37. *Hint*: Think about how SAS sorts numerals that are contained in character data.

*Answer*:

LIBNAME sasdata 'c:\MySASLib';

\*\* Part a);

\*\* You can determine the attributes of variables and

data sets interactively by viewing the data set (or

table) properties, or you can use PROC CONTENTS;

\*\* Part b);

PROC CONTENTS DATA = sasdata.crayons;

RUN;

\*\* The Color variable has a label 'Crayon name' and a length of 26;

\*\* Part c);

PROC FREQ DATA = sasdata.crayons;

TABLES Issued;

TITLE 'Number of Crayon Colors Issued per Year';

RUN;

\*\* In 1949 40 standard colors were issued;

\*\* Part d);

PROC SORT DATA = sasdata.crayons OUT = crayonsort

SORTSEQ = LINGUISTIC (NUMERIC\_COLLATION = ON);

BY RGB;

RUN;

\*\* Part e);

PROC PRINT DATA = crayonsort NOOBS;

VAR Color RGB;

TITLE 'Crayon Colors By Color Code';

RUN;

(sections 4.3, 4.4, 4.5)

38. *Hint*: Consider using IF-THEN/ELSE statements to distinguish between the two layout patterns.

*Answer*:

\*\* Part b);

PROC FORMAT;

VALUE mth 1 = 'Jan'

2 = 'Feb'

3 = 'Mar'

4 = 'Apr'

5 = 'May'

6 = 'Jun'

7 = 'Jul'

8 = 'Aug'

9 = 'Sep'

10 = 'Oct'

11 = 'Nov'

12 = 'Dec';

RUN;

\*\* Part a);

DATA \_NULL\_;

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

INPUT ID FirstName $ 6-16 LastName $ 20-29

Address $ 34-55 City $ 59-84 State $ Zip $ 94-98

Amount Month;

\*\* Parts b) and c);

FILE 'c:\MyRawData\Receipts.dat' PRINT;

TITLE;

IF FirstName ~= ' ' THEN

PUT \_PAGE\_ 'TO:' +1 FirstName LastName /

@5 Address /

@5 City +(-1) ',' +1 State Zip //

'Thank you for your support! Your donations'/

'help us to save hundreds of cats and dogs'/

'each year.'//

'Donations to Coastal Humane Society' /

'(Tax ID: 99-5551212)'/

Month mth. +1 Amount DOLLAR7.2;

ELSE IF FirstName = ' ' THEN

PUT Month mth. +1 Amount DOLLAR7.2;

RUN;

(sections 4.8, 4.9)

39. *Hint*: Find a way to group the month data into quarters without using a DATA step.

*Answer*:

LIBNAME sasdata 'c:\MySASLib';

\*\* Part a);

\*\* You can determine the attributes of variables and

data sets interactively by viewing the data set (or

table) properties, or you can use PROC CONTENTS;

PROC CONTENTS DATA = sasdata.gas;

RUN;

\*\* The GasPrice variable has a label 'U.S. average

price of unleaded regular gasoline (per gallon)' and

is a numeric variable;

\*\* Part b);

PROC MEANS DATA = sasdata.gas MIN MAX MAXDEC = 2;

CLASS Year;

VAR GasPrice;

TITLE 'Minimum and Maximum Gas Price per Year';

RUN;

\*\* Part c);

PROC FORMAT;

VALUE Quar 1 - 3 = 'Q1'

4 - 6 = 'Q2'

7 - 9 = 'Q3'

10 - 12 = 'Q4';

RUN;

PROC MEANS DATA = sasdata.gas MEAN STDDEV MAXDEC = 2;

\*\* This could also be accomplished with a CLASS

statement;

BY Year Month;

VAR GasPrice;

FORMAT Month Quar.;

TITLE1 'Average and Standard Deviation of Gas Price';

TITLE2 'per Quarter per Year';

\*\* Part d);

OUTPUT OUT = gasave MEAN(GasPrice) = AvePrice

STDDEV(GasPrice) = SdPrice;

RUN;

PROC PRINT DATA = gasave NOOBS;

VAR Year Month AvePrice SdPrice;

FORMAT AvePrice SdPrice DOLLAR5.2;

TITLE1 'Summary Statistics of Gasoline Price';

TITLE2 'per Quarter per Year ';

RUN;

(sections 4.2, 4.7, 4.8, 4.10, 4.11, 4.24)

40. *Hint*: Use optional statements to keep the report of potential errors looking clean.

*Answer*:

LIBNAME sasdata 'c:\MySASLib';

\*\* Part a);

\*\* You can determine the attributes of variables and

data sets interactively by viewing the data set (or

table) properties, or you can use PROC CONTENTS;

PROC CONTENTS DATA = sasdata.sff;

RUN;

\*\* The Continent variable has a length of 13 and the

Country variable has a length of 30;

\*\* Part b);

PROC FREQ DATA = sasdata.sff;

TABLES Continent / MISSING;

TITLE 'Number of Countries within Each Continent';

RUN;

\*\* Part c);

PROC FORMAT;

VALUE case . = 'None'

OTHER = 'Had Case';

RUN;

PROC FREQ DATA = sasdata.sff;

TABLES Continent \* Apr Continent \* Aug / MISSING;

FORMAT Apr Aug case.;

TITLE 'Case versus No Case in April and August';

RUN;

\*\* Part e);

PROC SORT DATA = sasdata.sff OUT = sffsort;

BY Continent;

RUN;

\*\* Part d);

PROC PRINT DATA = sffsort;

WHERE FirstCase = . AND FirstDeath ~= .;

\*\* Part e);

ID Continent;

BY Continent;

VAR Country FirstCase Latest FirstDeath;

FORMAT FirstDeath MMDDYY10.;

TITLE1 'Listing of countries with';

TITLE2 'reported deaths but no reported cases';

RUN;

(sections 4.2, 4.3, 4.5, 4.6, 4.7, 4.8, 4.12, 4.24)

41. *Hint*: Use the structure of this data set, with one categorical variable to describe Topping and one continuous variable to describe Rating, to help you choose which SAS statements to use for the reports.

*Answer*:

LIBNAME sasdata 'c:\MySASLib';

\*\* Part a);

\*\* You can determine the attributes of variables and

data sets interactively by viewing the data set (or

table) properties, or you can use PROC CONTENTS;

PROC CONTENTS DATA = sasdata.pizzaratings;

RUN;

\*\* The Rating variable has a label 'Customer rating

(1 = would never order to 5 = would order often)' and

is numeric;

\*\* Part b);

PROC FORMAT;

VALUE scale . = 'n/a'

1 = 'never'

2 = 'might'

3 = 'at least once'

4 = 'occasionally'

5 = 'often';

RUN;

PROC PRINT DATA = sasdata.pizzaratings;

FORMAT Rating scale.;

TITLE 'Pizza Topping Rating Data';

RUN;

\*\* Part c);

PROC MEANS DATA = sasdata.pizzaratings N MEAN

MAXDEC = 3;

CLASS Topping;

VAR Rating;

TITLE 'Average Rating per Topping';

RUN;

\*\* Part d);

PROC TABULATE DATA = sasdata.pizzaratings;

CLASS Topping;

VAR Rating;

\*\* Supressing the label for the Rating variable gives

a cleaner table;

TABLE Topping, N \* FORMAT = 3. \* Rating = ''

MEAN \* FORMAT = 5.3 \* Rating = '';

TITLE 'Average Rating per Topping';

RUN;

(sections 4.5, 4.6, 4.8, 4.10, 4.13, 4.14, 4.16, 4.17)

42. *Hint*: The variable labels will tell you which code corresponds to Yes and No for the VegMeal variable. Experiment with the various procedures in this chapter to find which ones will work best for parts d) and e).

*Answer*:

LIBNAME sasdata 'c:\MySASLib';

\*\* Part a);

\*\* You can determine the attributes of variables and

data sets interactively by viewing the data set (or

table) properties, or you can use PROC CONTENTS;

PROC CONTENTS DATA = sasdata.conference;

RUN;

PROC FREQ DATA = sasdata.conference;

TABLES RegType;

TITLE 'Number of Attendees per Registration Type';

RUN;

\*\* Part b);

PROC FORMAT;

VALUE yn 0 = 'No'

1 = 'Yes';

RUN;

PROC PRINT DATA = sasdata.conference;

FORMAT VegMeal yn.;

TITLE 'Attendees of User Group Conference';

RUN;

\*\* Part c);

PROC TABULATE DATA = sasdata.conference

FORMAT = DOLLAR11.2;

CLASS AreaCode RegType;

VAR Rate;

TABLE AreaCode, SUM \* Rate \* (RegType ALL) /

MISSTEXT = '$0';

TITLE1 'Total Amount Paid';

TITLE2 'per Area Code and Registration Type';

RUN;

\*\* Part d);

PROC FREQ DATA = sasdata.conference;

TABLES AreaCode \* RegType / NOROW NOCOL;

TITLE1 'Number of Attendees';

TITLE2 'per Area Code and Registration Type';

RUN;

\*\* Part e);

PROC TABULATE DATA = sasdata.conference;

CLASS AreaCode RegType;

TABLE AreaCode ALL, N = '' \*

(RegType = 'N' \* FORMAT = 4.

ALL \* FORMAT = 4.)

PCTN = '' \* (RegType = '%' \* FORMAT = 5.2

ALL \* FORMAT = 5.2) /

MISSTEXT = '0';

TITLE1 'Number of Attendees';

TITLE2 'per Area Code and Registration Type';

RUN;

(sections 4.6, 4.8, 4.12, 4.13, 4.14, 4.15, 4.16, 4.17)

43. *Hint*: Consider using a subsetting statement within the procedures to select the observations of interest.

*Answer*:

LIBNAME sasdata 'c:\MySASLib';

\*\* Part a);

\*\* You can determine the attributes of variables and

data sets interactively by viewing the data set (or

table) properties, or you can use PROC CONTENTS;

PROC CONTENTS DATA = sasdata.elliptical;

RUN;

PROC MEANS DATA = sasdata.elliptical MEAN MAXDEC = 2;

WHERE State ~= 'California';

CLASS State;

VAR Shipping;

TITLE1 'Average Shipping Cost per State,';

TITLE2 'Excluding California';

RUN;

\*\* Part b);

PROC REPORT DATA = sasdata.elliptical NOWINDOWS;

COLUMN State Machine Cost Tax Shipping Total;

DEFINE State / ORDER;

DEFINE Machine / ORDER;

DEFINE Cost / NOPRINT;

DEFINE Tax / NOPRINT;

DEFINE Shipping / NOPRINT;

DEFINE Total / COMPUTED;

COMPUTE Total;

Total = Cost.SUM + Tax.SUM + Shipping.SUM;

ENDCOMP;

TITLE 'Total Cost by State and Machine';

RUN;

\*\* Part c);

PROC FORMAT;

VALUE $division 'California',

'Oregon',

'Washington' = 'West Coast'

OTHER = 'Others';

RUN;

PROC REPORT DATA = sasdata.elliptical NOWINDOWS;

WHERE Machine CONTAINS 'HRT';

COLUMN State Machine Cost Tax Shipping Total;

DEFINE State / GROUP;

DEFINE Machine / ACROSS;

DEFINE Cost / NOPRINT;

DEFINE Tax / NOPRINT;

DEFINE Shipping / NOPRINT;

DEFINE Total / COMPUTED;

COMPUTE Total;

Total = Cost.SUM + Tax.SUM + Shipping.SUM;

ENDCOMP;

\*\* Variables can also be formatted within DEFINE

statements;

FORMAT State $division.;

TITLE 'Total Cost by Region for HRT Machines';

RUN;

(sections 4.2, 4.8, 4.10, 4.18, 4.19, 4.20, 4.23, 4.24)

44. *Hint*: Use a procedure to calculate new variables without the use of a DATA step.

*Answer*:

LIBNAME sasdata 'c:\MySASLib';

\*\* Part a);

PROC CONTENTS DATA = sasdata.diving;

RUN;

PROC MEANS DATA = sasdata.diving;

VAR Score1 Score2;

TITLE 'Descriptive Statistics of Scoring Methods';

RUN;

\*\* Part b);

PROC MEANS DATA = sasdata.diving NOPRINT;

CLASS Name;

VAR Score2;

OUTPUT OUT = totals SUM(Score2) = TotalScore;

RUN;

PROC SORT DATA = totals;

BY DESCENDING TotalScore;

RUN;

PROC PRINT DATA = totals;

WHERE \_TYPE\_ = 1;

TITLE 'Total Overall Score Using New Scoring Method';

RUN;

\*\* Gold = ZAKHAROV Ilya, Silver = QIN Kai,

Bronze = HE Chong;

\*\* Part c);

PROC FORMAT;

VALUE scale 0 -< 0.5 = 'Completely Failed'

0.5 -< 2.5 = 'Unsatisfactory'

2.5 -< 5 = 'Deficient'

5 -< 7 = 'Satisfactory'

7 -< 8.5 = 'Good'

8.5 -< 9.5 = 'Very Good'

9.5 - high = 'Excellent';

RUN;

PROC FREQ DATA = sasdata.diving;

TABLE Name \* J1 Name \* J2 Name \* J3 Name \* J4

Name \* J5 Name \* J6 Name \* J7

/ NOPERCENT NOROW NOCOL;

FORMAT J1 - J7 scale.;

TITLE 'Counts of Score Groupings';

RUN;

\*\* Part d);

PROC REPORT DATA = sasdata.diving NOWINDOWS;

COLUMN Name Dive J1 J2 J3 J4 J5 J6 J7

LowDive HighDive;

DEFINE Name / GROUP;

DEFINE Dive / ORDER;

DEFINE J1 / ANALYSIS NOPRINT;

DEFINE J2 / ANALYSIS NOPRINT;

DEFINE J3 / ANALYSIS NOPRINT;

DEFINE J4 / ANALYSIS NOPRINT;

DEFINE J5 / ANALYSIS NOPRINT;

DEFINE J6 / ANALYSIS NOPRINT;

DEFINE J7 / ANALYSIS NOPRINT;

DEFINE LowDive / COMPUTED 'Low Dive Score';

DEFINE HighDive / COMPUTED 'High Dive Score';

COMPUTE LowDive;

LowDive = MIN(J1.SUM,J2.SUM,J3.SUM,J4.SUM,J5.SUM,

J6.SUM,J7.SUM);

ENDCOMP;

COMPUTE HighDive;

HighDive = MAX(J1.SUM,J2.SUM,J3.SUM,J4.SUM,J5.SUM,

J6.SUM,J7.SUM);

ENDCOMP;

TITLE 'Minimum and Maximum Score per Dive';

RUN;

(sections 4.2, 4.3, 4.5, 4.8, 4.10, 4.11, 4.12, 4.18, 4.19, 4.20, 4.23, 4.24)