Chapter 11 Markdowns

By @TNTprizz80315

SAS: Output Control in DATA Step

Data Filtering

If we want to do something after writing the data into the set, we can use OUTPUT statement to do this. Filtering is also possible using this.

```
...
    INPUT DAT1 - DAT3;
    A = SUM(OF _NUMERIC_);
    OUTPUT;
    B = MEAN(OF DAT1 - DAT3); /* This row will not be shown in the result.*/
...
    /* Filtering */
    INPUT SEX AGE;
    IF SEX = 'M' THEN OUTPUT;
    IF AGE > 40 THEN OUTPUT;
    /* Only output the SEX = 'M' rows or whose age larger than 40.*/
```

Note: The OUTPUT statement is automatically done in the end of the step if no OUTPUT statement exist within the step.

If we want to delete a row, use DELETE.

```
...
    INPUT AGE NAME $;
    IF AGE > 120 THEN DELETE;
    /* Delete the rows with age larger than 120.*/
...
```

We can use WHERE statement if SET is used for data input. This would be more efficient than IF statements.

```
DATA nine_up;
    SET drinks;
    WHERE name IN ('9up', '九喜');
    /* Only include the 9up drinks in the set nine_up. */
RUN;
```

Note: No variable declaration is allowed if `WHERE` is used.

We can also use `STOP` to terminate the whole code execution.

```
...
    INPUT C1 - C10;
    IF C5 = 3 STOP;
    /* Once C5 = 3, the line and the records afterwards will not be
stored. */
    avg = MEAN(OF C1 - C10);
```

Column Managing

We can use `DROP` to delete unwanted columns. All the other columns will be kept.

```
...
    INPUT C1 - C10;
    ss = SUM(OF C1 - C10);
    avg = ss / 10;
    DROP C1 - C10; /* We do not want the original data. */
RUN;
```

Note: The dropped variable can still be used in the program. Only the output is affected.

We can use `KEEP` to keep wanted columns. All the other columns will be deleted.

```
...
    INPUT C1 - C10;
    ss = SUM(OF C1 - C10);
    avg = ss / 10;
    KEEP avg; /* We only want the processed result. */
RUN;
```

We can also include the statement in the `DATA` line.

```
DATA tsmc(drop = salary);
    *DROP salary;
    ...
```

Note: For all the column-managing related statements, variable list syntax can be applied.

Multiple Data Set in One `DATA` Step

Just state the affected/created dataset name in the step and after the `OUTPUT` statement.

```
DATA Junior Senior;
    SET emp;
    IF age < 30 OR salary < 24000 THEN
        OUTPUT Junior;
    ELSE
        OUTPUT Senior;
RUN;
```

Data Set Options

Almost all of the options are self-explanatory and optional.

```
DATA tsmc (DROP = age WHERE = (salary > 24000));
    /* No, the statement is deleting the age columns and filtering
the data separately, but not delete the age columns with salary
larger than 24000. */
    SET emp (FIRSTOBS = 10 OBS = 20);
    ...
RUN;
```

Here is a list of data set options:

Options	Explanation
`KEEP` `DROP`	See Column Managing section.
`RENAME`	Rename a column. `RENAME = (OLDN = NEWN OLDN2 = NEWN2)`
`WHERE`	See Data Filtering section.
`FIRSTOBS` `OBS`	Specify the range of rows to be used. `FIRSTOBS = 20 OBS = 40` Will read rows 20 to 40.
`LABEL`	Declare a data set label.

Note: This can apply to any DATA name, including those in `PROC` step.