CLASSROOM SIMULATION: ARE VARIANCE-STABILIZING TRANSFORMATIONS REALLY USEFUL?

Bruce E. TRUMBO, Eric A. SUESS, and Rebecca E. BRAFMAN; *California State University, Hayward.*
Department of Statistics; California State University, Hayward; Hayward, CA 94542 USA
btrumbo@csuhayward.edu or esuess@csuhayward.edu

Abstract: When population variances of observations in an ANOVA are a known function of their population means, many textbooks recommend using variance-stabilizing transformations. Examples are: square root transformation for Poisson data, arcsine of square root for binomial proportions, and log for exponential data. We investigate the usefulness of transformations in one-factor, 3-level ANOVAs with nonnormal data. Simulations approximate the true significance level and power of F-tests—with and without various variance-stabilizing transformations. Findings: logarithmic and rank transformations of exponential data can be useful when the number of replications is small and the separation in means is large. Simulation code for Minitab and S-Plus/R is provided. Classroom use of such simulations in a second statistics course reinforces concepts of significance level and power, encourages exploration, and teaches computer skills important in the job market.

Key words: Variance-stabilizing transformation; Nonnormal data; square-root/arcsine/log transformations, simulation, rejection probability; Poisson/binomial/exponential data; Minitab/S-Plus/R/SAS software, teaching.

1. Introduction.

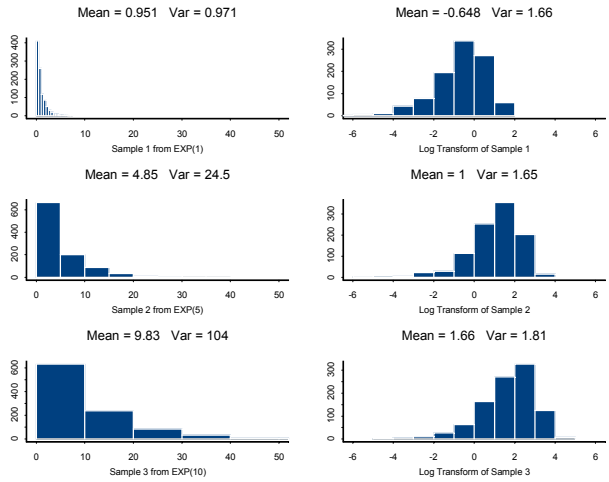
Background. Consider a one-factor ANOVA model with r observations on each of t groups: $X_{ij} = \mu_i + e_{ij}$, where e_{ij} are i.i.d. $N(0, \sigma^2)$; $i = 1, ..., t$; $j = 1, ..., r$. Important assumptions are (i) normality of the data, and (ii) the same variance σ^2 within all t groups.

The goal is to test the null hypothesis that all μ_i are equal. To do this, compute the mean squares MS(Factor) = $r\Sigma_i (M_i - G)^2 / (t-1)$ and MS(Error) = $\Sigma_i V_i / t$, and F = MS(Factor)/MS(Error), where $M_i = \Sigma_j X_{ij}/r$, $G = \Sigma_{ij} X_{ij}/rt$, and $V_i = S_i^2 = \Sigma_j (X_{ij} - M_i)^2 / (r-1)$. If the null hypothesis and the assumptions of the model are true, then F has the distribution $F(\nu_1, \nu_2)$ where $\nu_1 = t - 1$, and $\nu_2 = t(r-1)$. The null hypothesis is rejected at the 5% level if $F > F^*$, where F^* is the critical value, which cuts 5% of the area from the upper tail of this F-distribution.

Nonnormal data. In practice one sometimes seeks to test whether t *nonnormal* group populations have equal means μ_i. In this case the variance σ^2 of the distribution family may be a function of the mean: $\sigma^2 = \varphi(\mu)$. Common examples: Poisson with $\sigma^2 = \mu$, exponential with $\sigma^2 = \mu^2$, and binomial proportions (based on n trials and success probability p) with $\mu = p$ and $\sigma^2 = p(1-p)/n$. Then a finding that group means are unequal implies that group variances are unequal and thus that the model is invalid not only because of nonnormality but also, and perhaps more seriously, because group variances are unequal.

Variance-stabilizing transformations. In such a situation, many textbooks [e.g., 1, 2] recommend use of variance-stabilizing transformations. By methods of calculus one can show [3, 4, 5] that the function f with $df(\mu)/d\mu = [\varphi(\mu)]^{-1/2}$ stabilizes (approximately equalizes) variances. Data $Y_{ij} = f(X_{ij})$ are used in computing F and testing the null hypothesis.

Figure 1: Log Transforms (Right) of Three Exponential Samples Have Similar Variances
Population means are 1 (Top), 5, and 10; n = 1000.



Specifically, this method indicates the transformation $Y = X^{1/2}$ for Poisson data, $Y = \arcsin X^{1/2}$ for binomial proportions, and $Y = \ln X$ for exponential data. In particular situations, some adjustments are recommended: For Poisson data that have a large proportion of 0s, one may use $Y = (X + \frac{1}{2})^{1/2}$ or $Y = (X+1)^{1/2}$. Before transformation, binomial proportions 0 or 1 are often replaced by $1/4n$ or $1 - 1/4n$, respectively. If rounding of (inherently positive) exponential observations results in recording some 0s, then one must add a small constant to all observations before taking logs.

2. Data Transformations in Practice.

Discovering heteroscedasticity. Sometimes unequal group variances are revealed by diagnostic devices such as a plot of the residuals $X_{ij} - M_i$ against the corresponding fits M_i or formal tests of equality of variance (e.g., Hartley F_{max}, Bartlett, or Levene). Then one might look at ratios such as M_i/S_i, M_i/V_i to see if they are

approximately equal across all t groups and, if so, to discern what function φ may relate μ and σ^2.

Sometimes one can confidently guess the family of distributions from which the Xs may come. Poisson: counts of accidents, flaws, or contaminating particles. Binomial proportions: proportion of seeds among 20 placed in each pot that germinate, proportion of culture plates out of 20 that grow visible bacteria colonies. Exponential: certain kinds of reaction times, waiting times, and financial data.

Interpreting transformed data. When one knows or can reasonably surmise the type of data, the question arises whether it is useful in practice to transform the data. The goal is not to achieve a pretty residual plot, but to make the correct decision. Perhaps doing a transformation does not change the outcome that matters—whether or not the null hypothesis is rejected.

Also, if the null hypothesis is rejected for data transformed as square roots or arcsines of square roots, then interpretations of multiple comparison procedures on the scale of the original data can be confusing. For example, the difference between square roots is not the same as the square root of the difference. For logged data, multiple comparisons are easier to interpret: Differences in logs are logs of ratios.

In fortunate situations, the decision to accept or reject is the same whether or not data are transformed. Then one can interpret data on the original scale and relegate to a footnote the information that a transformation makes no practical difference. In our experience, this is often the case. We do not recall having seen examples based on real data in textbooks or in our statistical practice where a square root transformation on Poisson data or an arcsine-of-square-root transformation on binomial proportions made the difference between accepting or rejecting the null hypothesis at a reasonable level of significance.

One may wonder what would be the results of a poll of a large number of practicing statisticians, asking about the usefulness of transformations in real-life situations. But even with the results of such a poll in hand, one still might wonder whether the proper distribution families had always been identified and the "appropriate" transformations made.

Simulation. In the rest of this paper we conduct a number of "polls" by simulation. For example, we consider a population of one-factor ANOVAs with $t = 3$ and $r = 5$ and where we know the data are Poisson and that there are no differences among population means. We take a large sample of $m = 20\,000$ such datasets at random, and analyze them with and without the square root transformation. Overall, are we more likely to "discover" differences that do not really exist (falsely reject the null hypothesis) when we transform the data—or when we do not? Then we take another large sample of Poisson datasets where we know there is a certain pattern of differences among the groups. Do we detect these differences (correctly reject the null hypothesis) more often when we use the original data—or when we use the transformed data?

A simulation study is better in some ways than an actual poll of practicing statisticians because we can take huge samples, we know the data are exactly Poisson, and we know whether there are differences among groups. But a simulation study is not true to life in all respects because real statisticians can only consider the source of the data and look at the actual numbers to try to guess whether they are approximately Poisson distributed.

Even so, by doing simulations that are easy to understand, easy to program with modern software, and reasonably quick to run on today's computers, one can learn a lot about the usefulness of transformations. Perhaps more important, by following through the simulation procedure a statistics student will get some important insights into how computers are currently used in statistical practice and research.

Display 2: Minitab Code for Poisson Simulations

```
MTB > name c16 'm1' c17 'm2' c18 'm3' c20 'MSF'
MTB > name c21 'v1' c22 'v2' c23 'v3' c24 'MSE'
MTB > name c25 'F.stat' c26 'Rej1.Acc0'
MTB > rand 20000 c1-c5;
SUBC> pois 10.
MTB > rand 20000 c6-c10;
SUBC> pois 10.
MTB > rand 20000 c11-c15;
SUBC> pois 10.
MTB > stack c1-c15 c30;
SUBC> subs c31.
MTB > let c30 = sqrt(c30)
MTB > unstack c30 c1-c15;
SUBC> subs c31.
MTB > erase c30 c31
MTB > rmean c1-c5 c16
MTB > rmean c6-c10 c17
MTB > rmean c11-c15 c18
MTB > rstdev c16-c18 c19
MTB > let c20 = 5*c19*c19
MTB > rstdev c1-c5 c21
MTB > rstdev c6-c10 c22
MTB > rstdev c11-c15 c23
MTB > let c24 = (c21*c21 + c22*c22 + c23*c23)/3
MTB > let c25 = c20/c24
MTB > code (0:3.8852)0 (3.8853:10000)1 c25 c26
MTB > mean c26
```

3. Simulations with Poisson Data.

Simulation plan. We run four similar Minitab "programs" with simulated Poisson data:

(1) Means equal, original data,
(2) Means equal, transformed data,
(3) Means unequal, original data,
(4) Means unequal, transformed data.

In each case we estimate P(Reject) based on the critical value $F(.95, 2, 12) = F^* = 3.8853$,.

Refer to Display 2. We start by putting simulated data into columns c1-c15 of the Worksheet. Each of the 20 000 *rows* has data for one simulated ANOVA with Group 1 in c1-c5, Group 2 in c6-c10, and Group 3 in c11-c15. Notice that in this run all group means are equal: $\mu_1 = \mu_2 = \mu_3 = 10$. (For now, ignore the block of code in **bold** type.) Then we use Minitab's row arithmetic commands to manipulate the data to obtain MS(Factor), MS(Error), and the *F*-statistic. The code command puts a 1 in c26 if $F \geq F^* = 3.8853$ (null hypothesis rejected). The mean of this column, its proportion of 1s, approximates P(Reject | equal means).

This program, and others in this paper, are available online in digital form [6]. After you run this program (leaving out the six lines in **bold**), look at the data in the *first row* of c1-c15, and verify the value of *F* in the first cell of c25. The program does this same work 20 000 times, once in each row. In Minitab 14, if you use the command `base 1237` before you run the program, you get the data, and from them the ANOVA table, shown in Display 3. There $F > 0.866$ so $P > 0.05$, and we Accept the null hypothesis at the 5% level.

Display 3: First of 20 000 ANOVAs, Poisson Means 10

```
Group 1:  10,  8,  8, 11, 14
Group 2:   9,  5, 13, 11, 10
Group 3:   7, 11, 11,  9,  9

Source  DF     SS    MS     F      P
Factor   2   1.73  0.87  0.15  0.866
Error   12  71.20  5.93
Total   14  72.93
```

Based on all 20 000 rows in this run, we obtain P(Reject) ≈ 0.0487. Several additional runs give very similar results. The margin of error of this result is about $1.96\,[.0487(1-.0487)/20\,000]^{1/2} = 0.003$. Even though the data are not normal, the standard ANOVA procedure rejects the null hypothesis about 5% of the time when the Poisson populations have a common mean of 10. The 5% value also holds in other cases: for example, when $\mu_1 = \mu_2 = \mu_3 = 3$.

The lines of code in bold type in Display 2 stack all 15(20 000) = 300 000 simulated observations into c30 where square roots are taken. Then the information in c31 permits the transformed data to be "unstacked" to replace the original data in c1-c15. A run of the entire program in Display 2 estimates P(Reject) ≈ 5% for the transformed data. (Again for transformed data, we get a similar value when the group means are all equal to 3.) We conclude that the transformation does not make an important change in P(Reject | equal means).

By changing the three identical means **10** at the beginning of the program to 10, 15, and 20, respectively, we can approximate the power of the test against the alternative $\mu_1 = 10$, $\mu_2 = 15$, $\mu_3 = 20$. With or without transformation, we get P(Reject | 10, 15, 20) ≈ 0.91.

With a variety of other sets of unequal μ_i, we get different values for the power from one set to the next. But we see very little difference, if any, between results with and without transformation. When means are small enough to yield large numbers of 0 values, we use transformations $Y = (X + 0.5)^{1/2}$ and $Y = (X + 1)^{1/2}$, without seeing any change in results. (A tabulation of rejection probabilities, with and without transformation for various patterns of means is available online [6].)

In summary, within the limited scope of our simulations on small balanced one-factor ANOVAs, *we find no instance in which a square-root transformation of Poisson data substantially alters the probability of rejection.*

4. Simulations with Binomial Data.

Suppose that the data are binomial proportions based on $n = 10$ trials. Slight modifications of the code of Sect. 3 can be used to investigate the usefulness of the transformation $Y = \arcsin X^{1/2}$. Recall that $E(X) = \mu = p$, for a binomial proportion X with success probability is p. Because $V(X) = p(1-p)/n$ does not vary much for p between 0.3 and 0.7, we choose values of p that might yield meaningfully different variances.

Display 4 shows the changes to simulate P(Reject) for original (untransformed) proportions where $\mu_1 = 0.1$, $\mu_2 = 0.25$, and $\mu_3 = 0.4$. (We stack the data into c30 to change counts to proportions.) Subsequently, for the "arcsine" transformation, the line

```
MTB > let c30 = asin(sqrt(c30/10))
```

replaces the bold line in Display 4.

Display 4: Command Changes for Binomial Proportions

```
...
MTB > rand 20000 c1-c5;
SUBC> bino 10 .1.
MTB > rand 20000 c6-c10;
SUBC> bino 10 .25.
MTB > rand 20000 c11-c15;
SUBC> bino 10 .4.
MTB > stack c1-c15 c30;
SUBC> subs c31.
MTB > let c30 = c30/10
MTB > unstack c30 c1-c15;
SUBC> subs c31.
MTB > erase c30 c31
...
```

We find that P(Reject | .1, .25, .4) ≈ .82 and P(Reject | .2, .2, .2) ≈ .05, with or without the "arcsine" transformation. We also find that rejection probabilities are not changed substantially by the logistic transformation $Y = \ln[X/(1-X)]$, where values of X near 0 or 1 are fudged slightly to lie inside (0, 1). Add the line

```
MTB > code (0).025 (1).975 c30 c30
```

just after "stacking" the data to do the fudging.

Again here, within the limited scope of our simulations on small balanced one-factor ANOVAs (detailed in [6]), we do not find evidence that these transformations are useful in practice. Specifically, *we find no*

instance in which either an "arcsine" or a logistic transformation substantially alters rejection probabilities when the data are binomial proportions. As we see in Sect. 5, the situation is considerably different for log and rank transformations of exponential data.

5. Simulations with Exponential Data.

In order to investigate log transformations on exponential data, the only changes necessary in the code of Display 2 are to change `pois` to `expo` where the random observations are generated and `sqrt` to `loge` in the third bold line. So we take this opportunity to introduce code to do simulations in R.

A program in R. Display 5 shows the required R code. Here is a brief explanation of what it does. Additional information on each function can be obtained by typing a `?` followed by a space and the function name into the R Console window (or looking ahead to the next subsection, in the Commands window of S-Plus).

The program is written so that the number $r = 5$ of observations per group, the group population means, and (within limits of allocated memory space) the number m of ANOVAs simulated are easy to change, but it would have to be rewritten to change the number $t = 3$ of groups. Object names imitate the notation of Sect. 1.

Display 5: R Code for Exponential Simulations

```
r <- 5;  m <- 20000
mu1 <- 10;  mu2 <- 10;  mu3 <- 10
mu <- c(rep(mu1,r), rep(mu2,r), rep(mu3,r))
x <- rexp(3*r*m, rate=1/mu)
DTA <- matrix(x, m, byrow=T)
#DTA <- log(DTA)
     # Activate line above for log transf.
#DTA <- t(apply(DTA, 1, rank))
     # Activate line above for rank transf.
m1 <- rowMeans(DTA[,1:r])
m2 <- rowMeans(DTA[,(r+1):(2*r)])
m3 <- rowMeans(DTA[,(2*r+1):(3*r)])
v1 <- rowSums((DTA[,1:r] - m1)^2)/(r-1)
v2 <- rowSums((DTA[,(r+1):(2*r)] - m2)^2)/(r-1)
v3 <- rowSums((DTA[,(2*r+1):(3*r)] - m3)^2)/(r-1)
g <- (m1 + m2 + m3)/3
MSF <- r * rowSums((cbind(m1,m2,m3) - g)^2)/2
MSE <- rowMeans(cbind(v1, v2, v3))
F.rat <- MSF/MSE
rej <- (F.rat > qf(.95, 2, 3*(r-1)))
mean(rej)
```

The vector `mu` has $3r = 15$ elements. It is "recycled" m times as the function `rexp` simulates the vector `x` of $3rm$ exponential observations. The contents of `x` are reformatted (reading across rows) into an $m \times 3r$ matrix `DTA`, so that each row of this matrix contains data for one of the m simulated ANOVAs. If printed out, the matrix `DTA` would look very similar to columns c1-c15 of the Minitab worksheet considered above.

Brackets `[ ]` denote the elements making up a subvector or submatrix. Here, three ranges of column indices are used to break the large matrix `DTA` into three $m \times r$ submatrices, each corresponding to one of the groups. Column vectors such as `m1` and `v1` have m elements each of group sample means and variances, respectively. The function `cbind` combines column vectors into a matrix. The m-element column vector `rej` is a logical vector with entries `T` or `F`, where `T` indicates that the F-test rejects the null hypothesis (of equal group population means) for the ANOVA in a particular row, and `F` indicates acceptance. The mean of this vector gives the proportion of `T`s it contains, and so `mean(rej)` simulates P(Reject).

Table 6: Rejection Probabilities for Nominal 5% Level Tests
Log and Rank Transformed Exponential Data in One-Factor ANOVAs
(Where 3 decimal places are shown, the last digit may be ±2.)

	Transformation		
Group Means	None	Log	Rank
***r* = 5**			
10, 10, 10	0.040	0.046	0.056
1, 2, 4	0.24	0.28	0.31
1, 5, 10	0.39	0.65	0.68
1, 10 ,10	0.34	0.76	0.78
1, 10, 100	0.76	0.986	0.985
***r* = 10**			
10, 10, 10	0.043	0.047	0.052
1, 2, 4	0.59	**0.54**	0.60
1, 5 ,10	0.87	0.94	0.976
1, 10, 10	0.86	0.976	0.991

Lines beginning with #-signs are ignored when the code is executed. Thus, a log transformation can be performed by removing one #-sign, and a (more time consuming) rank transformation can be done by replacing that #-sign and removing another. (If no allowance were made for doing rank transformations, the program could have been written—perhaps a little more transparently for beginners—with three separate data matrices, one for each group. The result of using `apply` on the *rows*, designated by `1`, of `DTA` with the `rank` function yields a column vector of ranks for each row, hence the need for the transpose function `t`.)

Also in S-Plus. The same code can be run in S-Plus, but then the function `rowVars` could be used to make slight simplifications, so that the four lines

```
v1 <- rowVars(DTA[ , 1:r])
v2 <- rowVars(DTA[ , (r+1):(2*r)])
v3 <- rowVars(DTA[ , (2*r+1):(3*r)])
MSF <- r*rowVars(cbind(m1, m2, m3))
```

can replace the bolded lines of Display 5.

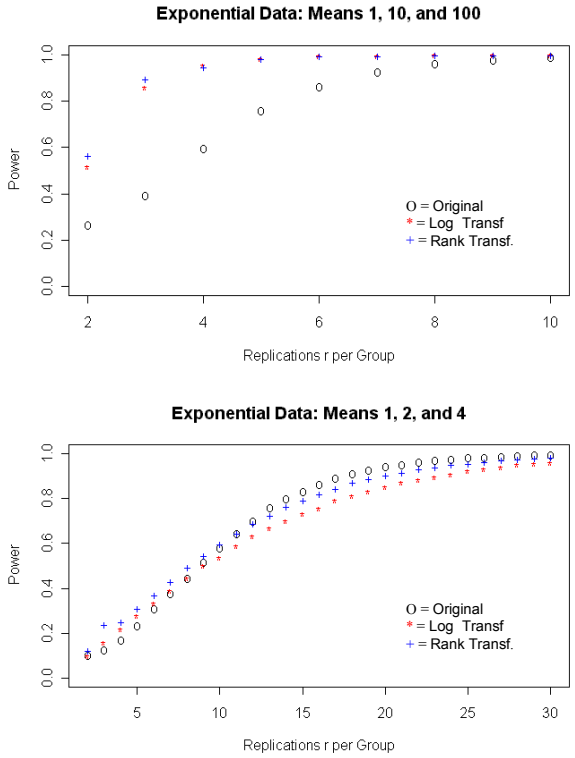
Results for log and rank simulations. In addition to log transformations, we consider rank transformations, where each of the tr observations is replaced by its rank. While rank transformations are often considered a cure for nonnormality, they also restrict the range of transformed data and hence diminish the opportunity for variances to be grossly different. (Exponential data can take arbitrarily large values, but their ranks must

take values between 1 and *tr*.) Because rank transformations sometimes work better for larger amounts of data, we consider ANOVAs with $r = 10$ as well as $r = 5$ replications in each of $t = 3$ groups.

Table 6 shows some simulation results for log and rank transformations on exponential data. The nominal significance level is 5%. Each value in the table is an average based on 10 runs of $m = 20\,000$ iterations each. Tests using the log transformations tend to have slightly smaller than the nominal 5% significance level. Tests with the rank transformations seem to result in slightly inflated significance levels, and consequently slightly better power.

For exponential data, both log and rank transformations can be beneficial when the sample size is small and the separation of group means is great. In particular, when there are $t = 3$ groups with $r < 10$ replications, we show some cases where the power is greatly improved by their use. (See Table 5 and Figures 7 and 8.) However, the lower panels of Figures 7 and 8 show situations in which it seems best not to transform the data—situations in which the power with original data is greater than the power with transformed data. (Also note the one bold entry in Table 6.) In the figures, each plotted point is based on $m = 20\,000$ simulated datasets.

Figure 7: Power at Various Sample Sizes
Log and rank transformation work best when *r* is small and population means are widely separated.

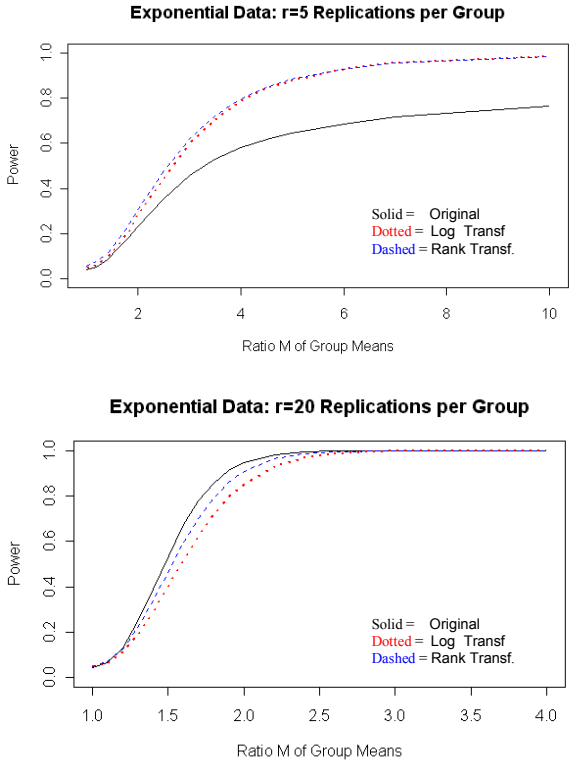


Comments and Acknowledgments.

The particular topic of this paper reinforces concepts of significance level and the relationship of power to sample sizes and alternatives. With R, S-Plus and Minitab running on computers currently in wide use, simulation projects such as the one shown here run at reasonable speed for classroom demonstrations; our SAS code runs much more slowly. See [6] for related computer code and teaching materials.

Thanks to Jaimyoung Kwon (who joins the CSU Hayward statistics faculty in Fall 2004) for the suggestion to broaden this study with the use of power curves. In part, this presentation and materials in [6] are based upon work supported under Rebecca Brafman's National Science Foundation Graduate Research Fellowship.

Figure 8: Power Against Various Alternatives
When $M = 1$, H_0 is true; when $M = 2$, the group means are 1, 2, 4; and when $M = 4$, the group means are 1, 4 , 16; etc.



References.

[1] G. Oehlert: *A First Course in Design and Analysis of Experiments,* Freeman (2000), Chapter 6.

[2] D. Montgomery: *Design and Analysis of Experiments,* 5th ed., Wiley (2001), Chapter 3.

[3] K. Brownlee: *Statistical Theory and Methodology in Science and Engineering,* 2nd ed., Wiley (1965). Chapter 3.

[4] H. Scheffé: *The Analysis of Variance,* Wiley 1959, Chapter 10.

[5] G. Snedecor and W. Cochran: *Statistical Methods,* 7th ed. Iowa State Univ. Press (1980), Chapter 15.

[6] Web pages including computer code and results for this paper: `www.sci.csuhayward.edu/~btrumbo/JSM2004/simtrans/`.