

# รายงาน

โปรแกรมสร้างกราฟจากข้อมูลพีชที่ถูกคุกคามในประเทศไทย

โดย

นายนิติธร กองแดง 65015088

เสนอ

ผศ.ธนา หงษ์สุวรรณ

รายงานนี้เป็นส่วนหนึ่งของวิชา Programming Project รหัสวิชา 01076104

ภาคเรียนที่ 1 ปีการศึกษา 2565

วิศวกรรมศาสตร์ สาขา วิศวกรรมคอมพิวเตอร์ (ต่อเนื่อง)

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

## ข้อมูลที่ใช้ในโปรแกรม

ข้อมูล : พืชที่ถูกคุกคามในประเทศไทย ก่อนค.ศ. 1994 และ ตั้งแต่ 1994 – 2001

จำนวนทั้งหมด : 1403 ข้อมูล

หัวข้อ : ลำดับที่,กลุ่มของพืช,Group of Plant,วงศ์,ชื่อวิทยาศาสตร์,ชื่อไทย,ลักษณะวิสัย,สถานภาพ (ก่อน ค.ศ. 1994),สถานภาพ (ในช่วง ค.ศ. 1994-2001)

นำเข้าจาก : Api จากเว็บไซต์ Data go

องค์กร : สำนักงานนโยบายและแผนทรัพยากรธรรมชาติและสิ่งแวดล้อม

URL : <https://data.go.th/dataset/plant-red-data>

โดยในแต่ละหัวข้อจะมีลักษณะดังนี้ :

- 1.ลำดับที่ มีข้อมูล เป็นตัวเลขแต่ละลำดับที่เรียกมา
- 2.กลุ่มของพืช มีข้อมูล กลุ่มของพืชอยู่โดยแบ่งเป็น กลุ่มพืชเมล็ดเปลือย กลุ่มเฟิน  
กลุ่มพืชดอก(ใบเลี้ยงเดี่ยว) กลุ่มพืชดอก(ใบเลี้ยงคู่)
- 3.Group of Plant มีข้อมูล กลุ่มของพืช (ภาษาอังกฤษ)
- 4.วงศ์ มีข้อมูล วงศ์ของพืช
- 5.ชื่อวิทยาศาสตร์ มีข้อมูล ชื่อวิทยาศาสตร์ของพืช
- 6.ชื่อไทย มีข้อมูล ชื่อของพืชที่เป็นภาษาไทย
- 7.ลักษณะวิสัย มีข้อมูล ลักษณะวิสัยของพืชเช่น ไม้เถา ไม้เลื้อย และ ไม้ต้น
- 8.สถานภาพ (ก่อน ค.ศ. 1994) มีข้อมูล สถานภาพโดยแบ่งเป็น R = หายากระดับโลก RT = หายาก  
ในประเทศไทย
- 9.สถานภาพ (ในช่วง ค.ศ. 1994-2001) มีข้อมูล สถานภาพโดยแบ่งเป็น EX = สูญพันธุ์ EW = สูญ  
พันธุ์ในธรรมชาติ CR = ใกล้สูญพันธุ์อย่างยิ่ง EN = ใกล้สูญพันธุ์ VU = มีแนวโน้มใกล้สูญพันธุ์ NT =  
ใกล้ถูกคุกคาม LC = เป็นกังวลน้อยที่สุด (CR),(EN),(VU),(NT) = หายากระดับโลกตามเกณฑ์ก่อนปี  
1944

## หน้า UI

ฟิชที่ถูกคุกคามในประเทศไทย

overview ข้อมูลทั้งหมด : 1403 จำนวน

2 — ประเภทของกลุ่มของฟิช ▾

ตกลง — 3

4 — ชื่อวงศ์ ▾

ตกลง — 5

เลือกหรือพิมพ์เพื่อค้นหารายละเอียด

6 — ชื่อไทย ▾

ค้นหา — 7

8 — ชื่อวิทยาศาสตร์ ▾

ค้นหา — 9

### รายละเอียด

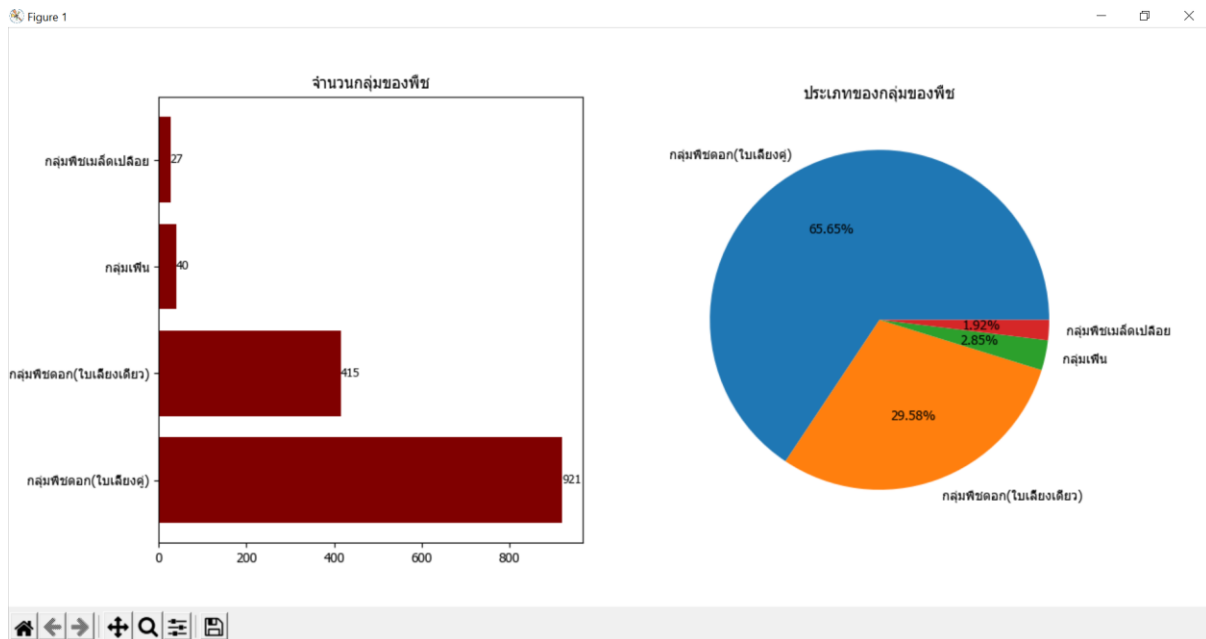
- 1.แสดงจำนวนข้อมูลทั้งหมดที่รับมาจาก API
- 2.Combo box(1) ที่มีค่า ประเภทของกลุ่มของฟิช สถานภาพ ลักษณะวิสัย
- 3.ปุ่มตกลง(1) ใช้เพื่อส่งข้อมูลเข้าไปในฟังก์ชันที่กำหนด
- 4.Combo box(2) ที่มีค่าวงศ์ของฟิช
- 5.ปุ่มตกลง(2) ใช้เพื่อส่งข้อมูลเข้าไปในฟังก์ชันที่กำหนด
- 6.Combo box(3) ที่มีค่าชื่อไทยของชื่อ
- 7.ปุ่มค้นหา(1) ใช้เพื่อส่งข้อมูลเข้าไปในฟังก์ชันที่กำหนด
- 8.Combo box(4) ที่มีค่าชื่อวิทยาศาสตร์ของฟิช
- 9.ปุ่มค้นหา(2) ใช้เพื่อส่งข้อมูลเข้าไปในฟังก์ชันที่กำหนด

เมื่อใส่ข้อมูลไม่ว่าจะในเป็นข้อมูลใน Combo box หรือพิมพ์ใส่ Combo box ที่ตรงกับข้อมูลที่มีก็จะแสดงกราฟดังนี้

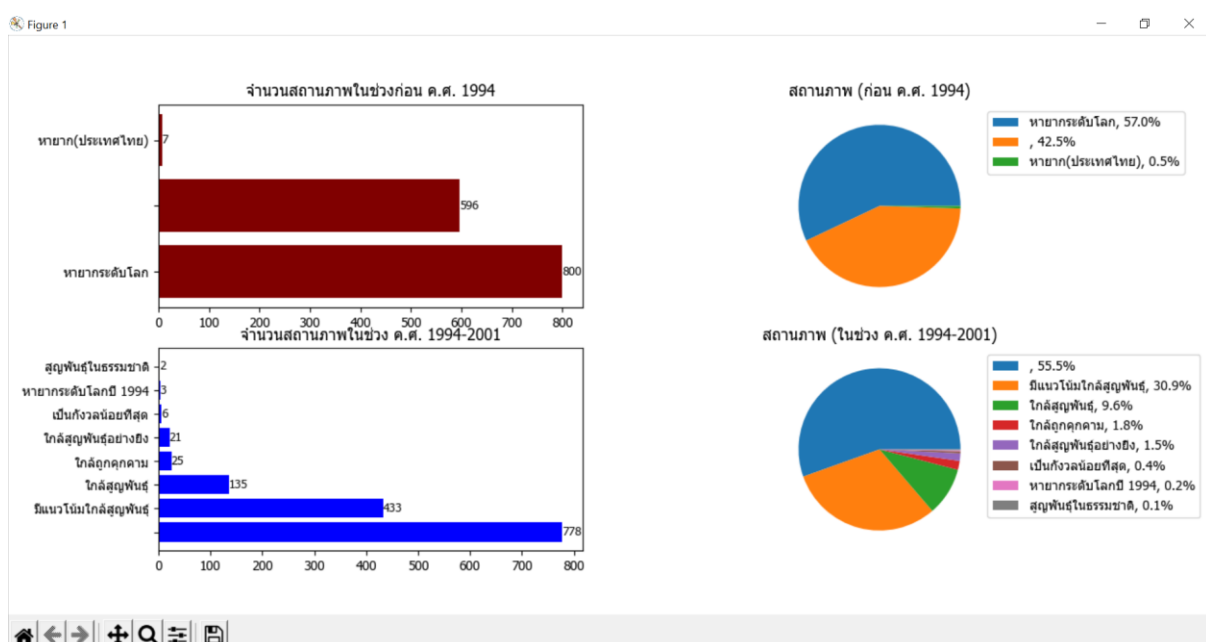
## หน้าต่างกราฟ

Combo box ที่มีค่า ประเภทของกลุ่มของพืช สถานภาพ ลักษณะนิสัย

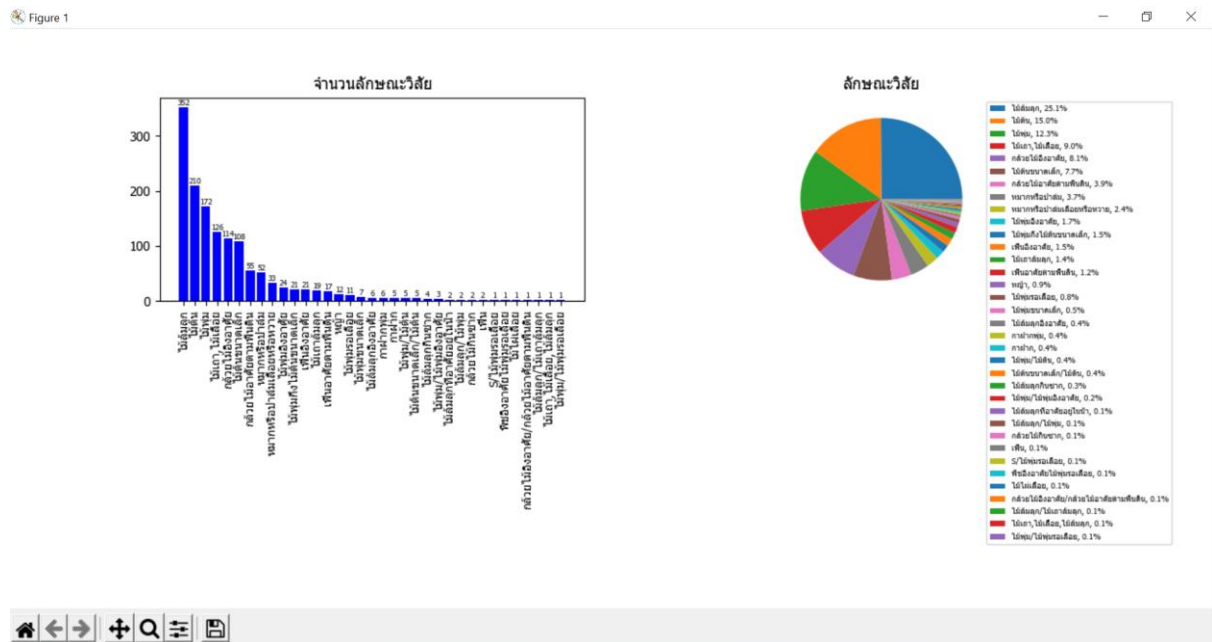
ประเภทของกลุ่มพืช



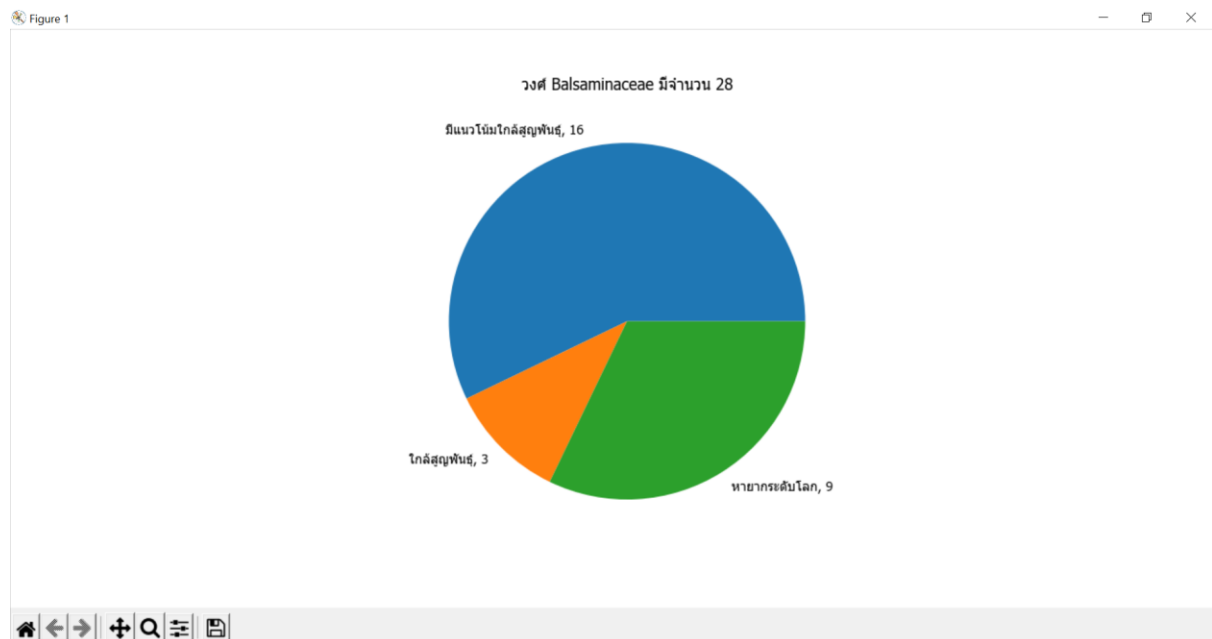
## สถานภาพ




## ลักษณะวิสัย




Combo box ที่มีค่าว่างของพีช



หน้าต่างแสดงข้อมูลจาก Combo box ชื่อไทย และ ชื่อวิทยาศาสตร์

 DATA ×



_id	321
ลำดับที่	321
กลุ่มของพืช	กลุ่มพืชดอก(ใบเลี้ยงคู่)
Group of Plant	Angiospermae(Dicotyledonae)
วงศ์	Dipterocarpaceae
ชื่อวิทยาศาสตร์	Shorea farinosa C.E.C. Fisch.
ชื่อไทย	กระบากดำ
ลักษณะวิสัย	ไม้ต้น
สถานภาพ (ก่อน ค.ศ. 1994)	หายากระดับโลก
สถานภาพ (ในช่วง ค.ศ. 1994-2001)	ใกล้สูญพันธุ์อย่างยิ่ง
Name: 320, dtype: object	

OK

## โครงสร้างของโปรแกรม

### Library

1. Matplotlib ใช้ในการสร้างกราฟ
2. Pandas ใช้ในการจัดการข้อมูลที่ได้มาจาก API
3. Requests ใช้ในการดึงข้อมูล
4. Tkinter ใช้ในการสร้าง UI ของโปรแกรม

```
import matplotlib.pyplot as plt
from pandas import DataFrame
from requests import get
from tkinter.ttk import Combobox
from tkinter import messagebox
from tkinter import *
```

### การดึงข้อมูล

ดึงข้อมูลจาก Data go โดยใช้ api key ที่ได้มาจากการสมัครสมาชิก และใช้ resource id จากหัวข้อ พืชที่ถูกคุกคาม limit ข้อมูลไว้ที่ 2000 ใช้ try กับ except เพื่อเช็คว่าได้เชื่อมต่อข้อมูล และดึงข้อมูลจาก api ได้หรือไม่ ถ้าไม่ให้แสดงข้อความไม่สามารถเชื่อมต่อกับ API กรุณาลองใหม่อีกครั้ง

```
try:
    #user token สำหรับใช้ผ่าน Data go
    api_key = "avxkbRLa0Guo9lEPxzhVWQIHahKIdruq"
    headers = {
        "api-key": api_key,
    }

    #resource id ใช้ในการดึงข้อมูลที่ต้องการ limit ไว้ที่ 2000
    params = {"resource_id": "c789e1be-ab06-43dd-93fa-80e55f6b44ac", "limit": 2000}
    #requests ไปที่ data go
    r = get(
        "https://opendata.go.th/get-ckan/datastore_search", params, headers=headers
    )
    #เช็คไม่สำเร็จให้แสดงข้อความไม่สามารถเชื่อมต่อได้
except:
    print("ไม่สามารถเชื่อมต่อกับ API ลองใหม่อีกครั้ง")
    exit()
#ถ้า requests สำเร็จ
if r.ok:
    try:
        j = r.json()
        records = j["result"][0]["records"]
    except:
        print("ไม่สามารถเชื่อมต่อกับ API ลองใหม่อีกครั้ง")
        exit()
```

เมื่อได้ข้อมูลมาแล้วสร้าง DataFrame จากข้อมูลที่ได้รับมาโดยคำอธิบายสถานะภาพด้วยความหมาย และตัดช่องว่างทิ้ง ส่วนอีก DataFrame ใช้ในการตัดสัญลักษณ์พิเศษทิ้งทั้งหมด ยกเว้น spacebar และวรรณยุกต์

```
df = DataFrame(records).replace(' ', '', regex=True).replace("VU", "มีแนวโน้มใกล้สูญพันธุ์").replace("EX", "สูญพันธุ์").replace("EW", "สูญพันธุ์ใกล้สูญพันธุ์")
dfforfind = DataFrame(records).replace(r'[\^\\$%&"]', '', regex=True).replace("VU", "มีแนวโน้มใกล้สูญพันธุ์").replace("EX", "สูญพันธุ์").replace("EW", "สูญพันธุ์ใกล้สูญพันธุ์")
```

ฟังก็ชื่น

ฟังก์ชัน send

เมื่อกดตกลง(1) จะทำการเรียกค่ามาจาก Combo box(1) และเช็คและเรียกฟังก์ชันขึ้นขึ้นมา

```
def send():
    #สร้างตัวแปรเก็บค่ามาจาก combobox
    get = sub.get()
    if get == 'ประเภทของกลุ่มของพืช':
        countofplant()
    elif get == 'สถานภาพ':
        status1()
    elif get == 'ลักษณะนิสัย':
        shape()
    else:
        messagebox.showinfo("ALERT", "ข้อมูลไม่ถูกต้อง")
```

ฟังก์ชัน countofplant

เมื่อเลือกประเภทของพีชจะเรียกใช้งานฟังก์ชัน countofplant เมื่อฟังก์ชันทำงานจะปิดกราฟที่เคยเปิดมาก่อน กำหนดตำแหน่งและสร้างกราฟจากข้อมูลกลุ่มของพีช

```
def countplant():
    plt.close()
    #กำหนด size หน้าจอ
    px = 1/plt.rcParams['figure.dpi']
    plt.subplots(figsize=(1200*px,800*px))
    #จัดเนื้อหาให้ดูเป็นสัดส่วนโดยจัดแบ่งช่องของกราฟ และ จำนวนกลุ่มของพืช
    countplantdf = df['กลุ่มของพืช'].value_counts().rename_axis('sub_value').reset_index(name='count_value')
    #สร้างตัวแปรเก็บข้อมูล countplantdf ในหัวขั้ว sub_value
    countplantlabel = countplantdf.sub_value
    #สร้างตัวแปรเก็บข้อมูล countplantdf ในหัวขั้ว count_value
    countplantcount = countplantdf.count_value
    #กำหนดให้มี row เดียว 2 column และจัดตำแหน่งที่ 1
    plt.subplot(1,2,1)
    #กำหนด title ของกราฟเป็น จำนวนกลุ่มของพืช
    plt.title('จำนวนกลุ่มของพืช')
    #สร้าง bar กราฟจาก dataframe
    countplant = plt.barh(countplantlabel, countplantcount, color= 'maroon')
    #นำ value ของ countplantcount ไปห้กับ bar กราฟ
    plt.bar_label(countplant, labels=countplantcount, fontsize=9)
    #กำหนดให้มี row เดียว 2 column และจัดตำแหน่งที่ 2
    plt.subplot(1,2,2)
    plt.title('ประเภทของกลุมของพืช')
```



## ฟังก์ชัน status1

เมื่อเลือกสถานะภาพจะทำการเรียกใช้งานฟังก์ชัน status1 เมื่อฟังก์ชันทำงานปิดกราฟที่เคยเปิดมาก่อน กำหนดตำแหน่งกราฟ และสร้างกราฟที่ได้มาจากข้อมูลสถานะภาพ

```
def status1():
    plt.close()
    #กำหนด size หน้าจอ
    px = 1/plt.rcParams['figure.dpi']
    plt.subplots(figsize=(1200*px,800*px))
    #จัดข้อมูลให้อยู่ในรูปที่มีแค่หัวชื่อของสถานะภาพ ก่อน ค.ศ. 1994
    status1beforedf = df['สถานะภาพ (ก่อน ค.ศ. 1994)'].value_counts().rename_axis('sub_value').reset_index(name='count_value')
    #สร้างตัวแปรมาเก็บข้อมูล status1beforedf ในหัวชื่อ sub_value
    status1dfbeforelabel = status1beforedf.sub_value
    #สร้างตัวแปรมาเก็บข้อมูล status1beforedf ในหัวชื่อ count_value
    status1dfbeforecount = status1beforedf.count_value
    #กำหนดให้มี row 2 2 column และอยู่ตำแหน่งที่ 1
    plt.subplot(2,2,1)
    plt.title('จำนวนสถานะภาพในช่วงก่อน ค.ศ. 1994')
    #ใช้ plot กราฟ bar ของ pandas
    statusbefore = plt.barh(status1dfbeforelabel, status1dfbeforecount, color= 'maroon')
    #นำ value ของ status1dfbeforecount ไปพิมพ์ bar กราฟ
    plt.bar_label(statusbefore,labels=status1dfbeforecount,fontsize=9)
    #เรียกข้อมูลจาก status1dfbeforecount และใส่เข้าไปใน valueforpie2
    valueforpie2 = status1dfbeforecount
```

## ฟังก์ชัน shape

เมื่อเลือกลักษณะนิสัยจะทำการเรียกฟังก์ชัน shape เมื่อฟังก์ชันทำงานปิดกราฟที่เคยเปิดมาก่อน กำหนดตำแหน่งกราฟและสร้างกราฟที่ได้มาจากข้อมูลลักษณะนิสัย

```
def shape():
    plt.close()
    #กำหนด size หน้าจอ
    px = 1/plt.rcParams['figure.dpi']
    plt.subplots(figsize=(1200*px,800*px))
    shapedf = df['ลักษณะนิสัย'].value_counts().rename_axis('sub_value').reset_index(name='count_value')
    shapedflabel = shapedf.sub_value
    shapedfcount = shapedf.count_value
    #กำหนดให้มี row 2 row 2 column และอยู่ตำแหน่งที่ 1
    plt.subplot(2,2,1)
    plt.title("จำนวนลักษณะนิสัย")
    plt.xticks(rotation = 90,fontsize=9)
    barplot = plt.bar(shapedflabel, shapedfcount,color= 'blue')
    plt.bar_label(barplot,labels=shapedfcount,fontsize=6)

    for index, value in enumerate(shapedfcount):
        value= (value/len(df))*100
        valueforpie4.append(float(f"{value:.1f}"))
    #กำหนดให้มี row 2 row 2 column และอยู่ตำแหน่งที่ 2
    plt.subplot(2,2,2)
    #title ลักษณะนิสัย
    plt.title('ลักษณะนิสัย')
    #pie ใช้ค่าจาก forpie2 ซึ่งเก็บค่ามาจากลักษณะนิสัย
    plt.pie(shapedfcount)
    #ใช้ตัวแปร labels เก็บ หัวชื่อ และ value
    labels = [f'{l}', {s:0.1f}% for l, s in zip(shapedflabel,valueforpie4)]
    #plot legend ออกมาด้วย ค่าจาก pie ที่ 2 และ labels คือ หัวชื่อ และ value ที่เก็บมา
    plt.legend(shapedfcount, labels=labels,prop={'size': 6},bbox_to_anchor=(1,1))
```

ฟังก์ชัน wong

เมื่อกดตกลง(2) เลือกวงศ์พีชที่ต้องการจะทำการเรียกฟังก์ชัน wong เมื่อฟังก์ชันทำงานปิดกราฟที่เคยเปิดมาก่อน กำหนดตำแหน่งกราฟและสร้างกราฟที่ได้มาจากข้อมูลของวงศ์พีชและสถานภาพก่อนปี ค.ศ. 1994 และ สถานภาพในปี 1994 - 2001

```
def wong():
    plt.close()
    new_dataframe = sub.column(4,8,9)
    df1 = dfforfind.iloc[:, [4,8,9]]
    find_get = sub4.get()
    pievalue = []
    pie_label = []
    if find_get:
        #สร้าง dataframe ขึ้นใหม่
        rowname = df1[df1['id'].str.contains(find_get)]
        #สร้างใหม่ไว้ส่วนที่ทำการหาใน column สถานภาพ (ในช่วง ค.ศ. 1994-2001)
        vu = rowname[rowname['สถานภาพ (ในช่วง ค.ศ. 1994-2001)'] == 'ไม่อยู่ในโศกฏพิณ'].count().reset_index(name='count_value')
        vublabelcount = list(vu.count_value)
        vu2 = rowname[rowname['สถานภาพ (ในช่วง ค.ศ. 1994-2001)'] == 'สูญพันธุ์ในธรรมชาติ'].count().reset_index(name='count_value')
        vublabelcount2 = list(vu2.count_value)
        vu3 = rowname[rowname['สถานภาพ (ในช่วง ค.ศ. 1994-2001)'] == 'ใกล้สูญพันธุ์อย่างสิ้นเชิง'].count().reset_index(name='count_value')
        vublabelcount3 = list(vu3.count_value)
        vu4 = rowname[rowname['สถานภาพ (ในช่วง ค.ศ. 1994-2001)'] == 'ใกล้สูญพันธุ์'].count().reset_index(name='count_value')
        vublabelcount4 = list(vu4.count_value)
        vu6 = rowname[rowname['สถานภาพ (ในช่วง ค.ศ. 1994-2001)'] == 'ใกล้สูญหาย'].count().reset_index(name='count_value')
        vublabelcount6 = list(vu6.count_value)
        vu7 = rowname[rowname['สถานภาพ (ในช่วง ค.ศ. 1994-2001)'] == 'มีถิ่นที่อยู่น้อยที่สุด'].count().reset_index(name='count_value')
        vublabelcount7 = list(vu7.count_value)
        vu8 = rowname[rowname['สถานภาพ (ค.ศ. 1994)'] == 'หายากทั่วโลก'].count().reset_index(name='count_value')
        vublabelcount8 = list(vu8.count_value)
        vu9 = rowname[rowname['สถานภาพ (ค.ศ. 1994)'] == 'หายากในประเทศไทย'].count().reset_index(name='count_value')
```

ฟังก์ชัน find\_plant\_TH

เมื่อกดตกลง(3) เลือกชื่อไทยและทำการ Show ข้อมูลรายละเอียดจากชื่อที่ทำการเลือกไว้ โดย message box

ฟังก์ชัน find\_plant\_EN

เมื่อกดตกลง(4) เลือกชื่อวิทยาศาสตร์และทำการ Show ข้อมูลรายละเอียดจากชื่อที่ทำการเลือกไว้ โดย message box

```
def find_plant_EN():
    f = ""
    #รับค่ามาจาก sub2 หรือชื่อวิทยาศาสตร์
    find_get = sub2.get()
    if find_get:
        rowname = dfforfind[dfforfind['ชื่อวิทยาศาสตร์'].str.contains(find_get)]
        for key,value in rowname.iterrows():
            f += str(value)
            messagebox.showinfo("DATA", f)
    else:
        messagebox.showinfo("ALERT", "กรุณาป้อนข้อมูล")
def find_plant_TH():
    f = ""
    find_get = sub3.get()
    if find_get:
        rowname = df[df['ชื่อไทย'].str.contains(find_get)]
        for key,value in rowname.iterrows():
            f += str(value)
            messagebox.showinfo("DATA", f)
    else:
        messagebox.showinfo("ALERT", "กรุณาป้อนข้อมูล")
```

## รูปแบบการวิเคราะห์ข้อมูล

กราฟประเภทของพืช จะทำการนับประเภทของพืชที่เหมือนกันใน column และ plot ออกมาเป็นกราฟโดยใช้ชื่อและจำนวนที่นับเอาไว้

โดยข้อมูลสรุปได้ว่ากลุ่มประเภทกลุ่มพืชดอก(ใบเลี้ยงคู่) มีจำนวนการถูกคุกคามเยอะที่สุด กลุ่มพืชดอก(ใบเลี้ยงเดี่ยว) มีการถูกคุกคามเป็นอันดับ 2 กลุ่มเฟินมีการถูกคุกคามเป็นอันดับ 3 และอันดับสุดท้ายเป็นกลุ่มพืชเมล็ดเปลือย

กราฟสถานภาพ จะทำการนับสถานภาพที่เหมือนกันจาก column สถานภาพ ก่อนปีค.ศ.1994 และ ปี ค.ศ.1994 – 2001 จากนั้นนับมา plot กราฟ 4 กราฟเปรียบเทียบกับ เนื่องจาก ก่อน ปี ค.ศ. 1994 มีการบันทึกไม่เหมือนกับปี ค.ศ. 1994 – 2001

โดยข้อมูลสรุปได้ว่าสถานภาพก่อนปี ค.ศ. 1994 มีพืชที่หายกระดับโลกถูกคุกคามเยอะที่สุด ไม่ระบุเป็นอันดับที่ 2 และหายาก(ประเทศไทย) อยู่อันดับสุดท้าย ส่วนสถานภาพในปี ค.ศ.1994 – 2001 ไม่ระบุเป็นอันดับที่ 1 มีแนวโน้มใกล้สูญพันธุ์เป็นอันดับที่ 2 ใกล้สูญพันธุ์เป็นอันดับที่ 3 ใกล้ถูกคุกคามเป็นอันดับที่ 4 ใกล้สูญพันธุ์อย่างยิ่งเป็นอันดับที่ 5 เป็นกังวลน้อยที่สุดเป็นอันดับที่ 6 หายกระดับโลก เกณฑ์ปี 1994 เป็นอันดับที่ 7 สูญพันธุ์โดยธรรมชาติเป็นอันดับสุดท้าย

กราฟลักษณะวิสัย จะทำการนับลักษณะวิสัยที่เหมือนกันใน column ลักษณะวิสัย จากนั้นนับมา plot กราฟ

โดยข้อมูลสรุปได้ว่า ไม้ล้มลุกถูกคุกคามมากที่สุดจากสถานภาพปี ค.ศ. 1994 – 2001

กราฟวงศ์ จะทำการนับวงศ์ที่เหมือนกันใน column วงศ์ และรวมกับ column สถานภาพ ก่อนปี ค.ศ. 1994 และ สถานภาพ ปี ค.ศ. 1994 - 2001