Fundamental Principles of Basic SAS

There are two main divisions of a SAS program: DATA step and PROCedure step.

- DATA step reads raw data and turns it into SAS data set.
- PROC step analyzes SAS data set.

DATA step overview:

- DATA step processes raw data one row (observation) at a time.
- The current row of data is held in PDV (Program Data Vector) as it is processed.
- DATA step statements make changes in the PDV.
- At end of the DATA step, the PDV is added to the SAS data set being created.
- Then the DATA step repeats automatically for each new row (observation) of data.

Example:

```
data records;
  input name $ salary age tenure gender $;
cards;
Bill  27000  32  7  M
Sally  30000   40  10  F
RUN;
```

This data step creates the SAS dataset `work.records`, containing two observations and five variables – three numeric variables and two categorical.

PROC step overview:

- PROC does statistical analysis or applies utility procedures to a SAS data set
- PROC inputs a SAS data set created in a previous DATA step
- PROC processes data by whole columns (variables) at once, rather than by rows
- Output of PROC is formatted automatically according to default report form for the PROC
- Tell PROC which SAS dataset to analyze with **DATA=<SAS DS>** option
- Tell PROC which variables to analyze with **VAR <LIST OF VARS>** option
- Tell PROC to analyze in subsets with **BY <VAR(S)>** option

Example:

```
proc means data=records;
  var salary age;
    by gender;
run;
```

This proc produces simple summary statistics on two variables, salary and age, in the existing SAS dataset `work.records`, separately for males and females.

Naming SAS datasets:

- SAS data set is named in the **DATA;** statement.
- SAS data sets have two names: **LIBREF.FILENAME**
- **LIBREF** is a SAS alias for the physical location of the SAS dataset.
- The default **LIBREF** is **WORK**, the alias for a temporary storage location
- The default **FILENAME** is **DATA<N>**
- You can define a permanent storage location by using **LIBNAME** (Ex: **LIBNAME EMP 'C:\MYFOLDER';**)

The three types of SAS data set names:

1) Default LIBREF, default FILENAME – Ex: **DATA;** creates WORK.DATA1
2) Default LIBREF, explicit FILENAME – Ex: **DATA RECORDS;** creates WORK.RECORDS
3) Explicit LIBREF, explicit FILENAME – Ex: **DATA EMP.RECORDS;** creates EMP.RECORDS. The third way also requires an explicit LIBNAME statement.

Only the third way preserves the SAS dataset on your computer after you exit SAS.
Example:

```
libname emp "/home/tomsager/business_analytics";
data emp.records;
     input name $ salary age tenure gender $;
cards;
   Bill  27000  32  7  M
   Sally  30000   40  10  F
run;
```

This data step creates a "permanent" SAS dataset called `records.sas7bdat`, which is stored in the SAS OnDemand folder `/home/tomsager/business_analytics`. If you are running PC SAS on your own computer, you can store SAS datasets permanently on your computer by using the Windows folder naming convention in the `libname` statement. Ex: `libname emp "c:\users\tomsager\myfolder";` However, this `libname` statement does not work in SAS OnDemand: You cannot directly reference your Windows folders in an SAS OnDemand program; but you can save your SAS dataset to a SAS OnDemand folder as shown above and then download it to your Windows computer.

Four types of data can be processed by DATA step:

1) "**CARDS**" data –
   - Data are physically included in the program.
   - **INPUT** statement describes the structure of the data.

   Example: Any of the preceding data steps that use `input`.

2) Text files –
   - Data are in a different file – a text file (like NOTEPAD's .TXT files – not Excel)
   - **INFILE** statement tells SAS where the text file is (Ex: **INFILE 'C:\MYFOLDER\RECORDS.TXT';**)
   - **INPUT** statement describes the structure of the data.

Example:

```
data records;
  infile "/home/tomsager/business_analytics/personnel.txt"
  input name $ salary age tenure gender $;
run;
```

This data step creates the SAS data set `work.records` out of the text file `personnel.txt`, which contains five variables in the order listed in the `input`.

3) SAS data sets –
   - Data are already in a SAS dataset, so SAS already knows the structure.
   - **SET <NAME OF SAS DATASET>;** brings in the data.
   - **INFILE** and **INPUT** are not used because SAS already knows the structure.

   Example:

```
data records;
  set emp.records;
  salary = 1.10*salary;
run;
```

   This data step creates the temporary SAS data set `work.records` out of the permanent SAS data set `records.sas7bdat`, stored in the `emp` location, and gives all employees a 10% raise. The permanent SAS data set continues to exist without change.

4) Structured files from other programs –
   - SAS can import Excel and Access data
   - Use the menu **File → Import Data** and let the Import Wizard guide you through the process

Notes for SAS OnDemand:

- The above comments also apply to SAS OnDemand.
- However, SAS OnDemand cannot reference files on your local computer.
- You must upload files from your computer to the cloud in order for SAS OnDemand to use them.
- Your **WORK** directory in SAS OnDemand is in the cloud, not on your computer.
- You can create a "permanent" storage location for your SAS files in SAS OnDemand, but it also will be in the cloud, not on your computer.
- You can upload files to and download files from SAS OnDemand by clicking on the upload and download icons on the left side of your SAS OnDemand screen. This moves your files to and from the cloud.

Ex: In SAS OnDemand, /HOME/<YOUR PROFILE NAME>/ is the proper way to refer to your root level ("My Folders") storage area. For example, the following program creates a permanent SAS dataset called `apts.sas7bdat` and stores it in the Business_Analytics subfolder of the root level folder.

```
libname  BA "/home/tomsager/Business_Analytics";
data BA.apts;
  input  rent  area ;
cards;
519  725
765  995
run;
```