# CSC110 Project Report: CO(VISION): COVID-19's Impact on Employment

Daniel Xu, Kirsten Sutantyo, Nicole Leung, Victor Zheng

Tuesday, December 14, 2021

## 1 Problem Description and Research Question

The COVID-19 pandemic has affected society in many ways, one of which is employment. COVID forced people to stay at home, quarantine, and dramatically transform their lifestyle to preserve their wellbeing and protect others against the spread of the virus. According to the Government of Canada (2020), certain industries have faced challenges, as many companies have been reluctant to hire because of resource shortages and stagnant business. Other industries have experienced an increase in demand for their products and services, increasing job opportunities. For instance, receptionists experienced a transition to automated administrative functions because of health concerns related to in-person work (Government of Canada, 2020). Another example of an industry benefiting from COVID is the computer engineer job market, which saw an increase in demand thanks to a prevalence of online work (Government of Canada, 2020). A real-life example of a company benefiting from COVID-19 is Amazon Canada, part of the goods-producing industry. According to a CTV News article, in September 2021, the company announced that it would hire 15,000 new warehouse and distribution workers while increasing their starting wage (Stephenson, 2021). But not all industries have had positive experiences with COVID-19. Air pilots experienced significant layoffs due to decreased travel demand and tourism (Government of Canada, 2020). A company suffering from COVID-19 is Cineplex, which is in the entertainment industry. According to CTV News, COVID-19 resulted in $103.7 million in lost revenue, which resulted in a decrease in employment. (Friend, 2021)

We were interested in exploring the correlation between the pandemic and employment because of the effect on our personal lives. From the results of our study, we can analyze the impact of COVID-19 on our career paths and modify our decisions accordingly. For instance, a topic that affects us is the program(s) that we must choose after our first year of university. Noticing that a particular industry has an increasing number of job opportunities with a positive correlation between COVID-19 and employment would mean that it would be a good idea to pursue a program affiliated with the industry. Choosing a program that thrives during COVID-19 indicates that our jobs will survive future pandemics. For this reason, our research question is: **How does the pandemic impact employment in Ontario? Are there certain industries that suffered or benefited more than others?**

## 2 Dataset Description

For our project, we chose to use two datasets, one from the Government of Ontario and the other from Statistics Canada.

From the Government of Ontario, we have taken their rolling data on COVID-19 positive cases from in csv format. It contains daily updated COVID-19 cases that have been recorded in Ontario since the start of the pandemic. From the variables, we only took the variable "Accurate_Episode_Date", which contains the date of the COVID case occurrence. The data from this website contained information from 2019, so we decided to filter the data from January 2020 onward and display that data in our final project. We filtered the data this way since the data from 2019 was not consistent enough to be used.

From Statistics Canada, we have taken data containing information about the employment by industry, monthly, and seasonally adjusted from "October 2019" to "November 2021" in csv format for Ontario. We used all industries, starting from line 14 to line 32 inclusive, totalling 18 industries including the "Total industries" observation.

# 3 Computational Plan

## 3.1 Filtering

First, we created a function called filter.py to filter our datasets to better suit our final product. We used a function called filter_employment_data() to filter the employment dataset file into a CSV file called filtered_employment_data.csv. We filtered the CSV to contain only the header of the original CSV file (changing the format of the dates in the header to match the format from the COVID-19 cases dataset), the number of employees in Ontario and the North American Industries that they are employed in. Then, we used a function called filter_covid() that filtered the COVID-19 case dataset file into a new CSV file named filtered_covid_cases.csv. We filtered this data to contain only the date (which we formatted to as month and year) and the number of cases in that given month.

Next, we created a python file named extract.py which creates dataclasses to encapsulate the filtered csv data as a COVID dataclass called CovidData and to extract the filtered employment data into a dataclass called Employed.

## 3.2 Employment Data data class

For our employment data, we created a data class called "Employed" that contains instance attributes such as industry (the name of the industry as a string), employment (a list containing the number of employees employed in this industry per month (in thousands) as a float), and date (a list containing the dates corresponding to employment numbers as a string). Then we created a function called add_employment_data() which takes in no input, and generates a list containing this data class object. The function loops through each industry, which are different rows in the filtered_employment.csv excluding the first row. Within each loop, the function loops through each date, or each column in filtered_employment.csv excluding the first column which is just the name of the industry. Within each nested loop, the function generates a data class object with the dates of the row as the attribute "date", the number of employees employed for that specific industry in each month as the attribute "employment", and lastly the name of the industry as the attribute "industry". At the end of the loop, the function appends this object into a list and returns it.

## 3.3 Covid Data data class

For COVID-19 cases, we created a data class called "CovidData" that contains instance attributes such as date (the date formatted as a string), and cases (the number of cases as an integer representing the number of cases during that month). We created a function called add_covid_data() which takes in no input, and generates a list containing this data class object. The function loops through each row, excluding the header, and generates a data class object with the date of that row as the attribute "date", and the number of cases of that row as the attribute "cases"'. At the end of the loop, the function appends this object into a list and returns it.

### 3.4 Linear Regression and Correlation Computation

To use linear regression, we need to confirm some assumptions. One of the main assumptions is that there is some sort of linear association between the x and y variables. Thus, we created a function called correlation_calculator to calculate correlations in visualizations.py to calculate the strength and direction of the linear association. This function uses the Pearson Correlation Coefficient Formula, the most commonly used correlation formula:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{(n\sum x^2 - (\sum x)^2)(n\sum y^2 - (\sum y)^2)}}$$

where "r" is the Pearson Coefficient and "n" is the number of pairs of values (Thakur, 2021). Note: scores in this case are interchangeable with coordinates. The function takes inputs of given x-coordinates as a list and given y-coordinates as a list and returns a float. To simplify this equation, we separate the Pearson Correlation Coefficient Formula into smaller equations, and combine them at the end of the function. First, we calculate "n", which is the number of coordinate pairs inputted to the function. Second, we calculate the sum for x coordinates and $x^2$ coordinates. Third, we calculate the sum for y coordinates and $y^2$ coordinates. Fourth, we calculate the sum for x coordinates multiplied by y coordinates. Lastly, we combine these smaller equations together in the form of the Pearson Correlation Coefficient formula and calculate a number r, which is our correlation. As a guideline, if r = -1, it implies a strong negative relationship. If r = 0, there is no relationship at all, and if r = 1, it implies a strong positive relationship (Thakur, 2021). Also note that r can only take a value between -1 and 1.

Assuming that there is a correlation, we have to now calculate linear regression. In our file called visualizations.py, we created a function called linear_regression_model which takes inputs of given x-coordinates as a list and given y-coordinates as a list and returns a tuple of two floats. The function works by following the least squares regression line formula, which is:

$$m = \frac{N\sum(xy) - \sum x \sum y}{N\sum(x^2) - (\sum x)^2}$$

where "m" is the slope and "N" is the number of coordinates (Pierce, 2019). To simplify this equation, we separate the Least Squares Regression Line formula into smaller equations, then combine them at the end of the function. First, we calculate the sum for x coordinates and $x^2$ coordinates. Second, we calculate the sum of y coordinates. Third, we calculate the sum for x coordinates multiplied by y coordinates, and lastly we combine these calculations together to fully calculate the linear regression using the least squares regression line formula. The first point returned in the tuple is m, the slope of the line. The second point of the returned tuple is b, the intercept of the line. This helps us to form an equation of a line using the formula $y = mx + b$.

### 3.5 Plot Graphing using matplotlib

For plot graphing, we used the library "matplotlib". There were two types of visualizations that we used for our project, scatterplots and line drawing. Using "matplotlib", we simply needed to provide the graphs with a list of x and y values, before customizing it to fit our needs. We learned how to change the text, orient the x ticks, change the window size, and change the colour. We used matplotlib's scatter function to input the scatter plot coordinates, and we color coded and labelled the plots accordingly. For matplotlib's plot function, we used our calculated linear regression model to determine the leftmost point and the rightmost line on the graph, before mapping that with a line.

The class is run whenever a user presses a button to view our visualizations in the Visual class. For our individual visualizations, we run both "display_individual_graphs" and "industry_covid_visualization". For all visualizations, we run only "display_multiple_associations".

We created functions "get_best_association", "get_worst_association" which calculated the top industries with positive and negative correlations. We did this by using our function "correlation_calculator", and calculating the five industries with the most positive correlations and the 5 industries with the most negative correlations.

We also created functions "get_struggling_industries", and "get_benefited_industries" which calculated the top industries with the steepest positive and steepest negative slopes. We did this by using our function "correlation_calculator", and calculating the five industries with the most positive slopes and negative slopes, respectively.

The function "add_linear_regression_model" creates a straight line through the provided linear regression model that is imputed.

The function "industry_covid_visualization" looks for the employment industry provided and creates a visualization from the start date until the end date with both COVID data and employment data mapped on the y-axis. Every month from the start date to the end date is mapped along the x-axis.

### 3.6 Graphical Interface using Pygame

For the graphical interface, we created a new class called Visual which utilises the module pygame, and is responsible for displaying all of the graphics and handling user interactions. For the graphics portion, we created a method for each of the menus in our application, one for the start menu, one for the individual comparison menu, and one for the all comparison menu. Each method then contains statements that help to display all the graphics onto the menu. This includes the buttons, the title, the subtitle and background images. To accomplish this while keeping our code as simple and clean as possible, we created another class called Button which allows us to create button objects much more easily, and helper functions such as draw_text for displaying text onto the menu. These methods utilised many of pygame's methods such as pygame.draw and pygame.rect for the buttons and texts, pygame.font for the custom fonts, and pygame.image for the application icon and small animations we have at the start menu. As for the user interactions, we accomplished this by having each of the menu methods be in an infinite loop that iterates until the user exits the program, constantly listening for any user input. The rate at which the loop iterates is controlled by the pygame.time.Clock().tick() which controls the refresh rate. As for user input, we used a brand new pygame concept called event, which is how pygame receives various user inputs such as mouse click, mouse position, and keyboard clicks. All of this allows us to create functional buttons that when clicked, perform certain actions such as manoeuvring between different menus and running the matplotlib function to display graphs.

# 4 Instructions

1. Download all our files from MarkUs.

   They will be downloaded into a zip file located in your downloads section, and you will be required to extract those files. They should automatically be extracted into a folder. If not, compile all our files into one directory and copy/move that file into PyCharm, which is what we will be using to run it.

2. Install all Python libraries listed under our requirements.txt file

   These files include:

   matplotlib 3.5.0

   pygame 2.0.1

   python_ta 2.0.0

3. Download the data sets as follows:

   (a) For the Covid-19 data, the URL is https://data.ontario.ca/en/dataset/confirmed-positive-cases-of-covid-19-in-ontario. Download the "Confirmed positive cases of COVID19 in Ontario csv. Then rename the file "employment_data.csv". Then, move the file into the same directory as main.py.

   (b) For the Employment data, the URL is https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1410035501. Our data was downloaded by selecting "Ontario" as the province, a date range from "October 2019 to November 2021", and "seasonally adjusted". Then, select "download" csv file and rename the dataset to covid_cases.csv. Then, move the file into the same directory as main.py.

   (c) Alternatively, we have also uploaded our dataset files to https://send.utoronto.ca/pickup.php. After downloading, move the files to the same directory as the main.py file. We included this link because our files are too large for uploading and Statistics Canada experienced multiple technical difficulties to the leadup of our submission.

   - Claim ID: tGd3SQ3yFqEsf5sT
   - Claim Passcode: o3pWaPdRnxWaXxJk

4. Run main.py

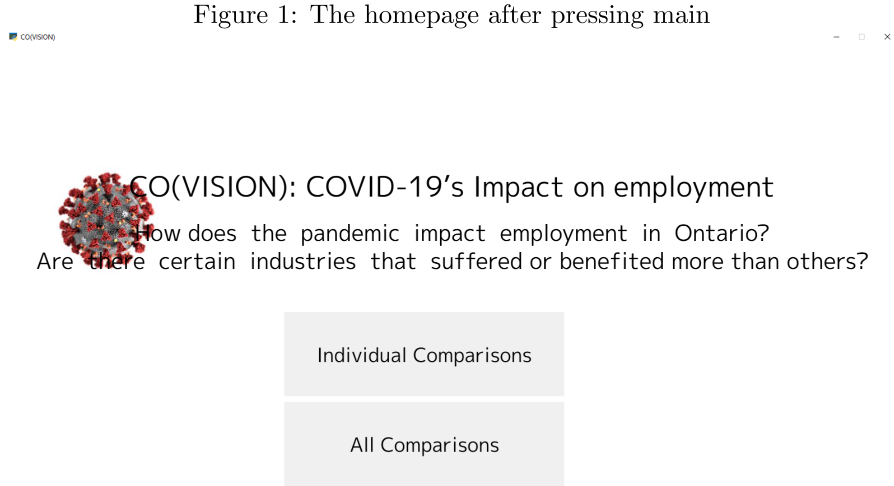
   Note: **Do not run with python console**. There may be an error if the code is run through the python console. Instead, click the green arrow in pycharm. Make sure that the run configurations has "Run with Python Console" unchecked. We believe there are problems with how matplotlib, pygame, and python3.9 are compatible with one another.

   (a) Main Menu

   Immediately after running "main.py", the main menu will open to display text that introduces our title "CO(VISION): COVID-19's Impact on employment" and our research question "How does the pandemic impact employment in Ontario? Are there certain industries that suffered or benefited more than others?".

The window name "CO(VISION)" will be displayed on the top left corner of the screen. There will be a bouncing png image of the COVID-19 virus; it changes direction once it hits a border of the 1500 by 800 screen. In the first frame, there will be two interactive buttons, "Individual Comparisons" and "All Comparisons" which will lead to separate pages of our application. If the user clicks on "Individual Comparisons", they will visit a new page (see 3.b. for information). If the user clicks on "Individual Comparisons", they will visit a new page (see 3.e. for information). The button will change colour if the user hovers over it.

Figure 1: The homepage after pressing main

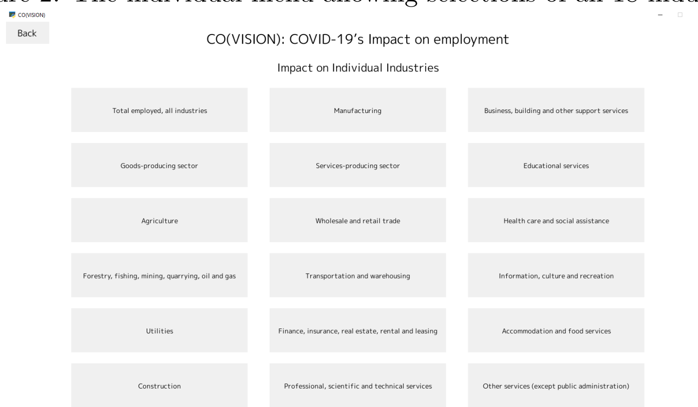



(b) Individual Comparisons

After clicking "Individual Comparisons" on the main menu. A new page will open to display a header "CO(VISION): COVID-19's Impact on employment" and "Impact on Individual Industries". The screen will change to show a selection of 18 interactive buttons, each with a different description, lined in 3 columns by 6 rows. The button will change colour if the user hovers over it. The user may click on the button labeled "Back" to return to the main menu (see 3. a. for information). The user may click on any of the 18 buttons to open a graph (see 3. c. for information).

Note: In case you are unable to see a back button or see the entire screen, try and show your windows side by side. This may be an issue with the compatibility of our program with the size of your screen.

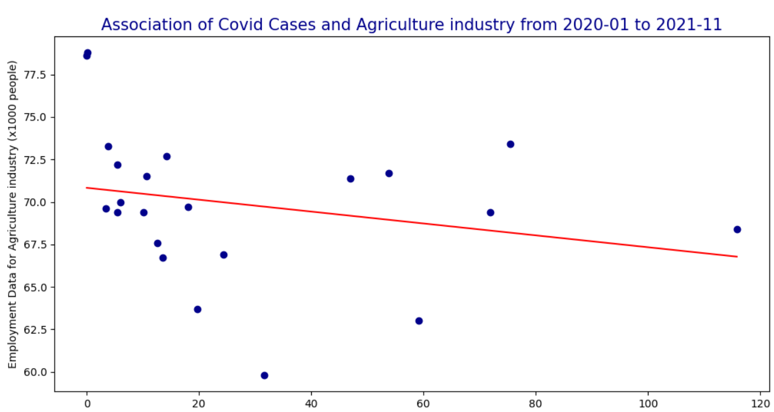
Figure 2: The individual menu allowing selections of all 18 industries




(c) Example of Individual Comparison part 2 for Agriculture industry

For example, the user may choose to click on the button labeled "Agriculture". We display a graph with a red line for linear regression, based on the blue points plotted with the x-axis of "Covid Cases (x1000 people)" and a y-axis "Employment Data for Total industry (x1000 people)". A header, "Association of Covid Cases and Agriculture industry from 2020-01 to 2021-11", is shown at the top of the graph. The browser caption is labeled "Figure 1" on the top left of the screen. The user may manipulate the graph by using the hotbar on the bottom left side of the screen. The user may choose to reset the graphs original view, return to the previous or next view, pan the graph, zoom in on a specific part of the graph, configure subplots, or save the graph as a Portable Network Graphics to the users own computer. The user may hover their mouse over the browser to view the location of a certain point, it will be shown on the bottom right of the screen. Once the user exits the screen, by clicking on the top right "x", a new window will pop-up (see 3.d. for more information). The user interface will remain the same for all other buttons displayed on the "Individual Comparisons" page (see 3.b. for more information). However, the points and header will be different since they depend on the data and name for the corresponding industry. The individual correlations and linear models are printed in the python console.

Figure 3: Relationship of Covid Cases and Agriculture - individual display with linear regression



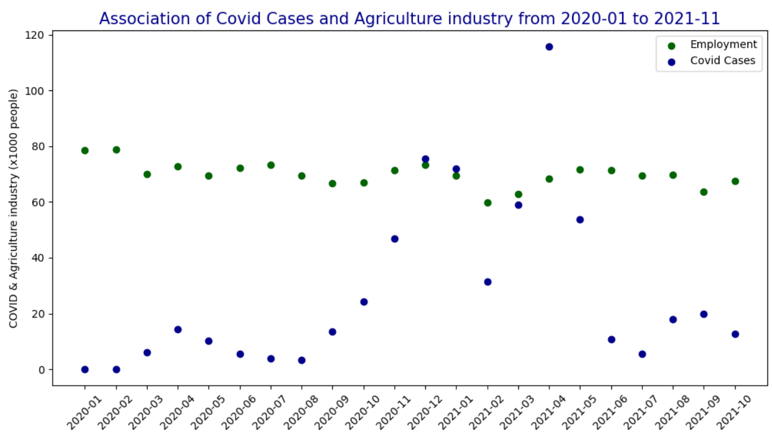
(d) Example of Individual Comparison part 2 for Agriculture industry

Note: Our application freezes the pygame UI until all matplotlib graphs are closed.

After the first visualization is closed, a second visualization will immediately open, displaying a graph with green points representing the employment and blue dots representing Covid cases plotted with the x-axis of months of 2020-01 to 2021-11. A legend will show on the right side for "Employment" in green, and "COVID Cases" in blue. The user can close the pop-up window by clicking the "x" on the top right side of the screen.

The user interface will remain the same for all other buttons displayed on the "Individual Comparisons" page (see 3.b. for more information). However, the points and header will be different since they depend on the data from the specified industry.

Figure 4: Association of COVID and Agriculture - individual display with months



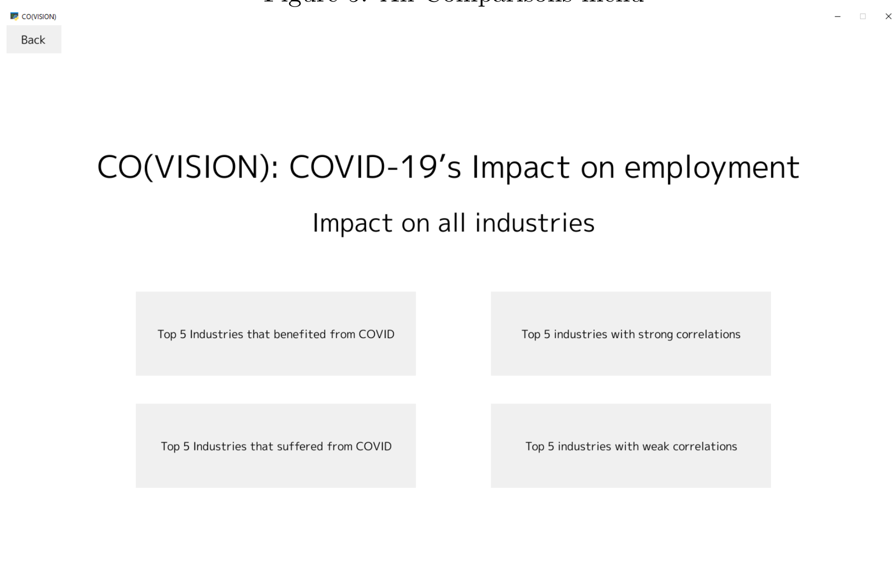

(e) All Comparisons

Back in the main menu, clicking "All Comparisons" will bring you to Figure 5. A new page will open to display a header "CO(VISION): COVID-19's Impact on employment" and "Impact on all Industries". The screen will change to show a selection of 4 interactive buttons, each with a different description, lined in 2 columns by 2 rows. The button will change colour if the user hovers over it. The user may click on the button labeled "Back" to return to the main menu (see 3. a. for information). The user may click on any of the 4 buttons to open a graph (see 3. f., 3. g., 3. h., 3. i., for information).

(f) Top 5 Industries that benefited from COVID

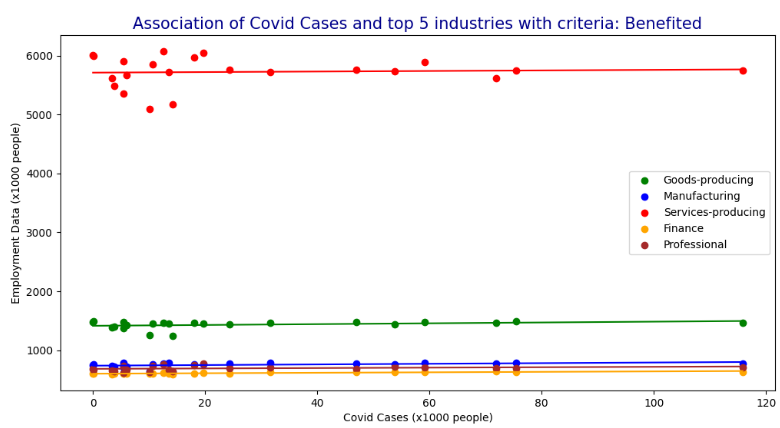
The user may choose to click on the button labeled "Top 5 Industries that benefited from COVID". A header will show above the graph, "Association of Covid Cases and top 5 industries with criteria: Benefited". A legend will show on the right side for "Goods-producing" in green, "Manufacturing" in blue, "Services-producing" in red, "Finance" in yellow, and "Professional" in brown. We display a graph with green, blue, red, yellow, and brown points and their respective linear regression; plotted with the x-axis "Covid Cases (x1000 people)" and a y-axis "Employment Data (x1000 people)". This graph

Figure 5: All Comparisons menu




shows a positive correlation. The graph has the same user interface as "Total employed, all industries (Graph 1)" (see 3. c. for more information). The user can close the pop-up window by clicking the "x" on the top right side of the screen.

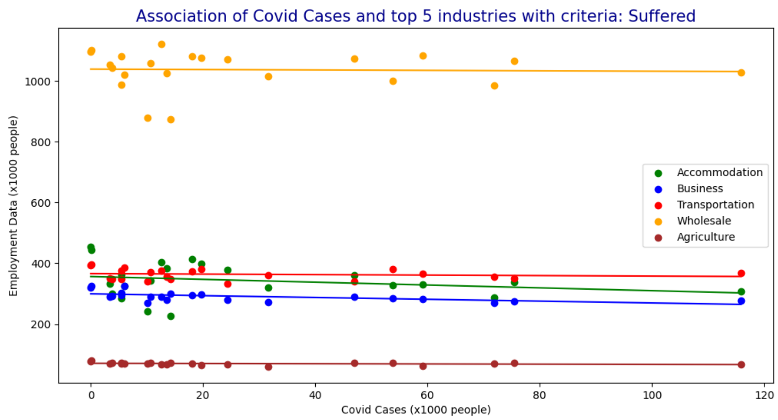
Figure 6: Top 5 industries with positive slopes




(g) Top 5 industries that suffered from COVID

The user may choose to click on the button labeled "Top 5 Industries that suffered from COVID". A header will show above the graph, "Association of Covid Cases and top 5 industries with criteria: Suffered". A legend will show on the right side for "Accommodation" in green, "Business" in blue, "Transportation" in red, "Wholesale" in yellow, and "Agriculture" in brown. We display a graph with green, blue, red, yellow, and brown points and their respective linear regression; plotted with the x-axis "Covid Cases (x1000 people)" and a y-axis "Employment Data (x1000 people)". This graph shows a negative correlation. The graph has the same user interface as "Total employed, all industries (Graph 1)" (see 3. c. for more information). The user can close the pop-up window by clicking the "x" on the top right side of the screen.

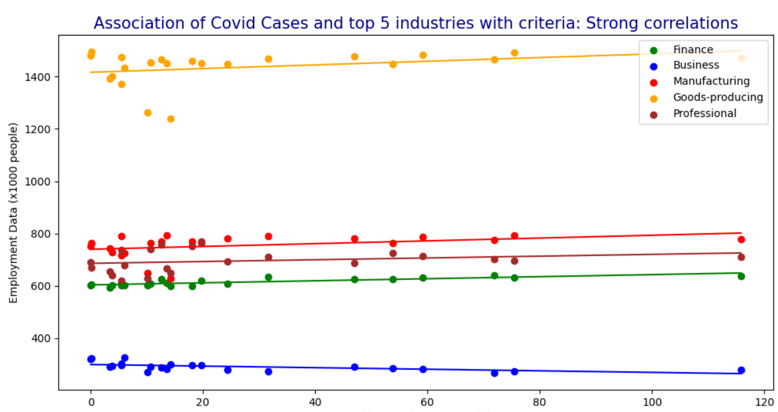
Figure 7: Top 5 industries with negative slopes




(h) Top 5 industries with strong correlations

The user may choose to click on the button labeled "Top 5 Industries with strong correlations". A header will show above the graph, "Association of Covid Cases and top 5 industries with criteria: Strong association". A legend will show on the right side for "Finance" in green, "Business" in blue, "Manufacturing" in red, "Goods-producing" in yellow, and "Professional" in brown. We display a graph with green, blue, red, yellow, and brown points and their respective linear regression; plotted with the x-axis "Covid Cases (x1000 people)" and a y-axis "Employment Data (x1000 people)". This graph shows both high positive and negative correlation. The graph has the same user interface as "Total employed, all industries (Graph 1)" (see 3. c. for more information). The user can close the pop-up window by clicking the "x" on the top right side of the screen.

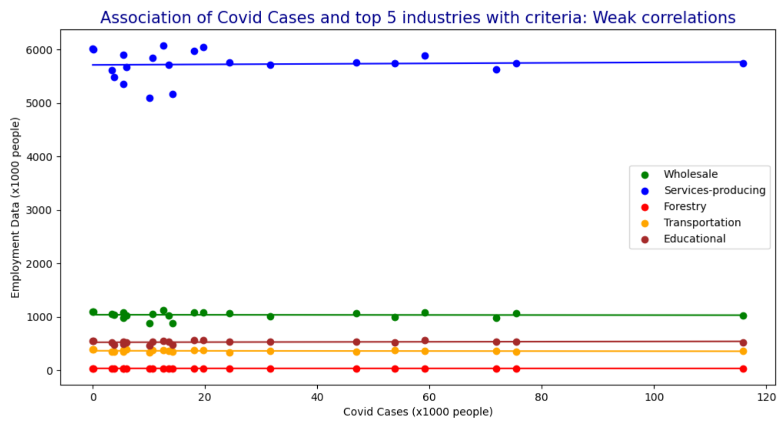
Figure 8: Top 5 industries with strong positive correlations




(i) Top 5 industries with weak correlations

The user may choose to click on the button labeled "Top 5 Industries with weak correlations". A header will show above the graph, "Association of Covid Cases and top 5 industries with criteria: Weak association". A legend will show on the right side for "Wholesale" in green, "Services-producing" in blue, "Forestry" in red, "Transportation" in yellow, and "Educational" in brown. We display a graph with green, blue, red, yellow, and brown points and their respective linear regression; plotted with the x-axis "Covid Cases (x1000 people)" and a y-axis "Employment Data (x1000 people)". This graph shows a neutral correlation. The graph has the same user interface as "Total employed, all industries (Graph 1)" (see 3. c. for more information). The user can close the pop-up window by clicking the "x" on the top right side of the screen.

Figure 9: Top 5 industries with strong negative correlations



# 5 Changes from Proposal

We mentioned how we would organize the COVID-19 data with a dictionary, but after trial and error, we decided that a dictionary would not be any more beneficial than a list of classes, which we chose to implement instead.

In our proposal, we mentioned how we wanted to do linear regression. However, we had not yet decided exactly how to accomplish what we wanted. We originally had stated that we wanted to have the "month" as the independent variable and have some sort of linear regression over time. However, we realized that this would not be possible because the months are categorical variables while employment and COVID are numerical variables.

We also did not mention the discrepancy in COVID cases and employment numbers. To tackle this problem, we made both numbers representative of 1000 people.

In our proposal, we explained how we would create a linear regression model in addition to a rate of change model, which did not make sense to us as we coded it. We decided to change the speed of change computation into a correlation computation. We also did not mention how linear regression or correlations in our project proposal was good, so we had to do additional research to find the Least Squares Linear Regression model and the Pearson Coefficient formula.

# 6 Results

Table 1: Industries with steepest positive slopes via linear regression

| **Industry** | **Slope** |
|---|---|
| Goods-producing sector | 0.7021303529165634 |
| Manufacturing | 0.5351943399132434 |
| Services-producing sector | 0.46622911271358136 |
| Finance, insurance, real estate, rental and leasing | 0.3919784646670556 |
| Professional, scientific and technical services | 0.3423614518254058 |

Table 2: Industries with strongest positive correlations

| **Industry** | **Slope** |
|---|---|
| Finance, insurance, real estate, rental and leasing | 0.8301563379109896 |
| Manufacturing | 0.3747475348256192 |
| Goods-producing sector | 0.3173450552333198 |
| Professional, scientific and technical services | 0.2482586381446119 |
| Construction | 0.247518396409972 |

Table 3: Industries with the steepest negative slopes

| **Industry** | **Slope** |
|---|---|
| Accommodation and food services | -0.4656417034139607 |
| Business, building and other support services | -0.30058904646127704 |
| Transportation and warehousing | -0.08140254454714102 |
| Wholesale and retail trade | -0.06958866856459278 |
| Agriculture | -0.03499257958744931 |

Table 4: Industries with strongest negative correlations

| **Industry** | **Slope** |
|---|---|
| Business, building and other support services | -0.5706787420029783 |
| Accommodation and food services | -0.241887820489557 |
| Agriculture | -0.2401865117473154 |
| Utilities | -0.19692219527741558 |
| Transportation and warehousing | -0.137103509292598 |

The results of our computational exploration are displayed above from Figures 1-4. Note: these are the results from the dates of January 2020 to November 2021.

Based on our data, we found that the industry that has the steepest positive slope, via linear regression from the pandemic, is the goods-producing industry. This industry had a slope of 0.702, which was the largest out of all industries we analyzed, implying that overall, an increase of 1000 covid cases resulted in an increase in 700 or so jobs. For linear regression to be valid, we needed a linear association between x and y. The goods-producing sector fulfils this requirement because the correlation is 0.317, which is part of our top 5 positive correlations (figure 2). A correlation of 0.317 indicates that there is a moderate positive linear association between the goods-producing sector and COVID-19. Similarly, the Manufacturing, Finance, and Professional, scientific and technical services industries were in the top 5 for positive slopes and were in the top 5 for positive correlations. This indicates that these industries have valid linear regression models. We can thereby conclude from our data that the Goods Producing, Manufacturing, Finance, and Professional, scientific and technical services benefited from the pandemic.

We might hypothesize that the goods-producing sector benefited because of an increase in purchases thanks to the closure of movies, travel, fairs, and recreational activities such as snowboard parks (Gov. of Ontario). This aligns with how companies such as Amazon faced a surge in their business due to more online purchases (Weise 2020).

However, surprisingly, in our top 5 industries with a positive slope, the services-producing sector had a positive slope yet was not part of our top 5 correlations. Their spot was taken by the Construction industry. Based on our data, we conclude that the weak correlation of the services-producing industry might indicate that the linear regression model might have invalid results. Looking at the individual graph of the services-producing sector, we see that there is a large fluctuation in the points to the fitted line. We see that there are a few instances where there were less than 20 thousand COVID cases and with employment over 6 million, while other months had employment down to the low 5 millions. One reason might be that at the start of the pandemic, services-producing markets such as restaurants faced fluctuating job numbers. Many chefs and waitresses were laid off, but when restrictions started to ease, there was a large increase in employment (Hansen 2020).

Conversely, our data suggests that the industry with the steepest negative slope via linear regression from the pandemic is the Accommodation and food services industry. This industry has the steepest negative slope of -0.465, which implies that overall, an increase of 1000 COVID cases resulted in a decrease of around 460 jobs. For linear regression to be valid, we needed a linear association between x and y. The Accommodation and food services industry correlation is -0.241, indicating a moderate negative linear association between x and y, fulfilling this assumption. Similarly, the Business, building and other support services, Agriculture, and Transportation and warehousing industries had steep negative slopes with strong negative correlations. Based on our data, we conclude that the Accommodation and food services industry, Business, building and other support services, Agriculture, and the Transportation and warehousing industries suffered from the pandemic.

In our top 5 industries with a steep negative slope, we found that Wholesale and retail trade had a negative slope significant enough to be in the top 5 but did not have a negative correlation significant enough to be in the top 5 negative correlations. Based on our data, we say that the linear regression model might have invalid results due to a weak negative correlation. Looking at the individual graph of the Wholesale and retail trade industry, we see that there is a significant fluctuation in the points to the fitted line. Similar to the services-producing sector, there are months with less than 20,000 COVID cases with over 1,100,000 jobs, while other months have below 900,000 jobs. We might reason that Wholesale and retail trade markets, such as department stores, faced fluctuating job numbers. At the pandemic's start, many of these retail workers were laid off since workers in department stores were viewed as non-essential. Thus, we see a decrease in employment at the pandemic's start.

# 7 Discussions

Our computational exploration helped us in identifying the different industries impacted by the pandemic as well as the strength of the correlation between the two variables, employment of a certain industry and COVID cases. The question of "how the pandemic impacted employment in Ontario" can be tough to answer since many of the industries we observed had many unique impacts. To identify industries that suffered or benefited more than others, we use our linear regression calculations along with correlation calculations and find those with a steep linear regression slope and compare their correlations. A steep slope on the graph refers to a strong positive or negative relationship between the employment of the specified industry and the pandemic.

## 7.1 Limitations

There may have also been limitations in our project, specifically in how some COVID cases were undetected. Some patients would have stayed home and recovered without getting tested. The CDC estimates that 1 in 4 COVID cases are reported (CDC, 2021). Therefore, asymptomatic and unreported cases would decrease employment even though the official tally of COVID would not be affected.

Public Health Ontario looked at the false positivity of testing and found that the sensitivity was between 70% and 90% (Gov. of Ontario). This indicates that between 10% and 30% of those with COVID and get tested are given negative COVID results. Those with false-negative results might decide to go back to work, which would increase employment even though the worker has COVID.

Another limitation we considered was the false-positive error, where you are told that you have COVID, but in fact, you don't. However, Public Health Ontario states that this rate is close to 100%, with a specificity of ¿99.99% (Gov. of Ontario). Such a high specificity means that the number of correct predictions of people with COVID as actually having COVID was high. Therefore, the false positivity would not have affected our results greatly.

## 7.2 Future Implementations/Further exploration

In our planning, we wanted to add a way to display the visualizations for a specific date. However, we couldn't accomplish this on time since we were not sure how to add a text input into pygame. Our computations do have the option to specify a date.

Another future implementation that we wanted to do was in looking at confounding variables that could have impacted the linear regression. With more knowledge on how to do this, we would have accomplished it.

# 8 Concluding Thoughts

Our project allowed us to identify the pandemic's impact on Ontario's industries in various ways. We visualised how the pandemic impacted employment in Ontario through different linear regression graphs and found that not all industries were affected equally. Industries with a positive linear association benefited from the effects of COVID while industries with a negative linear regression suffered. We found that Goods-producing sector, Manufacturing, Finance, insurance, real estate, rental and leasing, Professional, scientific and technical services, and construction industries had the most significant increase in employment rate during the pandemic. Accommodation and food services, Business, building, and other support services, Transportation and warehousing, and Agriculture industries had the largest decrease in the employment rate during the pandemic. By viewing the correlations, we found that Wholesale and retail trade along with Services-producing industries had linear regression models that did not fulfil assumptions of linear regression. For us, seeing the Professional, scientific and technical services industry as an industry that benefited during COVID tells us that we should look to make the POst in a STEM related field.

# References


Centers for Disease Control and Prevention (CDC). (2021, November 16). *Estimated covid-19 burden. Centers for Disease Control and Prevention.* Retrieved December 13, 2021, from https://www.cdc.gov/coronavirus/2019-ncov/cases-updates/burden.html.

Friend, D. (2021, August 12). *Cineplex faces another quarterly loss as movie theatres slowly reopened across Canada.* CTVNews. Retrieved November 5, 2021, from https://www.ctvnews.ca/business/cineplex-faces-another-quarterly-loss-as-movie-theatres-slowly-reopened-across-canada-1.5544740.

Government of Canada / Gouvernement du Canada. (2020, November 16). *Outlooks for COVID-19 Impacted Occupations in Ontario.* Job Bank. Retrieved November 5, 2021, from https://www.jobbank.gc.ca/trend-analysis/job-market-reports/ontario/prospects-report.

Government of Canada, Statistics Canada. (2021, October 8). *Employment by industry, monthly, seasonally adjusted and unadjusted, and trend-cycle, filtered for Ontario, Feb. 2020 - Oct. 2021.* Statistics Canada. Retrieved November 5, 2021, from https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1410035501.

Government of Ontario. (2020, March 18). *Update on Ontario Parks Operations in Response to COVID-19.* Ontario newsroom. Retrieved December 14, 2021, from https://news.ontario.ca/en/statement/56377/update-on-ontario-parks-operations-in-response-to-covid-19.

Government of Ontario. (2021, November 4). *Confirmed positive cases of COVID-19 in Ontario - Datasets - Ontario Data Catalogue.* Retrieved November 4, 2021, from https://data.ontario.ca/en/dataset/confirmed-positive-cases-of-covid-19-in-ontario.

Hansen, J. (2021, June 9). *Rehiring is finally on the table for more restaurants - but not all workers are coming back — CBC news.* CBCnews. Retrieved December 14, 2021, from https://www.cbc.ca/news/business/restaurant-rehiring-pandemic-jobs-1.6058110.

Matplotlib . (2021, August 13). Overview - *Matplotlib 3.4.3 documentation.* Retrieved November 5, 2021, from https://matplotlib.org/stable/contents.html.

Pierce, R. (2019, March 2). *Least Squares Regression. Math is Fun.* Retrieved December 13, 2021, from https://www.mathsisfun.com/data/least-squares-regression.html.

*Pygame Documentation.* Pygame Front Page - pygame v2.0.1.dev1 documentation. (n.d.). Retrieved November 5, 2021, from https://www.pygame.org/docs/.

Stephenson, A. (2021, September 13). *Amazon to hire 15,000 employees across Canada; increase wages.* CTVNews. Retrieved November 5, 2021, from https://www.ctvnews.ca/business/amazon-to-hire-15-000-employees-across-canada-increase-wages-1.5582942.

Thakur, M. (2021, November 22). *Pearson Correlation Coefficient.* WallStreetMojo. Retrieved December 13, 2021, from https://www.wallstreetmojo.com/pearson-correlation-coefficient/.

Weise, K. (2020, November 27). *Pushed by pandemic, Amazon goes on a hiring spree without ...* New York Times. Retrieved December 14, 2021, from https://www.nytimes.com/2020/11/27/technology/pushed-by-pandemic-amazon-goes-on-a-hiring-spree-without-equal.html.