```
/*********************************************************************************
 Program Name: jblubau1_hw15_script
 Date Created: 11/29/2016
 Author: Joseph Blubaugh
 Purpose: Homework Assignment 15
********************************************************************************** */

libname datadb 'C:\Users\Joseph\Projects\learning\Statistics\STAT_604\Materials' access=readonly;
libname output 'C:\Users\Joseph\Projects\learning\Statistics\STAT_604\Data';

filename outpdf 'C:\Users\Joseph\Projects\learning\Statistics\STAT_604\Homework\jblubau1_hw15_output.pdf';

* 2) Import and parse data;
data peg;
        infile 'C:\Users\Joseph\Projects\learning\Statistics\STAT_604\Materials\pegasus.dat';
        input @1 all $115. @;
                if substr(all,1,8) = '(Level1)' then
        input @7 Level $1.
                  @10 Details $96.
                  @106 Salary dollar10.;
                if substr(all,1,8) = '(Level2)' then
        input @16 Level $1.
                  @19 Details $87.
                  @106 Salary dollar10.;
                if substr(all,1,8) = '(Level3)' then
        input @25 Level $1.
                  @28 Details $74.
                  @106 Salary dollar10.;
                if substr(all,1,8) = '(Level4)' then
        input @34 Level $1.
                  @37 Details $69.
                  @106 Salary dollar10.;
                if substr(all,1,8) = '(Level5)' then
        input @43 Level $1.
                  @46 Details $60.
                  @106 Salary dollar10.;
                if substr(all,1,8) = '(Level6)' then
        input @52 Level $1.
                  @55 Details $51.
                  @106 Salary dollar10.;
        Job_Title = substr(Details, 1, find(Details, "(")-1);
        Employee_Name = compress(substr(Details, find(Details, "(")+1, find(Details, " )")),")");
        drop all details;
run;

ods pdf file=outpdf;

* 3) Use FREQ to identify inconsistent job titles with one employee;
proc freq data=peg;
        tables Job_Title;
        title Analysis of Pegasus Employee Data for Clean Up;
        title3 Frequency Report of Job Title;
run;

* 4) Use UNIVARIATE to validate salaries;
proc univariate data=peg;
        var salary;
        title2 Analysis of Salary Values;
        title3;
run;

* 5) Salaries that are suspicious;
proc print data=peg noobs;
        where salary > 500000 or salary < 1000;
        title2 Salary Values to be Investigated;
run;

* 6) Clean up Job Titles;
data output.pegasus;
        set peg;
        if job_title = 'Accountant i' then job_title = 'Accountant I';
        else if job_title = 'Accountant ii' then job_title = 'Accountant II';
        else if job_title = 'Accountant iii' then job_title = 'Accountant III';
        else if job_title = 'Warehouse Assistant i' then job_title = 'Warehouse Assistant I';
        else if job_title = 'Warehouse Assistant ii' then job_title = 'Warehouse Assistant II';
        else job_title = job_title;
run;

* 7) Verify Job Count;
proc freq data=output.pegasus nlevels;
        tables Job_Title / noprint;
        title Number of Different Jobs in Cleaned Data;
        title3;
        title3;
run;

* 8) Print listing of employees with Chief, Director, or Temp;
proc sort data=output.pegasus;
        by Level Job_Title Employee_Name;
run;

proc print data=output.pegasus;
        var Job_Title Employee_Name;
        by Level;
        id Level;
        where Job_Title like '%Chief%' or
                 Job_Title like '%President%' or
                 Job_Title like '%Director%' or
     Job_Title like '%Temp%';
        title List of Pegasus Employees to be Reviewed for Orion Positions;
run;

title;

ods pdf close;
```