Stat 604
Assignment 15 - SAS

This assignment primarily reinforces techniques covered in the SAS lessons 15 and 16 but you may also need to employ techniques learned in previous lectures.

1. The first part of this assignment involves reading a complex raw data file that we are going to treat as formatted input. This is the type of file for which advanced text editing software like Notepad++(for Windows) or Text Wrangler (Mac) can be extremely useful. Orion Star is in the process of acquiring Pegasus Athletics. We do not yet have access to the Pegasus database but a file named Pegasus.dat that contains the Pegasus Organization chart has been posted on eCampus. The acquisition team has asked you to create a temporary data set of employees from the Pegasus file. The level values are in fixed positions as shown in the file layout table below. After the level on each line of data is a string containing the job title and employee name, followed by the employee's annual salary. You should open the file and become familiar with the data before trying to write your program.

Position	Variable
1-8	Level 1
10-17	Level 2
19-26	Level 3
28-35	Level 4
37-44	Level 5
46-53	Level 6
??-100	Employee info
106-115	Salary

2. Create a data set from this file that contains the variable Level as a single digit numeric variable, along with Job_Title, Employee_Name and Salary. I know several of you will be able to think of a simpler way to get to the results than what is being required, but the objective of this exercise is to give you some experience working with fixed width raw data files that have no delimiters. Acceptable solutions to this exercise must have the following:
    a. Multiple input statements with pointer controls.
    b. No warnings or data errors in the log.

   The PC SAS log from this step is shown below:

```
NOTE: 424 records were read from the infile FIXEDIN.
      The minimum record length was 115.
      The maximum record length was 115.
NOTE: The data set WORK.PEGASUS has 424 observations and 4 variables.
```

   For this assignment you will need to rely heavily on all the material covered in SAS Lesson 15. I am treating this as data with "mixed record types" so the discussion beginning with slide 60 should be very relevant. For those of you who like to take shortcuts, be aware that the first group of slides in this section cover methods that do not quite achieve the desired results. The difference between this file and the example in the lecture is that the layout is actually different for each record type in our data. One possibility for dealing with these data would be to create one or more temporary character variables for each of the levels in the first input statement. Then a series of conditional statements could be used to test which level the current record matches and execute the appropriate input statement. You do not have to worry about ordering the conditional statements for "computational efficiency" but you should not execute remaining conditions once the test condition is true. You might find it helpful to create the data set using only the first input statement at the beginning to see how the values are being read. This should help you understand how to construct the logic for the remainder of the step. I also recommend that you wait until you are finished writing and debugging your code before you discard any variables. For simplicity, I recommend reading in the employee info altogether as one variable then use the character manipulation functions of your choice to parse out the title and name. Remember that length and

informat/format are not the same thing especially for numeric variables. Length is the number of bytes of storage (almost always 8 for numeric) and informats specify the number of digits to be read.

3. Use the FREQ procedure to identify possible inconsistent values of Job Title from the raw data. While there are a number of jobs that only have one employee, single-employee values should be reviewed to ensure there is not a similar job title that is only different due to a typographical error.
4. Use the UNIVARIATE procedure to validate Salary values.
5. Use a PRINT procedure to list the Pegasus employees that you feel need further investigation based on irregular salary values as indicated by the results of the UNIVARIATE procedure in the previous step. The second title line should read "Salary Values to be Investigated". We can assume that the Chief Executive Officer would be the highest paid employee in the company and that the lowest salaries would not be extremely below average. You must include this table in the output you submit to WebAssign but it has been removed from the sample output on eCampus so that you will have the opportunity to analyze this on your own. The professor's output has six observations. If you feel there should be more or less, write a justification in the comments of your program for this step. Credit will be given for any work with valid justification.
6. Write a data step that reads the recently created data set, cleans it up, and writes out the clean data to another SAS data set. Write statements that will replace incorrect job title values identified in step 3 with the standard values. When there are similar values, we will assume that the standard is the value with the highest frequency. If there is a tie, look for the pattern used in similar values in the data and clean up your data to match the overall pattern. Write your conditional assignment statements in such a way that other conditions will not be tested once a true statement is encountered.
7. Use a FREQ report to show the number of different job titles in the cleaned data set. NOTE: You can use the noprint option on a TABLES statement to prevent the creation of the actual frequency table. Ensure your output matches the output on eCampus.
8. Orion will probably not retain Executive officers or temporary employees. Print a listing of these employees from your cleaned data set based on job titles that contain Chief, Director, or Temp. Also, include the Vice President in your list. Your output should match the output posted on eCampus. You may want to refer to the documentation in SAS Help for more information on using BY and ID with the PRINT procedure.
9. Ensure your program contains the required documentation and housekeeping steps.
10. Convert the program and log to PDF files and submit them to WebAssign along with your SAS output.