EECS 1011: Lab F

Accessing and Analyzing Hydrometric Engineering Data Online

Overview of the activity

There are five steps in the lab, summarized in the figure below.

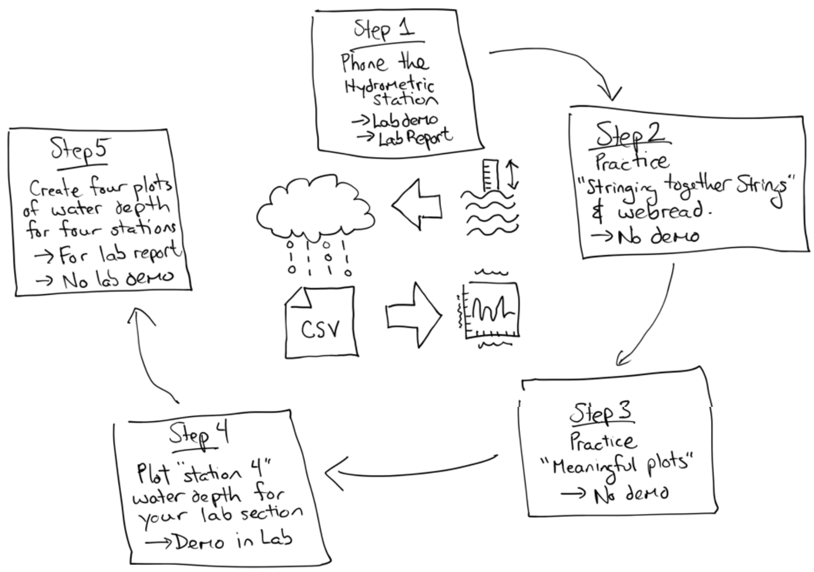


Figure 1 Summary of tasks for the lab.

Learning goals

Important concepts covered in this lab include

- Reading data files and extracting strings within it,
- concatenating strings,
- creating and labelling plots from these files, and
- modifying the data to deal with data anomalies (e.g. "not a number")

It is expected, by the end of this lab, that students can

- successfully collect and interpret data in CSV format from an open and remote website, and
- use a loop to create figures with multiple graphs to illustrate the data from that website.

Matlab is a key element in this activity as it is used to both obtain the remote data and to visualize it.

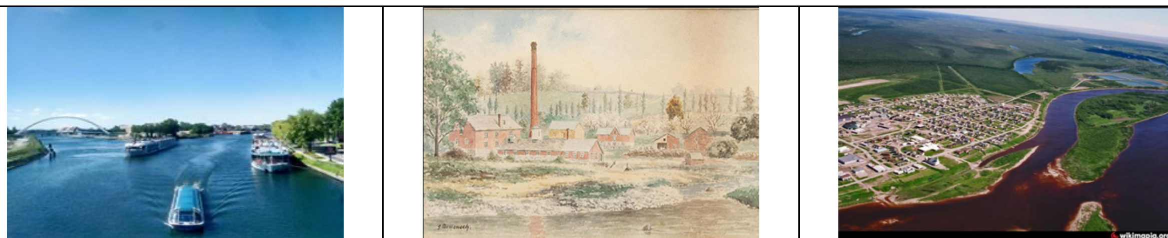
Students demonstrate alternative ways to obtain data elements (by telephone). They also continue to develop basic lab report writing skills in this lab.

Introduction

Climate change has increased the severity of weather and is having a noticeable impact on the bodies of water in and around cities, including Toronto, as well as European cities such as Strasbourg, on the border between France and Germany (where James did his sabbatical in 2018/19; https://bit.ly/3ouiwRu). To better understand the general trends, but also to make decisions on how to regulate the flow of water (in and around our cities, engineers and city officials measure the height and flow ("streamflow") of our rivers using hydrometric sensors such as the ones shown here: https://bit.ly/2Qjn5iY

Figure 2 York's Professor Usman Khan showing us a weir, which can be used to examine streamflow. (c/o Prof. Khan and CIVL 3220 class notes). More on his research here: https://bit.ly/2O7KS2v

Engineers like the Lassonde School's Professor Usman Khan use hydrometric data to examine the state of our watersheds. The information allows informs us about how water flows in and around first nations, cities, towns and rural areas so that we can make engineering and public policy decisions, like where to build dams, as well as how we should attempt to control water flow in our lakes and rivers. (https://bit.ly/2O7KS2v)



The Rhine River on the edge of Strasbourg, France[1]	Toronto's Don River @ Todmorden (historical painting)[2]	Attawapiskat First Nation[3]

Figure 3 Most cities and towns are built on bodies of water; for example, Strasbourg, Toronto and Attawapiskat First Nation.

EECS 1011 is about "computational thinking", using mechatronics as a platform to explore the concept. Mechatronics are about combining sensors and actuators in electro-mechanical systems – to make things move in response to external stimuli that are measured using sensors. The sensors used in Hydrometric applications are typical of the kinds of sensors found in many mechatronic applications, using SONAR, RADAR or encoders to measure depth, distance and flow. The output of these sensors needs to be processed and analyzed prior to reacting with actuators like the ones found in the dams or lock of our waterways.

Engineers need to monitor the water levels, sometimes on a daily or hourly basis. This can be done by

- Visiting the water gauge
- Phoning for water gauge information
- Using the internet to monitor live data

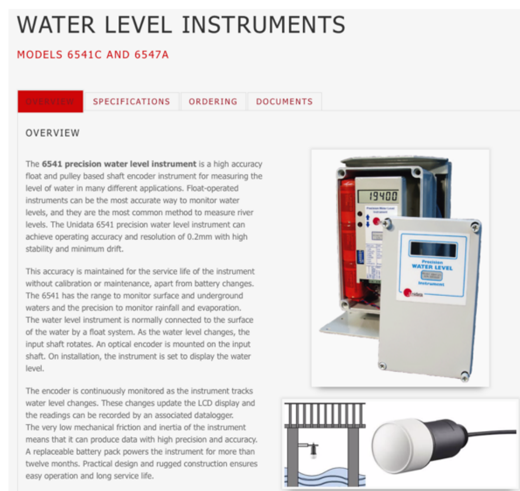


Figure 4 A example of an industrial water depth sensor (c/o https://www.unidata.com.au/products/water-monitoring-modules/precision-water-level-instrument/)

In this lab we will explore how to access some of these water monitoring stations remotely, by telephone and internet. Before we do that we have to go over some important MATLAB programming concepts and functions.

[1] Rhine River photo: James Andrew Smith (http://drsmith.blog.yorku.ca/)
[2] Don River @ Todmorden historical painting (http://citiesintime.ca/toronto/story/remnants-tor/)
[3] Attawapiskat First nation photo c/o Wikimapia (http://photos.wikimapia.org/p/00/02/11/09/06_full.jpeg)

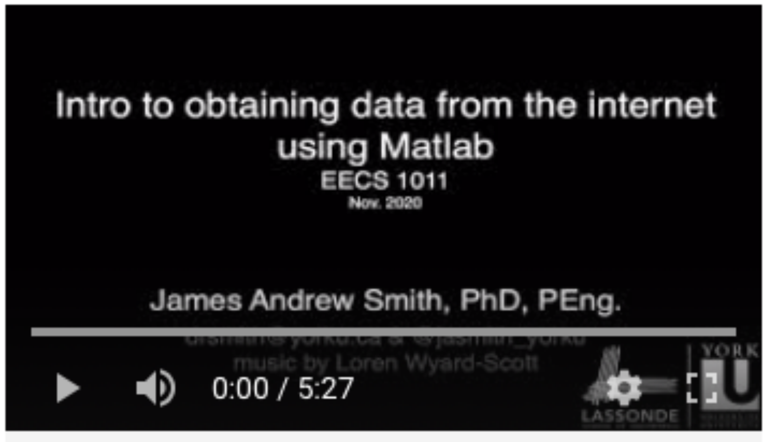


Figure 5 Intro to working with internet data sources using Matlab. [*https://youtu.be/tW95oj4W6R0*](https://youtu.be/tW95oj4W6R0)

York University acknowledges its presence on the traditional territory of many Indigenous Nations. The area known as Tkaronto has been care taken by the Anishinabek Nation, the Haudenosaunee Confederacy, the Huron-Wendat, and the Métis. It is now home to many Indigenous Peoples. We acknowledge the current treaty holders, the Mississaugas of the Credit First Nation. This territory is subject of the Dish With One Spoon Wampum Belt Covenant, an agreement to peaceably share and care for the Great Lakes region.

This lab activity includes an examination of water resources from a number of locations around Ontario, including the Attawapiskat First Nation, which you may have heard about in the news. You are encouraged to further explore online resources and news sites to inform yourself about the issues involving the First Nations.

Stringing Together Strings

In this lab you'll need to combine string values together. For instance, to read the data from a particular water level station in British Columbia at a daily frequency you need to combine different string (character array) variables together into a single string using "string concatenate" or **strcat**():

Matlab code

```
% Example for "web reading" a CSV file, but by breaking up the address
% into separate components so that you can make a loop that scans different
% stations, not just one.
%
% Normally, we would just do this once on a particular data set:
% % e.g. Read the water gauge data at BC station 07EC003:
% gauge_data = webread('https://dd.weather.gc.ca/hydrometric/csv/BC/daily/BC_07EC003_daily_hydrometric.csv')

base_url = 'dd.weather.gc.ca/hydrometric/csv/'
province = 'BC';
frequency = 'daily';
file_type = 'csv';
station_id = '07EA004';  % Station list is here: https://bit.ly/2qS3dJf

% use the string concatenation function to combine the different strings
my_url = strcat('https://', base_url, province, '/', frequency, '/', ...
                province, '_', station_id, ...
                '_', frequency, '_hydrometric.', file_type)

% note the use of the three periods (…) above.
% They tell Matlab that the command continues on the next line.

% use webread() to get the data found at the URL
gauge_data = webread(my_url);

% examine the gauge data variable in matlab.  The data
% is arranged in columns, with variable names at the top
% of each column.  The depth data can be called up using the "." that
% represents data in a structure. That column is called
% WaterLevel_NiveauD_eau_m_
% Here is an example of putting
% that data into the plot function:

plot(gauge_data.WaterLevel_NiveauD_eau_m_);
title("Water Level vs. sample num."); ylabel("[m]"); xlabel("Sample num.");
```

Making Meaningful Plots

Creating plots that communicate meaning clearly and effectively is really important. Plots created by engineers can have important effects in both technical and policy domains. Matlab is a really good tool for doing so and learning to use it will help you get your message.

Matlab code

```
% define some arbitrary data
y = [1 4 5 5 3 3 4 6 7 3  5 ...
    6 3 2 1 1 1 1 4 5 6 4 3 2 ];
x = 1:1:length(y);

% calculate avg & standard deviation
avg_y_scalar = mean(y);
avg_y_vector = avg_y_scalar * ...
    ones(1,length(y));
std_y = std(y);

% data + 1 std deviation (above)
y_avg_plus = avg_y_vector + std_y;
% data - 1 std deviation (below)
y_avg_minus = avg_y_vector - std_y;

% create a vector of the avg-std_dev that
are in reverse order
j=1;
for i = length(y):-1:1
    y_avg_minus_reverse(j) =
y_avg_minus(i);
    j=j+1;     % increment j positive
end

% create plot
figure(4)

% plot the data and average plot(x,y,'b-
>',x,avg_y_vector);
 plot(x,y,'b->',x,avg_y_vector);

 patch([1:1:length(y) ...
    length(y):-1:1],...
    [y_avg_plus y_avg_minus_reverse],...
    'b',...
    'facealpha',0.05,...% fill colour
    'edgecolor','r',...
    'edgealpha',0.05) % edge colour

% Legend, Title, axis labels.
legend('original data',...
    'average values',...
    'standard deviation')
title('Data (avg & std dev)')
xlabel('Sample Number [no units]');
ylabel('Magnitude [units: Quanta]');
```

Graph

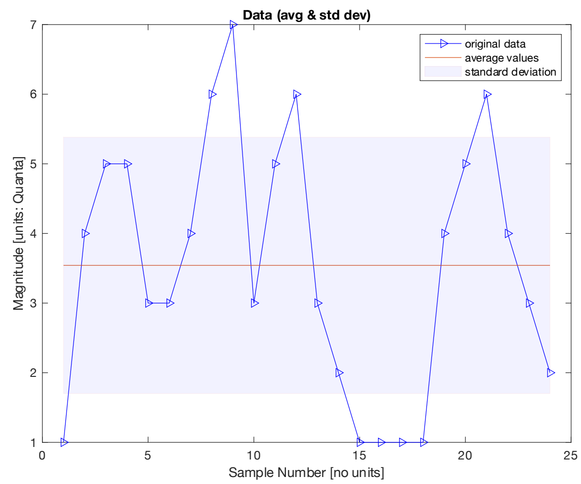


Figure 6 Graph showing data, average, and standard deviation (via patch & alpha channel transparency)

Part 1: Phone the Water Level Sensor

Use your phone to obtain water level at a variety of locations around Canada using the the Canadian Hydrographic service and the waterlevels.gc.ca "Bulletin" site (https://waterlevels.gc.ca/eng/info/bulletin):

1. Tuesday 8:30am (last name A-K) : Water level @ St. Lawrence River, above the lock at Iroquois:
    a. (613) 652-4426
2. Tuesday Noon (last name L-Z): Section 2: Water level @ St. Lawrence River, below the lock at Iroquois:
    a. (613) 652-4839
3. Tuesday Noon (last name A-K): Water level @ Lake Huron at Tobermory
    a. (519) 596-2085
4. Tuesday Noon (last name L-Z): Water level @ Lake Ontario at Port Weller
    a. (905) 646-9568
5. Tuesday 3pm (last name A-K): Water level @ Lake Ontario at Burlington
    a. (905) 544-5610
6. Tuesday 3pm (last name L-Z): Water level @ Sault Ste. Marie, above the lock
    a. (705) 949-2066
7. Thursday 8:30am (last name A-K): Water level @ Sault Ste. Marie, below the lock
    a. (705) 254-7989
8. Thursday 8:30am (last name L-Z): Water level @ St. Lawrence River at Kingston
    a. (613) 544-9264
9. Thursday Noon (last name A-K): Water level @ the Detroit River at Amherstburg
    a. (519) 736-4357
10. Thursday Noon (last name L-Z): Water level @ St. Lawrence River at Cornwall
    a. (613) 930-9373
11. Thursday 3pm (last name A-K): Water level @ Lake Superior at Thunder Bay
    a. (807) 344-3141
12. Thursday 3pm (last name L-Z): Water level @ Lake Superior at Gros Cap
    a. (705) 779-2052

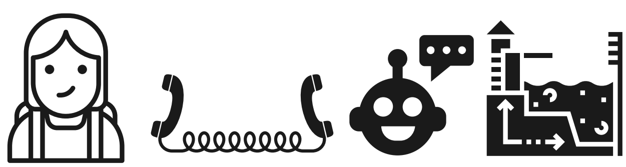


Figure 7 use your telephone to obtain the water level at an automated hydrometric station.[4]

Phone the particular location and [for the lab report] **write down (a) the current depth of the water, (b) the date and (c) the time of day** of the water in your lab report, right under the abstract.

Lab Demo 1: During the lab, state the water depth value to the TA when prompted. There is some variability to the value (typically within one standard deviation or less).

Note that your classmates may be *phoning at the same time*. **If it's busy**, wait and try again.

[4] Icons courtesy of the Noun Project.

Part 2: Getting Water Depth Information via Online Datasets.

Obtain the daily water depth for four different gauges, as listed below. Use the **webread**() function to get the data, stored on a Canadian government server as a "CSV" file. Assume that each new data point represents a new daily value.

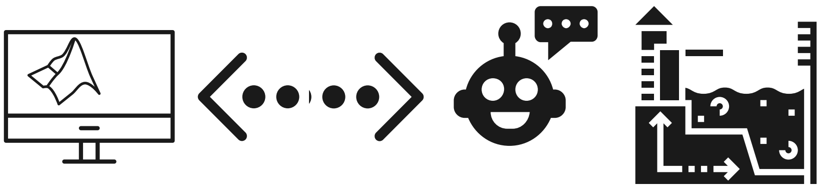


Figure 8 Use your computer and Matlab to obtain the water datasets at an automated hydrometric station.

Section	**Station 1**	**Station 2**	**Station 3**	**Station 4**
Tuesday 8:30am	"DON RIVER AT TODMORDEN" (02HC024)	"BLACK CREEK NEAR WESTON" (02HC027)	"ATTAWAPISKAT RIVER BELOW ATTAWAPISKAT LAKE" (04FB001)	"ATTAWAPISKAT RIVER BELOW MUKETEI RIVER" (04FC001)
Tuesday Noon	"HUMBER RIVER AT ELDER MILLS" (02HC025)	"DON RIVER AT TODMORDEN" (02HC024)	"ATTAWAPISKAT RIVER BELOW ATTAWAPISKAT LAKE" (04FB001)	"ATTAWAPISKAT RIVER ABOVE LAWASHI CHANNEL" (04FC002)
Tuesday 3pm	"LITTLE ROUGE CREEK NEAR LOCUST HILL" (02HC028)	"WINDIGO RIVER ABOVE MUSKRAT DAM LAKE" (04CB001)	"DON RIVER AT TODMORDEN" (02HC024)	"ATTAWAPISKAT RIVER BELOW MUKETEI RIVER" (04FC001)
Thursday 8:30am	"ETOBICOKE CREEK BELOW QUEEN ELIZABETH HIGHWAY" (02HC030)	"LITTLE ROUGE CREEK NEAR LOCUST HILL" (02HC028)	"HUMBER RIVER AT ELDER MILLS" (02HC025)	"DON RIVER AT TODMORDEN" (02HC024)
Thursday Noon	"REDHILL CREEK AT HAMILTON" (02HA014)	"ETOBICOKE CREEK BELOW QUEEN ELIZABETH HIGHWAY" (02HC030)	"LITTLE ROUGE CREEK NEAR LOCUST HILL" (02HC028)	"BLACK CREEK NEAR WESTON" (02HC027)
Thursday 3pm	"FRENCH RIVER AT PORTAGE DAM",(02DD016)	"REDHILL CREEK AT HAMILTON" (02HA014)	"ETOBICOKE CREEK BELOW QUEEN ELIZABETH HIGHWAY" (02HC030)	"DON RIVER AT TODMORDEN" (02HC024)

The **names** of stations and their **IDs** can be found here:
https://dd.weather.gc.ca/hydrometric/doc/hydrometric_StationList.csv

The **datasets** are found here, identified by province, daily or hourly data frequency and their station ID:
https://dd.weather.gc.ca/hydrometric/csv/

You'll need to **find the average value** of the daily depth value (use the **daily** sets, *not* the hourly sets) in the Ontario folder. However, many sensors have *bad data* in them, represented by a **NaN** (not a number) value in the dataset. "Bad data" is a reality of real-world sensing and engineers need to be able to handle that possibility.

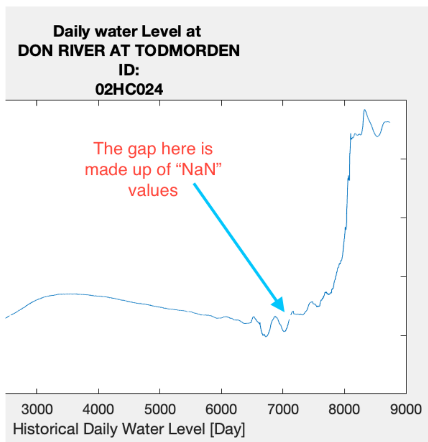


Figure 9 Some of the depth data is "bad" and is represented by NaN (not a number). This could be due to the sensor being faulty or turned off for a period of time.

You'll need to remove the "NaN" instances in the dataset before performing operations like mean() or std(). For example, if my variable is called

my_var

and my_var has a few NaN values in it and I need to remove them, I can update my_var in Matlab as follows:

```
my_var(find(isnan(my_var)))=[]
```

Recall from an earlier example that to get the water depth values from one of the remote sensors you do the following (here, for a sensor in BC):

```
my_url = 'https://dd.weather.gc.ca/hydrometric/csv/BC/daily/BC_07EC003_daily_hydrometric.csv';
gauge_data = webread(my_url);                          % all the data
depth_data = gauge_data.WaterLevel_NiveauD_eau_m_;   % just the depth values[5]
```

Next, process **depth_data** to remove any NaN values before running the "mean" or any other function on it.

[5] Notice how gauge_data is followed by a period and then something about "WaterLevel". The gauge_data variable is a structure variable that contains other variables *within* it, including names and numbers. The `WaterLevel_NiveauD_eau_m_` is a component *inside* of gauge_data that contains the numeric depth values.

Lab Demo 2: During the lab, demonstrate to your TA that you can graph the "Daily water level" at Station 4 for your lab section in Table 1. You only need **the daily value**… no average or standard deviation values required for the demo. **Make sure to put** titles, as well as x- and y-axis labels. These are **required** for full marks.

For your lab report, calculate the standard deviation and **plot** against the original data and the average for four different water depth stations, as shown below.

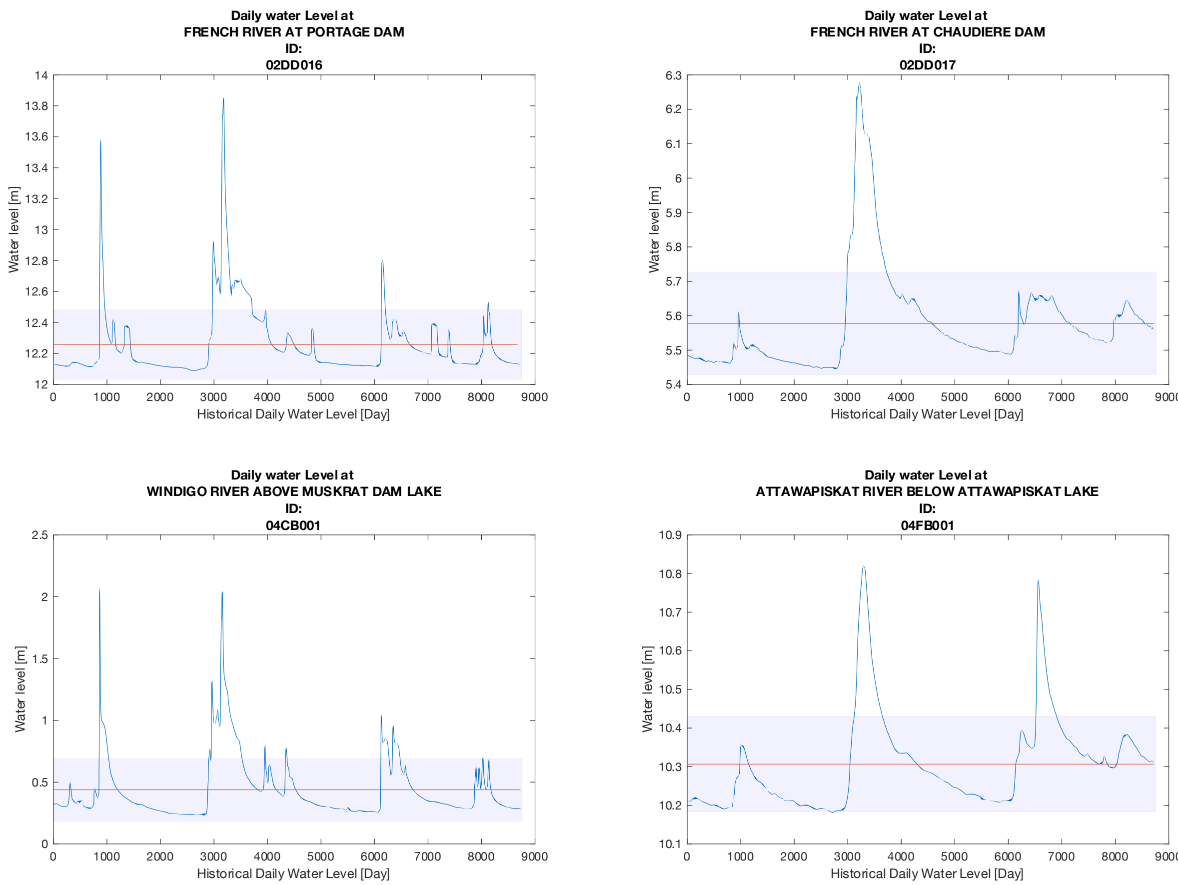


Figure 10 Example of plotting data, average and standard deviations at ***four <u>different</u> physical station locations.***

Add the four plots to your lab report. Each component of the graphs (the original data, the average and the standard deviation) is graded separately, as per the marking guide. See the marking guide for what components need to be included in the graph.

There are different ways that you can write your MATLAB script to create these graphs. **One way** is to make four separate calls to gathering the data from the online source. That's effective but not very efficient. **A better way** is to write a **single loop** that iterates through each water monitor station, one at a time, creating a plot for that particular station at the end of the particular iteration.

1. Make a list of the stations that you need to look at.
2. Enter the loop
3. Get the data for one station
4. Extract the depth data
5. Process the depth data (remove NaN, find average, find std deviation)
6. Create plot (either new figure or use subplot())
7. Unless you've done all four graphs, go to step 3.

Lab report

For this lab you need to produce a lab report. The **cover page** has the lab name, course name, your name, as well as your ID.

At the top of the **second page** write in the **value** you obtained for Part 1 (by **telephone**).

On the rest of the **second page** is your collection of four graphs, as illustrated earlier. You may use the subplot() command to generate the four graphs or you can create four different figures and paste them in separately.

Submit two files:

1. Lab report (.pdf), with cover sheet, depth value (by phone) and four graphs.
2. your MATLAB source code (.m) for the four graphs.

to eClass **by the Sunday after the lab @ 11:55pm.**

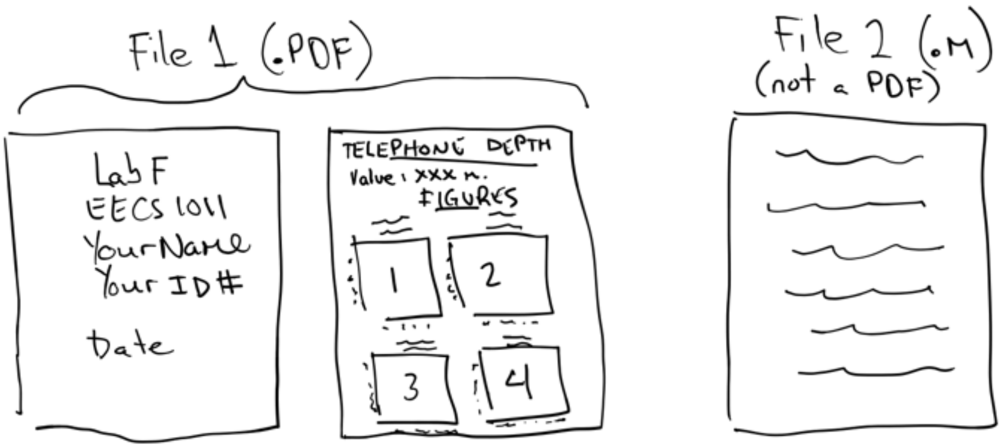


Figure 11 Summary of file submission for the report. Submit two files: a two-page PDF with a cover page and figures, as well as a text file of your Matlab script (saved as YourName.m or similar)

The marking guide for the lab report is found on the last page of this document.

Marking Guides

Demonstration Marking Guide

The demonstration is graded out of 1, just like all other assessments.[6]

Table 1 Marking guide for the lab demonstrations

Demonstration	Description	Maximum Mark	Notes
Demo 1	Oral statement of current water depth value determined from telephone contact with a specific station.	0.5	0.5 if the value is an exact match or within 20% of instructor's value; 0.25 if a value is given but it's not correct; 0 otherwise
Demo 2	**One** graph which shows **daily depth** history for a specific station.	0.5	0.5 if similar to (within tolerances) instructor graph, 0.4 if minor mistakes, 0.3 if many / major mistakes, 0.2 if attempted but isn't correct. 0 otherwise

For full marks, the graph must not contain any NaN anomalies, it must have an accurate title, as well as labels on the x- and y-axes (see example).

Note: instructors and teaching assistants need to capture the values on the day of the lab as the values do change over time.

Lab Report Marking Guide

The lab report is graded out of 1, just like all other assessments.

Table 2 Marking guide for the lab report.

Section	Item & Description	Maximum Mark	Notes
Title Page	Name, ID	None.	No grade w/o name and ID.
Part 1	Depth level via phone	0.2	All or nothing
Part 2(a)	**Four** graphs where each shows **daily depth** history a different station	0.2	0.2 if similar to (within tolerances) instructor graph, 0.15 if minor mistakes, 0.10 if many / major mistakes, 0.05 if attempted but isn't correct. 0 otherwise
Part 2(b)	Each of the four graphs shows **average depth** for *each* station	0.2	0.2 if similar to (within tolerances) instructor graph, 0.15 if minor mistakes, 0.10 if many / major mistakes, 0.05 if attempted but isn't correct. 0 otherwise
Part 2(c)	Each of the four graphs shows **standard deviation** of depth for a *each* station	0.2	0.2 if similar to (within tolerances) instructor graph, 0.15 if minor mistakes, 0.10 if many / major mistakes, 0.05 if attempted but isn't correct. 0 otherwise
Part 3	MATLAB **code** shows a single loop and each iteration obtains data and plots data for a different water station	0.2	0.2 if code components found as required, 0.15 if one major element missing, 0.10 if many / major mistakes or elements missing, 0.05 if attempted but isn't correct. 0 otherwise

For full marks, the graphs must not contain any NaN anomalies, it must have an accurate title, as well as labels on x-axis, y-axis (see example). There should be three entries in each legend (the original time series, average value and standard deviation).

When examining the source code, we check for a for loop (or indexed while loop), subplot calls, a call to legend and commands for x-axis, y-axis and title labels.

[6] Grading out of 1 is based on a specifications-based grading approach that "binned" certain categories of assessments together; see the work outlined in https://ojs.library.queensu.ca/index.php/PCEEA/article/view/14885.