(5) SELECTING AND RESTRICTING VARIABLES

Just as there are times when you will want to select or restrict the observations in a SAS data set, you may also want to select or restrict the variables. By default, all variables mentioned in a data step are added to the SAS data set being created. The default behavior can be altered by DROP and KEEP statements, or by DROP and KEEP data set options.

**Example 5.1...**

```sas
data males females;
  infile "k:\epi514\data\cancer99.txt";
  input @01 county $2. @03 gender $1. @04 age 3. @07 cause $4. @11 place $1.
  if gender eq '1' then output males;
  else                  output females;
  drop gender;
run;
```

This is a copy of example 4.10. Two gender-specific data sets are created using OUTPUT statements to direct observations to the appropriate data set. Since the data sets are gender-specific, the variable gender is not needed in either data set, so it is dropped using a DROP statement. The DROP statement in this example is used when reading raw data, but it can also be used when reading a data set. First create a data set that contains data from all records in the raw data file. Then use another data step to create the gender-specific data sets.

**Example 5.2...**

```sas
data cancer99;
  infile "k:\epi514\data\cancer99.txt";
  input @01 county $2. @03 gender $1. @04 age 3. @07 cause $4. @11 place $1.
  ;
run;
```

data males females;  
set cancer99;
if gender eq '1' then output males;
else                  output females;
drop gender;
runch;

Data set CANCER99 contains observations for both males and females. A second data step reads that data set with SET statement and creates the two gender-specific data sets. The DROP statement removes the variable GENDER from both of the new data sets, not from CANCER99.

There is also a DROP data set option. That is useful when different variables are to be dropped from various data sets. A DROP statement affects all data sets be created in a data step. The DROP data set option is data set-specific.
...Example 5.3...

data males (drop=gender)
     females (drop=gender)
     albany (drop=county)
;  
set cancer99;
if gender eq '1' then output males;
else                  output females;

if county eq '01' then output albany;
run;

title 'ORIGINAL DATA SET CANCER99';
proc contents data=cancer99 short;
run;

title 'NEW DATA SET MALES-VARIABLES SELECTED WITH DROP';
proc contents data=males short;
run;

title 'NEW DATA SET FEMALES-VARIABLES SELECTED WITH DROP';
proc contents data=females short;
run;

title 'NEW DATA SET ALBANY-VARIABLES SELECTED WITH DROP';
proc contents data=albany short;
run;

Three data sets are created in one data step. The variable GENDER is not needed in data sets MALES
and FEMALES ① while the variable COUNTY is not needed in the county-specific data set ALBANY ②.
The DROP data set options are data set-specific. Different variables can be dropped from different data
sets. If a DROP statement had been used instead ...

drop gender county;

the two variables would be dropped from all three data sets, not what you wanted. Output statement
create data sets MALES and FEMALES ③ and data set ALBANY ④. PROC CONTENTS is run using the
SHORT option to look at a list of variables in each data set ⑥. All variables remain in the original data set
used in the SET statement, CANCER99. The DROP data set options had the desired results for the
remaining three data set.

There are also KEEP statements and KEEP data set options. They both have the opposite effect of
DROP, they tell SAS what variables you want in a data set as opposed to specifying the ones you do not
want. Just like the DROP statement, a KEEP statement affects all data sets being created in a data step,
while a KEEP data set option is specific to a data set.

...Example 5.4...
data males females albany; ⑥
set cancer99;
if gender eq '1' then output males;
else                  output females;
if county eq '01' then output albany;
keep age cause;
run;
Once again, you are creating three data sets. However, you only want two variables in the data sets, \textit{AGE} and \textit{CAUSE}. The \texttt{KEEP} statement tells SAS to place only two variables in each of the newly created data sets. All variables remain in data set \texttt{CANCER99}.

As with \texttt{DROP}, if you want place different variables in different data sets, you use a \texttt{KEEP} data set option, not a \texttt{KEEP} statement.

\textit{Example 5.5...}

\begin{verbatim}
  data males (keep=age cause)  females (keep=age cause)
   albany (keep=gender age cause);
  set cancer99;
  if gender eq '1' then output males;
  else                  output females;
  if county eq '01' then output albany;
  run;

  title 'ORIGINAL DATA SET CANCER99';
  proc contents data=cancer99 short;
  run;

  title 'NEW DATA SET MALES-VARIABLES SELECTED WITH KEEP';
  proc contents data=males short;
  run;

  title 'NEW DATA SET FEMALES-VARIABLES SELECTED WITH KEEP';
  proc contents data=females short;
  run;

  title 'NEW DATA SET ALBANY-VARIABLES SELECTED WITH KEEP';
  proc contents data=albany short;
  run;
\end{verbatim}

Three data sets are created in one data step. The variables \textit{AGE} and \textit{CAUSE} are kept in data sets \texttt{MALES} and \texttt{FEMALES}. Variables \texttt{GENDER}, \textit{AGE}, and \textit{CAUSE} are kept in data set \texttt{ALBANY}. The \texttt{KEEP} data set options are data set-specific. Different variables can be dropped from different data sets. If a \texttt{KEEP} statement had been used instead ...

\begin{verbatim}
  keep gender age cause;
\end{verbatim}

the three variables would be kept in all three data sets, not what you wanted. Output statement create data sets \texttt{MALES} and \texttt{FEMALES} and data set \texttt{ALBANY}. \texttt{PROC CONTENTS} is run using the \texttt{SHORT} option to look at a list of variables in each data set. All variables remain in the original data set used in the \texttt{SET} statement, \texttt{CANCER99}. The \texttt{KEEP} data set options had the desired results for the remaining three data set.
You may now be asking when to use DROP and when to use KEEP. The answer is easy ... use the statement (or data set option) that requires the least amount of writing of SAS code get the desired result. For example, the output shown below lists all the variables in a data set that contains data from the 2000 census. To save you the counting, there are 215 variables in the data set.

Imagine that you want to create a new data set that contained only the variables COUNTY, PCT012001 (total population), PCT012002 (male population), and PCT012106 (female population). Would you write a KEEP statement and list the four variables you want, or a DROP statement and list the 211 variables you do not want? Hopefully you answered KEEP. However, in this and other similar situations where you select a small subset from a large number of variables, a KEEP statement is not the most efficient way to create the new data set.

Imagine that you want to create a new data set that contained only the variables COUNTY, PCT012001 (total population), PCT012002 (male population), and PCT012106 (female population). Would you write a KEEP statement and list the four variables you want, or a DROP statement and list the 211 variables you do not want? Hopefully you answered KEEP. However, in this and other similar situations where you select a small subset from a large number of variables, a KEEP statement is not the most efficient way to create the new data set.

...Example 5.6...

libname sf1 "k:\epi514\census";

data census;
  set sf1.pct12;
  keep county pct012001 pct012002 pct012106;  
  run;

data censusb (keep=county pct012001 pct012002 pct012106);
  set sf1.pct12;
  run;

data censusc;
  set sf1.pct12 (keep=county pct012001 pct012002 pct012106);
  run;

Each of the three data steps in example 5.6 produce a data set that contains all the observations, but only four of the variables from data set SF1.PCT12. The first data step uses a KEEP statement ①. The SET statement in that data step reads all the variables from each observation, then writes the values of the four kept variables to the new data set. The second data step uses a KEEP data set option on the new data set ②. Once again, the SET statement in that data step reads all the variables from each observation, then writes the values of the four kept variables to the new data set. The last data step uses a KEEP data set option, but this time it is used with the data set being read with the SET statement ③. Now, the SET statement only reads the values of four variables from the large data set, not 215. The third data step is the most efficient method since you only read and use the variables you want.
Using a KEEP data set option with the data set in a SET statement is not always possible even though it might appear that it is the best way to make a data set. In example 5.4, three new data sets were created that each contained only two variables from a larger data set. That situation appears to be perfect for using a KEEP data set option on the large data set in the SET statement ... why read all the variables when you only want two of them.

...Example 5.7...

```sas
data males females albany;
set cancer99;
if gender eq '1' then output males;
else output females;
if county eq '01' then output albany;
keep age cause;
run;
```

```sas
data males females albany;
set cancer99 (keep=age cause);
if gender eq '1' then output males;
else output females;
if county eq '01' then output albany;
run;
```

The first data step is a repeat of example 5.4. The KEEP statement results in there only being two variables in each of the new data sets. The second data step with a KEEP data set option on data set CANCER99 will run without any errors, but it does not produce the desired result. Here is the LOG showing what happens when each data step is run.

The first data step places the correct number of observations in each data set. But, the second data step places no observations in data sets MALES and ALBANY, and all the observations from CANCER99 in data set FEMALES. Why? The lower portion of the LOG contains two notes about variables being uninitialized, GENDER and COUNTY. When you read the data with SET, you tell SAS to keep only AGE and CAUSE. Then you use the variables GENDER and COUNTY in IF statements. Since you did not read them, you cannot use them in the data step. The variable is not available so GENDER is never equal to ‘1’ (no males). Since the IF statement says if GENDER is not ‘1’, write the observation to data set FEMALES, all observations are written to that data set. The next IF statement is never true since the variable COUNTY cannot be used in the data step. It was never read and no observations are written to data set ALBANY.

What you should remember...

When you create data sets, only keep the variables you really need. That saves both space used to store your data and time to process your data ... examples: if you have a data set with observations for only males, you do not need the variable GENDER; if you have a data set with observations for only the year 2000, you do not need the variable YEAR.

DROP and KEEP statements affect all data sets created in a data step. DROP and KEEP data set options affect only the data set they with which they are used. Whether you use DROP or KEEP very often is just an issue of what list of variables is shorter. As shown in example 5.6, for purposes of efficiency, it is also important to consider where you use a DROP or KEEP data set option. Finally, as shown in example 5.7, BE CAREFUL.