Black Death Assignment – Software Documentation

Contents

[1 Introduction 3](#_Toc502655187)

[2 Software Development Process 3](#_Toc502655188)

[2.1 Calculating number of black deaths 3](#_Toc502655189)

[2.2 Writing text to a file 3](#_Toc502655190)

[2.3 Plotting the maps and the data 4](#_Toc502655191)

[3 UML 4](#_Toc502655192)

1. Introduction

The Black Death Model aims to calculate the total number of deaths caused by the disease. It also aims to display the maps of the 100x100 meter areas where the deaths occurred. It displays both the given population density maps of the area as well as the average number of rats caught across those areas. Based on a given algorithm on how to calculate the absolute number of deaths, the program reads the input numbers from files and iterates through the different areas to give a final output as a text file holding the numbers. The code uses historical data from two csv files as input to display and calculate figures of total number of deaths across some given 100x100 meter area maps

1. Software Development Process

The software was developed initially as Python code in the Spyder IDE. It was developed and was running with input parameters given as input arguments given via the “Run Configuration per File” menu. The program displays the three different maps as output in the console and finally saves the total number of deaths as a text file called “absolute\_deaths.txt” in the same working directory as the code.

As a novice programmer, I had numerous issues when trying to solve the assignment. First of all I found it much easier to write the code in the Spyder IDE instead of in the Jupyter Notebook. I tried to copy the source code to Jyputer, but always had problems with displaying the data and the maps as the output. In the Spyder IDE, that worked better, so I did the development there. In the end I did not overcome the output issues in Jupyter, so the fully working code is to be run in the Spyder IDE. I used a lot of time searching for answers on online forums like Stack Overflow and using online manuals, but I could not figure out the plotting on the screen in Jupyter.

When I developed the code, I started with the code from Assignment 1 as input, but quickly noticed that there was not much of that I could use. It ended up being a completely new program instead. The first issues I had was to get the csv files read and their data inserted into the lists and being able to plot them out as graphical maps as output. I struggled a lot with getting the data properly inserted as lists within lists and had to try to visualise on paper what the data structure should look like. I also had problems in getting the correct data type inserted into the lists, since the values come as strings, but need to be integers. I wrote quite a lot of lines of code for each of the files separately, but finally saw that the code was essentially the same. So, I finally made the decision to place it in a “read\_file” function and call that with the filename as an input parameter. That call is done for each of the input files and the function returns a list of values.

* 1. Calculating number of black deaths

Once I had read all the file data and placed it in two different lists, I had to run the data through the given algorithm on how to calculate the total number of deaths per area. Here I struggled a lot with understanding how to run through the lists correctly and how to be able to set the values correctly. Again, I had to try to visualise the loops to be able to understand where to put the variable assignments to get the correct numbers out. I ended up creating a separate function for this as well, and this takes an input parameter defining how many loops are to be done. The value is set to 400, since the maps are in a 400x400 grid.

* 1. Writing text to a file

The “iterate\_lists” function also writes the calculated data to a text file, which is saved to the same directory as the code. I used a lot of time trying to find out how to write to a file and once I had solved that, how to get the data written on new lines in the file. I had to look up answers on various internet sites like Stack Overflow and try out many different ways before I got it working. I ended up using temporary variables to store data and then do operations on them before writing to the file. Now it takes the last character in the string and breaks the line after that.

* 1. Plotting the maps and the data

I ended up using matplotlib for plotting the maps on the console. I use a new figure (plt.figure()) call for each map. Initially I tried to plot all three lists as maps in one function call, but I could not get that working, so in the code the three lists are separately displayed. This is, however, something I could not get working on the Jupyter Notebook, so the only working version is to be run in the Spyder IDE.

1. UML

|  |
| --- |
| **bd\_model** |
| population\_density = []  rats\_caught = []  num\_deaths = []  txt\_list = [  rat\_weight = float  population\_weight = float |
| * read\_file(filename: string) : [] * iterate\_lists(num\_of\_iterations: int, txt\_list: [] rat\_weight: float, population\_weight: float) : [] * def write\_file(filename: string, txt\_list: []) |