



**P<sub>D</sub> MAT**

# **Precipitable-Water Model Analysis Tool Documentation**

**Spencer Riley**

**Feb 05, 2022**



## CONTENTS:

<b>1</b>	<b>Getting Started</b>	<b>3</b>
1.1	Introduction . . . . .	3
1.2	Installation and Deployment Tutorial . . . . .	3
<b>2</b>	<b>Data Formatting</b>	<b>5</b>
2.1	Data Collection Guidelines . . . . .	5
2.2	Data Formatting . . . . .	5
<b>3</b>	<b>Changelog</b>	<b>7</b>
3.1	PMAT Cirrus . . . . .	7
3.2	PMAT Altocumulus . . . . .	8
<b>4</b>	<b>PMAT Reference</b>	<b>11</b>
4.1	pmat_analysis.r . . . . .	11
4.2	pmat_processing.r . . . . .	12
4.3	pmat_products.r . . . . .	13
4.4	pmat_run.r . . . . .	15
4.5	pmat_utility.r . . . . .	15



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.



## 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](https://template.pmat.app)<sup>1</sup>.
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 finalizing updates on *cool\_data.csv*, the workflow will run automatically and the visual and data products will be generated

---

<sup>1</sup> <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 virtual 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 Virtualization.

#### 1.2.5 development



## DATA FORMATTING

### 2.1 Data Collection Guidelines

First and foremost, collect honest data.

### 2.2 Data Formatting

As we have previously mentioned, PMAT required two input data files. In this chapter we will thoroughly explain how to properly complete either data file.

#### 2.2.1 Configuration Input

Provided below is a template of the `_pmat.yml` file that is given in the template repository. While there are some optional fields, most are required.

```
- sensor:
  name:
  error:
  color:
  ratio:
  range:
  emissivity:
  poster:
  active:
- train_fraction:
  value:
- rel_difference:
  value:
- iteration:
  step:
  seed:
---
- noaa:
  - label:
    id:
- wyoming:
  - id:
```



## 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.



## PMAT REFERENCE

### 4.1 pmat\_analysis.r

**module** Precipitable Water Model Analysis Tool: Analysis

**synopsis** This module contains analysis functions

`exp.regression(results, t, range=c(1:length(results$date)))`

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

**Parameters**

- **results** (*list*) – output of sky.analysis
- **t** (*double*) – training fraction
- **range** (*integer*) – range of date indices to be used

**Returns** returns the data series and model statistics

**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

`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 identifier for the dataset (e.g. sky, gro, skyo, groo)

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

**Return type** list

`iterative.analysis(overcast, dir, obool)`

**Detail** computes regression statistics and outputs to a yaml file

**Parameters**

- **overcast** (*logical*) – boolean to determine label
- **dir** (*string*) – directory file path for \_output.yml
- **obool** (*logical*) – determine whether to generate new \_output.yml

**Returns** iterative stats and \_output.yml

**Return type** list

**Todo** make the output file pass through a data.products function

`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

`sky.analysis(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

## 4.2 pmat\_processing.r

**module** Precipitable Water Model Analysis Tool: Pre-processing

**synopsis** functions for preprocessing

`colscheme(range)`

`mean.filter(pw, n)`

`data.partition(x, y, train_size=0.7)`

`dna.filter(date, comments, sns_r_sky, sns_r_gro)`

`overcast.filter(col_con, col_date, col_com, pw_name, sns_r_name, cloud_bool)`



## 4.3 pmat\_products.r

**module** Precipitable Water Model Analysis Tool: Products

**synopsis** plotting functions for PMAT

`time_series.plots(datetime, overcast)`

**Parameters**

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

**Returns** All available time series plots

`time9()`

**Parameters**

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

**Returns** A sky temperature time series plot

`time.nth_range(range, title, color, leg.lab, ylab)`

**Parameters**

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

**Returns** A sky temperature time series plot

`time.composite(range, title, color, ylab)`

**Parameters**

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

**Returns** A sky temperature time series plot

`analytical.plots(overcast, iter)`

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

**Returns** All available analytical plots

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

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

**Returns** A sky temperature time series plot

`plots4(overcast, iter)`

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

**Returns** A sky temperature time series plot

**plots5**(*overcast, iter*)

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

**Returns** A sky temperature time series plot

**pac.plots**(*overcast*)

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

**Returns** All available Pac-Man plots

**pac1**(*overcast*)

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

**Returns** A sky temperature time series plot

**pac2**(*overcast*)

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

**Returns** A sky temperature time series plot

**charts**(...)

**Returns** PDF of charts

**chart1**(*range, xlabel, title*)

**Parameters**

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

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

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

**Returns** All available poster plots

**poster1**(...)

**poster2**(*overcast, iter*)

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

**instr**(*overcast*)

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

**Returns** Instrumentation time series plots and overcast distribution charts

**chart**(...)

**time**(...)

**data.products**(*overcast, dir, i*)

**Returns** datafiles

**data1**(*overcast, dir*)

**Parameters**

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

**data2**(*dir*)

**Parameters** **dir** – directory path

**visual.products**(*set, overcast*)

**Parameters**

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

## 4.4 pmat\_run.r

**module** Precipitable-Water Model Analysis Tool

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

## 4.5 pmat\_utility.r

**module** Precipitable Water Model Analysis Tool: Utility

**synopsis** general functions for PMAT

**logg**(*msglevel, msg*)

**Parameters**

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

**first**()

**startup()**

**closing()**

**save**(*func, name*)

**Parameters**

- **func** (*list*) – the plotting function that will be saved
- **name** (*character*) – the name of the file with the plots

**Returns** A pdf of the plot set

**reset\_time**(*datetime*)

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

**Returns** A datetime object with time 00:00:00

**Return type** double

**time\_axis\_init**(*date*)

**Parameters** **date** (*double*) – A date or datetime object

**Returns** The max, min, and tick mark positions

**Return type** list

**time\_axis**(*datetime*)

**Parameters** **date** (*double*) – A date or datetime object

**std\_title**(*des, overcast*)

**Parameters**

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

**Returns** a title string

**Return type** character