**Team Name:**

Delhi

**Group Members & Roles:**

Sara Adi (1129361),

Jennifer Lithgow (1134108), Sophie Mlodzik (1130462), Emily Kozatchiner (1149665)

| **Position** | **Primary** | **Secondary** |
| --- | --- | --- |
| **Product Owner** | Jen/Emily | Sara |
| **Team Leader/Project Manager**  **& Process Coordinator/Agile Lead** | Sara | Sophie |
| **Architect/Conceptual Interaction Lead** | Emily | Jen |
| **Communications Officer/Documentation Lead & Archivist/Revision Control Lead** | Sophie | Jen |
| **Quality Assurance/Testability Lead** | Jen | Emily |

Table of Contents

[**Section 1.0: Sara Adi**](#_cvtp08foe56) **2**

[Fields & Required Specific Values](#_tth2g9cubjm9) 2

[Preprocessing For Construction of Interactive Programs Data Sets](#_pqzciv3ikoau) 2

[Data File Organization](#_tpe58mvaqz13) 3

[How the Selection Of (Time) Periods Occur](#_pqzciv3ikoau) 3

[Incorporation of New Data](#_pqzciv3ikoau) 4

[**Section 2.0: Jennifer Lithgow**](#_f9pzipji74zt) **5**

[Fields & Required Specific Values](#_7x9aj3loex3g) 5

[Preprocessing for Construction of Interactive Programs Data Sets](#_jb1vhfzaeu5v) 5

[Data File Organization](#_hb97gtw3p1nf) 5

[How the Selection of (Time) Periods Occur](#_9x1bqzf7nwc3) 6

[Incorporation of New Data](#_zcy59zj9bs5s) 6

[**Section 3.0: Emily Kozatchiner**](#_2v3j7rp6px6e) **7**

[Fields & Required Specific Values](#_3uy53blzfab2) 7

[Preprocessing for Construction of Interactive Programs Data Sets](#_3vy1d075eaul) 7

[Data File Organization](#_u9pma0kvigmb) 7

[How the Selection of (Time) Periods Occur](#_3pjl4bhd5u0d) 8

[Incorporation of New Data](#_czsgjrrzd451) 8

[**Section 4.0: Sophie Mlodzik**](#_uej1soe3w8n5) **9**

[Fields & Required Specific Values](#_h0rs9baen0oy) 9

[Preprocessing for Construction of Interactive Programs Data Sets](#_xfu2aww4xvs3) 9

[Data File Organization](#_5dgd08o6e2oh) 9

[How the Selection of (Time) Periods Occur](#_pw6q566qm730) 10

[Incorporation of New Data](#_yxwty1vc6s0m) 10

[**Section 5.0: Delhi Team**](#_k6esrkwsxcd8) **11**

[How will you be organizing your team to look at the data? Who will be responsible for various data and questions?](#_g1s8g6bzj49l) 11

[Sara Adi](#_ib9v1wkd8m3k) 11

[Jennifer Lithgow](#_qhtlwu9czrl0) 11

[Emily Kozatchiner](#_v0idka89kvmj) 11

[Sophie Mlodzik](#_me05x1h2grqz) 11

# Section 1.0: Sara Adi

## Fields & Required Specific Values

To Run The Script:

Required Fields:

1. argv [0] = python script
2. argv [1] = filename with school related data
3. argv [2] = filename with Ontario data
4. argv [3] = specified year of search
5. argv [4] = specified month of search

Specific Values

1. argv [3] requires an integer in the format of a year \*\*\*\*
2. argv [4] requires an integer in the format of a month \*\*

For Preprocessing:

School File:

**Col. 1:** Collected date

**Col. 6:** New total school-related cases

Ontario CSV File:

**Col.6:** “Specimen Date”

## Preprocessing For Construction of Interactive Programs Data Sets

Two (2) data sets are constructed with preprocessing, at run time, for the interactive program (i.e. plotting the double bar graph).

Dataset 1: “y”, type: list

This data set holds five values. One for each week of a month (1-7, 8-14, 9-21, 22-28, 29-31).

Each value is of type int, and holds the total (sum of the) number of COVID-19 cases Ontario has reported within that week. These values are formed from a dictionary created from the original CSV file (a previously preprocessed file). This dictionary (which only holds keys/values for the specified time from the command line) is then filtered through, to compute the sum of the values within the selected ranges of the keys. Each sum (total of 5) is then added to the list.

Dataset 2: “z”, type: list

This dataset holds five values. It is preprocessed in the same order as dataset 1, but taking data from the school’s COVID-19 dataset, creating a similar dictionary that is filtered through in the same fashion as dataset 1.

## Data File Organization

**File 1: data/conposcovidloc.csv**

This is the original CSV data file from the Government of Ontario. It has 18 columns. The columns consist of information on data, age group, gender, Acquisition information, Outcome, PHU, and location of the patient. The file has over 300,000 rows (one per case)

**File 2: data/preprocessed\_ontario2.csv**

This is the pre-processed file from File 1. It contains one column, and as many rows as there were reported COVID-19 cases in Ontario since 2020-01-09. Each column represents a single case. The columns are organized by the row [Specimen\_Date].

**File 3: data/schoolcovidsummary.csv**

This is the original CSV data file from the Government of Ontario. The file consists of 21 columns, and 121 rows, which increase as each update occurs. Each row represents a new report/collected date. This set includes faculty and students, and the total cases per day in Ontario schools.

**File 4: data/preprocessed\_school\_covid.csv**

This is the pre-processed file from File 3. It contains two rows, with each column providing the total number of school cases reported that day. This file is organized by collected\_date and is in the form of [collected\_date, new\_total\_school\_related\_cases].

**File 5/6: preprocessing/prep\_ontario2.py prep\_school.py**

These files used the pandas library and the functions of .drop(), .sort\_values(), to\_datetime(), and to.csv(), to filter through Files 1 and 3 and aggregate and group the data. These files produced files 2 and 4 respectively.

## How the Selection Of (Time) Periods Occur

The user must specify a month and day they would like to analyze through the command line. This information will then be read by the program in order to pull out correct data from the preprocessed files and create the data sets needed for the interactive part of the program. The format of the year must be \*\*\*\* (i.e. 2020), and the format of the month must be \*\* (i.e. 10).

## Incorporation of New Data

The data is manually re-uploaded as a CSV file from the Government of Ontario website to the Repl. Then the preprocessing program is run.

# Section 2.0: Jennifer Lithgow

## Fields & Required Specific Values

To Run the Script:

Required Fields:

1. argv[0] = python script
2. argv[1] = filename with census related data
3. argv[2] = filename with covid infection related data
4. argv[3] = start of age range to calculate (inclusive)
5. argv[4] = end of age range to calculate (exclusive)

Specific Values:

1. argv[3] requires a positive integer for age
2. argv[4] requires a positive integer for age greater than argv[3]

For Preprocessing:

Ontario Census File:

**Col. 1:** Age

**Col. 2:** Population

Ontario CSV File:

**Col. 3:** Accurate episode date

**Col. 7:** Age group

## Preprocessing for Construction of Interactive Programs Data Sets

Two (2) data sets are constructed with preprocessing, at run time, for the interactive program (i.e. plotting the line graph).

Dataset 1: “y”, type: list

This dataset holds two pieces of information relevant to plotting the graph. The datasets combine so that the program can find the total population of the inputted age range from the user. It is collected from data/preprocessed\_census.csv.

Dataset 2: “x”, type: list

This dataset holds two other pieces of information: accurate date infected and age range of the infected person. These are combined to create a population of the age range defined by the user that has been infected on a given date. It is collected from data/preprocessed\_ontario.csv.

## Data File Organization

**File 1: data/10952620210317092505.CSV**

This is the original file containing the census data. It has four (4) columns, sorting population by age and/or gender. There are additional rows that do not contribute to the desired/relevant information. There are 134 rows.

**File 2: data/census\_data\_copy\_edited.csv**

This is a file containing information that was copied from the original census file, but the irrelevant rows were removed to allow for easier preprocessing of the data.

**File 3: data/preprocessed\_census.csv**

This is a filtered and sorted file of the census data, only containing the information relevant to answering the question investigated (that is, populations of age ranges).

**File 4: data/conposcovidloc.csv**

This is the original CSV data file from the Government of Ontario. It has 18 columns. The columns consist of information on data, age group, gender, Acquisition information, Outcome, PHU, and location of the patient. The file has over 300,000 rows (one per case).

**File 5: data/preprocessed\_ontario.csv**

This is the pre-processed file from File 4. It contains a column containing the accurate episode date of COVID-19 cases in Ontario since 2020-01-01 and a column containing the age of the infected person. Each row represents a single case. The rows are sorted by the column [Accurate\_Episode\_Date].

**File 6: prep\_census.py**

This file performed the preprocessing of the census data (file 2), outputting file 3. It used the pandas library and the .drop() function to filter through the columns and .loc[] and .contains() to filter the rows. The information is then aggregated according to the age groups defined in the preprocessed\_ontario.csv file and printed to its respective file.

## How the Selection of (Time) Periods Occur

There is no user selection of the time period; instead, the information is accumulated from all given dates from the source files.

## Incorporation of New Data

Census data is given on a period of approximately five years, which is a longer time frame than the one that this project took place. Thus, there was no opportunity to choose how to incorporate new data.

With infection data, it was incorporated by manually uploading the updated CSV file from the Government of Ontario website to the Repl and running the preprocessing program.

# Section 3.0: Emily Kozatchiner

## Fields & Required Specific Values

To Run the Script:

Required Fields:

1. argv[0] = python script
2. argv[1] = modified csv file holding the PHU data
3. argv[2] = PHU Id parameter

Specific Values:

1. argv[2] requires a positive PHU id that matches a PHU id in the available data

For Preprocessing:

Required Fields:

1. argv[0] = preprocessing python script
2. argv[1] = unmodified csv file (refer below) holding the PHU data

PHU Response CSV File:

**Col. 3:** PHU Id attached to specific PHU

**Col. 4:** PHU status

**Col. 5:** Starting date of newly applied status

## Preprocessing for Construction of Interactive Programs Data Sets

A data set is constructed with preprocessing (throughout the main script), at run time, for the interactive program (i.e. plotting the line graph).

Dataset:

x-value = “dates\_arr”, type: list

y-value = “status\_arr”, type: list

The line graph/plot dataset needs the two axes of information found within the mains script to plot the graph. The two lists contain data collected from the modified csv file (data/prep\_phu\_response.csv), where the first list contains all of the start-dates and the second, all of the statuses for each array. They are stored in order to find the rate of change of PHU status severity per each node in the data.

## Data File Organization

**File 1: data/phu\_response\_framework.csv**

The original csv file containing data on specific PHUs. The data file consists of seven (7) columns, sorted by numerical order from least to greatest PHU id. In total, there are currently 186 total records in the data file, with a multitude of data per PHU (one PHU may not have the same amount of changes in PHU status as another).

**File 2: data/prep\_phu\_response.csv**

A modified version of the “File 1” csv file above. The data file is more concise, only consisting of three(3) necessary columns, excluding unneeded data. The data is rid of intervening strings, columns, and enumerated accordingly to ease the main script’s intake of data.

**File 3: preprocessing/prep\_ph\_df.py data/phu\_response\_framework.csv**

A preprocessing script that is responsible for modifying the original csv data file into the concise “File 2” data file above. Through usage of the pandas library the data file was changed to fit the main script’s intake standards. This file only needs to be run once to retrieve a modified file from the original.

## How the Selection of (Time) Periods Occur

The user never selects a time period - time periods are only given and accessed in the csv files.

Set time periods are taken in by the main script and used in the calculation of the rate of change between all adjacent dates with the help of the datetime library. The date was split into separate values and each integer stored into a separate array (year\_arr, month\_arr, day\_arr) which were then used to complete calculations for the main script.

## Incorporation of New Data

During the span of this project, changes in the CSV file from the Government of Ontario were manually re-uploaded to the project Repl, and a new modified csv file was created with the preprocessing command.

Incorporation of the updated CSV file brought about unexpected changes to the data strings in the file (“Stay-at-home” was changed to “Shutdown”), which the preprocessing program had not accounted for. Changing the preprocessing to adjust to the new data only improved the code to account for such a change.

# Section 4.0: Sophie Mlodzik

## Fields & Required Specific Values

To Run the Script:

Required Fields:

1. argv[0] = name of python script
2. argv[1] = name of csv file containing vaccine-related data
3. argv[2] = name of graphics file
4. argv[3] = number of days into the future

Specific Values:

1. argv[3] requires a positive integer value less than 31

For Preprocessing:

Required Fields:

1. argv[0] = preprocessing python script
2. argv[1] = unmodified csv file (refer to format below) holding vaccine-related data

Vaccine Data CSV File:

**Col. 2:** Report Date

**Col. 3:** Previous Day Doses Administered

## Preprocessing for Construction of Interactive Programs Data Sets

A data set is constructed with preprocessing, at run time, for the interactive program (i.e. plotting the line graph).

Dataset:

x-value = “report\_date”, type: list

y-value = “previous\_day\_doses\_administered”, type: list

The data set for this line graph needs the two axes of information found in the main script to plot the graph. The two lists contain data collected from the preprocessed csv file, “data/preprocessed\_vaccines\_adminstered.csv”. The first list contains the reported dates and the second list contains the number of vaccine doses administered the previous day. This data is stored in order to find the average rate of growth of vaccines administered over the most recent 30 days, which is needed to make a prediction regarding the number of vaccines to be administered a specified number of days in the future.

## Data File Organization

**File 1: data/vaccines\_administered.csv**

This is the unmodified file containing vaccine-related data. It has six (6) columns, sorting the number of administered vaccines by doses administered the previous day, total doses administered, total doses in fully vaccinated individuals, and total individuals fully vaccinated. The most recent version contains 94 rows.

**File 2: data/preprocessed\_vaccines\_administered.csv**

This is the preprocessed file containing the filtered vaccine-related data. It has two (2) columns as it only contains information that is relevant to answering the question investigated. These columns are the report date and the number of doses administered the previous day.

**File 3: preprocessing/prep\_vaccines.py**

This script uses the following functions to sort through and reorganize the data from File 1 into the the relevant data remaining in File 2: .str.replace(), .drop(), and to.csv()

## How the Selection of (Time) Periods Occur

The user selects a number of days into the future in the command line and this is used to calculate the predicted number of vaccines to be administered that many days into the future.

A set time period of the most recent 30 days of data is used to calculate the average rate of change in doses administered daily. The daily rates of change, in comparison to the previous days, are then averaged and used to make predictions.

## Incorporation of New Data

During the span of the project, changes in the CSV file from the Government of Ontario website were manually re-uploaded to the Repl repository to ensure the most recent data was always included. The necessary preprocessing script was then run to remove any information that was irrelevant to the project.

# 

# Section 5.0: Delhi Team

## Organization of Group Members in Terms of Data Responsibilities

Each group member is responsible for one of the four questions. This responsibility includes:

* The formation of the question
* Research into the datasets required for the question
* The main script
* Pre-processing (unless files match, in the case of Sara Adi and Jennifer Lithgow)
* Visualizations through Matlab, pandas, seaborn, or numpy
* Milestone 1, 2 and 3’s specified report section on the question
* Demo days’ explanation and findings on the question

## Sara Adi

Question: What is the relation between the number of confirmed positive Covid-19 cases within Ontario schools, vs. the total number of cases for Ontario as a province over time as a function of year and month?

Responsible for the preprocessing of Ontario (same file as Jennifer) and school data files, preprocessing/prep\_ontario.py and preprocessing/prep\_school.py respectively, alongside individual script to answer questions and plotting (school\_covid.py).

## Jennifer Lithgow

Question: What is the percentage of a chosen Ontarian age population that has been diagnosed with Covid-19 since the beginning of the pandemic?

Responsible for the preprocessing of the Ontario census file, preprocessing/prep\_census.py and the individual script to answer questions and plotting (age\_percentage.py).

## Emily Kozatchiner

Question: What is the rate of change for a PHU status throughout a period of time? Is it a rise or decline in extremity?

Responsible for the preprocessing of the PHU response framework file (preprocessing/prep\_phu\_df.py) and the coding of the main script to answer the specific question chosen with plotting in the main script (phu\_status.py).

## Sophie Mlodzik

Question: Given a growth rate of vaccines administered daily, calculated through the number of vaccines previously administered per day over a period of the last 30 days, how many vaccines are predicted to be administered a specified number of days into the future?

Responsible for the preprocessing of vaccine-related data (preprocessing/prep\_vaccines.py) and the individual script to answer questions and plotting (vaccines\_administered.py).