

土地改良事業計画（排水）における 将来の降雨予測に基づく確率降雨量 算定マニュアル

令和 7 年 4 月

令和 7 年 10 月改定

農村振興局整備部設計課計画調整室計画基準班

目次

第 1 章 作業フロー	P 1
第 2 章 作業環境の構築	P 2
第 3 章 観測値の処理	P 5
第 4 章 実験データの処理	P17
第 5 章 過去実験データの処理	P25
第 6 章 将来実験データの処理	P38

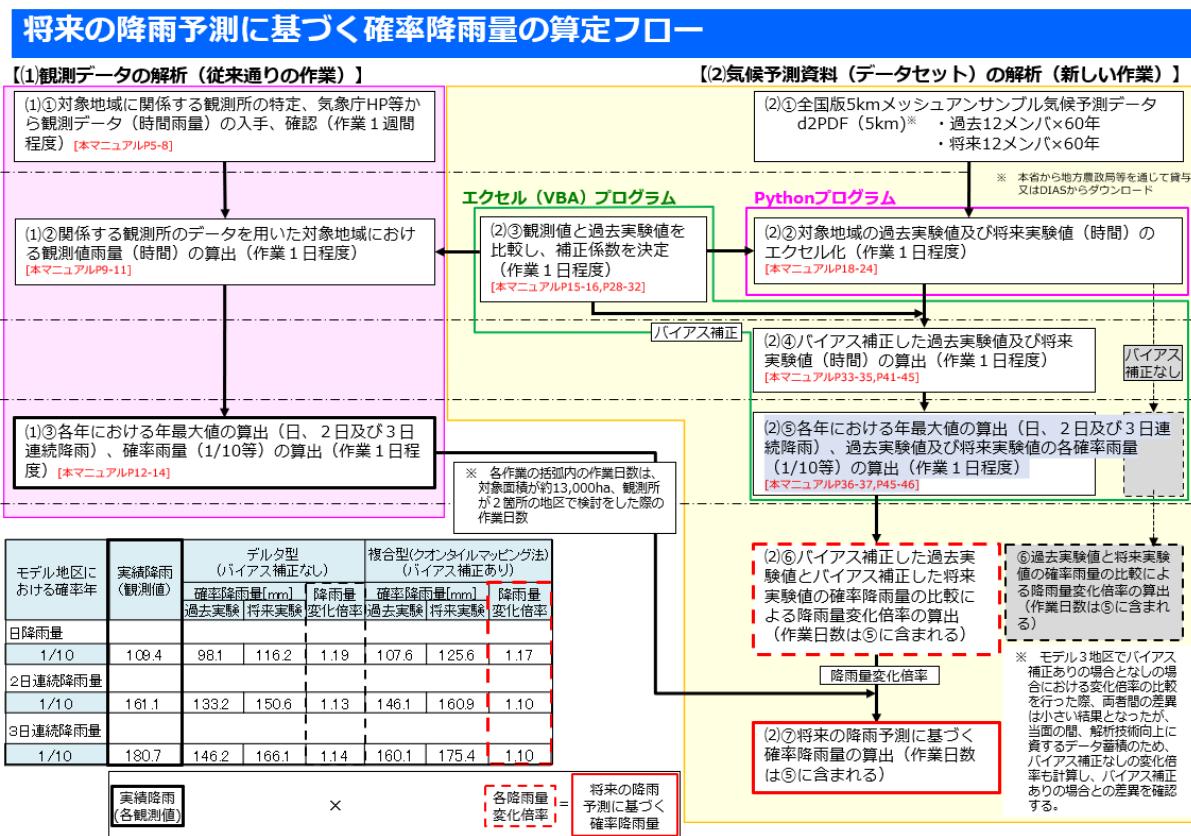
(参考)

附属資料 1 実験データの入手	P47
附属資料 2 Python プログラムコード	P51

第1章 作業フロー

将来の降雨予測に基づく確率降雨量の算定は、気象庁の観測降雨量データ、気候予測資料（データセット）のd2PDF（5 km）の過去実験データ及び将来実験データの3つのデータを使用して行う。なお、d2PDF（5 km）は「全国5 km メッシュアンサンブル気候予測データ」におけるRCP8.5シナリオに基づく2 °C上昇時点（2040年頃）の予測結果であり、正式名称は「d4PDF（2 °C上昇実験、5 km）」である。

その3つのデータを利用した確率降雨量の算定フローを下記に示す。



第2章 作業環境の設定（Python）

本作業で実行する Python プログラムコードは、Linux 上で動作するため、Linux 環境が整っていない場合は、以下第1節の環境構築が必要である。

第1節 Windows における環境構築例（Linux 環境未構築の場合）

WSL（Windows Subsystem for Linux）をインストールすることにより Windows に Linux 環境（Linux 系の OS である Ubuntu）を構築することが可能である（前提条件として Windows のバージョンは、Windows 10 の 2004 以上であること）。

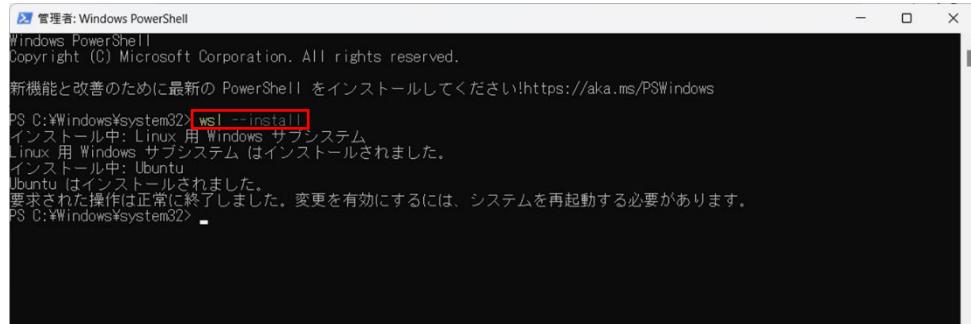
また、WSL のインストールにおいて、同時に Python もインストールすることが可能であり一度に作業環境が構築出来るため、下記のインストール方法を推奨する。

（インストールの詳細 <https://learn.microsoft.com/ja-jp/windows/wsl/install>）

1 WSL のインストール

Windows PowerShell 上で、「wsl --install」を入力し、WSL をインストールする。

（Windows PowerShell 画面）

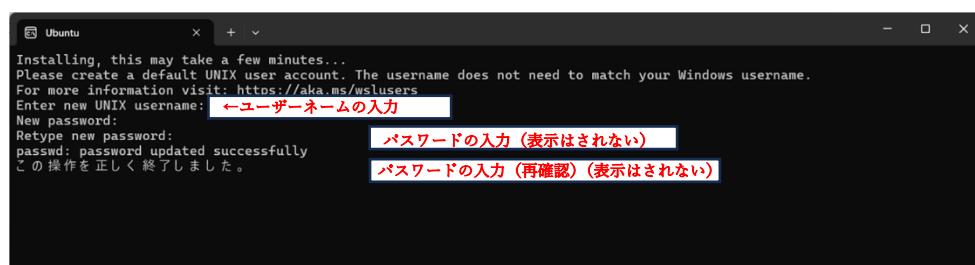


```
PS C:\Windows\system32> wsl --install
インストール中: Linux 用 Windows サブシステム
Linux 用 Windows サブシステム はインストールされました。
インストール中: Ubuntu
Ubuntu はインストールされました。
要求された操作は正常に終了しました。変更を有効にするには、システムを再起動する必要があります。
PS C:\Windows\system32>
```

2 ユーザー情報の登録

Microsoft Store から Linux で検索して Ubuntu アプリをクリックし、設定画面で、ユーザーネームとパスワードを設定しユーザー情報を登録する。

（Ubuntu 設定画面）



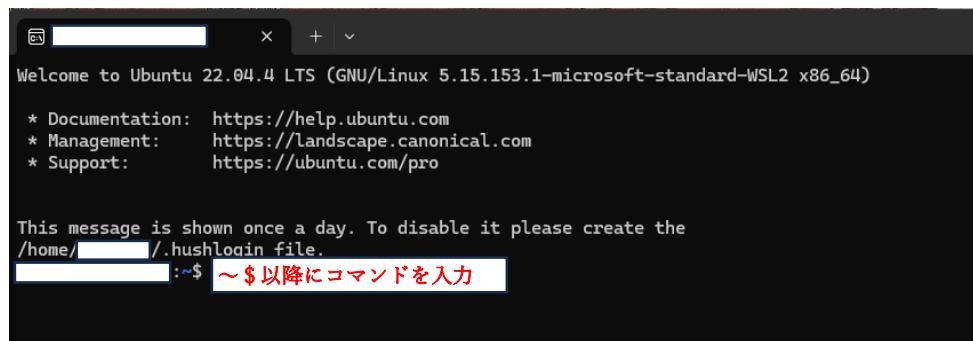
第2節 Python のライブラリのインストール (Linux 環境構築後)

1 PIP (ライブラリをインストールするための仕組み) のインストール

Python のプログラムコードを実行する際に必要なライブラリ (Python 上で簡単に数値計算等の機能を利用出来るようにするためのプログラムコードの集合体) をインストールするため、PIP をインストールする。

Ubuntu のターミナル画面に下記のコマンドを入力し、PIP のインストールを行う。

(Ubuntu のターミナル画面の例)



```
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.153.1-microsoft-standard-WSL2 x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/pro

This message is shown once a day. To disable it please create the
/home/____/.hushlogin file.
[Ubuntu:~]$ ~$以降にコマンドを入力
```

(コマンドの入力：PIP のインストール状況の確認)

```
$ pip3 -V
```



(未インストールの場合の表示例)

```
$ pip3 -V
pip3: command not found
```



(インストール済みの場合の表示例)

```
$ pip3 -V
pip 22.0.2 from /usr/lib/python3/dist-packages/pip (python 3.10)
```

(コマンドの入力：apt のアップデート)

```
$ sudo apt update
```



(コマンドの入力：apt のアップグレード)

```
$ sudo apt upgrade
```



(コマンドの入力：PIP のインストール)

```
$ sudo apt install python3-pip
```

※PIP のインストール状況は、再度「\$ pip3 -V」を入力し確認する。

2 ライブラリのインストール

必要なライブラリ (numpy, Pillow, netCDF4, pyshp) をインストールする。

(コマンドの入力：ライブラリのインストール)

```
$ pip3 install numpy Pillow netCDF4 pyshp
```

※ライブラリのインストール結果は、下記のコマンドを入力し確認する。

(コマンドの入力：ライブラリのインストール結果の確認)

```
$ pip list
```

(Ubuntu のターミナル画面の表示例)

Package	Version
blinker	1.4
certifi	2024.6.2
cftime	1.6.4
command-not-found	0.3
cryptography	3.4.8
dbus-python	1.2.18
distro	1.7.0
distro-info	1.1+ubuntu0.2
httpplib2	0.20.2
importlib-metadata	4.6.4
jeepney	0.7.1
keyring	23.5.0
launchpadlib	1.10.16
lazr.restfulclient	0.14.4
lazr.uri	1.0.6
more-itertools	8.10.0
netCDF4	1.7.1
netifaces	0.11.0
numpy	2.0.0
oauthlib	3.2.0
pillow	10.3.0
pip	22.0.2
PyGObject	3.42.1
PyJWT	2.3.0
pyparsing	2.4.7
pyshp	2.3.1
python-apt	2.4.0+ubuntu3
PyYAML	5.4.1
SecretStorage	3.3.1
setuptools	59.6.0
six	1.16.0
systemd-python	234
ubuntu-pro-client	8001
ufw	0.36.1
unattended-upgrades	0.1
wadllib	1.3.6
wheel	0.37.1
zipp	1.0.0

第3章 観測値の処理

第1節 データの入手

気象庁 HP ([気象庁|過去の気象データ・ダウンロード \(jma.go.jp\)](http://www.jma.go.jp)) 等から観測値（時間降雨量）に関するデータをダウンロードする。

また、ダウンロードする観測値（将来降雨量を算出する際に用いる観測値）は、実験データが9月1日～翌年8月31日を1年間として計算されていることから、9月1日～翌年8月31年を1年間として扱う。

■ダウンロードするデータ

1951年9月1日～2011年8月31日（過去実験値の計算期間）の内、2011年8月31日から遡った30年～50年程度のデータ

(例: 気象庁 HP からのダウンロード)

(地点の選択)



(項目の選択) データの種類: 時別値 項目: 降水量 (前1時間)

国土交通省 気象庁 Japan Meteorological Agency

ホーム 防災情報 各種データ・資料 地域の情報 知識・解説 各種申請・ご案内

過去の気象データ・ダウンロード

データ検索 関連ページ 過去の気象データ 過去の地域平均気象データ 取り組んでみませんか？ 気候リスク管理

ホーム > 各種データ・資料 > 過去の気象データ・ダウンロード

【お知らせ】時間帯によっては繋がらないことがありますやデータ取得に時間がかかることがあります多くなっています。繋がらない場合は時間をおいて再度お試しください。(2023.2.8~)

検索条件

地点を選ぶ 項目を選ぶ 期間を選ぶ 表示オプションを選ぶ

データの種類 時別値 最初に選択してください

過去の平均値との比較オプション

選択済みのデータ量 0% 100%(上限)

画面に表示 CSVファイルをダウンロード

選択地点・項目をクリア

選択された地点 鶴岡市 新潟 削除

選択された項目 降水量(前1時間) 削除

選択された期間(日本標準時) 1981年9月1日から

項目

降水量(前1時間) 選択

(期間の選択) 1回毎のダウンロード制限があるため、1地点1年毎にダウンロード

国土交通省 気象庁 Japan Meteorological Agency

ホーム 防災情報 各種データ・資料 地域の情報 知識・解説 各種申請・ご案内

過去の気象データ・ダウンロード

データ検索 関連ページ 過去の気象データ 過去の地域平均気象データ 取り組んでみませんか？ 気候リスク管理

ホーム > 各種データ・資料 > 過去の気象データ・ダウンロード

【お知らせ】時間帯によっては繋がらないことがありますやデータ取得に時間がかかることがあります多くなっています。繋がらない場合は時間をおいて再度お試しください。(2023.2.8~)

検索条件

地点を選ぶ 項目を選ぶ 期間を選ぶ 表示オプションを選ぶ

期間 連続した期間で表示する

選択済みのデータ量 0% 100%(上限)

画面に表示 CSVファイルをダウンロード

選択地点・項目をクリア

選択された地点 鶴岡市 新潟 削除

選択された項目 降水量(前1時間) 削除

選択された期間(日本標準時) 1981年9月1日から

期間

連続した期間で表示する

最近1年 [選択] 最近1か月

1981 年 9 月 1 日から 1982 年 8 月 31 日までの時別値を表示

特定の期間を複数年分、表示する

9 月 1 日から 8 月 31 日の時別値を表示
1981 年から 1982 年まで表示

特定の時間帯のデータのみ表示する

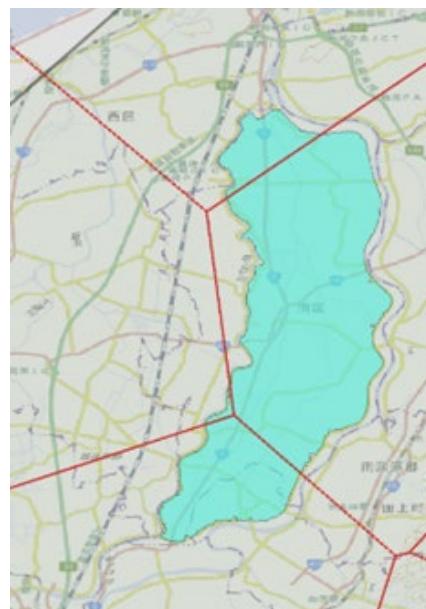
1 時間から 24 時間のデータを表示
(選択しない場合は24時間すべてのデータを表示します)

(例：気象庁 HP からダウンロードした 1 観測所の降雨量データ)

A	B	C	D	E	F	G	H
1	ダウンロードした時刻：2024/09/17 09:11:27						
2							
3	新潟	新潟	新潟	新潟			
4	年月日時	降水量(mm)	降水量(mm)	降水量(mm)	降水量(mm)		
5		現象なし情報	品質情報		均質番号		
6	1981/9/1 1:00	0	1	8	1		
7	1981/9/1 2:00	0	1	8	1		
8	1981/9/1 3:00	0	1	8	1		
9	1981/9/1 4:00	2.5	0	8	1		
10	1981/9/1 5:00	0.5	0	8	1		
11	1981/9/1 6:00	2	0	8	1		
12	1981/9/1 7:00	0.5	0	8	1		
13	1981/9/1 8:00	0.5	0	8	1		
14	1981/9/1 9:00	0.5	0	8	1		
15	1981/9/1 10:00	0	0	8	1		
16	1981/9/1 11:00	0	1	8	1		
17	1981/9/1 12:00	0	1	8	1		
18	1981/9/1 13:00	0	1	8	1		
19	1981/9/1 14:00	0	1	8	1		
20	1981/9/1 15:00	0	1	8	1		
21	1981/9/1 16:00	0	1	8	1		
22	1981/9/1 17:00	0	1	8	1		
23	1981/9/1 18:00	0	1	8	1		

また、各対象地区に必要な観測地点、面積割合の特定は、地図上からティーセン法等を用いて行う。

(例：GIS 上におけるティーセン法の使用)



複数の観測所を使用する場合は、全ての観測所の降雨量を1年毎にエクセルシートに整理する。

(例：1年毎の観測値の整理)

ダウンロードした観測所Bのダウンロードデータ

A B C		
1	ダウソードした時刻：2024/09/17 09:11:27	
2	観測所A 観測所B	
3	新潟	新津
4	年月日時	降水量(mm)
6	1981/9/1 1:00	0
7	1981/9/1 2:00	0
8	1981/9/1 3:00	0
9	1981/9/1 4:00	2.5
10	1981/9/1 5:00	0.5
11	1981/9/1 6:00	2
12	1981/9/1 7:00	0.5
13	1981/9/1 8:00	0.5
14	1981/9/1 9:00	0.5
15	1981/9/1 10:00	0
16	1981/9/1 11:00	0
17	1981/9/1 12:00	0
18	1981/9/1 13:00	0
19	1981/9/1 14:00	0
20	1981/9/1 15:00	0
21	1981/9/1 16:00	0
22	1981/9/1 17:00	0
23	1981/9/1 18:00	0

A B C D E F G H				
1	ダウソードした時刻：2024/09/17 13:59:40			
2				
3	新津	新津	新津	
4	年月日時	降水量(mm)	降水量(mm)	品質情報
5				均質番号
6	1981/9/1 1:00	0	8	1
7	1981/9/1 2:00	0	8	1
8	1981/9/1 3:00	0	8	1
9	1981/9/1 4:00	2	8	1
10	1981/9/1 5:00	2	8	1
11	1981/9/1 6:00	0	8	1
12	1981/9/1 7:00	0	8	1
13	1981/9/1 8:00	0	8	1
14	1981/9/1 9:00	0	8	1
15	1981/9/1 10:00	1	8	1
16	1981/9/1 11:00	0	8	1
17	1981/9/1 12:00	0	8	1
18	1981/9/1 13:00	0	8	1
19	1981/9/1 14:00	0	8	1
20	1981/9/1 15:00	0	8	1
21	1981/9/1 16:00	0	8	1
22	1981/9/1 17:00	0	8	1
23	1981/9/1 18:00	0	8	1

全ての観測所の時間降雨量データを1年毎に
シート上に並べておく。

第2節 データの処理

1 確率降雨量の算出

(1) システムへのデータの追加

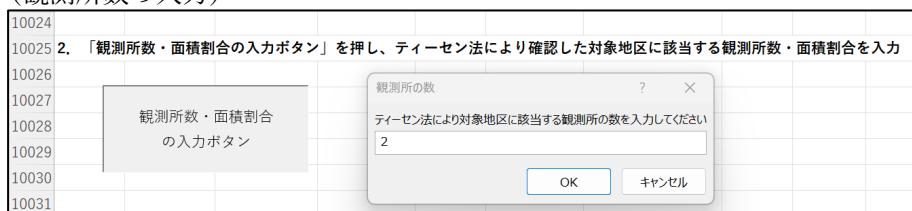
『(観測値) 年最大雨量算出システム』を開き、「解析スタート」シートの後ろに整理した観測値のシートを1年毎に追加(例: 1981年~2010年を選定した場合、「解析スタート」シートの後ろ(右側)に1981年~2010年のデータを追加)

(2) 対象流域の条件入力

「解析スタートシート」内の「観測所数・面積割合の入力ボタン」を押し、観測所数、面積割合等を入力（例は、対象地区に関する観測所が2箇所の場合）



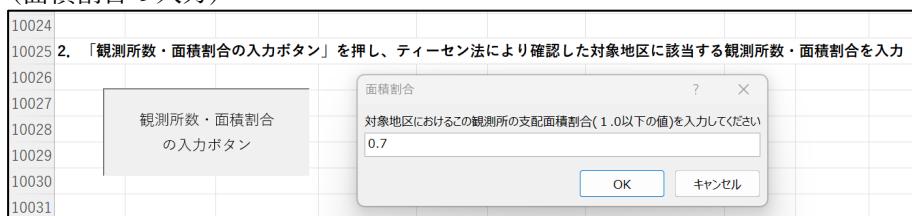
（観測所数の入力）



（観測所名の入力）



（面積割合の入力）





10024	
10025	2. 「観測所数・面積割合の入力ボタン」を押し、ティーセン法により確認した対象地区に該当する観測所数・面積割合を入力
10026	
10027	観測所数・面積割合 の入力ボタン
10028	観測所数 観測所名 割合 (合計1.00)
10029	2 新潟 0.7
10030	新津 0.3
10031	

(3) 時間毎の流域平均雨量の算出

「解析スタートシート」内の「流域平均雨量算出ボタン」を押し、各年の「降雨量シート」の R 列に各地区の流域平均雨量（時間）を算出

※ 使用する PC の環境によっては、処理に時間がかかる場合があるため、その場合は、一回に処理（追加）するシートを少なくするなどして分けて作業すること

(解析スタートシート)

10042	A B C D E F G H I J K L M N
10043	3. 「流域平均雨量算出ボタン」を押すと、各シートで時間雨量毎の流域平均雨量が算出
10044	※使用するPCの環境によっては、処理に時間がかかる場合があるため、その場合は、一回に処理（追加）するシートを少なくするなどして分けて計算すること
10045	
10046	流域平均雨量算出ボタン
10047	



(降雨量シート)

P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
時間雨量 (mm)													
6	1981/9/1 1:00	0											
7	1981/9/1 2:00	0											
8	1981/9/1 3:00	0											
9	1981/9/1 4:00	2.35											
10	1981/9/1 5:00	0.95											
11	1981/9/1 6:00	1.4											
12	1981/9/1 7:00	0.35											
13	1981/9/1 8:00	0.35											
14	1981/9/1 9:00	0.35											
15	1981/9/1 10:00	0.3											
16	1981/9/1 11:00	0											
17	1981/9/1 12:00	0											
18	1981/9/1 13:00	0											
19	1981/9/1 14:00	0											
20	1981/9/1 15:00	0											
21	1981/9/1 16:00	0											
22	1981/9/1 17:00	0											
23	1981/9/1 18:00	0											
24	1981/9/1 19:00	0											
25	1981/9/1 20:00	0											
26	1981/9/1 21:00	0											
27	1981/9/1 22:00	0											
28	1981/9/1 23:00	0											
29	1981/9/2 0:00	0											
30	1981/9/2 1:00	0											
31	1981/9/2 2:00	0											
32	1981/9/2 3:00	0											

(4) 年最大雨量の算出

「解析スタートシート」内の「雨量累積ボタン」を押すと、各降雨量シートのU～W列に時間雨量の累積計算（日・2日・3日）・年最大雨量（日・2日・3日）が算出

※ 使用するPCの環境によっては、処理に時間がかかる場合があるため、その場合は、一回に処理（追加）するシートを少なくするなどして分けて作業すること

(解析スタートシート)

A	B	C	D	E	F	G	H	I	J	K	L	M	N
10048													
10049	4. 「雨量累積ボタン」を押すと、各シートで時間雨量の累積計算・年最大雨量（日・2日・3日）の抽出が開始												
10050	※使用するPCの環境によっては、処理に時間がかかる場合があるため、その場合は、一回に処理（追加）するシートを少なくするなどして分けて計算すること												
10051													
10052	雨量累積ボタン												
10053													
10054													



(降雨量シート)

P	Q	R	S	T	U	V	W	X	Y	Z
1										
2										
3					最大日雨量 (mm) 最大2日雨量 (mm) 最大3日雨量 (mm)					
4					59.6	89.7	112.3			
5		時間雨量 (mm)			日雨量 (mm)	2日雨量 (mm)	3日雨量 (mm)			
6	1981/9/1 1:00	0	1981/9/1	6.05						
7	1981/9/1 2:00	0	1981/9/2	0	6.05					
8	1981/9/1 3:00	0	1981/9/3	0	0	6.05				
9	1981/9/1 4:00	2.35	1981/9/4	14.1	14.1	14.1				
10	1981/9/1 5:00	0.95	1981/9/5	0	14.1	14.1	14.1			
11	1981/9/1 6:00	1.4	1981/9/6	0	0	14.1				
12	1981/9/1 7:00	0.35	1981/9/7	0	0	0	0			
13	1981/9/1 8:00	0.35	1981/9/8	0	0	0	0			
14	1981/9/1 9:00	0.35	1981/9/9	8	8	8	8			
15	1981/9/1 10:00	0.3	1981/9/10	0.3	8.3	8.3				
16	1981/9/1 11:00	0	1981/9/11	0	0.3	8.3				
17	1981/9/1 12:00	0	1981/9/12	1.35	1.35	1.65				
18	1981/9/1 13:00	0	1981/9/13	16.1	17.45	17.45				
19	1981/9/1 14:00	0	1981/9/14	6.9	23	24.35				
20	1981/9/1 15:00	0	1981/9/15	0	6.9	23				
21	1981/9/1 16:00	0	1981/9/16	0	0	6.9				
22	1981/9/1 17:00	0	1981/9/17	0	0	0				
23	1981/9/1 18:00	0	1981/9/18	0	0	0				
24	1981/9/1 19:00	0	1981/9/19	0	0	0				
25	1981/9/1 20:00	0	1981/9/20	0	0	0				
26	1981/9/1 21:00	0	1981/9/21	0	0	0				
27	1981/9/1 22:00	0	1981/9/22	0	0	0				
28	1981/9/1 23:00	0	1981/9/23	0	0	0				
29	1981/9/2 0:00	0	1981/9/24	0	0	0				

(5) 年最大雨量の整理

「解析スタートシート」内の「年最大雨量集計ボタン」を押し、下表に各年の年最大雨量（日・2日・3日）を抽出

- ※ ボタンを押した際に実行時エラー表示されるが、そのまま終了ボタンを押すこと
- ※ 複数回に分けて（4）までの作業を行った場合は、（4）までの作業を行った全ての対象期間のシートを「解析スタート」シートの後ろ（右側）に追加してから実行すること

（解析スタートシート）

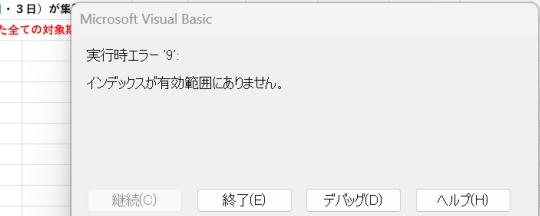
A	B	C	D	E	F	G	H	I	J	K	L	M	N
10055	5. 「年最大雨量集計ボタン」を押すと、下表に各年の年最大雨量（日・2日・3日）が集計												
10056	※（複数回に分けて4までの作業を行った場合は、）4までの作業を行った全ての対象期間のシートを「解析スタート」シートの後に追加してから実行すること												
10057	年最大雨量集計ボタン												
10058	最大日雨量 最大2日雨量 最大3日雨量												
10059													
10060	1												
10061	2												
10062	3												
10063	4												
10064	5												
10065	6												
10066	7												
10067	8												
10068	9												
10069	10												
10070	11												
10071	12												
10072	13												
10073	14												
10074	15												
10075	16												
10076	17												
10077	18												
10078	19												
10079	20												
10080	21												
10081	22												
< >	解析スタート	1981	1982	1983	1984	1985	1986	1987	1988	…	+	:	



（解析スタートシート）

A	B	C	D	E	F	G	H	I	J	K	L	M	N
10054													
10055	5. 「年最大雨量集計ボタン」を押すと、下表に各年の年最大雨量（日・2日・3日）が集計												
10056	※（複数回に分けて4までの作業を行った場合は、）4までの作業を行った全ての対象期間のシートを「解析スタート」シートの後に追加してから実行すること												
10057	年最大雨量集計ボタン												
10058	最大日雨量 最大2日雨量 最大3日雨量												
10059													
10060	1	59.6	89.7	112.3									
10061	2												
10062	3												
10063	4												
10064	5												
10065	6												
10066	7												
10067	8												
10068	9												
10069	10												
10070	11												
10071	12												
10072	13												
10073	14												
10074	15												
10075	16												
10076	17												
10077	18												
10078	19												
10079	20												
10080	21												
< >	解析スタート	1981	1982	1983	1984	1985	1986	1987	1988	…	+	:	

実行時エラー表示されるが、そのまま終了ボタンを押すこと



(6) 確率降雨量の算出

(5) で集計した値を用いて、日～3日連続雨量における1/10等の確率降雨量を算出

※ 確率降雨量の算出方法は、土地改良事業計画設計基準 計画「排水」技術書「6. 実績降雨に基づく計画基準降雨」を参照

2 パーセンタイル値の算出

以降の（1）及び（2）の作業の結果は、第5章第2節において実験値のバイアス補正をする上で必要となる。

（1）時間1mm以上の降雨値の抽出

「解析スタートシート」内の「1mm以上検出ボタン」を押し、各年「降雨量シート」のS列に1mm以上の値を抽出

（解析スタートシート）

	O	P	Q	R	S	T	U	V	W	X	Y	Z
1												
10011												
10012												
10013												
10014												
10015												

→ パーセンタイル値の算出作業

4. 「1mm以上検出ボタン」を押すと、1mm以上の値が各年シートのC列に検出（バイアス補正是1mm以上の時間雨量で行うため）

1mm以上検出



（降雨量シート）

	P	Q	R	S	T	U	V	W	X	Y	Z
1											
2											
3											
4											
5				時間雨量（mm）1mm以上の数値		日雨量（mm）	2日雨量（mm）	3日雨量（mm）			
6	1981/9/1 1:00	0		1981/9/1	6.05						
7	1981/9/1 2:00	0		1981/9/2	0	6.05					
8	1981/9/1 3:00	0		1981/9/3	0	0	6.05				
9	1981/9/1 4:00	2.35		1981/9/4	14.1	14.1	14.1				
10	1981/9/1 5:00	0.95		1981/9/5	0	14.1	14.1				
11	1981/9/1 6:00	1.4		1981/9/6	0	0	14.1				
12	1981/9/1 7:00	0.35		1981/9/7	0	0	0				
13	1981/9/1 8:00	0.35		1981/9/8	0	0	0				
14	1981/9/1 9:00	0.35		1981/9/9	8	8	8				
15	1981/9/1 10:00	0.3		1981/9/10	0.3	8.3	8.3				
16	1981/9/1 11:00	0		1981/9/11	0	0.3	8.3				
17	1981/9/1 12:00	0		1981/9/12	1.35	1.35	1.65				
18	1981/9/1 13:00	0		1981/9/13	16.1	17.45	17.45				
19	1981/9/1 14:00	0		1981/9/14	6.9	23	24.35				
20	1981/9/1 15:00	0		1981/9/15	0	6.9	23				
21	1981/9/1 16:00	0		1981/9/16	0	0	6.9				
22	1981/9/1 17:00	0		1981/9/17	0	0	0				
23	1981/9/1 18:00	0		1981/9/18	0	0	0				
24	1981/9/1 19:00	0		1981/9/19	0	0	0				
25	1981/9/1 20:00	0		1981/9/20	0	0	0				
26	1981/9/1 21:00	0		1981/9/21	0	0	0				
27	1981/9/1 22:00	0		1981/9/22	0	0	0				
28	1981/9/1 23:00	0		1981/9/23	0	0	0				
29	1981/9/2 0:00	0		1981/9/24	0	0	0				

< > 解析スタート 1981 1982 1983 1984 1985 1986 1987 1988 ... +

(2) パーセンタイル値の算出

「PERCENTILE.INC」関数を使用し、1mm以上の観測値における1%毎のパーセンタイル値を算出 ※ 必ず、全ての期間のシートを追加した上で作業すること

「解析スタートシート」内の R 列 10029 行目から 10128 行目の数式を全てのシートが計算対象範囲に含まれるように変更すること（例：計算するシートの 1 番目が「1981」で最後が「2010」であれば、R 列 10029 行目のセルの関数を「=PERCENTILE.INC(1981:2010!\$S\$6:\$S\$9000,P10029)」と修正し、数式を R 列 10128 行目までドラッグすること）

(解析スタートシート)

数式を変更すること

=PERCENTILE.INC(最初のシート名:最後のシート名!\$S\$6:\$S\$9000,P10029)

第4章 実験データの処理

気候変動を踏まえた排水計画を策定するにあたり、気候予測資料（データセット）は d2PDF (5 km) を使用する。d2PDF (5 km) データは、文部科学省の委託事業として開発が進められてきたデータ統合・解析システム（以降、DIAS）（<https://diasjp.net/>）上で「全国 5km メッシュアンサンブル気候予測データ」という名称で公開されているが、DIAS 上では、netCDF 形式のファイルで公開されており、事前に Python 等を使用したプログラミングにより CSV 形式のファイルに変換する必要があるため（降雨量数値の見える化）、その方法について説明する。

第1節 実験データの入手

気候予測資料（データセット）の d2PDF (5 km) データは、農林水産省農村振興局整備部設計課から地方農政局等を通じて貸与する。なお、貸与データは、本マニュアルにより土地改良事業計画（排水）における将来の降雨予測に基づく確率降雨量算定にのみ使用するものとし、配布先は各貸与元において管理するものとする。

また、DIAS より d2PDF (5 km) データを入手する場合は、巻末に付記している「附属資料 1 実験データの入手」を参照する。

第2節 対象地域の過去実験値及び将来実験値のcsv化

第1節で入手した「rain.nc」及び「rain.nc_pdef.ctl」ファイルを用いて実験データをcsv化する必要があるため、本項では、Pythonを用いたnetCDFファイルのcsv化の例を示す。

1 Python実行プログラム及びシェイプファイルの準備

巻末に付記している「附属資料2 Pythonプログラムコード」は、複数のディレクトリにあるrain.ncファイルから降雨量データを抽出し、対象地域（シェイプファイルで指定）に対応するデータをcsv形式で出力するもの。なお、csv形式で出力される対象地域の降雨量は、d2PDF（5km）のメッシュ（以降、グリッド）に含まれるシェイプファイルの面積に応じて加重平均されたものである。

「附属資料2 Pythonプログラムコード」をコマンドにより実行するため、事前に拡張子pyファイル（ファイル名：netcdf2csv_arealmean_cnst.py、別途配布）をPC上に保存しておく。d2PDF（5km）の緯度経度情報を得るため、DIASより「cnst.nc」をダウンロードしてPC上に保存しておく。また、GIS（地理情報システム）ソフトにより対象地域のシェイプファイルを作成し、PC上に保存しておく。

2 csv化コマンドの実行

コマンドを実行すると出力ディレクトリの直下に入力ディレクトリと同名のサブディレクトリが作成され、csvファイルが格納される。

（1）コマンドの記載方法

「\$」→「netcdf2csv_arealmean_cnst.py」→「-o 出力ディレクトリパス」→「-s シェイプファイルの保存ディレクトリパス」→「--shift-hours -48」→「--min-coverage 0」→「--export-cells-shp 出力シェイプファイル名」→「入力ディレクトリパス」の順番で記載する。

（ターミナル画面：コマンド入力例）

```
$ python3 netcdf2csv_arealmean_cnst.py -o ./csv -s 排水ブロック N_EPSG4326.shp --shift-hours -48 --min-coverage 0 --export-cells-shp ./csv/HFB_2K_CC_m101/selected_cells.shp ../HFB_2K_CC_m101/
```

上記コマンド入力例では、相対パスを例示しているが、絶対パスを使用できる。

（パス設定を行う際の参考）

現在の作業ディレクトリは下記のコマンドで確認できる。

(ターミナル画面: コマンド入力例)

```
$ pwd
```

(2) コマンドの説明

-o 「csv ファイルの出力パス」

- ・・・ output、出力ディレクトリパスを設定する。d2PDF (5 km) から抽出した降雨が
出力される。

-s 「対象地域のシェイプファイル」

・・・ shapefile、抽出対象の地点を指定するシェイプファイルを設定する。

--shift-hours -48

・ 時刻データを時間単位で補正 (-48 を指定) する。

--min-coverage 0

・採用する最小の被覆率を指定(0を指定)する(被覆率についてはP21に記載)。

--export-cells-shp 「出力シェイプファイル名」

- 採用グリッドの出力シェイプファイル名（拡張子は shp、gpkg、geojson から選択可能）を設定する。

「入力ファイルのディレクトリパス」

・・・ 入力ディレクトリパス(複数指定可)を設定する。

対象地域のシェイプファイル

```
$ python3 netcdf2csv_arealmean_cnst.py -o ./csv -s 排水プロック N_EPSG4326.shp --shift-hours -48 --
```

csv ファイルの出力パス

```
min-coverage 0 --export-cells-shp ./csv/HFB_2K_CC_m101/selected_cells.shp ./HFB_2K_CC_m101/
```

出力シェイプファイル名

入力ファイルの ディレクトリパス

(参考)

```
usage: netcdf2csv_arealmean_cnst.py [-h] -o OUTPUT -s SHAPEFILE --shift-hours --min-coverage  
--export-cells-shp inputs [inputs ...] ② ③ ⑤ ⑥  
⑦ ①
```

positional arguments:

① inputs
 入力ディレクトリパス(複数指定可)
 入力ディレクトリ下に存在するすべての rain.nc ファイルを探索し、入力ファイルとする
 出力サブディレクトリは入力ディレクトリと同じ名前で作成される
 ex) ./HFB_2K_*/

optional arguments:

② -h, --help
 show this help message and exit

② -o OUTPUT, --output OUTPUT
 出力ディレクトリパス
 ファイルの出力先ディレクトリ
 出力ディレクトリの直下に入力ディレクトリと同じ名前で作成される
 ex) -o ./csv/

③ -s SHAPEFILE, --shapefile SHAPEFILE
 対象地点 shapefile
 抽出対象の地点を指定する shapefile
 CRS は EPSG:4326(緯度経度)であること
 ex) -s ./排水ブロック N_EPSG4326.shp

④ -p POINT, --point POINT
 対象地点の緯度経度
 抽出対象地点の緯度経度
 緯度,経度 の順にカンマ区切りで指定する
 (カンマの前後にスペースを入れないこと)
 (-p と -s の両方を指定した場合、こちらは無視される)
 ex) -p 36.0329092313968,140.09731122949984

⑤ --shift-hours SHIFT_HOURS
 読み込んだ日時を指定時間だけシフトする
 ex) -48

⑥ --min-coverage MIN_COVERAGE
 採用する最小被覆率 (default=1e-12)
 0 で『少しでも重なれば』採用する

⑦ --export-cells-shp EXPORT_CELLS_SHP
 採用セルの範囲ポリゴンを出力(shp/gpkg/geojson)

④今回の作業では必要ないが、③シェイプファイルの情報の代わりに、緯度経度を指定すれば、1地点の降雨量データも確認することが可能

(3) 出力される csv ファイル

1) 対象グリッドの全雨量データ : rain_allgrids_[yyyy].csv

d2PDF (5 km) のうち、対象地域が該当するグリッドの全雨量データが年毎に出力される。対象地域のデータが抽出されているか、確認するための参考データとして使用する。

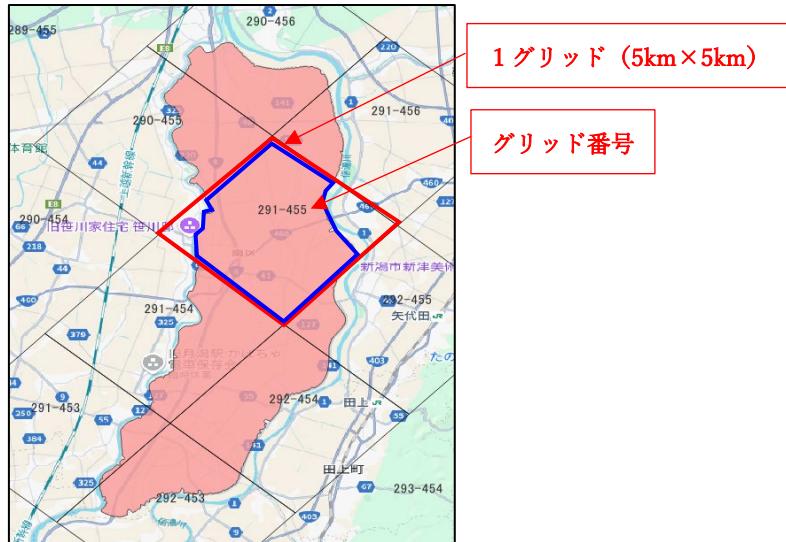
2) 地域平均雨量 : rain_arealmean_[yyyy].csv

地域平均雨量は、d2PDF (5 km) のグリッドに含まれるシェイプファイルの面積に応じて加重平均*されたものが年毎に出力される。本マニュアル第5章、第6章では、このデータを用いる。

※ 面積加重平均雨量 = 総和 (各グリッドの雨量 × 被覆率) / 総和 (被覆率)

被覆率 = グリッドに占めるポリゴンの面積 (青枠部分) / 1 グリッドの面積 (25km²)
(赤枠部分)

(例 : GIS 上における d2PDF (5 km) グリッドとの関係)



3) 対象グリッドのシェイプファイル出力 : selected_cells.shp

対象グリッドのシェイプファイルが出力される。このシェイプファイルを GIS 上にインポートすると、グリッド番号を確認できる。

4) 対象グリッドの面積率 : target_coverage.csv

対象グリッド毎に被覆率が出力される。面積加重平均雨量を確認するための参考データとして使用する。

(出力イメージ: 過去実験値 1981年~1983年を出力した場合)

- 1)  rain_allgrids_1981.csv
- 2)  rain_arealmean_1981.csv
- 3)  selected_cells.cpg
 selected_cells.dbf
 selected_cells.prj
 selected_cells.shp
 selected_cells.shx
- 4)  target_coverage.csv

1) 対象グリッドの全雨量データ

グリッド番号（地図上の数字）と整合しているか確認

各グリッドの時間雨量

2) 地域平均雨量

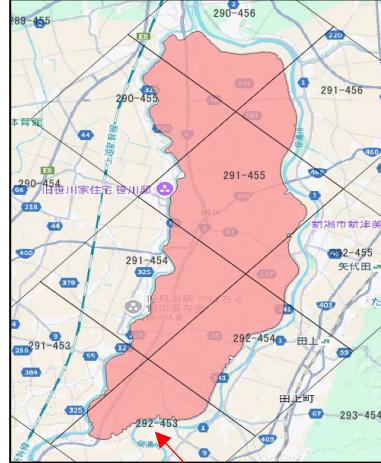
地域平均雨量

	A	B
1	date-hour	rainfall
2	1981/9/1 0:00	1.509467
3	1981/9/1 1:00	3.003704
4	1981/9/1 2:00	8.874269
5	1981/9/1 3:00	0.652825
6	1981/9/1 4:00	0.101944
7	1981/9/1 5:00	0.024269
8	1981/9/1 6:00	0.398258
9	1981/9/1 7:00	0.298949
10	1981/9/1 8:00	0.104757
11	1981/9/1 9:00	0.008587
12	1981/9/1 10:00	0.063337
13	1981/9/1 11:00	0.00052
14	1981/9/1 12:00	0.001284
15	1981/9/1 13:00	0

3) 対象グリッドのシェイプファイル出力

4) 対象グリッドの面積率

グリッドの被覆率



4) 対象グリッドの面積率

グリッドの被覆率

	A	B	C
1	i	j	fraction_area
2	291	453	0.062625
3	292	453	0.294764
4	291	454	0.391916
5	292	454	0.439995
6	290	455	0.279855
7	291	455	0.807092
8	292	455	0.095453
9	290	456	0.297298
10	291	456	0.14549

グリッド番号

→ グリッド番号

(4) 資料の整理

フォルダに格納された csv ファイルを整理する。

フォルダ名、ファイル名は以下のように整理しておくと後の作業がしやすくなる。



3 抽出データの確認方法

csv 化され、抽出されたデータが対象地域のデータとなっているか確認する方法は、以下のとおり。

(1) GIS（地理情報システム）ソフトによる確認

2 の (3) の 3) で出力されたシェイプファイルを GIS 上にインポートすると、d2PDF (5 km) の採用グリッドとグリッド番号が表示できる。

GIS 上に表示されたグリッド番号と対象地域のシェイプファイルの位置関係から、抽出したグリッドが間違いないかを確認する。

(2) 検算による確認

対象グリッドの全雨量データ (rain_allgrids_[yyyy].csv) 及び対象グリッドの面積率 (target_coverage.csv) を用いて地域平均雨量を検算し、地域平均雨量 (rain_arealmean_[yyyy].csv) との整合を確認する。

第5章 過去実験データの処理

第1節 確率降雨量の算出

1 データの追加

『(過去) 年最大雨量算出システム』を開き、「解析スタート」シートの後ろに d2PDF の降雨量シートを追加（例：1981年～2010年の期間のデータを選定した場合、「解析スタート」シートの後ろ（右側）に 12 メンバ×30 年（1981年～2010年）のデータを追加）

	A	B	C	D	E	F	G
1	date-hour	mean					
2	1981/9/1 0:00	0					
3	1981/9/1 1:00	0					
4	1981/9/1 2:00	0					
5	1981/9/1 3:00	0					
6	1981/9/1 4:00	0					
7	1981/9/1 5:00	0					•
8	1981/9/1 6:00	0					
9	1981/9/1 7:00	0					
10	1981/9/1 8:00	0					
11	1981/9/1 9:00	0					
12	1981/9/1 10:00	0					
13	1981/9/1 11:00	0					
14	1981/9/1 12:00	0					
15	1981/9/1 13:00	0					
16	1981/9/1 14:00	0					

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
(過去実験データ) 年最大雨量算出システム														
1. 整理した降雨量(過去実験)データ(エクセルシート)を「解析スタート」シートの後ろのシートに追加														
10011 2. 「雨量累積ボタン」を押すと、各シートで時間雨量の累積計算・年最大雨量(日・2日・3日)の抽出が開始														
10012 ※(複数回に分けて3までの作業を行った場合は、)4までの作業を行った全ての対象期間のシートを「解析スタート」シートの後ろに追加してから実行すること														
10029 雨量累積ボタン														
10030														
10031														
10032 3. 「年最大雨量集計ボタン」を押すと、下表に各年の年最大雨量(日・2日・3日)が集計														
10033 ※(複数回に分けて3までの作業を行った場合は、)4までの作業を行った全ての対象期間のシートを「解析スタート」シートの後ろに追加してから実行すること														
10034 年最大雨量集計ボタン														
10035 最大日雨量 最大2日雨量 最大3日雨量														
10036 1														
10037 2														
10038 3														
10039 A														
< > 解析スタート rain_1981 rain_1982 rain_1983 rain_1984 rain_1985 ... + : ◀														

※シート名は、どのメンバか分かるようにしたうえで追加する。

(例：HPB_m002→rain_1981(2)、HPB_m012→rain_1981(12)など)

2 年最大雨量の算出

「解析スタートシート」内の「雨量累積ボタン」を押すと、各降雨量シートで時間雨量の累積計算（日・2日・3日）・年最大雨量（日・2日・3日）が算出

※ 使用するPCの環境によっては、処理に時間がかかる場合があるため、その場合は、一回に処理（追加）するシートを少なくするなどして分けて作業すること

(解析スタートシート)

10025
10026 2. 「雨量累積ボタン」を押すと、各シートで時間雨量の累積計算・年最大雨量（日・2日・3日）の抽出が開始
10027 ※使用するPCの環境によっては、処理に時間がかかる場合があるため、その場合は、一回に処理（追加）するシートを少なくするなどして分けて作業すること
10028
10029  雨量累積ボタン
10030
10031



年最大雨量
(日～3日)

(降雨量シート)

A	B	C	D	E	F	G	H
1 date-hour	mean			最大日雨量 (mm)	最大2日雨量 (mm)	最大3日雨量 (mm)	
2 1981/9/1 0:00	0			53.93763856	56.98011744	60.25484833	
3 1981/9/1 1:00	0						
4 1981/9/1 2:00	0						
5 1981/9/1 3:00	0						
6 1981/9/1 4:00	0						
7 1981/9/1 5:00	0						
8 1981/9/1 6:00	0						
9 1981/9/1 7:00	0						
10 1981/9/1 8:00	0						
11 1981/9/1 9:00	0						
12 1981/9/1 10:00	0						
13 1981/9/1 11:00	0						
14 1981/9/1 12:00	0						
15 1981/9/1 13:00	0						
16 1981/9/1 14:00	0						
17 1981/9/1 15:00	0						
18 1981/9/1 16:00	0						
19 1981/9/1 17:00	0						
20 1981/9/1 18:00	0						

累積雨量
(日～3日)

3 年最大雨量の整理

「解析スタートシート」内の「年最大雨量集計ボタン」を押し、表に各年の年最大雨量

(日・2日・3日)を抽出

- ※ ボタンを押した際に実行時エラー表示されるが、そのまま終了ボタンを押すこと
- ※ 複数回に分けて2までの作業を行った場合は、2までの作業を行った全ての対象期間のシートを「解析スタート」シートの後ろに追加してから実行すること

(解析スタートシート)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
10031														
10032	3.	「年最大雨量集計ボタン」を押すと、下表に各年の年最大雨量（日・2日・3日）が集計												
10033	※（複数回に分けて2までの作業を行った場合は、）2までの作業を行った全ての対象期間のシートを「解析スタート」シートの後ろに追加してから実行すること													
10034	年最大雨量集計ボタン													
10035														



(解析スタートシート)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
10031														
10032	3.	「年最大雨量集計ボタン」を押すと、下表に各年の年最大雨量（日・2日・3日）が集計												
10033	※（複数回に分けて2までの作業を行った場合は、）2までの作業を行った全ての対象期間のシートを「解析スタート」シートの後ろに追加してから実行すること													
10034	年最大雨量集計ボタン													
10035														
10036		最大1日雨量	最大2日雨量	最大3日雨量										
10037	1	53.937639	56.98011744	60.2548483										
10038	2			0										
10039	3			0										
10040	4			0										
10041	5			0										
10042	6			0										
10043	7			0										
10044	8			0										
10045	9			0										
10046	10			0										
10047	11			0										
10048	12			0										
10049	13			0										
10050	14			0										
10051	15			0										
10052	16			0										
		解析スタート	rain_1981	rain_1982	rain_1983	rain_1984	rain_1985	rain_1986	...	+	:	◀	▶	⋮

4 確率降雨量の算出

3で集計した値を用いて、日～3日連続雨量における1/10等の確率降雨量を算出

- ※ 確率降雨量の算出方法は、土地改良事業計画設計基準 計画「排水」技術書「6. 実績降雨に基づく計画基準降雨」を参照

第2節 バイアス補正

過去実験値と観測値それぞれの時間降雨量のパーセンタイル値を比較し、1%毎に補正係数を求めて、過去実験値の時間降雨量に乗することにより、過去実験値のバイアス補正を行う。

1 時間 1mm 以上の降雨値の抽出

「解析スタートシート」内の「1mm 以上検出ボタン」を押し、各年降雨量シートの C 列に 1mm 以上の値を抽出

(解析スタートシート)

	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1													
10011													
10012													
10013													
10014													

バイアス補正作業

4. 「1mm以上検出ボタン」を押すと、1mm以上の値が各年シートのC列に検出（バイアス補正是1mm以上の時間雨量で行うため）

1mm以上検出



(降雨量シート)

	A	B	C	D	E	F	G	H
1	date-hour	mean	1mm以上の数値		最大日雨量 (mm)	最大 2 日雨量 (mm)	最大 3 日雨量 (mm)	
2	1981/9/1 0:00	0			53.93763856	56.98011744	60.25484833	
3	1981/9/1 1:00	0						
4	1981/9/1 2:00	0						
5	1981/9/1 3:00	0						
6	1981/9/1 4:00	0	1					
7	1981/9/1 5:00	0	m					
8	1981/9/1 6:00	0	m					
9	1981/9/1 7:00	0	以上					
10	1981/9/1 8:00	0	の					
11	1981/9/1 9:00	0	値					
12	1981/9/1 10:00	0	が					
13	1981/9/1 11:00	0	抽					
14	1981/9/1 12:00	0	出					
15	1981/9/1 13:00	0						
16	1981/9/1 14:00	0						
17	1981/9/1 15:00	0						
18	1981/9/1 16:00	0						
19	1981/9/1 17:00	0						
20	1981/9/1 18:00	0						

2 補正係数の算出

補正係数を求めるため、「PERCENTILE.INC」関数を使用し、1%毎にパーセンタイル値を算出 ※ 必ず、全ての期間のシートを追加した上で作業すること

(1) パーセンタイル値の算出

「解析スタートシート」内の S 列 10029 行目から 10128 行目の数式を全てのシートが対象範囲に含まれるように変更すること（例：計算するシートの 1 番目が rain_1981 で最後が rain_2010(12)であれば、S 列 10029 行目のセルの関数を「=PERCENTILE.INC(rain_1981:rain_2010(12)!\$C\$2:\$C\$9000,P10029)」と修正し、数式を S 列 10128 行目までドラッグすること）

(解析スタートシート)

	パーセンタイ	観測値	実験値（過去）	補正係数
10029	1	100	18.7	0
10030	0.99	99	11.5	0
10031	0.98	98	6.7	0
10032	0.97	97	5.6	0
10033	0.96	96	5.5	0
10034	0.95	95	5.3	0
10035	0.94	94	4.7	0
10036	0.93	93	4.6	0
10037	0.92	92	4.5	0
10038	0.91	91	4.2	0
10039	0.9	90	4.0	0
10040	0.89	89	4.0	0

数式を変更すること

S10029=PERCENTILE.INC(最初のシート名:最後のシート名!\$C\$2:\$C\$9000,P10029)

(2) 補正係数の算出

別途観測値エクセルで算出したパーセンタイル値を「解析スタートシート」内の R 列
10029 行目～10128 行目にコピー & ペースト

実験値（過去）と観測値を比較した結果が、補正係数として T 列に表示

(解析スタートシート)

バイアス補正作業

4. 「1mm以上検出ボタン」を押すと、1mm以上の値が各年シートのC列に検出（バイアス補正是1mm以上の時間雨量で行うため）
1mm以上検出

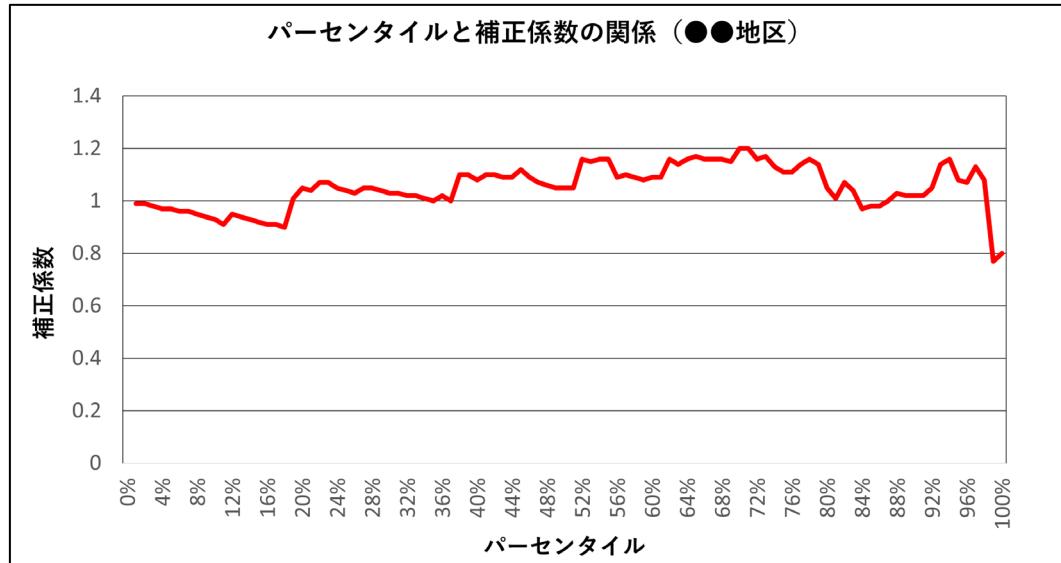
5. 補正係数を求めるため、①本シートS列「実験値（過去）」の数式を全てのシートが含まれるように変更すること
②観測値エクセルで算出したパーセンタイル値を本シートR列にコピー & ペースト

	パーセンタイル	観測値	実験値（過去）	補正係数
10029	1	100	14.9	18.74
10030	0.99	99	8.9005	11.56
10031	0.98	98	7.3	6.77
10032	0.97	97	6.3345	5.62
10033	0.96	96	5.95	5.54
10034	0.95	95	5.7275	5.31
10035	0.94	94	5.526	4.76
10036	0.93	93	5.247	4.62
10037	0.92	92	4.75	4.51
10038	0.91	91	4.3695	4.28
10039	0.9	90	4.11	4.01
10040	0.89	89	4	3.91
10041	0.88	88	3.926	3.82
10042	0.87	87	3.8	3.80
10043	0.86	86	3.7	3.76
10044	0.85	85	3.65	3.73

解析スタート rain_1981 rain_1982 rain_1983 rain_1984 rain_1985 ... +

(参考) 「解析スタートシート」内にパーセンタイルと補正係数の関係を示すグラフが作成される

(表示例)



(3) 数値の貼り付け

「解析スタートシート」内の R 列「観測値」10029 行目～10128 行目と S 列「実験値（過去）」10029 行目～10128 行目をコピーし、同じ場所に数値をペースト（数式をペーストしない）

(解析スタートシート)

関数が入力されている状態

バイアス補正作業

4. 「1mm以上検出ボタン」を押すと、1mm以上の値が各年シートのC列に検出（バイアス補正是1mm以上の時間雨量で行うため）

5. 補正係数を求めるため、①本シートS列「実験値（過去）」の数式を全てのシートが含まれるように変更すること
②観測値エクセルで算出したパーセンタイル値を本シートR列にコピー&ペースト

	パーセンタイル	観測値	実験値（過去）	補正係数	
10029	100	14.9	18.74	0.8	
10030	0.99	99	8.9005	11.56	0.77
10031	0.98	98	7.3	6.77	1.08
10032	0.97	97	6.3345	5.62	1.13
10033	0.96	96	5.95	5.54	
10034	0.95	95	5.7275	5.31	
10035	0.94	94	5.526	4.76	
10036	0.93	93	5.247	4.62	1.14
10037	0.92	92	4.75	4.51	1.05
10038	0.91	91	4.3695	4.28	1.02
10039	0.9	90	4.11	4.01	1.02
10040	0.89	89	4	2.01	1.02

コピー



(解析スタートシート)

値が入力されている状態

バイアス補正作業

4. 「1mm以上検出ボタン」を押すと、1mm以上の値が各年シートのC列に検出（バイアス補正是1mm以上の時間雨量で行うため）

5. 補正係数を求めるため、①本シートS列「実験値（過去）」の数式を全てのシートが含まれるように変更すること
②観測値エクセルで算出したパーセンタイル値を本シートR列にコピー&ペースト

	パーセンタイル	観測値	実験値（過去）	補正係数	
10029	100	14.9	18.74	0.8	
10030	0.99	99	8.9005	11.56	0.77
10031	0.98	98	7.3	6.77	
10032	0.97	97	6.3345	5.62	
10033	0.96	96	5.95	5.54	
10034	0.95	95	5.7275	5.31	1.08
10035	0.94	94	5.526	4.76	1.16
10036	0.93	93	5.247	4.62	1.14
10037	0.92	92	4.75	4.51	1.05
10038	0.91	91	4.3695	4.28	1.02
10039	0.9	90	4.11	4.01	1.02
10040	0.89	89	4	2.01	1.02

同じ場所に値のペースト

3 時間雨量の補正

「解析スタートシート」内の「バイアス補正ボタン」を押すと、各年降雨量シートのJ列にバイアス補正後の時間雨量が算出

- ※ 計算に時間がかかるため、1 メンバ毎に切り分けた計算（ボタンを押す）を推奨
 - ※ 100 パーセンタイルに相当する値のみ、エクセルブック内で検索を行い手動で補正係数をかけること

(解析スタートシート)

1	6. 「バイアス補正ボタン」を押すと、バイアス補正後の時間雨量が各年のシートで算出 ※計算に時間がかかるため、1メンバー毎に切り分けてボタンを押し計算をしてください ※100パーセンタイルに相当する値のみ、エクセルブック内で検索を行い手動で補正係数をかけてください
10011	
10012	
10013	
10014	
10025	
10026	



(降雨量シート)

A	B	C	D	E	F	G	H	I	J	K
1	date-hour	mean	1mm以上の数値	最大日雨量 (mm)	最大2日雨量 (mm)	最大3日雨量 (mm)	date-hour	バイアス補正後		
2	2031/9/1 0:00	0		90.04054067	93.69310133	101.3212879	2031/9/1 0:00	0		
3	2031/9/1 1:00	0					2031/9/1 1:00	0		
4	2031/9/1 2:00	0		日雨量 (mm)	2日雨量 (mm)	3日雨量 (mm)	2031/9/1 2:00	0		
5	2031/9/1 3:00	0	2031/9/1	0.706500556			2031/9/1 3:00	0		
6	2031/9/1 4:00	0	2031/9/2	0.123322222	0.829822778		2031/9/1 4:00	0		
7	2031/9/1 5:00	0	2031/9/3	0.567161556	0.690483778	1.396984333	2031/9/1 5:00	0		
8	2031/9/1 6:00	0.005284667	2031/9/4	0	0.567161556	0.690483778	2031/9/1 6:00	0.0052847		
9	2031/9/1 7:00	0.108332333	2031/9/5	0.063595222	0.063595222	0.630756778	2031/9/1 7:00	0.1083323		
10	2031/9/1 8:00	0.462775333	2031/9/6	2.190122222	2.253807444	2.253807444	2031/9/1 8:00	0.4627753		
11	2031/9/1 9:00	0.130108222	2031/9/7	0.539259444	2.729471667	2.793066889	2031/9/1 9:00	0.1301082		
12	2031/9/1 10:00	0	2031/9/8	2.368858556	2.908118	5.098330222	2031/9/1 10:00	0		
13	2031/9/1 11:00	0	2031/9/9	15.22911033	17.59796889	18.13722833	2031/9/1 11:00	0		
14	2031/9/1 12:00	0	2031/9/10	25.80585611	41.03496644	43.403825	2031/9/1 12:00	0		
15	2031/9/1 13:00	0	2031/9/11	60.28632144	86.09217756	101.3212879	2031/9/1 13:00	0		
16	2031/9/1 14:00	0	2031/9/12	0.012042444	60.29836389	86.10422	2031/9/1 14:00	0		
17	2031/9/1 15:00	0	2031/9/13	0	0.012042444	60.29836389	2031/9/1 15:00	0		
18	2031/9/1 16:00	0	2031/9/14	0	0	0.012042444	2031/9/1 16:00	0		
19	2031/9/1 17:00	0	2031/9/15	0	0	0	2031/9/1 17:00	0		
< >	解析スタート	rain_2031	rain_2032	rain_2033	rain_2034	rain_2035	...	+		

※ 100 パーセンタイルに相当する値の補正方法
(解析スタートシート)

バイアス補正作業

4. 「1mm以上検出ボタン」を押すと、1mm以上の値が各年シートのC列に検出（バイアス補正是1mm以上の時間雨量で行うため）
1mm以上検出

5. 補正係数を求めるため、①本シートS列「実験値（過去）」の数式を全てのシートが含まれるように変更すること
②観測値エクセルで算出したパーセンタイル値を本シートR列にコピー＆ペースト

※S列への数式入力例 :=PERCENTILE.INC(rain_1981:rain_2010(12)!\$C\$2:\$C\$9000,P10029)

	パーセンタイル	観測値	実験値（過去）	補正係数	
10029	1	100	14.9	18.74	0.8
10030	0.99	99	8.9005	11.56	0.77
10031	0.98	98	7.3	6.77	1.08

100 パーセンタイルに
相当する値をコピー

ホームタブのツールバー右側にある「検索と選択」より、検索をクリック
「検索と置換」の検索する文字列にコピーした値をペーストし、検索場所を「ブック」にして検索

検索(E)...

置換(R)...

検索(E)...

検索と置換

検索(D) 置換(P)

検索する文字列(N): 18.73729444444444

検索場所(H): ブック

検索方向(S): 行

検索対象(L): 数式

オプション(I) <<

すべて検索(I) 次を検索(E) 閉じる

(降雨量シート)

J688 : fx 18.7372944444444

	A	B	C	D	E	F	G	H	I	J	K
1	date-hour	mean	1mm以上の数値	最大日雨量最大2日雨量最大3日雨量 (mm)	date-hour					バイアス補正後	
684	1981/9/29 10:00	0.00017							1981/9/29 10:00	0.00017	
685	1981/9/29 11:00	2.239312	2.239312						1981/9/29 11:00	2.687174	
686	1981/9/29 12:00	3.121546	3.121546	バイアス補正前					1981/9/29 12:00	3.152762	
687	1981/9/29 13:00	3.081735	3.081735						1981/9/29 13:00	3.112552	
688	1981/9/29 14:00	18.73729	18.73729						1981/9/29 14:00	18.73729	バイアス補正後
689	1981	検索と置換									
690	1981										
691	1981	検索(D)	置換(E)								
692	1981	検索する文字列(N):	18.737294444444		書式セットなし	書式(M)...					
693	1981	検索場所(H):	ブック	<input type="checkbox"/>	大文字と小文字を区別する(C)						
694	1981	検索方向(S):	行	<input type="checkbox"/>	セル内容が完全に同一であるものを検索する(O)						
695	1981	検索対象(L):	数式	<input type="checkbox"/>	半角と全角を区別する(N)						
696	1981										
697	1981										
698	1981										
699	1981										
700	1981										
701	1981										
702	1981										
703	1981										
704	1981										
705	1981										
706	1981										

「バイアス補正後」欄の 100 パーセンタイルに相当する値のセルに、手動で計算式を入力して（「解析スタートシート」の補正係数を乗じて）補正する

J688 : fx =18.7372944444444*解析スタート!T10029

	A	B	C	D	E	F	G	H	I	J	K
1	date-hour	mean	1mm以上の数値	最大日雨量最大2日雨量最大3日雨量 (mm)	date-hour					バイアス補正後	
684	1981/9/29 10:00	0.00017							1981/9/29 10:00	0.00017	
685	1981/9/29 11:00	2.239312	2.239312						1981/9/29 11:00	2.687174	
686	1981/9/29 12:00	3.121546	3.121546						1981/9/29 12:00	3.152762	
687	1981/9/29 13:00	3.081735	3.081735						1981/9/29 13:00	3.112552	
688	1981/9/29 14:00	18.73729	18.73729						1981/9/29 14:00	14.98984	
689	1981/9/29 15:00	13.54229	13.54229						1981/9/29 15:00	10.42757	
690	1981/9/29 16:00	10.67389	10.67389								
691	1981/9/29 17:00	1.896301	1.896301								

100パーセンタイルに相当する値も
バイアス補正後の数値となったことを確認

4 年最大雨量の算出

「解析スタートシート」内の「雨量累積ボタン（補正後）」を押すと、各降雨量シートで時間雨量の累積計算（日・2日・3日）・年最大雨量（日・2日・3日）が算出
※ 使用するPCの環境によっては、処理に時間がかかる場合があるため、その場合は、一回に処理（追加）するシートを少なくするなどして分けて作業すること

(解析スタートシート)

10026	7. 「雨量累積ボタン」を押すと、各シートで時間雨量の累積計算・年最大雨量（日・2日・3日）の抽出が開始
10027	※使用するPCの環境によっては、処理に時間がかかる場合があるため、その場合は、一回に処理（追加）するシートを少なくするなどして分けて作業すること
10028	
10029	
10030	雨量累積ボタン（補正後）
10031	



最大雨量
(日～3日)

(降雨量シート)

H	I	J	K	L	M	N	O
1	date-hour	バイアス補正後		最大日雨量 (mm)	最大 2 日雨量 (mm)	最大 3 日雨量 (mm)	
2	1981/9/1 0:00	0		53.82963171	57.73729153	59.00581157	
3	1981/9/1 1:00	0					
4	1981/9/1 2:00	0					
5	1981/9/1 3:00	0		1981/9/1	0		
6	1981/9/1 4:00	0		1981/9/2	0	0	
7	1981/9/1 5:00	0		1981/9/3	0	0	0
8	1981/9/1 6:00	0		1981/9/4	53.82963171	53.82963171	53.82963171
9	1981/9/1 7:00	0		1981/9/5	3.907659811	57.73729153	57.73729153
10	1981/9/1 8:00	0		1981/9/6	0.020168556	3.92782367	57.75746008
11	1981/9/1 9:00	0		1981/9/7	8.918071687	8.938240242	12.84590005
12	1981/9/1 10:00	0		1981/9/8	3.427639667	12.34571135	12.36587991
13	1981/9/1 11:00	0		1981/9/9	0.480409556	3.908049222	12.82612091
14	1981/9/1 12:00	0		1981/9/10	0	0.480409556	3.908049222
15	1981/9/1 13:00	0		1981/9/11			0.480409556
16	1981/9/1 14:00	0		1981/9/12			0
17	1981/9/1 15:00	0		1981/9/13			0
18	1981/9/1 16:00	0		1981/9/14			0
19	1981/9/1 17:00	0		1981/9/15			0
20	1981/9/1 18:00	0		1981/9/16	5.922406298	5.922406298	5.922406298

累積雨量
(日～3日)

5 年最大雨量の整理

「解析スタートシート」内の「年最大雨量集計ボタン」を押すと、下表に各年の年最大雨量（日・2日・3日）が抽出

- ※ ボタンを押した際に実行時エラー表示されるが、そのまま終了ボタンを押すこと
- ※ 複数回に分けて4の作業を行った場合は、4の作業を行った全ての対象期間のシートを「解析スタート」シートの後ろに追加してから実行すること

(解析スタートシート)

10031	8.「年最大雨量集計ボタン」を押すと、下表に各年の年最大雨量（日・2日・3日）が保存されるので、値をコピーし、「確率雨量計算システム」に貼付け、確率雨量を計算する。
10032	※（複数回に分けて7の作業を行った場合は、）7の作業を行った全ての対象期間のシートを「解析スタート」シートの後ろに追加してから実行すること
10034	年最大雨量集計ボタン



(解析スタートシート)

AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS
10031																		
10032	8.「年最大雨量集計ボタン」を押すと、下表に各年の年最大雨量（日・2日・3日）が保存されるので、値をコピーし、「確率雨量計算システム」に貼付け、確率雨量を計算する。																	
10033	※（複数回に分けて7の作業を行った場合は、）7の作業を行った全ての対象期間のシートを「解析スタート」シートの後ろに追加してから実行すること																	
10034	年最大雨量集計ボタン																	
10035	最大日雨量最大2日雨量最大3日雨量																	
10036	1	対象期間の年最大日～3日降雨量が集計される	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
10037	18																	
10038																		
10039																		
10040																		
10041																		
10042																		
10043																		
10044																		
10045																		
10046																		
10047																		
10048																		
10049																		
10050																		
10051																		
10052																		
10053																		
10054																		

6 確率降雨量の算出

5で集計した値を用いて、日～3日連続雨量における1/10等の確率降雨量を算出

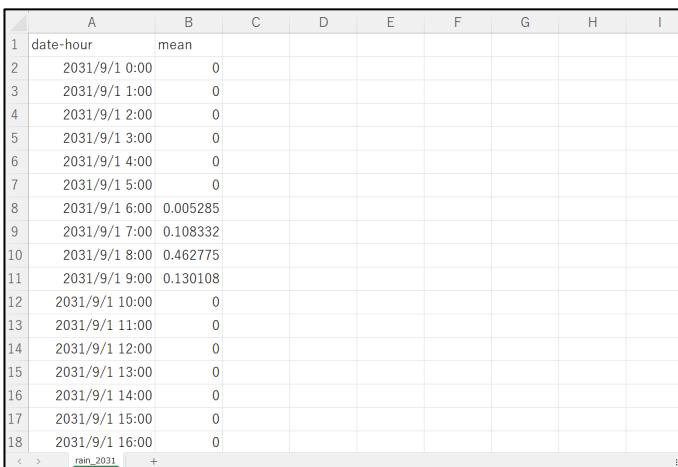
- ※ 確率降雨量の算出方法は、土地改良事業計画設計基準 計画「排水」技術書「6. 実績降雨に基づく計画基準降雨」を参照

第6章 将来実験データの処理

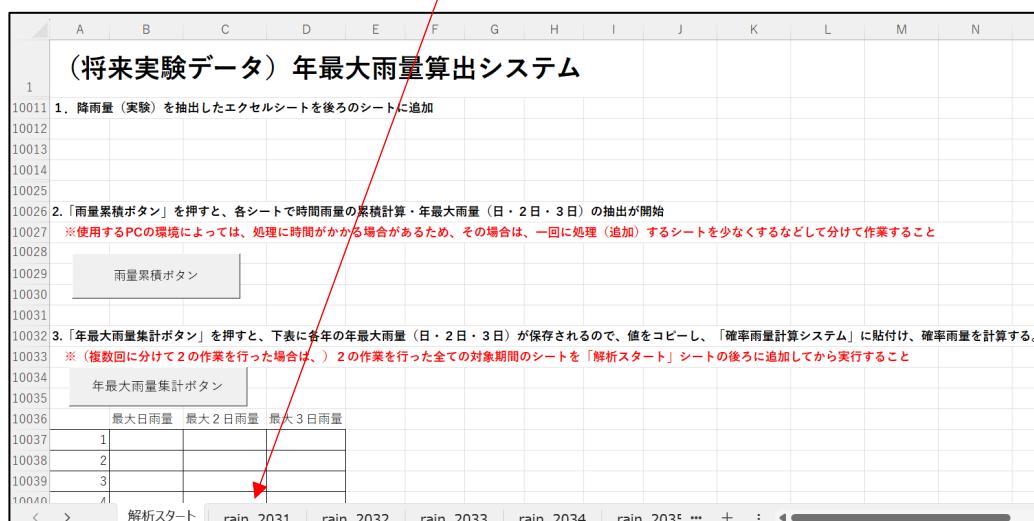
第1節 確率降雨量の算出

1 データの追加

『(将来) 年最大雨量算出システム』を開き、「解析スタート」シートの後ろ（右側）にd2PDFの降雨量（実験）エクセルシートを追加（12メンバ×60年分（2031年～2090年）



	A	B	C	D	E	F	G	H	I
1	date-hour	mean							
2	2031/9/1 0:00	0							
3	2031/9/1 1:00	0							
4	2031/9/1 2:00	0							
5	2031/9/1 3:00	0							
6	2031/9/1 4:00	0							
7	2031/9/1 5:00	0							
8	2031/9/1 6:00	0.005285							
9	2031/9/1 7:00	0.108332							
10	2031/9/1 8:00	0.462775							
11	2031/9/1 9:00	0.130108							
12	2031/9/1 10:00	0							
13	2031/9/1 11:00	0							
14	2031/9/1 12:00	0							
15	2031/9/1 13:00	0							
16	2031/9/1 14:00	0							
17	2031/9/1 15:00	0							
18	2031/9/1 16:00	0							



1. 降雨量（実験）を抽出したエクセルシートを後ろのシートに追加

2. 「雨量累積ボタン」を押すと、各シートで時間雨量の累積計算・年最大雨量（日・2日・3日）の抽出が開始
※使用するPCの環境によっては、処理に時間がかかる場合があるため、その場合は、一回に処理（追加）するシートを少なくするなどして分けて作業すること

3. 「年最大雨量集計ボタン」を押すと、下表に今年の年最大雨量（日・2日・3日）が保存されるので、値をコピーし、「確率雨量計算システム」に貼付け、確率雨量を計算する。
※（複数回に分けて2の作業を行った場合は、）2の作業を行った全ての対象期間のシートを「解析スタート」シートの後ろに追加してから実行すること

	最大日雨量	最大2日雨量	最大3日雨量
1			
2			
3			

※シート名は、どのメンバか分かるようにしたうえで追加する。

（例：HFB_2K_CC_m102→rain_2031(2)、HFB_2K_MR_m102→rain_2031(12)など）

2 年最大雨量の算出

「解析スタートシート」内の「雨量累積ボタン」を押すと、各降雨量シートで時間雨量の累積計算（日・2日・3日）・年最大雨量（日・2日・3日）が算出

※ 使用するPCの環境によっては、処理に時間がかかる場合があるため、その場合は、一回に処理（追加）するシートを少なくするなどして分けて作業すること

(解析スタートシート)

10025	
10026	2. 「雨量累積ボタン」を押すと、各シートで時間雨量の累積計算・年最大雨量（日・2日・3日）の抽出が開始
10027	※使用するPCの環境によっては、処理に時間がかかる場合があるため、その場合は、一回に処理（追加）するシートを少なくするなどして分けて作業すること
10028	
10029	雨量累積ボタン
10030	
10031	



最大雨量
(日～3日)

(降雨量シート)

A	B	C	D	E	F	G	H
1	date-hour	mean		最大日雨量 (mm)	最大2日雨量 (mm)	最大3日雨量 (mm)	
2	2031/9/1 0:00	0		90.04054067	93.69310133	101.3212879	
3	2031/9/1 1:00	0					
4	2031/9/1 2:00	0					
5	2031/9/1 3:00	0					
6	2031/9/1 4:00	0					
7	2031/9/1 5:00	0					
8	2031/9/1 6:00	0.005284667					
9	2031/9/1 7:00	0.108332333					
10	2031/9/1 8:00	0.462775333					
11	2031/9/1 9:00	0.130108222					
12	2031/9/1 10:00	0					
13	2031/9/1 11:00	0					
14	2031/9/1 12:00	0					
15	2031/9/1 13:00	0					
16	2031/9/1 14:00	0					
17	2031/9/1 15:00	0					
18	2031/9/1 16:00	0					
19	2031/9/1 17:00	0					

3 年最大雨量の整理

「解析スタートシート」内の「年最大雨量集計ボタン」を押すと、下表に各年の年最大雨量（日・2日・3日）が抽出

- ※ ボタンを押した際に実行時エラー表示されるが、そのまま終了ボタンを押すこと
- ※ 複数回に分けて2の作業を行った場合は、2の作業を行った全ての対象期間のシートを「解析スタート」シートの後ろに追加してから実行すること

(解析スタートシート)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
10031														
10032	3.	「年最大雨量集計ボタン」を押すと、下表に各年の年最大雨量（日・2日・3日）が集計												
10033	※ (複数回に分けて2までの作業を行った場合は、) 2までの作業を行った全ての対象期間のシートを「解析スタート」シートの後ろに追加してから実行すること													
10034	年最大雨量集計ボタン													
10035														



(解析スタートシート)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
10031														
10032	3.	「年最大雨量集計ボタン」を押すと、下表に各年の年最大雨量（日・2日・3日）が集計												
10033	※ (複数回に分けて2までの作業を行った場合は、) 2までの作業を行った全ての対象期間のシートを「解析スタート」シートの後ろに追加してから実行すること													
10034	年最大雨量集計ボタン													
10035														
10036		最大日雨量	最大2日雨量	最大3日雨量										
10037	1													
10038	2													
10039	3													
10040	4													
10041	5													
10042	6													
10043	7													
10044	8													
10045	9													
10046	10													
10047	11													
10048	12													
10049	13													
10050	14													
10051	15													
10052	16													
10053	17													
10054	18													
	<	>	解析スタート	rain_1981	rain_1982	rain_1983	rain_1984	rain_1985	...	+	:	◀	▶	⋮

4 確率降雨量の算出

3で集計した値を用いて、日～3日連続雨量における1/10等の確率降雨量を算出

- ※ 確率降雨量の算出方法は、土地改良事業計画設計基準 計画「排水」技術書「6. 実績降雨に基づく計画基準降雨」を参照

第2節 バイアス補正

1 時間 1mm 以上の降雨値の抽出

「解析スタートシート」内の「1mm 以上検出ボタン」を押し、各年降雨量シートの C 列に 1mm 以上の値を抽出

(解析スタートシート)

1	0	P	Q	R	S	T	U	V	W	X	Y	Z	AA
10011													
10012													
10013													
10014													

バイアス補正作業

4. 「1mm以上検出ボタン」を押すと、1mm以上の値が各年シートのC列に検出（バイアス補正是1mm以上の時間雨量で行うため）

1mm以上検出



(降雨量シート)

A	B	C	D	E	F	G	H
1	date-hour	mean	1mm以上の数値	最大日雨量 (mm)	最大 2 日雨量 (mm)	最大 3 日雨量 (mm)	
2	2031/9/1 0:00	0		90.04054067	93.69310133	101.3212879	
3	2031/9/1 1:00	0					
4	2031/9/1 2:00	0					
5	2031/9/1 3:00	0		日雨量 (mm)	2 日雨量 (mm)	3 日雨量 (mm)	
6	2031/9/1 4:00	0		2031/9/1	0.706500556		
7	2031/9/1 5:00	0		2031/9/2	0.123322222	0.829822778	
8	2031/9/1 6:00	0.005284667		2031/9/3	0.567161556	0.690483778	1.396984333
9	2031/9/1 7:00	0.108332333		2031/9/4	0	0.567161556	0.690483778
10	2031/9/1 8:00	0.462775333		2031/9/5	0.063595222	0.063595222	0.630756778
11	2031/9/1 9:00	0.130108222		2031/9/6	2.190212222	2.253807444	2.253807444
12	2031/9/1 10:00	0		2031/9/7	0.539259444	2.729471667	2.793066889
13	2031/9/1 11:00	0		2031/9/8	2.368858556	2.908118	5.098330222
14	2031/9/1 12:00	0		2031/9/9	15.22911033	17.59796889	18.13722833
15	2031/9/1 13:00	0		2031/9/10	25.80585611	41.03496644	43.403825
16	2031/9/1 14:00	0		2031/9/11	60.28632144	86.09217756	101.3212879
17	2031/9/1 15:00	0		2031/9/12	0.012042444	60.29836389	86.10422
18	2031/9/1 16:00	0		2031/9/13	0	0.012042444	60.29836389
19	2031/9/1 17:00	0		2031/9/14	0	0	0.012042444
				2031/9/15	0	0	0

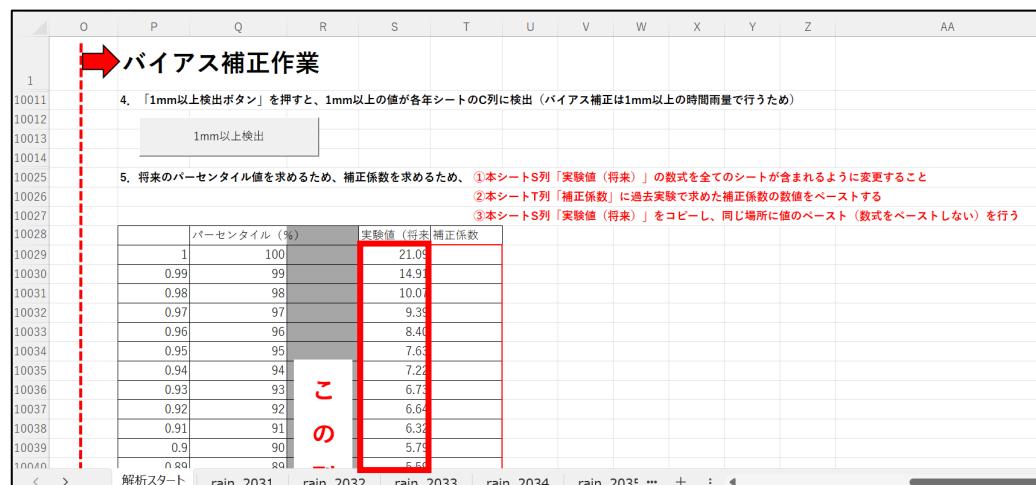
2 パーセンタイル値の算出

補正係数を求めるため、「PERCENTILE.INC」関数を使用し、1 %毎にパーセンタイル値を算出 ※ 必ず、全ての期間のシートを追加した上で作業すること

(1) 「PERCENTILE.INC」関数の適用

「解析スタートシート」内の S 列 10029 行目から 10128 行目の数式を全てのシートが対象範囲に含まれるように変更すること（例：計算するシートの 1 番目が rain_2031 で最後が rain_2090(12)であれば、S 列 10029 行目のセルの関数を「=PERCENTILE.INC(rain_2031:rain_2090(12)!\$C\$2:\$C\$9000,P10029)」と修正し、数式を S 列 10128 行目までドラッグすること）

(解析スタートシート)



	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1													
10011													
10012													
10013													
10014													
10025													
10026													
10027													
10028													
10029													
10030													
10031													
10032													
10033													
10034													
10035													
10036													
10037													
10038													
10039													
10040													
	<	>											
	解析スタート												
		rain_2031	rain_2032	rain_2033	rain_2034	rain_2035	...	+	⋮	◀			

S10029=PERCENTILE.INC(最初のシート名:最後のシート名!\$C\$2:\$C\$9000,P10029)

(2) 補正係数の引用

本シート T 列「補正係数」に別途観測値と過去実験により求めた補正係数の数値をベーストする

(解析スタートシート)

パーセンタイル(%)	実験値(将来)補正係数
1	21.09 0.8
0.99	14.91 0.77
0.98	10.07 1.08
0.97	9.39 1.13
0.96	8.40 1.07
0.95	7.63 1.08
0.94	7.22 1.16
0.93	6.73 1.14
0.92	6.64 1.05
0.91	6.32 1.02
0.9	5.79 1.02
0.89	5.50 1.02

 The formula bar shows '=PERCENTILE.INC(rain_2031!\$C\$2:\$C\$9000,P10029)'."/>

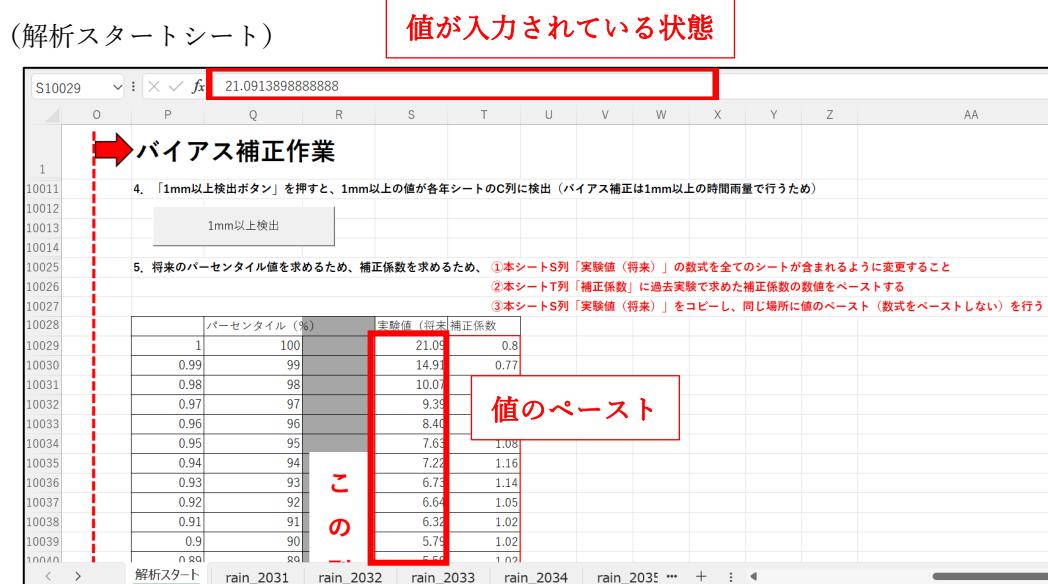
(3) 数値の貼り付け

本シート S 列「実験値 (将来)」をコピーし、同じ場所に値のペースト (数式をペーストしない) を行う

(解析スタートシート)

関数が入力されている状態

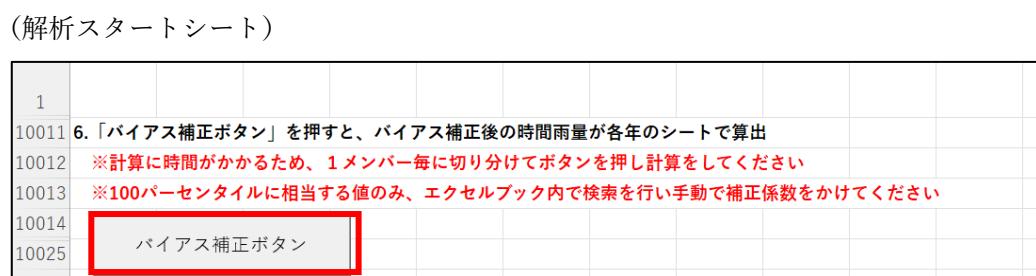




3 時間雨量の補正

「解析スタートシート」内の「バイアス補正ボタン」を押すと、各年降雨量シートのJ列にバイアス補正後の時間雨量が算出

- ※ 計算に時間がかかるため、1 メンバ毎に切り分けた計算（ボタンを押す）を推奨
 - ※ 100 パーセンタイルに相当する値のみ、エクセルブック内で検索を行い手動で補正係数をかけること（補正方法は P34 を参照）



(降雨量シート)

バイアス補正後の数値

A	B	C	D	E	F	G	A	B
1	date-hour	mean	1mm以上の数値	最大日雨量	最大2日雨量	最大3日雨量 (mm)	date-hour	バイアス補正後
2	2031/9/1 0:00	0		90.0405	93.6931	101.321	2031/9/1 0:00	0
3	2031/9/1 1:00	0					2031/9/1 1:00	0
4	2031/9/1 2:00	0					2031/9/1 2:00	0
5	2031/9/1 3:00	0	2031/9/1	0.7065			2031/9/1 3:00	0
6	2031/9/1 4:00	0	2031/9/2	0.12332	0.82982		2031/9/1 4:00	0
7	2031/9/1 5:00	0	2031/9/3	0.56716	0.69048	1.39698	2031/9/1 5:00	0
8	2031/9/1 6:00	0.00528	2031/9/4	0	0.56716	0.69048	2031/9/1 6:00	0.00528
9	2031/9/1 7:00	0.10833	2031/9/5	0.0636	0.0636	0.63076	2031/9/1 7:00	0.10833
10	2031/9/1 8:00	0.46278	2031/9/6	2.19021	2.25381	2.25381	2031/9/1 8:00	0.46278
11	2031/9/1 9:00	0.13011	2031/9/7	0.53926	2.72947	2.79307	2031/9/1 9:00	0.13011
12	2031/9/1 10:00	0	2031/9/8	2.36886	2.90812	5.09833	2031/9/1 10:00	0
13	2031/9/1 11:00	0	2031/9/9	15.2291	17.598	18.1372	2031/9/1 11:00	0
14	2031/9/1 12:00	0	2031/9/10	25.8059	41.035	43.4038	2031/9/1 12:00	0
15	2031/9/1 13:00	0	2031/9/11	60.2863	86.0922	101.321	2031/9/1 13:00	0
16	2031/9/1 14:00	0	2031/9/12	0.01204	60.2984	86.1042	2031/9/1 14:00	0
17	2031/9/1 15:00	0	2031/9/13	0	0.01204	60.2984	2031/9/1 15:00	0
18	2031/9/1 16:00	0	2031/9/14	0	0	0.01204	2031/9/1 16:00	0
19	2031/9/1 17:00	0	2031/9/15	0	0	0	2031/9/1 17:00	0
20	2031/9/1 18:00	0	2031/9/16	29.4332	29.4332	29.4332	2031/9/1 18:00	0
21	2031/9/1 19:00	0	2031/9/17	1.33524	30.7684	30.7684	2031/9/1 19:00	0
22	2031/9/1 20:00	0	2031/9/18	1.39441	2.72965	32.1628	2031/9/1 20:00	0

他の追加シート · · · · ·

4 年最大雨量の算出

「解析スタートシート」内の「雨量累積ボタン（補正後）」を押すと、各降雨量シートで時間雨量の累積計算（日・2日・3日）・年最大雨量（日・2日・3日）が算出

※ 使用するPCの環境によっては、処理に時間がかかる場合があるため、その場合は、一

回に処理（追加）するシートを少なくするなどして分けて作業すること

（解析スタートシート）

10027	7.「雨量累積ボタン」を押すと、各シートで時間雨量の累積計算・年最大雨量（日・2日・3日）の抽出が開始
10028	※使用するPCの環境によっては、処理に時間がかかる場合があるため、その場合は、一回に処理（追加）するシートを少なくするなどして分けて作業すること
10029	
10030	雨量累積ボタン（補正後）
10031	



（降雨量シート）

A	B	C	D	E	F	G	H	I	J	M	N	O
1	date-hour	mean	1mm以上の数値	最大日雨量 (mm)	最大2日雨量 (mm)	最大3日雨量 (mm)	date-hour	バイアス補正後	最大日雨量	最大2日雨量	最大3日雨量 (mm)	
2	2031/9/1 0:00	0		90.0405	93.6931	101.321	2031/9/1 0:00	0	90.49273	94.31775	106.3963	
3	2031/9/1 1:00	0					2031/9/1 1:00	0				
4	2031/9/1 2:00	0					2031/9/1 2:00	0				
5	2031/9/1 3:00	0	2031/9/1	0.706500556			2031/9/1 3:00	0				
6	2031/9/1 4:00	0	2031/9/2	0.12332222	0.82982778		2031/9/1 4:00	0				
7	2031/9/1 5:00	0	2031/9/3	0.567161556	0.690483778	1.396984333	2031/9/1 5:00	0				
8	2031/9/1 6:00	0.005284667	2031/9/4	0	0.567161556	0.690483778	2031/9/1 6:00	0.00528467	2031/9/3	0.567162	0.690484	1.396984
9	2031/9/1 7:00	0.108332333	2031/9/5	0.063595222	0.063595222	0.630756778	2031/9/1 7:00	0.063595223	2031/9/5	0.063595	0.063595	0.630757
10	2031/9/1 8:00	0.462775333	2031/9/6	2.19021222	2.253807444	2.253807444	2031/9/1 8:00	0.462775333	2031/9/6	2.190212	2.253807	2.253807
11	2031/9/1 9:00	0.130108222	2031/9/7	0.539259444	2.729471657	2.793066889	2031/9/1 9:00	0.130108222	2031/9/7	0.539259	2.729472	2.793067
12	2031/9/1 10:00	0	2031/9/8	2.368858556	2.90818	5.098330222	2031/9/1 10:00	0	2031/9/8	2.527494	3.066758	5.25697
13	2031/9/1 11:00	0	2031/9/9	15.22911033	17.59796889	18.13728333	2031/9/1 11:00	0	2031/9/9	15.82922	18.35672	18.89598
14	2031/9/1 12:00	0	2031/9/10	25.80585611	41.03496444	43.403825	2031/9/1 12:00	0	2031/9/10	25.9264	41.75567	42.28316
15	2031/9/1 13:00	0	2031/9/11	60.28632144	86.09217756	101.3212879	2031/9/1 13:00	0	2031/9/11	64.64061	90.56705	106.3963
16	2031/9/1 14:00	0	2031/9/12	0.012042444	60.29836399	86.10422	2031/9/1 14:00	0	2031/9/12	64.65265	90.5791	
17	2031/9/1 15:00	0	2031/9/13	0	0.012042444	60.29836389	2031/9/1 15:00	0	2031/9/13	0	0.012042	
18	2031/9/1 16:00	0	2031/9/14	0	0	0.012042444	2031/9/1 16:00	0	2031/9/14	0	0	
19	2031/9/1 17:00	0	2031/9/15	0	0	0	2031/9/1 17:00	0				
20	2031/9/1 18:00	0	2031/9/16	29.43319211	29.43319211	29.43319211	2031/9/1 18:00	0				
21	2031/9/1 19:00	0	2031/9/17	1.335237333	30.76842944	30.76842944	2031/9/1 19:00	0				
22	2031/9/1 20:00	0	2031/9/18	1.394414222	2.729851556	32.16284367	2031/9/1 20:00	0				
23	2031/9/1 21:00	0	2031/9/19	0.539176222	1.933590444	3.268827778	2031/9/1 21:00	0				
24	2031/9/1 22:00	0	2031/9/20	0.000756222	0.539932444	1.934346667	2031/9/22:00	0				
25	2031/9/1 23:00	0	2031/9/21	4.190256444	4.191012667	4.730188889	2031/9/23:00	0	2031/9/21	4.30351	4.204269	4.843442
26	2031/9/2 0:00	0	2031/9/22	0.109721111	4.299983556	4.300739778	2031/9/2:00	0	2031/9/22	0.109727	4.413237	4.413993
27	2031/9/2 1:00	0	2031/9/23	6.77778E-06	0.109733889	4.299990333	2031/9/21:00	0	2031/9/23	6.78E-06	0.109734	4.413244

累積雨量
(日～3日)

5 年最大雨量の整理

「解析スタートシート」内の「年最大雨量集計ボタン」を押すと、下表に各年の年最大雨量（日・2日・3日）が抽出

- ※ ボタンを押した際に実行時エラー表示されるが、そのまま終了ボタンを押すこと
- ※ 複数回に分けて4の作業を行った場合は、4の作業を行った全ての対象期間のシートを「解析スタート」シートの後ろに追加してから実行すること

(解析スタートシート)

10032	8. 「年最大雨量集計ボタン」を押すと、下表に各年の年最大雨量（日・2日・3日）が保存されるので、値をコピーし、「確率雨量計算システム」に貼付け、確率雨量を計算する。
10033	※（複数回に分けて7の作業を行った場合は、）7の作業を行った全ての対象期間のシートを「解析スタート」シートの後ろに追加してから実行すること
10034	年最大雨量集計ボタン



(解析スタートシート)

10032	8. 「年最大雨量集計ボタン」を押すと、下表に各年の年最大雨量（日・2日・3日）が保存されるので、値をコピーし、「確率雨量計算システム」に貼付け、確率雨量を計算する。
10033	※（複数回に分けて7の作業を行った場合は、）7の作業を行った全ての対象期間のシートを「解析スタート」シートの後ろに追加してから実行すること
10034	年最大雨量集計ボタン
10035	最大日雨量最大2日雨量最大3日雨量
10036	「降雨量シート」で1年毎に計算された最大日～3日雨量が、シート追加順に表示
10037	
10038	
10039	
10040	
10041	
10042	
10043	
10044	
10045	
10046	
10047	
10048	
10049	
10050	
10051	

6 確率降雨量の算出

5で集計した値を用いて、日～3日連続雨量における1/10等の確率降雨量を算出

- ※ 確率降雨量の算出方法は、土地改良事業計画設計基準 計画「排水」技術書「6. 実績降雨に基づく計画基準降雨」を参照

(附属資料 1)

実験データの入手

気候予測資料（データセット）の d2PDF（5 km）データは、本省から地方農政局等を通じて貸与するが、DIAS より入手する場合は、以下の手順で行う。

(DIAS HP <https://diasjp.net/>)



The screenshot shows the DIAS homepage with a central banner asking 'したいこと?' (What do you want to do?) and 'さがしもの?' (What are you looking for?). Below the banner are four blue boxes: 'データセットを取得 (ダウンロード) したい' (Want to get a data set (download)), 'データを解析・加工したい' (Want to analyze and process data), 'データセットを公開したい' (Want to publish a data set), and 'アプリケーションを利用したい' (Want to use an application). Below these are three more boxes: 'DIASデータ俯瞰検索システム' (DIAS Data Overview Search System), 'DIAS公開データセット一覧' (DIAS Public Data Set Catalog), and 'DIASアプリケーション一覧' (DIAS Application Catalog). The 'DIAS公開データセット一覧' box is highlighted with a red border.

(DIAS 「公開データセット一覧」)



The screenshot shows the 'DIAS公開データセット一覧' page. The header includes the DIAS logo and the text 'データ俯瞰・検索システム' (Dataset Search and Discovery). Below the header is a navigation bar with 'ホーム', '使い方', and 'このサイトについて'. A sidebar on the left lists data sets: 'CRSS_JPN200K', 'CRSS_TY_DDS', 'd4PDF_5kmDDS_Hokkaido', 'd4PDF_5kmDDS_JP', 'd4PDF_CDFM_S14RF', 'd4PDF_crop_yields', 'd4PDF_explosive_cyclone', 'd4PDF_GCM', 'd4PDF_RCM', 'd4PDF_RCM_3D_Plev', 'd4PDF_tropical_cyclone', and 'D8NDVI_J'. The main content area is titled '「DIAS公開データセット一覧」' and '「全国5kmメッシュアンサンブル気候予測データ」'. A table lists data sets with their descriptions and creators. The '全国5kmメッシュアンサンブル気候予測データ' entry is highlighted with a red box.

データセット名	データセット作成者の組織名
CRSS_JPN200K	名古屋大学宇宙地球環境研究所
CRSS_TY_DDS	名古屋大学 宇宙地球環境研究所
d4PDF_5kmDDS_Hokkaido	北海道大学
d4PDF_5kmDDS_JP	気象庁気象研究所
d4PDF_CDFM_S14RF	環境機構・農業環境変動研究センター
d4PDF_crop_yields	農研機構・農業環境変動研究センター
d4PDF_explosive_cyclone	統合的気候モデル・高度化研究プログラム
d4PDF_GCM	気候変動リスク情報創生プログラム
d4PDF_RCM	気候変動リスク情報創生プログラム
d4PDF_RCM_3D_Plev	海洋研究開発機構
d4PDF_tropical_cyclone	統合的気候モデル・高度化研究プログラム
D8NDVI_J	独立行政法人 海洋研究開発機構

(DIAS 「全国5kmメッシュアンサンブル気候予測データの諸元、規約画面」)

DIAS データ俯瞰・検索システム
Dataset Search and Discovery

全国5kmメッシュアンサンブル気候予測データ

このデータセットの引用文
川瀬宏明. (2023). 全国5kmメッシュアンサンブル気候予測データ [Data set]. データ統合・解析システム(DIAS). <https://doi.org/10.20783/DIAS.657>

このデータセットを引用した論文
Mahalo Buttonとは?

識別情報

名称	全国5kmメッシュアンサンブル気候予測データ
DOI	doi:10.20783/DIAS.657
メタデータID	d4PDF_5kmDDS_JP20240128190245+DIAS20221121113753-ja

問合せ先

データセットに関する問合せ先

名前	川瀬宏明
組織名	気象庁気象研究所
住所	日本 305-0052 英崎郡 つくば市 長崎 1-1



(DIAS 「ログイン画面」)

DIASサービスを利用する際にアカウント登録が必要

DIAS ログイン
Authentication for DIAS Systems

English 日本語 Français

DIASサービス利用規約が改定されました (2022年 11月)

下記よりログイン頂いた場合、改定後のDIASサービス利用規約に同意いただいたものとみなします。同意いただけない場合はアカウントを削除いたしますので DIAS事務局 (dias-office@diasjp.net) までお問い合わせください。

メールアドレスおよびパスワードを入力してください

1.アカウント登録

2.登録したアカウント情報を入力

メールアドレス:

パスワード:

ログイン

■ダウンロードするデータ

- ・過去実験：12 メンバ×60 年分（1951 年 9 月 1 日～2011 年 8 月 31 日）の内、観測値で選定した期間の時間降雨量データ
- ・将来実験（2 °C上昇時点）：12 メンバ×60 年分（2031 年 9 月 1 日～2091 年 8 月 31 日）の時間降雨量データ
- ※ 将来実験（2 °C上昇時点）では、産業革命以降 2 °C上昇時点（2040 年頃）の予測を行っており、データの年月日は便宜上設定されている。

（DIAS 「全国 5km メッシュアンサンブル気候予測データ ダウンロード画面」）

ディレクトリ指定

キーワード指定

単語一致

ファイル検索

0 / 1ページ, 1000 / レコード, 開始レコード 0, 終了レコード

タイトル

ファイルサイズ

ディレクトリ指定

キーワード指定

検索モードによる違いは以下の通りです。続き

① 各メンバ→各年→hourly を選択し、②ファイル検索を押す

② ファイル検索

0 / 1ページ, 1000 / レコード, 開始レコード 0, 終了レコード

タイトル

ファイルサイズ



1 / 1ページ, 64 / 64 レコード, 開始レコード 1 , 終了レコード 64	
タイトル	ファイル名
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > cla.nc	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > cla.nc_pdef.ctl	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > clb.nc	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > clb.nc_pdef.ctl	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > clc.nc	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > clc.nc_pdef.ctl	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > clm.nc	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > clm.nc_pdef.ctl	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > psea.nc	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > psea.nc_pdef.ctl	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > psurf.nc	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > psurf.nc_pdef.ctl	1
<input checked="" type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > rain.nc	1
<input checked="" type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > rain.nc_pdef.ctl	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > smqo.nc	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > smqr.nc_pdef.ctl	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > smqs.nc	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > smqs.nc_pdef.ctl	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > smcvr.nc	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > smcvr.nc_pdef.ctl	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > sndep.nc	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > sndep.nc_pdef.ctl	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > snswe.nc	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > snswe.nc_pdef.ctl	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > t.nc	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > tnc_pdef.ctl	1
<input type="checkbox"/> d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > tin1.nc	1



一括ダウンロード

選択ファイル 2個 2024544748バイト

ダウンロード	全チェック	全取消
取消	d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > rain.nc	
取消	d4PDF_5kmDDS_JP > HFB_2K_CC_m101 > 2031 > hourly > rain.nc_pdef.ctl	

<< 前へ | 次へ >>

画面の下部に、2個のファイルが選択されていることを確認し、ダウンロードボタンを押す



上記を 60 年分×24 メンバ (過去実験 12 メンバ+将来実験 12 メンバ) を繰り返し行い、データのダウンロードを行う。



名前	更新日時	種類
HFB_2K_CC_m101	2023/11/02 19:32	ファイル フォルダー
HFB_2K_CC_m102	2023/11/02 19:32	ファイル フォルダー
HFB_2K_GF_m101	2023/11/02 19:32	ファイル フォルダー
HFB_2K_GF_m102	2023/11/02 19:32	ファイル フォルダー
HFB_2K_HA_m101	2023/11/02 19:32	ファイル フォルダー
HFB_2K_HA_m102	2023/11/02 19:32	ファイル フォルダー
HFB_2K_MI_m101	2023/11/02 19:32	ファイル フォルダー
HFB_2K_MI_m102	2023/11/02 19:32	ファイル フォルダー
HFB_2K_MP_m101	2023/11/02 19:33	ファイル フォルダー
HFB_2K_MP_m102	2023/11/02 19:33	ファイル フォルダー
HFB_2K_MR_m101	2023/11/02 19:33	ファイル フォルダー
HFB_2K_MR_m102	2023/11/02 19:33	ファイル フォルダー
HPB_m001	2023/11/02 19:33	ファイル フォルダー
HPB_m002	2023/11/02 19:33	ファイル フォルダー

ダウンロードしたデータは、メンバ毎にフォルダにまとめておくと後に作業がしやすくなる

(附属資料 2)

d2PDF データを csv 化する際の Python プログラムコード

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""

netcdf2csv_arealmean_cnst.py — cnst.nc の flon/flat を唯一の位置情報として用い、
セル定義を選択 (voronoi) して shapefile と重なり率を計算し、
対象セルの coverage と降水 CSV (全格子・面積加重平均) を出力する。
```

主な仕様:

- 位置情報: cnst.nc の flon/flat を使用。
- セル定義: --cell-def voronoi|dual|primal (default=voronoi)
 - * voronoi: 各格子点をサイトとする Voronoi 多角形 (等距離直線の分割)。
投影空間(EPSG:3857)で有限化&クリップ&有効化し、WGS84 に逆変換。
 - * dual : 各格子点中心の東西南北“中点”四角形 (FCT=499 サンプリング近似)。
 - * primal : 通常セル ((j,i)-(j,i+1)-(j+1,i+1)-(j+1,i)) の四角形。
- 抽出: coverage > --min-coverage のセルを採用。
- 時刻: netCDF4.num2date で cftime 対応。--shift-hours で補正可能。
- rain 形状: (time, member, j, i) または (time, j, i) に対応 (member=0 固定)。
- 出力: target_coverage.csv は「中心点 (i,j)」をセル名で統一 (--index-base で 0/1 始まり)。
 - 追加出力: --export-cells-shp で採用セルのポリゴン出力 (shp/gpkg/geojson)。

使い方 (例) :

```
python netcdf2csv_arealmean_cnst.py ../../hourly/HPB_m009/ -o ./csv_cnst \
-s ./murayama_hokubu_wgs84.shp --min-coverage 0 \
--export-cells-shp ./csv_cnst/HPB_m009/selected_cells.shp
....
```

```
import time
import os
import sys
import glob
import argparse
import numpy as np
import pandas as pd
import datetime as dt
from datetime import timedelta
from netCDF4 import Dataset, num2date
```

```

import shapefile # pyshp
from matplotlib.path import Path

# 依存（ある場合のみ使用）
try:
    from scipy.spatial import Voronoi
    _HAVE SCIPY = True
except Exception:
    _HAVE SCIPY = False

try:
    import geopandas as gpd
    from shapely.geometry import Polygon, LineString, Point as ShpPoint, MultiPoint
    from shapely.ops import unary_union, transform
    from shapely.errors import TopologicalError
    try:
        from shapely.validation import make_valid # shapely>=2
        _HAVE_MAKE_VALID = True
    except Exception:
        _HAVE_MAKE_VALID = False
    _HAVE_GPD = True
except Exception:
    _HAVE_GPD = False
    _HAVE_MAKE_VALID = False

try:
    import pyproj
    from pyproj import Transformer
    _HAVE_PYPROJ = True
except Exception:
    _HAVE_PYPROJ = False

from functools import partial

# -----
# ユーティリティ
# -----
def ensure_2d(a, name):
    a = np.asarray(a)
    orig = a.shape
    a = np.squeeze(a)
    while a.ndim > 2:
        a = a[0]
    if a.ndim != 2:
        raise ValueError(f"{name} must be 2D, got {orig} -> {a.shape}")

```

```

    return a

def find_rain_files(input_dir: str):
    return sorted(glob.glob(os.path.join(input_dir, "**", "rain.nc"), recursive=True))

def find_cnst_for(rain_nc_path: str, max_up: int = 2) -> str:
    d = os.path.dirname(rain_nc_path)
    for _ in range(max_up + 1): # 0=同階層, 1=1つ上, ...
        cand = os.path.join(d, "cnst.nc")
        if os.path.exists(cand):
            return cand
        parent = os.path.dirname(d)
        if parent == d:
            break
        d = parent
    raise FileNotFoundError(f"cnst.nc not found within {max_up} levels up from: {rain_nc_path}")

def load_lonlat_from_cnst(cnst_path: str):
    ds = Dataset(cnst_path, "r")
    lon_candidates = ["flon", "lon", "g0_lon", "xx", "x", "longitude"]
    lat_candidates = ["flat", "lat", "g0_lat", "yy", "y", "latitude"]
    lon_name = next((v for v in lon_candidates if v in ds.variables), None)
    lat_name = next((v for v in lat_candidates if v in ds.variables), None)
    if lon_name is None or lat_name is None:
        keys = list(ds.variables.keys())
        ds.close()
        raise RuntimeError(f"cnst.nc に lon/lat 変数が見つかりません: {keys}")
    lon2d = ensure_2d(ds.variables[lon_name][:], lon_name)
    lat2d = ensure_2d(ds.variables[lat_name][:], lat_name)
    ds.close()
    if lon2d.shape != lat2d.shape:
        raise ValueError(f"lon/lat shape mismatch: {lon2d.shape} vs {lat2d.shape}")
    return lon2d, lat2d # (J, I)

def read_polygon_paths(shp_path: str):
    sf = shapefile.Reader(shp_path, encoding="utf-8", encodingErrors="ignore")
    paths = []
    shapely_polys = []
    for shp in sf.shapes():
        parts = list(shp.parts) + [len(shp.points)]
        for a, b in zip(parts[:-1], parts[1:]):

```

```

pts = np.array(shp.points[a:b])
if pts.size == 0:
    continue
if np.max(np.abs(pts[:, 0])) > 360 or np.max(np.abs(pts[:, 1])) > 90:
    raise ValueError("Shapefile の CRS は EPSG:4326 (lon/lat) にしてください。")
paths.append(Path(pts))
if _HAVE_GPD:
    try:
        shapely_polys.append(Polygon(pts))
    except Exception:
        pass
if not paths:
    raise ValueError("Shapefile 内に有効なポリゴンが見つかりません。")
return paths, shapely_polys

def bbox_from_paths(paths):
    lon_min = +1e30; lon_max = -1e30
    lat_min = +1e30; lat_max = -1e30
    for p in paths:
        v = p.vertices
        lon_min = min(lon_min, float(np.min(v[:, 0]))); lon_max = max(lon_max,
float(np.max(v[:, 0])))
        lat_min = min(lat_min, float(np.min(v[:, 1]))); lat_max = max(lat_max,
float(np.max(v[:, 1])))
    return lon_min, lon_max, lat_min, lat_max

def median_grid_step(a2d, axis):
    a2d = np.asarray(a2d)
    if axis == 1 and a2d.shape[1] > 1:
        dif = np.abs(a2d[:, 1:] - a2d[:, :-1]).ravel()
    elif axis == 0 and a2d.shape[0] > 1:
        dif = np.abs(a2d[1:, :] - a2d[:-1, :]).ravel()
    else:
        return 0.05
    dif = dif[~np.isnan(dif)]
    return float(np.median(dif)) if dif.size else 0.05

def nearest_cell_by_point(center_lon2d, center_lat2d, lat, lon):
    d2 = (center_lat2d - lat) ** 2 + (center_lon2d - lon) ** 2
    j, i = np.unravel_index(np.argmin(d2), d2.shape)
    return np.array([[j, i]])

```

```

# -----
# dual coverage (中点四角形 × ポリゴン)
# -----


def coverage_fraction_dual(lon2d, lat2d, sel_jis, paths, FCT=499):
    n = len(sel_jis)
    if n == 0:
        return np.array([], dtype=float)
    if FCT <= 1:
        return np.ones(n, dtype=float)

    J, I = lon2d.shape
    step = 1.0 / FCT
    offs = np.linspace(-0.5 + step / 2, 0.5 - step / 2, FCT)
    uu, vv = np.meshgrid(offs, offs, indexing="xy")
    uv = np.column_stack([uu.ravel(), vv.ravel()])

    fracs = np.empty(n, dtype=float)

    for k, (j, i) in enumerate(sel_jis):
        iW = max(i - 1, 0); iE = min(i + 1, I - 1)
        jN = max(j - 1, 0); jS = min(j + 1, J - 1)
        lonE = 0.5 * (lon2d[j, i] + lon2d[j, iE]); latE = 0.5 * (lat2d[j, i] + lat2d[j, iE])
        lonW = 0.5 * (lon2d[j, iW] + lon2d[j, i]); latW = 0.5 * (lat2d[j, iW] + lat2d[j, i])
        lonS = 0.5 * (lon2d[jS, i] + lon2d[j, i]); latS = 0.5 * (lat2d[jS, i] + lat2d[j, i])
        lonN = 0.5 * (lon2d[j, i] + lon2d[jN, i]); latN = 0.5 * (lat2d[j, i] + lat2d[jN, i])
        C = np.array([[lonE, latE], [lonS, latS], [lonW, latW], [lonN, latN]], dtype=float)
        u = (uv[:, 0] + 0.5); v = (uv[:, 1] + 0.5)
        w0 = (1 - u) * (1 - v); w1 = u * (1 - v); w2 = u * v; w3 = (1 - u) * v
        lon_sub = w0 * C[0, 0] + w1 * C[1, 0] + w2 * C[2, 0] + w3 * C[3, 0]
        lat_sub = w0 * C[0, 1] + w1 * C[1, 1] + w2 * C[2, 1] + w3 * C[3, 1]
        pts = np.column_stack([lon_sub, lat_sub])
        inside = np.zeros(pts.shape[0], dtype=bool)
        for p in paths:
            inside |= p.contains_points(pts)
        fracs[k] = inside.mean()

    return fracs

# -----
# Voronoi (等距離直線の分割セル) ヘルパ
# -----


def _project_xy(lon, lat, lat0=None):
    lon = np.asarray(lon); lat = np.asarray(lat)
    if _HAVE_PYPROJ:

```

```

tf = Transformer.from_crs("EPSG:4326", "EPSG:3857", always_xy=True)
x, y = tf.transform(lon, lat)
return np.asarray(x), np.asarray(y)

if lat0 is None:
    lat0 = float(np.nanmean(lat))
R = 6371000.0
x = np.deg2rad(lon) * R * np.cos(np.deg2rad(lat0))
y = np.deg2rad(lat) * R
return x, y

def _voronoi_finite_polygons_2d(vor, radius=None):
    if vor.points.shape[1] != 2:
        raise ValueError("Requires 2D input")
    new_regions = []
    new_vertices = vor.vertices.tolist()
    center = vor.points.mean(axis=0)
    if radius is None:
        radius = np.ptp(vor.points, axis=0).max()*2
    all_ridges = {}
    for (p1, p2), (v1, v2) in zip(vor.ridge_points, vor.ridge_vertices):
        all_ridges.setdefault(p1, []).append((p2, v1, v2))
        all_ridges.setdefault(p2, []).append((p1, v1, v2))
    for p1, region_idx in enumerate(vor.point_region):
        vertices = vor.regions[region_idx]
        if all(v >= 0 for v in vertices):
            new_regions.append(vertices); continue
        ridges = all_ridges[p1]
        new_region = [v for v in vertices if v >= 0]
        for p2, v1, v2 in ridges:
            if v2 < 0: v1, v2 = v2, v1
            if v1 >= 0 and v2 >= 0: continue
            t = vor.points[p2] - vor.points[p1]
            t /= np.linalg.norm(t)
            n = np.array([-t[1], t[0]])
            midpoint = vor.points[[p1, p2]].mean(axis=0)
            direction = np.sign(np.dot(midpoint - center, n)) * n
            far_point = vor.vertices[v2] + direction * (radius if radius is not None else 1.0)
            new_vertices.append(far_point.tolist())
            new_region.append(len(new_vertices)-1)
        vs = np.array([new_vertices[v] for v in new_region])
        c = vs.mean(axis=0)
        ang = np.arctan2(vs[:,1]-c[1], vs[:,0]-c[0])
        new_region = [v for _, v in sorted(zip(ang, new_region))]
        new_regions.append(new_region)
    return new_regions, np.asarray(new_vertices)

```

```

def build_voronoi_cells(lon2d, lat2d, sel_jis):
    if not _HAVE SCIPY or not _HAVE_GPD:
        raise RuntimeError("Voronoi requires scipy + shapely/geopandas")
    J, I = lon2d.shape
    lon_flat = lon2d.ravel(); lat_flat = lat2d.ravel()
    x, y = _project_xy(lon_flat, lat_flat)
    vor = Voronoi(np.column_stack([x, y]))
    regions, vertices = _voronoi_finite_polygons_2d(vor)

    # 投影空間で外枠（凸包+余白）を作ってクリップ
    mp_proj = MultiPoint([(xx, yy) for xx, yy in zip(x, y)])
    hull_proj = mp_proj.convex_hull.buffer(20000.0)
    minx, miny, maxx, maxy = hull_proj.bounds
    pad = 50000.0
    bbox_proj = Polygon([(minx-pad, miny-pad), (maxx+pad, miny-pad), (maxx+pad, maxy+pad), (minx-pad, maxy+pad)])
    try:
        hull_proj = hull_proj.intersection(bbox_proj)
    except TopologicalError:
        hull_proj = hull_proj.buffer(0).intersection(bbox_proj)

    if _HAVE_PYPROJ:
        tf_inv = pyproj.Transformer.from_crs("EPSG:3857", "EPSG:4326",
                                              always_xy=True)
        to_wgs84 = partial(tf_inv.transform)
    else:
        to_wgs84 = None

    polys = [None] * (J*I)
    for p_idx, reg in enumerate(regions):
        ring_xy = vertices[reg]
        poly_proj = Polygon(ring_xy)
        try:
            poly_proj = poly_proj.intersection(hull_proj)
        except TopologicalError:
            poly_proj = poly_proj.buffer(0).intersection(hull_proj)
        if not poly_proj.is_valid:
            poly_proj = make_valid(poly_proj) if _HAVE_MAKE_VALID else
            poly_proj.buffer(0)
        poly_ll = transform(to_wgs84, poly_proj) if to_wgs84 is not None else poly_proj
        polys[p_idx] = poly_ll

    out = []
    for j, i in sel_jis:

```

```

        out.append(polys[j*I + i])
    return out

#
# -----#
# セルポリゴン出力 (任意)
# -----#


def export_selected_cell_polys(out_path, cell_def, lon2d, lat2d, sel_jis, cov, index_base=0):
    if not _HAVE_GPD:
        raise RuntimeError("--export-cells-shp requires geopandas/shapely")
    J, I = lon2d.shape
    geoms = []
    recs = []
    if cell_def == "voronoi":
        polys = build_voronoi_cells(lon2d, lat2d, sel_jis)
        geoms = polys
    elif cell_def == "primal":
        for (j, i) in sel_jis:
            i1 = min(i+1, I-1); j1 = min(j+1, J-1)
            ring = [(lon2d[j, i], lat2d[j, i]),
                     (lon2d[j, i1], lat2d[j, i1]),
                     (lon2d[j1, i1], lat2d[j1, i1]),
                     (lon2d[j1, i], lat2d[j1, i]),
                     (lon2d[j, i], lat2d[j, i])]
            geoms.append(Polygon(ring))
    else: # dual
        for (j, i) in sel_jis:
            iW = max(i - 1, 0); iE = min(i + 1, I - 1)
            jN = max(j - 1, 0); jS = min(j + 1, J - 1)
            E = (0.5 * (lon2d[j, i] + lon2d[j, iE]), 0.5 * (lat2d[j, i] + lat2d[j, iE]))
            S = (0.5 * (lon2d[jS, i] + lon2d[j, i]), 0.5 * (lat2d[jS, i] + lat2d[j, i]))
            W = (0.5 * (lon2d[j, iW] + lon2d[j, i]), 0.5 * (lat2d[j, iW] + lat2d[j, i]))
            N = (0.5 * (lon2d[j, i] + lon2d[jN, i]), 0.5 * (lat2d[j, i] + lat2d[jN, i]))
            ring = [E, S, W, N, E]
            geoms.append(Polygon(ring))
    for (j, i), fr in zip(sel_jis, cov):
        recs.append({
            "i": int(i + (1 if index_base == 1 else 0)),
            "j": int(j + (1 if index_base == 1 else 0)),
            "fraction_area": float(fr)
        })
    gdf = gpd.GeoDataFrame(recs, geometry=geoms, crs="EPSG:4326")
    ext = os.path.splitext(out_path)[1].lower()
    if ext == ".gpkg":
        gdf.to_file(out_path, layer="cells", driver="GPKG")
    elif ext in (".geojson", ".json"):

```

```

        gdf.to_file(out_path, driver="GeoJSON")
    else:
        gdf.to_file(out_path)

# -----
# メイン処理
# -----
# -----



def main():
    ap = argparse.ArgumentParser(
        formatter_class=argparse.RawTextHelpFormatter,
        description=(
            "cnst.nc の flon/flat + 指定セル定義で格子選択し、"
            "rain.nc から時系列 CSV 出力 (全格子・面積加重平均)。"
        ),
    )
    ap.add_argument("inputs", nargs="+", help="入力ディレクトリ (再帰で **/rain.nc を探索) ")
    ap.add_argument("-o", "--output", required=True, help="出力ディレクトリ")
    ap.add_argument("-s", "--shapefile", default=None, help="対象ポリゴン shapefile (EPSG:4326)。-p と排他")
    ap.add_argument("-p", "--point", default=None, help="単一点 'lat,lon' (例: 36.03,140.09)。-s と排他")
    ap.add_argument("--shift-hours", type=int, default=0, help="読み込んだ日時を指定時間だけシフト (例: -48)")
    ap.add_argument("--index-base", type=int, choices=[0, 1], default=0,
    help="target_coverage の i,j を 0/1 始まりで出力 (default=0)")
    ap.add_argument("--cell-def", choices=["voronoi", "dual", "primal"], default="voronoi",
    help="セル定義を選択")
    ap.add_argument("--min-coverage", type=float, default=1e-12, help="採用する最小被覆率 (default=1e-12). 0 で『少しでも重なれば』")
    ap.add_argument("--export-cells-shp", default=None, help="採用セルの範囲ポリゴンを出力 (shp/gpkg/geojson)")

    args = ap.parse_args()

    if (args.shapefile is None) == (args.point is None):
        print("[ERROR] --shapefile か --point のどちらか一方を指定してください。",
        file=sys.stderr)
        sys.exit(1)

    input_dirs = sorted(set(args.inputs))
    os.makedirs(args.output, exist_ok=True)

    for in_dir in input_dirs:
        dname = os.path.basename(os.path.normpath(in_dir)) or "out"

```

```

out_dir = os.path.join(args.output, dname)
os.makedirs(out_dir, exist_ok=True)
print(f"¥ninput: {in_dir}¥n -> output: {out_dir}", flush=True)

rain_paths = find_rain_files(in_dir)
if not rain_paths:
    print(" (skip) rain.nc が見つかりません。", flush=True)
    continue

# 初回: cnst.nc から lon/lat を取得
try:
    cnst_path = find_cnst_for(rain_paths[0], max_up=2)
except Exception as e:
    print(f"[ERROR] {e}", file=sys.stderr)
    continue
lon2d, lat2d = load_lonlat_from_cnst(cnst_path)  # (J,I)
J, I = lon2d.shape

# ----- 対象セル抽出 -----
if args.shapefile:
    print(f" shapefile: {args.shapefile}")
    shp_paths, shapely_polys = read_polygon_paths(args.shapefile)

    # bbox で広めに候補抽出
    lon_min, lon_max, lat_min, lat_max = bbox_from_paths(shp_paths)
    lon_pad = max(median_grid_step(lon2d, axis=1), 1e-3)
    lat_pad = max(median_grid_step(lat2d, axis=0), 1e-3)
    lon_min -= lon_pad; lon_max += lon_pad
    lat_min -= lat_pad; lat_max += lat_pad

    bbox_mask = ((lon2d >= lon_min) & (lon2d <= lon_max) &
                 (lat2d >= lat_min) & (lat2d <= lat_max))
    jj, ii = np.where(bbox_mask)
    sel_candidates = np.column_stack([jj, ii])  # (m,2) (j,i)

if sel_candidates.size == 0:
    print(" (skip) bbox 内候補セルが見つかりませんでした。", flush=True)
    continue

# coverage 計算
if args.cell_def == "dual":
    cov_all = coverage_fraction_dual(lon2d, lat2d, sel_candidates, shp_paths,
FCT=499)
else:
    if not _HAVE_GPD:
        raise RuntimeError("cell-def voronoi/primal requires"

```

```

shapely/geopandas installed.")
    cell_polys = []
    if args.cell_def == "primal":
        for (j, i) in sel_candidates:
            i1 = min(i+1, I-1); j1 = min(j+1, J-1)
            ring = [(lon2d[j, i], lat2d[j, i]),
                     (lon2d[j, i1], lat2d[j, i1]),
                     (lon2d[j1, i1], lat2d[j1, i1]),
                     (lon2d[j1, i], lat2d[j1, i]),
                     (lon2d[j, i], lat2d[j, i])]
            cell_polys.append(Polygon(ring))
    elif args.cell_def == "voronoi":
        cell_polys = build_voronoi_cells(lon2d, lat2d, sel_candidates)
        # 被覆率（厳密面積）
        shp_u = unary_union(shapely_polys) if shapely_polys else None
        cov_all = np.zeros(len(sel_candidates), dtype=float)
        for idx, poly in enumerate(cell_polys):
            try:
                p = make_valid(poly) if (_HAVE_MAKE_VALID and not
poly.is_valid) else (poly if poly.is_valid else poly.buffer(0))
                if shp_u is not None:
                    inter = p.intersection(shp_u)
                    a = inter.area; A = p.area
                    cov_all[idx] = (a / A) if A > 0 else 0.0
                else:
                    # フオールバック: 中心点内外
                    j,i = sel_candidates[idx]
                    inside = False
                    for pp in shp_paths:
                        if pp.contains_point((lon2d[j,i], lat2d[j,i])):
                            inside = True; break
                    cov_all[idx] = 1.0 if inside else 0.0
            except Exception:
                cov_all[idx] = 0.0

mask = cov_all > float(args.min_coverage)
sel_jis = sel_candidates[mask]
cov = cov_all[mask]

if sel_jis.size == 0:
    print(" (skip) coverage>min_coverage のセルがありませんでした。",
flush=True)
    continue
else:
    # 単一点: 最寄りセル 1 点、被覆率は 1.0
    lat_str, lon_str = args.point.split(",")

```

```

sel_jis = nearest_cell_by_point(lon2d, lat2d, float(lat_str), float(lon_str))
cov = np.ones(len(sel_jis), dtype=float)

# coverage CSV (中心点の (i,j) をセル名として出力)
i_out = sel_jis[:, 1] + (1 if args.index_base == 1 else 0)
j_out = sel_jis[:, 0] + (1 if args.index_base == 1 else 0)
cov_df = pd.DataFrame({"i": i_out, "j": j_out, "fraction_area": cov})
cov_df.to_csv(os.path.join(out_dir, "target_coverage.csv"), index=False)
print(f" [INFO] target_coverage rows: {len(cov_df)} (index_base={args.index_base}, min_coverage={args.min_coverage}, cell_def={args.cell_def})")
print(f" [INFO] mean fraction: {cov_df['fraction_area'].mean():.6f}, min/max: {cov_df['fraction_area'].min():.6f}/{cov_df['fraction_area'].max():.6f}")

# セル範囲のポリゴン出力 (任意)
if args.export_cells_shp:
    export_selected_cell_polys(args.export_cells_shp, args.cell_def, lon2d, lat2d,
                                sel_jis, cov, index_base=args.index_base)
    print(f" [INFO] exported selected cell polygons -> {args.export_cells_shp}")

```

```
# break
```

```

# ----- rain 読み出し & 出力 -----
jmin, imin = sel_jis.min(axis=0)
jmax, imax = sel_jis.max(axis=0)
sel_local = sel_jis.copy(); sel_local[:, 0] -= jmin; sel_local[:, 1] -= imin
col_names = [f"i_{j}" for (j, i) in sel_local]

all_dates = []
all_rows = []
w_sum_all = []
w_tot_all = []

for k, f in enumerate(rain_paths, 1):
    print(f" [{k:3d}/{len(rain_paths)}] {f}", flush=True)
    ds = Dataset(f, "r")

    # 時刻
    tvar = ds.variables["time"]
    cal = getattr(tvar, "calendar", "standard")

```

```

t = num2date(tvar[:,], units=tvar.units, calendar=cal)
if args.shift_hours:
    td = timedelta(hours=args.shift_hours)
    t = [tt - td for tt in t]
t_str = [pd.Timestamp(str(tt)).strftime("%Y-%m-%d %H:%M:%S") for tt in
t]

# rain 切り出し
r = ds.variables["rain"]
if r.ndim == 4:
    sub = r[:, 0, jmin:jmax + 1, imin:imax + 1] # member=0
elif r.ndim == 3:
    sub = r[:, jmin:jmax + 1, imin:imax + 1]
else:
    ds.close()
    raise ValueError(f"想定外の rain 次元: {r.shape}")
year = int(t_str[0][0:4])
start_term = dt.datetime(year, 9, 1, 0)
end_term = dt.datetime(year + 1, 8, 31, 23)

for ti, ts in enumerate(t_str):
    ts_dt = pd.to_datetime(ts)
    if (start_term <= ts_dt <= end_term):
        vals = [float(sub[ti, jloc, iloc]) for (jloc, iloc) in sel_local]
        all_rows.append(vals)

        wsum = float(np.dot(vals, cov))
        wtot = float(cov.sum())
        w_sum_all.append(wsum)
        w_tot_all.append(wtot)

    all_dates.append(ts)

ds.close()

if not all_rows:
    print(" (skip) 有効な時刻がありませんでした。", flush=True)
    continue

df_all = pd.DataFrame(all_rows, columns=col_names)
df_all.insert(0, "date-hour", all_dates)
df_all.set_index("date-hour", inplace=True)
df_all.index = pd.to_datetime(df_all.index)

arr_mean = np.array(w_sum_all) / np.array(w_tot_all)
df_mean = pd.DataFrame({"rainfall": np.round(arr_mean, 6)}, index=df_all.index)

```

```

for yr in sorted(df_all.index.year.unique()):
    start = dt.datetime(yr, 9, 1, 0)
    end = dt.datetime(yr + 1, 8, 31, 23)
    d_all = df_all[(df_all.index >= start) & (df_all.index <= end)]
    d_mean = df_mean[(df_mean.index >= start) & (df_mean.index <= end)]
    if not d_all.empty:
        d_all.to_csv(os.path.join(out_dir, f"rain_allgrids_{yr:04d}.csv"))
    if not d_mean.empty:
        d_mean.to_csv(os.path.join(out_dir, f"rain_arealmean_{yr:04d}.csv"))

print(" -> done.", flush=True)

if __name__ == "__main__":
    try:
        t0 = time.time()
        main()
        t1 = time.time()
        print(f"Ynall done. elapsed time: {(t1 - t0)/60:.1f} min", flush=True)
    except KeyboardInterrupt:
        print("Interrupted.", file=sys.stderr)
        sys.exit(130)

```