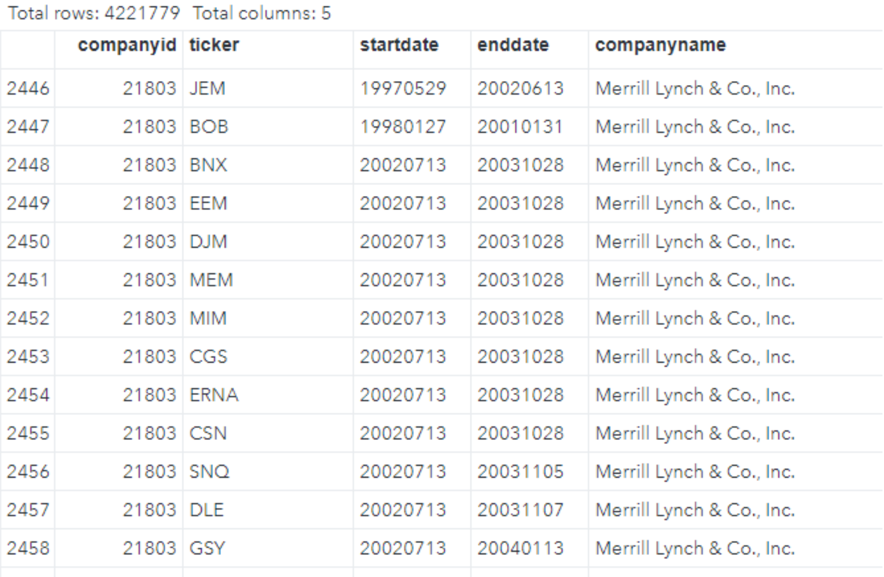
Gvkey Matching Update 1

<u>Check if Hassan is better than Compustat:</u>

- From Sixun's documentation: "In addition, we have Hassan's Firm-Level Political Risk Dataset, which also provides us with GVKEY and firm names. We think that this data set contains almost all the company's names that we have, since we use the same data source. In addition, in contrast to the Compustat dataset, Hassan's dataset has English spelling for all company's names, which helps us, since Conference Calls also has English spelling names."
- I'm not sure about the English spelling part, since I've looked at about 500 firm names on Compustat and they're all in English too.

<u>Check firm ticker – what it means to keep the first / last entry.</u>

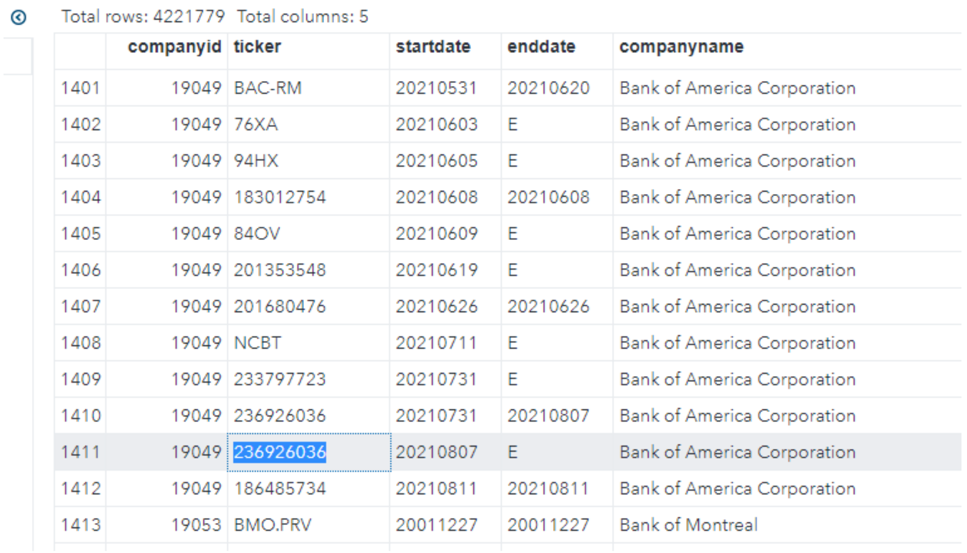
- Generally, yes, the tickers for a firm are sorted by date, so the last entry is the 'most recent'.
- But to be exact, tickers are first sorted by the startdate, then by the enddate.



Total rows: 4221779 Total columns: 5

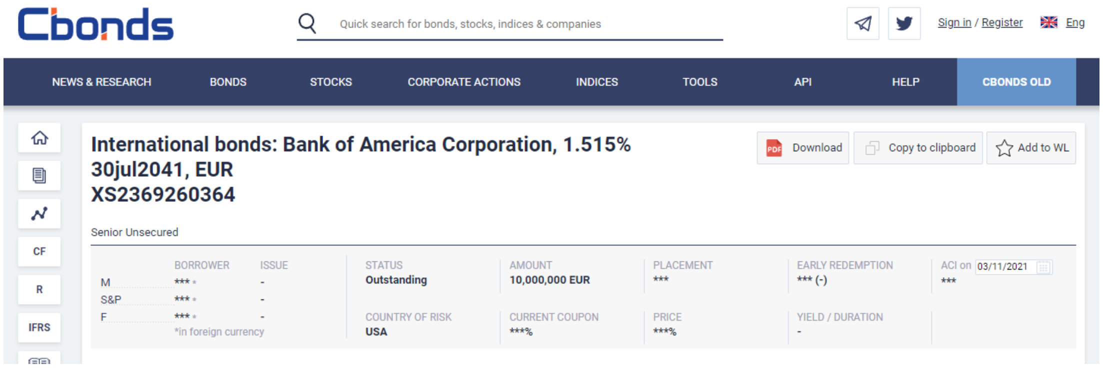
	companyid	ticker	startdate	enddate	companyname
2446	21803	JEM	19970529	20020613	Merrill Lynch & Co., Inc.
2447	21803	BOB	19980127	20010131	Merrill Lynch & Co., Inc.
2448	21803	BNX	20020713	20031028	Merrill Lynch & Co., Inc.
2449	21803	EEM	20020713	20031028	Merrill Lynch & Co., Inc.
2450	21803	DJM	20020713	20031028	Merrill Lynch & Co., Inc.
2451	21803	MEM	20020713	20031028	Merrill Lynch & Co., Inc.
2452	21803	MIM	20020713	20031028	Merrill Lynch & Co., Inc.
2453	21803	CGS	20020713	20031028	Merrill Lynch & Co., Inc.
2454	21803	ERNA	20020713	20031028	Merrill Lynch & Co., Inc.
2455	21803	CSN	20020713	20031028	Merrill Lynch & Co., Inc.
2456	21803	SNQ	20020713	20031105	Merrill Lynch & Co., Inc.
2457	21803	DLE	20020713	20031107	Merrill Lynch & Co., Inc.
2458	21803	GSY	20020713	20040113	Merrill Lynch & Co., Inc.

- Also, both stock and bond tickers are included, so in general, the last entry may not be the most recent *stock* ticker for the firm.



Total rows: 4221779 Total columns: 5

	companyid	ticker	startdate	enddate	companyname
1401	19049	BAC-RM	20210531	20210620	Bank of America Corporation
1402	19049	76XA	20210603	E	Bank of America Corporation
1403	19049	94HX	20210605	E	Bank of America Corporation
1404	19049	183012754	20210608	20210608	Bank of America Corporation
1405	19049	84OV	20210609	E	Bank of America Corporation
1406	19049	201353548	20210619	E	Bank of America Corporation
1407	19049	201680476	20210626	20210626	Bank of America Corporation
1408	19049	NCBT	20210711	E	Bank of America Corporation
1409	19049	233797723	20210731	E	Bank of America Corporation
1410	19049	236926036	20210731	20210807	Bank of America Corporation
1411	19049	236926036	20210807	E	Bank of America Corporation
1412	19049	186485734	20210811	20210811	Bank of America Corporation
1413	19053	BMO.PRV	20011227	20011227	Bank of Montreal



International bonds: Bank of America Corporation, 1.515% 30jul2041, EUR
XS2369260364

Senior Unsecured

	BORROWER	ISSUE
M	***	-
S&P	***	-
F	***	-

*in foreign currency

STATUS: Outstanding
AMOUNT: 10,000,000 EUR
PLACEMENT: ***
EARLY REDEMPTION: *** (-)
ACI on 03/11/2021: ***
COUNTRY OF RISK: USA
CURRENT COUPON: ***%
PRICE: ***%
YIELD / DURATION: -

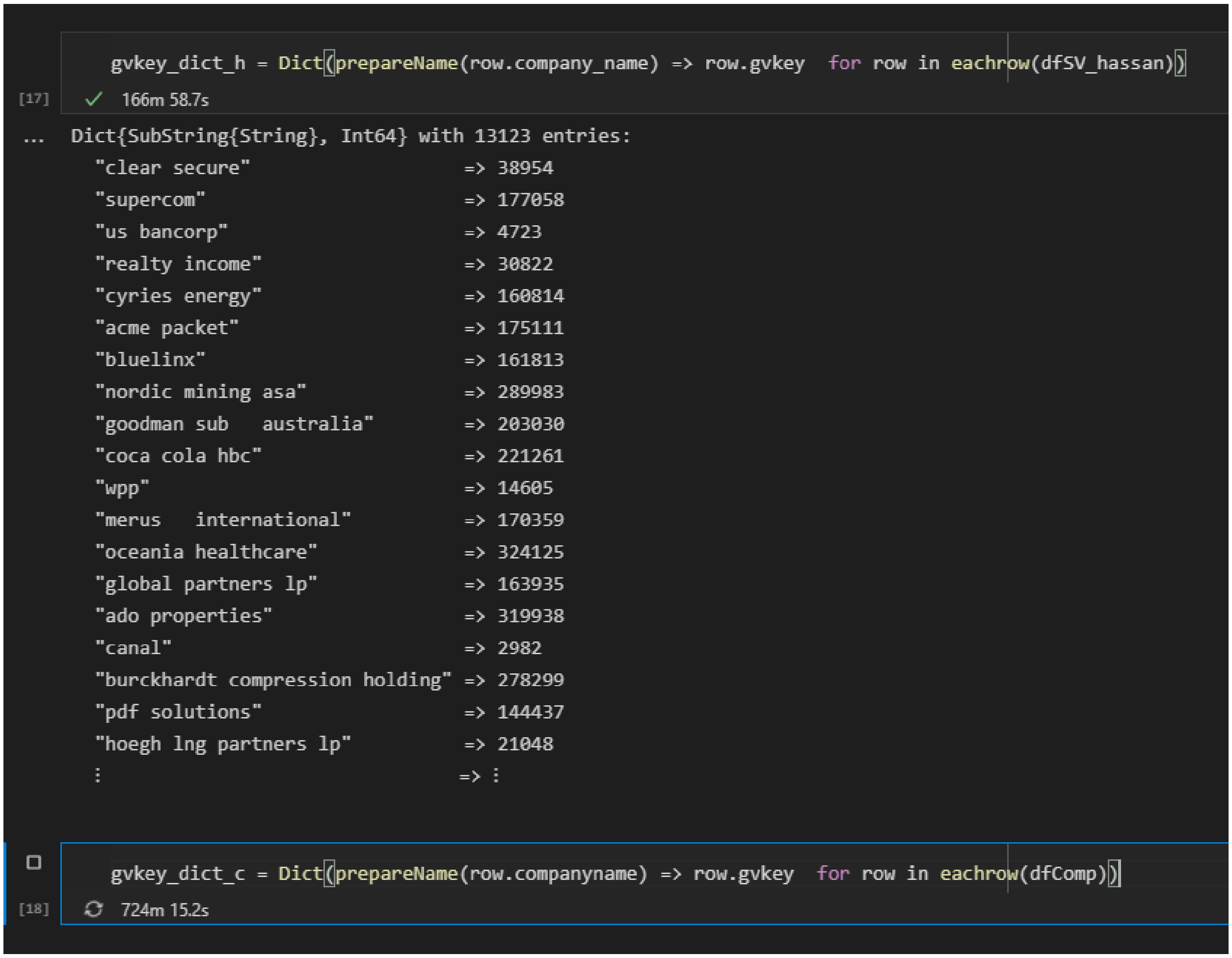
Check what the ‘country’ columns represent

- In the ciqcompany dataset, there are 2 country columns:
  - Countryid: likely the country of HQ.
  - Incorporationcountryid: likely the country in which the company is incorporated or legally registered.
  - I’m using Countryid as the country variable.
  - The manuals I found didn’t contain these variables, so I have emailed the Capital IQ Helpdesk to confirm.

In progress:

Produce the gvkeys using the original code/method

- The current bottleneck is the runtime. I’ve been running the code for 15+ hours, and the program is still executing a particular line, which is to turn a dataframe into a dictionary. The runtimes are displayed below: the first one is completed in about 3 hours, and the second is still running after 12 hours.



```
gvkey_dict_h = Dict(prepareName(row.company_name) => row.gvkey  for row in eachrow(dfSV_hassan))
```

```
Dict{SubString{String}, Int64} with 13123 entries:
  "clear secure"                   => 38954
  "supercom"                       => 177058
  "us bancorp"                     => 4723
  "realty income"                  => 30822
  "cyries energy"                  => 160814
  "acme packet"                    => 175111
  "bluelinx"                       => 161813
  "nordic mining asa"              => 289983
  "goodman sub   australia"        => 203030
  "coca cola hbc"                  => 221261
  "wpp"                            => 14605
  "merus   international"          => 170359
  "oceania healthcare"             => 324125
  "global partners lp"             => 163935
  "ado properties"                 => 319938
  "canal"                          => 2982
  "burckhardt compression holding" => 278299
  "pdf solutions"                  => 144437
  "hoegh lng partners lp"          => 21048
  ⋮                                => ⋮
```

```
gvkey_dict_c = Dict(prepareName(row.companyname) => row.gvkey  for row in eachrow(dfComp))
```

<ins>Produce some statistics after getting the gvkeys</ins>

- Number of perfect matches
- Number of imperfect matches > 80%
- Number of imperfect matches >90%
- Whether matched by fuzzy or by identifying the name perfectly.

<ins>Check for duplicates</ins>

- Some firms have multiple gvkeys. We want to avoid the situation where the same firm is matched to different gvkeys and thus looking different in our dataset.
- Stata duplicates report to check if the firm name appears twice