
UCD School of Mathematics and Statistics

# STAT40840: Data programming with SAS

Laura Kirwan

Lecture 4

# 4.1 Using SAS Formats

# 4.2 User-Defined Formats

# Objectives – Part 1

- Describe SAS formats.
- Apply SAS formats with the FORMAT statement.

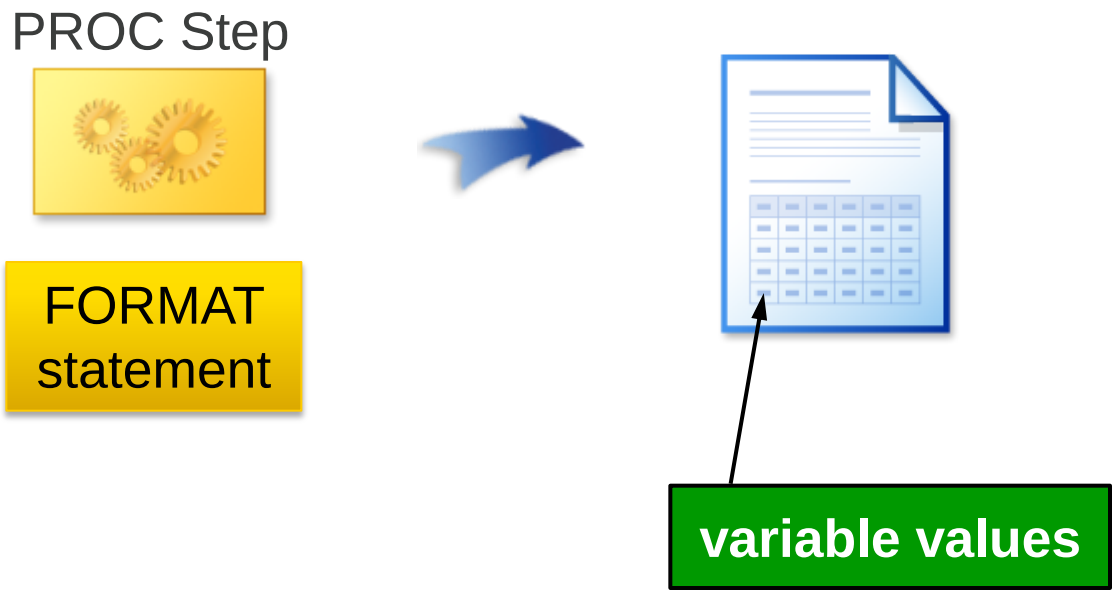
# Enhance the appearance of variable values in reports.

| Last_Name | First_ Name | Country | Job_Title | Salary | Hire_ Date |
|---|---|---|---|---|---|
| Zhou | Tom | AU | Sales Manager | 108255 | 12205 |
| Dawes | Wilson | AU | Sales Manager | 87975 | 6575 |
| Elvish | Irenie | AU | Sales Rep. II | 26600 | 6575 |

| Last_Name | First_ Name | Country | Job_Title | Salary | Hire_Date |
|---|---|---|---|---|---|
| Zhou | Tom | AU | Sales Manager | $108,255 | 06/01/1993 |
| Dawes | Wilson | AU | Sales Manager | $87,975 | 01/01/1978 |
| Elvish | Irenie | AU | Sales Rep. II | $26,600 | 01/01/1978 |

# SAS Formats

•*SAS formats* can be used in a PROC step to change how values are displayed in a report.



# FORMAT Statement

- The *FORMAT statement* associates a format with a variable.

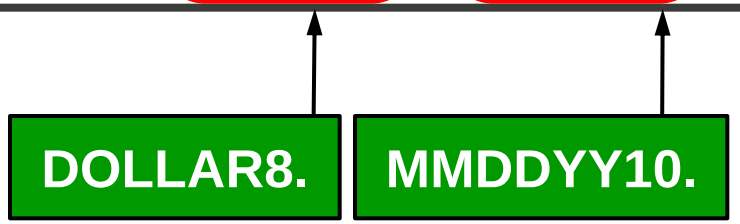
```
proc print data=orion.sales noobs;
    format Salary dollar8. Hire_Date mmddyy10.;
    var Last_Name First_Name Country
        Job_Title Salary Hire_Date;
run;
```

**FORMAT** *variable(s) format …*;

L4_D1.sas

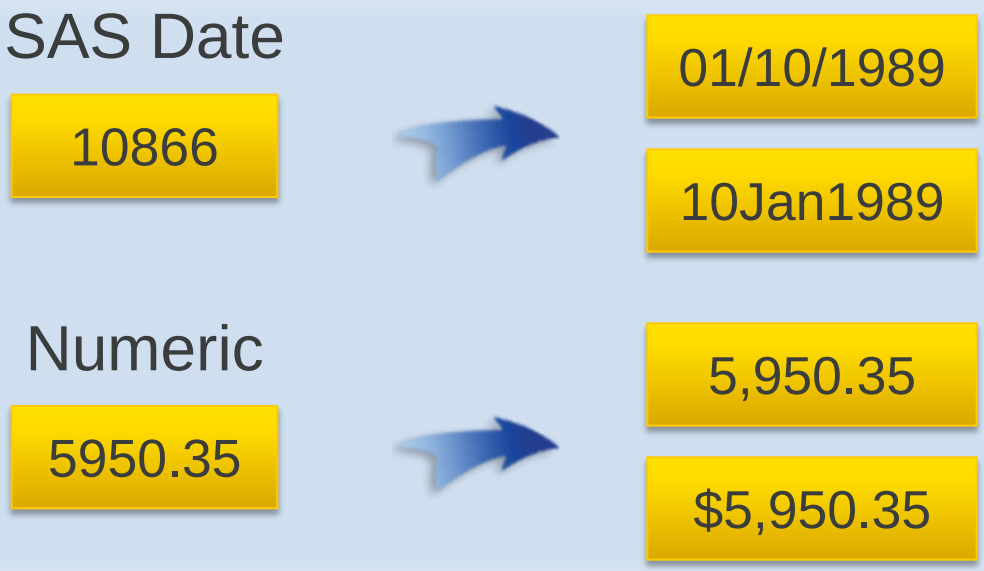
# Viewing the Output

| Last_Name | First_ Name | Country | Job_Title | Salary | Hire_Date |
|---|---|---|---|---|---|
| Zhou | Tom | AU | Sales Manager | $108,255 | 06/01/1993 |
| Dawes | Wilson | AU | Sales Manager | $87,975 | 01/01/1978 |
| Elvish | Irenie | AU | Sales Rep. II | $26,600 | 01/01/1978 |
| Ngan | Christina | AU | Sales Rep. II | $27,475 | 07/01/1982 |
| Hotstone | Kimiko | AU | Sales Rep. I | $26,190 | 10/01/1989 |



# What Is a Format?

- A *format* is an instruction to write data values.
  - A format changes the appearance of a variable's value in a report.
  - The values stored in the data set are ***not*** changed.



# SAS Formats

- SAS formats have the following form:

<$>*format*<*w*>.<*d*>

| | |
|---|---|
| $ | Indicates a character format. |
| *format* | Names the SAS format. |
| *w* | Specifies the total format width, including decimal places and special characters. |
| . | Is required syntax. Formats always contain a period (.) as part of the name. |
| *d* | Specifies the number of decimal places to display in numeric formats. |

# SAS Formats

| Format | Definition |
|---|---|
| $*w*. | Writes standard character data. |
| *w.D* | Writes standard numeric data. |
| COMMA*w.d* | Writes numeric values with a comma that separates every three digits and a period that separates the decimal fraction. |
| DOLLAR*w.d* | Writes numeric values with a leading dollar sign, a comma that separates every three digits, and a period that separates the decimal fraction. |
| COMMAX*w.d* | Writes numeric values with a period that separates every three digits and a comma that separates the decimal fraction. |
| EUROX*w.d* | Writes numeric values with a leading euro symbol (€), a period that separates every three digits, and a comma that separates the decimal fraction. |

# SAS Format Examples

| Format | Stored Value | Displayed Value |
|---|---|---|
| $4. | Programming | Prog |
| 12. | 27134.5864 | 27135 |
| 12.2 | 27134.5864 | 27134.59 |
| COMMA12.2 | 27134.5864 | 27,134.59 |
| DOLLAR12.2 | 27134.5864 | $27,134.59 |
| COMMAX12.2 | 27134.5864 | 27.134,59 |
| EUROX12.2 | 27134.5864 | €27.134,59 |

# SAS Format Examples

If the format width is not large enough to accommodate a numeric value, the displayed value is automatically adjusted to fit the width.

| Format | Stored Value | Displayed Value |
|---|---|---|
| DOLLAR12.2 | 27134.5864 | $27,134.59 |
| DOLLAR9.2 | 27134.5864 | $27134.59 |
| DOLLAR8.2 | 27134.5864 | 27134.59 |
| DOLLAR5.2 | 27134.5864 | 27135 |
| DOLLAR4.2 | 27134.5864 | 27E3 |

# Exercise 1

- Use SAS documentation or the SAS Help Facility to explore the *Zw.d* numeric format. What is it used for?

- Hint: Search for *Zw.d* or explore “Formats by Category.”

# SAS Date Format Examples

SAS date formats display SAS date values in standard date forms.

| Format | Stored Value | Displayed Value |
|---|---|---|
| MMDDYY10. | 0 | 01/01/1960 |
| MMDDYY8. | 0 | 01/01/60 |
| MMDDYY6. | 0 | 010160 |
| DDMMYY10. | 365 | 31/12/1960 |
| DDMMYY8. | 365 | 31/12/60 |
| DDMMYY6. | 365 | 311260 |

# SAS Date Format Examples

Additional date formats:

| Format | Stored Value | Displayed Value |
|---|---|---|
| DATE7. | -1 | 31DEC59 |
| DATE9. | -1 | 31DEC1959 |
| WORDDATE. | 0 | January 1, 1960 |
| WEEKDATE. | 0 | Friday, January 1, 1960 |
| MONYY7. | 0 | JAN1960 |
| YEAR4. | 0 | 1960 |

# Exercise 2

Which FORMAT statement creates the output shown below?

a.

```
format Birth_Date Hire_Date mmddyy10.
       Term_Date monyy7.;
```

b.

```
format Birth_Date Hire_Date ddmmyyyy.
       Term_Date mmmyyyy.;
```

c.

```
format Birth_Date Hire_Date ddmmyy10.
       Term_Date monyy7.;
```

output

| Birth_Date | Hire_Date | Term_Date |
|---|---|---|
| 21/05/1969 | 15/10/1992 | MAR2007 |

# Formatting Data Values

4.1 Using SAS Formats

4.2 User-Defined Formats

# Objectives – Part 2

- Use the FORMAT procedure to create user-defined formats.
- Use a FORMAT statement to apply user-defined formats in a report.
- Use formats to recode data values.
- Use formats to collapse or aggregate data.

Display country names instead of country codes in a report.

## Current Report (partial output)

| Obs | Employee_ID | Salary | Country | Birth_ Date | Hire_ Date |
|---|---|---|---|---|---|
| 1 | 120102 | $108,255 | AU | AUG1973 | JUN1993 |
| 2 | 120103 | $87,975 | AU | JAN1953 | JAN1978 |
| 3 | 120121 | $26,600 | AU | AUG1948 | JAN1978 |

## Desired Report (partial output)

| Obs | Employee_ID | Salary | Country | Birth_ Date | Hire_ Date |
|---|---|---|---|---|---|
| 1 | 120102 | $108,255 | Australia | AUG1973 | JUN1993 |
| 2 | 120103 | $87,975 | Australia | JAN1953 | JAN1978 |
| 3 | 120121 | $26,600 | Australia | AUG1948 | JAN1978 |

# User-Defined Formats: Part 1

Use PROC FORMAT to create a user-defined format.

```
proc format;
   value $ctryfmt   'AU'='Australia'
                    'US'='United States'
                   other='Miscoded';
run;
```

**PROC FORMAT;**
**VALUE** *format-name range1* **=** *'label '*
*range2* **=** *'label '*
*. . .* **;**
**RUN;**

L4_D2.sas

# User-Defined Formats: Part 2

Use a FORMAT statement in the PROC PRINT step to apply the format to a specific variable.

```
proc print data=orion.sales;
   var Employee_ID Salary Country
       Birth_Date Hire_Date;
   format Salary dollar10.
          Birth_Date Hire_Date monyy7.
          Country $ctryfmt.;
run;
```

L4_D2.sas

# Viewing the Output

| Obs | Employee_ID | Salary | Country | Birth_ Date | Hire_ Date |
|---|---|---|---|---|---|
| 1 | 120102 | $108,255 | Australia | AUG1973 | JUN1993 |
| 2 | 120103 | $87,975 | Australia | JAN1953 | JAN1978 |
| 3 | 120121 | $26,600 | Australia | AUG1948 | JAN1978 |
| 4 | 120122 | $27,475 | Australia | JUL1958 | JUL1982 |
| 5 | 120123 | $26,190 | Australia | SEP1968 | OCT1989 |

# VALUE Statement

**VALUE** *format-name range1='label'*
*range2='label'*
*. . .* ;

A format name

- can be up to 32 characters in length
- for character formats, must begin with a dollar sign ($), followed by a letter or underscore
- for numeric formats, must begin with a letter or underscore
- cannot end in a number
- cannot be given the name of a SAS format
- cannot include a period in the VALUE statement.

# VALUE Statement

**VALUE** *format-name range1='label' range2='label' . . . ;*

Each range can be

- a single value
- a range of values
- a list of values.

Labels

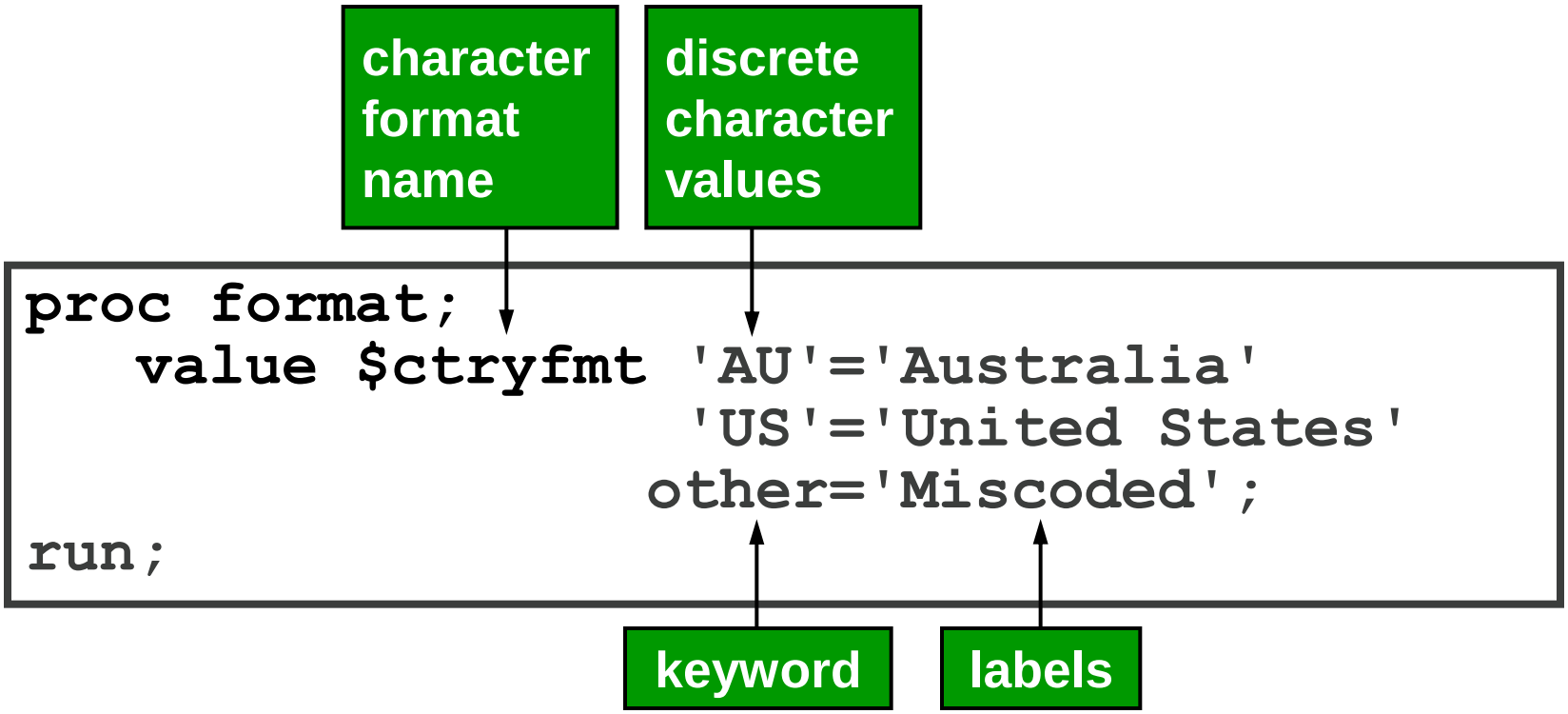
- can be up to 32,767 characters in length
- are enclosed in quotation marks.

# Exercise 3

Which names are invalid for user-defined formats?

a. $stfmt
b. $3levels
c. _4years
d. salranges
e. dollar

# Defining a Character Format




The OTHER keyword includes all values that do not match any other value or range.

L4_D2.sas

# Applying a Format

User-defined and SAS formats can be applied in a single FORMAT statement.

```
proc print data=orion.sales label;
   var Employee_ID Salary Country
       Birth_Date Hire_Date;
   format Salary dollar10.
          Birth_Date Hire_Date monyy7.
          Country $ctryfmt.;
run;
```

A period (for example, at the end of the $CTRYFMT format) is required when user-defined formats are used in a FORMAT statement.

L4_D2.sas

An Orion Star manager wants a report that shows employee salaries collapsed into three user-defined groups or tiers.

| Obs | Employee_ID | Last_Name | Salary |
|---|---|---|---|
| 1 | 120102 | Zhou | 108255 |
| 2 | 120103 | Dawes | 87975 |
| 3 | 120121 | Elvish | 26600 |
| 4 | 120122 | Ngan | 27475 |

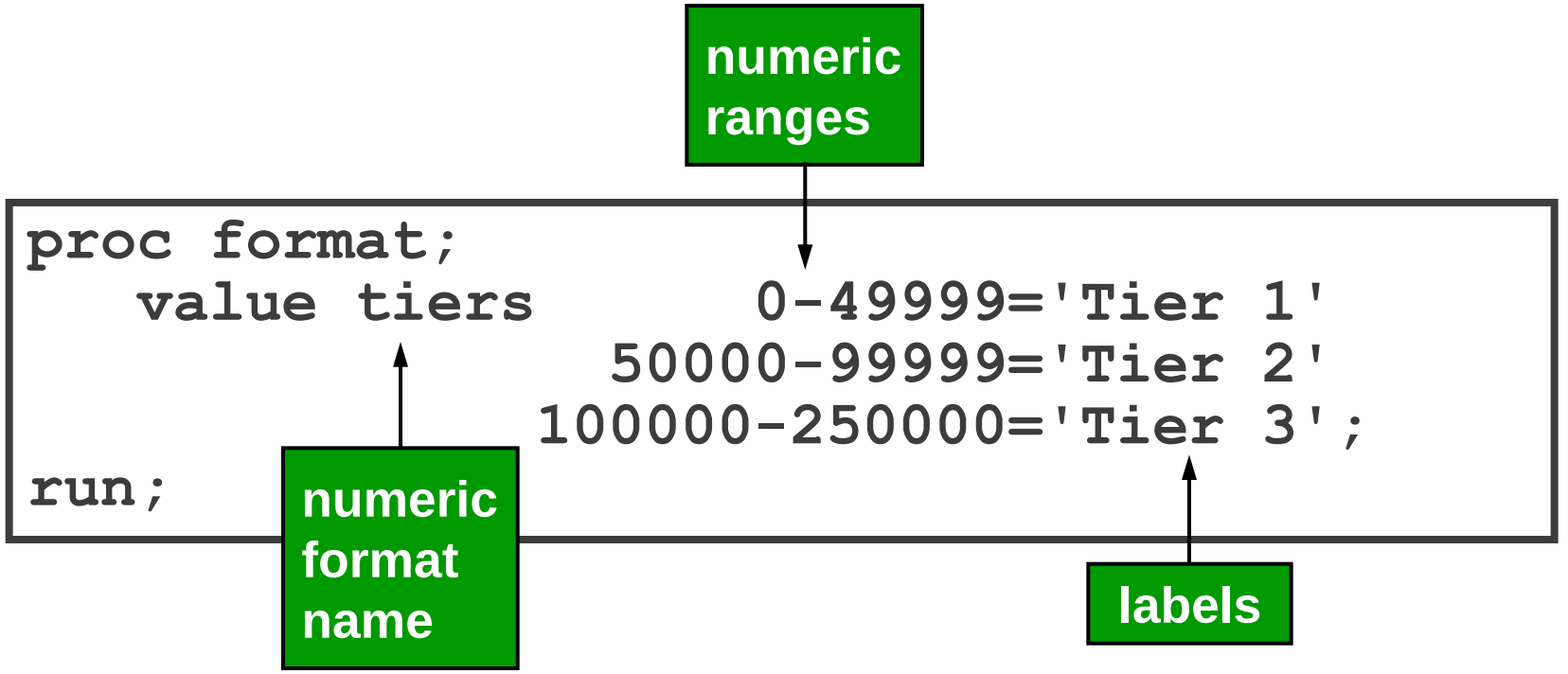
| Obs | Employee_ID | Last_Name | Salary |
|---|---|---|---|
| 1 | 120102 | Zhou | Tier 3 |
| 2 | 120103 | Dawes | Tier 2 |
| 3 | 120121 | Elvish | Tier 1 |
| 4 | 120122 | Ngan | Tier 1 |

# Specifying Ranges of Values

Use PROC FORMAT to specify the salary range for each tier.


| Salary | Value |
|---|---|
| 0 to 49,999 | Tier1 |
| 50,000 to 99,999 | Tier2 |
| 100,000 to 250,000 | Tier3 |

# Defining a Numeric Format




L4_D3.sas

# Defining a Continuous Range

The less than (<) symbol excludes the endpoint from a range, which enables a continuous range.

- Put < after the starting value in a range to exclude it.
- Put < before the ending value in a range to exclude it.

| Range | Starting Value | Ending Value |
|---|---|---|
| 50000 - 100000 | Includes 50000 | Includes 100000 |
| 50000 - < 100000 | Includes 50000 | Excludes 100000 |
| 50000 < - 100000 | Excludes 50000 | Includes 100000 |
| 50000 < - < 100000 | Excludes 50000 | Excludes 100000 |

# Exercise 4

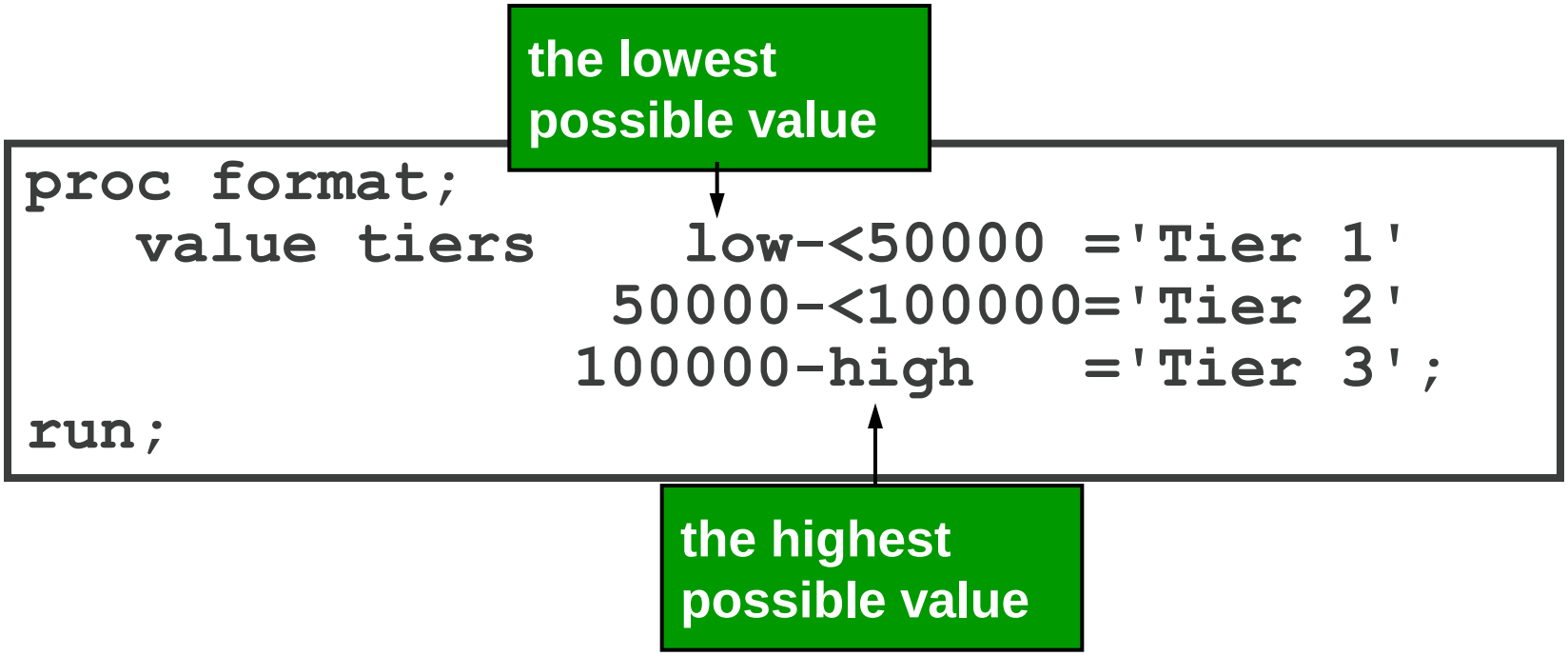
How is a value of *50000* displayed if the TIERS format below is applied to the value?

a. Tier 1
b. Tier 2
c. 50000
d. a missing value

```
proc format;
   value tiers  20000-<50000 ='Tier 1'
                50000-<100000='Tier 2'
                100000-250000='Tier 3';
run;
```

L4_E4.sas

# LOW and HIGH Keywords




The LOW keyword

L4_D4.sas

- includes missing values for character variables
- does not include missing values for numeric variables.

# Applying a Numeric Format

Part 1

```
proc format;
   value tiers      low-<50000 ='Tier 1'
                50000-<100000='Tier 2'
                100000-high  ='Tier 3';
run;
```

Part 2

```
proc print data=orion.sales;
   var Employee_ID Job_Title Salary
       Country Birth_Date Hire_Date;
   format Birth_Date Hire_Date monyy7.
          Salary tiers.;
run;
```

L4_D4.sas

# Viewing the Output

| Obs | Employee_ID | Job_Title | Salary | Country | Birth_ Date | Hire_ Date |
|---|---|---|---|---|---|---|
| 1 | 120102 | Sales Manager | Tier 3 | AU | AUG1973 | JUN1993 |
| 2 | 120103 | Sales Manager | Tier 2 | AU | JAN1953 | JAN1978 |
| 3 | 120121 | Sales Rep. II | Tier 1 | AU | AUG1948 | JAN1978 |
| 4 | 120122 | Sales Rep. II | Tier 1 | AU | JUL1958 | JUL1982 |
| 5 | 120123 | Sales Rep. I | Tier 1 | AU | SEP1968 | OCT1989 |

# User-Defined Format Example

Ranges can be specified using lists, ranges, discrete values, and keywords.

```
proc format;
   value mnthfmt 1,2,3='Qtr 1'
                 4-6='Qtr 2'
                 7-9='Qtr 3'
                10-12='Qtr 4'
                    .='missing'
                other='unknown';
run;
```

# Multiple User-Defined Formats

Multiple VALUE statements can be included in a single PROC FORMAT step.

```
proc format;
   value $ctryfmt   'AU'='Australia'
                    'US'='United States'
                   other='Miscoded';

   value tiers       low-<50000 ='Tier 1'
                  50000-<100000='Tier 2'
                  100000-high  ='Tier 3';
run;
```

L4_D5.sas

# Viewing the Output

```
proc print data=orion.sales;
   var Employee_ID Job_Title Salary
       Country Birth_Date Hire_Date;
   format Birth_Date Hire_Date monyy7.
          Country $ctryfmt.
          Salary tiers.;
run;
```

L4_D5.sas

| Obs | Employee_ID | Job_Title | Salary | Country | Birth_ Date | Hire_ Date |
|---|---|---|---|---|---|---|
| 1 | 120102 | Sales Manager | Tier 3 | Australia | AUG1973 | JUN1993 |
| 2 | 120103 | Sales Manager | Tier 2 | Australia | JAN1953 | JAN1978 |
| 3 | 120121 | Sales Rep. II | Tier 1 | Australia | AUG1948 | JAN1978 |
| 4 | 120122 | Sales Rep. II | Tier 1 | Australia | JUL1958 | JUL1982 |
| 5 | 120123 | Sales Rep. I | Tier 1 | Australia | SEP1968 | OCT1989 |