Guide to downloading and extracting CHIRPS rainfall estimates using Python

Author: Dr Ross Maidment (ross.i.maidment@gmail.com)

Summary
This guide provides guidance on how to execute a set of Python scripts which can be used to download and extract area-average daily CHIRPS rainfall estimates for a set of user-specified locations.

Overview of CHIRPS rainfall estimates

Climate Hazards Group InfraRed Precipitation with Station data (CHIRPS) is a 35+ year quasi-global rainfall data set. Spanning 50°S-50°N (and all longitudes) and ranging from 1981 to near-present, CHIRPS incorporates our in-house climatology, CHPclim, 0.05° resolution satellite imagery, and in-situ station data to create gridded rainfall time series for trend analysis and seasonal drought monitoring. More details about CHIRPS can be found here: https://www.chc.ucsb.edu/data/chirps.

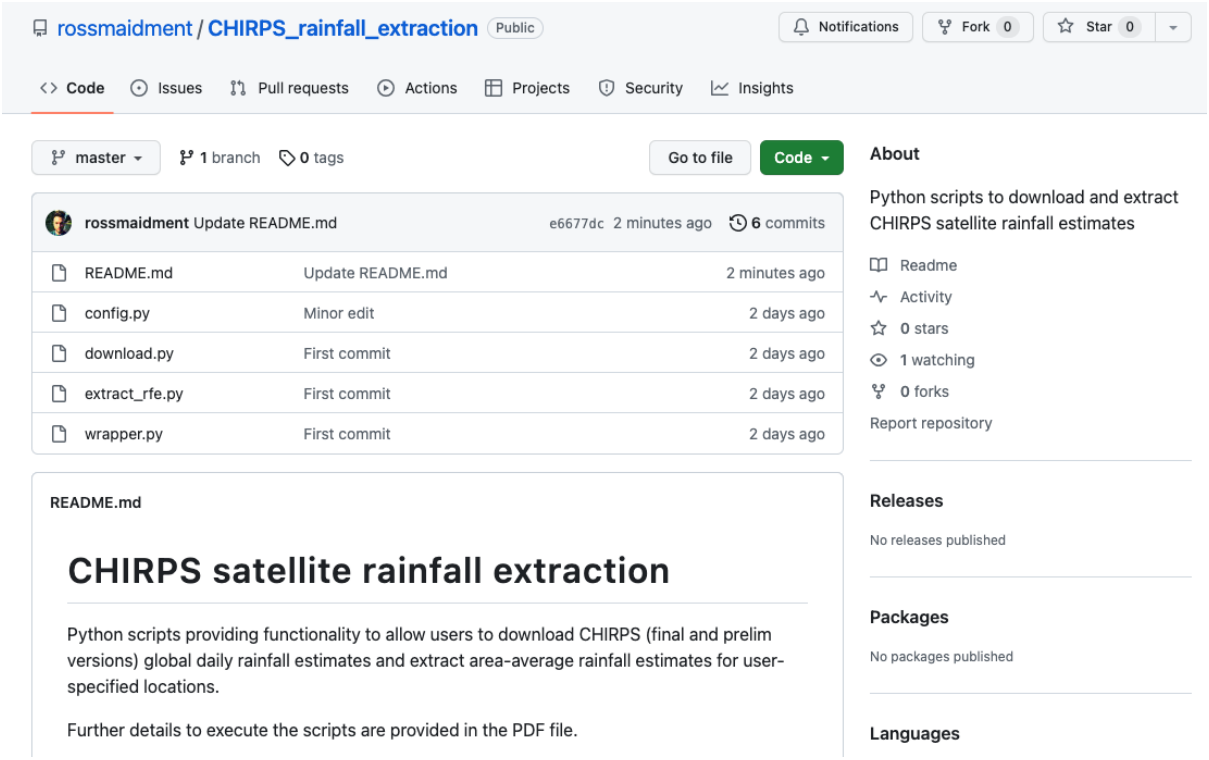
CHIRPS provides rainfall estimates at the daily, pentadal, dekadal and monthly timesteps. Two versions are available for CHIRPS:

- **CHIRPS-final**: Available with a latency of up to 6 weeks and includes both publicly available near-real time rain gauge information and additional proprietary rain gauge data that have become available after the near-real time public rain gauge has been included. Global daily files can be accessed here: https://data.chc.ucsb.edu/products/CHIRPS-2.0/global_daily/netcdf/p05/
- **CHIRPS-prelim**: Available within 7 days but does not include additional rain gauge data that the final version includes which may make it less reliable in places. One additional caveat is that the data is only available since 2015. In practice for most regions, CHIRPS-final and CHIRPS-prelim are very similar. Global prelim daily files can be accessed here: https://data.chc.ucsb.edu/products/CHIRPS-2.0/prelim/global_daily/netcdf/p05/.

Overview of CHIRPS download and extraction Python code

The Python code for processing CHIRPS data can be accessed from Ross Maidment's Github repository (Github is an online repository for managing and sharing code): https://github.com/rossmaidment/CHIRPS_rainfall_extraction

This repository is publicly available. A screenshot is given below.



The code base is comprised of following four Python scripts:

- `config.py` - The configuration file is where users specify details relevant to their computer and data requirements. This is the only script that needs to be edited by users. The following details need to be provided:
  - `startdate/enddate`: start and end dates of the period of interest (this will be between 1st January 1981 to the near present)
  - `tag`: a name to identify that particular job
  - `latlon_file`: name of the CSV file which contains the coordinates of the locations that rainfall estimates need to be extracted for (more on this below)
  - `product`: CHIRPS product to be processed – this will be '*final*' or '*prelim*'
  - `workingdir`: local path where the code will be stored
- `download.py` – This provides the functionality to download global daily 0.05° files from the CHIRPS server to your local machine. Users should not need to edit this script or execute it.
- `extract_rfe.py` – This provides the functionality to subset CHIRPS data for the coordinates of the locations specified in the `latlon_file` for the period specified by the start and end dates. Users should not need to edit this script or execute it.
- `wrapper.py` – This is the only script that users need to execute. It performs the download tasks followed by the extraction tasks. If successful, executing this script will produce a CSV file containing a time-series of the area-average CHIRPS rainfall estimates for the locations specified in the `latlon_file`.

Preparation steps

1. Create a directory on your local machine where you will store the code, CHIRPS rainfall files and extracted outputs. You can call this directory whatever you prefer, for example “chirps_rainfall_extraction” – make sure not to include any spaces in the name.

2. Within this directory, create the following subdirectories:
    - code
    - coordinates
    - data

You should end up with the following directory structure:

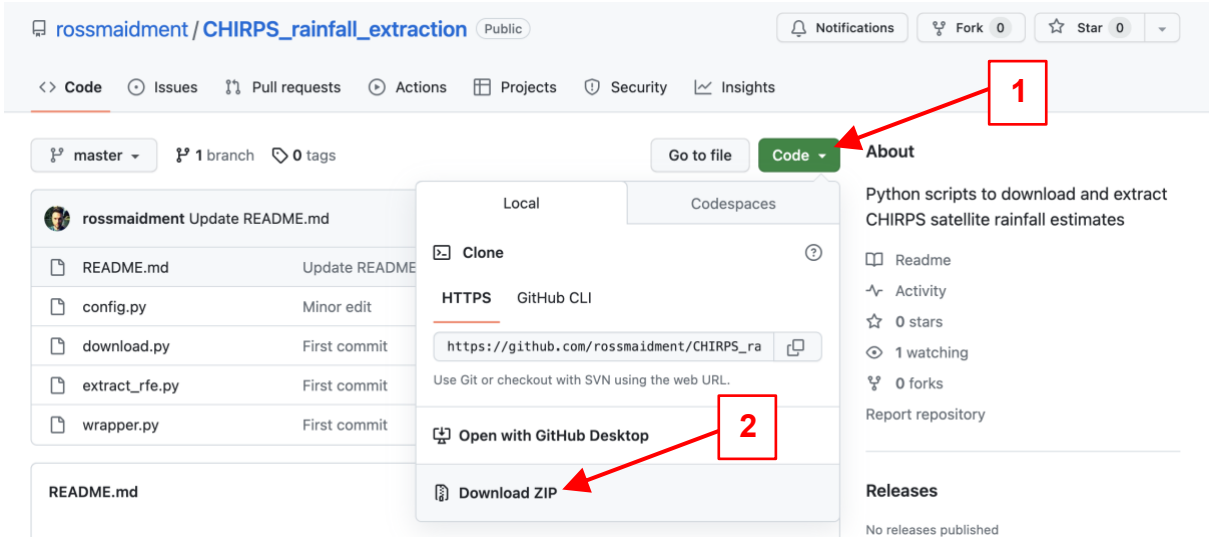
```
chirps_rainfall_extraction
├── code
├── coordinates
├── data
```

3. Create a CSV file containing the coordinates of the locations that CHIRPS rainfall data will be extracted for. The name of this file is what is specified by the `latlon_file` variable in the `config.py` file. It should be structured to contain columns that give the coordinates of the bounding box (i.e. N, S, W and E) over which the rainfall estimates will be averaged, as indicated by the following example (you can specify as many columns before N, S, W and E as is required):

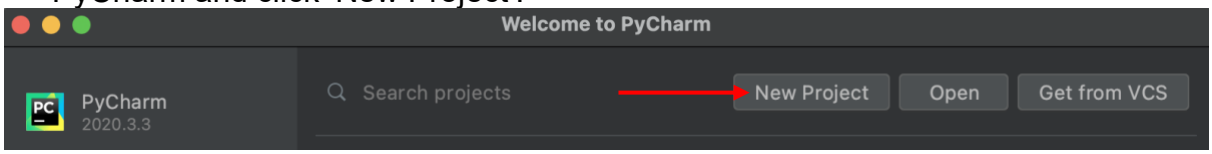
Province	District	Lat	Lon	N	S	W	E
A	A1	14	20	14.5	13.5	19.5	20.5
B	B1	15	20	15.5	14.5	19.5	20.5
C	C1	16	20	16.5	15.5	19.5	20.5

   Put this file in the `coordinates` directory.

4. Next, you need to download the Python code from Github and put it into the `code` directory. To do this, first you need to click on the green ‘Code’ button on the Github repository and click on ‘Download ZIP’. Once downloaded, unzip the zip file and copy all four Python files into the `code` directory.

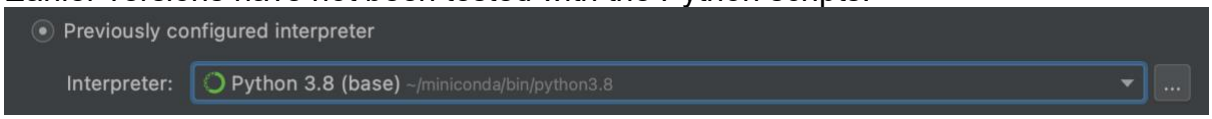


5. Next, you need to create a PyCharm project for the Python code. For this, open PyCharm and click 'New Project'.



Next, under the location field, use the full path where you created the `chirps_rainfall_extraction` directory.

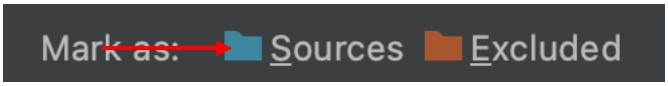
Now select the Python interpreter. In the example below, Python 3.8 has been used. Earlier versions have not been tested with the Python scripts.



Next, click 'Create' at the bottom. This will initialise the PyCharm project. You can view and navigate the three subdirectories (`code`, `coordinates` and `data`) using the navigation panel on the top left region of the PyCharm page.

Now, users need to make sure that PyCharm can recognise the code directory as source code. To do this, follow these steps:

- Open PyCharm preferences
- Click on 'Project' and then 'Project Structure'
- Click on the 'code' directory and then assign this as a source by clicking on the blue 'Sources' folder icon:



- Now click 'Apply'.

6. Installing Python libraries. The following Python libraries are required and need to be installed:

```
os, numpy, datetime, xarray, pandas, wget
```

Running the Python code

Once the above steps have been completed, running the Python code is straightforward and can be achieved by following these steps:

1. Open and edit the config.py file to your required specifications and save.
2. Now execute the wrappy.py file. This performs two tasks:
    a. Downloading of the CHIRPS daily global files. As CHIRPS produces yearly files, they are quite large (about 1.1 Gb per year, so users need to make sure they have enough disk space if handling multiple years). The files are given in netCDF format and stored under `/data/netcdf/`.
    b. Extracting of daily rainfall estimates for the specified locations which are saved as a single CSV file. Locations are given by row and dates given as columns. The output CSV file is stored under `/data/output/` with the following file naming convention:

        `chirps2.0-<product>_<startdate>-<enddate>_<tag>.csv`

Additional details

- If users have already downloaded CHIRPS files (netCDF format), they will not be downloaded again unless they are manually deleted. This is to avoid unnecessary downloading.
- Since CHIRPS produces yearly files, new rainfall estimates are appended to the existing current yearly file. As such, if users want to extract the most recent rainfall estimates, they need to delete the latest yearly netCDF file so that the Python code can redownload the file.
- NetCDF files can be viewed using a viewer called Panoply (https://www.giss.nasa.gov/tools/panoply/)
- If changes are made to the code on Github, users can download the latest Python code and repeat the steps outlined above.