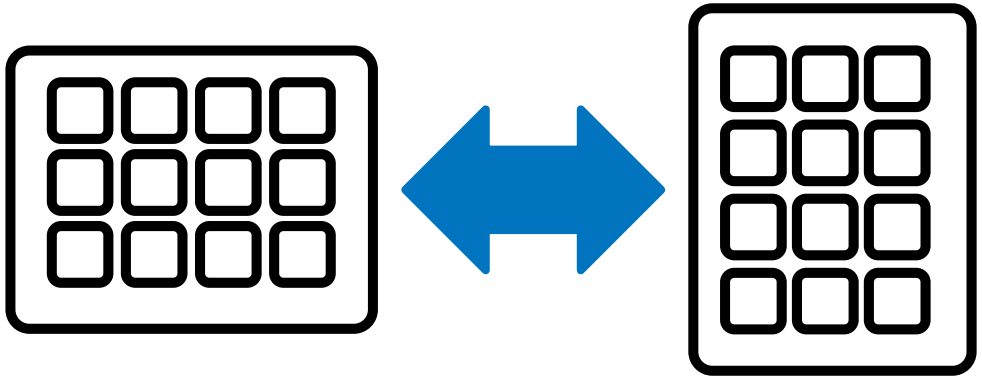
STAT604 SAS Lesson 15

Restructuring Tables

Restructuring Data with the DATA Step

Restructuring Tables



wide/flat table

Table Structure

Narrow table

class_test_wide

	Name	Math	Reading
1	Alfred	82	79
2	Alice	71	67
3	Barbara	96	86

class_test_narrow

	Name	TestSubject	TestScore
1	Alfred	Math	82
2	Alfred	Reading	79
3	Alice	Math	71
4	Alice	Reading	67
5	Barbara	Math	96
6	Barbara	Reading	86

Both tables include the same information, but they are structured differently.

Multiple Choice Question

Which table and column (or columns) could you use with PROC MEANS to calculate an average for all test scores combined?

a. **class_test_wide**, **Math** and **Reading**

b. **class_test_narrow**, **TestScore**

```
proc means data=???;
    var ???;
run;
```

	Name	Math	Reading
1	Alfred	82	79
2	Alice	71	67
3	Barbara	96	86

class_test_wide

	Name	TestSubject	TestScore
1	Alfred	Math	82
2	Alfred	Reading	79
3	Alice	Math	71
4	Alice	Reading	67
5	Barbara	Math	96
6	Barb		86

class_test_narrow

Multiple Choice Question – Correct Answer

Which table and column (or columns) could you use with PROC MEANS to calculate an average for all test scores combined?

a. class_test_wide, Math and Reading

b. class_test_narrow, TestScore

```
proc means data=pg2.class_test_narrow
           maxdec=1;
    var TestScore;
run;
```

The MEANS Procedure

Analysis Variable : TestScore				
N	Mean	Std Dev	Minimum	Maximum
38	77.7	11.3	55.0	99.0

Restructuring Data with the DATA Step

	Name	Math	Reading
1	Alfred	82	79
2	Alice	71	67
3	Barbara	96	86

wide

	Name	TestSubject	TestScore
1	Alfred	Math	82
2	Alfred	Reading	79
3	Alice	Math	71
4	Alice	Reading	67
5	Barbara	Math	96
6	Barbara	Reading	86

narrow

You can use the DATA step to read one row and write multiple rows.

Creating a Narrow Table with the DATA Step

```
data class_test_narrow(keep=Name Subject Score);
       set pg2.class_test_wide;
       length Subject $ 7;
       Subject="Math";
       Score=Math;
       output;
       Subject="Reading";
       Score=Reading;
       output;
run;
```

How could this be more efficient or programmer friendly?

Restructuring Data with the DATA Step

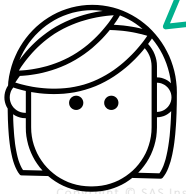
	Name	TestSubject	TestScore
1	Alfred	Math	82
2	Alfred	Reading	79
3	Alice	Math	71
4	Alice	Reading	67
5	Barbara	Math	96
6	Barbara	Reading	86

narrow

	Name	Math	Reading
1	Alfred	82	79
2	Alice	71	67
3	Barbara	96	86

wide

You can use the DATA step to read multiple rows before writing one row to the output table.



Restructuring Data with the DATA Step

	Name	TestSubject	TestScore
1	Alfred	Math	82
2	Alfred	Reading	79
3	Alice	Math	71
4	Alice	Reading	67
5	Barbara	Math	96
6	Barbara	Reading	86

narrow

	Name	Math	Reading
1	Alfred	82	79
2	Alice	71	67
3	Barbara	96	86

wide

```
if TestSubject="Math" then Math=TestScore;
else if TestSubject="Reading" then Reading=TestScore;
```

Activity

1. Examine the last DATA step code in **19-loops.sas** and run the program. What statement is necessary to carry the data from the first iteration over to the second?
2. Add a statement to include only the last row per student in the output table. Run the program.
3. What must be true of the input table for the DATA step to work?

continued...

Activity – Correct Answer

1. Examine the DATA step code and run the program. Add the RETAIN statement and run the program again. Why is the RETAIN statement necessary?

The RETAIN statement hold values in the PDV across multiple iterations of the DATA step. The last row for each student includes both test scores.

Name	Math	Reading
Alfred	82	.
Alfred	.	79
Alice	71	.
Alice	.	67

without RETAIN

Name	Math	Reading
Alfred	82	.
Alfred	82	79
Alice	71	79
Alice	71	67

with RETAIN

Activity – Correct Answer

2. Add a subsetting IF statement to include only the last row per student in the output table.

```
data class_wide;
    set pg2.class_test_narrow;
    by name;
    retain Name Math Reading;
    keep Name Math Reading;
    if TestSubject="Reading" then Reading=TestScore;
    else if TestSubject="Math" then Math=TestScore;
    if last.name=1 then output;
run;
```

3. What must be true of the input table for the DATA step to work?

The data must be sorted by Name.

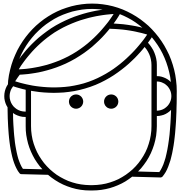
Restructuring Tables

Restructuring Data with the TRANSPOSE Procedure

Restructuring Data with PROC TRANSPOSE

```
PROC TRANSPOSE DATA=input-table <OUT=output-table>;
        <ID col-name;>
        <VAR col-name(s);>
RUN;
```

PROC TRANSPOSE can restructure data with simple statements.



Activity

Open the **20-transpose.sas** program and perform the following tasks:

1. Highlight the PROC PRINT step and run the selection. Note how many rows are in the **sashelp.class** table.
2. Highlight the PROC TRANSPOSE step and run the selection. Answer the following questions:

   Which columns from the input table are transposed into rows?
   What does each column in the output table represent?
   What is the name of the output table?

Keep this program open for the next activity.

Activity – Correct Answer

Which columns from the input table are transposed into rows?

Only the numeric columns are transposed (Age, Height, and Weight).

↳ each column must be all the same data type; by default only numeric cols are transposed

What does each column in the output table represent?

Each column corresponds to a student (row) from the input table.

NAME	COL1	COL2	COL3	COL4	COL5	COL6	COL7
Age	14	13	13	14	14	12	12
Height	69	56.5	65.3	62.8	63.5	57.3	59.8
Weight	112.5	84	98	102.5	102.5	83	84.5

What is the name of the output table?

work.data1

```
NOTE: There were 19 observations read from the data set SASHELP.CLASS.
NOTE: The data set WORK.DATA1 has 3 observations and 20 variables.
```

Keep this program open for the next activity.

Activity

Use the program from the previous activity to perform the following tasks:

1. Add the OUT= option in the PROC TRANSPOSE statement to create an output table named **class_t**.
2. Add the following ID statement and run the step. What changes in the results?

```
id Name;
```

3. Add the following VAR statement and run the step. What changes in the results?

```
var Height Weight;
```

Activity – Correct Answer

```
proc transpose data=sashelp.class out=class_t;
    id Name;
    var Height Weight;
run;
```

The values of the ID column are assigned as column names.

NAME	Alfred	Alice	Barbara	Carol	Henry
Height	69	56.5	65.3	62.8	63.5
Weight	112.5	84	98	102.5	102.5

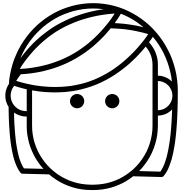
The VAR statement limits the columns that are transposed to rows.

Transposing Values within Groups

```
PROC TRANSPOSE DATA=input-table <OUT=output-table>;
        <VAR col-name(s);>
        <ID col-name;>
        <BY col-name(s);>
RUN;
```

Use the BY statement to transpose data within groups.

The input table must be sorted by the same columns that you specify in the BY statement.



Transposing Values within Groups

Season	Basin	Name	WindRank	WindMPH
1980	EP	AGATHA	1	100
1980	EP	AGATHA	2	95
1980	EP	AGATHA	3	90
1980	EP	AGATHA	4	85
1980	EP	BLAS	1	50
1980	EP	BLAS	2	50
1980	EP	BLAS	3	50
1980	EP	BLAS	4	45
1980	EP	CELIA	1	65

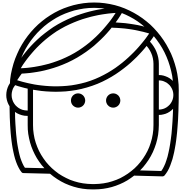
narrow

Season	Basin	Name	Wind1	Wind2	Wind3	Wind4
1980	EP	AGATHA	100	95	90	85
1980	EP	BLAS	50	50	50	45
1980	EP	CELIA	65	65	65	65

wide

```
by Season Basin Name;
```

Each unique combination of BY values creates one row in the output table.



Creating a Wide Table with PROC TRANSPOSE

This demonstration illustrates using PROC TRANSPOSE to transpose data values within groups into rows. The demo also uses options to customize the output table.

Transposing Values into Groups

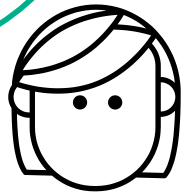
Season	Basin	Name	Wind1	Wind2	Wind3	Wind4
1980	EP	AGATHA	100	95	90	85
1980	EP	BLAS	50	50	50	45
1980	EP	CELIA	65	65	65	65

wide

Season	Basin	Name	WindRank	WindMPH
1980	EP	AGATHA	1	100
1980	EP	AGATHA	2	95
1980	EP	AGATHA	3	90
1980	EP	AGATHA	4	85
1980	EP	BLAS	1	50
1980	EP	BLAS	2	50
1980	EP	BLAS	3	50
1980	EP	BLAS	4	45
1980	EP	CELIA	1	65

narrow

You can also use PROC TRANSPOSE to convert one row into multiple rows.



Activity

Go back to **20-transpose.sas** and perform the following tasks:

1. Run the next program. Notice that, by default, PROC TRANSPOSE transposes all the numeric columns, **Wind1-Wind4**.
2. Add a VAR statement in PROC TRANSPOSE to transpose only the **Wind1** and **Wind2** columns. Run the program.
3. What are the names of the columns that contain the column names and values that have been transposed?

Activity – Correct Answer

2. Add a VAR statement in PROC TRANSPOSE to transpose only the **Wind1** and **Wind2** columns. Run the program.

```
var Wind1 Wind2;
```

3. What are the names of the columns that contain the column names and values that have been transposed? **_NAME_** and **COL1**

Storm Narrow

Obs	Season	Basin	Name	_NAME_	COL1
1	1980	EP	AGATHA	Wind1	100
2	1980	EP	AGATHA	Wind2	95
3	1980	EP	BLAS	Wind1	50
4	1980	EP	BLAS	Wind2	50
5	1980	EP	CELIA	Wind1	65
6	1980	EP	CELIA	Wind2	65

Changing Column Names

PROC TRANSPOSE DATA=*input-table* <OUT=*output-table*>
<NAME=*column*> <PREFIX=*column*>;

```
proc transpose data=pg2.storm_top4_wide name=WindRank
               prefix=WindMPH;
    by Season Basin Name;
    var wind1-wind4;
run;
```

Season	Basin	Name	WindRank	WindMPH1
1980	EP	AGATHA	Wind1	100
1980	EP	AGATHA	Wind2	95
1980	EP	AGATHA	Wind3	90
1980	EP	AGATHA	Wind4	85
1980	EP	BLAS	Wind1	50

Changing Column Names

```
proc transpose data=pg2.storm_top4_wide name=WindRank
               prefix=WindMPH;
    by Season Basin Name;
    var wind1-wind4;
run;
```

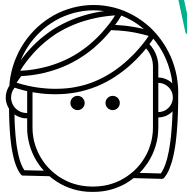
How could you change the name of the column in the output table to exclude the number 1?

Season	Basin	Name	WindRank	WindMPH1
1980	EP	AGATHA	Wind1	100
1980	EP	AGATHA	Wind2	95
1980	EP	AGATHA	Wind3	90
1980	EP	AGATHA	Wind4	85
1980	EP	BLAS	Wind1	50

Changing Column Names

```
proc transpose data=pg2.storm_top4_wide name=WindRank
                out=storm_rotate(rename=(col1=WindMPH));
    by Season Basin Name;
    var wind1-wind4;
run;
```

Create an output table and use the RENAME= data set option.



Season	Basin	Name	WindRank	WindMPH
1980	EP	AGATHA	Wind1	100
1980	EP	AGATHA	Wind2	95
1980	EP	AGATHA	Wind3	90
1980	EP	AGATHA	Wind4	85
1980	EP	BLAS	Wind1	50
1980	EP	BLAS	Wind2	50

Recap PROC TRANSPOSE Options

```
PROC TRANSPOSE DATA=input-table <OUT=output-table>
                <NAME=column> <PREFIX=column>;
```

- OUT – Controls name and library of output table
- NAME – Renames the _NAME_ column
- PREFIX – Changes COL1… to something meaningful

Recap PROC TRANSPOSE Statements

```
PROC TRANSPOSE DATA=input-table <OUT=output-table>;
        <VAR col-name(s);>
        <ID col-name;>
        <BY col-name(s);>
RUN;
```

- VAR – Specifies columns to be transposed (all numeric by default)
- ID – Values of the ID column are assigned as column names or suffix
- BY – Specifies grouping of transposed data

Discussion

When might you prefer to use the DATA step instead of PROC TRANSPOSE to restructure data and vice versa?

Producing Descriptive Statistics

The FREQ Procedure – Prep Guide Chapter 15

Business Scenario

Orion Star management wants to know the number of male and female sales employees in Australia.

Considerations

Use the FREQ procedure to analyze the **Gender** variable in a subset of **orion.sales**.

```
                 The FREQ Procedure

Gender      Frequency       Percent
------------------------------------
 F                 XX         XX.XX
 M                 XX         XX.XX
```

FREQ Procedure

The FREQ procedure produces a one-way frequency table for each variable named in the TABLES statement.

```
proc freq data=orion.sales;
   tables Gender;
   where Country='AU';
run;
```

PROC FREQ DATA=*SAS-data-set*;
    <TABLES *variable(s)* </ *options*>> ;
RUN;

If the TABLES statement is omitted, a one-way frequency table is produced for ***every*** variable in the data set. This can produce a large amount of output and is seldom preferred.

Viewing the Output

A one-way frequency table was created for **Gender**. It lists the discrete values found in the data set and the number of observations in which the variable has that value.

The FREQ Procedure

Gender	Frequency	Percent	Cumulative Frequency	Cumulative Percent
F	27	42.86	27	42.86
M	36	57.14	63	100.00

The default output includes frequency and percentage values, including cumulative statistics.

Options to Suppress Statistics

Use options in the TABLES statement to suppress the display of selected default statistics.

```
TABLES variable(s) / options ;
```

Option	Description
NOCUM	Suppresses the cumulative statistics.
NOPERCENT	Suppresses the percentage display.

Options to Suppress Statistics

NOCUM suppresses

The FREQ Procedure

Gender	Frequency	Percent	Cumulative Frequency	Cumulative Percent
F	27	42.86	27	42.86
M	36	57.14	63	100.00

NOPERCENT suppresses

Short Answer Poll

What change was needed to correct this program?

```
proc freq data=orion.sales;
   tables country nocum nopercent;
run;
```

Short Answer Poll – Correct Answer

What change was needed? **A slash is required before the options in the TABLES statement.**

```
31   proc freq data=orion.sales;
32       tables country nocum nopercent;
ERROR: Variable NOCUM not found.
ERROR: Variable NOPERCENT not found.
33   run;
```

```
proc freq data=orion.sales;
   tables country / nocum nopercent;
run;
```

The FREQ Procedure

Country	Frequency
AU	63
US	102

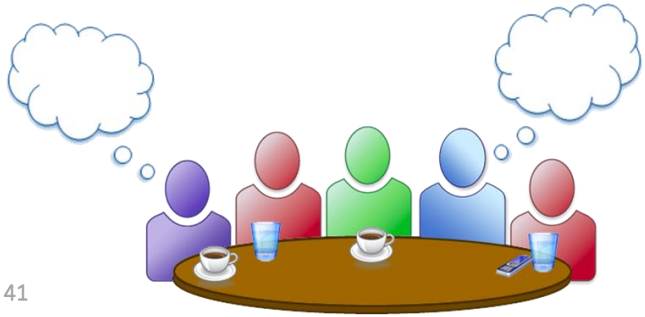
Idea Exchange

This step creates a table for every variable in the data set:

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

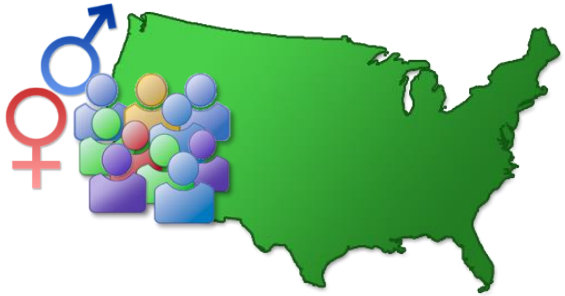
- Employee_ID
- First_Name
- Last_Name
- Gender
- Salary
- Job_Title
- Country
- Birth_Date
- Hire_Date

Which variables are most appropriate for a frequency analysis? Why?



Business Scenario

Orion Star management wants to know how many sales employees are in each country, as well as the count of males and females.



TABLES Statement

You can list multiple variables in a TABLES statement.
A separate table is produced for each variable.

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

The FREQ Procedure

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

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

BY Statement

The BY statement is used to request separate analyses for each BY group.

```
proc sort data=orion.sales out=sorted;
   by Country;
run;

proc freq data=sorted;
   tables Gender;
   by Country;
run;
```

The data set must be sorted or indexed by the variable (or variables) named in the BY statement.

Viewing the Output

Each group appears on a separate page with a BY line.

Country=AU

Gender	Frequency	Percent	Cumulative Frequency	Cumulative Percent
F	27	42.86	27	42.86
M	36	57.14	63	100.00

Country=US

Gender	Frequency	Percent	Cumulative Frequency	Cumulative Percent
F	41	40.20	41	40.20
M	61	59.80	102	100.00

Crosstabulation Table

An asterisk between two variables generates a two-way frequency table, or *crosstabulation table*.

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

rows | columns

A two-way frequency table generates a single table with statistics for each distinct combination of values of the selected variables.

Viewing the Output

PROC FREQ Output

The FREQ Procedure

Frequency Percent Row Pct Col Pct	Table of Gender by Country		
		Country	
Gender	AU	US	Total
F	27 16.36 39.71 42.86	41 24.85 60.29 40.20	68 41.21
M	36 21.82 37.11 57.14	61 36.97 62.89 59.80	97 58.79
Total	63 38.18	102 61.82	165 100.00

Options to Suppress Statistics

Use options in the TABLES statement to suppress the display of selected default statistics.

TABLES *variable(s)* / *options* ;

Option	Description
NOROW	Suppresses the display of the row percentage.
NOCOL	Suppresses the display of the column percentage.
NOPERCENT	Suppresses the percentage display.
NOFREQ	Suppresses the frequency display.

Options to Suppress Statistics

The FREQ Procedure

Frequency
Percent
Row Pct
Col Pct

Table of Gender by Country

Gender	Country AU	Country US	Total
F	27 16.36 39.71 42.86	41 24.85 60.29 40.20	68 41.21
M	36 21.82 37.11 57.14	61 36.97 62.89 59.80	97 58.79
Total	63 38.18	102 61.82	165 100.00

NOFREQ suppresses

NOPERCENT suppresses

Options to Suppress Statistics

The FREQ Procedure

Frequency Percent Row Pct Col Pct	Table of Gender by Country		
	Country		
Gender	AU	US	Total
F	27 16.36 39.71 42.86	41 24.85 60.29 40.20	68 41.21
M	36 21.82 37.11 57.14	61 36.97 62.89 59.80	97 58.79
Total	63 38.18	102 61.82	165 100.00

NOROW suppresses

NOCOL suppresses

Creating N-Way Tables

The FREQ procedure can create a series of two-way tables with a table for each level of the other tables.

```
proc freq data=orion.sales;
   tables country*gender*job_title;
run;
```

- You can include up to 50 variables in a single request.

Creating N-Way Tables

Partial output:

Frequency Percent Row Pct Col Pct	Table 1 of Gender by Job_Title Controlling for Country=AU							
	Job_Title							
Gender	Chief Sales Officer	Sales Manager	Sales Rep. I	Sales Rep. II	Sales Rep. III	Sales Rep. IV	Senior Sales Manager	Total
F	0 0.00 0.00 .	0 0.00 0.00 0.00	8 12.70 29.63 38.10	10 15.87 37.04 55.56	7 11.11 25.93 41.18	2 3.17 7.41 40.00	0 0.00 0.00 .	27 42.86
M	0 0.00 0.00 .	2 3.17 5.56 100.00	13 20.63 36.11 61.90	8 12.70 22.22 44.44	10 15.87 27.78 58.82	3 4.76 8.33 60.00	0 0.00 0.00 .	36 57.14
Total	0 0.00	2 3.17	21 33.33	18 28.57	17 26.98	5 7.94	0 0.00	63 100.00

LIST and CROSSLIST Options

You can use the LIST and CROSSLIST options in the TABLES statement to "flatten" the output.

```
proc freq data=orion.sales;
   tables Gender*Country /list;
run;
```

Gender	Country	Frequency	Percent	Cumulative Frequency	Cumulative Percent
F	AU	27	16.36	27	16.36
F	US	41	24.85	68	41.21
M	AU	36	21.82	104	63.03
M	US	61	36.97	165	100.00

LIST and CROSSLIST Options

You can use the LIST and CROSSLIST options in the TABLES statement to "flatten" the output.

```
proc freq data=orion.sales;
   tables Gender*Country /crosslist;
run;
```

Table of Gender by Country					
Gender	Country	Frequency	Percent	Row Percent	Column Percent
F	AU	27	16.36	39.71	42.86
	US	41	24.85	60.29	40.20
	Total	68	41.21	100.00	
M	AU	36	21.82	37.11	57.14
	US	61	36.97	62.89	59.80
	Total	97	58.79	100.00	
Total	AU	63	38.18		100.00
	US	102	61.82		100.00
	Total	165	100.00		

Business Scenario

A new data set, **orion.nonsales2**, must be validated. It contains information about non-sales employees and might include invalid and missing values.

Partial **orion.nonsales2**

Employee_ ID	First	Last	Gender	Salary	Job_Title	Country
120101	Patrick	Lu	M	163040	Director	AU
120104	Kareen	Billington	F	46230	Admin Mgr	au
120105	Liz	Povey	F	27110	Secretary I	AU
120106	John	Hornsey	M	.	Office Asst II	AU
120107	Sherie	Sheedy	F	30475	Office Asst II	AU
120108	Gladys	Gromek	F	27660	Warehouse Asst II	AU

Considerations

Use the FREQ procedure to screen for invalid, missing, and duplicate data values.

Requirements of non-sales employee data:

- **Employee_ID** values must be unique and not missing.
- **Gender** must be *F* or *M*.
- **Job_Title** must not be missing.
- **Country** must have a value of *AU* or *US*.
- **Salary** values must be in the numeric range of 24000 to 500000.

Short Answer Poll

What problems exist with the data in this partial data set?

Employee_ID	First	Last	Gender	Salary	Job_Title	Country
120101	Patrick	Lu	M	163040	Director	AU
120104	Kareen	Billington	F	46230	Administration Manager	au
120105	Liz	Povey	F	27110	Secretary I	AU
120106	John	Hornsey	M	.	Office Assistant II	AU
120107	Sherie	Sheedy	F	30475	Office Assistant III	AU
120108	Gladys	Gromek	F	27660	Warehouse Assistant II	AU
120108	Gabriele	Baker	F	26495	Warehouse Assistant I	AU
120110	Dennis	Entwisle	M	28615	Warehouse Assistant III	AU
120111	Ubaldo	Spillane	M	26895	Security Guard II	AU
120112	Ellis	Glattback	F	26550		AU
120113	Riu	Horsey	F	26870	Security Guard II	AU
120114	Jeannette	Buddery	G	31285	Security Manager	AU
120115	Hugh	Nichollas	M	2650	Service Assistant I	AU
.	Austen	Ralston	M	29250	Service Assistant II	AU
120117	Bill	Mccleary	M	31670	Cabinet Maker III	AU
120118	Darshi	Hartshorn	M	28090	Cabinet Maker II	AU

Hint: There are seven data problems.

Short Answer Poll – Correct Answer

What problems exist with the data in this partial data set?

Employee_ID	First	Last	Gender	Salary	Job_Title	Country
120101	Patrick	Lu	M	163040	Director	AU
120104	Kareen	Billington	F	46230	Administration Manager	au
120105	Liz	Povey	F	27110	Secretary I	AU
120106	John	Hornsey	M	.	Office Assistant II	AU
120107	Sherie	Sheedy	F	30475	Office Assistant III	AU
120108	Gladys	Gromek	F	27660	Warehouse Assistant II	AU
120108	Gabriele	Baker	F	26495	Warehouse Assistant I	AU
120110	Dennis	Entwisle	M	28615	Warehouse Assistant III	AU
120111	Ubaldo	Spillane	M	26895	Security Guard II	AU
120112	Ellis	Glattback	F	26550		AU
120113	Riu	Horsey	F	26870	Security Guard II	AU
120114	Jeannette	Buddery	G	31285	Security Manager	AU
120115	Hugh	Nichollas	M	2650	Service Assistant I	AU
.	Austen	Ralston	M	29250	Service Assistant II	AU
120117	Bill	Mccleary	M	31670	Cabinet Maker III	AU
120118	Darshi	Hartshorn	M	28090	Cabinet Maker II	AU

Hint: There are seven data problems.

FREQ Procedure for Data Validation

The FREQ procedure lists all discrete values for a variable and reports missing values.

```
proc freq data=orion.nonsales2;
   tables Gender Country / nocum nopercent;
run;
```

Viewing the Output

PROC FREQ Output

```
The FREQ Procedure

Gender      Frequency
------------------------
F                 110
G                   1
M                 123

Frequency Missing = 1


Country     Frequency
------------------------
AU                 33
US                196
au                  3
us                  3
```

NLEVELS Option

The *NLEVELS option* displays a table that provides the number of distinct values for each analysis variable.

```
proc freq data=orion.nonsales2 nlevels;
   tables Gender Country / nocum nopercent;
run;
```

PROC FREQ DATA=*SAS-data-set* NLEVELS;
    TABLES *variable(s)* ;
RUN;

Viewing the Output

PROC FREQ Output

Number of Variable Levels			
Variable	Levels	Missing Levels	Nonmissing Levels
Gender	4	1	3
Country	4	0	4

Gender	Frequency
F	110
G	1
M	123
Frequency Missing = 1	

Country	Frequency
AU	33
US	196
au	3
us	3

Check for Uniqueness

The values of **Employee_ID** must be unique and not missing. PROC FREQ can be used to check for duplicate or missing values.

```
proc freq data=orion.nonsales2 order=freq;
   tables Employee_ID / nocum nopercent;
run;
```

The ORDER=FREQ option displays the results in descending frequency order.

Viewing the Output

Partial PROC FREQ Output

The FREQ Procedure

Employee_ID	Frequency
120108	2
120101	1
120104	1
120105	1
120106	1
121134	1
121141	1
121142	1
121146	1
121147	1
121148	1

Frequency Missing = 1

NLEVELS Option

NLEVELS can also be used to identify duplicates, when the number of distinct values is known.

```
proc freq data=orion.nonsales2 nlevels;
   tables Employee_ID / noprint;
run;
```

This example uses the NOPRINT option to suppress the frequency table. Only the Number of Variable Levels table is displayed.

Viewing the Output

Partial PROC FREQ Output

The FREQ Procedure

Number of Variable Levels

Variable	Levels	Missing Levels	Nonmissing Levels
Employee_ID	234	1	233

There are 235 employees, but there are only 234 distinct **Employee_ID** values. Therefore, there is one duplicate value and one missing value for **Employee_ID**.

NLEVELS Option

The _ALL_ keyword with the NOPRINT option displays the number of levels for all variables without displaying frequency counts.

```
proc freq data=orion.nonsales2 nlevels;
   tables _all_ / noprint;
run;
```

Viewing the Output

PROC FREQ Output

The FREQ Procedure
Number of Variable Levels

Variable	Levels	Missing Levels	Nonmissing Levels
Employee_ID	234	1	233
First	204	0	204
Last	228	0	228
Gender	4	1	3
Salary	230	1	229
Job_Title	125	1	124
Country	4	0	4

No frequency tables were displayed.

Identifying Observations with Invalid Data

PROC FREQ uncovered the existence of invalid data values for **Gender**, **Country**, and **Employee_ID**. Use PROC PRINT to display the observations with invalid values.

```
proc print data=orion.nonsales2;
   where Gender not in ('F','M') or
         Country not in ('AU','US') or
             Job_Title is null or
             Employee_ID is missing or
             Employee_ID=120108;
run;
```

Viewing the Output

PROC PRINT Output

Obs	Employee_ID	First	Last	Gender	Salary	Job_Title	Country
2	120104	Kareen	Billington	F	46230	Administration Manager	au
6	120108	Gladys	Gromek	F	27660	Warehouse Assistant II	AU
7	120108	Gabriele	Baker	F	26495	Warehouse Assistant I	AU
10	120112	Ellis	Glattback	F	26550		AU
12	120114	Jeannette	Buddery	G	31285	Security Manager	AU
14	.	Austen	Ralston	M	29250	Service Assistant II	AU
84	120695	Trent	Moffat	M	28180	Warehouse Assistant II	au
87	120698	Geoff	Kistanna	M	26160	Warehouse Assistant I	au
101	120723	Deanna	Olsen		33950	Corp. Comm. Specialist II	US
125	120747	Zashia	Farthing	F	43590	Financial Controller I	us
197	120994	Danelle	Sergeant	F	31645	Office Administrator I	us
200	120997	Mary	Donathan	F	27420	Shipping Administrator I	us

original observation numbers

Business Scenario

The manager of Human Resources requested a report that shows the number and percent of sales employees who are hired each year.

Using Formats in PROC FREQ

A FORMAT statement can be used in PROC FREQ to format data values.

```
proc freq data=orion.sales;
   tables Hire_Date / nocum;
   format Hire_Date date9.;
run;
```

Partial PROC FREQ Output

The FREQ Procedure

Hire_Date	Frequency	Percent
01JAN1978	17	10.30
01FEB1978	2	1.21
01APR1978	1	0.61
01JUL1978	1	0.61
01AUG1978	1	0.61

many discrete values, and not what the manager requested

Using Formats in PROC FREQ

A FORMAT statement can also be used in PROC FREQ to group the data.

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

Partial PROC FREQ Output

The FREQ Procedure

Hire_Date	Frequency	Percent
1978	23	13.94
1979	2	1.21
1980	4	2.42
1981	3	1.82
1982	7	4.24

fewer discrete values

Short Answer Poll

Can user-defined formats be used to group data?

Short Answer Poll – Correct Answer

Can user-defined formats be used to group data? **yes**

The FREQ Procedure

Salary	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Tier1	1	0.61	1	0.61
Tier2	158	95.76	159	96.36
Tier3	4	2.42	163	98.79
Tier4	2	1.21	165	100.00

Lesson Quiz

6. Which statement is false concerning the TRANSPOSE procedure?

a. Columns are transposed into rows.
b. By default, numeric columns are transposed.
c. Use a BY statement to sort the data while transposing.
d. Use a VAR statement to specifically specify the character and numeric columns to transpose.

6. Which statement is false concerning the TRANSPOSE procedure?

a. Columns are transposed into rows.

b. By default, numeric columns are transposed.

c. Use a BY statement to sort the data while transposing.

d. Use a VAR statement to specifically specify the character and numeric columns to transpose.

7. Which statements are needed in a PROC TRANSPOSE step for the following example (narrow ⇨ wide)?

Day	Meal	Food
Saturday	Breakfast	Yogurt
Saturday	Lunch	Sandwich
Saturday	Dinner	Steak
Sunday	Breakfast	Pancakes
Sunday	Lunch	Salad
Sunday	Dinner	Lasagna

Day	Breakfast	Lunch	Dinner
Saturday	Yogurt	Sandwich	Steak
Sunday	Pancakes	Salad	Lasagna

a.
```
by Day;
var Meal Food;
```

b.
```
id Day;
var Food Meal;
```

c.
```
by Day;
id Food;
var Meal;
```

d.
```
by Day;
id Meal;
var Food;
```

7. Which statements are needed in a PROC TRANSPOSE step for the following example (narrow ⇨ wide)?

Day	Meal	Food
Saturday	Breakfast	Yogurt
Saturday	Lunch	Sandwich
Saturday	Dinner	Steak
Sunday	Breakfast	Pancakes
Sunday	Lunch	Salad
Sunday	Dinner	Lasagna

Day	Breakfast	Lunch	Dinner
Saturday	Yogurt	Sandwich	Steak
Sunday	Pancakes	Salad	Lasagna

a.
```
by Day;
var Meal Food;
```

b.
```
id Day;
var Food Meal;
```

c.
```
by Day;
id Food;
var Meal;
```

d.
```
by Day;
id Meal;
var Food;
```

8. Which statement or statements are needed in a PROC TRANSPOSE step for the following example (wide ⇨ narrow)?

Day	Breakfast	Lunch	Dinner
Saturday	Yogurt	Sandwich	Steak
Sunday	Pancakes	Salad	Lasagna

NAME	COL1	COL2
Breakfast	Yogurt	Pancakes
Lunch	Sandwich	Salad
Dinner	Steak	Lasagna

a.
```
by Day;
```

b.
```
var Breakfast Lunch Dinner;
```

c.
```
id Day;
```

d.
```
id Day;
var Breakfast Lunch Dinner;
```

8. Which statement or statements are needed in a PROC TRANSPOSE step for the following example (wide ⇨ narrow)?

Day	Breakfast	Lunch	Dinner
Saturday	Yogurt	Sandwich	Steak
Sunday	Pancakes	Salad	Lasagna

NAME	COL1	COL2
Breakfast	Yogurt	Pancakes
Lunch	Sandwich	Salad
Dinner	Steak	Lasagna

a.
```
by Day;
```

b.
```
var Breakfast Lunch Dinner;
```

c.
```
id Day;
```

d.
```
id Day;
var Breakfast Lunch Dinner;
```

9. Which option is needed in the PROC TRANSPOSE statement to rename the **_NAME_** column?

NAME	COL1	COL2
Breakfast	Yogurt	Pancakes
Lunch	Sandwich	Salad
Dinner	Steak	Lasagna

Meal	COL1	COL2
Breakfast	Yogurt	Pancakes
Lunch	Sandwich	Salad
Dinner	Steak	Lasagna

a. **`_name_=Meal`**

b. **`name=Meal`**

c. **`prefix=Meal`**

d. **`rename=Meal`**

9. Which option is needed in the PROC TRANSPOSE statement to rename the **_NAME_** column?

NAME	COL1	COL2
Breakfast	Yogurt	Pancakes
Lunch	Sandwich	Salad
Dinner	Steak	Lasagna

Meal	COL1	COL2
Breakfast	Yogurt	Pancakes
Lunch	Sandwich	Salad
Dinner	Steak	Lasagna

a. **`_name_=Meal`**

b. **`name=Meal`**

c. **`prefix=Meal`**

d. **`rename=Meal`**

10. Which option is needed in the PROC TRANSPOSE statement to rename the **COL** columns?

Meal	COL1	COL2
Breakfast	Yogurt	Pancakes
Lunch	Sandwich	Salad
Dinner	Steak	Lasagna

Meal	Day7	Day1
Breakfast	Yogurt	Pancakes
Lunch	Sandwich	Salad
Dinner	Steak	Lasagna

a. `out=meals2(COL1=Day_7 COL2=Day_1)`

b. `out=meals2(name=(COL1=Day_7 COL2=Day_1))`

c. `out=meals2(rename=(COL1=Day_7 COL2=Day_1))`

d. `out=meals2(prefix=(COL1=Day_7 COL2=Day_1))`

10. Which option is needed in the PROC TRANSPOSE statement to rename the **COL** columns?

Meal	COL1	COL2
Breakfast	Yogurt	Pancakes
Lunch	Sandwich	Salad
Dinner	Steak	Lasagna

Meal	Day7	Day1
Breakfast	Yogurt	Pancakes
Lunch	Sandwich	Salad
Dinner	Steak	Lasagna

a. `out=meals2(COL1=Day_7 COL2=Day_1)`

b. `out=meals2(name=(COL1=Day_7 COL2=Day_1))`

c. `out=meals2(rename=(COL1=Day_7 COL2=Day_1))`

d. `out=meals2(prefix=(COL1=Day_7 COL2=Day_1))`

1. Which of these procedures produces output that is most useful for detecting duplicate values?

   a. PROC PRINT
   b. PROC FREQ
   c. PROC MEANS
   d. PROC UNIVARIATE

1. Which of these procedures produces output that is most useful for detecting duplicate values?

   a. PROC PRINT
   b. PROC FREQ
   c. PROC MEANS
   d. PROC UNIVARIATE

2. Which of these programs is most useful for determining the exact observation that contains a numeric variable with an extreme value?

a.
```
proc print data=sales.totals;
    var ProdNum Sales Region;
run;
```

b.
```
proc freq data=sales.totals;
    tables ProdNum Sales Region;
run;
```

c.
```
proc univariate data=sales.totals;
run;
```

2. Which of these programs is most useful for determining the exact observation that contains a numeric variable with an extreme value?

a.
```
proc print data=sales.totals;
    var ProdNum Sales Region;
run;
```

b.
```
proc freq data=sales.totals;
    tables ProdNum Sales Region;
run;
```

c.
```
proc univariate data=sales.totals;
run;
```

3. A PROC FREQ analysis identified invalid and missing values in a data set. Which of these procedures displays the observations that contain invalid or missing values?

   a. PROC PRINT
   b. PROC FREQ
   c. PROC MEANS
   d. PROC UNIVARIATE

3. A PROC FREQ analysis identified invalid and missing values in a data set. Which of these procedures displays the observations that contain invalid or missing values?

   a. PROC PRINT
   b. PROC FREQ
   c. PROC MEANS
   d. PROC UNIVARIATE

4. Which PROC FREQ step creates the output shown here?

a.
```
proc freq data=orion.qtr1_2007;
    tables Order_Type;
run;
```

b.
```
proc freq data=orion.qtr1_2007
          nlevels;
    tables Order_Type / nocum;
run;
```

c.
```
proc freq data=orion.qtr1_2007
          nlevels;
    tables Order_Type / noprint;
run;
```

d.
```
proc freq data=otion.qtr1_2007
          nlevels;
    tables Order_Type nocum;
run;
```

Number of Variable Levels		
Variable	Label	Levels
Order_Type	Order Type	3

Order Type		
Order_Type	Frequency	Percent
1	13	59.09
2	2	9.09
3	7	31.82

4. Which PROC FREQ step creates the output shown here?

a.
```
proc freq data=orion.qtr1_2007;
    tables Order_Type;
run;
```

b.
```
proc freq data=orion.qtr1_2007
        nlevels;
    tables Order_Type / nocum;
run;
```

c.
```
proc freq data=orion.qtr1_2007
        nlevels;
    tables Order_Type / noprint;
run;
```

d.
```
proc freq data=otion.qtr1_2007
        nlevels;
    tables Order_Type nocum;
run;
```

Number of Variable Levels		
Variable	Label	Levels
Order_Type	Order Type	3

Order Type		
Order_Type	Frequency	Percent
1	13	59.09
2	2	9.09
3	7	31.82

5. This PROC MEANS step creates all of the statistics listed below.

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

- minimum and maximum
- the total number of observations that PROC MEANS processes for each subgroup (**N Obs**)
- mean and standard deviation
- the number of nonmissing values (**N**)

❍ True

❍ False

5. This PROC MEANS step creates all of the statistics listed below.

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

minimum and maximum

- the total number of observations that PROC MEANS processes for each subgroup (**N Obs**)
- mean and standard deviation
- the number of nonmissing values (**N**)

❍ True

❍ False

6. What must be added to the PROC MEANS statement to produce this output?

```
proc means data=orion.customer_dim
                                   ;
   var Customer_Age;
   class Customer_Gender;
   where Customer_Country ne 'US';
run;
```

The MEANS Procedure

Analysis Variable : Customer_Age Customer Age		
Customer Gender	Range	Mean
F	54.0	35.1
M	54.0	47.0

a. nonobs

b. range mean

c. range mean nonobs bestw.

d. range mean nonobs maxdec=1

6. What must be added to the PROC MEANS statement to produce this output?

```
proc means data=orion.customer_dim
                                   ;
   var Customer_Age;
   class Customer_Gender;
   where Customer_Country ne 'US';
run;
```

The MEANS Procedure

Analysis Variable : Customer_Age Customer Age		
Customer Gender	Range	Mean
F	54.0	35.1
M	54.0	47.0

a. nonobs

b. range mean

c. range mean nonobs bestw.

d. range mean nonobs maxdec=1

7. Which option enables you to specify the number of extreme observations that are displayed by PROC UNIVARIATE?

   a. NEXTROBS=
   b. NLEVELS
   c. NOPRINT
   d. _ALL_

7. Which option enables you to specify the number of extreme observations that are displayed by PROC UNIVARIATE?

a. NEXTROBS=
b. NLEVELS
c. NOPRINT
d. _ALL_