In this folder, there's an example metadata.csv file that contains information of an image collection run,
the csv format is:
09/04/19, 09:02:58, SamsClub6235_090419_090258_765350.jpg, 0.9012, -0.4137, 4.6466, 1335
date, time, filename, x, y, theta, distance
where
x, y, theta - robot coordinates in robot frame in meter
distance - how far robot is from the obstacle in mm

Task 1:
Write some filter logic that takes the path to the .csv file as input, and outputs a new .csv to the given output path, while filtering out images taken outside of desired distance ranges, for which the range will be given as input.

This main.py is to be called in the terminal with user inputs.

The resulting filtering operation can be used, in terminal, with these input params
python main.py path_to_input_csv path_to_output_csv min_dist max_dist

For example,
python ~/test/main.py ~/test/metadata.csv ~/test/output.csv 1000 2500

will take the input metadata.csv, filter out and remove all rows in which the distance value is out of range (given by above inputs) and write to output.csv

- os.path.expanduser(path) might be useful in dealing with '~' in relative file path
- sys.argv or argparse.ArgumentParser might be useful to grab input parameters

for the sake of this exercise, it can be assumed that all inputs are valid, i.e. no need to do input validation

Task 2:
On top of (or modify) what you wrote for the simple logic to filter by range, at the end of the filter operation output some stats about the result. Simple print() would be sufficient.

Useful stats here might be
- # of total rows
- # of rows discarded
- % of pictures within desired range
- total execution time of the filter operation

Task 3:
In a new script, design and implement a class interface with the same functionalities as above tasks. Include in main() to show how to interact with the class