weather.py documentation

Hotaka Shiokawa 12/31/2017
Send any questions to hshiokawa@cfa.harvard.edu or hotaka84@gmail.com

Overview

This script reads-in historical data files of opacity (tau) at zenith at various observational sites and plot statistical

- tau and tau/sin(elev) for given percentile values as a function of time of a day
- probability of tau < given tau value and tau/sin(elev) < given tau/sin(elev) as a function of time of a day

both for a given time range of a year, where elev=elevation of SgrA* at each site.

Here, I just show how to use 2 commands, plot_probability() and plot_percentile(). For actual modification, addition of sites, and debugging, please refer to the comments in the code. It is commented in very details. This script-data package includes all the required data to run the example commands in document, including historical data files for ALMA, LMT, and MaunaKea for real data (in the directory 'real') and data of Global Forecast System (GFS) processed using scripts written by Rodrigo Cordova and Scott Paine at ALMA, LMT, SMA, SMT, SPT, and PV for April 2007-2017 (in the directory 'GFS'). Any arbitrary sites' data can be obtained using the Rodrigo's script in the directory 'GFS' together with the original data files stored separately in hard drives.

Required Libraries

pandas, numpy, scipy, matplotlib
Python2 can produce more intended visuals, but python3 works fine, too.

Commands

The script is intended to be used in interactive mode of python.
First, import the weather script
$python
>> import weather as w

————————

"plot_probability(m1,d1,m2,d2,dt,val,GFS=False)"
This command plots probability of [tau < given tau value] and [tau/sin(elev) < given tau/sin(elev) value] as a function of time of a day for a given time range in a year.
Input parameters:

- m1,d1=month and date of starting date to take a statistics
- m2,d2=month and date of ending date to take a statistics
- dt = time bin size in hr
- val = arbitrary tau value to calculate the probabilities
- GFS = wether using real data or GFS data. It is set to False (use real data) by default.
- In addition, elevation cutoff value, elev_cutoff, is defined near the top of the script as a global variable, set to 15 degrees by default. Data points with elev < elev_cutoff will not be plotted in the tau/sin(elev) plot.

Examples
>> w.plot_probability(4,11,4,20,0.5,0.3,GFS=True)
(See figure_1.png for the result)
This plots probabilities that tau/sin(elev) < 0.3 (upper panel) and tau < 0.3 (lower panel) for all the GFS sites' data available in this package, for 4/11 - 4/20. It also plots SgrA* elevation for each site in the upper panel, and 'net probability' that all the available sites' probability at the moment are multiplied. Note the time bin size is set to 0.5 (30min), but since the GFS data is taken every 2.5 hours, the resultant plot uses dt=2.5 hrs.

>> w.plot_probability(4,11,4,20,0.5,0.3)
(see figure_2.png for the result)
This command takes the real data at ALMA, LMT, and MaunaKea instead of GFS. dt=0.5 is now effective because real data's time resolution can be as high as 10min.

————————

"plot_percentile(m1,d1,m2,d2,dt,site,GFS=False)"
This command plots tau for a given percentile values as a function of time of a day for a given time range in a year.
Input parameters:
- m1,d1=month and date of starting date to take a statistics
- m2,d2=month and date of ending date to take a statistics
- dt = time bin size in hr
- site = site of your choice, 'ALMA','LMT','Maunakea'(or 'SMA') for GFS=False, additional choices of 'SMT','SPT','PV' for GFS=True
- GFS = wether using real data or GFS data. It is set to False (use real data) by default.
- In addition, list of percentile values that you want to choose can be set in "per_list" defined right above 'def plot_percentile()' in the script.
- In addition, elevation cutoff value, elev_cutoff, is defined near the top of the script as a global variable, set to 15 degrees by default. Data points with elev < elev_cutoff will not be plotted in the tau/sin(elev) plot.

Example
>> w.plot_percentile(4,11,4,20,0.5,'LMT')
(see figure_3.png for the result)
This command plots tau/sin(elev) of percentile=25%, 50%, and 75% (upper panel) as well as that of tau (lower panel) for 4/11 - 4/20 using real LMT data, with time bin size=0.5hrs.

>> w.plot_percentile_all(4,11,4,20,0.5,GFS=True)
This command calls a set of plot_percentile() for all the GFS data available sites. Definitions of the input parameters are the same as that of plot_percentile().

Further Works

Adding sites:
After reading-in data in whatever form, the goal is to construct a Pandas Dataframe in the format:
time = pandas.DataFrame(data={'yr':yr, 'mon':mon, 'day':day, 'utc':utc, 'tau':tau})
The time must be sorted in ascending order.
For GFS data, Rodrigo's script can directly produce data files in a format that weather.py can read-in using get_data_gfs().

In addition, a file that store SgrA* elevation information needs to be created.

Calculation of Elevation:
Currently, the script reads-in '2018 SgrA* elevation files' that store SgrA* elevation information for each site produced by analysis/elev/elev.py which uses ehtim package. The file format is: yr, mon, day, MJD, elev at 1st time bin of a day, elev at 2nd time bin,…..
The script also requires a file that stores time of each time bin. This feature is apparently very inconvenient, inefficient, and not elegant, and I hope someone can put a function that can directly calculate source elevation for a given location and time,

Possible Bugs/Problems

- get_data_LMT(): The time zone in the read-in file is likely to be UTC, but could be local. The file is already processed by Dimitrios, but he doesn't quite remember the time zone either. We assume it is UTC for now, but need to be confirmed or read-in the original raw data files.

- get_data_Maunakea(): MaunaKea's final result obtained by this script doesn't look correct; tau doesn't vary much over a day while it is supposed to. There could be a bug in this function or misunderstanding of the read-in data file format/unit.. This definitely need to be checked, probably starting from plotting some example days' tau vs. utc. If the problem exists in get_binned_data(), that's fatal for many other functions.

Real Data Sources

ALMA 1995-2004:
Downloaded from 225GHz Transparency data column (text) at
http://legacy.nrao.edu/alma/site/Chajnantor/data.c.html

ALMA 2007-2017:
Downloaded from
http://archive.eso.org/wdb/wdb/eso/meteo_apex/form
Note pwd->tau relation at the site is calculated by Scott Paine (tau = 0.0375*d['pwv'] + 0.0115)

MaunaKea:
Data is provided by Simon Radford. The 'mk' directory in the link below
https://drive.google.com/drive/folders/1W0ugANyElV7gskl6jf1a4_sanTg_JFzX

LMT:
Data file processed by Dimitrios available in EHT (SAO?) Dropbox
EHT Weather/Sites Weather/LMT/lmtdata.dat
Note that time in this file is assumed to be (and likely to be) UTC, but need to confirm.