SIMULATION STUDY IN STATISTICS – Draft

Erning Li, Dennis Boos, and Marcia Gumpertz

Department of Statistics

NCSU

October 2, 2001

1 Introduction

Simulation is a numerical technique for conducting experiments on a computer. Computer simulation is used in engineering and the sciences to rerun physical experiments with selected changes in parameters or operating conditions. In statistics simulation experiments are most often used to study properties of statistical methods. Many real statistical processes can be simulated on a computer with the aid of random numbers. Monte Carlo simulations in statistics are computer experiments involving random sampling from probablility distributions to study properties of statistical methods. Here, we will focus on simulations using Monte Carlo methods.

Simulation is an invaluable and versatile tool in those problems where analytic techniques are inadequate. For example, in regression with correlated normally distributed errors, the properties of the generalized least squares estimator are easy to derive when the covariance matrix $\mathrm{Var}(y)$ is known, but what happens if $\mathrm{Var}(y)$ is not known and we substitute estimates of Var(y) into the GLS equations? Large sample properties can be derived for the case when the sample size tends to infinity, but the only way to get information about the behavior of the estimator for small samples is to do a simulation study.

2 How to Do Simulations: Survival Guide for Monte Carlo Studies by Dr. Dennis Boos

Monte Carlo methods in statistics usually refer to the computation of expected values by computer simulation. These expected values are often needed to find

- the bias and variance of estimators,
- the percentiles of test statistics and pivotal quantities,
- the power functions of hypothesis tests,
- the average length and coverage probabilities of confidence intervals.

In computing these quantities, typically a sample of n random variables $X_1, \ldots, X_n$ is generated and an estimator or test statistic $T = T(X_1, \ldots, X_n)$ is computed. This process is replicated N independent times resulting in a sample $T_1, \ldots, T_N$. At this point we just perform routine data analysis on the sample of T's. Often we compute several estimators for each sample. Then the comparison of estimators can take advantage of the natural blocking induced by the samples.

Example 1. A very simple study would be to compare the sample mean $\bar{X}$, the sample 20% trimmed mean $\bar{X}_{.20}$, and the sample median M as estimators of location for a variety of sample sizes and distributions. Here we may be interested in comparing small sample biases, variances, and mean square errors. A complicating aspect for skewed distributions is that the estimators are not estimating the same quantities.

Example 2. Another simple example would be to analyze the maximum likelihood estimator for a vector parameter θ in a parametric setting, such as a two-parameter Weibull distribution. Here we may be interested in the small sample bias of the parameter estimator $\hat{\theta}$, the covariance matrix of $\hat{\theta}$, and the small sample bias of the estimated Fisher information matrix and of the observed information matrix as estimators of the covariance matrix of $\hat{\theta}$.

2.1 Basic principles

Several principles in the design and analysis of Monte Carlo simulation studies are

Principle 1: A Monte Carlo experiment is just like any other experiment. It requires careful planning in addition to carrying out the experiment and analyzing the results.

There is no harm in jumping in and making some preliminary runs to get a feel for what the results will look like. But for a serious study to be later shared with others, at some point we should carefully think about

- the factors that will be varied during the experiment such as the sample size n and the distributions used to generate the data. (Keep in mind that the number of runs required to have every combination of factor levels in a full factorial can grow very large.)

- the Monte Carlo sample size N required to obtain the desired precision but also small enough to fit time and machine constraints. (A useful exercise is to think about how hard it would be to repeat the study if a minor modification is desired later.)

- the raw and summary statistics that will be printed out or saved in a file.

Principle 2: Save the individual T values (values inside the main loop) in addition to the summary statistics, whenever possible.

Principle 2 is crucial if you are making long runs that are time consuming to replicate. For example, if you decided to analyze data on the log scale after running your experiment, then having saved the individual T values would obviate the need to rerun the experiment. In fact it is often easier to save the individual T's in a file and analyze that file with SAS or S-plus rather than put everything in the basic Monte Carlo run.

Principle 3: Keep the Monte Carlo sample size N small at first until you are sure you know where you are headed.

This way you can keep perfecting your basic Monte Carlo program until you have everything in place. Then you can make the "production" runs that will appear in the final paper. Be careful of course not to search around for situations that are only favorable to your new method. Monte Carlo studies by statisticians are just as susceptible to bias as other studies.

Principle 4: Set a different seed for each run and keep records of the seeds so that

- the runs are independent,
- each run may be repeated when necessary.

Principle 5: Document the computer code so that

- you can understand what you did sometime later,
- someone else can understand and check your code.

2.2 Monte Carlo sample size

Determining the sample size in a Monte Carlo study is no different than for other types of studies. Basically we need to determine the acceptable error of estimates or power of tests and invert analytical functions of sample size set equal to those values. Here are a few examples.

Basic Estimation. Suppose that we will compute the parameter estimate $\hat{\theta}$ for N independent replications resulting in $\hat{\theta}_1, \ldots, \hat{\theta}_N$. For given true value θ_0, our estimate of bias will be $N^{-1} \sum \hat{\theta}_i - \theta_0$ with standard deviation $\mathrm{sd}(\hat{\theta})/\sqrt{N}$. So given a guess of $\mathrm{sd}(\hat{\theta})$, say $\tilde{\sigma}$, and an acceptable value, say d, for the standard deviation of our estimate, we solve to get $N = \tilde{\sigma}^2/d^2$. To compare the estimates of bias of two different estimators, say $\hat{\theta}^{(1)}$ and $\hat{\theta}^{(2)}$, using the same Monte Carlo data, then we should take into account the blocking induced and use $\mathrm{sd}(\hat{\theta}^{(1)} - \hat{\theta}^{(2)})/\sqrt{N}$.

Variance Estimation. When studying a new estimator $\hat{\theta}$, we will usually want to estimate its variance with the sample variance.

$$s_{N-1}^2 = \frac{1}{N-1} \sum_{i=1}^{N} (\hat{\theta}_i - \bar{\hat{\theta}})^2.$$

For large N, the approximate variance of the sample variance will be close to $\sigma_n^4(\beta_2 - 1)/N$, where $\sigma_n^2 = \mathrm{var}(\hat{\theta})$ and β_2 is the kurtosis of the distribution of $\hat{\theta}$. Since most estimators are approximately normal, we may assume that $\beta_2 \approx 3$, leading to $\mathrm{var}(s_{N-1}^2) = 2\sigma_n^4/N$. So the approximate standard deviation of our variance estimate is $\sqrt{2/N}\sigma_n^2$. For acceptable d, we invert to get $N = 2\sigma_n^4/d^2$.

If we prefer to think in terms of relative error, letting d_p be the acceptable proportional error, we have that the approximate standard deviation of s_N^2/σ^2 is $\sqrt{2/N}$, leading to $N = 2/d_p^2$. For example, if $d_p = .10$, we get $N = 2/.01 = 200$.

If we consider the sample standard deviation s_{N-1} instead of the sample variance s_{N-1}^2, then the approximate standard deviation of s_{N-1} is $\sigma_n/\sqrt{2N}$ (by the delta theorem). Thus, if d_s is an acceptable standard deviation for s_{N-1}, then $N = \sigma_n^2/(2d_s^2)$. If d_{sp} is an acceptable proportional error, then $N = 1/(2d_{sp}^2)$.

Power Estimation. For a new test procedure of the form "reject the null hypothesis if $T > c_\alpha$," we often estimate the power at a particular alternative by

$$\widehat{power} = \frac{1}{N}\sum_{i=1}^{N} I(T_i > c_\alpha),$$

where T_i is the test statistic for the ith Monte Carlo sample, c_α is a given critical value, and I is the indicator function having value 1 if $T_i > c_\alpha$ and 0 otherwise.

This is binomial sampling, and the worst variance of our estimate (occuring at $power = 1/2$) is giving by $1/(4N)$. Setting $d = 1/\sqrt{4N}$ yields $N = 1/(4d^2)$. For $d = .05$ we get $N = 100$, but for $d = .01$ we get $N = 2500$. If we want to compare the power of two test statistics, then we should take into account the blocking induced by computing both procedures on the same Monte Carlo samples. This could be accomplished by estimating the variance of

$$I(T^{(1)} > c_\alpha^{(1)}) - I(T^{(2)} > c_\alpha^{(2)})$$

in some preliminary runs.

In some cases, we may have a critical value c_α from asymptotic theory but would like to estimate the "correct" c_α and use it in the power estimate. This can be important when comparing two test procedures that have different true significance levels. In this case, Boos and Zhang (1994) suggest using an N for estimating the critical value that is 10 times the size of the N used for estimating the power at specific alternatives.

3 How to Analyze and Report Results

3.1 How to evaluate an estimator

(1) Estimating bias

Suppose we want to estimate a population parameter θ by the statistic $\hat{\theta}$. A key result to report is the bias of the estimator $\hat{\theta}$, defined by

$$BIAS = E(\hat{\theta}) - \theta,$$

where $E()$ denotes the expected value of whatever is in the brackets.

As discussed in section 2.2, the bias from a Monte Carlo simulation can be estimated as

$$\widehat{BIAS} = \frac{1}{N}\sum_{i=1}^{N}\hat{\theta}_i - \theta = \bar{\hat{\theta}} - \theta.$$

(2) Variance of an estimator

It is also important to study the variance of the estimator, which is a measure of how much the distribution of a statistic is spread from the mean:

$$\text{var}(\hat{\theta}) = E\left[\left(\hat{\theta} - E(\hat{\theta})\right)^2\right].$$

As discussed in section 2.2, the variance of an estimator from a Monte Carlo simulation can be estimated as

$$s_{\hat{\theta}}^2 = \widehat{\text{var}(\hat{\theta})} = \frac{1}{N-1}\sum_{i=1}^{N}(\hat{\theta}_i - \bar{\hat{\theta}})^2.$$

(3) Relative efficiency

Relative efficiency is a measure of the variability of a new estimator compared to another. Suppose two methods exist for estimating θ. Let $\hat{\theta}^{(1)}$ and $\hat{\theta}^{(2)}$ be two unbiased estimators produced by these methods such that

$$E(\hat{\theta}^{(1)}) = E(\hat{\theta}^{(2)}) = \theta.$$

The first method is more efficient than the second method if the relative efficiency

$$RE = \frac{\text{var}(\hat{\theta}^{(1)})}{\text{var}(\hat{\theta}^{(2)})} < 1.$$

Relative efficiency may be estimated from a Monte Carlo simulation by substituting in the empirical variance of each estimator:

$$\widehat{RE} = \frac{s^2_{\hat{\theta}^{(1)}}}{s^2_{\hat{\theta}^{(2)}}},$$

where $s^2_{\hat{\theta}^{(i)}}, i = 1, 2$, are defined above.

(4) Mean square error

The mean square error (MSE) is the expected squared deviation between the sample mean $\hat{\theta}$ and the parameter θ it is supposed to estimate. The MSE summarizes all the information about bias (accuracy) and variance of the estimator (precision) under study. In case only one measure of the precision and accuracy of an estimator can be reported, the MSE is the measure to report.

$$MSE = \text{var}(\hat{\theta}) + (BIAS)^2$$

We can estimate MSE by taking, say, N samples (of the same size), determining for each sample the statistic $\hat{\theta}_i (i = 1, 2, \ldots, N)$ assuming we know θ, the population mean.

$$\widehat{MSE} = \frac{1}{N-1} \sum_{i=1}^{N} (\hat{\theta}_i - \bar{\hat{\theta}})^2 + (\bar{\hat{\theta}} - \theta)^2$$

3.2 How to evaluate a testing procedure

(1) Level or size

Incorrect rejection of a null hypothesis when the null hypothesis (H_0) is in fact true is called a Type I error. The probability of a Type I error is symbolized by α and is referred to as the significance level or size of the test.

To test a hypothesis, we need to choose a probability α based on the seriousness of rejecting H_0 when H_0 is true. The usual significance level people use is $\alpha = .05$.

(2) Power

The general definition of the power of a hypothesis test is the probability of rejecting the null hypothesis H_0 when the alternative hypothesis H_a is in fact true.

$$power = P(\text{reject}H_0|H_0\text{is false})$$

The power equals the significance level α when the null hypothesis is true. Generally, the power of a hypothesis test increases as sample size increases.

As discussed in section 2.2, the power of a hypothesis test from a Monte Carlo simulation can be estimated as

$$\widehat{power} = \frac{1}{N}\sum_{i=1}^{N} I(T_i > c_\alpha),$$

where T_i is the test statistic for the ith Monte Carlo sample, c_α is a given critical value, and I is the indicator function having value 1 if $T_i > c_\alpha$ and 0 otherwise.

4 Monte Carlo Sampling

An all-purpose method for generating a random sample $X_1, \ldots, X_n$, where $P(X_i \leq x)$, is to first generate a random sample of Uniform$(0,1)$ random variables $U_1, \ldots, U_n$. Then form $X_1 = F^{-1}(U_1), \ldots, X_n = F^{-1}(U_n)$, where $F^{-1}(u)$ is the inverse of $F(x)$. This follows

$$P\left(F^{-1}(U_1) \leq x\right) = P\left(U_1 \leq F(x)\right) = F(x)$$

(see page 280, Monahan, 2001). Methods for generating Uniform$(0,1)$ random variables are given in Monahan (2001, Section 10.5), and other methods of generating random variables from $F(x)$ are given in Monahan (2000, Chapter 11). In these notes we will not worry about how to generate random samples since all major statistical packages like SAS and S-plus have random number generators.

5 Computing issues

It is often helpful to use matrix and built-in functions and procedures to make the simulation much faster and more error proof. For example, we can write modules in IML, functions in S-plus, and use prewritten functions for problems such as nonlinear optimization, numerical integration, etc.

Sometimes it is useful to divide dataset and repeat datastep or procedures for sub-datasets if necessary in order to have a large number of Monte Carlo replicates N.

Consult URL *http://www.stat.ncsu.edu/sas/faq/batch.html* or *SAS Companion for the UNIX Environment and Derivatives* for information on how to execute jobs in batch mode, increase memory, use the batch to submit jobs to be executed later, and other information about running SAS in the UNIX environment at NCSU.

Here, we are going to use SAS to demonstrate an example and to do several exercises. However, for time efficient computing, some general-purpose computer languages such as FORTRAN and C can be better choices.

6 An Example

To illustrate the steps in designing and carrying out a simulation study, an example is given below, including the process of choosing factors, levels of factors, simulation procedures, documenting and presenting results.

This example is a Monte Carlo simulation study investigating the effects of nonnormality, nonconstant variance, and imbalanced sample size on robustness of the F-test for treatment effects in the analysis of variance. It is commonly asserted that the F-test is robust to nonnormality but not as robust to nonconstant variance. We desire to understand these claims better and determine whether the robustness results are the same if the sample sizes for different treatments are not equal.

6.1 The objectives of the simulation study

For simplicity we will consider the case of two treatments.

The main objective of the study is to investigate whether the rejection rate of the test of equal treatment effects is maintained at the α level when H_0 is true (no treatment effects) if

1) y is not normally distributed,

2) variances are heterogeneous,

3) sample sizes are imbalanced.

We are interested in investigating how rejection rates changes as the total sample size gets larger.

We are also interested in studying whether the effect of nonnormality on the rejection rates depends on variance heterogeneity or sample size imbalance or both, and whether the effect of non-constant variance is the same for balanced and imbalanced sample sizes.

6.2 Experimental Design Issues

Enough factors and combinations of factors need to be chosen so that the questions in the last section can be answered. This will often lead to a factorial structure. In this example, since the effects of nonnormality, nonconstant variance, unequal sample sizes, and total sample size are our main interests, we take error distributional type, heterogeneity of variances, sample size imbalance, and total sample size as the four factors in the experiment we are going to study.

(1) Levels of factors

For a full factorial with four factors that have a, b, c and d levels, respectively, $a \times b \times c \times d$ combinations of factor levels would be needed in the experiment. If we choose many levels for factors, it would be time consuming to run an experiment. In this experiment, the four factors are taken at their minimum number of levels, i.e., two levels, in order to limit the number of factor-level combinations and to clearly and easily interpret the results to be obtained. Thus, if we use a full factorial with the four factors, we will have to run sixteen combinations of factor levels.

When there are many factors involved in an experiment, the number of combinations of factor levels in a full factorial can be extensively large. We may need to consider a fractional factorial structure.

Levels of error distribution factor. For the two levels of the error distribution factor, we choose the normal distribution and a nonnormal distribution. Since the lognormal distribution is one of the most common distributions for real data, especially in biological sciences, we choose the lognormal distribution as the other level of the error distribution factor.

Levels of variance heterogeneity factor. Since the normal and lognormal distributions are commonly used, we can directly choose their parameters according to statistical theory.

Suppose $Z \sim \text{Normal}(\mu, \sigma^2)$, so $E(Z) = \mu$, and $\text{var}(Z) = \sigma^2$.

- For the normal distribution, define $Y = Z \sim \text{Normal}(\mu, \sigma^2)$. Then

$$E(Y) = \mu, \qquad \text{var}(Y) = \sigma^2.$$

- For the lognormal distribution, define $Y = e^Z \sim \text{Lognormal}(\mu, \sigma^2)$, i.e. $log(Y) = Z \sim \text{Normal}(\mu, \sigma^2)$. We have (from the moment generating function of Z)

$$E(Y) = E(e^Z) = e^{\mu+\sigma^2/2}, \qquad \text{var}(Y) = e^{2\mu+2\sigma^2} - e^{2\mu+\sigma^2} = e^{2\mu}(e^{2\sigma^2} - e^{\sigma^2}).$$

Let subscripts 1 and 2 denote the parameters for the samples for treatments 1 and 2, respectively. We choose $\text{var}(Y_1) : \text{var}(Y_2) = 1 : 1$ for the constant variance case and $\text{var}(Y_1) : \text{var}(Y_2) = 1 : 100$ for the nonconstant variance case. We set the expected mean value equal to 0 for all cases so that there are no differences among the means of the distributions.

Notice that if $E(Y) = 0$ and $\text{var}(Y) = 1$, then $E(10Y) = 0$ and $\text{var}(10Y) = 100\text{var}(Y) = 100$.

1) *Normal distribution*

- For the constant variance case, let both Y_1 and Y_2 follow a Normal$(0, 1)$ distribution,

$$E(Y_1) = E(Y_2) = 0, \qquad \text{var}(Y_1) = \text{var}(Y_2) = 1.$$

- For the nonconstant variance case, let $Y_1 \sim$ Normal$(0, 1)$,

$$E(Y_1) = 0, \qquad \text{var}(Y_1) = 1.$$

Let $Y_2 = 10X$ where $X \sim$ Normal$(0, 1)$, then $Y_2 \sim$ Normal$(0, 100)$.

2) *Lognormal distribution*

- For the constant variance case, let both Y_1 and Y_2 follow Lognormal$(0, \sigma^2)$ (here $\mu = 0$) distributions. To have variance equal to 1, we need

$$\text{var}(Y_1) = \text{var}(Y_2) = e^{2\sigma^2} - e^{\sigma^2} = 1.$$

The solution to the above equation is $\sigma^2 = 0.4812$; thus correspondingly

$$E(Y_1) = E(Y_2) = e^{0.4812/2} = 1.27201.$$

To have expected mean value equals 0, we use $Y_1 - 1.27201$ and $Y_2 - 1.27201$ where both Y_1 and Y_2 have Lognormal$(0, 0.4812)$ distributions.

Notice that the skewness and kurtosis of a Lognormal(μ, σ^2) distribution depends on the value of σ^2 while they are free of μ. To keep the two nonconstant variance lognormal distributions to have the same skewness and kurtosis, the same value of σ^2 should be used.

- For the nonconstant variance case, we use $Y_1 - 1.27201$ where $Y_1 \sim$ Lognormal$(0, 0.4812)$ for the first treatment group and $Y_2 - 12.7201 = 10(X - 1.27201)$ where $Y_2 = 10X$ and $X \sim$ Lognormal$(0, 0.4812)$ for the second treatment group.

Levels of sample size imbalance factor. To choose the two levels for sample size imbalance, small preliminary experiments can be used to determine sample size ratios. Pilot studies are always helpful for planning experiments.

Table 1: **Rejection Rates at Different Sample Size Ratios**

$n_1 : n_2$		35:5	30:10	25:15	20:20
Lognormal	Heterogeneous	0.536	0.314	0.169	0.082
	Homogeneous	0.049	0.048	0.048	0.040
Normal	Heterogeneous	0.506	0.275	0.130	0.057
	Homogeneous	0.067	0.050	0.049	0.053

For all cases, $\alpha = 0.05, E(Y) = 0, N = 1000, seed = 289739$ for normal, $seed = 983681$ for lognormal.

Table 1 shows the relationship between the response, rejection rate and sample size ratio for imbalanced conditions when the total sample size is fixed at $n_1 + n_2 = 40$ and the larger variance is in the group with the smaller sample size. This is known to cause more problem than the reverse.

These results show that rejection rates tend to be close to α as the two sample sizes become similar, which is reasonable since the estimate $2 \cdot MSE/n = 2\frac{SSE}{n(n_1+n_2-2)}$ gets closer to the variance $\text{var}(\bar{Y_1} - \bar{Y_2}) = \frac{MSE_1}{n_1} + \frac{MSE_2}{n_2}$ when n_1 and n_2 are similar.

In the experiment, we will use the ratio $n_1 : n_2 = 1$ for the balanced case and $n_1 : n_2 = 3$ for the imbalanced case.

Levels of total sample size factor. The precision of the estimated parameters is affected by the sample size. This precision can be evaluated by the standard deviation of the estimated parameters for simulated samples. If a new set of random numbers for each run is used, each run yields an independent parameter estimate so traditional statistical techniques can be used to analyze the results of the simulation experiment.

In this example, $n_1 = n_2 = n$ for balanced sample sizes and $n_1 \neq n_2$ for imbalanced sample sizes. In order to make the simulated data balanced for each treatment and for each combination of factor levels, we also need to make sure that the total sample size in the two treatments satisfies $n_1 + n_2 = 2n$.

In the experiment, we will use $n_1 = n_2 = n = 10$ for balanced sample sizes and $n_1 = 15, n_2 = 5$ for imbalanced case. Thus the total sample size is 20 which is small. We will also use $n_1 = n_2 = n = 30$ for balanced sample sizes and $n_1 = 45, n_2 = 15$ for imbalanced case. Thus the total sample size is 60 which is relatively large.

(2) Monte Carlo replicates (N)

Since the experimentation is done on a computer, we can easily replicate the experiments. The larger the number of Monte Carlo replicates, N, the better the approximation will be. However, computing time and computer or software (e.g. SAS) memory may be limited, making it necessary to run the N Monte Carlo replicates in smaller batches.

The empirical standard deviation for the estimated true level $\hat{\alpha}$ for testing no treatment effects is

$$\mathrm{sd}(\hat{\alpha}) = \sqrt{\frac{(\hat{\alpha})(1-\hat{\alpha})}{N}}.$$

If $\hat{\alpha} = 0.05$, and we want $\mathrm{sd}(\hat{\alpha}) \leq 0.01$,

$$\mathrm{sd}(\hat{\alpha}) = \sqrt{\frac{(0.05)(0.95)}{N}} \leq 0.01,$$

then we must choose $N \geq 475$. If $\hat{\alpha} = 0.05$ and $N = 1000, \mathrm{sd}(\hat{\alpha}) = 0.0069$ and the corresponding 95% confidence interval for α is $0.05 \pm (1.96)(0.0069) = [0.0365, 0.0635]$.

In the experiment, we will use $N = 1000$.

(3) Seed

In the experiment, we will use arbitrarily chosen numbers for seeds. Different random samples are generated for normal and lognormal distributions so in this simulation experiment the simulated data from normal and lognormal distributions are uncorrelated.

(4) Summary

In this simulation study, the four two-level factors are summarized as

1) Distribution: lognormal distribution versus normal distribution;

2) Heterogeneity of variances: $\mathrm{var}(Y_1) : \mathrm{var}(Y_2) = 1 : 1$ versus $1 : 100$;

3) Sample size imbalance: $n_1 : n_2 = 1$ versus 3.

4) Total sample size $n_1 + n_2 = 20$ versus 60.

Specifically, for the constant variance case,

- for normal distribution, $Y_1 \sim \mathrm{Normal}(0, 1)$ and $Y_2 \sim \mathrm{Normal}(0, 1)$;

- for lognormal distribution, $Y_1 - 1.27201$ where $Y_1 \sim \mathrm{Lognormal}(0, 0.4812)$ and $Y_2 - 1.27201$ where $Y_2 \sim \mathrm{Lognormal}(0, 0.4812)$.

For nonconstant variance case,

- for normal distribution, $Y_1 \sim \mathrm{Normal}(0, 1)$ and $Y_2 \sim \mathrm{Normal}(0, 100)$;

- for lognormal distribution, $Y_1 - 1.27201$ where $Y_1 \sim \mathrm{Lognormal}(0, 0.4812)$ and $Y_2 - 12.7201 = 10(X - 1.27201)$ where $Y_2 = 10X$ and $X \sim \mathrm{Lognormal}(0, 0.4812)$.

This yields a $2 \times 2 \times 2 \times 2$ factorial experiment with a total of 16 combinations of factor levels.

6.3 The simulation procedure

The simulation procedure for this example is

Step 1). Set initial seeds for normal distribution and lognormal distribution.

Step 2). Generate a sample of n_i Y values. For the normal distribution, $Y = kZ$ where $Z \sim N(0,1)$; for the lognormal distribution, $Y = k(e^{\sqrt{0.4812}Z} - 1.27201)$ where $Z \sim N(0,1)$.

Step 3). Repeat step 2) for each of the 16 combinations of distribution nonnormality, variance heterogeneity, sample size imbalance and total sample size levels. If homogeneous then $k = 1$, if heterogeneous then $k = 10$. If the total sample size is small, $n_1 = n_2 = 10$ for balanced, $n_1 = 15$ but $n_2 = 5$ for imbalanced. If the total sample size is large, $n_1 = n_2 = 30$ for balanced, $n_1 = 45$ but $n_2 = 15$ for imbalanced.

Assign data to treatment 1 and treatment 2 so that for treatment 1 data have sample size n_1; but for treatment 2 data have sample size n_2. Notice that the simulated data are uncorrelated in this setup.

Step 4). Repeat steps 2) and 3) N times (e.g. $N = 1000$ Monte Carlo replicates).

Step 5). Use PROC GLM to perform F-tests for no treatment effects for every Monte Carlo replicate. In each Monte Carlo replicate, do a separate ANOVA for each combination of distribution level, variance heterogeneity level, sample size imbalance level and total sample size level.

In the SAS program, the simulated data are divided into several smaller datasets based on combinations of factor levels to perform the F-tests. This is so that $N = 1000$ can be used, otherwise only a small N can be chosen due to the size limitation in the outstat dataset in PROC GLM.

Step 6). Use $\alpha = .05$ as the significance level and calculate rejection rates for the 16 combinations of factor levels using PROC MEANS.

Step 7). We can construct normal probability plots to visualize which factors and interactions have the most effect on rejection rates. We can also use ANOVA to investigate the significance of main effects and some interactions by pooling higher order interactions and assuming those higher order interactions can be ignored.

6.4 Documenting the simulation experiment

(1) The design of the simulation study

Record the steps you take to develop an appropriate simulation and why you choose it and how you refine it. Keep track of the considerations for setting up the design including factors, levels of factors, design layout, simulation procedures, any assumptions made, and reasons for doing these, etc.

(2) Other information

Write notes on file names, corresponding input and output data sets, and dates.

Document computer codes and write descriptions and/or flow charts for the codes.

Number datasets. Record their corresponding codes and where they are used or obtained.

An example of documented SAS code is shown in Appendix (1).

6.5 Writing up

A Monte Carlo simulation study, however carefully done, is of little value unless its methods and results are understandable, believable and tailored to the questions of interest. It is essential that a simulation study report guides the reader through the objectives, design, methodology, assumptions, models, results, conclusions and discussions (if any).

(1) Methods

Summarize the analysis you conduct to address the questions of interest including assumptions, why you make the assumptions, models, etc.

(2) Results

The results of the simulation study must be represented in a form that clearly answers the questions of interest.

Graphs are often the best way to display data. Use graphs wherever appropriate. Graphs should be properly labeled. Graphs have the advantage of providing a quick, intuitive way to understand what is usually a fairly complex, number-intensive set of information.

Tables are a good way to demonstrate results too. Numbers give us the raw data and statistics from which we can make quantitative decisions. The organization of a table makes a big difference in the readability of the table and its impact. Put important information about the data in the title and in footnotes in graphs and tables. Readers should be able to look at a figure or a table and understand it without reading the text.

Discuss extraneous factors that are not negligible and include them in your interpretation of the results or suggestions for future research. Explain any simulation results you can with statistical theory.

(3) Conclusions and discussion of the results in the example

The rejection rates from the Monte Carlo simulation experiment are shown in Table 2, which indicate that in this experiment

- The most non-robust F-tests happen under nonconstant variance and imbalanced sample size cases regardless of nonnormality and total sample size.

- When data are normally distributed, the F-test is robust if sample sizes are balanced regardless of variance heterogeneity and total sample size. However, the F-test is not robust for either sample size if variances are not constant and sample sizes are not balanced.

Table 2: Monte Carlo simulation rejection rates for the F-test of the equality of two means when the variances and sample sizes are unequal and data come from either a normal or a lognormal distribution.

			Sample Size	
		Sample Size Balance	$n_1 + n_2 = 20$	$n_1 + n_2 = 60$
	Variance	$n_1 : n_2$ imbalanced:	15:5	45:15
Distribution	Homogeneity	$n_1 : n_2$ balanced:	10:10	30:30
Lognormal	Heterogeneous	Imbalanced	0.337	0.306
		Balanced	0.121	0.068
	Homogeneous	Imbalanced	0.041	0.044
		Balanced	0.037	0.043
Normal	Heterogeneous	Imbalanced	0.309	0.284
		Balanced	0.070	0.060
	Homogeneous	Imbalanced	0.070	0.050
		Balanced	0.051	0.048

For all cases, $\alpha = 0.05, E(Y) = 0, N = 1000, seed = 289739$ for normal, $seed = 983681$ for lognormal.

- When data are lognormally distributed, the F-test is robust if variances are constant regardless of sample sizes and sample size imbalance. The F-test is also robust for large samples if the sample sizes are balanced, regardless of variance heterogeneity. However, the F-test is not robust if the variances are not constant and sample sizes are not balanced for either small or large samples. The F-test is not robust if variances are not constant and sample sizes are small regardless of sample size imbalance.

- From the normal probability plot in Appendix (3), the main effects of variance heterogeneity, sample size imbalance, and their interaction stand out dramatically, which suggests that the effect of variance heterogeneity is different for the balanced sample size case and the imbalanced case.

- Based on the ANOVA from pooling several higher order interaction terms as residuals and assuming they can be ignored, similar conclusions as above can be made (see page 6 in the SAS output in Appendix (2)).

- Rejection rates tend to be slightly smaller as the total sample size gets larger for those situations with non-robust F-test.

- In this simulation study, $n_1 \geq n_2$. For future simulation experiment, $n_1 \leq n_2$ may also be studied and thus a new factor can be included in the experimental design.

- In this simulation study, $\mathrm{var}(Y_1) : \mathrm{var}(Y_2) = 1 : 100$ is fixed. For future simulation experiment, smaller $\mathrm{var}(Y_1) : \mathrm{var}(Y_2)$ ratio may also be studied and thus another new factor can be included in the experimental design.

References

[1] Brandt, Siegmund. Data analysis: statistical and computational methods for scientists and engineers, New York: Springer, 1999.

[2] Binder, K., Heermann, D. W. Monte Carlo simulation in statistical physics: an introduction, New York: Springer, 1997.

[3] Cox, D. R. Planning of Experiments, New York: Wiley, 1958.

[4] Kleijnen, Jack P. C. Statistical techniques in simulation, New York: M. Dekker, 1974-75.

[5] Monahan, John F. Numerical Methods of Statistics, Cambridge: Cambridge Univ Press, 2001.

[6] Rubinstein, Reuven Y. Simulation and the Monte Carlo method, New York: Wiley, 1981.

[7] Yakowitz, Sidney J. Computational probability and simulation, Mass.: Addison-Wesley Pub. Co., Advanced Book Program, 1977.

[8] Zhang, J. and Boos, D.D. Adjusted power estimates in Monte Carlo Experiments, Communications in Statistics – Simulation, Vol 23, 165-173.

7 Appendix

(1) SAS code and description for the example

```
/**********************************************************************
***                                                                 ***
***        SIMULATION STUDY IN STATISTICS - EXAMPLE                 ***
***                                                                 ***
***  Monte Carlo simulation to investigate the effects of           ***
***  nonnormality, nonconstant variance, and unequal sample sizes   ***
***  on robustness of the test for treatment effects in the         ***
***  analysis of variance.                                          ***
***                                                                 ***
***  2x2x2x2 factorial experiment                                   ***
***    distributions: normal, lognormal;                            ***
***    variance heterogeneity: 1:1 (homo), 1:100 (hetero);         ***
***    sample size imbalance: 1:1 (balanced), 3:1 (imbalanced);    ***
***    total sample size: 20 (small), 60 (big).                    ***
***                                                                ***
*** For normal,    y ~ N(0, 1) or y ~ N(0, 100)                    ***
*** For lognormal, y+1.27201 ~ Lognormal(0, 0.4812) or             ***
***                y+12.7201 ~ Lognormal(2.3026, 0.4812)           ***
***                                                                ***
*********************************************************************/

options pageno=1 ls=80; title;
goptions ftext=swiss htext=0.8;
title 'SIMULATION STUDY OF ROBUSTNESS OF F-TEST';
run;

*--------------------------------------------------------------------*;
******    GENERATE RANDOM NUMBERS USING MONTE CARLO METHODS:    ******;

data simu;

/**********************************************************************
*  The number of Monte Carlo replicates = 1000 (N=1000 in the notes). *
**********************************************************************/

  do mcrep = 1 to 1000;

/**********************************************************************
* Factor d = distribution :                                           *
* d = 1, normal                                                       *
*   = 2, lognormal                                                    *
**********************************************************************/

  do d = 1 to 2;
    if d=1 then distn='normal';
    if d=2 then distn='lognor';

/**********************************************************************
* Factor h = variance homogeneity :                                   *
* h = 1, homogeneous                                                  *
*   = 2, heterogeneous                                                *
**********************************************************************/

  do h=1 to 2;
    if h=1 then hratio = 1;
    if h=1 then homog='homogene';
    if h=2 then homog='heteroge';

/**********************************************************************
* Factor b = sample size balance :                                    *
* b = 1, balanced                                                     *
*   = 2, imbalanced                                                   *
**********************************************************************/

  do b=1 to 2;
    if b=1 then balanced='balanced';
    if b=2 then balanced='imbalanc';

/**********************************************************************
* Factor s = total sample size :                                      *
* s = 1, small                                                        *
*   = 2, big                                                          *
* define n1 and n2 for imbalanced                                     *
* define n = (n1+n2)/2 for balanced                                   *
**********************************************************************/

  do s = 1 to 2;
    if s=1 then do;  size='small';  n1 = 15; n2 =  5;  end;
    if s=2 then do;  size='big';    n1 = 45; n2 = 15;  end;
    if b=1 then n = (n1+n2)/2;

/**********************************************************************
* Assign treatments 1 and 2 :                                         *
**********************************************************************/

  do trt=1 to 2;
    if h=2 and trt=1 then hratio =  1;
    if h=2 and trt=2 then hratio = 10;
    if b=2 and trt=1 then n = n1;
    if b=2 and trt=2 then n = n2;

/**********************************************************************
* Generate random numbers :                                           *
* z ~ N ( 0, sigma^2 )                                                *
* y = hratio*(z-a),      if normal                                    *
*   = hratio*(exp(z)-a), if lognormal                                 *
* where,  sigma^2 = 1,        a = 0,        if normal                 *
*                 = 0.4812,     = 1.27201,  if lognormal              *
*        hratio =  1, if homogeneous                                  *
*                = 10, if heterogeneous                               *
**********************************************************************/
```

```
  do i=1 to n;
    if d=1 then y = hratio * ( rannor(289739) );
    if d=2 then y = hratio * ( exp( sqrt(0.4812)*rannor(983681) ) - 1.27201);
  output;
  end;

  end; end; end; end; end; end;
run;

/**********************************************************************
* Print simulated data for the 1st Monte Carlo replicate for          *
* illustration.                                                       *
**********************************************************************/

proc sort data=simu;
  by mcrep distn homog balanced size;
run;
/*
title2 'Simulated data (1st Monte Carlo replicate)';
proc print data=simu;
  where (mcrep=1);
  var mcrep distn homog balanced size trt y;
run;
*/

*--------------------------------------------------------------------*;
***   COMPUTE REJECTION RATES OF F-TESTS FOR TREATMENT EFFECTS:    ***;

/**********************************************************************
* Use PROC GLM to perform F-tests for treatment effects for every     *
* Monte Carlo replicate, every distribution level, every variance     *
* homogeneity level, every sample size balance level and              *
* every total size level.                                             *
* Divide whole data into 8 parts to perform F-test so that we can use *
* N=1000, otherwise only small N (e.g. a few hundreds) can be used    *
* due to the size limitation in outstat dataset.                      *
**********************************************************************/

data simu1; set simu;
  where (homog='homogene' and balanced='balanced' and size='small');
run;
proc glm data=simu1 noprint outstat=glmout1;
  by mcrep distn homog balanced size;
  class trt;
  model y=trt;
run;

data simu2; set simu;
  where (homog='homogene' and balanced='balanced' and size='big');
run;
proc glm data=simu2 noprint outstat=glmout2;
  by mcrep distn homog balanced size;
  class trt;
  model y=trt;
run;

data simu3; set simu;
  where (homog='homogene' and balanced='imbalanc' and size='small');
run;
proc glm data=simu3 noprint outstat=glmout3;
  by mcrep distn homog balanced size;
  class trt;
  model y=trt;
run;

data simu4; set simu;
  where (homog='homogene' and balanced='imbalanc' and size='big');
run;
proc glm data=simu4 noprint outstat=glmout4;
  by mcrep distn homog balanced size;
  class trt;
  model y=trt;
run;

data simu5; set simu;
  where (homog='heteroge' and balanced='balanced' and size='small');
run;
proc glm data=simu5 noprint outstat=glmout5;
  by mcrep distn homog balanced size;
  class trt;
  model y=trt;
run;

data simu6; set simu;
  where (homog='heteroge' and balanced='balanced' and size='big');
run;
proc glm data=simu6 noprint outstat=glmout6;
  by mcrep distn homog balanced size;
  class trt;
  model y=trt;
run;

data simu7; set simu;
  where (homog='heteroge' and balanced='imbalanc' and size='small');
run;
proc glm data=simu7 noprint outstat=glmout7;
  by mcrep distn homog balanced size;
  class trt;
  model y=trt;
run;

data simu8; set simu;
  where (homog='heteroge' and balanced='imbalanc' and size='big');
run;
proc glm data=simu8 noprint outstat=glmout8;
  by mcrep distn homog balanced size;
  class trt;
  model y=trt;
run;

data glmout;
  set glmout1 glmout2 glmout3 glmout4 glmout5 glmout6 glmout7 glmout8;
run;

/**********************************************************************
* Print the F-test results for 1st Monte carlo replicate for          *
* illustration.                                                       *
**********************************************************************/
/*
title2 'GLMOUT (1st Monte Carlo replicate)';
proc print data=glmout;
  where (mcrep=1);
run;
*/

/**********************************************************************
* Calculate rejection rates using PROC MEANS for the 16 combinations  *
* significance level=.05 for every total sample size                  *
**********************************************************************/

data glmout;  set glmout;
  if (_source_='trt' and _type_='SS3' or _source_='TRT' and _type_='SS3');
  drop _name_ _source_ _type_ df ss f;
  reject=(prob<.05);
```

```
run;

title2 'Rejection rates for treatment effect (1000 Monte Carlo replicates)';
proc means data=glmout nway noprint;
  class distn homog balanced size;
  var reject;
  output out=results mean=rejrate ;
proc print data=results;
run;


*-------------------------------------------------------------------*;
******           CONSTRUCT NORMAL PROBABILITY PLOT:           ******;

/*********************************************************************
* Define all factors in the experiment                              *
*                                                                   *
*   d =  1, normal                                                  *
*     = -1, lognormal                                               *
*                                                                   *
*   h =  1, constant variance                                       *
*     = -1, nonconstant variance                                    *
*                                                                   *
*   b =  1, sample sizes balance                                    *
*     = -1, sample sizes imbalance                                  *
*                                                                   *
*   s =  1, size is small                                           *
*     = -1, size is big                                             *
*                                                                   *
* and all interactions                                              *
*   dxh, dxb, dxs, hxb, hxs, bxs, dxhxb, dxhxs, dxbxs, hxbxs, dxhxbxs *
*********************************************************************/

data empiric;  set results;
  d = (distn='normal')-(distn='lognor');
  h = (homog='homogene')-(homog='heteroge');
  b = (balanced='balanced')-(balanced='imbalanc');
  s = (size='small')-(size='big');
  dxh=d*h;  dxb=d*b;  dxs=d*s;  hxb=h*b;  hxs=h*s;  bxs=b*s;
  dxhxb=d*h*b;  dxhxs=d*h*s;  dxbxs=d*b*s;  hxbxs=h*b*s;
  dxhxbxs=d*h*b*s;
run;

/*********************************************************************
* Use PROC REG to compute main effects and interactions and output to *
* a file.                                                            *
*********************************************************************/

title2 'Contrasts among factors and their interactions';
proc reg data=empiric outest=regout;
  model rejrate=d h b s dxh dxb dxs hxb hxs bxs dxhxb dxhxs dxbxs
                hxbxs dxhxbxs;
run;

proc transpose data=regout out=ploteff name=effect prefix=est;
  var d h b s dxh dxb dxs hxb hxs bxs dxhxb dxhxs dxbxs hxbxs dxhxbxs;
run;

/*********************************************************************
* Compute normal scores.                                             *
*********************************************************************/

proc rank data=ploteff normal=blom out=qqplot;
  var est1;
  ranks normalq;
run;

/*********************************************************************
* Plot normal scores vs the effect estimates to investigate the      *
* significance of the main effects and interactions                  *
*********************************************************************/

title;
proc gplot data=qqplot;
  plot normalq*est1=effect;
  label normalq='Standard normal quantiles' est1='betahat';
  symbol1  c=black h=0.9 f=simplex v='B';
  symbol2  c=black h=0.9 f=simplex v='BS';
  symbol3  c=black h=0.9 f=simplex v='D';
  symbol4  c=black h=0.9 f=simplex v='DB';
  symbol5  c=black h=0.9 f=simplex v='DBS';
  symbol6  c=black h=0.9 f=simplex v='DH';
  symbol7  c=black h=0.9 f=simplex v='DHB';
  symbol8  c=black h=0.9 f=simplex v='DHBS';
  symbol9  c=black h=0.9 f=simplex v='DHS';
  symbol10 c=black h=0.9 f=simplex v='DS';
  symbol11 c=black h=0.9 f=simplex v=H;
  symbol12 c=black h=0.9 f=simplex v='HB';
  symbol13 c=black h=0.9 f=simplex v='HBS';
  symbol14 c=black h=0.9 f=simplex v='HS';
  symbol15 c=black h=0.9 f=simplex v='S';
run;


*------------------------------------------------------------------*;
************:   TEST MAIN EFFECTS AND SOME INTERACTIONS:  ************;

/*********************************************************************
* ANOVA for main effects and some lower order interactions           *
* pooling all higher-order interaction terms as error                *
*********************************************************************/

title2 'Test main effects and some interactions';
proc glm data=empiric;
  class d h b s;
  model rejrate = d h b s d*h d*b d*s h*b h*s b*s d*h*b;
run;
```

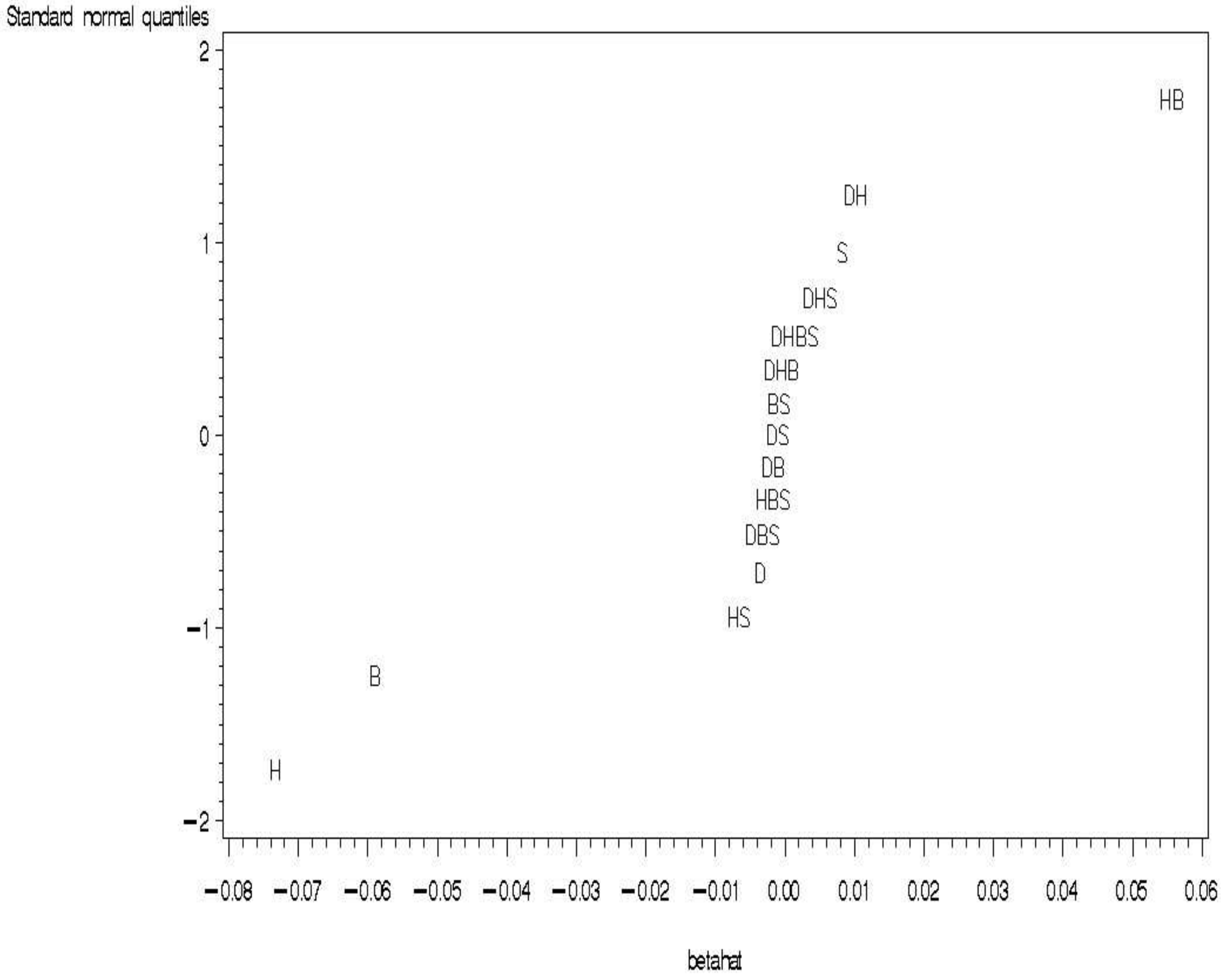
(2) SAS output for the example

```
                  SIMULATION STUDY OF ROBUSTNESS OF F-TEST                 1
      Rejection rates for treatment effect (1000 Monte Carlo replicates)
                                                  01:22 Tuesday, June 19, 2001

 Obs    distn     homog      balanced    size     _TYPE_    _FREQ_    rejrate

   1    lognor    heteroge    balanced    big        15        1000      0.068
   2    lognor    heteroge    balanced    small      15        1000      0.121
   3    lognor    heteroge    imbalanc    big        15        1000      0.306
   4    lognor    heteroge    imbalanc    small      15        1000      0.337
   5    lognor    homogene    balanced    big        15        1000      0.043
   6    lognor    homogene    balanced    small      15        1000      0.037
   7    lognor    homogene    imbalanc    big        15        1000      0.044
   8    lognor    homogene    imbalanc    small      15        1000      0.041
   9    normal    heteroge    balanced    big        15        1000      0.060
  10    normal    heteroge    balanced    small      15        1000      0.070
  11    normal    heteroge    imbalanc    big        15        1000      0.284
  12    normal    heteroge    imbalanc    small      15        1000      0.309
  13    normal    homogene    balanced    big        15        1000      0.048
  14    normal    homogene    balanced    small      15        1000      0.051
  15    normal    homogene    imbalanc    big        15        1000      0.050
  16    normal    homogene    imbalanc    small      15        1000      0.070


                  SIMULATION STUDY OF ROBUSTNESS OF F-TEST                 2
                 Contrasts among factors and their interactions
                                                  01:22 Tuesday, June 19, 2001
```

```
                              The REG Procedure
                                Model: MODEL1
                         Dependent Variable: rejrate

                             Analysis of Variance

                                    Sum of           Mean
Source                   DF        Squares         Square    F Value    Pr > F

Model                    15        0.19526        0.01302        .       .
Error                     0              0              .
Corrected Total          15        0.19526


             Root MSE                    .    R-Square     1.0000
             Dependent Mean        0.12119    Adj R-Sq          .
             Coeff Var                   .


                   SIMULATION STUDY OF ROBUSTNESS OF F-TEST                   3
                Contrasts among factors and their interactions
                                                   01:22 Tuesday, June 19, 2001

                              The REG Procedure
                                Model: MODEL1
                         Dependent Variable: rejrate

                             Parameter Estimates

                          Parameter       Standard
     Variable     DF       Estimate          Error    t Value    Pr > |t|

     Intercept     1        0.12119              .        .         .
     d             1       -0.00344              .        .         .
     h             1       -0.07319              .        .         .
     b             1       -0.05894              .        .         .
     s             1        0.00831              .        .         .
     dxh           1        0.01019              .        .         .
     dxb           1       -0.00156              .        .         .
     dxs           1       -0.00106              .        .         .
     hxb           1        0.05569              .        .         .
     hxs           1       -0.00656              .        .         .
     bxs           1    -0.00081250              .        .         .
     dxhxb         1    -0.00043750              .        .         .
     dxhxs         1        0.00506              .        .         .
     dxbxs         1       -0.00319              .        .         .
     hxbxs         1       -0.00169              .        .         .
     dxhxbxs       1        0.00144              .        .         .
```

```
                                                01:22 Tuesday, June 19, 2001   4
                    Test main effects and some interactions

                               The GLM Procedure

                            Class Level Information

                         Class         Levels    Values

                         d                  2    -1 1

                         h                  2    -1 1

                         b                  2    -1 1

                         s                  2    -1 1


                          Number of observations    16


                                                01:22 Tuesday, June 19, 2001   5
                    Test main effects and some interactions

                               The GLM Procedure

Dependent Variable: rejrate

                                       Sum of
 Source                      DF       Squares    Mean Square   F Value   Pr > F

 Model                       11    0.19461319     0.01769211    108.67   0.0002

 Error                        4    0.00065125     0.00016281

 Corrected Total             15    0.19526444


              R-Square     Coeff Var      Root MSE    rejrate Mean

              0.996665      10.52897      0.012760        0.121188


 Source                      DF     Type I SS    Mean Square   F Value   Pr > F

 d                            1    0.00018906     0.00018906      1.16   0.3419
 h                            1    0.08570256     0.08570256    526.39   <.0001
 b                            1    0.05557806     0.05557806    341.36   <.0001
 s                            1    0.00110556     0.00110556      6.79   0.0597
 d*h                          1    0.00166056     0.00166056     10.20   0.0331
 d*b                          1    0.00003906     0.00003906      0.24   0.6499
 d*s                         1     0.00001806     0.00001806      0.11   0.7558


                                                01:22 Tuesday, June 19, 2001   6
                    Test main effects and some interactions

                               The GLM Procedure

Dependent Variable: rejrate

 Source                     DF      Type I SS    Mean Square   F Value   Pr > F

 h*b                         1     0.04961756     0.04961756    304.75   <.0001
 h*s                         1     0.00068906     0.00068906      4.23   0.1088
 b*s                         1     0.00001056     0.00001056      0.06   0.8115
 d*h*b                       1     0.00000306     0.00000306      0.02   0.8975


 Source                     DF    Type III SS    Mean Square   F Value   Pr > F

 d                           1     0.00018906     0.00018906      1.16   0.3419
 h                           1     0.08570256     0.08570256    526.39   <.0001
 b                           1     0.05557806     0.05557806    341.36   <.0001
 s                           1     0.00110556     0.00110556      6.79   0.0597
 d*h                         1     0.00166056     0.00166056     10.20   0.0331
 d*b                         1     0.00003906     0.00003906      0.24   0.6499
 d*s                         1     0.00001806     0.00001806      0.11   0.7558
 h*b                         1     0.04961756     0.04961756    304.75   <.0001
 h*s                         1     0.00068906     0.00068906      4.23   0.1088
 b*s                         1     0.00001056     0.00001056      0.06   0.8115
 d*h*b                       1     0.00000306     0.00000306      0.02   0.8975
```

(3) Normal probability plot for the example



D = distribution, H = heterogeneity, B = imbalance, S = total sample size.