Below you will find tasks we have written to assess how you approach data. These tasks are written to allow some judgment in how you approach them. That is, you should think how best to structure the data, analyze it, and communicate your approach and findings. We are not looking for state-of-the-art statistics for these exercises. Graphs are often very effective, and so be encouraged to use them wherever they could be helpful.

Please submit your code for both tasks and indicate how long it took for you to answer each question. We would like to see how efficiently you write programs in terms of organization and commenting.

Thank you for taking the time to work on these tasks. Have fun, and good luck!

Task 1: Emergency Department Visits

In this task, you will be using made-up data of the flow of patients in an emergency department (ED). Physicians work in shifts, in which they begin work at a set time and stay until they discharge their patients (usually past the end of shift). Patients arrive and are immediately assigned to a physician, unless if the physician has not started his or her shift yet.

In the dataset test_data.txt, you will see comma-separated data in which each row represents a patient visit. The variables are as follows:

1. `visit_num`: Row identifier for the patient visit
2. `phys_name`: Physician
3. `shiftid`: String variable denoting the date and beginning and end times of the physician's shift. If the shift spans midnight, the date corresponds to the beginning time.
4. `ed_tc`: Date and time of patient arrival to ED
5. `dcord_tc`: Date and time of patient discharge order
6. `xb_lntdc`: Measure of expected log length of stay, where length of stay is the difference between `dcord_tc` and `ed_tc`, based on patient demographics and medical conditions (you can think of this as "patient severity")

Using a statistical program, perform the following tasks:

1. Some patients may arrive before their physician's shift starts and therefore would have to wait. Other patients may be discharged after their physician's shift ends (and the physician would have to stay past the end of shift). What percentages of visits fall in these categories?
2. Describe hourly patterns of patient arrivals and the average severity of these patients.
3. Create a dataset recording the "census," or number of patients under a physician's care (patients who have arrived and have not yet been discharged), during each hour of a physician's shift. How does the census vary with time relative to end of shift? You may assume that physicians stay at most 4 hours past end of shift and that all physicians see at least one patient in their shift. Hint: You will need to transform the text in `shiftid` into numerical shift beginning and end times capturing both date and hour.

4. Create a dataset in which each observation represents an hour in the two-month period that the data span. In each hour, calculate the number of physicians on shift (those who are between the start and end times of their shifts), the number of physicians still present because they have not discharged all their patients yet, and the number of patients arriving during that hour. You may assume that each physician saw at least one patient during each shift, and that physicians stay at most 4 hours past end of shift. Describe the relationship between number of physicians on shift, number of physicians present, and number of patients arriving, using a graphical approach, a regression-based approach, or both. (Optional, especially if you do Task 2)
5. Which physician appears to be the fastest at discharging patients? You should answer this with a regression of log length of stay. Discuss how you thought about which variables you control for. You may show results graphically. (Optional, especially if you do Task 2)

Task 2: MySQL Employees Database (Optional for those without SQL knowledge)

This task tests your ability to use MySQL to access data and to perform some analyses on the data. You will first need to install from the MySQL, which is available free at http://www.mysql.com/downloads/.[1] As documented on www.mysql.com, MySQL comes with the "Sakila" database of movie rentals as part of the installation.

After accessing the data from MySQL, you may perform subsequent analyses on another statistical program like Stata or SAS, but if you can perform analyses more efficiently in MySQL you will get extra credit. If you have not had much experience with SQL, this will be taken into consideration.

After loading the data, answer the following questions:

1. Which are the most popular actors in each film category?
2. Which actors have worked with the greatest numbers of other actors in the set of films observed in the data?

[1] When installing, be sure to include MySQL Workbench. Although you may find documentation on how to download MySQL at http://www.mysql.com, it is not necessary to read through this, and the overall installation process should not take more than 30 minutes.