STAT 425 and STAT 625
Statistical Software

Lecture 9
Summarizing and Reporting your Data

Saida Zardi

SAS Procedure

- PROC statement
  - *proc content data=banana;*
  - *Libname tropical 'J:CLASSES\STAT46';*
  - *proc content data=learn.banana;*
  - *proc contnent data = 'J:\CLASSES\STAT46\banana';*
- BY statement

  By State;

SAS Procedure

- TITLES and FOOTNOTES

  Title 'This is a Title';

  Footnote 'This is a footnote';

  Footnote3 'this is footnote# 3';

- Label statement

  Label Recepdate = 'Date package was received';

Sorting your data: PROC SORT

Proc sort;

By var1 var2 .....

- Controlling the output data set:

Proc sort data = messy out = neat ;

PROC sort DATA = messy OUT = neat
NODUPKEY DUPOUT = extraobs ;

Sorting your data: PROC SORT

- **Ascending versus Descending:**

  BY State DESCENDING city

Example:

```
data Marine;
input Name $ Family $ Length @@;
datalines;
beluga whale 15
basking shark 30
gray whale 50
mako shark 12
dwarf shark .5
humback . 50
blue whale 100
whale shark 40
sperm whale 60
whale shark 40
killer whale 30
;
```

```
proc sort data=Marine out=seasort NODUPKEY;
  by family descending length;
proc print data=seasort;
  title 'Whales and Sharks';
run;
```

Whales and Sharks

Obs	Name	Family	Length
1	humback		50.0
2	whale	shark	40.0
3	basking	shark	30.0
4	mako	shark	12.0
5	dwarf	shark	0.5
6	blue	whale	100.0
7	sperm	whale	60.0
8	gray	whale	50.0
9	killer	whale	30.0
10	beluga	whale	15.0

Changing the Sort Order For Character Variables

- ASCII versus EBCDIC:

ASCII	BLANK	numerals	Uppercase letters	Lowercase letters
EBCDIC	BLANK	Lowercase letters	Uppercase letters	numerals

- Options: *SORTSEQ = ASCII* and *SORTSEQ = EBCDIC*
- Linguistic sorting:

SORTSEQ = LINGUISTIC option with *STRENGTH = PRIMARY* suboption

Changing the Sort Order For Character Variables

- Example:

```
Proc sort data = salesone out=selone sortseq = linguistic (strength = primary);
    by customer;
proc print data = selone;
 title 'Sales sorted by Customer';
run;

proc sort data=salesone out=seltwo sortseq = Linguistic (Numeric_collation = on);
    by item;
proc print data=seltwo;
    title 'Sales sorted by item';
run;
```

Changing the Sort Order For Character Variables

Sales sorted by Customer

Obs	EmpID	Name	Region	Customer	Item	Quantity	UnitCost
1	1843	George Smith	North	Barco Corporation	144L	50	8.99
2	1843	George Smith	North	Barco Corporation	908X	1	5129.00
3	0177	Glenda Johnson	East	Barco Corporation	733	2	10000.00
4	1843	George Smith	South	Cost Cutter's	122	100	5.99
5	1843	George Smith	South	Cost Cutter's	855W	1	9109.00
6	9888	Sharon Lu	West	Cost Cutter's	122	50	5.99
7	1843	George Smith	South	Ely Corp.	122L	10	29.95
8	0177	Glenda Johnson	East	Food Unlimited	188X	100	6.99
9	1843	George Smith	North	Minimart Inc.	188S	3	5199.00
10	0177	Glenda Johnson	North	Minimart Inc.	777	5	10.50
11	1843	George Smith	North	Minimart Inc.	188S	3	5199.00

Changing the Sort Order For Character Variables

Sales sorted by item

Obs	EmpID	Name	Region	Customer	Item	Quantity	UnitCo
1	9888	Sharon Lu	West	Pet's are Us	100W	1000	1.9
2	1843	George Smith	South	Cost Cutter's	122	100	5.9
3	9888	Sharon Lu	West	Cost Cutter's	122	50	5.9
4	1843	George Smith	South	Ely Corp.	122L	10	29.9
5	0017	Jason Nguyen	East	Roger's Spirits	122L	500	39.9
6	1843	George Smith	North	Barco Corporation	144L	50	8.9
7	0177	Glenda Johnson	East	Shop and Drop	144L	100	8.9
8	1843	George Smith	North	Minimart Inc.	188S	3	5199.0
9	1843	George Smith	North	Minimart Inc.	188S	3	5199.0
10	0177	Glenda	East	Food Unlimited	188X	100	6.9

Printing your data: Proc Print

- Proc Print requires just one statement and uses by default the most recent SAS data set:

  Proc Print;

- To specify the data set:

  Proc Print Data= 'data set'

- To print without the observations numbers and have the labels printed instead of the variable names:

  Proc Print Data= 'data set' NOOBS LABEL;

Printing your data: Proc Print

- Other proc print options:
  - *BY variable – list;*
  - *ID variable – list;*
  - *SUM variable – list;*
  - *VAR variable – list;*

Example:

```
data selthree;
   set Lec9.sales;
proc sort data=selthree;
     by Name;
proc print data=selthree;
     by Name;
     sum Quantity;
     var Region Quantity UnitCost;
     title 'Sales by EmplID';
run;
```

Printing your data: Proc Print

Sales by EmpID

Name=George Smith

Obs	Region	Quantity	UnitCost
1	North	50	8.99
2	South	100	5.99
3	North	3	5199.00
4	North	1	5129.00
5	South	10	29.95
6	South	1	9109.00
7	North	3	5199.00
Name		168	

Name=Glenda Johnson

Obs	Region	Quantity	UnitCost
8	East	100	6.99
9	East	100	8.99
10	North	5	10.50
11	East	2	10000.00
Name		207	

Name=Jason Nguyen

Obs	Region	Quantity	UnitCost
12	East	500	39.99
13	South	100	19.95
Name		600	

Using Formats to change the Appearance of Printed Values

- General forms of a SAS format:

Character	Numeric	Date
$format.	*formatw.d*	*formatw.*

- Format statement:

Format *sales DOLLAR8.2 Date MMDDYY8. ;*

- Put statement useful when writing raw data files or reports:

Put *sales DOLLAR8.2 Date MMDDYY8. ;*

Creating your own format: PROC FORMAT

The FORMAT procedures creates formats that will be later used in a format statement. The procedures starts with a PROC FORMAT and continues with VALUE statements:

```
PROC FORMAT;
        VALUE   name  range-1 = 'formatted text 1'
                      range-2= 'formatted text 2'
                      ..................................
                      range-n = 'formatted text n'  ;
```

- *name in the VALUE statement is the name of the format you're creating.*
- *If character data, the name must start with a $.*

- The *name* cannot be more than 32 characters, including the $ sign for character data.
- *name* must not start or end with a number and must not contain any special character except underscore
- *name* cannot be the name of an existing format
- *range* is the value given to the text in between quotation marks
- The text can be up to 32,767 characters long, but some procedures print only the first 8 to 16 characters.

- Character values should be enclosed in between quotation marks
- The keywords LOW and HIGH: used to indicate lowest and highest values for the variable.
- The keyword OTHER: used to assign a format to any non listed values in the VALUE statement.

Example:

```
data Medic;
  set Lec9.medical;
proc print data = Medic;
    var Clinic VisitDate HR Weight;
    format VisitDate DATE9.;
    Title 'Medical Data using Formats';
run;
```

Medical Data using Formats

Obs	Clinic	VisitDate	HR	Weight
1	Mayo Clinic	21OCT2006	78	120
2	HMC	01SEP2006	58	166
3	Mayo Clinic	01OCT2006	68	210
4	HMC	11NOV2006	88	288
5	Mayo Clinic	01MAY2006	54	180
6	HMC	06JUL2006	60	199

REPORTING

PROC REPORT

Simple Custom Reports

For reporting:

- *Proc Print* is flexible and easy to use, but has its limitation.
- Use *File* and *PUT* statements in a *DATA* step:
  - *FILE statement:*

    FILE 'file – specification' PRINT;

    PRINT option to include carriage returns and page breaks needed for printing.

- PUT statements:
  - can be in list, column or formatted style, as in INPUT statements
  - Don’t take $ symbol after character variables
  - Control spacing with:
    - @*n* to move to column n
    - +*n* to move n columns
    - / to skip to the next line
    - #*n* to skip to line n
    - @ to hold the current line
    - Quotation marks to enclose a text

Example:

```
Data _NULL_;
    set Lec9.Sales;
    File 'J:\CLASSES\STAT46\Customers.txt'  PRINT;
    title;

    PUT @5 'Sales report for ' Name 'from region ' Region 'customer of ' Customer
        // @5 'your total sales are ' TotalSales ;
    Put _Page_;
run;
```

Un extract of the file Customers.txt

```
Sales report for George Smith from region North customer of Barco Corporation

your total sales are 449.5

Sales report for George Smith from region South customer of Cost Cutter's

your total sales are 599

Sales report for George Smith from region North customer of Minimart Inc.

your total sales are 15597
```

PROC REPORT

General form of a basic REPORT procedure:

```
PROC REPORT NOWINDOWS;

    COLUMN  variable list ;
```

Defaults for Numeric versus Character data:

- One character variable or more -→ a detailed report with one row per observation
- For numeric variable →Proc REPORT will sum the variables

Example

```
Proc Report Data=Medic NOWINDOWS;
     Column Clinic VisitDate Weight;
      title 'Report with Character andNumeric Variables';
run;

Proc Report data= Medic NOWINDOWS;
     column Weight;
     title 'Report with numeric variable only';
run;
```

Report with Character andNumeric Variables

Clinic	Visit Date	Weight
Mayo Clinic	10/21/2006	120
HMC	09/01/2006	166
Mayo Clinic	10/01/2006	210
HMC	11/11/2006	288
Mayo Clinic	05/01/2006	180
HMC	07/06/2006	199

Report with numeric variable only

Weight
1163

The DEFINE Statement

The DEFINE statement specifies options for an individual variable:

DEFINE variable / options 'column-header' ;

Possible values of usage options include:

ACROSS	creates a column for each unique value of the variable
ANALYSIS	calculates statistics for the variable
DISPLAY	Creates one row for each variable in the data set.
GROUP	Creates one row for each unique value of the variable.
ORDER	Creates one row for each observation with rows arranged according to the values of the order variable.

Example:

```
Proc Report data=medic NOWINDOWS;
     COLUMN Clinic VisitDate Weight HR;
     DEFINE Clinic / group;
     Define Weight / Analysis;
     Title 'Medical data arranged by Clinic group';
run;
```

```
proc report data=selthree NOWINDOWS;
     column Region Quantity TotalSales;
     Define Region /Group;
     Define TotalSales/Analysis ' Total Sales ';
     title 'Total Sales by Region';
run;
```

Medical data arranged by Clinic group

Clinic	Visit Date	Weight	Heart Rate
HMC	01/15/2100	653	206
Mayo Clinic	11/19/2099	510	200

Total Sales by Region

Region	Quantity	Total Sales
East	702	41593
North	62	36825
South	211	12002.5
West	1050	2289.5

Creating Summary Reports

Two usage types make the REPORT procedure to gather data into summary groups based on the values of a variable:

Group variables: Specify the GROUP usage in a DEFINE statement:

```
PROC REPORT DATA= employees NOWINDOWS ;
COLUMN  department  Salary  Bonus ;
DEFINE department  / GROUP;
```

Creating Summary Reports

Across Variables: use a DEFINE statement, the default produces counts, or specify the desired statistic (MAX, MIN, MEAN,…).

Example:

```
Proc report data=selthree NOWINDOWS;
     column Name   Region ,  (Quantity TotalSales);
     Define Name/group ;
     Define Region /Across;

     title 'Summary Report with a Group and Across Variable';
run;
```

Creating Summary Reports

Part of the example output:

Summary Report with a Group and Across Variable

	Region						
	East		North		South		
Name	Quantity	TotalSales	Quantity	TotalSales	Quantity	TotalSales	Quant
George Smith	.	.	57	36772.5	111	10007.5	
Glenda Johnson	202	21598	5	52.5	.	.	
Jason Nguyen	500	19995	.	.	100	1995	
Sharon Lu	.	.	.	.	.	.	1(

Adding Summary Breaks to Report Output

Two statements allow you to insert breaks:

- **BREAK** statement: adds a break for each unique value of the specified variable.

BREAK location variable / Option;

- **RBREAK** statement: does the same as BREAK for the entire report.

RBREAK location / Option;

Location has two possible values: **BEFORE** or **AFTER**

Option tells SAS the kind of break to insert.

- **Example:**

```
Proc report data=seltwo NOWINDOWS;
    COLUMN Name Region Quantity TotalSales;
    Define Name /Order;
    Break After Name / Summarize;
    Define Region / Order;
    Break After Region / Summarize;

    Title 'Report with Summary Breaks';
run;
```

Report with Summary Breaks

Name	Region	Quantity	TotalSales
George Smith	North	50	449.5
		3	15597
		3	15597
		1	5129
George Smith	North	57	36772.5
	South	100	599
		10	299.5
		1	9109
George Smith	South	111	10007.5
George Smith		168	46780
Glenda Johnson	East	100	899
		100	699