12-2022

Power Approximations for Generalized Linear Mixed Models in R Using Steep Priors on Variance Components

Sydney Geisler
Utah State University

POWER APPROXIMATIONS FOR GENERALIZED LINEAR MIXED MODELS IN R USING STEEP PRIORS ON VARIANCE COMPONENTS

by

Sydney Geisler

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Statistics

Approved:

John R. Stevens, Ph.D Major Professor	Yan Sun, Ph.D Committee Member
Brennan Bean, Ph.D Committee Member	Richard Cutler, Ph.D Vice Provost of Graduate Studies

Utah State University
Logan, Utah

2022

ABSTRACT

Power Approximations for Generalized Linear Mixed Models in R Using Steep Priors on Variance Components

Sydney Geisler

Utah State University, 2022

Major Professor: Dr. John R. Stevens
Department: Mathematics and Statistics

Many researchers are interested in power approximations from mixed effects models. The probability distribution method requires an exemplary dataset of a given sample size and holding constant the variance components to approximate power of the test for a specific fixed effect. Currently, the option to hold the variance components constant exists in SAS, but it is not available in R. We present here an R implementation for power approximation that will allow variance components to be essentially held constant by putting steep priors on those parameters. Examples of this will be shown with a Gaussian, binomial, and Poisson distribution of the response variables. The steep priors are called in a Bayesian linear mixed-effects model in R, where the resulting summary matches the corresponding SAS output. The end result of this thesis project is a generalized R function that can take on an exemplary dataset assuming a Gaussian family type, return the power approximation. The extension to generalized linear mixed models with non-Gaussian data is also discussed (36 pages).

Keywords: variance, components, steep, covariance, priors, R, exemplary, Bayesian

PUBLIC ABSTRACT

Power Approximations for Generalized Linear Mixed Models in R Using Steep Priors on Variance Components

Sydney Geisler

When designing an experiment, researchers often want to know how likely they are to detect statistically significant effects in the resulting data, i.e., they want to estimate their statistical power. The probability distribution method is a flexible way to do this, and it is currently implemented in the statistical software package SAS. This method requires a hypothetical data set (showing the magnitude of hypothesized effects) and constant values of variance components, which are critical elements of the statistical models used. The statistical software package R is increasingly popular, but the probability distribution method has not yet been implemented in R, and the statistical modeling approaches in R do not automatically allow constant values of the variance components. We present here an R implementation for power approximation that will allow variance components to be essentially held constant by assuming they follow a certain steep statistical distribution. We demonstrate this approach using normally-distributed data and explore issues in implementing it for non-normal data.

ACKNOWLEDGMENTS

I would like to start by giving thanks to my thesis advisor, Dr. John Stevens. He taught me much of what I know regarding mixed-effect models and power approximations, which were the basis of this thesis project. Dr. Stevens was very patient with me and was willing to do a lot of research on his own time in order to help me with R programming.

I would also like to express gratitude to my two thesis committee members, Dr. Brennan Bean and Dr. Yan Sun. Dr. Bean was the professor of STAT 6550, a statistical computing course which helped foster my knowledge of writing R functions, unit tests, and R package creation. In addition, Dr. Bean was able to help me publish the work I had done in R to GitHub, an online repository where the results of my thesis could become available to the general public. Dr. Sun taught STAT 5200, a design of experiments class which increased my depth of knowledge in SAS programming. I was also able to learn about different statistical designs in her class, which enhanced my understanding of the datasets that were used as part of this thesis project.

I appreciate the constant support of my parents Alisha and Lanny Geisler. They have always believed in me and my ability to succeed in my education no matter how difficult the task was at hand.

Lastly, I would like to thank Windstream, a telephone and internet service provider for whom I will be interning for over the summer. My experience with

them has been nothing but great so far and they have extended the possibility of a full-time position once my internship is complete.

CONTENTS

Page

LIST OF FIGURES

Figure Page

BACKGROUND

In an experimental design for a mixed-effects model, researchers are often interested in estimating the power of the tests they will be running. This is an important calculation because the power tells the experimenter the probability that a treatment effect of interest can be detected if it really exists. If the power is too low, a treatment effect will likely go undetected, and the experiment will not be worth the effort. If the power is higher than necessary, a lot of money can be wasted by involving more sample subjects than what is necessary to detect a statistical effect. Researchers often want to estimate what the power is before starting an experiment to ensure that their time and money is spent in the most efficient way possible.

Power approximations will be calculated through the probability distribution method. This approach to power approximation is computationally fast, flexible to any experimental design, and useful for any fixed effect in any generalized linear mixed model. This method may be used anytime the sampling distribution of the test statistic can be estimated through the conditions of the research hypothesis (Gbur et al. 2012). In the examples that will be shown below, the F-distribution will be used as a baseline for power estimation.

The following quote from the book titled *Analysis of Generalized Linear Mixed Models in the Agricultural and Natural Resources Sciences* (Gbur et al. 2012) gives a brief overview of the history of power approximations using the probability distribution method: “This basic approach originated with linear models using PROC GLM in SAS (Littell, 1980; O’Brien and Lohr, 1984; Lohr and O’Brien, 1984). Stroup

(1999) extended the method to linear mixed models using PROC MIXED. Stroup (2002) described the implementation of the probability distribution method for linear mixed models using PROC MIXED, focusing on experiments in the presence of spatial variation, and provided evidence of the accuracy of these methods via simulation. Littell et al. (2006) provided additional detail and examples for linear mixed models."

Three motivating examples will be examined for this power approximation. Each example involves the creation of an exemplary dataset, which is a dataset that mimics how the experiment would look in practice if the effect of a particular fixed effect term had at least some specified minimum magnitude. Exemplary datasets allow the power to be estimated before the experiment is conducted, which is why they are so valuable to researchers. A full description of each dataset will be given that details the values and significance of each explanatory variable as well as the family distribution of the response variable. Only relevant output will be displayed at each step as there are a lot of graphical and numerical summaries generated in both SAS and R, the two statistical programming languages that will be examined in this thesis. All coding that was required to generate the specified plots and values are included in the following GitHub repository: https://github.com/sydneykgeisler/power_mm.git

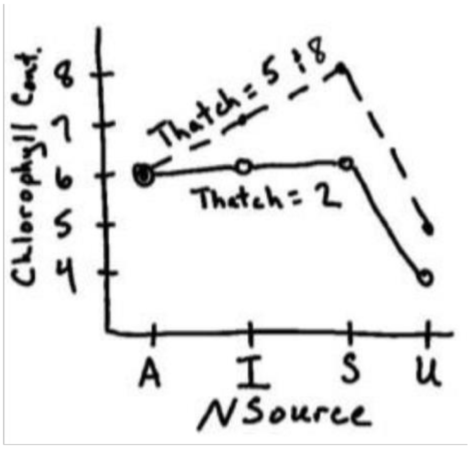
BACKGROUND: THREE MOTIVATING EXAMPLES

For the first example, consider a study where researchers are interested in the effects of a nitrogen source, field, and years of thatch accumulation on the chlorophyll content (mg/g) of grass clippings (Kuehl 1994, example 14.1). The four sources of nitrogen that will be investigated are denoted as "Urea", "AmmSulph", "IBDU", and "SCUrea". The three levels of thatch accumulation that will be tested are 2, 5, and 8 years. Each field is divided into four parts, with each part randomly assigned a nitrogen source level. Then each part is divided into three sub-regions, with each sub-region randomly assigned a level of thatch accumulation. This is a split plot design. The estimated y-value, which comes from the Gaussian family distribution, is the anticipated average chlorophyll measurement that an experimenter would expect to see given the nitrogen source, thatch level, and field number. For two years of thatch accumulation, researchers expect the average chlorophyll content to be 6 mg/g for nitrogen sources of "AmmSulph", "IBDU", and "SCUrea" and 4 mg/g for "Urea". For either five or eight years of thatch accumulation, the expected chlorophyll content for "AmmSulph", "IBDU", "SCUrea", and "Urea" is 6, 7, 8, and 5 mg/g, respectively.

In other words, for this motivating example, suppose that researchers hypothesize that the true interaction effect of nitrogen source and thatch is as least as large as represented in Figure 1.

Figure 1

A Plot Showing an Expected Interaction between Nitrogen Source and Thatch Levels



The goal for this example is to approximate the power of the test for the interaction term given that this interaction between nitrogen source and thatch level exists.

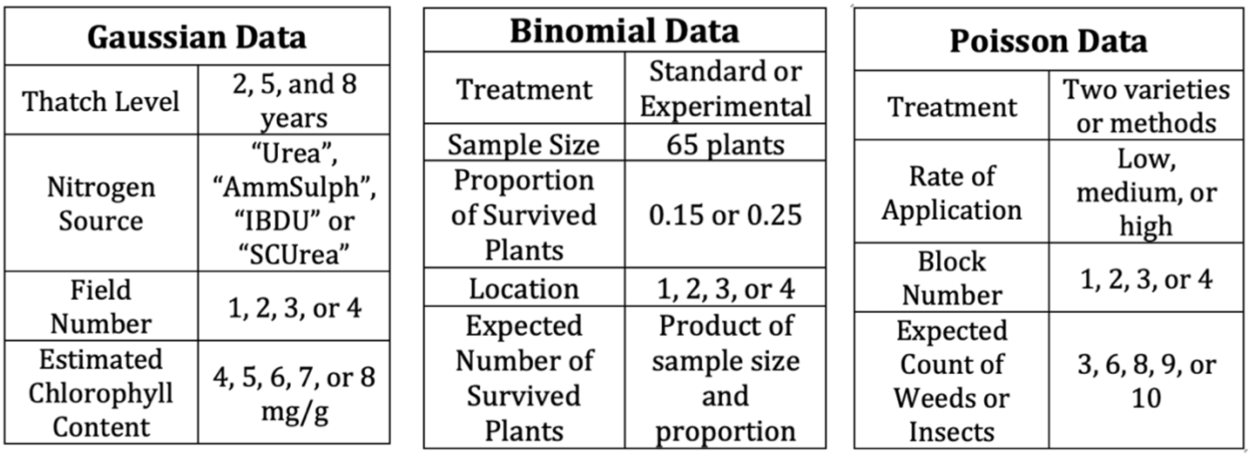
The second example (Gbur et al. 2012) seeks to compare the effectiveness of a standard treatment and experimental treatment on plants that have been exposed to a certain disease. Researchers believe the plants that are given the standard treatment have a 15% survival rate, while the plants that are given the experimental treatment will see the survival percentage increase to 25%. The treatment comparisons will be observed at four different locations with 65 plants each. There will be five total variables used in this experiment: 'trt', 'n', 'pi', 'location', and 'expected y'. The 'trt' variable is denoted as '0' for the standard treatment and '1' for the experimental treatment. The use of 'n' refers to the sample size at each treatment location. The 'pi' variable is the proportion of plants expected to survive the disease, notated either 0.15 for the standard treatment or 0.25 for the experimental treatment. The location has factors 1 through 4 to represent the treatment location. The response variable, ‘expected y’, is the product of ‘n’ and ‘pi’ and denotes the expected *number* of plants that will survive the disease. This expected number follows a binomial distribution. The goal for this example is to estimate the power of the test for the treatment effect given that the experimental treatment increases plant survival to 25% versus 15% for the standard treatment.

The last example that will be explored is a split plot design of a field experiment (Gbur et al. 2012). This exemplary dataset represents several kinds of treatment options that researchers might find useful in agronomic or land management practices. The variables used in the exemplary dataset are ‘trt’, ‘rate’, ‘block’, and ‘exp_count’. The treatment variable has factors 1 and 2 representing either two levels of application, two tillage methods, or two varieties in the treatment. The rate variable represents a rate of pesticide use or irrigation level, and it is a factor with numbers 1, 2, and 3, denoting levels ‘low’, ‘medium’, and ‘high’ respectively. The block variable is also a factor with numbers 1 through 4 to denote the number of blocks to be used within each experimental unit. Lastly, ‘exp_count’ represents the mean anticipated response, which could represent either the number of insects or weeds. For treatment 1, the expected counts are 10, 9, and 8 for rate levels ‘low’, ‘medium’, and ‘high’, respectively. For treatment 2, these counts are 9, 6, and 3, likewise for rate application levels of ‘low’, ‘medium’, and ‘high’, respectively. These counts are discrete and follow a Poisson distribution. The goal of this example is to approximate the power of the test for the rate variable given that these differences in average values of weed or insect counts really exist.

The tables in Figure 2 summarize the variables and values of each exemplary dataset defined above.

Figure 2

Summaries of Each Exemplary Dataset



Gaussian Data	
Thatch Level	2, 5, and 8 years
Nitrogen Source	"Urea", "AmmSulph", "IBDU" or "SCUrea"
Field Number	1, 2, 3, or 4
Estimated Chlorophyll Content	4, 5, 6, 7, or 8 mg/g

Binomial Data	
Treatment	Standard or Experimental
Sample Size	65 plants
Proportion of Survived Plants	0.15 or 0.25
Location	1, 2, 3, or 4
Expected Number of Survived Plants	Product of sample size and proportion

Poisson Data	
Treatment	Two varieties or methods
Rate of Application	Low, medium, or high
Block Number	1, 2, 3, or 4
Expected Count of Weeds or Insects	3, 6, 8, 9, or 10

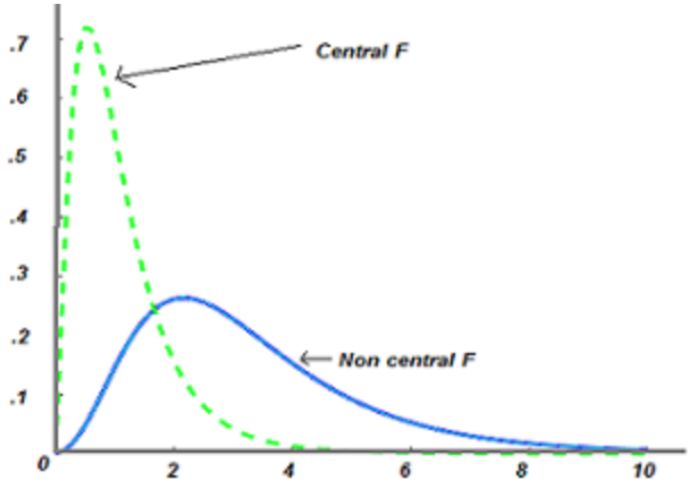
METHODS IN SAS

Each research question addressed above will first be evaluated in SAS. Here, the exemplary datasets are created, with all observations for fixed effects levels combinations held constant at values consistent with the minimum hypothesized effect magnitude (as in Figure 1 above). Once this initial step is completed, the GLIMMIX procedure can be used to run a mixed-effect analysis on the data. Here, the fixed and random effects are identified. A fixed effect is a term in the model whose levels are the only levels of interest to researchers, while a random effect is a term in the model whose levels are representative of some population of all possible levels for that term. The GLIMMIX procedure also introduces another critical step in power approximation: holding variance components constant within the model. If experimenters have some idea of what the variances of each random effect are beforehand or want to test worst-case-scenario values, then these can be specified in the ‘PARMS’ statement of the GLIMMIX procedure and will then appear as the variance components of the mixed-effect model analysis. One of the reasons that holding these variance components constant is important is because the exemplary dataset has no variation within fixed effects combinations, i.e., there is no residual variance unless the error variance component is held constant at some positive value.

As stated previously, the probability distribution method will be implemented through the F-distribution to approximate power. The F-values of the fixed effects are calculated through the GLIMMIX procedure. Researchers can choose any of these for the power approximation depending on which effects interest them the most. The F-value of the corresponding fixed effect of interest is then used as the sample statistic. After this statistic is obtained, a non-centrality parameter will be calculated next. This is a parameter that shifts the F-distribution further to the right if its value is greater than zero (Stephanie, 2017). Non-centrality parameters are calculated by multiplying the sample F-statistic by its numerator degrees of freedom, which can also be obtained through the GLIMMIX procedure. Figure 3 (Stephanie, 2017) shows an illustration of this effect.

Figure 3

A Non-central F-distribution is Shifted Further Right than an F-distribution Centered at Zero



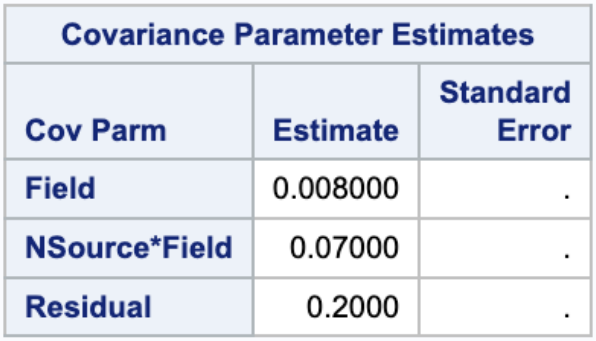
The third step of obtaining the power approximation is to calculate a critical value through a pre-specified significance level (usually 0.05) as well as the numerator and denominator degrees of freedom. For the last step, the area to the right of the sample statistic in the non-central F-distribution is obtained. This value is the power approximation (Gbur et al. 2012).

The fixed effects for the grass exemplary dataset are the nitrogen source, thatch level, and the interaction between nitrogen source and thatch level. The random effects are the nitrogen source, field number, and the interaction between the two. Due to a previous pilot study, it is believed that the variance component estimates for the field and field by nitrogen source interaction are 0.008 and 0.07, respectively. The pilot study estimate for the residual error variance is 0.2. Figures 4-5 show some of the SAS output from the GLIMMIX procedure; see the GitHub link which contains the SAS code used in this thesis:

https://github.com/sydneykgeisler/power_mm.git

Figure 4

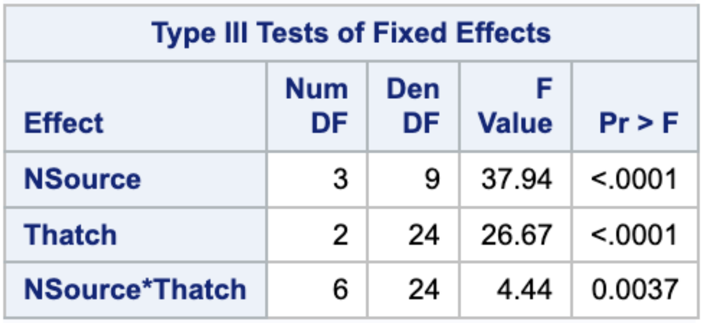
The Covariance Parameter Estimates Were Held Constant within the Gaussian Model



Covariance Parameter Estimates		
Cov Parm	Estimate	Standard Error
Field	0.008000	.
NSource*Field	0.07000	.
Residual	0.2000	.

Figure 5

The ANOVA Table Provided for the Fixed Effects of the Gaussian Model

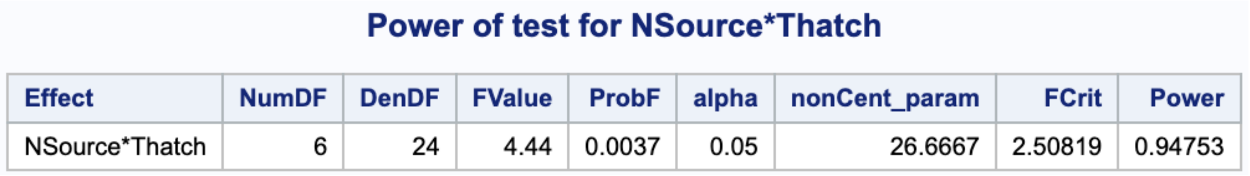


Type III Tests of Fixed Effects				
Effect	Num DF	Den DF	F Value	Pr > F
NSource	3	9	37.94	<.0001
Thatch	2	24	26.67	<.0001
NSource*Thatch	6	24	4.44	0.0037

For the grass dataset example, the experimenters wanted to calculate the power based on the nitrogen source and thatch level interaction and hence used the F-statistic of 4.44 for the power approximation. Figure 6 shows the power that was estimated by SAS.

Figure 6

The SAS Power Approximation for the Gaussian Data



Power of test for NSource*Thatch

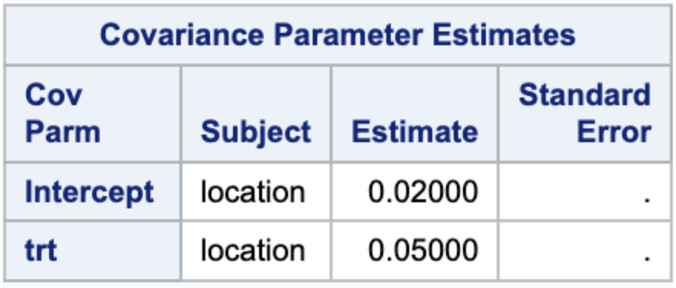
Effect	NumDF	DenDF	FValue	ProbF	alpha	nonCent_param	FCrit	Power
NSource*Thatch	6	24	4.44	0.0037	0.05	26.6667	2.50819	0.94753

The power of the nitrogen source and thatch level interaction test is about 0.95, meaning that the chance of detecting an interaction between nitrogen source and thatch levels at the specified chlorophyll content values (as shown in Figure 1) was 95%.

For the plant binomial dataset, the fixed effect is the treatment type. The random effects are the location and treatment by location interaction term. The “best guess” estimates for each covariance parameter are 0.02 for the location and 0.05 for the treatment and location interaction. Figures 7 and 8 show the results obtained in SAS, similar to the grass dataset.

Figure 7

The Covariance Parameters Were Held Constant within the Binomial Model

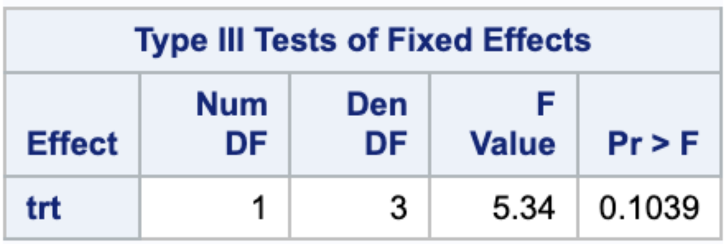


Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error
Intercept	location	0.02000	.
trt	location	0.05000	.

Figure 8

The ANOVA Table Provided for the Fixed Effect of the Binomial Model

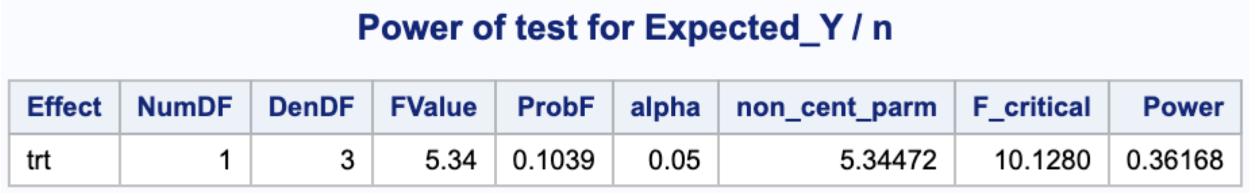


Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
trt	1	3	5.34	0.1039

Figure 9

The SAS Power Approximation for the Binomial Data



Power of test for Expected_Y / n

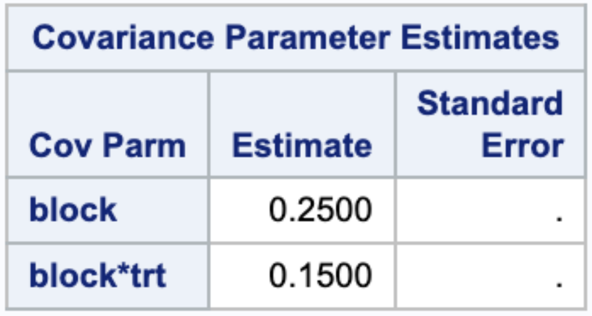
Effect	NumDF	DenDF	FValue	ProbF	alpha	non_cent_parm	F_critical	Power
trt	1	3	5.34	0.1039	0.05	5.34472	10.1280	0.36168

We can see that for the plant exemplary dataset, we only have a 36% chance of detecting the pre-specified difference in treatments.

Lastly, the split plot design with the Poisson family distribution has fixed effects of treatment type, rate of application, and the interaction term between these two variables. The random effects are the block and block by treatment interaction term. Researchers have estimated the block variance to be about 0.25, while the block and treatment interaction term has a variance of 0.15. The fixed effect variable of interest for this experiment is the rate of application. The results from SAS are shown in Figures 10-12.

Figure 10

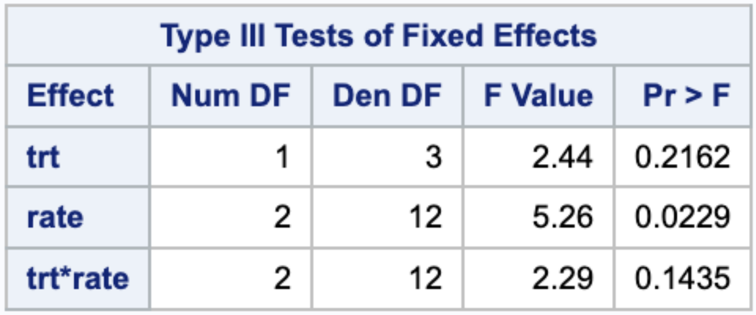
The Covariance Parameter Estimates Were Held Constant within the Poisson Model



Covariance Parameter Estimates		
Cov Parm	Estimate	Standard Error
block	0.2500	.
block*trt	0.1500	.

Figure 11

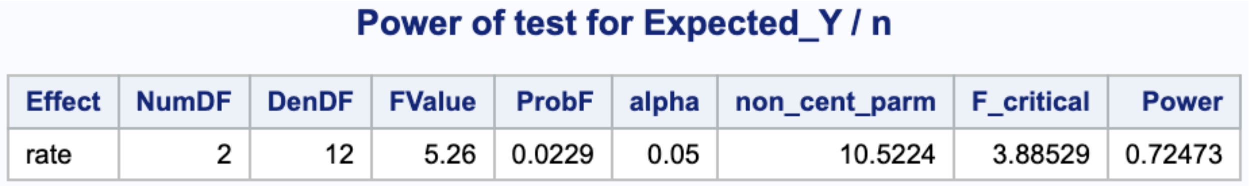
The ANOVA Table Provided for the Fixed Effects of the Poisson Model



Type III Tests of Fixed Effects				
Effect	Num DF	Den DF	F Value	Pr > F
trt	1	3	2.44	0.2162
rate	2	12	5.26	0.0229
trt*rate	2	12	2.29	0.1435

Figure 12

The Power Approximation for the Poisson Data



Power of test for Expected_Y / n

Effect	NumDF	DenDF	FValue	ProbF	alpha	non_cent_parm	F_critical	Power
rate	2	12	5.26	0.0229	0.05	10.5224	3.88529	0.72473

For this exemplary dataset, there is about a 72% chance that the differences in rate application can be detected.

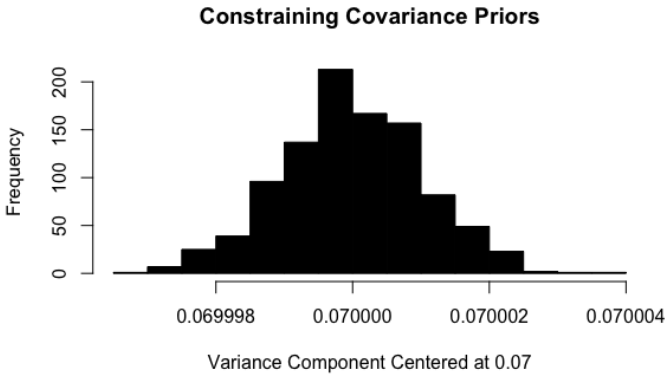
While there are current programming capabilities available in SAS to estimate the power of a mixed model procedure (as demonstrated above), it would also be of interest to be able to calculate these same values in R. This is in part due to the fact that R is a free software available to the public, while a SAS license can be quite expensive. The learning curve in R can also be less steep than in SAS, since many less lines of code are sometimes required in R to run similar functions. However, R is not currently capable of calculating a power approximation as was just demonstrated in SAS. This is because the package that produces mixed-effect model outputs, 'lme4' (Bates et al. 2015), does not have an option to specify variance components to be held constant. This issue is the motivation this thesis project: to develop a single function in R that can run a mixed effects model analysis and fix variance values in the model. It will then be shown that the power approximation obtained in R matches that of what SAS would produce.

METHODS IN R

In R, an alternative to 'lme4' (Bates et al. 2015) is the 'blme' package (Chung et al. 2013), which uses a Bayesian approach to fitting linear and generalized linear mixed models. The key difference between these two methods is that the Bayesian method uses a summary statistic such as the mean or mode of a prior distribution as an estimate for variance components, whereas the frequentist approach of 'lme4' (Bates et al. 2015) uses maximum likelihood estimation for components rather than priors (Hackenberger, 2019; Brownlee, 2019). Priors are useful for power approximations because they allow values to be imposed on variance components of the random effects to a high degree of accuracy. This is done by creating normal density functions for each random effect that are centered around the corresponding variance component with a standard deviation of 1×10^{-6} . The Bayesian method of 'blme' (Chung et al. 2013) then estimates the mode of each distribution as the variance component specified, which essentially holdss these values constant within the model. These functions are referred to as covariance priors. The 'blmer' (Chung et al. 2013) function can take in as many covariance priors as there are random effects, and residual variances can be specified in these statements if they are included in the model. For the grass exemplary dataset, the 'dnorm' function (R Core Team, 2020) was used to create the two density functions: one for the field random effect and the other for the field by nitrogen source random effect. Figure 13 demonstrates the covariance prior that was used for the field by nitrogen source interaction.

Figure 13

The Density Function Tightly Centers around the Variance Component



These functions are then referenced within the ‘cov.prior’ statement of the ‘blmer’ function (Chung et al. 2013) to hold the desired variance components constant. Because the steep covariance priors were calculated outside of the ‘blmer’ function, they would need to be referred to as a custom function inside the ‘cov.prior’ statement rather than one of the other built-in functions. It should be noted that as many covariance priors can be used as there are random effects specified in the formula statement. If multiple variance components are to be fixed in the model such as with the grass exemplary dataset, the ‘list’ option within ‘cov.prior’ should be used to separate each custom density function with a comma. Otherwise, the ‘cov.prior’ statement should just be set equal to the single custom function defined.

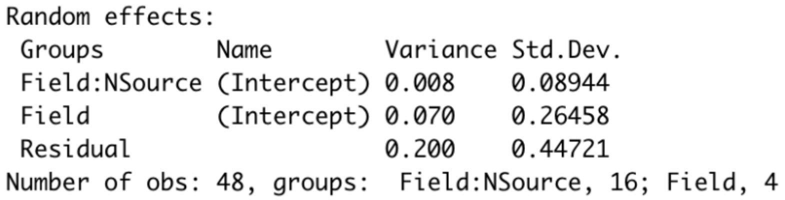
Similar to the power approximation in SAS, the F-values and degrees of freedom can be extracted from the ANOVA table of this object to calculate the non-centrality parameter, critical F-value, degrees of freedom, and power approximation once the mixed-effect model is calculated (Gbur *et al., 2012)*.

A function called ‘power_mm’ was created for this thesis project as a way to generalize the process of fitting a ‘blmer’ object (Chung et al. 2013) and estimate the power through the probability distribution method of Gbur et al. (2012). A user will need to supply the model formula with specified fixed and random effects, the exemplary dataset, variance components, family distribution type, significance level, and fixed effect of interest necessary to calculate the power approximation. The output of this function will be a single value representing the estimation of power.

Before demonstrating the use of ‘power_mm’, it will be useful to show that the variance components are, in fact, held constant within ‘blmer’ (Chung et al. 2013). Figure 14 shows the summary of the random effects from the grass exemplary dataset:

Figure 14

Partial Output of the Gaussian Model Summary in R

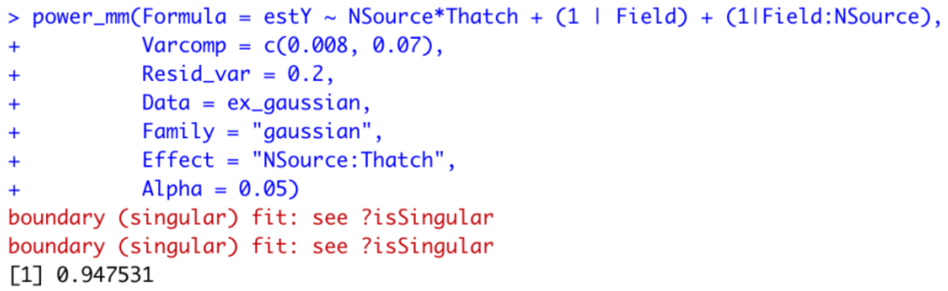


```
Random effects:
 Groups        Name        Variance Std.Dev.
 Field:NSource (Intercept) 0.008    0.08944
 Field         (Intercept) 0.070    0.26458
 Residual                  0.200    0.44721
Number of obs: 48, groups:  Field:NSource, 16; Field, 4
```

We can see that the variance components match those of Figure 4, which come from the GLIMMIX procedure in SAS. Next, it can be shown (see Figure 15) that the ‘power_mm’ R function produces a power approximation that is identical to the SAS output of Figure 6. The full source code of the ‘power_mm’ is provided in the following GitHub link: https://github.com/sydneykgeisler/power_mm.git

Figure 15

The Results of the power_mm Function



```
> power_mm(Formula = estY ~ NSource*Thatch + (1 | Field) + (1|Field:NSource),
+          Varcomp = c(0.008, 0.07),
+          Resid_var = 0.2,
+          Data = ex_gaussian,
+          Family = "gaussian",
+          Effect = "NSource:Thatch",
+          Alpha = 0.05)
boundary (singular) fit: see ?isSingular
boundary (singular) fit: see ?isSingular
[1] 0.947531
```

In Figure 15, the ‘Formula’ component takes the first argument specified as the response variable, which is ‘estY’ in this case. The arguments following represent the explanatory variables. A random effect is denoted with parenthesis as with the '(1 | Field)’ and ‘(1 | Field:NSource)’ statements. Fixed effects do not have these parentheses. Interaction random effects terms are represented with a colon between the two variables, such as with ‘Field’ and ‘NSource’. The ‘Varcomp’ argument of this function represents the values at which to hold the variance components constant. ‘Resid_var’ is null by default but can contain a residual variance value if it is provided. The ‘Data’ statement references the exemplary dataset, or ‘ex_gaussian’ for the grass example. The ‘Effect’ argument is specified by the user and can be any fixed effect that was written in the ‘Formula’ statement. Lastly, ‘Alpha’ is a significance level and the default value is 0.05.

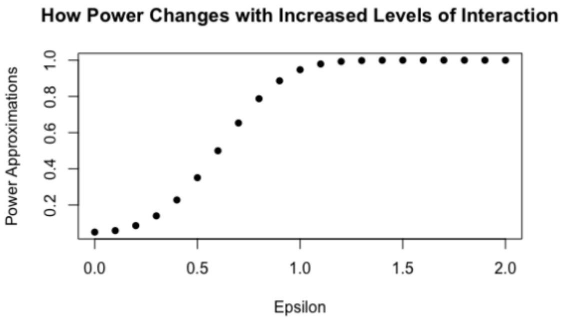
Although some warnings are generated involving the singularity of the dataset as seen above, this is not a concern for the power estimation. This is because the exemplary dataset does not have any variability in it due to the way it is constructed. Recall that the data is only a representation of what researchers expect to see in an experiment and does not contain any information from a true study. The variability will be imposed on the model through the use of steep covariance priors and thus does not need to be present in the exemplary dataset. Therefore, any warnings related to singularities in the optimization procedure can be ignored.

In addition to the coding for ‘power_mm’, a unit test is also included in the GitHub page. This test shows how increasing levels of interaction from the plot in Figure 1 lead to an increase in statistical power. Consider first that at levels ‘I’ and

'U' on the Figure 1 plot have a distance of $\varepsilon = 1$ between the lines of thatch 2 and thatch 5 & 8. However, at level 'S', this distance is equal to ε =2. If these lines had a distance of ε =0 and were completely parallel, this would signify that there is no interaction between nitrogen source and thatch level. The power in this case will only be equal to the significance level of 0.05. If ε is increased to 0.1, that means that the two lines of thatch will have a distance of 0.1 at levels 'I' and 'U' and 0.2 at level 'S'. This scenario will see an increase in power compared to the last example of $\varepsilon = 0$ since a small interaction effect is now present. This process was repeated iteratively by increasing levels of 0.1 until ε = 2, and the end result of this unit test is 21 power approximations. The power approximations for this unit test were calculated through the 'power_mm' function and stored in a matrix. Figure 16 showcases the relationship between power estimations and values of ε.

Figure 16

Increasing Interaction (ε) Increases the Power Approximation

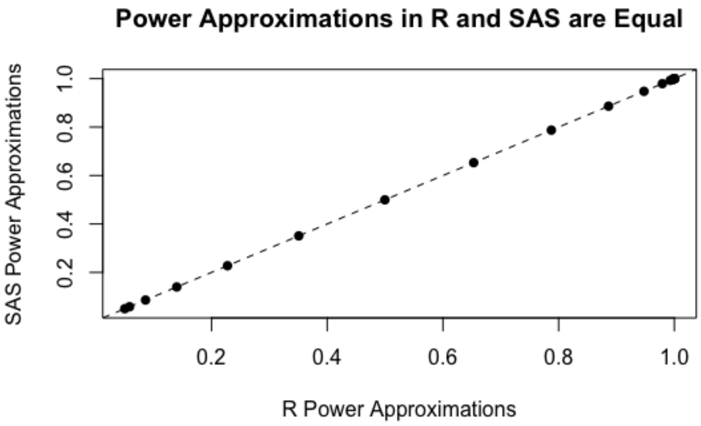


At a point of about $\varepsilon = 1.3$, we can see that the power estimation is virtually equal to 1, meaning that the ability to detect an interaction between nitrogen source and thatch level for the grass experiment is almost guaranteed.

The second portion of this unit test ensures (see Figure 17) that the results from ‘power_mm’ match the power approximations that are calculated in SAS. The reason for this is that R packages such as ‘blme’ (Chung et al. 2013) are subject to updates, whereas SAS procedures like GLIMMIX are never changed and thus should always produce the same results. If ‘blme’ (Chung et al. 2013) is ever changed in such a way that ‘power_mm’ no longer produces power approximations consistent with SAS, then the unit test will fail and users will be warned that the results are inaccurate. Figure 17 shows a scatterplot of power approximations in SAS and R for every value of ε given the current version of ‘blme’ (Chung et al. 2013).

Figure 17

R vs. SAS Power Approximations



The x and y coordinates for each point shown on the Figure 17 plot are the same, meaning that the SAS and R power approximations are equivalent.

While this R function can produce power estimations from Gaussian data that match SAS output exactly, this has not yet proven to be the case with non-Gaussian data. All errors and output that occurred are shown in the appendix. For now, brief descriptions of coding errors and possible reasons for them will be discussed.

With the binomial and Poisson exemplary datasets discussed above, there were convergence and class inheritance issues when used with the steep covariance prior method. One reason for this with the binomial data is that the estimated number of plants that survive, which is the product of the sample size and probability of survival, is not always an integer value. Since R requires whole numbers to be used for binomial data, this causes a conflict with a coding. It is not known why this error occurs with Poisson data.

These errors can be removed by using gamma density functions for the covariance priors instead of the normal distribution. However, the resulting variance components are not fixed at the values specified in the 'power_mm' function. This is likely because gamma functions are skewed to the right and do not center around the average value of the distribution. Decreasing the standard deviation to a value like 1×10^{-6} as was used for Gaussian data constrains the gamma distribution to zero rather than the values set for the covariance priors. Research into these different errors led to a GitHub page by Vincent Dorie, describing an optimization algorithm for fixing covariance priors into the model. This technique was implemented by applying 'glmer' (Bates et al. 2015) to the exemplary dataset and returning only the deviance evaluation function. This results in only the model parameters being calculated instead of the entire model summary. The ‘optim’ function (R Core Team, 2020) was then used to obtain coefficient estimates of the random effects given that the variances were held constant at a particular value. These values could then be installed into a ‘glmer’ (Bates et al. 2015) object by fixing them into the same working environment that houses the deviance function. Doing this, variance components *were* held constant at the desired values.

The issue with this optimization method occurs when extracting F-statistics from each ANOVA table to then use for power estimation. These values were not consistent with the F-values produced in SAS, meaning that resulting power approximations would also be inaccurate. The package developers of ‘lme4’ (Bates et al. 2015) recommended the use of the ‘Anova’ function from the ‘car’ package (John Fox and Sanford Weisberg, 2019) in order to specify whether a type-II or type-III ANOVA test should be performed on the ‘glmer’ (Bates et al. 2015) objects. While this was able to bring the F-statistics closer to those provided in SAS, the results still did not match. Increasing the sample sizes in both the binomial and Poisson exemplary data also brought the R calculations for F-statistics closer to those computed in SAS, but this likewise did not yield consistent results. This defeats the purpose of experimental design as well when sample sizes need to be arbitrarily large to obtain similar effects in both programming languages, so this is not practical in real life experimental design.

DISCUSSION

Due to the inaccuracy of both the steep covariance prior and optimization methods on non-Gaussian data in R, it is not possible at this time to provide power approximations for data from these kind of family distributions using the 'power_mm' function that was created for this thesis project. In order for this issue to be resolved, the generalized mixed-model procedure in R would need to be examined more closely to determine where this inaccuracy is occurring and, more importantly, how to fix it so that the power approximation can be fully generalized for all available family types in R and SAS. The 'power_mm' function would then need to be further generalized to accommodate all the distribution types available to use in 'bglmer' (Chung et al. 2013).

While 'power_mm' is only currently available for Gaussian data, it is able to provide power approximations that mirror the results in SAS to at least five decimal places. This function has been published to GitHub and it is available to the public at https://github.com/sydneykgeisler/power_mm.git. The exemplary datasets for the Gaussian, binomial, and Poisson examples discussed in this report were coded and loaded into this same repository.

CONCLUSIONS

In conclusion, the ‘blme’ package (Chung et al. 2013) can be used along with the concept of steep covariance priors to hold variance components constant in a mixed-effects model in R on Gaussian data. Custom functions are created using normal density functions to center a distribution tightly around a variance component value, and these functions serve as the steep priors within the ‘blmer’ function (Chung et al. 2013). This method is not currently reliable for non-Gaussian data as the results in R do not closely match those of what SAS would produce. Any future work on this project would involve examining the computational differences in generalized linear mixed-effect calculations between R and SAS in order to obtain accurate power approximations in both statistical programming languages.

REFERENCES

Douglas Bates, Martin Maechler, Ben Bolker, Steve Walker (2015). Fitting Linear Mixed-Effects Models Using lme4. Journal of Statistical Software, 67(1), 1-48. doi:10.18637/jss.v067.i01

Chung Y, Rabe-Hesketh S, Dorie V, Gelman A, Liu J (2013). "A nondegenerate penalized likelihood estimator for variance parameters in multilevel models." _Psychometrika_, *78*(4), 685-709. doi: 10.1007/s11336-013-9328-2 (URL: https://doi.org/10.1007/s11336-013-9328-2).

Kuehl (1994). "Statistical Principles of Research Design and Analysis" example 14.1.

Gbur et al. (2012). "Power Calculations in Mixed Models".

R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Dorie, V. (2021, February 24). *custom prior for bglmer #13*. GitHub. URL: https://github.com/vdorie/blme/issues/13

Littell, R.C., G.A, Milliken, W.W. Stroup, R.D. Wolfinger, and O. Schabenberger. 2006. SAS for mixed models. 2nd ed. SAS Institute, Cary, NC.

John Fox and Sanford Weisberg (2019). An {R} Companion to Applied Regression, Third Edition. Thousand Oaks CA: Sage. URL: https://socialsciences.mcmaster.ca/jfox/Books/Companion/

Stephanie (2019). Non Centrality Parameter (NCP). Statistics How To. URL: https://www.statisticshowto.com/non-centrality-parameter-ncp/

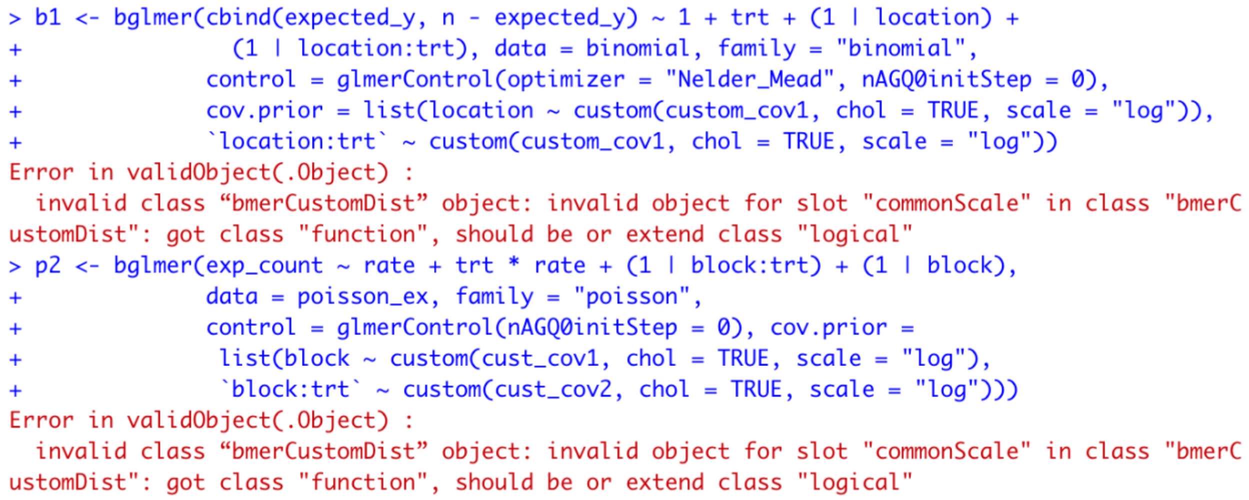
Brownlee, J. (2019, November 8). *A Gentle Introduction to Maximum a Posteriori (MAP) for Machine Learning.* Machine Learning Mastery. URL: **https://machinelearningmastery.com/maximum-a-posteriori-estimation/**

Hackenberger, B. K. (2019, February). *Bayes or not Bayes, is this the question?* National Library of Medicine. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6406060/

APPENDIX

Figure A1

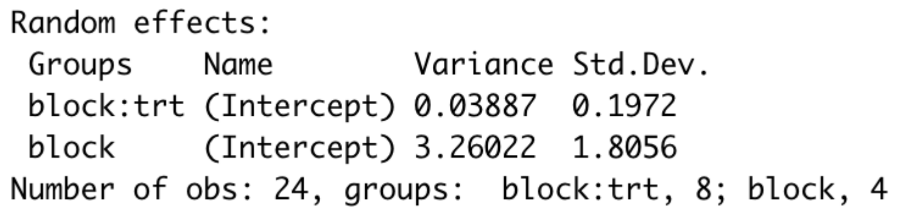
These Errors Occur with power_mm Function on Non-Gaussian Data



```
> b1 <- bglmer(cbind(expected_y, n - expected_y) ~ 1 + trt + (1 | location) +
+                (1 | location:trt), data = binomial, family = "binomial",
+              control = glmerControl(optimizer = "Nelder_Mead", nAGQ0initStep = 0),
+              cov.prior = list(location ~ custom(custom_cov1, chol = TRUE, scale = "log")),
+              `location:trt` ~ custom(custom_cov1, chol = TRUE, scale = "log"))
Error in validObject(.Object) : 
  invalid class “bmerCustomDist” object: invalid object for slot "commonScale" in class "bmerC
ustomDist": got class "function", should be or extend class "logical"
> p2 <- bglmer(exp_count ~ rate + trt * rate + (1 | block:trt) + (1 | block),
+              data = poisson_ex, family = "poisson",
+              control = glmerControl(nAGQ0initStep = 0), cov.prior =
+               list(block ~ custom(cust_cov1, chol = TRUE, scale = "log"),
+               `block:trt` ~ custom(cust_cov2, chol = TRUE, scale = "log")))
Error in validObject(.Object) : 
  invalid class “bmerCustomDist” object: invalid object for slot "commonScale" in class "bmerC
ustomDist": got class "function", should be or extend class "logical"
```

Figure A2

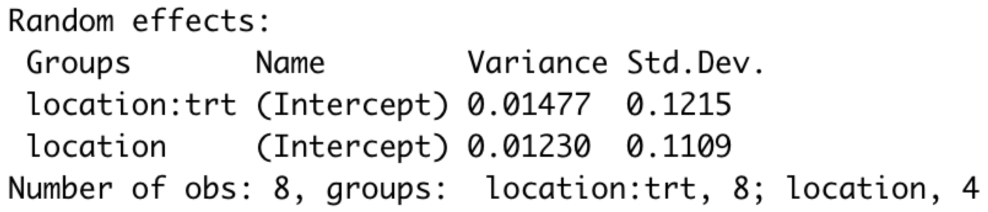
Variance Components are not Held Constant at the Desired Values for Poisson Data



```
Random effects:
 Groups    Name        Variance Std.Dev.
 block:trt (Intercept) 0.03887  0.1972  
 block     (Intercept) 3.26022  1.8056  
Number of obs: 24, groups:  block:trt, 8; block, 4
```

Figure A3

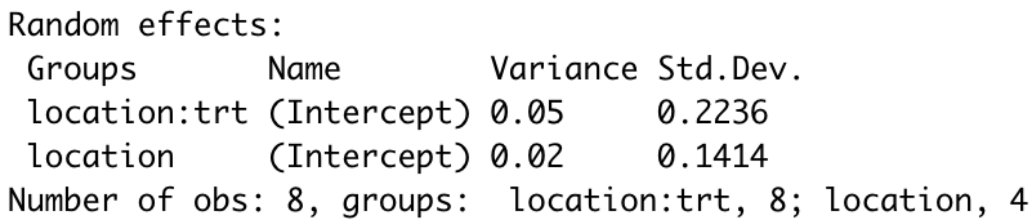
Variance Components are not Held Constant at the Desired Values for Binomial Data



```
Random effects:
 Groups       Name        Variance Std.Dev.
 location:trt (Intercept) 0.01477  0.1215  
 location     (Intercept) 0.01230  0.1109  
Number of obs: 8, groups:  location:trt, 8; location, 4
```

Figure A4

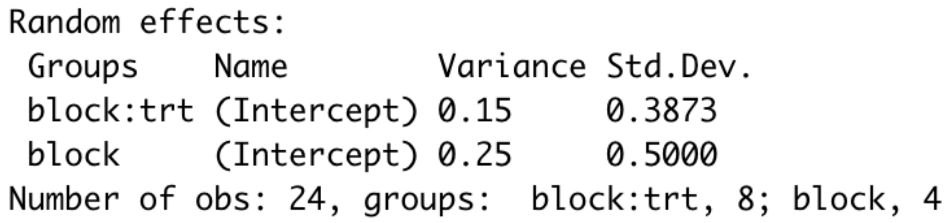
The Variance Components Using Optimization Method on Binomial Data



```
Random effects:
 Groups       Name        Variance Std.Dev.
 location:trt (Intercept) 0.05     0.2236
 location     (Intercept) 0.02     0.1414
Number of obs: 8, groups:  location:trt, 8; location, 4
```

Figure A5

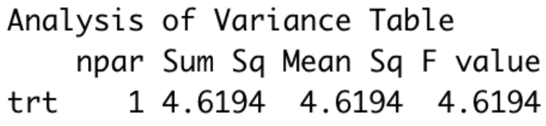
The Variance Components Using Optimization Method on Poisson Data



```
Random effects:
 Groups    Name        Variance Std.Dev.
 block:trt (Intercept) 0.15     0.3873
 block     (Intercept) 0.25     0.5000
Number of obs: 24, groups:  block:trt, 8; block, 4
```

Figure A6

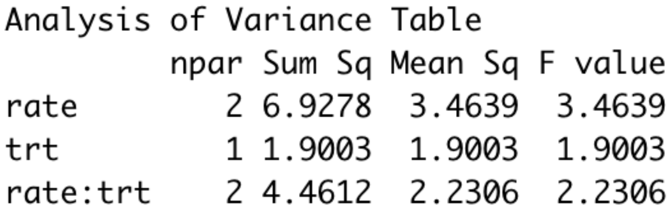
The ANOVA Results Do Not Match Figure 7



```
Analysis of Variance Table
    npar Sum Sq Mean Sq F value
trt    1 4.6194  4.6194  4.6194
```

Figure A7

The F-statistic Does Not Match Figure 10



```
Analysis of Variance Table
         npar Sum Sq Mean Sq F value
rate        2 6.9278  3.4639  3.4639
trt         1 1.9003  1.9003  1.9003
rate:trt    2 4.4612  2.2306  2.2306
```