China Stock Analysis

Austin McGhee

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

Data Import

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.

```
# Set key symbol groups and dates.
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")

# Set a dates column that from the data to use in graphs later.
```

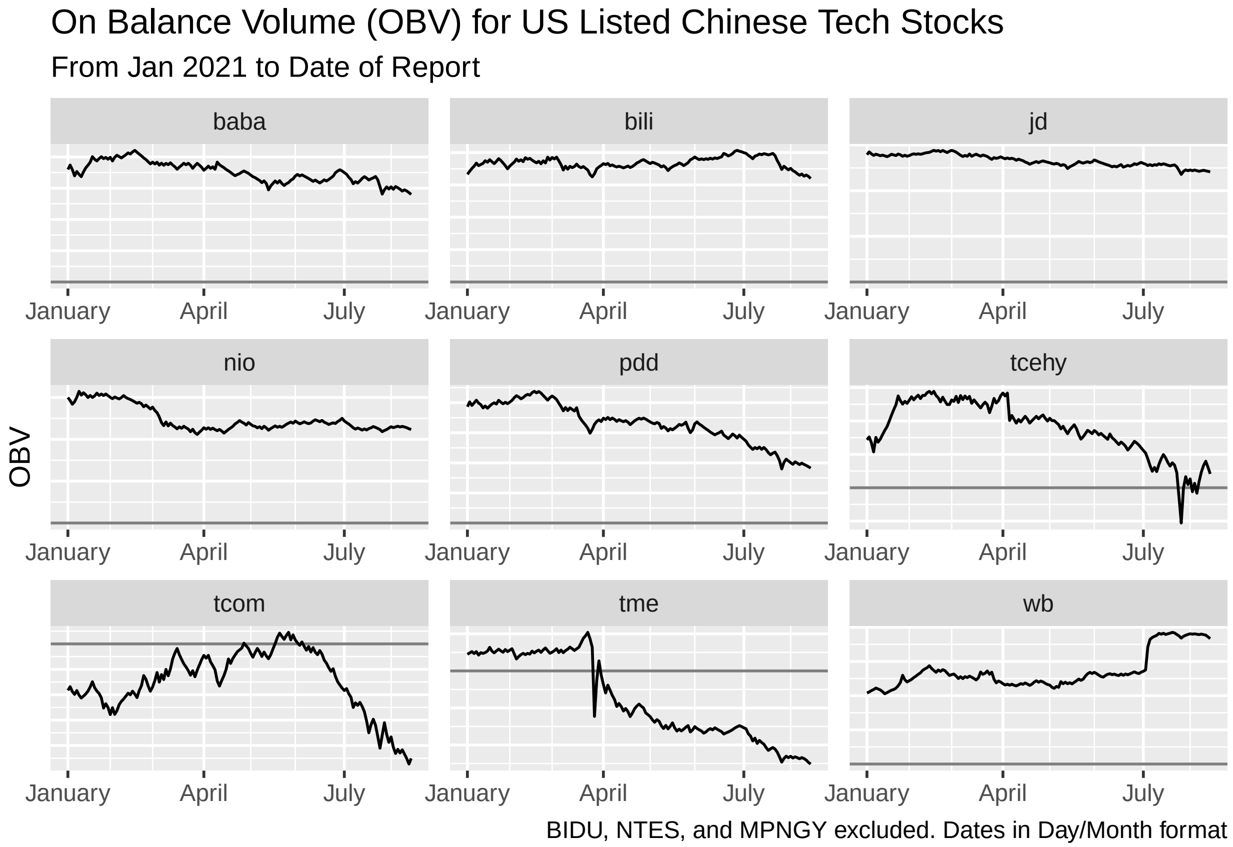
```
time <- stock.data$date %>%
  sort(descreasing = TRUE)
```

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")
```

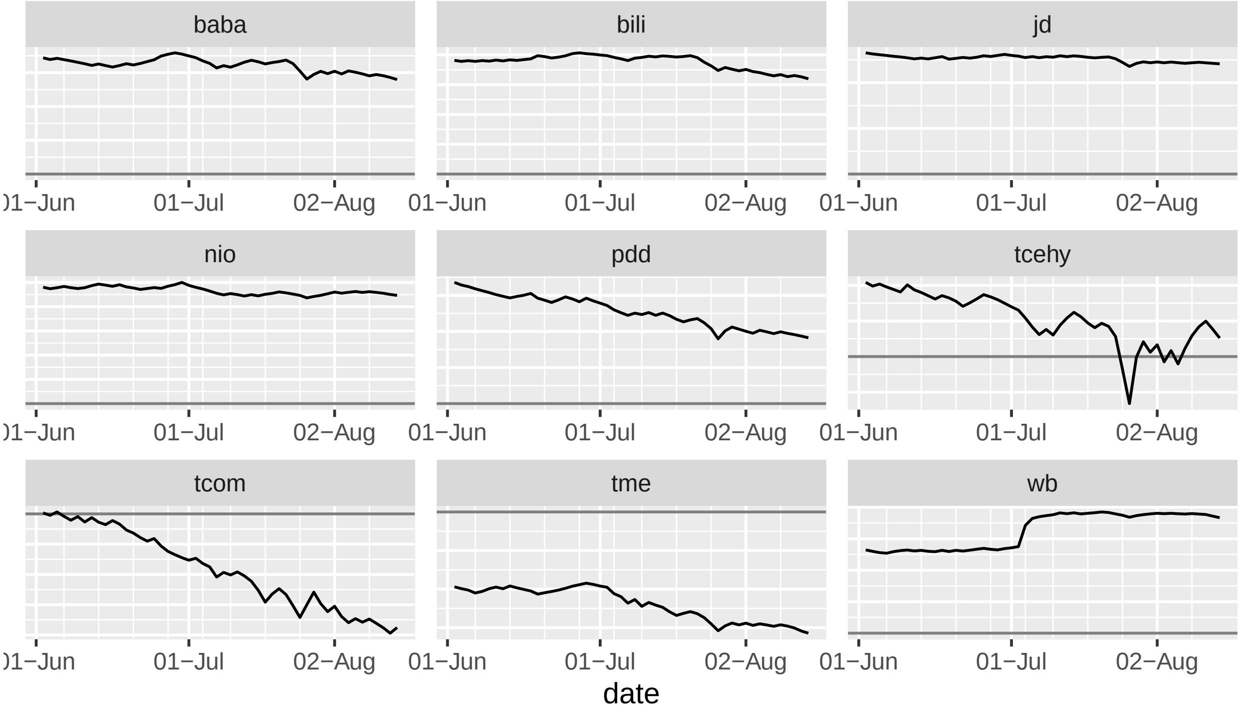


Next view is more zoomed in, same code but with date filter. We are excluding bidu and ntes because the charts appear not to format correctly with this indicator.

```r
# Zoomed in facet wrap for all symbols
df.OBV %>%
  filter(time > "2021-06-01") %>%
  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 for US Listed Chinese Tech Stocks",
       subtitle = "From Jan 2021 to Date of Report.  OBV on Y Axis",
       caption = "BIDU, NTES, and MPNGY excluded. Dates in Day/Month format",
       y= "OBV", x = "Date") +
  scale_x_bd(business.dates=time, labels = date_format(format = "%d-%b"),max.major.breaks=4) +
  theme(axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) +
  facet_wrap(. ~ symbol, scale = "free")
```

On Balance Volume for US Listed Chinese Tech Stocks

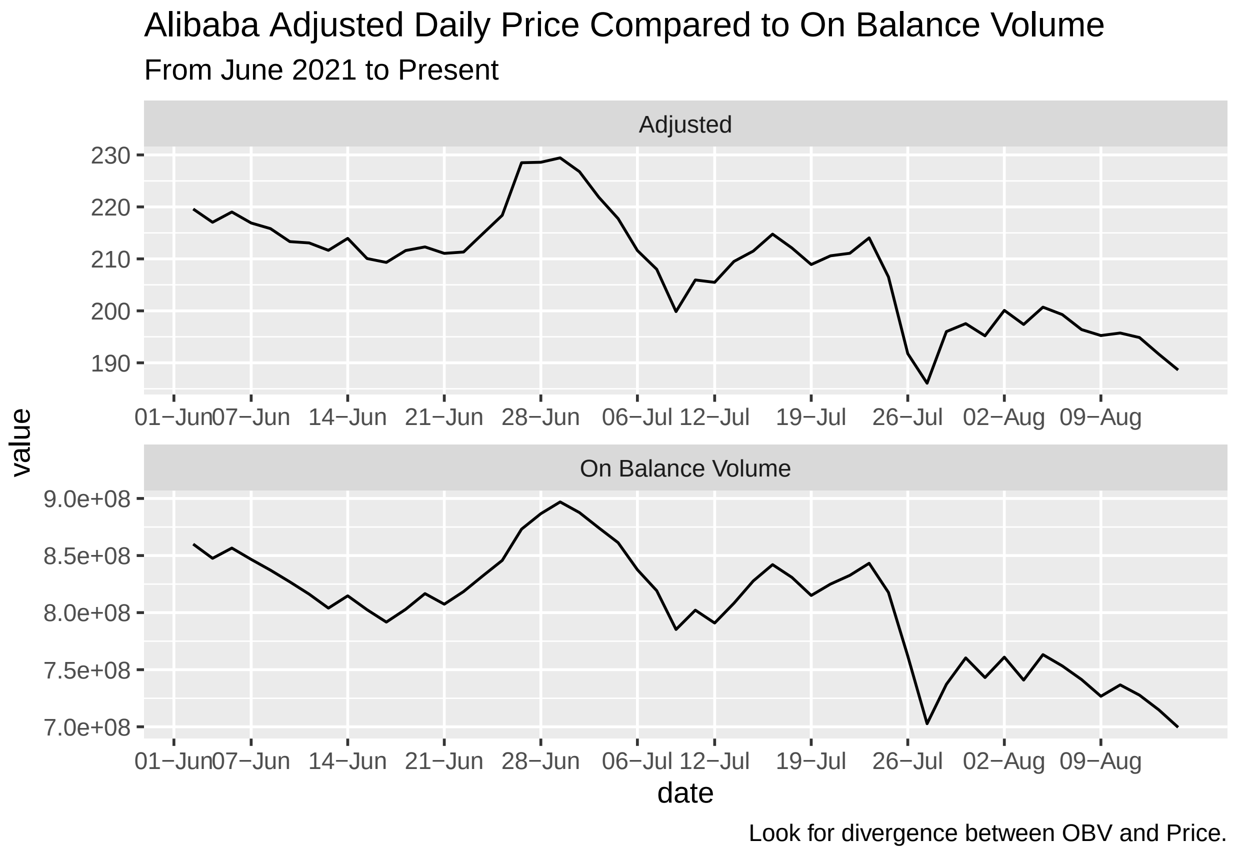
From Jan 2021 to Date of Report. OBV on Y Axis



BIDU, NTES, and MPNGY excluded. Dates in Day/Month format

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

```
# Next 2 blocks BABA and baba are stock price and OBV data.  The 3rd block combines them into a datafra
BABA <- stock.data %>%
  filter(date > chart.date) %>%
  filter(symbol %in% "BABA") %>%
  select(time = date, adjusted)

baba <- df.OBV %>%
  filter(time > chart.date) %>%
  filter(symbol %in% "baba") %>%
  select(time, OBV)

baba.fw <- merge(x = BABA, y = baba, by = "time", all = TRUE) %>%
  select(time, Adjusted = adjusted, "On Balance Volume" = OBV) %>%
  gather(key = "type", value = "value", -time)

# Now filter the dataframe by date and plot the price and OBV together.  On on top of another.  You loo
baba.fw %>%
  filter(time > "2021-06-01") %>%
  ggplot(aes(time, value)) +
  geom_line() +
  labs(title = "Alibaba Adjusted Daily Price Compared to On Balance Volume",
       caption = "Look for divergence between OBV and Price.",
       subtitle = "From June 2021 to Present") +
  facet_wrap( ~ type,nrow = 2,
              scales = "free") +
  scale_x_bd(business.dates=time, labels = date_format(format = "%d-%b"), max.major.breaks=12)
```

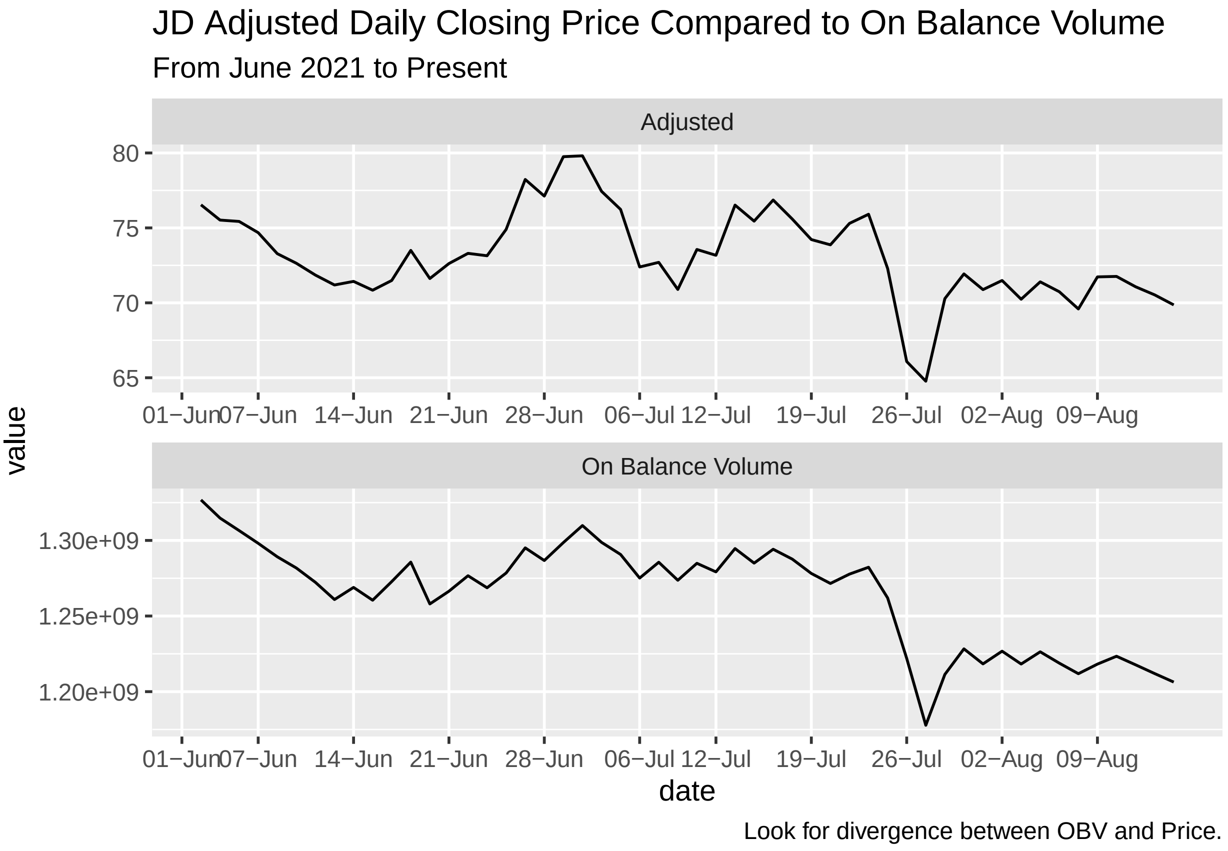


```
# We don't need the dataframes saved anymore.
rm(baba.fw, BABA, baba)
```

Now we do the same for JD.

```
## Same thing for JD now
# Filter and combine the data
JD <- stock.data %>%
  filter(date > chart.date) %>%
  filter(symbol %in% "JD") %>%
  select(time = date, adjusted)

jd <- df.OBV %>%
  filter(time > chart.date) %>%
  filter(symbol %in% "jd") %>%
  select(time, OBV)

jd.fw <- merge(x = JD, y = jd, by = "time", all = TRUE) %>%
  select(time, Adjusted = adjusted, "On Balance Volume" = OBV) %>%
  gather(key = "type", value = "value", -time)

# And Chart it
```

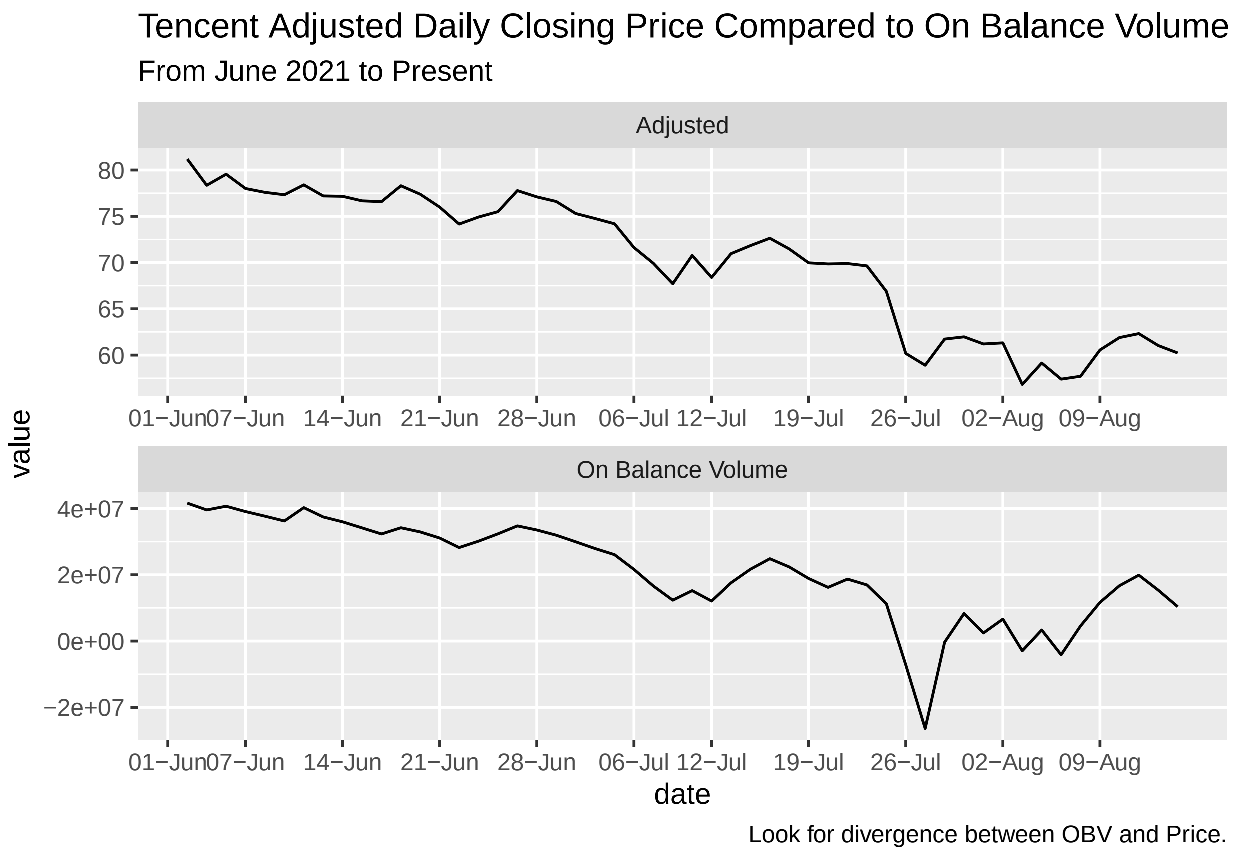
```
jd.fw %>%
  filter(time > "2021-06-01") %>%
  ggplot(aes(time, value)) +
  geom_line() +
  labs(title = "JD Adjusted Daily Closing Price Compared to On Balance Volume",
       caption = "Look for divergence between OBV and Price.",
       subtitle = "From June 2021 to Present") +
  facet_wrap( ~ type,nrow = 2,
              scales = "free") +
  scale_x_bd(business.dates=time, labels = date_format(format = "%d-%b"), max.major.breaks=12)
```



```
# Remove it
rm(jd.fw, JD, jd)
```

Lastly for Tencent.

```
# Now the same thing for TCEHY.
# Filter and combine
TCEHY <- stock.data %>%
  filter(date > chart.date) %>%
  filter(symbol %in% "TCEHY") %>%
  select(time = date, adjusted)

tcehy <- df.OBV %>%
  filter(time > chart.date) %>%
  filter(symbol %in% "tcehy") %>%
  select(time, OBV)

tcehy.fw <- merge(x = TCEHY, y = tcehy, by = "time", all = TRUE) %>%
  select(time, Adjusted = adjusted, "On Balance Volume" = OBV) %>%
  gather(key = "type", value = "value", -time)

# Same Chart
tcehy.fw %>%
  filter(time > "2021-06-01") %>%
  ggplot(aes(time, value)) +
  geom_line() +
  labs(title = "Tencent Adjusted Daily Closing Price Compared to On Balance Volume",
       caption = "Look for divergence between OBV and Price.",
       subtitle = "From June 2021 to Present") +
  facet_wrap( ~ type,nrow = 2,
              scales = "free") +
  scale_x_bd(business.dates=time, labels = date_format(format = "%d-%b"), max.major.breaks=12)
```

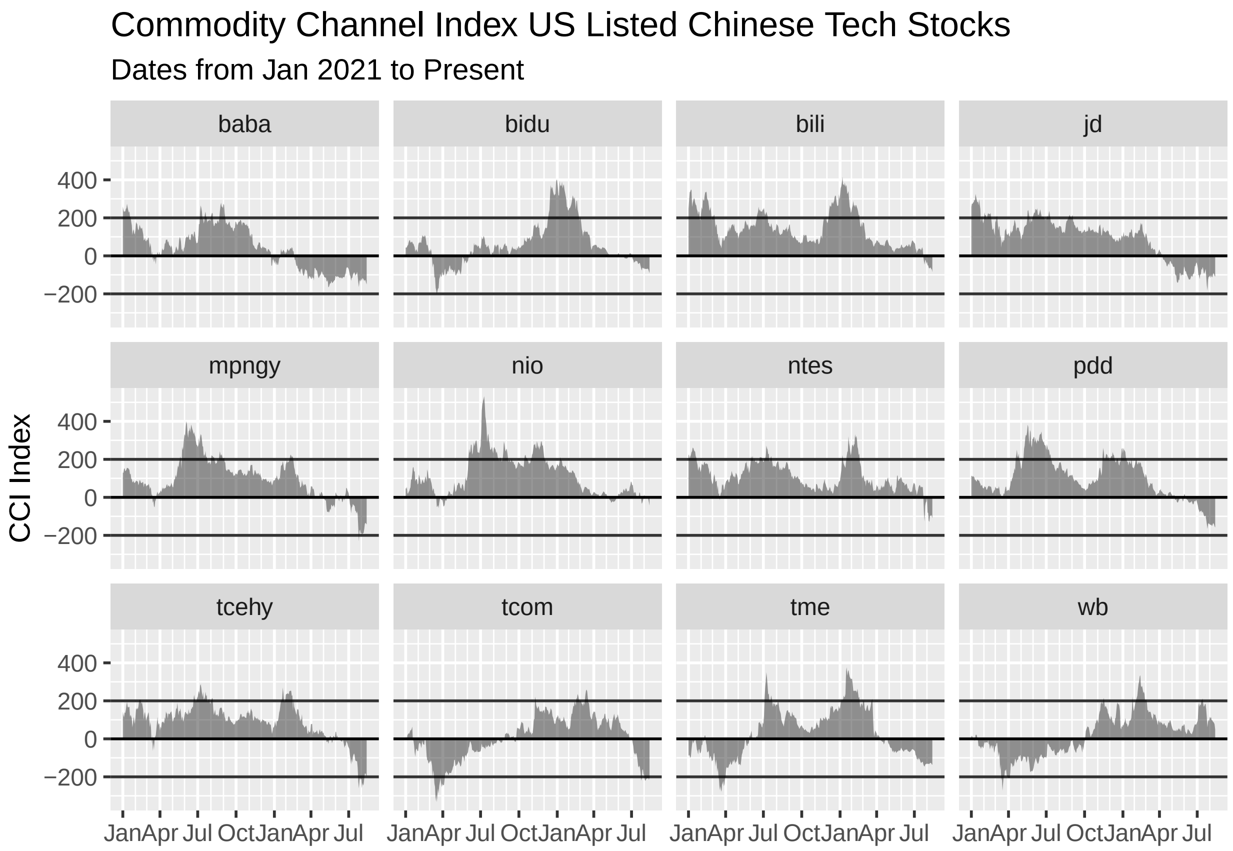


```
# Remove
rm(tcehy.fw, TCEHY, tcehy)
```

Commodity Channel Index

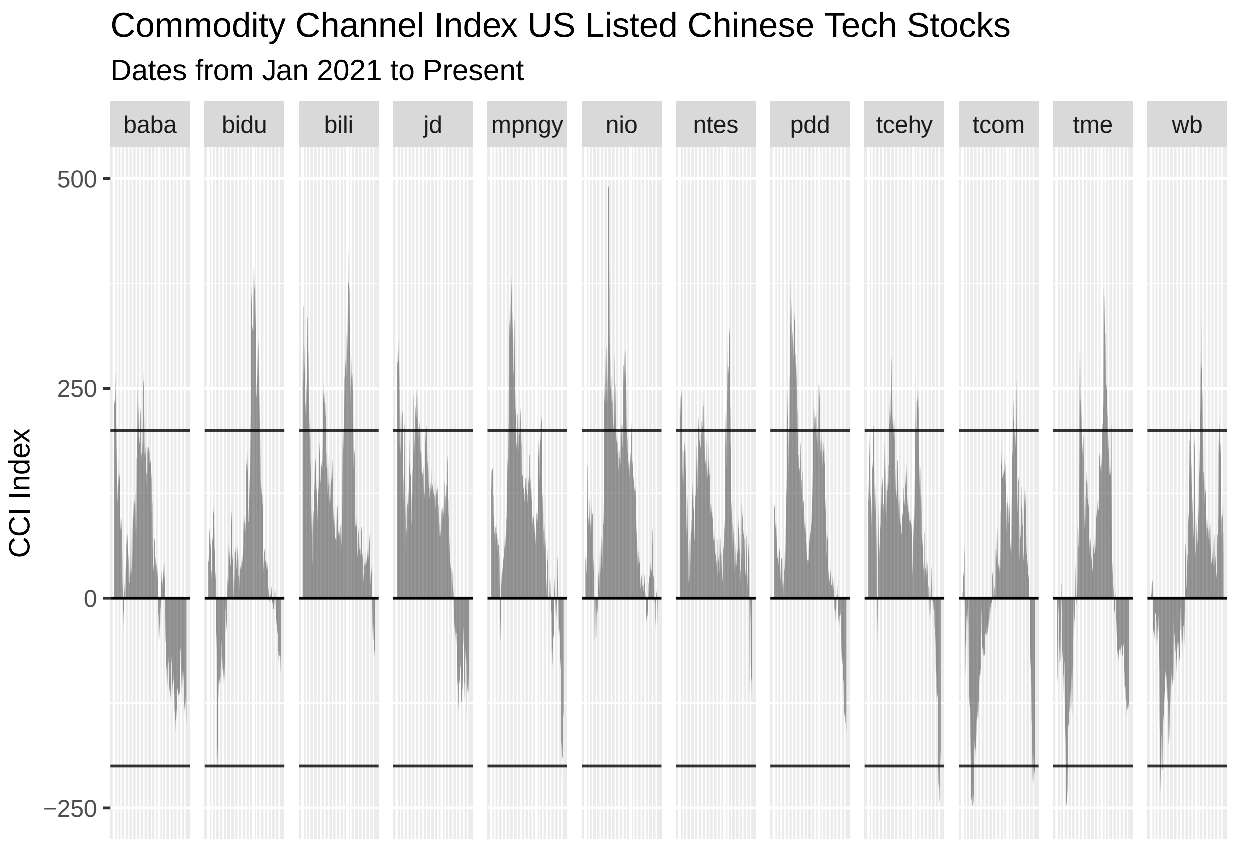
The CCI is a useful guage that is indexed. Calculating the mean and median CSI of a group of stocks can be an accurate indicator to the groups relative strength or weakness. We remove the index stock and do several calculations to come up with unique sets to use for performance analysis.

```
## 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 = "%b"), max.major.breaks=8) +
  theme(axis.title.x=element_blank()) +
  facet_wrap(. ~ symbol)
```



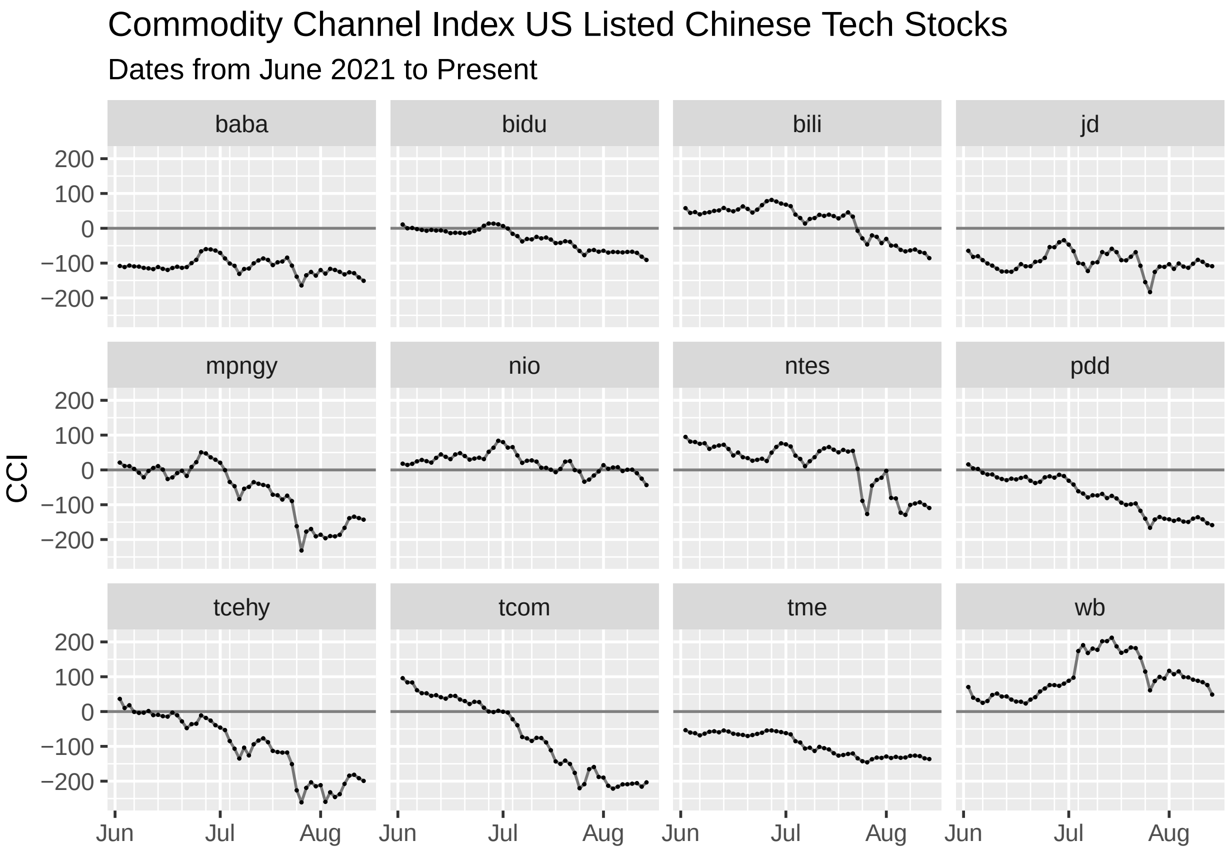
Next we look at the same data but change from facet_wrap to facet_grid. Now we can view the data lined up horizontally, which makes for better observation of highs and lows for all symbols.

```r
# Now done with facet grid instead of wrap.  Lines them up horizontally.
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 = "") +
  scale_x_bd(business.dates=time) +
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank()) +
  ylim(-250,500) +
  facet_grid(. ~ symbol)
```



Now back to the facet_wrap but zoomed in on dates to June 2021 and after.

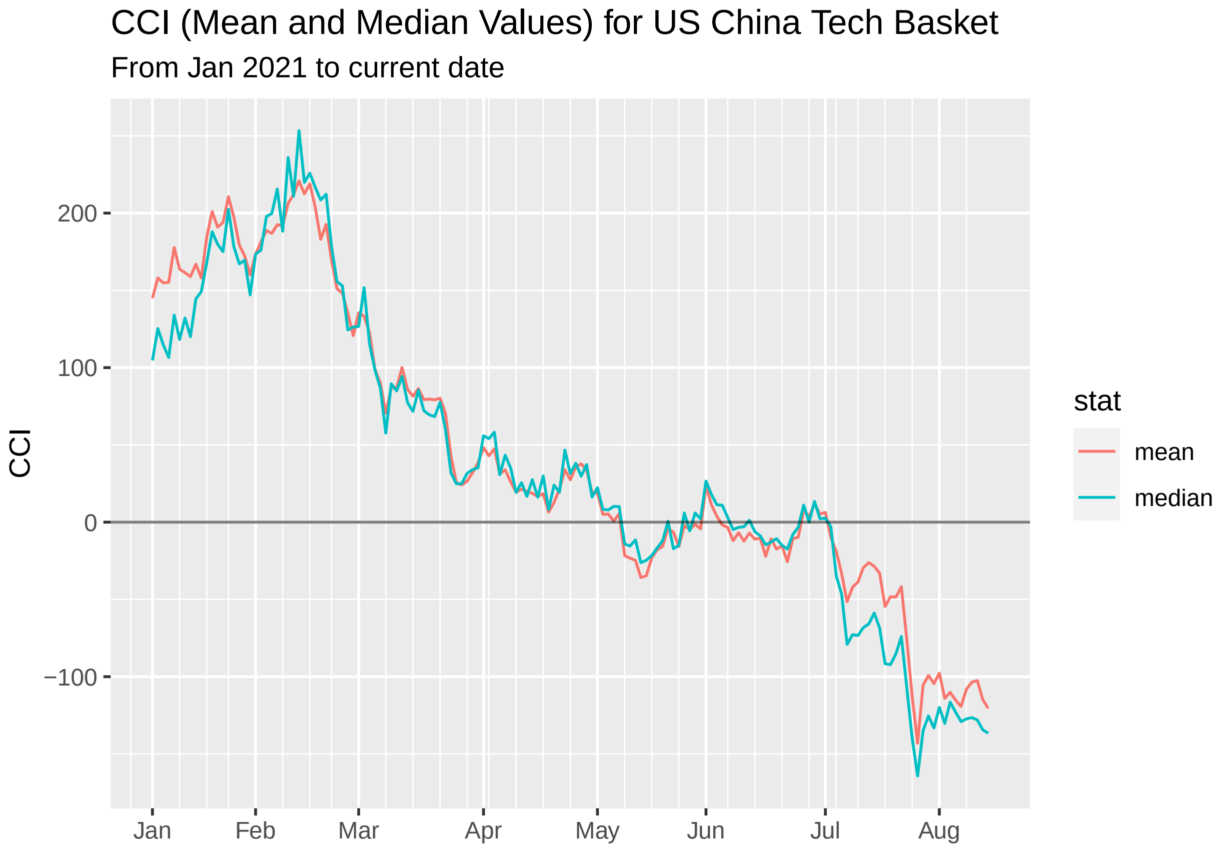
```r
# Back to a grid, introduced points.  Zoomed in view.
df.CCI %>%
  filter(time > "2021-06-01") %>%
  filter(!symbol %in% index) %>%
  ggplot(aes(time, CCI)) +
  geom_line( alpha = 0.5, show.legend = FALSE) +
  geom_hline(yintercept = 0, color = "black", alpha = 0.5) +
  geom_point(size = 0.1)+
  labs(title = "Commodity Channel Index US Listed Chinese Tech Stocks",
       subtitle = "Dates from June 2021 to Present",
       y= "CCI", x = "") +
  scale_x_bd(business.dates=time, labels = date_format(format = "%b")) +
  theme(axis.title.x=element_blank()) +
  facet_wrap(. ~ symbol)
```



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.

```
# New dataframe, filtering out the index ETF.
index.cci <- df.CCI %>%
  filter(symbol != index)
# Create a CCI number that is a median of the CCI for the non index stocks.
median.cci <- index.cci %>%
  group_by(time) %>%
  summarise(median = median(CCI, na.rm = TRUE))
# Same thing for mean
mean.cci <- index.cci %>%
  group_by(time) %>%
  summarise(mean = mean(CCI, na.rm = TRUE))
# Combine into dataframe.
stat.cci <- data.frame(median.cci, mean = mean.cci$mean)
# Remove
```

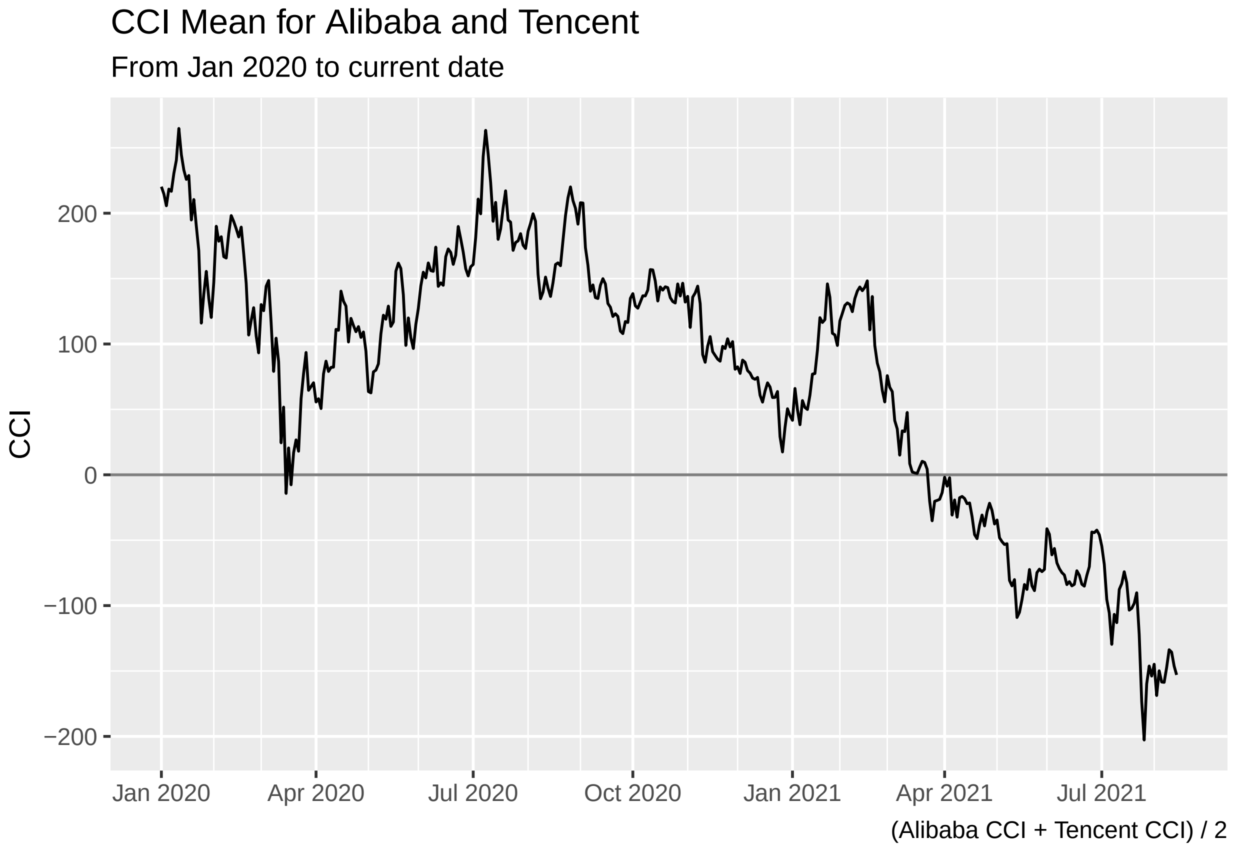
```
rm(median.cci, mean.cci)

# Gathered into a column for better graphing.
stat.cci.gather<- stat.cci %>%
  gather(key = "stat", value = "value", -time)

# Chart with date filter.  Showing Mean and Median values for the US China Tech Basket.
stat.cci.gather %>%
  filter(time > "2021-01-01") %>%
  ggplot(aes(time, value, color = stat)) +
  geom_line() +
  geom_hline(yintercept = 0, color = "black", alpha = 0.5) +
  labs(title = "CCI (Mean and Median Values) for US China Tech Basket",
       subtitle = "From Jan 2021 to current date",
       y= "CCI",
       x = " ") +
  scale_x_bd(business.dates=time, max.major.breaks=10, labels = date_format(format = "%b")) +
  theme(axis.title.x=element_blank())
```



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

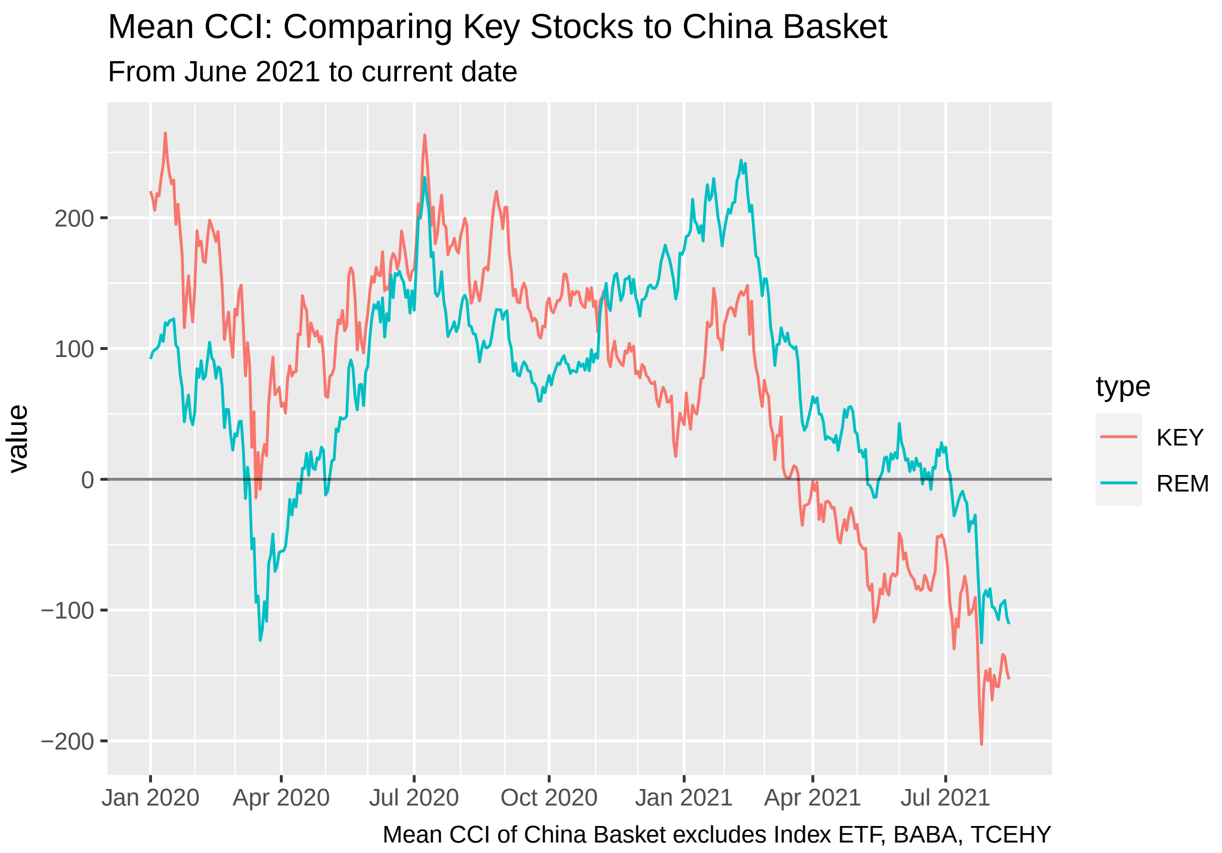
```
## Next part is isolating the key tech companies
key.cci <- df.CCI %>%
  filter(symbol %in% key.tech)
# Calculating median
```

```
key.median.cci <- key.cci %>%
  group_by(time) %>%
  summarise(median = median(CCI, na.rm = TRUE))
# Calculating median
key.mean.cci <- key.cci %>%
  group_by(time) %>%
  summarise(mean = mean(CCI, na.rm = TRUE))
# Combining
key.stat.cci <- data.frame(key.median.cci, mean = key.mean.cci$mean)
# Combining and merging into 1 column for charting.
key.stat.gather.cci <- key.stat.cci %>%
  gather(key = "stat", value = "value", -time)

# Needs work.
key.mean.cci %>%
  filter(time > "2020-01-01") %>%
  ggplot(aes(time, mean)) +
  geom_line() +
  geom_hline(yintercept = 0, color = "black", alpha = 0.5) +
  labs(title = "CCI Mean for Alibaba and Tencent",
       subtitle = "From Jan 2020 to current date",
       caption = "(Alibaba CCI + Tencent CCI) / 2",
       y= "CCI",
       x = "") +
  scale_x_bd(business.dates=time, labels = date_format(format = "%b %Y"), max.major.breaks=12) +
  theme(axis.title.x=element_blank())
```



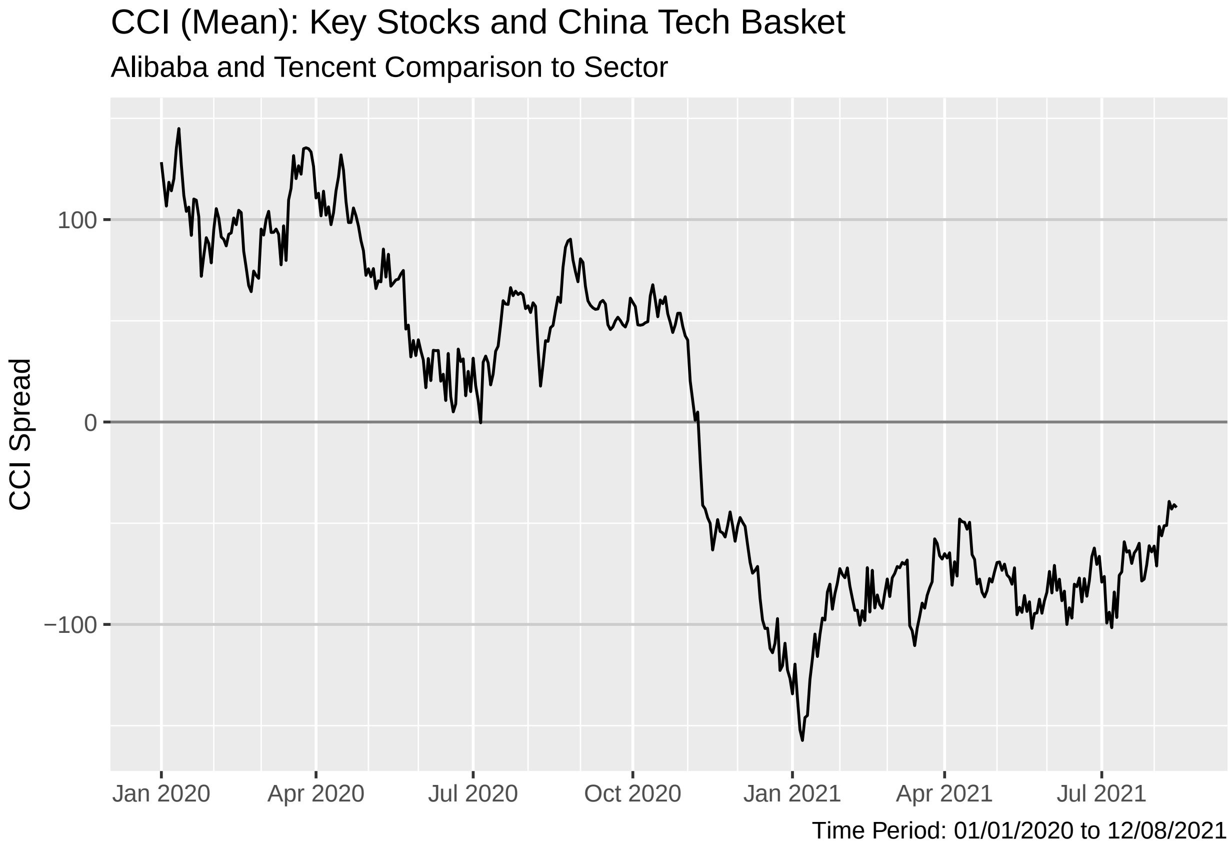
Now we will compare the mean of the key stocks to the mean of the China baskey of stocks. All based on CCI calulations.

```
index.mean.cci <- index.cci %>%
  filter(time > "2020-01-01") %>%
  filter(!symbol %in% key.tech) %>%
  group_by(time) %>%
  summarise(mean = mean(CCI, na.rm = TRUE))


comp.mean.cci <- merge(x = key.mean.cci, y = index.mean.cci, by = "time", all = TRUE) %>%
  select(time, KEY = mean.x, REM = mean.y) %>%
  gather(key = "type", value = "value", -time)

comp.mean.cci %>%
  ggplot(aes(time, value, color = type)) +
  labs(title = "Mean CCI: Comparing Key Stocks to China Basket",
       subtitle = "From June 2021 to current date",
       caption = "Mean CCI of China Basket excludes Index ETF, BABA, TCEHY") +
  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())
```



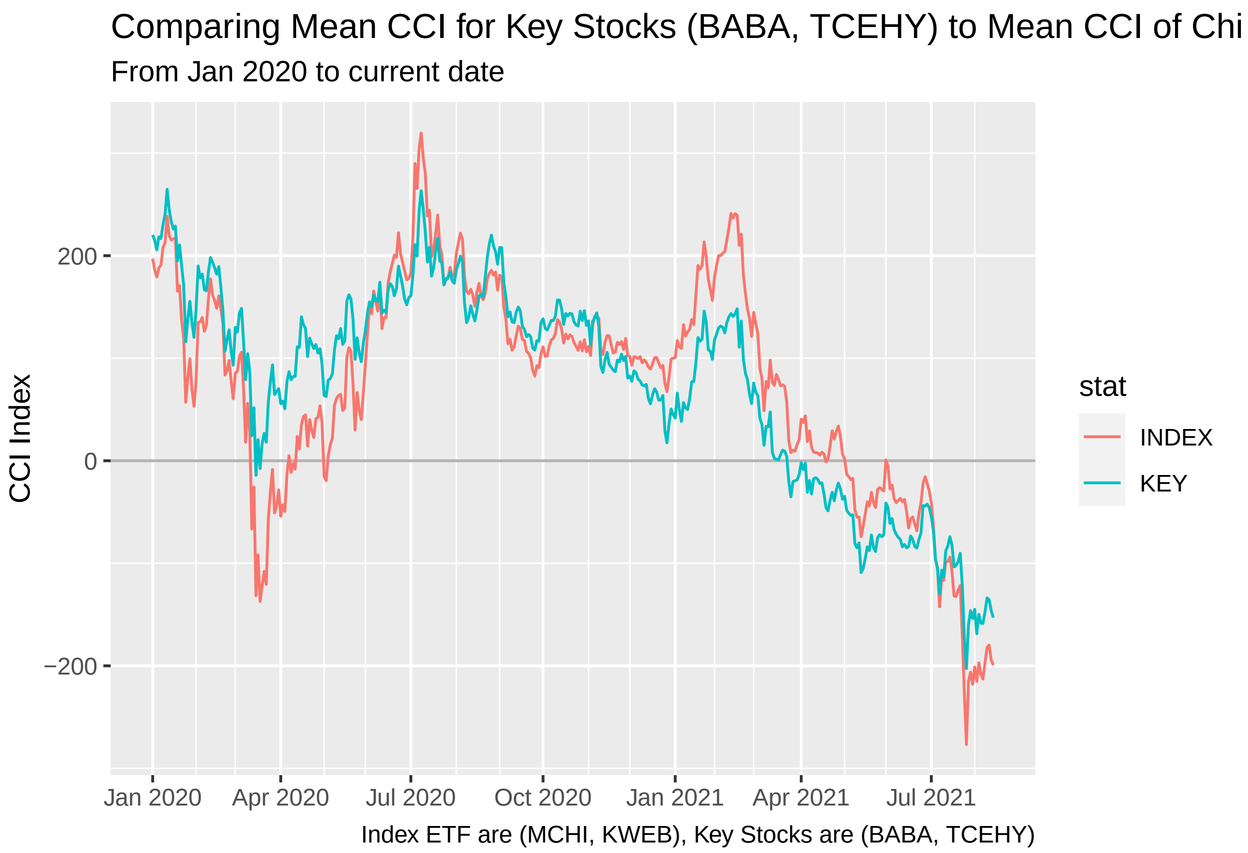
Now we will calculate the spread between the key stocks and mean. These are all still CCI measurements, this will be a spread of the CCI.

```
## WORKING OUT THE SPREAD BETWEEN THE TWO
merge(x = key.mean.cci, y = index.mean.cci, by = "time", all = TRUE) %>%
  select(time, mean.x, mean.y) %>%
  mutate(spread = mean.x - mean.y) %>%
  select(time, spread) %>%
  ggplot(aes(time, spread)) +
  geom_line() +
  labs(title = "CCI (Mean): Key Stocks and China Tech Basket",
       subtitle = "Alibaba and Tencent Comparison to Sector",
       y = "CCI Spread",
       caption = "Time Period: 01/01/2020 to 12/08/2021") +
  geom_hline(yintercept = 0, color="black", alpha = 0.5) +
  geom_hline(yintercept = 100, color="black", alpha = 0.2) +
  geom_hline(yintercept = -100, color="black", alpha = 0.2) +
  scale_x_bd(business.dates=time, labels = date_format(format = "%b %Y"), max.major.breaks=12) +
  theme(axis.title.x=element_blank())
```



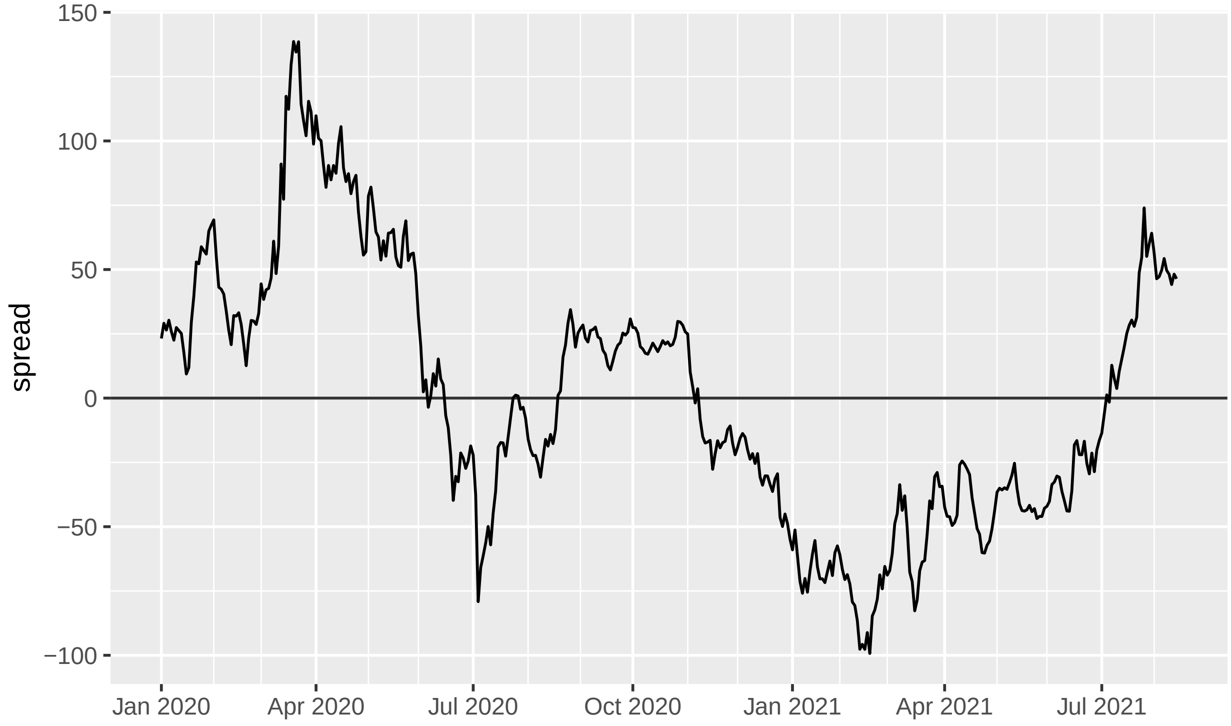
Now we will calculate the mean of the index. There are only 2 stocks in the index, KWEB and MCHI.

```
### GETTING THE MEAN OF INDEX ( MSCI AND KWEB )

idx.cci <- df.CCI %>%
  filter(symbol %in% index)

idex.mean.cci <- idx.cci %>%
  group_by(time) %>%
  summarise(mean = mean(CCI, na.rm = TRUE))

idx.stat.cci <- data.frame(time = key.mean.cci$time, KEY = key.mean.cci$mean, INDEX = idex.mean.cci$mea
  gather(key = "stat", value = "value", -time)

# PLOT

idx.stat.cci %>%
  ggplot(aes(time, value, color = stat)) +
  geom_hline(yintercept = 0, color="black", alpha = 0.3) +
  labs(title = "Comparing Mean CCI for Key Stocks (BABA, TCEHY) to Mean CCI of China Index ETF",
       subtitle = "From Jan 2020 to current date",
       caption = "Index ETF are (MCHI, KWEB), Key Stocks are (BABA, TCEHY)",
       y = "CCI Index") +
  geom_line() +
  scale_x_bd(business.dates=time, labels = date_format(format = "%b %Y"), max.major.breaks=12) +
  theme(axis.title.x=element_blank())
```



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 = "Spread of Mean CCI for Key Stocks (BABA, TCEHY) less Mean CCI of China Index",
       caption = "China index ETF is mean of KWEB and MCHI CCI") +
  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())
```

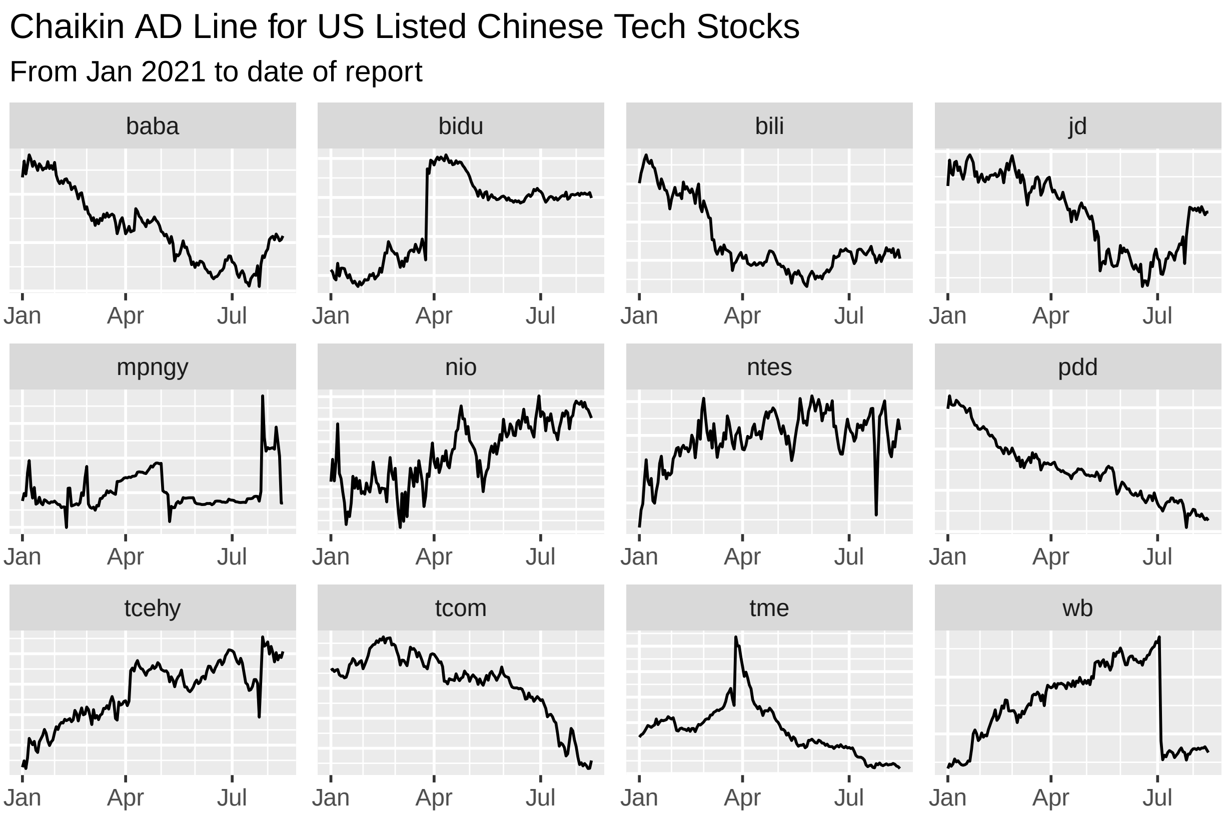

CCI: Key Stocks Spread to Index ETF

Spread of Mean CCI for Key Stocks (BABA, TCEHY) less Mean CCI of China Index

China index ETF is mean of KWEB and MCHI CCI

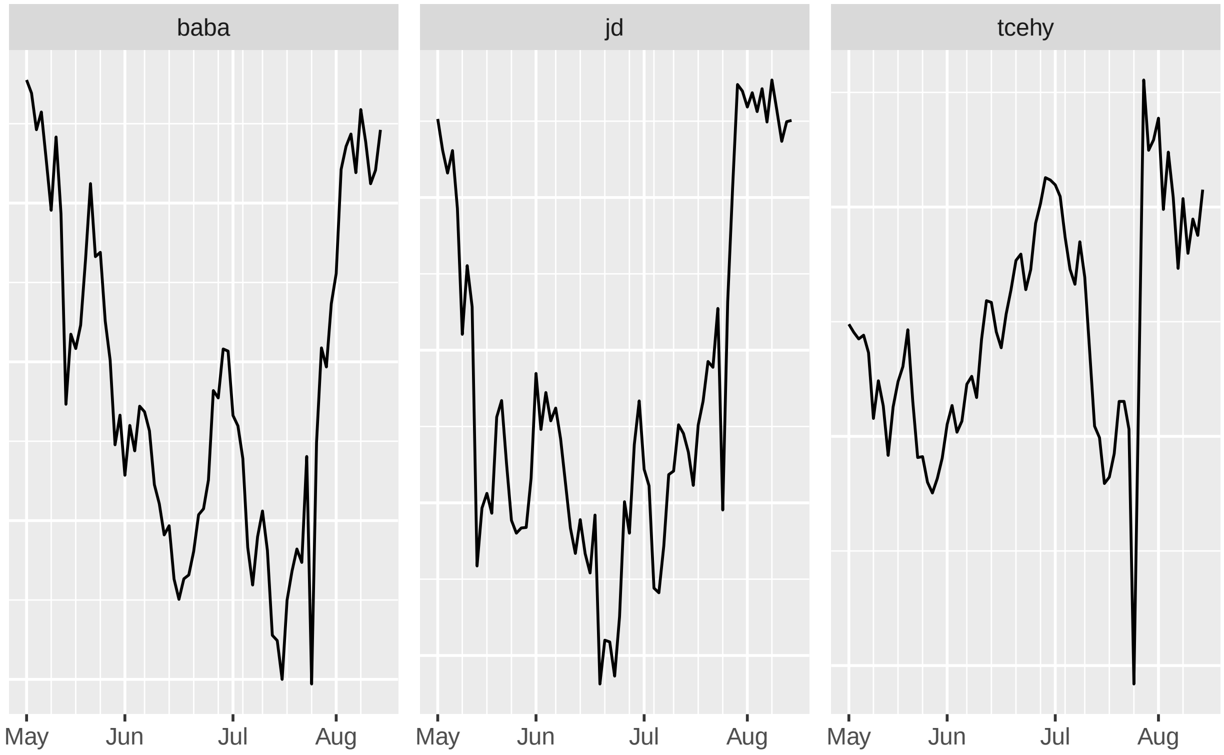
Chaikin AD Line

```
## AD CHARTINGS
df.AD %>%
  filter(time > chart.date) %>%
  filter(!symbol %in% index) %>%
  ggplot(aes(time, AD)) +
  geom_line() +
  labs(title = "Chaikin AD Line for US Listed Chinese Tech Stocks",
       subtitle = "From Jan 2021 to date of report",
       caption = "Data is not useful for NTES.",
       y= "AD", x = "") +
  facet_wrap(. ~ symbol, scale = "free") +
  scale_x_bd(business.dates=time, labels = date_format(format = "%b"), max.major.breaks=6) +
  theme(axis.title.y=element_blank(),
      axis.text.y=element_blank(),
      axis.ticks.y=element_blank(),
      axis.title.x=element_blank())
```



Data is not useful for NTES.

```
df.AD %>%
  filter(time > "2021-05-01") %>%
  filter(symbol %in% key.tech) %>%
  ggplot(aes(time, AD)) +
  geom_line() +
  labs(title = "Chaikin AD Line for Key US Listed Chinese Tech",
       subtitle = "From May 2021 to Date of Report",
       y= "AD", x = "") +
  facet_wrap(. ~ symbol, scale = "free") +
  scale_x_bd(business.dates=time, labels = date_format(format = "%b"), max.major.breaks=6) +
  theme(axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        axis.title.x=element_blank())
```



Chaikin AD Line for Key US Listed Chinese Tech

From May 2021 to Date of Report

baba

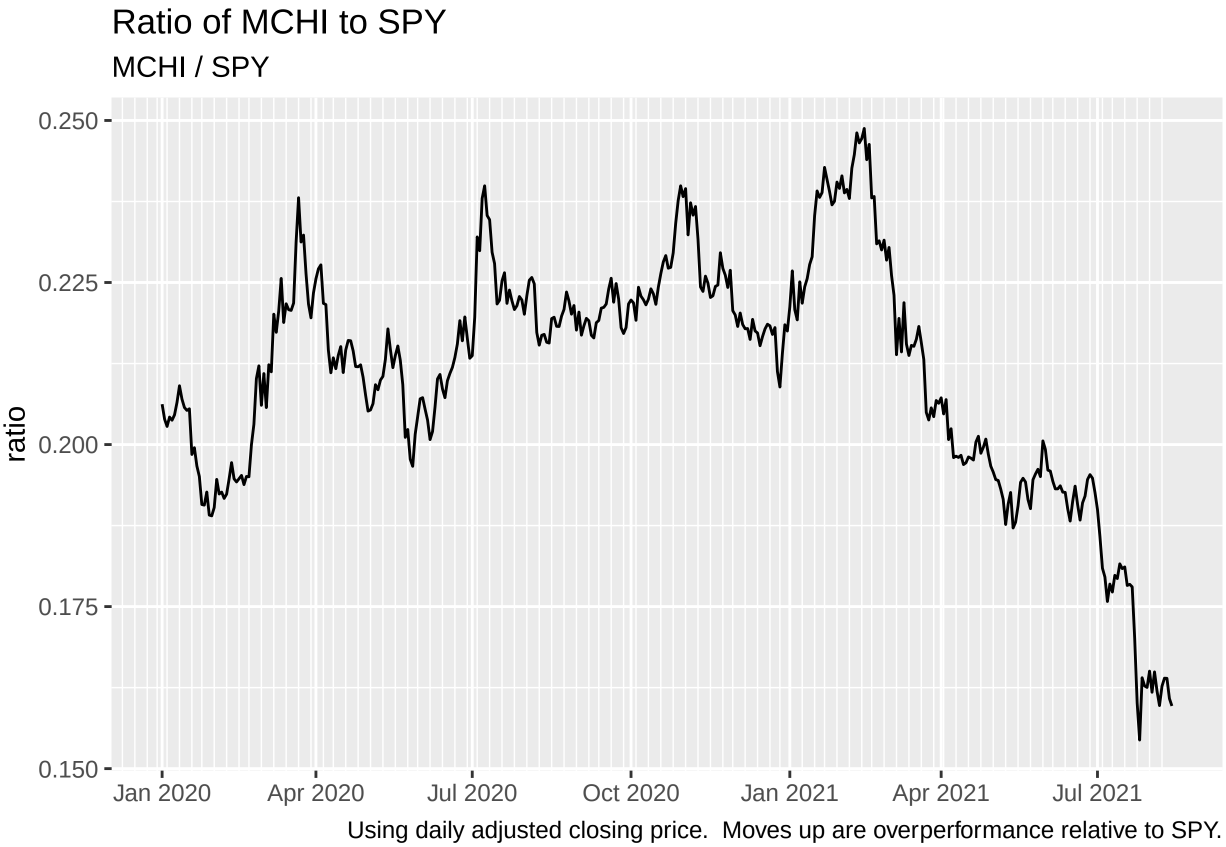
jd

tcehy

Stock Price Comparison

Comparing stock prices using Yahoo finance daily adjusted close data.

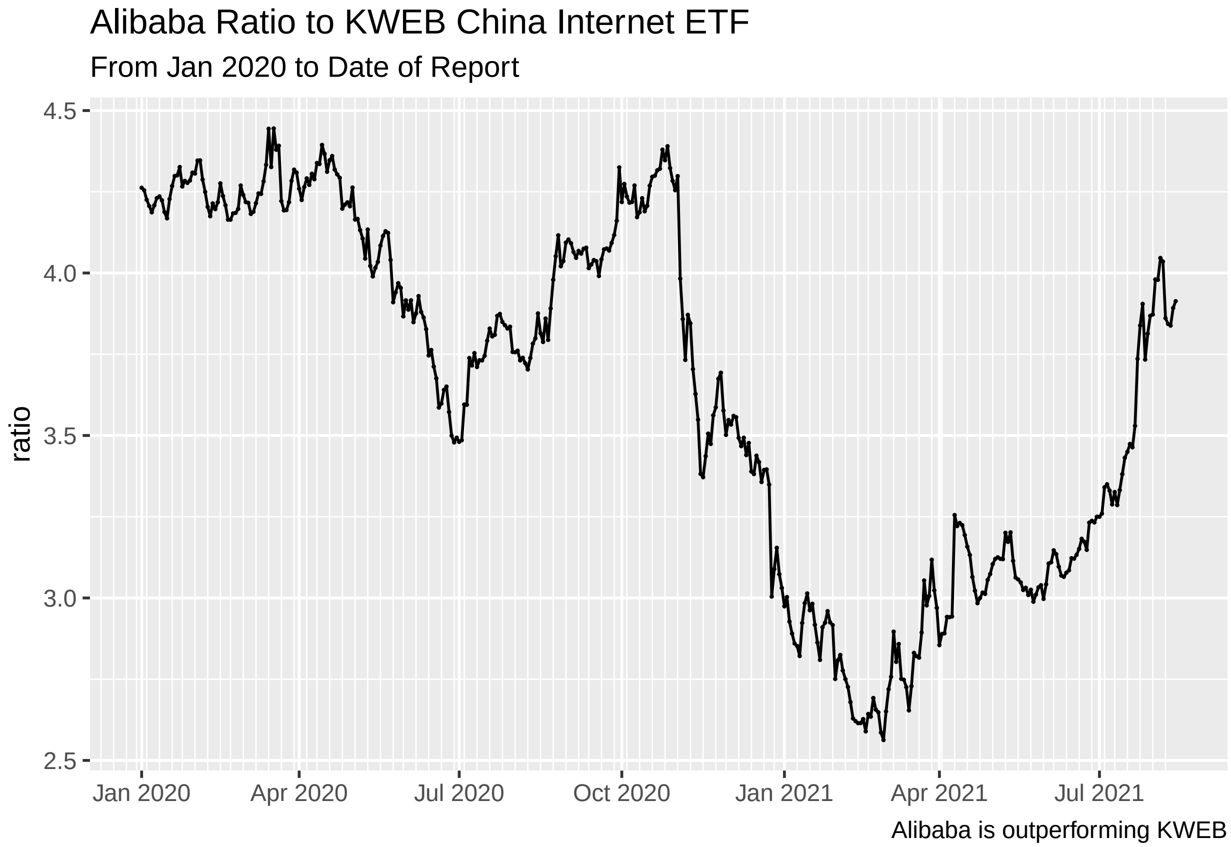
```
mchi <- stock.data %>%
  filter(date > "2020-01-01")%>%
  filter(symbol == "MCHI") %>%
  select(mchi = adjusted)
spy <- stock.data %>%
  filter(date > "2020-01-01")%>%
  filter(symbol == "SPY")
comb <- data.frame(spy, mchi)
rm(spy, mchi)

comb %>%
  mutate(ratio = mchi/adjusted) %>%
  select(date, ratio) %>%
  ggplot(aes(date, ratio)) +
  labs(title = "Ratio of MCHI to SPY",
       subtitle = "MCHI / SPY",
       caption = "Using daily adjusted closing price.  Moves up are overperformance relative to SPY.")
  geom_line() +
  scale_x_bd(business.dates=time, labels = date_format(format = "%b %Y"), max.major.breaks=20) +
  theme(axis.title.x=element_blank())
```


Ratio of MCHI to SPY

MCHI / SPY

Using daily adjusted closing price. Moves up are overperformance relative to SPY.

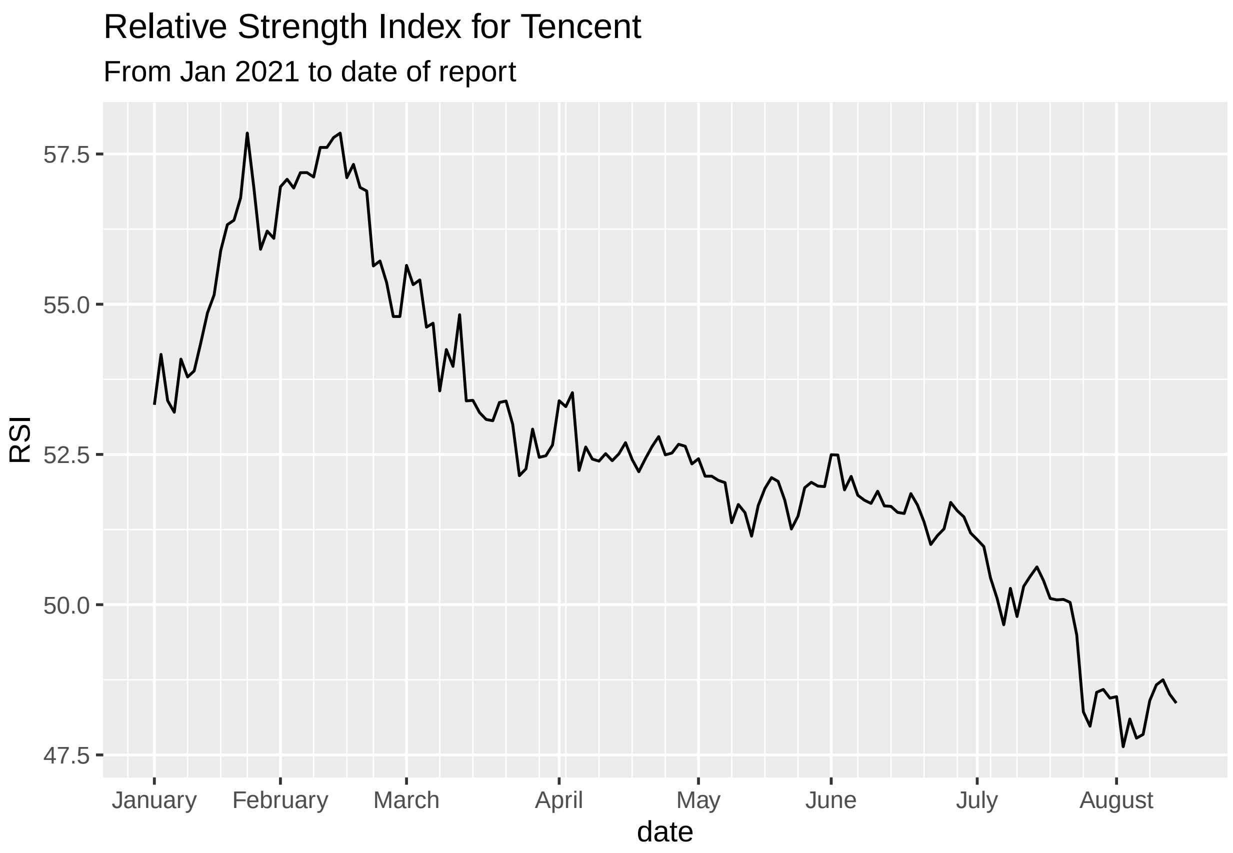
```
baba <- stock.data %>%
  filter(date > "2020-01-01")%>%
  filter(symbol == "BABA") %>%
  select(baba = adjusted)
kweb <- stock.data %>%
  filter(date > "2020-01-01")%>%
  filter(symbol == "KWEB")
comb <- data.frame(baba, kweb)
rm(baba, kweb)


comb %>%
  mutate(ratio = baba/adjusted) %>%
  select(date, ratio) %>%
  filter(date > "2020-01-01") %>%
  ggplot(aes(date, ratio)) +
  labs(title = "Alibaba Ratio to KWEB China Internet ETF",
       subtitle = "From Jan 2020 to Date of Report",
       caption = "Alibaba is outperforming KWEB") +
  geom_line() +
  geom_point(size=0.1) +
  scale_x_bd(business.dates=time, labels = date_format(format = "%b %Y"), max.major.breaks=20) +
  theme(axis.title.x=element_blank())
```



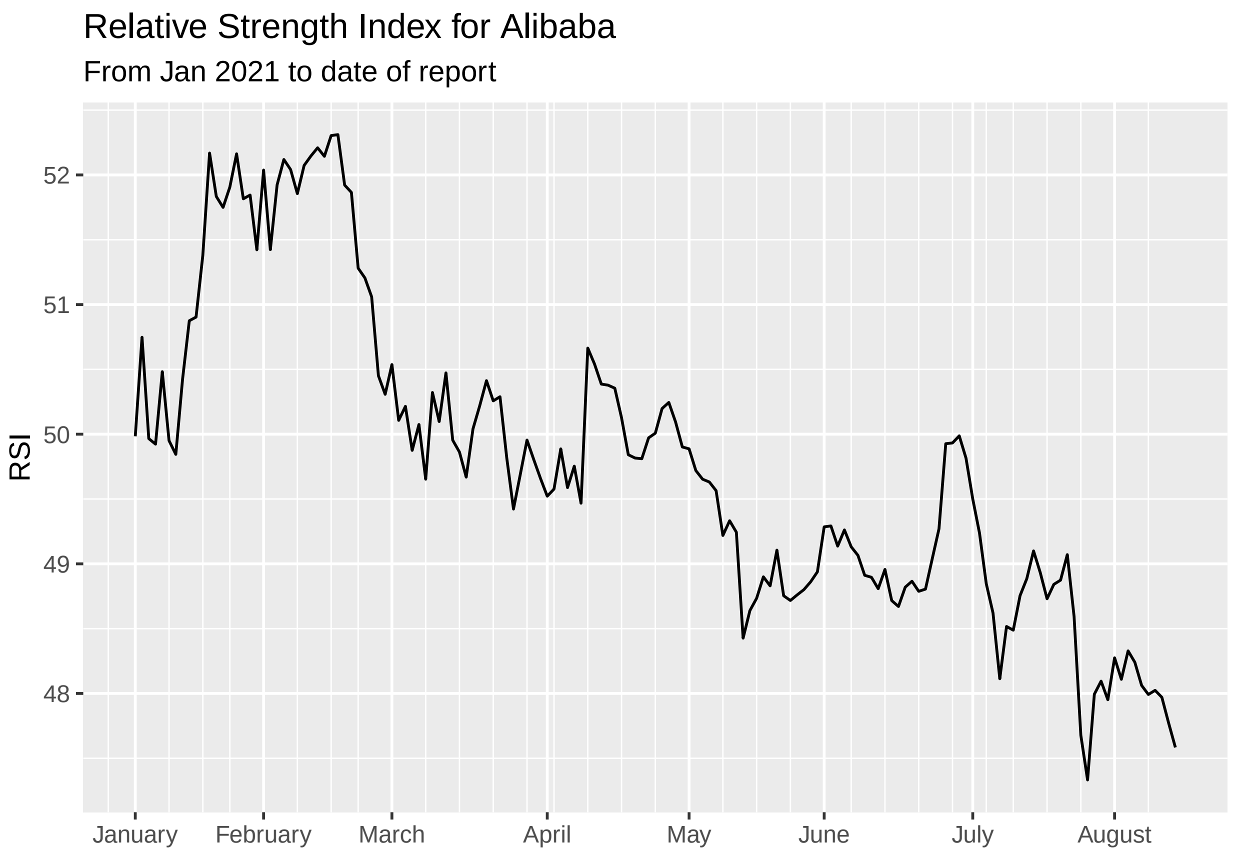
Relative Strength Index

This is an indexed calculation. Can be useful to compare the RSI scores for different symbols. First we plot TCEHY and BABA.

```
#### RSI

df.RSI %>%
  filter(time >"2021-01-01") %>%
  filter(symbol %in% "tcehy") %>%
  ggplot(aes(time, RSI)) +
  labs(title = "Relative Strength Index for Tencent",
       subtitle = "From Jan 2021 to date of report") +
  geom_line() +
  scale_x_bd(business.dates=time, labels = date_format(format = "%B"), max.major.breaks=12)
```

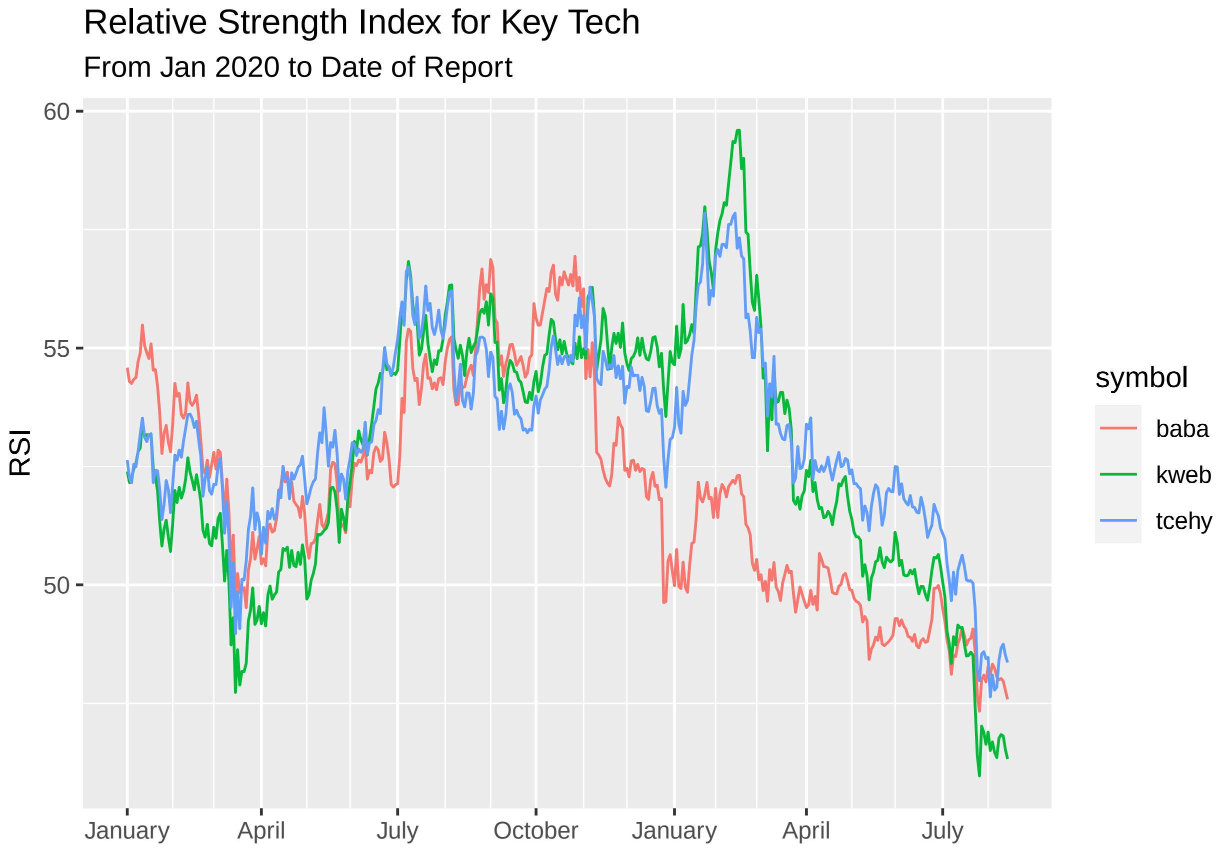


```
df.RSI %>%
  filter(time >"2021-01-01") %>%
  filter(symbol %in% "baba") %>%
  ggplot(aes(time, RSI)) +
  labs(title = "Relative Strength Index for Alibaba",
       subtitle = "From Jan 2021 to date of report") +
  geom_line() +
  scale_x_bd(business.dates=time, labels = date_format(format = "%B"), max.major.breaks=12) +
  theme(axis.title.x=element_blank())
```



Next we combine Tencent, Alibaba, and JD.

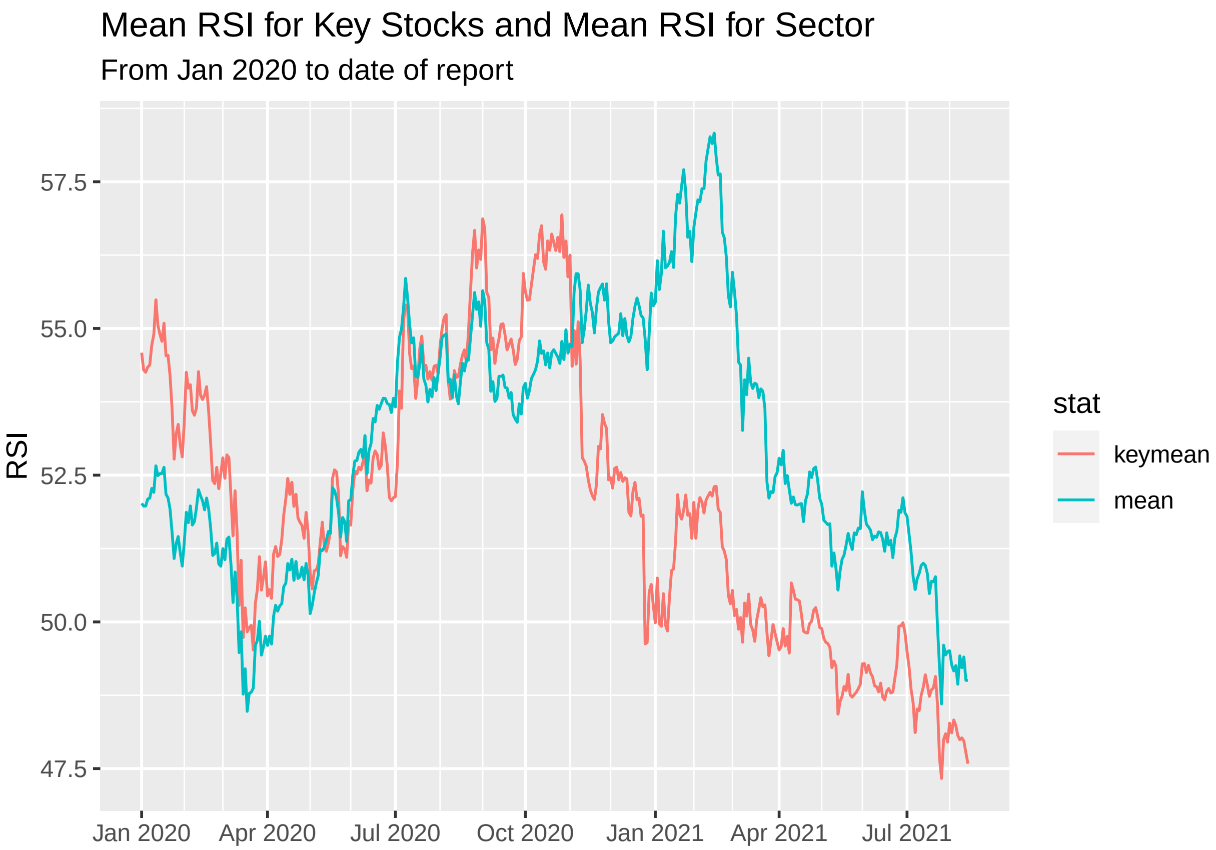
```
temp.select <- c("kweb", "baba", "tcehy")

df.RSI %>%
  filter(time >"2020-01-01") %>%
  filter(symbol %in% temp.select) %>%
  ggplot(aes(time, RSI, color = symbol)) +
  labs(title = "Relative Strength Index for Key Tech",
       subtitle = "From Jan 2020 to Date of Report") +
  geom_line() +
  scale_x_bd(business.dates=time, labels = date_format(format = "%B"), max.major.breaks=12) +
  theme(axis.title.x=element_blank())
```



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.

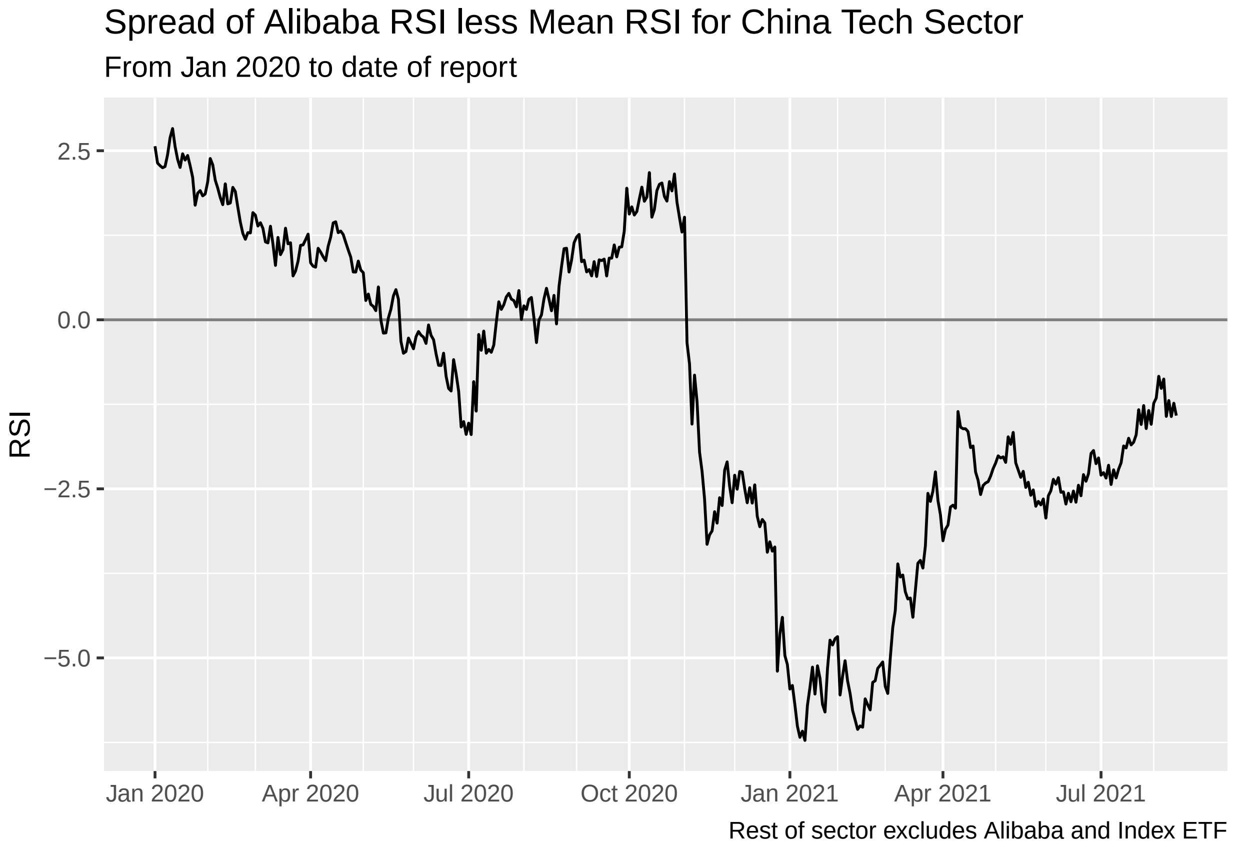
```
### Recent Edits

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())
```



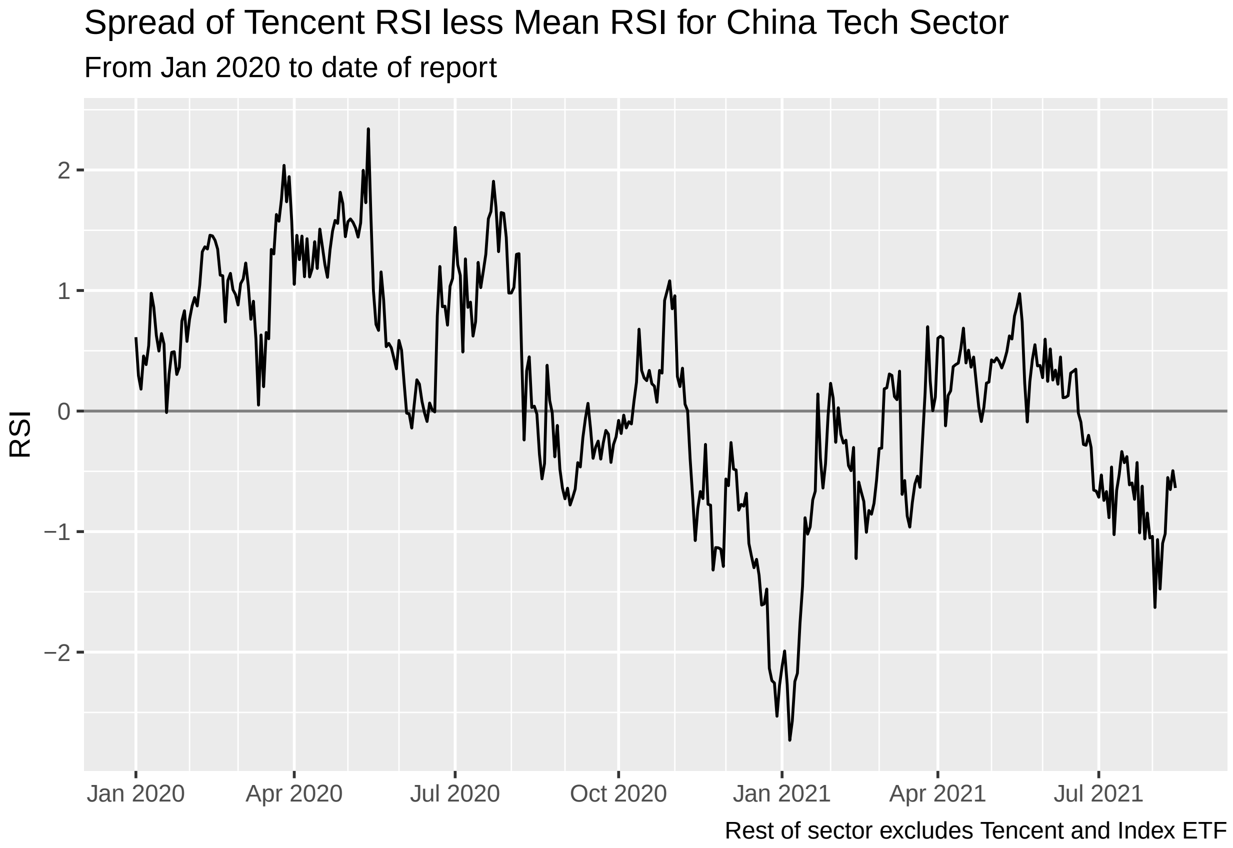
Next we will create the spread column and chart it. Taking the two values from the previous chart, subtracting them, and plotting the spread.

```
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())
```



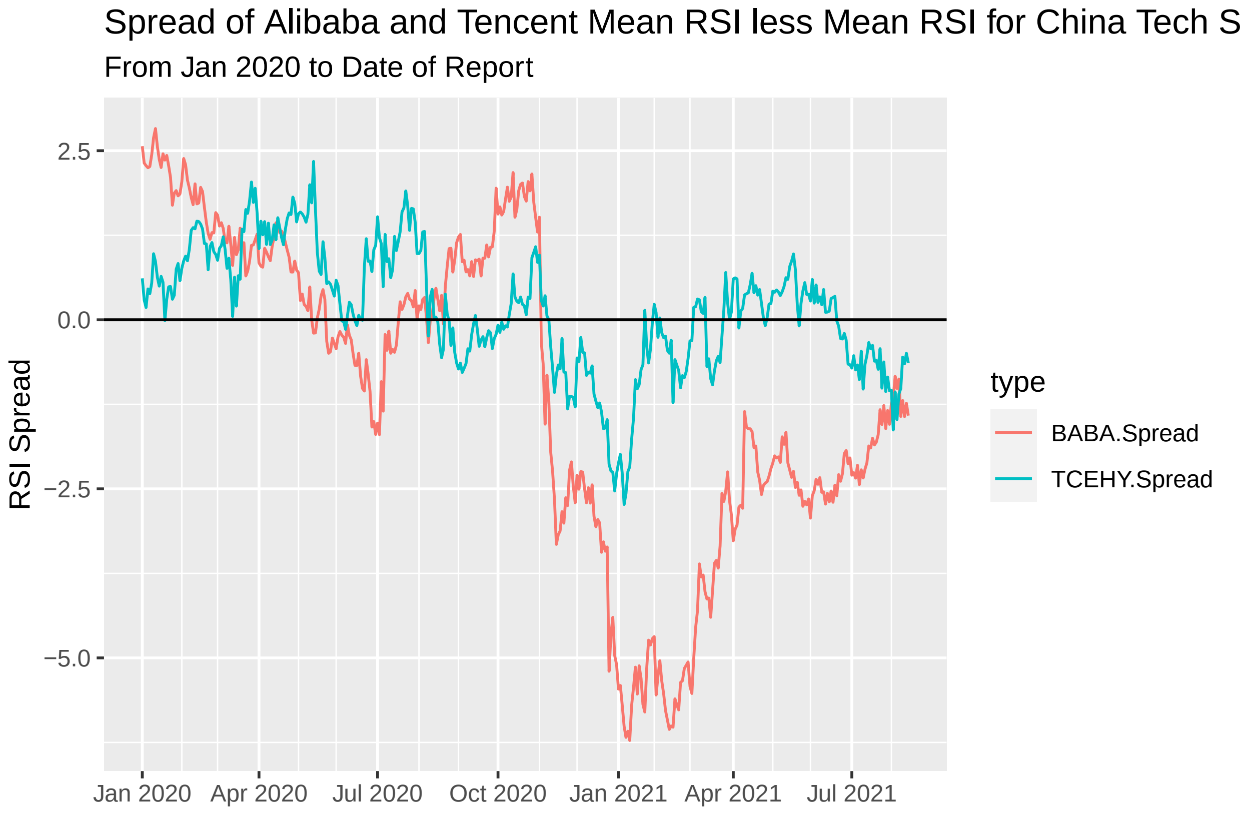
This next chunk will run the same calculation as the prior, but with Tencent instead. We will plot the spread of Tencents RSI spread to the China basket excluding Tencent.

```
mean.rsi <- df.RSI %>%
  filter(symbol != index) %>%
  filter(!symbol %in% "TCEHY") %>%
  group_by(time) %>%
  summarise(mean = mean(RSI, na.rm = TRUE))
key.rsi <- df.RSI %>%
  filter(symbol %in% "tcehy") %>%
  group_by(time) %>%
  summarise(mean = mean(RSI, na.rm = TRUE)) %>%
  select(keymean = mean)
stat.rsi <- data.frame(mean.rsi, key.rsi) %>%
  gather(key = "stat", value = "value", -time)

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.

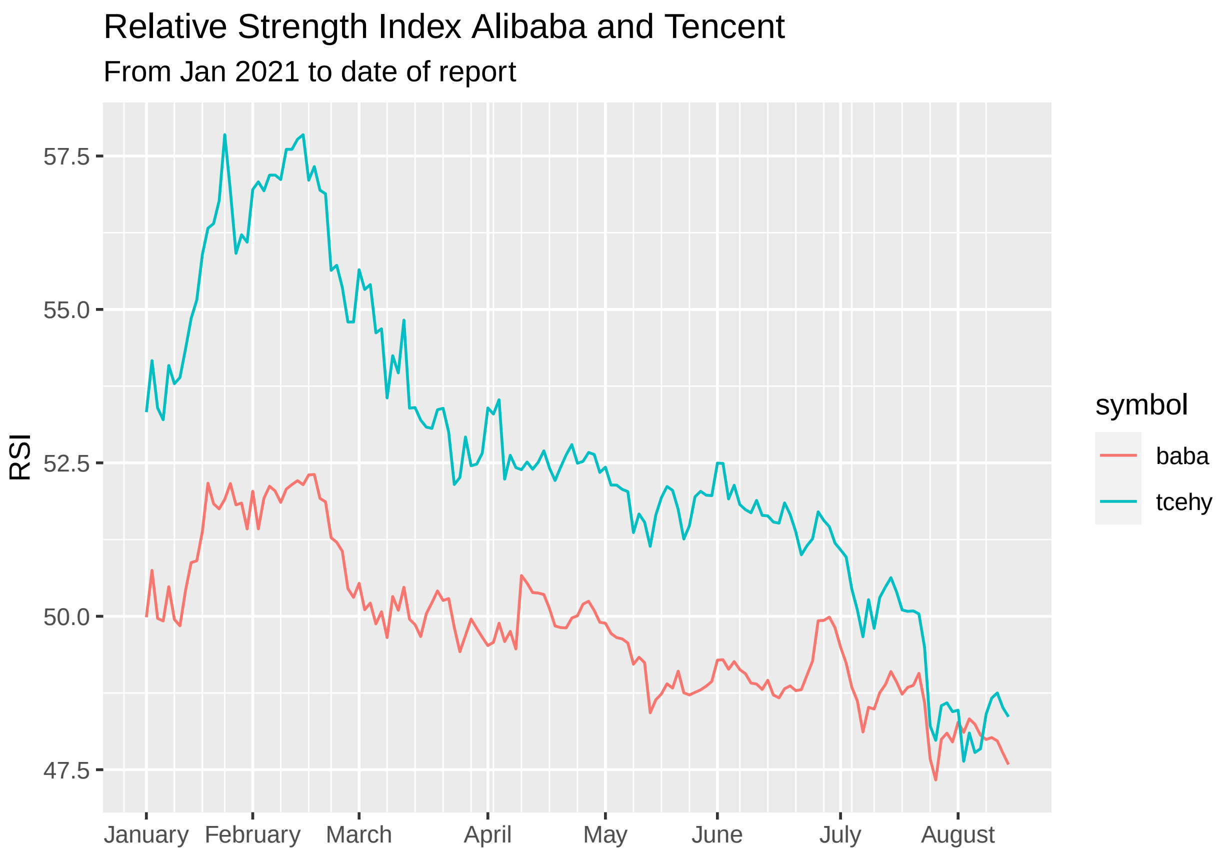
```
tcehy.spread.rsi <- data.frame(mean.rsi, key.rsi) %>%
  mutate(spread = keymean - mean) %>%
  select(time, TCEHY.Spread = spread)

stat.comp.rsi <- data.frame(tcehy.spread.rsi, BABA.Spread = baba.spread.rsi$BABA.Spread) %>%
  gather(key = "type", value = "value", -time)

stat.comp.rsi %>%
ggplot(aes(time, value, color = type)) +
  labs(title = "Spread of Alibaba and Tencent Mean RSI less Mean RSI for China Tech Sector",
       subtitle = "From Jan 2020 to Date of Report",
       y = "RSI Spread",
       caption = " ") +
  geom_line() +
  geom_hline(yintercept = 0, color = "black") +
  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())
```


Spread of Alibaba and Tencent Mean RSI less Mean RSI for China Tech S

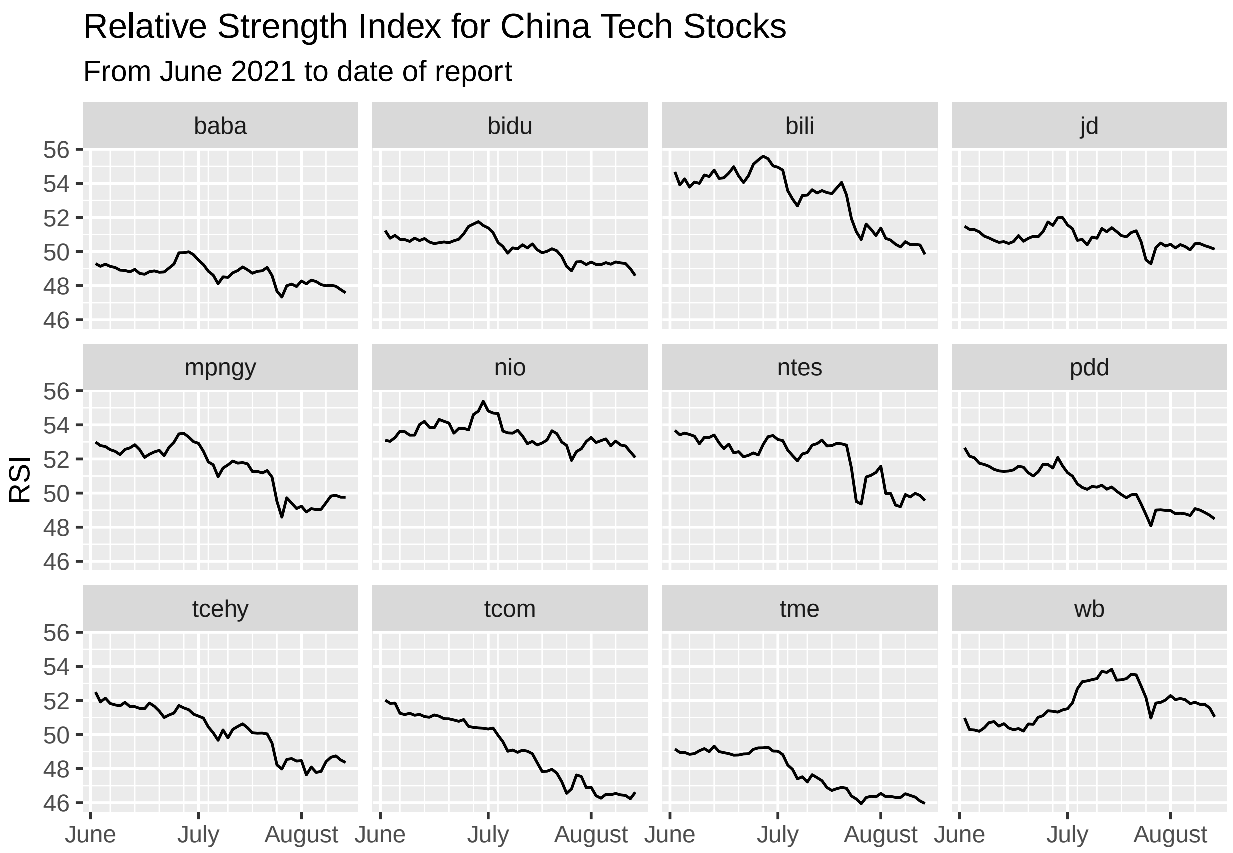
From Jan 2020 to Date of Report

Looking now at the RSI for Alibaba and Tencent from Jan 2021 and then June 2021 forward.

```
df.RSI %>%
  filter(time >"2021-01-01") %>%
  filter(symbol %in% key.tech) %>%
  filter(!symbol %in% "jd") %>%
  ggplot(aes(time, RSI, color = symbol)) +
  labs(title = "Relative Strength Index Alibaba and Tencent",
       subtitle = "From Jan 2021 to date of report") +
  geom_line() +
  scale_x_bd(business.dates=time, labels = date_format(format = "%B"), max.major.breaks=12) +
  theme(axis.title.x=element_blank())
```



```
df.RSI %>%
  filter(time >"2021-06-01") %>%
  filter(!symbol %in% index) %>%
  ggplot(aes(time, RSI)) +
  geom_line() +
  labs(title = "Relative Strength Index for China Tech Stocks",
       subtitle = "From June 2021 to date of report") +
  facet_wrap(. ~ symbol) +
  scale_x_bd(business.dates=time, labels = date_format(format = "%B"), max.major.breaks=5) +
  theme(axis.title.x=element_blank())
```



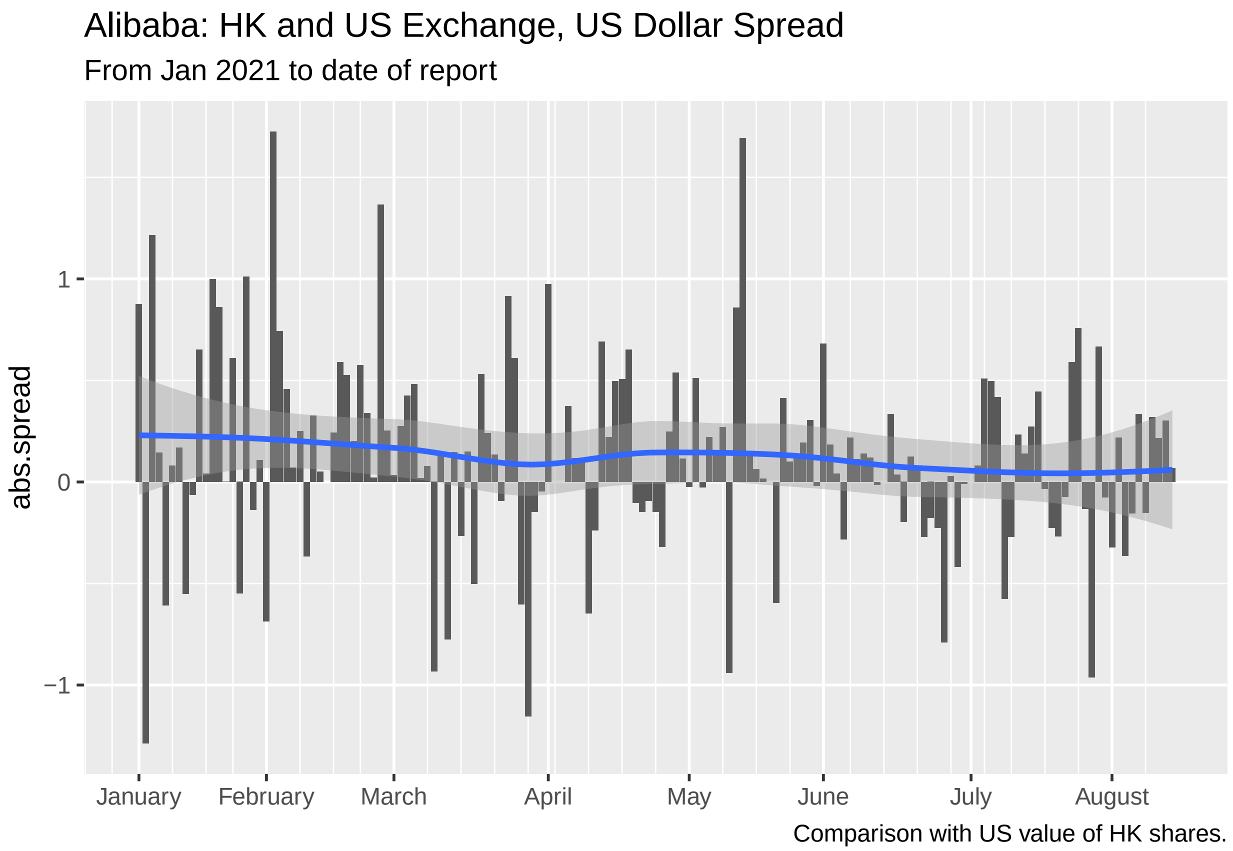
Exchange Spreads

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)
```

```
spread.sl.hk %>%
  ggplot(aes(date, abs.spread)) +
    geom_col() +
    geom_smooth() +
  labs(title = "Alibaba: HK and US Exchange, US Dollar Spread",
       subtitle = "From Jan 2021 to date of report",
       caption = "Comparison with US value of HK shares.") +
  scale_x_bd(business.dates=time, labels = date_format(format = "%B"), max.major.breaks=12) +
  theme(axis.title.x=element_blank())
```

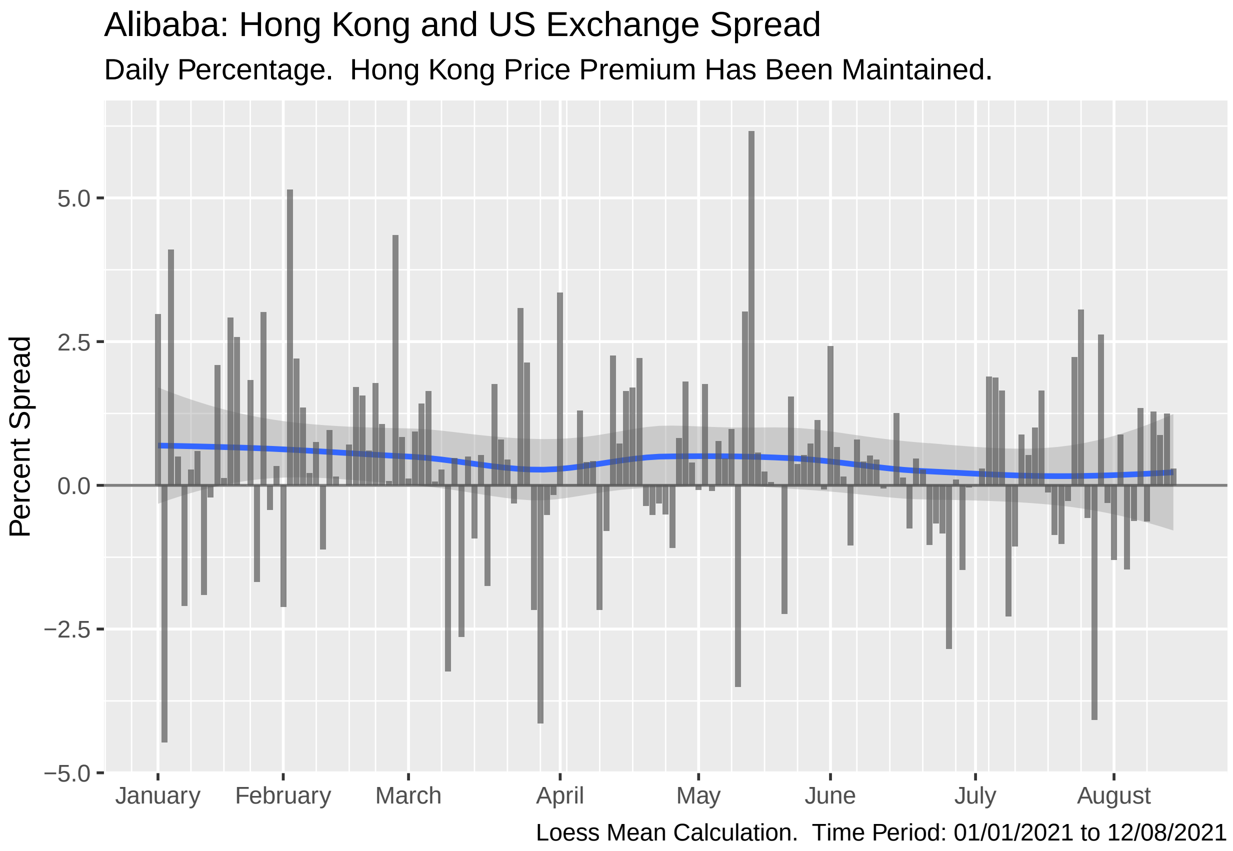
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



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())
```

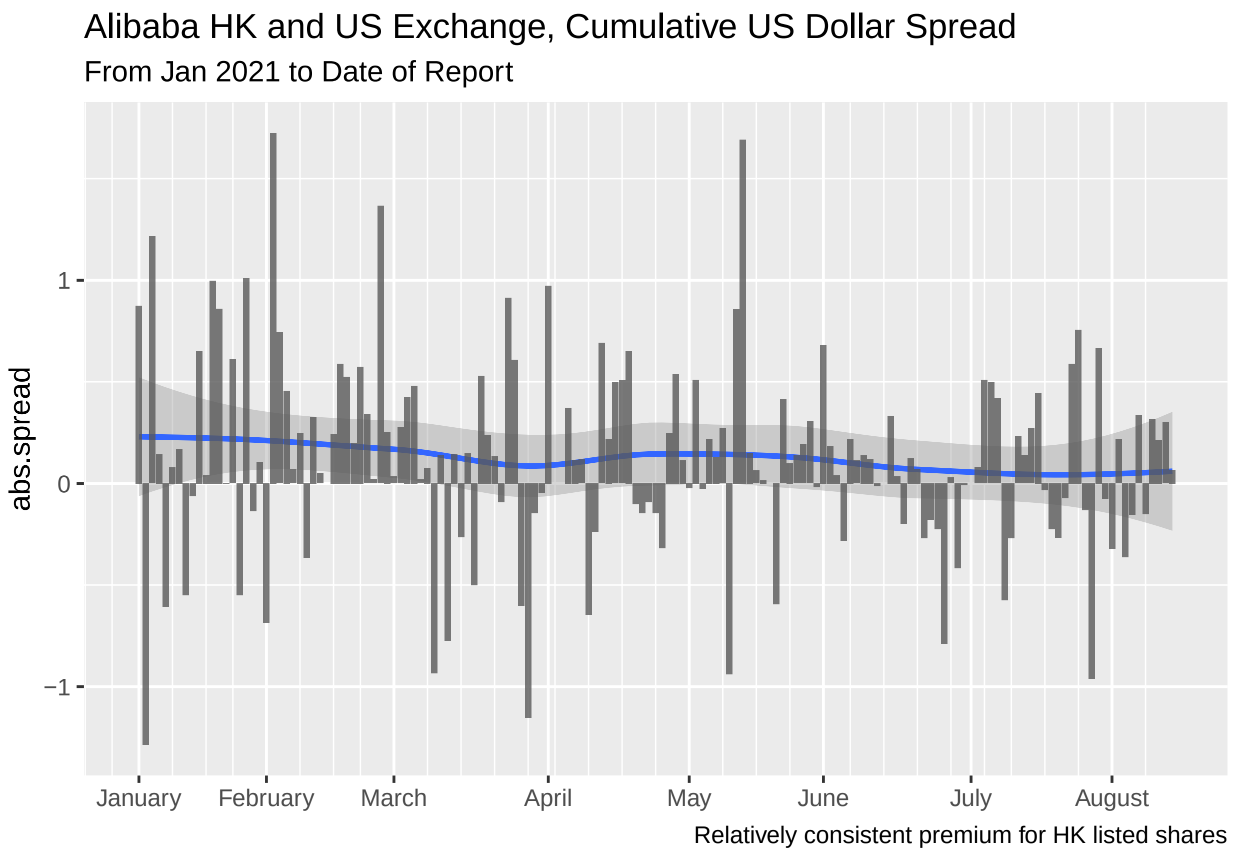
```
## `geom_smooth()` using formula 'y ~ x'
```



Lastly we will look at this exchange spread on a cumulative US dollar basis. Meaning the spread is carried through and added and subtracted from to form a rolling number.

```
spread.sl.hk.copy %>%
  group_by(date) %>%
  mutate(A_cum_sum = cumsum(abs.spread)) %>%
  ggplot(aes(date, abs.spread)) +
  labs(title = "Alibaba HK and US Exchange, Cumulative US Dollar Spread",
       subtitle = "From Jan 2021 to Date of Report",
       caption = "Relatively consistent premium for HK listed shares") +
  geom_smooth() +
  geom_col(alpha = 0.8) +
  scale_x_bd(business.dates=time, labels = date_format(format = "%B"), max.major.breaks=12) +
  theme(axis.title.x=element_blank())
```

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


Alibaba HK and US Exchange, Cumulative US Dollar Spread

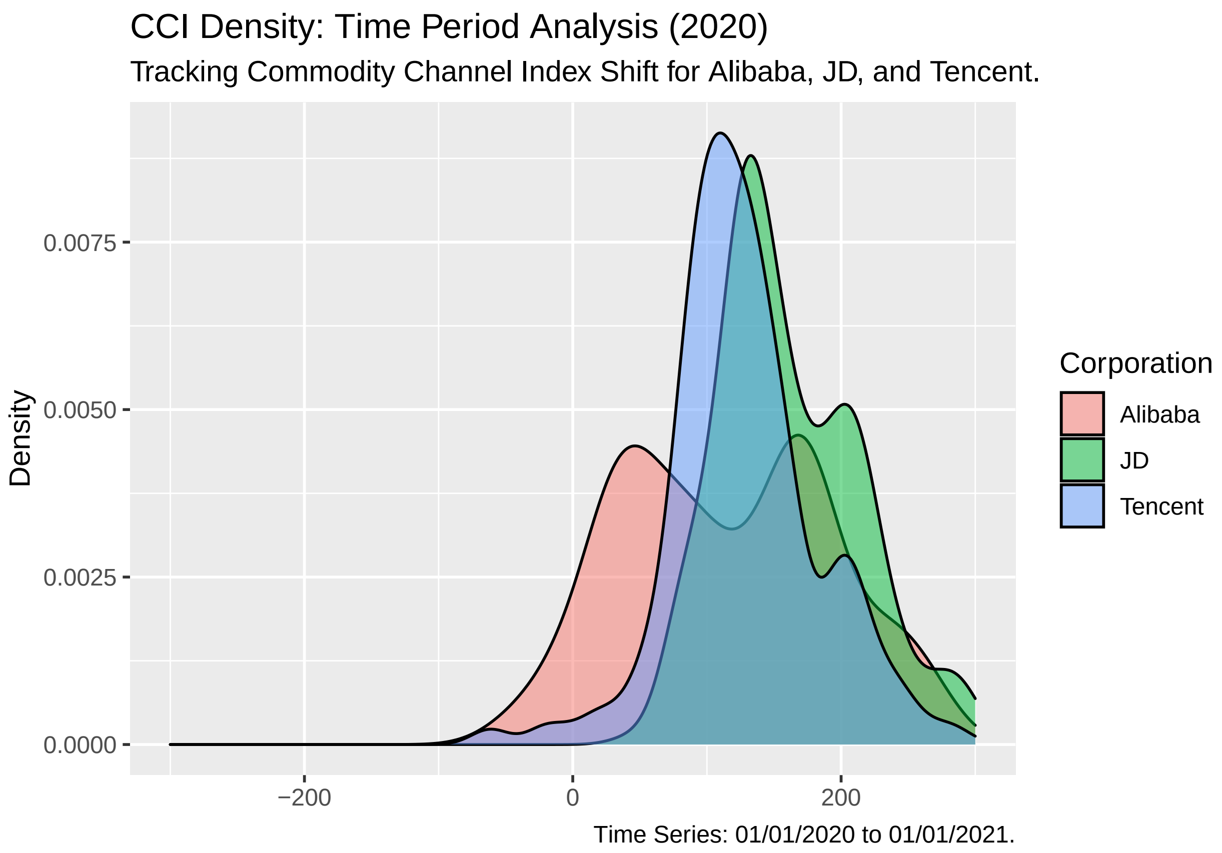
From Jan 2021 to Date of Report

Relatively consistent premium for HK listed shares

Density Plot Shirt

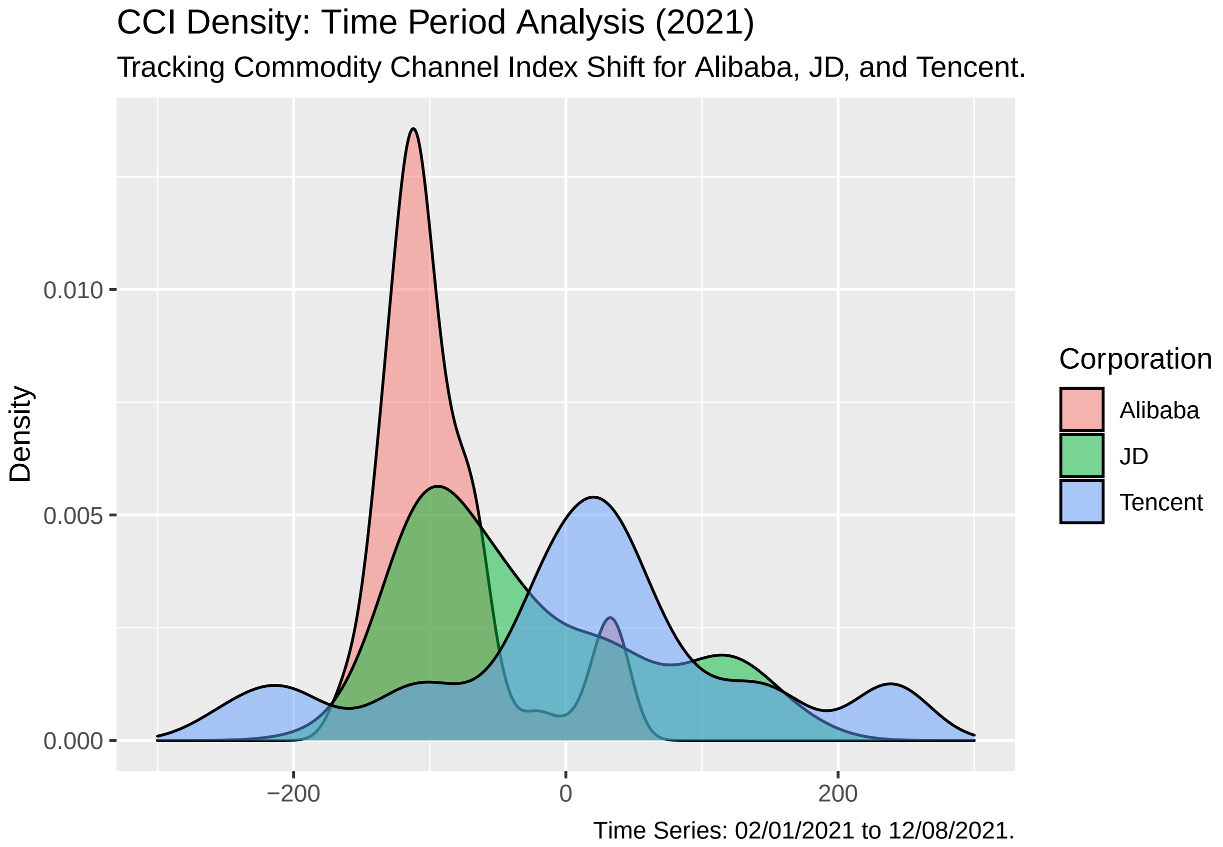
Lastly, we will come back to the CCI data for Alibaba, Tencent, and JD. We will review the density of CCI results for 2 time periods. The first is Jan 1, 2020 to Jan 1, 2021 which is presented below.

```
df.CCI$symbol[df.CCI$symbol == "jd"] <- "JD"
df.CCI$symbol[df.CCI$symbol == "baba"] <- "Alibaba"
df.CCI$symbol[df.CCI$symbol == "tcehy"] <- "Tencent"

new.key.tech <- c("Alibaba", "JD", "Tencent")

df.CCI %>%
  select(time, Corporation = symbol, CCI) %>%
  filter(time >"2020-01-01") %>%
  filter(time < "2021-01-01") %>%
  filter(Corporation %in% new.key.tech) %>%
  ggplot(aes(CCI, fill = Corporation)) +
  geom_density(alpha = 0.5) +
  labs(title = "CCI Density: Time Period Analysis (2020)",
       subtitle = "Tracking Commodity Channel Index Shift for Alibaba, JD, and Tencent.",
       caption = "Time Series: 01/01/2020 to 01/01/2021.",
       y = "Density") +
  xlim(-300,300) +
  theme(axis.title.x=element_blank())
```



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.

```
df.CCI %>%
  select(time, Corporation = symbol, CCI) %>%
  filter(time >"2021-02-01") %>%
  filter(time < "2021-12-08") %>%
  filter(Corporation %in% new.key.tech) %>%
  ggplot(aes(CCI, fill = Corporation)) +
  geom_density(alpha = 0.5) +
  labs(title = "CCI Density: Time Period Analysis (2021)",
       subtitle = "Tracking Commodity Channel Index Shift for Alibaba, JD, and Tencent.",
       caption = "Time Series: 02/01/2021 to 12/08/2021.",
       y = "Density") +
  xlim(-300,300) +
  theme(axis.title.x=element_blank())
```



Thank you for taking the time to read this report.