In [1]:

```
import pandas as pd
import numpy as np
from pandas import DataFrame
import os
```

Premium

In [4]:

```
fname = "STK_SSEandHKEX_AHRatio.txt"

with open(fname,"r") as fhand:
    chunk = fhand.readlines()[:5]
    print(chunk)
```

```
['InstitutionID\tTradingDate\tSecurityIDA\tSymbolA\tShortNameA\tSecurityID
H\tSymbolH\tShortNameH\tClosePriceA\tClosePriceH\tPreConsideration\tConsid
eration\n', '机构ID\t交易日期\tA股证券ID\tA股股票代码\tA股股票简称\tH股证券ID
\tH股股票代码\tH股股票简称\tA股收盘价\tH股收盘价\t前一交易日AH股比价\tAH股比价
\n', '没有单位\t没有单位\t没有单位\t没有单位\t没有单位\t没有单位\t没有单位\t没有
单位\t元\t港元\t没有单位\t没有单位\n', '103612\t2004-12-01\t201000003467\t600
115\t东方航空\t201000001817\t00670\t中国东方航空股份\t4.330\t1.730\t2.3152\t
2.3525\n', '103533\t1996-10-07\t\t\t\t201000002194\t01138\t中远海能\t\t.620
\t\t\n']
```

In [2]:

```
data_type = {"SymbolA": "S","SymbolH":"S","ClosePriceA":"f4","ClosePriceH":"f4",
                           "Consideration":"f4"}
df = pd.read_table("STK_SSEandHKEX_AHRatio.txt",skiprows = [1,2],encoding= 'unicode_esc
ape',
                   usecols= ["TradingDate","SymbolA","SymbolH","ClosePriceA","ClosePrice
H",
                             "Consideration"],dtype = data_type, parse_dates = True,inde
x_col = 0)
df.head()
```

Out[2]:

TradingDate	SymbolA	SymbolH	ClosePriceA	ClosePriceH	Consideration
2004-12-01	600115	00670	4.330000	1.730	2.3525
1996-10-07	NaN	01138	NaN	0.620	NaN
2006-07-21	NaN	00991	NaN	4.925	NaN
2003-01-09	NaN	01055	NaN	2.400	NaN
2009-11-19	600585	00914	45.540001	49.750	1.0390

In [ ]:

```
# 1. Make two dictionaries mapping Symbol A to SymbolH, and the other way round
# 2. Keep TradingDate, SymbolA, and Consideration only
# 3. Drop na's
# 4. Downsampling data to MonthEnd or BusinessMonthEnd frequency
```

In [28]:

```
# 1. Make two dictionaries mapping Symbol A to SymbolH, and the other way round
tickers = df[["SymbolA","SymbolH"]].dropna()
tickers = tickers.drop_duplicates()# Only consider certain columns for identifying dupl
icates,
                                   # by default use all of the columns.

print("Unique Rows: ")
print(tickers[:5])
print()

tickers_a = tickers["SymbolA"]
tickers_h = tickers["SymbolH"]


a_to_h = {}
for key, value in zip(tickers_a,tickers_h):
    a_to_h[key] = value

print("SymbolA to SymbolH: ")
print(list(a_to_h.items())[:5]) #dict.items is not subscriptable, we need to make a lis
t of it
print()

h_to_a = {}
for key, value in zip(tickers_h,tickers_a):
    h_to_a[key] = value
print("SymbolH to SymbolA: ")
print(list(h_to_a.items())[:5])

# Save the two dictionary to via json
import json

symbol_map = {
                "a_to_h": a_to_h,
                "h_to_a": h_to_a
                    }

with open("Symbol_Map","w") as fhand:
    json.dump(symbol_map, fhand)

del([symbol_map,tickers])
```

```
Unique Rows:
            SymbolA SymbolH
TradingDate
2004-12-01   600115   00670
2009-11-19   600585   00914
2010-05-18   600012   00995
2011-09-29   601618   01618
2010-12-06   601899   02899

SymbolA to SymbolH:
[('600115', '00670'), ('600585', '00914'), ('600012', '00995'), ('601618',
'01618'), ('601899', '02899')]

SymbolH to SymbolA:
[('00670', '600115'), ('00914', '600585'), ('00995', '600012'), ('01618',
'601618'), ('02899', '601899')]
```

In [32]:

```
# 2. Keep TradingDate, SymbolA, and Consideration only
cols_to_keep = ["SymbolA","Consideration"]
df = df.loc[:,cols_to_keep]

# 3. Drop na's of Consideration
df = df.dropna()
```

In [37]:

```
# 4. Downsampling data to MonthEnd or BusinessMonthEnd frequency
df = df.reset_index()
df.head()
```

Out[37]:

	TradingDate	SymbolA	Consideration
0	2004-12-01	600115	2.3525
1	2009-11-19	600585	1.0390
2	2010-05-18	600012	1.3470
3	2011-09-29	601618	2.2839
4	2010-12-06	601899	1.3482

In [45]:

```
df = df.pivot("TradingDate","SymbolA","Consideration")
df = df.resample("M").mean().stack() # Down sampling and take the mean of daily Conside
ration
df.name = "AH_Premium"
df.head()
```

Out[45]:

```
TradingDate  SymbolA
1993-08-31   600600     3.246700
1993-09-30   600600     2.821629
1993-10-31   600600     2.188340
             600685     2.214800
1993-11-30   600600     1.881818
Name: AH_Premium, dtype: float32
```

In [50]:

```
# Save data
df = df.reset_index()
df.to_csv("ah_premium.csv",index = False)
```

Volums in H-Share Market

In [93]:

```
filename = "STK_SSEtoHKEX_Quotation.txt"
data_type = {"TradingDate": "S","Symbol":"S","CurrencyCode":"S","ClosePrice":"f","Volum
e":"f","Amount":"f"}

df = pd.read_table(filename,skiprows = [1,2],encoding= 'unicode_escape',usecols= list(d
ata_type.keys()),
                   dtype = data_type, parse_dates = True,index_col = 0)
df = df.reset_index()
df.head()
```

Out[93]:

	TradingDate	Symbol	CurrencyCode	ClosePrice	Volume	Amount
0	1982-03-29	00001	HKD	3.00	3378800.0	10035000.0
1	1982-03-30	00001	HKD	2.96	5313100.0	15939000.0
2	1982-03-31	00001	HKD	2.96	2506000.0	7418000.0
3	1982-04-01	00001	HKD	3.04	3392000.0	10244000.0
4	1982-04-02	00001	HKD	3.15	5547900.0	17282000.0

In [94]:

```
# 1. Keep columns: TradingDate, Symbol, Volume
# 2. Map Symbol's elements to SymbolA to a new column called SymbolA
# 3. Drop Column Symbol
# 4. Downsampling with MonthEnd Frequency by adding daily volume up
# 5. Save it for Concatenation later...
```

In [95]:

```
# 1. Keep columns: TradingDate, Symbol, Volume
cols_to_keep = ["TradingDate","Symbol","Volume"]
df = df.loc[:,cols_to_keep]
```

In [96]:

```
# 2. Map Symbol's elements to SymbolA to a new column called SymbolA
df["SymbolA"] = df["Symbol"].map(h_to_a) # We only matched A-H shares, so there will be NAs.
df = df.dropna()
df.head()
```

Out[96]:

	TradingDate	Symbol	Volume	SymbolA
136605	1997-06-23	00038	108932000.0	601038
136606	1997-06-24	00038	46263000.0	601038
136607	1997-06-25	00038	68906000.0	601038
136608	1997-06-26	00038	21767000.0	601038
136609	1997-06-27	00038	16838000.0	601038

In [97]:

```
# 3. Drop Column Symbol
df = df.drop("Symbol",axis = 1)
df.head()
```

Out[97]:

	TradingDate	Volume	SymbolA
136605	1997-06-23	108932000.0	601038
136606	1997-06-24	46263000.0	601038
136607	1997-06-25	68906000.0	601038
136608	1997-06-26	21767000.0	601038
136609	1997-06-27	16838000.0	601038

In [98]:

```
# 4. Downsampling with MonthEnd Frequency by adding daily volume up
## Need to drop duplicates before pivoting the data
df = df.drop_duplicates()
df = df.pivot("TradingDate","SymbolA","Volume")
df.head()
```

Out[98]:

SymbolA	000002	000039	000063	000157	000166	000338	000488	000513	000585	000
TradingDate										
1993-07-15	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1993-07-16	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1993-07-19	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1993-07-20	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1993-07-21	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

5 rows × 118 columns

In [99]:

```
df = df.resample("M").sum().stack()
df = df.reset_index()
df.head()
```

Out[99]:

	TradingDate	SymbolA	0
0	1993-07-31	000002	0.0
1	1993-07-31	000039	0.0
2	1993-07-31	000063	0.0
3	1993-07-31	000157	0.0
4	1993-07-31	000166	0.0

In [100]:

```
df = df.rename(columns={0:"VolumeH"})
```

In [101]:

```
df.head()
```

Out[101]:

	TradingDate	SymbolA	VolumeH
0	1993-07-31	000002	0.0
1	1993-07-31	000039	0.0
2	1993-07-31	000063	0.0
3	1993-07-31	000157	0.0
4	1993-07-31	000166	0.0

In [102]:

```
# 5. Save it for Concatenation later..
df.to_csv("H_volume.csv",index = False)
```

Volums in A-Share Market

In [105]:

```
filename = "STK_HKEXtoSSE_Quotation.txt"

data_type = {"TradingDate": "S","Symbol":"S","CurrencyCode":"S","ClosePrice":"f","Volum
e":"f","Amount":"f"}

df = pd.read_table(filename,skiprows = [1,2],encoding= 'unicode_escape',usecols= list(d
ata_type.keys()),
                   dtype = data_type, parse_dates = True,index_col = 0)
df = df.reset_index()

df.head()
```

Out[105]:

	TradingDate	Symbol	CurrencyCode	ClosePrice	Volume	Amount
0	1991-04-03	000001	CNY	49.000000	100.0	5000.0
1	1991-04-04	000001	CNY	48.759998	300.0	15000.0
2	1991-04-05	000001	CNY	48.520000	200.0	10000.0
3	1991-04-06	000001	CNY	48.279999	700.0	34000.0
4	1991-04-08	000001	CNY	48.040001	200.0	10000.0

In [ ]:

```
# 1. Keep columns: TradingDate, Symbol, Volume
# 2. Rename Symbol as SymbolA
# 3. Downsampling by MonthEnd Frequency and sum up the daily volume
# 4. Save it for Concatenation later...
```

In [106]:

```
# 1. Keep columns: TradingDate, Symbol, Volume
cols_to_keep = ["TradingDate","Symbol","Volume"]
df = df.loc[:,cols_to_keep]

# 2. Rename Symbol as SymbolA
df.rename(columns = {"Symbol":"SymbolA","Volume":"VolumeA"},inplace = True)
df.head()
```

Out[106]:

	TradingDate	SymbolA	VolumeA
0	1991-04-03	000001	100.0
1	1991-04-04	000001	300.0
2	1991-04-05	000001	200.0
3	1991-04-06	000001	700.0
4	1991-04-08	000001	200.0

In [107]:

```
# 3. Downsampling by MonthEnd Frequency and sum up the daily volume
df = df.drop_duplicates()
df = df.pivot("TradingDate","SymbolA","VolumeA")
df = df.resample("M").sum().stack()
df.head()
```

Out[107]:

```
TradingDate  SymbolA
1990-12-31   000001     0.0
             000002     0.0
             000005     0.0
             000006     0.0
             000008     0.0
dtype: float32
```

In [109]:

```
df = df.reset_index()
df.head()
```

Out[109]:

	TradingDate	SymbolA	0
0	1990-12-31	000001	0.0
1	1990-12-31	000002	0.0
2	1990-12-31	000005	0.0
3	1990-12-31	000006	0.0
4	1990-12-31	000008	0.0

In [110]:

```
# 4. Save it for Concatenation later...
df = df.rename(columns= {0:"VolumeA"})
df.head()
```

Out[110]:

	TradingDate	SymbolA	VolumeA
0	1990-12-31	000001	0.0
1	1990-12-31	000002	0.0
2	1990-12-31	000005	0.0
3	1990-12-31	000006	0.0
4	1990-12-31	000008	0.0

In [112]:

```
df.to_csv("A_volume.csv",index = False)
```

Numbers of Shares

In [113]:

```
del(df)
data_type = {"Stkcd": "S","Shrchgdt":"S","Nshrttl":"f","Nshrstt":"f",
                          "Nshra":"f","Nshra":"f","Nshrh":"f"}
filename = "TRD_Capchg.txt"
df2 = pd.read_table(filename,skiprows = [1,2],encoding= 'unicode_escape',
                  usecols= ['Stkcd', 'Shrchgdt', 'Nshrttl', 'Nshrstt', 'Nshra', 'Nshrh'
],
                    dtype = data_type, parse_dates = True, index_col = 1)
df2 = df2[df2["Nshrh"] != 0] # The Number of H-share is not zeor, to get the AH shares.
df2.reset_index(inplace = True)
df2.rename(columns = {"Shrchgdt":"TradingDate", "Stkcd": "SymbolA"},inplace = True)
df2.head()
```

Out[113]:

	TradingDate	SymbolA	Nshrttl	Nshrstt	Nshra	Nshrh
0	2004-12-09	000063	948000128.0	462372832.0	302054400.0	148529040.0
1	2004-12-16	000063	959521664.0	462272384.0	302054400.0	160151040.0
2	2005-11-10	000063	959521664.0	462272384.0	302054400.0	160151040.0
3	2005-12-29	000063	959521664.0	392079904.0	376285952.0	160151040.0
4	2006-12-29	000063	959521664.0	310982752.0	487105792.0	160151040.0

In [114]:

```
# 1. Dropping the rows we don't need: Non-AH shares
# 2. Rename the columns in an easy to understand manner
# 3. Shifting data to MonthEnd Frequency
# 4. Save the data for later concatenation
```

In [117]:

```
# 1. Dropping the rows we don't need: Non-AH shares
df2 = df2.loc[df2["SymbolA"].isin(tickers_a),]

# 2. Rename the columns in an easy to understand manner
df2.rename(columns = {"Nshrttl":"Total_shares","Nshrstt":"State_own_shares","Nshra":"Ou
tstand_A","Nshrh":"Outstand_H"},inplace = True)
df2.head()
```

Out[117]:

	TradingDate	SymbolA	Total_shares	State_own_shares	Outstand_A	Outstand_H
0	2004-12-09	000063	948000128.0	462372832.0	302054400.0	148529040.0
1	2004-12-16	000063	959521664.0	462272384.0	302054400.0	160151040.0
2	2005-11-10	000063	959521664.0	462272384.0	302054400.0	160151040.0
3	2005-12-29	000063	959521664.0	392079904.0	376285952.0	160151040.0
4	2006-12-29	000063	959521664.0	310982752.0	487105792.0	160151040.0

In [119]:

```
# 3. Shifting data to MonthEnd Frequency
df2 = df2.set_index("TradingDate").shift(freq = "M")
df2.head()
```

Out[119]:

TradingDate	SymbolA	Total_shares	State_own_shares	Outstand_A	Outstand_H
2004-12-31	000063	948000128.0	462372832.0	302054400.0	148529040.0
2004-12-31	000063	959521664.0	462272384.0	302054400.0	160151040.0
2005-11-30	000063	959521664.0	462272384.0	302054400.0	160151040.0
2005-12-31	000063	959521664.0	392079904.0	376285952.0	160151040.0
2006-12-31	000063	959521664.0	310982752.0	487105792.0	160151040.0

In [122]:

```
df3 = df2.reset_index().pivot_table(index = "TradingDate",
                                    columns = "SymbolA",
                                    values = ["Total_shares","State_own_shares", "Outstand_A"
,"Outstand_H"])
df3.columns.names = ["ShareNum","SymbolA"]
df3.head()
```

Out[122]:

ShareNum	Outstand_A									
SymbolA	000002	000039	000063	000157	000166	000338	000488	000513	000585	00(
TradingDate										
1993-08-31	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1993-10-31	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1993-11-30	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1994-01-31	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1994-05-31	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

5 rows × 472 columns

In [130]:

```
# 4. Resampling with mean of number of shares
df3 =df3.resample("M").mean().stack("SymbolA").reset_index()
df3.head()
```

Out[130]:

ShareNum	TradingDate	SymbolA	Outstand_A	Outstand_H	State_own_shares	Total_shar
0	1993-08-31	600600	90000000.0	3.468500e+08	3.998200e+08	9.000000e+
1	1993-10-31	600685	98082600.0	1.573980e+08	2.108001e+08	4.946776e+
2	1993-11-30	600688	250000000.0	1.680000e+09	4.000000e+09	6.230000e+
3	1994-01-31	600806	53400000.0	6.500000e+07	1.023977e+08	2.450074e+
4	1994-01-31	600808	410000000.0	1.732930e+09	4.034560e+09	6.455300e+

In [141]:

```
# 5. Save the data for later concatenation
df3.to_csv("Number of Shares.csv",index = False)
```

Merging Tables and Cocatenation

In [143]:

```
df1 = pd.read_csv("ah_premium.csv",parse_dates = True, index_col = [0,1])
df2 = pd.read_csv("H_volume.csv",parse_dates = True, index_col = [0,1])
df3 = pd.read_csv("A_volume.csv",parse_dates = True, index_col = [0,1])
df4 = pd.read_csv("Number of Shares.csv",parse_dates = True, index_col = [0,1])
```

In [144]:

```
df1.head()
```

Out[144]:

		AH_Premium
TradingDate	**SymbolA**	
1993-08-31	**600600**	3.246700
1993-09-30	**600600**	2.821629
1993-10-31	**600600**	2.188340
	600685	2.214800
1993-11-30	**600600**	1.881818

In [145]:

```
df2.head()
```

Out[145]:

		VolumeH
TradingDate	**SymbolA**	
1993-07-31	**2**	0.0
	39	0.0
	63	0.0
	157	0.0
	166	0.0

In [146]:

```
df3.head()
```

Out[146]:

		VolumeA
TradingDate	**SymbolA**	
1990-12-31	**1**	0.0
	2	0.0
	5	0.0
	6	0.0
	8	0.0

In [147]:

```
df4.head()
```

Out[147]:

		Outstand_A	Outstand_H	State_own_shares	Total_shares
TradingDate	**SymbolA**				
1993-08-31	**600600**	90000000.0	3.468500e+08	3.998200e+08	9.000000e+08
1993-10-31	**600685**	98082600.0	1.573980e+08	2.108001e+08	4.946776e+08
1993-11-30	**600688**	250000000.0	1.680000e+09	4.000000e+09	6.230000e+09
1994-01-31	**600806**	53400000.0	6.500000e+07	1.023977e+08	2.450074e+08
	600808	410000000.0	1.732930e+09	4.034560e+09	6.455300e+09

In [153]:

```
data = df1.merge(df2,left_index=True,right_index=True, how = "left").merge(df3,left_ind
ex=True,right_index=True, how = "left")
data.head()
```

Out[153]:

		AH_Premium	VolumeH	VolumeA
TradingDate	**SymbolA**			
1993-08-31	**600600**	3.246700	231292000.0	23222500.0
1993-09-30	**600600**	2.821629	83585000.0	27336200.0
1993-10-31	**600600**	2.188340	53610000.0	19182180.0
	600685	2.214800	135815000.0	25725624.0
1993-11-30	**600600**	1.881818	45620000.0	23882448.0

In [155]:

```
data = data.join(df4, how = "left")
data.head()
```

Out[155]:

		AH_Premium	VolumeH	VolumeA	Outstand_A	Outstand_H	Sta
TradingDate	**SymbolA**						
1993-08-31	**600600**	3.246700	231292000.0	23222500.0	90000000.0	346850000.0	
1993-09-30	**600600**	2.821629	83585000.0	27336200.0	NaN	NaN	
1993-10-31	**600600**	2.188340	53610000.0	19182180.0	NaN	NaN	
	600685	2.214800	135815000.0	25725624.0	98082600.0	157398000.0	
1993-11-30	**600600**	1.881818	45620000.0	23882448.0	NaN	NaN	

In [156]:

```
data = data.reset_index()
data.head()
```

Out[156]:

	TradingDate	SymbolA	AH_Premium	VolumeH	VolumeA	Outstand_A	Outstand_H
0	1993-08-31	600600	3.246700	231292000.0	23222500.0	90000000.0	346850000.0
1	1993-09-30	600600	2.821629	83585000.0	27336200.0	NaN	NaN
2	1993-10-31	600600	2.188340	53610000.0	19182180.0	NaN	NaN
3	1993-10-31	600685	2.214800	135815000.0	25725624.0	98082600.0	157398000.0
4	1993-11-30	600600	1.881818	45620000.0	23882448.0	NaN	NaN

In [161]:

```
# Ffill missing number of shares values by SymbolA

# First define a function to returen a chunk a data by group(SymbolA)
def ffill_number(grouped):
    chunk = grouped.sort_values(by = "TradingDate").ffill()
    return chunk
# Data Aggregation
data2 = data.groupby("SymbolA").apply(ffill_number)
data2.head()
```

Out[161]:

		TradingDate	SymbolA	AH_Premium	VolumeH	VolumeA	Outstand_A
SymbolA							
2	**8628**	2014-06-30	2	0.753675	106431160.0	1.184962e+09	9.676243e+
	8713	2014-07-31	2	0.714136	360118820.0	1.937779e+09	9.671913e+
	8798	2014-08-31	2	0.751281	274480130.0	1.346537e+09	9.671913e+
	8883	2014-09-30	2	0.817415	222259360.0	1.556464e+09	9.671913e+
	8968	2014-10-31	2	0.820806	187117630.0	1.327366e+09	9.671913e+

In [167]:

```
# Clean up the data
data = data2.drop("SymbolA",axis = 1).reset_index().drop("level_1",axis = 1)
data.head()
```

Out[167]:

	SymbolA	TradingDate	AH_Premium	VolumeH	VolumeA	Outstand_A	Outstanc
0	2	2014-06-30	0.753675	106431160.0	1.184962e+09	9.676243e+09	1.314956e
1	2	2014-07-31	0.714136	360118820.0	1.937779e+09	9.671913e+09	1.314956e
2	2	2014-08-31	0.751281	274480130.0	1.346537e+09	9.671913e+09	1.314956e
3	2	2014-09-30	0.817415	222259360.0	1.556464e+09	9.671913e+09	1.314956e
4	2	2014-10-31	0.820806	187117630.0	1.327366e+09	9.671913e+09	1.314956e

In [170]:

```
# Take a look at the indexed data
data.set_index(["TradingDate","SymbolA"]).swaplevel(axis = 0)
```

Out[170]:

		AH_Premium	VolumeH	VolumeA	Outstand_A	Outstand_H
SymbolA	**TradingDate**					
2	**2014-06-30**	0.753675	1.064312e+08	1.184962e+09	9.676243e+09	1.314956e+0
	2014-07-31	0.714136	3.601188e+08	1.937779e+09	9.671913e+09	1.314956e+0
	2014-08-31	0.751281	2.744801e+08	1.346537e+09	9.671913e+09	1.314956e+0
	2014-09-30	0.817415	2.222594e+08	1.556464e+09	9.671913e+09	1.314956e+0
	2014-10-31	0.820806	1.871176e+08	1.327366e+09	9.671913e+09	1.314956e+0
	2014-11-30	0.838980	2.815322e+08	2.761624e+09	9.671913e+09	1.314956e+0
	2014-12-31	0.897586	2.681557e+08	6.499493e+09	9.671913e+09	1.314956e+0
	2015-01-31	0.953920	3.099107e+08	5.549189e+09	9.706677e+09	1.314956e+0
	2015-02-28	0.919753	1.429760e+08	2.598023e+09	9.706677e+09	1.314956e+0
	2015-03-31	0.935841	2.548235e+08	5.738467e+09	9.706677e+09	1.314956e+0
	2015-04-30	0.929016	3.637123e+08	8.827347e+09	9.706677e+09	1.314956e+0
	2015-05-31	0.916905	2.504307e+08	7.266022e+09	9.706677e+09	1.314956e+0
	2015-06-30	0.942362	2.104483e+08	6.716989e+09	9.706677e+09	1.314956e+0
	2015-07-31	0.990059	3.094091e+08	8.240797e+09	9.717758e+09	1.314956e+0
	2015-08-31	0.970067	2.119152e+08	3.167320e+09	9.717758e+09	1.314956e+0
	2015-09-30	0.959442	1.883097e+08	1.420292e+09	9.712425e+09	1.314956e+0
	2015-10-31	0.923625	1.416010e+08	9.011957e+08	9.712425e+09	1.314956e+0
	2015-11-30	0.921010	1.539477e+08	1.526507e+09	9.712425e+09	1.314956e+0
	2015-12-31	1.173405	3.735736e+08	4.023962e+09	9.712425e+09	1.314956e+0
	2016-01-31	1.581725	3.604593e+08	0.000000e+00	9.714456e+09	1.314956e+0
	2016-02-29	1.697588	7.151562e+07	0.000000e+00	9.714456e+09	1.314956e+0
	2016-03-31	1.553271	1.745995e+08	0.000000e+00	9.714456e+09	1.314956e+0
	2016-04-30	1.505440	1.222133e+08	0.000000e+00	9.708347e+09	1.314956e+0
	2016-05-31	1.604543	1.289761e+08	0.000000e+00	9.708347e+09	1.314956e+0
	2016-06-30	1.667325	3.110521e+08	0.000000e+00	9.708347e+09	1.314956e+0
	2016-07-31	1.316960	5.610594e+08	4.861542e+09	9.708108e+09	1.314956e+0
	2016-08-31	1.350677	4.934544e+08	5.299792e+09	9.708108e+09	1.314956e+0
	2016-09-30	1.420030	2.141210e+08	2.872686e+09	9.708108e+09	1.314956e+0
	2016-10-31	1.493007	1.956792e+08	1.308261e+09	9.708108e+09	1.314956e+0
	2016-11-30	1.373341	2.225266e+08	2.383330e+09	9.708108e+09	1.314956e+0
...	**...**	...	...	...	...	.
603993	**2017-09-30**	1.996452	2.150631e+09	5.887097e+09	1.295373e+10	3.933468e+0
	2017-10-31	1.741159	1.741082e+09	2.336210e+09	1.295373e+10	3.933468e+0
	2017-11-30	1.627259	1.836352e+09	3.445109e+09	1.295373e+10	3.933468e+0
	2017-12-31	1.681195	1.101044e+09	2.764047e+09	1.295373e+10	3.933468e+0
	2018-01-31	1.614595	1.522489e+09	4.898780e+09	1.295373e+10	3.933468e+0

SymbolA	TradingDate	AH_Premium	VolumeH	VolumeA	Outstand_A	Outstand_H
	2018-02-28	1.600133	1.659869e+09	3.313322e+09	1.295373e+10	3.933468e+0
	2018-03-31	1.737552	1.527188e+09	6.323814e+09	1.295373e+10	3.933468e+0
	2018-04-30	1.750265	1.039489e+09	2.391056e+09	1.295373e+10	3.933468e+0
	2018-05-31	1.740495	1.028151e+09	2.510955e+09	1.295373e+10	3.933468e+0
	2018-06-30	1.879340	1.317552e+09	1.522716e+09	1.295373e+10	3.933468e+0
	2018-07-31	1.916457	1.171416e+09	2.969335e+09	1.766577e+10	3.933468e+0
	2018-08-31	1.655735	1.157676e+09	2.956640e+09	1.766577e+10	3.933468e+0
	2018-09-30	1.670161	9.192936e+08	2.061657e+09	1.766577e+10	3.933468e+0
	2018-10-31	1.568418	6.912644e+08	2.220735e+09	1.766577e+10	3.933468e+0
	2018-11-30	1.452168	7.914282e+08	3.087786e+09	1.766577e+10	3.933468e+0
	2018-12-31	1.477706	3.550263e+08	1.611057e+09	1.766577e+10	3.933468e+0
	2019-01-31	1.516886	4.519112e+08	1.407568e+09	1.766577e+10	3.933468e+0
	2019-02-28	1.449493	9.973282e+08	3.032980e+09	1.766577e+10	3.933468e+0
	2019-03-31	1.617419	7.777743e+08	4.121255e+09	1.766577e+10	3.933468e+0
	2019-04-30	1.644674	8.761210e+08	3.543517e+09	1.766577e+10	3.933468e+0
	2019-05-31	1.793737	8.329630e+08	1.630053e+09	1.766577e+10	3.933468e+0
	2019-06-30	1.851700	2.948736e+08	1.477881e+09	1.766577e+10	3.933468e+0
	2019-07-31	1.848482	5.055103e+08	1.785900e+09	1.766577e+10	3.933468e+0
	2019-08-31	1.843418	9.250685e+08	3.727853e+09	1.766577e+10	3.933468e+0
	2019-09-30	1.633025	7.019603e+08	2.051498e+09	1.766577e+10	3.933468e+0
	2019-10-31	1.532572	3.247945e+08	7.329439e+08	1.766577e+10	3.933468e+0
	2019-11-30	1.491395	4.929850e+08	1.311308e+09	1.766577e+10	3.933468e+0
	2019-12-31	1.432395	7.148947e+08	3.568784e+09	1.766577e+10	3.933468e+0
	2020-01-31	1.440781	8.210302e+08	3.552448e+09	1.766577e+10	3.933468e+0
	2020-02-29	1.386200	2.494346e+08	9.476508e+08	1.766577e+10	3.933468e+0

15265 rows × 7 columns

In [171]:

```
# Save the data
#data.to_csv("AH_data.csv")
```

In [ ]:

```
```