assignment 2

February 27, 2018

0.1 Import library and data

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
import cufflinks as cf
cf.go_offline()
init_notebook_mode(connected=True)
%matplotlib inline
```

In [2]:
```
df_ALL06 = pd.read_csv('all_stocks_2006-01-01_to_2018-01-01.csv')
df_AMZN = pd.read_csv('AMZN_2006-01-01_to_2018-01-01.csv')
df_GOOGL = pd.read_csv('GOOGL_2006-01-01_to_2018-01-01.csv')
```

0.2 Explore data

In [3]:
```
df_ALL06.columns
```

Out[3]:
```
Index(['Date', 'Open', 'High', 'Low', 'Close', 'Volume', 'Name'], dtype='object')
```

In [4]:
```
df_ALL06.head()
```

Out[4]:

	Date	Open	High	Low	Close	Volume	Name
0	2006-01-03	77.76	79.35	77.24	79.11	3117200	MMM
1	2006-01-04	79.49	79.49	78.25	78.71	2558000	MMM
2	2006-01-05	78.41	78.65	77.56	77.99	2529500	MMM
3	2006-01-06	78.64	78.90	77.64	78.63	2479500	MMM
4	2006-01-09	78.50	79.83	78.46	79.02	1845600	MMM

In [5]:
```
df_ALL06.describe()
```

Out[5]:

	Open	High	Low	Close	Volume
count	93587.000000	93602.000000	93592.000000	93612.000000	9.361200e+04
mean	85.623260	86.387045	84.836664	85.641753	2.015667e+07
std	108.151723	108.956365	107.225361	108.121106	3.442108e+07
min	6.750000	7.170000	0.000000	6.660000	0.000000e+00
25%	33.950000	34.290000	33.600000	33.960000	5.040180e+06
50%	60.040000	60.630000	59.490000	60.050000	9.701142e+06
75%	94.000000	94.740000	93.250000	94.012500	2.075222e+07
max	1204.880000	1213.410000	1191.150000	1195.830000	8.432640e+08

```
In [6]: df_ALL06.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 93612 entries, 0 to 93611
Data columns (total 7 columns):
Date      93612 non-null object
Open      93587 non-null float64
High      93602 non-null float64
Low       93592 non-null float64
Close     93612 non-null float64
Volume    93612 non-null int64
Name      93612 non-null object
dtypes: float64(4), int64(1), object(2)
memory usage: 5.0+ MB
```

First I am interesting in the company which has the max difference in one day. I want to know if there are some relationship.

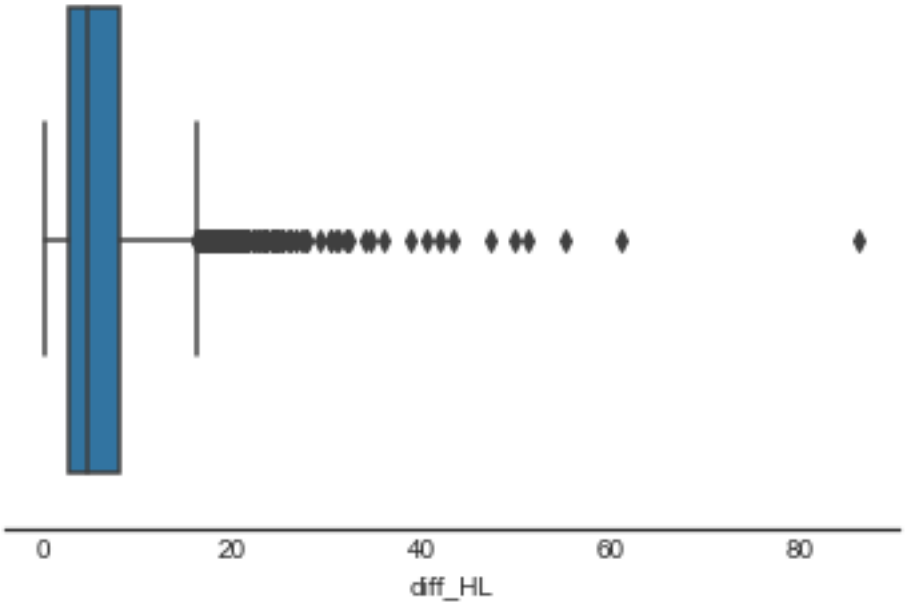
```
In [7]: d = df_ALL06.copy()
        d['Diff_HL'] = d['High']-d['Low']
        d[d['Diff_HL']==max(d['Diff_HL'])]
```

Out[7]:

	Date	Open	High	Low	Close	Volume	Name	Diff_HL
90451	2017-06-09	1012.5	1012.99	927.0	978.31	7647692	AMZN	85.99

Is 85.99 a big number? Now I am focusing on amazon data.

```
In [8]: d = df_AMZN.copy()
        d['Date']= pd.to_datetime(d['Date'])
        d['diff_HL'] = d['High']-d['Low']
        d = d.set_index('Date')
        #d['diff_HL'].plot.box(grid = True)
        sns.set_style("white")
        #sns.swarmplot(d.diff_HL)
        sns.boxplot(d.diff_HL)
        sns.despine(left = True);
```



Looks like amazon has a very special day in Jun. 9th, 2017. But, what happened?

So I googled it, and found a news: Tech stocks took a hit after a Goldman Sachs analyst questioned this year's run-up in the industry's five biggest names – Apple, Microsoft, Amazon, Facebook and Alphabet – the parent company of Google.

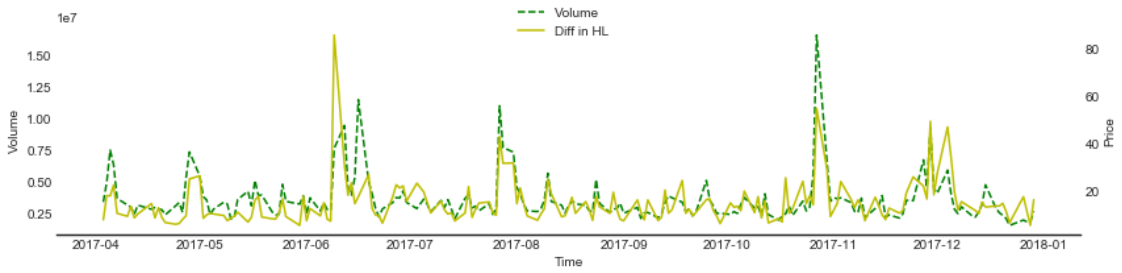
Let me see what happened during this time

```
In [9]: temp = d[d.index >pd.to_datetime('2017-04-01')]
        temp[['Open','Close','High','Low']].iplot(kind='spread',xTitle='Dates',yTitle='price',t
```

So, the difference of 'High' and 'Low' might have little influence on 'Open' and 'Close' prize.

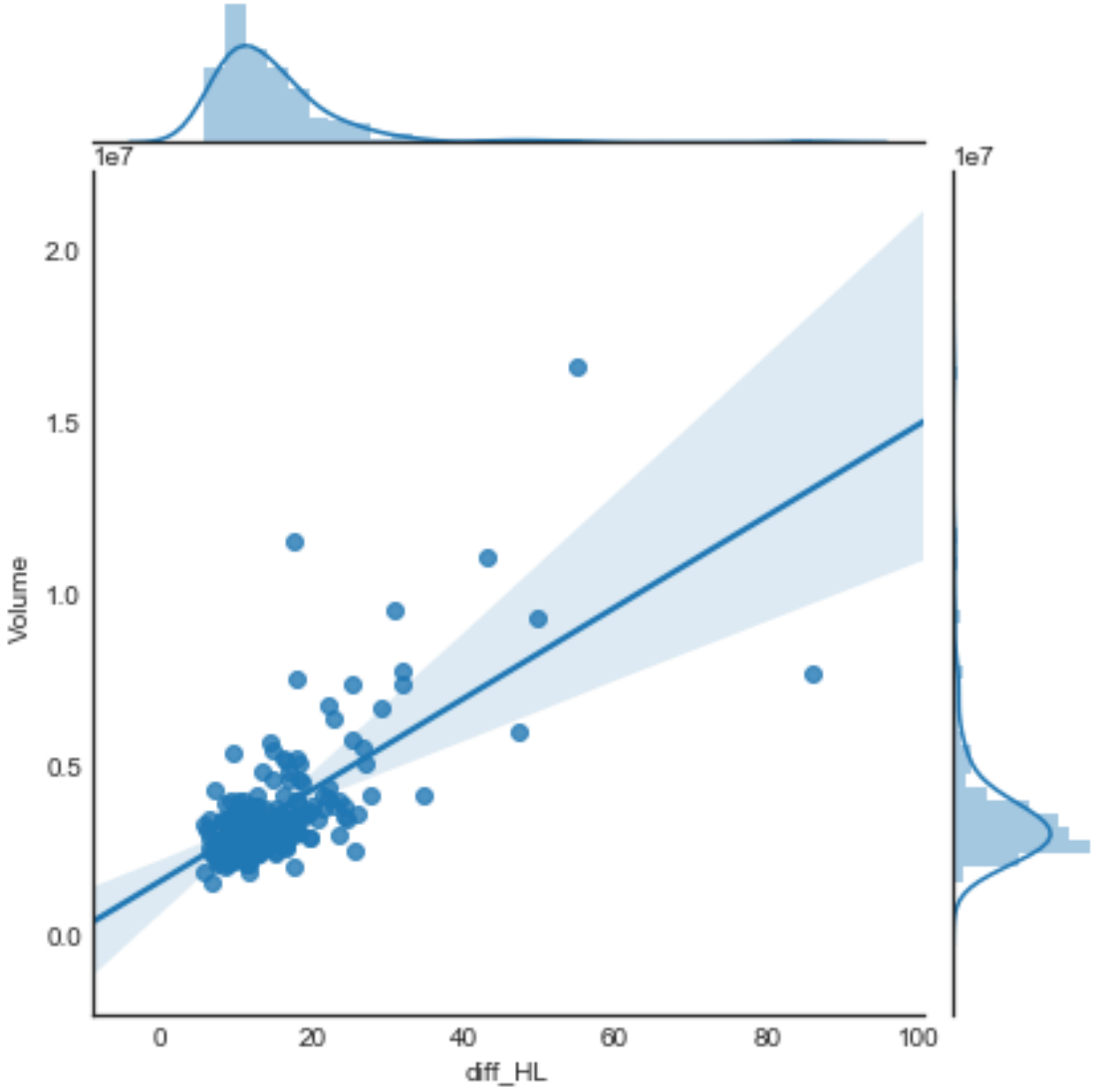
How about 'volume'?

```
In [10]: fig, ax = plt.subplots(figsize = (12,3))
         ax.plot(temp.Volume,'g--',label='Volume')
         ax2 =ax.twinx()
         ax2.plot(temp.diff_HL,'y',label='Diff in HL')
         sns.despine(ax=ax, right=True, left=True)
         sns.despine(ax=ax2, left=True, right=False)
         ax2.spines['right'].set_color('white')
         ax.set_xlabel('Time')
         ax.set_ylabel('Volume')
         ax2.set_ylabel('Price')
         fig.legend(loc =9)
         fig.tight_layout()
```



So, there might be some relationship.

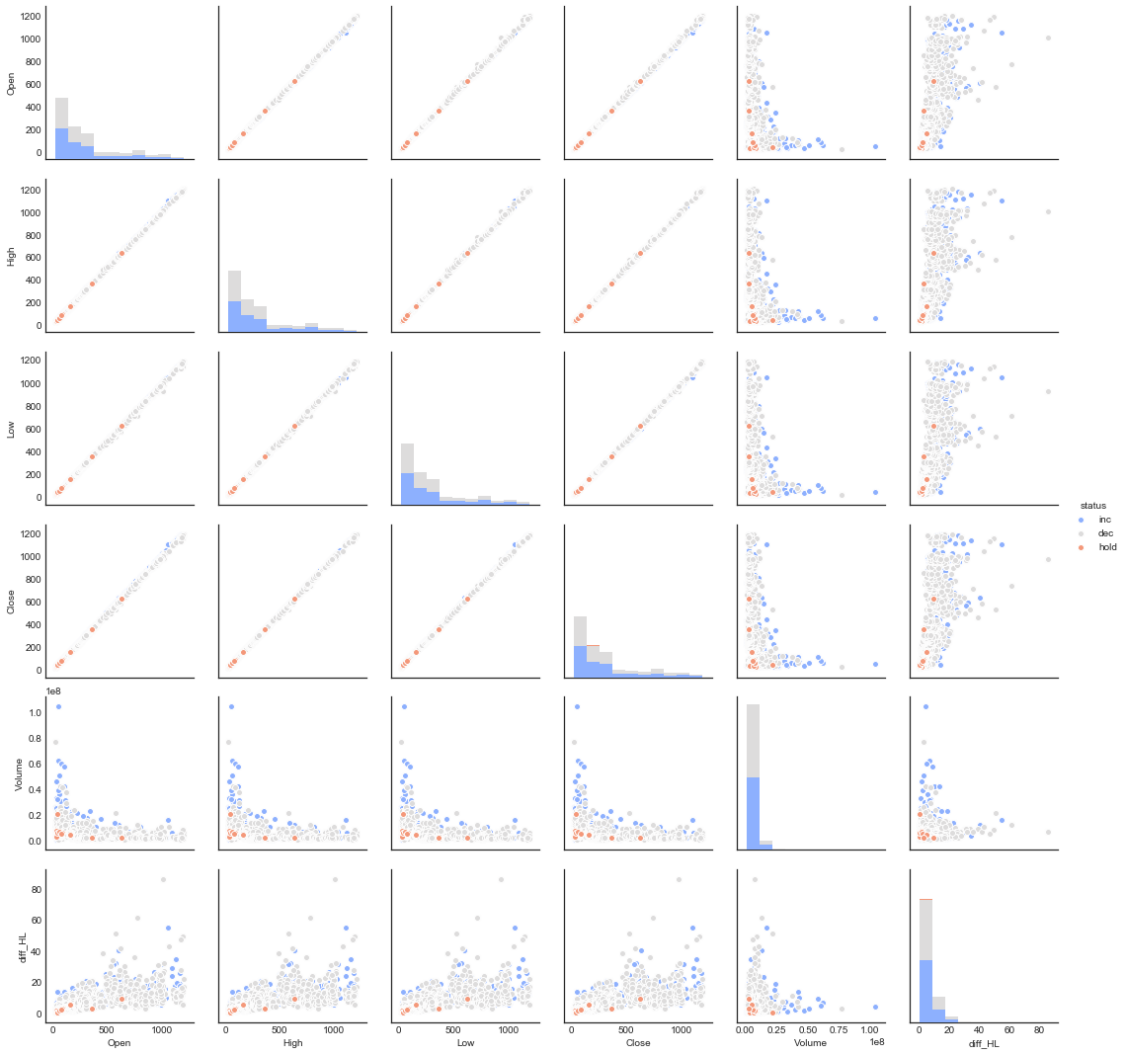
```
In [11]: #sns.lmplot(x='diff_HL',y='Volume',data=temp,palette='coolwarm', aspect=0.6,size=8)
         g = sns.JointGrid(x='diff_HL',y='Volume',data=temp)
         g = g.plot(sns.regplot, sns.distplot)
```



There might be some relationships. Normally people sell stock when the price change a lot. But the sample with high 'diff_HL' is too small. So I am not sure there is a linear relationship. I'll set a dummy variable 'status' for the whole amazon data. Details about the dummy variable is in next section.

```
In [12]: def set_status(s):
             o = s['Open']
             c = s['Close']
             if o>c:
                 return 'dec'
             elif o<c:
                 return 'inc'
             else:
                 return 'hold'
         d.is_copy=False
         d['status'] = d.apply(set_status,axis = 1)
         sns.pairplot(d,hue = 'status',palette='coolwarm')

Out[12]: <seaborn.axisgrid.PairGrid at 0x1f38e1c4940>
```



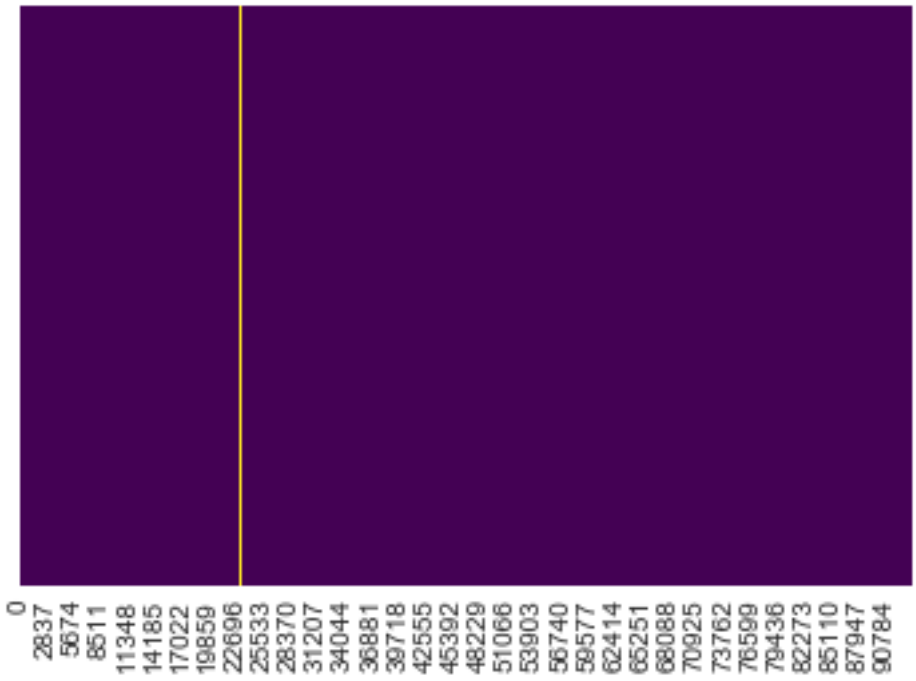
We can see there are a lot of samples when the price is low.

Create some dummies Let me create somd data. If the 'Open' is bigger than the 'Close', I will set a dummy variable 'status' to 'dec'. And if the 'Open' is small, 'status' will be 'inc'.

I should check whether is this two status enough

```
In [13]: d = df_ALL06.copy()
         d['Date']=pd.to_datetime(d['Date'])
         temp = d.set_index('Date')
         sns.heatmap([temp['Open'] == temp['Close']],yticklabels=False,cbar=False,cmap='viridis
```

```
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x1f38338fa20>
```



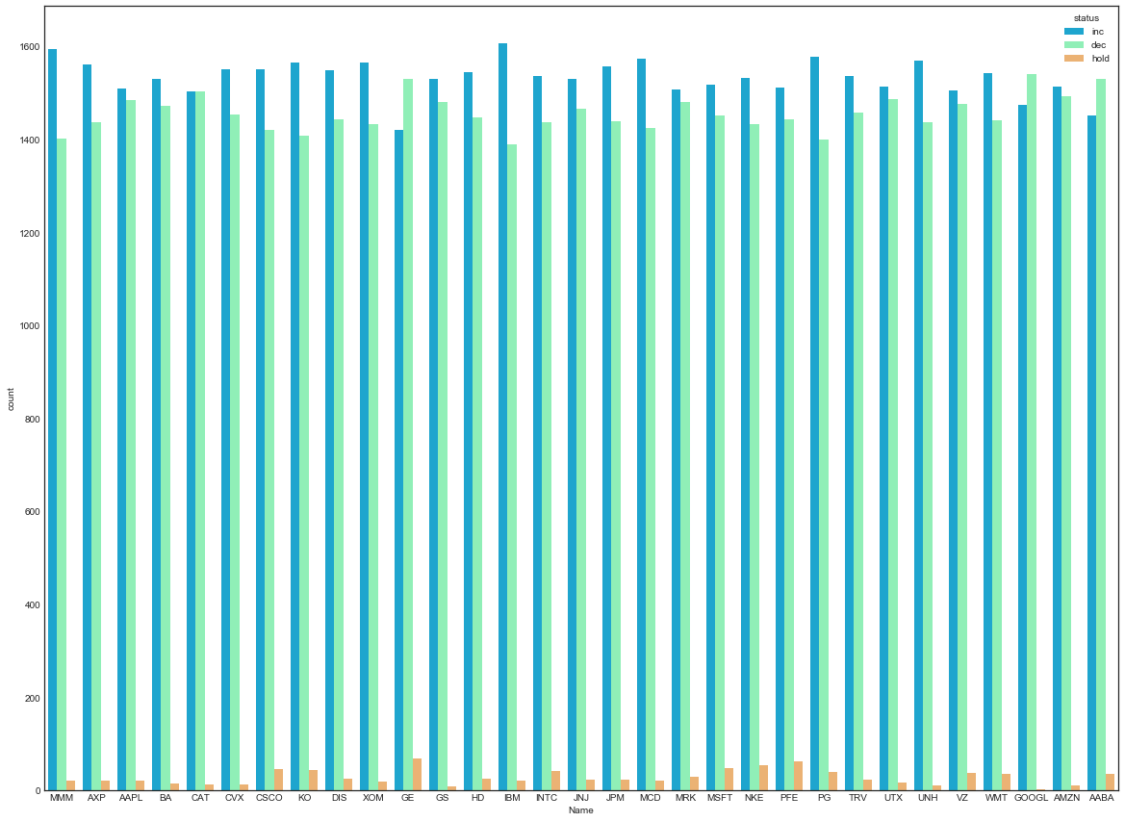
Yes, there are sometimes the 'Open' equal to the 'Close'. I will set 'status' to 'hold' for this situation.

```
In [14]: temp['status']=temp.apply(set_status,axis = 1)
```

Count the status

```
In [15]: plt.figure(figsize=(20,15))
         sns.countplot(x="Name", data=temp,hue = 'status',palette='rainbow')

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x1f3929a0518>
```

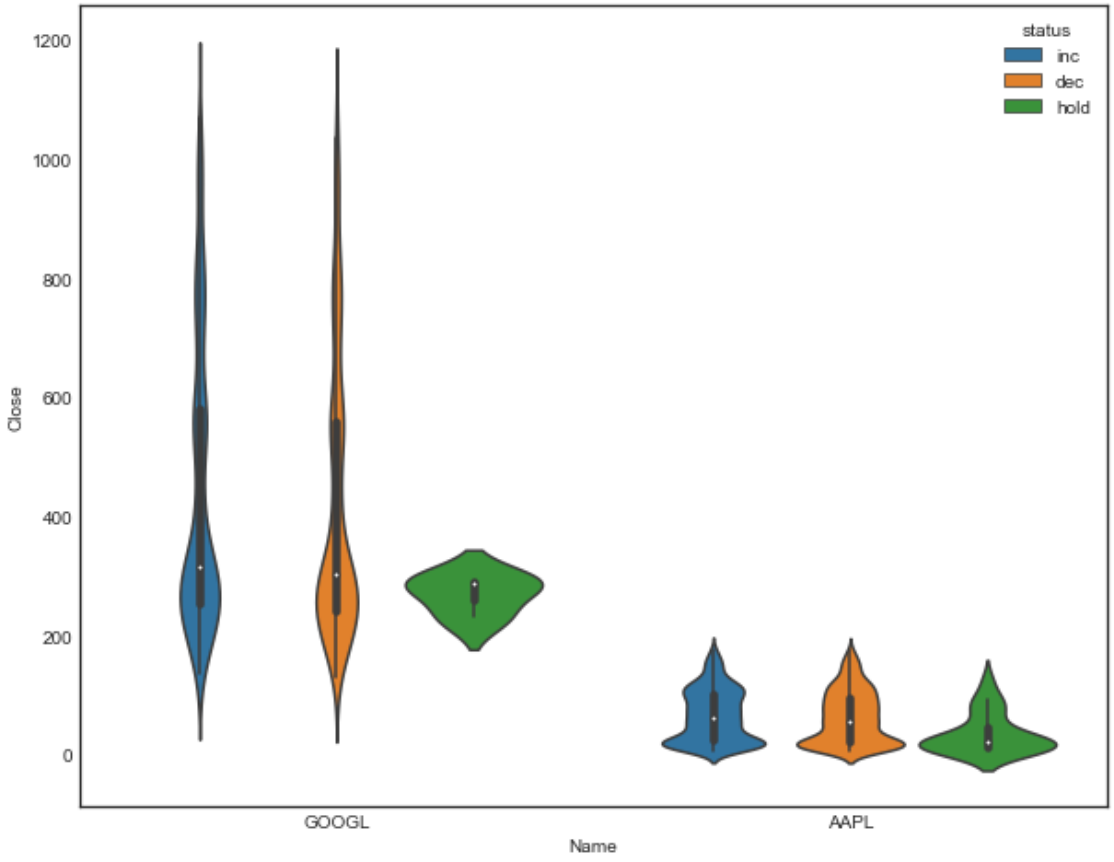


It seems like 'GOOGL' has the lowest of 'hold'.
Show data about the 'Close' price of 'Google', and 'Apple'

```
In [16]: temp = pd.concat([temp[temp['Name']=='GOOGL'],temp[temp['Name']=='AAPL']])
```

```
In [17]: plt.figure(figsize=(10,8))
         sns.violinplot(x="Name", y="Close",hue = 'status', data=temp)
```

```
Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x1f38e90a208>
```

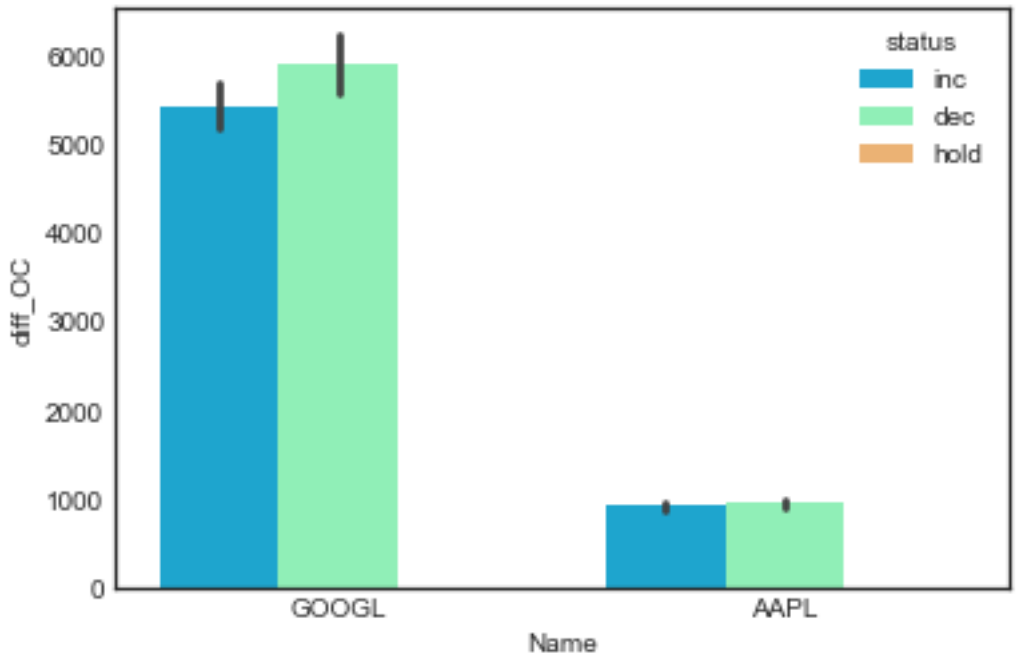


So, google keeps changing from 'inc' and 'dec', but not 'hold', when the stock close price of google is more than $400.

I want to plot a picture which show the sum of the difference per day in 'Open' price and 'Close' price, seperate by 'status'. I think the amount of 'inc' in google is much bigger than amount of 'dec'.

```
In [18]: import numpy as np
         temp['diff_OC'] = abs(temp['Open']-temp['Close'])
         sns.barplot(x='Name',y = 'diff_OC',hue='status',data=temp,palette='rainbow',estimator
```

```
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x1f393b40940>
```



According to this pic, I was wrong. The only reason for this is that the price for 'Open' and 'Close' are not consistent. The 'Close' price for yesterday is not equal to the 'Open' price for today.

0.2.1 Diff in data

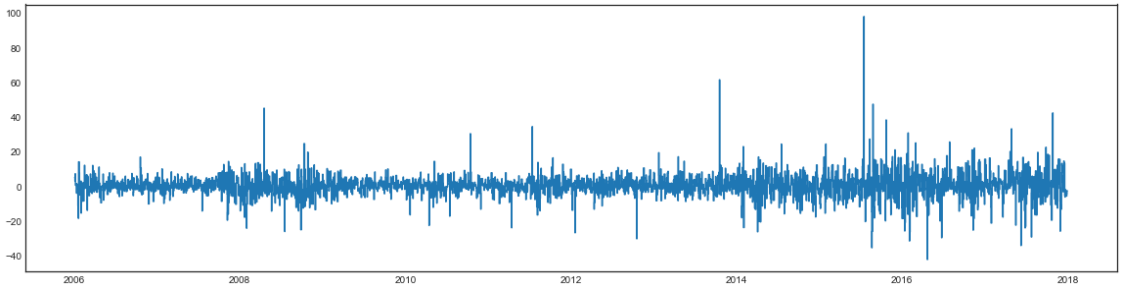
Now I am using google data. And I want to know how it changes.

```
In [19]: d = df_GOOGL.copy()
         d['Date']= pd.to_datetime(d['Date'])
         d.set_index('Date',inplace = True)
         d.drop('Name',axis = 1,inplace = True)
         temp = d['Close'].diff()
```

Draw the difference of 'Close'.

```
In [20]: fig, axes = plt.subplots(figsize = (20,5))
         axes.step(temp.index,temp)
```

```
Out[20]: [<matplotlib.lines.Line2D at 0x1f393b93be0>]
```



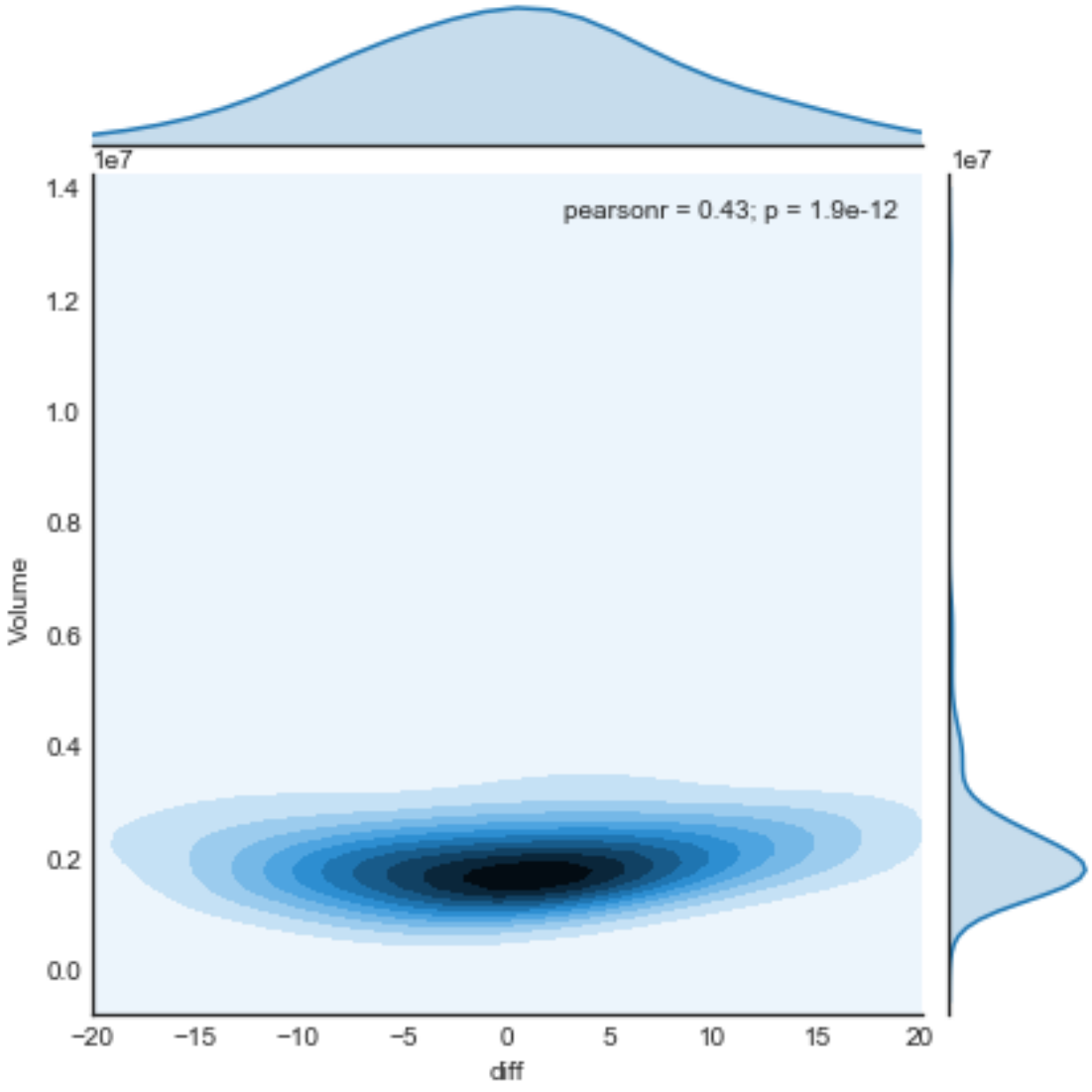
Changes are dramatic. Let me fous on the year of 2015.

```
In [21]: temp = d[(d.index <pd.to_datetime('2016-01-01')) & (d.index > pd.to_datetime('2015-01-
         temp['diff']=temp['Close'].diff()
```

I think when the diff is high (both negative and positive), the volume is high as well. And when diff is low (close to 0), the volume is low.

```
In [22]: sns.jointplot(x='diff',y='Volume',data=temp,kind='kde',xlim = [-20,20])
```

```
Out[22]: <seaborn.axisgrid.JointGrid at 0x1f3932c6e48>
```



I just focus on diff is between -20 to 20. And the low point is like to be when the diff is 0, which means that the lower 'Volume' are around the 'diff' = 0 area just like what I thought. This is totally reasonable.