**DS2000**

**Spring 2021**

**Homework 6**

**Assigned:** March 5, 2021

**Deadline:** March 14, 2021 at 9pm eastern **<==== a little extra time!**

**Late Token Deadline:** March 17, 2021 at 9pm eastern

|  |
| --- |
| Submit one .py file for each problem of this homework (filenames are specified below). You may submit multiple times right up until the deadline.  Make sure you read this week’s [handout](https://bit.ly/3skxZ7Y).  Everyone gets one late token this semester, which gives you an additional 3 days to submit a homework without penalty. You do not need to request a late token ahead of time. To cash in a late token, simply email your .py files to laneys@northeastern.edu by the late-token deadline.  Apart from your late token, **no late submissions will be accepted.**  Your solution will be graded following the [DS2000 grading rubric](https://course.ccs.neu.edu/ds2000sp21/ds2000_rubric.pdf). Clear, readable code is part of the rubric; make sure you follow the [style guide](https://course.ccs.neu.edu/ds2000sp21/ds2000_styleguide.pdf).  We expect that you will study with friends and often work out problem solutions together, but your solution must be your own. **Do not share code.**  Your code must be organized into functions. We won’t tell you what functions to create, but we expect each function to...   * Have ONE JOB, * be generic, and * be short, clear, and concise.   For full credit, we expect all functions to be commented with name/parameters/return type, and for your **main** to be simple and clear. |

**Problem 1**

* Filename: qr.py
* Data file: [positions.csv](https://course.ccs.neu.edu/ds2000/code/hw6/positions.csv)

Use the csv file provided to generate a [QR code](https://en.wikipedia.org/wiki/QR_code) using matplotlib. (We did something similar in lecture, using **plt.plot()** to draw squares, row by row, until we eventually had a pixelated image.)

Build your QR code using two colors. In our solution, we used *magenta* and *yellow*, but any contrasting colors will do (black/white, pink/green, red/yellow, etc.) Here’s how to read the file:

* Row *i* of the file: All the positions where a *magenta* square would appear in row *i* of the QR code. All remaining positions in row *i* would be *yellow*.

For example, if a row of the file contains...

0, 1, 3, 6, 7

...then that row of the QR code has magenta squares at 0, 1, 3, 6, and 7, and yellow everywhere else.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

You can assume that there are 37 total squares per row. You should end up with a scannable QR code, which you should definitely scan to see an important result.

**Problem 2**

* Filename: quakes.py
* Data file: [earthquake\_data.csv](https://course.ccs.neu.edu/ds2000/code/hw6/earthquake_data.csv)

Download the CSV file linked above. We got it from the United States Geological Survey (USGS) website, which regularly releases information about earthquakes and other geological events. It contains information on all the earthquakes that were recorded in 2020 in the U.S.

For this problem, use matplotlib to create a bar chart showing the total number of earthquakes in each month of 2020. Colors and other details are up to you. For full credit, your plot must include appropriate title, labels, and xticks.

*Note about the data:*

The timestamp of each earthquake is in column zero. It looks like this:

2020-12-31T22:37:30.850Z

Of all that information, we’re interested only in the month. Try using the *split* function, which we already know and love. We’ve mostly used *str.split()*, but you can also specify the character to break on, such as *str.split("-").*

**Problem 3**

* Filename: You can build onto quakes.py, or if you prefer to create a new file please name it magnitudes.py

This problem will use the same earthquake data as you used for Problem 2. We’re still interested in plotting month-by-month data for 2020.

But now, instead of the raw number of earthquakes per month, we’re interested in the number of low-, medium-, and high-magnitude earthquakes that happened each month. You can use whatever ranges are appropriate, as long as you don’t end up with empty categories. (We tried setting medium to 3.0-4.0, with low anything smaller and high anything greater.)

Use matplotlib to create a bar chart with three bars per month -- one for low-magntiude quakes, one for medium, and one for high. Use different colors for low/medium/high, and include a legend in your plot.

For full credit, your plot must include appropriate title, labels, and xticks.