



All data files are in the folder Homework 1 on SAS OnDemand. You are to write a SAS program that does all four problems below.

When you are done, you are to:

- Create a PDF of the LOG: Click the LOG tab. Find the PRINT logo  in SAS above the LOG. For Destination choose “Save as PDF”. Click the “More Settings” tab and then at the bottom checkmark “Headings and Options”.
- Create a PDF of the Results: Click the RESULTS tab. Find the PRINT logo  in SAS above the RESULTS. For Destination choose “Save as PDF”. Click the “More Settings” tab and then at the bottom checkmark “Headings and Options”.
- Name the two files LastNameAssignment1LOG.pdf and LastNameAssignment1RESULTS.pdf, respectively.
- Upload to Bb the two files.
- Bring a hard copy of the PDFs to class on the due date—it is easier for me to grade a hard copy than an online copy.

The program should have the following

- Your first and last name at the top of the program as a comment statement.
- A %LET statement that creates a SAS Macro variable called path that connects to the file Homework 1. Make sure the %LET statement corresponds to your SAS OnDemand, not mine or someone else.
- Before each Problem, type a comment statement that includes the Problem number (example: *Problem 1;)
- Include SAS code to accomplish what is requested in each problem.
- For each DATA step or PROC include at the top a comment statement that briefly tells what the DATA step or PROC does.
- In each DATA step or PROC include at least one comment that explain what specific statements do in the DATA step or PROC.
- Before each PROC PRINT include two TITLE statements that describes what is output.
 - The first title statement describes what is in the PROC PRINT—“Problem 1”, “Problem 2” is not acceptable as PROC PRINT titles. Briefly describe the data that is printed.
 - The second title statement includes your name.

The LOG should have no warnings or errors.

The output should match as closely as possible the “What the output should look like” HTML document (which is uploaded to Bb), with appropriate titles for the PROC.

Please contact me if you have any questions or if there are things you do not understand.

Problem 1: (10 points) The file *Oscars2.xlsx* contains data on previous winners of the Academy Awards (aka Oscars). The columns in the file represent, in order,

- ID (ID number)
- Year (year of the award)
- Hosts (person or persons who hosted the award show)
- best picture (movie recognized as best that year)
- best actor (male actor recognized as having the best performance in a movie that year)
- best actress (female actor recognized as having the best performance in a movie that year)

Download the file, open the file and look at it before writing code.

- a) Use PROC IMPORT to import the file and do a PROC PRINT with all variables to see what variable names SAS gave to the variables.
- b) Between the PROC IMPORT and the PROC PRINT add a DATA step where you add labels to all variables in the SAS data set created in (a), even if the label is identical to the variable name, so that the labels are the same as the column headings in the “final result” document.

Why do you have to label all variables, even if the label is identical to the variable name? It's to override a SAS quirk. When you used RENAME in PROC IMPORT, the original variable names (A, B, C, etc.) became labels for the variables. Hence, when you add the LABEL comment in PROC PRINT (which you will do in part (c)) you will see the SAS labels A, B, C, etc. for any variables you did not add labels for.

- c) Add a LABEL command to the first statement in the PROC PRINT so that the labels created in (b) are applied to the printout. These labels should match the “What the output should look like” HTML document.
- d) Finally, add two title statements to the PROC PRINT. The first title statement is an appropriate title to the output. The second title statement includes your name and the class day (Tuesday class or Thursday class)

Problem 2: (10 points) The file *MensMarathon.xlsx* contains information on World Record Men's marathon times (source: https://en.wikipedia.org/wiki/Marathon_world_record_progression). Download the file, open the file and look at it before writing code because the file includes a description of the columns below the data in the file.

- Use PROC IMPORT to import the data and do a PROC PRINT with all variables to see what variable names SAS gave to the variables. Hint: Expect to use a GETNAMES statement and a RANGE statement.
- To the PROC IMPORT in (a) add a RENAME statement to create appropriate variable names to the variables in the SAS data set. Run the PROC IMPORT and look at the Output Data tab to see that the RENAME statement worked.
- Between the PROC IMPORT and the PROC PRINT add a DATA step where you add labels to all variables in the SAS data set created in (a) so that the labels are the same as the column headings in the “final result” document. Also add in the DATA step the FORMAT statement

FORMAT _____ WORDDATE18.;

where you fill in the blank with your name for the date of the marathon, so that in the PROC PRINT you see the date of the marathon in words.

- Add a LABEL command to the first statement in the PROC PRINT so that the labels created in (c) are applied to the printout. These labels should match the “What the output should look like” HTML document.
- Finally, add two title statements to the PROC PRINT. The first title statement is an appropriate title to the output. The second title statement includes your name and the class day (Tuesday class or Thursday class)

In Problems 1 and 2 you used Excel files, which are examples of DBMS (data base management files).

In Problems 3 and 4 you use delimited text files. These are text files where variable values for each observation are separated by commas, spaces, tabs, or other symbols.

You can often (but not always) use PROC IMPORT to import delimited files. When you do so, you need to make the following adjustments to your PROC IMPORT:

In the first PROC IMPORT statement, you say DBMS = DLM where DLM stands for delimiter. In addition, with delimited text files, you need to add a DELIMITER = statement to the PROC IMPORT:

- For tab-delimited data, you add the statement DELIMITER='09'x;
- For space-delimited data, you add the statement DELIMITER=' ';
- For comma-delimited data, you add the statement DELIMITER=', ';

Problem 3: (10 points) The file *Tigers 2024 salaries.txt* contains the following information for players on the 2024 Detroit Tigers professional baseball team who made above the minimum Major League Player salary of \$740,000.

- Player name
- Player number (each player is assigned a number)
- Position (2-letter abbreviation); do not worry about what the 2-letter abbreviations mean.
- Salary
- Percentage of the payroll of the team that is paid to that player.

The file was created by “copying and pasting” from <https://www.spotrac.com/mlb/detroit-tigers/payroll/> and then deleting some columns.

Download, open and look at the file before you start writing code. What separates data values in each column: spaces, tabs or commas?

- Starting with the *Tigers 2024 salaries.txt* file, use PROC IMPORT to create a SAS data set. Expect to use a DLM = statement and a GETNAMES = statement. Also create a PROC PRINT to make sure that the data has been input correctly.
- To the PROC IMPORT in (a) add a RENAME statement so that the variables in the data set are identical to the headings in the “What the output should look like” HTML document. Run the PROC IMPORT and look at the Output Data tab to see that the RENAME statement worked.
- Between the PROC IMPORT and the PROC PRINT add a DATA step. In the DATA step add the FORMAT statement

FORMAT _____ dollar12.;

where you fill in the blank with your name for the salary variable, so that in the PROC PRINT you see a dollar sign and commas for the salary, just like in the printout.

- Add two title statements to the PROC PRINT. The first title statement is an appropriate title to the output. The second title statement includes your name and the class day (Tuesday class or Thursday class).

Add a LABEL command to the PROC PRINT. What happens to the output? Does it look the way you want it to look?

Now take out the LABEL command and rerun the PROC PRINT. Does it now look the way you want it to look?

Problem 4: (10 points) The file *Ben and Jerrys fixed.txt* contains various ice cream flavors served by the ice cream chain Ben and Jerry's and other information. Specifically, the file contains, respectively, the following 17 variables. Download, open and look at the file before you start writing code. What are used to separate variable values for each observation (spaces? tabs? commas? Something else?).

- flavor name
- portion size (g) (how many grams of ice cream were used for the later variables)
- calories
- calories from fat
- fat (g)
- saturated fat (g)
- trans fat (g)
- cholesterol (mg)
- sodium (mg)
- total carbohydrate (g)
- dietary fiber (g)
- sugars (g)
- protein (g)
- year introduced (the first year the flavor was sold)
- year retired (the last year the flavor was sold)
- content description (ingredients in the flavor of ice cream), and
- notes (additional information).

For this exercise, however, I am less concerned about renaming variables and more concerned about getting the data entered in the first place. Hence, I will not ask you to rename variables here.

a) Use PROC IMPORT to import the text file into SAS and create a SAS data set. Do not try to RENAME any variables here: Use the names that PROC IMPORT gives you. Expect to use a DBMS = command and a DLM = statement in your PROC IMPORT.

b) Use PROC PRINT with titles to print the SAS data set. The idea here is to get you to use PROC PRINT as a quick way to make sure you have read in the data correctly.

c) Next, do the following as a check. Observe that VAR14, VAR15, and VAR17, the “year introduced”, “year retired”, and “notes” variables respectively, have a lot of missing values. Run the following code changing the VAR14, VAR15, and VAR17 as needed and filling in the blank with the name of the SAS data set.

```
PROC FREQ DATA=_____;  
TABLES Var14 Var15 Var17;  
RUN;
```

Now look at the values for the VAR14 and VAR15 table. You should be getting 4-digit years as values for VAR14 and VAR15. *Are you getting 4-digit values?*

Next, read through the values in the VAR17 table. *Do there seem to be notes that are “cut off”?*

By now you probably see that you have a problem! You have years that are not 4-digit years and “notes” that are clearly cut off! What happened? More importantly, what is the fix?

HINT: If you don’t know how to fix the problem, go to pages 14-15 of the *Lecture 1* document on BB. The fix is somewhere in those pages.

d) Do the following to finish:

- Add the appropriate statement (it starts with a G) to the PROC IMPORT.
- Type /* and */ around the PROC PRINT but not the titles before the PROC PRINT, so that you “comment out” the PROC PRINT but not the titles.
- Rerun the PROC IMPORT and the PROC FREQ for the VAR14, VAR15 and VAR17 columns.

Is the PROC FREQ output now what it should be?

Before you create your PDF’s:

- Run the entire program.
- Check the LOG to see if there are any errors or warnings. Fix the program so that those errors and warnings no longer exist.
- Check the RESULTS and compare them with the “What the output should look like” document. Fix the program to match the “What the output should look like” document.

Finally, if the LOG is clean and the RESULTS are what they should be, create the PDFs and upload them.