

作物の気象災害における メッシュ気象データの 活用と展望

実習テキスト

平成 26 年度総合研究試験研究推進会議（農業気象分野）第 8 回農業気象研究会

実習1 MS-Excelによるメッシュ農業気象データの取得と利用

この実習では、メッシュ農業気象データを利用するために、一般に広く利用されているMicrosoft社のExcelを用いてデータを取得する方法を習得します。実習で配布するエクセルファイルは、緯度・経度・年（月日）・取得したい要素を指定するだけで、メッシュ農業気象データを取得することが可能です。

1. はじめに

実習では以下の3つのエクセルファイルを配布します。

[ファイル1]: メッシュ農業気象データ_ポイント単要素抽出_ver2.0.xlsm

[ファイル2]: メッシュ農業気象データ_ポイント全要素抽出_ver2.0.xlsm

[ファイル3]: メッシュ農業気象データ_2次元抽出_ver2.0.xlsm

この3つのファイルは、「webクエリ」と「VBAによるマクロ」の機能を使用しているため、通常はファイルを開いた際に「セキュリティの警告」が表示され、これらの機能が使用できない状態にされてしまいます。Excel 2007の場合、図の1-1のような表示です。

機能を有効にするには、「オプション」をクリックすると図1-2のような画面が現れますので、「マクロ」と「データ接続」について、それぞれ「このコンテンツを有効にする」を選択し、OKをクリックします。

これらのファイルを利用することで、メッシュ農業気象データシステム上の全ての農業気象要素にアクセスすることが可能です。つまり利用可能なデータは、以下の表1-1の通りです。なお、[ファイル3]においては、標高データも取得・表示が可能です。

どのファイルにおいても、取得したい地点の緯度・経度を指定する必要があります。メッシュ農業気象データシステムの測地系はJGD2000準拠ですが、WGS84との誤差は数cm以内であり、実質上どちらの測地系における緯度・経度情報で差し支えありません。

対象地点の緯度・経度は、web上で調べることも可能です。(例: <http://www.geocoding.jp>)

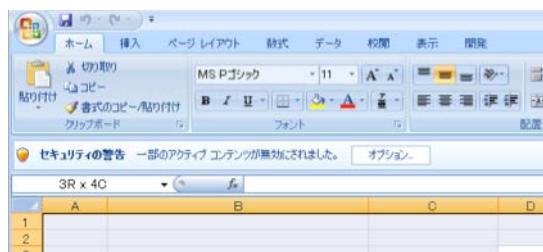


図1-1. 実習ファイルを開くと現れるセキュリティの警告(Excel2007の場合)。「オプション」ボタンを押すと図1-2の画面が現れる。

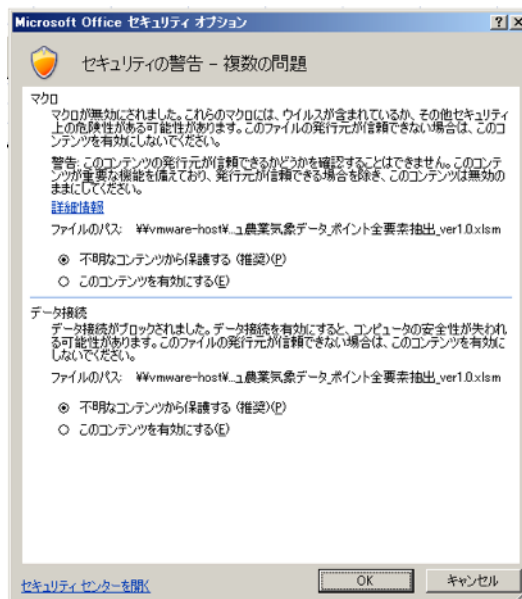


図1-2. この画面で、「マクロとデータ接続について、「このコンテンツを有効にする」を選択し、OK ボタンを押せば、実習ファイルを使用する準備完了です。

表1-1 実習ファイルで利用可能なデータ種類

要素名	2008年～現在	1980～2007年	平年値（1981～2010年）
平均気温	○	○	○
最高気温	○	○	○
最低気温	○	○	○
相対湿度	○		
日照時間	○	○	○
日射量	○	○	○
下向き長波放射量	○		
降水量	○	○	○
風速	○		

2. ある特定地点を含む1メッシュのデータを、単要素について1年分取得する

[ファイル1]を使用します。ここでは例として、石垣島中央部に広がる畑(図1-3)を対象とし、2013年の降水量データを取得しましょう。

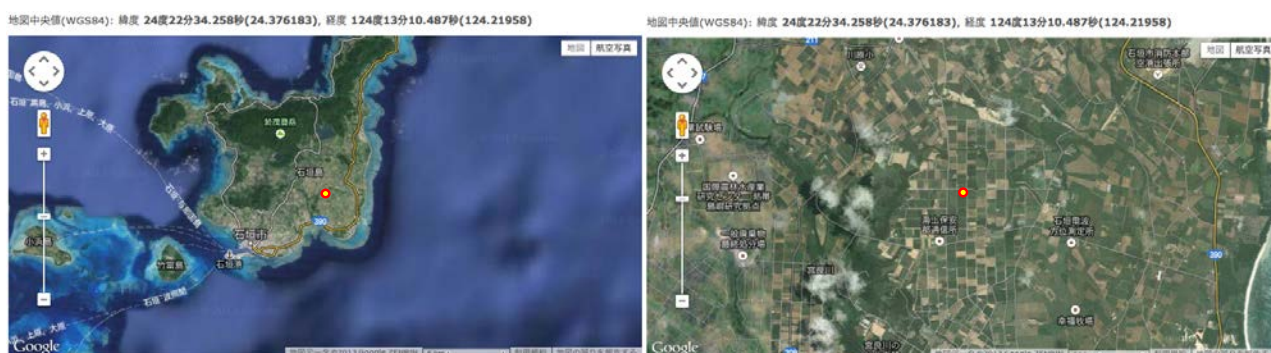


図1-3 データを取得する、石垣島中央部に広がる畑地。赤丸が対象地点。

取得するのに必要な情報	要素： 降水量	年： 2014
	緯度（北緯）： 24.351945	経度（東経）： 123.929193

[ファイル1]を開くと、「データ取得シート」の左上の方に、太枠で囲まれた「データ要素」「データ取得年」「地点の北緯」「地点の東経」を入力するセル(図1-4：A)があります。データ要素については、クリックするとプルダウンメニューが開きますので、その中から「降水量」を選択します。地点の北緯と東経を入力すると、その位置がシートの左側の図(図1-4：B)にその位置がプロットされますので、おおまかにですが、指定した地点が間違えていないかを確認できます。

必要な4つの情報を入力し終わったら、入力欄の右側にある「データ取得」ボタン(図1-4：C)をクリックします。すると、取得したデータが右側の図(図1-4：D)に反映され、そのデータが図の右側の列(図1-4：E)に表示されます。データ取得の方法は以上です。

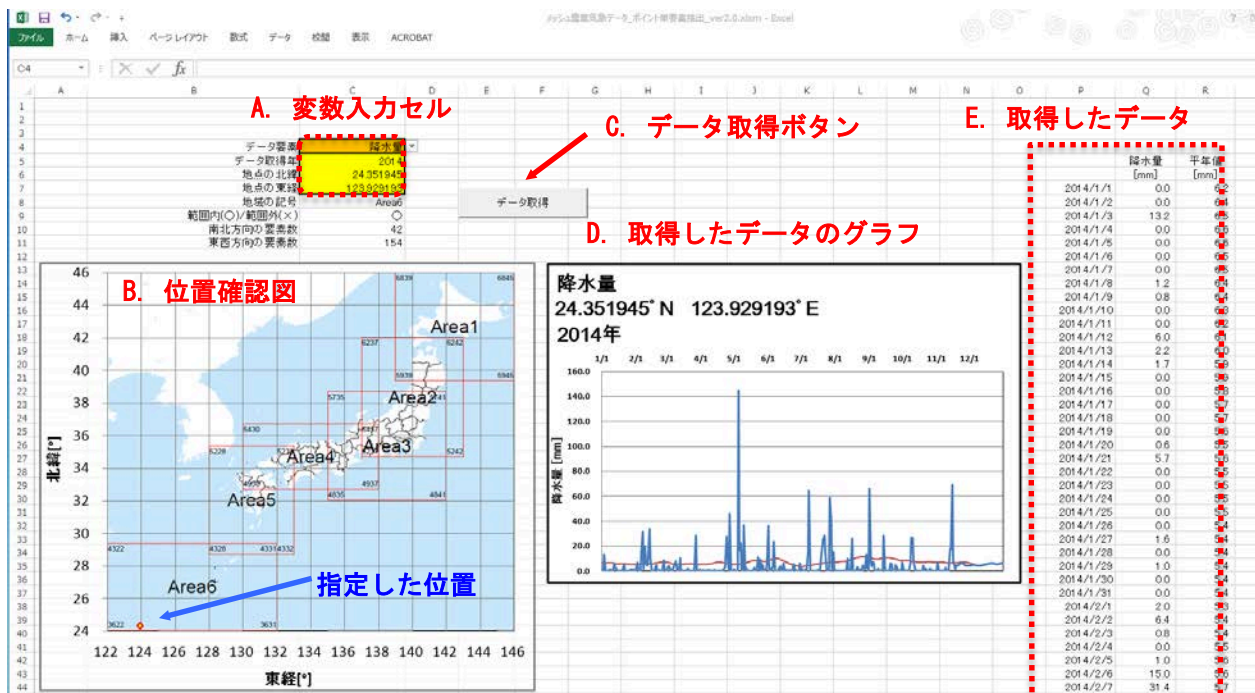


図1-4. ファイル1のデータ処理シート

「降水量」のように平年値のあるデータ要素の場合、その平年値も同時に取得します。グラフ上では平年値が赤線、対象年のデータが青線で示されます。また、メッシュ農業気象データは実況値、予報値以降は、平年値のデータで埋められています。そのため、取得した2013年の年の終わりの方のデータは、平年値と同一になっています。

緯度・経度を指定してデータを取得するため、ピンポイントのデータが得られるように感じますが、メッシュ農業気象データの空間解像度は、東西・南北方向共に約1kmですので、約1km²を代表した値となりますので、ご注意ください。

※ シートに取得したデータは、他のエクセルブックにコピー&ペーストして使用することをお勧めします。

3. ある特定地点を含む1メッシュのデータを、全要素について1年分取得する

[ファイル1] だけでも、取得気象要素を変えてデータ取得を繰り返すことで、様々なデータを抽出することが可能です。もし、同じ地点の様々な気象要素を同時に眺めたい時には [ファイル2] を利用することで、一度の操作で選択したメッシュの要素を全て取得することが可能です。

ここでは、長野県安曇野市のわさび農園近くの水田(図1-5)を対象とします。2012年のデータ要素全てを取得してみましょう。

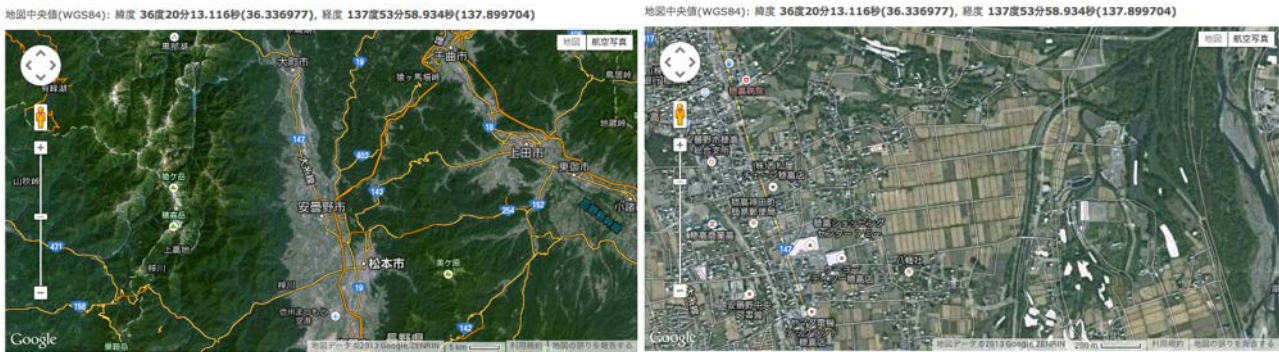


図1-5. データを取得する、長野県安曇野市のわさび農園近くの水田。赤丸が対象地点。

取得するのに必要な情報	年 :	2012
	緯度 (北緯) :	36.33697
	経度 (東経) :	137.899704

全ての要素を取得するため、指定する項目は上記3点のみです。これらの情報を [ファイル2] の太枠内 (図1-6:A) に入力し、右側の「データ取得ボタン」 (図1-6:C) を押すと、データの取得が始まります。

A: 変数入力セル

データ要素	全要素一括
データ取得年	2012
地点の北緯	36.336977
地点の東経	137.899704
地域の記号	Area2
範囲内(O)/範囲外(X)	<input type="radio"/>
南北方向の要素数	200
東西方向の要素数	71

C: データ取得ボタン

D: 取得したデータのグラフ

E: 取得したデータ

	平均気温 [°C]	均気温 [°C]	年間最高気温 [°C]
2012/1/1	0.0	-0.1	8.3
2012/1/2	-0.2	-0.1	3.8
2012/1/3	-2.6	-0.2	3.7
2012/1/4	-1.3	-0.3	1.8
2012/1/5	-3.7	-0.3	1.4
2012/1/6	-3.6	-0.4	3.3
2012/1/7	-1.6	-0.5	3.7
2012/1/8	-4.0	-0.5	2.6
2012/1/9	-1.4	-0.6	6.3
2012/1/10	-0.7	-0.7	5.2
2012/1/11	-1.2	-0.7	2.6
2012/1/12	-4.5	-0.8	1.0
2012/1/13	-1.4	-0.8	7.8
2012/1/14	-1.6	-0.8	1.8
2012/1/15	-1.0	-1.0	3.5
2012/1/16	-1.1	-1.0	1.3
2012/1/17	-1.3	-1.1	3.4
2012/1/18	-2.0	-1.2	6.6
2012/1/19	-0.1	-1.2	7.0
2012/1/20	0.5	-1.3	2.1
2012/1/21	0.2	-1.3	1.9
2012/1/22	1.5	-1.4	4.8
2012/1/23	0.1	-1.4	2.9
2012/1/24	-2.2	-1.5	1.3
2012/1/25	-1.2	-1.5	1.1
2012/1/26	-3.3	-1.5	-1.4
2012/1/27	-3.0	-1.5	-0.3
2012/1/28	-3.9	-1.6	-1.2
2012/1/29	-5.8	-1.6	-1.3
2012/1/30	-4.4	-1.5	1.1
2012/1/31	-3.6	-1.5	0.5
2012/2/1	-1.6	-1.5	3.0
2012/2/2	-5.7	-1.4	-2.9
2012/2/3	-6.2	-1.4	1.4
2012/2/4	-0.7	-1.3	4.7
2012/2/5	-1.0	-1.2	3.3
2012/2/6	-0.2	-1.1	3.7
2012/2/7	3.3	-0.9	5.5
2012/2/8	0.0	-0.8	3.8
2012/2/9	-2.1	-0.7	1.0

図1-6 ファイル2のデータ処理シート

今回は [ファイル1] のデータ取得に比べて時間がかかります。これは、最大で19要素のデータを取得するために「webクエリ」によるデータのダウンロードが行われるからです。シートの動作が終わるまで、焦らずに待ちましょう。

4. ある特定地点を中心とした空間分布のメッシュデータを取得する

特定地点の時系列だけでなく、空間分布も見てみたいと思います。ただし、エクセルでは大量のデータを扱う場合の処理時間が遅いため、広範囲のデータを一度に取得するのは現実的ではありません。そのため、2次元データを取得する [ファイル3] では、中心地点から南北方向に161メッシュ、東西方向に101メッシュに制限しています。

ここでは、北海道羽幌町の水田(図1-7)を中心とした、平均気温データを抽出してみます。

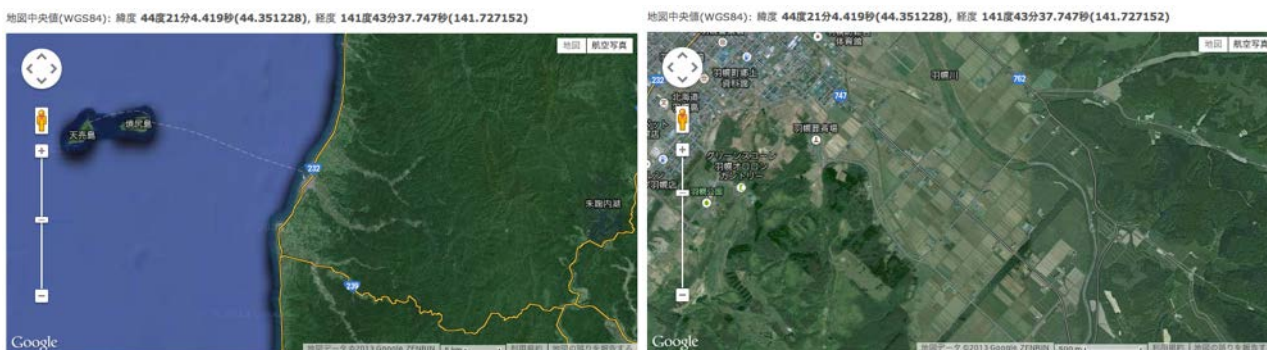


図1-7. 羽幌町の水田。赤丸が対象地点。

取得するのに必要な情報	要素： 平均気温	データ種類： 日別値
	年月日： 2010/6/30	地域： Area1
	中心緯度（北緯）： 44.351228	中心経度（東経）： 141.727152

この [ファイル3] では、入力する情報が増えています (図1-8:A)。選択できる要素は、[ファイル1] [ファイル2] と同じメッシュ農業気象要素 (表1-1) の他、標高データも選択出来ます。データ種類は「日別値」の他、平年値のある要素については、「平年値」と「平年偏差」も選ぶことができます。日付は、ある特定日を指定する必要があるため、年/月/日を指定する必要があります。

[ファイル1] [ファイル2] と大きく異なる点は、中心緯度・経度の指定だけでなく、地域を選択する必要があることです。メッシュ農業気象データは、全国を6つのエリアに分けて提供しています。位置確認図にあるArea1～6の領域です (図1-8:B)。現バージョンの [ファイル3] では、このエリアをまたぐような領域のデータを抽出できません。位置確認図 (図1-8:B) に、中心座標のプロットの他、選択したAreaが赤太破線でハイライトされ、その中でデータを抽出する南北161メッシュ、東西101メッシュの領域が、青太実線で示されています。青太線のエリアが十分に得られるよう、中心座標を指定して下さい。

全ての必要な情報を入力したら、利用可能なデータを取得するために2つのボタンが用意されています。「ステップ1: データ取得」 (図1-8:C) と「ステップ2: データ変換」 (図1-8:D) の2つです。これらは順番に、ステップ1の終了後に、ステップ2を実行する必要があります。ステップ1では、webクエリでメッシュ農業気象システムからデータを取得します。このデータをエクセル上で東西南北の向きを正しく眺めるために、データ変換のステップ2が必要となります。

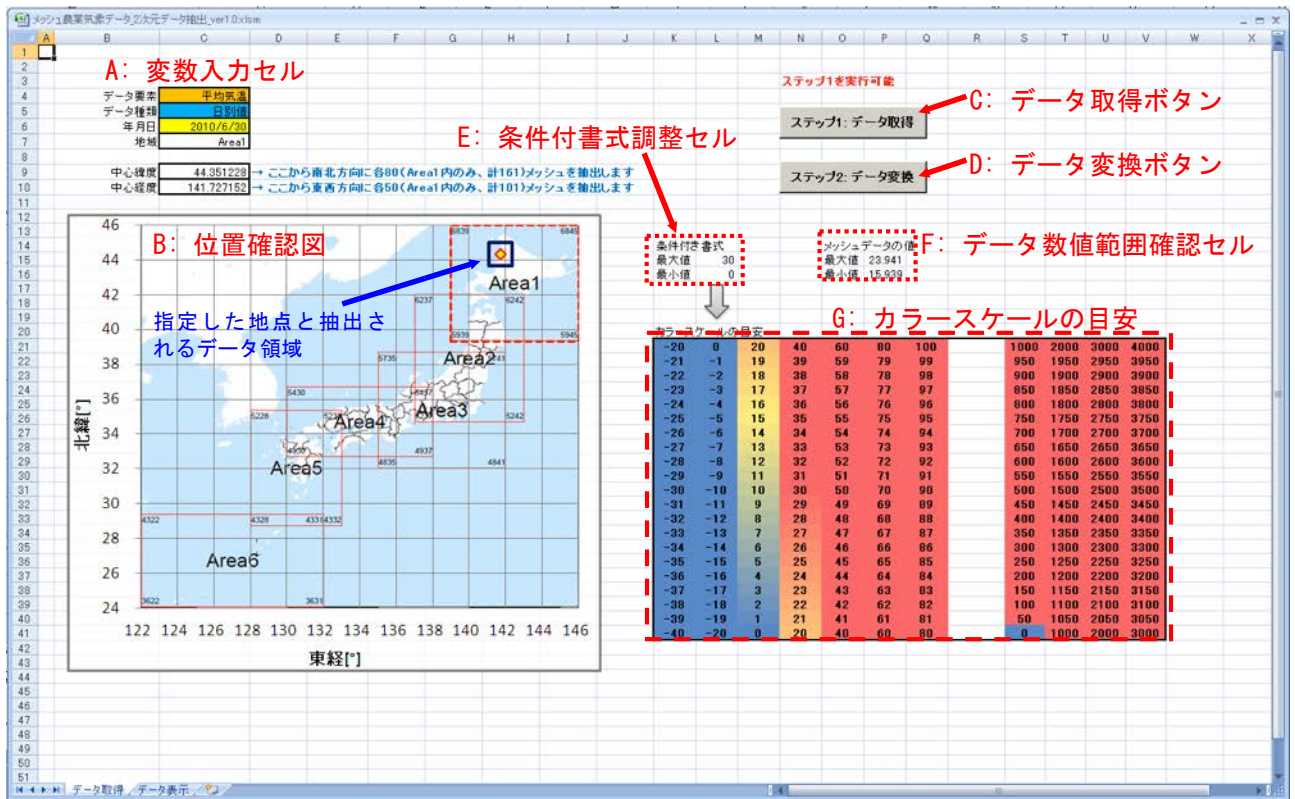


図1-8. ファイル3のデータ取得シート

ステップ2を終了したら、データ表示シートを見てみましょう（図1-9）。このシートには緯度情報の入ったセルがC列（図1-9:A）に、経度情報の入ったセルが2行目（図1-9:B）にあり、それぞれ背景色がグレーになっています。これらの右下の部分（図1-9:C）に取得した2次元のメッシュデータが各セルに数値として入っています。このシートにはエクセルの機能である「条件付き書式」により、セルの数値に応じて色が変わります。

この「条件付き書式」の数値と色の対応は、「データ取得」シート上で調整することが可能です。「データ取得」シートの位置確認図の右に、条件付き書式の最大値と最小値を指定するセル（図1-8:E）があります。今はこの値が、最小値：0、最大値：30になっていました。このセルの右側に、メッシュデータの値の最大値と最小値が示してあります（図1-8:F）。これは、「データ表示」に抽出されたメッシュデータ全ての値の最大値と最小値です。ここでは、最大値が22.941、最小値が15.939となっていますので、条件付き書式の0-30の範囲はかなり広がったようです。データ表示における色の変化を大きくするには、メッシュデータの値の範囲と条件付き書式の範囲を近づけてあげればよいので、ここでは、最大値を24、最小値を15としてみました。すると、図1-10のように、色のコントラストが大きくなりました。

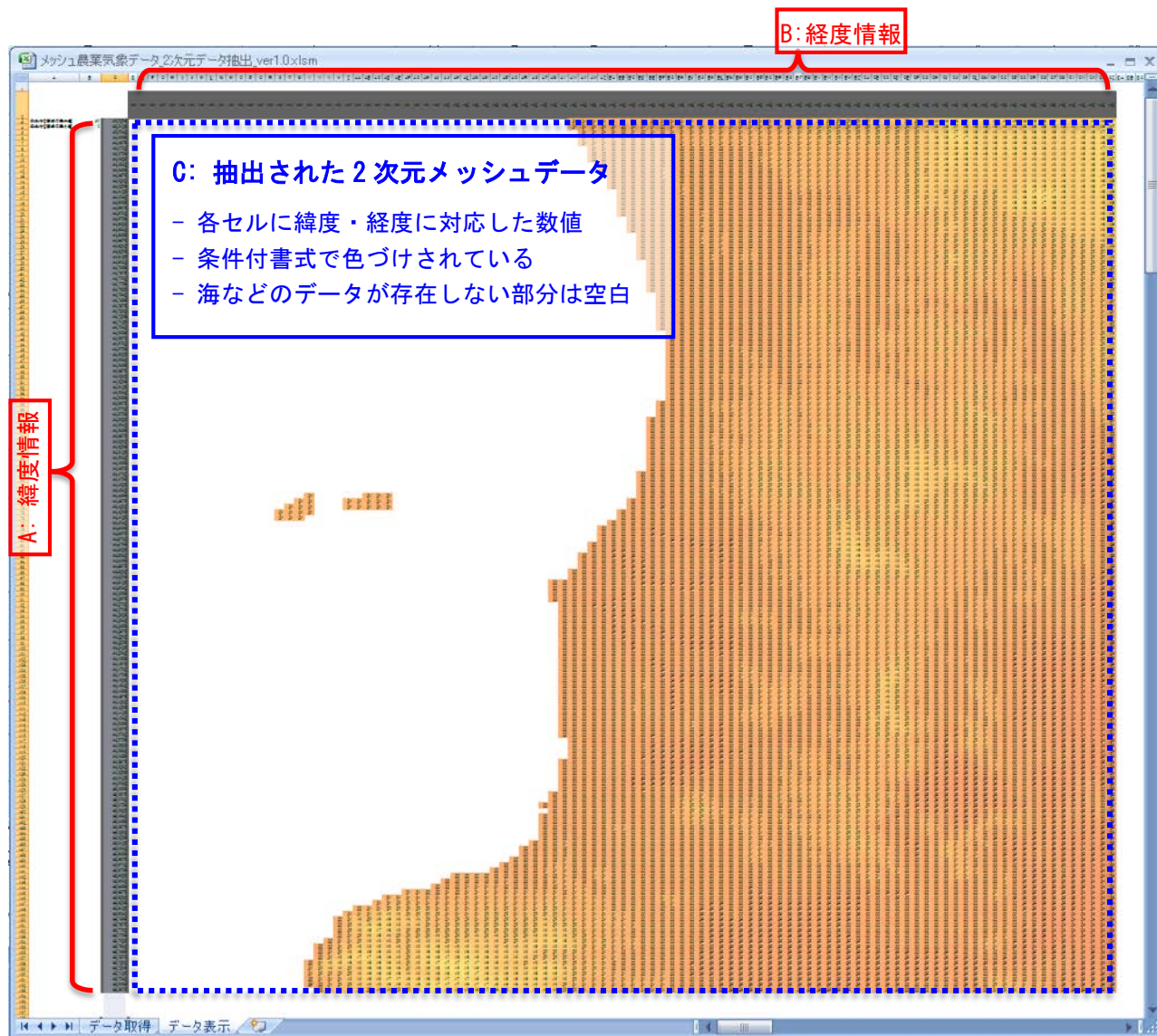


図1-9. 「ファイル3」の「データ表示」シート。条件付書式の範囲:最小値0 最大値30

5. おわりに

この実習では、あらかじめ用意した3つのファイルで、エクセル上でメッシュ農業気象データを取得し、単地点の農業気象要素1年分の取得・図示と、特定日の2次元分布を得る方法を学びました。この3つの実習ファイルを利用することでデータの取得・閲覧は容易ですし、その後の解析も簡単なものであればエクセルだけで全てが済むかと思えます。しかし、取得したデータを用いて複雑な数値計算をしたり、特に2次元分布の時系列変化などを調べたりするには、エクセルでは時間がかかり、現実的ではありません。この後の実習で扱うpythonやGISを習得して、メッシュ農業気象データを利用することをお勧めします。

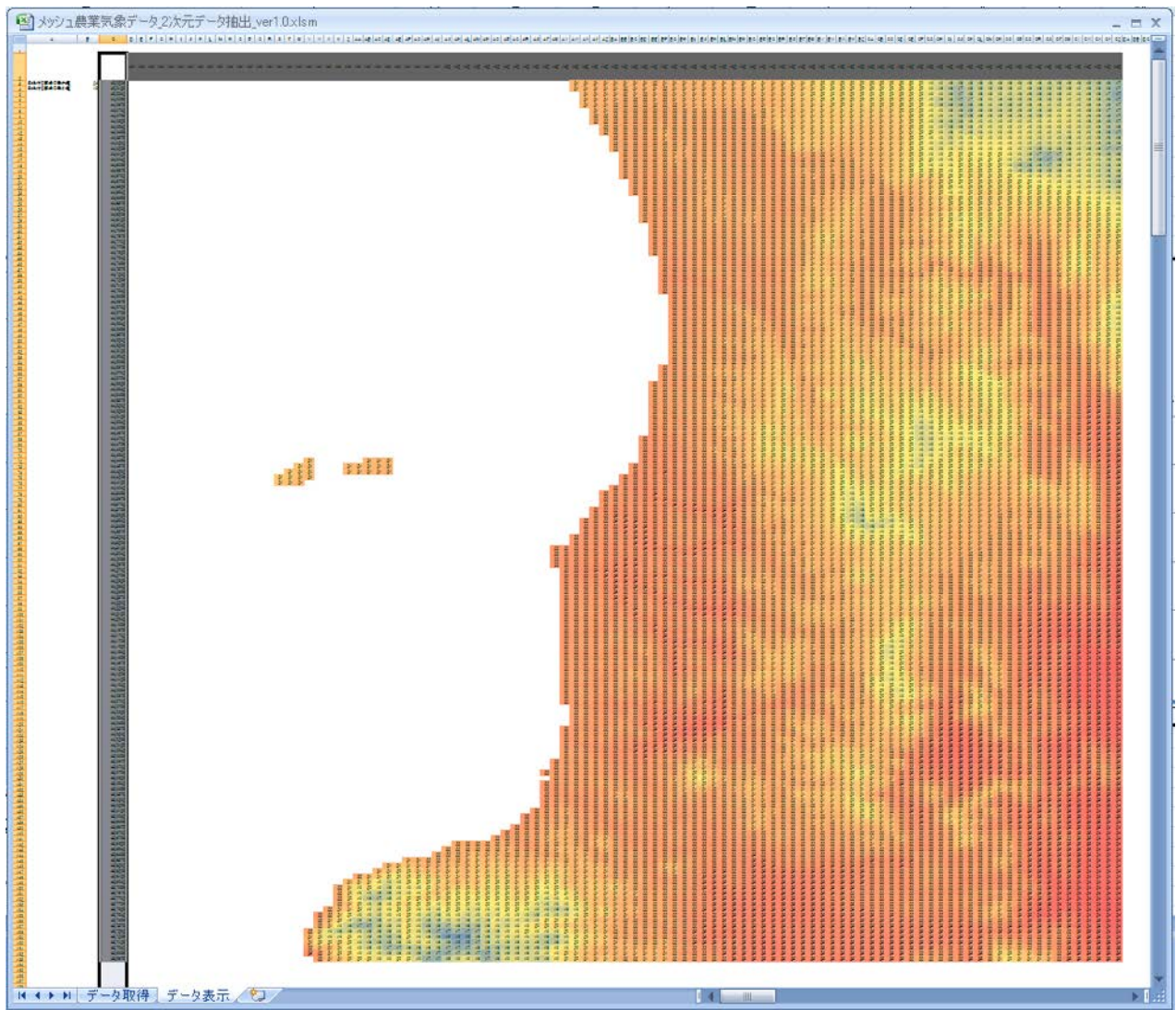


図1-10. 図1-9に対して、条件付き書式の範囲を変更した場合の「データ表示」シート。条件式の範囲:最小値15
最大値24

実習2 Pythonによるメッシュ農業気象データの取得と利用

この実習では、プログラミング言語の一つであるPythonで記述された幾つかのサンプルプログラムの編集と実行を行い、メッシュ農業気象データ配信サーバーからデータを取得して処理するための基礎的知識を身につけます。

1. Pythonプログラムの編集と実行

IPythonの起動：Pythonプログラムは、テキストエディターで編集をしてIPythonコンソールで実行します。

デスクトップにコピーしたフォルダPythonWorksをファイルエクスプローラーで開き、その中のショートカット「それゆけPython!」をダブルクリックします。すると、図2-1のような白いウィンドウが表示されます。これをIPythonコンソールと呼びます。このとき、黒いウィンドウが同時に開きますが、これは気にしないで結構です。邪魔と感じる場合は、タイトルバーの右にある最小化ボタンを押して目立たなくしてください。

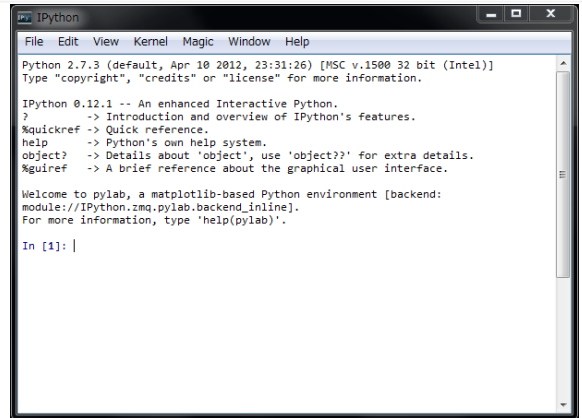


図2-1. IPythonコンソールのスナップショット。Pythonプログラムをここで実行する。

プログラムの実行：IPythonコンソールには、「In [1]:」と表示された行があります。この行をプロンプトと呼びます。ここに、「run プログラムファイル名」と入力してエンターキーを押すとPythonプログラムが実行されます。

○プログラム「meshmap.py」を実行してみましょう。

プログラムの編集：プログラム「meshmap.py」は、近畿中国四国地方における2014年8月1日の日平均気温の分布を表示させるプログラムです。これを対象にプログラムを編集します。

それでは、プログラムファイルをダブルクリックして専用のテキストエディターを開きます。まず、2014年8月1日ではなく、8月9日の温度分布図を描くようにプログラムを変更してみましょう。それには、10行目の文；

```
timedomain = ['2014-08-01', '2014-08-01']
```

を次のように書き換えます。

```
timedomain = ['2014-08-09', '2014-08-09']
```

編集が完了したら、[Ctrl]+sまたは、メニューFile > Saveで変更を保存します。なお、エディターを閉じる必要はありません。保存が終了したら、プログラムを実行します。先ほどと同様、IPythonコンソールのプロンプトに「run meshmap.py」と入力します。IPythonコンソールには、ヒストリー機能とよばれる機能があり、過去に打ち込んだ文字列を記憶しているので、上矢印キー(↑)でこれ呼び出してエンターキーを押してもかまいません。

このように、Pythonでのプログラム開発は、エディターでの編集とIPython上での実行結果の確認を繰り返して進めてゆきます。

○プログラム「meshmap.py」をもう一度実行して、結果の違いを確認してみましょう(図2-2)。

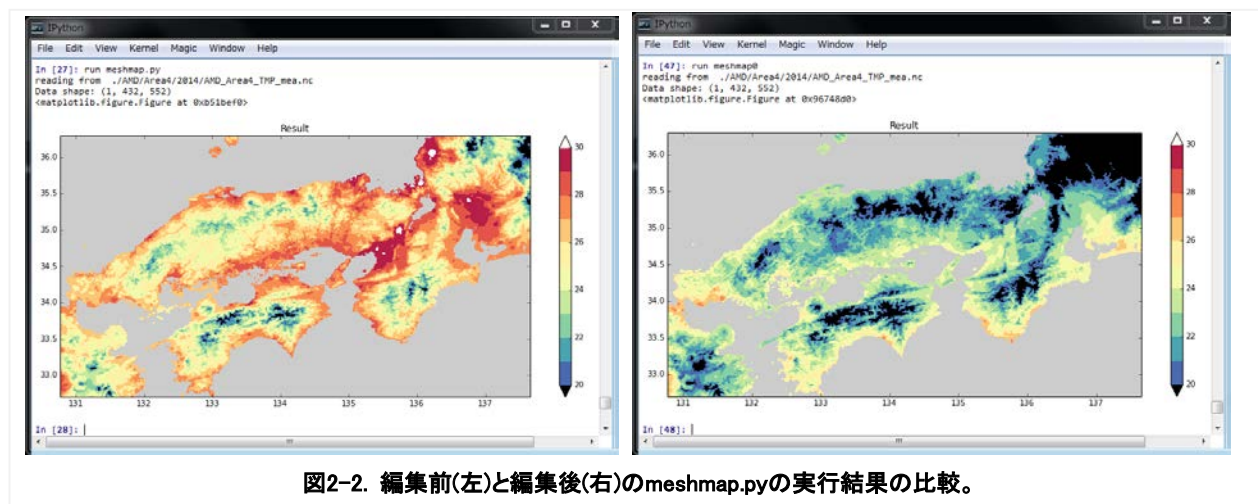


図2-2. 編集前(左)と編集後(右)のmeshmap.pyの実行結果の比較。

メッシュデータは積み上げられた段ボール箱の中

Python プログラムのなかで、メッシュ農業気象データは配列変数に格納されますが、配列変数にデータが格納されている様子は整然と積み上げられた段ボール箱の中の品物に似ています。ここで、品物に相当するのは、特定のメッシュにおける特定の日の特定の気象要素の値 (気温で言えば「23.0°C」など) です。データを処理する際にはこのような配列変数が複数必要になるので、Ta や Tmax などそれぞれに名前を付けて識別します。Python は変数名で大文字と小文字を識別します。

メッシュ農業気象データシステムでは、配列変数に個々のデータを格納する順序を決めています。段ボール箱の例えを用いると、南のメッシュのデータが入っている段ボール箱ほど手前に、西のメッシュのデータの箱ほど左に、日付が新しいデータの箱ほど上に積むことに決めています。そして、この山のなかの特定の箱を指し示すときには、必ず、「下から」「手前から」「左から」の順にその箱の場所を数えることにしています。話を配列変数に戻すと、期間初日から i 日目における、南から j 番目、西から k 番目のメッシュのデータは、配列変数の名前を Ta として $Ta[i,j,k]$ と表記されます。 i や j のことを添え字「そえじ」と呼びます。

Python では、添え字に整数のほかコロン「:」を使用することができ、「全部」を意味します。したがって、南から j 番目、西から k 番目のメッシュにおける期間全部のデータ(よって普通は複数のデータの集合)を Ta から取り出したいときは $Ta[:,i,j]$ と指定します。なお、Python では配列の順番を 0 から数える決まりになっています。

さて、メッシュデータのグラフや分布図を描くときには、 i や j や k が、実際何月何日で、北緯何度で、東経何度に相当するのかわかる必要があります。これらの情報は、GetData()関数から戻される 3 つの配列変数 tim, lat, lon に格納されます。つまり、データ $Ta[i,j,k]$ の日付、緯度、経度はそれぞれ、 $tim[i]$ 、 $lat[j]$ 、 $lon[k]$ に格納されています。段ボール箱の例えでは、Ta のほかに、tim, lat, lon という一列に並んだ段ボールの塊が 3 つあって、それぞれの箱の列の i 、 j 、 k 番目の中に $Ta[i,j,k]$ に対する日付、緯度、経度が書かれた紙が納められています。

逆に、日付・緯度・経度を決めてデータを取り出したいときは、箱列 $tim \cdot lat \cdot lon$ の中の日付や緯度経度をそれぞれ順に照合して一致や最も近い値を見つけ、それらが入っていた箱の番号(i や j や k)を明らかにしてから、求める箱を取り出します。段ボール箱の山相手にこの作業をするのは想像するだけでうんざりしますが、コンピュータはこのような作業が少しも苦でないので心配ありません。

2. 期間平均した日照時間の対平年比を計算するプログラムの作成

プログラム「meshmap.py」をさらに編集して、8月1日から8月31日の期間について平均した広島県における日照時間の分布図を作成するより実用的なプログラムを作成してみます。

別名保存：もともとのファイルが無くなってしまわないよう、ファイルに別名をつけて保存します。プログラムエディターのメインメニューFile > Save As ... を選択し、「meshmap2.py」に改名します。

平均期間の設定：平均期間である8月1日から8月31日のデータを取得するために、10行目を次のように書き換えます。

```
timedomain = [ '2014-08-01', '2014-08-31' ]
```

気象要素の設定：配信サーバーから取り寄せる気象要素を、日平均気温から、日照時間に変更します。それには、9行目を次のように書き換えます。

```
element = 'TMP_mea'
```



```
element = 'SSD'
```

表2-1. メッシュ農業気象データシステムで用いられる気象要素の記号。

農業気象要素	記号	農業気象要素	記号
1 日平均気温	TMP_mea	6 全天日射量	GSR
2 日最高気温	TMP_max	7 下向き長波放射量	DLR
3 日最低気温	TMP_min	8 日積算降水量	APCP
4 日平均相対湿度	RH	9 日平均風速	WIND
5 日照時間	SSD		

平均値計算処理の追加：1月分=31枚の日別最高気温分布データを時間方向に平均する処理を追加し、平均化された分布を得ます。それには、23-24行目を次のように書き換えます。

```
# データの整形(3次元から2次元へ変換)
mme = me[0, :, :]
```



```
# データを時間方向に(第0番目の添え字について)平均する
mme = np.ma.mean(me, axis=0)
```

「np.ma.mean()」は、Pythonの数値計算パッケージnumpyに定義される平均値を計算する関数で、第1引数には、計算対象の配列を、第2引数には、平均をする添え字を指定します。メッシュ農業気象データは、日付、緯度、経度を指定する3つの添え字(第0添え字～第2添え字)を持つデータです。このプログラムでは、日付方向の平均操作なので、「axis=0」として第0添え字について平均することを指示しています。

カラースケールの調整：「meshmap.py」に付けられていた青から赤の配色は、気温の数値に合わせて、20から30でした。今回表示させるのは日照時間ですから、色と数値の対応を調整します。それには、34-35行目を次のように書き換えます。

```
scl mi n = 20.0    #最小値
scl max = 30.0    #最大値
```



```
scl mi n = 0.0    #最小値
scl max = 12.0    #最大値
```

- 修正したプログラム「meshmap2.py」を保存しましょう。
- プログラム「meshmap2.py」を実行して、結果を確認しましょう。

36行目は色の刻みです。現在は1となっていてますから、1時間を単位として色が変化しています。この値を0.5に変更すると30分刻みで色が塗り分けられます(図2-3)。

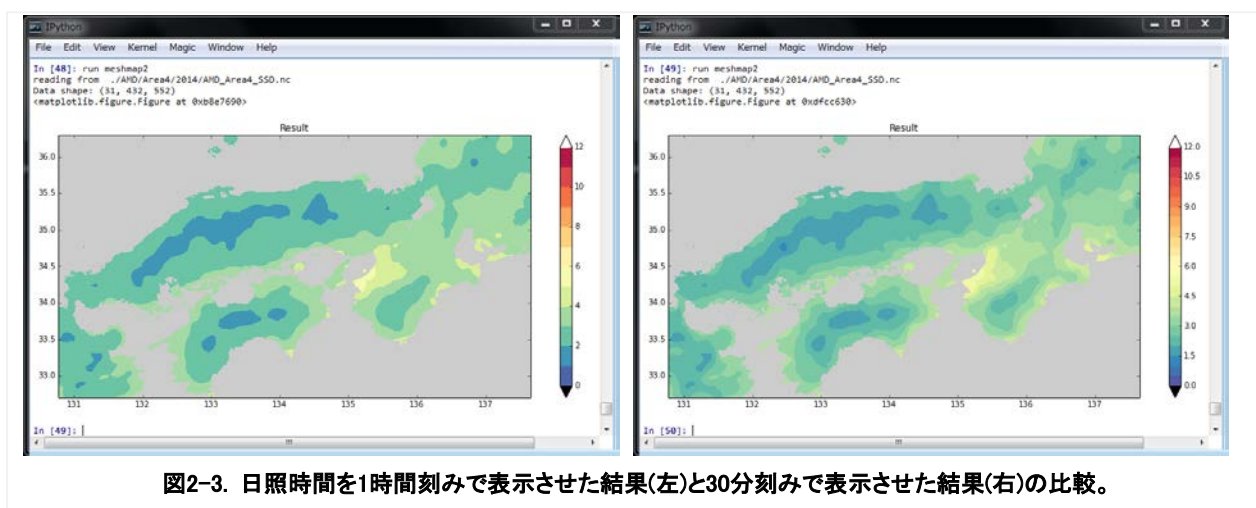


図2-3. 日照時間を1時間刻みで表示させた結果(左)と30分刻みで表示させた結果(右)の比較。

- さらにプログラムを編集します。そのために、プログラムを「meshmap3.py」として別名保存しましょう。

数値計算モジュール numpy

numpy は、Python の機能の拡張モジュールで、ベクトルや行列などの計算を行うための関数や変数サポートが納められています。「np.ma.mean()」は、平均計算のための関数で、他に「np.ma.max()」(最大値)、「np.ma.min()」(最小値)、「np.ma.sum()」(積算値)、なども提供されています。numpy を利用するには、プログラム冒頭でこのモジュールを使うことを宣言することが必要です。これが、4行目の文；

```
import numpy as np
```

です。この文により、Python は、冒頭が「np.」で始まる関数を、numpy というパッケージから探すようになります。なお、プログラム mesh_map.py は、numpy の他、グラフィックスモジュール matplotlib と、メッシュ農業気象データを利用するためのモジュール AMD_tools を使用しています。

平年比の計算：先ほど表示させた日照時間が平年値に対してどれくらいの比であったかを計算してみましょう。そのためには、平年値を取り込む必要があります。17行目に次の文を1行で書き入れます。

```
no, tim, lat, lon = AMD.GetData(element, area, timedomain, lalodomain, cli=1,
url='./AMD')
```

平年値も同じ期間について平均する必要があるので、25行目に以下を書き込みます。

```
mno = np.ma.mean(no, axis=0) # 日付について平均
```

対平年比を計算します。そのために27-28行目に次の分を書き入れます。

```
# 対平年値比の計算。
met = mme / mno
```

この計算で出力されるのは0～1の数値ですから、カラースケールを再調整します。

```
sclmin = 0.0 #最小値
sclmax = 1.2 #最大値
sclint = 0.05 #色の刻み
```

広島県以外の消去：メッシュ農業気象データシステムには、気象データだけでなく、各都道府県について、その都道府県が一部でも含まれる三次メッシュに1、全く含まれないメッシュには無効値が埋め込まれたメッシュデータも用意されています。そこで、このデータを利用して、日照時間の平均値分布図を広島県だけに限定することにします。そのために、19-21行目に次の文を書き入れます。

```
# 県領域データの読み込みとそれによるマスク。
pref, lat, lon = AMD.GetGeoData('pref_3400', area, lalodomain, url='./AMD')
me = me * pref
```

初めの文は、広島県の領域が1、それ以外が無効値であるメッシュデータを取り出す文です。どの都道府県の領域にするかは、文字列「pref_3400」で指定していて、最後の4桁の数字を変更すると、異なる県の領域データが変数prefに代入されます。続く文で、このデータを分布図にかけ算しています。これで、変数meの内容は、広島県内の領域についてはそのままの値、県外の領域については無効値に更新されます。都道府県と、数字の関係を表2-2に示します。

○修正したプログラム「meshmap3.py」を実行して、結果を確認しましょう。

作図範囲の変更：分布図は広島県だけになりましたが、図の範囲が広すぎてバランスが良くありません。分布図を広島県周辺だけに狭めるために、11行の行頭の「#」印を削除し、12行目の行頭に「#」を追加します。

```
l al odomai n = [ 34. 025, 35. 108, 132. 027, 133. 473] #広島県  
#l al odomai n = [ 32. 700, 36. 300, 130. 765, 137. 660] #近畿中国四国
```

Pythonでは、行頭が「#」で始まる行は実行されません。プログラムにコメントを書き込むために用いるのが本来の使用法ですが、一時的に処理を無効化するのに便利なのでこのような使われ方もよくします。この作業を「コメントアウトする」などと呼ぶこともあります。

○修正したプログラム「meshmap3.py」を実行して、結果を確認しましょう。

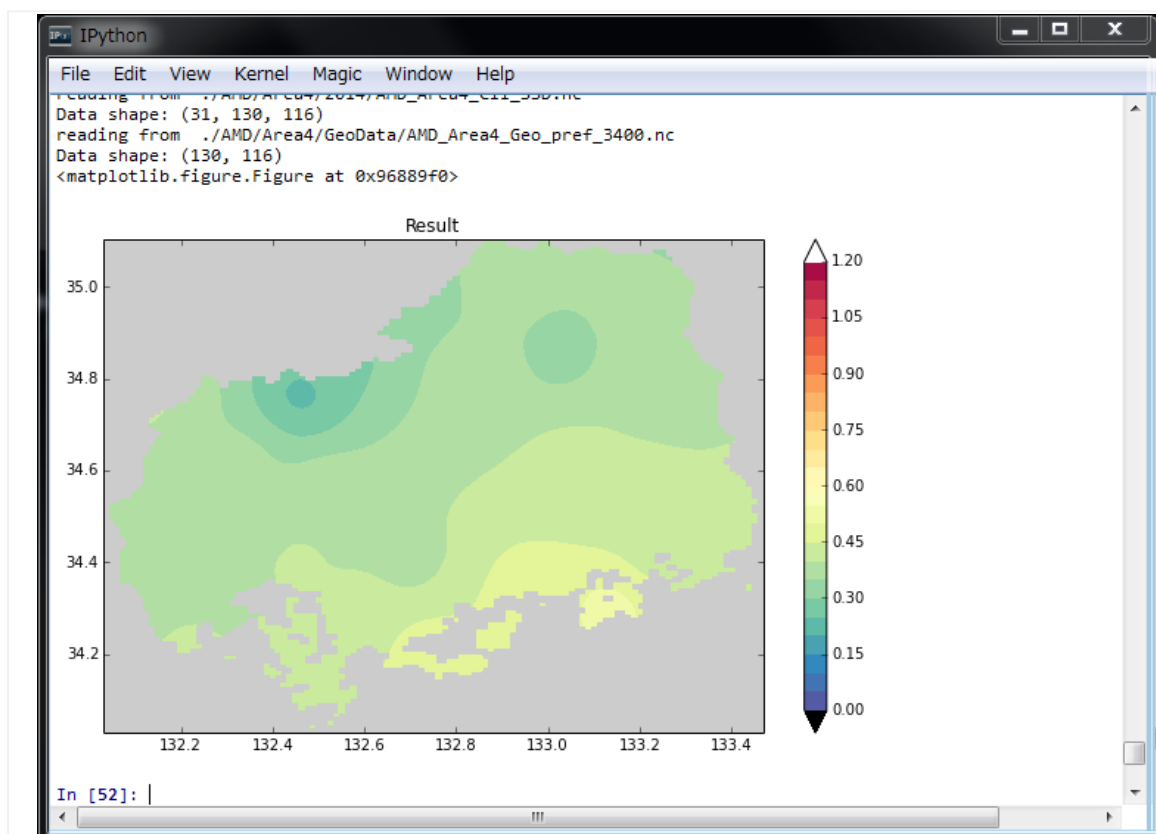


図2-4. 作図範囲を広間県周辺に再設定して表示させた結果。

計算結果の利用：これで綺麗な分布図が作成できました。このプログラムは、図2-4と同じ図を画像(png)ファイルで保存するように作られています。45行目の文でそれを実行しています。この計算結果をエクセルやGISでさらに利用できるよう、各メッシュ毎の値をCSVファイルで出力して保存することにしましょう。そのために、50行目と51行目のコメントアウトを解除します。

○プログラムをもう一度実行してからresult.pngとresult.csvをダブルクリックして、それぞれの内容を確認しましょう。

○実習3で使用できるよう、result.csvをC:\¥GIS_DATAにコピーしておきましょう。

表2-2. メッシュ農業気象データシステムが使用する都道府県を識別する番号。

番号	道(振興局)	番号	都府県	番号	都府県	番号	都府県
0100	北海道	0200	青森県	1800	福井県	3400	広島県
0101	北海道石狩振興局	0300	岩手県	1900	山梨県	3500	山口県
0102	北海道渡島総合振興局	0400	宮城県	2000	長野県	3600	徳島県
0103	北海道檜山振興局	0500	秋田県	2100	岐阜県	3700	香川県
0104	北海道後志総合振興局	0600	山形県	2200	静岡県	3800	愛媛県
0105	北海道空知総合振興局	0700	福島県	2300	愛知県	3900	高知県
0106	北海道上川総合振興局	0800	茨城県	2400	三重県	4000	福岡県
0107	北海道留萌振興局	0900	栃木県	2500	滋賀県	4100	佐賀県
0108	北海道宗谷総合振興局	1000	群馬県	2600	京都府	4200	長崎県
0109	北海道オホーツク総合振興局	1100	埼玉県	2700	大阪府	4300	熊本県
0110	北海道胆振総合振興局	1200	千葉県	2800	兵庫県	4400	大分県
0111	北海道日高振興局	1300	東京都	2900	奈良県	4500	宮崎県
0112	北海道十勝総合振興局	1400	神奈川県	3000	和歌山県	4600	鹿児島県
0113	北海道釧路総合振興局	1500	新潟県	3100	鳥取県	4700	沖縄県
0114	北海道根室振興局	1600	富山県	3200	島根県		
		1700	石川県	3300	岡山県		

3. 指定地点の気象経過グラフを描画するプログラム

プログラムts.pyは、予めプログラム内に記述した地点における気温や降水量などの気象の日変化を、平年値とともに表示します。このプログラムを例に、Pythonプログラムの仕組みを詳しく解説します。

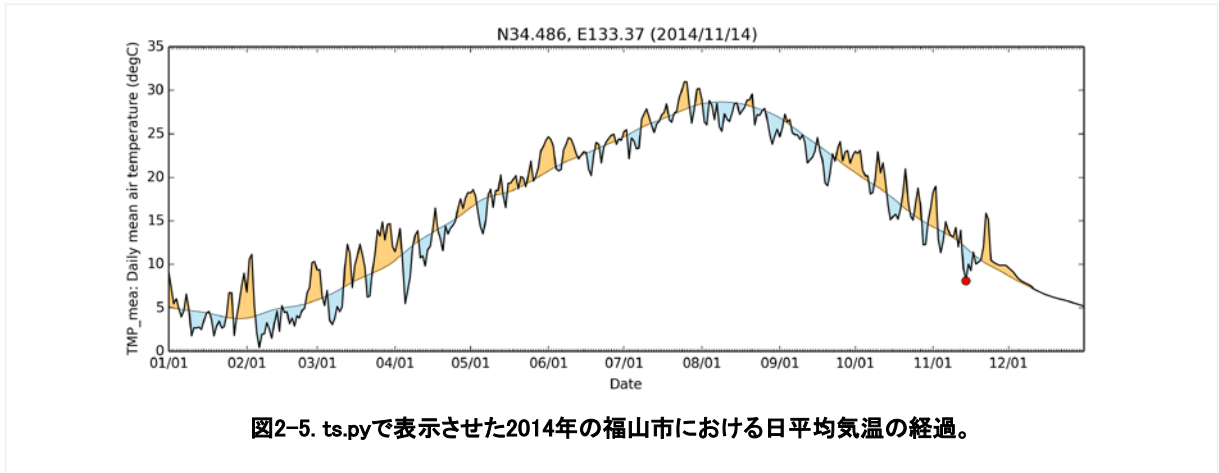


図2-5. ts.pyで表示させた2014年の福山市における日平均気温の経過。

プログラムts.pyの実行：IPythonコンソールのプロンプトに「run ts.py TMP_meas」と入力してエンターキーを押すと、プログラムが実行され福山市の今年の日平均気温の日々変化が表示されます(図2-5)。このプログラムでは、プログラム名の後に、気象要素の略号を付け加えて、プログラムに指示を与えています。このように、プログラム実行の際、追加的に付け加える情報をコマンドライン引数といいます。この場合は、文字列であって数ではありませんがTMP_measがコマンドライン引数です。

- 降水量の略号「APCP」を指定して、降水量の日々変化のグラフを表示させてみましょう。
- 引数に、さらに「-a」を追加で指定して、積算値のグラフを表示させてみましょう。

プログラムts.pyの説明：プログラムの内容について、順次解説を加えます。

1～30行 プログラムの名称や使い方を書いた説明です。Pythonでは、「#」記号の後ろは無視されるので、#を書いてからコメントを記述します。プログラムを作成しているときには、動作確認のために特定の文を一時的に無効にしたいときがあり、そのような時にもこれを挿入することがあります。

32～37行 このプログラムでは、描画や日付計算など、素のPythonに組み込まれていない機能を使用するので、import文を使ってそれぞれの機能を提供するモジュールを組み込みます。このとき、「as」を使用して、モジュール名に別名をつけることができます。

```

32 import sys # モジュール属性 argv を取得ので導入
33 import numpy as np # 数値計算をするので導入
34 import datetime # 日付の計算をするので導入
35 import matplotlib.pyplot as plt # グラフを描くので導入
36 import matplotlib.dates as mdates # グラフに日付を書き込むので導入
37 import AMD_Tools as AMD # メッシュ農業気象データを扱うので導入
    
```

39行 この文を実行すると、引数をプログラム変数**argvs**に格納することができます。

```
39 argvs = sys.argv # コマンドライン引数を格納したリストの取得
```

○ IPython コンソールのプロンプトに「argvs」と打ち込み、変数の中身を表示させてみましょう。

40行 関数len()は、括弧の中の変数が保有する配列要素の個数を求めます。

```
40 argc = len(argvs) # 引数の個数
```

41～44行

条件分岐文(if文)で、ifとコロン「:」に挟まれた式が真であればインデントされている44行まで実行し、偽の場合はこれらの文をひとまとめに無視します。なお、44行目はプログラムを直ちに終了する命令文です。

argvsは、最初に入力した文字列を第一番の要素として返すことになっています。そのため、**argvs**の要素個数が1と言うことは、引数が0個ということです。

```
41 -if argc == 1:
42     print '表示する気象要素の記号(下記参照)を、引数で指定して下さい。'
43     print 'TMP_mea/TMP_max/TMP_min/RH/SSD/GSR/DLR/APCP/WIND'
44     sys.exit()
```

Python とインデント

文の行頭を字下げ(インデント)して、プログラムを見やすくすることは、多くのプログラム言語で慣用となっていますが、python ではループなどひとまとまりで扱うべき文をインデントそのもので表現します。たとえば、5の階乗($1 \times 2 \times \dots \times 5$)を計算するプログラムは、次のように書かれます。

```
k = 1
for i in [1, 2, 3, 4, 5]:
    k = k * i
print k
```

この例では、繰り返し計算の中身は3行目だけで、これが5回繰り返されてから4行目に移ります。どこからどこまでを繰り返すかは同じインデントがどこからどこまで施されているかで決まります。「end for」など、ループの終わりを示す文はありません。従って、次のように、4行目をインデントすると、画面には、計算途中の値が5回表示されるようになります。

```
k = 1
for i in [1, 2, 3, 4, 5]:
    k = k * i
    print k
```

インデントには半角空白とタブが使用できますが、両者は絶対に混用しないようにします。両者が混ざっていると、人の目からは同じインデントでありながら Python には異なるインデントと見なされ、原因不明のエラーに悩まされることになるので、気をつけてください。

一方で、python は括弧の途中では自由に改行やインデントができます。たとえば、関数にたくさんの引数があり、一行で書くととても長くなるようなとき、見やすくなるよう、括弧のなかの適当な場所で改行し字下げをすることができます。

```
T0, tim, lat, lon, nam, uni = AMD.GetData(element, area,
                                         timedomai n, lal odomai n, cli=1, namuni=1)
```

- 47行 日々変化グラフの日付範囲を「yyyy-mm-dd」の形式の文字列で指定します。年を跨る指定もできます。
- 49行 データを取得する地点が属するデータ領域名を文字列で指定します。ここで指定しているのは講習会会場の福山プラザホテルが位置する場所の緯度経度で、この地点は「Area4」に含まれています。
- 50行 グラフを作成する地点の緯度と経度を数値で指定します。指定する場所が点の場合は同じ緯度と同じ経度を繰り返し指定します。このプログラムでは、特定の1メッシュにおける気象値を使用しますが、面の場合には緯度と経度の範囲を指定します。

```
47   timedomain = [ '2014-01-01', '2014-12-31' ]
48
49   area = 'Area4'
50   lalodomain = [ 34.486, 34.486, 133.370, 133.370]#福山プラザホテル
```

- 54行 37行で読み込んだAMD_Toolsモジュールの中の関数GetData()を用いてメッシュ農業気象データを配信サーバーから読み込みます。47-50行で指定した、期間や緯度経度などの情報がこの関数に与えられています。

関数は、通常、気象データ、時間グリッド、緯度グリッド、経度グリッドの4つのデータを左辺に返しますが、「namuni=1」というオプション引数が含まれていると、返されるデータに気象要素の正式名と単位の2つが加わり合計6個になります。これらは、左辺に用意された6つの変数に順に格納されます。

```
54   T1, tim, lat, lon, nam, uni = AMD.GetData(args[1], area, timedomain,
      lalodomain, namuni=1, url="./AMD")
```

- IPython コンソールのプロンプトに「lat」や「tim」と打ち込み、変数の中身を表示させてみましょう。

- 55行 関数GetData()は、引数の中に「cli=1」というオプション引数が含まれていると、平年値を返します。また、「namuni=1」というオプション引数が含まれていないので、返されるデータは4種類です。これらを受け取るために、左辺には4個の変数が並んでいます。

```
55   T0, tim, lat, lon = AMD.GetData(args[1], area, timedomain, lalodomain, cli=1,
      url="./AMD")
```

58,59行 大括弧の中のコロン(:)は「すべての要素」を意味します。メッシュ農業気象データは、時間、緯度、経度からなる3次元「空間」に並べられたデータなので、一般には、何グリッド目の時間、何グリッド目の緯度、何グリッド目の経度という3種類の添え字を指定して気象データを特定します。しかし、このプログラムでは単一メッシュのデータしか扱わないので、緯度と経度についての添え字を識別する必要がありません。これらの文は、3次元の体裁で得たデータを1次元の体裁に変更する文です。

これは、建物が1つしかないアパートに「1号棟」という番地情報が不要なのと似ています。

```
58 T1 = T1[:,0,0]
59 T0 = T0[:,0,0]
```

61行 気象データを積算するかしないかを判断するif文です。引数に「-a」が追加指定されたときは、引数の要素の数は3であり、かつ、3個目(pythonは0から数える約束)の要素が「-a」であることを条件としています。この条件が成り立つ場合は63行以下を処理します。

63行 気象データを積算する繰り返し計算を指示する文(for文)で、この文よりもう一段深いインデントの部分の繰り返しを繰り返します。for文にある「in」の後ろには、数字や文字の列(Pythonではリストと呼びます)を置く決まりになっていて、これらのリストが順に変数iに読み込まれてインデント部分が繰り返されます。「range()」は関数で、引数の数の連番のリストを作り出します。注意しなければならないのは、関数range()は、引数の一つ手前の数までしか連番を作らないことです。

○ IPython コンソールのプロンプトに「range(10)」、「range(1,10)」と打ち込んでみましょう。

64行 繰り返し(for文)の指定により、まず、日付1の気象データに日付0のデータが加えられて日付1に格納し直されます。次の繰り返しでは、日付2のデータに(日付0と日付1の和となっている)日付1のデータが加えられて日付2のデータとして格納されます(つまり日付0、日付1、日付2の和が格納される)。同様にしてその次の繰り返しでは、日付3のデータに日付0~3までの和が格納しなおされます。この繰り返しにより、積算データが作り出されます。

```
61 - if argc == 3 and argvs[2] == '-a':
62     #   for i in range(T0.shape[0]):
63     -   for i in range(len(T0)):
64         T0[i] = T0[i] + T0[i-1]
65         T1[i] = T1[i] + T1[i-1]
```

Pythonでの作図：プログラムts.pyの68行～94行は、気象データと平年値データをグラフにする部分です。この部分では、図を書く場所を用意し、データ目盛り線や目盛り数値を決め、データの折れ線を描き、プログラム実行日の丸印を打ち、グラフの上にタイトル文字を入れ、図をファイルや画面に出力しています。Pythonの描画モジュールmatplotlibはきわめて表現力に富み、かつ多機能ですが、その分、グラフ各要素を指定する文は複雑難解です。このため、ここでは大まかな機能を紹介するにとどめます。

- 70行 横12インチ縦4インチの作図領域を用意します。
- 72-78行 横軸の目盛りを日付で振ることを指定します。
- 80行 $T0 < T1$ 、すなわち、気象値が平年値よりも大きい部分を薄橙色に着色します。
- 81行 $T1 < T0$ 、すなわち、気象値が平年値よりも小さい部分を薄水色に着色します。
- 82,83行 $T1$ や $T0$ 、すなわち、気象値や平年値の折れ線を黒線で引きます。
- 85,87行 プログラムを実行した日の気象値の場所に赤丸を打ちます。
- 89行 横軸の見出しを書き込みます。
- 90行 気象要素の略号、名称、単位を縦軸の見出しに書き込みます。
- 91行 図のタイトルに、地点の緯度経度、作図年月日を書き込みます。
- 93行 図をpngファイルで出力します。

```
68 # 計算結果をグラフにします。-----
69 # ・領域の作成
70 fig = plt.figure(num=None, figsize=(12, 4))
71 # ・目盛の作成
72 ax = plt.axes()
73 xmajoPos = mdates.DayLocator(bymonthday=[1])
74 xmajoFmt = mdates.DateFormatter('%m/%d')
75 ax.xaxis.set_major_locator(xmajoPos)
76 ax.xaxis.set_major_formatter(xmajoFmt)
77 xminoPos = mdates.DayLocator()
78 ax.xaxis.set_minor_locator(xminoPos)
79 # ・データのプロット
80 plt.fill_between(tim,T1,T0,where=T0<T1,color='orange',alpha=0.5)
81 #平年値より高い部分を橙色に塗る
81 plt.fill_between(tim,T0,T1,where=T1<T0,color='skyblue',alpha=0.5)
82 #平年値より低い部分を水色に塗る
82 plt.plot(tim, T0, 'k', linewidth=0.3) #平年値の線
83 plt.plot(tim, T1, 'k') #今年の線
84 # ・「今日」印を付ける
85 p = datetime.datetime.today()
86 #今日の時刻オブジェクト
86 today = tim == datetime.datetime(p.year,p.month,p.day,0,0,0)
87 #今日の配列要素番号
87 plt.plot(tim[today], T1[today], "ro")
88 #今日に赤点を打つ
88 # ・タイトルの付加
89 plt.xlabel('Date')
90 plt.ylabel(args[1] + ': ' + nam + ' (' + uni + ')')
91 plt.title('N'+str(lalodomain[0])+' E'+str(lalodomain[2])+' ('+str(p.year)+'/'+
92 str(p.month)+'/'+str(p.day)+'')')
92 # ・図の保存
93 plt.savefig('Fig_'+args[1]+'.png', dpi=300)
94 plt.show()
```


結果のファイル出力：メッシュ農業気象データの処理結果を他のプログラムや表計算ソフトで使えるよう、配列変数に格納された数値を様々な形式のファイルで出力することができます。このプログラムでは、関数PutCSV_TS()を用いて計算結果をCSVファイルで出力しています。

97行 365行×1列の配列であるT0とT1を結合して365×2列の配列とします。

98行 平年値と気象値を日付とともにCSVファイルとして出力します。「AMD.PutCSV_TS」は、AMD_Tools.pyに書かれているPutCSV_TS()という関数を呼び出すことを意味しています。

```
96 # 計算結果をファイルに保存します。-----  
97 Table = np.array([T0,T1])  
98 AMD.PutCSV_TS(Table, tim, header='Date,Normal,Obs.')
```

データソースの切替え：このプログラムで使われている関数GetData()の引数には、「url='./AMD」が含まれています。このオプション引数は関数がデータを取得する場所を指定するために用います。「url='./AMD」は、作業フォルダ「PythonWorks」の中のサブフォルダ「AMD」の中に格納されるデータにアクセスすることを指示しています。関数GetData()はデフォルトで「メッシュ農業気象データ配信サーバー」からデータを取得するように作られているので、この引数を削除すると、このプログラムは、「メッシュ農業気象データ配信サーバー」からデータを取得してグラフを表示します。

○ 55行目の文から「url='./AMD」を削除して、最新のグラフを描いてみましょう。

4. 「メッシュ農業気象データ」の利用条件

「メッシュ農業気象データ」は、以下の条件のもと、登録した利用者者に無償で提供しています。「事前配布ファイルフォルダ」に申請書が納められているので、利用を希望される方は、利用条件に承諾の上、書面にて申請してください。

1. 「メッシュ農業気象データ」は、気象に関する農業専用の指標なので農業分野以外での利用はできません。
2. データ提供者は、利用者が「メッシュ農業気象データ」の利用によって生じた結果、ならびに、利用できないことによって生じた結果について、一切の責任を負いません。
3. 「メッシュ農業気象データ」を利用して作成した情報を販売することはできません。
4. 「メッシュ農業気象データ」の利用期間は、申請受理日からその年度の3月31日までとします。ただし、くりかえし再申請することができます。
5. 申請書に示す利用目的を越えて「メッシュ農業気象データ」を再配布することはできません。
6. 利用者は、利用期間の終了時に、別紙様式2により利用状況を報告することとします。
7. 中央農研が配布する「メッシュ農業気象データ」の利用に関するプログラムは"GNU General Public License" (注)に従います。改変や再配布の際にはこれを遵守してください。

実習3 処理結果のGISへの取り込みと利用

この実習では、無料のGISソフトウェアQGIS2.6を利用して、メッシュ農業気象データをマップ表示する方法と、他のGISデータと併用して同データを高度に利用する方法を身につけます。

1. 使用するデータおよび環境

- QGIS2.6がインストールされているWindows PC。
(QGIS自体は各種OSに対応しているので、フォルダ構成等、適宜読み替えて使用可能)
- c:\¥GIS_DATAフォルダを作成し、以下のファイル群を置く(事前配布ファイルをディレクトリ毎コピーする。
 1. result.csv (メッシュ農業気象データ出力)
 2. mesh3R.dbf, mesh3R.shp, mesh3R.shx (1kmメッシュシェープファイル)
 3. KS-META-N03-14_140401.xml, N03-14_140401.dbf, N03-14_140401.prj, N03-14_140401.sbn, N03-14_140401.sbx, N03-14_140401.shp, N03-14_140401.shx, N03-14_140401.xml (行政界シェープファイル)

本資料執筆時の各ファイルの入手先は以下のとおりである。

QGIS : <http://qgis.org/ja/site/>。

1. : 本講習にて作成。
2. : http://www.gis-okinawa.jp/GIS_data/mesh23/。
3. : <http://nlftp.mlit.go.jp/ksj/gml/datalist/KsjTmplt-N03.html>。

2. 背景

地理情報システム (GIS) の分野では近年、無料ソフトの進化が目覚しく、費用負担なく高度な利用が可能となっている。その結果、地方自治体、JA、普及センターでは、管轄地域内の圃場図や作付け状況など、各種情報を無料ソフトにて管理し始めている。メッシュ農業気象データは、国土数値情報3次メッシュ (1kmメッシュ) 形式の位置情報を持つデータであるため、広域的・俯瞰的にデータを眺めるだけではなく、GISシステム上の圃場情報との結合により、さらなる活用が期待できる。そこで本講習では、無料ソフトの1つであるQGISによるメッシュ農業気象データの利用について実習する。有料ソフトを含む他のGISソフトでも、本講習のデータが読み込めれば、同様の利用が可能であると思われる。

なお、本講習以外の無料ソフトの操作法については、先駆者がボランティアで多数の情報をインターネット上で公開しているので、困ったときには検索で解決することがある。また、OSGeo財団日本支部の主催によるFree Open Source Software for Geospatial (FOSS4G)など、講習会 (有料) も開催されている。(13.参考資料を参照)

3. QGISの起動

正常にインストールされていれば、Windowsのデスクトップ画面上にショートカットアイコンがあるので、これを使用して起動する。(スタートメニューから等、それぞれのOSでのソフトウェア起動方法によって起動する。)

起動後、QGISチップス! が画面表示されるが、OKする。



図3-1. Windows7デスクトップ上にあるQGIS2.6の起動アイコン

4. 1kmシェープファイルの読み込み

シェープファイルとは聞きなれない言葉だが、GISソフトで一般に使用されるファイル形式であり、ベクタデータである。

- レイヤ→レイヤの追加→ベクタデータの追加 (図3-2)

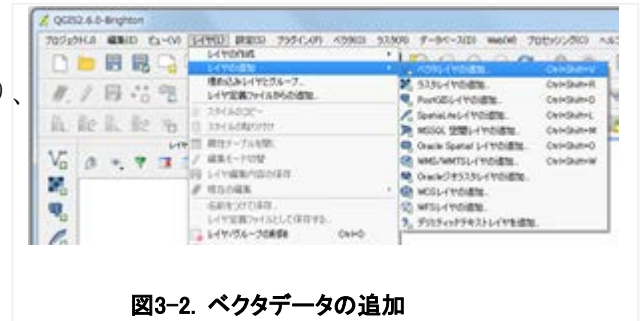


図3-2. ベクタデータの追加

- 「ベクタレイヤの追加」窓で
ソース→ブラウザ→mesh3R.shp
を選択し、「開く」をクリックする。
(図3-3)

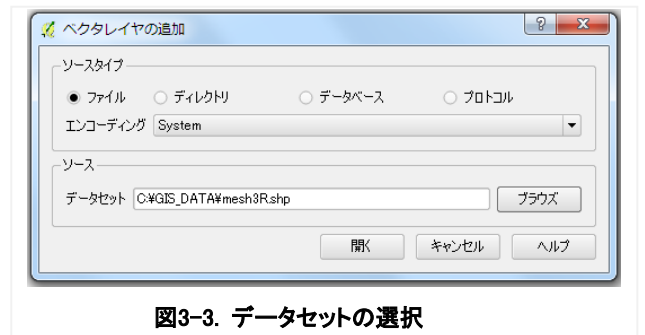


図3-3. データセットの選択

- 空間参照システムでJGD2000を選択し、「OK」をクリックする。
(図3-4)
(必要に応じて、使用するデータの座標系を指定する)

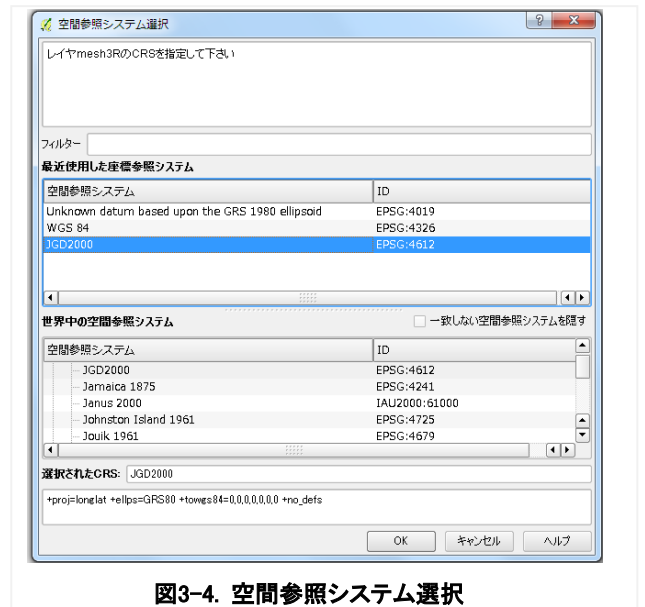


図3-4. 空間参照システム選択

- 1kmシェープファイルがうまく読み込めれば、QGIS中央窓に「日本列島」が描画される。また左のレイヤ窓に「mesh3R」の表記がされる。
(図3-5)



図3-5. QGIS中央部分の表示

- ・ QGISメニューの「虫眼鏡+」アイコンを選び、地図をクリックして拡大すると、タイル状の長方形が並んでいることがわかる。このタイルの1つ1つが、それぞれ1kmメッシュの位置と形を表している。(図3-6)

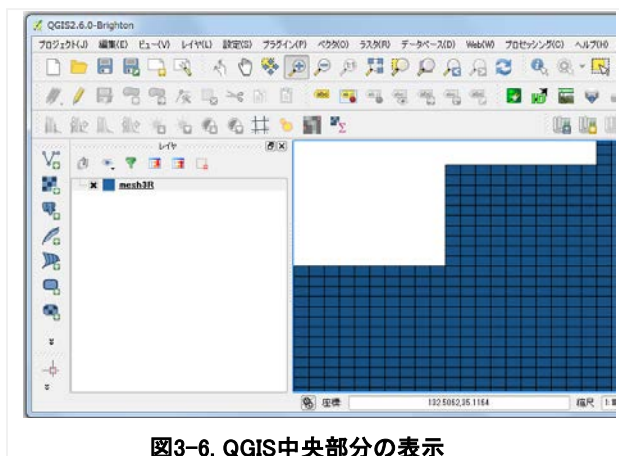


図3-6. QGIS中央部分の表示

5. メッシュ農業気象データの読み込みと結合

今回使用するメッシュ農業気象データresult.csvは、csv形式のテキストデータであり、1行目がヘッダ、2行目以降がデータであり、順にメッシュID、比、2014年日照時間、平年日照時間が並んでいる。

- ・ レイヤ→レイヤの追加→デリミティッドテキストレイヤを追加 (図3-7)

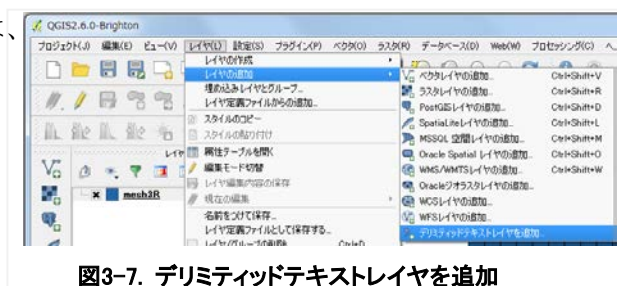


図3-7. デリミティッドテキストレイヤを追加

- ・ 「デリミティッドテキストファイルからレイヤを作成」で参照→result.csvを指定→ジオメトリなし(属性のみのテーブル)を選択→「OK」をクリックする。

(図3-8)

するとレイヤ窓に「result」の文字が現れ、読み込みを確認できる。

(その他、csvtファイルを使う方法もある)

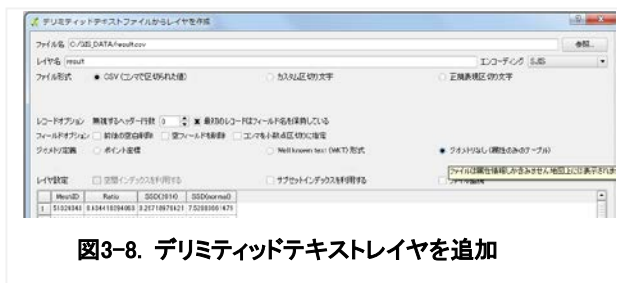


図3-8. デリミティッドテキストレイヤを追加

- ・ さて、レイヤ窓でmesh3Rを右クリックし、プロパティを選ぶ(図3-9)。レイヤプロパティ→結合で「ベクタ結合の追加」窓が表示される。

(図3-10)

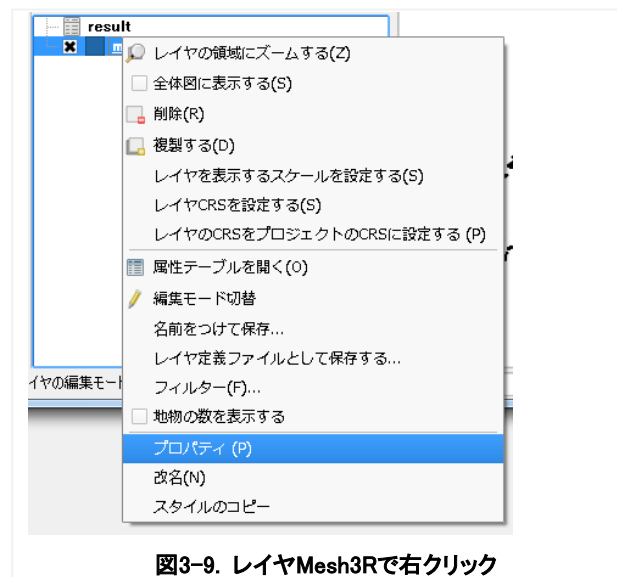


図3-9. レイヤMesh3Rで右クリック

結合するレイヤにresult、結合フィールドにMeshID、ターゲットフィールドにidを選び、「OK」をクリックする。次にレイヤプロパティ窓で「適用」「OK」をクリックする。Mesh3Rのidとresult.csvのMeshIDは、いずれも3次メッシュコードであり、両者の同じ数字が結合されることにより、メッシュ農業気象データがMesh3Rの位置情報にヒモつけされる。なお、もしデータがなければ、そこは空欄となる。



図3-10. レイヤMesh3Rで右クリック

6. メッシュ農業気象データの階級表示

再びMesh3Rのレイヤプロパティに戻る。スタイル→共通シンボル とは全てのデータ共通表示の意味なので、(図3-11)

[共通シンボル]ボタンの右の「▼」印を押して[段階に分けられた]に変更、

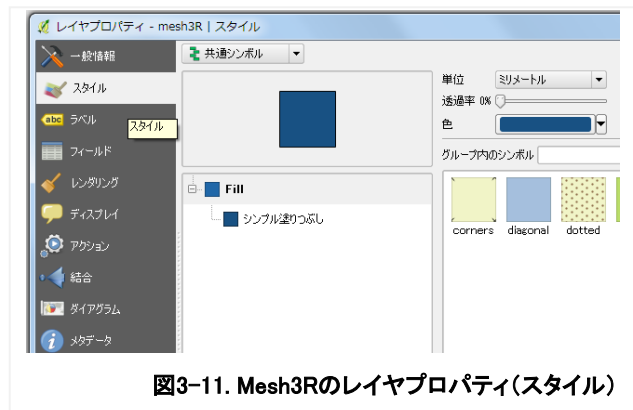


図3-11. Mesh3Rのレイヤプロパティ(スタイル)

カラムを「result_Ratio」に指定、モードを「プリティブレイク」に指定、「分類」「適用」「OK」とクリックする。(図3-12)



図3-12. Ratioの階級分類

すると、地図がresultの通り広島県のみとなり、ratioの指定区分通りの分類地図になる。(図3-13)

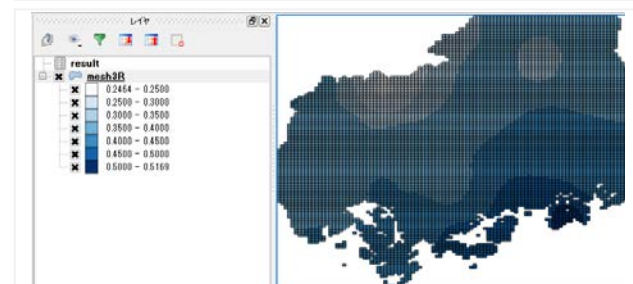


図3-13. Ratioの分類地図

ただしメッシュの境界線が邪魔なため、Mesh3Rレイヤプロパティ、スタイル→シンボル→境界線→ペン無し、ボーダー太さを「0」に指定する。
(図3-14)

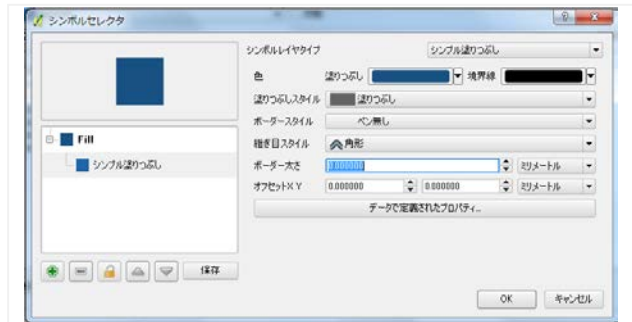


図3-14. 境界線指定変更

すると、境界線のないスッキリした分類地図にすることができる。
(図3-15)

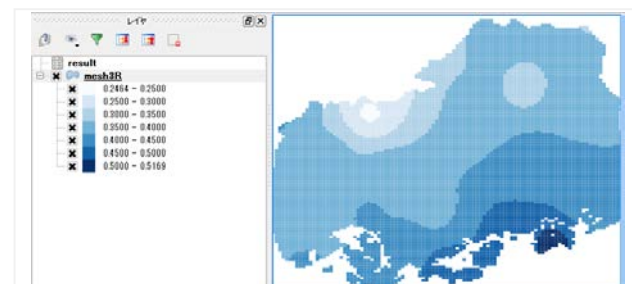


図3-15. Ratioの分類地図(境界線なし)

7. 行政界データの表示

今度は、行政界区分のシェープファイルを読み込む。ベクタレイヤの追加から、N03-14_140401.shp を指定する。
(図3-16 : 色は使用PCの設定ごとに異なる)

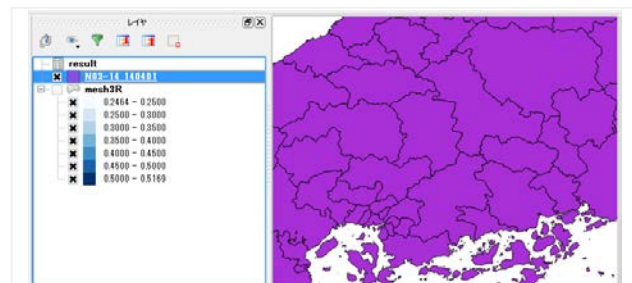


図3-16. 行政界の読み込み結果

N03-14_140401レイヤでも同様に、「レイヤプロパティ」→「スタイル」を表示する。今度は「共通シンボル」から「分類された」 「N03-001」を選択する。N03-001は都道府県界なので、都道府県ごとに色分けされる。しかし、希望するのは行政境界線（および海岸線）なので、N03-001のシンボルを「シンプル塗りつぶし」から「アウトラインシンプルライン」に変更する。
(図3-17)

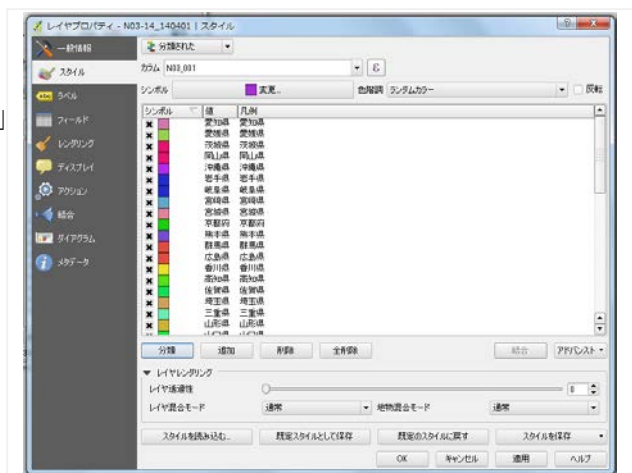


図3-17. 都道府県界の区分

すると行政界線のみになり、mesh3Rレイヤに行政界が入った図になる。

(図3-18 : 色は使用PCの設定ごとに異なる)

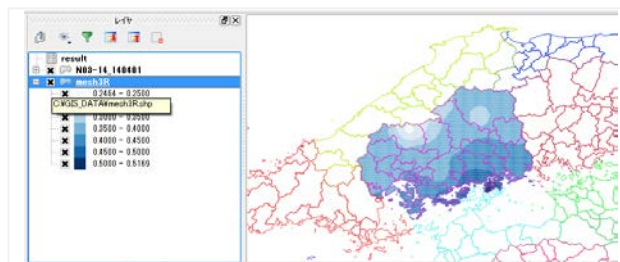


図3-18. 行政界区分線の入ったメッシュ分類図

ここでN03-14…レイヤを右クリックし、複製しておく。

N03-14…レイヤを右クリックし、「フィルター」を選択、クエリビルダ窓にて

「プロバイダ特有フィルター式」で
“N03_001” = ' 広島県' と入力、OKする。

(図3-19)

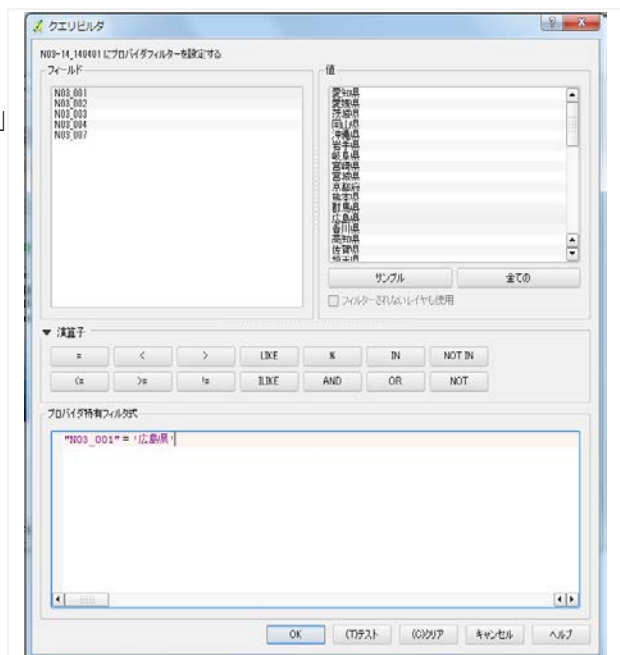


図3-19. フィルター式を入力

すると、広島県のみのが得られる。

ここで、N03-14…コピーレイヤ（全国表示）は表示させないため、ボックスのチェックを外しておく。(図3-20)

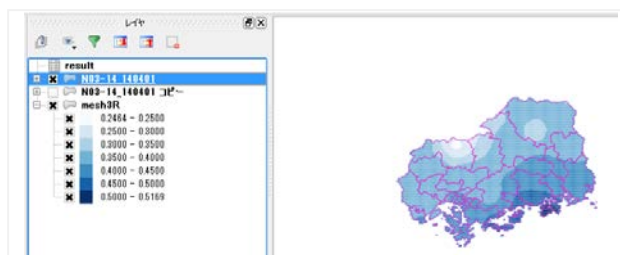


図3-20. 広島県のみのが得られる

コラム

本ソフトで多用している「レイヤ」とは、日本語では「層」の意味である。地層のように何層にも重ねることができる。このページの状態までたどり着いた場合、最下層に OpenLayers Plugin があり、その上にメッシュ農業気象データの階級分類した層があり、さらにその上に、N03-14…の行政界の層がある。私たちは、それらの層を上から眺めているので、レイヤの順序や透過性を調整することによって、最適な作図を得ることができる。なお、行政界は「シンプル塗りつぶし」から「アウトラインシンプルライン」に変更したため、塗りつぶし部分が透明となっており、透過性を調整する必要はない。

8. 地図や衛星画像の重ね合わせ

プラグインより、Web(W)→OpenLayers Plugin→

とする。

(図3-21)

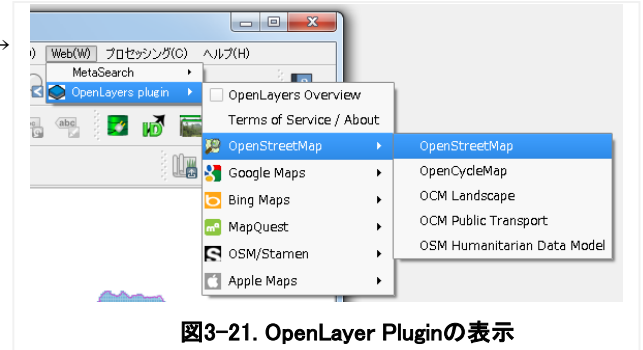


図3-21. OpenLayer Pluginの表示

すると地図や衛星画像と重ね合わせることができる。

(図3-22)

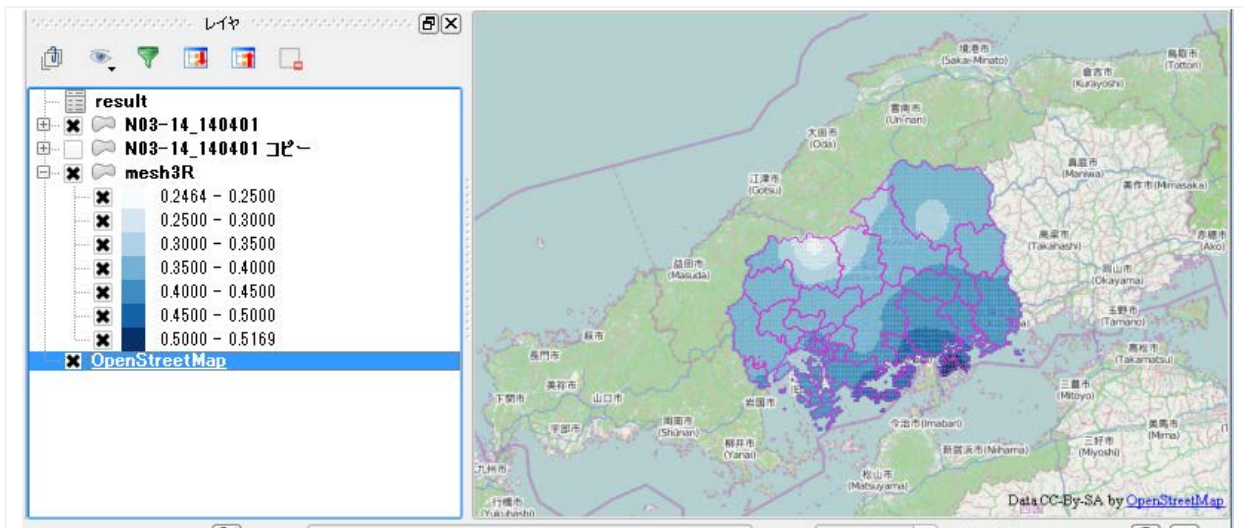


図3-22. OpenLayer Pluginによる地図の重ね合わせの例

その際、レイヤー順序や、レイヤプロパティ→スタイル内、レイヤレンダリングでレイヤ透過性を調整する必要がある。

(図3-23)



図3-23. レイヤ透過性の変更部

なお、OpenLayers Plugin は、各種 Web 地図サービスをそのまま取り込んで重ね合わせをしているため、印刷や外部公開に際しては、それぞれ提供元の引用ポリシーを理解し、適切に従う必要がある。

9. データの確認

OpenLayers Pluginによって、自分が調べたい地点を高精度で確認することが可能となる。すると、その地点におけるメッシュ農業気象データの範囲や数値を確認したいとの希望も出るだろう。レイヤにて、メッシュ農業気象データを包含する「mesh3R」を選択し、「ビュー」→「地物情報表示」とした上で希望するメッシュ部分をクリックすることにより、左下画面に当該メッシュのデータを表示することができる。(図3-24)

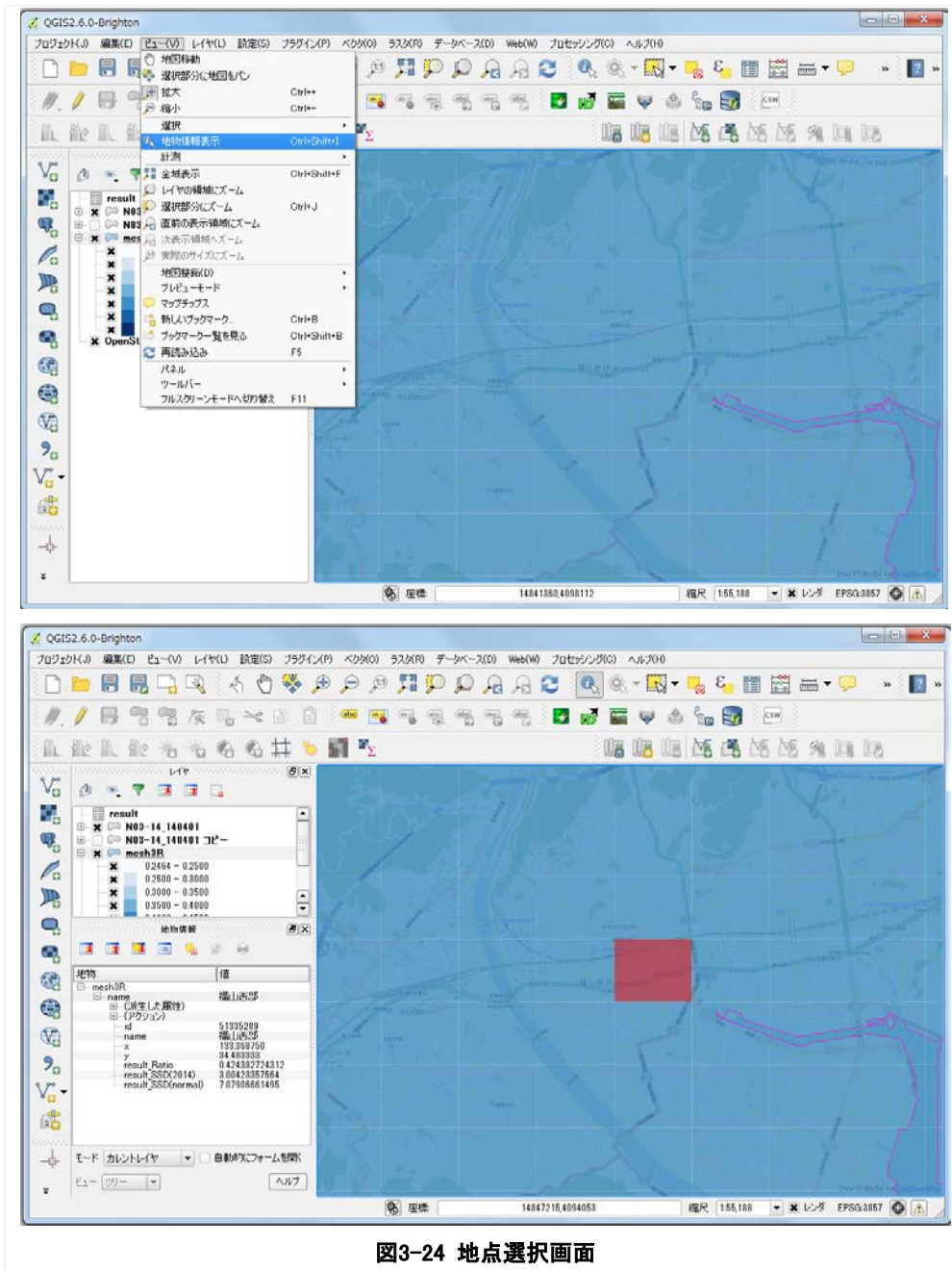


図3-24 地点選択画面

10. 図化

図化するためには、プリントコンポーザを使用する。「プロジェクト」→「新規プリントコンポーザ」を選択する。

(図3-25)

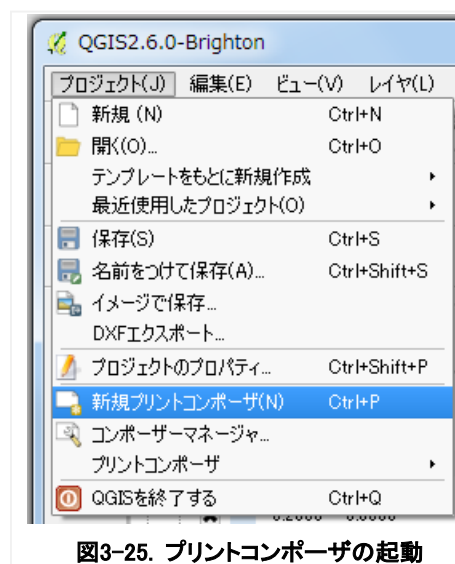


図3-25. プリントコンポーザの起動

すると、コンポーザ窓が開くので「レイアウト」→「地図を追加」を選択する。

(図3-26)

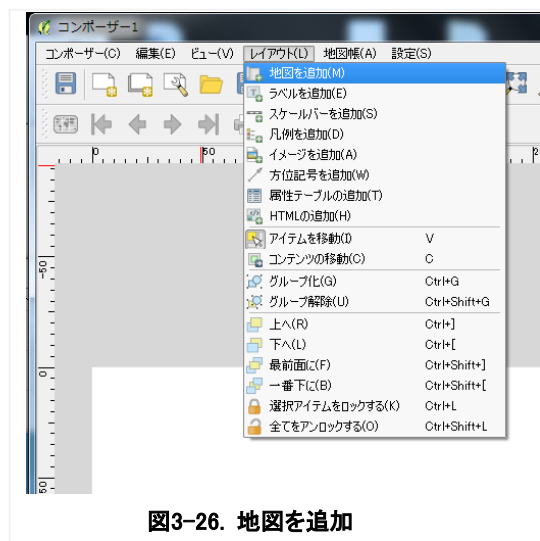


図3-26. 地図を追加

追加する領域をドラッグで示すと、地図画面が転載される。

(図3-27)

先に小さめの長方形で貼りこんだ後、長方形のサイズ拡大で大きめの図形を得ることもできる。「ビュー」→「更新」で再描画。

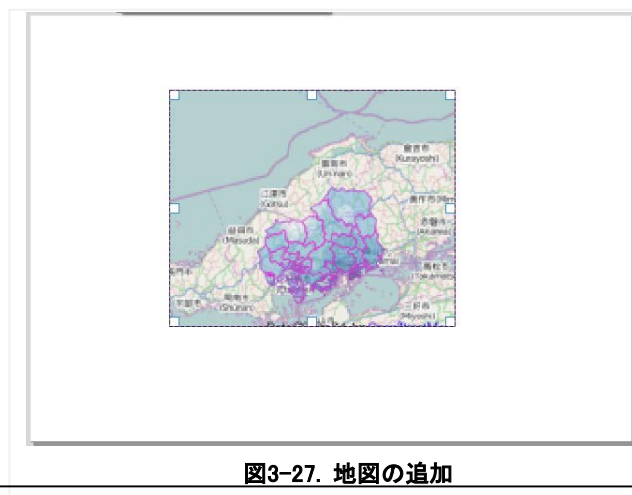


図3-27. 地図の追加

- ・「ラベルを追加」、「スケールバーを追加」
- ・「凡例を追加」では「アイテムプロパティ」
→「凡例アイテム」→右下緑逆三角アイコン
- ・「地図コンテンツによる凡例のフィルター」
- ・「イメージを追加」で「アイテムプロパティ」
→「検索ディレクトリ」→「方位記号」を表示。
等で装飾できる。（図3-28）

これらの作業により作成した図は、
「コンポーザ」→「画像としてエクスポート」、
「PDFとしてエクスポート」、「印刷」
等の出力が可能である。

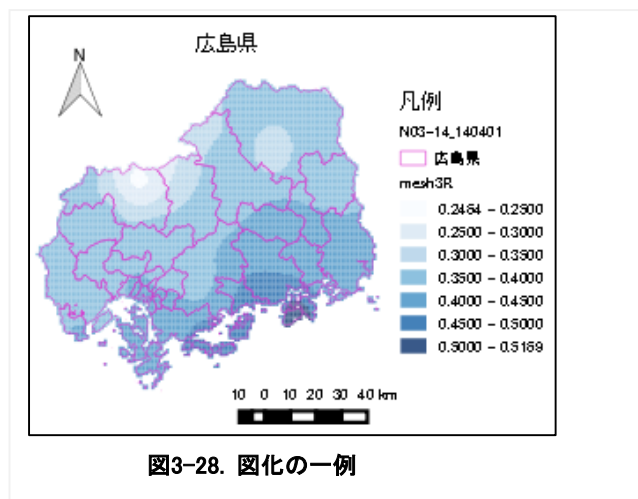


図3-28. 図化の一例

11. 作業記録の保存

作業した結果は、QGIS窓→「プロジェクト」→
「名前をつけて保存」で保存できる。

（図3-29）

ただし、QGISで取り込んだデータを保存するの
ではなく、それぞれのデータの関係性を保存するの
みであるため、元データの消去や移動によって再
現できなくなることがある。

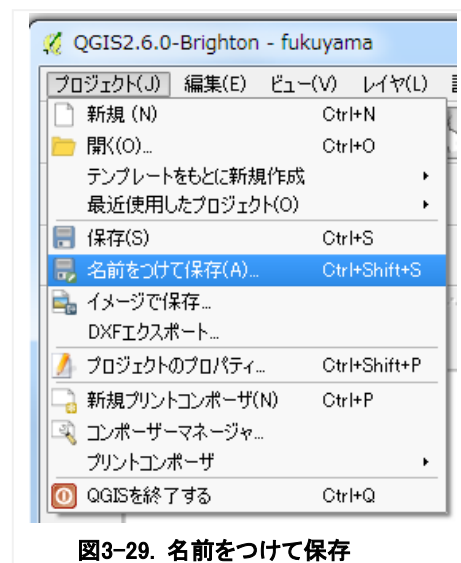


図3-29. 名前をつけて保存

12. 参考資料（順不同）

各種データ

- ・国土数値情報ダウンロードサービス

<http://nlftp.mlit.go.jp/ksj/index.html>

- ・地球地図日本

http://www.gsi.go.jp/kankyochiri/gm_jpn.html

- ・沖縄GIS研究室

http://www.gis-okinawa.jp/GIS_data/mesh23/

- ・Natural Earth

<http://www.naturalearthdata.com/>

- ・ コンサベーションGISコンソーシアムジャパン

http://cgisj.jp/download_type_list.php (旧日本測地系メッシュシェープファイル?)

- ・ 地図で見る統計 (統計GIS)

<http://www.e-stat.go.jp/SG1/estat/toukeiChiri.do?method=init>

- ・ 地理情報リンク

http://www.geocities.jp/rutile_67/g-link.htm

- ・ まとめ

<http://matome.naver.jp/odai/2135883023685089601?&page=3>

QGIS

- ・ OSGeo財団 (FOSS4G講習会情報)

<http://www.osgeo.jp/>

- ・ QGISマニュアル初級編

<http://www.slideshare.net/yasutofurukawa/qgis-handson-2014ss-36489774>

- ・ QGIS入門

<https://sites.google.com/site/qgisnoiriguchi/>

- ・ QGISリンク集

<http://wiki.osgeo.jp/index.php?title=QGIS%E3%83%AA%E3%83%B3%E3%82%AF%E9%9B%86>

- ・ オープンソースGIS

<http://www.geopacific.org/opensourcegis>

農業GIS

- ・ Finds.jp (農研機構 近中四農研)

<http://www.finds.jp/index.html.ja>
