

```

In [3]: import pandas as pd
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plt
import datetime
from dateutil.relativedelta import relativedelta
from datetime import date

d_parser = lambda x: pd.datetime.strptime(x, '%Y.%m.%d %H:%M:%S')
df = pd.read_excel(r'C:\Users\omarf\OneDrive\Desktop\TRADING.xlsx', parse_dates=['Open Time'], date_parser=d_parser)
#deals = pd.read_excel(r'C:\Users\omarf\OneDrive\Desktop\deals.xlsx', parse_dates=['Close Time'], date_parser=d_parser)

#EXPOSURE
df['Volume2'] = np.where(df['Type'] == 'sell', df['Volume'] *-1, df['Volume'])
df['Exposure'] = df.groupby(['Symbol'])['Volume2'].cumsum()

#DURATION
df['Open Time'] = pd.to_datetime(df['Open Time']).astype('datetime64[ns]')
df['Close Time'] = pd.to_datetime(df['Close Time']).astype('datetime64[ns]')
df['Duration'] = df['Close Time'] - df['Open Time']

df.set_index('Open Time')

#CLOSED PROFIT

df.head()

```

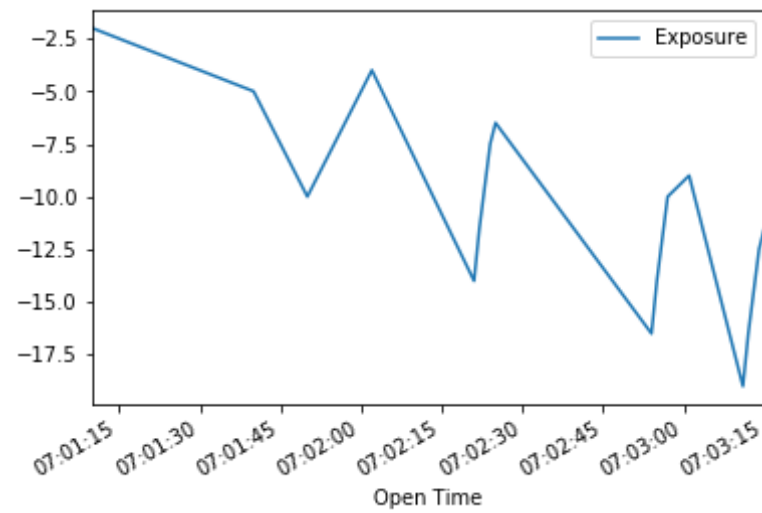
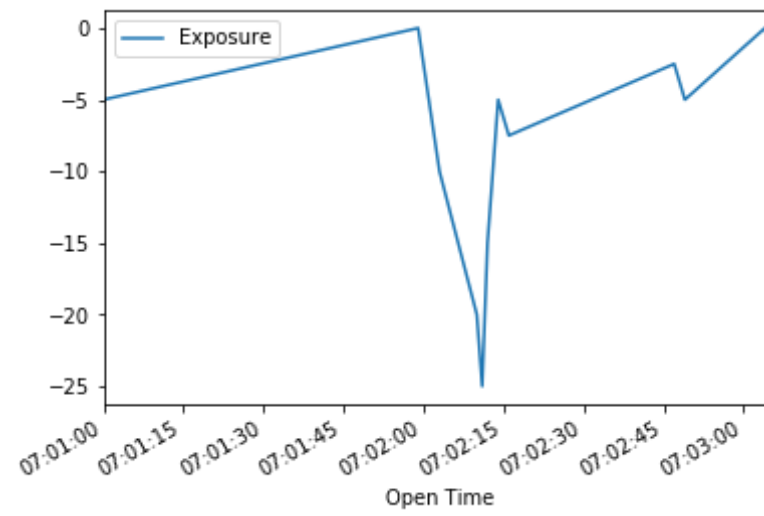
Out[3]:

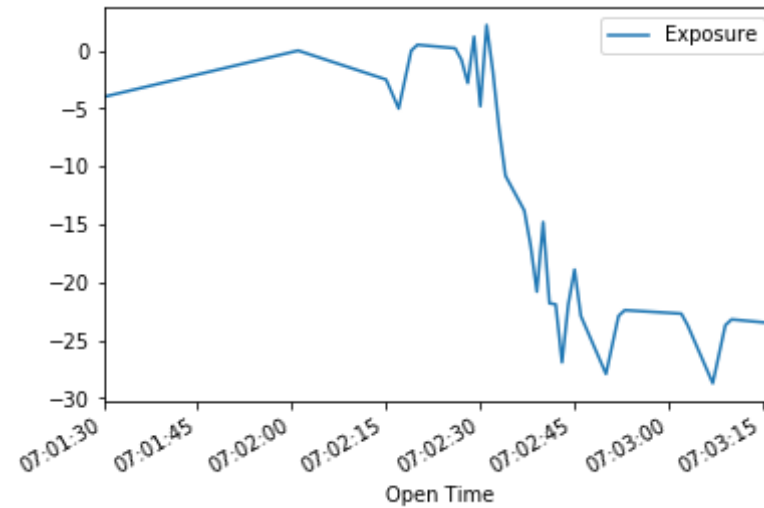
Open	Ticket	Client	Symbol	Type	Volume	Open Price	Close	Close	Profit
------	--------	--------	--------	------	--------	------------	-------	-------	--------

	Time	Number	Client	Symbol	Type	Volume	Open Price	Time	Price	Profit
	Open Time	Ticket Number	Client	Symbol	Type	Volume	Open Price	Close Time	Close Price	Profit
0	2020-07-21 07:01:00	11112	706111	EURUSD	sell	5.0	1.14377	2020-07-21 09:14:07	1.14286	-300
1	2020-07-21 07:01:10	11113	706112	GBPUSD	sell	2.0	1.26944	2020-07-21 09:19:26	1.26898	100
2	2020-07-21 07:01:20	11114	707111	GBPUSD	sell	1.0	1.26992	2020-07-21 09:19:18	1.26898	100
3	2020-07-21 07:01:30	11115	460600	XAUUSD	sell	4.0	1838.85000	2020-07-21 09:19:33	1837.35000	-300
4	2020-07-21 07:01:40	11116	707111	GBPUSD	sell	2.0	1.27056	2020-07-21 09:19:30	1.26898	-250

```
In [4]: EURUSD = df[df['Symbol'] == 'EURUSD'].copy().reset_index()
GBPUSD = df[df['Symbol'] == 'GBPUSD'].copy().reset_index()
XAUUSD = df[df['Symbol'] == 'XAUUSD'].copy().reset_index()
EURUSD.plot(x='Open Time', y='Exposure')
GBPUSD.plot(x='Open Time', y='Exposure')
XAUUSD.plot(x='Open Time', y='Exposure')
```

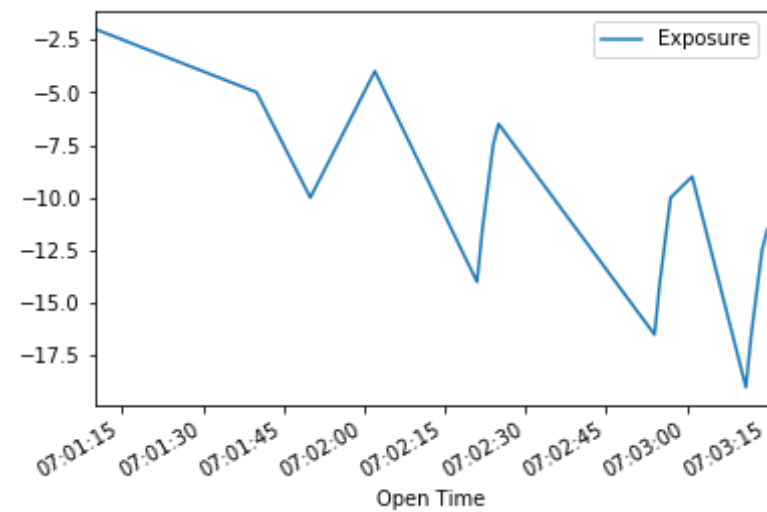
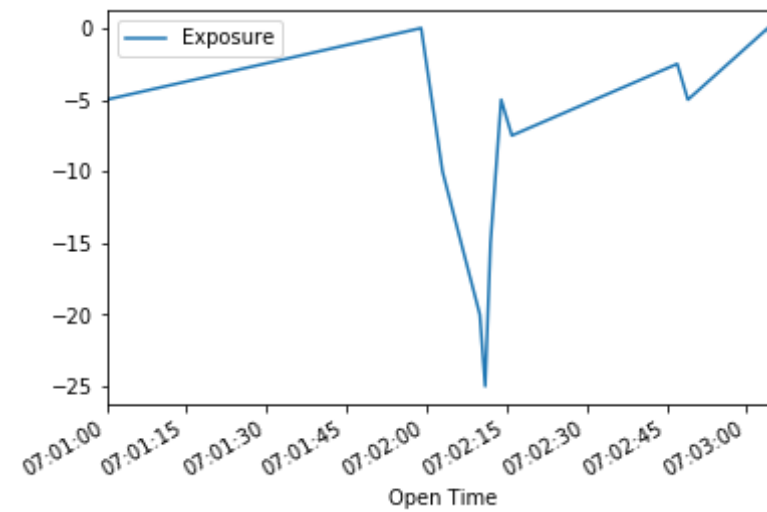
```
Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1eeb4dcadc8>
```

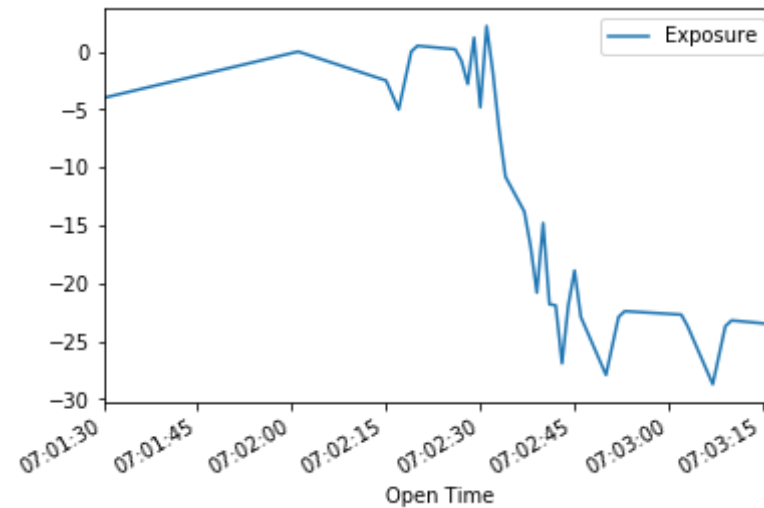




```
In [5]: df.groupby('Symbol').plot(x='Open Time', y='Exposure')
```

```
Out[5]: Symbol
EURUSD    AxesSubplot(0.125,0.2;0.775x0.68)
GBPUSD    AxesSubplot(0.125,0.2;0.775x0.68)
XAUUSD    AxesSubplot(0.125,0.2;0.775x0.68)
dtype: object
```





```
In [13]: df.groupby('Symbol')[].plot()
```

```
-----
----
TypeError                                Traceback (most recent call l
ast)
<ipython-input-13-d833f4e2da29> in <module>
----> 1 df.groupby['Symbol']('Exposure').plot()

TypeError: 'method' object is not subscriptable
```

```
In [6]: profit_client = df.groupby(['Client']).sum()["Profit"].reset_index().so
rt_values(['Profit'], ascending=[False])
profit_client
```

Out[6]:

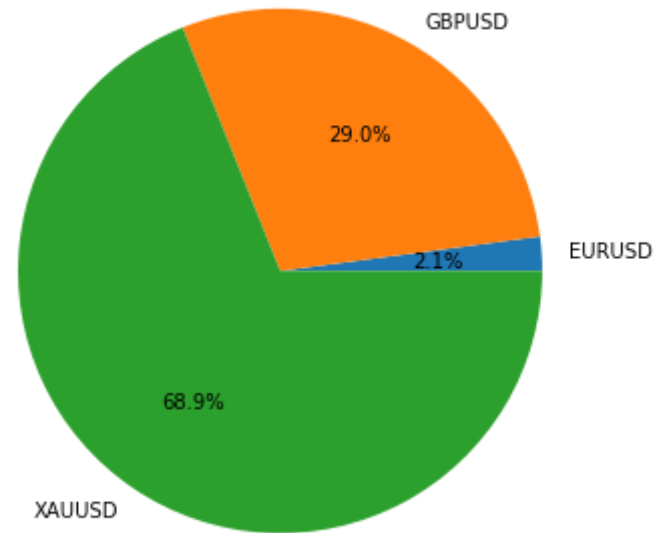
	Client	Profit
4	707111	1401
0	460600	-781
1	540303	-800
2	706111	-945
3	706112	-2915

```
In [7]: profit_symbol = df.groupby(['Symbol']).sum()["Profit"].reset_index().sort_values(['Profit'], ascending=False)
profit_symbol
```

Out[7]:

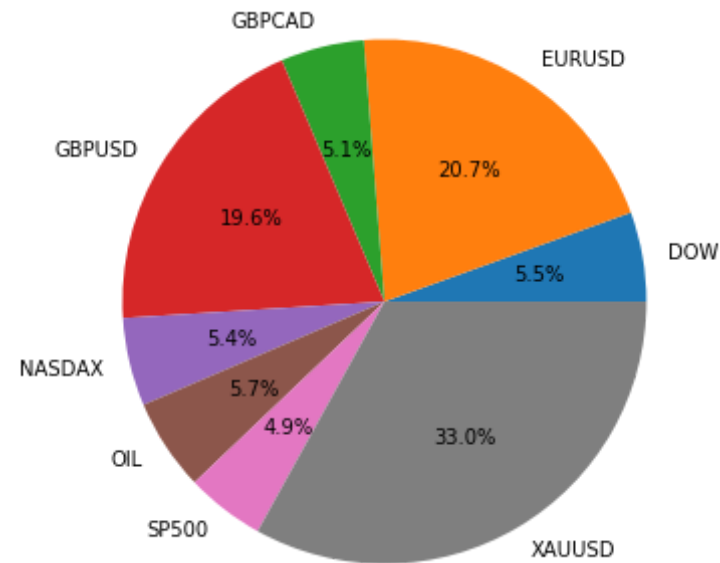
	Symbol	Profit
1	GBPUSD	-135
2	XAUUSD	-835
0	EURUSD	-3070

```
In [10]: profittrade = df[df.Profit > 0]
profittrade = profittrade.groupby(['Symbol']).sum()["Profit"].reset_index()
my_labels = profittrade['Symbol']
plt.pie(profittrade['Profit'], labels=my_labels, autopct='%1.1f%%', radius=1.5)
plt.show()
```



```
In [260]: total_volume = df.groupby(['Symbol']).sum()["Volume"].reset_index()

my_labels = total_volume['Symbol']
plt.pie(total_volume['Volume'], labels=my_labels, autopct='%1.1f%%', radius=1.5)
plt.savefig('Lots.png')
```

In []:

```
In [261]: profit_trade= df.sort_values(by='Profit', ascending=False).head()
profit_trade[['Ticket Number', 'Client', 'Type', 'Symbol', 'Volume', 'Profit']]
```

Out[261]:

	Ticket Number	Client	Type	Symbol	Volume	Profit
32	11144	707111	sell	XAUUSD	4.0	767
44	11156	460600	buy	XAUUSD	5.0	654
107	11219	303221	sell	NASDAX	2.0	600
20	11132	707111	buy	XAUUSD	0.5	533
53	11165	707111	buy	XAUUSD	0.5	501

```
In [262]: scalping_trade= df.sort_values(by='Duration', ascending=True).head()
scalping_trade[['Ticket Number', 'Client', 'Type', 'Symbol', 'Volume', 'Pr
```

```
ofit', 'Open Time', 'Close Time', 'Duration']].head()
```

Out[262]:

	Ticket Number	Client	Type	Symbol	Volume	Profit	Open Time	Close Time	Duration
47	11159	706112	buy	EURUSD	5.0	-340	2020-07-21 10:18:15	2020-07-21 10:18:18	00:00:03
61	11173	706112	buy	EURUSD	5.0	-340	2020-07-21 10:18:15	2020-07-21 10:18:18	00:00:03
14	11126	706112	buy	EURUSD	5.0	-340	2020-07-21 10:18:15	2020-07-21 10:18:18	00:00:03
75	11187	706112	buy	GBPCAD	5.0	-340	2020-07-21 10:18:15	2020-07-21 10:18:18	00:00:03
44	11156	460600	buy	XAUUSD	5.0	654	2020-07-29 15:04:58	2020-07-29 15:05:02	00:00:04

In [267]:

```
symbol = df[(df['Symbol'] == 'GBPUSD')].copy().reset_index()
symbol2 = df[(df['Symbol'] == 'EURUSD')].copy().reset_index()
symbol3 = df[(df['Symbol'] == 'GBPCAD')].copy().reset_index()
symbol['Total Profit'] = symbol.groupby(['Symbol'])['Profit'].cumsum()
symbol.head()
```

Out[267]:

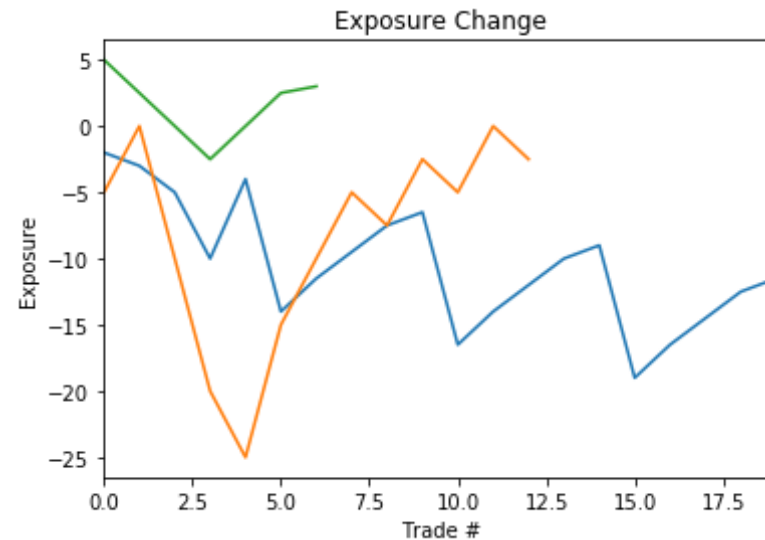
	index	Open Time	Ticket Number	Client	Symbol	Type	Volume	Open Price	Close Time	Close Price	Profit
0	1	2020-07-21 07:05:07	11113	706112	GBPUSD	sell	2.0	1.26944	2020-07-21 09:19:26	1.26898	100
1	2	2020-07-21 07:17:26	11114	707111	GBPUSD	sell	1.0	1.26992	2020-07-21 09:19:18	1.26898	100
2	4	2020-07-21 09:07:11	11116	707111	GBPUSD	sell	2.0	1.27056	2020-07-21 09:19:30	1.26898	-250
3	5	2020-07-21 09:11:09	11117	706111	GBPUSD	sell	5.0	1.26957	2020-07-21 09:19:22	1.26898	50

	index	Open Time	Ticket Number	Client	Symbol	Type	Volume	Open Price	Close Time	Close Price	Profit
4	8	2020-07-21 09:19:14	11120	706111	GBPUSD	buy	6.0	1.26898	2020-07-21 09:19:30	1.26978	-300

```
In [268]: symbol['Total Profit'] = symbol['Profit'].cumsum()
symbol['Total Profit'].plot()
plt.title("Profit Change")
plt.xlabel("Trade #")
plt.ylabel("Profit")
plt.savefig('Profit.png')
```



```
In [276]: symbol['Exposure'].plot()
symbol2['Exposure'].plot()
symbol3['Exposure'].plot()
plt.title("Exposure Change")
plt.xlabel("Trade #")
plt.ylabel("Exposure")
plt.savefig('Exposure.png')
```



```
In [270]: #EXPORT TO EXCEL

#FILE EXPORTER
writer = pd.ExcelWriter (r'C:\Users\omarf\OneDrive\Desktop\zzz9.xlsx',
engine='xlsxwriter')

#INSERT STATS TO EXCEL
profit_client.to_excel(writer, sheet_name = 'Trade Monitoring', startcol=0, startrow=0, index=False)
profit_symbol.to_excel(writer, sheet_name = 'Trade Monitoring', startcol=0, startrow=10, index=False)
total_volume.to_excel(writer, sheet_name = 'Trade Monitoring', startcol=0, startrow=40, index=False)
profit_trade.to_excel(writer, sheet_name = 'Trade Monitoring', startcol=0, startrow=60, index=False)
scalping_trade.to_excel(writer, sheet_name = 'Trade Monitoring', startcol=0, startrow=80, index=False)

writer.sheets['Trade Monitoring'].insert_image('K1', 'Lots.png')
writer.sheets['Trade Monitoring'].insert_image('K1', 'Profit.png')
```

```
plt.savefig('Lots.png')
```

```
writer.save()
```

```
<Figure size 432x288 with 0 Axes>
```

In []:

In []:

In []: