In [192]:

```
# TODO: Read data.
import pandas as pd
TWITTER = pd.read_csv('/Users/khaladdin/Desktop/Twitter Project/TWITTER.csv')
```

In [193]:

```
# TODO: Create different datasets for each stock.
ADI = TWITTER.loc[TWITTER['#RIC'] == 'ADI.OQ']
FID = TWITTER.loc[TWITTER['#RIC'] == 'FIS.N']
FIS = TWITTER.loc[TWITTER['#RIC'] == 'FISV.OQ']
GPN = TWITTER.loc[TWITTER['#RIC'] == 'GPN.N']
JUN = TWITTER.loc[TWITTER['#RIC'] == 'JNPR.N']
```

In [194]:

```
# TODO: Change the format of datetime colum.
ADI['Date'] = pd.to_datetime(ADI['Date-Time'])
FID['Date'] = pd.to_datetime(FID['Date-Time'])
FIS['Date'] = pd.to_datetime(FIS['Date-Time'])
GPN['Date'] = pd.to_datetime(GPN['Date-Time'])
JUN['Date'] = pd.to_datetime(JUN['Date-Time'])
```

```
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:1: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  """Entry point for launching an IPython kernel.
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:2: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy

/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:3: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  This is separate from the ipykernel package so we can avoid doin
g imports until
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:4: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  after removing the cwd from sys.path.
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:5: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  """
```

In [195]:

```
# TODO: Change the format of datetime colum.
ADI['Date1'] = ADI['Date'].dt.date
FID['Date1'] = FID['Date'].dt.date
FIS['Date1'] = FIS['Date'].dt.date
GPN['Date1'] = GPN['Date'].dt.date
JUN['Date1'] = JUN['Date'].dt.date
```

```
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:1: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  """Entry point for launching an IPython kernel.
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:2: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy

/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:3: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  This is separate from the ipykernel package so we can avoid doin
g imports until
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:4: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  after removing the cwd from sys.path.
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:5: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  """
```

In [196]:

```
# TODO: Add a column of 1. I added this 1s to compute number of messages for e
ach interval (day, minute)
ADI['CUM']= 1
FID['CUM']= 1
FIS['CUM']= 1
GPN['CUM']= 1
JUN['CUM']= 1
```

```
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:1: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  """Entry point for launching an IPython kernel.
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:2: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy

/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:3: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  This is separate from the ipykernel package so we can avoid doin
g imports until
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:4: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  after removing the cwd from sys.path.
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:5: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  """
```

In [197]:

```
# TODO: Compute number of messages for each day.
ADI['ACTIVITY'] = ADI.groupby(['Date1'])['CUM'].apply(lambda x: x.cumsum())
FID['ACTIVITY'] = FID.groupby(['Date1'])['CUM'].apply(lambda x: x.cumsum())
FIS['ACTIVITY'] = FIS.groupby(['Date1'])['CUM'].apply(lambda x: x.cumsum())
GPN['ACTIVITY'] = GPN.groupby(['Date1'])['CUM'].apply(lambda x: x.cumsum())
JUN['ACTIVITY'] = JUN.groupby(['Date1'])['CUM'].apply(lambda x: x.cumsum())
```

```
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:1: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  """Entry point for launching an IPython kernel.
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:2: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy

/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:3: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  This is separate from the ipykernel package so we can avoid doin
g imports until
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:4: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  after removing the cwd from sys.path.
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:5: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  """
```

In [198]:

```
# TODO: Delete quotes.
ADIa = ADI[ADI.Type != 'Quote']
FIDa = FID[FID.Type != 'Quote']
FISa = FIS[FIS.Type != 'Quote']
GPNa = GPN[GPN.Type != 'Quote']
JUNa = JUN[JUN.Type != 'Quote']
```

In [199]:

```
# TODO: Compute return.
import numpy as np
ADIa["return"] = ADIa.groupby("#RIC")['Price'].apply(lambda x: np.log(x) - np.
log(x.shift()))
FIDa["return"] = FIDa.groupby("#RIC")['Price'].apply(lambda x: np.log(x) - np.
log(x.shift()))
FISa["return"] = FISa.groupby("#RIC")['Price'].apply(lambda x: np.log(x) - np.
log(x.shift()))
GPNa["return"] = GPNa.groupby("#RIC")['Price'].apply(lambda x: np.log(x) - np.
log(x.shift()))
JUNa["return"] = JUNa.groupby("#RIC")['Price'].apply(lambda x: np.log(x) - np.
log(x.shift()))
```

```
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:3: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  This is separate from the ipykernel package so we can avoid doin
g imports until
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:4: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  after removing the cwd from sys.path.
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:5: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  """
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:6: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy

/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:7: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  import sys
```

In [200]:

```
# TODO: Compute momentum.
ADIa['momentum'] = ADIa.groupby(['Date1'])['return'].apply(lambda x: x.cumsum(
))
FIDa['momentum'] = FIDa.groupby(['Date1'])['return'].apply(lambda x: x.cumsum(
))
FISa['momentum'] = FISa.groupby(['Date1'])['return'].apply(lambda x: x.cumsum(
))
GPNa['momentum'] = GPNa.groupby(['Date1'])['return'].apply(lambda x: x.cumsum(
))
JUNa['momentum'] = JUNa.groupby(['Date1'])['return'].apply(lambda x: x.cumsum(
))
```

```
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:1: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  """Entry point for launching an IPython kernel.
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:2: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:3: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  This is separate from the ipykernel package so we can avoid doin
g imports until
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:4: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  after removing the cwd from sys.path.
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:5: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  """
```

In [201]:

```
# TODO: Compute standard deviation.
ADIb = ADIa.groupby('Date1')[['return']].std()
FIDb = FIDa.groupby('Date1')[['return']].std()
FISb = FISa.groupby('Date1')[['return']].std()
GPNb = GPNa.groupby('Date1')[['return']].std()
JUNb = JUNa.groupby('Date1')[['return']].std()
```

In [202]:

```
# TODO: Convert index into column.
ADIb['Date1'] = ADIb.index
FIDb['Date1'] = FIDb.index
FISb['Date1'] = FISb.index
GPNb['Date1'] = GPNb.index
JUNb['Date1'] = JUNb.index
```

In [203]:

```
# TODO: Change the name of columns.
ADIc = ADIb.rename (columns ={'return':'stddev'})
FIDc = FIDb.rename (columns ={'return':'stddev'})
FISc = FISb.rename (columns ={'return':'stddev'})
GPNc = GPNb.rename (columns ={'return':'stddev'})
JUNc = JUNb.rename (columns ={'return':'stddev'})
```

In [204]:

```
# TODO: Keep only last.
ADIa = ADIa.groupby('Date1', as_index=False).last()
FIDa = FIDa.groupby('Date1', as_index=False).last()
FISa = FISa.groupby('Date1', as_index=False).last()
GPNa = GPNa.groupby('Date1', as_index=False).last()
JUNa = JUNa.groupby('Date1', as_index=False).last()
```

In [205]:

```
# TODO: Merge datasets.
ADId = pd.merge(ADIc, ADIa, on='Date1', how='outer')
FIDd = pd.merge(FIDc, FIDa, on='Date1', how='outer')
FISd = pd.merge(FISc, FISa, on='Date1', how='outer')
GPNd = pd.merge(GPNc, GPNa, on='Date1', how='outer')
JUNd = pd.merge(JUNc, JUNa, on='Date1', how='outer')
```

In [206]:

```
# TODO: Read twitter data.
TWITTERADI = pd.read_csv('/Users/khaladdin/Desktop/Twitter Project/ADI.csv')
TWITTERFID = pd.read_csv('/Users/khaladdin/Desktop/Twitter Project/Fidelity.cs
v')
TWITTERFIS = pd.read_csv('/Users/khaladdin/Desktop/Twitter Project/Fiserv.csv'
)
TWITTERGPN = pd.read_csv('/Users/khaladdin/Desktop/Twitter Project/Global.csv'
)
TWITTERJUN = pd.read_csv('/Users/khaladdin/Desktop/Twitter Project/Juniper.csv
')
```

In [207]:

```
# TODO: Change the format of datetime.
TWITTERADI['Date'] =  pd.to_datetime(TWITTERADI['Date'], format='%d/%m/%Y')
TWITTERFID['Date'] =  pd.to_datetime(TWITTERFID['Date'], format='%d/%m/%Y')
TWITTERFIS['Date'] =  pd.to_datetime(TWITTERFIS['Date'], format='%d/%m/%Y')
TWITTERGPN['Date'] =  pd.to_datetime(TWITTERGPN['Date'], format='%d/%m/%Y')
TWITTERJUN['Date'] =  pd.to_datetime(TWITTERJUN['Date'], format='%d/%m/%Y')
```

In [208]:

```
# TODO: Change the format of datetime.
TWITTERADI['Date1'] = TWITTERADI['Date'].dt.date
TWITTERFID['Date1'] = TWITTERFID['Date'].dt.date
TWITTERFIS['Date1'] = TWITTERFIS['Date'].dt.date
TWITTERGPN['Date1'] = TWITTERGPN['Date'].dt.date
TWITTERJUN['Date1'] = TWITTERJUN['Date'].dt.date
```

In [209]:

```
# TODO: Compute number of retweets for each day.
TWITTERADI['RETWEET'] = TWITTERADI.groupby(['Date1'])['Number of Retweets'].ap
ply(lambda x: x.cumsum())
TWITTERFID['RETWEET'] = TWITTERFID.groupby(['Date1'])['Number of Retweets'].ap
ply(lambda x: x.cumsum())
TWITTERFIS['RETWEET'] = TWITTERFIS.groupby(['Date1'])['Number of Retweets'].ap
ply(lambda x: x.cumsum())
TWITTERGPN['RETWEET'] = TWITTERGPN.groupby(['Date1'])['Number of Retweets'].ap
ply(lambda x: x.cumsum())
TWITTERJUN['RETWEET'] = TWITTERJUN.groupby(['Date1'])['Number of Retweets'].ap
ply(lambda x: x.cumsum())
```

In [210]:

```
# TODO: Change the name of columns.
TWITTERADI = TWITTERADI.rename (columns ={'Number of Tweets':'TWEET','Number o
f Followers':'FOLLOWER'})
TWITTERFID = TWITTERFID.rename (columns ={'Number of Tweets':'TWEET','Number o
f Followers':'FOLLOWER'})
TWITTERFIS = TWITTERFIS.rename (columns ={'Number of Tweets':'TWEET','Number o
f Followers':'FOLLOWER'})
TWITTERGPN = TWITTERGPN.rename (columns ={'Number of Tweets':'TWEET','Number o
f Followers':'FOLLOWER'})
TWITTERJUN = TWITTERJUN.rename (columns ={'Number of Tweets':'TWEET','Number o
f Followers':'FOLLOWER'})
```

In [211]:

```
# TODO: Keep only last.
TWITTERADI = TWITTERADI.groupby('Date1', as_index=False).last()
TWITTERFID = TWITTERFID.groupby('Date1', as_index=False).last()
TWITTERFIS = TWITTERFIS.groupby('Date1', as_index=False).last()
TWITTERGPN = TWITTERGPN.groupby('Date1', as_index=False).last()
TWITTERJUN = TWITTERJUN.groupby('Date1', as_index=False).last()
```

In [212]:

```
# TODO: Keep some columns.
TWITTERADI1 = TWITTERADI[['Date1','RETWEET','TWEET','FOLLOWER']]
TWITTERFID1 = TWITTERFID[['Date1','RETWEET','TWEET','FOLLOWER']]
TWITTERFIS1 = TWITTERFIS[['Date1','RETWEET','TWEET','FOLLOWER']]
TWITTERGPN1 = TWITTERGPN[['Date1','RETWEET','TWEET','FOLLOWER']]
TWITTERJUN1 = TWITTERJUN[['Date1','RETWEET','TWEET','FOLLOWER']]
```

In [213]:

```
# TODO: Merge datasets.
ADIe = pd.merge(ADId, TWITTERADI1, on='Date1', how='outer')
FIDe = pd.merge(FIDd, TWITTERFID1, on='Date1', how='outer')
FISe = pd.merge(FISd, TWITTERFIS1, on='Date1', how='outer')
GPNe = pd.merge(GPNd, TWITTERGPN1, on='Date1', how='outer')
JUNe = pd.merge(JUNd, TWITTERJUN1, on='Date1', how='outer')
```

In [214]:

```
# TODO: Compute ln of Tweet.
ADIe['LOGTWEET'] = np.log(ADIe['TWEET'])
FIDe['LOGTWEET'] = np.log(FIDe['TWEET'])
FISe['LOGTWEET'] = np.log(FISe['TWEET'])
GPNe['LOGTWEET'] = np.log(GPNe['TWEET'])
JUNe['LOGTWEET'] = np.log(JUNe['TWEET'])
```

In [216]:

```
# TODO: Sort by date.
ADIe = ADIe.sort_values('Date1', ascending=True)
FIDe = FIDe.sort_values('Date1', ascending=True)
FISe = FISe.sort_values('Date1', ascending=True)
GPNe = GPNe.sort_values('Date1', ascending=True)
JUNe = JUNe.sort_values('Date1', ascending=True)
```

In [217]:

```
# TODO: Combine datasets.
Total = pd.concat([ADIe,FIDe,FISe,GPNe,JUNe])
```

In [218]:

```
# TODO: Clean missing values.
Total = Total[np.isfinite(Total['ACTIVITY'])]
```

In [219]:

```
# TODO: COmpute ln values for Retweet and Follower.
Total['LOGRETWEET'] = np.log(Total['RETWEET'])
Total['LOGFOLLOWER'] = np.log(Total['FOLLOWER'])
```

```
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:1: Ru
ntimeWarning: divide by zero encountered in log
  """Entry point for launching an IPython kernel.
```

In [220]:

```
# TODO: Create index for RIC (stock) and Date. It is import for Fixed Effect R
egression. So, we fixed stock and date
Total = Total.set_index(['#RIC','Date1'])
```

In [221]:

```
# TODO: Add constant.
Total['intercept'] = 1
```

In [222]:

```
# TODO: Clean datasets
Total2 = Total.replace([np.inf, -np.inf], np.nan)
```

In [223]:

```
# TODO: Clean datasets.
import numpy as np
Total3 = Total2[np.isfinite(Total2['TWEET'])]
```

In [225]:

```
# TODO: Keep some columns.
Total4 = Total3[['Date-Time','Price','Volume','ACTIVITY','stddev','momentum','
RETWEET','TWEET','FOLLOWER','LOGTWEET','LOGRETWEET','LOGFOLLOWER','intercept']
]
```

In [227]:

```
# TODO: Compute log of Tweet + Retweet.
Total4['twiret'] = Total4['TWEET'] + Total4['RETWEET']
Total4['logtwiret'] = np.log(Total4['twiret'])
```

```
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:1: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
  """Entry point for launching an IPython kernel.
/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:2: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pan
das-docs/stable/indexing.html#indexing-view-versus-copy
```

In [ ]:

```
# TODO: Fixed Effect Regression.
from linearmodels import PanelOLS
mod = PanelOLS(Total4.ACTIVITY, Total4[['intercept','momentum','stddev','LOGRE
TWEET','LOGTWEET','LOGFOLLOWER']], entity_effects=True)
res = mod.fit(cov_type='clustered', cluster_entity=True)
res
```

In [ ]:

```
# TODO: Fixed Effect Regression.
from linearmodels import PanelOLS
mod = PanelOLS(Total4.Volume, Total4[['intercept','momentum','stddev','LOGRETW
EET','LOGTWEET','LOGFOLLOWER']], entity_effects=True)
res = mod.fit(cov_type='clustered', cluster_entity=True)
res
```

In [242]:

```
# TODO: Winsorised.
from scipy.stats import mstats
def WinsorizeStats(Total4):
    out = mstats.winsorize(Total4, limits=[0.05, 0.05])
    return out
```

In [244]:

```
# TODO: Winsorised.
Total5 = Total4[['Price','Volume','ACTIVITY','stddev','momentum','RETWEET','TW
EET','FOLLOWER','LOGTWEET','LOGRETWEET','LOGFOLLOWER','intercept','twiret','lo
gtwiret']].apply(WinsorizeStats, axis=0)
```

In [ ]:

```
# TODO: Fixed Effect Regression.
from linearmodels import PanelOLS
mod = PanelOLS(Total5.ACTIVITY, Total5[['intercept','momentum','stddev','LOGRE
TWEET','LOGTWEET','LOGFOLLOWER']], entity_effects=True)
res = mod.fit(cov_type='clustered', cluster_entity=True)
res
```

In [ ]:

```
# TODO: Fixed Effect Regression.
from linearmodels import PanelOLS
mod = PanelOLS(Total5.ACTIVITY, Total5[['intercept','momentum','stddev','logtw
iret','LOGFOLLOWER']], entity_effects=True)
res = mod.fit(cov_type='clustered', cluster_entity=True)
res
```