

Copyright 2021 気象ビジネス推進コンソーシアム

(C) 2021 WXBC

<利用条件>

本書は、本書に記載した要件・技術・方式に関する内容が変更されないこと、および出典を明示いただくことを前提に、無償でその全部または一部を複製、翻案、翻訳、転記、引用、公衆送信等して利用できます。なお、全体を複製、翻案、翻訳された場合は、本書にある著作権表示および利用条件を明示してください。

<免責事項>

本書の著作権者は、本書の記載内容に関して、その正確性、商品性、利用目的への適合性等に関して保証するものではなく、特許権、著作権、その他の権利を侵害していないことを保証するものでもありません。本書の利用により生じた損害について、本書の著作権者は、法律上のいかなる責任も負いません。

事前準備の確認

事前準備ガイドに従って「メッシュ気象データ分析チャレンジ！」の事前準備が終了したら、簡単なプログラムを実行してチェックしましょう。

1. wgrib2 の確認

まず、GRIB/GRIB2形式のデータファイルの利用ツール **wgrib2** が正しくインストールされているかどうかを確認します。Windows PC の方は直下のスクリプトを実行してください。Macの方は、その一つ下のスクリプトを実行してください。実行は、スクリプトが記されている灰色の領域をクリックして緑色の枠を表示させ、その状態でツールバーの **[> Run]** ボタンをクリックするか、**[Ctrl]** キーを押しながら **[Enter]** キーを押します。

以下のような表示が出力されたら、wgrib2は正しく設定されています。

```
1.183:0:vt=2018070121:1.5 m above ground:15 hour fcst:TMP Temperature [K]:
ndata=242905:undef=0:mean=296.075:min=278.445:max=303.39
grid_template=0:winds(N/S):
lat-lon grid:(481 x 505) units 1e-06 input WE:NS output WE:SN res 48
lat 47.600000 to 22.400000 by 0.050000
lon 120.000000 to 150.000000 by 0.062500 #points=242905
```

```
In [ ]: # Windos PC の方はこれを実行(このスクリプトをクリックしてから、ツールバーの[> Run]ボタン)
import subprocess
import os

wgrib2 = "c:/wgrib2/wgrib2" # Macの場合 wgrib2 = "~/work/grib2/wgrib2/wgrib2"
grdir = "jmadata/msm/2018/201807"
grfile = "Z__C_RJTD_20180701060000_MSM_GPV_Rjp_Lsurf_FH00-15_grib2.bin"
grpath = grdir + "/" + grfile
kwds = '-match "\.183:"' #文字列中に「」を入れたいときは、文字列を「'」で囲みます
rc = subprocess.run(f'{wgrib2} -V {kwds} {grpath}', # 詳細な情報を表示させるため
                    shell=True,
                    stdout=subprocess.PIPE,
                    stderr=subprocess.PIPE,
                    universal_newlines=True)
for line in rc.stderr.splitlines():
    print(line)
for line in rc.stdout.splitlines():
    print(line)
```

```
In [ ]: # Mac PC の方はこれを実行(このスクリプトをクリックしてから、ツールバーの[> Run]ボタン)
import subprocess
import os

wgrib2 = "~/work/grib2/wgrib2/wgrib2" # Windows の場合 wgrib2 = "c:/wgrib2/wgrib2"
grdir = "jmadata/msm/2018/201807"
grfile = "Z__C_RJTD_20180701060000_MSM_GPV_Rjp_Lsurf_FH00-15_grib2.bin"
grpath = os.path.join(grdir, grfile) # ←このようなやり方もあります

ncdir = "./nc"
if not os.path.isdir(ncdir): #NetCDFファイルの置き場が無ければ作る
    os.makedirs(ncdir)
ncpath = os.path.join(ncdir, grfile) + ".nc"

rc = subprocess.run(f'{wgrib2} {grpath} -netcdf {ncpath}',
                    shell=True,
                    stdout=subprocess.PIPE,
                    stderr=subprocess.PIPE,
                    universal_newlines=True)
for line in rc.stderr.splitlines():
    print(line)
for line in rc.stdout.splitlines():
    print(line)
```

2. ファイル変換の確認

次に、GRIB/GRIB2形式のデータファイルが、**wgrib2** でNetCDFファイルに正しく変換できるかを確認します。Windos PC の方は直下のスクリプトを実行してください。Macの方は、**wgrib2 = "c:/wgrib2/wgrib2"** の行を書き換えた上で実行してください。

以下のような表示が出力されたら、wgrib2は正しく設定されています。

```
<class 'netCDF4._netCDF4.Variable'>
float32 TMP_1D5maboveground(time, latitude, longitude)
_FillValue: 9.999e+20
short_name: TMP_1D5maboveground
long_name: Temperature
```

level: 1.5 m above ground
units: K
unlimited dimensions: time
current shape = (16, 505, 481)
filling on

```
In [1]: import subprocess
import os
import netCDF4

wgrib2 = "c:/wgrib2/wgrib2" # Macの場合 wgrib2 = "~/work/grib2/wgrib2/wg
grdir = "jmadata/msm/2018/201807"
grfile = "Z__C_RJTD_20180701060000_MSM_GPV_Rjp_Lsurf_FH00-15_grib2.bin"
grpath = os.path.join(grdir, grfile) # ←このようなやり方もあります

ncdir = "./nc"
if not os.path.isdir(ncdir): #NetCDFファイルの置き場が無ければ作る
    os.makedirs(ncdir)
ncpath = os.path.join(ncdir, grfile) + ".nc"

rc = subprocess.run(f'{wgrib2} {grpath} -netcdf {ncpath}',
                    shell=True,
                    stdout=subprocess.PIPE,
                    stderr=subprocess.PIPE,
                    universal_newlines=True)
ds = netCDF4.Dataset("./nc/Z__C_RJTD_20180701060000_MSM_GPV_Rjp_Lsurf_FH00-
ds.variables['TMP_1D5maboveground']
```

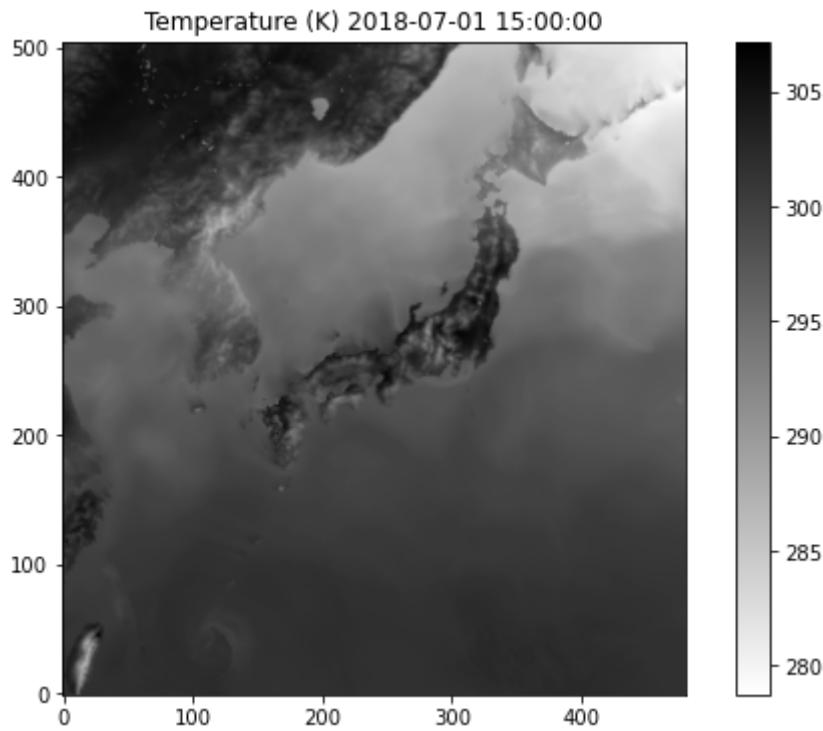
```
Out[1]: <class 'netCDF4._netCDF4.Variable'>
float32 TMP_1D5maboveground(time, latitude, longitude)
    _FillValue: 9.999e+20
    short_name: TMP_1D5maboveground
    long_name: Temperature
    level: 1.5 m above ground
    units: K
unlimited dimensions: time
current shape = (16, 505, 481)
filling on
```

3. wxbcgribモジュールの確認

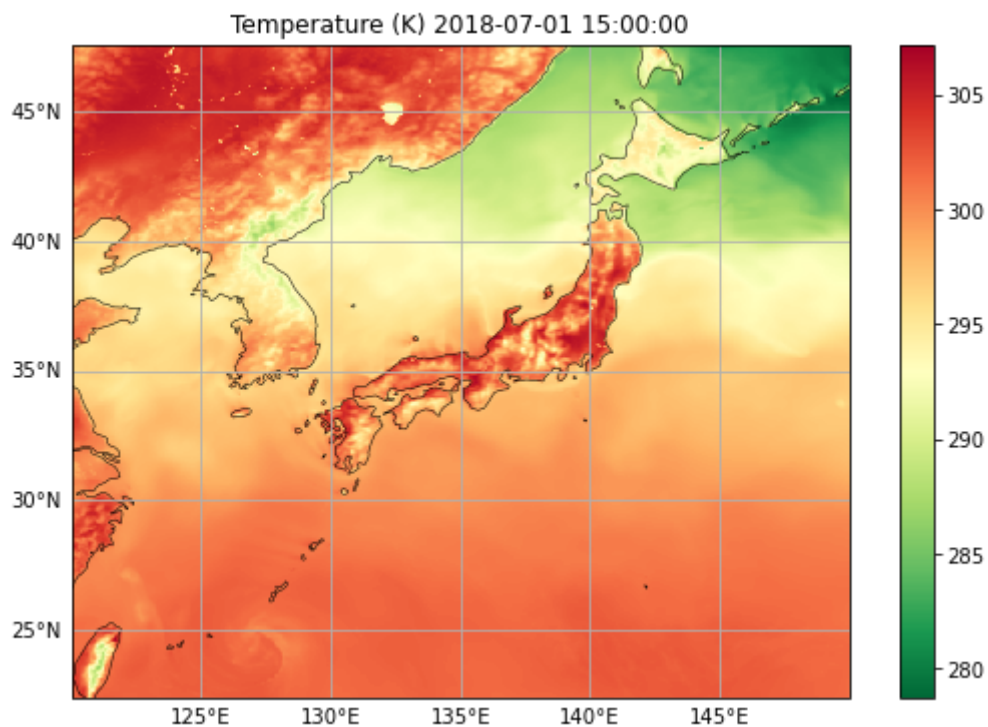
最後に、PythonでGRIB/GRIB2形式のメッシュデータをより簡単に操作可能となる **wxbcgrib** が正しく設定されているかを確認します。下のスクリプトを実行してください。最初に実行するときにだけ赤いテキストボックスで Warning が表示されますが、海岸線の地図をクラウド上からダウンロードする際に現れるものですので異常ではありません。

まず最初に白黒の気温分布図が表示され、暫くしてその下に、同じものが海岸線や緯線、経線とともにカラーで表示されたら **wxbcgrib** は正しく設定されています。

```
In [2]: from datetime import datetime
import wxbcgrib as wg
grdir = "./jmadata/msm/2018/201807/"
grfile = "Z__C_RJTD_20180701060000_MSM_GPV_Rjp_Lsurf_FH00-15_grib2.bin"
grpath = grdir + "/" + grfile
var = wg.DS_from_grnc(grpath, "TMP_1D5maboveground")
var.ql(cmapstr='gray_r')
map = var.yx(var.time[0], fig=True)
```



Name: Temperature Unit: K
Time: 2018-07-01 15:00:00
lat: 22.4 - 47.599999999999994 (0.05000000000000071)
lon: 120.0 - 150.0 (0.0625)
Shape: 16 x 505 x 481
FillVal: 9.999000260554009e+20 (# of FillVal: 0)



おわりに

このテストスクリプトで処理したのは、気象庁のメソ数値予報モデルGPVと呼ばれる、気象データです。それぞれのPythonスクリプトの内容については、「メッシュ気象データ分析チャレンジ!」で詳しく説明します。どうぞお楽しみに。