Technical Analysis of US Listed China Technology Stocks

Austin McGhee

August 2021

This dataset uses csv files I have downloaded via free API sources and uploaded to Kaggle. If you would like to know how to download the data yourself, then I recommend viewing the full Rmarkdown code on Github

Packages

The following packages are used.

- library(tidyverse)
- library(tidyquant) <- financial data
- library(lubridate) <- dates
- library(knitr) <- rmarkdown render
- library(ggrepel) <- chart labels
- library(scales) <- dates
- library(bdscale) <- dates

Load in the Data

The csv files used to import for this analysis were generated used AlphaVantage and Yahoo finance data. The data is downloaded using a seperate R script, then the dataframes are exported to csv files which are loaded into this script.

```
# Key lists and dates we will use later.
index <- c("kweb", "mchi")
key.tech <- c("tcehy", "baba", "jd")
exclude.bidu.ntes <- c("bidu", "ntes", "mpngy")
chart.date <- as.Date("2021-01-01") # charting this date

## Load in the On Balance Volume Data
df.OBV <- read_csv("df.OBV.csv")
## Load in the Commodity Channel index data
df.CCI <- read_csv("df.CCI.csv")
## Load in the Chaikin AD data
df.AD <- read_csv("df.AD.csv")
## Load in the Relative Strength index data
df.RSI <- read_csv("df.RSI.csv")
### Stock Data from Yahoo
stock.data <- read_csv("stock.list.csv")
##HK Hong Kong Stock Data from Yahoo
stock.data.hk <- read_csv("stock.list.hk.csv")
## Portfolio daily returns
```

```
stock.list.2 <- read_csv("stock_data_2.csv")

# Date order for plots later.
time <- stock.data$date %>%
  sort(descreasing = TRUE)
```

Price Data

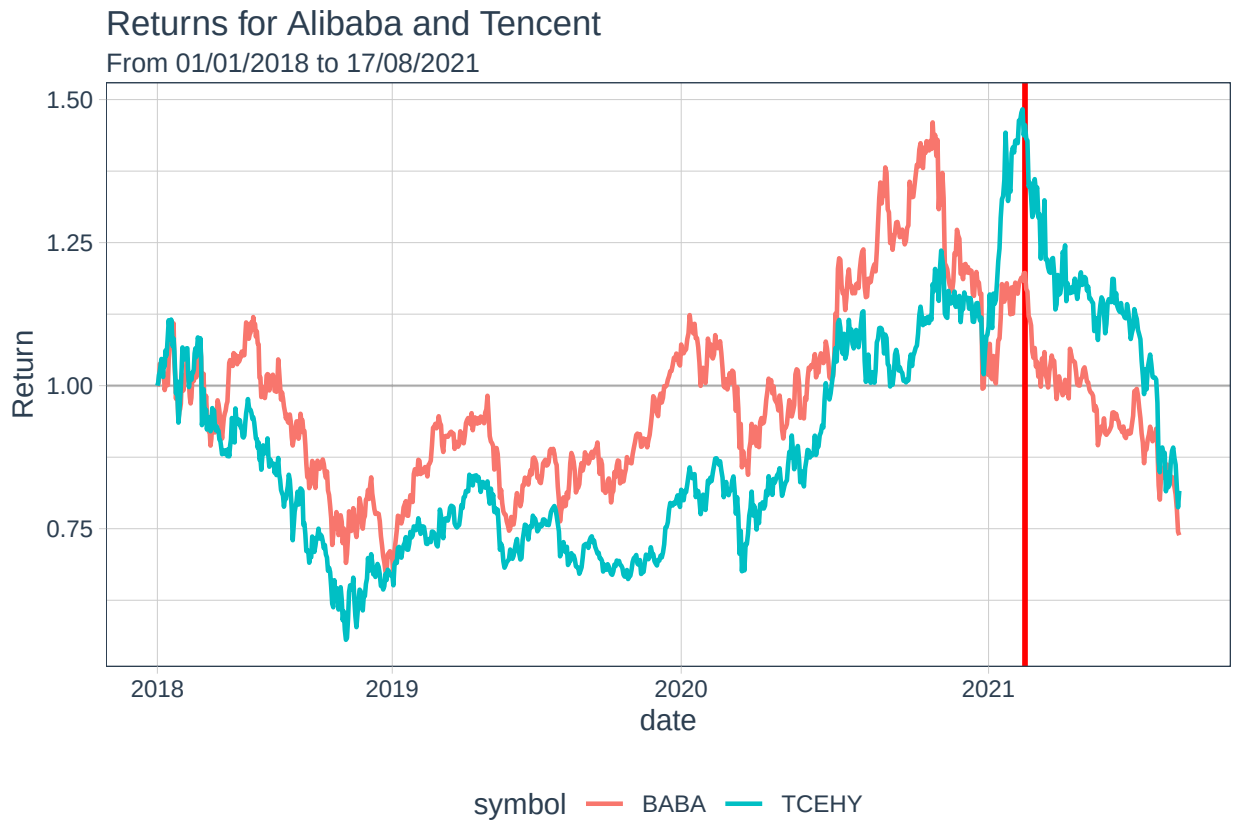
Tidyquant Analytics

Using the Tidyquant package to look at stock prices and returns for the China basket.

```
# Load in stock price dataframe and rename because we will modify.
sl3 <- stock.data %>%
  group_by(symbol)

# More lists
eq.only <- c( "SPY", "MCHI", "KWEB")
temp.list <- c("BABA", "TCEHY")

# Filter, convert prices to daily returns for index.
sl3 %>%
  filter(symbol %in% eq.only) %>%
  tq_transmute(adjusted,
               periodReturn,
               period = "daily",
               type = "log",
               col_rename = "returns") %>%
  mutate(index = 1 * cumprod(1 + returns)) %>%
  ggplot(aes(x = date, y = index, color = symbol)) +
    geom_vline(xintercept = as.Date("2021-02-17"), color = "red") +
    geom_line(size = 0.8) +
  labs(title = "Index Returns for S&P 500, China Broad Market, and China Internet Sector",
       subtitle = "From 01/01/2018 to 17/08/2021",
       caption = "Red line indicating max price of China Internet ETF KWEB (Top of market).",
       x = "Date", y = "Return") +
    theme_tq() +
  scale_x_bd(business.dates=time, labels = date_format(format = "%Y"), max.major.breaks=12)
```

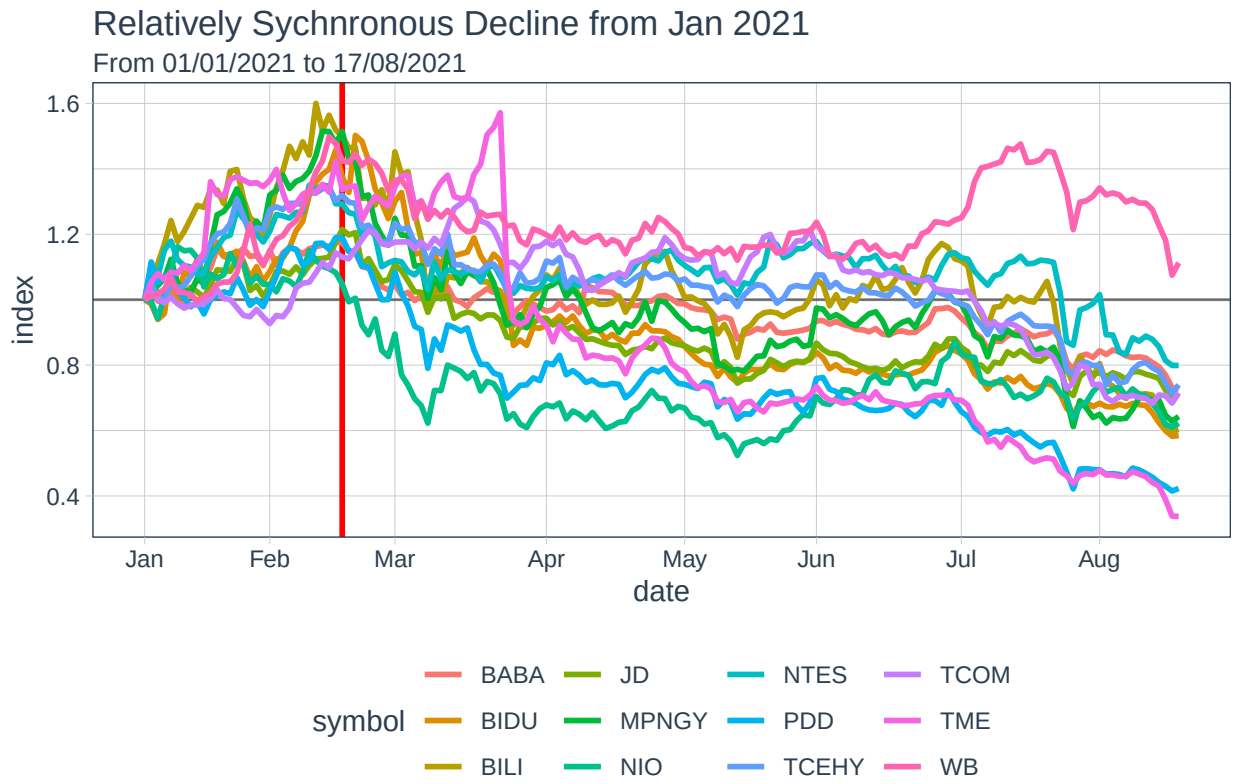
Red line indicating max price of China Internet ETF KWEB (Top of market).

```
# Filter, convert prices to daily for BABA, TCEHY.
sl3 %>%
  filter(symbol %in% temp.list) %>%
  tq_transmute(adjusted,
               periodReturn,
               period = "daily",
               type = "log",
               col_rename = "returns") %>%
  mutate(index = 1 * cumprod(1 + returns)) %>%
  ggplot(aes(x = date, y = index, color = symbol)) +
    geom_hline(yintercept = 1, alpha = 0.2) +
    geom_vline(xintercept = as.Date("2021-02-17"), color = "red", size = 1) +
    geom_line(size = 0.8) +
  labs(title = "Returns for Alibaba and Tencent",
       subtitle = "From 01/01/2018 to 17/08/2021",
       x = "Date", y = "Return") +
  theme_tq() +
  scale_x_bd(business.dates=time, labels = date_format(format = "%Y"), max.major.breaks=12)
```



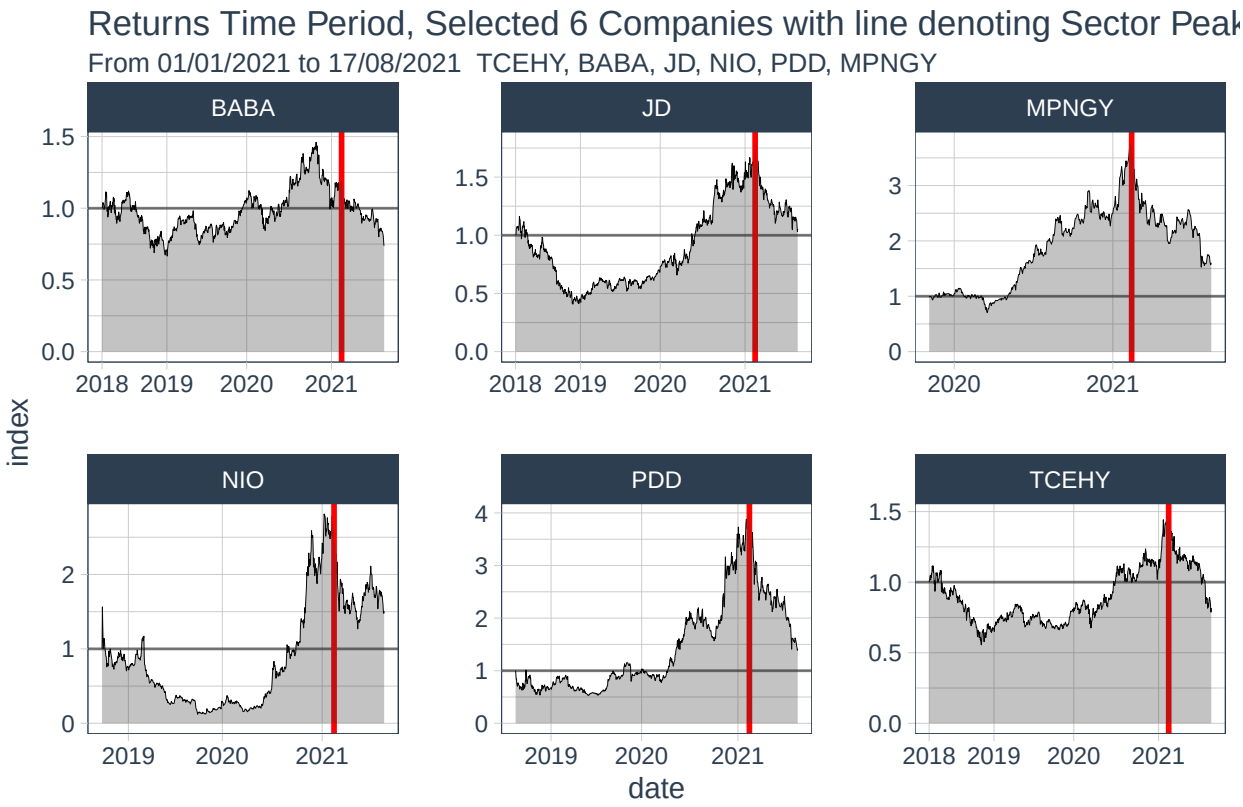
Looking at the overall decline of stocks in the China basket. Most stocks participated in the decline.

```
# Index list
eq.only <- c("SPY", "KWEB", "MCHI")

# Filter, convert prices to daily for non-index.
sl3 %>%
  filter(date > "2021-01-01") %>%
  filter(!symbol %in% eq.only) %>%
  tq_transmute(adjusted,
               periodReturn,
               period = "daily",
               type = "log",
               col_rename = "returns") %>%
  mutate(index = 1 * cumprod(1 + returns)) %>%
  ggplot(aes(x = date, y = index, color = symbol)) +
    geom_hline(yintercept = 1, alpha = 0.5) +
    geom_vline(xintercept = as.Date("2021-02-17"), color = "red", size = 1) +
    geom_line(size = 1) +
  labs(title = "Relatively Sychnronous Decline from Jan 2021",
       subtitle = "From 01/01/2021 to 17/08/2021",
       caption = "Red line indicating max price of China Internet ETF KWEB (Top of market).") +
  theme_tq() +
  scale_x_bd(business.dates=time, labels = date_format(format = "%b"), max.major.breaks=12)
```

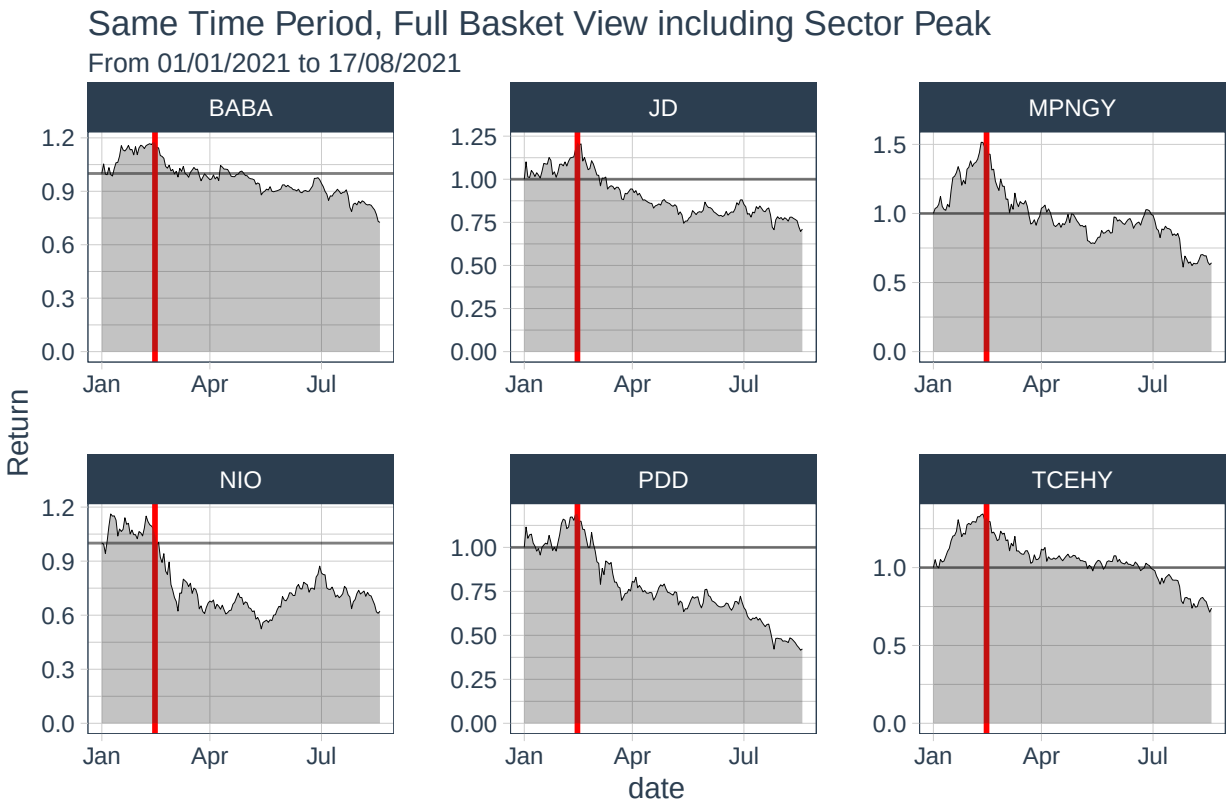


Next we look at the full 6 selected stocks (Tencent, Alibaba, JD, Meituan, Pinduoduo, Nio). These charts measure the stock return over time.

```
# Selecting 6 top stocks
top5 <- c("TCEHY", "BABA", "JD", "MPNGY", "PDD", "NIO")

# Filter, returns for top companies.  Facet wrap.
sl3 %>%
  filter(symbol %in% top5) %>%
  tq_transmute(adjusted,
               periodReturn,
               period = "daily",
               type = "log",
               col_rename = "returns") %>%
  mutate(index = 1 * cumprod(1 + returns)) %>%
  ggplot(aes(x = date, y = index)) +
  geom_hline(yintercept = 1, alpha = 0.5) +
  geom_line(size = 0.1) +
  geom_vline(xintercept = as.Date("2021-02-17"), color = "red", size = 1) +
  geom_area(alpha = 0.3) +
  labs(title = "Returns Time Period, Selected 6 Companies with line denoting Sector Peak",
       subtitle = "From 01/01/2021 to 17/08/2021  TCEHY, BABA, JD, NIO, PDD, MPNGY",
       caption = "Red line indicating max price of China Internet ETF KWEB (Top of market).") +
  scale_x_bd(business.dates=time, labels = date_format(format = "%Y"), max.major.breaks=6) +
  theme_tq() +
  facet_wrap(.~ symbol, scales = "free")
```




```
# Filter, returns for top companies.  Facet wrap.  After Jan 2021.
sl3 %>%
  filter(date > "2021-01-01") %>%
  filter(symbol %in% top5) %>%
  tq_transmute(adjusted,
               periodReturn,
               period = "daily",
               type = "log",
               col_rename = "returns") %>%
  mutate(index = 1 * cumprod(1 + returns)) %>%
  ggplot(aes(x = date, y = index)) +
    geom_hline(yintercept = 1, alpha = 0.5) +
  geom_line(size = 0.1) +
    geom_vline(xintercept = as.Date("2021-02-17"), color = "red", size = 1) +
    geom_area(alpha = 0.3) +
  labs(title = "Same Time Period, Full Basket View including Sector Peak",
       subtitle = "From 01/01/2021 to 17/08/2021",
       caption = "Red line indicating max price of China Internet ETF KWEB (Top of market).",
       y = "Return") +
  scale_x_bd(business.dates=time, labels = date_format(format = "%b"), max.major.breaks=6) +
    theme_tq() +
    facet_wrap(.~ symbol, scales = "free")
```



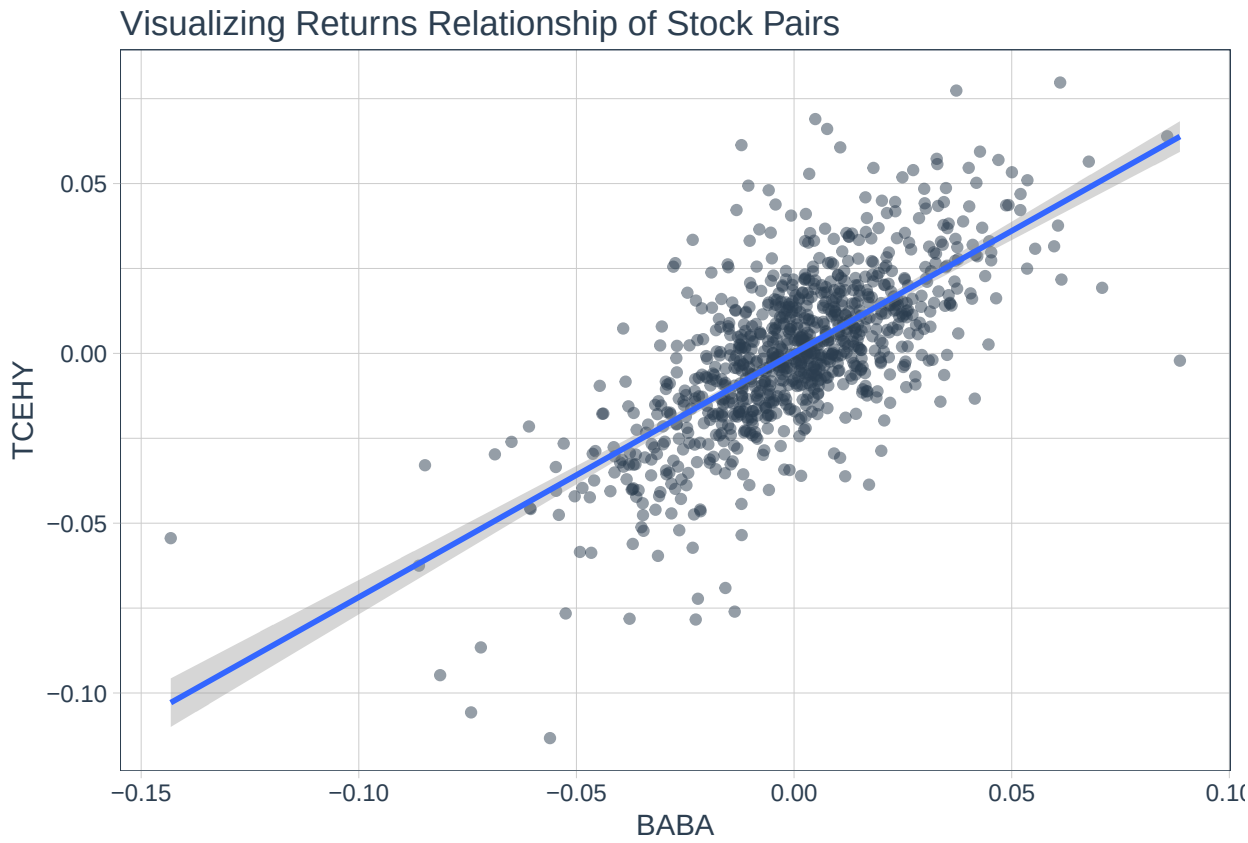
Red line indicating max price of China Internet ETF KWEB (Top of market).

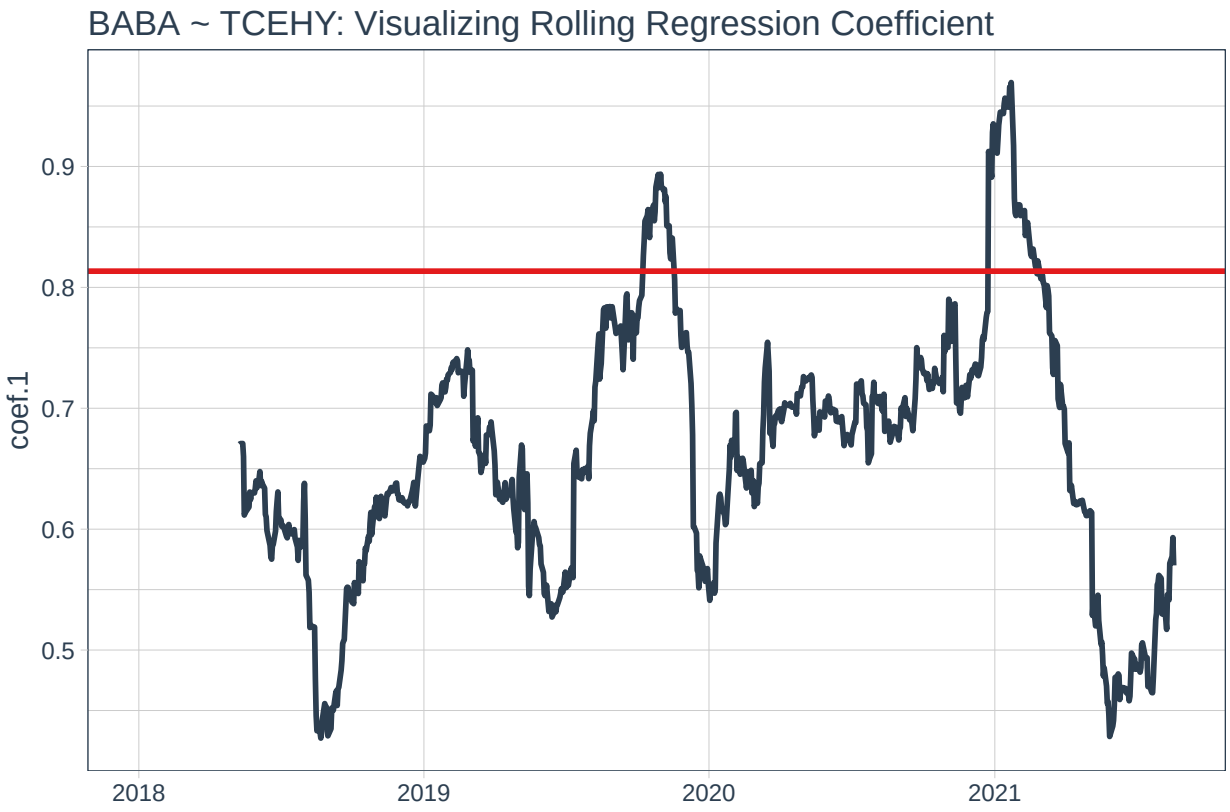
Pairs Analysis

Calculating Return Coefficient and stock return for Alibaba and Tencent.

```
## Pairs Analysis with TidyQuant
# Alibaba and Tencent pairs
temp.list <- c("BABA", "TCEHY")

regr_fun <- function(data) {
    coef(lm(BABA ~ TCEHY, data = timetk::tk_tbl(data, silent = TRUE)))
}

stock_pairs <- sl3 %>%
  filter(symbol %in% temp.list) %>%
  tq_transmute(select     = adjusted,
               mutate_fun = periodReturn,
               period     = "daily",
               type       = "log",
               col_rename = "returns") %>%
  spread(key = symbol, value = returns)
```

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


Visualizing Returns Relationship of Stock Pairs
TCEHY
BABA
0.05
0.00
−0.05
−0.10
−0.15
−0.10
−0.05
0.00
0.05

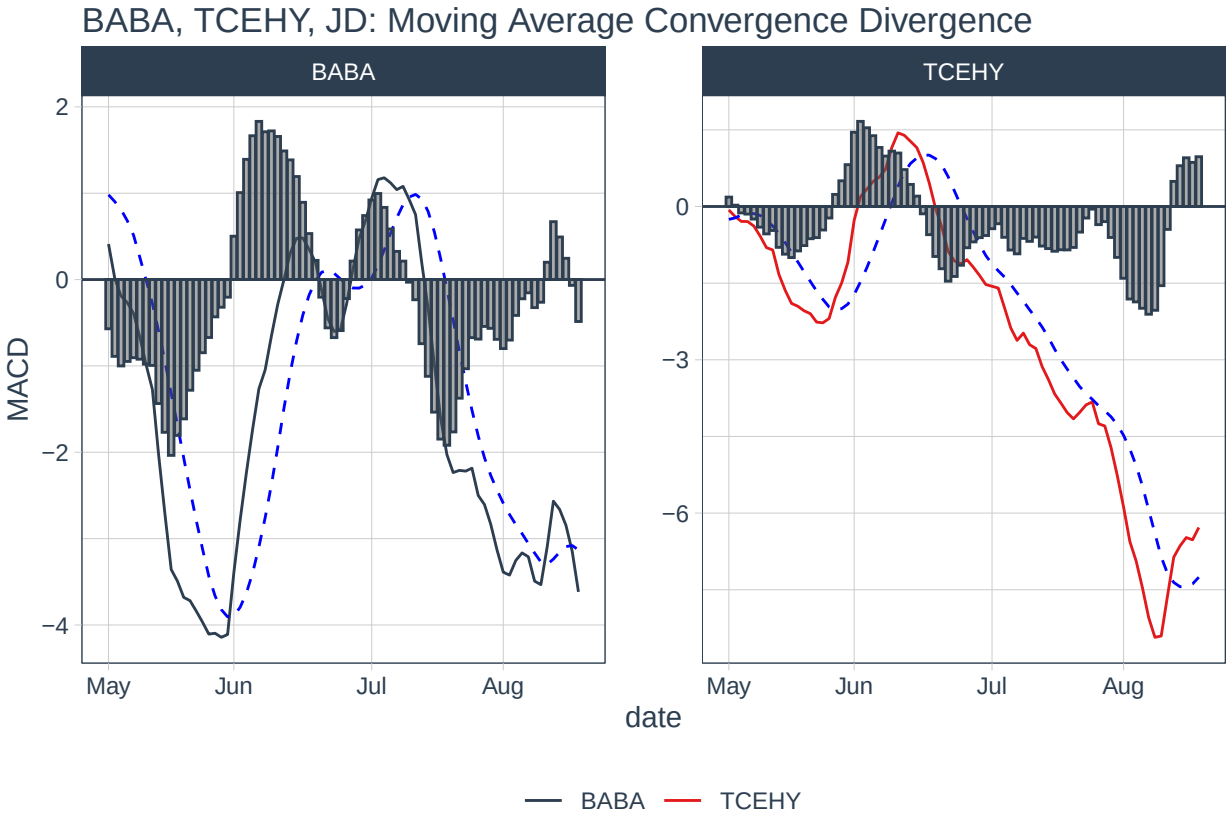


MACD

Now we will plot MACD for select stocks.

```
# Now plotting MACD
main <- c("BABA", "TCEHY")

sl3_macd <- sl3 %>%
  filter(symbol %in% main) %>%
  group_by(symbol) %>%
  tq_mutate(select     = adjusted,
            mutate_fun = MACD,
            nFast      = 12,
            nSlow      = 26,
            nSig       = 9,
            maType     = SMA) %>%
  mutate(diff = macd - signal)

main <- c("BABA", "TCEHY", "JD", "MPNGY")

sl3_macd %>%
  filter(symbol %in% main) %>%
  filter(date > as_date("2021-05-01")) %>%
  ggplot(aes(x = date)) +
  geom_hline(yintercept = 0, color = palette_light()[[1]]) +
```

```
  geom_line(aes(y = macd, col = symbol)) +
  geom_line(aes(y = signal), color = "blue", linetype = 2) +
  geom_bar(aes(y = diff), stat = "identity", alpha = 0.5, color = palette_light()[[1]]) +
  facet_wrap(~ symbol, ncol = 2, scale = "free") +
  labs(title = "BABA, TCEHY, JD: Moving Average Convergence Divergence",
       y = "MACD", x = "", color = "") +
  scale_x_bd(business.dates=time, labels = date_format(format = "%b"), max.major.breaks=12) +
  theme_tq() +
  scale_color_tq()
```



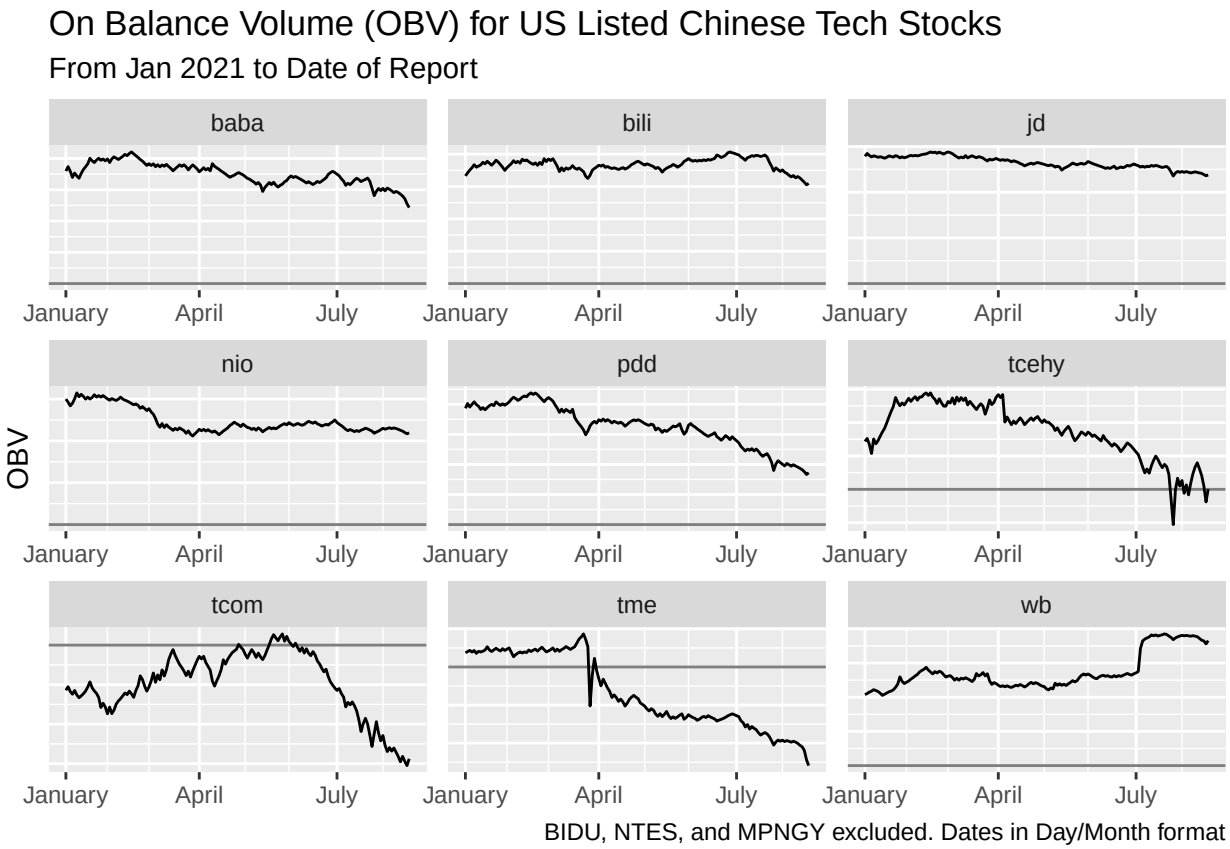
Indicators

On Balance Volume

Abbreviated OBV, running calculations for the China Group. Also individually for Alibaba, Tencent, and JD. We are excluding bidu and ntes by applying the exclude.bidu.ntes list we setup in the step above.

```
# Charting the On Balance Volume
# Facet wrap for all symbols
df.OBV %>%
  filter(time > chart.date) %>%
  filter(!symbol %in% index) %>%
  filter(!symbol %in% exclude.bidu.ntes) %>%
  ggplot(aes(time, OBV)) +
```

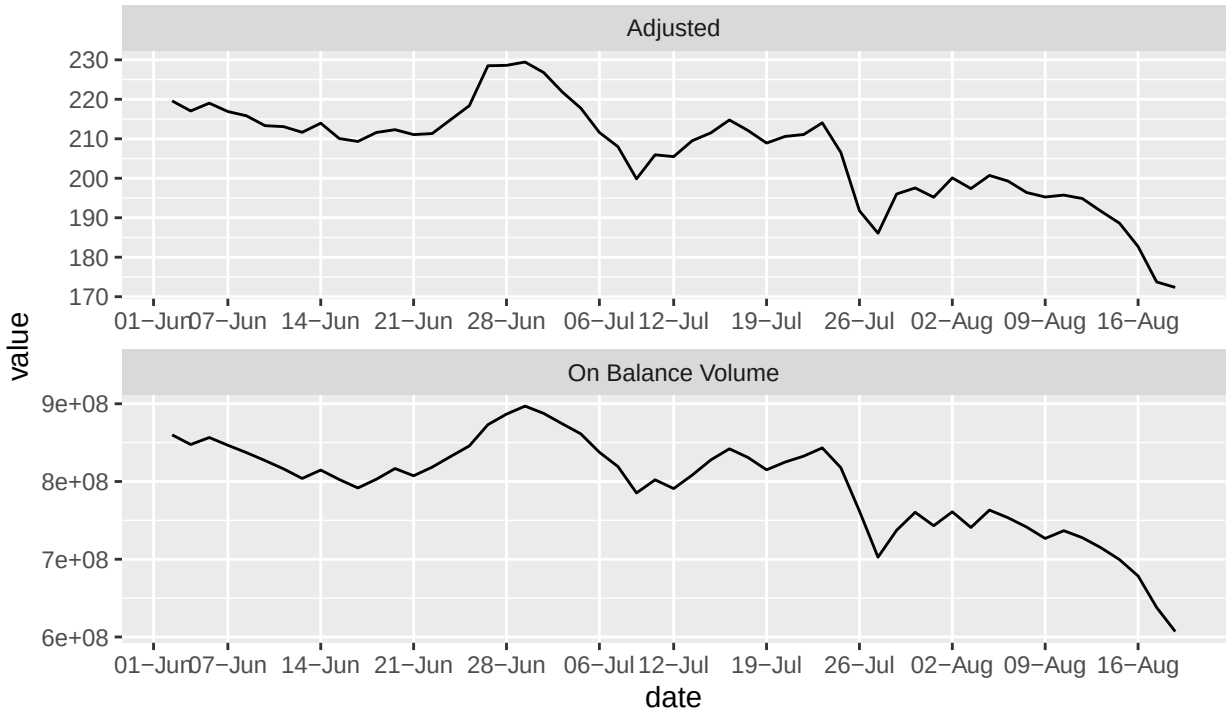
```
  geom_line() +
  geom_hline(yintercept = 0, alpha = 0.5, color = "black") +
  labs(title = "On Balance Volume (OBV) for US Listed Chinese Tech Stocks",
       subtitle = "From Jan 2021 to Date of Report",
       caption = "BIDU, NTES, and MPNGY excluded. Dates in Day/Month format",
       y= "OBV", x = "") +
  scale_x_bd(business.dates=time, labels = date_format(format = "%B"),max.major.breaks=6) +
  theme(axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        axis.title.x=element_blank()) +
  facet_wrap(. ~ symbol, scale = "free")
```



Now combining the Alibaba adjusted daily close data with the OBV data to compare in a graph.

Alibaba Adjusted Daily Price Compared to On Balance Volume

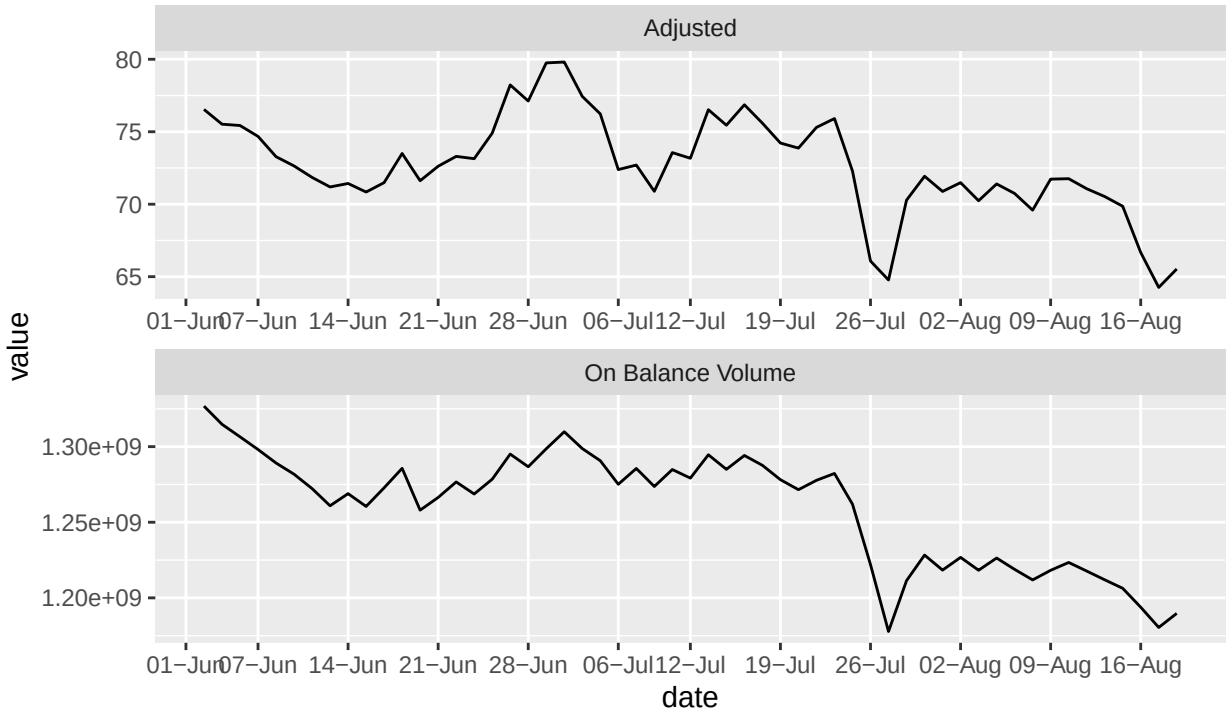
From June 2021 to Present



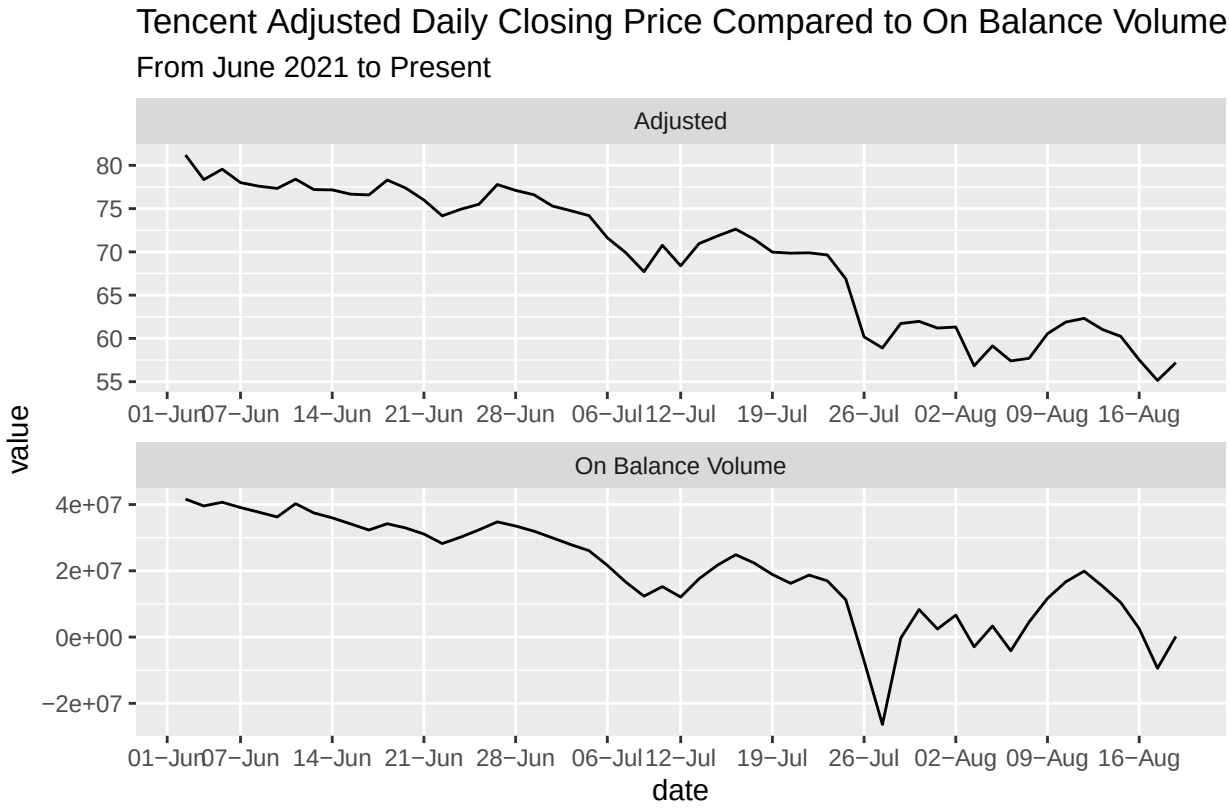
Look for divergence between OBV and Price.

JD Adjusted Daily Closing Price Compared to On Balance Volume

From June 2021 to Present



Look for divergence between OBV and Price.

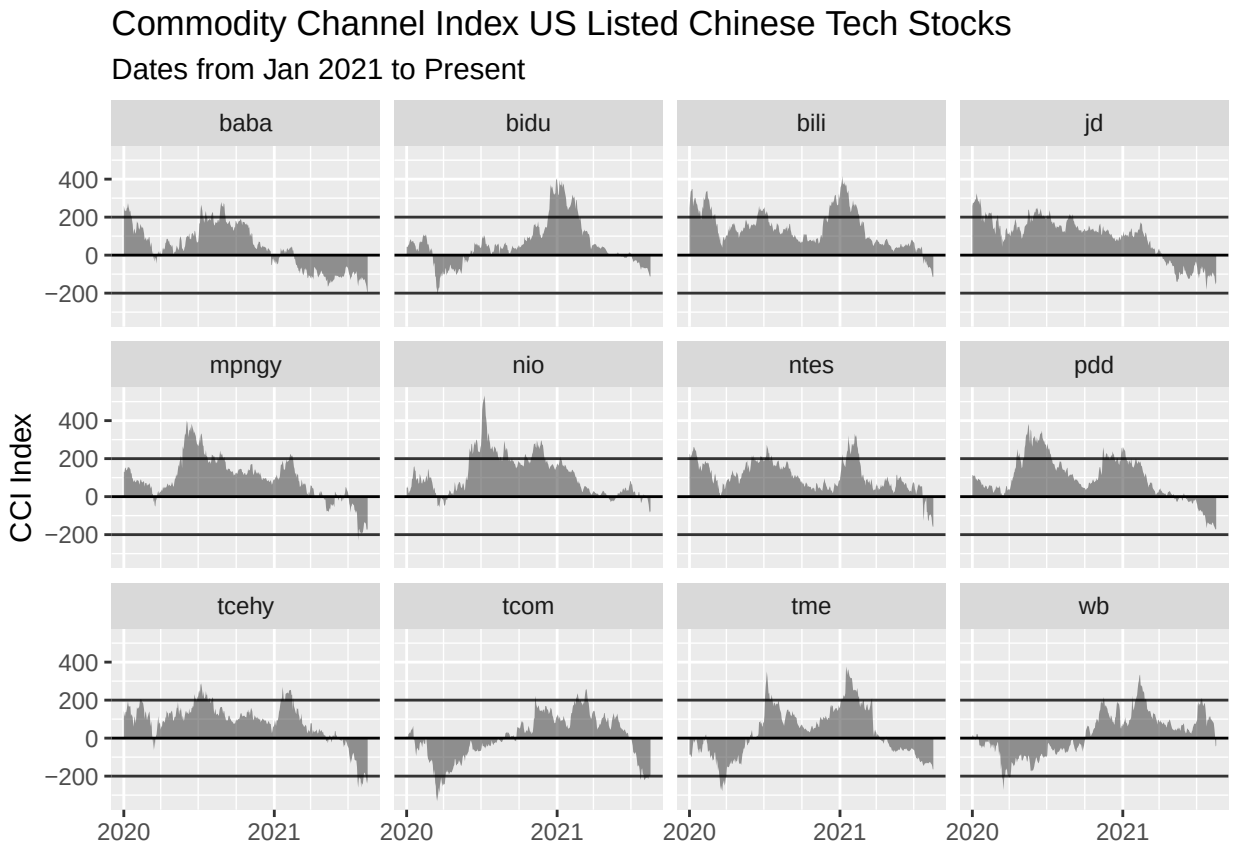


Look for divergence between OBV and Price.

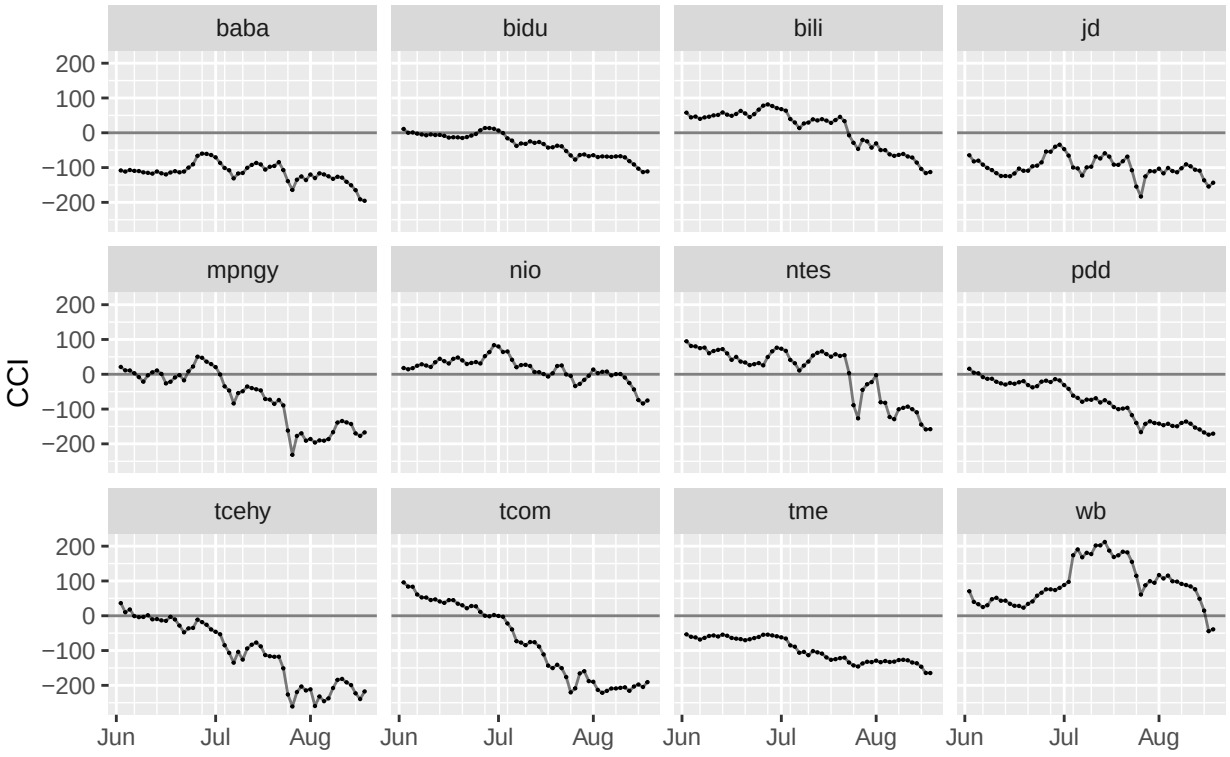
Commodity Channel Index

The CCI is a useful guage that is indexed. Calculating the mean and median CSI of a group of stocks can be an accurate indicator to the groups relative strength or weakness. We remove the index stock and do several calculations to come up with unique sets to use for performance analysis.

```
## Commodity Channel Index (CCI)
# Facet wrap to show a grid of symbols using area charts.
df.CCI %>%
  filter(!symbol %in% index) %>%
  ggplot(aes(time, CCI)) +
  geom_area( alpha = 0.5, show.legend = FALSE) +
  geom_hline(yintercept = 0, color="black") +
  geom_hline(yintercept = -200, color="black", alpha = 0.8) +
  geom_hline(yintercept = 200, color="black", alpha = 0.8) +
  labs(title = "Commodity Channel Index US Listed Chinese Tech Stocks",
       subtitle = "Dates from Jan 2021 to Present",
       y= "CCI Index", x = "Date") +
  scale_x_bd(business.dates=time, labels = date_format(format = "%Y"), max.major.breaks=4) +
  theme(axis.title.x=element_blank()) +
  facet_wrap(. ~ symbol)
```

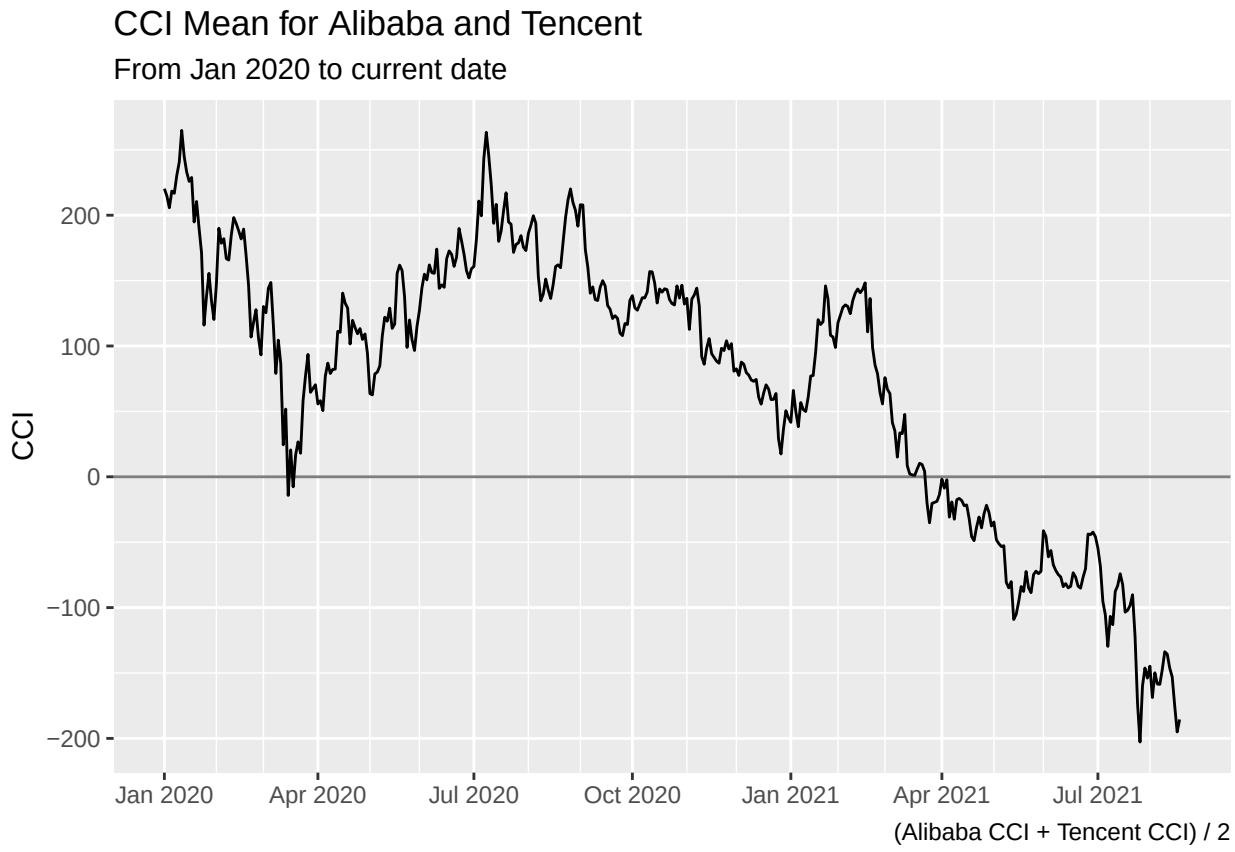


Similiar chart to the last one facet_wrap but zoomed in on dates to June 2021 and after. Switched to combination of line and point charts.

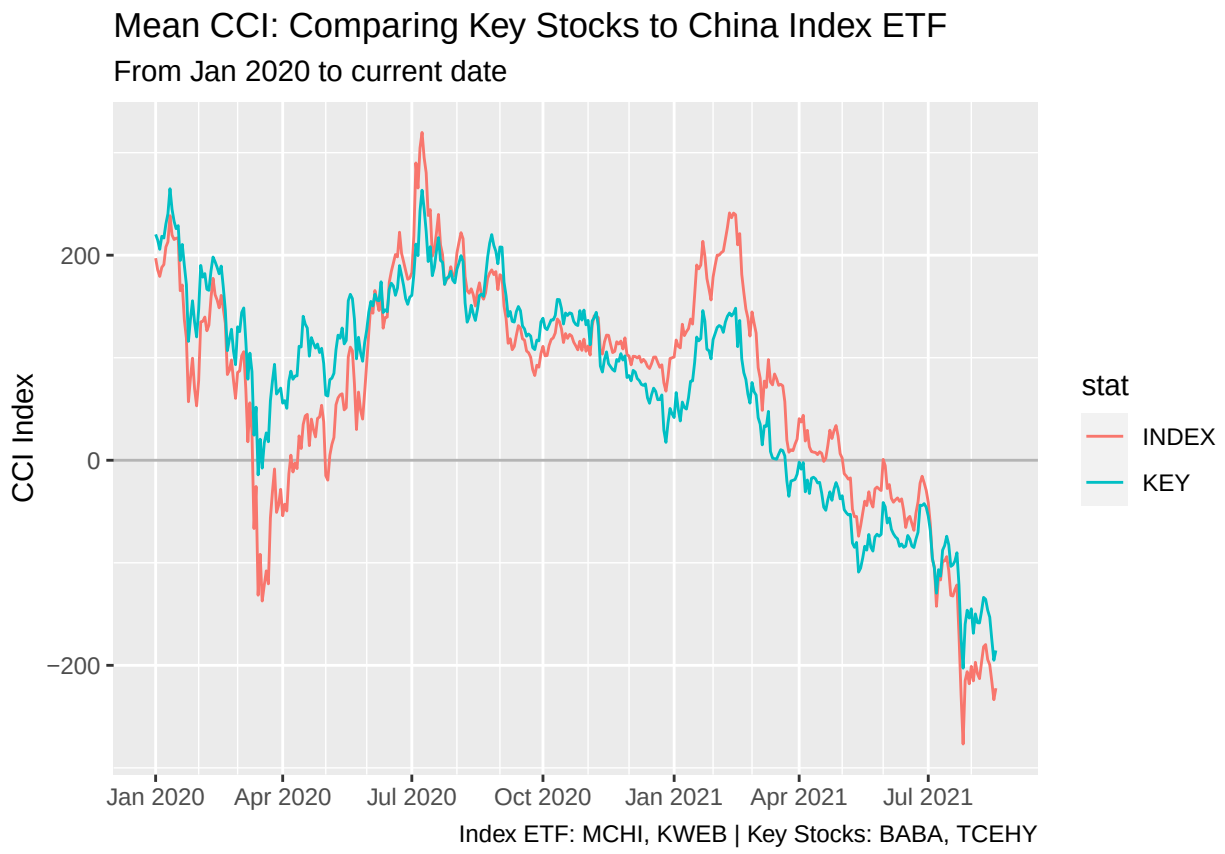


Now we use filters and our preset variable lists to create new dataframes for an index, mean, and median. We comebine them into a single dataframe for plotting.

Next we isoloate the key tech companies. Doing the same thing by getting the mean, median, and combining into a single column for plotting.

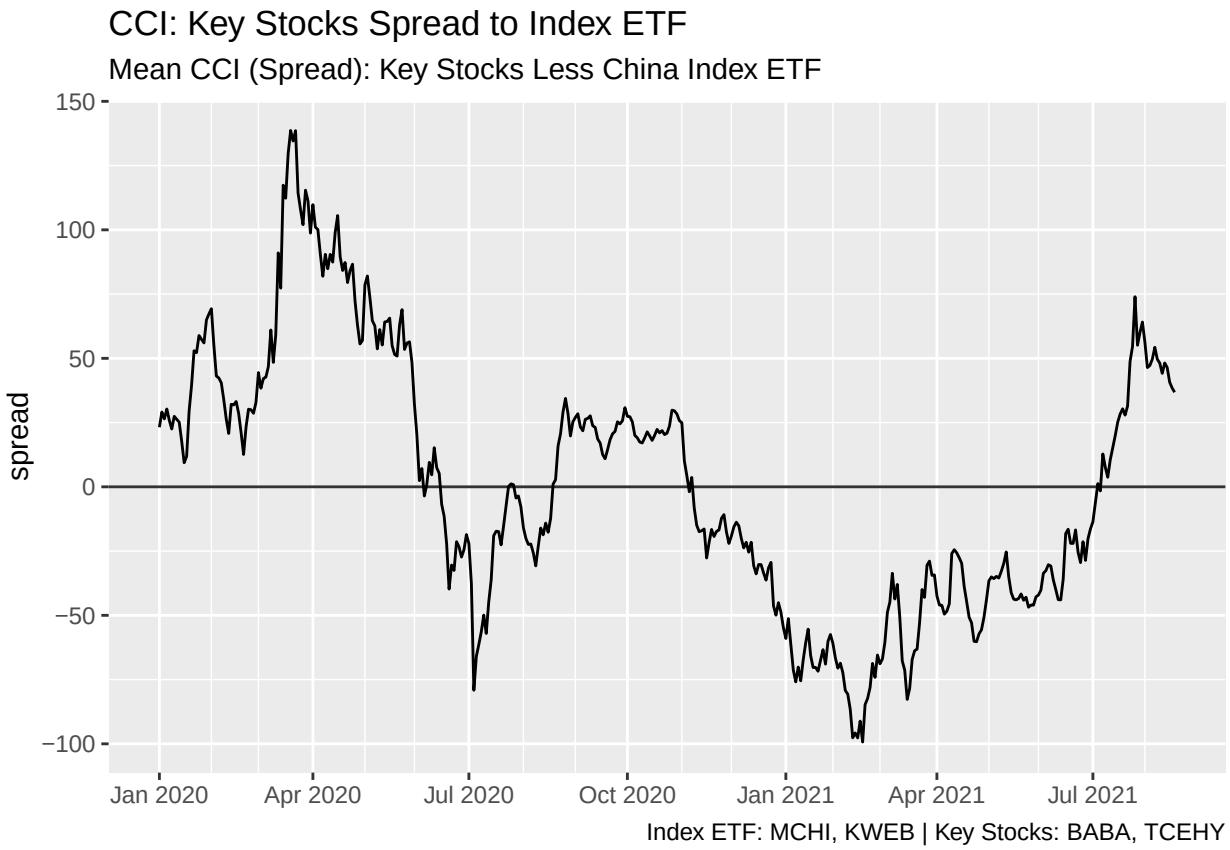


Now we will calculate the mean of the index. There are only 2 stocks in the index, KWEB and MCHI.



Now we will calculate the index spread.

```
## NOW CALCULATING THE INDEX SPREAD
data.frame(time = key.mean.cci$time, KEY = key.mean.cci$mean, INDEX = idex.mean.cci$mean) %>%
  select(time, KEY, INDEX) %>%
  mutate(spread = KEY - INDEX) %>%
  select(time, spread) %>%
  ggplot(aes(time, spread)) +
  labs(title = "CCI: Key Stocks Spread to Index ETF",
       subtitle = "Mean CCI (Spread): Key Stocks Less China Index ETF",
       caption = "Index ETF: MCHI, KWEB | Key Stocks: BABA, TCEHY") +
  geom_hline(yintercept = 0, color="black", alpha = 0.8) +
  geom_line() +
  scale_x_bd(business.dates=time, labels = date_format(format = "%b %Y"), max.major.breaks=12) +
  theme(axis.title.x=element_blank())
```




```
rm(index.cci, median.cci, mean.cci, stat.cci, key.cci, key.median.cci, key.stat.cci, key.mean.cci, key.s
```

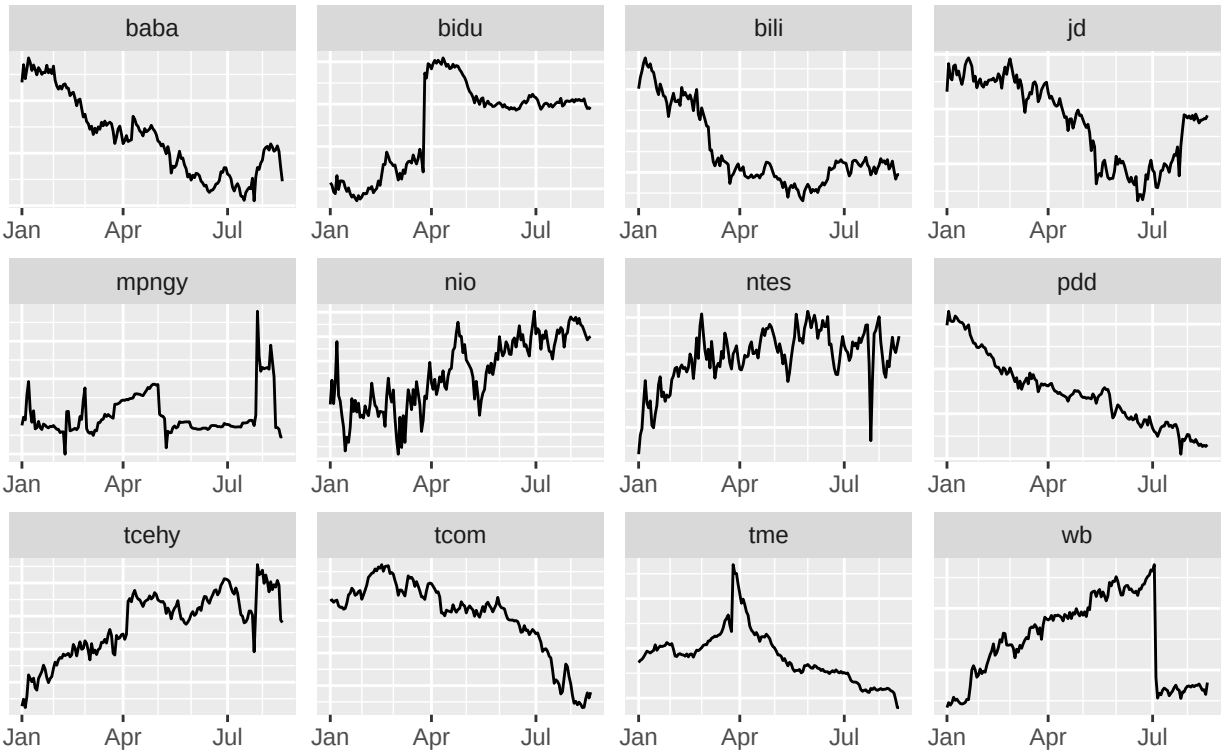
Chaikin AD Line

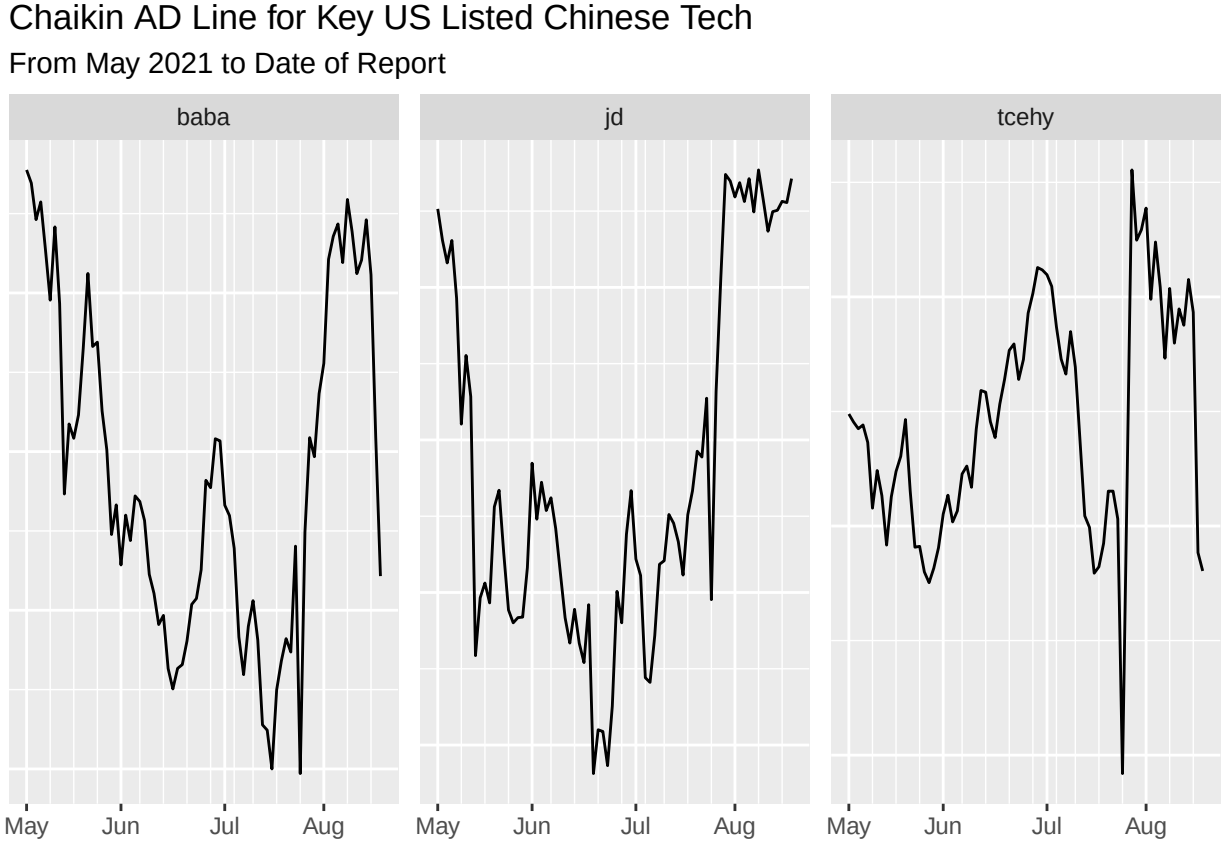
The Chaikin Oscillator (CO), also called the Chaikin Indicator, is used by traders analyze the strength of a price trend based on trading volume.

This oscillator is represented by charts: a divergence between this indicator and the price trend shows that most traders have less faith in the current price trend and believe that a trend reversal is in the making. Looking at this indicator on a relative basis for the China Internet sector could be useful in spotting the potential for a reversal. Starting with a facet_wrap.

Chaikin AD Line for US Listed Chinese Tech Stocks

From Jan 2021 to date of report



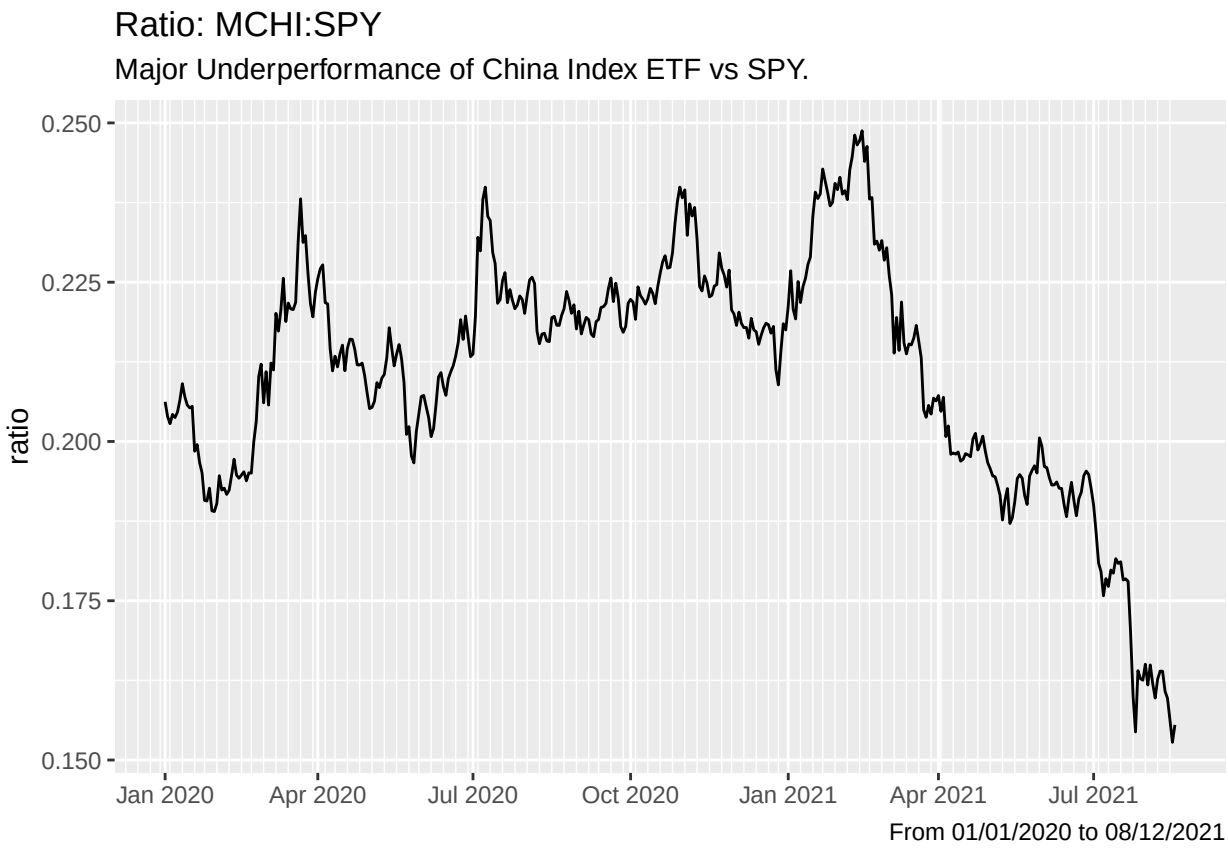


Sector Ratio

Comparing stock prices using Yahoo finance daily adjusted close data. First we will look at a ratio chart of MCHI (Broad China Index ETF) to SPY (S&P 500 US Index)

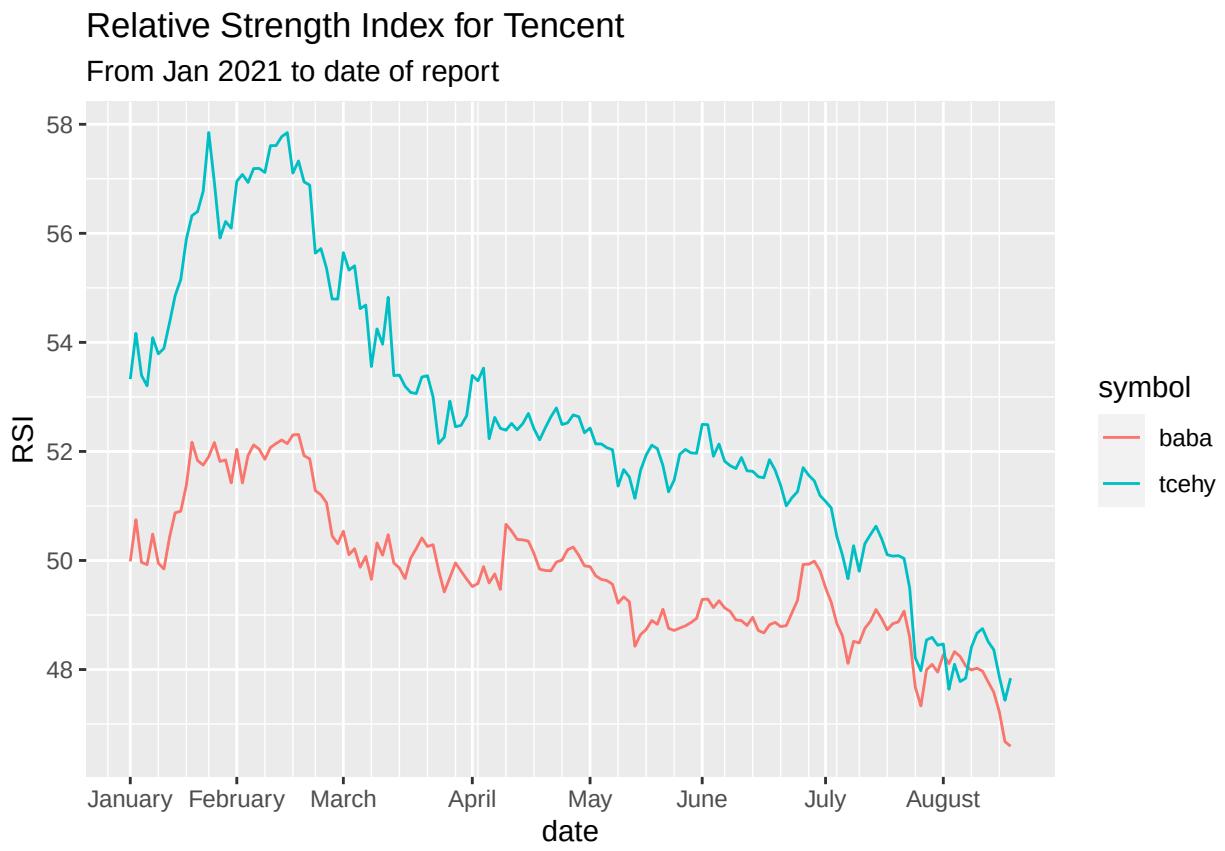
```
mchi <- stock.data %>%
  filter(date > "2020-01-01")%>%
  filter(symbol == "MCHI") %>%
  select(mchi = adjusted)
spy <- stock.data %>%
  filter(date > "2020-01-01")%>%
  filter(symbol == "SPY")
comb <- data.frame(spy, mchi)

rm(spy, mchi)

comb %>%
  mutate(ratio = mchi/adjusted) %>%
  select(date, ratio) %>%
  ggplot(aes(date, ratio)) +
  labs(title = "Ratio: MCHI:SPY",
       subtitle = "Major Underperformance of China Index ETF vs SPY.",
       caption = "From 01/01/2020 to 08/12/2021") +
  geom_line() +
  scale_x_bd(business.dates=time, labels = date_format(format = "%b %Y"), max.major.breaks=20) +
```

```
theme(axis.title.x=element_blank())
```



Relative Strength Index

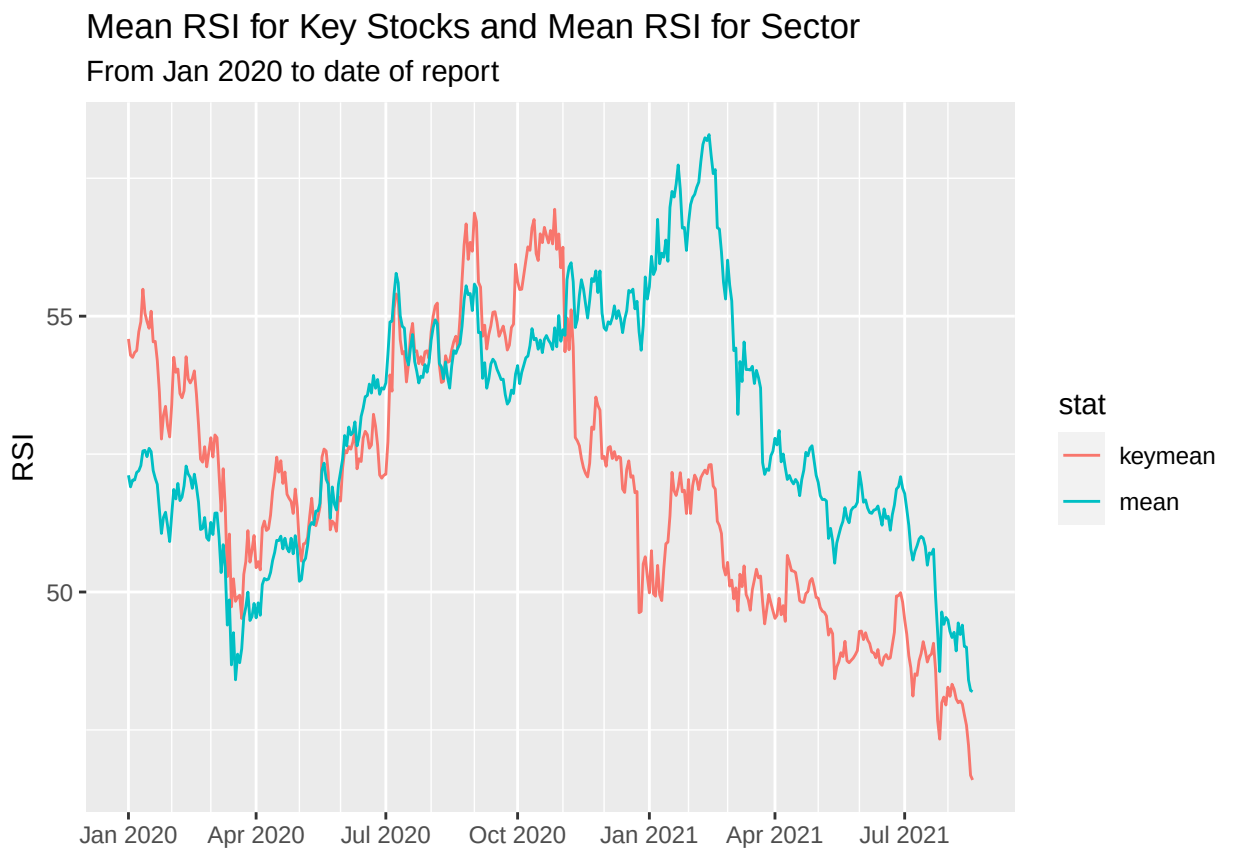
This is an indexed calculation. Can be useful to compare the RSI scores for different symbols.



First we will do some similiar comparisons. Here we will combine the mean RSI for the China Basket by including all stocks except the index stocks and Alibaba. We will then assign a seperate variable and filter the dataset for only Alibaba's RSI. Now we can compare Alibaba to the rest of the China basket, with Alibaba excluded.

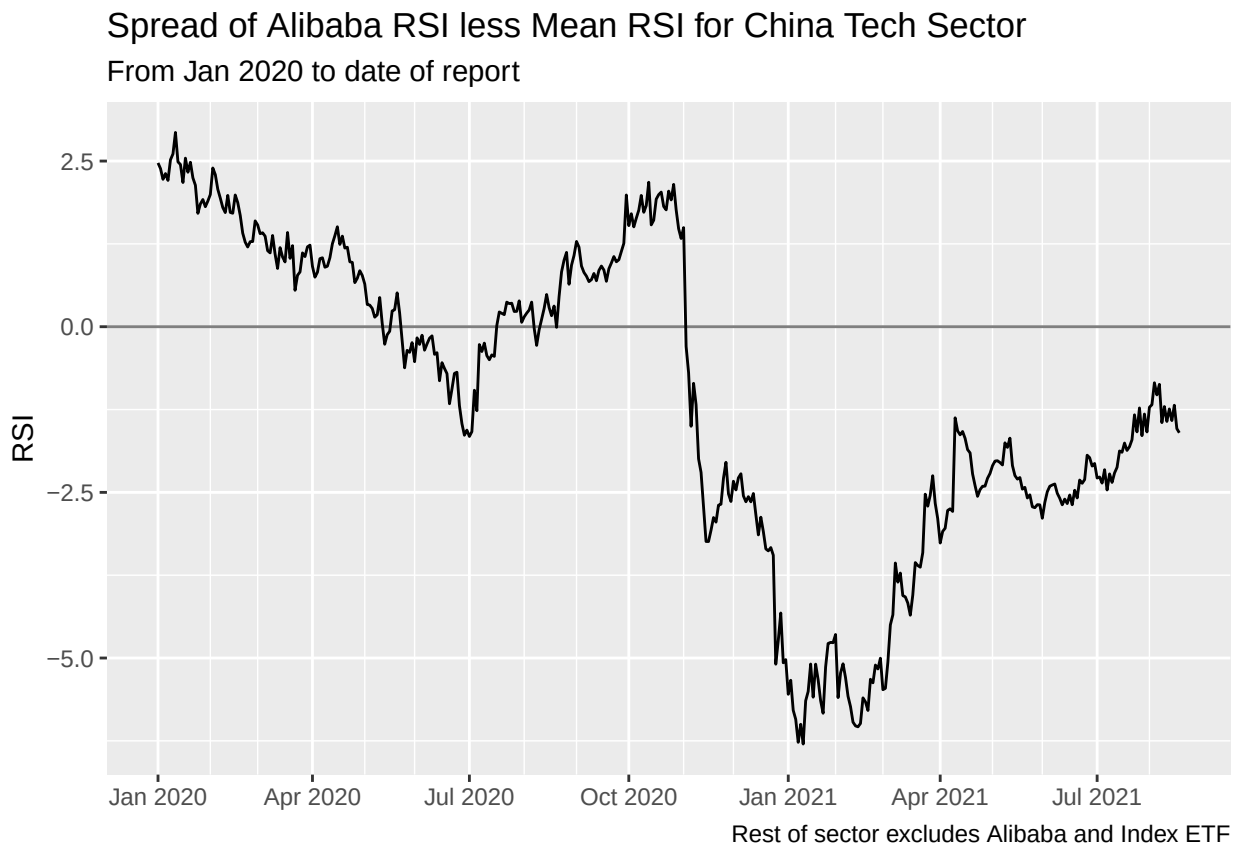
```
rm(temp.select)
### Recent Edits

mean.rsi <- df.RSI %>%
  filter(symbol != index) %>%
  filter(!symbol %in% "BABA") %>%
  group_by(time) %>%
  summarise(mean = mean(RSI, na.rm = TRUE))
key.rsi <- df.RSI %>%
  filter(symbol %in% "baba") %>%
  group_by(time) %>%
  summarise(mean = mean(RSI, na.rm = TRUE)) %>%
  select(keymean = mean)

baba.spread.rsi <- data.frame(mean.rsi, key.rsi) %>%
  mutate(spread = keymean - mean) %>%
  select(time, BABA.Spread = spread)

stat.rsi <- data.frame(mean.rsi, key.rsi) %>%
  gather(key = "stat", value = "value", -time)
```

```
stat.rsi %>%
  ggplot(aes(time, value, color = stat)) +
  labs(title = "Mean RSI for Key Stocks and Mean RSI for Sector",
       subtitle = "From Jan 2020 to date of report",
       y = "RSI") +
  geom_line() +
  scale_x_bd(business.dates=time, labels = date_format(format = "%b %Y"), max.major.breaks=12) +
  theme(axis.title.x=element_blank())
```



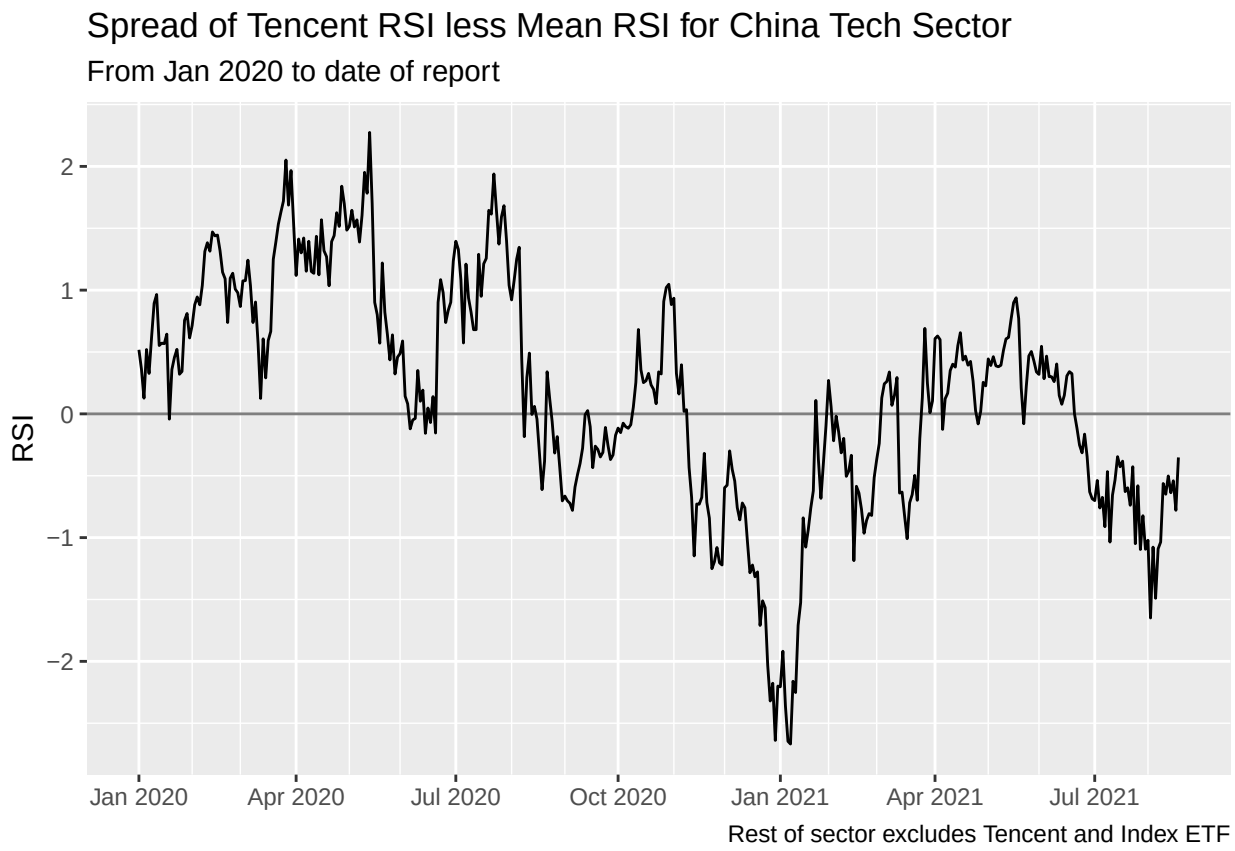
Next we will create the spread column and chart it. Taking the two values from the previous chart, subtracting them, and plotting the spread.

```
data.frame(mean.rsi, key.rsi) %>%
  mutate(spread = keymean - mean) %>%
  select(time, spread) %>%
ggplot(aes(time, spread)) +
  labs(title = "Spread of Alibaba RSI less Mean RSI for China Tech Sector",
       subtitle = "From Jan 2020 to date of report",
       caption = "Rest of sector excludes Alibaba and Index ETF",
       y = "RSI") +
  geom_line() +
  geom_hline(yintercept = 0, color = "black", alpha = 0.5) +
  scale_x_bd(business.dates=time, labels = date_format(format = "%b %Y"), max.major.breaks=12) +
  theme(axis.title.x=element_blank())
```

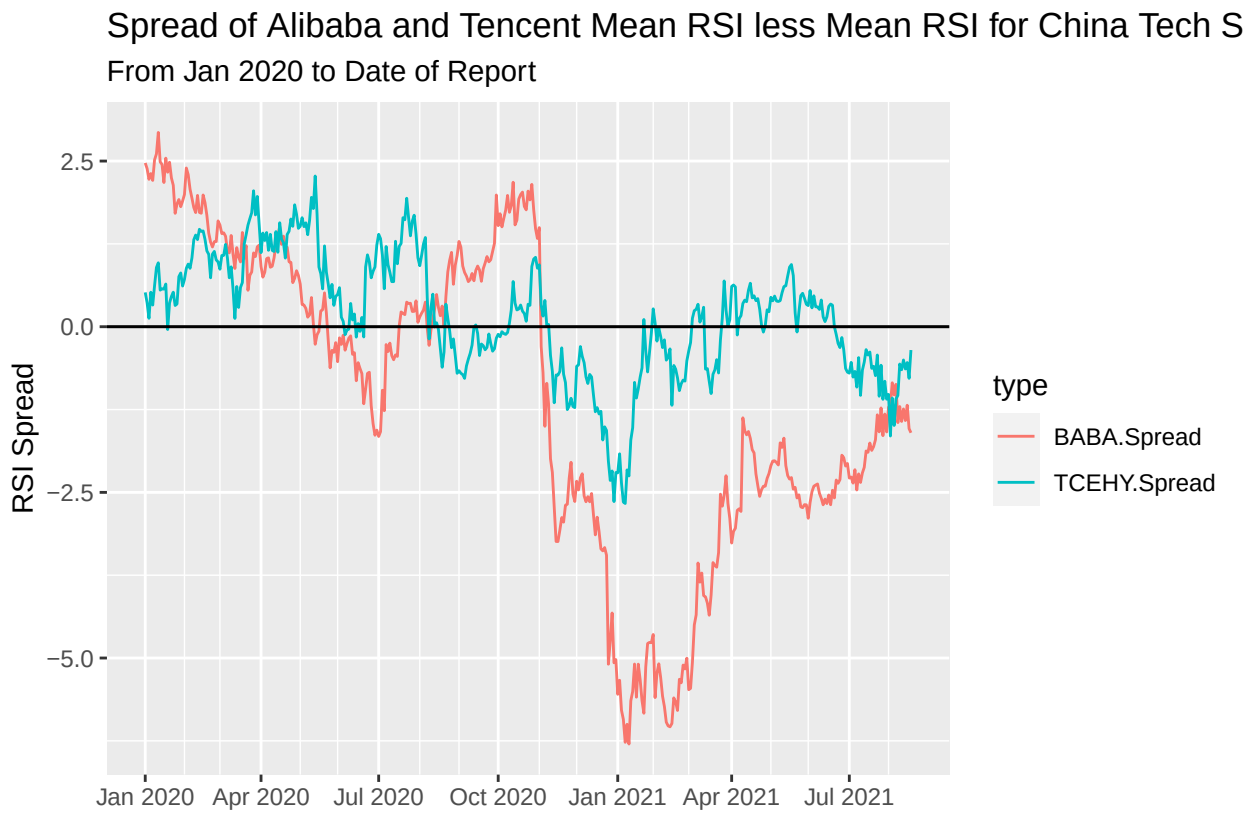


This next chunk will run the same calculation as the prior, but with Tencent instead. We will plot the spread of Tencents RSI spread to the China basket excluding Tencent.

```
data.frame(mean.rsi, key.rsi) %>%
  mutate(spread = keymean - mean) %>%
  select(time, spread) %>%
  ggplot(aes(time, spread)) +
  labs(title = "Spread of Tencent RSI less Mean RSI for China Tech Sector",
       subtitle = "From Jan 2020 to date of report",
       caption = "Rest of sector excludes Tencent and Index ETF",
       y = "RSI") +
  geom_line() +
  geom_hline(yintercept = 0, color = "black", alpha = 0.5) +
  scale_x_bd(business.dates=time, labels = date_format(format = "%b %Y"), max.major.breaks=12) +
  theme(axis.title.x=element_blank())
```



This next chart compares the RSI to index spreads for Tencent and Alibaba. An interesting visual showing their relative strength against the rest of the index. When looking at these results its important to consider that the mean and median calculations in this dataset are based off equal weight indexes. It would likely be more appropriate to do the same calculation on a market capitalization weighted basis to get a better guage of sector performance.

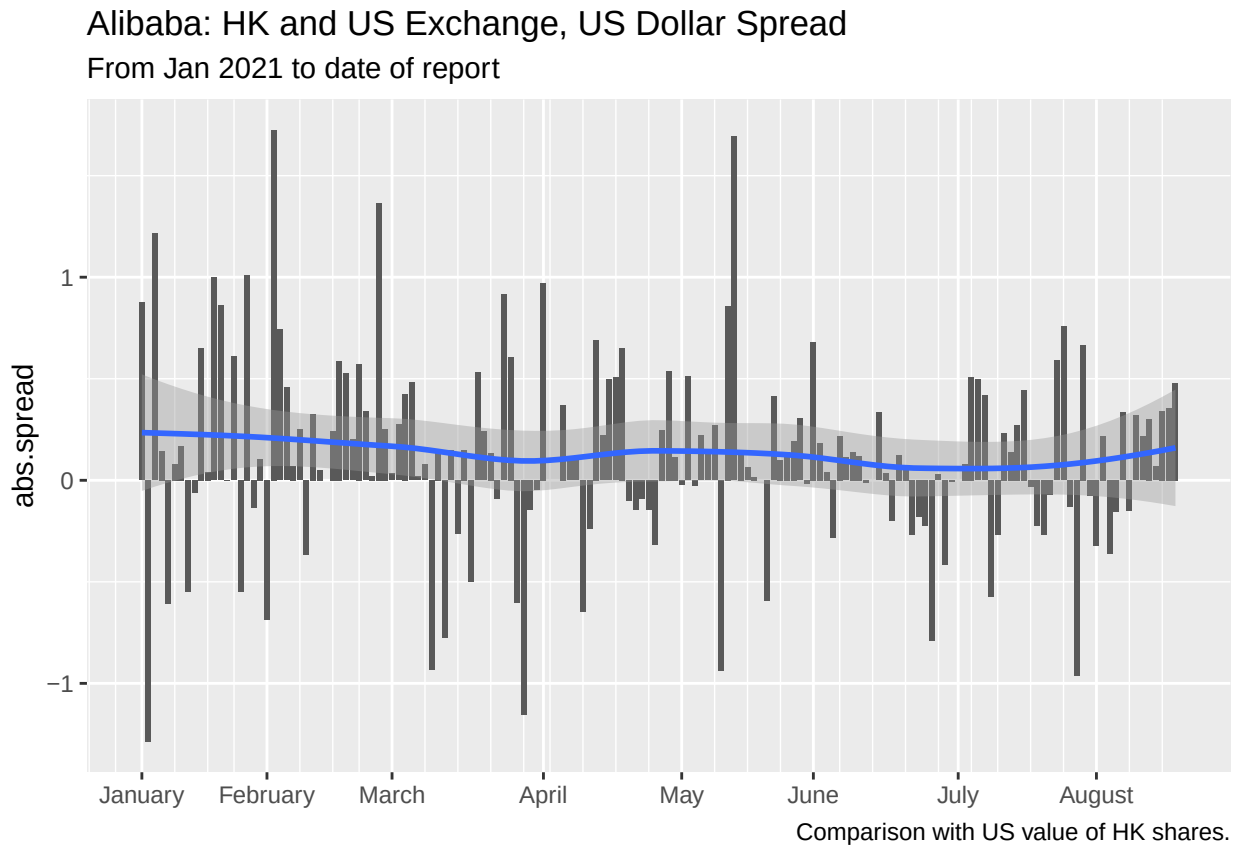


Exchange Spreads

Next we will look at the spread between the price of Alibaba on the US NYSE and the price on the Hong Kong market. To understand the relationship between the two, use this calculation. 1x US Share of Alibaba = 8x HK Shares. 1x HK / HK-USD exchange = HK Price in US. Comparisons are done on this equivalent basis. For this calculation the smooth (Loess) line above zero represents a premium for HK shares of the US shares. If the line goes below zero it represents a discount. This spread is calculated on a daily absolute US dollar basis.

```
spread.sl.hk <- stock.data.hk %>%
  spread(symbol, adjusted)

spread.sl.hk <- spread.sl.hk %>%
  mutate(BABA.conv = BABA / 8)

spread.sl.hk <- spread.sl.hk %>%
  filter(date > "2021-01-01") %>%
  select(date, BABA.conv, BABAHK = "9988.HK", HKD = "HKD=X") %>%
  mutate(BABACONV = BABAHK / HKD) %>%
  select(date, BABA.conv, BABACONV)

spread.sl.hk <- spread.sl.hk %>%
  mutate(abs.spread = BABACONV - BABA.conv)
spread.sl.hk.copy <- spread.sl.hk %>%
  mutate(abs.spread = BABACONV - BABA.conv)
```

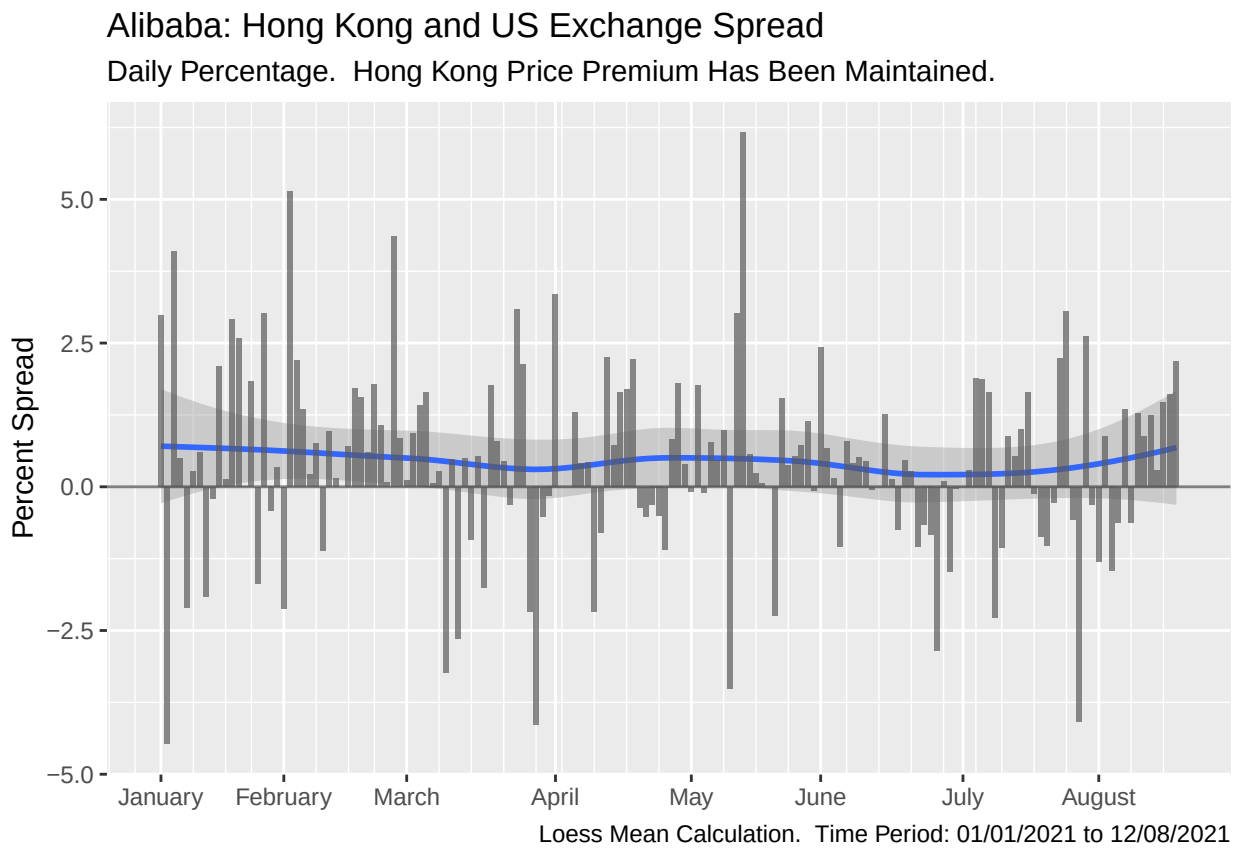
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Next we will look at the exchange spread on a daily percentage basis.

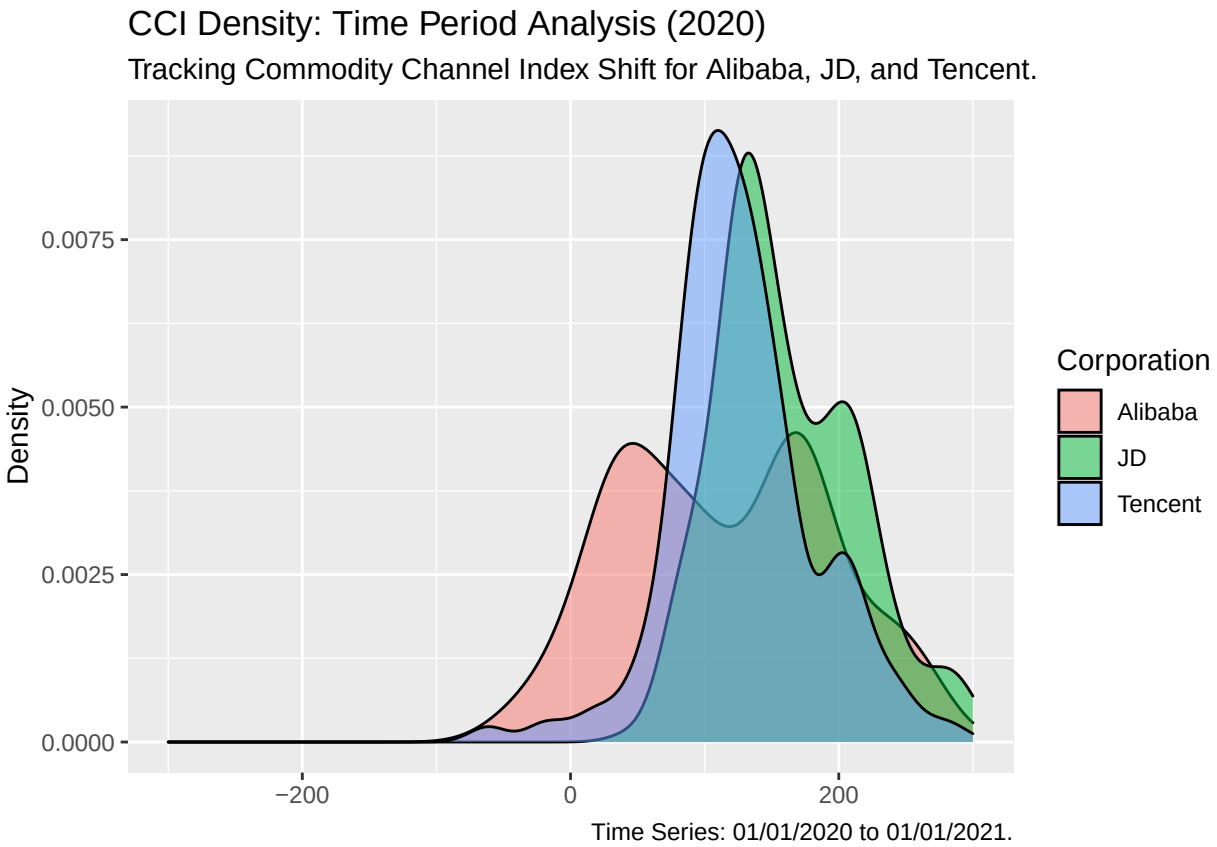
```
spread.sl.hk <- spread.sl.hk %>%
  mutate(spread.pct = (abs.spread/BABACONV)*100) %>%
  select(date, spread.pct)

spread.sl.hk %>%
  ggplot(aes(date, spread.pct)) +
  labs(title = "Alibaba: Hong Kong and US Exchange Spread",
       subtitle = "Daily Percentage.  Hong Kong Price Premium Has Been Maintained.",
       caption ="Loess Mean Calculation.  Time Period: 01/01/2021 to 12/08/2021",
       y = "Percent Spread") +
  stat_smooth(method = loess) +
  geom_hline(yintercept = 0, alpha = 0.5) +
  geom_col(alpha = 0.7) +
  scale_x_bd(business.dates=time, labels = date_format(format = "%B"), max.major.breaks=12) +
  theme(axis.title.x=element_blank())
```

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

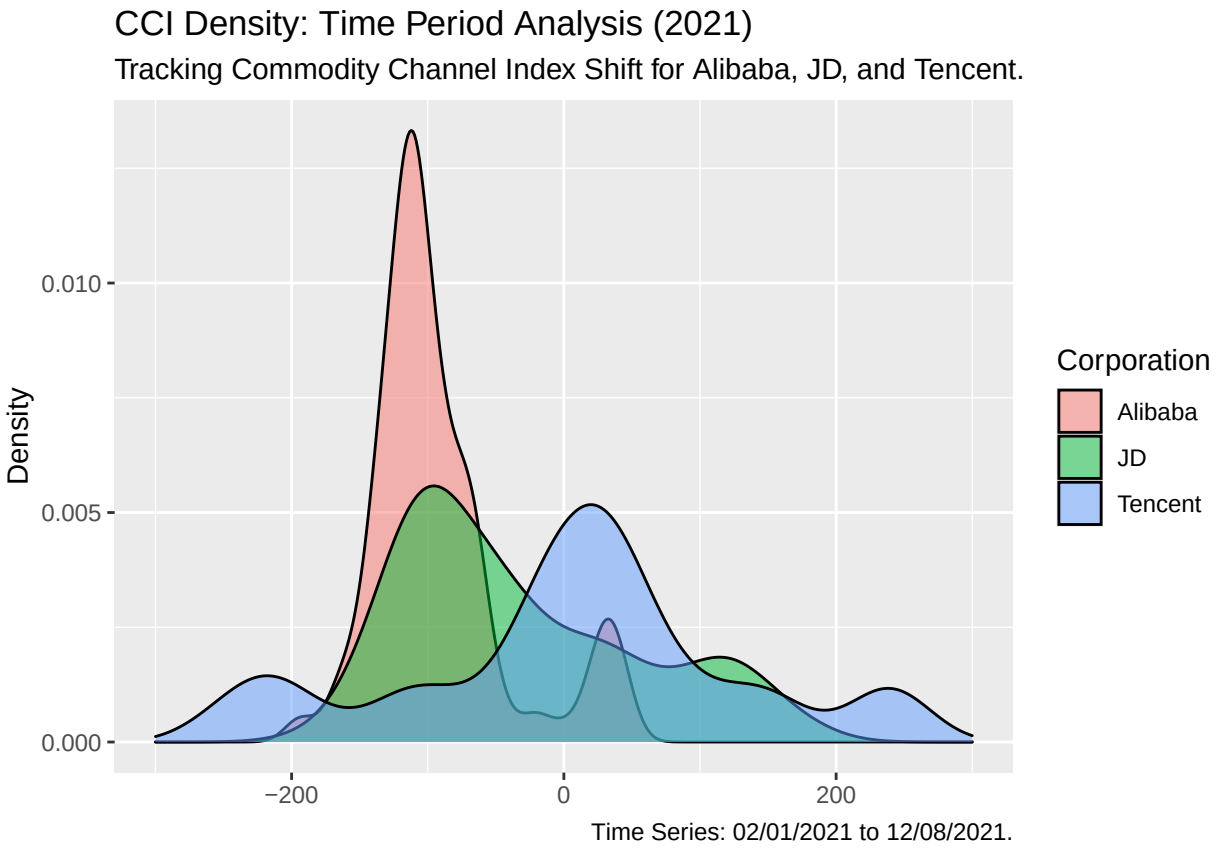


Density Plot Shift

Lastly, we will come back to the CCI data for Alibaba, Tencent, and JD. We will review the density of CCI results for 2 time periods. The first is Jan 1, 2020 to Jan 1, 2021 which is presented below.

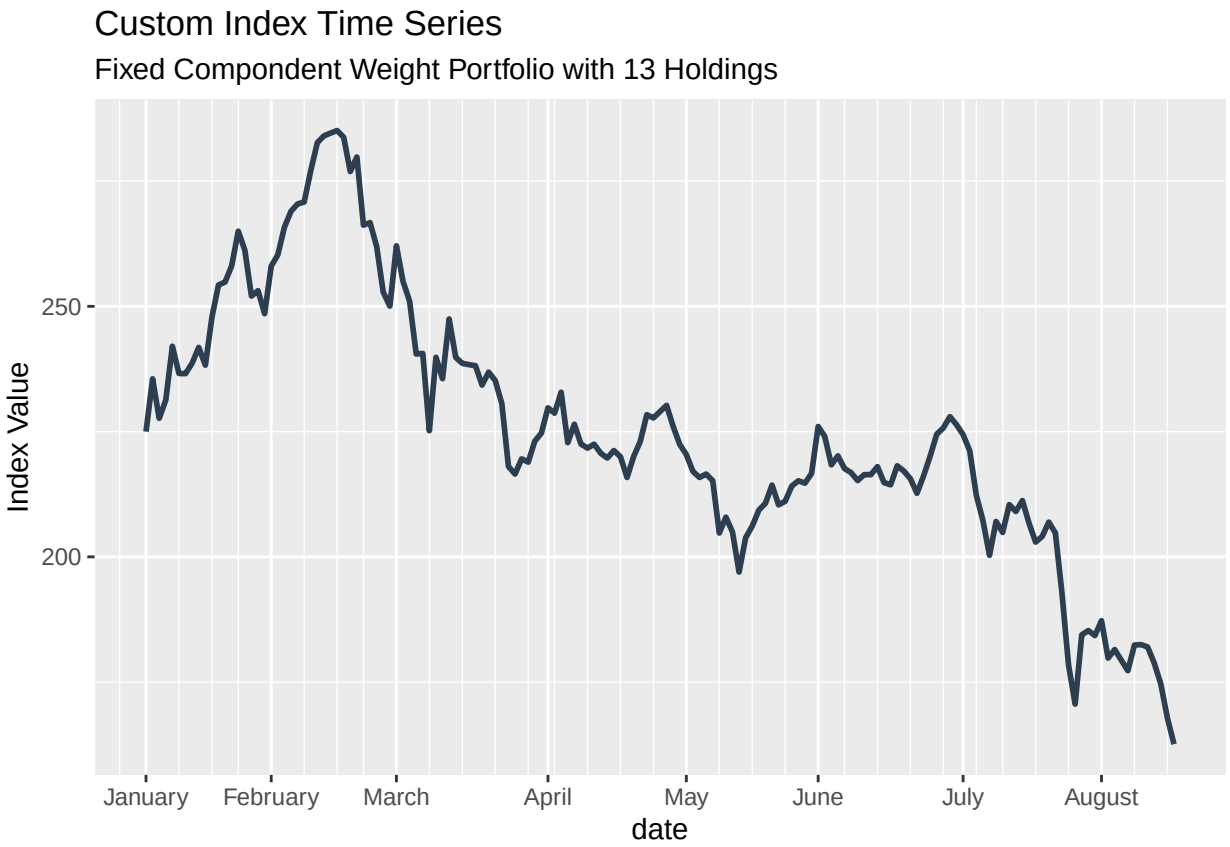


The second is from Jan 2, 2021 to August 12, 2021. These plots show a clear shift in the price density for the 3 key internet companes in China.



Custom Index

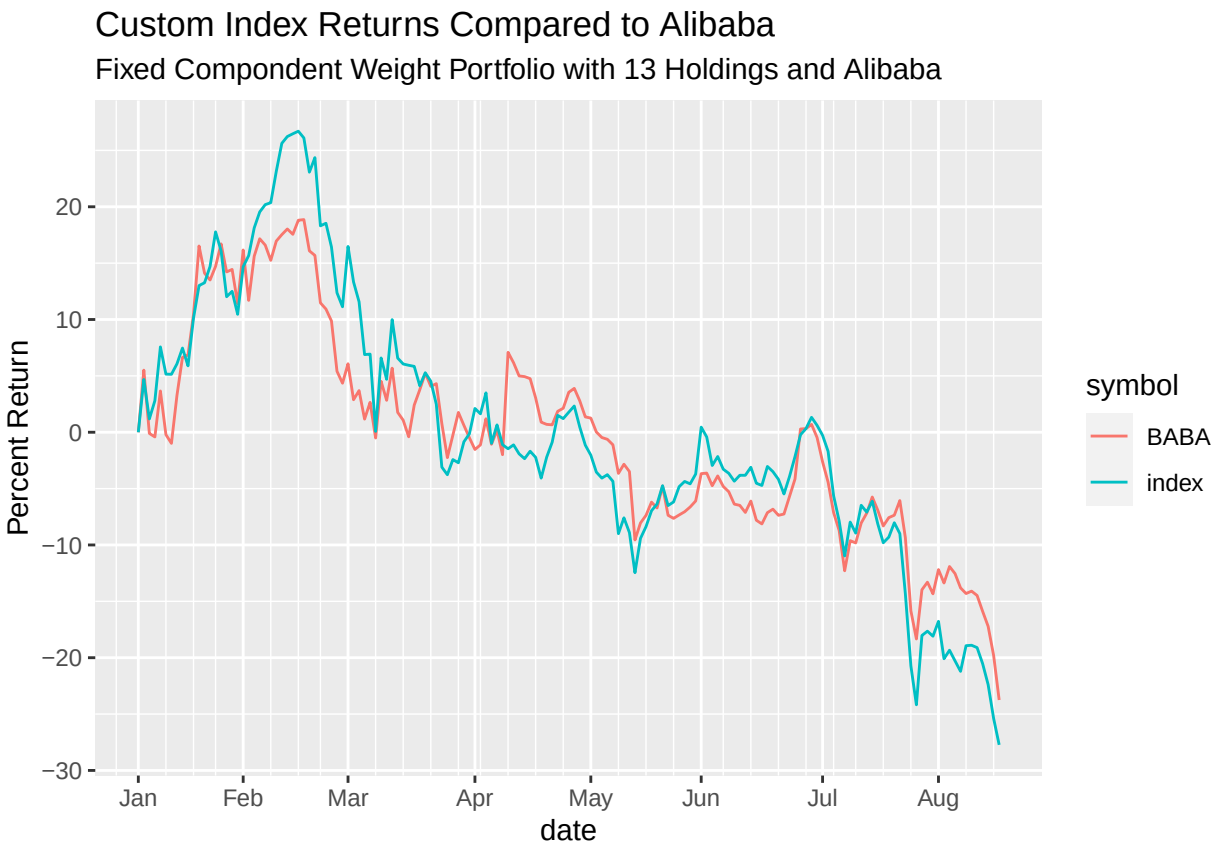
Calculating a custom index with pre-established weights that are setup based on current market capitalization. weights are printed at bottom of code chunk.



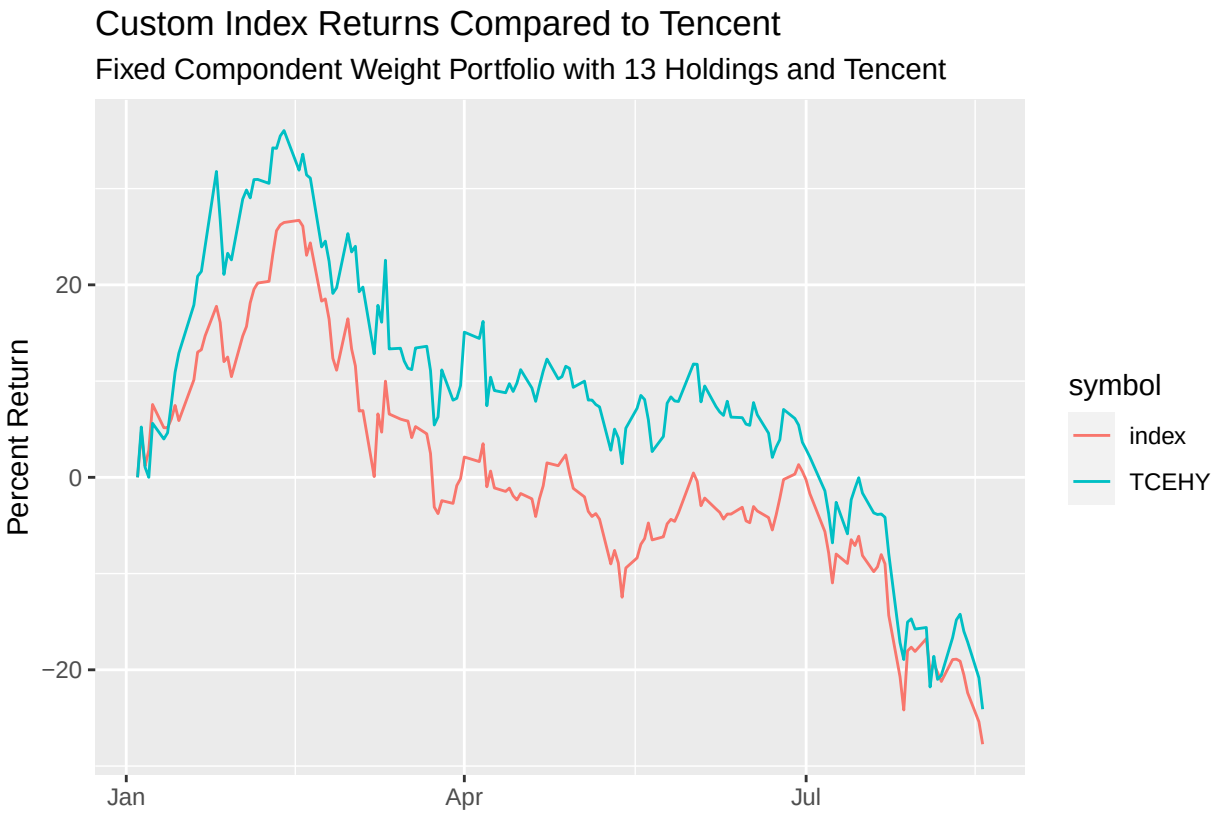
Now we can look performance of stocks on an return basis. Looking at comparisons of the index to Alibaba, Tencent and SPY. We use Tidyquant for the conversion from price series to daily return data.

```
sp.baba <- stock.list.2 %>%
  filter(date > "2021-01-01") %>%
  filter(symbol %in% "BABA") %>%
  select(symbol, date, adjusted)

merged <- full_join(sp.baba, basket, by = c('symbol','date', 'adjusted'))

merged.returns <- merged %>%
  group_by(symbol) %>%
  tq_transmute(select     = adjusted,
               mutate_fun = periodReturn,
               period     = "daily",
               col_rename = "daily_return")

# probably a better way to do this.
merged.returns <- merged.returns %>%
  mutate(cm.ret = 100*(cumprod(1 + daily_return)))

merged.returns %>%
  ggplot(aes(date, cm.ret-100, color = symbol)) +
    labs(title = "Custom Index Returns Compared to Alibaba",
       subtitle = "Fixed Compondent Weight Portfolio with 13 Holdings and Alibaba",
       x = "", y = "Percent Return") +
```

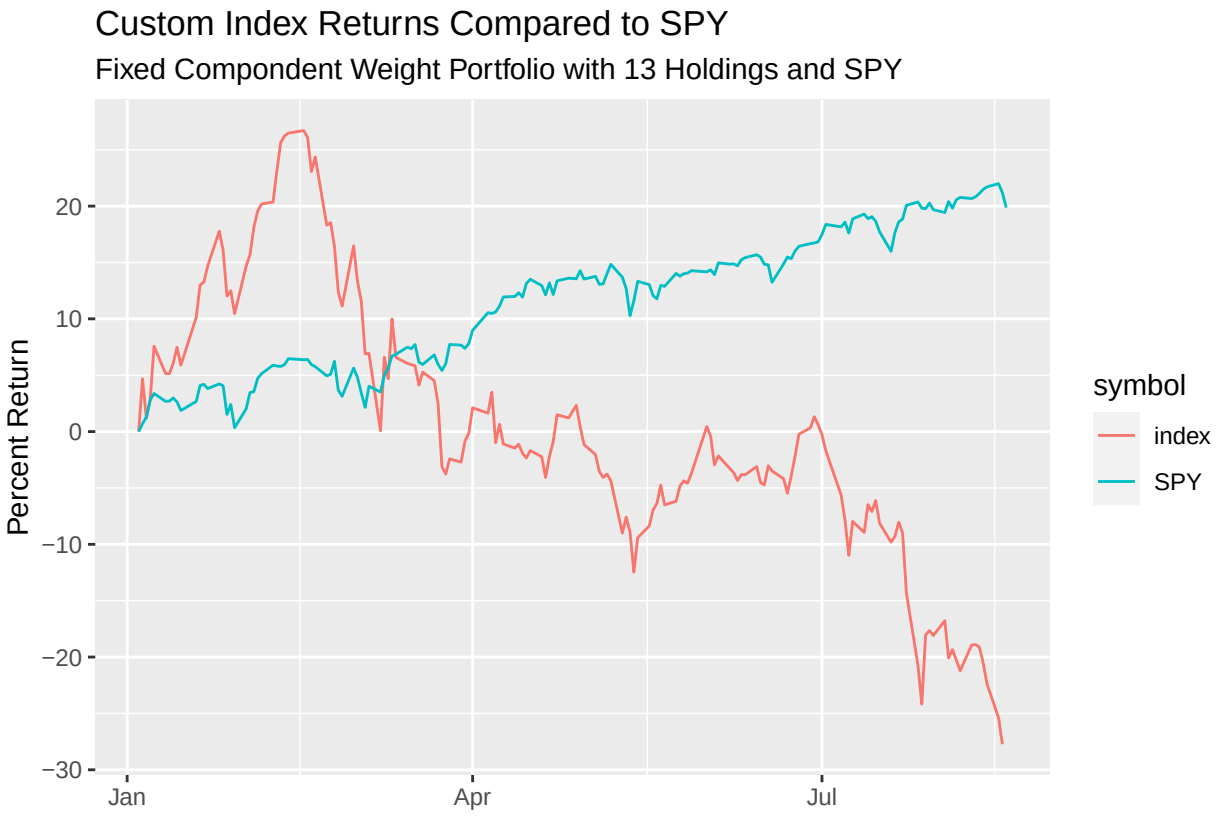
```
  geom_line() +
    scale_x_bd(business.dates=time, labels = date_format(format = "%b"), max.major.breaks=12)
```



Next we look at Tencent.



Lastly we will look at the index comparison to SPY.



Thank you for taking the time to read this report.