Replicated Regressions

March 7, 2019

In [43]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats
from sklearn import linear_model
from tqdm import tqdm_notebook as tqdm
```

In [44]:
```
column_names = ['Stock Symbol', 'YearMonth', 'NumEmployees', 'AverageAge', 'NumPeople
month_current = pd.read_csv('fastwhitepaper_month_current.csv', sep="\t", error_bad_li
month_join = pd.read_csv('fastwhitepaper_month_join.csv', sep="\t", error_bad_lines=Fa
month_leave = pd.read_csv('fastwhitepaper_month_leave.csv', sep="\t", error_bad_lines=
```

In [45]:
```
month_current = month_current.rename(columns={'NumEmployees': 'NumEmployeesCurrent'})
month_join = month_join.rename(columns={'NumEmployees': 'NumEmployeesJoin'})
month_leave = month_leave.rename(columns={'NumEmployees': 'NumEmployeesLeave'})
```

In [46]:
```
month_current_employees = month_current.iloc[:, 0:3]
month_join_employees = month_join.iloc[:, 2:3]
month_leave_employees = month_leave.iloc[:, 2:3]
```

In [47]:
```
frames = [month_current_employees, month_join_employees, month_leave_employees]
month_combined = pd.concat(frames, axis=1, join='inner')
month_combined.head()
```

Out[47]:

	Stock Symbol	YearMonth	NumEmployeesCurrent	NumEmployeesJoin	NumEmployeesLeave
0	AGO	199001	2	0	0
1	AGO	199002	2	0	0
2	AGO	199003	2	0	0
3	AGO	199004	3	1	0
4	AGO	199005	3	0	0

```
In [48]: num_employees_current = month_combined.loc[:,'NumEmployeesCurrent'].values.astype(flo
         num_employees_join = month_combined.loc[:,'NumEmployeesJoin'].values.astype(float)
         num_employees_leave = month_combined.loc[:,'NumEmployeesLeave'].values.astype(float)
         join_depart_sum = np.add(num_employees_join, num_employees_leave)
         turnover = np.divide(join_depart_sum,
                              num_employees_current,
                              out=(np.zeros_like(join_depart_sum)),
                              where=(num_employees_current > 0))
         turnover_df = pd.DataFrame(turnover, columns=['Turnover Rate'])

In [49]: frames = [month_combined, turnover_df]
         month_combined_with_turnover = pd.concat(frames, axis=1, join='inner')
         month_combined_with_turnover.head()

Out[49]:   Stock Symbol  YearMonth  NumEmployeesCurrent  NumEmployeesJoin  \
         0          AGO     199001                    2                 0
         1          AGO     199002                    2                 0
         2          AGO     199003                    2                 0
         3          AGO     199004                    3                 1
         4          AGO     199005                    3                 0

            NumEmployeesLeave  Turnover Rate
         0                  0       0.000000
         1                  0       0.000000
         2                  0       0.000000
         3                  0       0.333333
         4                  0       0.000000

In [50]: minDate = min(month_combined_with_turnover.loc[:, 'YearMonth'])
         maxDate = max(month_combined_with_turnover.loc[:, 'YearMonth'])
         maxDate

Out[50]: 201707

In [51]: winsorized_turnover = scipy.stats.mstats.winsorize(month_combined_with_turnover["Turn
         winsorized_turnover

Out[51]: masked_array(data=[0., 0., 0., ..., 0., 0., 0.],
                      mask=False,
                fill_value=1e+20)

In [52]: winsorized_turnover_df = pd.DataFrame(winsorized_turnover, columns=['Winsorized Turno

In [53]: frames = [month_combined, winsorized_turnover_df]
         month_combined_with_winsorized_turnover = pd.concat(frames, axis=1, join='inner')
         month_combined_with_winsorized_turnover.head()

Out[53]:   Stock Symbol  YearMonth  NumEmployeesCurrent  NumEmployeesJoin  \
         0          AGO     199001                    2                 0
```

```
1          AGO     199002                    2                  0
2          AGO     199003                    2                  0
3          AGO     199004                    3                  1
4          AGO     199005                    3                  0

   NumEmployeesLeave  Winsorized Turnover Rate
0                  0                      0.00
1                  0                      0.00
2                  0                      0.00
3                  0                      0.25
4                  0                      0.00
```

In [54]:
```
column_names = ['gvkey','datadate','fyearq','fqtr','indfmt','consol','popsrc','datafmt
compustat_data = pd.read_csv('Compustat_2000_2016.csv', sep="\t", names = column_names
compustat_data.head()
```

Out[54]:
```
   gvkey  datadate  fyearq  fqtr indfmt consol popsrc datafmt  tic      cusip  \
0   1004  20000229    1999     3   INDL      C      D     STD  AIR  000361105
1   1004  20000531    1999     4   INDL      C      D     STD  AIR  000361105
2   1004  20000831    2000     1   INDL      C      D     STD  AIR  000361105
3   1004  20001130    2000     2   INDL      C      D     STD  AIR  000361105
4   1004  20010228    2000     3   INDL      C      D     STD  AIR  000361105

     ...    datafqtr         rdq     ceqq   cshoq  epsf12  epsfxq  xrdq  \
0    ...      1999Q3  20000315.0  342.482  26.963    1.61    0.40   NaN
1    ...      1999Q4  20000628.0  339.515  26.865    1.28    0.09   NaN
2    ...      2000Q1  20000920.0  339.253  26.857    1.01    0.12   NaN
3    ...      2000Q2  20001220.0  341.264  26.932    0.77    0.16   NaN
4    ...      2000Q3  20010320.0  344.865  26.945    0.57    0.20   NaN

   costat   prccq     naics
0       A  23.750  423860.0
1       A  13.875  423860.0
2       A  11.250  423860.0
3       A  10.375  423860.0
4       A  13.600  423860.0

[5 rows x 22 columns]
```

In [55]:
```
report_date_df = pd.DataFrame(compustat_data['datadate'].values, columns=["Report Date
report_date_df.head()
```

Out[55]:
```
   Report Date
0     20000229
1     20000531
2     20000831
3     20001130
4     20010228
```

```
In [56]: ticker = compustat_data['tic'].values
         ticker_df = pd.DataFrame(ticker, columns=["Stock Symbol"])
         frames = [report_date_df, ticker_df]
         date_ticker_df = pd.concat(frames, axis=1, join='inner')
         date_ticker_df.head()
```

Out[56]:

	Report Date	Stock Symbol
0	20000229	AIR
1	20000531	AIR
2	20000831	AIR
3	20001130	AIR
4	20010228	AIR

```
In [57]: shares_outstanding = compustat_data['cshoq'].values
         price_per_share = compustat_data['prccq'].values
         market_capitalization = np.multiply(shares_outstanding, price_per_share)
         size = np.log(market_capitalization)
         size_df = pd.DataFrame(size, columns=["Size"])
         frames = [report_date_df, size_df]
         date_size_df = pd.concat(frames, axis=1, join='inner')
         date_size_df.head()
```

```
/Users/timothyhuang/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:4: RuntimeWarni
  after removing the cwd from sys.path.
```

Out[57]:

	Report Date	Size
0	20000229	6.462048
1	20000531	5.920913
2	20000831	5.710895
3	20001130	5.632714
4	20010228	5.903868

```
In [58]: book_value = compustat_data['ceqq'].values
         book_to_market_ratio = np.divide(book_value, market_capitalization, out=np.zeros_like
         book_to_market_ratio_df = pd.DataFrame(book_to_market_ratio, columns=["Book to Market
         frames = [report_date_df, book_to_market_ratio_df]
         date_book_to_market_ratio_df = pd.concat(frames, axis=1, join='inner')
         date_book_to_market_ratio_df.head()
```

Out[58]:

	Report Date	Book to Market Ratio
0	20000229	0.534818
1	20000531	0.910834
2	20000831	1.122829
3	20001130	1.221332
4	20010228	0.941092

```
In [59]: def slicer_vectorized(a,start,end):
             b = a.view((str,1)).reshape(len(a),-1)[:,start:end]
```

```
        return np.fromstring(b.tostring(),dtype=(str,end-start))

    naics_industry = compustat_data['naics'].values
    naics_industry = slicer_vectorized(naics_industry.astype(str),0,2) #slicing naics by

    naics_industry_df = pd.DataFrame(naics_industry, columns=["NAICS Industry Classificat
    frames = [report_date_df, naics_industry_df]
    date_naics_industry_df = pd.concat(frames, axis=1, join='inner')
    date_naics_industry_df.head()
```

```
/Users/timothyhuang/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:3: DeprecationW
  This is separate from the ipykernel package so we can avoid doing imports until
```

Out[59]:

	Report Date	NAICS Industry Classification
0	20000229	42
1	20000531	42
2	20000831	42
3	20001130	42
4	20010228	42

In [60]:
```
# combined controls data
frames = [report_date_df, ticker_df, size_df, book_to_market_ratio_df, naics_industry
combined_controls = pd.concat(frames, axis=1, join='inner')
combined_controls.head()
```

Out[60]:

	Report Date	Stock Symbol	Size	Book to Market Ratio	\
0	20000229	AIR	6.462048	0.534818	
1	20000531	AIR	5.920913	0.910834	
2	20000831	AIR	5.710895	1.122829	
3	20001130	AIR	5.632714	1.221332	
4	20010228	AIR	5.903868	0.941092	

	NAICS Industry Classification
0	42
1	42
2	42
3	42
4	42

In [62]:
```
# df1 = month_combined_with_winsorized_turnover
# row = df1.loc[(df1["Stock Symbol"] == 'PM') & (df1["YearMonth"] == 201405)]
# combined_controls = combined_controls.loc[(combined_controls['Stock Symbol'] == row
# nearest_index = combined_controls['Report Date'].idxmax()
# nearest_row = combined_controls.loc[nearest_index]
# nearest_row
```

In [68]:
```
# Function to match turnover data with most recent controls data
# Winsorized turnover dataframe has columns (Stock Symbol, YearMonth, NumEmployeesCur
```

```
        # NumEmployeesJoin, NumEmployeesLeave, Winsorized Turnover Rate)
        # Controls dataframe has columns (Report date, Stock Symbol, Size, Book to Market Rat
        # NAICS Industry Classification)

        def match_func(wins_turnover_df, controls_df):
            matched_dict = dict()
            wins_turnover_df = wins_turnover_df.loc[wins_turnover_df['YearMonth'] >= 200000]
            for index, row in tqdm(wins_turnover_df.iterrows()):
                nearest_row = None
                pruned_controls = controls_df.loc[(controls_df['Stock Symbol'] == row['Stock
                if (len(pruned_controls) > 0):
                    nearest_index = pruned_controls['Report Date'].idxmax()
                    nearest_row = pruned_controls.loc[nearest_index]
                matched_dict[(row['Stock Symbol'], row['YearMonth'], row['Winsorized Turnover
            return matched_dict
```

In [ ]:
```
matched_dict = match_func(month_combined_with_winsorized_turnover, combined_controls)
matched_dict
```

```
HBox(children=(IntProgress(value=1, bar_style='info', max=1), HTML(value='')))
```

In [ ]: