Data Science HW2

R07921010 余奇安

Problem1:

1.

Use panda to read data from csv and use DataFram to describe the data.

df.describe(), df.head()

```
import numpy as np
import pandas as pd
import mysql.connector
import sqlalchemy

df = pd.read_csv('data.csv', encoding='big5', skiprows=1)
mydb = mysql.connector.connect(host='localhost',user='root',passwd='cayu4686')
sql_engine = sqlalchemy.create_engine("mysql+mysqlconnector://root:cayu4686@localhost/test")
# df.to_sql('tsmc',sql_engine)
df.describe()
```

	開盤價	最高價	最低價	收盤價	漲跌價差
count	89.00000	89.000000	89.000000	89.000000	89.000000
mean	254.08427	255.735955	252.269663	254.247191	0.629213
std	12.32289	12.116642	12.399151	12.229604	3.496368
min	231.50000	232.000000	229.500000	232.000000	-6.500000
25%	244.50000	247.000000	242.500000	246.000000	-1.500000
50%	253.00000	254.500000	251.000000	254.000000	0.500000
75%	263.50000	264.500000	261.500000	263.500000	2.500000
max	283.50000	286.500000	282.500000	286.500000	9.500000

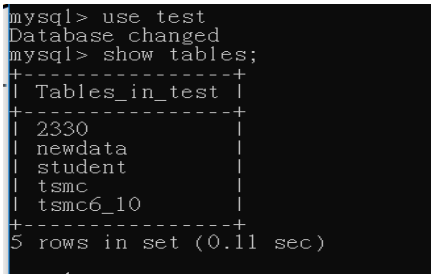
The columns that do not show up such as 成交股數,成交金額 is because the data type is string. As the TA mention, we do not need further process for these data.

```
df.head()
```

	日期	成交股數	成交金額	開盤價	最高價	最低價	收盤價	漲跌價差	成交筆數
0	108/06/03	36,687,092	8,651,389,851	235.5	238.5	232.0	238.0	2.5	8,857
1	108/06/04	24,443,428	5,745,045,809	237.5	238.0	233.0	233.0	-5.0	11,779
2	108/06/05	35,901,584	8,461,930,934	238.0	238.0	234.0	235.0	2.0	12,064
3	108/06/06	34,651,731	8,008,768,323	231.5	232.0	229.5	232.0	-3.0	14,723
4	108/06/10	35,521,888	8,442,986,620	237.5	240.0	234.5	240.0	8.0	15,588

2.

I create database “test” first, and use sqlalchemy.creat_engine to connect to it. After the execute “df.to_sql(‘tsmc’, sql_engine)”, the dataframe store to sql with the table tsmc.



The table in test dataset which open in SQL client shell.

Problem2:

1.

```
# df.to_sql('tsmc6_10',sql_engine)
amt_avg = pd.read_sql_query('select AVG(成交股數) from tsmc6_10',sql_engine)
amt_sum = pd.read_sql_query('select SUM(成交股數) from tsmc6_10',sql_engine)
vol_avg = pd.read_sql_query('select AVG(成交金額) from tsmc6_10',sql_engine)
vol_sum = pd.read_sql_query('select SUM(成交金額) from tsmc6_10',sql_engine)
print(amt_avg)
print(amt_sum)
print(vol_avg)
print(vol_sum)
```

```
   AVG(成交股數)
0  3.269030e+07
   SUM(成交股數)
0  2.909436e+09
   AVG(成交金額)
0  8.304750e+09
   SUM(成交金額)
0  7.391228e+11
```

成交股數平均:32690295.5843

成交股數總數:2909436307

成交金額平均:8304750158.4831

成交金額總數:739122764105

2. 5-day moving average of daily closing price and trading volume.

```
# df3 = df['收盤價'].rolling(5).mean().cloumn
# df3['m5_avg_收盤價'] = cl_price_m5_avg
# df3['m5_avg_成交股數'] = vol_m5_avg
# df3
# for i in range(len(df)):
#     print(df['日期'][i], cl_price_m5_avg[i], vol_m5_avg[i])
cl_price_m5_avg = df['收盤價'].rolling(5).mean().rename("clpr_m5_avg")
vol_m5_avg = df['成交股數'].rolling(5).mean().rename("vol_m5_avg")
pd.set_option('display.max_rows', None)
df3 = pd.concat([df['日期'],cl_price_m5_avg,vol_m5_avg],axis=1)
```

I create another data frame df3, with column name [日期, clpar_m5_avg, vol_m5_avg]

all the table value are show as followed.

	日期	clpr_m5_avg	vol_m5_avg
0	108/06/03	nan	nan
1	108/06/04	nan	nan
2	108/06/05	nan	nan
3	108/06/06	nan	nan
4	108/06/10	235.6000	33441144.6000
5	108/06/11	236.9000	33042060.2000
6	108/06/12	239.5000	34235285.8000
7	108/06/13	240.5000	33801313.8000
8	108/06/14	241.3000	33951678.8000
9	108/06/17	239.9000	37185703.6000
10	108/06/18	238.1000	36006987.0000
11	108/06/19	237.7000	39971950.2000
12	108/06/20	238.7000	41169063.6000
13	108/06/21	241.2000	46515613.2000
14	108/06/24	242.8000	44869312.2000
15	108/06/25	243.4000	45056751.4000
16	108/06/26	241.5000	40763885.2000
17	108/06/27	240.6000	41221993.8000
18	108/06/28	238.7000	34411775.4000
19	108/07/01	240.2000	38993849.0000
20	108/07/02	242.3000	38413232.8000
21	108/07/03	243.9000	38569179.4000
22	108/07/04	244.6000	33905860.6000
23	108/07/05	245.4000	33752233.0000
24	108/07/08	244.2000	26316239.8000
25	108/07/09	242.8000	24257596.8000
26	108/07/10	243.7000	23110068.2000
27	108/07/11	244.9000	25631078.0000
28	108/07/12	246.4000	24941349.6000
29	108/07/15	248.8000	25956027.8000
30	108/07/16	251.6000	27876480.2000
31	108/07/17	252.6000	29983383.6000
32	108/07/18	253.4000	27905840.8000
33	108/07/19	255.1000	35108114.0000
34	108/07/22	257.0000	34627981.8000
35	108/07/23	258.6000	35254376.6000
36	108/07/24	261.2000	33287254.8000
37	108/07/25	263.4000	35245248.8000
38	108/07/26	263.8000	27422903.4000
39	108/07/29	263.2000	24744839.6000
40	108/07/30	262.4000	23469317.4000
41	108/07/31	261.3000	25840229.8000

	日期	clpr_m5_avg	vol_m5_avg
42	108/08/01	259.6000	25722026.0000
43	108/08/02	257.7000	32401540.6000
44	108/08/05	254.8000	39274020.8000
45	108/08/06	252.5000	46478274.2000
46	108/08/07	250.2000	45092246.0000
47	108/08/08	249.6000	45098794.0000
48	108/08/12	249.5000	39212185.8000
49	108/08/13	249.5000	33654063.0000
50	108/08/14	249.7000	29138027.8000
51	108/08/15	249.7000	28906418.2000
52	108/08/16	249.0000	28017708.0000
53	108/08/19	249.2000	28090890.2000
54	108/08/20	250.8000	27205967.6000
55	108/08/21	251.8000	23992486.2000
56	108/08/22	253.0000	22764371.8000
57	108/08/23	253.8000	20579604.2000
58	108/08/26	253.1000	21908025.4000
59	108/08/27	252.2000	28516916.8000
60	108/08/28	251.7000	27519234.0000
61	108/08/29	251.7000	27109019.4000
62	108/08/30	252.7000	31214284.2000
63	108/09/02	254.5000	27821531.0000
64	108/09/03	255.3000	22305839.8000
65	108/09/04	256.4000	23989210.0000
66	108/09/05	258.2000	29780402.2000
67	108/09/06	259.1000	28042465.0000
68	108/09/09	260.6000	28550660.8000
69	108/09/10	262.1000	29337335.0000
70	108/09/11	263.2000	31850991.4000
71	108/09/12	263.1000	29801104.4000
72	108/09/16	263.5000	32261994.6000
73	108/09/17	263.5000	34812396.8000
74	108/09/18	264.6000	39080575.4000
75	108/09/19	265.0000	37260485.8000
76	108/09/20	265.3000	38413200.2000
77	108/09/23	265.0000	33474448.6000
78	108/09/24	265.0000	32321468.4000
79	108/09/25	264.8000	26698149.2000
80	108/09/26	265.4000	28090656.4000
81	108/09/27	267.0000	28028046.8000
82	108/10/01	270.2000	40233783.2000
83	108/10/02	273.1000	41487385.2000
84	108/10/03	275.2000	44013525.4000
85	108/10/04	276.9000	44350827.4000
86	108/10/07	278.1000	39052310.0000
87	108/10/08	279.4000	31976242.2000
88	108/10/09	279.9000	32840742.8000

Problem3:

1.

```
# problem 3
diff_highest_price = df['最高價'].diff().abs().max()
diff_highest_price_date = df['最高價'].diff().abs().idxmax()
diff_daily_trade = df['成交股數'].diff().abs().max()
diff_daily_trade_date = df['成交股數'].astype('float64').diff().abs().idxmax()
print(diff_highest_price, df['日期'][diff_highest_price_date])
print(diff_daily_trade, df['日期'][diff_daily_trade_date])
```

```
8.5 108/06/19
44675937 108/10/02
```

By absolute difference:

I use df.diff().abs().max()

	變化量	日期
Highest daily price:	8.5	108/06/19
Daily trade volume	44675937	108/10/02

2.

```
pct_diff_highest_price = df['最高價'].pct_change().abs().max()
pct_diff_highest_price_date = df['最高價'].pct_change().abs().idxmax()
pct_diff_daily_trade = df['成交股數'].pct_change().abs().max()
pct_diff_daily_trade_date = df['成交股數'].pct_change().abs().idxmax()
print(pct_diff_highest_price, df['日期'][pct_diff_highest_price_date])
print(pct_diff_daily_trade, df['日期'][pct_diff_daily_trade_date])
```

```
0.03609341825902335 108/06/19
1.8638870893879664 108/07/19
```

By percentage:

I use df.pct_change().abs().max()

	變化量	日期
Highest daily price:	0.03609	108/06/19
Daily trade volume	1.86388	108/07/19

Problem 4

Outlier

```
# problem 4 [成交股數] [關盤-開盤價] [最高-最低]
df_oc = [abs(df['收盤價'][i] - df['開盤價'][i]) for i in range(len(df))]
df_maxmin = [abs(df['最高價'][i] - df['最低價'][i]) for i in range(len(df))]
df['盤差'] = df_oc
df['高低差'] = df_maxmin
mn_vol = df['成交股數'].mean()
mn_oc = df['盤差'].mean()
mn_maxmin= df['高低差'].mean()
result = []
df2 = df[['成交股數','盤差','高低差']]
df2_normal = (df2 - df2.mean())/ (df2.std())
df2_normal
for i in range(len(df)):
    res = (df2_normal['成交股數'][i]**2 + (df2_normal["盤差"][i])**2 + (df2_normal['高低差'][i])**2)**0.5
    result.append((i,res))
result
sort_result = sorted(result,key=lambda x:x[1],reverse=True)
print(sort_result[:5])
index_result = [sort_result[i][0] for i in range(5)]
print(index_result)
df['日期'][index_result]

```

```
[(45, 6.524535953495741), (82, 5.524305584949438), (44, 3.1347667749170607), (19, 3.0359956059954873), (5, 2.9434312544114394)]
[45, 82, 44, 19, 5]

45    108/08/06
82    108/10/01
44    108/08/05
19    108/07/01
5     108/06/11
```

I take absolute values for features.

The outlier days are:

	108/08/06	108/10/01	108/08/05	108/07/01	108/06/11
values	6.5245	5.243	3.134	3.0359	2.9434

Problem5

By brute force, traverse all possible solution

```
# problem 5
# bruce force
ans = []

for i in range(len(df)-1):
    for j in range(i+1,len(df)):
        diff = df['收盤價'][j] - df['收盤價'][i]
        ans.append((df['日期'][i],df['日期'][j],diff))
final_ans = sorted(ans, key=lambda x:x[2], reverse=True)
final_ans[:5]
```

```
[('108/06/06', '108/10/08', 54.5),
 ('108/06/04', '108/10/08', 53.5),
 ('108/06/17', '108/10/08', 53.5),
 ('108/06/26', '108/10/08', 52.0),
 ('108/06/05', '108/10/08', 51.5)]
```