

R for Windows を使って netCDF

ファイルを CSV 形式に変換する手順

国立環境研究所気候変動適応研究センターでは「気候シナリオデータは netCDF 形式で提供されることが多いけど、Excel 形式(CSV 形式)でもらえませんか?」というリクエストを受けることがたびたびあります。実は R というフリーソフトを使えば、Windows 上で簡単に netCDF ファイルから CSV 形式のファイルに変換することができます。本書では

- (1) Windows の PC に R をインストールする方法
- (2) netCDF 形式データファイルを R で読む方法
- (3) ある地点のデータを CSV 形式のファイルとして出力する方法

を解説します。netCDF ファイルの読み方が分からなくて研究が進まない問題の解決に役立てば幸いです。

R とは

統計解析に特化したプログラミング言語であり、その実行環境を指す場合もあります。

Windows・Linux・Mac 向けに、オープンソースの実行環境ソフトウェア (コンソール) が提供されており、誰でも自由に使うことができます。ここでは Windows 向けの解説を行っていますが、Mac でも同じようにできると思われれます。

1. R for Windows のインストール方法について

【R for Windows のインストール】

「<https://cran.r-project.org/>」から、「Download R for Windows」をクリックする



CRAN
[Mirrors](#)
[What's new?](#)
[Task Views](#)
[Search](#)

About R
[R Homepage](#)
[The R Journal](#)

Software
[R Sources](#)
[R Binaries](#)
[Packages](#)
[Other](#)

Documentation
[Manuals](#)
[FAQs](#)
[Contributed](#)

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2020-02-29, Holding the Windsock) [R-3.6.3.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

(1) 次に「base」をクリック



CRAN
[Mirrors](#)
[What's new?](#)
[Task Views](#)
[Search](#)

About R
[R Homepage](#)
[The R Journal](#)

Software
[R Sources](#)
[R Binaries](#)
[Packages](#)
[Other](#)

Documentation
[Manuals](#)
[FAQs](#)
[Contributed](#)

R for Windows

Subdirectories:

[base](#)

[contrib](#)

[old contrib](#)

[Rtools](#)

Binaries for base distribution. This is what you want to [install R for the first time](#).

Binaries of contributed CRAN packages (for R >= 2.13.x; managed by Uwe Ligges). There is also information on [third party software](#) available for CRAN Windows services and corresponding environment and make variables.

Binaries of contributed CRAN packages for outdated versions of R (for R < 2.13.x; managed by Uwe Ligges).

Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the [R FAQ](#) and [R for Windows FAQ](#).

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

(2) 「Download R 3.1.2 for Windows」(最新版のリンク)をクリックすると、exe ファイルがダウンロードできます



R-3.6.3 for Windows (32/64 bit)

[Download R 3.6.3 for Windows](#) (83 megabytes, 32/64 bit)

[Installation and other instructions](#)

[New features in this version](#)

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the [md5sum](#) of the .exe to the [fingerprint](#) on the master server. You will need a version of md5sum for windows: both [graphical](#) and [command line versions](#) are available.

Frequently asked questions

- [Does R run under my version of Windows?](#)
- [How do I update packages in my previous version of R?](#)
- [Should I run 32-bit or 64-bit R?](#)

Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information.

Other builds

- Patches to this release are incorporated in the [r-patched snapshot build](#).
- A build of the development version (which will eventually become the next major release of R) is available in the [r-devel snapshot build](#).
- [Previous releases](#)

Note to webmasters: A stable link which will redirect to the current Windows binary release is [<CRAN MIRROR>/bin/windows/base/release.htm](#).

CRAN

[Mirrors](#)

[What's new?](#)

[Task Views](#)

[Search](#)

About R

[R Homepage](#)

[The R Journal](#)

Software

[R Sources](#)

[R Binaries](#)

[Packages](#)

[Other](#)

Documentation

[Manuals](#)

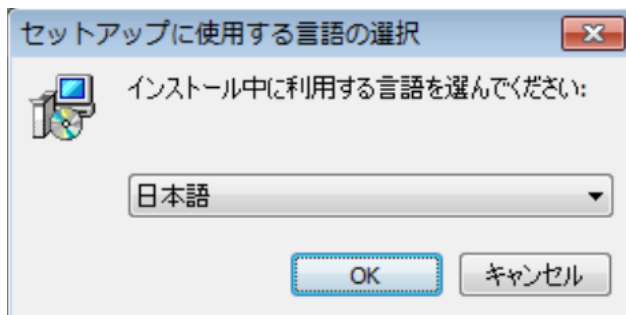
[FAQs](#)

[Contributed](#)

(3) ダウンロードした「R-3.1.2-win.exe」をダブルクリックして、インストールを実行します。

【R for windows の初期設定】

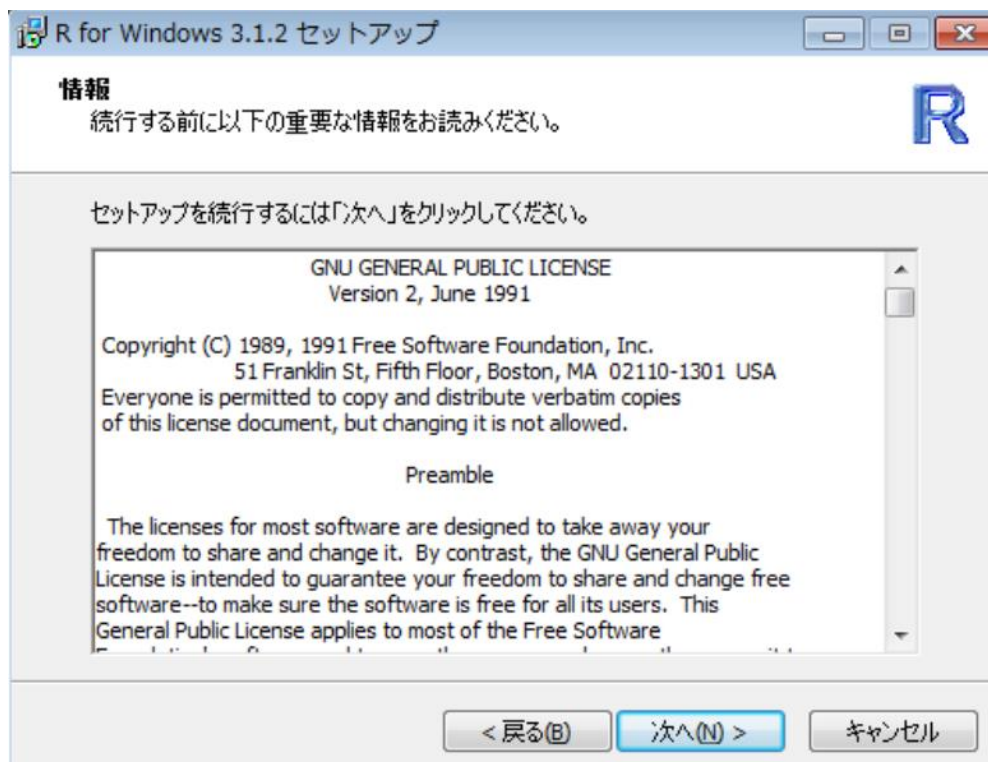
セットアップに使用する言語は「日本語」にします。



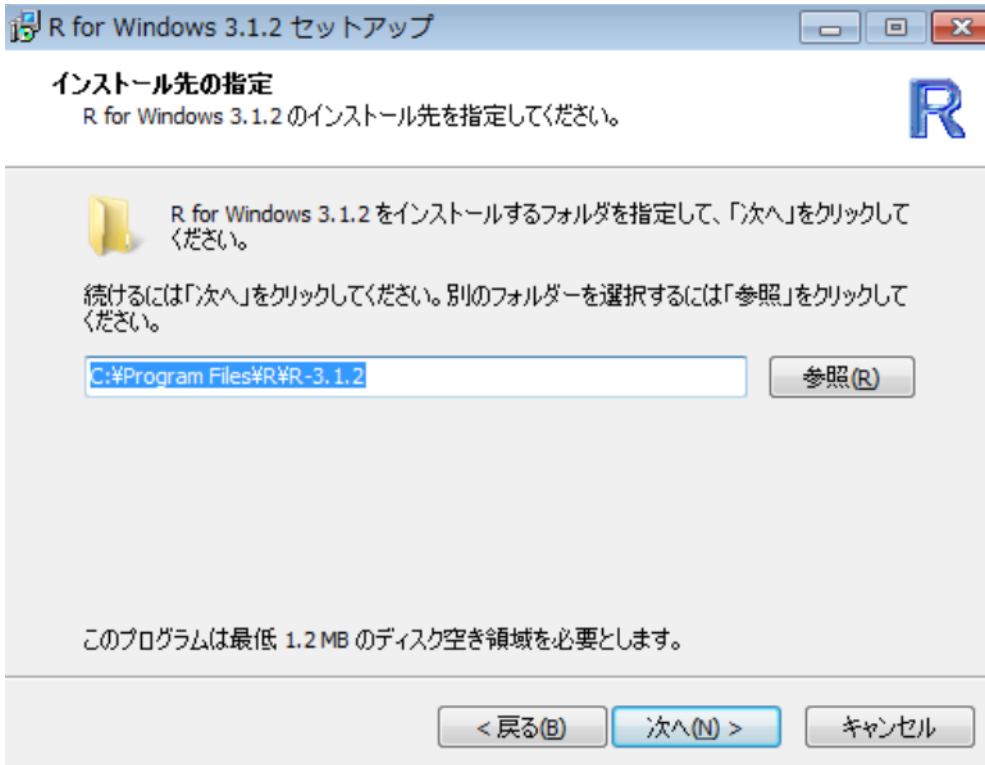
次へ



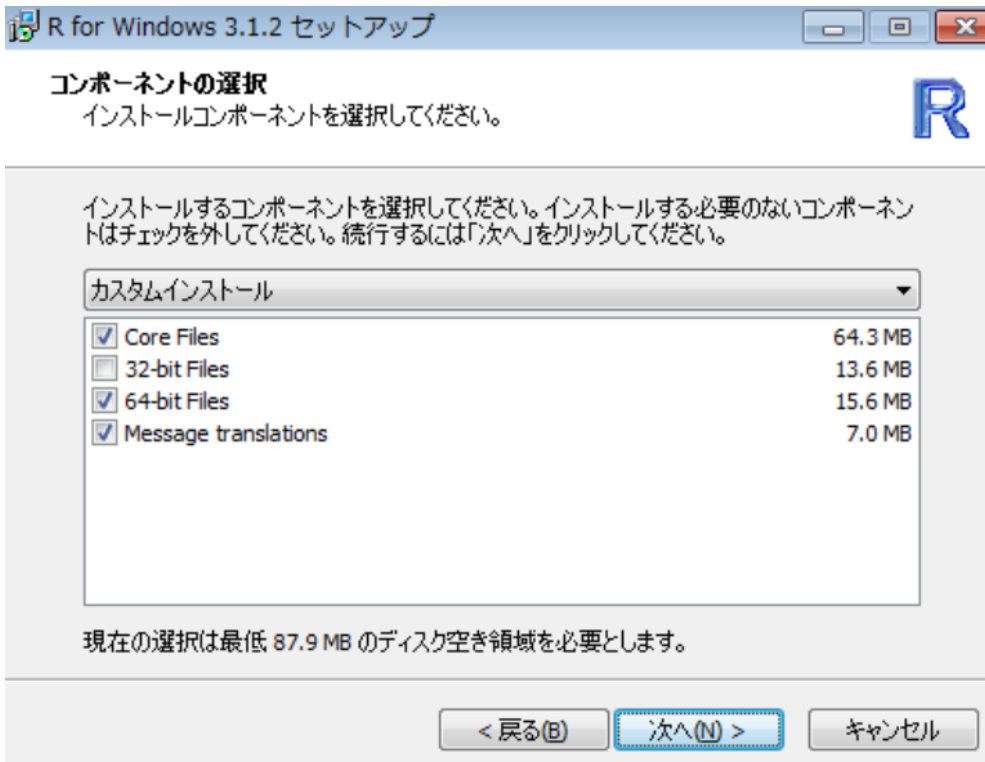
次へ



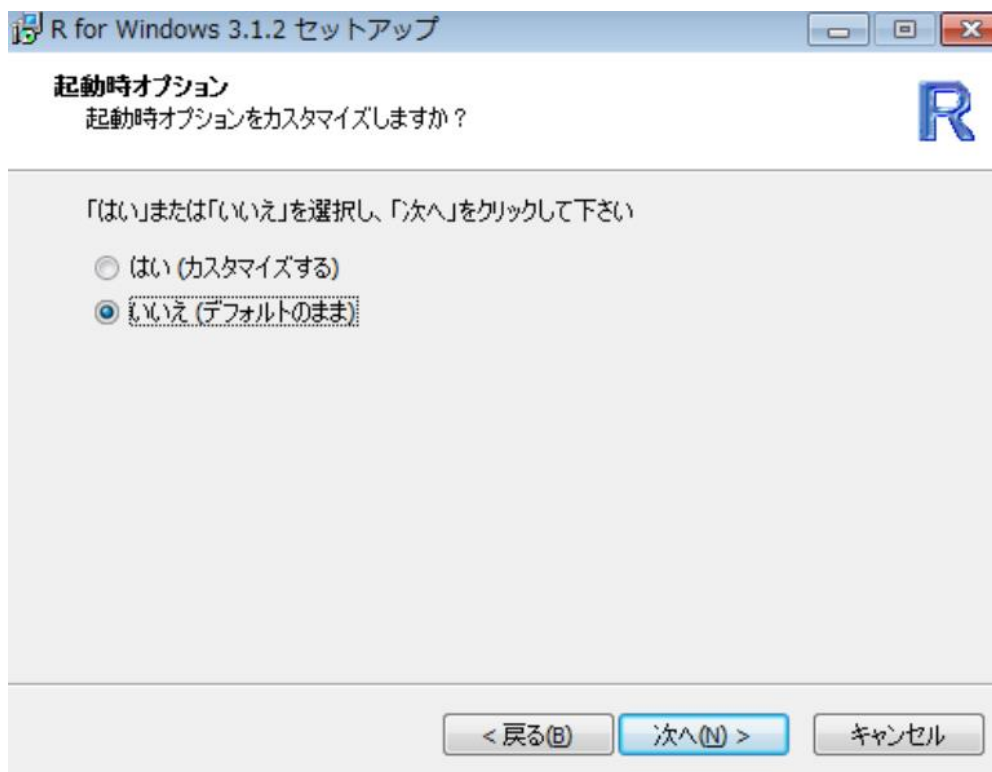
インストール先はそのまま（次へ）



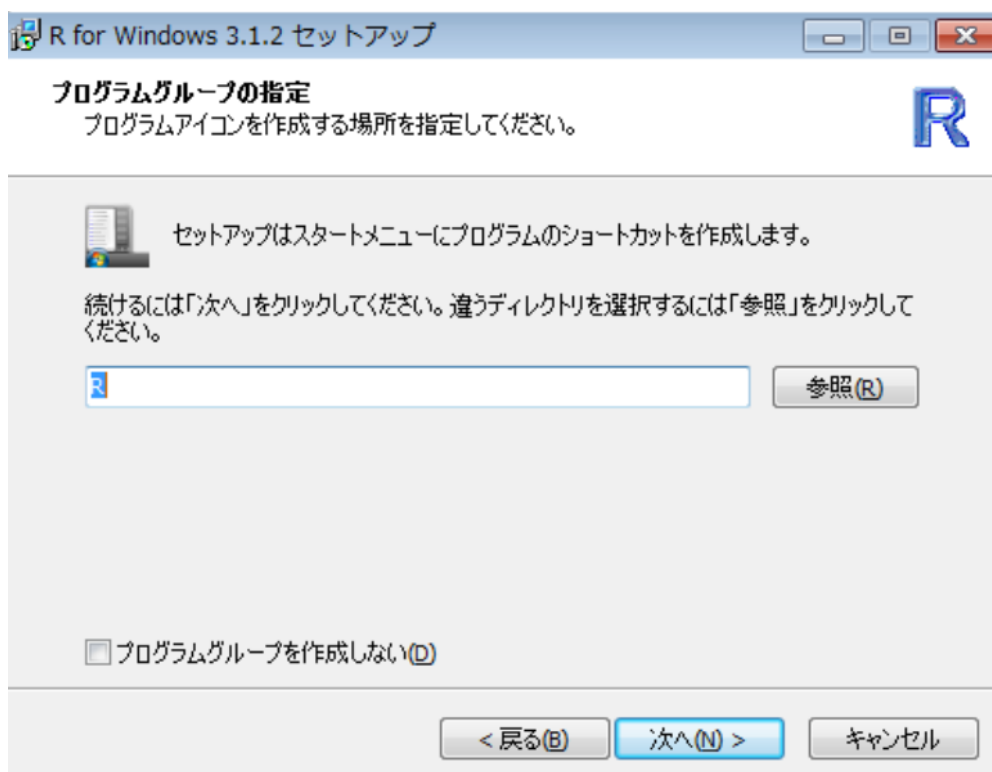
マシンが 64bit なので「32bit Files」のチェックは外しておきました。



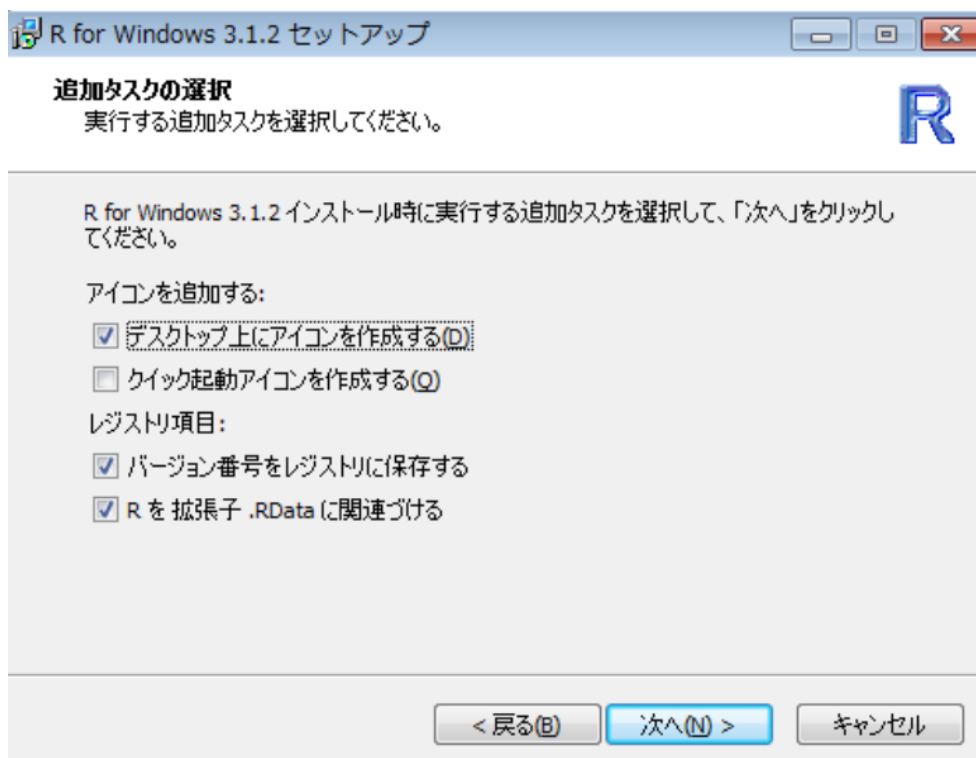
起動時オプションは「いいえ (デフォルトのまま)」で、変更せず



そのまま



そのまま



完了



【参考文献】

<https://akiyoko.hatenablog.jp/entry/2014/11/07/042801>

<https://code.i-harness.com/ja-jp/q/a6969c>

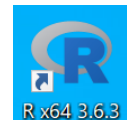
<https://qiita.com/IEOHS/items/eb6a5b0bb896ee024310>

2. R for Windows で netCDF データを CSV へ変換

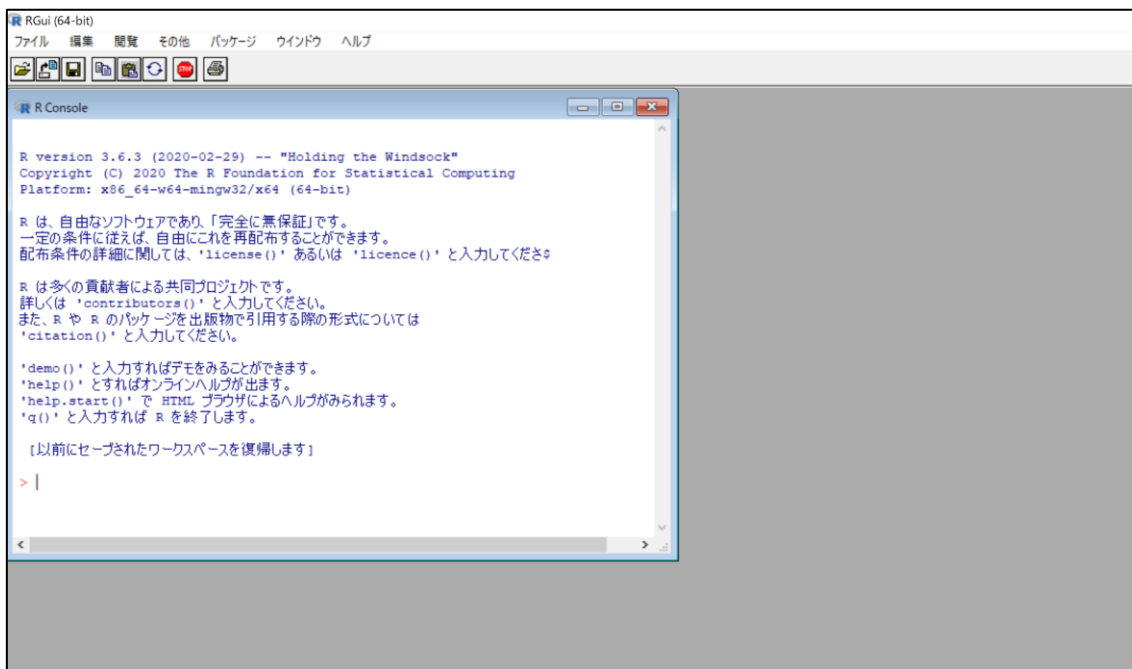
ここでは netCDF ファイルのデータを加工し、CSV ファイルとして出力する。

2.1 netCDF ファイルを入力する

(1) デスクトップ上で R for Windows のアイコンをダブルクリック

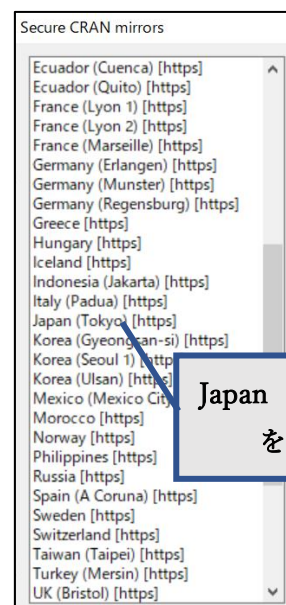


(2) 下の写真のような画面が出てくる



(3) 画面上部の「パッケージ」のタブをクリックし、「CRAN ミラーサイトの設定」をクリックする。

すると右図のような一覧が出てくるので、「Japan (Tokyo)」を選択する。



(4) R for Windows で netCDF データを扱うために、ncdf4 のパッケージをインストールする。

```
> install.packages("ncdf4")  
> library(ncdf4)
```

(5) Rで netCDF ファイルを開くには「nc_open()」というコマンドを使う。

```
> nc <- nc_open("C:/Users/Masumoto/Desktop/ccb_Amon_MIROC5.nc")
```

※nc_open()コマンドの () 内では入力する netCDF ファイルのディレクトリと名前を指定する。今回の例ではデスクトップに置いていた ccb_Amon_MIROC5.nc のファイルを入力した。また入力したいファイルのディレクトリを調べるためには、入力するファイルを右クリックして「プロパティ」をクリックすると、「場所」の欄にそのファイルのディレクトリが書かれている。(＼を/に置き換えることを忘れずに)

(6) print()コマンドで netCDF ファイルの中身を知ることができる。

(linux である ncdump -h と同じようなもの)

```
> print(nc)
```

実行すると下のような文がかかっている。

```
> print(nc)
File C:/Users/Masumoto/Desktop/ccb_Amon_MIROC5_abrupt4xCO2_rl0ilpl_210

4 variables (excluding dimension variables):
  double time_bnds[bnds,time]
  double lat_bnds[bnds,lat]
  double lon_bnds[bnds,lon]
  float ccb[lon,lat,time]
    standard_name: air_pressure_at_convective_cloud_base
    long_name: Air Pressure at Convective Cloud Base
    units: Pa
    original_name: CBASEP
    cell_methods: time: mean
    cell_measures: area: areacella
    history: 2011-12-28T08:30:11Z altered by CMOR: replaced mi
    missing_value: 1.00000002004088e+20
    _FillValue: 1.00000002004088e+20
    associated_files: baseURL: http://cmip-pcmdi.llnl.gov/CMIP
```

図中の「float」と書かれている箇所の右隣にあるものがこのファイルの変数である。今回の場合、「ccb」という変数が格納されている。また変数の右にあるもの(今回の場合、lon, lat, time)は、その変数の次元を示している。

2.2 データを加工する

(1) 抽出したいデータの変数を指定するために `ncvar_get()` コマンドを使う。

```
> var <- ncvar_get(nc, "ccb")
```

※ 「<-」で変数 `var` が定義され、そのなかに "ccb" のデータが入る。

(2) 抽出した変数の次元数を確認する → `dim()` コマンド

```
> dim(var)
[1] 256 128 63
```

※今回の場合、変数 (ccb) の次元数は lon(緯度情報)が 256 個、lat(経度情報)が 128 個、time(時間情報)が 63 個あることが確認できる。

(3) 次元を指定して、その時の変数の値を取得してみる。

例えば lon が 12、lat が 28、time が 55 の変数の値を取得してみると、

```
> var[12,28,55]
[1] 95885.64
```

と表示される

また lon が 12、lat が 28 におけるすべての time のデータを取得しようとすると、

```
> var[12,28,]
[1] 94462.98 96408.25 94840.66 96988.81 97758.14 96177.09 95172.75
[13] 93747.20 94800.75 96115.93 95623.91 97640.10 95381.67 95943.80
[25] 96656.59 96985.69 94777.03 96553.87 97246.79 96993.26 97083.07
[37] 96536.62 94992.66 95943.61 95939.80 95826.66 96514.73 96203.48
[49] 95444.99 96442.67 98262.78 95588.52 97427.27 97173.66 95885.64
[61] 96516.94 95899.20 97499.01
```

と表示される。

(4) 緯度、経度、時間の情報を確認する。

```
latarr <- ncvar_get(nc, "lat")
```

```
lonarr <- ncvar_get(nc, "lon")
```

```
timarr <- ncvar_get(nc, "time")
```

で緯度、経度、時間の情報を変数 `lonarr`, `latarr`, `timarr` に抽出する。

```
lon[12]
```

と打てば、12 番目の経度の値が表示される。

2.3 CSV ファイルを出力する

(1) まず CSV ファイルに出力したいデータを定義する。

今回は例として lon が 12、lat が 28 におけるすべての time の変数を取り出すことにします。(2.2 (3) で用いた例と同じ)

```
> T <- var[12,28,]
```

※ 何かを定義するときは 「<-」 という文字を使います。

今回の場合、lon が 12、lat が 28 におけるすべての time の変数を T と定義しています。

(2) (1) で定義した変数を CSV ファイルに出力する。

CSV ファイルとして出力するときには「write.csv()」というコマンドを用います。

```
> write.csv(T,file="C:/Users/Masumoto/Desktop/20200327.csv")
```

※今回の場合 (1) で T と定義した変数を CSV ファイルとして、PC のデスクトップに「20200327」という名前を付けて出力しています。

これで CSV ファイルが作成されます。

実際に作成したファイルを確認すると、、、

	A	B	C
1		x	
2	1	64856.31	
3	2	65129.14	
4	3	65460.86	
5	4	65720.44	
6	5	65832.02	
7	6	65683.73	
8	7	65428.35	
9	8	65420.95	
10	9	65896.97	
11	10	65484.12	
12	11	65444.87	

こんな感じになります。

A 列に time、B 列にその時の変数が書かれています。