|  |
| --- |
| gradsパッケージ |
| 内容  [1 パッケージの概要 2](#_Toc501469999)  [2 サンプルプログラム 3](#_Toc501470000)  [2.1 example1.py（build\_temp\_64\_64\_with\_file()の例） 3](#_Toc501470001)  [2.2 example2.py（create\_temp\_64\_64\_with\_dataframe()の例） 5](#_Toc501470002)  [2.3 example3.py（create\_temp\_map()の例） 6](#_Toc501470003)  [2.4 example4.py（create\_temp\_map\_ex()の例） 7](#_Toc501470004)  [3 補足 8](#_Toc501470005) |

|  |  |
| --- | --- |
| パッケージの概要 | |
| パッケージ「grads」はGrADSを介して漁場推定システムの基礎となる海水温データの取得を容易にするために開発されたものである | |
| パッケージにはgrads\_retrieve.pyが用意されており、ここで定義された関数を用いて海水温データを取得する |  |

|  |  |
| --- | --- |
| サンプルプログラム | |
| ここでは、サンプルプログラム（example1~2）の解説を行いながら、パッケージの使用方法を学習する。 |  |
| example1.py（build\_temp\_64\_64\_with\_file()の例） | |
| 関数build\_temp\_64\_64\_with\_file()はgrads.grads\_retrieval.pyに含まれる関数で、つぎの３つの引数を取り、指定された位置を中心とした6.4°四方の0.1°間隔の位置における海水温の情報を収集する。  注１）GrADSは領域を低緯度から走査するので、データの並び一般的な画像ファイルとは異なる  注２）1レコードのデータ数は4096（＝64\*64）であり、これは2次元の領域データを1次元にreshapeしたものである。   |  |  | | --- | --- | | src | 入力ファイル | | grads\_ts\_ctl | GrADSコントロールファイル | | dst | 出力ファイル | | |
| 入力ファイルはpandasで読み込み可能なCSVファイルで、下図のような形式で日付、緯度経度、水深の情報が保存されている。 | |
| プロジェクトを実行すると、出力ファイルに以下のように海水温データが出力される | |

|  |
| --- |
| example2.py（create\_temp\_64\_64\_with\_dataframe()の例） |
| 関数create\_temp\_64\_64\_with\_dataframe()はgrads.grads\_retrieval.pyに含まれる関数で、つぎの2つの引数を取り、戻り値は収集した海水温情報を保持したDataFrameである。   |  |  | | --- | --- | | df\_date\_lon\_lat\_depth | 入力データ（DataFrame） | | grads\_ts\_ctl | GrADSコントロールファイル | |
|  |
| *# １．入力データの作成*  df\_date\_lon\_lat\_depth = pd.DataFrame(  [  [**"2012/7/15"**, 45, 180, 100],  [**"2012/7/16"**, 45.1, 181, 100],  [**"2012/7/17"**, 45.2, 182, 100],  [**"2012/7/18"**, 45.3, 183, 100],  [**"2012/7/19"**, 45.4, 184, 100]  ],  columns=[**"date"**, **"lat"**, **"lon"**, **"depth"**])  *# ２．海水温情報の取得* df\_temp\_64\_64 = create\_temp\_64\_64\_with\_dataframe(df\_date\_lon\_lat\_depth, **'/mnt/seadata/ts.ctl'**) | |

|  |
| --- |
| example3.py（create\_temp\_map()の例） |
| 関数create\_temp\_map()はgrads.grads\_retrieval.pyに含まれる関数で、つぎの8つの引数を取り、戻り値は検索結果を保持したDataFrameである。  領域は左下と右上の座標で指定する。   |  |  | | --- | --- | | date | 書式「年/月/日」で日付を表現する文字列 | | depth | 水深 | | lon\_min | 指定領域の経度の下限値 | | lat\_min | 指定領域の緯度の下限値 | | lon\_max | 指定領域の経度の上限値 | | lat\_max | 指定領域の緯度の上限値 | | pitch | サンプリング間隔 | | grads\_ts\_ctl | GrADSコントロールファイル | |
| 1999年6月1日の水深100mでの緯度35°～47°、経度147°～198°の領域における0.1°間隔の海水温の情報を取得する場合は、以下のようにプログラムを記述する  df = create\_temp\_map(**'1999/6/1'**, 100, 147.0, 35.0, 198.0, 47.0, 0.1, **'/mnt/seadata/ts.ctl'**) |

|  |
| --- |
| example4.py（create\_temp\_map\_ex()の例） |
| 関数create\_temp\_map\_ex()はgrads.grads\_retrieval.pyに含まれる関数で、つぎの7つの引数を取り、戻り値は検索結果を保持したDataFrameである。  領域は中心と幅・高さで指定する。   |  |  | | --- | --- | | date | 書式「年/月/日」で日付を表現する文字列 | | depth | 水深 | | lon | 指定領域の中心座標の経度 | | lat | 指定領域の中心座標の緯度 | | pitch | サンプリング間隔 | | row\_col | 出力結果の次元（戻り値のshapeはここで指定する） | | grads\_ts\_ctl | GrADSコントロールファイル | |
|  |

|  |
| --- |
| 補足 |
| ライブラリ内部では領域指定においてオフセットを加えている。  これは、GrADSの戻り値で不可解な値が含まれるのを回避するためである。  1つは、先頭行、列の値が実際より小さいことである。もう一つは、最終列から1列または2列の値がすべて「---」でパディングされている場合が確認されている。  これらを回避するために、指定領域よりも境界から5行5列（前後左右であるから10づつ）広くなるよう緯度経度を調整してGrADSでデータの検索を行い、その結果からオフセット分をトリムした値を結果として返している    イメージ図（青が指定領域で赤がオフセット） |