US Stocks Fundamentals

Zhengkai Zhang

December 7, 2016

CS 170A - Professor Parker

Contents

Introduction

1.1 Data Description and background

This dataset contains US stocks fundamental data, such as income statement, balance sheet and cash flows.

- 12,129 companies
- 8,526 unique indicators
- ~20 indicators comparable across most companies
- Five years of data, yearly

The data is provided by http://usfundamentals.com.

I picked this dataset as my final project Since I really interesting on some of companies in this dataset which I hope I can work with in the future. In this project, I will virtualize the dataset, and try to analysis the relationship between each indicators of these companies using PCA,R-value and data description by table or graphs which I learned in this course.

The main script language I will use in this project is Python.

1.2 Data Select

The main dataset has a large amount of indicators which not every indicators assign to every companies,So I would like to pick some indicators share by large number of companies to do the analysis here.

Top 25 indicators share by companies:

```
Assets                                        6886.0
LiabilitiesAndStockholdersEquity              6789.0
StockholdersEquity                            6095.0
CashAndCashEquivalentsAtCarryingValue         5416.0
NetCashProvidedByUsedInOperatingActivities    4738.0
NetIncomeLoss                                 4490.0
NetCashProvidedByUsedInFinancingActivities    4374.0
CommonStockSharesAuthorized                   3939.0
CashAndCashEquivalentsPeriodIncreaseDecrease  3836.0
CommonStockValue                              3579.0
CommonStockSharesIssued                       3371.0
RetainedEarningsAccumulatedDeficit            3000.0
CommonStockParOrStatedValuePerShare           2749.0
NetCashProvidedByUsedInInvestingActivities    2628.0
PropertyPlantAndEquipmentNet                  2378.0
AssetsCurrent                                 2017.0
LiabilitiesCurrent                            1992.0
CommonStockSharesOutstanding                  1606.0
Liabilities                                   1141.0
OperatingIncomeLoss                           1067.0
IncomeTaxExpenseBenefit                        847.0
InterestExpense                                558.0
ShareBasedCompensation                         481.0
PaymentsToAcquirePropertyPlantAndEquipment     400.0
AccumulatedOtherComprehensiveIncomeLossNetOfTax  286.0
```

Part of Python Script to find first 25 popular indicators:

```
df=pd.read_csv('indicators_by_company.csv')
#number of indicators by company
df_ind_count = pd.concat([ df[['company_id', 'indicator_id', '2010']].dropna().groupby('company_id')['indicator_id'].co
,df[['company_id', 'indicator_id', '2011']].dropna().groupby('company_id')['indicator_id'].count()
,df[['company_id', 'indicator_id', '2012']].dropna().groupby('company_id')['indicator_id'].count()
,df[['company_id', 'indicator_id', '2013']].dropna().groupby('company_id')['indicator_id'].count()
,df[['company_id', 'indicator_id', '2014']].dropna().groupby('company_id')['indicator_id'].count()
,df[['company_id', 'indicator_id', '2015']].dropna().groupby('company_id')['indicator_id'].count()
,df[['company_id', 'indicator_id', '2016']].dropna().groupby('company_id')['indicator_id'].count()
], axis=1)
df_ind_count.columns=['2010','2011','2012','2013','2014','2015','2016']
#df_ind_count.head()
df_comp_count = pd.concat([
df[['company_id', 'indicator_id', '2010']].dropna().groupby('indicator_id')['company_id'].count().sort_values(ascending
df[['company_id', 'indicator_id', '2011']].dropna().groupby('indicator_id')['company_id'].count().sort_values(ascending
df[['company_id', 'indicator_id', '2012']].dropna().groupby('indicator_id')['company_id'].count().sort_values(ascending
df[['company_id', 'indicator_id', '2013']].dropna().groupby('indicator_id')['company_id'].count().sort_values(ascending
df[['company_id', 'indicator_id', '2014']].dropna().groupby('indicator_id')['company_id'].count().sort_values(ascending
df[['company_id', 'indicator_id', '2015']].dropna().groupby('indicator_id')['company_id'].count().sort_values(ascending
df[['company_id', 'indicator_id', '2016']].dropna().groupby('indicator_id')['company_id'].count().sort_values(ascending
], axis=1)

df_comp_count.columns=['2010','2011','2012','2013','2014','2015','2016']
```

```
list_s_int=[]
for c in df_comp_count.columns:
 df_comp_count.sort_values(c, axis=0, ascending=False ,inplace=True)
 li=df_comp_count.index
 s_int = pd.Series(np.zeros(len(li)), index=li)
 s1=df.loc[((df['indicator_id']==li[0]) & (df[c].notnull())),'company_id'].unique()
 s_int[li[0]]=len(s1)
 for i in range(1,len(li)):
  s2=df.loc[((df['indicator_id']==li[i]) & (df[c].notnull())),'company_id'].unique()
  s1=pd.Series(np.intersect1d(s1, s2))
  s_int[li[i]]=len(s1)
 list_s_int.append(s_int)

df_comp_int_count = pd.concat(list_s_int, axis=1)
df_comp_int_count.columns=['2010','2011','2012','2013','2014','2015','2016']
```

Selected Data indicators:

Assets
LiabilitiesAndStockholdersEquity
NetIncomeLoss
IncomeTaxExpenseBenefit
CommonStockSharesAuthorized
CommonStockValue
CommonStockSharesIssued
AssetsCurrent
OperatingIncomeLoss
PropertyPlantAndEquipmentNet

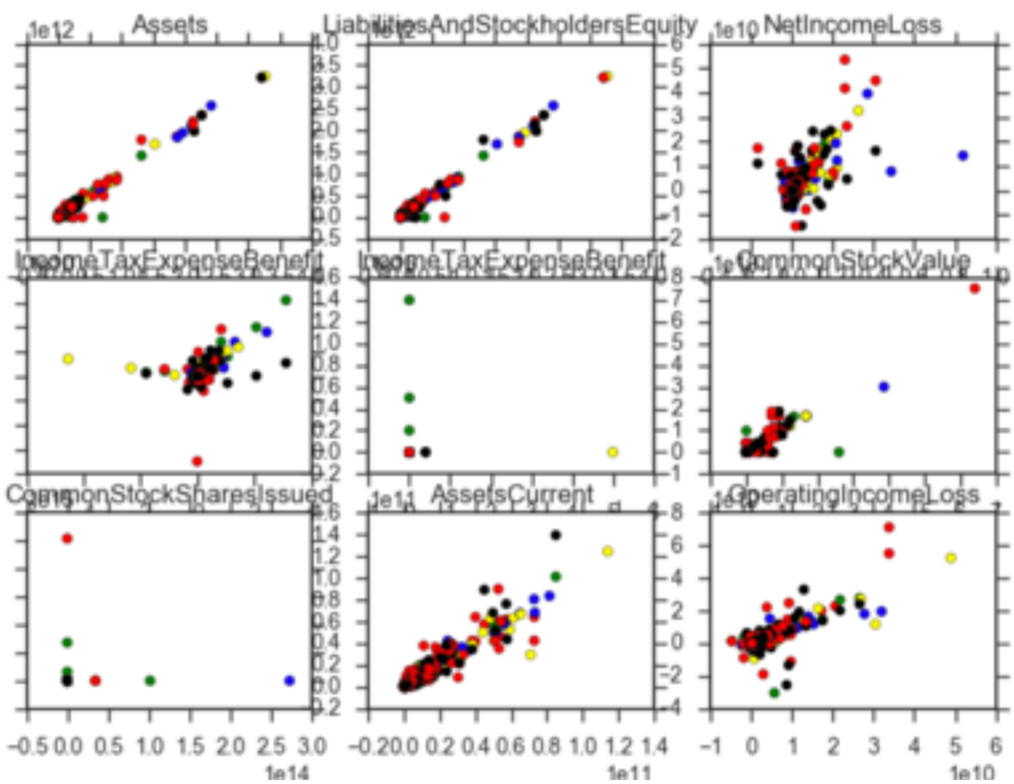
There are 2250 companies with not null values in the data set for these 10 indicators

These are the ten data indicators I want to use in my project which all shared with 2250 companies.

Analysis

2.1 Virtualization and mean of data

Virtualization of dataset indicators by years:



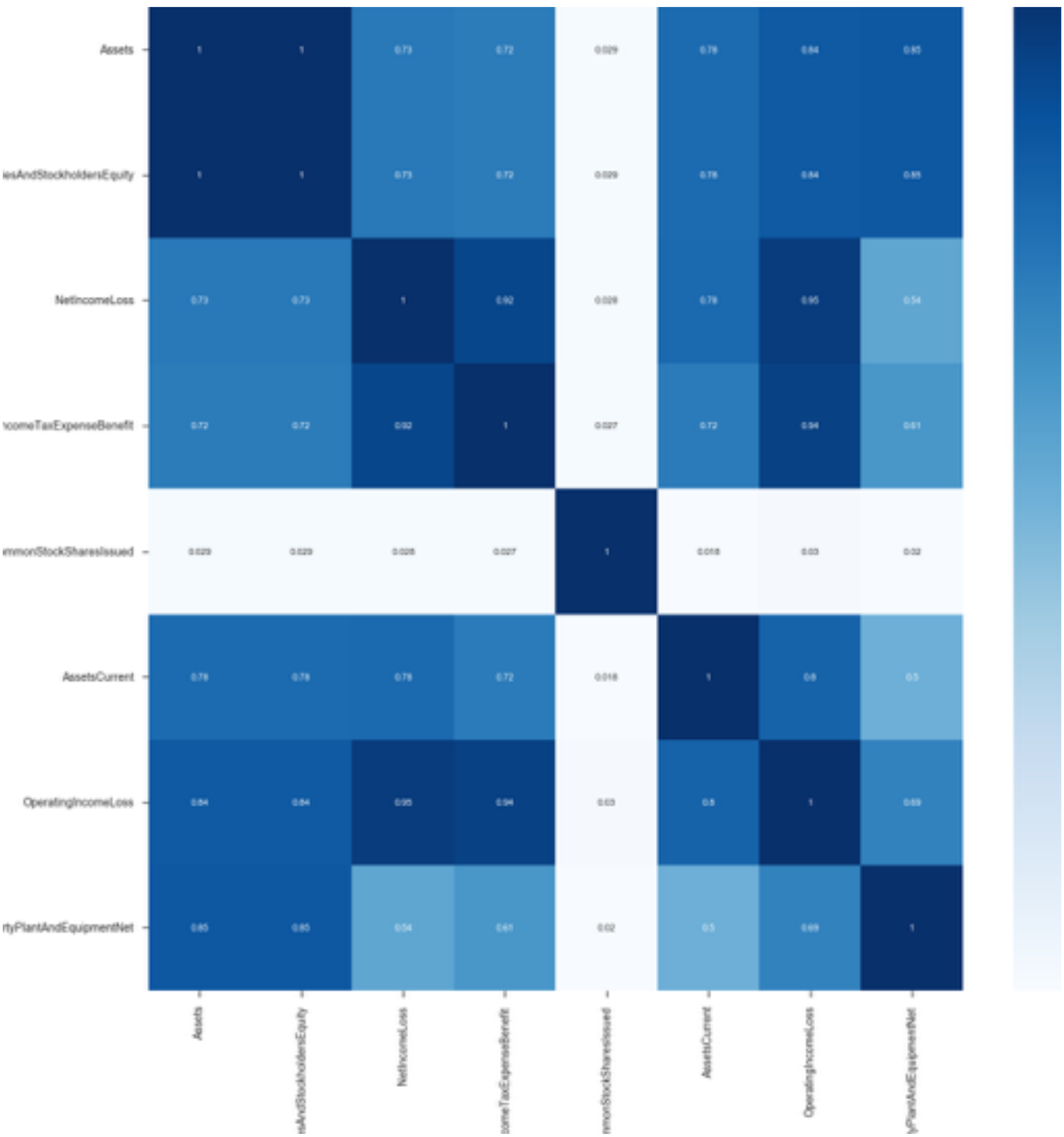
As Graphs show above, I virtualized these nine indicators by years of 2011,2012,2013,2014, and 2015 represent by colors red, green, blue, yellow ,and black. All these indicators have a different shape of scatter graph. Assets and LiabilitiesAndStockholdersEquity both represent by a scatter graph mostly linearly distributed, and NetIncomeLoss, IncomeTaxExpenseBenefit, AssetsCurrent, and OperatingIncomeLoss are distribute by 5 different shape with some data concentrated at one or two center. IncomeTaxExpenseBenefit and CommonStockSharesIssued are shaped similar in graph with few points.

Part of Python Script to make scatter graph:

```
companies_pd = pd.read_csv("companies.csv")
indicator_pd = pd.read_csv("indicators_by_company.csv")
balance_pd = pd.merge(companies_pd,indicator_pd,how='outer',on="company_id")
asset_col = balance_pd['indicator_id'] == "Assets"
asset_pd = balance_pd[asset_col]
plt.subplot(331)
plt.scatter(asset_pd['2011'], asset_pd['2012'], c=["red", "green"])
plt.scatter(asset_pd['2013'], asset_pd['2014'], c=["blue", "yellow"])
plt.scatter(asset_pd['2011'], asset_pd['2015'], c=["red", "black"])
plt.title('Assets')
#plt.show()
plt.subplot(332)
asset_col = balance_pd['indicator_id'] == "LiabilitiesAndStockholdersEquity"
asset_pd = balance_pd[asset_col]
plt.scatter(asset_pd['2011'], asset_pd['2012'], c=["red", "green"])
plt.scatter(asset_pd['2013'], asset_pd['2014'], c=["blue", "yellow"])
plt.scatter(asset_pd['2011'], asset_pd['2015'], c=["red", "black"])
plt.title('LiabilitiesAndStockholdersEquity')
plt.subplot(333)
asset_col = balance_pd['indicator_id'] == "NetIncomeLoss"
asset_pd = balance_pd[asset_col]
plt.scatter(asset_pd['2011'], asset_pd['2012'], c=["red", "green"])
plt.scatter(asset_pd['2013'], asset_pd['2014'], c=["blue", "yellow"])
plt.scatter(asset_pd['2011'], asset_pd['2015'], c=["red", "black"])
plt.title('NetIncomeLoss')
#plt.show()
plt.subplot(334)
asset_col = balance_pd['indicator_id'] == "IncomeTaxExpenseBenefit"
asset_pd = balance_pd[asset_col]
plt.scatter(asset_pd['2011'], asset_pd['2012'], c=["red", "green"])
plt.scatter(asset_pd['2013'], asset_pd['2014'], c=["blue", "yellow"])
plt.scatter(asset_pd['2011'], asset_pd['2015'], c=["red", "black"])
plt.title('IncomeTaxExpenseBenefit')
#plt.show()
plt.subplot(335)
asset_col = balance_pd['indicator_id'] == "CommonStockSharesAuthorized"
asset_pd = balance_pd[asset_col]
```

2.2 Relationship and data Description

R-Value:



Relationship between 8 indicators by R-Value:

In statistics, the correlation coefficient r measures the strength and direction of a linear relationship between two variables on a scatterplot. The value of r is always between **+1** and –1. To interpret its value, see which of the following values your correlation r is closest to: Exactly –1.

As the graph show above, the diagonal of this matrix is the r-value to it-self which are all zeros represent by dark blue cell in graph. The r value from 0 to 1 in graph with color from white to light blue to dark blue which shows how strong correlation between two indicators.

The indicator CommonStockSharesIssued has almost none relationship with all other indictors, so both row and col cells contain it is almost white here.

There are some very strong relationship between two different indicators such as asset and LiabilitiesAndStockholdersEquity with r-value 1 which means these two dataset is exactly positive related to each other. NetIncomeLoss and OperatingIncomeLoss are also really close related with r-value 0.95, both NetIncomeLoss and OperatingIncomeLoss relate to asset with r-value 0.73 and 0.72 which is a litter weaker.

There are reasonable r-value between these indicators. It is really easy to understand that why NetIncomeLoss and OperatingIncomeLoss are as strong as 0.95. It is a little surprise to me that asset and LiabilitiesAndStockholdersEquity related with r-value as high as 1, so I check google with the definition of both indicators.

Stockholders' equity represents the equity stake currently held on the books by a firm's equity investors. It is calculated either as a firm's total assets minus its total liabilities or as share capital plus retained earnings minus treasury shares.

Assets are sometimes defined as resources or things of value that are owned by a **company**. Some examples of **assets** which are obvious and will be reported on a **company's** balance sheet include: cash, accounts receivable, inventory, investments, land, buildings, and equipment.

As above definition show, it is still reasonable for assets and LiabilitiesAndStockholdersEquity have the r-values 1 in darkest blue cell.

Part of Python Script for r-value graph:

```
#heatmap visualization
def heatmap(data,title):
    fig, ax = plt.subplots(figsize=(15, 15))
    heatmap = sns.heatmap(data, cmap=plt.cm.Blues,annot=True, annot_kws={"size": 8})
    #ax.xaxis.tick_top()
    ax.set_title(title)
    # rotate
    plt.xticks(rotation=90)
    plt.yticks(rotation=0)
    plt.tight_layout()
    plt.show()
```

```
#skipy linregress
#Pearson Correlation
rvalue = DataFrame(np.nan,index=indicators,columns=indicators)
#PValue
pvalue = DataFrame(np.nan,index=indicators,columns=indicators)
#StdErr
stderr = DataFrame(np.nan,index=indicators,columns=indicators)
```

```
#
for c_X in indicators:
    for c_Y in indicators:
        R=linregress(Values[[c_X,c_Y]])
        rvalue.set_value(c_Y,c_X, R.rvalue)
        pvalue.set_value(c_Y,c_X, R.pvalue)
        stderr.set_value(c_Y,c_X, R.stderr)
```

```
heatmap(rvalue,'R-value')
```

Above defined function can do graph with R-value,P-value, and std error.

Data Description:

	Year	Indicator	count	mean	std	min	25%	50%	75%
0	2011	Assets	6886.0	7.202514e+09	7.278970e+10	0.000000e+00	7582307.50	229456000.0	1.6412
1	2011	LiabilitiesAndStockholdersEquity	6877.0	7.174764e+09	7.282371e+10	-1.885475e+07	6742805.00	223932000.0	1.6232
2	2011	NetIncomeLoss	6124.0	1.527939e+08	1.150205e+09	-1.685500e+10	-2494862.00	229500.0	3.4230
3	2011	IncomeTaxExpenseBenefit	4679.0	8.593623e+07	7.595561e+08	-1.803600e+10	0.00	2233000.0	3.0897
4	2011	CommonStockSharesAuthorizedCommonStockValue	0.0	NaN	NaN	NaN	NaN	NaN	NaN
5	2011	CommonStockSharesIssued	5680.0	2.436211e+10	1.425945e+12	0.000000e+00	11062962.00	33040728.0	8.8405
6	2011	AssetsCurrent	5276.0	8.834772e+08	3.685341e+09	-6.260200e+04	1484145.75	57659134.0	3.9774
7	2011	OperatingIncomeLoss	4933.0	2.531529e+08	1.212602e+09	-4.776000e+09	-1794724.00	2412240.0	9.5271
8	2011	PropertyPlantAndEquipmentNet	5320.0	1.191808e+09	6.160671e+09	0.000000e+00	1769880.50	23227214.5	2.3663
9	2012	Assets	6888.0	7.433265e+09	7.312632e+10	-1.700000e+01	7421109.25	245950000.0	1.8075
10	2012	LiabilitiesAndStockholdersEquity	6869.0	7.426497e+09	7.322370e+10	-3.436755e+06	6526836.00	243098978.0	1.7995
11	2012	NetIncomeLoss	6240.0	1.589616e+08	1.229245e+09	-4.326000e+09	-2682787.00	92204.5	3.7319
12	2012	IncomeTaxExpenseBenefit	4831.0	8.389121e+07	8.568514e+08	-3.483100e+10	0.00	2408000.0	3.2376
13	2012	CommonStockSharesAuthorizedCommonStockValue	0.0	NaN	NaN	NaN	NaN	NaN	NaN
14	2012	CommonStockSharesIssued	5680.0	2.386571e+11	1.399633e+13	-8.537050e+05	11565212.75	34742048.5	9.0243
15	2012	AssetsCurrent	5314.0	9.483120e+08	4.025543e+09	-2.840000e+02	1461378.50	61200500.0	4.3530
16	2012	OperatingIncomeLoss	5033.0	2.451229e+08	1.471333e+09	-3.036300e+10	-2105758.00	1572000.0	9.5883
17	2012	PropertyPlantAndEquipmentNet	5377.0	1.325307e+09	6.873712e+09	0.000000e+00	1662080.00	26512000.0	2.5590
18	2013	Assets	6799.0	7.828172e+09	7.417835e+10	0.000000e+00	8021700.00	276124000.0	2.0848
19	2013	LiabilitiesAndStockholdersEquity	6816.0	7.779820e+09	7.408425e+10	-2.037150e+05	7284328.75	265403500.0	2.0350
20	2013	NetIncomeLoss	6139.0	2.029816e+08	1.742852e+09	-3.573000e+09	-2994502.00	9876.0	4.2678
21	2013	IncomeTaxExpenseBenefit	4928.0	8.469089e+07	9.713848e+08	-4.541500e+10	0.00	2419000.0	3.6013

21	2013	IncomeTaxExpenseBenefit	4928.0	8.469089e+07	9.713848e+08	-4.541500e+10	0.00	2419000.0	3.6013
22	2013	CommonStockSharesAuthorizedCommonStockValue	0.0	NaN	NaN	NaN	NaN	NaN	NaN
23	2013	CommonStockSharesIssued	5634.0	6.042663e+10	3.743528e+12	-1.106000e+09	12622747.50	36478979.0	9.4785
24	2013	AssetsCurrent	5261.0	1.027348e+09	4.455280e+09	0.000000e+00	1702727.00	70404000.0	4.6717
25	2013	OperatingIncomeLoss	4986.0	2.804888e+08	1.453066e+09	-2.234600e+09	-2203175.50	1603210.0	1.0912
26	2013	PropertyPlantAndEquipmentNet	5332.0	1.423857e+09	7.337361e+09	0.000000e+00	1572885.75	27880000.0	2.8705
27	2014	Assets	6697.0	8.355857e+09	7.585223e+10	0.000000e+00	12841170.00	323073352.0	2.3535
28	2014	LiabilitiesAndStockholdersEquity	6709.0	8.313736e+09	7.578285e+10	0.000000e+00	11871222.00	315944195.0	2.3334
29	2014	NetIncomeLoss	6126.0	1.818968e+08	1.206398e+09	-7.224200e+09	-3847372.25	180935.5	5.0636
30	2014	IncomeTaxExpenseBenefit	4992.0	1.094332e+08	5.886248e+08	-2.619000e+09	0.00	2899757.5	3.8992
31	2014	CommonStockSharesAuthorizedCommonStockValue	0.0	NaN	NaN	NaN	NaN	NaN	NaN
32	2014	CommonStockSharesIssued	5559.0	1.675114e+08	1.008625e+09	-2.014000e+06	14134955.50	37847163.0	9.7832
33	2014	AssetsCurrent	5180.0	1.084693e+09	4.599384e+09	0.000000e+00	2509946.00	85633500.0	4.9499
34	2014	OperatingIncomeLoss	4970.0	2.877811e+08	1.417556e+09	-9.445300e+09	-3364695.25	1513519.5	1.1610
35	2014	PropertyPlantAndEquipmentNet	5295.0	1.518621e+09	7.692342e+09	-8.675470e+05	1737315.50	30731000.0	3.1004
36	2015	Assets	5997.0	9.114170e+09	7.567919e+10	0.000000e+00	26126768.00	463601000.0	2.8054
37	2015	LiabilitiesAndStockholdersEquity	6012.0	9.350417e+09	7.877916e+10	0.000000e+00	24327165.00	451729500.0	2.7690
38	2015	NetIncomeLoss	5492.0	1.746629e+08	1.442698e+09	-1.468500e+10	-7063904.75	18177.0	5.3891
39	2015	IncomeTaxExpenseBenefit	4614.0	9.185107e+07	6.159236e+08	-6.065000e+09	0.00	2704500.0	3.7098
40	2015	CommonStockSharesAuthorizedCommonStockValue	0.0	NaN	NaN	NaN	NaN	NaN	NaN
41	2015	CommonStockSharesIssued	4901.0	1.622055e+08	1.388816e+09	0.000000e+00	14559381.00	38359454.0	9.9107
42	2015	AssetsCurrent	4598.0	1.195995e+09	5.185320e+09	0.000000e+00	6232325.00	117133234.0	5.7027
43	2015	OperatingIncomeLoss	4522.0	2.805663e+08	1.881961e+09	-2.548100e+10	-6549250.00	1436571.0	1.2134
44	2015	PropertyPlantAndEquipmentNet	4508.0	1.757120e+09	8.471831e+09	0.000000e+00	2810817.00	29944500.0	3.7736

Above description shows the regular statistic description for all nine indicators such as count, mean, std, min, 25%, 50%, 75%,max which are all widely use in statistic description and very helpful.

We can check with this description table to get more information of our indicators statistic information.

Part of Python Script for Description table:

```
df_rtba=df.loc[df['indicator_id'].isin(indicators),['company_id','indicator_id','2011','2012','2013','2014','2015']]
```

```
l_df=[]
for y in years:
    for c in indicators:
        d=list(df_rtba.loc[df_rtba['indicator_id']==c,y].dropna().describe())# describe dataset
        d.insert(0,y)
        d.insert(1,c)
        l_df.append(d)
df_ind_desc=DataFrame(l_df,columns=['Year','Indicator','count','mean','std','min','25%','50%','75%','max'])
df_ind_desc.head(45)
```

2.3 Principal component analysis

First four Principal component with python code:

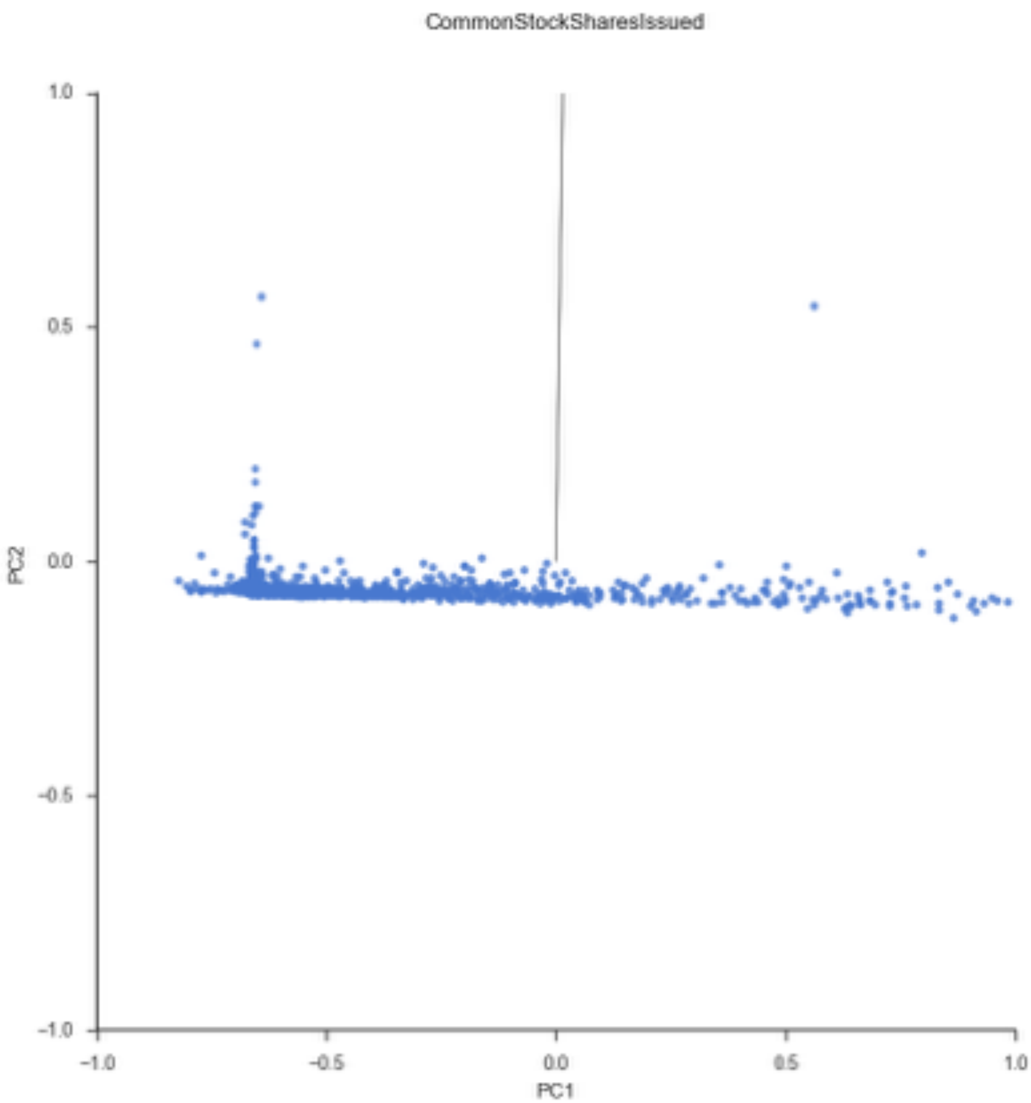
```
]: scaler = StandardScaler().fit(Values[indicators])
Values_Scaled = scaler.transform(Values[indicators])

print(Values_Scaled[:,0].mean())
print(Values_Scaled[:,0].std())
var_exp=[]
cum_var_exp=[]
pca = PCA(n_components=4)
pca.fit(Values_Scaled)
var_exp=pca.explained_variance_ratio_
cum_var_exp = np.cumsum(var_exp)
with plt.style.context('seaborn-whitegrid'):
    plt.figure(figsize=(10, 8))

    plt.bar(range(4), var_exp, alpha=0.5, align='center',
            label='individual explained variance')
    plt.step(range(4), cum_var_exp, where='mid',
             label='cumulative explained variance')
    plt.ylabel('Explained variance ratio')
    plt.xlabel('Principal components')
    plt.legend(loc='best')
    plt.tight_layout()
pca = PCA(n_components=4)
pc_scores = pd.DataFrame(pca.fit_transform(Values_Scaled))
pc_scores.columns = ['PC'+str(i+1) for i in range(len(pc_scores.columns))]
pc_scores.head()
```

```
-6.9186244962e-18
1.0
```

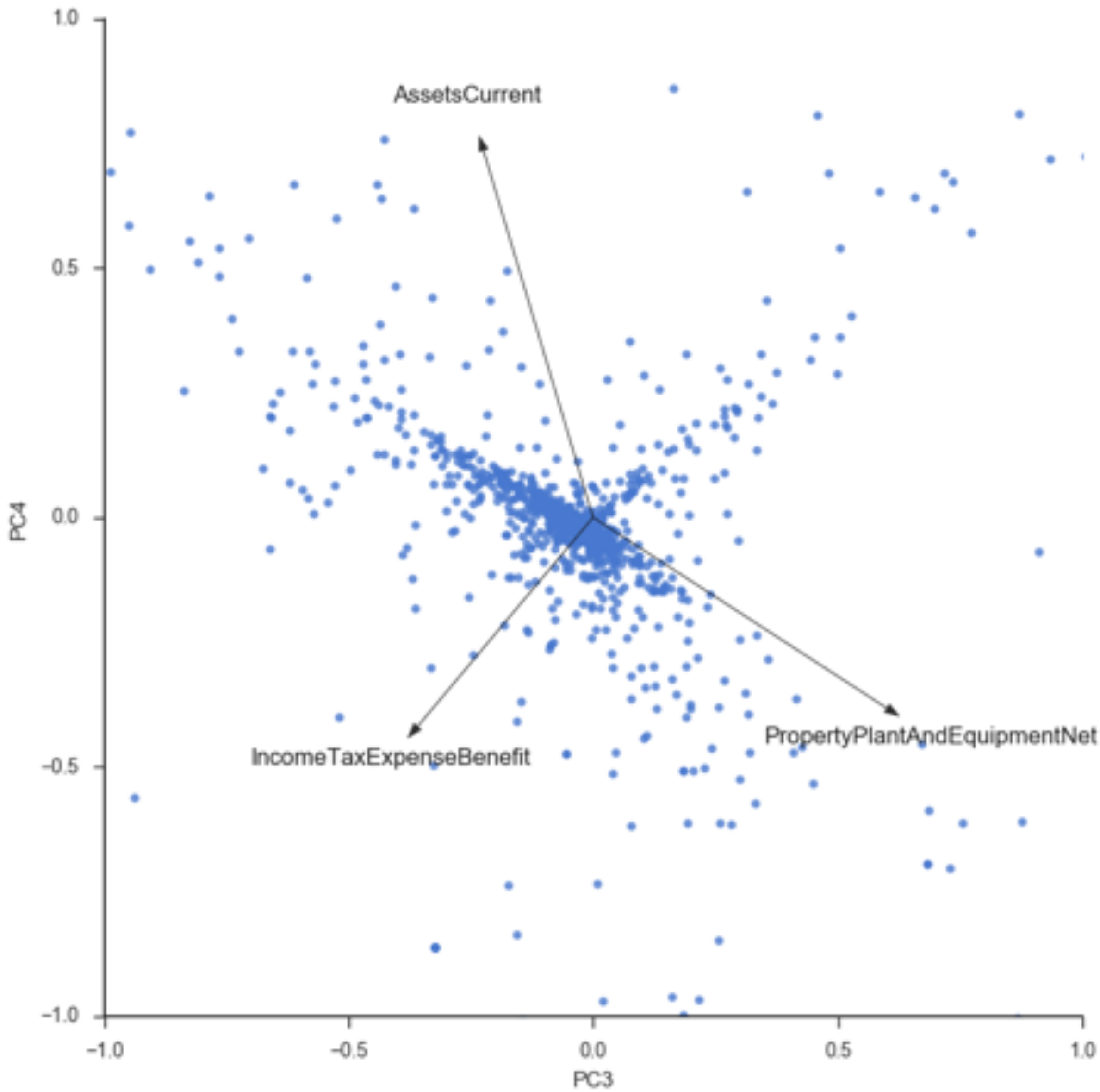
]:

	PC1	PC2	PC3	PC4
0	-0.591511	-0.068120	-0.035582	-0.006771
1	-0.641576	-0.069308	-0.001812	-0.003370
2	-0.557480	-0.072393	-0.044514	-0.000186
3	-0.652365	-0.067850	-0.002520	-0.010621
4	-0.346659	-0.069770	-0.151481	0.068830

2-D Principal component analysis:



This is the plot of first two principal of our indicators dataset, which shows the arrow of CommonStockShareIssued on graph which has no relationship with other indicators.

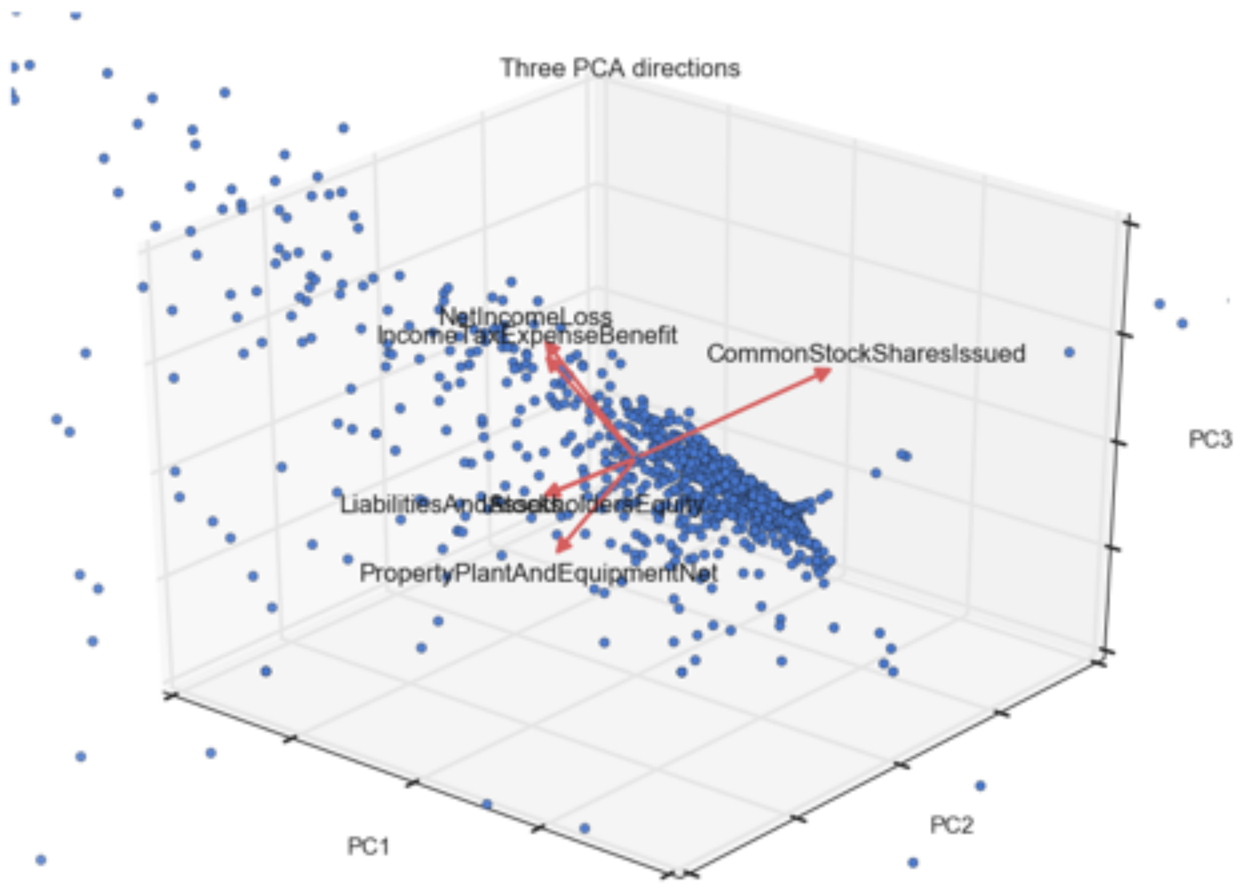


This graph shows the plot of third and fourth principal of our indicator dataset. In the 2-D graph we can see these 3 arrow represent AssetsCurrent, income TaxExpenseBenefit, and PropertyPlantAndEquipmentNet. These three indicators shows with blue or light blue color on our r-value graph which means middle close relationship between these indicators. AssetsCurrent to income TaxExpenseBenefit with r-value 0.72, to PropertyPlantAndEquipmentNet with 0.5. PropertyPlantAndEquipmentNet and TaxExpenseBenefit with r-value 0.61 which represent by different direction of each arrow on 2-D principal component graph.

2-D Principal component plot with python code:

```
def pca_biplot(x_pc=0, y_pc=1, max_arrow=0.2):
    n = pc.shape[1]
    sns.set(style="ticks", palette="muted", color_codes=True)

    g = sns.lmplot(x='PC{}'.format(x_pc + 1), y='PC{}'.format(y_pc + 1),  data=pc_scores,
                   fit_reg=False, size=8)
    for i in range(n):
        # Only plot the longer ones
        length = sqrt(pc.iloc[x_pc, i] ** 2 + pc.iloc[y_pc, i] ** 2)
        if length < max_arrow:
            continue
        plt.arrow(0, 0, pc.iloc[x_pc, i], pc.iloc[y_pc, i], color='k', alpha=0.9)
        plt.text(pc.iloc[x_pc, i] * 1.15, pc.iloc[y_pc, i] * 1.15,
                 pc.columns.tolist()[i], color='k', ha='center', va='center')
    g.set(ylim=(-1, 1))
    g.set(xlim=(-1, 1))
    plt.show()
```

3-D Principal component analysis:



3-D plot with first three principal component.

As graph show above, Assets and LiabilitiesAndStockholdersEquity shows in exactly same arrow which reflect their r-value is 1. NetIncomeLoss and IncomeTaxExpenseBenefit differ by a small angle in the same 2-D plane, and they have r-value equal to 0.92 which shows strong related to each other. The CommonStockSharesAuthorized is clearly shows in a different 3-D plane with other indicators since it has almost none relationship with other indicators.

3-D Principal component plot with python code:

```
class Arrow3D(FancyArrowPatch):
    def __init__(self, xs, ys, zs, *args, **kwargs):
         FancyArrowPatch.__init__(self, (0,0), (0,0), *args, **kwargs)
         self._verts3d = xs, ys, zs
    def draw(self, renderer):
        xs3d, ys3d, zs3d = self._verts3d
        xs, ys, zs = proj3d.proj_transform(xs3d, ys3d, zs3d, renderer.M)
        self.set_positions((xs[0],ys[0]),(xs[1],ys[1]))
        FancyArrowPatch.draw(self, renderer)
def pca_3Dplot(x_pc=0, y_pc=1, z_pc=2, max_arrow=0.2):
    fig = plt.figure(1, figsize=(8, 6))
    ax = Axes3D(fig, elev=-150, azim=50)
    sns.set(style="ticks", palette="muted", color_codes=True)

    ax.scatter(pc_scores.iloc[:, x_pc], pc_scores.iloc[:, y_pc], pc_scores.iloc[:, z_pc],
           cmap='plt.cm.Paired')
    n = pc.shape[1]
    for i in range(n):
        length = sqrt(pc.iloc[0, i] ** 2 + pc.iloc[1, i] ** 2+pc.iloc[2, i] ** 2)
        if length < max_arrow:
            continue
        a = Arrow3D([0, pc.iloc[0, i]], [0, pc.iloc[1, i]], |
                [0, pc.iloc[2, i]], mutation_scale=20,
                lw=2, arrowstyle="-|>", color="r")
        ax.add_artist(a)
        ax.text(x=pc.iloc[x_pc, i]*1.15, y=pc.iloc[y_pc, i]*1.15, z=pc.iloc[z_pc, i]*1.15,
                 s=pc.columns.tolist()[i],color='k', ha='center', va='center')
    ax.set_title("Three PCA directions")
    ax.set_xlabel('PC{}'.format(x_pc + 1))
    ax.w_xaxis.set_ticklabels([])
    ax.set_ylabel('PC{}'.format(y_pc + 1))
    ax.w_yaxis.set_ticklabels([])
    ax.set_zlabel('PC{}'.format(z_pc + 1))
    ax.w_zaxis.set_ticklabels([])
    ax.set_xlim3d(-1, 1)
    ax.set_ylim3d(-1, 1)
    ax.set_zlim3d(-1, 1)
    plt.show()
```

2.4 Analysis for target companies

Data and stock graph for Microsoft:

```
         company_id indicator_id          2010          2011          2012  \
1531221      789019       Assets  1.087040e+11  121271000000  142431000000

                 2013          2014          2015  2016
1531221  172384000000  176223000000  193694000000   NaN
         company_id                     indicator_id          2010  \
1531425      789019  LiabilitiesAndStockholdersEquity  1.087040e+11

                 2011          2012          2013          2014          2015  \
1531425  121271000000  142431000000  172384000000  176223000000  193694000000

         2016
1531425   NaN
         company_id   indicator_id          2010         2011         2012  \
1531450      789019  NetIncomeLoss  2.315000e+10  16978000000  21863000000

                2013         2014         2015  2016
1531450  22074000000  12193000000  16798000000   NaN
         company_id              indicator_id          2010        2011  \
1531396      789019  IncomeTaxExpenseBenefit  4.921000e+09  5289000000

               2012        2013        2014        2015  2016
1531396  5189000000  5746000000  6314000000  2953000000   NaN
         company_id                    indicator_id          2010         2011  \
1531278      789019  CommonStockSharesAuthorized  2.400000e+10  24000000000

                2012         2013         2014         2015  2016
1531278  24000000000  24000000000  24000000000  24000000000   NaN
         company_id   indicator_id          2010         2011          2012  \
1531222      789019  AssetsCurrent  7.491800e+10  85084000000  101466000000

                 2013          2014          2015  2016
1531222  114246000000  124712000000  139660000000   NaN
         company_id          indicator_id          2010         2011  \
1531454      789019  OperatingIncomeLoss  2.716100e+10  21763000000

                2012         2013         2014         2015  2016
1531454  26764000000  27759000000  18161000000  20182000000   NaN
         company_id                   indicator_id          2010        2011  \
1531510      789019  PropertyPlantAndEquipmentNet  8.162000e+09  8269000000

               2012         2013         2014         2015  2016
1531510  9991000000  13011000000  14731000000  18356000000   NaN
```

Data and stock graph for Google:

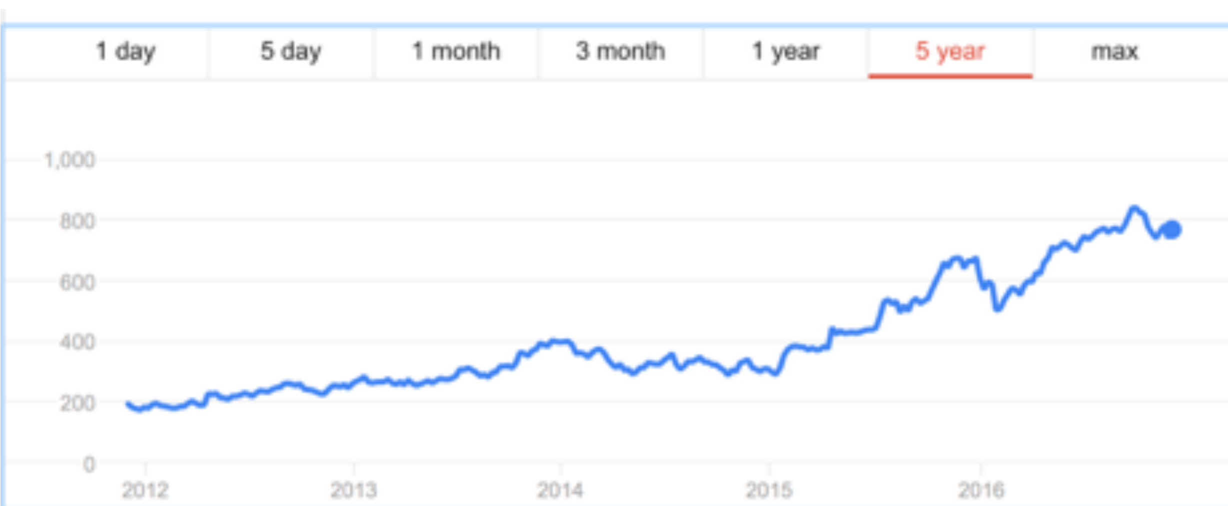
```
        company_id indicator_id  2010          2011          2012          2013  \
ick to scroll output; double click to hide  Assets   NaN  72574000000  93798000000  110920000000

                2014          2015  2016
494512  131133000000  147461000000   NaN
        company_id                        indicator_id  2010         2011  \
494729     1288776  LiabilitiesAndStockholdersEquity   NaN  72574000000

               2012          2013          2014          2015  2016
494729  93798000000  110920000000  131133000000  147461000000   NaN
        company_id   indicator_id  2010          2011          2012          2013  \
494748     1288776  NetIncomeLoss   NaN  9737000000  10737000000  12920000000

               2014          2015  2016
494748  14444000000  16348000000   NaN
        company_id                indicator_id  2010        2011        2012  \
494700     1288776  IncomeTaxExpenseBenefit   NaN  2589000000  2598000000

              2013        2014        2015  2016
494700  2282000000  3331000000  3303000000   NaN
        company_id                  indicator_id  2010        2011  \
494557     1288776  CommonStockSharesAuthorized   NaN  9000000000

               2012        2013 2014         2015  2016
494557  12000000000  12000000000  NaN  15000000000   NaN
        company_id      indicator_id  2010    2011    2012    2013 2014  \
494560     1288776  CommonStockValue   NaN  325000  330000  336000  NaN

          2015  2016
494560  687000   NaN
        company_id                indicator_id  2010        2011        2012  \
494558     1288776  CommonStockSharesIssued   NaN  324895000  329979000

             2013 2014       2015  2016
494558  335832000  NaN  687348000   NaN
        company_id   indicator_id  2010          2011          2012  \
494513     1288776  AssetsCurrent   NaN  52758000000  60454000000

               2013          2014          2015  2016
494513  72886000000  80685000000  90114000000   NaN
        company_id          indicator_id  2010          2011          2012  \
494755     1288776  OperatingIncomeLoss   NaN  11742000000  12760000000

               2013          2014          2015  2016
494755  13966000000  16496000000  19360000000   NaN
        company_id                   indicator_id  2010        2011  \
494816     1288776  PropertyPlantAndEquipmentNet   NaN  9603000000

               2012          2013          2014          2015  2016
494816  11854000000  16524000000  23883000000  29016000000   NaN
```

Data and stock graph for Facebook:

```
       company_id indicator_id  2010 2011         2012         2013  \
566659    1326801       Assets   NaN  NaN  15103000000  17895000000

              2014         2015  2016
566659 40184000000  49407000000   NaN
       company_id                  indicator_id  2010 2011         2012  \
566774    1326801  LiabilitiesAndStockholdersEquity   NaN  NaN  15103000000

              2013         2014         2015  2016
566774 17895000000  40184000000  49407000000   NaN
       company_id   indicator_id  2010 2011        2012        2013  \
566783    1326801  NetIncomeLoss   NaN  NaN    53000000  1500000000

             2014        2015  2016
566783 2940000000  3688000000   NaN
       company_id             indicator_id  2010 2011       2012        2013  \
566749    1326801  IncomeTaxExpenseBenefit   NaN  NaN  441000000  1254000000

             2014        2015  2016
566749 1970000000  2506000000   NaN
       company_id   indicator_id  2010 2011         2012         2013  \
566660    1326801  AssetsCurrent   NaN  NaN  11267000000  13070000000

              2014         2015  2016
566660 13670000000  21652000000   NaN
       company_id         indicator_id  2010 2011       2012        2013  \
566788    1326801  OperatingIncomeLoss   NaN  NaN  538000000  2804000000

             2014        2015  2016
566788 4994000000  6225000000   NaN
       company_id                 indicator_id  2010 2011        2012  \
566825    1326801  PropertyPlantAndEquipmentNet   NaN  NaN  2391000000

             2013        2014        2015  2016
566825 2882000000  3967000000  5687000000   NaN
```

Data and stock graph for Amazon:

```
      company_id indicator_id  2010         2011         2012         2013  \
55410    1018724       Assets   NaN  25278000000  32555000000  40159000000

              2014         2015  2016
55410  54505000000  65444000000   NaN
      company_id                     indicator_id  2010         2011  \
55580    1018724  LiabilitiesAndStockholdersEquity   NaN  25278000000

              2012         2013         2014         2015  2016
55580  32555000000  40159000000  54505000000  65444000000   NaN
      company_id   indicator_id  2010        2011       2012       2013  \
55603    1018724  NetIncomeLoss   NaN   631000000  -39000000  274000000

            2014       2015  2016
55603 -241000000  596000000   NaN
      company_id               indicator_id  2010       2011       2012  \
55558    1018724  IncomeTaxExpenseBenefit   NaN  291000000  428000000

            2013       2014       2015  2016
55558  161000000  167000000  950000000   NaN
      company_id                 indicator_id  2010        2011        2012  \
55454    1018724  CommonStockSharesAuthorized   NaN  5000000000  5000000000

             2013        2014        2015  2016
55454  5000000000  5000000000  5000000000   NaN
      company_id       indicator_id  2010     2011     2012     2013     2014  \
55457    1018724  CommonStockValue   NaN  5000000  5000000  5000000  5000000

         2015  2016
55457  5000000   NaN
      company_id               indicator_id  2010       2011       2012  \
55455    1018724  CommonStockSharesIssued   NaN  473000000  478000000

            2013       2014       2015  2016
55455  483000000  488000000  494000000   NaN
      company_id   indicator_id  2010         2011         2012         2013  \
55411    1018724  AssetsCurrent   NaN  17490000000  21296000000  24625000000

              2014         2015  2016
55411  31327000000  36474000000   NaN
      company_id         indicator_id  2010       2011       2012       2013  \
55605    1018724  OperatingIncomeLoss   NaN  862000000  676000000  745000000

            2014        2015  2016
55605  178000000  2233000000   NaN
      company_id                  indicator_id  2010        2011        2012  \
55658    1018724  PropertyPlantAndEquipmentNet   NaN  4417000000  7060000000

              2013         2014         2015  2016
55658  10949000000  16967000000  21838000000   NaN
```



Analysis:

All these software tech companies has very high value in Assets, PropertyPlantAndEquipmentNet, NetIncomeLoss and all other indicators represent the successful of a company. Compare to the description of our dataset these target companies are way above 75%. Amazon has a huge positive NetIncomeLoss at 2015 show in our dataset, and their stock price raise 200% during 2015 to 2016 show in Stock diagram. All of four companies have raised their stock value between 200% to 300%+, and our dataset indicators also shows their huge gain in Assets, NetIncomeLoss, and PropertyPlantAndEquipmentNet. These are three most important indicators related to the stock value in our dataset. The stock value graph I simply get from google search which shows a dynamic change for these target company in detail.

Part of Python Script to get target companies data indicators:

```
import pandas as pd
import numpy as np
df = pd.read_csv('indicators_by_company.csv',
dtype={'2011': str, '2012': str, '2013': str, '2014': str, '2015': str})
```

```
df = df[df['company_id']==1018724]
#df = df[df['year']=='2011']
#df["indicator_id"].unique()
print(df[df["indicator_id"]=='Assets'])
print(df[df["indicator_id"]=='LiabilitiesAndStockholdersEquity'])
print(df[df["indicator_id"]=='NetIncomeLoss'])
print(df[df["indicator_id"]=='IncomeTaxExpenseBenefit'])
print(df[df["indicator_id"]=='CommonStockSharesAuthorized'])
print(df[df["indicator_id"]=='CommonStockValue'])
print(df[df["indicator_id"]=='CommonStockSharesIssued'])
print(df[df["indicator_id"]=='AssetsCurrent'])
print(df[df["indicator_id"]=='OperatingIncomeLoss'])
print(df[df["indicator_id"]=='PropertyPlantAndEquipmentNet'])
```

End Of Analysis

Conclusion

3.1 Summary

This dataset is very interesting and contains with large amount of information about American companies. There are huge amount of indicators I can work with, but I do chose the ten most common and important indicators to analysis. The project help me to understand the relationship between each indicators and how they related to the real stock value change. The four target I chose are all software tech companies, and I may work with them in my future. I do get the partial reason that why these companies are so successful from this dataset. The flag tech companies are always the outlier in these 12,129 US companies. They do have a huge NetIncomeLoss and Assets.

3.2 FutureAnalysis

In this project, I focus on analysis with these indicators in US stocks fundamental data with yearly dataset. However I would like to try to find some weekly or daily dataset with these target tech companies to do more deep analysis to get know in more detail why they are so successfully and how their stock value raise that much in these five years. It will be great if we can try to predicate the stock value from data analysis!

THE END