

Concordia University
COMP 6721 Introduction of AI – Fall 2019
Project 1

Due Date:	By 11:55 pm Monday, October 14, 2019
Evaluation:	15% of the final mark
Late Submission:	20% per day per late submission
Purpose:	The purpose of this project is to help you learn Python semantics, syntax, libraries such as Numpy, Pandas, GeoPandas, Pyshp, Scikit-Learn and design heuristic search algorithms.
Programming Language:	Python
Teams:	Maximum two students in a team. The work should be evenly distributed to each team member. Teams must submit only one copy of your project via team leader.

General Guidelines When Writing Programs:

Include the following comments at the top of your source codes

```
# -----  
# Project (include number)  
# Written by (include your name and student id)  
# For COMP 6721 Section (your lab section) – Fall 2019  
# -----
```

- Include comments in your program describing the main steps in your program. The Focus in your comments rather on the why than the how.
 - Display clear prompts for users when you are expecting the user to enter data from the keyboard if needed.
 - All output should be displayed with clear messages and in an easy to read format.
 - End your program with a closing message so that the user knows that the program has terminated.
-

1. Montreal Crime Analytics

In this project, we locate four coordinates ([longitude, latitude]) (-73.59, 45.49), (-73.55, 45.49), (-73.55, 45.53), (-73.59, 45.53) in the center of Montreal downtown area. The blue square in the screenshot (see Figure 1.1) shows the located area. All the crime data in this area are saved in the file “*crime_dt.shp*”, which includes all the crime data considered at different points. The dataset is fetched and modified from the City of Montreal's open data portal.

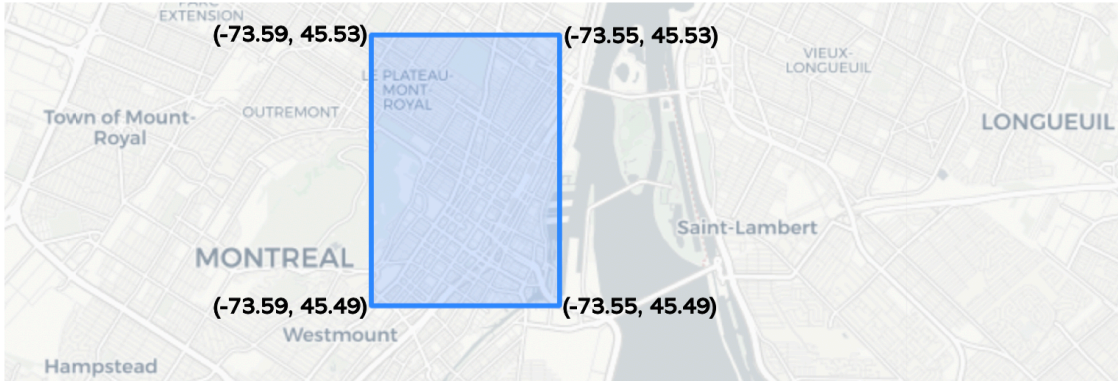


Figure 1.1 Located area in the center of Montreal map

Using this located area, you need to generate grids in this area to compute the crime rate statistics in each grid. The size of each grid could be changed due to the distribution of crime; however, your program should be flexible to change the size of each grid when it is required.

For example, in Figure 1.2 each grid has the size of $0.003 * 0.003$ in (a) and $0.002 * 0.002$ in (b). As the size of grids grows, the results of distribution may be focused on each grid evenly. Therefore, to ensure and display the accuracy of the data, smaller size less or equal to $0.002 * 0.002$ of each grid is recommended.

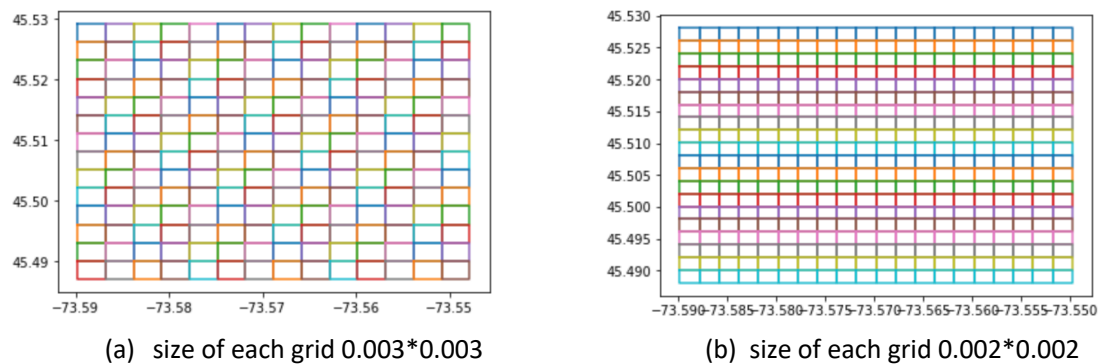


Figure 1.2 distribution of grids with different sizes

Crime rates are considered as the number of points in each block defined by the grid. The crime rates need to be calculated using statistics such as total count, mean, standard deviation in your results. A threshold of crime rate is introduced to define the high crime rate blocks and low crime rate blocks in the area selected. The threshold would be arbitrarily chosen. For example, if you choose the threshold using the median (the crime rate of a block higher than 50% of the crime rates of total blocks), blocks with the crime rate higher than median are labelled in yellow (highly risky blocks), and blocks with the crime rate lower than median are labelled in dark blue (less risky blocks). According to

each threshold, the crime rates in the located area will be presented as the following graphs.

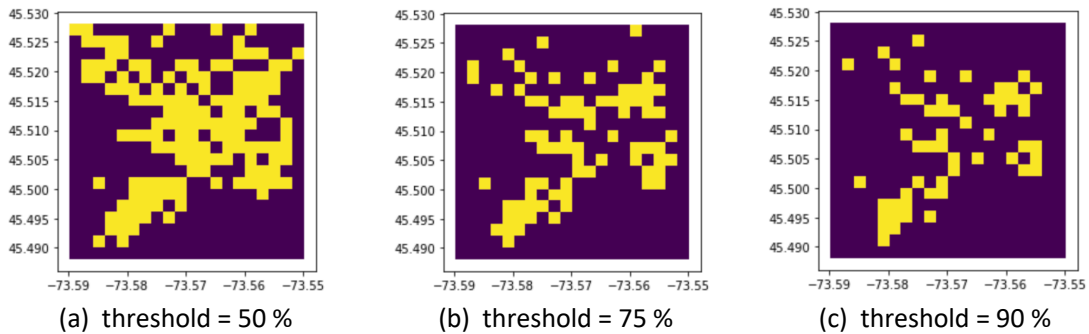


Figure 1.3 Using different threshold to present the crime rates on the map

2. Your Task

In this project, you will generate the crime risk map based on the crime rates and implement a heuristic search algorithm to find an optimal path between two coordinates on the map.

2.1 Display Results

1. Your program should be able to read the map area located in the area of four coordinates ([longitude, latitude]) (-73.59, 45.49), (-73.55, 45.49), (-73.55, 45.53), (-73.59, 45.53) in the center of Montreal downtown area.
2. Your program can draw the grids and generates graph indicating the crime rates in each grid using the "crime_dt.shp" file.
(Please note: to load and read the "crime_dt.shp" file, the other four files "crime_dt.cpg", "crime_dt.dbf", "crime_dt.prj", and "crime_dt.shx" in the same folder named "Shape" attached should be put in the same local path.)
3. Compute the number of total crimes, mean, average and standard deviation in each grid and generate graphs displaying the crime rates using different colors based on the local area.
4. Using the graph, which includes yellow blocks considers as obstacles, in the example above illustrates the high crime rate areas (see Figure 1.3), your program should implement a heuristic algorithm to find an optimal path from (-73.55, 45.49) to (-73.59, 45.53).
Please note your path should avoid all the high-risk areas in your map and the path could be changed according to the different chosen thresholds.
5. *Bonus points will be given if your program is implemented with two different heuristic algorithms. You need to compare both and explain their efficiency in your report.
6. *Bonus points will be given if your program can ask the user for the inputs of longitude, latitude and accordingly the distribution grids will be generated, an optimal path will be found as well. Please find the files of crime rates data in the attached folder name "Bonus".

2.2 Programming Details

1. To implement this project, you must use Python.
Please note the libraries can be used but not limited to Numpy, Pandas, GeoPandas, Pyshp, Scikit-Learn. If you use other libraries, please give a brief explanation when you demonstrate your program.
2. Your program must be able to run on the lab machines and must find the optimal path at most 10 seconds (real time in the lab).
3. It is not necessary to have a fancy user-interface. A simple command-line interface with a direct output is sufficient.
4. All the results listed in Section 2.1 *Display Details* should be presented in GUI and it should be easy to change the input grid size and the chosen threshold using the command-line interface when it is required.

3. Deliverables

3.1 Evaluation Scheme

Students will be given individual grades that will be a function of the team grade and the peer-evaluation.

At the end of the project, all team members will fill out a peer-evaluation form to evaluate the contribution of each team member. The grade of a student will be a function of his/her team grade and the peer evaluation received.

3.2 Demos

Project 1 will be demonstrated on the lab machines. You will not be able to demo on your laptop. Regardless of the demo time, you will demo the program that was uploaded as the official submission on or before the due date. The schedule of the demos will be posted on Moodle. No special preparation is necessary for the demo (no slides or presentation needed). Your TA will ask you questions about your code, and you will have to answer all the questions.

3.3 Report

Please note that this is a project, not an assignment. The difference is that is open-ended and in addition to the required work started above, you are expected to be creative, perform additional experiments, comparisons, and analysis. This means that in addition to stating *what* you did, you should also describe *why* you did it that way and what other options were available and why they were not retained.

Your report should have a reference section (not included in the page count) that properly cites all relevant resources that you have consulted (books, papers, websites, etc.), even if it was just to inspire you. Failure to properly cite your references constitutes plagiarism and will be reported.

Requirements:

Your report should be 5-6 pages (without references and appendix) and use the template provided on Moodle. The report should contain at least the following:

1. ½ to 1 page: Introduction and technical details.

2. 1 to 2 pages: A description and justification of your data analysis and heuristic algorithms. In particular, describe how you came up with it, what features of analysis and states were considered and why you chose them, how you solved the problems of using grids and apply it on the located map, and the heuristics used to find the optimal path.
3. 1 to 2 pages: A description and analysis of your result about the different thresholds presented which could be related to the strength/weaknesses of your heuristics used to find the path. Explain how you determined the proper size of grids and threshold to find the optimal path.
4. ½ to 1 page: A description of any difficulties that you have encountered and how you addressed them.
5. ½ page: In case of teamwork, a description of the responsibilities and contributions of each team member.
6. Not included in the page count: References, Appendix, Figures, Tables, and pseudocode.

Submitting Project 1

Your submission should include the following documents:

1. Read and sign the expectation of originality form (available on Moodle or at <https://www.concordia.ca/content/dam/ginacody/docs/Expectations-of-Originality-Feb14-2012.pdf>)
2. Create a README.txt file, which will contain specific and complete instructions on how to run your program on the lab computers. If the instructions in your file do not work or incomplete, you will not be given the benefit of the doubt.
3. Create one zip file, containing all your code, the README.txt, technical report and the expectation of originality.
4. Name your zip file: **P1_StudentID.zip** if there is only one student in the team;
P1_StudentID1_StudentID2.zip if there are 2 students in the team.
5. Have the team leader upload the zip file on the Moodle webpage.

Note: please check your course Moodle webpage on how to submit the project and follow the instructions to submit your project. Wrong submission files and codes will not be accepted.

6. In addition, print your report and submit a paper copy to your TA when you give your demonstration.

Have fun!!