**• Describe your framework domain, with examples of possible plugins your framework could support.**

dOur domain is data collected where each data entry is associated with a county in the US. Data plugins simply read tabular data from many types of sources (xls, csv, web), and return a dictionary where sets of entries are indexed by county+state names.d

dOur framework then creates a new data-structure that associates each county with geographic coordinates. This new data-structure also has library methods that return analysis of the given data in relation to time-frames, grouping of counties into bigger structures, and different types of analyses (sum, average, stdDev, etc.).d

dThis new data-structure is then passed to the display plug-in, which uses the data-structure’s API to acquire summaries of the data in the form of raw columns, where the ith entry of a column is aligned with other columns to correspond to the same entry.d

dPossible display plugins could be simple line plots, each line belonging to a county/state. Similarly for scatter plots, bubble charts, and even choropleths by using the coordinates provided. Given our time-sensitve data, display plug-ins could have a Google Earth functionality of displaying data with scroll-bar to change frames of data at different times.d

**• Describe your decisions about the generality and speciﬁcity of your framework (i.e., domain engineering): your key abstractions, the reusable functionality your framework provides, and the potential ﬂexibility of plugins.**

dWe decided to restrict our specificity of the data to be data associated to a county in the US. We did this so we could create a hierarchy of membership amongst the inputted counties and their states, thus summarizing data per state and even at the country level would be easier.

Our abstractions assumed that all data recorded in the types of results we did would have time and county attached.

The reusable functionality the framework provides is the ability to organize very flexible data in our domain from any input supported by plugins and provide an object with a library having a multitude of data-analysis/filter tools to the display plugins to use as necessary.d

**• Describe your overall project structure: the organization of the framework and plugins into packages or projects, the location of plugin interfaces and key data structures, and how plugins are loaded.**

[Look for the file ‘Project Structure.txt’ in Team28/hw5a supplementary/Project Structure.txt, it mimics the structure of the Recitation 9 example]

We’ll use the *ServiceLoader* class to load in the Data and Display plug-ins separately. The plugins specified to be loaded are in files located in resources/META-INF/services/\*, as per SerivceLoader’s specifications.

The list of available plug-ins are displayed to the user in the GUI and they’ll choose which to use one at a time.

**• Describe your plugin interfaces, including key methods and the data structures exchanged between plugins and the framework.**

**…………………………….TO BE WRITTEN SOON………………….**