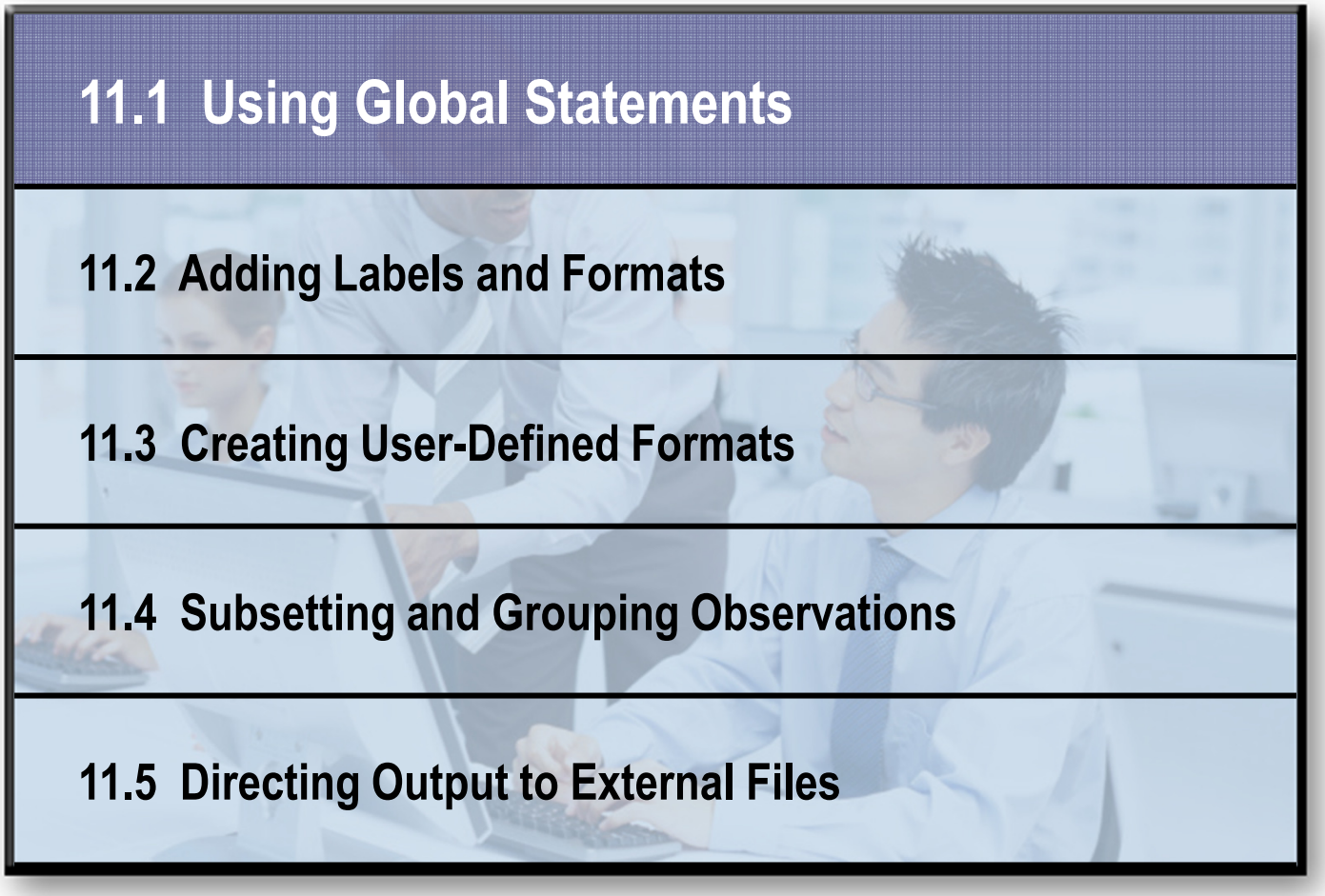
Chapter 11: Enhancing Reports

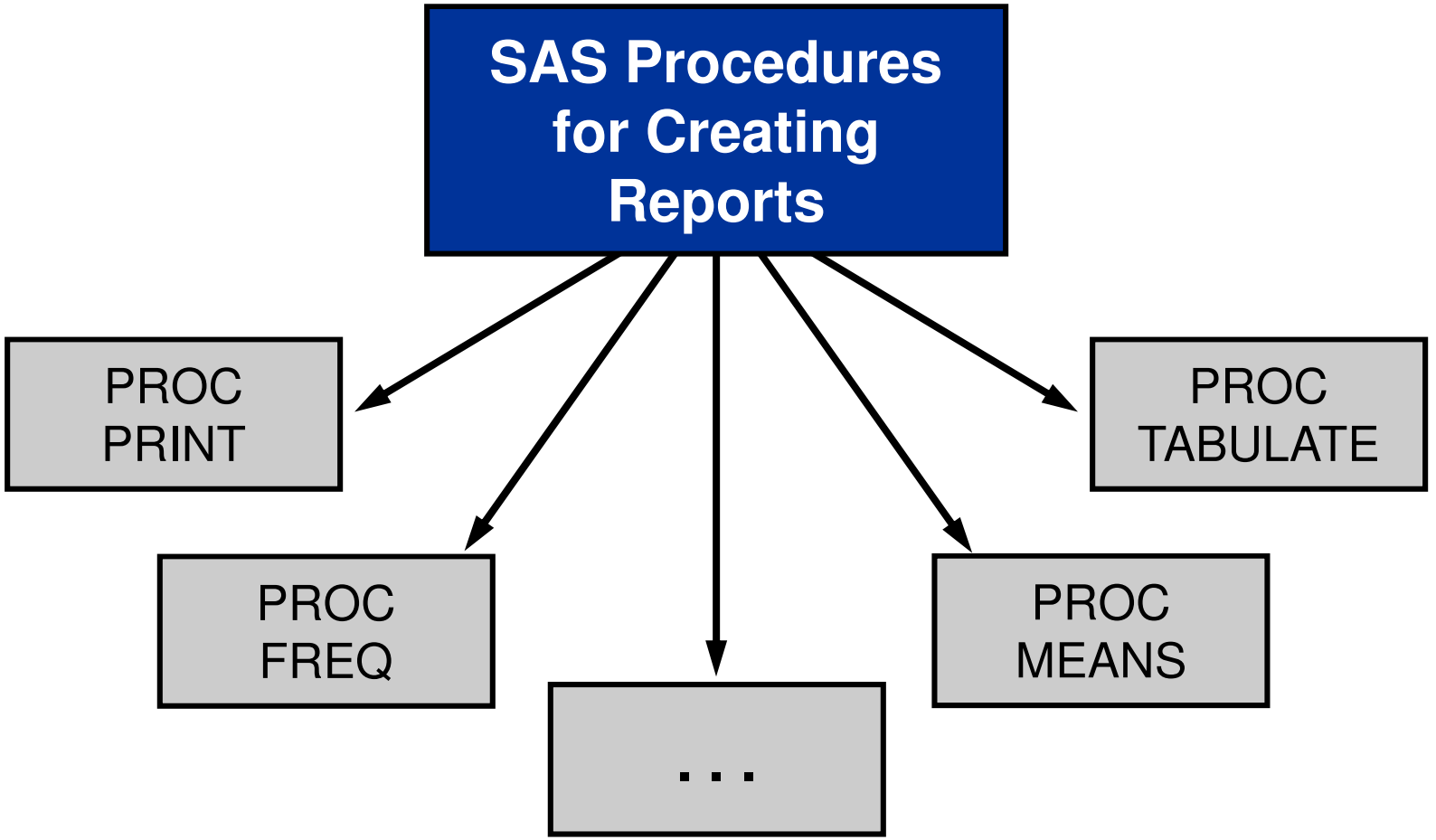


Objectives

- Identify SAS statements that are used with most reporting procedures.
- Enhance reports by using SAS system options.
- Enhance reports by adding titles and footnotes.
- Add dates and times to titles. (Self-Study)

Creating Reports

A procedure step is a primary method for creating reports.



Example of a Basic Report

```
proc print data=orion.sales;
   var Employee_ID First_Name Last_Name Salary;
run;
```

Partial PROC PRINT Output

Obs	Employee_ID	First_ Name	Last_Name	Salary
1	120102	Tom	Zhou	108255
2	120103	Wilson	Dawes	87975
3	120121	Irenie	Elvish	26600
4	120122	Christina	Ngan	27475
5	120123	Kimiko	Hotstone	26190
6	120124	Lucian	Daymond	26480
7	120125	Fong	Hofmeister	32040
8	120126	Satyakam	Denny	26780
9	120127	Sharryn	Clarkson	28100
10	120128	Monica	Kletschkus	30890

Example of an Enhanced Report

```
options nocenter;
ods html file='enhanced.html' style=sasweb;
proc print data=orion.sales label;
   var Employee_ID First_Name Last_Name Salary;
   title1 'Orion Sales Employees';
   title2 'Males Only';
   footnote 'Confidential';
   label Employee_ID='Sales ID'
         First_Name='First Name'
         Last_Name='Last Name'
         Salary='Annual Salary';
   format Salary dollar8.;
   where Gender='M';
   by Country;
run;
ods html close;
```

Example of an Enhanced Report

Partial PROC PRINT Output

Orion Sales Employees
Males Only

Country=AU

Obs	Sales ID	First Name	Last Name	Annual Salary
1	120102	Tom	Zhou	$108,255
2	120103	Wilson	Dawes	$87,975
6	120124	Lucian	Daymond	$26,480
7	120125	Fong	Hofmeister	$32,040
8	120126	Satyakam	Denny	$26,780
11	120129	Alvin	Roebuck	$30,070
12	120130	Kevin	Lyon	$26,955
13	120131	Marinus	Surawski	$26,910
16	120134	Sian	Shannan	$28,015
17	120135	Alexei	Platts	$32,490
18	120136	Atul	Leyden	$26,605

Statements That Enhance Reports

Many statements are used with most reporting procedures to enhance the report.

```
options nocenter;
ods html file='enhanced.html' style=sasweb;
proc print data=orion.sales label;
   var Employee_ID First_Name Last_Name Salary;
   title1 'Orion Sales Employees';
   title2 'Males Only';
   footnote 'Confidential';
   label Employee_ID='Sales ID'
         First_Name='First Name'
         Last_Name='Last Name'
         Salary='Annual Salary';
   format Salary dollar8.;
   where Gender='M';
   by Country;
run;
ods html close;
```

Global Statements

The following are global statements that enhance reports:

- OPTIONS
- TITLE
- FOOTNOTE
- ODS

Global statements are specified anywhere in your SAS program and they remain in effect until canceled, changed, or your SAS session ends.

The OPTIONS Statement

The *OPTIONS statement* changes the value of one or more SAS system options.

General form of the OPTIONS statement:

OPTIONS *option(s)*;

- Some SAS system options change the appearance of a report.
- The OPTIONS statement is **not** usually included in a PROC or DATA step.

SAS System Options for Reporting

Selected SAS System Options:

DATE (default)	displays the date and time that the SAS session began at the top of each page of SAS output.
NODATE	does not display the date and time that the SAS session began at the top of each page of SAS output.
NUMBER (default)	prints page numbers on the first line of each page of SAS output.
NONUMBER	does not print page numbers on the first line of each page of SAS output.
PAGENO=*n*	defines a beginning page number (*n*) for the next page of SAS output.

continued...

SAS System Options for Reporting

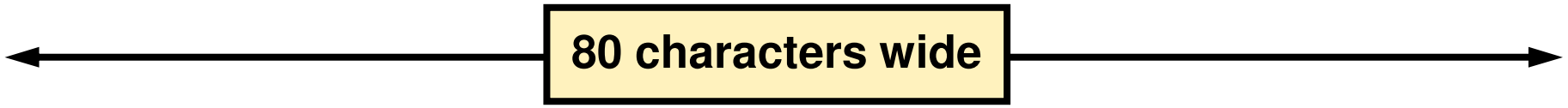
Selected SAS System Options:

CENTER (default)	centers SAS output.
NOCENTER	left-aligns SAS output.
PAGESIZE=*n* PS=*n*	defines the number of lines (*n*) that can be printed per page of SAS output.
LINESIZE=*width* LS=*width*	defines the line size (*width*) for the SAS log and SAS output.

SAS System Options for Reporting

```
options ls=80 date number;

proc means data=orion.sales;
   var Salary;
run;
```

```
                                                09:11 Monday, January 14, 2008   35

                                The MEANS Procedure

                                Analysis Variable : Salary

        N            Mean         Std Dev         Minimum         Maximum
-------------------------------------------------------------------------------
      165        31160.12        20082.67        22710.00       243190.00
-------------------------------------------------------------------------------
```



p111d02

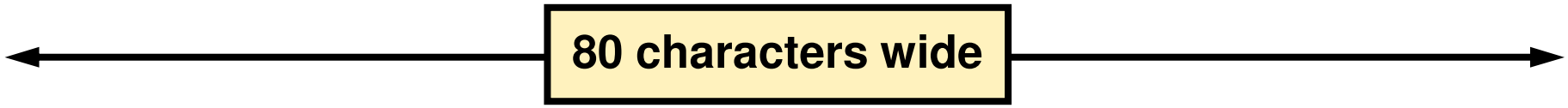
SAS System Options for Reporting

```
options nodate pageno=1;

proc freq data=orion.sales;
   tables Country;
run;
```

1

The FREQ Procedure

Country	Frequency	Percent	Cumulative Frequency	Cumulative Percent
AU	63	38.18	63	38.18
US	102	61.82	165	100.00



p111d02

Setup for the Poll

- Retrieve and submit program **p111a01**.
- Review the results including the date, time, and page number in the top-right corner of each page of output.
- Add the DTRESET system option to the OPTIONS statement.
- Submit the program and review the results.

DTRESET	updates date and time at the top of each page of SAS output.
NODTRESET (Default)	does not update date and time at the top of each page of SAS output.

11.01 Poll

Did the date and/or time change?

- Yes
- No

The TITLE Statement

The *TITLE statement* specifies title lines for SAS output.

General form of the TITLE statement:

TITLE*n* '*text* '**;**

- Titles appear at the top of the page.
- The default title is **The SAS System**.
- The value of ***n*** can be from 1 to 10.
- An unnumbered **TITLE** is equivalent to **TITLE1**.
- Titles remain in effect until they are changed, canceled, or you end your SAS session.

The FOOTNOTE Statement

The *FOOTNOTE statement* specifies footnote lines for SAS output.

General form of the FOOTNOTE statement:

FOOTNOTE*n* '*text* '**;**

- Footnotes appear at the bottom of the page.
- No footnote is printed unless one is specified.
- The value of ***n*** can be from 1 to 10.
- An unnumbered **FOOTNOTE** is equivalent to FOOTNOTE1.
- Footnotes remain in effect until they are changed, canceled, or you end your SAS session.

The TITLE and FOOTNOTE Statements

```
footnote1 'By Human Resource Department';
footnote3 'Confidential';

proc means data=orion.sales;
   var Salary;
   title 'Orion Star Sales Employees';
run;
```

The TITLE and FOOTNOTE Statements

Orion Star Sales Employees

The MEANS Procedure

Analysis Variable : Salary

N	Mean	Std Dev	Minimum	Maximum
165	31160.12	20082.67	22710.00	243190.00

By Human Resource Department

Confidential

Changing Titles and Footnotes

TITLE*n* or **FOOTNOTE*n***

- replaces a previous title or footnote with the same number
- cancels all titles or footnotes with higher numbers.

Canceling All Titles and Footnotes

- The null TITLE statement cancels all titles.

```
title;
```

- The null FOOTNOTE statement cancels all footnotes.

```
footnote;
```

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`proc print data=orion.sales;`
`  title1 'The First Line';`
`  title2 'The Second Line';`
`run;`	

`proc print data=orion.sales;`
`  title2 'The Next Line';`
`run;`	

`proc print data=orion.sales;`
`  title 'The Top Line';`
`run;`	

`proc print data=orion.sales;`
`  title3 'The Third Line';`
`run;`	

`proc print data=orion.sales;`
`  title;`
`run;`	

...

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`proc print data=orion.sales;`
`  title1 'The First Line';`
`  title2 'The Second Line';`
`run;`	

`proc print data=orion.sales;`
`  title2 'The Next Line';`
`run;`	

`proc print data=orion.sales;`
`  title 'The Top Line';`
`run;`	

`proc print data=orion.sales;`
`  title3 'The Third Line';`
`run;`	

`proc print data=orion.sales;`
`  title;`
`run;`	

...

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`proc print data=orion.sales;`
`  title1 'The First Line';`
`  title2 'The Second Line';`
`run;`
	The First Line
	The Second Line

`proc print data=orion.sales;`
`  title2 'The Next Line';`
`run;`	

`proc print data=orion.sales;`
`  title 'The Top Line';`
`run;`	

`proc print data=orion.sales;`
`  title3 'The Third Line';`
`run;`	

`proc print data=orion.sales;`
`  title;`
`run;`	

...

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`proc print data=orion.sales;`
`  title1 'The First Line';`
`  title2 'The Second Line';`
`run;`
	The First Line
	The Second Line

`proc print data=orion.sales;`
`  title2 'The Next Line';`
`run;`	

`proc print data=orion.sales;`
`  title 'The Top Line';`
`run;`	

`proc print data=orion.sales;`
`  title3 'The Third Line';`
`run;`	

`proc print data=orion.sales;`
`  title;`
`run;`	

...

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`proc print data=orion.sales;`
`  title1 'The First Line';`
`  title2 'The Second Line';`
`run;`
	The First Line
	The Second Line

`proc print data=orion.sales;`
`  title2 'The Next Line';`
`run;`
	The First Line
	The Next Line

`proc print data=orion.sales;`
`  title 'The Top Line';`
`run;`	

`proc print data=orion.sales;`
`  title3 'The Third Line';`
`run;`	

`proc print data=orion.sales;`
`  title;`
`run;`	

...

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`proc print data=orion.sales;`
`  title1 'The First Line';`
`  title2 'The Second Line';`
`run;`
	The First Line
	The Second Line

`proc print data=orion.sales;`
`  title2 'The Next Line';`
`run;`
	The First Line
	The Next Line

`proc print data=orion.sales;`
`  title 'The Top Line';`
`run;`	

`proc print data=orion.sales;`
`  title3 'The Third Line';`
`run;`	

`proc print data=orion.sales;`
`  title;`
`run;`	

...

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`proc print data=orion.sales;`
`  title1 'The First Line';`
`  title2 'The Second Line';`
`run;`
	The First Line
	The Second Line

`proc print data=orion.sales;`
`  title2 'The Next Line';`
`run;`
	The First Line
	The Next Line

`proc print data=orion.sales;`
`  title 'The Top Line';`
`run;`
	The Top Line

`proc print data=orion.sales;`
`  title3 'The Third Line';`
`run;`	

`proc print data=orion.sales;`
`  title;`
`run;`	

...

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`proc print data=orion.sales;`
`  title1 'The First Line';`
`  title2 'The Second Line';`
`run;`
	The First Line
	The Second Line

`proc print data=orion.sales;`
`  title2 'The Next Line';`
`run;`
	The First Line
	The Next Line

`proc print data=orion.sales;`
`  title 'The Top Line';`
`run;`
	The Top Line

`proc print data=orion.sales;`
`  title3 'The Third Line';`
`run;`	

`proc print data=orion.sales;`
`  title;`
`run;`	

...

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`proc print data=orion.sales;`
`  title1 'The First Line';`
`  title2 'The Second Line';`
`run;`
	The First Line
	The Second Line

`proc print data=orion.sales;`
`  title2 'The Next Line';`
`run;`
	The First Line
	The Next Line

`proc print data=orion.sales;`
`  title 'The Top Line';`
`run;`
	The Top Line

`proc print data=orion.sales;`
`  title3 'The Third Line';`
`run;`
	The Top Line
	
	The Third Line

`proc print data=orion.sales;`
`  title;`
`run;`	

...

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`proc print data=orion.sales;`
`  title1 'The First Line';`
`  title2 'The Second Line';`
`run;`
	The First Line
	The Second Line

`proc print data=orion.sales;`
`  title2 'The Next Line';`
`run;`
	The First Line
	The Next Line

`proc print data=orion.sales;`
`  title 'The Top Line';`
`run;`
	The Top Line

`proc print data=orion.sales;`
`  title3 'The Third Line';`
`run;`
	The Top Line
	
	The Third Line

`proc print data=orion.sales;`
`  title;`
`run;`	

...

Changing and Canceling Titles and Footnotes

PROC PRINT Code	Resultant Title(s)

`proc print data=orion.sales;`
`  title1 'The First Line';`
`  title2 'The Second Line';`
`run;`
	The First Line
	The Second Line

`proc print data=orion.sales;`
`  title2 'The Next Line';`
`run;`
	The First Line
	The Next Line

`proc print data=orion.sales;`
`  title 'The Top Line';`
`run;`
	The Top Line

`proc print data=orion.sales;`
`  title3 'The Third Line';`
`run;`
	The Top Line
	
	The Third Line

`proc print data=orion.sales;`
`  title;`
`run;`	

11.02 Quiz

Which footnote(s) appears in the second procedure output?

a.
```
Non Sales Employees
```

b.
```
Orion Star
Non Sales Employees
```

c.
```
Non Sales Employees
Confidential
```

d.
```
Orion Star
Non Sales Employees
Confidential
```

```
footnote1 'Orion Star';
proc print data=orion.sales;
   footnote2 'Sales Employees';
   footnote3 'Confidential';
run;
proc print data=orion.nonsales;
   footnote2 'Non Sales Employees';
run;
```

Titles with Dates and Times (Self-Study)

The automatic macro variables &SYSDATE9 and &SYSTIME can be used to add the SAS invocation date and time to titles and footnotes.

```
title1 'Orion Star Employee Listing';
title2 "Created on &sysdate9 at &systime";
```

Double quotation marks must be used when you reference a macro variable.

Example Title Output:

```
                    Orion Star Employee Listing
                  Created on 11MAR2008 at 15:53
```

Titles with Dates and Times (Self-Study)

The %LET statement can be used with %SYSFUNC and the TODAY function or the TIME function to create a macro variable with the current date or time.

%LET *macro-variable* **= %SYSFUNC(today(),** *date-format***);**

%LET *macro-variable* **= %SYSFUNC(time(),** *time-format***);**

- %LET is a macro statement that creates a macro variable and assigns it a value without leading or trailing blanks.
- %SYSFUNC is a macro function that executes SAS functions outside of a step.

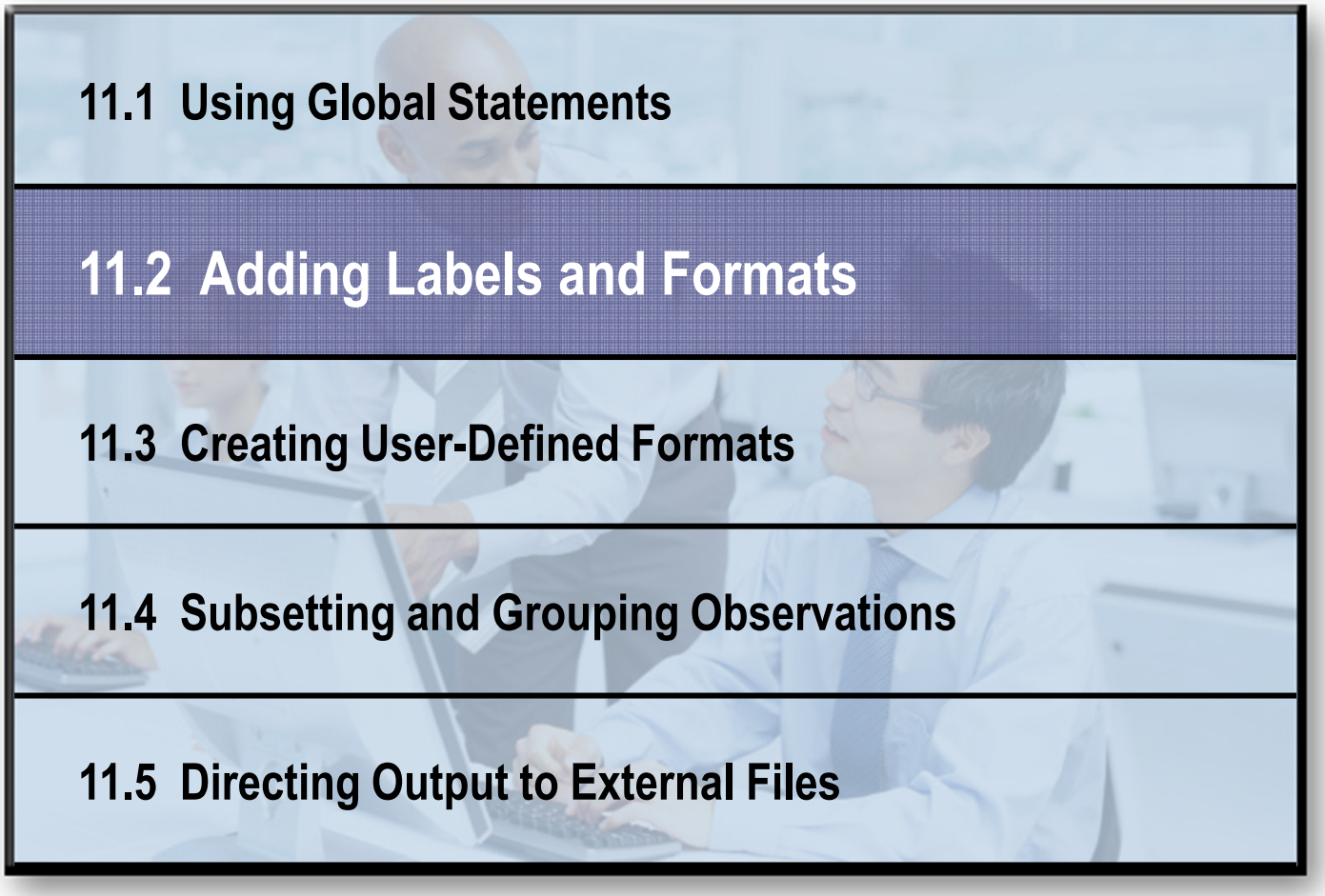
Titles with Dates and Times (Self-Study)

```
%let currentdate=%sysfunc(today(),worddate.);
%let currenttime=%sysfunc(time(),timeampm.);

proc freq data=orion.sales;
   tables Gender Country;
   title1 'Orion Star Employee Listing';
   title2 "Created &currentdate";
   title3 "at &currenttime";
run;
```

Example Title Output:

```
                Orion Star Employee Listing
                  Created March 11, 2008
                       at 4:09:43 PM
```

Chapter 11: Enhancing Reports



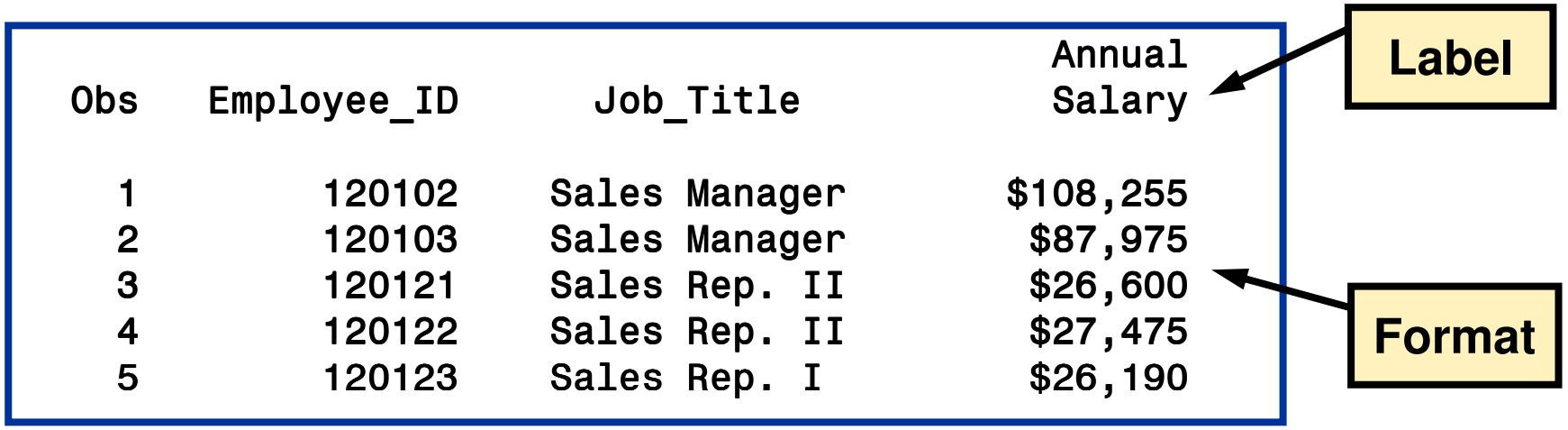
Objectives

- Display descriptive column headings using the LABEL statement.
- Display formatted values using the FORMAT statement.

Labels and Formats (Review)

When displaying reports,

- a *label* changes the appearance of a variable name
- a *format* changes the appearance of variable value.



Obs	Employee_ID	Job_Title	Annual Salary
1	120102	Sales Manager	$108,255
2	120103	Sales Manager	$87,975
3	120121	Sales Rep. II	$26,600
4	120122	Sales Rep. II	$27,475
5	120123	Sales Rep. I	$26,190

The LABEL Statement (Review)

The *LABEL statement* assigns descriptive labels to variable names.

General form of the LABEL statement:

LABEL *variable* = '*label*'
variable = '*label*'
variable = '*label*';

- A label can be up to 256 characters.
- Labels are used automatically by many procedures.
- The PRINT procedure uses labels when the LABEL or SPLIT= option is specified in the PROC PRINT statement.

Assigning Temporary Labels

PROC FREQ automatically uses labels.

```
proc freq data=orion.sales;
   tables Gender;
   label Gender='Sales Employee Gender';
run;
```

The FREQ Procedure

Sales Employee Gender

Gender	Frequency	Percent	Cumulative Frequency	Cumulative Percent
F	68	41.21	68	41.21
M	97	58.79	165	100.00

Assigning Temporary Labels

PROC PRINT does not automatically use labels.

```
proc print data=orion.sales;
   var Employee_ID Job_Title Salary;
   label Employee_ID='Sales ID'
         Job_Title='Job Title'
         Salary='Annual Salary';
run;
```

Partial PROC PRINT Output

Obs	Employee_ID	Job_Title	Salary
1	120102	Sales Manager	108255
2	120103	Sales Manager	87975
3	120121	Sales Rep. II	26600
4	120122	Sales Rep. II	27475
5	120123	Sales Rep. I	26190

p111d05

Assigning Temporary Labels

The LABEL option tells PROC PRINT to use labels.

```
proc print data=orion.sales label;
   var Employee_ID Job_Title Salary;
   label Employee_ID='Sales ID'
         Job_Title='Job Title'
         Salary='Annual Salary';
run;
```

Partial PROC PRINT Output

Obs	Sales ID	Job Title	Annual Salary
1	120102	Sales Manager	108255
2	120103	Sales Manager	87975
3	120121	Sales Rep. II	26600
4	120122	Sales Rep. II	27475
5	120123	Sales Rep. I	26190

p111d05

Assigning Temporary Labels

Instead of the LABEL option in PROC PRINT, the SPLIT= option can be used.

The *SPLIT= option* specifies the split character, which controls line breaks in column headers.

General form of the SPLIT= option:

SPLIT='*split-character***'**

Assigning Temporary Labels

The SPLIT= option makes PROC PRINT use labels.

```
proc print data=orion.sales split='*';
   var Employee_ID Job_Title Salary;
   label Employee_ID='Sales ID'
         Job_Title='Job*Title'
         Salary='Annual*Salary';
run;
```

Partial PROC PRINT Output

Obs	Sales ID	Job Title	Annual Salary
1	120102	Sales Manager	108255
2	120103	Sales Manager	87975
3	120121	Sales Rep. II	26600
4	120122	Sales Rep. II	27475
5	120123	Sales Rep. I	26190

p111d05

Assigning Permanent Labels (Review)

Using a LABEL statement in a DATA step permanently associates labels with variables by storing the label in the descriptor portion of the SAS data set.

```
data orion.bonus;
   set orion.sales;
   Bonus=Salary*0.10;
   label Salary='Annual*Salary'
         Bonus='Annual*Bonus';
   keep Employee_ID First_Name
        Last_Name Salary Bonus;
run;

proc print data=orion.bonus split='*';
run;
```

Assigning Permanent Labels (Review)

Partial PROC PRINT Output

Obs	Employee_ID	First_ Name	Last_Name	Annual Salary	Annual Bonus
1	120102	Tom	Zhou	108255	10825.5
2	120103	Wilson	Dawes	87975	8797.5
3	120121	Irenie	Elvish	26600	2660.0
4	120122	Christina	Ngan	27475	2747.5
5	120123	Kimiko	Hotstone	26190	2619.0
6	120124	Lucian	Daymond	26480	2648.0
7	120125	Fong	Hofmeister	32040	3204.0
8	120126	Satyakam	Denny	26780	2678.0
9	120127	Sharryn	Clarkson	28100	2810.0
10	120128	Monica	Kletschkus	30890	3089.0

11.03 Quiz

Which statement is true concerning the PROC PRINT output for **Bonus**?

a. Annual Bonus will be the label.

b. Mid-Year Bonus will be the label.

```
data orion.bonus;
   set orion.sales;
   Bonus=Salary*0.10;
   label Bonus='Annual Bonus';
run;

proc print data=orion.bonus label;
   label Bonus='Mid-Year Bonus';
run;
```

p111d05

The FORMAT Statement (Review)

The *FORMAT statement* assigns formats to variable values.

General form of the FORMAT statement:

FORMAT *variable(s) format*;

- A *format* is an instruction that SAS uses to write data values.
- Values in the data set are not changed.

11.04 Quiz

Which displayed value is incorrect for the given format?

Format	Stored Value	Displayed Value
$3.	Wednesday	Wed
6.1	1234.345	1234.3
COMMAX5.	1234.345	1.234
DOLLAR9.2	1234.345	$1,234.35
DDMMYY8.	0	01/01/1960
DATE9.	0	01JAN1960
YEAR4.	0	1960

Assigning Temporary Formats

```
proc print data=orion.sales label;
   var Employee_ID Job_Title Salary
       Country Birth_Date Hire_Date;
   . . .
   format Salary dollar10.0
          Birth_Date Hire_Date monyy7.;
run;
```

Partial PROC PRINT Output

Obs	Sales ID	Job Title	Annual Salary	Country	Date of Birth	Date of Hire
1	120102	Sales Manager	$108,255	AU	AUG1969	JUN1989
2	120103	Sales Manager	$87,975	AU	JAN1949	JAN1974
3	120121	Sales Rep. II	$26,600	AU	AUG1944	JAN1974
4	120122	Sales Rep. II	$27,475	AU	JUL1954	JUL1978
5	120123	Sales Rep. I	$26,190	AU	SEP1964	OCT1985

Assigning Temporary Formats

```
proc freq data=orion.sales;
   tables Hire_Date;
   format Hire_Date year4.;
run;
```

Partial PROC FREQ Output

The FREQ Procedure

Hire_Date	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1974	23	13.94	23	13.94
1975	2	1.21	25	15.15
1976	4	2.42	29	17.58
1977	3	1.82	32	19.39
1978	7	4.24	39	23.64
1979	3	1.82	42	25.45

Assigning Permanent and Temporary Formats

Using a FORMAT statement in a DATA step permanently associates formats with variables by storing the format in the descriptor portion of the SAS data set.

```
data orion.bonus;
   set orion.sales;
   Bonus=Salary*0.10;
   format Salary Bonus comma8.;
   keep Employee_ID First_Name
        Last_Name Salary Bonus;
run;

proc print data=orion.bonus;
   format Bonus dollar8.;
run;
```

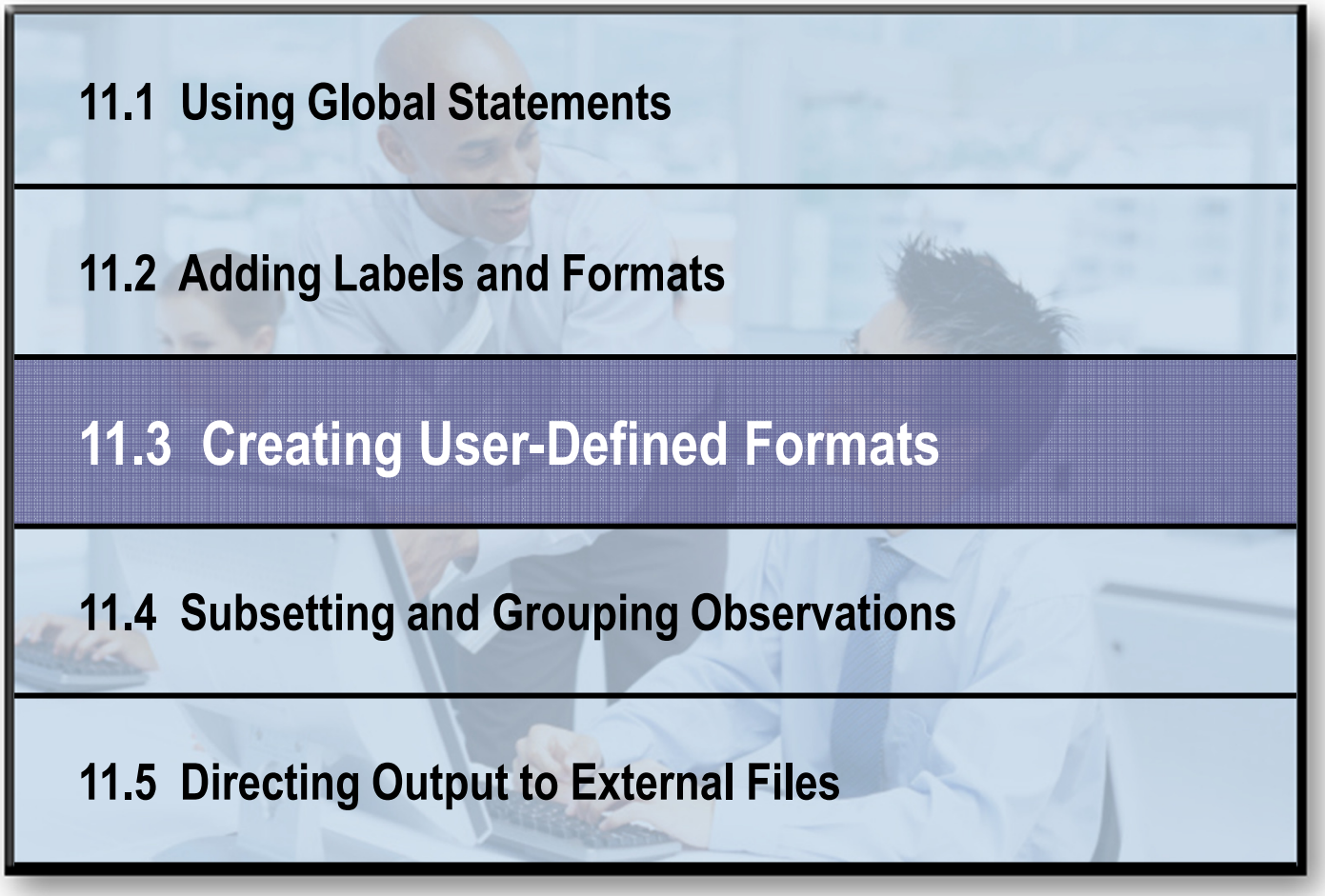
- Temporary formats override permanent formats.

Assigning Permanent and Temporary Formats

Partial PROC PRINT Output

Obs	Employee_ID	First_ Name	Last_Name	Salary	Bonus
1	120102	Tom	Zhou	108,255	$10,826
2	120103	Wilson	Dawes	87,975	$8,798
3	120121	Irenie	Elvish	26,600	$2,660
4	120122	Christina	Ngan	27,475	$2,748
5	120123	Kimiko	Hotstone	26,190	$2,619
6	120124	Lucian	Daymond	26,480	$2,648
7	120125	Fong	Hofmeister	32,040	$3,204
8	120126	Satyakam	Denny	26,780	$2,678
9	120127	Sharryn	Clarkson	28,100	$2,810
10	120128	Monica	Kletschkus	30,890	$3,089

Chapter 11: Enhancing Reports



Objectives

- Create user-defined formats using the FORMAT procedure.
- Apply user-defined formats to variables in reports.

User-Defined Formats

A user-defined format needs to be created for **`Country`**.

Current Report (partial output)

Obs	Sales ID	Job Title	Annual Salary	Country	Date of Birth	Date of Hire
61	120179	Sales Rep. III	$28,510	AU	MAR1974	JAN2004
62	120180	Sales Rep. II	$26,970	AU	JUN1954	DEC1978
63	120198	Sales Rep. III	$28,025	AU	JAN1988	DEC2006
64	120261	Chief Sales Officer	$243,190	US	FEB1969	AUG1987
65	121018	Sales Rep. II	$27,560	US	JAN1944	JAN1974
66	121019	Sales Rep. IV	$31,320	US	JUN1986	JUN2004

Desired Report (partial output)

Obs	Sales ID	Job Title	Annual Salary	Country	Date of Birth	Date of Hire
61	120179	Sales Rep. III	$28,510	Australia	MAR1974	JAN2004
62	120180	Sales Rep. II	$26,970	Australia	JUN1954	DEC1978
63	120198	Sales Rep. III	$28,025	Australia	JAN1988	DEC2006
64	120261	Chief Sales Officer	$243,190	United States	FEB1969	AUG1987
65	121018	Sales Rep. II	$27,560	United States	JAN1944	JAN1974
66	121019	Sales Rep. IV	$31,320	United States	JUN1986	JUN2004

User-Defined Formats

To create and use your own formats, do the following:

Part 1 Use the FORMAT procedure to create the user-defined format.

Part 2 Apply the format to a specific variable(s) by using a FORMAT statement in the reporting procedure.

The FORMAT Procedure

The *FORMAT procedure* is used to create user-defined formats.

General form of the FORMAT procedure with the VALUE statement:

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

The FORMAT Procedure

A *format-name*

- names the format that you are creating
- cannot be more than 32 characters in SAS®9
- for character values, must have a dollar sign ($) as the first character, and a letter or underscore as the second character
- for numeric values, must have a letter or underscore as the first character
- cannot end in a number
- cannot be the name of a SAS format
- does not end with a period in the VALUE statement.

11.05 Multiple Answer Poll

Which user-defined format names are invalid?

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

The FORMAT Procedure

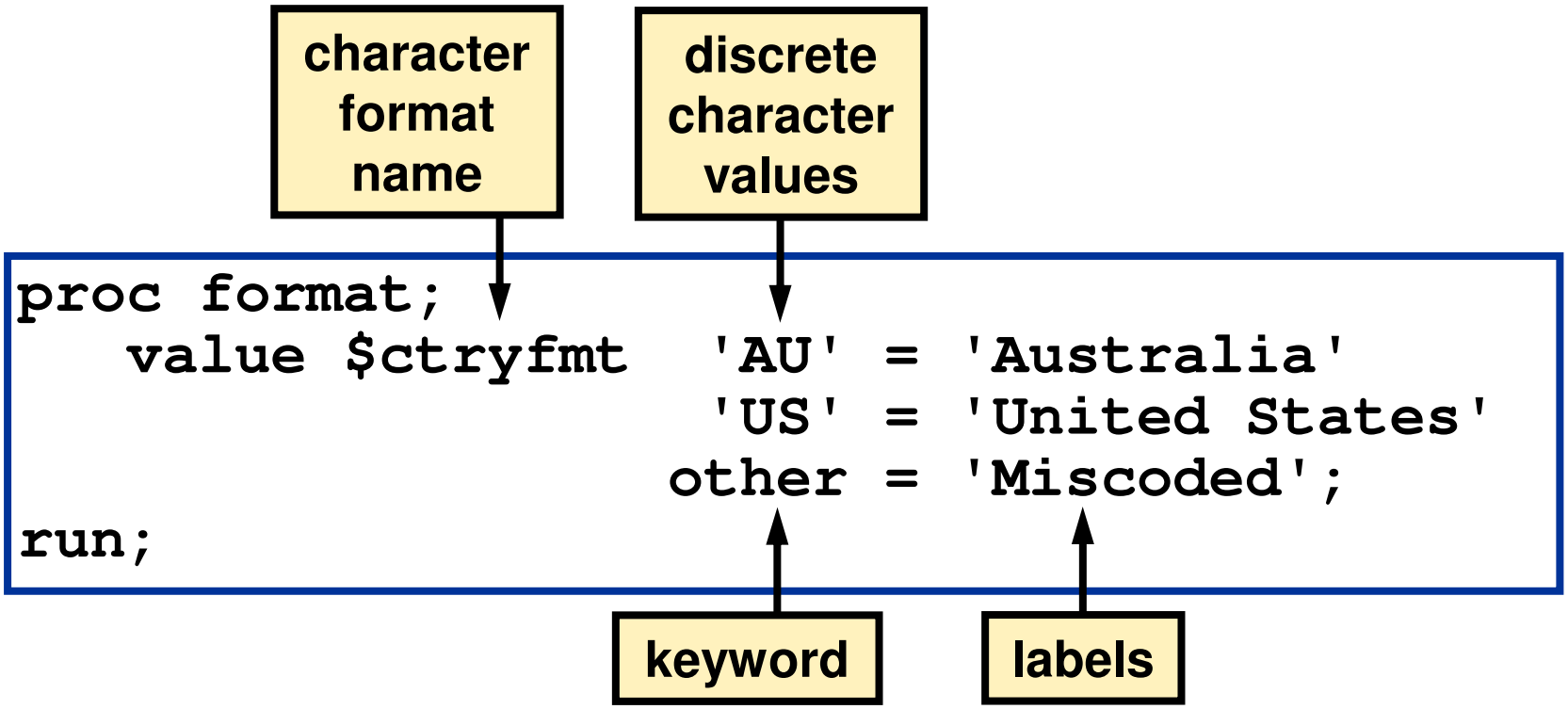
Range(s) can be

- single values
- ranges of values
- lists of values.

Labels

- can be up to 32,767 characters in length
- are typically enclosed in quotation marks, although it is not required.

Character User-Defined Format



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

Character User-Defined Format

Part 1

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

proc print data=orion.sales label;
   var Employee_ID Job_Title Salary
       Country Birth_Date Hire_Date;
   label Employee_ID='Sales ID'
         Job_Title='Job Title'
         Salary='Annual Salary'
         Birth_Date='Date of Birth'
         Hire_Date='Date of Hire';
   format Salary dollar10.0
          Birth_Date Hire_Date monyy7.
          Country $ctryfmt.;
run;
```

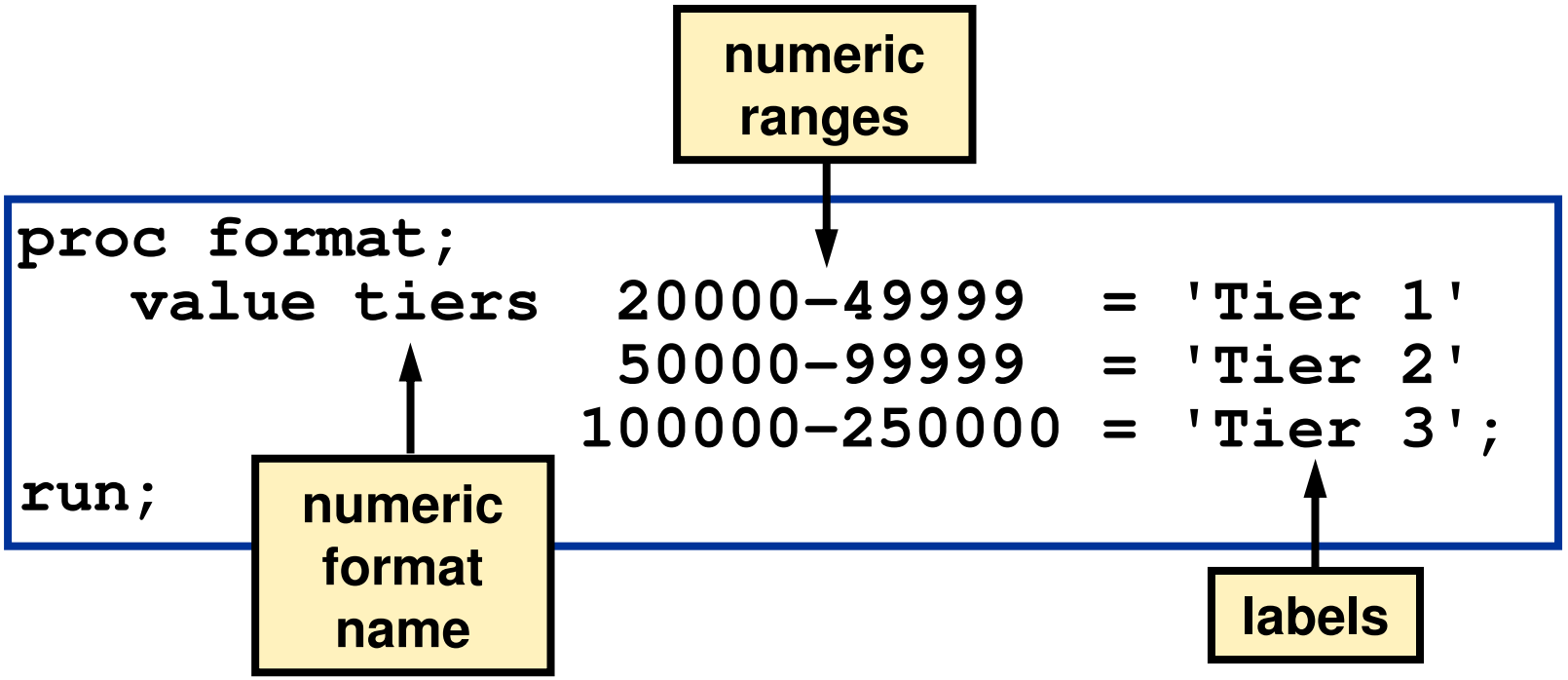
Part 2

Character User-Defined Format

Partial PROC PRINT Output

Obs	Sales ID	Job Title	Annual Salary	Country	Date of Birth	Date of Hire
60	120178	Sales Rep. II	$26,165	Australia	NOV1954	APR1974
61	120179	Sales Rep. III	$28,510	Australia	MAR1974	JAN2004
62	120180	Sales Rep. II	$26,970	Australia	JUN1954	DEC1978
63	120198	Sales Rep. III	$28,025	Australia	JAN1988	DEC2006
64	120261	Chief Sales Officer	$243,190	United States	FEB1969	AUG1987
65	121018	Sales Rep. II	$27,560	United States	JAN1944	JAN1974
66	121019	Sales Rep. IV	$31,320	United States	JUN1986	JUN2004
67	121020	Sales Rep. IV	$31,750	United States	FEB1984	MAY2002
68	121021	Sales Rep. IV	$32,985	United States	DEC1974	MAR1994
69	121022	Sales Rep. IV	$32,210	United States	OCT1979	FEB2002
70	121023	Sales Rep. I	$26,010	United States	MAR1964	MAY1989
71	121024	Sales Rep. II	$26,600	United States	SEP1984	MAY2004
72	121025	Sales Rep. II	$28,295	United States	OCT1949	SEP1975

Numeric User-Defined Format



p111d07

11.06 Quiz

If you have a value of 99999.87, how will it be displayed if the TIERS format is applied to the value?

a. Tier 2
b. Tier 3
c. a missing value
d. none of the above

```
proc format;
   value tiers  20000-49999   = 'Tier 1'
                50000-99999   = 'Tier 2'
               100000-250000  = 'Tier 3';
run;
```

Numeric User-Defined Formats

The less than (<) symbol excludes values from ranges.

- Put < after the value if you want to exclude the first value in a range.
- Put < before the value if you want to exclude the last value in a range.

50000 - 100000	Includes 50000	Includes 100000
50000 - < 100000	Includes 50000	Excludes 100000
50000 < - 100000	Excludes 50000	Includes 100000
50000 < - < 100000	Excludes 50000	Excludes 100000

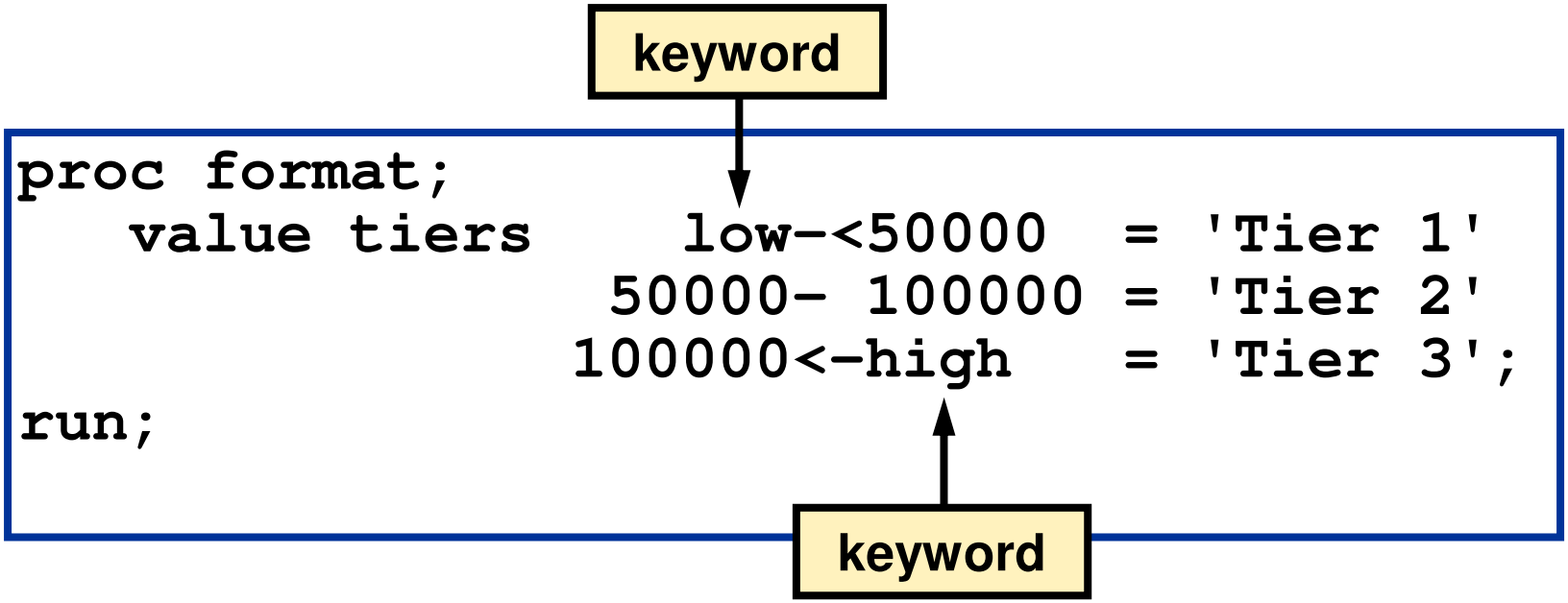
11.07 Quiz

If you have a value of 100000, how will it be displayed if the TIERS format is applied to the value?

a. Tier 2
b. Tier 3
c. 100000
d. a missing value

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

Numeric User-Defined Format



LOW encompasses the lowest possible value.

HIGH encompasses the highest possible value.

Numeric User-Defined Format

Part 1

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

proc print data=orion.sales label;
   var Employee_ID Job_Title Salary
       Country Birth_Date Hire_Date;
   label Employee_ID='Sales ID'
         Job_Title='Job Title'
         Salary='Annual Salary'
         Birth_Date='Date of Birth'
         Hire_Date='Date of Hire';
```

Part 2

```
   format Birth_Date Hire_Date monyy7.
          Salary tiers.;
run;
```

Numeric User-Defined Format

Partial PROC PRINT Output

Obs	Sales ID	Job Title	Annual Salary	Country	Date of Birth	Date of Hire
60	120178	Sales Rep. II	Tier 1	AU	NOV1954	APR1974
61	120179	Sales Rep. III	Tier 1	AU	MAR1974	JAN2004
62	120180	Sales Rep. II	Tier 1	AU	JUN1954	DEC1978
63	120198	Sales Rep. III	Tier 1	AU	JAN1988	DEC2006
64	120261	Chief Sales Officer	Tier 3	US	FEB1969	AUG1987
65	121018	Sales Rep. II	Tier 1	US	JAN1944	JAN1974
66	121019	Sales Rep. IV	Tier 1	US	JUN1986	JUN2004
67	121020	Sales Rep. IV	Tier 1	US	FEB1984	MAY2002
68	121021	Sales Rep. IV	Tier 1	US	DEC1974	MAR1994
69	121022	Sales Rep. IV	Tier 1	US	OCT1979	FEB2002
70	121023	Sales Rep. I	Tier 1	US	MAR1964	MAY1989
71	121024	Sales Rep. II	Tier 1	US	SEP1984	MAY2004
72	121025	Sales Rep. II	Tier 1	US	OCT1949	SEP1975

Other User-Defined Format Examples

```
proc format;
   value $grade      'A' = 'Good'
                 'B'-'D' = 'Fair'
                     'F' = 'Poor'
                 'I','U' = 'See Instructor'
                   other = 'Miscoded';
run;
```

```
proc format;
   value mnthfmt     1,2,3 = 'Qtr 1'
                     4,5,6 = 'Qtr 2'
                     7,8,9 = 'Qtr 3'
                  10,11,12 = 'Qtr 4'
                         . = 'missing'
                     other = 'unknown';
run;
```

Multiple User-Defined Formats

Multiple VALUE statements can be 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;
```

Multiple User-Defined Formats

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

Partial PROC PRINT Output

Obs	Sales ID	Job Title	Annual Salary	Country	Date of Birth	Date of Hire
60	120178	Sales Rep. II	Tier 1	Australia	NOV1954	APR1974
61	120179	Sales Rep. III	Tier 1	Australia	MAR1974	JAN2004
62	120180	Sales Rep. II	Tier 1	Australia	JUN1954	DEC1978
63	120198	Sales Rep. III	Tier 1	Australia	JAN1988	DEC2006
64	120261	Chief Sales Officer	Tier 3	United States	FEB1969	AUG1987
65	121018	Sales Rep. II	Tier 1	United States	JAN1944	JAN1974
66	121019	Sales Rep. IV	Tier 1	United States	JUN1986	JUN2004
67	121020	Sales Rep. IV	Tier 1	United States	FEB1984	MAY2002

p111d07

Multiple User-Defined Formats

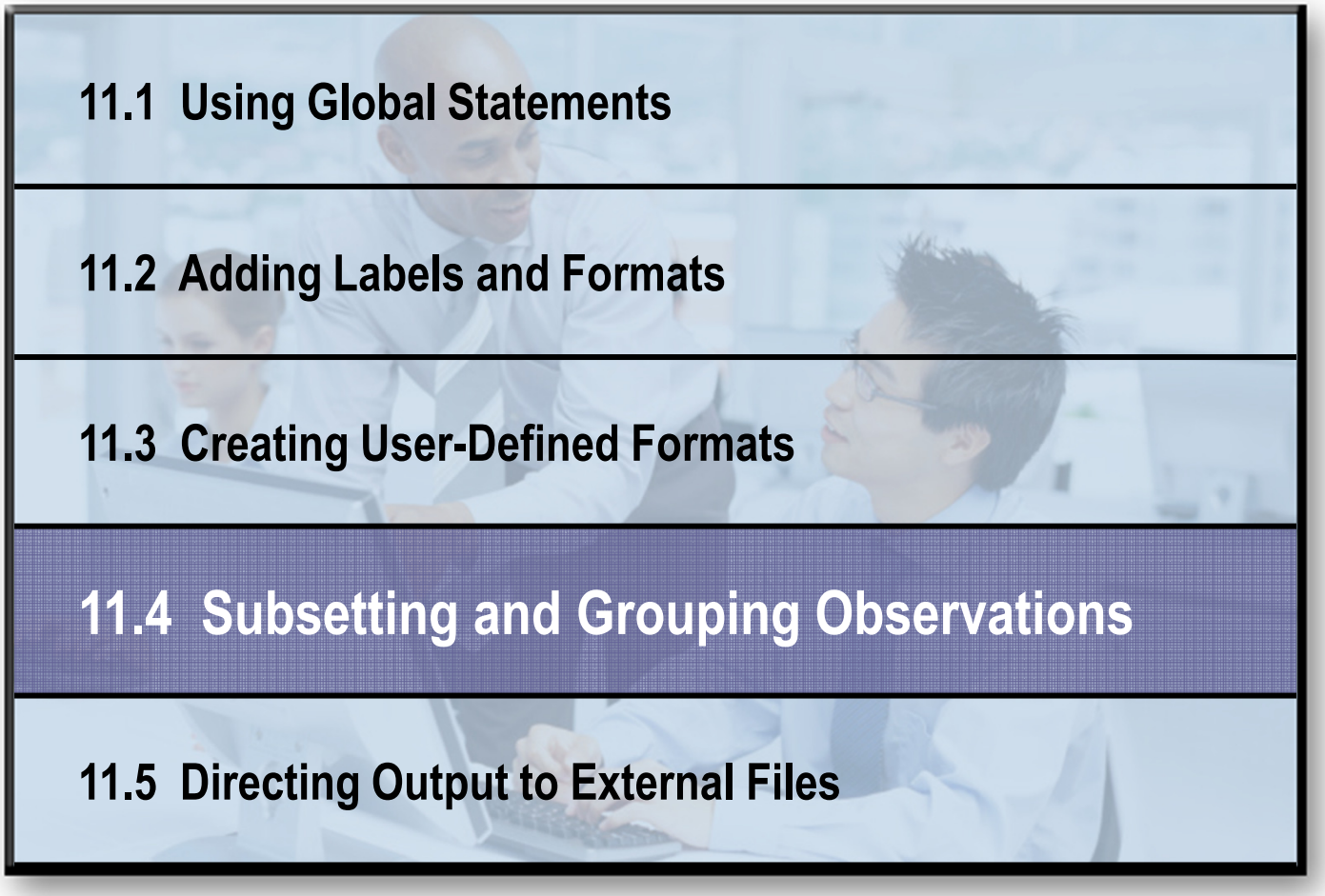
```
proc freq data=orion.sales;
   tables Country Salary;
   format Country $ctryfmt. Salary tiers.;
run;
```

The FREQ Procedure

Country	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Australia	63	38.18	63	38.18
United States	102	61.82	165	100.00

Salary	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Tier 1	159	96.36	159	96.36
Tier 2	4	2.42	163	98.79
Tier 3	2	1.21	165	100.00

Chapter 11: Enhancing Reports



Objectives

- Display selected observations in reports by using the WHERE statement.
- Display groups of observations in reports by using the BY statement.

The WHERE Statement (Review)

For subsetting observations in a report, the WHERE statement is used to select observations that meet a certain condition.

General form of the WHERE statement:

WHERE *where-expression*;

The *where-expression* is a sequence of operands and operators that form a set of instructions that define a condition for selecting observations.

- Operands include constants and variables.
- Operators are symbols that request a comparison, arithmetic calculation, or logical operation.

11.08 Quiz

Which of the following WHERE statements have invalid syntax?

a.
```
where Salary ne .;
```

b.
```
where Hire_Date >= '01APR2008'd;
```

c.
```
where Country in (AU US);
```

d.
```
where Salary + Bonus <= 10000;
```

e.
```
where Gender ne 'M' Salary >= 50000;
```

f.
```
where Name like '%N';
```

Subsetting Observations

```
proc print data=orion.sales;
    var First_Name Last_Name
        Job_Title Country Salary;
    where Salary > 75000;
run;
```

Obs	First_ Name	Last_Name	Job_Title	Country	Salary
1	Tom	Zhou	Sales Manager	AU	108255
2	Wilson	Dawes	Sales Manager	AU	87975
64	Harry	Highpoint	Chief Sales Officer	US	243190
163	Louis	Favaron	Senior Sales Manager	US	95090
164	Renee	Capachietti	Sales Manager	US	83505
165	Dennis	Lansberry	Sales Manager	US	84260

Subsetting Observations

```
proc means data=orion.sales;
    var Salary;
    where Country = 'AU';
run;
```

The MEANS Procedure

Analysis Variable : Salary

N	Mean	Std Dev	Minimum	Maximum
63	30158.97	12699.14	25185.00	108255.00

Setup for the Poll

- Retrieve and submit program **p111a02**.
- View the log to determine how SAS handles multiple WHERE statements.

```
proc freq data=orion.sales;
   tables Gender;
   where Salary > 75000;
   where Country = 'US';
run;
```

11.09 Multiple Choice Poll

Which statement is true concerning the multiple WHERE statements?

a. All the WHERE statements are used.
b. None of the WHERE statements is used.
c. The first WHERE statement is used.
d. The last WHERE statement is used.

The BY Statement

For grouping observations in a report, the BY statement is used to produce separate sections of the report for each BY group.

General form of the BY statement:

BY <DESCENDING> *by-variable(s)*;

The observations in the data set must be sorted by the variables specified in the BY statement.

Grouping Observations

```
proc sort data=orion.sales out=work.sort;
   by Country descending Gender Last_Name;
run;

proc print data=work.sort;
   by Country descending Gender;
run;
```

Grouping Observations

Partial PROC PRINT Output

---------- Country=AU Gender=M ----------

Obs	Employee_ID	First_ Name	Last_Name	Salary
1	120145	Sandy	Aisbitt	26060
2	120144	Viney	Barbis	30265
3	120146	Wendall	Cederlund	25985

---------- Country=AU Gender=F ----------

Obs	Employee_ID	First_Name	Last_Name	Salary
37	120168	Selina	Barcoe	25275
38	120198	Meera	Body	28025
39	120149	Judy	Chantharasy	26390
40	120127	Sharryn	Clarkson	28100
41	120138	Shani	Duckett	25795
42	120121	Irenie	Elvish	26600
43	120154	Caterina	Hayawardhana	30490
44	120123	Kimiko	Hotstone	26190

11.10 Quiz

Which is a valid BY statement for the PROC FREQ step?

a. `by Country Gender;`

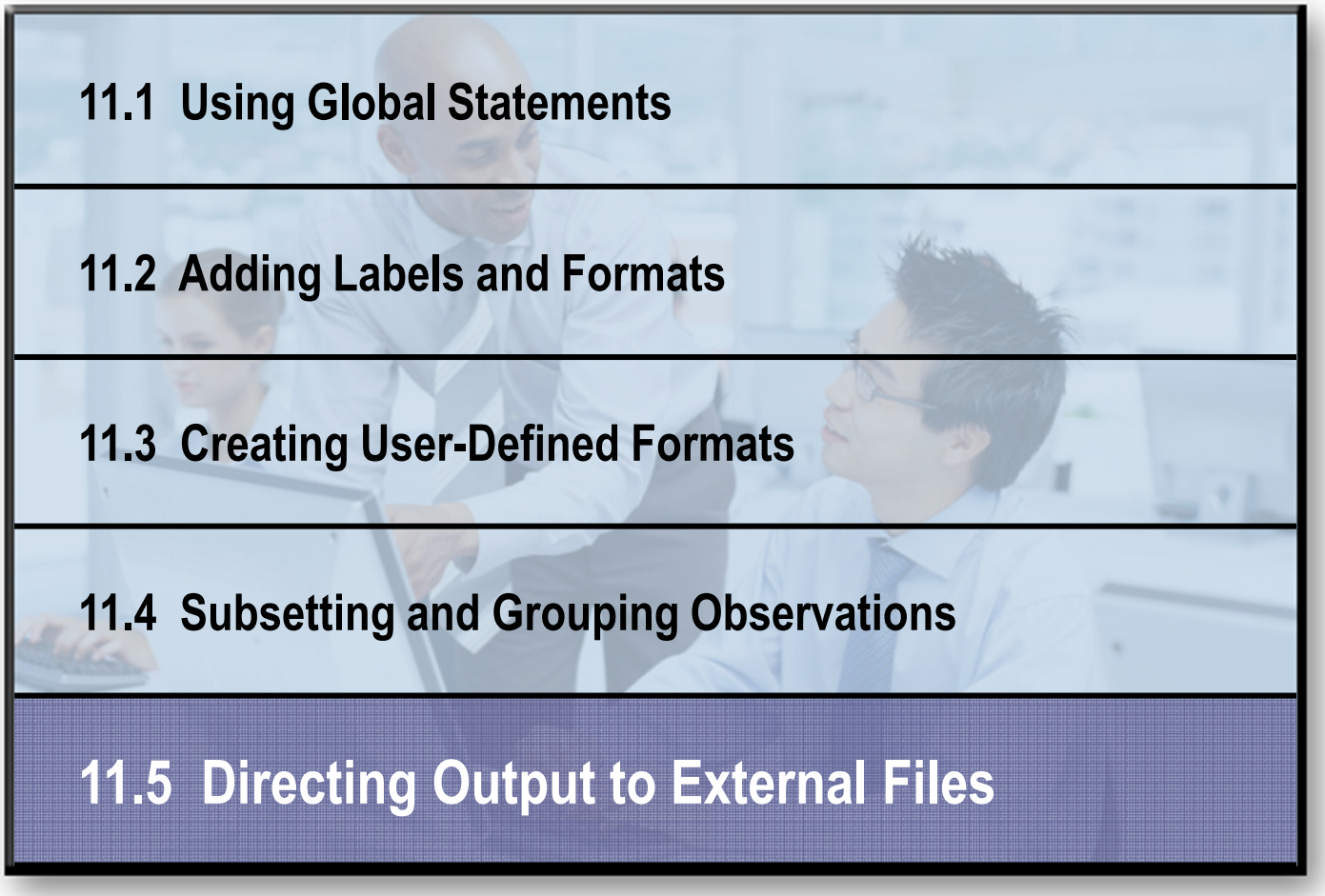
b. `by Gender Last_Name;`

c. `by Country;`

d. `by Gender;`

```
proc sort data=orion.sales out=work.sort;
   by Country descending Gender Last_Name;
run;

proc freq data=work.sort;
   tables Gender;
run;
```

Chapter 11: Enhancing Reports

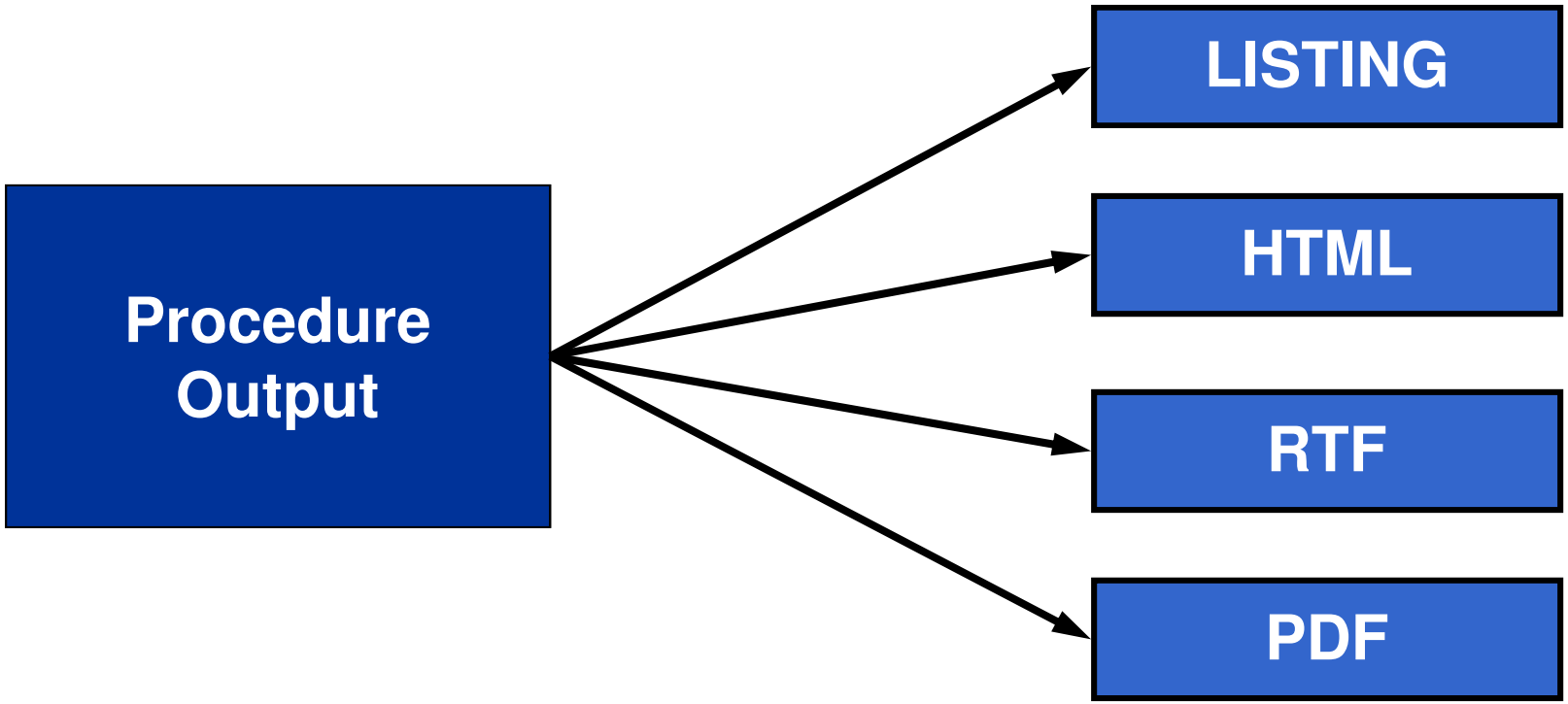


Objectives

- Direct output to ODS destinations by using ODS statements.
- Specify a style definition by using the STYLE= option.
- Create ODS files that can be opened in Microsoft Excel.

Output Delivery System

Output can be sent to a variety of destinations by using ODS statements.



Output Delivery System

Destination	Type of File	Viewed In
LISTING		SAS Output Window or SAS/GRAPH Window
HTML	Hypertext Markup Language	Web Browsers such as Internet Explorer
PDF	Portable Document Format	Adobe Products such as Acrobat Reader
RTF	Rich Text Format	Word Processors such as Microsoft Word

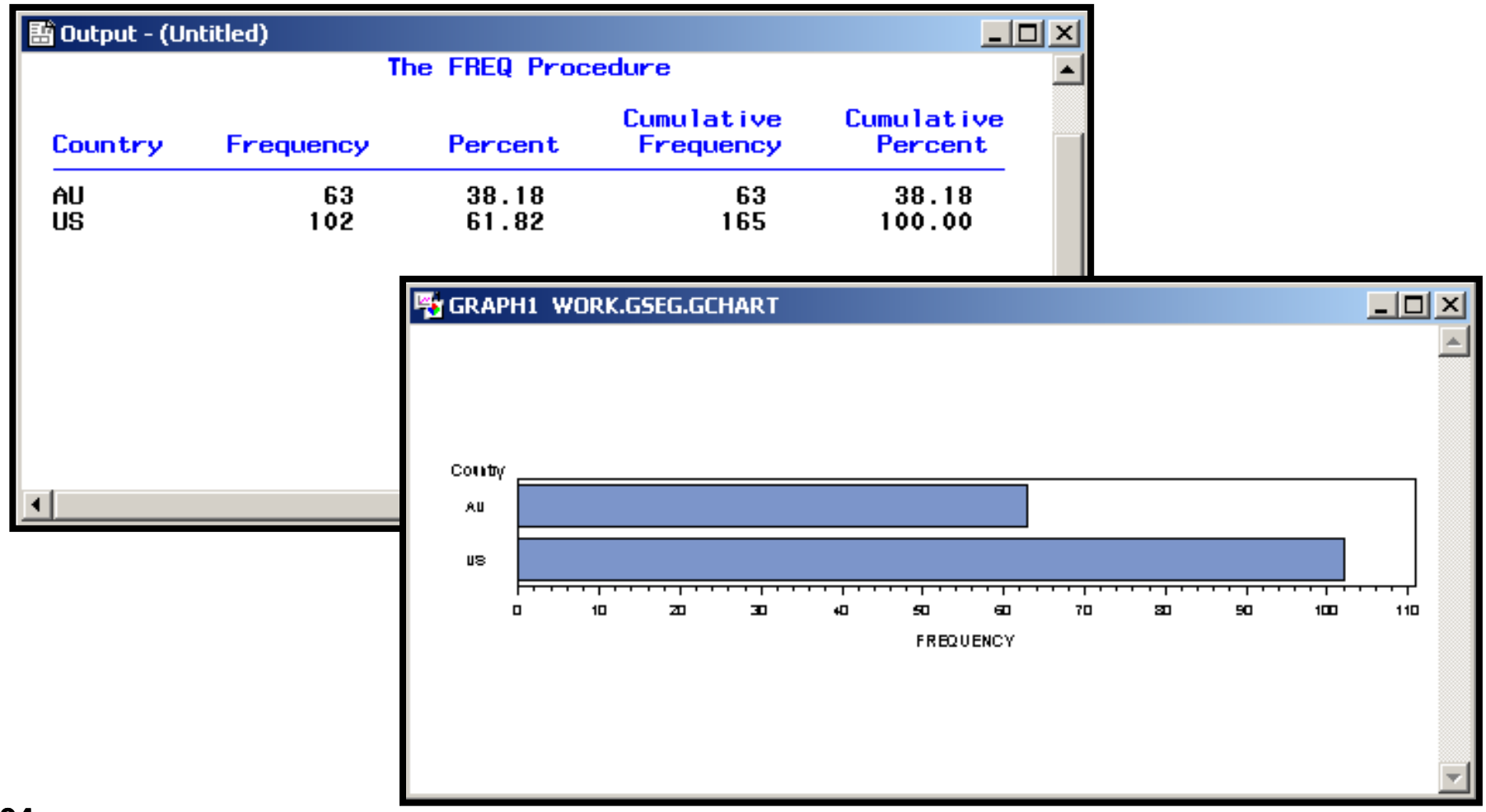
Default ODS Destination

The LISTING destination is the default ODS destination.

```
ods listing;

proc freq data=orion.sales;
   tables Country;
run;

proc gchart data=orion.sales;
   hbar Country / nostats;
run;
```

Default ODS Destination

The LISTING destination directs output to the OUTPUT window and the GRAPH window.



Default ODS Destination

The ODS LISTING CLOSE statement stops sending output to the OUTPUT and GRAPH windows.

```
ods listing close;

proc freq data=orion.sales;
   tables Country;
run;

proc gchart data=orion.sales;
   hbar Country / nostats;
run;
```

Default ODS Destination

A warning will appear in the SAS log if the LISTING destination is closed and no other destinations are active.

Partial SAS Log

```
23   ods listing close;
24
25   proc freq data=orion.sales;
26      tables Country;
27   run;

WARNING: No output destinations active.
NOTE: There were 165 observations read from the data set ORION.SALES.
```

HTML, PDF, and RTF Destinations

ODS destinations such as HTML, PDF, and RTF are opened and closed in the following manner:

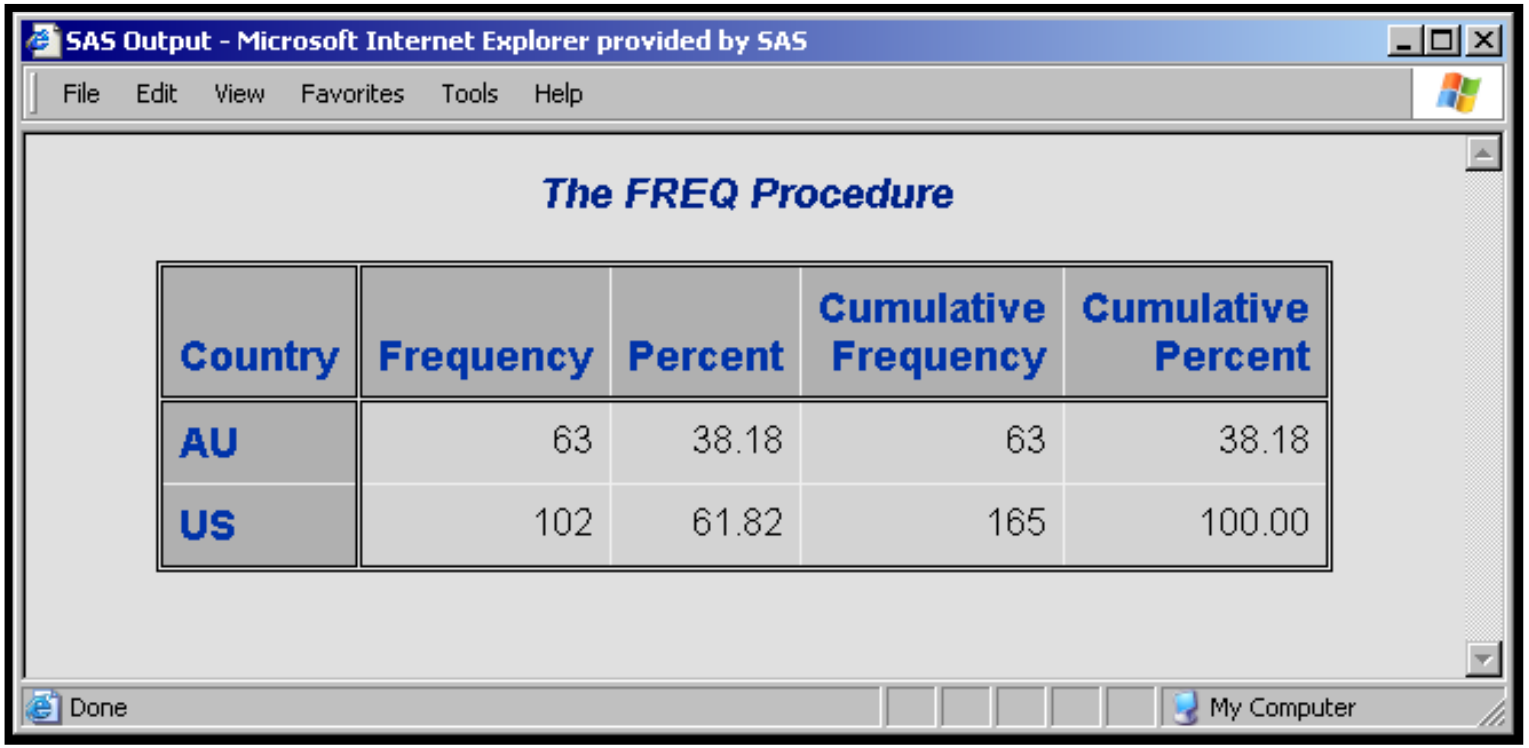
ODS *destination* FILE = ' *filename.ext* **'** <*options*>**;**

SAS code to generate a report(s)

ODS *destination* CLOSE;

HTML Destination

```
ods html file='myreport.html';
proc freq data=orion.sales;
   tables Country;
run;
ods html close;
```

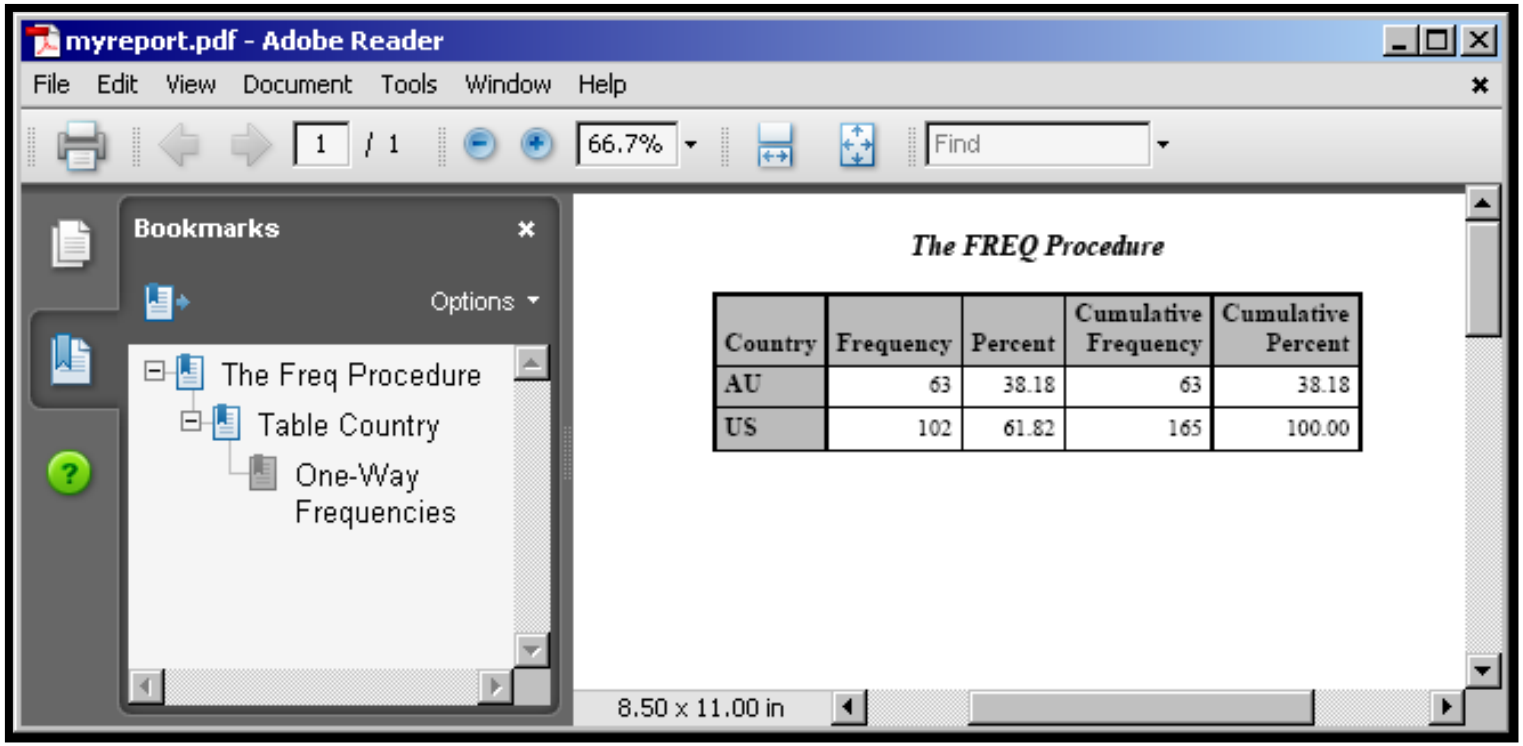


Country	Frequency	Percent	Cumulative Frequency	Cumulative Percent
AU	63	38.18	63	38.18
US	102	61.82	165	100.00

p111d11

PDF Destination

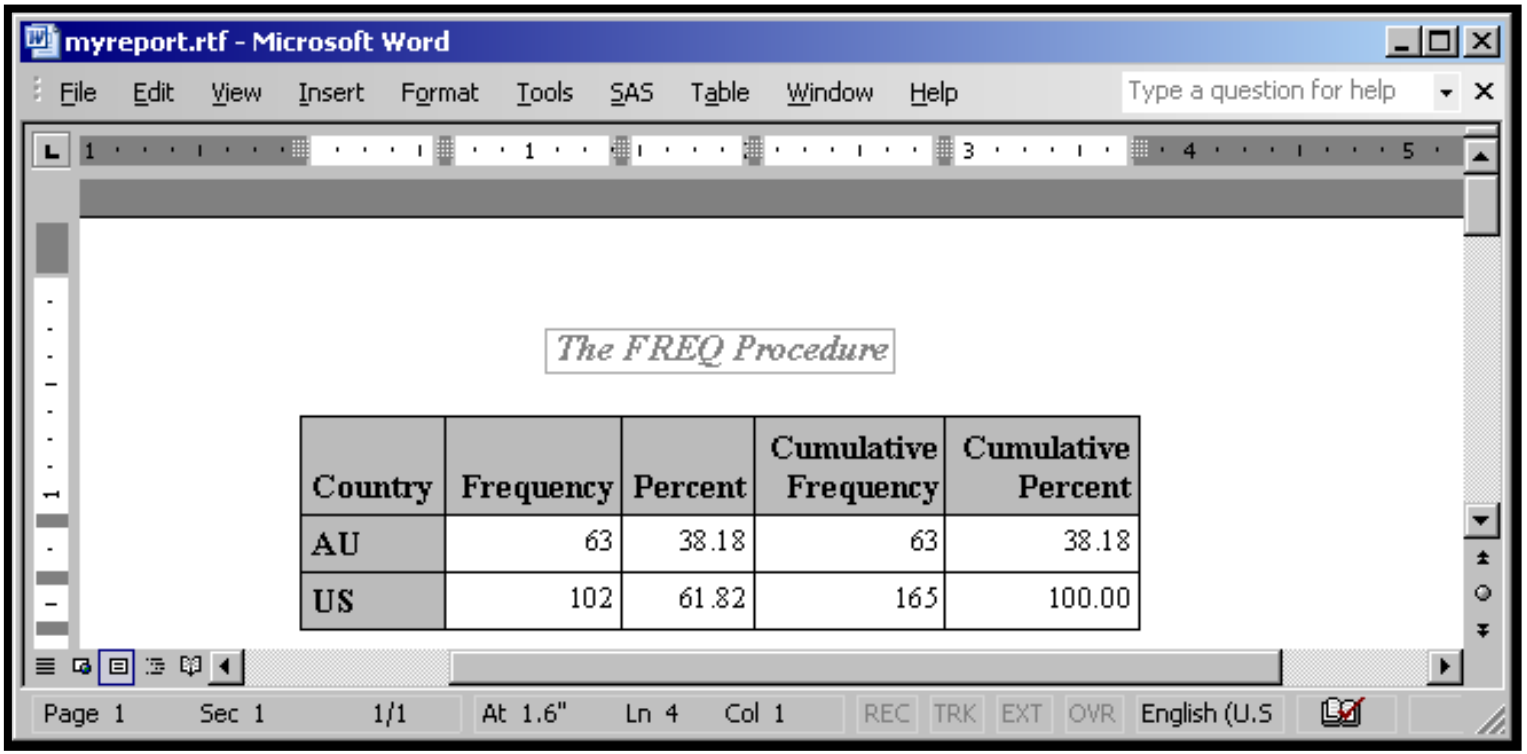
```
ods pdf file='myreport.pdf';
proc freq data=orion.sales;
   tables Country;
run;
ods pdf close;
```



p111d11

RTF Destination

```
ods rtf file='myreport.rtf';
proc freq data=orion.sales;
   tables Country;
run;
ods rtf close;
```



Country	Frequency	Percent	Cumulative Frequency	Cumulative Percent
AU	63	38.18	63	38.18
US	102	61.82	165	100.00

p111d11

11.11 Quiz

What is the problem with this program?

```
ods pdf file='myreport.pdf';

proc print data=orion.sales;
run;

ods close;
```

Single Destination

Output can be sent to only one destination.

```
ods listing close;

ods html file='example.html';

proc freq data=orion.sales;
   tables Country;
run;

ods html close;

ods listing;
```

It is a good habit to open the LISTING destination at the end of a program to guarantee an open destination for the next submission.

Multiple Destinations

Output can be sent to many destinations.

```
ods listing;
ods pdf file='example.pdf';
ods rtf file='example.rtf';

proc freq data=orion.sales;
   tables Country;
run;

ods pdf close;
ods rtf close;
```

To view the results, all destinations except the LISTING destination must be closed.

Multiple Destinations

Use _ALL_ in the ODS CLOSE statement to close all open destinations including the LISTING destination.

```
ods listing;
ods pdf file='example.pdf';
ods rtf file='example.rtf';

proc freq data=orion.sales;
   tables Country;
run;

ods _all_ close;
ods listing;
```

Multiple Procedures

Output from many procedures can be sent to ODS destinations.

```
ods listing;
ods pdf file='example.pdf';
ods rtf file='example.rtf';

proc freq data=orion.sales;
    tables Country;
run;

proc means data=orion.sales;
    var Salary;
run;

ods _all_ close;
ods listing;
```

File Location

A path can be specified to control the location of where the file is stored.

```
ods html file='s:\workshop\example.html';

proc freq data=orion.sales;
   tables Country;
run;

proc means data=orion.sales;
   var Salary;
run;

ods html close;
```

If no path is specified, the file is saved in the current default directory.

p111d11

Operating Environments

The Output Delivery System works on all operating environments.

z/OS (OS/390) Example:

```
ods html file='.workshop.report(example)'
          rs=none;

proc freq data=orion.sales;
   tables Country;
run;

ods html close;
```

Use the RS=NONE option when you create HTML and RTF files on z/OS (OS/390).

STYLE= Option

Use a STYLE= option in the ODS destination statement to specify a style definition.

```
ODS destination FILE = 'filename.ext'
                       STYLE = style-definition;
```

- A *style definition* describes how to display the presentation aspects such as colors and fonts of SAS output.
- STYLE= cannot be used with the LISTING destination.

SAS Supplied Style Definitions

Analysis	Astronomy	Banker	BarrettsBlue
Beige	blockPrint	Brick	Brown
Curve	D3d	Default	Education
EGDefault	Electronics	fancyPrinter	Festival
FestivalPrinter	Gears	Journal	Magnify
Meadow	MeadowPrinter	Minimal	Money
NoFontDefault	Normal	NormalPrinter	Printer
Rsvp	Rtf	sansPrinter	sasdocPrinter
Sasweb	Science	Seaside	SeasidePrinter
serifPrinter	Sketch	Statdoc	Statistical
Theme	Torn	Watercolor	

SAS Supplied Style Definitions

The following style definitions are new to SAS 9.2:

grayscalePrinter	Harvest	HighContrast
Journal2	Journal3	Listing
monochromePrinter	Ocean	Solutions

HTML Examples

STYLE=DEFAULT

The FREQ Procedure

Country	Frequency	Percent	Cumulative Frequency	Cumulative Percent
AU	63	38.18	63	38.18
US	102	61.82	165	100.00

STYLE=SASWEB

The FREQ Procedure

Country	Frequency	Percent	Cumulative Frequency	Cumulative Percent
AU	63	38.18	63	38.18
US	102	61.82	165	100.00

PDF Examples

STYLE=PRINTER

The FREQ Procedure

Country	Frequency	Percent	Cumulative Frequency	Cumulative Percent
AU	63	38.18	63	38.18
US	102	61.82	165	100.00

STYLE=JOURNAL

The FREQ Procedure

Country	*Frequency*	*Percent*	*Cumulative Frequency*	*Cumulative Percent*
AU	63	38.18	63	38.18
US	102	61.82	165	100.00

p111d13

RTF Examples

STYLE=RTF

The FREQ Procedure

Country	Frequency	Percent	Cumulative Frequency	Cumulative Percent
AU	63	38.18	63	38.18
US	102	61.82	165	100.00

STYLE=OCEAN

The FREQ Procedure

Country	Frequency	Percent	Cumulative Frequency	Cumulative Percent
AU	63	38.18	63	38.18
US	102	61.82	165	100.00

p111d13

Setup for the Poll

- Retrieve **p111a03**.
- Add a STYLE= option to the first ODS statement, and select one of the following style definitions:

HighContrast	Minimal	Listing	Journal3

- Submit the program and review the results.
- Modify the STYLE= option to use one of the following style definitions:

Education	Harvest	Rsvp	Solutions

- Submit the program and review the results.

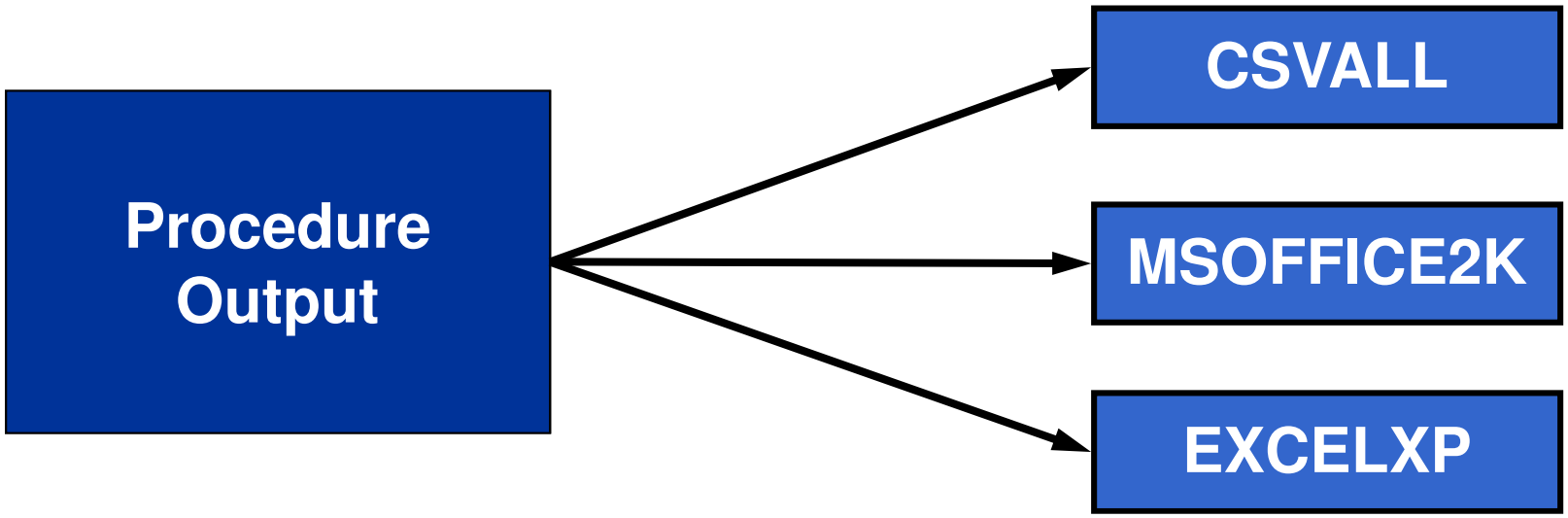
11.12 Poll

Did you notice a difference in the presentation aspects between the two style definitions?

- Yes
- No

Destinations Used with Excel

The following destinations create files that can be opened in Excel.



Destinations Used with Excel

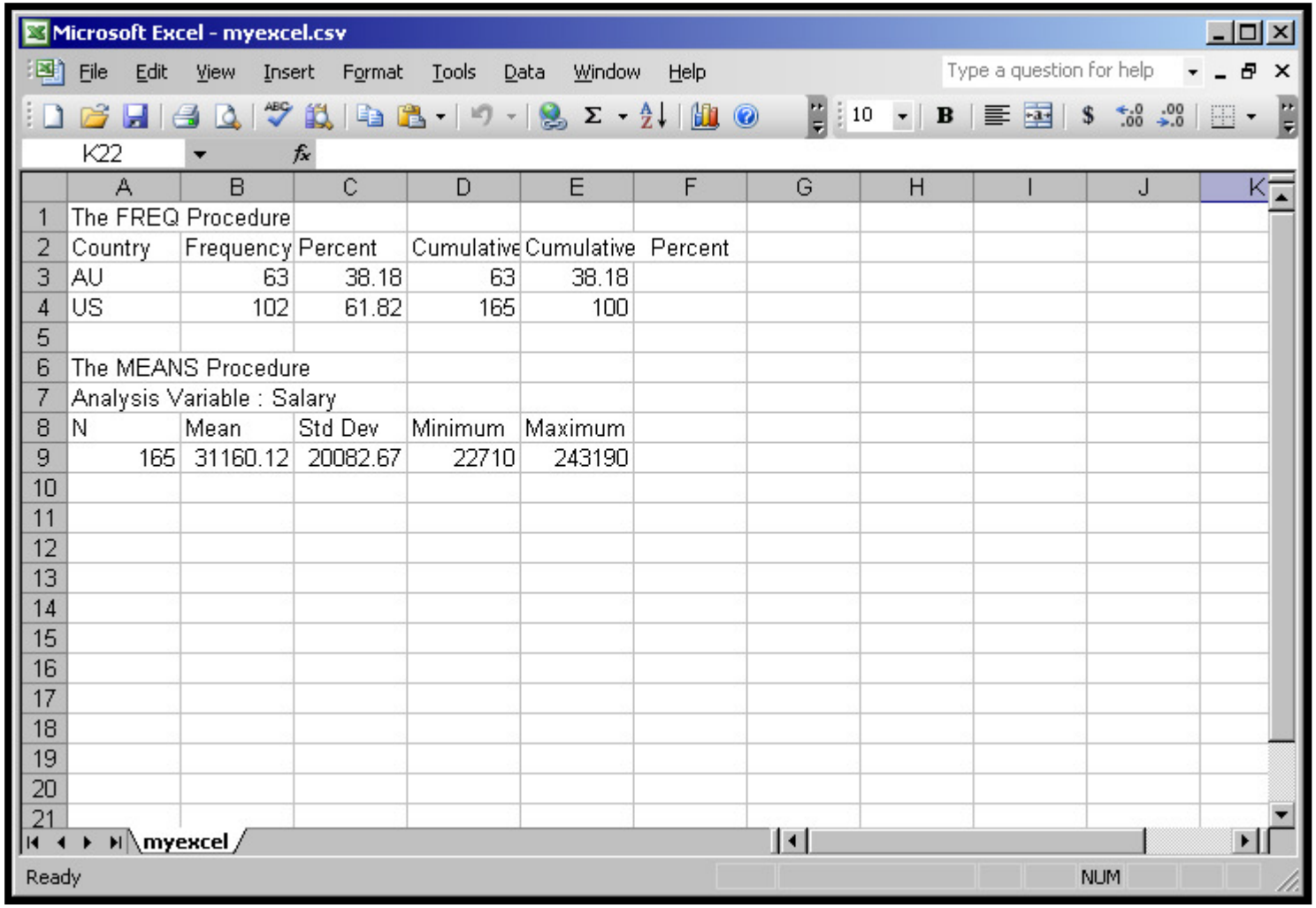
Destination	Type of File	Viewed In
CSVALL	Comma-Separated Value	Editor or Microsoft Excel
MSOFFICE2K	Hypertext Markup Language	Web Browser or Microsoft Word or Microsoft Excel
EXCELXP	Extensible Markup Language	Microsoft Excel

CSVALL Destination

```
ods csvall file='myexcel.csv';

proc freq data=orion.sales;
   tables Country;
run;

proc means data=orion.sales;
   var Salary;
run;

ods csvall close;
```

CSVALL Destination

CSVALL does not include any style information.



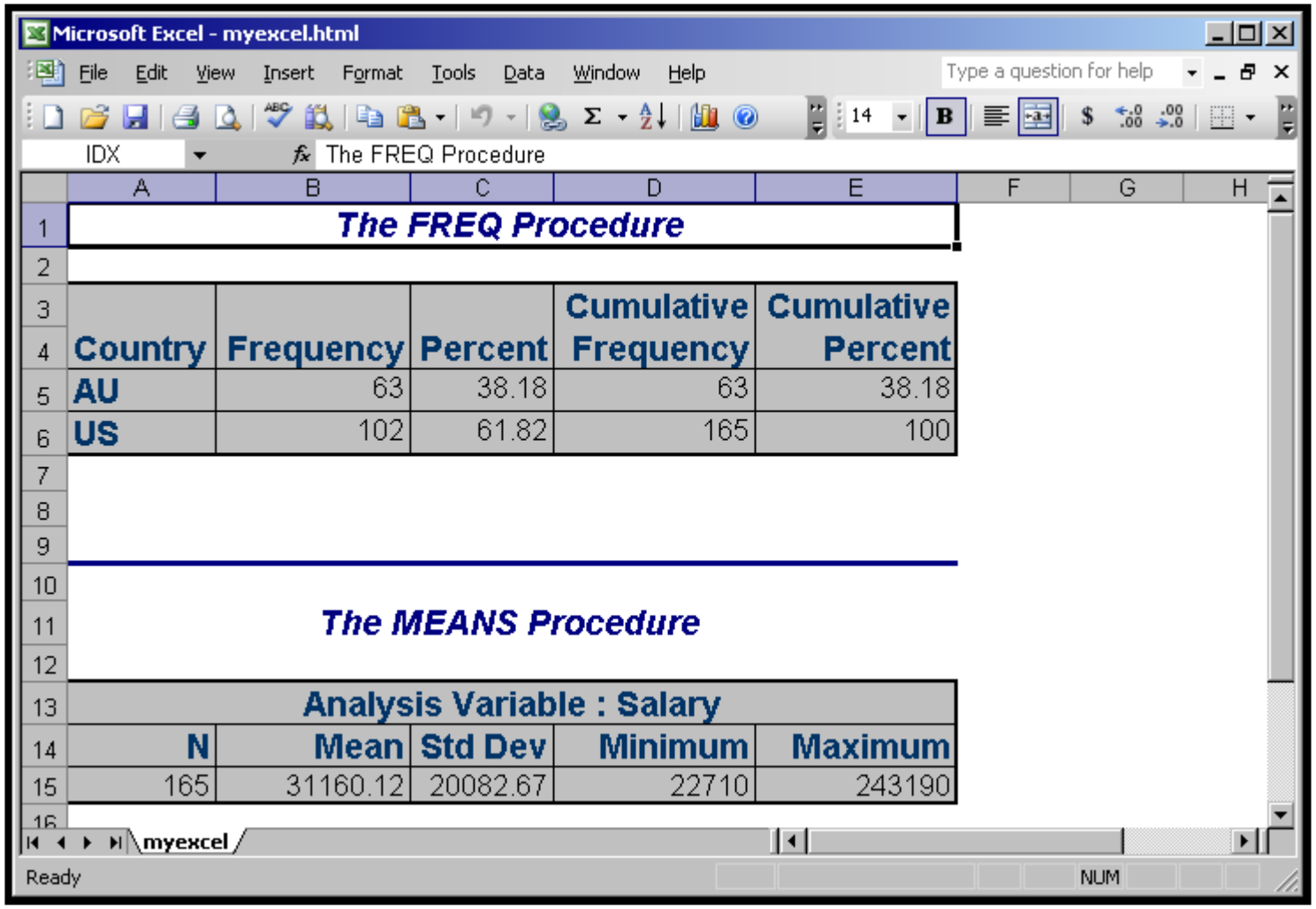
	A	B	C	D	E	F	G	H	I	J	K
1	The FREQ Procedure										
2	Country	Frequency	Percent	Cumulative	Cumulative	Percent					
3	AU	63	38.18	63	38.18						
4	US	102	61.82	165	100						
5											
6	The MEANS Procedure										
7	Analysis Variable : Salary										
8	N	Mean	Std Dev	Minimum	Maximum						
9	165	31160.12	20082.67	22710	243190						
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											
21											

MSOFFICE2K Destination

```
ods msoffice2k file='myexcel.html';

proc freq data=orion.sales;
   tables Country;
run;

proc means data=orion.sales;
   var Salary;
run;

ods msoffice2k close;
```

MSOFFICE2K Destination

MSOFFICE2K keeps the style information including spanning headers.

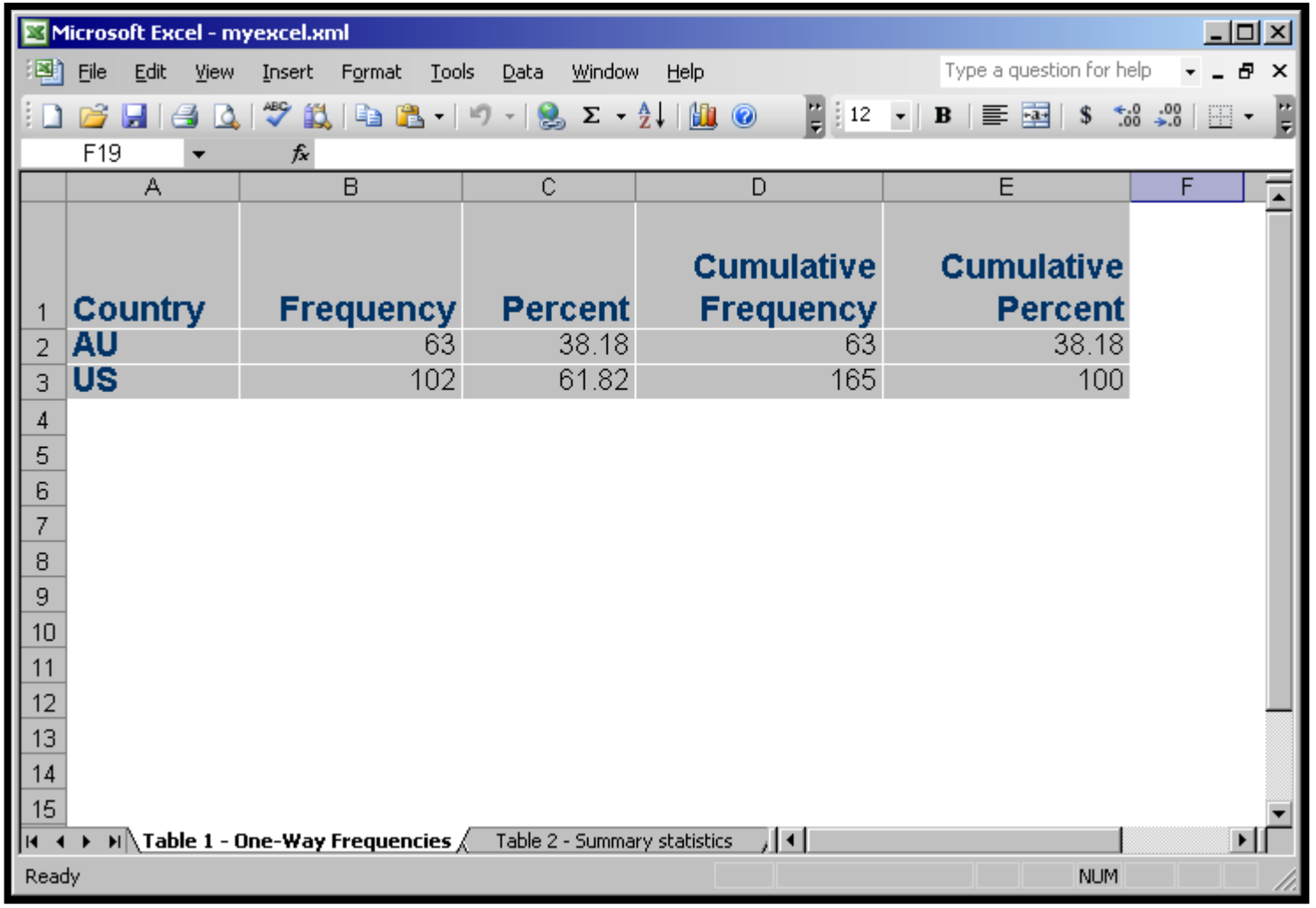


EXCELXP Destination

```
ods tagsets.excelxp file='myexcel.xml';

proc freq data=orion.sales;
   tables Country;
run;

proc means data=orion.sales;
   var Salary;
run;

ods tagsets.excelxp close;
```

EXCELXP Destination

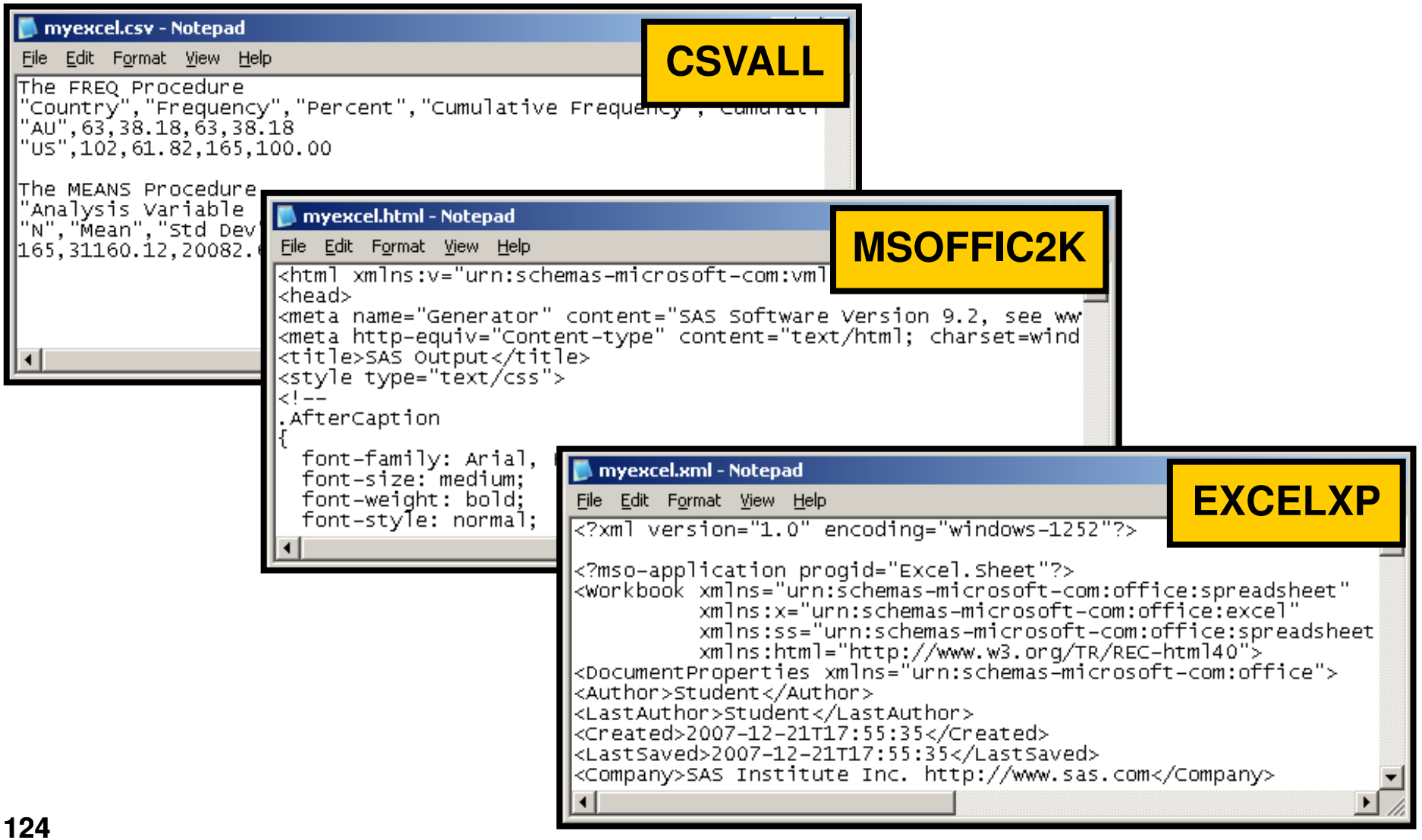
EXCELXP keeps the style information and each procedure is a separate sheet.



	A	B	C	D	E	F
1	Country	Frequency	Percent	Cumulative Frequency	Cumulative Percent	
2	AU	63	38.18	63	38.18	
3	US	102	61.82	165	100	
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						

Keep in Mind

The file you are creating is not an Excel file.



Chapter Review

1. What are some examples of global statements that enhance reports?

2. What is the maximum number of title or footnote lines?

3. How can you force a line break in a column header in PROC PRINT?

4. What is the difference between using a FORMAT statement in a PROC step versus a DATA step?

continued...

Chapter Review

5. How can you create a descriptive label for values of a variable such as a department name instead of a department code?

6. What are some examples of ODS destinations?