Introduction to SAS

Beate Henschel
Biostatistics Consulting Center
School of Public Health - Bloomington

INDIANA UNIVERSITY BLOOMINGTON

Introduction

- Biostatistics Consulting Center, School of Public Health

- This workshop is part of the SWiSS on SPSS, SAS and R software
  - SPSS – easy “point & click”, good for most “off the shelf” analyses
  - SAS – syntax based, industry standard, public health
  - R – syntax based, free & flexible
  - STATA – syntax w/ “point & click”, political science, sociology, economics
  - JMP – “point & click”, good mix of stats and graphs – good for exploring data
  - MATLAB – powerful numerical computing, matrix manipulations

Overview

1. Introduction to SAS
2. SAS Environment
3. Getting data into SAS
4. DATA step
5. PROC step
6. Practice in SAS

SECTION 1

Introduction to SAS

SAS – Statistical Analysis System

- Pronounced “sass”, never “Ess Aye Ess”
- Software package comprised of many different modules
- BASE SAS is centerpiece of all SAS software within SAS Foundation
  - consists of procedures and functions for data manipulation and basic analyses (correlation, frequencies, univariate statistics)
- SAS/STAT module includes more specialized and complex procedures like regressions, ANOVAs and mixed models
- SAS/GRAPH module includes procedures for data visualization

SAS – A little history

- Need for a computerized statistics program to analyze agricultural data collected through USDA grants
- Consortium of eight land-grant universities developed a general-purpose statistical software package funded by NIH in 1966
  - Funding from NIH discontinued in 1972
- With growing demand for statistical software SAS Institute Inc. was founded in 1976 to help customers in all sorts of industries
- One of fastest growing companies in US in 1980s
- 25th anniversary in 2001: new logo and new tagline

SAS – A little history (cont.)

Today:

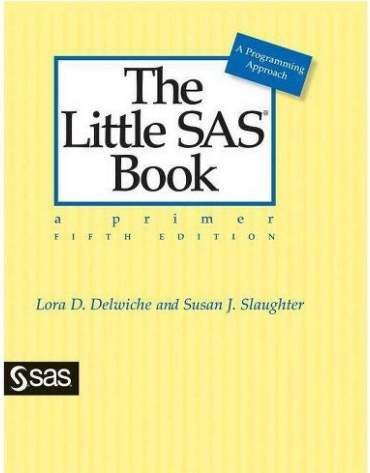
- Cofounders Jim Goodnight, CEO, and John Sall, Exec. Vice President, still own SAS as private company
- SAS world headquarters in Cary, NC, on 300-acre campus
- SAS software installed at >80,000 business, government and university sites
- Current version: SAS 9.4 m6

SAS at IU

- Most Windows computers in Student Technology Centers have SAS installed
- On your own computer
  - Buy a 1-year license from RADL (Research Applications & Deep Learning) for $125
  - On IUanyWare http://iuanyware.iu.edu
  - On Research Desktop (RED) (Carbonate, one of IU's supercomputer, Linux version) https://red.uits.iu.edu

Book recommendation

Delwiche and Slaughter: The Little SAS Book: A primer (5th Edition), SAS Institute, 2012



SECTION 2

SAS Environment

SAS Windows

Can arrange window placement

Editor window:

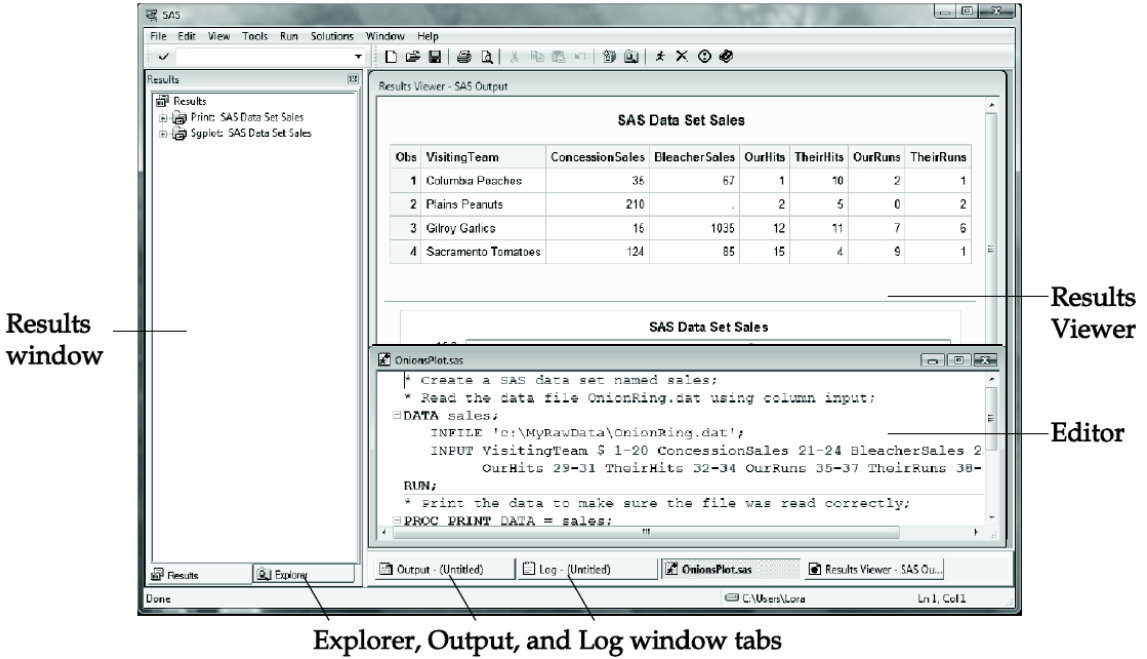
- Text editor to write, edit, and submit SAS programs, syntax sensitive and color codes the programs

Results viewer window:

- If any results are created, then this window displays the results

Results window:

- Like a table of contents for output from Results Viewer



SAS Windows

Explorer window:

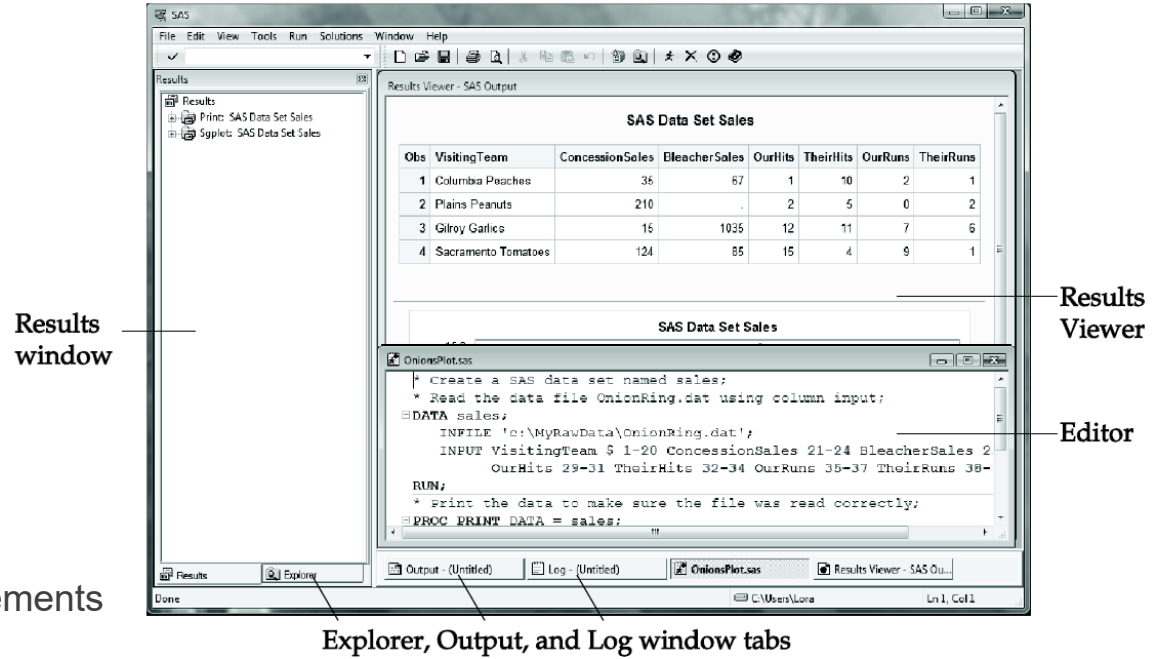
- Lists SAS files and libraries

Output window:

- Results will show here if html output is disabled

Log window:

- Contains notes about current SAS session
- After submitting program: any notes, errors, warnings and all statements



SAS Data Sets

- Before any analysis/reports: need to read in data into SAS
- SAS data sets have file extension: .sas7bdat
- Data in SAS in form of tables
  - Columns: variables
  - Rows: observations
- Two data types:
  - Numeric: numbers, can also contain + - . E
  - Character: may contain numerals, letters or special characters, up to 32,767 characters long

Variables (Also Called Columns)

Observations (Also Called Rows)	Id	Name	Height	Weight
1	53	Susie	42	41
2	54	Charlie	46	55
3	55	Calvin	40	35
4	56	Lucy	46	52
5	57	Dennis	44	.
6	58		43	50

SAS Data Sets

Missing data:

- Numeric variable: . (single period)
- Character variable: blanks

Variable names:

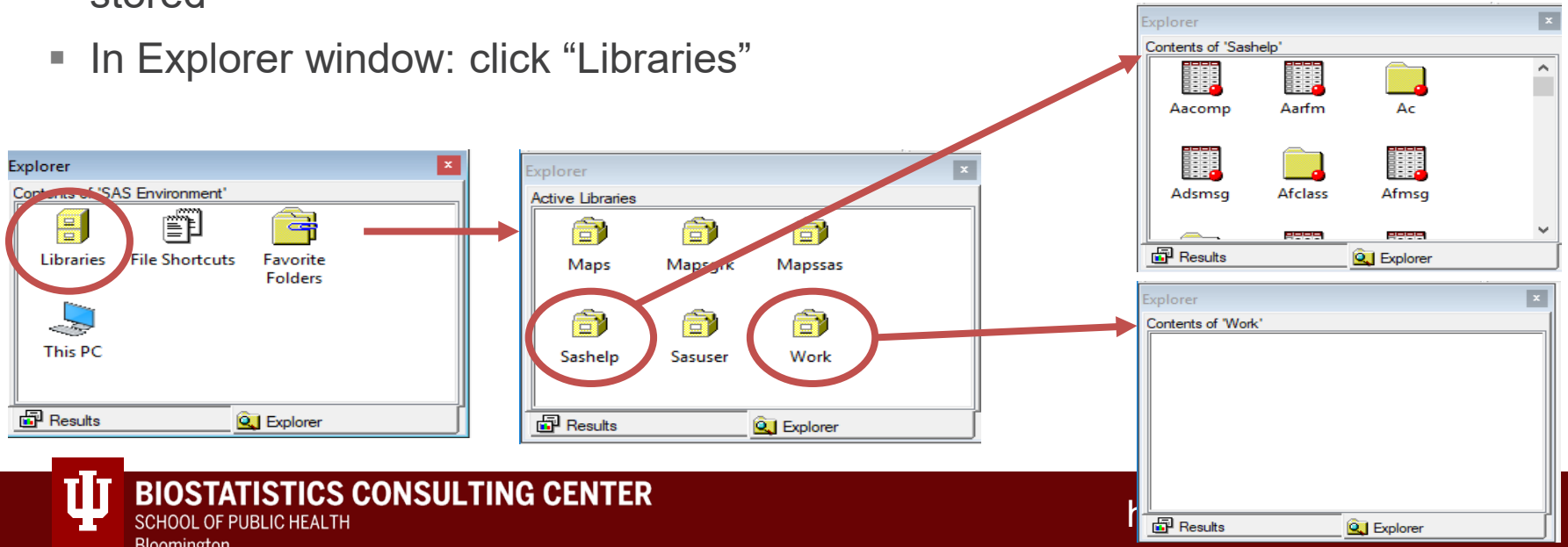
- Length: 32 characters or fewer
- Start with letter or underscore (_)
- Contain only letters, numerals, or underscores (_)
- No special characters

Variables (Also Called Columns)

Observations (Also Called Rows)	Id	Name	Height	Weight
1	53	Susie	42	41
2	54	Charlie	46	55
3	55	Calvin	40	35
4	56	Lucy	46	52
5	57	Dennis	44	.
6	58		43	50

SAS Data Libraries

- SAS library is location where SAS data sets (and other SAS files) are stored
- In Explorer window: click “Libraries”



SAS Data Libraries

- Work library:
  - Temporary storage location
  - Default library
- LIBNAME statement or New Library window
  - Right-click in Libraries window → select "New…" → type name of new library (called libref)
  - 8 characters or less, start with letter or underscore, contains only letters, numbers or underscores

If you create a SAS data set without specifying a library, SAS will put it in the WORK library, and then delete it when you end your session.

```
libname lib "\\Client\C$\Users\Beate\SASclass";
```

SAS Language: Programs and Statements

SAS programs:

- Sequence of statements executed in order
- DATA steps and PROC steps

SAS statements:

- **Every SAS statement ends with a semicolon.**
- SAS statements can be in upper- or lowercase.
- Statements can continue on the next line (but don't split words).
- Statements can be on the same line as other statements.

```
Global statement;
DATA step;
   Statement1;
   Statement2;
RUN;
/* Comment */
PROC step;
   Statement1;
   Statement2;
   Statement3;
RUN;
```

SAS Language: Comments

- Purpose: Annotation of programs

```
* Read animals' weights from file;
DATA animals;
   INFILE 'c:\MyRawData\Zoo.dat';
   INPUT Lions Tigers;
PROC PRINT DATA = animals;  /* Print the results */
RUN;
```

- Two styles:
  - Slash-asterisk style
  - Asterisk-semicolon style
  - Both styles equivalent

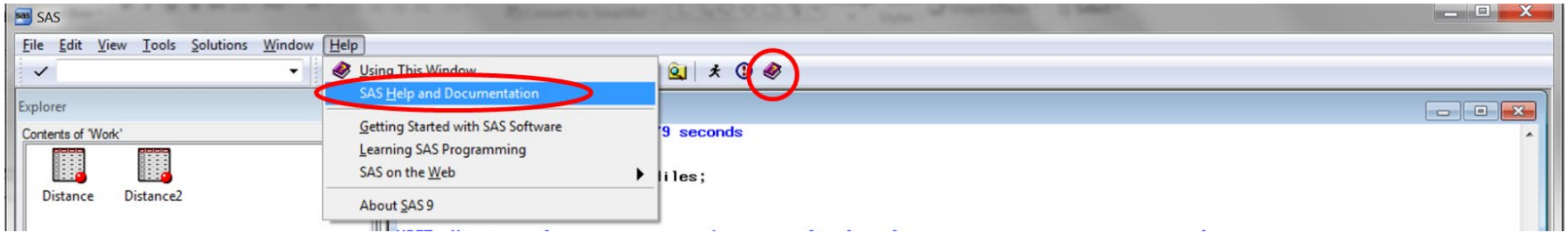
```
/* this is a comment */

* this is also a comment ;

/* you can put anything in a comment
data ; run ; proc ;
SAS will ignore it */
```

SAS Help and Documentation

- Separate window opens with the SAS Help and Documentation
- Same document can be found online as well



SECTION 3

Getting data into SAS

Getting data into SAS

- Enter data directly into SAS
  - Table Editor
  - DATA step with Datalines statements
- Use syntax
  - Read SAS data files directly into SAS: LIBNAME statement and DATA step
  - Convert data file into SAS dataset:
    PROC IMPORT or DATA step with INFILE and INPUT statements
- Import Wizard
  - point and click interface, can save the generated PROC IMPORT code

Importing using Syntax

- Importing using Syntax
  - PROC IMPORT

```
PROC IMPORT OUT= WORK.fliesin
     DATAFILE= "&mypath.\Fly_Numbers.xlsx"
          DBMS=xlsx REPLACE;
     SHEET="data";
     GETNAMES=yes;
RUN;
```

  - DATA step with INFILE and INPUT statements

```
data example3 ;
    infile 'I:\Q400 Documents\Week 5\sites.csv'
       dlm = ',' dsd firstobs = 2 ;
    input ID $ state $ spec $ fee date $ ;
run ;
```

  - DATA step with Datalines

```
DATA work.Example;
     input Name $ Role $ Sizeft sizein;
     datalines;
     Bendu Guard 5 10
     Aleksa Forward 6 3
     Linsey Center 6 2
     Ali Guard 5 11
     Brenna forward 6 0
     Teri coach . .
     Rhet assistant . .
     ;
RUN;
```

SECTION 4

DATA step

DATA step

- Create, read in, and manipulate datasets, create and manipulate variables
- No output in results viewer is produced
- Building blocks:

```
DATA dataset-name;          ← output dataset
   SET dataset-name;        ← input dataset, (optional)
   statement1;
   statement2;
RUN;
```

Statements to manipulate datasets

- Subsetting dataset: `WHERE condition;`
  - Example: `WHERE gender="Male";` or: `WHERE age > 65;`
  - Alternative: subsetting if: `IF gender="Male";`
- Selecting variables to keep: `KEEP variable-list;`
  - Example: `KEEP id gender weight height;`
- Selecting variables to delete: `DROP variable-list;`
  - Example: `DROP education birthdate;`

Statements to create/manipulate variables

- Assignment statement: `variable-name = expression;`
  - Assign numeric or character constant, assign another variable, add, subtract, multiply, divide, exponentiation
  - Example: `BMI = weight / (height**2);`
  - Note: can easily overwrite existing variables because SAS replaces existing variable values with new values
- Functions
  - Date: birthday and today function `birthday = MDY(month, day, year);`
  - Numeric: sum and mean function `Fruit = sum(apples, bananas, oranges);`

Statements to create/manipulate variables

- IF/THEN statement: `IF condition THEN expression1;`
- Example:

```
IF gender="M" THEN male = 1;
```
→ new variable male will either be 1 or .

```
IF gender="M" THEN male = 1;
IF gender="F" THEN male = 0;
```
→ new variable male will either be 1 or 0

- IF/THEN/ELSE statement:

```
IF condition THEN expression1; ELSE expression2;
```

```
IF gender="M" THEN male = 1; ELSE male=0;
```
→ new variable male will either be 1 or 0

SECTION 5

PROC step

PROC step

- Used to perform analyses on data
- Most procedures produce output in the results viewer
- Some procedures produce a new dataset, but they **do not** change the values of the input dataset
- Each procedure is unique with own syntax and set of options
- Building blocks:

```
PROC procname data=dataset-name;
   statement1 / statement1-options;
   statement2 / statement2-options;
RUN;
```

PROC step - examples

- Frequently used procedures to explore/describe data:
  - PRINT → prints the data
  - CONTENTS → prints list of variables
  - SORT → sorts the data
  - MEANS → creates summary statistics
  - UNIVARIATE → creates summary statistics
  - FREQ → creates frequency tables
  - SGPLOT → creates single plots
  - SGPANEL → creates panel of multiple plots

PROC step – examples (2)

- Frequently used procedures to analyze data:
  - FREQ with CHISQ option → Chi-square test for 2 categorical variables
  - TTEST → compare means for 2 groups
  - ANOVA → compare means for 3+ groups
  - CORR → correlation for continuous variables
  - REG → linear regression for continuous variables
  - GLM → fits general linear model (regression, ANOVA, ANCOVA,…)
  - LOGISTIC → logistic regression for binary variables

SORT and CONTENTS

- PROC SORT sorts the dataset by a single or set of variables

```
PROC SORT data=contest out=contest_sorted;
  BY NumberBooks;
RUN;
```

→ out= is optional

→ add `descending` before variable for descending order

- PROC CONTENTS displays description of dataset incl. a table with all variables

```
PROC CONTENTS data=contest varnum;
RUN;
```

→ varnum is optional, lists variables in order they appear in dataset instead of alphabetical order

MEANS and UNIVARIATE

- PROC MEANS produces the mean, number of observations, standard deviation, minimum, maximum by default

```
PROC MEANS data = dataset <statistic-keyword(s)> ;
  VAR variable(s);
RUN;
```

- PROC UNIVARIATE produces basic statistics and graphs describing the distribution
  - Mean, median, mode, standard deviation, skewness, kurtosis; quantiles; extreme observations
  - CDF plot, histogram, probability plot, qq plot

```
PROC UNIVARIATE data = dataset <options> ;
  VAR variable(s);
  HISTOGRAM variable;
RUN;
```

FREQ

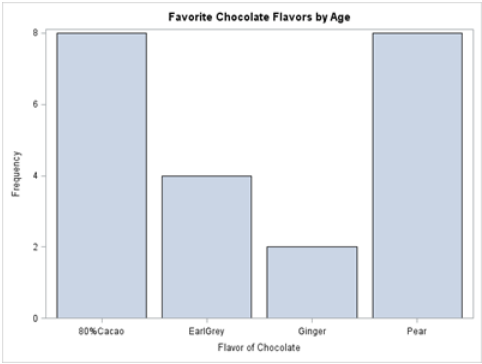
- PROC FREQ creates tables showing the distribution of categorical data values
  - Can help identify data entry errors or coding errors
  - Options include missing, nopercent, nocol, norow

```
PROC FREQ data = dataset;
  TABLE variable(s) / <options>;
RUN;
```

SGPLOT – visualizing data

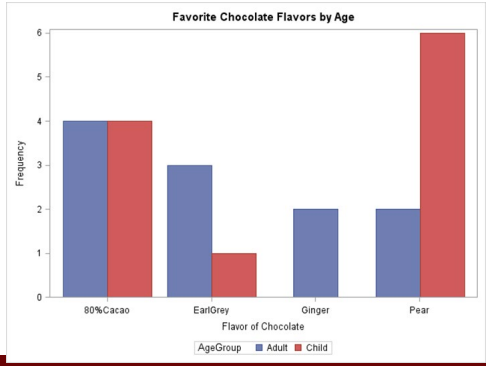
- Bar charts: distribution of categorical data

```
PROC SGPLOT data=chocolate;
  VBAR FavoriteFlavor ;
  TITLE 'Favorite Chocolate Flavors by Age';
RUN;
```



- Can add second variable to group data

```
PROC SGPLOT data=chocolate;
  VBAR FavoriteFlavor / GROUP=AgeGroup
                        GROUPDISPLAY=CLUSTER;
  TITLE 'Favorite Chocolate Flavors by Age';
RUN;
```



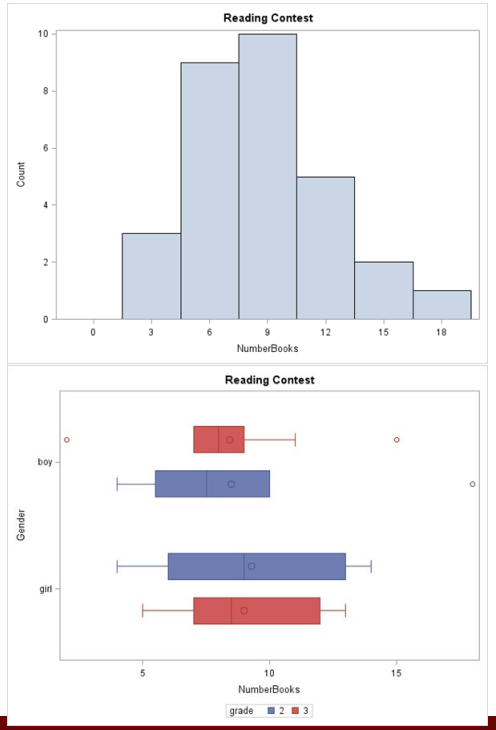
SGPLOT – visualizing data

- Histogram: distribution of continuous data

```
PROC SGPLOT DATA = contest;
  HISTOGRAM NumberBooks / BINWIDTH = 3
                          SHOWBINS SCALE = COUNT;
  TITLE 'Reading Contest';
RUN;
```

- Boxplot: distribution of continuous data

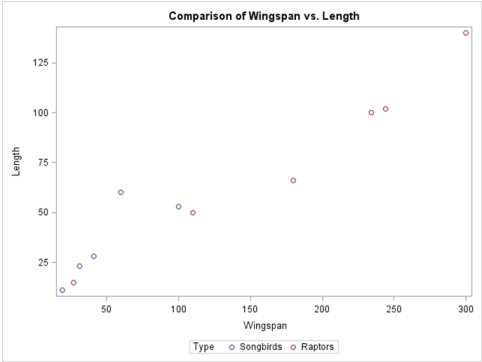
```
PROC SGPLOT DATA = contest;
  VBOX NumberBooks / CATEGORY=gender GROUP=grade;
RUN;
```



SGPLOT – visualizing data

- Scatterplot: relationship between two continuous variables
  - Independent variable on x-axis
  - Dependent variable on y-axis

```
PROC SGPLOT DATA = wings;
  SCATTER X = Wingspan Y = Length / GROUP = Type;
  TITLE 'Comparison of Wingspan vs. Length';
RUN;
```



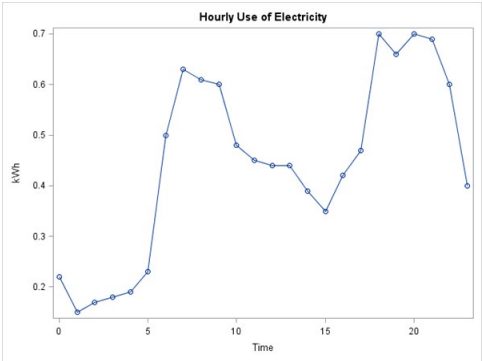
SGPLOT – visualizing data

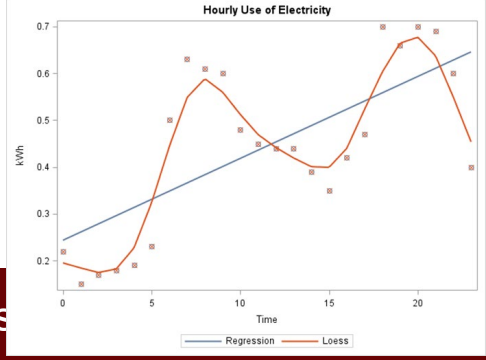
- Series plot: for time series, related to scatter plot but connecting points
  - Can have several SERIES statements (X needs to be the same)
  - Optionally add GROUP=variable-name to get separate lines by values of a categorical variable

```
PROC SGPLOT DATA = electricity;
  SERIES X = Time Y = kWh / MARKERS;
  TITLE 'Hourly Use of Electricity';
RUN;
```

- Can add regression or smoothed lines

```
PROC SGPLOT DATA = electricity;
  REG X=Time Y=kWh ;
  LOESS X=Time Y=kWh ;
RUN;
```



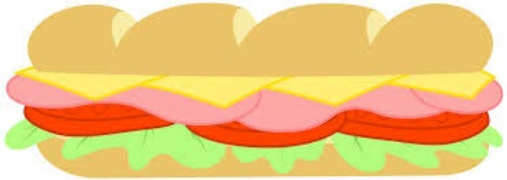


SGPLOT – visualizing data

- Improve graphs
  - Add labels to axes
  - Add reference line
  - Change legend
  - Change line/marker color, thickness, type
  - Add title
  - Etc…

ODS – Output delivery system

- To save output, need to add ODS destination statement and the ODS CLOSE statement
- ODS Sandwich!



→ ODS destination statement

Procedure

→ ODS CLOSE statement

- Example:

```
ODS rtf file='rtf-filename.rtf' ;
ODS pdf file='pdf-filename.pdf' ;
ODS excel file='C:\Documents\Example\excel-filename.xlsx';

<<< procedure/s >>>

ODS rtf CLOSE; ODS pdf CLOSE; ODS EXCEL CLOSE;
ODS _all_ CLOSE ;
```

← this closes all files that were created

PROC step – examples (2)

- Frequently used procedures to analyze data:
  - FREQ with CHISQ option → Chi-square test for 2 categorical variables
  - TTEST → compare means for 2 groups
  - ANOVA → compare means for 3+ groups
  - CORR → correlation for continuous variables
  - REG → linear regression for continuous variables
  - GLM → fits general linear model (regression, ANOVA, ANCOVA,…)
  - LOGISTIC → logistic regression for binary variables

FREQ with CHISQ option

- Chi-square test between two variables using the CHISQ option in PROC FREQ
  - Must have at least one variable combination in your TABLE statement
  - Can also conduct Cochran-Mantel-Haenszel statistics, Fisher's exact test, and relative risk
    - CMH, FISHER, RELRISK

```
PROC FREQ data = dataset;
  TABLE variable1*variable2 / CHISQ;
RUN;
```

TTEST

- Can compute one sample, independent two-sample, or paired t-tests
- Can also produce boxplot, histogram, qqplot, etc. with PLOTS = option

```
PROC TTEST data=dataset H0=n <options>;
  VAR variable(s);
RUN;
```

```
PROC TTEST data=dataset <options>;
  VAR variable ;
  CLASS variable ;
RUN;
```

```
PROC TTEST data=dataset <options>;
  PAIRED variable1*variable2;
RUN;
```

ANOVA

- Compares means between 3+ groups, Analysis of Variance, best for balanced data
  - CLASS statement contains the categorical effect variable and must come before the MODEL statement
  - MODEL statement defines the dependent variable and the effect variable
  - Optional MEANS statement for post-hoc comparisons

```
PROC ANOVA data = dataset;
  CLASS effect-variable;
  MODEL dependent-variable = effect-variable;
RUN;
```

CORR

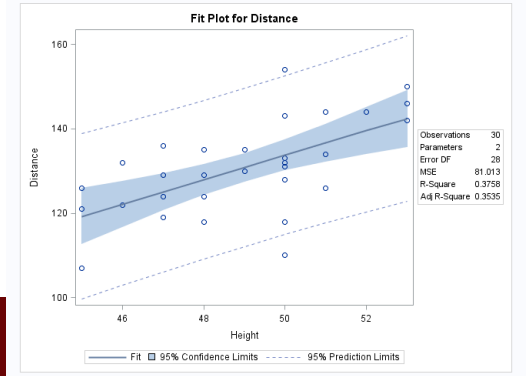
- Computes correlations (measures the strength of the linear relationship between two variables, can take values between -1 and 1)
- VAR statement includes variables that will appear across the top of the table
- Optional WITH statement includes variables that will appear on the side of the table, (if omitted, VAR statement variables will appear there)
- Optional plots to visualize linear relationship between variables (Ex: Matrix – scatter plot matrix for all variables)

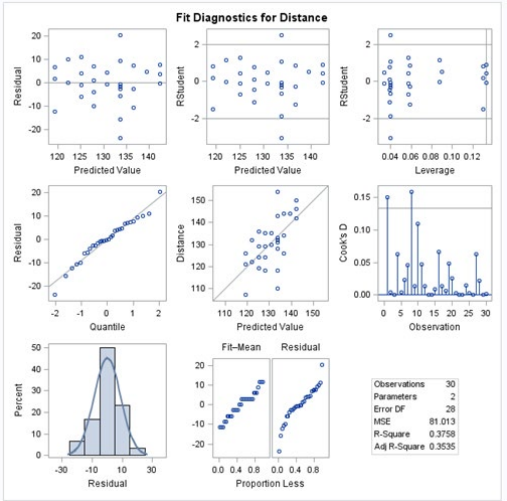
```
PROC CORR data = dataset PLOTS=(plot-list) ;
  VAR variable-list;
  WITH variable-list;
RUN;
```

REG

- Fits linear regression models by least squares method
  - Can only use continuous or dichotomous predictors, if you have a multi-category predictor, you will have to create dummy variables or use another procedure

```
PROC REG data = dataset;
  MODEL dependent-var = independent-variables;
RUN;
```





GLM

- Fits general linear regression models by least squares method
  - Simple/multiple regression, analysis of variance (unbalanced data), analysis of covariance, repeated measures ANOVA, and partial correlation

```
PROC GLM data = dataset;
  CLASS variable-list;
  MODEL dependent-var = independent-variables;
  LSMEANS variable;
RUN;
```

LOGISTIC

- Used when dependent variable has two levels or categories (if more than 2 levels: multinomial)
  - Models the probability that the dependent-variable is the lowest category (either numeric or in alphabetical order) → Can change with the descending option

```
PROC LOGISTIC data = dataset;
  CLASS variable-list;
  MODEL dependent-var = independent-variables;
RUN;
```

SECTION 6

PRACTICE

Practice in SAS

- Open SAS
- Download and open the SAS syntax file from https://go.iu.edu/2bZY

INDIANA UNIVERSITY BLOOMINGTON