Term Project

Group #9 - Section 17

Kevin Dao (#501162612)

Minh Quan Nguyen (#501158694)

Jason Nguyen (#501087930)

Instructor: Alexander Ufkes

TA: Sean Davies-Lee

INTRODUCTION

Diabetes is a severe public health issue in Canada, and its impact on the Canadian population has been extensively examined throughout the years. In this lab report, we will investigate Statistics Canada data from 2015 to 2021 to determine the average proportion of the population diagnosed with diabetes in Canada's four most populous provinces (Ontario, Quebec, British Columbia, and Alberta), as well as the country as a whole (excluding territories). The data collected is for those aged 35 and above, and we'll look at the averages for each province and the country as a whole.

In addition, we will compute the average percentage of diabetes for each age group (35-49, 60-64, and 65+) and the yearly averages for each province and the country as a whole. We will also assess which provinces have the greatest and lowest percentages of diabetes and compare these figures to the national average to see whether any provinces have rates that are higher or lower than the national average.

Also, we will create graphs to graphically represent the data, including a bar chart showing the average percentages of diabetes across the three age groups for the entire nation and a simple line plot indicating diabetes prevalence from 2015 to 2021 for all age groups combined.

This report's goal is to perform a detailed investigation of the prevalence of diabetes in Canada's four most populous provinces and to provide information on trends and patterns that might be used to guide public health policies and initiatives.

DATA RESULT

Question 1.A: Provincial Averages

```
Question 1.A --- Provincial Averages ---
Provincial diabetes average of Ontario: 11.70%
Provincial diabetes average of Quebec: 10.45%
Provincial diabetes average of British Columbia: 9.67%
Provincial diabetes average of Alberta: 10.86%
```

The code uses information from the arrays "geo" and "value" to determine the average diabetes prevalence for each of the four Canadian provinces of Ontario, Quebec, British Columbia, and Alberta. The programme first defines variables to keep track of the count and total prevalence of diabetes for each province. It then loops through each row of data, adding up the count and total for each province. To make sure that only correct data is utilized in the calculations, the code also checks for 0 values. The code then uses a prepared print statement to print the calculated provincial averages.

Question 1.B: National Averages

```
Question 1.B --- National Average ---
Canadian diabetes average: 10.87%
```

The national average of diabetes prevalence in Canada is determined by this code, which loops through all the data rows and determines if a row belongs to "Canada (excluding territories)" or not. The variable 'canada_count' keeps track of the number of non-zero values encountered, 'canada_sum' collects the sum of the non-zero values, and 'canada ave' calculates the average of the non-zero values. The final step is to publish the findings to the console along with a message that includes the national average.

Question 1.C: Yearly Averages

```
Question 1.C --- Yearly Averages ---
The averages for the year 2015
Canada: 10.60%
Ontario: 10.77%
Quebec: 10.90%
British Columbia: 9.30%
Alberta: 9.32%

The averages for the year 2016
Canada: 10.70%
Ontario: 12.20%
Quebec: 9.82%
British Columbia: 8.53%
Alberta: 9.77%

The averages for the year 2017
Canada: 10.95%
Ontario: 11.98%
Quebec: 9.58%
British Columbia: 10.14%
Alberta: 11.97%

The averages for the year 2018
Canada: 10.78%
Ontario: 11.28%
Quebec: 10.65%
British Columbia: 8.52%
Alberta: 11.02%

The averages for the year 2019
Canada: 11.70%
Ontario: 13.03%
Quebec: 10.48%
British Columbia: 11.44%
Alberta: 11.33%

The averages for the year 2020
Canada: 10.60%
Ontario: 11.17%
Quebec: 11.42%
British Columbia: 9.04%
Alberta: 12.88%

The averages for the year 2021
Canada: 10.75%
Ontario: 11.48%
Quebec: 10.47%
British Columbia: 11.65%
Alberta: 9.82%
```

From 2015 through 2021, this code estimates the yearly averages of diabetes prevalence for Canada, Ontario, Quebec, British Columbia, and Alberta. For usage in Question 5, it first generates a file called "plot Q5.txt" and writes a header into it. Then it loops through each year, initializing counters and sums for each region to zero for each year. After that, it runs through each data point and determines if it corresponds to the current year. If it does, it determines which region the data point belongs to and adjusts the sum and counter for that region. It prints the average values for each region for that year on the screen after calculating the averages for each region for the current year. Also, it prints the yearly averages for GNUPLOT question 5 into the file. It closes the file at the end.

Question 1.D: Age Group Averages

```
Question 1.D --- Age Group Averages ---
The averages for the age group 35-to-49
Canada: 4.06%
Ontario: 4.64%
Qubec: 3.35%
British Columbia: 3.43%
Alberta: 4.46%


The averages for the age group 50-to-64
Canada: 10.33%
Ontario: 11.22%
Qubec: 9.06%
British Columbia: 7.91%
Alberta: 10.29%


The averages for the age group 65-above
Canada: 18.21%
Ontario: 19.24%
Qubec: 18.44%
British Columbia: 15.44%
Alberta: 16.92%
```

This programme determines the average values of a particular attribute for several age ranges and geographical areas. The output is saved to a file called "plot Q6.txt" and printed to the console. The age categories are numbered from 35 to 65 in 15-year increments, with 65 and older constituting the last group. Counting the occurrences of each geographic location and adding the attribute values for each region, the code loops through each age group and each row of data. If the value is not 0, the region's counter is increased. The averages for each province are computed and printed to the output console at the conclusion of each age group loop. The average value for each age group is written into plot_Q6.txt for later use in graphing of GNUPLot.

Question 2: Highest Averages

```
Question 2 --- Highest Average ---
Ontario has the highest percentage 11.70%
British Columbia has the lowest percentage 9.67%
```

Using the data that we got from question 1A, this code computes and prints the percentage average of the four provinces: Ontario, Quebec, British Columbia, and Alberta. It initially defines max and min variables, which will be used to hold the maximum and lowest percentage averages, respectively. Next it invokes the min max method to get the highest and lowest averages and updates the appropriate variables.
The code then checks which province has the largest and lowest percentage averages using a sequence of if and else if statements. Printf statements are used to display the province with the greatest percentage first, followed by a new line, and finally the province with the lowest percentage. With the `%.2lf` format specifier, percentage numbers are presented with two decimal places.

Question 3: Provinces Below and Above National Averages

```
Question 3 --- Provinces Below and Above National Average ---
Provinces with diabetes average below national average:
Quebec
British Columbia
Alberta

Provinces with diabetes average above national average:
Ontario
```

The province percentage averages for Ontario, Quebec, British Columbia, and Alberta are contrasted with the national percentage average using this code (`canada_ave`). A string variable called `avebelow` receives the names of provinces that are below the national average as an appendix, while `aveabove` receives the names of provinces that are above the national average. Finally, it publishes the strings `avebelow` and`aveabove`, which display the names of the provinces that are below and above the national average, respectively, to the console.

Question 4: Max and Min Yearly Averages

```
Question 4 --- Yearly Averages Min Max ---
Year 2015
Lowest Percentage: British Columbia at 9.30%
Highest Percentage: Quebec at 10.90%

Year 2016
Lowest Percentage: British Columbia at 8.53%
Highest Percentage: Ontario at 12.20%

Year 2017
Lowest Percentage: Quebec at 9.58%
Highest Percentage: Ontario at 11.98%

Year 2018
Lowest Percentage: British Columbia at 8.52%
Highest Percentage: Ontario at 11.28%

Year 2019
Lowest Percentage: Quebec at 10.48%
Highest Percentage: Ontario at 13.03%

Year 2020
Lowest Percentage: British Columbia at 9.04%
Highest Percentage: Alberta at 12.88%

Year 2021
Lowest Percentage: Alberta at 9.82%
Highest Percentage: British Columbia at 11.65%
```

The code calculates and prints the lowest and highest yearly average diabetes rates for the four Canadian provinces of Ontario, Quebec, British Columbia, and Alberta using information contained in arrays. The average diabetes rates for each province for each year from 2015 to 2021 are then entered into a 2D array called "diabetes rates.". After determining the average rate for each province for that year, the code loops over the data to extract the rates for each province for each year and stores the findings in the diabetes rates array. The code then used the 'MinMax2dArray' function to compute the minimum and maximum diabetes rates for each year and province then printf results.

GRAPH

Question 5: Simple Line Plot for Yearly Canada Diabetes Averages

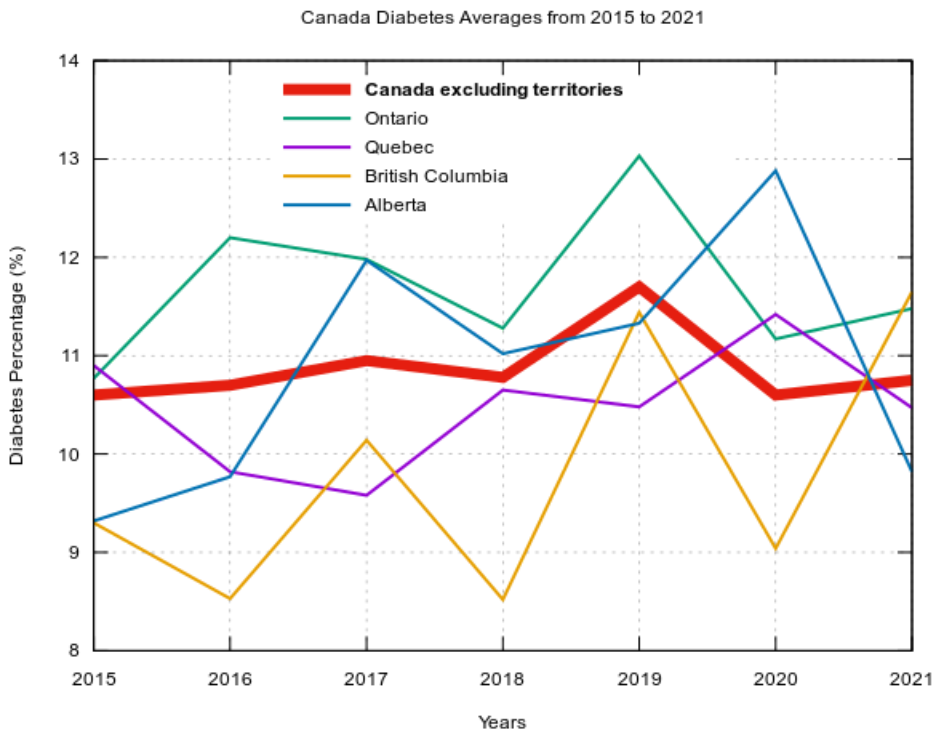

Figure 1.0: Canada Diabetes Averages from 2015 to 2021

This script was created using the graphing and charting application GNUPlot or GNUPLot online. A figure displaying the average diabetes proportion for Canada and its provinces from 2015 to 2021 is styled and formatted using code. The labels for the x-axis and y-axis are set appropriately, and the plot's title is "Canada Diabetes Averages from 2015 to 2021." With the labels arranged in reverse order, the legend (key) is placed to the left, horizontally centered, and aligned to the top of the plot. The y-axis range is adjusted to run from 8 to 14, and the grid is switched on. The "plot Q5.txt" file's data is plotted, with the first column into x axis displaying the years, and the second to sixth columns into y axis representing the percentage average for Canada and 4 provinces. The other provinces are shown with thinner lines (linewidth 2) in different colors (linecolor 2-9) for visual contrast, while the national average is set to be bold and in a thicker line (linewidth 7, linecolor 7) to make it more noticeable. Figure 1.0 was first seen as a SVG file, then converted to PNG in order to meet the report requirements.

Question 6: Bar Chart for Canada Diabetes Averages Per Age Groups

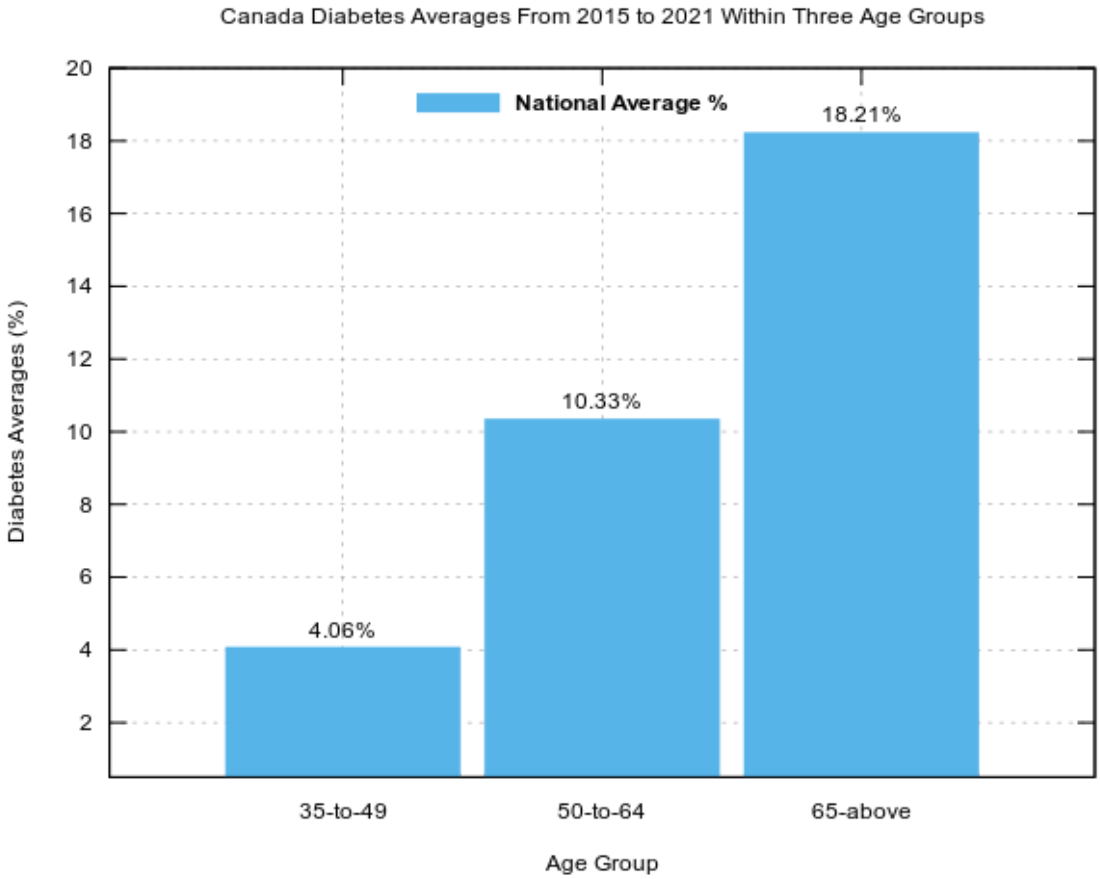

Figure 2.0: Canada Diabetes Averages within three age groups

This script was made for the Canada Diabetes Averages over three age ranges using the internet graphing and charting programme GNUPlot or GNUPLot. "Canada Diabetes Averages From 2015 to 2021 Within Three Age Groups" is the new title of the story. The fill style is solid, and the box width is set to 0.9 times the default width. The grid is turned on, and the y-range axis is set to run from 0.5 to 20. The legend is initially centered horizontally, positioned to the left of the plot, and aligned to the top. Age Group is the label for the x-axis, while Diabetes Averages (%) is the label for the y-axis. The plot then makes use of data from the "plot Q6.txt" file, utilizing the labels in the first column for the x-axis ticks with titles and the second column for the box heights. The national average is indicated on the graphic by the bold title and linecolor 3. Moreover, labels with the height value, expressed as a percentage, are added to each box by the code. The labels have an upward shift of 0.5 units. Overall, this code generates a graphic with clear labeling and style that displays the prevalence of diabetes in three age groups for Canada.

CONCLUSION

The study will look at real-world data from Statistics Canada on the prevalence of diabetes in the four most populous provinces in Canada: Ontario, Quebec, British Columbia, and Alberta. Data for age groups 35 and older were collected between 2015 and 2021. The year, province, age group, gender, and prevalence of diabetes are the key fields in the CSV-formatted data file.

The project requires the use of C programming to read data from a file, perform the necessary calculations, and display the findings as tables, graphs, and conclusions. The calculations include determining averages for each age group in each province and country, averages for each year, and averages for both provincial and national diabetes prevalence rates. The provinces with the highest and lowest rates of diabetes, as well as those that are higher and lower than the national average, must also be identified.

The project also calls for the production of two graphs: a bar chart displaying the average percentages of diabetes across the three age groups for the entire nation, and a line plot displaying the percentages of diabetes for the four provinces and the national average over time. The graph scripts must use GNUPlot capabilities to generate all labels, legends, and titles.

The general objective of the project is to provide a comprehensive analysis of the prevalence of diabetes in the four most populous provinces of Canada, highlighting trends, averages, and comparisons between provinces and age groups. Policymakers and healthcare professionals can benefit from the project's results and recommendations by better understanding the diabetes situation in Canada and using that knowledge to support actions that will improve diabetes treatment and prevention methods.

CODE

C Program Code

```
/*
* Kevin Dao - 501162612
* Jason Nguyen - 501078930
* Daniel Nguyen - 501158694
* */

#include <stdio.h>
#include <string.h>
#include <stdbool.h>
#include <stdlib.h>

#define MAX_ARRAY 300
#define MAX_CHARACTERS 1024

// Function to determine the min and max values of 4 numbers
double
min_max (double a, double b, double c, double d, double *min, double *max)
{
 /*Find minimum and maximum values */
 *min = a;
 *max = a;
 if (b < *min)
   {
     *min = b;
   }
 if (c < *min)
   {
     *min = c;
   }
 if (d < *min)
   {
     *min = d;
   }
 if (b > *max)
   {
     *max = b;
   }
 if (c > *max)
   {
```

```
      *max = c;
    }
  if (d > *max)
    {
      *max = d;
    }
  return *max;
}

// Function to determine the max and min of values from a 2d array
void
MinMax2dArray (double array[][4], int rows, double min[], double max[],
         int min_index[], int max_index[])
{
  for (int i = 0; i < rows; i++)
    {       // Runs through rows of array
      double minValue = array[i][0];
      double maxValue = array[i][0];
      int index_min = 0;
      int index_max = 0;

      for (int j = 1; j < 4; j++)
  {     //  Runs through columns of array
    if (array[i][j] < minValue)
      {     // Compares every number to determine min value
        minValue = array[i][j];
        index_min = j;

      }
    if (array[i][j] > maxValue)
      {     // Compares every number to determine max value
        maxValue = array[i][j];
        index_max = j;

      }
  }

      // Sets arrays to their respective values
      min[i] = minValue;
      max[i] = maxValue;
      max_index[i] = index_max;
      min_index[i] = index_min;
    }
```

```
}

int
main ()
{
 // Variable Declaration
 char row[MAX_ARRAY];
 char ref_dateH[MAX_ARRAY][MAX_CHARACTERS];
 char geo[MAX_ARRAY][MAX_CHARACTERS];
 char age_group[MAX_ARRAY][MAX_CHARACTERS];
 char sex[MAX_ARRAY][MAX_CHARACTERS];
 char value[MAX_ARRAY][MAX_CHARACTERS];
 int index = 0;

 // Open and reading the csv file
 FILE *input_file = fopen ("statscan_diabetes.csv", "r");
 if (input_file == NULL)
   {       // If file does not exist
     printf ("Error opening input file.");
     return 1;
   }

 // Runs through entire csv file until no more input/data
 while ( fgets (row, MAX_CHARACTERS, input_file)); // Get each row of the
csv file and stores it in "row"

   {
    char *token = strtok (row, ",\"");   // Cleans the string of "row" by
removing commas and quotation marks
     int columns = 0;    // Counter variable to count through the amount of
columns
     while (token != NULL && columns < 15) //We only need data from columns
1 - 14
 {
   switch (columns)
     {
     case 0:   // Column 1, REF_DATE
       strcpy (ref_date[index], token);
       break;
     case 1:   // Column 2, GEO
       strcpy (geo[index], token);
       break;
      case 2:    // Column 2, GEO for Canada as it is short one row, DGUID
```

```
for other values (Not needed)
       if (strcmp (geo[index], "Canada (excluding territories)") == 0)
   {
     strcpy (age_group[index], token);
     break;
   }
       break;
        case 3:    // Column 3, SEX for Canada as it is short one row.
AGE_GROUP for other values
       if (strcmp (geo[index], "Canada (excluding territories)") == 0)
   {
     strcpy (sex[index], token);
     break;
   }
       strcpy (age_group[index], token);
       break;
     case 4:   // Column 5, SEX for other values
       if (strcmp (geo[index], "Canada (excluding territories)") == 0)
   {
     break;
   }
       strcpy (sex[index], token);
       break;
     case 12:    // Column 14, VALUE for Canada, nothing for other
       if (strcmp (geo[index], "Canada (excluding territories)") == 0)
   {
     strcpy (value[index], token);
   }
       break;
     case 13:    // Column 14, Value for everything else
       if (strcmp (geo[index], "Canada (excluding territories)") == 0)
   {
     break;
   }
       strcpy (value[index], token);
       break;
     default:
       break;
     }
   token = strtok (NULL, ",\""); // Cleans the string by removing , and "
   columns++;
 }
     index++;
```

```
    }
  fclose (input_file);
  // Question 1.A ----------------------------------------------------
  // counter for each provinces
  // Variable declaration
  double ontario_count = 0.0;
  double ontario_sum = 0.0;
  double ontario_ave;

  double quebec_count = 0.0;
  double quebec_sum = 0.0;
  double quebec_ave;

  double bc_count = 0.0;
  double bc_sum = 0.0;
  double bc_ave;

  double alberta_count = 0.0;
  double alberta_sum = 0.0;
  double alberta_ave;

// Loop through all rows of data
  for (int i = 0; i <= index; i++)
    {
       // Check if the row corresponds to a specific province and accumulate
the sum and count for that province
      if (strcmp (geo[i], "Ontario") == 0)
 {
    if (atof (value[i]) != 0.0)
      {
        ontario_count++;
      }
    ontario_sum += atof (value[i]);
    ontario_ave = ontario_sum / ontario_count;
 }
      else if (strcmp (geo[i], "Quebec") == 0)
 {
    if (atof (value[i]) != 0.0)
      {
        quebec_count++;
      }
    quebec_sum += atof (value[i]);
    quebec_ave = quebec_sum / quebec_count;
```

```
 }
     else if (strcmp (geo[i], "British Columbia") == 0)
 {
   if (atof (value[i]) != 0.0)
     {
       bc_count++;
     }
   bc_sum += atof (value[i]);
   bc_ave = bc_sum / bc_count;
 }
     else if (strcmp (geo[i], "Alberta") == 0)
 {
   if (atof (value[i]) != 0.0)
     {
       alberta_count++;
     }
   alberta_sum += atof (value[i]);
   alberta_ave = alberta_sum / alberta_count;
 }
   }

// Print out the results
 printf ("\nQuestion 1.A --- Provincial Averages ---\n");
 printf ("Provincial diabetes average of Ontario: %0.2f%%\n", ontario_ave);
 printf ("Provincial diabetes average of Quebec: %0.2f%%\n", quebec_ave);
 printf ("Provincial diabetes average of British Columbia: %0.2f%%\n",
   bc_ave);
   printf   ("Provincial   diabetes   average   of   Alberta:   %0.2f%%\n\n",
alberta_ave);

 // Question 1.B ------------------------------------------------
 printf ("\nQuestion 1.B --- National Average ---\n");
 double canada_count = 0.0;
 double canada_sum = 0.0;
 double canada_ave;

 for (int i = 0; i <= index; i++)
   {
     if (strcmp (geo[i], "Canada (excluding territories)") == 0)
 {
   if (atof (value[i]) != 0.0)
     {
       canada_count++;
```

```
    }
   canada_sum += atof (value[i]);
   canada_ave = canada_sum / canada_count;
 }
   }
 printf ("Canadian diabetes average: %0.2f%%\n\n", canada_ave);

 // Question 1.C ------------------------------------------------
 printf ("\nQuestion 1.C --- Yearly Averages ---\n");
  // Creates file for later use in Question 5
 FILE* Q5_File = fopen("plot_Q5.txt", "w");
  fprintf(Q5_File, "#Year\tCan\tOn\tQub\tBC\tAlb\n"); // Prints Header for
file
  // Loop through years 2015-2021
 for (int year = 2015; year < 2022; year++)
   {
     // Initialize counters and sums for each region
     double canadaCounter = 0.0;
     double canadaSum = 0.0;
     double ontarioCounter = 0.0;
     double ontarioSum = 0.0;
     double quebecCounter = 0.0;
     double quebecSum = 0.0;
     double bcCounter = 0.0;
     double bcSum = 0.0;
     double albertaCounter = 0.0;
     double albertaSum = 0.0;

     // Loop through all data points
     for (int i = 0; i < index; i++)
 {
   // Check if data point matches current year
   if (atof (ref_date[i]) == year)
     {
        // Check which region data point is for and update counters and sums
accordingly
       if (strcmp (geo[i], "Canada (excluding territories)") == 0)
   {
     if (atof (value[i]) != 0)
       {
         canadaCounter++;
       }
     canadaSum += atof (value[i]);
```

```
    }
        else if (strcmp (geo[i], "Ontario") == 0)
    {
      if (atof (value[i]) != 0)
        {
          ontarioCounter++;
        }
      ontarioSum += atof (value[i]);
    }
        else if (strcmp (geo[i], "Quebec") == 0)
    {
      if (atof (value[i]) != 0)
        {
          quebecCounter++;
        }
      quebecSum += atof (value[i]);
    }
        else if (strcmp (geo[i], "British Columbia") == 0)
    {
      if (atof (value[i]) != 0)
        {
          bcCounter++;
        }
      bcSum += atof (value[i]);
    }
        else if (strcmp (geo[i], "Alberta") == 0)
    {
      if (atof (value[i]) != 0)
        {
          albertaCounter++;
        }
      albertaSum += atof (value[i]);
    }
        }
  }
    // Print averages into file for question 5 GNUPLOT
     fprintf(Q5_File, "%d\t%.2f\t%.2f\t%.2f\t%.2f\n", year, (canadaSum
/    canadaCounter),    (ontarioSum    /    ontarioCounter),    (quebecSum    /
quebecCounter), (bcSum / bcCounter), (albertaSum / albertaCounter));

      // Print the average values for each region for the current year
      printf ("The averages for the year %d\n", year);
      printf ("Canada: %.2f%%\n", (canadaSum / canadaCounter));
```

```
    printf ("Ontario: %.2f%%\n", (ontarioSum / ontarioCounter));
    printf ("Quebec: %.2f%%\n", (quebecSum / quebecCounter));
    printf ("British Columbia: %.2f%%\n", (bcSum / bcCounter));
    printf ("Alberta: %.2f%%\n\n", (albertaSum / albertaCounter));
  }
  fclose(Q5_File);

  // Question 1.D ------------------------------------------------
  printf("\nQuestion 1.D --- Age Group Averages ---\n");
  FILE* Q6_File = fopen("plot_Q6.txt", "w");
fprintf(Q6_File, "#AgeGroup\tAverage(%%)\n");
  for (int age = 35.0; age <= 65.0; age += 15.0){
      double CaCounter = 0.0;
      double CaSum = 0.0;
      double OnCounter = 0.0;
      double OnSum = 0.0;
      double QbCounter = 0.0;
      double QbSum = 0.0;
      double BcCounter = 0.0;
      double BcSum = 0.0;
      double AbCounter = 0.0;
      double AbSum = 0.0;

  for (int i=0; i<=index; i++){
    if (atoi(age_group[i]) == age){
       if (strcmp(geo[i], "Canada (excluding territories)") == 0){
                  if (atof(value[i]) != 0.0){
                      CaCounter++;
                  } CaSum += atof(value[i]);
      }else if (strcmp(geo[i], "Ontario") == 0){
                  if (atof(value[i]) != 0.0){
                      OnCounter++;
                  } OnSum += atof(value[i]);
              }else if (strcmp(geo[i], "Quebec") == 0){
                  if (atof(value[i]) != 0.0){
                      QbCounter++;
                  } QbSum += atof(value[i]);
              }else if (strcmp(geo[i], "British Columbia") == 0){
                  if (atof(value[i]) != 0.0){
                      BcCounter++;
                  } BcSum += atof(value[i]);
              }else if (strcmp(geo[i], "Alberta") == 0){
                  if (atof(value[i]) != 0.0){
```

```
                    AbCounter++;
                } AbSum += atof(value[i]);
        }
    }
    }
    if (age == 65){
    printf("The averages for the age group 65-above\n");
    fprintf(Q6_File, "65-above\t");
  }else{
    printf("The averages for the age group %d-to-%d\n", age, age+14);
    fprintf(Q6_File, "%d-to-%d\t", age, age+14);
  }
  printf("Canada: %.2f%%\n", CaSum/CaCounter);
    printf("Ontario: %.2f%%\n", OnSum/OnCounter);
    printf("Quebec: %.2f%%\n", QbSum/QbCounter);
    printf("British Columbia: %.2f%%\n", BcSum/BcCounter);
    printf("Alberta: %.2f%%\n", AbSum/AbCounter);

    fprintf(Q6_File, "%.2f\n", (CaSum)/(CaCounter));
  }


// Question 2 ----------------------------------------------------
 printf ("\nQuestion 2 --- Highest Average ---\n");
 // Declare variables for highest and lowest average
 double max, min;

// Call function to determine the highest and lowest average
 min_max (ontario_ave, quebec_ave, bc_ave, alberta_ave, &min, &max);

// Check which province has the highest average and print it
 if (max <= ontario_ave)
    {
      printf ("Ontario has the highest percentage %.2lf%%\n", max);
    }
 else if (max <= quebec_ave)
    {
      printf ("Quebec has the highest percentage %.2lf%%\n", max);
    }
 else if (max <= bc_ave)
    {
      printf ("British Columbia has the highest percentage %.2lf%%\n", max);
    }
```

```
 else if (max <= alberta_ave)
   {
     printf ("Alberta has the highest percentage %.2lf%%\n", max);
   }

// Check which province has the lowest average and print it
 if (min >= ontario_ave)
   {
     printf ("Ontario has the lowest percentage %.2lf%%\n", min);
   }
 else if (min >= quebec_ave)
   {
     printf ("Quebec has the lowest percentage %.2lf%%\n", min);
   }
 else if (min >= bc_ave)
   {
     printf ("British Columbia has the lowest percentage %.2lf%%\n", min);
   }
 else if (min >= alberta_ave)
   {
     printf ("Alberta has the lowest percentage %.2lf%%\n", min);
   }

// Print a new line for formatting purposes
 printf ("\n");


// Question 3 ---------------------------------------------------
 // Base string for print statement for provinces below and above national
average
  printf ("\nQuestion 3 --- Provinces Below and Above National Average
---\n");
 char avebelow[MAX_ARRAY] = "Provinces below national average: \n";
 char aveabove[MAX_ARRAY] = "Provinces above national average: \n";

// Compares which one is below and above and appends to base string
 if (canada_ave >= ontario_ave)
   {
     strcat (avebelow, "Ontario\n");
   }
 else
   {
     strcat (aveabove, "Ontario\n");
```

```
    }
  if (canada_ave >= quebec_ave)
    {
      strcat (avebelow, "Quebec\n");
    }
  else
    {
      strcat (aveabove, "Quebec\n");
    }
  if (canada_ave >= bc_ave)
    {
      strcat (avebelow, "British Columbia\n");
    }
  else
    {
      strcat (aveabove, "British Columbia\n");
    }
  if (canada_ave >= alberta_ave)
    {
      strcat (avebelow, "Alberta\n");
    }
  else
    {
      strcat (aveabove, "Alberta\n");
    }

// Prints strings
  puts (avebelow);
  puts (aveabove);

// Question 4 ------------------------------------------------
  printf ("\nQuestion 4 --- Yearly Averages Min Max ---\n");

  double diabetes_rates[7][4];
  // Loop through the data array to extract diabetes rates for the current
year and province
  for (int year = 2015; year < 2022; year++)
    {
      int row = year - 2015;
      double ontarioCounter = 0.0;
      double ontarioSum = 0.0;
      double quebecCounter = 0.0;
      double quebecSum = 0.0;
```

```
    double bcCounter = 0.0;
    double bcSum = 0.0;
    double albertaCounter = 0.0;
    double albertaSum = 0.0;

    for (int i = 0; i < index; i++)
{
  if (atof (ref_date[i]) == year)
    {     // Check if the current record is for the current year
      if (strcmp (geo[i], "Ontario") == 0)
  {
    if (atof (value[i]) != 0)
      {
        ontarioCounter++;
      }
    ontarioSum += atof (value[i]);
  }
      else if (strcmp (geo[i], "Quebec") == 0)
  {
    if (atof (value[i]) != 0)
      {
        quebecCounter++;
      }
    quebecSum += atof (value[i]);
  }
      else if (strcmp (geo[i], "British Columbia") == 0)
  {
    if (atof (value[i]) != 0)
      {
        bcCounter++;
      }
    bcSum += atof (value[i]);
  }
      else if (strcmp (geo[i], "Alberta") == 0)
  {
    if (atof (value[i]) != 0)
      {
        albertaCounter++;
      }
    albertaSum += atof (value[i]);
  }
    }
}
```

```
      // Calculate the average diabetes rate for each province and store in
the 2D array
     diabetes_rates[row][0] = ontarioSum / ontarioCounter;
     diabetes_rates[row][1] = quebecSum / quebecCounter;
     diabetes_rates[row][2] = bcSum / bcCounter;
     diabetes_rates[row][3] = albertaSum / albertaCounter;
   }
// Declare arrays to store the minimum and maximum diabetes rates for each
year and their corresponding provinces
 double min_each_year[7];
 double max_each_year[7];
 int index_max_each_year[7];
 int index_min_each_year[7];
// Declare an array to store province names for printing the results
 char province_name[][30] =
   { "Ontario", "Quebec", "British Columbia", "Alberta" };

 // Calls function to determine the max and min value and appends to array
 MinMax2dArray (diabetes_rates, 7, min_each_year, max_each_year,
    index_min_each_year, index_max_each_year);

 // Print the results
 for (int i = 0; i < 7; i++)
   {
      printf("Year %d\nLowest Percentage: %s at %.2f%%\nHighest Percentage:
%s at %.2f%%\n\n",
  2015 + i, province_name[index_min_each_year[i]], min_each_year[i],
  province_name[index_max_each_year[i]], max_each_year[i]);
   }
}
```

GNUPLot Code

Figure 1.0 - Question 5:

```
# Set the title of the plot
# Set the title of the plot
set title "Canada Diabetes Averages from 2015 to 2021"

# Set the label for the x-axis
set xlabel 'Years'

# Set the label for the y-axis
set ylabel 'Diabetes Percentage (%)'

# Set the position of the legend (key) to the left, centered horizontally,
# and aligned to the top of the plot, with the labels in reverse order
set key Left center top reverse

# Turn on the grid
set grid

# Set the range of the y-axis to go from 8 to 14
set yrange [8:14]

# Plot the data from the "plot_Q5.txt" file, using the first column for the
x-axis (years),
# and using the second column for the average of other provinces
# The title of the plot is set to "{/:Bold Canada excluding territories}"
# The linewidth and lincolor of national average is set to 7 to make it
more spot on
plot 'plot_Q5.txt' using 1:2 title "{/:Bold Canada excluding territories}"
linewidth 7 linecolor 7 w lines, \
            '' using 1:3 title "Ontario" linewidth 2 linecolor 2 w lines,\
            '' using 1:4 title "Quebec" linewidth 2 linecolor 9 w lines,\
            '' using 1:5 title "British Columbia" linewidth 2 linecolor 4 w
lines,\
            '' using 1:6 title "Alberta" linewidth 2 linecolor 6 w lines
```

Figure 2.0 - Question 6:

```
# Set the title of the plot
set title "Canada Diabetes Averages From 2015 to 2021 Within Three Age
```

```
Groups"

# Set the position of the legend (key) to the left, centered horizontally,
# and aligned to the top of the plot, with the labels in reverse order
set key Left center top reverse

# Set the fill style for the boxes to solid
set style fill solid

# Set the width of the boxes to 0.9 times the default width
set boxwidth 0.9

# Turn on the grid
set grid

# Set the range of the y-axis to go from 0.5 to 20
set yrange [0.5:20]

# Set the label for the y-axis
set ylabel "Diabetes Averages (%)"

# Set the label for the x-axis
set xlabel "Age Group"

# Plot the data from the "plot_Q6.txt" file, using the second column for
the heights
# of the boxes and using the labels in the first column for the x-axis
ticks with a title
plot 'plot_Q6.txt' using 2:xtic(1) with boxes title "{/:Bold National
Average %}" linecolor 3, \
        'plot_Q6.txt' using 0:($2+.5):(sprintf("%3.2f%%",$2)) with labels
notitle
# Add labels to each box with the value of the height, formatted as a
percentage
# The labels are shifted upwards by 0.5 units
```