A study on the type I error rate and power for generalized linear mixed model containing one random effect

by

Yu Wang

B.S., Dalian University of Technology, 2007
M.S., Huazhong University of Science and Technology, 2009

A REPORT

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Statistics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2017

Approved by:

Major Professor
Dr. Christopher I. Vahl

Copyright

Abstract

In animal health research, it is quite common for a clinical trial to be designed to demonstrate the efficacy of a new drug where a binary response variable is measured on an individual experimental animal (i.e., the observational unit). However, the investigational treatments are applied to groups of animals instead of an individual animal. This means the experimental unit is the group of animals and the response variable could be modeled with the binomial distribution. Also, the responses of animals within the same experimental unit may then be statistically dependent on each other. The usual logit model for a binary response assumes that all observations are independent. In this report, a logit model with a random error term representing the group of animals is considered. This is model belongs to a class of models referred to as generalized linear mixed models and is commonly fit using the SAS System procedure PROC GLIMMIX. Furthermore, practitioners often adjust the denominator degrees of freedom of the test statistic produced by PROC GLIMMIX using one of several different methods. In this report, a simulation study was performed over a variety of different parameter settings to compare the effects on the type I error rate and power of two methods for adjusting the denominator degrees of freedom, namely "DDFM = KENWARDROGER" and "DDFM = NONE". Despite its reputation for fine performance in linear mixed models with normally distributed errors, the "DDFM = KENWARDROGER" option tended to perform poorly more often than the "DDFM = NONE" option in logistic regression model with one random effect.

Table of Contents

List of Figures

List of Tables

Acknowledgements

First foremost, I would like to express my sincere gratitude to the Kansas State University and the Statistics Department for giving me the opportunity to achieve my master program and this research experience.

Then, I would like to acknowledge my major professor Dr. Christopher Vahl who presented me with this topic and for his incredible patience, support and encouragement during the time of my master research in this program.

Especially, I would also like to give many thanks to my report committee members, Dr. Perla Reyes and Dr. Abigail Jager for their support and always quick replies to my endless emails.

Furthermore, I would like to express my gratitude to Dr. Gary L. Gadbury and Dr. Weixing Song for providing me with the GTA position.

Finally, I want to dedicate this work to my family and friends for their support and belief in my success. Special recognition goes out to my husband and two lovely kids, for their support and encouragement during my research and study.

Chapter 1 - Introduction

1.1 Background and Problem

In animal health experiments, it is fairly common that the treatments are applied to groups of animals. This means the experimental unit is defined as the entire group instead of an individual animal. However, the response variable is often measured on the each animal separately so that the observational unit is an individual animal rather than the group of animals. Under such circumstances, the observational units within the same group may be correlated with each other.

For example, consider a study conducted on young, largemouth bass. A total number of 1,600 fingerlings from the reference population were randomly assigned to the eight test tanks. We assume that approximately equal numbers of fish (about 200) were allocated to each tank. The external and internal environment for the reference tanks were kept approximately the same. The investigational treatment was Chloramine-T which was administered as a bath to four randomly selected tanks out of the total of eight. The other four tanks received no chemical treatment. Therefore, there are two treatment groups, test and control, where the test group received Chloramine-T and the control group did not. The test treatment is intended to prevent an early death of the treated fish, so the number of dead fish out of the total fish allotted to the tank was observed for each tank. (FDA, 2014).

The design structure of the experiment described above is a typically called a completely randomized design (CRD) where the tank is the experimental unit and the individual fish is the observational unit. There are four replicates in each of the two treatment groups. The test tanks were assumed to be independent of each other but the fish within same tank were not completely independent from one another since they were fed with the same food and received the same dose of medicine while experiencing the same conditions of the given tank. The response variable of this experiment is the cumulative mortality rate of the fingerlings in a given tank, but it is calculated by first determining the health status of each fish. Each fish is assigned a number equal to 1 or 0 where 1 means the fish died before the end of the study and 0 means the fish was still alive at the end of the study. Under this circumstance, the response for each fish is considered as a binary variable with outcomes "dead" and "alive". The response variable of the whole tank, computed by summing the status values over all fish in the tank, follows a binomial distribution with the parameters n (number of fish in the tank) and probability of death equal to p within a given tank.

The traditional methodology for the analysis of a binary response is to fit a logistic regression model ignoring the effect of the individual tanks. The treatment was defined as the fixed effect and the tank was defined as a random effect. For this situation, the logistic regression must be modified to account for the additional uncertainty due to tank-to-tank variability. We will still use a logistic regression model but we will assume that tanks are random effects. This takes us from a generalized linear regression model to a generalized linear mixed model.

1.2 Generalized Linear Mixed Model

In this section, we will address the major algorithms of generalized linear mixed models. I will start with a brief review of linear regression models and linear mixed model. In general, the linear mixed models are extensions of linear regression models. In both these two types of models, the relationship between the response variables and predictor variables is that the response variables are normally distributed with the mean. The way to determine the models is through a linear function by the predictor variables (Pinherio, 2004). The distinction between these two models is that linear mixed models have both fixed and random effects. However, in some common cases, the response variables could come from any distribution besides normal distributions, a binary response for instance. As in the case stated in this report, the response variable is a count, it is not normally distributed but is still included in the exponential distribution family. Therefore, the linear mixed model is not applicable to this case. We need consider more applicable model to this situation where the generalized linear mixed models come into the place. Generalized linear mixed models are also extensions of generalized linear models (Fikret Isik, 2011). It enables a way to fit the model where the response variables are not necessarily normally distributed. Alternatively, one could consider the generalized linear mixed model (GLMM) is an extension of the linear mixed model (Fikret Isik, 2011). “Generalized” stands for the response variables that are non-normal distributions and “mixed” refers to the model contains random effects in addition to the fixed effects (Wang, 2004). “Rather than modeling the responses directly through the linear function, it allows predictor variables to be related to the mean response via a link function (e.g., the logistic function) and by allowing one to also model the variance component of each random effect” (SAS 9.2User's Guide).

In the fish experiment, let π_i be the probability that a fish within the i^{th} treatment group dies during the study period. If the individual fish was the experimental unit in a completely randomized design that includes treatments as the fixed effect, the response variable for each treatment group (i.e. the total number of dead fish) would follow a binomial distribution and we get the traditional logit model in the following form:

$$\log(\pi_i / 1 - \pi_i) = \mu + \tau_i ,$$

where μ is the overall mean, and τ_i is the i^{th} treatment effects, both of μ and τ_i are fixed effects. If y_i is the number of dead fish in the i^{th} treatment group, then y_i is a binomial random variable with probability of success π_i which is a function of μ and τ_i through the inverse of the logit link.

In many experiments, the observations exhibit a certain form of dependency. In this study, fish within a tank are all more (or less) likely to die than fish in another tank even within the same treatment group. We need to adjust the model above to contain the random effects on the probability of death due to the tank the fish are allotted. This leads to a generalized linear mixed model (GLMM) with the binomial response and one random effect. The treatment is as the fixed effect and the tank is the random effect.

In the model, the response variable for the k^{th} fish in the j^{th} tank within the i^{th} treatment group is y_{ijk} with y_{ijk} = 1 or 0 (1 if the fish died or 0 if it still lived) where $i = 1$ or 2 (with 1 = "test" and 2 = "control"), $j = 1, \ldots, 4$, and $k = 1, \ldots, n_{ij}$ where n_{ij} is equal to the number of fish in the tank (i.e., approximately 200). Now $y_{ij} = \sum_{k} y_{ijk} \sim binomial(\pi_{ij}, n_{ij})$ where π_{ij} is the true probability of success for j^{th} tank within the i^{th} treatment group.

Considering that the observational units within the tank are correlated with each other, we add the random effect of the tank within a treatment, β_{ij} , to the previous model which then becomes:

$$\log(\pi_{ij} / 1 - \pi_{ij}) = \mu + \tau_i + \beta_{ij} \ .$$

Both μ and τ_i are defined as before, but now $\beta_{ij} \sim N(o, \sigma^2_{\text{tank}})$ are independent errors associated with the tanks and represent a slight increase (or decrease) in the probability of death for all fish in the same tank.

Based on this generalized linear mixed model, $\log(\pi_i / 1 - \pi_i) = \mu + \tau_i$ represents the logit of the marginal probability of death in the i^{th} treatment group averaging over all possible tanks.

We will now examine how the choice of DDFM methods for approximating the degrees of freedom affects the type I error rates and power of the hypothesis test.

1.3 The brief introduction of DDFM Options for GLMM

In the last a few decades, generalized linear mixed model has been broadly used in fitting model from agronomy, animal science, clinical trials, economics and etc. The generalized linear mixed model is particularly useful in the analysis of animal experiments (Kackar,1984), it provides better flexibility and practicability over many regression models and assumes a very straightforward structure of dependence among response and predictors.

In SAS software, generalized linear mixed model data analysis is achieved by using the PROC GLIMMIX procedure. The way to control the denominator degrees of freedom for the model is by the “DDFM = Methods” options in the PROC GLMMIX procedure. The “DDFM = Methods” option allows you to specify the method of computation for the denominator degrees of freedom. There are several options you can select. “The computed degrees of freedom are also used in determining the degrees of freedom which can be used in hypothesis tests and confidence intervals from the other SAS statement.” (See the SAS/STAT User’s Guide, Mix Modeling,SAS Institute Inc., for more details).

Based on the model of the designed experiment of Chapter 1, there are two DDFM options that will be compared in terms of how the different methods affect the type I error rates and power for generalized linear mix model. The present report considers the “DDFM = NONE” method and “DDFM = KENWARDROGER” method.

1.3.1 Brief discussion of the DDFM = NONE Method

“DDFM = NONE”, just as it implies, indicates that none denominator degrees of freedom were applied when the clause execute. For example, when we execute the F test, PROC GLIMMIX procedure assumes that infinite degrees of freedom of the denominators are used while p-values are being calculated. So F test is convert to the Chi-square test, in this case, the p-value in this case is equal to $p = \Pr\{F_{l,\infty} > F_{obs}\} = \Pr\{\chi_l^2 > lF_{obs}\}$. See the SAS/STAT User's Guide, Mix Modeling (SAS Institute Inc.) for more details.

1.3.2 Brief discussion of the DDFM = KENWARDROGER Method

The "DDFM = KENWARDROGER" option applied the Kenward-Roger method to compute denominator degrees of freedom. The method was derived by Keyword and Roger (1997). Since the method has been implemented in the model procedure of the SAS system, it has become well known and widely used by many statisticians and researchers. Generally speaking, this method adjusts the denominator degrees of freedom and was derived from the linear mixed model with a continuous normal response. It was not derived for use in a generalized linear mixed model.

1.4 Research Objective of This Report

Many researchers use the "DDFM = KENWARDROGER" option when executing the PROC GLIMMIX procedure, this may cause some inaccurate results. This method was initially proposed by Kenward and Roger (1997) and it performs well when the parameterizations of the covariance matrix are linear. However, this approach does not perform as well when covariance structures are nonlinear. When we use the "DDFM = KENWARDROGER" option to calculate the degrees of freedom in PROC GLIMMIX procedure, computational formula of the underlying Taylor series expansion has a missing part, which is the critical reason of the problem when using it to accommodate the estimators of the covariance structure (Kenward and Roger, 2009). This report provides insight through a simulative example of an animal experiment data that were fit to a generalized linear mixed model with a logit link where the response variable follows a binomial distribution. The discussion and comparison of this study is focused on how the “DDFM = KENWARDROGER” option affects the type I error rates and power of the hypothesis test under multiple parameter settings here. Because the Kenward-Roger adjustment was not created for the generalized linear mixed model, we hypothesized that it would make it more difficult to reject the null hypothesis of no difference and decrease the power of the test as well as depress the type I error rate to an unacceptably conservative level.

Chapter 2 - Simulated Data Example of the CRD Experiment

2.1 Profile of the Example Data

In the fish experiment, we assumed the response variable for a given tank follows a binomial distribution, i.e. $y_{ij} \sim binomial(n_{ij}, \pi_{ij})$, where n_{ij} represents the number of fish in each test tank. For simplicity, we assumed the number of fish in each tank was the same, i.e. $n_{ij} = n$. The number of dead fish collected from each test tank was simulated using the binomial distribution. Because the mortality rate of fish will be analyzed by a generalized linear mixed model with a binomial response variable and a logit link function, the probability a fish dies in the *j*th tank within the *i*th treatment groups was set to:

$$\pi_{ij} = \frac{\exp\left\{\mu + \tau_i + \beta_{ij}\right\}}{1 + \exp\left\{\mu + \tau_i + \beta_{ij}\right\}},$$

where $\pi_i = \exp\left\{\mu + \tau_i\right\} / 1 + \exp\left\{\mu + \tau_i\right\}$ is the marginal mortality rate in the *i*th treatment group and the random effects denoted by β_{ij} are independent and identically distributed as $N(0, \sigma^2_{\text{tank}})$ random variables. The model for this experiment is then the generalized linear mixed model given by $\log(\pi_{ij} / 1 - \pi_{ij}) = \mu + \tau_i + \beta_{ij}$ where *i* = 1 or 2, *j* = 1, … , 4 and *k* = 1, … , *n*.

2.2 Value of the Variance Component for the Random Effect

For a given simulated dataset, we propose to test for an effect due to treatment using the following hypotheses:

$$H_0 : \pi_1 = \pi_2$$
$$H_a : \pi_1 \neq \pi_2$$

using the *F*-test based on the type III sums of squares provided by SAS Proc GLIMMIX. Care must be taken when specifying the variance component of the tank-to-tank variability since the linear model is fitted on the logit scale. For example, consider two binary populations with probabilities of success π_1 and π_2, respectively. These two populations are often compared by using an odds ratio. An odds ratio of 1.5 indicates a modest difference between π_1 and π_2. The log odds ratio, (i.e. on the same scale as the logit model) is then approximately 0.4, that is

$$\log \frac{\pi_1(1-\pi_2)}{\pi_2(1-\pi_1)} \approx 0.4.$$

Considering that almost an entire normal distribution is within three standard deviations, a standard deviation larger than 0.4/3 ≈ 0.13 would virtually wash out this difference and make it, on a practical level unimportant as the tank-to-tank variability would be the true driving force behind the mortality rates. Therefore, we have selected the two possible values for σ_{tank} to study as 0.1 and 0.05 so that the variance of the tank effects is either $\sigma_{\text{tank}}^2 = 0.01$ or $\sigma_{\text{tank}}^2 = 0.0025$.

2.3 Other Parameter Settings of the Simulation Study

The simulation study was conducted to compare between the “DDFM = NONE” and “DDFM = KENWARDROGER” methods impact on the type I error rates and the power of the hypothesis test.

2.3.1 Simulation Study on Type I Error Rates of Hypothesis Test

When we do hypothesis testing, we actually want to test if the difference is existing between two reference populations (treatment groups and control groups). The type I error is defined as when incorrectly rejecting a null hypothesis when it is actually true. In other words, the type I error is often referred to as a “false positive”, it is when rejecting H_0 in the favor of alternative hypothesis H_α. If we reject H_0 when the p-value is less than or equal to α, the significance level, then α is the nominal type I error rate. When H_0 above is true, $\pi_1 = \pi_2 = \pi$.

Here we consider only one value for the common probability of death, $\pi = 0.5$. We fix the number of tanks within the two treatment groups to be 4 tanks each and then allocate the same number of fish to each tank (i.e. n) and simulate data under values of n = 50, 80, 100, 120, 140, 160, 180 and 200. This results in 16 different scenarios of parameter settings (2 possible tank variances × 8 different possible tank sizes) – see Table 2.1 for an enumeration of all settings. Ten thousand datasets were simulated under each of the 16 scenarios with p-values calculated using the two different DDFM methods. The type I error rate was estimated as the proportion of p-values less than or equal to α 0.05 out of 10,000.

Table 2.1 Parameter settings of the sample data for comparing the type I error rates of the hypothesis

n	σ_{tank}	$\pi_1 = \pi_2$
50	0.1/0.05	0.5
80	0.1/0.05	0.5
100	0.1/0.05	0.5
120	0.1/0.05	0.5
140	0.1/0.05	0.5
160	0.1/0.05	0.5
180	0.1/0.05	0.5
200	0.1/0.05	0.5

2.3.2 Simulation Study on Power of the Hypothesis Test

In practice, researchers are mostly interested in whether or not there is difference between the treatments. Power is the probability that the null hypothesis will be rejected when it is actually false (Cohen 1988). Thus, power quantifies the chance to make a correct decision in rejecting the null hypothesis under a particular setting of the parameters that is true under H_α. Here we assumed that $\pi_1 = 0.5$ and $\pi_2 = 0.6$ so that the log odds ratio is approximately -0.4 (the test treatment decreases the log odds of death). We consider the same values of variance component for tank to be $\sigma_{\text{tank}} = 0.10$ and $\sigma_{\text{tank}} = 0.05$ as well as the same numbers of fish allocated to each tank (*n*= 50, 80, 100, 120, 140, 160, 180 and 200). Then we conduct the PROC GLIMMIX procedure to build the generalized linear mixed model and then use "DDFM = KENWARDROGER" method and "DDFM = NONE" method in the model clause. We analyze them and see how many times the PROC GLIMMIX procedure would reject H_0 for these two "DDFM = Methods" options at significant level $\alpha = 0.05$, in order to calculate and estimate of the power of the two DDFM methods when testing the two-sided alternative hypothesis vs. a null of no difference. All combinations of the parameter setting are summarized in Table 2.2 below.

Table 2.2 Parameter settings of the sample data for comparing the power of the hypothesis

n	σ_{tank}	π_1	π_2
50	0.1/0.05	0.5	0.6
80	0.1/0.05	0.5	0.6
100	0.1/0.05	0.5	0.6
120	0.1/0.05	0.5	0.6
140	0.1/0.05	0.5	0.6
160	0.1/0.05	0.5	0.6
180	0.1/0.05	0.5	0.6
200	0.1/0.05	0.5	0.6

Chapter 3 - Simulation Study

3.1 Simulation Procedure

For each selected configuration of parameters:

(1) Generate a CRD dataset with binomial responses for each tank with the treatment as the fixed effect and the tanks as random effects with normal distributions for a given setting the parameters.

(2) Use PROC GLIMMIX procedure, with logit link function to fit the generalized linear mixed models and using "DDFM = KENWARDROGER" or "DDFM = NONE" methods for computing the denominator degrees of freedom for hypothesis test and observe a p-value.

(3) Repeat step (1) and (2) for N = 10,000 times.

(4) Compute the type I error rates or the power for the hypothesis test at significant level $\alpha = 0.05$, then report the estimated Type I error rates or the estimated power of the hypothesis test under each parameter setting.

(5) perform steps (1) to (4) for each of the 16 parameter settings.

(6) Display and summarize the results in the table.

The complete SAS code is provided in Appendix B.

3.2 Simulation Result

3.2.1 Simulation Result for Type I Error Rates of Hypothesis Test

Table 3.1 displays a complete list of simulation result of the Type I error rates. Sample size *n* (representing the number of fish in each tank) for the control groups and treatment groups were assumed to be the same. We assume that the marginal probability of success for treatment groups and control groups are identical to 0.5. The comparison is based on the two different values of variance component, $\sigma_{\text{tank}} = 0.1$ and $\sigma_{\text{tank}} = 0.05$. From this table, we can see for both two methods, the type I error rates are increasing as the sample size are getting larger. The type I error rates of “DDFM = KENWARDROGER” method is much smaller than the type I error rates of “DDFM = NONE” method in each scenario. The type I error rates for “DDFM = NONE” method are approximate 0.05, this is very close to the significant level $\alpha = 0.05$. The type I error rates for “DDFM = KENWARDROGER” slightly increase as sample sizes increase but is still far below the nominal significance level of $\alpha = 0.05$. Each scenario shows similar performance, “DDFM = NONE” method more liberally rejects H_0 in favor of H_a than it should at a significance level of $\alpha = 0.05$. Therefore, we can conclude that the “DDFM = KENWARDROGER” method for the type I error rates of hypothesis test is too conservative in the generalized linear mix model.

3.2.2 Simulation Result for Power of Hypothesis Test

The simulation results for power are shown in the table 3.2. Sample size *n* (represent the number of fish in each tank) for the control groups and treatment groups were assumed to be the same. We consider the different cases for sample size from 50 to 200, and we assume that the marginal probability of success for treatment groups is equal 0.5 and the marginal probability of success for control groups is equal 0.6, respectively. The comparison is based on the two different values $\sigma_{\text{tank}} = 0.1$ and $\sigma_{\text{tank}} = 0.05$. From table 3.2, we can see that, as expected, bigger sample size increases the power of hypothesis for both two methods. However, the “DDFM = KENWARDROGER” method shrinks the power of hypothesis. Even for the small value of variance component and large sample size, “DDFM = KENWARDROGER” method does not perform as well as the “DDFM = NONE” method. Under ideal circumstances, the power of the test above 80% is generally accepted as good. However, the biggest power of “DDFM = KENWARDROGER” method among the 16 scenarios is still below 0.5.

Table 3.1 Comparison of the type I error rates for different parameter settings on DDFM = KENWARDROGER (KR) and DDFM = NONE options

n	$\pi_1 = \pi_2 = 0.5\,;\sigma_{\text{tank}} = 0.1$		$\pi_1 = \pi_2 = 0.5\,;\sigma_{\text{tank}} = 0.05$	
50	KR	0.0029	KR	0.0024
	NONE	0.0465	NONE	0.0398
80	KR	0.0059	KR	0.0039
	NONE	0.0546	NONE	0.0435
100	KR	0.0085	KR	0.0067
	NONE	0.0579	NONE	0.0463
120	KR	0.0076	KR	0.0055
	NONE	0.0579	NONE	0.0434
140	KR	0.0076	KR	0.0040
	NONE	0.0608	NONE	0.0452
160	KR	0.0077	KR	0.0027
	NONE	0.0580	NONE	0.0421
180	KR	0.0080	KR	0.0027
	NONE	0.0609	NONE	0.0439
200	KR	0.0088	KR	0.0025
	NONE	0.0620	NONE	0.0446

Table 3.2 Comparison of the power for different parameter settings on DDFM = KENWARDROGER (KR) and DDFM = NONE options

n	$\pi_1 = 0.5;\,\pi_2 = 0.6\,;\sigma_{\text{tank}} = 0.1$		$\pi_1 = 0.5;\,\pi_2 = 0.6\,;\sigma_{\text{tank}} = 0.05$	
50	KR	0.0891	KR	0.0820
	NONE	0.4287	NONE	0.4480
80	KR	0.1705	KR	0.1542
	NONE	0.6946	NONE	0.6425
100	KR	0.2324	KR	0.2046
	NONE	0.7058	NONE	0.7363
120	KR	0.2873	KR	0.2520
	NONE	0.7486	NONE	0.8067
140	KR	0.3428	KR	0.2958
	NONE	0.7971	NONE	0.8615
160	KR	0.3721	KR	0.3345
	NONE	0.8271	NONE	0.9010
180	KR	0.4300	KR	0.3690
	NONE	0.8635	NONE	0.9257
200	KR	0.4695	KR	0.3981
	NONE	0.8827	NONE	0.9471

Chapter 4 - Discussion of Simulation Study

To get a better understanding the range of the type I error rates and power of hypothesis for the two methods, trend line and dot plots were constructed to show graph of type I error rates and power for each simulation setting.

4.1 The Discussion and Comparison of the Type I Error Rate for the DDFM = KENWARDROGER and DDFM = NONE

We perform simulation studies on different parameter settings of the type I error rates for the generalized linear mixed model with one random effect. The figure 4.1-4.2 displays the comparison of the type I error rates for the two methods. We could visually compare the trend of type I error rates over different sample sizes. The red line is representing the type I error rates of "DDFM = NONE" methods and the blue line is representing the type I error rates for "DDFM = KENWARDROGER" methods.

Figure 4.1 displays the estimated type I error rates when the variance component $\sigma_{\text{tank}} = 0.1$. We could see that the type I error rates for the "DDFM = NONE" seems to be acceptable under all simulation settings. They are approximately close to 0.05, if not slightly too liberal. Under ideal circumstances it would be preferred to have a type I error rate around to the value of significant level $\alpha = 0.05$, indicated by the reference line. The type I error rates for "DDFM = KENWARDROGER" are quite away from the reference line, closer to the interval of 0 to 0.01.

Figure 4.2 displays the estimated type I error rates when the variance component $\sigma_{\text{tank}} = 0.05$. We could see the type I error rates for the "DDFM = NONE" are below the reference line, but still within the tolerance bound. It becomes clearly apparent that the type I error rates for

“DDFM = KENWARDROGER” are more conservative (smaller) than expected. Each simulation parameters setting combination was replicated 10,000 times and the convergence rate for simulation studies on type I error rates are between 99.52% to 99.98%.

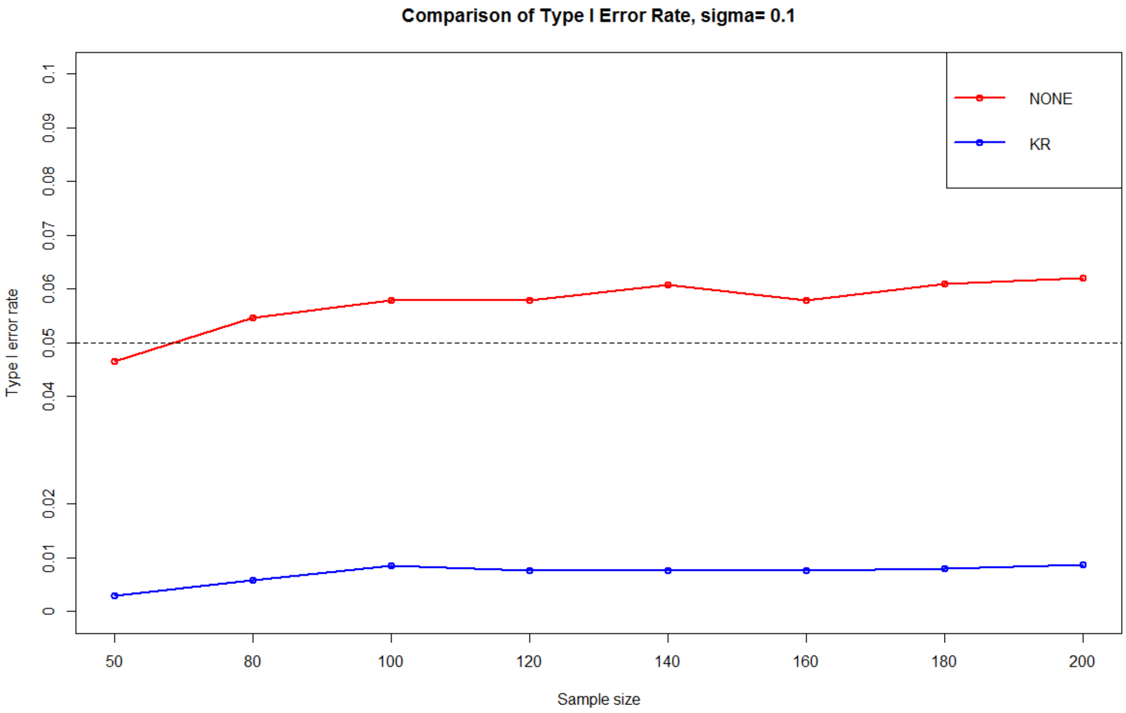

Figure 4.1 The trend line of type I error rates for the two DDFM methods when the variance component equal 0.1, *n* = 50, 80, 100, 120, 140, 160, 180, 200.

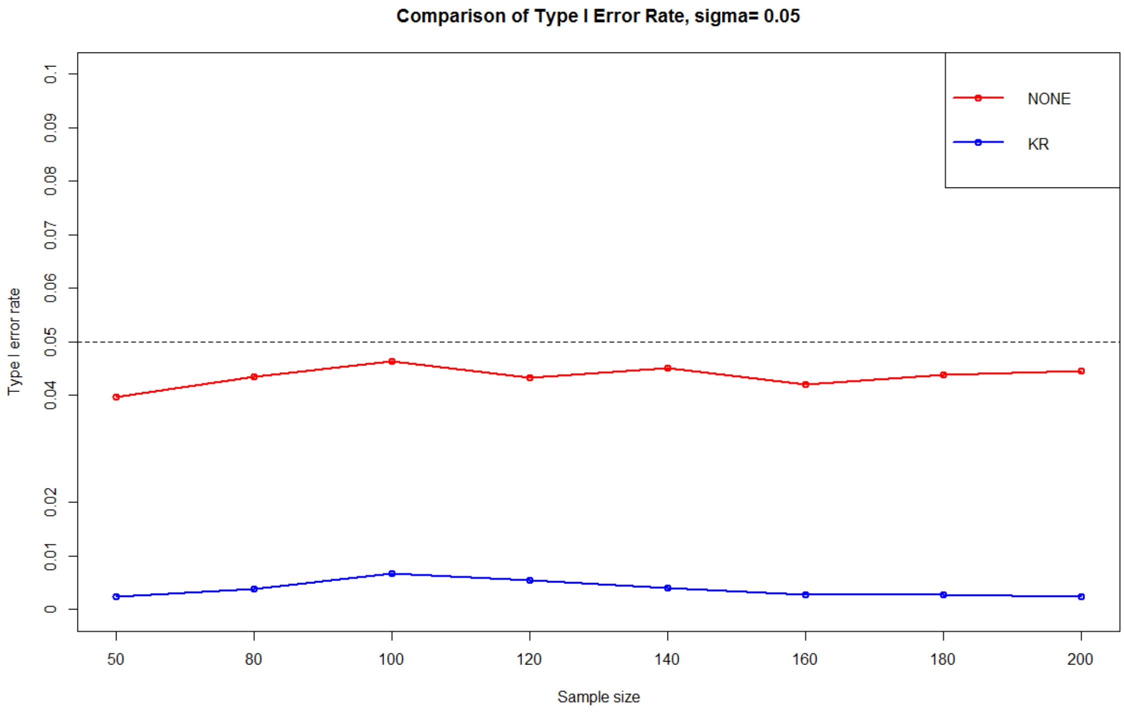

Figure 4.2 The trend line of type I error rates for the two DDFM methods when the variance component equal 0.05, *n* = 50, 80, 100, 120, 140, 160, 180, 200.

4.2 The Discussion and Comparison of the Power for the DDFM = KENWARDROGER and DDFM = NONE

This section we will further properly shows the comparison of the power of these two methods. We performed simulation studies on different parameters settings for the power of the generalized linear mixed model with one random effect.

Figure 4.3 displays the estimated power of the two "DDFM = Methods" options using eight different sample sizes and the variance component $\sigma_{\text{tank}} = 0.1$. As expected, the power increases with larger sample sizes for both "DDFM = Method" options. It is clear that the "DDFM = NONE" method performs better, since it has bigger power under all simulation settings. Especially, when the sample size reach to 140 and above, the power for the "DDFM = NONE" method achieve above 80%. However, the power for the “DDFM = KENWARDROGER” still below 50%, even the sample size is achieved to 200. In addition, each simulation parameters setting combination was replicated 10,000 times and the convergence rates for simulation studies on power are between 99.74% to 99.96%.

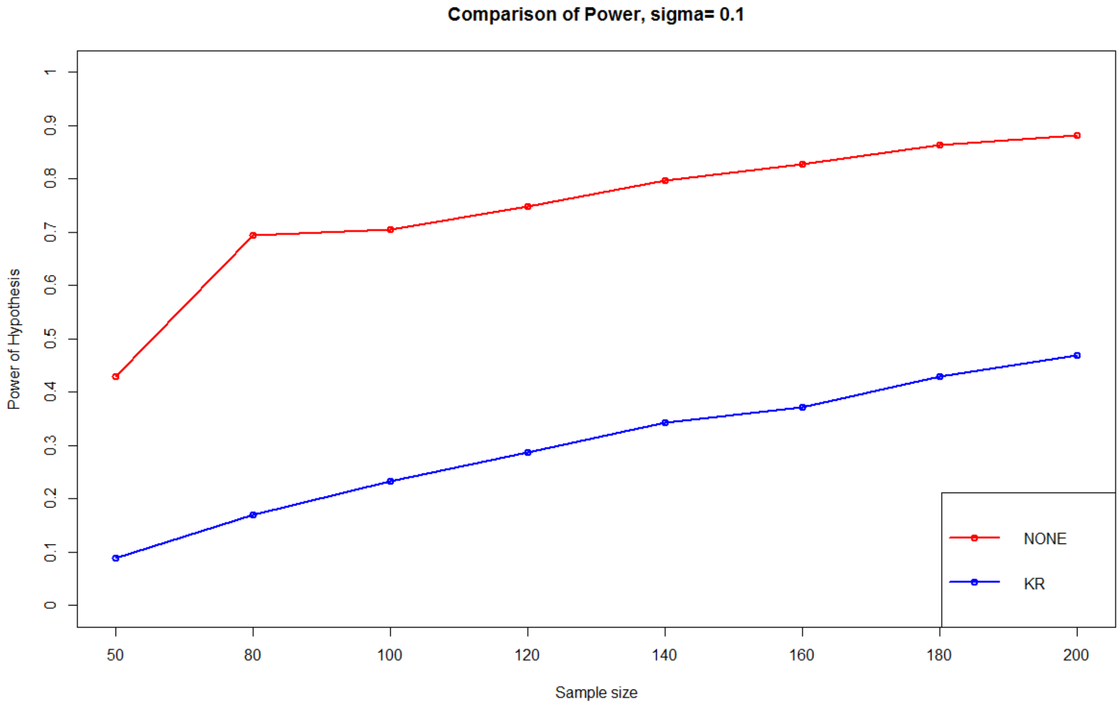

Figure 4.3 The trend line of power for the two DDFM methods when the variance component equal 0.1, *n* = 50, 80, 100, 120, 140, 160, 180, 200.

We can draw the similar conclusions from Figure 4.3. This graphic shows the estimated power of the two “DDFM = Methods” options when the value of variance component $\sigma_{\text{tank}} = 0.05$. It is clear that the “DDFM = KENWARDROGER” option performs worse than the “DDFM = NONE” option, since it has super small power under all simulation settings.

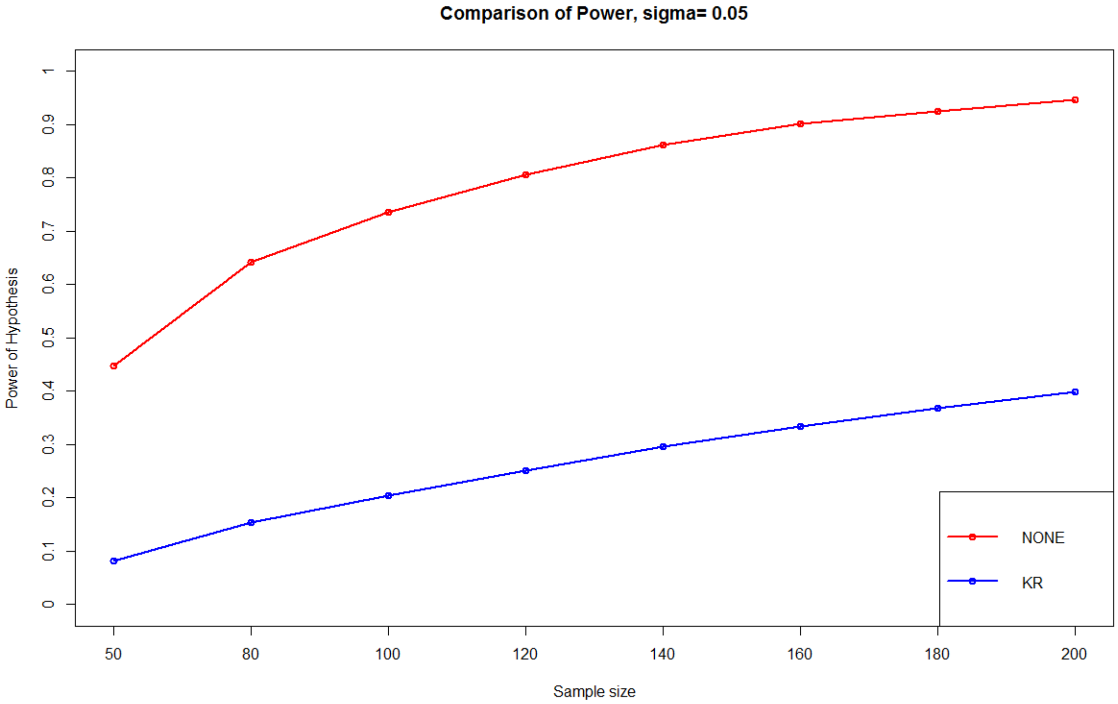

Figure 4.4 The trend line of power for the two DDFM methods when the variance component equal 0.05, *n* = 50, 80, 100, 120, 140, 160, 180, 200.

Chapter 5 - Conclusion

In this report, we investigated two “DDFM = Method” options in the SAS PROC GLIMMIX procedure with respect to type I error rates and power of a two-sided hypothesis test under a generalized linear mixed model containing one random effect. The simulation study was conducted by various sample sizes (n = 50, 80, 100, 120, 140, 160, 180 and 200), and with two different values of the variance component ($\sigma_{\text{tank}} = 0.1$ and) $\sigma_{\text{tank}} = 0.05$, as well as the marginal probability of success $\pi_1 = 0.5$ and $\pi_2 = 0.6$ for the treatment groups and control groups, respectively. The conclusions drawn from the simulation results are as follows.

First, for the generalized linear mixed model with binomial response variable and containing one random effect, the type I error rates of “DDFM = KENWARDROGER” method are much smaller than the type I error rates of “DDFM = NONE” method. It suggests that the “DDFM = KENWARDROGER” method deflates the type I error rates and causes the type I error rate to be more conservative than expected, especially for smaller sample sizes with smaller variance component.

Secondly, the power of hypothesis test based on two “DDFM = Method” options increases with larger sample sizes. As expected, the power of “DDFM = NONE” method seems to perform better than the “DDFM = KENWARDROGER” method, since it had greater power under all simulation settings. We conclude from this that the “DDFM = KENWARDROGER” method shrinks the power of hypothesis test in the generalized linear mixed model under the settings studied.

Lastly, many researchers believe the “DDFM = KENWARDROGER” is suitable for the most mixed models. However, it does not perform well on the generalized linear mixed model studied here and should not be used for these types of models.

Bibliography

1. FDA (2014). NADA 141-423 Freedom of Information Summary, pp. 12-15. http://www.fda.gov/downloads/AnimalVeterinary/Products/ApprovedAnimalDrugProducts/FOIADrugSummaries/UCM408897.pdf (last accessed 10-24-2016).
2. Pinherio, J. C., and D. M. Bates. (2004). Mixed-Effects Models in S and S-PLUS. Statistics and Computing Series, Springer.
3. Fikret Isik. (2011). Generalized Linear Mixed Models-An Introduction for Tree Breeders and Pathologists. Fourth International Workshop on the Genetics of Host-Parasite Interactions in Forestry July 31 – August 5. Eugene, Oregon, USA.
4. SAS/STAT(R) 9.2 User's Guide, Second Edition. Available from: https://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#statug_mixed_sect015.htm (last accessed 10-24-2016).
5. Wang, Z. and Goonewardene, L. A. (2004). The use of MIXED models in the analysis of animal experiments with repeated measures data. Canadian Journal of Animal Science, 84(1), 1-11.
6. Kackar, R. and Harville, D. (1984). Approximations for Standard Errors of Estimators of Fixed and Random Effect in Mixed Linear Models. Journal of the American Staistical Association 79, 853-862.
7. Kenward, M. and Roger, J. (1997). Small Sample Inference for Fixed Effects from Restricted Maximum Likelihood. Biometrics 53, 983-997.
8. Kenward, M. and Roger, J. (2009). An improved approximation to the precision of fixed effects from restricted maximum likelihood. Computational Statistics and Data Analysis 53, 2583-2595.

9. Prasad, N. and Rao, J. (1990). The Estimation of the Mean Squared Error of Small-Area Estimators. Journal of the American Statistical Association 85, 163-171.

10. Cohen, Jacob. (1988). Statistical Power Analysis for the Behavioral Sciences, 2nded. Hillsdale, NJ: L. Erlbaum Associates.

Appendix A – Simulation SAS Code for Type I error Rate

Below is SAS code used to generate simulated data for type I error rate of hypothesis testing.

```
/*********influence on Type I error rate*****************/

/*
Notes: The different parameter setting combination for the different scenarios.
sigma parameter option 0.05 or 0.1
sample size options 50, 80, 100,120,140,160,180,200.
There should be 16 different scenarios in total.
*/

%let sd=0.05; /*sigma */

%let pi=0.5; /*parameter*/

%let n=200;

/*sample size */

/*********generate the binomial data set*****************/

Data fish;

call streaminit (65896);

Do sim =1 to 10000;

          Do trt = 1 to 2;

                    Do tank =1 to 4; Rho=rand("normal",0,&sd);

                              n=&n;

                       Tau =log(&pi/(1-&pi));

                              Mu=0;

                              p=exp (Mu + Tau + Rho)/(1+exp( Mu + Tau +Rho));

                              y=rand ("binomial", p, n);

                              output;

                    End;

          End;

End;

Run;

data  fish1;

          set fish;

run;
```

```
/*********Generalized linear mix model*****************/
/*************DDFM=KENWARDROGER*************************/
*ods select Tests3 LSMeans CovParms;
ods exclude all;
proc glimmix data=fish1;
   by sim;
        class tank trt;
        model y/n=trt/link=logit dist=binomial ddfm=KR COVB;
        random tank(trt);
        covtest/clm;
        lsmeans trt/ilink;
ods output Tests3=T3 LSMeans=LSM CovParms=Cova;
run;

/**********create a table contain p-value$LSmeans$variance component**********/
proc contents data=T3;run;
proc contents data=LSM;run;
proc contents data=Cova;run;
proc sort data=T3(keep=sim ProbF DenDF); by sim;run;
proc sort data=LSM(keep=sim trt Mu StdErrMu);by sim; run;
proc sort data=Cova(keep=sim CovParm Estimate); by sim; run;

proc transpose data=LSM out = LSM1    /*Long-to-Wide table*/
        prefix=Mu;
        var Mu;
        by sim;
run;
proc transpose data=LSM out = LSM2    /*Long-to-Wide table*/
        prefix=StdErrMu;
        var StdErrMu;
        by sim;
run;
data LSM3;
        merge LSM1 LSM2 ;
        by sim;
        drop _NAME_ _LABEL_;
run;

data summary;
        merge T3 LSM3 Cova;
        by sim;
run;
/**********calculate the type1 error rate**********/
ods select all;
data summary2;                    /*for each simulation p value to decide reject or do not reject */
        set summary ;
   reject=0;                      /* This creates a variable called reject in the newdataset if p
value<0.05,with all values 1 */
        if ProbF<0.05 then reject=1;
                label ProbF="P-value"
                    Mu1="Trt1"
              Mu2="Trt2"
              StdErrMu1="SEtrt1"
              StdErrMu2="SEtrt2"
              CovParm="Covariance"
              Estimate="CovEst";
run;
proc print data=summary2(obs=10) label; title'Summary of Simulation
DDFM=KENWARDROGE'; run;

proc means data= summary2 sum;
```

```
        var reject;
        output out=type1 sum=totalreject; run;
data type1;
        set type1;
   alphat=totalreject/_FREQ_;
   label _FREQ_="simnumber";
   run;
   proc print data=type1 label;title'Type1 Error For DDFM=KENWARDROGER';run;
proc means data= summary2;
        var Mu1 Mu2 StdErrMu1 StdErrMu2 Estimate;
output out=MeanEst; title'Summary of Simulation Estimate For
DDFM=KENWARDROGER';Run;
/**********export txt file for Cov.estimate**********/
proc export data=summary2
outfile='C:\Users\Yu Wang\Desktop\fish.txt' DBMS=tab replace;
run;




/*********generalized linear mix model*****************/
/*************DDFM=NONE*************************/
*ods select Tests3 LSMeans CovParms;
ods exclude all;
proc glimmix data=fish1;
   by sim;
        class tank trt;
        model y/n=trt/link=logit dist=binomial ddfm=NONE COVB;
        random tank(trt);
        covtest/clm;
        lsmeans trt/ilink;
```

```
ods output Tests3=T3 LSMeans=LSM CovParms=Cova;
run;
/**********create a table contain p-value$LSmeans$variance component**********/
proc contents data=T3;run;
proc contents data=LSM;run;
proc contents data=Cova;run;
proc sort data=T3(keep=sim ProbF DenDF); by sim;run;
proc sort data=LSM(keep=sim trt Mu StdErrMu);by sim; run;
proc sort data=Cova(keep=sim CovParm Estimate); by sim; run;

proc transpose data=LSM out = LSM1    /*Long-to-Wide table*/
        prefix=Mu;
        var Mu;
        by sim;
run;
proc transpose data=LSM out = LSM2    /*Long-to-Wide table*/
        prefix=StdErrMu;
        var StdErrMu;
        by sim;
run;
data LSM3;
        merge LSM1 LSM2 ;
        by sim;
        drop _NAME_ _LABEL_;
run;
data summary;
        merge T3 LSM3 Cova;
        by sim;
run;
/**********calculate the type1 error**********/
ods select all;
```

```
data summary2;                      /*for each simulation p value to decide reject or do not reject */
        set summary ;
   reject=0;                    /* This creates a variable called reject in the newdataset if p
value<0.05,with all values 1 */
        if ProbF<0.05 then reject=1;
                  label ProbF="P-value"
                      Mu1="Trt1"
                Mu2="Trt2"
                StdErrMu1="SEtrt1"
                StdErrMu2="SEtrt2"
                CovParm="Covariance"
                Estimate="CovEst";
run;
proc print data=summary2(obs=10)label;title'Summary of Simulation For DDFM=NONE';run;

proc means data= summary2 sum;
        var reject;
        output out=type1 sum=totalreject;run;
data type1;
        set type1;
   alphat=totalreject/_FREQ_;
   label _FREQ_="simnumber";
   run;
   proc print data=type1 label;title'Type1 Error For DDFM=NONE';run;
proc means data= summary2;
        var Mu1 Mu2 StdErrMu1 StdErrMu2 Estimate;
output out=MeanEst;title'Summary of Simulation Estimate For DDFM=NONE';Run;
/**********export txt file for Cov.estimate**********/
proc export data=summary2
outfile='C:\Users\Yu Wang\Desktop\fish.txt' DBMS=tab replace;
run;
```

Appendix B – Simulation SAS Code for Power

Below is SAS code used to generate simulated data for power of hypothesis testing.

```
/*********influence on power*****************/

/*
Notes: The different parameter setting combination for the different scenarios.
sigma parameter option 0.05 or 0.1
sample size options 50, 80, 100,120,140,160,180,200.
There should be 16 different scenarios in total.
*/

/*********generate the binomial data set*****************/

%let sd=0.05; /*sigma option 0.1 or 0.05*/

%let n=180;  /*sample size option 50,80,100,120,140,160,180,200*/

Data fish;

    Call streaminit(52256);

        Do sim =1 to 10000;

            Do trt = 1 to 2;

                Do tank =1 to 4; Rho=rand("normal",0,&sd);

                    If trt =1 then pi =0.5; /*treatment group pi=0.5*/

                    else pi=0.6;                /*control group pi=0.6*/

                n=&n;

                Tau =log(pi/(1-pi));

                    Mu=0;

                    p=exp( Mu + Tau + Rho)/(1+exp( Mu + Tau +Rho));

                    y=rand("binomial",p,n);

                    output;

                End;

            End;

        End;

Run;

data  fish1;

    set fish;

run;
```

```
/*********generalized linear mix model*****************/
/*************DDFM=KENWARDROGER*************************/
*ods select Tests3 LSMeans CovParms;
ods exclude all;
proc glimmix data=fish1;
        by sim;
        class tank trt;
        model y/n=trt/link=logit dist=binomial ddfm=KR COVB;
        random tank(trt);
        covtest/clm;
        lsmeans trt/ilink;
        ods output Tests3=T3 LSMeans=LSM CovParms=Cova;
run;

/**********create a table contain p-value$LSmeans$variance component**********/
proc contents data=T3;run;
proc contents data=LSM;run;
proc contents data=Cova;run;
proc sort data=T3(keep=sim ProbF DenDF); by sim;run;
proc sort data=LSM(keep=sim trt Mu StdErrMu);by sim; run;
proc sort data=Cova(keep=sim CovParm Estimate); by sim; run;

proc transpose data=LSM out = LSM1    /*Long-to-Wide table*/
        prefix=Mu;
        var Mu;
        by sim;
run;

proc transpose data=LSM out = LSM2    /*Long-to-Wide table*/
        prefix=StdErrMu;
        var StdErrMu;
```

```
          by sim;
run;

data LSM3;
          merge LSM1 LSM2 ;
          by sim;
          drop _NAME_ _LABEL_;
run;

data summary;
          merge T3 LSM3 Cova;
          by sim;
run;
/**********calculate the Power**********/
ods select all;
data summary2;                     /*for each simulation p value to decide reject or do not reject */
          set summary ;
   reject=0;                       /* This creates a variable called reject in the newdataset if p
value<0.05,with all values 1 */
          if ProbF<0.05 then reject=1;
                    label ProbF="P-value"
                         Mu1="Trt1"
                  Mu2="Trt2"
                  StdErrMu1="SEtrt1"
                  StdErrMu2="SEtrt2"
                  CovParm="Covariance"
                  Estimate="CovEst";
run;
proc print data=summary2(obs=20)label;
title'Summary of Simulation DDFM=KENWARDROGE';run;
```

```
proc means data= summary2 sum;
        var reject;
        output out=power sum=totalreject;run;

data power;
        set power;
   Power=totalreject/_FREQ_;
   label _FREQ_="Simnumber";
   run;

proc print data=power label;title'Power For DDFM=KENWARDROGER';run;
proc means data= summary2;
        var Mu1 Mu2 StdErrMu1 StdErrMu2 Estimate;
output out=MeanEst;
title'Summary of Simulation Estimate For DDFM=KENWARDROGER';Run;
/**********export txt file for Cov.estimate**********/
*proc export data=summary2
*outfile='C:\Users\Yu Wang\Desktop\fish.txt' DBMS=tab replace;
*run;

/*********genernalized linear mix model*****************/
/*************DDFM=NONE*************************/
*ods select Tests3 LSMeans CovParms;
ods exclude all;
proc glimmix data=fish1;
   by sim;
        class tank trt;
        model y/n=trt/link=logit dist=binomial ddfm=NONE COVB;
        random tank(trt);
        covtest/clm;
        lsmeans trt/ilink;
```

```
ods output Tests3=T3 LSMeans=LSM CovParms=Cova;
run;

/**********create a table contain p-value$LSmeans$variance component**********/
proc contents data=T3;run;
proc contents data=LSM;run;
proc contents data=Cova;run;
proc sort data=T3(keep=sim ProbF DenDF); by sim;run;
proc sort data=LSM(keep=sim trt Mu StdErrMu);by sim; run;
proc sort data=Cova(keep=sim CovParm Estimate); by sim; run;

proc transpose data=LSM out = LSM1    /*Long-to-Wide table*/
        prefix=Mu;
        var Mu;
        by sim;
run;

proc transpose data=LSM out = LSM2    /*Long-to-Wide table*/
        prefix=StdErrMu;
        var StdErrMu;
        by sim;
run;

data LSM3;
        merge LSM1 LSM2 ;
        by sim;
        drop _NAME_ _LABEL_;
run;

data summary;
        merge T3 LSM3 Cova;
```

```
            by sim;
run;
/**********calculate the Power**********/
ods select all;
data summary2;                          /*for each simulation p value to decide reject or do not reject */
            set summary ;
    reject=0;                       /* This creates a variable called reject in the newdataset if p
value<0.05,with all values 1 */
            if ProbF<0.05 then reject=1;
                        label ProbF="P-value"
                              Mu1="Trt1"
                      Mu2="Trt2"
                      StdErrMu1="SEtrt1"
                      StdErrMu2="SEtrt2"
                      CovParm="Covariance"
                      Estimate="CovEst";
run;
proc print data=summary2(obs=20)label;
title'Summary of Simulation For DDFM=NONE';run;
proc means data= summary2 sum;
            var reject;
            output out=power sum=totalreject;run;
data power;
            set power;
    Power=totalreject/_FREQ_;
    label _FREQ_="Simnumber";
    run;
    proc print data=power label;title'Power For DDFM=NONE';run;
proc means data= summary2;
            var Mu1 Mu2 StdErrMu1 StdErrMu2 Estimate;
output out=MeanEst;
```

```
title'Summary of Simulation Estimate For DDFM=NONE';Run;
/**********export txt file for Cov.estimate**********/
*proc export data=summary2
*outfile='C:\Users\Yu Wang\Desktop\fish.txt' DBMS=tab replace;
*run;
```

Appendix C – R Code for Figure 4.1-4.2

Below is R code used to draw the trend line plot of Type I error rate.

```
#Plot for type I error
# read txt data in to R
Typeerror0.1<-read.table("C:/Users/Yu Wang/Desktop/Type I
error/typeerror0.1.txt",header=T,na.strings = "NA")
Typeerror0.05<-read.table("C:/Users/Yu Wang/Desktop/Type I
error/typeerror0.05.txt",header=T,na.strings = "NA")
# Create Line Chart for power sigma=0.1
# Calculate range from 0 to max value of cars and trucks
g_range <- range(0, Typeerror0.1$rateKR, Typeerror0.1$ratenone)
plot( Typeerror0.1$ratenone,type='o',ylim=c(0,0.1),col='red',axes=F,lwd=2.5,main="Comparison
of Type I Error Rate, sigma= 0.1",
    xlab="Sample size", ylab="Type I error rate")
# Make Y axis using labels
axis(1, at=1:8,lab=c("50","80","100","120","140","160","180","200"))
lines(Typeerror0.1$ratekr,type='o',col='blue',lwd=2.5)
# add reference line
abline(h = 0.05, col="black", lwd=1, lty=2)
legend("topright", c("NONE","KR"), lwd=c(2.5,2.5),pch=1,col=c("red","blue"),cex=1)
# Make Y axis using labels
lab=c(0,0.01,0.02,0.30,0.04,0.05,0.06,0.07,0.08,0.09,0.1)
axis(2, at=lab, lab=c(0,0.01,0.02,0.03,0.04,0.05,0.06,0.07,0.08,0.09,0.1))
# Create box around plot
box()




# Create Line Chart for power sigma=0.05
# Calculate range from 0 to max value of cars and trucks
g_range <- range(0, Typeerror0.05$rateKR, Typeerror0.05$ratenone)
plot(
Typeerror0.05$ratenone,type='o',ylim=c(0,0.1),col='red',axes=F,lwd=2.5,main="Comparison of
Type I Error Rate, sigma= 0.05",
    xlab="Sample size", ylab="Type I error rate")
# Make Y axis using labels
axis(1, at=1:8,lab=c("50","80","100","120","140","160","180","200"))

lines(Typeerror0.05$ratekr,type='o',col='blue',lwd=2.5)
legend("topright", c("NONE","KR"), lwd=c(2.5,2.5),pch=1,col=c("red","blue"),cex=1)
abline(h = 0.05, col="black", lwd=1, lty=2)

# Make Y axis using labels
lab=c(0,0.01,0.02,0.30,0.04,0.05,0.06,0.07,0.08,0.09,0.1)
axis(2, at=lab, lab=c(0,0.01,0.02,0.03,0.04,0.05,0.06,0.07,0.08,0.09,0.1))
# Create box around plot
box()
```

R Code for Figure 4.3-4.4

Below is R code used to draw the trend line plot of Power.

```
#Plot for power
# read txt data in to R
power0.1<-read.table("C:/Users/Yu Wang/Desktop/power/power0.1.txt",header=T,na.strings = "NA")
power0.05<-read.table("C:/Users/Yu Wang/Desktop/power/power0.05.txt",header=T,na.strings = "NA")
# Create Line Chart for power sigma=0.1
# Calculate range from 0 to max value of cars and trucks
g_range <- range(0, power0.1$KR, power0.1$NONE)
plot(power0.1$NONE,type='o',ylim=c(0:1),col='red',axes=F,lwd=2.5,main="Comparison of Power, sigma= 0.1",
    xlab="Sample size", ylab="Power of Hypothesis")
# Make Y axis using labels
axis(1, at=1:8,lab=c("50","80","100","120","140","160","180","200"))
lines(power0.1$KR,type='o',col='blue',lwd=2.5)
legend("bottomright", c("NONE","KR"), lwd=c(2.5,2.5),pch=1,col=c("red","blue"),cex=1)

# Make Y axis using labels
lab=c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)
axis(2, at=lab, lab=c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1))
# Create box around plot
box()
# Create Line Chart for power sigma=0.05
g_range <- range(0, power0.05$KR, power0.05$NONE)
plot(power0.05$NONE,type='o',ylim=c(0:1),col='red',axes=F,lwd=2.5,main="Comparison of Power, sigma= 0.05",
    xlab="Sample size", ylab="Power of Hypothesis")
# Make Y axis using labels
axis(1, at=1:8,lab=c("50","80","100","120","140","160","180","200"))
```

```
lines(power0.05$KR,type='o',col='blue',lwd=2.5)
legend("bottomright", c("NONE","KR"), lwd=c(2.5,2.5),pch=1,col=c("red","blue"),cex=1)
# Make Y axis using labels
lab=c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)
axis(2, at=lab, lab=c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1))
# Create box around plot
box()
```