Project - SCOTT MCCOY.ipynb

Using Industry Indicators to Predict Stock Returns

For this project, I will compare different industry classification systems and how well each can be used to explain stock returns for early and late 2020.

The three systems used are:

- Global Industry Classification System (GICS)
- North American Industry Classification System (NAICS)
- Standard Industrial Classification (SIC).

Each of these systems has multiple levels of specificity. For example the GICS system ranges from the most general (Sector - 11 classifications) to the most specific (Sub-Industry - 158 classifications). For each of these systems, I will analyze four levels of specificity to see what is the ideal level and system for analyzing, explaining, and predicting 2020 stock returns.

The dataset consists of stocks in the Russell 3000 index, for which there are about 2700 companies with complete data on industry and returns.

The analysis takes on two levels:

- Predicting stock returns during the initial pandemic recession (Jan - Mar)
- Predicting stock returns during the later pandemic recovery (Apr - Dec)

Classification Systems

> GICS codes:

GICS website - [https://www.msci.com/gics (https://www.msci.com/gics)](https://www.msci.com/gics)

NAICS Codes:

https://www.census.gov/eos/www/naics/faqs/faqs.html (https://www.census.gov/eos/www/naics/faqs/faqs.html) - NAICS code info

https://www.census.gov/eos/www/naics/downloadables/downloadables.html (https://www.census.gov/eos/www/naics/downloadables/downloadables.html) - lookup table

"The first two digits designate the economic sector, the third digit designates the subsector, the fourth digit designates the industry group, the fifth digit designates the NAICS industry, and the sixth digit designates the national industry"

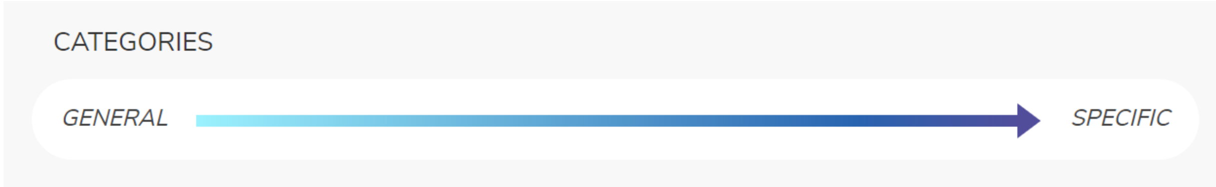
NAICS CODE

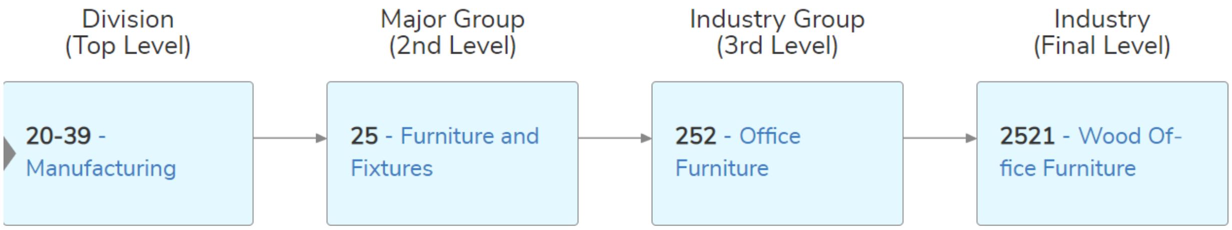
Main Sector	51	Information
Subsector	511	Publishing industries
Industry Group	5111	Newspaper, book and directory publishers
Industry	51119	Other publishers
U.S. Industry	511191	Greeting card publishers

SIC Codes:

https://siccode.com/page/what-is-a-sic-code (https://siccode.com/page/what-is-a-sic-code) - SIC code info

"The first two digits of the code identify the major industry group, the third digit identifies the industry group and the fourth digit identifies the industry."





Loading and Pre-Processing

In [ ]:
```
!pip install pandasql
```

In [2]:
```
#imports
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from scipy.stats import mstats
import pandasql as ps
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: Future
Warning: pandas.util.testing is deprecated. Use the functions in the public API
at pandas.testing instead.
  import pandas.util.testing as tm
```

Loading data:

In [3]:
```
# Loading csvs
df1 = pd.read_csv('temp1.csv') # tickers and returns (early & late 2020)
df2 = pd.read_csv('industry_codes.csv') # wrds industry codes -
```

Adding naics and sic subset codes:

In [4]:
```
# naics
df2['nsector'] = df2.naics.astype('str').str[:2]   # first two digits of naics co
df2['nsubsector'] = df2.naics.astype('str').str[:3]
df2['nindgroup'] = df2.naics.astype('str').str[:4]
df2['nindustry'] = df2.naics.astype('str').str[:5]
#sic
df2['sic'] = df2.sic.astype('str').str.zfill(4) # left padding sic codes to 4 dig
df2['sdivision'] = df2.sic.astype('str').str[:2] # first two digits of sic code r
df2['smig'] = df2.sic.astype('str').str[:2]
df2['sig'] = df2.sic.astype('str').str[:3]
df2['sindustry'] = df2.sic.astype('str').str[:4]

#merging outcomes and industry codes/names
df = df1.merge(df2, how = 'left', left_on = 'TICKER', right_on = 'tic')
df = df[(df['RetEarly2020'] != "0") & (df['RetLate2020'] != "0") & (df['RetEarly2
```

Adding lookup tables to match codes to industry names:

In [5]:
```
# GICS
gsector = pd.read_excel('Industry_Code_Lookups.xlsx', sheet_name = 'gsector')
ggroup = pd.read_excel('Industry_Code_Lookups.xlsx', sheet_name = 'ggroup')
gind = pd.read_excel('Industry_Code_Lookups.xlsx', sheet_name = 'gind')
gsubind = pd.read_excel('Industry_Code_Lookups.xlsx', sheet_name = 'gsubind')

# NAICS
nsector = pd.read_excel('Industry_Code_Lookups.xlsx', sheet_name = 'nsector')
nsector['Code'] = nsector['Code'].astype('str')
nsubsector = pd.read_excel('Industry_Code_Lookups.xlsx', sheet_name = 'nsubsector
nsubsector['Code'] = nsubsector['Code'].astype('str')
nindgroup = pd.read_excel('Industry_Code_Lookups.xlsx', sheet_name = 'nindgroup')
nindgroup['Code'] = nindgroup['Code'].astype('str')
nindustry = pd.read_excel('Industry_Code_Lookups.xlsx', sheet_name = 'nindustry')
nindustry['Code'] = nindustry['Code'].astype('str')

# SIC
sdivision = pd.read_excel('Industry_Code_Lookups.xlsx', sheet_name = 'sdivision')
sdivision['Code'] = sdivision['Code'].astype('str').str.zfill(2)
smig = pd.read_excel('Industry_Code_Lookups.xlsx', sheet_name = 'smig')
smig['Code'] = smig['Code'].astype('str').str.zfill(2)
sig = pd.read_excel('Industry_Code_Lookups.xlsx', sheet_name = 'sig')
sig['Code'] = sig['Code'].astype('str').str.zfill(3)
sindustry = pd.read_excel('Industry_Code_Lookups.xlsx', sheet_name = 'sindustry')
sindustry['Code'] = sindustry['Code'].astype('str').str.zfill(4)
```

Joining industry classification codes with industry names:

In [ ]:
```
# creating dataframes for each subset of industry classification
def ind_codes(vari):
  query = f"""SELECT a.tic, a.conm,a.RetEarly2020, a.RetLate2020 ,b.Code AS icode
  LEFT JOIN {vari} AS b ON a.{vari} = b.Code"""
  new_df = ps.sqldf(query)
  new_df = new_df[new_df['iname'].str.len() > 0]

  return new_df

dfg1 = ind_codes('gsector') # GICS
dfg2 = ind_codes('ggroup')
dfg3 = ind_codes('gind')
dfg4 = ind_codes('gsubind')
dfn1 = ind_codes('nsector') # NAICS
dfn2 = ind_codes('nsubsector')
dfn3 = ind_codes('nindgroup')
dfn4 = ind_codes('nindustry')
dfs1 = ind_codes('sdivision') # SIC
dfs2 = ind_codes('smig')
dfs3 = ind_codes('sig')
dfs4 = ind_codes('sindustry')
```

In [7]:
```
print('Unique GICS in dataset: ', len(dfg1.iname.unique()), len(dfg2.iname.unique
print('Unique NAICS in dataset: ', len(dfn1.iname.unique()), len(dfn2.iname.uniqu
print('Unique SIC in dataset: ', len(dfs1.iname.unique()), len(dfs2.iname.unique(
```

```
Unique GICS in dataset:  11 24 66 146
Unique NAICS in dataset:  18 80 230 386
Unique SIC in dataset:  10 67 198 215
```

In [8]:
```
print(dfg1.shape, dfg2.shape, dfg3.shape, dfg4.shape)
print(dfn1.shape, dfn2.shape, dfn3.shape, dfn4.shape)
print(dfs1.shape, dfs2.shape, dfs3.shape, dfs4.shape)
```

```
(2668, 6) (2668, 6) (2594, 6) (2582, 6)
(2666, 6) (2666, 6) (2666, 6) (2666, 6)
(2668, 6) (2668, 6) (2526, 6) (1771, 6)
```

Regressions

GICS

GICS-1

In [9]:
```
gresults = pd.DataFrame()
```

In [10]:
```
y1 = dfg1['RetEarly2020'].astype('float')
y2 = dfg1['RetLate2020'].astype('float')
x = dfg1['iname']
x = pd.get_dummies(x, columns=['iname'])

mod1 = sm.OLS(y1, x)
res1 = mod1.fit()
print(res1.summary())

mod2 = sm.OLS(y2, x)
res2 = mod2.fit()
print(res2.summary())

gresults = gresults.append({'Adj_rsqrd_Early': res1.rsquared_adj, 'Adj_rsqrd_Late
                            'Sig_Coef_Early': (res1.pvalues < .05).sum(), 'Sig_Coe
```

```
                            OLS Regression Results
============================================================================
==
Dep. Variable:           RetEarly2020   R-squared:                       0.1
19
Model:                            OLS   Adj. R-squared:                  0.1
15
Method:                 Least Squares   F-statistic:                     35.
81
Date:                Wed, 07 Jul 2021   Prob (F-statistic):           2.98e-
66
Time:                        21:13:21   Log-Likelihood:                -526.
72
No. Observations:                2668   AIC:                             107
5.
Df Residuals:                    2657   BIC:                             114
0.
Df Model:                          10
Covariance Type:            nonrobust
============================================================================
==================
                            coef    std err          t      P>|t|
[0.025      0.975]
----------------------------------------------------------------------------
------------------
Consumer Discretionary      -0.4142      0.017    -24.287      0.000
-0.448      -0.381
Consumer Staples            -0.1958      0.029     -6.759      0.000
-0.253      -0.139
Energy                      -0.5807      0.029    -20.050      0.000
-0.638      -0.524
Financials                  -0.3472      0.013    -26.415      0.000
-0.373      -0.321
Health Care                 -0.1397      0.013    -10.381      0.000
-0.166      -0.113
Industrials                 -0.3237      0.015    -21.081      0.000
-0.354      -0.294
Information Technology      -0.2168      0.016    -13.594      0.000
```

```
-0.248      -0.186
Materials                     -0.3538      0.027    -13.122      0.000
-0.407      -0.301
Real Estate                   -0.3293      0.022    -14.830      0.000
-0.373      -0.286
Telecommunication Services    -0.2914      0.030     -9.563      0.000
-0.351      -0.232
Utilities                     -0.1498      0.036     -4.211      0.000
-0.219      -0.080
============================================================================
==
Omnibus:                    4223.542   Durbin-Watson:                   1.8
64
Prob(Omnibus):                 0.000   Jarque-Bera (JB):          4599798.8
02
Skew:                          9.705   Prob(JB):                         0.
00
Kurtosis:                    205.486   Cond. No.                         2.
71
============================================================================
==

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is corre
ctly specified.
                            OLS Regression Results
============================================================================
==
Dep. Variable:           RetLate2020   R-squared:                       0.0
96
Model:                           OLS   Adj. R-squared:                  0.0
92
Method:                Least Squares   F-statistic:                     28.
14
Date:               Wed, 07 Jul 2021   Prob (F-statistic):           9.44e-
52
Time:                       21:13:21   Log-Likelihood:                 -385
0.2
No. Observations:               2668   AIC:                             772
2.
Df Residuals:                   2657   BIC:                             778
7.
Df Model:                         10
Covariance Type:           nonrobust
============================================================================
==================
                                 coef    std err          t      P>|t|
[0.025      0.975]
----------------------------------------------------------------------------
------------------
Consumer Discretionary         1.5656      0.059     26.415      0.000
1.449       1.682
Consumer Staples               0.5416      0.101      5.380      0.000
0.344       0.739
Energy                         1.0175      0.101     10.108      0.000
0.820       1.215
Financials                     0.4990      0.046     10.923      0.000
```

```
0.409       0.589
Health Care                    0.7435      0.047     15.901      0.000
0.652       0.835
Industrials                    0.8109      0.053     15.194      0.000
0.706       0.916
Information Technology         1.0034      0.055     18.102      0.000
0.895       1.112
Materials                      0.8805      0.094      9.396      0.000
0.697       1.064
Real Estate                    0.4592      0.077      5.951      0.000
0.308       0.611
Telecommunication Services     0.7862      0.106      7.425      0.000
0.579       0.994
Utilities                      0.1838      0.124      1.488      0.137
-0.058       0.426
============================================================================
==
Omnibus:                     2868.913   Durbin-Watson:                   1.9
49
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           329917.9
14
Skew:                           5.187   Prob(JB):                         0.
00
Kurtosis:                      56.480   Cond. No.                         2.
71
============================================================================
==

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is corre
ctly specified.
```

GICS-2

In [11]:
```
y1 = dfg2['RetEarly2020'].astype('float')
y2 = dfg2['RetLate2020'].astype('float')
x = dfg2['iname']
x = pd.get_dummies(x, columns=['iname'])

mod1 = sm.OLS(y1, x)
res1 = mod1.fit()
print(res1.summary())

mod2 = sm.OLS(y2, x)
res2 = mod2.fit()
print(res2.summary())

gresults = gresults.append({'Adj_rsqrd_Early': res1.rsquared_adj, 'Adj_rsqrd_Late
                           'Sig_Coef_Early': (res1.pvalues < .05).sum(), 'Sig_Coe
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:           RetEarly2020   R-squared:                       0.131
Model:                            OLS   Adj. R-squared:                  0.123
Method:                 Least Squares   F-statistic:                     17.25
Date:                Wed, 07 Jul 2021   Prob (F-statistic):           1.46e-64
Time:                        21:13:21   Log-Likelihood:                -508.83
No. Observations:                2668   AIC:                             1066.
Df Residuals:                    2644   BIC:                             1207.
Df Model:                          23
Covariance Type:            nonrobust
===============================================================================
===================================
                                                          coef    std err          t
P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
-----------------------------------
Automobiles & Components                               -0.4295      0.055     -7.863
0.000      -0.537      -0.322
Banks                                                  -0.3696      0.017    -21.617
0.000      -0.403      -0.336
Capital Goods                                          -0.3200      0.019    -16.678
0.000      -0.358      -0.282
Commercial  & Professional Services                    -0.3325      0.032    -10.299
0.000      -0.396      -0.269
Consumer Durables & Apparel                            -0.3963      0.033    -12.127
0.000      -0.460      -0.332
Consumer Services                                      -0.4539      0.031    -14.557
0.000      -0.515      -0.393
Diversified Financials                                 -0.3600      0.026    -13.682
0.000      -0.412      -0.308
Energy                                                 -0.5807      0.029    -20.135
0.000      -0.637      -0.524
Food & Staples Retailing                               -0.1759      0.063     -2.804
0.005      -0.299      -0.053
Food, Beverage & Tobacco                               -0.1887      0.039     -4.887
0.000      -0.264      -0.113
Health Care Equipment & Services                       -0.1356      0.022     -6.081
0.000      -0.179      -0.092
```

```
Household & Personal Products                     -0.2310      0.060     -3.848
0.000      -0.349      -0.113
Insurance                                         -0.2495      0.032     -7.773
0.000      -0.312      -0.187
Materials                                         -0.3538      0.027    -13.178
0.000      -0.406      -0.301
Media & Entertainment                             -0.3460      0.034    -10.119
0.000      -0.413      -0.279
Pharmaceuticals, Biotechnology & Life Sciences    -0.1420      0.017     -8.471
0.000      -0.175      -0.109
Real Estate                                       -0.3293      0.022    -14.893
0.000      -0.373      -0.286
Retailing                                         -0.3892      0.029    -13.297
0.000      -0.447      -0.332
Semiconductors & Semiconductor Equipment          -0.2265      0.037     -6.064
0.000      -0.300      -0.153
Software & Services                               -0.1777      0.021     -8.281
0.000      -0.220      -0.136
Technology Hardware & Equipment                   -0.2896      0.031     -9.493
0.000      -0.349      -0.230
Telecommunication Services                        -0.0892      0.066     -1.356
0.175      -0.218       0.040
Transportation                                    -0.3266      0.041     -8.006
0.000      -0.407      -0.247
Utilities                                         -0.1498      0.035     -4.229
0.000      -0.219      -0.080
==============================================================================
Omnibus:                     4258.963   Durbin-Watson:                   1.871
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          4830140.576
Skew:                           9.864   Prob(JB):                         0.00
Kurtosis:                     210.510   Cond. No.                         3.92
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctl
y specified.
                            OLS Regression Results
==============================================================================
Dep. Variable:            RetLate2020   R-squared:                       0.111
Model:                            OLS   Adj. R-squared:                  0.103
Method:                 Least Squares   F-statistic:                     14.31
Date:                Wed, 07 Jul 2021   Prob (F-statistic):           2.18e-52
Time:                        21:13:22   Log-Likelihood:                -3828.0
No. Observations:                2668   AIC:                             7704.
Df Residuals:                    2644   BIC:                             7845.
Df Model:                          23
Covariance Type:            nonrobust
===============================================================================
===================================
                                                    coef    std err          t
P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
-----------------------------------
Automobiles & Components                          1.8114      0.190      9.558
0.000       1.440       2.183
Banks                                             0.4351      0.059      7.335
0.000       0.319       0.551
```

```
Capital Goods                                          0.9058      0.067     13.606
0.000       0.775       1.036
Commercial  & Professional Services                    0.6055      0.112      5.405
0.000       0.386       0.825
Consumer Durables & Apparel                            1.6140      0.113     14.234
0.000       1.392       1.836
Consumer Services                                      1.1778      0.108     10.888
0.000       0.966       1.390
Diversified Financials                                 0.7446      0.091      8.157
0.000       0.566       0.924
Energy                                                 1.0175      0.100     10.168
0.000       0.821       1.214
Food & Staples Retailing                               0.4041      0.218      1.857
0.063      -0.023       0.831
Food, Beverage & Tobacco                               0.5648      0.134      4.215
0.000       0.302       0.828
Health Care Equipment & Services                       0.7392      0.077      9.555
0.000       0.588       0.891
Household & Personal Products                          0.6115      0.208      2.935
0.003       0.203       1.020
Insurance                                              0.3584      0.111      3.219
0.001       0.140       0.577
Materials                                              0.8805      0.093      9.451
0.000       0.698       1.063
Media & Entertainment                                  0.8663      0.119      7.302
0.000       0.634       1.099
Pharmaceuticals, Biotechnology & Life Sciences         0.7460      0.058     12.828
0.000       0.632       0.860
Real Estate                                            0.4592      0.077      5.986
0.000       0.309       0.610
Retailing                                              1.7979      0.102     17.705
0.000       1.599       1.997
Semiconductors & Semiconductor Equipment               1.2186      0.130      9.402
0.000       0.964       1.473
Software & Services                                    1.0246      0.074     13.766
0.000       0.879       1.171
Technology Hardware & Equipment                        0.8171      0.106      7.721
0.000       0.610       1.025
Telecommunication Services                             0.4901      0.228      2.147
0.032       0.043       0.938
Transportation                                         0.7097      0.142      5.015
0.000       0.432       0.987
Utilities                                              0.1838      0.123      1.496
0.135      -0.057       0.425
==============================================================================
Omnibus:                     2876.628   Durbin-Watson:                   1.967
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           338795.295
Skew:                           5.202   Prob(JB):                         0.00
Kurtosis:                      57.216   Cond. No.                         3.92
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctl
y specified.
```

GICS-3

In [12]:
```
y1 = dfg3['RetEarly2020'].astype('float')
y2 = dfg3['RetLate2020'].astype('float')
x = dfg3['iname']
x = pd.get_dummies(x, columns=['iname'])

mod1 = sm.OLS(y1, x)
res1 = mod1.fit()
print(res1.summary())

mod2 = sm.OLS(y2, x)
res2 = mod2.fit()
print(res2.summary())

gresults = gresults.append({'Adj_rsqrd_Early': res1.rsquared_adj, 'Adj_rsqrd_Late
                            'Sig_Coef_Early': (res1.pvalues < .05).sum(), 'Sig_Coe
```

```
                            OLS Regression Results
==============================================================================
==
Dep. Variable:           RetEarly2020   R-squared:                       0.1
69
Model:                            OLS   Adj. R-squared:                  0.1
48
Method:                 Least Squares   F-statistic:                     7.9
26
Date:                Wed, 07 Jul 2021   Prob (F-statistic):           1.43e-
63
Time:                        21:13:22   Log-Likelihood:                -452.
38
No. Observations:                2594   AIC:                             103
7.
Df Residuals:                    2528   BIC:                             142
4.
Df Model:                          65
Covariance Type:            nonrobust
```

GICS-4

In [13]:
```
y1 = dfg4['RetEarly2020'].astype('float')
y2 = dfg4['RetLate2020'].astype('float')
x = dfg4['iname']
x = pd.get_dummies(x, columns=['iname'])

mod1 = sm.OLS(y1, x)
res1 = mod1.fit()
print(res1.summary())

mod2 = sm.OLS(y2, x)
res2 = mod2.fit()
print(res2.summary())

gresults = gresults.append({'Adj_rsqrd_Early': res1.rsquared_adj, 'Adj_rsqrd_Late
                           'Sig_Coef_Early': (res1.pvalues < .05).sum(), 'Sig_Coe
```

```
                            OLS Regression Results                            
============================================================================
==
Dep. Variable:            RetEarly2020   R-squared:                       0.2
05
Model:                             OLS   Adj. R-squared:                  0.1
58
Method:                  Least Squares   F-statistic:                     4.3
35
Date:                 Wed, 07 Jul 2021   Prob (F-statistic):           1.47e-
52
Time:                         21:13:22   Log-Likelihood:                -393.
99
No. Observations:                 2582   AIC:                             108
0.
Df Residuals:                     2436   BIC:                             193
5.
Df Model:                          145                                         
Covariance Type:             nonrobust                                         
```

NAICS

NAICS-1

In [14]:
```
nresults = pd.DataFrame()
```

In [15]:
```
y1 = dfn1['RetEarly2020'].astype('float')
y2 = dfn1['RetLate2020'].astype('float')
x = dfn1['iname']
x = pd.get_dummies(x, columns=['iname'])

mod1 = sm.OLS(y1, x)
res1 = mod1.fit()
print(res1.summary())

mod2 = sm.OLS(y2, x)
res2 = mod2.fit()
print(res2.summary())


nresults = nresults.append({'Adj_rsqrd_Early': res1.rsquared_adj, 'Adj_rsqrd_Late
                           'Sig_Coef_Early': (res1.pvalues < .05).sum(), 'Sig_Coe
```

```
                            OLS Regression Results
==============================================================================
==
Dep. Variable:           RetEarly2020   R-squared:                       0.0
71
Model:                            OLS   Adj. R-squared:                  0.0
65
Method:                 Least Squares   F-statistic:                     11.
96
Date:                Wed, 07 Jul 2021   Prob (F-statistic):           1.48e-
32
Time:                        21:13:22   Log-Likelihood:                -597.
24
No. Observations:                2666   AIC:                             123
0.
Df Residuals:                    2648   BIC:                             133
6.
Df Model:                          17
Covariance Type:            nonrobust
```

NAICS-2

In [16]:
```
y1 = dfn2['RetEarly2020'].astype('float')
y2 = dfn2['RetLate2020'].astype('float')
x = dfn2['iname']
x = pd.get_dummies(x, columns=['iname'])

mod1 = sm.OLS(y1, x)
res1 = mod1.fit()
print(res1.summary())

mod2 = sm.OLS(y2, x)
res2 = mod2.fit()
print(res2.summary())

nresults = nresults.append({'Adj_rsqrd_Early': res1.rsquared_adj, 'Adj_rsqrd_Late
                           'Sig_Coef_Early': (res1.pvalues < .05).sum(), 'Sig_Coe
```

```
                            OLS Regression Results
==============================================================================
==
Dep. Variable:           RetEarly2020   R-squared:                       0.1
45
Model:                            OLS   Adj. R-squared:                  0.1
18
Method:                 Least Squares   F-statistic:                     5.5
34
Date:                Wed, 07 Jul 2021   Prob (F-statistic):           3.79e-
46
Time:                        21:13:23   Log-Likelihood:              -487.
65
No. Observations:                2666   AIC:                             113
5.
Df Residuals:                    2586   BIC:                             160
6.
Df Model:                          79
Covariance Type:            nonrobust
```

NAICS-3

In [17]:
```
y1 = dfn3['RetEarly2020'].astype('float')
y2 = dfn3['RetLate2020'].astype('float')
x = dfn3['iname']
x = pd.get_dummies(x, columns=['iname'])

mod1 = sm.OLS(y1, x)
res1 = mod1.fit()
print(res1.summary())

mod2 = sm.OLS(y2, x)
res2 = mod2.fit()
print(res2.summary())

nresults = nresults.append({'Adj_rsqrd_Early': res1.rsquared_adj, 'Adj_rsqrd_Late
                           'Sig_Coef_Early': (res1.pvalues < .05).sum(), 'Sig_Coe
```

```
                            OLS Regression Results
============================================================================
==
Dep. Variable:           RetEarly2020   R-squared:                       0.1
89
Model:                            OLS   Adj. R-squared:                  0.1
13
Method:                 Least Squares   F-statistic:                     2.4
84
Date:                Wed, 07 Jul 2021   Prob (F-statistic):           1.81e-
26
Time:                        21:13:23   Log-Likelihood:                -416.
11
No. Observations:                2666   AIC:                             129
2.
Df Residuals:                    2436   BIC:                             264
7.
Df Model:                         229
Covariance Type:            nonrobust
```

NAICS-4

In [18]:
```
y1 = dfn4['RetEarly2020'].astype('float')
y2 = dfn4['RetLate2020'].astype('float')
x = dfn4['iname']
x = pd.get_dummies(x, columns=['iname'])

mod1 = sm.OLS(y1, x)
res1 = mod1.fit()
print(res1.summary())

mod2 = sm.OLS(y2, x)
res2 = mod2.fit()
print(res2.summary())

nresults = nresults.append({'Adj_rsqrd_Early': res1.rsquared_adj, 'Adj_rsqrd_Late
                           'Sig_Coef_Early': (res1.pvalues < .05).sum(), 'Sig_Coe
```

```
                            OLS Regression Results                            
=============================================================================
==
Dep. Variable:           RetEarly2020   R-squared:                       0.2
20
Model:                            OLS   Adj. R-squared:                  0.0
89
Method:                 Least Squares   F-statistic:                     1.6
72
Date:                Wed, 07 Jul 2021   Prob (F-statistic):           1.10e-
12
Time:                        21:13:24   Log-Likelihood:                 -364.
31
No. Observations:                2666   AIC:                             150
1.
Df Residuals:                    2280   BIC:                             377
4.
Df Model:                         385                                         
Covariance Type:            nonrobust                                         
```

SIC

SIC-1

In [19]:
```
sresults = pd.DataFrame()
```

In [20]:
```
y1 = dfs1['RetEarly2020'].astype('float')
y2 = dfs1['RetLate2020'].astype('float')
x = dfs1['iname']
x = pd.get_dummies(x, columns=['iname'])

mod1 = sm.OLS(y1, x)
res1 = mod1.fit()
print(res1.summary())

mod2 = sm.OLS(y2, x)
res2 = mod2.fit()
print(res2.summary())

sresults = sresults.append({'Adj_rsqrd_Early': res1.rsquared_adj, 'Adj_rsqrd_Late
                           'Sig_Coef_Early': (res1.pvalues < .05).sum(), 'Sig_Coe
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:           RetEarly2020   R-squared:                       0.054
Model:                            OLS   Adj. R-squared:                  0.051
Method:                 Least Squares   F-statistic:                     16.93
Date:                Wed, 07 Jul 2021   Prob (F-statistic):           1.93e-27
Time:                        21:13:25   Log-Likelihood:                -621.03
No. Observations:                2668   AIC:                             1262.
Df Residuals:                    2658   BIC:                             1321.
Df Model:                           9
Covariance Type:            nonrobust
================================================================================
==========================================
                                                            coef    std err
t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
------------------------------------------
Agriculture, Forestry, And Fishing                       -0.1849      0.116
-1.599      0.110      -0.412       0.042
Construction                                             -0.3744      0.052
-7.239      0.000      -0.476      -0.273
Finance, Insurance, And Real Estate                      -0.3429      0.012
-29.506      0.000      -0.366      -0.320
Manufacturing                                            -0.2471      0.010
-25.953      0.000      -0.266      -0.228
Mining                                                   -0.6046      0.035
-17.114      0.000      -0.674      -0.535
Public Administration                                    -0.2885      0.177
-1.633      0.103      -0.635       0.058
Retail Trade                                             -0.3897      0.026
-14.852      0.000      -0.441      -0.338
Services                                                 -0.2429      0.015
-16.290      0.000      -0.272      -0.214
Transport, Comm, Electric, Gas, And Sanitary Services    -0.2513      0.022
-11.643      0.000      -0.294      -0.209
Wholesale Trade                                          -0.3290      0.038
-8.603      0.000      -0.404      -0.254
==============================================================================
Omnibus:                     4090.592   Durbin-Watson:                   1.863
```

```
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          3873577.493
Skew:                           9.120   Prob(JB):                         0.00
Kurtosis:                     188.774   Cond. No.                         18.6
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctl
y specified.
                            OLS Regression Results
==============================================================================
Dep. Variable:            RetLate2020   R-squared:                       0.068
Model:                            OLS   Adj. R-squared:                  0.065
Method:                 Least Squares   F-statistic:                     21.57
Date:                Wed, 07 Jul 2021   Prob (F-statistic):           1.30e-35
Time:                        21:13:25   Log-Likelihood:                -3890.5
No. Observations:                2668   AIC:                             7801.
Df Residuals:                    2658   BIC:                             7860.
Df Model:                           9
Covariance Type:            nonrobust
===============================================================================
==========================================
                                                             coef    std err
t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
------------------------------------------
Agriculture, Forestry, And Fishing                         0.1579      0.394
0.401      0.689      -0.614       0.930
Construction                                               1.1497      0.176
6.527      0.000       0.804       1.495
Finance, Insurance, And Real Estate                        0.4889      0.040
12.350      0.000       0.411       0.566
Manufacturing                                              0.8960      0.032
27.637      0.000       0.832       0.960
Mining                                                     1.2435      0.120
10.335      0.000       1.008       1.479
Public Administration                                      0.3538      0.602
0.588      0.557      -0.826       1.533
Retail Trade                                               1.5480      0.089
17.325      0.000       1.373       1.723
Services                                                   0.9691      0.051
19.083      0.000       0.870       1.069
Transport, Comm, Electric, Gas, And Sanitary Services     0.4564      0.073
6.210      0.000       0.312       0.601
Wholesale Trade                                            0.7763      0.130
5.960      0.000       0.521       1.032
==============================================================================
Omnibus:                     2892.288   Durbin-Watson:                   1.978
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           353429.393
Skew:                           5.238   Prob(JB):                         0.00
Kurtosis:                      58.403   Cond. No.                         18.6
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctl
y specified.
```

SIC-2

In [21]:
```
y1 = dfs2['RetEarly2020'].astype('float')
y2 = dfs2['RetLate2020'].astype('float')
x = dfs2['iname']
x = pd.get_dummies(x, columns=['iname'])

mod1 = sm.OLS(y1, x)
res1 = mod1.fit()
print(res1.summary())

mod2 = sm.OLS(y2, x)
res2 = mod2.fit()
print(res2.summary())

sresults = sresults.append({'Adj_rsqrd_Early': res1.rsquared_adj, 'Adj_rsqrd_Late
                            'Sig_Coef_Early': (res1.pvalues < .05).sum(), 'Sig_Coe
```

```
                            OLS Regression Results
=============================================================================
==
Dep. Variable:           RetEarly2020   R-squared:                       0.1
28
Model:                            OLS   Adj. R-squared:                  0.1
06
Method:                 Least Squares   F-statistic:                     5.7
98
Date:                Wed, 07 Jul 2021   Prob (F-statistic):           2.64e-
42
Time:                        21:13:25   Log-Likelihood:                 -512.
29
No. Observations:                2668   AIC:                             115
9.
Df Residuals:                    2601   BIC:                             155
3.
Df Model:                          66
Covariance Type:            nonrobust
```

SIC-3

In [22]:
```
y1 = dfs3['RetEarly2020'].astype('float')
y2 = dfs3['RetLate2020'].astype('float')
x = dfs3['iname']
x = pd.get_dummies(x, columns=['iname'])

mod1 = sm.OLS(y1, x)
res1 = mod1.fit()
print(res1.summary())

mod2 = sm.OLS(y2, x)
res2 = mod2.fit()
print(res2.summary())

sresults = sresults.append({'Adj_rsqrd_Early': res1.rsquared_adj, 'Adj_rsqrd_Late
                           'Sig_Coef_Early': (res1.pvalues < .05).sum(), 'Sig_Coe
```

```
                            OLS Regression Results
==============================================================================
==
Dep. Variable:           RetEarly2020   R-squared:                       0.1
79
Model:                            OLS   Adj. R-squared:                  0.1
09
Method:                 Least Squares   F-statistic:                     2.5
70
Date:                Wed, 07 Jul 2021   Prob (F-statistic):           3.12e-
25
Time:                        21:13:25   Log-Likelihood:                -446.
62
No. Observations:                2526   AIC:                             128
9.
Df Residuals:                    2328   BIC:                             244
4.
Df Model:                         197
Covariance Type:            nonrobust
```

SIC-4

In [23]:
```
y1 = dfs4['RetEarly2020'].astype('float')
y2 = dfs4['RetLate2020'].astype('float')
x = dfs4['iname']
x = pd.get_dummies(x, columns=['iname'])

mod1 = sm.OLS(y1, x)
res1 = mod1.fit()
print(res1.summary())

mod2 = sm.OLS(y2, x)
res2 = mod2.fit()
print(res2.summary())

sresults = sresults.append({'Adj_rsqrd_Early': res1.rsquared_adj, 'Adj_rsqrd_Late
                           'Sig_Coef_Early': (res1.pvalues < .05).sum(), 'Sig_Coe
```

```
                            OLS Regression Results
==============================================================================
==
Dep. Variable:           RetEarly2020   R-squared:                       0.2
08
Model:                            OLS   Adj. R-squared:                  0.0
99
Method:                 Least Squares   F-statistic:                     1.9
05
Date:                Wed, 07 Jul 2021   Prob (F-statistic):            5.31e-
12
Time:                        21:13:26   Log-Likelihood:                -438.
49
No. Observations:                1771   AIC:                             130
7.
Df Residuals:                    1556   BIC:                             248
5.
Df Model:                         214
Covariance Type:            nonrobust
```

Results

Comparing GICS/NAICS/SIC

The tables below show the regression results for models by:

- Type of Industry Indicator (GICS, NAICS, SIC)
- Early vs Late 2020
- Level of Specificity (Num_Unique)
    - Broad Industry - Fewer unique industries
    - Specific Industry - Many unique industries

GICS

In [24]:
```
gresults.index = [len(dfg1.iname.unique()), len(dfg2.iname.unique()), len(dfg3.ir
gresults.index.name = 'Num_Unique'
gresults
```

Out[24]:

Num_Unique	Adj_rsqrd_Early	Adj_rsqrd_Late	Sig_Coef_Early	Sig_Coef_Late
11	0.115457	0.092359	11.0	10.0
24	0.122945	0.102984	23.0	22.0
66	0.147931	0.136639	56.0	50.0
146	0.157788	0.183491	112.0	84.0

NAICS

In [25]:
```
nresults.index  = [len(dfn1.iname.unique()), len(dfn2.iname.unique()), len(dfn3.i
nresults.index.name = 'Num_Unique'
nresults
```

Out[25]:

Num_Unique	Adj_rsqrd_Early	Adj_rsqrd_Late	Sig_Coef_Early	Sig_Coef_Late
18	0.065346	0.069405	17.0	14.0
80	0.118474	0.123370	66.0	58.0
230	0.113097	0.149253	132.0	104.0
386	0.088525	0.170758	171.0	115.0

SIC

In [26]:
```
sresults.index = [ len(dfs1.iname.unique()), len(dfs2.iname.unique()), len(dfs3.i
sresults.index.name = 'Num_Unique'
sresults
```

Out[26]:

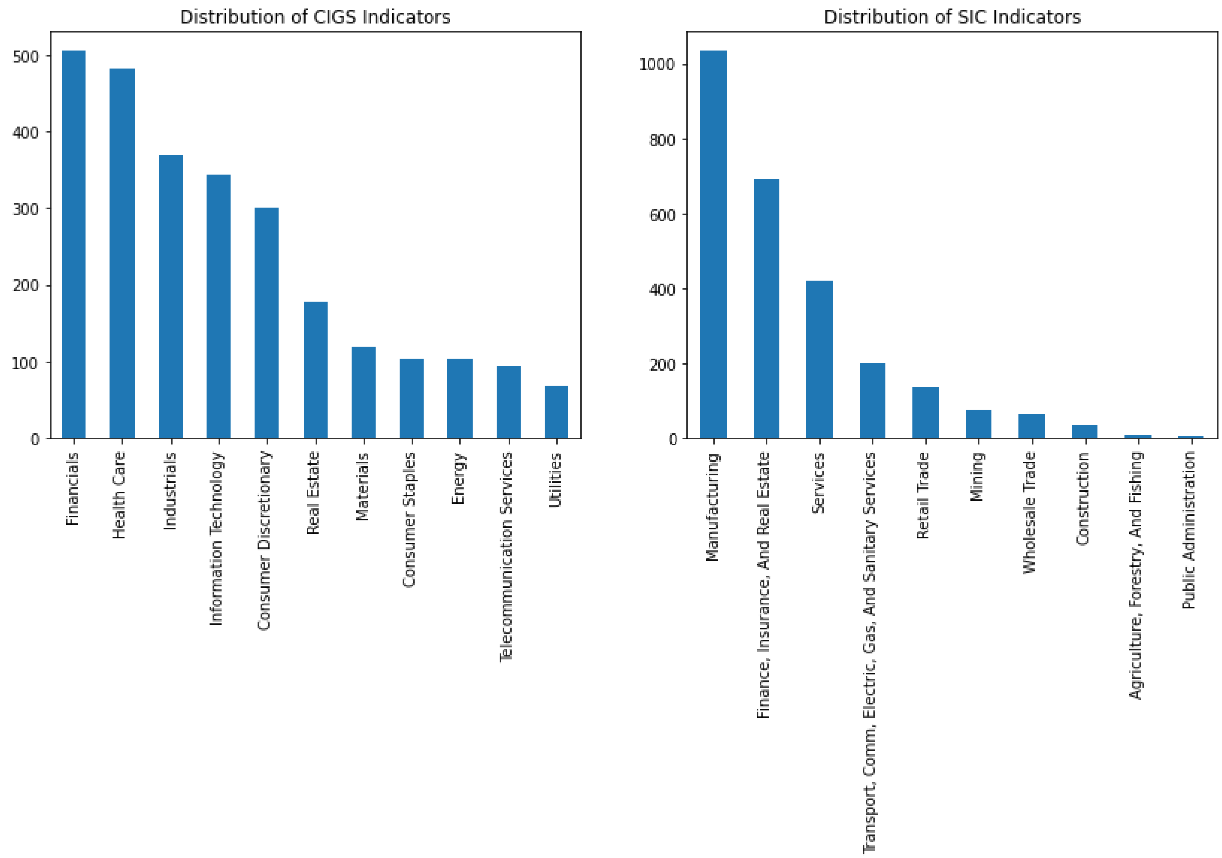
Num_Unique	Adj_rsqrd_Early	Adj_rsqrd_Late	Sig_Coef_Early	Sig_Coef_Late
10	0.051015	0.064907	8.0	8.0
67	0.106131	0.114422	56.0	48.0
198	0.109120	0.115062	128.0	87.0
215	0.098662	0.136705	96.0	75.0

In general as the specificity of an industry indicator system increases, the explanitory power of the fixed effect regressions also increases. However, this increases comes with the tradeoff that it becomes more difficult to interpret the results when there are so many industry coefficients.

Since we are comparing adjusted r-squared values, having more x variables will penalize adjusted r-squared. For NAICS and SIC indicators, the adjusted r-squared for early 2020 actually decreases at the most specific indicator level.

For this application, the GICS indicators seem to do better explaining early and late 2020 returns compared to NAICS and SIC, especially when the number of unique indicators is low.

In [27]:
```
fig, (ax1, ax2) = plt.subplots(1,2, figsize=(14,5))
ax1.title.set_text('Distribution of CIGS Indicators')
ax2.title.set_text('Distribution of SIC Indicators')
dfg1.iname.value_counts().plot.bar(ax = ax1)
dfs1.iname.value_counts().plot.bar(ax = ax2)
plt.show()
```



One reason why GICS performed better could be that the companies in our dataset are more balanced by industry than in SIC or NAICS.

The above plot shows which industry indicators make up our dataset for GICS and SIC. For SIC, almost half of companies are classified as "Manufacturing". SIC also has very broad catch-all indicators like "Finance, Insurance, and Real Estate" that would group very different companies together.

Though it is not a uniform distribution, the GICS indicators are much more balanced in this dataset.

Comparing Broad vs Specific Indicators - GICS

We saw earlier that using more numerous and specific industry indicators explains more of the variation in early and late 2020 return outcomes. This is likely because using the broad indicators groups together companies with very different business models, where the specific indicators mean that companies with the same code are more likely to be similar.

Below, regression results are compared for the most broad (Sector) and most specific (sub-industry) GICS indicators.

In [96]:
```
# Sector Regression
y1 = dfg1['RetEarly2020'].astype('float')
y2 = dfg1['RetLate2020'].astype('float')
x = dfg1.icode.astype('int').astype('str') + '_' + dfg1.iname
x = pd.get_dummies(x, columns=['iname'])
mod1 = sm.OLS(y1, x)
res1 = mod1.fit()

# Sub-Industry Regression
y1 = dfg4['RetEarly2020'].astype('float')
y2 = dfg4['RetLate2020'].astype('float')
x = dfg4.icode.astype('int').astype('str') + '_' + dfg4.iname
x = pd.get_dummies(x, columns=['iname'])
mod2 = sm.OLS(y1, x)
res2 = mod2.fit()
```

Information Technology

The information sector has an average early 2020 return of -21.7%. When we compare that to the averages of the sub-industries that fall within the larger IT sector, we see that most are relatively close together.

In [145]:
```
# IT sector
res1.params[7:8]
```

```
45_Information Technology   -0.216825
dtype: float64
```

In [98]:
```
# IT sub-industries
res2.params[113:125]
```

Out[98]:
```
45102010_IT Consulting & Other Services                 -0.236266
45102020_Data Processing & Outsourced Services          -0.312362
45103010_Application Software                           -0.159787
45103020_Systems Software                               -0.101528
45201020_Communications Equipment                       -0.246546
45202030_Technology Hardware, Storage & Peripherals    -0.316359
45203010_Electronic Equipment & Instruments             -0.297740
45203015_Electronic Components                          -0.319009
45203020_Electronic Manufacturing Services              -0.289516
45203030_Technology Distributors                        -0.352021
45301010_Semiconductor Equipment                        -0.221918
45301020_Semiconductors                                 -0.229218
dtype: float64
```

Health Care

The Health Care sector has an average early 2020 return of -14%. When we compare that to the averages of the sub-industries that fall within the larger sector, we see that there is much more variation. Returns range from -35% at the lowest to 3% at the highest.

This is one sector that likely benefits from the increased specificity of having more industry indicators.

In [99]:
```
res1.params[5:6]
```

Out[99]:
```
35_Health Care   -0.139677
dtype: float64
```

In [100]:
```
res2.params[87:97]
```

Out[100]:
```
35101010_Health Care Equipment            -0.090319
35101020_Health Care Supplies             -0.163101
35102010_Health Care Distributors          0.029810
35102015_Health Care  Services            -0.179376
35102020_Health Care Facilities           -0.350752
35102030_Managed Health Care              -0.191974
35103010_Health Care Technology           -0.089736
35201010_Biotechnology                    -0.113588
35202010_Pharmaceuticals                  -0.222985
35203010_Life Sciences Tools & Services   -0.173646
dtype: float64
```

Out of Sample Regressions

Another issue with the very specific industry indicators is that there might not be enough companies in the dataset to accurately generalize and make accurate out of sample predictions.

Below are results of out of sample predictions for 2 regressions; one with very broad industry indicators and one with specific industry indicators.

In [128]:
```
# Sector Regression - Broad
y1 = dfg1['RetEarly2020'].astype('float')
y2 = dfg1['RetLate2020'].astype('float')
x = dfg1.icode.astype('int').astype('str') + '_' + dfg1.iname
x = pd.get_dummies(x, columns=['iname'])

X_train, X_test, y_train, y_test = train_test_split(x, y1, test_size=0.2, random_

reg1 = LinearRegression().fit(X_train, y_train)
pred = reg1.predict(X_test)

mse1 = np.sqrt(np.mean((y_test - pred)**2))
mse1
```

Out[128]:
```
0.28156881756342594
```

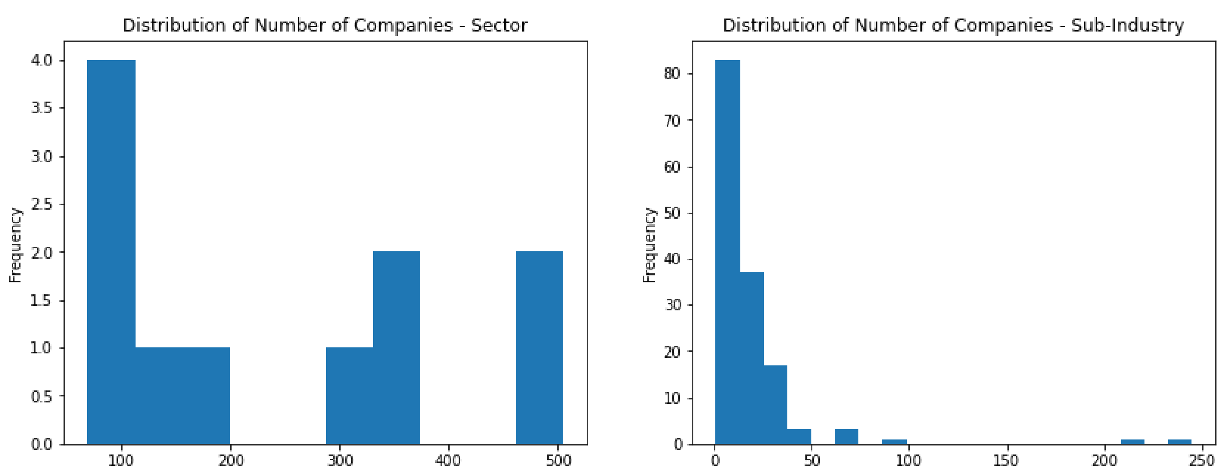
In [127]:
```
# Industry Regression - Specific
y1 = dfg3['RetEarly2020'].astype('float')
y2 = dfg3['RetLate2020'].astype('float')
x = dfg3.icode.astype('int').astype('str') + '_' + dfg3.iname
x = pd.get_dummies(x, columns=['iname'])

X_train, X_test, y_train, y_test = train_test_split(x, y1, test_size=0.2, random_

reg2 = LinearRegression().fit(X_train, y_train)
pred = reg2.predict(X_test)

mse2 = np.sqrt(np.mean((y_test - pred)**2))
mse2
```

Out[127]: 0.3062310828465986

For out of sample prediction, the broad industry indicators actually have a lower RMSE than the second most specific (industry). The most specific indicator (sub-industry) could not be used since there were so many codes that only had a single company.

With so many different industries, the specific indicator regression likely does not have enough data to make good predictions.

In [142]:
```
fig, (ax1, ax2) = plt.subplots(1,2, figsize=(14,5))
ax1.title.set_text('Distribution of Number of Companies - Sector')
ax2.title.set_text('Distribution of Number of Companies - Sub-Industry')
dfg1.iname.value_counts().plot.hist(ax = ax1)
dfg4.iname.value_counts().plot.hist(ax = ax2, bins = 20)
plt.show()
```



The above two plots show the distribution of the number of companies in our dataset that have the same classification code.

- For the broad classification on the left, all of the sectors have at least or nearly 100 different companies with the same classification.
- For the specific classification on the right, almost all of the sub-industries have less than 25 different companies with the same classification. Less than 12 is by far the most common.

With these results, it makes sense that the very specific classifications would have a difficult time generalizing and making accurate out of sample predictions.

In [ ]:

```

```