Lecture Notes
Intro to SAS

Christopher Kinson

Department of Statistics
University of Illinois at Urbana-Champaign

Is **R** Your Preferred Statistics Software?

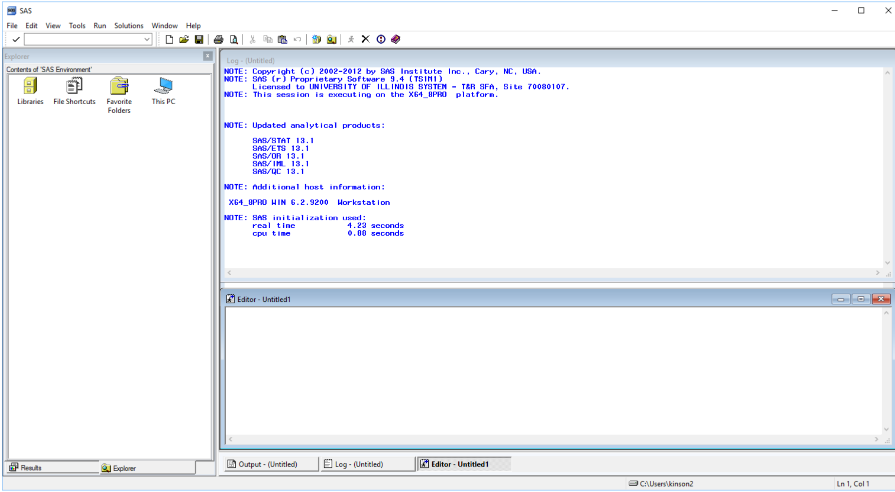
Raise your hand

Is **Python** Your Preferred Statistics Software?

Raise your hand

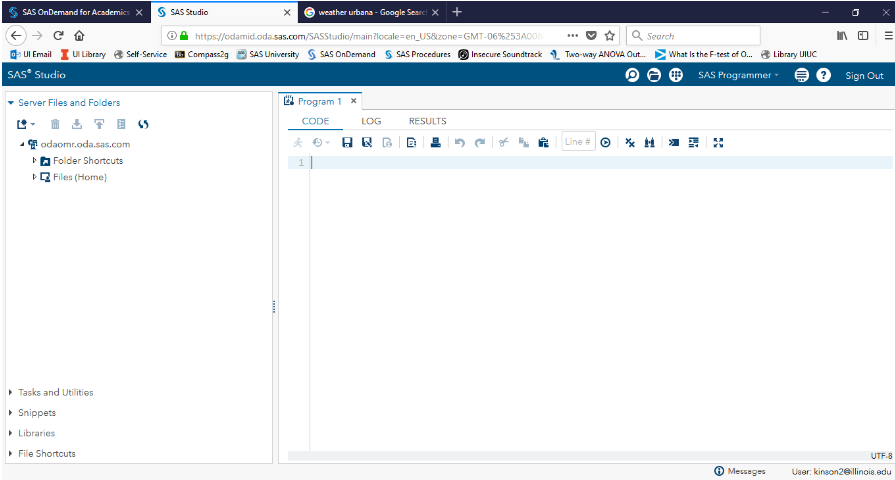
Now for a more formal introduction to SAS!

What Does Base SAS 9.4 Look Like?

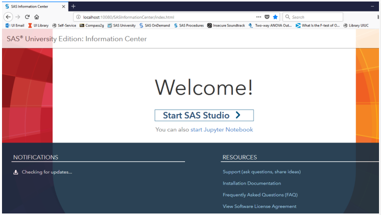


```
NOTE: Copyright (c) 2002-2012 by SAS Institute Inc., Cary, NC, USA.
NOTE: SAS (r) Proprietary Software 9.4 (TS1M1)
      Licensed to UNIVERSITY OF ILLINOIS SYSTEM - T&R SFA, Site 70080107.
NOTE: This session is executing on the X64_8PRO  platform.



NOTE: Updated analytical products:

      SAS/STAT 13.1
      SAS/ETS 13.1
      SAS/OR 13.1
      SAS/IML 13.1
      SAS/QC 13.1

NOTE: Additional host information:

 X64_8PRO WIN 6.2.9200  Workstation

NOTE: SAS initialization used:
      real time           4.23 seconds
      cpu time            0.88 seconds
```

What Does SAS OnDemand Look Like? (cont.)



What Does SAS University Look Like? (cont.)



SAS® University Edition: Information Center

Welcome!

You can also start Jupyter Notebook

NOTIFICATIONS

Checking for updates...

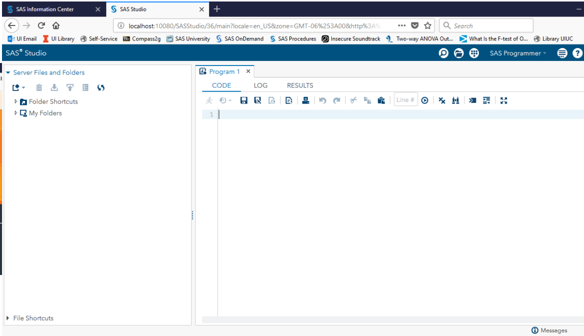
RESOURCES

Support (ask questions, share ideas)

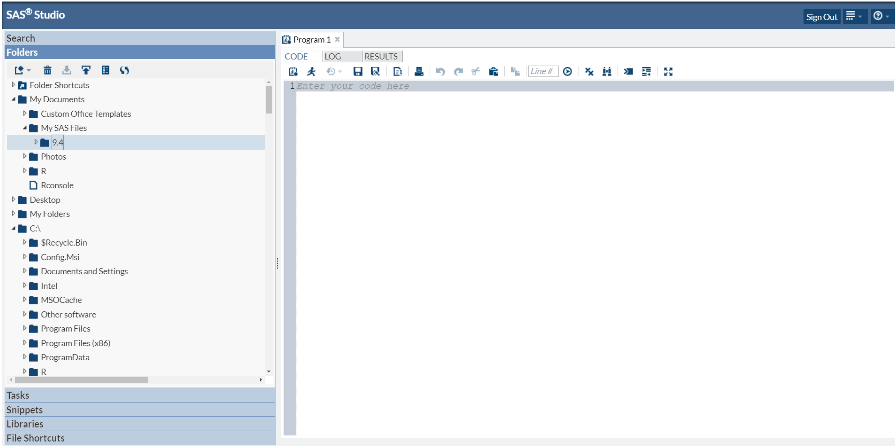
Installation Documentation

Frequently Asked Questions (FAQ)

View Software License Agreement



What Does SAS Studio 3.1 Look Like? (cont.)



The Interface

- Editor Window - you write your code here
- Log Window - useful for debugging your code
- Results Window - results show up here
- Explorer Window - where you can search the directory for files

Steps of a Program

Creating a SAS data set

- Start with a **data** step
- Give names to data sets and variables in the data set
- Submit the program via the "Run" tab or tool bar icons to run the program and generate results

Analyzing a SAS data set

- Use a **proc** step (assuming the data is already defined)
- Give names to data sets and variables in the data set
- Submit the program via the "Run" tab or tool bar icons to run the program and generate results

All SAS statements must end with a semicolon ;

Let’s begin with the Data step

The Data Step

Includes

- a name for the data set
- names and types for the input variables
- a data source
- and any additional processing/cleaning code
    - subsetting, adding variables, merging, etc.

Data Sources

Data can be

- entered manually
- in a data file that is read in with an **infile** statement
  - using list input
  - using column input
  - using formatted inputs called informats
- an existing SAS data set
  - with a **set** statement
  - SAShelp data sets

Data Step Examples

Most data steps will be provided for you. Here are some examples.

```
*body fat data set;
data bodyfatCol;
 infile 'C:/STAT_448/bodyfat0.txt' ;
 input age 1-2 pctfat 4-7 sex $9;
run;

/*Fisher's famous iris data set which
is a part of the SAS library*/
data iris;
 set sashelp.iris;
run;
```

Data Step Examples (cont.)

```
*Reading a comma-separated values file;
data crime07;
 infile ’C:/STAT_448/crime_rate_2007.csv’ dlm=’,’
   missover firstobs=2;
 input state $ violentcrime murder rape robbery
   assault propertycrime burglary larceny cartheft ;
 drop violentcrime propertycrime;
run;
```

Data Step Examples (cont.)

```
*A manually entered data set;
data Hypnosis;
   length Emotion $ 10;
   input Subject Emotion $ SkinResponse @@;
   datalines;
1 fear 23.1  1 joy 22.7  1 sadness 22.5  1 calmness 22.6
2 fear 57.6  2 joy 53.2  2 sadness 53.7  2 calmness 53.1
3 fear 10.5  3 joy  9.7  3 sadness 10.8  3 calmness  8.3
4 fear 23.6  4 joy 19.6  4 sadness 21.1  4 calmness 21.6
5 fear 11.9  5 joy 13.8  5 sadness 13.7  5 calmness 13.3
6 fear 54.6  6 joy 47.1  6 sadness 39.2  6 calmness 37.0
7 fear 21.0  7 joy 13.6  7 sadness 13.7  7 calmness 14.8
8 fear 20.3  8 joy 23.6  8 sadness 16.3  8 calmness 14.8
;
```

Documenting Your Code: Syntax Errors

Syntax errors are easy to spot with color coding in the editor. But do keep these things in mind.

- File paths and character strings using quotation marks will appear in purple color in the editor

  ```
  ''Quotation marks.They'll match except for
  contractions.'';
  ```

- Valid SAS statements will be blue in the editor
- Comments will be green in the editor

Documenting Your Code: Using Comments

Comments are notes or reminders that are not intended to be run or shown in the results. Often, programmers want to leave themselves notes or instructions in the code.

```
* this is a an example of a line comment;

/* this is a block comment
because it can be on multiple
lines */
```

Any Questions?

Let’s Discuss the Proc Step

The Proc Step

Typically includes

- name of the procedure used (type of analysis)
- name of the data set to be analyzed (assuming it is already defined)
- options based on the procedure
- statement(s) specifying the names variables in the data set and how they will be analyzed (depends on the procedure)
- Ends with a run statement

The Proc Step - A Block of Statements

Can have pretty basic syntax

```
proc procName data=dataName;
run;
```

but if needed . . .

Can have pretty complicated syntax

```
proc procName data=dataName procOptions;
 statementName1  variableNames / statement1Options;
 statementName2  variableNames / statement2Options;
 ...
run;
```

Some Basic Proc Statements

- **proc sgplot**
  - can produce a range of statistical graphics
- **proc print**
  - prints the data
- **proc contents**
  - displays the data attributes in a summary without printing the data
- **proc sort**
  - sorts the data after specifying which variables will have a certain type of ordering

Using **proc sgplot**

To produce certain plots and graphs in **proc sgplot**, we may use

- scatter plot

```
proc sgplot data=dataName;
 scatter y=vertVarName x=horiVarName / group=groupVarName;
run;
```

- bar graph

```
*vertical bar graph;
proc sgplot data=dataName;
  vbar varName / group=groupVarName groupdisplay=cluster;
run;
*horizontal bar graph;
proc sgplot data=dataName;
  hbar varName / group=groupVarName groupdisplay=cluster;
run;
```

Using **proc sgplot** (cont.)

- box plot

```
proc sgplot data=dataName;
 title "A Horizontal Box Plot";
 hbox variableName / category=groupVarName;
run;
proc sgplot data=dataName;
 title "A Vertical Box Plot";
 vbox variableName / category=groupVarName;
run;
```

- regression plot - regression line added to scatter plot
  - `nomarkers` is used to prevent double layer of plotted points

```
proc sgplot data=dataName;
 reg y=vertVarName x=horiVarName / nomarkers;
run;
```

Basic Statements for Any Proc Statements

- **var** - variables to be used in the procedure
- **class** - classification variables
  - the variables must be character strings or discrete values
- **where** - for subsetting the data set
  - followed by a logical condition and only the true observations are included in the analysis
- **by** - for organizing results based on a specific variable
  - observations are grouped according to the values in the by statement and a separate analysis is done for each group
  - but first, the data set must be sorted using the by variable

Using the **where** Statement

- When using a **where** statement, the variables must already be in the input dataset
- Within a proc step, a **where** statement should be used only once
- Subsetting with a character variable requires a value in quotation marks
  - `where sex = ’F’;`
- Subsetting with a character variable does not require a value in quotation marks
  - `where age >= 21;`
- Subsetting with a date variable requires a value in quotation marks in the form `’ddmonyyyy’d`
  - `where Hire_Date < ’01Jan2000’d;`

Various Useful Operators in SAS

```
where Gender eq ' ';
where Salary ne .;
where Hire_Date < '01Jan2000'd;
where Country in ('AU','US');
where Order_Type in (1,2,3);
where Country ne 'AU' and Salary >= 50000;
where Country not in ('AU','US');
where Job_Title contains 'Rep';
where salary between 50000 and 100000;
where Last_Name between 'A' and 'L';
where Employee_ID is null;
where Employee_ID is not missing;
where Name like '%N';
where Name like 'T_m';
```

Operator Name	(Symbol) Description
eq	(=) equal to
ne	(ˆ=) not equal to
ge [le]	(>=) more [less] than or equal to
le	(<=) less than or equal to
in	equal to one of the items in a list
not	(ˆ) negates an operator
and	(&) if both operands are true
or	(!) if either or both operands is true
contains	(?) selects observation with specified string in ' '. This is case sensitive
between and	specifies an inclusive range
is null [is missing]	selects missing observations
like	selects observations with specified pattern % means any number of characters _ means exactly one character

Here's one data set to think about!

CU Jail Data

This data set is a subset of the Champaign County Sheriff Office data on bookings throughout the county for the year 2012. It contains no personal identifying information for people who were booked, but does include demographic variables and details about the crime and category of crime. The data set is a comma separated values (.csv) file called ccso_subset_2 and contains 2078 rows and 11 columns.

CU Jail Data

Variable Name	Description
employment_status	category of employment
jacket_number	ID number
race	person's racial identity
sex	male or female
citizenship	person's country
age_at_arrest	person's age
days_in_jail	whole number greater than 0
offense_level	category of offense
high_level_crime	category of crime
city	of person's residence
state	of person's residence

CU Jail Data

What questions might a data analyst have about this dataset?

What might be their goals for analysis?

Example: CU Jail Data

Goal: We are interested in the proportion of the races of folks in the Champaign or Urbana.
We could use a bar plot to visualize that.

```
data cujail;
 length employment_status $25. race $12. sex $6.
   citizenship $8. offense_level $24. high_level_crime $12.
   city $24. state $8.;
 infile 'location/ccso_subset_2.csv' dlm=',' dsd missover
   firstobs=2;
 input employment_status $ jacket_number race $ sex $
   citizenship $ age_at_arrest days_in_jail offense_level $
   high_level_crime $ city $ state $;
run;
proc sgplot data=cujail pctlevel=graph;
 vbar city /group=race groupdisplay=cluster ;
 where city in ('CHAMPAIGN','URBANA');
run;
```

Example: CU Jail Data

How might we tweak or enhance this plot?

What alternative plots might accomplish the goal?

Using **title** Statements to Enhance Results

- A **title** statement will add your specific titles to the top of the results page
- Titles remain in effect until they are changed or canceled
- To change a title, submit a new title statement with the same number but different text

The following SAS code replaces a previous title with the same title number and cancels all titles with higher numbers

```
title1 'ABC Cola';
title2 'Analytics Department';
title1 '123 Cola';
```

The Output Delivery System (ODS)

The following is useful for base SAS 9.4, not SAS Studio.

- Determines format for outputs (e.g. **ods html**, **ods rtf**, **ods pdf**)
- Can be used to control which graphics and tables are generated as well as the style of the SAS results.
- Provides useful controls for producing clean-looking results in SAS. See the coding sample below.

Producing Reports with the ODS (cont.)

Here's a coding sample of how to setup your SAS program code.

```
/* these are options below to make your reports look clean */
options nodate nonumber;
title ;
ods noproctitle;

/* this is where we name the file we are creating.
Only useful if you're using Base SAS 9.4*/
ods rtf file='fileName.rtf';

*this is where you can put your code;
data whatever; run; proc whatever; run;

/* when you open the rtf, you must close it
in order to access the rtf file on your computer*/
ods rtf close;
```

Creating the HW Report with SAS .rtf Results

When creating your HW Report you can do the following:

1. Create a results file which will be saved as a .rtf, using the coding sample on next slide for base SAS 9.4.
    - If using SAS OnDemand, simply click save as .rtf on the Results tab.
2. Once you have the .rtf file, use copy the tables and plots within it to a new Word document to create your HW Report file.
3. Then save it as a .pdf.

See the sample HW Report file in Compass for the correct formatting.

The Help System & SAS Procedures By Name

When curious or confused about a data step or a proc step or virtually anything in SAS, use the SAS Help Documentation.

- For SAS Studio, SAS University, or SAS OnDemand, click on the question mark in the top right corner of your screen.
- For SAS 9.4, click on the Help menu tab at the top of your screen.

You may also find the document in the link SAS Procedures By Name helpful for understanding syntax and meaning of various procedures that we will use this semester.

Exercise: CU Jail Data

1. How many people who were arrested lived outside of Champaign-Urbana?
2. When focusing on age at arrest, how many observations are missing?
3. Has anyone over the age of 70 and in Urbana been arrested?
4. Create a scatter plot of days in jail vs age at arrest for nonmissing ages only. What do you see?
5. Now create a scatter plot of days in jail vs age at arrest for black and white people arrested ages 35 and older. Describe the plot.