

# **Introduction to SAS**

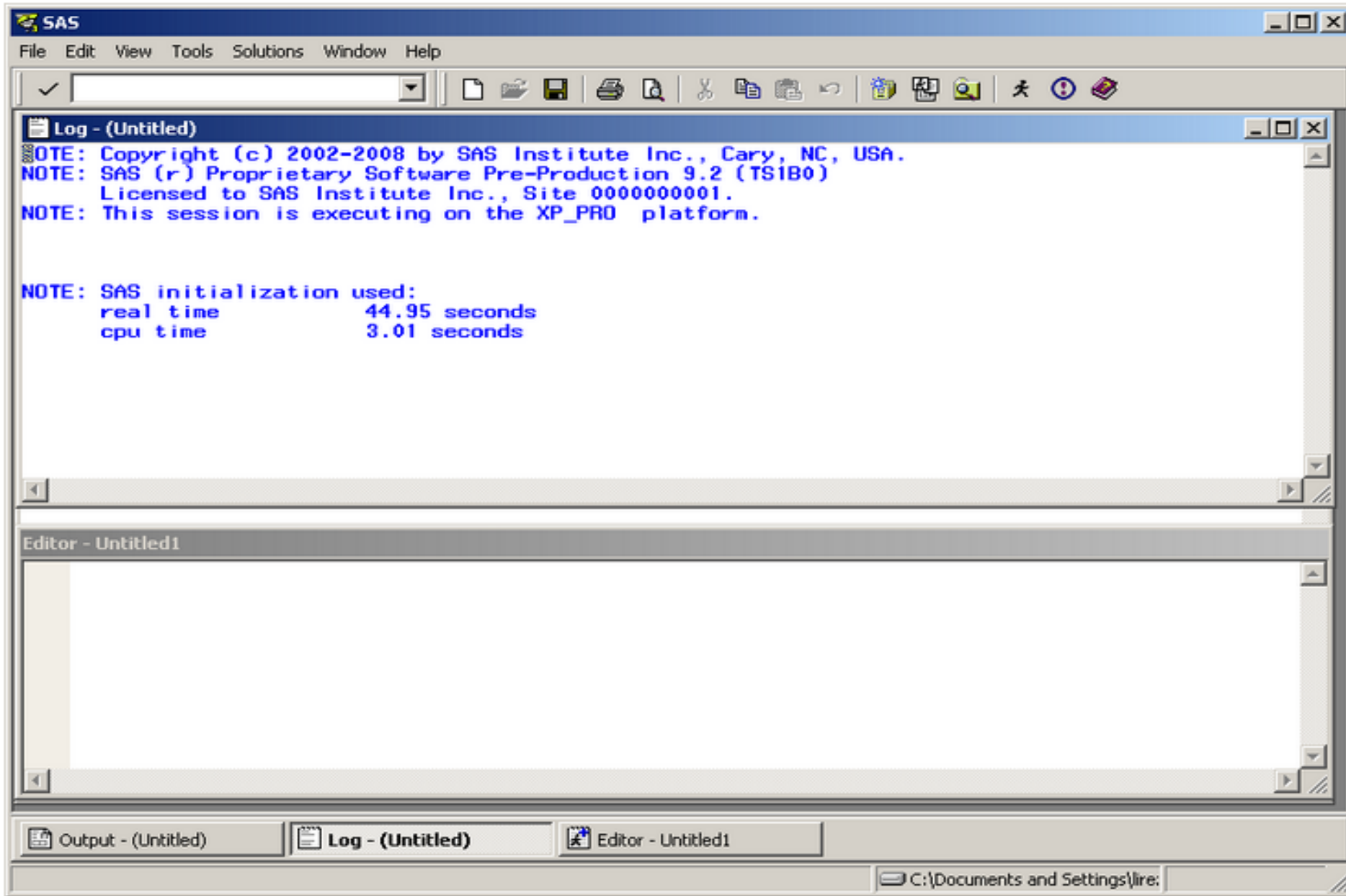
**BIO 226 - Spring 2015**

**Yoonyoung Park**

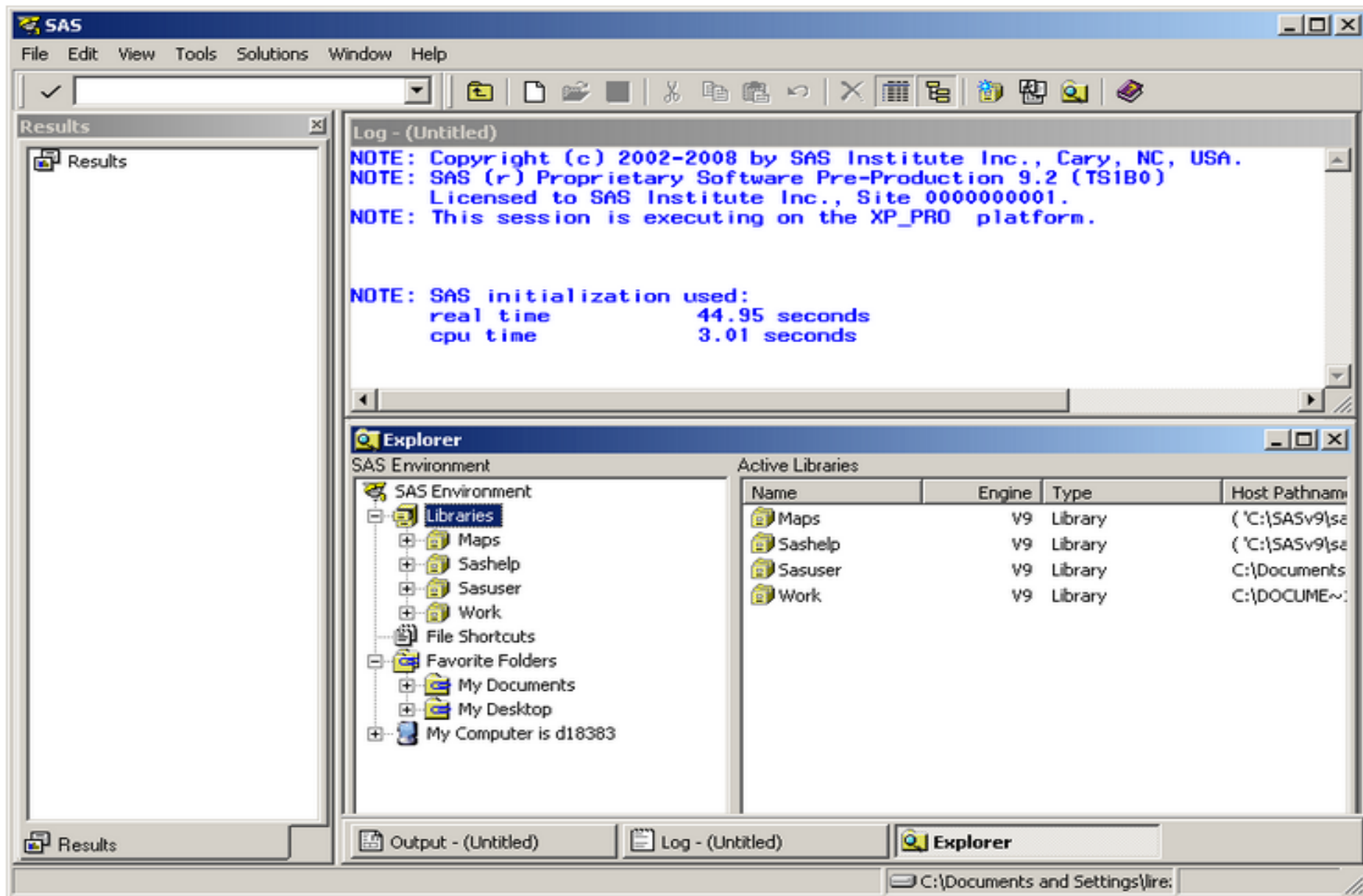
**Feb 3, 2015**

# The different SAS windows

- *Explorer*: contains SAS files and libraries
- *Editor*: where you can open or type SAS programs
- *Log*: stores details about your SAS session (code run, dataset created, errors...)
- *Results*: table of contents for output of programs
- *Output*: printed results of SAS programs



Editor and the Log windows are displayed, and the Output window is positioned behind these two windows.



The Log, Results, and Explorer windows.

# Basic SAS rules (1)

- Variable names must:
  - be 1 to 32 characters in length
  - begin with letter (A-Z) or underscore (\_)
  - continue with any combination of number, letters or underscores
- A variable's type is either *character* or *numeric*
- Missing values:
  - missing character data is left blank
  - missing numeric data is denoted by a period (.)

# Basic SAS rules (2)

- Two ways to make comments:
  - `* write comment here;`
  - `/* write comment here */`
- SAS is insensitive to case

# Basic programming rules (1)

SAS reads code line by line.

The end of a statement is marked by a semicolon ;

Statements are organized in **DATA** steps and **PROC** steps

**DATA** step: gives dataset a name, manipulates dataset

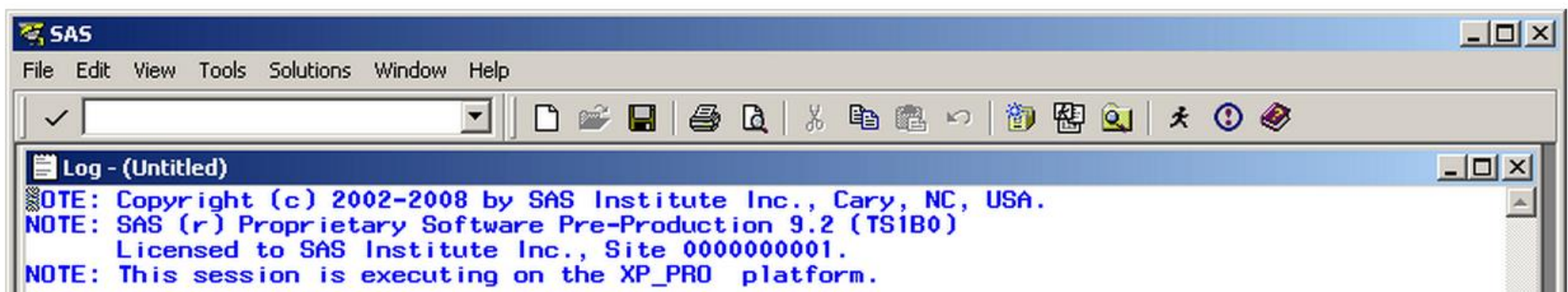
**PROC** step: procedure or analysis you want SAS to carry out

The **RUN**; statement signals to SAS that the previous statements can be executed.

Quotes can be single or double.

# Basic programming rules (2)

- SAS statements are free-format:
  - One statement can continue over several lines
  - Several statements can be on one line
- To submit program, highlight the code to run and click on the submit button (running silhouette)



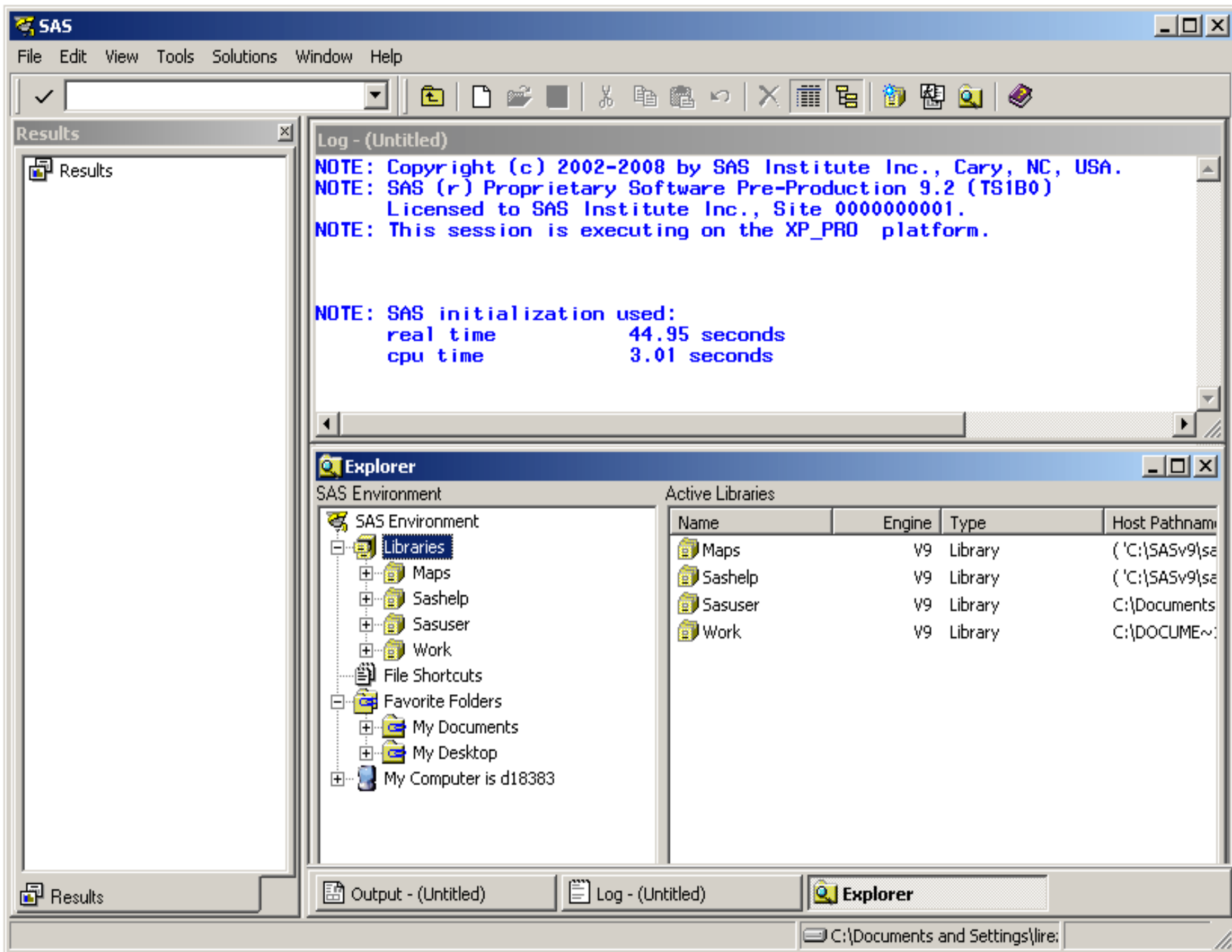


# Loading data

- If you have SAS data set (*sasintro.sas7bdat*) you can double click on it and it will load itself.
- If you don't have SAS data set (*sasintro.txt*), and the first row of your dataset contains the variable names, you can import it using *File > Import Data...* and specify the directory.
- Or you can use the following code:

```
DATA mydata;  
INFILE 'D:\BIO226\sasintro.txt' ;  
INPUT weight bmi id age activity education smoking;  
RUN;
```

- Setting your current directory: on the bottom line of the main SAS window, you should see it set to *C:\WINDOWS\system32*. Double click on it to change it.



# How to view the loaded data?

- Go in the *Explorer window*, double click on *Libraries*, then *Work* and *sasintro.sas7bdat*
- To view general information about the data set, like variables' name and type:

```
PROC CONTENTS DATA=mydata;  
RUN;
```

- Use the PRINT procedure to view the first 25 records:

```
PROC PRINT DATA=mydata (OBS=25) ;  
RUN;
```

# Variables from sasintro.txt

<u>#</u>	<u>Variable</u>	<u>Type</u>	<u>Unit</u>
5	activity	Num	kcal/week
4	age	Num	years
2	bmi	Num	kg/m <sup>2</sup>
6	education	Num	years
3	id	Num	
7	smoking	Num	1:current smoker, 0:non-smoker
1	weight	Num	lbs

# Manipulating data (1)

- selecting a subset of rows (= observations)

```
DATA mydata_s;  
SET mydata;  
IF smoking=1;  
RUN;
```

- deleting columns (= variables)

```
DATA mydata2;  
SET mydata;  
DROP weight education;  
RUN;
```

# Manipulating data (2)

- adding columns (= variables)

```
DATA mydata3;  
SET mydata;  
weight_kg=weight*0.453;  
IF age <= 60 THEN agegroup=1;  
ELSE IF age<=70 THEN agegroup=2;  
ELSE agegroup=3;  
RUN;
```

# Sorting data

```
PROC SORT DATA=mydata OUT=mydata4;  
BY id age weight;
```

```
PROC PRINT DATA=mydata (OBS=25);  
PROC PRINT DATA=mydata4 (OBS=25);  
RUN;
```

# Summarizing data (1)

- Summarizing weight:

```
PROC MEANS DATA=mydata;  
VAR weight;  
RUN;
```

- Summarizing weight in the youngest agegroup:

```
PROC MEANS DATA=mydata3;  
VAR weight;  
WHERE agegroup=1;  
RUN;
```



# Summarizing data (2)

- Summarizing weight by smoking status (two possible codes):

```
PROC SORT DATA=mydata OUT=mydata5;  
BY smoking;  
PROC MEANS DATA=mydata5;  
VAR weight;  
BY smoking;  
RUN;
```

or

```
PROC MEANS DATA=mydata;  
CLASS smoking;  
VAR weight;  
RUN;
```

- All these summarizing measures can be obtained with PROC UNIVARIATE also.

# Categorical data and correlation

- Summarizing categorical data

```
PROC FREQ DATA=mydata3;  
TABLES smoking*agegroup /chisq exact;  
RUN;
```

- Examining correlation

```
PROC CORR DATA=mydata;  
VAR weight;  
WITH bmi age;  
RUN;
```

# Basic procedures: plots

- Barcharts

```
PROC CHART DATA=mydata3;  
VBAR agegroup /DISCRETE;  
RUN;
```

- Scatterplot

```
PROC PLOT DATA=mydata3;  
PLOT bmi*weight='*';  
RUN;
```

- Histogram, Boxplot, Normal Probability Plot

```
PROC UNIVARIATE DATA=mydata3 PLOT;  
VAR weight;  
RUN;
```

# /\* Libraries \*/

- A library is the directory where your SAS dataset is stored.
- The default library is named *Work* and stores your SAS datasets temporarily: they will be deleted when you end your SAS session
- If you want to save your SAS datasets and use them again later, create your own library by using the menus.
- Then enter the code

```
LIBNAME SAS_Lab 'D:\BIO226\SAS';  
DATA SAS_Lab.mydata;  
INFILE 'D:\BIO226\sasintro.txt';  
INPUT weight bmi id age activity education  
smoking;  
RUN;
```

# SAS output and Word

- To send you SAS output to a Word document:

```
ODS RTF FILE='D:\BIO226\output.RTF'  
style=minimal;
```

```
PROC CORR DATA =mydata;
```

```
    VAR weight;
```

```
    WITH bmi age;
```

```
    RUN;
```

```
ODS RTF CLOSE;
```

- Other styles: **Journal, Analysis, Statistical**

# For further references

- SAS9 Documentation on the Web:  
<http://support.sas.com/onlinedoc/913/docMainpage.jsp>
- Applied Statistics and the SAS Programming Language  
(5th Edition) Ron P. Cody and Jeffrey K. Smith
- The Little SAS Book, L.D. Delwiche and S.J. Slaughter
- See *SAS\_help.doc* on course website