Introduction to SAS

Statistics 135

Autumn 2005

Copyright ©2005 by Mark E. Irwin

SAS

Probably the most popular Statistical software worldwide.

SAS claims that its products are used at over 40,000 sites, including at 90% of the Fortune 500.

This will not be all SAS as they make other products, such as JMP (a menu and dialogue based stat package)

Huge in the pharmaceutical industry. There is a belief, though 100% FALSE, that the FDA requires the analysis of all clinical trials to be done in SAS. You can do it in anything, you just need to document it as part of your approval submission.

SAS, the company, has the reputation of being a fantastic place to work as well.

Extremely powerful package. You name it, it probably does it. If it doesn't, they are probably working on it.

It is available for many platforms including Windows, Unix, Macintosh, mainframes (z/OS, CMS, VSE, VMS, MVS). However they don't keep all versions updated at the same rate (e.g. Mac version only goes to version 6.12, OS/2 is at 8.2, whereas Windows is at 9.1).

It's a program based package. You need to write a program for your analysis. There is no menu based approach like Stata, Minitab, or SPSS have available. You can't type commands at a prompt as you can with **R** and **S-Plus**.

These programs will have a block structure, with each block corresponding to a different part of the analysis.

Each block will usually start with a different PROC statement, such as PROC REGRESS, PROC SORT, PROC LOGISTIC, etc. Within each block, commands will be given, options set, etc.

It is also extendable with the built-in Macro language.

Example SAS Analysis

```
* Sample SAS program
* Data set is from Dean and Voss (1999) Design and Analysis of
* Experiments.  Problem 3, page 129.

options linesize=75; /* set the output width to 75 characters */

data temp;
  infile 'p147.3';
  input brand time;
  invtime=1/time;

* print the data to see if everything is ok

proc print data=temp;
  title 'Margarine Experiment';
```

```
* Run the ANOVA

proc glm;
  classes brand;    /* declare brand to be a factor */
  model invtime = brand;
  estimate 'marg vs but' brand 1 1 1 -3/divisor=3;
  output out=resids predicted=pred residual=z;
run;

* Switch from data file temp to data file resids
data;

  set resids;

* Standardize the residuals to have standard deviation 1

proc standard std=1.0;
  var z;
```

```
* Calculate the normal scores with Blom's adjustment

proc rank normal=blom;
  var z;
  ranks nscores;

proc print;

* Generate diagnostic plots

proc plot;
  plot invtime*brand z*pred z*brand z*nscores;

run;

quit;  /* Ends the program */
```

Obs	brand	time	invtime
1	1	167	.005988024
2	1	171	.005847953
3	1	178	.005617978
4	1	175	.005714286
5	1	184	.005434783
		.	
		.	
		.	
36	4	223	.004484305
37	4	209	.004784689
38	4	219	.004566210
39	4	212	.004716981
40	4	210	.004761905

The GLM Procedure

Class Level Information

Class	Levels	Values
brand	4	1 2 3 4

Number of observations 40

The GLM Procedure

Dependent Variable: invtime

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	0.00001782	0.00000594	208.33	<.0001
Error	36	0.00000103	0.00000003		
Corrected Total	39	0.00001884			

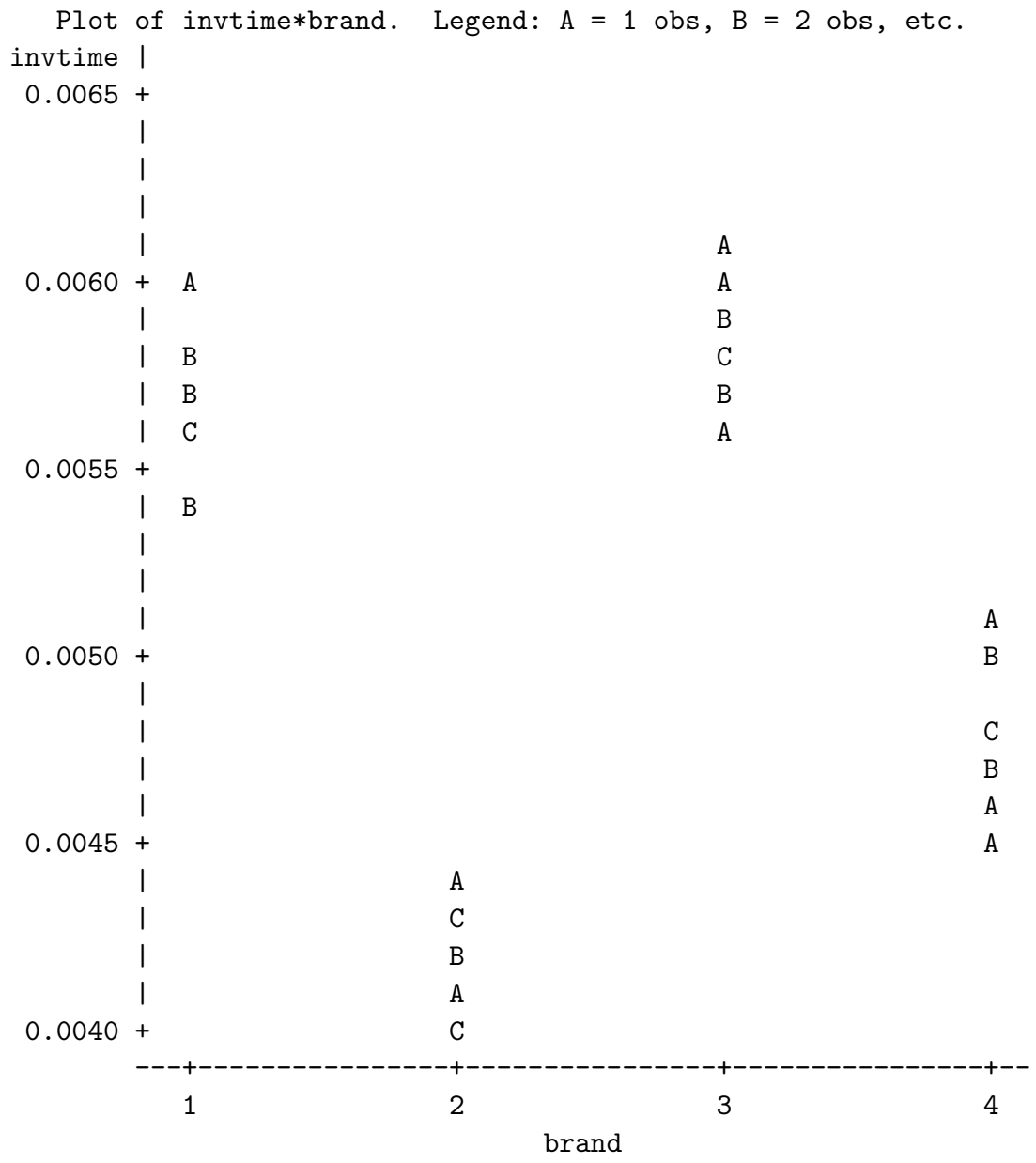
R-Square	Coeff Var	Root MSE	invtime Mean
0.945537	3.294115	0.000169	0.005125

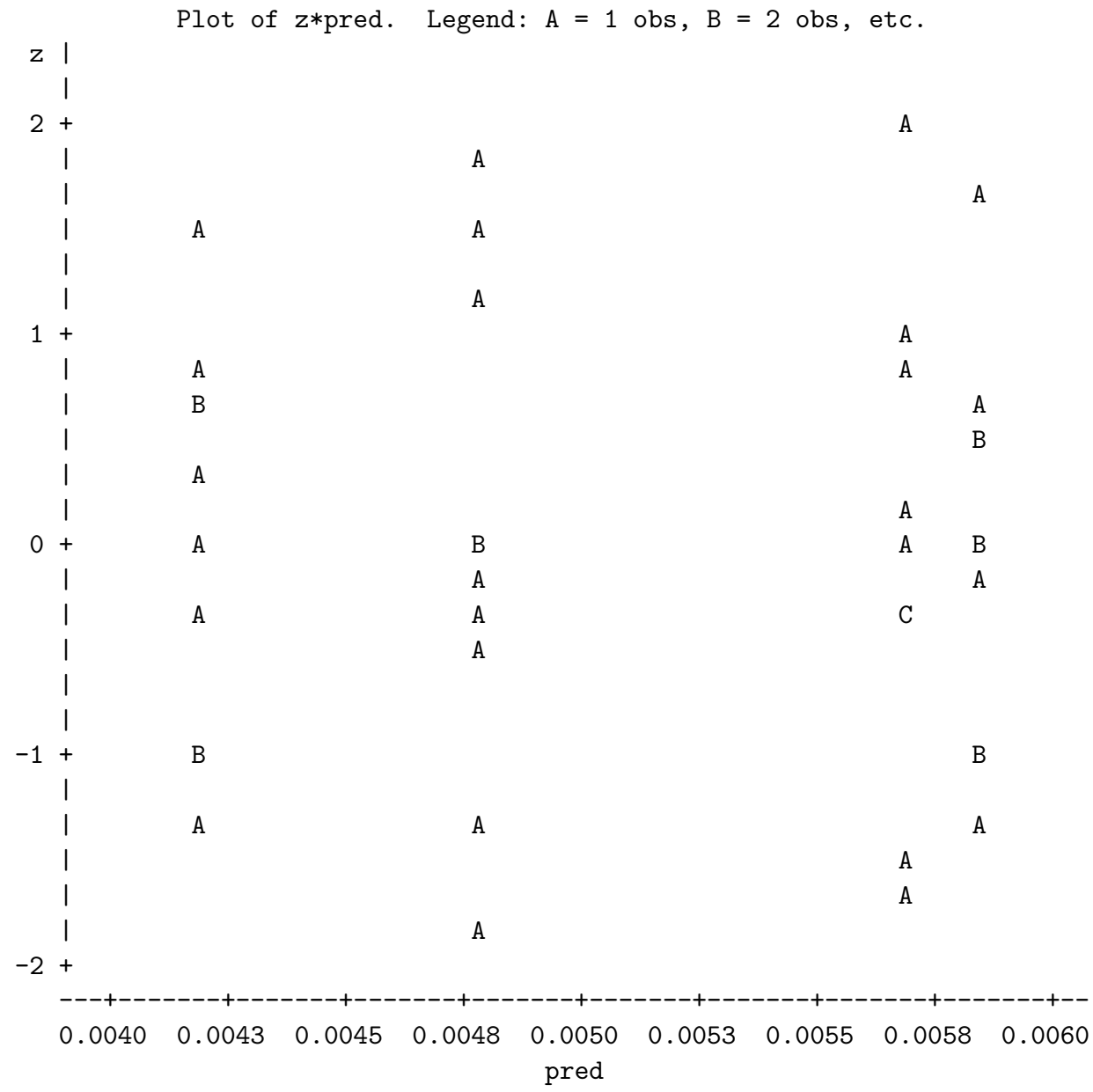
Source	DF	Type I SS	Mean Square	F Value	Pr > F
brand	3	0.00001782	0.00000594	208.33	<.0001

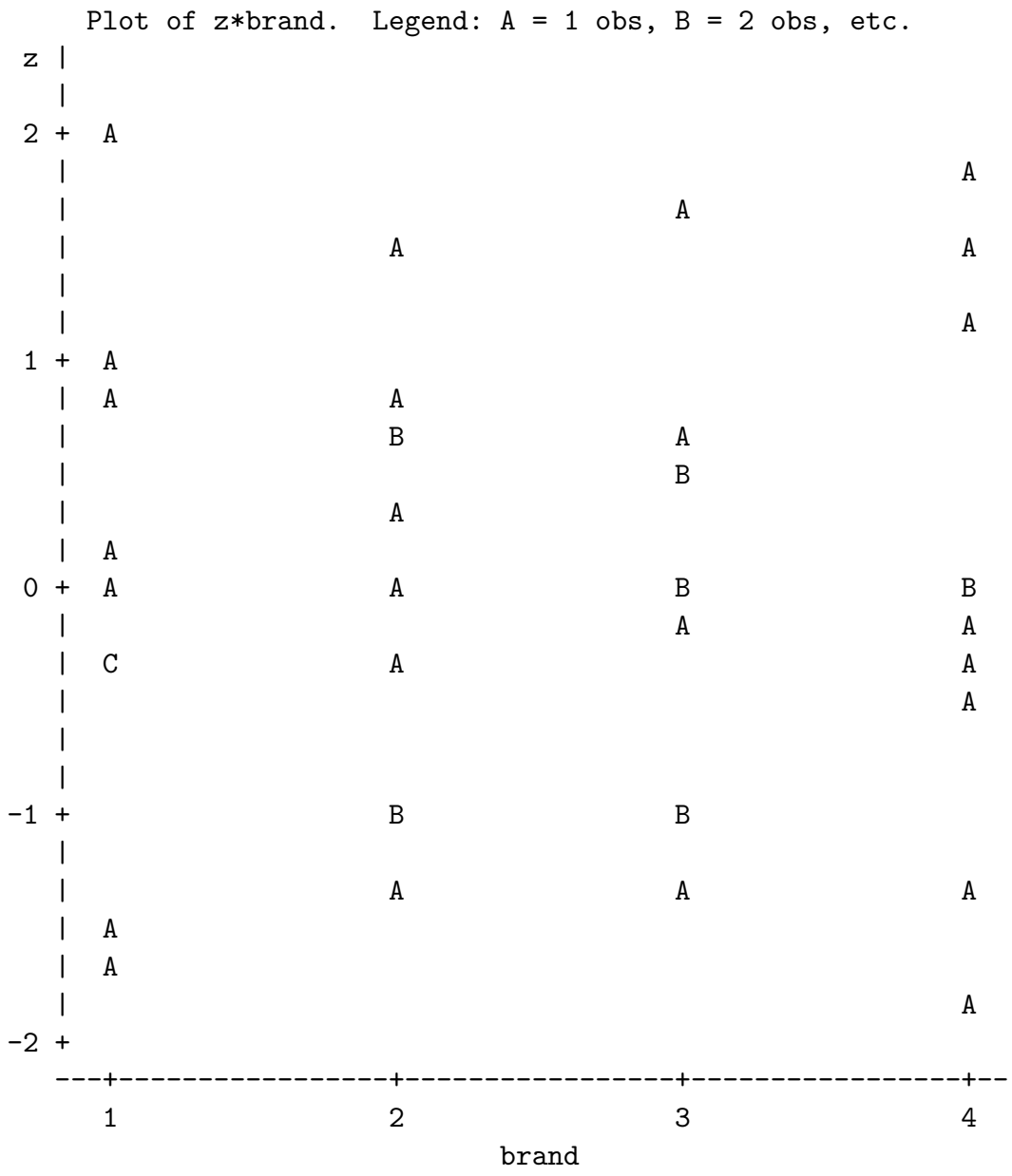
Source	DF	Type III SS	Mean Square	F Value	Pr > F
brand	3	0.00001782	0.00000594	208.33	<.0001

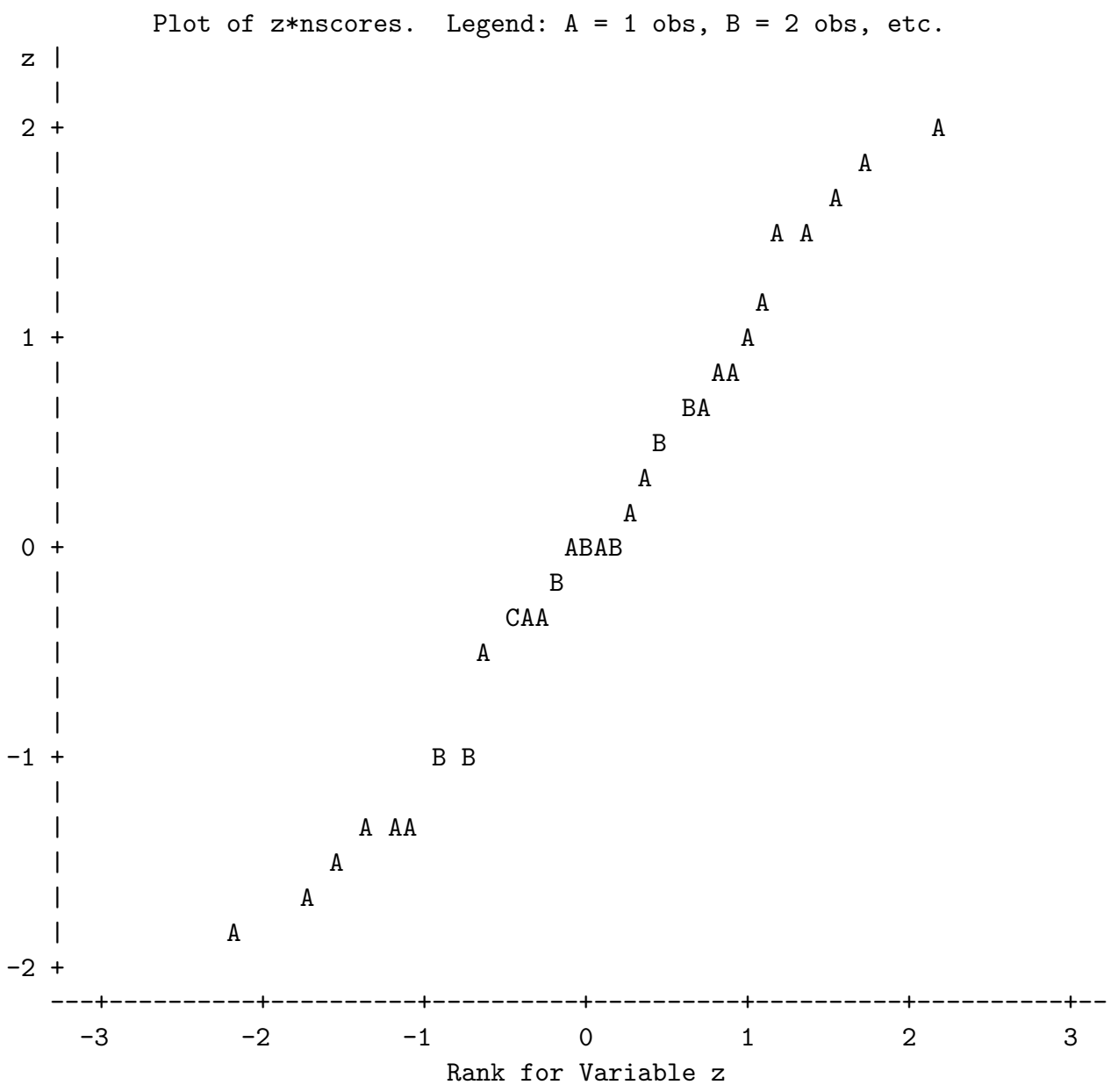
Parameter	Estimate	Standard Error	t Value	Pr > |t|
marg vs but	0.00044184	0.00006165	7.17	<.0001

Obs	brand	time	invtime	pred	z	nscores
1	1	167	.005988024	.005674016	1.93577	2.15636
2	1	171	.005847953	.005674016	1.07227	0.97574
3	1	178	.005617978	.005674016	-0.34546	-0.48629
4	1	175	.005714286	.005674016	0.24825	0.28402
			.			
			.			
			.			
36	4	223	.004484305	.004794041	-1.90943	-2.15636
37	4	209	.004784689	.004794041	-0.05765	0.00000
38	4	219	.004566210	.004794041	-1.40451	-1.34037
39	4	212	.004716981	.004794041	-0.47505	-0.63114
40	4	210	.004761905	.004794041	-0.19811	-0.21972


Plot of invtime*brand. Legend: A = 1 obs, B = 2 obs, etc.
invtime
0.0065
0.0060
0.0055
0.0050
0.0045
0.0040
1
2
3
4
brand


Plot of z*pred. Legend: A = 1 obs, B = 2 obs, etc.
z
2
1
0
-1
-2
0.0040 0.0043 0.0045 0.0048 0.0050 0.0053 0.0055 0.0058 0.0060
pred


Plot of z*brand. Legend: A = 1 obs, B = 2 obs, etc.
z
2
1
0
-1
-2
1
2
3
4
brand


Plot of z*nscores. Legend: A = 1 obs, B = 2 obs, etc.
z
2
1
0
-1
-2
-3
-2
-1
0
1
2
3
Rank for Variable z

Also generated during a **SAS** run is a log file, describing how things went during the run. Usually its not useful, though if there is a bug in your **SAS** code, it can help find it.

The following is part of the log file for the previous analysis, just to give you the flavour of what you can get.

```
NOTE: Copyright (c) 1999-2001 by SAS Institute Inc., Cary, NC, USA.
NOTE: SAS (r) Proprietary Software Release 8.2 (TS2M0)
      Licensed to OHIO STATE UNIVERSITY -ACADEMIC TECHNOLOGY SERVICE, Site 00163
55002.
NOTE: This session is executing on the SunOS 5.8 platform.



This message is contained in the SAS news file, and is presented upon
initialization.  Edit the files "news" in the "misc/base" directory to
display site-specific news and information in the program log.
The command line option "-nonews" will prevent this display.
```

```
NOTE: SAS initialization used:
      real time           2:12.18
      cpu time            1.05 seconds

203   * Sample SAS program
204   * Data set is from Dean and Voss (1999) Design and Analysis of
205   * Experiments.  Problem 3, page 129.
206
207   * This data set is from an experiment which examined whether there are
208   * differences in melting times for margarine.  Three different
209   * margarines (brands 1 - 3) were studied and butter was also used as a
210   * control group (brand 4).;
211
212   options linesize=75;   /* set the output width to 75 characters */
213



214   data temp;
215     infile 'p147.3';
216     input brand time;
217     invtime=1/time;
```

```
218
219  * print the data to see if everything is ok;

NOTE: The infile 'p147.3' is:
      File Name=/home/irwin/SAS/p147.3,
      Owner Name=irwin,Group Name=parstaff,
      Access Permission=rw-------,
      File Size (bytes)=240

NOTE: 40 records were read from the infile 'p147.3'.
      The minimum record length was 5.
      The maximum record length was 5.
NOTE: The data set WORK.TEMP has 40 observations and 3 variables.
NOTE: DATA statement used:
      real time           1.75 seconds
      cpu time            0.01 seconds


220  proc print data=temp;
221    title 'Margarine Experiment';
222
```

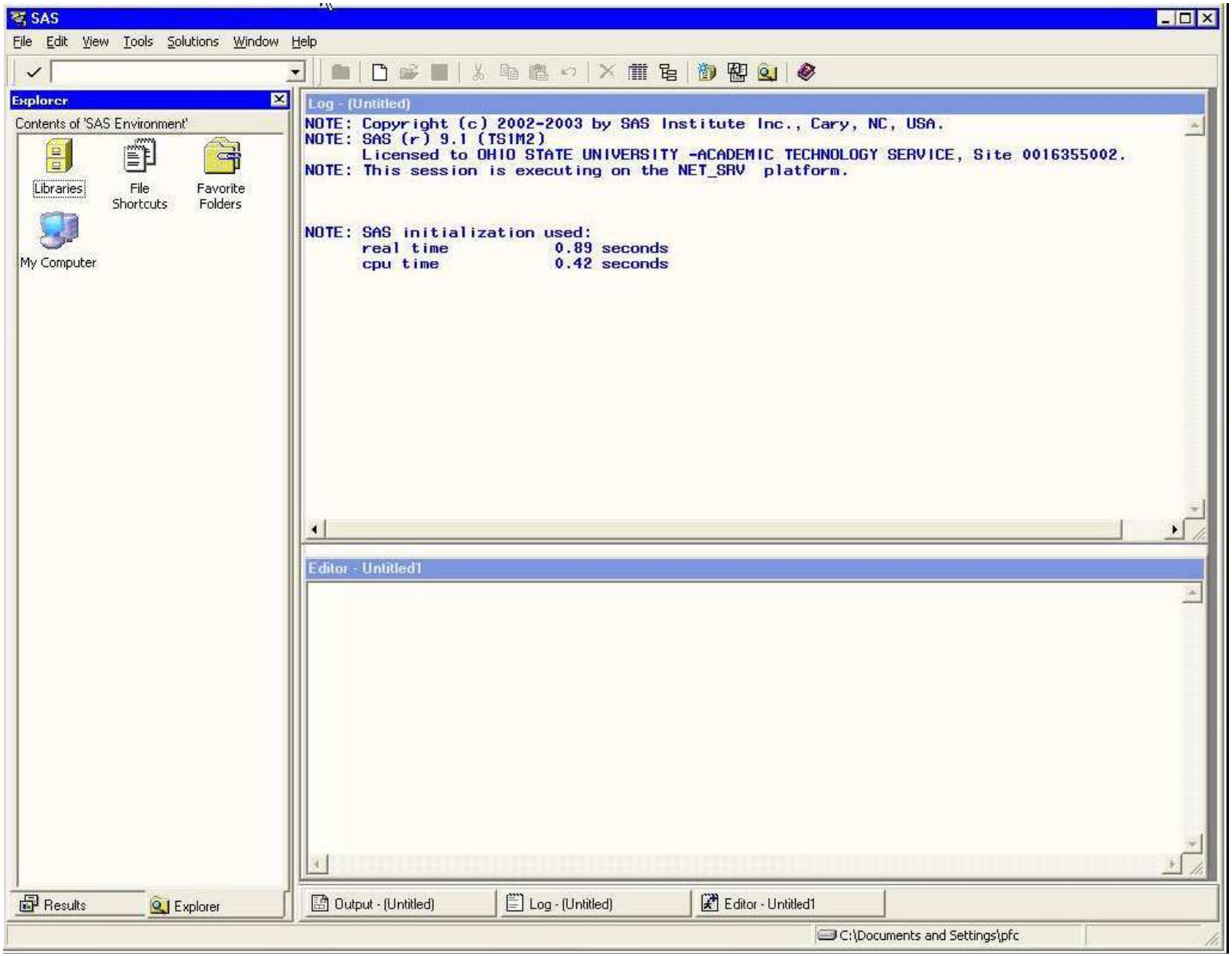
```
223  * Run the ANOVA;

NOTE: There were 40 observations read from the data set WORK.TEMP.
NOTE: PROCEDURE PRINT used:
      real time           0.61 seconds
      cpu time            0.01 seconds



224  proc glm;
225    classes brand;
226    model invtime = brand;
227    estimate 'marg vs but' brand 1 1 1 -3/divisor=3;
228    output out=resids predicted=pred residual=z;
229  run;

230
231  * Switch from data file temp to data file resids;

NOTE: The data set WORK.RESIDS has 40 observations and 5 variables.
NOTE: PROCEDURE GLM used:
      real time           1.32 seconds
```

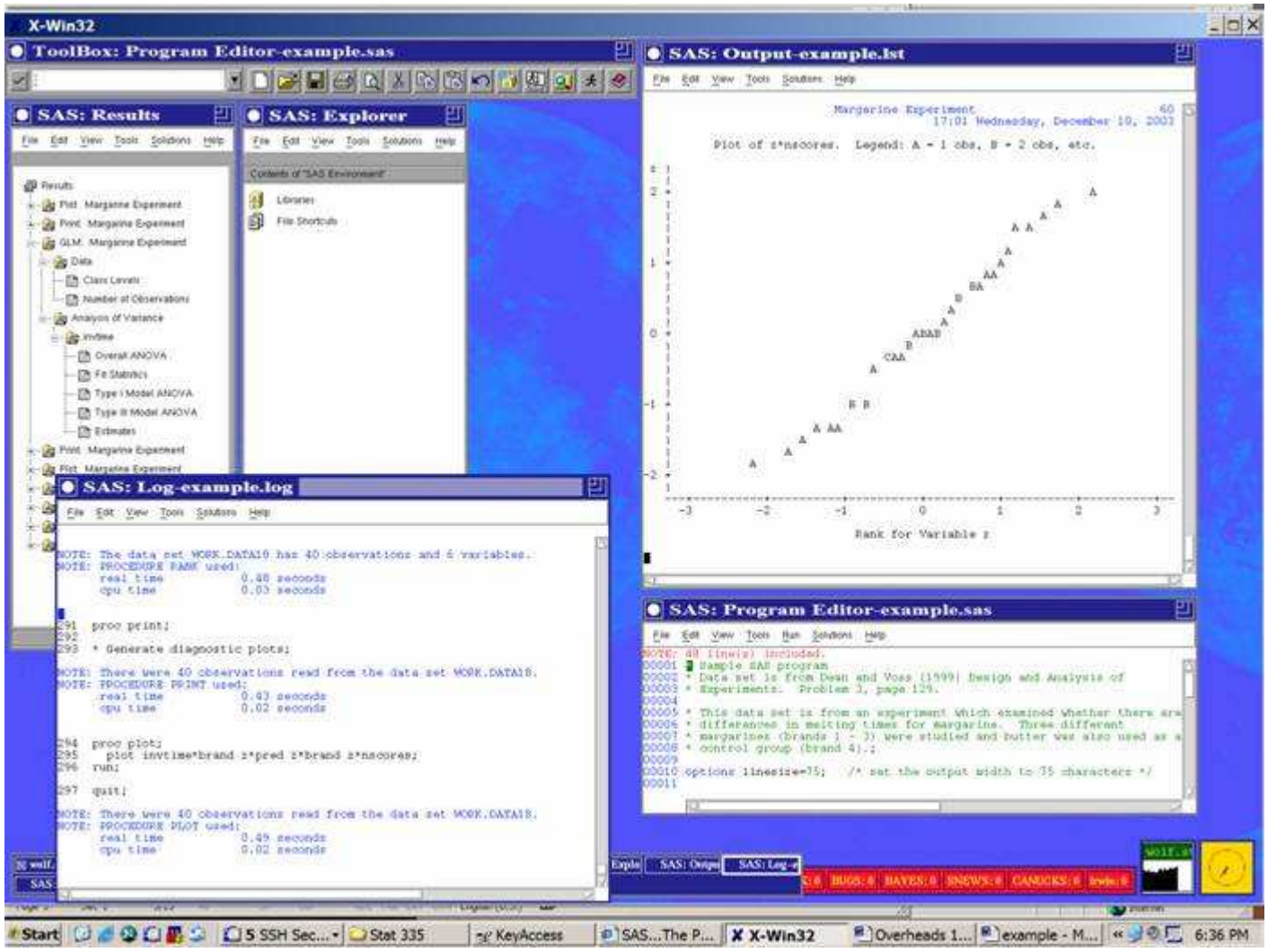
```
      cpu time            0.09 seconds


232  data;
233    set resids;
234
235  * Standardize the residuals to have standard deviation 1;

NOTE: There were 40 observations read from the data set WORK.RESIDS.
NOTE: The data set WORK.DATA13 has 40 observations and 5 variables.
NOTE: DATA statement used:
      real time           0.69 seconds
      cpu time            0.01 seconds
```

The Look of SAS

On most platforms, **SAS** looks similar. There will be some minor differences, depending on the windowing system used. There will be 5 different windows available

- Results: Stores objects created by **SAS**. Allows for easier searching of **SAS** output
- Explorer: A view of the **SAS** files
- Output: Program output
- Log: Informs you of current status. Messages appear here to tell you how things are going. Error messages also go here.
- Editor: Editor for writing and submitting **SAS** jobs.


SAS
File Edit View Tools Solutions Window Help
Explorer
Contents of 'SAS Environment'
Libraries
File Shortcuts
Favorite Folders
My Computer
Log - (Untitled)
NOTE: Copyright (c) 2002-2003 by SAS Institute Inc., Cary, NC, USA.
NOTE: SAS (r) 9.1 (TS1M2)
Licensed to OHIO STATE UNIVERSITY -ACADEMIC TECHNOLOGY SERVICE, Site 0016355002.
NOTE: This session is executing on the NET_SRV platform.
NOTE: SAS initialization used:
real time 0.89 seconds
cpu time 0.42 seconds
Editor - Untitled1
Results
Explorer
Output - (Untitled)
Log - (Untitled)
Editor - Untitled1
C:\Documents and Settings\pfc



On Unix systems, batch jobs can be run from the unix prompt. In this case the windowing system will not start up.

PROCs

As mentioned earlier, most commands in **SAS** are based on PROC statements. Examples of some useful PROCs are

- UNIVARIATE (BASE): univariate summary statistics
- CORR (BASE): Correlations
- MEANS (BASE): Similar to UNIVARIATE, but adds in some simple testing as well.
- FREQ (BASE): Creates and analyzes contingency tables
- TABULATE (BASE): Descriptive statistics in tabular format
- TTEST (STAT): 1 and 2 sample t tests

- `ANOVA` (STAT): Analysis of variance for balanced designs
- `REG` (STAT): Linear regression
- `GLM` (STAT): General linear model. Similar to `lm()` in **S**
- `LOGISTIC` (STAT): Logistic regression

The designation in the above list of BASE or STAT indicates which subpackage includes the `PROC`. It indicates which manual you want to lookup, which includes the online documentation available from the **SAS** website. The online documentation pages are available from the course website. Go to the SAS page to find them.