Histogram program

Documentation

Yuna Otrante-Chardonnet

May 2021

Content

I. Introduction:

The goal of my project is to create a program that computes the distance between each pair of cows of the barn over a whole day or over a giver timeframe. The program should also be able to compute the distance between two given individuals.

II. Information about the program

a. Difficulties :

There were two main difficulties:

- The missing data and therefore the missing coordinates.
- The large set of data (=> could make the program slower)

b. Required files to use the program:

In the code folder:

- Distance.py
- Functions.py
- Initialization.py
- Interface.py
- Main.py

In the data folder:

- Barn.csv
- FA data file (.csv)

Barn.csv is a file containing the coordinates of all the different areas. Those coordinates are used by the program to separate the areas so if an area is modified, the coordinates need to be changed in this file.

c. Computation time:

For one cow ≈ 30 seconds
For 200 cows ≈ 1h40min

III. How to use the program:

1. Run the main.py file

   The interface is displayed

2. Select a date (the matching FA data file must be in the data folder)
3. Click on the “Load data from file” button
4. Choose a timeframe
5. Set the number of bars in the histogram
6. Select the area
7. Optional: tick the checkbox if you want both cows to be in the chosen area at the same time (otherwise only cow1 will have to be in the area)
8. Optional: tick the checkbox if you want to save in the “merged data” folder the merge data file which contains the distance between two cows for every second.
9. Cows:

   9.1. Get the histogram between two cows:

   - Enter tag ids of the cows
   - Click on the “Get histogram” button

   9.2. Get the histogram for all the pairs:

   - Click on the “Get all histograms” button
   - The created histograms are saved in the “histograms” folder (data/YYYY-MM-DD/histograms)

The computed data is saved in the “cows” folder (data/YYYY-MM-DD/cows) and is reused to avoid remaking the computation.

IV. Explanation of the code

Rounded time: epoch time rounded to the second

Initialisation.py:

csv_read(file):

Parameters: .csv file (FA data)
Return: Dataframe
Description: Creates a dataframe from a .csv file and drops the inactive tags.

csv_read_bis(file):

Parameters: .csv file (FA data)
Return: Dataframe (6 cows)
Description: Creates a dataframe from a .csv file just used for the demo.

csv_read_cow(file):

Parameters: .csv file (FA data of one cow)
Return: Dataframe
Description: Creates a dataframe from a .csv file to retrieve the cow's information.

feeding_area(df_cow, barn):

Parameters: cow's dataframe, barn's dataframe
Return: Dataframe
Description: The function determines the extremities of the feeding areas and returns the cow's dataframe with only the rows matching the feeding area.

bedding_area(df_cow, barn):

Parameters: cow's dataframe, barn's dataframe
Return: Dataframe
Description: The function determines the extremities of the beds and returns the cow's dataframe with only the rows matching the bedding area.

cubicle_area(df_cow, barn):

Parameters: cow's dataframe, barn's dataframe
Return: Dataframe
Description: The function determines the extremities of the cubicles and returns the cow's dataframe with only the rows matching the area.

alley (df_cow, barn):

Parameters: cow’s dataframe, barn’s dataframe

Return: Dataframe

Description: The alley is the barn without the cubicles, the bedding, feeding areas so the function returns the cow’s dataframe without the rows matching those areas.

custom_area (df_cow, x1, x2, y1, y2):

Parameters: cow’s dataframe, X and Y coordinates

Return: Dataframe

Description: The function returns the cow’s dataframe with only the rows matching the custom area.

area_delimitation (df_cow, area):

Parameters: cow’s dataframe, area’s name

Return: Dataframe

Description: The function creates the dataframe of the barn and drops the rows of the cow’s dataframe which are outside of the barn area. If the area is “feeding”, "bedding", "cubicle" or “alley” then it calls the proper function to format the cow’s dataframe to the appropriate area.

create_cow (cow_id, df, e_min, e_max, area):

Parameters: tag_id, dataframe, starting and ending epoch time, area’s name

Return: Dataframe

Description: Creates a subset from a dataframe matching the given tag id, starting and ending epoch time and area.

create_coordinates (df_cow, i, time_gap, t, t2, x2, y2):

Parameters: cow’s dataframe, index of a gap/missing data, array of the “rounded_time” column of the dataframe, array of every second in epoch time between the first timestamp of the cow’s dataframe and the last one, arrays of the interpolated coordinates (X,Y)

Return: Dataframe

Description: Add the missing coordinates to the dataframe by data interpolation

fill_data (df_ cow, area):

Parameters: cow’s dataframe, area’s name

Return: Dataframe

Description: Computes the missing data

Distance.py:

compare(df_cow1, df_cow2):

Parameters: dataframes of the cows
Return: Dataframe
Description: Compares dataframes of two cows and return the merged dataframe when the rows match (on rounded time)

compute_distance(df):

Parameters: The merged dataframe from the compare function
Return: Dataframe
Description: Computes the distance between two cows for each row of the merged dataframe

histogram(df, show, path, nb_bar, t_min, t_max):

Parameters: The merged dataframe from the compare function, path to know where to save the histogram, variable to show or save the histogram, number of bars to plot, the starting and ending time.
Return: None
Description: Creates a histogram and if show == 1 than it shows it otherwise it saves it in the given folder.
File name pattern: TAG ID1_ TAG ID2_STARTING TIME_ ENDING TIME_AREA (_both).png

Interface.py:

clicked_file():

Parameters: None
Return: None
Description: Loads the data from a .csv file to create a dataframe using the date that the user gave. The function also creates folders (if they do not exist yet) to store the data that will be created.

compute_cow (path, cow_id, e_min, e_max, area):

Parameters: path to save cow's data, tag_id, starting and ending epoch time, area's name
Return: Dataframe
Description: Checks if the wanted data has already been computed.
If the data exists (for the given area, or for the whole barn), the file is loaded and if needed, the area is delimited, and the new data is saved in a file.
If the data does not exist, then the cow's dataframe is created and saved in a file in the "cows" folder.
The file name pattern is TAG ID_STARTING TIME_ ENDING TIME_AREA.csv

get_histogram ():

Parameters: None
Return: None
Description: Create histogram for a pair of cows.

compute_side(list_side, e_min, e_max, t_min, t_max, nb_histo, side, i):

Parameters: list of all the ids of the cows on the chosen side of the barn, starting and ending epoch time, starting and ending time, the number of histograms that will be created, the side of the barn, the number of histograms created.
Return: the number of histograms created
Description: Creates histograms for cows on one side of the barn.

get_all_histograms ():

Parameters: None
Return: None
Description: Separates left cows from right cows and calls the function "compute_side" to get all the histograms for each side.

quit_me():

Parameters: None
Return: None
Description: Quits the interface

defaul_nb_bar()

Parameters: None
Return: None
Description: Sets number of bars to 50

is_custom(value)

Parameters: None
Return: None
Description: Displays entries to enter the coordinates or hide them