



PYTHON

PROJECT REPORT

OF รายได้และค่าใช้จ่ายการท่องเที่ยวจากนักท่องเที่ยวชาวต่างชาติที่เดินทางเข้าประเทศไทย ปี 2019

BACHELOR OF ENGINEERING

COMPUTER ENGINEERING

SUBMITTED BY

WATTHACHAI TAECHALUE

KMITL No. 65015137

SUBMITTED TO

THANA HONGSUWAN

DEPARTMENT OF COMPUTER ENGINEERING



King Mongkut's Institute of Technology Ladkrabang

Session 2022-Q1

โครงสร้างข้อมูลที่นำมาใช้ในโครงการ

โครงสร้างข้อมูลที่ผู้ออกแบบนำมาใช้งานคือ Pandas Data Structure โดยใช้รูปแบบ DataFrames(Dictionary) เป็นโครงสร้างข้อมูลแบบ 2 มิติ ประกอบด้วยแถวและคอลัมน์ คล้ายตาราง

วิธีการสร้าง DataFrame



ข้อเสนอข้อมูลที่จะนำมาใช้งาน

Country	No. of Arrivals	Length of Stay	Per Capita Spending USD	Per Capita Spending Baht	Tourism Receipts M\$ Baht	Tourism Receipts M\$ USD	Country By GROUP TOUR	No. of Arrivals By GROUP TOUR	Length of Stay By GROUP TOUR	Per Capita Spending Baht By GROUP TOUR	Per Capita Spending USD By GROUP TOUR	Tourism Receipts M\$ Baht By GROUP TOUR	Tourism Receipts M\$ USD
East Asia	2622446	6.9	5907.36	182.83	1040873.87	30884.86	East Asia	7646167	5.46	6060.5	200.52	276226.25	8302.62
Japan	1094768	5.65	5453.63	166.76	518461.91	15687.46	Japan	2219876	4.29	5699.47	170.21	176226.25	5392.62
Korea	18880	6.67	8884.64	272.86	845.23	25.38	Korea	2063	6.73	7232.94	221.85	181.39	5392.62
Cambodia	95080	6.81	5668.88	175.45	16712.65	513.26	Cambodia	46764	6.86	5861.95	181.43	1881.52	5392.62
Indonesia	646479	5.89	4736.38	146.39	17560.72	537.36	Indonesia	178018	5.4	5217.19	161.47	4608.81	1345.76
Laos	1882304	5.36	5181.86	166.37	50139.81	1538.09	Laos	10899	4.43	6258.29	193.7	1545.76	4608.81
Malaysia	404295	4.84	5343.44	164.52	104025.81	3235.09	Malaysia	184042	3.88	5668.55	175.44	20402.55	5945.76
Myanmar	176880	6.71	5976.51	184.97	15234.09	471.5	Myanmar	47913	5.68	5568.38	170.49	1471.34	4345.76
Philippines	407832	6.01	5076.34	157.11	16543.04	513.25	Philippines	84786	4.79	5545.17	171.56	2226.21	6345.76
Singapore	1388112	5.67	6361.62	196.95	42146.11	1266.45	Singapore	114841	5.38	6294.62	193.56	1707.81	4945.76
Vietnam	3391723	6.54	5087.74	157.47	33961.42	1055.16	Vietnam	164090	4.83	4638.87	143.51	7780.83	2245.76
China	2825697	7.76	6134.22	186.05	522246.76	16164.16	China	418712	6.67	7086.21	215.38	18812.55	5392.62
Hong Kong	1047039	5.82	6604.37	204.81	46265.44	1404.22	Hong Kong	10543	5.57	6242.68	193.21	1882.36	5392.62
Japan	1462712	6.97	5457.74	166.92	80420.36	2469.07	Japan	254576	6.35	7037.65	218.44	12495.76	3645.76
Korea	1761647	7.22	5526.98	170.88	71138.5	2202.07	Korea	164688	5.37	6503.85	202.84	21117.34	6145.76
Taiwan	616172	8.21	5592.42	173.89	30953.75	958.02	Taiwan	145324	6.05	5954.99	184.31	5176.85	1545.76
Others in East Asia	301261	7.22	6035.9	185.17	3093.11	93.78	Others in East Asia	12876	6.33	6891.42	211.66	546.97	1545.76
Europe	688080	16.85	4179.71	129.38	40506.23	1240.5	Europe	111001	12.34	4760.54	146.51	6846.14	1945.76
Australia	134768	16.19	4566.44	141.33	8894.47	269.1	Australia	10384	14.09	5312.64	165.03	780.22	2245.76
Belgium	133446	16.89	3926.36	121.52	7753.43	238.8	Belgium	1206	13.5	4846.84	146.86	606.09	1745.76
Denmark	170444	16.7	5160.26	160.9	15121.72	471.11	Denmark	22641	13.48	6061.95	188.55	1886.91	5392.62
Finland	131044	16.72	4846.01	149.98	10955.54	331.51	Finland	24387	12.36	5761.34	175.11	1720.48	4945.76
France	703020	17.19	5460.35	169.37	44591.81	1382.29	France	100844	13.47	4651.42	141.96	4296.77	1245.76
Germany	872385	17.29	5610.72	171.75	54485.95	1655.26	Germany	81452	14.47	4611.35	143.34	5408.59	1545.76
Italy	205112	15.8	4022.85	124.14	14687.33	435.3	Italy	30734	13.56	4946.96	152.95	2052.11	5945.76
Netherlands	226818	17.35	4097.41	126.82	16110.31	488.62	Netherlands	22739	14.39	5790.51	179.22	1910.38	5392.62
Norway	131020	16.18	5342.91	162.36	13441.46	360.11	Norway	12343	14.34	5752.31	176.03	955.46	2745.76
Russia	1464657	16.92	4213.9	131.03	105026.29	3204.46	Russia	512061	12.45	4145.01	128.23	24689.97	7145.76
Spain	176097	15.38	4069.79	125.34	10999.11	340.43	Spain	27340	12.54	5145.01	159.26	1764.14	5145.76
Sweden	364206	15.13	4646.02	141.61	27902.44	835.75	Sweden	26767	12.96	5796.14	175.24	2012.42	5945.76
Switzerland	222885	16.86	4513.98	139.71	17236.72	532.3	Switzerland	17979	12.01	5912.35	182.99	1736.84	5145.76
United Kingdom	954044	17.71	4288.86	132.88	72453.57	2242.41	United Kingdom	76946	14.72	4973.37	151.93	5630.72	1645.76
East Europe	481768	18.2	4462.26	136.26	30551.49	917.46	East Europe	108465	11.49	5964.1	182.74	1846.81	5392.62
Others in Europe	212147	14.82	4527.52	140.13	15376.55	462.1	Others in Europe	39613	10.89	5193.58	166.93	2126.71	6145.76
The Americas	154668	14.86	4468.21	135.84	111486.21	3441.68	The Americas	182086	11.97	5768.59	179	9519.92	2745.76
Argentina	61885	15.08	4438.92	137.39	2910.84	90.09	Argentina	5612	14.34	5709.71	177.08	610.96	1745.76
Brazil	37205	13.41	5204.69	161.49	4384.23	127.04	Brazil	4484.25	12.17	4484.25	200.69	395.12	1145.76
Canada	251793	17.84	4254.15	131.87	13015.47	397.68	Canada	19817	16.36	5618.17	173.8	1620.12	4645.76
USA	1096177	14.33	5013.72	155.79	79670.59	2447.25	USA	85844	11.02	5670.15	175.55	1886.46	5392.62
Others in America	44478	13.08	5111.08	154.36	3685.58	111.41	Others in America	13345	10.13	6060.54	206.11	776.84	2245.76
South Asia	1918883	7.68	5467.95	168	87956.66	2684.46	South Asia	547606	6.01	5175.56	172.44	16110.18	4645.76
Bangladesh	124809	8.2	4387.17	134.4	6486.27	200.75	Bangladesh	24887	6.45	5104.4	164.79	861.18	2445.76
India	1260379	7.59	5777.08	176.8	69173.38	2095.1	India	405950	5.87	5510.21	175.25	14057.51	4245.76
Nepal	50482	10.09	5475.51	169.47	3905.27	94.87	Nepal	19755	6.87	5521.97	170.91	726.17	2145.76
Pakistan	76254	10.25	5362.5	163.16	4798.79	144.12	Pakistan	14727	7.59	5703.55	176.52	617.51	1745.76
Sri Lanka	62047	7.86	6024.29	184.45	2779.51	84.04	Sri Lanka	18516	5.1	6246.5	188.93	611.39	1745.76
Others in South Asia	36761	9.35	6751.87	206.97	1204.54	36.18	Others in South Asia	9802	6.67	6476.67	200.55	555.72	1645.76
Oceania	802334	13.36	5163.8	157.42	87012.64	2609.18	Oceania	66212	11.21	5760.58	178.29	1780.33	5145.76
Australia	802334	13.43	5592.75	171.1	60256.29	1864.94	Australia	58618	11.21	5760.58	178.29	1780.33	5145.76
New Zealand	308887	13.17	5069.65	156.91	6755.55	208.48	New Zealand	8978	11.21	6080.4	188.75	615.24	1745.76
Others in Oceania	7917	11.36	5686.16	175.99	120.4	18.11	Others in Oceania	303	12.63	6296.18	194.81	46.12	1145.76
Middle East	767796	12.89	4341.92	134.25	61795.44	1823.08	Middle East	123108	5.9	4365.2	137	7763.18	2245.76
North Africa	11464	11.26	7973.76	243.15	14462.51	44.44	North Africa	7619	6.13	7619	215.4	167.66	4645.76
Egypt	101664	12.35	5218.09	161.1	16761.9	515.4	Egypt	14662	11.1	6813.26	210.87	1157.12	3245.76
Israel													

Library ที่นำมาใช้งาน

Customtkinter - ผมมีส่วนเสริมช่วยให้โปรแกรมสวยขึ้นถ้ายังไม่มีกรุณาติดตั้งก่อนนะครับด้วย pip install customtkinter

PIL - อันนี้เป็นตัวช่วยให้โปรแกรมแสดงรูปภาพได้นะครับ ถ้ายังไม่มีกรุณาติดตั้งก่อนนะครับด้วย pip install pillow

Pandas - ถ้ายังไม่มีให้ pip install pandas นะครับ เพื่อเอาไว้อ่านไฟล์ฐานข้อมูล

Matplotlib - แสดงกราฟข้อมูล

Plotly - โลกบาลีแสดงข้อมูลเหมือนกันแต่ จะมีความ Dynamic มากขึ้น ถ้ายังไม่มีให้ pip install plotly นะครับ

Numpy - ใช้เก็บข้อมูลเป็นรูปแบบ Array

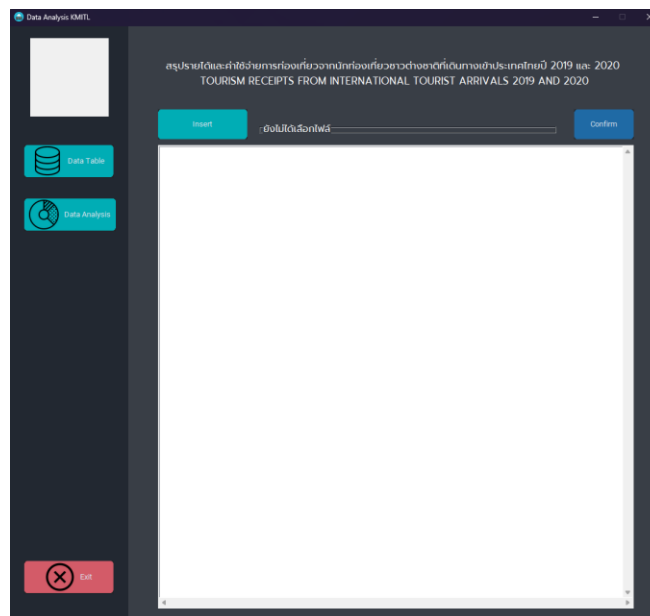
แหล่งข้อมูล

จาก Open Government Data of Thailand ศูนย์กลางข้อมูลเปิดภาครัฐ (Open Government Data) ภายใต้ชื่อ “data.go.th” ที่สำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน) หรือ สพร. พัฒนาขึ้นภายใต้แนวคิดการเป็นศูนย์กลางในการเข้าถึงข้อมูลเปิดภาครัฐของประเทศที่ให้ประชาชนสามารถเข้าถึงข้อมูลภาครัฐได้สะดวก รวดเร็ว ตลอดเวลา โดยข้อมูลที่เผยแพร่อยู่ในรูปแบบไฟล์ที่สามารถแสดงตัวอย่างข้อมูล (Preview) การแสดงข้อมูลด้วยภาพ (Visualization) และเอพีไอ (API) แบบอัตโนมัติให้กับชุดข้อมูลที่เผยแพร่ได้ รวมทั้งยังสามารถจัดการชุดข้อมูลและเมทาดาทาของข้อมูลได้อีกด้วย

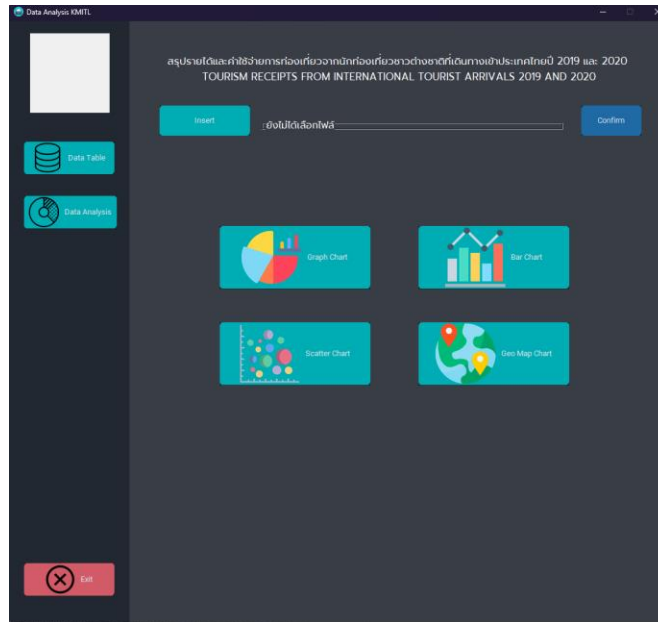
Source : <https://data.go.th/dataset/tourism-receipts-2018> และ <https://data.go.th/dataset/tourism-receipts-2019>

หน้าจอ UI และการใช้งาน

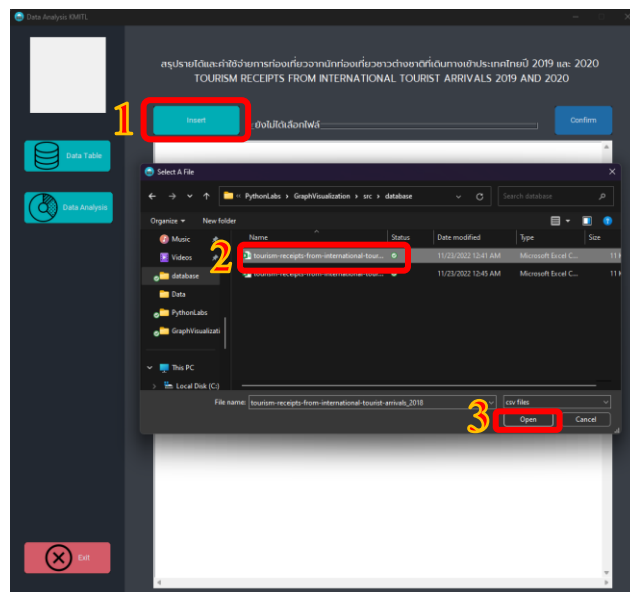
1. หน้าต่างเริ่มต้นโปรแกรม



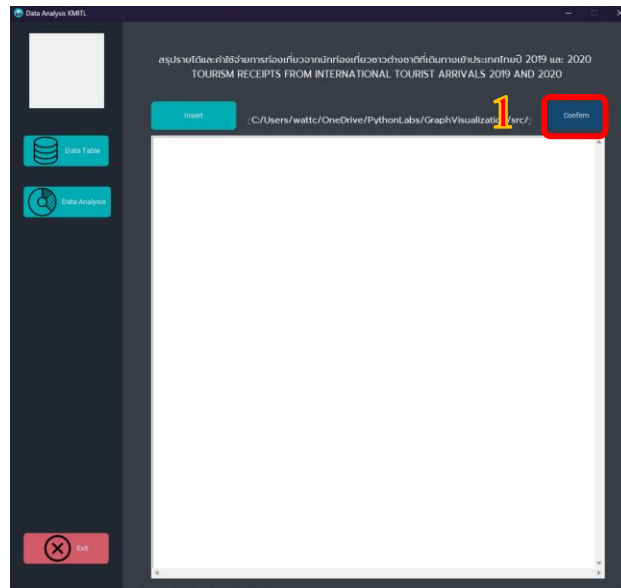
2. หน้าต่างการคำนวณกราฟต่างๆ



3. ท่านสามารถเลือก ไฟล์ เพื่อดูข้อมูลในไฟล์ .CSV หรือ .XLSX ได้ โดยผู้เขียนได้นำไฟล์ไว้แล้วที่ Folder: Data CSV Files



4. หลังจากที่ใช้ได้เลือกไฟล์แล้ว ให้ทำการกดปุ่ม Confirm ครับ

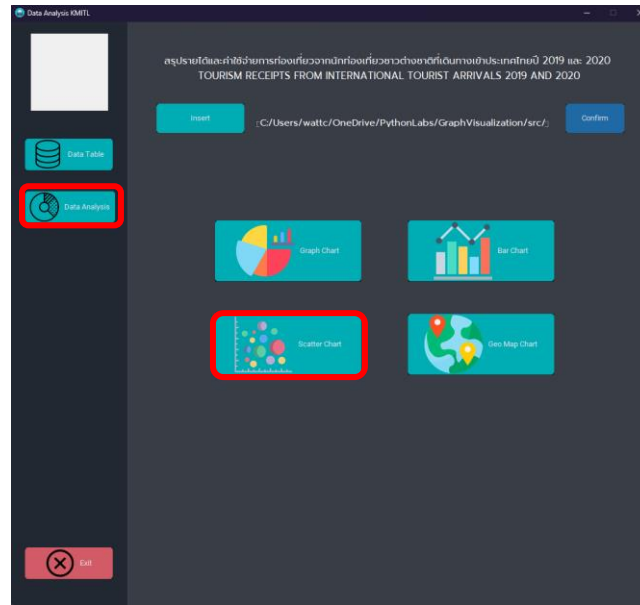


5. นี่คือตารางข้อมูลหลังจากที่ผู้ใช้งานยืนยันข้อมูล

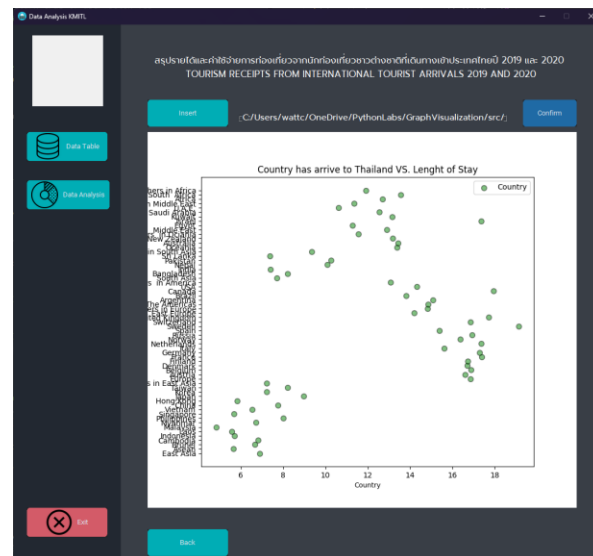
Country	No. of Arrivals	Length of Stay	Per Capita Spending Baht
East Asia	25225446	6.9	5907.36
ASEAN	10246750	5.65	5452.61
Burma	15962	6.67	6964.64
Cambodia	950980	6.81	5668.88
Indonesia	644479	5.69	4736.38
Laos	1682394	5.56	5181.86
Malaysia	4042998	4.84	5341.64
Myanmar	379880	6.71	5976.51
Philippines	407832	8.01	5076.34
Singapore	1188113	5.87	6363.42
Vietnam	1051723	6.54	5081.74
China	10625167	7.76	6334.22
Hong Kong	1047359	5.82	6004.37
Japan	1642712	8.97	5457.74
Korea	1785147	7.22	5520.98
Taiwan	674172	8.21	5592.42
Others in East Asia	101921	7.22	6919.9
Europe	6601689	16.83	4175.71
Austria	114768	16.59	4566.44
Belgium	116346	16.89	3926.36
Denmark	170344	16.7	5360.26
Finland	115384	16.72	4846.01
France	724350	17.39	3540.35
Germany	872185	17.29	3610.72
Italy	263519	15.6	4010.85
Netherlands	226618	17.35	4097.41
Norway	133020	16.38	5342.91
Russia	1466457	16.82	4232.9
Spain	176197	15.38	4046.73
Sweden	304206	19.13	4640.02
Switzerland	222985	16.86	4513.08
United Kingdom	954624	17.71	4266.86
East Europe	487668	14.2	4467.29
Others in Europe	232147	14.82	4527.52
The Americas	1534008	14.96	4890.21
Argentina	43485	15.58	4400.82
Brazil	57105	13.81	5204.69

6. หลังจากที่เราตรวจสอบข้อมูลเรียบร้อยแล้วผู้ใช้สามารถนำข้อมูลมาประมวลผลกราฟได้ตามกราฟด้านล่างครับ

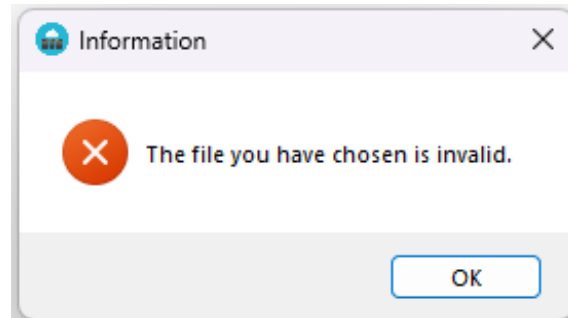
1



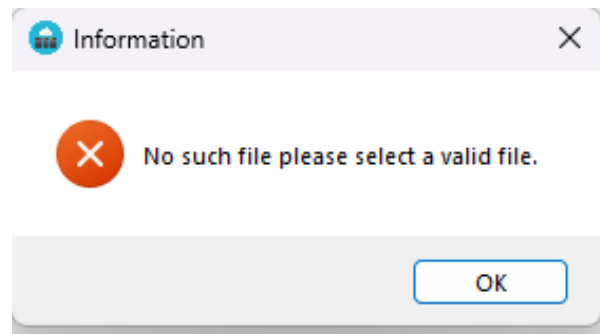
7. หลังจากผู้ใช้เลือกข้อมูลที่จะนำมาประมวลผลกราฟ ก็จะได้กราฟตามที่ใช้เลือก



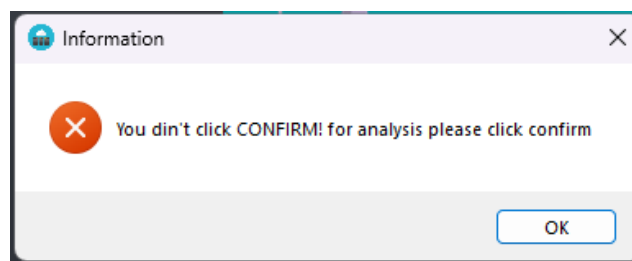
กรณีเกิดการ Error เกิดขึ้น



Error: โดยที่ผู้ใช้เลือกไฟล์มาประมวลผลข้อมูลที่ไม่ใช่ .CSV



Error: โดยที่ผู้ใช้ไม่ได้นำไฟล์มาประมวลผล



Error: โดยที่ผู้ใช้เลือกไม่ได้ยืนยันไฟล์ที่เลือกมาประมวลผล

การแบ่งเป็นฟังก์ชัน

```
1 def main():
2     #setting title
3     root.title("Data Analysis KMITL")
4     icon = PhotoImage(file = 'GraphVisualization/src/images/icon.png')
5     root.iconphoto(False, icon)
6     root.option_add("*Font", "Prompt 12")
7
8     #setting window size
9     window_width=1100
10    window_height=1000
11    screen_width = root.winfo_screenwidth()
12    screen_height = root.winfo_screenheight()
13    x_coordinate = int((screen_width/2) - (window_width/2))
14    y_coordinate = int((screen_height/2) - (window_height/2))
15    root.geometry("{}x{}+{}+{}".format(window_width, window_height, x_coordinate, y_coordinate))
16    # Useless THEME but Beautiful
17    #style = ttk.Style(root)
18    #root.tk.call('source', 'GraphVisualization/src/theme/azure dark.tcl')
19    #style.theme_use('azure')
20    root.resizable(width=False, height=False)
```

Function : main() คือฟังก์ชันทั้งหมดที่เขียนภายในโปรแกรม ตามโคดด้านบนคือ การตั้งค่าหน้าจอของโปรแกรมให้อยู่ตรงกลางเสมอโดยที่ไม่สนใจ Resolution ใหญ่หรือเล็กเพียงใดก็จะอยู่ตรงกลางเสมอ และเปลี่ยนฟอนต์กับรูปไอคอนของโปรแกรม

```
1 def data_table_page():
2     global data_table_page
3     global data_input_btn
4     global data_label_frame
5     global data_confirm_btn
6     global label_file
7     global result_data_table
8     global treescroll, treescrollx
9     data_frame_HG = "H193C46"
10
11     # button data frame
12     data_input_btn = customtkinter.CTkButton(master=root, text="input", command=load_file, width=150, height=50, compound="left",
13     fg_color="black", hover_color="black")
14     data_input_btn.place(x=150, y=100)
15
16     # Show label of the screen
17     Label(master=root, text="กรุณาใส่ค่าให้ถูกต้องกับรายการของสินค้าที่จะซื้อปี 2019 and 2020 \n TOURISM RECEIPTS FROM INTERNATIONAL TOURIST ARRIVALS 2019 AND 2020", background="H303C46", foreground="white", font=("Prompt", 12)).place(x=200, y=50)
18
19     # frame for treescroll
20     data_table_frame = tk.Label(root, text="treeview")
21     data_label_frame.place(height=100, width=500, x=200, y=200)
22
23     # data confirm btn
24     data_confirm_btn = customtkinter.CTkButton(text="Confirm", command=lambda: load_data())
25     data_confirm_btn.place(x=350, y=150, width=100, height=50)
26
27     # the file path text
28     label_file = tk.LabelFrame(text="ไฟล์ข้อมูล", background=data_frame_HG, fg="white")
29     label_file.place(x=50, y=100, height=50, width=500)
30
31     # treescroll widget
32     result_data_table = tk.Treewidget(data_label_frame)
33     result_data_table.place(relwidth=1, relheight=1) # set the height and width of the widget to 100% of its container (frame).
34
35     treescroll = tk.Scrollbar(data_label_frame, orient="vertical", command=result_data_table.yview) # command means update the y-axis view of the widget
36     treescrollx = tk.Scrollbar(data_label_frame, orient="horizontal", command=result_data_table.xview) # command means update the x-axis view of the widget
37     result_data_table.configure(scrollcommand=treescroll.set, yscrollcommand=treescrollx.set) # assign the scrollbar to the Treewidget widget
38     treescroll.pack(side="bottom", fill="y") # make the scrollbar fill the y-axis of the Treewidget widget
39     treescrollx.pack(side="right", fill="x") # make the scrollbar fill the x-axis of the Treewidget widget
```

Function : data_table_page() หน้าต่างโปรแกรมที่เราสามารถนำไฟล์มา Locate และแสดงข้อมูลของตารางในหน้าหลัก


```

1
2 def exit_btn():
3     #askyesno is a function that return true if user click yes
4     if messagebox.askyesno("ออกโปรแกรม?", "คุณต้องการออกจากโปรแกรมจริงหรือ?"):
5         root.destroy()

```

Function : exit_btn() คือการออกจากโปรแกรม

```

1 def read_file():
2     """This Function will open the file explorer and assign the chosen file path to label_file"""
3     result_data = filedialog.askopenfilename(initialdir="/",
4                                             title="Select A File",
5                                             filetype=(("csv files", "*.csv"), ("xlsx files", "*.xlsx"), ("All Files", "*.*")))
6     label_file["text"] = result_data
7     return None

```

Function : read_file() คือการให้ระบบเปิดหน้าต่างเลือกไฟล์ขึ้นมา

```

1 def load_data():
2     global file_path
3     """If the file selected is valid this will load the file into the Treeview"""
4     file_path = label_file["text"]
5     try:
6         excel_filename = r"{}".format(file_path)
7         if excel_filename[-4:] == ".csv":
8             df = pd.read_csv(excel_filename)
9         else:
10            df = pd.read_excel(excel_filename)
11
12    except ValueError:
13        tk.messagebox.showerror("Information", "The file you have chosen is invalid.")
14        return None
15    except FileNotFoundError:
16        tk.messagebox.showerror("Information", f"No such file please select a valid file.")
17        return None
18
19    clear_data()
20    result_data_table["column"] = list(df.columns)
21    result_data_table["show"] = "headings"
22    for column in result_data_table["columns"]:
23        result_data_table.heading(column, text=column) # Let the column heading = column name
24
25    df_rows = df.to_numpy().tolist() # turns the dataframe into a list of Lists
26    for row in df_rows:
27        result_data_table.insert("", "end", values=row) # inserts each List into the treeview. For parameters see http
28        s://docs.python.org/3/Library/tkinter.ttk.html#tkinter.ttk.Treeview.insert
29    return None

```

Function : load_data() คือโปรแกรมที่หลังจากเราเลือกไฟล์มาแล้ว แล้วฟังก์ชันนี้จะรับข้อมูลจาก ฟังก์ชัน read_file() นำมาอ่านข้อมูลที่ผู้ใช้งานเลือกไฟล์

```
1 clear_data()
2     result_data_table["column"] = list(df.columns)
3     result_data_table["show"] = "headings"
4     for column in result_data_table["columns"]:
5         result_data_table.heading(column, text=column) # Let the column heading = column name
6
```

Function : clear_data() หลังจากใช้ฟังก์ชัน load_data() สำเร็จแล้วฟังก์ชันนี้จะทำงานโดยล้างข้อมูลที่ดึงไฟล์มาเพื่อไม่ให้ไฟล์โหลดซ้ำในโปรแกรมและเกิด Error ที่จะไม่สามารถเลือกไฟล์อื่นได้

```

1 def bar_chart():
2     try:
3         select_dataset1 = tk.StringVar()
4         dataset_1 = ttk.Combobox(root, textvariable=select_dataset1, state='readonly')
5         dataset_1.set('No. of Arrivals')
6         dataset_1['values'] = ('No. of Arrivals', 'Length of Stay', 'Per Capita Spending Baht', 'Per Capita Spending USD'
7         , 'Tourism Receipts Mil. Baht', 'Tourism Receipts Mil. USD')
8         dataset_1.place(x=470, y=950, width=230, height=30)
9
10        def selected_dataset():
11            try:
12                global result_dataset
13                result_dataset = select_dataset1.get()
14
15                print(file_path)
16                graph_chart = pd.read_csv(file_path)
17                #make a graph chart
18                data_frame = pd.DataFrame(graph_chart)
19                figure1 = plt.Figure(figsize=(8, 7), dpi=100)
20                ax1 = figure1.add_subplot(111)
21                graph_chart1 = FigureCanvasTkAgg(figure1, root)
22                graph_chart1.get_tk_widget().place(x=250, y=200)
23
24                data_frame = data_frame[['Country', f'{result_dataset} By GROUP TOUR', f'{result_dataset} By NON GROUP T
25                OUR']].groupby('Country').sum().astype(float)
26                data_frame.plot(kind='bar', legend=True, ax=ax1, fontsize=8)
27                ax1.set_title('Country has arrive to Thailand By Group Tour and NON Group Tour')
28            except ValueError:
29                tk.messagebox.showerror("Information", "The file you have chosen is invalid")
30                return None
31            except FileNotFoundError:
32                tk.messagebox.showerror("Information", f"You din't click CONFIRM! for analysis please click confirm")
33                return None
34            except NameError:
35                tk.messagebox.showerror("Information", f"Plese choose file for analysis and then click confirm")
36                return None
37
38            #button for graph chart
39            select_data_btn = customtkinter.CTkButton(master=root, text="Confirm Data", command=selected_dataset, width=150,
40            height=50, compound="left",
41            fg_color="#00ADB5", hover_color="#C77C78")
42            select_data_btn.place(x=740, y=940)
43
44            back_btn = customtkinter.CTkButton(master=root, text="Back", command=back_to_chart_analysis , width=150, height=5
45            0, compound="left",
46            fg_color="#00ADB5", hover_color="#C77C78")
47            back_btn.place(x=250, y=940)
48
49            except ValueError:
50                tk.messagebox.showerror("Information", "The file you have chosen is invalid")
51                return None
52            except FileNotFoundError:
53                tk.messagebox.showerror("Information", f"You din't click CONFIRM! for analysis please click confirm")
54                return None
55            except NameError:
56                tk.messagebox.showerror("Information", f"Plese choose file for analysis and then click confirm")
57                return None

```

Function : bar_chart() ฟังก์ชันนี้จะสร้างการวิเคราะห์แบบ กราฟแท่ง

```

1 def graph_chart():
2     global back_btn
3     global graph_chart1
4     global figure1
5     global select_data_btn
6     global dataset_1
7     #place the combobox
8     try:
9         select_dataset1 = tk.StringVar()
10        dataset_1 = ttk.Combobox(root, textvariable=select_dataset1, state='readonly')
11        dataset_1.set('No. of Arrivals')
12        dataset_1['values'] = ('No. of Arrivals', 'Length of Stay', 'Per Capita Spending Baht', 'Per Capita Spending USD',
13        'Tourism Receipts Mil. Baht', 'Tourism Receipts Mil. USD')
14        dataset_1.place(x=470, y=950, width=230, height=30)
15
16    def selected_dataset():
17        try:
18            global result_dataset
19            result_dataset = select_dataset1.get()
20
21            print(file_path)
22            graph_chart = pd.read_csv(file_path)
23            #make a graph chart
24            data_frame = pd.DataFrame(graph_chart)
25            figure1 = plt.figure(figsize=(8, 7), dpi=100)
26            ax1 = figure1.add_subplot(111)
27            graph_chart1 = FigureCanvasTkAgg(figure1, root)
28            graph_chart1.get_tk_widget().place(x=250, y=200)
29
30            data_frame = data_frame[['Country', f'{result_dataset} By GROUP TOUR', f'{result_dataset} By NON GROUP T
31            OUR']].groupby('Country').sum().astype(float)
32            data_frame.plot(kind='line', legend=True, ax=ax1, fontsize=8)
33            ax1.set_title('Country has arrive to Thailand By Group Tour and NON Group Tour')
34            except ValueError:
35                tk.messagebox.showerror("Information", "The file you have chosen is invalid")
36                return None
37            except FileNotFoundError:
38                tk.messagebox.showerror("Information", f"You din't click CONFIRM! for analysis please click confirm")
39                return None
40            except NameError:
41                tk.messagebox.showerror("Information", f"Plese choose file for analysis and then click confirm")
42                return None
43            #button for graph chart
44            select_data_btn = customtkinter.CTkButton(master=root, text="Confirm Data", command=selected_dataset, width=150,
45            height=50, compound="left",
46            fg_color="#00ADB5", hover_color="#C77C78")
47            select_data_btn.place(x=740, y=940)
48
49            back_btn = customtkinter.CTkButton(master=root, text="Back", command=back_to_chart_analysis, width=150, height=5
50            0, compound="left",
51            fg_color="#00ADB5", hover_color="#C77C78")
52            back_btn.place(x=250, y=940)
53
54        except ValueError:
55            tk.messagebox.showerror("Information", "The file you have chosen is invalid")
56            return None
57        except FileNotFoundError:
58            tk.messagebox.showerror("Information", f"You din't click CONFIRM! for analysis please click confirm")
59            return None
60        except NameError:
61            tk.messagebox.showerror("Information", f"Plese choose file for analysis and then click confirm")
62            return None

```

Function : graph_chart() ฟังก์ชันนี้จะสร้างการวิเคราะห์แบบ กราฟเส้น

```

1  def scatter_chart():
2
3      global back_btn
4      global scatter1
5      try:
6          print(file_path)
7          scatter_chart = pd.read_csv(file_path)
8          result_data = pd.DataFrame(scatter_chart)
9          #success: make scatter chart
10         figure1 = plt.figure(figsize=(8, 7), dpi=100)
11         ax1 = figure1.add_subplot(111)
12         ax1.set_ylabel('Lenght of Stay', fontsize=9)
13         ax1.set_xlabel('Country', fontsize=9)
14         ax1.scatter(result_data['Length of Stay'], result_data['Country'], color='g', s=50, alpha=0.5, edgecolors='black', linewidths=1, )
15     scatter1 = FigureCanvasTkAgg(figure1, root)
16     scatter1.get_tk_widget().place(x=250, y=200)
17     ax1.legend(['Country'])
18     ax1.set_title('Country has arrive to Thailand VS. Lenght of Stay')
19     except ValueError:
20         tk.messagebox.showerror("Information", "The file you have chosen is invalid")
21         return None
22     except FileNotFoundError:
23         tk.messagebox.showerror("Information", f"You din't click CONFIRM! for analysis please click confirm")
24         return None
25     except NameError:
26         tk.messagebox.showerror("Information", f"Plese choose file for analysis and then click confirm")
27         return None
28
29     #button for graph chart
30     back_btn = customtkinter.CTkButton(master=root, text="Back",command=back_to_chart_analysis , width=150, height=50, compound="left", fg_color="#00ADB5", hover_color="#C77C78")
31     back_btn.place(x=250, y=940)

```

Function : scatter_chart() ฟังก์ชันนี้จะสร้างการวิเคราะห์แบบ กราฟจุด

```

1  def geo_map_chart():
2      try:
3          print(file_path)
4          graph_chart = pd.read_csv(file_path)
5          #make a graph chart
6          data = pd.DataFrame(graph_chart)
7          print(data.shape)
8
9          data['Country'] = data['Country'].dropna().apply(lambda x : x.replace(' ','').replace(', ','').split(','))
10         lst_col = 'Country'
11         data2 = pd.DataFrame({
12             col : np.repeat(data[col].values, data[lst_col].str.len())
13             for col in data.columns.drop(lst_col)
14         }).assign(**{lst_col:np.concatenate(data[lst_col].values)})(data.columns.tolist())
15
16         year_country2 = data2.groupby('No. of Arrivals')['Country'].value_counts().reset_index(name='counts')
17
18         fig = px.choropleth(year_country2, locations="Country", color="counts",
19                             locationmode='country names',
20                             animation_frame='No. of Arrivals',
21                             range_color=[0, 1],
22                             color_continuous_scale=px.colors.sequential.OrRd
23                         )
24
25         fig.update_layout(title='Comparison by country')
26         fig.show()
27
28     except ValueError:
29         tk.messagebox.showerror("Information", "The file you have chosen is invalid")
30         return None
31     except FileNotFoundError:
32         tk.messagebox.showerror("Information", f"You din't click CONFIRM! for analysis please click confirm")
33         return None
34     except NameError:
35         tk.messagebox.showerror("Information", f"Plese choose file for analysis and then click confirm")
36         return None
37

```

Function : bar_chart() ฟังก์ชันนี้จะสร้างการวิเคราะห์แบบ กราฟแท่ง

```

1  def data_analysis_page():
2      global data_analysis_frame
3      global graph_chart_btn
4      global bar_chart_btn
5      global scatter_chart_btn
6      global geo_chart_btn
7
8      # destroy the previous page
9      data_table_frame.destroy()
10
11
12      data_analysis_frame = tk.Frame(dataframe, background=data_frame_bg)
13      data_analysis_frame.place(relx=0.0, rely=0.0, relheight=1.0, relwidth=1.0)
14      #lb = tk.Label(data_analysis_frame)
15      #lb.pack()
16
17      #chartimages
18      graph_chart_img = ImageTk.PhotoImage(Image.open("GraphVisualization/src/images/graph_chart.png").resize((100,100), I
19      image.ANTIALIAS))
20      bar_chart_img = ImageTk.PhotoImage(Image.open("GraphVisualization/src/images/bar_chart.png").resize((100,100), Imag
21      e.ANTIALIAS))
22      scatter_chart_img = ImageTk.PhotoImage(Image.open("GraphVisualization/src/images/scatter_chart.png").resize((100,100
23      ), Image.ANTIALIAS))
24      geo_chart_img = ImageTk.PhotoImage(Image.open("GraphVisualization/src/images/geomap_chart.png").resize((100,100), Im
25      age.ANTIALIAS))
26
27      #buttons
28      graph_chart_btn = customtkinter.CTkButton(master=root, image=graph_chart_img, text="Graph Chart",command=lambda: gra
29      ph_chart() , width=250, height=80, compound="left",
30      fg_color="#00ADB5", hover_color="#C77C78")
31
32      bar_chart_btn = customtkinter.CTkButton(master=root, image=bar_chart_img, text="Bar Chart",command=lambda: bar_chart
33      () , width=250, height=80, compound="left",
34      fg_color="#00ADB5", hover_color="#C77C78")
35
36      scatter_chart_btn = customtkinter.CTkButton(master=root, image=scatter_chart_img, text="Scatter Chart",command=lambd
37      a:scatter_chart() , width=250, height=80, compound="left",
38      fg_color="#00ADB5", hover_color="#C77C78")
39
40      geo_chart_btn = customtkinter.CTkButton(master=root, image=geo_chart_img, text="Geo Map Chart",command=lambda: geo_m
41      ap_chart() , width=250, height=80, compound="left",
42      fg_color="#00ADB5", hover_color="#C77C78")
43
44      graph_chart_btn.place(x=350, y=340)
45      bar_chart_btn.place(x=680, y=340)
46      scatter_chart_btn.place(x=350, y=500)
47      geo_chart_btn.place(x=680, y=500)

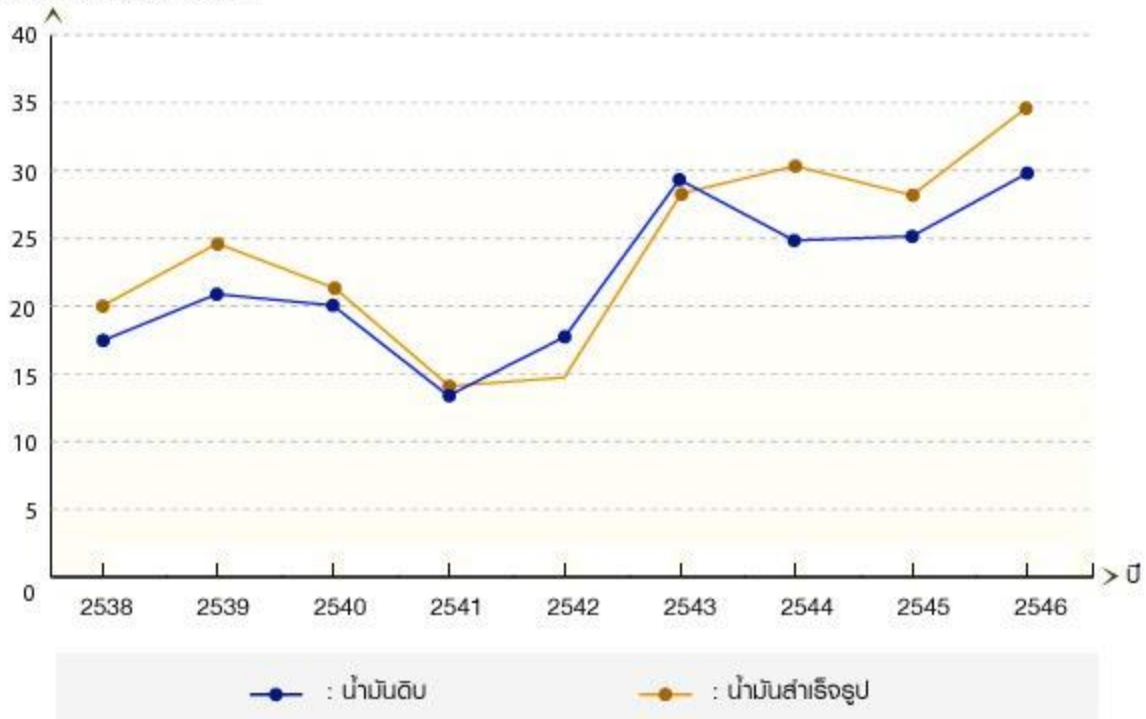
```

Function : data_analysis_page() คือหน้าที่ผู้ใช้งานจะเลือกกราฟข้อมูลมาวิเคราะห์

รูปแบบการวิเคราะห์

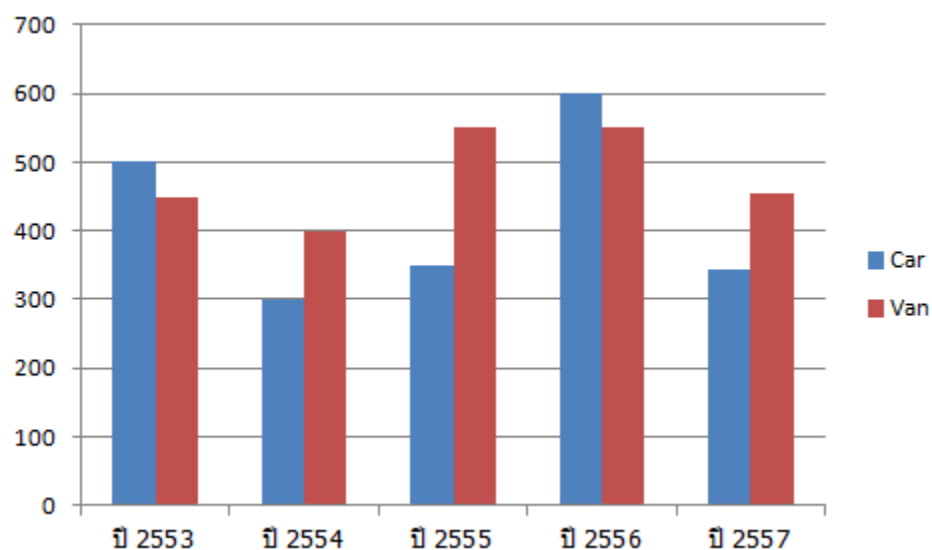
กราฟแสดงราคาเฉลี่ยของน้ำมันดิบและน้ำมันสำเร็จรูป ปี 2538 - 2546

ล้านดอลลาร์สหรัฐ / บาร์เรล



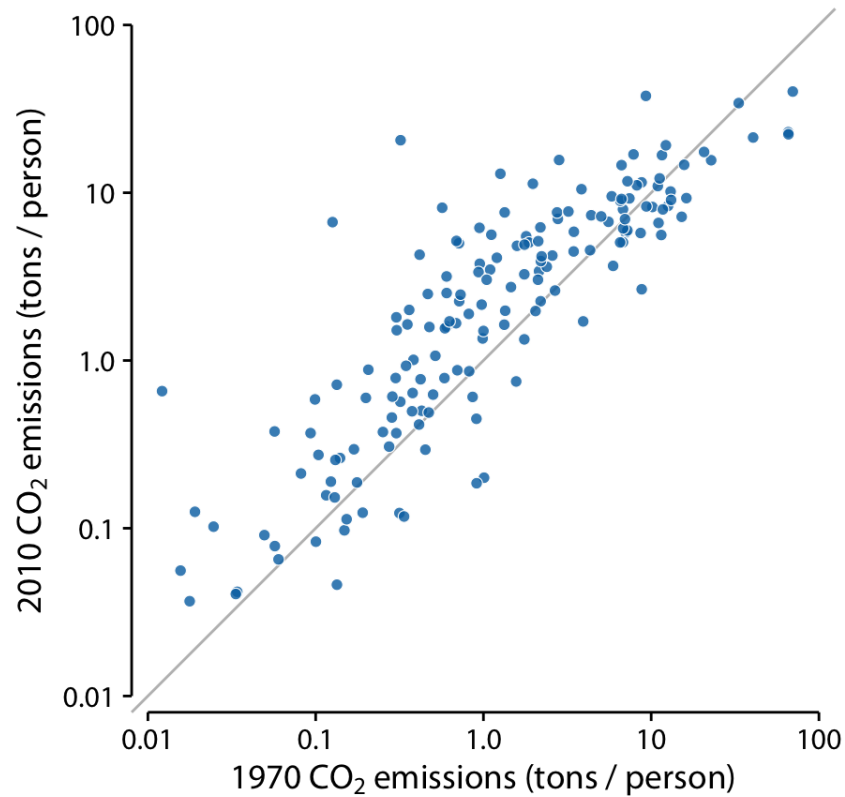
Line Chart

Line Chart คือ การแสดงกราฟของประวัติราคาของสินทรัพย์ดิจิทัลบางประเภท ในรูปแบบแผนภูมินี้ จะใช้เพียงราคาปิดของหลักทรัพย์และแสดงตามช่วงเวลา ซึ่งจะแตกต่างกับแผนภูมิราคาหรือ Price chart สามารถใช้ในช่วงเวลาใดก็ได้ แต่แผนภูมิเส้น Line chart มักใช้เพื่อแสดงถึงการเปลี่ยนแปลงของราคาแบบวันต่อวัน ซึ่งเป็นแผนภูมิประเภทที่พบบ่อย



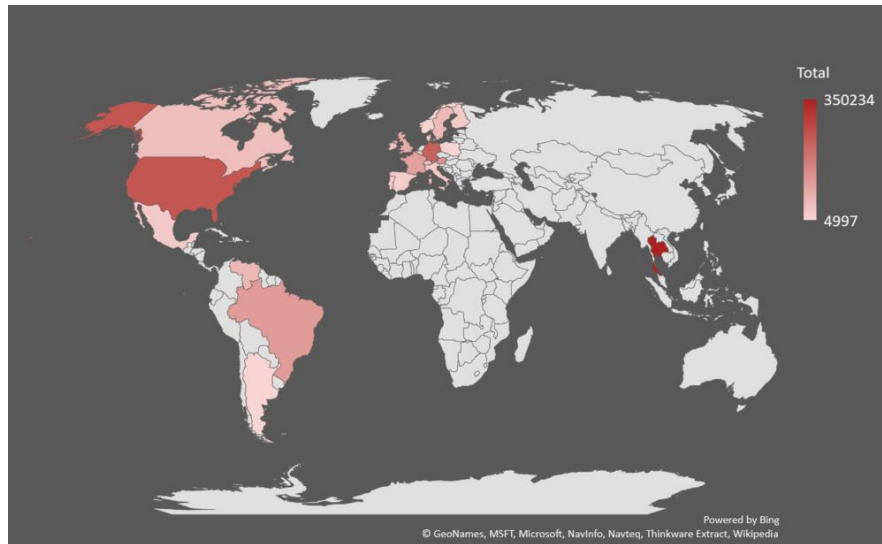
Bar Chart

แผนภูมิแท่ง (Bar Chart) เป็น แผนภาพที่ง่ายที่สุด ที่สามารถใช้เปรียบเทียบข้อมูลในแต่ละหมวดหมู่ สามารถแสดงถึงลำดับและขนาดได้ชัดเจน โดยแกนควรจะเริ่มต้นที่เลขศูนย์เพื่อให้ความสูงของแท่งแสดงถึงปริมาณที่ต้องการจะนำเสนอ ไม่ควรย่อกราฟ



Scatter Chart

Scatter Chart คือกราฟที่นำมาเพื่อวิเคราะห์ข้อมูลความสัมพันธ์ของข้อมูล โดยเปรียบเทียบระหว่างแกน X และ แกน Y ซึ่งในทางธุรกิจนิยมนำมาใช้ดูยอดขาย(Sale Amount), %Achieve, %YoY ของสินค้า(Products) หรือดูกลุ่มลูกค้า (Segment) และอื่นตามที่ใช้ต้องการวิเคราะห์



Map Chart

กราฟแบบแผนที่ (Map) เป็นการนำเสนอข้อมูลที่เป็นข้อมูลที่อยู่ ไม่ว่าจะเป็น ชื่อประเทศ (Country) ชื่อจังหวัด (Province) ชื่อรัฐ (State) ชื่อเขต (County) ชื่อเมือง (City) ชื่ออำเภอ ชื่อตำบล/แขวง (District) มานำเสนอเป็นแผนที่ ซึ่งแผนที่ประเภทนี้จะใช้ engine ของ Bing Map ต้องเชื่อมต่ออินเทอร์เน็ต เพื่อนำเสนอในรูปแบบของแผนที่