In [98]:
```
# author: Nikolas Hawley
# course: CEIE 450
# date: 202240925
```

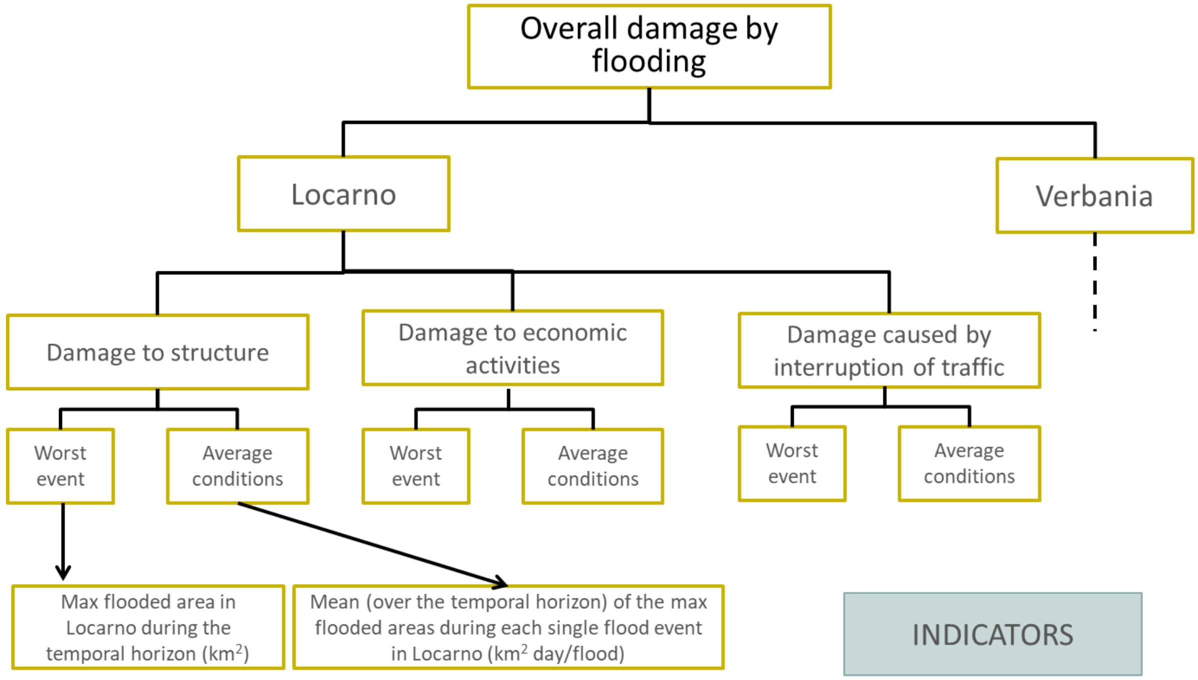
LAB 1

CEIE 450 - Environmental Systems Engineering
25 September 2024

PROBLEM:

Your team of consultants is asked to provide indicators for the Lake Maggiore Upstream Sector whose main objective is to minimize upstream flooding.

You are given historical data of Lake Maggiore water level data [m] measured at Sesto Calende from 01/01/1974 to 10/21/1998 and corresponding daily max flooded area.



We begin our python notebook by importing libraries and packages we will use in our analysis...

In [99]:

```
# imports
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from datetime import datetime
```

Next, we extract the given dataset(s) from given raw .txt file(s) to pandas dataframes and view the head and descriptive statistics of each dataframe...

In [100...

```
# read file with data
# ...txt
df00 = pd.read_csv('LakeMaggioreWaterLevels.txt')
num_rows = len(df00)
start_date = '1974-01-01'
df00['date'] = pd.date_range(start=start_date, periods=num_rows)

# print stats and the first 5 rows of the dataframe
print('mean: \n', df00.mean())
print('\nvariance: \n', df00.var())
print('\nstd dev: \n', df00.std())
print('\ndf length:', len(df00))
df00.head()
```

```
mean:
 h (m)    0.912083
dtype: float64

variance:
 h (m)    0.2243
dtype: float64

std dev:
 h (m)                          0.473603
date     2618 days 10:16:02.297052704
dtype: object

df length: 9070
```

```
C:\Users\Niko\AppData\Local\Temp\ipykernel_18412\1491118473.py:9: FutureWarning: Data
Frame.mean and DataFrame.median with numeric_only=None will include datetime64 and da
tetime64tz columns in a future version.
  print('mean: \n', df00.mean())
C:\Users\Niko\AppData\Local\Temp\ipykernel_18412\1491118473.py:10: FutureWarning: The
default value of numeric_only in DataFrame.var is deprecated. In a future version, it
will default to False. In addition, specifying 'numeric_only=None' is deprecated. Sel
ect only valid columns or specify the value of numeric_only to silence this warning.
  print('\nvariance: \n', df00.var())
```

Out[100]:

	h (m)	date
0	0.88	1974-01-01
1	0.89	1974-01-02
2	0.91	1974-01-03
3	0.94	1974-01-04
4	0.96	1974-01-05

In [101…

```
# read next file with data
# ...txt
df01 = pd.read_csv('LakeMaggioreFloodedAreas.txt')
num_rows = len(df01)
start_date = '1974-01-01'
df01['date'] = pd.date_range(start=start_date, periods=num_rows)

# print stats and the first 5 rows of the dataframe
print('mean: \n', df01.mean())
print('\nvariance: \n', df01.var())
print('\nstd dev: \n', df01.std())
print('\ndf length:', len(df01))
df01.head()
```

```
mean:
 Flooded Area [km2]    0.035634
dtype: float64

variance:
 Flooded Area [km2]    0.026964
dtype: float64

std dev:
 Flooded Area [km2]                           0.164206
date                    2618 days 10:16:02.297052704
dtype: object

df length: 9070
```

```
C:\Users\Niko\AppData\Local\Temp\ipykernel_18412\3464415806.py:9: FutureWarning: Data
Frame.mean and DataFrame.median with numeric_only=None will include datetime64 and da
tetime64tz columns in a future version.
  print('mean: \n', df01.mean())
C:\Users\Niko\AppData\Local\Temp\ipykernel_18412\3464415806.py:10: FutureWarning: The
default value of numeric_only in DataFrame.var is deprecated. In a future version, it
will default to False. In addition, specifying 'numeric_only=None' is deprecated. Sel
ect only valid columns or specify the value of numeric_only to silence this warning.
  print('\nvariance: \n', df01.var())
```

Out[101]:

	Flooded Area [km2]	date
0	0.0	1974-01-01
1	0.0	1974-01-02
2	0.0	1974-01-03
3	0.0	1974-01-04
4	0.0	1974-01-05

Both of the datasets match the same timeframe, period and number of rows, and can be combined using `pd.merge()` function into a single dataframe...

In [102...

```
df = pd.merge(df00, df01)
# print the first 5 rows of the dataframe
print('df length:', len(df))
df.head()
```

```
df length: 9070
```

Out[102]:

	h (m)	date	Flooded Area [km2]
0	0.88	1974-01-01	0.0
1	0.89	1974-01-02	0.0
2	0.91	1974-01-03	0.0
3	0.94	1974-01-04	0.0
4	0.96	1974-01-05	0.0

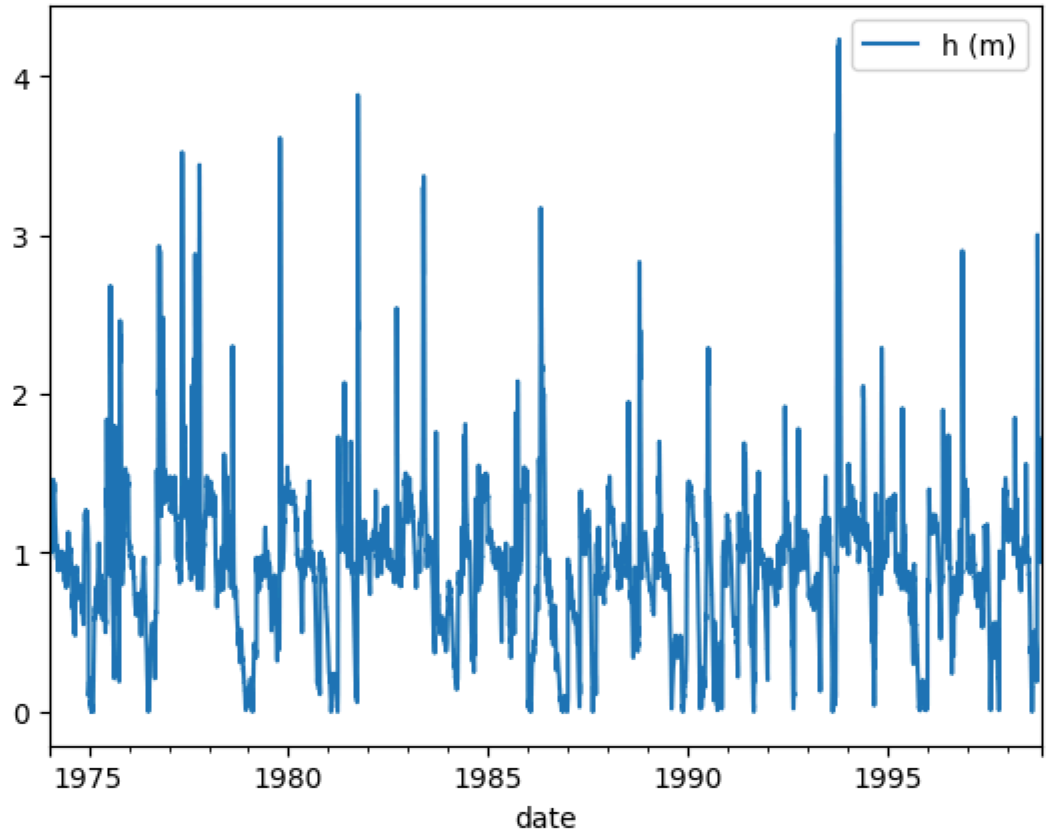
Plotting each timeseries...

In [103…

```
# PLOT...
df00.set_index('date').plot()
```

Out[103]:

```
<Axes: xlabel='date'>
```

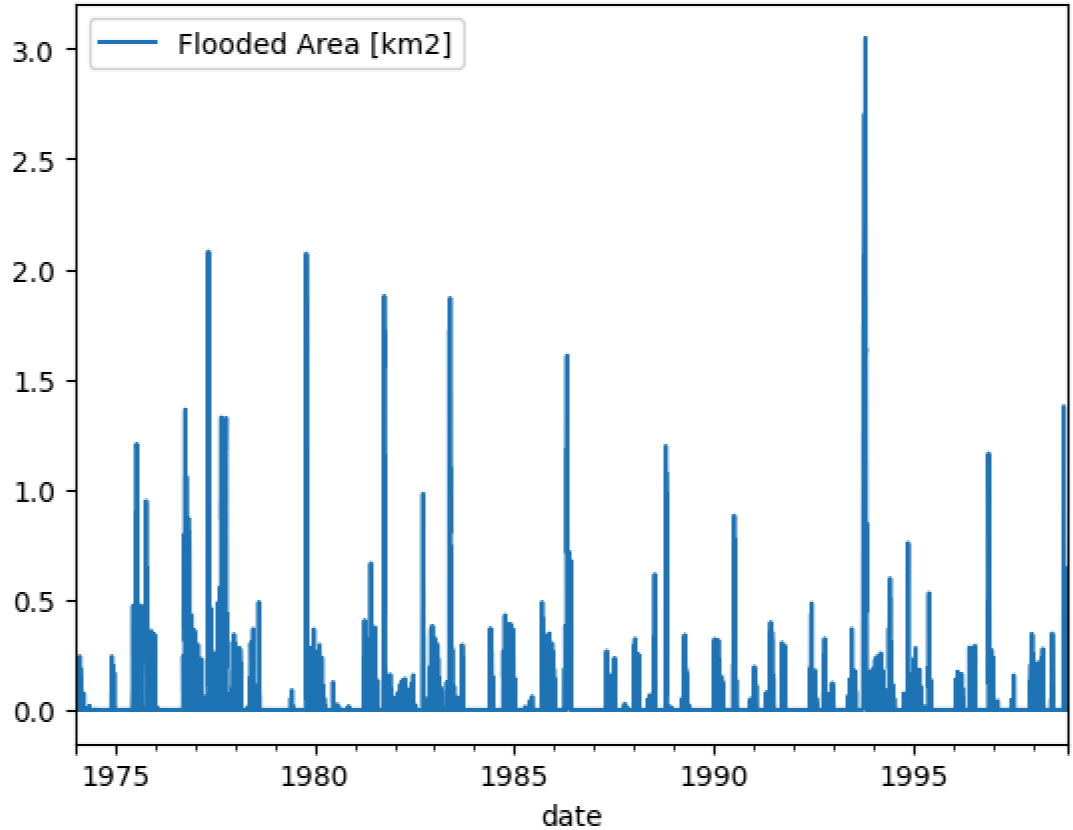


In [104…

```
# PLOT...
df01.set_index('date').plot()
```

Out[104]:

```
<Axes: xlabel='date'>
```



OBJECTIVE FUNCTION:

The objective is to find the maximum local flooded area at time t, S_t^{loc}, as a function of the lake water level height recorded in the hydraulically downgrade town of Sesto Calende h_t^{SC}.

with,

$i = \text{indicator}$
$FL = \text{flood}$
$U = \text{upstream}$
$S1 = \text{indicator 1 for flooded area } (S)$
$loc = \text{locality (Locarno)}$
$SC = \text{Sesto Calende}$
$h = \text{water level height (Sesto Calende)}$

The Objective Function is given as:

$$i_{FL\_U\_S1\_Loc} = \max_{t \in H} \left[S_t^{Loc} \left(h_t^{SC} \right) \right]$$

METHODS:

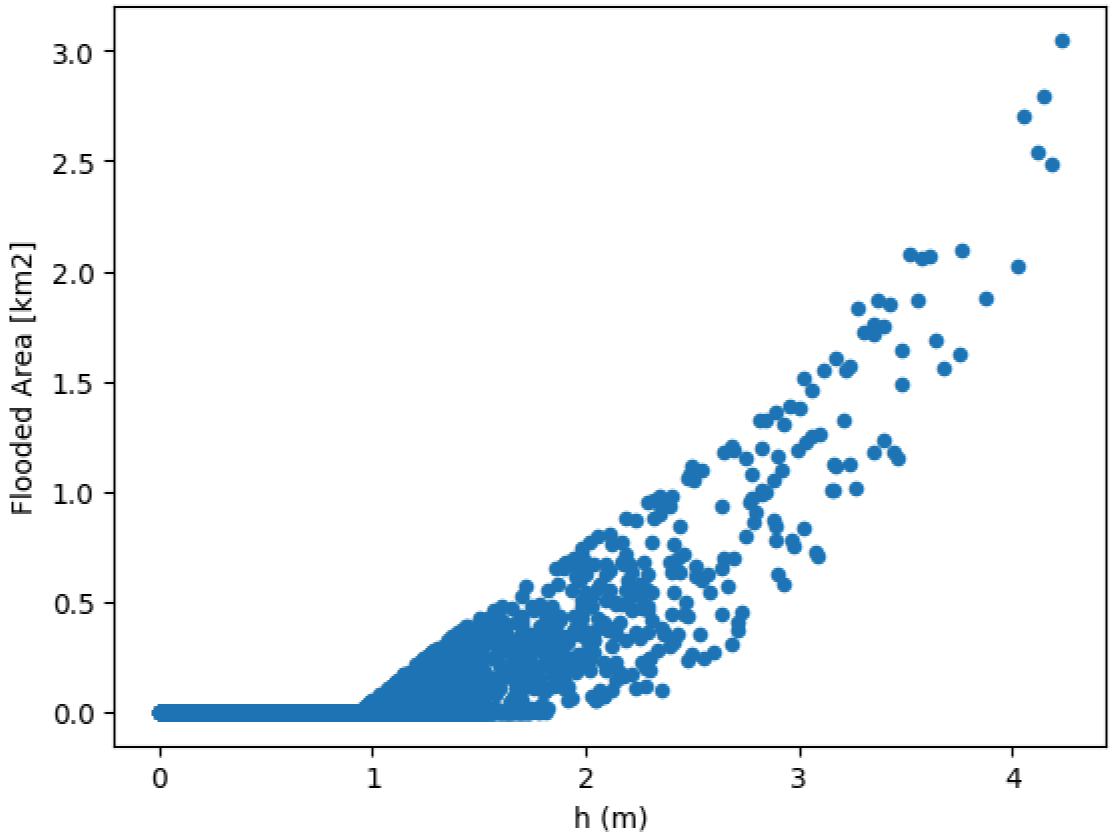
Creating a scatterplot of Flooded Area, $S[\mathrm{km}^2]$, as a function of lake water level height, $h[\mathrm{m}]$...

In [105…

```
# PLOT...
df.plot.scatter(x='h (m)', y='Flooded Area [km2]')
```

Out[105]:

```
<Axes: xlabel='h (m)', ylabel='Flooded Area [km2]'>
```



computing 3rd order polynomial (cubic) regression model...

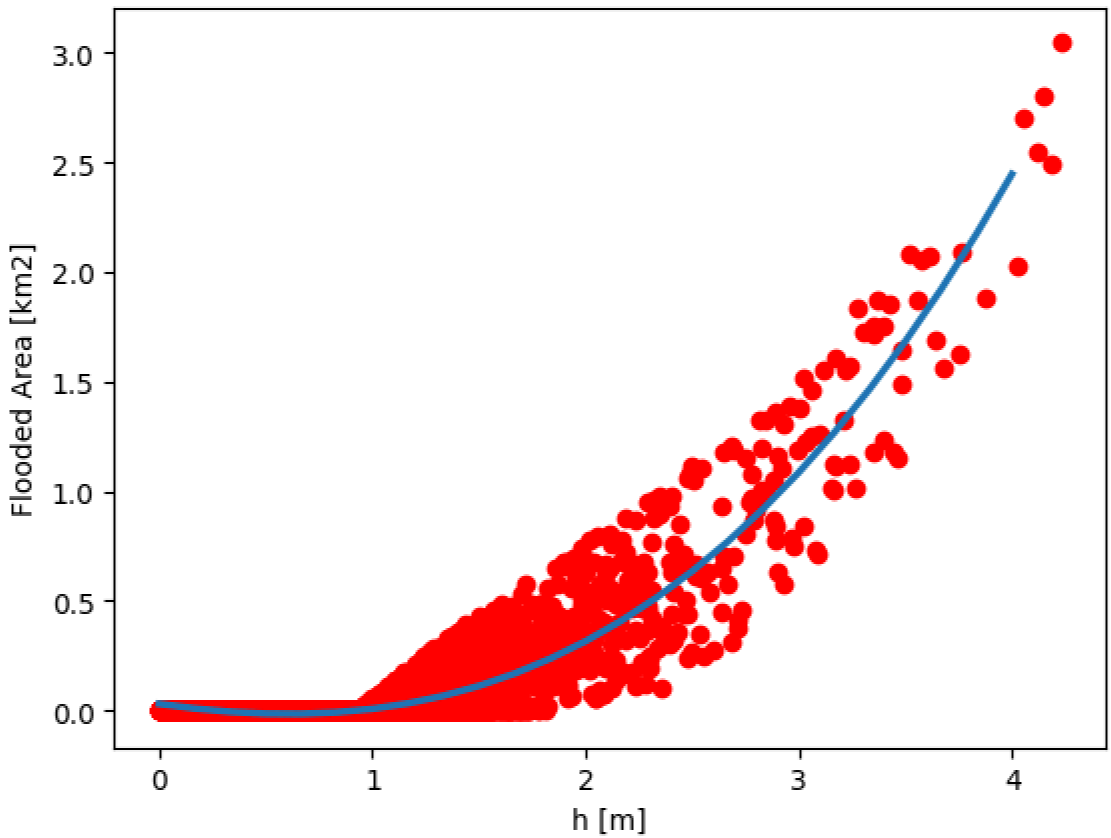
In [106…

```
# fit cubic regression model
model = np.poly1d(np.polyfit(df['h (m)'].values, df['Flooded Area [km2]'].values, 3))

# add fitted cubic regression line to scatterplot
polyline = np.linspace(0, 4, 25)
plt.scatter(df['h (m)'].values, df['Flooded Area [km2]'].values, color='red')
plt.plot(polyline, model(polyline), linewidth=2.5)

# add axis labels
plt.xlabel('h [m]')
plt.ylabel('Flooded Area [km2]')

# display plot
plt.show()
print('S(x = h [m]) = \n\n', model)
```




```
S(x = h [m]) =

           3          2
0.02162 x + 0.101 x - 0.1458 x + 0.02963
```

We notice in the plot and deduce intuitively that the model should reflect the discontinuity in the dataset - a piecewise function - and should give a value of $S = 0$ for all Flooded Area values below a certain lake water level height threshold at or about $1\ \mathrm{m}$. In otherwords, our regression model should ignore any rows where Flooded Area, S is less than or equal to 0, which will help us identify a minimum threshold for h values, below which downstream flooding is not observed...

In [107…

```
# drop rows WHERE Flooded Area, S, is less than or equal to 0
df_mod = df.drop(df[df['Flooded Area [km2]'] <= 0].index)
print('df_mod length:', len(df_mod))
df_mod.head()
```

```
df_mod length: 1078
```

Out[107]:

	h (m)	date	Flooded Area [km2]
12	1.02	1974-01-13	0.061755
18	1.02	1974-01-19	0.044554
31	1.08	1974-02-01	0.089962
40	1.40	1974-02-10	0.175797
41	1.38	1974-02-11	0.164427

This reduces our dataframe from `9070` rows to `1078` rows, which removes `7992` rows (88% of the data) which do not contribute to the regression model's accuracy. Next, we find the min and max (bounds) of the heights, h, for the remaining rows filtered such that $S > 0$...

In [108...

```
# sort df by flooded area to find lake height, h,
# threshold based on minimum recorded flooded area.
df_mod.sort_values(by=['Flooded Area [km2]'])
h_min = df_mod['h (m)'].min()
h_max = df_mod['h (m)'].max()
print('threshold, height, h [m]: ', h_min)
df_mod.head()
```

```
threshold, height, h [m]:  0.95
```

Out[108]:

	h (m)	date	Flooded Area [km2]
12	1.02	1974-01-13	0.061755
18	1.02	1974-01-19	0.044554
31	1.08	1974-02-01	0.089962
40	1.40	1974-02-10	0.175797
41	1.38	1974-02-11	0.164427

Re-computing 3rd order polynomial (cubic) regression model with adjusted bounds...

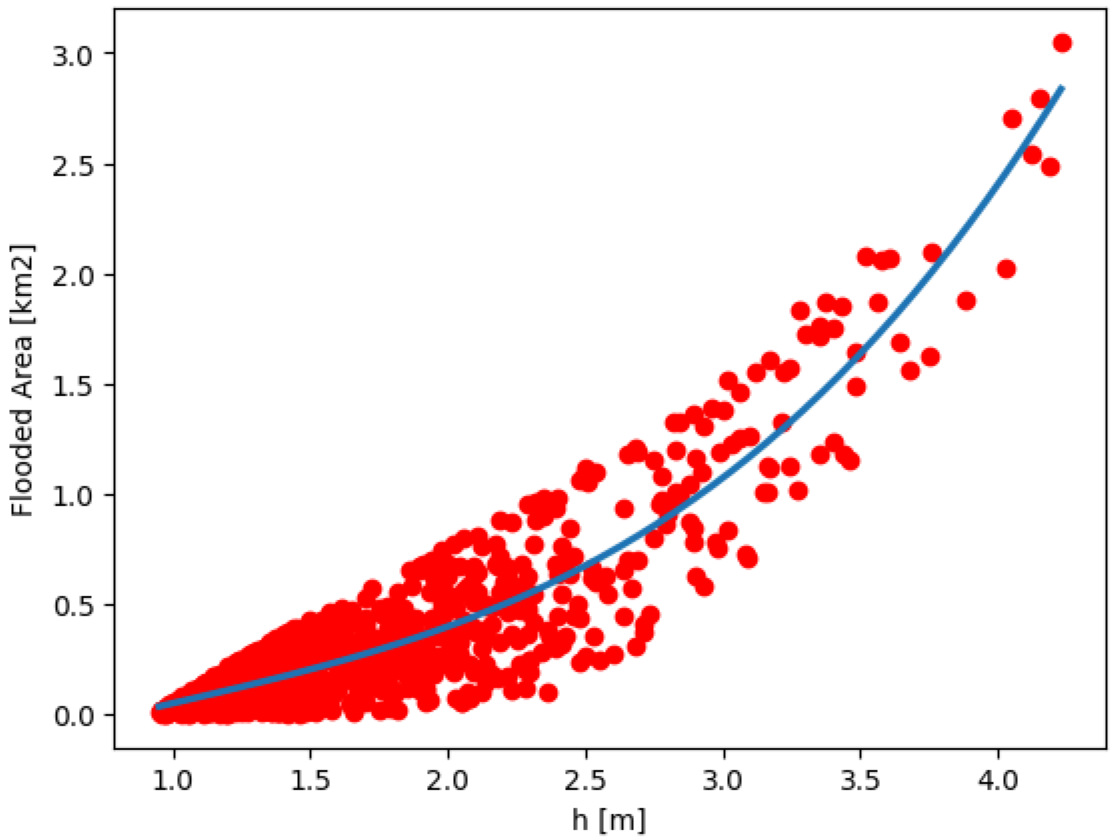
In [109...

```
# fit cubic regression model
model = np.poly1d(np.polyfit(df_mod['h (m)'].values, df_mod['Flooded Area [km2]'].valu

# add fitted cubic regression line to scatterplot
polyline = np.linspace(h_min, h_max, 50)
plt.scatter(df_mod['h (m)'].values, df_mod['Flooded Area [km2]'].values, color='red',
plt.plot(polyline, model(polyline), linewidth=2.5)

# add axis labels
plt.xlabel('h [m]')
plt.ylabel('Flooded Area [km2]')

# display plot
plt.show()
print('S(x = h [m]) = \n\n', model)
```




```
S(x = h [m]) =

           3          2
0.05486 x - 0.1658 x + 0.4616 x - 0.3023
```

With our adjusted regression model, we can now find predicted max flooded area for three alternatives that would result from the following lake water levels:

h1 = 0.75m

h2 = 2.0m

h3 = 4.0m

Using the modeled cubic regression function, we can compute the theoretical predicted flooded area for different upstream lake water level alternatives, $S_t^{loc}(h_t^{SC})$, as...

In [110…

```
def S_model(h_loc):
    if (h_loc <= 0.95):
        return 0
    else:
        return model(h_loc)

print(float("{:.2f}".format(S_model(0.75))),
      float("{:.2f}".format(S_model(2))),
      float("{:.2f}".format(S_model(4))))
```

```
0.0 0.4 2.4
```

DISCUSSION:

Which alternative would be the one preferred by each of the following stakeholders? Explain your answers.

• Environment activists... ***Environmental activists would most likely prefer whichever alternative helps maintain stable water resources for all environs in connection with the system(s) of interest.***

• Lake cruise company... ***A lake cruise company would most likely prefer alternative 3 in order to maximize water depth in the lake for increased sub-vessel clearance for navigation and safety.***

• Farmers' consortium... ***The Farmers' consortium would most likely prefer alternative 3 in order to ensure maximum water availability as a reservoir for potential drought condition periods.***

• Lake tour operators... ***Lake tour operators would most likely prefer alternative 1 to minimize local flooding in and about the lake which could deter lake visitors, and maintain safety in lakeside localities.***

• Fishermen... ***Fishermen would most likely prefer alternative 1, which would be closest to maintaining de facto lake water level and conditions***

• Lakeside population... ***The population of lakeside localities would most likely prefer alternative 1 in order to minimize local flooding in and about the lake which could deter visitors and have negative effects on local economy, and in order to maintain safety from lake flooding.***

• Riverside population... ***The population of riverside localities downstream of the lake would most likely prefer alternative 1 in order to minimize local flooding in and about the river, which could deter visitors and have negative effects on local economy, and in order to maintain safety from local riverine flooding.***

CONCLUSION:

The analysis presented uses recorded water level heights, h, and flooded area, S, for hydraulically connected towns of Locarno (upgrade) and Sesto Calende (downgrade) to model the relationship between the two with h as an indicator for S... $S(h)$. The resulting model is a piecewise function with a threshold of $h = 0.95$ m, where the model predicts a flooded area of $S = 0$ for any h values below this threshold, and computes a S value for h values equal to or greater than the threshold using on a third order polynomial function obtained using cubic regression.

The model solely accounts for and computes predictive values based on the best fit regression line. Further implementation should included additional functions which account for confidence intervals about the best fit line in order to compute potential *maximum* flooded area for h values in addition to a most likely value.