K2 Analytics
Building Skills, Building Individuals

Welcome to SAS Training

- Rajesh Jakhotia

30-Sep-2017

Earning is in Learning
- Rajesh Jakhotia

About K2 Analytics

At K2 Analytics, we believe that skill development is very important for the growth of an individual, which in turn leads to the growth of Society & Industry and ultimately the Nation as a whole. For this it is important that access to knowledge and skill development trainings should be made available easily and economically to every individual.

Our Vision: *"To be the preferred partner for training and skill development"*

Our Mission: *"To provide training and skill development training to individuals, make them skilled & industry ready and create a pool of skilled resources readily available for the industry"*

*We have chosen Business Intelligence and Analytics as our focus area. With this endeavour we make this "**SAS Training**" accessible to all those who wish to learn SAS. We hope it is of help to you. For any feedback / suggestion feel free to write back to us at ar.jakhotia@k2analytics.co.in*

Welcome to Base SAS!!!

Introduction to SAS

What is SAS?
Basic Components of Base SAS
Base SAS GUI
What is Libname
SAS Language Components (Data Step, Proc Step, & Open Code)
SAS Dataset
Data types (String, Date, & Numeric)

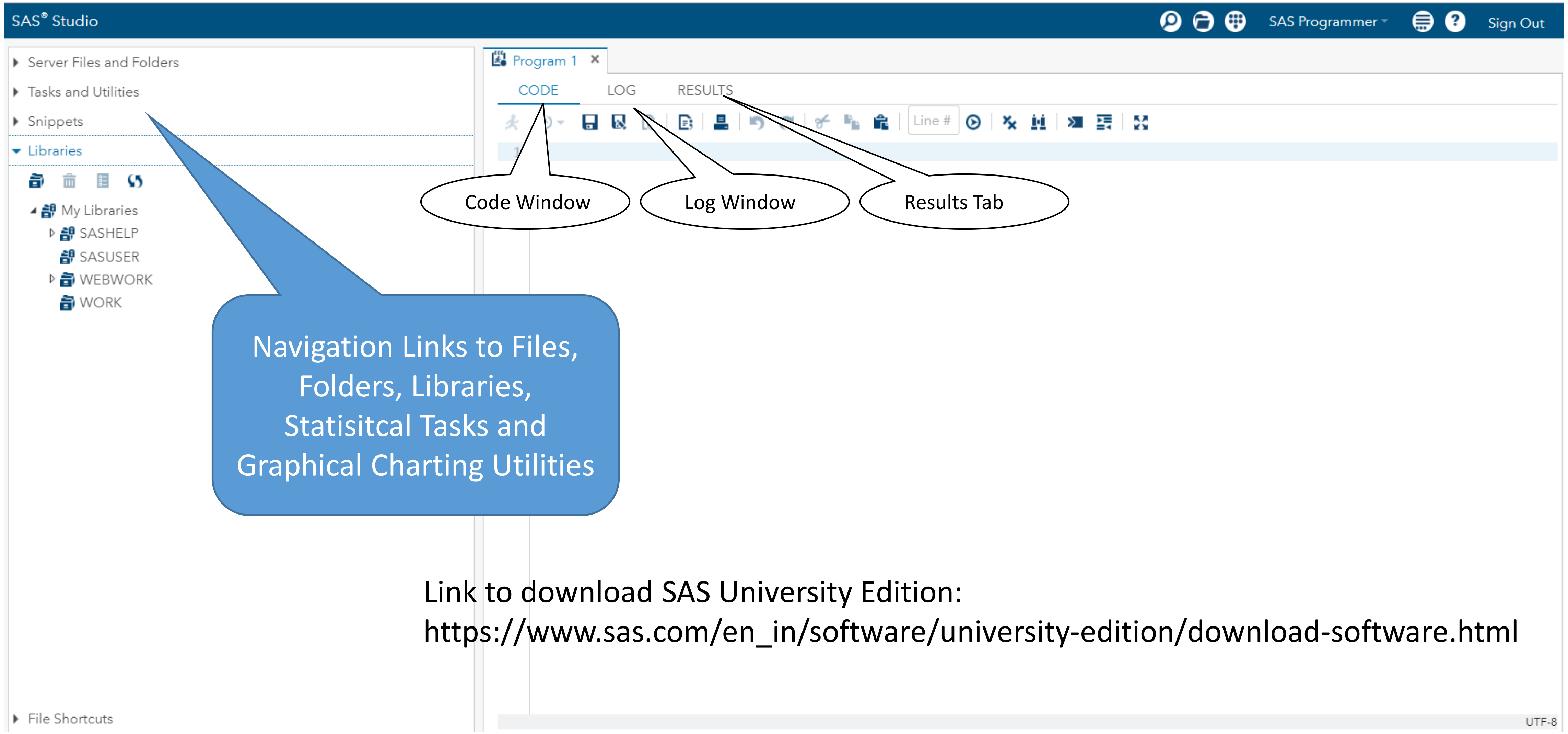
Earning is in Learning
- Rajesh Jakhotia

What is SAS?

SAS

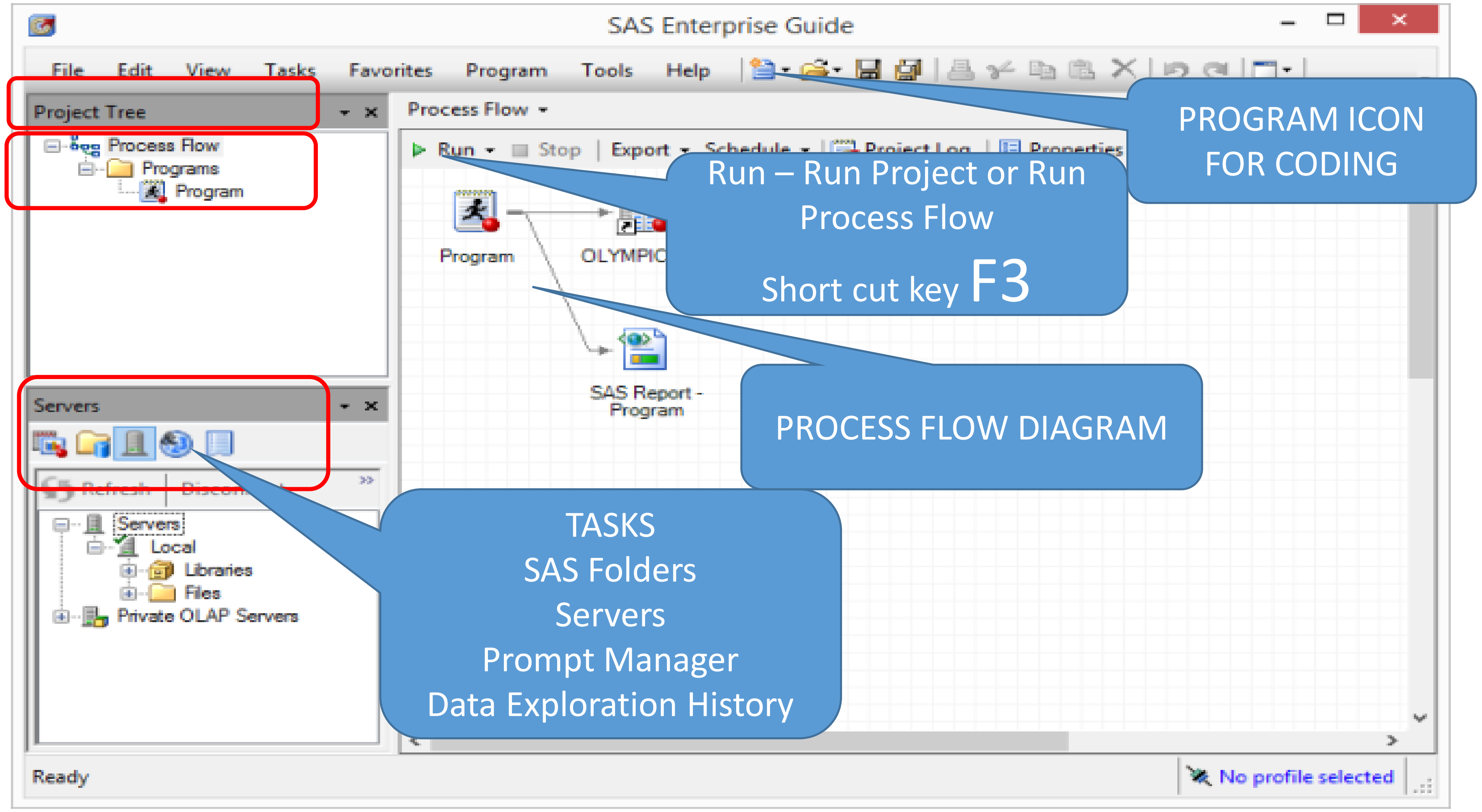
Statistical Analysis Software

SAS University Edition

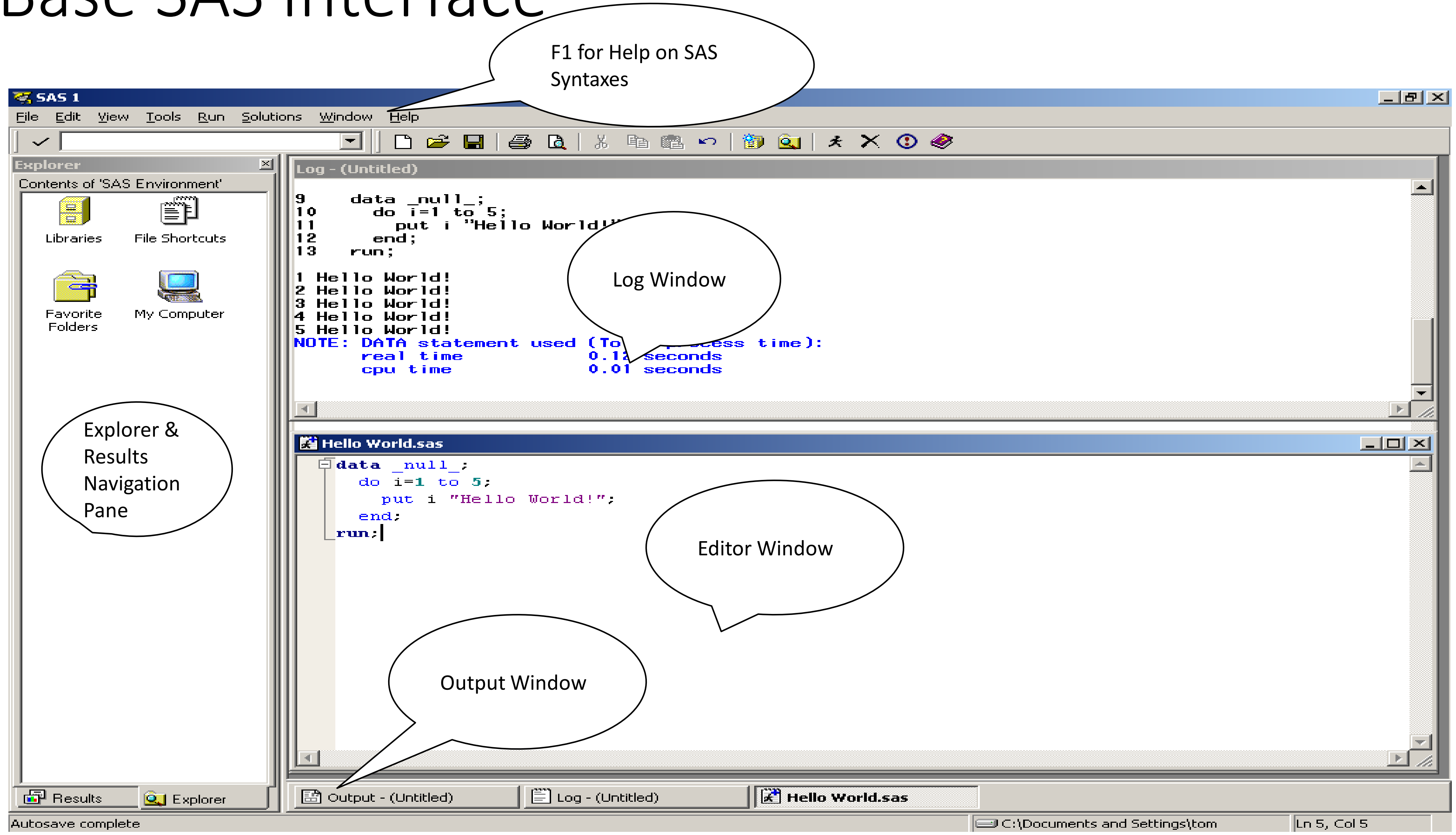


Link to download SAS University Edition:
https://www.sas.com/en_in/software/university-edition/download-software.html

SAS EG Interface



Base SAS Interface



```
9     data _null_;
10      do i=1 to 5;
11        put i "Hello World!";
12      end;
13    run;

1 Hello World!
2 Hello World!
3 Hello World!
4 Hello World!
5 Hello World!
NOTE: DATA statement used (Total process time):
      real time           0.12 seconds
      cpu time            0.01 seconds
```

```
data _null_;
  do i=1 to 5;
    put i "Hello World!";
  end;
run;
```

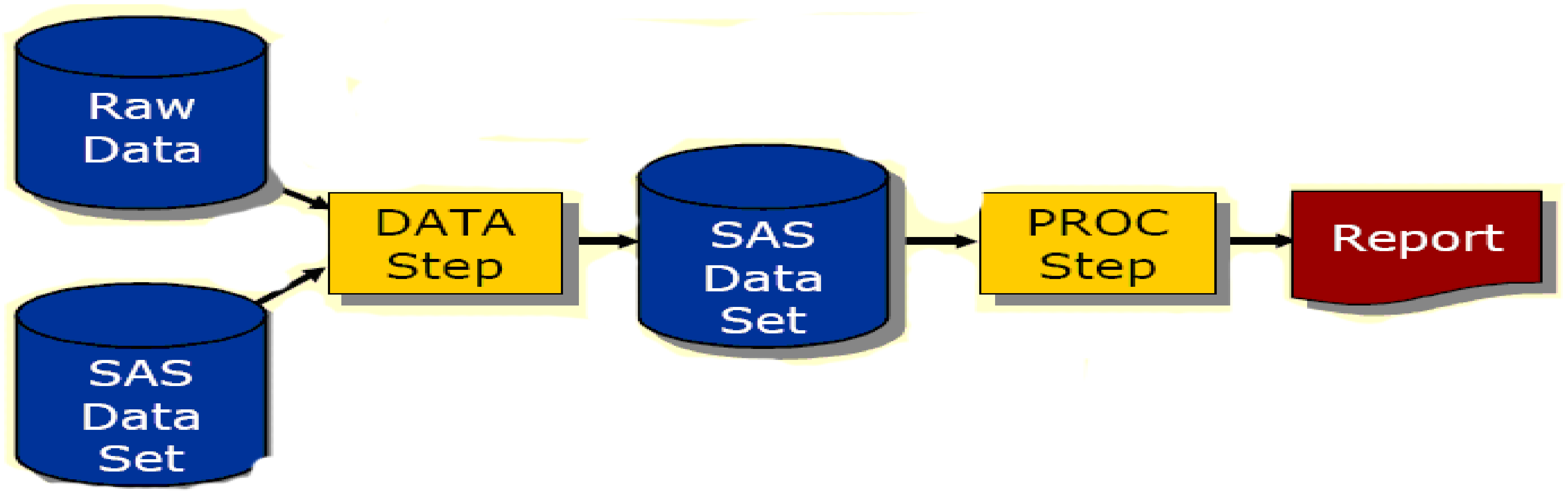
Key Components of SAS Program

- DATA step
  - Creates one or more SAS datasets
  - From existing SAS datasets or external sources

- PROC step
  - Performs various operations on SAS datasets
  - Used to create Data Summary and Reports
  - Usually does not operate on external sources

- Open Code
  - Any statement which is not part of a DATA step or PROC step

SAS Program Execution Flow Chart



- Note: SAS does Row by Row Processing

SAS Libname & Datasets

- Libname
  - A SAS data library is a collection of SAS files that are recognized as a unit by SAS
  - Types of Library – Temporary & Permanent
  - Work library is temporary library, when SAS is closed, all the datasets in the Work library are deleted; create a permanent SAS dataset via your own library

- SAS Dataset
  - SAS in-built format of storing data
  - Extension of SAS Dataset is .sas7bdat
  - Permanent Datasets are referred by “two level” name

SAS Variable Naming Rules

- Rules for SAS names
    - Names must be 32 characters or fewer in length
    - Names must start with a character or underscore
    - Names can contain only letters, numerals, or underscores (_) **No %$*@#,;**
    - Names are case insensitive on Windows Interface but sensitive on Unix interface

LIBNAME

- LIBNAME *libref "file-folder-location";*
- Rules for naming a libref:
  - The name must be 8 characters or less
  - The name must begin with a letter or underscore
  - The remaining characters must be letters, numbers or underscores.

```
LIBNAME myFirstLib "c:\training";
```

Wrong LibName as it is more than 8 characters

```
LIBNAME FirstLib "c:\training";
```

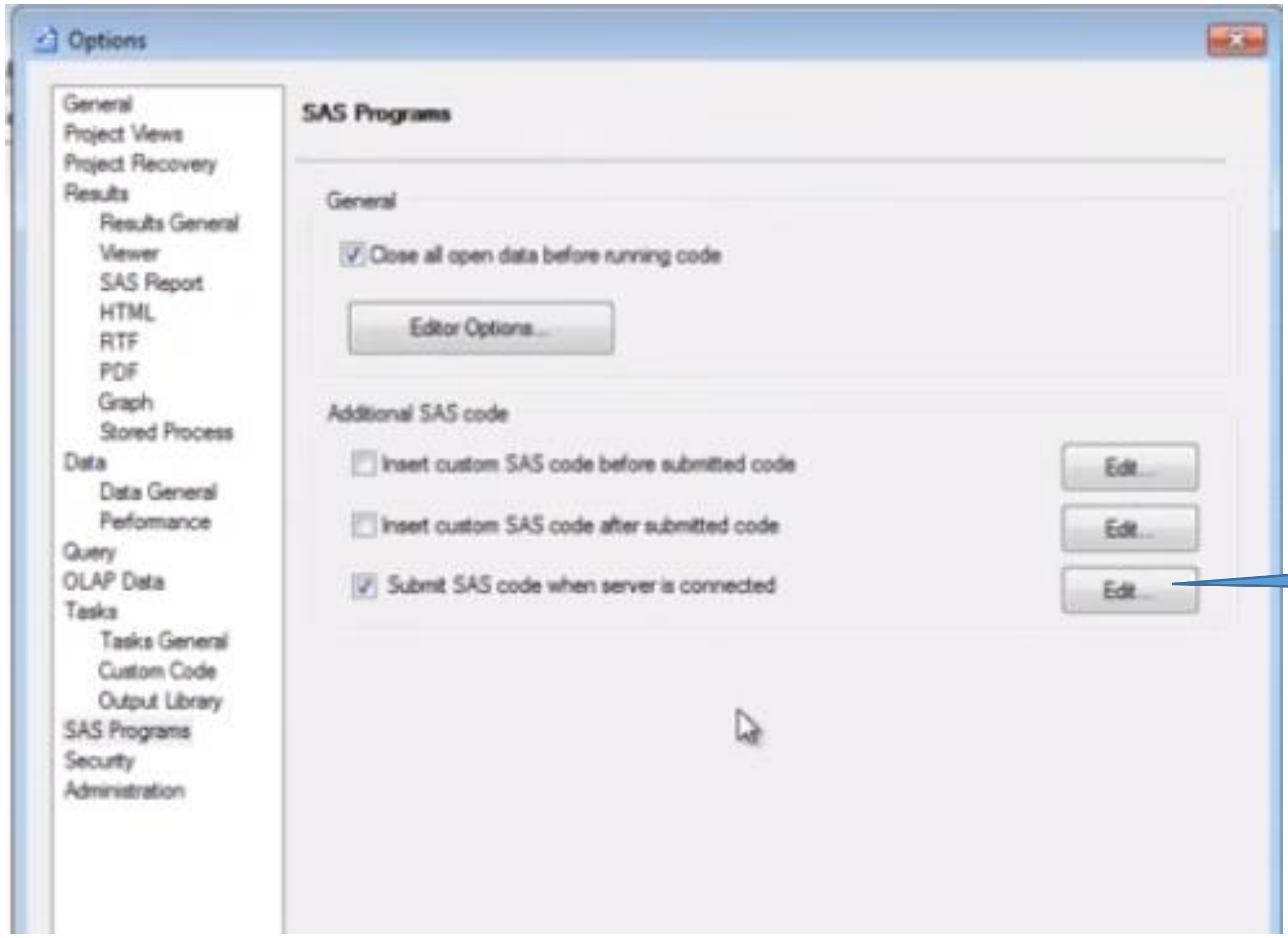
```
libname dst "/folders/myfolders/datasets/";
```

University Edition require the folder path be given relatively

SAS EG

TOOS > ASSIGN PROJECT LIBRARY

Autoexec Certain Code in SAS EG



SAS Programs

General

[✓] Close all open data before running code

Additional SAS code

[ ] Insert custom SAS code before submitted code

[ ] Insert custom SAS code after submitted code

[✓] Submit SAS code when server is connected

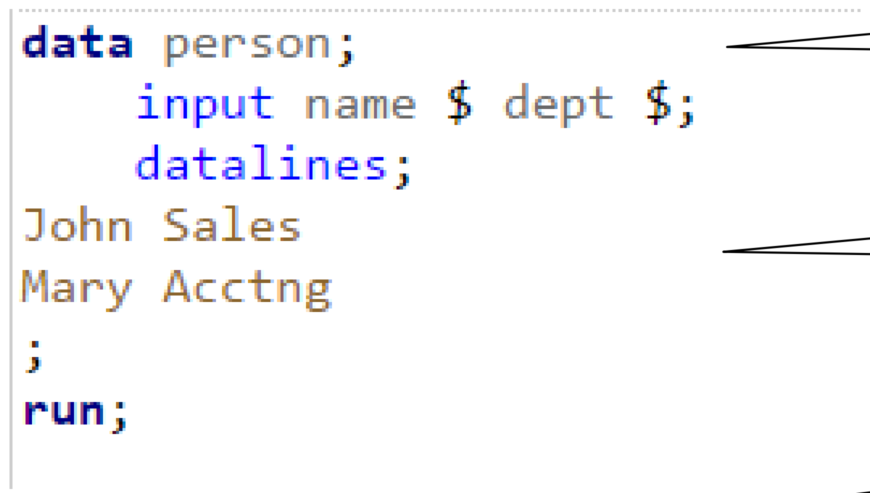
Tools > Global Options

Comments

- Comments in Programming Languages are used to explain the code
- Styles of commenting in SAS

```
/* Write you comment here */

/*
        This is a multiline comment
        Second line of the comment
        ........
*/

* Quick comment in single line... start with star and end with semi-colon;
```

Let us Create First Dataset



```
data person;
    input name $ dept $;
    datalines;
John Sales
Mary Acctng
;
run;
```

Step 1: Type the code on SAS Editor

Step 2: Select the code

Step 3: Click the run icon

Or use F3 short cut to execute the code

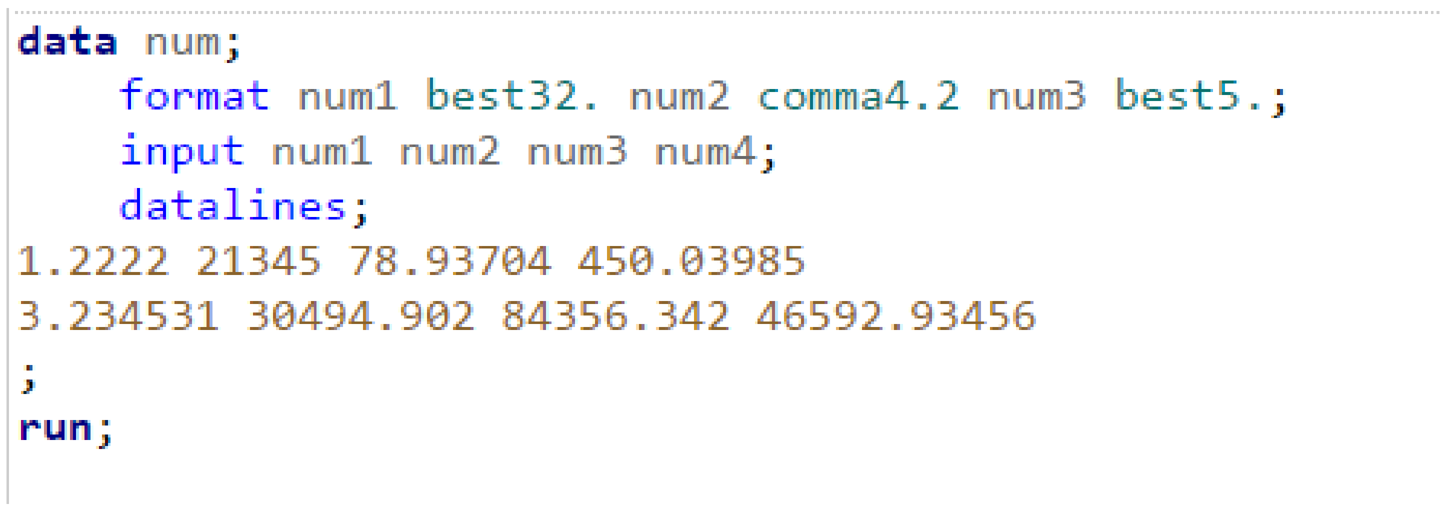
What-if the name and dept values are more than 8 characters…
Try putting some values which are greater than 8 characters???

Best Practices

- Always check the log after code execution
- In case of warnings assess the impact and proceed accordingly
  - Data Truncation – High impact
  - Divide by zero – In known situation Low impact
- In case of Error check the error message to identify cause
- Eyeball the data or output for its correctness
- Type of dataset (temp/permanent) should be decided based on the future usage/requirements of the dataset
- Follow standard and informative naming convention

Data Types - Numeric

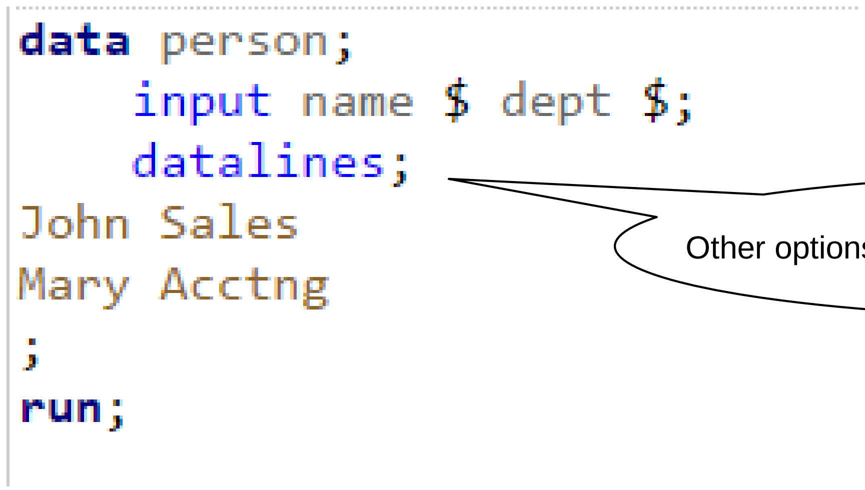
- Numeric variables contain numbers and are stored as doubles.



```
data num;
    format num1 best32. num2 comma4.2 num3 best5.;
    input num1 num2 num3 num4;
    datalines;
1.2222 21345 78.93704 450.03985
3.234531 30494.902 84356.342 46592.93456
;
run;
```

Data Types - String

- Data type that contain Character values are strings

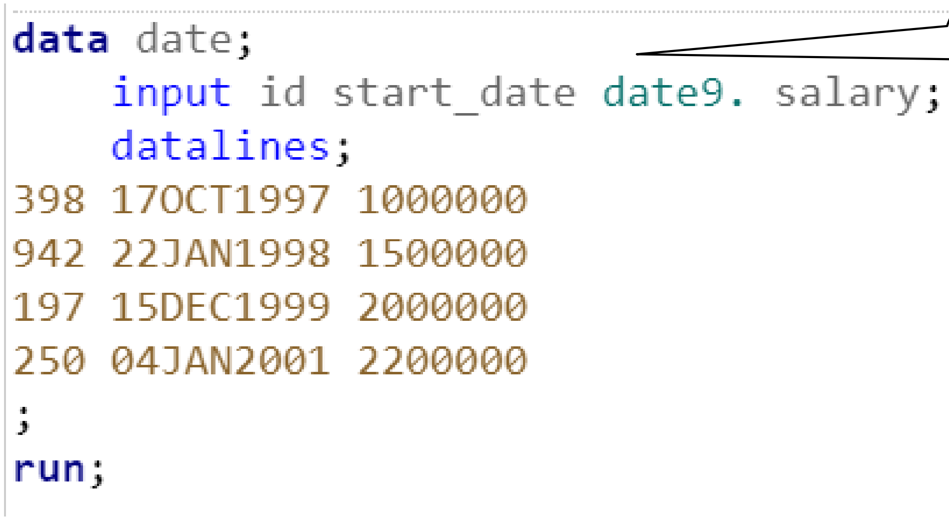


```
data person;
    input name $ dept $;
    datalines;
John Sales
Mary Acctng
;
run;
```

Other options in place of **datalines** is to use **cards** or **lines**

Data Types - Date

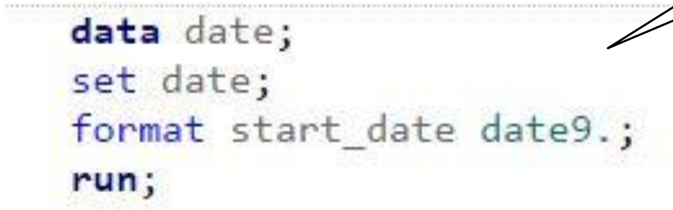
- Data type that contain date values



```
data date;
    input id start_date date9. salary;
    datalines;
398 17OCT1997 1000000
942 22JAN1998 1500000
197 15DEC1999 2000000
250 04JAN2001 2200000
;
run;
```

Execute the Code & see the dataset

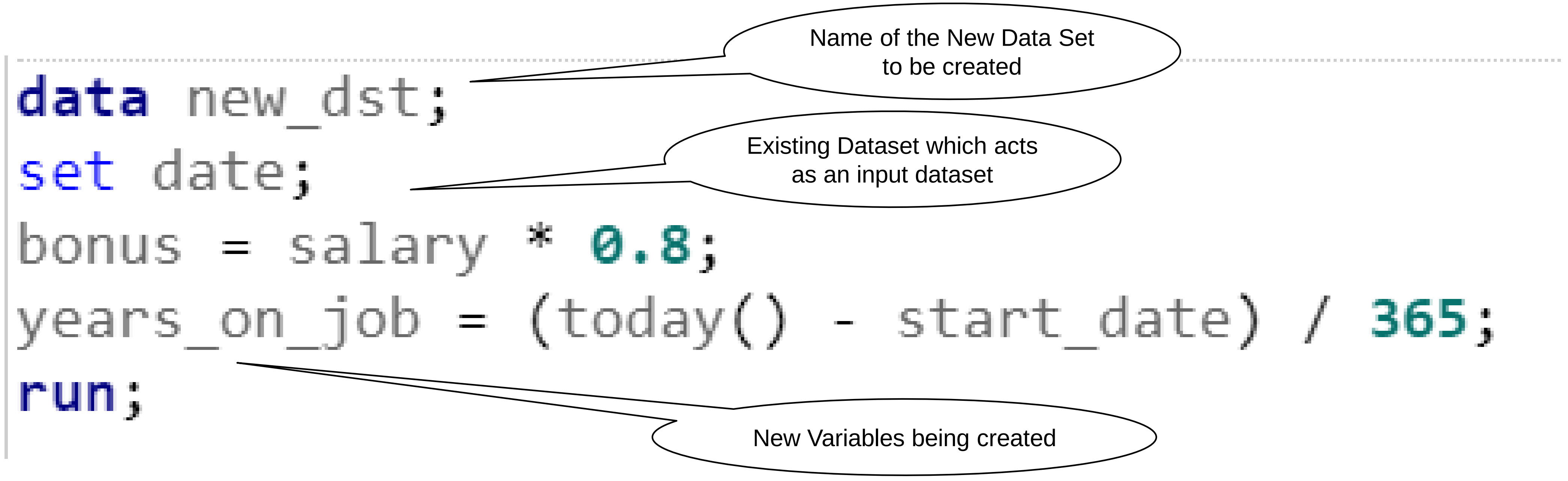
Execute the Code & once again see the dataset

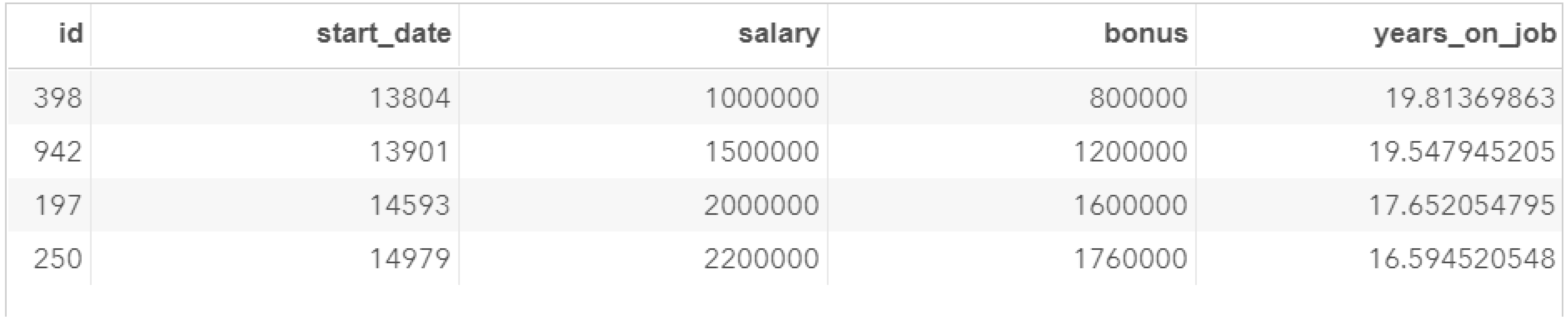


```
data date;
set date;
format start_date date9.;
run;
```

Note : In SAS, dates are measured as the number of days since January 1, 1960.

Creating New Variables


```
data new_dst;
set date;
bonus = salary * 0.8;
years_on_job = (today() - start_date) / 365;
run;
```
Name of the New Data Set to be created
Existing Dataset which acts as an input dataset
New Variables being created



id	start_date	salary	bonus	years_on_job
398	13804	1000000	800000	19.81369863
942	13901	1500000	1200000	19.547945205
197	14593	2000000	1600000	17.652054795
250	14979	2200000	1760000	16.594520548

QUIZ

Earning is in Learning
- Rajesh Jakhotia

Quiz

What does a Data step typically create?

Options

A) SAS Data Set

B) Summary Report

C) Work Library

D) Code File

Quiz

What does a Proc step typically create?

Options

A) Data Summary

B) Reports

C) Data Manipulation

D) Code File

Quiz

Where do you typically write the open code?

Options

A) In Data Step

B) In Proc Step

C) Outside Data Step and Proc Step

D) In Libname Syntax

Quiz

You closed and restarted your SAS Application? You have lost all the datasets created before restarting. You were storing datasets in which Library?

Options

A) Temporary Library

B) Permanent Library

C) Work Library

D) FirstLib Library

Quiz

Match the following

A) Extension of SAS Program	1) .sas7bdat
B) Extension of SAS Dataset	2) .egp
C) Extension of SAS Enterprise Guide Project	3) .sas
D) SAS Variable Name Length Limit	4) 32
E) SAS Libname Length Limit	5) 8

K2 Analytics
Building Skills, Building Individuals

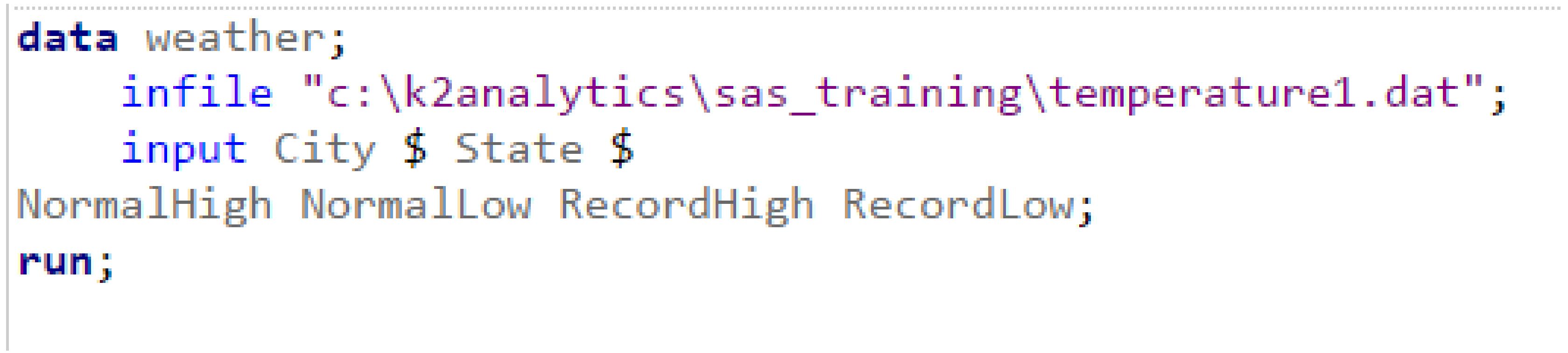
Getting Data in SAS

Using Data Step
Using Import Procedure
Reading tabular datafiles
Reading CSV files
Importing data from Excel
Usage of various options like FirstObs, Obs, MISSOVER

Earning is in Learning
- Rajesh Jakhotia

Using Data step

- Infile statement is used to import data from raw files



```
data weather;
    infile "c:\k2analytics\sas_training\temperature1.dat";
    input City $ State $
NormalHigh NormalLow RecordHigh RecordLow;
run;
```

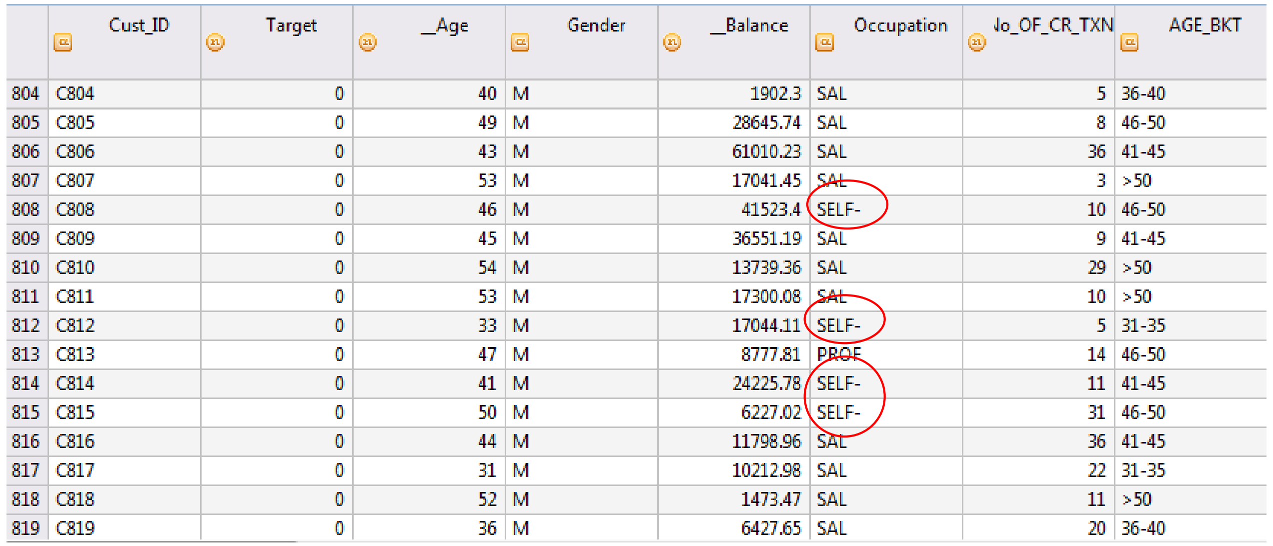
Note: File type can be in csv, txt, dat or any delimited file

Using Proc Import

```
PROC IMPORT DATAFILE="c:\k2analytics\sas_training\sample_files\LR_Sample.csv"
          out=LR1
          dbms=dlm
          replace;
     delimiter=',';
     getnames=yes;
run;
```

Execute the code;
Open and see the dataset ….

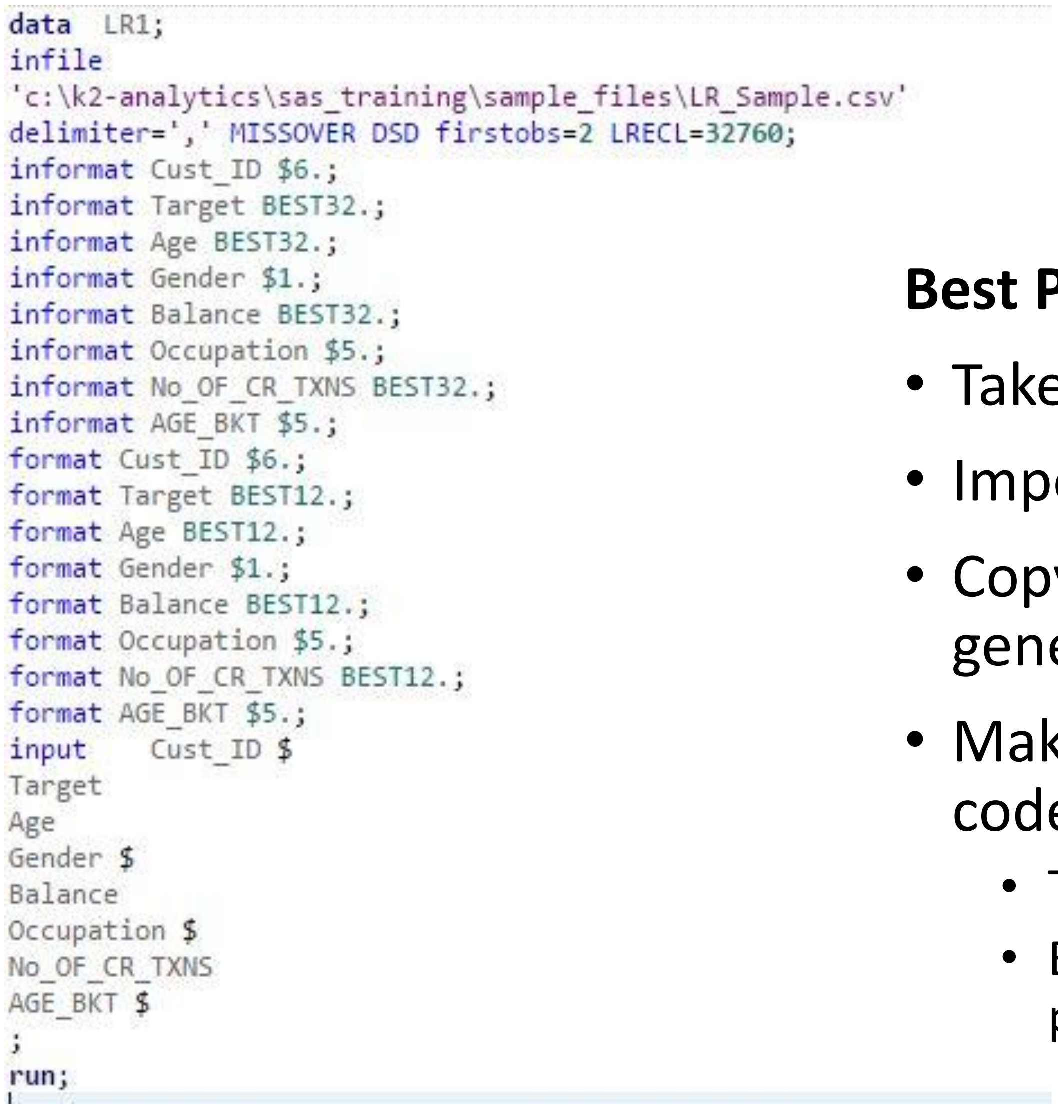
you will see data truncation occurring



	Cust_ID	Target	_Age	Gender	_Balance	Occupation	No_OF_CR_TXN	AGE_BKT
804	C804	0	40	M	1902.3	SAL	5	36-40
805	C805	0	49	M	28645.74	SAL	8	46-50
806	C806	0	43	M	61010.23	SAL	36	41-45
807	C807	0	53	M	17041.45	SAL	3	>50
808	C808	0	46	M	41523.4	SELF-	10	46-50
809	C809	0	45	M	36551.19	SAL	9	41-45
810	C810	0	54	M	13739.36	SAL	29	>50
811	C811	0	53	M	17300.08	SAL	10	>50
812	C812	0	33	M	17044.11	SELF-	5	31-35
813	C813	0	47	M	8777.81	PROF	14	46-50
814	C814	0	41	M	24225.78	SELF-	11	41-45
815	C815	0	50	M	6227.02	SELF-	31	46-50
816	C816	0	44	M	11798.96	SAL	36	41-45
817	C817	0	31	M	10212.98	SAL	22	31-35
818	C818	0	52	M	1473.47	SAL	11	>50
819	C819	0	36	M	6427.65	SAL	20	36-40

Why sometime data truncation happens in PROC IMPORT????

Proc Import… best practices to import a file



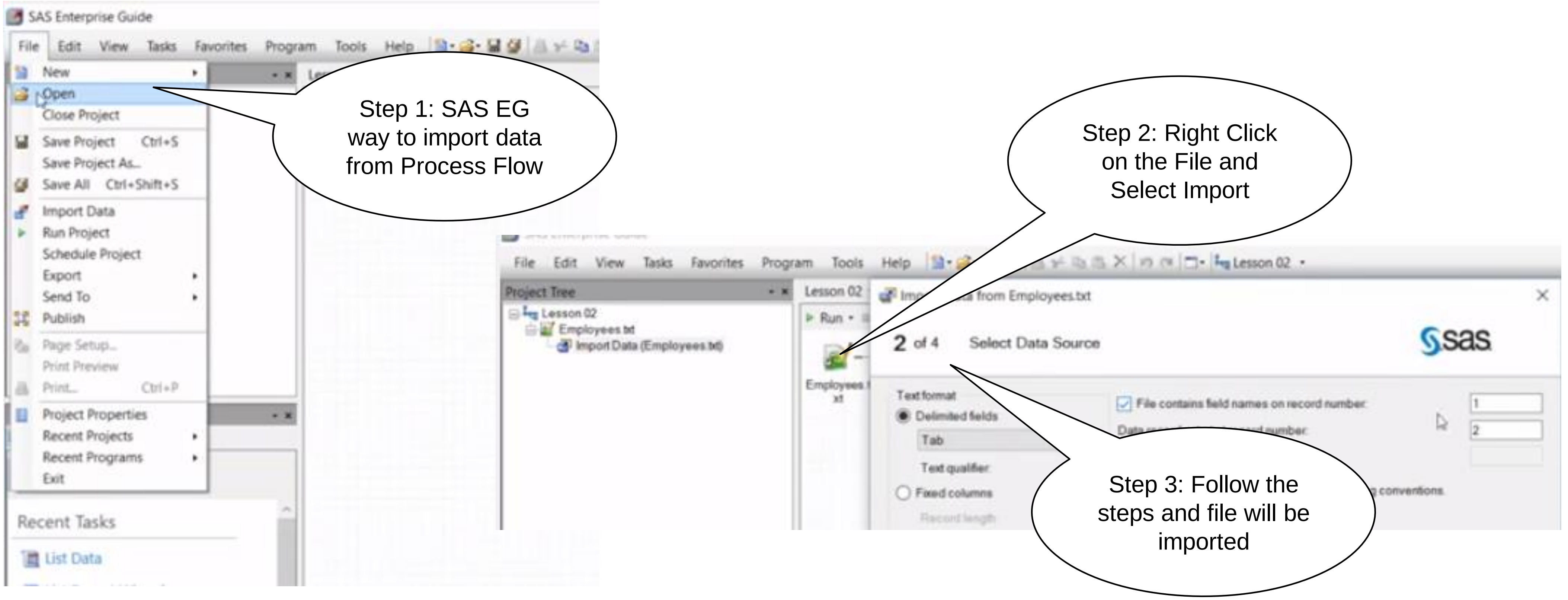
```
data  LR1;
infile
'c:\k2-analytics\sas_training\sample_files\LR_Sample.csv'
delimiter=',' MISSOVER DSD firstobs=2 LRECL=32760;
informat Cust_ID $6.;
informat Target BEST32.;
informat Age BEST32.;
informat Gender $1.;
informat Balance BEST32.;
informat Occupation $5.;
informat No_OF_CR_TXNS BEST32.;
informat AGE_BKT $5.;
format Cust_ID $6.;
format Target BEST12.;
format Age BEST12.;
format Gender $1.;
format Balance BEST12.;
format Occupation $5.;
format No_OF_CR_TXNS BEST12.;
format AGE_BKT $5.;
input    Cust_ID $
Target
Age
Gender $
Balance
Occupation $
No_OF_CR_TXNS
AGE_BKT $
;
run;
```

Best Practices:

- Take sample file of the Data File to be imported
- Import the sample file using PROC IMPORT
- Copy the DATA INFILE statement code generated in the log file
- Make necessary changes in the DATA INFILE code
  - Take special care to proper naming of the columns
  - Ensure the data type (Informat) of each column is properly defined else it may lead to data truncation

SAS EG Way of Importing

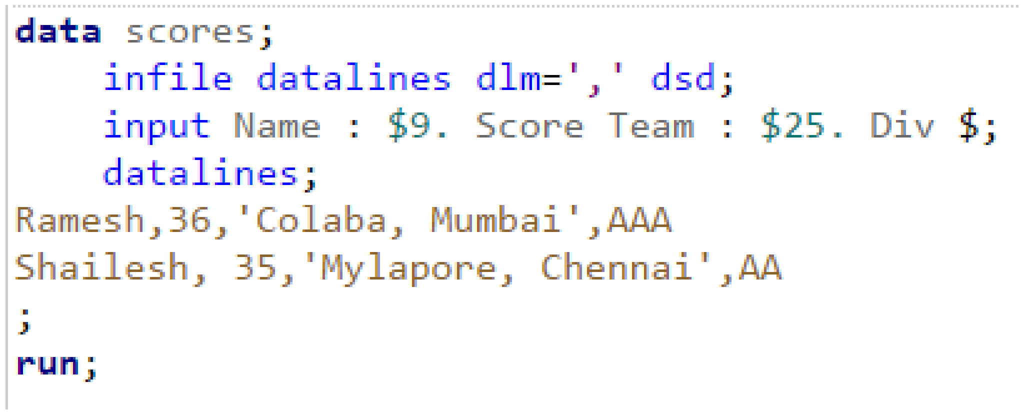
File > Open > Select the File from Folder
Right click on file in Process Flow Diagram …and follow the steps on screen and data is imported.



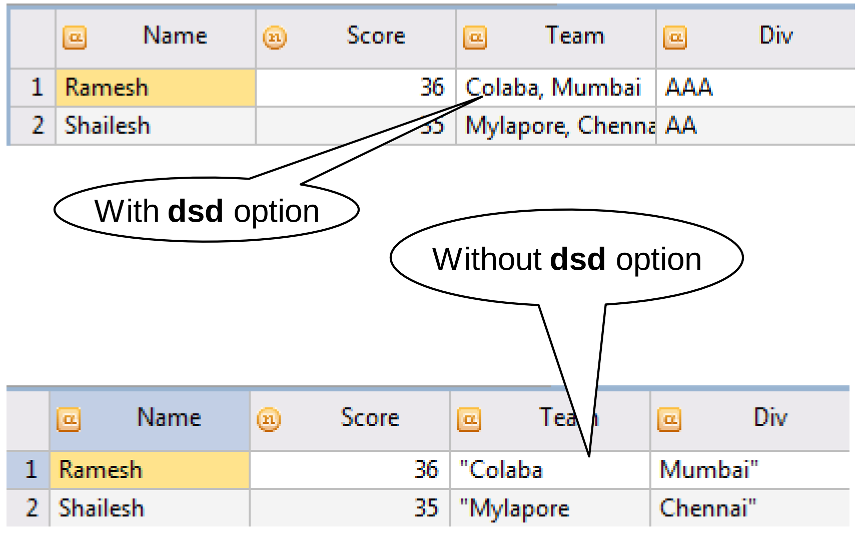
File contains field names on record number: 1
Data records start at record number: 2

DSD - Delimited Separated Data option

- Ignores delimiters in data values enclosed in quotes (especially useful while importing csv file having commas)
- Does not read quotes as part of the data value
- It treats two delimiters in a row as a missing value
- DSD assumes comma as the delimiter by default
- For delimiter other than comma use DLM option



```
data scores;
    infile datalines dlm=',' dsd;
    input Name : $9. Score Team : $25. Div $;
    datalines;
Ramesh,36,'Colaba, Mumbai',AAA
Shailesh, 35,'Mylapore, Chennai',AA
;
run;
```



With **dsd** option

	Name	Score	Team	Div
1	Ramesh	36	Colaba, Mumbai	AAA
2	Shailesh	35	Mylapore, Chenna	AA

Without **dsd** option

	Name	Score	Team	Div
1	Ramesh	36	"Colaba	Mumbai"
2	Shailesh	35	"Mylapore	Chennai"

Understanding MISSOVER Option

- Open the sample file "**LR_Sample_MISSOVER.csv**" provided to you

```
Cust_ID,Target, Age ,Gender, Balance ,Occupation,No_OF_CR_TXNS,AGE_BKT
C1,0,30,M,160378.6,SAL,2,26-30
C2,0,43,M,26275.55,PROF,23
C3,0,53,M,33616.47,SAL,45,>50
C4,0,45,M,1881.37,PROF,3
C5,0,37,M,3274.37,PROF,33,36-40
C6,0,41,M,197632.53,SAL,6,41-45
C7,1,46,M,36022.2,PROF,39,46-50
C8,1,33,M,81362.58,PROF,13,31-35
C9,1,43,M,70759.97,PROF,15,41-45
C10,1,41,M,84370.59,PROF,14,41-45
```

Note: For 2nd & 4th record the values for AGE_BKT columns are missing

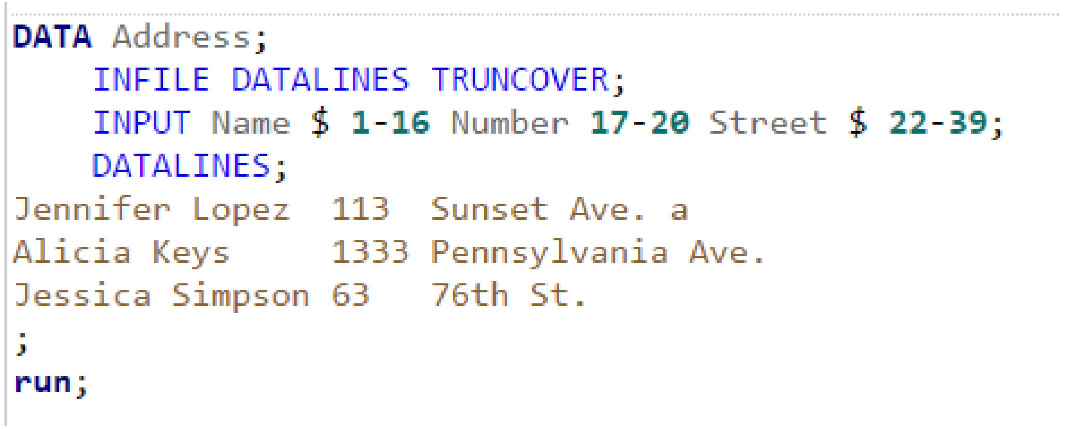
- Import the file using DATA INFILE statement
  - First without MISSOVER Option
  - Then with MISSOVER Option

Options in the INFILE Statement

- **FIRSTOBS:** This option tells SAS at what line to begin reading data. It's useful when the data file contains descriptive text or header information at the beginning
- **OBS:** This option can be used anytime you want to read only a part of your data file. It can be used with the FIRSTOBS= option to read lines from the middle of the file. For example, with FIRSTOBS=3 and OBS=5, SAS will start reading the data on the third line and stop reading after the fifth line.
- **MISSOVER**: By default, SAS will go to the next data line to read more data if SAS runs out of data and there are still more variables in the INPUT statement that have not been assigned values. The MISSOVER option tells SAS that if it reaches the end of the data line, don't go to the next data line. Instead, assign missing values to any remaining variables
- **TRUNCOVER**: The TRUNCOVER option is necessary when you are reading data using column or formatted input and some data lines are shorter than others. If a variable's field extends past the end of the data line, SAS will automatically go to the next line to start reading the variable's value. This option tells SAS to read data for the variable until it reaches the end of the data line, or the last column specified in the informat or column range, whichever comes first. For example,

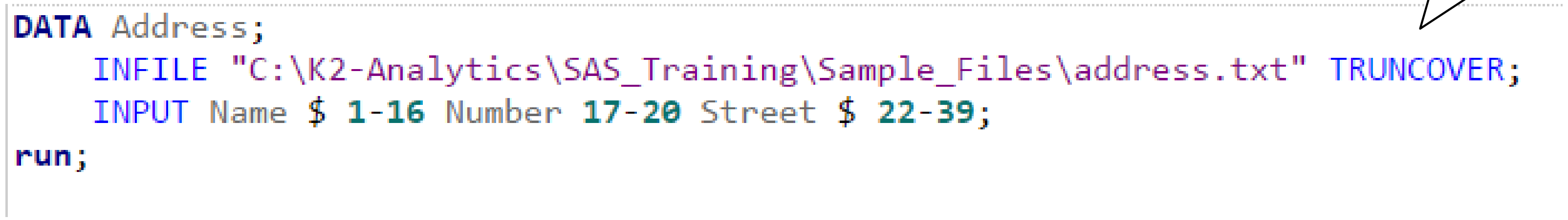
http://sites.stat.psu.edu/~xzhan/stat597c/sp04/Chapter2.htm

TRUNCOVER e.g.



```
DATA Address;
    INFILE DATALINES TRUNCOVER;
    INPUT Name $ 1-16 Number 17-20 Street $ 22-39;
    DATALINES;
Jennifer Lopez  113  Sunset Ave. a
Alicia Keys     1333 Pennsylvania Ave.
Jessica Simpson 63   76th St.
;
run;
```

Run this e.g. first with TRUNCOVER Option and then without TRUNCOVER Option

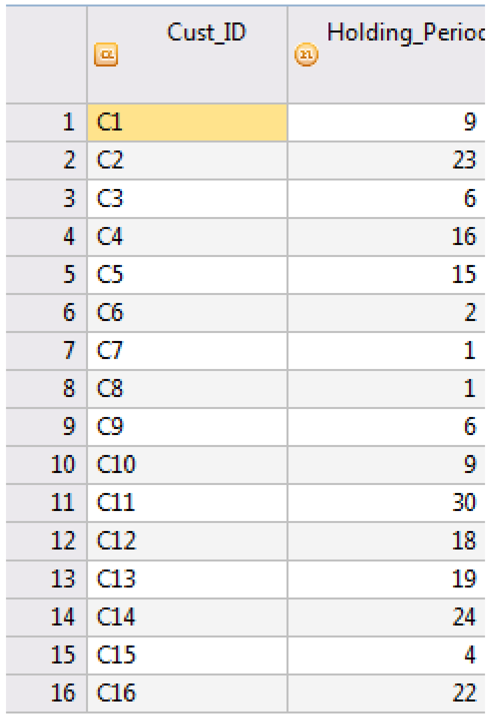


```
DATA Address;
    INFILE "C:\K2-Analytics\SAS_Training\Sample_Files\address.txt" TRUNCOVER;
    INPUT Name $ 1-16 Number 17-20 Street $ 22-39;
run;
```

Importing Excel files

```
PROC IMPORT datafile="C:\K2-Analytics\SAS_Training\Sample_Files\LR_Sample_Xls.xls"
    out=LR2
    dbms=excel
    replace;
    getnames=yes;
    sheet="HoldingPeriod";
run;
```

If **excel** option is not working try **xls**



	Cust_ID	Holding_Perioc
1	C1	9
2	C2	23
3	C3	6
4	C4	16
5	C5	15
6	C6	2
7	C7	1
8	C8	1
9	C9	6
10	C10	9
11	C11	30
12	C12	18
13	C13	19
14	C14	24
15	C15	4
16	C16	22

Note: Depending on the file type we may have to give different dbms options like **CSV, TAB, DLM, EXCEL2000, ACCESS**

Importing Fixed Width Format & TAB File

```
data LR3;
    infile "C:\K2-Analytics\SAS_Training\Sample_Files\LR_Sample_FWF.txt"
    FIRSTOBS=2;
    input CUST_ID $ 1-6 SCR 7-9 +2 DT_SCR DATE11.;
    FORMAT DT_SCR DATE9.;
run;
```

```
PROC IMPORT datafile="C:\K2-Analytics\SAS_Training\Sample_Files\LR_Sample_tab.txt"
     out=LR3
    dbms=tab
    replace;
    getnames=yes;
run;
```

After importing using PROC IMPORT follow the Best Practices Tip given in earlier slides

QUIZ

Earning is in Learning
- Rajesh Jakhotia

Quiz

From Which all Data Source(s) can you import data in SAS?

Options

A) Delimited Flat File

B) Fixed Width Format

C) Excel - Spreadsheets

D) Databases

E) XML

F) SPSS Data File

G) R Data

Quiz

Match the following

A) DSD	1) Line Number from where to start reading the data
B) MISSOVER	2) Line Number where to stop reading the data
C) FIRSTOBS	3) The format in which the values should be displayed
D) OBS	4) Ignore delimiters in data values enclosed in quotes
E) FORMAT	5) Read till end of the line is reached. Assigning MISSING value for any variables that are left
F) INFORMAT	6) The format in which the input values should be read and interpreted

Manipulating Data

Format & Informat
Data Merge
Data Sorting (Proc Sort)
Column KEEP – DROP Option
Relabeling the Column Names
Creating Multiple Datasets in Data Step
Reordering the Columns
Appending Data

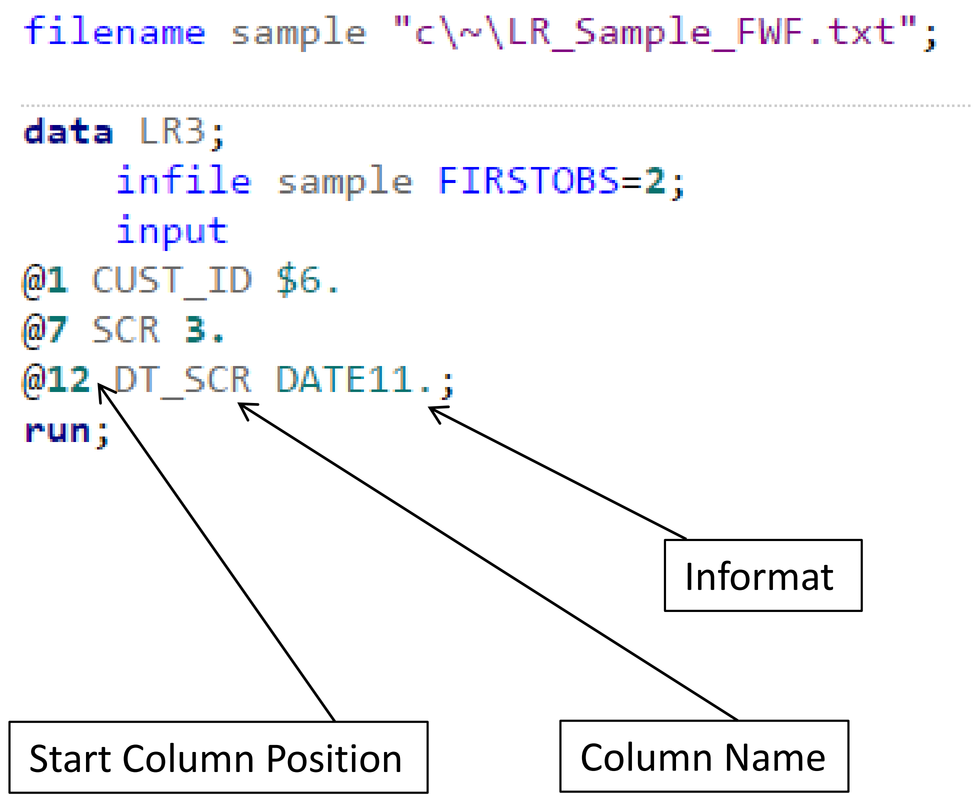
Earning is in Learning
- Rajesh Jakhotia

Informats

- Informats are used to tell SAS the format in which the input data should be read from external file
- Types of SAS Informats

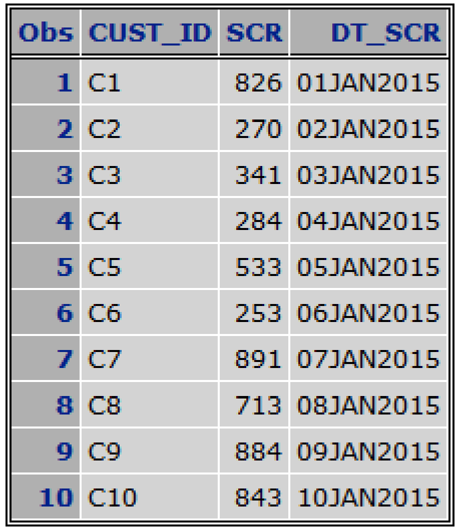
Informat Type	Informat Form	Informat Guidelines
Character	$INFORMATw.	$ indicates character format w indicates the width d the number of decimal places All informats must contain a dot so that SAS can differentiate an informat from variable name
Numeric	INFORMATw.d	
Date	INFORMATw.	

Using Informat for Import



```
filename sample "c\~\LR_Sample_FWF.txt";

data LR3;
    infile sample FIRSTOBS=2;
    input
@1 CUST_ID $6.
@7 SCR 3.
@12 DT_SCR DATE11.;
run;
```

```
proc print data = LR3;
run;
```



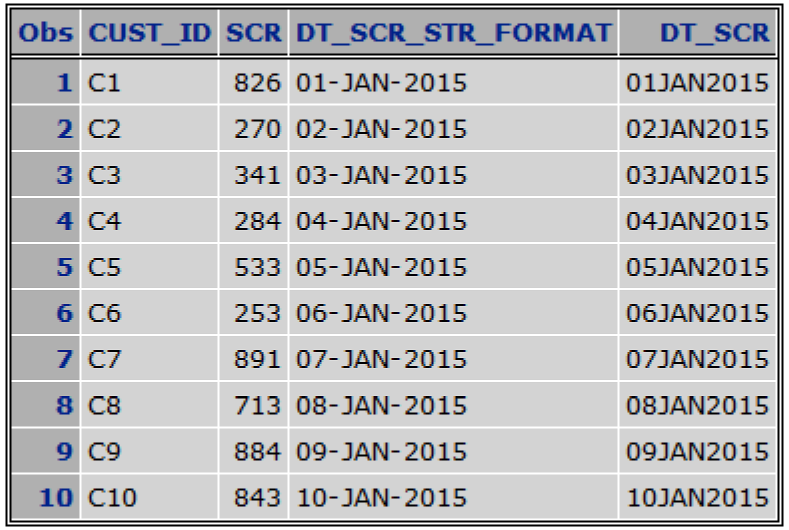
Obs	CUST_ID	SCR	DT_SCR
1	C1	826	01JAN2015
2	C2	270	02JAN2015
3	C3	341	03JAN2015
4	C4	284	04JAN2015
5	C5	533	05JAN2015
6	C6	253	06JAN2015
7	C7	891	07JAN2015
8	C8	713	08JAN2015
9	C9	884	09JAN2015
10	C10	843	10JAN2015

Using input() function to change formats

```
data LR3;
    infile sample
    FIRSTOBS=2;
    input
    @1 CUST_ID   $6.
    @7 SCR 3.
    @12 DT_SCR_STR_FORMAT    $11.
    ;
    DT_SCR=input(DT_SCR_STR_FORMAT, DATE11.);
    FORMAT DT_SCR DATE9.;
run;

proc print data=LR3;
run;
```

The INPUT() function is used for data type transformation

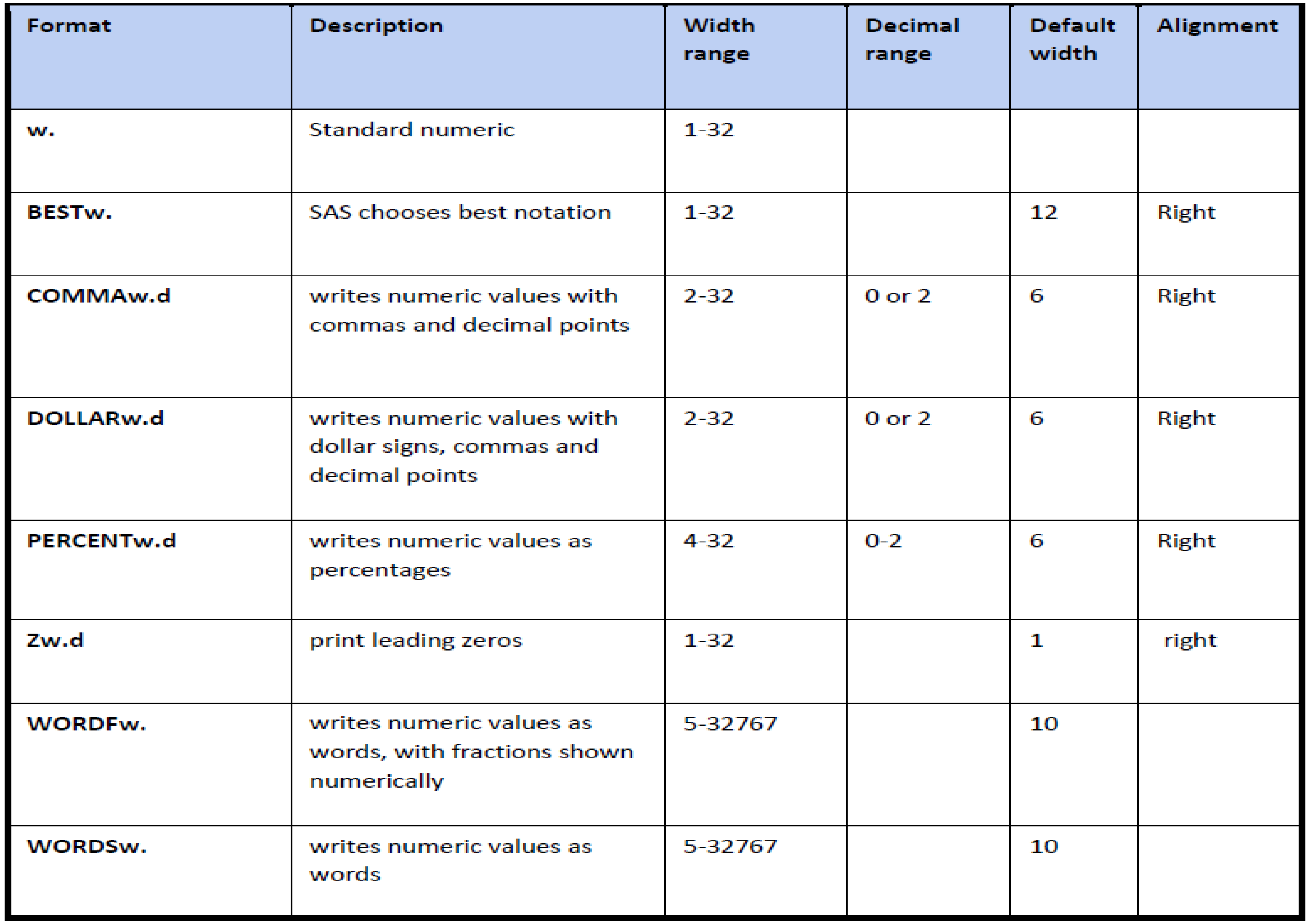
FORMAT are instructions for formatting the output data



Obs	CUST_ID	SCR	DT_SCR_STR_FORMAT	DT_SCR
1	C1	826	01-JAN-2015	01JAN2015
2	C2	270	02-JAN-2015	02JAN2015
3	C3	341	03-JAN-2015	03JAN2015
4	C4	284	04-JAN-2015	04JAN2015
5	C5	533	05-JAN-2015	05JAN2015
6	C6	253	06-JAN-2015	06JAN2015
7	C7	891	07-JAN-2015	07JAN2015
8	C8	713	08-JAN-2015	08JAN2015
9	C9	884	09-JAN-2015	09JAN2015
10	C10	843	10-JAN-2015	10JAN2015

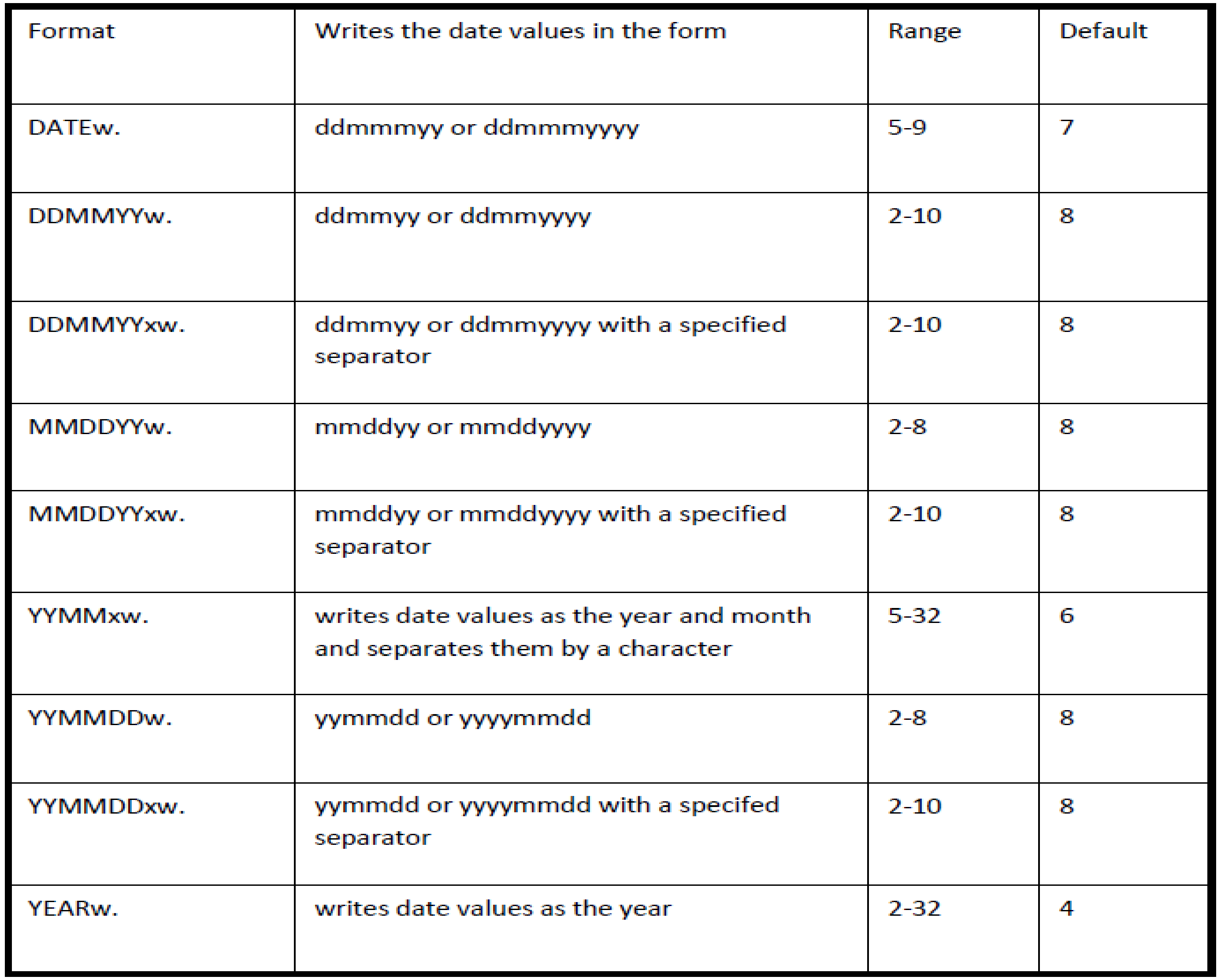
Note: In place of FORMAT we could have used put() function
DT_SCR = put(input(DT_SCR_STR_FORMAT, DATE11.), DATE9.)

Some Numeric Formats



Format	Description	Width range	Decimal range	Default width	Alignment
w.	Standard numeric	1-32			
BESTw.	SAS chooses best notation	1-32		12	Right
COMMAw.d	writes numeric values with commas and decimal points	2-32	0 or 2	6	Right
DOLLARw.d	writes numeric values with dollar signs, commas and decimal points	2-32	0 or 2	6	Right
PERCENTw.d	writes numeric values as percentages	4-32	0-2	6	Right
Zw.d	print leading zeros	1-32		1	right
WORDFw.	writes numeric values as words, with fractions shown numerically	5-32767		10	
WORDSw.	writes numeric values as words	5-32767		10	

Some Date Formats



Format	Writes the date values in the form	Range	Default
DATEw.	ddmmmyy or ddmmmyyyy	5-9	7
DDMMYYw.	ddmmyy or ddmmyyyy	2-10	8
DDMMYYxw.	ddmmyy or ddmmyyyy with a specified separator	2-10	8
MMDDYYw.	mmddyy or mmddyyyy	2-8	8
MMDDYYxw.	mmddyy or mmddyyyy with a specified separator	2-10	8
YYMMxw.	writes date values as the year and month and separates them by a character	5-32	6
YYMMDDw.	yymmdd or yyyymmdd	2-8	8
YYMMDDxw.	yymmdd or yyyymmdd with a specifed separator	2-10	8
YEARw.	writes date values as the year	2-32	4

Understanding Data Merge Concept

- DATA MERGE is used when you have to join data from 2 or more datasets
- Types of Merge
  - Naïve merge (it is not a match merge)
  - One-to-one match merge
  - One-to-many match merge
  - Many-to-many match merge
- SAS Syntax for merge

DATA *new-data-set*;

MERGE *data-set-1 data-set-2 data-set-3* …;

BY *by-variable(s)*; */* indicates the variable(s) that control which observations to match */*

RUN;

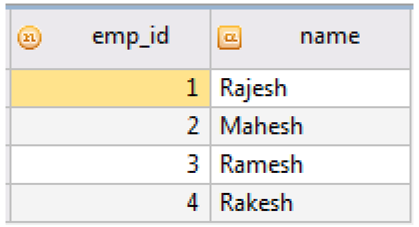
Naïve Merge

```
data emp;
    input emp_id name$;
    datalines;
1 Rajesh
2 Mahesh
3 Ramesh
4 Rakesh
;
```

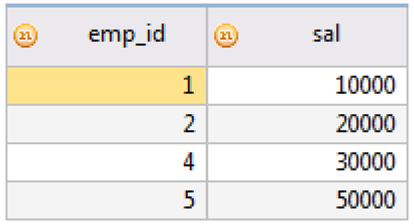
```
data emp_sal;
    input emp_id sal;
    datalines;
1 10000
2 20000
4 30000
5 50000
;
run;
```

```
data emp_dtl_fo;
merge emp emp_sal;
run;
```

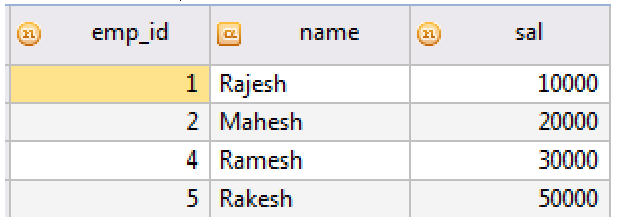
In merge we have not specified the BY clause and this has led to data corruption



emp_id	name
1	Rajesh
2	Mahesh
3	Ramesh
4	Rakesh



emp_id	sal
1	10000
2	20000
4	30000
5	50000



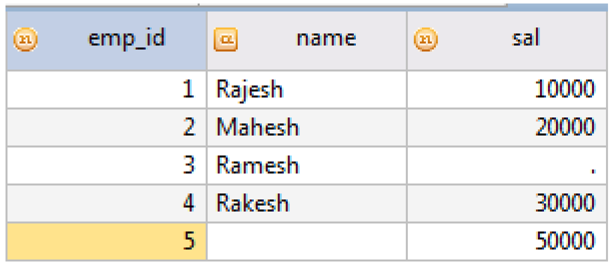
emp_id	name	sal
1	Rajesh	10000
2	Mahesh	20000
4	Ramesh	30000
5	Rakesh	50000

One-to-One merge

- Full Outer Join

(Note the usage of ***by*** option)

```
data emp_dtl_fo;
    merge emp emp_sal;
    by emp_id;
run;
```

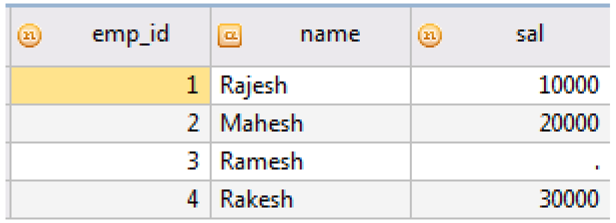


emp_id	name	sal
1	Rajesh	10000
2	Mahesh	20000
3	Ramesh	.
4	Rakesh	30000
5		50000

- Left Outer Join

(Note the usage of ***in*** & ***if*** option)

```
data emp_dtl_fo;
    merge emp(in=aa) emp_sal(in=bb);
    by emp_id;
    if aa;
run;
```

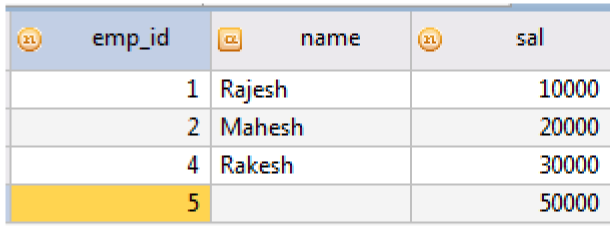


emp_id	name	sal
1	Rajesh	10000
2	Mahesh	20000
3	Ramesh	.
4	Rakesh	30000

- Right Outer Join

(Note the usage of ***in*** & ***if*** option)

```
data emp_dtl_fo;
    merge emp(in=aa) emp_sal(in=bb);
    by emp_id;
    if bb;
run;
```



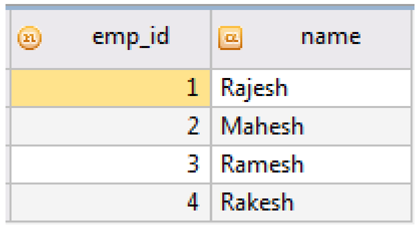
emp_id	name	sal
1	Rajesh	10000
2	Mahesh	20000
4	Rakesh	30000
5		50000

One-to-Many merge

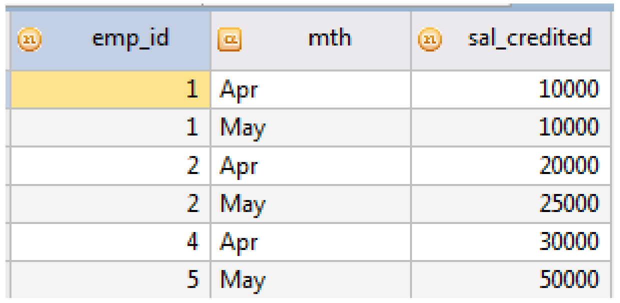
```
data emp;
    input emp_id name$;
    datalines;
1 Rajesh
2 Mahesh
3 Ramesh
4 Rakesh
;
```

```
data emp_mth_sal;
    input emp_id mth $ sal_credited;
    datalines;
1 Apr 10000
1 May 10000
2 Apr 20000
2 May 20000
4 Apr 30000
5 May 50000
;
run;
```

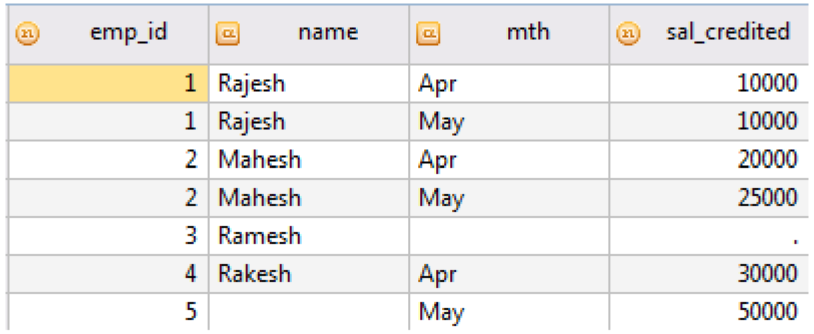
```
data emp_mth_sal_fo;
    merge emp emp_mth_sal;
    by emp_id;
run;
```



emp_id	name
1	Rajesh
2	Mahesh
3	Ramesh
4	Rakesh



emp_id	mth	sal_credited
1	Apr	10000
1	May	10000
2	Apr	20000
2	May	25000
4	Apr	30000
5	May	50000



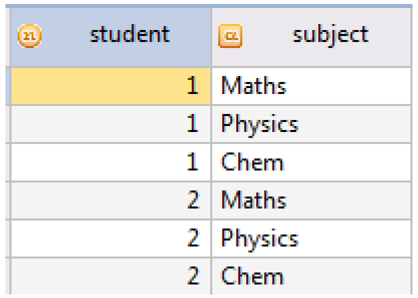
emp_id	name	mth	sal_credited
1	Rajesh	Apr	10000
1	Rajesh	May	10000
2	Mahesh	Apr	20000
2	Mahesh	May	25000
3	Ramesh		.
4	Rakesh	Apr	30000
5		May	50000

Many-to-Many merge

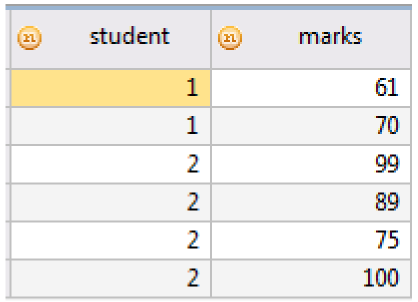
```
data subject;
    input student subject $;
    datalines;
1  Maths
1  Physics
1  Chem
2  Maths
2  Physics
2  Chem
;
run;
```

```
data marks_dst;
    input student marks;
    datalines;
1  61
1  70
2  99
2  89
2  75
2  100
;
run;
```

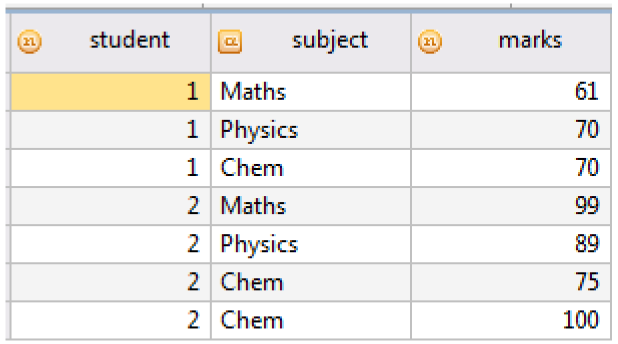
```
data sub_marks;
    merge subject marks_dst;
    by student;
run;
```



student	subject
1	Maths
1	Physics
1	Chem
2	Maths
2	Physics
2	Chem



student	marks
1	61
1	70
2	99
2	89
2	75
2	100

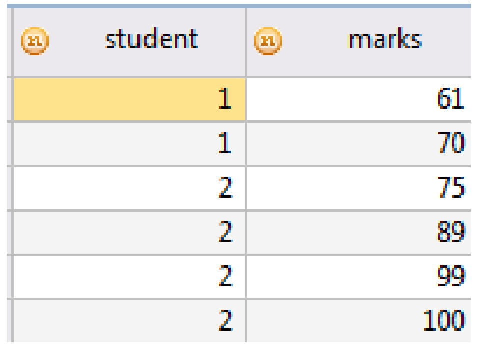


student	subject	marks
1	Maths	61
1	Physics	70
1	Chem	70
2	Maths	99
2	Physics	89
2	Chem	75
2	Chem	100

Data Sort

- Let us sort the "**marks_dst**" dataset by the field **marks**

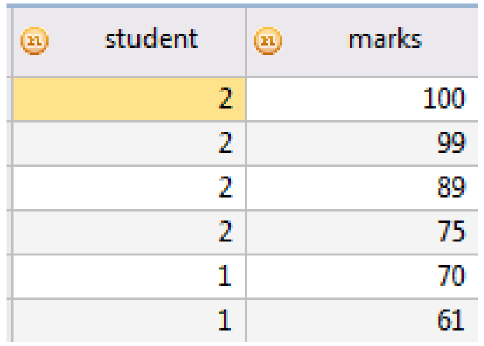
```
proc sort data=marks_dst;
     by marks;
run;
```



student	marks
1	61
1	70
2	75
2	89
2	99
2	100

- To sort by **marks** in descending order

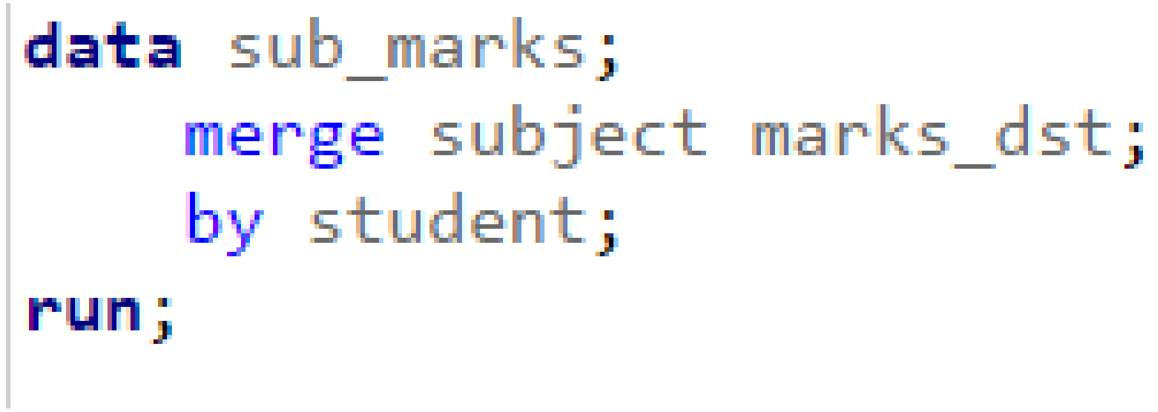
```
proc sort data=marks_dst;
     by descending marks;
run;
```



student	marks
2	100
2	99
2	89
2	75
1	70
1	61

Data Sort & Merge

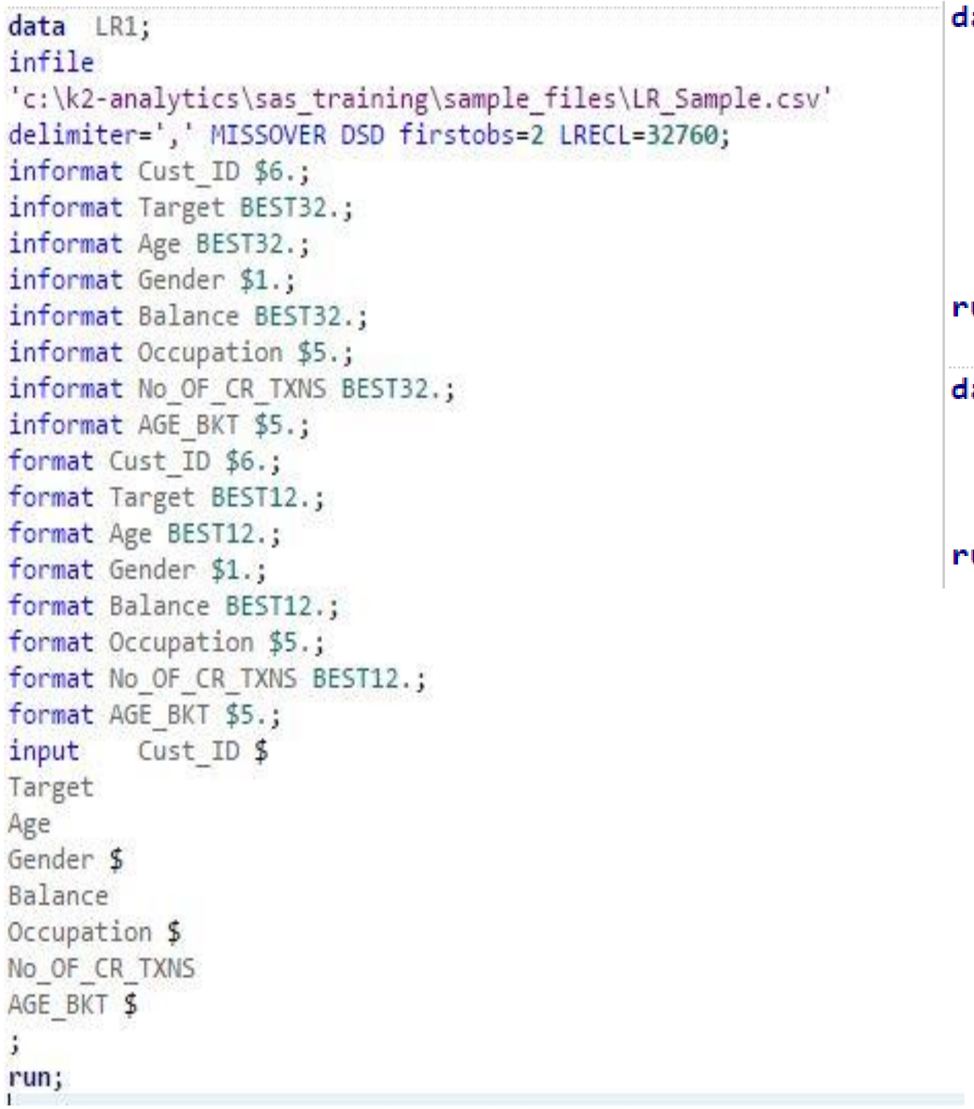
- Let us try running our previous merge code on the data sorted by **marks** field



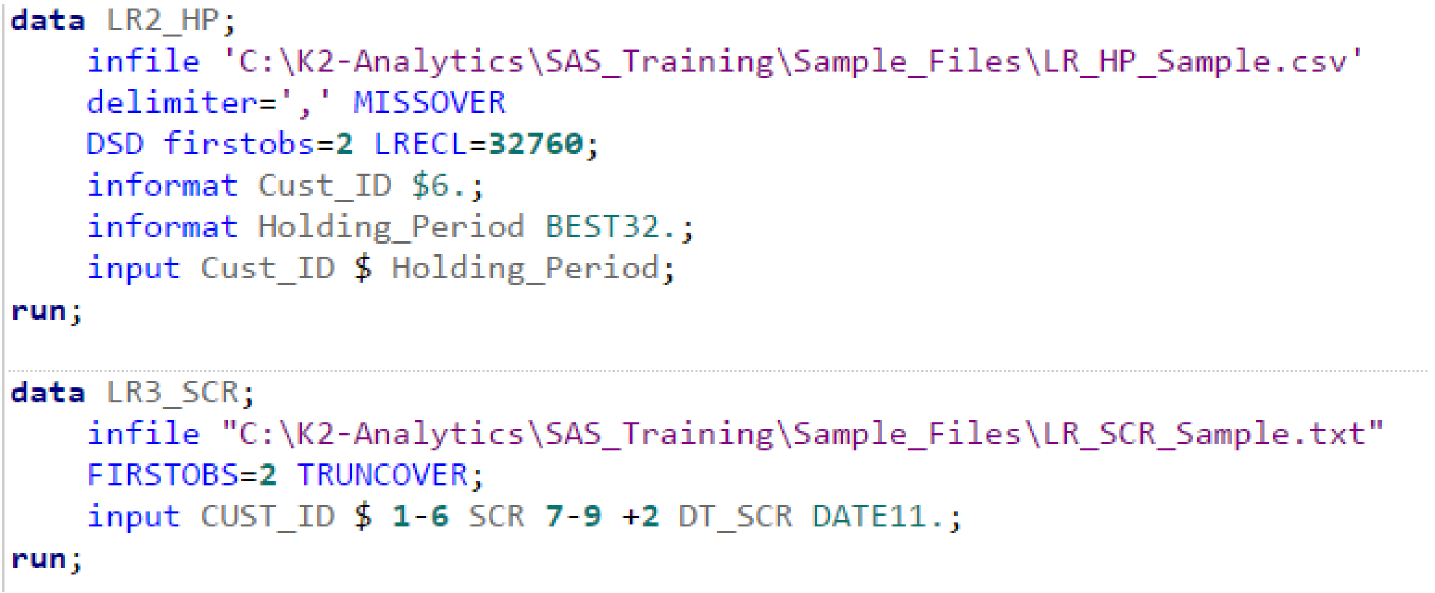
```
data sub_marks;
     merge subject marks_dst;
     by student;
run;
```

Error: The merge BY variables are not properly sorted

Let us create the datasets to be merged



```
data  LR1;
infile
'c:\k2-analytics\sas_training\sample_files\LR_Sample.csv'
delimiter=',' MISSOVER DSD firstobs=2 LRECL=32760;
informat Cust_ID $6.;
informat Target BEST32.;
informat Age BEST32.;
informat Gender $1.;
informat Balance BEST32.;
informat Occupation $5.;
informat No_OF_CR_TXNS BEST32.;
informat AGE_BKT $5.;
format Cust_ID $6.;
format Target BEST12.;
format Age BEST12.;
format Gender $1.;
format Balance BEST12.;
format Occupation $5.;
format No_OF_CR_TXNS BEST12.;
format AGE_BKT $5.;
input    Cust_ID $
Target
Age
Gender $
Balance
Occupation $
No_OF_CR_TXNS
AGE_BKT $
;
run;
```



```
data LR2_HP;
    infile 'C:\K2-Analytics\SAS_Training\Sample_Files\LR_HP_Sample.csv'
    delimiter=',' MISSOVER
    DSD firstobs=2 LRECL=32760;
    informat Cust_ID $6.;
    informat Holding_Period BEST32.;
    input Cust_ID $ Holding_Period;
run;

data LR3_SCR;
    infile "C:\K2-Analytics\SAS_Training\Sample_Files\LR_SCR_Sample.txt"
    FIRSTOBS=2 TRUNCOVER;
    input CUST_ID $ 1-6 SCR 7-9 +2 DT_SCR DATE11.;
run;
```

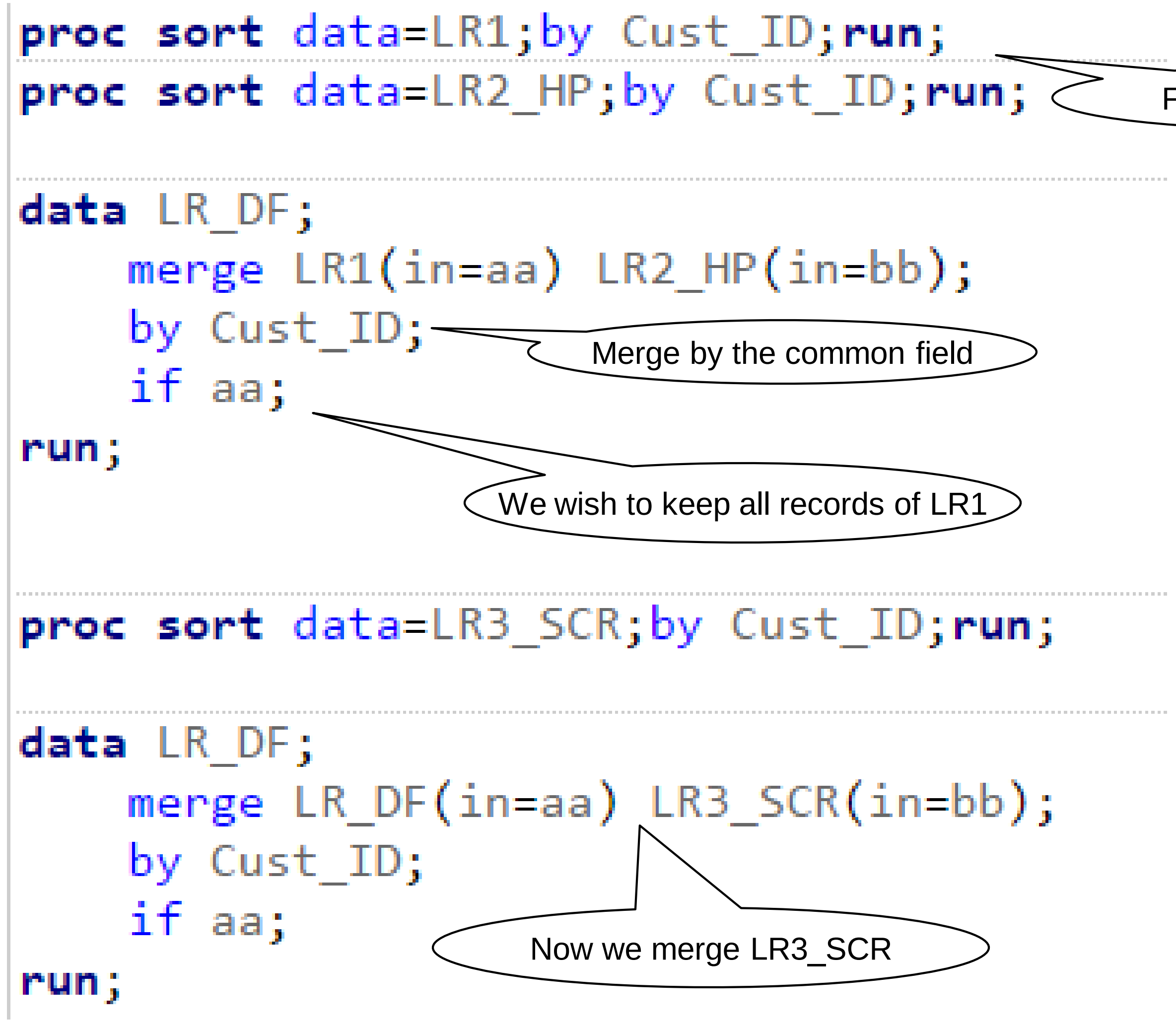
Let us create the datasets to be merged

```
data  LR1;
infile
'c:\k2-analytics\sas_training\sample_files\LR_Sample.csv'
delimiter=',' MISSOVER DSD firstobs=2 LRECL=32760;
informat Cust_ID $6.;
informat Target BEST32.;
informat Age BEST32.;
informat Gender $1.;
informat Balance BEST32.;
informat Occupation $5.;
informat No_OF_CR_TXNS BEST32.;
informat AGE_BKT $5.;
format Cust_ID $6.;
format Target BEST12.;
format Age BEST12.;
format Gender $1.;
format Balance BEST12.;
format Occupation $5.;
format No_OF_CR_TXNS BEST12.;
format AGE_BKT $5.;
input    Cust_ID $
Target
Age
Gender $
Balance
Occupation $
No_OF_CR_TXNS
AGE_BKT $
;
run;
```

```
data  LR2_HP;
infile 'C:\K2-
Analytics\SAS_Training\Sample_Files\LR_HP_Sample.csv'
delimiter=',' MISSOVER DSD firstobs=2 LRECL=32760;
informat Cust_ID $6.;
informat Holding_Period BEST32.;
input    Cust_ID $
Holding_Period
;
run;
```

```
data LR3_SCR;
infile "C:\K2-
Analytics\SAS_Training\Sample_Files\LR_SCR_Sample.txt"
FIRSTOBS  = 2 TRUNCOVER;
input CUST_ID $ 1-6 SCR 7-9 +2 DT_SCR DATE11.;
run;
```

Write code to merge the datasets



```
proc sort data=LR1;by Cust_ID;run;
proc sort data=LR2_HP;by Cust_ID;run;

data LR_DF;
    merge LR1(in=aa) LR2_HP(in=bb);
    by Cust_ID;
    if aa;
run;

proc sort data=LR3_SCR;by Cust_ID;run;

data LR_DF;
    merge LR_DF(in=aa) LR3_SCR(in=bb);
    by Cust_ID;
    if aa;
run;
```

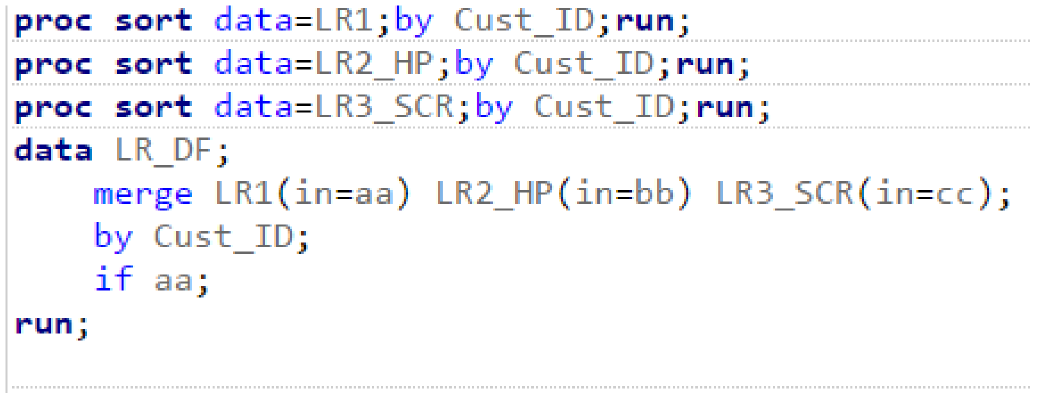
First sort the dataset

Merge by the common field

We wish to keep all records of LR1

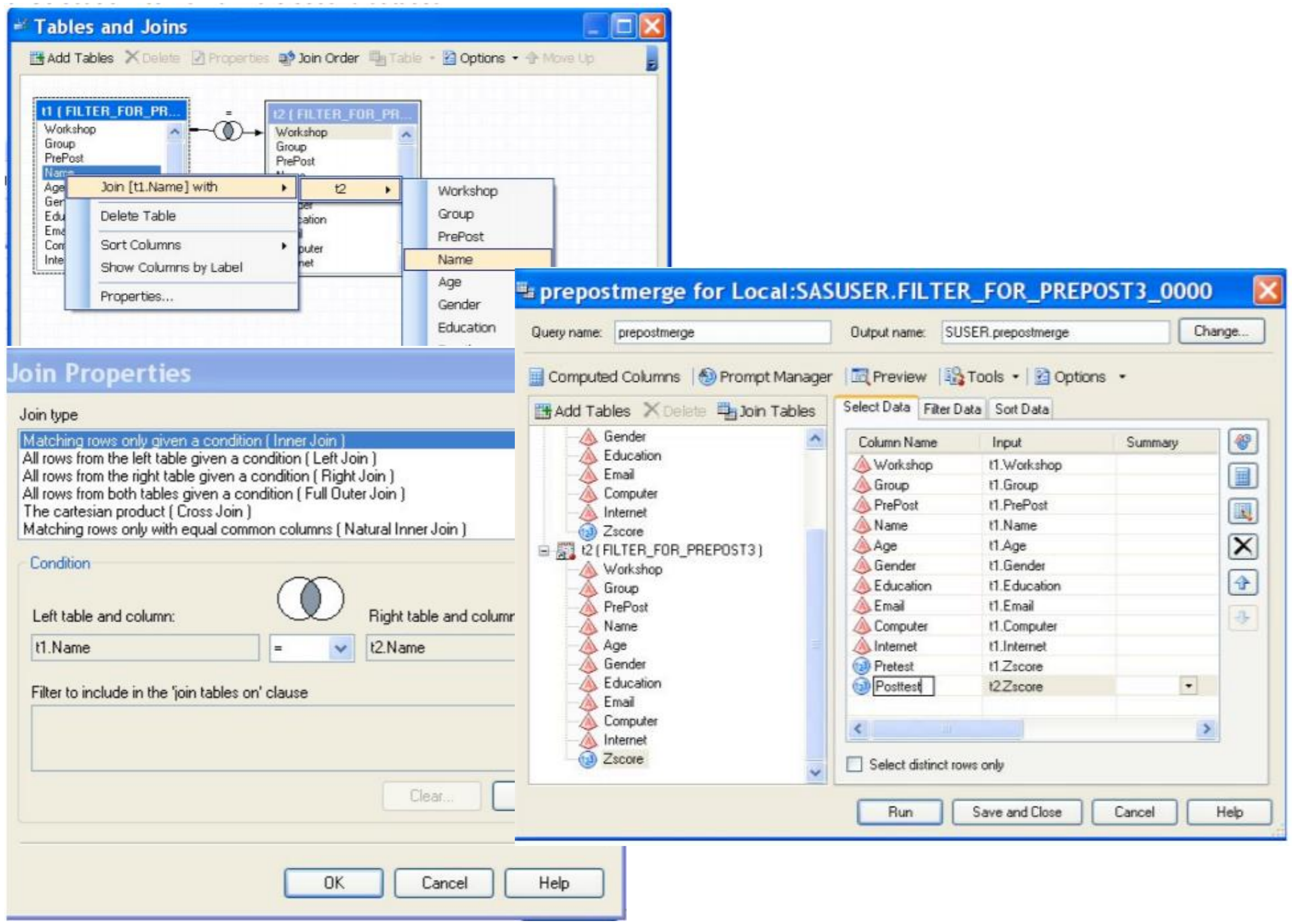
Now we merge LR3_SCR

Alternatively we can merge all dataset in one step



```
proc sort data=LR1;by Cust_ID;run;
proc sort data=LR2_HP;by Cust_ID;run;
proc sort data=LR3_SCR;by Cust_ID;run;
data LR_DF;
    merge LR1(in=aa) LR2_HP(in=bb) LR3_SCR(in=cc);
    by Cust_ID;
    if aa;
run;
```

Merging using SAS EG



Join type

- Matching rows only given a condition (Inner Join)
- All rows from the left table given a condition (Left Join)
- All rows from the right table given a condition (Right Join)
- All rows from both tables given a condition (Full Outer Join)
- The cartesian product (Cross Join)
- Matching rows only with equal common columns (Natural Inner Join)

Condition

Left table and column: t1.Name = Right table and column: t2.Name

Filter to include in the 'join tables on' clause

prepostmerge for Local:SASUSER.FILTER_FOR_PREPOST3_0000

Query name: prepostmerge   Output name: SUSER.prepostmerge

t2 (FILTER_FOR_PREPOST3): Workshop, Group, PrePost, Name, Age, Gender, Education, Email, Computer, Internet, Zscore

Column Name	Input	Summary
Workshop	t1.Workshop	
Group	t1.Group	
PrePost	t1.PrePost	
Name	t1.Name	
Age	t1.Age	
Gender	t1.Gender	
Education	t1.Education	
Email	t1.Email	
Computer	t1.Computer	
Internet	t1.Internet	
Pretest	t1.Zscore	
Posttest	t2.Zscore	

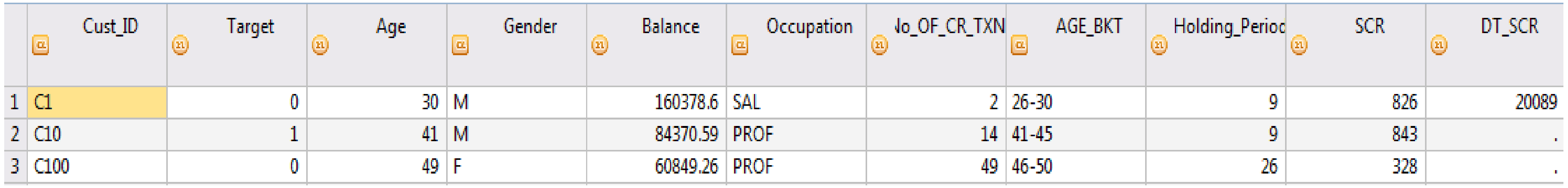
Select distinct rows only

Steps to Merge in SAS

1) Select Dataset 1
2) Click on Query Builder
3) Click on Join Table
4) Select Dataset 2 to be merged
5) Right Click on Column to be used for Merge
6) Select Join to column from other dataset
7) Select the TYPE of JOIN
8) Click OK & Close
9) From Query Builder select Columns in Final Dataset
10) Give Output Dataset Name
11) Click Run

Drop Statement

- Say we have to DROP the DT_SCR column as it is mostly blank

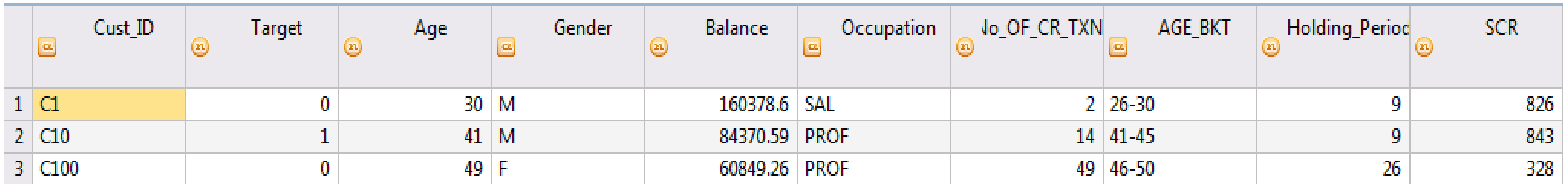


	Cust_ID	Target	Age	Gender	Balance	Occupation	No_OF_CR_TXN	AGE_BKT	Holding_Perioc	SCR	DT_SCR
1	C1	0	30	M	160378.6	SAL	2	26-30	9	826	20089
2	C10	1	41	M	84370.59	PROF	14	41-45	9	843	.
3	C100	0	49	F	60849.26	PROF	49	46-50	26	328	.

```
data LR_DF_DROP;
    set LR_DF(drop=DT_SCR);
run;
```

```
data LR_DF_DROP;
    set LR_DF;
    drop=DT_SCR;
run;
```

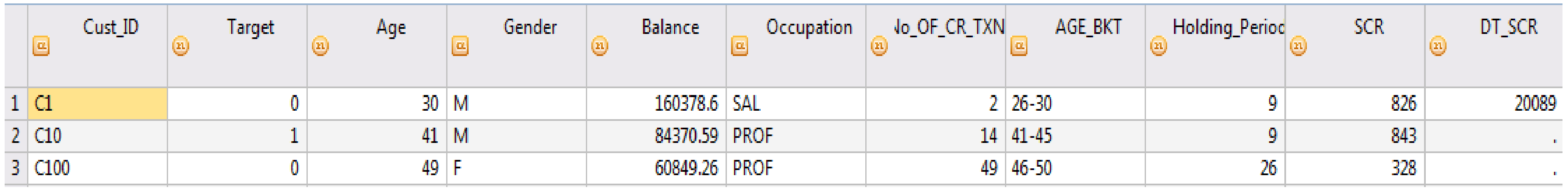
```
data LR_DF_DROP(drop=DT_SCR);
    set LR_DF;
run;
```



	Cust_ID	Target	Age	Gender	Balance	Occupation	No_OF_CR_TXN	AGE_BKT	Holding_Perioc	SCR
1	C1	0	30	M	160378.6	SAL	2	26-30	9	826
2	C10	1	41	M	84370.59	PROF	14	41-45	9	843
3	C100	0	49	F	60849.26	PROF	49	46-50	26	328

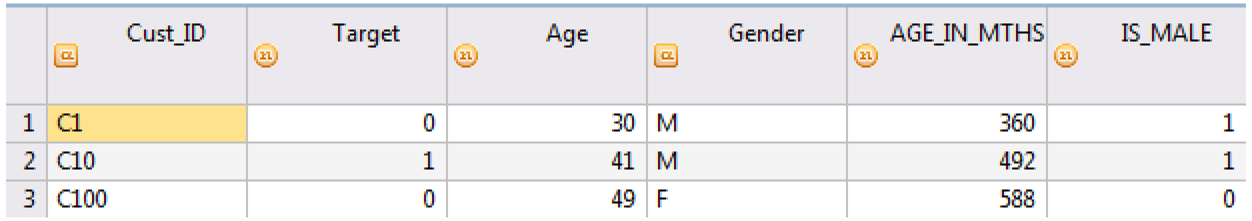
KEEP Statement

- Say we wish to KEEP only the CUST_ID, TARGET, AGE, GENDER columns



	Cust_ID	Target	Age	Gender	Balance	Occupation	lo_OF_CR_TXN	AGE_BKT	Holding_Perioc	SCR	DT_SCR
1	C1	0	30	M	160378.6	SAL	2	26-30	9	826	20089
2	C10	1	41	M	84370.59	PROF	14	41-45	9	843	.
3	C100	0	49	F	60849.26	PROF	49	46-50	26	328	.

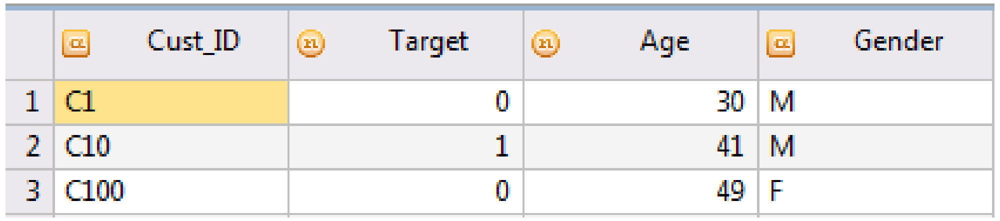
```
data LR_DF_KEEP;
    set LR_DF(keep=Cust_ID Target Age Gender);
    Age_In_mths=Age*12;
    if Gender="M"
    then Is_Male=1;
    else Is_Male=0;
run;
```



	Cust_ID	Target	Age	Gender	AGE_IN_MTHS	IS_MALE
1	C1	0	30	M	360	1
2	C10	1	41	M	492	1
3	C100	0	49	F	588	0

```
data LR_DF_KEEP;
    set LR_DF;
    keep=Cust_ID Target Age Gender;
    Age_In_mths=Age*12;
    if Gender="M"
    then Is_Male=1;
    else Is_Male=0;
run;
```

```
data LR_DF_KEEP(keep=Cust_ID Target Age Gender);
    set LR_DF;
    Age_In_mths=Age*12;
    if Gender="M"
    then Is_Male=1;
    else Is_Male=0;
run;
```



	Cust_ID	Target	Age	Gender
1	C1	0	30	M
2	C10	1	41	M
3	C100	0	49	F

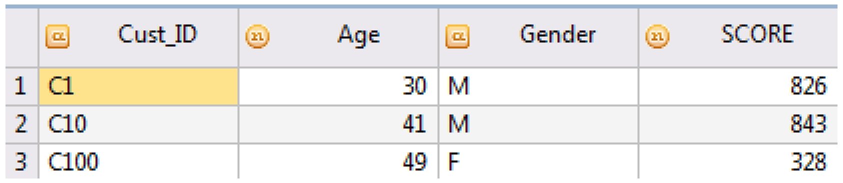
Relabeling Column Names

- Let’s rename SCR to SCORE

```
data LR_DF_RENAME (rename=(SCR=SCORE));
    set LR_DF(KEEP=Cust_ID Age SCR );
run;
```

```
data LR_DF_RENAME;
    set LR_DF(KEEP=Cust_ID Age SCR rename=(SCR=SCORE));
run;
```

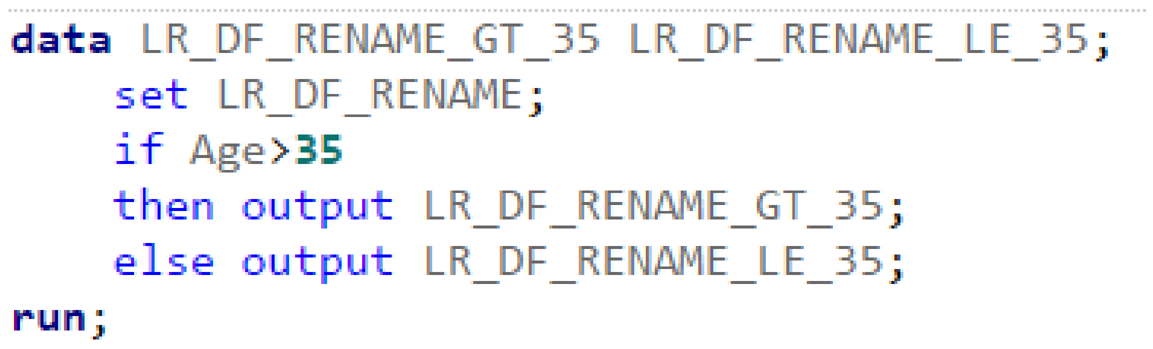
```
data LR_DF_RENAME ;
    set LR_DF(KEEP=Cust_ID Age SCR );
    rename SCR=SCORE;
run;
```



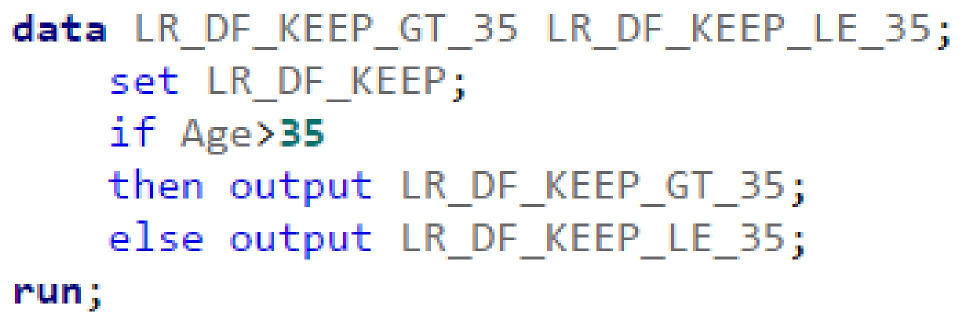
	Cust_ID	Age	Gender	SCORE
1	C1	30	M	826
2	C10	41	M	843
3	C100	49	F	328

Creating Multiple Datasets in Data Step

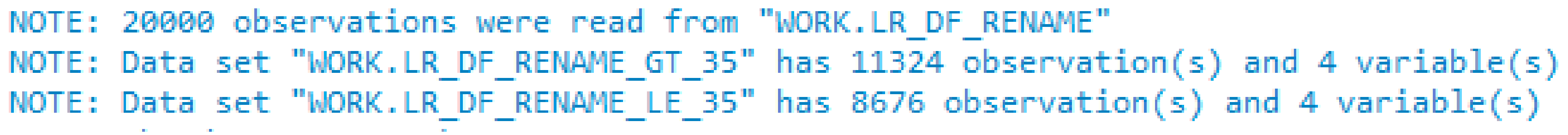
- Suppose we wish to split LR_DF_RENAME and LR_DF_KEEP datasets by AGE value above and below 35 years



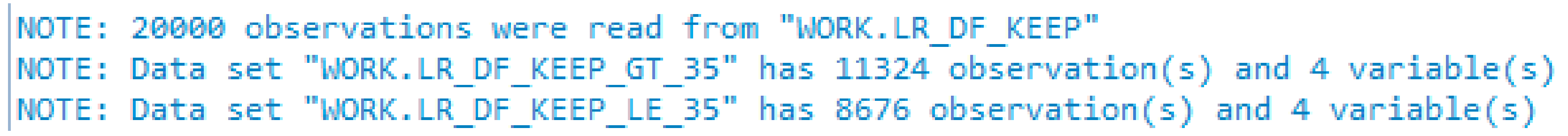
```
data LR_DF_RENAME_GT_35 LR_DF_RENAME_LE_35;
     set LR_DF_RENAME;
     if Age>35
     then output LR_DF_RENAME_GT_35;
     else output LR_DF_RENAME_LE_35;
run;
```



```
data LR_DF_KEEP_GT_35 LR_DF_KEEP_LE_35;
     set LR_DF_KEEP;
     if Age>35
     then output LR_DF_KEEP_GT_35;
     else output LR_DF_KEEP_LE_35;
run;
```

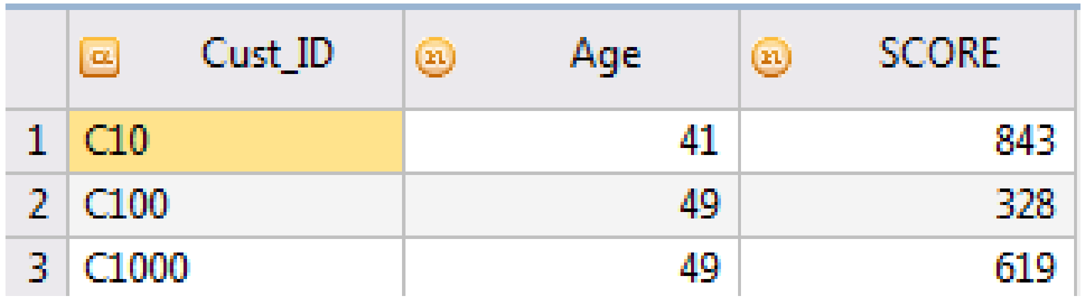


```
NOTE: 20000 observations were read from "WORK.LR_DF_RENAME"
NOTE: Data set "WORK.LR_DF_RENAME_GT_35" has 11324 observation(s) and 4 variable(s)
NOTE: Data set "WORK.LR_DF_RENAME_LE_35" has 8676 observation(s) and 4 variable(s)
```



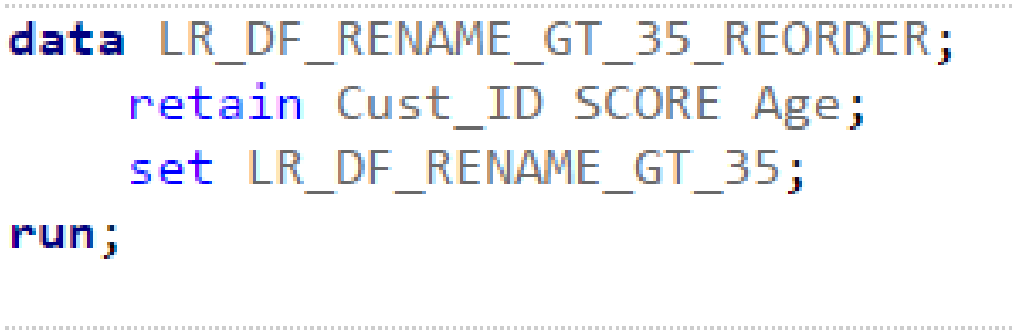
```
NOTE: 20000 observations were read from "WORK.LR_DF_KEEP"
NOTE: Data set "WORK.LR_DF_KEEP_GT_35" has 11324 observation(s) and 4 variable(s)
NOTE: Data set "WORK.LR_DF_KEEP_LE_35" has 8676 observation(s) and 4 variable(s)
```

Reordering the Columns



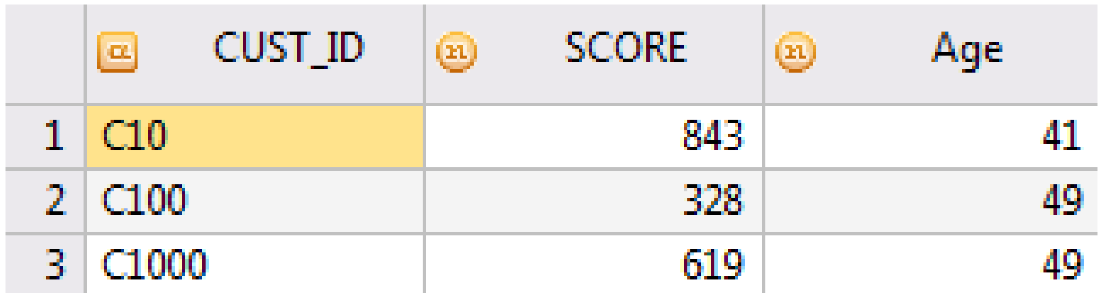
	Cust_ID	Age	SCORE
1	C10	41	843
2	C100	49	328
3	C1000	49	619

Before Reordering



```
data LR_DF_RENAME_GT_35_REORDER;
    retain Cust_ID SCORE Age;
    set LR_DF_RENAME_GT_35;
run;
```

Reordering Syntax - RETAIN

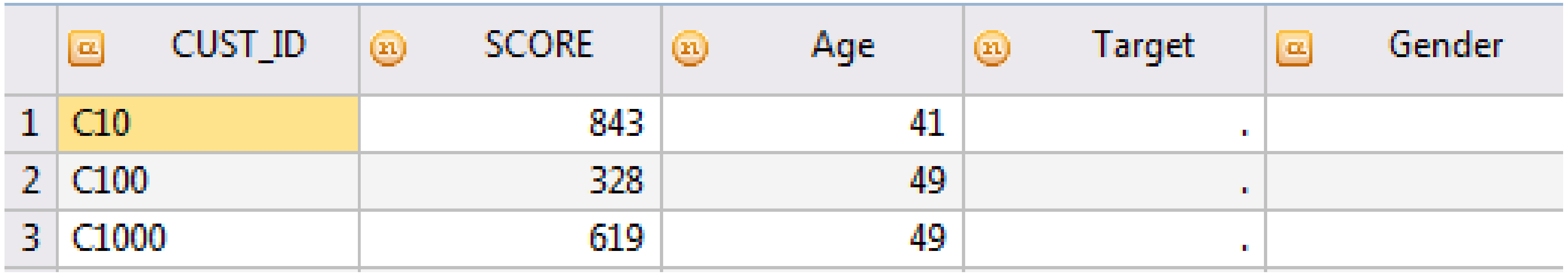


	CUST_ID	SCORE	Age
1	C10	843	41
2	C100	328	49
3	C1000	619	49

After Reordering

Combining SAS Datasets | Append

```
data LR_DF_APPEND;
    set LR_DF_RENAME_GT_35_REORDER LR_DF_KEEP_LE_35;
run;
```

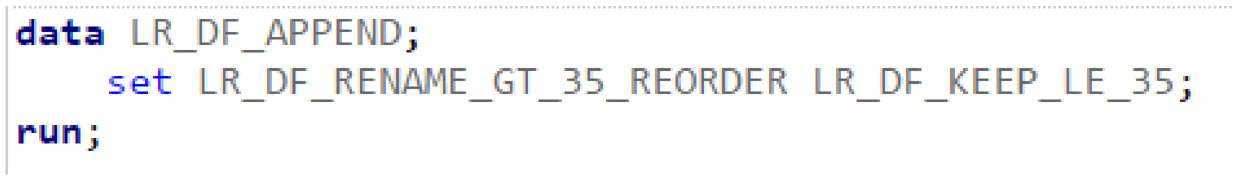


	CUST_ID	SCORE	Age	Target	Gender
1	C10	843	41	.	
2	C100	328	49	.	
3	C1000	619	49	.	

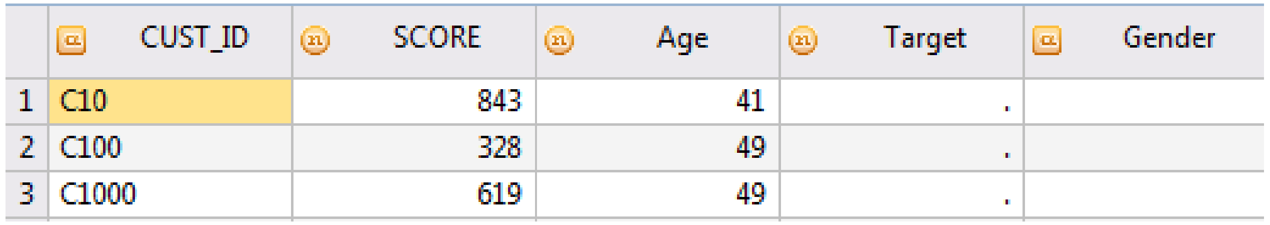
```
Proc sort data=LR_DF_APPEND;
    by descending Target;
run;
```

Combining SAS Datasets

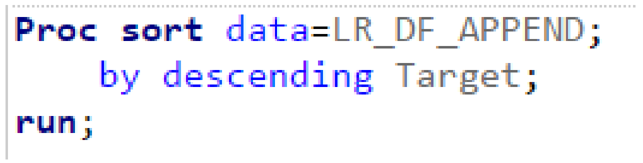
- Let us combine LR_DF_RENAME_GT_35_REORDER with LR_DF_KEEP_LE_35



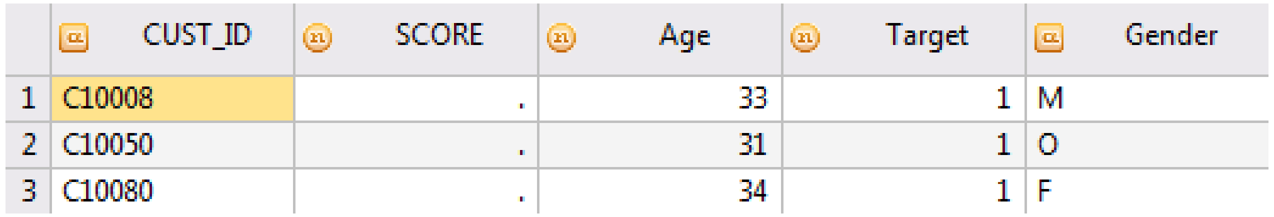
```
data LR_DF_APPEND;
    set LR_DF_RENAME_GT_35_REORDER LR_DF_KEEP_LE_35;
run;
```



	CUST_ID	SCORE	Age	Target	Gender
1	C10	843	41	.	
2	C100	328	49	.	
3	C1000	619	49	.	

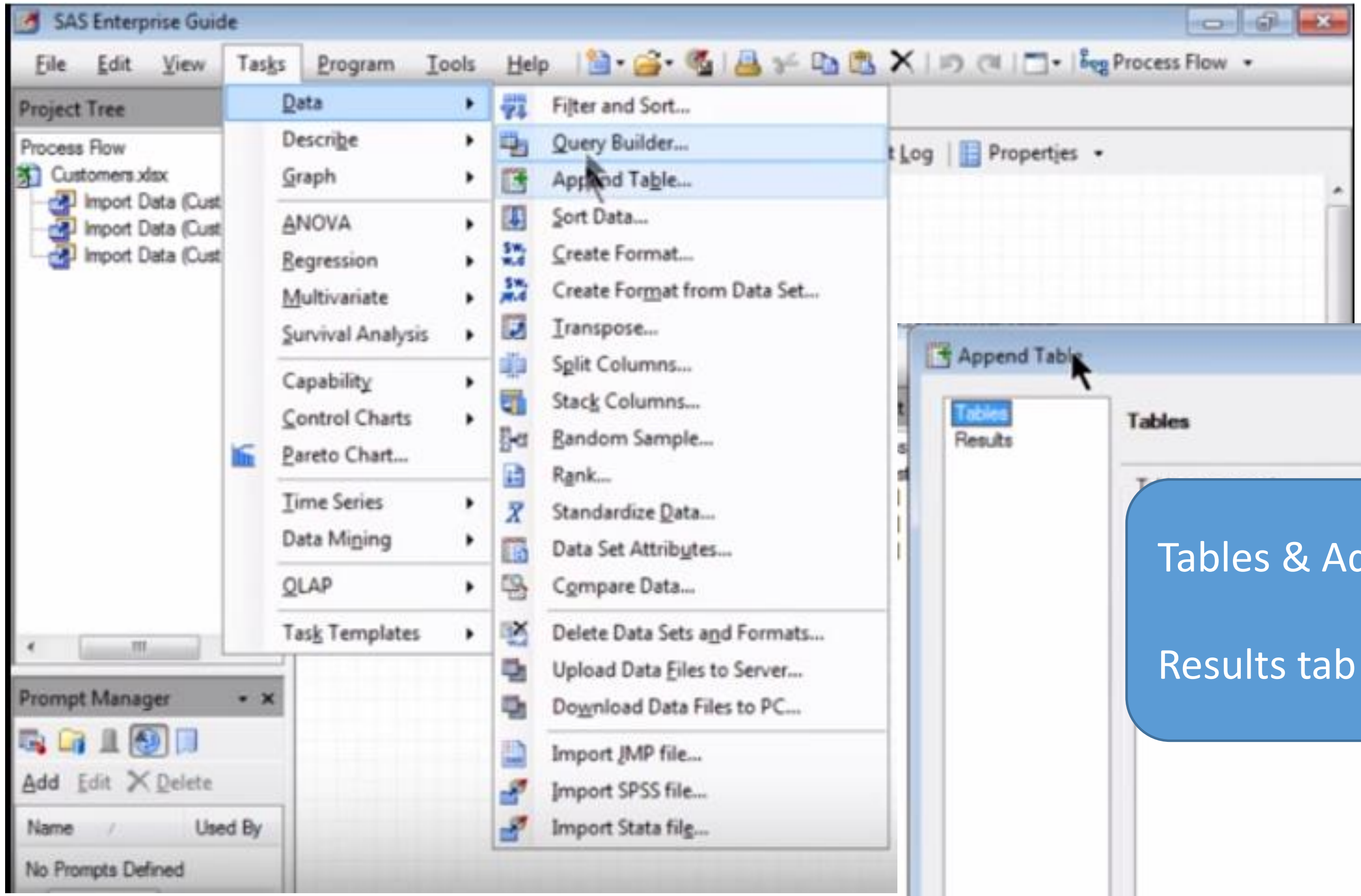


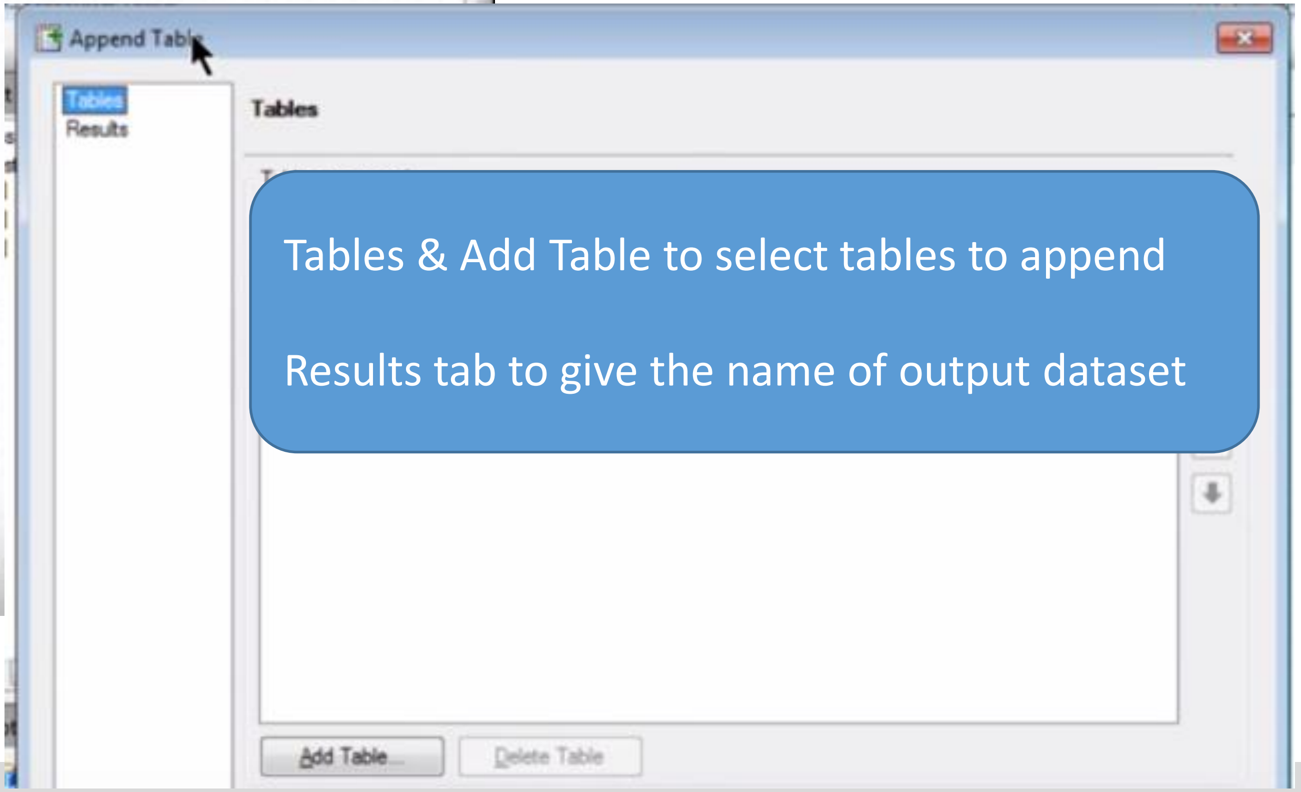
```
Proc sort data=LR_DF_APPEND;
    by descending Target;
run;
```



	CUST_ID	SCORE	Age	Target	Gender
1	C10008	.	33	1	M
2	C10050	.	31	1	O
3	C10080	.	34	1	F

SAS EG Append





Select a Data Set to Append

Tasks > Append Table

Tables & Add Table to select tables to append

Results tab to give the name of output dataset

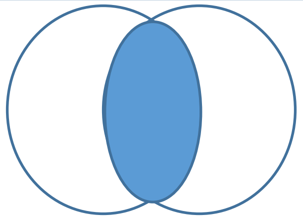
QUIZ

Earning is in Learning
- Rajesh Jakhotia

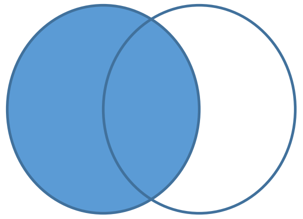
Quiz

Match the following types of Joins

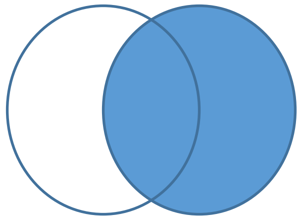
A)



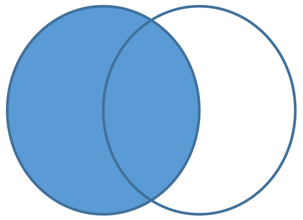
B)



C)



D)



1) Full Outer Join

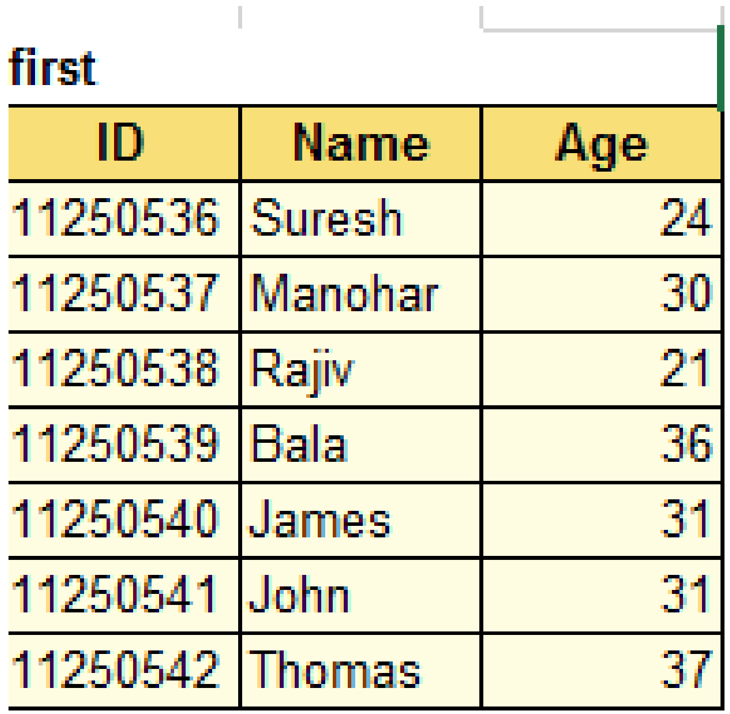
2) Right Outer Join

3) Inner Join

4) Left Outer Join

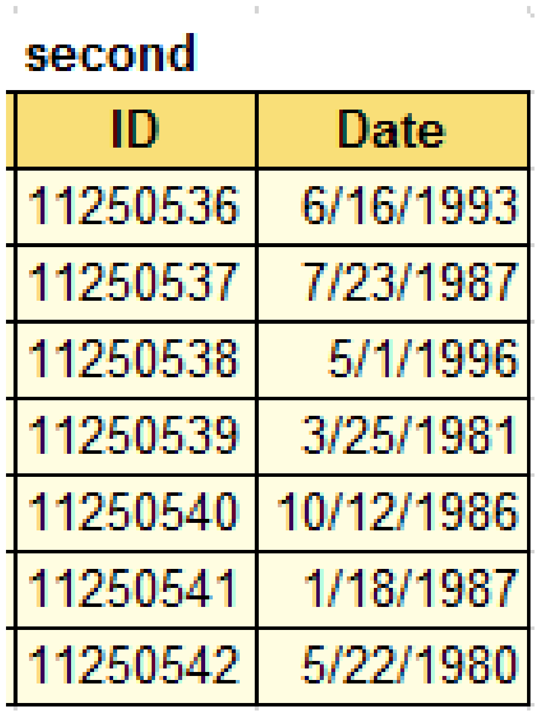
Quiz

What is the relationship of the data set ”first” to the data set ”second” when merged by the variable ID?



first

ID	Name	Age
11250536	Suresh	24
11250537	Manohar	30
11250538	Rajiv	21
11250539	Bala	36
11250540	James	31
11250541	John	31
11250542	Thomas	37



second

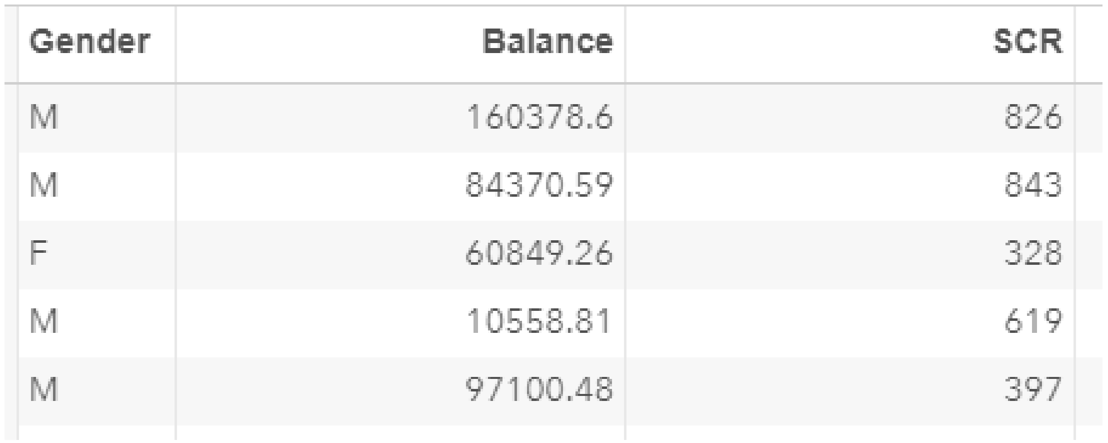
ID	Date
11250536	6/16/1993
11250537	7/23/1987
11250538	5/1/1996
11250539	3/25/1981
11250540	10/12/1986
11250541	1/18/1987
11250542	5/22/1980

<u>Options:</u>

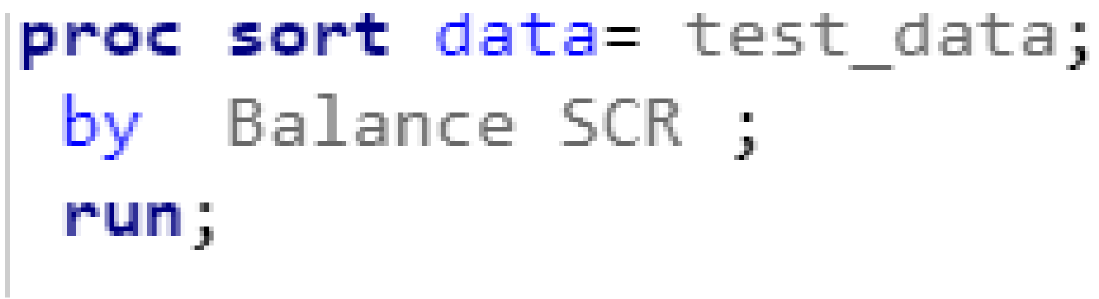
1. **one-to-one**
2. **one-to-many**
3. **many-to-one**
4. **many-to-many**

Quiz

What is the expected output if you run PROC SORT query on given Dataset

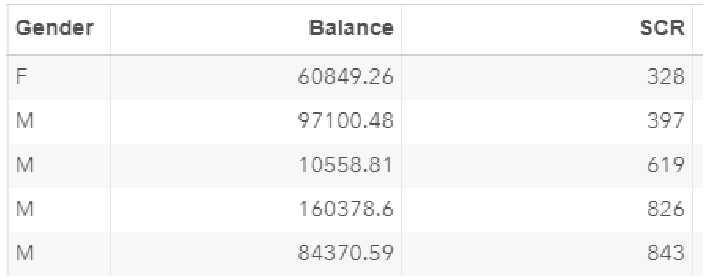


Gender	Balance	SCR
M	160378.6	826
M	84370.59	843
F	60849.26	328
M	10558.81	619
M	97100.48	397



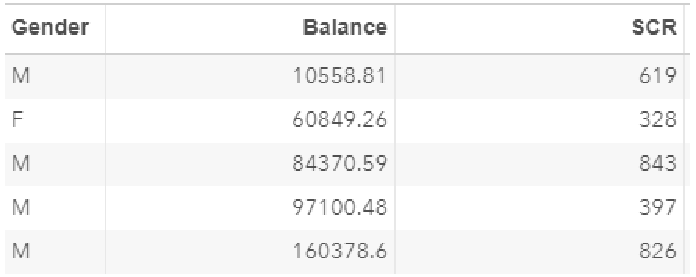
```
proc sort data= test_data;
 by  Balance SCR ;
 run;
```

Output 1



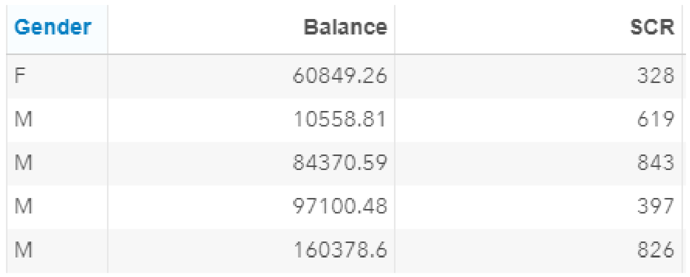
Gender	Balance	SCR
F	60849.26	328
M	97100.48	397
M	10558.81	619
M	160378.6	826
M	84370.59	843

Output 2



Gender	Balance	SCR
M	10558.81	619
F	60849.26	328
M	84370.59	843
M	97100.48	397
M	160378.6	826

Output 3



Gender	Balance	SCR
F	60849.26	328
M	10558.81	619
M	84370.59	843
M	97100.48	397
M	160378.6	826

Quiz

If you want to create multiple file in one step which procedure you will choose?

Options

A) merge

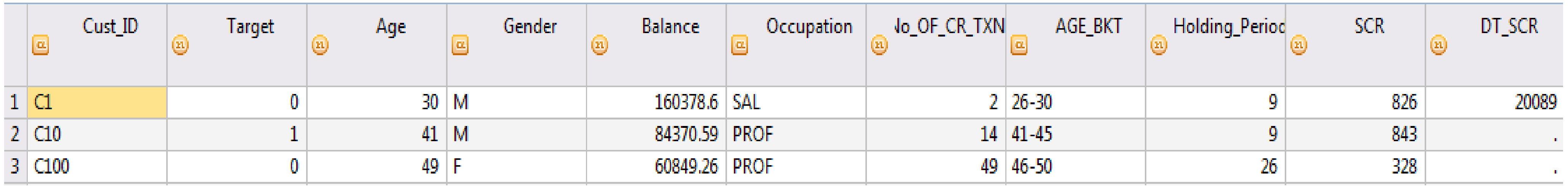
B) keep-drop

C) retains

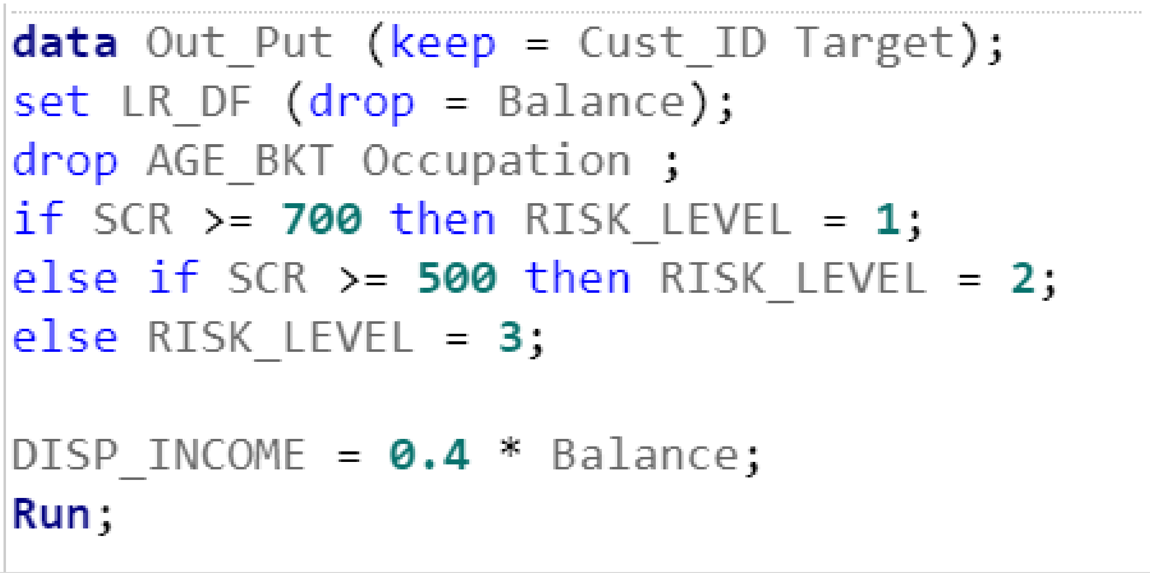
D) if-then-else

Quiz

How many variables will the Out_Put Dataset contain when below step is executed on "LR_DF" dataset as shown below?



	Cust_ID	Target	Age	Gender	Balance	Occupation	lo_OF_CR_TXN	AGE_BKT	Holding_Perioc	SCR	DT_SCR
1	C1	0	30	M	160378.6	SAL	2	26-30	9	826	20089
2	C10	1	41	M	84370.59	PROF	14	41-45	9	843	.
3	C100	0	49	F	60849.26	PROF	49	46-50	26	328	.



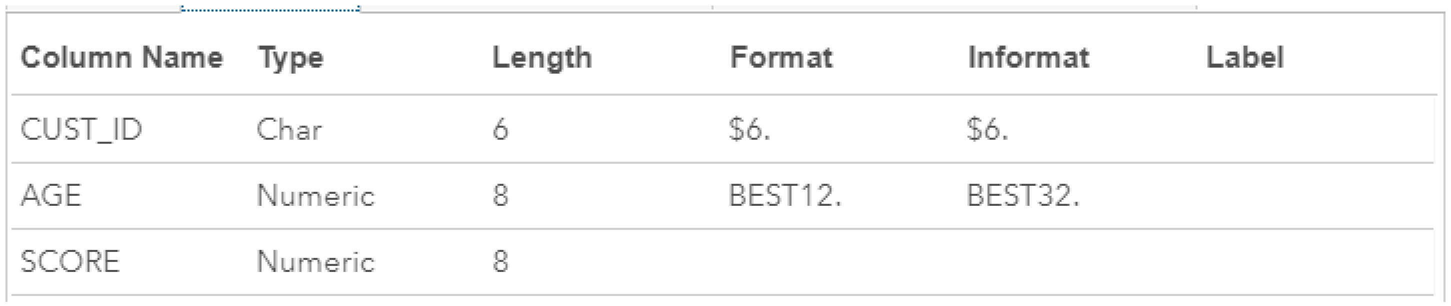
```
data Out_Put (keep = Cust_ID Target);
set LR_DF (drop = Balance);
drop AGE_BKT Occupation ;
if SCR >= 700 then RISK_LEVEL = 1;
else if SCR >= 500 then RISK_LEVEL = 2;
else RISK_LEVEL = 3;

DISP_INCOME = 0.4 * Balance;
Run;
```

Options:

1. **0 – (ERROR in Code)**
2. **2**
3. **3**
4. **4**

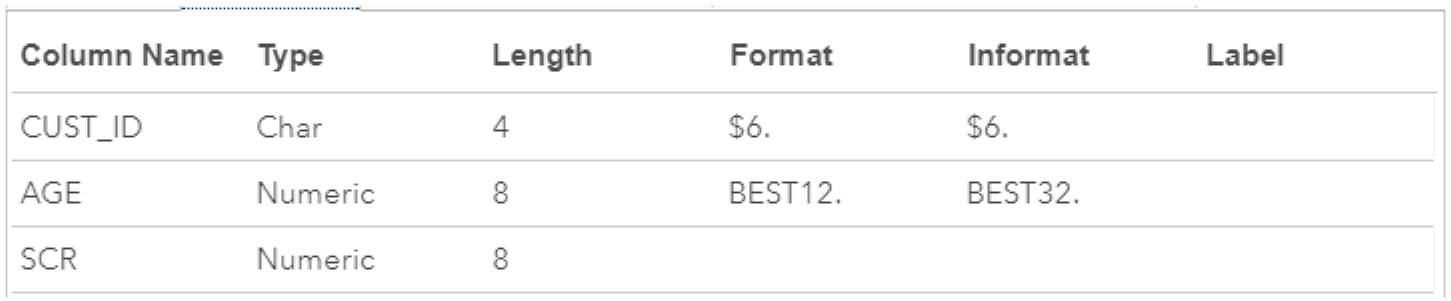
Quiz

What will happen when you append DST1 and DST2 to form DST3



Column Name	Type	Length	Format	Informat	Label
CUST_ID	Char	6	$6.	$6.	
AGE	Numeric	8	BEST12.	BEST32.	
SCORE	Numeric	8			

DST1



Column Name	Type	Length	Format	Informat	Label
CUST_ID	Char	4	$6.	$6.	
AGE	Numeric	8	BEST12.	BEST32.	
SCR	Numeric	8			

DST2

DST3 will have 4 columns? Y / N

CUST_ID values of DST1 will get truncated? Y/N

CUST_ID length in DST3 will be 4? Y/N

K2 Analytics
Building Skills, Building Individuals

Thank you

Contact us:

ar.jakhotia@k2analytics.co.in