PSTAT 130

SAS BASE PROGRAMMING

- Lecture 3 -

Dr. Julie Swenson

Objectives

- SAS Formats
- User Defined Formats
- Titles and Footnotes
- System Options

Formats

- Variables vs. Observations
  - In a table view?
- Labels change the way variable names are displayed (as column headings).
- Formats change the way observations are displayed.

Format Examples

- Why might this be useful?
  - Need to recode/clarify variable values
    - Numeric variables:
      - Ex: MOVIE_TICKET

        0 = ‘child'

        1 = ‘adult'
      - Ex: DAY_OF_WEEK

        0 = 'Sunday'

        1 = 'Monday'

        …

        6 = 'Saturday'

Format Examples

- Numeric variables (continued):
  - Ex: CREDIT_SCORE

    781-850 = 'excellent'

    661-780 = 'good'

    501-600 = 'fair'

    0-500 = 'bad'

- Character variables:
  - Ex: STATE

    'AL' = 'Alabama'

    'AK' = 'Alaska'

    'AZ' = 'Arizona'

Formats

- Goal: to enhance the readability of reports
  - Possible solution: format the data values

Salary Report

EmpID	Last Name	First Name	JobCode	Annual Salary
0031	GOLDENBERG	DESIREE	PILOT	$50,221.62
0040	WILLIAMS	ARLENE M.	FLTAT	$23,666.12
0071	PERRY	ROBERT A.	FLTAT	$21,957.71
0082	MCGWIER-WATTS	CHRISTINA	PILOT	$96,387.39
0091	SCOTT	HARVEY F.	FLTAT	$32,278.40
0106	THACKER	DAVID S.	FLTAT	$24,161.14
0355	BELL	THOMAS B.	PILOT	$59,803.16
0366	GLENN	MARTHA S.	PILOT	$120,202.38

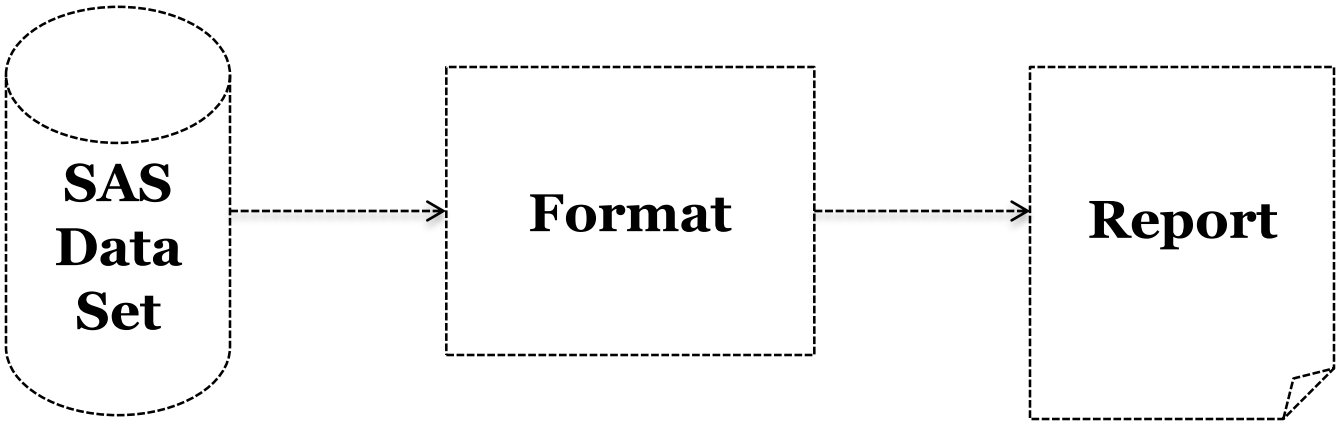
Custom Formats

- Create custom formats to recode data values in a report

Salary Report in Categories

Emp ID	Last Name	First Name	Job Code	Annual Salary
0031	GOLDENBERG	DESIREE	Pilot	More than 50,000
0040	WILLIAMS	ARLENE M.	Flight Attendant	Less than 25,000
0071	PERRY	ROBERT A.	Flight Attendant	Less than 25,000
0082	MCGWIER-WATTS	CHRISTINA	Pilot	More than 50,000
0091	SCOTT	HARVEY F.	Flight Attendant	25,000 to 50,000
0106	THACKER	DAVID S.	Flight Attendant	Less than 25,000
0355	BELL	THOMAS B.	Pilot	More than 50,000
0366	GLENN	MARTHA S.	Pilot	More than 50,000

Format Application



Format Statement

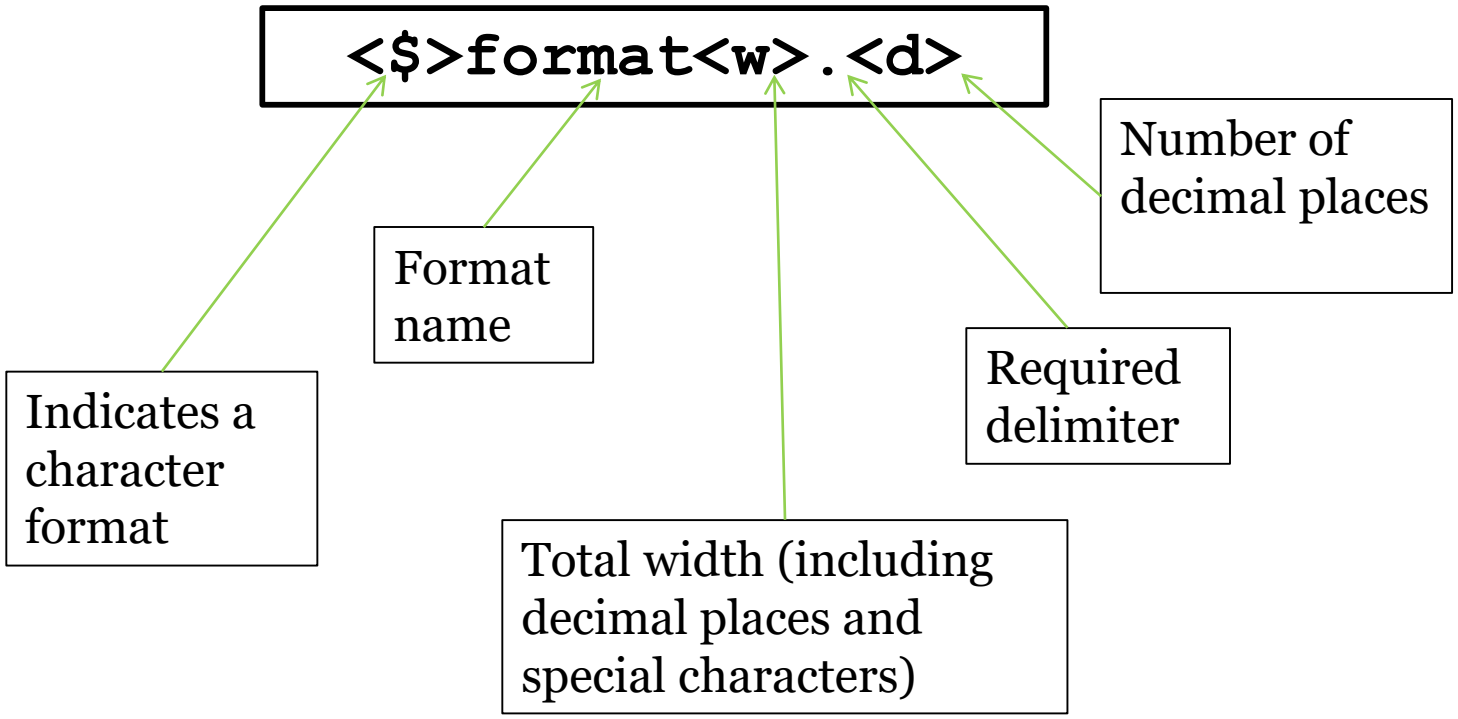
- General form of the FORMAT statement

```
FORMAT variable(s) format;
```

- Example

```
proc print data=data1.empdata;
     format Salary dollar11.2;
run;
```

Format Structure



Selected Formats

- Selected SAS Formats

Format/ Example	Result	Description
`w.d` `8.2`	`12234.21`	**standard numeric format** **Width=8, 2 decimal places**
`$w.` `$5.`	`KATHY`	**standard character format** **Width=5**
`COMMAw.d` `COMMA9.2`	`12,234.21`	**Commas in a number** **Width=9, 2 decimal places**
`DOLLARw.d` `DOLLAR10.2`	`$12,234.21`	**Dollar signs and commas in a number** **Width=10, 2 decimal places**

Numeric Formats

Stored Value	Format	Displayed Value
27134.2864	COMMA12.2	27,134.29
27134.2864	12.2	27134.29
27134.2864	DOLLAR12.2	$27,134.29
27134.2864	DOLLAR9.2	$27134.29
27134.2864	DOLLAR8.2	27134.29
27134.2864	DOLLAR5.2	27134
27134.2864	DOLLAR4.2	27E3

Numeric Format Example

EmpID	LastName	FirstName	JobCode	Salary
0031	GOLDENBERG	DESIREE	PILOT	50221.62
0040	WILLIAMS	ARLENE M.	FLTAT	23666.12
0071	PERRY	ROBERT A.	FLTAT	21957.71

data1.empdata

```
proc print data=data1.empdata label noobs;
        label LastName='Last Name'
              FirstName='First Name'
              Salary='Annual Salary';
        format salary dollar11.2;
run;
```

EmpID	Last Name	First Name	JobCode	Annual Salary
0031	GOLDENBERG	DESIREE	PILOT	$50,221.62
0040	WILLIAMS	ARLENE M.	FLTAT	$23,666.12
0071	PERRY	ROBERT A.	FLTAT	$21,957.71
0082	MCGWIER-WATTS	CHRISTINA	PILOT	$96,387.39
0091	SCOTT	HARVEY F.	FLTAT	$32,278.40
0106	THACKER	DAVID S	FLTAT	$24,161.14

How SAS Stores Date Values

- SAS stores dates with a numeric variable.

- The value of this variable represents the number of days since January 1st, 1960.
  - Example:
    - January 1st, 1960 → 0
    - January 2nd, 1960 → 1
    - December 31st, 1959 → -1
    - June 6th, 2020 → 22072

How SAS Stores Date Values

- Example:

```
data datetest;
input days;
date=days;
datalines;
0
01
59
365
366
22072
;
run;

proc print;
format date mmddyy10.;
run;
```

Obs	days	date
1	0	01/01/1960
2	1	01/02/1960
3	59	02/29/1960
4	365	12/31/1960
5	366	01/01/1961
6	22072	06/06/2020

Date Formats

MMDDYYw.

Format	Displayed Value
MMDDYY6.	101601
MMDDYY8.	10/16/01
MMDDYY10.	10/16/2001

DATEw.

Format	Displayed Value
DATE7.	16OCT01
DATE9.	16OCT2001

Date Formats

- Date formats and serial date values

Stored Value	Format	Displayed Value
0	MMDDYY8.	01/01/60
0	MMDDYY10.	01/01/1960
1	DATE9.	02JAN1960
-1	WORDDATE.	December 31, 1959
365	DDMMYY10.	31/12/1960
366	WEEKDATE.	Sunday, January 1, 1961

User-Defined Formats

- To create and use your own formats
  - Use the FORMAT procedure to create the format
  - Apply the format to specific variables by using a FORMAT statement

- General form of the PROC FORMAT step

```
PROC FORMAT;
       VALUE format-name    range1='display-value1'
                            range2='display-value2'
                            . . . ;
RUN;
```

User-Defined Formats

- Format-name
  - Name of the format you are creating (i.e. AgeFmt)
  - Character Formats
    - must have a dollar $ as the first character
    - a letter or underscore as the second character
    - letters, numbers, and underscores as additional characters
  - Numeric Formats
    - must have a letter or underscore as the first character
    - then letters, numbers, and underscores as additional characters

User-Defined Formats

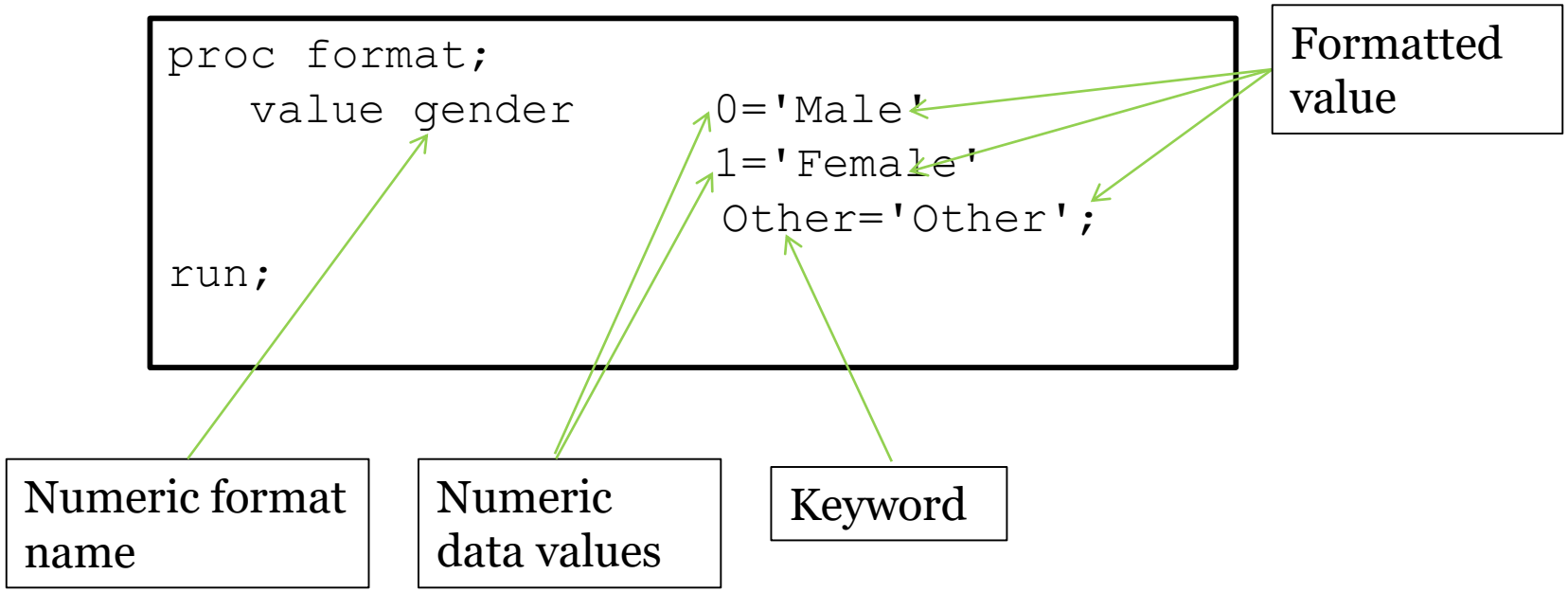
- Format-name (continued)
  - Cannot end in a number
  - Cannot be the same as an existing SAS format
  - Does not end with a period in the VALUE statement

User-Defined Formats

- Range(s)
  - Can be single values or
  - Ranges of values
- Display-value(s)
  - Can be up to 32,767 characters in length
  - Are typically enclosed in quotes, although it is not required

User-Defined Formats

- Assign display-values to single numbers



User-Defined Formats

- Assign display-values to a range of numbers

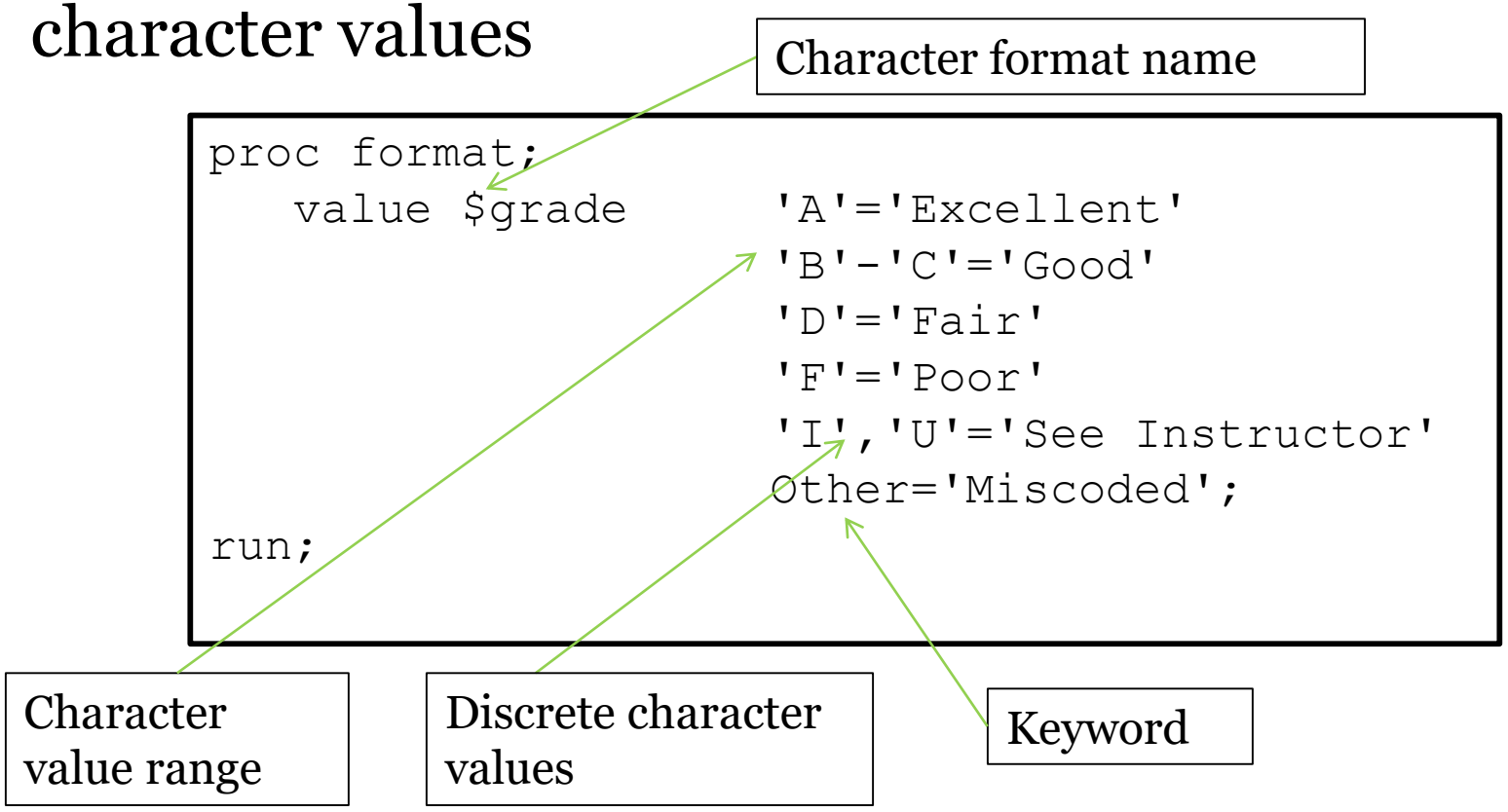
```
proc format;
   value boardfmt   low-49='Below Average'
                    50-99='Average'
                    100-high='Above Average';
run;
```

Numeric data values

Keywords

User-Defined Formats

- Assign display-values to characters and ranges of character values



User-Defined Formats

- Step 1: Create the format

```
proc format;
     value $codefmt 'FLTAT'='Flight Attendant'
                    'PILOT'='Pilot';
run;
```

- Step 2: Apply the format

```
proc print data=data1.empdata;
     format Jobcode $codefmt.;
run;
```

User-Defined Formats

- Step 1: Create the format

```
proc format;
     value money     low-<25000='Less than 25,000'
                     25000-50000='25,000 to 50,000'
                     50000-high='More than 50,000';
run;
```

- Step 2: Apply the format

```
proc print data=data1.empdata;
     format Salary money.;
run;
```

User-Defined Formats

- Define multiple formats within the same FORMAT procedure

```
proc format;
     value $codefmt 'FLTAT'='Flight Attendant'
                    'PILOT'='Pilot';

     value money    low-<25000='Less than 25,000'
                    25000-50000='25,000 to 50,000'
                    50000-high='More than 50,000';
run;
```

Applying User-Defined Formats

```
proc print data=data1.empdata split=' ';
   label LastName='Last Name'
         FirstName='First Name'
         Salary='Annual Salary';
   format Salary money. Jobcode $codefmt.;
run;
```

EmpID	Last Name	First Name	JobCode	Annual Salary
0031	GOLDENBERG	DESIREE	Pilot	More than 50,000
0040	WILLIAMS	ARLENE M.	Flight Attendant	Less than 25,000
0071	PERRY	ROBERT A.	Flight Attendant	Less than 25,000
0082	MCGWIER-WATTS	CHRISTINA	Pilot	More than 50,000
0091	SCOTT	HARVEY F.	Flight Attendant	25,000 to 50,000
0106	THACKER	DAVID S.	Flight Attendant	Less than 25,000
0355	BELL	THOMAS B.	Pilot	More than 50,000
0366	GLENN	MARTHA S.	Pilot	More than 50,000

Class Exercise 1

- Use the **`navigators`** data set in the data1 folder.
- Format the following variables:
  - STATE
    - Change the abbreviations to the full name
  - JOBCODE
    - Change the values NA1 and NA2 to ‘Navigator 1’ and ‘Navigator 2’, respectively
  - SALARY
    - low-9999 = ‘Below 10k’
    - 10000-50000 = ‘Between 10k and 50k’
    - 50001-high = ‘More than 50k’

Class Exercise 2

- Use the **`staff`** data set in the data1 folder.
- Create a format that assigns:
  - 'Salary' to the value S
  - 'Hourly' to the value H
- Create a format that assigns:
  - 'Low' to a value less than 800
  - 'Moderate' to a value between 800 and 1200
  - 'High' to a value greater than 1200

Class Exercise 2 - continued

- Create a report that displays Name, DOB, WageCategory, Bonus (in this order)
  - Suppress observation numbers
  - Use ‘Full Name’, ‘Birthdate’, ‘Wage Category’, and ‘Bonus Level’ as the column headers
  - Apply the appropriate formats to WageCategory and Bonus.
  - Apply a date format to DOB that displays dates in the form 06JUN2020.
- Now create the same report, but only select individuals whose bonus is greater than 1000.

Define Titles

- Features of titles
  - Titles appear at the top of the report
  - The default title in SAS 9.4 is 'The SAS System'
    - There is no default title in SAS Studio
  - You can have more than one title line
  - The value of n ranges from 1 to 10 and refers to the title line
  - A unnumbered TITLE is equivalent to TITLE1
  - Titles remain in effect until they are changed
  - The null TITLE statement (`title;`) cancels all titles

Define Footnotes

- Features of footnotes
  - Footnotes appear at the bottom of the output page
  - There is no default footnote
  - You can have more than one footnote line
  - The value of n ranges from 1 to 10 and refers to the footnote line
  - Footnotes remain in effect until they are changed
  - The null FOOTNOTE statement (`footnote;`) cancels all footnotes

Define Titles and Footnotes

- General form of the TITLE statement

```
TITLEn 'text';
```

- General form of the FOOTNOTE statement

```
FOOTNOTEn 'text';
```

- Examples

```
title1 'PSTAT 130 Students';
footnote2 'Summer 2020';
```

Change Titles and Footnotes

- **`TITLEn or FOOTNOTEn`**
  - Replaces a previous title or footnote with the same number
  - Cancels all titles or footnotes with higher numbers

Title and Footnote Options

- Color= / C=
  - Set the color (i.e. red, green, blue)
- Height= / H=
  - Set the size of the text (can also specify unit of measurement)
- Font= / F=
  - Set the system font

Title and Footnote Options

- Bold
- Italic
- Justify = Left | Center | Right
- Note: These options are not applied to the output directed to listing reports. They are applied to output directed to the Results window.

Title Options Example

```
title1 color=red height=4 bold justify=left
   'Do not go gentle into that good night';
title2 color=green height=3 font='Helvetica' italic
   'Old age should burn and rave at close of day';
title3 color=blue height=4 justify=right
   'Rage, rage against the dying of the light';
```

Do not go gentle into that good night

Old age should burn and rave at close of day

Rage, rage against the dying of the light

Further Enhancements

- How do we further enhance our reports?
  - SAS system options
- Think of a MS Word document
  - Some basic options
    - Displaying dates
    - Displaying page numbers
    - Changing page width

System Options

- General form of the OPTIONS statement

```
options <option1> <option2> <...>;
```

- See a full list of available options

```
proc options;
run;
```

  - Options displayed in the log file
  - Includes a brief description of each option

System Options

- Select options:
  - date/nodate
    - Displays the date / Suppresses the date
  - number/nonumber
    - Displays page numbers / Suppresses page numbers
  - pageno=
    - Sets the starting page number for future output
  - pagesize= (ps=)
    - Sets the number of lines in a page (range: 15 – 32,767)

System Options

- Select options (continued)
  - linesize= (ls=)
    - Sets the line size in characters (range: 64-256)
  - firstobs=
    - Sets the first observation to be read
  - obs=
    - Sets the last observation to be read
  - Note: When firstobs= and obs= are used together, the number of observations that are read
    - (obs – firstobs)+1

System Options

- Properties:
  - Global
    - Can be inside or outside of DATA or PROC steps
  - Stay in effect until changed
  - Most apply to output directed to a listing report.
- Example:
  - `options nodate pageno=10 ls=80 ps=25;`
  - `options nodate nonumber firstobs=5 obs=30;`

How to Create a List Report

- SAS Studio:
  - Place the following code before your procedure(s)
    - ODS LISTING file='/home/user/list_report_name.lst';
  - Place the following code after the procedure(s)
    - ODS LISTING close;
  - Example:

```
ods listing file='/home/user/list_report_name.lst';
proc print data=data1.empdata;
run;
proc print data=data1.staff;
run;
ods listing close;
```