Regression to test initial hypothesis (more links better sustainability)

AlexSh

27 July, 2023

library(tidyverse)

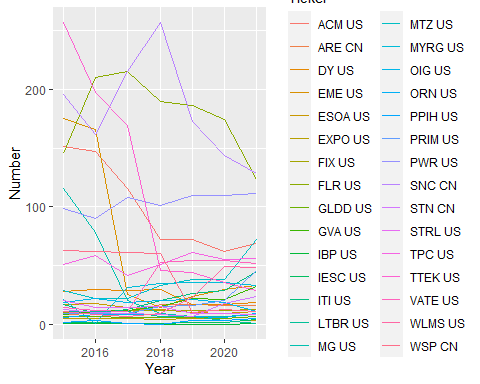
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(knitr)  
library(rmarkdown)  
library(readxl)  
library(ggplot2)  
library(dplyr)  
library(tidyr)  
library(readr)

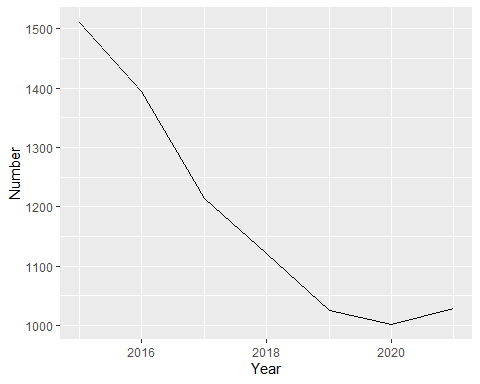
# download network data of construction companies  
splc <- read\_excel("../data/raw/SPLC\_NA.xlsx", sheet=2)  
# exclude NAs, convert column to numeric  
splc <- splc %>%   
 filter(Ticker != "BDGI CN" & Ticker != 'J US') %>%   
 mutate(Number = as.numeric(Number))

# aggregated connections of each ticker  
splc %>%   
 group\_by(Year,Ticker) %>%   
 summarise(Number = sum(Number)) %>%  
 ggplot(aes(x = Year, y = Number, color = Ticker)) +  
 geom\_line()

## `summarise()` has grouped output by 'Year'. You can override using the  
## `.groups` argument.

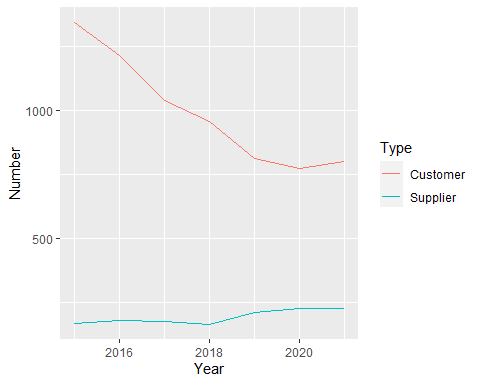


# aggregated connections  
splc %>%   
 group\_by(Year) %>%   
 summarize(Number = sum(Number)) %>%   
 ggplot(aes(x = Year, y = Number)) +  
 geom\_line()



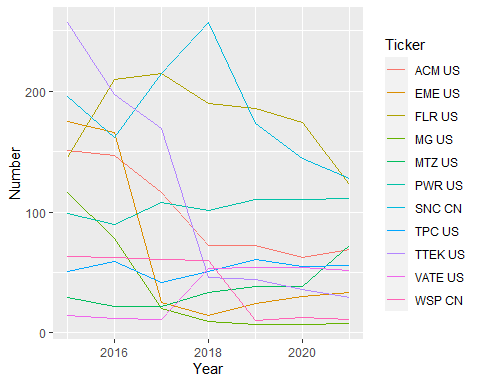
# connections with customer&supplier  
splc %>%   
 group\_by(Year,Type) %>%   
 summarize(Number = sum(Number)) %>%   
 ggplot(aes(x = Year, y = Number, color = Type)) +   
 geom\_line()

## `summarise()` has grouped output by 'Year'. You can override using the  
## `.groups` argument.



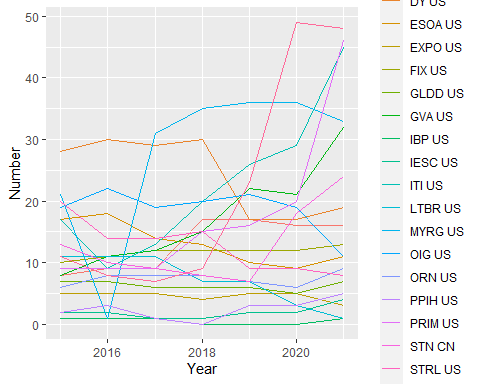
# tickers with high supply chain  
splc %>%   
 group\_by(Ticker) %>%   
 filter(mean(Number) >= 17) %>%  
 group\_by(Year,Ticker) %>%   
 summarise(Number = sum(Number)) %>%  
 ggplot(aes(x = Year, y = Number, color = Ticker)) +  
 geom\_line()

## `summarise()` has grouped output by 'Year'. You can override using the  
## `.groups` argument.



# tickers with low supply chain  
splc %>%   
 group\_by(Ticker) %>%   
 filter(mean(Number) < 17) %>%  
 group\_by(Year,Ticker) %>%   
 summarise(Number = sum(Number)) %>%  
 ggplot(aes(x = Year, y = Number, color = Ticker)) +  
 geom\_line()

## `summarise()` has grouped output by 'Year'. You can override using the  
## `.groups` argument.



# download sustainability data  
sus <- read\_excel("../data/raw/C&S.xlsx") %>%   
 filter(Country %in% c('CANADA','UNITED STATES')) %>%   
 mutate\_at(5:ncol(.),as.numeric) %>%  
# exclude Tickets without any sustainability data  
 filter(!rowSums(is.na(.[,5:ncol(.)])) == ncol(.)-4) %>%   
# exclude all ESG Disclosure Score data  
 select(-c(RX317\_FY2013,RX317\_FY2014,  
 RX317\_FY2015,RX317\_FY2016,  
 RX317\_FY2016,RX317\_FY2017,  
 RX317\_FY2018,RX317\_FY2019,  
 RX317\_FY2020,RX317\_FY2021))

## Warning: There were 114 warnings in `mutate()`.  
## The first warning was:  
## ℹ In argument: `RX317\_FY2013 = .Primitive("as.double")(RX317\_FY2013)`.  
## Caused by warning:  
## ! NAs introduced by coercion  
## ℹ Run `dplyr::last\_dplyr\_warnings()` to see the 113 remaining warnings.

# remove Tickers with all NAs  
sus <- sus %>%   
 filter(!rowSums(is.na(.[,5:ncol(.)])) == ncol(.)-4)  
# keep companies without NAs  
sus <- sus %>%   
 filter(!Ticker %in% c("FTDR US", "DY US"))

# format to long  
sus\_long <- sus %>%  
 pivot\_longer(cols = starts\_with("SR"),   
 names\_to = "variable",   
 values\_to = "value") %>%  
 separate(variable,   
 c("SR\_variable", "Year"),   
 sep = "\_FY") %>%  
 mutate(Year = as.numeric(Year)) %>%   
 pivot\_wider(names\_from = SR\_variable,   
 values\_from = value)  
  
# add number of suppliers to number of customers  
splc\_aggr <- splc %>%   
 group\_by(Ticker,Year) %>%   
 summarize(Number = sum(Number))

## `summarise()` has grouped output by 'Ticker'. You can override using the  
## `.groups` argument.

# merge with network data  
merged\_df <- left\_join(sus\_long, splc\_aggr, by = c('Year' = 'Year', 'Ticker' = 'Ticker'))  
remove(sus,sus\_long,splc,splc\_aggr)  
  
# exclude Tickers with NAs in Number  
merged\_df <- merged\_df %>%   
 filter(!Ticker %in% c('J US', 'BLD US'))

# correlation analysis  
sr\_cols <- grep('^SR', names(merged\_df), value = T)  
correlations <- sapply(sr\_cols, function(x)   
 cor(merged\_df[[x]],merged\_df$Number))  
correlations <- sapply(sr\_cols, function(x) cor(merged\_df[[x]], merged\_df$Number))  
  
cor\_df <- data.frame(SR\_variable = sr\_cols, Correlation = correlations)  
  
# display in table format  
kable(cor\_df)

|  | SR\_variable | Correlation |
| --- | --- | --- |
| SR001 | SR001 | 0.0950299 |
| SR002 | SR002 | 0.0364326 |
| SR235 | SR235 | -0.0945836 |
| SR200 | SR200 | 0.1900227 |
| SR238 | SR238 | -0.1361049 |
| SR231 | SR231 | 0.3052561 |
| SR003 | SR003 | 0.1244264 |
| SR727 | SR727 | 0.0777263 |
| SR711 | SR711 | 0.3508718 |
| SR731 | SR731 | -0.0689360 |
| SR004 | SR004 | -0.0553369 |
| SR005 | SR005 | 0.0191490 |
| SR035 | SR035 | -0.2134413 |
| SR064 | SR064 | 0.0424045 |
| SR084 | SR084 | -0.1428598 |