

CSCI 2000U: Assignment 4

due Dec. 06, 2015, 24:00 EST

Instructions

- Pull the most recent version of the class `csci-2000-class-resources` repository from GITHUB.
- Create the directory `csci-2000-personal/Assignments/Assignment-4/` within your personal repository (`csci-2000-personal`).
- Create a text file `README.txt` in `csci-2000-personal/Assignments/Assignment-4/` with your name and student number at the top.
- Add the file `README.txt` into version control and commit the changes with the comment "Assignment 4: Created my `README.txt` file."
- Commit changes between completing each question, i.e., *there should be 4 separate commits* (one for each question and one for the file `README.txt`) into your repository when the final solution is committed (you may choose to make more intermediate commits while constructing your solution).
- On completion, you should have added and committed the following files:
 - `README.txt`
 - `white_wash.ipynb`
 - `Source_data_for_CFR_vaccine_map_abridged.csv`
 - `cfr_map.ipynb`
 - `data_process.ipynb`
- Your files and directories *must* be named *exactly* as specified (penalties will be applied).
- In all your PYTHON scripts:
 - Put your name and student number in a comment at the top of the script.
 - Provide sufficient documentation to make your script clear and readable.
- To submit your assignment:
 - Edit the `README.txt` file to describe the contents of your directory (i.e., what the files are in the directory). You can also use the `README.txt` file to let us know if you received help from anyone (the instructor, the TA, or any of your peers) and to let us know any other comments you have.
 - Push changes from your local machine to your remote repository `csci-2000-personal` on GITHUB.

1. Write a PYTHON function `WhiteWash` as a `white_wash.ipynb` document that accepts a color map (i.e., an $m \times 3$ matrix of values between 0 and 1) as input and returns a new color map (i.e., a matrix of the same dimension) with each color C obtained by averaging with the color white (i.e., $[1, 1, 1]$).
 - The function `WhiteWash` should be a PYTHON function (not a script) that accepts exactly one input value and returns exactly one output value.
 - The input is a color map (i.e., an array of dimensions $m \times 3$ with entries between 0 and 1).
 - The output is also a color map (i.e., an array of dimensions $m \times 3$ with entries between 0 and 1).
 - You can use either PYTHON's list/tuple or NUMPY's array.

2. Your task here is to replicate a plot similar to that shown in the IFLScience article *One Map Sums Up The Damage Caused By The Anti-Vaccination Movement*. The original interactive map by the Council on Foreign Relations (CFR) also provides an Excel spreadsheet containing the relevant data.
 - (a) Use Earthquakes tutorial at <https://github.com/arminms/csci-2000-class-resources/blob/master/matplotlib/animation.ipynb> as a starting point, but remember that there is no need for the drip drop animation. A static plot would be enough.
 - (b) Download the Excel spreadsheet `Source_data_for_CFR_vaccine_map.xlsx` from the CFR's interactive map of vaccine-preventable outbreaks at http://www.cfr.org/interactives/GH_Vaccine_Map/#map (there is a button labelled "DOWNLOAD DATA" in the top-right corner of the web page). By a method of your choice, extract the data from the Excel spreadsheet columns A, E, F, and I (that is, the first, fifth, sixth, and ninth columns) into a comma-separated values (CSV) file `Source_data_for_CFR_vaccine_map_abridged.csv`. Save the abridged CSV file in your directory `csci-2000-personal/Assignments/Assignment-4/` and describe in your `README.txt` file precisely how you produced the CSV file.
 - (c) Write an IPYTHON NOTEBOOK document `cfr_map.ipynb` that does the following.
 - The `cfr_map.ipynb` parses the CSV file `Source_data_for_CFR_vaccine_map_abridged.csv` loading the relevant columns into NUMPY arrays.
 - The `cfr_map.ipynb` then uses the rows of the data that correspond to measles, and plots the world map as described in part (a) with discs coloured red. Give the plot the title *Measles outbreaks* along with your student number.

The centre of each disk is at the coordinates of an outbreak of measles; the radius of each disk is determined by the number of infections in the measles outbreak. To choose the radius of each disk, find the largest measles outbreak and select an appropriate radius for that outbreak. Scale the radii of the other disks for the remaining outbreaks as fractions of the largest radius. You will have to decide how to choose the radius; describe your choice in the file `README.txt`.
 - Repeat the preceding step with the rows corresponding to rubella outbreaks shown by magenta disks. Produce the plot in a new cell with the title *Rubella outbreaks* along with your student number.
 - Repeat the preceding step with the rows corresponding to mumps outbreaks shown by green disks. Produce the plot in a new cell with the title *Mumps outbreaks* along with your student number.
 - Repeat the preceding step with the rows corresponding to polio outbreaks shown by blue disks. Produce the plot in a new cell with the title *Polio outbreaks* along with your student number.

3. Download and extract the file archive `nbody.tar.gz` (this should take some time, the file is quite large) from BLACKBOARD. Within the `tar.gz` archive, you will find three files: `snapshot_010.dat`, `halo.dat`, and `README.txt`. Unpack these files within your directory `csci-2000-personal/Large_Data/`. The file `snapshot_010.dat` is a binary file that describe the masses, positions, and velocities of about 2 million bodies from a simulation of galactic evolution. The file `halo.dat` is a smaller subset of data points extracted from `snapshot_010.dat`.

As described in the file `info.txt`, each of the files `snapshot_010.dat` and `halo.dat` begins with a line containing a single 32-bit integer (the number of bodies described in the file) and a line containing a single 64-bit double precision floating-point number (the time associated with this snapshot of the simulation). Following the first two lines, each line contains seven 32-bit single precision floating-point numbers: in order, they are the mass m , the Cartesian coordinates (x, y, z) , and the velocity components (v_x, v_y, v_z) of one of the bodies.

For this exercise, you will create a single IPYTHON NOTEBOOK document `data_process.ipynb` that contains all the commands necessary to carry out the computations described below.

- (a) Use <http://stackoverflow.com/questions/8710456/reading-a-binary-file-with-python> at Stack Overflow to read the file `halo.dat` and extract all the data line-by-line.
- Read the first line into an integer variable `N_halo` (the number of bodies).
 - Read the second line into a double precision variable `T_halo` (the time of this snapshot).
 - Read the remaining lines into an array `data_halo` of dimensions $7 \times N_{\text{halo}}$.

Be careful to read the numbers in appropriately; see `struct.unpack` documentation for details to ensure that the binary data is read correctly.

- (b) Having extracted the data from `halo.dat`, compute a vector `v_halo` that stores the speed $v = \sqrt{v_x^2 + v_y^2 + v_z^2}$ of each body described in the file.

- (c) Plot a histogram of the speeds recorded in `v_halo` using the PYPLOT method `hist`.
- Use 50 bins for the histogram.
 - Label the axes appropriately.
 - Give the plot the title *Velocity histogram (halo)* along with your student number.

- (d) Repeat parts (a) through (c) using the larger data file `snapshot_010.dat` instead. You will want to name the variables `N_010`, `T_010`, `data_010`, and `v_010` instead. You will also want to give the new histogram the title *Velocity histogram (010)* along with your student number.