

TRAFFIC SAFETY ANALYSIS USING SAS

A. Reza Saremi, Charles C. Liu
AEPCO, Inc., McLean, Virginia

Abstract

This paper documents an example application of SAS where various highway database files are merged in order to analyze a specific traffic safety problem. The data base files are obtained from the Highway Safety Information System (HSIS) developed by the Federal Highway Administration.

HSIS is a roadway-based database system which provides quality data on a large number of accident variables, roadway locations, and traffic volumes for problem analyses. Data files in HSIS are collected from five states and maintained in SAS format. An application of HSIS in investigating side-swipe accidents is reported. Specifically, the objective is to identify if there are any age differences in the frequency of side-swipe accidents involving vehicles travelling in the same direction on non-intersections of highways.

The study procedures are presented along with the SAS programs used in the analysis. In particular, we illustrate the steps to identify locations based on roadway variables, and to attach the required accident and vehicle data to these locations. The findings from the side-swipe accident investigation is discussed. The paper concludes with a list of recommendations for future safety analysis efforts using HSIS.

Introduction

The Highway Safety Information System (HSIS) is a highway safety database developed by the Federal Highway Administration and the University of North Carolina Highway Safety Research Center (HSRC). HSIS is designed to provide a detailed system linking accident, roadway, and traffic data for problem analysis. HSIS is a location-based accident system and it combines high-quality accident, roadway, and traffic data in a computer-linkable format. The system uses raw data collected by a select group of states. Figure 1 shows the HSIS problem analysis process. The data files are all in SAS format so that the analyses can be done using SAS.

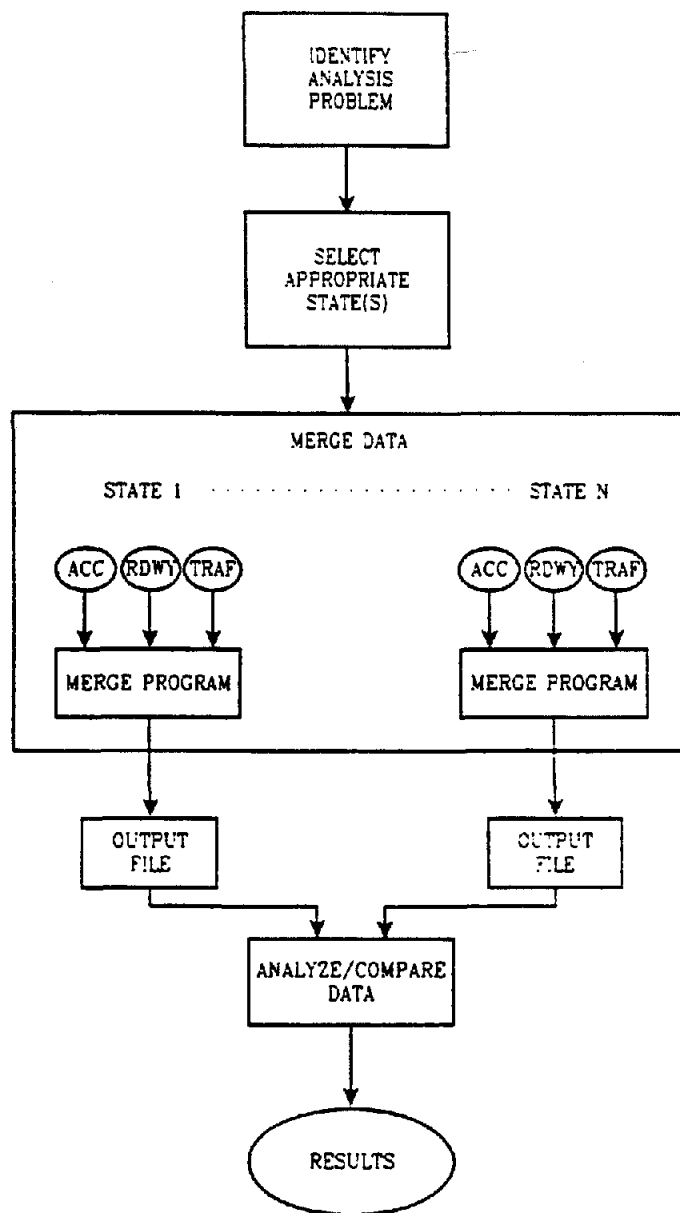


Figure 1- Flowchart of the HSIS problem analysis process

Problem

The problem discussed here is to identify if there are age differences in side-swipe accidents that occur on non-intersections of roads, same direction only. Four out of five existing HSIS states were used in the analysis based on the availability of the variables needed for the analysis.

HSIS Analysis Steps

Three HSIS data files, the accident file, the vehicle file, and the roadway data file were needed for the analysis. The first step was to identify which variables from each data file were needed for the analysis. In order to keep these selected variables the **KEEP** statement in SAS was used. By using a **KEEP** statement, the size of the data files will be reduced. This is important when working with large data bases like some of the HSIS data bases which can be as large as 200 mega bytes at times. The second step in the analysis was to add the necessary constraints to the SAS programs to obtain extract files. The SAS program in Appendix A describes the procedure to use for getting an extract file from the accident file for one of the states used in the analysis.

Once the extract files from each of the accident and vehicle files are obtained, these files will be merged together by using the **MERGE** statement in SAS. The program to use is included in Appendix B.

The final step of the analysis process is to put the accidents on the roadway file. The program that was used is included in Appendix C. In terms of the logic of this program, for the route numbers specified on the accident file and the roadway file, if they are the same then the mileposts are checked. The accident file has the variable **A_MILEPT** and the roadway file has the variables **BEGMP** and **ENDMP**. The logic of the program is first to check if the route numbers are the same in the accident and roadway files. Next, the milepost in the accident file is compared with the beginning and ending milepost of the roadway file. If the milepost in the accident file falls between the beginning and ending milepost in the roadway file, then there is a match and the accident record is put on the roadway file. As can be seen from the program (Appendix C), the use of **IF** then **ELSE** and the **LOOPS** and the **PUT** statement is illustrated in this program.

Findings

As it can be seen from Figure 2, for the side-swipe accidents on non-intersections of roads, same direction only, it was found that there were differences among the four age groups. For the four states chosen, and the four different age groups analyzed, it was found that the percentage of side-swipe accidents were less for elderly drivers (age over 65) for all the four states. The highest percentage of side-swipe accidents are for the age group 26 to 45, followed by the age group 16 to 25.

These results can be used to determine whether older drivers really have problems regarding side-swipe accidents. Of course, it also needs to be mentioned that older drivers do not drive as frequently as other drivers either.

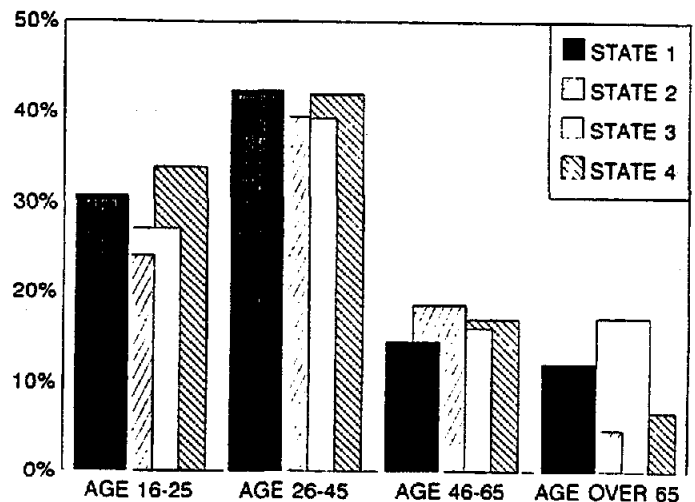


Figure 2. Percentage of side-swipe accidents by age groups

APPENDIX A

```
LIBNAME DD1 'E:\';
DATA ACC857;
```

In this part of the program, a temporary file(ACC857) is created where the constraints regarding non-intersections of highways and side-swipe accidents are selected.

```
=====
```

```
SET DD1.ACC857;
```

```
=====
```

Here the variable TYPE showing the type of accident is set to 1 initially.

```
=====
```

```

TYPE = 1;
IF OFF_TYPE = 0 THEN DELETE;
IF ACCDIGM = 2 OR ACCDIGM = 98 OR ACCDIGM = 99 THEN
DO;
  IF ACCTYPE = 1 THEN
  DO;
    IF INTERCH = ' ' THEN
    DO;
      IF INTERSE = 5 OR INTERSE = 8 OR INTERSE = 9 THEN
      DO;
        IF RDESIGN = 1 OR RDESIGN = 2 OR RDESIGN = 4 THEN
        DO;
          IF VEH_MOVM = 22 OR VIH_MOVN = 23 THEN
          DO;
            TYPE=2;
          END;
        END;
      END;
    END;
  END;
END;
END;
END;
END;
```

```
RUN;
```

NOTE: The data set WORK.ACC857 has 218158 observations and 36 variables.

NOTE: The DATA statement used 28.42 minutes.

```
=====
```

Here, by using the FORMAT statement, the two types of accidents, one meeting all the conditions for non-intersection, same side side-swipe accidents, and second accidents not meeting all the conditions are formatted.

```
=====
```

```
PROC FORMAT;
```

```
VALUE TYPEF 1 = 'NOT ALL MET'
            2 = 'ALL MET';
```

NOTE: Format TYPEF has been output.

```
RUN;
```

NOTE: The PROCEDURE FORMAT used 2.00 seconds.

```
PROC FREQ DATA=ACC857;
```

```
TABLES TYPE;
```

```
FORMAT TYPE TYPEF.;
```

```
RUN;
```

NOTE: The PROCEDURE FREQ used 3.90 minutes.

APPENDIX B

```
*****
* THIS PROGRAM IS WRITTEN TO MERGE THE ACCIDENT AND *
* VEHICLE FILES . *
***** ;
libname in 'd:\task9104\';
libname out 'd:\';
data out.all;
merge in.nmnve859(in=v) in.nmnac859;
by accnum;
if v;
run;
```

NOTE: The data set OUT.ALL has 796155 observations and 3 variables.

NOTE: The DATA statement used 43.17 minutes.

APPENDIX C

```
/* PROGRAM TO MERGE ACCIDENT WITH ROADWAY*/
```

```
DATA DD1.MRGACC;
LENGTH RTNUM A_ROUTE $6 ;
```

```
DROPAC=0;
RTLOW = 0; NOMATCH = 0; RTMATCH = 0;
```

```
LINK READLOG;
LINK READACC;
```

```
=====
Here in this part of the program in the LOOP, route numbers in the accident file and the roadway file are checked to see
if they are the same. By using LINK statement, the accident file and then the roadlog file are accessed .
=====
```

```
LOOP: IF RTNUM = 'AAAAAA' THEN STOP;
```

```
/* VERIFY STATE ROUTE NUMBERS ARE THE SAME*/
```

```
IF RTNUM > A_ROUTE THEN
DO;
NOACC + 1;
LINK READACC;
GO TO LOOP;
END;
```

```
IF RTNUM < A_ROUTE THEN
DO;
* NOLOG + 1;
LINK READLOG;
GO TO LOOP;
END;
```

```
=====
In this part of the program, the milepost in the accident file is checked against the beginning and ending mileposts in the
roadway file to check and see if the milepost falls between the beginning and ending mileposts of the roadway file.
=====
```

```
/* AT THIS POINT YOU HAVE A MATCH*/
```

```
/* COMPARE A_MILEPT ON ACC FILE WITH BEGMP AND ENDMP ON LOG*/
```

```
IF A_MILEPT >= BEGMP AND A_MILEPT < ENDMP THEN
DO;
RTMATCH + 1;
LINK GETYPE;
OUTPUT DD1.MRGACC;
LINK READACC;
GO TO LOOP;
END;
ELSE IF A_MILEPT < BEGMP THEN
DO;
NOACC + 1;
LINK READACC;
GO TO LOOP;
END;
```

```
ELSE IF A_MILEPT >= ENDMP THEN
DO;

LINK READLOG;
GO TO LOOP;
END;

RETURN;
```

=====

In this part of the program, the roadway file and the accident file are accessed. By using PUT statement, the values for some of the statistics to check for accuracy of the program are cessed.

=====

```
READLOG:
  IF EOFLOG THEN DO;
    NOLOG2 + 1;
    RTNUM = 'AAAAAA';
    PUT RDCOUNT = 'TOTAL ROADLOG RECORDS READ';
    PUT ACC_CNT = 'TOTAL ACCVEH RECORDS READ';
    PUT NOACC = 'NO ACC SECTION FOR ROUTE';
    *   PUT NOLOG = 'NO LOG FOR ACC SECTIO-NOLOG';
    *   PUT NOLOG2 = 'NO LOG FOR ACC SECTION-NOLOG-2';
    PUT RTMATCH= 'ACCS COUNTED';
    END;
  ELSE DO;
    SET ROADS88  END=EOFLOG;
    RDCOUNT + 1;
    END;
  RETURN;

READACC:
  IF EOFACC THEN DO;
    A_ROUTE = 'AAAAAA'
    END;
  ELSE DO;
    SET DD1.UTAV859  END=EOFACC;
    ACC_CNT+1;
  END;
  return;
```

=====

In this part of the program, the type of accidents, whether side-swipe accidents meeting all the conditions or not are output.

=====

```
GETYPE:
  IF COLLTYPE = '07' AND ACC_TYPE = '02'
  AND (LOCATN = '1' OR LOCATN = '4')
  THEN TYPE = 1;
  ELSE TYPE =2;
  RETURN;
  RUN;
```

```
RDCOUNT=2166 TOTAL ROADLOG RECORDS READ
ACC_CNT=291844 TOTAL ACCVEH RECORDS READ
NOACC=146777 NO ACC SECTION FOR ROUTE
RTMATCH=145066 ACCS COUNTED
```

NOTE: The data set DD1.MRGACC has 145066 observation and 22 variables.
 NOTE: The DATA statement used 14.68 minutes.