# Code review for Inverse Cosine

The review was conducted on August 2nd, 2021, using a bottom-up approach, starting from the file, and moving towards the code.

## The File

To begin, the function’s file name directly follows the PEP8 convention, which was the agreed-on standard for this project. The agreed upon standard was lowercase characters with underscore separation: standard\_deviation.py. The name also perfectly describes what the module is expected to carry and is straight to the point making the file access easy.



Figure : File name

## The Code

The file header is concise, making it clear exactly what this module will do.

Text

Description automatically generated

Figure : File header

One concern is the public declaration of the attribute “value”. Object oriented design guidelines suggest encapsulation of class attributes to avoid privacy leak and allow for manipulation of the attribute through safer means like accessors and mutators.

Text

Description automatically generated

Figure 3: Public attribute

The class and function headers are generally concise however, they’re in comment format. Following PEP 257 docstring conventions of python that are referenced in the PEP8 style guide it’s best to have class and function headers in docstring format similar to the file header. This however does not pertain to inline comments as they have a different purpose.

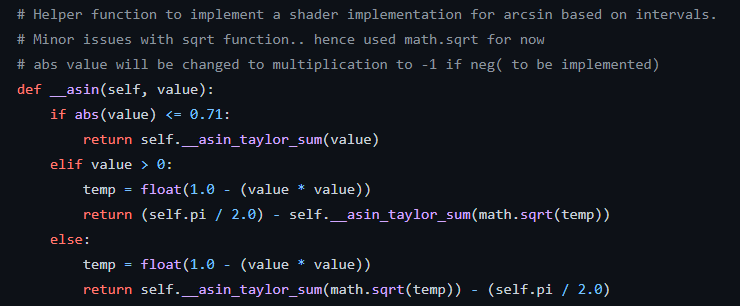


Figure 4: Function header

As for the implementation of the function and all its ‘helper’ functions, they are concise and serve a useful purpose. The code is well structured, it is very easy to read through and understand and follows the PEP8 naming conventions. The implementation of setters/getters is a nice addition in case the user wishes to reuse the object, therefore reducing the amount of program waste. In short, the code is very well written.