Lesson 1: Controlling Input and Output

Summary

Main Points

Outputting Multiple Observations

```
OUTPUT <SAS-data-set(s)>;
```

- You can control when SAS writes an observation to a SAS data set by using an explicit OUTPUT statement in your code. After you use an explicit OUTPUT statement, there is no implicit OUTPUT statement at the bottom of the DATA step.

Writing to Multiple SAS Data Sets

```
DATA SAS-data-set-name SAS-data-set-name-n;
```

- To create more than one data set, you specify the names of the SAS data sets you want to create in the DATA statement. Separate SAS data set names with a space.
- You can use OUTPUT statements with conditional logic to create multiple data sets that contain observations based on the value of a variable in the input data set.

```
SELECT <(select-expression)>;
  WHEN-1 (when-expression -1 <..., when-expression-n>) statement;
  WHEN-n (when-expression -1 <..., when-expression-n>) statement;
  <OTHERWISE statement>;
END;
```

- You can use a SELECT group for conditional processing in a DATA step. The SELECT group begins with the keyword SELECT. The optional SELECT expression specifies any SAS expression that evaluates to a single value. Often a variable name is used as the SELECT expression.
- The WHEN statement begins with the keyword WHEN followed by at least one WHEN expression. The WHEN expression can be any SAS expression, including a compound expression. The WHEN expression is followed by a statement, which can be any executable SAS statement.
- The optional OTHERWISE statement specifies a statement to be executed if no WHEN condition is met.
- The keyword END signals the end of the select group.
Lesson 1: Controlling Input and Output

- When a SELECT expression is specified, SAS evaluates the SELECT expression and compares that value to each WHEN expression and returns either a true or false. If no SELECT expression is specified, SAS evaluates each WHEN expression in order until it finds a true expression. If no WHEN expression is true, SAS executes the OTHERWISE statement if one is present.
- You can use functions in a SELECT expression.
- You can use DO-END groups in a SELECT group. You can execute multiple statements when a WHEN expression is true by using DO-END groups.

Controlling Variable Input and Output

```
SAS-data-set-name (DROP=variable(s))
SAS-data-set-name (KEEP=variable(s))
```

- The DROP= and KEEP= data set options can be used to specify variables to drop or keep in the output data.
- When the DROP= and KEEP= data set options are used in the SET statement, the variables are not processed and are not available in the program data vector.
- When the DROP= and KEEP= data set options are used in the DATA statement, they affect the variables in the output data set they are associated with.
- DROP and KEEP statements affect all output data sets listed in the DATA statement.
- You can use a combination of data set options and statements. If you use them together, statements are applied before data set options. If you attempt to drop and keep the same variable, you will get a warning.

Controlling Observation Input and Output

```
SAS-data-set-name (OBS=n)
SAS-data-set-name (FIRSTOBS=n)
```

- You can use the OBS= and FIRSTOBS= data set options to limit the number of observations that SAS processes.
- The OBS= data set option specifies the number of the last observation to process. It does not specify how many observations should be processed.
- The FIRSTOBS= data set option specifies a starting point for processing an input data set. By default, FIRSTOBS=1.
- You can use FIRSTOBS= and OBS= together to define a range of observations for SAS to process.
Lesson 1: Controlling Input and Output

- FIRSTOBS= and OBS= can be used in a procedure step to limit the number of observations that are processed. If a WHERE statement is used to subset the observations, it is applied before the data set options.

Sample Code

Outputting Multiple Observations

```sas
data forecast;
   set orion.growth;
   Year=1;
   Total_Employees=Total_Employees*(1+Increase);
   output;
   Year=2;
   Total_Employees=Total_Employees*(1+Increase);
   output;
run;
```

Writing to Multiple SAS Data Sets (Using a SELECT Group)

```sas
data usa australia other;
   set orion.employee_addresses;
   select (Country);
   when ('US') output usa;
   when ('AU') output australia;
   otherwise output other;
   end;
run;
```
Lesson 1: Controlling Input and Output

Writing to Multiple Data Sets (Using a SELECT Group with DO-END Group in the WHEN statement)

```sas
data usa australia other;
   set orion.employee_addresses;
   select (upcase(Country));
   when ('US') do;
      Benefits=1;
      output usa;
   end;
   when ('AU') do;
      Benefits=2;
      output australia;
   end;
   otherwise;
      Benefits=0;
      output other;
   end;
   end;
run;
```

Controlling Variable Input and Output

```sas
data usa australia(drop=State) other;
   drop Country;
   set orion.employee_addresses
     (drop=Employee_ID);
   if Country='US' then output usa;
   else if Country='AU' then output australia;
   else output other;
run;
```
Lesson 1: Controlling Input and Output

Controlling Observation Input and Output

data australia;
    set orion.employee_addresses
        (firstobs=50 obs=100);
    if Country='AU' then output;
run;

proc print data=orion.employee_addresses
    (obs=10);
    where Country='AU';
    var Employee_Name City State Country;
run;