In [1]:
```
pip install apyori
```

```
Requirement already satisfied: apyori in /opt/conda/lib/python3.7/site-package
s (1.1.2)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
import pandas as pd
# load the bank transaction dataset
df = pd.read_csv('PatientRoute.csv')
# info and the first 10 transactions
print(df.info())
print(df.head(10))
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6714 entries, 0 to 6713
Data columns (total 6 columns):
patient_id    6714 non-null int64
global_num    3571 non-null float64
date          6714 non-null object
location      6714 non-null object
latitude      6714 non-null float64
longitude     6714 non-null float64
dtypes: float64(3), int64(1), object(2)
memory usage: 314.8+ KB
None
   patient_id  global_num        date             location   latitude  \
0  1000000001         2.0  22/01/2020  Gyeonggi-do_Gimpo-si  37.615246
1  1000000001         2.0  24/01/2020         Seoul_Jung-gu  37.567241
2  1000000002         5.0  25/01/2020     Seoul_Seongbuk-gu  37.592560
3  1000000002         5.0  26/01/2020     Seoul_Seongbuk-gu  37.591810
4  1000000002         5.0  26/01/2020    Seoul_Seongdong-gu  37.563992
5  1000000002         5.0  26/01/2020     Seoul_Seongbuk-gu  37.590330
6  1000000002         5.0  26/01/2020     Seoul_Seongbuk-gu  37.589590
7  1000000002         5.0  27/01/2020     Seoul_Seongbuk-gu  37.592057
8  1000000002         5.0  27/01/2020   Seoul_Dongdaemun-gu  37.566262
9  1000000002         5.0  28/01/2020     Seoul_Seongbuk-gu  37.591669

    longitude
0  126.715632
1  127.005659
2  127.017048
3  127.016822
4  127.029534
5  127.015221
6  127.009766
7  127.018898
8  127.065815
9  127.018420
```

In [3]:
```
df.location.value_counts()
```

Out[3]:
```
Chungcheongnam-do_Cheonan-si     470
Seoul_Gangnam-gu                 293
Gangwon-do_Wonju-si              275
Gyeongsangbuk-do_Yecheon-gun     257
Seoul_Guro-gu                    206
                                ...
Gyeonggi-do_Hwaseong-si            1
Jeollanam-do_Damyang-gun           1
Busan_Yeongdo-gu                   1
Gyeongsangnam-do_Namhae-gun        1
Gyeonggi-do_Gwangju-si             1
Name: location, Length: 174, dtype: int64
```

transaction - patient's route item - a single patient can have multiple rows in the dataset, and each of the row represents the location he/she visited.

In [4]:
```
df.location.unique()
```

Out[4]:
```
array(['Gyeonggi-do_Gimpo-si', 'Seoul_Jung-gu', 'Seoul_Seongbuk-gu',
       'Seoul_Seongdong-gu', 'Seoul_Dongdaemun-gu', 'Seoul_Jungnang-gu',
       'Seoul_Gangnam-gu', 'Seoul_Jongno-gu', 'Gyeonggi-do_Goyang-si',
       'Gyeonggi-do_Seongnam-si', 'Seoul_Songpa-gu', 'Incheon_Yeonsu-gu',
       'Seoul_Seodaemun-gu', 'Seoul_Mapo-gu', 'Gyeonggi-do_Uijeongbu-si',
       'Gyeonggi-do_Dongducheon-si', 'Incheon_Jung-gu',
       'Seoul_Eunpyeong-gu', 'Daegu_Nam-gu', 'Seoul_Seocho-gu',
       'Gyeonggi-do_Gwacheon-si', 'Seoul_Gwangjin-gu', 'Seoul_Guro-gu',
       'Daegu_Jung-gu', 'Seoul_Dobong-gu', 'Seoul_Dongjak-gu',
       'Seoul_Yongsan-gu', 'Seoul_Yeongdeungpo-gu',
       'Gyeonggi-do_Icheon-si', 'Gyeonggi-do_Suwon-si',
       'Gyeonggi-do_Pyeongtaek-si', 'Gyeonggi-do_Paju-si',
       'Gyeongsangbuk-do_Cheongdo-gun', 'Seoul_Geumcheon-gu',
       'Seoul_Gwanak-gu', 'Daejeon_Yuseong-gu', 'Seoul_Gangdong-gu',
       'Seoul_Nowon-gu', 'Seoul_Yangcheon-gu', 'Daegu_Dong-gu',
       'Seoul_Gangseo-gu', 'Daegu_Dalseo-gu',
       'Gyeongsangnam-do_Tongyeong-si', 'Incheon_Namdong-gu',
       'Daejeon_Seo-gu', 'Gyeonggi-do_Anyang-si', 'Gyeonggi-do_Gunpo-si',
       'Jeollanam-do_Suncheon-si', 'Seoul_Gangbuk-gu', 'Jeju-do_Jeju-si',
       'Gyeonggi-do_Yongin-si', 'Gyeonggi-do_Gwangju-si', 'Daegu_Buk-gu',
       'Gyeonggi-do_Namyangju-si', 'Gyeonggi-do_Ansan-si',
       'Gyeonggi-do_Bucheon-si', 'Gyeonggi-do_Hanam-si',
       'Incheon_Gyeyang-gu', 'Gyeonggi-do_Siheung-si',
       'Gyeonggi-do_Yangpyeong-gun', 'Chungcheongbuk-do_Cheongju-si',
       'Gyeonggi-do_Yeoncheon-gun', 'Jeollabuk-do_Gunsan-si',
       'Chungcheongbuk-do_Yeongdong-gun', 'Gyeongsangnam-do_Jinju-si',
       'Jeollanam-do_Gwangyang-si', 'Jeollabuk-do_Jeonju-si',
       'Gyeonggi-do_Gwangmyeong-si', 'Jeollanam-do_Muan-gun',
       'Chungcheongbuk-do_Chungju-si', 'Busan_Dongnae-gu',
       'Busan_Suyeong-gu', 'Busan_Haeundae-gu', 'Busan_Yeonje-gu',
       'Busan_Dong-gu', 'Busan_Busanjin-gu', 'Busan_Seo-gu',
       'Busan_Geumjeong-gu', 'Gwangju_Gwangsan-gu', 'Gwangju_Seo-gu',
       'Gwangju_Buk-gu', 'Busan_Buk-gu', 'Busan_Nam-gu', 'Busan_Saha-gu',
       'Daegu_Seo-gu', 'Busan_Jung-gu', 'Busan_Gangseo-gu',
       'Incheon_Seo-gu', 'Busan_Gijang-gun',
       'Gyeongsangbuk-do_Gyeongju-si', 'Gyeongsangnam-do_Yangsan-si',
       'Busan_Sasang-gu', 'Ulsan_Nam-gu', 'Daegu_Dalseong-gun',
       'Gyeongsangnam-do_Gimhae-si', 'Jeollanam-do_Gurye-gun',
       'Gyeonggi-do_Anseong-si', 'Gyeongsangnam-do_Haman-gun',
       'Daegu_Suseong-gu', 'Gwangju_Nam-gu', 'Jeollanam-do_Naju-si',
       'Gwangju_Dong-gu', 'Jeollanam-do_Damyang-gun',
       'Jeollanam-do_Goheung-gun', 'Jeollanam-do_Boseong-gun',
       'Jeollanam-do_Hwasun-gun', 'Incheon_Bupyeong-gu',
       'Incheon_Dong-gu', 'Incheon_Michuhol-gu',
       'Chungcheongbuk-do_Eumseong-gun', 'Gangwon-do_Gangneung-si',
       'Gangwon-do_Jeongseon-gun', 'Chungcheongnam-do_Seosan-si',
       'Chungcheongnam-do_Asan-si', 'Gangwon-do_Samcheok-si',
       'Daejeon_Dong-gu', 'Daejeon_Jung-gu', 'Ulsan_Jung-gu',
       'Ulsan_Ulju-gun', 'Ulsan_Buk-gu', 'Gyeongsangnam-do_Uiryeong-gun',
       'Gyeongsangbuk-do_Gyeongsan-si', 'Ulsan_Dong-gu',
       'Gyeongsangbuk-do_Gimcheon-si', 'Gyeonggi-do_Guri-si',
       'Gyeongsangnam-do_Sacheon-si', 'Gangwon-do_Wonju-si',
       'Gangwon-do_Donghae-si', 'Gyeonggi-do_Yeoju-si',
       'Gyeongsangnam-do_Changwon-si', 'Gyeongsangbuk-do_Chilgok-gun',
       'Gyeongsangbuk-do_Bonghwa-gun', 'Gangwon-do_Taebaek-si',
       'Chungcheongbuk-do_Jeungpyeong-gun',
       'Chungcheongbuk-do_Jincheon-gun', 'Chungcheongbuk-do_Goesan-gun',
       'Gyeonggi-do_Hwaseong-si', 'Gangwon-do_Sokcho-si',
       'Chungcheongnam-do_Gyeryong-si', 'Chungcheongnam-do_Cheonan-si',
       'Busan_Yeongdo-gu', 'Chungcheongnam-do_Taean-gun',
       'Chungcheongnam-do_Gongju-si', 'Chungcheongnam-do_Dangjin-si',
       'Gyeongsangbuk-do_Yecheon-gun', 'Chungcheongnam-do_Seocheon-gun',
       'Chungcheongnam-do_Nonsan-si', 'Jeollabuk-do_Iksan-si',
       'Jeollanam-do_Yeosu-si', 'Jeollanam-do_Gangjin-gun',
       'Jeollanam-do_Mokpo-si', 'Gyeongsangbuk-do_Uljin-gun',
       'Gyeongsangbuk-do_Yeongcheon-si', 'Gangwon-do_Pyeongchang-gun',
       'Gyeongsangbuk-do_Cheongsong-gun', 'Chungcheongnam-do_Boryeong-si',
       'Gyeongsangnam-do_Changnyeong-gun', 'Gyeongsangbuk-do_Pohang-si',
       'Gyeongsangbuk-do_Gumi-si', 'Gyeongsangbuk-do_Andong-si',
       'Gyeongsangnam-do_Geochang-gun', 'Gyeongsangbuk-do_Sangju-si',
       'Gyeongsangbuk-do_Yeongju-si', 'Chungcheongbuk-do_Danyang-gun',
       'Gyeongsangbuk-do_Mungyeong-si', 'Gangwon-do_Chuncheon-si',
       'Jeollanam-do_Jangheung-gun', 'Gyeongsangbuk-do_Goryeong-gun',
       'Gyeongsangnam-do_Hapcheon-gun', 'Gyeongsangnam-do_Namhae-gun',
       'Gyeongsangnam-do_Geoje-si', 'Gyeonggi-do_Osan-si',
       'Gyeongsangnam-do_Miryang-si', 'Gyeongsangnam-do_Goseong-gun'],
      dtype=object)
```

In [5]:

```
df.patient_id.value_counts()
```

Out[5]:

```
1000000417    45
3009000014    42
1400000021    38
3009000013    37
6016000012    36
              ..
6100000067     1
6100000079     1
6100000045     1
6100000029     1
1100000017     1
Name: patient_id, Length: 1211, dtype: int64
```

In [6]:

```
print(df.patient_id.unique())
```

```
[1000000001 1000000002 1000000003 ... 6100000088 6100000089 6100000090]
```

As we are looking to generate association rulesfrom locations visited by each patient, we need to group the patients and then generate list of places visited

In [7]:

```
# group by account, then list all services
transactions = df.groupby(['patient_id'])['location'].apply(list)
print(transactions.head(5))
```

```
patient_id
1000000001                [Gyeonggi-do_Gimpo-si, Seoul_Jung-gu]
1000000002    [Seoul_Seongbuk-gu, Seoul_Seongbuk-gu, Seoul_S...
1000000003                   [Seoul_Jongno-gu, Seoul_Jongno-gu]
1000000004                                  [Seoul_Jungnang-gu]
1000000005                                  [Seoul_Jungnang-gu]
Name: location, dtype: object
```

Run the apriori model with min_support of 0.05

In [8]:

```
from apyori import apriori

transaction_list = list(transactions)
results = list(apriori(transaction_list, min_support=0.05))

def convert_apriori_results_to_pandas_df(results):
    rules = []

    for rule_set in results:
        for rule in rule_set.ordered_statistics:
            # items_base = left side of rules, items_add = right side
            # support, confidence and lift for respective rules
            rules.append([','.join(rule.items_base), ','.join(rule.items_add)
                            rule_set.support, rule.confidence, rule.lift])

    # typecast it to pandas df
    return pd.DataFrame(rules, columns=['Left_side', 'Right_side', 'Support',


result_df = convert_apriori_results_to_pandas_df(results)
# sort all acquired rules descending by lift
result_df = result_df.sort_values(by='Lift', ascending=False)
print(result_df.head(10))
```

```
  Left_side                    Right_side   Support  Confidence  Lift
0                         Busan_Yeonje-gu  0.060281    0.060281   1.0
1           Chungcheongnam-do_Cheonan-si  0.078448    0.078448   1.0
2                           Daegu_Jung-gu  0.061932    0.061932   1.0
3                         Incheon_Jung-gu  0.123865    0.123865   1.0
4                       Seoul_Dongjak-gu  0.085879    0.085879   1.0
5                       Seoul_Gangnam-gu  0.086705    0.086705   1.0
6                          Seoul_Guro-gu  0.066887    0.066887   1.0
7                          Seoul_Jung-gu  0.057803    0.057803   1.0
8                      Seoul_Jungnang-gu  0.075970    0.075970   1.0
9                     Seoul_Yangcheon-gu  0.052023    0.052023   1.0
```

With min_support 0.02

In [9]:

```
from apyori import apriori

# type cast the transactions from pandas into normal list format and run apri
transaction_list = list(transactions)
results = list(apriori(transaction_list, min_support=0.02))

result_df = convert_apriori_results_to_pandas_df(results)
result_df = result_df.sort_values(by='Lift', ascending=False)
print(result_df.head(10))
```

```
           Left_side          Right_side   Support  Confidence      Lift
37   Seoul_Songpa-gu     Incheon_Jung-gu  0.020644    0.454545  3.669697
36   Incheon_Jung-gu     Seoul_Songpa-gu  0.020644    0.166667  3.669697
34  Seoul_Gangnam-gu     Incheon_Jung-gu  0.030553    0.352381  2.844889
33   Incheon_Jung-gu    Seoul_Gangnam-gu  0.030553    0.246667  2.844889
27                     Seoul_Seodaemun-gu  0.022296    0.022296  1.000000
21                        Seoul_Gwanak-gu  0.034682    0.034682  1.000000
22                        Seoul_Jongno-gu  0.034682    0.034682  1.000000
23                          Seoul_Jung-gu  0.057803    0.057803  1.000000
24                      Seoul_Jungnang-gu  0.075970    0.075970  1.000000
25                          Seoul_Mapo-gu  0.025599    0.025599  1.000000
```

With min_support 0.003

In [10]:

```
from apyori import apriori

# type cast the transactions from pandas into normal list format and run apri
transaction_list = list(transactions)
results = list(apriori(transaction_list, min_support=0.003))

result_df = convert_apriori_results_to_pandas_df(results)
result_df = result_df.sort_values(by='Lift', ascending=False)
print(result_df.head(10))
```

```
                                              Left_side  \
567                                  Incheon_Yeonsu-gu
568  Chungcheongnam-do_Cheonan-si,Gyeongsangbuk-do_...
569     Incheon_Yeonsu-gu,Chungcheongnam-do_Cheonan-si
```

```
566                   Gyeongsangbuk-do_Andong-si
281                 Gyeongsangnam-do_Hapcheon-gun
282                 Gyeongsangnam-do_Uiryeong-gun
257                              Gwangju_Dong-gu
258                               Gwangju_Nam-gu
255                              Gwangju_Dong-gu
254                               Gwangju_Buk-gu

                                           Right_side   Support  Confidence  \
567  Chungcheongnam-do_Cheonan-si,Gyeongsangbuk-do_...  0.003303    0.333333
568                                   Incheon_Yeonsu-gu  0.003303    1.000000
569                          Gyeongsangbuk-do_Andong-si  0.003303    1.000000
566     Incheon_Yeonsu-gu,Chungcheongnam-do_Cheonan-si  0.003303    0.266667
281                       Gyeongsangnam-do_Uiryeong-gun  0.003303    0.800000
282                       Gyeongsangnam-do_Hapcheon-gun  0.003303    0.307692
257                                      Gwangju_Nam-gu  0.005780    0.636364
258                                     Gwangju_Dong-gu  0.005780    0.538462
255                                      Gwangju_Buk-gu  0.003303    0.363636
254                                     Gwangju_Dong-gu  0.003303    0.500000

           Lift
567  100.916667
568  100.916667
569   80.733333
566   80.733333
281   74.523077
282   74.523077
257   59.279720
258   59.279720
255   55.045455
254   55.045455
```

In [11]:

```python
result_df.loc[result_df['Left_side'] == 'Daegu_Buk-gu']
```

Out[11]:

	Left_side	Right_side	Support	Confidence	Lift
579	Daegu_Buk-gu	Daegu_Jung-gu,Daegu_Nam-gu	0.004129	0.128205	17.250712
572	Daegu_Buk-gu	Daegu_Jung-gu,Daegu_Dong-gu	0.003303	0.102564	11.291375
206	Daegu_Buk-gu	Daegu_Nam-gu	0.004129	0.128205	10.350427
212	Daegu_Buk-gu	Gyeongsangbuk-do_Pohang-si	0.009083	0.282051	8.758054
203	Daegu_Buk-gu	Daegu_Jung-gu	0.015690	0.487179	7.866325
200	Daegu_Buk-gu	Daegu_Dong-gu	0.003303	0.102564	5.175214
209	Daegu_Buk-gu	Gyeongsangbuk-do_Gyeongsan-si	0.003303	0.102564	3.268556

As we would like to find out 10 routes for positive patients who visted Daegu_Buk-gu, setting up min_support with 0.003 can't meet the requirement, we than reduce the min_support further.

With min_support 0.002

In [20]:

```python
from apyori import apriori

# type cast the transactions from pandas into normal list format and run apri
transaction_list = list(transactions)
results = list(apriori(transaction_list, min_support=0.002))

result_df = convert_apriori_results_to_pandas_df(results)
result_df = result_df.sort_values(by='Lift', ascending=False)
print(result_df)
```

```
                                 Left_side  \
1283  Daegu_Jung-gu,Daegu_Nam-gu,Daegu_Seo-gu
1269              Daegu_Buk-gu,Daegu_Seo-gu
1275             Seoul_Jung-gu,Daegu_Nam-gu
1270             Seoul_Jung-gu,Daegu_Buk-gu
1222              Daegu_Buk-gu,Daegu_Seo-gu
...                                     ...
648                        Seoul_Gangnam-gu
690                           Seoul_Guro-gu
691                       Seoul_Jungnang-gu
348                           Daegu_Jung-gu
349                        Seoul_Gangnam-gu

                                 Right_side   Support  Confidence  \
1283             Seoul_Jung-gu,Daegu_Buk-gu  0.002477    1.000000
1269  Seoul_Jung-gu,Daegu_Jung-gu,Daegu_Nam-gu  0.002477    1.000000
1275   Daegu_Jung-gu,Daegu_Buk-gu,Daegu_Seo-gu  0.002477    1.000000
1270   Daegu_Jung-gu,Daegu_Nam-gu,Daegu_Seo-gu  0.002477    1.000000
1222             Seoul_Jung-gu,Daegu_Nam-gu  0.002477    1.000000
...                                     ...       ...         ...
648                           Seoul_Jung-gu  0.002477    0.028571
690                       Seoul_Jungnang-gu  0.002477    0.037037
691                           Seoul_Guro-gu  0.002477    0.032609
348                        Seoul_Gangnam-gu  0.002477    0.040000
349                           Daegu_Jung-gu  0.002477    0.028571

            Lift
1283  403.666667
1269  403.666667
1275  403.666667
1270  403.666667
1222  403.666667
...          ...
648     0.494286
690     0.487520
691     0.487520
348     0.461333
349     0.461333

[1292 rows x 5 columns]
```

In [18]:

```
print(result_df.head(5))
```

```
                                 Left_side  \
1283  Daegu_Jung-gu,Daegu_Nam-gu,Daegu_Seo-gu
1269              Daegu_Buk-gu,Daegu_Seo-gu
1275             Seoul_Jung-gu,Daegu_Nam-gu
1270             Seoul_Jung-gu,Daegu_Buk-gu
1222              Daegu_Buk-gu,Daegu_Seo-gu

                                 Right_side   Support  Confidence  \
1283             Seoul_Jung-gu,Daegu_Buk-gu  0.002477         1.0
1269  Seoul_Jung-gu,Daegu_Jung-gu,Daegu_Nam-gu  0.002477         1.0
1275   Daegu_Jung-gu,Daegu_Buk-gu,Daegu_Seo-gu  0.002477         1.0
1270   Daegu_Jung-gu,Daegu_Nam-gu,Daegu_Seo-gu  0.002477         1.0
1222             Seoul_Jung-gu,Daegu_Nam-gu  0.002477         1.0

            Lift
1283  403.666667
1269  403.666667
1275  403.666667
1270  403.666667
1222  403.666667
```

Top 5 frequently occuring rules are as follows:

1. Daegu_Jung-gu,Daegu_Seo-gu,Daegu_Nam-gu ==> Daegu_Buk-gu,Seoul_Jung-gu
2. Daegu_Buk-gu,Daegu_Seo-gu ==> Daegu_Jung-gu,Daegu_Nam-gu,Seoul_Jung-gu
3. Daegu_Nam-gu,Seoul_Jung-gu ==> Daegu_Buk-gu,Daegu_Jung-gu,Daegu_Seo-gu
4. Daegu_Buk-gu,Seoul_Jung-gu ==> Daegu_Jung-gu,Daegu_Seo-gu,Daegu_Nam-gu
5. Daegu_Buk-gu,Daegu_Seo-gu ==> Daegu_Nam-gu,Seoul_Jung-gu

In [16]:
```
daegu = result_df.loc[result_df['Left_side'] == 'Daegu_Buk-gu']
daegu.head(10)
```

Out[16]:

	Left_side	Right_side	Support	Confidence	Lift
968	Daegu_Buk-gu	Seoul_Jung-gu,Daegu_Seo-gu	0.002477	0.076923	31.051282
961	Daegu_Buk-gu	Seoul_Jung-gu,Daegu_Nam-gu	0.002477	0.076923	31.051282
1202	Daegu_Buk-gu	Daegu_Jung-gu,Seoul_Jung-gu,Daegu_Seo-gu	0.002477	0.076923	31.051282
1217	Daegu_Buk-gu	Seoul_Jung-gu,Daegu_Nam-gu,Daegu_Seo-gu	0.002477	0.076923	31.051282
1262	Daegu_Buk-gu	Seoul_Jung-gu,Daegu_Jung-gu,Daegu_Nam-gu,Daegu...	0.002477	0.076923	31.051282
1172	Daegu_Buk-gu	Daegu_Jung-gu,Daegu_Nam-gu,Daegu_Seo-gu	0.002477	0.076923	31.051282
1187	Daegu_Buk-gu	Daegu_Jung-gu,Daegu_Nam-gu,Seoul_Jung-gu	0.002477	0.076923	31.051282
954	Daegu_Buk-gu	Daegu_Nam-gu,Daegu_Seo-gu	0.002477	0.076923	23.288462
940	Daegu_Buk-gu	Daegu_Jung-gu,Daegu_Seo-gu	0.002477	0.076923	18.630769
933	Daegu_Buk-gu	Daegu_Jung-gu,Daegu_Nam-gu	0.004129	0.128205	17.250712

We are not able to get atleast 10 routes from "Daegu_Buk-gu". So, we set min_support=0.002.

As it can be observed, we will be able to get atleast 10 common routes from "Daegu_Buk-gu" with min_support=0.002.