


STAT 481

Intermediate SAS

14.4 - The RENAME= option

 [Printer-friendly version \(https://onlinecourses.science.psu.edu/stat481/print/book/export/html/17\)](https://onlinecourses.science.psu.edu/stat481/print/book/export/html/17)

There may be occasions in which you want to change some of the variable names in your SAS data set. To do so, you'll want to use the RENAME= option. As its name suggests, the RENAME= option allows you to change the variable names within a SAS data set.

The format of the RENAME= option is:

```
RENAME = (old1=new1 old2=new2 .... oldk=newk);
```

where *old1*, *old2*, ..., *oldk* are the variable names as they appear in the data set that precedes the RENAME= option, and *new1*, *new2*, ..., *newk* are the corresponding new variable names.

The effect of the RENAME= option depends on where it appears:

- If the RENAME= option appears in the SET statement, then the new variable name takes effect when the program data vector is created. Therefore, all programming statements within the DATA step must refer to the *new* variable name.
- If the RENAME= option appears in the DATA statement, then the new variable name takes effect only when the data are written to the SAS data set. Therefore, all programming statements within the DATA step must refer to the *old* variable name.

Example 14.12. The following program illustrates the use of the RENAME= option in the SET statement. Specifically, the variable *sex* is changed to *gender*, and *b_date* is changed to *birth*, **when the program data vector is created**:

```
DATA back7 (keep = subj gender v_date birth age);
  set back3 (rename={sex=gender b_date=birth});
  age = (v_date - birth)/365;  *MUST use NEW name for date of birth;
RUN;

PROC PRINT data=back7;
  title 'Output Dataset: BACK7';
RUN;
```

(/stat481/sites/onlinecourses.science.psu.edu/stat481/files/lesson02/sasdata/sas_L0310.sas)



(/stat481/sites/onlinecourses.science.psu.edu/stat481/files/lesson02/sasdata/sas_L0212.sas)





(/stat481/sites/onlinecourses.science.psu.edu/stat481/files/lesson02/sas_L0212_output.gif)

Because the RENAME= option appears in the SET statement, SAS no longer recognizes the variable name *sex* as the gender, nor *b_date* as the birth date, of the subject. Instead, SAS recognizes the variable names *gender* and *birth*. Hence, when we subsequently calculate the subjects' ages (*age*) in the DATA step, we must refer to the new variable name *birth*.

Again, pay particular attention to the syntax of the RENAME= option ... it too can be tricky. The entire RENAME= option must be contained in parentheses immediately following the data set to which you want the name changes to apply. The variable names must also be placed in parentheses. So, in general, the syntax, when applied to a DATA statement, should look like this:

```
DATA dsname (RENAME = (o1=n1 o2=n2 ...));
```

where *dsname* is the data set name and *o1* and *o2* are the old variable names, and *n1* and *n2* are the new variable names.

Launch and run  the SAS program. Review the output from the PRINT procedure. Convince yourself that the variable names *sex* and *b_date* have been changed as advertised to *gender* and *birth*, respectively. Also, verify that the ages of the subjects have been calculated appropriately. Then, in the SAS program, change the variable name *birth* back to the variable name *b_date*, and re-run  the program. Does SAS indeed hiccup?

Example 14.13. The following program illustrates use of the RENAME= option, when it appears in the DATA statement. Specifically, the variable *sex* is changed to *gender*, and *b_date* is changed to *birth* **when SAS writes the data to the output data set**:

```

DATA back8 (rename=(sex=gender b_date=birth)
            keep = subj sex v_date b_date age);
    set back3;
    age = (v_date - b_date)/365;      *MUST use OLD name for date of birth;
RUN;

PROC PRINT data=back8;
    title 'Output Dataset: BACK8';
RUN;

```

(/stat481/sites/onlinecourses.science.psu.edu.stat481/files/lesson02/sasndata/sas_L0310.sas)




(/stat481/sites/onlinecourses.science.psu.edu.stat481/files/lesson02/sasndata/sas_L0213.sas)



(/stat481/sites/onlinecourses.science.psu.edu.stat481/files/lesson02/sas_L0213_output.gif)

Because the RENAME= option appears in the DATA statement, SAS only recognizes the variable names as they appear in the input data set *back3*. That is, for example, SAS recognizes the variable name *b_date* as the birth date of the subjects. Hence, when we subsequently calculate the subjects' ages in the DATA step, we must refer to the old variable name *b_date*. Also note that the KEEP= option in the DATA statement must refer to the original variable names as they appear in the *back3* data set.

This program also illustrates how to use more than one DATA step option at a time. Specifically, the RENAME= and KEEP= options are used to modify the *back8* data set. As such, both options are placed within one set of parentheses immediately following the data set to which you want the changes to apply. Then, within those parentheses, the basic syntax for each option is followed.

Launch and run  the SAS program, and review the output from the PRINT procedure. Convince yourself that the variable names *sex* and *b_date* have been changed as advertised to *gender* and *birth*, respectively. Also, verify that the ages of the subjects have been calculated appropriately.

◀ 14.3 - The WHERE= option (/stat481/node/16)

up
(/stat481/node/13)

14.5 - The IN= option ▶ (/stat481/node/18)