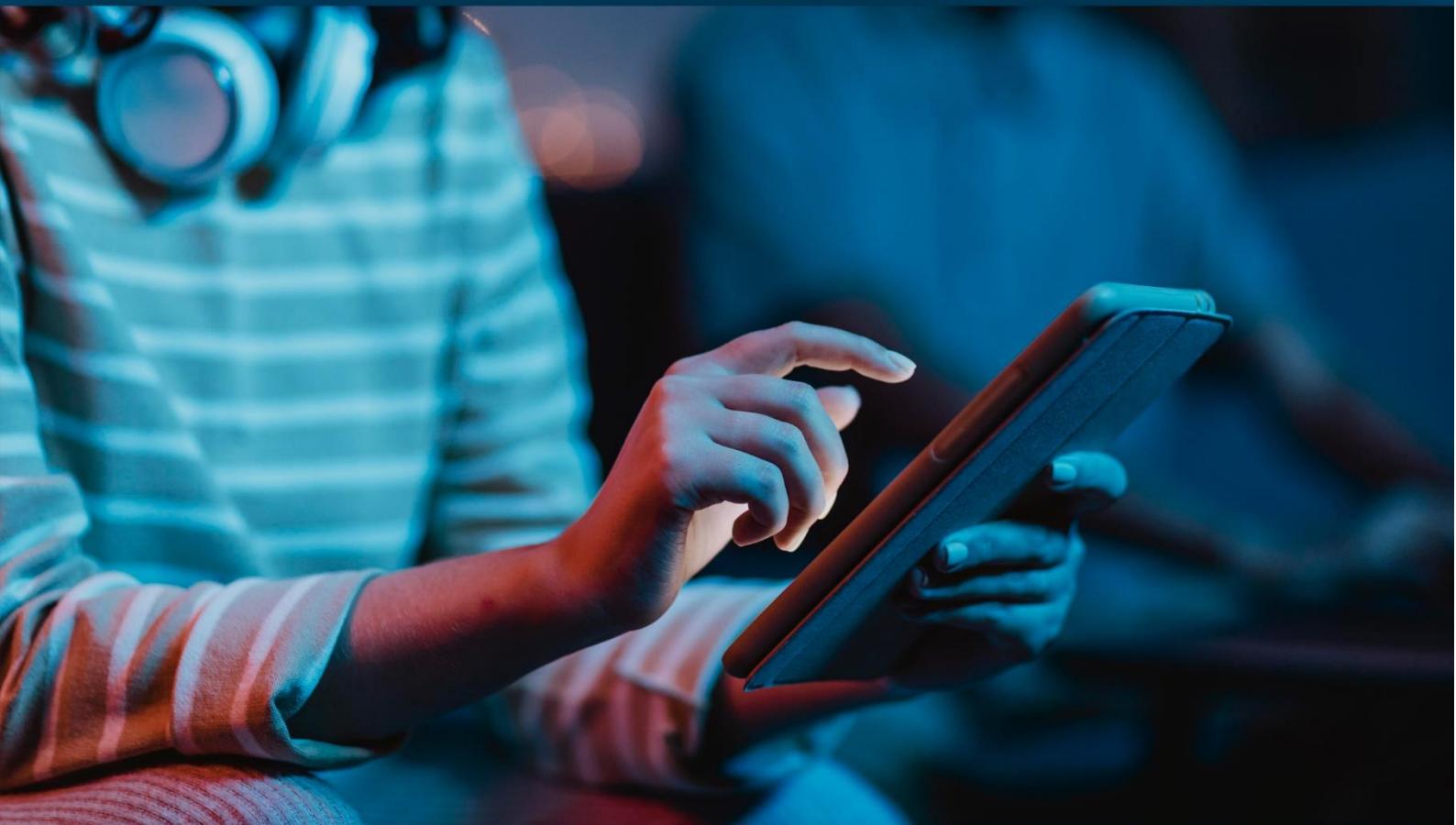
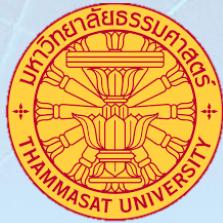


ວຸດສະກອດສົມດ້ານ ເກົດບອນໂຄຍືສາຮານແກຕະດວມພິວເຕວຣ່ ແລະກາຮສົງສາດ





รายงาน

อุตสาหกรรมด้านเทคโนโลยีสารสนเทศ คอมพิวเตอร์ และการสื่อสาร

จัดทำโดย

กลุ่ม ส่งน้อยแต่ส่งนะ

6209680831 นาย ศิริวิชญ์ คล้ายรัตน์

6209610366 นางสาว กนกพร อุบล

6209610028 นางสาว น้ำทนมน เลขะวัฒนะ

เสนอ

ผศ. ดร. ธนาธร ทะนาณทอง

รายงานนี้เป็นส่วนหนึ่งของรายวิชา คพ.246 การแสดงข้อมูล

ปีการศึกษา 2563 / ภาคเรียนที่ 2

สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์

คำนำ

รายงานเล่มนี้จัดทำขึ้นเพื่อเป็นส่วนหนึ่งของรายวิชา คพ.246 การแสดงข้อมูล มีการแสดงกระบวนการ ขั้นตอนการทำ รวมถึงผลลัพธ์จากการทำ data visualization โดยมีจุดประสงค์เพื่อให้เห็นภาพรวมของอุตสาหกรรมในด้านเทคโนโลยีสารสนเทศฯ รวมถึงผลกระทบและบทบาทของเทคโนโลยีสารสนเทศฯ ในสถานการณ์การระบาดของไวรัส COVID-19 โดยเนื้อหาประกอบด้วย ขั้นตอนของการทำ data visualization ตั้งแต่การทำ data preparation , data cleansing รวมทั้งผลลัพธ์หลังจากการ data visualization ได้แก่ สกิลเกี่ยวกับการใช้อินเตอร์เน็ต กิจกรรมที่ใช้ผ่านทางอินเตอร์เน็ต ที่ประกอบด้วย การประชุม/ทำงานออนไลน์, การซื้อสินค้าออนไลน์, การใช้โซเชียลมีเดีย และการชำระเงินอิเล็กทรอนิกส์

ผู้จัดทำคาดหวังเป็นอย่างยิ่งว่าเอกสารฉบับนี้จะมีข้อมูลที่เป็นประโยชน์ต่อผู้อ่านและผู้ที่มีความสนใจเกี่ยวกับการทำ data visualization หรือมีความสนใจในอุตสาหกรรมทางด้านเทคโนโลยีสารสนเทศฯ เพื่อสามารถนำไปศึกษาหรือต่อยอดในอนาคต หากมีข้อเสนอแนะหรือข้อผิดพลาดประการใด ผู้จัดทำขอรับฟังและต้องขออภัยมา ณ ที่นี่

คณะผู้จัดทำ

3 เมษายน 2564

สารบัญ

หัวข้อ	หน้า
1. คำนำ	C
2. สารบัญ	D
3. ทีมและความสำคัญ	1
4. เครื่องมือที่ใช้การรันโค้ด	3
5. เฟรมเวิร์คที่จำเป็นต่อการรันโค้ด	3
6. ชุดข้อมูล	3
7. ขั้นตอนการเตรียมไฟล์ข้อมูล	5
8. Preparation for Data Cleaning	7
9. Data Cleansing	12
10. Data Visualization	24
11. สรุปผล	85
12. บรรณานุกรม	86

ที่มาและความสำคัญ

ในปัจจุบันเทคโนโลยีสารสนเทศฯได้เข้ามามีบทบาทและความสำคัญในการดำเนินชีวิตของมนุษย์เป็นอย่างมาก เนื่องจากสามารถช่วยอำนวยความสะดวกและส่งเสริมคุณภาพชีวิตให้มีประสิทธิภาพที่ดีมากยิ่งขึ้น โดยมีการนำมาประยุกต์ใช้ตั้งแต่การดำเนินชีวิตประจำวัน จนถึงการเพิ่มประสิทธิภาพการทำงานให้แก่องค์กรโดยการใช้เทคโนโลยีสารสนเทศ ในการช่วยจัดการระบบการทำงาน รวมถึงช่วยในการตัดสินใจและประเมินผล ซึ่งจากประโยชน์เหล่านี้จึงส่งผลให้ในปัจจุบันมีการประยุกต์ใช้เทคโนโลยีสารสนเทศร่วมกับอุตสาหกรรมด้านต่างๆอย่างกว้างขวาง

และการรายงาน “DIGITAL 2021” ของ We Are Social และ Hootsuite ได้มีการเปิดเผยว่าในปี 2020 ประเทศไทยมีประชากร 69.88 ล้านคน มีผู้ใช้อินเทอร์เน็ต 48.59 ล้านคน ซึ่งเพิ่มขึ้นจากปี 2019 ถึง 3.4 ล้านคน นอกจากนี้ยังมีการใช้อินเทอร์เน็ตสูงเป็นอันดับ 9 ของโลก โดยประเทศไทยใช้อินเทอร์เน็ตอยู่ที่ประมาณ 8 ชั่วโมง 44 นาทีเฉลี่ยต่อวัน ส่วนทางด้านของแอปพลิเคชัน พบร่วมกับแอปพลิเคชันโซเชียลมีเดียได้รับความนิยมสูงมากที่สุด รองลงมาคือแอปพลิเคชันซื้อสินค้าออนไลน์และ internet banking

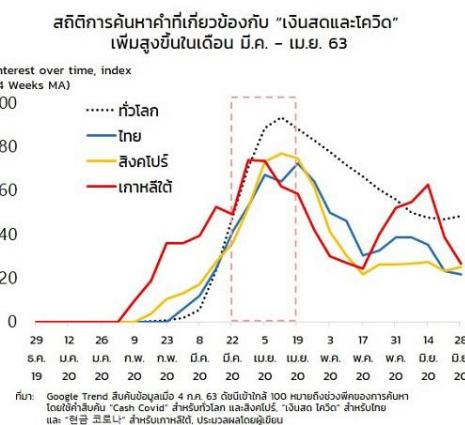
ซึ่งจากสถานการณ์การระบาดของเชื้อไวรัส COVID-19 ที่กระทบต่อชีวิตของผู้คนทั่วโลก ทำให้หลาย ๆ ประเทศรวมถึงประเทศไทยมีการพยายามลดความเสี่ยงในการติดเชื้อด้วยขอความร่วมมือจากประชาชนในการเว้นระยะห่าง (social distancing) ซึ่งวิธีการนี้ทำให้ประชาชนหลายคนถูกจำกัดสิทธิในการทำกิจกรรมต่าง ๆ และการเดินทาง พฤติกรรมหลายอย่างจำเป็นต้องมีการปรับเปลี่ยน ส่งผลให้เกิดความเครียด เสียโอกาสทางด้านอาชีพการทำงาน, การรับความบันเทิง รวมถึงความพึงพอใจในการเข้าถึงสินค้าและบริการ จึงได้มีการนำเทคโนโลยีมาทำการปรับเปลี่ยนรูปแบบใหม่เพื่อรองรับกับสถานการณ์ การแพร่ระบาดของโควิดนี้ให้การดำเนินชีวิตมีความสะดวกขึ้น เช่น ระบบค้าปลีกออนไลน์, การชำระเงินอิเล็กทรอนิกส์, การทำงานที่บ้าน (Work From Home) ด้วยการประชุมระบบออนไลน์ (video conference) และ แพลตฟอร์มโซเชียลมีเดียต่าง ๆ ที่มีความสอดคล้องกับวิถีชีวิตรูปแบบใหม่ หรือ New Normal

ในส่วนของการทำงานที่บ้าน หรือ Work From Home ทำให้เกิดการเปลี่ยนแปลงอย่างมากในระบบการบริหารจัดการองค์กร เนื่องจากต้องปรับพฤติกรรมคนและกลไกทางธุรกิจ รวมทั้งทำการอัพเดต อุปกรณ์และระบบให้มีประสิทธิภาพกับความสามารถที่สูงขึ้น อีกทั้งต้องการสัญญาณ internet/Wi-Fi ที่มีสัญญาณแรงขึ้น มีระบบการเงินที่รวดเร็วคล่องตัวขึ้น (mobile payment) ซึ่งสิ่งเหล่านี้คือโอกาสทางธุรกิจที่จะพัฒนาเทคโนโลยีและเครื่องมืออุปกรณ์ต่าง ๆ รวมถึงการพัฒนาแอปพลิเคชันเฉพาะด้าน, วีดีโอแพลตฟอร์ม, ดิจิทัล คอนเทนต์ และการเรียนรู้แบบออนไลน์ เพื่อให้รองรับความต้องการด้านต่าง ๆ ได้อย่างครอบคลุม

ในด้านโซเชียลมีเดียของประเทศไทยมองว่าโซเชียลมีเดียมีความตัว ซึ่งสะท้อนได้จากผู้ใช้ Facebook YouTube Line ที่มีจำนวน 40-50 ล้านบัญชีในหลายปีที่ผ่านมา ซึ่งค่อนข้างมีความคงที่ โดยบังบัดการเข้าถึงโซเชียลมีเดีย คิดเป็นสัดส่วน 75% ของประชากรทั่วประเทศ และจากการแพร่ระบาดของไวรัส COVID-19 ที่มีการกระตุ้นให้ประชาชนมีการกักตัวอยู่บ้าน ทำให้ประชาชนมีเวลาใช้อินเตอร์เน็ตเพิ่มขึ้น ส่งผลให้การติดต่อสื่อสารทางออนไลน์เข้ามามีบทบาทสำคัญ และโซเชียลมีเดียได้รับความนิยมสูงขึ้น นอกจากนี้ยังเปิดโอกาสให้โซเชียลมีเดียใหม่ๆ เข้ามาสู่ตลาด จากการเปลี่ยนแปลงพฤติกรรมของผู้ใช้เนื่องจากสถานการณ์ดังกล่าว

จากการที่เรามีการเว้นระยะห่างทางสังคม (Social distancing) ส่งผลทำให้การซื้อของออนไลน์ได้รับความนิยมมากขึ้น ซึ่งก็ได้มีข้อมูลที่นำเสนอจากงาน Focal 2020 ในหัวข้อ “Thailand E-Commerce Outlook” มีการเปิดเผยข้อมูลว่าในประเทศไทยมีการเติบโตของ E-Commerce เทียบจากตลาดโลกอยู่ที่ 2% ในปี 2020 และพบว่าคนไทยนิยมซื้อของและสั่งอาหารออนไลน์มากขึ้น โดย 80% ของผู้ใช้บริการครั้งแรกตัดสินใจกลับมาซื้อของในระบบออนไลน์อีกรอบ นอกจากนี้ยังพบว่า แอปพลิเคชัน E-Commerce ยังเป็น 10 อันดับของแอปพลิเคชันที่คนไทยใช้มากที่สุด ซึ่งมูลค่าตลาด E-Commerce ในปี 2019 อยู่ที่ 163,300 ล้านบาท และ เนื่องจากสถานการณ์ COVID-19 ในปี 2020 เป็นตัวเร่งให้เติบโตมากขึ้น โดยในปี 2020 โตขึ้น 35% คือมีมูลค่าในตลาดถึง 220,000 ล้านบาท

ส่วนของการชำระเงินอิเล็กทรอนิกส์ หรือว่า e-Payment ถึงแม้ว่าการใช้ e-Payment ของคนไทยยังไม่เป็นที่นิยมอย่างแพร่หลาย แต่เนื่องจากสถานการณ์ COVID-19 ทำให้มีการกระตุ้นให้คนไทยมีการใช้ e-Payment เพิ่มมากขึ้น สาเหตุส่วนหนึ่งเป็นเพราะต้องการลดการสัมผัสรูปเงินสด โดยกระแสนี้ได้เกิดขึ้นทั่วโลกสะท้อนจากสถิติการค้นหาคำที่เกี่ยวข้องกับ “เงินสดและโควิด” โดยอ้างอิงจากงานศึกษาของธนาคารเพื่อการชำระหนี้ระหว่างประเทศ (BIS) ที่พบว่า การค้นหาเพิ่มสูงขึ้นในเดือน มี.ค. - เม.ย. 63 เพราะเป็นช่วงที่มีการแพร่ระบาดสูงสุด



ເຄີຍມືອທີ່ໃຊ້ກາຮັນໂດດ

1. Python
2. Google Colab

ເພິ່ມເວົ້າກຳທີ່ຈໍາເປັນຕ່ອງກາຮັນໂດດ

1. NumPy
2. Pandas
3. Matplotlib
4. Seaborn

ຊຸດຂໍ້ມູນ

ໂດຍຊຸດຂໍ້ມູນເຮົານຳມາກຳທີ່ຈໍາເປັນຕ່ອງກາຮັນໂດດ ສາມາດແປ່ງຕາມແລ້ວທີ່ມາຂອງຂໍ້ມູນລອກເປັນ 3 ແຫ່ງທີ່ມາໄດ້ແກ່

1. ETDA (ສພຂອ. ອີຣີ ສໍານັກງານພັບນາຫຼຽກຮ່ຽມທາງອີເລັກທຣອນິກສົ່ງ)
 ໄຟລ໌ : ຮາຍງານຜລກາຮັນສໍາວົງພຖືກຮ່ຽມຜູ້ໃຊ້ອິນເຕອົຣິນິຕິໃນປະເທດໄທຢຂອງປີ 2561 ປຶ້ງ 2563
 ປະເກດໄຟລ໌ : PDF

ສາມາດເຂົ້າຄືໄດ້ຈາກ :

ປີ 2018 : <https://www.slideshare.net/ETDAofficialRegist/thailand-internet-user-profile-2018-147392373>

ປີ 2019 : <https://www.etda.or.th/th/Useful-Resource/publications/Thailand-Internet-User-Behavior-2019.aspx>

ປີ 2020 : <https://www.etda.or.th/th/Useful-Resource/publications/Thailand-Internet-User-Behavior-2020.aspx>

2. DataReportal

ໄຟລ໌ : DIGITAL: THAILAND ຂອງປີ 2018 ປຶ້ງ 2021

ປະເກດໄຟລ໌ : PDF

ສາມາດເຂົ້າຄືໄດ້ຈາກ : <https://datareportal.com/digital-in-thailand>

3. ธนาคารแห่งประเทศไทย

ไฟล์ : 1) ปริมาณการชำระเงินผ่านระบบการชำระเงินและช่องทางต่าง ๆ

2) มูลค่าการชำระเงินผ่านระบบการชำระเงินและช่องทางต่าง ๆ

3) ธุรกรรมการชำระเงินผ่านบริการ Mobile banking และ Internet banking

ประเภทไฟล์ : CSV

สามารถเข้าถึงได้จาก :

<https://www.bot.or.th/Thai/Statistics/PaymentSystems/Pages/StatPaymentTransactions.aspx>

รายละเอียดข้อมูล :

1) เป็นสถิติข้อมูลเกี่ยวกับจำนวนครั้งของการชำระเงินผ่านช่องทางต่าง ๆ ตั้งแต่ เดือน

มกราคมปี 2560 ถึง กุมภาพันธ์ปี 2564 โดยมีหน่วยเป็น พันรายการ

2) เป็นสถิติข้อมูลเกี่ยวกับมูลค่าเงินรวมทั้งหมดของการชำระเงินผ่านช่องทางต่าง ๆ ตั้งแต่

เดือนมกราคมปี 2560 ถึง กุมภาพันธ์ปี 2564 โดยมีหน่วยเป็น พันล้านบาท

3) เป็นสถิติข้อมูลเกี่ยวกับรายละเอียดของธุรกรรมการชำระเงินผ่าน Mobile banking และ

Internet Banking ประกอบด้วยข้อมูลจำนวนบัญชี, ปริมาณรายการ(หน่วยพันรายการ) และ

มูลค่าของรายการ(หน่วยพันล้านบาท)

ຂັ້ນຕອນກາຮເຕີຣີມໄຟລ໌ຂໍ້ມູນ

import library ທີ່ເກີຍວ່າຂໍ້ມູນເຂົ້າມາ ໄດ້ແກ່ pandas, numpy, matplotlib ແລະ seaborn ເພື່ອສະດວກຕ່ອກເຮົາໃຊ້ຈຳກັດການ library ໃນກາຍທັງ

```
[ ] import pandas as pd
import numpy as np
from matplotlib import style
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
```

ທຳກາຮເຕີຣີມຂໍ້ມູນໂດຍທຳກາຮດຶງຂໍ້ມູນ dataset ທັ້ງໝາດມາຈາກ google drive ທີ່ມີກາຮເຕີຣີມໄວ້ໂດຍມີຂັ້ນຕອນດັ່ງຕ່ອໄປນີ້

1. import library ທີ່ໃຊ້ໃນກາຮເຂົ້າມື້ງ Google Drive
2. ທຳກາຮຕຽບສອບສິຫຼຸງກາຮເຂົ້າມື້ງ Google Drive
3. Import ໂມດູລມາຍັງ Google Colab ໂດຍກາຮນໍາ URL link ຂອງແຕ່ລະໄຟລ໌ມາເຊື່ອມ
4. ທຳກາຮດຶງຂໍ້ມູນຈາກຊື່ໄຟລ໌ csv. ທີ່ອຸ່ນໃນ google drive ທີ່ເຕີຣີມໄວ້

```
[ ] #1) Install PyDrive
#!pip install -U -q PyDrive
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials
from matplotlib.ticker import MaxNLocator

#2) Authenticate
auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)
```

[import library ທີ່ໃຊ້ໃນກາຮເຂົ້າມື້ງ Google Drive ແລະ ຕຽບສອບສິຫຼຸງກາຮເຂົ້າມື້ງ Google Drive]

```
fileDownloaded = []
fileID = ['1CItxx0GdhMz0zep0krua5Zt4kGQe-19z', '1R5LwrfXhiSsTNiYYxLSTjEdloejv3FJ3', '1hQq9sQVWKHRWB8ffCopeQYWYvNIFNA',
          '1okc_YBTs3IalQ65B6MKGGa0bVQ0pyFA0', '1DI_wpMn1HrB540T8o5Ufcb08-JkHQ_uK', '1LsT00JTwcZCDUKITyfv_y-SZh8RINNFL',
          '1_gmA0Ymm2_u158aq32ifYYAKttDaOPey', '1EPSg10-JDpGic8450VigBt1Fg6UiZxMX', '10TTd9przc4mq_quN6A1c6QqbDE09qMPW',
          '12gFNbZioFXiwN7pNJiwNcCL9C9rvX6D', '122z6lkPW9WJ0KOW-1FDEfd4Ntdm_PDaj', '1gvN26bVcc_Myq1JhA_DGtsLHEHfZbDPj',
          '18VCq-NyFMLQ8PR8nXo73x-xYww7QxZkV', '18Ra4V8Z63DO3NCuHCAPDshewa3N8dmPR']]

for i in range(len(fileID)):
    fileDownloaded.append(drive.CreateFile({'id':fileID[i]}))
```

[Import ໂມດູລມາຍັງ Google Colab]

```
#6) Load the csv
file_name = ['data1.csv', 'data2.csv', 'data3.csv',
            'data4.csv', 'data5.csv', 'data6.csv',
            'data7.csv', 'data8.csv', 'data9.csv',
            'data10.csv', 'data11.csv', 'data_money.csv',
            'data_value.csv', 'data_cash.csv']

for i in range(len(file_name)):
    fileDownloaded[i].GetContentFile(file_name[i])
```

[ຈຶ່ງຂໍ້ມູນຈາກຊື່ໄຟ້ csv. ທີ່ອຳນົດໃນ google drive ທີ່ເຕີມໄວ້]

ນໍາ dataset ທັງໝາດມາແປລັງເກີບໃຫ້ອູ່ໃນຂໍ້ມູນທີ່ມີຮູບແບບຂອງລິສົດ ເພື່ອໃໝ່ງຍ່າຍຕ່ອກເຮັດໃຫ້ກາຍໜັງ ໂດຍຈະເຮັດໃຫ້ຜ່ານ index ທີ່ຂໍ້ມູນນັ້ນໆຖຸກເກີບໄວ້

```
[ ] dataList = [pd.read_csv('data1.csv'), pd.read_csv('data2.csv'), pd.read_csv('data3.csv'), pd.read_csv('data4.csv'),
               pd.read_csv('data5.csv'), pd.read_csv('data6.csv'), pd.read_csv('data7.csv'), pd.read_csv('data8.csv'),
               pd.read_csv('data9.csv'), pd.read_csv('data10.csv'), pd.read_csv('data11.csv'),pd.read_csv('data_money.csv'),
               pd.read_csv('data_value.csv'),pd.read_csv('data_cash.csv')]
```

Preparation for Data Cleaning

```
[ ] for i in range(len(dataList)):
    dataList[i].info()
```

ทำการตรวจสอบว่าข้อมูลที่ได้มามีค่า missing value หรือไม่ โดยจะใช้ฟังก์ชัน info() เพื่อดูรายละเอียดของ dataset

ตัวอย่างบางส่วนของผลลัพธ์ที่ได้

<class 'pandas.core.frame.DataFrame'> RangeIndex: 8 entries, 0 to 7 Data columns (total 2 columns): # Column Non-Null Count Dtype --- 0 Year 8 non-null int64 1 Timeperday 8 non-null int64 dtypes: int64(2) memory usage: 256.0 bytes	<class 'pandas.core.frame.DataFrame'> RangeIndex: 4 entries, 0 to 3 Data columns (total 26 columns): # Column Non-Null Count Dtype --- 0 Year 4 non-null int64 1 Social Media 4 non-null float64 2 Searching Information 4 non-null float64 3 Receive-send E-mail 4 non-null float64 4 Watching TV/Listening Music 4 non-null float64 5 Buying goods and service 4 non-null float64 6 Downloading softwares/songs/movies/games 4 non-null float64 7 Financial transactions 3 non-null float64 8 Playing online games 4 non-null float64 9 E-book 4 non-null float64 10 E-learning 4 non-null float64 11 Finding jobs/Job application 4 non-null float64 12 Booking/buying concert/movie ticket 4 non-null float64 13 Booking/buying ticket 2 non-null float64 14 Selling goods and service 4 non-null float64 15 Watching stock exchange 4 non-null float64 16 Paying taxes 4 non-null float64 17 Booking the hotel 4 non-null float64 18 Receive-send document 4 non-null float64 19 Ordering Foods 4 non-null float64 20 Calling taxi 4 non-null float64 21 Sending massage 3 non-null float64 22 Live application 1 non-null float64 23 Paying goods and service 1 non-null float64 24 Working via online meeting 1 non-null float64 25 Buying insurance 1 non-null float64 memory usage: 416.0 bytes	<class 'pandas.core.frame.DataFrame'> RangeIndex: 2 entries, 0 to 1 Data columns (total 6 columns): # Column Non-Null Count Dtype --- 0 Year 2 non-null int64 1 Shopee 2 non-null float64 2 Lazada 2 non-null float64 3 Facebook Fanpage 2 non-null float64 4 LINE 2 non-null float64 5 Instagram 2 non-null float64 dtypes: float64(5), int64(1) memory usage: 224.0 bytes
<class 'pandas.core.frame.DataFrame'> RangeIndex: 4 entries, 0 to 3 Data columns (total 2 columns): # Column Non-Null Count Dtype --- 0 Year 4 non-null int64 1 User_Number 4 non-null float64 dtypes: float64(1), int64(1) memory usage: 192.0 bytes	<class 'pandas.core.frame.DataFrame'> RangeIndex: 9 entries, 0 to 8 Data columns (total 3 columns): # Column Non-Null Count Dtype --- 0 Year 9 non-null int64 1 Platform 9 non-null object 2 Rate 9 non-null float64 dtypes: float64(1), int64(1), object(1) memory usage: 344.0+ bytes	<class 'pandas.core.frame.DataFrame'> RangeIndex: 60 entries, 0 to 59 Data columns (total 4 columns): # Column Non-Null Count Dtype --- 0 Year 60 non-null int64 1 Platform 60 non-null object 2 Rate 60 non-null float64 3 Type 56 non-null object dtypes: float64(1), int64(1), object(2) memory usage: 2.0+ KB

```
[ ] for i in range(len(dataList)):
    print(dataList[i].isnull().sum())
    print()
```

ทำการหาว่า DataFrame ของแต่ละdataset มีค่าmissing valueจำนวนกี่ค่า โดยใช้ฟังก์ชัน isnull() หรือ pandas.DataFrame.isnull() เพื่อหาว่าข้อมูลใดเป็นmissing value และใช้ฟังก์ชัน sum() ต่อท้ายฟังก์ชัน isnull() เพื่อทำการหาค่าผลรวมของmissing value ที่พบ ทำได้ค่าmissing value ออกมากอยู่ในรูปแบบของจำนวนเลข

ຜລລັບທີ່ດີ

Year	0	ກົມກາ	0	ກົມກາ	0	ກົມກາ	0	
Timeperday	0	Youtube	0	ກ.ພ.	2564 p	0	ກ.ພ.	2564 p
dtype: int64		Facebook	0	ມ.ຄ.	2564 p	0	ມ.ຄ.	2564 p
Year	0	LINE	0	ສ.ຮ.	2563 p	0	ສ.ຮ.	2563 p
Z_Weekday	0	Instagram	0	ນ.ວ.	2563 p	0	ນ.ວ.	2563
X_Weekends	0	Pantip	0	ວ.ຊ.	2563 p	0	ວ.ຊ.	2563
Y_Weekday	0	Twitter	0	ສ.ຣ.	2563 p	0	ສ.ຣ.	2563
Y_Weekends	0	WhatsApp	0	ນ.ຣ.	2563 p	0	ນ.ຣ.	2563
X_Weekday	0	Facebook Messenger	2	ນ.ບ.	2563 p	3	ນ.ບ.	2563 r
BabyB_Weekday	0	TikTok	2	ວ.ນ.	2563 p	3	ວ.ນ.	2563 r
BabyB_Weekends	0	Tinder	2	ພ.ບ.	2563 p	3	ພ.ບ.	2563 r
dtype: int64		dtype: int64		ມ.ບ.	2563 p	3	ມ.ບ.	2563 r
Year	0	Year	0	ກ.ວ.	2563 p	3	ກ.ວ.	2563 r
Student	0	Shopee	0	ມ.ວ.	2563 p	3	ມ.ວ.	2563 r
Freelance	0	Lazada	0	ສ.ວ.	2562 p	3	ສ.ວ.	2562 r
Sel-employed	0	Facebook Fanpage	0	ວ.ດ.	2562 p	3	ວ.ດ.	2562 r
Unemployed	0	LINE	0	ພ.ດ.	2562 p	3	ພ.ດ.	2562 r
Official	0	Instagram	0	ສ.ດ.	2562 p	3	ສ.ດ.	2562 r
Maid	0	dtype: int64		ນ.ດ.	2562 p	3	ນ.ດ.	2562 r
dtype: int64		Year	0	ວ.ນ.	2562 p	3	ວ.ນ.	2562 r
Year	0	User_Number	0	ມ.ນ.	2562 p	3	ມ.ນ.	2562 r
Home	0	dtype: int64		ກ.ນ.	2562 p	3	ກ.ນ.	2562 r
Workplace	0	Platform	0	ສ.ນ.	2561 p	3	ສ.ນ.	2561 r
School	0	Rate	0	ວ.ນ.	2561 p	3	ວ.ນ.	2561 r
Publicplace	0	dtype: int64		ພ.ນ.	2561 p	3	ພ.ນ.	2561 r
On_the_way	0	Year	0	ກ.ດ.	2561 p	3	ກ.ດ.	2561 r
Internet_cafe	0	Watching	0	ສ.ດ.	2561 p	3	ສ.ດ.	2561 r
dtype: int64		TV/listening	0	ວ.ດ.	2561 p	3	ວ.ດ.	2561 r
Year	0	Music	0	ພ.ດ.	2561 p	3	ພ.ດ.	2561 r
Social_Media	0	Buying goods and service	0	ສ.ດ.	2561 p	3	ສ.ດ.	2561 r
Searching_Information	0	Downloading softwares/songs/movies/games	0	ວ.ດ.	2561 p	3	ວ.ດ.	2561 r
Receive-send_E-mail	0	Financial transactions	1	ພ.ດ.	2561 p	3	ພ.ດ.	2561 r
Watching_TV/listening_Music	0	Playing online games	0	ສ.ດ.	2561 p	3	ສ.ດ.	2561 r
Buying goods and service	0	Type	4	ວ.ດ.	2561 p	3	ວ.ດ.	2561 r
Downloadig softwares/songs/movies/games	0	dtype: int64		ພ.ດ.	2561 p	3	ພ.ດ.	2561 r
Financial transactions	1	Year	0	ກ.ດ.	2561 p	3	ກ.ດ.	2561 r
Playing online games	0	Platform	0	ສ.ດ.	2561 p	3	ສ.ດ.	2561 r
E-learning	0	Rate	0	ວ.ດ.	2561 p	3	ວ.ດ.	2561 r
Finding jobs/Job application	0	dtype: int64		ພ.ດ.	2561 p	3	ພ.ດ.	2561 r
Booking/buying concert/movie ticket	0	Year	0	ສ.ດ.	2561 p	3	ສ.ດ.	2561 r
Booking/buying ticket	2	Messenger	0	ວ.ດ.	2560 p	3	ວ.ດ.	2560 r
Selling goods and service	0	Social Network	2	ມ.ດ.	2560 p	3	ມ.ດ.	2560 r
Watching stock exchange	0	Video	0	ວ.ດ.	2560 p	3	ວ.ດ.	2560 r
Paying taxes	0	Music	2	ມ.ດ.	2560 p	3	ມ.ດ.	2560 r
Booking the hotel	0	Game	0	ວ.ດ.	2560 p	3	ວ.ດ.	2560 r
Receive-send document	0	Shopping	2	ມ.ດ.	2560 p	3	ມ.ດ.	2560 r
Ordering Foods	0	Map	0	ວ.ດ.	2560 p	3	ວ.ດ.	2560 r
Calling taxi	0	Banking	0	ມ.ດ.	2560 p	3	ມ.ດ.	2560 r
Sending message	1	Health and Fitness	2	ວ.ດ.	2560 p	3	ວ.ດ.	2560 r
Live application	1	Dating	2	ວ.ດ.	2560 p	3	ວ.ດ.	2560 r
Paying goods and service	3	dtype: int64		ວ.ດ.	2560 p	3	ວ.ດ.	2560 r
Working via online meeting	3			ວ.ດ.	2560 p	3	ວ.ດ.	2560 r
Buying insurance	3			ວ.ດ.	2560 p	3	ວ.ດ.	2560 r
dtype: int64				dtype: int64			dtype: int64	

ເຫັນໄດ້ວ່າຄ່າmissing value ທີ່ໄດ້ອອກມາ ມີຄ່າທີ່ເປັນ 0 ແລະ ໄມເປັນ 0 ອູ້ດ້ວຍ ຜົງທາກຄ່າmissing value ເທົກນັບ 0 ດື່ນ ຂອລັນນັ້ນໄມ້ຂໍ້ມູນລົງທຶນທີ່ເປັນmissing value ແຕ່ທາກຄ່າmissing value ໄມເປັນ 0 ພໍາຍຄວາມວ່າ ຂອລັນນັ້ນມີຂໍ້ມູນລົງທຶນທີ່ເປັນ missing value ອູ້ ດັ່ງນັ້ນເຮົາຈຶ່ງພບວ່າມີ missing value ອູ້ໃນ dataList[4] , dataList[5] , dataList[9] , dataList[10] ແລະ dataList[11]

[] dataList[4]

Year	Social Media	Searching Information	Receive-send E-mail	Watching TV/listening Music	Buying goods and service	Downloadig softwares/songs/movies/games	Financial transactions	Playing online games	E-book learning	E-learning	Finding jobs/Job application	Booking/buying concert/movie ticket
0	2017.0	88.9	88.5	70.5	60.7	50.8	47.4	46.2	35.3	30.8	28.3	18.0
1	2018.0	93.6	70.8	74.2	60.7	51.3	43.2	49.2	41.0	48.3	28.5	13.9
2	2019.0	91.2	70.7	82.5	71.2	57.0	28.4	NaN	34.1	57.1	24.4	13.2
3	2020.0	95.3	82.2	69.0	85.0	67.3	53.0	56.5	58.8	64.2	57.5	28.1
												34.3
Booking/buying ticket	Selling goods and service	Watching stock exchange	Paying taxes	Booking the hotel	Receive-send document	Ordering Foods	Calling taxi	Sending message	Live application	Paying goods and service	Working via online meeting	Buying insurance
14.0	13.7	12.9	11.1	11.0	7.7	6.8	4.8	NaN	NaN	NaN	NaN	NaN
19.3	24.5	15.1	14.1	20.7	7.2	11.4	12.1	50.6	31.6	NaN	NaN	NaN
NaN	15.9	12.6	20.1	25.9	18.2	26.5	21.4	50.0	29.6	60.6	NaN	NaN
NaN	26.6	14.4	25.7	30.8	39.0	49.8	32.0	77.8	40.4	NaN	38.4	16.4

ກໍານົດສະແດງຂໍ້ມູນໃນ dataList[4] ອອກມາ

[] dataList[4].isnull

	Year	Social Media	...	Working via online meeting	Buying insurance
0	2017.0	86.9	...	NaN	NaN
1	2018.0	93.6	...	NaN	NaN
2	2019.0	91.2	...	NaN	NaN
3	2020.0	95.3	...	36.4	16.4

[4 rows x 26 columns]>

ทำการแสดงຂໍ້ມູນໃນ dataList[4] ທີ່ມີຄ່າເປັນmissing value ອອກມາ ໂດຍໃຊ້ພັກໜ້າ isnull

[] dataList[5]

	Year	Youtube	Facebook	LINE	Instagram	Pantip	Twitter	WhatsApp	Facebook Messenger	TikTok	Tinder
0	2017	97.1	96.6	95.8	56.0	54.7	27.6	12.1	NaN	NaN	NaN
1	2018	98.8	96.0	98.6	67.2	64.2	43.0	10.6	88.4	NaN	NaN
2	2020	97.5	98.2	96.0	80.4	30.6	71.9	6.7	NaN	35.8	6.0

ทำการแสดงຂໍ້ມູນໃນ dataList[5] ອອກມາ

dataList[5].isnull

	Year	Youtube	Facebook	LINE	...	WhatsApp	Facebook Messenger	TikTok	Tinder
0	2017	97.1	96.6	95.8	...	12.1	NaN	NaN	NaN
1	2018	98.8	96.0	98.6	...	10.6	88.4	NaN	NaN
2	2020	97.5	98.2	96.0	...	6.7	NaN	35.8	6.0

[3 rows x 11 columns]>

ทำการแสดงຂໍ້ມູນໃນ dataList[5] ທີ່ມີຄ່າເປັນmissing value ອອກມາ ໂດຍໃຊ້ພັກໜ້າ isnull

[] dataList[9]

Year	Platform	Rate	Type	30	2019	Line	85.0	Messenger/Chat App/VOIP	31	2019	Messenger	70.0	Messenger/Chat App/VOIP	
0	2017	Facebook	75.0	Social Network	31	2019	Messenger	70.0	Messenger/Chat App/VOIP	32	2019	Instagram	65.0	Social Network
1	2017	Youtube	72.0	Social Network	33	2019	Twitter	55.0	Social Network	34	2019	Tiktok	38.0	NaN
2	2017	Line	68.0	Messenger/Chat App/VOIP	35	2019	Pinterest	32.0	Social Network	36	2019	LinkedIn	27.0	Social Network
3	2017	Messenger	55.0	Messenger/Chat App/VOIP	37	2019	Wechat	27.0	Messenger/Chat App/VOIP	38	2019	WhatsApp	27.0	Messenger/Chat App/VOIP
4	2017	Instagram	50.0	Social Network	39	2019	Twitch	25.0	Social Network	40	2019	Skype	25.0	Messenger/Chat App/VOIP
5	2017	Google+	45.0	Social Network	41	2019	Snapchat	23.0	Social Network	42	2019	Reddit	20.0	NaN
6	2017	Twitter	38.0	Social Network	43	2019	Tumblr	19.0	Social Network	44	2020	Youtube	94.2	Social Network
7	2017	Skype	22.0	Messenger/Chat App/VOIP	45	2020	Facebook	93.3	Social Network	46	2020	Line	88.2	Messenger/Chat App/VOIP
8	2017	Linkedin	18.0	Social Network	47	2020	Messenger	77.1	Messenger/Chat App/VOIP	48	2020	Instagram	64.2	Social Network
9	2017	Pinterest	17.0	Social Network	49	2020	Twitter	57.8	Social Network	50	2020	Tiktok	54.8	NaN
10	2017	WhatsApp	17.0	Messenger/Chat App/VOIP	51	2020	Pinterest	31.7	Social Network	52	2020	Wechat	24.9	Messenger/Chat App/VOIP
11	2017	Wechat	17.0	Messenger/Chat App/VOIP	53	2020	Twitch	24.5	Social Network	54	2020	LinkedIn	24.4	Social Network
12	2018	Facebook	93.0	Social Network	55	2020	WhatsApp	24.4	Messenger/Chat App/VOIP	56	2020	Skype	22.6	Messenger/Chat App/VOIP
13	2018	Youtube	91.0	Social Network	57	2020	Snapchat	19.8	Social Network	58	2020	Badoo	18.5	NaN
14	2018	Line	84.0	Messenger/Chat App/VOIP	59	2020	Vimeo	18.0	Social Network					
15	2018	Messenger	72.0	Messenger/Chat App/VOIP										
16	2018	Instagram	65.0	Social Network										
17	2018	Twitter	52.0	Social Network										
18	2018	Skype	27.0	Messenger/Chat App/VOIP										
19	2018	Linkedin	25.0	Social Network										
20	2018	Pinterest	25.0	Social Network										
21	2018	Wechat	25.0	Messenger/Chat App/VOIP										
22	2018	WhatsApp	25.0	Messenger/Chat App/VOIP										
23	2018	Twitch	24.0	Social Network										
24	2018	Snapchat	20.0	Social Network										
25	2018	Sina Weibo	19.0	Social Network										
26	2018	Tumber	19.0	Social Network										
27	2018	Vimeo	19.0	Social Network										
28	2019	Facebook	94.0	Social Network										
29	2019	Youtube	94.0	Social Network										

ทำการแสดงຂໍ້ມູນໃນ dataList[9] ອອກມາ

```
[ ] dataList[9].loc[dataList[9]['Type'].isnull()]
```

	Year	Platform	Rate	Type
34	2019	Tiktok	36.0	NaN
42	2019	Reddit	20.0	NaN
50	2020	Tiktok	54.8	NaN
58	2020	Badoo	18.5	NaN

ทำการแสดงเฉพาะค่าที่เป็นmissing value ของ dataList[9] เท่านั้น โดยใช้ method .loc ในการดึงข้อมูลที่ต้องการออกมานี้ เราทำการดึงข้อมูลมาเฉพาะคอลัมน์ Type ใน dataList[9] และใช้ฟังก์ชัน isnull() ต่อท้าย เพื่อให้ได้ในค่าที่เป็นmissing value ในคอลัมน์ Type ออกมานี้

```
[ ] dataList[10]
```

	Year	Messenger	Social Network	Video	Music	Game	Shopping	Map	Banking	Health and Fitness	Dating
0	2017	77.0		Nan	75.0	Nan	66.0	Nan	64.0	56.0	Nan
1	2018	95.0		Nan	95.0	Nan	85.0	Nan	80.0	74.0	Nan
2	2019	95.0		97.0	89.0	65.0	67.0	58.0	72.0	65.0	30.0
3	2020	97.4		97.6	87.8	65.6	68.6	83.4	71.6	68.1	32.4

ทำการแสดงข้อมูลใน dataList[10] ออกมานี้

```
[ ] dataList[10].isnull
```

```
<bound method DataFrame.isnull of>
0 2017    77.0      Nan ...   56.0      Nan      Nan
1 2018    95.0      Nan ...   74.0      Nan      Nan
2 2019    95.0    97.0 ...   65.0      30.0    12.0
3 2020    97.4    97.6 ...   68.1      32.4    13.4
[4 rows x 11 columns]>
```

ทำการแสดงข้อมูลใน dataList[10] ที่มีค่าเป็นmissing value ออกมานี้ โดยใช้ฟังก์ชัน isnull

[] dataList[11]

ຮາມການ	ດ.ພ. 2564 ປ.	ມ.ຄ. 2564 ປ.	ສ.ພ. 2563 ປ.	ວ.ພ. 2563 ປ.	ດ.ພ. 2563 ປ.	ມ.ຄ. 2563 ປ.	ວ.ພ. 2563 ປ.	ດ.ພ. 2563 ປ.	ມ.ຄ. 2563 ປ.	ວ.ພ. 2563 ປ.	ດ.ພ. 2563 ປ.	ມ.ຄ. 2563 ປ.	ວ.ພ. 2563 ປ.	ດ.ພ. 2562 ປ.	ມ.ຄ. 2562 ປ.	ວ.ພ. 2562 ປ.	ດ.ພ. 2562 ປ.	ມ.ຄ. 2562 ປ.	ວ.ພ. 2562 ປ.				
ຈຳນວນ ອົງການບັນຊີ ລັດວັນຍົດ	Nan																						
ຈຳນວນ ອົງການ ບັນຊີ ນີ້ໃຫຍ່ ມີຄວາມ ສຳເນົາ ການ	20.71	20.08	20.08	19.98	19.82	19.82	19.90	19.91	19.85	19.83	19.80	19.74	19.60	19.48	19.34	19.14	19.00	18.75	18.47	18.30	18.15	18.10	18.01
ຈຳນວນ ອົງການ ບັນຊີ ນີ້ໃຫຍ່ ມີຄວາມ ສຳເນົາ ການ	90.50	89.38	87.71	85.77	84.12	83.03	81.78	80.91	79.58	78.09	76.17	74.43	72.72	71.39	70.00	67.98	66.15	64.21	62.45	63.57	72.45	72.35	71.57
ສົ່ງເຖິງ ສົ່ງຈຳນວນ ອົງການ	111.22	109.44	107.77	105.73	104.04	102.85	101.88	100.81	99.43	97.92	95.97	94.17	92.32	90.84	89.34	87.10	85.16	82.98	80.93	81.87	90.80	90.44	89.58
ມູນຄວາມ ບັນຊີ(ລົງທະບຽນ)	Nan	Nan	Nan	Nan																			
ມູນຄວາມ ບັນຊີ(ລົງທະບຽນ ທີ່ມີມີ ສຳເນົາ)	2,901.51	3,053.33	3,352.65	2,783.24	2,665.54	2,652.70	2,308.82	5,188.67	2,529.23	2,348.88	2,013.50	4,121.53	4,504.84	4,017.23	4,837.84	4,325.92	5,349.77	14,788.82	14,370.23	4,963.88	5,341.89	10,278.81	6,990.89
ມູນຄວາມ ບັນຊີ(ນັບ ຈຳນວນ ການ)	26,109.95	26,613.59	29,160.41	27,455.51	25,768.97	24,106.49	23,271.40	23,278.30	23,250.76	22,258.08	19,602.47	20,884.84	18,057.24	19,007.24	19,802.70	18,551.85	17,530.81	17,073.79	16,224.88	16,131.19	14,914.37	15,550.68	14,888.03
ສົ່ງເຖິງ ສົ່ງຜົນ ການ	29,071.48	29,888.92	32,503.08	30,219.75	28,432.50	28,659.19	25,578.23	28,482.97	25,779.99	24,804.06	21,615.97	24,988.38	22,581.88	23,024.47	24,840.54	22,877.77	22,880.58	31,080.61	30,595.11	21,096.07	20,266.28	25,829.47	21,888.92
ມູນຄວາມ ບັນຊີ(ລົງທະບຽນ)	Nan	Nan	Nan	Nan																			
ນັບຈຳນວນ ການ	2,578.84	2,663.32	2,569.99	2,239.93	2,234.07	2,286.08	2,246.91	4,900.03	2,165.92	2,095.16	2,089.26	2,707.84	3,659.26	4,083.25	4,414.58	3,747.33	3,805.49	12,672.42	12,038.92	3,288.98	3,770.98	9,133.74	6,373.38
ມູນຄວາມ ບັນຊີ(ນັບ ຈຳນວນ ການ)	28,486.94	28,707.89	29,107.73	27,492.33	25,673.40	24,220.32	23,515.87	23,251.02	23,483.14	22,383.95	20,564.73	19,790.44	18,128.98	18,824.00	19,730.81	18,388.08	17,084.15	17,012.12	16,173.78	16,180.93	15,070.91	15,411.95	15,433.99
ສົ່ງເຖິງ ສົ່ງຜົນ ການ	29,042.79	29,371.21	31,877.72	29,702.28	27,907.47	28,488.40	28,782.79	28,151.06	26,629.06	24,459.11	22,683.98	22,468.07	21,688.21	22,907.25	24,145.37	22,138.41	20,889.84	29,684.54	28,210.70	19,487.89	18,841.87	24,545.69	21,807.35

ກໍານົດສະເໜີຂອໍມູນໃນ dataList[11] ອອກມາ

dataList[11].isnull

```
<bound method DataFrame.isnull of
0      ຈຳນວນປົດ/ບັນຊີ (ລົງທະບຽນ) 2/ ...      Nan
1      ຈຳນວນປົດ/ບັນຊີ(ຜູ້ໃຫຍ່ເກົາກົດທີ່ມີມີສຳເນົາ) ...      2.11
2      ຈຳນວນປົດ/ບັນຊີ(ຜູ້ໃຫຍ່ເກົາກົດທີ່ມີມີສຳເນົາ) ...      37.72
3      ຮວມທັງສັນ(ຈຳນວນປົດ/ບັນຊີ) ...      39.83
4      ມູນຄວາມບັນຊີ(ລົງທະບຽນ) ...      NaN
5      ມູນຄວາມບັນຊີ(ນັບຈຳນວນການ) ...      709.75
6      ມູນຄວາມບັນຊີ(ນັບຈຳນວນການ) ...      8,768.57
7      ຮວມທັງສັນ(ມູນຄວາມບັນຊີ) ...      9,478.33
8      ມູນຄວາມໃຊ້ຈໍາຍ (ລົງທະບຽນ) ...      NaN
9      ມູນຄວາມໃຊ້ຈໍາຍ(ຜູ້ໃຫຍ່ເກົາກົດທີ່ມີມີສຳເນົາ) ...      698.40
10     ມູນຄວາມໃຊ້ຈໍາຍ(ຜູ້ໃຫຍ່ເກົາກົດທີ່ມີມີສຳເນົາ) ...      8,839.02
11     ຮວມທັງສັນ(ມູນຄວາມໃຊ້ຈໍາຍ) ...      9,537.42
```

รายงาน ຢຸ່ໂມ. ສ. 2560

[12 rows x 51 columns]>

ກໍານົດສະເໜີຂອໍມູນໃນ dataList[11] ທີ່ມີຄ່າເປັນmissing value ອອກມາ ໂດຍໃຊ້ພັກໜັນ isnull

Data Cleansing

หากเราพบว่ามีค่า missing value อยู่ใน DataFrame เราจะต้องทำการ data cleansing เพื่อให้ค่าข้อมูลที่เป็น missing value ให้หายไป ซึ่งเราได้ทำการ data cleansing ข้อมูล ทั้งหมด 2 วิธีการ คือ

- ใช้ฟังก์ชัน `fillna()` ที่ใช้ในการค่าmissing value ซึ่งเราได้ทำการแทนที่ค่า missing value ด้วยเครื่องหมาย “-”(dash)
- ใช้ฟังก์ชัน `dropna()` ในการดรอปข้อมูลทั้งแถวทั้ง เนื่องจากdatasetที่เราทำการดรอปนั้นจำเป็นต้องมีข้อมูลทุกจำนวนขาดข้อมูลใดไปไม่ได้ โดยหากใส่ `axis=0` หรือไม่ใส่คำสั่งในวงเล็บใน `dropna()` หมายถึง การดรอปแถว(row) แต่หากใส่ `axis=1` ในวงเล็บใน `dropna()` หมายถึง การดรอปคอลัมน์(column)

```

dataArray[4]['Social Media'] = dataArray[4]['Social Media'].fillna('-')
dataArray[4]['Searching Information'] = dataArray[4]['Searching Information'].fillna('-')
dataArray[4]['Receive-send E-mail'] = dataArray[4]['Receive-send E-mail'].fillna('-')
dataArray[4]['Watching TV/Listening Music'] = dataArray[4]['Watching TV/Listening Music'].fillna('-')
dataArray[4]['Buying goods and service'] = dataArray[4]['Buying goods and service'].fillna('-')
dataArray[4]['Downloading softwares/songs/movies/games'] = dataArray[4]['Downloading softwares/songs/movies/games'].fillna('-')
dataArray[4]['Financial transactions'] = dataArray[4]['Financial transactions'].fillna('-')
dataArray[4]['Playing online games'] = dataArray[4]['Playing online games'].fillna('-')
dataArray[4]['E-book'] = dataArray[4]['E-book'].fillna('-')
dataArray[4]['E-learning'] = dataArray[4]['E-learning'].fillna('-')
dataArray[4]['Finding jobs/job application'] = dataArray[4]['Finding jobs/job application'].fillna('-')
dataArray[4]['Booking/buying concert/movie ticket'] = dataArray[4]['Booking/buying concert/movie ticket'].fillna('-')
dataArray[4]['Booking/buying ticket'] = dataArray[4]['Booking/buying ticket'].fillna('-')

dataArray[4]['selling goods and service'] = dataArray[4]['selling goods and service'].fillna('-')
dataArray[4]['Watching stock exchange'] = dataArray[4]['Watching stock exchange'].fillna('-')
dataArray[4]['Paying taxes'] = dataArray[4]['Paying taxes'].fillna('-')
dataArray[4]['Booking the hotel'] = dataArray[4]['Booking the hotel'].fillna('-')
dataArray[4]['Receive-send document'] = dataArray[4]['Receive-send document'].fillna('-')
dataArray[4]['Ordering Foods'] = dataArray[4]['Ordering Foods'].fillna('-')
dataArray[4]['Calling taxi'] = dataArray[4]['Calling taxi'].fillna('-')
dataArray[4]['Sending massage'] = dataArray[4]['Sending massage'].fillna('-')

dataArray[4]['Live application'] = dataArray[4]['Live application'].fillna('-')
dataArray[4]['Paying goods and service'] = dataArray[4]['Paying goods and service'].fillna('-')
dataArray[4]['Working via online meeting'] = dataArray[4]['Working via online meeting'].fillna('-')
dataArray[4]['Buying insurance'] = dataArray[4]['Buying insurance'].fillna('-')

```

ทำการแทนที่ค่า missing value ใน `dataArray[4]` ด้วยเครื่องหมาย “-”(dash) โดยใช้ฟังก์ชัน `fillna('-')`

[] `dataArray[4]`

Year	Social Media	Searching Information	Receive-send E-mail	Watching TV/Listening Music	Buying goods and service	Downloading softwares/songs/movies/games	Financial transactions	Playing online games	E-book	E-learning	Finding jobs/job application	Booking/buying concert/movie ticket	Booking/buying ticket
0	2017.0	86.9	86.5	70.5	60.7	50.8	47.4	45.2	35.3	30.8	28.3	16.0	14.6
1	2018.0	93.6	70.8	74.2	60.7	51.3	43.2	49.2	41.0	48.3	28.5	13.9	21.7
2	2019.0	91.2	70.7	62.5	71.2	57.0	28.4	-	34.1	57.1	24.4	13.2	26.1
3	2020.0	95.3	82.2	69.0	85.0	67.3	53.0	50.5	56.8	64.2	57.5	28.1	34.3

Selling goods and service	Watching stock exchange	Paying taxes	Booking the hotel	Receive-send document	Ordering Foods	Calling taxi	Sending massage	Live application	Paying goods and service	Working via online meeting	Buying insurance
13.7	12.9	11.1	11.0	7.7	6.8	4.8	-	-	-	-	-
24.5	15.1	14.1	20.7	7.2	11.4	12.1	50.6	31.6	-	-	-
15.9	12.6	20.1	25.9	18.2	26.5	21.4	50	29.6	60.6	-	-
26.6	14.4	25.7	30.6	39.0	49.8	32.0	77.8	40.4	-	36.4	16.4

ทำการแสดงข้อมูลใน `dataArray[4]` จะเห็นได้ว่าค่าที่เป็น missing value จะถูกแทนที่ด้วยเครื่องหมาย “-”

```
[ ] dataList[5]=dataList[5].dropna(axis=1)
```

ทำการดรอปคอลัมน์ที่มีค่าเป็น missing value ใน dataList[5] ด้วยฟังก์ชัน dropna(axis=1)

```
[ ] dataList[5]
```

Before

	Year	Youtube	Facebook	LINE	Instagram	Pantip	Twitter	WhatsApp	Facebook	Messenger	TikTok	Tinder
0	2017	97.1	98.6	95.8	58.0	54.7	27.6	12.1		NaN	NaN	NaN
1	2018	98.8	98.0	98.6	67.2	64.2	43.0	10.6		88.4	NaN	NaN
2	2020	97.5	98.2	96.0	80.4	30.6	71.9	6.7		NaN	35.8	6.0

After

	Year	Youtube	Facebook	LINE	Instagram	Pantip	Twitter	WhatsApp
0	2017	97.1	96.6	95.8	56.0	54.7	27.6	12.1
1	2018	98.8	96.0	98.6	67.2	64.2	43.0	10.6
2	2020	97.5	98.2	96.0	80.4	30.6	71.9	6.7

ทำการแสดงข้อมูลใน dataList[5] จะเห็นได้ว่าข้อมูลในคอลัมน์ที่มีค่าเป็น missing value จะถูกดรอปออกไป

```
[ ] dataList[9] = dataList[9].dropna()
```

ทำการดรอปแถวที่มีค่าเป็น missing value ใน dataList[9] ด้วยฟังก์ชัน dropna()

```
[ ] dataList[9]
```

Before

	Year	Platform	Rate	Type							
0	2017	Facebook	75.0	Social Network	26	2018	Tumber	19.0	Social Network		
1	2017	Youtube	72.0	Social Network	27	2018	Vladeo	19.0	Social Network		
2	2017	Line	88.0	Messenger/Chat App/VOIP	28	2019	Facebook	94.0	Social Network		
3	2017	Messenger	55.0	Messenger/Chat App/VOIP	29	2019	Youtube	94.0	Social Network		
4	2017	Instagram	50.0	Social Network	30	2019	Line	85.0	Messenger/Chat App/VOIP		
5	2017	Google+	45.0	Social Network	31	2019	Messenger	76.0	Messenger/Chat App/VOIP		
6	2017	Twitter	38.0	Social Network	32	2019	Instagram	65.0	Social Network		
7	2017	Skype	22.0	Messenger/Chat App/VOIP	33	2019	Twitter	55.0	Social Network		
8	2017	LinkedIn	18.0	Social Network	34	2019	Tiktok	38.0	NaN		
9	2017	Pinterest	17.0	Social Network	35	2019	Pintere	32.0	Social Network		
10	2017	Whatapp	17.0	Messenger/Chat App/VOIP	36	2019	LinkedIn	27.0	Social Network		
11	2017	Wechat	17.0	Messenger/Chat App/VOIP	37	2019	Wechat	27.0	Messenger/Chat App/VOIP		
12	2018	Facebook	93.0	Social Network	38	2019	Whatapp	27.0	Messenger/Chat App/VOIP		
13	2018	Youtube	91.0	Social Network	39	2019	Twitch	25.0	Social Network		
14	2018	Line	84.0	Messenger/Chat App/VOIP	40	2019	Skype	25.0	Messenger/Chat App/VOIP		
15	2018	Messenger	72.0	Messenger/Chat App/VOIP	41	2019	Snapchat	23.0	Social Network		
16	2018	Instagram	65.0	Social Network	42	2019	Reddit	20.0	NaN		
17	2018	Twitter	52.0	Social Network	43	2019	Tumber	19.0	Social Network		
18	2018	Skype	27.0	Messenger/Chat App/VOIP	44	2020	Youtube	94.2	Social Network		
19	2018	LinkedIn	25.0	Social Network	45	2020	Facebook	63.3	Social Network		
20	2018	Pinterest	25.0	Social Network	46	2020	Line	88.2	Messenger/Chat App/VOIP		
21	2018	Wechat	25.0	Messenger/Chat App/VOIP	47	2020	Messenger	77.1	Messenger/Chat App/VOIP		
22	2018	Whatapp	25.0	Messenger/Chat App/VOIP	48	2020	Instagram	84.2	Social Network		
23	2018	Twitch	24.0	Social Network	49	2020	Twitter	57.8	Social Network		
24	2018	Snapchat	20.0	Social Network	50	2020	Tiktok	54.8	NaN		
25	2018	Sina Weibo	19.0	Social Network	51	2020	Pintere	31.7	Social Network		

After

Year	Platform	Rate	Type
0	2017	Facebook	75.0
1	2017	Youtube	72.0
2	2017	Line	68.0 Messenger/Chat App/VOIP
3	2017	Messenger	55.0 Messenger/Chat App/VOIP
4	2017	Instagram	50.0 Social Network
5	2017	Google+	45.0 Social Network
6	2017	Twitter	38.0 Social Network
7	2017	Skyper	22.0 Messenger/Chat App/VOIP
8	2017	LinkedIn	18.0 Social Network
9	2017	Pinterest	17.0 Social Network
10	2017	WhatsApp	17.0 Messenger/Chat App/VOIP
11	2017	Wechat	17.0 Messenger/Chat App/VOIP
12	2018	Facebook	93.0 Social Network
13	2018	Youtube	91.0 Social Network
14	2018	Line	84.0 Messenger/Chat App/VOIP
15	2018	Messenger	72.0 Messenger/Chat App/VOIP
16	2018	Instagram	65.0 Social Network
17	2018	Twitter	52.0 Social Network
18	2018	Skyper	27.0 Messenger/Chat App/VOIP
19	2018	LinkedIn	25.0 Social Network
20	2018	Pinterest	25.0 Social Network
21	2018	Wechat	25.0 Messenger/Chat App/VOIP
22	2018	WhatsApp	25.0 Messenger/Chat App/VOIP
23	2018	Twitch	24.0 Social Network
24	2018	Snapchat	20.0 Social Network
25	2018	Sina Weibo	19.0 Social Network
26	2018	Tumber	19.0 Social Network
27	2018	Vladeo	19.0 Social Network
28	2019	Facebook	94.0 Social Network
29	2019	Youtube	94.0 Social Network
30	2019	Line	85.0 Messenger/Chat App/VOIP
31	2019	Messenger	78.0 Messenger/Chat App/VOIP
32	2019	Instagram	65.0 Social Network
33	2019	Twitter	55.0 Social Network
35	2019	Pinterest	32.0 Social Network
36	2019	LinkedIn	27.0 Social Network
37	2019	Wechat	27.0 Messenger/Chat App/VOIP
38	2019	WhatsApp	27.0 Messenger/Chat App/VOIP
39	2019	Twitch	25.0 Social Network
40	2019	Skyper	25.0 Messenger/Chat App/VOIP
41	2019	Snapchat	23.0 Social Network
43	2019	Tumber	19.0 Social Network
44	2020	Youtube	94.2 Social Network
45	2020	Facebook	93.3 Social Network
46	2020	Line	88.2 Messenger/Chat App/VOIP
47	2020	Messenger	77.1 Messenger/Chat App/VOIP
48	2020	Instagram	64.2 Social Network
49	2020	Twitter	57.8 Social Network
51	2020	Pinterest	31.7 Social Network
52	2020	Wechat	24.9 Messenger/Chat App/VOIP
53	2020	Twitch	24.5 Social Network
54	2020	LinkedIn	24.4 Social Network
55	2020	WhatsApp	24.4 Messenger/Chat App/VOIP
56	2020	Skyper	22.8 Messenger/Chat App/VOIP
57	2020	Snapchat	19.8 Social Network
59	2020	Vladeo	18.0 Social Network

ทำการแสดงข้อมูลใน dataList[9] เห็นได้ว่าข้อมูลในแต่ละค่าที่มีค่าเป็น missing value จะถูกครอบคลุมไป

```
dataList[10]['Messenger'] = dataList[10]['Messenger'].fillna('-')
dataList[10]['Social Network'] = dataList[10]['Social Network'].fillna('-')
dataList[10]['Video'] = dataList[10]['Video'].fillna('-')
dataList[10]['Music'] = dataList[10]['Music'].fillna('-')
dataList[10]['Game'] = dataList[10]['Game'].fillna('-')
dataList[10]['Messenger'] = dataList[10]['Messenger'].fillna('-')
dataList[10]['Shopping'] = dataList[10]['Shopping'].fillna('-')
dataList[10]['Map'] = dataList[10]['Map'].fillna('-')
dataList[10]['Banking'] = dataList[10]['Banking'].fillna('-')
dataList[10]['Health and Fitness'] = dataList[10]['Health and Fitness'].fillna('-')
dataList[10]['Dating'] = dataList[10]['Dating'].fillna('-')
```

ทำการแทนที่ค่า missing value ใน dataList[10] ด้วยเครื่องหมาย “-(dash) โดยใช้ฟังก์ชัน fillna(' -)'

dataList[10]

Before

	Year	Messenger	Social Network	Video	Music	Game	Shopping	Map	Banking	Health and Fitness	Dating
0	2017	77.0	NaN	75.0	NaN	66.0	NaN	64.0	56.0	NaN	NaN
1	2018	95.0	NaN	95.0	NaN	85.0	NaN	80.0	74.0	NaN	NaN
2	2019	95.0	97.0	89.0	65.0	67.0	58.0	72.0	65.0	30.0	12.0
3	2020	97.4	97.6	87.8	65.6	68.6	83.4	71.6	68.1	32.4	13.4

After

Year	Messenger	Social Network	Video	Music	Game	Shopping	Map	Banking	Health and Fitness	Dating
0	2017	77.0	-	75.0	-	66.0	-	64.0	56.0	-
1	2018	95.0	-	95.0	-	85.0	-	80.0	74.0	-
2	2019	95.0	97	89.0	65	67.0	58	72.0	65.0	30 12
3	2020	97.4	97.6	87.8	65.6	68.6	83.4	71.6	68.1	32.4 13.4

ทำการแสดงข้อมูลใน dataList[10] จะเห็นได้ว่าค่าที่เป็นmissing value จะถูกแทนที่ด้วยเครื่องหมาย “-”

```
[ ] dataList[11]=dataList[11].dropna()
```

ทำการดรอปແກ່ທີ່ມີຄ່າເປັນ missing value ໃນ dataList[11] ດ້ວຍພັງກັນ dropna() ເນື່ອຈາກແກ່ທີ່ทำการดรອບເປັນເພີ່ມຂໍ້ອ້ວຂ້ອຂອງຂໍ້ມູນ ໄນໄດ້ເປັນຂໍ້ມູນທີ່ຈໍາເປັນຕ້ອງແສດງອອກມາຈຶ່ງทำการດຽບຂໍ້ມູນທີ່ແກ່ວທີ່

dataList[11]

Before

ລາຍການ	ຄ.ນ.	ມ.ນ.	ດ.ນ.	ວ.ນ.	ດ.ນ.	ນ.ນ.	ດ.ນ.	ນ.ນ.	ມ.ນ.	ວ.ນ.	ດ.ນ.	ນ.ນ.	ດ.ນ.	ນ.ນ.	ດ.ນ.	ວ.ນ.	ດ.ນ.	ນ.ນ.	ມ.ນ.	ວ.ນ.	ດ.ນ.	ນ.ນ.	ມ.ນ.
0 ລົງທະບຽນ (ລົງທະບຽນ ລົງທະບຽນ)	NaN																						
1 ຈຳກັດ ສົ່ງເອົາ ໃຫຍ່ເຊີ້ມ ສົ່ງເອົາ	20.71	20.08	20.06	19.96	19.92	19.82	19.90	19.91	19.85	19.83	19.80	19.74	19.80	19.46	19.34	19.14	19.00	18.76	18.47	18.30	18.15	18.10	18.01
2 ສົ່ງເອົາ ສົ່ງເອົາ ສົ່ງເອົາ	90.50	89.38	87.71	85.77	84.12	83.03	81.78	80.91	79.58	78.09	78.17	74.43	72.72	71.39	70.00	67.98	66.15	64.21	62.45	63.57	72.45	72.35	71.57
3 ສົ່ງເອົາ ສົ່ງເອົາ ສົ່ງເອົາ	111.22	109.44	107.77	105.73	104.04	102.85	101.68	100.81	99.43	97.02	95.97	94.17	92.32	90.84	89.34	87.10	85.16	82.99	80.93	81.87	90.80	90.44	89.58
4 ລົງທະບຽນ (ລົງທະບຽນ ລົງທະບຽນ)	NaN																						
5 ລົງທະບຽນ (ລົງທະບຽນ ລົງທະບຽນ)	2,801.51	3,053.33	3,352.65	2,763.24	2,865.54	2,682.70	2,306.82	5,186.67	2,629.23	2,346.88	2,013.50	4,121.53	4,504.64	4,017.23	4,837.84	4,325.92	5,349.77	14,788.82	14,370.23	4,963.88	5,341.89	10,278.81	5,990.89
6 ລົງທະບຽນ (ລົງທະບຽນ ລົງທະບຽນ ລົງທະບຽນ)	28,180.95	28,613.59	29,150.41	27,458.51	28,788.97	24,106.49	23,271.40	23,278.30	23,250.78	22,258.08	19,802.47	20,884.84	18,057.24	19,007.24	19,802.70	18,651.88	17,530.81	17,073.79	16,224.88	16,131.19	14,914.37	15,550.08	14,888.03
7 ສົ່ງເອົາ ສົ່ງເອົາ ສົ່ງເອົາ	29,071.48	29,686.92	32,503.08	30,219.75	28,432.50	28,659.19	25,578.23	28,482.97	25,779.99	24,804.95	21,815.97	24,988.38	22,561.88	23,024.47	24,840.54	22,877.77	22,880.58	31,880.51	30,595.11	21,095.07	20,259.28	25,829.47	21,858.92
8 ລົງທະບຽນ (ລົງທະບຽນ ລົງທະບຽນ)	NaN																						
9 ລົງທະບຽນ (ລົງທະບຽນ ລົງທະບຽນ ລົງທະບຽນ)	2,570.84	2,663.32	2,569.99	2,239.03	2,234.07	2,206.08	2,246.91	4,900.03	2,165.92	2,095.16	2,089.28	2,707.84	3,559.25	4,083.25	4,414.56	3,747.33	3,805.49	12,672.42	12,036.92	3,286.98	3,770.96	9,133.74	6,373.36
10 ລົງທະບຽນ (ລົງທະບຽນ ລົງທະບຽນ ລົງທະບຽນ)	28,485.94	28,707.89	29,107.73	27,482.33	25,673.40	24,220.32	23,281.02	23,483.14	22,983.95	20,694.73	19,790.44	18,128.98	18,824.00	19,730.81	18,388.08	17,084.15	17,012.12	16,173.78	16,180.93	15,070.91	15,411.95	15,433.99	
11 ສົ່ງເອົາ ການໂດຍ ການໂດຍ	29,042.79	29,371.21	31,877.72	29,702.28	27,907.47	28,486.40	25,762.79	28,151.06	25,829.08	24,459.11	22,683.98	22,498.07	21,688.21	22,907.25	24,145.37	20,889.64	28,684.54	28,210.70	19,487.89	18,841.87	24,545.69	21,807.35	

After

ລາຍການ	ຄ.ວ.	2564 p	ຄ.ວ.	2564 p	ຄ.ວ.	2563 p	ຄ.ວ.	2562 p	ຄ.ວ.	2562 p	ຄ.ວ.	2562	ຄ.ວ.	2562	ຄ.ວ.	2562										
1 ສົມບັດ ຄົມບັດ ເມືອງບັດ ບັດ	20.71	20.08	20.00	19.90	19.92	19.82	19.00	19.91	19.85	19.83	19.80	19.74	19.00	19.40	19.34	19.14	19.00	18.75	18.47	18.30	18.10	18.10	18.01			
2 ສົມບັດ ຄົມບັດ ເມືອງບັດ ບັດ	90.50	89.38	87.71	85.77	84.12	83.03	81.78	80.91	79.58	78.09	78.17	74.43	72.72	71.39	70.00	67.98	66.15	64.21	62.45	63.57	72.45	72.35	71.57			
3 ສົມບັດ ຄົມບັດ ເມືອງບັດ ບັດ	111.22	106.44	107.77	105.73	104.04	102.85	101.68	100.81	99.43	97.92	95.97	94.17	92.32	90.84	89.34	87.10	85.16	82.98	80.93	81.87	80.80	80.44	80.58			
5 ໄກສາບັດ ຄົມບັດ ເມືອງບັດ ບັດ	2,901.51	3,053.33	3,382.05	2,793.24	2,885.84	2,652.70	2,308.92	5,188.87	2,829.23	2,348.88	2,013.50	4,121.53	4,504.84	4,017.23	4,637.84	4,328.92	5,349.77	14,788.82	14,370.23	4,963.88	5,341.89	10,278.81	0,990.89			
6 ໄກສາບັດ ຄົມບັດ ເມືອງບັດ ບັດ	28,169.95	26,813.59	29,150.41	27,459.51	25,788.97	24,106.49	23,271.40	23,278.30	23,250.76	22,258.08	19,602.47	20,884.84	18,057.24	19,007.24	19,802.70	18,551.88	17,530.81	17,073.79	18,224.88	18,131.19	14,914.37	15,550.88	14,988.03			
7 ໄກສາບັດ ຄົມບັດ ເມືອງບັດ ບັດ	28,071.45	26,988.92	32,503.08	30,219.75	28,432.50	26,656.19	25,578.23	28,402.07	25,779.99	24,804.95	21,615.97	24,988.38	22,551.88	23,024.47	24,840.54	22,877.77	22,880.58	31,850.81	30,565.11	21,065.07	20,256.26	25,829.47	21,858.92			
9 ໄກສາບັດ ຄົມບັດ ເມືອງບັດ ບັດ	2,578.84	2,093.32	2,599.99	2,239.93	2,234.07	2,286.08	2,248.91	4,900.03	2,165.92	2,096.10	2,089.29	2,707.64	3,559.25	4,083.25	4,414.58	3,747.33	3,805.49	12,072.42	12,036.92	3,288.99	3,770.99	9,133.74	0,373.30			
10 ໄກສາບັດ ຄົມບັດ ເມືອງບັດ ບັດ	28,458.94	28,707.89	29,107.73	27,462.33	25,673.40	24,220.32	23,515.87	23,251.02	23,463.14	22,383.05	20,594.73	19,790.44	18,128.98	18,824.00	19,730.81	18,388.08	17,084.15	17,012.12	16,173.78	16,180.03	15,070.01	15,411.95	15,433.00			
14 ໄກສາບັດ ຄົມບັດ ເມືອງບັດ ບັດ	29,042.79	29,371.21	31,677.72	29,702.26	27,907.47	26,408.40	25,782.79	28,151.06	25,620.08	24,459.11	22,683.98	22,468.07	21,688.21	22,007.25	24,145.87	22,185.41	20,888.64	26,684.54	23,210.70	19,467.69	18,841.87	24,546.89	21,807.36			

ทำการแสดงข้อมูลใน dataList[11] เห็นได้ว่าข้อมูลในแຄוที่มีค่าเป็น missing value จะถูกกรอปออกไป

```
col = np.array(dataList[11].columns)
new_col_name = np.array(['List', '2021-02', '2021-01', '2020-12', '2020-11', '2020-10', '2020-09', '2020-08', '2020-07', '2020-06', '2020-05', '2020-04', '2020-03', '2020-02', '2020-01', '2019-12', '2019-11', '2019-10', '2019-09', '2019-08', '2019-07', '2019-06', '2019-05', '2019-04', '2019-03', '2019-02', '2019-01', '2018-12', '2018-11', '2018-10', '2018-09', '2018-08', '2018-07', '2018-06', '2018-05', '2018-04', '2018-03', '2018-02', '2018-01', '2017-12', '2017-11', '2017-10', '2017-09', '2017-08', '2017-07', '2017-06', '2017-05', '2017-04', '2017-03', '2017-02', '2017-01'])
```

ทำการสร้างอาร์เรย์ใหม่ด้วย numpy.array โดยให้ อาร์เรย์ col เก็บข้อมูลที่เป็นคอลัมน์ของ dataList[10] ส่วน อาร์เรย์ new_col_name ใช้เก็บข้อมูลชื่อที่จะนำไปเปลี่ยนชื่อคอลัมน์ใหม่ใน dataList[10]

```
for i in range(len(col)):
    dataList[11] = dataList[11].rename(columns={col[i]:new_col_name[i]})
```

ทำการเปลี่ยนชื่อคอลัมน์ด้วยอาร์เรย์ โดยการใช้ฟังก์ชัน rename และทำการเปลี่ยนชื่อด้วยการเทียบระหว่างอาร์เรย์เก่าที่เป็นชื่อเก่า(col) กับ อาร์เรย์ที่เป็นชื่อใหม่(new_col_name)

dataList[11]

List	2021-02	2021-01	2020-12	2020-11	2020-10	2020-09	2020-08	2020-07	2020-06	2020-05	2020-04	2020-03	2020-02	2020-01	2019-12	2019-11	2019-10	2019-09	2019-08
1 ຈ່າຍງານ ນິ້ນກຳທີ່ ເປັນ ສາມາດຢືນ ກ...	20.71	20.08	20.06	19.96	19.92	19.82	19.90	19.91	19.85	19.83	19.80	19.74	19.60	19.46	19.34	19.14	19.00	18.75	18.47
2 ມີມູນຄົງໃຫ້ ກຳທີ່ ໄດ້ກຳທີ່ ໄດ້ມູນຄົງ ກ...	90.50	89.36	87.71	85.77	84.12	83.03	81.78	80.91	79.58	78.09	76.17	74.43	72.72	71.39	70.00	67.96	66.15	64.21	62.45
3 ສິນຈຳລັງ ນິ້ນ ນິ້ນ	111.22	109.44	107.77	105.73	104.04	102.85	101.68	100.81	99.43	97.92	95.97	94.17	92.32	90.84	89.34	87.10	85.16	82.96	80.93
5 ມຸນຄົງກາງ ເພີ້ມເປົ້າ ໃຫ້ກຳທີ່ ທີ່ເປົ້າ ສາມາດຢືນ ກ...	2,901.51	3,053.33	3,352.65	2,763.24	2,665.54	2,552.70	2,306.82	5,186.67	2,529.23	2,346.88	2,013.50	4,121.53	4,504.64	4,017.23	4,837.84	4,325.92	5,349.77	14,786.82	14,370.23
6 ມຸນຄົງກາງ ເພີ້ມເປົ້າ ໃຫ້ກຳທີ່ ທີ່ເປົ້າ ສາມາດຢືນ ກ...	26,169.95	26,613.59	29,150.41	27,456.51	25,766.97	24,106.49	23,271.40	23,276.30	23,250.76	22,258.08	19,602.47	20,864.84	18,057.24	19,007.24	19,802.70	18,551.85	17,530.81	17,073.79	16,224.88
7 ສິນຈຳລັງ ກາງເສີມ ເວັບ	29,071.46	29,666.92	32,503.06	30,219.75	28,432.50	26,659.19	25,578.23	28,462.97	25,779.99	24,604.95	21,615.97	24,986.36	22,561.88	23,024.47	24,640.54	22,877.77	22,880.58	31,860.61	30,595.11
9 ມຸນຄົງກາງ ເຈົ້າຍົງ ໃຫ້ກຳທີ່ ທີ່ເຈົ້າຍົງ ສາມາດຢືນ ກ...	2,576.84	2,663.32	2,569.99	2,239.93	2,234.07	2,266.08	2,246.91	4,900.03	2,165.92	2,095.16	2,089.26	2,707.64	3,559.25	4,083.25	4,414.56	3,747.33	3,805.49	12,672.42	12,036.92
10 ມຸນຄົງກາງ ເຈົ້າຍົງ ໃຫ້ກຳທີ່ ທີ່ເຈົ້າຍົງ ສາມາດຢືນ ກ...	26,465.94	26,707.89	29,107.73	27,462.33	25,673.40	24,220.32	23,515.87	23,251.02	23,463.14	22,363.95	20,594.73	19,790.44	18,128.96	18,824.00	19,730.81	18,388.08	17,084.15	17,012.12	16,173.78
11 ສິນມຸນຄົງ ກາງໄຊ ຈາກ	29,042.79	29,371.21	31,677.72	29,702.26	27,907.47	26,486.40	25,762.79	28,151.06	25,629.06	24,459.11	22,683.98	22,498.07	21,688.21	22,907.25	24,145.37	22,135.41	20,889.64	29,684.54	28,210.70

ทำการแสดงข้อมูลใน dataList[11] หลังจากการทำการเปลี่ยนชื่อคอลัมน์แล้ว

```
dataList[11].index = np.arange(len(dataList[11].index))
```

```
ind = dataList[11]['List']
new_ind_name = np.array(['Number_accounts_bank', 'Number_accounts_non_bank', 'Total_Number_accounts',
                        'Topup_value_bank', 'Topup_value_non_bank', 'Total_Topup_value',
                        'Spending_value_bank', 'Spending_value_non_bank', 'Total_Spending_value'])
```

ทำการสร้างอาร์เรย์ใหม่ด้วย numpy.array โดยให้ อาร์เรย์ ind เก็บข้อมูลที่เป็นแค่วง List ใน dataList[11] ส่วน อาร์เรย์ new_ind_name ใช้เก็บข้อมูลชื่อที่จะนำไปเปลี่ยนชื่อแคล้วใหม่ใน dataList[11] ในคอลัมน์Type

```
[ ] for i in range(len(ind)):
    dataList[11]['List'][i] = new_ind_name[i]
```

ทำการเปลี่ยนชื่อคอลัมน์ด้วยอาร์เรย์ โดยเปลี่ยนชื่อจากการนำอาร์เรย์ที่เป็นชื่อใหม่ (new_ind_name) ไปใส่ในdataList[11]['List'] ตามในแต่ละindex

▶ dataList[11]

	List	2021-02	2021-01	2020-12	2020-11	2020-10	2020-09	2020-08	2020-07	2020-06	2020-05	2020-04	2020-03	2020-02
0	Number_accounts_bank	20.71	20.08	20.06	19.96	19.92	19.82	19.90	19.91	19.85	19.83	19.80	19.74	19.60
1	Number_accounts_non_bank	90.50	89.36	87.71	85.77	84.12	83.03	81.78	80.91	79.58	78.09	76.17	74.43	72.72
2	Total_Number_accounts	111.22	109.44	107.77	105.73	104.04	102.85	101.68	100.81	99.43	97.92	95.97	94.17	92.32
3	Topup_value_bank	2,901.51	3,053.33	3,352.65	2,763.24	2,665.54	2,552.70	2,306.82	5,186.67	2,529.23	2,346.88	2,013.50	4,121.53	4,504.64
4	Topup_value_non_bank	26,169.95	26,613.59	29,150.41	27,456.51	25,766.97	24,106.49	23,271.40	23,276.30	23,250.76	22,258.08	19,602.47	20,864.84	18,057.24
5	Total_Topup_value	29,071.46	29,666.92	32,503.06	30,219.75	28,432.50	26,659.19	25,578.23	28,462.97	25,779.99	24,604.95	21,615.97	24,986.36	22,561.88
6	Spending_value_bank	2,576.84	2,663.32	2,569.99	2,239.93	2,234.07	2,266.08	2,246.91	4,900.03	2,165.92	2,095.16	2,089.26	2,707.64	3,559.25
7	Spending_value_non_bank	26,465.94	26,707.89	29,107.73	27,462.33	25,673.40	24,220.32	23,515.87	23,251.02	23,463.14	22,363.95	20,594.73	19,790.44	18,128.96
8	Total_Spending_value	29,042.79	29,371.21	31,677.72	29,702.26	27,907.47	26,486.40	25,762.79	28,151.06	25,629.06	24,459.11	22,683.98	22,498.07	21,688.21

ทำการแสดงข้อมูลใน dataList[11] หลังจากการทำการเปลี่ยนชื่อแคล้วแล้ว

```
new_df11 = dataList[11].groupby('List', sort=False).sum()
```

```
for i in range(len(new_df11.values)):
    for j in range(len(new_df11.values[i])):
        new_df11.values[i][j] = new_df11.values[i][j].replace(',', '')
new_df11.values[:2]
```

```
array([[ '20.71', '20.08', '20.06', '19.96', '19.92', '19.82', '19.90',
       '19.91', '19.85', '19.83', '19.80', '19.74', '19.60', '19.46',
       '19.34', '19.14', '19.00', '18.75', '18.47', '18.30', '18.15',
       '18.10', '18.01', '17.97', '17.80', '17.73', '17.65', '14.51',
       '11.51', '11.93', '11.82', '8.13', '7.97', '7.77', '7.56',
       '6.81', '3.55', '2.87', '2.58', '2.39', '2.09', '2.12', '2.08',
       '2.04', '2.06', '2.09', '2.08', '2.10', '2.11'],
      [ '90.50', '89.36', '87.71', '85.77', '84.12', '83.03', '81.78',
       '80.91', '79.58', '78.09', '76.17', '74.43', '72.72', '71.39',
       '70.00', '67.96', '66.15', '64.21', '62.45', '63.57', '72.45',
       '72.35', '71.57', '70.71', '70.87', '76.62', '74.10', '73.33',
       '66.99', '66.11', '65.14', '62.41', '60.37', '58.76', '57.43',
       '55.78', '54.30', '51.83', '51.69', '50.26', '48.68', '46.81',
       '44.58', '43.40', '42.37', '41.24', '40.01', '39.22', '38.30',
       '37.72]], dtype=object)
```

ทำการลบเครื่องหมาย ','(comma) ออก โดยการใช้ฟังก์ชัน replace

```
new_df11 = new_df11.astype('float')
new_df11
```

	List	2021-02	2021-01	2020-12	2020-11	2020-10	2020-09	2020-08	2020-07	2020-06	2020-05	2020-04	2020-03
	Number_accounts_bank	20.71	20.08	20.06	19.96	19.92	19.82	19.90	19.91	19.85	19.83	19.80	19.74
	Number_accounts_non_bank	90.50	89.36	87.71	85.77	84.12	83.03	81.78	80.91	79.58	78.09	76.17	74.43
	Total_Number_accounts	111.22	109.44	107.77	105.73	104.04	102.85	101.68	100.81	99.43	97.92	95.97	94.17
	Topup_value_bank	2901.51	3053.33	3352.65	2763.24	2665.54	2552.70	2306.82	5186.67	2529.23	2346.88	2013.50	4121.53
	Topup_value_non_bank	26,169.95	26,613.59	29,150.41	27,456.51	25,766.97	24,106.49	23,271.40	23,276.30	23,250.76	22,258.08	19,602.47	20,864.84
	Total_Topup_value	29,071.46	29,666.92	32,503.06	30,219.75	28,432.50	26,659.19	25,578.23	28,462.97	25,779.99	24,604.95	21,615.97	24,986.36
	Spending_value_bank	2,576.84	2,663.32	2,569.99	2,239.93	2,234.07	2,266.08	2,246.91	4,900.03	2,165.92	2,095.16	2,089.26	2,707.64
	Spending_value_non_bank	26,465.94	26,707.89	29,107.73	27,462.33	25,673.40	24,220.32	23,515.87	23,251.02	23,463.14	22,363.95	20,594.73	19,790.44
	Total_Spending_value	29,042.79	29,371.21	31,677.72	29,702.26	27,907.47	26,486.40	25,762.79	28,151.06	25,629.06	24,459.11	22,683.98	22,498.07

ทำการแปลงค่าข้อมูลจากString ให้กลายเป็นfloat โดยใช้ฟังก์ชัน astype('float')

```
col = np.array(dataList[12].columns)
new_col_name = np.array(['List', '2021-02', '2021-01', '2020-12', '2020-11', '2020-10', '2020-09', '2020-08', '2020-07', '2020-06', '2020-05', '2020-04', '2020-03', '2020-02', '2020-01',
'2019-12', '2019-11', '2019-10', '2019-09', '2019-08', '2019-07', '2019-06', '2019-05', '2019-04', '2019-03', '2019-02', '2019-01',
'2018-12', '2018-11', '2018-10', '2018-09', '2018-08', '2018-07', '2018-06', '2018-05', '2018-04', '2018-03', '2018-02', '2018-01',
'2017-12', '2017-11', '2017-10', '2017-09', '2017-08', '2017-07', '2017-06', '2017-05', '2017-04', '2017-03', '2017-02', '2017-01'])
```

ทำการสร้างอาร์เรย์ใหม่ด้วย numpy.array โดยให้ อาร์เรย์ col เก็บข้อมูลที่เป็นคอลัมน์ ใน dataList[12] ส่วน อาร์เรย์ new_col_name ใช้เก็บข้อมูลชื่อที่จะนำไปเปลี่ยนชื่อคอลัมน์ใหม่ใน dataList[12]

```
for i in range(len(col)):
    dataList[12] = dataList[12].rename(columns={col[i]:new_col_name[i]})
```

```
[ ] ind = dataList[12]['List']
new_ind_name = np.array(['Check', 'Cheak_Inhouse', 'Cheak_Interbank', 'e-Payments', 'transfer_BAHTNET',
'transfer_Bulk_Payment', 'transfer_Direct_credit', 'transfer_Direct_debit', 'transfer_ITMX Bulk Payment', 'Online_Retail_Funds_Transfer',
'Transfer_sub_cash_ATM', 'Transfer_sub_cash_Internet_and_mobile', 'Transfer_sub_cash_across_bank_by_branch', 'Intra_bank_transfer', 'transfer_payment_by_ATM',
'transfer_payment_internet_smartphone', 'Payment_cards', 'Debit_card', 'Credit_card', 'E-Money', 'Total'])
```

ทำการสร้างอาร์เรย์ใหม่ด้วย numpy.array โดยให้ อาร์เรย์ ind เก็บข้อมูลที่เป็นแต่ละของ คอลัมน์List ในdataList[12] ส่วน อาร์เรย์ new_ind_name ใช้เก็บข้อมูลชื่อที่จะนำไปเปลี่ยนชื่อเดาของ คอลัมน์Listใหม่ใน dataList[12]

```
[ ] for i in range(len(ind)):
    dataList[12]['List'][i] = new_ind_name[i]
```

ทำการเปลี่ยนชื่อคอลัมน์ด้วยอาร์เรย์ โดยเปลี่ยนชื่อจากการนำอาร์เรย์ที่เป็นชื่อใหม่(new_ind_name)ไปใส่ในdataList[12]['List'] ตามในแต่ละindex

	List	2021-02	2021-01	2020-12	2020-11	2020-10	2020-09	2020-08	2020-07	2020-06	2020-05	2020-04	2020-03	2020-02	2020-01	2019-12	2019-11	2019-10	2019-09	2019-08
0	Check	2,941	3,299	4,069	3,449	3,097	3,297	3,467	3,317	3,451	3,079	3,332	4,318	3,808	4,168	4,652	4,306	4,250	4,465	4,256
1	Cheak_Inhouse	980	1,162	1,679	1,260	1,082	1,159	1,298	1,135	1,135	1,085	1,179	1,489	1,295	1,403	1,725	1,700	1,469	1,563	1,456
2	Cheak_Interbank	1,961	2,137	2,390	2,189	2,015	2,138	2,169	2,182	2,316	1,994	2,153	2,829	2,513	2,765	2,927	2,606	2,781	2,902	2,800
3	e-Payments	33,159	36,741	37,566	35,570	34,334	36,471	35,705	37,064	38,701	34,180	38,386	42,440	35,336	37,897	36,920	33,360	33,791	35,038	35,627
4	transfer_BAHTNET	23,927	27,280	27,668	26,412	25,896	28,043	27,529	28,711	30,421	26,427	30,653	33,296	27,754	29,829	28,860	25,846	25,781	27,341	27,957
5	transfer_Bulk_Payment	2,880	2,880	3,017	2,916	2,564	2,634	2,452	2,618	2,865	2,700	2,907	3,369	2,515	2,774	2,574	2,585	2,899	2,840	2,752
6	transfer_Direct_credit	2,009	1,898	1,959	1,920	1,720	1,733	1,595	1,719	1,748	1,610	1,907	2,289	1,618	1,826	1,679	1,696	2,003	1,846	1,727
7	transfer_Direct_debit	568	657	679	642	537	544	550	599	722	632	582	746	592	604	538	586	582	620	706
8	transfer_ITMX Bulk Payment	332	325	379	354	307	357	307	300	395	458	418	334	305	344	357	303	314	374	319
9	Online_Retail_Funds_Transfer	2,219	2,196	2,277	2,001	1,909	1,830	1,863	1,815	1,682	1,571	1,355	1,559	1,426	1,424	1,484	1,323	1,298	1,216	1,214
10	Transfer_sub_cash_ATM	22	23	27	23	23	23	23	24	24	23	22	27	27	28	33	30	31	30	32
11	Transfer_sub_cash_Internet_and_mobile	2,190	2,165	2,242	1,970	1,878	1,799	1,832	1,783	1,650	1,541	1,326	1,522	1,390	1,386	1,441	1,283	1,257	1,176	1,172
12	Transfer_sub_cash_across_bank_by_branch	7	8	8	8	8	8	8	8	8	7	7	10	9	10	10	10	10	10	10
13	Intra_bank_transfer	3,974	4,222	4,385	4,057	3,795	3,798	3,688	3,748	3,567	3,345	3,357	4,050	3,459	3,660	3,752	3,400	3,611	3,441	3,500
14	transfer_payment_by_ATM	54	56	63	55	56	55	52	54	52	52	50	69	68	72	84	75	78	77	85
15	transfer_payment_internet_smartphone	3,920	4,166	4,322	4,002	3,739	3,743	3,636	3,694	3,515	3,293	3,307	3,981	3,391	3,588	3,668	3,325	3,533	3,364	3,415

ทำการแสดงข้อมูลใน dataList[12] หลังจากการทำการเปลี่ยนชื่อแล้วและคอลัมน์แล้ว

```
[ ] new_df12 = dataList[12].groupby('List', sort=False).sum()

[ ] for i in range(len(new_df12.values)):
    for j in range(len(new_df12.values[i])):
        new_df12.values[i][j] = new_df12.values[i][j].replace(',', '')
new_df12.values

array([['2941', '3299', '4069', ..., '5281', '4655', '4897'],
       ['980', '1162', '1679', ..., '2081', '1802', '1947'],
       ['1961', '2137', '2390', ..., '3200', '2853', '2950'],
       ...,
       ['115', '119', '170', ..., '135', '122', '131'],
       ['29', '29', '32', ..., '10', '9', '10'],
       ['36100', '40040', '41635', ..., '36738', '30198', '31651']],
       dtype=object)
```

ການລົບເຄື່ອງໜາຍ ','(comma) ອອກ ໂດຍການໃຫ້ຝຶກໜັນ replace

	2021-02	2021-01	2020-12	2020-11	2020-10	2020-09	2020-08	2020-07	2020-06	2020-05	2020-04	2020-03	2020-02	2020-01	2019-12	2019-11	2019-10	
List	Check	2941.0	3299.0	4069.0	3449.0	3097.0	3297.0	3467.0	3317.0	3451.0	3079.0	3332.0	4318.0	3808.0	4168.0	4652.0	4306.0	4250.0
Check_Inhouse	980.0	1162.0	1679.0	1260.0	1082.0	1159.0	1298.0	1135.0	1135.0	1085.0	1179.0	1489.0	1295.0	1403.0	1725.0	1700.0	1469.0	
Cheak_Interbank	1961.0	2137.0	2390.0	2189.0	2015.0	2138.0	2169.0	2182.0	2316.0	1994.0	2153.0	2829.0	2513.0	2765.0	2927.0	2606.0	2781.0	
e-Payments	33159.0	36741.0	37566.0	35570.0	34334.0	36471.0	35705.0	37064.0	38701.0	34180.0	38386.0	42440.0	35336.0	37897.0	36920.0	33360.0	33791.0	
transfer_BAHTNET	23927.0	27280.0	27668.0	26412.0	25896.0	28043.0	27529.0	28711.0	30421.0	26427.0	30653.0	33296.0	27754.0	29829.0	28860.0	25846.0	25781.0	
transfer_Bulk_Payment	2880.0	2880.0	3017.0	2916.0	2564.0	2634.0	2452.0	2618.0	2865.0	2700.0	2907.0	3369.0	2515.0	2774.0	2574.0	2585.0	2899.0	
transfer_Direct_credit	2009.0	1898.0	1959.0	1920.0	1720.0	1733.0	1595.0	1719.0	1748.0	1610.0	1907.0	2289.0	1618.0	1826.0	1679.0	1696.0	2003.0	
transfer_Direct_debit	568.0	657.0	679.0	642.0	537.0	544.0	550.0	599.0	722.0	632.0	582.0	746.0	592.0	604.0	538.0	586.0	582.0	
transfer_ITMX_Bulk_Payment	332.0	325.0	379.0	354.0	307.0	357.0	307.0	300.0	395.0	458.0	418.0	334.0	305.0	344.0	357.0	303.0	314.0	
Online_Retail_Funds_Transfer	2219.0	2196.0	2277.0	2001.0	1909.0	1830.0	1863.0	1815.0	1682.0	1571.0	1355.0	1559.0	1426.0	1424.0	1484.0	1323.0	1298.0	
Transfer_sub_cash_ATM	22.0	23.0	27.0	23.0	23.0	23.0	23.0	24.0	24.0	23.0	22.0	27.0	27.0	28.0	33.0	30.0	31.0	
Transfer_sub_cash_Internet_and_mobile	2190.0	2165.0	2242.0	1970.0	1878.0	1799.0	1832.0	1783.0	1650.0	1541.0	1326.0	1522.0	1390.0	1386.0	1441.0	1283.0	1257.0	
Transfer_sub_cash_across_bank_by_branch	7.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	7.0	7.0	10.0	9.0	10.0	10.0	10.0	10.0	
Intra_bank_transfer	3974.0	4222.0	4385.0	4057.0	3795.0	3798.0	3688.0	3748.0	3567.0	3345.0	3357.0	4050.0	3459.0	3660.0	3752.0	3400.0	3611.0	
transfer_payment_by_ATM	54.0	56.0	63.0	55.0	56.0	55.0	52.0	54.0	52.0	52.0	50.0	69.0	68.0	72.0	84.0	75.0	78.0	
transfer_payment_internet_smartphone	3920.0	4166.0	4322.0	4002.0	3739.0	3743.0	3636.0	3694.0	3515.0	3293.0	3307.0	3981.0	3391.0	3588.0	3668.0	3325.0	3533.0	
Payment_cards	130.0	134.0	187.0	154.0	142.0	140.0	147.0	144.0	140.0	113.0	91.0	144.0	160.0	187.0	225.0	183.0	180.0	
Debit_card	15.0	16.0	18.0	16.0	15.0	15.0	17.0	17.0	15.0	14.0	11.0	17.0	19.0	24.0	22.0	20.0	20.0	
Credit_card	115.0	119.0	170.0	138.0	127.0	125.0	130.0	127.0	125.0	99.0	80.0	127.0	141.0	163.0	203.0	163.0	160.0	
E-Money	29.0	29.0	32.0	30.0	28.0	26.0	26.0	28.0	26.0	24.0	23.0	22.0	22.0	23.0	25.0	23.0	22.0	
Total	36100.0	40040.0	41635.0	39019.0	37431.0	39768.0	39172.0	40381.0	42152.0	37259.0	41718.0	46758.0	39144.0	42065.0	41572.0	37666.0	38041.0	

ການແປງຄ່າຂໍ້ມູນຈາກString ໃຫ້ລາຍເປັນfloat ໂດຍໃຫ້ຝຶກໜັນ astype('float')

```
[ ] col = np.array(dataList[13].columns)
new_col_name = np.array(['List', '2021-02', '2021-01', '2020-12', '2020-11', '2020-10', '2020-09', '2020-08', '2020-07', '2020-06', '2020-05', '2020-04', '2020-03', '2020-02', '2020-01',
'2019-12', '2019-11', '2019-10', '2019-09', '2019-08', '2019-07', '2019-06', '2019-05', '2019-04', '2019-03', '2019-02', '2019-01',
'2018-12', '2018-11', '2018-10', '2018-09', '2018-08', '2018-07', '2018-06', '2018-05', '2018-04', '2018-03', '2018-02', '2018-01',
'2017-12', '2017-11', '2017-10', '2017-09', '2017-08', '2017-07', '2017-06', '2017-05', '2017-04', '2017-03', '2017-02', '2017-01'])
```

ການສ້າງອົງເຮີຍໃໝ່ດ້ວຍ numpy.array ໂດຍໃຫ້ ອົງເຮີຍ col ເກີບຂໍ້ມູນທີ່ເປັນຄອລັນໃນ dataList[12] ສ່ວນ ອົງເຮີຍ new_col_name ໃຊ້ເກີບຂໍ້ມູນຊື່ທີ່ຈະນໍາໄປເປີ່ມຢືນຊື່ຄອລັນໃໝ່ໃນ dataList[12]

```
[ ] for i in range(len(col)):
    dataList[13] = dataList[13].rename(columns={col[i]:new_col_name[i]})

[ ] ind = dataList[13]['List']
new_ind_name = np.array(['Check', 'Cheak_Inhouse', 'Cheak_Interbank', 'e-Payments', 'transfer_BAHTNET',
                        'transfer_Bulk_Payment', 'transfer_Direct_credit', 'transfer_Direct_debit', 'transfer_ITMX_Bulk_Payment', 'Online_Retail_Funds_Transfer',
                        'Transfer_sub_cash_ATM', 'Transfer_sub_cash_Internet_and_mobile', 'Transfer_sub_cash_across_bank_by_branch', 'Intra_bank_transfer', 'transfer_payment_by_ATM',
                        'transfer_payment_internet_smartphone', 'Payment_cards', 'Debit_card', 'Credit_card', 'E-Money', 'Total'])
```

ทำการสร้างอาร์เรย์ใหม่ด้วย numpy.array โดยให้ อาร์เรย์ ind เก็บข้อมูลที่เป็นແຕວຂອງ
คอลัมน์List ในdataList[13] ส่วน อาร์เรย์ new_ind_name ใช้เก็บข้อมูลชื่อที่จะนำไปเปลี่ยนชื่อແຕວຂອງ
คอลัมน์Listใหม่ใน dataList[13]

```
[ ] for i in range(len(ind)):
    dataList[13]['List'][i] = new_ind_name[i]

[ ] dataList[13]
```

	List	2021-02	2021-01	2020-12	2020-11	2020-10	2020-09	2020-08	2020-07	2020-06	2020-05	2020-04	2020-03	2020-02	2020-01	2019-12	2019-11	2019-10
0	Check	5,758	6,161	6,756	6,638	6,311	7,180	6,675	6,576	6,825	5,876	6,399	8,060	7,115	8,051	8,158	7,469	8,044
1	Cheak_Inhouse	2,459	2,631	2,873	2,754	2,666	3,187	2,946	2,787	2,794	2,452	2,642	3,354	3,072	3,408	3,418	3,168	3,334
2	Cheak_Interbank	3,299	3,530	3,883	3,884	3,645	3,993	3,929	3,789	4,031	3,424	3,757	4,706	4,043	4,643	4,740	4,301	4,710
3	e-Payments	1,390,637	1,401,565	1,416,346	1,333,362	1,249,598	1,184,387	1,187,397	1,164,781	1,116,023	1,072,989	911,075	973,030	897,026	893,748	913,619	850,588	846,849
4	transfer_BAHTNET	330	368	378	366	371	372	362	381	374	330	360	429	375	406	355	355	372
5	transfer_Bulk_Payment	90,535	70,775	77,172	69,851	49,420	63,124	50,621	61,133	71,009	70,827	60,460	52,552	48,471	48,810	48,093	44,566	46,041
6	transfer_Direct_credit	71,385	52,346	58,320	41,904	32,443	43,357	32,381	42,981	40,422	41,072	34,935	33,885	30,729	31,773	30,776	28,221	29,871
7	transfer_Direct_debit	14,252	13,633	13,569	14,089	12,282	13,045	12,951	12,636	12,616	11,698	12,544	13,306	12,756	12,190	12,020	11,573	11,346
8	transfer_ITMX_Bulk_Payment	4,898	4,796	5,283	4,858	4,695	6,722	5,289	5,516	17,969	18,057	12,981	5,361	4,986	4,847	5,297	4,772	4,824
9	Online_Retail_Funds_Transfer	585,345	600,291	583,312	552,607	493,752	451,271	462,256	442,908	421,085	405,261	332,666	336,457	299,371	286,531	288,457	264,894	255,902
10	Transfer_sub_cash_ATM	3,020	3,394	3,669	3,460	3,498	3,483	3,558	3,611	3,558	3,669	3,472	3,977	3,942	4,018	4,536	4,466	4,666
11	Transfer_sub_cash_Internet_and_mobile	582,101	596,653	579,369	548,882	489,980	447,500	458,414	439,013	417,239	401,335	328,920	332,116	295,072	282,132	283,538	260,041	250,830
12	Transfer_sub_cash_across_bank_by_branch	224	244	274	265	274	288	284	284	288	257	274	364	357	381	383	387	406
13	Intra_bank_transfer	462,343	471,239	471,204	447,872	435,949	412,884	414,792	406,435	384,332	379,655	329,147	350,305	316,736	311,689	320,208	303,303	303,666
14	transfer_payment_by_ATM	5,005	5,533	5,837	5,681	5,551	5,649	5,268	5,529	5,337	5,651	5,560	6,506	7,035	6,953	8,236	7,713	7,964
15	transfer_payment_internet_smartphone	457,338	465,706	465,367	442,191	430,398	407,238	409,524	400,906	378,995	374,004	323,587	343,790	309,701	304,735	311,972	295,590	295,702
16	Payment_cards	66,591	68,987	78,194	73,028	71,256	69,025	69,919	69,535	66,235	58,260	50,093	67,741	67,759	75,142	79,579	70,634	70,757
17	Debit_card	15,566	16,911	17,747	16,182	16,211	15,292	15,405	15,296	14,624	13,979	11,631	14,049	13,965	15,860	15,890	14,363	14,268
18	Credit_card	51,025	52,338	60,749	56,846	55,045	53,733	54,514	54,239	51,607	44,281	38,462	53,692	53,791	59,282	63,689	56,271	56,489
19	E-Money	191,493	189,905	206,086	198,638	198,846	187,711	189,447	184,389	172,988	158,656	138,349	165,551	164,316	171,171	176,927	166,116	170,111
20	Total	1,402,395	1,407,726	1,423,102	1,340,000	1,255,905	1,191,567	1,194,272	1,171,357	1,122,848	1,078,865	917,474	981,095	904,143	901,799	921,777	858,037	854,893

ทำการแสดงข้อมูลใน dataList[13] หลังจากการทำการเปลี่ยนชื่อແຕວและคอลัมน์แล้ว

```
[ ] new_df13 = dataList[13].groupby('List').sum()

[ ] for i in range(len(new_df13.values)):
    for j in range(len(new_df13.values[i])):
        new_df13.values[i][j] = new_df13.values[i][j].replace(',', '')
new_df13.values

array([['15566', '16910', '17747', ..., '7886', '7056', '7810'],
       ['2459', '2631', '2873', ..., '4601', '3760', '3791'],
       ['3299', '3530', '3883', ..., '6132', '5166', '5414'],
       ...,
       ['4898', '4796', '5283', ..., '4654', '4207', '3877'],
       ['5005', '5533', '5837', ..., '23333', '18981', '20971'],
       ['457338', '465706', '465367', ..., '86836', '74665', '72839']],
      dtype=object)
```

ทำการลบเครื่องหมาย ','(comma) ออก โดยการใช้ฟังก์ชัน replace

	2021-02	2021-01	2020-12	2020-11	2020-10	2020-09	2020-08	2020-07	2020-06	2020-05	2020-04	2020-03	2020-02	2020-01	2019-12	2019-11	2019-10	2019-09	2019-08
List																			
Debit_card	15568	16910	17747	16182	16211	15292	15405	15296	14628	13979	11631	14049	13968	15860	15890	14363	14268	13353	13768
Cheak_Inhouse	2459	2631	2873	2754	2666	3187	2946	2787	2794	2452	2642	3354	3072	3408	3418	3168	3334	3841	3554
Cheak_Interbank	3299	3530	3883	3884	3645	3993	3929	3789	4031	3424	3757	4706	4043	4643	4740	4301	4710	4969	4637
Check	5758	6161	6756	6638	6311	7180	6875	6765	6825	5876	6399	8060	7115	8051	8158	7469	8044	8810	8191
Credit_card	51025	52338	60749	56846	55045	53733	54514	54239	51607	44281	38462	53929	53791	59282	63689	56271	56489	54497	54531
E-Money	191493	189905	206086	198638	198846	187711	189447	184389	172988	158656	138349	165551	164316	171171	176927	166816	170111	178673	178525
Intra_bank_transfer	462343	471239	474872	435949	412884	414792	406435	384332	379655	329147	350305	316736	311688	320208	303303	303666	289166	286665	
Online_Retail_Funds_Transfer	585345	600291	583312	552607	493752	451271	462256	442908	421085	405261	332666	336457	299371	286531	288457	264994	255902	238748	228422
Payment_cards	66591	68987	78194	73028	71256	69025	69919	69535	66235	58260	50093	67741	67759	75142	79579	70634	70757	67850	68299
Total	1402395	1407726	1423102	1340000	1255095	1191507	1194272	1171357	1122848	1078665	917474	981095	904143	901799	921777	858037	854893	829837	812040
Transfer_sub_cash_ATM	3020	3994	3669	3460	3498	3483	3558	3611	3558	3669	3472	3977	3942	4018	4536	4466	4666	4727	4969
Transfer_sub_cash_Internet_and_mobile	582101	596953	579396	548882	489980	447500	458414	439013	417239	401335	328920	332116	295072	282132	283538	260041	250830	233612	223029
Transfer_sub_cash_across_bank_by_branch	224	244	274	265	274	288	284	284	288	257	274	364	357	381	383	387	406	409	424
e-Payments	1396637	1401565	1416346	1333362	1249594	1184387	1187397	1164781	1116023	1072989	911075	973035	897028	893748	913619	850568	846849	821027	803849
transfer_BAHTNET	330	368	378	366	371	372	362	381	374	330	360	429	375	406	355	355	372	371	386
transfer_Bulk_Payment	90535	70775	77172	60851	49420	63124	50621	61133	71009	70827	60460	52552	48471	48810	48093	44566	46041	47999	41552
transfer_Direct_credit	71385	52346	58320	41904	32433	42381	42981	40422	41072	34935	33885	30729	31773	30776	28221	29871	30376	25887	
transfer_Direct_debit	14252	13633	13569	14089	12282	13045	12951	12636	12618	11698	12544	13306	12756	12190	12020	11573	11346	11492	10949
transfer_ITMX_Bulk_Payment	4898	4796	5283	4858	4695	6722	5289	5516	17969	18057	12981	5361	4986	4847	5297	4772	4824	6131	4716
transfer_payment_by_ATM	5005	5533	5837	5681	5551	5649	5268	5529	5337	5651	5560	6506	7035	6953	8236	7713	7964	8326	8644
transfer_payment_internet_smartphone	457338	465706	465367	442191	430398	407235	406524	400006	378995	374004	323587	343799	309701	304735	311972	295590	295702	280860	278021

ການແປ່ງຄໍາຂໍ້ມູນຈາກ String ໄກສາຍເປັນ integer ໂດຍໃຊ້ຟັງກືນ astype('int')

```
for i in range(len(dataList)):
    dataList[i].info()
```

ການຈົບສົນວ່າຂໍ້ມູນທີ່ໄດ້ມາມີຄ່າ missing value ອີ່ໄມ່ ໂດຍຈະໃຊ້ຟັງກືນ info() ເພື່ອຊັບສິນຂອງ dataset ອີກຮັ້ງ

ຕົວອ່າງບາງສ່ວນຂອງຜລົດຟັງກືນ

<class 'pandas.core.frame.DataFrame'>	RangeIndex: 8 entries, 0 to 7	Data columns (total 2 columns):	# Column Non-Null Count Dtype	---	0 Year 4 non-null float64	1 Social Media 4 non-null float64	2 Searching Information 4 non-null float64	3 Recreational 4 non-null float64	4 Watching TV/listening Music 4 non-null float64	5 Buying goods and service 4 non-null float64	6 Downloading softwares/songs/movies/games 4 non-null float64	7 Financial transactions 4 non-null object	8 Playing online games 4 non-null float64	9 E-learning 4 non-null float64	10 Finding jobs/Job application 4 non-null float64	11 Booking/buying concert/movie ticket 4 non-null float64	12 Booking/buying ticket 4 non-null object	13 Booking/renting a service 4 non-null float64	14 Watching stock exchange 4 non-null float64	15 Paying taxes 4 non-null float64	16 Booking the hotel 4 non-null float64	17 Receive/send document 4 non-null float64	18 Booking/renting a room 4 non-null float64	19 Buying Foods 4 non-null float64	20 Calling taxi 4 non-null float64	21 Sending message 4 non-null object	22 Live application 4 non-null object	23 Playing games and service 4 non-null object	24 Buying goods via online meeting 4 non-null object	25 Buying products 4 non-null object	dtypes: float64(19), object(7)	memory usage: 416.0 bytes
<class 'pandas.core.frame.DataFrame'>	RangeIndex: 4 entries, 0 to 3	Data columns (total 2 columns):	# Column Non-Null Count Dtype	---	0 Year 4 non-null int64	1 User_Number 4 non-null float64	dtypes: float64(1), int64(1)	memory usage: 192.0 bytes																								
<class 'pandas.core.frame.DataFrame'>	RangeIndex: 9 entries, 0 to 8	Data columns (total 3 columns):	# Column Non-Null Count Dtype	---	0 Year 9 non-null int64	1 Platform 9 non-null object	2 Rate 9 non-null float64	dtypes: float64(1), int64(1), object(1)	memory usage: 344.0+ bytes																							
<class 'pandas.core.frame.DataFrame'>	Int64Index: 56 entries, 0 to 59	Data columns (total 4 columns):	# Column Non-Null Count Dtype	---	0 Year 56 non-null int64	1 Platform 56 non-null object	2 Rate 56 non-null float64	3 Type 56 non-null object	dtypes: float64(1), int64(1), object(2)	memory usage: 2.24 KB																						
<class 'pandas.core.frame.DataFrame'>	RangeIndex: 4 entries, 0 to 3	Data columns (total 11 columns):	# Column Non-Null Count Dtype	---	0 Year 4 non-null int64	1 Messenger 4 non-null float64	2 Social Network 4 non-null object	3 Video 4 non-null float64	4 Music 4 non-null object	5 Game 4 non-null float64	6 Shopping 4 non-null object	7 Map 4 non-null float64	8 Banking 4 non-null float64	9 Health and Fitness 4 non-null object	10 Diet 4 non-null object	dtypes: float64(5), int64(1), object(5)	memory usage: 488.0+ bytes															

```
[ ] for i in range(len(dataList)):
    print(dataList[i].isnull().sum())
    print()
```

ทำการหาว่า DataFrame ของแต่ละdataset ยังมีค่า missing value เหลืออยู่จำนวนกี่ค่า โดยใช้ฟังก์ชัน isnull() หรือ pandas.DataFrame.isnull() และใช้ฟังก์ชัน sum() ต่อท้ายอีกรอบ

ผลลัพธ์ที่ได้

Year 0 Timeperday 0 dtype: int64	Year 0 Youtube 0 Facebook 0 LINE 0 Instagram 0 Pantip 0 Twitter 0 WhatsApp 0 dtype: int64	List 0 2021-02 0 2021-01 0 2020-12 0 2020-11 0 2020-10 0 2020-09 0 2020-08 0 2020-07 0 2020-06 0 2020-05 0 2020-04 0 2020-03 0 2020-02 0 2020-01 0 2019-12 0 2019-11 0 2019-10 0 2019-09 0 2019-08 0 2019-07 0 2019-06 0 2019-05 0 2019-04 0 2019-03 0 2019-02 0 2019-01 0 2018-12 0 2018-11 0 2018-10 0 2018-09 0 2018-08 0 2018-07 0 2018-06 0 2018-05 0 2018-04 0 2018-03 0 2018-02 0 2018-01 0 2017-12 0 2017-11 0 2017-10 0 2017-09 0 2017-08 0 2017-07 0 2017-06 0 2017-05 0 2017-04 0 2017-03 0 2017-02 0 2017-01 0 dtype: int64
--	---	--

เมื่อทำการตรวจสอบค่า missing value อีกพบว่าจำนวนmissing valueที่ได้ออกมาก็หักหมัดนั้นเป็น 0 หักหมัด หมายความว่า ไม่มีค่า missing value เหลืออยู่แล้ว

Data Visualization

```

plt.rcParams['figure.figsize'] = 15, 10

f, ax = plt.subplots(1)
x_df0 = np.array(dataList[0]['Year'])
y_df0 = np.array(dataList[0]['Timeperday'])
style.use("ggplot")

plt.title("Average internet usage between 2013-2020")
plt.plot(x_df0, y_df0, 'ob-', label="Time", color='red')
plt.xlabel("Year")
plt.ylabel("Timeperday(minite)")

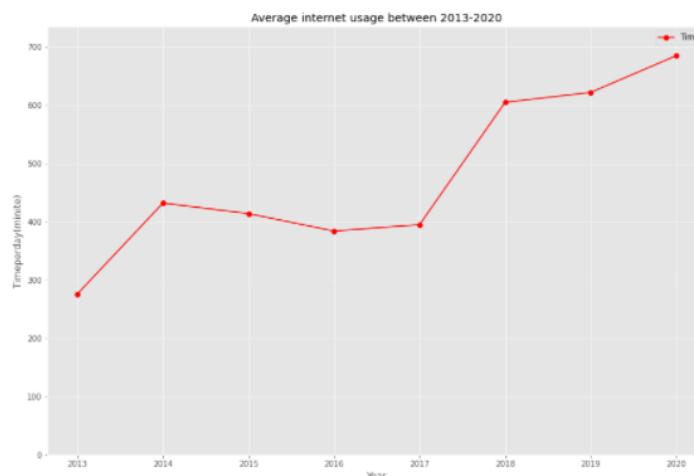
ax.set_xlim(0, max(y_df0)+50)

plt.legend()
plt.show()

```

ใช้ฟังก์ชัน plt.rcParams['figure.figsize'] ในการกำหนดขนาดรูปของกราฟเท่ากับ 15*10 จากนั้นสร้างอาร์เรย์ใหม่จากฟังก์ชัน numpy.array โดยให้ x_df0 เก็บค่าในคอลัมน์ Year ใน dataList[0] และให้ y_df0 เก็บค่าในคอลัมน์ Timeperday ใน dataList[0] แล้วทำการ plot กราฟเส้นโดยใช้ plt.plot() ใช้ plt.title , plt.xlabel , plt.ylabel ในการกำหนดชื่อ หัวข้อกราฟ ชื่อของกราฟแกน x ชื่อของกราฟแกน y ตามลำดับ ใช้ plt.legend() เพื่อแสดงคำอธิบายของกราฟออกแบบ และใช้ plt.show() ในการแสดงผลลัพธ์ทั้งหมด

ผลลัพธ์



เป็นกราฟแสดงจำนวนชั่วโมงในการใช้อินเตอร์เน็ตของคนไทยเฉลี่ยต่อวัน ในช่วงปี 2013 ถึง 2020 โดยมีหน่วยเป็นนาที ซึ่งแสดงให้เห็นถึงการใช้อินเตอร์เน็ตที่เพิ่มขึ้นในทุก ๆ ปี จากราฟที่มีการเพิ่มขึ้นของจำนวนชั่วโมงในการใช้อินเตอร์เน็ต และจากผลบังเอิญได้ว่าในปี 2020 มีชั่วโมงการใช้งานอินเตอร์เน็ตเฉลี่ยต่อวันเพิ่มมากขึ้นจากปี 2019 ถึง 1 ชั่วโมง 3 นาที (63 นาที)

```

df1_2 = dataList[1].groupby('Year').sum()
df1_2
df1_2_col = np.array(df1_2.index)
df1_2_col

array([2017, 2018, 2019, 2020])

```

สร้างตัวแปร df1_2 ในการเก็บข้อมูลใน dataList[1] คอลัมน์ Year และทำการสร้างอาร์เรย์ใหม่ จากฟังก์ชัน numpy.array โดยให้ df1_2_col เก็บค่าใน df1_2 และทำการแสดงค่าออกมา

```

z1 = np.array(dataList[1]['Z_Weekday'])
z2 = np.array(dataList[1]['Z_Weekends'])

y1 = np.array(dataList[1]['Y_Weekday'])
y2 = np.array(dataList[1]['Y_Weekends'])

x1 = np.array(dataList[1]['X_Weekday'])
x2 = np.array(dataList[1]['X_Weekends'])

bb1 = np.array(dataList[1]['BabyB_Weekday'])
bb2 = np.array(dataList[1]['BabyB_Weekends'])

year= df1_2_col
width = 0.2

#เพื่อแสดง
plt.bar(year,z2, width,label="GenZ_Weekend",alpha=1,color='pink')
plt.bar(year,z1, width,label="GenZ_Weekday",alpha=0.75,color = 'red')

plt.bar(year+width,y2, width,label="GenY_Weekend",alpha=0.25,color = 'blue')
plt.bar(year+width,y1, width,label="GenY_Weekday",alpha=0.75,color = 'RoyalBlue')

plt.bar(year+width*x2, width,label="GenX_Weekend",alpha=1.0 ,color = 'lime')
plt.bar(year+width*x1, width,label="GenX_Weekday",alpha=0.25 ,color = 'green')

plt.bar(year+width*3,bb2, width,label="BabyBoom_Weekend",alpha=0.75 ,color = 'orange')
plt.bar(year+width*3,bb1, width,label="BabyBoom_Weekday",alpha=0.75 ,color = 'salmon')

plt.title("Internet Hours by Generation 2017-2020")
plt.xlabel("Year")
plt.ylabel("TimePerDay(minite)")
#linux ให้มีค่าไฟ B
plt.xticks(year)
plt.legend()
plt.show()

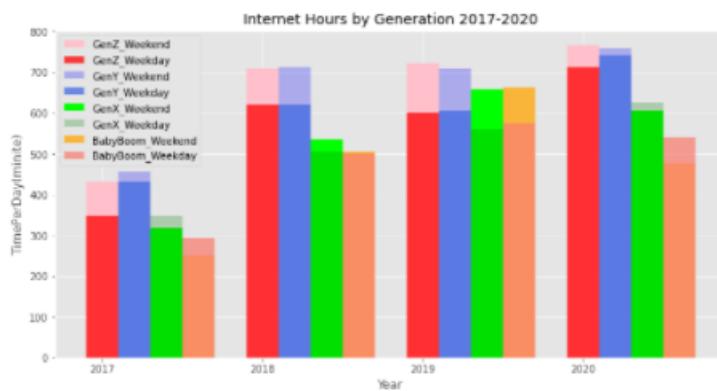
```

ทำการสร้างอาร์เรย์ใหม่ด้วย numpy.array โดยให้ :

- z1 เก็บข้อมูลที่เป็นคอลัมน์ Z_Weekday ใน dataList[1]
- z2 เก็บข้อมูลที่เป็นคอลัมน์ Z_Weekends ใน dataList[1]
- y1 เก็บข้อมูลที่เป็นคอลัมน์ Y_Weekday ใน dataList[1]
- y2 เก็บข้อมูลที่เป็นคอลัมน์ Y_Weekends ใน dataList[1]
- x1 เก็บข้อมูลที่เป็นคอลัมน์ X_Weekday ใน dataList[1]
- x2 เก็บข้อมูลที่เป็นคอลัมน์ X_Weekends ใน dataList[1]
- bb1 เก็บข้อมูลที่เป็นคอลัมน์ BabyB_Weekday ใน dataList[1]
- bb2 เก็บข้อมูลที่เป็นคอลัมน์ BabyB_Weekends ใน dataList[1]

ใช้ตัวแปร year ในการเก็บค่าอาร์เรย์ df1_2_col ที่เตรียมไว้ และทำการ plot กราฟ โดยใช้ พังก์ชัน plt.bar() ใช้ plt.title , plt.xlabel , plt.ylabel ในการกำหนดชื่อ หัวข้อกราฟ ชื่อของกราฟแกน x ชื่อของกราฟแกน y ตามลำดับ ใช้ plt.legend() เพื่อแสดงคำอธิบายของกราฟออกมา และใช้ plt.show() ในการแสดงผลลัพธ์ทั้งหมด

ผลลัพธ์



เป็นกราฟแสดงจำนวนชั่วโมงในการใช้อินเตอร์เน็ตแยกตามเจนเนอเรชัน ได้แก่ Gen Z ,Gen Y ,Gen X และ Baby Boomer ตามลำดับ ซึ่งแต่ละเจนเนอเรชันก็สามารถแบ่งจำนวนชั่วโมงได้อีก 2 ประเภทด้วยกันคือ จำนวนชั่วโมงในการใช้อินเตอร์เน็ตในวันธรรมดा(Weekday) และ จำนวนชั่วโมงในการใช้อินเตอร์เน็ตที่ใช้ในวันหยุด(Weekends)

โดยกราฟแท่งแรกของแต่ละปีเป็นกราฟของ Gen Z โดยแท่งสีแดงจะเป็นกราฟของวันธรรมด้า ส่วนกราฟสีชมพูจะเป็นกราฟของวันหยุด กราฟแท่งที่สองเป็นกราฟของ Gen Y โดยแท่งสีน้ำเงินเข้ม (Royal Blue) จะเป็นกราฟของวันธรรมด้า ส่วนกราฟสีน้ำเงิน(blue)จะเป็นกราฟของวันหยุด กราฟแท่งที่สามเป็นกราฟของ Gen Z โดยแท่งสีเขียว(green)จะเป็นกราฟของวันธรรมด้า ส่วนกราฟสีเหลืองแกมเขียว(lime)จะเป็นกราฟของวันหยุด จ]กราฟแท่งที่สี่เป็นกราฟของ Baby Boomer โดยแท่งสีส้ม(orange)จะเป็นกราฟของวันธรรมด้า ส่วนกราฟ สีชมพูเนื้อปลาแซลมอน(salmon)จะเป็นกราฟของวันหยุด

ซึ่งจากผลลัพธ์ของกราฟแสดงจำนวนชั่วโมงในการใช้งานอินเตอร์เน็ตเมื่อยแยกตามเจนเนอเรชันแล้ว จะเห็นว่า GenZ(ที่มีอายุ 9 – 24 ปี) และ GenY(ที่มีอายุ 25 – 40 ปี) ที่เป็นช่วงวัยเรียนและวัยทำงาน จะมีการใช้งานอินเตอร์เน็ตที่มากกว่า Gen อื่นๆ และผลจากสถานการณ์โควิดเราจะเห็นได้ว่าวันธรรมด้าของทั้ง GenZ และ GenY มีชั่วโมงการใช้งานอินเตอร์เน็ตเพิ่มขึ้นจากปี 2019 ถึง 1 ชั่วโมง 53 นาที และ 2 ชั่วโมง 14 นาที ตามลำดับ

dataList[3]

	Year	Home	Workplace	School	Publicplace	On_the_way	Internet_cafe
0	2018	87.6	49.5	19.7	15.9	14.0	3.7
1	2017	85.6	52.4	17.5	18.8	24.1	2.3
2	2018	91.5	52.2	20.7	21.0	33.5	1.1
3	2019	95.7	67.0	18.2	23.7	38.4	1.3
4	2020	67.8	30.1	17.3	45.4	16.2	21.3

ทำการแสดงข้อมูลใน dataList[3] ออกมำ

```

ho = dataList[3].groupby('Year')[ 'Home'].sum()
work = dataList[3].groupby('Year')[ 'Workplace'].sum()
sc = dataList[3].groupby('Year')[ 'School'].sum()
pp = dataList[3].groupby('Year')[ 'Publicplace'].sum()
oty = dataList[3].groupby('Year')[ 'On_the_way'].sum()
inter = dataList[3].groupby('Year')[ 'Internet_cafe'].sum()

style.use("ggplot")

fig = plt.figure()
ax = fig.gca()
ax.xaxis.set_major_locator(MaxNLocator(integer=True))

plt.title("Internet use place")
plt.rcParams['figure.figsize'] = 12, 6

plt.plot(ho,'ob-', color = 'black',label="Home")
plt.plot(work,'ob-', color = 'pink',label="Workplace")
plt.plot(sc,'ob-', color = 'purple',label="School")
plt.plot(pp,'ob-', color = 'red',label="Publicplace")
plt.plot(oty,'ob-', color = 'green',label="On_the_way")
plt.plot(inter,'ob-', color = 'yellow',label="Internet_cafe")

plt.xlabel("Year")
plt.ylabel("Rate(%)")
plt.legend(loc="lower center", bbox_to_anchor=(1.0, 1.0))
plt.show()

```

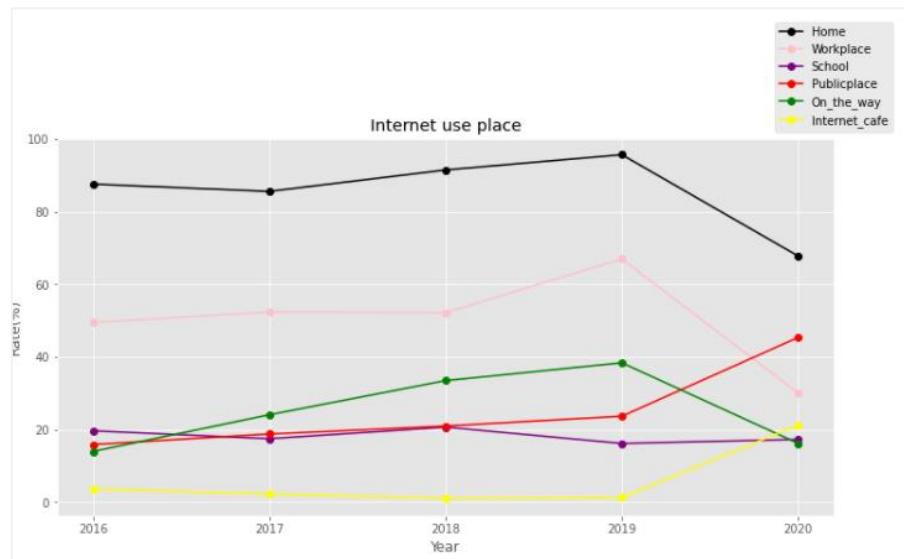
ทำการสร้าง DataFrame ย่อขึ้นมาใหม่ โดยใช้ฟังก์ชัน groupby ในการจับกลุ่มข้อมูล ซึ่งกำหนดตัวแปรให้ :

- ho เก็บค่าข้อมูลแต่ละปี (Year) ใน colum น์ Home
- work เก็บค่าข้อมูลแต่ละปี (Year) ใน colum น์ Workspace
- sc เก็บค่าข้อมูลแต่ละปี (Year) ใน colum น์ School
- pp เก็บค่าข้อมูลแต่ละปี (Year) ใน colum น์ Publicplace
- oty เก็บค่าข้อมูลแต่ละปี (Year) ใน colum น์ On_the_way
- inter เก็บค่าข้อมูลแต่ละปี (Year) ใน colum น์ Internet_cafe

ใช้ฟังก์ชัน rcParam['fugure.figsize'] ในการกำหนดขนาดรูปของกราฟที่จะออกมากให้มีขนาดเท่ากับ 12*6 และใช้ฟังก์ชัน ax.xaxis.set_major_locator(MaxNLocator(integer=True)) เพื่อให้ค่าในแกน x มีชนิดข้อมูลเป็น integer

จากนั้นทำการ plot กราฟ จาก DataFrame ย่อที่เราเตรียมไว้จากการ groupby โดยใช้ฟังก์ชัน plt.plot() ในการ plot กราฟเส้น และใช้ plt.title , plt.xlabel , plt.ylabel ในการกำหนดชื่อ หัวข้อกราฟ ชื่อของกราฟแกน x ชื่อของกราฟแกน y ตามลำดับ ใช้ plt.legend() เพื่อแสดงคำอธิบายของกราฟออกมานั้น และใช้ plt.show() ในการแสดงผลลัพธ์ทั้งหมด

ຜລລັບທີ່



ເປັນກາຟແສດງຮ້ອຍລະຸ້ມໍໃຊ້ອິນເຕົຣ໌ເນື້ອຕຳແນກຕາມສຖານທີ່ໄດ້ແກ່ ບ້ານ(Home) , ສຖານທີ່ກຳນົດ (Workspace) , ໂຮງຮຽນ(School) ,ສຖານທີ່ສາຫະລະ(Publicplace),ຮ່ວ່າງກາຣເດີນທາງ(On_the_way) ແລະ ຮ້ານອິນເຕົຣ໌ເນື້ອຕຳ(Internat_cafe) ຕາມລຳດັບ

ໂດຍກາຟສີ່ຈຳ(black) ເປັນກາຟຂອງກາຣໃຊ້ອິນເຕົຣ໌ເນື້ອຕຳທີ່ບ້ານ ກາຟສີ່ໜົມພູ(pink)ຈະເປັນກາຟຂອງກາຣໃຊ້ອິນເຕົຣ໌ເນື້ອຕຳທີ່ສຖານທີ່ ກາຟເຂົ້າວ(green) ເປັນກາຟຂອງກາຣໃຊ້ອິນເຕົຣ໌ເນື້ອຕຳທີ່ໂຮງຮຽນ ກາຟສີ່ແດງ(red) ເປັນກາຟຂອງກາຣໃຊ້ອິນເຕົຣ໌ເນື້ອຕຳທີ່ສາຫະລະ ກາຟສີ່ມ່ວງ(purple) ເປັນກາຟຂອງກາຣໃຊ້ອິນເຕົຣ໌ເນື້ອຕຳທີ່ຮ່ວ່າງເດີນທາງ ແລະກາຟສີ່ເໜືອງ(yellow) ເປັນກາຟຂອງກາຣໃຊ້ອິນເຕົຣ໌ເນື້ອຕຳທີ່ຮ້ານອິນເຕົຣ໌ເນື້ອຕຳ

ໜຶ່ງຈາກກາຟເມື່ອຈຳແນກຕາມສຖານທີ່ຈະເກີດໄດ້ວ່າບ້ານເປັນສຖານທີ່ທີ່ມີກາຣໃຊ້ອິນເຕົຣ໌ເນື້ອຕຳມາກທີ່ສຸດຮອງລົງມາຈະເປັນສຖານທີ່ສາຫະລະ ອັນດັບສາມປັນສຖານທີ່ກຳນົດສັງເກົດໄດ້ວ່າກາຟຝລດລົງຈາກປີ2019ເປັນອິ່ງມາກໜຶ່ງຈາກກາຣທີ່ອາຊີພສ່ວນໃໝ່ເປີ່ມຍິນຈາກທີ່ສຖານທີ່ກຳນົດປົກຕິມາເປັນກາຣກຳນົດທີ່ບ້ານແກ່ ອັນດັບຕ່ອມາເປັນຮ້ານອິນເຕົຣ໌ເນື້ອຕຳ ສ່ວນໂຮງຮຽນຍັງມີອັຕຣາກາຣໃຊ້ອິນເຕົຣ໌ເນື້ອຕຳຄ່ອນຂ້າງຄົງທີ່ ກາຣໃຊ້ອິນເຕົຣ໌ເນື້ອຕຳຮ່ວ່າງກາຣເດີນທາງກົດລົງຈາກປີ2019ອ່າງເໜີນໄດ້ຊັດ ເຫຼຸ້ມຈະເປັນພຽງ ປະຊາຊົນສ່ວນໃໝ່ເຫຼຸ້ມເລື່ອຍກາຣເດີນທາງໃນຊ່ວງສຖານກາຣໂຄວິດເພື່ອລົດໂອກາສໃນກາຣຕິດເຊື້ອໂຄວິດ-19

```
df3_x_2020 = np.array(dataList[3].columns.drop('Year'))
df3_y_2020 = np.array(dataList[3][dataList[3]['Year']==2020].drop('Year', axis=1))
df3_y_2020 = df3_y_2020.reshape(6)
```

ทำการสร้างอาร์เรย์ใหม่ด้วย numpy.array โดยให้ df3_x_2020 เก็บข้อมูลคอลัมน์ใน dataList[3] โดยจะดรอปค่าคอลัมน์ Year ทิ้ง และ ให้ df3_y_2020 เก็บข้อมูลคอลัมน์ใน dataList[3] โดยเก็บเฉพาะข้อมูลของปี 2020 เท่านั้นและทำการดรอปค่าคอลัมน์ Year ทิ้ง เช่นกัน แล้วนำ อาร์เรย์ df3_y_2020 ไปจัดเรียงใหม่ โดยใช้ method reshape

```
df3_new_df = pd.DataFrame({'Place':df3_x_2020,'Rate':df3_y_2020})
```

	Place	Rate
0	Home	67.8
1	Workplace	30.1
2	School	17.3
3	Publicplace	45.4
4	On_the_way	16.2
5	Internet_cafe	21.3

ทำการสร้าง DataFrame (df_new_df) ขึ้นมาใหม่ โดยกำหนดคอลัมน์ทั้งหมดสองคอลัมน์ คือ Place และ Rate ซึ่ง Place ได้ข้อมูลจากอาร์เรย์ df_x_2020 ส่วน Rate ได้ข้อมูลจากอาร์เรย์ df_y_2020

```
df3_new_df_sorted = df3_new_df.sort_values('Rate', ascending=False)
df3_new_df_sorted
```

	Place	Rate
0	Home	67.8
3	Publicplace	45.4
1	Workplace	30.1
5	Internet_cafe	21.3
2	School	17.3
4	On_the_way	16.2

ทำการเรียงข้อมูลของอาร์เรย์ df3_new_df จากค่ามากไปน้อย

```

plt.rcParams['figure.figsize'] = 10, 10
style.use("ggplot")
sns.barplot(x='Place', y='Rate', data=df3_new_df_sorted)

for index,data in enumerate(df3_new_df_sorted['Rate']):
    plt.text(x=index-0.15 , y =data+1 , s=f"{data}" , fontdict=dict(fontsize=13))

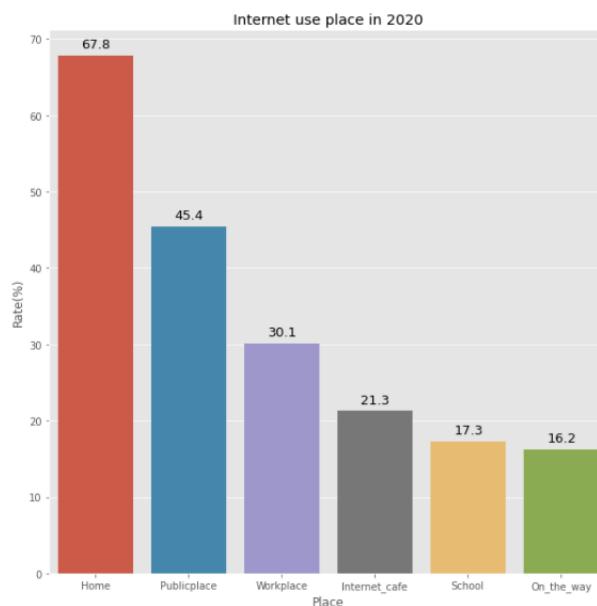
plt.title("Internet use place in 2020")
plt.xlabel("Place")
plt.ylabel("Rate(%)")
plt.show()

```

ใช้ฟังก์ชัน rcParam['fugure.figsize'] ในการกำหนดขนาดรูปของกราฟเท่ากับ 10*10 ใช้ฟังก์ชัน sns.barplot() ในการ plot กราฟแท่ง โดยใช้ข้อมูลของอาร์เรย์ df3_new_df ที่เตรียมไว้มาทำการ plot ซึ่งแกน x คือ คอลัมน์ Place ส่วนแกน y คือ คอลัมน์ Rate และใช้ plt.text() ในการใส่ค่าของ Rate ในกราฟแต่ละแท่ง

ใช้ plt.title , plt.xlabel , plt.ylabel ในการกำหนดชื่อ หัวข้อกราฟ ชื่อของกราฟแกน x ชื่อของกราฟแกน y ตามลำดับ และใช้ plt.show() ในการแสดงผลลัพธ์ทั้งหมด

ผลลัพธ์



จากการเฝ้าระวังตามสถานที่ในปี 2020 จะเห็นได้ว่าบ้านเป็นสถานที่ที่มีการใช้งาน
อินเตอร์เน็ตมากที่สุด ซึ่งเหตุผลอาจมาจากการเว้นระยะห่างทางสังคมทำให้ต้องมีการทำงานหรือเรียน
จากที่บ้านเป็นหลัก

	Year	Student	Freelance	Employee	Self-employed	Unemployed	Official	Maid
0	2019	650	630	610	634	632	622	638
1	2020	763	688	670	621	610	581	541

ทำการแสดงข้อมูลใน dataList[2] ออกมາ

```
df2_x_career = np.array(dataList[2].columns.drop('Year'))
df2_y_career_2019 = np.array(dataList[2][dataList[2]['Year']==2019].drop('Year', axis=1)).reshape(7)
df2_y_career_2020 = np.array(dataList[2][dataList[2]['Year']==2020].drop('Year', axis=1)).reshape(7)
```

ทำการสร้างอาร์เรย์ใหม่ด້ວຍ numpy.array โดยໃຫ້ :

- df2_x_career เก็บข้อมูลคอลัมน์ใน dataList[2] โดยจะดรอปค่าคอลัมน์ Year ทີ່
- df2_y_career_2019 เก็บข้อมูลคอลัมน์ใน dataList[2] โดยเก็บเฉพาะข้อมูลของປີ 2019 ເທົ່ານັ້ນ ແລະ ทำการดรอปค่าคอลัมน์ Year ທີ່ ຈາກນັ້ນນຳໄປຈັດເຮືອງໃໝ່ ໂດຍໃຊ້ method reshape
- df2_y_career_2020 เก็บข้อมูลคอลัมน์ใน dataList[2] โดยเก็บเฉพาะข้อมูลของປີ 2020 ເທົ່ານັ້ນ ແລະ ทำการดรอปค่าคอลัมน์ Year ທີ່ ຈາກນັ້ນນຳໄປຈັດເຮືອງໃໝ່ ໂດຍໃຊ້ method reshape

```
x = np.arange(len(df2_x_career))
width = 0.35
plt.rcParams['figure.figsize'] = 10, 10

style.use("ggplot")

fig, ax = plt.subplots()
rects1 = ax.bar(x - width/2, df2_y_career_2019, width, label='2019')
rects2 = ax.bar(x + width/2, df2_y_career_2020, width, label='2020')

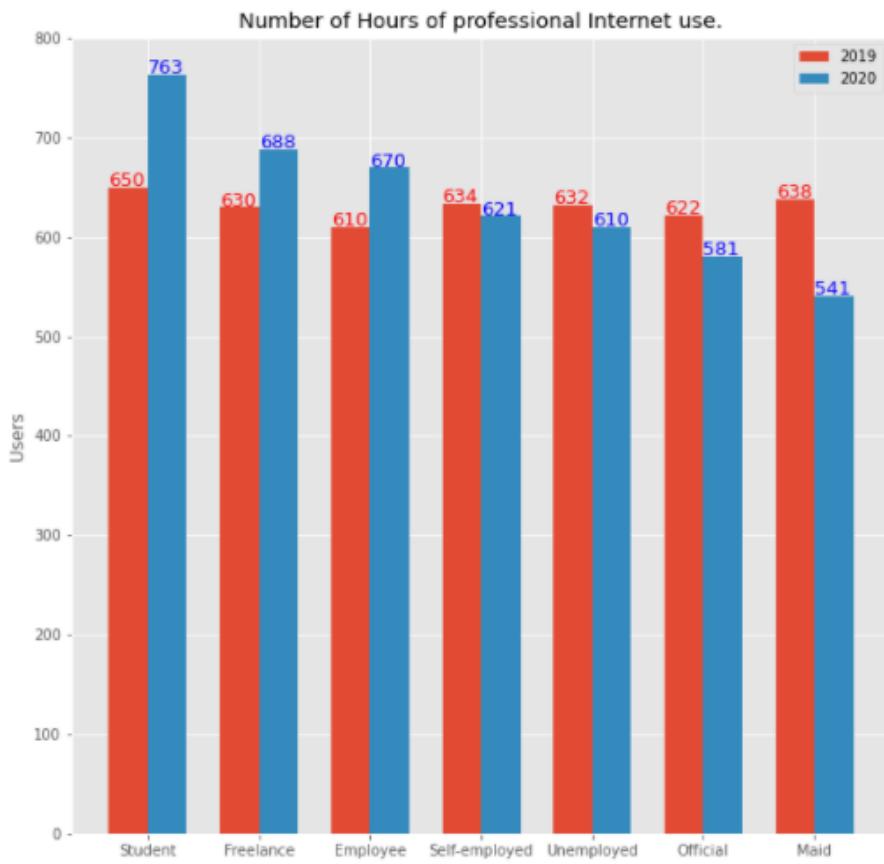
for index,data in enumerate(df2_y_career_2019):
    plt.text(x=index-width , y =data+1 , s=f'{data}' , fontdict=dict(fontsize=13),color = 'red')
for index,data in enumerate(df2_y_career_2020):
    plt.text(x=index , y =data+1 , s=f'{data}' , fontdict=dict(fontsize=13),color = 'blue')

ax.set_ylabel('Users')
ax.set_title('Number of Hours of professional Internet use.')
ax.set_xticks(x)
ax.set_xticklabels(df2_x_career)
ax.legend()
plt.show()
```

ໃຊ້ພັກສັນ rcParam['fugure.figsize'] ໃນການກຳຫັດຂາດຽບປັບອງກາຮັກເກີບ 10*10 ໃຊ້ພັກສັນ sns.barplot() ໃນກາຮັກກາຮັກແທ່ງ ໂດຍໃຊ້ຂໍ້ມູນຂອງອາຮົມຍໍ df2_y_career_2019 ແລະ df2_y_career_2020 ທີ່ເຕີມໄວ້ມາກຳທຳກາຮັກ ແລະ ໃຊ້ plt.text() ໃນກາຮັກໄສຄ່າຂອງ Rate ໃນກາຮັກແຕ່ລະແທ່ງ ໂດຍຈະແຍກທຳຂອງແຕ່ລະປີ

ໃຊ້ plt.title , plt.xlabel , plt.ylabel ໃນການກຳຫັດຊື່ ມັນຂອງກາຮັກ ຊື່ຂອງກາຮັກແກນ x ຊື່ຂອງກາຮັກແກນ y ຕາມລຳດັບ ແລ້ວ ໃຊ້ plt.show() ໃນກາຮັກແສດງຜລັພັກທັງໝົດ

ผลลัพธ์



เป็นกราฟแสดงจำนวนชั่วโมงในการใช้อินเตอร์เน็ตแยกตามอาชีพ ได้แก่ นักเรียน/นักศึกษา (Student) , พรีแลนซ์/อาชีพอิสระ(Freelance) , พนักงานบริษัท(Employee) ,เจ้าของธุรกิจส่วนตัว(Self-employed) ,ว่างงาน(unemployed) , ข้าราชการ(Official) และ แม่บ้าน/พ่อบ้าน(Maid) ตามลำดับ โดยจะมีกราฟสีแดง(red) เป็นกราฟของจำนวนชั่วโมงในการใช้อินเตอร์เน็ตในปี2019 และ กราฟสีม่วง (purple) เป็นกราฟของจำนวนชั่วโมงในการใช้อินเตอร์เน็ตในปี2020

ซึ่งจากการเมื่อจำแนกตามอาชีพจะเห็นได้ว่านักเรียน/นักศึกษามีชั่วโมงการใช้อินเตอร์เน็ตมากที่สุดโดยเพิ่มขึ้นจากปี2019เป็นอย่างมาก ซึ่งนั้นเป็นผลมาจากการณ์โควิดที่มีแพร่ระบาดอยู่ในขณะนี้ จึงทำให้ต้องมีการเรียนและการประชุมงานผ่านทางช่องทางออนไลน์เป็นส่วนมาก

dataList[4]													
Year	Social Media	Searching Information	Receive-send E-mail	Watching TV/Listening Music	Buying goods and service	Download softwares/songs/movies/games	Financial transactions	Playing online games	E-book	E-learning	Finding jobs/Job application	Booking/buying concert/movie ticket	Buying insurance
0	2017.0	86.9	86.5	70.5	60.7	50.8		47.4	45.2	35.3	30.8	28.3	16.0
1	2018.0	93.6	70.8	74.2	60.7	51.3		43.2	49.2	41.0	48.3	28.5	13.9
2	2019.0	91.2	70.7	62.5	71.2	57.0		28.4	-	34.1	57.1	24.4	13.2
3	2020.0	95.3	82.2	69.0	85.0	67.3		53.0	56.5	56.8	64.2	57.5	28.1
Booking/buying ticket	Selling goods and service	Watching stock exchange	Paying taxes	Booking the hotel	Receive-send document	Ordering Foods	Calling taxi	Sending massage	Live application	Paying goods and service	Working via online meeting	Buying insurance	
14	13.7	12.9	11.1	11.0	7.7	6.8	4.8	-	-	-	-	-	-
19.3	24.5	15.1	14.1	20.7	7.2	11.4	12.1	50.6	31.6	-	-	-	-
-	15.9	12.6	20.1	25.9	18.2	26.5	21.4	50	29.6	60.6	-	-	-
-	26.6	14.4	25.7	30.6	39.0	49.8	32.0	77.8	40.4	-	36.4	-	16.4

ทำการแสดงข้อมูลใน dataList[4] ออกมาก็งหมด

```
found_nan_y2017 = ['Sending massage', 'Live application', 'Paying goods and service', 'Working via online meeting', 'Buying insurance']
df4_y2017 = dataList[4][dataList[4]['Year']==2017].drop(found_nan_y2017, axis=1)
found_nan_y2018 = ['Paying goods and service', 'Working via online meeting', 'Buying insurance']
df4_y2018 = dataList[4][dataList[4]['Year']==2018].drop(found_nan_y2018, axis=1)
found_nan_y2019 = ['Financial transactions', 'Booking/buying ticket', 'Working via online meeting', 'Buying insurance']
df4_y2019 = dataList[4][dataList[4]['Year']==2019].drop(found_nan_y2019, axis=1)
found_nan_y2020 = ['Booking/buying ticket', 'Paying goods and service']
df4_y2020 = dataList[4][dataList[4]['Year']==2020].drop(found_nan_y2020, axis=1)
```

สร้างตัวแปรในการเก็บอาร์เรย์พร้อมทั้ง DataFrame ขึ้นมาใหม่ โดย :

- found_nan_y2017 เก็บอาร์เรย์ที่มีค่าข้างในเป็น ['Sending massage', 'Live application', 'Paying goods and service', 'Working via online meeting', 'Buying insurance'] และทำการสร้าง DataFrame (df4_y2017) ขึ้นมาใหม่ เพื่อเก็บค่าของ dataList[4] เฉพาะปี 2017 โดยข้อมูลนี้จะทำการดร็อปคอลัมน์ที่มีชื่อเหมือนในอาร์เรย์ found_nan_y2017 ออกไป
- found_nan_y2018 เก็บอาร์เรย์ที่มีค่าข้างในเป็น ['Paying goods and service', 'Working via online meeting', 'Buying insurance'] และทำการสร้าง DataFrame (df4_y2018) ขึ้นมาใหม่ เพื่อเก็บค่าของ dataList[4] เฉพาะปี 2018 โดยข้อมูลนี้จะทำการดร็อปคอลัมน์ที่มีชื่อเหมือนในอาร์เรย์ found_nan_y2018 ออกไป
- found_nan_y2019 เก็บอาร์เรย์ที่มีค่าข้างในเป็น ['Financial transactions', 'Booking/buying ticket', 'Working via online meeting', 'Buying insurance'] และทำการสร้าง DataFrame (df4_y2019) ขึ้นมาใหม่ เพื่อเก็บค่าของ dataList[4] เฉพาะปี 2019 โดยข้อมูลนี้จะทำการดร็อปคอลัมน์ที่มีชื่อเหมือนในอาร์เรย์ found_nan_y2019 ออกไป
- found_nan_y2020 เก็บอาร์เรย์ที่มีค่าข้างในเป็น ['Booking/buying ticket', 'Paying goods and service'] และทำการสร้าง DataFrame (df4_y2020) ขึ้นมาใหม่ เพื่อเก็บค่าของ dataList[4] เฉพาะปี 2020 โดยข้อมูลนี้จะทำการดร็อปคอลัมน์ที่มีชื่อเหมือนในอาร์เรย์ found_nan_y2020 ออกไป

```

df4_x2017 = np.array(df4_y2017.columns.drop('Year'))
df4_y2017 = np.array(df4_y2017.drop('Year', axis=1)).reshape(len(df4_x2017))
df4_x2018 = np.array(df4_y2018.columns.drop('Year'))
df4_y2018 = np.array(df4_y2018.drop('Year', axis=1)).reshape(len(df4_x2018))
df4_x2019 = np.array(df4_y2019.columns.drop('Year'))
df4_y2019 = np.array(df4_y2019.drop('Year', axis=1)).reshape(len(df4_x2019))
df4_x2020 = np.array(df4_y2020.columns.drop('Year'))
df4_y2020 = np.array(df4_y2020.drop('Year', axis=1)).reshape(len(df4_x2020))

```

ทำการสร้างอาร์เรย์ใหม่ด้วย numpy.array โดยให้ :

- df4_x2017 เก็บข้อมูลคอลัมน์ใน df4_y2017 โดยจะดรอปค่าคอลัมน์ Year ทิ้ง
- df4_x2018 เก็บข้อมูลคอลัมน์ใน df4_y2018 โดยจะดรอปค่าคอลัมน์ Year ทิ้ง
- df4_x2019 เก็บข้อมูลคอลัมน์ใน df4_y2019 โดยจะดรอปค่าคอลัมน์ Year ทิ้ง
- df4_x2020 เก็บข้อมูลคอลัมน์ใน df4_y2020 โดยจะดรอปค่าคอลัมน์ Year ทิ้ง

จากนั้นทำการต่อคอลัมน์ Year ใน DataFrame df4_y2017, df4_y2018, df4_y2019 และ df4_y2020 แล้วทำการจัดเรียงใหม่ โดยใช้ method reshape โดยให้มีมิติเหมือนกับอาร์เรย์ df4_x2017, df4_x2018, df4_x2019 และ df4_x2020 ตามลำดับ

```

new_df4_2017 = pd.DataFrame({'Activity':df4_x2017, 'Rate':df4_y2017})
new_df4_2018 = pd.DataFrame({'Activity':df4_x2018, 'Rate':df4_y2018})
new_df4_2019 = pd.DataFrame({'Activity':df4_x2019, 'Rate':df4_y2019})
new_df4_2020 = pd.DataFrame({'Activity':df4_x2020, 'Rate':df4_y2020})

```

ทำการสร้าง DataFrame ขึ้นมาใหม่ โดยกำหนดคอลัมน์ คือ Activity และ Rate

- new_df4_2017 เก็บข้อมูลโดยคอลัมน์ Activity ได้ข้อมูลจากอาร์เรย์ df4_x2017 ส่วน Rate ได้ข้อมูลจากอาร์เรย์ df4_y2017
- new_df4_2018 เก็บข้อมูลโดยคอลัมน์ Activity ได้ข้อมูลจากอาร์เรย์ df4_x2018 ส่วน Rate ได้ข้อมูลจากอาร์เรย์ df4_y2018
- new_df4_2019 เก็บข้อมูลโดยคอลัมน์ Activity ได้ข้อมูลจากอาร์เรย์ df4_x2019 ส่วน Rate ได้ข้อมูลจากอาร์เรย์ df4_y2019
- new_df4_2020 เก็บข้อมูลโดยคอลัมน์ Activity ได้ข้อมูลจากอาร์เรย์ df4_x2020 ส่วน Rate ได้ข้อมูลจากอาร์เรย์ df4_y2020

```

new_df4_2017_sorted = new_df4_2017.sort_values('Rate', ascending=False)
new_df4_2018_sorted = new_df4_2018.sort_values('Rate', ascending=False)
new_df4_2019_sorted = new_df4_2019.sort_values('Rate', ascending=False)
new_df4_2020_sorted = new_df4_2020.sort_values('Rate', ascending=False)

```

ทำการสร้าง DataFrame ขึ้นมาใหม่ โดย :

- new_df4_2017_sorted เก็บค่าข้อมูลการเรียงค่าจากค่ามากไปน้อยของ colum น์ Rate ใน DataFrame (new_df4_2017)
- new_df4_2018_sorted เก็บค่าข้อมูลการเรียงค่าจากค่ามากไปน้อยของ colum น์ Rate ใน DataFrame (new_df4_2018)
- new_df4_2019_sorted เก็บค่าข้อมูลการเรียงค่าจากค่ามากไปน้อยของ colum น์ Rate ใน DataFrame (new_df4_2019)
- new_df4_2020_sorted เก็บค่าข้อมูลการเรียงค่าจากค่ามากไปน้อยของ colum น์ Rate ใน DataFrame (new_df4_2020)

```

print(new_df4_2017_sorted)
print()
print(new_df4_2018_sorted)
print()
print(new_df4_2019_sorted)
print()
print(new_df4_2020_sorted)

```

	Activity	Rate		Activity	Rate
0	Social Media	86.9	0	Social Media	91.2
1	Searching Information	86.5	3	Watching TV/listening Music	71.2
2	Receive-send E-mail	70.5	1	Searching Information	70.7
3	Watching TV/Listening Music	60.7	2	Receive-send E-mail	62.5
4	Buying goods and service	50.8	20	Paying goods and service	60.6
5	Downloading softwares/songs/movies/games	47.4	7	E-book	57.1
6	Financial transactions	45.2	4	Buying goods and service	57
7	Playing online games	35.3	18	Sending massage	50
8	E-book	30.8	6	Playing online games	34.1
9	E-learning	28.3	19	Live application	29.6
10	Finding jobs/Job application	16	5	Downloading softwares/songs/movies/games	28.4
11	Booking/buying concert/movie ticket	14.6	16	Ordering Foods	26.5
12	Booking/buying ticket	14	10	Booking/buying concert/movie ticket	26.1
13	Selling goods and service	13.7	14	Booking the hotel	25.9
14	Watching stock exchange	12.9	8	E-learning	24.4
15	Paying taxes	11.1	17	Calling taxi	21.4
16	Booking the hotel	11	13	Paying taxes	20.1
17	Receive-send document	7.7	15	Receive-send document	18.2
18	Ordering Foods	6.8	11	Selling goods and service	15.9
19	Calling taxi	4.8	9	Finding jobs/Job application	13.2
			12	Watching stock exchange	12.6
	Activity	Rate		Activity	Rate
0	Social Media	93.6	0	Social Media	91.2
2	Receive-send E-mail	74.2	3	Watching TV/listening Music	71.2
1	Searching Information	70.8	1	Searching Information	70.7
3	Watching TV/Listening Music	60.7	2	Receive-send E-mail	62.5
4	Buying goods and service	51.3	20	Paying goods and service	60.6
20	Sending massage	50.6	7	E-book	57.1
6	Financial transactions	49.2	4	Buying goods and service	57
8	E-book	48.3	18	Sending massage	50
5	Downloading softwares/songs/movies/games	43.2	6	Playing online games	34.1
7	Playing online games	41	19	Live application	29.6
21	Live application	31.6	5	Downloading softwares/songs/movies/games	28.4
9	E-learning	28.5	16	Ordering Foods	26.5
13	Selling goods and service	24.5	10	Booking/buying concert/movie ticket	26.1
11	Booking/buying concert/movie ticket	21.7	14	Booking the hotel	25.9
16	Booking the hotel	20.7	8	E-learning	24.4
12	Booking/buying ticket	19.3	17	Calling taxi	21.4
15	Watching stock exchange	15.1	13	Paying taxes	20.1
10	Paying taxes	14.1	15	Receive-send document	18.2
19	Calling taxi	12.1	11	Selling goods and service	15.9
18	Ordering Foods	11.4	9	Finding jobs/Job application	13.2
17	Receive-send document	7.2	12	Watching stock exchange	12.6

ทำการแสดง DataFrame ที่ได้รับการเรียงค่าจากมากไปน้อยออกมา ได้แก่

new_df4_2017_sorted, new_df4_2018_sorted, new_df4_2019_sorted และ new_df4_2020_sorted

```

plt.rcParams['figure.figsize'] = 10, 10
sns.barplot(x='Rate', y='Activity', data=new_df4_2017_sorted)

for index, value in enumerate(new_df4_2017_sorted['Rate']):
    plt.text(value, index, str(value))

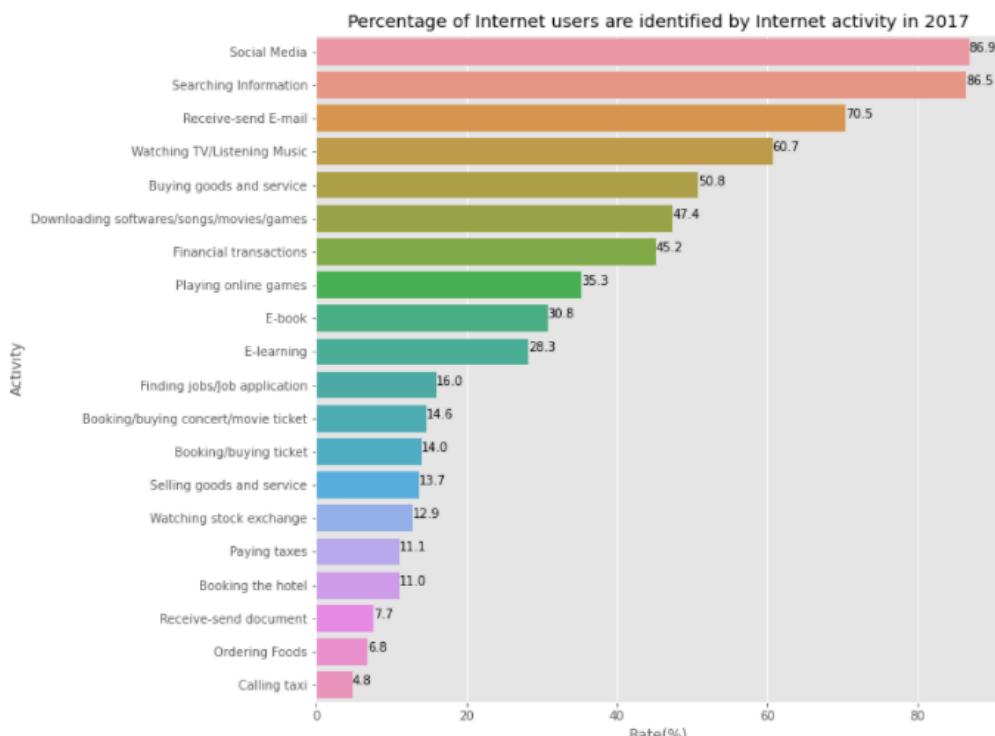
plt.title("Percentage of Internet users are identified by Internet activity in 2017")
plt.xlabel("Rate(%)")
plt.ylabel("Activity")
plt.show()

```

ใช้ฟังก์ชัน plt.rcParams['figure.figsize'] ในการกำหนดขนาดรูปของกราฟเท่ากับ 10*10 ใช้ฟังก์ชัน sns.barplot() ในการ plot กราฟแท่ง โดยใช้ข้อมูลของอาร์เรย์ new_df4_2017_sorted ที่เตรียมไว้มาทำการ plot ซึ่งแกน x คือ คอลัมน์ Activity ส่วนแกน y คือ คอลัมน์ Rate และใช้ plt.text() ในการใส่ค่าของ Rate ในกราฟแต่ละแท่ง

ใช้ plt.title , plt.xlabel , plt.ylabel ในการกำหนดชื่อ หัวข้อกราฟ ชื่อของกราฟแกน x ชื่อของกราฟแกน y ตามลำดับ และใช้ plt.show() ในการแสดงผลลัพธ์ทั้งหมด

ผลลัพธ์



จากการจะเห็นว่าในปี 2017 โซเชียลมีเดียเป็นกิจกรรมที่ใช้งานผ่านทางอินเตอร์เน็ตที่ผู้ใช้นิยมมากที่สุด คิดเป็น 86.9% รองลงมาเป็นการค้นหาข้อมูลคิดเป็น 86.5% และ อันดับสามเป็นการรับ-ส่ง อีเมล คิดเป็น 70.5%

```

plt.rcParams['figure.figsize'] = 10, 10
sns.barplot(x='Rate', y='Activity', data=new_df4_2018_sorted, palette='viridis')

for index, value in enumerate(new_df4_2018_sorted['Rate']):
    plt.text(value, index, str(value))

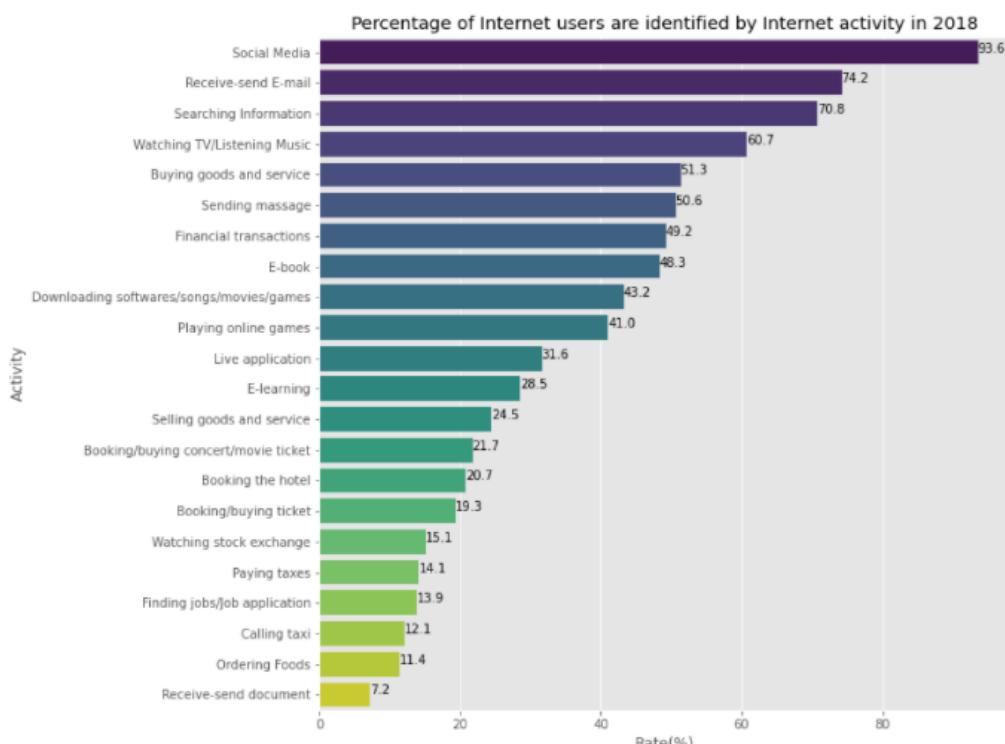
plt.title("Percentage of Internet users are identified by Internet activity in 2018")
plt.xlabel("Rate(%)")
plt.ylabel("Activity")
plt.show()

```

ใช้ฟังก์ชัน rcParam['fugure.figsize'] ในการกำหนดขนาดรูปของกราฟเท่ากับ 10*10 ใช้ฟังก์ชัน sns.barplot() ในการ plot กราฟแท่ง โดยใช้ข้อมูลของอาร์เรย์ new_df4_2018_sorted ที่เตรียมไว้มาทำการ plot ซึ่งแกน x คือ คอลัมน์ Activity ส่วนแกน y คือ คอลัมน์ Rate และใช้ plt.text() ในการใส่ค่าของ Rate ในกราฟแต่ละแท่ง

ใช้ plt.title , plt.xlabel , plt.ylabel ในการกำหนดชื่อ หัวข้อกราฟ ชื่อของกราฟแกน x ชื่อของกราฟแกน y ตามลำดับ และใช้ plt.show() ในการแสดงผลลัพธ์ทั้งหมด

ผลลัพธ์



จากการจะเห็นว่าในปี 2018 โซเชียลมีเดียเป็นกิจกรรมที่ใช้งานผ่านทางอินเตอร์เน็ตที่ผู้ใช้นิยมมากที่สุด เช่นเดียวกับปี 2017 ซึ่งคิดเป็น 93.6% รองลงมาเป็นการรับ-ส่งอีเมล์ที่ขึ้นมาจากอันดับสามในปี ที่แล้ว คิดเป็น 74.2% การค้นหาข้อมูลตกลงไปอยู่เป็นอันดับที่สาม คิดเป็น 70.8%

```
[ plt.rcParams['figure.figsize'] = 10, 10
sns.barplot(x='Rate', y='Activity', data=new_df4_2019_sorted, palette='flare')

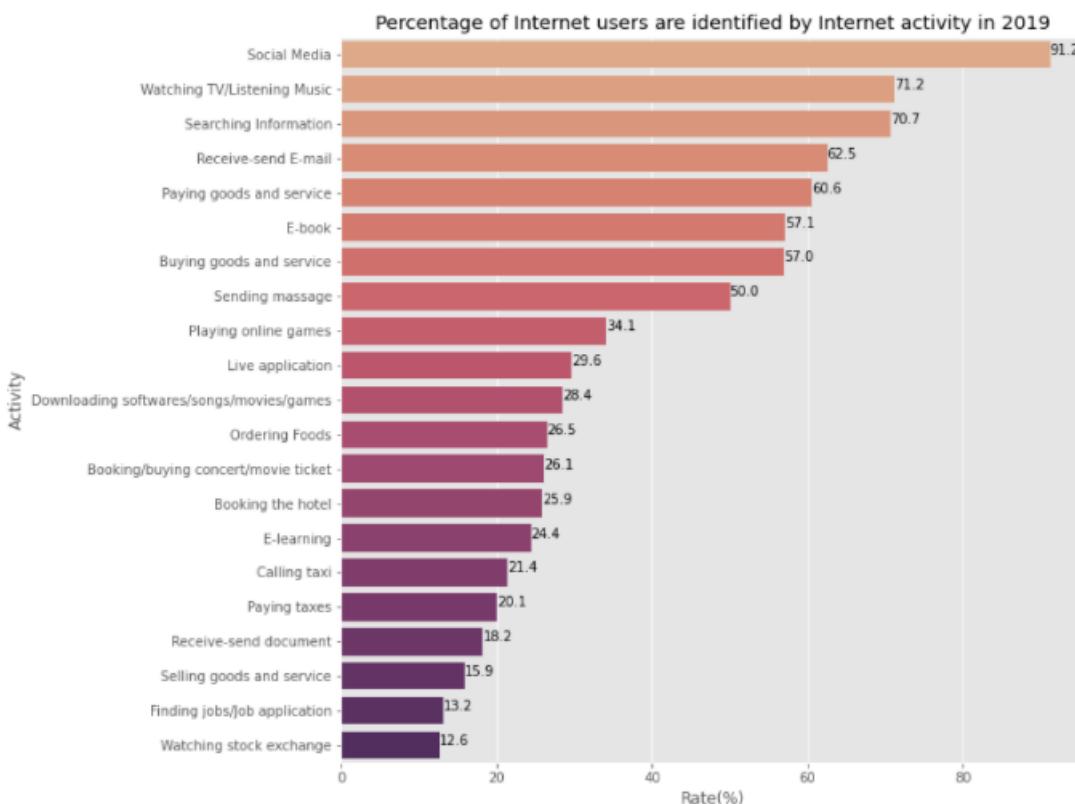
for index, value in enumerate(new_df4_2019_sorted['Rate']):
    plt.text(value, index, str(value))

plt.title("Percentage of Internet users are identified by Internet activity in 2019")
plt.xlabel("Rate(%)")
plt.ylabel("Activity")
plt.show()
```

ใช้ฟังก์ชัน rcParam['fugure.figsize'] ในการกำหนดขนาดรูปของกราฟเท่ากับ 10*10 ใช้ฟังก์ชัน sns.barplot() ในการ plot กราฟ แต่ง โดยใช้ข้อมูลของ อาร์เรย์ new_df4_2019_sorted ที่เตรียมไว้มาทำการ plot ซึ่งแกน x คือ คอลัมน์ Activity ส่วนแกน y คือ คอลัมน์ Rate และใช้ plt.text() ในการใส่ค่าของ Rate ในกราฟแต่ละแท่ง

ใช้ plt.title , plt.xlabel , plt.ylabel ในการกำหนดชื่อ หัวข้อกราฟ ชื่อของกราฟแกน x ตามลำดับ และใช้ plt.show() ในการแสดงผลลัพธ์ทั้งหมด

ผลลัพธ์



จากการจะเห็นว่าในปี 2019 โซเชียลมีเดียยังคงเป็นกิจกรรมที่ใช้งานผ่านทางอินเตอร์เน็ตที่ผู้ใช้尼ยมมากที่สุดเหมือนสองปีที่ผ่านมา คิดเป็น 91.2% รองลงมาเป็นการดูหนังฟังเพลงคิดเป็น 71.2% และการค้นหาข้อมูลยังคงอยู่เป็นอันดับที่สามเหมือนปี 2018 คิดเป็น 70.7%

```

plt.rcParams['figure.figsize'] = 10, 10
sns.barplot(x='Rate', y='Activity', data=new_df4_2020_sorted, palette='crest')

for index, value in enumerate(new_df4_2020_sorted['Rate']):
    plt.text(value, index, str(value))

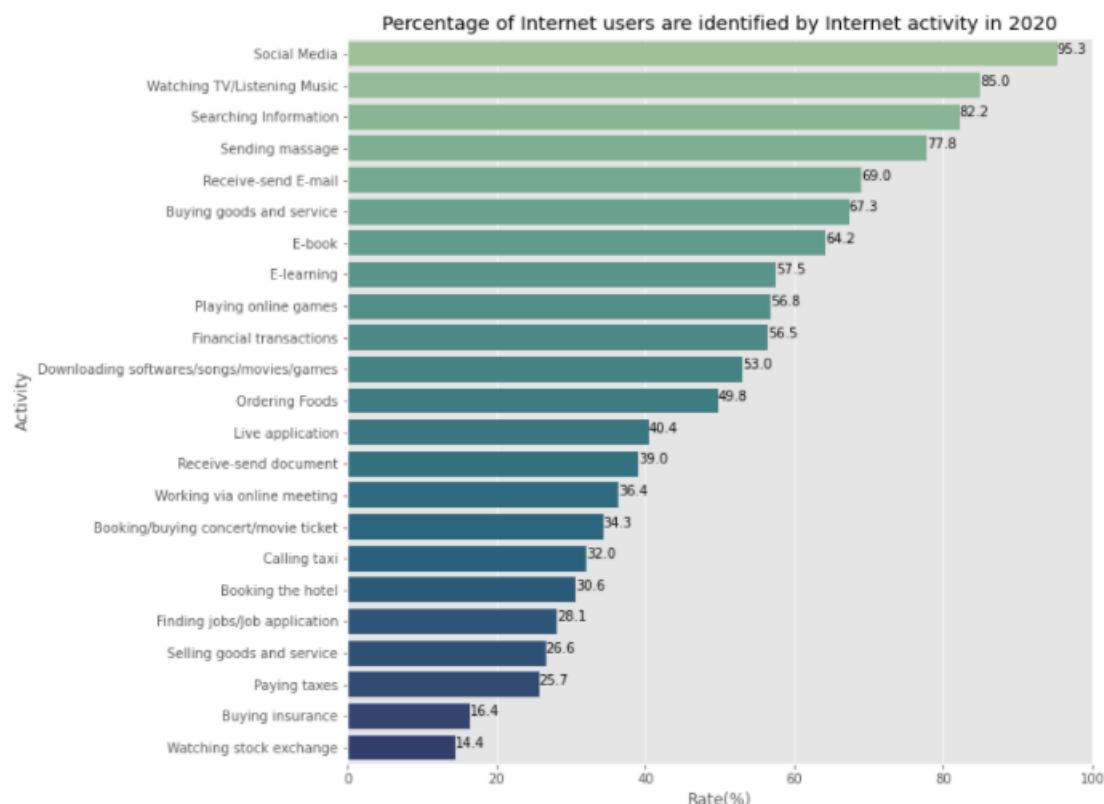
plt.title("Percentage of Internet users are identified by Internet activity in 2020")
plt.xlabel("Rate(%)")
plt.ylabel("Activity")
plt.show()

```

ใช้ฟังก์ชัน rcParam['fugure.figsize'] ในการกำหนดขนาดรูปของกราฟเท่ากับ 10*10 ใช้ฟังก์ชัน sns.barplot() ในการ plot กราฟ แต่ง โดยใช้ข้อมูลของ อาร์เรย์ new_df4_2020_sorted ที่เตรียมไว้มาทำการ plot ซึ่งแกน x คือ คอลัมน์ Activity ส่วนแกน y คือ คอลัมน์ Rate และใช้ plt.text() ในการใส่ค่าของ Rate ในกราฟแต่ละแท่ง

ใช้ plt.title , plt.xlabel , plt.ylabel ในการกำหนดชื่อ หัวข้อกราฟ ชื่อของกราฟแกน x ชื่อของกราฟแกน y ตามลำดับ และใช้ plt.show() ในการแสดงผลลัพธ์ทั้งหมด

ผลลัพธ์



จากการจะเห็นว่าในปี 2020 โซเชียลมีเดียยังคงเป็นกิจกรรมที่ใช้งานผ่านทางอินเตอร์เน็ตที่ผู้ใช้นิยมมากที่สุด เช่นเดียวกับปีก่อนๆ ซึ่งคิดเป็น 95.3% รองลงมาเป็นการดูหนังฟังเพลงที่ยังคงอยู่เป็นอันดับสองเมื่อปี 2019 คิดเป็น 85.0% และ การค้นหาข้อมูลยังคงอยู่เป็นอันดับที่สาม เมื่อปี 2018 และ 2019 คิดเป็น 82.2%

```

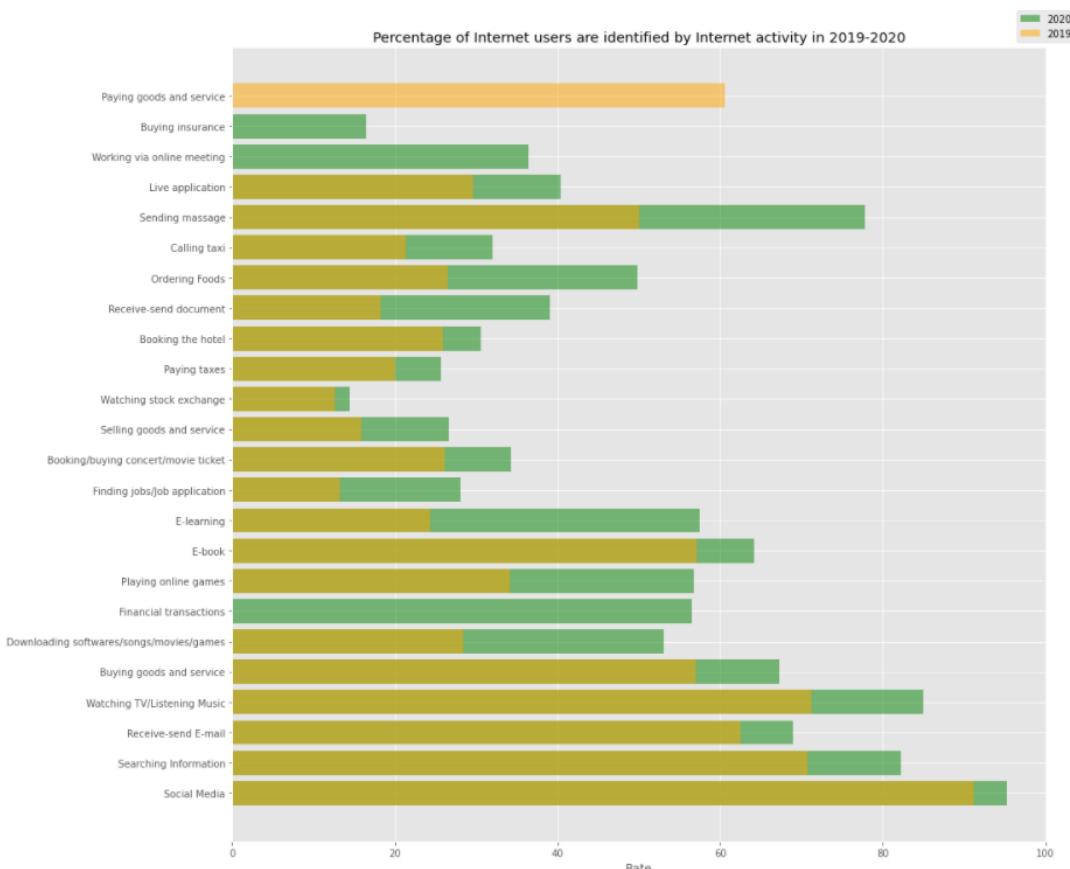
plt.rcParams['figure.figsize'] = 15, 15
plt.title('Percentage of Internet users are identified by Internet activity in 2019-2020')
plt.xlabel('Rate')

plt.barh(df4_x2020, df4_y2020,color='green',alpha=0.5,label='2020')
plt.barh(df4_x2019, df4_y2019,color='orange',alpha=0.5,label='2019')

plt.legend(loc="lower center", bbox_to_anchor=(1.0, 1.0))
plt.show()

```

ผลลัพธ์



เป็นกราฟแสดงจำนวน ร้อยละผู้ใช้อินเตอร์เน็ตจำแนกตามกิจกรรมที่ใช้ทางอินเตอร์เน็ต โดยจะมีกราฟสีเหลือง(yellow) เป็นกราฟของร้อยละผู้ใช้อินเตอร์เน็ตจำแนกตามกิจกรรมที่ใช้ทางอินเตอร์เน็ต ในปี2019 และ กราฟสีเขียว(green) เป็นกราฟของร้อยละผู้ใช้อินเตอร์เน็ตจำแนกตามกิจกรรมที่ใช้ทางอินเตอร์เน็ตในปี2020

จากการ เราจะเห็นว่าผู้ใช้มีการใช้งานอินเตอร์เน็ตทำกิจกรรมทางออนไลน์เพิ่มขึ้นในปี2020 เช่น e-learning ,การซื้อของออนไลน์ และ การใช้โซเชียลมีเดีย ซึ่งสิ่งเหล่านี้เป็นผลที่มาจากการณ์โควิดในปัจจุบันที่ทำให้หลาย ๆ กิจกรรมต้องปรับเปลี่ยนมาทำในรูปแบบออนไลน์กันเพิ่มมากขึ้น เพื่อเป็นการลดการรวมตัวรวมถึงเพื่อลดโอกาสในการแพร่กระจายเชื้อให้แก่ผู้คนรอบข้างด้วย

ทำการแสดงข้อมูลของ dataList[8] เพื่อตรวจสอบความถูกต้องของข้อมูล ในการทำกราฟ

	Year	Platform	Rate
0	2020	Zoom	70.6
1	2020	Line	61.0
2	2020	Microsoft Teams	38.4
3	2020	Google Meet	35.3
4	2020	Google Drive	29.8
5	2020	Team Viewer	17.8
6	2020	One Drive	14.2
7	2020	Linked in	3.6
8	2020	Slack	3.4

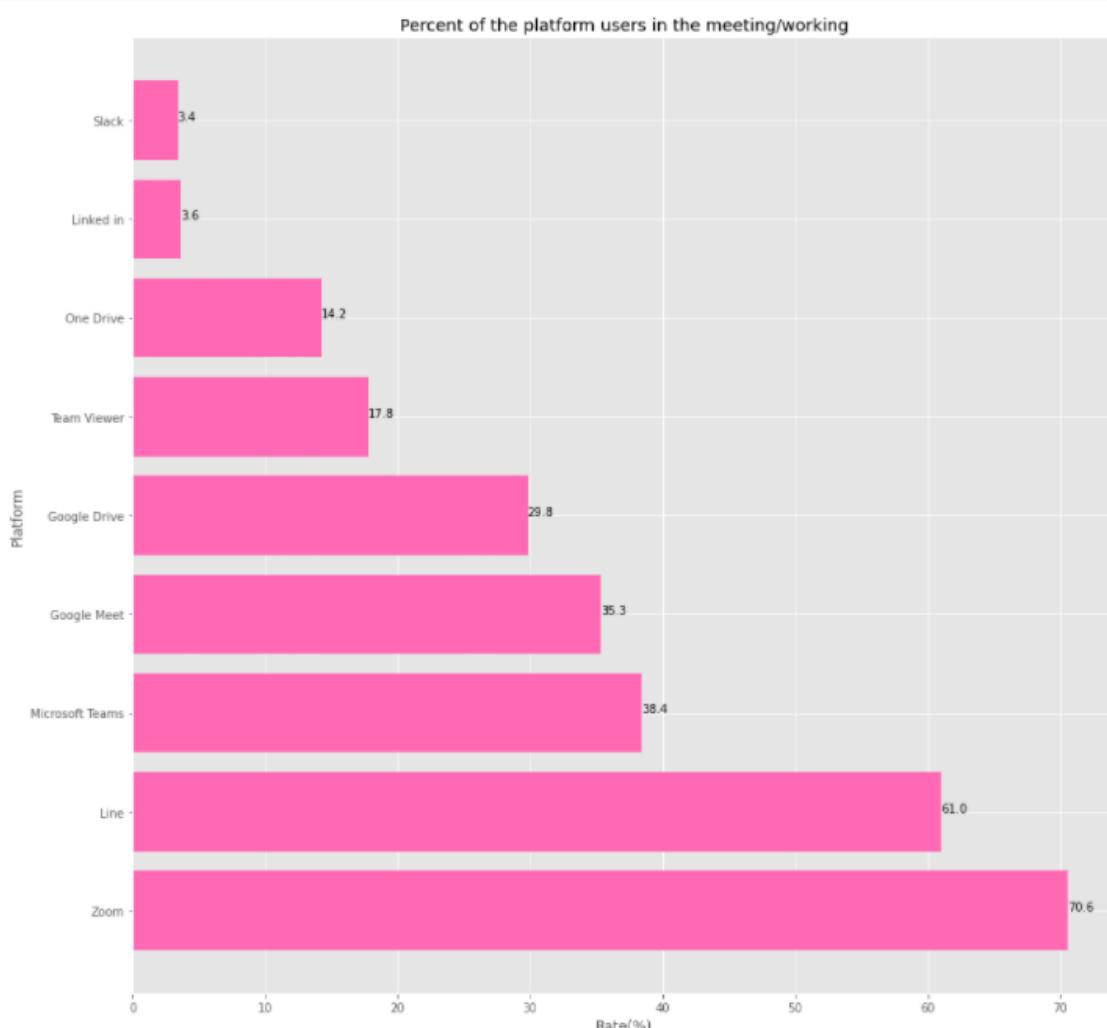
กำหนดให้ ค่า x เป็น คอลัมน์ของ Platform และ ค่า y เป็น คอลัมน์ของ Rate แล้วนำมารีสอร์ท ในฟังก์ชัน plt.barh(x, y, color = 'hotpink') โดยกำหนดขนาดความกว้างและยาวของกราฟเป็น 15*15 พร้อมกับบอกอัตราเฉลี่ยบนแท่งกราฟ

```
[ ] plt.rcParams['figure.figsize'] = 15, 15
style.use("ggplot")
x = np.array(dataList[8]['Platform'])
y = np.array(dataList[8]['Rate'])

plt.barh(x, y,color = 'hotpink')
for index, value in enumerate(dataList[8]['Rate']):
    plt.text(value, index, str(value))

plt.title('Percent of the platform users in the meeting/working')
plt.xlabel('Rate(%)')
plt.ylabel('Platform')
plt.show()
```

จากการที่มีการใช้งาน e-learning เพิ่มมากขึ้นในปี2020และเมื่อเรามาดูถึงแพลตฟอร์มการประชุมและทำงานจะเห็นว่า แพลตฟอร์มที่เป็นที่นิยม 3 อันดับแรก คือ Zoom, Line และ Microsoft Team โดยจะมีการใช้งาน Zoom มากที่สุดถึง 70.6% ตามมาด้วย Line 61% และอันดับสามคือ Microsoft Team 38.4%



แสดงข้อมูลของ dataList[6] เพื่อตรวจสอบความถูกต้องของข้อมูล ในการทำกราฟ

```
[ ] dataList[6]
```

	Year	Shopee	Lazada	Facebook Fanpage	LINE	Instagram
0	2019	75.6	65.5		47.5	38.9
1	2020	91.0	72.9		55.1	41.6

กำหนดให้ค่า df6_x_year เป็น คอลัมน์ของ data7.csv ที่เอาคอลัมน์ Year ออก และ ค่า df6_x_year_2019 และ df6_x_year_2020 เป็นข้อมูลของผู้ใช้อินเตอร์เน็ตในการซื้อสินค้า/บริการทางออนไลน์ในปี 2019 และ ปี 2020 ตามลำดับ นำค่าทั้งสองทำการแปลงเป็น numpy array จากนั้นเอาคอลัมน์ Year ออก และทำการ reshape ให้เป็นอารเรย์ 1 มิติ ที่มีขนาดเหมือนกันกับขนาดของ df6_x_year

```
[ ] df6_x_year = np.array(dataList[6].columns.drop('Year'))
df6_y_year_2019 = np.array(dataList[6][dataList[6]['Year']==2019].drop('Year', axis=1)).reshape(5)
df6_y_year_2020 = np.array(dataList[6][dataList[6]['Year']==2020].drop('Year', axis=1)).reshape(5)
```

กำหนดให้ค่า x เป็น ตำแหน่งของกราฟแท่งตามขนาดของ df6_x_year และ ความกว้างของกราฟเป็น 0.35 จากนั้นสร้างกราฟสองกราฟ โดยที่กราฟแรกมีระยะห่างเป็น $x - \text{width}/2$ (ฝั่งซ้าย) และ กราฟที่สองมีระยะห่างเป็น $x + \text{width}/2$ (ฝั่งขวา) พร้อมกับบอกอัตราเฉลี่ยบนแท่งกราฟ

```
x = np.arange(len(df6_x_year))
width = 0.35
plt.rcParams['figure.figsize'] = 10, 10

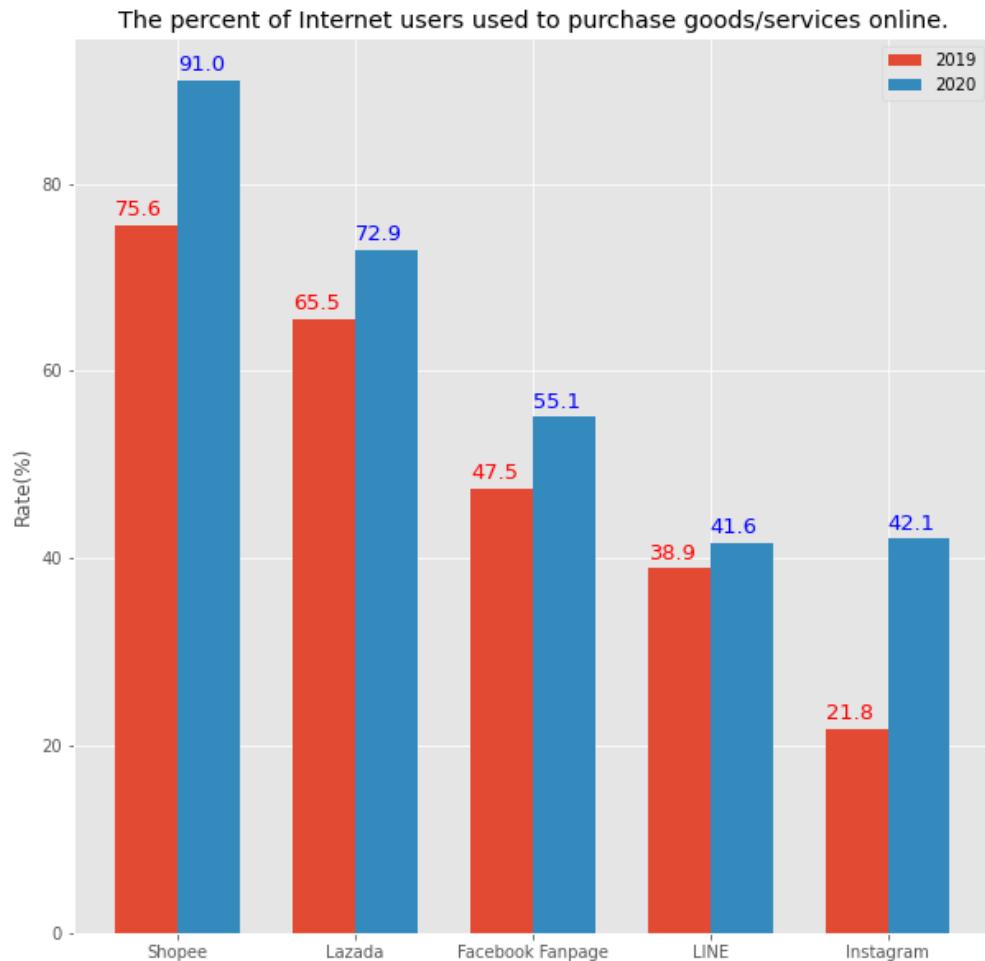
style.use("ggplot")

fig, ax = plt.subplots()
rects1 = ax.bar(x - width/2, df6_y_year_2019, width, label='2019')
rects2 = ax.bar(x + width/2, df6_y_year_2020, width, label='2020')

for index,data in enumerate(df6_y_year_2019):
    plt.text(x=index-width , y =data+1 , s=f'{data}' , fontdict=dict(fontsize=13),color = 'red')
for index,data in enumerate(df6_y_year_2020):
    plt.text(x=index , y =data+1 , s=f'{data}' , fontdict=dict(fontsize=13),color = 'blue')

ax.set_ylabel('Rate(%)')
ax.set_title('The percent of Internet users used to purchase goods/services online.')
ax.set_xticks(x)
ax.set_xticklabels(df6_x_year)
ax.legend()
plt.show()
```

จากการที่มีการใช้งานเพื่อซื้อสินค้าออนไลน์เพิ่มมากขึ้นในปี2020 และเมื่อเราดูถึงแพลตฟอร์มการซื้อสินค้าออนไลน์ จะเห็นว่า ในปี2020มีการใช้งานเพื่อซื้อสินค้าออนไลน์เพิ่มขึ้นจากปี2019ในทุกๆแพลตฟอร์ม



ทำการแสดงข้อมูลของ new_df11

[new_df11](#)

	2021-02	2021-01	2020-12	2020-11	2020-10	2020-09	2020-08	2020-07	2020-06	2020-05	2020-04	2020-03	2020-02	2020-01
List														
Number_accounts_bank	20.71	20.08	20.06	19.96	19.92	19.82	19.90	19.91	19.85	19.83	19.80	19.74	19.60	19.46
Number_accounts_non_bank	90.50	89.36	87.71	85.77	84.12	83.03	81.78	80.91	79.58	78.09	76.17	74.43	72.72	71.39
Total_Number_accounts	111.22	109.44	107.77	105.73	104.04	102.85	101.68	100.81	99.43	97.92	95.97	94.17	92.32	90.84
Topup_value_bank	2901.51	3053.33	3352.65	2763.24	2665.54	2552.70	2306.82	5186.67	2529.23	2346.88	2013.50	4121.53	4504.64	4017.23
Topup_value_non_bank	26169.95	26613.59	29150.41	27456.51	25766.97	24106.49	23271.40	23276.30	23250.76	22258.08	19602.47	20864.84	18057.24	19007.24
Total_Topup_value	29071.46	29666.92	32503.06	30219.75	28432.50	26659.19	25578.23	28462.97	25779.99	24604.95	21615.97	24986.36	22561.88	23024.47
Spending_value_bank	2576.84	2663.32	2569.99	2239.93	2234.07	2266.08	2246.91	4900.03	2165.92	2095.16	2089.26	2707.64	3559.25	4083.25
Spending_value_non_bank	26465.94	26707.89	29107.73	27462.33	25673.40	24220.32	23515.87	23251.02	23463.14	22363.95	20594.73	19790.44	18128.96	18824.00
Total_Spending_value	29042.79	29371.21	31677.72	29702.26	27907.47	26486.40	25762.79	28151.06	25629.06	24459.11	22683.98	22498.07	21688.21	22907.25

ทำการจัดเรียงวันที่ให้ถูกต้อง โดยนำวันที่ 2017-01 ขึ้นมาก่อน

```
new_df11_val = new_df11.loc[:, '2017-01':'2021-02':-1]
new_df11_val
```

แสดงข้อมูลของ new_df11 ที่ทำการเรียงวันที่ถูกต้อง

	2017-01	2017-02	2017-03	2017-04	2017-05	2017-06	2017-07	2017-08	2017-09	2017-10	2017-11	2017-12	2018-01
List													
Number_accounts_bank	2.11	2.10	2.08	2.08	2.09	2.06	2.04	2.08	2.12	2.09	2.39	2.58	2.87
Number_accounts_non_bank	37.72	38.30	39.22	40.01	41.24	42.37	43.40	44.58	46.81	48.68	50.26	51.69	51.83
Total_Number_accounts	39.83	40.41	41.30	42.09	43.34	44.43	45.44	46.65	48.93	50.77	52.66	54.27	54.70
Topup_value_bank	709.75	740.51	957.82	813.27	856.68	962.35	931.30	908.80	964.50	935.33	1104.73	1390.24	1454.87
Topup_value_non_bank	8768.57	8314.10	9538.77	8891.50	9113.55	9076.47	9411.70	9836.21	10233.78	10575.79	10817.06	11736.08	11530.74
Total_Topup_value	9478.33	9054.61	10496.59	9704.77	9970.23	10038.82	10343.00	10745.01	11198.28	11511.12	11921.79	13126.31	12985.61
Spending_value_bank	698.40	719.74	938.71	823.49	860.12	960.35	870.00	897.73	950.92	944.22	981.74	1006.97	861.65
Spending_value_non_bank	8839.02	8284.48	9329.76	9014.14	9137.91	8985.23	9484.26	9808.24	9996.96	10572.59	10782.39	11284.49	11658.27
Total_Spending_value	9537.42	9004.22	10268.47	9837.63	9998.03	9945.58	10354.26	10705.96	10947.88	11516.82	11764.13	12291.46	12519.92

กำหนดให้ค่า new_df11_ind และ new_df11_col เป็น numpy array ของ index และ คอลัมน์ ของ new_df11_val ตามลำดับ

```
[ ] new_df11_ind = np.array(new_df11_val.index)
new_df11_col = np.array(new_df11_val.columns)

print(new_df11_ind)
print(new_df11_col)

['Number_accounts_bank' 'Number_accounts_non_bank' 'Total_Number_accounts'
 'Topup_value_bank' 'Topup_value_non_bank' 'Total_Topup_value'
 'Spending_value_bank' 'Spending_value_non_bank' 'Total_Spending_value']
['2017-01' '2017-02' '2017-03' '2017-04' '2017-05' '2017-06' '2017-07'
 '2017-08' '2017-09' '2017-10' '2017-11' '2017-12' '2018-01' '2018-02'
 '2018-03' '2018-04' '2018-05' '2018-06' '2018-07' '2018-08' '2018-09'
 '2018-10' '2018-11' '2018-12' '2019-01' '2019-02' '2019-03' '2019-04'
 '2019-05' '2019-06' '2019-07' '2019-08' '2019-09' '2019-10' '2019-11'
 '2019-12' '2020-01' '2020-02' '2020-03' '2020-04' '2020-05' '2020-06'
 '2020-07' '2020-08' '2020-09' '2020-10' '2020-11' '2020-12' '2021-01'
 '2021-02']
```

สร้าง list new_df11_data ใช้ในการเก็บข้อมูลของกิจกรรมทางการเงิน เพื่อสร้าง dataframe ใหม่ โดยนำข้อมูลในแต่ละ index เก็บไว้ใน new_df11_data และทำการ reshape ให้เป็นอาร์เรย์ 2 มิติ ที่มีขนาดเหมือนกับ new_df11_ind และ new_df11_col

```
new_df11_data = []
for name in new_df11_ind:
    new_df11_data.append(new_df11_val[new_df11_val.index == name])
new_df11_data = np.array(new_df11_data).reshape(len(new_df11_ind), len(new_df11_col))
```

สร้าง dataframe ใหม่ ชื่อว่า new_df11_data_v2

```
new_df11_data_v2 = pd.DataFrame({'Date':new_df11_col})
for i in range(len(new_df11_ind)):
    new_df11_data_v2[new_df11_ind[i]] = new_df11_data[i]
new_df11_data_v2.head()
```

ทำการแสดงข้อมูล new_df11_data_v2

	Date	Number_accounts_bank	Number_accounts_non_bank	Total_Number_accounts	Topup_value_bank	Topup_value_non_bank	Total_Topup_value
0	2017-01	2.11	37.72	39.83	709.75	8768.57	9478.33
1	2017-02	2.10	38.30	40.41	740.51	8314.10	9054.61
2	2017-03	2.08	39.22	41.30	957.82	9538.77	10496.59
3	2017-04	2.08	40.01	42.09	813.27	8891.50	9704.77
4	2017-05	2.09	41.24	43.34	856.68	9113.55	9970.23

ส่งคอลัมน์ Date ทั้งหมดเข้าพังก์ชัน pd.to_datetime() จะได้อาร์เรย์ของวันที่ทุกແຕว

```
new_df11_data_v2['Date'] = pd.to_datetime(new_df11_data_v2['Date'].astype('string'))
```

เปลี่ยนคอลัมน์ Date เป็น index ของ dataframe

```
new_df11_data_v2.set_index('Date', inplace=True)
```

ແສດງຂໍ້ມູນ new_df11_data_v2

```
new_df11_data_v2.head()
```

	Number_accounts_bank	Number_accounts_non_bank	Total_Number_accounts	Topup_value_bank	Topup_value_non_bank
Date					
2017-01-01	2.11	37.72	39.83	709.75	8768.57
2017-02-01	2.10	38.30	40.41	740.51	8314.10
2017-03-01	2.08	39.22	41.30	957.82	9538.77
2017-04-01	2.08	40.01	42.09	813.27	8891.50
2017-05-01	2.09	41.24	43.34	856.68	9113.55

นำຂໍ້ມູນຕັ້ງແຕ່ຄອລິນ໌ Number_account_bank ຈຶນກື່ງ Total_Number_account

```
df11_plot_data1 = new_df11_data_v2.loc[:, 'Number_accounts_bank':'Total_Number_accounts']
```

ຈາກນັ້ນນຳດ້ວຍແປຣ df11_plot_data1 ມາພຶດຕີເປັນກາຟເສັ້ນ ໂດຍກຳຫັນດັບນາດຄວາມກວ້າງແລະ
ຍາວຂອງກາຟເປັນ 12*6

```
plt.rcParams['figure.figsize'] = 12, 6
style.use("ggplot")
df11_plot_data1.plot()
plt.ylabel('Value')
plt.show()
```

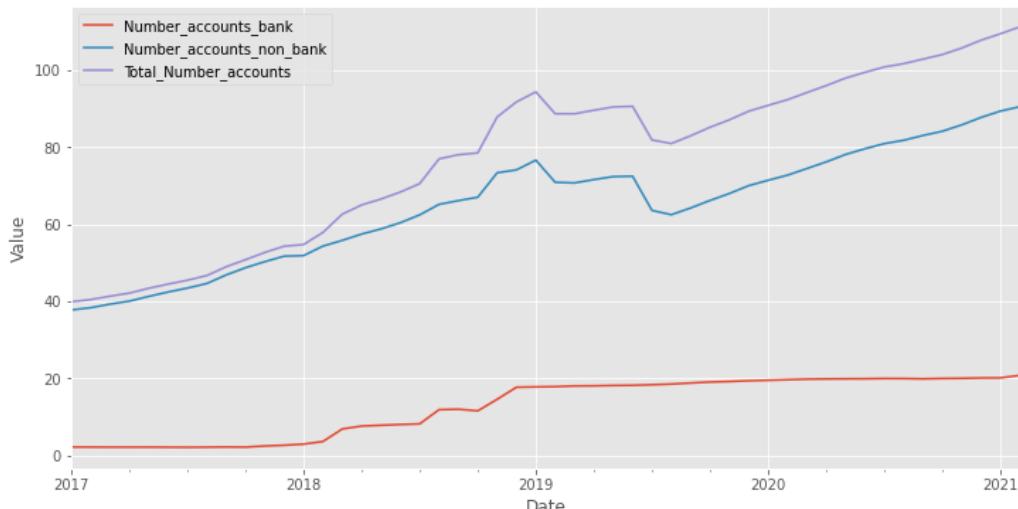
ສ່ວນກາຟທີ່ສອງ ນຳຂໍ້ມູນຕັ້ງແຕ່ຄອລິນ໌ Top_value_bank ກື່ງ Total_Topup_value

```
df11_plot_data2 = new_df11_data_v2.loc[:, 'Topup_value_bank':'Total_Topup_value']
```

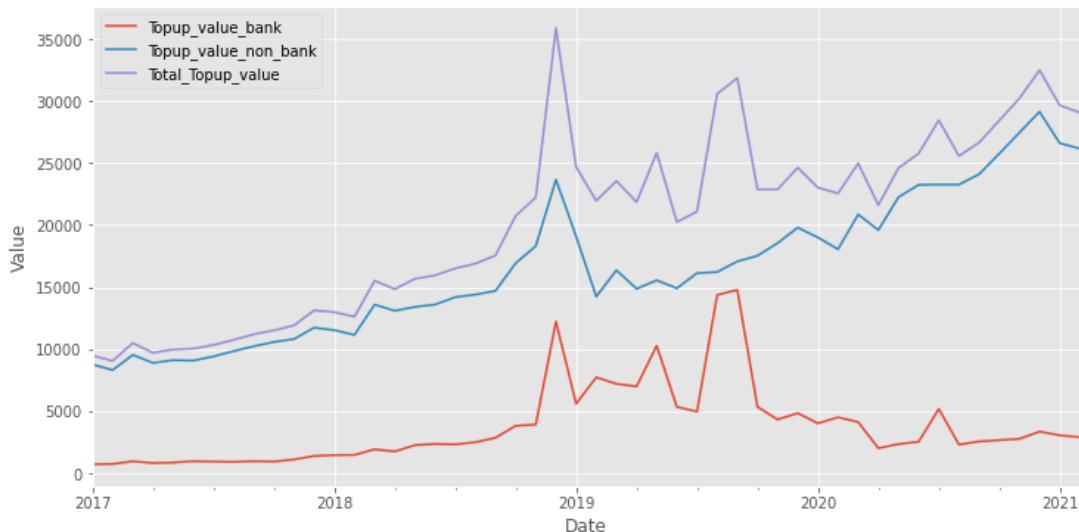
นำตัวแปร df11_plot_data2 มาพล็อตเป็นกราฟเส้น โดยกำหนดขนาดความกว้างและยาวของกราฟเป็น 12*6

```
plt.rcParams['figure.figsize'] = 12, 6
style.use("ggplot")
df11_plot_data2.plot()
plt.ylabel('Value')
plt.show()
```

จากราฟนี้จะเห็นได้ว่าในปี 2020 มีผู้ใช้เงินอิเล็กทรอนิกส์ แบบผู้ให้บริการที่ไม่ใช่สถาบัน การเงินเพิ่มขึ้นสูงกว่าผู้ให้บริการที่เป็นสถาบันการเงิน และเมื่อเราดูถึงรายละเอียดของการใช้เงินจากสถานการณ์โควิด-19 จะพบว่า



หลังจากผ่อนคลายล็อกดาวน์ระลอกแรก มูลค่าในการเติมเงินแบบให้บริการที่ไม่ใช่สถาบัน การเงินเพิ่มสูงขึ้น ส่วนผู้ให้บริการที่เป็นสถาบันการเงินเพิ่มขึ้นเล็กน้อย แต่พอเจอโควิดระลอกสอง ทำให้การเติมเงินทั้งสองแบบลดลงเล็กน้อย



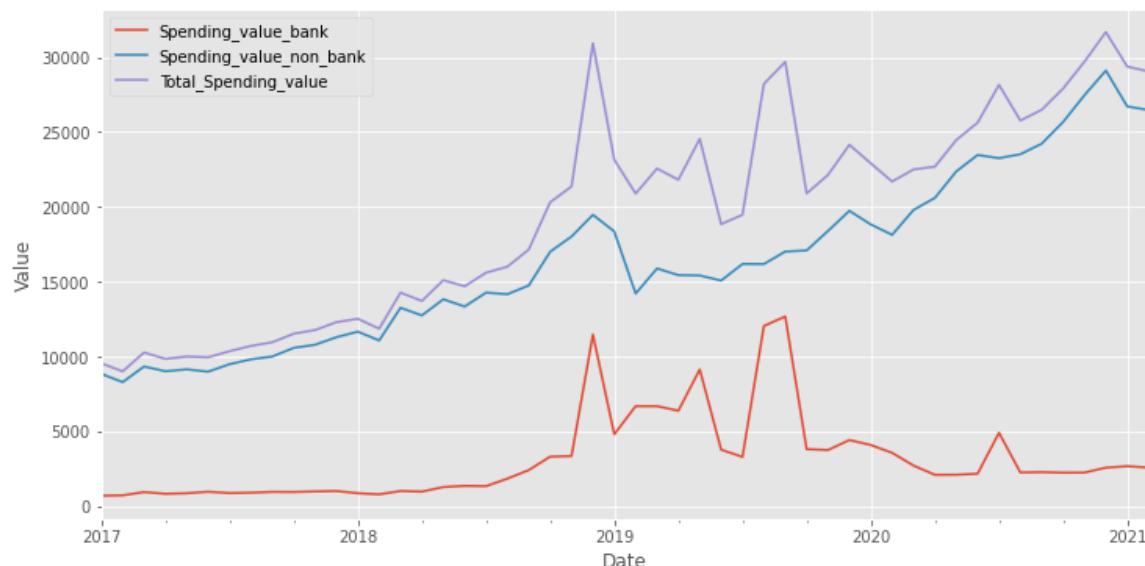
กราฟອันທີສາມ นำຂໍອມມຸລຕັ້ງແຕ່ຄອລັນນີ້ Spending_value_bank ປຶ້ງ Total_Spending_value

```
df11_plot_data3 = new_df11_data_v2.loc[:, 'Spending_value_bank':'Total_Spending_value']
```

นำຕັ້ງແປຣ df11_plot_data3 ມາພລືອຕເປັນກາຮົບເສັ້ນ ໂດຍກຳຫັນດີນາດຄວາມກວ້າງແລະຍາວຂອງ
ກາຮົບເສັ້ນ 12*6

```
plt.rcParams['figure.figsize'] = 12, 6
style.use("ggplot")
df11_plot_data3.plot()
plt.ylabel('Value')
plt.show()
```

จากการพบร่วมกัน บัญชีค่าใช้จ่ายในช่วงสถานการณ์โควิด-19 โดยเฉพาะการใช้จ่ายแบบ
ให้บริการที่ไม่ใช่สถาบันการเงินเพิ่มสูงขึ้น ส่วนการใช้จ่ายแบบให้บริการที่เป็นสถาบันการเงินเพิ่มขึ้น
เล็กน้อยและคงที่ในเดือนถัดไป แต่พอเจอโควิดระลอกสอง ทำให้มูลค่าการใช้จ่ายทั้งสองแบบลดลง
เล็กน้อย



นำข้อมูล new_df12 ทำการจัดเรียงวันที่ให้ถูกต้อง โดยนำวันที่ 2017-01 ขึ้นมาก่อน

```
new_df12_val = new_df12.loc[:, '2017-01':'2021-02':-1]
new_df12_val.head()
```

	2017-01	2017-02	2017-03	2017-04	2017-05	2017-06	2017-07	2017-08	2017-09	2017-10	2017-11	2017-12	2018-01
List													
Check	4897.0	4655.0	5281.0	4553.0	5488.0	5000.0	4678.0	5037.0	4975.0	4592.0	5069.0	5304.0	4939.0
Cheak_Inhouse	1947.0	1802.0	2081.0	1836.0	2255.0	1908.0	1859.0	1957.0	2002.0	1740.0	2044.0	2220.0	1890.0
Cheak_Interbank	2950.0	2853.0	3200.0	2717.0	3233.0	3092.0	2819.0	3080.0	2973.0	2852.0	3025.0	3084.0	3049.0
e-Payments	26754.0	25543.0	31457.0	24517.0	29913.0	28176.0	26987.0	31393.0	29771.0	26827.0	29396.0	28660.0	30873.0
transfer_BAHTNET	20739.0	20464.0	25658.0	19459.0	24357.0	22571.0	21459.0	25719.0	23908.0	20995.0	23374.0	22280.0	24478.0

กำหนดให้ค่า new_df12_ind และ new_df12_col เป็น numpy array ของ index และ คอลัมน์
ของ new_df12_val ตามลำดับ

```
new_df12_ind = np.array(new_df12_val.index)
new_df12_col = np.array(new_df12_val.columns)
```

สร้าง list new_df12_data ใช้ในการเก็บข้อมูลของกิจกรรมทางการเงิน เพื่อสร้าง Dataframe ใหม่ โดยนำข้อมูลในแต่ละ index เก็บไว้ใน new_df12_data และทำการ reshape ให้เป็นอาร์เรย์ 2 มิติ ที่มีขนาดเหมือนกับ new_df12_ind และ new_df12_col

```
new_df12_data = []
for name in new_df12_ind:
    new_df12_data.append(new_df12_val[new_df12_val.index == name])
new_df12_data = np.array(new_df12_data).reshape(len(new_df12_ind), len(new_df12_col))
```

สร้าง dataframe ใหม่ ชื่อว่า new_df12_v2

```
new_df12_v2 = pd.DataFrame({'Date':new_df12_col})
for i in range(len(new_df12_data)):
    new_df12_v2[new_df12_ind[i]] = new_df12_data[i]
new_df12_v2.head()
```

แสดงข้อมูลของ new_df12_v2

	Date	Check	Cheak_Inhouse	Cheak_Interbank	e-Payments	transfer_BAHTNET	transfer_Bulk_Payment	transfer_Direct_credit	transfer_Direct_debit
0	2017-01	4897.0	1947.0	2950.0	26754.0	20739.0	2432.0	1524.0	671.0
1	2017-02	4655.0	1802.0	2853.0	25543.0	20464.0	2202.0	1373.0	599.0
2	2017-03	5281.0	2081.0	3200.0	31457.0	25658.0	2506.0	1545.0	700.0
3	2017-04	4553.0	1836.0	2717.0	24517.0	19459.0	2202.0	1397.0	562.0
4	2017-05	5488.0	2255.0	3233.0	29913.0	24357.0	2319.0	1437.0	596.0

แปลงข้อมูลคอลัมน์ของ Date ให้เป็น String

```
new_df12_v2['Date'] = pd.to_datetime(new_df12_v2['Date'].astype('string'))
```

เปลี่ยนคอลัมน์ Date เป็น index ของ dataframe

```
new_df12_v2.set_index('Date', inplace=True)
```

ແສດງຂໍ້ມູນ new_df12_data_v2

```
new_df12_v2.head()
```

	Check	Cheak_Inhouse	Cheak_Interbank	e-Payments	transfer_BAHTNET	transfer_Bulk_Payment	transfer_Direct_credit	transfer_Direct_debit
Date								
2017-01-01	4897.0	1947.0	2950.0	26754.0	20739.0	2432.0	1524.0	671.0
2017-02-01	4655.0	1802.0	2853.0	25543.0	20464.0	2202.0	1373.0	599.0
2017-03-01	5281.0	2081.0	3200.0	31457.0	25658.0	2506.0	1545.0	700.0
2017-04-01	4553.0	1836.0	2717.0	24517.0	19459.0	2202.0	1397.0	562.0
2017-05-01	5488.0	2255.0	3233.0	29913.0	24357.0	2319.0	1437.0	596.0

ກາຮົບອັນແຮກ ນຳຂໍ້ມູນທີ່ຕ່າງລົມ້ນ e-payments ອື່ນ transfer_BAHTNET ແລະ ຂໍ້ມູນຂອງ

Online_Retail_Funds_Transfer, Intra_bank_transfer ແລະ Payment_cards

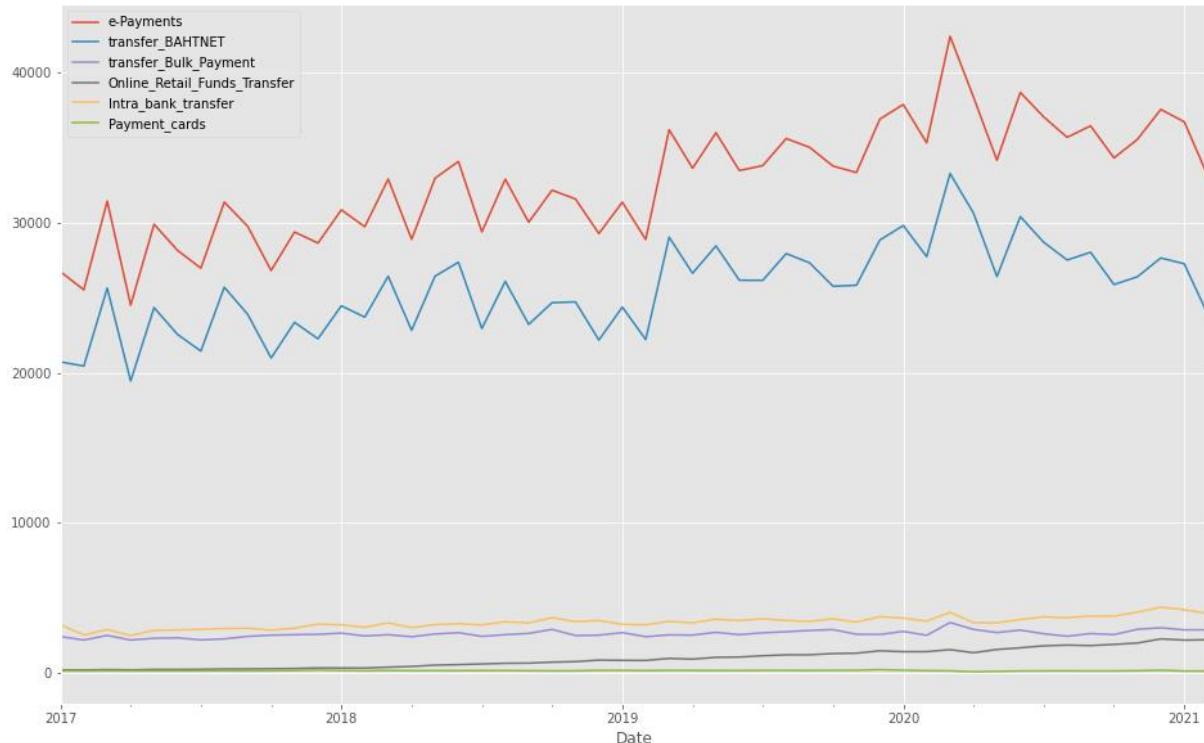
```
plot_data1 = new_df12_v2.loc[:, 'e-Payments':'transfer_Bulk_Payment']
plot_data1 = plot_data1.assign(Online_Retail_Funds_Transfer = new_df12_v2['Online_Retail_Funds_Transfer'])
plot_data1 = plot_data1.assign(Intra_bank_transfer = new_df12_v2['Intra_bank_transfer'])
plot_data1 = plot_data1.assign(Payment_cards = new_df12_v2['Payment_cards'])
plot_data1.head()
```

	e-Payments	transfer_BAHTNET	transfer_Bulk_Payment	Online_Retail_Funds_Transfer	Intra_bank_transfer	Payment_cards
Date						
2017-01-01	26754.0	20739.0	2432.0	201.0	3225.0	147.0
2017-02-01	25543.0	20464.0	2202.0	200.0	2531.0	137.0
2017-03-01	31457.0	25658.0	2506.0	229.0	2903.0	151.0
2017-04-01	24517.0	19459.0	2202.0	205.0	2501.0	140.0
2017-05-01	29913.0	24357.0	2319.0	238.0	2845.0	144.0

ຈາກນັ້ນນຳຂໍ້ມູນຂອງ plot_data1 ມາພຶດວິທີເປັນກາຮົບເສັ້ນ ໂດຍກຳຫົວດ້ານດ້ານຄວາມກວ້າງແລະຍາວ
ຂອງກາຮົບເປັນ 16*10

```
plt.rcParams['figure.figsize'] = 16, 10
style.use("ggplot")
plot_data1.plot()
plt.show()
```

จากราฟนี้พบว่า ปริมาณการชำระเงินผ่านระบบ BAHTNET ลดลงหลังการแพร่ระบาดของโควิดระลอกแรก และต่อเนื่องในเดือนถัดไป ส่วนปริมาณการชำระเงินผ่านการโอนเงินรายย่อยข้ามธนาคารเพิ่มขึ้นเล็กน้อยและปริมาณการชำระเงินผ่านระบบอื่นๆ บังคับที่ในปี 2020 จนถึงปี 2021



สร้าง dataframe ใหม่ ชื่อว่า plot_data2 เพื่อใช้ในการพล็อตกราฟแท่ง

```
plot_data2 = pd.DataFrame({'Date':new_df12_col})
for i in range(len(new_df12_data)):
    plot_data2[new_df12_ind[i]] = new_df12_data[i]
plot_data2.head()
```

	Date	Check	Cheak_Inhouse	Cheak_Interbank	e-Payments	transfer_BAHTNET	transfer_Bulk_Payment	transfer_Direct_credit
0	2017-01	4897.0	1947.0	2950.0	26754.0	20739.0	2432.0	1524.0
1	2017-02	4655.0	1802.0	2853.0	25543.0	20464.0	2202.0	1373.0
2	2017-03	5281.0	2081.0	3200.0	31457.0	25658.0	2506.0	1545.0
3	2017-04	4553.0	1836.0	2717.0	24517.0	19459.0	2202.0	1397.0
4	2017-05	5488.0	2255.0	3233.0	29913.0	24357.0	2319.0	1437.0

กราฟอันที่สอง นำข้อมูลตั้งแต่คอลัมน์ Transfer_sub_cash_ATM ถึง Transfer_sub_cash_across_bank_by_branch จากนั้นเพิ่มคอลัมน์ Date เพื่อใช้ในการระบุวันที่ของข้อมูล

```
new_plot_data2 = plot_data2.loc[:, 'Transfer_sub_cash_ATM':'Transfer_sub_cash_across_bank_by_branch']
new_plot_data2 = new_plot_data2.assign(date = plot_data2['Date'])
new_plot_data2.head()
```

	Transfer_sub_cash_ATM	Transfer_sub_cash_Internet_and_mobile	Transfer_sub_cash_across_bank_by_branch	date
0	74.0	112.0		15.0 2017-01
1	69.0	116.0		15.0 2017-02
2	76.0	136.0		17.0 2017-03
3	61.0	131.0		13.0 2017-04
4	72.0	151.0		15.0 2017-05

ทำการแบ่งข้อมูลออกเป็นสองคอลัมน์ และเอาคอลัมน์ Date และ Month ออก

```
new_plot_data2[['Year', 'Month']] = new_plot_data2['date'].str.split('-', 1, expand=True)
new_plot_data2.drop(['date', 'Month'], axis='columns', inplace=True)
```

แสดงข้อมูลหลังจากแยกข้อมูลเสร็จแล้ว

```
new_plot_data2.head()
```

	Transfer_sub_cash_ATM	Transfer_sub_cash_Internet_and_mobile	Transfer_sub_cash_across_bank_by_branch	Year
0	74.0	112.0		15.0 2017
1	69.0	116.0		15.0 2017
2	76.0	136.0		17.0 2017
3	61.0	131.0		13.0 2017
4	72.0	151.0		15.0 2017

ทำการจัดกลุ่มโดยใช้พังก์ชัน groupby() เพื่อทำให้ข้อมูลอยู่ในรูปลิสต์ที่จำแนกตามปี เก็บในตัวแปรใหม่แยกตามแต่ละข้อมูล ซึ่งใช้รวมกับพังก์ชัน sum() จากนั้นนำข้อมูลตั้งแต่ index ในปี 2020 ถึง 2021

```
new_plot_data2 = new_plot_data2.groupby('Year').sum()
new_plot_data2 = new_plot_data2.loc['2020':'2021', :]
new_plot_data2
```

Year	Transfer_sub_cash_ATM	Transfer_sub_cash_Internet_and_mobile	Transfer_sub_cash_across_bank_by_branch
2020	294.0	20319.0	99.0
2021	45.0	4355.0	15.0

กำหนดให้ plot_data2_x เป็น numpy array ของ index ในช่วงปี 2020 ถึง 2021 plot_data2_y1, plot_data2_y2, plot_data2_y3 เป็น numpy array ข้อมูลของ Transfer_sub_cash_ATM, Transfer_sub_cash_Internet_and_mobile และ Transfer_sub_cash_bank_by_branch ตามลำดับ

```
plot_data2_x = np.array(new_plot_data2.index)
plot_data2_y1 = np.array(new_plot_data2['Transfer_sub_cash_ATM'])
plot_data2_y2 = np.array(new_plot_data2['Transfer_sub_cash_Internet_and_mobile'])
plot_data2_y3 = np.array(new_plot_data2['Transfer_sub_cash_across_bank_by_branch'])
```

กำหนดให้ค่า X_axis เป็น ตำแหน่งของกราฟแท่งตามขนาดของ plot_data2_x และ ความกว้างของกราฟเป็น 0.25 จากนั้นสร้างกราฟสามกราฟ โดยที่กราฟแรกมีระยะห่างเป็น X_axis (ผังซ้าย) กราฟที่สองมีระยะห่างเป็น X_axis + width (ตรงกลาง) และกราฟที่สองมีระยะห่างเป็น X_axis + width*2 พร้อมกับบอกปริมาณการชำระเงินบนแท่งกราฟ โดยกำหนดขนาดความกว้างและยาวของกราฟเป็น 10*6

```
plt.rcParams['figure.figsize'] = 10, 6
style.use("ggplot")

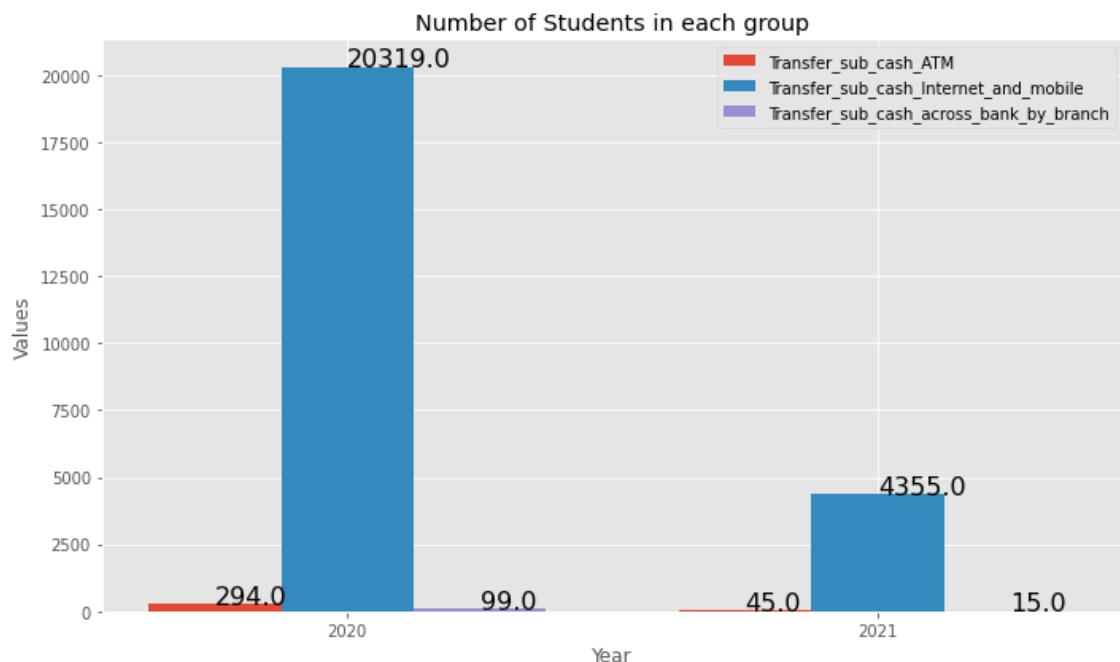
X_axis = np.arange(len(plot_data2_x))
width = 0.25

plt.bar(X_axis, plot_data2_y1, width, label = 'Transfer_sub_cash_ATM')
plt.bar(X_axis+width, plot_data2_y2, width, label = 'Transfer_sub_cash_Internet_and_mobile')
plt.bar(X_axis+width*2, plot_data2_y3, width, label = 'Transfer_sub_cash_across_bank_by_branch')

for index,data in enumerate(plot_data2_y1):
    plt.text(x=index , y =data+1 , s=f"{data}" , fontdict=dict(fontsize=16))
for index,data in enumerate(plot_data2_y2):
    plt.text(x=index+width , y =data+1 , s=f"{data}" , fontdict=dict(fontsize=16))
for index,data in enumerate(plot_data2_y3):
    plt.text(x=index+width*2 , y =data+1 , s=f"{data}" , fontdict=dict(fontsize=16))

plt.xticks(X_axis+width, plot_data2_x)
plt.xlabel("Year")
plt.ylabel("Values")
plt.title("Number of Students in each group")
plt.tight_layout()
plt.legend()
plt.show()
```

จากการแพร่ระบาดที่รุนแรงในสถานการณ์โควิด-19 การโอนเงินรายย่อยข้ามธนาคารผ่านอินเตอร์เน็ตและโทรศัพท์เคลื่อนที่ เพิ่มมากขึ้นในปี 2020 แต่การโอนเงินรายย่อยข้ามธนาคารผ่านเครือข่ายอีเมลและผ่านสาขาห้องลับเป็นอย่างมาก อาจเนื่องจากมาตรการล็อกดาวน์ในประเทศทำให้ไม่สามารถออกจากบ้านได้



แสดงข้อมูลของ new_df13 เพื่อทำการตรวจสอบความถูกต้องของข้อมูล

new_df13	2021-02	2021-01	2020-12	2020-11	2020-10	2020-09	2020-08	2020-07	2020-06
List									
Check	5758.0	6161.0	6756.0	6638.0	6311.0	7180.0	6875.0	6576.0	6825.0
Cheak_Inhouse	2459.0	2631.0	2873.0	2754.0	2666.0	3187.0	2946.0	2787.0	2794.0
Cheak_Interbank	3299.0	3530.0	3883.0	3884.0	3645.0	3993.0	3929.0	3789.0	4031.0
e-Payments	1396637.0	1401565.0	1416346.0	1333362.0	1249594.0	1184387.0	1187397.0	1164781.0	1116023.0
transfer_BAHTNET	330.0	368.0	378.0	366.0	371.0	372.0	362.0	381.0	374.0
transfer_Bulk_Payment	90535.0	70775.0	77172.0	60851.0	49420.0	63124.0	50621.0	61133.0	71009.0
transfer_Direct_credit	71385.0	52346.0	58320.0	41904.0	32443.0	43357.0	32381.0	42981.0	40422.0
transfer_Direct_debit	14252.0	13633.0	13569.0	14089.0	12282.0	13045.0	12951.0	12636.0	12618.0
transfer_ITMX Bulk Payment	4898.0	4796.0	5283.0	4858.0	4695.0	6722.0	5289.0	5516.0	17969.0
Online_Retail_Funds_Transfer	585345.0	600291.0	583312.0	552607.0	493752.0	451271.0	462256.0	442908.0	421085.0
Transfer_sub_cash_ATM	3020.0	3394.0	3669.0	3460.0	3498.0	3483.0	3558.0	3611.0	3558.0
Transfer_sub_cash_Internet_and_mobile	582101.0	596653.0	579369.0	548882.0	489980.0	447500.0	458414.0	439013.0	417239.0
Transfer_sub_cash_across_bank_by_branch	224.0	244.0	274.0	265.0	274.0	288.0	284.0	284.0	288.0

ทำการจัดเรียงวันที่ให้ถูกต้อง โดยนำวันที่ 2017-01 ขึ้นมาก่อน

```
new_df13_val = new_df13.loc[:, '2017-01':'2021-02':-1]
new_df13_val
```

แสดงข้อมูลหลังจากจัดเรียงวันที่เสร็จแล้ว

	2017-01	2017-02	2017-03	2017-04	2017-05	2017-06	2017-07	2017-08	2017-09	2017-10	2017-11	2017-12
List												
Check	9205.0	8926.0	10733.0	8564.0	10795.0	9788.0	9166.0	9959.0	9238.0	9162.0	9261.0	8987.0
Cheak_Inhouse	3791.0	3760.0	4601.0	3752.0	4564.0	4144.0	3844.0	4214.0	4052.0	3637.0	3807.0	3835.0
Cheak_Interbank	5414.0	5166.0	6132.0	4812.0	6231.0	5644.0	5322.0	5745.0	5186.0	5525.0	5454.0	5152.0
e-Payments	307239.0	293188.0	331603.0	312108.0	337600.0	336164.0	349284.0	365334.0	370954.0	386356.0	395886.0	414151.0
transfer_BAHTNET	323.0	325.0	386.0	310.0	353.0	334.0	304.0	351.0	340.0	325.0	341.0	336.0
transfer_Bulk_Payment	33921.0	33442.0	35435.0	32630.0	35814.0	33610.0	32626.0	34197.0	36019.0	34357.0	35646.0	38532.0
transfer_Direct_credit	20514.0	19833.0	20496.0	19574.0	20380.0	20005.0	19462.0	20285.0	21540.0	20545.0	21565.0	24267.0
transfer_Direct_debit	9530.0	9402.0	10285.0	8800.0	9760.0	9627.0	9552.0	9741.0	9364.0	10138.0	10008.0	9911.0
transfer_ITMX Bulk Payment	3877.0	4207.0	4654.0	4256.0	5674.0	3978.0	3612.0	4171.0	5115.0	3674.0	4073.0	4354.0
Online_Retail_Funds_Transfer	27900.0	27843.0	31942.0	28962.0	34747.0	35571.0	38494.0	42460.0	44229.0	47753.0	49384.0	53842.0
Transfer_sub_cash_ATM	11080.0	10571.0	11520.0	9084.0	11180.0	10285.0	10308.0	10490.0	9746.0	10147.0	9510.0	9076.0
Transfer_sub_cash_Internet_and_mobile	16044.0	16519.0	19592.0	19243.0	22782.0	24536.0	27478.0	31203.0	33787.0	36909.0	39162.0	44092.0
Transfer_sub_cash_across_bank_by_branch	776.0	753.0	830.0	635.0	785.0	750.0	708.0	767.0	696.0	697.0	712.0	674.0

กำหนดให้ค่า new_df13_ind และ new_df13_col เป็น numpy array ของ index และ คอลัมน์ ของ new_df13_val ตามลำดับ

```
new_df13_ind = np.array(new_df13_val.index)
new_df13_col = np.array(new_df13_val.columns)
```

สร้าง list new_df13_data ใช้ในการเก็บข้อมูลของกิจกรรมทางการเงิน เพื่อสร้าง Dataframe ใหม่ โดยนำข้อมูลในแต่ละ index เก็บไว้ใน new_df13_data แล้วทำการ reshape ให้เป็นอาร์ย 2 มิติ ที่มีขนาดเหมือนกับ new_df13_ind และ new_df13_col

```
new_df13_data = []
for name in new_df13_ind:
    new_df13_data.append(new_df13_val[new_df13_val.index == name])
new_df13_data = np.array(new_df13_data).reshape(len(new_df13_ind), len(new_df13_col))
new_df13_data
```

array([[9205., 8926., 10733., ..., 6756., 6161., 5758.],
[3791., 3760., 4601., ..., 2873., 2631., 2459.],
[5414., 5166., 6132., ..., 3883., 3530., 3299.],
[...,
[40360., 36234., 39840., ..., 60749., 52338., 51025.],
[103115., 94642., 105945., ..., 206086., 189905., 191493.],
[316444., 302114., 342336., ..., 1423102., 1407726., 1402395.]])

สร้าง Dataframe ใหม่ ชื่อว่า new_df13_v2

```
new_df13_v2 = pd.DataFrame({'Date':new_df13_col})
for i in range(len(new_df13_data)):
    new_df13_v2[new_df13_ind[i]] = new_df13_data[i]
new_df13_v2.head()
```

แสดงข้อมูลของ new_df13_v2

	Date	Check	Cheak_Inhouse	Cheak_Interbank	e-Payments	transfer_BAHTNET	transfer_Bulk_Payment	transfer_Direct_credit	transfer_Direct_debit
0	2017-01	9205.0	3791.0	5414.0	307239.0	323.0	33921.0	20514.0	9530.0
1	2017-02	8926.0	3760.0	5166.0	293188.0	325.0	33442.0	19833.0	9402.0
2	2017-03	10733.0	4601.0	6132.0	331603.0	386.0	35435.0	20496.0	10285.0
3	2017-04	8564.0	3752.0	4812.0	312108.0	310.0	32630.0	19574.0	8800.0
4	2017-05	10795.0	4564.0	6231.0	337600.0	353.0	35814.0	20380.0	9760.0

สร้างตัวแปร plot_data3 เพื่อ copy ข้อมูลของ new_df13_v2

```
plot_data3 = new_df13_v2.copy()
```

ส่งคอลัมน์ Date ทั้งหมดเข้าพังก์ชัน pd.to_datetime() จะได้อาร์เรย์ของวันที่ทุกๆ เดือน

```
new_df13_v2.set_index('Date', inplace=True)
```

เปลี่ยนคอลัมน์ Date เป็น index ของ Dataframe

```
new_df13_v2['Date'] = pd.to_datetime(new_df13_v2['Date'].astype('string'))
```

ແສດງຂໍ້ມູນ new_df13_data_v2

```
new_df13_v2.head()
```

Date	Check	Cheak_Inhouse	Cheak_Interbank	e-Payments	transfer_BAHTNET	transfer_Bulk_Payment	transfer_Direct_credit
2017-01-01	9205.0	3791.0	5414.0	307239.0	323.0	33921.0	20514.0
2017-02-01	8926.0	3760.0	5166.0	293188.0	325.0	33442.0	19833.0
2017-03-01	10733.0	4601.0	6132.0	331603.0	386.0	35435.0	20496.0
2017-04-01	8564.0	3752.0	4812.0	312108.0	310.0	32630.0	19574.0
2017-05-01	10795.0	4564.0	6231.0	337600.0	353.0	35814.0	20380.0

ກາຮັບອັນແຮກ ນໍາຂໍ້ມູນຕັ້ງແຕ່ຄອລັນນີ້ e-payments ຕີ້ງ transfer_BAHTNET ແລະ ຂໍ້ມູນຂອງ

Online_Retail_Funds_Transfer, Intra_bank_transfer ແລະ Payment_cards

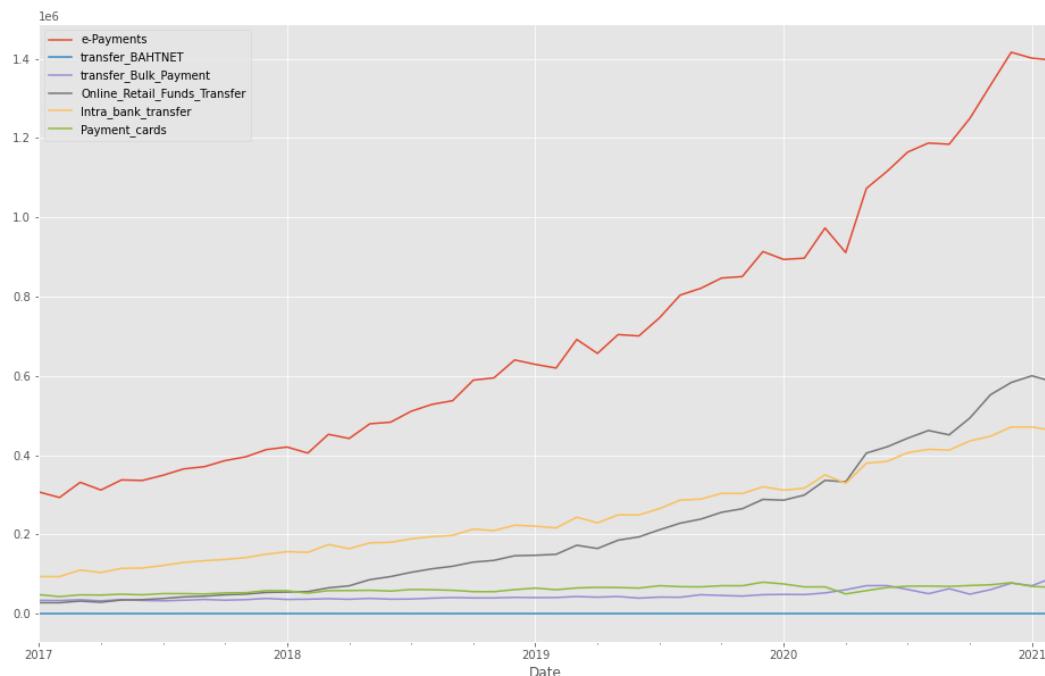
```
plot13_data1 = new_df13_v2.loc[:, 'e-Payments':'transfer_Bulk_Payment']
plot13_data1 = plot13_data1.assign(Online_Retail_Funds_Transfer = new_df13_v2['Online_Retail_Funds_Transfer'])
plot13_data1 = plot13_data1.assign(Intra_bank_transfer = new_df13_v2['Intra_bank_transfer'])
plot13_data1 = plot13_data1.assign(Payment_cards = new_df13_v2['Payment_cards'])
plot13_data1.head()
```

Date	e-Payments	transfer_BAHTNET	transfer_Bulk_Payment	Online_Retail_Funds_Transfer	Intra_bank_transfer
2017-01-01	307239.0	323.0	33921.0	27900.0	93810.0
2017-02-01	293188.0	325.0	33442.0	27843.0	93646.0
2017-03-01	331603.0	386.0	35435.0	31942.0	110169.0
2017-04-01	312108.0	310.0	32630.0	28962.0	103975.0
2017-05-01	337600.0	353.0	35814.0	34747.0	114553.0

ຈາກນັ້ນນໍາຂໍ້ມູນຂອງ plot_13_data1 ມາພື້ນຕາເປັນກາຮັບອັນແຮກ ໂດຍກຳທັນດຸනາດຄວາມກວ້າງແລະຍາວຂອງ ກາຮັບເປັນ 16*10

```
plt.rcParams['figure.figsize'] = 16, 10
style.use("ggplot")
plot13_data1.plot()
plt.show()
```

จากราฟนี้จะเห็นได้ว่า ในปี 2020 มูลค่าการชำระเงินผ่านการโอนเงินรายบุคคลข้ามธนาคารและการโอนเงินภายในธนาคารเพิ่มสูงขึ้น ส่วนมูลค่าการชำระเงินผ่านระบบอื่นๆ ก็ยังคงที่ พอมากๆ พร้อมแล้ว มูลค่าการชำระเงินผ่าน e-Payments เพิ่มสูงขึ้นทุกๆ ปี



กราฟอันแรก นำข้อมูลตั้งแต่คอลัมน์ e-payments ถึง transfer_Bulk_Payment และข้อมูลของ Online_Retail_Funds_Transfer, Intra_bank_transfer และ Payment_cards

```
#แสดงรายละเอียดปี 2020-2021
plot_data13_1 = new_df13_v2.loc['2020-01-01':'2021-02-01','e-Payments':'transfer_Bulk_Payment']
plot_data13_1 = plot_data13_1.assign(Online_Retail_Funds_Transfer = new_df13_v2['Online_Retail_Funds_Transfer'])
plot_data13_1 = plot_data13_1.assign(Intra_bank_transfer = new_df13_v2['Intra_bank_transfer'])
plot_data13_1 = plot_data13_1.assign(Payment_cards = new_df13_v2['Payment_cards'])
plot_data13_1
```

Date	e-Payments	transfer_BAHTNET	transfer_Bulk_Payment	Online_Retail_Funds_Transfer	Intra_bank_transfer	Payment_cards
2020-01-01	893748.0	406.0	48810.0	286531.0	311688.0	75142.0
2020-02-01	897028.0	375.0	48471.0	299371.0	316736.0	67759.0
2020-03-01	973035.0	429.0	52552.0	336457.0	350305.0	67741.0
2020-04-01	911075.0	360.0	60460.0	332666.0	329147.0	50093.0
2020-05-01	1072989.0	330.0	70827.0	405261.0	379655.0	58260.0
2020-06-01	1116023.0	374.0	71009.0	421085.0	384332.0	66235.0
2020-07-01	1164781.0	381.0	61133.0	442908.0	406435.0	69535.0
2020-08-01	1187397.0	362.0	50621.0	462256.0	414792.0	69919.0
2020-09-01	1184387.0	372.0	63124.0	451271.0	412884.0	69025.0
2020-10-01	1249594.0	371.0	49420.0	493752.0	435949.0	71256.0
2020-11-01	1333362.0	366.0	60851.0	552607.0	447872.0	73028.0
2020-12-01	1416346.0	378.0	77172.0	583312.0	471204.0	78194.0
2021-01-01	1401565.0	368.0	70775.0	600291.0	471239.0	68987.0
2021-02-01	1396637.0	330.0	90535.0	585345.0	462343.0	66591.0

กราฟันที่สอง นำข้อมูลตั้งแต่คอลัมน์ Transfer_sub_cash_ATM ถึง Transfer_sub_cash_across_bank_by_branch จากนั้นเพิ่มคอลัมน์ Date เพื่อใช้ในการระบุวันที่ของข้อมูล

```
new_plot_data3 = plot_data3.loc[:, 'Transfer_sub_cash_ATM':'Transfer_sub_cash_across_bank_by_branch']
new_plot_data3 = new_plot_data3.assign(date = plot_data3['Date'])
new_plot_data3.head()
```

	Transfer_sub_cash_ATM	Transfer_sub_cash_Internet_and_mobile	Transfer_sub_cash_across_bank_by_branch	date
0	11080.0	16044.0	776.0	2017-01
1	10571.0	16519.0	753.0	2017-02
2	11520.0	19592.0	830.0	2017-03
3	9084.0	19243.0	635.0	2017-04
4	11180.0	22782.0	785.0	2017-05

ทำการแบ่งข้อมูลออกเป็นสองคอลัมน์ แล้วเอาคอลัมน์ Date และ Month ออก

```
new_plot_data3[['Year', 'Month']] = new_plot_data3['date'].str.split('-', 1, expand=True)
new_plot_data3.drop(['date', 'Month'], axis='columns', inplace=True)
```

แสดงข้อมูลหลังจากแยกข้อมูลเสร็จแล้ว

```
new_plot_data3.head()
```

	Transfer_sub_cash_ATM	Transfer_sub_cash_Internet_and_mobile	Transfer_sub_cash_across_bank_by_branch	Year
0	11080.0	16044.0	776.0	2017
1	10571.0	16519.0	753.0	2017
2	11520.0	19592.0	830.0	2017
3	9084.0	19243.0	635.0	2017
4	11180.0	22782.0	785.0	2017

ทำการจัดกลุ่มโดยใช้พังก์ชัน groupby() เพื่อทำให้ข้อมูลอยู่ในรูปลิสต์ที่จำแนกตามปี เก็บในตัวแปรใหม่แยกตามแต่ละข้อมูล ซึ่งใช้ร่วมกับพังก์ชัน sum() จากนั้นนำข้อมูลตั้งแต่ index ในปี 2020 ถึง 2021

```
new_plot_data3 = new_plot_data3.groupby('Year').sum()
new_plot_data3 = new_plot_data3.loc['2020':'2021', :]
new_plot_data3
```

Year	Transfer_sub_cash_ATM	Transfer_sub_cash_Internet_and_mobile	Transfer_sub_cash_across_bank_by_branch
2020	43915.0	5019972.0	3590.0
2021	6414.0	1178754.0	468.0

กำหนดให้ plot_data3_x เป็น numpy array ของ index ในช่วงปี 2020 ถึง 2021 plot_data3_y1, plot_data3_y2, plot_data3_y3 เป็น numpy array ข้อมูลของ Transfer_sub_cash_ATM, Transfer_sub_cash_Internet_and_mobile และ Transfer_sub_cash_bank_by_branch ตามลำดับ

```
plot_data3_x = np.array(new_plot_data3.index)
plot_data3_y1 = np.array(new_plot_data3['Transfer_sub_cash_ATM'])
plot_data3_y2 = np.array(new_plot_data3['Transfer_sub_cash_Internet_and_mobile'])
plot_data3_y3 = np.array(new_plot_data3['Transfer_sub_cash_across_bank_by_branch'])
```

กำหนดให้ค่า X_axis เป็น ตำแหน่งของกราฟแท่งตามขนาดของ plot_data3_x และ ความกว้างของกราฟเป็น 0.25 จากนั้นสร้างกราฟสามกราฟ โดยที่กราฟแรกมีระยะห่างเป็น X_axis (ผึ้งช้าย) กราฟที่สองมีระยะห่างเป็น X_axis + width (ตรงกลาง) และกราฟที่สองมีระยะห่างเป็น X_axis + width*2 พร้อมกับบอกปริมาณการชำระเงินบันแท่งกราฟ โดยกำหนดขนาดความกว้างและยาวของกราฟเป็น 10^6

```
plt.rcParams['figure.figsize'] = 10, 6
style.use("ggplot")

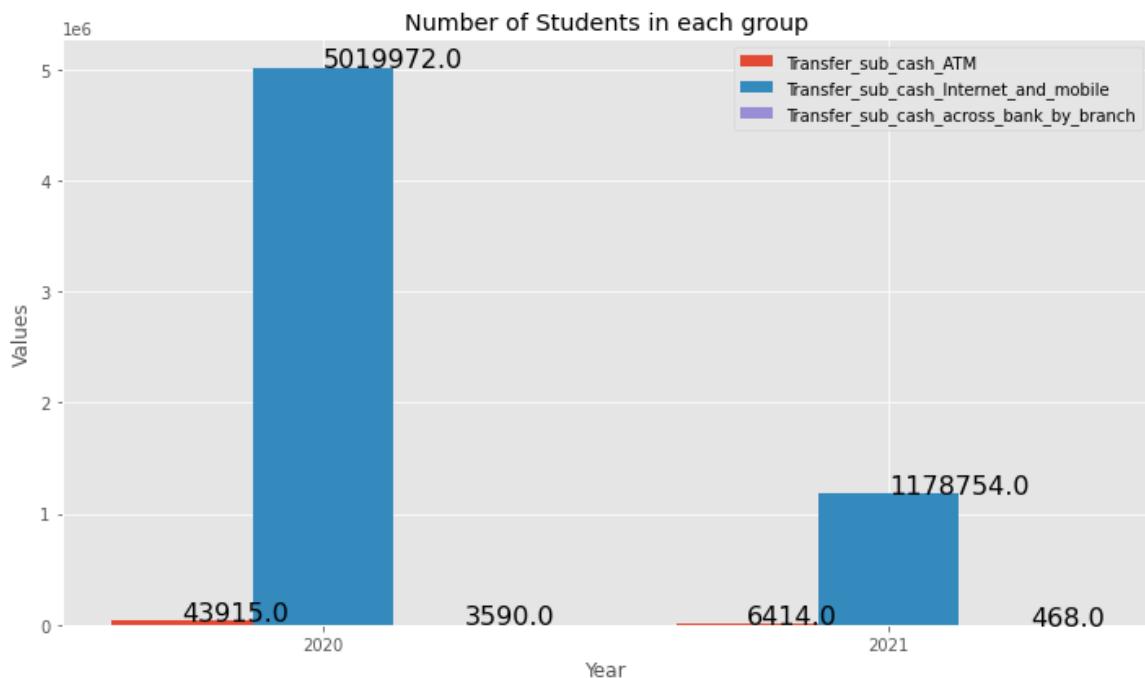
X_axis = np.arange(len(plot_data2_x))
width = 0.25

plt.bar(X_axis, plot_data3_y1, width, label = 'Transfer_sub_cash_ATM')
plt.bar(X_axis+width, plot_data3_y2, width, label = 'Transfer_sub_cash_Internet_and_mobile')
plt.bar(X_axis+width*2, plot_data3_y3, width, label = 'Transfer_sub_cash_across_bank_by_branch')

for index,data in enumerate(plot_data3_y1):
    plt.text(x=index , y =data+1 , s=f"{data}" , fontdict=dict(fontsize=16))
for index,data in enumerate(plot_data3_y2):
    plt.text(x=index+width , y =data+1 , s=f"{data}" , fontdict=dict(fontsize=16))
for index,data in enumerate(plot_data3_y3):
    plt.text(x=index+width*2 , y =data+1 , s=f"{data}" , fontdict=dict(fontsize=16))

plt.xticks(X_axis+width, plot_data3_x)
plt.xlabel("Year")
plt.ylabel("Values")
plt.title("Number of Students in each group")
plt.tight_layout()
plt.legend()
plt.show()
```

จากการแท่งนี้พบว่า มูลค่าการโอนเงินรายย่อยข้ามธนาคารผ่านอินเตอร์เน็ตและโทรศัพท์เคลื่อนที่ เพิ่มสูงขึ้นในปี 2020 แต่มูลค่าการโอนเงินรายย่อยข้ามธนาคารผ่านเครื่องเอทีเอ็มและผ่านสาขาห้องลงมาก ๆ



ทำการแสดงข้อมูลใน dataList 5 ออกมา เพื่อทำการตรวจสอบความถูกต้องของข้อมูลก่อนที่จะทำการนำไปplot ภาพในขั้นตอนถัดไป

▶ dataList[5]

	Year	Youtube	Facebook	LINE	Instagram	Pantip	Twitter	WhatsApp
0	2017	97.1	96.6	95.8	56.0	54.7	27.6	12.1
1	2018	98.8	96.0	98.6	67.2	64.2	43.0	10.6
2	2020	97.5	98.2	96.0	80.4	30.6	71.9	6.7

เตรียมข้อมูลที่ต้องการพล็อต โดยใช้ฟังก์ชัน groupby() เพื่อทำให้ข้อมูลอยู่ในรูปผลลัพธ์ที่จำแนกตามปี เก็บในตัวแปรใหม่แยกตามแต่ละแพลตฟอร์ม ซึ่งใช้รวมกับฟังก์ชัน sum()

```
▶ you = dataList[5].groupby('Year')[['Youtube']].sum()
face = dataList[5].groupby('Year')[['Facebook']].sum()
lin = dataList[5].groupby('Year')[['LINE']].sum()
inst = dataList[5].groupby('Year')[['Instagram']].sum()
pt = dataList[5].groupby('Year')[['Pantip']].sum()
tw = dataList[5].groupby('Year')[['Twitter']].sum()
wa = dataList[5].groupby('Year')[['WhatsApp']].sum()
```

ทำการกำหนดขนาดความกว้างและยาวของกราฟเป็น 12×6 , เรียกใช้ฟังก์ชัน `style.use("ggplot")` เพื่อเพิ่มเส้นตารางเป็นพื้นหลังกราฟ และทำการกำหนดค่าในแกน X ให้เป็นเลขจำนวนเต็ม โดยใช้ `xaxis.set_major_locator(MaxNLocator(integer = True))`

```
plt.rcParams['figure.figsize'] = 12, 6
style.use("ggplot")

fig = plt.figure()
ax = fig.gca()
ax.xaxis.set_major_locator(MaxNLocator(integer=True))
```

ผลลัพธ์จากการตัวแปรที่ได้ทำจัดกลุ่มออกมา โดยมีการกำหนดสีให้กราฟแต่ละเส้นซึ่งแทนด้วย แต่ละแพลตฟอร์ม เพื่อให้สามารถเปรียบเทียบความแตกต่างได้อย่างชัดเจนมากยิ่งขึ้น และสร้าง label เพื่อเพิ่มส่วนของคำอธิบายรายละเอียดกราฟ

```
plt.plot(you, color = 'black',label="Youtube")
plt.plot(face, color = 'pink',label="Facebook")
plt.plot(lin, color = 'orchid',label="LINE")
plt.plot(inst, color = 'red',label='Instagram')
plt.plot(pt, color = 'green',label="Pantip")
plt.plot(tw, color = 'yellow',label="Twitter")
plt.plot(wa, color = 'blue',label="WhatsApp")
```

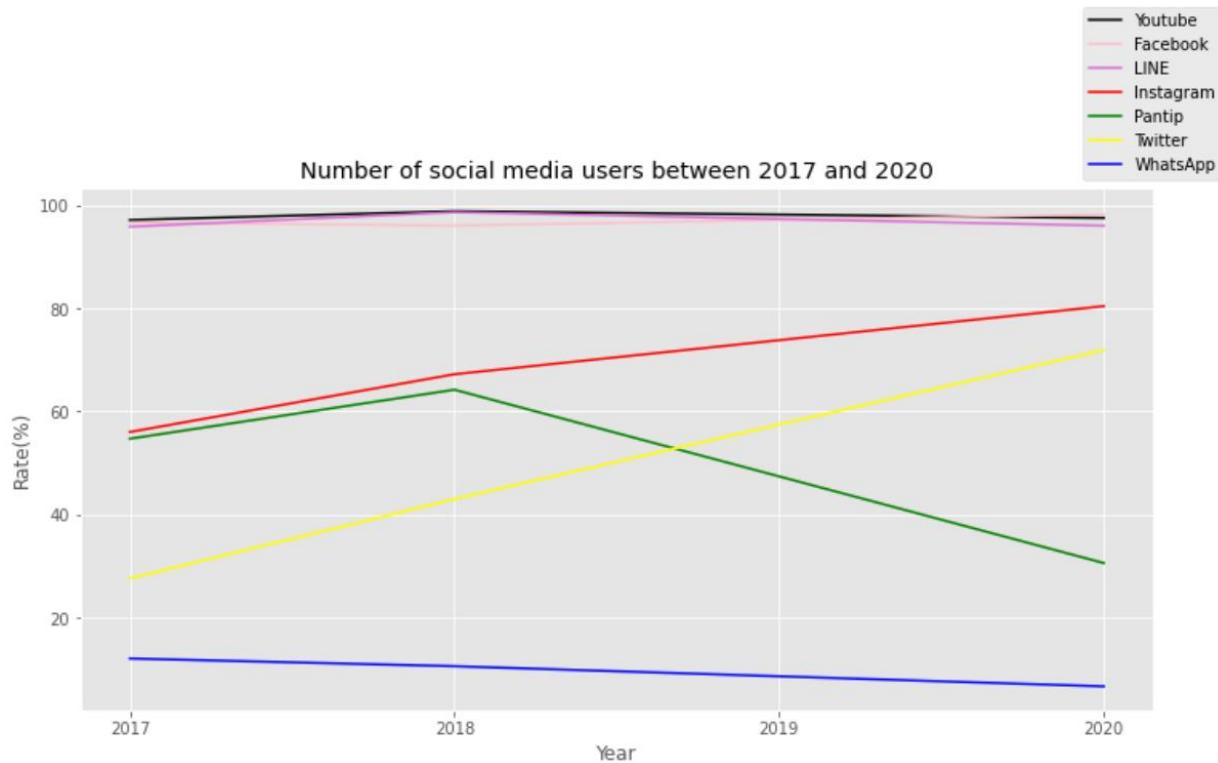
ตั้งชื่อกราฟว่า Number of social media users between 2017 and 2020 โดยใช้ฟังก์ชัน `title()` และสร้างlabelของแกน X,Y โดยใช้ฟังก์ชัน `xlabel()` และ `ylabel()` ซึ่งแทนด้วยYearและRate(%) ตามลำดับ

```
plt.title("Number of social media users between 2017 and 2020")
plt.xlabel("Year")
plt.ylabel("Rate(%)")
```

ใช้ฟังก์ชัน `legend()` และ `show()` เพื่อแสดงผลส่วนคำอธิบายกราฟและตัวกราฟออกมา โดยมี การเพิ่มพารามิเตอร์เพื่อจัดตำแหน่งของคำอธิบายกราฟให้อยู่ในตำแหน่งที่เหมาะสม

```
plt.legend(loc="lower center", bbox_to_anchor=(1.0, 1.0))
plt.show()
```

โดยจากการพิจารณาของจำนวนผู้ใช้งานแพลตฟอร์มโซเชียลมีเดียตั้งแต่ปี 2017 ถึง 2020 ที่ได้ทำการแสดงออกความสามารถเห็นจากภาพรวมได้ว่ามีการใช้งานโซเชียลมีเดียแต่ละแพลตฟอร์มในปี 2020 เพิ่มสูงขึ้นจากปี 2019 โดยแพลตฟอร์มที่มีการใช้งานเพิ่มขึ้นได้แก่ Facebook, Instagram และ Twitter



ทำการแสดงข้อมูลใน `DataList 7` ออกแบบ เพื่อทำการตรวจสอบความถูกต้องของข้อมูลก่อนที่จะทำการนำไปพล็อตกราฟในขั้นตอนถัดไป

```
[ ] dataList[7]
```

	Year	User_Number
0	2017	51.0
1	2018	51.0
2	2019	52.0
3	2020	55.0

กำหนดขนาดความกว้างและยาวของกราฟเป็น $10*10$, เรียกใช้ฟังก์ชัน `subplot()` และทำการเตรียมข้อมูลที่ต้องการพล็อต โดยใช้ฟังก์ชัน `array()` เพื่อทำให้ข้อมูลอยู่ในรูปอาร์เรย์ที่เก็บข้อมูลปี ทำการแปลงให้อยู่ในรูปแบบ `string` และเก็บในตัวแปร `x_df7` ซึ่งแทนด้วยข้อมูลในแกน X และ อาร์เรย์ที่เก็บข้อมูลจำนวนผู้ใช้ เก็บในตัวแปร `y_df7` ซึ่งแทนด้วยข้อมูลในแกน Y

```
[ ] #version start 0
plt.rcParams['figure.figsize'] = 10, 10
f, ax = plt.subplots(1)
x_df7 = np.array(dataList[7]['Year'].astype("string"))
y_df7 = np.array(dataList[7]['User_Number'])
```

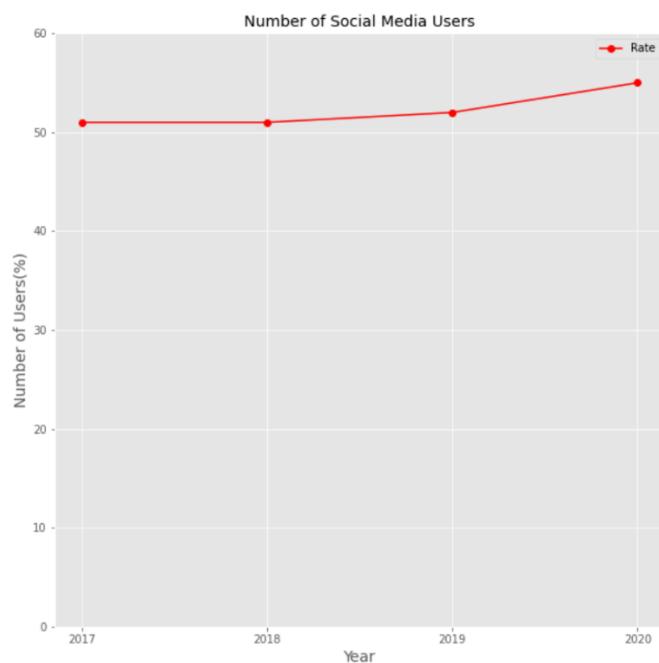
ผลลัพธ์ค่าที่มีการเตรียมไว้ลงบนแกน X และ Y โดยมีการเพิ่มพารามิเตอร์ในการกำหนดสีของกราฟเป็นสีแดง และ เพิ่ม label แสดงรายละเอียดของกราฟซึ่งแทนด้วย Rate ทำการตั้งชื่อกราฟว่า Number of Social Media Users โดยใช้ฟังก์ชัน title() และเพิ่มพารามิเตอร์กำหนดขนาดตัวอักษรของชื่อกราฟเท่ากับ 14, สร้าง label ของแกน X,Y โดยใช้ฟังก์ชัน xlabel() และ ylabel() ซึ่งแทนด้วย Year และ Number of Users (%) ตามลำดับ โดยเพิ่มพารามิเตอร์กำหนดขนาดตัวอักษรของ label ทั้ง 2 แกนดังกล่าวให้เท่ากับ 14

```
plt.plot(x_df7, y_df7,'ob-',color='red',label='Rate')
plt.title("Number of Social Media Users",fontsize=14)
plt.xlabel("Year",fontsize=14)
plt.ylabel("Number of Users(%)",fontsize=14)
```

ทำการกำหนดค่าเริ่มต้นแกน Y ให้เริ่มต้นที่ 0 ถึง ค่าสูงสุดของตัวแปร y_df7 ซึ่งแทนด้วยข้อมูลในแกน Y บวกด้วยค่าคงที่ 5 และ ทำการแสดงผลส่วนคำอธิบายรายละเอียดกราฟและตัวกราฟออกมาโดยใช้ฟังก์ชัน legend() และ show()

```
ax.set_ylim(0, max(y_df7)+5)
plt.legend()
plt.show()
```

จากการจำนวนผู้ใช้งานโซเชียลมีเดียตั้งแต่ปี 2017-2020 ที่ได้ทำการแสดงออกมา จะเห็นได้ว่าในปี 2020 มีอัตราการเติบโตของจำนวนผู้ใช้โซเชียลมีเดียสูงมากเมื่อเทียบกับปี 2019 คือเพิ่มขึ้นจาก 52 ล้านคน เป็น 55 ล้านคน คือมีจำนวนผู้ใช้งานเพิ่มขึ้นถึง 3 ล้านคน ในขณะที่ในปีก่อนหน้าตั้งแต่ปี 2017-2019 มีอัตราการเติบโตที่ค่อนข้างคงที่โดยมีจำนวนผู้ใช้งาน 51 ล้านคนจากปี 2017 ถึง 2018 และเพิ่มขึ้นเป็น 52 ล้านคนเป็นในปี 2019



ทำการแสดงข้อมูลใน `DataList 9` ออกมา เพื่อทำการตรวจสอบความถูกต้องของข้อมูลก่อนที่จะทำการนำไปเพล้อตกราฟในขั้นตอนถัดไป

```
[ ] dataList[9]
```

	Year	Platform	Rate	Type
0	2017	Facebook	75.0	Social Network
1	2017	Youtube	72.0	Social Network
2	2017	Line	68.0	Messenger/Chat App/VOIP
3	2017	Messenger	55.0	Messenger/Chat App/VOIP
4	2017	Instagram	50.0	Social Network
5	2017	Google+	45.0	Social Network
6	2017	Twitter	38.0	Social Network
7	2017	Skype	22.0	Messenger/Chat App/VOIP
8	2017	Linkedin	18.0	Social Network
9	2017	Pinterest	17.0	Social Network
10	2017	Whatapp	17.0	Messenger/Chat App/VOIP
11	2017	Wechat	17.0	Messenger/Chat App/VOIP
12	2018	Facebook	93.0	Social Network
13	2018	Youtube	91.0	Social Network
14	2018	Line	84.0	Messenger/Chat App/VOIP
15	2018	Messenger	72.0	Messenger/Chat App/VOIP
16	2018	Instagram	65.0	Social Network
17	2018	Twitter	52.0	Social Network
18	2018	Skype	27.0	Messenger/Chat App/VOIP
19	2018	Linkedin	25.0	Social Network
20	2018	Pinterest	25.0	Social Network
21	2018	Wechat	25.0	Messenger/Chat App/VOIP
22	2018	Whatapp	25.0	Messenger/Chat App/VOIP
23	2018	Twitch	24.0	Social Network
24	2018	Snapchat	20.0	Social Network
25	2018	Sina Weibo	19.0	Social Network
26	2018	Tumber	19.0	Social Network
27	2018	Viadeo	19.0	Social Network
28	2019	Facebook	94.0	Social Network
29	2019	Youtube	94.0	Social Network
30	2019	Line	85.0	Messenger/Chat App/VOIP
31	2019	Messenger	76.0	Messenger/Chat App/VOIP
32	2019	Instagram	65.0	Social Network
33	2019	Twitter	55.0	Social Network
35	2019	Pinterest	32.0	Social Network
36	2019	Linkedin	27.0	Social Network

37	2019	Wechat	27.0	Messenger/Chat App/VOIP
38	2019	Whatapp	27.0	Messenger/Chat App/VOIP
39	2019	Twitch	25.0	Social Network
40	2019	Skype	25.0	Messenger/Chat App/VOIP
41	2019	Snapchat	23.0	Social Network
43	2019	Tumber	19.0	Social Network
44	2020	Youtube	94.2	Social Network
45	2020	Facebook	93.3	Social Network
46	2020	Line	86.2	Messenger/Chat App/VOIP
47	2020	Messenger	77.1	Messenger/Chat App/VOIP
48	2020	Instagram	64.2	Social Network
49	2020	Twitter	57.8	Social Network
51	2020	Pinterest	31.7	Social Network
52	2020	Wechat	24.9	Messenger/Chat App/VOIP
53	2020	Twitch	24.5	Social Network
54	2020	Linkedin	24.4	Social Network
55	2020	Whatapp	24.4	Messenger/Chat App/VOIP
56	2020	Skype	22.6	Messenger/Chat App/VOIP
57	2020	Snapchat	19.8	Social Network
59	2020	Viadeo	18.0	Social Network

เตรียมข้อมูลที่จะนำมาพล็อต โดยทำให้ข้อมูลอยู่ในรูปแบบลิสต์ที่จำแนกตามแพลตฟอร์ม และเก็บค่าไว้ในตัวแปรที่สร้างขึ้นใหม่ เพื่อความสะดวกต่อการเรียกใช้ในภายหลัง

```
[ ] df9_fb = dataList[9][dataList[9]['Platform']=="Facebook"]
df9_yt = dataList[9][dataList[9]['Platform']=="Youtube"]
df9_tw = dataList[9][dataList[9]['Platform']=="Twitter"]
df9_li = dataList[9][dataList[9]['Platform']=="Line"]
df9_msg = dataList[9][dataList[9]['Platform']=="Messenger"]
df9_ig = dataList[9][dataList[9]['Platform']=="Instagram"]

df9_sk = dataList[9][dataList[9]['Platform']=="Skype"]
df9_lin = dataList[9][dataList[9]['Platform']=="Linkedin"]
df9_pi = dataList[9][dataList[9]['Platform']=="Pinterest"]
df9_wh = dataList[9][dataList[9]['Platform']=="Whatapp"]
df9_we = dataList[9][dataList[9]['Platform']=="Wechat"]
```

ทำการเปลี่ยนประเภทของคอลัมน์ Year ให้เป็นประเภท string

```
[ ] dataList[9]['Year'] = dataList[9]['Year'].astype('string')
```

เตรียมข้อมูลที่จะนำมาพล็อต โดยทำให้ข้อมูลอยู่ในรูปแบบลิสต์ที่จำแนกตามปี และเก็บค่าไว้ในตัวแปรที่สร้างขึ้นใหม่ซึ่งแทนข้อมูลแต่ละปี เพื่อความสะดวกต่อการเรียกใช้ในภายหลัง

```
[ ] df9_y2017 = dataList[9][dataList[9]['Year']=='2017']
df9_y2018 = dataList[9][dataList[9]['Year']=='2018']
df9_y2019 = dataList[9][dataList[9]['Year']=='2019']
df9_y2020 = dataList[9][dataList[9]['Year']=='2020']
```

สร้างตัวแปรใหม่ในการเก็บข้อมูลใน dataList[9] คอลัมน์ Year โดยใช้ฟังก์ชัน groupby() และ sum()

```
[ ] df_new = dataList[9].groupby('Year').sum()
```

เตรียมข้อมูลที่จะนำมาพล็อต โดยการทำให้ข้อมูลอยู่ในรูปอาร์เรย์ที่เก็บข้อมูล Rate และเก็บค่าไว้ในตัวแปรที่สร้างขึ้นใหม่ซึ่งแทนด้วยแต่ละแพลตฟอร์ม

```
[ ] df9_x = np.array(df_new.index)
df9_fb = np.array(df9_fb['Rate'])
df9_yt = np.array(df9_yt['Rate'])
df9_tw = np.array(df9_tw['Rate'])
df9_li = np.array(df9_li['Rate'])
df9_msg = np.array(df9_msg['Rate'])
df9_ig = np.array(df9_ig['Rate'])

df9_sk = np.array(df9_sk['Rate'])
df9_lin = np.array(df9_lin['Rate'])
df9_pi = np.array(df9_pi['Rate'])
df9_wh = np.array(df9_wh['Rate'])
df9_we = np.array(df9_we['Rate'])
```

กำหนดขนาดความกว้างและยาวของกราฟเป็น 12*10 และเรียกใช้ฟังก์ชัน style.use ("ggplot") เพื่อเพิ่มเส้นตารางเป็นพื้นหลังกราฟ

```
[ ] plt.rcParams['figure.figsize'] = 12, 10
style.use("ggplot")
```

ใช้ฟังก์ชัน subplot() และ พล็อตกราฟจากตัวแปรที่ได้ทำการเตรียมไว้ โดยมีการเพิ่มพารามิเตอร์ในการกำหนดสีให้กับกราฟแต่ละเส้นซึ่งแทนแพลตฟอร์มแต่ละประเภท เพื่อให้การแสดงผลกราฟสามารถเปรียบเทียบความแตกต่างได้อย่างชัดเจนมากขึ้น รวมทั้งเพิ่มพารามิเตอร์ label เพื่อเพิ่มส่วนอธิบายรายละเอียดของกราฟ

```
f, ax = plt.subplots(1)
plt.plot(df9_x, df9_fb, label="Facebook", color='black')
plt.plot(df9_x, df9_yt, label="Youtube", color='red')
plt.plot(df9_x, df9_tw, label="Twitter", color='pink')
plt.plot(df9_x, df9_li, label="Line", color='salmon')
plt.plot(df9_x, df9_msg, label="Messenger", color='orange')
plt.plot(df9_x, df9_ig, label="Instagram", color='yellow')
plt.plot(df9_x, df9_sk, label="Skype", color='green')
plt.plot(df9_x, df9_lin, label="Linkedin", color='skyblue')
plt.plot(df9_x, df9_pi, label="Pinterest", color='blue')
plt.plot(df9_x, df9_wh, label="Whatapp", color='gray')
plt.plot(df9_x, df9_we, label="Wechat", color='purple')
```

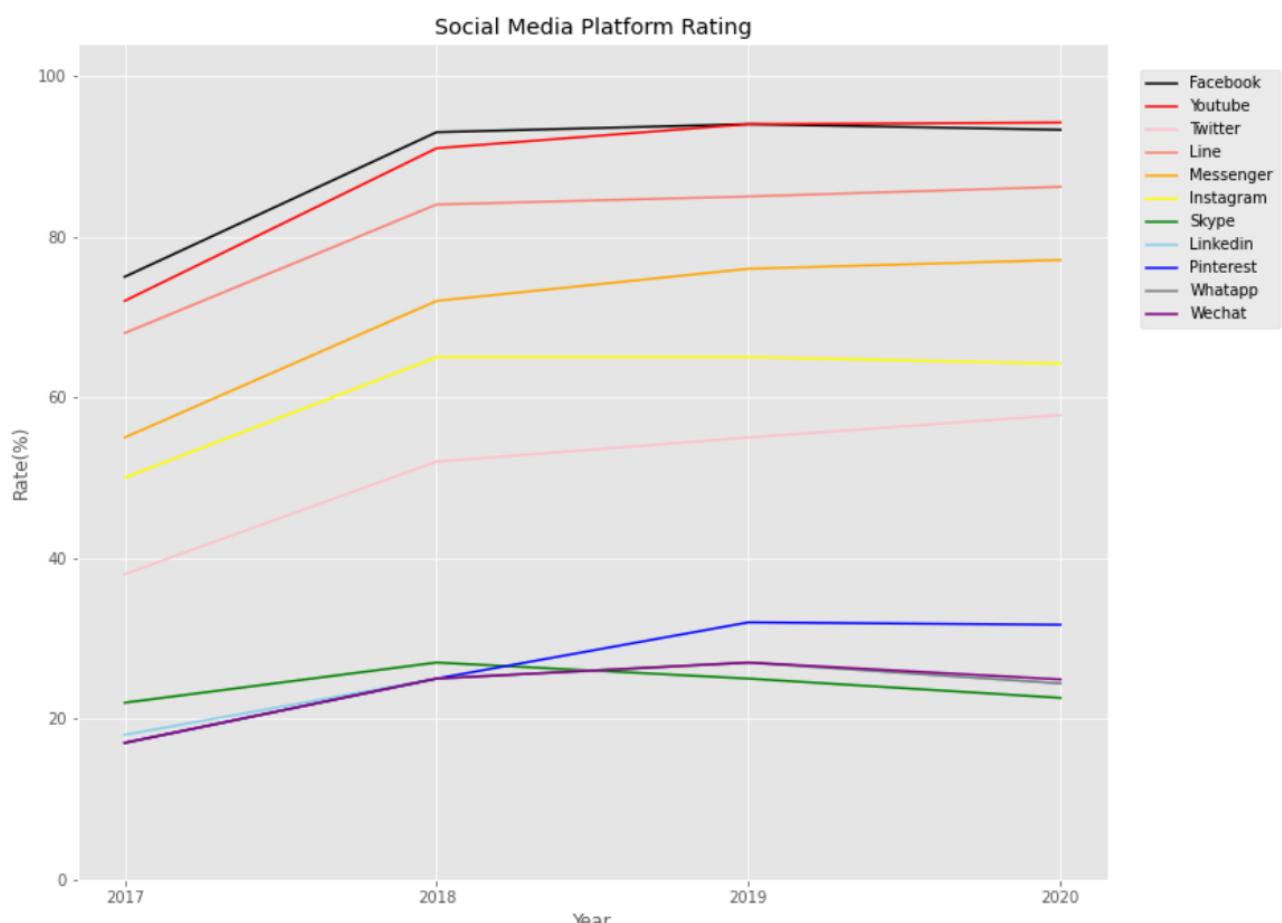
ทำการตั้งชื่อหัวข้อของกราฟว่า Social Media Platform Rating โดยใช้ฟังก์ชัน title() และสร้าง label ของแกน X,Y โดยใช้ฟังก์ชัน xlabel() และ ylabel() ซึ่งแทนด้วย Year และ Rate(%) ตามลำดับ

```
plt.title('Social Media Platform Rating')
plt.xlabel('Year')
plt.ylabel('Rate(%)')
```

ทำการกำหนดค่าเริ่มต้นแกน Y ให้เริ่มต้นที่ 0 ถึง ค่าสูงสุดของตัวแปร df9_fb บวกด้วยค่าคงที่ 10 และ ทำการแสดงผลส่วนคำอธิบายรายละเอียดกราฟโดยใช้ฟังก์ชัน legend() โดยมีการเพิ่มพารามิเตอร์เพื่อจัดตำแหน่งของคำอธิบายกราฟให้อยู่ในตำแหน่งที่เหมาะสม และแสดงผลด้วยกราฟออกมากโดยใช้ฟังก์ชัน show()

```
ax.set_ylim(0, max(df9_fb)+10)
plt.legend(loc="lower center", bbox_to_anchor=(1.1, 0.65))
plt.show()
```

จากราฟ ร้อยละของการใช้งานโซเชียลมีเดียตั้งแต่ปี 2017-2020 ที่ได้ทำการแสดงออกมากแสดงเห็นภาพรวมว่ามีการใช้งานโซเชียลมีเดียแต่ละแพลตฟอร์มในปี 2020 เพิ่มมากขึ้นจากปี 2019 โดยแพลตฟอร์มที่มีการใช้งานเพิ่มขึ้นได้แก่ Facebook, Instagram, Messenger และ Twitter



เตรียมข้อมูลที่จะนำมาพล็อต โดยการทำข้อมูลให้อยู่ในรูปแบบลิสต์ที่เก็บข้อมูลร้อยละและชื่อของแต่ละแพลตฟอร์ม เก็บค่าไว้ในตัวแปรที่สร้างขึ้นใหม่ เพื่อความสะดวกต่อการเรียกใช้ในภายหลัง

```
[ ] df9_y2017_Platform = df9_y2017['Platform']
df9_y2017_Rate = df9_y2017['Rate']
df9_y2018_Platform = df9_y2018['Platform']
df9_y2018_Rate = df9_y2018['Rate']
df9_y2019_Platform = df9_y2019['Platform']
df9_y2019_Rate = df9_y2019['Rate']
df9_y2020_Platform = df9_y2020['Platform']
df9_y2020_Rate = df9_y2020['Rate']
```

กำหนดขนาดความกว้างและยาวของกราฟเป็น 18*15 และเรียกใช้ฟังก์ชัน style.use ("ggplot") เพื่อเพิ่มเส้นตารางเป็นพื้นหลังกราฟ

```
[ ] plt.rcParams['figure.figsize'] = 18, 15
style.use("ggplot")
```

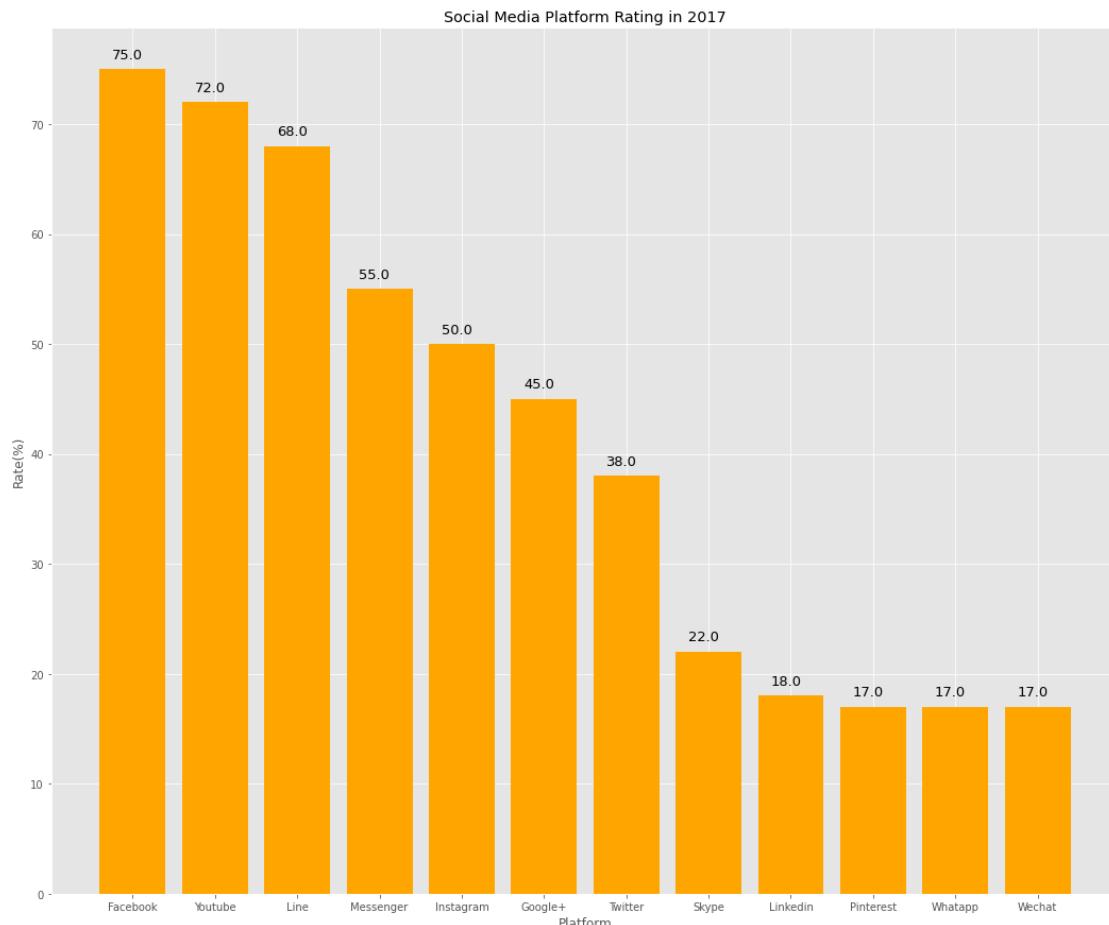
ใช้ฟังก์ชัน bar() เพื่อสร้างแผนภูมิแท่ง และทำการพล็อตค่าที่มีการเตรียมไว้ โดยมีการกำหนดพารามิเตอร์สีของแท่งแผนภูมิให้เป็นสีส้ม และใช้ฟังก์ชัน plt.text() ในการใส่ค่าของ Rate ในกราฟแต่ละแท่ง

```
plt.bar(df9_y2017_Platform, df9_y2017_Rate, color='orange')
for index,data in enumerate(df9_y2017_Rate):
    plt.text(x=index-0.25 , y =data+1 , s=f'{data}' , fontdict=dict(fontsize=13))
```

ตั้งชื่อหัวข้อของกราฟว่า Social Media Platform Rating in 2017 โดยใช้ฟังก์ชัน title() และสร้าง label ของแกน X,Y โดยใช้ฟังก์ชัน xlabel() และ ylabel() ซึ่งแทนด้วย Platform และ Rate (%) ตามลำดับ และใช้ฟังก์ชัน show() ในการแสดงผล

```
plt.title('Social Media Platform Rating in 2017')
plt.xlabel('Platform')
plt.ylabel('Rate(%)')
plt.show()
```

โดยจากราฟที่แสดงรายละเอียด เปอร์เซ็นต์ของการใช้งานแพลตฟอร์มโซเชียลมีเดียในปี 2017 ด้านล่าง แสดงให้เห็นว่ามีการใช้งานแพลตฟอร์ม Facebook สูงที่สุดคิดเป็น 75% ตามมาด้วย อันดับที่ 2 คือ YouTube คิดเป็น 72% ,อันดับที่ 3 คือ Line คิดเป็น 68%, อันดับที่ 4 Messenger 55%, อันดับที่ 5 Instagram 50%, อันดับที่ 6 Google+ 45%, อันดับที่ 7 Twitter 38%, อันดับที่ 8 Skype 22%, อันดับที่ 9 LinkedIn 18%, และ อันดับที่ 10 Pinterest, WhatsApp, WeChat 17%



กำหนดขนาดความกว้างและยาวของกราฟเป็น 18*15 และเรียกใช้พังก์ชัน `style.use("ggplot")` เพื่อเพิ่มเส้นตารางเป็นพื้นหลังกราฟ

```
[ ] plt.rcParams['figure.figsize'] = 18, 15
style.use("ggplot")
```

ใช้พังก์ชัน `bar()` เพื่อสร้างแผนภูมิแท่ง และทำการพลอตค่าที่มีการเตรียมไว้ โดยมีการกำหนดพารามิเตอร์สีของแท่งแผนภูมิให้เป็นสีชมพู และใช้พังก์ชัน `plt.text()` ในการใส่ค่าของRateในกราฟแต่ละแท่ง

```
plt.bar(df9_y2018_Platform, df9_y2018_Rate, color='pink')
for index,data in enumerate(df9_y2018_Rate):
    plt.text(x=index-0.25 , y =data+1 , s=f'{data}' , fontdict=dict(fontsize=13))
```

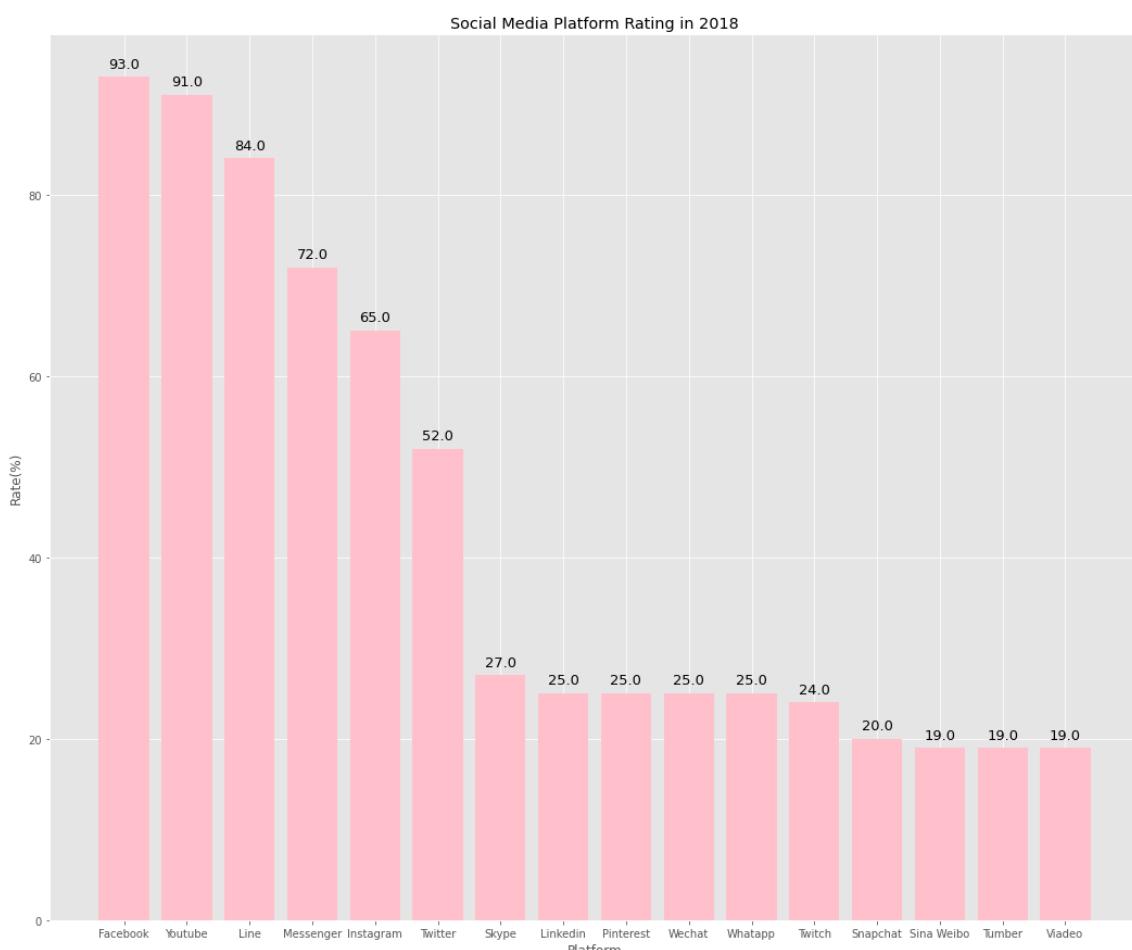
ตั้งชื่อหัวข้อของกราฟว่า Social Media Platform Rating in 2018 โดยใช้พังก์ชัน `title()` และสร้าง label ของแกน X,Y โดยใช้พังก์ชัน `xlabel()` และ `ylabel()` ซึ่งแทนด้วย Platform และ Rate (%) ตามลำดับ และใช้พังก์ชัน `show()` ในการแสดงผล

```

plt.title('Social Media Platform Rating in 2018')
plt.xlabel('Platform')
plt.ylabel('Rate(%)')
plt.show()

```

โดยจากราฟที่แสดงรายละเอียดเบอร์เซ็นต์การใช้งานแพลตฟอร์มโซเชียลมีเดียของปี 2018 ด้านล่าง แสดงให้เห็นการใช้งาน Facebook สูงที่สุดคิดเป็น 93% ตามมาด้วยอันดับที่ 2 คือ YouTube คิดเป็น 91%, อันดับที่ 3 คือ Line คิดเป็น 84%, อันดับที่ 4 Messenger 72%, อันดับที่ 5 Instagram 65%, อันดับที่ 6 Twitter 52%, อันดับที่ 7 Skype 27%, อันดับที่ 8 คือ LinkedIn, Pinterest, WhatsApp, และ WeChat 25%, อันดับที่ 9 คือ Twitch 24%, อันดับที่ 10 Snapchat 20%, และอันดับที่ 11 Sina Weibo, Tumblr, Viadeo 19%



กำหนดขนาดความกว้างและยาวของกราฟเป็น 15*15 และเรียกใช้พังก์ชัน style.use ("ggplot") เพื่อเพิ่มเส้นตารางเป็นพื้นหลังกราฟ

```

[ ] plt.rcParams['figure.figsize'] = 15, 15
      style.use("ggplot")

```

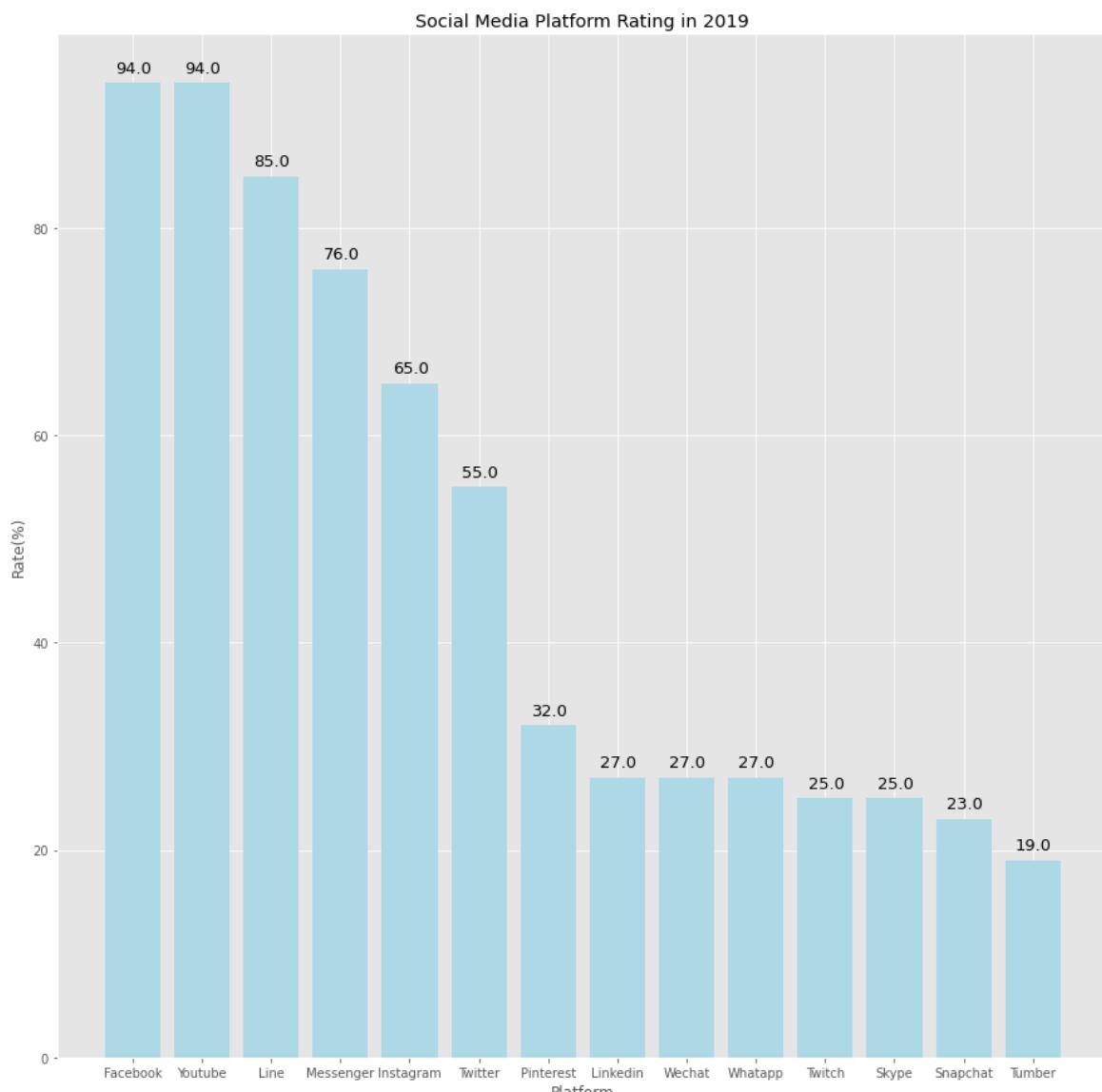
ใช้ฟังก์ชัน bar() เพื่อสร้างแผนภูมิแท่ง และทำการพล้อตค่าที่เตรียมไว้ โดยมีการกำหนดพารามิเตอร์สีของแท่งแผนภูมิให้เป็นสีฟ้าอ่อน และใช้ฟังก์ชัน plt.text() ในการใส่ค่าของRateในกราฟแต่ละแท่ง

```
plt.bar(df9_y2019_Platform, df9_y2019_Rate, color='lightblue')
for index,data in enumerate(df9_y2019_Rate):
    plt.text(x=index-0.25 , y =data+1 , s=f"{data}" , fontdict=dict(fontsize=13))
```

ตั้งชื่อหัวข้อของกราฟว่า Social Media Platform Rating in 2019 โดยใช้ฟังก์ชัน title() และสร้าง label ของแกน X,Y โดยใช้ฟังก์ชัน xlabel() และ ylabel() ซึ่งแทนด้วย Platform และ Rate (%) ตามลำดับ และใช้ฟังก์ชัน show() ในการแสดงผล

```
plt.title('Social Media Platform Rating in 2019')
plt.xlabel('Platform')
plt.ylabel('Rate(%)')
plt.show()
```

โดยจากราฟที่แสดงรายละเอียดเบอร์เซ็นต์การใช้งานแพลตฟอร์มโซเชียลมีเดียของปี 2019 ด้านล่าง แสดงให้เห็นการใช้งาน Facebook และ YouTube สูงที่สุดคิดเป็น 94% ตามมาด้วยอันดับที่ 2 คือ Line คิดเป็น 85%, อันดับที่ 3 คือ Messenger คิดเป็น 76%, อันดับที่ 4 Instagram 65%, อันดับที่ 5 Twitter 55%, อันดับที่ 6 Pinterest 32%, อันดับที่ 7 LinkedIn, WhatsApp, WeChat 27%, อันดับที่ 8 คือ Twitch, Skype 25%, อันดับที่ 9 คือ Snapchat 23% และอันดับที่ 10 Tumblr 19%



ກຳທັດຂາດຄວາມກວ້າງແລະຍາວຂອງການແຈ້ງເປົ້າການໃຊ້ພັ້ນຖານ style.use ("ggplot")
ເພື່ອເພີ່ມເສັ້ນຕາງເປັນພື້ນຫຼັກຮາບ

```
[ ] plt.rcParams['figure.figsize'] = 15, 15
style.use("ggplot")
```

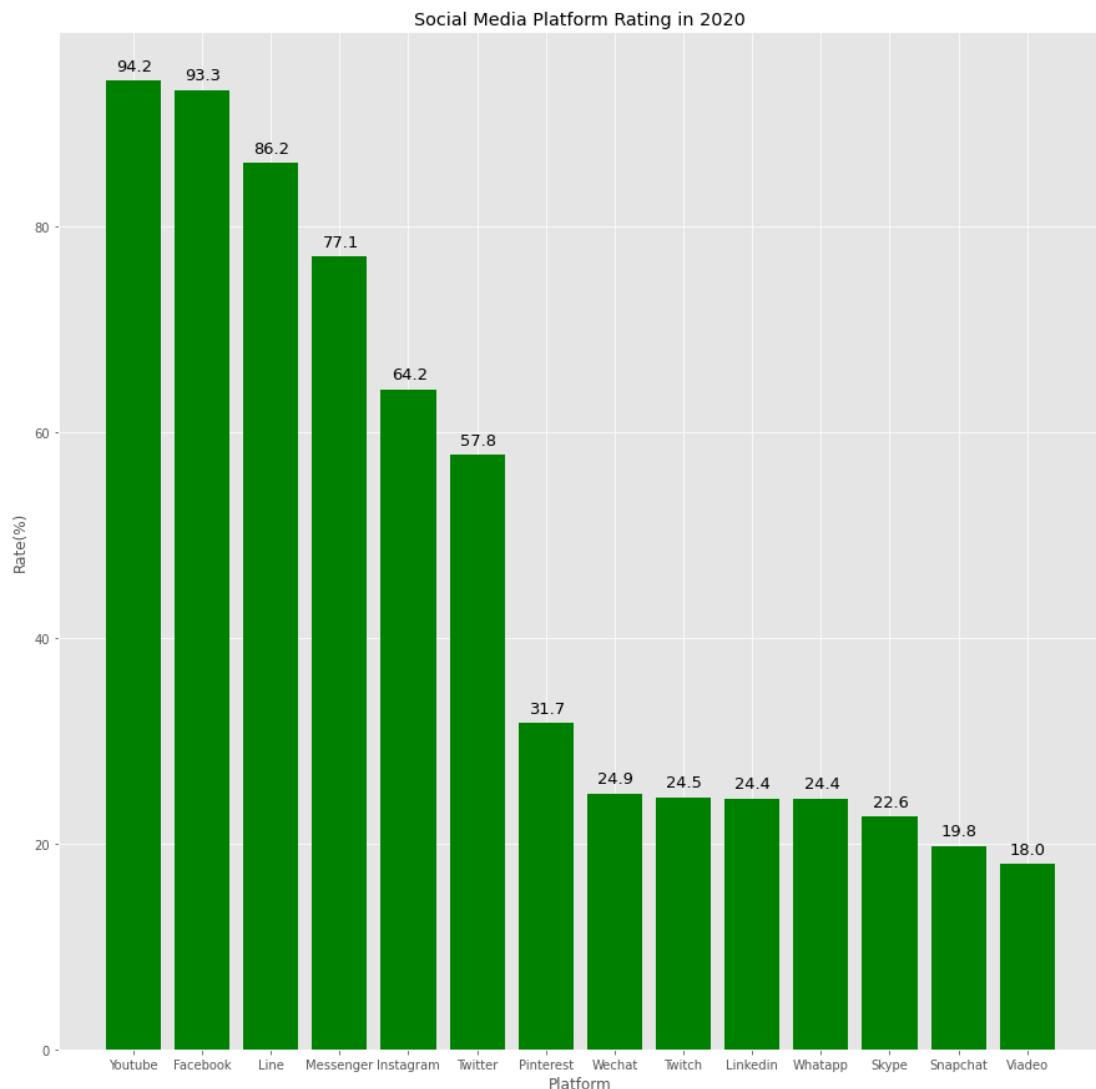
ໃຊ້ພັ້ນຖານ bar() ເພື່ອສ້າງແຜນກຸມີແທ່ງ ແລະທຳກາຣພລືອຕົກ່າທີ່ມີກາຣເຕີຢີມໄວ້ ໂດຍມີກາຣກຳທັດ
ພາຣາມີເຕີຣ໌ສີຂອງແທ່ງແຜນກຸມີໃໝ່ເປັນສີເຂົ້າວ ແລະໃຊ້ພັ້ນຖານ plt.text() ໃນກາຣໄສ່ຄ່າຂອງRateໃນການແຈ້ງ
ລະແທ່ງ

```
plt.bar(df9_y2020_Platform, df9_y2020_Rate, color='green')
for index,data in enumerate(df9_y2020_Rate):
    plt.text(x=index-0.25 , y =data+1 , s=f'{data}' , fontdict=dict(fontsize=13))
```

ตั้งชื่อหัวข้อของกราฟว่า Social Media Platform Rating in 2020 โดยใช้ฟังก์ชัน title() และสร้าง label ของแกน X,Y โดยใช้ฟังก์ชัน xlabel() และ ylabel() ซึ่งแทนด้วย Platform และ Rate (%) ตามลำดับ และใช้ฟังก์ชัน show() ในการแสดงผล

```
plt.title('Social Media Platform Rating in 2020')
plt.xlabel('Platform')
plt.ylabel('Rate(%)')
plt.show()
```

โดยจากการที่แสดงรายละเอียดเบื้องต้นการใช้งานแพลตฟอร์มโซเชียลมีเดียของปี 2020 ด้านล่าง แสดงให้เห็นการใช้งาน YouTube สูงที่สุดคิดเป็น 94.2% ตามมาด้วยอันดับที่ 2 คือ Facebook คิดเป็น 93.3%, อันดับที่ 3 คือ Line คิดเป็น 86.2%, อันดับที่ 4 Messenger 77.1%, อันดับที่ 5 Instagram 64.2%, อันดับที่ 6 Twitter 57.8%, อันดับที่ 7 Pinterest 31.7%, อันดับที่ 8 WeChat 24.9%, อันดับที่ 9 Twitch 24.5%, อันดับที่ 10 LinkedIn, WhatsApp 24.4%, อันดับที่ 11 Skype 22.6%, อันดับที่ 12 Snapchat 19.8% และ Viadeo 18%



ใช้ฟังก์ชัน subplots กำหนดมิติของกราฟให้เป็นขนาด 2×2 และ กำหนดขนาดความกว้างและยาวของกราฟเป็น 25×25

```
[ ] fig, axs = plt.subplots(2, 2)
plt.rcParams['figure.figsize'] = 25, 25
```

ในตำแหน่งมิติของกราฟแถวที่ 0 คอลัมน์ที่ 0 ให้ทำการสร้างแผนภูมิแท่ง โดยใช้ฟังก์ชัน bar() และทำการพล็อตค่าที่มีการเตรียมไว้ ซึ่งแทนด้วยข้อมูลของปี 2017 โดยมีการกำหนดพารามิเตอร์สีของแท่งแผนภูมิให้เป็นสีส้ม, ระดับความเข้มของกราฟให้เป็น 0.5 และชื่อ label เป็น 2017 และใช้ฟังก์ชัน set_title() ในการตั้งชื่อหัวข้อของกราฟว่า Social Media Platform Rating in 2017

```
axs[0, 0].bar(df9_y2017_Platform, df9_y2017_Rate, color='orange', alpha=0.5, label='2017')
axs[0, 0].set_title('Social Media Platform Rating in 2017')
```

ในตำแหน่งมิติของกราฟแถวที่ 0 คอลัมน์ที่ 1 ให้ทำการสร้างแผนภูมิแท่ง โดยใช้ฟังก์ชัน bar() และทำการพล็อตค่าที่มีการเตรียมไว้ ซึ่งแทนด้วยข้อมูลของปี 2018 โดยมีการกำหนดพารามิเตอร์สีของแท่งแผนภูมิให้เป็นสีแดง, ระดับความเข้มของกราฟให้เป็น 0.5 และชื่อ label เป็น 2018 และใช้ฟังก์ชัน set_title() ในการตั้งชื่อหัวข้อของกราฟว่า Social Media Platform Rating in 2018

```
axs[0, 1].bar(df9_y2018_Platform, df9_y2018_Rate, color='red', alpha=0.5, label='2018')
axs[0, 1].set_title('Social Media Platform Rating in 2018')
```

ในตำแหน่งมิติของกราฟแถวที่ 1 คอลัมน์ที่ 0 ให้ทำการสร้างแผนภูมิแท่ง โดยใช้ฟังก์ชัน bar() และทำการพล็อตค่าที่มีการเตรียมไว้ ซึ่งแทนด้วยข้อมูลของปี 2019 โดยมีการกำหนดพารามิเตอร์สีของแท่งแผนภูมิให้เป็นสีน้ำเงิน, ระดับความเข้มของกราฟให้เป็น 0.5 และชื่อ label เป็น 2019 และใช้ฟังก์ชัน set_title() ในการตั้งชื่อหัวข้อของกราฟว่า Social Media Platform Rating in 2019

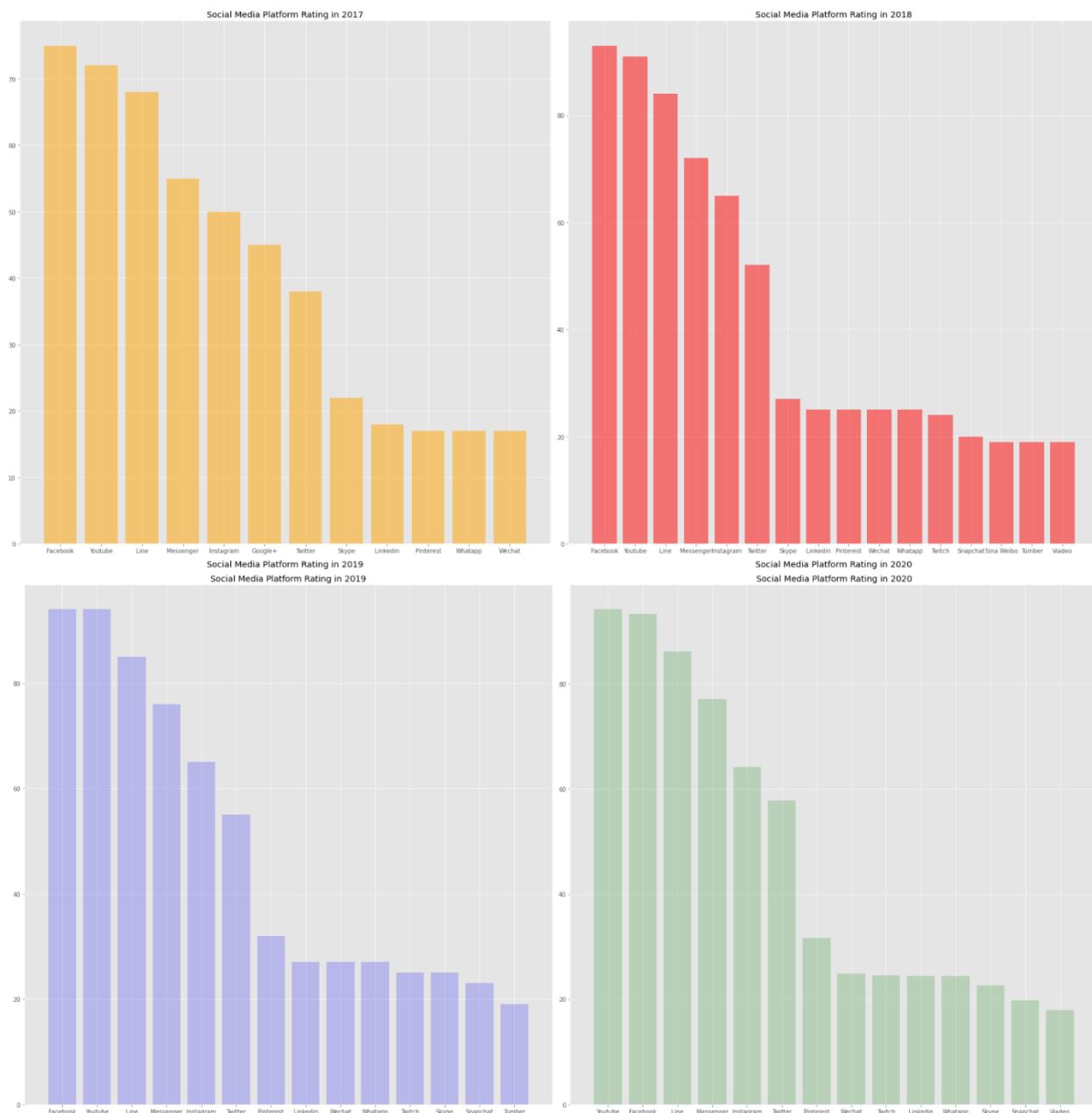
```
axs[1, 0].bar(df9_y2019_Platform, df9_y2019_Rate, color='blue', alpha=0.2, label='2019')
axs[1, 0].set_title('Social Media Platform Rating in 2019')
```

ในตำแหน่งมิติของกราฟแถวที่ 1 คอลัมน์ที่ 1 ให้ทำการสร้างแผนภูมิแท่ง โดยใช้ฟังก์ชัน bar() และทำการพล็อตค่าที่มีการเตรียมไว้ ซึ่งแทนด้วยข้อมูลของปี 2020 โดยมีการกำหนดพารามิเตอร์สีของแท่งแผนภูมิให้เป็นสีเขียว, ระดับความเข้มของกราฟให้เป็น 0.5 และชื่อ label เป็น 2020 และใช้ฟังก์ชัน set_title() ในการตั้งชื่อหัวข้อของกราฟว่า Social Media Platform Rating in 2020

```
axs[1, 1].bar(df9_y2020_Platform, df9_y2020_Rate, color='green', alpha=0.2, label='2020')
axs[1, 1].set_title('Social Media Platform Rating in 2020')
```

สร้าง label ของแกน X,Y โดยใช้ฟังก์ชัน set_xlabel() และ set_ylabel() ซึ่งแทนด้วย Platform และ Rate (%) ตามลำดับ และใช้ฟังก์ชัน Tight_layout() ในการปรับรูปกราฟให้มีความพอดี เพื่อแสดงผลแบบไม่เกิดซ่องว่างที่มากเกินไป

```
ax.set_ylabel('Rate(%)')
ax.set_xlabel('Platform')
plt.tight_layout()
```



ກຳນົດຂາດຄວາມກວ້າງແລະຍາວຂອງກາຮົບເປັນ 15×15 ແລະເຮີຍໃຊ້ `style.use("ggplot")` ເພື່ອ
ເພີ່ມເສັ້ນຕາງເປັນພື້ນໜັກກາຮົບ

```
[ ] plt.rcParams['figure.figsize'] = 15, 15
style.use("ggplot")
```

ทำการสร้างแผนภูมิแท่ง โดยใช้ฟังก์ชัน bar() และทำการพลอตค่าที่มีการเตรียมไว้ ซึ่งแทนด้วยข้อมูลของปี 2019 และ ปี 2020 โดยมีการกำหนดพารามิเตอร์สีของแท่งแผนภูมิให้เป็นสีน้ำเงินและเขียวตามลำดับ ปรับระดับความเข้มของกราฟให้เป็น 0.2 และตั้งชื่อ label เป็นชื่อปี คือ 2019 และ 2020

```
plt.bar(df9_y2019_Platform, df9_y2019_Rate, color='blue', alpha=0.2, label='2019')
plt.bar(df9_y2020_Platform, df9_y2020_Rate, color='green', alpha=0.2, label='2020')
```

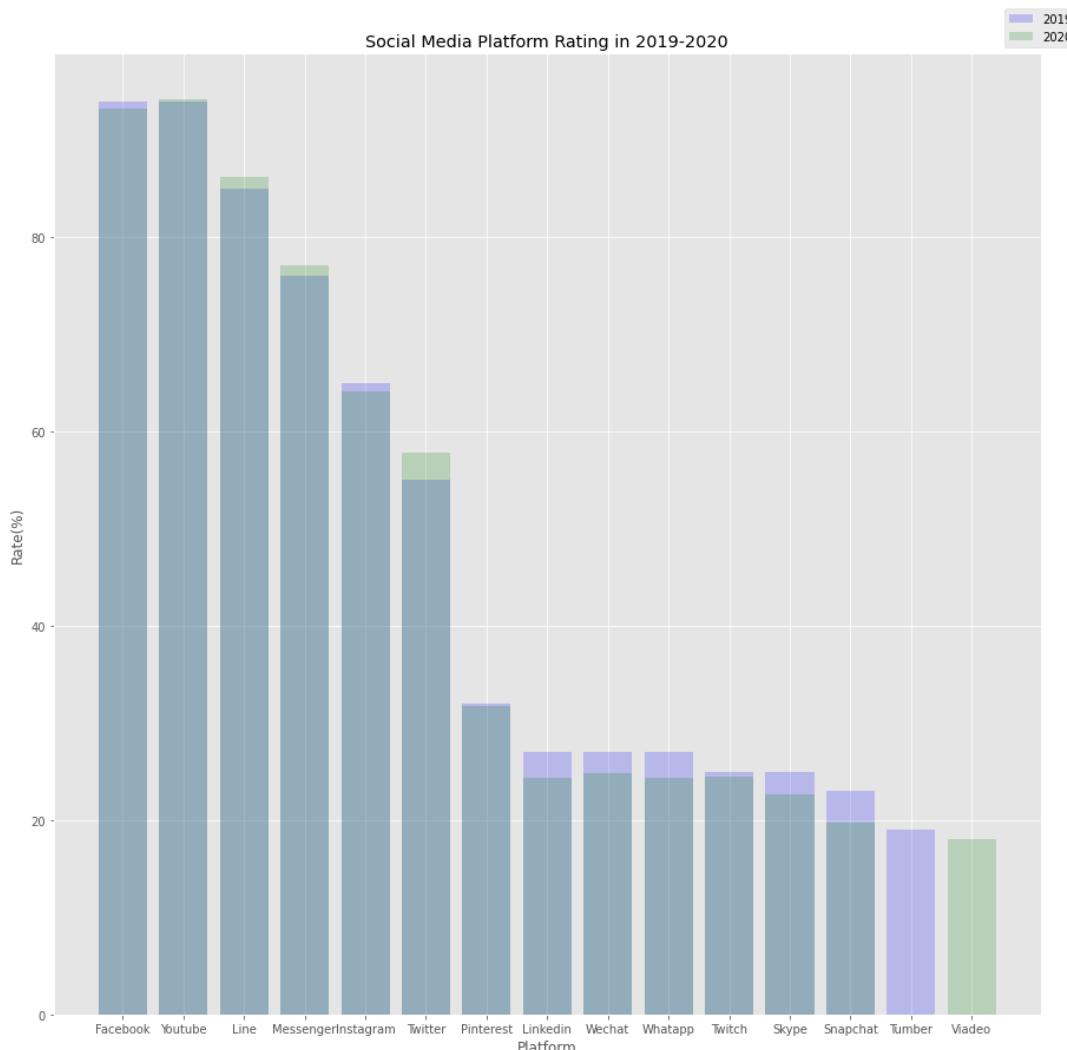
ตั้งชื่อกราฟว่า Social Media Platform Rating in 2019-2020 โดยใช้ฟังก์ชัน title() และสร้าง label ของแกน X,Y โดยใช้ฟังก์ชัน xlabel() และ ylabel() ซึ่งแทนด้วย Platform และ Rate (%) ตามลำดับ

```
plt.title('Social Media Platform Rating in 2019-2020')
plt.xlabel('Platform')
plt.ylabel('Rate(%)')
```

ใช้ฟังก์ชัน show() ในการแสดงผลกราฟ และ legend() ในการแสดงผลคำอธิบายข้อมูล โดยเพิ่มพารามิเตอร์การจัดตำแหน่งของคำอธิบายข้อมูลให้อยู่ในตำแหน่งที่เหมาะสม และใช้ฟังก์ชัน show() ในการแสดงผล

```
plt.legend(loc="lower center", bbox_to_anchor=(1.0, 1.0))
plt.show()
```

โดยจากแผนภูมิแท่งด้านล่างแสดงให้เห็นการเปลี่ยนแปลงเชิงบวกของคะแนนแพลตฟอร์มโซเชียลมีเดียของปี 2019 กับ 2020 ว่ามีการใช้งานโซเชียลมีเดียเพิ่มขึ้นในปี 2020 ในหลายแพลตฟอร์ม ได้แก่ YouTube, Line, Messenger, Twitter และ Viadeo



ทำการแสดงຂໍ້ມູນໃນ dataList 10 ອອກມາ ເພື່ອทำการตรวจສອບຄວາມຖຸກຕ້ອງຂອງຂໍ້ມູນກ່ອນທີ່
ຈະການົາໄປເພື່ອຕົກລາງໃນຂໍ້ຕອນດັ່ງນີ້

[636] dataList[10]

	Year	Messenger	Social Network	Video	Music	Game	Shopping	Map	Banking	Health and Fitness	Dating
0	2017	77.0		-	75.0	-	66.0	-	64.0	56.0	-
1	2018	95.0		-	95.0	-	85.0	-	80.0	74.0	-
2	2019	95.0		97	89.0	65	67.0	58	72.0	65.0	30
3	2020	97.4		97.6	87.8	65.6	68.6	83.4	71.6	68.1	32.4

ເຕີຍມຂໍ້ມູນທີ່ຈະນຳມາພຶດຕ ໂດຍກຳໄໝຂໍ້ມູນລູ່ຢູ່ໃນຮູບແບບລິສົດທີ່ຈຳແນກຕາມປີ ແລະເກີບຄ່າໄວ້ໃນ
ຕົວແປຣທີ່ສ້າງຂຶ້ນໃໝ່ເຫັນຂໍ້ມູນແຕ່ລະປີ ເພື່ອຄວາມສະດວກຕ່ອງກາງເຮັດໃຊ້ໃນກາຍໜັງ

```
[ ] df10_y2017 = dataList[10][dataList[10]['Year']==2017]
df10_y2018 = dataList[10][dataList[10]['Year']==2018]
df10_y2019 = dataList[10][dataList[10]['Year']==2019]
df10_y2020 = dataList[10][dataList[10]['Year']==2020]
```

ทำการครอบค่าของคอลัมน์ตามชื่อคอลัมน์ที่กำหนด

```
[ ] df10_y2017 = df10_y2017.drop(['Social Network', 'Music', 'Shopping', 'Health and Fitness', 'Dating'], axis=1)
df10_y2018 = df10_y2018.drop(['Social Network', 'Music', 'Shopping', 'Health and Fitness', 'Dating'], axis=1)
```

ทำการสร้างอาร์เรย์ใหม่ด้วย numpy.array โดยให้ :

- df10_x2017 เก็บข้อมูลคอลัมน์ใน df10_y2017 โดยจะครอบค่าคอลัมน์ Year ทั้งหมด
- df10_x2018 เก็บข้อมูลคอลัมน์ใน df10_y2018 โดยจะครอบค่าคอลัมน์ Year ทั้งหมด
- df10_x2019 เก็บข้อมูลคอลัมน์ใน df10_y2019 โดยจะครอบค่าคอลัมน์ Year ทั้งหมด
- df10_x2020 เก็บข้อมูลคอลัมน์ใน df10_y2020 โดยจะครอบค่าคอลัมน์ Year ทั้งหมด

จากนั้นทำการครอบคอลัมน์ Year ใน DataFrame df10_y2017, df10_y2018, df10_y2019 และ df10_y2020 และทำการจัดเรียงใหม่ โดยใช้ฟังก์ชัน reshape โดยให้มีมิติเหมือนกับอาร์เรย์ df10_x2017, df10_x2018, df10_x2019 และ df10_x2020 ตามลำดับ

```
[ ] df10_x2017 = np.array(df10_y2017.columns.drop('Year'))
df10_y2017 = np.array(df10_y2017.drop('Year', axis=1)).reshape(len(df10_x2017))
df10_x2018 = np.array(df10_y2018.columns.drop('Year'))
df10_y2018 = np.array(df10_y2018.drop('Year', axis=1)).reshape(len(df10_x2018))
df10_x2019 = np.array(df10_y2019.columns.drop('Year'))
df10_y2019 = np.array(df10_y2019.drop('Year', axis=1)).reshape(len(df10_x2019))
df10_x2020 = np.array(df10_y2020.columns.drop('Year'))
df10_y2020 = np.array(df10_y2020.drop('Year', axis=1)).reshape(len(df10_x2020))
```

ใช้ฟังก์ชัน subplots กำหนดมิติของกราฟให้เป็นขนาด 2*2 และกำหนดขนาดความกว้างและยาวของกราฟเป็น 15*15

```
[ ] fig, axs = plt.subplots(2, 2)
plt.rcParams['figure.figsize'] = 15, 15
```

ในตำแหน่งมิติของกราฟแล้วที่ 0 คอลัมน์ที่ 0 ให้ทำการสร้างแพนกวินิวัลกลม โดยใช้ฟังก์ชัน pie() และทำการพล็อตค่าที่มีการเตรียมไว้ ซึ่งแทนด้วยข้อมูลของปี 2017 โดยมีการเพิ่มค่าพารามิเตอร์ autopct เพื่อกำหนดจำนวนตำแหน่งทศนิยมที่ต้องการแสดงผล และใช้ฟังก์ชัน set_title() ในการตั้งชื่อหัวข้อของกราฟว่า Mobile Application Type in 2017

```
axs[0, 0].pie(df10_y2017, labels=df10_x2017, autopct=".1f%%")
axs[0, 0].set_title('Mobile Application Type in 2017')
```

ในตำแหน่งมิติของกราฟแคลว์ 0 คอลัมน์ที่ 1 ให้ทำการสร้างแผนภูมิวงกลม โดยใช้ฟังก์ชัน pie() และทำการพล็อตค่าที่มีการเตรียมไว้ ซึ่งแทนด้วยข้อมูลของปี 2018 โดยมีการเพิ่มค่าพารามิเตอร์ autopct เพื่อกำหนดจำนวนตำแหน่งทศนิยมที่ต้องการแสดงผล และใช้ฟังก์ชัน set_title() ในการตั้งชื่อหัวข้อของกราฟว่า Mobile Application Type in 2018

```
axs[0, 1].pie(df10_y2018, labels=df10_x2018, autopct=".1f%%")
axs[0, 1].set_title('Mobile Application Type in 2018')
```

ในตำแหน่งมิติของกราฟแคลว์ 1 คอลัมน์ที่ 0 ให้ทำการสร้างแผนภูมิวงกลม โดยใช้ฟังก์ชัน pie() และทำการพล็อตค่าที่มีการเตรียมไว้ ซึ่งแทนด้วยข้อมูลของปี 2019 โดยมีการเพิ่มค่าพารามิเตอร์ autopct เพื่อกำหนดจำนวนตำแหน่งทศนิยมที่ต้องการแสดงผล และใช้ฟังก์ชัน set_title() ในการตั้งชื่อหัวข้อของกราฟว่า Mobile Application Type in 2019

```
axs[1, 0].pie(df10_y2019, labels=df10_x2019, autopct=".1f%%")
axs[1, 0].set_title('Mobile Application Type in 2019')
```

ในตำแหน่งมิติของกราฟแคลว์ 1 คอลัมน์ที่ 1 ให้ทำการสร้างแผนภูมิวงกลม โดยใช้ฟังก์ชัน pie() และทำการพล็อตค่าที่มีการเตรียมไว้ ซึ่งแทนด้วยข้อมูลของปี 2020 โดยมีการเพิ่มค่าพารามิเตอร์ autopct เพื่อกำหนดจำนวนตำแหน่งทศนิยมที่ต้องการแสดงผล และใช้ฟังก์ชัน set_title() ในการตั้งชื่อหัวข้อของกราฟว่า Mobile Application Type in 2020

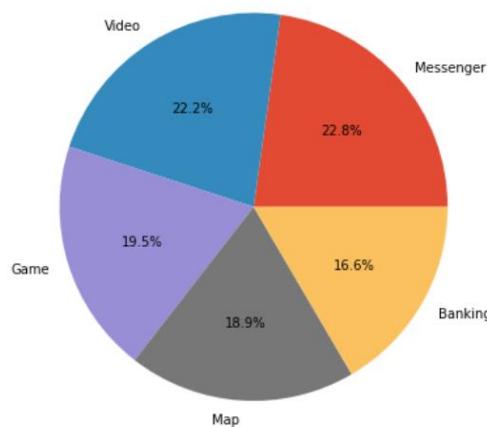
```
axs[1, 1].pie(df10_y2020, labels=df10_x2020, autopct=".1f%%")
axs[1, 1].set_title('Mobile Application Type in 2020')
```

แสดง label ของ pie chart โดยใช้ฟังก์ชัน ax.label_outer()

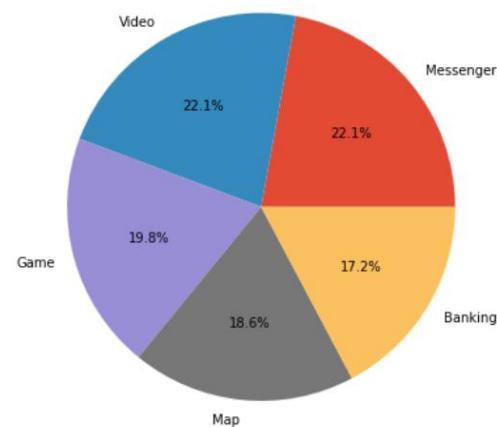
```
for ax in axs.flat:
    ax.label_outer()
```

โดยจากการแสดงเปอร์เซ็นต์ของประเภทการใช้งานแอปพลิเคชันโทรศัพท์มือถือในปี 2017-2020 ด้านล่าง แสดงให้เห็นว่าประเภทของแอปพลิเคชันโทรศัพท์มือถือที่ได้รับความนิยมสูง ได้แก่ แอปพลิเคชันเพื่อความบันเทิง, แอปพลิเคชันแผนที่, Messenger, โซเชียลมีเดีย, e-Commerce และ e-Payment

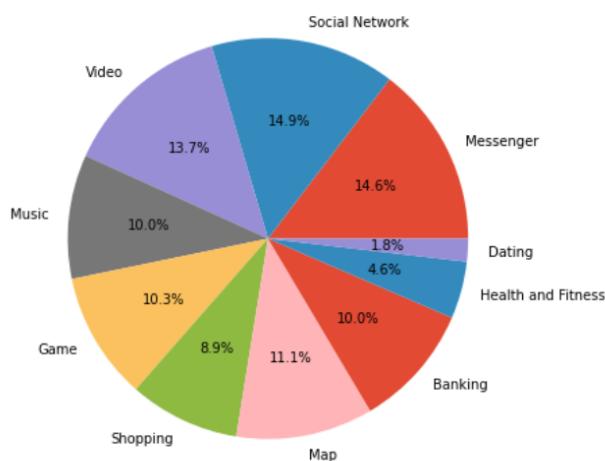
Mobile Application Type in 2017



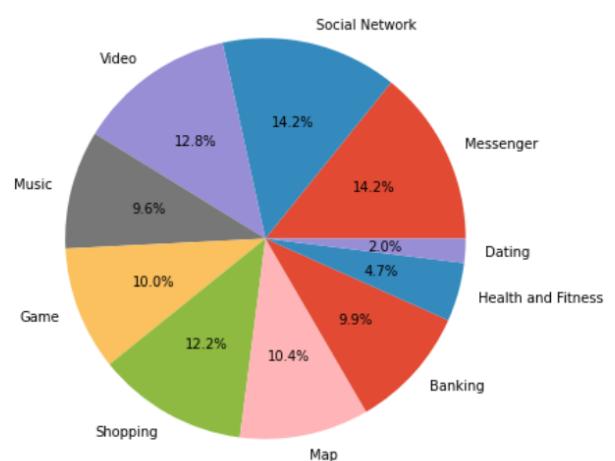
Mobile Application Type in 2018



Mobile Application Type in 2019



Mobile Application Type in 2020



สรุปผล

จากการที่กลุ่มของเราได้นำข้อมูลที่เก็บรวบรวมมาจากการหลากหลายแหล่งมาทำการวิเคราะห์ องค์ประกอบและภาพรวมของอุตสาหกรรมเทคโนโลยีสารสนเทศและการสื่อสาร จะเห็นได้ว่า สถานการณ์โควิดนั้นทำให้เกิดผลกระทบต่อการเปลี่ยนแปลงของพฤติกรรมการใช้อินเตอร์เน็ตของคนไทยเป็นอย่างมากไม่ว่าจะทั้งในเรื่อง การเรียนการทำงานออนไลน์ การซื้อของออนไลน์ การใช้โซเชียลมีเดีย รวมถึงการทำธุรกรรมทางการเงินต่างๆ ผ่านทางสื่ออิเล็กทรอนิกส์ ซึ่งกิจกรรมต่างๆ เราจะเห็นว่า มีการเริ่มมาปรับเปลี่ยนให้มาทำในรูปแบบออนไลน์เพิ่มมากขึ้น ทั้งนี้เพื่อเป็นการลดความเสี่ยงในการติดเชื้อและแพร่กระจายเชื้อไวรัสแก่คนรอบข้าง ที่เป็นผลมาจากการการเว้นระยะห่างของสังคม (Social Distancing)

ซึ่งการที่เราปรับเปลี่ยนมาทำกิจกรรมส่วนใหญ่ในรูปแบบออนไลน์ ทำให้จำนวนชั่วโมงการอินเตอร์เน็ตของคนไทยในแต่ละวันเพิ่มสูงขึ้นจากเดิมเป็นอย่างมาก โดยสังเกตุได้อย่างชัดเจนจากจำนวนชั่วโมงการอินเตอร์เน็ตในวันหยุดซึ่งต่างจากปกติที่มีค่าพุ่งสูงขึ้นจากเดิม จนใกล้เคียงกับจำนวนชั่วโมงการอินเตอร์เน็ตในวันธรรมดายิ่งไม่เกิดการแพร่ระบาดของโควิด-19 ซึ่งสาเหตุที่ทำให้จำนวนชั่วโมงการอินเตอร์เน็ตเพิ่มมากขึ้นอาจเป็นเพราะ มีนักเรียนและนักศึกษาที่จะต้องเรียนผ่านช่องทางแพลตฟอร์มออนไลน์ ตัวอย่างเช่น Zoom, Microsoft team เป็นต้น พนักงานบริษัทกับคุณที่ทำงานอาชีพอื่นที่ไม่จำเป็นต้องทำงานในสถานที่ทำงานที่ต้องมีการนัดประชุมงาน หรือส่งงานให้กับลูกค้าหรือบริษัทผ่านทางอินเตอร์เน็ต รวมถึงการที่ทุกคนไม่ได้ออกมารวมตัวพบปะกันอย่างปกติ ทำให้ผู้คนใช้โซเชียลมีเดียในการติดต่อสื่อสารรวมถึงสร้างความบันเทิงให้กับตัวเองจากการเล่นเกมหรือดูหนังฟังเพลง

การซื้อของออนไลน์เองก็เป็นกิจกรรมทางอินเตอร์เน็ตที่ผู้คนให้ความสนใจมากและเป็นที่นิยม เช่นกัน เนื่องจากมีความสะดวกสบาย ช่วยลดความเสี่ยงในการติดเชื้อไวรัสโควิด และตอบสนองต่อ สถานการณ์ในปัจจุบันที่เริ่มมีการปรับเปลี่ยนมาเป็นสังคมไร้เงินสด ที่จะชำระเงินผ่านทางระบบดิจิทัล แทนเงินสด ซึ่งทั้งหมดที่กล่าวมาข้างต้นส่งผลทำให้มีการใช้งานแต่ละแพลตฟอร์มออนไลน์ประเภทต่างๆ สูงขึ้นตามเช่นกัน

บรรณาธุกกรม

นางสาว รัชนีกร ดีดวงพันธ์.ธรรมมากิบาลเทคโนโลยีสารสนเทศในองค์กร.สีบคัน 1 พฤศจิกายน 2564, จาก sites.google.com/site/sarasnthesnix2528/bth-thi-2-khwam-hmay-laea-khwam-sakhay-khxng-thekhnoloyi-sarsnthes/2-4-kar-phathnakar-khxng-thekhnoloyi-sar-thes

ETDAofficialRegist.(24 May 2019). Thailand Internet User Profile 2018.

สีบคัน 27 กุมภาพันธ์ 2564, จาก www.slideshare.net/ETDAofficialRegist/thailand-internet-user-profile-2018-147392373

สำนักยุทธศาสตร์สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกซ์กระทรวงดิจิทัลเพื่อเศรษฐกิจและสังคม.(มกราคม 2563).รายงานผลการสำรวจพฤติกรรมผู้ใช้อินเทอร์เน็ตในประเทศไทย ปี 2562. www.etda.or.th/th/Useful-Resource/publications/Thailand-Internet-User-Behavior-2019.aspx

สำนักยุทธศาสตร์สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกซ์กระทรวงดิจิทัลเพื่อเศรษฐกิจและสังคม.(ธันวาคม 2563).รายงานผลการสำรวจพฤติกรรมผู้ใช้อินเทอร์เน็ตในประเทศไทย ปี 2563. สีบคัน 11 เมษายน 2564, จาก www.etda.or.th/th/Useful-Resource/publications/Thailand-Internet-User-Behavior-2020.aspx

SIMON KEMP.(1 February 2018).DIGITAL 2018 THAILAND. สีบคัน 11 เมษายน 2564, <https://datareportal.com/reports/digital-2018-thailand>

SIMON KEMP.(31 January 2019).DIGITAL 2019 THAILAND. สีบคัน 11 เมษายน 2564, <https://datareportal.com/reports/digital-2019-thailand>

SIMON KEMP.(18 February 2020).DIGITAL 2020 THAILAND. สีบคัน 11 เมษายน 2564, จาก <https://datareportal.com/reports/digital-2020-thailand>

SIMON KEMP.(11 February 2021).DIGITAL 2021 THAILAND. สีบคัน 11 เมษายน 2564, จาก <https://datareportal.com/reports/digital-2021-thailand>

ธนาคารแห่งประเทศไทย.(30 เม.ย. 2564).ธุรกรรมภาคร่วมระบบการชำระเงิน. สีบคัน 2 พฤษภาคม 2564, www.bot.or.th/Thai/Statistics/PaymentSystems/Pages/StatPaymentTransactions.aspx

PPTV Online .(23 ก.พ. 2564).เปิดสถิติใช้อินเทอร์เน็ตคนไทย ปี 2020 "ช้อปเก่ง โอนเก่ง". สืบค้น 1 พฤษภาคม 2564, จาก www.prachachat.net/ict/news-603829 www.pptvhd36.com/news/สั่งคุม/142614

ประชาชาติธุรกิจ. (1 กุมภาพันธ์ 2564). อ่านเกรท “โซเชียล” พลิกเกมต่อยอดธุรกิจด้วย “ข้อมูล”. สืบค้น 2 พฤษภาคม 2564, จาก www.prachachat.net/ict/news-603829

Creative Talk Team,(November 2019). สืบค้น สืบค้น 1 พฤษภาคม 2564 , จาก creativetalklive.com/thailand-e-commerce-statistics-2020/

ฝ่ายนโยบายการเงินและฝ่ายนโยบายระบบการชำระเงิน.(21 ก.ค. 2020).มุ่งสู่เศรษฐกิจไร้เงินสด: พฤติกรรมผู้บริโภคและโอกาสของธุรกิจไทยช่วงโควิด19 .สืบค้น 1 พฤษภาคม 2564, จาก www.bot.or.th/Thai/ResearchAndPublications/articles/Pages/Article_21Jul2020.aspx

