

Robustness of Degrees of Freedom Approximation in Linear Mixed Models with Nonnormal Distributions

Jessica Yu
APRIL 20, 2022

Submitted to the Department of
Mathematics and Statistics
of Amherst College in partial fulfillment
of the requirements for the degree of
Bachelor of Arts with honors.

ADVISOR:
Dr. Brittney Bailey

Abstract

Linear mixed-effects models are widely used in social and biological sciences, and are also used for longitudinal data analysis. In cases where sample sizes are small, evaluating the significance of fixed effects becomes challenging, and Type I error rates across different degrees-of-freedom (DF) methods vary. While previous studies have shown that Kenward-Roger and Satterthwaite DF produce robust Type I error rates for smaller samples, the performance of these methods on small nonnormal distributions has not been explored. Through a simulation study, we show that model complexity, sample size, and number of measurements play a role in the performance of 4 DF methods across distributions of varying skewness and kurtosis. Additionally, we find that Kenward-Roger and Satterthwaite DF methods produce favorable Type I error rates in comparison to DF methods designed for larger sample sizes.

Acknowledgments

I would like to thank the Amherst statistics department for their continuous support during my undergraduate career. I especially want to thank Professor Horton, whose encouragement and constant enthusiasm inspired me to pursue statistics. Additionally, this thesis would not be possible without the guidance of Professor Bailey, who has helped me through this challenging process and allowed me to learn about myself as a researcher. I also want to shout out my SDS partner-in-crime, Kenny. There's no one else I'd rather do group projects with. Thank you to my wonderful friends Maya, Clara, Reylyn, Mina, Libby, and Umaimah for always being by my side and making my college experience so much better. Lastly, I want to thank my family for the sacrifices they've made so I can pursue my passions. They've instilled a love of learning in me that I will have for the rest of my life.

Table of Contents

Abstract	i
Acknowledgments	iii
List of Tables	v
List of Figures	vii
Chapter 1: Introduction	1
1.1 Characteristics of Longitudinal Data	2
1.2 Linear Models for Longitudinal Data	4
1.2.1 Notation and Distribution Assumptions	4
1.2.2 Response Profile Analysis	5
1.2.3 Parametric Time Models	7
1.2.4 Linear Mixed Effects	9
1.3 Choosing the Best Model	12
1.4 Conclusion	13
Chapter 2: Degrees of Freedom Methods and Simulation Study . . .	15
2.1 Estimation	15
2.2 Inference	17
2.2.1 Inference in Small Sample Sizes	18
2.3 Satterthwaite & Kenward Roger	19
2.4 Existing Literature	20

2.5	Goals of This Study:	21
2.6	Simulation Set Up:	21
2.6.1	Generating Data: Sample Size	22
2.6.2	Generating Data: Fixed Effects	22
2.6.3	Generating Data: Random Effects	22
2.6.4	Linear Mixed Effects Model	24
2.7	Evaluating and Results	25
2.7.1	Exponential Distribution	28
2.7.2	Lognormal Distribution	30
2.7.3	Normal Distribution	32
2.7.4	KR vs Satterthwaite	34
2.8	Discussion	35
Chapter 3: Application		39
3.1	Application to Longitudinal Study about Children’s Health	39
3.1.1	Background	39
3.1.2	Exploration	40
3.1.3	Linear Mixed Model	42
3.1.4	Intercept Only Model	43
3.1.5	Intercept and Random Slope	45
3.2	Discussion	47
Conclusion		49
Appendix A: The First Appendix		51
A.1	All Code:	51
Appendix B: P-values from DF Methods		71
Corrections		75

References	77
----------------------	----

List of Tables

3.1	Linear Model Output	42
3.2	Fixed Effects for Random Intercept Model Using KR	43
3.3	Random Effects for Random Intercept Model	43
3.4	Fixed Effects for Random Intercept Model Using Satterthwaite	44
3.5	Fixed Effects for Random Slope Model Using KR	45
3.6	Random Effects for Random Slope Model	45
3.7	Fixed Effects for Random Slope Model Using Satterthwaite	46

List of Figures

2.1	Type I Error Rates by DF method	26
2.2	Type 1 Error Rates by DF method for Exponential Distribution . . .	28
2.3	Type I Error Rates by DF Method for Lognormal Distribution	30
2.4	Type I Error Rates by DF Method for Normal Distribution	32
2.5	Type I Error Rates Produced by KR vs Satterthwaite	34
3.1	Distribution of BMI Across Sample	40
3.2	Changes in BMI by Individual and Alcohol Use	41

Chapter 1 Introduction

In standard undergraduate curricula, there is a strong focus on cross sectional data, and thus no emphasis on how time-sequence data is analyzed. However, a significant portion of data that we encounter in the real world is dependent on time. If we want to track trends and changes over time, such as an effect of a certain drug on the body or growth of a company, longitudinal data and analysis will help us examine those points of interest. For example, the Chinese Longitudinal Healthy Longevity Survey from Duke University assessed physical and mental well-being of Chinese elders for over almost two decades and re-interviewed survivors every few years (Zeng, Vaupel, Xiao, Liu, & Zhang, 2017). This follow up in data collection allowed researchers to investigate the aging process over time and identify risk factors and causes leading up to death.

Not only can we observe change over time in individuals, but we can look at higher-level grouping, such as change in schools, counties, and organizations. It should be emphasized that only longitudinal data can capture changes within a subject or group; cross-sectional data contain responses that are captured at only one occasion that are then compared to other subjects. Ultimately, it cannot provide information about changes over time.

One key aspect of longitudinal data is that there needs to be repeated measurements of the same individuals across multiple periods of time. If there aren't repeated observations, then it is not possible to make any comparisons between two or more

time points. Having repeated measurements of the same individual allows for removal of potential confounding effects, such as gender or socioeconomic status, from the analysis. Since we assume that these confounding variables are fixed effects that do not vary from measurement to measurement, all changes from an individual cannot be attributed to these effects.

The measure that captures the observed changes within an individual is referred to as a response trajectory. There are different ways of comparing response trajectories. For example, it is possible to compare the post-treatment vs baseline changes across multiple treatment groups, or it is also possible to compare the rate of change. The method chosen depends on the specific question of the study.

Apart from comparing just the response trajectories, it may also be of interest to compare individual differences in the relationship between covariates and the response trajectory. This can be captured using various statistical models. The choice of model depends on several characteristics of the data.

1.1 Characteristics of Longitudinal Data

While the only requirement of longitudinal data is that there is more than one observation for a given individual, there are other characteristics that affect the model chosen. Data can be unbalanced or balanced: *balanced* data refers to when all individuals have the same number of repeated measurements taken at the same occasions. For example, a study measuring the weights of children over time may be unbalanced if some children are measured at 5 and 10 years old, while others are measured at 5, 7 and 15 years old, since measurements are not taken at the same time points. If children drop out of the study as they get older, this will result in missing data points. If there are missing data, it is automatically considered as unbalanced.

Another unique characteristic of longitudinal data is that repeated measurements within each individual are typically positively correlated, while measurements between individuals are independent (Fitzmaurice, Laird, & Ware, 2011). This feature violates conditions of other common statistical methods such as linear regression, where measurements are assumed to be independent. This positive correlation allows for more accurate estimates of the model coefficients and response trajectories since there is reduced uncertainty knowing that a previous measurement can help predict the next one.

We can capture the associations among repeated measures within each individual by constructing a covariance matrix. The diagonal elements of this matrix represent the variance of each of the measurements, which may not be constant over time for longitudinal data. The off-diagonal elements of the matrix are non-zero to account for the lack of independence between measurements, and are usually not constant because correlations between measurements tend to decrease over time. While these values are rarely 0, they are also rarely 1 (Fitzmaurice et al., 2011). There are different covariance pattern structures that are imposed that account for these features.

These features of the covariance of longitudinal data can be explained when we separate the total variation into three distinct parts: 1) between-individual variation, 2) within-individual variation, and 3) measurement error.

Between-individual variation helps explain why measurements from the same individual are more likely to be positively correlated than measurements between different individuals. Within-individual variation helps explain why correlations between repeated measurements decrease with increasing time differences, and measurement error along with individual variations explains why correlations are never one. These three types of variation may contribute to total variation in unequal amounts, but may not need to be differentiated depending on the type of longitudinal analysis

desired.

1.2 Linear Models for Longitudinal Data

1.2.1 Notation and Distribution Assumptions

Throughout the rest of the text, we will use a standard set of notation. Y_{ij} represents the response variable for the i^{th} individual at the j^{th} measurement. When we have repeated n_i measurements for an individual, we can construct a vector,

$$Y_i = \begin{pmatrix} Y_{i1} \\ Y_{i2} \\ \vdots \\ Y_{in_i} \end{pmatrix}, i = 1, \dots, N.$$

For each response Y_{ij} , there is a $p \times 1$ vector X_{ij} of covariates whose first entry is 1. Covariates in X_{ij} can be fixed or change over time.

As mentioned previously, there are multiple ways to model longitudinal data. When the response variable is continuous, we can consider a model that relates the response and the covariates in a linear way. In a linear model all components can be represented using vectors and matrices. The most general form of the linear model can be represented in matrix form as:

$$Y_i = X_i\beta + e_i,$$

where X_i is a matrix representing the grouped collection of X_{ij} vectors, with each row representing a unique measurement, and p columns for each covariate that is associated with Y_i . $\beta = [\beta_1, \dots, \beta_{p-1}]'$ is a vector of regression coefficients that quantifies the

relationship between the response and each covariate. e_i is a $n_i \times 1$ vector of random errors for each measurement.

The linear model can be divided into a systematic component, $X_i\beta$, and a random component e_i . These two components contribute to the distribution assumptions of Y_i . Y_i is assumed to have a conditional multivariate normal distribution with mean response vector

$$E(Y_i|X_i) = X_i\beta,$$

which is the systematic component, and the covariance matrix

$$\Sigma_i = \text{Cov}(Y_i|X_i),$$

which captures the random variability of Y_i , the random component, and its role in shaping the overall distribution. In addition, this distribution is considered conditional because of its dependence on the covariates X_i .

In the following sections, we will discuss three specific methods of linear models: 1) response profile analysis, 2) parametric time model, and 3) linear mixed effects model.

1.2.2 Response Profile Analysis

In response profile analysis, we allow for arbitrary patterns in the mean response over time. A sequence of means over time is known as the mean response profile. The main goal of this analysis is to identify differences in pattern of change in mean response profile among two or more groups. This method requires that the data be balanced.

There are three effects of interest when analyzing response profiles in longitudinal analysis: 1. group \times time interaction effect (are the mean response profiles different in groups over time?) 2. time effect (assuming mean response profiles are parallel

between groups, are the means changing over time?) 3. group effect (do the mean response profiles differ?)

Analyzing response profiles can be modeled using

$$E(Y_i|X_i) = X_i\beta.$$

In an example where two groups with three measurements each are compared, $\mu(1) = \beta_1, \beta_2, \beta_3$ represent regression coefficients for group 1, and $\mu(2) = \beta_4, \beta_5, \beta_6$ represent regression coefficients for group 2. The group \times time interaction effect can be expressed as a null hypothesis in the form

$$H_0 : \beta_5 = \beta_6 = 0.$$

An unstructured covariance model is typically assumed for response profile analysis. “Unstructured” means that there is no explicit structure or pattern imposed on the covariance for the repeated measures. This is represented as

$$\Sigma_i = \text{Cov}(Y_i) = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \dots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \dots & \sigma_n^2 \end{pmatrix}.$$

For n repeated measures, there are n variances and $n \times (n - 1)/2$ covariances to be estimated. In a study where there are 10 repeated measurements, there 55 total covariance parameters to be estimated, which can become computationally intensive.

Overall, response profile analysis is a straightforward method in investigating differences between groups for longitudinal data. Since both the covariance and mean

responses have no imposed structure, the analysis is more robust and immune to inaccurate results due to model misspecification. However, there are drawbacks as well. Response profile analysis does not consider time-order of the measurements and does not distinguish between between-individual variation and within-individual variation. In addition, it can only provide a broad analysis of whether there are differences across groups and time, but does not provide the amount of detail needed to answer certain research questions, such as how exactly measurements taken towards the end of the study compare to measurements taken at the beginning. In this method, time is treated as a categorical covariate rather than a continuous one. Another method that addresses the issue of examining time order of the data is parametric time models.

1.2.3 Parametric Time Models

Parametric time models are able to capture time order of the data by fitting linear or quadratic curves to capture an increasing or decreasing pattern over time. Time is treated as a continuous covariate rather than a categorical one. In addition, unlike response profile analysis, parametric time models are able to handle unbalanced and missing data. Rather than fitting a complex and perfect model onto the observed mean response profile, parametric time models fit simple curves that produce covariate effects of greater power. The same question, such as examining group \times time effect, requires n parameters in response profile analysis, but only requires one parameter in parametric time models.

Additionally, while in the mean response profile analysis an unstructured covariance pattern is assumed, here there is flexibility in choice of the covariance model; there are several options such as Toeplitz or compound symmetric that impose various structures on the model. For example, a Toeplitz model:

$$\Sigma_i = \sigma^2 = \begin{pmatrix} 1 & \rho_1 & \rho_2 & \dots & \rho_{n-1} \\ \rho_1 & 1 & \rho_1 & \dots & \rho_{n-2} \\ \rho_2 & \rho_1 & 1 & \dots & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{n-1} & \rho_{n-2} & \rho_{n-3} & \dots & 1 \end{pmatrix}$$

structures the covariance matrix such that any pair of responses that are equally separated in time have the same correlation. A compound symmetry model:

$$\Sigma_i = \sigma^2 \begin{pmatrix} 1 & \rho & \rho & \dots & \rho \\ \rho & 1 & \rho & \dots & \rho \\ \rho & \rho & 1 & \dots & \rho \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \rho & \dots & 1 \end{pmatrix},$$

assumes constant variance and same correlation across all measurements of Y_i . It is possible to choose an unstructured covariance model as well, but can be computationally intense if there are a large number of measurements.

We can use parametric time models in two ways: through polynomial trends and linear splines. More information about linear splines can be found in Fitzmaurice et al. (2011). Using polynomial trends such as linear or quadratic, we can model longitudinal data as a function of time. Linear trends are the most common and interpretable ways to model change in mean over time. In an example comparing a treatment group to a control group, we can fit a linear trend using the following equation:

$$E(Y_{ij}) = \beta_1 + \beta_2 Time_{ij} + \beta_3 Group_i + \beta_4 Time_{ij} \times Group_i.$$

If $\beta_4 = 0$, then any difference in response between groups is constant over time.

For quadratic trends, the changes in mean are no longer constant since the rate of change depends on the time. Thus, we fit an additional parameter to express the rate of change. Using the previous example of treatment vs. control group, we have the model:

$$E(Y_{ij}) = \beta_1 + \beta_2 Time_{ij} + \beta_3 Time_{ij}^2 + \beta_4 Group_i + \beta_5 Time_{ij} \times Group_i + \beta_6 Time_{ij}^2 \times Group_i.$$

As we can see from the models above, the inclusion of an additional parameter $Time_{ij}^2$ changes the mean response rate. One problem that may arise from using quadratic trends is that there is collinearity between $Time_{ij}$ and $Time_{ij}^2$, which can affect the estimation of β . To account for this, we can center the $Time_{ij}$ variable around the mean time value for all individuals, instead of centering it around zero as done in normal analysis. For example if we have a set of times $Time = 0, 1, 2, \dots, 10$, then the mean time value is five. Thus time zero would be recentered as -5 . The interpretation of the intercept changes to represent the mean response at that recentered mean time value.

1.2.4 Linear Mixed Effects

In both response profile analysis and parametric time models, the regression parameters are considered to be universal for each population group. However, in instances where we want to account for heterogeneity within a population, we can use a linear mixed effects model and consider a subset of the regression parameters to be random. This model distinguishes between fixed effects, which are population characteristics shared by all individuals, and subject specific effects, also known as random effects, which pertain to each individual. These subject specific effects mean that parame-

ters are random, which induces a structure onto the covariance model. In addition, distinguishing between fixed and random effects allows for differentiation between within-subject and between-subject variation.

One example of the linear mixed effects model is the random intercept model, which is the simplest version of the linear mixed effects model:

$$Y_{ij} = X'_{ij}\beta + b_i + \epsilon_{ij}$$

This model is very similar to the general linear model with a few additions. b_i is the random subject effect and ϵ is the measurement error. Both effects are random, with mean 0 and $\text{Var}(b_i) = \sigma_b^2, \text{Var}(\epsilon_{ij}) = \sigma^2$.

$X'_{ij}\beta$ is the population mean, and b_i represents the differing subject effect that is unique to each individual. b_i is interpreted as how the subject's intercept deviates from the population mean intercept while accounting for covariates.

As mentioned previously, the random effects are responsible for inducing a structure on the covariance model. This structure is not to be confused with the covariance structures that can be chosen when using parametric time models. For a given individual, it can be shown that variance of each response is:

$$\text{Var}(Y_{ij}) = \sigma_b^2 + \sigma^2$$

and the covariance between two measurements Y_{ij} and Y_{ik} is equal to σ_b^2 . The resulting covariance matrix

$$\text{Cov}(Y_i) = \begin{pmatrix} \sigma_b^2 + \sigma^2 & \sigma_b^2 & \sigma_b^2 & \dots & \sigma_b^2 \\ \sigma^2 & \sigma_b^2 + \sigma^2 & \sigma_b^2 & \dots & \sigma_b^2 \\ \sigma_b^2 & \sigma_b^2 & \sigma_b^2 + \sigma^2 & \dots & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_b^2 & \sigma_b^2 & \sigma_b^2 + \sigma^2 & \dots & \dots \end{pmatrix}$$

implies correlation between measurements, and also highlights the role played by the random effects in determining the covariance.

Extending beyond the random intercept model, multiple random effects can be incorporated.

A linear mixed effects model can be expressed as

$$Y_i = X_i\beta + Z_ib_i + \epsilon_i,$$

where: β is a $p \times 1$ vector of fixed effects b_i is a $q \times 1$ vector of random effects X_i is a $n \times p$ matrix of covariates Z_i is a $n \times q$ matrix of covariates

The subset of regression covariates that vary randomly are found in Z_i . We assume that b_i comes from a multivariate normal distribution with mean 0 and covariance matrix G . We also assume that ϵ_i are independent of b_i , and come from multivariate normal distribution with mean 0 and covariance matrix R_i .

The covariance of Y_i can be modeled by

$$\text{Cov}(Z_ib_i) + \text{Cov}(\epsilon_i) = Z_iGZ_i' + R_i.$$

This model, which outlines a distinction between G and R_i , allows for separate analysis of between-subject and within-subject variation. Unlike other covariance models, in linear mixed effects models the covariance is a function of the times of measurement. This allows for unbalanced data to be used for the model since each individual can have their unique set of measurement times. Lastly, the model allows for variance and covariance to change as a function of time. To illustrate, consider the following model:

In an example where individuals can vary both in their baseline response and their rate of change, we have:

$$Y_i = X_i\beta + Z_ib_i + \epsilon_i,$$

where both X_i and

$$Z_i = \begin{pmatrix} 1 & t_{i1} \\ 1 & t_{i2} \\ \dots & \dots \\ 1 & t_{in} \end{pmatrix}$$

. For the i^{th} subject at the j^{th} measurement, the equation is as follows:

$$Y_{ij} = \beta_1 + \beta_2 t_{ij} + b_{1i} + b_{2i} t_{ij} + \epsilon_{ij}.$$

If $\text{Var}(b_{1i}) = g_{11}$, $\text{Var}(b_{2i}) = g_{22}$, and $\text{Cov}(b_{1i}, b_{2i}) = g_{12}$ where these three components represent the G covariance for b_i , then it can be shown that $\text{Cov}(Y_{ij}, Y_{ik}) = g_{11} + (t_{ij} + t_{ik})g_{12} + t_{ij}t_{ik}g_{22}$.

Here in the covariance matrix we can see the dependence of the covariance on time. In this example there are four covariance parameters that arise from the two random effects of intercept and time. The number of covariance parameters is represented by $q \times (q+1)/2 + 1$, where q is the number of random effects. To choose the most optimal model for covariance, we compare two nested models, one with $q + 1$ random effects and one with q random effects. We use the likelihood ratio test to make a decision for which model to use.

1.3 Choosing the Best Model

After presenting three methods of evaluating longitudinal data, the natural question arises of how to choose the most appropriate model. While there is no definite correct

answer, there are several factors to consider. If data are unbalanced, response profile analysis should not be considered; rather, parametric time model or linear mixed effect model would be more optimal. If time order is important to the analysis, then only parametric time model and linear mixed effect model should be used. If there is a need to distinguish between the two types of variation that can occur, then only linear mixed effect models are appropriate. The model should ultimately be chosen based on the characteristics and constraints of the data, as well as the specificity of the research question at hand.

1.4 Conclusion

Longitudinal analysis is a valuable method to analyze changes over time. It is important to understand the unique characteristics that come with this analysis and to choose the best model that can capture the salient patterns that arise from the data.

In subsequent chapters we will dive more deeply into how inference in longitudinal analysis is affected when sample sizes are not efficient through both simulation and application.

Chapter 2 Degrees of Freedom Methods and Simulation Study

In chapter 1, we outlined the basics of analyzing longitudinal data and introduced linear mixed models. Next, we will examine inference of linear mixed models, and how methods such as Kenward-Roger (KR) and Satterthwaite can be used in situations where standard procedures for inference may produce questionable results.

2.1 Estimation

Regression coefficient values β and the covariance matrix Σ_i can be estimated using maximum likelihood estimation, which identifies values of β and Σ_i that best explain the observed data. These values are estimates that are denoted by $\hat{\beta}$ and $\hat{\Sigma}_i$. When observations are independent of one another, maximizing the likelihood function for β is equivalent to finding a value of $\hat{\beta}$ that minimizes the sum of the squares of the residuals. However, since there are repeated measurements of each individual that are not independent of one another we use the generalized least squares (GLS) estimator as shown by Fitzmaurice et al. (2011):

$$\hat{\beta} = \sum_{i=1}^N (X_i' \Sigma_i^{-1} X_i)^{-1} \sum_{i=1}^N (X_i' \Sigma_i^{-1} y_i).$$

The sampling distribution of $\hat{\beta}$ has mean β and covariance:

$$\widehat{\text{Cov}}(\beta) = \sum_{i=1}^N (X_i' \Sigma_i^{-1} X_i)^{-1}.$$

The GLS estimator assumes that Σ_i is known. However, since this isn't usually the case, we can substitute Σ_i with a maximum likelihood estimate $\hat{\Sigma}_i$. It can be shown that the properties of $\hat{\beta}$ still hold using an estimate of the covariance. These properties include $\hat{\beta}$ being a consistent estimator of β and having sampling distribution that is multivariate normal with mean β and covariance $\text{Cov}(\hat{\beta}) = \sum_{i=1}^N (X_i' \Sigma_i^{-1} X_i)^{-1}$.

While the maximum likelihood estimate of Σ_i is adequate, a modified method known as restricted maximum likelihood (REML) estimation is suggested to reduce bias in finite samples (Fitzmaurice et al., 2011). The bias originates from the fact that β itself is also estimated from data, but is not accounted for when estimating covariance. In REML estimate of Σ_i , β is removed from the likelihood function. This REML estimation of Σ_i can be used in the GLS estimator for $\hat{\beta}$ mentioned above, and is recommended in place of the ML estimator.

One additional analysis that is possible with linear mixed effects models is predicting subject-specific responses. Given that b_i is a random variable, we can predict it using:

$$E(b_i|Y_i) = GZ_i(\Sigma)_i^{-1}(Y_i - X_i\hat{\beta}).$$

Because the covariance of Y_i is unknown, we can estimate both G and Σ_i using REML, creating \hat{b}_i , also known as the empirical BLUP (Fitzmaurice et al. (2011)). Thus, the equation for predicting the response profile is:

$$\hat{Y}_i = X_i\hat{\beta} + Z_i\hat{b}_i.$$

This equation to estimate the mean response profile can be extended to incorporate R_i , which represents within-subject variability. From this extension, we see that the equation and the empirical BLUP account for the weighting of both the within-subject variability and between-subject variability. If there is more within-subject variability, then more weight is assigned to $X_i\hat{\beta}$, the population mean response profile, in comparison to the subject's individual responses, and vice versa.

2.2 Inference

In statistical inference, the goal is to make conclusions about the underlying characteristics of a set of data and establish a relationship between certain variables. Hypothesis testing is one of the primary examples of inference, and is carried out in order to assess the true value of a population parameter. In linear models, the significance of a slope parameter, β_k , is often assessed, where the null hypothesis, H_0 is $\beta_k = 0$, and the alternative hypothesis H_a is $\beta_k \neq 0$. A test of the null hypothesis involves using a Wald statistic in the form

$$Z = \frac{\hat{\beta}_k}{\sqrt{\widehat{Var}(\hat{\beta}_k)}},$$

which is then compared to the normal distribution, and a subsequent p-value is obtained.

Building on foundations of a general linear hypothesis test, given a matrix L of size $r \times p$, where r represents the number of estimable functions of β , the Wald statistic

$$(L\hat{\beta} - L\beta)'[L(\widehat{Cov}(\hat{\beta}))L']^{-1}(L\hat{\beta} - L\beta)$$

is approximately $\chi^2(r)$ (Rencher & Schaalje, 2007). For a null hypothesis $H_0 : L\beta = 0$,

the test statistic is

$$W^2 = (L\hat{\beta})'[L(\widehat{Cov}(\hat{\beta}))L']^{-1}(L\hat{\beta}).$$

Likelihood ratio tests are another method to make inferences about β . While there are benefits to using the likelihood ratio test, the rest of this study will focus on method of using the Wald statistic.

2.2.1 Inference in Small Sample Sizes

One crucial assumption when conducting inference using the maximum likelihood estimate for β is that the sample size is sufficient enough where it does not affect the estimate for Σ_i . However, what happens when the sample size is too small? This causes $\hat{\Sigma}_i$ to underestimate the true variance, which in turn causes $\widehat{Cov}(\hat{\beta})$ to be too small since it relies on the covariance estimator. If $\widehat{Cov}(\hat{\beta})$ is too small, the denominator of the Wald test statistic is too small, so the test statistic is too big, leading to increased Type I error. One can see that the bias of the covariance estimator weakens the entire foundation of estimation and inference.

There are limited cases where exact small-sample inference can be performed, but in most scenarios, an approximate small-sample method must be used, in which the statistic

$$F = \frac{(L\hat{\beta})'[L(\widehat{Cov}(\hat{\beta}))^{-1}L']^{-1}(L\hat{\beta})}{g}$$

follows a F distribution with numerator degrees of freedom g , and unknown denominator degrees of freedom, DDF. There are several ways to approximate the DDF.

Both Satterthwaite and KR are proposed methods of approximations to the DDF when conducting tests in order to account for the uncertainty of the covariance estimator. The KR method goes one step forward to also adjust the test statistic itself.

2.3 Satterthwaite & Kenward Roger

Hrongs-Tai Fai & Cornelius (1996) propose using Satterthwaite approximation of DF for the aforementioned F statistic when the hypothesis of interest is $H_0 : L\beta = 0$, where L is an estimable contrast matrix. Details about how the DF is approximated can be found in the original paper. In a simpler case where $H_0 : l'\beta = 0$ and $r = 1$, the Wald statistic is

$$t = \frac{l'\hat{\beta}}{\sqrt{l'\widehat{\text{Cov}}(\hat{\beta})l}},$$

which follows a t distribution with unknown DF ν . Satterthwaite (1946) assumed the quantity

$$\frac{\nu(l'\widehat{\text{Cov}}(\hat{\beta})l)}{l'\widehat{\text{Cov}}(\hat{\beta})l},$$

and used a method-of-moments approach to obtain $\hat{\nu} = \frac{2(l'\widehat{\text{Cov}}(\hat{\beta})l)^2}{\widehat{\text{Var}}(l'\widehat{\text{Cov}}(\hat{\beta})l)}$, where the denominator is determined by a univariate delta method (Kuznetsova, Brockhoff, & Christensen, 2017).

KR uses the same approach, but adjusts for the underestimation of $\text{Cov}(\hat{\beta})$ in small sample sizes by including an adjusted estimator for the covariance matrix of $\hat{\beta}$ that inflates the covariance estimate in the numerator of the Wald statistic (Kuznetsova et al., 2017).

When $L = 1$ the KR and Satterthwaite approximation will produce the same denominator degrees of freedom. However, since the statistic used for the two methods are not the same, the results for inference will not be the same, as KR adjusts the estimate for the covariance matrix of $\hat{\beta}$.

2.4 Existing Literature

Both KR and Satterthwaite methods are frequently used and compared, and its performance is highly dependent on the structure of the data. A majority of studies focusing on DF method comparison in mixed models use split-plot design, as small sample sizes are more common in agricultural and biological fields. Schaalje, McBride, & Fellingham (2002) found that in comparison to other degrees of freedom-adjusting methods like Satterthwaite, KR was the most suitable for small sample data. Using factors such as imbalance, covariance structure, and sample size, they demonstrated that the KR method produced simulated Type I error rates closest to target values. However, their focus was primarily on complexity of covariance structure, and they found that more complicated structures, such as ante-dependence, produced inflated error rates when coupled with small sample size. Arnau, Bono, & Vallejo (2009) found that KR produces more robust results compared to Satterthwaite and Between-Within approaches, especially in cases where larger sample size was paired with covariance matrices with larger values.

These studies are conducted with data drawn from normal distributions. However, real-world data used in fields such as psychometrics have distributions that are nonnormal. In their paper, Arnau, Bono, Blanca, & Bendayan (2012) extend their evaluation of KR for split-plot data that follow a log-normal or exponential distribution, and for when the kurtosis and skewness values are manipulated. They found that, compared to normal distribution, KR is less robust for log-normal distributions, but that there is no significant difference in performance between exponential and normal distributions. In addition, they suggest that skewness has a bigger effect on robustness of KR compared to kurtosis.

Existing research evaluating the performance of methods that reduce Type I error

rate in small samples are thorough, however, the differences in simulation setup and structure of data used make generalizations difficult. Although the KR method has been shown as a viable option for analysis of small samples in many occasions, it should continue to be evaluated against other methods. To date, there is no literature on the performance of Satterthwaite for nonnormal longitudinal data design. Given the prevalence of nonnormal and small data samples, it is important to continue exploring methods that ensure robust results.

2.5 Goals of This Study:

In this study, we aim to expand on previous work evaluating how methods for estimating fixed effects perform under different nonnormal distributions, sample sizes, number of measurements, and complexity. The aforementioned studies often use a split-plot design and impose a covariance structure, but the goal of this study will be to compare performance of KR and Satterthwaite methods for repeated measures longitudinal data fitted with a linear mixed effects model, and an unstructured covariance structure imposed by the random effects.

2.6 Simulation Set Up:

The overarching goal is to compare performance of DF methods across various conditions, with a specific goal of looking at scenarios with smaller sample sizes and number of measurements in order to see variability in Type I error rates. There is flexibility in how the parameters are chosen; in order to narrow the scope, we aim to design conditions that are relatively similar to an application data set that will be further explored in Chapter 3. In the following sections, the range of parameters used will be similar to the characteristics of a longitudinal study about Children’s Health.

2.6.1 Generating Data: Sample Size

In this study, we consider a linear mixed effects model with time as a continuous variable and treatment as a factor. The range of possible values that time takes on depends on how many number of measurements per individual, which can be 4 or 8. The treatment covariate takes on values of 0 or 1, and each assigned to half of the sample. The number of individuals take on possible values of 10, 18, and 26. These were chosen to reflect possible samples that would not hold under the common assumption that the sample size must be at least 30 for it to be considered sufficient enough for the Central Limit Theorem to hold.

2.6.2 Generating Data: Fixed Effects

We have three fixed effects: the intercept value, time and treatment. The intercept, an arbitrary value, is set at $\beta_0 = 3.1$. The coefficients for time and treatment have a value of $\beta_1 = \beta_2 = 0$ for simplicity and so we can evaluate Type 1 error rates.

2.6.3 Generating Data: Random Effects

We generate our intercept and slope random effects values from nonnormal distributions, which are either exponential or lognormal. Previous research shows that many data used in social and health sciences follow nonnormal distributions (Limpert, Stahel, & Abbt, 2001). More specifically many follow lognormal distributions, such as age of onset of Alzheimer’s disease (Horner, 1987), or exponential distribution to model survival data. In order to cover a wide range of exponential and lognormal distributions, parameters were chosen to model distinct distributions. For exponential distributions, $b_i \sim \text{Exp}(\lambda)$, where $\lambda = .2$ or $.9$. Lognormal distributions were $b_i \sim \text{LogNormal}(0, .25)$ and $b_i \sim \text{LogNormal}(1, .5)$.

Using the *SimMultiCorrData* package, values are generated through Fleishman’s method for simulating nonnormal data by matching moments using mean, variance, skew, and kurtosis and then transforming normally distributed values (Fialkowski, 2018).

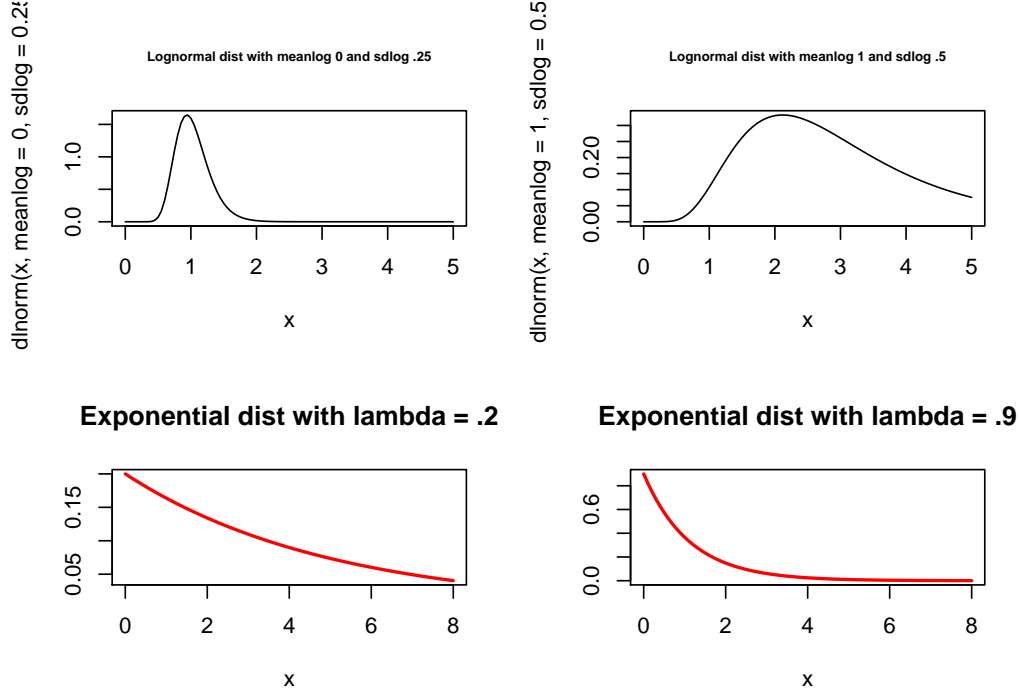
When simulation data from nonnormal distributions, we use the standard deviation produced by Fleishman’s method as the parameter. In order to align the simulation data from the Children’s Health study, we use the same ratio of variances between the residual variance and variance of the random effects of time and intercept derived from modeling linear mixed models on that data in our simulation.

In the intercept only model, only one non-normal continuous variable is generated for the random effect, so the function `nonnormvar1()` is used. In order to generate measurement error, we simulate values from a $t_{ij} \sim N(0, \sqrt{4}\sigma)$, where σ is the standard deviation from each distribution we want explore.

In the case of the linear model that has both random effects for intercept and slope, we want to generate random effects values that are correlated. Using `SimMultiCorrData::rcorrvar()`, we use a similar process for generating one non-normal continuous variable, but extend it to generating variables from multivariate normal distribution that take in to account a specified correlation matrix, and are then transformed to be nonnormal. Also, in order to maintain the same ratio of variances to the application data, the variance of the distribution that the slope random effects are generated is equal to $\frac{1}{8}\sigma^2$, where σ^2 refers to the variance of the distribution that the random effects for the intercept are generated from. On a similar note, we use a correlation value of -.38 to generate the random effects for the intercept and slope, which is based off the correlation observed between the random effects of time and intercept when fitting a random slope model in Chapter 3. For random slope models, the measurement error is simulated for a distribution

$$e_{ij} \sim N(0, \sqrt{.168SD}).$$

Distributions used are depicted below.



2.6.4 Linear Mixed Effects Model

In a linear mixed effects model, the amount of random effects that will be modeled depends on the research question at hand. Here, we will examine both a random intercepts model, where only the intercept of the model is assumed to have a random effects structure, as well as a random intercept and slope model, where in addition to intercept, the covariate time will also have a random effects structure. The random intercept model can be represented as:

$$Y_{ij} = \beta_0 + \beta_1 \text{Trt}_i + \beta_2 \text{Time}_{ij} + b_{0j} + e_{ij},$$

and the random slope model can be represented as:

$$Y_{ij} = \beta_0 + \beta_1 Trt_i + \beta_2 Time_{ij} + b_{0j} + b_{1i} Time_{ij} + e_{ij}.$$

We use the *lme4* package to fit the linear mixed effects model and *lmerTest* to evaluate random and fixed effects. To evaluate significance of the treatment variable, we compare the performance and resulting p-values from 4 different Wald-type tests: KR, Satterthwaite, standard DF method, and *t-as-z*. *t-as-z* and standard DF formula are not adjustments to account for smaller sample sizes, and are used as comparison to Satterthwaite and KR, since they are expected to be anti-conservative. Because the value of the treatment in our model is fixed at 0 in order to identify Type I error, we expect to see that the p-value to not be significant ($p > .05$) in an ideal scenario.

2.7 Evaluating and Results

After performing 400 replications of each condition at a significance level of .05, we evaluate robustness using Bradley’s criterion, which considers a test to robust if the empirical error rate is between .025 and .075 (Bradley, 1978). In the following section, we will compare Type I error rates produced from KR and Satterthwaite methods as well as *t-as-z* and using the standard DF formula, further stratified by distribution and other manipulated parameters.

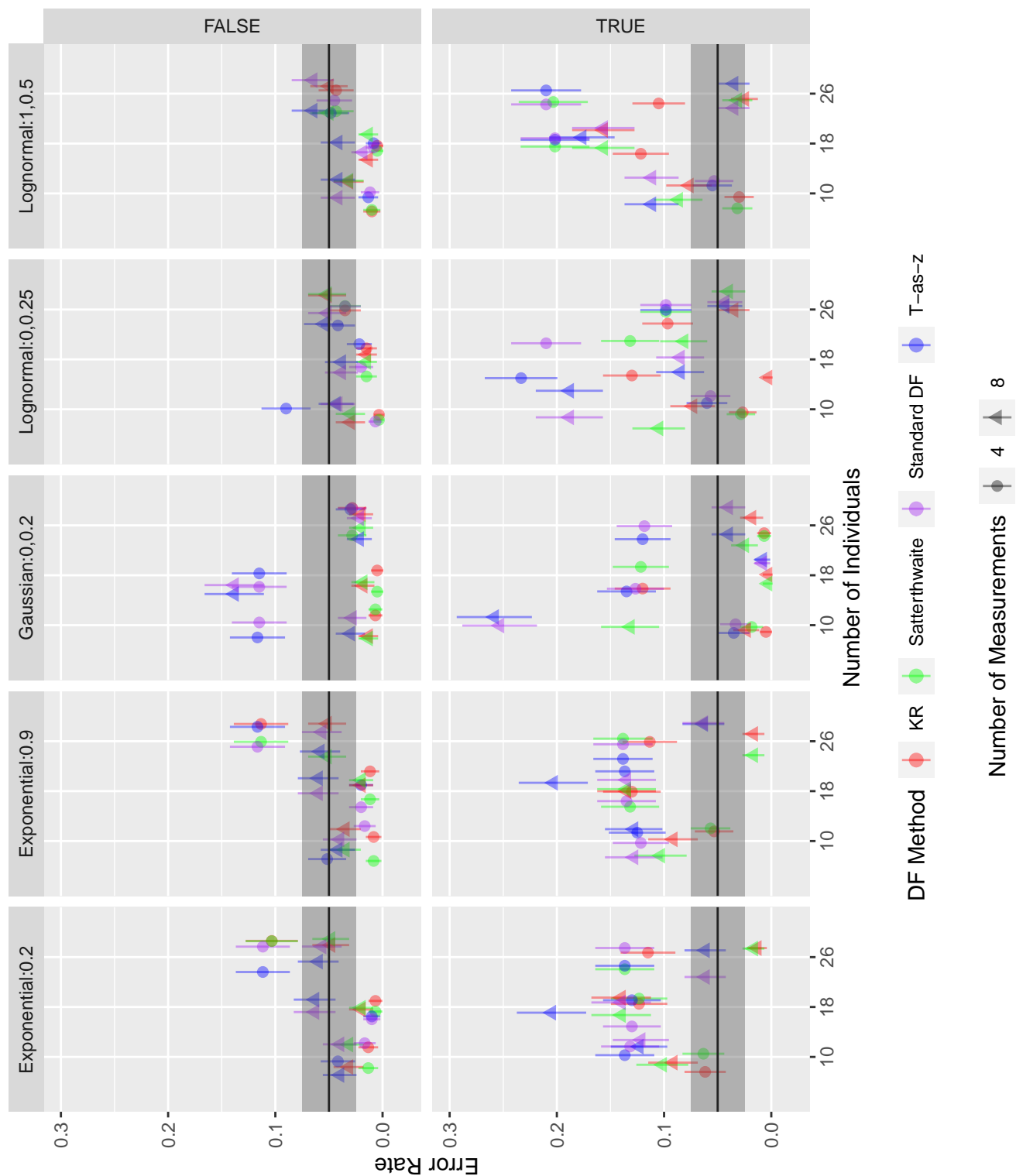


Figure 2.1: Type I Error Rates by DF method

Figure 2.1 displays error rates from all 4 degrees of freedom methods by distribution, parameters, complexity of random model, number of measurements, and number of samples. The shaded region indicates error rates that are considered robust by Bradley (1978)’s criterion. It is evident that there are varying patterns of performance by distribution. The common conception that larger sample sizes or large number of measurements can improve robustness is not necessarily evident across all distributions, for example in the case of the exponential distribution. One trend that appears to be evident across all three distributions is that when the degrees of freedom methods are applied to a random intercept model, a more structurally simple model, they yield more robust error rates in comparison to an application to the random slopes model.

In addition, when looking at performance of the 4 methods overall, we can see that the t -as- z and standard DF approach produce significantly more anti-conservative results, regardless of the values of other parameters. These trends align closely with a previous study by Luke (2017) examining only normal distributions.

In order to make more specific observations and identify trends, we will examine performance within each of the three distributions by sample size and number of measurements.

2.7.1 Exponential Distribution

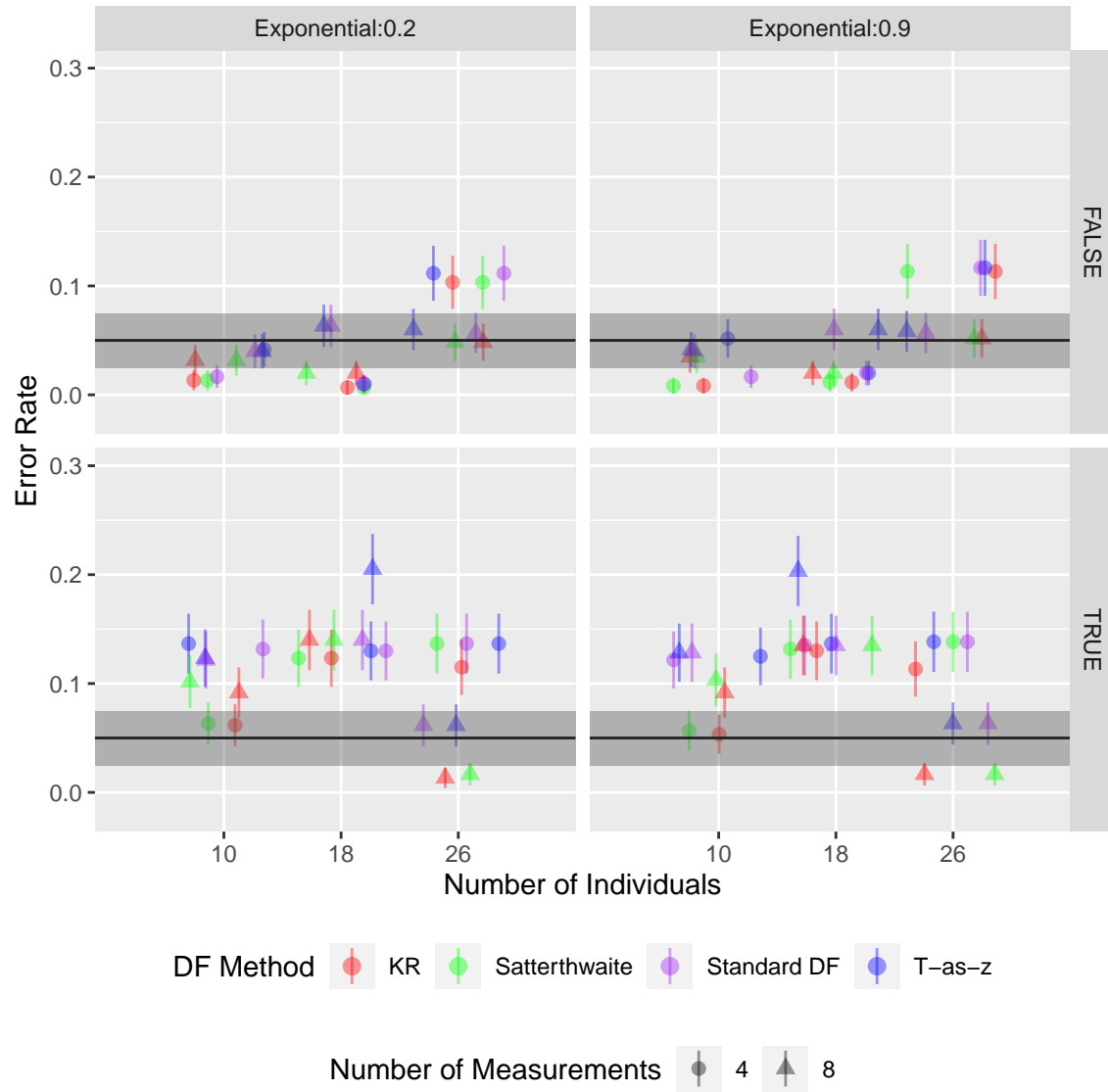


Figure 2.2: Type 1 Error Rates by DF method for Exponential Distribution

Our simulation results explore two exponential distributions, one with $\lambda = .9$ and $\lambda = .2$. Looking at figure 2.2, at $\lambda = .2$, we can see that in random slope models, comparing sample size of 10 to 26 marginally improves the DF methods, but only

on those that are applied to conditions with more repeated measurements. The relationship between sample size and robustness is not linear, as increasing the sample size from 10 to 18 does not help DF methods achieve less anti-conservative error rates. On the other hand, in random intercept models, increasing the sample size did not improve the performance of DF methods, and in some cases were associated with more anti-conservative Type I error rates; in the case of sample size 26 and 4 measurements, Type I error rates performed significantly worse in comparison to smaller sample sizes, holding other conditions constant.

At $\lambda = .9$, we see virtually the same trends in terms of the effect of sample size, complexity of model, and number of measurements on the performance of the DF methods. Overall, across both distributions, increasing the number of repeated measures impacted the DF methods' performance and increased robustness, while the effect of sample size was hard to pinpoint. Additionally, Kenward-Roger and Satterthwaite methods tended to produce more conservative error rates.

Despite having different parameter values, the application of DF methods to these two exponential distributions produce similar different trends in error rates. Considering that the two distributions have the same skewness and kurtosis values, we hypothesize whether this similarity in values contributes to the performance of the DF methods. Next, we will examine the performance of DF methods in lognormal distributions.

2.7.2 Lognormal Distribution

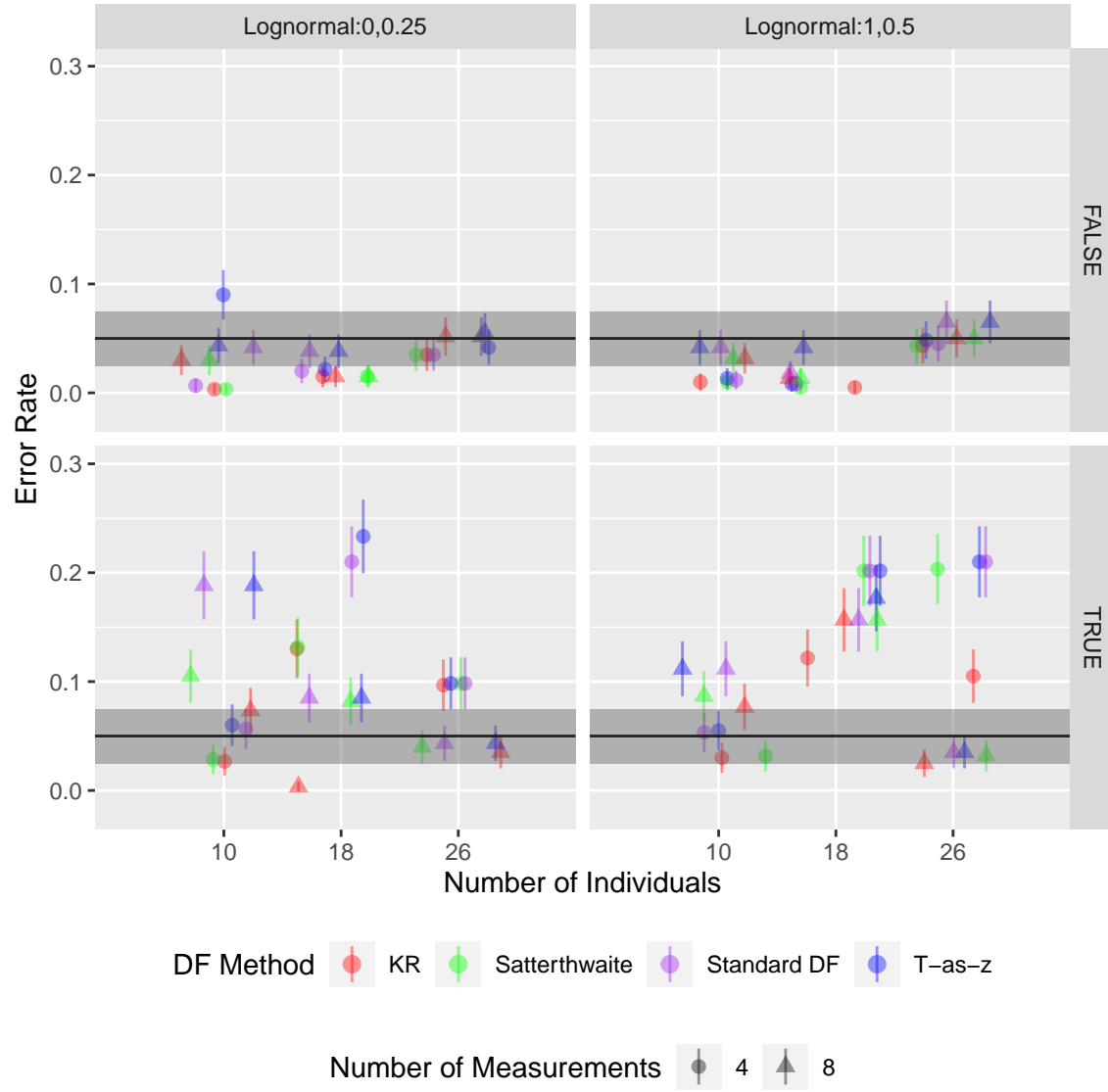


Figure 2.3: Type I Error Rates by DF Method for Lognormal Distribution

Figure 2.3 highlights Type I error rates produced by DF method for lognormal data. As seen in the exponential distribution, across the lognormal distributions, DF methods applied to random intercept models had consistently more robust error rates in

comparison to random slope.

Our first lognormal distribution with parameters $(0, .25)$ has lower values of kurtosis and skewness. In the random slope model, increasing sample size from 10 to 18 only improves performance of the t -as- z and standard DF method when individuals have 8 measurements each, and in all other cases leads to less robustness. At sample size 26, DF methods are significantly more robust when applied to 8 measurements per individual rather than 4. In the random intercept model, we see a similar lack of consistent pattern of Type I error rates when sample size increases from 10 to 18, but at sample size 26 all methods regardless of number of measurements are robust.

On the other hand, with higher levels of skewness and kurtosis with a lognormal distribution with parameters $(1, .5)$, the effect of number of measurements and sample size are slightly different. Increasing the sample size from 10 to 18 has either no effect or an adverse effect on DF methods; in random intercept models the DF methods generally produce Type I error rates that are too conservative, and the opposite is seen in random slope models. Similar to the other lognormal distribution, DF methods applied to random intercept models of size 26 are all robust. The same difference between performance of DF methods of 4 and 8 measurements at size 26 in random slope models are observed, but that difference is widened. All DF methods at 4 measurements aside from the KR method have Type I error rates that almost 4 times the error rates at 8 measurements.

Based on comparisons between the two distributions, our results suggest that there are only slight differences that arise, suggesting that to a certain extent, distributions with greater skew and kurtosis may not significantly impact the performance of DF methods. In both distributions, DF methods applied to models with size 26 and 8 measurements were the most robust. Increasing sample size from 10 to 18 yields different Type I error rates across the two distributions, but it is difficult to discern

a definite pattern.

2.7.3 Normal Distribution

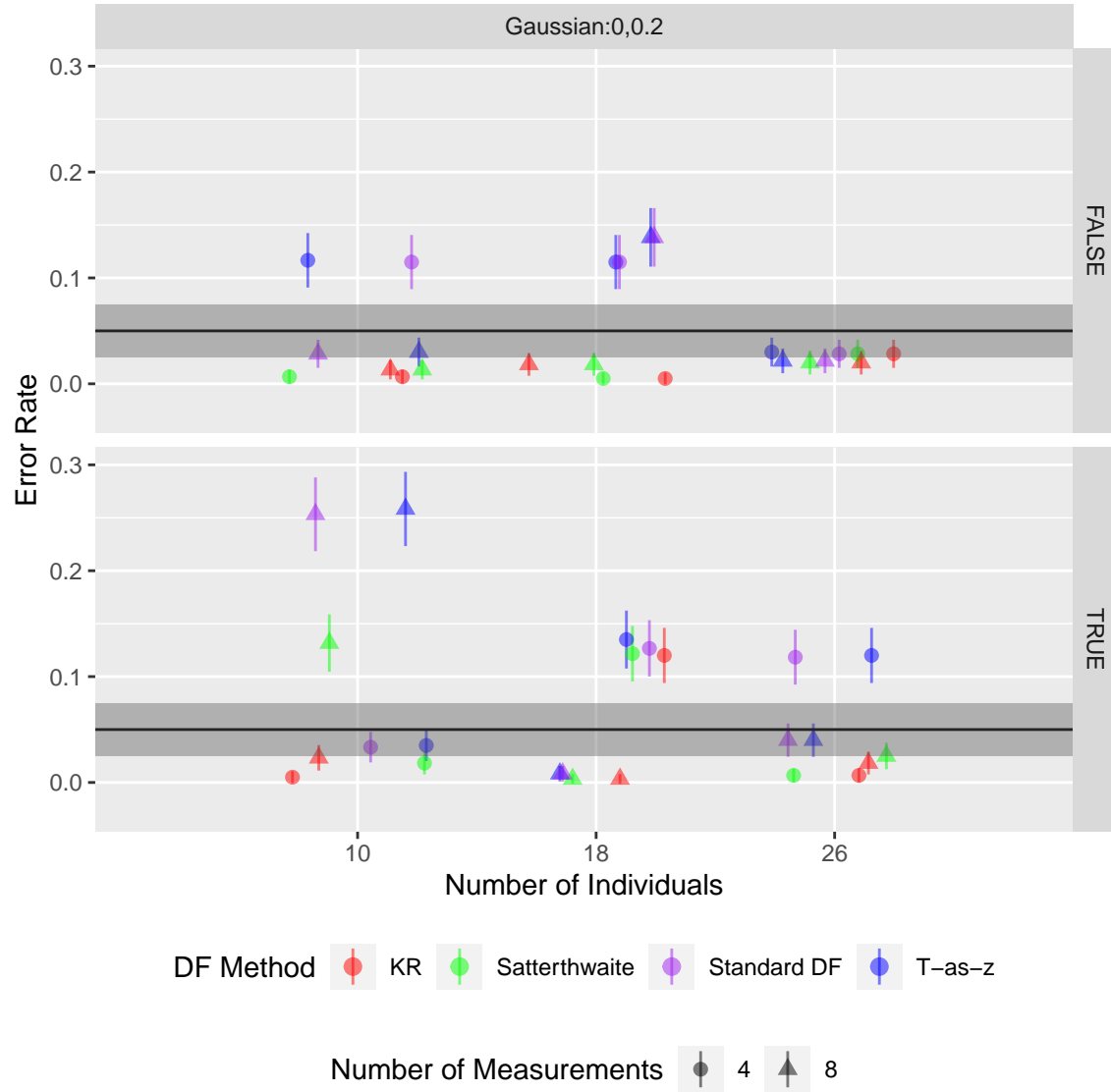


Figure 2.4: Type I Error Rates by DF Method for Normal Distribution

While nonnormal distributions are the focus of this study, comparing performance of DF methods to the normal distribution is important as a point of reference. As we can see in figure 2.4, it is interesting to note how robustness has decreased for DF

methods applied to the normal distribution compared to lognormal and exponential distributions, but it is also important to acknowledge that the methods, especially KR and Satterthwaite, produce conservative error rates, rather than anti-conservative ones. In most scenarios, KR and Satterthwaite remain unaffected when changing sample size. On the other hand, t -as- z and standard DF oftentimes produce the most anti-conservative error rates, but improvements arise when sample size is at 26.

Overall, in the normal distribution, there is a clearer distinction in performance of KR and Satterthwaite vs t -as- z and standard DF methods. Increasing the number of measurements is another factor that can affect Type I error rates produced by DF method within a particular method, rather than comparing across them.

2.7.4 KR vs Satterthwaite

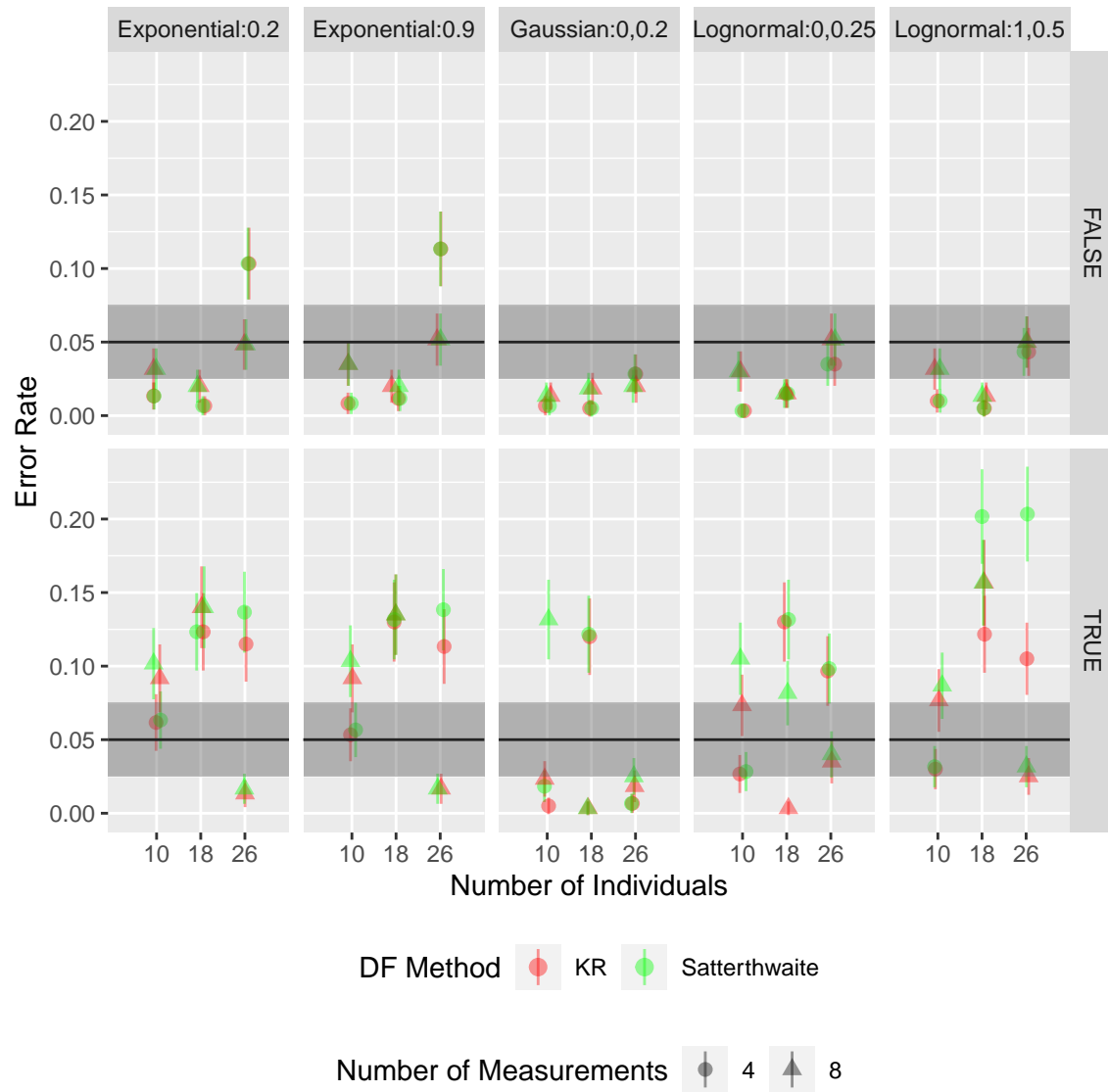


Figure 2.5: Type I Error Rates Produced by KR vs Satterthwaite

Comparing performance across all 4 methods has yielded significant evidence that KR and Satterthwaite are superior methods when using linear mixed models on small samples. Luke (2017) suggests that both KR and Satterthwaite are comparable solutions to obtain adequate Type I error. Figure 2.5 aims to narrow in on differences in perfor-

mance between the two methods. One can see that across random intercept models, KR and Satterthwaite methods have identical performance. Looking more closely at random slope models, it appears that KR consistently produces more conservative error rates when holding distribution, sample size, and number of measurements constant. In two scenarios, of which models were of size 10 and 18, the KR method produced more conservative error rates, but were not robust, while the Satterthwaite method produced robust error rates.

While in a majority of cases KR is a viable solution when aiming to keep Type I error rates low, it is not necessarily the optimal DF choice in every scenario if the goal is to produce error rates closest to .05.

Tables for all error rates produced by 4 DF methods are located in Appendix B

2.8 Discussion

After comparing the error rates produced by DF methods across three distributions, three sample sizes, and two numbers of measurements and complexity of models totaling to 96 unique conditions, we find that overall, KR and Satterthwaite DF methods yield the most robust error rates when constructing LMM with small sample sizes. In addition, random slope models are significantly more complicated in terms of using DF methods to produce robust error rates, and also challenge the superiority of the KR method over the Satterthwaite method when looking at robustness of Type I errors of random slope models of smaller sample sizes.

This stark contrast in performance between random slope and random intercept models across all distributions requires further investigation. Barr, Levy, Scheepers, & Tily (2013) suggest that linear mixed models with maximal effects (each predictor having its own random effect) are preferable and can reduce Type I error rates,

as random slope models can account for between individual variation in slopes and reduce residual variance. However, in their study the distribution of the variable of interest is normal, and random effects were generated from a bivariate normal distribution. In this simulation, random effects for the intercept and time were drawn from two nonnormal distributions which can further complicate the distribution of the outcome variable and complicate the fit of the model.

When comparing and contrasting performance of Satterthwaite and KR methods, which have both been demonstrated to be suitable options for producing robust Type I error, it seems that complexity of the model as well as sample size are two parameters that can differentiate their performance. In random intercept models, performance between the two methods are identical. In random slope models, KR is more conservative, and in a few conditions where the sample size is less than 26, it is too conservative in comparison to Satterthwaite. Given that KR is a further adjustment to Satterthwaite, its more conservative performance is not surprising; however, when evaluating fixed effects, conservative Type I error rates are preferable to anti-conservative rates. Thus, we prefer KR as the DF approximation in small and nonnormally distributed samples. In terms of sample size and number of measurements, it appears that increasing the number of measurements increases the proportion of DF methods that produce robust Type I error rates in most conditions. Difference in DF method performance across measurement numbers is most evident in random slope models with samples of size 26. On the other hand, increasing sample size is not strongly associated with increased robustness of Type I error rates produced by DF methods. Increasing from sample size from 10 to 18 produces inconclusive patterns: sometimes DF methods produce even more anti-conservative error rates, and in other cases there is virtually no difference. In most cases, sample size of 26 is the most ideal for robustness, but in some nonnormal distributions robustness

can only be achieved by also having more repeated measurements. It is plausible that the differences between a sample size of 10 and 18 are insignificant because both are too small in terms of a “large-enough” sample of size 30, so the DF methods are not impacted by this increase in sample. Models of sample size 26 may be ideal in some circumstances mainly because of its proximity to being considered sufficiently large enough.

After conducting our simulation study, we recommend KR and Satterthwaite DF methods when working with smaller sample sizes. However, there are other considerations and alternatives that have not been discussed. Similar to what has been discussed in Luke (2017), this exploration of performance of DF methods does not include analysis of power. While the authors of that study initially found that KR and Satterthwaite approximations yielded inferior power compared to other DF methods, after adjusting for Type I error, corrected power of KR and Satterthwaite methods were comparable. Additionally, there appears to be a relationship between sample size, number of measurements, and power, that could be further studied. Given that the sample sizes are small, caution is needed when interpreting results; furthermore, when attempting to generalize those results to this simulation, we must acknowledge that the effects of skewness and kurtosis on power in linear models are unknown.

Although this project focuses on evaluating significance of fixed effects using DF approximation, it would be remiss to not mention other possible methods that do not make assumptions about the degrees of freedom. Parametric bootstrapping involves taking repeated measures and computing an alternative LR test statistic. Luke (2017) found that it produced p-values with better Type I error rates than using LRT or t -as- z DF method, although it was sensitive to changes in sample size. The parametric bootstrap approach is worth exploring for small and nonnormal data, as Ye, Xu, Luo, & Jiang (2019) found adequate Type I error rates even when random effects

were skew-normal. The downside of parametric bootstrapping is that it is a computationally complex procedure, so DF approximation methods are a more feasible option (Luke, 2017). Future studies should explore the performance of a variety of methods involving and not involving DF approximation in nonnormal distributions.

In summary, while previous studies comparing DF methods focus primarily on normal distributions, the results from this study demonstrate that applying DF methods to LMM with nonnormal distributions and small sample sizes can still produce robust Type I error rates. However, careful consideration of the complexity of the model, number of measurements and individuals, as well as the skewness and kurtosis of the data must be considered, as they each have an effect on the DF method not only individually, but when interacting with other parameters as well. Overall, these results also indicate that Type I error rates are closest to .05 when employing Kenward-Roger and Satterthwaite methods.

Chapter 3 Application

3.1 Application to Longitudinal Study about Children's Health

In this chapter, we will apply linear mixed models to a longitudinal study about children's health, and explore how inference of fixed effects can possibly change when using various degrees of freedom approximation methods.

3.1.1 Background

The National Longitudinal Study of Adolescent to Adult Health is a longitudinal study spanning 1994 to 2008 that surveyed 15,000 U.S. students in 7-12th grade in the 1994-95 school year (Harris & Udry, 2022). Four waves of data were collected in 1994, 1996, 2001, and 2008; the sample during the first wave was aged 13-18. Questions about mental health, socioeconomic status, and family background were collected, as well as physical measurements of height and weight.

One question of interest to consider is how salient life experiences that occur during adolescence, such as being exposed to alcohol or being in a physical altercation, may impact changes to one's physical health over time. One way to capture physical health is through BMI, which tends to follow a right-skewed distribution (Penman & Johnson, 2006). We can see this in the distribution of children's BMI in figure ??.

While this dataset is large and encompasses approximately 5,000 students, the scope of this application will be narrowed in order to examine the performance of degrees of freedom methods, which are sensitive to sample size.

We will be focusing on Chinese female respondents who completed at least three waves of the study, which amounts to a sample size of 12. It is hypothesized that early exposure to substances could potentially affect changes in weight. Previous studies have demonstrated a significant relationship between alcohol consumption and BMI (Pasch, Velazquez, Cance, Moe, & Lytle, 2012). Thus, our model will use alcohol use as the predictor of BMI. At the first wave of the study in 1994, children were asked whether they had consumed alcohol before. This, along with their age at the start of the first wave, will be used to model changes in BMI across three waves.

3.1.2 Exploration

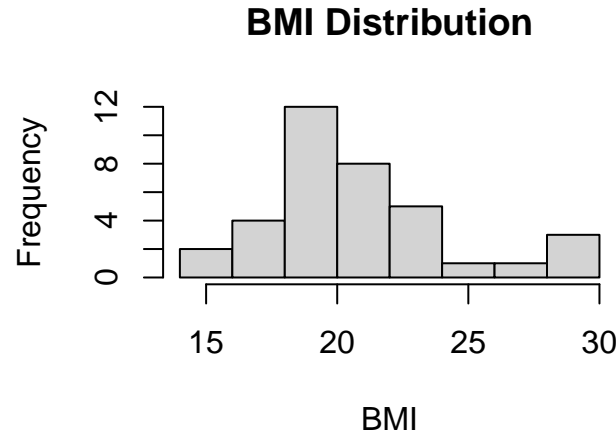


Figure 3.1: Distribution of BMI Across Sample

From this histogram depicting distribution of BMI, we can see that it follows a nonnormal trend. The skewness is .95 and the kurtosis is 3.39. The skewness

and kurtosis values for align most closely with that of the lognormal distribution $X \sim \text{Lognormal}(0, .25)$ used in the simulation study.

Figure 3.2 depicts changes in BMI over time for each individual in the study. The wide variation in BMI values between individuals suggest that including random effects for the intercept would be beneficial. Additionally, there are slight variations in pattern of change in BMI over time. At least two individuals in the sample have a pattern of BMI change that is noticeably different from the rest. Given that the sample size is small, it may not be necessary to examine implementing random effects for time, but it will be explored briefly.

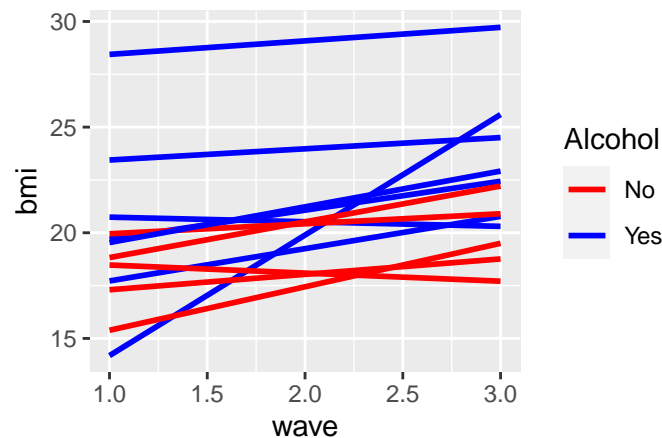


Figure 3.2: Changes in BMI by Individual and Alcohol Use

In a longitudinal study examining youth substance use and body composition, Pasch et al. (2012) found that baseline alcohol use is associated with lower values of BMI at follow-up. The opposite pattern can be seen in Figure 3.2. Individuals who had consumed alcohol during the first wave had higher BMI values across all time points. While the data may not be consistent with other research, it highlights the importance of substance use as a potential predictor of BMI.

While our sample size is small, our initial exploration of alcohol use suggests that it may be salient predictor for BMI. In addition, there are unique patterns of BMI

change over time by individual, which suggests that creating a mixed effects model with random effects for time and intercept may be useful.

3.1.3 Linear Mixed Model

Because we have repeated measurements of the same individual in this study, a regular linear model would not be appropriate as it assumes independence; implementing this model would inflate Type I error rates.

Table 3.1: Linear Model Output

term	estimate	std.error	statistic	p.value
(Intercept)	17.688	0.981	18.04	0.000
alcohol1	3.134	1.037	3.02	0.005
age_at_start_scaled	0.420	0.286	1.47	0.152
time	0.547	0.248	2.20	0.035

Table 3.1 shows the results of implementing a linear model with alcohol, age at first wave, and time as predictors. Alcohol use and time are significant predictors in this model. This will only used as a comparison to the other linear mixed models.

3.1.4 Intercept Only Model

Table 3.2: Fixed Effects for Random Intercept Model Using KR

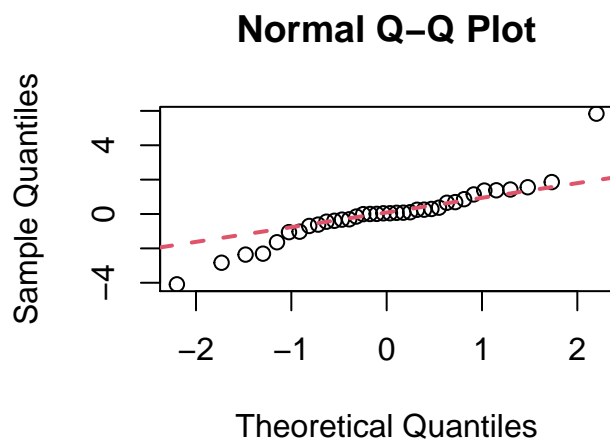
term	estimate	std.error	statistic	df	p.value
(Intercept)	17.688	1.313	13.475	10.5	0.000
alcohol1	3.134	1.653	1.897	9.0	0.090
age_at_start_scaled	0.420	0.456	0.921	9.0	0.381
time	0.547	0.157	3.491	23.0	0.002

Table 3.3: Random Effects for Random Intercept Model

group	term	estimate
AID	sd__(Intercept)	2.58
Residual	sd__Observation	1.93

Table 3.2 shows that time is the only significant effect in our model with only a random intercept. There is no significant relationship between early alcohol use and BMI. For individuals that have the same value for the random intercept, each year after the initial wave increases the predicted BMI by .5. In this model, an individual who is 16 years old at the initial wave and hasn't consumed alcohol has a predicted BMI of 17.7. Next, we turn to the random effects output in table 3.3. The variance for individuals (represented by AID), which depicts variability across individuals, is 6.7, while the residual variance, representing within-subject variability is 3.7. The significantly larger variance across individuals compared to within individuals suggests that distinguishing different types of variation is useful. The intraclass correlation is .64, which indicates that weight measurements taken of the same individual have

slightly higher similarity than those of different individuals.



When checking to see if the assumptions of the linear mixed model hold, we see that in the QQ plot the normality assumption of the residuals is not met. This is to be expected as the random effects generated are nonnormal, as well as the continuous outcome variable. We will proceed with evaluating the fixed effects in order to see how DF methods perform when normal assumptions do not hold.

Table 3.4: Fixed Effects for Random Intercept Model Using Satterthwaite

term	estimate	std.error	statistic	df	p.value
(Intercept)	17.688	1.313	13.475	10.5	0.000
alcohol1	3.134	1.653	1.897	9.0	0.090
age_at_start_scaled	0.420	0.456	0.921	9.0	0.381
time	0.547	0.157	3.491	23.0	0.002

The summary output referenced above uses Kenward-Roger DF approximation. Table 3.4 shows the output using Satterthwaite method. There is no significant

difference between performance of the two DF methods, which aligns with results of our random intercept models from our simulation study.

3.1.5 Intercept and Random Slope

While we explore the random slope model, it is important to recognize that this model may not be optimal given the sample size. Smaller sample sizes make parameter estimation more complicated; considering that the random slope model is more complex, we turn to the random intercept model as the most parsimonious.

Table 3.5: Fixed Effects for Random Slope Model Using KR

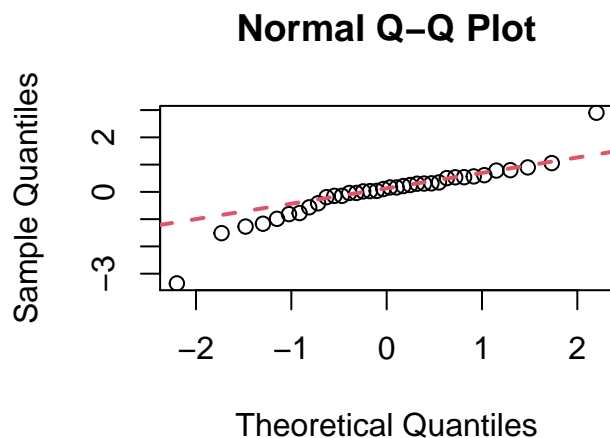
term	estimate	std.error	statistic	df	p.value
(Intercept)	17.500	1.448	12.083	11.5	0.000
alcohol1	3.458	1.756	1.969	9.0	0.080
age_at_start_scaled	0.346	0.485	0.714	9.0	0.493
time	0.547	0.193	2.835	11.0	0.016

Table 3.6: Random Effects for Random Slope Model

group	term	estimate
AID	sd__(Intercept)	3.328
AID	cor__(Intercept).time	-0.651
AID	sd__time	0.540
Residual	sd__Observation	1.401

In this model, we add a random effect of time as well as random intercept. This means that we are assuming that for each individual, the relationship between time and BMI is unique. Similar to the random intercept model, there is no significant

relationship between early alcohol use and BMI. Variability in BMI across individuals is 11.1, and the residual variance is .3 (as shown in table 3.6. The variance for time is 1.9, which represents variability across individual's BMI rates of change. We see that imposing variability between each individual's relationship of time vs BMI does not affect the residual variance.



Similar to the random intercept model, the normality condition is not met. We will proceed to evaluation.

Table 3.7: Fixed Effects for Random Slope Model Using Satterthwaite

term	estimate	std.error	statistic	df	p.value
(Intercept)	17.500	1.381	12.674	11.7	0.000
alcohol1	3.458	1.589	2.177	9.0	0.057
age_at_start_scaled	0.346	0.439	0.789	9.0	0.450
time	0.547	0.193	2.835	11.0	0.016

Earlier evaluation of fixed effects uses KR DF method, and table 3.7 shows significance of fixed effects when using the Satterthwaite. In the random slope model,

alcohol not a significant effect when using KR, nor is it significant when using Satterthwaite. However, the p-value for the fixed effect of alcohol use is lower when using Satterthwaite DF compared to KR. While the lower p-value was not enough to create differences in significance of predictors between the two methods, it is enough to justify why a simulation study comparing KR and Satterthwaite is relevant. It is likely that in other models, a lower p-value produced by Satterthwaite method could cause a fixed effect to be interpreted as significant, while it is interpreted as not significant by KR. By using the Satterthwaite method, we could possibly falsely conclude that a predictor is significant when it isn't. This aligns with our simulation results that demonstrated that Satterthwaite is slightly more anti-conservative than KR.

3.2 Discussion

We have demonstrated implementing two linear mixed models on a data set that is small and with a nonnormal continuous outcome. Imposing random effects structure to appropriately account for the correlation between repeated measures within each person improved the model and reduced the number of predictors that were significant. If we were to incorrectly use the linear model to explore BMI, we could have falsely concluded that alcohol use is a significant predictor of BMI. Additionally, one key result was that comparing KR and Satterthwaite DF methods resulted in varying p-values for fixed effects in the random slope model. Which DF method is preferable?

Looking at the results from our simulation study, we can identify which condition most closely resembles the children's health data, and determine if KR or Satterthwaite would produce more robust results. There was no significant difference in the p-values of the fixed effects of the random intercept model, so we only focus on the

random slopes model. The skewness and kurtosis values for the application data align most closely with that of the lognormal distribution $X \sim \text{Log}(0, .25)$. We will look at the performance of the two DF methods in this distribution, in a random slope model with a sample size of 10 and 4 measurements per individual. In reference to Figure 2.5, we see that the performance of KR and Satterthwaite are virtually the same. If this is the case, our preference will still be towards KR as it tends to produce slightly less anti-conservative Type I error rates.

Ultimately, our application study supports the idea that linear mixed models can be applied to small samples when using KR or Satterthwaite DF methods. While performance between the two can be equally robust in theory, they can possibly lead to different conclusions with the same fitted model.

Conclusion

In this study, we explored how Type I error rates from 4 different DF methods: Kenward-Roger, Satterthwaite, standard DF method, and t -as- z compare when evaluating significance of fixed effects of linear mixed models, particularly in small samples and in nonnormal distributions. Using a range of sample sizes, number of measurements, lognormal, and exponential distributions, as well as both random intercept and random slope models, we found distinct differences in performance of DF methods. DF methods applied to random slope models produced more anti-conservative Type I error rates in comparison to random intercept models. While increasing the number of measurements appears to increase robustness of most methods, the same effect for sample size was not identified. Across the 4 DF methods, KR and Satterthwaite DF were consistently more robust and less anti-conservative; while both are preferable, KR is consistently less anti-conservative, which makes it a more optimal choice. When fitting linear mixed models to a subset of a children's longitudinal health survey, we find that the p-values of fixed effects differ between KR and Satterthwaite DF, which has the potential to produce different conclusions about whether early alcohol use is associated with changes in BMI. Overall, our results indicate that performance of DF methods can vary greatly in small sample sizes, which can in turn impact the conclusions drawn about the fixed effects. Further exploration into other nonnormal distributions and other criteria that could impact DF method is needed.

Appendix A The First Appendix

This first appendix includes all of the R chunks of code that were hidden throughout the document (using the `include = FALSE` chunk tag) to help with readability and/or setup.

A.1 All Code:

```
# acthesis package loads dplyr, ggplot2, knitr, bookdown, and remotes
require(acthesis)
# Default width of R output
options(width = 65, digits = 3)
# Define a new code chunk option 'linewidht' for wrapping unruly R output
hook_output = knitr::knit_hooks$get('output')
knitr::knit_hooks$set(output = function(x, options) {
  # this hook is used only when the linewidht option is not NULL
  if (!is.null(n <- options$linewidht)) {
    x = knitr:::split_lines(x)
    # any lines wider than n should be wrapped
    if (any(nchar(x) > n)) x = strwrap(x, width = n, exdent = 3, indent = 3)
    x = paste(x, collapse = '\n')
  }
  hook_output(x, options)
})
# Note: set linewidht = 65 when needed.
# download latest thesis cover sheet
if(!file.exists("CoverSheet.pdf")){
  download.file("https://www.amherst.edu/system/files/media/Thesis%2520Copyright.p
}
library(mosaic)
```

```

library(kableExtra)
library(ggplot2)
library(SimMultiCorrData)
library(here)
library(tidyverse)
library(kableExtra)
knitr::opts_chunk$set(echo = FALSE, message = F, warning = F)
par(mfrow = c(2, 2))
curve(dlnorm(x, meanlog = 0, sdlog = .25), from = 0, to = 5,
      main = "Lognormal dist with meanlog 0 and sdlog .25", cex.main = .6)
curve(dlnorm(x, meanlog = 1, sdlog = .5), from = 0, to = 5,
      main = "Lognormal dist with meanlog 1 and sdlog .5", cex.main = .6)
x <- seq(0, 8, 0.1)
x_norm <- seq(-5, 5, .1)
plot(x, dexp(x, .2),
     type = "l",
     ylab = "", lwd = 2, col = "red", main = "Exponential dist with lambda = .2"
)

plot(x, dexp(x, .9),
     type = "l",
     ylab = "", lwd = 2, col = "red", main = "Exponential dist with lambda = .9"
)
# plot(x, dnorm(x, 0, .2))
options(scipen = 9999)
all_sim <- readRDS(here("SimulationData", "all_sim.rds"))

all_sim_sum <- all_sim %>%
  filter(KR_term == "treatment") %>%
  group_by(
    distribution,
    number_individuals,
    params,
    number_measurements,
    rslope,
    KR_effect, KR_term, skew, kurtosis
  ) %>%
  summarize(
    n = n(),
    KR_t1err = mean(KR_sig),
    KR_t1err_sd = sd(KR_sig),
    S_t1err = mean(S_sig),

```

```

    S_t1err_sd = sd(S_sig),
    Z_t1err = mean(Z_sig),
    Z_t1err_sd = sd(Z_sig),
    t_t1err = mean(t_sig),
    t_t1err_sd = sd(t_sig),
  )
transformed_data <- all_sim_sum %>%
  filter(KR_term == "treatment") %>%
  mutate(
    KR_t1err_1 = paste(KR_t1err, KR_t1err_sd),
    S_t1err_1 = paste(S_t1err, S_t1err_sd),
    Z_t1err_1 = paste(Z_t1err, Z_t1err_sd),
    t_t1err_1 = paste(t_t1err, t_t1err_sd),
  ) %>%
  pivot_longer(cols = c("KR_t1err_1", "S_t1err_1", "Z_t1err_1", "t_t1err_1"),
               names_to = "DF_method", values_to = "error_rate_1") %>%
  mutate(type = paste(distribution, params, sep = ":"))
transformed_data[c("error_rate", "sd")] <- str_split_fixed(transformed_data$error_rate_1,
                                                           " ", 2)

transformed_data <- transformed_data %>%
  mutate(
    error_rate = as.double(error_rate),
    sd = as.double(sd),
    method_number = paste(DF_method, number_measurements)
  ) %>%
  mutate(interval = 1.96 * (sd / sqrt(600)))

ggplot(aes(
  x = as.factor(number_individuals), y = error_rate, color = DF_method,
  ymin = error_rate - interval, ymax = error_rate + interval,
  group = as.factor(number_measurements)
), data = transformed_data) +
  facet_grid(rslope ~ type) +
  geom_pointrange(alpha = .4, aes(shape = as.factor(number_measurements)),
                 position = position_jitter(h = 0, w = 0.4)) +
  geom_hline(yintercept = .05) +
  scale_y_continuous(limits = c(-.02, .3)) +
  annotate("rect", xmin = -.1, xmax = 4, ymin = 0.025, ymax = .075,
         alpha = .4) +
  scale_color_manual(labels = c("KR", "Satterthwaite", "Standard DF", "T-as-z"),
                    values = c("red", "green", "purple", "blue")) +
  xlab("Number of Individuals") +

```

```

ylab("Error Rate") +
labs(color = "DF Method", shape = "Number of Measurements") +
theme(legend.position = "bottom", legend.box = "vertical")

ggplot(aes(
  x = as.factor(number_individuals), y = error_rate, color = DF_method,
  ymin = error_rate - interval, ymax = error_rate + interval,
  group = as.factor(number_measurements)
), data = transformed_data %>% filter(distribution == "Exponential")) +
  facet_grid(rslope ~ type) +
  geom_pointrange(alpha = .4, aes(shape = as.factor(number_measurements)),
    position = position_jitter(h = 0, w = 0.4)) +
  geom_hline(yintercept = .05) +
  scale_y_continuous(limits = c(-.02, .3)) +
  annotate("rect", xmin = -.1, xmax = 4, ymin = 0.025, ymax = .075,
    alpha = .4) +
  xlab("Number of Individuals") +
  scale_color_manual(labels = c("KR", "Satterthwaite", "Standard DF", "T-as-z"),
    values = c("red", "green", "purple", "blue")) +
  ylab("Error Rate") +
  labs(color = "DF Method", shape = "Number of Measurements") +
  theme(legend.position = "bottom", legend.box = "vertical")

ggplot(aes(
  x = as.factor(number_individuals), y = error_rate, color = DF_method,
  ymin = error_rate - interval, ymax = error_rate + interval,
  group = as.factor(number_measurements)
), data = transformed_data %>% filter(distribution == "Lognormal")) +
  facet_grid(rslope ~ type) +
  geom_pointrange(alpha = .4, aes(shape = as.factor(number_measurements)),
    position = position_jitter(h = 0, w = 0.4)) +
  geom_hline(yintercept = .05) +
  scale_y_continuous(limits = c(-.02, .3)) +
  annotate("rect", xmin = -.1, xmax = 4, ymin = 0.025, ymax = .075,
    alpha = .4) +
  xlab("Number of Individuals") +
  scale_color_manual(labels = c("KR", "Satterthwaite", "Standard DF", "T-as-z"),
    values = c("red", "green", "purple", "blue")) +
  ylab("Error Rate") +
  labs(color = "DF Method", shape = "Number of Measurements") +

```

```

    theme(legend.position = "bottom", legend.box = "vertical")
  ggplot(aes(
    x = as.factor(number_individuals), y = error_rate, color = DF_method,
    group = as.factor(number_measurements), ymin = error_rate - interval,
    ymax = error_rate + interval,
  ), data = transformed_data %>% filter(distribution == "Gaussian")) +
  facet_grid(rslope ~ type) +
  geom_pointrange(alpha = .5, aes(shape = as.factor(number_measurements)),
    position = position_jitter(h = 0, w = 0.3)) +
  geom_hline(yintercept = .05) +
  scale_y_continuous(limits = c(-.03, .30)) +
  annotate("rect",
    xmin = -.1, xmax = 4, ymin = 0.025, ymax = .075,
    alpha = .4
  ) +
  xlab("Number of Individuals") +
  scale_color_manual(labels = c("KR", "Satterthwaite", "Standard DF", "T-as-z"),
    values = c("red", "green", "purple", "blue")) +
  ylab("Error Rate") +
  labs(color = "DF Method", shape = "Number of Measurements") +
  theme(legend.position = "bottom", legend.box = "vertical")
  ggplot(aes(
    x = as.factor(number_individuals), y = error_rate,
    color = DF_method, group = as.factor(number_measurements),
    ymin = error_rate - interval, ymax = error_rate + interval
  ), data = transformed_data %>%
    filter(DF_method %in% c("KR_t1err_1", "S_t1err_1"))) +
  facet_grid(rslope ~ type) +
  geom_pointrange(alpha = .4, aes(shape = as.factor(number_measurements)),
    position = position_jitter(h = 0, w = 0.1)) +
  geom_hline(yintercept = .05) +
  annotate("rect", xmin = -.1, xmax = 4, ymin = 0.025, ymax = .075,
    alpha = .4) +
  xlab("Number of Individuals") +
  scale_color_manual(labels = c("KR", "Satterthwaite"),
    values = c("red", "green")) +
  ylab("Error Rate") +
  labs(color = "DF Method", shape = "Number of Measurements") +
  theme(legend.position = "bottom", legend.box = "vertical")
  all_sim_sum %>%
    filter(KR_term == "treatment") %>%
    pivot_longer(cols = c("KR_t1err", "S_t1err", "Z_t1err", "t_t1err"),

```

```

        names_to = "DF_method", values_to = "error_rate") %>%
filter(DF_method %in% c("KR_t1err", "S_t1err")) %>%
pivot_wider(
  names_from = c("rslope"),
  values_from = "error_rate", names_prefix = "slope"
) %>%
group_by(number_individuals, skew, kurtosis, DF_method) %>%
summarize(
  random_intercept = mean(slopeFALSE),
  random_slope = mean(slopeTRUE)
) %>%
pivot_wider(
  names_from = c("number_individuals"),
  values_from = c("random_intercept", "random_slope")
) %>%
select(
  skew, kurtosis, DF_method, random_intercept_10, random_slope_10,
  random_intercept_18,
  random_slope_18, random_intercept_26, random_slope_26
) %>%
ungroup() %>%
kbl() %>%
pack_rows("Exponential", 7, 8) %>%
pack_rows("Normal", 1, 2) %>%
pack_rows("Lognormal", 3, 6) %>%
add_header_above(c(
  " " = 3, "Random Intercept" = 1, "Random Slope" = 1, "Random Intercept" = 1,
  "Random Slope" = 1, "Random Intercept" = 1, "Random Slope" = 1
)) %>%
add_header_above(c(" " = 3, "10" = 2, "18" = 2, "26" = 2)) %>%
add_header_above(c(" " = 3, "Sample Size" = 6)) %>%
landscape()

all_sim_sum %>%
filter(KR_term == "treatment") %>%
pivot_longer(cols = c("KR_t1err", "S_t1err", "Z_t1err", "t_t1err"),
  names_to = "DF_method", values_to = "error_rate") %>%
filter(DF_method %in% c("KR_t1err", "S_t1err")) %>%
pivot_wider(
  names_from = c("rslope"),
  values_from = "error_rate", names_prefix = "slope"
) %>%

```

```

group_by(number_individuals, skew, kurtosis, DF_method) %>%
summarize(
  random_intercept = mean(slopeFALSE),
  random_slope = mean(slopeTRUE)
) %>%
pivot_wider(
  names_from = c("number_individuals"),
  values_from = c("random_intercept", "random_slope")
) %>%
select(
  skew, kurtosis, DF_method, random_intercept_10, random_slope_10,
  random_intercept_18,
  random_slope_18, random_intercept_26, random_slope_26
) %>%
ungroup() %>%
kbl() %>%
pack_rows("Exponential", 7, 8) %>%
pack_rows("Normal", 1, 2) %>%
pack_rows("Lognormal", 3, 6) %>%
add_header_above(c(
  " " = 3, "Random Intercept" = 1, "Random Slope" = 1, "Random Intercept" = 1,
  "Random Slope" = 1, "Random Intercept" = 1, "Random Slope" = 1
)) %>%
add_header_above(c(" " = 3, "10" = 2, "18" = 2, "26" = 2)) %>%
add_header_above(c(" " = 3, "Sample Size" = 6)) %>%
landscape()
all_sim_sum %>%
filter(KR_term == "treatment") %>%
pivot_longer(cols = c("KR_t1err", "S_t1err", "Z_t1err", "t_t1err"),
  names_to = "DF_method", values_to = "error_rate") %>%
filter(DF_method %in% c("Z_t1err", "t_t1err")) %>%
pivot_wider(
  names_from = c("rslope"),
  values_from = "error_rate", names_prefix = "slope"
) %>%
group_by(number_individuals, skew, kurtosis, DF_method) %>%
summarize(
  random_intercept = mean(slopeFALSE),
  random_slope = mean(slopeTRUE)
) %>%
pivot_wider(
  names_from = c("number_individuals"),

```

```

    values_from = c("random_intercept", "random_slope")
  ) %>%
  select(
    skew, kurtosis, DF_method, random_intercept_10, random_slope_10,
    random_intercept_18,
    random_slope_18, random_intercept_26, random_slope_26
  ) %>%
  ungroup() %>%
  kbl() %>%
  pack_rows("Exponential", 7, 8) %>%
  pack_rows("Normal", 1, 2) %>%
  pack_rows("Lognormal", 3, 6) %>%
  add_header_above(c(
    " " = 3, "Random Intercept" = 1, "Random Slope" = 1, "Random Intercept" = 1,
    "Random Slope" = 1, "Random Intercept" = 1, "Random Slope" = 1
  )) %>%
  add_header_above(c(" " = 3, "10" = 2, "18" = 2, "26" = 2)) %>%
  add_header_above(c(" " = 3, "Sample Size" = 6)) %>%
  landscape()
library(tidyverse)
library(SimMultiCorrData)
options(scipen = 99999)
library(lme4)
library(kableExtra)
library(lmerTest)
knitr::opts_chunk$set(echo = FALSE, message = F, warning = F)
child_data <- readRDS(here::here("child_data.Rds"))
child_data <- child_data %>%
  group_by(AID) %>%
  mutate(age_at_start = min(real_age)) %>%
  mutate(time = case_when(wave == 1 ~ 0, wave == 2 ~ 2, wave == 3 ~ 5))

child_data[, "age_at_start_scaled"] <- scale(child_data$age_at_start, scale = F)

hist(child_data$bmi, xlab = "BMI", main = "BMI Distribution")
ggplot(child_data, aes(wave, bmi, color = alcohol)) +
  geom_smooth(method = "lm", se = FALSE, aes(group = AID)) +
  scale_color_manual(labels = c("No", "Yes"), values = c("red", "blue")) +
  labs(color = "Alcohol")
lm.model <- lm(bmi ~ alcohol + age_at_start_scaled + time, child_data)
broom::tidy(lm.model) %>%
  kbl(caption = "Linear Model Output", booktabs = TRUE, digits = 3) %>%

```



```

kable_classic(full_width = F) %>%
kable_styling(latex_options = "HOLD_position")
intercept_model <- lmer(
  bmi ~ 1 + alcohol + age_at_start_scaled + (1 | AID) + time,
  child_data
)
KR <- broom.mixed::tidy(intercept_model, ddf = "Kenward-Roger") %>%
  filter(effect == "fixed") %>%
  dplyr::select(term, estimate, std.error, statistic, df, p.value)
S <- broom.mixed::tidy(intercept_model, ddf = "Satterthwaite") %>%
  filter(effect == "fixed") %>%
  dplyr::select(term, estimate, std.error, statistic, df, p.value)
KR %>%
  kbl(caption = "Fixed Effects for Random Intercept Model Using KR",
      booktabs = T, digits = 3) %>%
  kable_classic(full_width = F)%>%
  kable_styling(latex_options = "HOLD_position")
broom.mixed::tidy(intercept_model, ddf = "Kenward-Roger") %>%
  filter(effect == "ran_pars") %>%
  select(group, term, estimate) %>%
  kbl(caption = "Random Effects for Random Intercept Model",
      booktabs = T, digits = 3) %>%
  kable_classic(full_width = F)%>%
  kable_styling(latex_options = "HOLD_position")

qqnorm(residuals(intercept_model))
qqline(residuals(intercept_model), col = 2, lwd = 2, lty = 2)
S %>%
  kbl(caption = "Fixed Effects for Random Intercept Model Using Satterthwaite",
      booktabs = T, digits = 3) %>%
  kable_classic(full_width = F) %>%
  kable_styling(latex_options = "HOLD_position")
intercept_slope_model <- lmer(bmi ~ alcohol + age_at_start_scaled + time +
  (1 + time | AID), child_data)

S_s <- broom.mixed::tidy(intercept_slope_model, ddf = "Satterthwaite") %>%
  filter(effect == "fixed") %>%
  dplyr::select(term, estimate, std.error, statistic, df, p.value)

KR_s <- broom.mixed::tidy(intercept_slope_model, ddf = "Kenward-Roger") %>%
  filter(effect == "fixed") %>%
  dplyr::select(term, estimate, std.error, statistic, df, p.value)

```

```

KR_s %>%
  kbl(caption = "Fixed Effects for Random Slope Model Using KR",
      booktabs = T, digits = 3) %>%
  kable_classic(full_width = F) %>%
  kable_styling(latex_options = "HOLD_position")

broom.mixed::tidy(intercept_slope_model, ddf = "Kenward-Roger") %>%
  filter(effect == "ran_pars") %>%
  select(group, term, estimate) %>%
  kbl(caption = "Random Effects for Random Slope Model",
      booktabs = T, digits = 3) %>%
  kable_classic(full_width = F) %>%
  kable_styling(latex_options = "HOLD_position")
qqnorm(residuals(intercept_slope_model))
qqline(residuals(intercept_slope_model), col = 2, lwd = 2, lty = 2)

S_s %>%
  kbl(caption = "Fixed Effects for Random Slope Model Using Satterthwaite",
      booktabs = T, digits = 3) %>%
  kable_classic(full_width = F) %>%
  kable_styling(latex_options = "HOLD_position")
library(here)
library(tidyverse)
all_sim <- readRDS(here("SimulationData", "all_sim.rds"))

all_sim_sum <- all_sim %>%
  filter(KR_term == "treatment") %>%
  group_by(
    distribution,
    number_individuals,
    params,
    number_measurements,
    rslope,
    KR_effect, KR_term, skew, kurtosis
  ) %>%
  summarize(
    n = n(),
    KR_tterr = mean(KR_sig),
    KR_tterr_sd = sd(KR_sig),
    S_tterr = mean(S_sig),
    S_tterr_sd = sd(S_sig),

```

```

    Z_t1err = mean(Z_sig),
    Z_t1err_sd = sd(Z_sig),
    t_t1err = mean(t_sig),
    t_t1err_sd = sd(t_sig),
  )

all_sim_sum %>%
  filter(KR_term == "treatment") %>%
  pivot_longer(cols = c("KR_t1err", "S_t1err", "Z_t1err", "t_t1err"),
               names_to = "DF_method", values_to = "error_rate") %>%
  filter(DF_method %in% c("KR_t1err", "S_t1err")) %>%
  pivot_wider(
    names_from = c("rslope"),
    values_from = "error_rate", names_prefix = "slope"
  ) %>%
  group_by(number_individuals, skew, kurtosis, DF_method) %>%
  summarize(
    random_intercept = mean(slopeFALSE, na.rm = TRUE),
    random_slope = mean(slopeTRUE, na.rm = TRUE)
  ) %>%
  pivot_wider(
    names_from = c("number_individuals"),
    values_from = c("random_intercept", "random_slope")
  ) %>%
  select(
    skew, kurtosis, DF_method, random_intercept_10, random_slope_10,
    random_intercept_18,
    random_slope_18, random_intercept_26, random_slope_26
  ) %>%
  ungroup() %>%
  kbl() %>%
  pack_rows("Exponential", 7, 8) %>%
  pack_rows("Normal", 1, 2) %>%
  pack_rows("Lognormal", 3, 6) %>%
  add_header_above(c(
    " " = 3, "Random Intercept" = 1, "Random Slope" = 1, "Random Intercept" = 1,
    "Random Slope" = 1, "Random Intercept" = 1, "Random Slope" = 1
  )) %>%
  add_header_above(c(" " = 3, "10" = 2, "18" = 2, "26" = 2)) %>%
  add_header_above(c(" " = 3, "Sample Size" = 6)) %>%
  kable_styling(latex_options="scale_down")

```

```

all_sim_sum %>%
  filter(KR_term == "treatment") %>%
  pivot_longer(cols = c("KR_t1err", "S_t1err", "Z_t1err", "t_t1err"),
               names_to = "DF_method", values_to = "error_rate") %>%
  filter(DF_method %in% c("Z_t1err", "t_t1err")) %>%
  pivot_wider(
    names_from = c("rslope"),
    values_from = "error_rate", names_prefix = "slope"
  ) %>%
  group_by(number_individuals, skew, kurtosis, DF_method) %>%
  summarize(
    random_intercept = mean(slopeFALSE, na.rm = TRUE),
    random_slope = mean(slopeTRUE, na.rm = TRUE)
  ) %>%
  pivot_wider(
    names_from = c("number_individuals"),
    values_from = c("random_intercept", "random_slope")
  ) %>%
  select(
    skew, kurtosis, DF_method, random_intercept_10, random_slope_10,
    random_intercept_18,
    random_slope_18, random_intercept_26, random_slope_26
  ) %>%
  ungroup() %>%
  kbl() %>%
  pack_rows("Exponential", 7, 8) %>%
  pack_rows("Normal", 1, 2) %>%
  pack_rows("Lognormal", 3, 6) %>%
  add_header_above(c(
    " " = 3, "Random Intercept" = 1, "Random Slope" = 1, "Random Intercept" = 1,
    "Random Slope" = 1, "Random Intercept" = 1, "Random Slope" = 1
  )) %>%
  add_header_above(c(" " = 3, "10" = 2, "18" = 2, "26" = 2)) %>%
  add_header_above(c(" " = 3, "Sample Size" = 6)) %>%
  kable_styling(latex_options="scale_down")

```

```

library(tidyverse)
library(lmerTest)
library(SimMultiCorrData)
library(purrr)
library(broom)

```

```

library(broom.mixed)

## Function that returns output from LME model using KR & Satterthwaite DF
sim_mixedeffects <- function(dist, parameters, random_slopes = TRUE,
                             n.indiv, n.measurements, corr) {
  corr.data <- c(1, corr, corr, 1)
  corr1 <- matrix(corr.data, nrow = 2)

  # save kurtosis and skew values
  m <- round(calc_theory(Dist = dist, params = parameters), 8)

  if (random_slopes == TRUE) {
    r_effects <- rcorrvar(
      n = n.indiv, k_cont = 2, method = "Fleishman",
      means = c(m[1], m[1]), vars = c(m[2]^2, (m[2]^2) / 18),
      skews = c(m[3], m[3]),
      skurts = c(m[4], m[4]),
      errorloop = TRUE, rho = corr1, seed = sample(1000:9999, 1)
    )

    # centering the mean at 0
    r_effects1 <- r_effects$continuous_variables$V1 - m[1]
    r_effects2 <- r_effects$continuous_variables$V2 - m[1]

    errors <- rnorm(n.indiv * n.measurements, 0, m[2] * sqrt(.168))
  } else {
    # random intercept model
    r_effects1 <- nonnormvar1("Fleishman",
      means = m[1], vars = m[2]^2, skews = m[3],
      skurts = m[4], n = n.indiv, seed = sample(1000:9999, 1)
    )$continuous_variable$V1

    r_effects1 <- r_effects1 - m[1]
    r_effects2 <- rep(0, n.indiv)

    errors <- rnorm(n.indiv * n.measurements, 0, m[2] * sqrt(.4))
  }

  data <- expand.grid(time = 1:n.measurements, indiv = as.factor(1:n.indiv))
  data$treatment <- rep(0:1, each = (n.indiv * n.measurements) / 2)

  # model matrix for fixed effects

```

```

matrix_fixedeffects <- model.matrix(~ time + treatment, data)

# fixed effects coefficient vector
betas <- c(3.1, 0, 0)

# model matrix for random intercepts
matrix_intercept <- model.matrix(~ 0 + indiv, data)

# model matrix for random slopes
matrix_slope <- model.matrix(~ 0 + indiv, data) * data$time

data$Y_val <- (matrix_fixedeffects %*% betas +
  matrix_intercept %*% r_effects1 +
  matrix_slope %*% r_effects2 + errors)[, 1]

# construct model
if (random_slopes == FALSE) {
  model <- lmerTest::lmer(Y_val ~ time + (1 | indiv) + treatment, data)
} else {
  model <- lmerTest::lmer(Y_val ~ time + (time | indiv) + treatment, data)
}

# extract p-values/summary output from KR and Satterthwaite
KR <- broom.mixed::tidy(model, ddf = "Kenward-Roger")
colnames(KR) <- paste("KR", colnames(KR), sep = "_")

S <- broom.mixed::tidy(model, ddf = "Satterthwaite") %>%
  rename_with(.cols = everything(), .fn = ~ str_c("S_", .x)) %>%
  add_column(
    .before = "S_effect",
    distribution = dist, number_individuals = n.indiv,
    params = paste0(parameters, collapse = ","),
    number_measurements = n.measurements,
    rslope = random_slopes, mean = m[1], sd = m[2], skew = m[3],
    kurtosis = m[4],
    fifth = m[5], sixth = m[6]
  )

```

```

    return(cbind(S, KR))
}

## Setting up factors for lognormal condition
dist <- c("Lognormal")
corr <- c(-.38)
parameters <- list(c(0, .25), c(1, .5))
random_slopes <- c(TRUE, FALSE)
n.indiv <- c(10, 18, 26)
n.measurements <- c(4, 8)

# all possible combinations
combos_lnormal <- crossing(
  dist, parameters, random_slopes,
  n.indiv, n.measurements, corr
)

## Setting up factors for exponential condition
dist <- c("Exponential")
corr <- c(-.38)
parameters <- c(.2, .9)
random_slopes <- c(TRUE, FALSE)
n.indiv <- c(10, 18, 26)
n.measurements <- c(4, 8)

# all possible combinations
combos_exp <- crossing(
  dist, parameters, random_slopes,
  n.indiv, n.measurements, corr
)

numsim <- 80

##### Lognormal Distribution: 400 total simulations
set.seed(2021)

simulations_lnormal_1 <- 1:numsim %>%
  map_df(~ pmap_dfr(list(

```

```

    combos_lnormal$dist, combos_lnormal$parameters,
    combos_lnormal$random_slopes, combos_lnormal$n.indiv,
    combos_lnormal$n.measurements, combos_lnormal$corr
  ),
  .f = sim_mixedeffects
))

# saveRDS(simulations_lnormal_1, "simulations_lnormal_1.rds")
set.seed(2005)

simulations_lnormal_2 <- 1:numsim %>%
  map_df(~ pmap_dfr(list(
    combos_lnormal$dist, combos_lnormal$parameters,
    combos_lnormal$random_slopes, combos_lnormal$n.indiv,
    combos_lnormal$n.measurements, combos_lnormal$corr
  )),
  .f = sim_mixedeffects
))
# saveRDS(simulations_lnormal_2, "simulations_lnormal_2.rds")

set.seed(2011)

simulations_lnormal_3 <- 1:numsim %>%
  map_df(~ pmap_dfr(list(
    combos_lnormal$dist, combos_lnormal$parameters,
    combos_lnormal$random_slopes, combos_lnormal$n.indiv,
    combos_lnormal$n.measurements, combos_lnormal$corr
  )),
  .f = sim_mixedeffects
))
# saveRDS(simulations_lnormal_3, "simulations_lnormal_3.rds")

set.seed(1012)
simulations_lnormal_4 <- 1:numsim %>%
  map_df(~ pmap_dfr(list(
    combos_lnormal$dist, combos_lnormal$parameters,
    combos_lnormal$random_slopes, combos_lnormal$n.indiv,
    combos_lnormal$n.measurements, combos_lnormal$corr
  )),
  .f = sim_mixedeffects
))
# saveRDS(simulations_lnormal_4, "simulations_lnormal_4.rds")

```



```

set.seed(3434)
simulations_lnormal_5 <- 1:numsim %>%
  map_df(~ pmap_dfr(list(
    combos_lnormal$dist, combos_lnormal$parameters,
    combos_lnormal$random_slopes, combos_lnormal$n.indiv,
    combos_lnormal$n.measurements, combos_lnormal$corr
  )),
  .f = sim_mixedeffects
)
# saveRDS(simulations_lnormal_5, "simulations_lnormal_5.rds")

##### Normal distribution: 400 simulations

set.seed(7006)
simulations_normal_1 <- 1:numsim %>%
  map_df(~ pmap_dfr(list(
    combos_normal$dist, combos_normal$parameters,
    combos_normal$random_slopes, combos_normal$n.indiv,
    combos_normal$n.measurements, combos_normal$corr
  )),
  .f = sim_mixedeffects
)
# saveRDS(simulations_normal_1, "simulations_normal_1.rds")

set.seed(3692)
simulations_normal_2 <- 1:numsim %>%
  map_df(~ pmap_dfr(list(combos_normal$dist, combos_normal$parameters,
    combos_normal$random_slopes, combos_normal$n.indiv,
    combos_normal$n.measurements, combos_normal$corr),
    .f = sim_mixedeffects))
# saveRDS(simulations_normal_2, "simulations_normal_2.rds")

set.seed(0814)
simulations_normal_3 <- 1:numsim %>%
  map_df(~ pmap_dfr(list(combos_normal$dist, combos_normal$parameters,
    combos_normal$random_slopes, combos_normal$n.indiv,
    combos_normal$n.measurements, combos_normal$corr),
    .f = sim_mixedeffects))

```

```

# saveRDS(simulations_normal_3, "simulations_normal_3.rds")

set.seed(0904)
simulations_normal_4 <- 1:numsim %>%
  map_df(~ pmap_dfr(list(combos_normal$dist, combos_normal$parameters,
                        combos_normal$random_slopes, combos_normal$n.indiv,
                        combos_normal$n.measurements, combos_normal$corr),
                    .f = sim_mixedeffects))
# saveRDS(simulations_normal_4, "simulations_normal_4.rds")

set.seed(1111)
simulations_normal_5 <- 1:numsim %>%
  map_df(~ pmap_dfr(list(combos_normal$dist, combos_normal$parameters,
                        combos_normal$random_slopes, combos_normal$n.indiv,
                        combos_normal$n.measurements, combos_normal$corr),
                    .f = sim_mixedeffects))
# saveRDS(simulations_normal_5, "simulations_normal_5.rds")

##### Exponential Distribution: 400 simulations

set.seed(9998)
simulations_exp_1 <- 1:numsim %>%
  map_df(~ pmap_dfr(list(combos_exp$dist, combos_exp$parameters,
                        combos_exp$random_slopes, combos_exp$n.indiv,
                        combos_exp$n.measurements, combos_exp$corr),
                    .f = sim_mixedeffects))
# saveRDS(simulations_exp_1, "simulations_exp_1.rds")

set.seed(9997)
simulations_exp_2 <- 1:numsim %>%
  map_df(~ pmap_dfr(list(combos_exp$dist, combos_exp$parameters,
                        combos_exp$random_slopes, combos_exp$n.indiv,
                        combos_exp$n.measurements, combos_exp$corr),
                    .f = sim_mixedeffects))
# saveRDS(simulations_exp_2, "simulations_exp_2.rds")

set.seed(4035)
simulations_exp_3 <- 1:numsim %>%
  map_df(~ pmap_dfr(list(combos_exp$dist, combos_exp$parameters,
                        combos_exp$random_slopes, combos_exp$n.indiv,

```

```

        combos_exp$n.measurements, combos_exp$corr),
        .f = sim_mixedeffects))
# saveRDS(simulations_exp_3, "simulations_exp_3.rds")

set.seed(42847)
simulations_exp_4 <- 1:numsim %>%
  map_df(~ pmap_dfr(list(combos_exp$dist, combos_exp$parameters,
                        combos_exp$random_slopes, combos_exp$n.indiv,
                        combos_exp$n.measurements, combos_exp$corr),
                        .f = sim_mixedeffects))
# saveRDS(simulations_exp_4, "simulations_exp_4.rds")

set.seed(0505)
simulations_exp_5 <- 1:numsim %>%
  map_df(~ pmap_dfr(list(combos_exp$dist, combos_exp$parameters,
                        combos_exp$random_slopes, combos_exp$n.indiv,
                        combos_exp$n.measurements, combos_exp$corr),
                        .f = sim_mixedeffects))
# saveRDS(simulations_exp_5, "simulations_exp_5.rds")

```


Appendix B P-values from DF Methods

```
library(here)
library(tidyverse)
all_sim <- readRDS(here("SimulationData", "all_sim.rds"))

all_sim_sum <- all_sim %>%
  filter(KR_term == "treatment") %>%
  group_by(
    distribution,
    number_individuals,
    params,
    number_measurements,
    rslope,
    KR_effect, KR_term, skew, kurtosis
  ) %>%
  summarize(
    n = n(),
    KR_t1err = mean(KR_sig),
    KR_t1err_sd = sd(KR_sig),
    S_t1err = mean(S_sig),
    S_t1err_sd = sd(S_sig),
    Z_t1err = mean(Z_sig),
    Z_t1err_sd = sd(Z_sig),
    t_t1err = mean(t_sig),
    t_t1err_sd = sd(t_sig),
  )

all_sim_sum %>%
  filter(KR_term == "treatment") %>%
  pivot_longer(cols = c("KR_t1err", "S_t1err", "Z_t1err", "t_t1err"),
               names_to = "DF_method", values_to = "error_rate") %>%
  filter(DF_method %in% c("KR_t1err", "S_t1err")) %>%
```

skew kurtosis DF_method			Sample Size					
			10		18		26	
			Random Intercept	Random Slope	Random Intercept	Random Slope	Random Intercept	Random Slope
skew	kurtosis	DF_method	random_intercept_10	random_slope_10	random_intercept_18	random_slope_18	random_intercept_26	random_slope_26
Normal								
0.000	0.0	KR_tterr	0.010	0.014	0.012	0.062	0.024	0.013
0.000	0.0	S_tterr	0.010	0.075	0.012	0.062	0.024	0.016
Lognormal								
0.778	1.1	KR_tterr	0.017	0.050	0.015	0.067	0.043	0.066
0.778	1.1	S_tterr	0.017	0.067	0.015	0.107	0.043	0.069
1.750	5.9	KR_tterr	0.021	0.053	0.009	0.139	0.047	0.065
1.750	5.9	S_tterr	0.021	0.059	0.009	0.179	0.047	0.118
Exponential								
2.000	6.0	KR_tterr	0.022	0.075	0.015	0.132	0.079	0.065
2.000	6.0	S_tterr	0.022	0.081	0.015	0.132	0.079	0.077

```

pivot_wider(
  names_from = c("rslope"),
  values_from = "error_rate", names_prefix = "slope"
) %>%
group_by(number_individuals, skew, kurtosis, DF_method) %>%
summarize(
  random_intercept = mean(slopeFALSE, na.rm = TRUE),
  random_slope = mean(slopeTRUE, na.rm = TRUE)
) %>%
pivot_wider(
  names_from = c("number_individuals"),
  values_from = c("random_intercept", "random_slope")
) %>%
select(
  skew, kurtosis, DF_method, random_intercept_10, random_slope_10,
  random_intercept_18,
  random_slope_18, random_intercept_26, random_slope_26
) %>%
ungroup() %>%
kbl() %>%
pack_rows("Exponential", 7, 8) %>%
pack_rows("Normal", 1, 2) %>%
pack_rows("Lognormal", 3, 6) %>%
add_header_above(c(
  " " = 3, "Random Intercept" = 1, "Random Slope" = 1, "Random Intercept" = 1,
  "Random Slope" = 1, "Random Intercept" = 1, "Random Slope" = 1
)) %>%
add_header_above(c(" " = 3, "10" = 2, "18" = 2, "26" = 2)) %>%
add_header_above(c(" " = 3, "Sample Size" = 6)) %>%
kable_styling(latex_options="scale_down")

```

```

all_sim_sum %>%
  filter(KR_term == "treatment") %>%
  pivot_longer(cols = c("KR_t1err", "S_t1err", "Z_t1err", "t_t1err"),
               names_to = "DF_method", values_to = "error_rate") %>%
  filter(DF_method %in% c("Z_t1err", "t_t1err")) %>%
  pivot_wider(
    names_from = c("rslope"),
    values_from = "error_rate", names_prefix = "slope"
  ) %>%
  group_by(number_individuals, skew, kurtosis, DF_method) %>%
  summarize(
    random_intercept = mean(slopeFALSE, na.rm = TRUE),
    random_slope = mean(slopeTRUE, na.rm = TRUE)
  ) %>%
  pivot_wider(
    names_from = c("number_individuals"),
    values_from = c("random_intercept", "random_slope")
  ) %>%
  select(
    skew, kurtosis, DF_method, random_intercept_10, random_slope_10,
    random_intercept_18,
    random_slope_18, random_intercept_26, random_slope_26
  ) %>%
  ungroup() %>%
  kbl() %>%
  pack_rows("Exponential", 7, 8) %>%
  pack_rows("Normal", 1, 2) %>%
  pack_rows("Lognormal", 3, 6) %>%
  add_header_above(c(
    " " = 3, "Random Intercept" = 1, "Random Slope" = 1, "Random Intercept" = 1,
    "Random Slope" = 1, "Random Intercept" = 1, "Random Slope" = 1
  )) %>%
  add_header_above(c(" " = 3, "10" = 2, "18" = 2, "26" = 2)) %>%
  add_header_above(c(" " = 3, "Sample Size" = 6)) %>%
  kable_styling(latex_options="scale_down")

```

			Sample Size					
			10		18		26	
			Random Intercept	Random Slope	Random Intercept	Random Slope	Random Intercept	Random Slope
skew	kurtosis	DF_method	random_intercept_10	random_slope_10	random_intercept_18	random_slope_18	random_intercept_26	random_slope_26
Normal								
0.000	0.0	t_tlerr	0.072	0.143	0.127	0.068	0.025	0.079
0.000	0.0	Z_tlerr	0.073	0.147	0.127	0.072	0.026	0.080
Lognormal								
0.778	1.1	t_tlerr	0.024	0.122	0.029	0.148	0.043	0.071
0.778	1.1	Z_tlerr	0.067	0.124	0.030	0.159	0.048	0.071
1.750	5.9	t_tlerr	0.027	0.082	0.013	0.179	0.055	0.122
1.750	5.9	Z_tlerr	0.028	0.083	0.025	0.189	0.057	0.122
Exponential								
2.000	6.0	t_tlerr	0.028	0.126	0.038	0.135	0.085	0.100
2.000	6.0	Z_tlerr	0.044	0.128	0.038	0.169	0.087	0.100

Corrections

A list of corrections after submission to department.

Corrections may be made to the body of the thesis, but every such correction will be acknowledged in a list under the heading “Corrections,” along with the statement “When originally submitted, this honors thesis contained some errors which have been corrected in the current version. Here is a list of the errors that were corrected.” This list will be given on a sheet or sheets to be appended to the thesis. Corrections to spelling, grammar, or typography may be acknowledged by a general statement such as “30 spellings were corrected in various places in the thesis, and the notation for definite integral was changed in approximately 10 places.” However, any correction that affects the meaning of a sentence or paragraph should be described in careful detail. The files `samplethesis.tex` and `samplethesis.pdf` show what the “Corrections” section should look like. Questions about what should appear in the “Corrections” should be directed to the Chair.

References

- Arnau, J., Bono, R., Blanca, M. J., & Bendayan, R. (2012). Using the linear mixed model to analyze nonnormal data distributions in longitudinal designs. *Behavior Research Methods*, 44(4), 1224–1238. <http://doi.org/10.3758/s13428-012-0196-y>
- Arnau, J., Bono, R., & Vallejo, G. (2009). Analyzing small samples of repeated measures data with the mixed-model adjusted *f* test. *Communications in Statistics - Simulation and Computation*, 38(5), 1083–1103. <http://doi.org/10.1080/03610910902785746>
- Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language*, 68(3), 255–278. <http://doi.org/10.1016/j.jml.2012.11.001>
- Bradley, J. V. (1978). Robustness? *British Journal of Mathematical and Statistical Psychology*, 31(2), 144–152. <http://doi.org/https://doi.org/10.1111/j.2044-8317.1978.tb00581.x>
- Fialkowski, A. C. (2018). *SimMultiCorrData: Simulation of correlated data with multiple variable types*. Retrieved from <https://CRAN.R-project.org/package=SimMultiCorrData>
- Fitzmaurice, G. M., Laird, N. M., & Ware, J. H. (2011). *Applied longitudinal analysis*

(2nd ed). Hoboken, N.J: Wiley.

Harris, K. M., & Udry, J. R. (2022). National longitudinal study of adolescent to adult health (add health), 1994-2018 [public use]. Carolina Population Center, University of North Carolina-Chapel Hill [distributor], Inter-university Consortium for Political; Social Research [distributor]. <http://doi.org/10.3886/ICPSR21600.v24>

Horner, R. D. (1987). Age At Onset Of Alzheimer’S Disease: Clue To The Relative Importance Of Etiologic Factors?1. *American Journal of Epidemiology*, 126(3), 409–414. <http://doi.org/10.1093/oxfordjournals.aje.a114672>

Hrongs-Tai Fai, A., & Cornelius, P. L. (1996). Approximate f-tests of multiple degree of freedom hypotheses in generalized least squares analyses of unbalanced split-plot experiments. *Journal of Statistical Computation and Simulation*, 54(4), 363–378. <http://doi.org/10.1080/00949659608811740>

Kuznetsova, A., Brockhoff, P. B., & Christensen, R. H. B. (2017). **LmerTest** package: Tests in linear mixed effects models. *Journal of Statistical Software*, 82(13). <http://doi.org/10.18637/jss.v082.i13>

Limpert, E., Stahel, W. A., & Abbt, M. (2001). Log-normal distributions across the sciences: Keys and clues: On the charms of statistics, and how mechanical models resembling gambling machines offer a link to a handy way to characterize log-normal distributions, which can provide deeper insight into variability and probability—normal or log-normal: That is the question. *BioScience*, 51(5), 341–352. [http://doi.org/10.1641/0006-3568\(2001\)051%5B0341:LNDATS%5D2.0.CO;2](http://doi.org/10.1641/0006-3568(2001)051%5B0341:LNDATS%5D2.0.CO;2)

Luke, S. G. (2017). Evaluating significance in linear mixed-effects models in r. *Be-*

- havior Research Methods*, 49(4), 1494–1502. <http://doi.org/10.3758/s13428-016-0809-y>
- Pasch, K. E., Velazquez, C. E., Cance, J. D., Moe, S. G., & Lytle, L. A. (2012). Youth substance use and body composition: Does risk in one area predict risk in the other? *Journal of Youth and Adolescence*, 41(1), 14–26. <http://doi.org/10.1007/s10964-011-9706-y>
- Penman, A. D., & Johnson, W. D. (2006). The changing shape of the body mass index distribution curve in the population: Implications for public health policy to reduce the prevalence of adult obesity. *Preventing Chronic Disease*, 3(3), A74–A74. Retrieved from <https://pubmed.ncbi.nlm.nih.gov/16776875>
- Rencher, A. C., & Schaalje, G. B. (2007). Linear models in statistics. <http://doi.org/10.1002/9780470192610>
- Satterthwaite, F. E. (1946). An approximate distribution of estimates of variance components. *Biometrics Bulletin*, 2(6), 110–114. Retrieved from <http://www.jstor.org/stable/3002019>
- Schaalje, G. B., McBride, J. B., & Fellingham, G. W. (2002). Adequacy of approximations to distributions of test statistics in complex mixed linear models. *Journal of Agricultural, Biological, and Environmental Statistics*, 7(4), 512–524. <http://doi.org/10.1198/108571102726>
- Ye, R.-d., Xu, L.-j., Luo, K., & Jiang, L. (2019). A parametric bootstrap approach for one-way classification model with skew-normal random effects. *Applied Mathematics-A Journal of Chinese Universities*, 34(4), 423–435. <http://doi.org/10.1007/s11766-019-3564-x>
- Zeng, Y., Vaupel, J., Xiao, Z., Liu, Y., & Zhang, C. (2017). Chinese longitudinal

healthy longevity survey (CLHLS), 1998-2014. Inter-university Consortium for Political; Social Research [distributor]. <http://doi.org/10.3886/ICPSR36692>.

v1