Programming Exercises

43. *Hint:* Remember to tell SAS which variables are character.

*Answer:*

\*\* Part a);

\*\* There are three variables and 10 observations;

\*\* Part b);

DATA cancer;

INFILE 'c:\MyRawData\CancerRates.dat';

INPUT Rank Site $ Rate;*Instructor Solutions* **13**

RUN;

\*\* Part c);

PROC PRINT DATA = cancer;

TITLE 'Top 10 Cancer Sites in the United States';

RUN;

\*\* Part d);

DATA cancer;

\*\* Where x:\ is the letter of the desktop or

flash drive;

INFILE 'x:\CancerRates.dat';

INPUT Rank Site $ Rate;

RUN;

(section 2.5)

44. *Hint:* List input alone will not work for this raw data file because of missing data and embedded spaces in names of breeds.

*Answer:*

\*\* Part a);

\*\* There are five variables and 173 observations;

\*\* Part b);

DATA akc;

INFILE 'c:\MyRawData\AKCbreeds.dat';

\*\* Column input is used in every year except Y1 to

pick up the missing data;

\*\* The MISSOVER option in section 2.14 could also

solve this problem of missing data at the end of

a record;

INPUT Breed $ 1-39 Y1 Y2 45-47 Y3 50-52 Y4 55-57;

RUN;

\*\* Part c);

PROC PRINT DATA = akc;

TITLE 'Dog Breed Rankings';

RUN;

(sections 2.6, 2.9)

45. *Hint:* Pay attention to missing data and be careful to designate character data correctly.

*Answer:*

\*\* Part a);

\*\* There are 17 variables and 21 observations;

\*\* Part b);

DATA vaccines;

INFILE 'c:\MyRawData\Vaccines.dat';

INPUT Vaccine $ 1-29 Transmission $ 32-49

Incidence 51-58 Deaths 61-67 Chile $ 71-73

Cuba $ 77-79 USA $ 83-85 UK $ 89-91

Finland $ 95-97 Germany $ 101-103

SaudiArabia $ 107-109 Ethiopia $ 113-115

Botswana $ 119-121 India $ 125-127

Australia $ 131-133 China $ 137-139

Japan $ 143-145;

RUN;

\*\* Part c);

PROC PRINT DATA = vaccines;

TITLE 'International Vaccines';

RUN;

(section 2.6)

46. *Hint:* Consider informats for any nonstandard numeric data. You may also need to position the pointer before reading values for some variables.

*Answer:*

\*\* Part a);

\*\* Company name and country must be read in as

character variables. Rank can be read in as a

character or numeric variable. Sales, profits,

assets, and market value should be read in as

numeric;

\*\* Part b);

DATA bigcomp;

INFILE 'c:\MyRawData\BigCompanies.dat';

INPUT Rank Company $ 6-34 Country $ 35-52

Sales DOLLAR6. +4 Profits DOLLAR5.

+2 Assets DOLLAR8. +4 MarketValue DOLLAR6.;

RUN;

\*\* Part c);

PROC PRINT DATA = bigcomp;

TITLE "World's 100 Biggest Companies";

RUN;

(section 2.9)

47. *Hint:* Pay attention to the varying columns and missing values for the last three variables.

*Answer:*

\*\* Part a);

\*\* Crayon name, hexadecimal code, and RGB triplet must

be read in as character variables. Crayon number can

be read in as a character or numeric variable. Pack

size, year issued, and year retired should be read in

as numeric variables;

\*\* Part b);

LIBNAME sasdata 'c:\MySASLib';

DATA sasdata.crayons;

INFILE 'c:\MyRawData\Crayons.dat' MISSOVER;

INPUT Number Color $ 4-29 Hex $ RGB $ 41-55 Pack

Issued 4. +1 Retired 4.;

RUN;

\*\* Part c);

PROC PRINT DATA = sasdata.crayons;

TITLE 'Standard Crayon Colors';

RUN;

(sections 2.9, 2.14, 2.18, 2.19)

48. *Hint:* Because some of the data values contain spaces, you cannot read the data with simple list input. Be careful reading the data values for prominence which vary in length at the end of each record.

*Answer:*

\*\* Part a);

\*\* Mountain name and first year of ascent (due to

unclimbed data) must be read in as character

variables. Height (m), height (ft), and prominence

should be read in as numeric variables;

\*\* Part b);

DATA mtns;

INFILE 'c:\MyRawData\Mountains.dat';

INPUT Name $37. @40 Height\_m COMMA5. @48

Height\_ft COMMA6. Ascent :$9.

Prominence :COMMA.;

RUN;

\*\* Part c);

PROC PRINT DATA = mtns;

TITLE 'Tallest Mountains';

RUN;

(section 2.10)

49. *Hint:* For the rows of data for students, think about how to tell SAS to skip a line and remove the observation.

*Answer:*

\*\* Part a);

DATA users1;

INFILE 'c:\MyRawData\CompUsers.dat';

INPUT UserID Class $ Fname :$20. Lname :$20. /

@'email:' Email :$55. @'phone:' Phone $ Dept $;

RUN;

\*\* Part b);

PROC PRINT DATA = users1;

TITLE 'All Grid Users';

RUN;

\*\* Part c);

DATA users2;

INFILE 'c:\MyRawData\CompUsers.dat';

INPUT UserID Class $ @;

\*\* Tell SAS to skip a line so that it

can start reading on the correct line for the next

observation;

IF Class = 'Student' THEN INPUT /;

IF Class = 'Student' THEN DELETE;

INPUT Fname :$20. Lname :$20. /

@'email:' Email :$55. @'phone:' Phone $ Dept $;

RUN;

\*\* Part d);

PROC PRINT DATA = users2;

TITLE 'Faculty and Staff Grid Users';

RUN;

(sections 2.8, 2.9, 2.10, 2.11, 2.13)

50. *Hint:* Consider the size and layout of the raw data file and what options might be required to read this large data set.

*Answer:*

\*\* Part a);

DATA sff;

INFILE 'c:\MyRawData\SwineFlu2009.dat' LRECL = 300

TRUNCOVER;

\*\* Methods for input could vary, for example using the

INFORMAT statement which will work for most of the

data except the dates;

INPUT @1 ByDate 3. @14 ByCont 4. @28 Country $30.

@61 FirstCase ANYDTDTE10. @79 Apr 3. @89 May 3.

@99 June 5. @109 July 5. @119 Aug 6.

@129 Latest 6. @145 ByDate\_d 3. @155 ByCont\_d 4.

@169 FirstDeath ANYDTDTE10.

@194 May\_d 3. @204 June\_d 3. @214 July\_d 3.

@224 Aug\_d 4. @234 Sep\_d 4. @244 Oct\_d 4.

@256 Nov\_d 4. @266 Dec\_d 4.;

\*\* Part b);

LABEL ByDate = 'Sorting ID by 1st case date'

ByCont = 'Sorting ID by 1st case date

within continent'

Country = 'Country'

FirstCase = 'Date of first case reported'

Apr = '# cumulative cases on 1st day of Apr'

May = '# cumulative cases on 1st day of May'

June = '# cumulative cases on 1st day of Jun'

July = '# cumulative cases on 1st day of Jul'

Aug = '# cumulative cases on 1st day of Aug'

Latest = 'Last cumulative # of cases reported'

ByDate\_D = 'Sorting ID by 1st death date'

ByCont\_D = 'Sorting ID by 1st death date

within continent'

FirstDeath = 'Date of 1st death'

May\_D = '# cumulative deaths on 1st day of May'

June\_D = '# cumulative deaths on 1st day of Jun'

July\_D = '# cumulative deaths on 1st day of Jul'

Aug\_D = '# cumulative deaths on 1st day of Aug'

Sep\_D = '# cumulative deaths on 1st day of Sep'

Oct\_D = '# cumulative deaths on 1st day of Oct'

Nov\_D = '# cumulative deaths on 1st day of Nov'

Dec\_D = '# cumulative deaths on 1st day of Dec';

RUN;

\*\* Part c);

PROC CONTENTS DATA = sff;

RUN;

(sections 2.4, 2.8, 2.9, 2.14, 2.21)

51. *Hint:* Examine how missing values are recorded in the raw data file. Also pay attention to the lengths of the character variables.

*Answer:*

\*\* Part a);

DATA baj;

INFILE 'c:\MyRawData\BenAndJerrys.dat' LRECL = 350

MISSOVER DLM = ',' DSD;

\*\* Note that Fiber has some values equal to <1

that need to be read in as character;

INPUT Flavor :$75. Portion Cal CalFat Fat SatFat

TransFat Chol Sod Carb Fib $ Sugar Protein

YearIntro YearRetire Desc :$150. Notes :$200.;

RUN;

\*\* Part b);

PROC PRINT DATA = baj;

TITLE 'Ice Cream Flavors';

RUN;

(sections 2.4, 2.10, 2.14, 2.15)

52. *Hint:* Notice that in the Microsoft Excel file there are no variable names. If you have a mixture of 32-bit and 64-bit applications installed on your computer, then you may not be able to use PROC IMPORT to read Excel files.

*Answer:*

\*\* Part a);

LIBNAME sasdata 'c:\MySASLib';

PROC IMPORT DATAFILE = 'c:\MyRawData\Oscars.xlsx'

OUT = sasdata.oscars DBMS = XLSX;

GETNAMES = NO;

RUN;

\*\* Part b);

PROC CONTENTS DATA = sasdata.oscars;

RUN;

\*\* Part c);

\*\* Because there are no variable names in the Microsoft

Excel file, the resulting variables in the SAS data

set have non-descriptive names that do not reflect

the data;

(sections 2.16, 2.17, 2.18, 2.19)

53. *Hint:* Think about how SAS may need to move the pointer depending on if the data will be kept in the data set or deleted. What will tell SAS to stay on the line of data through multiple iterations of the DATA step?

*Answer:*

\*\* Part a);

DATA tchol1;

INFILE 'c:\MyRawData\TChol.dat';

INPUT ID Group :$9. Diff Pre Post @@;

RUN;

\*\* Part b);

PROC PRINT DATA = tchol1;

TITLE 'All Cholesterol Measurements';

RUN;

\*\* Part c);

DATA tchol2;

INFILE 'c:\MyRawData\TChol.dat';

\*\* @ will not hold the line for multiple observations

@@ will hold for further statements AND hold the

line of data;

INPUT ID Group :$9. @@;

\*\* For controls the pointer needs to move past the

cholesterol data and then delete the observation;

IF Group = 'Control' THEN INPUT +10 @@;

IF Group = 'Control' THEN DELETE;

INPUT Diff Pre Post @@;

\*\* Later in the book the topics of IF-THEN/ELSE and DO

groups will be explained which would make this code

even more efficient;

RUN;

\*\* Part d);

PROC PRINT DATA = tchol2;

TITLE 'Treatment Group Cholesterol Measurements';

RUN;

(sections 2.7, 2.10, 2.12, 2.13)

54. *Hint:* Viewing the raw data file in a simple editor such as WordPad will enable you to see the original format of the file. Opening the file in Microsoft Excel will incorrectly display the raw data. Missing values in the beginning of a data file may cause PROC IMPORT to read the data incorrectly. In the final SAS data set, the ID variable should be character and all other variables should be numeric.

*Answer:*

\*\* Part a);

PROC IMPORT DATAFILE = 'c:\MyRawData\Pizza.csv'

OUT = pizzaimport REPLACE;

\*\* Using GUESSINGROWS= will prevent the ratings data

for shrimp and eggplant from being read in as

character;

GUESSINGROWS = 120;

RUN;

\*\* Part b);

PROC PRINT DATA = pizzaimport;

TITLE 'Pizza Ratings';

RUN;

\*\* Part c);

PROC CONTENTS DATA = pizzaimport;

RUN;

\*\* Part d);

\*\* The leading zeros from the variable SurveyNum are

missing. This presents a problem because we might

not be able to tell if SurveyNum = 101 is January 1st

or October 1st. Because PROC IMPORT sees only digits

in SurveyNum, it assumes it is numeric and therefore

the leading zeros are removed. One way to keep the

leading zeros is to define the variable as character;

\*\* Part e);

DATA pizzainput;

INFILE 'c:\MyRawData\Pizza.csv' DLM = ',' DSD

FIRSTOBS = 2;

INPUT SurveyNum $ Arugula PineNut Squash Shrimp

Eggplant;

RUN;

\*\* Part f);

PROC PRINT DATA = pizzainput;

TITLE 'Pizza Ratings';

RUN;

(sections 2.14, 2.15, 2.16, 2.21)

55. *Hint:* In the SAS windowing environment, you can create a libref for permanent SAS data sets by selecting **File > Save As** and then clicking the Create New Library icon in the upper-right corner of the Save As window.

*Answer:*

\*\* Part e);

LIBNAME sasdata 'c:\MySASLib';

PROC PRINT DATA = sasdata.rushmore;

TITLE 'Presidents at Mount Rushmore';

RUN;

PROC CONTENTS DATA = sasdata.rushmore;

RUN;

(sections 2.2, 2.18, 2.19, 2.21)