Programming Exercises

19. *Hint:* Be sure to make the necessary changes to the TABLES, WHERE, and TITLE statements.

*Answer:*

LIBNAME sasdata 'c:\MySASLib';

\*\* Parts a) and b);

OPTIONS MPRINT;

%LET classvar = Derivation;

%LET selectvalue = Mutation;

PROC FREQ DATA = sasdata.cats;

TABLES &classvar;

TITLE "Cat Breeds by &classvar";

RUN;

PROC PRINT DATA = sasdata.cats;

WHERE &classvar = "&selectvalue";

TITLE "Cat Breeds with &classvar = &selectvalue";

RUN;

\*\* Part c);

%MACRO catreport(classvar=,selectvalue=);

\*\* Part d);

ODS PDF FILE = "c:\MyPDFFiles\CatRpt&classvar..pdf";

PROC FREQ DATA = sasdata.cats;

TABLES &classvar;

TITLE "Cat Breeds by &classvar";

RUN;

PROC PRINT DATA = sasdata.cats;

WHERE &classvar = "&selectvalue";

TITLE "Cat Breeds with &classvar = &selectvalue";

RUN;

ODS PDF CLOSE;

%MEND catreport;

%catreport(classvar = Hair, selectvalue = Long)

(sections 7.2, 7.4, 7.5)

20. *Hint:* For part b) consider using macro statements that will control which standard SAS statements will be resolved by the macro processor. Remember you are writing a program that writes a program.

*Answer:*

LIBNAME sasdata 'c:\MySASLib';

\*\* Parts a), b), and c);

%MACRO StType(type=,var=,label=);

%IF &type = QL %THEN %DO;

PROC FREQ DATA = sasdata.virginiakey;

TABLES &var;

\*\* part d)

TITLE "Frequencies for &label";

RUN;

%END;

%ELSE %IF &type = QN %THEN %DO;

PROC MEANS DATA = sasdata.virginiakey;

VAR &var;

\*\* part d)

TITLE "Descriptive Statistics for &label";

RUN;

%END;

%MEND StType;

\*\* Part e);

%StType(type = QL, var = Month,

label = Month)

%StType(type = QL, var = Warning,

label = Marine Warning)

%StType(type = QN, var = AirTemp,

label = Air Temp)

%StType(type = QN, var = WindSpeed,

label = Wind Speed)

(sections 7.4, 7.5, 7.6)

21. *Hint:* Consider how best to name the variables and data sets created in part c) so that they will combine correctly in part d).

*Answer:*

LIBNAME sasdata 'c:\MySASLib';

\*\* Parts a) and f);

%LET selectedyear = 1999;

PROC MEANS DATA = sasdata.airtraffic NOPRINT;

WHERE Year = &selectedyear;

BY Airline;

VAR BOSFlights BOSPassengers;

OUTPUT OUT = bos SUM(BOSFlights BOSPassengers) =

Flights Passengers;

RUN;

\*\* Part b);

PROC SORT DATA = bos;

BY DESCENDING Passengers;

RUN;

DATA mostpassengersbos;

SET bos;

IF \_N\_ = 1;

PassengersPerFlight = ROUND(Passengers / Flights);

RUN;

\*\* Part c);

%MACRO flights(airport=);

PROC MEANS DATA = sasdata.airtraffic NOPRINT;

WHERE Year = &selectedyear;

BY Airline;

VAR &airport.Flights &airport.Passengers;

OUTPUT OUT = &airport SUM (&airport.Flights

&airport.Passengers) =

Flights Passengers;

RUN;

PROC SORT DATA = &airport;

BY DESCENDING Passengers;

RUN;

DATA most&airport;

SET &airport;

IF \_N\_ = 1;

PassengersPerFlight = ROUND(Passengers / Flights);

AirportCode = "&airport";

RUN;

%MEND flights;

%flights(airport = BOS)

%flights(airport = EWR)

%flights(airport = MIA)

%flights(airport = SEA)

%flights(airport = SFO)

%flights(airport = LAX)

%flights(airport = SAN)

%flights(airport = DEN)

%flights(airport = ORD)

%flights(airport = ATL)

%flights(airport = DFW)

%flights(airport = HNL)

\*\* Part d);

DATA all;

SET mostbos mostewr mostmia mostsea mostsfo mostlax

mostsan mostden mostord mostatl mostdfw mosthnl;

RUN;

\*\* Part e);

PROC PRINT DATA = all NOOBS;

\*\* The ID statement provides nicer output but

AirportCode could be included in the VAR

statement instead;

ID AirportCode;

VAR Airline Flights Passengers PassengersPerFlight;

FORMAT Flights Passengers COMMA12.;

TITLE1 'Airlines With the Most Passengers';

TITLE2 "for &selectedyear";

RUN;

(sections 7.2, 7.3, 7.4, 7.5)

22. *Hint:* Consider using a procedure to calculate the average, and creating data-driven macro variables for the footnote.

*Answer:*

LIBNAME sasdata 'c:\MySASLib';

\*\* Part a);

DATA students;

SET sasdata.studytime;

TimePerUnit = Time / Units;

RUN;

\*\* Part b);

PROC MEANS DATA = students NOPRINT;

CLASS Section;

VAR TimePerUnit;

OUTPUT OUT = MeanTime Mean(TimePerUnit) = MeanHrs;

RUN;

\*\* Part d)

OPTIONS SYMBOLGEN;

\*\* Part c);

%MACRO StudyRpt(sect=);

\*\* Parts e) and f);

DATA \_NULL\_;

SET MeanTime;

WHERE Section = "&sect";

IF MeanHrs < 2 THEN DO;

\*\* Some special handling has to be used on the

average in order for it to align properly in

the footnote;

CALL SYMPUT("avetime",

LEFT(PUT(ROUND(MeanHrs,0.01),4.2)));

CALL SYMPUT("foot",

"Minimum Average Study Time Not Met");

END;

ELSE IF MeanHrs >= 2 THEN DO;

CALL SYMPUT("avetime",

LEFT(PUT(ROUND(MeanHrs,0.01),4.2)));

CALL SYMPUT("foot",

"Minimum Average Study Time Met");

END;

RUN;

\*\* Part c);

PROC PRINT DATA = Students NOOBS;

WHERE Section = "&sect";

TITLE1 "Section &sect of Orientation to the";

TITLE2 "Statistics Major";

FOOTNOTE "&foot, Average = &avetime";

RUN;

%MEND StudyRpt;

\*\* Part g);

%StudyRpt(sect = 01)

%StudyRpt(sect = 02)

(sections 7.5, 7.7)

23. *Hint*: Using conditional logic within the macro will enable you to write one macro that will work for all property types for part c). For part d), you will need to save the means for each of the property types in separate data sets, and then combine them together outside the macro.

*Answer*:

LIBNAME sasdata 'c:\MySASLib';

\*\* part c);

OPTIONS MPRINT;

\*\* part e);

PROC FORMAT;

VALUE Branch . = 'ALL Branches'

1 = 'LIV925'

2 = 'SV408'

3 = 'SLO805'

4 = 'GLN626'

5 = 'COR760';

VALUE Ptype 1 = 'Primary Residence'

2 = 'Secondary Residence'

3 = 'Investment or Rental'

4 = 'Commercial';

RUN;

%MACRO loans(PT=,Limit=);

\*\* part c);

%IF &PT = 1 %THEN %DO;

PROC MEANS DATA = sasdata.loanapp;

WHERE LoanApproved = 1 AND PropType = &PT AND

Price > &limit;

CLASS Branch;

VAR CreditScore Interest PercentDown;

\*\* part d);

\*\* The WHERE statement can be used with an OUTPUT

statement in order to limit to certain

observations. An alternative would be to use

the WHERE statement in the SET statement for

part d) when reading the data set PROP1;

OUTPUT OUT = Prop&PT (WHERE = (\_TYPE\_ = 1))

MEAN(CreditScore Interest PercentDown) =

AveCreditScore AveInterest

AvePercentDown ;

%END;

%ELSE %DO;

\*\* parts a) and b);

PROC MEANS DATA = sasdata.loanapp;

WHERE LoanApproved = 1 AND PropType = &PT AND

Price > &limit;

VAR CreditScore Interest PercentDown;

\*\* part d);

OUTPUT OUT = Prop&PT

MEAN(CreditScore Interest PercentDown) =

AveCreditScore AveInterest

AvePercentDown;

RUN;

%END;

\*\* part d);

DATA Prop&PT;

SET Prop&PT;

PropType = &PT;

Limit = &limit;

RUN;

%MEND loans;

%loans(PT = 1, limit = 800000)

%loans(PT = 2, limit = 800000)

%loans(PT = 3, limit = 1000000)

%loans(PT = 4, limit = 1200000)

\*\* part d);

DATA allprop;

SET prop1 prop2 prop3 prop4;

RUN;

\*\* part e);

PROC PRINT DATA = allprop LABEL NOOBS;

\*\* The ID statement provides nicer output but it is

not required;

ID Branch;

VAR PropType Limit \_FREQ\_ AveCreditScore AveInterest

AvePercentDown;

FORMAT PropType ptype. Branch branch. Limit DOLLAR11.

AveCreditScore 3.

AveInterest AvePercentDown 5.3;

LABEL \_FREQ\_ = 'Number of Loans'

PropType = 'Property Type'

Limit = 'Properties Over'

Branch = 'Branch';

TITLE1 'Mean Credit Score, Interest Rate,';

TITLE2 'and Down Payment for High Priced Properties';

RUN;

(sections 7.3, 7.4, 7.5, 7.6, 7.8)

24. *Hint*: Be careful to type the program into the editor *exactly as written*.

*Answer*:

\*\* Part c);

OPTIONS MPRINT MLOGIC;

\*\* Parts a) and e);

%MACRO tt(NumRows=,NumCols=);

\*\* Replace macro variable references with i and j;

DATA table (DROP = i j);

ARRAY col(&NumCols) col1 - col&NumCols;

\*\* Remove quotation marks from macro variable

references;

DO i = 1 TO &NumRows;

DO j = 1 TO &NumCols;

col(j) = i \* j;

END;

OUTPUT;

END;

RUN;

PROC PRINT DATA = TABLE;

TITLE1 'Times Table for';

TITLE2 '&NumRows by &NumCols';

TITLE3 "Printed on &sysdate";

RUN;

%MEND tt;

%tt(NumRows = 12,NumCols = 12)

\*\* Part b);

\*\* There are are several errors that indicate SAS has a

problem with the DROP statement. There is also a note

that indicates that SAS has automatically converted

character values to numeric;

\*\* Part d);

\*\* At the top of the DATA step the MPRINT(TT) message

for the DROP option shows that the macro variables

were resolved to the value 12. These are not

variables in the data set which is causing SAS to

have a problem with the DROP option. To fix this

the DROP option could be removed from the program

or variables that do not need to stay in the data

set, like i and j, could replace the macro variable

references. The MPRINT(TT) messages for the DO i and

DO j loops also indicate that SAS resolved the

&NumRows and &NumCols macro variables as character

text strings. Removing the double quotation marks

from &NumRows and &NumCols will tell SAS to resolve

these macro variables as numeric;

\*\* Part e);

\*\* The first two titles in the output use &NumRows and

&NumCols instead of 12 and 12. This occurred because

these titles uses single quotation marks instead of

double quotation marks, so SAS will not resolve the

macro variables;

(sections 7.2, 7.4, 7.5, 7.8)