Chapter 14
Introduction to the FACTEX Procedure

Chapter Table of Contents

OVERVIEW 429
Features 429
Learning about the FACTEX Procedure 430

GETTING STARTED 431
Example of a Two-Level Full Factorial Design 431
Example of a Full Factorial Design in Two Blocks 433
Example of a Half-Fraction Factorial Design 435
Using the FACTEX Procedure Interactively 437

Chapter 14
Introduction to the FACTEX Procedure

Overview

The FACTEX procedure constructs orthogonal factorial experimental designs. These designs can be either full or fractional factorial designs, and they can be with or without blocks. Once you have constructed a design with the FACTEX procedure and run the experiment, you can analyze the results with a variety of SAS procedures including PROC GLM and PROC REG.

Factorial experiments are useful for studying the effects of various factors on a response. Texts that discuss experimental design include Box, Hunter, and Hunter (1978); Cochran and Cox (1957); Hogg and Ledolter (1992); Mason, Gunst, and Hess (1989); and Montgomery (1996). For details on the general mathematical theory of orthogonal factorial designs, refer to Bose (1947).

Note: For two-level designs, instead of using FACTEX directly, a more appropriate tool for you may be the `ADX Interface`. The ADX Interface, which has been completely revised in Version 7, is designed primarily for engineers and researchers who require a point-and-click solution for the entire experimental process, from building the designs through determining significant effects to optimization and reporting. ADX gives you most of the two-level designs provided by the FACTEX procedure in a system that integrates construction and analysis of designs and without the need for programming. In addition to two-level designs for standard models (with and without blocking), ADX makes it easy to use FACTEX to construct designs for estimating particular effects of interest. Morever, ADX also uses the OPTEX procedure to construct two-level designs of non-standard sizes. Information about the ADX Interface can be found at http://www.sas.com/rnd/app/qc/newadx/newadx.html.

Features

There is no inherent limit to the number of factors and the size of the design that you can construct using the FACTEX procedure. Instead of looking up designs in an internal table, the FACTEX procedure uses a general algorithm to search for the construction rules for a specified design.

You can use the FACTEX procedure to generate designs such as the following:

- factorial designs, such as 2^3 designs, with and without blocking
- fractional factorial designs, such as 2_{IV}^{4-1}, with and without blocking
- three-level designs, with and without blocking
- mixed-level factorial designs, such as 4×3 designs, with and without blocking
- randomized complete block design
- factorial designs with outer arrays
- hyper-Graeco Latin square designs

You can also create more complex designs, such as incomplete block designs, by using the FACTEX procedure in conjunction with the DATA step.

You can save the design constructed by the FACTEX procedure in a SAS data set. Once you have run your experiment, you can add the values of the response variable and use the GLM procedure to perform analysis of variance and study significance of effects.

The FACTEX procedure is an interactive procedure. After specifying an initial design, you can submit additional statements without reinvoking the procedure. Once you have constructed a design, you can

- print the design points
- examine the alias structure for the design
- modify the design by changing its size, changing the use of blocking, or respecifying the effects of interest in the model
- output the design to a data set
- examine the confounding rules that generate the design
- randomize the design
- replicate the design
- recode the design from standard values (such as -1 and $+1$) to values appropriate for your situation
- find another design

Learning about the FACTEX Procedure

To learn the basic syntax of the FACTEX procedure, read the “Getting Started” section on page 431, which contains some simple introductory examples. The summary tables in the “Summary of Functions” section on page 442 provide an overview of the syntax. The “Summary of Designs” section on page 444 shows simple ways to construct full factorial designs and fractional factorial designs. The “Advanced Examples” section on page 457 illustrate construction of complex designs.

Getting Started

The following introductory examples illustrate the capabilities of the FACTEX procedure. See “Advanced Examples” on page 457 for illustrations of complex features.

Example of a Two-Level Full Factorial Design

This example FACTEXG1 introduces the basic syntax used with the FACTEX procedure.

An experimenter is interested in studying the effects of three factors—cutting speed (SPEED), feed rate (FEED), and tool angle (ANGLE)—on the surface finish of a metallic part. Two levels of each factor are chosen (shown below), and the experimenter decides to run a complete factorial experiment.

Factor	Low Level	High Level
Cutting Speed	300	500
Feed Rate	20	30
Tool Angle	6	8

This is a 2^3 factorial design—in other words, a complete factorial experiment with three factors, each at two levels. Hence there are eight runs in the experiment. Since complete factorial designs have full resolution, all of the main effects and interaction terms can be estimated. For a definition of the design resolution, see page 491.

You can use the following statements to create the required design:

```
proc factex;                        /* Invoke the FACTEX procedure */
   factors speed feed angle;        /* List factor names           */
   examine design;                  /* Display coded design points */
run;
```

By default, the FACTEX procedure assumes that the size of the design is a full factorial and that each factor has only two levels.

After you submit the preceding statements, you will see the following messages in the SAS log:

```
NOTE: No design size specified.
      Default is a full replicate in 8 runs.
NOTE: Design has 8 runs, full resolution.
```

The output is shown in Figure 14.1. The two factor levels are represented by the coded values 1 and -1.

```
                           Design Points

                Experiment
                    Number     speed     feed     angle
                ---------------------------------------
                         1        -1       -1        -1
                         2        -1       -1         1
                         3        -1        1        -1
                         4        -1        1         1
                         5         1       -1        -1
                         6         1       -1         1
                         7         1        1        -1
                         8         1        1         1
```

Figure 14.1. 2^3 Factorial Design

If you prefer to work with the actual (decoded) values of the factors, you can specify these values in an OUTPUT OUT= statement, as follows:

```
proc factex;
   factors speed feed angle;
   output out=savdesgn             /* Recode factor levels and  */
          speed nvals=(300 500)    /*   save design in SAVDESGN */
          feed  nvals=(20  30 )
          angle nvals=(6   8  );
proc print;
run;
```

Since the levels in this example are of numeric type, you use the NVALS= option to list the factor levels. Optionally, you can use the CVALS= option for levels of character type (see page 434 for an example). The design is saved in a user-specified output data set (SAVDESGN). This is verified by the following message in the SAS log:

```
NOTE: The data set WORK.SAVDESGN has 8 observations
      and 3 variables.
```

Figure 14.2 shows a listing of the data set SAVDESGN.

```
                         Obs    speed    feed    angle

                          1      300      20       6
                          2      300      20       8
                          3      300      30       6
                          4      300      30       8
                          5      500      20       6
                          6      500      20       8
                          7      500      30       6
                          8      500      30       8
```

Figure 14.2. 2^3 Factorial Design after Decoding

Although small complete factorial designs are not difficult to create manually, you can easily extend this example to construct a design with many factors.

Example of a Full Factorial Design in Two Blocks

See FACTEXG2 in the SAS/QC Sample Library

The previous example illustrates a complete factorial experiment involving eight runs and three factors: cutting speed (SPEED), feed rate (FEED), and tool angle (ANGLE).

Now, suppose two machines (A and B) are used to complete the experiment, with four runs being performed on each machine. To allow for the possibility that the machine affects the part finish, you should consider machine as a block factor and account for the block effect in assigning the runs to machines.

The following statements construct a blocked design:

```
proc factex;
    factors speed feed angle;  /* Specify 3 factors, a 2^3 design   */
    blocks nblocks=2;          /* Specify two blocks                */
    model resolution=max;      /* Specify maximum resolution        */
    examine design;
run;
```

The RESOLUTION=MAX option in the MODEL statement specifies a design with the highest resolution, that is, the best design in a general sense. Optionally, if you know the resolution of the design, you can replace RESOLUTION=MAX with RESOLUTION=r where r is the resolution number. For information on resolution, see page 491.

By default, the FACTEX procedure assumes the size of the design is a full factorial and that each factor is at two levels.

After you submit the preceding statements, you will see the following messages in the SAS log:

```
NOTE: No design size specified.
      Default is a full replicate in 8 runs.
NOTE: Design has 8 runs in 2 blocks of size 4,
      resolution = 6.
```

The output is shown in Figure 14.3. Note that, by default, the name for the block variable is BLOCK and its levels are 1 and 2. Also, note that the default factor levels for a two-level design are -1 and 1.

```
                          Design Points

            Experiment
               Number     speed     feed     angle     Block
            ------------------------------------------------
                    1        -1       -1        -1         1
                    2        -1       -1         1         2
                    3        -1        1        -1         2
                    4        -1        1         1         1
                    5         1       -1        -1         2
                    6         1       -1         1         1
                    7         1        1        -1         1
                    8         1        1         1         2
```

Figure 14.3. 2^3 Factorial Design in Two Blocks before Decoding

You can rename the block variable and use actual levels for the block variable appropriate for your situation as follows:

```
proc factex;
   factors speed feed angle;
   blocks nblocks=2;
   model resolution=max;
   output out=blocdesn
     speed nvals=(300 500)
     feed  nvals=(20  30 )
     angle nvals=(6   8  )
     blockname=machine cvals=('A' 'B');
run;

proc print;
run;
```

Figure 14.4 shows the listing of the design saved in the data set BLOCDESN.

```
                 Obs    machine    speed    feed    angle

                  1        A        300      20       6
                  2        A        300      30       8
                  3        A        500      20       8
                  4        A        500      30       6
                  5        B        300      20       8
                  6        B        300      30       6
                  7        B        500      20       6
                  8        B        500      30       8
```

Figure 14.4. 2^3 Factorial Design in Two Blocks after Decoding

Example of a Half-Fraction Factorial Design

See FACTEX3 in the SAS/QC Sample Library

Often you do not have the resources for a full factorial design. In this case, a fractional factorial design is a reasonable alternative, provided that the effects of interest can be estimated.

Box, Hunter, and Hunter (1978) describe a fractional factorial design for studying a chemical reaction to determine what percentage of the chemicals responded in a reactor. The researchers identified the following five treatment factors that were thought to influence the percentage of reactant:

- the feed rate of the chemicals (FEEDRATE), ranging from 10 to 15 liters per minute
- the percentage of the catalyst (CATALYST), ranging from 1% to 2%
- the agitation rate of the reactor (AGITRATE), ranging from 100 to 120 revolutions per minute
- the temperature (TEMPERAT), ranging from 140 to 180 degrees Centigrade
- the concentration (CONCENTN), ranging from 3% to 6%

The complete 2^5 factorial design requires 32 runs, but it was decided to use a half-fraction design, which requires 16 runs.

Suppose that all main effects and two-factor interactions are to be estimated. An appropriate design for this situation is a design of resolution 5 (denoted as 2^{5-1}_{V}), in which no main effect or two-factor interaction is aliased with any other main effect or two-factor interaction but in which two-factor interactions are aliased with three-factor interactions. This design loses the ability to estimate interactions between three or more factors, but this is usually not a serious loss. For more on resolution, see page 491.

You can use the following statements to construct a 16-run factorial design that has five factors and resolution 5:

```
proc factex;
   factors feedrate catalyst agitrate temperat concentn;
   size design=16;
   model resolution=5;
   output out=reaction feedrate  nvals=(10  15 )
                       catalyst  nvals=(1   2  )
                       agitrate  nvals=(100 120)
                       temperat  nvals=(140 180)
                       concentn  nvals=(3   6  );
proc print;
run;
```

The design saved in the REACTION data set is listed in Figure 14.5.

```
       Obs    feedrate    catalyst    agitrate    temperat    concentn

         1       10           1          100         140          6
         2       10           1          100         180          3
         3       10           1          120         140          3
         4       10           1          120         180          6
         5       10           2          100         140          3
         6       10           2          100         180          6
         7       10           2          120         140          6
         8       10           2          120         180          3
         9       15           1          100         140          3
        10       15           1          100         180          6
        11       15           1          120         140          6
        12       15           1          120         180          3
        13       15           2          100         140          6
        14       15           2          100         180          3
        15       15           2          120         140          3
        16       15           2          120         180          6
```

Figure 14.5. Half-Fraction of a 2^5 Design for Reactors

The use of a half-fraction causes some interaction terms to be confounded with each other. You can use the EXAMINE statement with the ALIASING option to determine which interaction terms are aliased, as follows:

```
proc factex;
   factors feedrate catalyst agitrate temperat concentn;
   size design=16;
   model resolution=5;
   examine aliasing;
run;
```

The alias structure summarizes the estimability of all main effects and two- and three-factor interactions. Figure 14.6 indicates that each of the three-factor interactions is confounded with a two-factor interaction. Thus, if a particular three-factor interaction is believed to be significant, the aliased two-factor interaction cannot be estimated with this half-fraction design.

```
Aliasing Structure

 feedrate
 catalyst
 agitrate
 temperat
 concentn
 feedrate*catalyst = agitrate*temperat*concentn
 feedrate*agitrate = catalyst*temperat*concentn
 feedrate*temperat = catalyst*agitrate*concentn
 feedrate*concentn = catalyst*agitrate*temperat
 catalyst*agitrate = feedrate*temperat*concentn
 catalyst*temperat = feedrate*agitrate*concentn
 catalyst*concentn = feedrate*agitrate*temperat
 agitrate*temperat = feedrate*catalyst*concentn
 agitrate*concentn = feedrate*catalyst*temperat
 temperat*concentn = feedrate*catalyst*agitrate
```

Figure 14.6. Alias Structure of Reactor Design

When you submit the preceding statements, the following message is displayed in the SAS log:

```
NOTE: Design has 16 runs, resolution = 5.
```

This message confirms that the design exists. If you specify a factorial design that does *not* exist, an error message is displayed in the SAS log. For instance, suppose that you replaced the MODEL statement in the preceding example with the following statement:

```
      model resolution=6;
```

Since the maximum resolution of a 2^{5-1} design is 5, the following message appears in the SAS log:

```
ERROR: No such design exists.
```

In general, it is good practice to check the SAS log to see if a design exists.

Using the FACTEX Procedure Interactively

By using the FACTEX procedure interactively, you can quickly explore many design possibilities. The following steps provide one strategy for interactive use:

1. Invoke the procedure with the PROC FACTEX statement, and use a FACTORS statement to identify factors in the design.
2. For a design that involves blocking, use the BLOCKS and MODEL statements. You may want to use the optimization features for the BLOCKS statement.
3. For a fractional replicate of a design, use the SIZE and MODEL statements to specify the characteristics of the design. If the design involves blocking, use a BLOCKS statement as well. If you are unsure of the size of the design or of the number of blocks, use the optimization features for either the BLOCKS or SIZE statement.
4. Enter a RUN statement and check the SAS log to see if the design exists. If a design exists, go on to the next step; otherwise, modify the characteristics given in the SIZE, BLOCKS, and MODEL statements.
5. Examine the alias structure of the design. If it is not appropriate for your situation, go back to step 2 and search for another design.
6. After you have repeated steps 2, 3, and 4 and found an acceptable design, use the OUTPUT statement to save the design. You can optionally recode factor values, recode and rename the block factor, and create new factors using output-value-settings.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/QC® User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999. 1994 pp.

SAS/QC® User's Guide, Version 8

ISBN 1–58025–493–4

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

1st printing, October 1999

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