In [4]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats
from sklearn import linear_model
```

In [5]:
```
column_names = ['Stock Symbol', 'YearMonth', 'NumEmployees', 'AverageAge', 'NumPeopleWithKnownAge', 'NumFemale', 'NumMal
e', 'NumNoSkills', 'Average Tenure', 'SkillsFreq']
month_current = pd.read_csv('fastwhitepaper_month_current.csv', sep="\t", error_bad_lines=False, names = column_names)
month_join = pd.read_csv('fastwhitepaper_month_join.csv', sep="\t", error_bad_lines=False, names = column_names)
month_leave = pd.read_csv('fastwhitepaper_month_leave.csv', sep="\t", error_bad_lines=False, names = column_names)
```

In [6]:
```
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 [7]:
```
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 [8]:
```
frames = [month_current_employees, month_join_employees, month_leave_employees]
month_combined = pd.concat(frames, axis=1, join='inner')
month_combined.head()
```

Out[8]:

	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 [9]:
```
num_employees_current = month_combined.loc[:,'NumEmployeesCurrent'].values.astype(float)
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 [10]:
```
frames = [month_combined, turnover_df]
month_combined_with_turnover = pd.concat(frames, axis=1, join='inner')
month_combined_with_turnover.head()
```

Out[10]:

	Stock Symbol	YearMonth	NumEmployeesCurrent	NumEmployeesJoin	NumEmployeesLeave	Turnover Rate
0	AGO	199001	2	0	0	0.000000
1	AGO	199002	2	0	0	0.000000
2	AGO	199003	2	0	0	0.000000
3	AGO	199004	3	1	0	0.333333
4	AGO	199005	3	0	0	0.000000

In [11]:
```
minDate = min(month_combined_with_turnover.loc[:, 'YearMonth'])
maxDate = max(month_combined_with_turnover.loc[:, 'YearMonth'])
maxDate
```

Out[11]:
```
201707
```

In [12]:
```
winsorized_turnover = scipy.stats.mstats.winsorize(month_combined_with_turnover["Turnover Rate"].values, limits=[0.01,0.
01])
winsorized_turnover
```

Out[12]:
```
masked_array(data=[0., 0., 0., ..., 0., 0., 0.],
             mask=False,
       fill_value=1e+20)
```

In [13]:
```
winsorized_turnover_df = pd.DataFrame(winsorized_turnover, columns=['Winsorized Turnover Rate'])
```

In [14]:

```
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[14]:

	Stock Symbol	YearMonth	NumEmployeesCurrent	NumEmployeesJoin	NumEmployeesLeave	Winsorized Turnover Rate
0	AGO	199001	2	0	0	0.00
1	AGO	199002	2	0	0	0.00
2	AGO	199003	2	0	0	0.00
3	AGO	199004	3	1	0	0.25
4	AGO	199005	3	0	0	0.00

In [15]:

```
column_names = ['gvkey','datadate','fyearq','fqtr','indfmt','consol','popsrc','datafmt','tic','cusip','curcdq','datacqt
r','datafqtr','rdq','ceqq','cshoq','epsf12','epsfxq','xrdq','costat','prccq','naics']
compustat_data = pd.read_csv('Compustat_2000_2016.csv', sep="\t", names = column_names)
compustat_data.head()
```

Out[15]:

	gvkey	datadate	fyearq	fqtr	indfmt	consol	popsrc	datafmt	tic	cusip	...	datafqtr	rdq	ceqq	cshoq	epsf12	epsfxq	xrdq	costat	prc
0	1004	20000229	1999	3	INDL	C	D	STD	AIR	000361105	...	1999Q3	20000315.0	342.482	26.963	1.61	0.40	NaN	A	23.7
1	1004	20000531	1999	4	INDL	C	D	STD	AIR	000361105	...	1999Q4	20000628.0	339.515	26.865	1.28	0.09	NaN	A	13.8
2	1004	20000831	2000	1	INDL	C	D	STD	AIR	000361105	...	2000Q1	20000920.0	339.253	26.857	1.01	0.12	NaN	A	11.2
3	1004	20001130	2000	2	INDL	C	D	STD	AIR	000361105	...	2000Q2	20001220.0	341.264	26.932	0.77	0.16	NaN	A	10.3
4	1004	20010228	2000	3	INDL	C	D	STD	AIR	000361105	...	2000Q3	20010320.0	344.865	26.945	0.57	0.20	NaN	A	13.6

5 rows × 22 columns

In [16]:

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

Out[16]:

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

In [21]:

```
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[21]:

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

In [23]:

```
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: RuntimeWarning: divide by zero encou
ntered in log
  after removing the cwd from sys.path.
```

Out[23]:

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

In [25]:

```
book_value = compustat_data['ceqq'].values
book_to_market_ratio = np.divide(book_value, market_capitalization, out=np.zeros_like(book_value), where=market_capitali
zation!=0)
book_to_market_ratio_df = pd.DataFrame(book_to_market_ratio, columns=["Book to Market Ratio"])
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[25]:

	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 [98]:

```
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 first 2 digits and storing as str

naics_industry_df = pd.DataFrame(naics_industry, columns=["NAICS Industry Classification"])
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: DeprecationWarning: The binary mode
of fromstring is deprecated, as it behaves surprisingly on unicode inputs. Use frombuffer instead
  This is separate from the ipykernel package so we can avoid doing imports until
```

Out[98]:

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

In [99]:

```
frames = [report_date_df, ticker_df, size_df, book_to_market_ratio_df, naics_industry_df]
combined_controls = pd.concat(frames, axis=1, join='inner')
combined_controls.head()
```

Out[99]:

	Report Date	Stock Symbol	Size	Book to Market Ratio	NAICS Industry Classification
0	20000229	AIR	6.462048	0.534818	42
1	20000531	AIR	5.920913	0.910834	42
2	20000831	AIR	5.710895	1.122829	42
3	20001130	AIR	5.632714	1.221332	42
4	20010228	AIR	5.903868	0.941092	42

In [100]:

```
# combined_controls[combined_controls['Report Date'] == 20000229]
```

In [ ]:

```
def combined_func()
```

In [ ]:

```
```