CRS\_auto Readme

Hello, this is a guide to using crs\_auto.py in order to perform constant rate of supply age calculations on sediment cores. In order to use this, you need to have a python3 installed on your computer with the packages Numpy and Pandas. If you do not have python installed, it’s recommended that you install python from the anaconda repository, which comes with Numpy and Pandas preinstalled, which can be found [here](https://www.anaconda.com/products/distribution). If you have python installed, but are missing [Numpy](https://numpy.org/install/) or [Pandas](https://pandas.pydata.org/docs/getting_started/install.html), the installation pages for both packages are linked here, which have tutorials on how to remedy that.

Set-up

In order to utilize this, you first must set up a csv with the correct data and column headers, so the program can read it properly. You need to have relevant data for the 5 following columns (column titles are case-sensitive)

1. depth\_low (cm)
   1. This is the lower depth of the range from which this gamma sample was taken. For example, if a certain data point came from a gamma sample taken from 130cm-131cm, that row would read 131
2. volume (cm3)
   1. This is the total volume of the gamma sample taken
3. weight (g)
   1. This is the dry weight of the gamma sample
4. 210Pb
   1. This is the lead 210 activity, as derived from the gamma data
5. 210Pb\_err
   1. This is the error in lead 210 activity, as derived from the gamma data as well

The first three the column header names include units, make sure you enter the data in those units to ensure the output has the correct units as well.

An example of a correctly formatted CSV can be seen below

Graphical user interface, application, table, Excel

Description automatically generated

To make a csv file, you can format the data with the correct column headers in excel, and then “file -> save as” and change the file type from “Excel Workbook (\*.xlsx)” to “CSV UTF-8 (Comma delimited) (\*.csv)”

Lastly, once the csv is prepared and properly formatted, it should be moved into the same folder as CRS\_auto.py for easy access by the program later on

Pre-analysis

Before using CRS\_auto.py, you must also do some preliminary analysis, identifying where the pb210 plateaus in the data (where it stops decreasing as you go deeper). When you do, record what row of the CSV file is the first one where the plateau happens and record that for input into the program. In the example below, at the lower depth of 9.5 cm is where this plateau is found, meaning it is in row 14.

Graphical user interface, table

Description automatically generated

Additionally, if you want to use a reference date, you need to determine the age of one row for use as the reference. If you are utilizing the Cs137 peak, you want to find what row of the csv file contains the highest Cs137 activity, and, if this is the Cs137 spike caused by large amounts of nuclear tests prior to the 1963 Partial Nuclear Test Ban Treaty, the age is how many years before the core was taken 1963 was. If this is the Cs137 spike caused by the Chernobyl disaster, the age is how many years before the core was taken 1986 was. Note that the Chernobyl spike will only be visible in cores taken in and around Europe and the Cs137 series will show a distinctive 2-peaked distribution. The highest point of any single-peaked distribution is almost-certainly caused by the 1963 test ban treaty.

You can optionally determine a constant percentage error in the dry measure measurements if you feel that the default value of 7% is not reasonable for this sample.

You can also optionally determine the Pb210 background to be provided as a parameter. If you don’t this is determined by looking at the average Pb210 activity of the equilibrium row determined and all rows after it. If there is an outlier in that set, it is recommended that you determine the Pb210 background activity via some other method and input that instead.

Running CRS\_auto.py

To run CRS\_auto.py, open command prompt (windows) or terminal (mac) on your computer; if applicable, activate your python environment that has Numpy and Pandas; navigate to the folder containing both CRS\_auto.py and your formatted CSV file. Tutorials on navigating folders in command prompt for [windows](https://www.howtogeek.com/659411/how-to-change-directories-in-command-prompt-on-windows-10/) and [mac](https://appletoolbox.com/navigate-folders-using-the-mac-terminal/) are linked here. Once there, you can run CRS\_auto by typing “python CRS\_auto.py”

If it works correctly, a prompt should appear asking for the filename. As a note on inputs, the way the python text interpreter works, it doesn’t like mixing numbers and letters, so when entering a measured or derived value, enter it as a unitless quantity (“14” instead of “14 years” for example)

Upon completion, a new csv file with the name you provided will be created in that same folder with the age of each measurement (how many years ago that layer was buried) and the uncertainty in that. If you want to get the year each layer was buried instead, simply do [year the core was taken] – [age]. The uncertainty is unchanged. You will notice that the ages only go up to the equilibrium row, since this method cannot get how old the sediment is after the lead 210 reaches equilibrium. Additionally, if you use a reference date, the age of any layer after the reference date cannot be gotten, so instead the ages of the layers between the reference date and equilibrium layer are found using the default reference-dateless method, which can lead to some weirdness with older sediment supposedly on top of younger sediments, so keep an eye out for that.