Q1. How would you implement the complete set of tasks in HW3-PC using a single MapReduce job? Note that your program should emit outputs for each task into a separate file.

I think the easiest way to implement the complete set of tasks in HW3-PC using a single MapReduce job would be to implement a MultipleOutputs in the Job which allows for a mapping of each output to a key that we could create for each of the different outputs desired. This would mean that in the mapper we would have to parse all of the required information so that in the multiple reducers we can deal with the data however is needed for the output to the final file.

Q2. Suppose that the U.S. Census Bureau has introduced a new scheme to disseminate demographic data. In addition to releasing the entire dataset as a batch at the end of a census cycle, the Census Bureau is now providing changes to the population (number of new births or deaths for each census block for a particular time period) as a continuous stream of updates. How would you extend your solution to support this new scheme in order to provide a more up-to-date view of the U.S. population?

One way I would incorporate this is to show how the US population is changing over time by plotting these changes and storing them to a local file that keeps a count of the total population and then the birth and deaths on a per-state-basis. This would lead to interesting looks to see which states are have consistent addition and loss of life and which have a more consistent life change. Also, this might lead to health problems that certain states see and would also be interesting to see if different ages of people seem to be moving from state to state through looking at the number of age groups changing. Plotting this would lead to some very interesting data.

Q3. Instead of supporting a set of fixed queries as you have done in HW3-PC, you are being asked to implement a solution that can run any arbitrary query on the census dataset using the Hadoop infrastructure. How would you design your solution to cope with this requirement?

Supporting arbitrary queries sounds like a difficult task, but I think there could be a way where we can separate the data into sections such that has been done already but doing some basic analysis of the data such as creating the median for a section or something which allows for a user to take these statistics and go further.

Q4. How would you extend your current solution to identify locations that are most similar to each other and also those that are most dissimilar to each other? Your similarity measures may include distributions relating to: age, gender, etc

I think the easiest way would be to compute a multiple linear regression that took in as many factors such as median house cost, population size, average population age, and many other social factors. From this data we can then try to learn which states are more similar based upon each of these factors through using Spark and after training our model we can start to include other factors to see which factors are able to create similarities and dissimilarities between states. This would be interesting from a data standpoint to see which factors are important for looking at how states are similar and would help in people looking to move or go to another state. Also, this would allow multiple people to change which factors they valued more and create a more robust field.

Q5. Consider the case where two additional datasets has been made available to you. The first dataset includes information about migration patterns (alongside demographics such as age, gender, and educational levels) into and out of a state. The second dataset includes economic data such as number of new jobs that were created, the sectors that these jobs were created in, the average pay, educational levels requirements for jobs, etc. Describe how you will design a framework that allows you to make reasonably accurate projections about the expected population levels in a particular state. Assume that you have census, migration patterns, and economic data for 1990, 2000, and 2010. In the case of migration patterns and economic data assume that you also have yearly data for the past 10 years.

This is a very interesting question as it would be interesting to see which factors actually influence movement. The past couple years it appears we have seen a lot of movement from major areas such as the Bay Area and Seattle to Midwest locations for some obvious pricing reasons, but it would be interesting to see if there could be a way to create breakpoints. The first thing would be to create a MapReduce job that for each person that moved we can look at their data and see how the changes from each state for leaving and each state for arriving. Once we have this data we can use Spark to attempt to train the model looking at the data based on this data on where they moved to and from to see if we can create a graph so that given population distribution we can attempt to predict how the next year will go. This would be how I would attempt to do this and there are many areas that could be fine tuned such as looking at different possible factors and seeing if leaving these out lead to a better model which can be done from the necessary testing of the model.