

---

# **ng-npn-data-quality Documentation**

***Release 0.1***

**Brandon Taylor**

**Apr 27, 2020**



**CONTENTS:**

<b>1</b>	<b>Indices and tables</b>	<b>3</b>
	<b>Python Module Index</b>	<b>5</b>
	<b>Index</b>	<b>7</b>



Filename: profiler\_metr.py

Purpose: A collection of meteorological calculations used to manipulate profiler data into a form usable for analysis.

Author: Brandon Taylor

Date: 20200310

Last Modified: 20200310

`profiler_metr.hypsometric` (*specific\_humidities, temperatures, pressures, elev*)

Calculates the thickness of the layer using the hypsometric equation. Returns the resulting geometric heights in a numpy array.

`profiler_metr.interpolate_uv` (*interpolation\_tuple*)

Interpolates the two observation sets to a regular grid.

`profiler_metr.pressure_to_height` (*pressure, elev*)

Converts pressure to height using the U.S. Standard Atmosphere, subtracting station elevation to yield height above Mean Sea Level (MSL).

`profiler_metr.wind_components` (*speed, wdir\_deg*)

Computes the vector components of wind from speed and direction. Wind components are return as U (east-west) and V (north-south).

`profiler_metr.wind_direction` (*u\_vec, v\_vec*)

Computes the wind direction from u and v components.

`profiler_metr.wind_direction_difference` (*wdir\_hrrr, wdir\_npn*)

Rotates the wind difference calculation, so that they lie between -180 and 180.

`profiler_metr.wind_speed` (*u\_vec, v\_vec*)

Computes the wind speed from u and v components.

Filename: compare\_npn\_to\_model.py

Purpose: Retrieves NPN data from either ROCSTAR or WEATHER.GOV and compares to BUFR soundings from either the HRRR or NAM 3-KM model.

Author: Brandon Taylor

Date: 20190411

Last Modified: 20200420

`compare_npn_to_model.available` (*date, icao, hours, overall=False, npn\_data=False*)

Computes height availability

`compare_npn_to_model.calc_min_max` (*nnp\_heights, hrrr\_heights*)

Calculates the height bounds across a time-series

`compare_npn_to_model.compare_profiles` ()

endpoint for compare method

`compare_npn_to_model.connect_db` ()

Create a connection to the SQLite database. Arguments: @return {obj} - sqlite3 connection object.

`compare_npn_to_model.data_availability` ()

endpoint for availability method

`compare_npn_to_model.data_outages` ()

endpoint for data outage tracking

`compare_npn_to_model.data_outages_metadata` ()

endpoint for data outage tracking metadata including icao and dates

`compare_npn_to_model.difference` (*user\_params*, *hourly='t'*, *overall=False*, *npn\_data=False*,  
*raob=False*, *qc=False*)  
Computes difference between NPN data and HRRR data by interpolating to regular height levels, starting at 100 meters, going to 10 km, at 100 meter intervals.

`compare_npn_to_model.difference_profiles` ()  
endpoint for difference method

`compare_npn_to_model.extra_B3` (*date*, *icao*, *hours*)  
reads Build 3 data from 2017

`compare_npn_to_model.extra_heights` (*date*, *icao*, *hours*, *hourly*)  
Tests the extra heights algorithm

`compare_npn_to_model.generate_expected_dates` (*start\_date\_str*, *end\_date\_str*, *hourly*)  
Generates expected dates for data outage tracking purposes.

`compare_npn_to_model.hourly` (*npn\_data*, *hours*)  
Returns the percentage availability of hourly data encountered, from given expected number of hours.

`compare_npn_to_model.index_html` ()  
Sends main page static HTML

`compare_npn_to_model.model` ()  
endpoint for model check method

`compare_npn_to_model.model_check` (*date*, *icao*, *hours*, *variable*)  
Computes difference between NPN data and HRRR data by interpolating to regular height levels, starting at 100 meters, going to 10 km, at 100 meter intervals.

`compare_npn_to_model.overview` ()  
endpoint for dashboard overview

`compare_npn_to_model.profile_html` ()  
Sends profile comparison static HTML

`compare_npn_to_model.read_b3_buf` (*fname*)  
Reads NPN Build 3 style BUFR files. Extracts and converts height and wind speed/direction.

`compare_npn_to_model.read_ncep_buf` (*fname*, *convert\_uv*, *offset*)  
Reads NCEP BUFR type files, which include the local table as the first message. Extracts and converts height and wind speed/direction.

`compare_npn_to_model.read_npn_csv` (*fname*)  
Reads NGNPN CSV files. Extracts and converts height and wind speed/direction.

`compare_npn_to_model.retrieve_hrrr_winds` (*date*, *hour*, *user\_params\_dict*, *offset=0*, *convert\_uv=False*)  
Reads in hrrr bufr sounding from local archive

`compare_npn_to_model.retrieve_npn_winds` (*user\_params*, *hourly='t'*, *hours=24*)  
Reads in npn data from ROCSTAR. Tries the primary ROCSTAR server first with timeout, then tries the backup ROCSTAR server.

`compare_npn_to_model.retrieve_raob` (*date*, *hour\_str*, *icao*)  
Reads in npn data from ROCSTAR. Tries the primary ROCSTAR server first with timeout, then tries the backup ROCSTAR.

`compare_npn_to_model.sqlite_date_parse` (*date*)  
Returns a data in hyphenated format for SQLite purposes. Example: input 20200101, output 2020-01-01

`compare_npn_to_model.track_html` ()  
Sends data outages static HTML

## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`





## PYTHON MODULE INDEX

### c

`compare_npn_to_model`, 1

### p

`profiler_metr`, 1



## A

available() (in module *compare\_npn\_to\_model*), 1

## C

calc\_min\_max() (in module *compare\_npn\_to\_model*), 1

compare\_npn\_to\_model (module), 1

compare\_profiles() (in module *compare\_npn\_to\_model*), 1

connect\_db() (in module *compare\_npn\_to\_model*), 1

## D

data\_availability() (in module *compare\_npn\_to\_model*), 1

data\_outages() (in module *compare\_npn\_to\_model*), 1

data\_outages\_metadata() (in module *compare\_npn\_to\_model*), 1

difference() (in module *compare\_npn\_to\_model*), 1

difference\_profiles() (in module *compare\_npn\_to\_model*), 2

## E

extra\_B3() (in module *compare\_npn\_to\_model*), 2

extra\_heights() (in module *compare\_npn\_to\_model*), 2

## G

generate\_expected\_dates() (in module *compare\_npn\_to\_model*), 2

## H

hourly() (in module *compare\_npn\_to\_model*), 2

hypsonetric() (in module *profiler\_metr*), 1

## I

index\_html() (in module *compare\_npn\_to\_model*), 2

interpolate\_uv() (in module *profiler\_metr*), 1

## M

model() (in module *compare\_npn\_to\_model*), 2

model\_check() (in module *compare\_npn\_to\_model*), 2

## O

overview() (in module *compare\_npn\_to\_model*), 2

## P

pressure\_to\_height() (in module *profiler\_metr*), 1

profile\_html() (in module *compare\_npn\_to\_model*), 2

profiler\_metr (module), 1

## R

read\_b3\_buf() (in module *compare\_npn\_to\_model*), 2

read\_ncep\_buf() (in module *compare\_npn\_to\_model*), 2

read\_npn\_csv() (in module *compare\_npn\_to\_model*), 2

retrieve\_hrrr\_winds() (in module *compare\_npn\_to\_model*), 2

retrieve\_npn\_winds() (in module *compare\_npn\_to\_model*), 2

retrieve\_raob() (in module *compare\_npn\_to\_model*), 2

## S

sqlite\_date\_parse() (in module *compare\_npn\_to\_model*), 2

## T

track\_html() (in module *compare\_npn\_to\_model*), 2

## W

wind\_components() (in module *profiler\_metr*), 1

wind\_direction() (in module *profiler\_metr*), 1

wind\_direction\_difference() (in module *profiler\_metr*), 1

wind\_speed() (in module *profiler\_metr*), 1