

Chapter 30

Working with Other SAS Products

Chapter Table of Contents

VIEWING RESULTS FROM SAS/STAT SOFTWARE	442
SUBMITTING SAS/INSIGHT STATEMENTS	448
RECORDING SAS/INSIGHT STATEMENTS	451
REFERENCES	452

Chapter 30

Working with Other SAS Products

This chapter illustrates how to use SAS/INSIGHT software with other components of the SAS System.

A typical usage is to create an analysis in another SAS product and then view the results using SAS/INSIGHT software. For example, you can use SAS/STAT software to create an analysis and use SAS/INSIGHT software to display its results. This enables you to take advantage of the strengths of both products.

You can also use grammar statements to drive SAS/INSIGHT software from other SAS products. This enables you to save time by automating repetitive tasks.

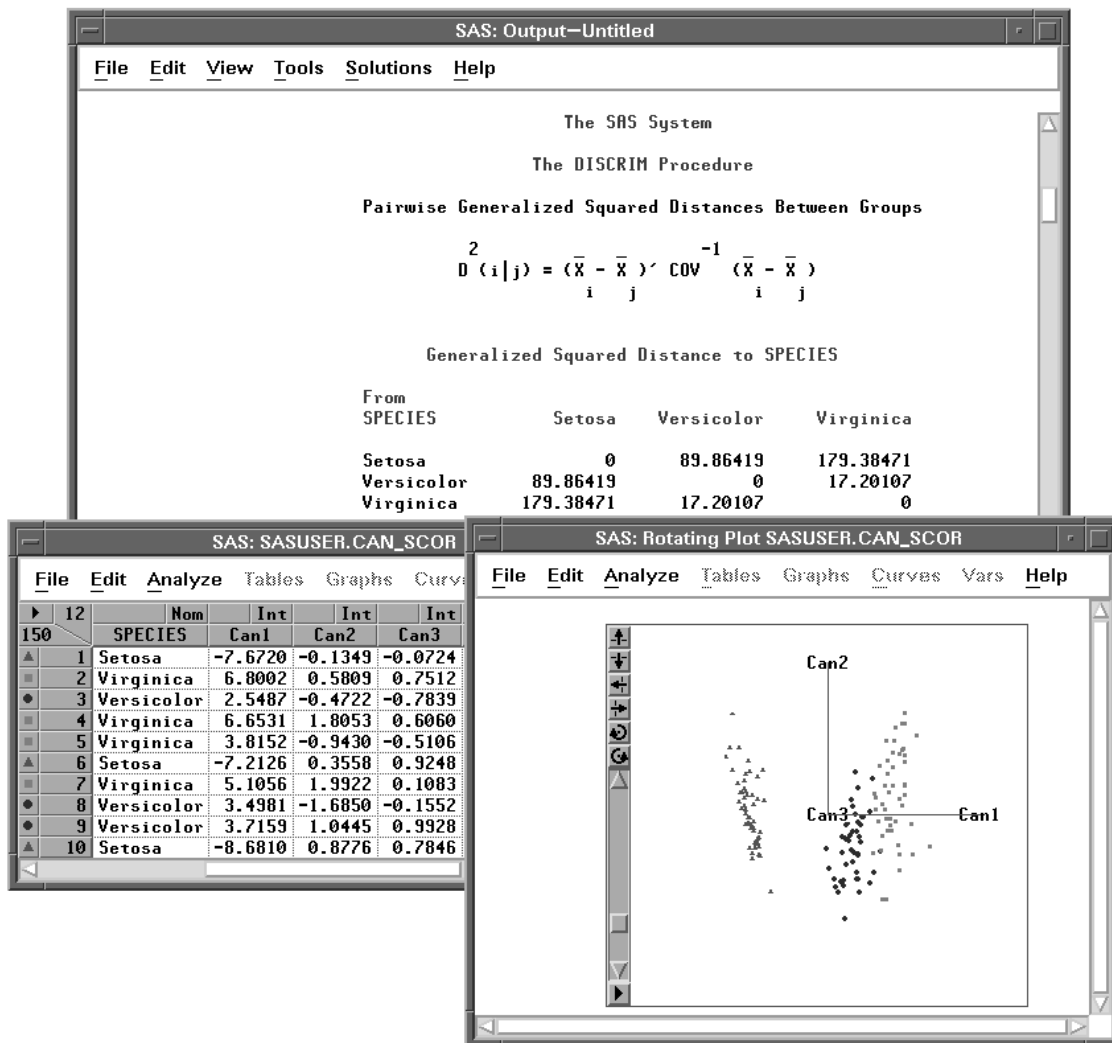
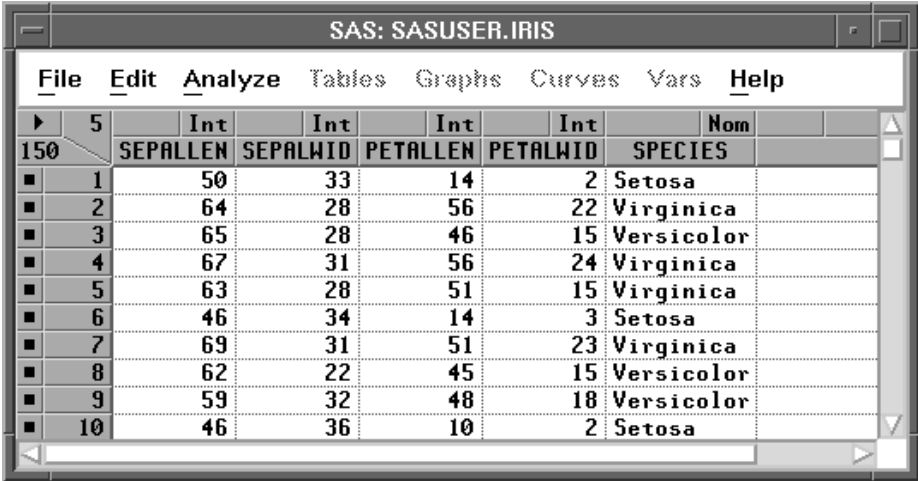


Figure 30.1. Viewing Results from SAS/STAT Software

Viewing Results from SAS/STAT Software

The **IRIS** data, published by Fisher (1936), have been used widely for examples in discriminant analysis. The goal of the analysis is to find functions of a set of quantitative variables that best summarize the differences among groups of observations determined by the classification variable. The **IRIS** data contain four quantitative variables measured on 150 specimens of iris plants. These include sepal length (**SEPALLEN**), sepal width (**SEPALWID**), petal length (**PETALLEN**), and petal width (**PETALWID**). The classification variable, **SPECIES**, represents the species of iris from which the measurements were taken. There are three species in the data: *Iris setosa*, *Iris versicolor*, and *Iris virginica*.



	5	Int	Int	Int	Int	Nom
150	SEPALLEN	SEPALWID	PETALLEN	PETALWID	SPECIES	
1	50	33	14	2	Setosa	
2	64	28	56	22	Virginica	
3	65	28	46	15	Versicolor	
4	67	31	56	24	Virginica	
5	63	28	51	15	Virginica	
6	46	34	14	3	Setosa	
7	69	31	51	23	Virginica	
8	62	22	45	15	Versicolor	
9	59	32	48	18	Versicolor	
10	46	36	10	2	Setosa	

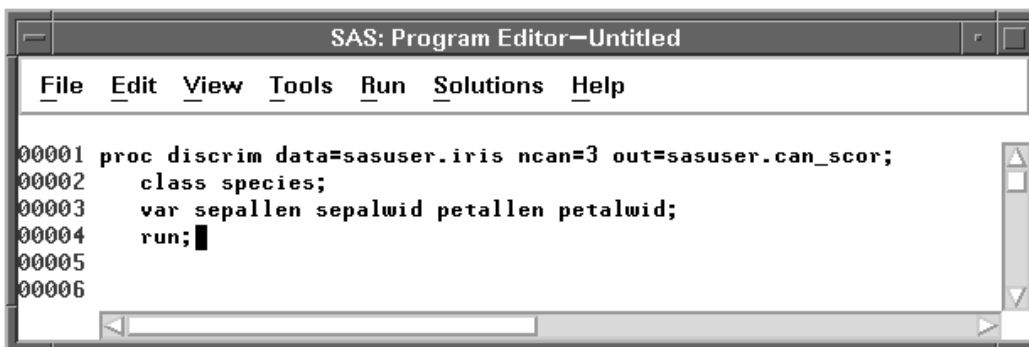
Figure 30.2. IRIS Data Set

Linear combinations of the four measurement variables best summarize the differences among the three species, assuming multivariate normality with covariance constant among groups. This requires a canonical discriminant analysis that is available in both SAS/INSIGHT software and SAS/STAT software. The following steps illustrate how to create an output data set that contains scores on the canonical variables in SAS/STAT software and how to use SAS/INSIGHT software to plot them.

⇒ **If you are running the SAS System in interactive line mode, exit the SAS System and reenter under the display manager.**

You must invoke SAS/INSIGHT software from a command line or from the **Solutions** menu to use SAS/INSIGHT software and the Program Editor concurrently.

⇒ **In the Program Editor, enter the statements shown in Figure 30.3.**



```

SAS: Program Editor—Untitled
File Edit View Tools Run Solutions Help
00001 proc discrim data=sasuser.iris ncan=3 out=sasuser.can_scor;
00002     class species;
00003     var sepallen sepalwid petallen petalwid;
00004     run;
00005
00006

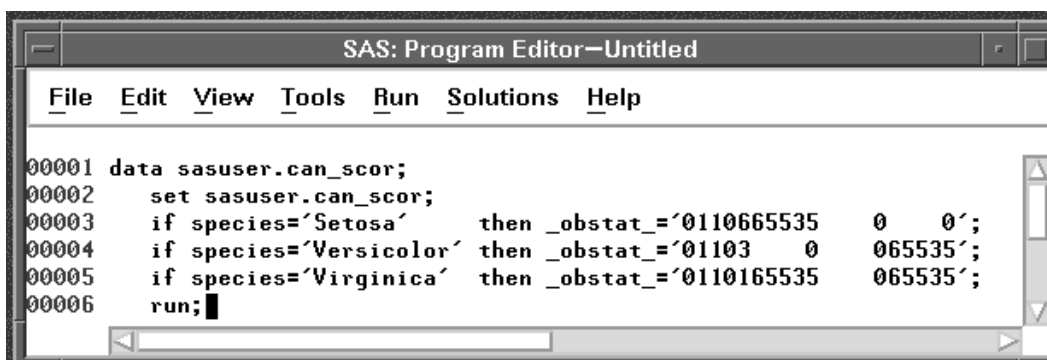
```

Figure 30.3. Program Editor with PROC Statement

The **OUT=** option in the **PROC DISCRIM** statement puts the scores and the original variables in the **SASUSER** library in a data set called **CAN_SCOR**. For complete documentation on the **DISCRIM** procedure, refer to the chapter titled “The **DISCRIM** Procedure,” in the *SAS/STAT User’s Guide*.

⇒ In the Program Editor, enter the statements in Figure 30.4.

These statements create the **_OBSTAT_** variable, which stores observation colors, shapes, and other states. If you create the **_OBSTAT_** variable as shown, **SETOSA** observations will be red triangles, **VERSICOLOR** observations will be blue circles, and **VIRGINICA** observations will be magenta squares.



```

SAS: Program Editor—Untitled
File Edit View Tools Run Solutions Help
00001 data sasuser.can_scor;
00002     set sasuser.can_scor;
00003     if species='Setosa' then _obstat_='0110665535 0 0';
00004     if species='Versicolor' then _obstat_='01103 0 065535';
00005     if species='Virginica' then _obstat_='0110165535 065535';
00006     run;

```

Figure 30.4. Program Editor with DATA Step

OBSTAT is a character variable. You can use it to set other observation states in addition to color and shape. The format of the **_OBSTAT_** variable is as follows.

- | | |
|-------------|--|
| Character 1 | stores the observation’s selection state. It is '1' for selected observations and '0' for observations that are not selected. |
| Character 2 | stores the observation’s Show/Hide state. It is '1' for observations that are displayed in graphs and '0' for observations that are not displayed in graphs. |

Part 2. Introduction

- Character 3 stores the observation's Include/Exclude state. It is '1' for observations that are included in calculations and '0' for observations that are excluded from calculations.
- Character 4 stores the observation's Label/UnLabel state. It is '1' for observations whose label is displayed by default, and '0' for observations whose label is not displayed by default.
- Character 5 stores the observation's marker shape, a value between '1' and '8':
- 1 Square
 - 2 Plus
 - 3 Circle
 - 4 Diamond
 - 5 X
 - 6 Up Triangle
 - 7 Down Triangle
 - 8 Star
- Characters 6–20 store the observation's color as Red-Green-Blue (RGB) components. The RGB color model represents colors as combinations of the colors red, green, and blue. You can obtain intermediate colors by varying the proportion of these primary colors.
- Each component is a 5-digit decimal number between 0 and 65535. Characters 6–10 store the red component. Characters 11–15 store the green component. Characters 16–20 store the blue component.

The `_OBSTAT_` variable can be used to create color blends as well as discrete colors. For an example of this usage, refer to Robinson (1995).

⇒ **Choose Run:Submit to submit the SAS statements.**

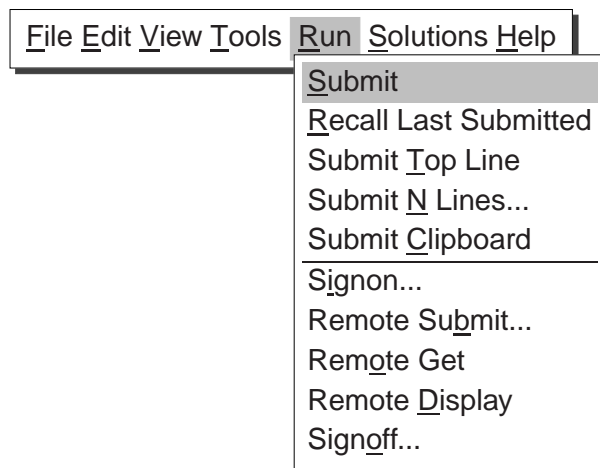


Figure 30.5. Run Menu

This produces the PROC DISCRIM output shown in Figure 30.6 and creates the **CAN_SCOR** data set.

The SAS System

The DISCRIM Procedure

Observations	150	DF Total	149
Variables	4	DF Within Classes	147
Classes	3	DF Between Classes	2

Class Level Information

SPECIES	Variable Name	Frequency	Weight	Proportion	Prior Probability
Setosa	Setosa	50	50.0000	0.333333	0.333333
Versicolor	Versicolor	50	50.0000	0.333333	0.333333
Virginica	Virginica	50	50.0000	0.333333	0.333333

Figure 30.6. PROC DISCRIM Output

⇒ Invoke SAS/INSIGHT software, and open the **CAN_SCOR** data set.

⇒ Scroll to the right to see the canonical variables **CAN1**, **CAN2**, and **CAN3**.

These variables represent the linear combinations of the four measurement variables that summarize the differences among the three species.

	Int	Int	Nom	Int	Int	Int
150	PETALLEN	PETALWID	SPECIES	Can1	Can2	Can3
1	14	2	Setosa	-7.6720	-0.1349	-0.0724
2	56	22	Virginica	6.8002	0.5809	0.7512
3	46	15	Versicolor	2.5487	-0.4722	-0.7839
4	56	24	Virginica	6.6531	1.8053	0.6060
5	51	15	Virginica	3.8152	-0.9430	-0.5106
6	14	3	Setosa	-7.2126	0.3558	0.9248
7	51	23	Virginica	5.1056	1.9922	0.1083
8	45	15	Versicolor	3.4981	-1.6850	-0.1552
9	48	18	Versicolor	3.7159	1.0445	0.9928
10	10	2	Setosa	-8.6810	0.8776	0.7846

Figure 30.7. **CAN_SCOR** Data

By plotting the canonical variables, you can visualize how well the variables discriminate among the three groups. Canonical variables, having more discriminatory power, show more separation among the groups in their associated axes on a plot, while variables having little discriminatory power show little separation among groups.

⇒ Choose **Analyze:Rotating Plot (Z Y X)**. Assign **CAN3** the **Z** role, **CAN2** the **Y** role, and **CAN1** the **X** role.

This produces a plot with the **CAN3** axis pointing toward you, showing clear separation of the species.



Figure 30.8. Rotating Plot Dialog

⇒ Click **OK** in the dialog to create the rotating plot.

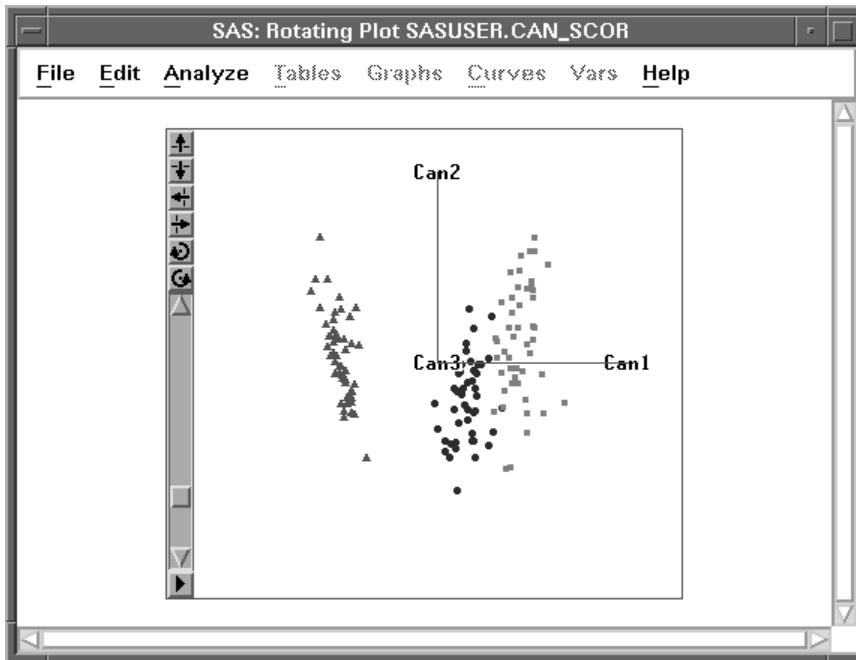


Figure 30.9. Rotating Plot, **CAN3** Toward Viewer

⇒ **Rotate the plot so the axis representing CAN1 points toward you.**

Refer to Chapter 6, “Exploring Data in Three Dimensions,” for information on how to rotate plots. This orientation shows little, if any, differentiation among species. This is because **CAN2** and **CAN3** contribute little information towards separating the groups.

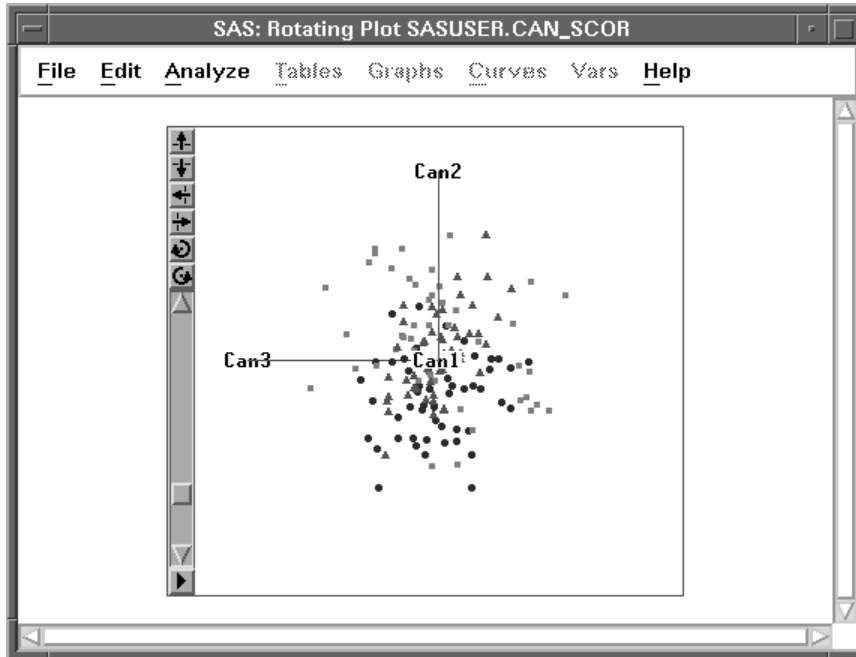


Figure 30.10. Rotating Plot, **CAN1** Toward Viewer

Another way of illustrating this would be to create a scatter plot matrix of **CAN1**, **CAN2**, and **CAN3**. Only plots involving **CAN1** would show much group differentiation. The **CAN2**-by-**CAN3** plot would show little or no group differentiation.

⊕ **Related Reading:** Rotating Plots, Chapter 6, Chapter 37.

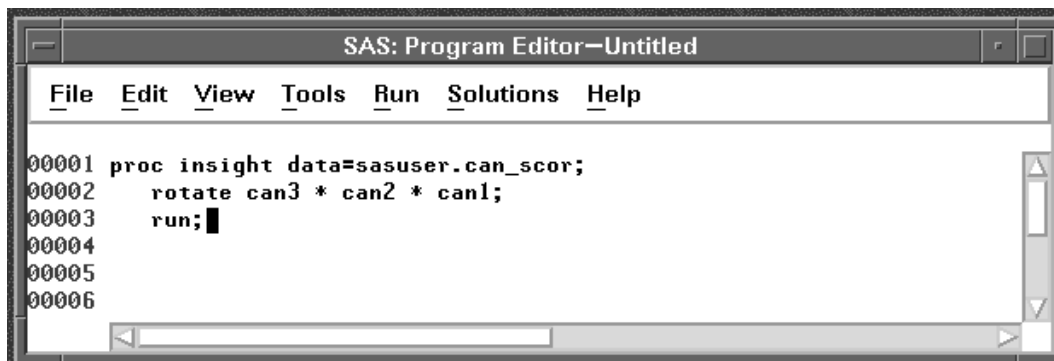
Submitting SAS/INSIGHT Statements

If this analysis were a task you perform frequently, you could save time by automating the creation of the rotating plot. To do this, you can submit SAS/INSIGHT statements in the Program Editor.

You can submit statements when SAS/INSIGHT is executing either as a procedure or as a task. To submit statements to the procedure, do the following.

- ⇒ **Choose File:End in the data window to exit SAS/INSIGHT.**
- ⇒ **In the Program Editor, enter the statements shown in Figure 30.11.**

The DATA option opens the **CAN_SCOR** data set. The ROTATE statement creates the rotating plot.



The screenshot shows a window titled "SAS: Program Editor—Untitled". The menu bar includes File, Edit, View, Tools, Run, Solutions, and Help. The main text area contains the following SAS code:

```
00001 proc insight data=sasuser.can_scor;  
00002     rotate can3 * can2 * can1;  
00003     run;  
00004  
00005  
00006
```

Figure 30.11. SAS/INSIGHT Statements in Program Editor

- ⇒ **Choose Run:Submit to submit the SAS statements.**
This opens the data set and creates the plot.

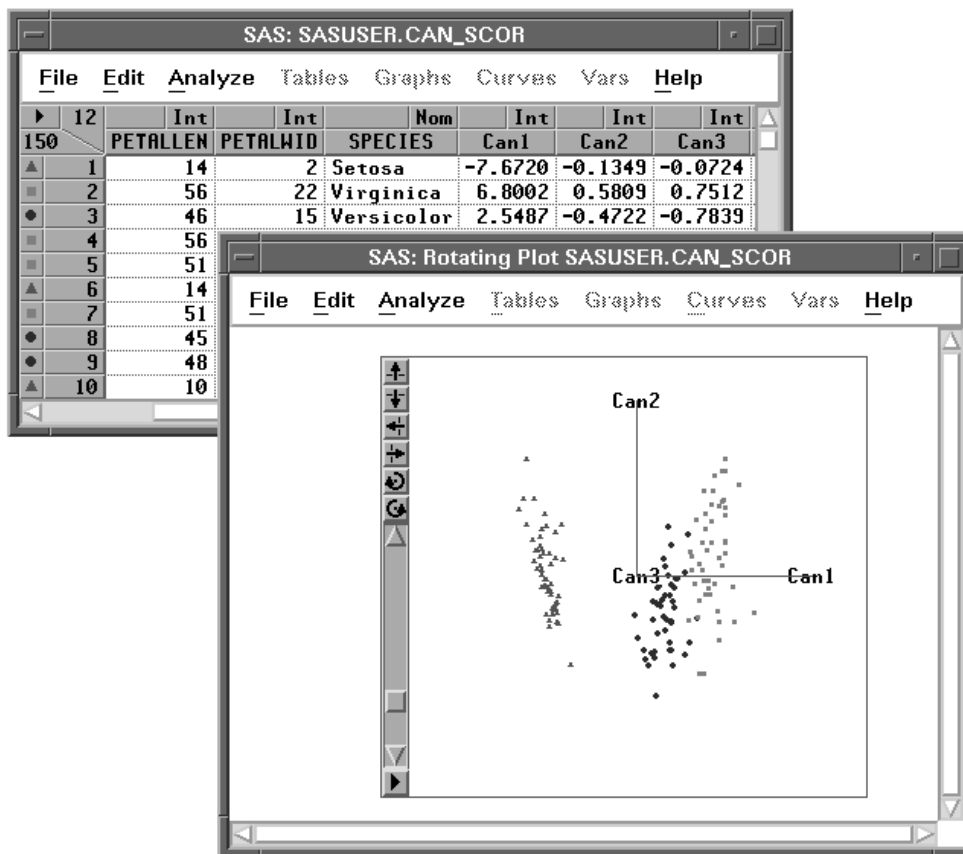


Figure 30.12. Data Window and Rotating Plot

It is often preferable to invoke SAS/INSIGHT as a task instead of a procedure. It is sometimes preferable to open a data set without displaying it. To invoke SAS/INSIGHT as a task and display a rotating plot without a data window, follow these steps.

⇒ Store the following three statements in a text file called **myfile**.

```
open sasuser.can$_$_$scor / nodisplay;
rotate can3 * can2 * can1;
run;
```

⇒ In the Program Editor, enter the **FILENAME** statement shown in Figure 30.13. The **FILENAME** statement assigns a fileref.

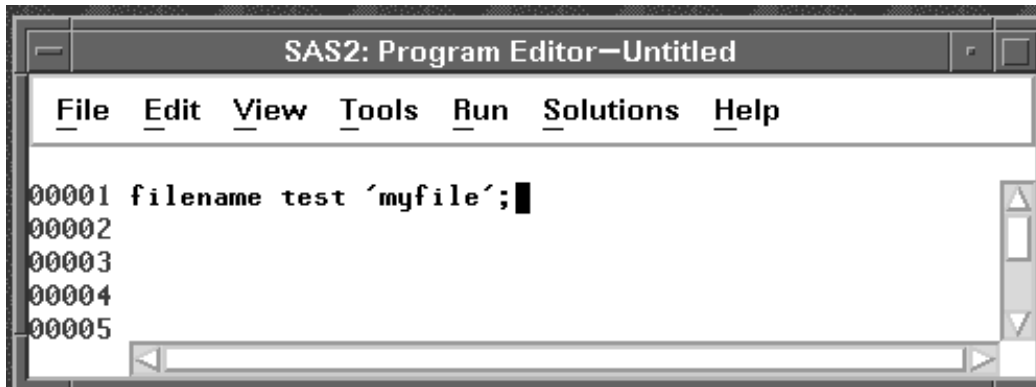


Figure 30.13. Submitting a FILENAME statement

⇒ Choose **Run:Submit** to submit the statement.

⇒ Invoke **SAS/INSIGHT** as a task with the **INFILE=** option.

You can invoke SAS/INSIGHT on the command line with the statement

```
insight infile=test
```

This opens the data set **SASUSER.CAN_SCOR** without displaying it and then creates a rotating plot of **CAN3** versus **CAN2** versus **CAN1**.

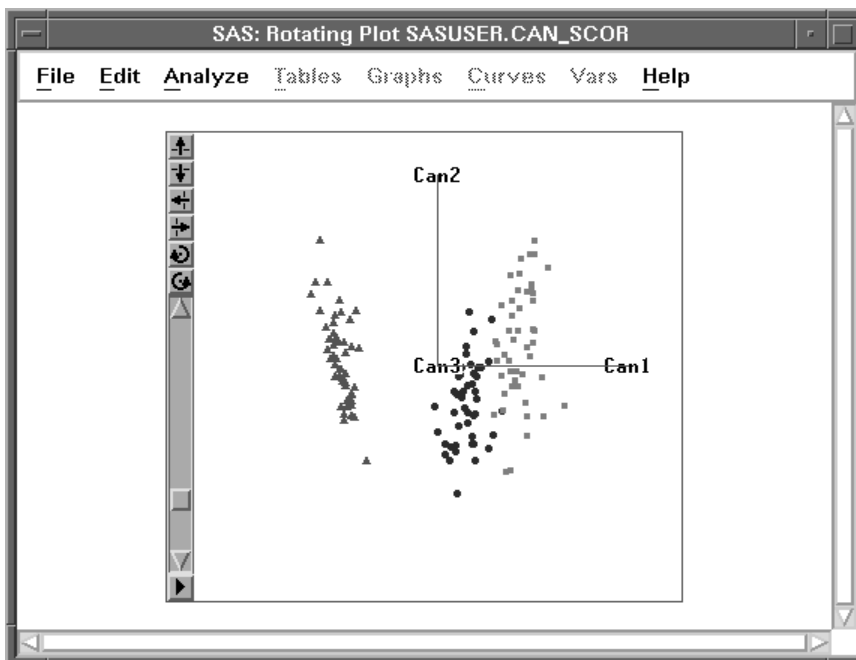


Figure 30.14. Rotating Plot

You can use grammar statements such as these to drive SAS/INSIGHT software from SAS/AF software. For portability, statements can be stored in catalog entries by using a FILENAME statement with the keyword LIBRARY. For example, if you stored statements in a catalog entry **sasuser.insight.test.source**, you could assign the fileref with the statement

```
filename test library 'sasuser.insight.test.source';
```

For SAS/AF applications, you can improve the display of SAS/INSIGHT windows by suppressing the display of menus, buttons, and confirmation dialogs. You can also save options to configure your graphs and analyses. These techniques are described in Chapter 41, “SAS/INSIGHT Statements,” and Chapter 29, “Configuring SAS/INSIGHT Software.”

Recording SAS/INSIGHT Statements

SAS/INSIGHT statements also provide a record of the analyses you create, including model equations. You can record your SAS/INSIGHT session using the **File:Save:Statements** menu or the FILE= option.

To create a record of your SAS/INSIGHT session, follow these steps.

⇒ **Invoke SAS/INSIGHT and open the BUSINESS data set.**

⇒ **Choose File:Save:Statements.**

This toggles the recording of statements to the SAS log.

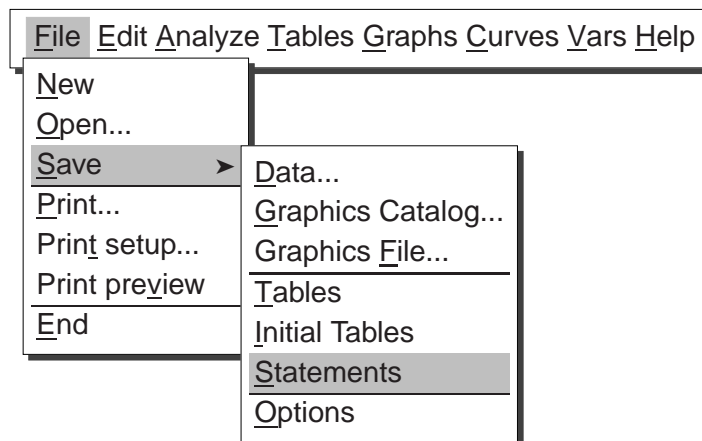


Figure 30.15. File:Save Menu

⇒ **Create graphs and analyses as you like.**

The Log window displays a record of your actions. For example, a record of three model fits might look like the following.

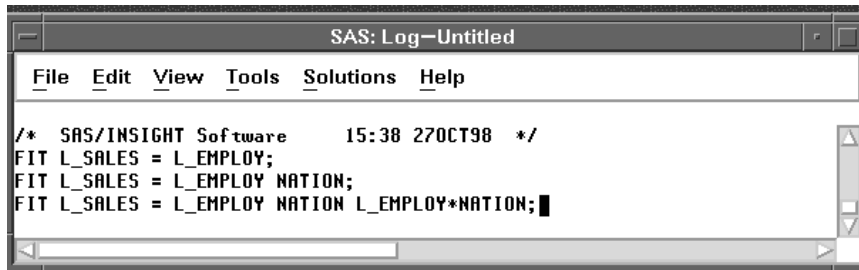


Figure 30.16. Log Window

Recorded output uses the same syntax as statement input, so you can replay the statements you record. However, intermediate events such as transformation of variables, exclusion of observations, and data entry are not recorded. Therefore, replaying will not always reproduce the original analysis.

As an alternative to the **File:Save:Statements** menu, you can use the FILE= option when you invoke SAS/INSIGHT. The FILE= option and other options are described in Chapter 41, “SAS/INSIGHT Statements.”

References

- Fisher, R.A. (1936), “The Use of Multiple Measurements in Taxonomic Problems,” *Annals of Eugenics*, 7, 179–188.
- Robinson, H. (1995), “Batch Processing in SAS/INSIGHT Software,” *Proceedings of the 20th Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc., 560.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/INSIGHT User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999. 752 pp.

SAS/INSIGHT User's Guide, Version 8

Copyright © 1999 by SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-490-X

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

The Institute is a private company devoted to the support and further development of its software and related services.