

ข้อมูล Insight ของ Group7

- ภาพรวมของข้อมูล

คำอธิบายแบบย่อ: ข้อมูลทั้งหมด 222 ข้อมูล เป็นการเก็บข้อมูลของผู้ใช้งานอายุเฉลี่ย 26 ปี จำนวน 4 คน ชาย 1 หญิง 3 คนจากช่วงระยะเวลา 16 วัน ตั้งแต่วันที่ 23 พฤศจิกายน ถึง 8 ธันวาคม 2021

Result:

```
Successfully connected to the database

===== ภาพรวมของข้อมูลทั้งหมด =====

จำนวนข้อมูลทั้งหมด: 222 rows
เริ่มเก็บเมื่อวันที่ 2021-11-23 ถึง 2021-12-08

จำนวนผู้ใช้งาน: 4 คน
อายุเฉลี่ย(ปี): 26 ซึ่งอยู่ในช่วง 18 ถึง 30
น้ำหนักเฉลี่ย(kg): 54 ซึ่งอยู่ในช่วง 47.5 ถึง 70.0

จำนวน Male: 1 คน
จำนวน Female: 3 คน

Database closed
```

Code:

```
In [78]: import mysql.connector
from getpass import getpass

user = "root"
pwd = "anyamane28"

my_db, my_cursor = None, None
try:
    my_db = mysql.connector.connect(host='localhost',
                                    user=user,
                                    password=pwd,
                                    database='dads4002_final')
    print("Successfully connected to the database")

    # 1. Create a cursor for this database connection.
    my_cursor = my_db.cursor()

    # SQL command
    # Create strings of MySQL commands
    sql_commands = """
        SELECT count(log_id) as data_number,
        min(date) as start_date,
        max(date) as end_date
        FROM foodlogs;

        -----
        SELECT count(user_id) as subject_number,
        avg(timestampdiff(year,birthday,now())) as Avg_Age,
        max(timestampdiff(year,birthday,now())) as MAX_Age,
        min(timestampdiff(year,birthday,now())) as MIN_Age,
        avg(init_weight) as Avg_Weight,
        max(init_weight) as MAX_Weight,
        min(init_weight) as MIN_Weight
        FROM users;

        -----
        SELECT gender, count(user_id) as number_byGender
        FROM users
        group by gender;"""

    # Execute multiple SQL commands and iterate for each query result
    # Iterate on the returned value of execute() in order to access results from each query separately.
    for i, result in enumerate(my_cursor.execute(sql_commands, multi = True)):
        #print(f"===== Query result no.{i+1} =====")
        if result.with_rows:
            #print(result.fetchall())
            # Loop through the rows
            for row in result.fetchall():
                print()
                if i == 1:
                    #print("===== ภาพรวมของข้อมูลทั้งหมด =====\n")
                    print(f' จำนวนผู้ใช้งาน: {row[0]} คน \n อายุเฉลี่ย(ปี): {row[1]:.0f} ซึ่งอยู่ในช่วง {row[3]} ถึง {row[2]}\n น้ำหนักเฉลี่ย(kg): {row[5]}')
                if i == 0:
                    #print("===== ภาพรวมของข้อมูลทั้งหมด =====\n")
                    print(f' จำนวนข้อมูลทั้งหมด: {row[0]} rows \n เริ่มเก็บเมื่อวันที่ {str(row[1])} ถึง {str(row[2])}')
                if i == 2:
                    print(f' จำนวน {row[0]}: {row[1]} คน ')
            else:
                print(f"Number of rows affected by statement '{result.statement}':{result.rowcount}")
```

- ข้อมูลของอาหารแต่ละประเภท

Code ส่วนการ query data จาก mySQL:

Query ข้อมูลอาหารแต่ละประเภท

```
In [5]: 1 import mysql.connector
2 from getpass import getpass
3
4 user = "root"
5 pwd = "MtZ_2982535*"
6
7 my_db, my_cursor = None, None
8 try:
9     my_db = mysql.connector.connect(host='localhost',
10                                     user=user,
11                                     password=pwd,
12                                     database='dads4002_final')
13     print("Successfully connected to the database")
14
15     # 1.Create a cursor for this database connection.
16     my_cursor = my_db.cursor()
17
18     # SQL command
19     # Create strings of MySQL commands
20     sql_commands = """
21         SELECT foodtype.type_name,
22             count(foodlogs.type_id) as count_food_type,
23             (count(foodlogs.type_id)/max(foodlogs.log_id))*100 as percent_f
24         FROM foodlogs
25             join foodtype
26             on foodlogs.type_id = foodtype.type_id
27         group by foodlogs.type_id
28         order by count_food_type desc;
29         -----
30         SELECT food_national,
31             count(foodlogs.food_national) as count_food_national,
32             (count(foodlogs.food_national)/max(foodlogs.log_id))*100 as per
33         FROM foodlogs
34         group by foodlogs.food_national
35         order by count_food_national desc;
36         -----
37         SELECT foodlogs.type_id,
38             foodtype.type_name,
39             foodlogs.food_group,
40             count(foodlogs.food_group) as count_food_group,
41             (count(foodlogs.food_group)/( select count(foodlogs.food_group)
42                                             FROM foodlogs
43                                             where foodlogs.type_id = "T
44         FROM foodlogs
45             join foodtype
46             on foodlogs.type_id = foodtype.type_id
47         where foodlogs.type_id = "T00002"
48         group by foodlogs.food_group
49         order by count_food_group desc;
50         -----
51         SELECT foodlogs.type_id,
52             foodtype.type_name,
53             foodlogs.food_group,
54             count(foodlogs.food_group) as count_food_group,
55             (count(foodlogs.food_group)/( select count(foodlogs.food_group)
56                                             FROM foodlogs
57                                             where foodlogs.type_id = "T
58         FROM foodlogs
59             join foodtype
60             on foodlogs.type_id = foodtype.type_id
61         where foodlogs.type_id = "T00001"
62         group by foodlogs.food_group
63         order by count_food_group desc;
```

```

65         SELECT foodlogs.type_id,
66                foodtype.type_name,
67                foodlogs.food_group,
68                count(foodlogs.food_group) as count_food_group,
69                (count(foodlogs.food_group)/( select count(foodlogs.food_group)
70                                              FROM foodlogs
71                                              where foodlogs.type_id not
72                                              FROM foodlogs
73                                              join foodtype
74                                              on foodlogs.type_id = foodtype.type_id
75                                              where foodlogs.type_id not in ( "T00001","T00002")
76                                              group by foodlogs.food_group
77                                              order by count_food_group desc;""
78
79     # Execute multiple SQL commands and iterate for each query result
80     # Iterate on the returned value of execute() in order to access results from ea
81     for i,result in enumerate(my_cursor.execute(sql_commands, multi = True)):
82         if i == 0:
83             result11 = result.fetchall()
84         if i == 1:
85             result12 = result.fetchall()
86         if i == 2:
87             result13 = result.fetchall()
88         if i == 3:
89             result14 = result.fetchall()
90         if i == 4:
91             result15 = result.fetchall()
92
93     #else:
94         #print(f"Number of rows affected by statement '{result.statement}':{res
95
96 except Exception as e:
97     print("Something went wrong while attempting to connect the database.")
98     if my_db is not None:
99         my_db.rollback()
100    print(f">>>Error{e}")
101 finally:
102     if my_cursor is not None:
103         my_cursor.close()
104     if my_db is not None:
105         my_db.close()
106     print("\nDatabase closed")

```

Successfully connected to the database

Database closed

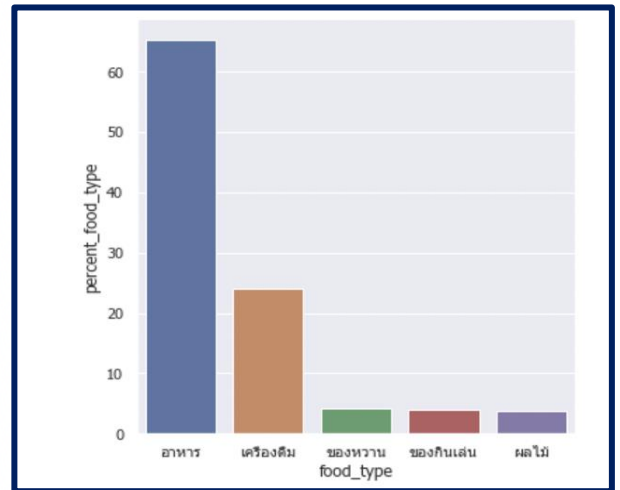
- **Insight1:** ข้อมูลการแบ่งตามประเภทของอาหาร

คำอธิบายแบบย่อ: อาหารจะถูกแบ่งประเภทเป็น เครื่องดื่ม อาหาร ของหวาน ของกินเล่น และผลไม้ ซึ่งอาหารจะเป็นอาหารหลักที่ผู้ใช้รับประทานอาหารมากที่สุด ถึง 65% รองลงมาคือ เครื่องดื่ม 23.9% ของหวาน 4.2% และ ของกินเล่น ผลไม้ เฉลี่ยการกิน อยู่ที่ 3.9% และ 3.8% ตามลำดับ

RESULT:

Out[24]:

	food_type	count_number	percent_food_type
0	อาหาร	145	65.3153
1	เครื่องดื่ม	53	23.9819
2	ของหวาน	9	4.2056
3	ของกินเล่น	8	3.9801
4	ผลไม้	7	3.8462



CODE:

ข้อมูลของอาหารแบ่งตามประเภทอาหาร

```
In [24]: 1 food_type = pd.DataFrame(result11)
2 food_type = food_type.rename(columns = {0:'food_type', 1:'count_number', 2:'percent_food_type'})
3 food_type
```

```
In [28]: 1 sns.set(font="Tahoma", font_scale = 1)
2 sns.catplot(x = "food_type", y = 'percent_food_type', data = food_type, kind = 'bar')
3 plt.show()
```

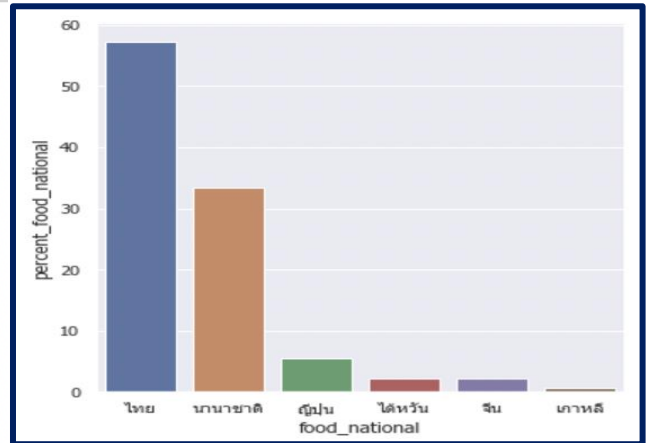
- **Insight2:** ข้อมูลการแบ่งตามสัญชาติอาหาร

คำอธิบายแบบย่อ: อาหารถูกจัดประเภทตามสัญชาติ โดยส่วนใหญ่ร้อยละ 57 นิยมรับประทาน อาหารไทย ,อาหารนานาชาติ ร้อยละ 33.48 , ญี่ปุ่น ร้อยละ 5.5 ไต้หวันและจีน ร้อยละ2.2 เกาหลี ร้อยละ0.6

RESULT:

Out[30]:

	food_national	count_number	percent_food_national
0	ไทย	127	57.2072
1	นานาชาติ	74	33.4842
2	ญี่ปุ่น	12	5.5300
3	ไต้หวัน	4	2.2599
4	จีน	4	2.1739
5	เกาหลี	1	0.6061



CODE:

ข้อมูลของอาหารแบ่งตามสัญชาติอาหาร

```

1 food_national = pd.DataFrame(result12)
2 food_national = food_national.rename(columns = {0:'food_national', 1:'count_number', 2:'percent_food_natio
3 food_national

1 sns.set(font="Tahoma", font_scale = 1)
2 sns.catplot(x = "food_national", y = 'percent_food_national' ,data = food_national, kind = 'bar')
3 plt.show()

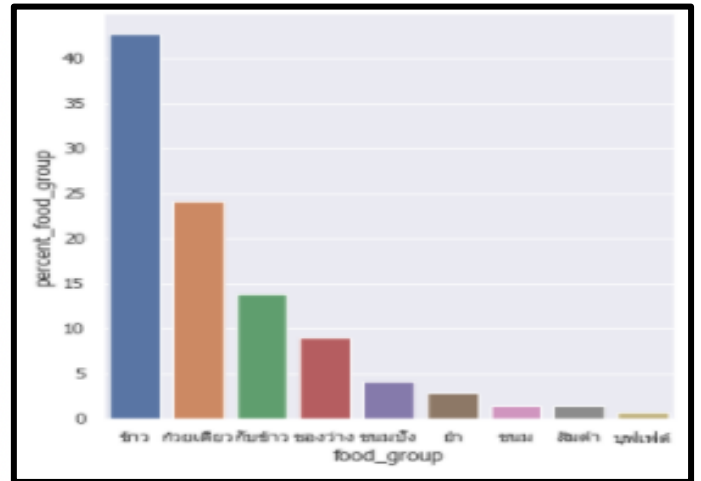
```

- **Insight3: ข้อมูลเมนูอาหารที่ได้รับความนิยม**

คำอธิบายแบบย่อ: เมนูอาหารที่ได้รับความนิยมสูงสุด คือ อาหารจานเดียว แบ่งเป็นประเภทข้าว 42.76% รองลงมาคือ ก๋วยเตี๋ยว 24.14% รองลงมาอันดับ 3 คือ ประเภทกับข้าว 13.79% ขณะที่เครื่องดื่มที่ได้รับความนิยมสูงสุด คือ กาแฟ 66.04% และชา 18.87% สำหรับเมนูของว่างที่ได้รับความนิยมสูงสุด คือ ของหวาน 37.50% และผลไม้ 29.17%

RESULT:

	type_id	type_name	food_group	count_food_group	percent_food_group
0	T00002	อาหาร	ข้าว	62	42.7586
1	T00002	อาหาร	ก๋วยเตี๋ยว	35	24.1379
2	T00002	อาหาร	กับข้าว	20	13.7931
3	T00002	อาหาร	ของว่าง	13	8.9655
4	T00002	อาหาร	ขนมปัง	6	4.1379
5	T00002	อาหาร	ยำ	4	2.7586
6	T00002	อาหาร	ขนม	2	1.3793
7	T00002	อาหาร	ส้มตำ	2	1.3793
8	T00002	อาหาร	บุฟเฟ่ต์	1	0.6897



CODE:

ข้อมูลอาหารที่จำแนกตามหมวดหมู่อาหาร

```
In [32]: #อาหาร
food_group1 = pd.DataFrame(result13)
food_group1 = food_group1.rename(columns = {0:'type_id', 1:'type_name', 2:'food_group', 3:'count_food_group', 4:'percent_food_group'})
food_group1
```

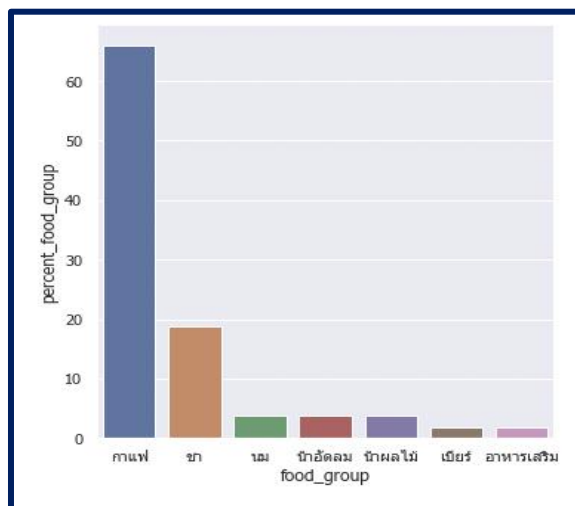
```
sns.set(font="Tahoma", font_scale = 1)
sns.catplot(x = "food_group", y = 'percent_food_group', data = food_group1, kind = 'bar')
plt.show()
```

- Insight4: ข้อมูลอาหารที่จำแนกตามหมวดหมู่เครื่องดื่ม

คำอธิบาย: เครื่องดื่มที่ได้รับความนิยมสูงสุด คือ กาแฟ 66.04% ชา 18.87% ขณะที่ น้ำอัดลม น้ำผลไม้ มีความนิยมใกล้เคียงกัน อยู่ที่อย่างละ 3.77%

RESULT:

type_id	type_name	food_group	count_food_group	percent_food_group	
0	T00001	เครื่องดื่ม	กาแฟ	35	66.0377
1	T00001	เครื่องดื่ม	ชา	10	18.8679
2	T00001	เครื่องดื่ม	นม	2	3.7738
3	T00001	เครื่องดื่ม	น้ำอัดลม	2	3.7738
4	T00001	เครื่องดื่ม	น้ำผลไม้	2	3.7738
5	T00001	เครื่องดื่ม	เบียร์	1	1.8888
6	T00001	เครื่องดื่ม	อาหารเสริม	1	1.8888



CODE:

```

1 #เครื่องดื่ม
2 food_group2 = pd.DataFrame(result14)
3 food_group2 = food_group2.rename(columns = {0:'type_id', 1:'type_name', 2:'food_group'})
4 food_group2

1 sns.set(font="Tahoma", font_scale = 1)
2 sns.catplot(x = "food_group", y = 'percent_food_group' ,data = food_group2, kind = 'bar')
3 plt.show()

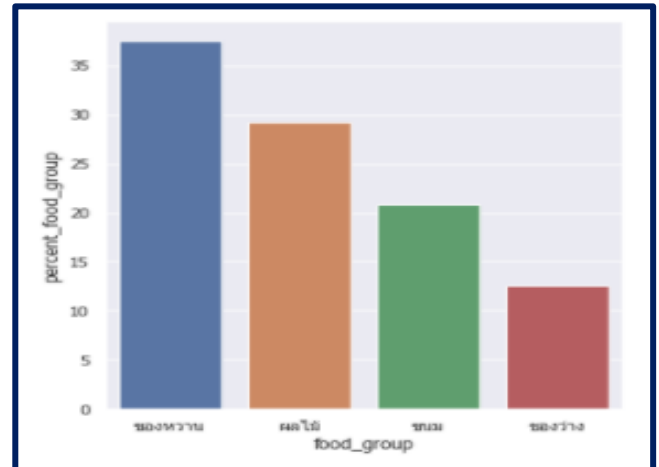
```


- **Insight5:** ข้อมูลอาหารที่จำแนกตามหมวดหมู่ของว่าง

คำอธิบาย: สำหรับเมนูของว่างที่ได้รับความนิยมสูงสุด คือ ของหวาน 37.50% ผลไม้ 29.17% และขนม 20.83%

RESULT:

	type_id	type_name	food_group	count_food_group	percent_food_group
0	T00004	ของหวาน	ของหวาน	9	37.5000
1	T00005	ผลไม้	ผลไม้	7	29.1667
2	T00003	ของกินเล่น	ขนม	5	20.8333
3	T00003	ของกินเล่น	ของว่าง	3	12.5000



CODE:

```
#ของว่าง
food_group3 = pd.DataFrame(result15)
food_group3 = food_group3.rename(columns = {0:'type_id', 1:'type_name', 2:'food_group', 3:'count_food_group', 4:'percent_food_group'})
food_group3

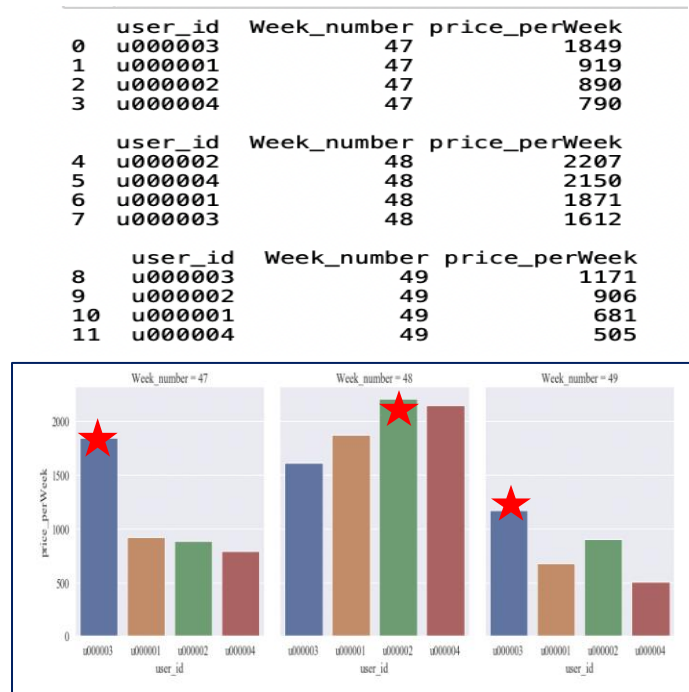
sns.set(font="Tahoma", font_scale = 1)
sns.catplot(x = "food_group", y = 'percent_food_group' ,data = food_group3, kind = 'bar')
plt.show()
```


ข้อสรุปที่จากการวิเคราะห์ข้อมูล

- ข้อมูลค่าใช้จ่ายเฉลี่ยในแต่ละสัปดาห์

คำอธิบายแบบย่อ: จากข้อมูล 3 สัปดาห์ พบว่าผู้ใช้ ID u000003 มีแนวโน้มการใช้จ่ายสำหรับค่าอาหารเฉลี่ยสูงสุดมากกว่าผู้ใช้คนอื่นๆ ในสัปดาห์ที่ 47 และ 48 ซึ่งเป็นการเปรียบเทียบข้อมูลของทุกคนในสัปดาห์เดียวกัน

RESULT:



CODE:

```
1 #Price per week
2 price_perWeek = pd.DataFrame(result21)
3 price_perWeek = price_perWeek.rename(columns = {0:'user_id', 1:'Week_number', 2:'price_perWeek'})
4
5 price_week1 = price_perWeek[price_perWeek['Week_number'] == 47]
6 price_week2 = price_perWeek[price_perWeek['Week_number'] == 48]
7 price_week3 = price_perWeek[price_perWeek['Week_number'] == 49]
8
9 print(f"{price_week1}\n")
10 print(f"{price_week2}\n")
11 print(f"{price_week3}")
12
```



```
1 sns.catplot(x = "user_id", y = 'price_perWeek' , col="Week_number",data = price_perWeek)
2 sns.set(font="Angsana New", font_scale = 1)
3 plt.show()
```

- ข้อมูลเปรียบเทียบพฤติกรรมของ USER ในแต่ละสัปดาห์

Query ข้อมูลพฤติกรรมของ user

```
1 user = "root"
2 pwd = 
3
4 my_db,my_cursor = None, None
5 try:
6     my_db = mysql.connector.connect(host='localhost',
7                                     user=user,
8                                     password=pwd,
9                                     database='dads4002_final')
10    print("Successfully connected to the database")
11
12    # 1.Create a cursor for this database connection.
13    my_cursor = my_db.cursor()
14
15    # SQL command
16    # Create strings of MySQL commands
17    sql_commands = """
18        -- sum price
19        SELECT foodlogs.user_id,
20        WEEK(foodlogs.`date`,0) as Week_number,
21        sum(foodlogs.food_price) as price_perWeek
22        FROM foodlogs
23        group by WEEK(foodlogs.`date`,0),user_id
24        order by WEEK(foodlogs.`date`,0), price_perWeek desc;
25        -----
26        SELECT user_id, Week_number, avg(daily_cals ) as Avgcalories_pe
27        FROM (select user_id,`date`, WEEK(`date`,0) as Week_number, sum
28        group by Week_number,user_id
29        order by Week_number, Avgcalories_perWeek desc;
30        -----
31        SELECT user_id, WEEK(`date`,0) as Week_number,
32        avg(daily_step) as Avgstep_perWeek
33        FROM usertracking
34        group by Week_number,user_id
35        order by Week_number, Avgstep_perWeek desc;
36        -----
37        -- avg sleep
38        SELECT user_id, WEEK(`date`,0) as Week_number,
39        avg(daily_sleep) as Avgsleep_perWeek
40        FROM usertracking
41        group by Week_number,user_id
42        order by Week_number, Avgsleep_perWeek desc;
43        -----
44        SELECT user_id, WEEK(`date`,0) as Week_number,
45        avg(daily_drink) as Avgdrink_perWeek
46        FROM usertracking
47        group by Week_number,user_id
48        order by Week_number, Avgdrink_perWeek desc;
49        -----
50        select distinct hour(time), count(hour(time))
51        from foodlogs
52        group by hour(time)
53        order by hour(time);
54        """
```

```

55
56 # Execute multiple SQL commands and iterate for each query result
57 # Iterate on the returned value of execute() in order to access results from each
58 for i,result in enumerate(my_cursor.execute(sql_commands, multi = True)):
59     if i == 0:
60         result21 = result.fetchall()
61     if i == 1:
62         result22 = result.fetchall()
63     if i == 2:
64         result23 = result.fetchall()
65     if i == 3:
66         result24 = result.fetchall()
67     if i == 4:
68         result25 = result.fetchall()
69     if i == 5:
70         result26 = result.fetchall()
71
72     #else:
73         #print(f"Number of rows affected by statement '{result.statement}':{result.affected_rows}")
74
75 except Exception as e:
76     print("Something went wrong while attempting to connect the database.")
77     if my_db is not None:
78         my_db.rollback()
79     print(f">>>Error{e}")
80 finally:
81     if my_cursor is not None:
82         my_cursor.close()
83     if my_db is not None:
84         my_db.close()
85
86     print("\n Database closed")

```

Successfully connected to the database

Database closed

- ข้อมูลแคลอรีเฉลี่ยของอาหารที่รับประทานในแต่ละสัปดาห์

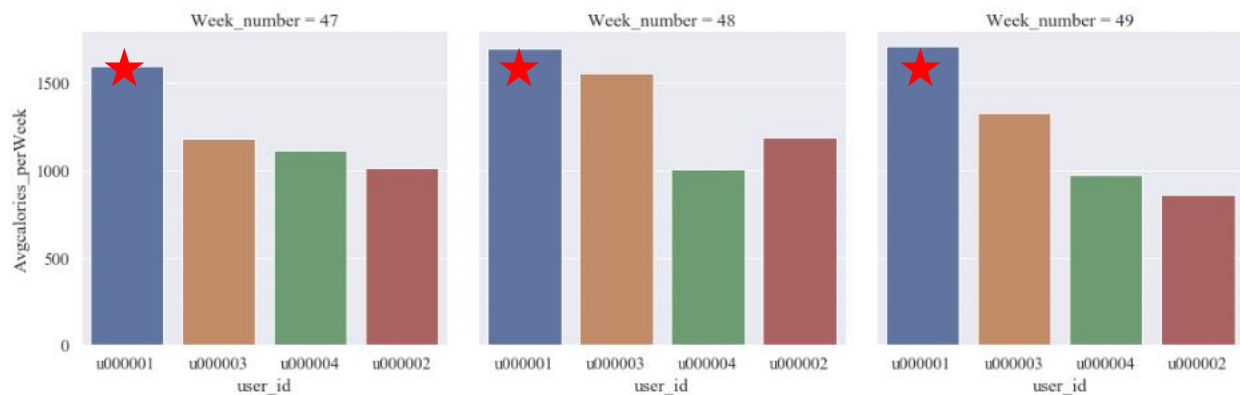
คำอธิบายแบบย่อ : แคลอรีเฉลี่ยในแต่ละสัปดาห์จะพบว่า พบว่าผู้ใช้ ID u000001 ซึ่งเป็นผู้ชายจะรับประทานอาหารที่มีแคลอรีค่อนข้างสูงในทุกสัปดาห์ที่เราวิเคราะห์ข้อมูล และผู้ใช้ส่วนใหญ่กินอาหารเฉลี่ยมากกว่า 1000-1700 cal.

RESULT:

	user_id	Week_number	Avgcalories_perWeek
0	u000001	47	1596.0000
1	u000003	47	1180.0000
2	u000004	47	1109.6000
3	u000002	47	1014.7500

	user_id	Week_number	Avgcalories_perWeek
4	u000001	48	1692.0000
5	u000003	48	1549.8000
6	u000002	48	1184.1667
7	u000004	48	1007.2857

	user_id	Week_number	Avgcalories_perWeek
8	u000001	49	1704.3333
9	u000003	49	1323.6667
10	u000004	49	972.6667
11	u000002	49	855.5000



CODE:

```

1 #Average Calories per week
2 avgcal_perWeek = pd.DataFrame(result22)
3 avgcal_perWeek = avgcal_perWeek.rename(columns = {0:'user_id', 1:'Week_number', 2:'Avgcalories_perWeek'})
4
5 avgcal_week1 = avgcal_perWeek[avgcal_perWeek['Week_number'] == 47]
6 avgcal_week2 = avgcal_perWeek[avgcal_perWeek['Week_number'] == 48]
7 avgcal_week3 = avgcal_perWeek[avgcal_perWeek['Week_number'] == 49]
8
9 print(f"{avgcal_week1}\n")
10 print(f"{avgcal_week2}\n")
11 print(f"{avgcal_week3}\n")

```

```

1 sns.set(font="Angsana New", font_scale = 2)
2 sns.catplot(x = "user_id", y = 'Avgcalories_perWeek', col="Week_number",data = avgcal_perWeek)
3 plt.show()

```


- ข้อมูลจำนวนก้าวเฉลี่ยในแต่ละสัปดาห์

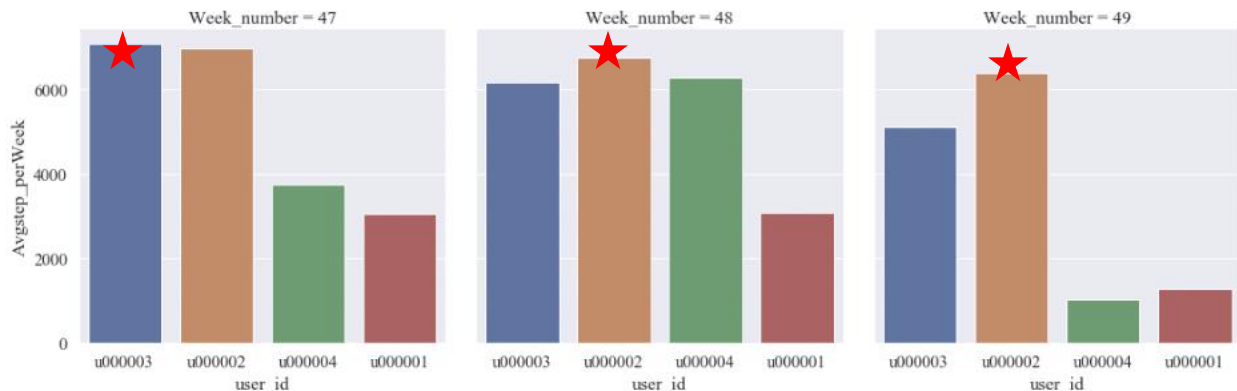
คำอธิบายแบบย่อ : จากข้อมูลจะพบว่าผู้ใช้ u000003 และ u000002 มีค่าเฉลี่ยในการเดินที่ใกล้เคียงกัน ระหว่าง 6,000-7000 ก้าวในทุก ๆ สัปดาห์ ในทางตรงกันข้ามผู้ขาย user u000001 มีการเดินเฉลี่ยน้อยกว่าผู้หญิง ถึง 3,000 ก้าวตลอดสัปดาห์ ในสัปดาห์สุดท้าย u000004 มีการเดินน้อยที่สุด เพียง 1,047 ก้าว

RESULT:

	user_id	Week_number	Avgstep_perWeek
0	u000003	47	7072.6667
1	u000002	47	6960.7500
2	u000004	47	3754.4000
3	u000001	47	3053.2000

	user_id	Week_number	Avgstep_perWeek
4	u000002	48	6742.6667
5	u000004	48	6279.5714
6	u000003	48	6160.0000
7	u000001	48	3075.8571

	user_id	Week_number	Avgstep_perWeek
8	u000002	49	6402.5000
9	u000003	49	5111.6667
10	u000001	49	1290.0000
11	u000004	49	1047.0000



CODE:

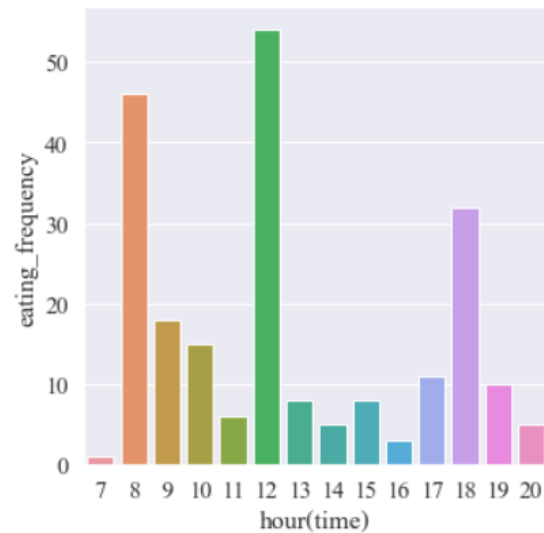
```
1 #Average Step per week
2 avgstep_perWeek = pd.DataFrame(result23)
3 avgstep_perWeek = avgstep_perWeek.rename(columns = {0:'user_id', 1:'Week_number', 2:'Avgstep_perWeek'})
4
5 avgstep_week1 = avgstep_perWeek[avgstep_perWeek['Week_number'] == 47]
6 avgstep_week2 = avgstep_perWeek[avgstep_perWeek['Week_number'] == 48]
7 avgstep_week3 = avgstep_perWeek[avgstep_perWeek['Week_number'] == 49]
8
9 print(f"{avgstep_week1}\n")
10 print(f"{avgstep_week2}\n")
11 print(f"{avgstep_week3}\n")
```

```
1 sns.set(font="Angsana New", font_scale = 2)
2 sns.catplot(x = "user_id", y = 'Avgstep_perWeek' , col="Week_number" ,data = avgstep_perWeek)
3 plt.show()
```

คำอธิบายแบบย่อ: ตลอดระยะเวลาเก็บข้อมูล 2 อาทิตย์ users ทั้ง 4 คน กินอาหารระหว่างเวลา 7.00 - 21.00 น. (ทั้งนี้ ช่วง 7 หมายถึง 7.00-7.59 น.) โดยตลอดทั้งวัน ช่วงที่มีความถี่ในการกินสูงสุด คือ ช่วงเวลากลางวัน ระหว่าง 12.00-12.59 น. มีความถี่ 54 ครั้ง ในมือเช้ามีความถี่ สูงสุดระหว่างเวลา 8.00-8.59 น. 46 ครั้ง และช่วงเย็น เวลา 18.00-18.59 น. ความถี่ 32 ครั้ง

RESULT:

	hour(time)	eating_frequency
0	7	1
1	8	46
2	9	18
3	10	15
4	11	6
5	12	54
6	13	8
7	14	5
8	15	8
9	16	3
10	17	11
11	18	32
12	19	10
13	20	5



CODE:

```
#Eating Frequency per hour
count_hour = pd.DataFrame(result26)
count_hour = count_hour.rename(columns = {0:'hour(time)', 1:'eating_frequency'})
count_hour

sns.catplot(x = "hour(time)", y = 'eating_frequency' ,data = count_hour, kind = 'bar')
plt.show()
```