Lab 9: Scatterplot Matrix
with Coordinated Multiple Views (CMV)

The scatterplot matrix technique (SPLOM) is a powerful tool for the multidimensional dataset. It shows scatterplots of each pair of data dimensions, laid out in a grid:

![Scatterplot Matrix Example](image)

Each dimension/variable/feature/column is represented by both a row and a column in the grid (matrix). The cell corresponding to a given pair of dimensions contains a scatterplot of the data using the pair of dimensions as axes. On its own, this visualization can make interesting relationships between dimensions clear, but interaction can make it much more powerful.

Overview:
For this lab, you will **add a coordinated multiple views interaction capability**. Specifically, highlighting a point in one scatterplot with the mouse cursor or highlighting several by drawing a rectangle around them will cause the same points to be highlighted in all the other scatterplots.
Goals:
The following diagram explains the support code, which uses a model-view-controller architecture to allow communication between several visualizations (the scatterplots), which together comprise a coordinated multiple views interface. There is a Controller class that functions as a headquarters for distributing messages. The SPLOM has its own derived version SPLOMController (implemented in Controller.pde). When the program is started, the controller is responsible for creating all the different views. In this case, that means creating the scatterplots, which are instances of AbstractView (specifically its subclass ScatterplotView). When the controller creates these view objects, each one is given a link back to the Controller (via the setController method).

In the the MVC architecture, when a view detects interaction, it creates a message explaining what happened and sends that to the controller. For our purposes, this means creating a Message object which carries an array of Condition objects that together specify which points should be highlighted. The model portion of this MVC implementation is simple data storage and does not need any commentary, but effectively resides within variables.pde.

Task:
The support code includes an implementation of SPLOM, so you don’t need to code it at all! The message passing infrastructure is already available. You need only to change the class that represents the actual scatterplots (ScatterplotView) so that it sends Messages to the Controller about points that are selected. The other scatterplots will then receive the message and highlight appropriately. You will edit two functions of the ScatterplotView class, hover and handleThisArea, which deal with hovering over a point and selecting a rectangle respectively. The comments in the support code explain what code you need to write.
Requirements:
1. Watch the demo to see how the final result should work.
2. Download the support code and review it alongside the diagram above to understand how the message passing works.
3. Fill out the two functions in ScatterplotView with missing elements (denoted by comments)
   a. Add the code to send the message about highlighting conditions to the controller in hover.
   b. Add the code to create the rectangle highlighting conditions in handleThisArea.
4. Answer the following questions (just use a piece of scrap paper or type it and save somewhere that will be easy to access when we grade the lab).
   a. As you draw a selection rectangle that starts across the top of a cell and gets dragged down to capture every point from top to bottom, what happens within the same row? Column?
   b. How about drawing the rectangle from left to right: within the same row? Column?
   c. Use one of the class plots to highlight all the points with the same class. Notice one point that isn’t as beautifully grouped with its class as the rest. Which dimension seems to be causing this distinction? Any ideas for how you might have figured that out without this handy visualization (no one correct answer)?

Grading:
We will ask you to demo your work on Thursday, November 30, 2017.