
Eddington-GUI Documentation

Sagi Shadur

Feb 15, 2023

Contents

1	Installation	3
1.1	Community	3
1.2	Tutorials	5



Eddington is a user-friendly data fitting platform for common uses, based on open-source libraries such as *numpy*, *scipy*, and *matplotlib*. *Eddington* can be integrated with python code as a library, can be used as a CLI and has an easy to use GUI.

Note: This is the Eddington user interface documentation. In order to read about the Eddington python library go [here](#).

In order to install Eddington-GUI go to the [latest release page](#) and download the appropriate installation file.

1.1 Community

1.1.1 Contribution Guide

Thank you for choosing to spend your time contributing to the Eddington platform. Contributors are the living, beating heart of any open-source project, and we really appreciate each and every contribution made to our code-base, big or small.

Here is a relatively short guide on how to contribute code to Eddington. Please follow the next steps carefully when making a pull request.

Step 1 – Writing Your Code

Like any other open-source project in Github, contribution to Eddington is done via a [pull-request \(PR\)](#). Fork the desired repository, open a feature branch, and write your code in it.

When writing code, please pay attention that you:

1. Make sure your *master* branch is [up-to-date](#) with the latest changes in the Eddington platform, and make your feature branch based upon it. This will help you avoid merge conflicts.
2. Write your code clearly, with self-explainable variables, functions and classes.
3. Reuse existing code when possible.
4. Document new functions, classes, and modules (especially if they're public).

The code reviews you'll receive would often address the following guidelines, as well as any existing design issues.

Step 2 – Testing Your Code In Development Mode

While at the moment Eddington-GUI does not have unit tests, you can still run it in development environment in order to test the new feature you've been added.

In order to do so, follow the following steps:

1. Create a [virtual environment](#) in order to isolate your working space from outside dependencies that can affect Eddington-GUI.
2. Install [Briefcase](#) using `pip install briefcase`
3. Run Eddington-GUI in dev mode using `briefcase dev`

Note: Running `briefcase dev` will install automatically the required dependencies for Eddington-GUI. If you wish to install this dependencies without running the program, use `briefcase dev --no-run`

Step 3 – Cleaning Your Code

Writing a working code can sometimes be really hard, but writing a **clean** code is always harder.

Here on the Eddington platform we believe that code should be clean, and we want to make sure that writing clean code is as easy as possible. We do that by using automatic tools that would help you achieve that along the way.

We use state-of-the-art static code analysis tools such as *black*, *flake8*, *pylint*, *mypy*, *pydocstyle*, etc. [Statue](#) is orchestrating all these tools by running each of them on all of our code-base.

In order to use *Statue*, follow the next steps:

1. Run `pip install statue`.
2. Run `statue install`. If needed, this command will install missing packages.
3. Go to the main repository directory and run `statue run --context format`. This will change your code to fit styling guidelines. Save the changes in a commit or append them to an existing commit.
4. Run `statue run` again (now without any arguments) and it will check if there are any issues that it wasn't able to solve on its own. If there are any errors, fix them.
5. Save all changes in a commit or append them to an existing commit.

You may find some of the errors presented by those tools tedious or irrelevant, but rest assured that we take those errors seriously.

If you think that in a specific line an error should be ignored (using `# noqa` or `# pylint: disable` for example), please make sure that this skip is justified before applying it.

Step 4 - Testing Your Code in Production Mode

In order to make sure that your code works in production mode, follow the following steps:

1. Run `briefcase create`. This command will create a production environment for Eddington-GUI
2. Run `briefcase run`. This command will run Eddington-GUI in production mode.

Note: Once you did these two steps once, you can use `briefcase update` to re-install your code and `briefcase update -d` to reinstall dependencies. After you do that, run `briefcase run` again to re-run Eddington-GUI.

Step 5 – Adding Yourself to the Acknowledgment File

We acknowledge each and every one of our contributors in *docs/acknowledgment.rst*. Add your name to the contributors file, keeping alphabetical order.

Step 6 – Receiving a Code Review and Merging

Push the branch and open a PR. We will make our best efforts to review your PR as soon as possible.

Once you receive a code-review, address the issues presented to you by changing the code or commenting back. Once all the issues are resolved, your PR will be merged to master!

1.1.2 Acknowledgement

Creator

- Sagi Shadur

Logo Design

- Rotem Shadur

Contributors

- Assaf Zohar
- Ohad Nir
- Yuval Bardugo
- Ron Pelled

Special Thanks

- Adi Shadur for linguistic advisory.
- The amazing [Beeware](#) team, and especially Russell Keith-Magee, for their strong will to dream big and make everything python - possible.
- The warm and supportive members of the open-source community, who share their code and their hearts with the rest of the world.
- The great minds behind [numpy](#), [scipy](#) and [matplotlib](#), which without them this project could never come to exist.

1.2 Tutorials

1.2.1 Background

Hello, My name is Sagi Shadur, and I'm the creator of the Eddington platform. Eddington helps you fit data according to theoretical models, with easy to use tools such as the Eddington python library, Eddington CLI and Eddington GUI.

Eddington is an open-source platform, and as such, it's free to use. It's based on other state-of-the-art open-source projects such as numpy, scipy, matplotlib and etc.

I invite you to go to <https://github.com/EddLabs/eddington/> to learn more.

In the following videos, I will show you how to install and use Eddington-GUI. I want to thank you for choosing Eddington as your data fitting platform and wish you a good experience using it.

1.2.2 Step 1 - Installation

In this video, we will learn how to install Eddington GUI on your computer. First, open your browser and enter the following address: <https://github.com/EddLabs/eddington-gui/releases/latest>

This will open the latest Eddington GUI release page. Here you can see we have two installation files: MSI file for Windows and DMG file for Mac. Choose the appropriate file according to your operating system. I'm using Windows. Therefore I will download the MSI file.

Once you downloaded the installation file, run it, and follow the instructions. It may raise an error that the software is not authorized. Please ignore this error message and download it without any worries. *We're working on a solution to that problem.*

Once you've finished installing, search for Eddington in your toolbar search box, and you'll see it there. Now you can use the Eddington GUI.

Thank you for watching this video. Next, I'll show you how a fitting data file should look.

Note: If you're using **Mac**, please install the DMG file, extract the APP file and open it following the instructions described [here](#). This is required only on first opening.

1.2.3 Step 2 - Data Files

In this video, we'll see how fitting data files should look like for Eddington to read them.

Eddington accepts fitting data files in three formats: CSV, Excel and Json.

The fitting data in Excel files should be written as follows: it should have a headers' row with a name for each column, and the following rows should contain the data to fit, each row represents a data record.

Pay attention that you can have as many columns as you want. Once you open Eddington-GUI, you'll be able to choose which of them to use as your x-axis and which as your y-axis.

If you want to learn more about writing data files go to https://eddington.readthedocs.io/en/latest/tutorials/writing_data_file.html

Thank you for watching. In the next video, we'll see how to upload the fitting data to Eddington-GUI.

1.2.4 Step 3 - Upload Data

In this video, we'll see how to upload your fitting data to Eddington-GUI.

Open Eddington-GUI and click the "choose file" button. It will open a dialog that allows you to choose which file to upload to Eddington-GUI. Choose The appropriate file and click open.

If you choose an Excel file, Eddington-GUI will search for the first sheet with appropriate syntax, as we've seen in the previous video. You can change that sheet by clicking on the "sheet" selection dropdown. Pay attention that if you try to choose a sheet that does not match the Eddington syntax, an error will pop-up.

Next, you'll be able to choose which column to consider as your x-axis, x error axis, y-axis, and y-error axis. By default, Eddington will automatically choose the first 4 columns as your data columns

To view your data points on a graph, click "Plot data."

Thank you for watching this video. Next, we'll see how to fit the data according to a fitting function.

1.2.5 Step 4 - Fitting The Data

In this video, we'll see how to fit your data according to a fitting function.

Eddington offers you an array of default fitting functions such as linear, parabolic, hyperbolic, exponential, normal, and many others. You can choose which one of them to use by clicking the "Fitting function" dropdown. In our example, we'll select the hyperbolic fitting function.

Once you choose which fitting function to use, you can press the "Fit" button and get the fitting result. You can visualize your fitting by clicking the "Plot Fitting" button and see how close was your fit to the data. Lastly, you can press the "Residuals" button to see the errors of your fitting.

If you wish to add titles and labels to the axes, you simply enter them in the designated text box, that's located on the right of the main window. If you do not specify a title, the title would be the fitting function. As for the axes label, the default label would be the name of the column chosen for the x and y axes. You can also add grid lines to the plot by checking in the "Grid" checkbox.

If you wish to save your fitting results, first choose an output directory in which you want to save those results. Once you select the output directory, press the "Save" button, and all your fitting results and plots will be kept there.

Thank you for watching. Next, we'll see how to select and unselect records from your data.

1.2.6 Step 5 - Selecting Records

Sometimes, for various reasons, you want to omit a record from the fitting process. You may want it in the case that a record is contaminated or you see that it is very far-off from your fitting result. In this video, we will show you how to do so.

Instead of deleting the record from the file and reloading it, you can omit the record in Eddington-GUI directly. Just press the "Choose Records" button, go to the record you want to omit, and uncheck it. After selecting and deselecting all the records you wish to use, press "Save" and refit your data.

Pay attention, your selection won't take any effect unless you press "Save".

Thank you for watching. In the next video, we will demonstrate how to use custom fitting functions, differently from default ones.

1.2.7 Step 6 - Custom Fitting Functions

Sometimes, the default fitting functions presented by Eddington aren't enough for your specific use-case and you may want to write your own fitting function instead. In this video, you will see how to load a fitting function into Eddington-GUI and use it to fit the data.

Here, we'll assume you are familiar with the Python programming language and that you read the tutorial for how to write a fitting function. You can read this tutorial in the following URL (https://eddingon.readthedocs.io/en/latest/tutorials/writing_your_own_fitting_function.html).

Here, we wrote a fitting function in a python file called "fitting.py". You can see that we wrote a fitting function, wrapped with the "fitting_function" decorator, in order to indicate that this is the actual fitting function.

Pay attention, Eddington-GUI won't recognize a fitting function unless it is wrapped with the "fitting_function" decorator, and that the "save" parameter is not set to false, as described in the tutorial.

In Eddington-GUI's main window, click the "Load module" button and find the python module you've written. After you load the file, your new fitting function will be added to the fitting functions selection box. Select it and use it as any other fitting function.

Thank you for watching this video. In the next and final video we'll summarize everything we walked through in these videos

1.2.8 Summary

In these videos, we've scanned through the process of using Eddington-GUI as your data fitting platform. We've seen how to install it, how a data file should look like, and how to upload it. We've seen how to run the fit, how to display it in a plot, and how to save this plot and the results to files. Lastly, we've seen how to select and unselect records from the fitting, and how to upload your own fitting functions to Eddington-GUI.

I wish you a useful experience using Eddington-GUI. A lot of effort has been made in order to make the Eddington platform as easy to use as possible. If you find any bugs or issues, or if you wish to propose new abilities to the Eddington platform, I strongly encourage you to go to <https://github.com/EddLabs/eddington/issues> and post your ideas and thoughts there.

I am Sagi Shadur. Thank you for watching.

1.2.9 Frequently Asked Questions (FAQ)

Can't Open Eddington-GUI in Windows

On some Windows computers, Eddington-GUI fails to open due to a problem in *Matplotlib* (see [issue here](#)). In order to solve this problem install the *Visual C++ Redistributable for Visual Studio* from [here](#). After installing, Eddington-GUI should be able to open.