1.2 Data Collection - Collect company data from compustat

First, we setup our connection to the wrds database.

In [1]:

```
cd ..
```

```
/home/julian/PycharmProjects/corporate_disruptions
```

In [2]:

```
import parameters
import wrds_tools
```

In [3]:

```
wrds = wrds_tools.WrdsConnection(wrds_username=parameters.wrds_username)
```

```
Loading library list...
Done
```

Setup observation period and grab the basic info we need.

In [4]:

```
from datetime import date

wrds.set_selection_period(start_date=date(year=1998, month=1, day=1),
                          end_date=date(year=2017, month=12, day=31))
wrds.build_sp500()
```

Add names and industry classification system GICS

In [5]:

```
wrds.add_names()
wrds.dataset['name'] = wrds.dataset['name'].str.title()

wrds.add_industry_classifiers(get_gics=True)

wrds.head(3)
```

Out[5]:

	gvkey	name	SIC	NAICS	GICS_group	GICS_industry	GICS_sector	GICS_subindustry
0	001013	Adc Telecommunications Inc	3661	334210	4520	452010	45	45201020
1	001045	American Airlines Group Inc	4512	481111	2030	203020	20	20302010
2	001075	Pinnacle West Capital Corp	4911	2211	5510	551010	55	55101010

1.2.1 Filter by industry

We use the GICS industry classification system to filter. For an overview, see [https://en.wikipedia.org/wiki/Global_Industry_Classification_Standard (https://en.wikipedia.org/wiki/Global_Industry_Classification_Standard)](https://en.wikipedia.org/wiki/Global_Industry_Classification_Standard) . Industry Group 2550 is Retailing.

In [6]:

```
wrds.filter_by_industry(industry_code='2550', classification_system='GICS_grou
```

In [7]:

```
print('Number of observations: ', len(wrds.dataset),
      '\n\n----------------\n')
print(wrds.dataset['name'])
wrds.dataset.head()
```

```
Number of observations:  51

----------------

0                 Best Buy Co Inc
1                  Officemax Inc
2             Charming Shoppes Inc
3           Circuit City Stores Inc
4                    Target Corp
5               Dillards Inc  -Cl A
6               Dollar General Corp
7             Family Dollar Stores
8                     Macy'S Inc
9                        Gap Inc
10                Genuine Parts Co
11                  Home Depot Inc
12             Sears Holdings Corp
13                  L Brands Inc
14            Lowe'S Companies Inc
15        May Department Stores Co
16        Mercantile Stores Co Inc
17                  Nordstrom Inc
18                Penney (J C) Co
19       Pep Boys-Manny Moe & Jack
20                  Autonation Inc
21                Ross Stores Inc
22              Sears Roebuck & Co
23                 Rs Legacy Corp
24                  Toys R Us Inc
25                Foot Locker Inc
26               Tjx Companies Inc
27                   Big Lots Inc
28                  Tiffany & Co
29                Office Depot Inc
30            Signet Jewelers Ltd
31                    Staples Inc
32                   Autozone Inc
33                    Kohl'S Corp
34           Bed Bath & Beyond Inc
35         O'Reilly Automotive Inc
36                  Petsmart Inc
37            Urban Outfitters Inc
38              Tractor Supply Co
39                Dollar Tree Inc
40      Abercrombie & Fitch  -Cl A
41                    Carmax Inc
42                 Amazon.Com Inc
43            Booking Holdings Inc
44              Expedia Group Inc
45                  Gamestop Corp
46          Advance Auto Parts Inc
47                    Netflix Inc
48                       Lkq Corp
49               Ulta Beauty Inc
50               Tripadvisor Inc
Name: name, dtype: object
```

Out[7]:

gvkey	name	SIC	NAICS	GICS_group	GICS_industry	GICS_sector	GICS_subindustry

1.2.2 Filter out Internet & Direct Marketing Retail

Internet & Direct Marketing Retail is the GICS Sub-Industry 25502020.

In [8]:

```python
wrds.dataset[wrds.dataset['GICS_subindustry'] == '25502020']
```

Out[8]:

	gvkey	name	SIC	NAICS	GICS_group	GICS_industry	GICS_sector	GICS_subindustry
42	064768	Amazon.Com Inc	5961	454111	2550	255020	25	25502020
43	119314	Booking Holdings Inc	7370	519130	2550	255020	25	25502020
44	126296	Expedia Group Inc	4700	561510	2550	255020	25	25502020
47	147579	Netflix Inc	7841	532230	2550	255020	25	25502020
50	199356	Tripadvisor Inc	7370	519130	2550	255020	25	25502020

In [9]:

```python
wrds.dataset = wrds.dataset[wrds.dataset['GICS_subindustry'] != '25502020']
wrds.dataset = wrds.dataset.reset_index(drop=True)
wrds.dataset.head(3)
```

Out[9]:

	gvkey	name	SIC	NAICS	GICS_group	GICS_industry	GICS_sector	GICS_subindustry
0	002184	Best Buy Co Inc	5731	443142	2550	255040	25	25504020
1	002290	Officemax Inc	5110	424120	2550	255040	25	25504040
2	002938	Charming Shoppes Inc	5621	448120	2550	255040	25	25504010

1.3 Add executives data

Set observation period.

In [10]:

```python
wrds.set_observation_period(start_date=date(year=2008, month=1, day=1),
                            end_date=date(year=2017, month=12, day=31))
```

In [11]:

```python
len(wrds.dataset)
```

Out[11]:

```
46
```

In [12]:

```
wrds.add_executives()

wrds.head()
```

Out[12]:

	gvkey	name	SIC	NAICS	GICS_group	GICS_industry	GICS_sector	GICS_subindustry	execid	yea
0	002184	Best Buy Co Inc	5731	443142	2550	255040	25	25504020	06175	200
1	002184	Best Buy Co Inc	5731	443142	2550	255040	25	25504020	06175	200
2	002184	Best Buy Co Inc	5731	443142	2550	255040	25	25504020	28397	200
3	002184	Best Buy Co Inc	5731	443142	2550	255040	25	25504020	28397	200
4	002184	Best Buy Co Inc	5731	443142	2550	255040	25	25504020	28397	201

In [13]:

```
len(wrds.dataset)
```

Out[13]:

```
2242
```

In [14]:

```
wrds.add_executive_info(add_ceo_flag=True)
```

```
Executives already added to the dataset. Executives will not be merged in aga
in.
Added the following info on executives: ['ceoann']
```

In [15]:

```
len(wrds.dataset)
```

Out[15]:

```
2242
```

1.4 Output dataframe as .feather

The .feather format allows us to further manipulate the dataframe in R without having to specify column types again.

In [16]:

```
selection = wrds.return_dataframe()
```

We transform the year column to datatype 'object' (character strings) because feather cannot handle pandas 'Int64' datatype yet.

In [17]:

```
selection['year'] = selection['year'].astype('object')
```

In [18]:

```
from datetime import date

selection.to_feather('downloads/sample_{}.feather'.format(str(date.today())))
```