

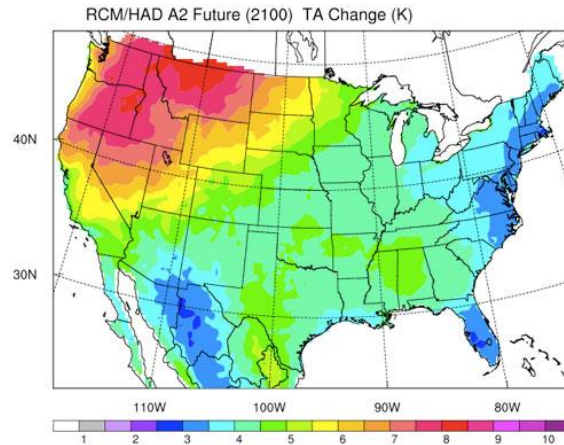
Climate Change Project

นายเชษฐากร จิรเพิ่มพูนทรัพย์ 5801012630041

นายสุเมธ บุญญา 5801012630181

Climate Change คืออะไร??

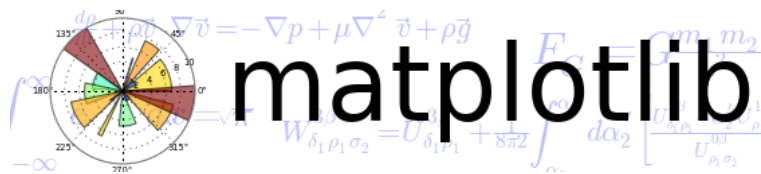
Climate Change คือโปรแกรมสำหรับตรวจสอบข้อมูลสภาพอากาศเช่น อุณหภูมิ, ปริมาณความชื้นในอากาศ และทำการคาดการณ์ข้อมูลในอนาคตว่าจะมีสภาพเป็นเช่นไร



ที่มาและความสำคัญของ Project

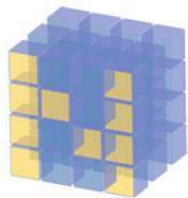
เนื่องจากในปัจจุบัน คนเราต้องออกเดินทางไปทำงาน หรือทำกิจกรรมนอกสถานที่เป็นประจำ ซึ่งบางสถานการณ์นั้นเราก็ไม่สามารถหลีกเลี่ยงได้ เราจึงจำเป็นต้องเตรียมพร้อมเสมอโดยการตรวจสอบข้อมูลของสภาพอากาศเพื่อเตรียมการป้องกันล่วงหน้า หรือเพื่อเป็นข้อมูลประกอบการตัดสินใจในการทำงานหรือการเดินทางต่างๆ จึงเป็นที่มาของ Weather Forecast หรือการพยากรณ์สภาพอากาศ และได้มีการพัฒนามาอย่างต่อเนื่อง กลุ่มของเราจึงได้ทำการออกแบบ Climate Change Program ขึ้นเพื่อทดลองการทำงานของ การแสดงข้อมูลของสภาพอากาศในแต่ละช่วงเวลาเพื่อตรวจสอบ และหาแนวโน้มของข้อมูลสภาพอากาศในอนาคต

องค์ประกอบของโปรแกรม



matplotlib

{JSON}

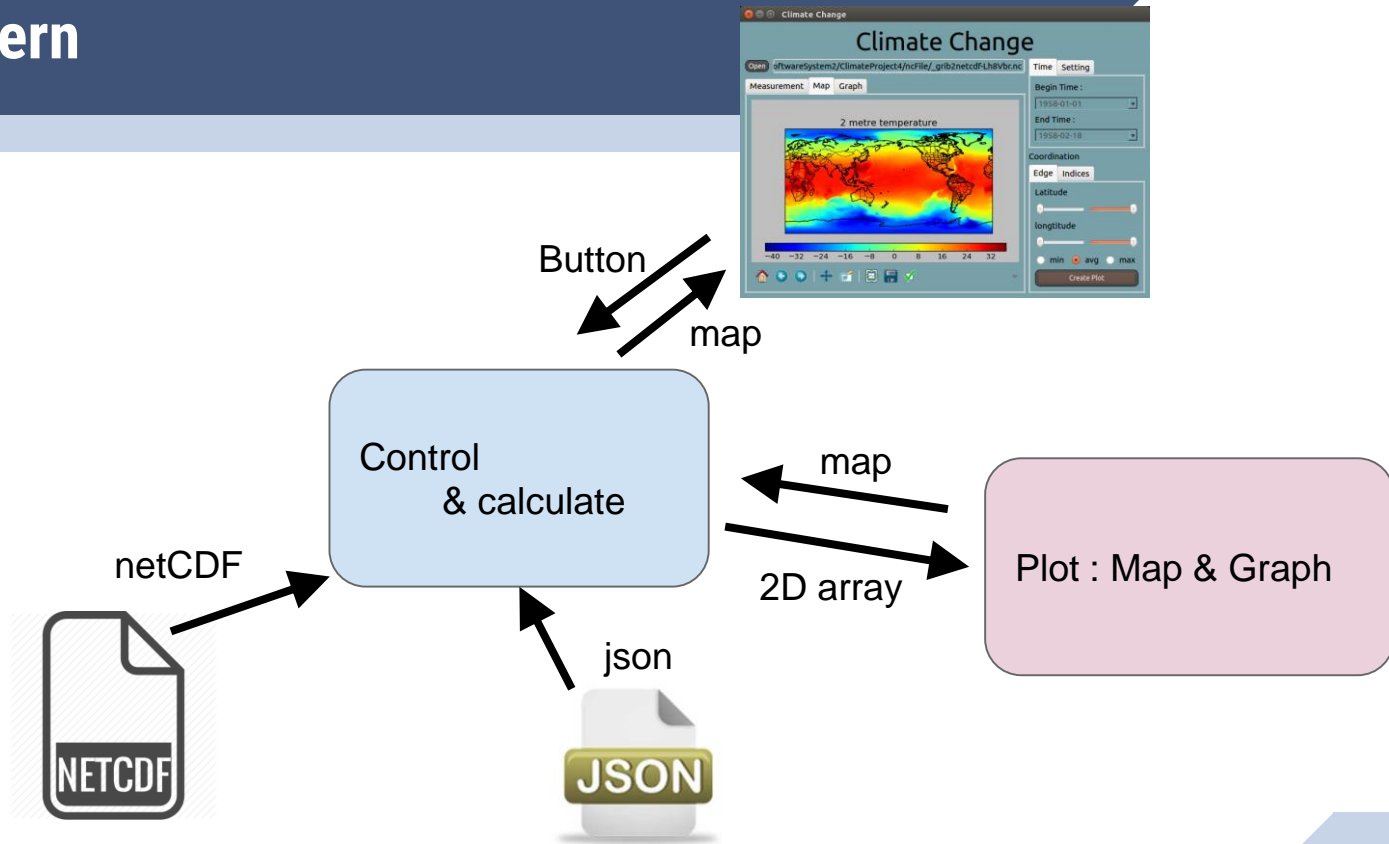


NumPy



Pattern

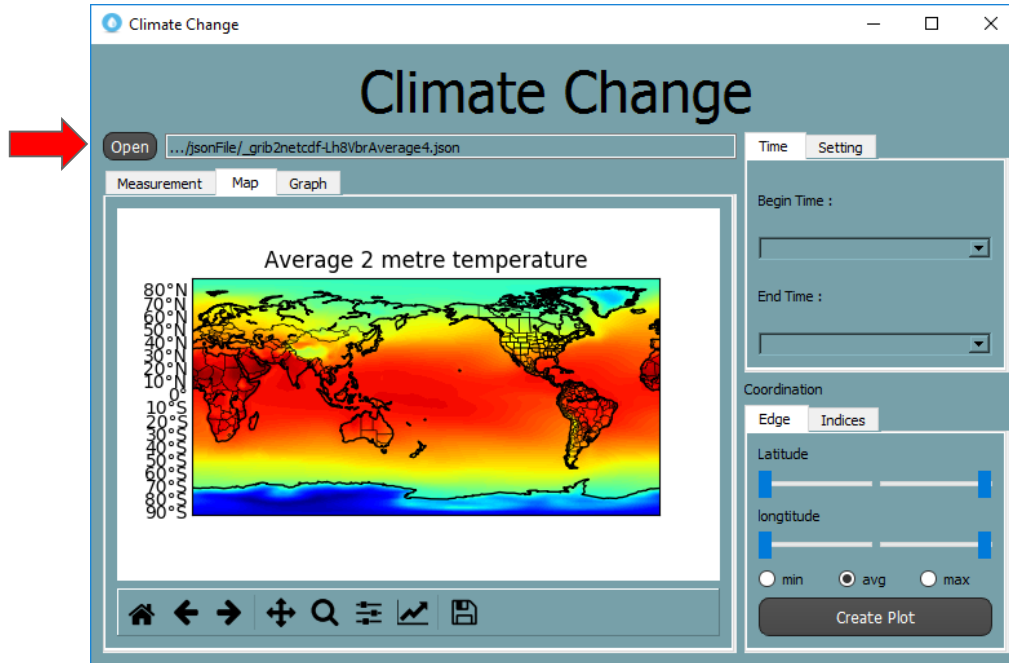
GUI



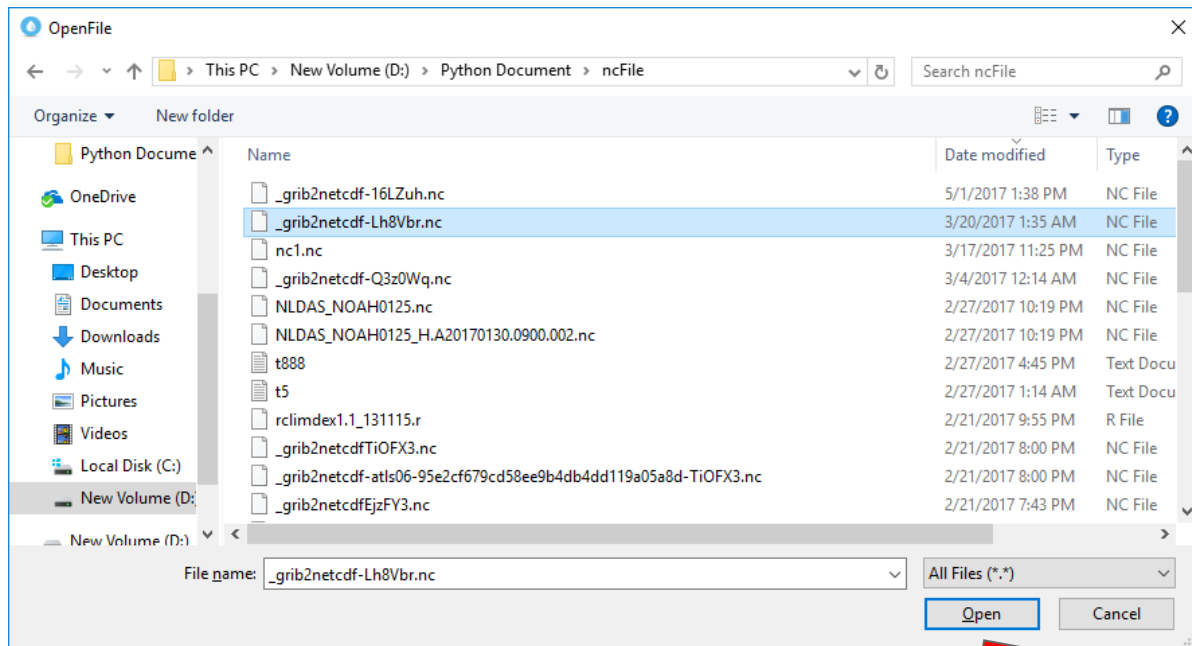
การทำงานเบื้องต้นของโปรแกรม

- สามารถเลือก netCDF file ที่เราต้องการ plot ได้
- สามารถเลือกค่า Measurement ที่เราต้องการ plot ได้
- สามารถเลือก time series หรือช่วงเวลาในการ plot ได้
- สามารถเลือกความละเอียดหรือตำแหน่งที่ต้องการ plot ได้
- สามารถเลือก Indices ในการ plot map และ graph ได้
- สามารถอ่านข้อมูลจาก Json file มา plot ได้

การเลือกไฟล์ netCDF สำหรับใช้ plot



เมื่อเปิดโปรแกรมมา ข้อมูลจะถูก set default ไว้เพื่อโชว์ข้อมูลจากไฟล์ json ชั้นแรกให้คลิกที่ open เพื่อเปิดหน้าจอสำหรับเลือกไฟล์



เลือกไฟล์ที่ต้องการ ในที่นี่จะ
เลือกเป็น

_grib2netcdf-Lh8Vbr.nc แล้ว
จากนั้นคลิก Open

Climate Change

Open D:/Python Document/ndFile/_grib2netcdf-Lh8Vbr.nc

Measurement Map Graph

	Name
1	2 metre temperature
2	Total column water vapour

Time Setting

Begin Time : 1958-01-01

End Time : 1958-01-01

Coordination

Edge Indices

Latitude

longitude

☐ min ☒ avg ☐ max

Create Plot

จะเข้าสู่หน้าจอการเลือก
Measurement และ
ส่วนประกอบอื่นๆของการ
plot

การเลือก Measurement ที่ต้องการ plot

Time - ใช้สำหรับเลือก
ช่วงเวลาในการ plot

Setting - ใช้เลือกการ
แสดงข้อมูลบน map เช่น
ความละเอียด และ
ลักษณะของการ plot

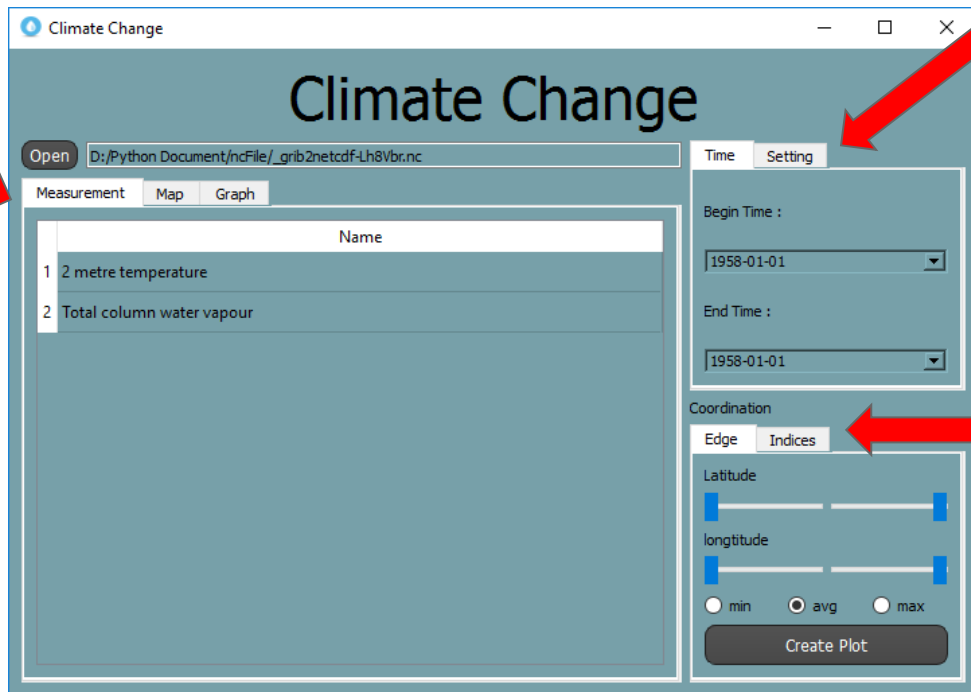
Edge - ใช้สำหรับเลือก
พิกัดในการ plot รวมไปถึง
ถึงค่า Max Min Avg

Indices - ใช้แสดงข้อมูล
Indices ทั้งในรูปแบบ
ของ map และ graph

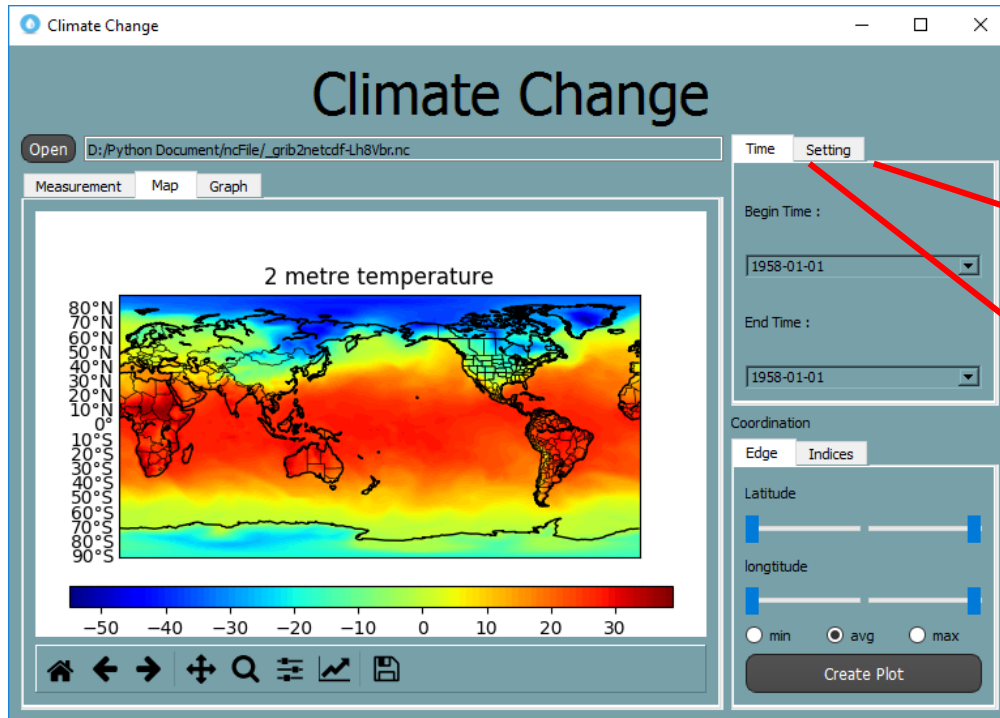
Measurement - ใช้
เลือกตัวแปรสำหรับ plot

Map - ใช้แสดงแผนที่ที่
ทำการ plot แล้ว

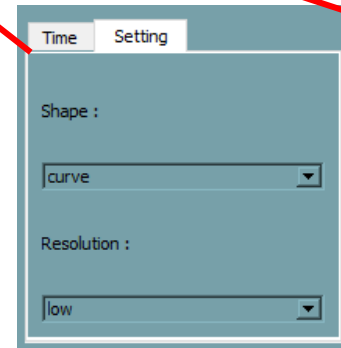
Graph - ใช้แสดงค่า
Index ที่เก็บไว้ในไฟล์
นำมาแสดงเป็น Graph

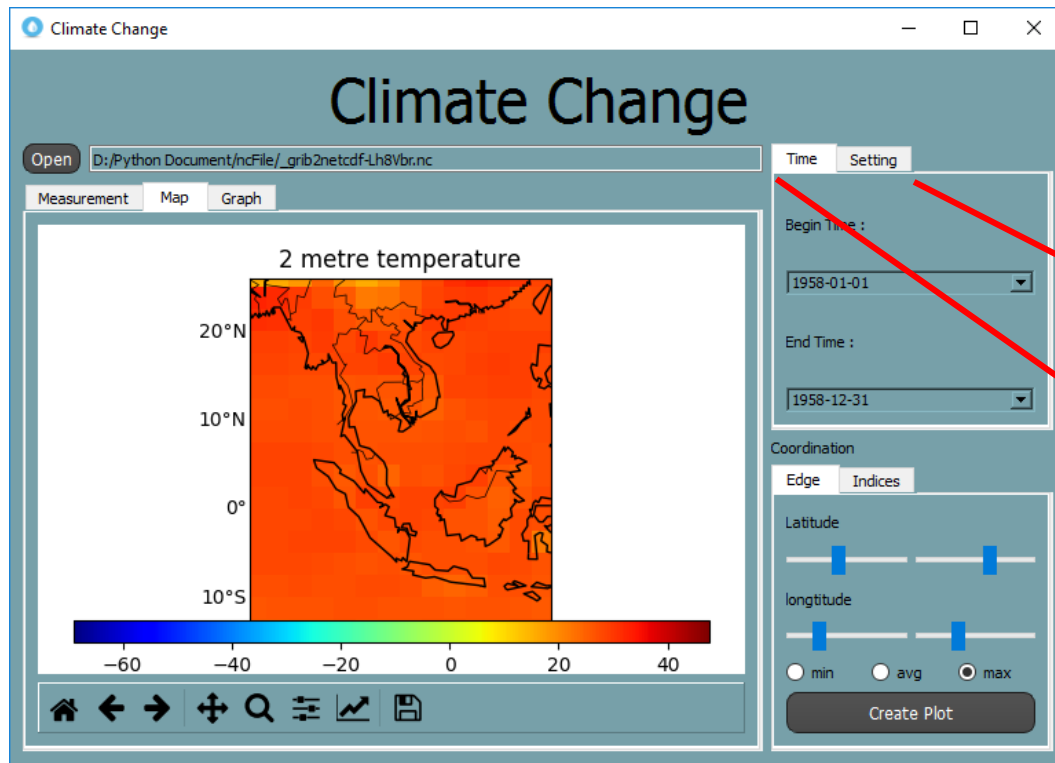


การเลือกองค์ประกอบเพิ่มเติมสำหรับการ plot

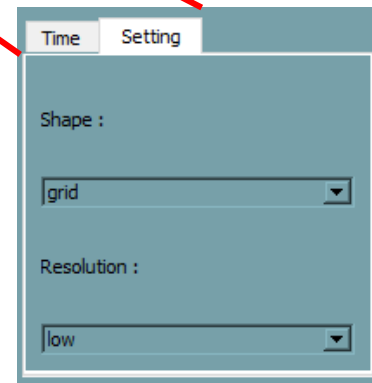


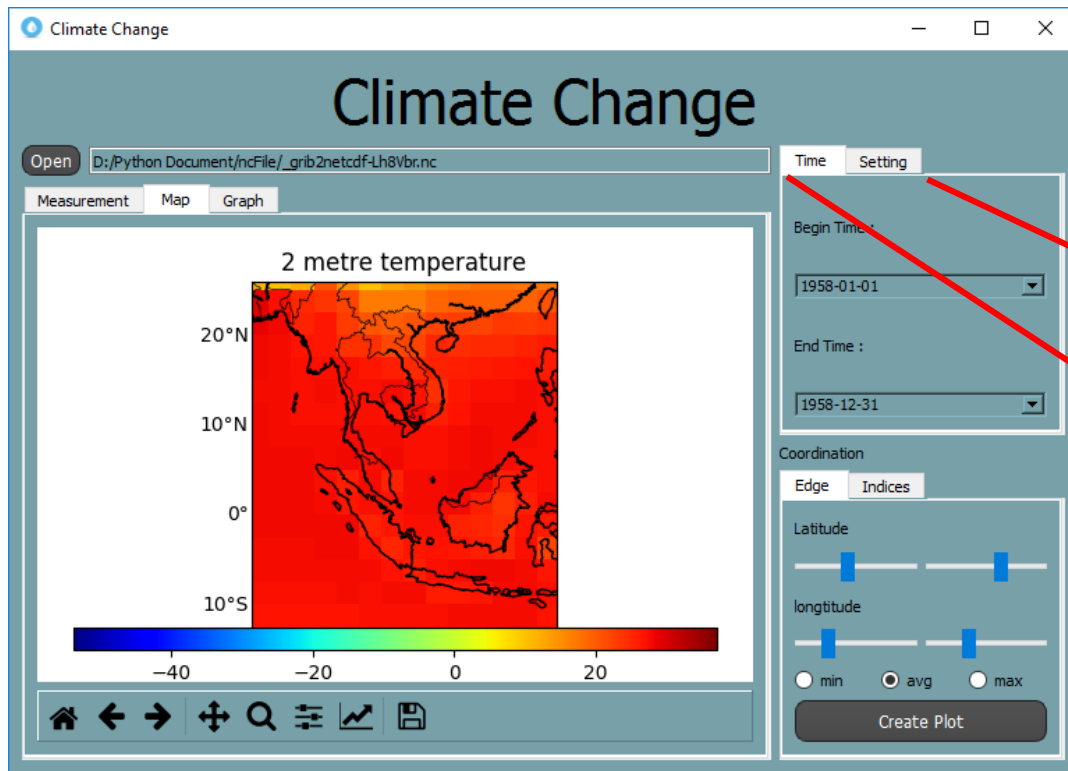
การ Plot map แบบ
ปกติ ซึ่งไม่ได้เลือก
ช่วงเวลาและ Option
เพิ่มเติม โดยลักษณะการ
Plot จะถูกตั้งไว้ที่ Curve



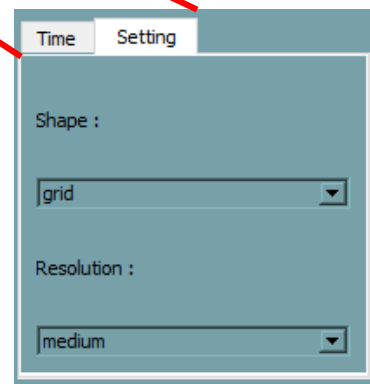


การ Plot map แบบ
เลือกตำแหน่ง latitude
และ longitude เอง รวมไปถึง
ถึง กำหนดค่า plot เป็น
max เลือกช่วงการแสดงผล
เป็น 1 ปี ลักษณะการ
plot เป็นแบบ Grid



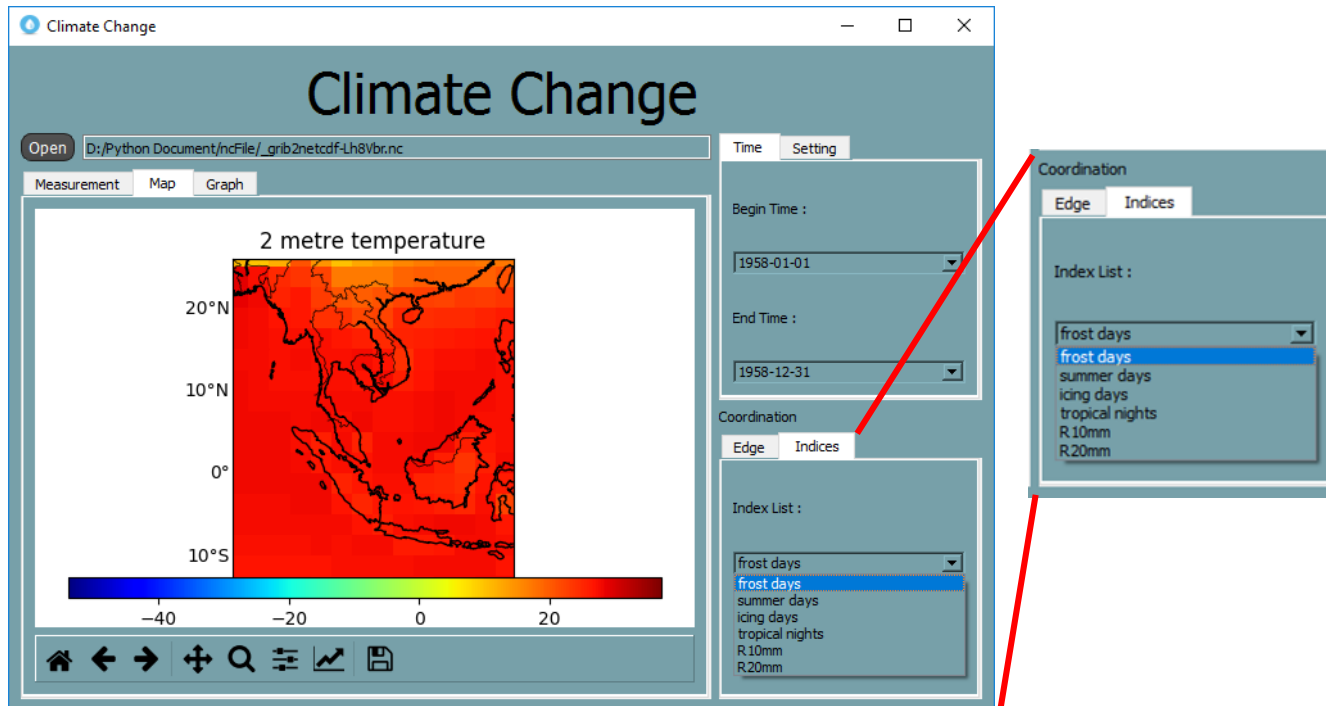


การ Plot map แบบ
เลือกตำแหน่ง latitude
และ longitude เอง รวมไปถึง
ถึง กำหนดค่า plot เป็น
avg เลือกช่วงการแสดง
เป็น 1 ปี ลักษณะการ
plot เป็นแบบ Grid ความ
ละเอียด medium



การเลือก plot indice ต่างๆจากไฟล์ Json

ในส่วนของ Indice นั้นจะทำการอ่านค่าตาม key ที่กำหนดไว้ใน Json เพื่อนำมาใช้ Plot ในรูปแบบที่เราต้องการ

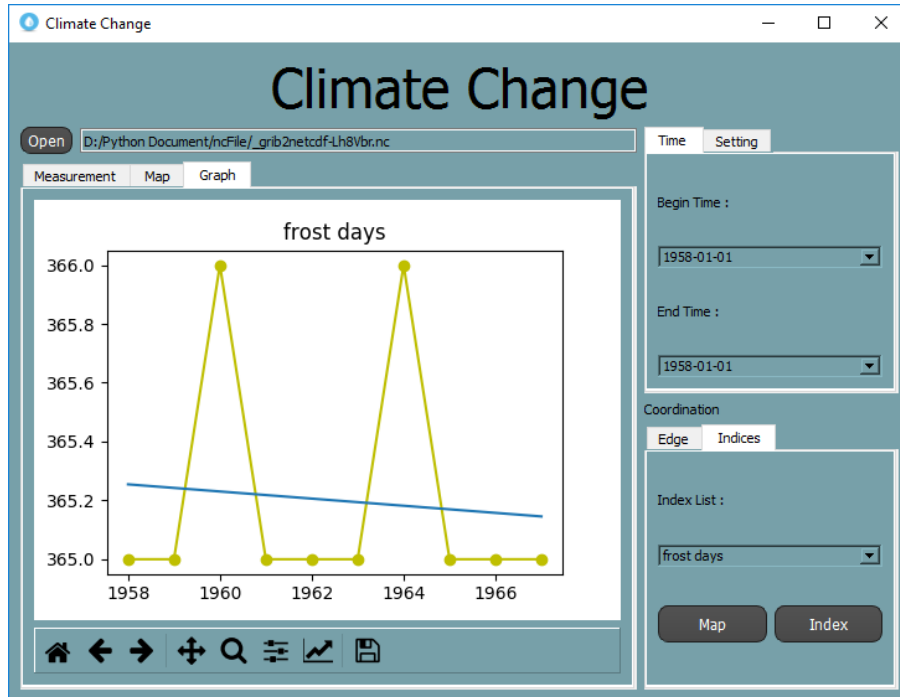


การเลือก plot indice ต่างๆจากไฟล์ Json

ในส่วนของ Indices ที่สามารถเลือกมาแสดงนั้นมี 6 ตัว ดังนี้

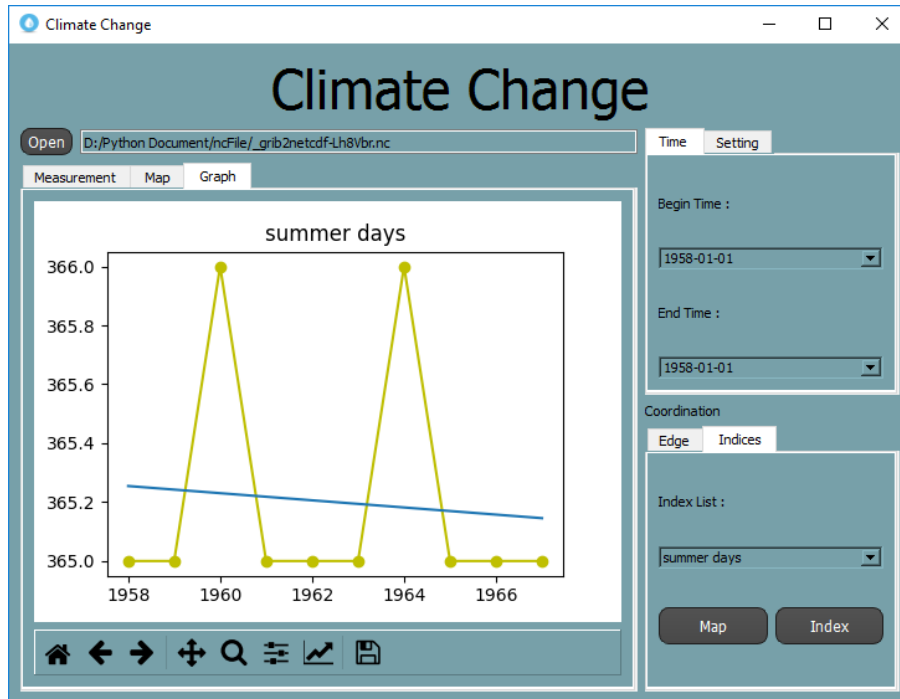
1. Frost Days จำนวนวันในแต่ละปีที่มีค่าอุณหภูมิต่ำสุดต่อวันน้อยกว่า 0 องศาเซลเซียส
2. Summer Days จำนวนวันในแต่ละปีที่มีค่าอุณหภูมิสูงสุดต่อวันมากกว่า 25 องศาเซลเซียส
3. Icing Days จำนวนวันในแต่ละปีที่มีค่าอุณหภูมิสูงสุดต่อวันน้อยกว่า 0 องศาเซลเซียส
4. Tropical Nights จำนวนวันในแต่ละปีที่มีค่าอุณหภูมิต่ำสุดต่อวันมากกว่า 20 องศาเซลเซียส
5. R10mm จำนวนวันที่มีปริมาณของหยาดน้ำฟ้าต่อวันตั้งแต่ 10 mm ขึ้นไป
6. R20mm จำนวนวันที่มีปริมาณของหยาดน้ำฟ้าต่อวันตั้งแต่ 20 mm ขึ้นไป

การ plot indices แบบ Graph



เลือกการ plot Indices ของ Frost Days ในรูปแบบของ Graph โดยในที่นี้เป็นข้อมูลแบบ Annual ที่แสดง ซึ่งมีจำนวนสูงสุดคือ 10 ปี แบ่งกราฟเป็น 2 เส้นคือ เส้นของข้อมูล และเส้นของ Trend ซึ่งเป็นเส้นที่ใช้บอกแนวโน้มของข้อมูลว่า ในอนาคตค่าที่ได้จะสูงขึ้นหรือลดลง

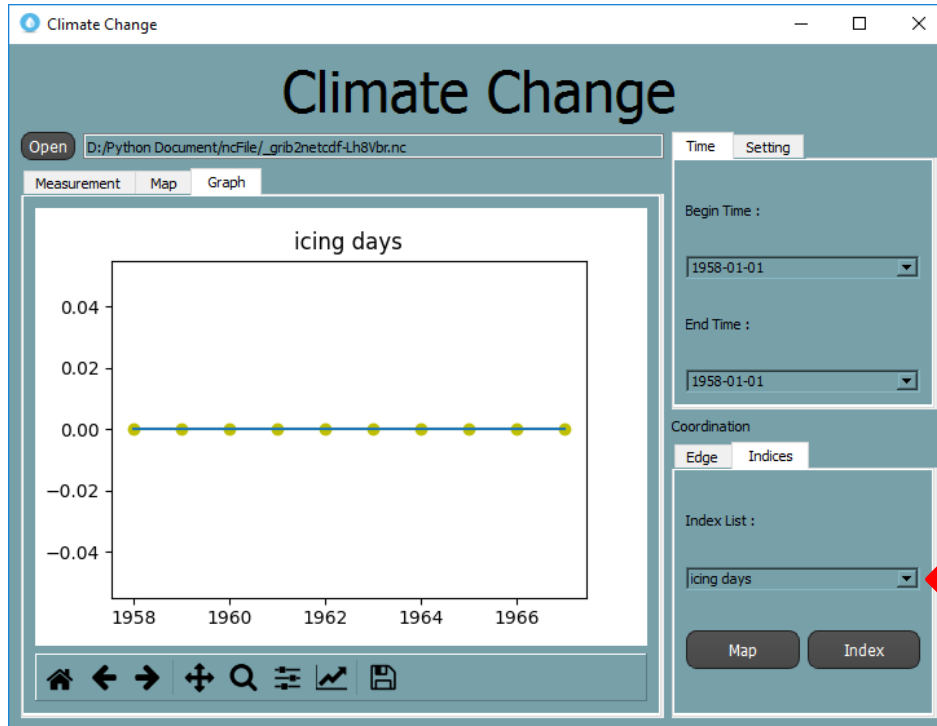
การ plot indices แบบ Graph



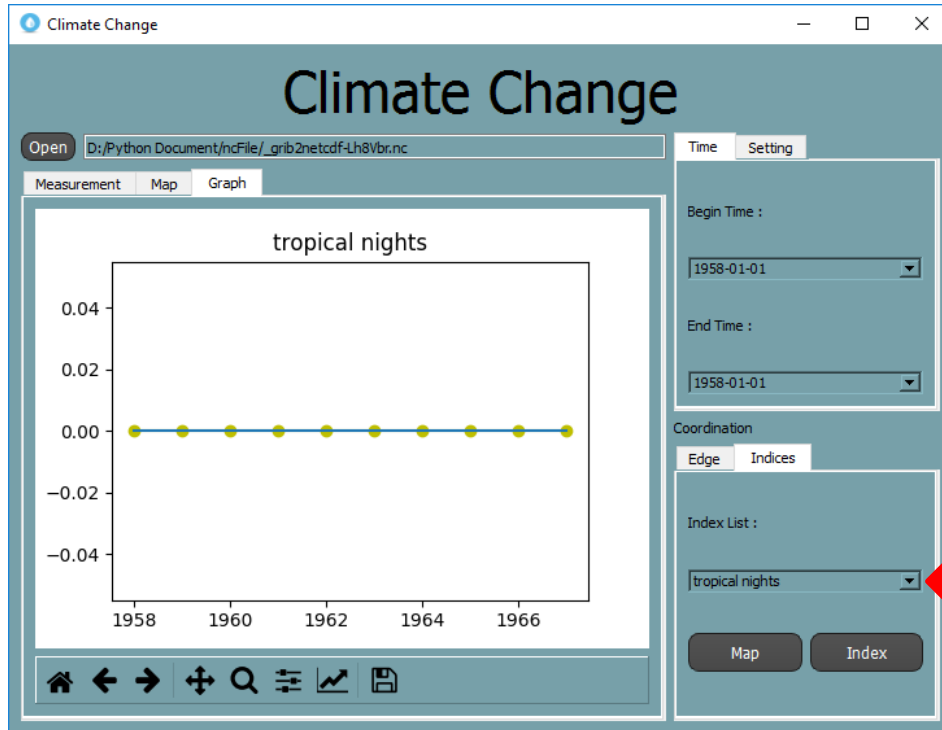
เลือกการ plot Indices ของ Summer Days ในรูปแบบของ Graph โดยในที่นี้เป็นข้อมูลแบบ Annual ที่แสดง ซึ่งมีจำนวนสูงสุดคือ 10 ปี แบ่งกราฟเป็น 2 เส้นคือ เส้นของข้อมูล และเส้นของ Trend ซึ่งเป็นเส้นที่ใช้ออกแนวโน้มของข้อมูลว่า ในอนาคตค่าที่ได้จะสูงขึ้นหรือลดลง

การ plot indices แบบ Graph

เลือกการ plot Indices ของ Icing Days ในรูปแบบของ Graph โดยในที่นี้เป็นข้อมูลแบบ Annual ที่แสดง ซึ่งมีจำนวนสูงสุดคือ 10 ปี แบ่งกราฟเป็น 2 เส้นคือ เส้นของข้อมูล และเส้นของ Trend ซึ่งเป็นเส้นที่ใช้บอกแนวโน้มของข้อมูลว่า ในอนาคตค่าที่ได้จะสูงขึ้นหรือลดลง

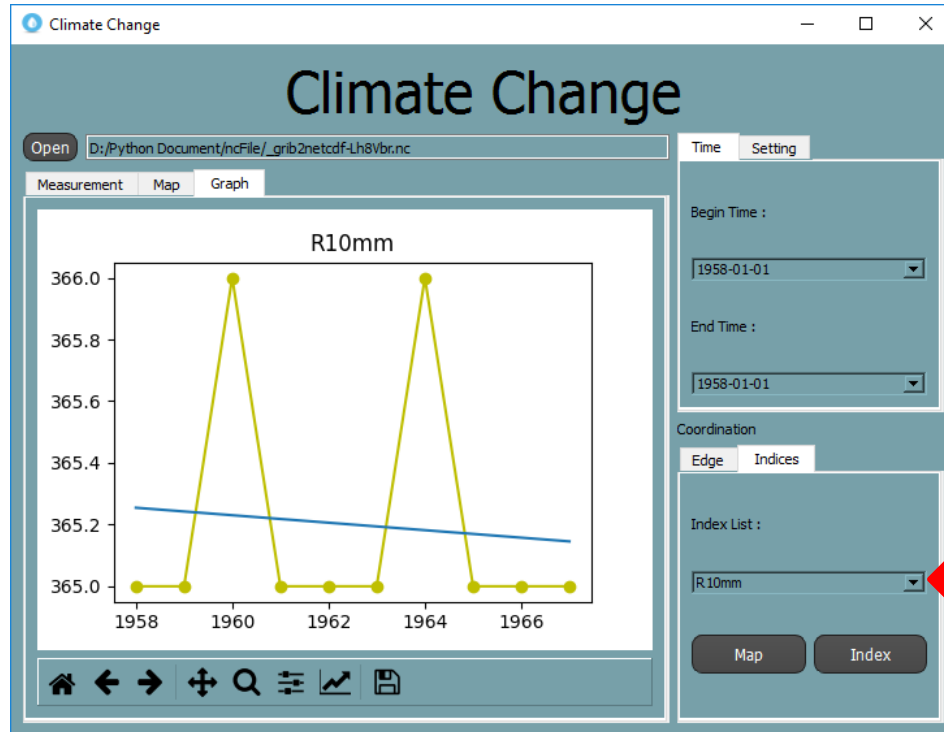


การ plot indices แบบ Graph



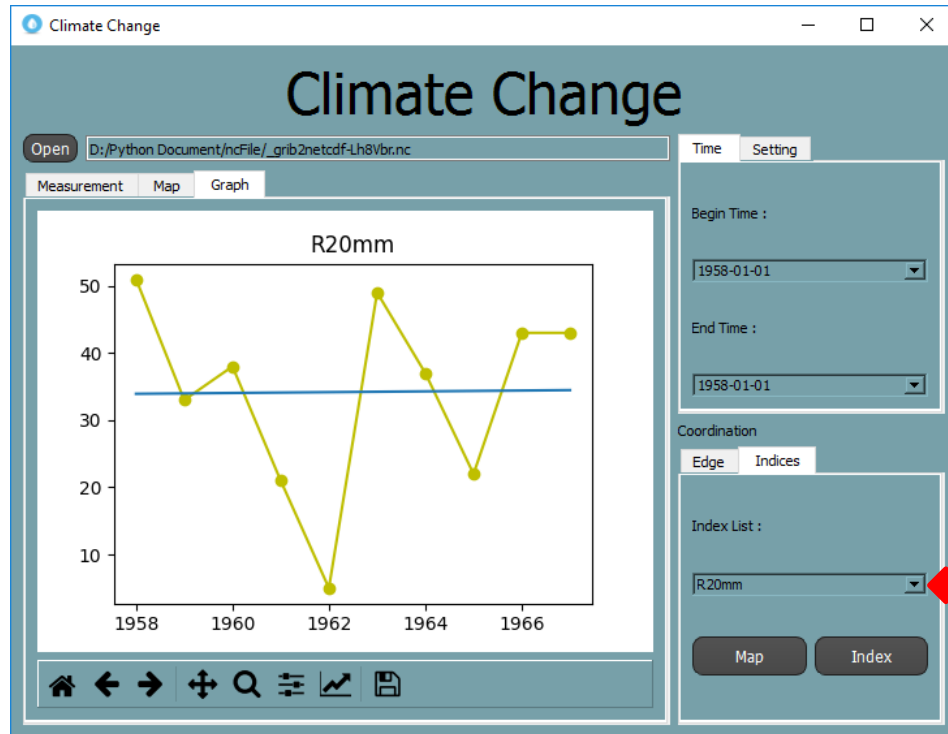
เลือกการ plot Indices ของ Tropical Nights ในรูปแบบของ Graph โดยในที่นี้เป็นข้อมูลแบบ Annual ที่แสดง ซึ่งมีจำนวนสูงสุดคือ 10 ปี แบ่งกราฟเป็น 2 เส้นคือ เส้นของข้อมูล และเส้นของ Trend ซึ่งเป็นเส้นที่ใช้บอกแนวโน้มของข้อมูลว่า ในอนาคตค่าที่ได้จะสูงขึ้นหรือลดลง

การ plot indices แบบ Graph



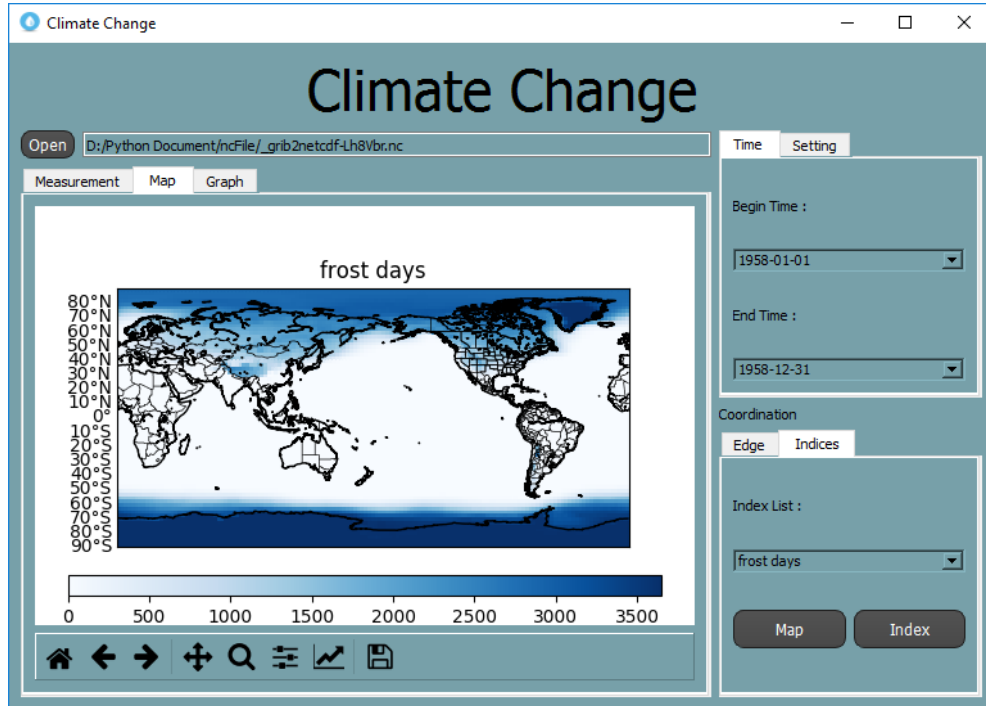
เลือกการ plot Indices ของ R10mm ในรูปแบบของ Graph โดยในที่นี้เป็นข้อมูลแบบ Annual ที่แสดง ซึ่งมีจำนวนสูงสุดคือ 10 ปี แบ่งกราฟเป็น 2 เส้นคือ เส้นของข้อมูล และเส้นของ Trend ซึ่งเป็นเส้นที่ใช้บอกแนวโน้มของข้อมูลว่า ในอนาคตค่าที่ได้จะสูงขึ้นหรือลดลง

การ plot indices แบบ Graph



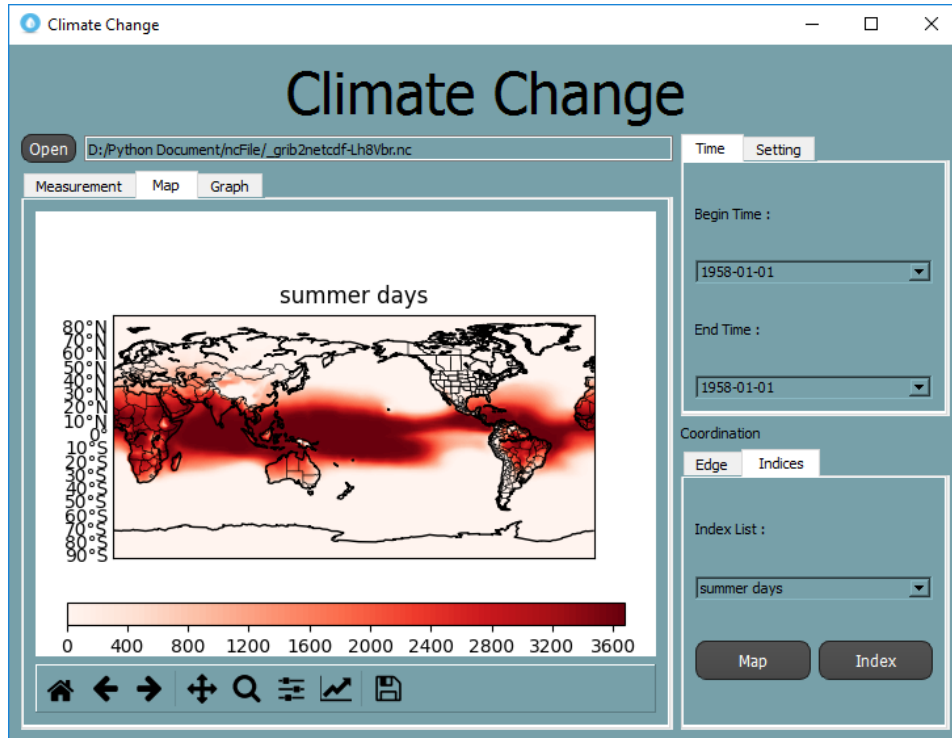
เลือกการ plot Indices ของ R20mm ในรูปแบบของ Graph โดยในที่นี้เป็นข้อมูลแบบ Annual ที่แสดง ซึ่งมีจำนวนสูงสุดคือ 10 ปี แบ่งกราฟเป็น 2 เส้นคือ เส้นของข้อมูล และเส้นของ Trend ซึ่งเป็นเส้นที่ใช้บอกแนวโน้มของข้อมูลว่า ในอนาคตค่าที่ได้จะสูงขึ้นหรือลดลง

การ plot indices แบบ Map



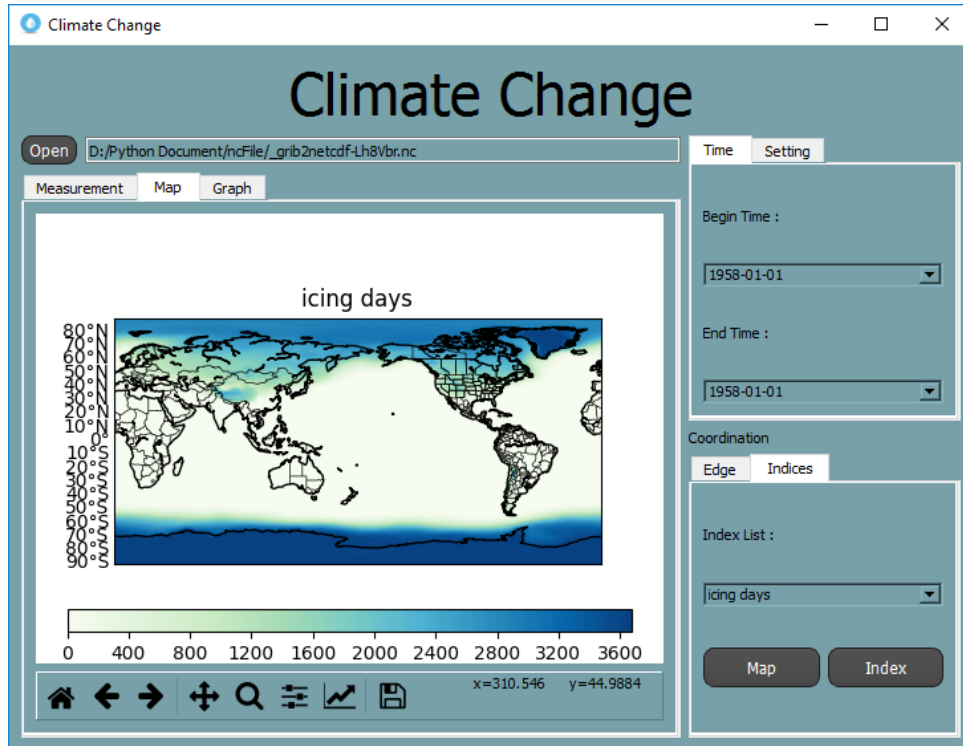
เลือกการ plot Indices
ของ Frost Days ในรูปแบบ
ของ Map โดยในที่นี้เป็น
ข้อมูลแบบ Daily ที่แสดง ซึ่ง
มีจำนวนสูงสุดคือ 3652 วัน

การ plot indices แบบ Map



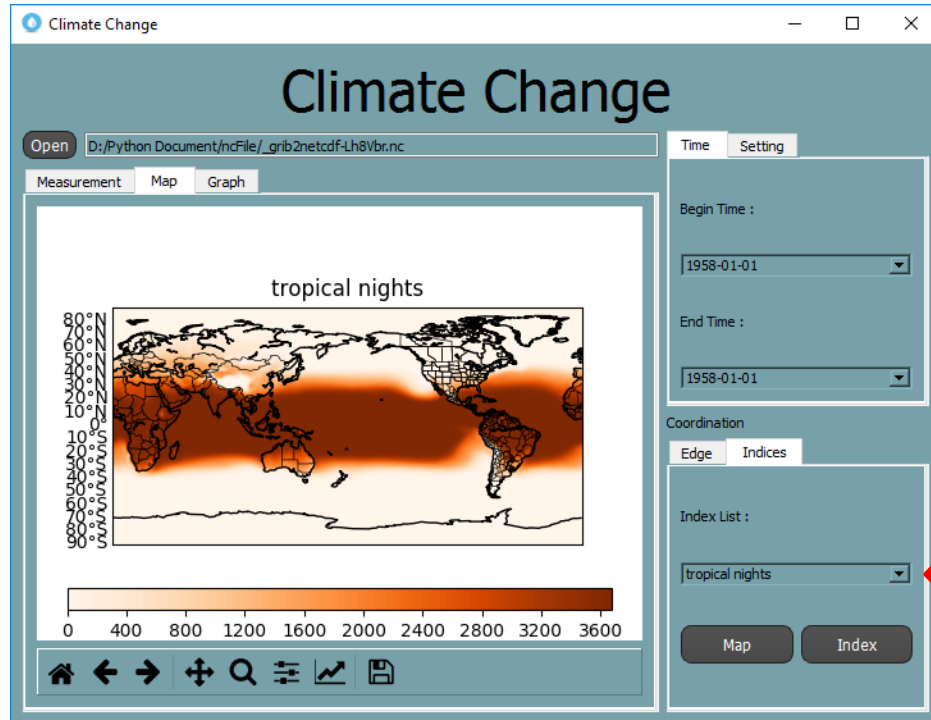
เลือกการ plot Indices
ของ Summer Days ใน
รูปแบบของ Map โดยในที่นี้
เป็นข้อมูลแบบ Daily ที่
แสดง ซึ่งมีจำนวนสูงสุดคือ
3652 วัน

การ plot indices แบบ Map



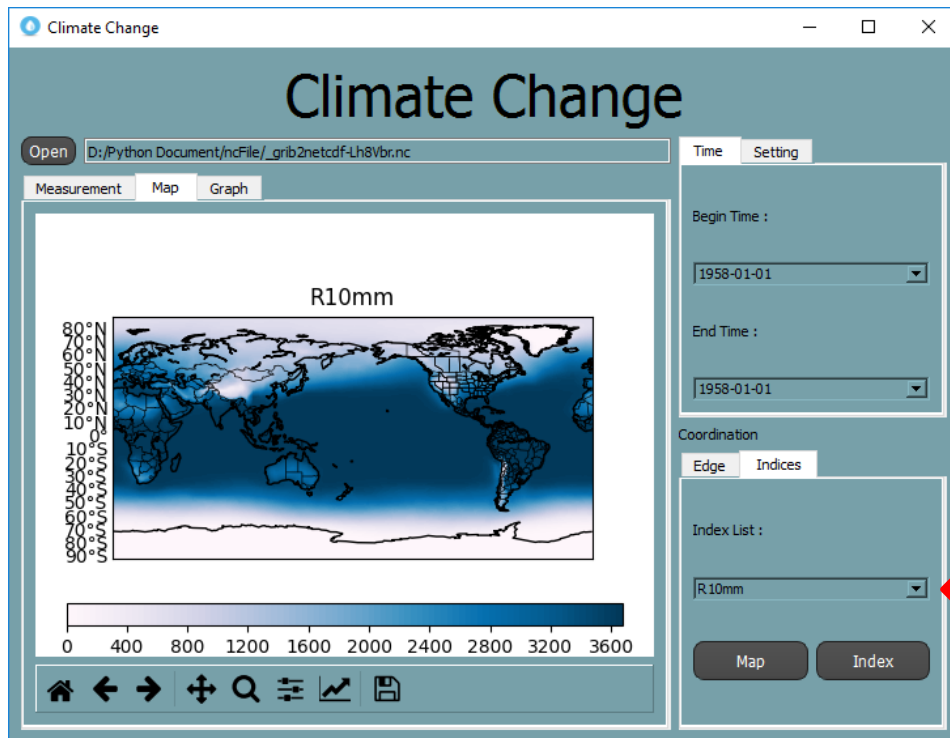
เลือกการ plot Indices
ของ Icing Days ในรูปแบบ
ของ Map โดยในที่นี้เป็น
ข้อมูลแบบ Daily ที่แสดง ซึ่ง
มีจำนวนสูงสุดคือ 3652 วัน

การ plot indices แบบ Map



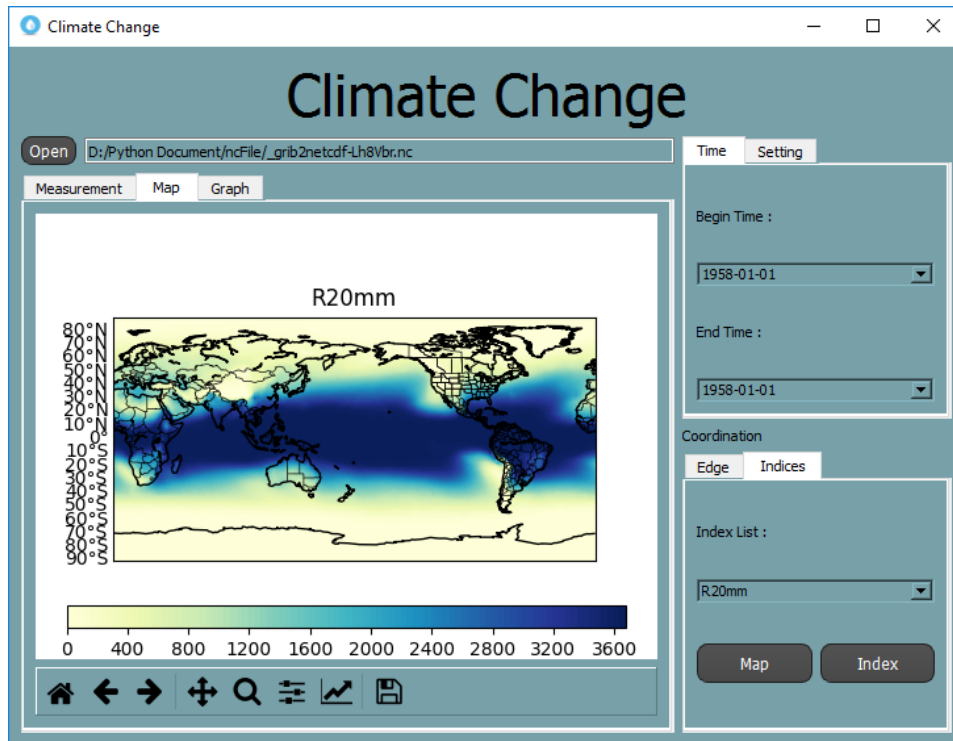
เลือกการ plot Indices
ของ Tropical Nights ใน
รูปแบบของ Map โดยในที่นี้
เป็นข้อมูลแบบ Daily ที่
แสดง ซึ่งมีจำนวนสูงสุดคือ
3652 วัน

การ plot indices แบบ Map



เลือกการ plot Indices
ของ R10mm ในรูปแบบ
ของ Map โดยในที่นี้เป็น
ข้อมูลแบบ Daily ที่แสดง ซึ่ง
มีจำนวนสูงสุดคือ 3652 วัน

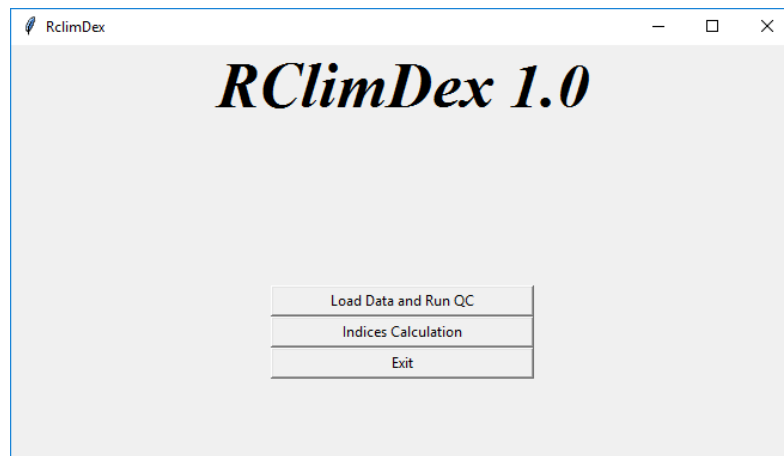
การ plot indices แบบ Map



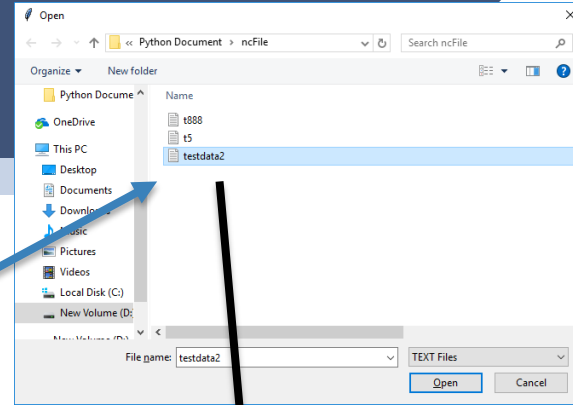
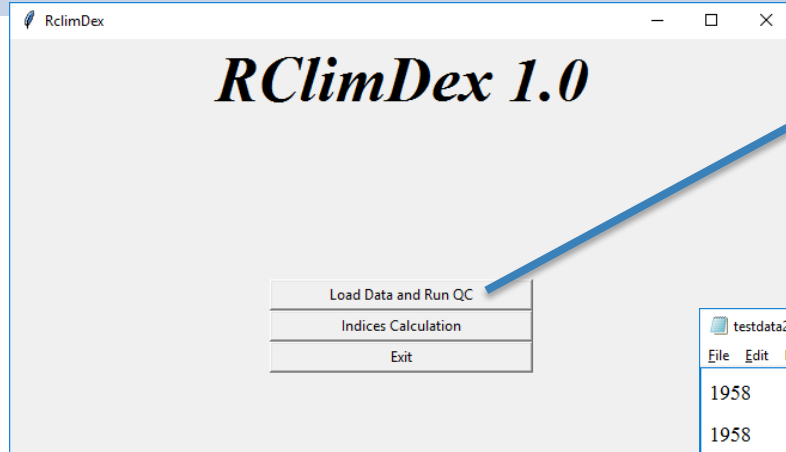
เลือกการ plot Indices
ของ R20mm ในรูปแบบ
ของ Map โดยในที่นี้เป็น
ข้อมูลแบบ Daily ที่แสดง ซึ่ง
มีจำนวนสูงสุดคือ 3652 วัน

การเปรียบเทียบข้อมูล

- RClimDex เป็น tool ที่ใช้ในการหา Indices



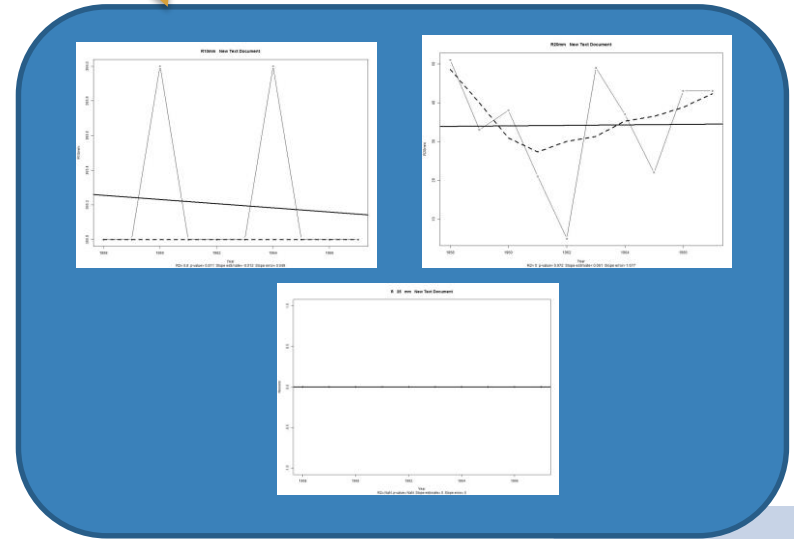
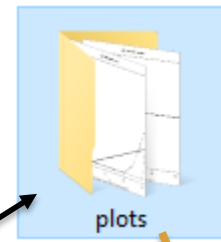
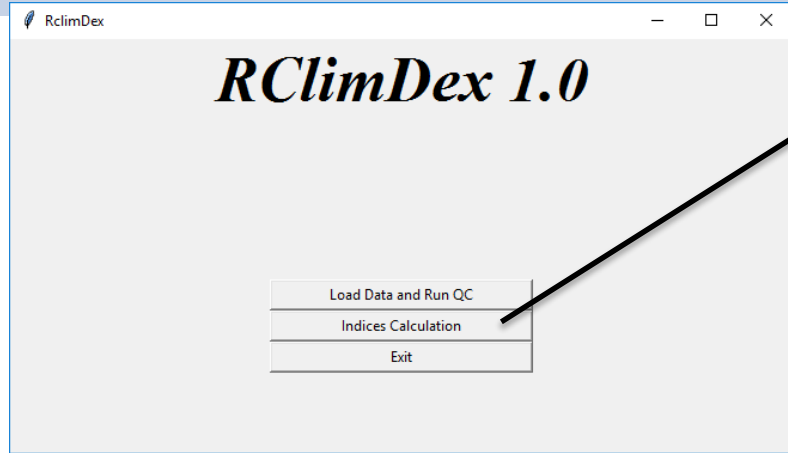
RClimDex



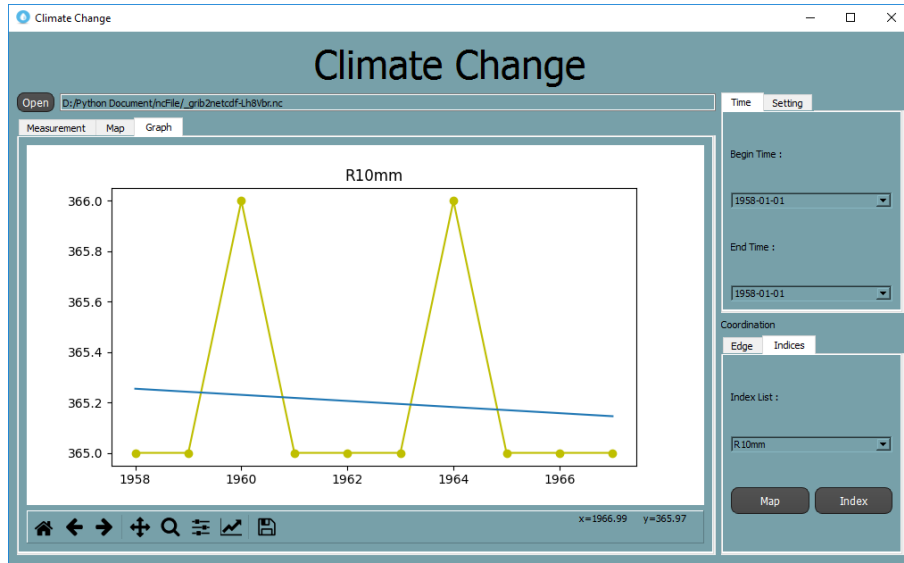
The screenshot shows a Notepad window titled "testdata2 - Notepad". The menu bar includes "File", "Edit", "Format", "View", and "Help". The text content is as follows:

1958	1	1	16.86045782	-54.0799	38.5058
1958	1	2	16.81187754	-49.6956	39.0627
1958	1	3	16.81633837	-51.8037	39.7371
1958	1	4	16.81336899	-47.7029	38.6273
1958	1	5	16.90952067	-47.1298	37.7018
1958	1	6	16.98437561	-47.0144	39.2612
1958	1	7	17.08352444	-48.7053	39.6378

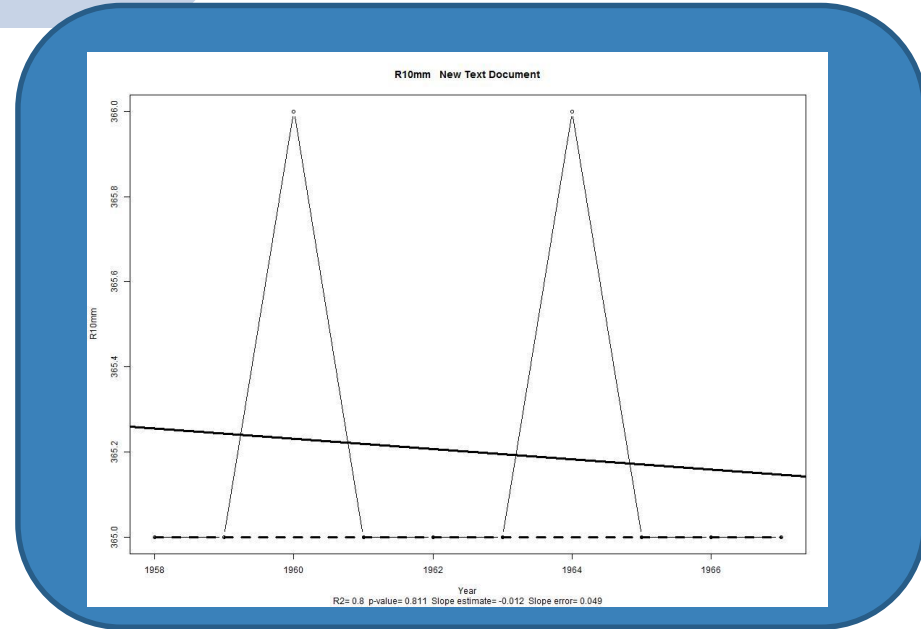
RClimDex



R10mm

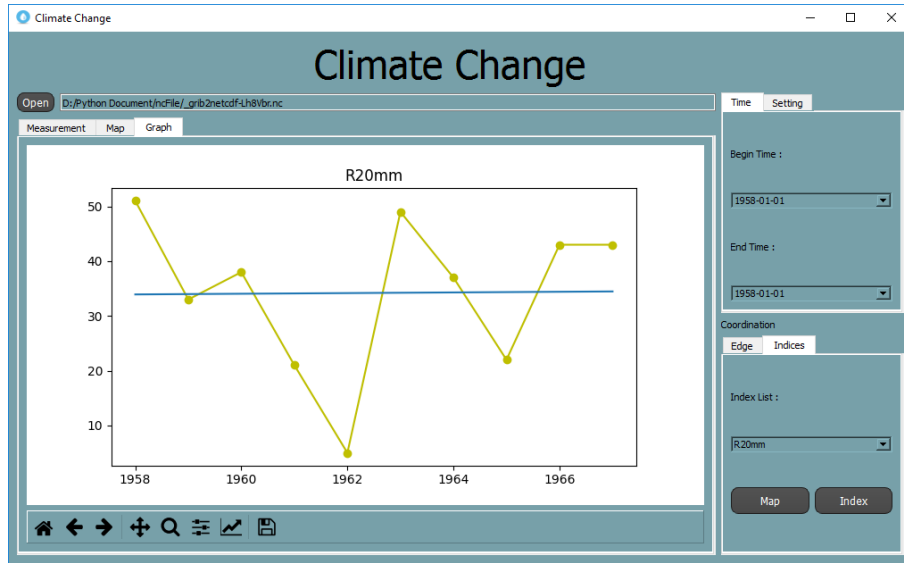


■ Climate Change App

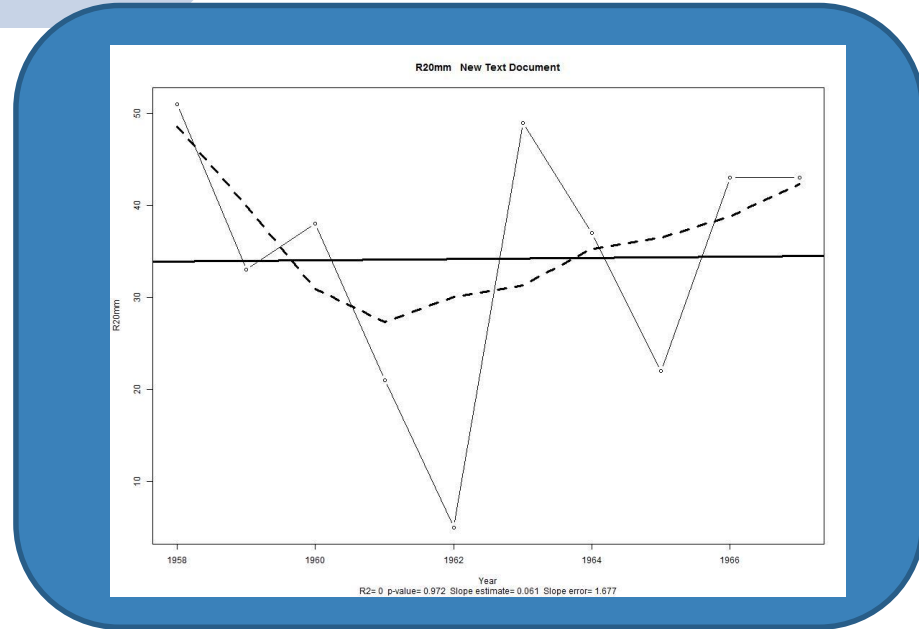


■ RClimDex

R20mm



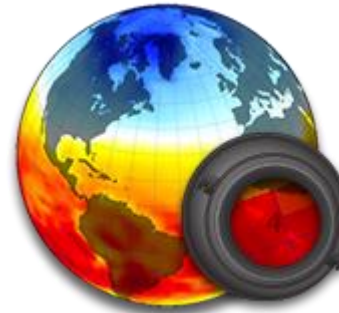
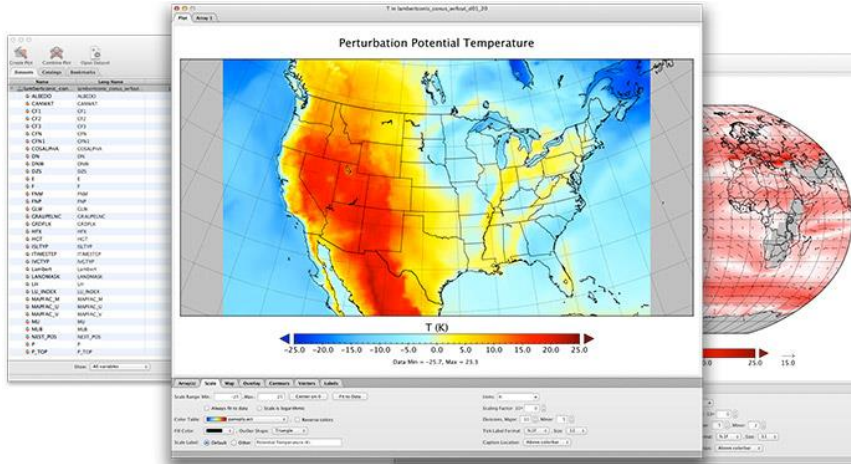
Climate Change App



RClimDex

Iceclim & Panoply

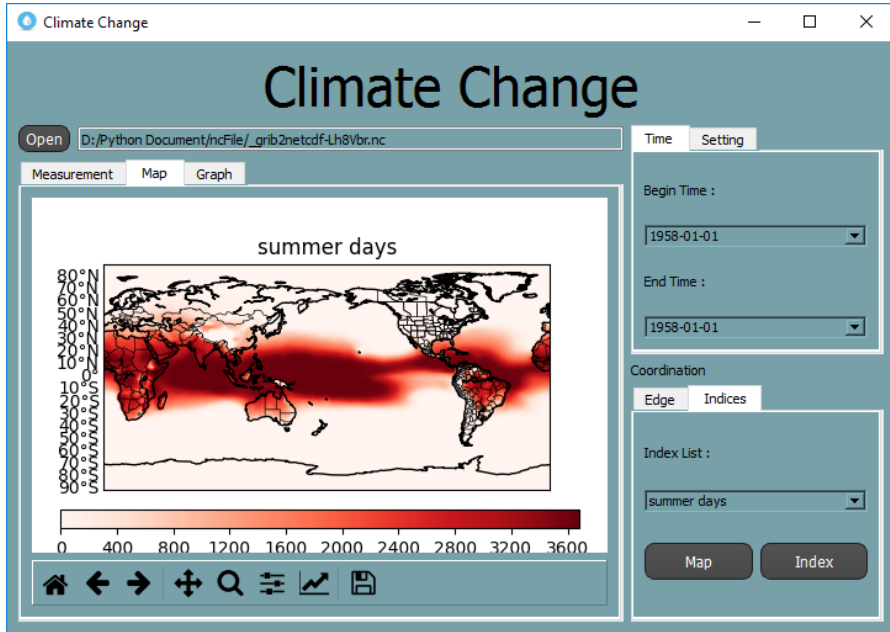
- lcclim -> 'filename.nc'
- 'filename.nc' -> Panoply



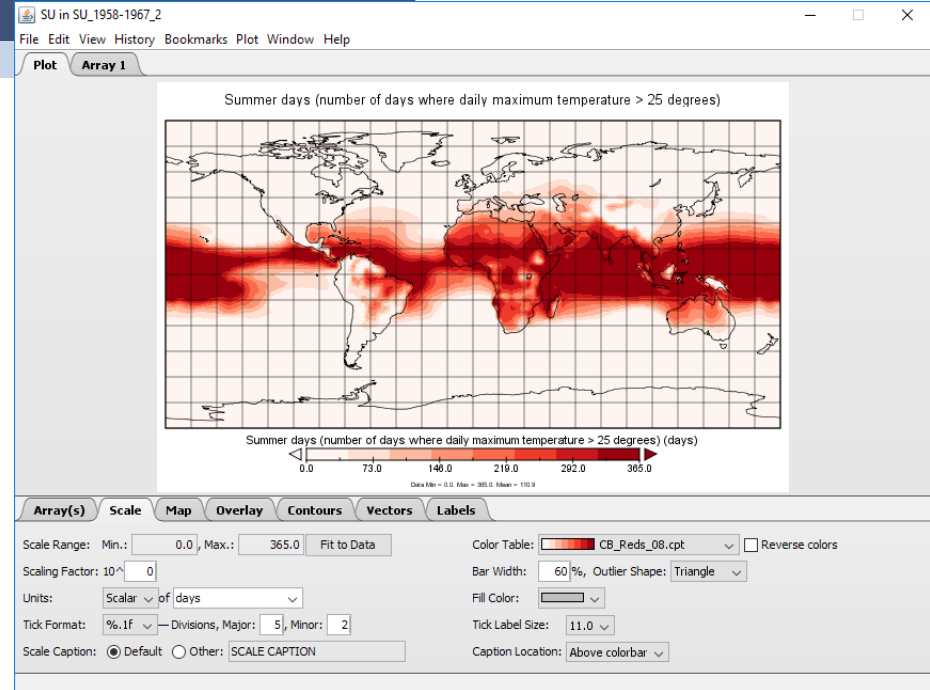
https://www.giss.nasa.gov/tools/panoply/panoply_400_icon.png

https://www.giss.nasa.gov/tools/panoply/panoply_400.jpg

Icclim & Panoply



■ Climate Change App



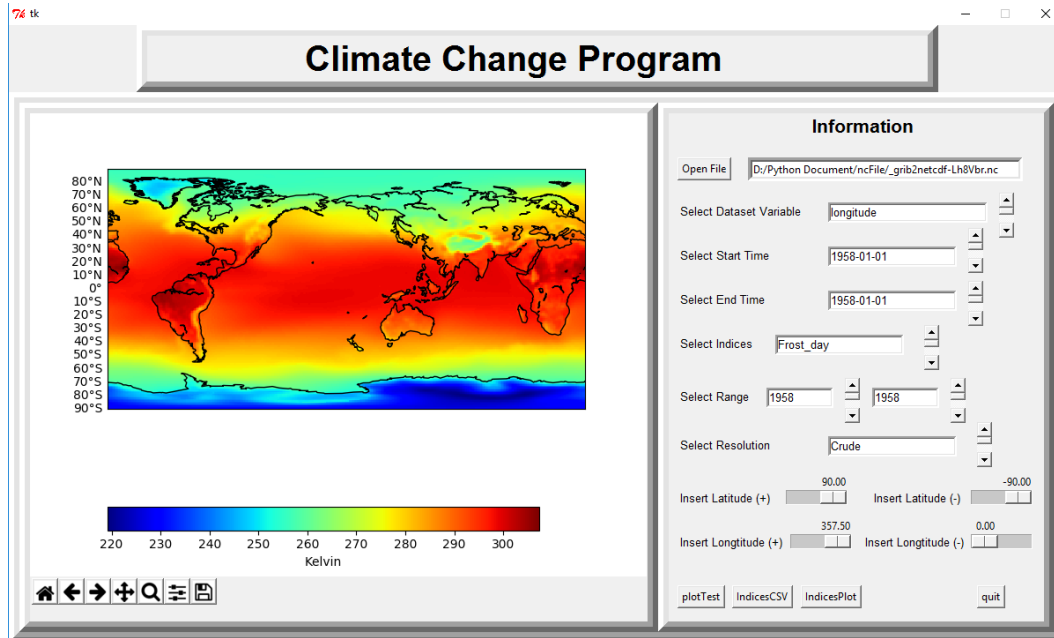
■ Panoply

ปัญหาต่างๆที่พบในการทำงานและการแก้ไข

ในการพัฒนาโปรแกรมส่วนแรกนั้นได้ทดลองใช้ Tkinter ในการออกแบบ GUI แต่เนื่องจาก Tkinter มีข้อจำกัดในการทำงานที่มากทำให้เกิดปัญหาต่อการพัฒนาดังนี้

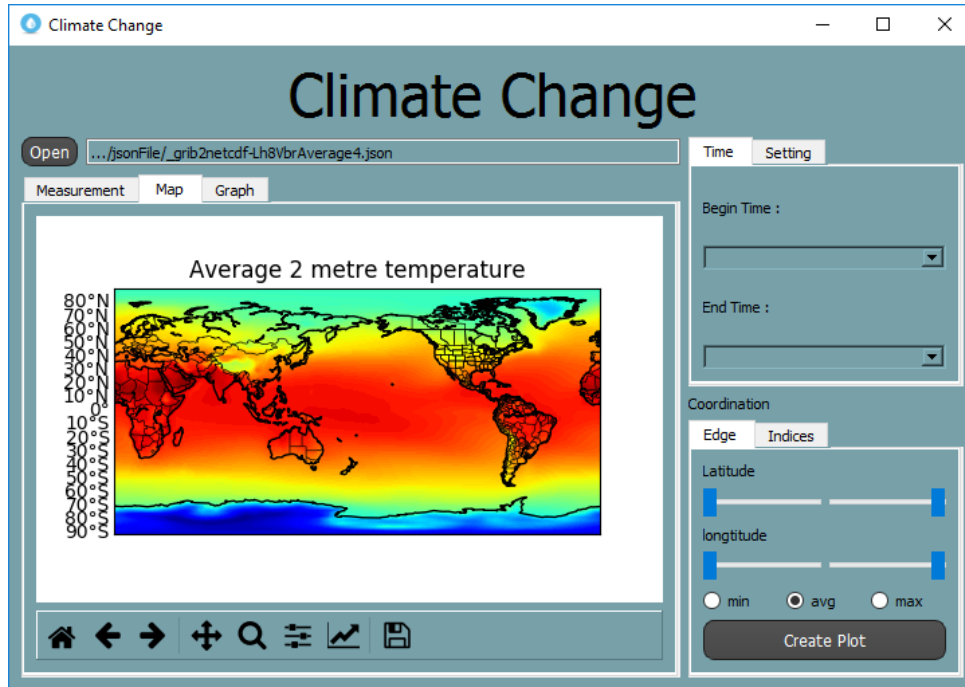
- การสร้าง Frame สำหรับออกแบบ GUI นั้นซับซ้อนและแก้ไขได้ยาก
- การปรับขนาดหน้าจอหรือ Resizable นั้นทำได้ยากเนื่องจากการวางตำแหน่งของ Frame นั้นต้องกำหนดตามแกน x,y ทำให้ตัว GUI มีรูปร่างผิดเพี้ยนไปเมื่อทำการปรับขนาด จึงต้อง fix window size
- การเพิ่มเติม Widget ลงไปในนั้นทำได้ค่อนข้างยากเนื่องจากการ fix size ทำให้ขอบเขตของจอแคบลง

ปัญหาต่างๆที่พบในการทำงานและการแก้ไข



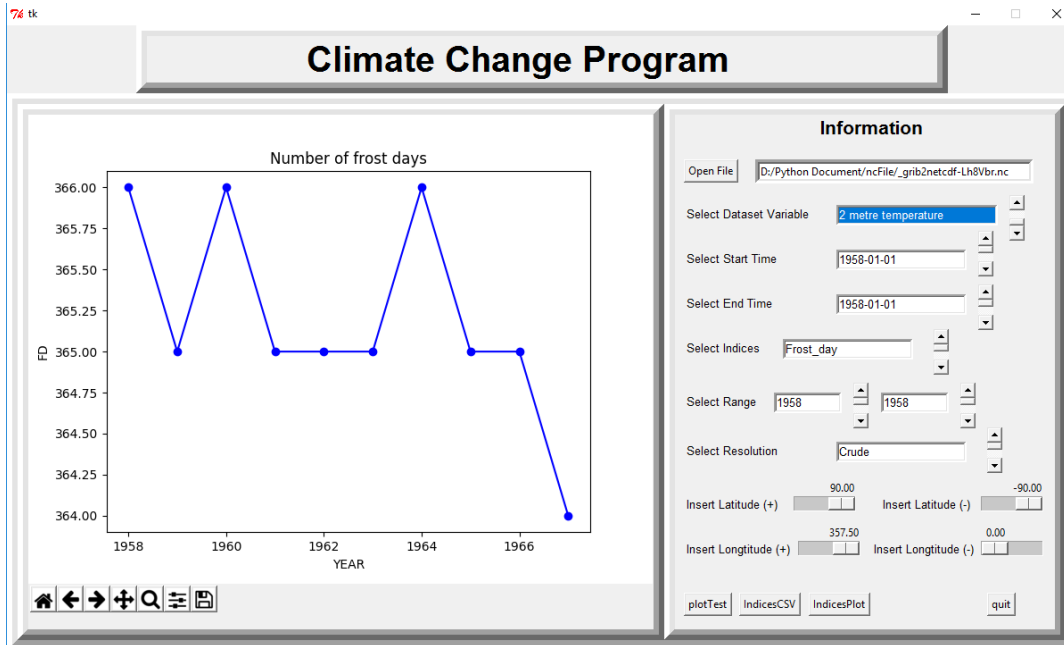
จากภาพโปรแกรมข้างต้นจะพบว่า Frame ที่ออกแบบมาสำหรับเลือก Measurement ต่างๆนั้น ค่อนข้างแออัดเนื่องจากมีพื้นที่ที่จำกัดและตัว Scrollbar ที่ใช้สำหรับเลือกข้อมูลนั้นใช้งานได้ยาก

ปัญหาต่างๆที่พบในการทำงานและการแก้ไข



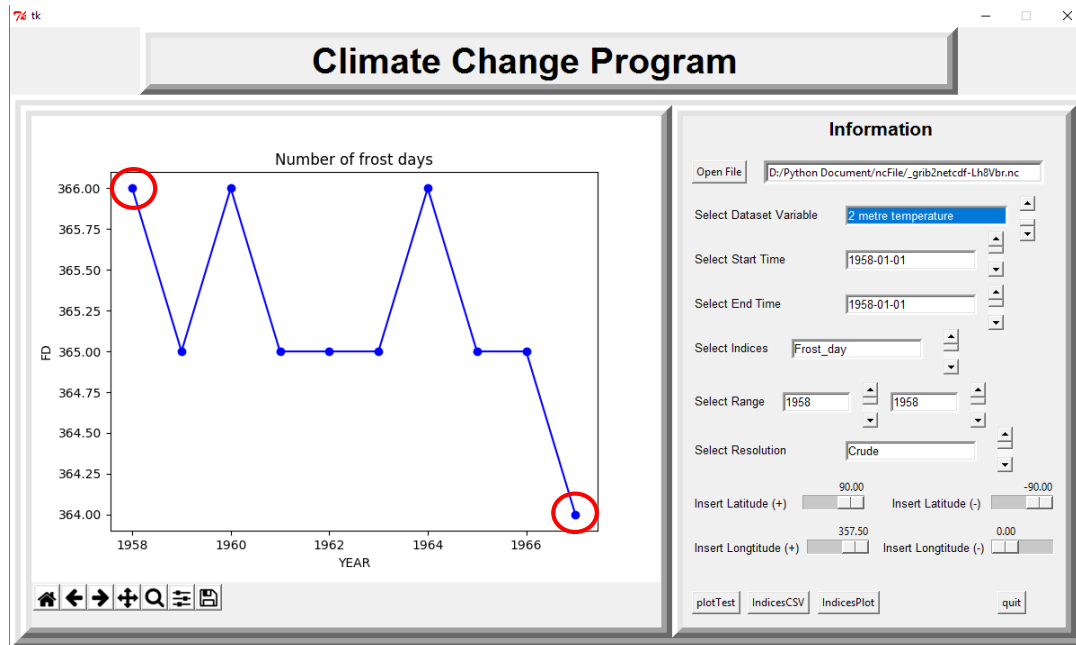
ในส่วนของการแก้ไขนั้น เราได้ทำการเปลี่ยน Platform ที่สร้าง GUI จาก Tkinter เป็น PyQt ซึ่งให้ผลตรงกับที่เราต้องการ และการทำงานที่ง่าย ทำให้สามารถออกแบบ Widget เพิ่มเติมได้และยังสามารถตกแต่งตัว GUI ให้สวยงามน่าให้ได้อีกด้วย

ปัญหาต่างๆที่พบในการทำงานและการแก้ไข



ปัญหาต่อมาที่พบคือการ generate ข้อมูลออกมาจาก ncfile โดยแบ่ง Daily , Monthly และ Annual นั้นมีความเพี้ยนขอข้อมูลเกิดขึ้น คือ วันที่ทำการนับนั้นเกิดการเลื่อนทำให้ข้อมูลคลาดเคลื่อนเช่น ข้อมูลที่ใช้ในการเฉลี่ย และข้อมูลที่ใช้ในการสร้าง Indices

ปัญหาต่างๆที่พบในการทำงานและการแก้ไข



หากสังเกตที่วงกลมจะพบว่า ปี 1958 นั้นไม่ใช่ leap year จึงไม่ควรจะมี 366 วัน และปี 1967 ควรจะมีข้อมูลวันเต็มๆอยู่ที่ 365 วัน แต่กลับมีเพียง 364 วัน เท่านั้นที่แสดงขึ้นมา แสดงว่าการลำดับข้อมูลที่ใช้ในการคิด Indices และ Average ไม่ถูกต้อง

ปัญหาต่างๆที่พบในการทำงานและการแก้ไข

```
for i in range (len(Dates)):
    data += temps[i]
    count += 1
    if (firstMonth != Dates[i].strftime("%B") or i == len(Dates)-1):
        firstMonth = Dates[i].strftime("%B")
        test.append(data/count)
        count = 0
        data = 0
print len(test[0][0])
```

ปัญหาในส่วนนี้เกิดขึ้นจากการนับจำนวนวันที่ผิดพลาดของเงื่อนไขใน Program เนื่องจากเงื่อนไขไม่ครอบคลุมการทำงานและตรวจสอบไม่ครบ ทำให้มีข้อมูลที่ตกหล่นหรือเกินไป

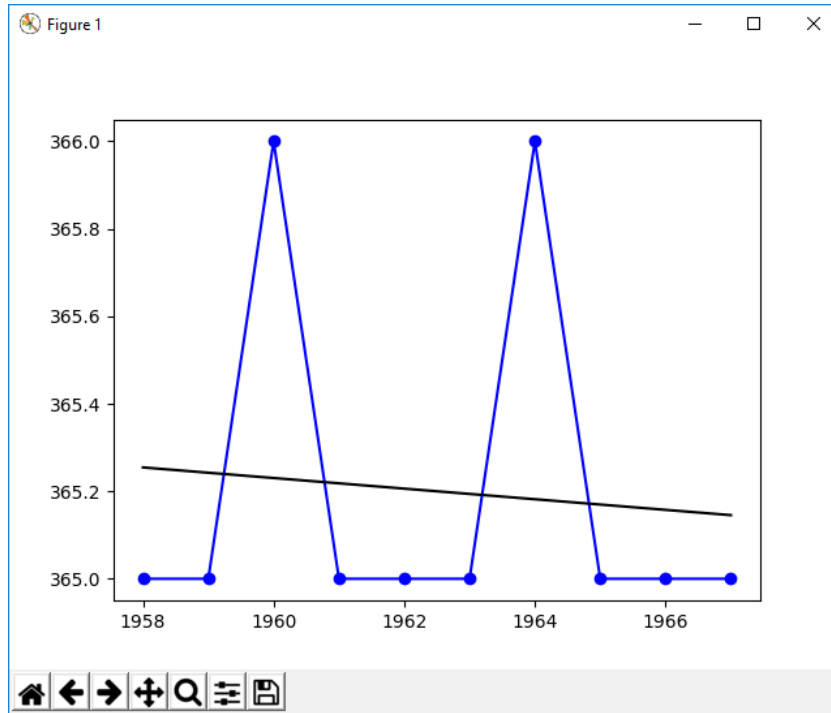
จากโค้ดด้านบนตรงส่วนการ count นั้นก่อนเข้าเงื่อนไข ได้ทำการนับค่าเกินจำนวนที่ควรจะเป็นเพราะอยู่นอกเงื่อนไขเช่น เมื่อ $i = 30$ ค่า count ควรจะหยุดที่ 31 แต่เนื่องจาก 30 ยังไม่เข้าเงื่อนไขที่เปลี่ยนเดือนเมื่อ $i = 31$ ค่า count จึงเกินมา 1 ค่า และทำให้ข้อมูลเดือนเพี้ยนไปก่อนที่จะคิดเดือนใหม่

ปัญหาต่างๆที่พบในการทำงานและการแก้ไข

```
for i in range (len(Dates)):
    if ( firstMonth != Dates[i].strftime("%B") ):
        firstMonth = Dates[i].strftime("%B")
        self.monthdata.append(data/count)
        count = 1
        data = 0
        data += nc.variables[self.key][i]
    elif ( i == len(Dates)-1 ):
        count += 1
        data += nc.variables[self.key][i]
        self.monthdata.append(data/count)
        firstMonth = Dates[0].strftime("%B")
    else:
        data += nc.variables[self.key][i]
        count +=1
```

ได้ทำการแก้ไขโดยปรับให้การเพิ่มค่านั้นอยู่ในเงื่อนไข if else ทั้งหมด เพื่อป้องกันการอัปเดตค่าที่เกินเงื่อนไข และตรวจสอบข้อมูลตัวสุดท้ายเพื่อป้องกันการตกหล่นของข้อมูล

ปัญหาต่างๆที่พบในการทำงานและการแก้ไข



ผลจากการแก้ไขทำให้ข้อมูลแต่ละวัน
ของ Indices กลับมาตรงตามที่จะ
เป็นอีกครั้ง

การทดสอบโดยใช้ Unittest

Unittest คือโปรแกรมสำหรับตรวจสอบการทำงานของตัว Method ใน Class นั้นๆ ว่าทำงานได้ถูกต้องหรือไม่ หากเรากำหนด Input และ Output แล้ว ค่า input ที่เข้าไปคิดใน Method จะได้ผลลัพธ์ตรงกับ Output ที่เราตั้งไว้หรือไม่ โดยในที่นี้จะแบ่งการ test ออกเป็น 4 ส่วนหลักๆ คือ

- ส่วนของข้อมูลที่ใช้สำหรับ real time plot และ average plot
- ส่วนของการทำงานของ GUI
- ส่วนของการสร้างไฟล์ Json
- ส่วนของการสร้าง Indices Data

การทดสอบโดยใช้ Unittest

ในส่วนของ real time plot และ average plot นั้นแบ่งออกเป็น 7 tests ย่อยๆได้ดังนี้

- ทดสอบการอ่าน measurement key จากไฟล์ netCDF
- ทดสอบการอ่านค่าเวลาจากไฟล์ netCDF
- ทดสอบการเก็บค่า longname ของ measurement key
- ทดสอบการอ่านค่าของ latitude และ longitude ของไฟล์
- ทดสอบการอ่านช่วงเวลาที่จะนำมาใช้เฉลี่ยข้อมูล
- ทดสอบการเฉลี่ยข้อมูลเป็นรายเดือนโดยตรวจสอบจากจำนวนเดือนที่ได้มา
- ทดสอบการเฉลี่ยข้อมูลรายเดือนตามช่วงเวลาที่ได้รับมา

การทดสอบโดยใช้ Unittest

```
Indices_plot.py x Default_Plot.py x trend_test.py x _GUI_Climate6.py x _4Controller4.py x Climate_Change_V7.py x test_nc.py x unit_test_climate.py x averageData2.py x tot.py x
1 import unittest
2 from data_class import Get_Data
3 from averageData2 import AverageData
4 from netCDF4 import Dataset,num2date
5
6 class TestClimate(unittest.TestCase):
7     def setUp(self):
8         self.file_name = 'ncFile/ grib2netcdf-Lh8Vbr.nc' # set test file
9         self.test = Get_Data(self.file_name) # set data class
10        self.test.variable() # set variable to use in unittest
11        self.check = AverageData(0,7,self.file_name,'t2m','latitude','longitude') # set average data class
12
13    def test_get_key_method_returns_correct_result(self):
14        key = ['longitude', 'latitude', 'time', 'tcwv', 't2m'] # set sample key
15        self.assertEqual(key, self.test.variable()) # assert test with key from variable method in Get_Data class
16
17    def test_get_timezone_method_returns_correct_result(self):
18        self.assertEqual(self.test.get_timezone(), self.test.set_time()) # assert test with time data from set time method in Ge
19
20    def test_get_key_name_method_returns_correct_result(self):
21        keyname = ['longitude', 'latitude', 'time', 'Total column water vapour', '2 metre temperature'] # set sample key name
22        self.assertEqual(keyname, self.test.key_check()) # assert test with key name from key check method in Get_Data class
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
.....
Ran 7 tests in 8.633s
OK
[Finished in 9.9s]
```

ผลการทดสอบและเวลา
ที่ใช้ไปทั้งหมดในการ
ทดสอบ

การทดสอบโดยใช้ Unittest

```
52
53 def test_add_method_returns_correct_Controller(self):
54     self.assertEqual('Mean sea level pressure', self.c.setKey('Mean sea level pressure'))
55     self.assertEqual('_grib2netcdfChRz9Z', self.c.getFileName('_grib2netcdfChRz9Z.nc'))
56     self.assertEqual('.nc', self.c.getExtension('_grib2netcdfChRz9Z.nc'))
57
58     self.assertEqual([1, 2, 3], self.c.setKey([1, 2, 3]))
59     self.assertEqual('2 metre', self.c.setPoint('2 metre'))
60     self.assertEqual(0, self.c.setFirstIndex(0))
61     self.assertEqual(0, self.c.setLastIndex(0))
62     self.assertEqual('i', self.c.setResolution('high'))
63
64     data = Dataset('ncFile/_grib2netcdfChRz9Z.nc', 'r')
65     time = data.variables['time']
66     self.assertEqual('1957-09-01', self.c.ncToTime(time, 0))
67     self.assertEqual(3, self.c.averageValue([1, 2, 3, 4, 5]))
68     temp = [[[1, 2, 3], [1, 2, 3], [1, 2, 3]],
69             [[4, 5, 6], [7, 8, 9], [7, 7, 7]]]
70
71     self.assertEqual(temp[0], self.c.getMinValue(temp))
72     self.assertEqual(temp[1], self.c.getMaxValue(temp))
73
74     self.assertEqual(True, self.c.setShape('curve'))
75     self.assertEqual('ICE', self.c.setIndexName('ICE'))
76
```

บรรณานุกรม

- ภาพ Matplotlib <http://javarevisited.blogspot.com/2014/12/how-to-read-write-json-string-to-file.html#axzz4gK6al6NY>
- ภาพ Json <http://javarevisited.blogspot.com/2014/12/how-to-read-write-json-string-to-file.html#axzz4gK6al6NY>
- ภาพ netCDF <https://clas-pages.uncc.edu/techne/netcdf/>
- ภาพ PyQt <https://en.wikipedia.org/wiki/PyQt>
- ภาพ NumPy <https://bids.berkeley.edu/research/numpy>