307	0.003450	0.669	0.504
rm	1.206637	0.090195	13.378	<2e-16 ***
jan_dummy	-0.009112	0.012181	-0.748	0.455

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.06076 on 354 degrees of freedom

Multiple R-squared: 0.3358, Adjusted R-squared: 0.332

F-statistic: 89.49 on 2 and 354 DF, p-value: < 2.2e-16

```
cat("\nBAC:\n")
```

BAC:

```
print(run_model("BAC"))
```

Call:

```
lm(formula = rp ~ rm + jan_dummy, data = data)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-0.207901	-0.034834	-0.004002	0.038635	0.236591

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.002307	0.003450	0.669	0.504
rm	1.206637	0.090195	13.378	<2e-16 ***
jan_dummy	-0.009112	0.012181	-0.748	0.455

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.06076 on 354 degrees of freedom

Multiple R-squared: 0.3358, Adjusted R-squared: 0.332  
F-statistic: 89.49 on 2 and 354 DF, p-value: < 2.2e-16

```
# In this case the market premium definitely matters but January still does not!
```

**E)**

From my analysis I can conclude that for the companies I looked at and in the time frame of interest, January was not in fact different than the other months of the year. I tested this hypothesis in various ways including a simple jan dummy in addition to running a CAPM with a separate January intercept, none of which turned up results that were statistically significant. January is NOT different.

## 5) COVID Lockdown - Event Studies

**A)**

```
library(tidyquant)
library(lubridate)
library(tidyverse)
# Define tickers and date range
tickers <- c("DRI", "ZM", "PTON", "COST", "WMT", "^GSPC")
start_date <- "2019-09-19"
end_date <- "2022-09-18"

# Download daily stock prices for each ticker
prices_daily <- tq_get(
  tickers,
  from = start_date,
  to = end_date,
  get = "stock.prices",
  periodicity = "daily"
)

# defining date:
event_date <- as.Date("2020-03-19")

prices_daily <- prices_daily %>%
  mutate(event_day = as.numeric(date - event_date))

# calculate % change in prices before and after the event_date
```

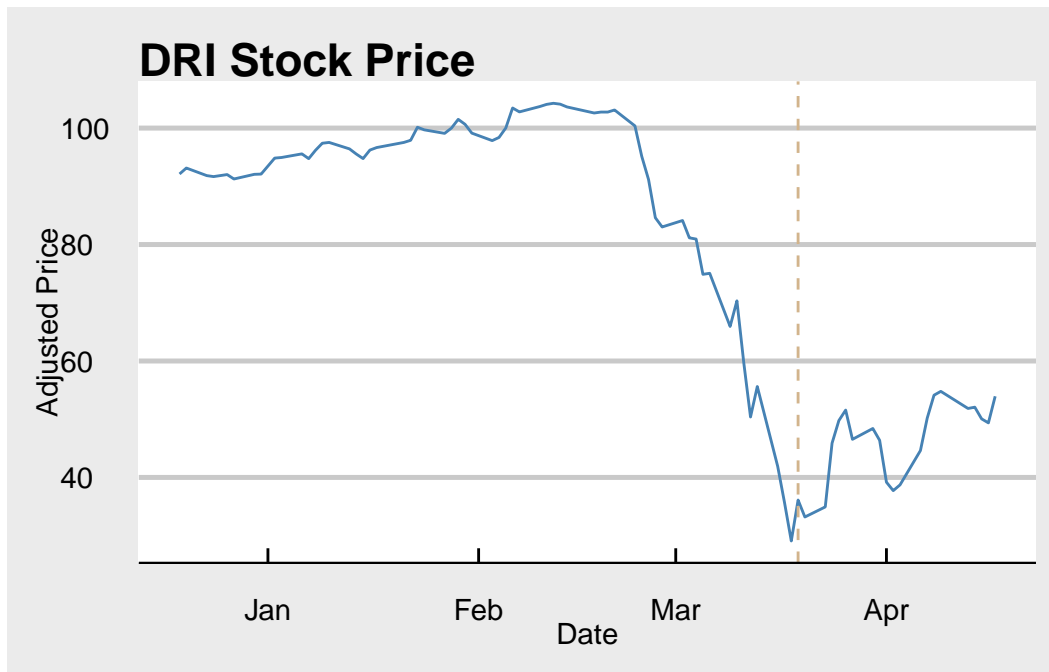
```

before <- prices_daily %>%
  filter(event_day == -30 | event_day == -1) %>%
  group_by(symbol) %>%
  arrange(date) %>%
  summarise(change = last(adjusted) / first(adjusted) - 1)
after <- prices_daily %>%
  filter(event_day == 0 | event_day == 30) %>%
  group_by(symbol) %>%
  arrange(date) %>%
  summarise(change = last(adjusted) / first(adjusted) - 1)

# make df for the plot
to_plot <- prices_daily %>%
  filter(date >= event_date %m-% months(3), date <= event_date %m+% months(1))

# plotting darden adjusted over time
to_plot %>% filter(symbol == "DRI") %>%
  ggplot(aes(x = date, y = adjusted)) +
  geom_line(color = "steelblue") +
  geom_vline(xintercept = as.numeric(event_date), linetype = "dashed", color = "tan") +
  labs(title = "DRI Stock Price",
       x = "Date", y = "Adjusted Price") +
  theme_economist_white()

```



```
to_plot %>% filter(symbol == "ZM") %>%  
  ggplot(aes(x = date, y = adjusted)) +  
  geom_line(color = "steelblue") +  
  geom_vline(xintercept = as.numeric(event_date), linetype = "dashed", color = "tan") +  
  labs(title = "ZM Stock Price",  
        x = "Date", y = "Adjusted Price") +  
  theme_economist_white()
```





```
#daily returns
daily_returns <- prices_daily %>%
  group_by(symbol) %>%
  arrange(date) %>%
  mutate(return = (adjusted / lag(adjusted)) - 1)

est_start <- event_date %m-% months(3)
est_end <- event_date - 1

darden <- daily_returns %>% filter(symbol == "DRI")
zoom <- daily_returns %>% filter(symbol == "ZM")

# Mean return for each
dr_mean <- darden %>%
  filter(date >= est_start, date <= est_end) %>%
  summarise(mu = mean(return, na.rm = TRUE)) %>% pull(mu)
zm_mean <- zoom %>%
  filter(date >= est_start, date <= est_end) %>%
  summarise(mu = mean(return, na.rm = TRUE)) %>% pull(mu)

# Event window (e.g., -5 to +5 days around event)
darden_event <- darden %>%
  filter(event_day >= -5, event_day <= 5) %>%
```

```

    mutate(abnormal = return - dr_mean)
zoom_event <- zoom %>%
  filter(event_day >= -5, event_day <= 5) %>%
  mutate(abnormal = return - zm_mean)

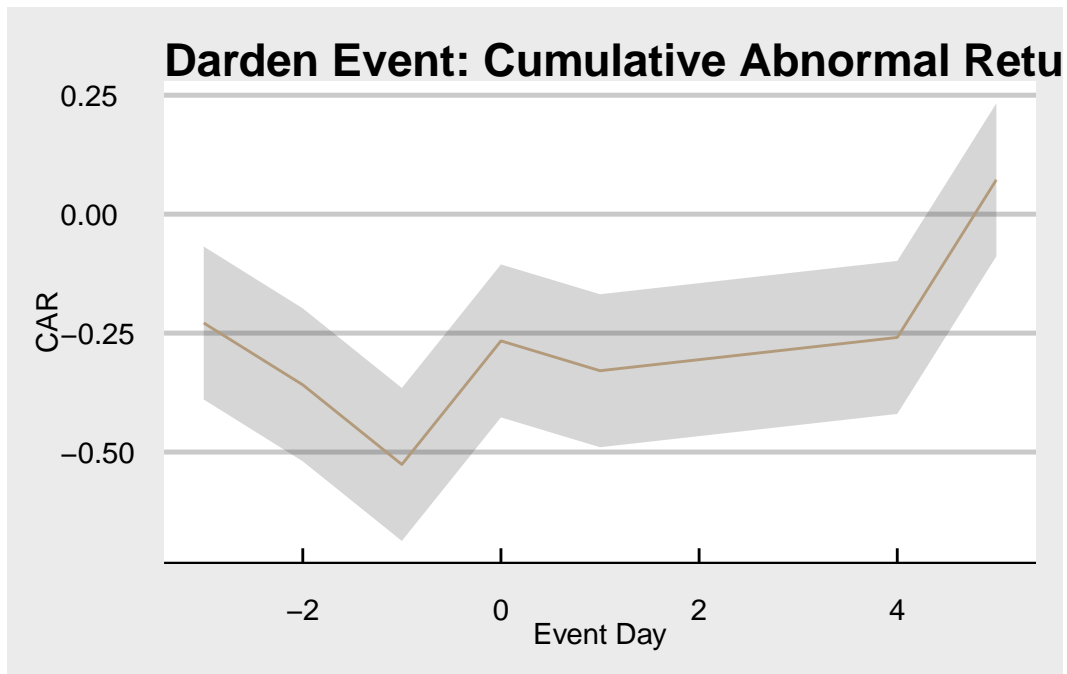
calc_dar <- function(darden_event) {
  darden_event %>%
    mutate(CAR = cumsum(abnormal)) %>%
    mutate(se = sd(abnormal, na.rm=TRUE)/sqrt(n()),
           lower = CAR - 1.96*se, upper = CAR + 1.96*se)
}

calc_zoo <- function(zoom_event) {
  zoom_event %>%
    mutate(CAR = cumsum(abnormal)) %>%
    mutate(se = sd(abnormal, na.rm=TRUE)/sqrt(n()),
           lower = CAR - 1.96*se, upper = CAR + 1.96*se)
}

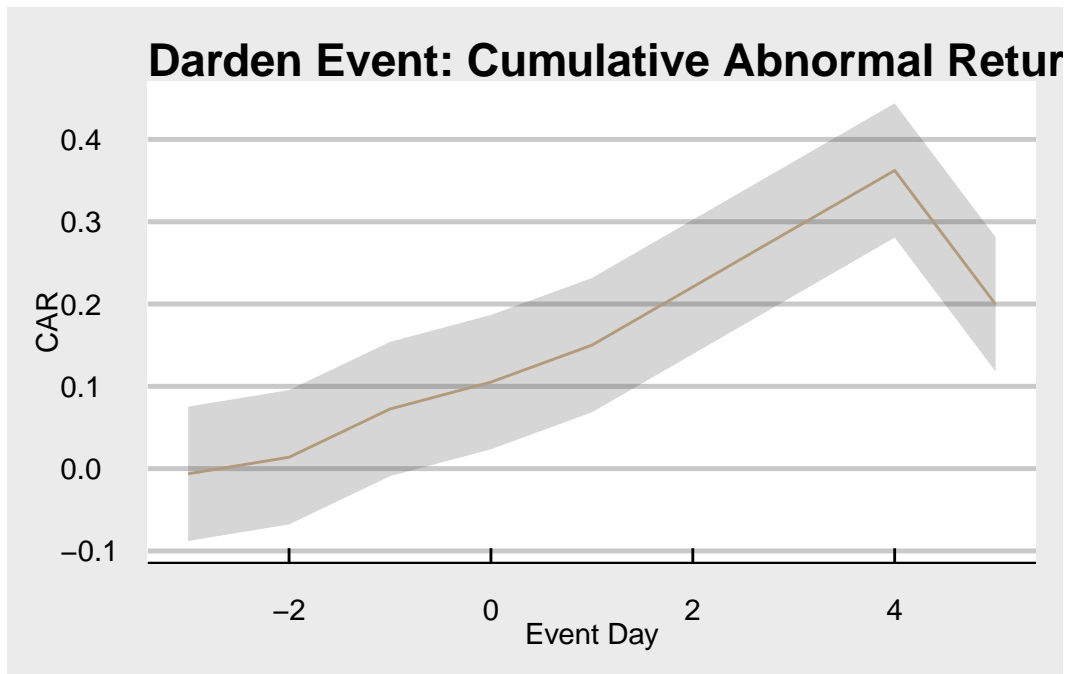
darden_event <- calc_dar(darden_event)
zoom_event <- calc_zoo(zoom_event)

# Plots for both darden and zoom for Cumulative Abnormal Returns
ggplot(darden_event, aes(x = event_day, y = CAR)) +
  geom_line(color = 'tan') +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.2) +
  labs(title="Darden Event: Cumulative Abnormal Returns",
       x = "Event Day", y = "CAR") +
  theme_economist_white()

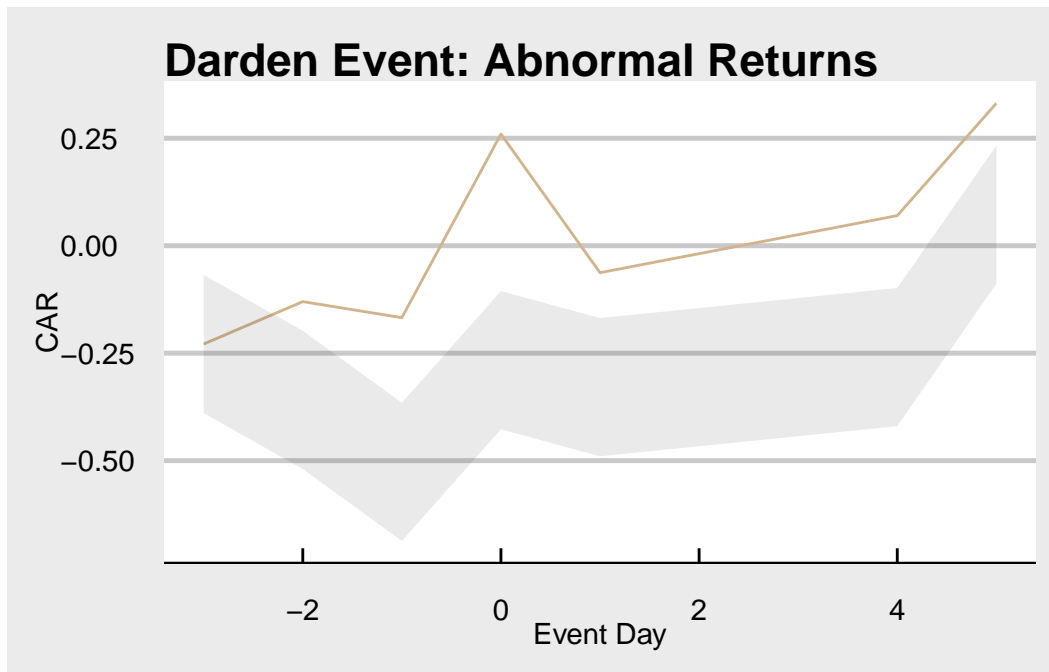
```



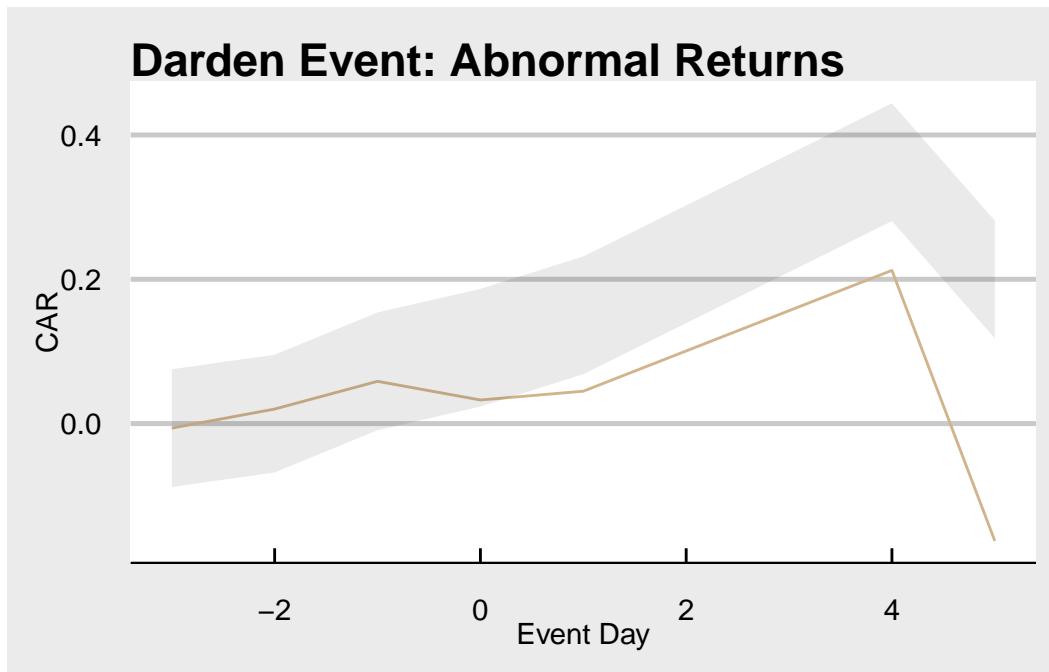
```
ggplot(zoom_event, aes(x = event_day, y = CAR)) +  
  geom_line(color = 'tan') +  
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.2) +  
  labs(title="Darden Event: Cumulative Abnormal Returns",  
        x = "Event Day", y = "CAR") +  
  theme_economist_white()
```



```
ggplot(darden_event, aes(x = event_day, y = abnormal)) +  
  geom_line(color = 'tan') +  
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.1) +  
  labs(title="Darden Event: Abnormal Returns",  
        x = "Event Day", y = "CAR") +  
  theme_economist_white()
```



```
ggplot(zoom_event, aes(x = event_day, y = abnormal)) +
  geom_line(color = 'tan') +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.1) +
  labs(title="Darden Event: Abnormal Returns",
       x = "Event Day", y = "CAR") +
  theme_economist_white()
```



Before the lockdown, prices changes by change COST -4.52% DRI -72.65% PTON -4.76% WMT +2.47% ZM +23.16% ^GSPC -28.85%

The daily return for Darden during the event window was -24.66%, -14.79%, -18.53%, +24.18%, -8%, +5%, and +31.34% while for Zoom it was +0.36%, +3%, +6.89%, +4.26%, +5.48%, +22.22%, and -15.28%.

**B)C)**

```
gspc_df <- daily_returns %>% filter(symbol == "^GSPC") %>% select(date, sp500_return = return
```

Adding missing grouping variables: `symbol`

```
dri <- daily_returns %>% filter(symbol == "DRI") %>% left_join(gspc_df, by = "date")

# Generate the window prior to event
event_date <- as.Date("2020-03-19")
est_start <- event_date %m-% months(1)
est_end <- event_date - 1

dri_est <- dri %>% filter(date >= est_start, date <= est_end)
```

```

# run the market model similar to before
mod <- lm(return ~ sp500_return, data = dri_est)
alpha <- coef(mod)[1]
beta  <- coef(mod)[2]
# a=-0.036 and b= 1.24055

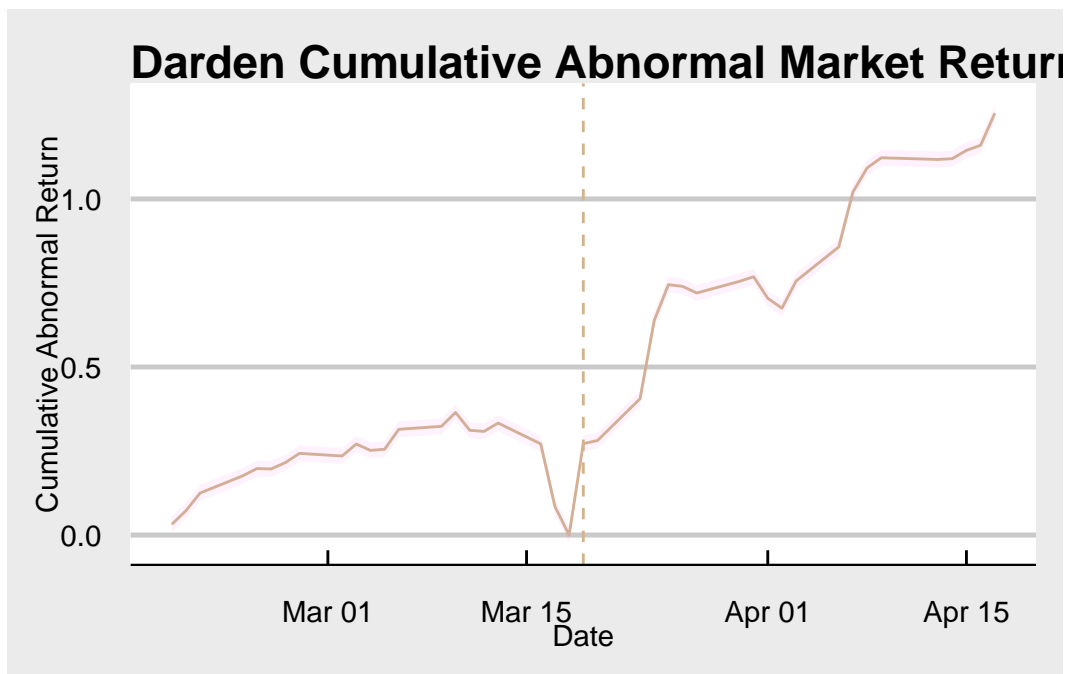
# define window (1 month before/after)
window <- dri %>% filter(date >= event_date %m-% months(1), date <= event_date %m+% months(1))
window <- window %>% mutate(abnormal = return - (alpha + beta * sp500_return))

# grab cumulative abnormal returns
window <- window %>%
  mutate(CAR = cumsum(abnormal))

# generate CI around the cumulative abnormal returns
ar_sd <- sd(window$abnormal, na.rm = TRUE)
n <- nrow(window)
window <- window %>%
  mutate(lower = CAR - 1.96 * ar_sd / sqrt(n),
         upper = CAR + 1.96 * ar_sd / sqrt(n))

# create a graphic of cumulative abnormal returns for the market
ggplot(window, aes(x = date, y = CAR)) +
  geom_line(color = 'tan') +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.1, fill = "violet") +
  geom_vline(xintercept = as.numeric(event_date), linetype = "dashed", color = "tan") +
  labs(title = "Darden Cumulative Abnormal Market Returns", x = "Date", y = "Cumulative Abn")
  theme_economist_white()

```



```
### For "ZM", "PTON", "COST", and "WMT"
```

```
### ZM
```

```
gspc_df <- daily_returns %>% filter(symbol == "^GSPC") %>% select(date, sp500_return = return)
```

Adding missing grouping variables: `symbol`

```
zm <- daily_returns %>% filter(symbol == "ZM") %>% left_join(gspc_df, by = "date")
```

```
# Generate the window prior to event
```

```
event_date <- as.Date("2020-03-19")
```

```
est_start <- event_date %m-% months(1)
```

```
est_end <- event_date - 1
```

```
zm_est <- zm %>% filter(date >= est_start, date <= est_end)
```

```
# run the market model similar to before
```

```
mod <- lm(return ~ sp500_return, data = zm_est)
```

```
alpha <- coef(mod)[1]
```

```
beta <- coef(mod)[2]
```

```
# a=0.00986 and b= -0.0784
```

```
# define window (1 month before/after)
```



```

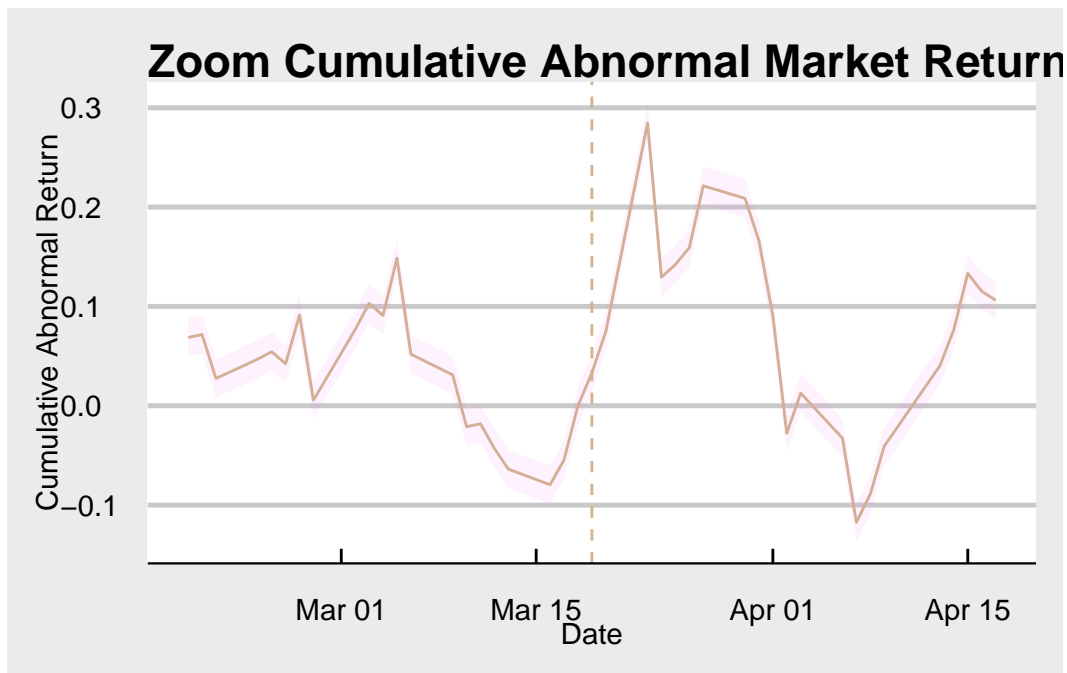
window <- zm %>% filter(date >= event_date %m-% months(1), date <= event_date %m+% months(1))
window <- window %>% mutate(abnormal = return - (alpha + beta * sp500_return))

# grab cumulative abnormal returns
window <- window %>%
  mutate(CAR = cumsum(abnormal))

# generate CI around the cumulative abnormal returns
ar_sd <- sd(window$abnormal, na.rm = TRUE)
n <- nrow(window)
window <- window %>%
  mutate(lower = CAR - 1.96 * ar_sd / sqrt(n),
         upper = CAR + 1.96 * ar_sd / sqrt(n))

# create a graphic of cumulative abnormal returns for the market
ggplot(window, aes(x = date, y = CAR)) +
  geom_line(color = 'tan') +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.1, fill = "violet") +
  geom_vline(xintercept = as.numeric(event_date), linetype = "dashed", color = "tan") +
  labs(title = "Zoom Cumulative Abnormal Market Returns", x = "Date", y = "Cumulative Abnormal Return") +
  theme_economist_white()

```



```
### PTON
```

```
gspc_df <- daily_returns %>% filter(symbol == "^GSPC") %>% select(date, sp500_return = return
```

Adding missing grouping variables: `symbol`

```
pton <- daily_returns %>% filter(symbol == "PTON") %>% left_join(gspc_df, by = "date")
```

```
# Generate the window prior to event
```

```
event_date <- as.Date("2020-03-19")
```

```
est_start <- event_date %m-% months(1)
```

```
est_end <- event_date - 1
```

```
pton_est <- pton %>% filter(date >= est_start, date <= est_end)
```

```
# run the market model similar to before
```

```
mod <- lm(return ~ sp500_return, data = pton_est)
```

```
alpha <- coef(mod)[1]
```

```
beta <- coef(mod)[2]
```

```
# a=0.0041 and b= 0.302
```

```
# define window (1 month before/after)
```

```
window <- pton %>% filter(date >= event_date %m-% months(1), date <= event_date %m+% months(1))
```

```
window <- window %>% mutate(abnormal = return - (alpha + beta * sp500_return))
```

```
# grab cumulative abnormal returns
```

```
window <- window %>%
```

```
  mutate(CAR = cumsum(abnormal))
```

```
# generate CI around the cumulative abnormal returns
```

```
ar_sd <- sd(window$abnormal, na.rm = TRUE)
```

```
n <- nrow(window)
```

```
window <- window %>%
```

```
  mutate(lower = CAR - 1.96 * ar_sd / sqrt(n),
```

```
          upper = CAR + 1.96 * ar_sd / sqrt(n))
```

```
# create a graphic of cumulative abnormal returns for the market
```

```
ggplot(window, aes(x = date, y = CAR)) +
```

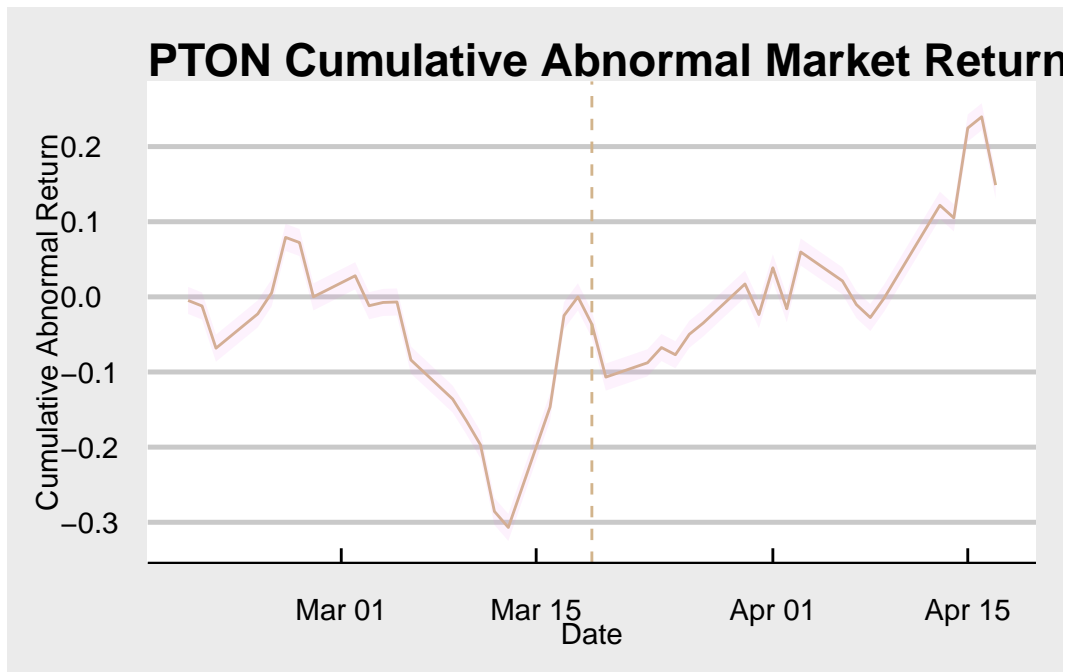
```
  geom_line(color = 'tan') +
```

```
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.1, fill = "violet") +
```

```
  geom_vline(xintercept = as.numeric(event_date), linetype = "dashed", color = "tan") +
```

```
  labs(title = "PTON Cumulative Abnormal Market Returns", x = "Date", y = "Cumulative Abnormal
```

```
  theme_economist_white())
```



```
### COST
```

```
gspc_df <- daily_returns %>% filter(symbol == "^GSPC") %>% select(date, sp500_return = return
```

Adding missing grouping variables: `symbol`

```
cost <- daily_returns %>% filter(symbol == "COST") %>% left_join(gspc_df, by = "date")
```

```
# Generate the window prior to event
```

```
event_date <- as.Date("2020-03-19")
```

```
est_start <- event_date %m-% months(1)
```

```
est_end <- event_date - 1
```

```
cost_est <- cost %>% filter(date >= est_start, date <= est_end)
```

```
# run the market model similar to before
```

```
mod <- lm(return ~ sp500_return, data = cost_est)
```

```
alpha <- coef(mod)[1]
```

```
beta <- coef(mod)[2]
```

```
# a=0.01039 and b= 0.7857
```

```
# define window (1 month before/after)
```

```
window <- cost %>% filter(date >= event_date %m-% months(1), date <= event_date %m+% months(1))
```

```

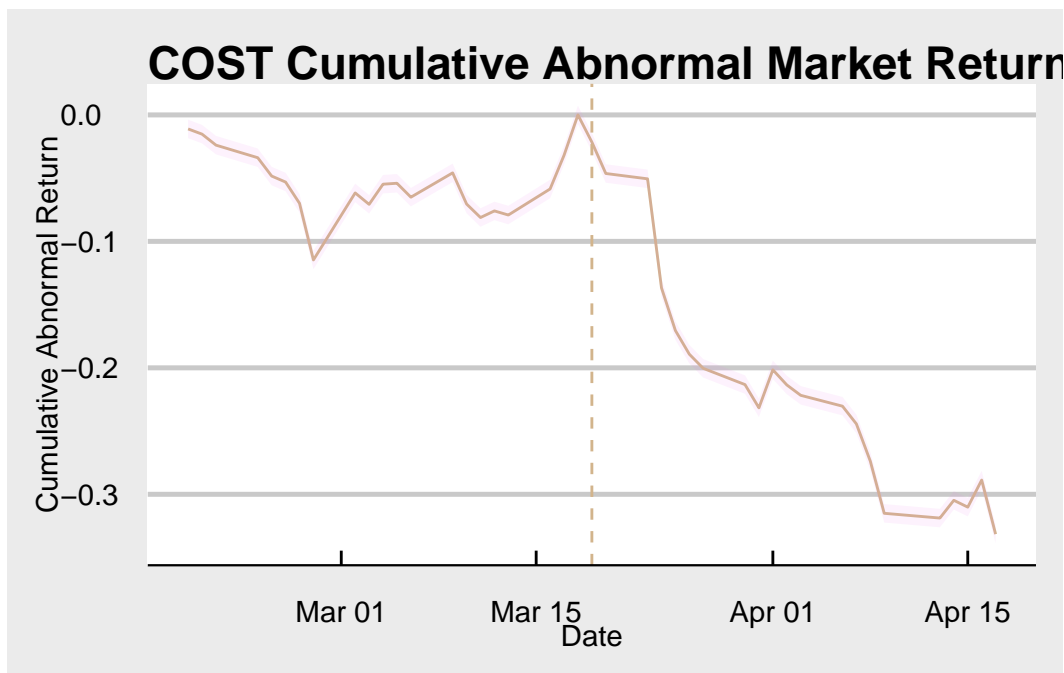
window <- window %>% mutate(abnormal = return - (alpha + beta * sp500_return))

# grab cumulative abnormal returns
window <- window %>%
  mutate(CAR = cumsum(abnormal))

# generate CI around the cumulative abnormal returns
ar_sd <- sd(window$abnormal, na.rm = TRUE)
n <- nrow(window)
window <- window %>%
  mutate(lower = CAR - 1.96 * ar_sd / sqrt(n),
         upper = CAR + 1.96 * ar_sd / sqrt(n))

# create a graphic of cumulative abnormal returns for the market
ggplot(window, aes(x = date, y = CAR)) +
  geom_line(color = 'tan') +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.1, fill = "violet") +
  geom_vline(xintercept = as.numeric(event_date), linetype = "dashed", color = "tan") +
  labs(title = "COST Cumulative Abnormal Market Returns", x = "Date", y = "Cumulative Abnormal Return") +
  theme_economist_white()

```



```
### WMT
```

```
gspc_df <- daily_returns %>% filter(symbol == "^GSPC") %>% select(date, sp500_return = return
```

Adding missing grouping variables: `symbol`

```
wmt <- daily_returns %>% filter(symbol == "WMT") %>% left_join(gspc_df, by = "date")
```

```
# Generate the window prior to event
```

```
event_date <- as.Date("2020-03-19")
```

```
est_start <- event_date %m-% months(1)
```

```
est_end <- event_date - 1
```

```
wmt_est <- wmt %>% filter(date >= est_start, date <= est_end)
```

```
# run the market model similar to before
```

```
mod <- lm(return ~ sp500_return, data = wmt_est)
```

```
alpha <- coef(mod)[1]
```

```
beta <- coef(mod)[2]
```

```
# a=0.0139 and b= 0.7895
```

```
# define window (1 month before/after)
```

```
window <- wmt %>% filter(date >= event_date %m-% months(1), date <= event_date %m+% months(1))
```

```
window <- window %>% mutate(abnormal = return - (alpha + beta * sp500_return))
```

```
# grab cumulative abnormal returns
```

```
window <- window %>%
```

```
  mutate(CAR = cumsum(abnormal))
```

```
# generate CI around the cumulative abnormal returns
```

```
ar_sd <- sd(window$abnormal, na.rm = TRUE)
```

```
n <- nrow(window)
```

```
window <- window %>%
```

```
  mutate(lower = CAR - 1.96 * ar_sd / sqrt(n),
```

```
          upper = CAR + 1.96 * ar_sd / sqrt(n))
```

```
# create a graphic of cumulative abnormal returns for the market
```

```
ggplot(window, aes(x = date, y = CAR)) +
```

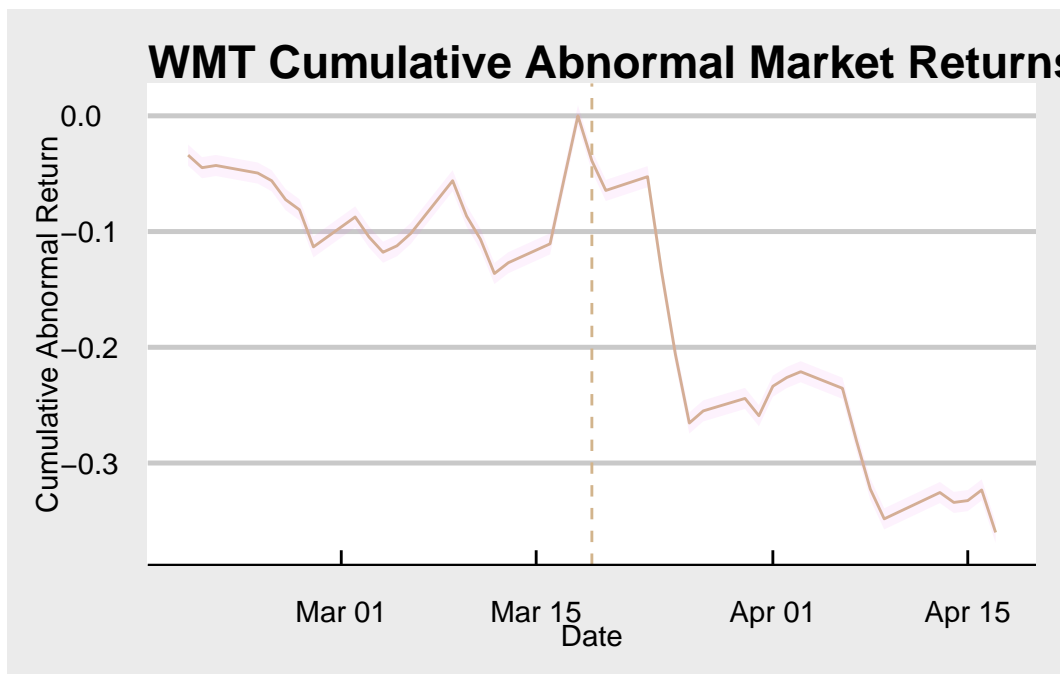
```
  geom_line(color = 'tan') +
```

```
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.1, fill = "violet") +
```

```
  geom_vline(xintercept = as.numeric(event_date), linetype = "dashed", color = "tan") +
```

```
  labs(title = "WMT Cumulative Abnormal Market Returns", x = "Date", y = "Cumulative Abnormal
```

```
  theme_economist_white())
```



I have successfully computed and plotted all of the CAR using a market model for each stock over the specified event window. Surprisingly, Darden spiked a bit in a period where, given that it's industry, would be unexpected. After using the market model rather than just price changes, the outcome does change and the event seems to have a significant impact on Darden. This effect however, is positive??? This is odd.

It seems like whether the market sees the lockdown as a good or a bad thing depends on the industry. WTM and COST have decreasing returns on average during the post lockdown period, suggesting that grocery stores see a decrease. This feels counterintuitive to me as since more people are staying home I would expect them to eat out less. Perhaps it could be that people are ordering more food now rather than going to buy things to avoid going out. Darden and Peloton see positive returns on average in the post period, which in some ways backs my argument of people ordering takeout more often rather than going to grocery stores. I think the PTON could be explained by the fact that they focus on "connective fitness" which is largely remote (makes sense if people stay home). The "Zoom Boom" makes a lot of sense intuitively, as people would now be more likely to meet remotely since they aren't leaving their homes. I think the market viewed the effect of lockdowns on different industries differently because different industries could benefit more or less from people's tendency to leave their homes.