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.
The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Louisiana Transportation Research Center, the Louisiana Department of Transportation and Development, or the Federal Highway Administration. This report does not constitute a standard, specification or regulation.
WET WEATHER HIGHWAY ACCIDENT

ANALYSIS AND SKID RESISTANCE

DATABASE MANAGEMENT SYSTEM

REFERENCE MANUAL
# Table of Contents

CHAPTER 1. Introduction.........................................................1

CHAPTER 2. The Wet Weather Highway Accident Analysis  
             Skid Resistance Database Management System...........3

CHAPTER 3. Analysis Programs................................................7  
            3.1 Instructions for running the Wet Weather Highway  
            Accident Analysis Programs............................7  
            3.2 Analysis Programs' Listing...........................13

CHAPTER 4. The Menu Driven Database Management...............57  
            4.1 Menu Structure........................................57  
            4.2 Menu Programs' Listing...............................61
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
   DRVRR88
   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.

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.

<table>
<thead>
<tr>
<th>PROMPT</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROJECT</td>
<td>IEKLEE</td>
</tr>
<tr>
<td>GROUP</td>
<td>LTRC</td>
</tr>
<tr>
<td>TYPE</td>
<td>FINAL</td>
</tr>
<tr>
<td>MEMBER</td>
<td>PGM1 (to run program 1)</td>
</tr>
</tbody>
</table>

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 pgml 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.
3.2 ANALYSIS PROGRAM LISTING
* THIS PROGRAM TAKES AS INPUT THE HOURLY RAINFALL DATA FROM
  THE FOLLOWING PRECIPITATION FILES;

* INSTRUCTIONS TO RUN 1989 AND 1990 DATA FOR NEW ORLEANS;

* COLLECT 1989 AND 1990 PRECIPITATION DATA AND
  NAME IT AS P8089.NOR FOR NEWORLEANS;

* MAKE SURE THAT THE FORMAT IS THE SAME AS IN PREVIOUS YEARS
  COMARE WITH FILE P6660.HUF;

* CHANGE JCL STATEMENT HAVING FILEREF UNI TO DD DSN=IEKLEE.P8089.HUF;

* GET HOURLY SURFACE OBSERVATION DATA FOR 1989 AND 1990 AND NAME IT
  AS HSONO.HLY;

* MAKE SURE FORMAT IS SAME AS IN HOURLY.DATA(NEWORLNS);

* CHANGE JCL STATEMENT HAVING FILEREF HLY TO DD DSN=IEKLEE.HOURLY.DATA
  (NEWORLNS);

* CHANGE YEAR NUMBER TO CURRENT YEAR TO BE RUN IN LINES
  WHERE I HAVE COMMENTED CHANGE YEARS;

* RUN THE PROGRAM ONE YEAR AT A TIME (MEMORY PROBLEMS);

* FOR RERUNNING PREVIOUS YEARS PRIOR TO 1989 DO THE FOLLOWING;
  *INFILE UNI P6660.HUF FOR NEW ORLEANS STATION;
  *INFILE UNI P0549.HUF FOR BATON ROUGE STATION;
  *INFILE UNI P5078.HUF FOR LAKE CHARLES STATION;
  *INFILE UNI P8440.HUF FOR SHREVEPORT STATION;

* THIS ALSO TAKES HOURLY SURFACE OBSERVATIONS DATA FROM
  THE FOLLOWING HOURLY FILES TO BE MAINTAINED IN DISKS;

*INFILE HLY HOURLY.DATA(NEWORLNS) FOR NEW ORLEANS;
*INFILE HLY HOURLY.DATA(LAKECHAS) FOR LAKE CHARLES;
*INFILE HLY BR8088.HLY OR BR4882.HLY FOR BATON ROUGE FOR THE RESPECTIVE
  YEARS;
*INFILE HLY SHRVTPT.HLY FOR SHREVEPORT;

*OUTPUT FILE WETHOURS.LAKECHAS FOR LAKE CHARLES;
*OUTPUT FILE WETHOURS.NEWORLNS FOR NEW ORLEANS;
*OUTPUT FILE WETHOURS.BTR FOR BATON ROUGE;
*OUTPUT FILE WETHOURS.SHR FOR SHREVEPORT;

******************************************************************************
* THIS PROGRAM CALCULATES THE TOTAL WET HOURS OF RAINFALL BASED ON THE WETTIME MODEL FOR A STATION WITH UNIVERSAL RAIN GAUGEanych;
DATA ONE;
  INFILE UNI;
  INPUT (DATE AR1 AR2-AR24) ($9. $5. 23*5.$.);
YEAR=SUBSTR(DATE,1,4);
MONTH=SUBSTR(DATE,5,2);
DAY=SUBSTR(DATE,7,2);
DATE=MDY(MONTH,DAY,YEAR);
DATE=COMPRESS(DATE);

*CHANGE YEAR HERE;
IF YEAR='1959';

ARRAY AOUR[24] $ AR1-AR24;
ARRAY HUR[24] HR1-HR24;
ARRAY MISSING[24] $ MISS1-MISS24;
* EXTRACT MISSING INFORMATION;
DO J=1 TO 24;
A=AOUR[J];
B='A'; C='D'; D='M';
IF INDEX(A,B) = 0 OR INDEX(A,C) = 0 OR INDEX(A,D) = 0 THEN DO;
MISSING[J]=1; AOUR[J]=''0';
END;
ELSE MISSING[J]=0;
STR=' '; AOUR[J]=STR || AOUR[J]; AOUR[J]=TRIM(AOUR[J]);
HUR[J]=(AOUR[J]+0)/100;
END;

DROP AR1-AR24 B C D A STR;

PROC SORT; BY DATE;

DATA I;
  INFILE HLY;
  INPUT YR 6-7 MONTH $ 8-9 DAY $ 10-11 HUR 12-13 FROZEN1 27 FROZEN2 28
  FROZEN3 29 FOG 30 DEWPT 36-38 WIND 41-42 TEMP 47-49 RH 53-55 CC $ 56;

* CHANGE YEAR HERE;
* EXAMPLE (FOR MULTIPLE YEARS) IF YR>=85 AND YR<=89;
  IF YR=59;
    HUR=HUR+1;
    YR1='1900+YR;
    A='';
  YEAR= A || YR1;
  YEAR=TRIM(YEAR);
  DATE=MDY(MONTH,DAY,YEAR);
  DATE=COMPRESS(DATE);
* CONVERSION OF KNOT WIND SPEEDS TO MILES PER HOUR;
  WIND = WIND*1.15;

* EDITING OF FROZEN PRECIPITATION DATA;
FRZN=SUM(FROZEN1,FROZEN2);
FRZN=FRZN+SUM(FROZEN2,FROZEN3);

* FOR MISSING VALUES;
  IF RH=. THEN RH=60;
  IF WIND=. THEN WIND=2;
  IF TEMP=. THEN TEMP=70;
  IF DEWPT=. THEN DEWPT=50;

* TRANSLATION OF CLOUD COVER DATA;
IF CC = 'A' OR CC = '1' THEN CL = 0.1;
IF CC = 'B' OR CC = '2' THEN CL = 0.2;
IF CC = 'C' OR CC = '3' THEN CL = 0.3;
IF CC = 'D' OR CC = '4' THEN CL = 0.4;
IF CC = 'E' OR CC = '5' THEN CL = 0.5;
IF CC = 'F' OR CC = '6' THEN CL = 0.6;
IF CC = 'G' OR CC = '7' THEN CL = 0.7;
IF CC = 'H' OR CC = '8' THEN CL = 0.8;
IF CC = 'I' OR CC = '9' THEN CL = 0.9;
IF CC = 'J' OR CC = '0' THEN CL = 0.0;
IF CC = 'X' THEN CL = 1.0;
IF CC = '-' OR CC = '' THEN CL = 0.5;

*CONVERSION OF CLOUD COVER TO SOLAR RADIATION CORRECTED FOR CLOUDS RCL; 00780099
*SOLAR RADIATION AT LATITUDE 30 DEGREES NORTH;
* FROM NET RADIATION RECEIVED FROM A HORIZONTAL SURFACE BY DE JONG;
00790099
00800099
00810099

IF MONTH = 1 THEN RO = 520 / (24 * 60);
IF MONTH = 2 THEN RO = 630 / (24 * 60);
IF MONTH = 3 THEN RO = 775 / (24 * 60);
IF MONTH = 4 THEN RO = 895 / (24 * 60);
IF MONTH = 5 THEN RO = 975 / (24 * 60);
IF MONTH = 6 THEN RO = 1000 / (24 * 60);
IF MONTH = 7 THEN RO = 990 / (24 * 60);
IF MONTH = 8 THEN RO = 925 / (24 * 60);
IF MONTH = 9 THEN RO = 820 / (24 * 60);
IF MONTH = 10 THEN RO = 685 / (24 * 60);
IF MONTH = 11 THEN RO = 560 / (24 * 60);
IF MONTH = 12 THEN RO = 490 / (24 * 60);

*SOLAR RADIATION CORRECTED FOR CLOUD COVER;
* FROM V P SINGH;
00930099
RCL = RO * (1 - 0.65 * CL);
00940099

*SETTING SOLAR RADIATION AT ZERO LEVELS FOR NIGHT TIME;
* FROM SUNSET TABLES FOR LOUISIANA CITIES;
00950099
00960099
00970099
00980099
00990099
10000099
10010099

IF MONTH = 1 AND (HUR < 9 OR HUR > 15) THEN RCL = 0;
IF MONTH = 2 AND (HUR < 9 OR HUR > 15) THEN RCL = 0;
IF MONTH = 3 AND (HUR < 8 OR HUR > 16) THEN RCL = 0;
IF MONTH = 4 AND (HUR < 8 OR HUR > 16) THEN RCL = 0;
IF MONTH = 5 AND (HUR < 7 OR HUR > 17) THEN RCL = 0;
IF MONTH = 6 AND (HUR < 7 OR HUR > 17) THEN RCL = 0;
IF MONTH = 7 AND (HUR < 7 OR HUR > 17) THEN RCL = 0;
IF MONTH = 8 AND (HUR < 8 OR HUR > 16) THEN RCL = 0;
IF MONTH = 9 AND (HUR < 8 OR HUR > 16) THEN RCL = 0;
IF MONTH = 10 AND (HUR < 8 OR HUR > 16) THEN RCL = 0;
IF MONTH = 11 AND (HUR < 8 OR HUR > 15) THEN RCL = 0;
IF MONTH = 12 AND (HUR < 9 OR HUR > 15) THEN RCL = 0;

DIFF = ABS(TEMP - DEWPT);
10100099

DROP CC CL RO FROZEN1 - FROZEN3;
01140099

PROC SORT; BY YEAR MONTH DAY HUR;
01150099
01160099
01170099
01180099
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 FOOG DEWPT WIND
  TEMP RH RCL DIFF @@ ;

DATA III;
INFILE TEMP;
INPUT DATE $ FRZN1 FOOG1 DEWPT1 WIND1 TEMP1 RH1 RCL1 DIFF1
  FRZN2 FOOG2 DEWPT2 WIND2 TEMP2 RH2 RCL2 DIFF2
  FRZN3 FOOG3 DEWPT3 WIND3 TEMP3 RH3 RCL3 DIFF3
  FRZN4 FOOG4 DEWPT4 WIND4 TEMP4 RH4 RCL4 DIFF4
  FRZN5 FOOG5 DEWPT5 WIND5 TEMP5 RH5 RCL5 DIFF5
  FRZN6 FOOG6 DEWPT6 WIND6 TEMP6 RH6 RCL6 DIFF6
  FRZN7 FOOG7 DEWPT7 WIND7 TEMP7 RH7 RCL7 DIFF7
  FRZN8 FOOG8 DEWPT8 WIND8 TEMP8 RH8 RCL8 DIFF8
  FRZN9 FOOG9 DEWPT9 WIND9 TEMP9 RH9 RCL9 DIFF9
  FRZN10 FOOG10 DEWPT10 WIND10 TEMP10 RH10 RCL10 DIFF10
  FRZN11 FOOG11 DEWPT11 WIND11 TEMP11 RH11 RCL11 DIFF11
  FRZN12 FOOG12 DEWPT12 WIND12 TEMP12 RH12 RCL12 DIFF12
  FRZN13 FOOG13 DEWPT13 WIND13 TEMP13 RH13 RCL13 DIFF13
  FRZN14 FOOG14 DEWPT14 WIND14 TEMP14 RH14 RCL14 DIFF14
  FRZN15 FOOG15 DEWPT15 WIND15 TEMP15 RH15 RCL15 DIFF15
  FRZN16 FOOG16 DEWPT16 WIND16 TEMP16 RH16 RCL16 DIFF16
  FRZN17 FOOG17 DEWPT17 WIND17 TEMP17 RH17 RCL17 DIFF17
  FRZN18 FOOG18 DEWPT18 WIND18 TEMP18 RH18 RCL18 DIFF18
  FRZN19 FOOG19 DEWPT19 WIND19 TEMP19 RH19 RCL19 DIFF19
  FRZN20 FOOG20 DEWPT20 WIND20 TEMP20 RH20 RCL20 DIFF20
  FRZN21 FOOG21 DEWPT21 WIND21 TEMP21 RH21 RCL21 DIFF21
  FRZN22 FOOG22 DEWPT22 WIND22 TEMP22 RH22 RCL22 DIFF22
  FRZN23 FOOG23 DEWPT23 WIND23 TEMP23 RH23 RCL23 DIFF23
  FRZN24 FOOG24 DEWPT24 WIND24 TEMP24 RH24 RCL24 DIFF24

PROC SORT; BY DATE;

DATA ONE1;
  MERGE ONE III; BY DATE;

DATA TWO;
  * GET FIVE HRS OF PRECI OF PREVIOUS DAY AND PUT IT IN THE SAME LINE;
  SET ONE1; BY DATE;
  DATE=DATE+1;
  ARRAY HR[10] HR20-HR24 HR25-HR29;
  ARRAY FG[10] FOG20-FOG24 FOG25-FOG29;
  ARRAY DW[10] DEWPT20-DEWPT24 DEWPT25-DEWPT29;
  ARRAY WD[10] WIND20-WIND24 WIND25-WIND29;
  ARRAY DIFF[10] DIFF20-DIFF24 DIFF25-DIFF29;

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;

DATE=COMPRESS(DATE);
```plaintext
PROC SORT; BY DATE;

DATA THREE;
MERGE TWO ONE1;
BY DATE;

DATA TWOA;
* GET FIVE HOURS 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 1= 6 TO 29;
* WET HOUR DUE TO FOG;
IF HUR[I]=0 THEN DO;
  IF FOG[I]=1 OR FOG[I]=2 OR FOG[I]=3 THEN DO;
    IF DIFF[I] <=2 AND WIND[I] < 3 THEN DO;
      HUR[I]=0.06;
    END;
  END;
END;
* WET HOUR DUE TO PRECIPITATION;
IF HUR[I]=0 THEN DO;
* DURATION OF RAINFALL DEPENDING ON INTENSITY;
  IF HUR[I] >=0.01 AND HUR[I]<0.02 THEN WETHR[I-5]=15;
  IF HUR[I] >=0.02 AND HUR[I]<0.03 THEN WETHR[I-5]=30;
  IF HUR[I] >=0.03 AND HUR[I]<0.05 THEN WETHR[I-5]=45;
  IF HUR[I] >=0.05 THEN WETHR[I-5]=60;
* RUNOFF TIME ASSUMED TO BE 5 MINUTES;
  WETHR[I-5]=WETHR[I-5]+5;
* DRYING TIME; *DEVIATIONS FROM MEAN DRYING TIME OF 31.6;
* TEMPERATURE RECORDS;
  IF TEMP[I]<67.5 THEN WETHR[I-5]=WETHR[I-5]+3.7;
  IF TEMP[I]>67.5 AND TEMP[I]<82.5 THEN WETHR[I-5]=WETHR[I-5]-0.7;
  ELSE WETHR[I-5]=WETHR[I-5]-3.0;
* RELATIVE HUMIDITY RECORDS;
  IF RH[I]<50 THEN WETHR[I-5]=WETHR[I-5]-4.5;
  ELSE WETHR[I-5]=WETHR[I-5]+6.1;
* SOLAR RADIATION RECORDS;
  IF RCL[I]<0.4 THEN WETHR[I-5]=WETHR[I-5]+11.6;
  IF RCL[I]>0.4 AND RCL[I]<0.85 THEN WETHR[I-5]=WETHR[I-5]+5.6;
  ELSE WETHR[I-5]=WETHR[I-5]-17.2;
* WIND SPEED RECORDS;
  IF WIND[I]>1.5 AND RCL[I]<0.85 THEN WETHR[I-5]=WETHR[I-5]-11.6;
* PAVEMENT MATERIAL;
  WETHR[I-5]=WETHR[I-5]+3.9;
END;
IF I=6 THEN DO;
* WET HOUR DUE TO FOG;
IF HUR[I-1]=0 THEN DO;
  IF FOG[I-1]=1 OR FOG[I-1]=2 OR FOG[I-1]=3 THEN DO;
    IF DIFF[I-1] <=2 AND WIND[I-1] < 3 THEN DO;
      HUR[I-1]=0.06;
    END;
  END;
END;
IF HUR[I-1]=0 THEN DO;
  IF HUR[I-1]>=0.01 AND HUR[I-1]<0.02 THEN WETHR[25]=15;
  IF HUR[I-1]>=0.02 AND HUR[I-1]<0.03 THEN WETHR[25]=30;
  IF HUR[I-1]>=0.03 AND HUR[I-1]<0.05 THEN WETHR[25]=45;
  IF HUR[I-1]>=0.05 THEN WETHR[25]=60;
* RUNOFF TIME ASSUMED TO BE 5 MINUTES;
* DRYING TIME; *DEVIATIONS FROM MEAN DRYING TIME OF 31.6;
* TEMPERATURE RECORDS;
  IF TEMP[I-1]>67.5 AND TEMP[I-1]<82.5 THEN WETHR[25]=WETHR[25]-0.7;
* RELATIVE HUMIDITY RECORDS;
*SOLAR RADIATION RECORDS;
03210099
03220099
03230099
IF RCL[I-1]>0.4 AND RCL[I-1]<0.85 THEN WETHR[25]=WETHR[25]+5.6;
03240099
03250099
*WIND SPEED RECORDS;
03260099
03270099
03280099
*PAVEMENT MATERIAL;
03290099
03300099
END;
03310099
END;
03320099
IF I=29 THEN DO;
03330099
* WET HOUR DUE TO FOG;
03340099
IF HUR[I+1]=0 THEN DO;
03350099
IF FOG[I+1]=1 OR FOG[I+1]=2 OR FOG[I+1]=3 THEN DO;
03360099
IF DIFF[I+1] <=2 AND WIND[I+1]< 3 THEN DO;
03370099
HUR[I+1]=0.06;
03380099
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]>=0.05 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
* TEMPERATURE RECORDS;
03530099
1) TEMP[I+1]<=67.5 THEN WETHR[27]=WETHR[27]+3.7;
03540099
2) 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
1) RCL[I+1]<=0.4 THEN WETHR[27]=WETHR[27]+11.6;
03620099
2) 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;
03680099
WETHR[27]=WETHR[27]+3.9;
03690099
END;
03700099
END;
03710099
*FOR AN HOUR WHOSE PREVIOUS HOUR HAD BUT NEXT DID NOT HAVE RAINFALL;
03720099
IF HUR[I-1]>0 AND HUR[I]<0 AND HUR[I+1]=0 THEN DO;
03730099
IF I=6 THEN DO;
03740099
IF WETHR[I-6]>=60 AND WETHR[I-5]>=60 THEN WETHR[I-6]=60;
03750099
IF WETHR[I-6]<=60 AND WETHR[I-5]<60 THEN DO;
03760099
WETHR[I-5]=WETHR[I-5]+WETHR[I-6]-60;
03770099
WETHR[I-6]=60;
03780099
IF WETHR[I-5] >=60 THEN WETHR[I-5]=60;
03790099
END;
03800099
IF I=6 THEN DO;
IF WETHR[25] >= 60 AND WETHR[1-5] < 60 THEN DO;
WETHR[25] = 60;
END;
IF WETHR[1-5] >= 60 THEN WETHR[1-5] = 60;
END;
END;
END;

DATA FIVE;
SET FOUR; BY DATE;
DATE = DATE - 1;
DATE = COMPRESS (DATE);
WHR26 = WHR25;
KEEP DATE WHR26;

PROC SORT; BY DATE;
DATA SIX;
MERGE FIVE FOUR;
BY DATE;

DATA SEVEN;
SET SIX;
ARRAY HUR[34] HR25-HR29 HR1-HR24 HR30-HR34;
ARRAY WETHR[28] WHR1-WHR28;
IF WHR26 = . AND WHR26 = 0 THEN WHR24 = WHR26;
DO I = 6 TO 29;
* FOR AN HOUR WHOSE NEXT HOUR HAD RAINFALL BUT PREVIOUS DID NOT;
IF HUR[I] = 0 AND HUR[I-1] = 0 AND HUR[I+1] = 0 THEN DO;
IF WETHR[I-5] >= 60 AND I = 29 THEN DO;
IF WETHR[I-4] >= 60 THEN WETHR[I-5] = 60;
IF WETHR[I-4] < 60 THEN DO;
IF WETHR[I-4] >= 60 THEN WETHR[I-4] = 60;
WETHR[I-5] = 60;
END;
END;
END;
IF WETHR[I-5] >= 60 AND I = 29 THEN DO;
IF WETHR[27] >= 60 THEN WETHR[I-5] = 60;
ELSE DO;
WETHR[27] = WETHR[27] + WETHR[I-5] - 60;
WETHR[I-5] = 60;
IF WETHR[27] >= 60 THEN WETHR[27] = 60;
END;
END;
END;

DATA FIVEA;
SET SEVEN; BY DATE;
DATE = DATE + 1;
DATE = COMPRESS (DATE);
WHR28 = WHR27;
KEEP DATE WHR28;

PROC SORT; BY DATE;

DATA SIXA;

03820099
03830099
03840099
03850099
03860099
03870099
03880099
03890099
03900099
03910099
03920099
03930099
03940099
03950099
03960099
03970099
03980099
03990099
04000099
04010099
04020099
04030099
04040099
04050099
04060099
04070099
04080099
04090099
04100099
04110099
04120099
04130099
04140099
04150099
04160099
04170099
04180099
04190099
04200099
04210099
04220099
04230099
04240099
04250099
04260099
04270099
04280099
04290099
04300099
04310099
04320099
04330099
04340099
04350099
04360099
04370099
04380099
04390099
04400099
04410099
04420099
04430099
04440099
BY DATE;

DATA EIGHTA;
SET Sixa;
ARRAY HUR[34] HR25-HR29 HR1-HR24 HR30-HR34;
ARRAY WETHR[28] WHR1-WHR28;
IF WHR28=0 AND WHR28=. THEN WHR1=WHR28;

DO I = 6 TO 29;

*FOR AN HUR WITH BOTH PREVIOUS AND NEXT RAINFALL HURS;
IF HUR[I]=0 AND HUR[I-1]=0 AND HUR[I+1]=0 THEN DO;
   IF I=6 THEN DO;
   IF WETHR[I-6]=60 AND WETHR[I-5]<60 THEN DO;
   WETHR[I-6]=60; WETHR[I-5]=WETHR[I-5]+WETHR[I-6]-60;
   IF WETHR[I-5]=60 THEN WETHR[I-5]=60;
END;
   IF I=6 THEN DO;
   IF WETHR[I-25]=60 AND WETHR[I-5]=60 THEN DO;
   IF WETHR[I-5]=60 THEN WETHR[I-5]=60;
END;
END;
END;

DATA NINE;
SET EIGHTA; BY DATE;
DATE=DATE+1;
DATE=COMPRESS(DATE);
WHR26=WHR25;
KEEP DATE WHR26;
PROC SORT; BY DATE;

DATA TEN;
MERGE NINE EIGHTA;
BY DATE;
IF WHR26=0 AND WHR26=. THEN WHR24=WHR26;

DATA ELEVEN;
SET TEN; BY DATE;
DATE=DATE+1;
DATE=COMPRESS(DATE);
WHR28=WHR27;
KEEP DATE WHR28;
PROC SORT; BY DATE;

DATA TWELVE;
MERGE ELEVEN TEN;
BY DATE;
IF WHR28=0 AND WHR28=. THEN WHR1=WHR28;
ARRAY WETHR[24] WHR1-WHR24;
ARRAY RCL[24] RCL1-RCL24;
ARRAY DIFF[24] DIFF1-DIFF24;
ARRAY TEMP[24] TEMP1-TEMP24;
ARRAY WIND(24) WIND1-WIND24;
ARRAY RH(24) RH1-RH24;

DO I=1 TO 24;
  IF WETHR(I)=. THEN WETHR(I)=0;

*DAY TIME SATURATED CONDITIONS AFFECTING DRYING;
  IF WETHR(I)=0 AND WETHR(I)=60 AND
     RCL(I)>0 AND DIFF(I)<2 THEN DO;
     M=I+2;
     DO K=I TO M UNTIL (DIFF(K)>2 OR K=24);
     IF (DIFF(K)<2 AND K=M) THEN WETHR(K)=60;
     IF (DIFF(K)>2 OR K=M) AND WETHR(K)=0 THEN DO;
     * DRYING TIME; *DEVIATIONS FROM MEAN DRYING TIME OF 31.6;
     WETHR(K)=WETHR(K)*31.6;
     * TEMPERATURE RECORDS;
     IF TEMP(K)<67.5 THEN WETHR(K)=WETHR(K)+3.7;
     IF TEMP(K)>67.5 AND TEMP(K)<82.5 THEN WETHR(K)=WETHR(K)-0.7;
     ELSE WETHR(K)=WETHR(K)-3.0;
     * RELATIVE HUMIDITY RECORDS;
     IF RH(K)<50 THEN WETHR(K)=WETHR(K)-4.5;
     IF RH(K)>50 AND RH(K)<82.5 THEN WETHR(K)=WETHR(K)-1.6;
     ELSE WETHR(K)=WETHR(K)+6.1;
     *SOLAR RADIATION RECORDS;
     IF RCL(K)<0.4 THEN WETHR(K)=WETHR(K)+11.6;
     IF RCL(K)>0.4 AND RCL(K)<0.85 THEN WETHR(K)=WETHR(K)+5.6;
     ELSE WETHR(K)=WETHR(K)-17.2;
  END;
END;

* NIGHT TIME SATURATED DRYING CONDITIONS;
  IF WETHR(I)=0 AND WETHR(I)=60 AND DIFF(I)< 2
  AND RCL(I)=0 THEN DO;
  DO K=1 TO 24 UNTIL (DIFF(K)>2 OR K=24);
  IF K=24 THEN SATUR=1;
  IF (DIFF(K)<2 AND WETHR(K)=0) THEN WETHR(K)=60;
  IF DIFF(K)>2 AND WETHR(K)=0 THEN DO;
  * DRYING TIME; *DEVIATIONS FROM MEAN DRYING TIME OF 31.6;
  WETHR(K)=WETHR(K)*31.6;
  * TEMPERATURE RECORDS;
  IF TEMP(K)<67.5 THEN WETHR(K)=WETHR(K)+3.7;
  IF TEMP(K)>67.5 AND TEMP(K)<82.5 THEN WETHR(K)=WETHR(K)-0.7;
  ELSE WETHR(K)=WETHR(K)-3.0;
  * RELATIVE HUMIDITY RECORDS;
  IF RH(K)<50 THEN WETHR(K)=WETHR(K)-4.5;
  IF RH(K)>50 AND RH(K)<82.5 THEN WETHR(K)=WETHR(K)-1.6;
  ELSE WETHR(K)=WETHR(K)+6.1;
  *SOLAR RADIATION RECORDS;
  IF RCL(K)<0.4 THEN WETHR(K)=WETHR(K)+11.6;
  IF RCL(K)>0.4 AND RCL(K)<0.85 THEN WETHR(K)=WETHR(K)+5.6;
  ELSE WETHR(K)=WETHR(K)-17.2;
  *WIND SPEED RECORDS;
  IF WIND(K)<1.5 THEN WETHR(K)=WETHR(K)+11.6;
  IF WIND(K)>1.5 AND RCL(K)<0.85 THEN WETHR(K)=WETHR(K)-11.6;
  *PAVEMENT MATERIAL;
  WETHR(K)=WETHR(K)+3.9;
END;

END;
END;
END;

PROC SORT; BY DATE;

DATA X3IIA;
   SET TWELVE; BY DATE;
   DATE=DATE+1;
   IF SATUR=. THEN SATUR=0;
   KEEP DATE SATUR;

PROC SORT; BY DATE;

DATA X3IIB;
   MERGE X3IIA TWELVE;
   BY DATE;
   ARRAY WETHR[24] WHR1-WHR24;
   ARRAY RCL[24] RCL1-RCL24;
   ARRAY DIFF[24] DIFF1-DIFF24;
   ARRAY TEMP[24] TEMP1-TEMP24;
   ARRAY WIND[24] WIND1-WIND24;
   IF SATUR=1 THEN DO;
       DO I=1 TO 24;
           IF WETHR[I]=0 AND RCL[I]<0 AND DIFF[I]<2 THEN DO;
               DO K=I TO 24 UNTIL (DIFF[K]>2 OR K=24);
               IF (DIFF[K]<2 AND K<H) THEN WETHR[K]=60;
               IF DIFF[K]>2 AND WETHR[K]=0 THEN DO;
                   WETHR[K]=WETHR[K]+31.6;
                   * DRYING TIME; * DEVIATIONS FROM MEAN DRYING TIME OF 31.6;
                   WETHR[K]=WETHR[K]+3.7;
                   * TEMPERATURE RECORDS;
                   IF TEMP[K]<67.5 THEN WETHR[K]=WETHR[K]-0.7;
                   ELSE WETHR[K]=WETHR[K]-3.0;
                   * RELATIVE HUMIDITY RECORDS;
                   ELSE WETHR[K]=WETHR[K]+6.1;
                   * SOLAR RADIATION RECORDS;
                   IF RCL[K]<0.4 THEN WETHR[K]=WETHR[K]+11.6;
                   IF RCL[K]>0.4 AND RCL[K]<0.85 THEN WETHR[K]=WETHR[K]+5.6;
                   ELSE WETHR[K]=WETHR[K]-17.2;
                   * WIND SPEED RECORDS;
                   IF WIND[K]>1.5 AND RCL[K]<0.85 THEN WETHR[K]=WETHR[K]-11.6;
                   ELSE WETHR[K]=WETHR[K]-1.6;
                   * PAVEMENT MATERIAL;
                   WETHR[K]=WETHR[K]+3.9;
           END;
       END;
   END;
END;
END;
END;

DATA X3IIIC;
   SET X3IIB;
   ARRAY WETHR[24] WHR1-WHR24;
   ARRAY PCWHR[24] PCWHR1-PCWHR24;
   ARRAY MISSING[24] MISS1-MISS24;
```plaintext
CWHR[I]=WETHR[I];
IF WETHR[I]<>60 AND WETHR[I]<>0 THEN DO;
PCWHR[I]=WETHR[I]-7.8;
END;
DO I=1 TO 24;
IF I=1 THEN ASUMHR=0;
IF I=1 THEN PSUMHR=0;
IF I=1 THEN ASUMMI=0;
IF I=1 THEN PSUMMI=0;
ASUMHR=SUM(ASUMHR,WETHR[I]);
PSUMHR=SUM(PSUMHR,PCWHR[I]);
ASUMMI=SUM(ASUMMI,MISSING[I]);
PSUMMI=SUM(PSUMMI,MISSING[I]);
END;
ATOTHR=ASUMHR/60;
PTOTHR=PSUMHR/60;
ATOTMIS=ASUMMI;
PTOTMIS=PSUMMI;
PROC SORT; BY YEAR;
PROC MEANS NOPRINT SUM N;
VAR ATOTHR ASUMHR ATOTMIS PTOTHR PSUMHR PTOTMIS;
BY YEAR;
OUTPUT OUT=STAT SUM=ATOTHR ATOTMIN AMISSING PTOTHR PTOTMIN PMISSING N=NUM NUM1 NUM2 NUM3 NUM4 NUM5;
DATA XIII;
SET STAT;
LABEL NUM= NUMBER OF DAYS
ATOTHR=TOTAL WET HOURS (ASPHLT)
ATOTMIN=TOTAL WET MINUTES (ASPHLT)
AMISSING=MISSING HOURS
APROPRWT= PROPORTION WET (ASPHLT)
PTOTHR=TOTAL WET HOURS (PRLNLD)
PTOTMIS=TOTAL WET MINUTES (PRLNLD)
PMISSING=MISSING HOURS
PPROPRWT= PROPORTION WET (PRLNLD);
APROPRWT=(ATOTHR)/((NUM-(AMISSING/24))*24);
PPROPRWT=(PTOTHR)/((NUM-(PMISSING/24))*24);
DROP NUM1 NUM2 NUM3 NUM4 NUM5 PMISSING;
IF APROPRWT=0 AND PPROPRWT=0 THEN DELETE;
DATA _NULL_;
SET XIII;
FILE OUT NOPRINT NOTITLE MOD;
PUT @ 2 YEAR @ 10 NUM @ 20 ATOTHR @ 30 ATOTMIN @ 40 AMISSING
@ 44 APROPRWT @ 56 PTOTHR @ 71 PTOTMIN @ 85 PPROPRWT;
PROC PRINT DATA=XIII LABEL;
TITLE 'WET HOUR PERCENTAGES BY MONTH FOR BATON ROUGE PAVEMENTS';
//
```
DATA ONE;
INFILE SEC;
  INPUT INDEX NAME & $14. LATD LATH LOND LONM PARISH NORMAL;
  M=1;

DATA TWO;
INFILE PRI;
  INPUT INDEX1 PARISH1 LATD1 LATH1 LOND1 LONM1 NORMAL1
  INDEX2 PARISH2 LATD2 LATH2 LOND2 LONM2 NORMAL2
  INDEX3 PARISH3 LATD3 LATH3 LOND3 LONM3 NORMAL3
  INDEX4 PARISH4 LATD4 LATH4 LOND4 LONM4 NORMAL4;
  M=1;

DATA THREE;
MERGE ONE TWO;
BY M;

DROP PARISH1-PARISH4;
ARRAY INDEXP[5] INDEX INDEX1-INDEX4;
ARRAY LATDP[5] LATD LATD1-LATD4;
ARRAY LATHP[5] LATH LATM1-LATM4;
ARRAY LONDP[5] LOND LOND1-LOND4;
ARRAY LONMP[5] LONM LONM1-LONM4;
ARRAY LATI[5] LAT1-LAT5;
ARRAY LONI[5] LON1-LON5;
ARRAY DIST[5] DIS1-DISS;
ARRAY DISTA[5] DISA1-DISA5;
ARRAY NORMALP[5] NORMAL NORMAL1-NORMAL4;

D2R=ATAN(1)/45;

DO I= 1 TO 5;
  LATI[I]=D2R*(LATDP[I]+(LATMP[I]/60));
  LONI[I]=D2R*(LONDP[I]+(LONMP[I]/60));
END;

DO I=2 TO 5;
  DIST[I]=SIN(LATI[I])*SIN(LATI[I]);
  DISTA[I]=COS(LATI[I])*COS(LATI[I])*COS(ABS(LONI[I]-LONI[I]));
  DIST[I]= ARCCOS(DIST[I]+ DISTA[I]);
END;

MINDIS=MIN(DIST,DIST3,DIST4,DIST5);
SMINDIS=MAX(DIST,DIST3,DIST4,DIST5);

DO I=2 TO 5;
  IF DIST[I] <= SMINDIS AND DIST[I] > MINDIS THEN
    SMINDIS=DIST[I];
  END;

DO I=2 TO 5;
  IF DIST[I]=MINDIS THEN NEAREST=INDEXP[I];
  IF DIST[I]=MINDIS THEN NNRRNL=NORMALP[I];
  IF DIST[I]=SMINDIS THEN SNNRRNL=NORMALP[I];
  IF DIST[I]=SMINDIS THEN SNEAREST=INDEXP[I];
END;

DROP DISA1-DISA5 LATD LOND LONM LATH LATD1-LATD4 LOND1-LOND4 LONM1-LONM4000920099
  I INDEX1-INDEX4 LAT2-LAT5 LON2-LON5
  LATM1-LATM4 DIS1 D2R;

DATA FOUR;
INFILE WBR;
INPUT YEAR DAYS AWETHR1 AWETMIN1 MISSING APWT1 PWETHR1 PWETMIN1 PPWT1;
KEEP YEAR AWETHR1 PWETHR1 APWT1 PPWT1;
PROC SORT; BY YEAR;

DATA FIVE;
INFILE WLC;
INPUT YEAR DAYS AWETHR2 AWETMIN2 MISSING APWT2 PWETHR2 PPWT2;
KEEP YEAR AWETHR2 PWETHR2 APWT2 PPWT2;
PROC SORT; BY YEAR;

DATA SIX;
INFILE WNO;
INPUT YEAR DAYS AWETHR3 AWETMIN3 MISSING APWT3 PWETHR3 PPWT3;
KEEP YEAR AWETHR3 PWETHR3 APWT3 PPWT3;
PROC SORT; BY YEAR;

DATA SIXA;
INFILE WSH;
INPUT YEAR DAYS AWETHR4 AWETMIN4 MISSING APWT4 PWETHR4 PPWT4;
KEEP YEAR AWETHR4 PWETHR4 APWT4 PPWT4;
PROC SORT; BY YEAR;

DATA SEVEN;
MERGE FOUR FIVE SIX SIXA;

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=AYR 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=APPT 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;
SET OUT9;
SET OUT10;
SET OUT10A;
SET OUT11;
SET OUT12;
SET OUT13;
SET OUT13A;
M=1;

DATA NINE;
MERGE EIGHT THREE;
BY M;

ARRAY ABWH[50] ABWH1-ABWH50;
ARRAY PBWH[50] PBWH1-PBWH50;
ARRAY ABPT[50] ABPT1-ABPT50;
ARRAY PBPT[50] PBPT1-PBPT50;
ARRAY ALWH[50] ALWH1-ALWH50;
ARRAY PLWH[50] PLWH1-PLWH50;
ARRAY ALPT[50] ALPT1-ALPT50;
ARRAY PLPT[50] PLPT1-PLPT50;
ARRAY ANWH[50] ANWH1-ANWH50;
ARRAY PNWH[50] PNWH1-PNWH50;
ARRAY ANPT[50] ANPT1-ANPT50;
ARRAY PNPT[50] PNPT1-PNPT50;
ARRAY ASWH[50] ASWH1-ASWH50;
ARRAY PSWH[50] PSWH1-PSWH50;
ARRAY ASPT[50] ASPT1-ASPT50;
ARRAY PSPT[50] PSPT1-PSPT50;
ARRAY AWHX[50] AWHX1-AWHX50;
ARRAY PWXH[50] PWXH1-PWXH50;
ARRAY APWTX[50] APWTX1-APWTX50;
ARRAY PPWTX[50] PPWTX1-PPWTX50;
ARRAY YR[50] YR1-YR50;

DO I=1 TO 50;
IF YR[I]<. AND MINDIS = 0 THEN DO;

*BAYON ROUGE SHREVEPORT BEING TWO NEAREST;
   IF (NEAREST=549 AND SNEAREST=8440)
      OR (NEAREST=8440 AND SNEAREST=549) THEN DO;
      AWHX[I]=(ABWH[I]*DIS5*(NORMAL/NORMAL1));
      AWHX[I]=AWHX[I]+(ASWH[I]*DIS2*(NORMAL/NORMAL4));
      AWHX[I]=AWHX[I]/(DIS5+DIS2);
      PWXH[I]=(PBWH[I]*DIS5*(NORMAL/NORMAL1));
      PWXH[I]=PWXH[I]+(PSWH[I]*DIS2*(NORMAL/NORMAL4));
      PWXH[I]=PWXH[I]/(DIS5+DIS2);
      APWTX[I]=(ABPT[I]*DIS5*(NORMAL/NORMAL1));
      APWTX[I]=APWTX[I]+(ASPT[I]*DIS2*(NORMAL/NORMAL4));
      APWTX[I]=APWTX[I]/(DIS5+DIS2);
      PPWTX[I]=(PBPT[I]*DIS5*(NORMAL/NORMAL1));
      PPWTX[I]=PPWTX[I]+(PSPT[I]*DIS2*(NORMAL/NORMAL4));
      PPWTX[I]=PPWTX[I]/(DIS5+DIS2);
   END;

*SHREVEPORT NEW ORLEANS BEING TWO NEAREST;
   IF (NEAREST=8440 AND SNEAREST=6660)
      OR (NEAREST=6660 AND SNEAREST=8440) THEN DO;
      AWHX[I]=(ASWH[I]*DIS4*(NORMAL/NORMAL4));
      AWHX[I]=AWHX[I]+(ANWH[I]*DIS5*(NORMAL/NORMAL3));
      AWHX[I]=AWHX[I]/(DIS4+DIS5);
      PWXH[I]=PWXH[I]+(PNWH[I]*DIS4*(NORMAL/NORMAL4));
      PWXH[I]=PWXH[I]/(DIS4+DIS5);
      APWTX[I]=APWTX[I]+(PNPT[I]*DIS4*(NORMAL/NORMAL4));
      APWTX[I]=APWTX[I]/(DIS4+DIS5);
      PPWTX[I]=PPWTX[I]+(PSPT[I]*DIS2*(NORMAL/NORMAL4));
      PPWTX[I]=PPWTX[I]/(DIS5+DIS2);
   END;
A WHX[I] = A WHX[I] / (DIS5 + DIS4);

P WHX[I] = (P SHW[I] * DIS4 * (NORMAL / NORMAL4));
P WHX[I] = P WHX[I] + (P NW[I] * DIS5 * (NORMAL / NORMAL3));
P WHX[I] = P WHX[I] / (DIS5 + DIS4);

A PWTX[I] = (AP SPT[I] * DIS4 * (NORMAL / NORMAL4));
P WTX[I] = A PWTX[I] + (AP NT[I] * DIS5 * (NORMAL / NORMAL3));
P WTX[I] = A PWTX[I] / (DIS5 + DIS4);

A WTX[I] = (ALPT[I] * DIS5 * (NORMAL / NORMAL2));
P WTX[I] = A WTX[I] + (ASPT[I] * DIS3 * (NORMAL / NORMAL4));
P WTX[I] = A WTX[I] / (DIS5 + DIS3);

A WHX[I] = (AWHX[I] * DIS5 * (NORMAL / NORMAL2));
P WHX[I] = P WHX[I] + (PSHW[I] * DIS3 * (NORMAL / NORMAL4));
P WHX[I] = P WHX[I] / (DIS5 + DIS3);

A PWTX[I] = (ALPT[I] * DIS5 * (NORMAL / NORMAL2));
P WTX[I] = A PWTX[I] + (APNT[I] * DIS3 * (NORMAL / NORMAL4));
P WTX[I] = A PWTX[I] / (DIS5 + DIS3);

A PWTX[I] = (PLPT[I] * DIS5 * (NORMAL / NORMAL2));
P WTX[I] = P WTX[I] + (PSPT[I] * DIS3 * (NORMAL / NORMAL4));
P WTX[I] = P WTX[I] / (DIS5 + DIS3);

* SHREVEPORT AND LAKE CHARLES BEING THE TWO NEAREST;

IF (NEAREST = 5078 AND SNEAREST = 8440) OR (NEAREST = 8440 AND SNEAREST = 5078) THEN DO;
A WHX[I] = (ALWH[I] * DIS5 * (NORMAL / NORMAL2));
P WHX[I] = P WHX[I] + (ANWH[I] * DIS3 * (NORMAL / NORMAL4));
P WHX[I] = P WHX[I] / (DIS5 + DIS3);

A PWTX[I] = (AP SPT[I] * DIS5 * (NORMAL / NORMAL2));
P WTX[I] = A PWTX[I] + (ANPT[I] * DIS3 * (NORMAL / NORMAL4));
P WTX[I] = A PWTX[I] / (DIS5 + DIS3);

A WTX[I] = (AWHX[I] * DIS5 * (NORMAL / NORMAL2));
P WTX[I] = A WTX[I] + (ASPT[I] * DIS3 * (NORMAL / NORMAL4));
P WTX[I] = A WTX[I] / (DIS5 + DIS3);

* BAYON ROUGE NEW ORLEANS BEING TWO NEAREST;

IF (NEAREST = 549 AND SNEAREST = 6660) OR (NEAREST = 6660 AND SNEAREST = 549) THEN DO;
A WHX[I] = (ABWH[I] * DIS4 * (NORMAL / NORMAL1));
P WHX[I] = P WHX[I] + (ANWH[I] * DIS2 * (NORMAL / NORMAL3));
P WHX[I] = P WHX[I] / (DIS4 + DIS2);

A PWTX[I] = (ABPT[I] * DIS4 * (NORMAL / NORMAL1));
P WTX[I] = A PWTX[I] + (APNT[I] * DIS2 * (NORMAL / NORMAL3));
P WTX[I] = A PWTX[I] / (DIS4 + DIS2);

A WTX[I] = (AWHX[I] * DIS4 * (NORMAL / NORMAL1));
P WTX[I] = A WTX[I] + (ASPT[I] * DIS2 * (NORMAL / NORMAL3));
P WTX[I] = A WTX[I] / (DIS4 + DIS2);

* NEW ORLEANS AND LAKE CHARLES BEING THE TWO NEAREST;

IF (NEAREST = 5078 AND SNEAREST = 6660) OR (NEAREST = 6660 AND SNEAREST = 5078) THEN DO;
A WHX[I] = (ALWH[I] * DIS4 * (NORMAL / NORMAL2));
P WHX[I] = P WHX[I] + (ANWH[I] * DIS2 * (NORMAL / NORMAL3));
P WHX[I] = P WHX[I] / (DIS4 + DIS2);
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)); 02290099
PWHX[I]=PWHX[I]+(PLWH[I]*DIS3*(NORMAL/NORMAL3)); 02300099
PWHX[I]=PWHX[I]/(DIS4+DIS3); 02310099
02320099
APWXTX[I]=(ALPT[I]*DIS4*(NORMAL/NORMAL2)); 02290099
APWXTX[I]=APWXTX[I]+(ANPT[I]*DIS3*(NORMAL/NORMAL3)); 02300099
APWXTX[I]=APWXTX[I]/(DIS4+DIS3); 02310099
02320099
PFPWTX[I]=(PLPT[I]*DIS4*(NORMAL/NORMAL2)); 02290099
PPFWTX[I]=PPFWTX[I]+(PNPT[I]*DIS3*(NORMAL/NORMAL3)); 02300099
PPFWTX[I]=PPFWTX[I]/(DIS4+DIS3); 02310099
02320099
END;
02400099
02410099
02420099
02430099
02440099
02450099
02460099
02470099
02480099
02490099
02500099
02510099
02520099
02530099
02540099
02550099
02560099
02570099
02580099
02590099
02600099
02610099
02620099
02630099
02640099
02650099
02660099
02670099
02680099
02690099
02700099
02710099
02720099
02730099
02740099
02750099
02760099
02770099
02780099
02790099
02800099
02810099
02820099
02830099
02840099
02850099
02860099
02870099
02880099
Y=LAT1; X=lon1;
KEEP YR1-YR50 AWHX1-AWHX50 PWHX1-PWHX50 APWXTX1-APWXTX50 PPFWTX1-PPFWTX50
INDEX Y X ;

GOPTIONS NOTEXT82;
GOPTIONS HSIZE=12 VSIZE=12;
GOPTIONS DEVICE=Ben9215;
DATA ONE;
LENGTH COUNTY 4;
SET NINE;
COUNTY=-1;
IF INDEX=4842.2 OR INDEX=1707.2 THEN DELETE;
DATA MAP;
SET MAPS.COUNTIES;
IF STATE=22;
DATA MAP1;
SET MAP ONE;
PROC GPROJECT DATA=MAP1
   OUT=PROJCOM
   PROJECT=ALBERS
   PARALEL1=29.5
   PARALEL2=45.5;
IF COUNTY;
DATA PROJLOU PROJSTA ;
SET PROJCOM;
IF COUNTY = -1 THEN OUTPUT PROJSTA;
ELSE OUTPUT PROJLOU;
PROC G3GRID DATA=PROJSTA OUT=GRIDONE OUTTRI=TRIONE;
GRID X*Y=YM1/ AXIS1=-0.039 TO 0.04 BY 0.05
        AXIS2=-0.18 TO -0.075 BY 0.05;
PROC MEANS NOPRINT DATA=TRIONE;
OUTPUT OUT=OUTA N=N;
BY TRIANGLE;
DATA TRITWO;
MERGE TRIONE OUTA;
BY TRIANGLE;
IF N != 3 THEN DELETE;
X=ROUND(X,0.0000001);
Y=ROUND(Y,0.0000001);
DATA PROJST;
SET PROJSTA;
X=ROUND(X,0.0000001);
Y=ROUND(Y,0.0000001);
DROP COUNTY SEGMENT STATE DENSITY;
PROC SORT ; BY X Y;
PROC TRANPOSE DATA=TRITWO PREFIX=X OUT=OUTB1;
VAR X;
BY TRIANGLE;
PROC TRANPOSE DATA=TRITWO PREFIX=Y OUT=OUTB2;
VAR Y;
BY TRIANGLE;
DATA COMB2;
SET OUTB1;
SET OUTB2;
BY TRIANGLE;
DROP _NAME_ _LABEL_;

PROC SORT; BY X1 Y1;

DATA COMB3; SET PROJST; X1=X; Y1=Y; ARRAY AWHX[50] AWHX1-AWHX50; ARRAY AWX[50] AWX1-AWX50; ARRAY PWHX[50] PWHX1-PWHX50; ARRAY PWX[50] PWX1-PWX50; ARRAY APWTX[50] APWTX1-APWTX50; ARRAY APW1X[50] APW1X1-APW1X50; ARRAY PWXTX[50] PWXTX1-PWXTX50; ARRAY PW1X[50] PW1X1-PW1X50; ARRAY YR[50] YR1-YR50; DO I=1 TO 50; AW1X[I]=AWHX[I]; PW1X[I]=PWHX[I]; APW1X[I]=APWTX[I]; PWXTX[I]=PW1X[I]; END;

DROP X Y AWHX1-AWHX50 PWHX1-PWHX50 APWTX1-APWTX50 PWXTX1-PWXTX50;
PROC SORT; BY X1 Y1;

DATA COMB4; MERGE COMB2 COMB3; BY X1 Y1;
if x2=. and y2=. and x3=. and y3=. then delete;

PROC SORT DATA=COMB4; BY X2 Y2;

DATA COMB5; SET PROJST; X2=X; Y2=Y; ARRAY AWHX[50] AWHX1-AWHX50; ARRAY AWX[50] AWX1-AWX50; ARRAY PWHX[50] PWHX1-PWHX50; ARRAY PWX[50] PWX1-PWX50; ARRAY APWTX[50] APWTX1-APWTX50; ARRAY APW2X[50] APW2X1-APW2X50; ARRAY PWXTX[50] PWXTX1-PWXTX50; ARRAY PW2X[50] PW2X1-PW2X50; ARRAY YR[50] YR1-YR50; DO I=1 TO 50; AW2X[I]=AWHX[I]; PW2X[I]=PWHX[I]; APW2X[I]=APWTX[I]; PWXTX[I]=PW2X[I]; END;

DROP X Y AWHX1-AWHX50 PWHX1-PWHX50 APWTX1-APWTX50 PWXTX1-PWXTX50 YR1-YR50;
PROC SORT; BY X2 Y2;

DATA COMB6; MERGE COMB4 COMB5; BY X2 Y2;
if x3=. and y3=. and x1=. and y1=. then delete;

PROC SORT; BY X3 Y3;
DATA COMB7; SET PROJST; X3=X; Y3=Y;
ARRAY AWHX[50] AWHX1-AWHX50;
ARRAY AW3X[50] AW3X1-AW3X50;
ARRAY PWHX[50] PWHX1-PWHX50;
ARRAY PW3X[50] PW3X1-PW3X50;
ARRAY APWTX[50] APWTX1-APWTX50;
ARRAY APW3X[50] APW3X1-APW3X50;
ARRAY PPWTX[50] PPWTX1-PPWTX50;
ARRAY PPW3X[50] PPW3X1-PPW3X50;
ARRAY YR[50] YR1-YR50;

DO I=1 TO 50;
AW3X[I]=AWHX[I];
PWHX[I]=PW3X[I];
APW3X[I]=APWTX[I];
PPW3X[I]=PPWTX[I];
END;

DROP X Y AWHX1-AWHX50 PWHX1-PWHX50 APWTX1-APWTX50 PPWTX1-PPWTX50 YR1-YR50;
PROC SORT; BY X3 Y3;
DATA COMB8; MERGE COMB6 COMB7; BY X3 Y3;
if x2=. and y2=. and x1=. and y1=. then delete;

ARRAY A2X[50] A2X1-A2X50;
ARRAY A3X[50] A3X1-A3X50;
ARRAY PWX[50] PWX1-PWX50;
ARRAY PW2X[50] PW2X1-PW2X50;
ARRAY PW3X[50] PW3X1-PW3X50;
ARRAY APW1X[50] APW1X1-APW1X50;
ARRAY APW2X[50] APW2X1-APW2X50;
ARRAY APW3X[50] APW3X1-APW3X50;
ARRAY PPW1X[50] PPW1X1-PPW1X50;
ARRAY PPW2X[50] PPW2X1-PPW2X50;
ARRAY PPW3X[50] PPW3X1-PPW3X50;
ARRAY YR[50] YR1-YR50;
ARRAY AAW1X[50] AAW1X1-AAW1X50;
ARRAY AAW2X[50] AAW2X1-AAW2X50;
ARRAY AAW3X[50] AAW3X1-AAW3X50;
ARRAY APAW1X[50] APAW1X1-APA1WX50;
ARRAY APAW2X[50] APAW2X1-APA2WX50;
ARRAY APAW3X[50] APAW3X1-APA3WX50;
ARRAY AAPW1X[50] AAPW1X1-AAP1WX50;
ARRAY AAPW2X[50] AAPW2X1-AAP2WX50;
ARRAY AAPW3X[50] AAPW3X1-AAP3WX50;
ARRAY APPW1X[50] APPW1X1-APP1WX50;
ARRAY APPW2X[50] APPW2X1-APP2WX50;
ARRAY APPW3X[50] APPW3X1-APP3WX50;
ARRAY AWHMN[50] AWHMN1-AWHMN50;
ARRAY PWHMN[50] PWHMN1-PWHMN50;
ARRAY APWHN[50] APWHN1-APWHN50;
ARRAY PWHHN[50] PWHHN1-PWHHN50;

B1=X2*X3-X3*Y2; B2=X3*Y1-X1*Y3; B3=X1*Y2-X2*Y1;
C1=Y2-Y3; C2=Y3-Y1; C3=Y1-Y2;
D1=X3-X2; D2=X1-X3; D3=X2-X1;

AREA= (B1+B2+B3)/2;

DO I=1 TO 50;
AAW1X[I]=(B1*AW1X[I]+B2*AW2X[I]+B3*AW3X[I])/(2*AREA);
AAW2X[I]=(C1*AW1X[I]+C2*AW2X[I]+C3*AW3X[I])/(2*AREA);
AAW3X[I]=(D1*AW1X[I]+D2*AW2X[I]+D3*AW3X[I])/(2*AREA);

AWHM[N][I]=AAW1X[I]+(AAW2X[I]*(X1+X2+X3)/3)+(AAW3X[I]*(Y1+Y2+Y3)/3);

APAW1X[I]=(B1*PW1X[I]+B2*PW2X[I]+B3*PW3X[I])/(2*AREA);
APAW2X[I]=(C1*PW1X[I]+C2*PW2X[I]+C3*PW3X[I])/(2*AREA);
APAW3X[I]=(D1*PW1X[I]+D2*PW2X[I]+D3*PW3X[I])/(2*AREA);

PHHM[N][I]=APAW1X[I]+(APAW2X[I]*(X1+X2+X3)/3)+(APAW3X[I]*(Y1+Y2+Y3)/3);

APAPW1X[I]=(B1*APW1X[I]+B2*APW2X[I]+B3*APW3X[I])/(2*AREA);
APAPW2X[I]=(C1*APW1X[I]+C2*APW2X[I]+C3*APW3X[I])/(2*AREA);
APAPW3X[I]=(D1*APW1X[I]+D2*APW2X[I]+D3*APW3X[I])/(2*AREA);

APW[N][I]=APAW1X[I]+(APAW2X[I]*(X1+X2+X3)/3)+(APAW3X[I]*(Y1+Y2+Y3)/3);

APPW1X[I]=(B1*PPW1X[I]+B2*PPW2X[I]+B3*PPW3X[I])/(2*AREA);
APPW2X[I]=(C1*PPW1X[I]+C2*PPW2X[I]+C3*PPW3X[I])/(2*AREA);
APPW3X[I]=(D1*PPW1X[I]+D2*PPW2X[I]+D3*PPW3X[I])/(2*AREA);

PPW[N][I]=APPW1X[I]+(APPW2X[I]*(X1+X2+X3)/3)+(APPW3X[I]*(Y1+Y2+Y3)/3);

END;

IF AWHMN1=. THEN DELETE;

KEEP TRIANGLE APW[N]-APW[N50]
PPWN[N]-PPWN[N50];

PROC SORT; BY TRIANGLE;

DATA EMPBAYES;
SET COMB8;
BY TRIANGLE;
G=_N_;
IF G <= 3 THEN G=G;
ELSE DO;
G=MOD(G,3);
IF G=0 THEN G=3;
END;

PROC SORT; BY G;

DATA EMPBAYI;
SET EMPBAYES; BY G;
ARRAY APW[N][50] APW[N]-APW[N50];
ARRAY PPW[N][50] PPW[N]-PPW[N50];
P=0; K=0; SUMA=0; SUMP=0;
DO I=1 TO 50;
IF APW[N][I]=. THEN DO;
SUMA=SUMA+APW[N][I];
K=K+1;
END;
IF PPW[N][I]=. THEN DO;
SUMP=SUMP+PPW[N][I];
P=P+1;
END;
END;

XIABAR=SUMA/K; XIBAR=SUMP/P;
PROC SORT; BY G;
PROC MEANS NOPRINT MEAN N;
VAR APWNN1-APWNN50 PPWNN1-PPWNN50;
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] APWNN1-APWNN50;
ARRAY PPMN[50] PPWNN1-PPWNN50;
ARRAY AMN[50] AMN1-AMN50;
ARRAY PNN[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 PPMN[I]-=. THEN DO;
   SUMSQP=SUMSQP+(PPMN[I]-(PNN[I]/PN[I]))**2;
END;
END;
PROC SORT; BY G;
PROC MEANS SUM N;
VAR SUMSQA SUMSQP;
BY G;
OUTPUT OUT=VARIANCE SUM= SUMSQA SUMSQP N=NA NP;
PROC SORT DATA=VARIANCE; BY G;
DATA EMBYIII;
MERGE VARIANCE EMPBAYII;
BY G;
ARRAY APWMN[50] APWNN1-APWNN50;
ARRAY PPMN[50] PPWNN1-PPWNN50;
ARRAY MUEBA[50] MUEBA1-MUEBA50;
ARRAY VREBA[50] VREBA1-VREBA50;
ARRAY MUEBP[50] MUEBP1-MUEBP50;
ARRAY VREBP[50] VREBP1-VREBP50;
SGSFHA=SUMSQA/(NA*(NA-1)*K);
SGSFHP=SUMSQP/(NP*(NP-1)*P);
MUPHAI=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);
BHATP=((P-3)/(P-1))*SGSFHP/(SGSFHP+SGSPIHP);
DO I= 1 TO 50;
IF APWMN[I] ~= . THEN DO;
FACT=APWMN[I]-MUPIAH;
MUEBA[I]=APWMN[I]-BHATA*FACT;
VREBA[I]=SGSFHA*((1-((K-1)/K)*BHATA) + (2/(K-3))*BHATA**2*FACT**2);
END;
IF PPWMN[I] ~= . THEN DO;
FACTP= PPWMN[I]-MUPIPH;
MUEBP[I]=PPWMN[I]-BHATP*FACTP;
VREBP[I]=SGSFHP*((1-((P-1)/P)*BHATP) + (2/(P-3))*BHATP**2*FACTP**2);
END;
END;
DO I=2 TO 50;
* GET OUT THE LATEST YEARS EMPIRICAL BAYES ESTIMATE;
IF APWMN[I] = . AND APWMN[I-1] ~= . THEN DO;
VREBAL=VREBA[I-1]; APWHNL=APWMN[I-1];
MUEBAL=MUEBA[I-1];
END;
IF PPWMN[I] = . AND PPWMN[I-1] ~= . THEN DO;
VREBPL=VREBP[I-1]; PPWHNL=PPWMN[I-1];
MUEBPL=MUEBP[I-1];
END;
END;
*PROBABILITY THAT A TRIANGLE HAS EXPERIENCED LESS NUMBER OF WET HOURS THAN THE REGIONAL NUMBER OF WET HOURS;
* P[THETHAI <= OBSERVED AVERAGE | Y];
* TRANSFORM FIRST INTO 0,1 NORMAL;
KEEP TRIANGLE C MUPIAH APWHNL PPWHNL MUEBAL VREBAL MUEBPL VREBPL
MUPIPH;
PROC MEANS NOPRINT MEAN;
VAR APWHNL PPWHNL;
BY G;
OUTPUT OUT=STATSII MEAN= APWHN PIWRMN;
PROC SORT DATA=STATSII; BY G;
DATA EMBYIV;
MERGE EMBYIII STATSII; BY G;
ZA=(APWHN-MUEBAL)/SQRT(VREBAL);
ZP=(PPWHN-MUEBPL)/SQRT(VREBPL);
WTA=PROBNORM(ZA);
WTP=PROBNORM(ZP);
LABEL MUEBAL=EMP. BAYES EST. OF MEAN
VREBAL=EMO. BAYES EST. OF VAR.
APWHNL=LATEST YEAR PROPORTION WET TIME
WTA=PROB. OF BEING LESS THAN REGIONALMEAN;
LABEL MUEBPL=EMP. BAYES EST. OF MEAN
VREBPL=EMO. BAYES EST. OF VAR.
PPWHNL=LATEST YEAR PROPORTION WET TIME
WTP=PROB. OF BEING LESS THAN REGIONALMEAN;
KEEP MUEBPL MUEBAL VREBPL VREBAL TRIANGLE PPWHNL APWHNL WTA WTP ;
PROC SORT; BY TRIANGLE;
PROC PRINT;
DATA CSEC;
INFILE CSC;
INPUT CONTROL SECTION TRIANGLE;
PROC SORT; BY TRIANGLE;
DATA TRIANGLE;
MERGE CSEC EMBYIV; BY TRIANGLE;
IF APWMNL=. THEN DELETE;
PROC PRINT;
DATA _NULL_; 
SET TRIANGLE;
FILE OUT NOPRINT NOTITLE;
PUT CONTROL SECTION TRIANGLE MUEBAL MUEBPL VREBAL VREBPL PPWMNL APWMNL 06910099
   WTP WTP;
   */
   //
//LIRC 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 00250099
//SYSIN DD * 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
***************************************************************************************************;

*READS ALL INPUT DATA FROM MASTER ACCIDENT FILE; 00320099
*OPTIONS TIME=MAX TLS=60 PS=60 LS=80; 00330099
OPTIONS NOCAPS; 00340099
OPTIONS NOTEXT82; 00350099
OPTIONS DEVICE=GDDMFAM4 GDDMNICKNAME=IBM3820 GDDMTOKEN=IMG240; 00360099
00370099
DATA MASACC;
INFILE ACCI RECfm=FB LRECl=225 BLKSIZE=11250;
00390099
00400099
INPUT
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
PAVTYPE 123-124 00560099
PAWID 125-126 00570099
ADT 134-139 ; 00580099

*ELIMINATES UNWANTED DATA;

IF BCLM<0 OR ECLM<0 THEN DELETE;
IF BCLM=0 AND ECLM=0 THEN DELETE;
IF BCLM >= ECLM THEN LENGTH=(BCLM-ECLM);
ELSE LENGTH=(ECLM-BCLM);
IF BCLM>ECLM THEN DO;
TEMP=BCLM;
BCLM=ECLM;
ECLM=TEMP;
DROP TEMP;
END;
IF ADT<=0 THEN DELETE;
IF MPH < 0 THEN DELETE;
IF INT ^=0 AND INT ^=1 THEN DELETE;
IF CONTROL<0 OR SECTION <0 THEN DELETE;

*CLASSIFIES INTERSECTIONS;
IF INT=1 THEN LTYPE='INTERSECTION';
ELSE IF INT=0 THEN LTYPE='SECTION';

IF LTYPE='INTERSECTION' THEN LENGTH=1;
IF LTYPE='INTERSECTION' THEN DELETE;

*ELIMINATES LOCAL AND ARTERIAL ROADS;
IF HTYPE ^=1 AND HTYPE ^=2 AND HTYPE ^=3 THEN DELETE;

LABEL SECTION= SECTIION NUMBER
BCLM-BEGINNING OF SECTION (LOG MILE)
ECLM=END OF SECTION (LOG MILE)
HHNUM=HIGHWAY NUMBER
MP=MILEPOST
LTYPE=LOCATION TYPE
SURCON=SURFACE CONDITION
HCLASS=HIGHWAY CLASS;

OUTPUT;

*SORTS THE ACCIDENT DATA AS GIVEN IN BY STATEMENT;

PROC SORT ;
BY LTYPE HCLASS CONTROL SECTION BCLM ;
/*
PROC FORMAT;
VALUE $SURCON
(MAX=100 MIN=0)
'A'='DRY CONDITION'
'B'='RAINY CONDITION'
'C'='MUDDY CONDITION'
'D'='SNOWY/ICY CONDITION'
'E'='OTHER CONDITIONS'
OTHER='MISSING VALUES' ;

PROC FORMAT;
VALUE HCL
(MAX=100 MIN=0)
1='RURL 2 LANE'
2='RURL OTHER'
3='RURL MULT LN DIV'
4='RURL INTERSTATE'
5='URBN 2 LANE'
6='URBN OTHER'
7='URBN MULT LN DIV'
8='URBN INTERSTATE'
OTHER='MISSING VALUES';
*/

DATA MASACC1;
SET MASACC;
*CHECKS FOR WET CRITERION;
IF SURCON='A' THEN DELETE;

* CHECKS FOR MISSING AND UNKNOWN SURFACE CONDITIONS;
IF SURCON='E' OR SURCON=' ' THEN DO;
   IF RDCON='J' AND RDCON='K' THEN DELETE;
   IF WEATH='A' OR WEATH='F' OR WEATH='G' THEN DELETE;
END;

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;
INFILE INF;
INPUT CONTROL SECTION TRIANGLE MUEBAL MUEBPL VRE3AL VREBPL PPWMNL
APWMNL WTA WTP;

PROC SORT; BY CONTROL SECTION;

DATA RAT; MERGE NEWTR3 TRI; BY CONTROL SECTION;
IF MUEBAL=.. THEN DELETE;
IF YEAR=.. AND ADT=.. THEN DELETE;
IF PAVTYP=50  
OR PAVTYP=50 THEN DO;  
MU=MUEBPL; VR=VREBPL; WMN=APWMNL; PR=WTA;  
END;  
ELSE DO;  
MU=MUEBPL; VR=VREBPL; WMN=PPWMNL; PR=WTP;  
END;  
DROP TRIANGLE MUEBPL MUEBPL VREBPL VREBPL WTA WTP APWMNL PPWMNL;  
PROC SORT; BY LTYPE HCLASS CONTROL SECTION;  
DATA RATE; SET RAT;  
BY LTYPE HCLASS CONTROL SECTION;  
*CALCULATES THE ACCIDENTS PER MILLION VEHICLE MILES;  
*EXPOSURE IS DEPENDENT ON PROPORTION WET TIME AND LENGTH;  
ADT=ADT*LENGTH*WMN;  
MVM=365*ADT/1000000;  
ACC_MVM=TOT_ACC/MVM;  
ACC_MIL=TOT_ACC/LENGTH;  
VOL=ADT*365;  
SQRVOL=VOL**2;  
TRU_RAT=TOT_ACC/(MU*LENGTH*ADT*365);  
NIBYVI=TOT_ACC/VOL;  
nibysqvi=tot_acc/sqrvol;  
onebyvol=1/vol;  
LABEL TOT_ACC=ACCIDENTS PER INTERSECTION  
ACC_MIL=ACCIDENTS PER MILE  
ACC_MVM=ACCIDENTS PER MILLION VEHICLE MILES;  
OUTPUT;  
PROC MEANS NOPRINT SUM MEAN STD VAR RANGE N;  
VAR TOT_ACC ACC_MVM ADT NIBYVI NIBYSQVI ONEBYVOL TRU_RAT;  
BY LTYPE HCLASS;  
*FORMAT HCLASS HCL.;  
OUTPUT OUT=STATAV SUM=STATSUM MVSMV SUM ADTSUM NVSUM  
MEAN=STATSTD MVSTD STAADT STANV STASQV MAVR VOL MNR  
STD=STATSTD MVSTD ADTSSTD NVSTD NVSSID ONESTD TRSTD  
VAR=TOTVAR MVVVAR ADTVAR NVVAR NVVAR  
RANGE=TOTRANGE MVMRG ADTRG VRNG NSRNG  
N=TNR MN AN NVN NSN;  
PROC SORT DATA=STATAV;  
BY LTYPE HCLASS;  
DATA COMBINE;  
MERGE RATE STATAV;  
BY LTYPE HCLASS;  
* FLAGGING BY C2 RATE QUALITY CONTROL METHOD;  
SIG_NBYV=STATSUM/(ADTSUM*365);  
C2=SIG_NBYV+(1.645*SQRT(SIG_NBYV/VOL)) + (1/(2*VOL));  
*FLAGGING BY BAYESIAN CRITERION 2;
BETA=STANV/NVVAR;
ALPHA=BETA*STANV;

BETAI=BETA+VOL;
ALPHAI=ALPHA + TOT_ACC;

ADJ_STA1= STANV*BETAI;
ADJ_STA2= SIG_NBYV*BETAI;

B2= 1-PROBGAM(ADJ_STA2,ALPHAI);

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

DATA OUT.SNCLS88;
SET COMBINE;

//
//LTRC  JOB (1318,05886,10,9999), 'SRIKANTH', NOTIFY=IEKLEE, 00010099
// MSGCLASS=S 00020099
//*JOBPARM  SHIFT=N 00030099
//STEP1  EXEC SAS, TIME=10 00040099
//ACC1 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 00240099
//SYST DD * 00250099

* TO RUN THE PROGRAM FOR 1989 AND 1990 CHANGE ACC1 STATEMENT IN THE JCL 00260099
TO ACC.MASTER89 AND ACC.MASTER90 RESPECTIVELY;

******************************************************************************;
* THIS PROGRAM FLAGS INTERSECTIONS AND SECTIONS FOR THE BAYESIAN; 00270099
* CRITERIONS DEVELOPED FOR ACCIDENT DATA IN LOUISIANA *; 00280099
******************************************************************************;

*READS ALL INPUT DATA FROM MASTER ACCIDENT FILE; 00290099
*OPTIONS TIME=MAX TLS=80 PS=60 LS=80; 00300099
OPTIONS NOCAPS; 00310099
GOPTIONS NOTEXT2; 00320099
GOPTIONS DEVICE=GDDMFAM4 GDDMNICKNAME=IBM3820 GDDMTOKEN=IMG240; 00330099

DATA MASACC; 00340099
     INFILE ACC1 RECFM=FB LRECL=225 BLKSIZE=11250; 00350099
     INPUT 00360099
         YEAR 13 00400099
         PARISH 17-18 00410099
         MP 25-28 1 00420099
         HNUM $ 29-33 00430099
         HTYPE 34 00440099
         SURCON $ 55 00450099
         RDCON $ 58 00460099
         WEATH $ 60 00470099
         INT 78 00480099
         HCLASS 87 00490099
         CONTROL 101-103 00500099
         SECTION 104-105 00510099
         CLOCA 106-109 2 00520099
         BCLM 115-118 2 00530099
         ECLM 119-122 2 00540099
         PAVTYP 123-124 00550099
         PAVWID 125-126 00560099
         ADT 134-139 ; 00570099

*ELIMINATES UNWANTED DATA; 00580099
     IF BCLM<0 OR ECLM<0 THEN DELETE; 00590099
     IF BCLM=0 AND ECLM=0 THEN DELETE; 00600099
     IF BCLM >= ECLM THEN LENGTH=(BCLM-ECLM); 00610099
ELSE LENGTH=(ECLM-BCLM); 00620099
     IF BCLM=ECLM THEN DO; 00630099
         TEMP=BCLM; 00640099
         BCLM=ECLM; 00650099
         ECLM=TEMP; 00660099
         DROP TEMP; 00670099
     END; 00680099
     IF ADT<=0 THEN DELETE; 00690099
     IF MP < 0 THEN DELETE; 00700099
     IF INT ~=0 AND INT ~=1 THEN DELETE; 00710099
IF CONTROL<0 OR SECTION <0 THEN DELETE;

*CLASSIFIES INTERSECTIONS;
IF INT=1 THEN LTYPE='INTERSECTION';

*CLASSIFIES SECTIONS;
ELSE IF INT=0 THEN LTYPE='SECTION';

IF LTYPE='INTERSECTION' THEN LENGTH=1;

*ELIMINATES LOCAL AND ARTERIAL ROADS;
IF HTYPE ^=1 AND HTYPE ^=2 AND HTYPE ^=3 THEN DELETE;

LABEL SECTION= SECTION NUMBER
  BCLM=BEGINNING OF SECTION (LOG MILE)
  ECLM=END OF SECTION (LOG MILE)
  HNUM=HIGHWAY NUMBER
  HPL=MILEPOST
  LTYPE=LOCATION TYPE
  SURCON=SURFACE CONDITION
  HCLASS=HIGHWAY CLASS;

OUTPUT;

*SORTS THE ACCIDENT DATA AS GIVEN IN BY STATEMENT;

PROC SORT ;
BY LTYPE HCLASS CONTROL SECTION BCLM ;

/*PROC FORMAT;
  VALUE $SURCON
    (MAX=100 MIN=0)
    'A'='DRY CONDITION'
    'B'='RAINY CONDITION'
    'C'='MUDY CONDITION'
    'D'='SNOWY/ICY CONDITION'
    'E'='OTHER CONDITIONS'
    OTHER='MISSING VALUES';

PROC FORMAT;
  VALUE HCL
    (MAX=100 MIN=0)
    1='RURL 2 LANE'
    2='RURL OTHER'
    3='RURL MULT LN DIV'
    4='RURL INTERSTATE'
    5='URBN 2 LANE'
    6='URBN OTHER'
    7='URBN MULT LN DIV'
    8='URBN INTERSTATE'
    OTHER='MISSING VALUES';
  */

DATA HSCACC1;
SET HSCACC;
*CHECKS FOR WET CRITERION;
IF SURCON='A' THEN DELETE;

* CHECKS FOR MISSING AND UNKNOWN SURFACE CONDITIONS;
IF SURCON='E' OR SURCON=' ' THEN DO;
  IF RDCON='J' AND RDCON='K' THEN DELETE;
  IF WEATH='A' OR WEATH='F' OR WEATH='G' THEN DELETE;

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;
INFILE OUT;
INPUT CONTROL SECTION TRIANGLE MUEBAL MUEBPL VREBAL VREBPL PPWHNL APWHNL WTA WTP;

PROC SORT; BY CONTROL SECTION;

DATA RAT; MERGE MASACC3 TK1; BY CONTROL SECTION;
IF MUEBAL=. THEN DELETE;
   IF YEAR=. AND ADT=. THEN DELETE;
   IF PAVTYPE=50
   OR PAVTYPE=60 THEN DO;
   MU=MUEBAL; VR=VREBAL; WMN=APWHNL; PR=WTA;
   END;

   ELSE DO;
   MU=MUEBPL; VR=VREBPL; WMN=PPWHNL; PR=WTP;
   END;

   DROP TRIANGLE MUEBAL MUEBPL VREBAL VREBPL WTA WTP APWHNL PPWHNL;

PROC SORT; BY LTYPE HCLASS CONTROL SECTION;

DATA RATE; SET RAT;
BY LTYPE HCLASS CONTROL SECTION;

*CALCULATES THE ACCIDENTS PER MILLION VEHICLE MILES;

*EXPOSURE IS DEPENDENT ON PROPORTION WET TIME AND LENGTH;
ADT=ADT*LENGTH*WMN;
MVM=365*ADT/1000000;
ACC_MVM=TOT_ACC/MVM;
ACC_MIL=TOT_ACC/LENGTH;
VOL=ADT*365;
SQRVOL=VOL**2;
TRU_RAT=TOT_ACC/(MU*LENGTH*ADT*365);
NIBYVI=TOT_ACC/VOL;

01440099
01450099
01460099
01470099
01480099
01490099
01500099
01510099
01520099
01530099
01540099
01550099
01560099
01570099
01580099
01590099
01600099
01610099
01620099
01630099
01640099
01641199
01641299
01690099
01700099
01710099
01760099
01820099
01830099
01840099
01850099
01860099
01870099
01880099
01890099
01900099
01910099
NIBSYQVI=TOT_ACC/SQRVOL;
ONEBYVOL=1/VOL;

LABEL TOT_ACC=ACCIDENTS PER INTERSECTION
   ACC_MIL=ACCIDENTS PER MILE
   ACC_MVM=ACCIDENTS PER MILLION VEHICLE MILES;

OUTPUT;

PROC MEANS NOPRINT SUM MEAN STD VAR RANGE N;
VAR TOT_ACC ACC_MVM ADT NIBYVI NIBSYQVI ONEBYVOL TRU_RAT;
BY LTYPE HCLASS;
*FORMAT HCLASS HCL.;
OUTPUT OUT=STATAV SUM=STATSUM MVSMV ADTSMV NVSMV MANAGE;
MEAN=STATOT STAMVM STAADT STANV STASQV NVHRVL MVHRVL
STD=STASTD MVSTSD ADTSTD NVSTD NENSTD TPRSTD
VAR=TOTVAR MVVAR ADTVAR NVVAR NSVVAR
RANGE=TOTRG MVNRG ADTRG NVRNG NSRNG
N=TN MN AN NVN NSN;

PROC SORT DATA=STATAV;
BY LTYPE HCLASS;

DATA COMBINE;
MERGE RATE STATAV;
BY LTYPE HCLASS;

* FLAGGING BY C2;

SIG_NBYV=STATSUM/(ADTSMV+365);

C2=SIG_NBYV+(1.645 *SQRT(SIG_NBYV/VOL)) + (1/(2*VOL));

*FLAGGING BY BAYESIAN CRITERION 2;

BETA=STANV/NVVAR;
ALPHA=BETA*STANV;

BETAI=BETA*VOL;
ALPHA1=ALPHA + TOT_ACC;

ADJ_STA1= STANV*BETAI;
ADJ_STA2= SIG_NBYV*BETAI;

B2= 1-PROB(XGAM(ADJ_STA2,ALPHA1));

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

DATA OUT2 SNINT88;
SET COMBINE ;
IEKLEE.LTRC.FINAL(PGM5)

//LTRC JOB (1318,05886,10,9999), 'SRIKANTH', NOTIFY=IEKLEE,
// MSGCLASS=S, CLASS=H
//*JOBPARM SHIFT=H
//STEP1 EXEC SAS606, TIME=10
//ACCI DD DSN=IEKLEE.ACC.MASTER88, DISP=SHR
//OUT DD DSN=IEKLEE.SRIKANTH.OUT, DISP=SHR
//OUT2 DD DSN=IEKLEE.NEW2.SASDATA, DISP=OLD
//SYSDN DD *

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

******************************************************************************
* THIS PROGRAM FLAGS SPOTS BY THE WET ACCIDENT*
* CRITERIONS DEVELOPED FOR ACCIDENT DATA IN LOUISIANA *
******************************************************************************

*READS ALL INPUT DATA FROM MASTER ACCIDENT FILE;
*OPTIONS TIME=MAX TLS=80 PS=60 LS=80;
OPTIONS NOCAPS;
GOPTIONS NOTEXT82;
GOPTIONS DEVICE=GDDFM4A GDDMNICKNAME=IBM3820 GDDMTOKEN=IMG240;

DATA MASACC;
  INFILE ACCI RECFM=F8 LRECL=225 BLKSIZE=11250;
  INPUT
    YEAR 13
    PARISH 17-18
    MP 25-28 1
    HNUM $ 29-33
    HTYPE 34
    SURCON $# 55
    RDCON $# 58
    WEATH $# 60
    INT 78
    HCLASS 87
    CONTROL 101-103
    SECTION 104-105
    CLOCA 106-109 2
    BCLM 115-118 2
    ECLM 119-122 2
    PAVTYP 123-124
    PAVWID 125-126
    ADT 134-139 ;

*ELIMINATES UNWANTED DATA;
  IF BCLM<0 OR ECLM<0 THEN DELETE;
  IF BCLM=0 AND ECLM=0 THEN DELETE;
  IF BCLM >= ECLM THEN LENGTH=(BCLM-ECLM);
ELSE LENGTH=(ECLM-BCLM);
  IF BCLM=ECLM THEN DO;
    TEMP=BCLM;
    BCLM=ECLM;
    ECLM=TEMP;
    DROP TEMP;
    END;
  IF ADT<=0 THEN DELETE;
  IF MP < 0 THEN DELETE;

00400099
00410099
00420099
00430099
00440099
00450099
00460099
00470099
00480099
00490099
00500099
00510099
00520099
00530099
00540099
00550099
00560099
00570099
00580099
00590099
00600099
00610099
00620099
00630099
00640099
00650099
00660099
00670099
00680099
00690099
00700099
00710099
00720099
00010099
00020099
00030099
00040099
00050099
00060099
00070099
00250099
00260099
IF INT ^=0 AND INT ^=1 THEN DELETE;
IF CONTROL<0 OR SECTION <0 THEN DELETE;

*CLASSIFIES INTERSECTIONS;
IF INT=1 THEN LTYPE='INTERSECTION';

*CLASSIFIES SECTIONS;
ELSE IF INT=0 THEN LTYPE='SECTION';

IF LTYPE='INTERSECTION' THEN LENGTH=1;

*ELIMINATES LOCAL AND ARTERIAL ROADS;
IF HTYPE ^=1 AND HTYPE ^=2 AND HTYPE ^=3 THEN DELETE;

LABEL SECTION= SECTION NUMBER
BCLM=BEGINNING OF SECTION (LOG MILE)
ECLM=END OF SECTION (LOG MILE)
HNUM=HIGHWAY NUMBER
MP=MILEPOST
LTYPE=LOCATION TYPE
SURCON=SURFACE CONDITION
HCLASS=HIGHWAY CLASS;

OUTPUT;

*SORTS THE ACCIDENT DATA AS GIVEN IN BY STATEMENT;

PROC SORT ;
BY LTYPE HCLASS HNUM MP ;

/* PROC FORMAT;
VALUE $SURCON
(MAX=100 MIN=0)
'A'='DRY CONDITION'
'B'='RAINY CONDITION'
'C'='MUDDY CONDITION'
'D'='SNOWY/ICY CONDITION'
'E'='OTHER CONDITIONS'
OTHER='MISSING VALUES' ;

PROC FORMAT;
VALUE HCL
(MAX=100 MIN=0)
1='RURL 2 LANE'
2='RURL OTHER'
3='RURL MULT LN DIV'
4='RURL INTERSTATE'
5='URBN 2 LANE'
6='URBN OTHER'
7='URBN MULT LN DIV'
8='URBN INTERSTATE'
OTHER='MISSING VALUES';
*/

DATA MASACC1;
SET MASACC;

*CHECKS FOR WET CRITERION;
IF SURCON='A' THEN DELETE;

* CHECKS FOR MISSING AND UNKNOWN SURFACE CONDITIONS;
IF SURCON='E' OR SURCON=' ' THEN DO;

DATA;
IF RDCON = 'J' AND RDCON -= 'K' THEN DELETE;
IF WEATH = 'A' OR WEATH = 'F' OR WEATH = 'G' THEN DELETE;
END;

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;
INFILE OUT;
INPUT CONTROL SECTION TRIANGLE MUEBAL MUEBPL VREBAL VREBL PL PPMNL
APWNL WTA WTP;

PROC SORT; BY CONTROL SECTION;

DATA RAT; MERGE MASACC3 TRI; BY CONTROL SECTION;
IF MUEBAL=. THEN DELETE;
   IF YEAR=. AND ADT=. THEN DELETE;
   IF PAVTYP=50
   OR PAVTYP=60 THEN DO;
   MUEBAL; VR=VREBAL; WHN=APWNL; PR=WTA;
   END;
ELSE DO;
   MUEBPL; VR=VREBL; WHN=PPMNL; PR=WTP;
   END;

DROP TRIANGLE MUEBAL MUEBPL VREBAL VREBL PL WTA WTP PPMNL APWNL;

PROC SORT; BY LTYPE HCLASS HNUM MP;

DATA RATE; SET RAT;
BY LTYPE HCLASS HNUM MP;

*CALCULATES THE ACCIDENTS PER MILLION VEHICLE MILES;

*EXPOSURE IS DEPENDENT ON PROPORTION WET TIME AND LENGTH;
ADT=ADT*LENGTH*WHN;
MVN=365*ADT/1000000;
ACC_MVN=TOT_ACC/MVN;
ACC_MIL=TOT_ACC/LENGTH;
VOL=ADT*365;
SQRVOL=VOL**2;
TRU_RAT=TOT_ACC/(MU*LENGTH*ADT*365);
NIBYVI=TOT_ACC/VOL;
NIBYSQVI=TOT_ACC/SQRT(VOL);
ONEBYVOL=1/VOL;

LABEL TOT_ACC=ACCIDENTS PER INTERSECTION
       ACC_MIL=ACCIDENTS PER MILE
       ACC_HVM=ACCIDENTS PER MILLION VEHICLE MILES;

OUTPUT;

PROC MEANS NOPRINT SUM MEAN STD VAR RANGE N;
VAR TOT_ACC ACC_MVM ADT NIBYVI NIBYSQVI ONEBYVOL TRU_RAT;
BY LTYPE HCLASS;
*FORMAT HCLASS HCL.;
OUTPUT OUT=STATAV SUM=STATSUM HVHSUM ADTSUM NVSUM
       MEAN=STATOT STANHM STAADT STANV STASQNV HARVOL MNTR
       STD=STATSTD HVHSTD ADTSSTD NVSTD NVTNSD ONETSTD TRSTD
       VAR=TOTVAR HVHVVAR ADTVAR NVVAR NVTNVAR
       RANGE=TOTRANGE MVRENG ADTRNG NVRNG NSVRNG
       N=TN MN AN NVN NSN;

PROC SORT DATA=STATAV;
BY LTYPE HCLASS;

DATA COMBINE;
MERGE RATE STATAV;
BY LTYPE HCLASS;

* FLAGGING BY C2;

SIG_NBYV=STATSUM/(ADTSUM*365);
C2=SIG_NBYV+(1.645 * SQRT(SIG_NBYV/VOL)) + (1/(2*VOL));

*FLAGGING BY BAYESIAN CRITERION 2;

BETA=STANV/NVVAR;
ALPHA=BETA*STANV;

BETAI=BETA+VOL;
ALPHAI=ALPHA + TOT_ACC;

ADJ_STA1= STANV*BETAI;
ADJ_STA2= SIG_NBYV*BETAI;

B2= 1-PROBGAM(ADJ_STA2,ALPHAI);

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

DATA OUT2.SNSPT88;
SET COMBINE ;
//
4.2 MENU PROGRAM LISTING
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.
SEQQ    : 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.
SECR1-7: Follows SECPR, making a choice for a particular parish.
SECPRL : Follows SECR1-7, 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 INTER CLSST
SPOTDT INTDT CLSST
SPOTDTL INTDTL CLSDT
SPOTPR INTPR CLSDTL
SPOTPR1-7 INTPR1-7 CLSPR
      CLSPR1-7
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;
call endblock();
return;
term:
return;
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(1,1,19,60,'cmdline');
        choice=block('print','print tables',22,
                      'accident','driver','vehicle',',',
                      'section',',','skid',',',
                      ',','end',');
    select(choice);
    when(1) do;
    call display('ep12.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('ep12.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('ep12.program');
    submit continue ;
        proc printto unit=20 new;
        proc print data=newlib.vhc188;
        run;
        proc printto;run;
        filename ft20f001 clear;
    endsubmit;
    end;
    when(5) do;
    call display('ep12.program');
    submit continue ;
        proc printto unit=20 new;
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;
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

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', 'newlib1ltrc.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;
call endblock();
return;
term:
call endlegend();
*call poplegend();
return;
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- FF10 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- FF10 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- FF10 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.vhc188','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;
call endblock();
return;
term:
call endlegend();
*call poplegend();
return;
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('rean1.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;
***** SOURCE *****

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

init:
return;
main:
choice=1;
loop:
do while (choice ne 0);
   choice=block('reports', 'reports section', 9,
                 'dotd', 'litr', '; ', ' ',
                 '; ', '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;
init:
return;
main:
choice=1;
loop:
do while (choice ne 0);
   choice=block('reports', 'dotd reports', '9',
                 'sections', 'spots', '9',
                 'intersections', 'clusters', '9',
                 'help', '9',
                 'end');
select(choice);
when(1) do;
submit continue sql;
create view newlib.r1 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.r1 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.r1 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.r1 as select parish,district,hnum,htype,hclass,control,
   section,bclm,adt,ltype,b2  from newlib.snc1s88
   where ltype='4'order by b2;
endsubmit;
call display('clust.program');
end;
when (9) call display('hex321.help');
when(12) leave loop;
otherwise
   do;
     if (choice<0) then call display('help.program', choice);
   end;
 end;
call endblock();
return;
term:
return;
ENTRY: SEC.PROGRAM   SEC1.PROGRAM   Last updated: 04/06/92

***** SOURCE *****

init:
return;
main:
choice=1;
loop:
dowhile(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;
call endblock();
return;
term:
return;
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('secdtl.program');
end;
when(2) do;
submit continue sql;
create view newlib.r2 as select * from newlib.r1
where district = 03;
endsubmit;
call display('secdtl.program');
end;
when(3) do;
submit continue sql;
create view newlib.r2 as select * from newlib.r1
where district = 04;
endsubmit;
call display('secdtl.program');
end;
when(4) do;
submit continue sql;
create view newlib.r2 as select * from newlib.r1
where district = 05;
endsubmit;
call display('secdtl.program');
end;
when(5) do;
submit continue sql;
create view newlib.r2 as select * from newlib.r1
where district = 07;
endsubmit;
call display('secdtl.program');
end;
when(6) do;
submit continue sql;
create view newlib.r2 as select * from newlib.r1
where district = 08;
endsubmit;
call display('secdtl.program');
end;
when(7) do;
submit continue sql;
create view newlib.r2 as select * from newlib.r1
where district = 58;
endsubmit;
call display('secdtl.program');
end;
when(8) do;
submit continue sql;
create view newlib.r2 as select * from newlib.r1
where district = 61;
endsubmit;
call display('secdtl.program');
end;
when(9) do;
submit continue sql;
create view newlib.r2 as select * from newlib.r1
where district = 62;
endsubmit;
call display('secdtl.program');
end;
when(12) leave loop;
otherwise 
do;
if (choice<0) then call display('help.program', choice);
end;
end;
call endblock();
return;
term:
return;
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,'cadline');
    call fsvview('newlib.r3');
    call display('fpcl.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,'cadline');
    call fsvview('newlib.r3');
    call display('fpcl.program');
    end;
when(3) do;