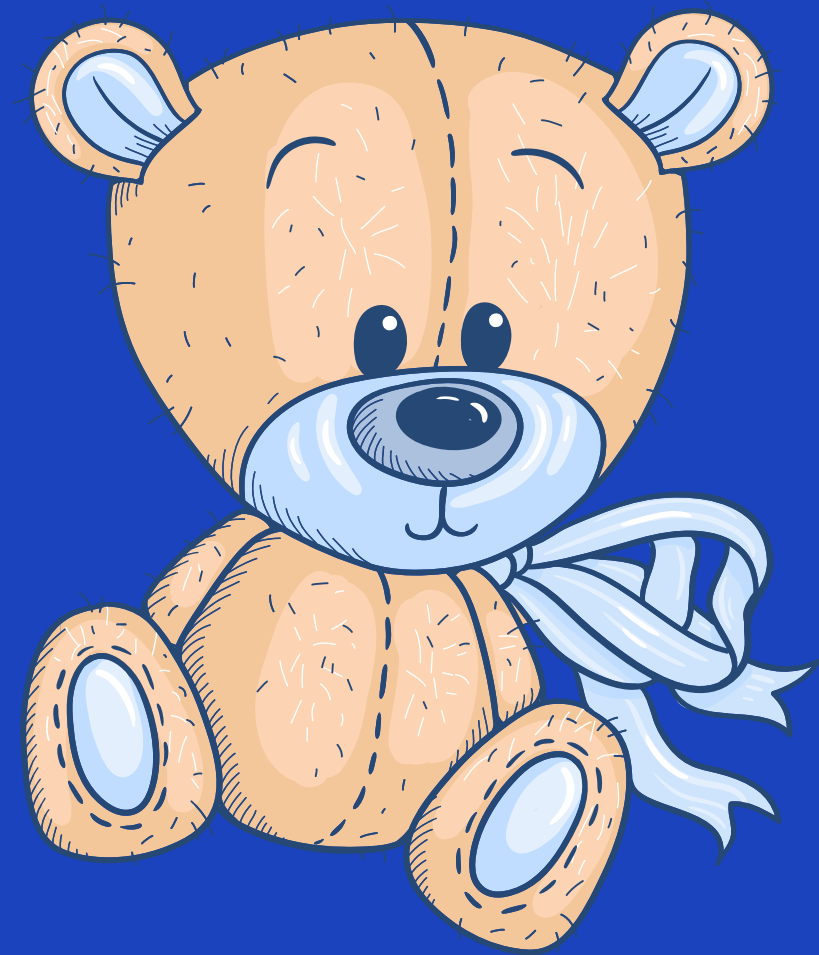


กลุ่มลูกหมี



01

PROJECT_DWDM

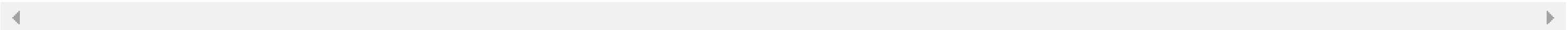
ข้อมูลชุดที่ 1 จำนวนผู้ลงทะเบียนใช้บริการอินเทอร์เน็ตความเร็วสูง

In [4]: `data_subscriber = pd.read_csv(os.path.join(path, '10broadband-subscribers.csv'))` *#จำนวนผู้ลงทะเบียนใช้บริการอินเทอร์เน็ตความเร็วสูงเก็บไว้ในตัวแปร data_subscriber*
`data_subscriber`

Out[4]:

	no.	quarter	Thailand Internet Users_subscriber	year	value_subscriber
0	4	4	Total Internet subscribers	2003	652726.00
1	3	3	Total Internet subscribers	2003	529530.00
2	2	2	Total Internet subscribers	2003	453043.00
3	1	1	Total Internet subscribers	2003	365219.00
4	8	4	Total Internet subscribers	2004	1231344.00
...
68	72	4	Total Internet subscribers	2020	11478264.92
69	71	3	Total Internet subscribers	2020	11282645.58
70	70	2	Total Internet subscribers	2020	10912996.14
71	69	1	Total Internet subscribers	2020	10264995.67
72	73	1	Total Internet subscribers	2021	11876158.44

73 rows × 5 columns



ข้อมูลชุดที่ 2 อัตราการเข้าถึงของบริการอินเทอร์เน็ต ความเร็วสูงต่อจำนวนครัวเรือน

03

In [6]: `data_household = pd.read_csv(os.path.join(path,'12broadband-penetration-per-household-.csv'))` #อัตราการเข้าถึงของบริการอินเทอร์เน็ตความเร็วสูงต่อจำนวนครัวเรือนเก็บไว้
ในตัวแปร `data_household`
`data_household`

Out[6]:

	no.	type_household	quarter	year	value_household
0	4	per household (%)	4	2003	0.000649
1	3	per household (%)	3	2003	0.000459
2	2	per household (%)	2	2003	0.000343
3	1	per household (%)	1	2003	0.000281
4	8	per household (%)	4	2004	0.008716
...
68	72	per household (%)	4	2020	0.514683
69	71	per household (%)	3	2020	0.515873
70	70	per household (%)	2	2020	0.498972
71	69	per household (%)	1	2020	0.469344
72	73	per household (%)	1	2021	0.532524

73 rows × 5 columns

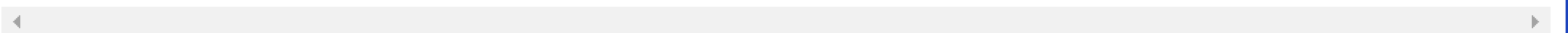
ข้อมูลชุดที่ 3 อัตราการเข้าถึงของบริการอินเทอร์เน็ตความเร็วสูง ต่อจำนวนประชากร

In [8]: `data_population = pd.read_csv(os.path.join(path, '11broadband-penetration-per-population-.csv'))` #อัตราการเข้าถึงของบริการอินเทอร์เน็ตความเร็วสูงต่อจำนวนประชากร ชื่อ `data_population`
`data_population`

Out[8]:

	no.	type_population	quarter	year	value_population
0	1	per population (%)	1	2003	0.0001
1	2	per population (%)	2	2003	0.0001
2	3	per population (%)	3	2003	0.0001
3	4	per population (%)	4	2003	0.0002
4	5	per population (%)	1	2004	0.0005
...
68	69	per population (%)	1	2020	0.1510
69	70	per population (%)	2	2020	0.1605
70	71	per population (%)	3	2020	0.1659
71	72	per population (%)	4	2020	0.1685
72	73	per population (%)	1	2021	0.1740

73 rows x 5 columns



ข้อมูลชุดที่ 4 จำนวนเลขหมายโทรศัพท์เคลื่อนที่ ที่ได้รับการจดทะเบียน

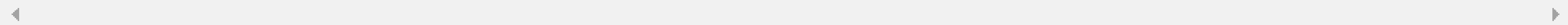
05

```
[16]: data_mobile = pd.read_csv(os.path.join(path,'05mobile-subscribers.csv')) #จำนวนเลขหมายโทรศัพท์เคลื่อนที่ที่ได้รับการจดทะเบียน เก็บในตัวแปรชื่อdata_mobile
data_mobile
```

[16]:

	no.	Mobile Market Report	quarter	year	value_mobile
0	4	Mobile Subscribers	4	2002	17449890
1	3	Mobile Subscribers	3	2002	15743776
2	2	Mobile Subscribers	2	2002	12416261
3	1	Mobile Subscribers	1	2002	9669909
4	8	Mobile Subscribers	4	2003	21616910
...
72	76	Mobile Subscribers	4	2020	116294420
73	75	Mobile Subscribers	3	2020	119169759
74	74	Mobile Subscribers	2	2020	131881072
75	73	Mobile Subscribers	1	2020	132594791
76	77	Mobile Subscribers	1	2021	117562201

77 rows × 5 columns



ปัญหาและวัตถุประสงค์

ปัญหา

เราไม่สามารถพยากรณ์ข้อมูลปี
ต่อๆไปได้ เนื่องจากยังไม่ทราบ
ความสัมพันธ์ของข้อมูลปัจจุบัน
ที่มีอยู่

วัตถุประสงค์

เพื่อศึกษาความสัมพันธ์ของ
ตารางที่ 1 ถึง 4 ว่าความ
สัมพันธ์ที่ได้จากการคาดเดาปี
ต่อๆไปเป็นอย่างไร?

Preprocessing

07

```
In [20]: merge_table3 = data_mobile3.merge(merge_table2,how='left', left_on='no.',right_on='no.') # เชื่อมตาราง merge_table2 และ data_mobile โดยให้ตาราง merge_table2 เป็นตารางหลักอยู่ทางด้านซ้าย
merge_table3
```

Out[20]:

	no.	Mobile Market Report	value_mobile	quarter	Thailand Internet Users_subscriber	year	value_subscriber	type_population	value_population	type_household	value_household
0	5	Mobile Subscribers	18763102	1.0	Total Internet subscribers	2004.0	785931.00	per population (%)	0.0005	per household (%)	0.001556
1	6	Mobile Subscribers	19788956	2.0	Total Internet subscribers	2004.0	914400.00	per population (%)	0.0007	per household (%)	0.002265
2	7	Mobile Subscribers	20655866	3.0	Total Internet subscribers	2004.0	1091652.00	per population (%)	0.0015	per household (%)	0.004964
3	8	Mobile Subscribers	21616910	4.0	Total Internet subscribers	2004.0	1231344.00	per population (%)	0.0026	per household (%)	0.008716
4	9	Mobile Subscribers	23217311	1.0	Total Internet subscribers	2005.0	1197942.00	per population (%)	0.0045	per household (%)	0.015449
...
68	73	Mobile Subscribers	132594791	1.0	Total Internet subscribers	2021.0	11876158.44	per population (%)	0.1740	per household (%)	0.532524
69	74	Mobile Subscribers	131881072	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
70	75	Mobile Subscribers	119169759	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
71	76	Mobile Subscribers	116294420	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
72	77	Mobile Subscribers	117562201	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

73 rows × 11 columns

ต่อตารางทั้ง 4 ตารางเข้าด้วยกันด้วยคำสั่ง merge

จากนั้นทำการตรวจสอบค่า Missing

```
[23] 1 # เช็ค missing ตาราง merge_table3  
      2 merge_table3.isnull().any()
```

no.	False
Mobile Market Report	False
value_mobile	False
quarter	True
Thailand Internet Users_subscriber	True
year	True
value_subscriber	True
type_population	True
value_population	True
type_household	True
value_household	True
dtype: bool	

พบว่า มีค่า missing อยู่

In [22]: `merge_table3_drop = merge_table3.dropna()` #จัดการกับ Missing ด้วยการ dropna
`merge_table3_drop`

Out[22]:

	no.	Mobile Market Report	value_mobile	quarter	Thailand Internet Users_subscriber	year	value_subscriber	type_population	value_population	type_household	value_household
0	5	Mobile Subscribers	18763102	1.0	Total Internet subscribers	2004.0	785931.00	per population (%)	0.0005	per household (%)	0.001556
1	6	Mobile Subscribers	19788956	2.0	Total Internet subscribers	2004.0	914400.00	per population (%)	0.0007	per household (%)	0.002265
2	7	Mobile Subscribers	20655866	3.0	Total Internet subscribers	2004.0	1091652.00	per population (%)	0.0015	per household (%)	0.004964
3	8	Mobile Subscribers	21616910	4.0	Total Internet subscribers	2004.0	1231344.00	per population (%)	0.0026	per household (%)	0.008716
4	9	Mobile Subscribers	23217311	1.0	Total Internet subscribers	2005.0	1197942.00	per population (%)	0.0045	per household (%)	0.015449
...
64	69	Mobile Subscribers	122173569	1.0	Total Internet subscribers	2020.0	10264995.67	per population (%)	0.1510	per household (%)	0.469344
65	70	Mobile Subscribers	124366678	2.0	Total Internet subscribers	2020.0	10912996.14	per population (%)	0.1605	per household (%)	0.498972
66	71	Mobile Subscribers	126137528	3.0	Total Internet subscribers	2020.0	11282645.58	per population (%)	0.1659	per household (%)	0.515873
67	72	Mobile Subscribers	129613743	4.0	Total Internet subscribers	2020.0	11478264.92	per population (%)	0.1685	per household (%)	0.514683
68	73	Mobile Subscribers	132594791	1.0	Total Internet subscribers	2021.0	11876158.44	per population (%)	0.1740	per household (%)	0.532524

69 rows × 11 columns



จัดการ Missing ด้วยการ dropna

จากนั้นทำการตรวจสอบค่า Missing ไม่พบค่าที่ missing แล้ว

```
[25] 1 # เช็ค missing ตาราง merge_table3_drop
      2 merge_table3_drop.isnull().any()
```

no.	False
Mobile Market Report	False
value_mobile	False
quarter	False
Thailand Internet Users_subscriber	False
year	False
value_subscriber	False
type_population	False
value_population	False
type_household	False
value_household	False
dtype: bool	

ต่อมาทำการตรวจสอบว่า การลบ
missing ด้วยการ `.dropna()`
จะเสียข้อมูลกี่เปอร์เซ็นต์

```
[26] 1 # จากการทำ dropna() ทำให้ข้อมูลหายไปกี่ %
      2 removed = merge_table3.shape[0] - merge_table3_drop.shape[0]
      3
      4 print(f'size before drop = {merge_table3.shape[0]}')
      5 print(f'size aftre drop = {merge_table3_drop.shape[0]}')
      6 print(f'we loss {100*(removed/merge_table3.shape[0])}% of data')
```

```
size before drop = 73
size aftre drop = 69
we loss 5.47945205479452% of data
```

Classification

11

แก้ไขข้อมูลโดยกำหนดค่าในตาราง เพื่อให้ข้อมูลในตารางนำไปใช้ในการ Assosiation ได้ โดยการเพิ่ม column ใหม่ในตาราง merge_table3_drop และ column ที่สร้างใหม่นั้นจะตามหลังด้วย _group โดยการจัดกลุ่มให้ค่าทั้งหมดในแต่ละ column ซึ่งจัดกลุ่มดังนี้

ทำการ import package

```
In [27]: from pandas.api.types import CategoricalDtype
```

ดูชื่อ column ในตาราง merge_table3_drop

```
In [28]: merge_table3_drop.columns
```

```
Out[28]: Index(['no.', 'Mobile Market Report', 'value_mobile', 'quarter',  
               'Thailand Internet Users_subscriber', 'year', 'value_subscriber',  
               'type_population', 'value_population', 'type_household',  
               'value_household'],  
              dtype='object')
```

จากนั้นเพิ่ม column ใหม่ในตาราง merge_table3_drop
โดยกำหนดให้ column ชื่อว่า subscriber_group

- ถ้ามีค่าตั้งแต่ 0 - 2000000 จะให้เป็น low_subscriber
- ถ้ามีค่าตั้งแต่ 2000000 - 5000000 จะให้เป็น median_subscriber
- ถ้ามีค่าตั้งแต่ 5000000 - 12000000 จะให้เป็น high_subscriber

```
In [56]: bucket = ['low_subscriber','median_subscriber','high_subscriber']
merge_table3_drop['subscriber_group'] = pd.cut(merge_table3_drop.value_subscriber,
                                             bins=[0,2000000,5000000,12000000],
                                             labels=bucket,
                                             right=False).astype(str).astype(CategoricalDtype(bucket,ordered=True))
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: SettingWithCopyWarning:
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: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
.....

column ชื่อว่า household_group

- ถ้ามีค่าตั้งแต่ 0 - 0.1 จะให้เป็น low_household
- ถ้ามีค่าตั้งแต่ 0.1 - 0.3 จะให้เป็น median_household
- ถ้ามีค่าตั้งแต่ 0.3 - 0.6 จะให้เป็น high_household

```
In [30]: bucket = ['low','median','high']
merge_table3_drop['household_group'] = pd.cut(merge_table3_drop.value_household,
                                             bins=[0,0.1,0.3,0.6],
                                             labels=bucket,
                                             right=False).astype(str).astype(CategoricalDtype(bucket,ordered=True))
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: SettingWithCopyWarning:
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: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

column ชื่อว่า `population_group`

- ถ้ามีค่าตั้งแต่ 0 - 0.05 จะให้เป็น `low_population`
- ถ้ามีค่าตั้งแต่ 0.05 - 0.10 จะให้เป็น `median_population`
- ถ้ามีค่าตั้งแต่ 0.10 - 0.18 จะให้เป็น `high_population`

```
In [45]: bucket = ['low_population','median_population','high_population']
merge_table3_drop['population_group'] = pd.cut(merge_table3_drop.value_population,
        bins=[0,0.05,0.10,0.18],
        labels=bucket,
        right=False).astype(str).astype(CategoricalDtype(bucket,ordered=True))
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: SettingWithCopyWarning:
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: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
#####

column ชื่อว่า mobile_group

- ถ้ามีค่าตั้งแต่ 0 - 0.05 จะให้เป็น low_mobile
- ถ้ามีค่าตั้งแต่ 0.05 - 0.10 จะให้เป็น median_mobile
- ถ้ามีค่าตั้งแต่ 0.10 - 0.18 จะให้เป็น high_mobile

```
In [47]: bucket = ['low_mobile','median_mobile','high_mobile']
merge_table3_drop['mobile_group'] = pd.cut(merge_table3_drop.value_mobile,
                                           bins=[0,50000000,100000000,140000000],
                                           labels=bucket,
                                           right=False).astype(str).astype(CategoricalDtype(bucket,ordered=True))
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: SettingWithCopyWarning:
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: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
.....

In [58]: `merge_table3_drop.columns`

Out[58]: Index(['no.', 'Mobile Market Report', 'value_mobile', 'quarter',
 'Thailand Internet Users_subscriber', 'year', 'value_subscriber',
 'type_population', 'value_population', 'type_household',
 'value_household', 'subscriber_group', 'household_group',
 'population_group', 'mobile_group'],
 dtype='object')

In [59]: `data_cut = merge_table3_drop[['subscriber_group', 'household_group', 'population_group', 'mobile_group']]`
`data_cut`

Out[59]:

	subscriber_group	household_group	population_group	mobile_group
0	low_subscriber	low_household	low_popution	low_mobile
1	low_subscriber	low_household	low_popution	low_mobile
2	low_subscriber	low_household	low_popution	low_mobile
3	low_subscriber	low_household	low_popution	low_mobile
4	low_subscriber	low_household	low_popution	low_mobile
...
64	high_subscriber	high_household	high_population	high_mobile
65	high_subscriber	high_household	high_population	high_mobile
66	high_subscriber	high_household	high_population	high_mobile
67	high_subscriber	high_household	high_population	high_mobile
68	high_subscriber	high_household	high_population	high_mobile

69 rows × 4 columns

Assosiation

17

In [52]: `!pip install apyori`

Requirement already satisfied: apyori in /usr/local/lib/python3.7/dist-packages (1.1.2)

In [53]: `from apyori import apriori`

In [59]: `transacs = [] ## กำหนด transacs ให้เป็น list ว่าง
for i in range(0,len(data_cut)):
 transacs.append([str(data_cut.values[i,j]) for j in range(0,4)])`

In [63]: `Association = list(apriori(transacs,min_support = 0.40, min_confidence = 0.40))
Association`

Out[63]: [RelationRecord(items=frozenset({'high_subscriber'}), support=0.42028985507246375, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'high_subscriber'}), confidence=0.42028985507246375, lift=1.0)]),
RelationRecord(items=frozenset({'low_population'}), support=0.4057971014492754, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'low_population'}), confidence=0.4057971014492754, lift=1.0)]),
RelationRecord(items=frozenset({'median_household'}), support=0.43478260869565216, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'median_household'}), confidence=0.43478260869565216, lift=1.0)]),
RelationRecord(items=frozenset({'median_mobile'}), support=0.4492753623188406, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'median_mobile'}), confidence=0.4492753623188406, lift=1.0)]),
RelationRecord(items=frozenset({'median_mobile', 'median_household'}), support=0.43478260869565216, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'median_mobile', 'median_household'}), confidence=0.43478260869565216, lift=1.0), OrderedStatistic(items_base=frozenset({'median_household'}), items_add=frozenset({'median_mobile'}), confidence=1.0, lift=2.225806451612903), OrderedStatistic(items_base=frozenset({'median_mobile'}), items_add=frozenset({'median_household'}), confidence=0.967741935483871, lift=2.2258064516129035)])]

In [41]: `# กำหนดข้อมูลให้เป็นค่าตัวเลข โดยข้อมูลที่ยกขึ้นด้วย low=0 , median=1 , high=2`
`data_cut['subscriber_group'] = data_cut['subscriber_group'].map({'low_subscriber':0,'median_subscriber':1,'high_subscriber':2})`
`data_cut['household_group'] = data_cut['household_group'].map({'low_household':0,'median_household':1,'high_household':2})`
`data_cut['population_group'] = data_cut['population_group'].map({'low_population':0,'median_population':1,'high_population':2})`
`data_cut['mobile_group'] = data_cut['mobile_group'].map({'low_mobile':0,'median_mobile':1,'high_mobile':2})`
`data_cut`

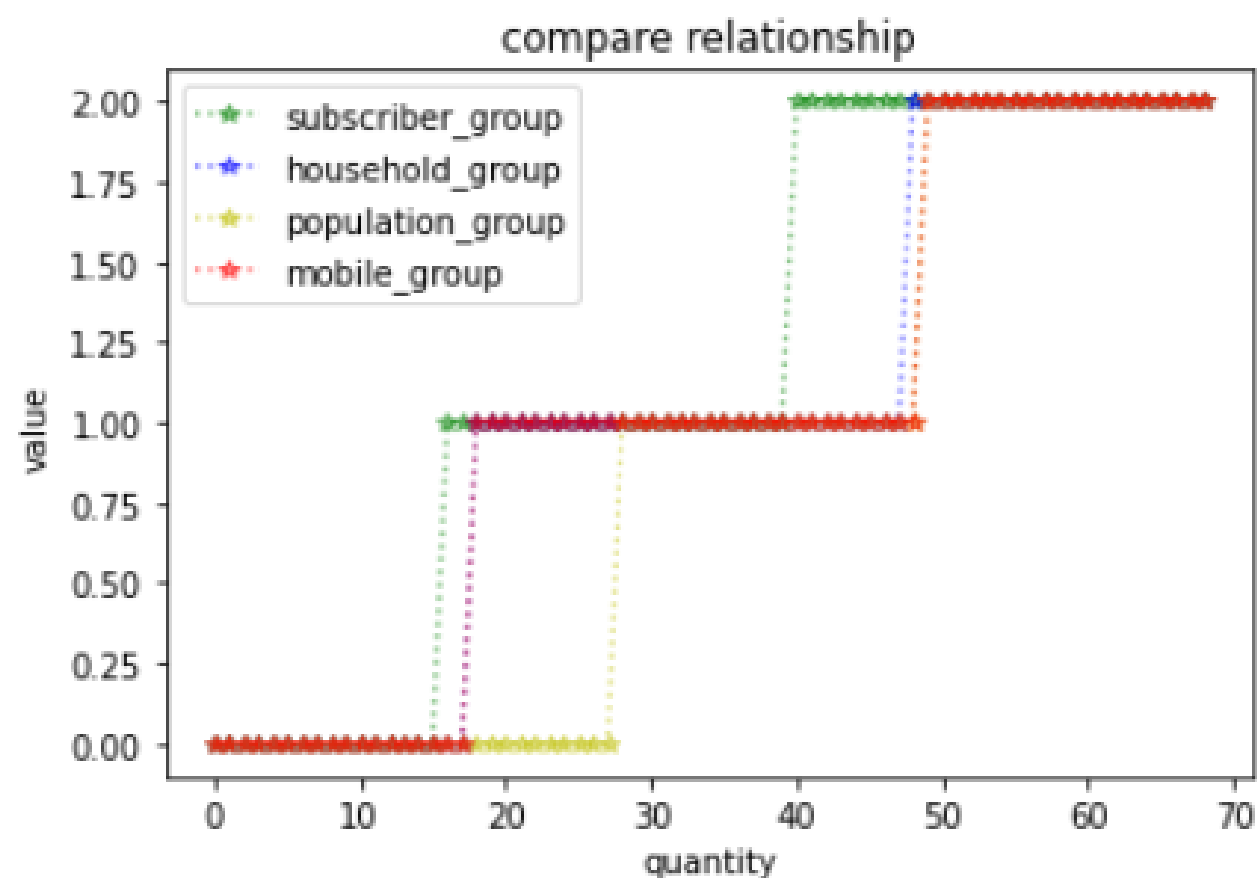
Out[41]:

	subscriber_group	household_group	population_group	mobile_group
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
...
64	2	2	2	2
65	2	2	2	2
66	2	2	2	2
67	2	2	2	2
68	2	2	2	2

69 rows x 4 columns

```
[48] 1 from matplotlib import pyplot as plt

[49] 1 plt.plot(range(len(data_cut.iloc[:, :-3])), data_cut.iloc[:, :-3], '*:g', alpha=0.5, label='subscriber_group')
      2 plt.plot(range(len(data_cut.iloc[:, 1:-2])), data_cut.iloc[:, 1:-2], '*:b', alpha=0.5, label='household_group')
      3 plt.plot(range(len(data_cut.iloc[:, 2:-1])), data_cut.iloc[:, 2:-1], '*:y', alpha=0.5, label='population_group')
      4 plt.plot(range(len(data_cut.iloc[:, 3:])), data_cut.iloc[:, 3:], '*:r', alpha=0.5, label='mobile_group')
      5
      6 plt.ylabel('value')
      7 plt.xlabel('quantity')
      8 plt.title('compare relationship')
      9 plt.legend();
```



จากกราฟค่าข้อมูลจัดอยู่ในกลุ่มเดียวกัน
ข้อมูลมีความสัมพันธ์ในตัวเอง
แต่ไม่มีความสัมพันธ์กันกับข้อมูลชุดอื่น

รายชื่อสมาชิก (กลุ่มลูกหมี)

นางสาวมินตรา ทิพย์รัตน์สุนทร

623020041-2

นางสาวกัลยารัตน์ แสนสมบัติ

623020513-7

นางสาวฐิติชญา ไกรวงศ์

623020520-0

นางสาวนันทิชา วิชิต

623020526-8

นางสาวศศิกานต์ บุญมี

623020539-9

