CS/INFO 3300; INFO 5100
Homework 3
Due 11:59pm Wednesday, September 14

Goals: Practice using d3 to create SVG elements and set their aesthetic properties. Recognize the effect of data transformations through direct data changes and through scale functions.

Your work should be in the form of an HTML file called index.html with one <p> element per problem. Wrap any SVG code for each problem in a <svg> element following the <p>.

For this homework we will be using d3.js. **In the <head> section of your file, please import d3 using this tag: <script src="https://d3js.org/d3.v7.min.js"></script>**

Create a zip archive containing your **HTML file and all associated data files** and upload it to CMS before the deadline. Submissions that do not include data files may be penalized. Your submission will be graded using a Python web server run in a parent directory containing your zip file contents (e.g. server started in `~/student_hw`, with your homework at `~/student_hw/your_netid/hw3/index.html`) – be sure that it works.

1. In your HTML, please create a **400x400 pixel SVG element**. Then, select it using `d3.select()` in the `<script>` section of your code. Unlike in HW2 where you drew things by hand, in this problem you are going to **use `.append()` and `.style()` functions** to build and decorate this canvas. Please use d3 functions to **create the following elements** in your canvas:

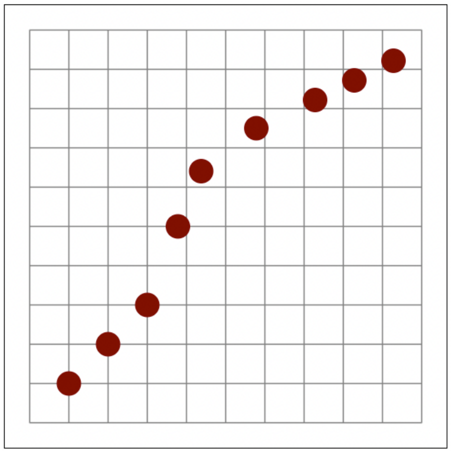
- A `<text>` **element** with the word "Cornell!" centered in the exact middle of the SVG canvas. Use `.attr()` to locate it. You are welcome to use text-anchor or adjust the position manually to center it. The <text> element should be styled to use a dark red 20px Verdana typeface.

- A `<circle>` **element** at `(200,200)` with a 5px radius so that we can verify that the text is correctly centered. Please give it a light blue `fill` color and no `stroke`. It should appear *behind* the <text> element.

- **Three (3)** `<rect>` **elements** located in the white space around the text. They should be no larger than 50px x 50px. Give each of them a different `stroke` color and `fill` color. No two rectangles should overlap or be the same size. Make sure that the colors you choose will make them stand out to the grader.

(see next page)

2. In HW2 you reproduced a plot from scratch using SVG. Now **create the same plot again**, but this time use **d3 functions to create it programmatically in a `<script>` tag**. While it should resemble the example image to the right, you don't need to recreate it exactly, so long as your point and line positions are correct.



Create a **360x360 pixel SVG element in HTML**. Use a CSS style to give the canvas a **1px solid black border**. The main plot region, excluding labels, should be a square 320x320 pixels in size, running from (20,20) to (340, 340). Reserve the remaining pixels as padding for the labels.

On the last page of this assignment, we have included a code version of the dataset. Go ahead and copy it into your `<script>` tag. First create x and y **scale functions** that map from data coordinates to SVG pixel coordinates, using the same minimum and maximum values as the chart *domain* (0 to 10 for both axes). Remember to account for the "padding" pixels when determining the *range* of pixel positions. If things behave oddly, make sure that you are following the specific syntax that `.domain()` and `.range()` expect.

Next, create the **grid of lines for your chart**. While there is a way to make gridlines using d3 axes, please **manually create gridlines using a `for` loop**. You should create one horizontal line and one vertical line for each number between 0 and 10 (inclusive) in a grey color. Use d3 functions and your scales to `.append`, set `.attr`, and `.style` the lines

Now, add **circles** for each point with positions determined by your scales. You don't need to use a data join; it's fine if you just create circles one-by-one in a `data.forEach` loop. Circles should have a radius of 10px and have a dark red color.

Data to copy into your code for #2:

```
data = [{"x":1.0 ,"y":1.0},
        {"x":2.0 ,"y":2.0},
        {"x":3.0 ,"y":3.0},
        {"x":3.8 ,"y":5.0},
        {"x":4.4 ,"y":6.4},
        {"x":5.8 ,"y":7.5},
        {"x":7.3 ,"y":8.2},
        {"x":8.3 ,"y":8.7},
        {"x":9.3 ,"y":9.2}];
```