CMSC110 Introduction to Computing

Lab#8: Visualizing data
Week of October 24, 2016

Today, we will learn and practice the basics of data visualization. We will learn how to read data from a data file, store it in an array, and then visualize it. Please read the first three steps below before doing anything.

1. **Creating a data file:**
   The data file is normally either a plain text (.txt) comma-separated file (.csv) of data. For example, consider the following data:

   ```
   1992,108
   1996,101
   2000,92
   2004,103
   2008,110
   2012,104
   2016,121
   ```

   It represents the total medals won by Team USA in each of the Olympics from 1992 to 2016. You can enter this data in a plain text file (using the Notepad application in Windows, for example). Let's say the file is called **USAOlymipcs.txt**

2. **The Sketch Folder:**
   Next, you need to save or move the data file in your Sketch folder. First, create a new, preliminary/skeletal sketch:

   ```
   void setup() {
     size(500, 500);
     background(255);
   } // setup()

   void draw() {

   } // draw()
   ```

   Name the sketch **MedalsViz**

3. **The Data Folder:**
   Next, go to the MedalsViz sketch folder and in it create a new folder called, **Data**. Then copy the

data file **USAOlympics.txt** into it.

4. **Reading data into an array:**
Modify your sketch as shown below:

```
int[] year; // an array storing the years of the medals
int[] medals; // the # of medals USA won in a given year

void setup() {
   size(500, 500);
   background(255);

   // Input the data from data file
   String[] lines = loadStrings("USAOlympics.txt");

   // create the two data arrays
   year = new int[lines.length];
   medals = new int[lines.length];

   // Convert data as strings into int values
   // and populate data arrays
   for (int i=0; i < lines.length; i++) { // split
      String[] pieces = split(lines[i], ",");
      year[i] = int(pieces[0]);
      medals[i] = int(pieces[1]);
  }
} // setup()

void draw() {

} // draw()
```

5. **Print out the data:**
To make sure that the data Is being input correctly, write the following loop at the end of setup() to print out the two data arrays:

```
for (int i=0; i < year.length; i++) {
   println(year[i] + ": " + medals[i]);
}
```

6. **Design a plot – Bar Graph:**
Now that we know the data is sitting in the data arrays, we can start to plot it. First let's draw a bar graph of this data. Modify the draw() function as shown below:

```
void draw() {
  float minX = width*0.05;  // Plot dimensions
  float maxX = width*0.95;
  float minY = height*0.05;
  float maxY = height*0.95;

  float N = year.length;
  float barSpace = (maxX-minX)/N;
  float barW = barSpace*0.9;

  float x = minX;
  for (int i=0; i < N; i++) {
    // plot bar for year[i] with medals[i]
    float h = map(medals[i], 0, 200, 0, maxY-minY);
    float y = maxY-h;
    fill(0);
    rect(x, y, barW, h);

    x = x + barSpace;
  }
} // draw()
```

Experiment with this code a bit. What happens if you change the parameters passed to `map`? What about the other calculations? By experimenting with the code, you can understand it more.

7. **Design a plot – line graph:**

Next, modify the bar graph code to the one below:

```
void draw() {
  background(255);
  float minX = width*0.05;  // Plot dimensions
  float maxX = width*0.95;
  float minY = height*0.05;
  float maxY = height*0.95;

  float N = year.length;
  float barSpace = (maxX-minX)/N;

  float x = minX;
```

```
  noFill();
  beginShape();
  for (int i=0; i < N; i++) {
    // plot bar for year[i] with medals[i]
    float h = map(medals[i], 0, 200, maxY-minY, 0);
    float y = h;

    vertex(x, y);

    x = x + barSpace;
  }
  endShape();
} // draw()
```

Once again, experiment. How do these lines of code produce the output you see?

8. **Drawing annotations:**
   So that the plot obtained above is legible, you can add the following at the end of draw():

```
  x = minX;
  for (int i=0; i < N; i++) {
    float h = map(medals[i], 0, 200, maxY-minY, 0);
    float y = h;
    fill(0);
    text(medals[i], x, y-20);
    text(year[i], x, y+20);
    stroke(255, 0, 0);
    line(x, y-20, x, y+20);
    println(year[i], x, y-20);
    x = x + barSpace;
  }
```

9. **Creating your own dataset:**
   In the remainder of this lab we would like you to select a specific data set and discuss with the instructor possible ideas for visualizing the data. As a part of this exercise, you should create a skeletal sketch (so you have a folder) and in it place the data set in the Data folder.

Lab 8: Self Assessment

The questions below pertain to essential knowledge you require to implement a visualization of any data set. Based on what you have learned in this lab, and from the readings in the text, answer the questions below. In case you are stuck, please consult your instructor or a TA.

1. Write commands to define two arrays. (Don't worry yet about putting data in the arrays.)
    a. **state** -an array of strings to store the names of US states;
    b. **population** -an array of integers to store the population of each state.

2. Next, suppose the data file **USpopulation.txt** contains lines of the form:

```
...
Ohio,11613423
Pennsylvania,12787209
...
```

    That is, the name of the state, followed by its population. Write commands to input/read the entire data file in an array of strings called, **lines**:

3. Draw below a picture of what the array **lines** will look like (use the snapshot of data above as your example):

4. Next, describe (or write the Processing command) how you will find out the number of states (or entries) in the **lines** array:

5. Write commands to create the two arrays, **state** and **population**, defined in (1) above, to store the entries from the data file:

6. Write the commands to parse the data input into the **lines** array and store it in the arrays **state** and **population**:

7. Write commands to output the contents of the **state** and **population** in the Console window:

8. Write commands to print out the minimum and maximum populations:

```
int minPop =


int maxPop =
```

9. Write a command to show how you would transform the value of a state's population (say **population[i]**) into a proportional value between 0 and 400:

10. Write down what you think would be an appropriate visual form to visualize this data set?