SAS Lesson 07

Business Scenario

Include only the employees from Australia who have a bonus month in December.

```
data work.december;
    set orion.sales;
    where Country='AU';
    BonusMonth=month(Hire_Date);
    if BonusMonth=12;
    Bonus=500;

Compensation=sum(Salary,Bonus);
run;
```

Partial SAS Log

```
NOTE: There were 63 observations read from the data set ORION.SALES.
      WHERE Country='AU';
NOTE: The data set WORK.DECEMBER has 3 observations and 12 variables.
```

Quiz

Could you write only an IF statement?

- Yes
- No

```
data work.december;
   set orion.sales;
   where Country='AU';
   BonusMonth=month(Hire_Date);
   if BonusMonth=12;
   Bonus=500;
   Compensatio
run;
```

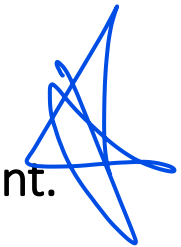
```
data work.december;
   set orion.sales;
   BonusMonth=month(Hire_Date);
   if BonusMonth=12 and Country='AU';
   Bonus=500;
   Compensation=sum(Salary,Bonus);
run;
```

Quiz – Correct Answer

Could you write only an IF statement?

- ◉ Yes
- ○ No

Yes, but the program using both the WHERE and IF statements is more efficient.



Both methods create a data set with three observations. The program using both statements reads 63 observations into the PDV. The program using only the IF statement reads 165 observations into the PDV.

WHERE Statement versus Subsetting IF Statement

Step and Usage	WHERE	IF
PROC step (Except IMPORT)	Yes	No
DATA step (source of variable)		
INPUT statement	No	Yes
assignment statement	No	Yes
SET statement (single data set)	Yes	Yes
SET/MERGE statement (multiple data sets)		
Variable in ALL data sets	Yes	Yes
Variable not in ALL data sets	No	Yes

The IF-THEN DELETE Statement

An alternative to the subsetting IF statement is the DELETE statement in an IF-THEN statement.

General form of the IF-THEN DELETE statement:

IF *expression* **THEN DELETE;**

The *DELETE statement* stops processing the current observation.

The IF-THEN DELETE Statement

```
data work.december;
   set orion.sales;
   where Country='AU';
   BonusMonth=month(Hire_Date);
   if BonusMonth ne 12 then delete;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
run;
```

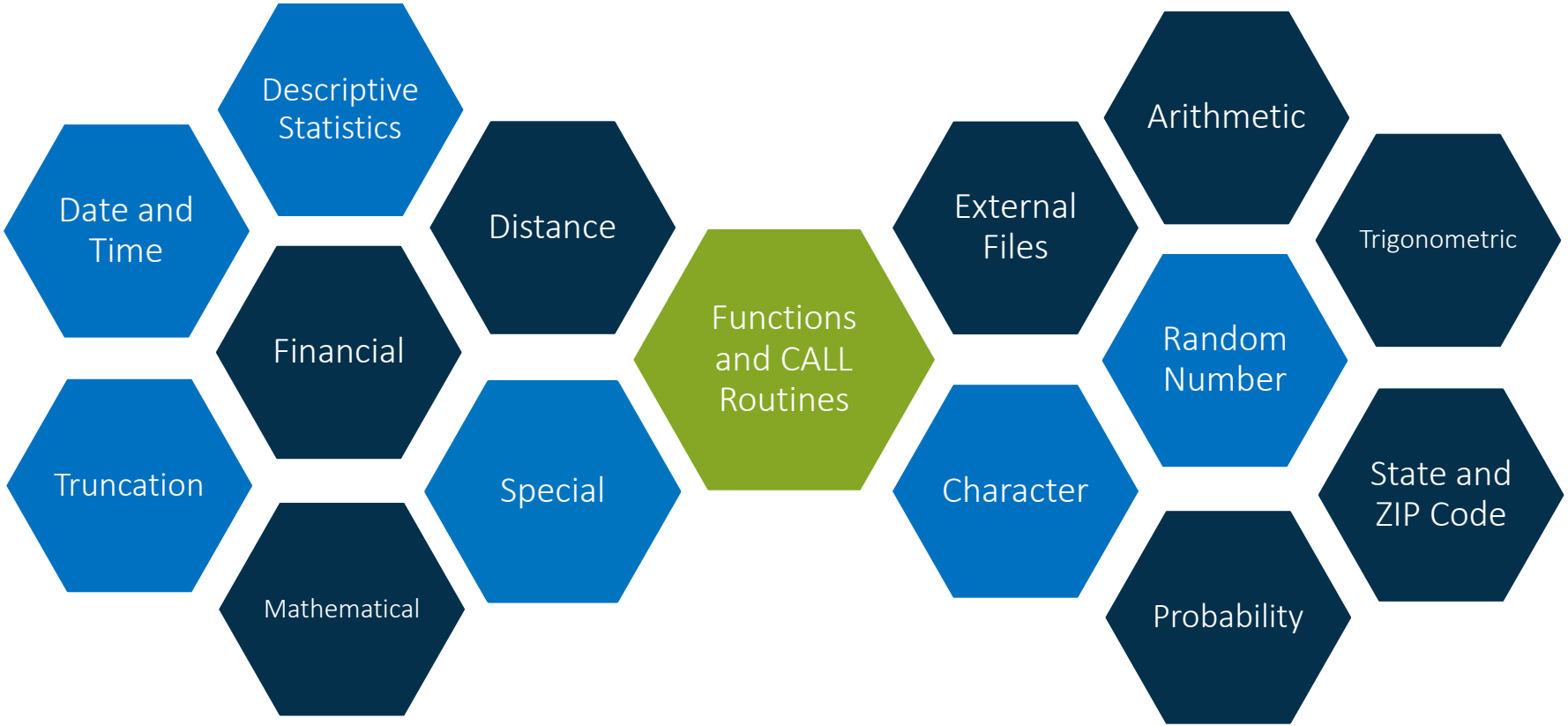
equivalent

```
data work.december;
   set orion.sales;
   where Country='AU';
   BonusMonth=month(Hire_Date);
   if BonusMonth=12;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
run;
```

Manipulating Data with Functions

Understanding SAS Functions and CALL Routines – Ch. 14

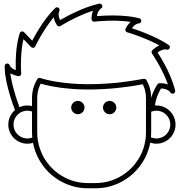
SAS Functions and CALL Routines



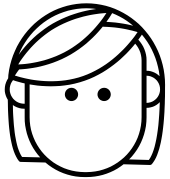
What Is a SAS Function?

function(*argument1*, *argument2*, ...);

An *argument* can be a constant, a variable, or any SAS expression, including another function.



A function performs a specific computation or manipulation and **returns a value**.

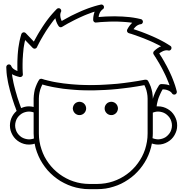


Target Variables

```
Full_Name=cat (First_Name, ' ' , Last_Name);
```

A target variable is the variable to which a function result is assigned.

Default lengths depend on the function and can be as long as 200 characters.



Add a LENGTH statement to avoid using unnecessary space.

Using SAS Functions

You can use functions in DATA step statements anywhere that an expression can appear.

```
data contrib;
   set orion.employee_donations;
   Total=sum(Qtr1,Qtr2,Qtr3,Qtr4);
   if Total ge 50;
run;
```

Contributions $50 and Over

Employee_ID	Qtr1	Qtr2	Qtr3	Qtr4	Total
120267	15	15	15	15	60
120269	20	20	20	20	80
120271	20	20	20	20	80
120275	15	15	15	15	60
120660	25	25	25	25	100

SAS Variable Lists

An alternative to typing variables separately is to use a *SAS variable list*.

```
data contrib;
   set orion.employee_donations;
   Total=sum(of Qtr1-Qtr4);
   if Total ge 50;
run;
```

When you use a SAS variable list in a SAS function, use the keyword OF in front of the first variable name in the list.

Quiz

What happens if you forget the keyword OF?

```
data contrib;
   set orion.employee_donations;
   Total=sum(Qtr1-Qtr4);
   if Total ge 50;
run;
```

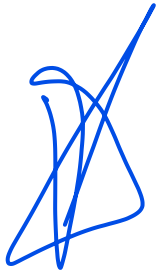
Quiz – Correct Answer

What happens if you forget the keyword OF?

```
data contrib;
   set orion.employee_donations;
   Total=sum(Qtr1-Qtr4);
   if Total ge 50;
run;
```

QTR1-QTR4 will be interpreted as QTR1 minus QTR4.

SAS Variable Lists



A SAS *variable list* is an abbreviated method of referring to a group of variable names. SAS enables you to use the following variable lists:

- Numbered range (X1-Xn)
- Name range (X--A, X-numeric-A, X-character-A)
- Name prefix (X:)
- Special SAS name lists, by variable type (_ALL_, _NUMERIC_, _CHARACTER_)

SAS Variable Lists – Examples

Numbered Range List

PDV

Qtr1	Qtr2	Var1	Qtr3	Qtr4

```
Total = sum(of Qtr1-Qtr4)
```

Var1 not included

Name Range List (by position in PDV)

PDV

Qtr1	Second	Q3	Fourth	Var2

```
Total = sum(of Qtr1--Fourth)
```

Var2 not included

SAS Variable Lists – Examples

Name Prefix List

PDV

TotJan	Qtr2	TotFeb	TotMar

```
Total = sum(of Tot:)
```

Qtr2 not included

Special Name Lists

PDV

Qtr1 N	Name $	Q3 N	Fourth N

```
Total = sum(of _Numeric_)
```

Name not included

Specifying Variable Lists

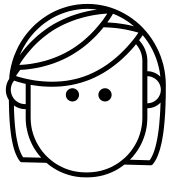
```
data quiz_summary;
     set pg2.class_quiz;
     Name=upcase(Name);
     AvgQuiz=mean(of Q:);
     format Quiz1--AvgQuiz 3.1;
     /*OR*/
     format _numeric_ 3.1;
run;
```

The name prefix includes all columns whose name begins with Q.

The keyword _NUMERIC_ includes all numeric columns.

The double dash includes all columns between and including the two specified columns as they are ordered in the PDV.

Variable lists can be used in statements as well!



Quiz

Complete the assignment statement for **Total** by using a SAS variable list and the SUM function to add the values for **Year1**, **Year2**, **Year3**, and **Year4**.

PDV

Year2	Year1	Year3	Year4	Sales

```
Total = sum(of   ?   )
```

Correct Answer

Any of these assignment statements would give the correct value for **Total**.

PDV

Year2	Year1	Year3	Year4	Sales

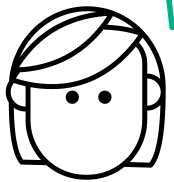
```
Total = sum(of Year1-Year4);

Total = sum(of Year2--Year4);

Total = sum(of Year:);
```

What Is a SAS CALL Routine?

CALL *routine*(*argument-1* <, ...*argument-n*>);

A CALL routine is used in a CALL statement.

CALL routines **alter** column values or perform system **actions**. They **cannot** be used in assignment statements or expressions.



Using a CALL Routine to Modify Data

```
data quiz_report;
     set pg2.class_quiz;
     if Name in("Barbara", "James")
          then call missing(of Q:);
run;
```

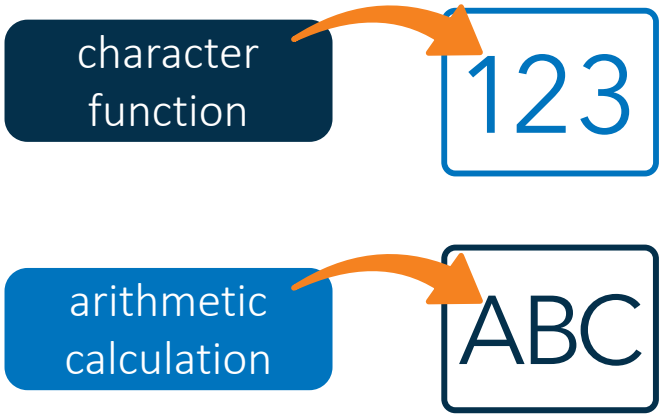
CALL MISSING assigns missing values to each variable in the argument list (character or numeric).

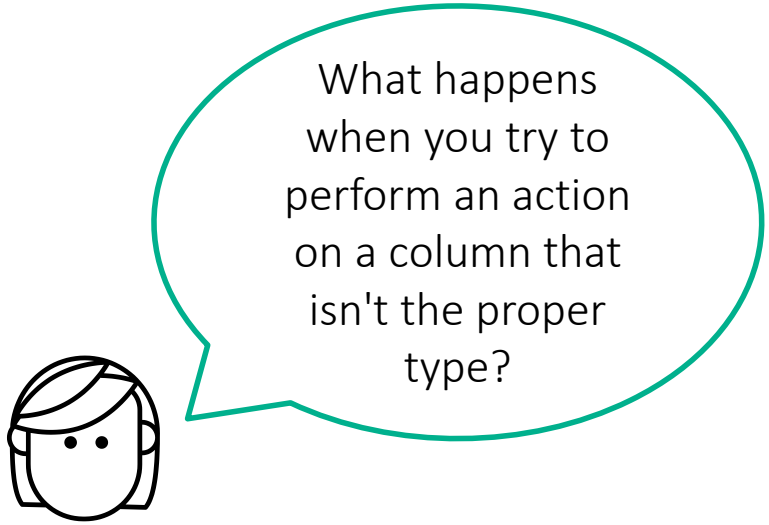
	Name	Quiz1	Quiz2	Quiz3	Quiz4	Quiz5
1	Alfred	8	7	6	9	8
2	Alice	7	6	4	9	8
3	Barbara	.	.	.	.	.
4	Carol	6	5	5	8	8
5	Henry	8	.	6	10	7
6	James	.	.	.	.	.
7	Jane	8	7	6	9	6

Manipulating Data with Functions

Using Special Functions to Convert Column Type

Handling Column Type





Automatic Conversion

This demonstration illustrates the automatic conversion of values from character to numeric.

Automatic Conversion of Column Type

```
Range = High-Low;
```

High	Low	Range
89.92	81.56	8.36
89.94	80.64	9.3
84.6	78.7	5.9
82.11	76.93	5.18
84.2	79.87	4.33

Automatic conversion is successful because **High** contains standard numeric values.

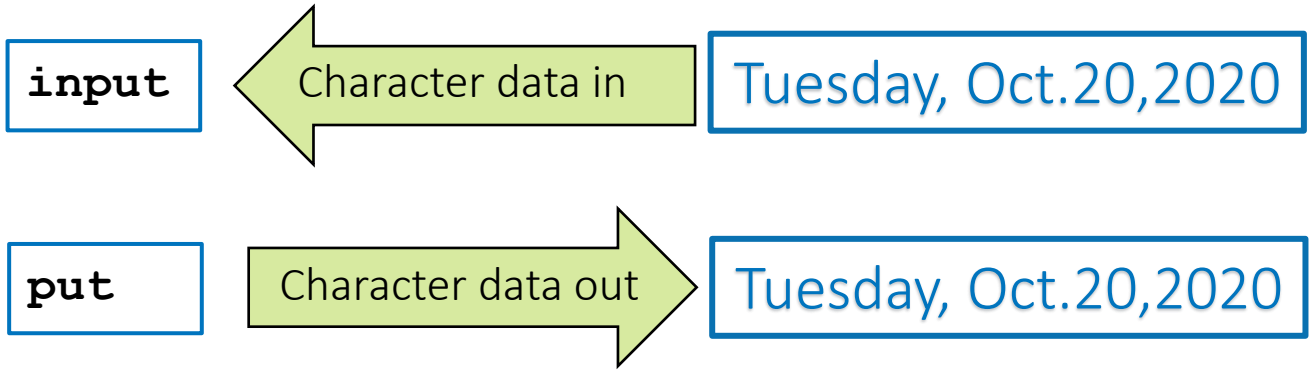
```
DailyVol = Volume/30;
```

Volume	DailyVol
5,025,627	.
4,455,657	.
5,960,945	.
5,043,800	.
5,754,876	.

Automatic conversion fails because **Volume** contains nonstandard numeric values.

Conversion Functions

Function	What it does
INPUT(*source*, *informat*)	Converts character values to numeric or other character values using a specified informat
PUT(*source*, *format*)	Converts numeric or character values to character values using a specified format



Converting Character Values to Numeric Values

character to numeric

Stock	Date	Open	Close
ABC Company	01DEC2017	89.15	82.2
ABC Company	01NOV2017	81.85	88.9
ABC Company	02OCT2017	80.22	81.88
ABC Company	01SEP2017	80.16	80.22

Stock	Date2	Open	Close
ABC Company	21154	89.15	82.2
ABC Company	21124	81.85	88.9
ABC Company	21094	80.22	81.88
ABC Company	21063	80.16	80.22

```
Date2=input(Date,date9.);
```

source

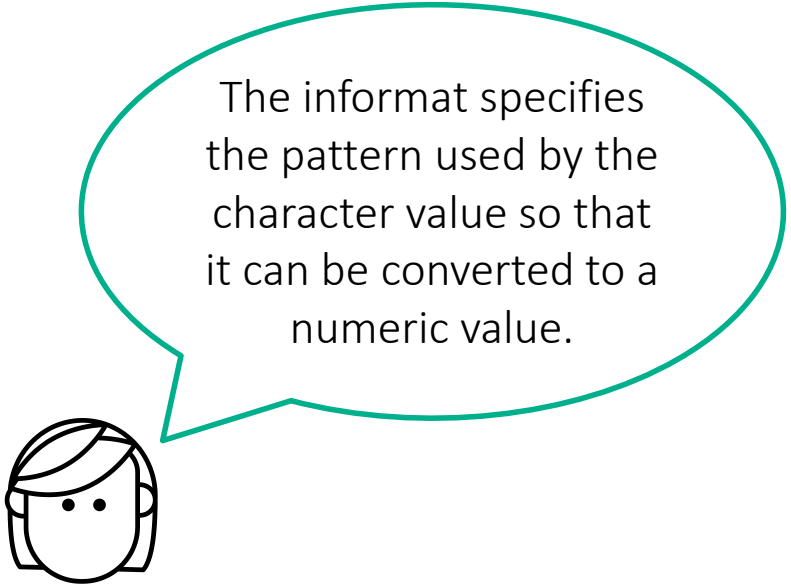
informat

What is an informat?

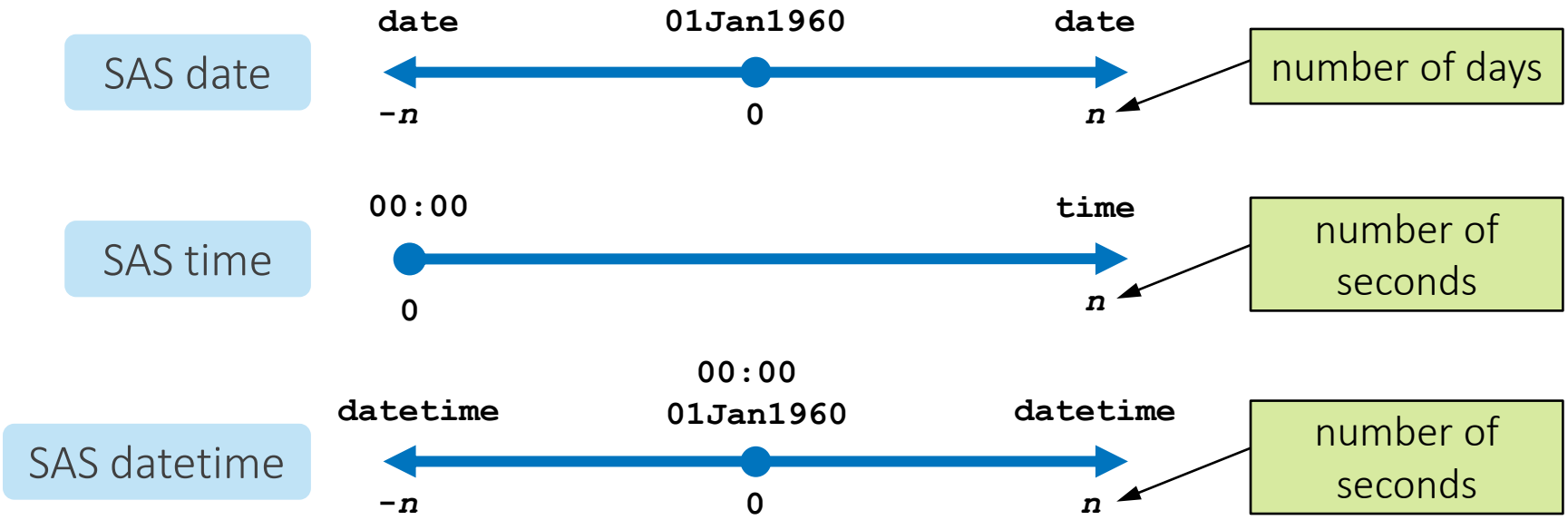
- Predefined **in**struction
- **In**forms SAS how to **in**terpret data
- Usually has a name
- Begins with $ to indicate character informat
- Has a field width
- Must have a period delimiter
- Determines data type of variables created with input function

Informats for Converting Character to Numeric

Character	Informat	Numeric
15OCT2018	DATE9.	21472
10/15/2018	MMDDYY10.	21472
15/10/2018	DDMMYY10.	21472
123,456.78 $123,456.78	COMMA12. DOLLAR12.	123456.78
123456	6.	123456



Date, Datetime, and Time Values



Additional Date and Time Informats

Character	Informat	Numeric
101518	MMDDYY6.	21472
6:45 PM	TIME8.	67500
18:45	TIME8.	67500
20Oct2020 6:45 PM	DATETIME18.	1918838700
20Oct2020 18:45:00	DATETIME18.	1918838700
10/20/2020 18:45:00	ANYDTDTE19.	22208
10/20/2020 18:45:00	ANYDTDTM19.	1918838700

Informats for Converting Character to Numeric

Character
15OCT2018
10/15/2018
10152018
20181015
Oct 15, 2018
October 15, 2018

ANYDTDTEw.

ANYDTDTMw.

SAS Date
21472
21472
21472
21472
21472
21472

The ANYDTDTE and ANYDTDTM informats can read dates written in many ways.

Informats for Converting Character to Numeric

```
data work.stocks2;
     set pg2.stocks2;
     NewVolume1=input(Volume,comma12.);
     NewVolume2=input(Volume,comma12.2);
     keep volume newvolume:;
run;
```

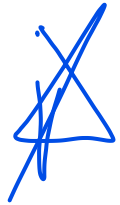
Volume	NewVolume1	NewVolume2
5,976,252	5976252	59762.52
5,556,471	5556471	55564.71
7,019,666	7019666	70196.66
5,772,280	5772280	57722.8

Be careful not to specify a decimal value with the informat unless you want to insert a new decimal point.

Explicit Conversion

This demonstration illustrates the explicit conversion of date/time values from character to numeric.

Facts about the PUT Function



- Always returns a character string
- Returns the source written with a format
- The format must the same type as the source
- Numeric formats right align results
- Character formats left align results
- Length of new variables is equal to the format width

What is a format?

- Controls the way data values are displayed or printed
- Usually has a name
- Begins with $ to indicate character format
- Width must be sufficient for data and all characters and decimal places
- Must have a period delimiter
- Permanently applied in DATA step
- Does NOT change the actual value unless assigned with a PUT function
- Temporarily applied in PROC steps

Formatting Data Values in Results

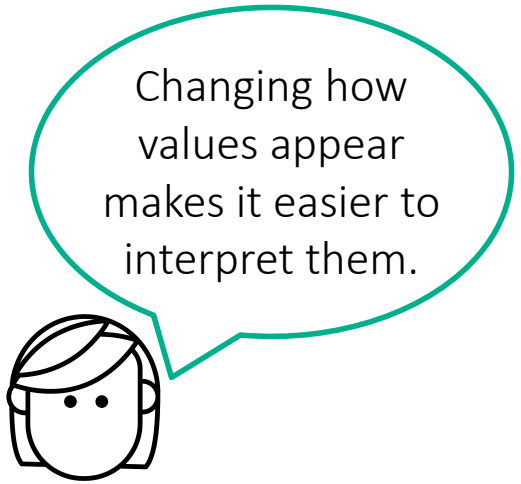
data

ID	HireDate	Salary
120201	21020	62000
145675	20742	76000
184430	21000	35000

$w.
w.d

report

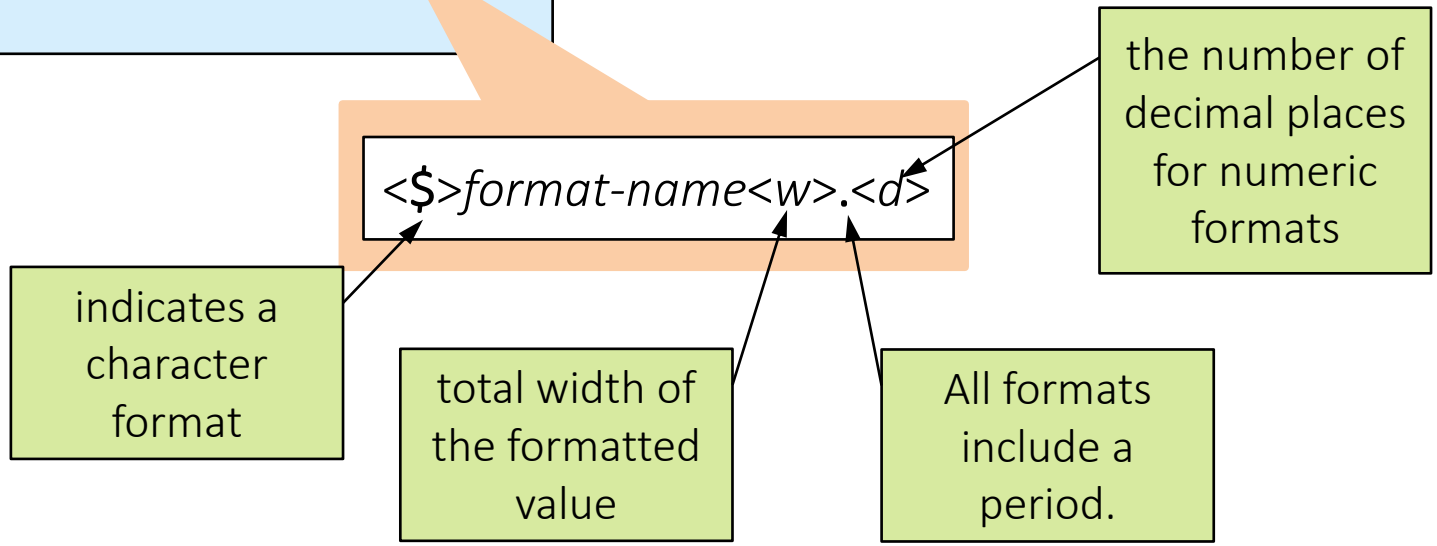
ID	HireDate	Salary
120201	01FEB2015	$62,000
145675	15OCT2016	$76,000
184430	30JUN2017	$35,000



Formatting Values in Data and Results

```
DATA output-table;
      SET input-table;
      FORMAT col-name(s)  format;
RUN;
```

```
PROC PRINT DATA=input-table;
      FORMAT col-name(s)  format;
RUN;
```



Common Formats for Numeric Values

Format Name	Example Value	Format Applied	Formatted Value
w.d	12345.67	5.	12346
w.d	12345.67	8.1	12345.7
COMMA*w.d*	12345.67	COMMA8.1	12,345.7
DOLLAR*w.d*	12345.67	DOLLAR10.2	$12,345.67
DOLLAR*w.d*	12345.67	DOLLAR10.	$12,346
YEN*w.d*	12345.67	YEN7.	¥12,346
EUROX*w.d*	12345.67	EUROX10.2	€12.345,67

SAS Format Widths

If you do not specify a format width that is large enough to accommodate a numeric value, the displayed value is automatically adjusted to fit into the width. Numeric values are rounded.

Format	Stored Value	Displayed Value
DOLLAR12.2	27134.2864	$27,134.29
DOLLAR9.2	27134.2864	$27134.29
DOLLAR8.2	27134.2864	27134.29
DOLLAR5.2	27134.2864	27134
DOLLAR4.2	27134.2864	27E3

Common Formats for Date Values

Value	Format	Formatted Value
21199	DATE7.	15JAN18
21199	DATE9.	15JAN2018
21199	MMDDYY10.	01/15/2018
21199	DDMMYY8.	15/01/18
21199	MONYY7.	JAN2018
21199	MONNAME.	January
21199	WEEKDATE31.	Monday, January 15, 2018
21199	WORDDATE19.	January 15, 2018

Using Formats

This demonstration illustrates the effects of different width values on date formats.

Converting Numeric Values to Character Values

numeric to character

Stock	Date
ABC Company	21154
ABC Company	21124
ABC Company	21094
ABC Company	21063

Stock	Date	Day
ABC Company	21154	Fri
ABC Company	21124	Wed
ABC Company	21094	Mon
ABC Company	21063	Fri

```
Day=put(Date,downame3.);
```

source

format