# Created functions and calculated parameters:

**PART 1: read and filter data, keep only fish points in the river**

1. Function to return all full path of files in a directory
2. Function to get for every fish type all corresponding files
3. Function to create a dictionary in a way for every fish type (key) all files are present (values).
4. Function to read the dataframes of the receivers (stations)
5. Function to read the observed fish dataframe and make index a datetime object
6. Function to convert coordinates from longitude, latitude (wgs84) to x, y (utm32)
7. Function to check if a point is in the river shapefile polygon or not
8. Function to find if the observed fish points are in the river polygon and add information to the dataframes
9. Function to filter the data, keep only points that are in the river, save output to new dataframes in the form 'fish\_xxxxx\_points\_in\_river.csv'
10. Function to read the dataframe with only the points in the river and to drop all non-required columns

* **Parameters in part 1: intersect observations with river shapefile, save output**

**PART 2: read dataframes points in the river, filter based on Velocity, HPE, RMSE**

1. Function to calculate to distance between two consecutive points
2. Function to calculate the fish swimming velocity between two consecutive points
3. Function to filter the data based on a variable threshold (below or above) and to keep first observation (variable example: Velocity or HPE or RMSE)
4. Function to filter the data based on a variable threshold (below or above) and to keep second observation (variable example: Velocity or HPE or RMSE)

1. Function to filter the data based on two variable thresholds and to keep first observation (example: HPE and RMSE)
2. Function to filter the data based on one variable conditioned on other two variables (example: filter velocity based on HPE and RMSE values)

* **Parameters in part 2: Velocity, travelled distance, filter based on Velocity, HPE, RMSE, save output dataframes**

**PART 3: read dataframes points in the river and filtered based on Velocity, HPE, RMSE, plot different variables, Heatmaps, filter and plot based on time periods**

1. Function to generate google earth KML and KMZ files for longitude and latitude
2. Function to transform two variables to uniform marginal and to plot the pure dependence between the two variables
3. Function to make a customized colormap (similar to jet)
4. Function to plot the histogram of a certain variable (Velocity, HPE, RMSE)
5. Function to plot the Ortho image of the study location
6. Function to plot the spatial distribution of the different variables (Velocity, HPE, RMSE), used once before filtering and again after filtering the data
7. Function to calculate weights to be used in the heatmaps (the time difference between each location and the next one is used as a weight for every point)
8. Function to calculate and plot the heatmaps (either with or without weights, with or without background Ortho image)
9. Function to plot the time difference between two points (for every consecutive two points find the difference in time of observation and plot this information)
10. Function to select a dataframe between two time periods
11. Function to filter the data based on 3 predefined time periods and to plot the locations per period
12. Function to save the filtered dataframe per period and fish type

* **Parameters in part 3: histogram of each variable, scatter plot of the marginal between two variables, plot spatial distribution of the variables, do heatmaps, weight for heatmaps (time difference), plot time difference between observations**

**PART 4: combine and plot fish observations with flow data and river hydraulics**

1. Function to calculate the angle between two consecutive fish positions (fish swimming direction as compared to the x-axis)
2. Function to calculate the angle between the flow vectors (Vx, Vy), find flow direction as compared to the x-axis
3. Function to plot the movement direction between two Fish positions
4. Function to aggregate the values in a grid cell and find mean behaviour per grid cell (example find mean fish swimming direction and swimming velocity per grid cell)
5. Function to plot the average grid cell values (velocity and swimming direction)
6. Function to read the observed flow data and observed fish locations, find for every position, based on time of measure the corresponding observed flow and categorize the output based on different flow categories. Save the resulted dataframes
7. Function to calculate the difference in the fish swimming direction and flow direction, add information to the dataframes (fish\_swimming\_direction, flow direction, angle difference between both)
8. Function to plot the difference between fish swimming direction and flow direction and categorize it (with flow, against flow, perpendicular to flow)
9. Function to calculate the difference between depth and flow values between every node and surrounding 4 nodes, find the maximum difference, calculate the angle between the fish swimming direction and the direction of the maximum gradient difference, add information to the dataframe

* **Parameters in part 4: link flow data to fish observations, for every position assign corresponding hydraulic information (nearest node grid, coordinates of node, depth, flow vectors and magnitude), find fish swimming direction, flow direction, difference between both, find direction of maximum gradient and calculate difference between swim direction and direction of maximum gradient for depth and flow data**

**PART 5: do 3d plots to deduct behaviour of fish within time**

1. Function to plot 3d plots of x,y, time along the fish path location, use at first time of day as colours (blue is morning, red is night) and then use the flow velocity as colour (blue is low, red is high)