**Analysis Tool for Lunar Crescent Visibility Criteria Using Integrated Lunar Crescent Sighting Database**

**Hilalpy**

HilalPy is a python library that function as an analysis tool for lunar crescent visibility criteria using integrated lunar crescent sighting database. HilalPy is the product of the thesis, as discussed in Chapter 4. HilalPy is deposited at Python Package Index, PyPi.

**Installation of Hilalpy and its dependencies**

HilalPy is located at the PyPi repository. This mean that Hilalpy is publicly available on the internet and can be download from any server. Installation of Hilalpy is as follows

1. Navigate to HilalPy repository at PyPi to confirm HilalPy version. HilalPy is updated from time to time, thus it is importance to ensure that installation of HilalPy uses the latest version of HilalPy.

**Figure 5.14. HilalPy Repository at PyPi**

Graphical user interface, text, application, email

Description automatically generated

2. Installation of HilalPy can be done by typing command “pip install HilalPy” on PowerShell Windows.

**Figure 5.15. Command for HilalPy Installation**

**Text

Description automatically generated**

**Figure 5.16. Successful Installation of HilalPy**

Text

Description automatically generated

4. Confirming the version of HilalPy. To confirm the version of HilalPy, a command “pip show HilalPy” can be entered on the PowerShell.

**Figure 5.17. HilalPy version confirmation command**

Text

Description automatically generated

**Figure 5.18. Hilalpy version confirmation result**

**Text

Description automatically generated**

5. Should the version of the Hilalpy is differ from the version of HilalPy from the PyPi website, the command “pip install Hilalpy --upgrade" can used to upgrade the python module.

6. Hilalpy has various dependencies in order to operate its lunar crescent observation analysis for lunar crescent visibility assessment. To install all HilalPy dependencies, a command below can be executed on the windows PowerShell.



**Figure 5.19. Command Execution for Dependencies Installation**

Text

Description automatically generated

Should all of the steps done properly, Hilalpy would be ready to use.

**Demonstration of Hilalpy**

HilalPy has three main function, visibility threshold analysis, contradiction rate analysis of condition style lunar crescent visibility criterion, and contradiction rate analysis of equation style lunar crescent visibility criterion.

**Lunar Crescent Visibility Variable Statistical Analysis**

Visibility threshold analysis requires four inputs from user, which are

1. figure

The output figure of the boxplot from the variable statistical analysis. The default location of the file is at the python output folder.



Should the user want the output to be located at the desktop folder, the input would be;



2. csv\_stats

csv\_stats is the descriptive statistics result of the variable analysis. It contains the information of the mean, median, 1st quartile, 3rd quartile, maximum and minimum of the variable parameter. User requires to input the intended location for the output file. The input is;



Should the user want the output to be located at the desktop folder, the input would be;



3. csv\_maxmin

csv\_maxmin is the variable parameter that sits at the minimum and maximum value. It represents as a threshold for lunar crescent visibility, and vital in the formation of a lunar crescent visibility criterion. User requires to input the intended location for the output file. The input would be;



Should the user want the output to be located at the desktop folder, the input would be,



4. x

x is the variable that the user intended to apply the statistical analysis. There is specific syntax for x, which user need to refer to. The syntax is as shown in Table 5.3.

**Table 5.3. Syntax of Lunar Crescent Visibility for Hilalpy**

|  |  |
| --- | --- |
| Variable | Syntax |
| Lag Time | LT |
| Moon Age | MA |
| Arc of Vision | ArcV |
| Arc of Light | ArcL |
| Different in Azimuth | DAZ |
| Width | W |
| Moon Altitude | MAlt |

User require to input the intended variable for statistical analysis. For example, if the user wants to conduct a statistical analysis for Lag time, the input would be;



Should the user want to conduct a statistical analysis for Arc of vision, the input would be;



5. To call the visibility threshold analysis , the code is



Should, all of the input is completed, the result are a boxplot and a descriptive statistical table, as shown in Figure 5.20.

**Figure 5.20. Hilalpy Variable Statistical Analysis Output**

Chart, box and whisker chart

Description automatically generated

Text

Description automatically generated

6. Use cases of HilalPy visibility threshold analysis.

A. User want to conduct a visibility threshold analysis on different in azimuth, and to store the output on the download folder.



The result would be as shown in Figure 5.21.

**Figure 5.21. Output File of Hilalpy Thres Function**

Chart, box and whisker chart

Description automatically generated

Graphical user interface, application

Description automatically generated

B. User want to conduct a visibility threshold analysis on width, and to store the output on the document folder.



The result would be as shown in Figure 5.22.

**Figure 5.22. Output File of Hilalpy Thres Function**

Chart, box and whisker chart

Description automatically generated

Graphical user interface, application

Description automatically generated

**Conditional Style Lunar Crescent Visibility Criterion Contradiction Rate Analysis**

Hilalpy has the function to perform a contradiction rate analysis on conditional style lunar crescent visibility criterion called Cond. Cond require numbers of inputs from the user which are,

1. *poserrorratedata*

*poserrorratedata* is the csv file of the negative data of lunar crescent sighting that located above the criterion. The coding input for the *poserrorratedata* is;

poserrorratedata **=** "Positive\_Error\_Rate\_Data.csv"

2. *negerrorratedata*

*negerrorratedata*  is the csv file of the positive data of lunar crescent sighting that located below the criterion. The coding input for the *negerrorratedata* is;

negerrorratedata = "Negative\_Error\_Rate\_Data.csv"

3. x

x is the conditional variable of the lunar crescent visibility criterion. x located at the x-axis of the graph simulation. x user input must follow syntax from Table 5.3.

4. y

y is the conditional variable of the lunar crescent visibility criterion. y located at the y-axis of the graph simulation. y user input must follow syntax from Table 5.3.

5. conditionx

conditionx is the conditional variable parameter of the lunar crescent visibility criterion. conditionx located at the x-axis of the graph simulation.

6. conditiony

conditiony is the conditional variable parameter of the lunar crescent visibility criterion. conditiony located at the y-axis of the graph simulation.

7. limitx

limitx is the conditional variable parameter of the lunar crescent visibility criterion. limitx located at the x-axis of the graph simulation.

8. limity

limity is the conditional variable parameter of the lunar crescent visibility criterion. conditiony located at the y-axis of the graph simulation.

9. Criterion

Criterion is the name of the assessed criterion. For example, if the tested criterion is MABIMS 2021, the input code is

Criterion = 'MABIMS 2021'

10. doc\_loc

The report of the assessment of lunar crescent visibility criterion, in a docx file. The input code is,

doc\_loc ="report.docx"

11. csv\_stats

csv\_stats is placeholder file to regenerate statistics inside the code. The input code is

csv\_stats **=**"stats.csv"

9. To use the HilalPy cond functions, the code is



Should, all the input is completed, a document file named report.docx will produce at the default python output folder, some of the analysis, is as shown in Figure 5.24.

**Figure 5.23. Output of HilalPy Cond function**

Chart, scatter chart

Description automatically generated

10. Use cases of HilalPy Cond.

A. User want to conduct a contradiction rate on a MABIMS 2021 lunar crescent visibility criterion, which has condition of; moon altitude larger than or equal to 3.0 degree, and arc of light larger than and equal to 6.3 degree. The code execution would be;



Some of the result is as shown in Figure 5.24.

**Figure 5.24. Output File of Hilalpy Cond Function**

Chart, scatter chart

Description automatically generated

A. User want to conduct a contradiction rate on a lunar crescent visibility criterion, which has condition of; different in azimuth larger than or equal to 4.0 degree, and width larger than and equal to 10 minute. The code execution would be;



The result is shown in Figure 5.25.

**Figure 5.25. Output File of Hilalpy Cond Function**

Chart, scatter chart

Description automatically generated

**Equation Style Lunar Crescent Visibility Criterion Contradiction Rate Analysis**

HilalPy has the function to perform a contradiction rate analysis on equation style lunar crescent visibility criterion called *equa*. *equa* require a number of inputs from the user which are,

1. *poserrorratedata*

*poserrorratedata* is the csv file of the negative data of lunar crescent sighting that located above the criterion. The coding input for the *poserrorratedata* is;

poserrorratedata **=** "Positive\_Error\_Rate\_Data.csv"

2. *negerrorratedata*

*negerrorratedata*  is the csv file of the positive data of lunar crescent sighting that located below the criterion. The coding input for the *negerrorratedata* is;

negerrorratedata = "Negative\_Error\_Rate\_Data.csv"

3. a

a is the equation variable of the lunar crescent visibility criterion. a located at the x-axis of the graph simulation. a user input must follow syntax from Table 5.3.

4. b

b is the equation variable of the lunar crescent visibility criterion. b located at the y-axis of the graph simulation. b user input must follow syntax from Table 5.3.

5. equation

equation is the equation of a lunar crescent visibility criterion. Equation is express in a in term of b. The equation must follow the python syntax for numeral expression and does not consider trigonometry function as an input. For example, Fotheringham lunar crescent visibility criterion is expressed as,

In the HilalPy *equa* function, Arc of Vision is expressed as a and equation, while difference in azimuth is expressed as x.

Another example is Alrefay lunar crescent visibility criterion, which expressed as,

In the HilalPy *equa* function, Arc of Vision is expressed as ‘a’ and ‘equation’, while width is expressed as ‘x’.

6. limita

limita is the conditional variable parameter of the lunar crescent visibility criterion. limita located at the x-axis of the graph simulation.

7. limitb

limitb is the conditional variable parameter of the lunar crescent visibility criterion. limitb located at the y-axis of the graph simulation.

9. Criterion

Criterion is the name of the assessed criterion. For example, if the tested criterion is MABIMS 2021, the input code is

Criterion = 'MABIMS 2021'

10. doc\_loc

The report of the assessment of lunar crescent visibility criterion, in a docx file. The input code is,

doc\_loc ="report.docx"

11. csv\_stats

csv\_stats is placeholder file to

8. To use the HilalPy *equa* function, the code is;



Should, all of the input is completed, a line graph overlay on scattered plot and a contradiction rate table shown as in Figure 5.26.

**Figure 5.26. Output of HilalPy *Equa* function**

Chart, scatter chart

Description automatically generated

10. Use cases of HilalPy *equa*.

A. User want to conduct a contradiction rate on a lunar crescent visibility criterion, which has equation of ;

. The code execution would be;



The some of the result is as shown in Figure 5.27.

**Figure 5.27. Output File of Hilalpy *equa* Function**

Chart, scatter chart

Description automatically generated

B. User want to conduct a contradiction rate on a lunar crescent visibility criterion, which has equation of ;

. The code execution would be;



The some of the result is shown in Figure 5.28.

**Figure 5.28. Result of the HilalPy *equa* Function**

Chart, scatter chart

Description automatically generated

**Multirange Lunar Crescent Visibility Criteria**

Multirange lunar crescent visibility criteria are criteria that based on ranges of visibility, commonly include, easily visibility by naked eye, visibility by optical aided and not visibility. The multirange lunar crescent visibility criterion is adopted by Yallop (1998), Odeh (2004) and Qureshi (2010) in their lunar crescent visibility criterion. HilalPy *equa* function able to examine the contradiction rate analysis on the multirange lunar crescent visibility criterion, however it requires a mathematical manipulation on the expression of the lunar crescent visibility criterion, Taking the example on Yallop Lunar crescent visibility criterion, Yallop equation can be manipulated to expressed ArcV in term of q and w.

As the value of q is already supplied and ArcV and W is based on lunar crescent sighting data, and contradiction rate analysis using HilalPy *equa* function can be conducted. For criterion A, HilalPy code execution and result would be



**Figure 5.29. Odeh Criterion A**

Chart, scatter chart

Description automatically generated

For criterion B, HilalPy code execution and result would be



**Figure 5.30. Odeh Criterion B**

Chart, scatter chart

Description automatically generated

For criterion C, HilalPy code execution and result would be



**Figure 5.31. Odeh Criterion C**

Chart, scatter chart

Description automatically generated

For criterion D, HilalPy code execution and result would be



**Figure 5.32. Odeh Criterion D**

Chart, scatter chart

Description automatically generated

For criterion E, HilalPy code execution and result would be



**Figure 5.33. Odeh Criterion E**

Chart, scatter chart

Description automatically generated

The user requires to combine the positive and negative contradiction rate to from the analysis of multirange lunar crescent visibility criterion.