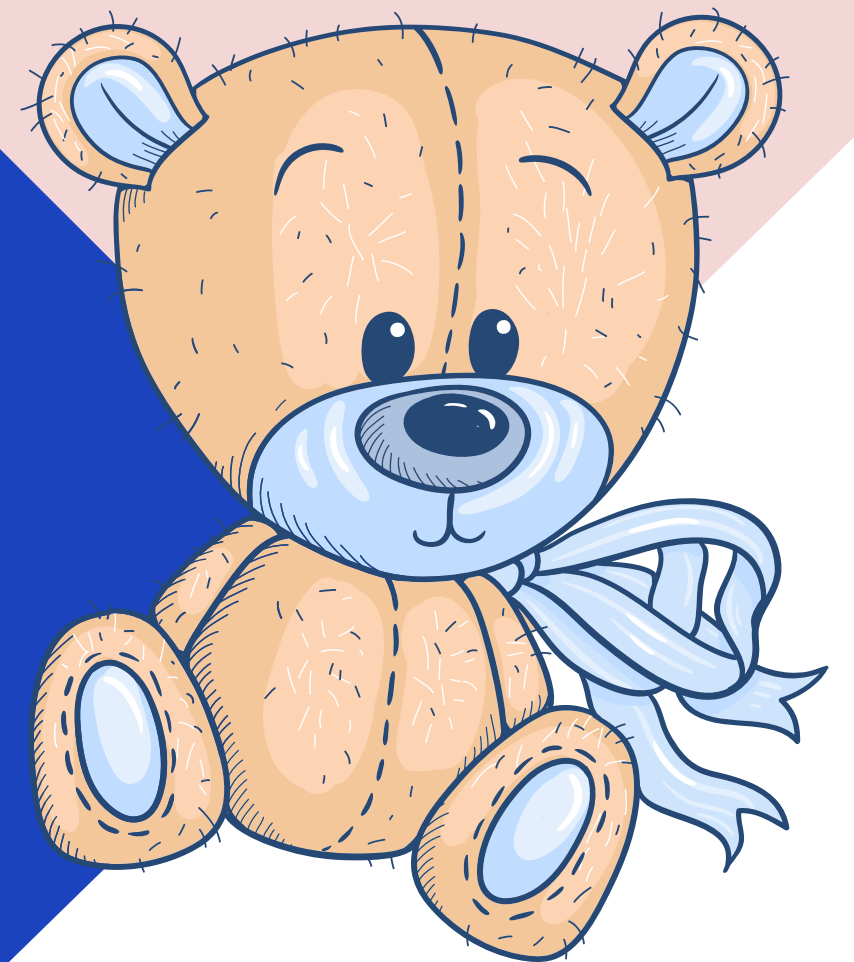


กลุ่มลูกหมี

01

PROJECT\_DWDM



# ข้อมูลชุดที่ 1

## จำนวนผู้ลงทะเบียนใช้บริการอินเทอร์เน็ตความเร็วสูง

```
In [5]: data_subscriber = pd.read_csv(os.path.join(path, '10broadband-subscribers.csv'))
data_subscriber
```

```
Out[5]:
```

	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

ที่มา : [https://data.go.th/dataset/broadband-subscribers?](https://data.go.th/dataset/broadband-subscribers?fbclid=IwAR2hJLTNGiEnMNwRApUdbzzaFUj-1_Ag7HWgdZPxX-0aEKw2PBU0_W3-_m0)

[fbclid=IwAR2hJLTNGiEnMNwRApUdbzzaFUj-1\\_Ag7HWgdZPxX-0aEKw2PBU0\\_W3-\\_m0](https://data.go.th/dataset/broadband-subscribers?fbclid=IwAR2hJLTNGiEnMNwRApUdbzzaFUj-1_Ag7HWgdZPxX-0aEKw2PBU0_W3-_m0)

องค์กร : สำนักงานคณะกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมแห่งชาติ

# ข้อมูลชุดที่ 2

## อัตราการเข้าถึงของบริการอินเทอร์เน็ตความเร็วสูงต่อจำนวนครัวเรือน

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

ที่มา : [https://data.go.th/dataset/broadband-penetration-per-household?](https://data.go.th/dataset/broadband-penetration-per-household?fbclid=IwAR0tNweDa7pmiAaygo0Z88iFFIYLtQRFjujfalyEpUub_bFVQEkCcbXAZJc)

[fbclid=IwAR0tNweDa7pmiAaygo0Z88iFFIYLtQRFjujfalyEpUub\\_bFVQEkCcbXAZJc](https://data.go.th/dataset/broadband-penetration-per-household?fbclid=IwAR0tNweDa7pmiAaygo0Z88iFFIYLtQRFjujfalyEpUub_bFVQEkCcbXAZJc)

องค์กร : สำนักงานคณะกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมแห่งชาติ

# ข้อมูลชุดที่ 3

## อัตราการเข้าถึงของบริการอินเทอร์เน็ตความเร็วสูงต่อจำนวนประชากร

```
In [7]: data_population = pd.read_csv(os.path.join(path, '11broadband-penetration-per-population-.csv'))  
data_population
```

```
Out[7]:
```

	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 × 5 columns

ที่มา : [https://data.go.th/dataset/broadband-penetration-per-population?](https://data.go.th/dataset/broadband-penetration-per-population?fbclid=IwAR0D5Z4yxZOXJYPM9hgBJpAVbuWBkxwNzEdW_jWwlvvkCKQxaGERqrxffFs)

[fbclid=IwAR0D5Z4yxZOXJYPM9hgBJpAVbuWBkxwNzEdW\\_jWwlvvkCKQxaGERqrxffFs](https://data.go.th/dataset/broadband-penetration-per-population?fbclid=IwAR0D5Z4yxZOXJYPM9hgBJpAVbuWBkxwNzEdW_jWwlvvkCKQxaGERqrxffFs)

องค์กร : สำนักงานคณะกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมแห่งชาติ

# ข้อมูลชุดที่ 4

## จำนวนเลขหมายโทรศัพท์เคลื่อนที่ที่ได้รับการจดทะเบียน

```
In [17]: data_mobile = pd.read_csv(os.path.join(path, '05mobile-subscribers.csv'))  
data_mobile
```

```
Out[17]:
```

	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

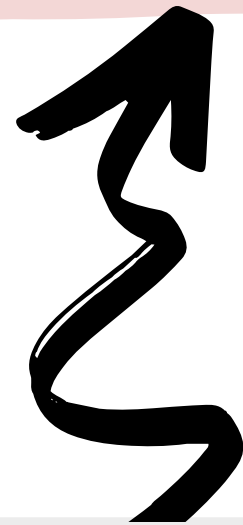
77 rows × 5 columns

ที่มา : [https://data.go.th/dataset/mobile-subscribers?](https://data.go.th/dataset/mobile-subscribers?fbclid=IwAR3ZoY2gqT_ogfZwHHI40nC_ejcmi3O9xZM0mcac_rCCdA5brXcHVh85Ayw)

[fbclid=IwAR3ZoY2gqT\\_ogfZwHHI40nC\\_ejcmi3O9xZM0mcac\\_rCCdA5brXcHVh85Ayw](https://data.go.th/dataset/mobile-subscribers?fbclid=IwAR3ZoY2gqT_ogfZwHHI40nC_ejcmi3O9xZM0mcac_rCCdA5brXcHVh85Ayw)

องค์กร : สำนักงานคณะกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมแห่งชาติ

## ปัญหา



เราไม่สามารถพยากรณ์ข้อมูล  
ปีต่อๆ ไปได้ เนื่องจากยังไม่  
ทราบความสัมพันธ์ของข้อมูล  
ปัจจุบันที่มีอยู่

เพื่อศึกษาความสัมพันธ์ของ  
ตารางที่ 1 ถึง 4 ว่าความ  
สัมพันธ์ที่ได้ไปเป็นอย่างไร?



## วัตถุประสงค์



In [22]: `# เชื่อมตาราง merge_table2 และ data_mobile เข้าด้วยกัน โดยให้ตาราง merge_table2 เป็นตารางหลักอยู่ที่ด้านซ้าย  
merge_table3 = data_mobile3.merge(merge_table2,how='left',left_on='no.',right_on='no.')  
merge_table3`

Out[22]:

	no.	Mobile Market Report	value_mobile	quarter	Thailand Internet Users_subscriber	year	value_subscriber	type_population	value_population	type_household	value_househol
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

```
In [23]: # เช็ค missing ตาราง merge_table3  
merge_table3.isnull().any()
```

```
Out[23]: 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
```

พบว่ามีค่า missing อยู่



# จัดการ Missing ด้วยการ dropna

```
In [24]: # จัดการ missing โดยการ dropna()
merge_table3_drop = merge_table3.dropna()
merge_table3_drop
```

Out[24]:

	no.	Mobile Market Report	value_mobile	quarter	Thailand Internet Users_subscriber	year	value_subscriber	type_population	value_population	type_household	value_househol
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

## จากนั้นทำการตรวจสอบค่า Missing ไม่พบค่าที่ missing แล้ว

```
In [25]: # เช็ค missing ตาราง merge_table3_drop
merge_table3_drop.isnull().any()
```

```
Out[25]: 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() ว่าจะเสียข้อมูลไปที่เปอร์เซ็นต์

```
In [26]: # จากการทำ dropna() ทำให้ข้อมูลหายไปกี่ %
removed = merge_table3.shape[0] - merge_table3_drop.shape[0]

print(f'size before drop = {merge_table3.shape[0]}')
print(f'size aftre drop = {merge_table3_drop.shape[0]}')
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 ที่สร้างใหม่นั้น จะตามหลังด้วย \_group โดยการจัดกลุ่มให้ค่าทั้งหมดในแต่ละ column ซึ่งจัดกลุ่มดังนี้

**ทำการ import package**

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

**เพื่อดูชื่อ 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 [29]: bucket = ['low_subscriber', 'median_subscriber', 'high_subscriber'] # กำหนดกลุ่มข้อมูล
merge_table3_drop['subscriber_group'] = pd.cut(merge_table3_drop.value_subscriber, # เลือก column ที่ต้องการแปลงข้อมูล
        bins=[0,2000000,5000000,12000000], # กำหนดช่วงของข้อมูล
        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-versu
s-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_household', 'median_household', 'high_household'] # กำหนดกลุ่มข้อมูล
merge_table3_drop['household_group'] = pd.cut(merge_table3_drop.value_household, # เลือก column ที่ต้องการแปลงข้อมูล
                                             bins=[0,0.1,0.3,0.6], # กำหนดช่วงของข้อมูล
                                             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-versu
s-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 [31]: bucket = ['low_population', 'median_population', 'high_population'] # กำหนดกลุ่มข้อมูล
merge_table3_drop['population_group'] = pd.cut(merge_table3_drop.value_population, # เลือก column ที่ต้องการแปลงข้อมูล
                                             bins=[0,0.05,0.10,0.18], # กำหนดช่วงของข้อมูล
                                             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-versu
s-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 [32]: bucket = ['low_mobile', 'median_mobile', 'high_mobile'] # กำหนดกลุ่มข้อมูล
merge_table3_drop['mobile_group'] = pd.cut(merge_table3_drop.value_mobile, # เลือก column ที่ต้องการแปลงข้อมูล
      bins=[0,500000000,1000000000,1400000000], # กำหนดช่วงของข้อมูล
      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](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

"""

# ตัด column ที่เราต้องการ แล้วตั้งชื่อตารางใหม่ว่า data\_cut

```
In [35]: data_cut = merge_table3_drop[['subscriber_group', 'household_group', 'population_group', 'mobile_group']] # เลือ  
อก column ที่ต้องการนำไปใช้งาน  
data_cut
```

Out[35]:

	subscriber_group	household_group	population_group	mobile_group
0	low_subscriber	low_household	low_population	low_mobile
1	low_subscriber	low_household	low_population	low_mobile
2	low_subscriber	low_household	low_population	low_mobile
3	low_subscriber	low_household	low_population	low_mobile
4	low_subscriber	low_household	low_population	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

# Association

17

```
In [37]: transaction = [] # คือกำหนด transacs ให้เป็น list ว่าง
        for i in range(0,len(data_cut)):
            transaction.append([str(data_cut.values[i,j]) for j in range(0,4)])
```

```
In [38]: !pip install apyori # ทำการติดตั้งแพ็คเกจ apyori
```

```
Collecting apyori
  Downloading apyori-1.1.2.tar.gz (8.6 kB)
Building wheels for collected packages: apyori
  Building wheel for apyori (setup.py) ... done
  Created wheel for apyori: filename=apyori-1.1.2-py3-none-any.whl size=5974 sha256=6eb089ad0fd3314fc917bb69c9e3c9ba8efeb30cf7bbfd54879f16cacd502e
  Stored in directory: /root/.cache/pip/wheels/cb/f6/e1/57973c631d27efd1a2f375bd6a83b2a616c4021f24aab84080
Successfully built apyori
Installing collected packages: apyori
Successfully installed apyori-1.1.2
```

```
In [39]: from apyori import apriori # ทำการ import package เข้ามาใช้งาน
```

```
In [40]: Asso = list(apriori(transaction,min_support = 0.4, min_confidence = 0.4))
```

```
Asso
```

```
# min_support คือ ค่าสนับสนุนต่ำสุดที่ยอมรับได้ ต่ำกว่านี้จะไม่นำไปวิเคราะห์
```

```
# min_confidence คือ ค่าเชื่อมั่นต่ำสุดที่สนใจ
```

```
Out[40]: [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)))]
```

# สรุปผล Association

19

- 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)]),

**หมายความว่า** จำนวนผู้ลงทะเบียนใช้บริการอินเทอร์เน็ตความเร็วสูงอยู่ในระดับสูง มีค่าสนับสนุนต่ำสุดที่ยอมรับได้ 42.03 % ค่าเชื่อมั่นต่ำสุดที่สนใจ 42.03 % ค่า lift เท่ากับ 1

- 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)]),

**หมายความว่า** อัตราการเข้าถึงของบริการอินเทอร์เน็ตความเร็วสูงต่อจำนวนประชากรอยู่ในระดับต่ำ มีค่าสนับสนุนต่ำสุดที่ยอมรับได้ 40.58 % ค่าเชื่อมั่นต่ำสุดที่สนใจ 40.58 % ค่า lift เท่ากับ 1

- `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)])`,

**หมายความว่า** อัตราการเข้าถึงของบริการอินเทอร์เน็ตความเร็วสูงต่อจำนวนครัวเรือนอยู่ในระดับกลาง มีค่าสนับสนุนต่ำสุดที่ยอมรับได้ 43.48 % ค่าเชื่อมั่นต่ำสุดที่สนใจ 43.48 % ค่า lift เท่ากับ 1

- `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)])`,

**หมายความว่า** จำนวนเลขหมายโทรศัพท์เคลื่อนที่ที่ได้รับการจดทะเบียนอยู่ในระดับกลาง มีค่าสนับสนุนต่ำสุดที่ยอมรับได้ 44.93 % ค่าเชื่อมั่นต่ำสุดที่สนใจ 44.93 % ค่า lift เท่ากับ 1

จากกฎ 4 ข้อด้านบนจะสังเกตเห็นว่าค่า lift = 1 แสดงว่ากฎพวกนี้ยังไม่น่าสนใจและยังไม่น่าสนใจ



- `RelationRecord(items=frozenset({'median_household', 'median_mobile'}), support=0.43478260869565216, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'median_household', 'median_mobile'}), 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)])`

### แปลผลได้ว่า

- `-RelationRecord(items=frozenset({'median_mobile', 'median_household'}), support=0.43478260869565216`

**หมายความว่า** ค่าสนับสนุนต่ำสุดที่ยอมรับได้ของทั้ง 2 itemsets คือ อัตราการเข้าถึงของบริการอินเทอร์เน็ต ความเร็วสูงต่อจำนวนครัวเรือนอยู่ในระดับปานกลางและจำนวนเลขหมายโทรศัพท์เคลื่อนที่ที่ได้รับการจดทะเบียนอยู่ในระดับปานกลาง มีค่าสนับสนุนต่ำสุดที่ยอมรับได้ทั้งหมด 43.48 %

- OrderedStatistic(items\_base=frozenset({'median\_household'}),  
items\_add=frozenset({'median\_mobile'}), confidence=1.0, lift=2.225806451612903)

**หมายความว่า** มีการเข้าถึงบริการอินเทอร์เน็ตความเร็วสูงต่อครัวเรือนที่อยู่ในระดับกลาง จึงค่อยมีการจดทะเบียนหมายเลขโทรศัพท์เคลื่อนที่ที่อยู่ในระดับกลางมีค่าเชื่อมั่นต่ำสุดที่สนใจทั้งหมด 100 %

- OrderedStatistic(items\_base=frozenset({'median\_mobile'}),  
items\_add=frozenset({'median\_household'}), confidence=0.967741935483871,  
lift=2.2258064516129035)

**หมายความว่า** มีการจดทะเบียนหมายเลขโทรศัพท์เคลื่อนที่ที่อยู่ในระดับกลางก่อนจึงค่อยมีการเข้าถึงบริการอินเทอร์เน็ตความเร็วสูงต่อครัวเรือนที่อยู่ในระดับกลาง มีค่าเชื่อมั่นต่ำสุดที่สนใจทั้งหมด 96.77 %

- code : lift=2.2258064516129035

**หมายความว่า** ดัชนีที่น่าสนใจ มีความสัมพันธ์กันมาก

# แปลงข้อมูลให้เป็นตัวเลข

In [41]: `# กำหนดข้อมูลให้เป็นค่าตัวเลข โดยใช้อัตราที่เริ่มต้นด้วย 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]:

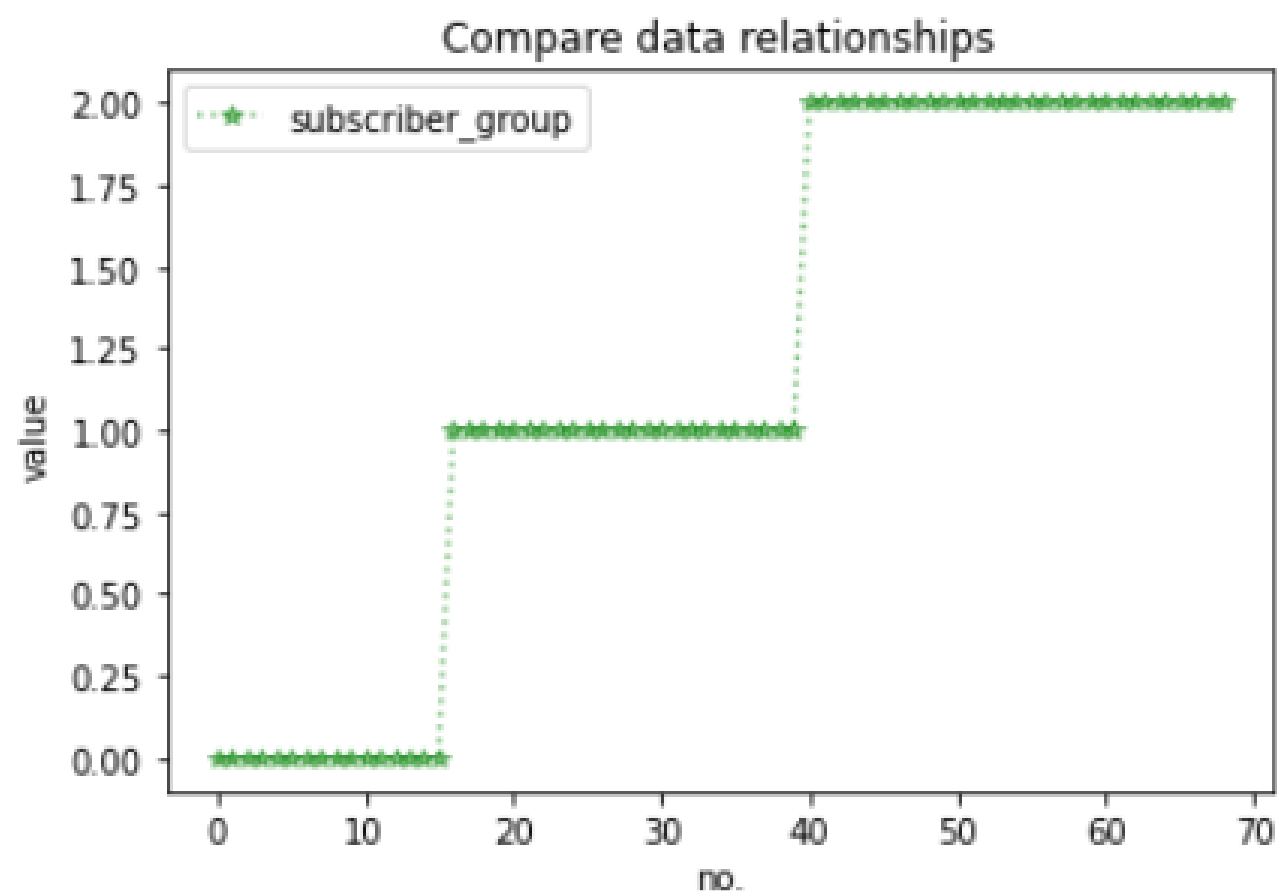
	subscriber_group	household_group	population_group	mobile_group
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
...	...	...	...	...
64	2	2	2	2
65	2	2	2	2
66	2	2	2	2
67	2	2	2	2
68	2	2	2	2

69 rows × 4 columns

กำหนดให้ข้อมูลที่เริ่มต้นด้วย  
**low = 0**  
**median = 1**  
**high = 2**

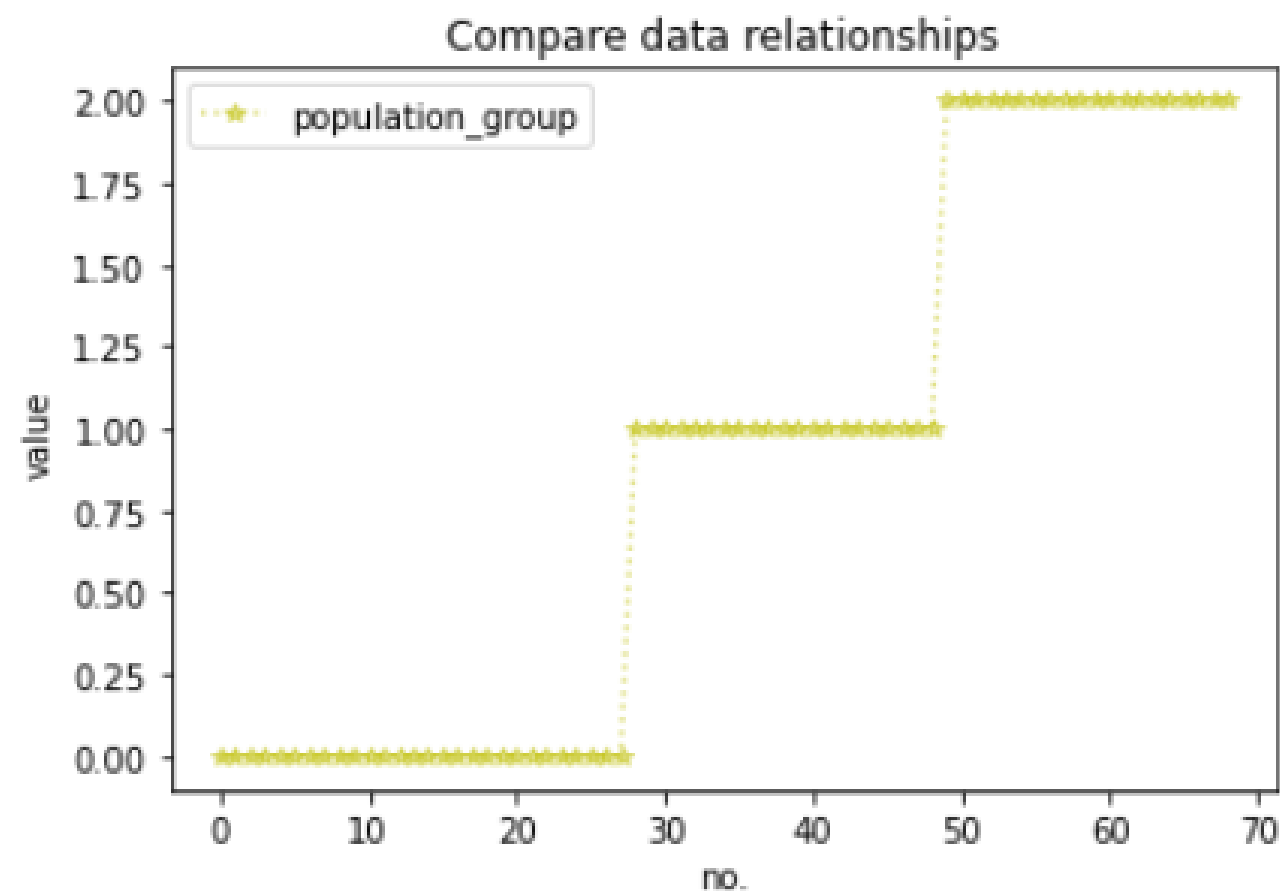
## ผล Asso code 1 มา plot กราฟ subscriber\_group โดย high ดูกราฟที่ข้อมูลอยู่ในกลุ่ม high หรือ 2

```
In [63]: plt.plot(range(len(data_cut.iloc[:, :-3])), data_cut.iloc[:, :-3], '*:g', alpha=0.5, label='subscriber_group')
plt.ylabel('value') # แกน y คือ value บอกว่าข้อมูลในแต่ละ column มีค่าข้อมูลเท่ากับเท่าไร โดยค่าข้อมูลมี 3 ตัวตามที่ได้กำหนดไว้ คือ 0,1,2
plt.xlabel('no.') # แกน x คือ บอกว่าข้อมูลในแต่ละ column อยู่จุดที่เท่าไร
plt.title('Compare data relationships') # ชื่อกราฟคือ Compare data relationships/เปรียบเทียบความสัมพันธ์ของข้อมูล
plt.legend();
```



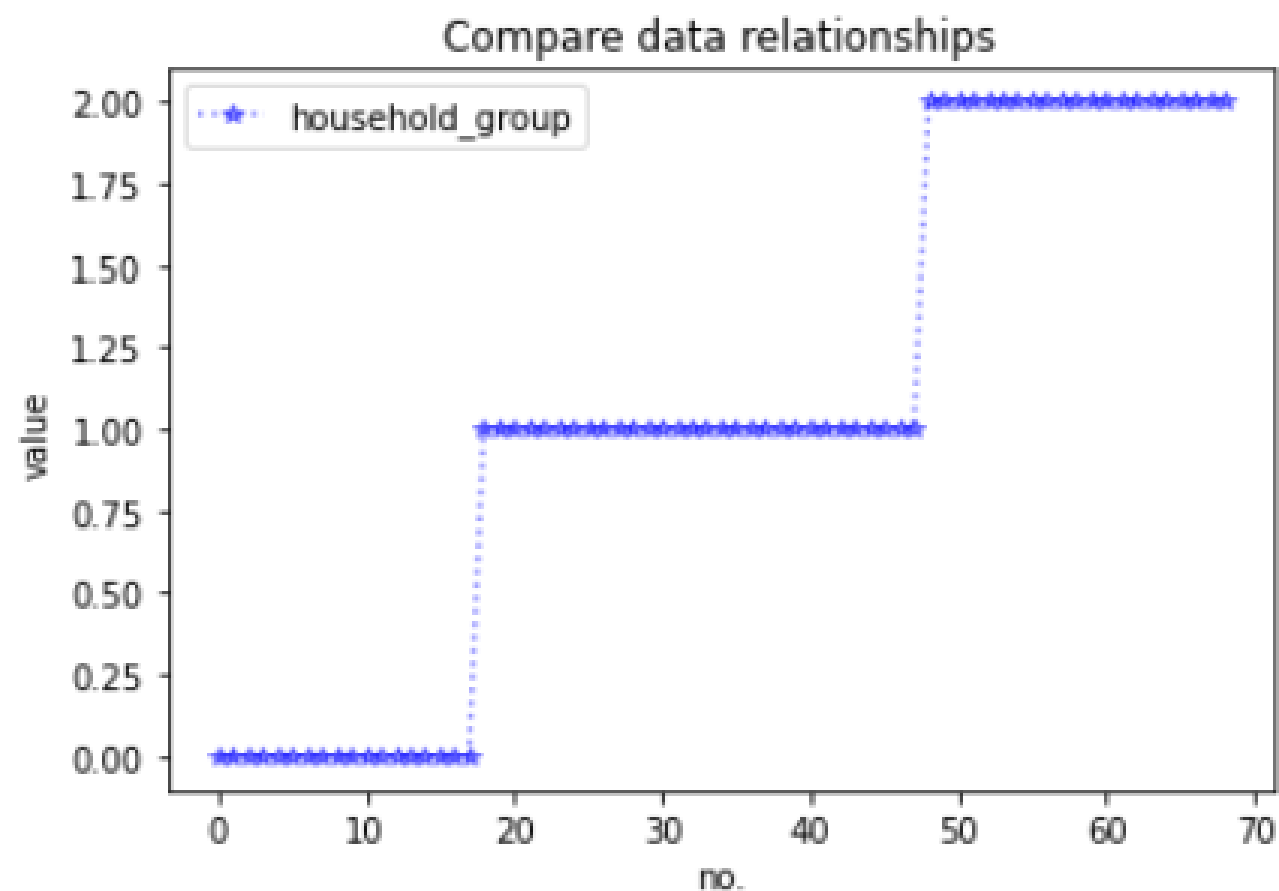
## ผล Asso code 2 มา plot กราฟ population\_group โดย low\_population ดูกราฟที่ข้อมูลอยู่ในกลุ่ม low หรือ 0

```
In [64]: plt.plot(range(len(data_cut.iloc[:,2:-1])),data_cut.iloc[:,2:-1],':y',alpha=0.5,label='population_group')
plt.ylabel('value') # แกน y คือ value บอกว่าข้อมูลในแต่ละ column มีค่าข้อมูลเท่ากับเท่าไร โดยค่าข้อมูลมี 3 ตัวตามที่ได้กำหนดไว้ คือ 0,1,2
plt.xlabel('no.') # แกน x คือ บอกว่าข้อมูลในแต่ละ column อยู่จุดที่เท่าไร
plt.title('Compare data relationships') # ชื่อกราฟคือ Compare data relationships/เปรียบเทียบความสัมพันธ์ของข้อมูล
plt.legend();
```



## ผล code 3 มา plot กราฟ household\_group โดย median\_household ดูกราฟที่ข้อมูลอยู่ในกลุ่ม median หรือ 1

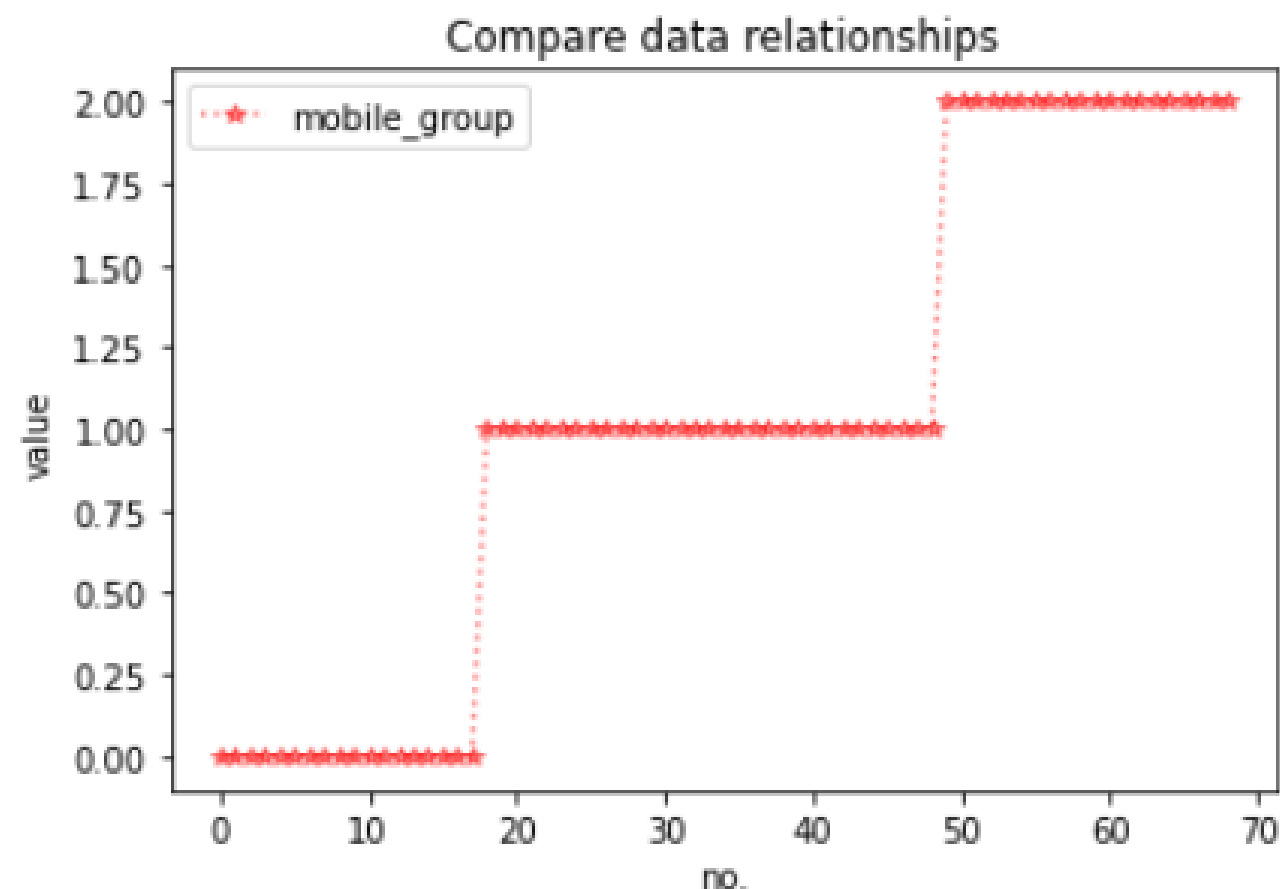
```
In [65]: plt.plot(range(len(data_cut.iloc[:,1:-2])),data_cut.iloc[:,1:-2],':b',alpha=0.5,label='household_group')
plt.ylabel('value') # แกน y คือ value บอกว่าข้อมูลในแต่ละ column มีค่าข้อมูลเท่ากับเท่าไร โดยค่าข้อมูลมี 3 ตัวตามที่ได้กำหนดไว้ คือ 0,1,2
plt.xlabel('no.') # แกน x คือ บอกว่าข้อมูลในแต่ละ column อยู่จุดที่เท่าไร
plt.title('Compare data relationships') # ชื่อกราฟคือ Compare data relationships/เปรียบเทียบความสัมพันธ์ของข้อมูล
plt.legend();
```





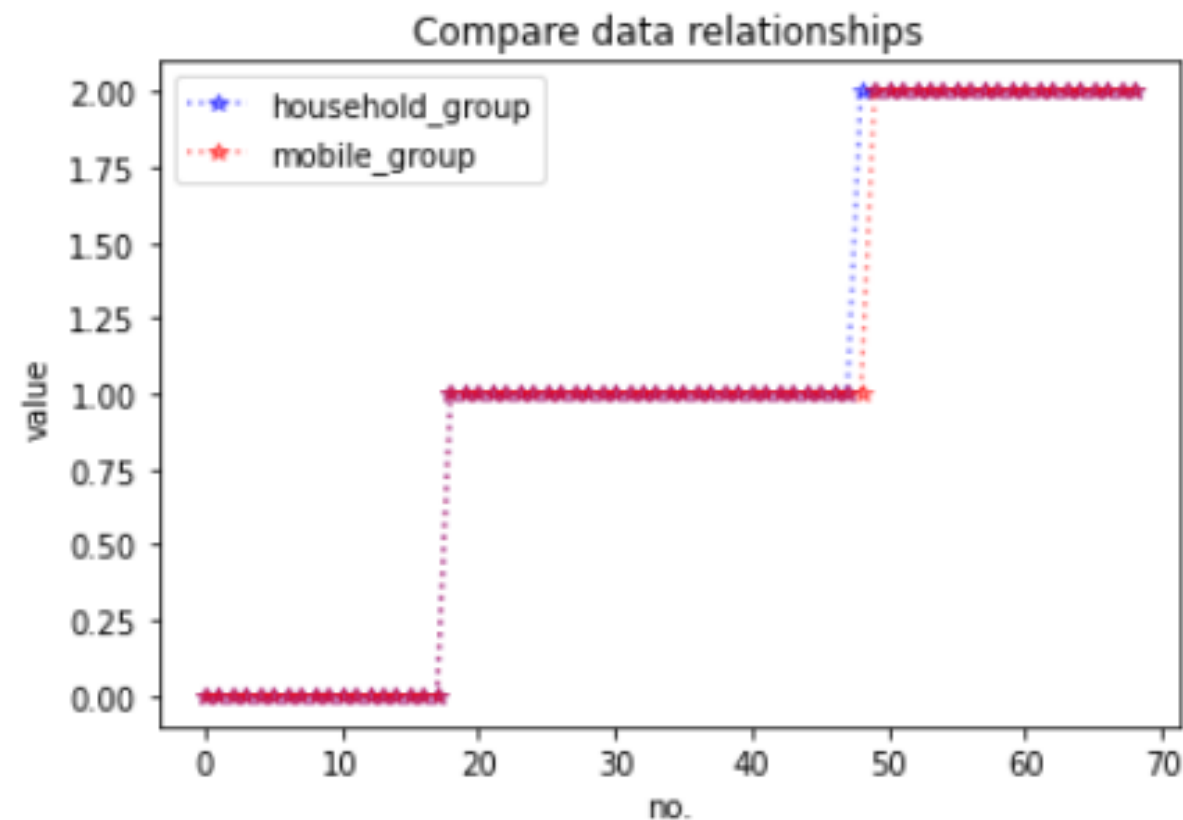
## ผล Asso code 4 มา plot กราฟ mobile\_group โดย median\_mobile ดูกราฟที่ข้อมูลอยู่ในกลุ่ม median หรือ 1

```
In [66]: plt.plot(range(len(data_cut.iloc[:,3:])),data_cut.iloc[:,3:],':r',alpha=0.5,label='mobile_group')
plt.ylabel('value') # แกน y คือ value บอกว่าข้อมูลในแต่ละ column มีค่าข้อมูลเท่ากับเท่าไร โดยค่าข้อมูลมี 3 ตัวตามที่ได้กำหนดไว้ คือ 0,1,2
plt.xlabel('no.') # แกน x คือ บอกว่าข้อมูลในแต่ละ column อยู่จุดที่เท่าไร
plt.title('Compare data relationships') # ชื่อกราฟคือ Compare data relationships/เปรียบเทียบความสัมพันธ์ของข้อมูล
plt.legend();
```



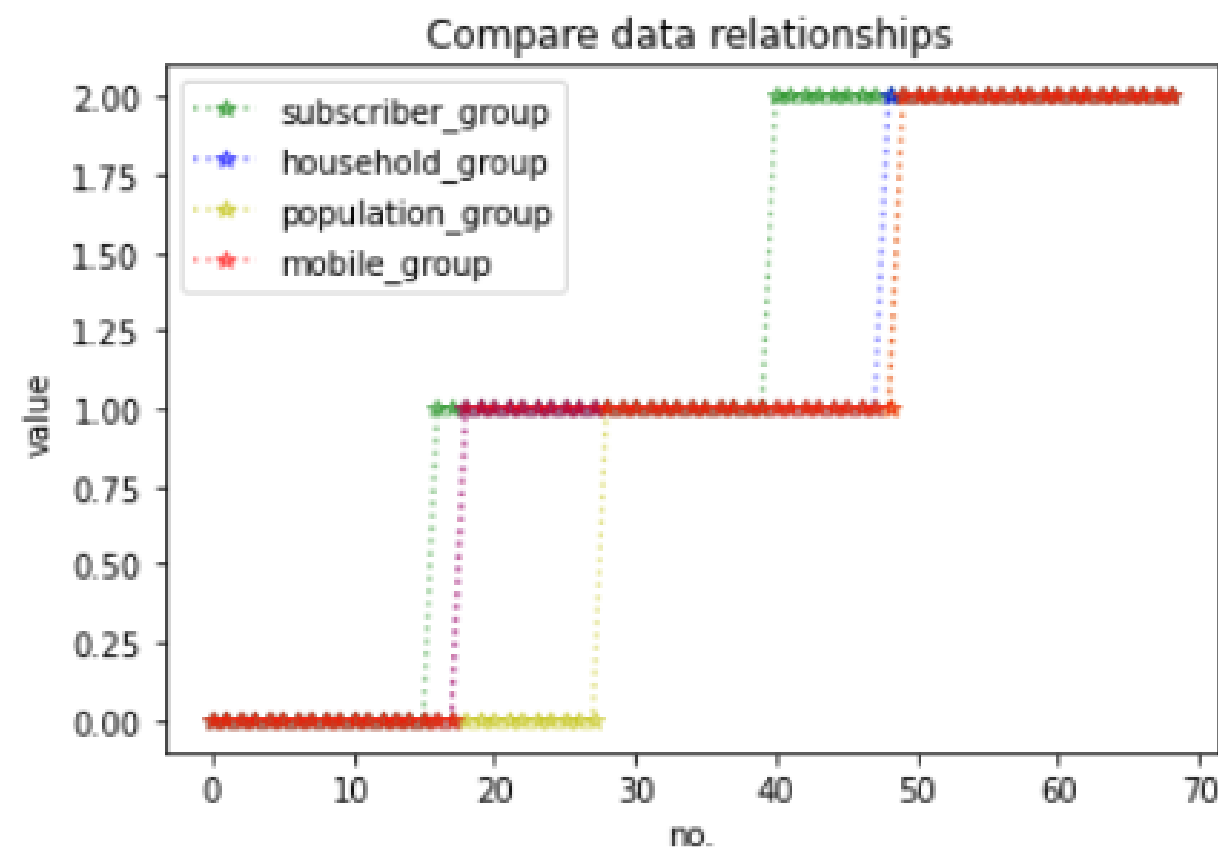
ผล Asso code 5 มา plot กราฟ household\_group กับ mobile\_group median\_household กับ median\_mobile ดูกราฟตรงที่ข้อมูล อยู่ในกลุ่ม median หรือ 1

```
In [67]: plt.plot(range(len(data_cut.iloc[:,1:-2])),data_cut.iloc[:,1:-2],':b',alpha=0.5,label='household_group') #household_group คือสี b=
สีน้ำเงิน
plt.plot(range(len(data_cut.iloc[:,3:])),data_cut.iloc[:,3:],':r',alpha=0.5,label='mobile_group') # mobile_group คือสี r= สีแดง
plt.ylabel('value') # แกน y คือ value บอกว่าข้อมูลในแต่ละ column มีค่าข้อมูลเท่ากับเท่าไร โดยค่าข้อมูลมี 3 ตัวตามที่ได้กำหนดไว้ คือ 0,1,2
plt.xlabel('no.') # แกน x คือ บอกว่าข้อมูลในแต่ละ column อยู่จุดที่เท่าไร
plt.title('Compare data relationships') # ชื่อกราฟคือ Compare data relationships/เปรียบเทียบความสัมพันธ์ของข้อมูล
plt.legend();
```



## เปรียบเทียบข้อมูลทุก column

```
In [68]: plt.plot(range(len(data_cut.iloc[:, :-3])), data_cut.iloc[:, :-3], '*:g', alpha=0.5, label='subscriber_group') # subscriber_group คือสี g = สีเขียว
plt.plot(range(len(data_cut.iloc[:, 1:-2])), data_cut.iloc[:, 1:-2], '*:b', alpha=0.5, label='household_group') # household_group คือสี b = สีน้ำเงิน
plt.plot(range(len(data_cut.iloc[:, 2:-1])), data_cut.iloc[:, 2:-1], '*:y', alpha=0.5, label='population_group') # population_group คือสี y = สีเหลือง
plt.plot(range(len(data_cut.iloc[:, 3:])), data_cut.iloc[:, 3:], '*:r', alpha=0.5, label='mobile_group') # mobile_group คือสี r = สีแดง
plt.ylabel('value') # แกน y คือ value บอกว่าข้อมูลในแต่ละ column มีค่าข้อมูลเท่ากับเท่าไร โดยค่าข้อมูลมี 3 ตัวตามที่ได้กำหนดไว้ คือ 0,1,2
plt.xlabel('no.') # แกน x คือ บอกว่าข้อมูลในแต่ละ column อยู่จุดที่เท่าไร
plt.title('Compare data relationships') # ชื่อกราฟคือ Compare data relationships/เปรียบเทียบความสัมพันธ์ของข้อมูล
plt.legend();
```



จากกราฟข้อมูลส่วนใหญ่ซ้อนทับกัน จึงสรุปได้ว่าข้อมูลมีความสัมพันธ์กัน

# ยกตัวอย่าง

จากกราฟจะสังเกตได้ว่า

ข้อมูล mobile\_group , subscriber\_group , household\_group และ population\_group  
จากจุดที่ no. 10 ก็จะจัดอยู่ในกลุ่มที่ 0 คือในระดับ low

In [50]:

```
# ตัวอย่าง : ข้อมูล mobile_group ที่ no.10 ก็จะจัดอยู่ในกลุ่มที่ 0 คือในระดับ Low
# อ่านกราฟได้ว่า "จุดที่ 10 สีแดง มีค่าข้อมูลเท่ากับ 0"
# จุดสีแดงคือข้อมูลจาก mobile_group

# ลอง code เพื่อเปรียบเทียบข้อมูลที่อยู่ในกราฟว่าตรงกันรึเปล่า?
data_cut.iloc[10:-58,3:]
```

Out[50]:

	mobile_group
10	0

In [52]:

```
# ตัวอย่าง : ข้อมูล household_group ที่ no.10 ก็จะจัดอยู่ในกลุ่มที่ 0 คือในระดับ Low
# อ่านกราฟได้ว่า "จุดที่ 10 สีน้ำเงิน มีค่าข้อมูลเท่ากับ 0"
# จุดสีน้ำเงินคือข้อมูลจาก household_group

# ลอง code เพื่อเปรียบเทียบข้อมูลที่อยู่ในกราฟว่าตรงกันรึเปล่า?
data_cut.iloc[10:-58,1:-2]
```

Out[52]:

	household_group
10	0

In [51]:

```
# ตัวอย่าง : ข้อมูล subscriber_group ที่ no.10 ก็จะจัดอยู่ในกลุ่มที่ 0 คือในระดับ Low
# อ่านกราฟได้ว่า "จุดที่ 10 สีเขียว มีค่าข้อมูลเท่ากับ 0"
# จุดสีเขียวคือข้อมูลจาก subscriber_group

# ลอง code เพื่อเปรียบเทียบข้อมูลที่อยู่ในกราฟว่าตรงกันรึเปล่า?
data_cut.iloc[10:-58,:-3]
```

Out[51]:

	subscriber_group
10	0

In [53]:

```
# ตัวอย่าง : ข้อมูล population_group ที่ no.10 ก็จะจัดอยู่ในกลุ่มที่ 0 คือในระดับ Low
# อ่านกราฟได้ว่า "จุดที่ 10 สีเหลือง มีค่าข้อมูลเท่ากับ 0"
# จุดสีเหลืองคือข้อมูลจาก population_group

# ลอง code เพื่อเปรียบเทียบข้อมูลที่อยู่ในกราฟว่าตรงกันรึเปล่า?
data_cut.iloc[10:-58,2:-1]
```

Out[53]:

	population_group
10	0

จากนั้นลองทำการรัน code  
เพื่อเปรียบเทียบว่าข้อมูลตรงกันหรือไม่? และข้อมูลในจุดนี้มีความสัมพันธ์กันหรือไม่?

- ข้อมูล mobile\_group ที่ no.10 ก็จะจัดอยู่ในกลุ่มที่ 0 คือในระดับ low
- ข้อมูล subscriber\_group ที่ no.10 ก็จะจัดอยู่ในกลุ่มที่ 0 คือในระดับ low
- ข้อมูล household\_group ที่ no.10 ก็จะจัดอยู่ในกลุ่มที่ 0 คือในระดับ low
- ข้อมูล population\_group ที่ no.10 ก็จะจัดอยู่ในกลุ่มที่ 0 คือในระดับ low

จากตัวอย่างแสดงให้เห็นว่าข้อมูลที่จุดเดียวกันมีค่าตรงกันหรือค่าเท่ากัน และการเปรียบเทียบระหว่างกราฟและการเขียนโค้ดเพื่อหาคำตอบ จึงสรุปได้ว่า ข้อมูลในจุดนี้มีความสัมพันธ์กัน

# รายชื่อสมาชิก (กลุ่มลูกหมี)

นางสาวมินตรา ทิพย์รัตน์สุนทร

623020041-2

นางสาวกัลยารัตน์ แสนสมบัติ

623020513-7

นางสาวฐิติชญา ไกรวงศ์

623020520-0

นางสาวนันทิชา วิชิต

623020526-8

นางสาวศศิกานต์ บุญมี

623020539-9

