Precipitable-Water Model Analysis Tool Documentation

Spencer Riley

Mar 14, 2022

CONTENTS:

1 Getting Started **3**
- 1.1 Introduction 3
- 1.2 Installation and Deployment Tutorial 4

2 Working with data and PMAT **7**
- 2.1 Input Data Formatting 7
- 2.2 Output Data Formatting 9

3 Changelog **11**
- 3.1 PMAT Cirrus 11
- 3.2 PMAT Altocumulus 14

4 Tables **15**
- 4.1 Input Data File 15

5 Templates **19**
- 5.1 Input Data File 19

6 PMAT Reference **21**
- 6.1 pmat_analysis.r 21
- 6.2 pmat_processing.r 24

6.3 pmat_products.r 27
6.4 pmat_run.r 31
6.5 pmat_utility.r 32

The Precipitable-Water Model Analysis Tool is an open-source suite for analyzing the relationship between atmospheric brightness temperature and precipitable water.

Warning: This documentation is under active development.

CHAPTER

ONE

GETTING STARTED

1.1 Introduction

The Precipitable-water Model Analysis Tool (*PMAT*) is a computational utility that is used to analyze the data collected from this project to understand the relationship between the zenith sky temperature and precipitable water in the atmosphere. *PMAT* has three different modules that work together to present data.

The first is the Deployment Module. This module acts as the user interface for the software suite, whether it be locally or through cloud services.

The second is the Pre-processing Module, this module imports data from University of Utah's MesoWest and the University of Wyoming UpperAir Databases.

The third module is the main program to run the analysis, the DAnalysis Module. Here the all of the data is presented and the regression analysis between precipitable water and zenith sky temperature is conducted.

1.2 Installation and Deployment Tutorial

We also require two data files. One that contains the raw data collected by the temperature sensors, that also includes date and time information (`cool_data.csv`). The second should contain sensor definitions with additional parameters for the preprocessing and analysis phases (`_pmat.yml`). A template and detailed breakdown of the configuration file is provided in Chapter 2.1, followed by a detailed breakdown on the data file format.

1.2.1 Github

This version of the Deployment module is, for the most part, automated and recommended. Follow the steps in this section to successfully deploy PMAT through GitHub with GitHub Actions.

1. Create a GitHub repository from the template repository[1].
2. Edit the README.md page based on your location and username
3. Update all files that are contained in the `data/` directory, and utilize the documentation on data formatting that is provided
4. Upon finializing updates on `cool_data.csv`, the workflow will run automatically and the visual and data products will be generated

[1] https://template.pmat.app

1.2.2 Amazon Web Service (AWS)

For Amazon Web Services, PMAT can be configured through the EC2 virtual machines. Once they have been configured, connect to the virutal machine. Once connected, enter the following commands

```
sudo yum update -y
```

```
sudo amazon-linux-extras install docker
```

```
sudo docker pull ghcr.io/physicsgoddess1972/
↪pmat:latest
```

From here, data files can be added and utilizing the `local_deploy.sh` script, PMAT can be executed.

1.2.3 Google Cloud Console (GCloud)

1.2.4 Local

We fully support Ubuntu and Debian systems. We do have minimal Windows support through the usage of Windows Subsystem for Linux (WSL) and Virutalization.

1.2.5 Development

CHAPTER

TWO

WORKING WITH DATA AND PMAT

This chapter will discuss the data components associated with PMAT. The first section will detail the formatting guidelines of the input files, followed by another discussion regarding the various output files generated by the software suite.

2.1 Input Data Formatting

The two input files discussed in this section include a YAML configuration file and a Comma Separated Value data file.

2.1.1 Configuration Input

The role of the configuration file is to store a series of parameters that includes sensor information, analytic parameters, logging options, and the site identifiers for the University of Wyoming's Upper-Air database and the University of Utah's MesoWest database.

A template of the configuration file presented in Template XX shows the structure of the data fields. The filename of this file

must be `_pmat.yml`.

2.1.2 Raw Data File

The raw data file is processed, through pattern identification, allowing for a

- Dataset Example 1[2]
- Dataset Example 2[3]
- Dataset Example 3[4]

It should be noted that the columns do not have to be in any set order, with one small caveat, the model pulls the data from columns with headers containing specific words or phrases. The caveat is with regards to Ground and Sky temperature readings. The temperature measurements must go in consecutive order by sensor as determined by `_pmat.yml`.

For example, if the order of the sensors in `_pmat.yml` is 1610 TE, FLIR i3, and then AMES 1. Then the order of the ground and sky temperature measurements in the dataset should be: 1610 TE, FLIR i3, and then AMES 1. (As seen in Dataset 2).

[2] https://github.com/physicsgoddess1972/Precipitable-Water-Model/blob/master/data/example/example1.csv

[3] https://github.com/physicsgoddess1972/Precipitable-Water-Model/blob/master/data/example/example2.csv

[4] https://github.com/physicsgoddess1972/Precipitable-Water-Model/blob/master/data/example/example1.csv

2.2 Output Data Formatting

There are a variety of data files generated by the software suite. The data files are stored as CSV files, with each row presenting data for a single day.

2.2.1 General data files

The primary data file [`master_data.csv`] generated is the full dataset that

- Date
- time
- sky condition (clear sky/overcast)
- ground temperature
- sky temperature
- Radiosonde PWV
- Relative Humidity
- Dewpoint
- User comments

2.2.2 Machine learning

The machine learning data file includes five columns:

- Date
- Average brightness temperature

- Average PWV
- Relative Humidity
- Sky Condition

This data set supports the classification of data by the sky condition label.

2.2.3 Analytic results

The main analytical results are stored as YAML configuration files. The results of each step in the iterative analysis process are saved to a file with the name `_output.yml`. An example of this file is presented below. [sample of _output.yml] [table of the fields in _output.yml]

The averaged results of the steps are also stored in a YAML file. [sample of _results.yml] [table of the fields in _results.yml]

CHAPTER

THREE

CHANGELOG

3.1 PMAT Cirrus

Version 2.0

Date 6 Mar 2021

Tagline New and Improved PMAT

3.1.1 Overall

- [Updated] Compatible with R 4.0

3.1.2 Data-input

- [Added] Now includes relative humidity imports.
- [Added] Now pulls data from MesoWest..
- [Added] New guidelines for sensors that are not active (See Documentation Page for further info.)

3.1.3 Setup-script

- [Updated] Now installs R 4.0
- [Added] Additional argument to configure database imports (run *bash setup.sh -h* for more information)

3.1.4 Plots

- [Fixed] Fixed issues with bar charts where if there were more than three sensors, not all bar charts would be added for the remaining sensors.
- [Added] Added more time series plots and more composite plots.
- [Updated] Changed the x-axis labeling system to have tick marks at the 1st of the month.
- [Updated] Redesigned the main analytical plot, confidence interval is now a shaded region, and the plot is now monochromatic.
- [Updated] Pac-Man residual was removed from this plot set.
- [Updated] Pac-man residual now resides in a new plot set (run *Rscript model.r –pacman*)
- [Added] Mean TPW and Mean temperature comparison can now be visualized in a Pac-Man plot.

3.1.5 Web-applications

- [Added] Two web-apps are active. One is a Data Dashboard, which allows for the viewing of time series data as a scatter plot or a heat map, and analytical comparisons between data that has been collected.
- [Added] The Data Dashboard also allows for custom time series data to be uploaded.
- [Added] The Machine Learning dashboard now allows for custom data to be uploaded.

3.1.6 Documentation

- [Fixed] Fixed multiple CSS issues.
- [Updated] Altered Pac-Man residual plot documentation to refer to the package documentation
- [Updated] Updated procedure to include the new command-line arguments
- [Added] Included buttons on the dashboard's "Project Updates" card to include Pac-Man plots and Poster plots that are generated from data we have collected.
- [Updated] We also scored a *.tech* domain for the page.

3.1.7 Automation

- [Misc] This is a work in progress

3.2 PMAT Altocumulus

Version 1.0

Date 10 Nov 2019

Tagline Initial Deployment of The Precipitable Water Model

3.2.1 Overall

- [Added] Flexible data input
- [Added] Easy Hands-off setup.
- [Added] Command-line arguments to access the different plots available
- [Added] Time Series plots for zenith sky temperature and precipitable water
- [Added] Analytical plots showing the correlation between zenith sky temperature and precipitable water
- [Added] Poster ready plots for presentations
- [Added] A data set including the average temperature and precipitable water
- [Added] The Pac-Man Residual
- [Updated] Documentation Page.

CHAPTER

FOUR

TABLES

4.1 Input Data File

4.1.1 `cool_data.csv`

Data Type	Header	Format
Date	Date	YYYY-MM-DD
Time	Time	HH:MM
String	Condition	clear sky / overcast
Sky Temperature	Sensor Name (Sky)	float
Ground Temperature	Sensor Name (Ground)	float

Data Type	Header	Format
datetime	Date	YYYY-MM-DD
datetime	Time	HH:MM
string	Condition	clear sky / overcast
float	Sensor Name (Sky)	float
float	Sensor Name (Ground)	float

4.1.2 _pmat.yml

Field Name	Description	Example
name	The name of the sensor. If there are multiple of the same sensor use the notation _*N* with N being the index of the sensor.	Sensor 09 Sensor 10_1 Sensor 10_2

5.0 | +———+———+———+ | color | A hexadecimal color code that will be used to identify the sensor on visualizations. | FF0000 0000FF | +———+———+———+ | ratio | The distance to spot ratio that is reported by the manufacturer. (OPTIONAL) | 12 to 1 21 to 1 | +———+———+———+ | emissivity | The emissivity of the sensor as reported by the manufacturer. (OPTIONAL) | 0.95 | +———+———+———+ | poster | A boolean that will decide whether the sensor

will be shown in the poster-specific plots. | true false | +———+———+———+ | active | A boolean that will decide whether the sensor will be used in the analysis | true false | +———+———+———+

CHAPTER

FIVE

TEMPLATES

5.1 Input Data File

5.1.1 _pmat.yml

```
- instruments:
  - sensor:
      name:
      error:
      color:
      ratio:
      range:
      emissivity:
      poster:
      active:
- analysis:
  - train_fraction:
  - rel_difference:
  - iteration:
      step:
```

(continues on next page)

(continued from previous page)

```
      seed:
- logging:
  - verbose:
- import:
  - mesowest:
    - id:
  - wyoming:
    - id:
      weight:
```

CHAPTER

SIX

PMAT REFERENCE

6.1 pmat_analysis.r

module Precipitable Water Model Analysis Tool: Analysis

synopsis This module contains analysis functions

`exp.regression`(*t=NULL*, *mean.out*)

Detail Function includes all of the stuff to generate the exponential regression model with intervals

Parameters

- **t** (*double*) – training fraction
- **mean.out** (*list*) – the output of mean.filter

Returns returns the data series and model statistics

Return type list

`lin.regression`(*x*, *y*)

Detail Linear regression function

Parameters

- **x** (*double*) – the domain of the dataset
- **y** (*double*) – the range of the dataset

Returns returns the data series and model statistics

Return type list

`data.partition`(*x*, *y*, *tr.sz=0.7*)

Detail splits the data into a training/testing set

Parameters

- **x** (*double*) – domain of the data
- **y** (*double*) – range of the data
- **tr.sz** (*double*) – fraction of the data in the testing set

Returns a list containing the training and testing sets

Return type list

`iterative.analysis`(*obool*, *mean.out*)

Detail computes regression statistics and outputs to a yaml file

Parameters

- **obool** (*logical*) – determine whether to generate new _output.yml
- **mean.out** (*list*) – output of mean.filter

Returns iterative stats and _output.yml

Return type list

lsvm(*x*, *y*, *l*, *tr.sz=0.7*, *seed=sample(1:2^15, 1)*)

Detail Generates a Linear Support Vector Machine and draws the decision hyperplane and support vectors

Parameters

- **x** (*double*) – domain of dataset
- **y** (*double*) – range of dataset
- **l** (*double*) – labels of the dataset
- **tr.sz** (*double*) – fraction of data to be used for model training
- **seed** (*integer*) – the random seed

Returns list of data, labels, and the coefficients

Return type list

6.2 pmat_processing.r

module Precipitable Water Model Analysis Tool: Pre-processing

synopsis functions for preprocessing

`colscheme`(*range*)

Detail a function that generates an array of colors based on the number of elements

Parameters **`range`** (*`list`*) – a list of data series

Returns a list of colors

Return type list

`mean.filter`(*nan.out*, *n*)

Detail filters the data based on the comparison of the daily std and the average std of the dataset

Parameters

- **`nan.out`** (*`list`*) – the output of nan.filter
- **`n`** (*`integer`*) – threshold

Returns an array of indicies for PWV values to be analyzed

Return type list

`dna.filter`(*fover*)

Detail removes data labels as Do Not Analyze

Parameters **fover** (*list*) – overcast.filter results

Returns overcast.filter results with DNA points removed

Return type list

nan.**filter**(*stuff*)

Detail removes nan values from a set of lists

Parameters **stuff** (*list*) – list of arrays

Returns returns list with filtered data and the indicies with nans

Return type list

inf.**counter**(*bool*, *snsr_data*, *label*)

Detail identifies the -Inf values

Parameters

- **bool** (*logical*) – decides if -Inf is not replaced with NaN
- **snsr_data** (*list*) – the dataset
- **label** (*character*) – the identifer for the dataset (e.g. sky, gro, skyo, groo)

Returns data set that replaces all -Infs for NaN (If bool == FALSE).

Return type list

index.**norm**(*x*)

Detail calculates the normalized index of the dataset

Parameters **x** (*double*) – data range

Returns an array of values between 0 and 1

Return type double

overcast.**filter**(*col_con*, *col_date*, *col_com*, *pw_name*, *snsr_name*, *cloud_bool*)

Detail Filters our data with overcast condition

Parameters

- **col_con** (*integer*) – column index for condition labels
- **col_date** (*integer*) – column index for date stamp
- **col_com** (*integer*) – column index for comments
- **pw_name** (*list*) – pw measurement labels
- **snsr_name** (*list*) – sensor labels
- **cloud_bool** (*logical*) –

Returns A list of lists containing either clear-sky/overcast data

Return type list

sky.**processing**(*overcast*)

Detail Computes average values and weighted averages

Parameters **overcast** (*list*) – results of the overcast.filter function

Returns series of arrays including average PWV, RH, etc.

Return type list

6.3 pmat_products.r

module Precipitable Water Model Analysis Tool: Products

synopsis plotting functions for PMAT

time.**pwindex**(*datetime*)

Detail Normalized PWV index for both clear sky and overcast data

Parameters **date** – the datestamp of the data

time.**nth_range**(*range*, *title*, *color*, *leg.lab*, *ylab*, *datetime*, *overcast*)

Detail Multirange Time Series plot series

Parameters

- **date** – the datestamp of the data
- **overcast** (*bool*) – the condition of data (clear sky/overcast)

time.**composite**(*range*, *title*, *color*, *ylab*, *datetime*, *overcast*)

Detail Time Series composite plot series

Parameters

- **date** – the datestamp of the data
- **overcast** (*bool*) – the condition of data (clear sky/overcast)

Returns A sky temperature time series plot

time.**multiyear**(*range*, *title*, *color*, *datetime*, *ylab*, *overcast*)

analysis.**nth_range**(*overcast*, *x*, *y*, *title*, *label*, *color*, *leg.lab*)

Detail Super Average Plot with Exponential Fit

Parameters **overcast** (*bool*) – the condition of data (clear sky/overcast)

Returns A sky temperature time series plot

analysis.**regression**(*overcast*, *x*, *y*, *des*, *label*, *iter*)

Detail Super Average Plot with Exponential Fit

Parameters **overcast** (*bool*) – the condition of data (clear sky/overcast)

Returns A sky temperature time series plot

analysis.**svm**(*model*)

pac.**compare**(*overcast*, *des*, *x*, *y*, *angular*, *radial*)

Detail Pac-Man plot of Super Average Plot

Parameters **overcast** (*bool*) – the condition of data (clear sky/overcast)

Returns A sky temperature time series plot

pac.**regression**(*overcast*)

Detail Pac-Man residual plot

Parameters **overcast** (*bool*) – the condition of data (clear sky/overcast)

Returns A sky temperature time series plot

chart.**histogram**(*range*, *xlabel*, *title*)

Detail Histograms of defined quantities

Parameters

- **range** – a data range
- **xlabel** – the xaxis label
- **title** – the title of the histogram

poster.**plots**(*overcast*, *iter*, *mean.out*)

Detail The set of all poster

Parameters **overcast** (*bool*) – the condition of data (clear sky/overcast)

Returns All available poster plots

poster1(...)

poster2(*overcast*, *iter*, *mean.out*)

Detail The analytics poster plot

Parameters **overcast** (*bool*) – the condition of data (clear sky/overcast)

sensor.**chart**(...)

Detail overcast distribution charts

sensor.**time**(*overcast*)

Detail Instrumentation time series plots

data.**gen**(*overcast*, *dir*)

Detail creates a datafile containing the date, avg temp, and avg pwv for a defined condition

Parameters

- **overcast** (*bool*) – the condition of the data (clear sky/overcast)
- **dir** – directory path

`data.`**`ml`**`(`*dir*`)`

Detail creates a datafile containing the machine learning relavant information

Parameters **`dir`** – directory path

`data.`**`step`**`(`*seed*, *i*, *coef*, *r*, *S*`)`

`data.`**`final`**`(`*dir*, *clear.len*, *over.len*, *train.len*, *nan.len*, *frac.kept*, *coef*, *std*, *rmse*`)`

`visual.`**`products`**`(`*set*, *mean.out*, *datetime=datetime*, *overcast=args$overcast*`)`

Detail saves plot sets

Parameters

- **`set`** (*`character`*) – the set identifier
- **`overcast`** (*`logical`*) – ovecast boolean

6.4 pmat_run.r

module Precipitable-Water Model Analysis Tool

synopsis The main file for PMAT. Documentation available at <https://docs.pmat.app>.

6.5 pmat_utility.r

module Precipitable Water Model Analysis Tool: Utility

synopsis general functions for PMAT

`logg`(*msglevel*, *msg*, *dir=out.dir*, *lev='INFO'*)

Detail creates log entries for _log.txt

Parameters

- **msglevel** (*character*) –
- **msg** (*character*) –

`first`()

`startup`()

Detail shows title banner for program

`closing`()

Detail cleans up files and ends the program

`reset_time`(*datetime*)

Detail A function that sets the time to 00:00:00

Parameters **datetime** (*character*) – a Date or datetime object

Returns A datetime object with time 00:00:00

Return type double

`time_axis_init`(*date*)

Detail A function that calculates the min, max, and position of the tick marks for

Parameters `date` (*double*) – A date or date-time object

Returns The max, min, and tick mark positions

Return type list

`time_axis`(*datetime*)

Detail A function that sets the x-axis format for time series plots

Parameters `date` (*double*) – A date or date-time object

`stnd_title`(*des*, *overcast*)

Detail A function that generates the title based on

Parameters

- `des` (*character*) – the description of the plot
- `overcast` (*logical*) – the sky condition

Returns a title string

Return type character