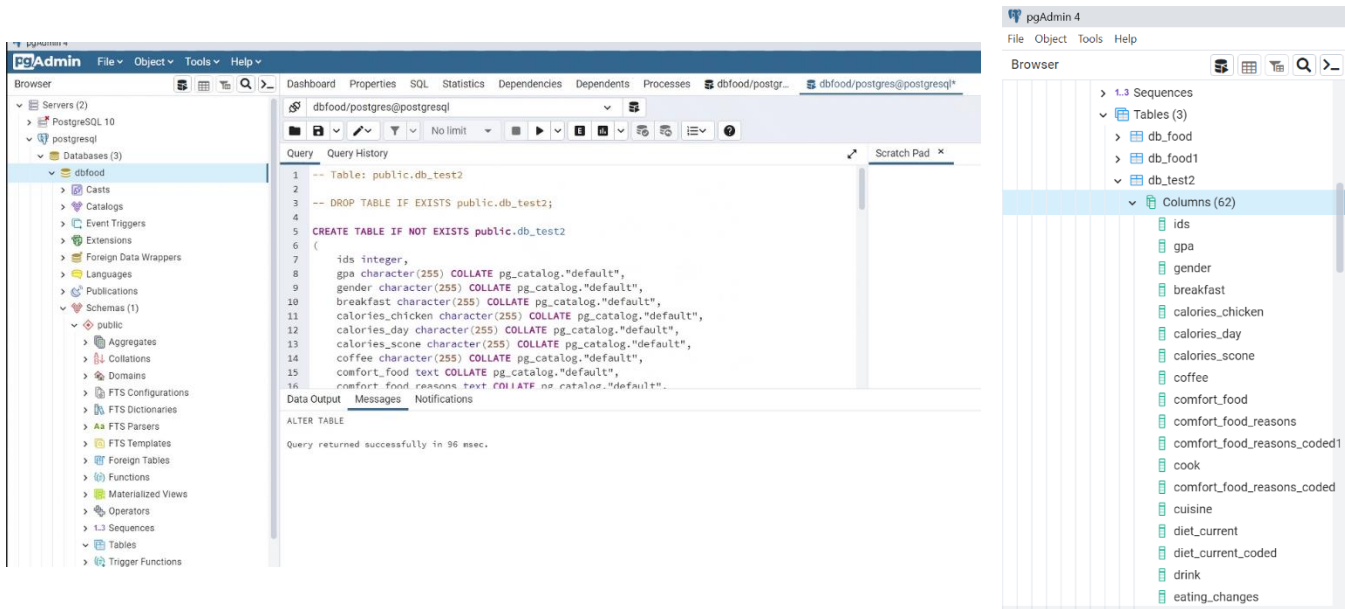


```
D:\_1_project.py X  
D: > DADS6005_DataStreaming > Project > \_1_project.py > ...  
  
47 #here %S means string values  
48 sql = """INSERT INTO myfood.food_coded (GPA, Gender, breakfast, calories_chicken, calories_  
49 coffee, comfort_food, comfort_food_reasons, comfort_food_reasons_coded, cool  
50 diet_current, diet_current_coded, drink, eating_changes, eating_changes_coded,  
51 eating_changes_coded1, eating_out, employment, ethnic_food, exercise, father  
52 father_profession, fav_cuisine, fav_cuisine_coded, fav_food, food_childhood,  
53 grade_level, greek_food, healthy_feeling, healthy_meal, ideal_diet, ideal_d  
54 indian_food, italian_food, life_rewarding, marital_status, meals_dinner_fri  
55 mother_profession, nutritional_check, on_off_campus, parents_cook, payMeal,  
56 self_perception_weight, soup, sports, thai_food, tortilla_calories, turkey_  
57 veggies_day, vitamins, waffle_calories, weight)  
58 VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,  
59 %s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,  
60 cursor.execute(sql, tuple(row.fillna(0)))  
61 print(f"Record inserted {i+1}")  
62 # the connection is not auto committed by default, so we must commit to save our changes  
63 conn.commit()  
  
64 except Error as e:  
65     print("Error while connecting to MySQL", e)
```



2. ใช้ Python เพื่อเขียนแหล่งข้อมูลจาก CSV ในฐานข้อมูล SQL (DB) (ส่วนต้นทาง) จาก Postgres ระยะเวลาการเขียนอยู่ที่ 1 วินาที

In [1]: `!pip show psycopg2`

```

Name: psycopg2
Version: 2.9.6
Summary: psycopg2 - Python-PostgreSQL Database Adapter
Home-page: https://psycopg.org/
Author: Federico Di Gregorio
Author-email: fog@initd.org
License: LGPL with exceptions
Location: c:\users\khanistha\anaconda3\lib\site-packages
Requires:
Required-by:

```

In [2]: `!pip install psycopg2`

Requirement already satisfied: psycopg2 in c:\users\khanistha\anaconda3\lib\site-packages (2.9.6)

## เชื่อมต่อ ฐานข้อมูล Postgres

```

In [3]: import psycopg2
from sqlalchemy import create_engine
import pandas as pd
import os
import time

from sqlalchemy import create_engine

engine = psycopg2.connect(database="dbfood",
                        host="localhost",
                        user="postgres",
                        password="123456",
                        port="5433")

```

Read data from history table

```
In [7]: df = pd.read_sql('Select * from public.db_test2',engine)
df = df[['ids','gpa','gender',
'breakfast','calories_chicken',
'calories_day','calories_scone',
'coffee','comfort_food','comfort_food_reasons',
'comfort_food_reasons_coded1','cook',
'comfort_food_reasons_coded','cuisine',
'diet_current','diet_current_coded','drink',
'eating_changes','eating_changes_coded',
'eating_changes_coded1','eating_out','employment',
'ethnic_food','exercise','father_education',
'father_profession','fav_cuisine','fav_cuisine_coded',
'fav_food','food_childhood','fries','fruit_day','grade_level',
'greek_food','healthy_feeling','healthy_meal','ideal_diet',
'ideal_diet_coded','income','indian_food','italian_food',
'life_rewarding','marital_status','meals_dinner_friend',
'mother_education','mother_profession','nutritional_check',
'on_off_campus','parents_cook','pay_meal_out','persian_food',
'self_perception_weight','soup','sports','thai_food','tortilla_calories',
'turkey_calories','type_sports','veggies_day','vitamins','waffle_calories','weights']]

df.head()
```

```
Out[7]:
```

	ids	gpa	gender	breakfast	calories_chicken	calories_day	calories_scone	coffee	comfort_food	comfort_food_reasons	...	soup	sports	thai_food	tortil
0	1	2.4 ...	2 ...	1 ...	430 ...	0 ...	315 ...	1 ...	none	we dont have comfort	...	1 ...	1 ...	1 ...	
1	2	3.654 ...	1 ...	1 ...	610 ...	3 ...	420 ...	2 ...	chocolate, chips, ice cream	Stress, bored, anger	...	1 ...	1 ...	2 ...	
2	3	3.3 ...	1 ...	1 ...	720 ...	4 ...	420 ...	2 ...	frozen yogurt, pizza, fast food	stress, sadness	...	1 ...	2 ...	5 ...	

## Insert data into FactInernetSales

```
In [8]: print("Row Inserted ")
```

Row Inserted

```
In [*]: import sqlalchemy
import pandas as pd
import time

for index, row in df.head(50).iterrows():
    #mod = df.iloc[i:i+chunksize]
    mod = pd.DataFrame(row.to_frame().T)
    #mod.to_sql('db_food_coded2_stream', engine, if_exists='append', index=False)

    print("Row Food name: " + mod.comfort_food_reasons.astype(str) + " " + mod.tortilla_calories.astype(str).astype(str))
    time.sleep(1)
```

```
0 Row Food name: we dont have comfort 1165 ...
dtype: object
1 Row Food name: Stress, bored, anger 725 ...
dtype: object
2 Row Food name: stress, sadness 1165 ...
dtype: object
3 Row Food name: Boredom 725 ...
dtype: object
4 Row Food name: Stress, boredom, cravings 940 ...
dtype: object
5 Row Food name: None, i don't eat comfort food....
dtype: object
```

3. เชื่อมต่อ SQL DB กับ Kafka Cluster ผ่าน topic 1 และส่งต่อให้ topic 2 โดยส่วนต้นทาง (NoSQL DB) จะใช้ข้อมูลจาก topic 3 ที่เพิ่มขึ้นอย่างต่อเนื่อง



```
In [1]: from kafka import KafkaConsumer
bootstrap_servers = ['localhost:29092']
consumer = KafkaConsumer(bootstrap_servers=bootstrap_servers)
consumer.topics()
```

```
Out[1]: {'__confluent.support.metrics',
'_schemas',
'connect-status',
'connect_configs',
'connect_offsets',
'source.public.db_food_coded2_stream',
'source.public.db_test2_streaming',
'test',
'topic1'}
```

```
In [*]: import json

bootstrap_servers = ['localhost:29092']

topicName = 'source.public.db_food_coded2_stream'

consumer = KafkaConsumer(topicName, auto_offset_reset='earliest',
                          bootstrap_servers = bootstrap_servers)

# Read and print message from consumer
for msg in consumer:
    print(json.loads(msg.value))
```

```
In [ ]:
```

4. จะเห็นว่ามีข้อมูลรายละเอียด topics ที่เกิดขึ้นจากหน้า docker ดังนี้

CMAC **datastreaming** Cluster ▾ Brokers Topic ▾ Preferred Replica Election Schedule Leader Election Reassign Partitions Consumers

Clusters / datastreaming / Topics

## Operations

Generate Partition  
Assignments

Run Partition  
Assignments

Add  
Partitions

## Topics

Show 25 entries

Search:

Topic	# Partitions	# Brokers	Brokers Spread %	Brokers Skew %	Brokers Leader Skew %	# Replicas	Under Replicated %	Producer Message/Sec	Summed Recent Offsets
__confluent.support.metrics	1	1	100	0	0	1	0	0.00	1
__consumer_offsets	50	1	100	0	0	1	0	0.20	1,256
_schemas	1	1	100	0	0	1	0	0.00	2
connect-status	5	1	100	0	0	1	0	0.00	0
connect_configs	1	1	100	0	0	1	0	0.00	2
connect_offsets	25	1	100	0	0	1	0	0.00	0
source.public.db_food_coded2_stream	1	1	100	0	0	1	0	0.00	0
source.public.db_test2_streaming	1	1	100	0	0	1	0	0.00	0
test	1	1	100	0	0	1	0	0.00	0
topic1	1	1	100	0	0	1	0	0.00	8
topic2	1	1	100	0	0	1	0	0.00	0
topic3	1	1	100	0	0	1	0	0.00	0

## 5. และเมื่อนำมา Plot กราฟจากข้อมูลที่ได้ (ด้วยการใช้งาน Plotly)

```
In [ ]: import sys
import pandas as pd
import numpy as np
import plotly.express as px
import matplotlib as mpl
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
import seaborn
plt.rcParams['font.family']='Tahoma'
print( f"Python {sys.version}" )
print( f"Pandas {pd.__version__}" )
print( f"NumPy {np.__version__}" )
```

```
In [ ]: import pandas as pd      #(version 1.0.0)
import plotly                  #(version 4.5.0)
import plotly.express as px

import dash                    #(version 1.8.0)
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output, State

app = dash.Dash(__name__)

#-----

df = pd.read_csv("food_coded.csv")
df.info()
```

```
In [ ]: # create a list of our conditions :gender
conditions1 = [(df['gender'] == 1),(df['gender'] == 2),]
values1 = ['Female', 'Male']
df['gender_mean'] = np.select(conditions1, values1)

# create a list of our conditions :calories_day
conditions2 = [(df['calories_day'] == 1),(df['calories_day'] == 2),(df['calories_day'] == 3),(df['calories_day'] == 4)]
values2 = ['เงินไม่รู้ว่าเงินควรกินกี่แคล', 'มันไม่สำคัญเลย', 'มีความสำคัญปานกลาง', 'มันสำคัญมาก']
df['calories_day_mean'] = np.select(conditions2, values2)

df
```

```
In [ ]: #graph 1 Top 20 Members of Count Message

groupby_gender=(df.groupby(['gender_mean']).count()[['ids']].sort_values('gender_mean', ascending=False))
groupby_gender

import plotly.graph_objects as go
labels = ['Male', 'Female']
values = [49,76]
```

```
In [ ]: #graph 4 คุณทำอาหารบ่อยแค่ไหน ?

cook_freq = (df.groupby(['cook']).count()[['cook1']].sort_values('cook', ascending=False))
#top_comment = Member_comment.nlargest(20, 'Message')
cook_freq

#graph 5 คุณกิน comfort food เมื่อมีอาการใดแบบใด

com_food_reason = (df.groupby(['comfort_food_reasons_coded_1']).count()[['comfort_food_reasons']].sort_values('comfort_food_reas
#top_comment = Member_comment.nlargest(20, 'Message')
com_food_reason
```

```
# Develop Analysis Dashboard

from dash import Dash, dcc, html, Input, Output
from jupyter_dash import JupyterDash
import pandas as pd
import plotly # (version 4.5.4) #pip install plotly==4.5.4
import plotly.express as px
import plotly.io as pio

app = JupyterDash(__name__)

#graph 1 gender

fig1 = go.Figure(data=[go.Pie(labels=labels, values=values, hole=.3)])
fig1.update_layout(title_text='The proportion of respondents separated by males and females')

#graph 4 คุณทำอาหารบ่อยแค่ไหน ?

fig4 = px.bar(cook_freq, x=cook_freq.index, y="cook1",title='คุณทำอาหารบ่อยแค่ไหน',
,template='simple_white', height=500,width=600)

#graph 5 คุณกิน comfort food เมื่อมีอารมณ์แบบใด ?

fig5 = px.bar(com_food_reason, x=com_food_reason.index, y="comfort_food_reasons",title='คุณกิน comfort food เมื่อมีอารมณ์แบบใด',
,template='simple_white', height=500,width=600)

#graph 6 คุณมีพฤติกรรมกินอาหารอย่างไร ?

fig6 = px.bar(diet_current, x=diet_current.index, y="diet_current",title='คุณมีพฤติกรรมกินอาหารอย่างไร',
,template='simple_white', height=500,width=600)
```

```
#dashboard

app.layout = html.Div([

    html.Div(children=[
        html.H1(children='Data-streaming and real-time analytics',style={'textAlign':'center'}),
        html.Div(children='DAOS6005 - Real-Time Projectd'),

        dcc.Graph(
            id='example-graph1',
            figure=fig1
        )

    ], style={'padding': 10, 'flex': 5}),

    html.Div(children=[
        html.H1(children='Data-streaming and real-time analytics',style={'textAlign':'center'}),
        html.Div(children='DAOS6005 - Real-Time Projectd'),

        dcc.Graph(
            id='example-graph4',
            figure=fig4
        )

    ], style={'padding': 10, 'flex': 5}),

    html.Div(children=[
        html.H1(children='Data-streaming and real-time analytics',style={'textAlign':'center'}),
        html.Div(children='DAOS6005 - Real-Time Projectd'),

        dcc.Graph(
            id='example-graph5',
            figure=fig5
        )

    ], style={'padding': 10, 'flex': 5}),

    html.Div(children=[
        html.H1(children='Data-streaming and real-time analytics',style={'textAlign':'center'}),
        html.Div(children='DAOS6005 - Real-Time Projectd'),

        dcc.Graph(
            id='example-graph6',
            figure = fig6
        )

    ], style={'padding': 10, 'flex': 5}),

], style={'display': 'flex', 'flexDirection': 'row', 'flex-wrap': 'wrap'})

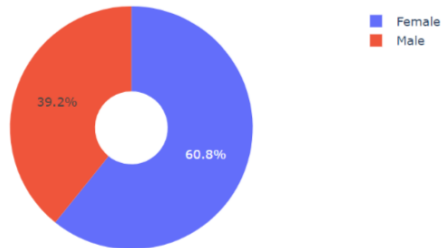
if __name__ == '__main__':
    app.run_server(debug=True, port=8052)
```

จะได้ 5 กราฟ ดังนี้

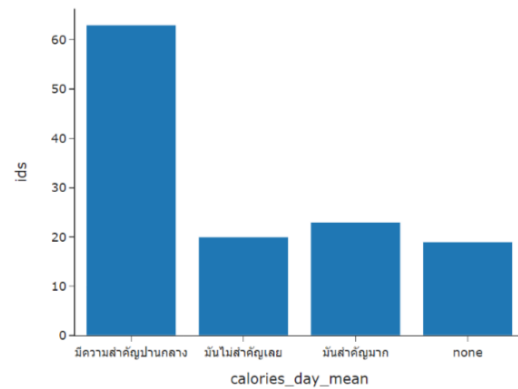
## Data-streaming and real-time analytics

DADS6005 - Real-Time Projectd

The proportion of respondents separated by males and females



The proportion of students assessed the Importance of daily calories

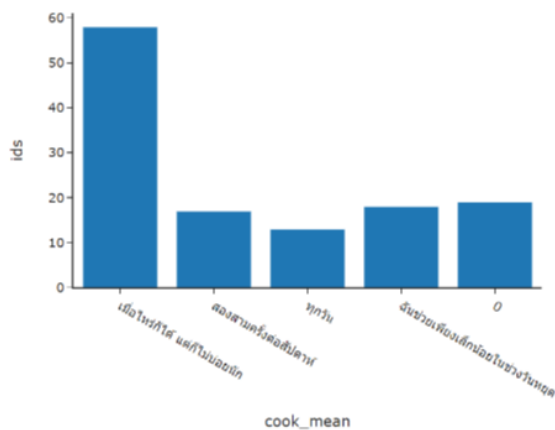


ภาพซ้าย : Pie chart จะเห็นว่าสัดส่วนผู้ตอบแบบสอบถามเป็นฝ่ายหญิงมากกว่าชายที่ 21.6% จากหญิงที่ 60.8% และชายที่ 39.2%

ภาพขวา : กราฟแท่งจะแสดงสัดส่วนความสำคัญของ Calories ต่อวัน โดยคนส่วนใหญ่เห็นว่ามีมีความสำคัญปานกลางมากกว่า 60 คน

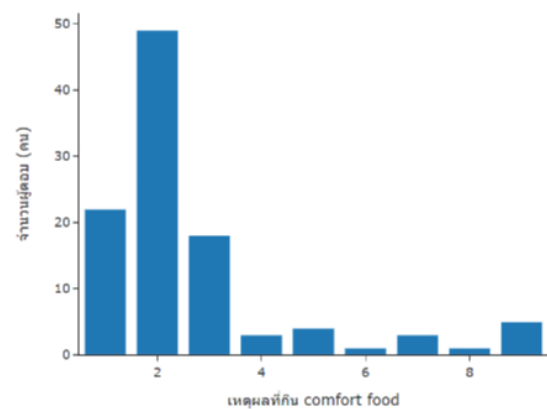
DADS6005 - Real-Time Projectd

คุณทำอาหารบ่อยแค่ไหน



DADS6005 - Real-Time Projectd

เหตุผลที่คุณกิน comfort food

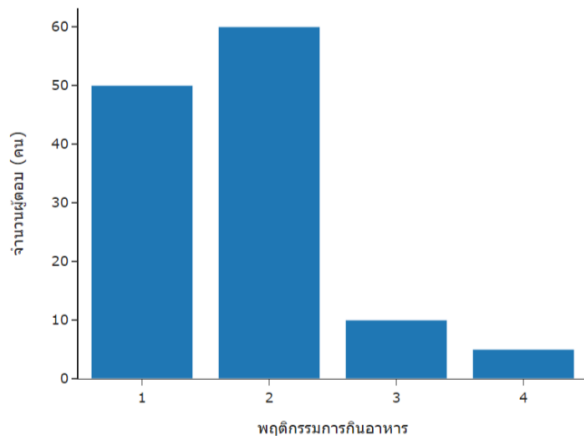


ภาพซ้าย : กราฟแท่งแสดงระดับความบ่อยในการทำอาหารโดยระดับสูงสุดเป็นระดับที่ 3 (Whenever I can, but that is not very often ; เมื่อไหร่ก็ได้ แต่ก็ไม่บ่อยนัก)

ภาพขวา : กราฟแท่งจะแสดงสัดส่วนอารมณ์ที่จะรับประทาน Comfort Food โดยสัดส่วนสูงสุดเป็นระดับ 2 คือ เมื่อมีอารมณ์ความเบื่อ (boredom) เป็นจำนวนเกือบ 50 คน และมีระดับ 1 คือ มีอารมณ์ความเครียด (stress) รองลงมา เป็นจำนวน 22 คน



คุณมีพฤติกรรมการกินอาหารอย่างไร



กราฟสุดท้ายจะแสดงพฤติกรรมการกินของผู้ตอบแบบสอบถาม (Diet Current) โดยระดับที่ 2 (unhealthy/cheap/too much/random) เป็นระดับสูงสุด จำนวน 60 คน ขณะที่ระดับที่ 1 (healthy/balanced/moderated) รองลงมา 50 คน

#### Reference :

Food choices College students' food and cooking preferences

>> [https://www.kaggle.com/datasets/borapajo/food-choices?select=food\\_coded.csv](https://www.kaggle.com/datasets/borapajo/food-choices?select=food_coded.csv)