Date
April 14, 2015

Version
0.1

Author
Ronald van Haren

Summer in the city

UHI validation

Netherlands eScience Center

Science Park 140
1098 XG Amsterdam

+31 (0)20 888 41 97

info@eScienceCenter.com

www.eScienceCenter.com

by SURF & NWO

CHAPTER

1

Download data from *Weather Underground* and KNMI

1.1 Extract Dutch stations

The script *wunderground_dump_stationid.py* parses the Weather Underground website to find all stations in the Netherlands. For each station found, the following variables are extracted: *station id*, *neighborhood*, *city*, *station type*, and *location* (latitude, longitude, height). The zipcode of the station location is extracted using the api of Google Maps. The extracted data is saved to a csv file.

The script takes one (optional) argument, which is the name of the output csv file. The default name of the output csv file is *wunderground_stations.csv* in the current working directory.

Program 1.1 wunderground_dump_stationid.py

```
usage: wunderground_dump_stationid.py [-h] [-o OUTPUT]

Extract all Wunderground stations in the Netherlands and write the station
names and locations to a csv file

optional arguments:
  -h, --help            show this help message and exit
  -o OUTPUT, --output OUTPUT
                        CSV output file [default: wunderground_stations.csv]
```

1.2 Download Wunderground data

Downloading data from Weather Underground is performed using the script *wunderground_getdata.py*. A station id can be supplied to the script, using the (-s, --stationid) switch, to download data for a single station. Alternatively, the csv file that results from the *wunderground_dump_stationid.py* script can be supplied to the script using the (-c, --csvfile) switch to download data for all stations in the csv file.

Other relevant arguments to the script are:

- (-b, --startyear) and (-e, --endyear): Defines the begin and end of the period that data is downloaded for.
- (-o, --outputdir): Data output directory. Data files will be saved to ${outputdir}/${STATIONID}

- (-k, --keep): Use the argument switch to keep already downloaded data. If the argument switch is not used existing data is overwritten.

Program 1.2 wunderground_getdata.py

```
usage: wunderground_getdata.py [-h] [-o OUTPUTDIR] [-b STARTYEAR] [-e ENDYEAR]
                               [-s STATIONID] [-c CSVFILE] [-k]
                               [-l {debug,info,warning,critical,error}]

Combine csv files weather underground in one output file

optional arguments:
  -h, --help            show this help message and exit
  -o OUTPUTDIR, --outputdir OUTPUTDIR
                        Data output directory
  -b STARTYEAR, --startyear STARTYEAR
                        Start year
  -e ENDYEAR, --endyear ENDYEAR
                        End year
  -s STATIONID, --stationid STATIONID
                        Station id
  -c CSVFILE, --csvfile CSVFILE
                        CSV data file
  -k, --keep            Keep downloaded files
  -l {debug,info,warning,critical,error}, --log {debug,info,warning,critical,error}
                        Log level
```

After the data is downloaded, the directory ${outputdir}/${STATIONID} contains a separate csv file for each day. In order to combine this data in a single netCDF file, the script *combine_wunderground_data.py* is used. The script takes two arguments:

- (-i, --inputdir): Input directory containing daily csv files that need to be combined
- (-o, --outputdir): Output directory of the resulting netCDF file

Multiple input directories can be processed at once using a simple shell command. For example, to process all subdirectories within the current working directory and save the resulting netCDF files to a subdirectory called ncfiles:

```
for directory in *; do ./wunderground_getdata.py -i ${directory} -o ncfiles; done
```

1.3 KNMI reference data

KNMI station data is used as reference data for rural temperatures. The script *knmi_getdata.py* downloads the reference data from the KNMI website.

- (-o, --outputdir): Output directory where the data should be saved.
- (-s, --stationid): Station id of the KNMI reference station to download data from. If the argument is not used, the KNMI website is parsed to get all available stationids and data for all stations is downloaded.

Program 1.3 combine_wunderground_data.py

```
  usage: combine_wunderground_data.py [-h] -i INPUTDIR [-o OUTPUTDIR]

Combine csv files weather underground in one output file

optional arguments:
  -h, --help            show this help message and exit
  -i INPUTDIR, --inputdir INPUTDIR
                        Data input directory containing txt files
  -o OUTPUTDIR, --outputdir OUTPUTDIR
                        Data output directory
```

- (-c, --csvfile): Name of output csv file that contains information about the KNMI stations and their location.
- (-k, --keep): Use the argument switch to keep already downloaded data. If the argument switch is not used existing data is overwritten.

Program 1.4 knmi_getdata.py

```
  usage: knmi_getdata.py [-h] [-o OUTPUTDIR] [-s STATIONID] -c CSVFILE [-k]
                       [-l {debug,info,warning,critical,error}]

Get data KNMI reference stations

optional arguments:
  -h, --help            show this help message and exit
  -o OUTPUTDIR, --outputdir OUTPUTDIR
                        Data output directory
  -s STATIONID, --stationid STATIONID
                        Station id
  -c CSVFILE, --csvfile CSVFILE
                        CSV data file
  -k, --keep            Keep downloaded files
  -l {debug,info,warning,critical,error}, --log {debug,info,warning,critical,error}
                        Log level
```

CHAPTER

2 Urban heat island

An urban heat island (UHI) is an metropolitan area that is warmer than its surrounding rural areas due to human activities. The temperature difference usually is larger at night than during the day, and is most apparent when the winds are weak.

2.1 Determining the UHI effect

Time filtering Wunderground data

The data from Weather Underground is measured at random time steps. In order to compare it with hourly KNMI reference data, the data should be processed to create a dataset on the same temporal density. This is done by the functionality in the script *time_filter_wunderground_data.py*. This script is just a helper script and does not need to be called directly, the filtering functionality is imported in the script *UHI_reference.py*.

The script supports two ways to time filter the Weather underground data to a fixed time step, namely:

- **interpolate**: depending on the measurements available, the value at the time step is determined according to the following rules
  - if a measurement coincides directly with the time step, the value at the time step is set equal to the measurement
  - if within the timewindow set measurements are available both before and after the time step, the value at the time step is set to the value obtained from interpolating the last measurement before the time step and the first measurement after the time step
  - if only measurements within the time window set are available before the time step, the value is set to the last measurement before the time step
  - if only measurements within the time window set are available after the time step, the value is set to the first measurement after the time step
  - if no measurements are available within the time window the value is set to None
- **average**: all measurements within the selected time window are averaged

UHI calculation script

Calculation of the UHI effect for all Wunderground stations is performed using the script *UHI_reference.py*. The script has one required argument, the numbers of the months over which the UHI needs to be calculated (-m, --months). Optional arguments are:

- (-w, --wundfile): Csv file with station from Weather Underground. Output from the script *wunderground_dump_stationid.py*. If the argument is not specified, the default filename *wunderground_stations.csv* is used.
- (-k, --knmifile): Csv file with KNMI reference stations. Output from the script *knmi_getdata.py*. If the argument is not specified, the default filename *knmi_reference_data.csv* is used.
- (-i, --interpolate): Specify if time filtering of Weather Underground data should use interpolation instead of time averaging.
- (-s, --stationtype): Specify if only station that use an instrument that contains the (converted to lowercase) string specified.

Program 2.1 UHI_reference.py

```
usage: UHI_reference.py [-h] [-w WUNDFILE] [-k KNMIFILE] [-i] [-s STATIONTYPE]
                        -m MONTHS [MONTHS ...]

Time filter Wunderground netCDF data

optional arguments:
  -h, --help            show this help message and exit
  -w WUNDFILE, --wundfile WUNDFILE
                        Wunderground csv file [default:
                        wunderground_stations.csv]
  -k KNMIFILE, --knmifile KNMIFILE
                        KNMI csv file [default: knmi_reference_data.csv]
  -i, --interpolate     Distance weighted interpolation of KNMI reference data
                        instead of nearest reference station
  -s STATIONTYPE, --stationtype STATIONTYPE
                        Require a certain instrument for the Wunderground
                        station
  -m MONTHS [MONTHS ...], --months MONTHS [MONTHS ...]
                        month numbers (1-12) separated by space used to
                        calculate UHI
```

2.2 Methods

describe methods

CHAPTER

3 Results

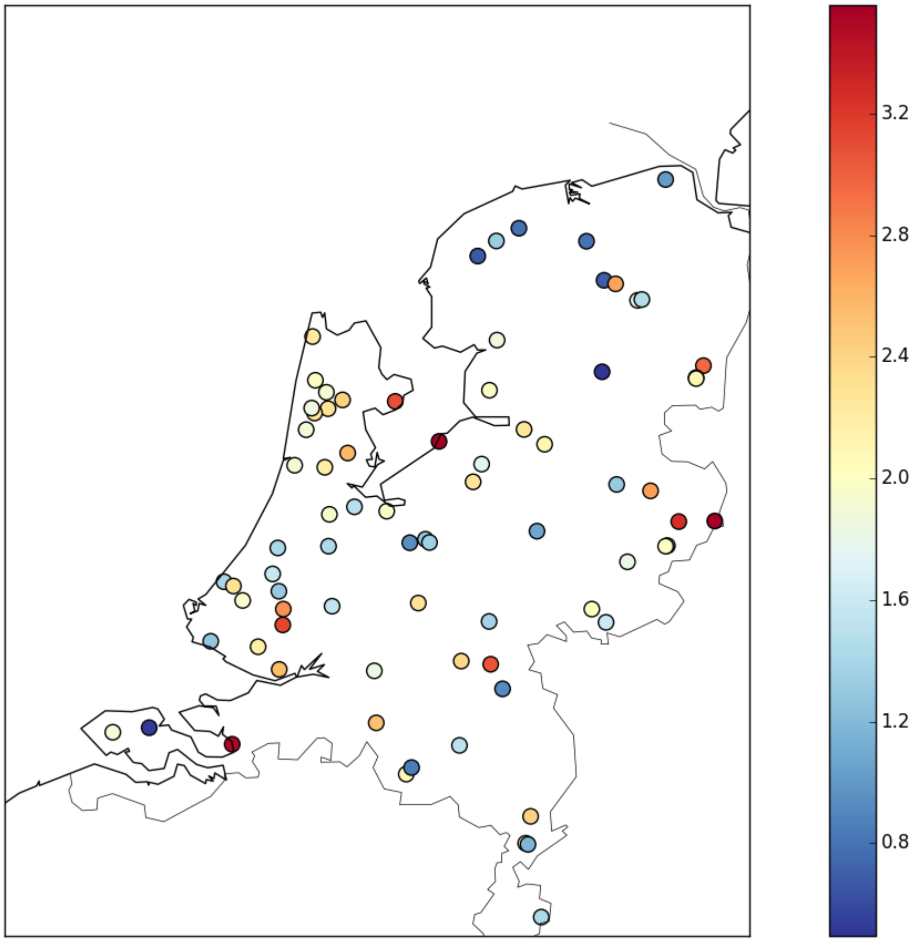

Figure 3.1: UHI

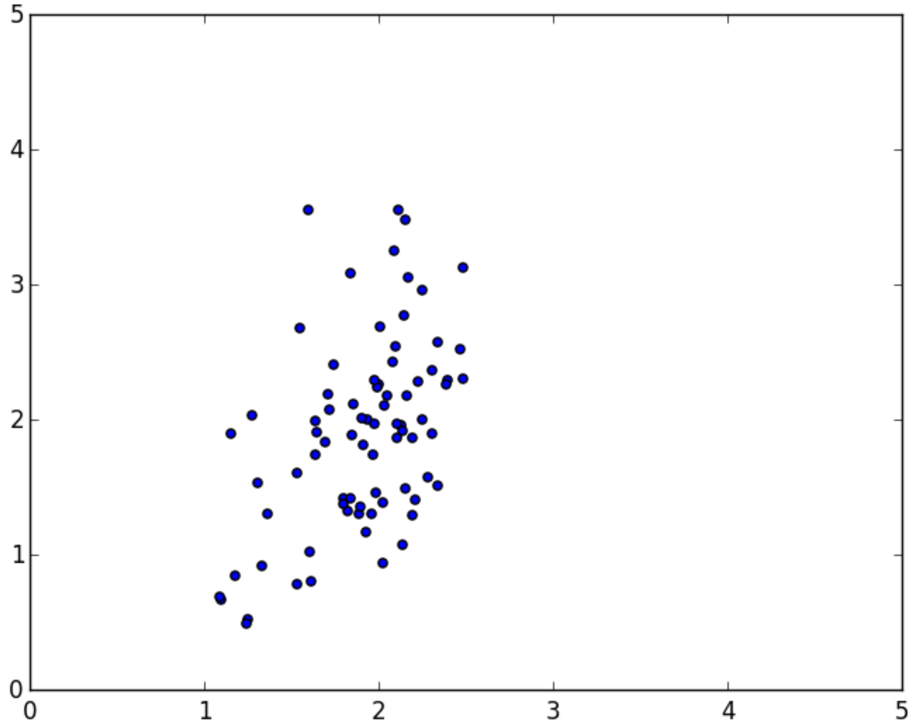

Figure 3.2: Reconstruction