TECHNICAL REPORT STANDARD PAGE

1. Report No. FHWA/LA-92/248	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Wet Weather Highway Accident Analysis and Skid Resistance Data Management System	5. Report Date June 1992	
	6. Performing Organization Code	
7. Author(s) R. C. McIlhenny, K. S. Lee, Y. S, Chen	8. Performing Organization Report No. 248 (Volume III)	
9. Performing Organization Name and Address Industrial Engineering Department Louisiana State University Baton Rouge, LA 70803-6409	10. Work Unit No.	
	11. Contract or Grant No. 90-4SS	
12. Sponsoring Agency Name and Address Louisiana Transportation Research Center 4101 Gourrier Avenue Baton Rouge, LA 70808	13. Type of Report and Period Covered Final Report (Volume III) 2-1-90 Thru 6-30-92	
	14. Sponsoring Agency Code HPR No. 0010(15)	

15. Supplementary Notes
Conducted in cooperation with the U.S. Department of Transportation Federal Highway Administration.

16. Abstract

The objectives and scope of this research are to establish an effective methodology for wet weather accident analysis and to develop a database management system to facilitate information processing and storage for the accident analysis process, skid resistance testing, and other related tasks. The methodology employed consists of four phases: review and documentation of current LDOTD and LTRC procedures, engineering and statistical review of literature and procedures in the area of accident analysis, identification and recommendation of improvements which may facilitate data management and recovery, and design and development of a new computer information system based on recommendations defined in the third task. An effective wet weather accident analysis, testing, and database management system that allows only needed locations to be identified, tested, and reported is implemented.

Volume II of this report consists of the data base management systems Users manual.

Volume III of this report consists of data base management systems Reference manual.

17. Key Words wet weather accidents, skid resistance testing, data management, accident data analysis		18. Distribution Statement No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 238	22. Price

Form DOT F1700.7 (1-92)

WET WEATHER HIGHWAY ACCIDENT ANALYSIS AND SKID RESISTANCE DATA MANAGEMENT SYSTEM
(Volume III: Reference Manual)

by

R. C. McIlhenny, Ph.D.
Associate Professor of Industrial Engineering
Louisiana State University
Baton Rouge, LA 70803

K. S. Lee, Ph.D., P.E.
Associate Professor of Industrial Engineering
Louisiana State University
Baton Rouge, LA 70803

Y. S. Chen, Ph.D.
Associate Professor of Quantitative Business Analysis
Louisiana State University
Baton Rouge, LA 70803

conducted for

LOUISIANA DEPARTMENT OF TRANSPORTATION AND DEVELOPMENT
LOUISIANA TRANSPORTATION RESEARCH CENTER

in cooperation with
U.S. Department of Transportation
FEDERAL HIGHWAY ADMINISTRATION

MAY 1992

WET WEATHER HIGHWAY ACCIDENT ANALYSIS AND SKID RESISTANCE DATABASE MANAGEMENT SYSTEM

REFERENCE MANUAL

Table of Contents

1. INTRODUCTION

The purpose of this document is to give the DBA (Database Administrator) the flexibility of further customizing the Wet Weather Accident Analysis and Skid Resistance Database Management System to suit the end-user requirements any time in the future. This system has been implemented on an IBM 3090 machine at Louisiana State University System Network Computer Center. The environment is TSO/SPF. To get the system working, it is necessary to have the following packages installed on the computer system.

1) SAS/BASE Version 5.18 or above
2) SAS/SQL Version 5.18 or above
3) SAS/AF Version 6.06 or above

It is recommended that before getting started on the system, the DBA should have a manual of each of the packages mentioned above. For any further questions or clarifications regarding the database system, the following persons could be contacted.

Dr. Kwan S. Lee
Department of Industrial Engineering
Louisiana State University
Baton Rouge, LA 70803
Phone # (504) 388 5369

Dr. Ye-Sho Chen
Department of Quantitative Business Analysis
Louisiana State University
Baton Rouge, LA 70803
Phone # (504) 388 2510

For any questions on SAS, SAS Institute could be contacted at the following address.

SAS Institute Inc.
SAS Circle ⁞ Box 8000
Cary, NC 27512-8000

2. THE WET WEATHER ACCIDENT ANALYSIS AND SKID RESISTANCE MANAGEMENT SYSTEM

The overall picture of the entire system has been given in figure 1. The information flow goes on as follows:

(1) The input to the system is the yearly accident data stored in the DOTACC tapes. These tapes contain the accident data in an ASCII format and can be obtained from the Department of Transportation and Development, Louisiana.

(2) A set of SAS programs transform the data in these tapes into a relational database. This database contains 5 Relational tables, namely - Accident, Section, Skid, Driver and Vehicle.

(3) The data from these tables is an input to the Wet Weather Analysis programs.

(4) PGM1 AND PGM2 take the hourly data and the hourly precipitation data as an input and calculate the mean proportion of wet time for all the existing highways in the state of Louisiana. PGM3, PGM4 and PGM5 take the above and the accident data from the relational tables as an input and identify the Wet Hazardous Locations on Louisiana highways.

(5) The outputs of the analysis programs are the big outputs, one for the Hazardous sections and intersections, one for Hazardous clusters and one for Hazardous spots. These outputs are accessed by some SAS/SQL views. These views are created by the end-user as he proceeds along the menus. The menus, at each stage/level, dynamically add conditions to create SQL

statement. Further, it's possible to run the analysis programs in the background.

(6) The Maintenance and Archives functions of the database management system creates a huge view to browse and edit the relational tables. These views are different from SAS/SQL views, since it's possible to modify these views from SAS/AF.

LEGEND

1: DOTDACC TAPES

2: SAS PROGRAMS TO TRANSFORM THE DATA INTO RELATIONAL TABLES
```
PROJECT => IEKLEE
GROUP   => LTRC
TYPE    => FINAL
MEMBER  => ACC16 <SECTION/SKID/DRIVER/VEHICLE>
```

3: RELATIONAL TABLES
Can be seen by getting into the DISPLAY MANAGER SYSTEM (DMS) of SAS606. Program TEST (present in SASUSER.PROFILE) can be copied into the DMS and submitted. To see the contents of the relational tables, a PROC CONTENTS of Library NEWLIB and NEWLIB1 can be performed. The tables have been named as:
ACC188
DRVR88
VHCL88
SECTN88
SKID
The physical existence of these tables is in:
IEKLEE.NEW2.SASDATA and IEKLEE.NEW6.SASDATA

4: INFILE statements in programs PGM3, PGM4, and PGM5.

5: Analysis programs PGM3, PGM4, and PGM5 stored in the following dataset:
```
PROJECT => IEKLEE
GROUP   => LTRC
TYPE    => FINAL
MEMBER  => PGM3 <PGM4/PGM5>
```

6: OUT statements in programs PGM3, PGM4, and PGM5.

7: Outputs of PGM3, PGM4 and PGM5:
   SNSI88 stored in NEWLIB for sections/intersections of 1988.
   SNCLS88 stored in NEWLIB for clusters of 1988.
   SNSPT88 stored in NEWLIB for spots of 1988.

8: FSVIEW statements in the menu programs (see the very last program of any chain in Figure 2).

9: Embedded SAS/SQL statements in the menu programs.

10: SCREEN CONTROL LANGUAGE (SCL) statements in the menu programs to trigger SAS/SQL statements.

11: MENU PROGRAMS stored in NEWLIB as a catalog called LTRC.

12: TSO Submit statements to submit programs PGM3, PGM4, and PGM5.

13: FSVIEW statements of SCL in Browse mode for viewing tables.

3. ANALYSIS PROGRAMS

3.1 INSTRUCTIONS FOR RUNNING THE WET WEATHER HIGHWAY ACCIDENT ANALYSIS PROGRAMS

The Wet Weather Highway Accident Analysis algorithms have been implemented in SAS-Statistics using the TSO environment. To run the programs, the user will have to get into the TSO operating system and logon to the project account. The following steps explain the procedure.

1) The user has to key in 't' next to the SELECT prompt in the main menu.
2) The system requests the user to key in the LOGON ID. Enter IEKLEE.
3) Now, the system requests the user to enter the password. The user has to enter the password to gain access to the package.
4) Three asterisk symbols, ***, appear on the screen. The user has to repeatedly hit the 'ENTER' key till the READY prompt appears on the screen.
5) At the READY prompt, type 'spf' i.e Screen Productivity Facility to get into the ISPF/PDF primary option menu.
6) A list of options appears on the menu. Key in the required option next to the OPTION prompt on the upper-left corner on the screen. To run the Wet Weather Highway Accident Analysis programs, key in option 2 i.e EDIT.
7) The user reaches an EDIT-ENTRY PANEL on selecting the EDIT option.

Four prompts, PROJECT, GROUP, TYPE and MEMBER appear on the screen, under the heading ISPF LIBRARY. To run the analysis programs (pgm1, pgm2, pgm3, pgm4, pgm5), enter, next to the prompts, the following commands.

PROMPT		COMMAND
PROJECT	====>	IEKLEE
GROUP	====>	LTRC
TYPE	====>	FINAL
MEMBER	====>	PGM1 (to run program 1)

The user is taken to the SAS source code. At this juncture the user can operate on pgm1 and run it, if needed.

PROGRAM 1

What does this program do?

This program takes as input, the hourly rainfall data from the precipitation files. It calculates the total wet hours of rainfall based on the WETTIME model for a station with universal rain gage. It does not distinguish between frozen and non-frozen precipitation.

How to run the program for 1989 & 1990 ?

1) Collect precipitation data for 1989 & 1990 and name it P8089._____ (NOR for New Orleans)
2) Make sure that the format is the same as in previous years. Compare with file P6660.HUF
3) Change the JCL (Job Control List) statement having FILE REF UNI to DD SDN = IEKLEE.P8089.HUF

4) Get the hourly surface observation data for 1989 and name it HSNO.HLY

5) Make sure that the data is in the same format as in HOURLY.DATA (New Orleans)

6) Change the JCL statement having FILE REF.HLY to DD SDN = IEKLEE.HOURLY.DATA (New Orleans)

7) Run the program for one year at a time. To do this, go to line 002430 of pgm1 and key in between the inverted commas, the year for which the analysis is required.

8) Next, go to line 004910 and key in the same year, next to IF YR= without disturbing the ';'. Enter only the last two digits of the year to be analyzed.

FOR RUNNING PROGRAM FOR YEARS PRIOR TO 1989:

1) Read INFILE UNI P6660.HUF for New Orleans.

2) Read INFILE UNI P0549.HUF for Baton Rouge.

3) Read INFILE UNI P5078.HUF for Lake Charles.

4) Read INFILE UNI P8440.HUF for Shreveport.

After making the required changes, key in 'sub' next to the COMMAND prompt to run the program.

A message 'JOB LTRC (Job No .....) SUBMITTED' appears on the screen.

PROGRAM 2

What does this program do?

It calculates the distance between a first order and second order station and lists the nearest and second nearest first order station to each second order station distance in arc distance. It conducts an empirical Bayesian analysis of the proportion wet time based on WETTIME calculations, for every triangle formed in the mesh of weather stations.

How to run the program ?

1) To run program 2, the user has to once again get back to the EDIT-ENTRY PANEL by entering '=2' next to the COMMAND prompt.
2) As explained before, four prompts PROJECT, GROUP, TYPE and MEMBER appear on the screen under the heading ISPF LIBRARY. Retain the same commands for the first three prompts. Key in 'pgm2' next to the MEMBER prompt.
3) There is no need to change any statement as this program automatically reads the output from program 1.
4) Key in 'sub' next to the COMMAND prompt to run the program. A message 'JOB LTRC (Job No ....) SUBMITTED' appears on the screen.

PROGRAM 3

What does this program do?

This program flags clusters by the wet accident criterions developed for accident data in Louisiana by the Rate Quality Control method and the second Bayesian criterion on the basis of simulation runs conducted before.

PROGRAM 4

What does this program do?

This program flags intersections and sections for the Bayesian criterions developed for accident data in Louisiana.

PROGRAM 5

What does this program do?

This program flags spots by the wet accident criterion developed for accident data in Louisiana.

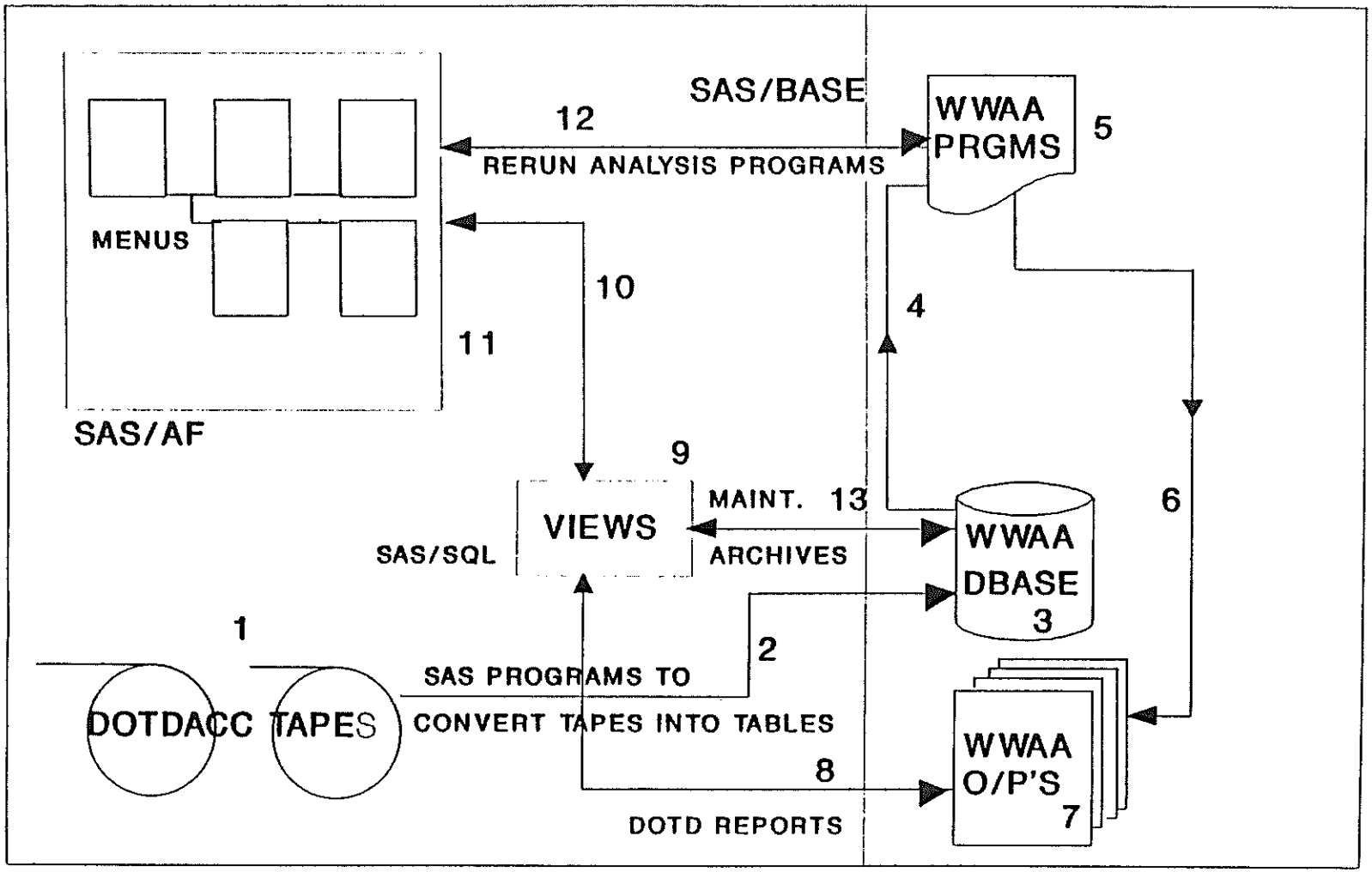
How to run programs 3, 4 and 5 for 1989 and 1990 ?

1) As explained before, the user has to get back to the EDIT-ENTRY PANEL and key in 'pgm3' or 'pgm4' or 'pgm5' depending on which program is to be run.

2) To run programs 3, 4 and 5 for 1989 and 1990 change the JCL ACCI statement to ACC.MASTER 89 AND ACC.MASTER 90, respectively.

3) After making the changes to the code, key in 'sub' next to the COMMAND prompt to run the program.

WET WEATHER ACCIDENT ANALYSIS AND SKID RESISTANCE DATABASE MANAGEMENT SYSTEM



INFORMATION FLOW

Figure 1.

3.2 ANALYSIS PROGRAM LISTING

```
//LTRC JOB (1318,05886,0030,9999),'SRIKANTH',NOTIFY=IEKLEE,                00010099
//          MSGCLASS=S                                                     00020099
/*JOBPARM SHIFT=H                                                         00040099
//STEP1 EXEC SAS,TIME=30                                                  00041099
//UNI DD DSN=IEKLEE.P8440.HUF,DISP=SHR                                    00050099
//HLY DD DSN=IEKLEE.SHRVPT.HLY,DISP=SHR                                   00060099
//TEMP DD DSN=IEKLEE.HOURLY.NAT,DISP=SHR                                  00070099
//OUT DD DSN=IEKLEE.WETHOURS.SHV,DISP=MOD                                 00071099
//SYSIN DD *                                                              00080099
                                                                          00090099
                                                                          00090199
* THIS PROGRAM TAKES AS INPUT THE HOURLY RAINFALL DATA FROM               00090299
  THE FOLLOWING PRECIPITATION FILES ;                                     00090399
                                                                          00090499
* INSTRUCTIONS TO RUN 1989 AND 1990 DATA FOR NEW ORLEANS;                 00090599
                                                                          00090699
* COLLECT 1989 AND 1990 PRECIPITATION DATA AND                            00090799
   NAME IT AS P8089.NOR FOR NEWORLEANS;                                   00090899
                                                                          00090999
* MAKE SURE THAT THE FORMAT IS THE SAME AS IN PREVIOUS YEARS              00091099
  COMPARE  WITH FILE P6660.HUF;                                           00091199
                                                                          00091299
* CHANGE JCL STATEMENT HAVING FILEREF UNI TO DD DSN=IEKLEE.P8089.HUF;     00091399
                                                                          00091499
* GET HOURLY SURFACE OBSERVATION DATA FOR 1989 AND 1990 AND NAME IT       00091599
  AS HSONO.HLY;                                                           00091699
                                                                          00091799
* MAKE SURE FORMAT IS SAME AS IN HOURLY.DATA(NEWORLNS);                   00091899
                                                                          00091999
* CHANGE JCL STATEMENT HAVING FILEREF HLY TO DD DSN=IEKLEE.HOURLY.DATA    00092099
  (NEWORLNS);                                                             00092199
                                                                          00092299
* CHANGE YEAR NUMBER TO CURRENT YEAR TO BE RUN IN LINES                   00092399
   WHERE I HAVE COMMENTED CHANGE YEARS;                                   00092499
                                                                          00092599
* RUN THE PROGRAM ONE YEAR AT A TIME (MEMORY PROBLEMS);                   00092699
                                                                          00092799
* FOR RERUNNING PREVIOUS YEARS PRIOR TO 1989 DO THE FOLLOWING;            00092899
*INFILE UNI P6660.HUF FOR NEW ORLEANS STATION;                            00092999
*INFILE UNI P0549.HUF FOR BATON ROUGE STATION;                            00093099
*INFILE UNI P5078.HUF FOR LAKE CHARLES STATION;                           00093199
*INFILE UNI P8440.HUF FOR SHREVEPORT STATION;                             00093299
                                                                          00094099
 * THIS ALSO TAKES HOURLY SURFACE OBSERVATIONS DATA FROM                  00094199
   THE FOLLOWING HOURLY FILES TO BE MAINTAINED IN DISKS;                  00094299
                                                                          00094399
*INFILE HLY HOURLY.DATA(NEWORLNS) FOR NEW ORLEANS;                        00095099
*INFILE HLY HOURLY.DATA(LAKECHAS) FOR LAKE CHARLES;                       00096099
*INFILE HLY BR8088.HLY OR BR4882.HLY FOR BATON ROUGE FOR THE RESPECTIVE   00097099
 YEARS;                                                                   00097199
*INFILE HLY SHRVPT.HLY FOR SHREVEPORT;                                    00098099
                                                                          00099099
*OUTPUT FILE WETHOURS.LAKECHAS FOR LAKE CHARLES;                          00099199
*OUTPUT FILE WETHOURS.NEWORLNS FOR NEW ORLEANS;                           00099299
*OUTPUT FILE WETHOURS.BTR FOR BATON ROUGE;                                00099399
*OUTPUT FILE WETHOURS.SHR FOR SHREVEPORT;                                 00099499
                                                                          00100099
*************************************************************************; 00110099
*  THIS PROGRAM CALCULATES THE TOTAL WET HOURS  OF RAINFALL BASED     *;  00120099
*  ON THE WETTIME MODEL FOR A STATION WITH UNIVERSAL RAIN GAUGE       *;  00130099
```

```
* IT DOES NOT DISTINGUISH BETWEEN FROZEN AND NONFROZEN PRECIPITATION *; 00131099
*************************************************************************; 00140099
                                                                          00150099
DATA ONE;                                                                 00160099
  INFILE UNI;                                                             00170099
   INPUT (DATE AR1 AR2-AR24) ($9. $5.  23*$5.);                           00180099
YEAR=SUBSTR(DATE,1,4);                                                    00190099
MONTH=SUBSTR(DATE,5,2);                                                   00200099
DAY=SUBSTR(DATE,7,2);                                                     00210099
DATE=MDY(MONTH,DAY,YEAR);                                                 00220099
DATE=COMPRESS(DATE);                                                      00230099
                                                                          00242099
*CHANGE YEAR HERE;                                                       00242199
IF YEAR='1959';                                                           00243099
                                                                          00250099
ARRAY AOUR{24} $  AR1-AR24;                                               00260099
ARRAY HUR{24} HR1-HR24;                                                   00270099
ARRAY MISSING{24} $  MISS1-MISS24;                                        00280099
* EXTRACT MISSING INFORMATION;                                            00290099
DO J=1 TO 24;                                                             00300099
A=AOUR{J};                                                                00310099
B='A'; C='D'; D='M';                                                      00320099
IF INDEX(A,B) ¬= 0 OR INDEX(A,C) ¬= 0 OR INDEX(A,D) ¬= 0 THEN DO;         00330099
MISSING{J}=1; AOUR{J}='0';                                                00340099
END;                                                                      00350099
ELSE MISSING{J}=0;                                                        00360099
STR= ' '; AOUR{J}= STR || AOUR{J}; AOUR{J}=TRIM(AOUR{J});                 00370099
HUR{J}=(AOUR{J}+0)/100;                                                   00380099
END;                                                                      00390099
                                                                          00400099
DROP AR1-AR24 B C D A STR;                                                00410099
                                                                          00420099
PROC SORT; BY DATE;                                                       00430099
                                                                          00440099
DATA I;                                                                   00450099
INFILE HLY;                                                               00460099
INPUT YR 6-7 MONTH $ 8-9 DAY $ 10-11 HUR 12-13   FROZEN1 27 FROZEN2 28   00470099
 FROZEN3 29 FOG 30 DEWPT 36-38 WIND 41-42 TEMP 47-49 RH 53-55 CC $ 56;    00480099
                                                                          00481099
* CHANGE YEAR HERE;                                                       00482099
* EXAMPLE (FOR MULTIPLE YEARS) IF YR>=85 AND YR<=89;                      00490099
IF YR=59;                                                                 00491099
HUR=HUR+1;                                                                00500099
YR1=1900+YR;                                                              00510099
A=' ';                                                                    00520099
YEAR= A || YR1;                                                           00530099
YEAR=TRIM(YEAR);                                                          00540099
DATE=MDY(MONTH,DAY,YEAR);                                                 00550099
DATE=COMPRESS(DATE);                                                      00560099
* CONVERSION OF KNOT WIND SPEEDS TO MILES PER HOUR;                       00570099
WIND = WIND*1.15;                                                         00580099
                                                                          00590099
* EDITING OF FROZEN PRECIPITATION DATA;                                   00600099
FRZN=SUM(FROZEN1,FROZEN2);                                                00610099
FRZN=FRZN+SUM(FROZEN2,FROZEN3);                                           00611099
                                                                          00620099
* FOR MISSING VALUES;                                                     00621099
IF RH=. THEN RH=60;                                                       00622099
IF WIND=. THEN WIND=2;                                                    00623099
IF TEMP=. THEN TEMP=70;                                                   00624099
IF DEWPT=. THEN DEWPT=50;                                                 00625099
                                                                          00630099
* TRANSLATION OF CLOUD COVER DATA;                                        00640099
```

```
IF CC ='A' OR CC='1' THEN CL=0.1;                                          00660099
IF CC ='B' OR CC='2' THEN CL=0.2;                                          00670099
IF CC ='C' OR CC='3' THEN CL=0.3;                                          00680099
IF CC ='D' OR CC='4' THEN CL=0.4;                                          00690099
IF CC ='E' OR CC='5' THEN CL=0.5;                                          00700099
IF CC ='F' OR CC='6' THEN CL=0.6;                                          00710099
IF CC ='G' OR CC='7' THEN CL=0.7;                                          00720099
IF CC ='H' OR CC='8' THEN CL=0.8;                                          00730099
IF CC ='I' OR CC='9' THEN CL=0.9;                                          00740099
IF CC ='{' OR CC='0' THEN CL=0.0;                                          00750099
IF CC ='X' THEN CL=1.0;                                                    00760099
IF CC ='-' OR CC=' ' THEN CL=0.5;                                          00761099
                                                                           00770099
*CONVERSION OF CLOUD COVER TO SOLAR RADIATION CORRECTED FOR CLOUDS RCL;    00780099
*SOLAR RADIATION AT LATITUDE 30 DEGREES NORTH;                             00790099
* FROM NET RADIATION RECEIVED FROM A HORIZONTAL SURFACE BY DE JONG;        00800099
                                                                           00810099
IF MONTH = 1 THEN R0=520/(24*60);                                          00820099
IF MONTH = 2 THEN R0=630/(24*60);                                          00830099
IF MONTH = 3 THEN R0=775/(24*60);                                          00840099
IF MONTH = 4 THEN R0=895/(24*60);                                          00850099
IF MONTH = 5 THEN R0=975/(24*60);                                          00860099
IF MONTH = 6 THEN R0=1000/(24*60);                                         00870099
IF MONTH = 7 THEN R0=990/(24*60);                                          00880099
IF MONTH = 8 THEN R0=925/(24*60);                                          00890099
IF MONTH = 9 THEN R0=820/(24*60);                                          00900099
IF MONTH =10 THEN R0=685/(24*60);                                          00910099
IF MONTH =11 THEN R0=560/(24*60);                                          00920099
IF MONTH =12 THEN R0=490/(24*60);                                          00930099
                                                                           00940099
*SOLAR RADIATION CORRECTED FOR CLOUD COVER;                                00950099
* FROM V P SINGH;                                                          00960099
RCL=R0*(1-0.65*CL);                                                        00970099
                                                                           00980099
*SETTING SOLAR RADIATION AT ZERO LEVELS FOR NIGHT TIME;                    00990099
* FROM SUNSET TABLES FOR LOUISIANA CITIES;                                 01000099
                                                                           01010099
IF MONTH =1 AND (HUR<9 OR HUR >15)THEN RCL=0;                              01020099
IF MONTH =2 AND (HUR<9 OR HUR >15)THEN RCL=0;                              01030099
IF MONTH =3 AND (HUR<8 OR HUR >16)THEN RCL=0;                              01040099
IF MONTH =4 AND (HUR<8 OR HUR >16)THEN RCL=0;                              01050099
IF MONTH =5 AND (HUR<7 OR HUR >17)THEN RCL=0;                              01060099
IF MONTH =6 AND (HUR<7 OR HUR >17)THEN RCL=0;                              01070099
IF MONTH =7 AND (HUR<7 OR HUR >17)THEN RCL=0;                              01080099
IF MONTH =8 AND (HUR<8 OR HUR >16)THEN RCL=0;                              01090099
IF MONTH =9 AND (HUR<8 OR HUR >16)THEN RCL=0;                              01100099
IF MONTH =10 AND (HUR<8 OR HUR >16)THEN RCL=0;                             01110099
IF MONTH =11 AND (HUR<8 OR HUR >15)THEN RCL=0;                             01120099
IF MONTH =12 AND (HUR<9 OR HUR >15)THEN RCL=0;                             01130099
                                                                           01140099
DIFF=ABS(TEMP-DEWPT);                                                      01150099
                                                                           01160099
DROP CC CL R0 FROZEN1-FROZEN3;                                             01170099
                                                                           01180099
PROC SORT; BY YEAR MONTH DAY HUR;                                          01190099
                                                                           01200099
DATA _NULL_;                                                               01210099
SET I; BY YEAR MONTH DAY HUR;                                              01220099
                                                                           01230099
FILE TEMP NOPRINT NOTITLE;                                                 01240099
IF FIRST.DAY THEN PUT DATE FRZN FOG DEWPT WIND                             01250099
                      TEMP RH RCL DIFF @@;                                 01260099
IF LAST.DAY THEN PUT  FRZN FOG DEWPT WIND                                  01270099
                      TEMP RH RCL DIFF;                                    01280099
```

```
ELSE PUT FRZN FOG DEWPT WIND                                                    01290099
                        TEMP RH RCL DIFF @@ ;                                   01300099
                                                                                01310099
                                                                                01320099
DATA III;                                                                       01330099
INFILE TEMP;                                                                    01340099
INPUT DATE $ FRZN1 FOG1 DEWPT1 WIND1 TEMP1 RH1 RCL1 DIFF1                       01350099
                     FRZN2 FOG2 DEWPT2 WIND2 TEMP2 RH2 RCL2 DIFF2               01360099
                     FRZN3 FOG3 DEWPT3 WIND3 TEMP3 RH3 RCL3 DIFF3               01370099
                     FRZN4 FOG4 DEWPT4 WIND4 TEMP4 RH4 RCL4 DIFF4               01380099
                     FRZN5 FOG5 DEWPT5 WIND5 TEMP5 RH5 RCL5 DIFF5               01390099
                     FRZN6 FOG6 DEWPT6 WIND6 TEMP6 RH6 RCL6 DIFF6               01400099
                     FRZN7 FOG7 DEWPT7 WIND7 TEMP7 RH7 RCL7 DIFF7               01410099
                     FRZN8 FOG8 DEWPT8 WIND8 TEMP8 RH8 RCL8 DIFF8               01420099
                     FRZN9 FOG9 DEWPT9 WIND9 TEMP9 RH9 RCL9 DIFF9               01430099
                FRZN10 FOG10 DEWPT10 WIND10 TEMP10 RH10 RCL10 DIFF10            01440099
                FRZN11 FOG11 DEWPT11 WIND11 TEMP11 RH11 RCL11 DIFF11            01450099
                FRZN12 FOG12 DEWPT12 WIND12 TEMP12 RH12 RCL12 DIFF12            01460099
                FRZN13 FOG13 DEWPT13 WIND13 TEMP13 RH13 RCL13 DIFF13            01470099
                FRZN14 FOG14 DEWPT14 WIND14 TEMP14 RH14 RCL14 DIFF14            01480099
                FRZN15 FOG15 DEWPT15 WIND15 TEMP15 RH15 RCL15 DIFF15            01490099
                FRZN16 FOG16 DEWPT16 WIND16 TEMP16 RH16 RCL16 DIFF16            01500099
                FRZN17 FOG17 DEWPT17 WIND17 TEMP17 RH17 RCL17 DIFF17            01510099
                FRZN18 FOG18 DEWPT18 WIND18 TEMP18 RH18 RCL18 DIFF18            01520099
                FRZN19 FOG19 DEWPT19 WIND19 TEMP19 RH19 RCL19 DIFF19            01530099
                FRZN20 FOG20 DEWPT20 WIND20 TEMP20 RH20 RCL20 DIFF20            01540099
                FRZN21 FOG21 DEWPT21 WIND21 TEMP21 RH21 RCL21 DIFF21            01550099
                FRZN22 FOG22 DEWPT22 WIND22 TEMP22 RH22 RCL22 DIFF22            01560099
                FRZN23 FOG23 DEWPT23 WIND23 TEMP23 RH23 RCL23 DIFF23            01570099
                FRZN24 FOG24 DEWPT24 WIND24 TEMP24 RH24 RCL24 DIFF24            01580099
                      ;                                                         01590099
                                                                                01600099
                                                                                01610099
PROC SORT; BY DATE;                                                             01620099
                                                                                01630099
DATA ONE1;                                                                      01640099
MERGE ONE III; BY DATE;                                                         01650099
                                                                                01660099
DATA TWO;                                                                       01670099
* GET FIVE HURS OF PRECI OF PREVIOUS DAY AND PUT IT IN THE SAME LINE;           01680099
SET ONE1; BY DATE;                                                              01690099
DATE=DATE+1;                                                                    01700099
ARRAY HR{10} HR20-HR24 HR25-HR29;                                               01710099
ARRAY FR{10} FRZN20-FRZN24 FRZN25-FRZN29;                                       01720099
ARRAY FG{10} FOG20-FOG24 FOG25-FOG29;                                           01730099
ARRAY DW{10} DEWPT20-DEWPT24 DEWPT25-DEWPT29;                                   01740099
ARRAY WD{10} WIND20-WIND24 WIND25-WIND29;                                       01750099
ARRAY TM{10} TEMP20-TEMP24 TEMP25-TEMP29;                                       01760099
ARRAY RH{10} RH20-RH24 RH25-RH29;                                               01770099
ARRAY RCL{10} RCL20-RCL24 RCL25-RCL29;                                          01780099
ARRAY DIFF{10} DIFF20-DIFF24 DIFF25-DIFF29;                                     01790099
                                                                                01800099
DO I= 6 TO 10;                                                                  01810099
HR{I}=HR{I-5};                                                                  01820099
FR{I}=FR{I-5};                                                                  01830099
FG{I}=FG{I-5};                                                                  01840099
DW{I}=DW{I-5};                                                                  01850099
WD{I}=WD{I-5};                                                                  01860099
TM{I}=TM{I-5};                                                                  01870099
RH{I}=RH{I-5};                                                                  01880099
RCL{I}=RCL{I-5};                                                                01890099
END;                                                                            01900099
                                                                                01910099
DATE=COMPRESS(DATE);                                                            01920099
```

```
KEEP DATE HR25-HR29 FRZN25-FRZN29 FOG25-FOG29 DEWPT25-DEWPT29
     WIND25-WIND29 TEMP25-TEMP29 RH25-RH29 RCL25-RCL29 DIFF25-DIFF29;

PROC SORT; BY DATE;

DATA THREE;
MERGE TWO ONE1;
BY DATE;


DATA TWOA;
* GET FIVE HURS OF NEXT DAY AND PUT IT IN THE SAME LINE;
SET THREE; BY DATE;
DATE=DATE-1;
DATE=COMPRESS(DATE);
ARRAY HR{10} HR1-HR5 HR30-HR34;
ARRAY FR{10} FRZN1-FRZN5 FRZN30-FRZN34;
ARRAY FG{10} FOG1-FOG5 FOG30-FOG34;
ARRAY DW{10} DEWPT1-DEWPT5 DEWPT30-DEWPT34;
ARRAY WD{10} WIND1-WIND5 WIND30-WIND34;
ARRAY TM{10} TEMP1-TEMP5 TEMP30-TEMP34;
ARRAY RH{10} RH1-RH5 RH30-RH34;
ARRAY RCL{10} RCL1-RCL5 RCL30-RCL34;
ARRAY DIFF{10} DIFF1-DIFF5 DIFF30-DIFF34;

DO I= 6 TO 10;
HR{I}=HR{I-5};
FR{I}=FR{I-5};
FG{I}=FG{I-5};
DW{I}=DW{I-5};
WD{I}=WD{I-5};
TM{I}=TM{I-5};
RH{I}=RH{I-5};
RCL{I}=RCL{I-5};
END;

KEEP DATE HR30-HR34 FRZN30-FRZN34 FOG30-FOG34 DEWPT30-DEWPT34
     WIND30-WIND34 TEMP30-TEMP34 RH30-RH34 RCL30-RCL34 DIFF30-DIFF34;


PROC SORT; BY DATE;

DATA THREEA;
MERGE TWOA THREE;
BY DATE;


DATA FOUR;
SET THREEA;

ARRAY HUR{34} HR25-HR29 HR1-HR24 HR30-HR34;
ARRAY WETHR{28} WHR1-WHR28;
ARRAY FRZN{34} FRZN25-FRZN29 FRZN1-FRZN24 FRZN30-FRZN34;
ARRAY FOG{34} FOG25-FOG29 FOG1-FOG24 FOG30-FOG34;
ARRAY DEWPT{34} DEWPT25-DEWPT29 DEWPT1-DEWPT24 DEWPT30-DEWPT34;
ARRAY WIND{34} WIND25-WIND29 WIND1-WIND24 WIND30-WIND34;
ARRAY TEMP{34} TEMP25-TEMP29 TEMP1-TEMP24 TEMP30-TEMP34;
ARRAY RH{34} RH25-RH29 RH1-RH24 RH30-RH34;
ARRAY RCL{34} RCL25-RCL29 RCL1-RCL24 RCL30-RCL34;
ARRAY DIFF{34} DIFF25-DIFF29 DIFF1-DIFF24 DIFF30-DIFF34;
```

```
DO I= 6 TO 29;                                                                  02570099
* WET HOUR DUE TO FOG;                                                          02580099
IF HUR{I}=0 THEN DO;                                                            02590099
  IF FOG{I}=1 OR FOG{I}=2 OR FOG{I}=3 THEN DO;                                  02600099
     IF DIFF{I} <=2 AND WIND{I}< 3 THEN DO;                                     02610099
     HUR{I}=0.06;                                                               02620099
     END;                                                                       02630099
  END;                                                                          02640099
END;                                                                            02660099
*WET HOUR DUE TO PRECIPITATION ;                                                02670099
IF HUR{I}¬=0 THEN DO;                                                           02680099
* DURATION OF RAINFALL DEPENDING ON INTENSITY;                                  02690099
   IF HUR{I}>=0.01 AND HUR{I}<0.02 THEN WETHR{I-5}=15;                          02700099
   IF HUR{I}>=0.02 AND HUR{I}<0.03 THEN WETHR{I-5}=30;                          02710099
   IF HUR{I}>=0.03 AND HUR{I}<0.05 THEN WETHR{I-5}=45;                          02720099
   IF HUR{I}>=0.05 THEN WETHR{I-5}=60;                                          02730099
* RUNOFF TIME ASSUMED TO BE 5 MINUTES;                                          02740099
  WETHR{I-5}=WETHR{I-5}+5;                                                      02750099
* DRYING TIME; *DEVIATIONS FROM MEAN DRYING TIME OF 31.6;                       02760099
   WETHR{I-5}=WETHR{I-5}+31.6;                                                  02770099
* TEMPERATURE RECORDS;                                                          02780099
IF TEMP{I}<67.5 THEN  WETHR{I-5}=WETHR{I-5}+3.7;                                02790099
IF TEMP{I}>=67.5 AND TEMP{I}<82.5 THEN WETHR{I-5}=WETHR{I-5}-0.7;               02800099
ELSE WETHR{I-5}=WETHR{I-5}-3.0;                                                 02810099
* RELATIVE HUMIDITY RECORDS;                                                    02820099
IF RH{I}<50 THEN  WETHR{I-5}=WETHR{I-5}-4.5;                                    02830099
IF RH{I}>=50 AND RH{I}<82.5 THEN WETHR{I-5}=WETHR{I-5}-1.6;                     02840099
ELSE WETHR{I-5}=WETHR{I-5}+6.1;                                                 02850099
*SOLAR RADIATION RECORDS;                                                       02860099
IF RCL{I}<=0.4 THEN  WETHR{I-5}=WETHR{I-5}+11.6;                                02870099
IF RCL{I}>0.4 AND RCL{I}<0.85 THEN WETHR{I-5}=WETHR{I-5}+5.6;                   02880099
ELSE WETHR{I-5}=WETHR{I-5}-17.2;                                                02890099
*WIND SPEED RECORDS;                                                           02900099
IF WIND{I}<=1.5 THEN  WETHR{I-5}=WETHR{I-5}+11.6;                               02910099
IF WIND{I}>1.5 AND  RCL{I}< 0.85 THEN WETHR{I-5}=WETHR{I-5}-11.6;               02920099
*PAVEMENT MATERIAL;                                                            02921099
WETHR{I-5}=WETHR{I-5}+3.9;                                                      02922099
 END;                                                                           02930099
                                                                                02940099
 IF I=6 THEN DO;                                                                02950099
* WET HOUR DUE TO FOG;                                                          02960099
IF HUR{I-1}=0 THEN DO;                                                          02970099
  IF FOG{I-1}=1 OR FOG{I-1}=2 OR FOG{I-1}=3 THEN DO;                            02980099
       IF DIFF{I-1} <=2 AND WIND{I-1}< 3 THEN DO;                               02990099
       HUR{I-1}=0.06;                                                           03000099
       END;                                                                     03010099
  END;                                                                          03020099
END;                                                                            03040099
   IF HUR{I-1}¬=0 THEN DO;                                                      03050099
   IF HUR{I-1}>=0.01 AND HUR{I-1}<0.02 THEN WETHR{25}=15;                       03060099
   IF HUR{I-1}>=0.02 AND HUR{I-1}<0.03 THEN WETHR{25}=30;                       03070099
   IF HUR{I-1}>=0.03 AND HUR{I-1}<0.05 THEN WETHR{25}=45;                       03080099
   IF HUR{I-1}>=.05 THEN WETHR{25}=60;                                          03090099
* RUNOFF TIME ASSUMED TO BE 5 MINUTES;                                          03100099
  WETHR{25}=WETHR{25}+5;                                                        03110099
* DRYING TIME; *DEVIATIONS FROM MEAN DRYING TIME OF 31.6;                       03120099
   WETHR{25}=WETHR{25}+31.6;                                                    03130099
* TEMPERATURE RECORDS;                                                          03140099
IF TEMP{I-1}<67.5 THEN  WETHR{25}=WETHR{25}+3.7;                                03150099
IF TEMP{I-1}>=67.5 AND TEMP{I-1}<82.5 THEN WETHR{25}=WETHR{25}-0.7;             03160099
ELSE WETHR{25}=WETHR{25}-3.0;                                                   03170099
* RELATIVE HUMIDITY RECORDS;                                                    03180099
IF RH{I-1}<50 THEN  WETHR{25}=WETHR{25}-4.5;                                    03190099
IF RH{I-1}>=50 AND RH{I-1}<82.5 THEN WETHR{25}=WETHR{25}-1.6;                   03200099
```

```
ELSE WETHR{25}=WETHR{25}+6.1;                                                    03210099
*SOLAR RADIATION RECORDS;                                                        03220099
IF RCL{I-1}<=0.4 THEN  WETHR{25}=WETHR{25}+11.6;                                 03230099
IF RCL{I-1}>0.4 AND RCL{I-1}<0.85 THEN WETHR{25}=WETHR{25}+5.6;                  03240099
ELSE WETHR{25}=WETHR{25}-17.2;                                                   03250099
*WIND SPEED RECORDS;                                                            03260099
IF WIND{I-1}<=1.5 THEN  WETHR{25}=WETHR{25}+11.6;                                03270099
IF WIND{I-1}>1.5 AND  RCL{I-1}< 0.85 THEN WETHR{25}=WETHR{25}-11.6;              03280099
*PAVEMENT MATERIAL;                                                             03281099
WETHR{25}=WETHR{25}+3.9;                                                         03282099
 END;                                                                            03290099
END;                                                                             03300099
                                                                                 03310099
 IF I=29 THEN DO;                                                                03320099
* WET HOUR DUE TO FOG;                                                          03330099
IF HUR{I+1}=0 THEN DO;                                                           03340099
  IF FOG{I+1}=1 OR FOG{I+1}=2 OR FOG{I+1}=3 THEN DO;                             03350099
    IF DIFF{I+1} <=2 AND WIND{I+1}< 3 THEN DO;                                   03360099
    HUR{I+1}=0.06;                                                               03370099
    END;                                                                         03390099
  END;                                                                           03400099
  ELSE WETHR{27}=0;                                                              03410099
END;                                                                             03420099
* WET HOUR DUE TO PRECIPITATION;                                                03430099
   IF HUR{I+1}¬=0 THEN DO;                                                       03440099
   IF HUR{I+1}>=0.01 AND HUR{I+1}<0.02 THEN WETHR{27}=15;                        03450099
   IF HUR{I+1}>=0.02 AND HUR{I+1}<0.03 THEN WETHR{27}=30;                        03460099
   IF HUR{I+1}>=0.03 AND HUR{I+1}<0.05 THEN WETHR{27}=45;                        03470099
   IF HUR{I+1}>=.05 THEN WETHR{27}=60;                                           03480099
   * RUNOFF TIME ASSUMED TO BE 5 MINUTES;                                       03490099
   WETHR{27}=WETHR{27}+5;                                                        03500099
   * DRYING TIME; *DEVIATIONS FROM MEAN DRYING TIME OF 31.6;                    03510099
   WETHR{27}=WETHR{27}+31.6;                                                     03520099
*  EMPERATURE RECORDS;                                                           03530099
IF TEMP{I+1}<67.5 THEN  WETHR{27}=WETHR{27}+3.7;                                 03540099
IF TEMP{I+1}>=67.5 AND TEMP{I+1}<82.5 THEN WETHR{27}=WETHR{27}-0.7;              03550099
ELSE WETHR{27}=WETHR{27}-3.0;                                                    03560099
* RELATIVE HUMIDITY RECORDS;                                                    03570099
IF RH{I+1}<50 THEN  WETHR{27}=WETHR{27}-4.5;                                     03580099
IF RH{I+1}>=50 AND RH{I+1}<82.5 THEN WETHR{27}=WETHR{27}-1.6;                    03590099
ELSE WETHR{27}=WETHR{27}+6.1;                                                    03600099
*SOLAR RADIATION RECORDS;                                                       03610099
IF RCL{I+1}<=0.4 THEN  WETHR{27}=WETHR{27}+11.6;                                 03620099
IF RCL{I+1}>0.4 AND RCL{I+1}<0.85 THEN WETHR{27}=WETHR{27}+5.6;                  03630099
ELSE WETHR{27}=WETHR{27}-17.2;                                                   03640099
*WIND SPEED RECORDS;                                                            03650099
IF WIND{I+1}<=1.5 THEN  WETHR{27}=WETHR{27}+11.6;                                03660099
IF WIND{I+1}>1.5 AND  RCL{I+1}< 0.85 THEN WETHR{27}=WETHR{27}-11.6;              03670099
*PAVEMENT MATERIAL;                                                             03671099
WETHR{27}=WETHR{27}+3.9;                                                         03672099
   END;                                                                          03680099
 END;                                                                            03690099
                                                                                 03700099
 *FOR AN HOUR WHOSE PREVIOUS HOUR HAD BUT NEXT DID NOT HAVE RAINFALL;            03710099
 IF HUR{I-1}>0 AND HUR{I}>0 AND HUR{I+1}=0 THEN DO;                              03720099
  IF I¬=6 THEN DO;                                                               03730099
     IF WETHR{I-6}>=60 AND WETHR{I-5}>=60 THEN WETHR{I-6}=60;                    03740099
     IF WETHR{I-6}>=60 AND WETHR{I-5}<60 THEN DO;                                03750099
     WETHR{I-5}=WETHR{I-5}+WETHR{I-6}-60;                                        03760099
     WETHR{I-6}=60;                                                              03770099
     IF WETHR{I-5} >=60 THEN WETHR{I-5}=60;                                      03780099
     END;                                                                        03790099
  END;                                                                           03800099
  IF I=6 THEN DO;                                                                03810099
```

```
     IF WETHR{25}>=60 AND WETHR{I-5}>=60 THEN WETHR{25}=60;          03820099
     IF WETHR{25}>=60 AND WETHR{I-5}<60 THEN DO;                     03830099
     WETHR{I-5}=WETHR{I-5}+WETHR{25}-60;                             03840099
     WETHR{25}=60;                                                   03850099
     IF WETHR{I-5}>=60 THEN WETHR{I-5}=60;                           03860099
     END;                                                            03870099
  END;                                                               03880099
END;                                                                 03890099
END;                                                                 03900099
                                                                     03910099
DATA FIVE;                                                           03920099
SET FOUR; BY DATE;                                                   03930099
DATE=DATE-1;                                                         03940099
DATE=COMPRESS(DATE);                                                 03950099
WHR26=WHR25;                                                         03960099
KEEP DATE WHR26;                                                     03970099
                                                                     03980099
PROC SORT; BY DATE;                                                  03990099
                                                                     04000099
DATA SIX;                                                            04010099
MERGE FIVE FOUR;                                                     04020099
BY DATE;                                                             04030099
                                                                     04040099
                                                                     04050099
DATA SEVEN;                                                          04060099
SET SIX;                                                             04070099
ARRAY HUR{34} HR25-HR29 HR1-HR24 HR30-HR34;                          04080099
ARRAY WETHR{28} WHR1-WHR28;                                          04090099
IF WHR26¬=. AND WHR26¬=0 THEN WHR24=WHR26;                           04100099
                                                                     04110099
DO I= 6 TO 29;                                                       04120099
* FOR AN HOUR WHOSE NEXT HOUR HAD RAINFALL BUT PREVIOUS DID NOT;     04130099
 IF HUR{I}¬=0 AND HUR{I-1}=0 AND HUR{I+1}¬=0 THEN DO;                04140099
 IF WETHR{I-5}>=60 AND I¬=29 THEN DO;                                04150099
    IF WETHR{I-4}>=60 THEN WETHR{I-5}=60;                            04160099
    IF WETHR{I-4}<60 THEN DO;                                        04170099
    WETHR{I-4}=WETHR{I-4}+WETHR{I-5}-60;                             04180099
    IF WETHR{I-4}>=60 THEN WETHR{I-4}=60;                            04190099
    WETHR{I-5}=60;                                                   04200099
    END;                                                             04210099
 END;                                                                04220099
 IF WETHR{I-5}>=60 AND I=29 THEN DO;                                 04230099
    IF WETHR{27}>=60 THEN WETHR{I-5}=60;                             04240099
    ELSE DO;                                                         04250099
    WETHR{27}=WETHR{27}+WETHR{I-5}-60;                               04260099
    WETHR{I-5}=60;                                                   04270099
    IF WETHR{27}>=60 THEN WETHR{27}=60;                              04280099
    END;                                                             04290099
 END;                                                                04300099
 END;                                                                04310099
 END;                                                                04320099
                                                                     04330099
DATA FIVEA;                                                          04340099
SET SEVEN; BY DATE;                                                  04350099
DATE=DATE+1;                                                         04360099
DATE=COMPRESS(DATE);                                                 04370099
WHR28=WHR27;                                                         04380099
KEEP DATE WHR28;                                                     04390099
                                                                     04400099
PROC SORT; BY DATE;                                                  04410099
                                                                     04420099
                                                                     04430099
DATA SIXA;                                                           04440099
```

```
 BY DATE;                                                                                                                    04460099
                                                                                                                             04470099
                                                                                                                             04480099
 DATA EIGHTA;                                                                                                                04490099
 SET SIXA;                                                                                                                   04500099
 ARRAY HUR{34} HR25-HR29 HR1-HR24 HR30-HR34;                                                                                 04510099
 ARRAY WETHR{28} WHR1-WHR28;                                                                                                 04520099
 IF WHR28¬=0 AND WHR28¬=. THEN WHR1=WHR28;                                                                                   04530099
                                                                                                                             04540099
 DO I= 6 TO 29;                                                                                                              04550099
                                                                                                                             04560099
  *FOR AN HUR WITH BOTH PREVIOUS AND NEXT RAINFALL HURS;                                                                     04570099
  IF HUR{I}¬=0 AND HUR{I-1}¬=0 AND HUR{I+1}¬=0 THEN DO;                                                                      04580099
    IF I¬=29 AND WETHR{I-4}>=60 AND WETHR{I-5}>=60 THEN WETHR{I-5}=60;                                                       04590099
    IF I=29 AND WETHR{27}>=60 AND WETHR{I-5}>=60 THEN WETHR{I-5}=60;                                                        04600099
    IF I ¬= 6 THEN DO;                                                                                                      04610099
    IF WETHR{I-6}>=60 AND WETHR{I-5}<60 THEN DO;                                                                            04620099
    WETHR{I-6}=60; WETHR{I-5}=WETHR{I-5}+WETHR{I-6}-60;                                                                     04630099
    IF WETHR{I-5}>=60 THEN WETHR{I-5}=60;                                                                                   04640099
    END;                                                                                                                    04650099
    IF I=6 THEN DO;                                                                                                         04660099
    IF WETHR{25}>=60 AND WETHR{I-5}<60 THEN DO;                                                                             04670099
    WETHR{25}=60; WETHR{I-5}=WETHR{I-5}+WETHR{25}-60;                                                                       04680099
    IF WETHR{I-5}>=60 THEN WETHR{I-5}=60;                                                                                   04690099
    END;                                                                                                                    04700099
  END;                                                                                                                      04710099
END;                                                                                                                        04720099
END;                                                                                                                        04730099
                                                                                                                            04740099
END;                                                                                                                        04750099
                                                                                                                            04760099
                                                                                                                            04770099
DATA NINE;                                                                                                                  04780099
SET EIGHTA; BY DATE;                                                                                                        04790099
DATE=DATE-1;                                                                                                                04800099
DATE=COMPRESS(DATE);                                                                                                        04810099
WHR26=WHR25;                                                                                                                04820099
KEEP DATE WHR26;                                                                                                            04830099
                                                                                                                            04840099
PROC SORT; BY DATE;                                                                                                         04850099
                                                                                                                            04860099
DATA TEN;                                                                                                                   04870099
MERGE NINE EIGHTA;                                                                                                          04880099
BY DATE;                                                                                                                    04890099
IF WHR26¬=0 AND WHR26¬=. THEN WHR24=WHR26;                                                                                  04900099
                                                                                                                            04910099
DATA ELEVEN;                                                                                                                04920099
SET TEN; BY DATE;                                                                                                           04930099
DATE=DATE+1;                                                                                                                04940099
DATE=COMPRESS(DATE);                                                                                                        04950099
WHR28=WHR27;                                                                                                                04960099
KEEP DATE WHR28;                                                                                                            04970099
                                                                                                                            04980099
PROC SORT; BY DATE;                                                                                                         04990099
                                                                                                                            04991099
                                                                                                                            05000099
DATA TWELVE;                                                                                                                05010099
MERGE ELEVEN TEN;                                                                                                           05020099
BY DATE;                                                                                                                    05030099
IF WHR28¬=0 AND WHR28¬=.  THEN WHR1=WHR28;                                                                                  05040099
ARRAY WETHR{24} WHR1-WHR24;                                                                                                 05041099
ARRAY RCL{24} RCL1-RCL24;                                                                                                   05041199
ARRAY DIFF{24} DIFF1-DIFF24;                                                                                                05041299
ARRAY TEMP{24} TEMP1-TEMP24;                                                                                                05041399
```

```
ARRAY WIND{24} WIND1-WIND24;                                                    05041499
ARRAY RH{24} RH1-RH24;                                                          05041599
                                                                                05041699
DO I=1 TO 24;                                                                   05041799
IF WETHR{I}=. THEN WETHR{I}=0;                                                  05041899
                                                                                05041999
*DAY TIME SATURATED CONDITIONS AFFECTING DRYING;                                05042099
  IF WETHR{I}¬=0 AND WETHR{I}¬=60  AND                                          05042199
      RCL{I}>0 AND DIFF{I}<2 THEN DO;                                           05042299
  M=I+2;                                                                        05042399
        DO K=I TO M UNTIL (DIFF{K}>2 OR K=24);                                  05042499
        IF (DIFF{K}<2 AND K<M) THEN WETHR{K}=60;                                05042599
            IF (DIFF{K}>2 OR K=M) AND WETHR{K}=0 THEN DO;                       05042699
          * DRYING TIME; *DEVIATIONS FROM MEAN DRYING TIME OF 31.6;              05042799
            WETHR{K}=WETHR{K}+31.6;                                             05042899
          * TEMPERATURE RECORDS;                                                05042999
            IF TEMP{K}<67.5 THEN  WETHR{K}=WETHR{K}+3.7;                        05043099
          IF TEMP{K}>=67.5 AND TEMP{K}<82.5 THEN WETHR{K}=WETHR{K}-0.7;         05043199
           ELSE WETHR{K}=WETHR{K}-3.0;                                          05043299
          * RELATIVE HUMIDITY RECORDS;                                          05043399
            IF RH{K}<50 THEN  WETHR{K}=WETHR{K}-4.5;                            05043499
            IF RH{K}>=50 AND RH{K}<82.5 THEN WETHR{K}=WETHR{K}-1.6;             05043599
            ELSE WETHR{K}=WETHR{K}+6.1;                                         05043699
          *SOLAR RADIATION RECORDS;                                             05043799
            IF RCL{K}<=0.4 THEN  WETHR{K}=WETHR{K}+11.6;                        05043899
            IF RCL{K}>0.4 AND RCL{K}<0.85 THEN WETHR{K}=WETHR{K}+5.6;           05043999
            ELSE WETHR{K}=WETHR{K}-17.2;                                        05044099
         *WIND SPEED RECORDS;                                                   05044199
            IF WIND{K}<=1.5 THEN  WETHR{K}=WETHR{K}+11.6;                       05044299
          IF WIND{K}>1.5 AND  RCL{K}< 0.85 THEN WETHR{K}=WETHR{K}-11.6;         05044399
         *PAVEMENT MATERIAL;                                                    05044499
          WETHR{K}=WETHR{K}+3.9;                                                05044599
            END;                                                                05044699
        END;                                                                    05044799
    END;                                                                        05044899
                                                                                05044999
* NIGHT TIME SATURATED DRYING CONDITIONS;                                       05045099
  IF WETHR{I}¬=0 AND WETHR{I}¬=60 AND DIFF{I}< 2                                05045199
        AND RCL{I}=0 THEN DO;                                                   05045299
        DO K=I TO 24 UNTIL (DIFF{K}>2 OR K=24);                                 05045399
        IF K=24 THEN SATUR=1;                                                   05045499
        IF (DIFF{K}<2 AND WETHR{K}=0) THEN WETHR{K}=60;                         05045599
            IF DIFF{K}>2 AND WETHR{K}=0 THEN DO;                                05045699
          * DRYING TIME; *DEVIATIONS FROM MEAN DRYING TIME OF 31.6;              05045799
            WETHR{K}=WETHR{K}+31.6;                                             05045899
          * TEMPERATURE RECORDS;                                                05045999
            IF TEMP{K}<67.5 THEN  WETHR{K}=WETHR{K}+3.7;                        05046099
          IF TEMP{K}>=67.5 AND TEMP{K}<82.5 THEN WETHR{K}=WETHR{K}-0.7;         05046199
           ELSE WETHR{K}=WETHR{K}-3.0;                                          05046299
          * RELATIVE HUMIDITY RECORDS;                                          05046399
            IF RH{K}<50 THEN  WETHR{K}=WETHR{K}-4.5;                            05046499
            IF RH{K}>=50 AND RH{K}<82.5 THEN WETHR{K}=WETHR{K}-1.6;             05046599
            ELSE WETHR{K}=WETHR{K}+6.1;                                         05046699
          *SOLAR RADIATION RECORDS;                                             05046799
            IF RCL{K}<=0.4 THEN  WETHR{K}=WETHR{K}+11.6;                        05046899
            IF RCL{K}>0.4 AND RCL{K}<0.85 THEN WETHR{K}=WETHR{K}+5.6;           05046999
            ELSE WETHR{K}=WETHR{K}-17.2;                                        05047099
         *WIND SPEED RECORDS;                                                   05047199
            IF WIND{K}<=1.5 THEN  WETHR{K}=WETHR{K}+11.6;                       05047299
          IF WIND{K}>1.5 AND  RCL{K}< 0.85 THEN WETHR{K}=WETHR{K}-11.6;         05047399
         *PAVEMENT MATERIAL;                                                    05047499
          WETHR{K}=WETHR{K}+3.9;                                                05047599
            END;                                                                05047699
        END;                                                                    05047799
```

```
    END;                                                                                                                05047899
END;                                                                                                                    05047999
                                                                                                                        05048099
                                                                                                                        05048199
PROC SORT; BY DATE;                                                                                                     05048299
                                                                                                                        05048399
DATA XIIIA;                                                                                                             05048499
  SET TWELVE; BY DATE;                                                                                                  05048599
  DATE=DATE+1;                                                                                                          05048699
  IF SATUR=. THEN SATUR=0;                                                                                              05048799
  KEEP DATE SATUR;                                                                                                      05048899
                                                                                                                        05048999
                                                                                                                        05049099
PROC SORT; BY DATE;                                                                                                     05049199
                                                                                                                        05049299
DATA XIIIB;                                                                                                             05049399
  MERGE XIIIA TWELVE;                                                                                                   05049499
  BY DATE;                                                                                                              05049599
ARRAY WETHR{24} WHR1-WHR24;                                                                                             05049699
ARRAY RCL{24} RCL1-RCL24;                                                                                               05049799
ARRAY DIFF{24} DIFF1-DIFF24;                                                                                            05049899
ARRAY TEMP{24} TEMP1-TEMP24;                                                                                            05049999
ARRAY WIND{24} WIND1-WIND24;                                                                                            05050099
ARRAY RH{24} RH1-RH24;                                                                                                  05050199
                                                                                                                        05050299
IF SATUR=1 THEN DO;                                                                                                     05050399
  DO I=1 TO 24;                                                                                                         05050499
    IF WETHR{I}=0 AND                                                                                                   05050599
      RCL{I}<0 AND DIFF{I}<2 THEN DO;                                                                                   05050699
        DO K=I TO 24 UNTIL (DIFF{K}>2 OR K=24);                                                                         05050799
        IF (DIFF{K}<2 AND K<M) THEN WETHR{K}=60;                                                                        05050899
            IF DIFF{K}>2 AND WETHR{K}=0 THEN DO;                                                                        05050999
          * DRYING TIME; *DEVIATIONS FROM MEAN DRYING TIME OF 31.6;                                                      05051099
            WETHR{K}=WETHR{K}+31.6;                                                                                     05051199
          * TEMPERATURE RECORDS;                                                                                        05051299
            IF TEMP{K}<67.5 THEN  WETHR{K}=WETHR{K}+3.7;                                                                05051399
          IF TEMP{K}>=67.5 AND TEMP{K}<82.5 THEN WETHR{K}=WETHR{K}-0.7;                                                 05051499
           ELSE WETHR{K}=WETHR{K}-3.0;                                                                                  05051599
          * RELATIVE HUMIDITY RECORDS;                                                                                  05051699
            IF RH{K}<50 THEN  WETHR{K}=WETHR{K}-4.5;                                                                    05051799
            IF RH{K}>=50 AND RH{K}<82.5 THEN WETHR{K}=WETHR{K}-1.6;                                                     05051899
            ELSE WETHR{K}=WETHR{K}+6.1;                                                                                 05051999
          *SOLAR RADIATION RECORDS;                                                                                     05052099
            IF RCL{K}<=0.4 THEN  WETHR{K}=WETHR{K}+11.6;                                                                05052199
            IF RCL{K}>0.4 AND RCL{K}<0.85 THEN WETHR{K}=WETHR{K}+5.6;                                                   05052299
            ELSE WETHR{K}=WETHR{K}-17.2;                                                                                05052399
          *WIND SPEED RECORDS;                                                                                          05052499
            IF WIND{K}<=1.5 THEN  WETHR{K}=WETHR{K}+11.6;                                                               05052599
           IF WIND{K}>1.5 AND  RCL{K}< 0.85 THEN WETHR{K}=WETHR{K}-11.6;                                                05052699
          *PAVEMENT MATERIAL;                                                                                           05052799
           WETHR{K}=WETHR{K}+3.9;                                                                                       05052899
            END;                                                                                                        05052999
         END;                                                                                                           05053099
      END;                                                                                                              05053199
    END;                                                                                                                05053299
  END;                                                                                                                  05053399
                                                                                                                        05053499
                                                                                                                        05053599
  DATA XIIIC;                                                                                                           05053699
    SET XIIIB;                                                                                                          05053799
    ARRAY WETHR{24} WHR1-WHR24;                                                                                         05053899
    ARRAY PCWHR{24} PCWHR1-PCWHR24;                                                                                     05054099
   ARRAY MISSING{24} MISS1-MISS24;                                                                                      05055099
```

```
 CWHR{I}=WETHR{I};                                                          05056199
  IF WETHR{I}¬=60 AND WETHR{I}¬=0 THEN DO;                                  05057099
  PCWHR{I}=WETHR{I}-7.8;                                                    05058099
  END;                                                                      05060099
END;                                                                        05070099
                                                                            05080099
DO I=1 TO 24;                                                               05133099
IF I=1 THEN ASUMHR=0;                                                       05134099
IF I=1 THEN PSUMHR=0;                                                       05134199
IF I=1 THEN ASUMMI=0;                                                       05135099
IF I=1 THEN PSUMMI=0;                                                       05136099
ASUMHR=SUM(ASUMHR,WETHR{I});                                                05140099
PSUMHR=SUM(PSUMHR,PCWHR{I});                                                05141099
ASUMMI=SUM(ASUMMI,MISSING{I});                                              05150099
PSUMMI=SUM(PSUMMI,MISSING{I});                                              05151099
END;                                                                        05160099
                                                                            05170099
ATOTHR=ASUMHR/60;                                                           05180099
PTOTHR=PSUMHR/60;                                                           05181099
ATOTMIS=ASUMMI;                                                             05190099
PTOTMIS=PSUMMI;                                                             05191099
                                                                            05230099
PROC SORT; BY YEAR;                                                         05240099
                                                                            05250099
PROC MEANS NOPRINT SUM N;                                                   05260099
VAR ATOTHR ASUMHR ATOTMIS PTOTHR PSUMHR PTOTMIS;                            05270099
BY YEAR;                                                                    05280099
OUTPUT OUT=STAT SUM=ATOTHUR ATOTMIN AMISSING PTOTHUR PTOTMIN PMISSING       05290099
                N=NUM NUM1 NUM2 NUM3 NUM4 NUM5 ;                            05300099
DATA XIII;                                                                  05310099
SET STAT;                                                                   05320099
LABEL NUM= NUMBER OF DAYS                                                   05330099
      ATOTHUR=TOTAL WET HOURS (ASPHLT)                                      05340099
      ATOTMIN=TOTAL WET MINUTES (ASPHLT)                                    05350099
      AMISSING=MISSING HOURS                                                05360099
      APROPRWT= PROPORTION WET (ASPHLT)                                     05370099
      PTOTHUR=TOTAL WET HOURS (PRTLND)                                      05371099
      PTOTMIN=TOTAL WET MINUTES (PRTLND)                                    05372099
      PMISSING=MISSING HOURS                                                05373099
      PPROPRWT= PROPORTION WET (PRTLND);                                    05374099
APROPRWT=(ATOTHUR)/((NUM-(AMISSING/24))*24);                                05380099
PPROPRWT=(PTOTHUR)/((NUM-(PMISSING/24))*24);                                05381099
DROP NUM1 NUM2 NUM3 NUM4 NUM5 PMISSING;                                     05390099
IF APROPRWT=0 AND PPROPRWT=0 THEN DELETE;                                   05391099
                                                                            05400099
DATA _NULL_;                                                                05401099
SET XIII;                                                                   05402099
FILE OUT NOPRINT NOTITLE MOD;                                               05403099
PUT @ 2 YEAR @ 10 NUM @ 20 ATOTHUR @ 30 ATOTMIN @ 40 AMISSING               05404099
    @ 44 APROPRWT @ 56 PTOTHUR @ 71 PTOTMIN @ 85 PPROPRWT ;                 05405099
                                                                            05406099
PROC PRINT DATA=XIII LABEL;                                                 05410099
TITLE 'WET HOUR PERCENTAGES BY MONTH FOR BATON ROUGE  PAVEMENTS';           05420099
//                                                                          05430099
```

```
//LTRC JOB (1318,05886,10,9999),'SRIKANTH',NOTIFY=IEKLEE,                00010099
//          MSGCLASS=S,CLASS=H                                           00020099
/*JOBPARM SHIFT=H                                                        00030099
//STEP1 EXEC SAS,TIME=10                                                 00040099
//PRI   DD DSN=IEKLEE.SRIKANTH.PROGRAM(PRIMARY),DISP=SHR                 00050099
//SEC   DD DSN=IEKLEE.SRIKANTH.PGM2(NORMALS),DISP=SHR                    00060099
//WLC   DD DSN=IEKLEE.WETHOURS.LAKECHAS,DISP=SHR                         00070099
//WBR   DD DSN=IEKLEE.WETHOURS.BTR,DISP=SHR                              00080099
//WNO   DD DSN=IEKLEE.WETHOURS.NEWORLNS,DISP=SHR                         00090099
//WSH   DD DSN=IEKLEE.WETHOURS.SHV,DISP=SHR                              00091099
//CSC   DD DSN=IEKLEE.DATA.FILES(TRIANGLE),DISP=SHR                      00100099
//OUT   DD DSN=IEKLEE.SRIKANTH.OUT,DISP=SHR                              00110099
//MAPS  DD DSN='SAS.R518.MAPS',DISP=SHR                                  00120099
//SYSIN DD *                                                             00130099
                                                                         00140099
OPTIONS NOCAPS;                                                          00150099
OPTIONS TLS=80 PS=60 LS=80;                                              00160099
GOPTIONS NOTEXT82;                                                       00170099
GOPTIONS DEVICE=GDDMFAM4 GDDMNICKNAME=IBM3820 GDDMTOKEN=IMG240;          00180099
                                                                         00190099
*************************************************************************;00200099
*  THIS PROGRAM CALCULATES THE DISTANCE BETWEEN A FIRST AND    *;          00210099
*  SECOND ORDER STATION AND LISTS THE NEAREST AND SECOND       *;          00220099
*  NEAREST FIRST ORDER STATION TO EACH SECOND ORDER  STATION   *;          00230099
*  DISTANCE IN ARC DISTANCE                                    *;          00240099
*                                                              *;          00250099
*                                                              *;          00260099
* AN EMPIRICAL BAYES ANALYSIS OF THE PROPORTION WET TIME       *;          00270099
* BASED ON THE WETTIME CALCULATIONS IS THEN MADE FOR EVERY     *;          00280099
* TRIANGLE FORMED BY THE MESH OF WEATHER STATIONS              *;          00290099
*                                                              *;          00300099
*                                                              *;          00310099
*************************************************************************;00320099
                                                                         00330099
DATA ONE;                                                                00340099
INFILE SEC;                                                              00350099
   INPUT INDEX NAME & $14. LATD LATM LOND LONM PARISH NORMAL;            00360099
   M=1;                                                                  00370099
                                                                         00380099
DATA TWO;                                                                00390099
INFILE PRI;                                                              00400099
   INPUT INDEX1 PARISH1 LATD1 LATM1 LOND1 LONM1 NORMAL1                  00410099
         INDEX2 PARISH2 LATD2 LATM2 LOND2 LONM2 NORMAL2                  00420099
         INDEX3 PARISH3 LATD3 LATM3 LOND3 LONM3 NORMAL3                  00430099
         INDEX4 PARISH4 LATD4 LATM4 LOND4 LONM4 NORMAL4;                 00440099
         M=1;                                                            00450099
                                                                         00460099
DATA THREE;                                                              00470099
MERGE ONE TWO;                                                           00480099
BY M;                                                                    00490099
                                                                         00500099
DROP PARISH1-PARISH4;                                                    00510099
ARRAY INDEXP{5} INDEX INDEX1-INDEX4;                                     00520099
ARRAY LATDP{5} LATD LATD1-LATD4;                                         00530099
ARRAY LATMP{5} LATM LATM1-LATM4;                                         00540099
ARRAY LONDP{5} LOND LOND1-LOND4;                                         00550099
ARRAY LONMP{5} LONM LONM1-LONM4;                                         00560099
ARRAY LATI{5} LAT1-LAT5;                                                 00570099
ARRAY LONI{5} LON1-LON5;                                                 00580099
ARRAY DIST{5} DIS1-DIS5;                                                 00590099
```

```
ARRAY DISTA{5} DISA1-DISA5;                                                    00600099
ARRAY NORMALP{5} NORMAL NORMAL1-NORMAL4;                                       00610099
                                                                               00620099
D2R=ATAN(1)/45;                                                                00630099
                                                                               00640099
DO I= 1 TO 5;                                                                  00650099
LATI{I}=D2R*(LATDP{I}+(LATMP{I}/60));                                          00660099
LONI{I}=D2R*(LONDP{I}+(LONMP{I}/60));                                          00670099
END;                                                                           00680099
                                                                               00690099
DO I=2 TO 5;                                                                   00700099
DIST{I}=SIN(LATI{1})* SIN(LATI{I});                                            00710099
DISTA{I}=COS(LATI{1})*COS(LATI{I})*COS(ABS(LONI{1}-LONI{I}));                  00720099
DIST{I}= ARCOS(DIST{I}+ DISTA{I});                                             00730099
END;                                                                           00740099
                                                                               00750099
                                                                               00760099
MINDIS=MIN(DIS2,DIS3,DIS4,DIS5);                                               00770099
SMINDIS=MAX(DIS2,DIS3,DIS4,DIS5);                                              00780099
                                                                               00790099
DO I=2 TO 5;                                                                   00800099
IF DIST{I} <= SMINDIS AND DIST{I} > MINDIS THEN                                00810099
SMINDIS=DIST{I};                                                               00820099
END;                                                                           00830099
                                                                               00840099
DO I=2 TO 5;                                                                   00850099
IF DIST{I}=MINDIS THEN NEAREST=INDEXP{I};                                      00860099
IF DIST{I}=MINDIS THEN NRNRML=NORMALP{I};                                      00870099
IF DIST{I}=SMINDIS THEN SNRNRML=NORMALP{I};                                    00880099
IF DIST{I}=SMINDIS THEN SNEAREST=INDEXP{I};                                    00890099
END;                                                                           00900099
                                                                               00910099
DROP DISA1-DISA5 LATD LOND LONM LATM LATD1-LATD4 LOND1-LOND4 LONM1-LONM400920099
   I INDEX1-INDEX4 LAT2-LAT5 LON2-LON5                                         00930099
   LATM1-LATM4 DIS1 D2R;                                                       00940099
                                                                               00950099
                                                                               00960099
DATA FOUR;                                                                     00970099
INFILE WBR;                                                                    00980099
INPUT YEAR DAYS AWETHR1 AWETMIN1 MISSING APWT1 PWETHR1 PWETMIN1 PPWT1;         00990099
KEEP YEAR AWETHR1 PWETHR1 APWT1 PPWT1;                                         01000099
PROC SORT; BY YEAR;                                                            01010099
                                                                               01020099
                                                                               01030099
DATA FIVE;                                                                     01040099
INFILE WLC;                                                                    01050099
INPUT YEAR DAYS AWETHR2 AWETMIN2 MISSING APWT2 PWETHR2 PPWT2;                  01060099
KEEP YEAR AWETHR2 PWETHR2 APWT2 PPWT2;                                         01070099
PROC SORT; BY YEAR;                                                            01080099
                                                                               01090099
DATA SIX;                                                                      01100099
INFILE WNO;                                                                    01110099
INPUT YEAR DAYS AWETHR3 AWETMIN3 MISSING APWT3 PWETHR3 PPWT3;                  01120099
KEEP YEAR AWETHR3 PWETHR3 APWT3 PPWT3;                                         01130099
PROC SORT; BY YEAR;                                                            01140099
                                                                               01150099
DATA SIXA;                                                                     01151099
INFILE WSH;                                                                    01152099
INPUT YEAR DAYS AWETHR4 AWETMIN4 MISSING APWT4 PWETHR4 PPWT4;                  01153099
KEEP YEAR AWETHR4 PWETHR4 APWT4 PPWT4;                                         01154099
PROC SORT; BY YEAR;                                                            01155099
                                                                               01156099
DATA SEVEN;                                                                    01160099
MERGE FOUR FIVE SIX SIXA;                                                      01170099
```

```
BY YEAR;

PROC TRANSPOSE PREFIX=ABWH OUT=OUT1;
VAR AWETHR1;

PROC TRANSPOSE DATA=SEVEN PREFIX=ALWH OUT=OUT2;
VAR AWETHR2;

PROC TRANSPOSE DATA=SEVEN PREFIX=ANWH OUT=OUT3;
VAR AWETHR3;

PROC TRANSPOSE DATA=SEVEN PREFIX=ASWH OUT=OUT3A;
VAR AWETHR4;

PROC TRANSPOSE DATA=SEVEN PREFIX=YR OUT=OUT4;
VAR YEAR;

PROC TRANSPOSE DATA=SEVEN PREFIX=PBWH OUT=OUT5;
VAR PWETHR1;

PROC TRANSPOSE DATA=SEVEN PREFIX=PLWH OUT=OUT6;
VAR PWETHR2;

PROC TRANSPOSE DATA=SEVEN PREFIX=PNWH OUT=OUT7;
VAR PWETHR3;

PROC TRANSPOSE DATA=SEVEN PREFIX=PSWH OUT=OUT7A;
VAR PWETHR4;

PROC TRANSPOSE DATA=SEVEN PREFIX=ABPT OUT=OUT8;
VAR APWT1;

PROC TRANSPOSE DATA=SEVEN PREFIX=ALPT OUT=OUT9;
VAR APWT2;

PROC TRANSPOSE DATA=SEVEN PREFIX=ANPT OUT=OUT10;
VAR APWT3;

PROC TRANSPOSE DATA=SEVEN PREFIX=ASPT OUT=OUT10A;
VAR APWT4;

PROC TRANSPOSE DATA=SEVEN PREFIX=PBPT OUT=OUT11;
VAR PPWT1;

PROC TRANSPOSE DATA=SEVEN PREFIX=PLPT OUT=OUT12;
VAR PPWT2;

PROC TRANSPOSE DATA=SEVEN PREFIX=PNPT OUT=OUT13;
VAR PPWT3;

PROC TRANSPOSE DATA=SEVEN PREFIX=PSPT OUT=OUT13A;
VAR PPWT4;

DATA EIGHT;
SET OUT1;
SET OUT2;
SET OUT3;
SET OUT3A;
SET OUT4;
SET OUT5;
SET OUT6;
SET OUT7;
SET OUT7A;
SET OUT8;
```

```
01180099
01190099
01200099
01210099
01220099
01230099
01240099
01250099
01260099
01270099
01280099
01281099
01282099
01283099
01290099
01300099
01310099
01320099
01330099
01340099
01350099
01360099
01370099
01380099
01390099
01400099
01401099
01402099
01403099
01410099
01420099
01430099
01440099
01450099
01460099
01470099
01480099
01490099
01491099
01492099
01493099
01500099
01510099
01520099
01530099
01540099
01550099
01560099
01570099
01580099
01581099
01582099
01583099
01590099
01600099
01610099
01620099
01621099
01630099
01640099
01650099
01660099
01661099
01670099
```

```
SET OUT9;                                                                    01680099
SET OUT10;                                                                   01690099
SET OUT10A;                                                                  01691099
SET OUT11;                                                                   01700099
SET OUT12;                                                                   01710099
SET OUT13;                                                                   01720099
SET OUT13A;                                                                  01721099
M=1;                                                                         01730099
                                                                             01740099
DATA NINE;                                                                   01750099
MERGE EIGHT THREE;                                                           01760099
BY M;                                                                        01770099
                                                                             01780099
ARRAY ABWH{50} ABWH1-ABWH50;                                                 01790099
ARRAY PBWH{50} PBWH1-PBWH50;                                                 01800099
ARRAY ABPT{50} ABPT1-ABPT50;                                                 01810099
ARRAY PBPT{50} PBPT1-PBPT50;                                                 01820099
ARRAY ALWH{50} ALWH1-ALWH50;                                                 01830099
ARRAY PLWH{50} PLWH1-PLWH50;                                                 01840099
ARRAY ALPT{50} ALPT1-ALPT50;                                                 01850099
ARRAY PLPT{50} PLPT1-PLPT50;                                                 01860099
ARRAY ANWH{50} ANWH1-ANWH50;                                                 01870099
ARRAY PNWH{50} PNWH1-PNWH50;                                                 01880099
ARRAY ANPT{50} ANPT1-ANPT50;                                                 01890099
ARRAY PNPT{50} PNPT1-PNPT50;                                                 01900099
ARRAY ASWH{50} ASWH1-ASWH50;                                                 01901099
ARRAY PSWH{50} PSWH1-PSWH50;                                                 01902099
ARRAY ASPT{50} ASPT1-ASPT50;                                                 01903099
ARRAY PSPT{50} PSPT1-PSPT50;                                                 01904099
ARRAY AWHX{50} AWHX1-AWHX50;                                                 01910099
ARRAY PWHX{50} PWHX1-PWHX50;                                                 01920099
ARRAY APWTX{50} APWTX1-APWTX50;                                              01930099
ARRAY PPWTX{50} PPWTX1-PPWTX50;                                              01940099
ARRAY YR{50} YR1-YR50;                                                       01950099
                                                                             01960099
DO I=1 TO 50;                                                                01970099
IF YR{I}¬=. AND MINDIS ¬= 0 THEN DO;                                         01980099
                                                                             02010199
 *BATON ROUGE SHREVEPORT  BEING TWO NEAREST;                                 02011099
         IF (NEAREST=549 AND SNEAREST=8440)                                  02012099
             OR (NEAREST=8440 AND SNEAREST=549) THEN DO;                     02013099
       AWHX{I}=(ABWH{I}*DIS5*(NORMAL/NORMAL1));                              02014099
       AWHX{I}=AWHX{I}+(ASWH{I}*DIS2*(NORMAL/NORMAL4));                      02015099
       AWHX{I}=AWHX{I}/(DIS5+DIS2);                                          02016099
                                                                             02017099
       PWHX{I}=(PBWH{I}*DIS5*(NORMAL/NORMAL1));                              02018099
       PWHX{I}=PWHX{I}+(PSWH{I}*DIS2*(NORMAL/NORMAL4));                      02019099
       PWHX{I}=PWHX{I}/(DIS5+DIS2);                                          02019199
                                                                             02019299
       APWTX{I}=(ABPT{I}*DIS5*(NORMAL/NORMAL1));                             02019399
       APWTX{I}=APWTX{I}+(ASPT{I}*DIS2*(NORMAL/NORMAL4));                    02019499
       APWTX{I}=APWTX{I}/(DIS5+DIS2);                                        02019599
                                                                             02019699
       PPWTX{I}=(PBPT{I}*DIS5*(NORMAL/NORMAL1));                             02019799
       PPWTX{I}=PPWTX{I}+(PSPT{I}*DIS2*(NORMAL/NORMAL4));                    02019899
       PPWTX{I}=PPWTX{I}/(DIS5+DIS2);                                        02019999
                                                                             02020099
       END;                                                                  02020199
                                                                             02021099
*SHREVEPORT  NEW ORLEANS BEING TWO NEAREST;                                  02022099
        IF (NEAREST=8440 AND SNEAREST=6660)                                  02023099
            OR (NEAREST=6660 AND SNEAREST=8440) THEN DO;                     02024099
      AWHX{I}=(ASWH{I}*DIS4*(NORMAL/NORMAL4));                               02025099
      AWHX{I}=AWHX{I}+(ANWH{I}*DIS5*(NORMAL/NORMAL3));                       02026099
```

```
      AWHX{I}=AWHX{I}/(DIS5+DIS4);                                    02027099
                                                                      02028099
      PWHX{I}=(PSWH{I}*DIS4*(NORMAL/NORMAL4));                        02029099
      PWHX{I}=PWHX{I}+(PNWH{I}*DIS5*(NORMAL/NORMAL3));                02029199
      PWHX{I}=PWHX{I}/(DIS5+DIS4);                                    02029299
                                                                      02029399
      APWTX{I}=(ASPT{I}*DIS4*(NORMAL/NORMAL4));                       02029499
      APWTX{I}=APWTX{I}+(ANPT{I}*DIS5*(NORMAL/NORMAL3));              02029599
      APWTX{I}=APWTX{I}/(DIS5+DIS4);                                  02029699
                                                                      02029799
      PPWTX{I}=(PSPT{I}*DIS4*(NORMAL/NORMAL4));                       02029899
      PPWTX{I}=PPWTX{I}+(PNPT{I}*DIS5*(NORMAL/NORMAL3));              02029999
      PPWTX{I}=PPWTX{I}/(DIS5+DIS4);                                  02030099
                                                                      02030199
      END;                                                            02030299
                                                                      02030399
* SHREVEPORT  AND LAKE CHARLES BEING THE TWO NEAREST;                 02030499
                                                                      02030599
        IF (NEAREST=5078 AND SNEAREST=8440)                           02030699
            OR (NEAREST=8440 AND SNEAREST=5078) THEN DO;              02030799
      AWHX{I}=(ALWH{I}*DIS5*(NORMAL/NORMAL2));                        02030899
      AWHX{I}=AWHX{I}+(ASWH{I}*DIS3*(NORMAL/NORMAL4));                02030999
      AWHX{I}=AWHX{I}/(DIS5+DIS3);                                    02031099
                                                                      02031199
      PWHX{I}=(PLWH{I}*DIS5*(NORMAL/NORMAL2));                        02031299
      PWHX{I}=PWHX{I}+(PSWH{I}*DIS3*(NORMAL/NORMAL4));                02031399
      PWHX{I}=PWHX{I}/(DIS5+DIS3);                                    02031499
                                                                      02031599
      APWTX{I}=(ALPT{I}*DIS5*(NORMAL/NORMAL2));                       02031699
      APWTX{I}=APWTX{I}+(ASPT{I}*DIS3*(NORMAL/NORMAL4));              02031799
      APWTX{I}=APWTX{I}/(DIS5+DIS3);                                  02031899
                                                                      02031999
      PPWTX{I}=(PLPT{I}*DIS5*(NORMAL/NORMAL2));                       02032099
      PPWTX{I}=PPWTX{I}+(PSPT{I}*DIS3*(NORMAL/NORMAL4));              02032199
      PPWTX{I}=PPWTX{I}/(DIS5+DIS3);                                  02032299
                                                                      02032399
      END;                                                            02032499
                                                                      02032599
*BATON ROUGE NEW ORLEANS BEING TWO NEAREST;                           02033099
        IF (NEAREST=549 AND SNEAREST=6660)                            02040099
            OR (NEAREST=6660 AND SNEAREST=549) THEN DO;               02050099
      AWHX{I}=(ABWH{I}*DIS4*(NORMAL/NORMAL1));                        02060099
      AWHX{I}=AWHX{I}+(ANWH{I}*DIS2*(NORMAL/NORMAL3));                02070099
      AWHX{I}=AWHX{I}/(DIS4+DIS2);                                    02080099
                                                                      02090099
      PWHX{I}=(PBWH{I}*DIS4*(NORMAL/NORMAL1));                        02100099
      PWHX{I}=PWHX{I}+(PNWH{I}*DIS2*(NORMAL/NORMAL3));                02110099
      PWHX{I}=PWHX{I}/(DIS4+DIS2);                                    02120099
                                                                      02130099
      APWTX{I}=(ABPT{I}*DIS4*(NORMAL/NORMAL1));                       02140099
      APWTX{I}=APWTX{I}+(ANPT{I}*DIS2*(NORMAL/NORMAL3));              02150099
      APWTX{I}=APWTX{I}/(DIS4+DIS2);                                  02160099
                                                                      02170099
      PPWTX{I}=(PBPT{I}*DIS4*(NORMAL/NORMAL1));                       02180099
      PPWTX{I}=PPWTX{I}+(PNPT{I}*DIS2*(NORMAL/NORMAL3));              02190099
      PPWTX{I}=PPWTX{I}/(DIS4+DIS2);                                  02200099
                                                                      02210099
      END;                                                            02220099
                                                                      02230099
* NEW ORLEANS AND LAKE CHARLES BEING THE TWO NEAREST;                 02240099
                                                                      02250099
        IF (NEAREST=5078 AND SNEAREST=6660)                           02260099
            OR (NEAREST=6660 AND SNEAREST=5078) THEN DO;              02270099
      AWHX{I}=(ALWH{I}*DIS4*(NORMAL/NORMAL2));                        02280099
```

```
      AWHX{I}=AWHX{I}+(ANWH{I}*DIS3*(NORMAL/NORMAL3));                  02290099
      AWHX{I}=AWHX{I}/(DIS4+DIS3);                                      02300099
                                                                        02310099
      PWHX{I}=(PLWH{I}*DIS4*(NORMAL/NORMAL2));                          02320099
      PWHX{I}=PWHX{I}+(PNWH{I}*DIS3*(NORMAL/NORMAL3));                  02330099
      PWHX{I}=PWHX{I}/(DIS4+DIS3);                                      02340099
                                                                        02350099
      APWTX{I}=(ALPT{I}*DIS4*(NORMAL/NORMAL2));                         02360099
      APWTX{I}=APWTX{I}+(ANPT{I}*DIS3*(NORMAL/NORMAL3));                02370099
      APWTX{I}=APWTX{I}/(DIS4+DIS3);                                    02380099
                                                                        02390099
      PPWTX{I}=(PLPT{I}*DIS4*(NORMAL/NORMAL2));                         02400099
      PPWTX{I}=PPWTX{I}+(PNPT{I}*DIS3*(NORMAL/NORMAL3));                02410099
      PPWTX{I}=PPWTX{I}/(DIS4+DIS3);                                    02420099
                                                                        02430099
      END;                                                              02440099
                                                                        02450099
        * BATON ROUGE AND LAKE CHARLES BEING THE TWO NEAREST STATIONS;  02460099
        IF (NEAREST=549 AND SNEAREST=5078)                              02470099
            OR (NEAREST=5078 AND SNEAREST=549) THEN DO;                 02480099
      AWHX{I}=(ABWH{I}*DIS3*(NORMAL/NORMAL1));                          02490099
      AWHX{I}=AWHX{I}+(ALWH{I}*DIS2*(NORMAL/NORMAL2));                  02500099
      AWHX{I}=AWHX{I}/(DIS3+DIS2);                                      02510099
                                                                        02520099
      PWHX{I}=(PBWH{I}*DIS3*(NORMAL/NORMAL1));                          02530099
      PWHX{I}=PWHX{I}+(PLWH{I}*DIS2*(NORMAL/NORMAL2));                  02540099
      PWHX{I}=PWHX{I}/(DIS3+DIS2);                                      02550099
                                                                        02560099
      APWTX{I}=(ABPT{I}*DIS3*(NORMAL/NORMAL1));                         02570099
      APWTX{I}=APWTX{I}+(ALPT{I}*DIS2*(NORMAL/NORMAL2));                02580099
      APWTX{I}=APWTX{I}/(DIS3+DIS2);                                    02590099
                                                                        02600099
      PPWTX{I}=(PBPT{I}*DIS3*(NORMAL/NORMAL1));                         02610099
      PPWTX{I}=PPWTX{I}+(PLPT{I}*DIS2*(NORMAL/NORMAL2));                02620099
      PPWTX{I}=PPWTX{I}/(DIS3+DIS2);                                    02630099
      END;                                                              02640099
 END;                                                                   02650099
 IF MINDIS=0 THEN DO;                                                   02660099
  IF NEAREST=8440 THEN DO;                                              02670099
  AWHX{I}=ASWH{I}; PWHX{I}=PSWH{I}; APWTX{I}=ASPT{I}; PPWTX{I}=PSPT{I}; 02680099
  END;                                                                  02690099
  IF NEAREST=549 THEN DO;                                               02700099
  AWHX{I}=ABWH{I}; PWHX{I}=PBWH{I}; APWTX{I}=ABPT{I}; PPWTX{I}=PBPT{I}; 02710099
  END;                                                                  02720099
  IF NEAREST=6660 THEN DO;                                              02730099
  AWHX{I}=ANWH{I}; PWHX{I}=PNWH{I}; APWTX{I}=ANPT{I}; PPWTX{I}=PNPT{I}; 02740099
  END;                                                                  02750099
  IF NEAREST=5078 THEN DO;                                              02760099
  AWHX{I}=ALWH{I}; PWHX{I}=PLWH{I}; APWTX{I}=ALPT{I}; PPWTX{I}=PLPT{I}; 02770099
  END;                                                                  02780099
 END;                                                                   02790099
END;                                                                    02800099
                                                                        02810099
Y=LAT1; X=LON1;                                                         02820099
KEEP YR1-YR50 AWHX1-AWHX50 PWHX1-PWHX50 APWTX1-APWTX50 PPWTX1-PPWTX50   02830099
     INDEX  Y  X ;                                                      02840099
                                                                        02850099
                                                                        02860099
                                                                        02870099
                                                                        02880099
GOPTIONS NOTEXT82;                                                      02890099
GOPTIONS HSIZE=12 VSIZE=12;                                             02900099
GOPTIONS DEVICE=BEN9215;                                                02910099
                                                                        02920099
```

```
 DATA ONE;                                                          02930099
 LENGTH COUNTY 4;                                                   02940099
 SET NINE;                                                          02950099
 COUNTY=-1;                                                         02960099
 IF INDEX=4842.2 OR INDEX=1707.2 THEN DELETE;                       02970099
                                                                    02980099
DATA MAP;                                                           02990099
SET MAPS.COUNTIES;                                                  03000099
IF STATE=22;                                                        03010099
                                                                    03020099
DATA MAP1;                                                          03030099
SET MAP ONE;                                                        03040099
                                                                    03050099
PROC GPROJECT DATA=MAP1                                             03060099
              OUT=PROJCOM                                           03070099
              PROJECT=ALBERS                                        03080099
              PARALEL1=29.5                                         03090099
              PARALEL2=45.5;                                        03100099
ID COUNTY;                                                          03110099
                                                                    03120099
DATA PROJLOU PROJSTA ;                                              03130099
SET PROJCOM;                                                        03140099
IF COUNTY = -1 THEN OUTPUT PROJSTA;                                 03150099
ELSE OUTPUT PROJLOU;                                                03160099
                                                                    03170099
PROC G3GRID DATA=PROJSTA OUT=GRIDONE OUTTRI=TRIONE;                 03180099
GRID X*Y=YR1/ AXIS1=-0.039 TO 0.04 BY 0.05                          03190099
                AXIS2=-0.18 TO -0.075 BY 0.05;                      03200099
                                                                    03210099
PROC MEANS NOPRINT DATA=TRIONE;                                     03220099
OUTPUT OUT=OUTA N=N;                                                03230099
BY TRIANGLE;                                                        03240099
                                                                    03250099
DATA TRITWO;                                                        03260099
MERGE TRIONE OUTA;                                                  03270099
BY TRIANGLE;                                                        03280099
IF N ¬= 3 THEN DELETE;                                              03290099
X=ROUND(X,0.000001);                                                03300099
Y=ROUND(Y,0.000001);                                                03310099
                                                                    03320099
                                                                    03330099
                                                                    03340099
DATA PROJST;                                                        03350099
SET PROJSTA;                                                        03360099
X=ROUND(X,0.000001);                                                03370099
Y=ROUND(Y,0.000001);                                                03380099
DROP COUNTY SEGMENT STATE DENSITY;                                  03390099
                                                                    03400099
PROC SORT ; BY X Y;                                                 03410099
                                                                    03420099
                                                                    03430099
PROC TRANSPOSE DATA=TRITWO PREFIX=X OUT=OUTB1;                      03440099
VAR X;                                                              03450099
BY TRIANGLE;                                                        03460099
                                                                    03470099
                                                                    03472099
PROC TRANSPOSE DATA=TRITWO PREFIX=Y OUT=OUTB2;                      03480099
VAR Y;                                                              03490099
BY TRIANGLE;                                                        03500099
                                                                    03510099
DATA COMB2;                                                         03520099
SET OUTB1;                                                          03530099
SET OUTB2;                                                          03540099
BY TRIANGLE;                                                        03550099
```

```
DROP _NAME_ _LABEL_;                                                    03560099
                                                                        03570099
PROC SORT; BY X1 Y1;                                                    03590099
                                                                        03600099
                                                                        03610099
                                                                        03620099
DATA COMB3;                                                             03630099
SET PROJST;                                                             03640099
X1=X; Y1=Y;                                                             03650099
ARRAY AWHX{50} AWHX1-AWHX50;                                            03660099
ARRAY AW1X{50} AW1X1-AW1X50;                                            03670099
ARRAY PWHX{50} PWHX1-PWHX50;                                            03680099
ARRAY PW1X{50} PW1X1-PW1X50;                                            03690099
ARRAY APWTX{50} APWTX1-APWTX50;                                         03700099
ARRAY APW1X{50} APW1X1-APW1X50;                                         03710099
ARRAY PPWTX{50} PPWTX1-PPWTX50;                                         03720099
ARRAY PPW1X{50} PPW1X1-PPW1X50;                                         03730099
ARRAY YR{50} YR1-YR50;                                                  03740099
                                                                        03750099
DO I=1 TO 50;                                                           03760099
AW1X{I}=AWHX{I};                                                        03770099
PW1X{I}=PWHX{I};                                                        03780099
APW1X{I}=APWTX{I};                                                      03790099
PPW1X{I}=PPWTX{I};                                                      03800099
END;                                                                    03810099
                                                                        03820099
DROP X Y AWHX1-AWHX50 PWHX1-PWHX50 APWTX1-APWTX50 PPWTX1-PPWTX50;       03830099
                                                                        03840099
PROC SORT; BY X1 Y1;                                                    03850099
                                                                        03860099
DATA COMB4; MERGE COMB2 COMB3; BY X1 Y1;                                03870099
if x2=. and y2=. and x3=. and y3=. then delete;                          03871099
                                                                        03880099
PROC SORT DATA=COMB4; BY X2 Y2;                                         03890099
                                                                        03893099
                                                                        03900099
DATA COMB5; SET PROJST; X2=X; Y2=Y;                                     03910099
ARRAY AWHX{50} AWHX1-AWHX50;                                            03920099
ARRAY AW2X{50} AW2X1-AW2X50;                                            03930099
ARRAY PWHX{50} PWHX1-PWHX50;                                            03940099
ARRAY PW2X{50} PW2X1-PW2X50;                                            03950099
ARRAY APWTX{50} APWTX1-APWTX50;                                         03960099
ARRAY APW2X{50} APW2X1-APW2X50;                                         03970099
ARRAY PPWTX{50} PPWTX1-PPWTX50;                                         03980099
ARRAY PPW2X{50} PPW2X1-PPW2X50;                                         03990099
ARRAY YR{50} YR1-YR50;                                                  04000099
                                                                        04010099
DO I=1 TO 50;                                                           04020099
AW2X{I}=AWHX{I};                                                        04030099
PW2X{I}=PWHX{I};                                                        04040099
APW2X{I}=APWTX{I};                                                      04050099
PPW2X{I}=PPWTX{I};                                                      04060099
END;                                                                    04070099
                                                                        04080099
DROP X Y AWHX1-AWHX50 PWHX1-PWHX50 APWTX1-APWTX50 PPWTX1-PPWTX50        04090099
     YR1-YR50;                                                          04100099
PROC SORT; BY X2 Y2;                                                    04110099
                                                                        04120099
DATA COMB6; MERGE COMB4 COMB5; BY X2 Y2;                                04130099
if x3=. and y3=. and x1=. and y1=. then delete;                          04131099
                                                                        04132099
PROC SORT; BY X3 Y3;                                                    04140099
                                                                        04141099
                                                                        04150099
```

```
DATA COMB7; SET PROJST; X3=X; Y3=Y;                                    04170099
ARRAY AWHX{50} AWHX1-AWHX50;                                           04180099
ARRAY AW3X{50} AW3X1-AW3X50;                                           04190099
ARRAY PWHX{50} PWHX1-PWHX50;                                           04200099
ARRAY PW3X{50} PW3X1-PW3X50;                                           04210099
ARRAY APWTX{50} APWTX1-APWTX50;                                        04220099
ARRAY APW3X{50} APW3X1-APW3X50;                                        04230099
ARRAY PPWTX{50} PPWTX1-PPWTX50;                                        04240099
ARRAY PPW3X{50} PPW3X1-PPW3X50;                                        04250099
ARRAY YR{50} YR1-YR50;                                                 04260099
                                                                       04270099
DO I=1 TO 50;                                                          04280099
AW3X{I}=AWHX{I};                                                       04290099
PW3X{I}=PWHX{I};                                                       04300099
APW3X{I}=APWTX{I};                                                     04310099
PPW3X{I}=PPWTX{I};                                                     04320099
END;                                                                   04330099
                                                                       04340099
DROP X Y AWHX1-AWHX50 PWHX1-PWHX50 APWTX1-APWTX50 PPWTX1-PPWTX50       04350099
     YR1-YR50;                                                         04360099
PROC SORT; BY X3 Y3;                                                   04370099
                                                                       04380099
DATA COMB8; MERGE COMB6 COMB7; BY X3 Y3;                               04390099
if x2=. and y2=. and x1=. and y1=. then delete;                         04391099
                                                                       04392099
                                                                       04412099
ARRAY AW1X{50} AW1X1-AW1X50;                                           04420099
ARRAY AW2X{50} AW2X1-AW2X50;                                           04430099
ARRAY AW3X{50} AW3X1-AW3X50;                                           04440099
ARRAY PW1X{50} PW1X1-PW1X50;                                           04450099
ARRAY PW2X{50} PW2X1-PW2X50;                                           04460099
ARRAY PW3X{50} PW3X1-PW3X50;                                           04470099
ARRAY APW1X{50} APW1X1-APW1X50;                                        04480099
ARRAY APW2X{50} APW2X1-APW2X50;                                        04490099
ARRAY APW3X{50} APW3X1-APW3X50;                                        04500099
ARRAY PPW1X{50} PPW1X1-PPW1X50;                                        04510099
ARRAY PPW2X{50} PPW2X1-PPW2X50;                                        04520099
ARRAY PPW3X{50} PPW3X1-PPW3X50;                                        04530099
ARRAY YR{50} YR1-YR50;                                                 04540099
ARRAY AAW1X{50} AAW1X1-AAW1X50;                                        04550099
ARRAY AAW2X{50} AAW2X1-AAW2X50;                                        04560099
ARRAY AAW3X{50} AAW3X1-AAW3X50;                                        04570099
ARRAY APAW1X{50} APAW1X1-APAW1X50;                                     04580099
ARRAY APAW2X{50} APAW2X1-APAW2X50;                                     04590099
ARRAY APAW3X{50} APAW3X1-APAW3X50;                                     04600099
ARRAY AAPW1X{50} AAPW1X1-AAPW1X50;                                     04610099
ARRAY AAPW2X{50} AAPW2X1-AAPW2X50;                                     04620099
ARRAY AAPW3X{50} AAPW3X1-AAPW3X50;                                     04630099
ARRAY APPW1X{50} APPW1X1-APPW1X50;                                     04640099
ARRAY APPW2X{50} APPW2X1-APPW2X50;                                     04650099
ARRAY APPW3X{50} APPW3X1-APPW3X50;                                     04660099
ARRAY AWHMN{50} AWHMN1-AWHMN50;                                        04670099
ARRAY PWHMN{50} PWHMN1-PWHMN50;                                        04680099
ARRAY APWMN{50} APWMN1-APWMN50;                                        04690099
ARRAY PPWMN{50} PPWMN1-PPWMN50;                                        04700099
                                                                       04710099
 B1=X2*Y3-X3*Y2; B2=X3*Y1-X1*Y3; B3=X1*Y2-X2*Y1;                        04720099
 C1=Y2-Y3; C2=Y3-Y1; C3=Y1-Y2;                                         04730099
 D1=X3-X2; D2=X1-X3; D3=X2-X1;                                         04740099
                                                                       04750099
 AREA= (B1+B2+B3)/2;                                                   04760099
                                                                       04770099
 DO I=1 TO 50;                                                         04780099
                                                                       04790099
```

```
 AAW1X{I}=(B1*AW1X{I}+B2*AW2X{I}+B3*AW3X{I})/(2*AREA);                    04800099
 AAW2X{I}=(C1*AW1X{I}+C2*AW2X{I}+C3*AW3X{I})/(2*AREA);                    04810099
 AAW3X{I}=(D1*AW1X{I}+D2*AW2X{I}+D3*AW3X{I})/(2*AREA);                    04820099
                                                                          04830099
 AWHMN{I}=AAW1X{I}+(AAW2X{I}*(X1+X2+X3)/3)+(AAW3X{I}*(Y1+Y2+Y3)/3);       04840099
                                                                          04850099
                                                                          04860099
 APAW1X{I}=(B1*PW1X{I}+B2*PW2X{I}+B3*PW3X{I})/(2*AREA);                   04870099
 APAW2X{I}=(C1*PW1X{I}+C2*PW2X{I}+C3*PW3X{I})/(2*AREA);                   04880099
 APAW3X{I}=(D1*PW1X{I}+D2*PW2X{I}+D3*PW3X{I})/(2*AREA);                   04890099
                                                                          04900099
 PWHMN{I}=APAW1X{I}+(APAW2X{I}*(X1+X2+X3)/3)+(APAW3X{I}*(Y1+Y2+Y3)/3);    04910099
                                                                          04920099
 AAPW1X{I}=(B1*APW1X{I}+B2*APW2X{I}+B3*APW3X{I})/(2*AREA);                04930099
 AAPW2X{I}=(C1*APW1X{I}+C2*APW2X{I}+C3*APW3X{I})/(2*AREA);                04940099
 AAPW3X{I}=(D1*APW1X{I}+D2*APW2X{I}+D3*APW3X{I})/(2*AREA);                04950099
                                                                          04960099
 APWMN{I}=AAPW1X{I}+(AAPW2X{I}*(X1+X2+X3)/3)+(AAPW3X{I}*(Y1+Y2+Y3)/3);    04970099
                                                                          04980099
 APPW1X{I}=(B1*PPW1X{I}+B2*PPW2X{I}+B3*PPW3X{I})/(2*AREA);                04990099
 APPW2X{I}=(C1*PPW1X{I}+C2*PPW2X{I}+C3*PPW3X{I})/(2*AREA);                05000099
 APPW3X{I}=(D1*PPW1X{I}+D2*PPW2X{I}+D3*PPW3X{I})/(2*AREA);                05010099
                                                                          05020099
 PPWMN{I}=APPW1X{I}+(APPW2X{I}*(X1+X2+X3)/3)+(APPW3X{I}*(Y1+Y2+Y3)/3);    05030099
                                                                          05040099
 END;                                                                     05050099
                                                                          05060099
 IF AWHMN1=. THEN DELETE;                                                 05070099
                                                                          05080099
 KEEP TRIANGLE APWMN1-APWMN50                                             05090099
      PPWMN1-PPWMN50;                                                     05100099
                                                                          05110099
PROC SORT; BY TRIANGLE;                                                   05120099
                                                                          05130099
                                                                          05140099
DATA EMPBAYES;                                                            05150099
 SET COMB8;                                                               05160099
 BY TRIANGLE;                                                             05170099
 G=_N_;                                                                   05180099
 IF G <= 3 THEN G=G;                                                      05190099
 ELSE DO;                                                                 05200099
   G=MOD(G,3);                                                            05210099
   IF G=0 THEN G=3;                                                       05220099
 END;                                                                     05230099
                                                                          05240099
PROC SORT; BY G;                                                          05250099
                                                                          05260099
DATA EMPBAYI;                                                             05270099
 SET EMPBAYES; BY G;                                                      05280099
 ARRAY APWMN{50} APWMN1-APWMN50;                                          05290099
 ARRAY PPWMN{50} PPWMN1-PPWMN50;                                          05300099
 P=0; K=0; SUMA=0; SUMP=0;                                                05310099
 DO I=1 TO 50;                                                            05320099
   IF APWMN{I}¬=. THEN DO;                                                05330099
    SUMA=SUMA+APWMN{I};                                                   05340099
    K=K+1;                                                                05350099
   END;                                                                   05360099
   IF PPWMN{I}¬=. THEN DO;                                                05370099
    SUMP=SUMP+PPWMN{I};                                                   05380099
    P=P+1;                                                                05390099
   END;                                                                   05400099
 END;                                                                     05410099
                                                                          05420099
 XIABAR=SUMA/K;  XIPBAR=SUMP/P;                                           05430099
```

```
PROC SORT; BY G;

PROC MEANS NOPRINT MEAN N;
VAR APWMN1-APWMN50 PPWMN1-PPWMN50;
BY G;
OUTPUT OUT=NAT MEAN=AMN1-AMN50 PMN1-PMN50 N=AN1-AN50 PN1-PN50;

PROC SORT DATA=NAT; BY G;

DATA EMPBAYII; MERGE EMPBAYI NAT; BY G;

ARRAY APWMN{50} APWMN1-APWMN50;
ARRAY PPWMN{50} PPWMN1-PPWMN50;
ARRAY AMN{50} AMN1-AMN50;
ARRAY PMN{50} PMN1-PMN50;
ARRAY AN{50} AN1-AN50;
ARRAY PN{50} PN1-PN50;
SUMSQA=0;
SUMSQP=0;

DO I= 1 TO 50;
   IF APWMN{I}¬=. THEN DO;
    SUMSQA=SUMSQA+(APWMN{I}-(AMN{I}/AN{I}))**2;
   END;
   IF PPWMN{I}¬=. THEN DO;
    SUMSQP=SUMSQP+(PPWMN{I}-(PMN{I}/PN{I}))**2;
   END;
END;

PROC SORT; BY G;

PROC MEANS SUM N;
 VAR SUMSQA SUMSQP;
 BY G;
 OUTPUT OUT=VARIANCE SUM= SUMSQNA SUMSQNP N=NA NP;

PROC SORT DATA=VARIANCE; BY G;

DATA EMBYIII;
MERGE VARIANCE EMPBAYII;
BY G;

 ARRAY APWMN{50} APWMN1-APWMN50;
 ARRAY PPWMN{50} PPWMN1-PPWMN50;
 ARRAY MUEBA{50} MUEBA1-MUEBA50;
 ARRAY VREBA{50} VREBA1-VREBA50;
 ARRAY MUEBP{50} MUEBP1-MUEBP50;
 ARRAY VREBP{50} VREBP1-VREBP50;
SGSFHA=SUMSQNA/(NA*(NA-1)*K);
SGSFHP=SUMSQNP/(NP*(NP-1)*P);

MUPIAH=XIABAR; MUPIPH=XIPBAR;
SSQA=SUMSQA; SSQP=SUMSQP;

SGSPIHA=(SSQA/(K-1))-SGSFHA;
IF SGSPIHA < 0 THEN SGSPIHA=0;

SGSPIHP=(SSQP/(P-1))-SGSFHP;
IF SGSPIHP < 0 THEN SGSPIHP=0;

BHATA=((K-3)/(K-1))*SGSFHA/(SGSFHA+SGSPIHA);
```

```
05440099
05450099
05460099
05470099
05480099
05490099
05500099
05510099
05520099
05530099
05540099
05550099
05560099
05570099
05580099
05590099
05600099
05610099
05620099
05630099
05640099
05650099
05660099
05670099
05680099
05690099
05700099
05710099
05720099
05730099
05740099
05750099
05760099
05770099
05780099
05790099
05800099
05810099
05820099
05830099
05840099
05850099
05860099
05870099
05880099
05890099
05900099
05910099
05920099
05930099
05940099
05950099
05960099
05970099
05980099
05990099
06000099
06010099
06020099
06030099
06040099
06050099
06060099
```

```
DO I= 1 TO 50;                                                          06080099
 IF APWMN{I} ¬=. THEN DO;                                               06090099
 FACT=APWMN{I}-MUPIAH;                                                  06100099
 MUEBA{I}=APWMN{I}-BHATA*FACT;                                          06110099
 VREBA{I}=SGSFHA*(1-((K-1)/K)*BHATA) + (2/(K-3))*BHATA**2*FACT**2;      06120099
 END;                                                                   06130099
                                                                        06140099
 IF PPWMN{I} ¬=. THEN DO;                                               06150099
 FACTP= PPWMN{I}-MUPIPH;                                                06160099
 MUEBP{I}=PPWMN{I}-BHATP*FACTP;                                         06170099
 VREBP{I}=SGSFHP*(1-((P-1)/P)*BHATP) + (2/(P-3))*BHATP**2*FACTP**2;     06180099
 END;                                                                   06190099
 END;                                                                   06200099
                                                                        06210099
 DO I=2 TO 50;                                                          06220099
 * GET OUT THE LATEST YEARS  EMPIRICAL BAYES ESTIMATE;                  06230099
                                                                        06240099
 IF APWMN{I}=. AND APWMN{I-1} ¬=. THEN DO;                              06250099
 VREBAL=VREBA{I-1}; APWMNL=APWMN{I-1};                                  06260099
 MUEBAL=MUEBA{I-1};                                                     06270099
 END;                                                                   06280099
                                                                        06290099
 IF PPWMN{I}=. AND PPWMN{I-1} ¬=. THEN DO;                              06300099
 VREBPL=VREBP{I-1}; PPWMNL=PPWMN{I-1};                                  06310099
 MUEBPL=MUEBP{I-1};                                                     06320099
 END;                                                                   06330099
END;                                                                    06340099
                                                                        06350099
*PROBABILITY THAT A TRIANGLE HAS EXPERIENCED  LESS NUMBER OF WET HOURS  06360099
  THAN THE REGIONAL NUMBER OF WET HOURS;                                06370099
                                                                        06380099
* P{THETHAI <= OBSERVED AVERAGE  | Y};                                 06390099
* TRANSFORM FIRST INTO 0,1 NORMAL;                                      06400099
KEEP TRIANGLE  G MUPIAH APWMNL PPWMNL MUEBAL VREBAL MUEBPL VREBPL       06410099
     MUPIPH;                                                            06420099
                                                                        06430099
PROC MEANS NOPRINT MEAN;                                                06440099
VAR APWMNL PPWMNL;                                                      06450099
BY G;                                                                   06460099
OUTPUT OUT=STATSII MEAN= APWRMN PPWRMN;                                 06470099
                                                                        06480099
PROC SORT DATA=STATSII; BY G;                                           06490099
                                                                        06500099
DATA EMBYIV;                                                            06510099
MERGE EMBYIII STATSII; BY G;                                            06520099
                                                                        06530099
ZA=(APWRMN-MUEBAL)/SQRT(VREBAL);                                        06540099
ZP=(PPWRMN-MUEBPL)/SQRT(VREBPL);                                        06550099
                                                                        06560099
WTA=PROBNORM(ZA);                                                       06570099
WTP=PROBNORM(ZP);                                                       06580099
                                                                        06590099
LABEL MUEBAL=EMP. BAYES EST. OF MEAN                                    06600099
      VREBAL=EMO. BAYES EST. OF VAR.                                    06610099
      APWMNL=LATEST YEAR PROPORTION WET TIME                            06620099
      WTA=PROB. OF BEING LESS THAN REGIONALMEAN;                        06630099
LABEL MUEBPL=EMP. BAYES EST. OF MEAN                                    06640099
      VREBPL=EMO. BAYES EST. OF VAR.                                    06650099
      PPWMNL=LATEST YEAR PROPORTION WET TIME                            06660099
      WTP=PROB. OF BEING LESS THAN REGIONALMEAN;                        06670099
                                                                        06680099
KEEP MUEBPL MUEBAL VREBPL VREBAL TRIANGLE PPWMNL APWMNL WTA WTP         06690099
     ;                                                                  06700099
                                                                        06710099
```

```
PROC SORT; BY TRIANGLE;                                                    06720099
                                                                           06730099
PROC PRINT;                                                                06740099
                                                                           06750099
 DATA CSEC;                                                                06760099
INFILE CSC;                                                                06770099
INPUT CONTROL SECTION TRIANGLE;                                            06780099
                                                                           06790099
PROC SORT; BY TRIANGLE;                                                    06800099
                                                                           06810099
DATA TRIANGLE;                                                             06820099
MERGE CSEC EMBYIV; BY TRIANGLE;                                            06830099
IF APWMNL=. THEN DELETE;                                                   06840099
                                                                           06850099
PROC PRINT;                                                                06860099
                                                                           06870099
DATA _NULL_;                                                               06880099
SET TRIANGLE;                                                              06890099
FILE OUT NOPRINT NOTITLE;                                                  06900099
PUT CONTROL SECTION TRIANGLE MUEBAL MUEBPL VREBAL VREBPL PPWMNL APWMNL     06910099
    WTA WTP;                                                               06920099
     */                                                                    06921099
//                                                                         06930099
```

```
//LTRC  JOB (1318,05886,10,9999),'SRIKANTH',NOTIFY=IEKLEE,                00010099
//          MSGCLASS=S,CLASS=H                                            00020099
/*JOBPARM SHIFT=H                                                         00030099
//STEP1 EXEC SAS606,TIME=10                                               00040099
//ACCI  DD DSN=IEKLEE.ACC.MASTER88,DISP=SHR                               00050099
//INP   DD DSN=IEKLEE.SRIKANTH.OUT,DISP=SHR                               00240099
//OUT   DD DSN=IEKLEE.NEW2.SASDATA,DISP=OLD
//SYSIN DD *                                                              00250099
                                                                          00260099
 * TO RUN FOR 1989 AND 1990 CHANGE THE ACCI LINE IN THE JCL
   STATEMENT TO ACC.MASTER89 AND ACC.MASTER90 RESPECTIVELY;

*******************************************************************;      00270099
*   THIS PROGRAM  FLAGS CLUSTERS BY THE WET ACCIDENT              *;      00280099
*    CRITERIONS DEVELOPED FOR ACCIDENT DATA IN LOUISIANA          *;      00290099
*    BY THE RATE QUALITY CONTROL METHOD C2  AND THE SECOND        *;      00290099
*    BAYESIAN CRITERIONS ON THE BASIS OF SIMULATIONS RUN BEFORE   *;      00290099
*******************************************************************;      00300099
                                                                          00310099
*READS ALL INPUT DATA FROM MASTER ACCIDENT FILE;                         00320099
*OPTIONS TIME=MAX TLS=80 PS=60 LS=80;                                    00330099
OPTIONS NOCAPS;                                                           00340099
GOPTIONS NOTEXT82;                                                        00350099
GOPTIONS DEVICE=GDDMFAM4 GDDMNICKNAME=IBM3820 GDDMTOKEN=IMG240;           00360099
                                                                          00370099
DATA MASACC;                                                              00380099
    INFILE ACCI RECFM=FB LRECL=225 BLKSIZE=11250;                         00390099
    INPUT                                                                 00400099
        YEAR 13                                                           00410099
        PARISH 17-18                                                      00420099
        MP 25-28 1                                                        00430099
        HNUM $ 29-33                                                      00440099
        HTYPE 34                                                          00450099
        SURCON $ 55                                                       00460099
        RDCON $ 58                                                        00470099
        WEATH $ 60                                                        00480099
        INT  78                                                           00490099
        HCLASS 87                                                         00500099
        CONTROL 101-103                                                   00510099
        SECTION 104-105                                                   00520099
        CLOCA  106-109 2
        BCLM 115-118 2                                                    00530099
        ECLM 119-122 2                                                    00540099
        PAVTYP 123-124                                                    00550099
        PAVWID 125-126                                                    00560099
        ADT 134-139 ;                                                     00570099
                                                                          00580099
    *ELIMINATES UNWANTED DATA;                                            00590099
                                                                          00600099
    IF BCLM<0 OR ECLM<0 THEN DELETE;                                      00610099
    IF BCLM=0 AND ECLM=0 THEN DELETE;                                     00620099
    IF BCLM >= ECLM THEN LENGTH=(BCLM-ECLM);                              00630099
    ELSE LENGTH=(ECLM-BCLM);                                              00640099
    IF BCLM>=ECLM THEN DO;                                                00650099
    TEMP=BCLM;                                                            00660099
    BCLM=ECLM;                                                            00670099
    ECLM=TEMP;                                                            00680099
    DROP TEMP;                                                            00690099
    END;                                                                  00700099
    IF ADT<=0 THEN DELETE;                                                00710099
```

```
IF MP < 0 THEN DELETE;                                                  00720099
IF INT ¬=0 AND INT ¬=1 THEN DELETE;                                     00730099
IF CONTROL<0 OR SECTION <0 THEN DELETE;                                 00740099
                                                                        00750099
*CLASSIFIES INTERSECTIONS;                                              00760099
IF INT=1 THEN LTYPE='INTERSECTION';                                     00770099
                                                                        00780099
*CLASSIFIES SECTIONS;                                                   00790099
ELSE IF INT=0 THEN LTYPE='SECTION';                                     00800099
                                                                        00810099
IF LTYPE='INTERSECTION' THEN LENGTH=1;                                  00820099
IF LTYPE='INTERSECTION' THEN DELETE;
                                                                        00830099
*ELIMINATES LOCAL AND ARTERIAL ROADS;                                   00840099
IF HTYPE ¬=1 AND HTYPE ¬=2 AND HTYPE ¬=3 THEN DELETE;                   00850099
                                                                        00860099
                                                                        00870099
  LABEL SECTION= SECTION NUMBER                                         00880099
        BCLM=BEGINNING OF SECTION (LOG MILE)                            00890099
        ECLM=END OF SECTION (LOG MILE)                                  00900099
        HNUM=HIGHWAY NUMBER                                             00910099
        MP=MILEPOST                                                     00920099
        LTYPE=LOCATION TYPE                                             00930099
        SURCON=SURFACE CONDITION                                        00940099
        HCLASS=HIGHWAY CLASS;                                           00950099
                                                                        00960099
OUTPUT;                                                                 00970099
                                                                        00980099
*SORTS THE ACCIDENT DATA AS GIVEN IN BY STATEMENT;                      00990099
                                                                        01000099
PROC SORT  ;                                                            01010099
BY LTYPE HCLASS CONTROL SECTION BCLM ;                                  01020099
/*                                                                      01030099
PROC FORMAT;                                                            01040099
VALUE $SURCON                                                           01050099
(MAX=100 MIN=0)                                                         01060099
'A'='DRY CONDITION'                                                     01070099
'B'='RAINY CONDITION'                                                   01080099
'C'='MUDDY CONDITION'                                                   01090099
'D'='SNOWY/ICY CONDITION'                                               01100099
'E'='OTHER CONDITIONS'                                                  01110099
OTHER='MISSING VALUES' ;                                                01120099
                                                                        01130099
PROC FORMAT;                                                            01140099
VALUE HCL                                                               01150099
(MAX=100 MIN=0)                                                         01160099
1='RURL 2 LANE'                                                         01170099
2='RURL OTHER'                                                          01180099
3='RURL MULT LN DIV'                                                    01190099
4='RURL INTERSTATE'                                                     01200099
5='URBN 2 LANE'                                                         01210099
6='URBN OTHER'                                                          01220099
7='URBN MULT LN DIV'                                                    01230099
8='URBN INTERSTATE'                                                     01240099
OTHER='MISSING VALUES';                                                 01250099
  */                                                                    01260099
                                                                        01270099
                                                                        01280099

DATA MASACC1;
SET MASACC;
*CHECKS FOR WET CRITERION;                                              01330099
IF SURCON='A' THEN DELETE;                                              01340099
                                                                        01350099
* CHECKS FOR MISSING AND UNKNOWN SURFACE CONDITIONS;                    01360099
```

```
IF SURCON='E' OR SURCON= ' ' THEN DO;                                   01370099
  IF RDCON¬='J' AND RDCON ¬= 'K' THEN DELETE;                           01380099
  IF WEATH='A' OR WEATH='F' OR WEATH='G' THEN DELETE;                   01390099
END;                                                                    01400099
```



```
 IF PARISH=01 THEN DISTRICT=03;
 IF PARISH=02 THEN DISTRICT=07;
 IF PARISH=03 THEN DISTRICT=61;
 IF PARISH=04 THEN DISTRICT=61;
 IF PARISH=05 THEN DISTRICT=08;
 IF PARISH=06 THEN DISTRICT=07;
 IF PARISH=07 THEN DISTRICT=04;
 IF PARISH=08 THEN DISTRICT=04;
 IF PARISH=09 THEN DISTRICT=04;
 IF PARISH=10 THEN DISTRICT=07;
 IF PARISH=11 THEN DISTRICT=58;
 IF PARISH=12 THEN DISTRICT=07;
 IF PARISH=13 THEN DISTRICT=58;
 IF PARISH=14 THEN DISTRICT=04;
 IF PARISH=15 THEN DISTRICT=58;
 IF PARISH=16 THEN DISTRICT=04;
 IF PARISH=17 THEN DISTRICT=61;
 IF PARISH=18 THEN DISTRICT=05;
 IF PARISH=19 THEN DISTRICT=61;
 IF PARISH=20 THEN DISTRICT=07;
 IF PARISH=21 THEN DISTRICT=58;
 IF PARISH=22 THEN DISTRICT=08;
 IF PARISH=23 THEN DISTRICT=03;
 IF PARISH=24 THEN DISTRICT=61;
 IF PARISH=25 THEN DISTRICT=05;
 IF PARISH=26 THEN DISTRICT=02;
 IF PARISH=27 THEN DISTRICT=07;
 IF PARISH=28 THEN DISTRICT=03;
 IF PARISH=29 THEN DISTRICT=03;
 IF PARISH=30 THEN DISTRICT=58;
 IF PARISH=31 THEN DISTRICT=05;
 IF PARISH=32 THEN DISTRICT=62;
 IF PARISH=33 THEN DISTRICT=05;
 IF PARISH=34 THEN DISTRICT=05;
 IF PARISH=35 THEN DISTRICT=08;
 IF PARISH=36 THEN DISTRICT=02;
 IF PARISH=37 THEN DISTRICT=05;
 IF PARISH=38 THEN DISTRICT=02;
 IF PARISH=39 THEN DISTRICT=61;
 IF PARISH=40 THEN DISTRICT=08;
 IF PARISH=41 THEN DISTRICT=04;
 IF PARISH=42 THEN DISTRICT=05;
 IF PARISH=43 THEN DISTRICT=08;
 IF PARISH=44 THEN DISTRICT=02;
 IF PARISH=45 THEN DISTRICT=02;
 IF PARISH=46 THEN DISTRICT=62;
 IF PARISH=47 THEN DISTRICT=61;
 IF PARISH=48 THEN DISTRICT=02;
 IF PARISH=49 THEN DISTRICT=03;
 IF PARISH=50 THEN DISTRICT=03;
 IF PARISH=51 THEN DISTRICT=03;
 IF PARISH=52 THEN DISTRICT=62;
 IF PARISH=53 THEN DISTRICT=62;
 IF PARISH=54 THEN DISTRICT=58;
 IF PARISH=55 THEN DISTRICT=02;
 IF PARISH=56 THEN DISTRICT=05;
 IF PARISH=57 THEN DISTRICT=03;
 IF PARISH=58 THEN DISTRICT=08;
```

```
     IF PARISH=59 THEN DISTRICT=62;
     IF PARISH=60 THEN DISTRICT=04;
     IF PARISH=61 THEN DISTRICT=61;
     IF PARISH=62 THEN DISTRICT=05;
     IF PARISH=63 THEN DISTRICT=61;
     IF PARISH=64 THEN DISTRICT=08;

    PROC SORT;
    BY HCLASS HNUM;

PROC FASTCLUS OUT=TR2 MEAN=TR3 CLUSTER=MEMBER MAXC=250 RADIUS=0.5
REPLACE=PART MAXITER=10 CONV=0.001 DRIFT NOPRINT IMPUTE;
    VAR CLOCA;
    BY HCLASS HNUM;

 PROC SORT DATA=TR2;
 BY HCLASS HNUM MEMBER;

 PROC MEANS DATA=TR2 NOPRINT MIN MAX;
 VAR CLOCA;
 BY  HCLASS HNUM MEMBER ;
 OUTPUT OUT=CLMEAN MIN=BCLUS MAX=ECLUS;

 PROC SORT DATA=CLMEAN;
 BY  HCLASS HNUM MEMBER;

DATA CHANGE;
SET TR3;
AVCLOCA=CLOCA;
DROP CLOCA;

 PROC SORT DATA=CHANGE;
 BY HCLASS HNUM MEMBER ;

 DATA OTR3;
 MERGE TR2 CHANGE;
 BY HCLASS HNUM MEMBER ;

  DATA NEWTR3;
  MERGE CLMEAN OTR3;
  BY HCLASS HNUM MEMBER;
  LTYPE='4';

    IF FIRST.MEMBER THEN TOT_ACC=0;
     TOT_ACC+1;

    IF LAST.MEMBER THEN DO;
     LENGTH=1;
    OUTPUT;
    END;

  PROC SORT; BY CONTROL SECTION;

  DATA TRI;                                                          01440099
  INFILE INP;                                                        01450099
  INPUT CONTROL SECTION TRIANGLE MUEBAL MUEBPL VREBAL VREBPL PPWMNL  01460099
        APWMNL WTA WTP;                                              01470099
                                                                     01480099
  PROC SORT; BY CONTROL SECTION;                                     01490099
                                                                     01500099
                                                                     01510099
  DATA RAT; MERGE NEWTR3 TRI; BY CONTROL SECTION;                    01520099
  IF MUEBAL=. THEN DELETE;                                           01530099
      IF YEAR=. AND ADT=. THEN DELETE;                               01540099
```

```
  IF PAVTYP=50                                                          01550099
  OR PAVTYP=60 THEN DO;                                                 01560099
  MU=MUEBAL; VR=VREBAL; WMN=APWMNL; PR=WTA;                             01570099
  END;                                                                  01580099
                                                                        01590099
  ELSE DO;                                                              01600099
  MU=MUEBPL; VR=VREBPL; WMN=PPWMNL; PR=WTP;                             01610099
  END;                                                                  01620099
                                                                        01630099
  DROP TRIANGLE MUEBAL MUEBPL VREBAL VREBPL WTA WTP APWMNL PPWMNL;      01640099
                                                                        01641099
  PROC SORT; BY LTYPE HCLASS CONTROL SECTION;                           01641199
                                                                        01641299
DATA  RATE; SET RAT;                                                    01690099
BY LTYPE HCLASS CONTROL SECTION;                                        01700099

                                                                        01430099
                                                                        01710099
*CALCULATES THE ACCIDENTS PER MILLION VEHICLE MILES;                    01760099
                                                                        01820099
 *EXPOSURE IS DEPENDENT ON PROPORTION WET TIME AND LENGTH;              01830099
 ADT=ADT*LENGTH*WMN;                                                    01840099
 MVM=365*ADT/1000000;                                                   01850099
 ACC_MVM=TOT_ACC/MVM;                                                   01860099
 ACC_MIL=TOT_ACC/LENGTH;                                                01870099
 VOL=ADT*365;                                                           01880099
 SQRVOL=VOL**2;                                                         01890099
 TRU_RAT=TOT_ACC/(MU*LENGTH*ADT*365);                                   01900099
 NIBYVI=TOT_ACC/VOL;                                                    01910099
 NIBYSQVI=TOT_ACC/SQRVOL;                                               01920099
 ONEBYVOL=1/VOL;                                                        01930099
                                                                        01970099
LABEL  TOT_ACC=ACCIDENTS PER INTERSECTION                               01980099
       ACC_MIL=ACCIDENTS PER MILE                                       01990099
       ACC_MVM=ACCIDENTS PER MILLION VEHICLE MILES;                     02000099
                                                                        02010099
 OUTPUT;                                                                02020099
                                                                        02040099
 PROC MEANS NOPRINT SUM MEAN STD VAR RANGE N;                           02050099
 VAR TOT_ACC ACC_MVM ADT NIBYVI NIBYSQVI ONEBYVOL TRU_RAT;              02060099
 BY LTYPE HCLASS;                                                       02070099
 *FORMAT HCLASS HCL.;                                                   02080099
 OUTPUT OUT=STATAV SUM=STATSUM MVMSUM ADTSUM NVSUM                      02090099
 MEAN=STATOT STAMVM STAADT STANV STASQNV HARVOL MNTR                    02100099
 STD=STASTD MVMSTD ADTSTD NVSTD NVSSTD ONESTD TRSTD                     02110099
 VAR=TOTVAR MVMVAR ADTVAR NVVAR NSVVAR                                  02120099
 RANGE=TOTRG MVMRG ADTRG NVRNG NSRNG                                    02121099
 N=TN MN AN NVN NSN;                                                    02122099
                                                                        02130099
 PROC SORT DATA=STATAV;                                                 02140099
 BY LTYPE HCLASS;                                                       02150099
                                                                        02170099
  DATA COMBINE;                                                         02180099
  MERGE RATE STATAV;                                                    02190099
  BY LTYPE HCLASS;                                                      02200099
                                                                        02210099
                                                                        02211099
  * FLAGGING BY C2  RATE QUALITY CONTROL METHOD;                        02340099
                                                                        02350099
  SIG_NBYV=STATSUM/(ADTSUM*365);                                        02360099
                                                                        02370099
   C2=SIG_NBYV+(1.645 *SQRT(SIG_NBYV/VOL)) + (1/(2*VOL));               02380099
                                                                        02410099
  *FLAGGING BY BAYESIAN CRITERION 2;                                    02420099
```

```
                                                                    02430099
      BETA=STANV/NVVAR;                                             02440099
      ALPHA=BETA*STANV;                                             02450099
                                                                    02460099
      BETAI=BETA+VOL;                                               02470099
      ALPHAI=ALPHA + TOT_ACC;                                       02480099
                                                                    02490099
      ADJ_STA1= STANV*BETAI;                                        02500099
      ADJ_STA2= SIG_NBYV*BETAI;                                     02510099
                                                                    02520099
                                                                    02590099
      B2= 1-PROBGAM(ADJ_STA2,ALPHAI);                               02600099

    KEEP DISTRICT ACC_MVM TOT_ACC CONTROL SECTION HNUM HTYPE LTYPE
         PARISH BCLM BCLUS ECLUS HCLASS ADT C2 B2;


DATA OUT.SNCLS88;
SET COMBINE;
                                                                    01680099

//                                                                  05440099
```

```
//LTRC  JOB (1318,05886,10,9999),'SRIKANTH',NOTIFY=IEKLEE,                00010099
//          MSGCLASS=S                                                    00020099
/*JOBPARM SHIFT=H                                                         00030099
//STEP1 EXEC SAS,TIME=10                                                  00040099
//ACCI   DD DSN=IEKLEE.ACC.MASTER88,DISP=SHR                              00050099
//OUT    DD DSN=IEKLEE.SRIKANTH.OUT,DISP=SHR                              00240099
//OUT2   DD DSN=IEKLEE.ALOK.OUTPUT,DISP=NEW
//SYSIN DD *                                                              00250099
                                                                          00260099
* TO RUN THE PROGRAM FOR 1989 AND 1990 CHANGE ACCI STATEMENT IN THE JCL
   TO  ACC.MASTER89 AND ACC.MASTER90 RESPECTIVELY;

*********************************************************************;    00270099
*   THIS PROGRAM  FLAGS INTERSECTIONS AND SECTIONS FOR THE BAYESIAN;      00280099
*    CRITERIONS DEVELOPED FOR ACCIDENT DATA IN LOUISIANA          *;      00290099
*********************************************************************;    00300099
                                                                          00310099
*READS ALL INPUT DATA FROM MASTER ACCIDENT FILE;                          00320099
*OPTIONS TIME=MAX TLS=80 PS=60 LS=80;                                     00330099
OPTIONS NOCAPS;                                                           00340099
GOPTIONS NOTEXT82;                                                        00350099
GOPTIONS DEVICE=GDDMFAM4 GDDMNICKNAME=IBM3820 GDDMTOKEN=IMG240;           00360099
                                                                          00370099
DATA MASACC;                                                              00380099
    INFILE ACCI RECFM=FB LRECL=225 BLKSIZE=11250;                         00390099
    INPUT                                                                 00400099
        YEAR 13                                                           00410099
        PARISH 17-18                                                      00420099
        MP 25-28 1                                                        00430099
        HNUM $ 29-33                                                      00440099
        HTYPE 34                                                          00450099
        SURCON $ 55                                                       00460099
        RDCON $ 58                                                        00470099
        WEATH $ 60                                                        00480099
        INT  78                                                           00490099
        HCLASS 87                                                         00500099
        CONTROL 101-103                                                   00510099
        SECTION 104-105                                                   00520099
        CLOCA  106-109 2
        BCLM 115-118 2                                                    00530099
        ECLM 119-122 2                                                    00540099
        PAVTYP 123-124                                                    00550099
        PAVWID 125-126                                                    00560099
        ADT 134-139 ;                                                     00570099
                                                                          00580099
    *ELIMINATES UNWANTED DATA;                                            00590099
                                                                          00600099
    IF BCLM<0 OR ECLM<0 THEN DELETE;                                      00610099
    IF BCLM=0 AND ECLM=0 THEN DELETE;                                     00620099
    IF BCLM >= ECLM THEN LENGTH=(BCLM-ECLM);                              00630099
    ELSE LENGTH=(ECLM-BCLM);                                              00640099
    IF BCLM>=ECLM THEN DO;                                                00650099
    TEMP=BCLM;                                                            00660099
    BCLM=ECLM;                                                            00670099
    ECLM=TEMP;                                                            00680099
    DROP TEMP;                                                            00690099
    END;                                                                  00700099
    IF ADT<=0 THEN DELETE;                                                00710099
    IF MP < 0 THEN DELETE;                                                00720099
    IF INT ¬=0 AND INT ¬=1 THEN DELETE;                                   00730099
```

```
   IF CONTROL<0 OR SECTION <0 THEN DELETE;                          00740099
                                                                    00750099
   *CLASSIFIES INTERSECTIONS;                                       00760099
   IF INT=1 THEN LTYPE='INTERSECTION';                              00770099
                                                                    00780099
   *CLASSIFIES SECTIONS;                                            00790099
   ELSE IF INT=0 THEN LTYPE='SECTION';                              00800099
                                                                    00810099
   IF LTYPE='INTERSECTION' THEN LENGTH=1;                           00820099
                                                                    00830099
   *ELIMINATES LOCAL AND ARTERIAL ROADS;                            00840099
   IF HTYPE ¬=1 AND HTYPE ¬=2 AND HTYPE ¬=3 THEN DELETE;            00850099
                                                                    00860099
                                                                    00870099
     LABEL SECTION= SECTION NUMBER                                  00880099
           BCLM=BEGINNING OF SECTION (LOG MILE)                     00890099
           ECLM=END OF SECTION (LOG MILE)                           00900099
           HNUM=HIGHWAY NUMBER                                      00910099
           MP=MILEPOST                                              00920099
           LTYPE=LOCATION TYPE                                      00930099
           SURCON=SURFACE CONDITION                                 00940099
           HCLASS=HIGHWAY CLASS;                                    00950099
                                                                    00960099
  OUTPUT;                                                           00970099
                                                                    00980099
  *SORTS THE ACCIDENT DATA AS GIVEN IN BY STATEMENT;                 00990099
                                                                    01000099
  PROC SORT  ;                                                      01010099
BY LTYPE HCLASS CONTROL SECTION BCLM ;                              01020099
                                                                    01030099
/*PROC FORMAT;                                                      01040099
  VALUE $SURCON                                                     01050099
  (MAX=100 MIN=0)                                                   01060099
  'A'='DRY CONDITION'                                               01070099
  'B'='RAINY CONDITION'                                             01080099
  'C'='MUDDY CONDITION'                                             01090099
  'D'='SNOWY/ICY CONDITION'                                         01100099
  'E'='OTHER CONDITIONS'                                            01110099
  OTHER='MISSING VALUES' ;                                          01120099
                                                                    01130099
  PROC FORMAT;                                                      01140099
  VALUE HCL                                                         01150099
  (MAX=100 MIN=0)                                                   01160099
  1='RURL 2 LANE'                                                   01170099
  2='RURL OTHER'                                                    01180099
  3='RURL MULT LN DIV'                                              01190099
  4='RURL INTERSTATE'                                               01200099
  5='URBN 2 LANE'                                                   01210099
  6='URBN OTHER'                                                    01220099
  7='URBN MULT LN DIV'                                              01230099
  8='URBN INTERSTATE'                                               01240099
  OTHER='MISSING VALUES';                                           01250099
    */                                                              01260099
                                                                    01270099
                                                                    01280099
  DATA MASACC1;
  SET MASACC;
  *CHECKS FOR WET CRITERION;                                        01330099
  IF SURCON='A' THEN DELETE;                                        01340099
                                                                    01350099
  * CHECKS FOR MISSING AND UNKNOWN SURFACE CONDITIONS;              01360099
  IF SURCON='E' OR SURCON= ' ' THEN DO;                             01370099
    IF RDCON¬='J' AND RDCON ¬= 'K' THEN DELETE;                     01380099
    IF WEATH='A' OR WEATH='F' OR WEATH='G' THEN DELETE;             01390099
```

```
IF PARISH=01 THEN DISTRICT=03;
IF PARISH=02 THEN DISTRICT=07;
IF PARISH=03 THEN DISTRICT=61;
IF PARISH=04 THEN DISTRICT=61;
IF PARISH=05 THEN DISTRICT=08;
IF PARISH=06 THEN DISTRICT=07;
IF PARISH=07 THEN DISTRICT=04;
IF PARISH=08 THEN DISTRICT=04;
IF PARISH=09 THEN DISTRICT=04;
IF PARISH=10 THEN DISTRICT=07;
IF PARISH=11 THEN DISTRICT=58;
IF PARISH=12 THEN DISTRICT=07;
IF PARISH=13 THEN DISTRICT=58;
IF PARISH=14 THEN DISTRICT=04;
IF PARISH=15 THEN DISTRICT=58;
IF PARISH=16 THEN DISTRICT=04;
IF PARISH=17 THEN DISTRICT=61;
IF PARISH=18 THEN DISTRICT=05;
IF PARISH=19 THEN DISTRICT=61;
IF PARISH=20 THEN DISTRICT=07;
IF PARISH=21 THEN DISTRICT=58;
IF PARISH=22 THEN DISTRICT=08;
IF PARISH=23 THEN DISTRICT=03;
IF PARISH=24 THEN DISTRICT=61;
IF PARISH=25 THEN DISTRICT=05;
IF PARISH=26 THEN DISTRICT=02;
IF PARISH=27 THEN DISTRICT=07;
IF PARISH=28 THEN DISTRICT=03;
IF PARISH=29 THEN DISTRICT=03;
IF PARISH=30 THEN DISTRICT=58;
IF PARISH=31 THEN DISTRICT=05;
IF PARISH=32 THEN DISTRICT=62;
IF PARISH=33 THEN DISTRICT=05;
IF PARISH=34 THEN DISTRICT=05;
IF PARISH=35 THEN DISTRICT=08;
IF PARISH=36 THEN DISTRICT=02;
IF PARISH=37 THEN DISTRICT=05;
IF PARISH=38 THEN DISTRICT=02;
IF PARISH=39 THEN DISTRICT=61;
IF PARISH=40 THEN DISTRICT=08;
IF PARISH=41 THEN DISTRICT=04;
IF PARISH=42 THEN DISTRICT=05;
IF PARISH=43 THEN DISTRICT=08;
IF PARISH=44 THEN DISTRICT=02;
IF PARISH=45 THEN DISTRICT=02;
IF PARISH=46 THEN DISTRICT=62;
IF PARISH=47 THEN DISTRICT=61;
IF PARISH=48 THEN DISTRICT=02;
IF PARISH=49 THEN DISTRICT=03;
IF PARISH=50 THEN DISTRICT=03;
IF PARISH=51 THEN DISTRICT=03;
IF PARISH=52 THEN DISTRICT=62;
IF PARISH=53 THEN DISTRICT=62;
IF PARISH=54 THEN DISTRICT=58;
IF PARISH=55 THEN DISTRICT=02;
IF PARISH=56 THEN DISTRICT=05;
IF PARISH=57 THEN DISTRICT=03;
IF PARISH=58 THEN DISTRICT=08;
IF PARISH=59 THEN DISTRICT=62;
IF PARISH=60 THEN DISTRICT=04;
IF PARISH=61 THEN DISTRICT=61;
```

```
   IF PARISH=62 THEN DISTRICT=05;
   IF PARISH=63 THEN DISTRICT=61;
   IF PARISH=64 THEN DISTRICT=08;

  PROC SORT;
  BY LTYPE HCLASS CONTROL SECTION;


DATA MASACC3;
SET MASACC1;
BY LTYPE HCLASS CONTROL SECTION BCLM;
IF LTYPE='INTERSECTION' THEN LTYPE='1';
IF LTYPE='SECTION' THEN LTYPE='2';

  IF FIRST.SECTION THEN TOT_ACC=0;
   TOT_ACC+1;

  IF LAST.SECTION THEN DO;
   LENGTH=1;
  OUTPUT;
  END;

PROC SORT; BY CONTROL SECTION;

DATA TRI;                                                              01440099
INFILE OUT;                                                            01450099
INPUT CONTROL SECTION TRIANGLE MUEBAL MUEBPL VREBAL VREBPL PPWMNL       01460099
      APWMNL WTA WTP;                                                  01470099
                                                                       01480099
PROC SORT; BY CONTROL SECTION;                                         01490099
                                                                       01500099
                                                                       01510099
DATA RAT; MERGE MASACC3 TRI; BY CONTROL SECTION;                       01520099
IF MUEBAL=. THEN DELETE;                                               01530099
    IF YEAR=. AND ADT=. THEN DELETE;                                   01540099
     IF PAVTYP=50                                                      01550099
     OR PAVTYP=60 THEN DO;                                             01560099
     MU=MUEBAL; VR=VREBAL; WMN=APWMNL; PR=WTA;                         01570099
     END;                                                              01580099
                                                                       01590099
     ELSE DO;                                                          01600099
     MU=MUEBPL; VR=VREBPL; WMN=PPWMNL; PR=WTP;                         01610099
     END;                                                              01620099
                                                                       01630099
     DROP TRIANGLE MUEBAL MUEBPL VREBAL VREBPL WTA WTP APWMNL PPWMNL;  01640099
                                                                       01641099
     PROC SORT; BY LTYPE HCLASS CONTROL SECTION;                       01641199
                                                                       01641299
   DATA  RATE; SET RAT;                                                01690099
   BY LTYPE HCLASS CONTROL SECTION;                                    01700099

                                                                       01430099
                                                                       01710099
   *CALCULATES THE ACCIDENTS PER MILLION VEHICLE MILES;                 01760099
                                                                       01820099
    *EXPOSURE IS DEPENDENT ON PROPORTION WET TIME AND LENGTH;           01830099
    ADT=ADT*LENGTH*WMN;                                                01840099
    MVM=365*ADT/1000000;                                               01850099
    ACC_MVM=TOT_ACC/MVM;                                               01860099
    ACC_MIL=TOT_ACC/LENGTH;                                            01870099
    VOL=ADT*365;                                                       01880099
    SQRVOL=VOL**2;                                                     01890099
    TRU_RAT=TOT_ACC/(MU*LENGTH*ADT*365);                               01900099
    NIBYVI=TOT_ACC/VOL;                                                01910099
```

```
      NIBYSQVI=TOT_ACC/SQRVOL;                                          01920099
      ONEBYVOL=1/VOL;                                                   01930099
                                                                        01970099
     LABEL  TOT_ACC=ACCIDENTS PER INTERSECTION                          01980099
            ACC_MIL=ACCIDENTS PER MILE                                  01990099
            ACC_MVM=ACCIDENTS PER MILLION VEHICLE MILES;                02000099
                                                                        02010099
      OUTPUT;                                                           02020099
                                                                        02040099
      PROC MEANS NOPRINT SUM MEAN STD VAR RANGE N;                      02050099
      VAR TOT_ACC ACC_MVM ADT NIBYVI NIBYSQVI ONEBYVOL TRU_RAT;         02060099
      BY LTYPE HCLASS;                                                  02070099
      *FORMAT HCLASS HCL.;                                              02080099
      OUTPUT OUT=STATAV SUM=STATSUM MVMSUM ADTSUM NVSUM                 02090099
      MEAN=STATOT STAMVM STAADT STANV STASQNV HARVOL MNTR               02100099
      STD=STASTD MVMSTD ADTSTD NVSTD NVSSTD ONESTD TRSTD                02110099
      VAR=TOTVAR MVMVAR ADTVAR NVVAR NSVVAR                             02120099
      RANGE=TOTRG MVMRG ADTRG NVRNG NSRNG                               02121099
      N=TN MN AN NVN NSN;                                               02122099
                                                                        02130099
      PROC SORT DATA=STATAV;                                            02140099
      BY LTYPE HCLASS;                                                  02150099
                                                                        02170099
       DATA COMBINE;                                                    02180099
       MERGE RATE STATAV;                                               02190099
       BY LTYPE HCLASS;                                                 02200099
                                                                        02210099
                                                                        02211099
       * FLAGGING BY C2;                                                02340099
                                                                        02350099
       SIG_NBYV=STATSUM/(ADTSUM*365);                                   02360099
                                                                        02370099
        C2=SIG_NBYV+(1.645 *SQRT(SIG_NBYV/VOL)) + (1/(2*VOL));          02380099
                                                                        02410099
       *FLAGGING BY BAYESIAN CRITERION 2;                               02420099
                                                                        02430099
       BETA=STANV/NVVAR;                                                02440099
       ALPHA=BETA*STANV;                                                02450099
                                                                        02460099
       BETAI=BETA+VOL;                                                  02470099
       ALPHAI=ALPHA + TOT_ACC;                                          02480099
                                                                        02490099
       ADJ_STA1= STANV*BETAI;                                           02500099
       ADJ_STA2= SIG_NBYV*BETAI;                                        02510099
                                                                        02520099
                                                                        02590099
       B2= 1-PROBGAM(ADJ_STA2,ALPHAI);                                  02600099

      KEEP DISTRICT ACC_MVM TOT_ACC CONTROL SECTION HNUM HTYPE LTYPE
           PARISH BCLM HCLASS ADT C2 B2;

DATA OUT2.SNINT88;
SET COMBINE ;                                                           01680099

//                                                                      05440099
```

```
//LTRC  JOB (1318,05886,10,9999),'SRIKANTH',NOTIFY=IEKLEE,                00010099
//          MSGCLASS=S,CLASS=H                                            00020099
/*JOBPARM SHIFT=H                                                         00030099
//STEP1 EXEC SAS606,TIME=10                                               00040099
//ACCI  DD DSN=IEKLEE.ACC.MASTER88,DISP=SHR                               00050099
//OUT   DD DSN=IEKLEE.SRIKANTH.OUT,DISP=SHR                               00240099
//OUT2  DD DSN=IEKLEE.NEW2.SASDATA,DISP=OLD
//SYSIN DD *                                                              00250099
                                                                          00260099

* TO RUN THIS FOR 1989 AND 1990 CHANGE THE JCL ACCI STATEMENT TO
  ACC.MASTER89 AND ACC.MASTER90 RESPECTIVELY;

*********************************************************************;    00270099
*   THIS PROGRAM  FLAGS SPOTS BY THE WET ACCIDENT*;                       00280099
*    CRITERIONS DEVELOPED FOR ACCIDENT DATA IN LOUISIANA          *;      00290099
*********************************************************************;    00300099
                                                                          00310099
*READS ALL INPUT DATA FROM MASTER ACCIDENT FILE;                          00320099
*OPTIONS TIME=MAX TLS=80 PS=60 LS=80;                                     00330099
OPTIONS NOCAPS;                                                           00340099
GOPTIONS NOTEXT82;                                                        00350099
GOPTIONS DEVICE=GDDMFAM4 GDDMNICKNAME=IBM3820 GDDMTOKEN=IMG240;           00360099
                                                                          00370099
DATA MASACC;                                                              00380099
    INFILE ACCI RECFM=FB LRECL=225 BLKSIZE=11250;                         00390099
    INPUT                                                                 00400099
        YEAR 13                                                           00410099
        PARISH 17-18                                                      00420099
        MP 25-28 1                                                        00430099
        HNUM $ 29-33                                                      00440099
        HTYPE 34                                                          00450099
        SURCON $ 55                                                       00460099
        RDCON $ 58                                                        00470099
        WEATH $ 60                                                        00480099
        INT  78                                                           00490099
        HCLASS 87                                                         00500099
        CONTROL 101-103                                                   00510099
        SECTION 104-105                                                   00520099
        CLOCA  106-109 2                                                  00530099
        BCLM 115-118 2                                                    00540099
        ECLM 119-122 2                                                    00550099
        PAVTYP 123-124                                                    00560099
        PAVWID 125-126                                                    00570099
        ADT 134-139 ;                                                     00580099
                                                                          00590099
    *ELIMINATES UNWANTED DATA;                                            00600099
                                                                          00610099
    IF BCLM<0 OR ECLM<0 THEN DELETE;                                      00620099
    IF BCLM=0 AND ECLM=0 THEN DELETE;                                     00630099
    IF BCLM >= ECLM THEN LENGTH=(BCLM-ECLM);                              00640099
    ELSE LENGTH=(ECLM-BCLM);                                              00650099
    IF BCLM>=ECLM THEN DO;                                                00660099
    TEMP=BCLM;                                                            00670099
    BCLM=ECLM;                                                            00680099
    ECLM=TEMP;                                                            00690099
    DROP TEMP;                                                            00700099
    END;                                                                  00710099
    IF ADT<=0 THEN DELETE;                                                00720099
    IF MP < 0 THEN DELETE;
```

```
  IF INT ¬=0 AND INT ¬=1 THEN DELETE;                                  00730099
  IF CONTROL<0 OR SECTION <0 THEN DELETE;                              00740099
                                                                       00750099
  *CLASSIFIES INTERSECTIONS;                                           00760099
  IF INT=1 THEN LTYPE='INTERSECTION';                                  00770099
                                                                       00780099
  *CLASSIFIES SECTIONS;                                                00790099
  ELSE IF INT=0 THEN LTYPE='SECTION';                                  00800099
                                                                       00810099
  IF LTYPE='INTERSECTION' THEN LENGTH=1;                               00820099
                                                                       00830099
  *ELIMINATES LOCAL AND ARTERIAL ROADS;                                00840099
  IF HTYPE ¬=1 AND HTYPE ¬=2 AND HTYPE ¬=3 THEN DELETE;                00850099
                                                                       00860099
                                                                       00870099
    LABEL SECTION= SECTION NUMBER                                      00880099
          BCLM=BEGINNING OF SECTION (LOG MILE)                         00890099
          ECLM=END OF SECTION (LOG MILE)                               00900099
          HNUM=HIGHWAY NUMBER                                          00910099
          MP=MILEPOST                                                  00920099
          LTYPE=LOCATION TYPE                                          00930099
          SURCON=SURFACE CONDITION                                     00940099
          HCLASS=HIGHWAY CLASS;                                        00950099
                                                                       00960099
  OUTPUT;                                                              00970099
                                                                       00980099
  *SORTS THE ACCIDENT DATA AS GIVEN IN BY STATEMENT;                    00990099
                                                                       01000099
  PROC SORT  ;                                                         01010099
BY LTYPE HCLASS HNUM MP ;                                              01020099
                                                                       01030099
 /* PROC FORMAT;                                                       01040099
  VALUE $SURCON                                                        01050099
  (MAX=100 MIN=0)                                                      01060099
  'A'='DRY CONDITION'                                                  01070099
  'B'='RAINY CONDITION'                                                01080099
  'C'='MUDDY CONDITION'                                                01090099
  'D'='SNOWY/ICY CONDITION'                                            01100099
  'E'='OTHER CONDITIONS'                                               01110099
  OTHER='MISSING VALUES' ;                                             01120099
                                                                       01130099
  PROC FORMAT;                                                         01140099
  VALUE HCL                                                            01150099
  (MAX=100 MIN=0)                                                      01160099
  1='RURL 2 LANE'                                                      01170099
  2='RURL OTHER'                                                       01180099
  3='RURL MULT LN DIV'                                                 01190099
  4='RURL INTERSTATE'                                                  01200099
  5='URBN 2 LANE'                                                      01210099
  6='URBN OTHER'                                                       01220099
  7='URBN MULT LN DIV'                                                 01230099
  8='URBN INTERSTATE'                                                  01240099
  OTHER='MISSING VALUES';                                              01250099
    */                                                                 01260099
                                                                       01270099
                                                                       01280099
  DATA MASACC1;
  SET MASACC;

  *CHECKS FOR WET CRITERION;                                           01330099
  IF SURCON='A' THEN DELETE;                                           01340099
                                                                       01350099
  * CHECKS FOR MISSING AND UNKNOWN SURFACE CONDITIONS;                 01360099
  IF SURCON='E' OR SURCON= ' ' THEN DO;                                01370099
```

```
  IF RDCON¬='J' AND RDCON ¬= 'K' THEN DELETE;                    01380099
  IF WEATH='A' OR WEATH='F' OR WEATH='G' THEN DELETE;            01390099
END;                                                             01400099
```



```
 IF PARISH=01 THEN DISTRICT=03;
 IF PARISH=02 THEN DISTRICT=07;
 IF PARISH=03 THEN DISTRICT=61;
 IF PARISH=04 THEN DISTRICT=61;
 IF PARISH=05 THEN DISTRICT=08;
 IF PARISH=06 THEN DISTRICT=07;
 IF PARISH=07 THEN DISTRICT=04;
 IF PARISH=08 THEN DISTRICT=04;
 IF PARISH=09 THEN DISTRICT=04;
 IF PARISH=10 THEN DISTRICT=07;
 IF PARISH=11 THEN DISTRICT=58;
 IF PARISH=12 THEN DISTRICT=07;
 IF PARISH=13 THEN DISTRICT=58;
 IF PARISH=14 THEN DISTRICT=04;
 IF PARISH=15 THEN DISTRICT=58;
 IF PARISH=16 THEN DISTRICT=04;
 IF PARISH=17 THEN DISTRICT=61;
 IF PARISH=18 THEN DISTRICT=05;
 IF PARISH=19 THEN DISTRICT=61;
 IF PARISH=20 THEN DISTRICT=07;
 IF PARISH=21 THEN DISTRICT=58;
 IF PARISH=22 THEN DISTRICT=08;
 IF PARISH=23 THEN DISTRICT=03;
 IF PARISH=24 THEN DISTRICT=61;
 IF PARISH=25 THEN DISTRICT=05;
 IF PARISH=26 THEN DISTRICT=02;
 IF PARISH=27 THEN DISTRICT=07;
 IF PARISH=28 THEN DISTRICT=03;
 IF PARISH=29 THEN DISTRICT=03;
 IF PARISH=30 THEN DISTRICT=58;
 IF PARISH=31 THEN DISTRICT=05;
 IF PARISH=32 THEN DISTRICT=62;
 IF PARISH=33 THEN DISTRICT=05;
 IF PARISH=34 THEN DISTRICT=05;
 IF PARISH=35 THEN DISTRICT=06;
 IF PARISH=36 THEN DISTRICT=02;
 IF PARISH=37 THEN DISTRICT=05;
 IF PARISH=38 THEN DISTRICT=02;
 IF PARISH=39 THEN DISTRICT=61;
 IF PARISH=40 THEN DISTRICT=08;
 IF PARISH=41 THEN DISTRICT=04;
 IF PARISH=42 THEN DISTRICT=05;
 IF PARISH=43 THEN DISTRICT=08;
 IF PARISH=44 THEN DISTRICT=02;
 IF PARISH=45 THEN DISTRICT=02;
 IF PARISH=46 THEN DISTRICT=62;
 IF PARISH=47 THEN DISTRICT=61;
 IF PARISH=48 THEN DISTRICT=02;
 IF PARISH=49 THEN DISTRICT=03;
 IF PARISH=50 THEN DISTRICT=03;
 IF PARISH=51 THEN DISTRICT=03;
 IF PARISH=52 THEN DISTRICT=62;
 IF PARISH=53 THEN DISTRICT=62;
 IF PARISH=54 THEN DISTRICT=58;
 IF PARISH=55 THEN DISTRICT=02;
 IF PARISH=56 THEN DISTRICT=05;
 IF PARISH=57 THEN DISTRICT=03;
 IF PARISH=58 THEN DISTRICT=08;
 IF PARISH=59 THEN DISTRICT=62;
```

```
   IF PARISH=60 THEN DISTRICT=04;
   IF PARISH=61 THEN DISTRICT=61;
   IF PARISH=62 THEN DISTRICT=05;
   IF PARISH=63 THEN DISTRICT=61;
   IF PARISH=64 THEN DISTRICT=08;

  PROC SORT;
  BY LTYPE HCLASS HNUM MP;


DATA MASACC3;
SET MASACC1;
BY LTYPE HCLASS HNUM MP;

  LTYPE='3';

  IF FIRST.MP      THEN TOT_ACC=0;
   TOT_ACC+1;

  IF LAST.MP      THEN DO;
   LENGTH=1;
  OUTPUT;
  END;

PROC SORT; BY CONTROL SECTION;

DATA TRI;                                                                    01440099
INFILE OUT;                                                                  01450099
INPUT CONTROL SECTION TRIANGLE MUEBAL MUEBPL VREBAL VREBPL PPWMNL     01460099
      APWMNL WTA WTP;                                                        01470099
                                                                             01480099
PROC SORT; BY CONTROL SECTION;                                               01490099
                                                                             01500099
                                                                             01510099
DATA RAT; MERGE MASACC3 TRI; BY CONTROL SECTION;                             01520099
IF MUEBAL=. THEN DELETE;                                                     01530099
    IF YEAR=. AND ADT=. THEN DELETE;                                         01540099
     IF PAVTYP=50                                                            01550099
     OR PAVTYP=60 THEN DO;                                                   01560099
     MU=MUEBAL; VR=VREBAL; WMN=APWMNL; PR=WTA;                               01570099
     END;                                                                    01580099
                                                                             01590099
     ELSE DO;                                                                01600099
     MU=MUEBPL; VR=VREBPL; WMN=PPWMNL; PR=WTP;                               01610099
     END;                                                                    01620099
                                                                             01630099
     DROP TRIANGLE MUEBAL MUEBPL VREBAL VREBPL WTA WTP APWMNL PPWMNL;        01640099
                                                                             01641099
     PROC SORT; BY LTYPE HCLASS HNUM MP;                                     01641199
                                                                             01641299
   DATA  RATE; SET RAT;                                                      01690099
   BY LTYPE HCLASS HNUM MP;                                                  01700099

                                                                             01430099
                                                                             01710099
   *CALCULATES THE ACCIDENTS PER MILLION VEHICLE MILES;                      01760099
                                                                             01820099
    *EXPOSURE IS DEPENDENT ON PROPORTION WET TIME AND LENGTH;                 01830099
    ADT=ADT*LENGTH*WMN;                                                      01840099
    MVM=365*ADT/1000000;                                                     01850099
    ACC_MVM=TOT_ACC/MVM;                                                     01860099
    ACC_MIL=TOT_ACC/LENGTH;                                                  01870099
    VOL=ADT*365;                                                             01880099
    SQRVOL=VOL**2;                                                           01890099
```

```
 TRU_RAT=TOT_ACC/(MU*LENGTH*ADT*365);                                    01900099
 NIBYVI=TOT_ACC/VOL;                                                     01910099
 NIBYSQVI=TOT_ACC/SQRVOL;                                                01920099
 ONEBYVOL=1/VOL;                                                         01930099
                                                                         01970099
LABEL  TOT_ACC=ACCIDENTS PER INTERSECTION                                01980099
       ACC_MIL=ACCIDENTS PER MILE                                        01990099
       ACC_MVM=ACCIDENTS PER MILLION VEHICLE MILES;                      02000099
                                                                         02010099
 OUTPUT;                                                                 02020099
                                                                         02040099
 PROC MEANS NOPRINT SUM MEAN STD VAR RANGE N;                            02050099
 VAR TOT_ACC ACC_MVM ADT NIBYVI NIBYSQVI ONEBYVOL TRU_RAT;               02060099
 BY LTYPE HCLASS;                                                        02070099
 *FORMAT HCLASS HCL.;                                                    02080099
 OUTPUT OUT=STATAV SUM=STATSUM MVMSUM ADTSUM NVSUM                       02090099
 MEAN=STATOT STAMVM STAADT STANV STASQNV HARVOL MNTR                     02100099
 STD=STASTD MVMSTD ADTSTD NVSTD NVSSTD ONESTD TRSTD                      02110099
 VAR=TOTVAR MVMVAR ADTVAR NVVAR NSVVAR                                   02120099
 RANGE=TOTRG MVMRG ADTRG NVRNG NSRNG                                     02121099
 N=TN MN AN NVN NSN;                                                     02122099
                                                                         02130099
 PROC SORT DATA=STATAV;                                                  02140099
 BY LTYPE HCLASS;                                                        02150099
                                                                         02170099
 DATA COMBINE;                                                           02180099
 MERGE RATE STATAV;                                                      02190099
 BY LTYPE HCLASS;                                                        02200099
                                                                         02210099
                                                                         02211099
 * FLAGGING BY C2;                                                       02340099
                                                                         02350099
 SIG_NBYV=STATSUM/(ADTSUM*365);                                          02360099
                                                                         02370099
  C2=SIG_NBYV+(1.645 *SQRT(SIG_NBYV/VOL)) + (1/(2*VOL));                 02380099
                                                                         02410099
 *FLAGGING BY BAYESIAN CRITERION 2;                                      02420099
                                                                         02430099
 BETA=STANV/NVVAR;                                                       02440099
 ALPHA=BETA*STANV;                                                       02450099
                                                                         02460099
 BETAI=BETA+VOL;                                                         02470099
 ALPHAI=ALPHA + TOT_ACC;                                                 02480099
                                                                         02490099
 ADJ_STA1= STANV*BETAI;                                                  02500099
 ADJ_STA2= SIG_NBYV*BETAI;                                               02510099
                                                                         02520099
                                                                         02590099
 B2= 1-PROBGAM(ADJ_STA2,ALPHAI);                                         02600099

KEEP DISTRICT ACC_MVM TOT_ACC CONTROL SECTION HNUM HTYPE LTYPE
     PARISH BCLM HCLASS ADT C2 B2;


DATA OUT2.SNSPT88;
SET COMBINE ;
//                                                                       05440099
```

4.2 MENU PROGRAM LISTING

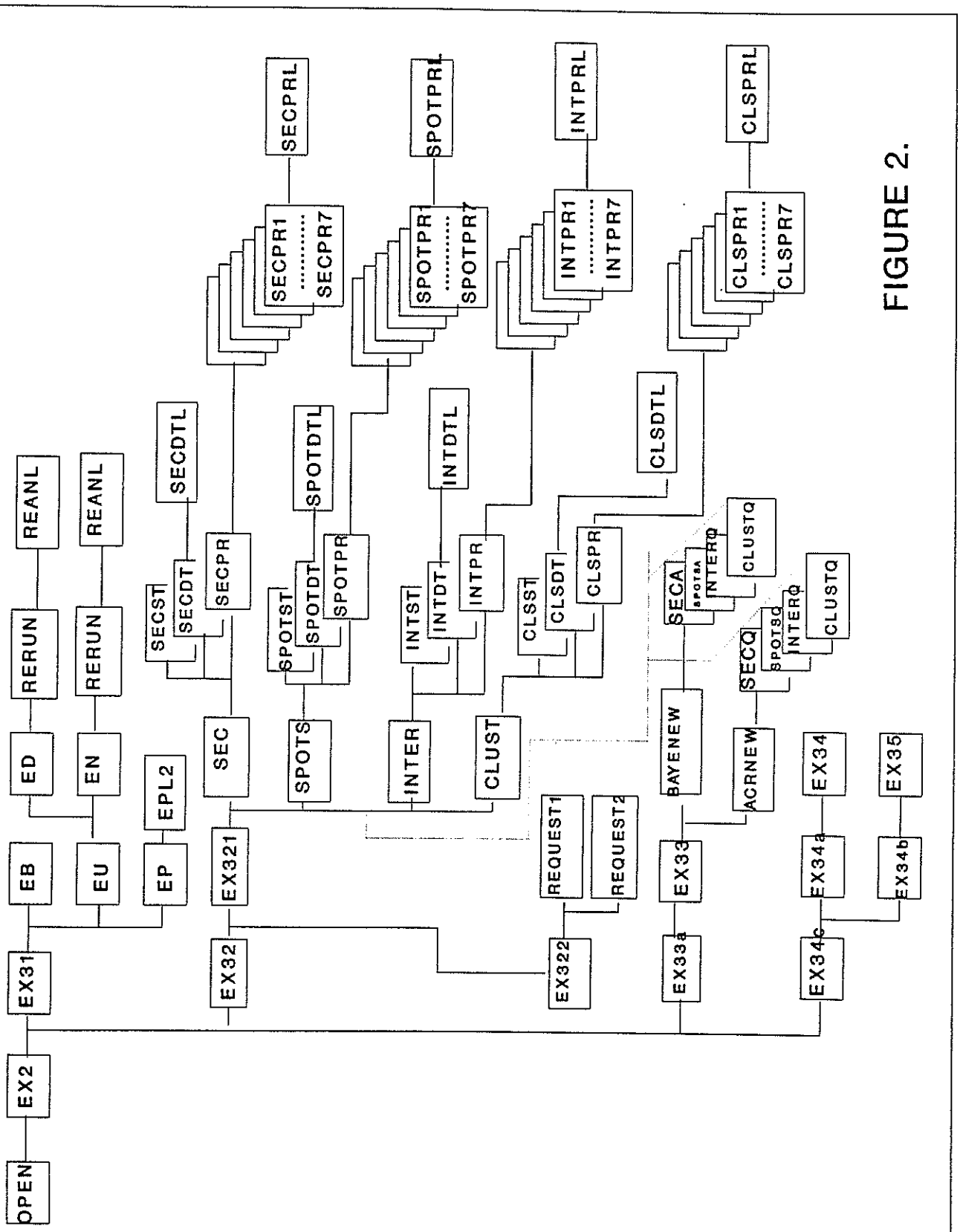

OPEN
EX2
EX31
EB
EU
EP
ED
EN
EPL2
REANL
REANL
RERUN
RERUN
EX32
EX321
SEC
SECST
SECDT
SECPR
SECDTL
SECPR1
SECPR7
SECPRL
SPOTS
SPOTST
SPOTDT
SPOTPR
SPOTDTL
SPOTPR1
SPOTPR7
SPOTPRL
INTER
INTST
INTDT
INTPR
INTDTL
INTPR1
INTPR7
INTPRL
CLUST
CLSST
CLSDT
CLSPR
CLSDTL
CLSPR1
CLSPR7
CLSPRL
EX322
REQUEST1
REQUEST2
EX33a
EX33
BAYENEW
SECA
INTERA
CLUSTA
ACRNEW
SECQ
INTERQ
CLUSTQ
EX34c
EX34a
EX34
EX34b
EX35

FIGURE 2.

SPOTPRL INTPRL CLSPRL

EX33 : Follows EX33a, for choice of analysis schemes.
ACRNEW : Follows EX33, for Bayesian Results.
SECA : Bayesian Sections report.
SPOTSA : Bayesian Spots report.
INTERA : Bayesian Intersections report.
CLUSTA : Bayesian Clusters report.
QLTYNEW : Follows EX33, for Quality Control Results.
SECQ : Quality Control, Sections report.
SPOTSQ : Quality Control, Spots report.
INTERQ : Quality Control, Intersections report.
CLUSTQ : Quality Control, Clusters report.

EX34a : Follows EX34c, for browsing the previous years tables.
EX34b : Follows EX34c, for deleting the previous years tables.
EX34 : Table selection.
EX35 : Table selection.

4. THE MENU DRIVEN DATABASE SYSTEM

4.1 THE MENU STRUCTURE

The complete menu structure of the entire database system has been given in figure 2. The menu generating programs have been named in the following order.

OPEN : This is the opening menu.

EX2 : This menu displays the utilities implemented.

EX31 : This menu is for maintenance utility.
EX32 : This menu is for the reports utility.
EX33a : This menu is for the analysis utility.
EX34c : This menu is for the archives utility.

EB : Follows EX31, for browsing tables.
EU : Follows EX31, for updating tables.
ED : Follows EU, for modifying tables.
EN : Follows EU, for inserting new records in the tables.
RERUN : Confirms the changes made.
REANL : Triggers off the analysis programs.
EP : Follows EX31, for printing tables (Onscreen/Hard Copy).
EPL3 : Confirmation for printing tables.
EPL2 : Destination options.

EX321 : Follows EX32, for DOTD reports.
EX322 : Follows EX32, for LTRC reports.
SEC : Follows EX321, for hazardous sections.
SPOTS : Follows EX321, for hazardous spots.
INTER : Follows EX321, for hazardous intersections.
CLUST : Follows EX321, for hazardous clusters.
SECST : Follows SEC, for a statewise report.
SECDT : Follows SEC, for a districtwise report.
SECDTL : Follows SECDT, report for a particular district.
SECPR : Follows SEC, for a parishwise report.
SECPR1-7: Follows SECPR, making a choice for a particular parish.
SECPRL : Follows SECPR1-7, displays the report.

On similar lines, menus have been developed for reports on spots, intersections and clusters. They are as follows:

Follows SPOTS:	Follows INTER:	Follows CLUST:
SPOTST	INTST	CLSST
SPOTDT	INTDT	CLSDT
SPOTDTL	INTDTL	CLSDTL
SPOTPR	INTPR	CLSPR
SPOTPR1-7	INTPR1-7	CLSPR1-7

NTRY: EX31.PROGRAM EX31.PROGRAM Last updated: 04/06/92

***** SOURCE *****

```
init:
return;
main:
choice=1;
loop:
do while (choice ne 0);
    choice=block('manipulation','manipulate dbase',25,
                 'browse','update',' ',' ',
                 ' ','print','help',' ',
                 'end',' ',' ',' ');
select(choice);
when(1) call display('eb.program');
when(2) call display('eu.program');
when(6) call display('epc.program');
when(7) call display('hex31.help');
when(9) leave loop;
otherwise
    do;
      if (choice<0) then call display('help.program', choice);
    end;
  end;
end;
call endblock();
return;
term:
return;
```

ENTRY: EU.PROGRAM EU.PROGRAM Last updated: 04/06/92

***** SOURCE *****

```
init:
return;
main:
choice=1;
loop:
do while (choice ne 0);
    choice=block('update','update tables',25,
                 'modify/delete','insert',' ',' ',
                 ' ',' ',' ',' ',' ',' ',' ',
                 'end',' ',' ',' ',' ');
select(choice);
when(1) call display('ed.program');
when(2) call display('en.program');
when(9) leave loop;
otherwise
    do;
      if (choice<0) then call display('help.program', choice);
    end;
  end;
end;
call endblock();
return;
term:
return;
```

ENTRY: EP.PROGRAM EP.PROGRAM Last updated: 04/06/92

***** SOURCE *****

```
  init:
  return;
  main:
  choice=1;
  loop:
  do while (choice ne 0);
  call wregion(1,1,19,80,'cmdline');
      choice=block('print','print tables',22,
                   'accident','driver','vehicle',' ',
                   'section',' ','skid',' ',
                   ' ',' ','end',' ');
  select(choice);
  when(1) do;
  call display('epl2.program');
  submit continue ;
   proc printto unit=20 new;
   proc print data=newlib.acc188;
   proc printto;run;
   run;
   filename ft20f001 clear;
  endsubmit;
  end;
  when(2) do;
  call display('epl2.program');
  submit continue ;
   proc printto unit=20 new;
   proc print data=newlib.drvr88;
   run;
   proc printto; run;
   filename ft20f001 clear;
  endsubmit;
  end;
  when(3) do;
  call display('epl2.program');
  submit continue ;
   proc printto unit=20 new;
   proc print data=newlib.vhcl88;
   run;
   proc printto;run;
   filename ft20f001 clear;
  endsubmit;
  end;
  when(5) do;
  call display('epl2.program');
  submit continue ;
   proc printto unit=20 new;
```

ENTRY: EP.PROGRAM EP.PROGRAM Last updated: 04/06/92

```
     proc print data=newlib.sectn88;
     run;
     proc printto;run;
     filename ft20f001 clear;
    endsubmit;
    end;
    when(7) do;
    call display('epl2.program');
    submit continue ;
     proc printto unit=20 new;
     proc print data=newlib.skid88;
     run;
     proc printto;run;
     filename ft20f001 clear;
    endsubmit;
    end;
    when(11) leave loop;
    otherwise
        do;
          if (choice<0) then call display('help.program', choice);
        end;
      end;
    end;
    call endblock();
    return;
    term:
    *call endlegend();
    *call poplegend();
    return;
```

ENTRY: EPL2.PROGRAM EPL2.PROGRAM Last updated: 04/06/92

***** SOURCE *****

```
 init:
 return;
 main:
 choice=1;
 loop:
 do while (choice ne 0);
 call wregion(1,1,19,80,'cmdline');
     choice=block('printers','select destination',22,
                  'ceba','ioroom',' ',' ',' ',
                  'stubbs','pageprint',' ',' ',
                  ' ',' ','end',' ');
 select(choice);
 when(1) do;
 submit continue ;
 filename ft20f001 sysout=a dest=ceba;
 endsubmit;
 end;
 when(2) do;
 submit continue ;
 filename ft20f001 sysout=a dest=ioroom;
 endsubmit;
 end;
 when(5) do;
 submit continue ;
 filename ft20f001 sysout=a dest=stubbs;
 endsubmit;
 end;
 when(6) do;
 submit continue ;
 filename ft20f001 sysout=a dest=pageprt;
 endsubmit;
 end;
 when(11) leave loop;
 otherwise
     do;
       if (choice<0) then call display('help.program', choice);
     end;
   end;
 end;
 call endblock();
 return;
 term:
 *call endlegend();
 *call poplegend();
 return;
```

ENTRY: ED.PROGRAM ED.PROGRAM Last updated: 05/12/92

```
     call display('rerun.program');
    end;
    when(5) do;
    call wregion(16,1,25,80,' ');
    call putlegend(1,'Modifying Section Table....','green','blinking');
    call legend('Section Information','','yellow','reverse');
    call putlegend(3,'Alt- PF3    END,   PF7    UP,  PF8    DOWN','cyan','none')
    call putlegend(4,'Alt- PF10   RIGHT, PF11   LEFT','cyan','none');
    call putlegend(5,'USE ARROW KEYS TO GET TO THE DESIRED','pink','none');
    call putlegend(6,'OBSERVATION, THEN PRESS <ENTER>','pink','none');
     call wregion(1,1,15,80,'cmdline');
     call fsview('newlib1.sectn88','edit','newlib.ltrc.sectn88.formula');
     call display('rerun.program');
    end;
    when(7) do;
    call wregion(16,1,25,80,' ');
    call putlegend(1,'Modifying Skid Table....','green','blinking');
    call legend('Skid Information','','yellow','reverse');
    call putlegend(3,'Alt- PF3    END,   PF7    UP,  PF8    DOWN','cyan','none')
    call putlegend(4,'Alt- PF10   RIGHT, PF11   LEFT','cyan','none');
    call putlegend(5,'USE ARROW KEYS TO GET TO THE DESIRED','pink','none');
    call putlegend(6,'OBSERVATION, THEN PRESS <ENTER>','pink','none');
     call wregion(1,1,15,80,'cmdline');
     call fsview('newlib1.skid','edit')  ;
     call display('rerun.program');
    end;
    when(11) leave loop;
    otherwise
        do;
          if (choice<0) then call display('help.program', choice);
        end;
      end;
    end;
    call endblock();
    return;
    term:
    call endlegend();
    *call poplegend();
    return;
```

ENTRY: EN.PROGRAM EN.PROGRAM Last updated: 04/06/92

***** SOURCE *****

```
init:
return;
main:
choice=1;
loop:
do while (choice ne 0);
call wregion(1,1,19,80,'cmdline');
    choice=block('insert','insert tables',18,
                 'accident','driver','vehicle',' ',
                 'section',' ','skid',' ',
                 ' ',' ','end',' ');
select(choice);
when(1) do;
call wregion(16,1,25,80,' ');
call putlegend(1,'Inserting into Accident Table....','green','blinking');
call legend('Accident 1','','yellow','reverse');
call putlegend(3,'  HOT KEYS !!!','red','none');
call putlegend(5,'Alt- PF3    END,   PF7    UP,  PF8    DOWN','cyan','none')
call putlegend(6,'Alt- PF10   RIGHT, PF11   LEFT','cyan','none');
call putlegend(8,'USE ARROW KEYS TO GET TO THE DESIRED','pink','none');
call putlegend(9,'OBSERVATION, THEN PRESS <ENTER>','pink','none');
 call wregion(1,1,15,80,'cmdline');
 call fsview('newlib.acc188','add');
 call display('rerun.program');
end;
when(2) do;
call wregion(16,1,25,80,' ');
call putlegend(1,'Inserting into Driver Table....','green','blinking');
call legend('Driver Information','','yellow','reverse');
call putlegend(3,'  HOT KEYS !!!','red','none');
call putlegend(5,'Alt- PF3    END,   PF7    UP,  PF8    DOWN','cyan','none')
call putlegend(6,'Alt- PF10   RIGHT, PF11   LEFT','cyan','none');
call putlegend(8,'USE ARROW KEYS TO GET TO THE DESIRED','pink','none');
call putlegend(9,'OBSERVATION, THEN PRESS <ENTER>','pink','none');
 call wregion(1,1,15,80,'cmdline');
 call fsview('newlib.drvr88','add');
 call display('rerun.program');
end;
when(3) do;
call wregion(16,1,25,80,' ');
call putlegend(1,'Inserting into Vehicle Table....','green','blinking');
call legend('Vehicle Information','','yellow','reverse');
call putlegend(3,'  HOT KEYS !!!','red','none');
call putlegend(5,'Alt- PF3    END,   PF7    UP,  PF8    DOWN','cyan','none')
call putlegend(6,'Alt- PF10   RIGHT, PF11   LEFT','cyan','none');
call putlegend(8,'USE ARROW KEYS TO GET TO THE DESIRED','pink','none');
```

ENTRY: EN.PROGRAM EN.PROGRAM Last updated: 04/06/92

```
    call putlegend(9,'OBSERVATION, THEN PRESS <ENTER>','pink','none');
     call wregion(1,1,15,80,'cmdline');
     call fsview('newlib.vhcl88','add');
     call display('rerun.program');
    end;
    when(5) do;
    call wregion(16,1,25,80,' ');
    call putlegend(1,'Inserting into Section Table....','green','blinking');
    call legend('Section Information','','yellow','reverse');
    call putlegend(3,'  HOT KEYS !!!','red','none');
    call putlegend(5,'Alt- PF3    END,   PF7    UP,  PF8    DOWN','cyan','none')
    call putlegend(6,'Alt- PF10   RIGHT, PF11   LEFT','cyan','none');
    call putlegend(8,'USE ARROW KEYS TO GET TO THE DESIRED','pink','none');
    call putlegend(9,'OBSERVATION, THEN PRESS <ENTER>','pink','none');
     call wregion(1,1,15,80,'cmdline');
     call fsview('newlib1.sectn88','add');
     call display('rerun.program');
    end;
    when(7) do;
    call wregion(16,1,25,80,' ');
    call putlegend(1,'Inserting into Skid Table....','green','blinking');
    call legend('Skid Information','','yellow','reverse');
    call putlegend(3,'  HOT KEYS !!!','red','none');
    call putlegend(5,'Alt- PF3    END,   PF7    UP,  PF8    DOWN','cyan','none')
    call putlegend(6,'Alt- PF10   RIGHT, PF11   LEFT','cyan','none');
    call putlegend(8,'USE ARROW KEYS TO GET TO THE DESIRED','pink','none');
    call putlegend(9,'OBSERVATION, THEN PRESS <ENTER>','pink','none');
     call wregion(1,1,15,80,'cmdline');
     call fsview('newlib1.skid','add');
     call display('rerun.program');
    end;
    when(11) leave loop;
    otherwise
        do;
          if (choice<0) then call display('help.program', choice);
        end;
      end;
    end;
    call endblock();
    return;
    term:
    call endlegend();
    *call poplegend();
    return;
```

ENTRY: RERUN.PROGRAM CONFIRM.PROGRAM Last updated: 04/06/92

***** SOURCE *****

```
  init:
  return;
  main:
  choice=9;
  loop:
  do while (choice ne 0);
      choice=block('rerun analysis program','analysis programs to be rerun?',
                   24,' ',' ',' ',' ',' ',' ',' ',' ',
                   'yes','no','end',' ');
  select(choice);
  when(9) call display('reanl.program');
  when(10) leave loop;
  when(11) leave loop;
  otherwise
      do;
        if (choice<0) then call display('help.program', choice);
  alarm;
      end;

    end;
  end;
  call endblock();
  return;
  term:
  return;
```

ENTRY: REANL.PROGRAM REANL.PROGRAM Last updated: 04/06/92

***** SOURCE *****

```
init:
return;
main:
*tso submit ltrc.final(pgm33) ;
*tso submit ltrc.final(pgm44) ;
*tso submit ltrc.final(pgm55) ;
return;
term:
return;
```

ENTRY: EX32.PROGRAM EX32.PROGRAM Last updated: 04/06/92

***** SOURCE *****

```
  init:
  return;
  main:
  choice=1;
  loop:
  do while (choice ne 0);
      choice=block('reports','reports section',9,
                   'dotd','ltrc',' ',' ',
                   ' ','help',' ',' ',
                   ' ',' ',' ',' ',
                    'end');
  select(choice);
  when(1) call display('ex321.program');
  when(2) call display('ex322.program');
  when(6) call display('hex32.help');
  when(12) leave loop;
  otherwise
      do;
        if (choice<0) then call display('help.program', choice);
      end;
    end;
  end;
  call endblock();
  return;
  term:
  return;
```

ENTRY: EX321.PROGRAM EX321.PROGRAM Last updated: 04/07/92

***** SOURCE *****

```
 init:
 return;
 main:
 choice=1;
 loop:
 do while (choice ne 0);
     choice=block('reports','dotd reports ',9,
                  'sections','spots',' ',' ',
                  'intersections','clusters',' ',' ',
                  'help',' ',' ',
                   'end');
 select(choice);
 when(1) do;
 submit continue sql;
 create view newlib.rl as select parish,district,hnum,htype,hclass,control,
          section,bclm,adt,ltype,b2  from newlib.snsi88
 where ltype='2'order by b2;
 endsubmit;
 call display('sec.program');
 end;
 when(2) do;
 submit continue sql;
 create view newlib.rl as select parish,district,hnum,htype,hclass,control,
          section,bclm,adt,ltype,b2  from newlib.snspt88
 where ltype='3'order by b2;
 endsubmit;
 call display('spots.program');
 end;
 when(5) do;
 submit continue sql;
 create view newlib.rl as select parish,district,hnum,htype,hclass,control,
          section,bclm,adt,ltype,b2  from newlib.snsi88
 where ltype='1'order by b2;
 endsubmit;
 call display('inter.program');
 end;
 when(6) do;
 submit continue sql;
 create view newlib.rl as select parish,district,hnum,htype,hclass,control,
          section,bclm,adt,ltype,b2  from newlib.sncls88
 where ltype='4'order by b2;
 endsubmit;
 call display('clust.program');
 end;
 when (9) call display('hex321.help');
 when(12) leave loop;
```

ENTRY: EX321.PROGRAM EX321.PROGRAM Last updated: 04/07/92

```
   otherwise
       do;
         if (choice<0) then call display('help.program', choice);
       end;
     end;
   end;
   call endblock();
   return;
   term:
   return;
```

ENTRY: SEC.PROGRAM SEC1.PROGRAM Last updated: 04/06/92

***** SOURCE *****

```
init:
return;
main:
choice=1;
loop:
do while (choice ne 0);
    choice=block('recommended method','sections',23,
                 'statewise','districtwise',' ',' ',
                 'parishwise','help',' ',' ',
                 'end',' ',' ',' ');
select(choice);
when(1) call display('secst.program');
when(2) call display('secdt.program');
when(5) call display('secpr.program');
when(6) call display('hloc.help');
when(9) leave loop;
otherwise
    do;
      if (choice<0) then call display('help.program', choice);
    end;
  end;
end;
call endblock();
return;
term:
return;
```

ENTRY: SECDT.PROGRAM SECDT.PROGRAM Last updated: 04/06/92

***** SOURCE *****

```
init:
return;
main:
choice=1;
loop:
do while (choice ne 0);
    choice=block('sections','districtwise analysis',12,
                 'district # 02',
                 'district # 03',
                 'district # 04',
                 'district # 05',
                 'district # 07',
                 'district # 08',
                 'district # 58',
                 'district # 61',
                 'district # 62',
                  '  ',' ','end');
select(choice);
when(1) do;
submit continue sql;
create view newlib.r2 as select * from newlib.r1
where district = 02;
endsubmit;
call display('secdt1.program');
end;
when(2) do;
submit continue sql;
create view newlib.r2 as select * from newlib.r1
where district = 03;
endsubmit;
call display('secdt1.program');
end;
when(3) do;
submit continue sql;
create view newlib.r2 as select * from newlib.r1
where district = 04;
endsubmit;
call display('secdt1.program');
end;
when(4) do;
submit continue sql;
create view newlib.r2 as select * from newlib.r1
where district = 05;
endsubmit;
call display('secdt1.program');
end;
```

ENTRY: SECDT.PROGRAM SECDT.PROGRAM Last updated: 04/06/92

```
    when(5) do;
    submit continue sql;
    create view newlib.r2 as select * from newlib.r1
    where district = 07;
    endsubmit;
    call display('secdt1.program');
    end;
    when(6) do;
    submit continue sql;
    create view newlib.r2 as select * from newlib.r1
    where district = 08;
    endsubmit;
    call display('secdt1.program');
    end;
    when(7) do;
    submit continue sql;
    create view newlib.r2 as select * from newlib.r1
    where district = 58;
    endsubmit;
    call display('secdt1.program');
    end;
    when(8) do;
    submit continue sql;
    create view newlib.r2 as select * from newlib.r1
    where district = 61;
    endsubmit;
    call display('secdt1.program');
    end;
    when(9) do;
    submit continue sql;
    create view newlib.r2 as select * from newlib.r1
    where district = 62;
    endsubmit;
    call display('secdt1.program');
    end;
    when(12) leave loop;
    otherwise
        do;
          if (choice<0) then call display('help.program', choice);
        end;
      end;
    end;
    call endblock();
    return;
    term:
    return;
```

ENTRY: SECDTL.PROGRAM SECDTL.PROGRAM Last updated: 04/20/92

***** SOURCE *****

```
init:
return;
main:
choice=1;
loop:
do while (choice ne 0);
    choice=block('sections','districtwise analysis',9,
                 '2 ln rural',
                 '4 ln rural',
                 '4 ln dvd rural',
                 'freeway rural',
                 '2 ln urban',
                 '4 ln urban',
                 '4 ln dvd urban',
                 'freeway urban',
                  ' ',' ',' ','end');
select(choice);
when(1) do;
submit continue sql;
create view newlib.r3 as select * from newlib.r2
where hclass = 1;
endsubmit;
call wregion(18,1,25,80,' ');
call putlegend(1,'Two Lane Rural','yellow','blinking');
call legend('Districtwise Sections Report',' ','green','reverse');
call putlegend(3,'Alt- PF3    END,   PF7    UP,  PF8    DOWN','cyan','none')
call putlegend(4,'Alt- PF10   RIGHT, PF11   LEFT','cyan','none');
 call wregion(1,1,17,80,'cmdline');
call fsview('newlib.r3');
call display('fpc1.program');
end;
when(2) do;
submit continue sql;
create view newlib.r3 as select * from newlib.r2
where hclass = 2;
endsubmit;
call wregion(18,1,25,80,' ');
call putlegend(1,'Four Lane Rural','yellow','blinking');
call legend('Districtwise Sections Report',' ','green','reverse');
call putlegend(3,'Alt- PF3    END,   PF7    UP,  PF8    DOWN','cyan','none')
call putlegend(4,'Alt- PF10   RIGHT, PF11   LEFT','cyan','none');
 call wregion(1,1,17,80,'cmdline');
call fsview('newlib.r3');
call display('fpc1.program');
end;
when(3) do;
```