WCSSP Component C-1 Code Reference Guide version 2.0

DOWNLOAD DATA CODE SECTION

download_data module:
Description: download or update the desired data depending on the time

class Download_Data
Initial Attributes:
- date_start (string)
  - Text of the latest date (YYYYMMDD) of successful download data
- hour_start (string)
  - Text of the latest hour (HH) of successful download data
- data_type (string)
  - type of the data for download
  - Choices:
    - UM (Unified Model Global)
    - GSM_0.25 (Global Spectral Model 0.25 deg)
    - WRF (Weather Research and Forecasting Model)
    - GSMaPNRT (GSMaP NRT version)
    - GSMaPGauge (GSMaP Gauge version)

Python module/package used:
- datetime and timedelta from datetime package
- os
- ftplib
- numpy
- time
- pandas
- sys
- HTTPError and URLError from urlib.error
- timeout from socker package
- urlib.request
- requests
- Other self made modules (see Supporting Python Modules):
  - search_missing file
  - check_download
  - functions

Functions of class Download_Data:
- **download_Data()**
  - Description: downloads the latest available data
  - Function used:
    - latestDateAvailable (see functions module under Supporting Python Modules)
    - getFilesAvailable (see functions module, under Supporting Python Modules)
    - download (under class Download_UK)

    - get_base_folder (under class Download_Data)
    - get_file_path (under class Download_Data)
    - get_file_directory (under class Download Data)
    - search_missing (see search_missing_file module, under Supporting Python Modules)
    - check_download (see check_download module, under Supporting Python Modules)

update_Data(missing_flag, files)

- Description: download the missing data relative to the latest date (date_start and hour_start)
- Parameters:
    - missing_flag (boolean)
        - if True, missing_files variable is equal to files
        - if False, calculate the missing_files variable
    - files (list) – list of file that wish to be download
- Function used:
    - latestDateAvailable (see functions module, under Supporting Python Modules)
    - download (under class Download_UK)
    - get_base_folder (under class Download_Data)
    - get_file_path (under class Download_Data)
    - get_file_directory (under class Download Data)
    - search_missing (see search_missing_file module, under Supporting Python Modules)
    - check_download (see check_download module, under Supporting Python Modules)

download(ftp, username, password, filepath, filename, data_type)

- Description: retrieve the file on the ftp server
- Parameters:
    - ftp (string) – ftp server address
    - username (string) – username of the ftp account
    - password (string) – password of the ftp account
    - filepath (string) – directory address to where to download the file
    - filename (string) – name of the file to be downloaded
    - data_type (string) – type of data for download
        - Choices:
            - UM (Unified Model Global)
            - GSM_0.25 (Global Spectral Model 0.25 deg)
            - WRF (Weather Research and Forecasting Model)
            - GSMaPNRT (GSMaP NRT version)
            - GSMaPGauge (GSMaP Gauge version)

DATA EXTRACTION CODE SECTION

gsmap module:
Description: extract rain rate from the zip file and convert to accumulated rainfall

class GSMap

Python module/package used:

- datetime
- pandas
- numpy
- zipfile
- os

Functions of class Check_Download:

extract_precipitation (start_date, end_date, data_type, interval)

- Description: extracts rain rate from the zip file and convert to accumulated rainfall based on the interval
- Parameters:
  - start_date (datetime) – start date of the extraction
  - end_date (datetime) – end date of the extraction
  - data_type (string) – type of GSMaP data to be extracted
    - Choices:
      - NRT (Global Rainfall Map in Near Real Time)
      - Gauge_NRT (Gauge-calibrated Rainfall Product in Near Real Time)
      - MVK (Global Satellite Mapping of Precipitation Microwave-IR Combined Product)
      - Gauge (Gauge-calibrated Rainfall Product)
- Returns:
  - dprecip_arr (array) - array of accumulated precipitation
  - latitude (array) – latitude of grid (1-D)
  - longitude (array) – longitude of grid (1-D)

model module:
Description: extract rainfall data from a given date range

class Model
Initial Attributes:

model_type (string)

- Text of the desired model
  - Choices are:
    - UM (Unified Global Model)
    - GSM (0.25 degrees GSM)
    - WRF_12km (Weather Research and Forecasting Model 12 km)
    - WRF_3km (Weather Research and Forecasting Model 3 km)

Python module/package used:

- pygrib
- numpy
- pandas
- datetime and timedelta from datetime package

Functions of class Model:

extract(start_date, end_date, variable, filename, interval)

- Description: extracts the variable from a given date range and compute its accumulated data with a given interval
- Function used:
  - select_files (under model module)
- Parameters:
  - start_date (datetime) – start date of the extraction
  - end_date (datetime) – end date of the extraction
  - variable (string) – meteorological variable for extraction (only rainfall is available)
  - filename (string) - name of the folder which inside the desired data (must be in YYYYMMDDHH format)
  - interval (int) – time range for its accumulation (in terms of hours)
- Returns:
  - met_var_arr (array) - array of extracted variable
  - lat_model (array) – latitude of the grid
  - lon_model (array) – longitude of the grid
  - dates (array) – array of datetime representing the extracted variable
  - date_range_arr (array) – array of string which represents the computation of the accumulation

extract_points (start_date, end_date, data_type, interval, coordinate)

- Not Yet Available

select_files (filename, start_date, end_date, interval)

- Description: get the appropriate files in a given date range and interval
- Parameters:
  - filename (string) - name of the folder which inside the desired data (must be in YYYYMMDDHH format)
  - start_date (datetime) – start date of the extraction
  - end_date (datetime) – end date of the extraction
  - interval (int) – time range for its accumulation (in terms of hours)
- Returns:
  - files (array) - array of selected data files
  - date_arr (array) – array of datetime representing the extracted variable
  - date_range_arr (array) – array of string which represents the computation of the accumulation
  - var_index (array) – array of index in each selected file
  - file_index (array) – array of index within the files array

POST-PROCESSING CODE SECTION

graph module:
Description: create basemap or graph in a given input data specifically in Philippines area

class Graph(output, model_type, lat = None, Lon = None, basemap = True)
Initial Attributes:

model_type (string)

- Text of the desired model
    - Choices are:
        - UM (Unified Global Model)
        - GSM (0.25 degrees GSM)
        - WRF_12km (Weather Research and Forecasting Model 12 km)
        - WRF_3km (Weather Research and Forecasting Model 3 km)

main_title (string)

- main title of the figure (default: “Insert Main Title”)

main_title_font_size (int)

- font size of the main title (default: 12)

dpi (int)

- resolution of the image (default: 300)

main_title_y (int)

- y-position of the main title (default: 1)

if basemap is True:

lon (array)

- array of longitude based on the input data

lat (array)

- array of latitude based on the input data

resolution (string)

- resolution of the basemap (default: ‘c’)

linewidth (float)

- line width of the basemap (default: 1)

sub_title_font_size (int)

- sub title font size of the figure (default: 10)

if basemap is False:

x_name (array)

- array of names of the x-axis (default: [“Insert X Name]) x-label (string)

- label of the plot from the x-axis (default: “Insert X Label”)

y-label (string)

- label of the plot from the y-axis (default: “Insert Y Labe”l)

x_legend_name (string)

- array of legend names (default: [“Insert Y Label”])

alpha (float)

- transparency of the plots (default: 1)

Python module/package used:

- pandas
- numpy
- matplotlib
- Basemap from mpl_toolkits.basemap
- ListedColormap from matplotlib.colors
- Patch and PathPatch from matplotlib.patches
- os
- Other self made modules (see Supporting Python Modules):
  - categorical

Functions of class Graph:

create_basemap(file_path, output, map_boundary, sub_title = None, color_bar = ”default”, graph_type = ”single”, plot_type = ”land_water”, shapefile = True)

- Description: creates a basemap based on the input value within the boundary (specifically for model data and GSMaP data)
- Function used:
  - create_figure (under graph module)
  - get_shape_bound (under graph module)
  - create_cmap (under graph module)
- Parameters:
  - file_path (string) – full directory (including the filename) of the desired storage
  - output (array) – input data with dimensions of attributes lon and lat
  - map_boundary (list) – list of the upper and lower boundaries of latitude and longitude (format: [lower latitude, upper latitude, lower longitude, upper longitude])
  - subt_title (array of string) – title of each sub plots (only applicable in multiple graph_type)
  - color_bar (string) – color scheme used in the legend (Choices: bias, rmse and default)
  - graph_type (string) – type of graph in the figure (singular or multiple)
  - plot_type (string) – plotting type of the basemap, includes water surface (land_water) or land only (land_only) for the graph
  - shapefile (boolean) – use shapefile as the basemap layer

create_plot(file_path, output, cat_type = None, graph_type = “single”, plot_style = “bar_graph”, plot_type = “land_water”)

- Description: creates a graph (bar graph or line graph)
- Function used:
  - category_min (under categorical module)
  - category_max (under categorical module)
  - create_cmap (under graph module)
- Parameters:
  - file_path (string) – full directory (including the filename) of the desired storage
  - info (string) – additional information on the title
  - initTime (string) – initial time of the model
  - output (array) – array of input values
  - cat_type (string) – type of the categorical verification (see categorical module, under Supporting Python Modules for the options)
  - graph_type (string) – type of graph in the figure (singular or multiple)
  - plot_style (string) – style of the graph (bar or line graph)
  - plot_type (string) – plotting type of the basemap, includes water surface (land_water) or land only (land_only) for the graph

create_figure (graph_type)

- Description: create figure depending on its graph type. If graph type is multiple, automatically divides the available output
- Parameters:
  - graph_type (string) - type of graph in the figure (singular or multiple)
- Returns:
  - fig (matplotlib.pyplot) - new figure
  - ax (matplotlib.pyplot) – axes of the new figure (fig)
  - loop_index (int) – number of available output or sub plots

get_shape_bound (map_boundary)

- Description: calculates the shape of latitude and longitude based on the boundary
- Parameters:
  - map_boundary (list) – list of the upper and lower boundaries of latitude and longitude (format: [lower latitude, upper latitude, lower longitude, upper longitude])
- Returns:
  - shape (list) - shape of the bounded latitude and longitude [lon bound, lat bound]

create_cmap (color_bar)

- Description: create a color map of the figure
- Parameters:
  - color_bar (string) – color scheme used in the legend (bias, rmse and default)
- Returns:
  - cmap (matplotlib.colors.Colormap) - new color map
  - norm (matplotlib.colors.Normalize) – normalized data into [0.0, 1.0] interval
  - boundaries (list or array) – list or array of the ticks in the legend

SUPPORTING PYTHON CODES

functions module:
Description: get the latest date available or filename of the certain data

class Functions
Initial Attributes:

DATA (string)

- Text of the desired data
  - Choices are:
    - UM (Unified Global Model)
    - GSM_0.25 (0.25 degrees GSM)
    - WRF (Weather Research and Forecasting Model)
    - GSMaPNRT (GSMaP NRT version)
    - GSMaPGauge (GSMaP Gauge version)

Python module/package used:

- datetime and timedelta from datetime package

Functions of class Functions:

latestDateAvailable(date)

- Description: get the latest date available on a certain data depending on its input date
- Parameters:
  - date (datetime) – input date with a type of datetime
- Returns:
  - latest_date_available (dateime) - latest date available of the certain data including its initial time

getFilesAvailable(initTime, date)

- Description: get the list of the available data filename
- Parameters:
  - initTime (int) – hour property of the datetime (applicable only on 'GSM_0.25' and 'GSMaPNRT' data)
  - date (datetime) – desired date to be converted to filename
- Returns:
  - default_dir (list of string) - list of the available filename on a particular date which converted from the date parameter

search_missing_file module:
Description: determine the missing file of the data from the given directories

class Search_Missing
Initial Attributes:

DATA (string)

- Text of the desired data
- Choices are:
  - UM (Unified Global Model)
  - GSM_0.25 (0.25 degrees GSM)

    - WRF (Weather Research and Forecasting Model)
    - GSMaPNRT (GSMaP NRT version)
    - GSMaPGauge (GSMaP Gauge version)

df_download_log (pandas)
- information of the download log file (csv format)

missing_data (dataframe)
- list of the missing data from the specific data

Python module/package used:
- datetime and timedelta from datetime package
- pandas
- numpy
- os
- Other self made modules (see Supporting Python Modules):
    - functions

Functions of class Search_Missing:

createMissingFolder(directory, folder_name, latest_date_available)
- Description: creates missing folder based on the initial date from the folder_name parameter and the latest_date_available parameter
- Parameters:
    - directory (list) – list of the directory to be checked
    - folder_name (list) – name of the folder (must be in YYYYMMDDHH format)
    - latest_date_available – recent date available for the specific data
- Returns:
    - directory (list) - updated directory list (includes the missing folder/s if applicable)
    - folder_name (list) – updated folder_name list (includes the missing folder name/s if applicable)

filesCount(hour)
- Description: get the total number of files available in a specific data
- Parameters:
    - hour (int) – hour property of the datetime (applicable only on ‘GSM_0.25’)
    - date (datetime) – desired date to be converted to filename
- Returns: count (int) - count of the total files available

missingFileFormat(file)
- Description: convert the filename of the data same as the format of the missing_data dataframe
- Parameters:
    - file (string) – filename of the data
- Returns:
    - file_format (string) – converted file parameter

search_missing(directory, main_dir, latest_date_available)

- Description: determine the missing file from the given directory
- Parameters:
  - directory (list) – list of the full directory containing the specific data
  - main_dir (list) – folder name of the directory containing the specific data
  - latest_date_available (datetime) - recent date available for the specific data
- Returns:
  - missing_files (array) – array of the missing files

check_download module:
Description: check if the files are downloaded successfully

class Check_Download
Initial Attributes:
DATA (string)

- Text of the desired data
  - Choices are:
    - UM (Unified Global Model)
    - GSM_0.25 (0.25 degrees GSM)
    - WRF (Weather Research and Forecasting Model)
    - GSMaPNRT (GSMaP NRT version)
    - GSMaPGauge (GSMaP Gauge version)

BASE_DIR (string)

- directory of the stored data

Python module/package used:

- pandas
- numpy
- os

Functions of class Check_Download:

getFileInformation(file)

- Description: extracts necessary information in a filename
- Parameters:
  - file (string) – filename of the data
- Returns:
  - directory (string) - name of the folder
  - filename (string) – name of the file
  - missing_file (string) – missing file format
  - index (int) – index of the log file based on the data

check_download(files, date, date_start)

- Description: check if the files are downloaded successfully and updates the log files (download_log, missing_data and pending_Data)
- Parameters:

    - files (list) – list of the filename that needs to be checked
    - date (datetime) – date of the recent file from the file list
    - date_start (datetime) – start date from the download log file
- Returns:
    - directory (string) - name of the folder
    - filename (string) – name of the file
    - missing_file (string) – missing file format
    - index (int) – index of the log file based on the data

categorical module:
Description: categorical verification based on the two inputs

class Categorical

Python module/package used:
- numpy

Functions of class Categorical:

categorical(obs, model, threshold, cat_type)

- Description: filter the empty array and computes the categorical verification
- Function used: categorical_type (under categorical module)
- Parameters:
    - obs (array) – array of the first input values (normally the observed data). Must be the same shape as to the model parameter
    - model (array) – array of the second input values (normally the model data). Must be the same shape as to the obs parameter
    - threshold (float) – threshold value for the input value to be occurred flag
    - cat_type (string) – categorical verification type
        - Options:
            - prop_cor (Proportion Correct)
            - FA_Ratio (False alarm Ratio)
            - UER (Undetected Error Rate)
            - HR (Hit Rate)
            - FA_Rate (False Alarm Rate)
            - BS (Bias score)
            - TS (Threat Score)
            - ETS (Equitable Threat Score)
- Returns:
    - output (array) – array of the calculated selected categorical verification

category_type(obs, model, threshold, cat_type)

- Description: computes categorical verification based on the two inputs and threshold
- Parameters:
    - obs (array) – array of the first input values (normally the observed data). Must be the same shape as to the model parameter

    - model (array) – array of the second input values (normally the model data). Must be the same shape as to the obs parameter
    - threshold (float) – threshold value for the input value to be occurred flag
    - cat_type (string) – categorical verification type
- Returns:
    - output (array) – array of the calculated selected categorical verification

category_title(category_type)
- Description: name of the category type
- Parameters:
    - category type (string) – categorical verification type
- Returns:
    - output (string) – name of the category type

cateogry_max(category_type)
- Description: maximum value can attain by the categorical verification
- Parameters:
    - category type (string) – categorical verification type
- Returns:
    - output (float) – maximum value

cateogry_min(category_type)
- Description: minimum value can attain by the categorical verification
- Parameters:
    - category type (string) – categorical verification type
- Returns:
    - output (float) – minimum value

idw module:
Description: inverse distance weighting interpolation

class IDW

Python module/package used:
- numpy
- spatial from scipy package

Functions of class IDW:
haversine_np(lon1, lat1, lon2, lat2)
- Description: determine distance between two points using haversine formula
- Parameters:
    - lon1 (array) – array of longitude of the new grid points
    - lat1 (array) – array of latitude of the new grid points
    - lon2 (array) – array of longitude of the old grid points
    - lat2 (array) – array of latitude of the old grid points
- Returns:

  - km (array) – array of distances between two coordinates (in kilometers)

interpolate(x_new, y_new, x, y, z, power)

- Description: interpolate z array to a new grid points (x_new, y_new) using IDW
- Parameters:
  - x_new (array) – array of longitude of the new grid points
  - y_new (array) – array of latitude of the new grid points
  - x (array) – array of longitude of the old grid points
  - y (array) – array of latitude of the old grid points
  - z (array) – output values based in (x,y) grid points
  - power (float) – power component of the distance between points from the haversine_np function
- Returns:
  - idw_arr (array) – IDW interpolated output value