

SAS Methodology

Introduction

The following describes the methodology used to generate the data in this report. While this methodology is by necessity abbreviated, it should give a researcher reasonably familiar with SAS software enough information to duplicate the numbers herein. The section “Supplemental Information” should be consulted for additional technical information that is relevant to this methodology but would only serve to clutter the main presentation. Inclusion of the actual queries used to create the derived¹ variables allows a researcher to assess their effectiveness in summarizing the data elements in question. In certain instances the construction of a given query has a subjective nature since different researchers may interpret state codes differently. This effect is compounded if several primary variables are used to determine a specific derived variable. Every effort was made to determine logical and appropriate queries for the data elements. Reports from the individual states were consulted to determine the most appropriate query for a given derived variable. However, there is a wide variance in the particular information reported by each state, so a query used herein for a given state may not be entirely supported by the state’s published reports.

Conventions used in this appendix:

1. Actual SAS code is listed in SAS monospace font (size 8). Comments (SAS code that is not executed at run time) are contained within paired “/” and “*” symbols.²
2. When referencing primary variables (other than in SAS code), the variable description is in uppercase and the variable itself is given in parentheses in lowercase.
3. When referencing derived variables (other than in SAS code), the variables are in lowercase and italicized.
4. When referencing SAS reserved words (other than in SAS code), the reserved words are in uppercase and italicized.
5. When referencing pooled data set names or their subsets (other than in SAS code), the names are underlined.

¹ Derived variables are variables created from a given state’s primary variables, i.e., the variables contained in the NCSA SAS data sets and created from the state-supplied raw data files. Refer to the section “Creation of Derived Variables” later in this appendix.

² Note that SAS allows multiple statements (separated by semicolons) to be included on the same line of text. Also, a given SAS statement can extend over multiple lines of text.

Creation of Primary Data Sets

The program methodology pools 10 years of data (1990-1999)³ for a given state creating three pooled data sets:

- 1) Crash (apool)
- 2) Vehicle (vpool)
- 3) Person (ppool)

The variable *year* is created for each data set using SAS *IN=*⁴ variables. The following is simplified sample code for the Missouri Crash data set illustrating the method⁵.

```
/* creation of pooled crash data set */
data test.apool;6
  set missouri.mo90acc (in=y90) missouri.mo91acc (in=y91) missouri.mo92acc (in=y92)
    missouri.mo93acc (in=y93) missouri.mo94acc (in=y94) missouri.mo95acc (in=y95)
    missouri.mo96acc (in=y96) missouri.mo97acc (in=y97) missouri.mo98acc (in=y98)
    missouri.mo99acc;
  if y90 then year='1990';
  else if y91 then year='1991';
  else if y92 then year='1992';
  else if y93 then year='1993';
  else if y94 then year='1994';
  else if y95 then year='1995';
  else if y96 then year='1996';
  else if y97 then year='1997';
  else if y98 then year='1998';
  else year='1999';
run;
```

³ 9 years for Georgia (1990-1998), 8 years for North Carolina (1992-1999), and 7 years for Washington (1990-1996). Researchers interested in a specific period (e.g. 1995-1998) should build the data sets as appropriate. Each state-year can be analyzed individually if desired. For example, if just California 1999 is desired, a similar methodology to that described here is still appropriate, but the %assignyr and %uniqcase macros (see "Supplemental Information") are not needed.

⁴ SAS *IN=* variables are not included in the resultant data set. They essentially serve to identify the source of each observation in the pooled data set.

⁵ Statements creating the derived variables, e.g., rollover, have been omitted. The *LIBNAME* "test" is an arbitrary library for data sets.

⁶ When creating the Maryland Crash file, insert the following *LENGTH* statement between the *DATA* and *SET* statements: "length severity \$ 2;". Similarly, for the Maryland Person file, use "length inj pos sex pppo ped_loc \$ 2;". For the Michigan Person file, use "length pos rest1 \$ 2". The *LENGTH* statements are necessary since the length of these variables varies between years.

Creation of Merged Data Sets

Three merged data sets are created using the primary data sets: Crash-Vehicle ([avpool](#)), Crash-Person ([apool](#)), and Vehicle-Person ([vppool](#)). The following sample code for the Vehicle-Person data set merge illustrates the method⁷:

```
data test.vppool;  
  merge test.vpool test.ppool (in=A);  
  by caseno vehno;  
  if A;8  
run;
```

One additional merged data set [avp](#) is used in certain instances⁹. It is created by merging the Crash data set [apool](#) with the [vppool](#) data set:

```
data test.avp;  
  merge test.apool test.vppool;  
  by caseno;  
run;
```

Creation of Secondary Data Sets¹⁰

The three primary data sets and the four merged data sets described above are used to create various subsets of the data for the individual analyses. The diagrams contained on the following three pages detail the procedures used to subset the data. The diagrams are conceptual in nature.¹¹ Data sets are denoted by squares. The individual name(s) of the data set(s) are contained in parentheses within the squares. Subsetting *IF* conditions using the derived variables are listed along the arrows.¹² The following examples serve to illustrate how to create the data sets used in this report based on the diagrams.

⁷ Note that the data sets must be sorted on the *BY* variables prior to merging.

⁸ The line “if A” in conjunction with “(in=A)” characterizes a conditional merge. Observations from the Vehicle and Person data sets are merged only if an observation exists in the Person data set. In the example above, the statement serves to exclude any unoccupied vehicles from the Vehicle-Person data set.

⁹ Used primarily in alcohol and/or speed determination where the primary variables are present in the Crash, Vehicle, and Person data sets.

¹⁰ Creation of the secondary alcohol and speeding data sets are described in the next section.

¹¹ While individual SAS *DATA* steps can be used to create the data sets, appropriate use of *DO* loops and *OUTPUT* statements can reduce the processing time. Note that multiple data sets can be created in one *DATA* step. See “Supplementary Information”.

¹² Refer to the section “Creation of Derived Variables” later in this appendix for specific information on each derived variable.

In the first example below, the Crash-Person data set is used to create the Driver data set consisting only of vehicle drivers¹³ using the derived variable *p*type:

```
data test.driver;
  set test.apool;
  if ptype='DR';
run;
```

The next example creates the Passenger Car/Light Truck Single-Vehicle Rollover data set using the Crash-Vehicle data set and the derived variables *v*type, *c*type, and *rollover*.¹⁴

```
data test.roll;
  set test.avpool;
  if vtype in ('PC','ST') and ctype='S' and rollover=1;
run;
```

The next example creates the Child Restraint data set from the Vehicle-Person data set using the variable AGE (age) and the derived variables *p*type and *v*type:

```
data test.child;
  set test.vppool;15
  if ptype='PA' and vtype in ('PC','ST') and age<6;
run;
```

The final example creates the Motorcyclists Killed and Motorcyclists Injured data sets (*cyci1*, *cyci2*) from the Vehicle Occupants data set using the derived variables *v*type and *pinj*.

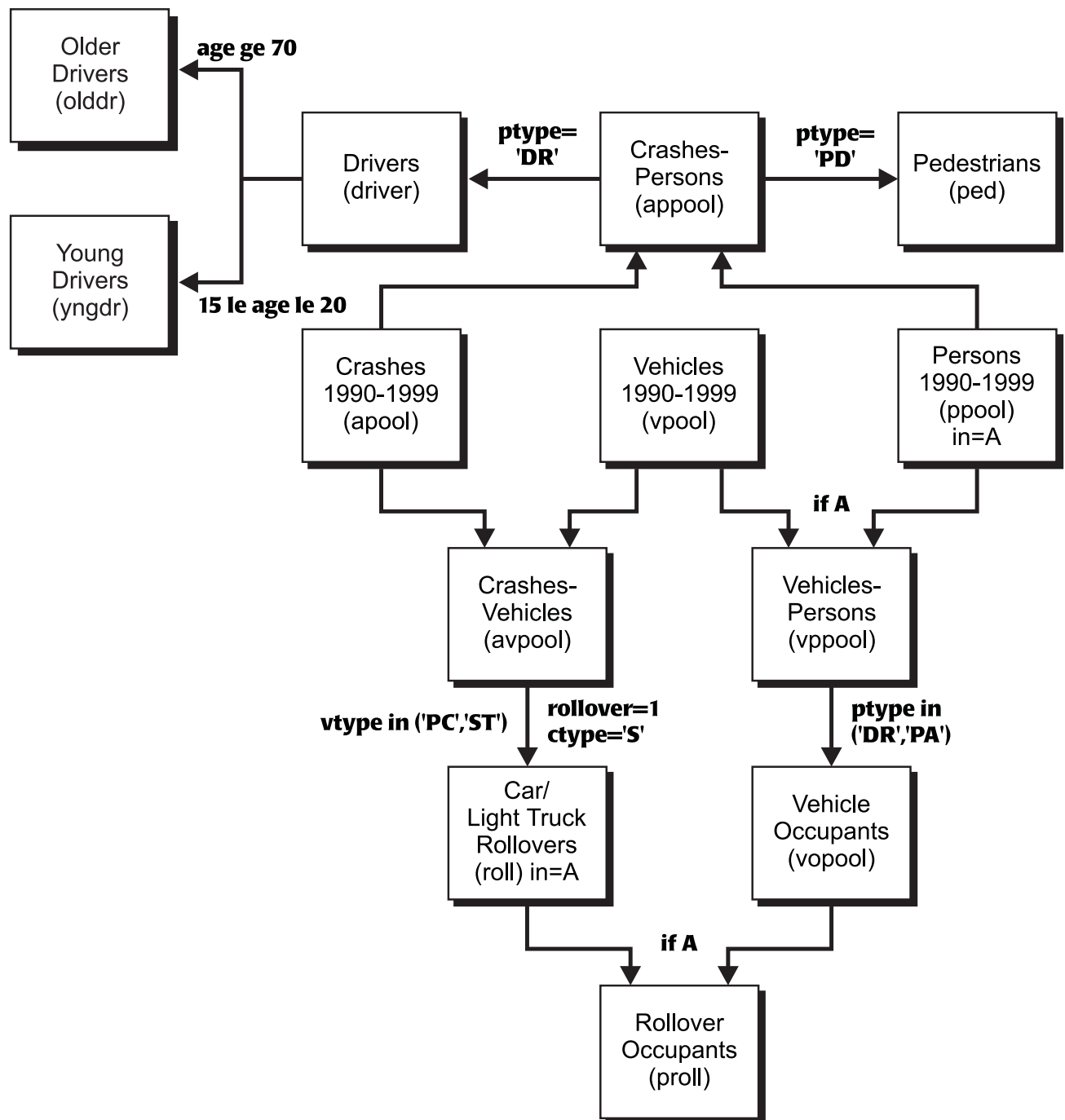
```
data test.cyci1 test.cyci2;
  set test.vopool;
  if vtype='MC' then do;
    if pinj=1 then output test.cyci1; /* killed motorcyclists */
    if pinj=2 then output test.cyci2; /* injured motorcyclists */
  end;
run;
```

¹³ Useful for determining driver involvement by crash severity.

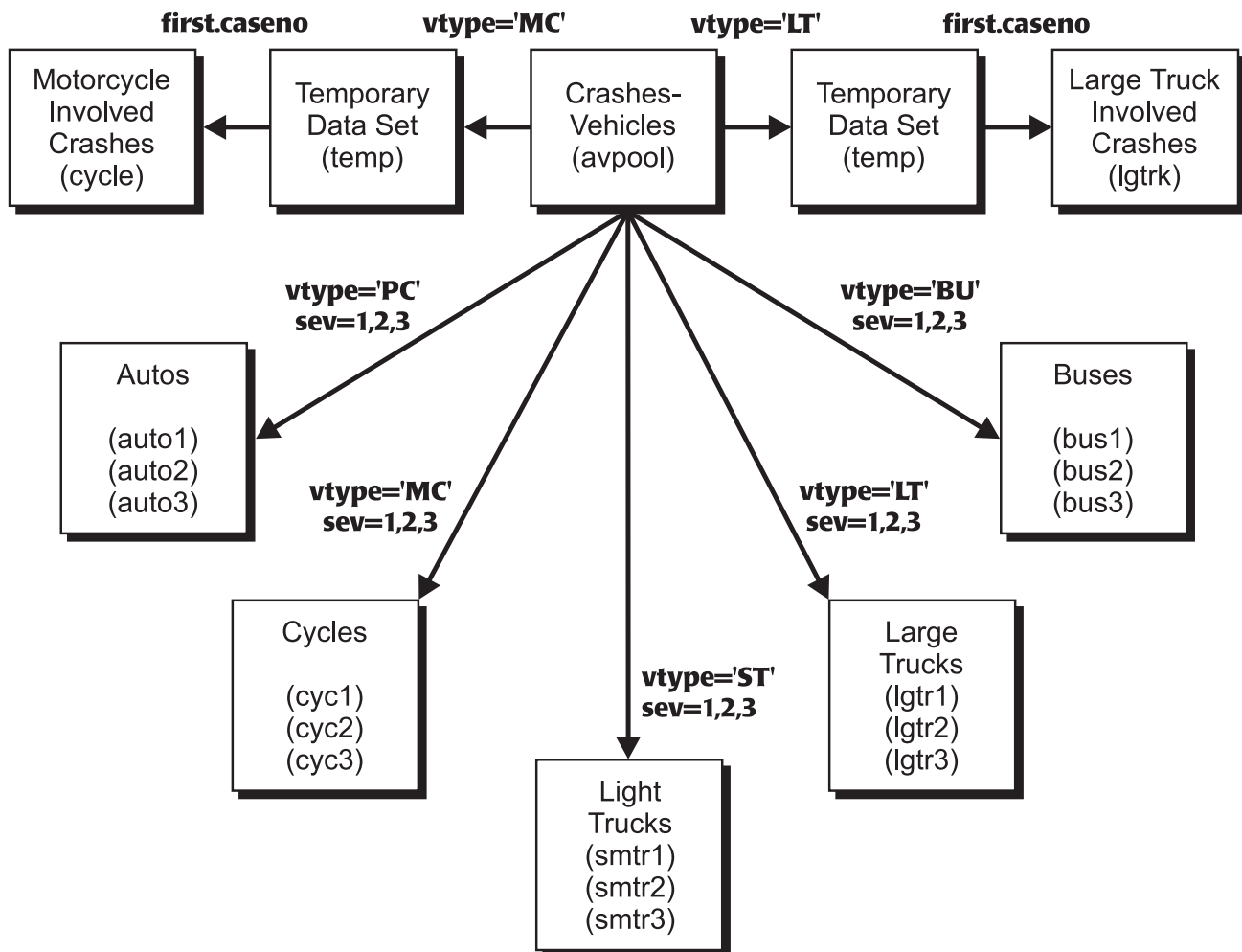
¹⁴ Refer to the rollover query in the section "Creation of Derived Variables" for an alternate method.

¹⁵ Either *vopool* or *vcpool* could also be used here.

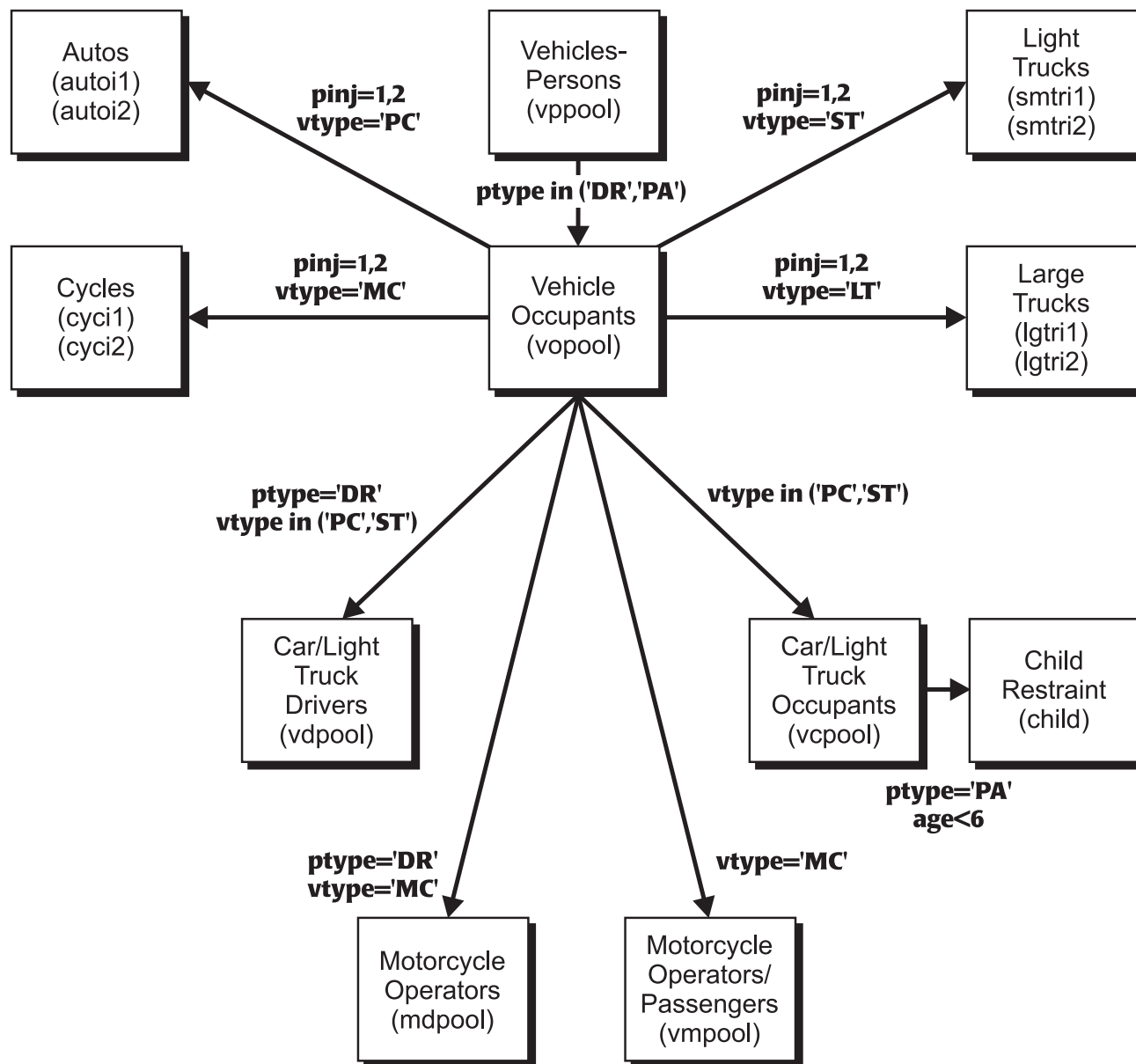
CRASH DATA SETS



VEHICLE DATA SETS



PERSON DATA SETS



Creation of Alcohol/Speeding Data Sets

Since alcohol may be contained in any or all of the primary data sets, it is generally necessary to use a merged data set for determining alcohol-involved crashes and persons. The example below assumes that alcohol information is contained in all primary data sets; hence, the data set `avp` is used in conjunction with the derived variables `alcohol1`, `alcohol2`, and `alcohol3`. A similar procedure can be used for speeding determinations. The diagrams contained on the next two pages graphically describe the alcohol procedure listed below and the analogous procedure for speeding determination. Note that the diagrams detail the most general case.

```
data temp (keep=caseno);16
  set test.avp;17
  by caseno;18
  retain alcohol 0;19
  if first.caseno then do;20
    alcohol=0;
    flag=0;
  end;
  if (alcohol1=1 or alcohol2=1 or alcohol3=1) then flag=1;21
  if alcohol<flag then alcohol=flag;
  if last.caseno and alcohol=1 then output;22
run;

/* creates alcohol-related crash data set */
data test.alc;
  merge temp (in=A) test.apool;
  by caseno;
  if A;
run;

/* creates alcohol-related person data set */
data test.alcper;
  merge test.alc (in=A) test.vppool;23
  by caseno;
  if A;
run;
```

¹⁶ Creates a temporary data set consisting only of case numbers involving alcohol.

¹⁷ The `avp` data set will always work here, although other data sets may be used in certain situations (e.g., if alcohol information is only contained in the Crash and Vehicle data sets, then `avpool` could be substituted here).

¹⁸ A SAS formalism needed with the `SET` statement when using `FIRST` and `LAST` variables. `FIRST` and `LAST` variables cannot be referenced unless the data set is sorted accordingly.

¹⁹ The `RETAIN` statement maintains a variable's value across iterations of the `DATA` step.

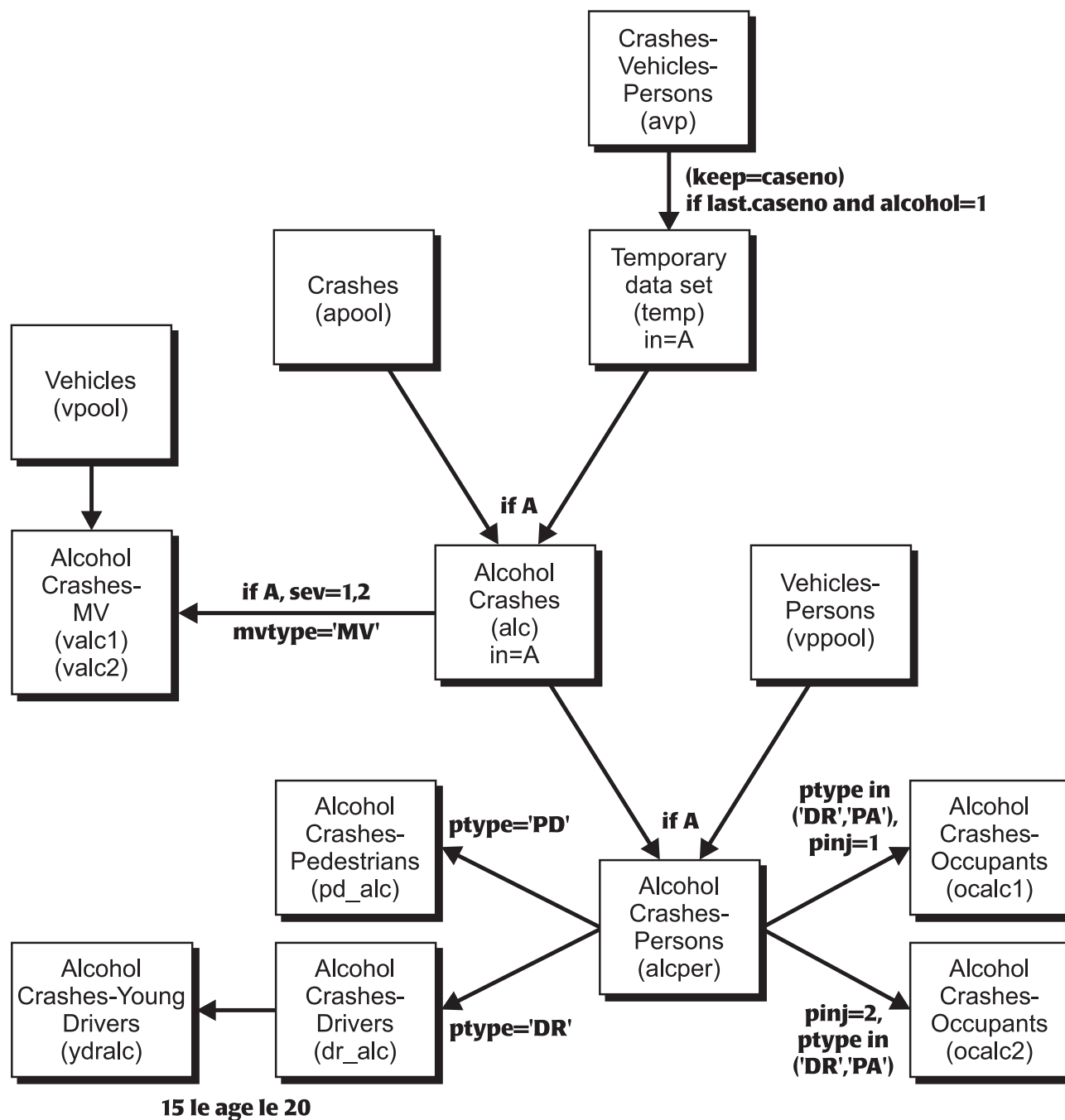
²⁰ Acts on the first occurrence of a given case number in the merged data set.

²¹ This statement should be modified depending on the number of derived alcohol variables present for a given state (refer to the alcohol queries). For example, the statement "if alcohol1=1 then flag=1;" is used for Kansas (the `ppool` data set can also be substituted for `avp`).

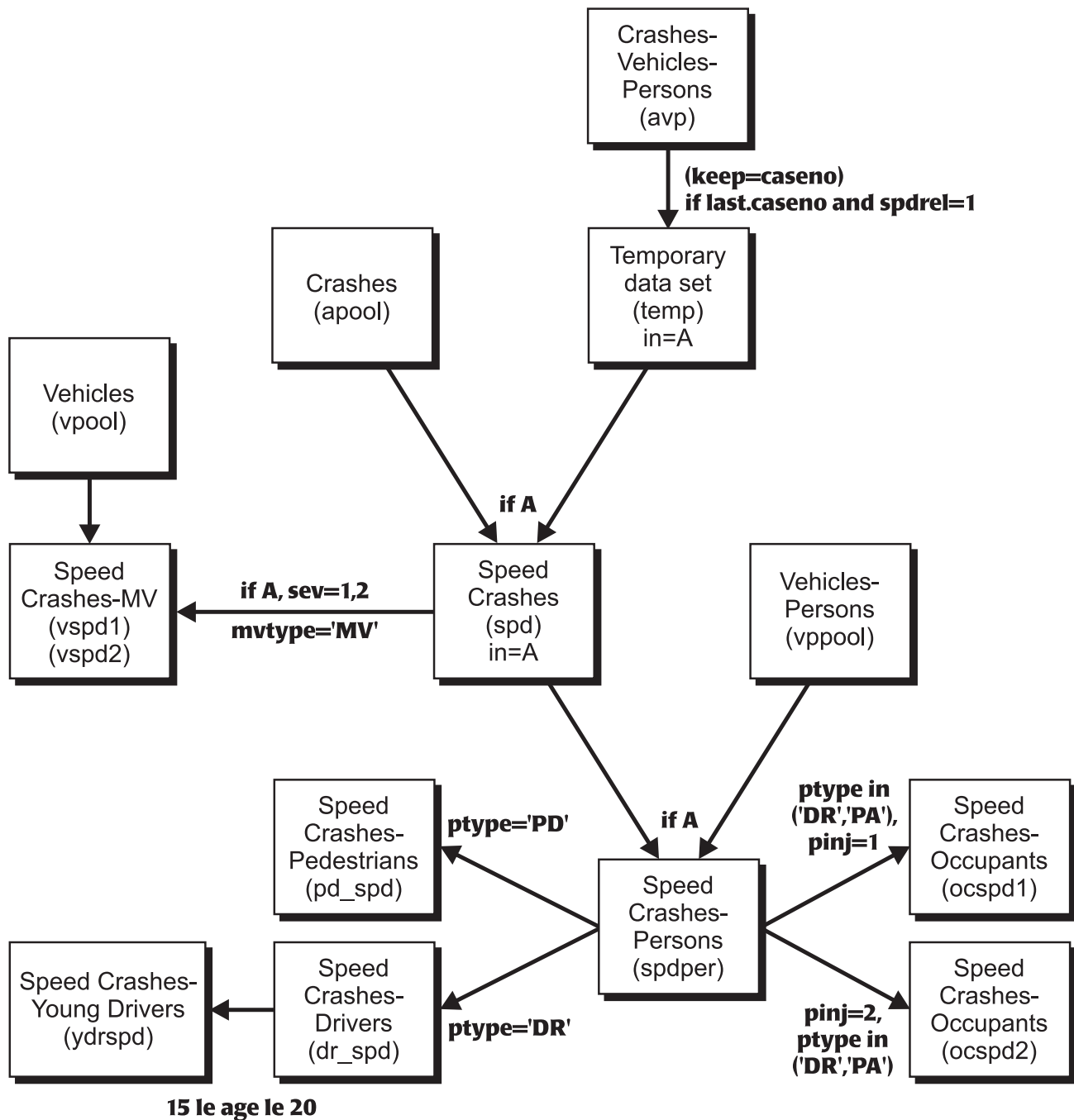
²² Acts on the last occurrence of a given case number in the merged data set. The `OUTPUT` statement controls the creation of the `temp` data set. Only case numbers for cases involving alcohol are included in the `temp` data set.

²³ The `ppool` data set may be substituted for the `vppool` data set in those cases where the Vehicle data set is not needed to determine the derived variable `ptype` (see "Creation of Derived Variables").

ALCOHOL DATA SETS



SPEED DATA SETS



Creation Of Derived Variables

VARIABLE CATEGORY: AGE

The variable *age1* is derived from the variable AGE (age) in the Person data set. The categories age1=2 through age1=10 are identical for all states. The categories age1=1 and age1=11 are state-specific. The category age1=12 designates unknown and missing ages.

STATE	QUERY
California	<pre> if (1 le age le 4) then age1=1; else if (5 le age le 9) then age1=2; else if (10 le age le 15) then age1=3; else if (16 le age le 20) then age1=4; else if (21 le age le 24) then age1=5; else if (25 le age le 34) then age1=6; else if (35 le age le 44) then age1=7; else if (45 le age le 54) then age1=8; else if (55 le age le 64) then age1=9; else if (65 le age le 74) then age1=10; else if (75 le age le 99) then age1=11; else age1=12; </pre>
Florida	<pre> if (1 le age le 4) then age1=1; else if (5 le age le 9) then age1=2; else if (10 le age le 15) then age1=3; else if (16 le age le 20) then age1=4; else if (21 le age le 24) then age1=5; else if (25 le age le 34) then age1=6; else if (35 le age le 44) then age1=7; else if (45 le age le 54) then age1=8; else if (55 le age le 64) then age1=9; else if (65 le age le 74) then age1=10; else if (75 le age le 99) then age1=11; else age1=12; </pre>
Georgia	<pre> if (0 le age le 4) then age1=1; else if (5 le age le 9) then age1=2; else if (10 le age le 15) then age1=3; else if (16 le age le 20) then age1=4; else if (21 le age le 24) then age1=5; else if (25 le age le 34) then age1=6; else if (35 le age le 44) then age1=7; else if (45 le age le 54) then age1=8; else if (55 le age le 64) then age1=9; else if (65 le age le 74) then age1=10; else if (75 le age le 98) then age1=11; else age1=12; </pre>

STATE	QUERY
Illinois	<pre> if year in ('1990','1991','1992','1993','1994','1995') then do; if (1 le age le 4) then age1=1; else if (5 le age le 9) then age1=2; else if (10 le age le 15) then age1=3; else if (16 le age le 20) then age1=4; else if (21 le age le 24) then age1=5; else if (25 le age le 34) then age1=6; else if (35 le age le 44) then age1=7; else if (45 le age le 54) then age1=8; else if (55 le age le 64) then age1=9; else if (65 le age le 74) then age1=10; else if (75 le age le 99) then age1=11; else age1=12; end; else do; /* 1996-1999 */ if (0 le age le 4) then age1=1; else if (5 le age le 9) then age1=2; else if (10 le age le 15) then age1=3; else if (16 le age le 20) then age1=4; else if (21 le age le 24) then age1=5; else if (25 le age le 34) then age1=6; else if (35 le age le 44) then age1=7; else if (45 le age le 54) then age1=8; else if (55 le age le 64) then age1=9; else if (65 le age le 74) then age1=10; else if (75 le age le 98) then age1=11; else age1=12; end; </pre>
Indiana	<pre> if (1 le age le 4) then age1=1; else if (5 le age le 9) then age1=2; else if (10 le age le 15) then age1=3; else if (16 le age le 20) then age1=4; else if (21 le age le 24) then age1=5; else if (25 le age le 34) then age1=6; else if (35 le age le 44) then age1=7; else if (45 le age le 54) then age1=8; else if (55 le age le 64) then age1=9; else if (65 le age le 74) then age1=10; else if (75 le age le 99) then age1=11; else age1=12; </pre>
Kansas	<pre> if (1 le age le 4) then age1=1; else if (5 le age le 9) then age1=2; else if (10 le age le 15) then age1=3; else if (16 le age le 20) then age1=4; else if (21 le age le 24) then age1=5; else if (25 le age le 34) then age1=6; else if (35 le age le 44) then age1=7; else if (45 le age le 54) then age1=8; else if (55 le age le 64) then age1=9; else if (65 le age le 74) then age1=10; else if (75 le age le 99) then age1=11; else age1=12; </pre>

STATE	QUERY
Maryland	<pre> if (0 le age le 4) then age1=1; else if (5 le age le 9) then age1=2; else if (10 le age le 15) then age1=3; else if (16 le age le 20) then age1=4; else if (21 le age le 24) then age1=5; else if (25 le age le 34) then age1=6; else if (35 le age le 44) then age1=7; else if (45 le age le 54) then age1=8; else if (55 le age le 64) then age1=9; else if (65 le age le 74) then age1=10; else if (75 le age le 99) then age1=11; else age1=12; </pre>
Michigan	<pre> if (0 le age le 4) then age1=1; else if (5 le age le 9) then age1=2; else if (10 le age le 15) then age1=3; else if (16 le age le 20) then age1=4; else if (21 le age le 24) then age1=5; else if (25 le age le 34) then age1=6; else if (35 le age le 44) then age1=7; else if (45 le age le 54) then age1=8; else if (55 le age le 64) then age1=9; else if (65 le age le 74) then age1=10; else if (75 le age le 98) then age1=11; else if year not in ('1990','1991') and age=99 then age1=11; else age1=12; </pre>
Missouri	<pre> if (0 le age le 4) then age1=1; else if (5 le age le 9) then age1=2; else if (10 le age le 15) then age1=3; else if (16 le age le 20) then age1=4; else if (21 le age le 24) then age1=5; else if (25 le age le 34) then age1=6; else if (35 le age le 44) then age1=7; else if (45 le age le 54) then age1=8; else if (55 le age le 64) then age1=9; else if (65 le age le 74) then age1=10; else if (75 le age le 98) then age1=11; else age1=12; </pre>
New Mexico	<pre> if (1 le age le 4) then age1=1; else if (5 le age le 9) then age1=2; else if (10 le age le 15) then age1=3; else if (16 le age le 20) then age1=4; else if (21 le age le 24) then age1=5; else if (25 le age le 34) then age1=6; else if (35 le age le 44) then age1=7; else if (45 le age le 54) then age1=8; else if (55 le age le 64) then age1=9; else if (65 le age le 74) then age1=10; else if (75 le age le 99) then age1=11; else age1=12; </pre>

STATE	QUERY
North Carolina	<pre> if (0 le age le 4) then age1=1; else if (5 le age le 9) then age1=2; else if (10 le age le 15) then age1=3; else if (16 le age le 20) then age1=4; else if (21 le age le 24) then age1=5; else if (25 le age le 34) then age1=6; else if (35 le age le 44) then age1=7; else if (45 le age le 54) then age1=8; else if (55 le age le 64) then age1=9; else if (65 le age le 74) then age1=10; else if (75 le age le 98) then age1=11; else age1=12; </pre>
Ohio	<pre> if (1 le age le 4) then age1=1; else if (5 le age le 9) then age1=2; else if (10 le age le 15) then age1=3; else if (16 le age le 20) then age1=4; else if (21 le age le 24) then age1=5; else if (25 le age le 34) then age1=6; else if (35 le age le 44) then age1=7; else if (45 le age le 54) then age1=8; else if (55 le age le 64) then age1=9; else if (65 le age le 74) then age1=10; else if (75 le age le 99) then age1=11; else age1=12; </pre>
Pennsylvania	<pre> if (1 le age le 4) then age1=1; else if (5 le age le 9) then age1=2; else if (10 le age le 15) then age1=3; else if (16 le age le 20) then age1=4; else if (21 le age le 24) then age1=5; else if (25 le age le 34) then age1=6; else if (35 le age le 44) then age1=7; else if (45 le age le 54) then age1=8; else if (55 le age le 64) then age1=9; else if (65 le age le 74) then age1=10; else if (75 le age le 98) then age1=11; else age1=12; </pre>
Texas	<pre> if (0 le age le 4) then age1=1; else if (5 le age le 9) then age1=2; else if (10 le age le 15) then age1=3; else if (16 le age le 20) then age1=4; else if (21 le age le 24) then age1=5; else if (25 le age le 34) then age1=6; else if (35 le age le 44) then age1=7; else if (45 le age le 54) then age1=8; else if (55 le age le 64) then age1=9; else if (65 le age le 74) then age1=10; else if (75 le age le 98) then age1=11; else age1=12; </pre>

STATE	QUERY
Utah	<pre> if (1 le age le 4) then age1=1; else if (5 le age le 9) then age1=2; else if (10 le age le 15) then age1=3; else if (16 le age le 20) then age1=4; else if (21 le age le 24) then age1=5; else if (25 le age le 34) then age1=6; else if (35 le age le 44) then age1=7; else if (45 le age le 54) then age1=8; else if (55 le age le 64) then age1=9; else if (65 le age le 74) then age1=10; else if (75 le age le 99) then age1=11; else age1=12; </pre>
Virginia	<pre> if (1 le age le 4) then age1=1; else if (5 le age le 9) then age1=2; else if (10 le age le 15) then age1=3; else if (16 le age le 20) then age1=4; else if (21 le age le 24) then age1=5; else if (25 le age le 34) then age1=6; else if (35 le age le 44) then age1=7; else if (45 le age le 54) then age1=8; else if (55 le age le 64) then age1=9; else if (65 le age le 74) then age1=10; else if (75 le age le 99) then age1=11; else age1=12; </pre>
Washington	<pre> if (1 le age le 4) then age1=1; else if (5 le age le 9) then age1=2; else if (10 le age le 15) then age1=3; else if (16 le age le 20) then age1=4; else if (21 le age le 24) then age1=5; else if (25 le age le 34) then age1=6; else if (35 le age le 44) then age1=7; else if (45 le age le 54) then age1=8; else if (55 le age le 64) then age1=9; else if (65 le age le 74) then age1=10; else if (75 le age le 99) then age1=11; else age1=12; </pre>

VARIABLE CATEGORY: ALCOHOL

An alcohol-related crash is defined as a crash where the driver of a motor vehicle, or a pedestrian or pedalcyclist struck by a vehicle, had a positive blood alcohol content (BAC) level²⁴, or in which other indicators of alcohol consumption are present. It should be noted that there is a wide variety in the type and number of alcohol-related variables coded by each state. The determination of alcohol-related crashes generally require use of the Crash, Vehicle and/or Person data sets. The queries used and the data sets for which the queries apply are listed below. For example, California alcohol determinations require using the first query with the Crash data set (apool), the second query with the Vehicle data set (vpool), and the third query with the Person data set (ppool). The individual indicator variables *alcohol1*, *alcohol2*, and *alcohol3* are combined using the method described earlier in this appendix (see “Creation of Alcohol/Speeding Data Sets”). Note that the variable ALCOHOL TEST RESULTS (*tst_res1*), which gives the actual BAC level of the person involved, is not present in all state data sets; also, a given state may use certain numeric codes for other purposes, e.g., a ‘.99’ code might indicate an unknown BAC level.

STATE	QUERY
California	<pre> (1) if cause1 in ('01','23') then alcohol1=1; else alcohol1=0; /* apool */ (2) if ((year in ('1990','1991') and violatn='20') or (year in ('1992','1993','1994','1995','1996','1997','1998','1999') and confac1='20')) then alcohol2=1; else alcohol2=0; /* vpool */ (3) if alc_drug in ('2','3','4') then alcohol3=1; else alcohol3=0; /* ppool */ </pre>
Florida	<pre> (1) if alcdrug in ('1','3') then alcohol1=1; else alcohol1=0; /* apool */ (2) if alc_drug in ('2','4','5') or confac1 in ('07','09') or confac2 in ('07','09') or confac3 in ('07','09') or ('.01' le tst_res1 le '.99') then alcohol2=1; else alcohol2=0; /* ppool */ </pre>
Georgia	<pre> (1) if confac1='02' or confac2='02' or confac3='02' or confac4='02' then alcohol1=1; else alcohol1=0; /* vpool */ (2) if year in ('1990','1991','1992','1993') then do; if ('.01' le tst_res1 le '.99') or alc_drug='1' or percond1 in ('2','4') then alcohol2=1; else alcohol2=0; end; else if year in ('1994','1995','1996','1997') then do; if ('.01' le tst_res1 le '.99') or alc_drug='1' or percond1 in ('3','4') then alcohol2=1; else alcohol2=0; end; else do; /* 1998 */ if ('0.01' le tst_res1 le '0.99') or tst_res2='POS' or percond1 in ('3','4') then alcohol2=1; else alcohol2=0; end; /* ppool */ </pre>

²⁴ Alcohol information may also be coded for passengers in some states. The derived variable *ptype* and the alcohol queries can be used for this determination.

STATE	QUERY
Illinois	<pre> (1) if year in ('1990','1991','1992') then do; if tst_res1 in ('2','3','4','5','6','7') or percond1='9' then alcohol1=1; else alcohol1=0; end; else if year in ('1993','1994','1995') then do; if percond1='9' then alcohol1=1; else alcohol1=0; end; else do; /* 1996-1999 */ if ('.01' le tst_res1 le '.94') or percond1 in ('2','7') then alcohol1=1; else alcohol1=0; end; /* ppool */ </pre>
Indiana	<pre> (1) if cause1='01' then alcohol1=1; else alcohol1=0; /* apool */ (2) if confac1='01' or confac2='01' then alcohol2=1; else alcohol2=0; /* vpool */ (3) if percond1='2' or ('.010' le tst_res1 le '.999') or ('.010' le tst_res2 le '.999') then alcohol3=1; else alcohol3=0; /* ppool */ </pre>
Kansas	<pre> (1) if asa1 in ('AC','AP') or asa2 in ('AC','AP') or asa3 in ('AC','AP') or ('.01' le tst_res1 le '.99') or pcc1='02' or pcc2='02' or pcc3='02' or pcc4='02' or pcc5='02' or dcc1='02' or dcc2='02' or dcc3='02' or dcc4='02' or dcc5='02' or dcc6='02' or dcc7='02' or dcc8='02' or dcc9='02' or dcc10='02' then alcohol1=1; else alcohol1=0; /* ppool */ </pre>
Maryland	<pre> (1) if year in ('1990','1991','1992') and (cause1='21' or cause2='21') then alcohol1=1; else alcohol1=0; /* apool */ (2) if year in ('1990','1991','1992') and confac1='21' then alcohol2=1; else alcohol2=0; /* vpool */ (3) if year in ('1993','1994','1995','1996','1997','1998','1999') then do; if (tst_res1 ne '.88' and ('.01' le tst_res1 le '.98')) or alc_drug in ('11','21') or percond='02' or con_cir1 in ('02','04') or con_cir2 in ('02','04') or con_cir3 in ('02','04') or con_cir4 in ('02','04') then alcohol3=1; else alcohol3=0; end; else do; /* 1990-1992 */ if (tst_res1 ne '.88' and ('.01' le tst_res1 le '.98')) or percond1='2' or percond2='2' or percond3='2' or pedcond='2' then alcohol3=1; else alcohol3=0; end; /* ppool */ </pre>

STATE	QUERY
Michigan	<pre> (1) if year in ('1990','1991') then do; if cause1='07' or alcdrug='1' then alcohol1=1; else alcohol1=0; end; else do; /* 1992-1999 */ if alcdrug='1' then alcohol1=1; else alcohol1=0; end; /* apool */ (2) if year in ('1990','1991') then do; if alc_drug='1' or con_cir1='01' or tst_res1 in ('2','3') then alcohol2=1; else alcohol2=0; end; else if year='1992' then do; if ('.010' le tst_res1 le '.900') or alcohol='1' then alcohol2=1; else alcohol2=0; end; else do; /* 1993-1999 */ if alcohol='1' then alcohol2=1; else alcohol2=0; end; /* ppool */ </pre>
Missouri	<pre> (1) if ((year in ('1990','1991','1992') and (confac1='15' or confac2='15' or confac3='15' or confac4='15' or confac5='15')) or (year in ('1993','1994') and (confac1='16' or confac2='16' or confac3='16' or confac4='16' or confac5='16')) or (year in ('1995','1996','1997','1998','1999') and (confac1='18' or confac2='18' or confac3='18' or confac4='18' or confac5='18')))) then alcohol1=1; else alcohol1=0; /* vpool */ (2) if ((year in ('1990','1991','1992') and (con_cir1='15' or con_cir2='15' or con_cir3='15' or con_cir4='15')) or (year in ('1993','1994') and (con_cir1='16' or con_cir2='16' or con_cir3='16')) or (year in ('1995','1996','1997','1998','1999') and (con_cir1='18' or con_cir2='18' or con_cir3='18')))) then alcohol2=1; else alcohol2=0; /* ppool */ </pre>
New Mexico	<pre> (1) if substr(confac2,1,1)='4' or substr(confac2,2,1)='4' or substr(confac2,3,1)='4' then alcohol1=1; else alcohol1=0; /* vpool */ (2) if alc_drug in ('1','2','5') or tst_res1 ge '.01' then alcohol2=1; else alcohol2=0; /* ppool */ </pre>
North Carolina	<pre> (1) if alcohol=1 then alcohol1=1; else alcohol1=0; /* apool */ (2) if intoxc=1 or violat1=2 or violat2=2 or violat3=2 or violat4=2 or violat5=2 then alcohol2=1; else alcohol2=0; /* vpool */ </pre>

STATE	QUERY
Ohio	(1) if alcdrug='1' then alcohol1=1; else alcohol1=0; /* apool */ (2) if alcohol in ('2','3','4') or ('.01' le tst_res1 le '.98') then alcohol2=1; else alcohol2=0; /* ppool */
Pennsylvania	(1) if alcdrug in ('1','4') or (cause1 in ('02','74') and year in ('1990','1991','1992')) or (cause1 in ('A2','V1') and year in ('1993','1994','1995','1996','1997','1998','1999')) then alcohol1=1; else alcohol1=0; /* apool */ (2) if percond1='2' then alcohol2=1; else alcohol2=0; /* vpool */ (3) if alc_drug in ('1','4') or ('.01' le tst_res1 le '.96') then alcohol3=1; else alcohol3=0; /* ppool */
Texas	(1) if confac2='9' then alcohol1=1; else alcohol1=0; /* vpool */ (2) if ('01' le tst_res1 le '.97') or alc_drug='1' then alcohol2=1; else alcohol2=0; /* ppool */
Utah	(1) if violat in ('13','14') or confac1 in ('09','28') or confac2 in ('09','28') then alcohol1=1; else alcohol1=0; /* vpool */ (2) if ('.01' le tst_res1 le '.99') then alcohol2=1; else alcohol2=0; /* ppool */
Virginia	(1) if alcdrug='1' then alcohol1=1; else alcohol1=0; /* apool */ (2) if ('.01' le tst_res1 le '.99') or alc_drug in ('2','3','4','5') then alcohol2=1; else alcohol2=0; /* ppool */
Washington	(1) if confac1='01' or confac2='01' or confac3='01' then alcohol1=1; else alcohol1=0; /* vpool */ (2) if alc_drug in ('1','2','3','5','6') or ('.01' le tst_res1 le '.97') then alcohol2=1; else alcohol2=0; /* ppool */

VARIABLE CATEGORY: CHILD RESTRAINT

The variable *child* is normally derived using the variable RESTRAINT DEVICE (rest1) in the Person data set. Additional variables such as OCCUPANT RESTRAINT (belt) or RESTRAINT DEVICE 2 (rest2), contained in the Person data set, may also be used if applicable. For this analysis, the categories '(Child) Restraint Improperly Used' and '(Child) Restraint Failure' have been classified as 'Restraint Not Used'. Also, airbag codes without explicit reference to restraint use (used/not used) are classified as 'Restraint Use Unknown'. The variable *child* has the following codes:

- 1= Child restraint used
- 2=Other restraint used
- 3=No restraint used
- 4=Restraint use unknown.

STATE	QUERY
California	<pre> if rest1='Q' then child=1; else if rest1 in ('C','E','G','J') then child=2; else if rest1 in ('A','D','F','H','K','P','R','T','U') then child=3; else child=4; </pre>
Florida	<pre> if year in ('1990','1991','1992') then do; if rest1='3' then child=1; else if rest1='2' then child=2; else if rest1='1' then child=3; else child=4; end; else do; /* 1993-1999 */²⁵ if rest1='0' then do; if rest2='1' then child=3; else if rest2='2' then child=2; else if rest2='3' then child=1; else child=4; end; else if rest1='1' then do; if rest2 in ('2','3','5','6') then child=4; else child=3; end; else if rest1='2' then do; if rest2='3' then child=1; else if rest2 in ('1','5','6') then child=4; else child=2; end; else if rest1='3' then do; if rest2 in ('1','5','6') then child=4; else child=1; end; else if rest1='4' then do; if rest2='1' then child=3; else if rest2='2' then child=2; else if rest2='3' then child=1; else child=4; end; else if rest1 in ('5','6') then child=4; else do; /* rest1 not in '0','1','2','3','4','5','6' */ if rest2='1' then child=3; else if rest2='2' then child=2; else if rest2='3' then child=1; else child=4; end; end; </pre>

²⁵ This query sets contradictory information between Florida's two restraint variables to "Restraint use unknown" for the period 1993-1999.

STATE	QUERY
Georgia	<pre> if rest1='4' then child=1; else if rest1 in ('1','2') or (year in ('1994','1995','1996','1997','1998') and rest1='3') then child=2; else if rest1 in ('0','5') then child=3; else child=4; </pre>
Illinois	<pre> if year in ('1990','1991','1992','1993','1994','1995') then do; if rest1='5' then child=1; else if rest1='1' then child=2; else if rest1 in ('2','6','7') then child=3; else child=4; end; else do; /* 1996-1999 */ if rest1='6' then child=1; else if rest1='2' then child=2; else if rest1 in ('1','3','7','8') then child=3; else if rest1='9' then do; if airbag='1' then child=2; else if airbag='2' then child=3; else child=4; end; else child=4; end; end; </pre>
Indiana	<pre> if rest1='4' or rest2='4' then child=1; else if rest1 in ('2','3') or rest2 in ('2','3') then child=2; else if rest1='1' or rest2='1' then child=3; else child=4; </pre>
Kansas	<pre> if rest1='C' then child=1; else if rest1 in ('L','S','X') then child=2; else if rest1 in ('N','Y') then child=3; else child=4; </pre>
Maryland	<pre> if year in ('1990','1991','1992') then do; if rest1='04' then child=1; else if rest1 in ('01','02','03') then child=2; else if rest1 in ('10','11') then child=3; else child=4; end; else do; /* 1993-1999 */ if rest1='14' then child=1; else if rest1 in ('11','12','13','32') then child=2; else if rest1 in ('00','01','31') then child=3; else child=4; end; </pre>

STATE	QUERY
Michigan	<pre> if year in ('1990','1991') then do; if rest1='5' then child=1; else if rest1='2' then child=2; else if rest1 in ('1','3','6','7') then child=3; else child=4; end; else do; /* 1992-1999 */ if rest1='06' then child=1; else if rest1 in ('02','03','04') then child=2; else if rest1 in ('01','05','07','08') then child=3; else child=4; end; </pre>
Missouri	<pre> if year in ('1990','1991','1992') then do; if rest1='4' then child=1; else if rest1 in ('3','6') then child=2; else if rest1 in ('1','2') then child=3; else child=4; end; else do; /* 1993-1999 */ if rest1='6' then child=1; else if rest1 in ('3','4','5') then child=2; else if rest1 in ('1','2') then child=3; else child=4; end; </pre>
New Mexico	<pre> if year in ('1990','1991','1992','1993','1994','1995') then do; if rest1='9' then child=1; else if rest1 in ('3','5','6') then child=2; else if rest1 in ('1','2','4','7') then child=3; else child=4; end; else do; /* 1996-1999 */ if rest1='8' then child=1; else if rest1 in ('3','5','6') then child=2; else if rest1 in ('1','2','4','7') then child=3; else child=4; end; </pre>
North Carolina	<pre> if belt=4 then child=1; else if belt in (2,3) then child=2; else if belt=1 then child=3; else child=4; </pre>
Ohio	<pre> if rest1='6' then child=1; else if rest1 in ('3','4','5') then child=2; else if rest1 in ('1','2') then child=3; else child=4; </pre>

APPENDIX D

STATE	QUERY
Pennsylvania	<pre> if rest_use='1' then do; if rest1='4' then child=1; else if rest1 in ('1','2','3') or pas_rest='3' then child=2; else child=4; end; else if rest_use in ('0','2') then child=3; else child=4; </pre>
Texas	<pre> if rest1='3' then child=1; else if rest1 in ('1','2','5') then child=2; else if rest1='6' then child=3; else child=4; </pre>
Utah	<pre> if year in ('1990','1991','1992') then do; if rest1='5' then child=1; else if rest1 in ('1','2') then child=2; else if rest1 in ('3','4') then child=3; else child=4; end; else do; /* 1993-1999 */ if rest1='5' then child=1; else if rest1 in ('1','2','6','A') then child=2; else if rest1 in ('3','4','7') then child=3; else child=4; end; </pre>
Virginia	<pre> if rest1='5' then child=1; else if rest1 in ('2','3','4') then child=2; else if rest1='1' then child=3; else child=4; </pre>
Washington	<pre> if rest1='5' then child=1; else if rest1 in ('2','3','4','6','8') then child=2; else if rest1 in ('1','7','9') then child=3; else child=4; </pre>

VARIABLE CATEGORY: CRASH SEVERITY

The variable *sev* is derived using ACCIDENT SEVERITY (severity) in the Crash data set and uses the following codes:

- 1=Fatal crash
- 2=Injury crash
- 3=Property damage only (PDO) crash.

STATE	QUERY
California	<pre>select (severity); when ('1') sev=1; when ('2') sev=2; when ('3') sev=2; when ('4') sev=2; when ('0') sev=3; otherwise sev=.; end;</pre>
Florida	<pre>select (severity); when ('1') sev=3; when ('2') sev=2; when ('3') sev=2; when ('4') sev=2; when ('5') sev=1; otherwise sev=.; end;</pre>
Georgia	<pre>if num_fat>0 then sev=1; else if num_inj>0 then sev=2; else sev=3;</pre>
Illinois	<pre>select (severity); when ('1') sev=1; when ('2') sev=2; when ('3') sev=3; otherwise sev=.; end;</pre>
Indiana	<pre>select (severity); when ('1') sev=1; when ('2') sev=2; when ('3') sev=3; otherwise sev=.; end;</pre>
Kansas	<pre>select (severity); when ('4') sev=1; when ('1','2','3') sev=2; when ('0') sev=3; otherwise sev=.; end;</pre>
Maryland	<pre>if year in ('1990','1991','1992') then do; if severity='5' then sev=1; else if severity in ('4','3','2') then sev=2; else if severity='1' then sev=3; else sev=.; end; else do; /* 1993-1999 */ if severity='05' then sev=1; else if severity in ('04','03','02') then sev=2; else if severity='01' then sev=3; else sev=.; end;</pre>

APPENDIX D

STATE	QUERY
Michigan	<pre> if year in ('1990','1991') then sev=severity; else do; /* 1992-1999 */ if fatal='1' then sev=1; else if injury='1' then sev=2; else sev=3; end; </pre>
Missouri	<pre> select (severity); when ('1') sev=1; when ('2') sev=2; when ('3') sev=3; otherwise sev=.; end; </pre>
New Mexico	<pre> sev=severity; </pre>
North Carolina	<pre> select (severity); when (5) sev=1; when (4) sev=2; when (3) sev=2; when (2) sev=2; when (1) sev=3; otherwise sev=.; end; </pre>
Ohio	<pre> select (severity); when ('1') sev=1; when ('2') sev=2; when ('3') sev=3; otherwise sev=.; end; </pre>
Pennsylvania	<pre> select (severity); when ('1') sev=1; when ('2') sev=2; when ('3') sev=2; when ('4') sev=2; when ('5') sev=2; when ('6') sev=3; otherwise sev=.; end; </pre>
Texas	<pre> select (severity); when ('1') sev=2; when ('2') sev=2; when ('3') sev=2; when ('4') sev=1; when ('5') sev=3; otherwise sev=.; end; </pre>
Utah	<pre> select (severity); when ('1') sev=3; when ('2') sev=2; when ('3') sev=2; when ('4') sev=2; when ('5') sev=1; otherwise sev=.; end; </pre>
Virginia	<pre> select (severity); when ('0') sev=1; when ('1') sev=1; when ('2') sev=2; when ('3') sev=2; when ('4') sev=3; when ('5') sev=3; otherwise sev=.; end; </pre>
Washington	<pre> select (severity); when ('1') sev=3; when ('2') sev=2; when ('3') sev=1; otherwise sev=.; end; </pre>

VARIABLE CATEGORY: CRASH TYPE

The variable (*ctype*) is normally derived using the variable NUMBER OF VEHICLES (*num_veh*) in the Crash data set. For North Carolina, the variable NUMBER OF UNITS IN ACCIDENT (*numunit*) is used. For New Mexico and Washington, the NUMBER OF VEHICLES (*num_veh*) variable must be derived. The method is shown below. The variable *ctype* has the following codes:

S=Single vehicle
M=Multiple vehicle
U=Unknown.

All states excepting North Carolina:

```
if num_veh=1 then ctype='S'; else if num_veh>1 then ctype='M'; else ctype='U';
```

North Carolina:

```
if numunit=1 then ctype='S'; else if numunit>1 then ctype='M'; else ctype='U';
```

Derivation of the variable NUMBER OF VEHICLES (*num_veh*):
(New Mexico, Washington)

```
data temp (keep=caseno num_veh);
  set test.vpool;
  by caseno vehno;
  retain num_veh 0;
  if first.caseno then num_veh=0;
  num_veh+1;
  if last.caseno then output;
run;

data test.apool;26
  merge test.apool (in=A) temp;
  by caseno;
  if A;
run;
```

²⁶ This step adds the variable *num_veh* to the existing *apool* data set.

VARIABLE CATEGORY: DAY OF WEEK (Weekday vs Weekend)²⁷

The variable *wend1* is derived from the TIME OF ACCIDENT (time) variable and the DAY OF WEEK (weekday) variable, present in most state Crash data sets. For states where TIME OF ACCIDENT is not available, the variable HOUR OF ACCIDENT (hour) is used.

STATE	QUERY
California	<pre> /* mon=1 sun=7 */ if (weekday in ('6','7') or (weekday='1' and (0 le time le 559)) or (weekday='5' and (1800 le time le 2359))) then wend1='Weekend'; else if (weekday in ('2','3','4') or (weekday='1' and (600 le time le 2359)) or (weekday='5' and (0 le time le 1759))) then wend1='Weekday'; else wend1='Unknown'; </pre>
Florida	<pre> /* mon=1 sun=7 */ if (weekday in ('6','7') or (weekday='1' and (1 le time le 559)) or (weekday='5' and (1800 le time le 2400))) then wend1='Weekend'; else if (weekday in ('2','3','4') or (weekday='1' and (600 le time le 2359)) or (weekday='5' and (1 le time le 1759))) then wend1='Weekday'; else wend1='Unknown'; </pre>
Georgia	<pre> /* sun=1 sat=7 */ if (weekday in ('1','7') or (weekday='2' and (0 < time < 600)) or (weekday='6' and (1800 le time le 2400))) then wend1='Weekend'; else if (weekday in ('3','4','5') or (weekday='2' and (600 le time le 2400)) or (weekday='6' and (0 < time < 1800))) then wend1='Weekday'; else wend1='Unknown'; </pre>
Illinois	<pre> /* mon=1 sun=7 */ if year in ('1996','1997','1998','1999') then do; if (weekday in ('6','7') or (weekday='1' and (0 le time le 559)) or (weekday='5' and (1800 le time le 2359))) then wend1='Weekend'; else if (weekday in ('2','3','4') or (weekday='1' and (600 le time le 2359)) or (weekday='5' and (0 le time le 1759))) then wend1='Weekday'; else wend1='Unknown'; end; else do; /* 1990-1995 */ if (weekday in ('6','7') or (weekday='1' and (0 le hour le 5)) or (weekday='5' and (18 le hour le 23))) then wend1='Weekend'; else if (weekday in ('2','3','4') or (weekday='1' and (6 le hour le 23)) or (weekday='5' and (0 le hour le 17))) then wend1='Weekday'; else wend1='Unknown'; end; </pre>

²⁷ Some states designate midnight as time 0, while others use time 2400.

STATE	QUERY
Indiana	<pre> /* sun=1 sat=7 */ if (weekday in ('1','7') or (weekday='2' and (1 le time le 559)) or (weekday='6' and (1800 le time le 2400))) then wend1='Weekend'; else if (weekday in ('3','4','5') or (weekday='2' and (600 le time le 2400)) or (weekday='6' and (1 le time le 1759))) then wend1='Weekday'; else wend1='Unknown'; </pre>
Kansas	<pre> /* mon=1 sun=7 */ if (weekday in ('6','7') or (weekday='1' and ('0000' le time < '0600')) or (weekday='5' and ('1800' le time le '2359'))) then wend1='Weekend'; else if (weekday in ('2','3','4') or (weekday='1' and ('0600' le time le '2359')) or (weekday='5' and ('0000' le time < '0600'))) then wend1='Weekday'; else wend1='Unknown'; </pre>
Maryland	<pre> /* sun=1 sat=7 */ if (weekday in ('1','7') or (weekday='2' and (1 le time le 559)) or (weekday='6' and (1800 le time le 2400))) then wend1='Weekend'; else if (weekday in ('3','4','5') or (weekday='2' and (600 le time le 2400)) or (weekday='6' and (1 le time le 1759))) then wend1='Weekday'; else wend1='Unknown'; </pre>
Michigan	<pre> /* sun=1 sat=7 */ if year in ('1990','1991') then do; if (weekday in ('1','7') or (weekday='2' and (1 le hour le 6)) or (weekday='6' and (19 le hour le 24))) then wend1='Weekend'; else if (weekday in ('3','4','5') or (weekday='2' and (7 le hour le 24)) or (weekday='6' and (1 le hour le 18))) then wend1='Weekday'; else wend1='Unknown'; end; else do; /* 1992-1999 */ if (weekday in ('1','7') or (weekday='2' and (0 le hour le 5)) or (weekday='6' and (18 le hour le 24))) then wend1='Weekend'; else if (weekday in ('3','4','5') or (weekday='2' and (6 le hour le 24)) or (weekday='6' and (0 le hour le 17))) then wend1='Weekday'; else wend1='Unknown'; end; </pre>
Missouri	<pre> /* sun=1 sat=7 */ if (weekday in ('1','7') or (weekday='2' and (1 le time le 559)) or (weekday='6' and (1800 le time le 2400))) then wend1='Weekend'; else if (weekday in ('3','4','5') or (weekday='2' and (600 le time le 2400)) or (weekday='6' and (1 le time le 1759))) then wend1='Weekday'; else wend1='Unknown'; </pre>

APPENDIX D

STATE	QUERY
New Mexico	<pre> /* sun=1 sat=7 */ if (weekday in ('1','7') or (weekday='2' and (1 le time le 559)) or (weekday='6' and (1800 le time le 2400))) then wend1='Weekend'; else if (weekday in ('3','4','5') or (weekday='2' and (600 le time le 2400)) or (weekday='6' and (1 le time le 1759))) then wend1='Weekday'; else wend1='Unknown'; </pre>
North Carolina	<pre> /* mon=1 sun=7 */ if (weekday in ('6','7') or (weekday='1' and (0 le time le 559)) or (weekday='5' and (1800 le time le 2359))) then wend1='Weekend'; else if (weekday in ('2','3','4') or (weekday='1' and (600 le time le 2359)) or (weekday='5' and (0 le time le 1759))) then wend1='Weekday'; else wend1='Unknown'; </pre>
Ohio	<pre> /* sun=1 sat=7 */ if (weekday in ('1','7') or (weekday='2' and (0 le hour le 5)) or (weekday='6' and (18 le hour le 23))) then wend1='Weekend'; else if (weekday in ('3','4','5') or (weekday='2' and (6 le hour le 23)) or (weekday='6' and (0 le hour le 17))) then wend1='Weekday'; else wend1='Unknown'; </pre>
Pennsylvania	<pre> /* sun=1 sat=7 */ if (weekday in ('1','7') or (weekday='2' and (0 le time le 599²⁸)) or (weekday='6' and (1800 le time le 2399))) then wend1='Weekend'; else if (weekday in ('3','4','5') or (weekday='2' and (600 le time le 2399)) or (weekday='6' and (0 le time le 1799))) then wend1='Weekday'; else wend1='Unknown'; </pre>
Texas	<pre> /* sun=1 sat=7 */ if (weekday in ('1','7') or (weekday='2' and ('00' le hour le '05')) or (weekday='6' and ('18' le hour le '23'))) then wend1='Weekend'; else if (weekday in ('3','4','5') or (weekday='2' and ('06' le hour le '23')) or (weekday='6' and ('00' le hour le '17'))) then wend1='Weekday'; else wend1='Unknown'; </pre>

²⁸For Pennsylvania, times ending in 99 indicate that the hour is known but not minutes.

STATE	QUERY
Utah	<pre> /* mon=1 sun=7 */ if (weekday in ('6','7') or (weekday='1' and (1 le time le 559)) or (weekday='5' and (1800 le time le 2400))) then wend1='Weekend'; else if (weekday in ('2','3','4') or (weekday='1' and (600 le time le 2400)) or (weekday='5' and (1 le time le 1759))) then wend1='Weekday'; else wend1='Unknown'; </pre>
Virginia	<pre> /* mon=1 sun=7 */ if (weekday in ('6','7') or (weekday='1' and (0 le hour le 5)) or (weekday='5' and (18 le hour le 23))) then wend1='Weekend'; else if (weekday in ('2','3','4') or (weekday='1' and (6 le hour le 23)) or (weekday='5' and (0 le hour le 17))) then wend1='Weekday'; else wend1='Unknown'; </pre>
Washington	<pre> /* mon=1 sun=7 */ if (weekday in ('6','7') or (weekday='1' and (1 le time le 559)) or (weekday='5' and (1800 le time le 2400))) then wend1='Weekend'; else if (weekday in ('2','3','4') or (weekday='1' and (600 le time le 2400)) or (weekday='5' and (1 le time le 1759))) then wend1='Weekday'; else wend1='Unknown'; </pre>

VARIABLE CATEGORY: INJURY SEVERITY

The variable *pinj* is derived using the variable INJURY SEVERITY (inj) in the Person data set.

The variable *pinj* uses the following codes:

- 1=Fatality
- 2=Injury
- 3=Not injured/Unknown.

STATE	QUERY
California	<pre> if inj='1' then pinj=1; else if inj in ('2','3','4') then pinj=2; else pinj=3; </pre>
Florida	<pre> if inj='5' then pinj=1; else if inj in ('2','3','4') then pinj=2; else pinj=3; </pre>
Georgia	<pre> if inj='1' then pinj=1; else if inj in ('2','3','4') then pinj=2; else pinj=3; </pre>
Illinois	<pre> if inj='4' then pinj=1; else if inj in ('1','2','3') then pinj=2; else pinj=3; </pre>
Indiana	<pre> if inj_stat='6' then pinj=1; else if inj_stat in ('1','2','3','4','5','7','U') then pinj=2; else pinj=3; </pre>
Kansas	<pre> if inj='F' then pinj=1; else if inj in ('D','I','P') then pinj=2; else pinj=3; </pre>
Maryland	<pre> if year in ('1990','1991','1992') then do; if inj='5' then pinj=1; else if inj in ('4','3','2') then pinj=2; else pinj=3; end; else do; /* 1993-1999 */ if inj='05' then pinj=1; else if inj in ('04','03','02') then pinj=2; else pinj=3; end; </pre>

STATE	QUERY
Michigan	if inj='1' then pinj=1; else if inj in ('2','3','4') then pinj=2; else pinj=3;
Missouri	if inj='1' then pinj=1; else if inj in ('2','3','4') then pinj=2; else pinj=3;
New Mexico	if inj='K' then pinj=1; else if inj in ('A','B','C') then pinj=2; else pinj=3;
North Carolina	if inj=5 then pinj=1; else if inj in (2,3,4) then pinj=2; else pinj=3;
Ohio	if inj='1' then pinj=1; else if inj in ('2','3','4') then pinj=2; else pinj=3;
Pennsylvania	if inj='1' then pinj=1; else if inj in ('2','3','4','9') then pinj=2; else pinj=3;
Texas	if inj='4' then pinj=1; else if inj in ('1','2','3') then pinj=2; else pinj=3;
Utah	if inj='5' then pinj=1; else if inj in ('2','3','4') then pinj=2; else pinj=3;
Virginia	if inj in ('1','5') then pinj=1; else if inj in ('2','3','4') then pinj=2; else pinj=3;
Washington	if inj in ('2','3','4') then pinj=1; else if inj in ('5','6','7') then pinj=2; else pinj=3;

VARIABLE CATEGORY: PEDESTRIAN LOCATION

The variable *itype* is derived using variables in the Crash or Person data sets; see the individual queries for the variable names and locations. For this analysis, the classification ‘In Roadway’ includes shoulders. Driveway, sidewalk, median, and pedestrian island codes are classified as ‘Not in Roadway’, as well as explicit ‘Not in Roadway’ codes. The following descriptions, when an explicit ‘Not in Roadway’ code is also available for use, are assumed ‘In Roadway-Location Unknown’: ‘Playing’²⁹, ‘Standing’, ‘Pushing/Working on Vehicle’, ‘Coming From Behind Parked Vehicle’ and ‘Walking With/Against Traffic’. ‘Approaching/Entering/Leaving (School) Bus/Vehicle’ codes are implicitly considered ‘In Roadway-Location Unknown’. ‘Crossing With/Against Signal’ codes are implicitly considered ‘In Roadway-Intersection’. The handling of situations not noted above is specified in the individual queries. Finally, it is assumed that the researcher is using this query in conjunction with ptype=‘PD’ (see Person Type). The variable *itype* uses the following codes:

- RI=In roadway, intersection
- RN=In roadway, not in intersection
- RU=In roadway, location unknown
- NR=Not in roadway
- UK=Unknown.

STATE	QUERY
California	<pre>/* ped_actn (Pedestrian Action) in Crash data set */ if ped_actn='2' then itype='RI'; else if ped_actn='3' then itype='RN'; else if ped_actn in ('4','5','7') then itype='RU'; else if ped_actn='6' then itype='NR'; else itype='UK';</pre>
Florida	<pre>/* ped_act (Pedestrian Action) in Person data set */ /* codes 3&4 (90-92), 4&5 (93-99) are assumed shoulders, not sidewalks */ if year in ('1990','1991','1992') then do; if ped_act='02' then itype='RI'; else if ped_act='01' then itype='RN'; else if ped_act in ('03','04','05','06','07') then itype='RU'; else itype='UK'; end; else do; /* 1993-1999 */ if ped_act='03' then itype='RI'; else if ped_act in ('01','02') then itype='RN'; else if ped_act in ('04','05','06','07','08') then itype='RU'; else if ped_act='09' then itype='NR';³⁰ else itype='UK'; end;</pre>

²⁹ In this situation, the code description is ‘Playing’ rather than ‘Playing in Roadway’, etc.
³⁰ Florida has no explicit ‘Not in Roadway’ classification. For 1993-1999, code ‘09’=‘Pedestrian Island’.

STATE	QUERY
Georgia	<pre> /* ped_act (Pedestrian Maneuver) in Person data set */ if ped_act='02' then itype='RI'; else if ped_act='01' then itype='RN'; else if ped_act in ('03','04','05','06','07','08') then itype='RU'; else if ped_act='09' then itype='NR'; else itype='UK'; </pre>
Illinois	<pre> /* ped_act (Pedestrian Action), ped_act2 (Pedestrian Action 2), and ped_loc (Pedestrian Location) in Person data set */ if year in ('1990','1991','1992','1993','1994','1995') then do; if ped_act='10' then itype='NR'; else if ped_act in ('01','02','03') then itype='RI'; else if ped_act='04' then itype='RN'; else if ped_act in ('05','06','07','08','09','11','12') then itype='RU'; else itype='UK'; end; else do; /* 1996-1999 */ if ped_loc in ('05','06') or ped_act2='59' then itype='NR';³¹ else if ped_act2 in ('51','52') or ped_act in ('01','02','03') then itype='RI'; else if ped_act='04' then itype='RN'; else if ped_act2 in ('53','54','55','56','57','58','60','61','62','63') or ped_act in ('05','06','07','08','09','11','12') or ped_loc in ('01','02','03','04') then itype='RU'; else itype='UK'; end; </pre>
Indiana	<pre> /* ped_act (Pedestrian Action) in Person data set */ if ped_act='11' then itype='RI'; else if ped_act='10' then itype='RN'; else if ped_act in ('02','03','04','05','06','07','08','09') then itype='RU'; else if ped_act='01' then itype='NR'; else itype='UK'; </pre>
Kansas	<pre> /* ped_loc (Pedestrian Location) in Person data set */ if ped_loc in ('01','02','03') then itype='RI'; else if ped_loc in ('11','12','13') then itype='RN'; else if ped_loc='25' then itype='NR'; else if ped_act in ('01','02','03','04','05','06') then itype='RU'; else itype='UK'; </pre>

³¹ The Illinois variable PEDESTRIAN ACTION (ped_act) code 10='Not in Roadway' is unreliable after 1993. Data for 1994-1995 (and some late-1993 data) were recoded from a new PAR into the old format. The conversion was only approximate. PEDESTRIAN LOCATION (ped_loc) is used for 'Not in Roadway' in 1996-1999. However, for 1994-1995, ped_act had to be used for 'Not in Roadway'.

STATE	QUERY
Maryland	<pre> /* ped_act (Pedestrian Action) and ped_loc (Pedestrian Location) in Person data set */ if year in ('1990','1991','1992') then do; if ped_loc in ('2','3') then itype='NR'; else if ped_act='01' then itype='RI'; else if ped_act='02' then itype='RN'; else if ped_act in ('03','04','05','06','07','08','09','10','11') or ped_loc in ('1','5','6','7') then itype='RU'; else itype='UK'; end; else do; /* 1993-1999 */ if ped_loc in ('02','03') then itype='NR'; else if ped_act='51' then itype='RI'; else if ped_act='52' then itype='RN'; else if ped_act in ('53','54','55','56','57','58','59','60','61') or ped_loc in ('01','05','06','07') then itype='RU'; else itype='UK'; end; </pre>
Michigan	<pre> /* intent (Driver/Pedestrian Intent) and per_act (Driver/Pedestrian Action) in Person data set */ if year in ('1990','1991') then do; if intent='01' then itype='RI'; else if intent='02' then itype='RN'; else if intent in ('03','04','05','06','07','08','09','10') then itype='RU'; else if intent='11' then itype='NR'; else itype='UK'; end; else if year='1992' then do; if per_act='23' then itype='RI'; else if per_act='24' then itype='RN'; else if per_act in ('25','26','27','28','29','30','31','32') then itype='RU'; else if per_act='33' then itype='NR'; else itype='UK'; end; else do; /* 1993-1999 */ if per_act='24' then itype='RI'; else if per_act='25' then itype='RN'; else if per_act in ('26','27','28','29','30','31','32','33') then itype='RU'; else if per_act='34' then itype='NR'; else itype='UK'; end; </pre>
Missouri	<pre> /* ped_act (Pedestrian Action) in Person data set */ if ped_act in ('01','02','03','04') then itype='RI'; else if ped_act in ('05','06','07','08','09','10','11','12','13','15') then itype='RN'; else if ped_act='14' then itype='NR'; else itype='UK'; </pre>

STATE	QUERY
New Mexico ³²	<pre> /* ped_int (Pedestrian at Intersection), ped_act (Pedestrian Action) and ped_act2 (Pedestrian Action 2) in Person data set */ if ped_int in ('1','2','3','4') then itype='RI'; else if ped_act1 in ('1','2','3','4') or ped_act2 in ('1','2','3','4') then itype='RN'; else itype='UK'; </pre>
North Carolina	<pre> /* maneuver (Vehicle/Pedestrian Maneuver) in Vehicle data set */ if maneuver=17 then itype='RI'; else if maneuver=18 then itype='RN'; else if maneuver in (19,20,21,22,23,24,25,26,27,29) then itype='RU'; else if maneuver=28 then itype='NR'; else itype='UK'; </pre>
Ohio	<pre> /* per_act (Driver/Pedestrian Action) in Person data set */ if per_act='18' then itype='RI'; else if per_act='19' then itype='RN'; else if per_act in ('20','21','22','23','24','25','26') then itype='RU'; else if per_act='27' then itype='NR';³³ else itype='UK'; </pre>
Pennsylvania	<pre> /* ped_loc (Pedestrian Location) in Person data set, rd_rel (Relation to Road), int_type (Intersection Type) in Crash data set */³⁴ if year in ('1990','1991','1992','1993','1994','1995','1996') then do; if ped_loc in ('02','03','04','19') then itype='RI'; else if ped_loc in ('05','06','07','20') then itype='RN'; else if ped_loc in ('01','14') then itype='RU'; else if ped_loc in ('08','09','10','11','12','13','15','16','17','18','21') then itype='NR'; else itype='UK'; end; else do; /* 1997-1999 */ if rd_rel in ('5','6','7','8') then itype='NR'; else if rd_rel in ('1','2','3','4') then do; if int_type in ('01','02','03','04','05','06','07','08','09') then itype='RI'; else if int_type='00' then itype='RN'; else itype='RU'; end; else itype='UK'; end; </pre>

³² New Mexico does not have an explicit 'Not in Roadway' code.

³³ Code 27= 'On sidewalk or shoulder'. This is the only non-roadway code used by Ohio.

³⁴ This query should be used with the avp data set. Note that, in 1997-1999, Pennsylvania's PEDESTRIAN LOCATION (ped_loc) is coded as unknown with a very high frequency. This necessitated the use of the Crash variables INTERSECTION TYPE (int_type) and RELATION TO ROAD (rd_rel) instead. Comparisons made between 1990-1996 and 1997-1999 should be regarded as only approximate.

STATE	QUERY
Texas	<pre> /* ped_act (Pedestrian Action) in Person data set */ if ped_act='1' then itype='RI'; else if ped_act='2' then itype='RN'; else if ped_act in ('0','3','4','5','6','7','8','9') then itype='RU'; else if ped_act='-' then itype='NR'; else itype='UK'; </pre>
Utah	<pre> /* ped_actn (Pedestrian Action) in Crash data set, ped_act (Pedestrian Action) in Person data set */ if year in ('1990','1991','1992','1993','1994') then do; if ped_actn in ('01','02','03','04') then itype='RI'; else if ped_actn='05' then itype='RN'; else if ped_actn in ('06','07','09','10','11','12','13','14','15','16','17','18','19', '21','22') then itype='RU'; else if ped_actn in ('08','20') then itype='NR'; else itype='UK'; end; else do; /* 1995-1999 */ if ped_act in ('01','02','03','04') then itype='RI'; else if ped_act='05' then itype='RN'; else if ped_act in ('06','07','09','10','11','12','13','14','15','16','17','18','19', '21','22') then itype='RU'; else if ped_act in ('08','20','24','25') then itype='NR'; else itype='UK'; end; </pre>
Virginia	<pre> /* per_act (Driver/Pedestrian Action) in Person data set */ if per_act in ('01','02','03','04') then itype='RI'; else if per_act in ('05','06') then itype='RN'; else if per_act in ('07','08','09','10','11','12','13','14','15','16','17','18') then itype='RU'; else if per_act='19' then itype='NR'; else itype='UK'; </pre>
Washington	<pre> /* ped_act (Pedestrian Action) in Person data set */ if ped_act in ('01','02','03','04','19') then itype='RI'; else if ped_act in ('06','07') then itype='RN'; else if ped_act in ('05','08','09','10','11','12','13','14','15') then itype='RU'; else if ped_act='16' then itype='NR'; else itype='UK'; </pre>

VARIABLE CATEGORY: PERSON TYPE

The variable *ptype* is typically derived using PERSON TYPE/INVOLVEMENT CODE (*loc*) and/or SEAT POSITION (*pos* or *seatpos*) in the Person data set. In certain instances, the variables PEDESTRIAN TYPE (*pppo* or *ped_type*) in the Person data set and/or VEHICLE TYPE (*veh_type*) in the Vehicle data set are also needed. Note that neither the Driver category nor the Passenger category include pedalcyclists; the pedalcyclist category includes both pedalcycle drivers and passengers. The variable *ptype* uses the following codes:

DR=Driver
PA=Passenger
PC=Pedalcyclist
PD=Pedestrian
OT=Other.

STATE	QUERY
California	<pre> if loc='1' then ptype='DR'; else if loc='2' then ptype='PD'; else if loc='4' then ptype='PC'; else if loc in ('6','7') then ptype='PA'; else ptype='OT'; </pre>
Florida	<pre> if year in ('1990','1991','1992') then do; if veh_type='10' then ptype='PC'; else if '01' le perno le '99' then ptype='PA'; else if pos='P' then ptype='PD'; else if pos='*' then ptype='DR'; else ptype='OT'; end; else do; /* 1993-1999 */ if veh_type='09' then ptype='PC'; else if '01' le perno le '99' then ptype='PA'; else if pos='P' then ptype='PD'; else if pos='*' then ptype='DR'; else ptype='OT'; end; </pre>
Georgia	<pre> if year='1998' then do; if veh_type='19' then ptype='PC'; else if loc='1' then ptype='DR'; else if loc='2' then ptype='PA'; else if loc='5' then ptype='PD'; else ptype='OT'; end; else do; /* 1990-1997 */ if veh_type='19' then ptype='PC'; else if pos='1' then ptype='DR'; else if pos in ('2','3','4','5','6','7','8') then ptype='PA'; else if pos='P' then ptype='PD'; else ptype='OT'; end; </pre>

STATE	QUERY
Illinois	<pre> if year in ('1990','1991','1992','1993','1994','1995') then do; if loc='0' then ptype='DR'; else if loc='1' then ptype='PD'; else if loc='3' then ptype='PC'; else if loc='5' then ptype='PA'; else ptype='OT'; end; else do; /* 1996-1999 */ if loc='A' then ptype='DR'; else if loc='E' then ptype='PD'; else if loc='G' then ptype='PC'; else if loc in ('C','D') then ptype='PA'; else ptype='OT'; end; </pre>
Indiana	<pre> if pppo='P' then ptype='PD'; else if pppo='B' then ptype='PC'; else if pos='1' then ptype='DR'; else if '01' le perno le '99' then ptype='PA'; else ptype='OT'; </pre>
Kansas	<pre> if ped_type='21' then ptype='PD'; else if ped_type='22' then ptype='PC'; else if pos='01' then ptype='DR'; else if pos in ('02','03','04','05','06','07','08','09','11','12', '13','14','15','16','17','88','99') then ptype='PA'; else ptype='OT'; </pre>
Maryland	<pre> if year in ('1990','1991','1992') then do; if pppo='1' then ptype='PD'; else if pppo='2' then ptype='PC'; else if pos='1' then ptype='DR'; else if pos in ('2','3','4','5','6','7','8','9') then ptype='PA'; else ptype='OT'; end; else do; /* 1993-1999 */ if pppo='01' then ptype='PD'; else if pppo in ('02','03') then ptype='PC'; else if pos='01' then ptype='DR'; else if pos in ('02','03','04','05','06','07','08','09','10') then ptype='PA'; else ptype='OT'; end; </pre>

STATE	QUERY
Michigan	<pre> if year in ('1990','1991') then do; if loc='3' then ptype='PD'; else if loc='4' then ptype='PC'; else if loc='2' or (loc='5' and pos='7') then ptype='PA'; else if loc='1' or (loc='5' and pos='0') then ptype='DR'; else ptype='OT'; end; else do; /* 1992-1999 */ if pos='P' then ptype='PD'; else if pos='B' then ptype='PC'; else if pos='01' then ptype='DR'; else if pos in ('02','03','04','05','06','07','08','09','10', '11','12','13','14') then ptype='PA'; else ptype='OT'; end; </pre>
Missouri	<pre> if loc='3' then ptype='PD'; else if ((loc in ('U','1','2') and pos='B') or loc='4') then ptype='PC'; else if loc in ('U','1') and pos NE 'B' then ptype='DR'; else if loc='2' and pos ne 'B' then ptype='PA'; else ptype='OT'; </pre>
New Mexico	<pre> if pos in ('LF','MD') then ptype='DR'; else if pos='PD' then ptype='PD'; else if pos='PC' then ptype='PC'; else if pos in ('CF','CR','LR','MP','OT','RF','RR') then ptype='PA'; else ptype='OT'; </pre>
North Carolina	<pre> if veh_type=18 then ptype='PC'; else if veh_type=21 then ptype='PD'; else if seatpos=1 then ptype='DR'; else if seatpos in (2,3,4,5,6) then ptype='PA'; else ptype='OT'; </pre>
Ohio	<pre> if veh_type='31' then ptype='PC'; else if loc='D' then ptype='DR'; else if loc='O' then ptype='PA'; else if loc='P' then ptype='PD'; else ptype='OT'; </pre>
Pennsylvania	<pre> if bodytype in ('90','91','92') then ptype='PC'; else if pos='1' then ptype='DR'; else if pos in ('2','3','4','5','6','8','9') then ptype='PA'; else if pos='7' then ptype='PD'; else ptype='OT'; </pre>

APPENDIX D

STATE	QUERY
Texas	<pre>if pppo='2' then ptype='PD'; else if pppo='3' then ptype='PC'; else if pos='1' then ptype='DR'; else if pos in ('2','3','4','5','6','7','8','9','+') then ptype='PA'; else ptype='OT';</pre>
Utah	<pre>if pos in ('011','11') then ptype='DR'; else if pos in ('001','01') then ptype='PD'; else if pos in ('004','04') then ptype='PC'; else if (('012' le pos le '055') or ('12' le pos le '55')) then ptype='PA'; else ptype='OT';</pre>
Virginia	<pre>if veh_type='09' then ptype='PC'; else if loc='1' then ptype='DR'; else if loc='2' then ptype='PA'; else if loc='3' then ptype='PD'; else ptype='OT';</pre>
Washington	<pre>if loc='1' then ptype='DR'; else if loc='2' then ptype='PA'; else if loc='3' then ptype='PD'; else if loc='4' then ptype='PC'; else ptype='OT';</pre>

VARIABLE CATEGORY: RESTRAINT USE/HELMET USE³⁵

The variable *rest* is normally derived using the variable RESTRAINT DEVICE (*rest1*) in the Person data set. Additional variables such as HELMET INFORMATION (*helmet*) or RESTRAINT DEVICE 2 (*rest2*), contained in the Person data set, may also be used. For this analysis, the categories '(Child) Restraint Improperly Used' and '(Child) Restraint Failure' have been classified as 'Restraint Not Used'. Also, airbag codes without explicit reference to restraint use (used/not used) are classified as 'Restraint Use Unknown'. The variable *rest* uses the following codes:

- 1=Restraint used
- 2=Restraint not used
- 3=Helmet Use Unknown/Restraint use unknown
- 4=Helmet used
- 5=Helmet not used.

STATE	QUERY
California	<pre> if rest1 in ('C','E','G','J','Q') then rest=1; else if rest1 in ('A','D','F','H','K','P','R','T','U') then rest=2; else if rest1 in ('W','Y') then rest=4; else if rest1 in ('V','X') then rest=5; else rest=3; </pre>
Florida	<pre> if year in ('1990','1991','1992') then do; if rest1 in ('2','3') then rest=1; else if rest1='1' then rest=2; else if rest1='4' then rest=4; else rest=3; end; else do; /* 1993-1999 */³⁶ if rest1='0' then do; if rest2='1' then rest=2; else if rest2 in ('2','3') then rest=1; else if rest2='5' then rest=4; else rest=3; end; else if rest1='1' then do; if rest2 in ('2','3','5') then rest=3; else rest=2; end; else if rest1 in ('2','3') then do; if rest2 in ('1','5','6') then rest=3; else rest=1; end; end; </pre>

³⁵ It should be noted that some states do not explicitly specify "Helmet not used". In these cases, "Restraint not used" codes in conjunction with *vtype*='MC' imply "Helmet not used". Also, observations in the Passenger Car/Light Truck data sets *vdpool* and *vcpool* with *rest*=4 or *rest*=5 are recoded as *rest*=3; observations in the Motorcycle data sets *mdpool* and *vmppool* with *rest*=1 (and *rest*=2 if the state uses explicit "Helmet not used" codes) are recoded as *rest*=3. That is, inconsistent observations are recoded as "Restraint use unknown" for vehicle occupants and "Helmet use unknown" for motorcyclists.

³⁶ For the period 1993-1999, when contradictory information between Florida's two restraint variables is found, the use of safety equipment is considered unknown.

STATE	QUERY
Florida (Cont.)	<pre> else if rest1='4' then do; if rest2='1' then rest=2; else if rest2 in ('2','3') then rest=1; else rest=3; end; else if rest1='5' then do; if rest2 in ('0','5','6') then rest=4; else rest=3; end; else if rest1='6' then do; if rest2='1' then rest=2; else if rest2='5' then rest=4; else rest=3; end; else do; /* rest1 not in '0', '1','2','3','4','5','6' */ if rest2='1' then rest=2; else if rest2 in ('2','3') then rest=1; else if rest2='5' then rest=4; else rest=3; end; end; </pre>
Georgia	<pre> if rest1 in ('1','2','4') or (year in ('1994','1995','1996','1997','1998') and rest1='3') then rest=1; else if rest1 in ('0','5') then rest=2; else if (year in ('1990','1991','1992','1993') and rest1='3') or (year in ('1994','1995','1996','1997','1998') and rest1='6') then rest=4; else rest=3; </pre>
Illinois	<pre> if year in ('1990','1991','1992','1993','1994','1995') then do; if rest1 in ('1','5') then rest=1; else if rest1 in ('2','6','7') then rest=2; else if rest1='3' then rest=4; else if rest1='4' then rest=5; else rest=3; end; else do; /* 1996-1999 */ if rest1 in ('2','6') then rest=1; else if rest1 in ('1','3','7','8') then rest=2; else if rest1='4' then rest=4; else if rest1='5' then rest=5; else if rest1='9' then do; if airbag='1' then rest=1; else if airbag='2' then rest=2; else rest=3; end; else rest=3; end; end; </pre>
Indiana	<pre> if rest1 in ('2','3','4') or rest2 in ('2','3','4') then rest=1; else if rest1='1' or rest2='1' then rest=2; else if rest1='5' or rest2='5' then rest=4; else rest=3; </pre>

STATE	QUERY
Kansas	<pre> if rest1 in ('C','L','S','X') then rest=1; else if rest1 in ('N','Y') then rest=2; else if rest1 in ('B','H') then rest=4; else if rest1='E' then rest=5; else rest=3; </pre>
Maryland³⁷	<pre> if year in ('1990','1991','1992') then do; if vtype='MC' then do; if rest1 in ('06','08') then rest=4; else if rest1 in ('07','10','11') then rest=5; else rest=3; end; else do; if rest1 in ('01','02','03','04') then rest=1; else if rest1 in ('10','11') then rest=2; else rest=3; end; end; else do; /* 1993-1999 */ if vtype='MC' then do; if rest1 in ('21','23') then rest=4; else if rest1 in ('22','00','01') then rest=5; else rest=3; end; else do; if rest1 in ('11','12','13','14','32') then rest=1; else if rest1 in ('00','01','31') then rest=2; else rest=3; end; end; end; </pre>
Michigan³⁸	<pre> if year in ('1990','1991') then do; if vtype='MC' then do; if helmet='1' then rest=4; else if helmet='2' then rest=5; else rest=3; end; else do; if rest1 in ('2','5') then rest=1; else if rest1 in ('1','3','6','7') then rest=2; else rest=3; end; end; else do; /* 1992-1999 */ if rest1 in ('02','03','04','06') then rest=1; else if rest1 in ('01','05','07','08') then rest=2; else if rest1='09' then do; if year='1992' then rest=4; else rest=3; end; end; </pre>

³⁷ The Maryland restraint use query should be used with [vppool](#).

³⁸ The Michigan restraint use query should be used with [vppool](#).

STATE	QUERY
Michigan (Cont.)	<pre> else if rest1='10' then do; if year='1992' then rest=5; else rest=4; end; else if rest1='11' then do; if year='1992' then rest=3; else rest=5; end; else rest=3; end; </pre>
Missouri	<pre> if year in ('1990','1991','1992') then do; if rest1 in ('3','4','6') then rest=1; else if rest1 in ('1','2') then rest=2; else rest=3; end; else do; /* 1993-1999 */ if rest1 in ('3','4','5','6') then rest=1; else if rest1 in ('1','2') then rest=2; else if rest1='7' then rest=4; else if rest1='8' then rest=5; else rest=3; end; </pre>
New Mexico³⁹	<pre> if year in ('1990','1991','1992','1993','1994','1995') then do; if vtype='MC' then do; if helmet='Y' then rest=4; else if helmet='N' then rest=5; else rest=3; end; else do; if rest1 in ('3','5','6','9') then rest=1; else if rest1 in ('1','2','4','7') then rest=2; else rest=3; end; end; else do; /* 1996-1999 */ if vtype='MC' then do; if helmet='Y' then rest=4; else if helmet='N' then rest=5; else rest=3; end; else do; if rest1 in ('3','5','6','8') then rest=1; else if rest1 in ('1','2','4','7') then rest=2; else rest=3; end; end; end; </pre>
North Carolina	<pre> if belt in (2,3,4) then rest=1; else if belt=1 then rest=2; else if belt=7 then rest=4; else rest=3; </pre>

³⁹ The New Mexico restraint use query should be used with vppool.

STATE	QUERY
Ohio ⁴⁰	<pre> if vtype='MC' then do; if helmet in ('2','3','4') then rest=4; else if helmet='1' then rest=5; else rest=3; end; else do; if rest1 in ('3','4','5','6') then rest=1; else if rest1 in ('1','2') then rest=2; else rest=3; end; </pre>
Pennsylvania	<pre> if rest_use='1' then do; if rest1 in ('1','2','3','4') or pas_rest='3' then rest=1; else if rest1='7' then rest=4; else rest=3; end; else if rest_use in ('0','2') then rest=2; else rest=3; </pre>
Texas	<pre> if helmet in ('1','2','3') then rest=4; else if helmet='4' then rest=5; else if rest1 in ('1','2','3','5') then rest=1; else if rest1='6' then rest=2; else rest=3; </pre>
Utah	<pre> if year in ('1990','1991','1992') then do; if rest1 in ('1','2','5') then rest=1; else if rest1 in ('3','4') then rest=2; else if rest1 in ('7','9') then rest=4; else if rest1='8' then rest=5; else rest=3; end; else do; /* 1993-1999 */ if rest1 in ('1','2','5','6','A') then rest=1; else if rest1 in ('3','4','7') then rest=2; else if rest1 in ('0','8') then rest=4; else if rest1='9' then rest=5; else rest=3; end; </pre>
Virginia ⁴¹	<pre> if rest1 in ('2','3','4','5') then rest=1; else if rest1='1' then rest=2; else rest=3; </pre>
Washington	<pre> if helmet='1' then rest=4; else if helmet='2' then rest=5; else if rest1 in ('2','3','4','5','6','8') then rest=1; else if rest1 in ('1','7','9') then rest=2; else rest=3; </pre>

⁴⁰ The Ohio restraint use query should be used with vppool.

⁴¹ Virginia does not currently use helmet codes.

VARIABLE CATEGORY: ROLLOVER

The variable *rollover* is typically derived using the variable FIRST HARMFUL EVENT⁴² (event1) in the Crash data set. In some cases, subsequent event codes, e.g., SECOND HARMFUL EVENT (event2) may also be used. A few states code rollover information at the vehicle level. In these cases, vehicle harmful event codes, e.g., VEHICLE FIRST HARMFUL EVENT⁴³ (vevent1), may be used instead. The variable *rollover* uses the following codes:

1=Rollover
0=No Rollover.

Since the rollover indicator is generally contained in the state's Crash data set, statistics based on vehicle type were only generated for single-vehicle crashes (cases where the vehicle rolling over could be definitively determined). It is a simple matter to select single-vehicle rollovers using the Crash-Vehicle data set, e.g.

```
data test.sv_roll; /* selects rollovers for autos, light trucks involved in single
                    vehicle crashes                                     */
set test.avpool;
if rollover=1 and ctype='S' and vtype in ('PC','ST');
run;
```

Occupants of passenger cars and light trucks involved in single-vehicle rollovers can be selected using a conditional merge, e.g.

```
data test.occ_roll;
merge test.sv_roll (in=A) test.vopool;
by caseno vehno;
if A;
run;
```

Note that there is a wide variability in the coding of rollover information by the states. For this reason, the rollover queries used in this report were evaluated using information from the Fatality Analysis Reporting System (FARS). FARS single-vehicle rollover crashes involving passenger vehicles were selected. The total number of crashes with a rollover in the first harmful event (*rollover=1*) was considered a lower bound. The total number of crashes with a rollover in either the first harmful event or in a subsequent event (*rollover in (1,2)*) was considered an upper bound. The FARS upper and lower bounds found for each state were then compared to State Data System (SDS) fatal single-vehicle rollover crash totals.

⁴² For California, the variable COLLISION TYPE (col_type) is used. For Pennsylvania, the variable event1 corresponds to MOST HARMFUL EVENT.

⁴³ For North Carolina, the Vehicle-level variable MOST HARMFUL EVENT (mostharm) is used in conjunction with the Crash-level variable FIRST HARMFUL EVENT (event1). For Virginia, the variable VEHICLE DAMAGE (veh_dam1) is used.

For SDS, states with a single rollover criterion either at the Crash-level or Vehicle-level were reported as stated. These states included California, Florida, Georgia, Illinois, Indiana, Kansas, Missouri, New Mexico, Ohio, Pennsylvania, Texas, Virginia, and Washington. Generally, states with a single rollover criterion had SDS rollover totals in a region bracketed by the FARS upper and lower bounds. States with rollover totals typically in the middle portion of the region included California, Florida, Georgia, Illinois, Missouri, Virginia, and Washington. Ohio SDS rollover totals were generally much closer to the FARS lower bound than the upper bound. Indiana, Kansas, and Pennsylvania had SDS rollover totals generally near the FARS upper bound. New Mexico and Texas had SDS rollover totals that considerably exceeded the FARS upper bound. Using FARS as a standard, it appears that Texas SDS rollover totals are overestimated by at least 30-40% for 1990-1996, and overestimated by at least 60% for 1997-1999. New Mexico SDS rollover totals, using FARS as a standard, appear to be overestimated by 200-300% for 1990-1999.⁴⁴

States using multiple rollover criteria tended to have a low incidence of rollovers as first harmful events, i.e., SDS rollover totals calculated using only first event indicators were generally much lower than the FARS lower bound. In these cases, additional subsequent event or most harmful event variables were used in the queries to place the totals within the region bracketed by the FARS upper and lower bounds. Generally, when using additional Crash-level or Vehicle-level variables, the SDS rollover totals tended to be close to the FARS upper bound. This is expected since the FARS upper bound includes both first event and subsequent event rollovers. States falling in this category included Maryland, Michigan, North Carolina⁴⁵ and Utah.

As mentioned above, using FARS as a standard, excepting Ohio, SDS rollover totals generally tended to be considerably higher than the FARS lower bound. A number of factors may be involved. When subsequent event or most harmful event variables are used, the SDS rollover totals will naturally tend towards the FARS upper bound. In other cases, when a single FIRST HARMFUL EVENT variable is used, it is possible that the SDS rollover totals may be higher due to the inclusion of fixed object crashes with rollovers as subsequent events. There is a wide variability in the coding of this variable between states; some states have many categories, others only a few. Another likely contributing factor is that the FARS "Passenger Vehicles" and SDS "Passenger Cars/Light Trucks" categories have only an approximate correspondence. It is quite possible, in some cases, that the SDS rollover totals include light trucks that would not be classified as such by FARS. In several instances, the SDS distinction between light and large trucks is only approximate.⁴⁶ The table below illustrates the findings for 1999 as an example of the FARS/SDS comparison described above.

⁴⁴ For New Mexico, a likely contributing factor in the large discrepancy between FARS and SDS totals is that New Mexico's light truck classification is only approximate, and involves use of three variables (bodytype, model, veh_type). A similar situation occurs for Texas where the light truck classification involves two variables (bodytype, veh_type). In general, when several primary variables are used to partition a derived variable into categories, the accuracy of the partitioning can vary due to a number of factors.

⁴⁵ Queries used for North Carolina and Pennsylvania use MOST HARMFUL EVENT variables which record the most severe event occurring at the Crash-level (Pennsylvania) or Vehicle-level (North Carolina) of all events in the crash. Since rollovers frequently cause severe vehicle damage and personal injury, the use of MOST HARMFUL EVENT is reasonably equivalent to evaluating a series of individual events (FIRST HARMFUL EVENT, SECOND HARMFUL EVENT, etc.) for the purposes of this analysis.

⁴⁶ Consult the individual Vehicle Type queries that follow. Comparison of the queries with the actual vehicle classifications in the State User manuals should allow a researcher to evaluate the impact of light truck categorizations on the rollover data.

Comparison of 1999 FARS and SDS Single-Vehicle Passenger Car/Light Truck Rollover Crashes⁴⁷

State	FARS First Event	SDS	FARS First or Subsequent Event
California	108	191	354
Florida ⁴⁸	57	146	141
Georgia	32	96	132
Illinois ⁴⁹	59	68	169
Indiana ⁵⁰	14	103	79
Kansas ⁵¹	26	73	59
Maryland	4	38	40
Michigan ⁵²	32	136	96
Missouri	34	78	130
New Mexico ⁵³	25	119	43
North Carolina ⁵⁴	18	169	144
Ohio	9	32	108
Pennsylvania ⁵⁵	17	128	127
Texas ⁵⁶	138	407	259
Utah ⁵⁷	23	53	32
Virginia	15	37	87
Washington	47	77	87

⁴⁷ Georgia 1999 data is unavailable. The numbers stated are for 1998. Illinois 1999 data is incomplete. The numbers stated are for 1995, the last complete year of Illinois data in SDS. Washington 1999 data is unavailable. The numbers stated are for 1996.

⁴⁸ Florida's SDS totals generally fall in the middle of the region bracketed by the FARS upper and lower bounds.

⁴⁹ Illinois' SDS totals generally fall in the middle of the region.

⁵⁰ Indiana's SDS totals tend to be near the FARS upper bound.

⁵¹ Kansas' SDS totals tend to be near the FARS upper bound.

⁵² Best agreement with FARS was found using Michigan's first three HARMFUL VEHICLE EVENT variables (vevent1, vevent2, vevent3). However, the query chosen also uses FOURTH HARMFUL VEHICLE EVENT (vevent4) as this provided best agreement with Michigan's 1990-1991 time period, and generally was not significantly greater than the FARS upper bound.

⁵³ As mentioned previously, the New Mexico light truck classification is only approximate.

⁵⁴ North Carolina SDS totals tend to be near the FARS upper bound.

⁵⁵ Pennsylvania SDS totals are generally slightly under the FARS upper bound.

⁵⁶ As mentioned previously, the Texas light truck classification is only approximate.

⁵⁷ Utah's SDS totals tend to be near the FARS upper bound.

STATE	QUERY
California	if col_type='06' then rollover=1; else rollover=0;
Florida	if year in ('1990','1991','1992') then do; if event1 in ('18','30') then rollover=1; else rollover=0; end; else do; /* 1993-1999 */ if event1 in ('31','33') then rollover=1; else rollover=0; end;
Georgia	if year in ('1990','1991','1992','1993') then do; if event1='06' then rollover=1; else rollover=0; end; else do; /* 1994-1998 */ if event1='01' then rollover=1; else rollover=0; end;
Illinois	if year in ('1990','1991','1992','1993','1994','1995') then do; if event1 in ('02','32') then rollover=1; else rollover=0; end; else do; /* 1996-1999 */ if event1='05' then rollover=1; else rollover=0; end;
Indiana	if event1='3' then rollover=1; else rollover=0;
Kansas	if event1='01' then rollover=1; else rollover=0;
Maryland	if event1='11' or event2='11' then rollover=1; else rollover=0;
Michigan	if (year in ('1990','1991') and event1='01') or (year='1992' and (vevent1='01' or vevent2='01' or vevent3='01' or vevent4='01')) or (year in ('1993','1994','1995','1996','1997','1998','1999') and (vevent1='06' or vevent2='06' or vevent3='06' or vevent4='06')) then rollover=1; else rollover=0; /* avpool */
Missouri	if event1='10' then rollover=1; else rollover=0;
New Mexico	if event1='01' then rollover=1; else rollover=0;

APPENDIX D

STATE	QUERY
North Carolina	if (event1=4 or (event1 in (1,2,3,5,10,16,17,18,19) and mostharm=4)) then rollover=1; else rollover=0; /* avpool */
Ohio	if event1='16' then rollover=1; else rollover=0;
Pennsylvania	if event1='04 then rollover=1; else rollover=0;
Texas	if event1='0' then rollover=1; else rollover=0;
Utah	if (event1='8' or event2='8') then rollover=1; else rollover=0;
Virginia	if veh_dam1='3' then rollover=1; else rollover=0; /* avpool */
Washington	if event1='52' then rollover=1; else rollover=0;

VARIABLE CATEGORY: SEX

The variable *stype* is derived using the variable SEX (*sex*) in the Person data set and uses the following codes:

F=Female
M=Male
U=Unknown.

STATE	QUERY
California	<pre> if sex='1' then stype='M'; else if sex='2' then stype='F'; else stype='U'; </pre>
Florida	<pre> if sex='1' then stype='M'; else if sex='2' then stype='F'; else stype='U'; </pre>
Georgia	<pre> if year in ('1990','1991','1992','1993') then do; if sex='1' then stype='M'; else if sex='2' then stype='F'; else stype='U'; end; else do; /* 1994-1998 */ if sex='M' then stype='M'; else if sex='F' then stype='F'; else stype='U'; end; </pre>
Illinois	<pre> if sex='1' then stype='M'; else if sex='2' then stype='F'; else stype='U'; </pre>
Indiana	<pre> if sex='M' then stype='M'; else if sex='F' then stype='F'; else stype='U'; </pre>
Kansas	<pre> if sex='M' then stype='M'; else if sex='F' then stype='F'; else stype='U'; </pre>
Maryland	<pre> if year in ('1990','1991','1992') then do; if sex='1' then stype='M'; else if sex='2' then stype='F'; else stype='U'; end; else do; /* 1993-1999 */ if sex='01' then stype='M'; else if sex='02' then stype='F'; else stype='U'; end; </pre>

APPENDIX D

STATE	QUERY
Michigan	<pre>if year in ('1990','1991') then do; if sex='1' then stype='M'; else if sex='2' then stype='F'; else stype='U'; end; else do; /* 1992-99 */ if sex='M' then stype='M'; else if sex='F' then stype='F'; else stype='U'; end;</pre>
Missouri	<pre>if sex='M' then stype='M'; else if sex='F' then stype='F'; else stype='U';</pre>
New Mexico	<pre>if sex='M' then stype='M'; else if sex='F' then stype='F'; else stype='U';</pre>
North Carolina	<pre>if sex=1 then stype='M'; else if sex=2 then stype='F'; else stype='U';</pre>
Ohio	<pre>if sex in ('1','M') then stype='M'; else if sex in ('2','F') then stype='F'; else stype='U';</pre>
Pennsylvania	<pre>if sex='M' then stype='M'; else if sex='F' then stype='F'; else stype='U';</pre>
Texas	<pre>if sex='1' then stype='M'; else if sex='2' then stype='F'; else stype='U';</pre>
Utah	<pre>if sex in ('1','M') then stype='M'; else if sex in ('2','F') then stype='F'; else stype='U';</pre>
Virginia	<pre>if sex='1' then stype='M'; else if sex='2' then stype='F'; else stype='U';</pre>
Washington	<pre>if sex='1' then stype='M'; else if sex='2' then stype='F'; else stype='U';</pre>

VARIABLE CATEGORY: SPEEDING

The determination of speeding-related crashes requires use of the Crash, Vehicle and/or Person data sets. It should be noted that there is a wide variety in the type and number of speeding-related variables coded by each state. The queries used and the data sets they are applied to are listed below. For example, for California, speeding determinations require using the first query with the Crash data set (apool) and using the second query with the Vehicle data set (vpool). The individual indicator variables *speed1* and *speed2* are combined using the method described earlier in this appendix (see “Creation of Alcohol/Speeding Data Sets”).

STATE	QUERY
California	<pre>(1) if cause1='03' then speed1=1; else speed1=0; /* apool */ (2) if ((year in ('1990','1991') and violatn in ('25','52')) or (year in ('1992','1993','1994','1995','1996','1997','1998','1999') and confac1='25')) then speed2=1; else speed2=0; /* vpool */</pre>
Florida	<pre>(1) if confac1 in ('12','17') or confac2 in ('12','17') or confac3 in ('12','17') then speed1=1; else speed1=0; /* ppool */</pre>
Georgia	<pre>(1) if year in ('1990','1991','1992','1993') then do; if confac1 in ('13','14') or confac2 in ('13','14') or confac3 in ('13','14') or confac4 in ('13','14') then speed1=1; else speed1=0; end; else do; /* 1994-1998 */ if confac1 in ('05','22') or confac2 in ('05','22') or confac3 in ('05','22') or confac4 in ('05','22') then speed1=1; else speed1=0; end; /* vpool */</pre>
Illinois	<pre>(1) if cause1='01' or cause2='01' or (year in ('1996','1997','1998','1999') and (cause1='27' or cause2='27')) then speed1=1; else speed1=0; /* apool */ (2) if con_cir1='04' and year in ('1996','1997','1998','1999') then speed2=1; else speed2=0; /* ppool */</pre>
Indiana	<pre>(1) if cause1='07' then speed1=1; else speed1=0; /* apool */ (2) if confac1='07' or confac2='07' then speed2=1; else speed2=0; /* vpool */</pre>
Kansas	<pre>(1) if dcc1 in ('05','06') or dcc2 in ('05','06') or dcc3 in ('05','06') or dcc4 in ('05','06') or dcc5 in ('05','06') or dcc6 in ('05','06') or dcc7 in ('05','06') or dcc8 in ('05','06') or dcc9 in ('05','06') or dcc10 in ('05','06') then speed1=1; else speed1=0; /* ppool */</pre>

APPENDIX D

STATE	QUERY
Maryland	<pre> (1) if year in ('1990','1991','1992') and (cause1 in ('01','03') or cause2 in ('01','03')) then speed1=1; else speed1=0; /* apool */ (2) if year in ('1990','1991','1992') and confac1 in ('01','03') then speed2=1; else speed2=0; /* vpool */ (3) if year in ('1993','1994','1995','1996','1997','1998','1999') and (con_cir1 in ('18','21') or con_cir2 in ('18','21') or con_cir3 in ('18','21') or con_cir4 in ('18','21')) then speed3=1; else speed3=0; /* ppool */ </pre>
Michigan	<pre> (1) if year in ('1990','1991') then do; if cause1='01' then speed1=1; else speed1=0; end; else speed1=0; /* apool */ (2) if year='1992' then do; if confac1='02' then speed2=1; else speed2=0; end; else if year in ('1993','1994','1995','1996','1997','1998','1999') then do; if confac1='01' then speed2=1; else speed2=0; end; else speed2=0; /* vpool */ (3) if year in ('1990','1991') and haz_act='02' then speed3=1; else speed3=0; /* ppool */ </pre>
Missouri	<pre> (1) if ((year in ('1990','1991','1992','1993','1994') and (confac1 in ('01','02') or confac2 in ('01','02') or confac3 in ('01','02') or confac4 in ('01','02') or confac5 in ('01','02')) or (year in ('1995','1996','1997','1998','1999') and (confac1 in ('04','05') or confac2 in ('04','05') or confac3 in ('04','05') or confac4 in ('04','05') or confac5 in ('04','05')))) then speed1=1; else speed1=0; /* vpool */ </pre>
New Mexico	<pre> (1) if substr(confac1,1,1) in ('1','2') or substr(confac1,2,1) in ('1','2') or substr(confac1,3,1) in ('1','2') then speed1=1; else speed1=0; /* vpool */ </pre>
North Carolina	<pre> (1) if violat1 in (7,8) or violat2 in (7,8) or violat3 in (7,8) or violat4 in (7,8) or violat5 in (7,8) then speed1=1; else speed1=0; /* vpool */ </pre>

STATE	QUERY
Ohio	(1) if confac1='03' then speed1=1; else speed1=0; /* ppool */
Pennsylvania	(Method detailed on the following page.)
Texas	(1) if confac1 in ('1','2') then speed1=1; else speed1=0; /* vpool */
Utah	(1) if violat='02' or confac1='01' or confac2='01' then speed1=1; else speed1=0; /* vpool */
Virginia	(1) if max_spd in ('1','2') then speed1=1; else speed1=0; /* apool */ (2) if ptype='DR' and per_act in ('02','03') then speed2=1; else speed2=0; /* vppool */
Washington	(1) if confac1 in ('03','04') or confac2 in ('03','04') or confac3 in ('03','04') then speed1=1; else speed1=0; /* vpool */

Speeding Determination for Pennsylvania

Determination of speeding-related crashes involves use of a subsidiary Contributing Factor data set. The Contributing Factor data set contains all of the contributing factors for a given crash. The variable PRIMARY CONTRIBUTING FACTOR (cause1) in the Crash data set is the most important of the factors listed under the variable CONTRIBUTING FACTOR (con_fac) in the Contributing Factor data set.

The procedure is as follows:

- 1) The yearly Contributing Factor data sets are pooled (fpool created) in a manner similar to that detailed in the section “Creation of Primary Data Sets”. During creation of fpool, the speeding-related contributing factors are determined by deriving the variable *speed*.
- 2) The fpool data set is reduced to a crash level data set (spdcrash) by defining a second derived variable *speed1* and using the SAS *FIRST* and *LAST* variables.
- 3) The spdcrash data set is then merged with the apool data set for subsequent processing.

```
data test.fpool (keep=caseno seq_no con_fac speed);
set penn.pa90fac (in=y90) penn.pa91fac (in=y91) penn.pa92fac (in=y92)
    penn.pa93fac (in=y93) penn.pa94fac (in=y94) penn.pa95fac (in=y95)
    penn.pa96fac (in=y96) penn.pa97fac (in=y97) penn.pa98fac (in=y98)
    penn.pa99fac (in=y99);
if y90 then year='1990'; else if y91 then year='1991'; else if y92 then year='1992';
else if y93 then year='1993'; else if y94 then year='1994'; else if y95 then year='1995';
else if y96 then year='1996'; else if y97 then year='1997'; else if y98 then year='1998';
else year='1999';
if (con_fac in ('38','39') and year in ('1990','1991','1992')) or
    (con_fac in ('M1','M2','M3','M4','M5')
    and year in ('1993','1994','1995','1996','1997','1998','1999')) then speed=1;
else speed=0;
run;

proc sort data=test.fpool;
  by caseno;
run;

data spdcrash (keep=caseno speed1);
set test.fpool;
by caseno;
retain speed1 0;
if first.caseno then speed1=0;
if speed1<speed then speed1=speed;
if last.caseno then output;
run;

data test.apool;
merge test.apool spdcrash;
by caseno;
run;
```

VARIABLE CATEGORY: TIME OF CRASH (Day vs. Night)⁵⁸

The variable *daynight* is derived using the variable TIME OF ACCIDENT (time) present in most state crash data sets. For states where TIME OF ACCIDENT is not available, the variable HOUR OF ACCIDENT (hour) is used.

STATE	QUERY
California	<pre> if (0 le time le 559) then daynight='Night'; else if (600 le time le 1759) then daynight='Day'; else if (1800 le time le 2359) then daynight='Night'; else daynight='Unknown'; </pre>
Florida	<pre> if (1 le time le 559) then daynight='Night'; else if (600 le time le 1759) then daynight='Day'; else if (1800 le time le 2400) then daynight='Night'; else daynight='Unknown'; </pre>
Georgia	<pre> if (0 < time < 600) then daynight='Night'; else if (600 le time < 1800) then daynight='Day'; else if (1800 le time le 2400) then daynight='Night'; else daynight='Unknown'; </pre>
Illinois	<pre> if year in ('1996','1997','1998','1999') then do; if (0 le time le 559) then daynight='Night'; else if (600 le time le 1759) then daynight='Day'; else if (1800 le time le 2359) then daynight='Night'; else daynight='Unknown'; end; else do; /* 1990-1995 */ if (0 le hour le 5) then daynight='Night'; else if (6 le hour le 17) then daynight='Day'; else if (18 le hour le 23) then daynight='Night'; else daynight='Unknown'; end; </pre>
Indiana	<pre> if (1 le time le 559) then daynight='Night'; else if (600 le time le 1759) then daynight='Day'; else if (1800 le time le 2400) then daynight='Night'; else daynight='Unknown'; </pre>
Kansas	<pre> if ('0000' le time < '0600') then daynight='Night'; else if ('0600' le time < '1800') then daynight='Day'; else if ('1800' le time le '2359') then daynight='Night'; else daynight='Unknown'; </pre>

⁵⁸Some states designate midnight as time 0, while others use time 2400. Also, use the statement "length daynight \$7" which explicitly sets the field size prior to using this query.

APPENDIX D

STATE	QUERY
Maryland	<pre> if (1 le time le 559) then daynight='Night'; else if (600 le time le 1759) then daynight='Day'; else if (1800 le time le 2400) then daynight='Night'; else daynight='Unknown'; </pre>
Michigan	<pre> if year in ('1990','1991') then do; if (1 le hour le 6) then daynight='Night'; else if (7 le hour le 18) then daynight='Day'; else if (19 le hour le 24) then daynight='Night'; else daynight='Unknown'; end; else do; /* 1992-1999 */ if (0 le hour le 5) then daynight='Night'; else if (6 le hour le 17) then daynight='Day'; else if (18 le hour le 24) then daynight='Night'; else daynight='Unknown'; end; </pre>
Missouri	<pre> if (1 le time le 559) then daynight='Night'; else if (600 le time le 1759) then daynight='Day'; else if (1800 le time le 2400) then daynight='Night'; else daynight='Unknown'; </pre>
New Mexico	<pre> if (1 le time le 559) then daynight='Night'; else if (600 le time le 1759) then daynight='Day'; else if (1800 le time le 2400) then daynight='Night'; else daynight='Unknown'; </pre>
North Carolina	<pre> if (0 le time le 559) then daynight='Night'; else if (600 le time le 1759) then daynight='Day'; else if (1800 le time le 2359) then daynight='Night'; else daynight='Unknown'; </pre>
Ohio	<pre> if (0 le hour le 5) then daynight='Night'; else if (6 le hour le 17) then daynight='Day'; else if (18 le hour le 23) then daynight='Night'; else daynight='Unknown'; </pre>
Pennsylvania	<pre> if (0 le time le 599⁵⁹) then daynight='Night'; else if (600 le time le 1799) then daynight='Day'; else if (1800 le time le 2399) then daynight='Night'; else daynight='Unknown'; </pre>

⁵⁹ For Pennsylvania, times ending in 99 indicate that hour is known but not minute.

STATE	QUERY
Texas	<pre>if ('00' le hour le '05') then daynight='Night'; else if ('06' le hour le '17') then daynight='Day'; else if ('18' le hour le '23') then daynight='Night'; else daynight='Unknown';</pre>
Utah	<pre>if (1 le time le 559) then daynight='Night'; else if (600 le time le 1759) then daynight='Day'; else if (1800 le time le 2400) then daynight='Night'; else daynight='Unknown';</pre>
Virginia	<pre>if (0 le hour le 5) then daynight='Night'; else if (6 le hour le 17) then daynight='Day'; else if (18 le hour le 23) then daynight='Night'; else daynight='Unknown';</pre>
Washington	<pre>if (1 le time le 559) then daynight='Night'; else if (600 le time le 1759) then daynight='Day'; else if (1800 le time le 2400) then daynight='Night'; else daynight='Unknown';</pre>

VARIABLE CATEGORY: VEHICLE TYPE

The variables *vtype* and *mvtype* are derived using the VEHICLE TYPE (veh_type) variable. In certain cases, other variables such as VEHICLE BODY TYPE (bodytype) and SPECIAL VEHICLE TYPE (spectype) may also be used. The variable *vtype* has the following codes:

BU=bus
 LT=large truck
 MC=motorcycle
 PC=passenger car/station wagon
 ST=Light truck.

The variable *mvtype* designates a motor vehicle of any type and has the following code:

MV=motor vehicle.⁶⁰

STATE	VEHICLE TYPE	QUERY
California	All Motor Vehicles	if veh_type not in ('12','14') then vtype='MV';
	Buses	if veh_type in ('08','09') then vtype='BU';
	Large Trucks	if veh_type in ('06','07') then vtype='LT';
	Light Trucks	if veh_type in ('04','05') then vtype='ST';
	Motorcycles	if veh_type in ('03','15') then vtype='MC';
	Passenger Cars	if veh_type in ('01','02') then vtype='PC';
Florida	All Motor Vehicles	if (veh_type not in ('10','25') and year in ('1990','1991','1992')) or (veh_type not in ('09','13') and year in ('1993','1994','1995','1996','1997','1998','1999')) then mvtype='MV';
	Buses	if (veh_type='08' and year in ('1993','1994','1995','1996','1997','1998','1999')) or (veh_type in ('16','17','18','19','20') and year in ('1990','1991','1992')) then vtype='BU';
	Large Trucks	if veh_type in ('05','06') or (veh_type='04' and year in ('1993','1994','1995','1996','1997','1998','1999')) or (veh_type='27' and year in ('1990','1991','1992')) then vtype='LT';
	Light Trucks	if veh_type='02' or (veh_type='03' and year in ('1993','1994','1995','1996','1997','1998','1999')) or (veh_type='04' and year in ('1990','1991','1992')) then vtype='ST';
	Motorcycles	if (veh_type in ('10','11') and year in ('1993','1994','1995','1996','1997','1998','1999')) or (veh_type in ('07','09') and year in ('1990','1991','1992')) then vtype='MC';
	Passenger Cars	if veh_type='01' or (veh_type='15' and year in ('1990','1991','1992')) then vtype='PC';

⁶⁰Non-motor vehicles are designated by missing values.

STATE	VEHICLE TYPE	QUERY
Georgia	All Motor Vehicles	if ((year in ('1990','1991','1992','1993') and veh_type not in ('19','21')) or (year in ('1994','1995','1996','1997','1998') and veh_type ne '19')) then mvtype='MV';
	Buses	if veh_type='13' then vtype='BU';
	Large Trucks	if ((year in ('1990','1991','1992','1993') and veh_type in ('03','04','05','06','07','08','09','11','14')) or (year in ('1994','1995','1996','1997','1998') and veh_type in ('03','04','05','06','07','08','14')) then vtype='LT';
	Light Trucks	if ((year in ('1990','1991','1992','1993') and veh_type in ('02','10','12')) or (year in ('1994','1995','1996','1997','1998') and veh_type in ('02','09','10','11','12')) then vtype='ST';
	Motorcycles	if veh_type in ('17','18') then vtype='MC';
	Passenger Cars	if veh_type='01' then vtype='PC';
Illinois	All Motor Vehicles	mvtype='MV'; ⁶¹
	Buses	if (veh_type='08' and year in ('1990','1991','1992','1993','1994','1995')) or (veh_type in ('04','05') and year in ('1996','1997','1998','1999')) then vtype='BU';
	Large Trucks	if (veh_type in ('03','04') and year in ('1990','1991','1992','1993','1994','1995')) or (veh_type in ('06','07','08') and year in ('1996','1997','1998','1999')) then vtype='LT';
	Light Trucks	if (veh_type in ('05','06') and year in ('1990','1991','1992','1993','1994','1995')) or (veh_type in ('02','03','15') and year in ('1996','1997','1998','1999')) then vtype='ST';
	Motorcycles	if (veh_type='09' and year in ('1990','1991','1992','1993','1994','1995')) or (veh_type in ('10','11') and year in ('1996','1997','1998','1999')) then vtype='MC';
	Passenger Cars	if veh_type='01' or (veh_type='02' and year in ('1990','1991','1992','1993','1994','1995')) then vtype='PC';
Indiana	All Motor Vehicles	mvtype='MV'; ⁶²
	Buses	if veh_type in ('09','10') then vtype='BU';
	Large Trucks	if veh_type in ('04','05','06','6A') then vtype='LT';
	Light Trucks	if veh_type in ('02','03') then vtype='ST';
	Motorcycles	if veh_type in ('14','15','17') then vtype='MC';
	Passenger Cars	if veh_type='01' then vtype='PC';

⁶¹For Illinois, all vehicle categories are motor vehicles.

⁶²For Indiana, all vehicle categories are motor vehicles.

APPENDIX D

STATE	VEHICLE TYPE	QUERY
Kansas	All Motor Vehicles	if veh_type ne '25' then mvtype='MV';
	Buses	if veh_type in ('13','14','15') then vtype='BU';
	Large Trucks	if veh_type in ('10','11','12') then vtype='LT';
	Light Trucks	if veh_type in ('04','05','06') then vtype='ST';
	Motorcycles	if veh_type in ('02','03') then vtype='MC';
	Passenger Cars	if veh_type='01' then vtype='PC';
Maryland	All Motor Vehicles	mvtype='MV'; ⁶³
	Buses	if veh_type in ('10','11','12') then vtype='BU';
	Large Trucks	if veh_type in ('06','07') or (year in ('1993','1994','1995','1996','1997','1998','1999') and veh_type='05') then vtype='LT';
	Light Trucks	if (veh_type='05' and year in ('1990','1991','1992')) or (year in ('1993','1994','1995','1996','1997','1998','1999') and veh_type in ('08','20','21')) then vtype='ST';
	Motorcycles	if veh_type in ('01','19') then vtype='MC';
	Passenger Cars	if veh_type in ('02','03') then vtype='PC';
Michigan	All Motor Vehicles	if year in ('1990','1991') ⁶⁴ or (veh_type ne '00' and year in ('1992','1993','1994','1995','1996','1997','1998','1999')) then mvtype='MV';
	Buses	if (year in ('1990','1991') and veh_type in ('04','05')) or (year='1992' and (veh_type in ('14','23','26') or spectype='3')) or (year in ('1993','1994','1995','1996','1997','1998','1999') and spectype='3' and veh_type='11') then vtype='BU';
	Large Trucks	if (year in ('1990','1991') and veh_type='02' and bodytype in ('08','09','10')) or (year='1992' and veh_type in ('05','11','12','13','15','16','17','18','19','20','21','22','24','25','27')) or (year in ('1993','1994','1995','1996','1997','1998','1999') and veh_type='11' and spectype NE '3') then vtype='LT';
	Light Trucks	if (year in ('1990','1991') and veh_type='02' and bodytype in ('05','06','07')) or (year in ('1992','1993','1994','1995','1996','1997','1998','1999') and veh_type in ('02','03','04')) then vtype='ST';
	Motorcycles	if (year in ('1990','1991') and veh_type='03') or (year='1992' and veh_type in ('06','07')) or (year in ('1993','1994','1995','1996','1997','1998','1999') and veh_type in ('05','06')) then vtype='MC';
	Passenger Cars	if veh_type='01' then vtype='PC';

⁶³ For Maryland, all vehicle categories are motor vehicles.

⁶⁴ For Michigan (1990-1991), all vehicle categories are motor vehicles.

STATE	VEHICLE TYPE	QUERY
Missouri	All Motor Vehicles	if (veh_type ne '09' and year in ('1990','1991','1992')) or (veh_type ne '11' and year in ('1993','1994','1995','1996','1997','1998','1999')) then mvtype='MV';
	Buses	if veh_type in ('05','06') or (veh_type='07' and year in ('1993','1994','1995','1996','1997','1998','1999')) then vtype='BU';
	Large Trucks	if (veh_type in ('16','17') and year in ('1990','1991','1992')) or (veh_type='15' and gvwr not in ('A','9') and year in ('1990','1991','1992')) or (veh_type in ('18','19','20') and year in ('1993','1994','1995','1996','1997','1998','1999')) then vtype='LT';
	Light Trucks	if veh_type='04' or (veh_type='15' and gvwr in ('A','9') and year in ('1990','1991','1992')) or (veh_type in ('03','17') and year in ('1993','1994','1995','1996','1997','1998','1999')) then vtype='ST';
	Motorcycles	if veh_type='08' or (veh_type in ('07','18','19') and year in ('1990','1991','1992')) or (veh_type in ('10','21','22','23','24') and year in ('1993','1994','1995','1996','1997','1998','1999')) then vtype='MC';
	Passenger Cars	if veh_type in ('01','02') or (veh_type='03' and year in ('1990','1991','1992')) then vtype='PC';
New Mexico	All Motor Vehicles	mvtype='MV'; ⁶⁵
	Buses	if bodytype in ('BS','BP','BU') or (bodytype=' ' and model in ('BS','BP','BU')) or (bodytype=' ' and model=' ' and veh_type in ('BS','BP','BU')) then vtype='BU';
	Large Trucks	if bodytype in ('CB','CL','CM','CS','DP','DS','FB','FT','GG','ST','TA','TN','TR','TT') or (bodytype in (' ','TK') and model in ('CB','CL','CM','CS','DP','DS','FB','FT','GG','ST','TA','TN','TR','TT')) or (bodytype in (' ','TK') and model in (' ','TK') and veh_type in ('CB','CL','CM','CS','DP','DS','FB','FT','GG','ST','TA','TN','TR','TT')) then vtype='LT';

⁶⁵For New Mexico, all vehicle categories are motor vehicles. For the bus, large truck, light truck, motorcycle, and passenger car queries, using only the first of the three OR conditions provides a reasonable approximation.

APPENDIX D

STATE	VEHICLE TYPE	QUERY
New Mexico (Cont.)	Light Trucks	if bodytype in ('JP','PK','PU','PM','PN','VN','WG','UT','SV','VT') or (bodytype in (' ','TK') and model in ('JP','PK','PU','PM','PN','VN','WG','UT','SV','VT')) or (bodytype in (' ','TK') and model in (' ','TK') and veh_type in ('JP','PK','PU','PM','PN','VN','WG','UT','SV','VT')) then vtype='ST';
	Motorcycles	if bodytype='MC' or (bodytype=' ' and model='MC') or (bodytype=' ' and model=' ' and veh_type='MC') then vtype='MC';
	Passenger Cars	if bodytype in ('CV','CP','HT','LM','RD','RH','SD','2D','2H','2T','3D','3H','3T','4D','4H','4T','SW') or (bodytype=' ' and model in ('CV','CP','HT','LM','RD','RH','SD','2D','2H','2T','3D','3H','3T','4D','4H','4T','SW')) or (bodytype=' ' and model=' ' and veh_type in ('CV','CP','HT','LM','RD','RH','SD','2D','2H','2T','3D','3H','3T','4D','4H','4T','SW')) then vtype='PC';
North Carolina	All Motor Vehicles	if veh_type not in (18,21) then mvtype='MV';
	Buses	if veh_type in (4,5,6) then vtype='BU';
	Large Trucks	if veh_type in (8,9,10,24) then vtype='LT';
	Light Trucks	if veh_type in (3,7,23,25) then vtype='ST';
	Motorcycles	if veh_type in (14,15,16) then vtype='MC';
	Passenger Cars	if veh_type in (1,2,11) then vtype='PC';
Ohio	All Motor Vehicles	if veh_type not in ('24','29','30','31') then mvtype='MV';
	Buses	if veh_type in ('16','17','18') then vtype='BU';
	Large Trucks	if veh_type in ('07','08','09','10','11','33','34') then vtype='LT';
	Light Trucks	if veh_type in ('05','06') then vtype='ST';
	Motorcycles	if veh_type in ('12','13','14','15') then vtype='MC';
	Passenger Cars	if veh_type in ('01','02','03','04','22') then vtype='PC';
Pennsylvania	All Motor Vehicles	if veh_type in ('1','2','3','4','5') or (veh_type='6' and bodytype not in ('90','91','92','93','94','95','96','97','98')) then mvtype='MV';
	Buses	if veh_type='3' then vtype='BU';
	Large Trucks	if veh_type='5' then vtype='LT';
	Light Trucks	if veh_type='4' then vtype='ST';
	Motorcycles	if veh_type='2' then vtype='MC';
	Passenger Cars	if veh_type='1' then vtype='PC';

STATE	VEHICLE TYPE	QUERY
Texas	All Motor Vehicles	mvtype='MV'; ⁶⁶
	Buses	if veh_type in ('11','12') then vtype='BU';
	Large Trucks	if veh_type in ('04','05','06','07','08') and ((year in ('1990','1991','1992','1993','1994','1995') and bodytype='33') or bodytype in ('++','20','21','22','23','24','25','26','27','28', '31','32','34','36','37','39') or (bodytype='40' and veh_type in ('05','06','07','08'))) then vtype='LT';
	Light Trucks	if veh_type in ('04','05','06','07','08') and ((year in ('1996','1997','1998','1999') and bodytype='33') or bodytype in ('29','30','35','38') or (bodytype='40' and veh_type='04')) then vtype='ST';
	Motorcycles	if veh_type in ('13','14','16') then vtype='MC';
	Passenger Cars	if veh_type in ('01','02','03') then vtype='PC';
Utah	All Motor Vehicles	mvtype='MV'; ⁶⁷
	Buses	if veh_type in ('17','18') then vtype='BU';
	Large Trucks	if veh_type in ('13','14','15','16','31','32','33','34','35','36', '37','38','41','44','47','48','49','50','51') or veh_type='39' and year in ('1990','1991','1992','1993','1994')) then vtype='LT';
	Light Trucks	if veh_type in ('07','08','09','10','11','12','27','43') or (veh_type='40' and year in ('1990','1991','1992','1993','1994')) then vtype='ST';
	Motorcycles	if veh_type in ('19','20','21') or (veh_type='30' and year in ('1990','1991','1992','1993','1994')) then vtype='MC';
	Passenger Cars	if veh_type in ('01','02','03','04','05','06','42') then vtype='PC';
Virginia	All Motor Vehicles	if veh_type ne '09' then mvtype='MV';
	Buses	if veh_type in ('13','14','15') then vtype='BU';
	Large Trucks	if veh_type in ('04','05','06') then vtype='LT';
	Light Trucks	if veh_type in ('02','03') then vtype='ST';
	Motorcycles	if veh_type in ('10','11') then vtype='MC';
	Passenger Cars	if veh_type='01' then vtype='PC';
Washington	All Motor Vehicles	mvtype='MV'; ⁶⁸
	Buses	if veh_type in ('10','11') then vtype='BU';
	Large Trucks	if veh_type in ('03','04','05','06','07') then vtype='LT';
	Light Trucks	if veh_type='02' then vtype='ST';
	Motorcycles	if veh_type in ('12','13','15') then vtype='MC';
	Passenger Cars	if veh_type in ('01','09') then vtype='PC';

⁶⁶ For Texas, all vehicle categories are motor vehicles.

⁶⁷ For Utah, all vehicle categories are motor vehicles.

⁶⁸ For Washington, all vehicle categories are motor vehicles.

Use of Proc Freq for Data Table Generation

The SAS *PROC FREQ* procedure is used to generate the data tables. The example below generates a frequency table corresponding to the information for California in Table 4 “Motor Vehicle Crashes by Crash Type and Crash Severity”.

```
proc freq data=test.apool;
  tables year*sev*ctype /list sparse69 out=crashtyp70;
run;
```

Single/Multiple Vehicle Crashes: California 1990-1999

Year	Sev	Ctype	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1990	1	M	2,989	0.06	2,989	0.06
1990	1	S	1,669	0.03	4,658	0.10
1990	2	M	196,025	4.01	200,683	4.10
1990	2	S	40,491	0.83	241,174	4.93
1990	3	M	254,664	5.21	495,838	10.14
1990	3	S	55,993	1.14	551,831	11.28
1991	1	M	2,673	0.05	554,504	11.34
1991	1	S	1,491	0.03	555,995	11.37
1991	2	M	185,976	3.80	741,971	15.17
1991	2	S	38,028	0.78	779,999	15.95
1991	3	M	233,204	4.77	1,013,203	20.72
1991	3	S	53,018	1.08	1,066,221	21.80
1992	1	M	2,425	0.05	1,068,646	21.85
1992	1	S	1,282	0.03	1,069,928	21.88
1992	2	M	179,848	3.68	1,249,776	25.56
1992	2	S	35,224	0.72	1,285,000	26.28
1992	3	M	221,068	4.52	1,506,068	30.80
1992	3	S	51,161	1.05	1,557,229	31.84

⁶⁹ SAS options are specified after the “/”. *LIST* is an alternative output format for *PROC FREQ*. The *SPARSE* option adds combinations of variables to the output data set that do not exist in the initial data set. This is useful when combinations whose frequency count=0 are needed in the final data set for tabulation purposes.

⁷⁰ The *OUT=* option allows the frequency table information to be output to a summary data set. In the production code, the creation of output data sets using *PROC FREQ* was automated using macro *DO* loops and macro variables; appropriate final data sets were then exported to spreadsheets for presentation. The *OUT=* option can be omitted in the example.

Year	Sev	Ctype	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1993	1	M	2,436	0.05	1,559,665	31.89
1993	1	S	1,242	0.03	1,560,907	31.92
1993	2	M	170,252	3.48	1,731,159	35.40
1993	2	S	32,404	0.66	1,763,563	36.06
1993	3	M	218,471	4.47	1,982,034	40.53
1993	3	S	52,685	1.08	2,034,719	41.61
1994	1	M	2,503	0.05	2,037,222	41.66
1994	1	S	1,269	0.03	2,038,491	41.68
1994	2	M	170,842	3.49	2,209,333	45.18
1994	2	S	32,381	0.66	2,241,714	45.84
1994	3	M	218,989	4.48	2,460,703	50.32
1994	3	S	54,109	1.11	2,514,812	51.42
1995	1	M	2,405	0.05	2,517,217	51.47
1995	1	S	1,231	0.03	2,518,448	51.50
1995	2	M	164,877	3.37	2,683,325	54.87
1995	2	S	31,692	0.65	2,715,017	55.52
1995	3	M	216,430	4.43	2,931,447	59.94
1995	3	S	55,123	1.13	2,986,570	61.07
1996	1	M	2,322	0.05	2,988,892	61.12
1996	1	S	1,233	0.03	2,990,125	61.14
1996	2	M	162,420	3.32	3,152,545	64.46
1996	2	S	31,385	0.64	3,183,930	65.10
1996	3	M	222,056	4.54	3,405,986	69.64
1996	3	S	56,269	1.15	3,462,255	70.80
1997	1	M	2,150	0.04	3,464,405	70.84
1997	1	S	1,102	0.02	3,465,507	70.86
1997	2	M	155,846	3.19	3,621,353	74.05
1997	2	S	30,106	0.62	3,651,459	74.66
1997	3	M	220,301	4.50	3,871,760	79.17
1997	3	S	54,389	1.11	3,926,149	80.28

APPENDIX D

Year	Sev	Ctype	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1998	1	M	2,009	0.04	3,928,158	80.32
1998	1	S	1,066	0.02	3,929,224	80.34
1998	2	M	158,778	3.25	4,088,002	83.59
1998	2	S	30,229	0.62	4,118,231	84.21
1998	3	M	232,285	4.75	4,350,516	88.96
1998	3	S	58,241	1.19	4,408,757	90.15
1999	1	M	1,979	0.04	4,410,736	90.19
1999	1	S	1,165	0.02	4,411,901	90.21
1999	2	M	159,172	3.25	4,571,073	93.47
1999	2	S	29,558	0.60	4,600,631	94.07
1999	3	M	235,864	4.82	4,836,495	98.90
1999	3	S	54,026	1.10	4,890,521	100.00

Other data tables are generated in a similar fashion using the appropriate data sets and variables. For example, the information for California in Table 49 “Passenger Car and Light Truck Occupants Killed or Injured by Restraint Use and Person Type” can be generated using the following *PROC FREQ* statements:

```
proc freq data=test.vcpool;
  tables year*pinj*rest*ptype /list sparse;
  title1 "INJURY STATUS BY PERSON TYPE AND RESTRAINT USE, CALIFORNIA 1990-1999";
  title2 "OCCUPANTS OF CARS AND LIGHT TRUCKS";
run;

proc freq data=test.vcpool;
  tables year*pinj*rest /list sparse;
  title1 "INJURY STATUS BY RESTRAINT USE, CALIFORNIA 1990-1999";
  title2 "OCCUPANTS OF CARS AND LIGHT TRUCKS";
run;

proc freq data=test.vcpool;
  tables year*pinj /list sparse;
  title1 "INJURY STATUS, OCCUPANTS OF CARS AND LIGHT TRUCKS, CALIFORNIA 1990-1999";
run;
```

Supplemental Information

This section describes additional technical procedures used to generate the data herein.

Use of Macros

When incorporating the queries above in the state processing code, it is convenient to encapsulate them in a SAS construct termed a macro that, in its simplest form, is equivalent to a subroutine in other programming languages. A macro begins with the statement `%MACRO` and ends with the statement `%MEND`. For example, the code that assigns a year to each observation in the data set can be encapsulated in a macro and invoked multiple times⁷¹ while specifying the code only once:

```
%macro assignyr72;
  if y90 then year='1990'; else if y91 then year='1991'; else if y92 then year='1992';
  else if y93 then year='1993'; else if y94 then year='1994';
  else if y95 then year='1995'; else if y96 then year='1996';
  else if y97 then year='1997'; else if y98 then year='1998'; else year='1999';
%mend assignyr;

/* creation of pooled crash data set */
data test.apool (keep=caseno ctype daynight num_fat num_inj rollover sev wend1 year);
  set missouri.mo90acc (in=y90) missouri.mo91acc (in=y91) missouri.mo92acc (in=y92)
      missouri.mo93acc (in=y93) missouri.mo94acc (in=y94) missouri.mo95acc (in=y95)
      missouri.mo96acc (in=y96) missouri.mo97acc (in=y97) missouri.mo98acc (in=y98)
      missouri.mo99acc;
  %assignyr73
run;
```

Procedure Used When Case Numbers are Not Unique Between State Years

In certain instances, the identifying variable ACCIDENT NUMBER (caseno) is not unique between years. This will cause errors in subsequent MERGE operations. The macro below can be used to create unique case numbers.

```
%macro uniqcase;
  length temp $6;74
  temp=caseno;
  casenum=year||temp;75
%mend uniqcase;
```

⁷¹ The %assignyr macro can be invoked during the build of the Crash, Vehicle, and Person data sets.

⁷² When first declaring a macro, it must be given a name in order to reference it later.

⁷³ Note that a semicolon is generally not needed as in other SAS statements. Adding a trailing semicolon will not affect correct operation of the macros needed herein.

⁷⁴ Note that the length needed depends on the length of the case numbers in the state under consideration. The following states require the macro %uniqcase (the recommended length is given in parentheses): California, Washington (6), Florida, Illinois, Maryland, Michigan, New Mexico, Utah (10), Georgia (25). The long length for Georgia is due to a long case number for 1998.

⁷⁵ Note that the derived variable *casenum* replaces the variable ACCIDENT NUMBER (caseno) and is used in any subsequent SET or MERGE operations needed (i.e., use "by casenum" and "by casenum vehno" as appropriate). The symbol '||' is the SAS concatenation operator.

The following code illustrates the method for California.⁷⁶

```
/* creation of pooled crash data set */
data test.apool;
  set calif.ca90acc calif.ca91acc calif.ca92acc calif.ca93acc calif.ca94acc
      calif.ca95acc calif.ca96acc calif.ca97acc calif.ca98acc calif.ca99acc;
  %assignyr
  %uniqcase77
run;
```

Using the Queries with Primary Data Sets

The following SAS code illustrates the use of the person type query for California.

```
%macro pertype;
  if loc='1' then ptype='DR';
  else if loc='2' then ptype='PD';
  else if loc='4' then ptype='PC';
  else if loc in ('6','7') then ptype='PA';
  else ptype='OT';
%mend pertype;

/* creation of pooled person data set */
data test.ppool;
  set calif.ca90per calif.ca91per calif.ca92per calif.ca93per calif.ca94per
      calif.ca95per calif.ca96per calif.ca97per calif.ca98acc calif.ca99per;
  %assignyr
  %uniqcase
  %perotype
run;
```

Note that the code above using macros is equivalent to the code below:

```
/* creation of pooled person data set */
data test.ppool;
  set calif.ca90per calif.ca91per calif.ca92per calif.ca93per calif.ca94per
      calif.ca95per calif.ca96per calif.ca97per calif.ca98acc calif.ca99per;
  if y90 then year='1990'; else if y91 then year='1991'; else if y92 then year='1992';
  else if y93 then year='1993'; else if y94 then year='1994';
  else if y95 then year='1995'; else if y96 then year='1996';
  else if y97 then year='1997'; else if y98 then year='1998'; else year='1999';
  length temp $6;
  temp=caseno;
  casenum=year||temp;
  if loc='1' then ptype='DR';
  else if loc='2' then ptype='PD';
  else if loc='4' then ptype='PC';
  else if loc in ('6','7') then ptype='PA';
  else ptype='OT';
run;
```

⁷⁶ Note that the macro %assignyr must precede the macro %uniqcase since the variable year, created in %assignyr, is needed in %uniqcase.

⁷⁷ Note that if this procedure is used for one of the primary data sets, it must be used for all primary data sets (Crash, Vehicle, and Person).

Using the Queries with Secondary Data Sets

The following code illustrates the use of queries with secondary data sets. Note that the derived variable *pctype* for North Carolina uses information contained in both the Vehicle and Person data sets.

```
%macro pctype;
if seatpos=1 then do;
  if veh_type=18 then pctype='PC';
  else if veh_type=21 then pctype='PD';
  else pctype='DR';
end;
else if seatpos in (2,3,4,5,6) then pctype='PA';
else pctype='OT';
%mend pctype;

data test.vppool;
merge test.vpool test.ppool (in=A);
by caseno vehno;
if A;
%pctype
run;
```

Removal of Inconsistent Observations

In certain instances, a small number of inconsistent observations were removed from the data sets before processing. These deleted observations fall into two categories: 1) multiple observations with the same key (i.e., multiple cases with identical case numbers, cases with multiple vehicles with identical vehicle numbers), and 2) cases without matching vehicles in the Vehicle data set, or vehicles without matching cases in the Crash data set. The code below illustrates the method for the testing of Illinois data 1990-1999.

```
/* test for duplicate accident records in 1990-1999 */
%macro acctotal;
%do i=90 %to 99;78
  data test.ilcdup&i;
    set illinois.il&i.acc;79
  by caseno;
  retain ctotat 0;
  if first.caseno then ctotat=0;
  ctotat+1;
  if last.caseno and ctotat>1 then output;
run;
%end;
%mend acctotal;
```

⁷⁸ This is an example of a SAS %DO loop using macro variables. The SAS keywords %DO, %TO, and %END are similar to counter loops in other programming languages, and to the corresponding DO, TO, and END reserved words in normal SAS code. The % symbol must be used since the inner portion of the loop references the macro counter variable i (as &i). Note that the loop runs 10 times. In the first iteration, the macro variable i=90, the initial data set is il90acc, and the resultant data set containing duplicate case numbers is ilcdup90. For testing Georgia, use "%do i=90 %to 98;". For testing North Carolina use "%do i=92 %to 99;". For testing Washington, use "%do i=90 %to 96;".

⁷⁹ Change this line appropriately for other states, e.g. for Missouri use "set missouri.mo&i.acc;".

```
%accttotal /* invokes the macro */

/* test for duplicate vehicle records in 1990-1999 */
%macro vehtotal;
%do i=90 %to 99;
  data temp (keep=caseno);
    set illinois.il&i.veh;
    by caseno vehno;
    retain vttotal 0;
    if first.vehno then vttotal=0;
    vttotal+1;
    if last.vehno and vttotal>1 then output;
  run;

  data test.ilvdup&i (keep=caseno);
    set temp;
    by caseno;
    if first.caseno then output;
  run;
%end;
%mend vehtotal;

%vehtotal /* invokes the macro */

/* test for missing accident records and vehicle records for 1990-1999 */
%macro missing;
%do i=90 %to 99;
  data temp (keep=caseno);
    merge illinois.il&i.acc (in=A) illinois.il&i.veh (in=B);
    by caseno;
    if A and B then delete;
  run;

  data test.ilmiss&i (keep=caseno);
    set temp;
    by caseno;
    if first.caseno then output;
  run;
%end;
%mend missing;

%missing /* invokes the macro */
```

The resultant data sets in the code above are built regardless of whether the initial data sets are consistent or not. Tests on consistent data sets will contain the appropriate data vectors but will have no observations. If any inconsistent case numbers are found⁸⁰, the case numbers in the individual output data sets (e.g., `ilcdup90`) can be combined into one data set using the *SET* statement. Duplicate case numbers should be removed using the statement “if first.caseno then output;” as in previous examples. Finally, the resultant set of bad case numbers is used in a conditional merge with each of the primary data sets (Crash, Vehicle, Person) to remove the bad case numbers from subsequent processing.

⁸⁰ Consult the SAS LOG after running the programs to determine the error data sets with observations.

For example, the code below removes bad case numbers contained in a data set named errors from the primary Crash data set (apool).

```
data test.apool;
  merge errors (in=A) test.apool;
  if A then delete;
run;
```

Suggestions for Efficient Data Processing

Judicious use of the SAS *KEEP* statement can dramatically reduce processing time when creating and merging the pooled data sets. The *KEEP* statement can be used with data sets in the *DATA*, *SET*, and *MERGE* statements. The following code illustrates the method with the creation of the California Person data set.

```
%macro alc3;81
  if alc_drug in ('2','3','4') then alcohol3=1;
  else alcohol3=0;
%mend alc3;

data test.ppool (keep=casenum vehno pinj ptype stype age age1 alcohol3 rest year child);
  set calif.ca90per (in=y90 keep=caseno vehno inj age sex loc pos rest1 alc_drug)
      calif.ca91per (in=y91 keep=caseno vehno inj age sex loc pos rest1 alc_drug)
      calif.ca92per (in=y92 keep=caseno vehno inj age sex loc pos rest1 alc_drug)
      calif.ca93per (in=y93 keep=caseno vehno inj age sex loc pos rest1 alc_drug)
      calif.ca94per (in=y94 keep=caseno vehno inj age sex loc pos rest1 alc_drug)
      calif.ca95per (in=y95 keep=caseno vehno inj age sex loc pos rest1 alc_drug)
      calif.ca96per (in=y96 keep=caseno vehno inj age sex loc pos rest1 alc_drug)
      calif.ca97per (in=y97 keep=caseno vehno inj age sex loc pos rest1 alc_drug)
      calif.ca98per (in=y98 keep=caseno vehno inj age sex loc pos rest1 alc_drug)
      calif.ca99per (in=y99 keep=caseno vehno inj age sex loc pos rest1 alc_drug);
  %assignyr
  %uniqcase
  %alc3      /* alcohol */
  %perinj    /* injury severity */
  %per type  /* person type */
  %sext type /* sex */
  %agecalc   /* age */
  %safety    /* restraint use */
  %child     /* child restraint use */
run;
```

⁸¹ The macros referenced in the creation of the Person data set can be created using the queries listed earlier. For example, the information needed to create the macro %perinj can be found in the "Injury Severity" section of "Creation of Derived Variables". The alcohol query is contained here as an example. The macros %assignyr and %uniqcase were referenced earlier in this appendix.

In the code above, the *KEEP* statements used in the *SET* statement result in faster processing because SAS will only search for the variables listed in the *KEEP* statement when processing observations. The *KEEP* statement in the *DATA* statement results in a dramatically smaller resultant data set since only the variables listed are kept in the data vector. In this example, all the variables that are kept are derived variables excepting VEHICLE NUMBER (vehno), used in merging operations, and the variable AGE (age), which is needed for creating the Young Driver, Older Driver and Child data sets. The analogous SAS *DROP* statement is also useful.

Another useful procedure for improving processing time is to build multiple data sets in a single *DATA* step. The method is illustrated using the Crash-Person data set.

```
data test.ped (keep=pinj itype age1 year) /* pedestrians */
    test.driver (keep=pinj age1 year) /* all drivers */
    test.yngdr (keep=sev stype year) /* young drivers */
    test.olddr (keep=sev stype year); /* older drivers */
set test.appool;
if ptype='PD' then output test.ped;
if ptype='DR' then do;
    output test.driver;
    if 15 le age le 20 then output test.yngdr;
    if 70 le age le 99 then output test.olddr; /* this condition is state-specific */
end;
run;
```