Introduction to Measurement Error Models

GEORGE TSALTAS

SUPERVISOR: FOTIOS SIANNIS

May 11, 2023

Contents

Contents 2

1 Measurment error **5**

1.1 Introduction to measurment error 5

1.2 True Score Theory 9

1.3 Measurement Error in regression models with one independent variable 12

1.4 Measurment Error in Multivariate Regression Models 19

1.5 Instrumental Variables 22

1.5.1 Popular Choices of Instruments in Measurment Error Models 23

1.5.2 Example of IV in Multivariate OLS 24

1.6 Measurment Error in Dummy variables 26

2 SIMEX **29**

2.1 SIMEX 29

2.1.1 The SIMEX Method 30

2.2 Likelihood Functions and Estimators 33

2.3 SIMEX when X is Modeled Non-parametrically 35

2.3.1 Standard Error Estimation for the Parametric Part . . 36
2.3.2 Standard Error Estimation for the Non-parametric Part 38
2.4 Simex when X is Modeled Parametrically 39
2.4.1 Standard Error Estimation for the Parametric and Non-Parametric Part 41
2.5 SIMEX in Rstudio 43
2.5.1 SIMEX example in Rstudio 45
2.5.2 Extra example in Rstudio 53
2.6 Significant result for the multiple linear model 59

3 Large US Data Analysis **65**
3.1 Introduction 65
3.2 Data Preview 66

4 Appendix **80**

5 References **100**

Introduction

In statistical practice, variables are often contaminated with measurement error. This may be the case due to bad measurement tools or just because the true variable cannot be measured directly. In the case of discrete variables, measurement error is referred to as misclassification. In the framework of general regression, measurement error or misclassification can lead to serious bias in the estimated parameters.

In most cases the estimated effect of the contaminated variable is attenuated. Among many other methods the simulation and extrapolation (SIMEX) by Cook and Stefanski (1994) has become a useful tool for correcting effect estimates in the presence of additive measurement error. The same basic idea of simulation and extrapolation is transferred to the case of misclassification. The Rstudio package simex provides functions to use the SIMEX method for various kinds of regression objects and to produce graphics and summary statistics for corrected objects.

We will start with the concept of measurement error and how it enters and affects our variables and also ways to avoid it. Then we will talk about the SIMEX method and give the theoretical background of the method as well as some simple and easy examples to better understand the method

and how it works. Finally, we will do a study on databases where there is measurement error in survival data. We will use the SIMEX method to deal with the measurement error using the ready-made simex function that exists in Rstudio. We will also create a function in Rstudio for the SIMEX method which extends the ways and models we can work with.

Chapter 1

Measurment error

1.1 Introduction to measurment error

First, we would like to start with some simple examples so that we can better understand the concept of measurement error. In a research project, people will give different answers to your questions and that is because they are generally different. At the same time the answers we receive are not just a product of peoples genuine differences. Perhaps a good place to start when discussing measurement problems is to differentiate two sources of variation , two reasons that people give the score they give and why people give different scores.

Those two sources of variation are true variation and measurement error. True variation means that a measure is capturing people as they really are and they are giving different scores because they are actually different. Measurement error occurs because there's a problem in the question. It's not

completely capturing the characteristics or all the information that you are trying to get from your subjects. Also not all measurement errors are the same, there are two types, random and systematic error.

Random error is unpredictable, errors that don't necessarily push a study's result in one direction or the other. So when you are conducting a study there are all types of reasons that a respondent might not give an accurate answer for himself or herself. For example, maybe a respondent misread the question or maybe the question asks the respondent to perform mental calculations and they made a mistake. That's error because your measurement isn't reflecting your respondents true characteristics but it's random error, you don't know exactly who's going to commit the error. It's not like all of your respondents are going to commit the same type of error in a way that they bias your overall findings.

Systematic error happens when there is something in the question or the measurement scheme itself that pushes all respondents to answer you erroneously and in the same way. Here is an example of a question that would probably produce systematic error. Let's say we would like to study the maximum speed that a student has driven with his car. We asked the following question: Are you one of those idiots who run more than 120 km/h , yes or no ? Now the problem with this question is obvious, by labeling people who run more than 120 km/h as idiots i have created a disincentive to people to answer yes. You could imagine people who run more than 120 km/h but don't want the label of idiot attached to them and so they answer

no even though the true answer is yes. In that type of situation our studies will probably underestimate the amount of students who run more than 120 km/h because the question was formulated in a way that encouraged a specific answer.

A really common way to describe the measurement error and visualize it, is to give this bullseye graph in Figure 1.1 . Imagine the center of the bulls-

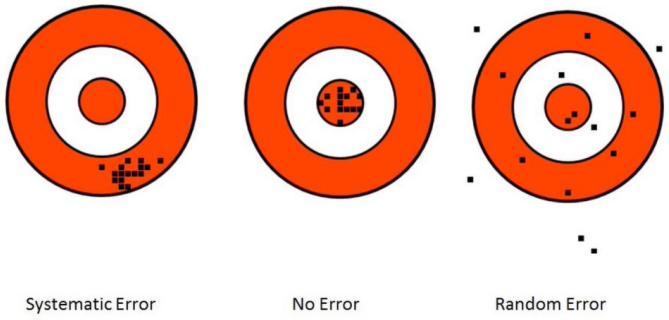

Figure 1.1: Bullseye Graph

eye is the true answer for everybody and the shots are examples of the scores registered when we measure their answers. If there's a lot of random error in my study, answers will be all over the place, maybe someone will accidentally answer A instead of B but we can assume that for every person who accidentally answers one question too low someone else will answer too high, someone will put an B instead of a C. In situations with a lot of random error the worst case scenario is that we get just a lot of noise and we can't really tell from our measurements what the population is really like.

Systematic error is much more dangerous because there is a possibility that we will get what appears to be very highly or very precise answer but the

answer is wrong. Why is that the problem? Well think about how research is used in decision making, a lot of random error is given to a decision maker and the decision maker is probably going to say "i don't know, i can't tell from the data what my best choice is, the answers are all over the place". In contrast, a lot of systematic error could give the impression of highly definitive findings. It looks like your studies are conclusive and your client looks at your results and goes: Well it's quite clear what i have to do based on your results and he or she makes a decision based on your faulty results. A confident wrong finding is worse than a uncertain finding that doesn't tell as much.

Measurement error is an inescapable problem, it plagues all studies, it's an issue that you can't get rid off . At best, you can be aware of its existence and try to keep your eyes open for problems in your research measurement design that will cause these types of errors. You have to use your judgment but there are some basic tips that you might want to follow.

• Make your questions and potential answers as clear as possible. Don't make it hard for people to understand the question. Give a direct question or code it. Those type of pitfalls can create error.

• When you design a survey, always pre test it. That means, bring it to some people, give them the test and then go over the answers with them. Make sure that the answers that you receive are what they really meant to be. Talk to them about whether or not the wording of your question made them feel like they should answer a particular way, or whether or not the

design of your survey caused them to stop paying attention. It is crucial not to disregard an issue, especially when it comes to bad data. When your survey contains a lot of errors, it can negatively impact your analysis.

1.2 True Score Theory

True score theory is a model used to explain how measurement works. It proposes that any measurement we make is made up of two parts: the true ability or level of the person being measured, and random error. Although we can observe the score or measurement that is obtained, we cannot observe the two components that make up the true score theory. Instead, we assume that each measurement is a combination of the true ability and random error. In simpler terms, the theory suggests that any measurement we make is the sum of what a person is actually capable of and any random factors that may have affected the measurement. Consider the following equation:

$$X = T + e,$$

where X is the observed score, T is the true ability and e is the random error.

The true score theory is a useful way to think about measuring things, but it may not always be completely accurate. This theory assumes that any measurement we make is a combination of the true value of what we

are measuring, plus some random error that occurs during the measurement process. However, it's possible that some errors are not random at all, but instead occur in a systematic way that affects all or most of the things we are measuring.

To account for this possibility, we can revise the true score model by breaking down the error component into two parts: random error and systematic error. By doing this, we can better understand and account for the different sources of error in our measurements:

$$e = e_r + e_s,$$

where e_r is the random error and e_s is the systematic error. Here, we'll look at the differences between these two types of errors and try to diagnose their effects on our research.

What is Random Error? Random error is a type of error that occurs due to various random factors that can affect the measurement of a variable across a sample. For example, a person's mood can influence their performance on a particular task, causing some to perform better and others to perform worse. The key characteristic of random error is that it has no consistent effects on the entire sample, and instead, it causes the observed scores to vary randomly. In other words, it can cause some scores to be higher or lower than they should be by chance. However, it's important to note that

random error doesn't have any impact on the average performance of the group as a whole. Therefore, it's often referred to as "noise" in the data, adding variability but not affecting the overall results. This means that if we were to add up all the random errors in a dataset, the total would have to be zero, with an equal number of negative and positive errors canceling each other out. (Figure 1.2).

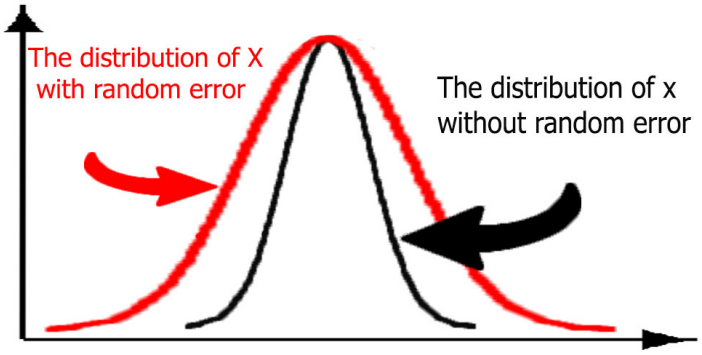

Figure 1.2: The Random Error. Notice that random error doesn't affect the average, only the variability around the average.

What is Systematic Error? Systematic errors are errors that consistently affect the measurement of a variable in a particular sample. These errors can be caused by various factors and tend to produce either consistently positive or negative results. For example, if a classroom is located next to a busy street, the noise from passing traffic may systematically lower the test scores of all students in that classroom. Systematic errors are often considered to introduce bias into the measurement process. In contrast to random errors, which are unpredictable and tend to balance out over time, systematic errors are persistent and can be difficult to identify and correct. (Figure 1.3).

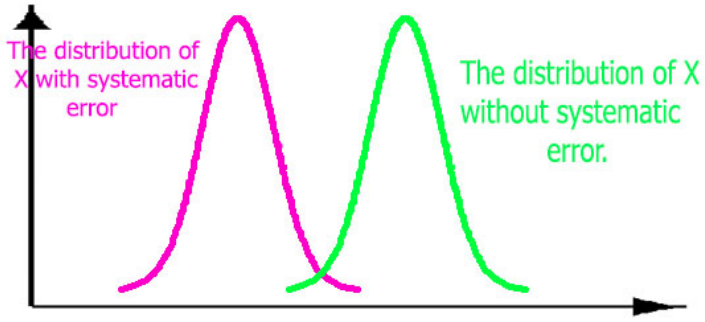

Figure 1.3: The Systematic Error. Notice that systematic error affects the average, we call this a bias.

1.3 Measurement Error in regression models with one independent variable

The equations for this section can be found in "Lecture Notes on Measurement Error,Steve Pischke,Spring 2007". Let's begin with basic regression models that have only one independent variable. For the sake of simplicity, let's assume that both the variable we are trying to predict and the variable we are using to make predictions have mean zero. Our goal is to estimate the relationship between these two variables in the overall population. We consider the following:

$$y = \beta x + \epsilon. \tag{1.1}$$

Regrettably, the values we observe for our variables contain a component of random error that is added to the true value of the variable. This means that the observed values may not be an exact reflection of the true values, but rather an approximation with some degree of uncertainty. Hence the information available is limited to:

$$\tilde{x} = x + u \tag{1.2}$$

$$\tilde{y} = y + v. \tag{1.3}$$

To simplify the analysis, let's assume the following:

$$E(u) = 0 \tag{1.4}$$

$$plim\ \frac{1}{n}(y'u) = 0 \tag{1.5}$$

$$plim\ \frac{1}{n}(x'u) = 0 \tag{1.6}$$

$$plim\ \frac{1}{n}(\epsilon'u) = 0, \tag{1.7}$$

where *plim* is the Probability Limit [1]. The measurement error in the explanatory variable from the above equations can be described as follows:

1. It has mean zero, meaning that on average, the measurement error does

[1]Definition: Convergence in probability. Let θ be a constant, $\epsilon > 0$, and n be the index of the sequence of random variable x_n. If $\lim_{n->\infty}$ Prob[$|x_n - \theta| > \epsilon$] = 0 for any $\epsilon > 0$, we say that x_n converges in probability to θ.

not consistently overestimate or underestimate the true value of the explanatory variable.

2. It is unrelated to the true values of the dependent variable, meaning that the measurement error does not systematically vary with changes in the dependent variable.

3. It is also unrelated to the true values of the independent variable, meaning that the measurement error does not systematically vary with changes in the independent variable.

4. Finally, it is uncorrelated with the error term in the regression equation, meaning that the measurement error does not impact the relationship between the dependent and independent variables as captured by the regression model.

We will also assume ${\sigma^2}_v = 0$ where ${\sigma^2}_v$ is the variance caused by the measurement error in the variable y , i.e. there is only measurement error in x. (If we consider measurement error in the dependent variable y, i.e. let ${\sigma^2}_v>0$ while ${\sigma^2}_u = 0$ and then substitute (1.3) into (1.1) we get:

$$\tilde{y} = \beta x + \epsilon + v.$$

Since v is uncorrelated with x we can estimate β consistently by OLS in this case. Of course, the estimates will be less precise than with perfect data because we have the measurment error included in the data in a way that only icreases the ϵ "$\tilde{y} = \beta x + (\epsilon + v)$").

Substitute (1.2) into (1.1):

$$y = \beta(\tilde{x} - u) + \epsilon = \beta\tilde{x} + (\epsilon - \beta u). \tag{1.8}$$

When there is a measurement error in the independent variable x used in a regression analysis, it can lead to endogeneity bias. This happens because the measurement error in x becomes a part of the error term in the regression equation as we see from the (1.8), which can affect the estimation of the coefficients and lead to biased results. In other words, the error in the measurement of x is not independent of the error in the dependent variable y. Since $\tilde{x}$ and u are positively correlated (from (1.2)) we can see that OLS estimation will lead to a negative bias in $\hat{\beta}$ if the true β is positive and a positive bias if β is negative. To assess the size of the bias, we will find the OLS-estimator for β. First consider that

$$\begin{aligned} \hat{\beta} =& (x'x)^{-1}(x'y) \\ while \quad (x'x)^{-1} =& \frac{1}{(x'x)} = \frac{1}{var(x)} \\ x'y =& \, cov(x, y). \end{aligned}$$

From the above equations

$$\hat{\beta} = \frac{cov(\tilde{x}, y)}{var(\tilde{x})} = \frac{cov(x + u, \beta x + \epsilon)}{var(x + u)}$$

and

$$plim\ \hat{\beta} = plim\ \frac{cov(x + u, \beta x + \epsilon)}{var(x + u)} = plim \frac{\beta\sigma_x^2}{\sigma_x^2 + \sigma_u^2} = \lambda\beta$$

where

$$\lambda \equiv \frac{\sigma_x^2}{\sigma_x^2 + \sigma_u^2}.$$

The quantity λ is a measure of the reliability of the predictor variable in a linear regression model, where $0 < \lambda < 1$. It represents the proportion of the total variance in the predictor variable that is attributable to its true underlying values, as opposed to measurement error or other sources of variation. When λ is less than one, this means that there is some degree of measurement error or other extraneous variation in the predictor variable. As a result, the coefficient $\hat{\beta}$ estimated from the regression analysis will tend to be biased towards zero, or attenuated. In other words, the true effect of the predictor variable on the outcome variable is being underestimated.

Therefore, λ is referred to as the attenuation factor, because it represents the degree to which the estimated effect is attenuated due to measurement error or other sources of variation. A lower value of λ indicates a higher degree of attenuation, and vice versa.

Now that we have defined λ we can find the bias through

$$plim\ \hat{\beta} - \beta = \lambda\beta - \beta = -(1 - \lambda)\beta = -\frac{\sigma_u^2}{\sigma_x^2 + \sigma_u^2}\beta,$$

which again brings out the fact that the bias depends on the sign and size of β. To understand how the estimated standard error changes, it's important

to calculate the residual variance from the regression.

$$\hat{\epsilon} = y - \hat{\beta}\tilde{x} = y - \hat{\beta}(x + u).$$

Add and subtract the true error $\epsilon = y\text{-}\beta x$ from this equation and collect terms

$$\begin{aligned}\hat{\epsilon} &= \epsilon - (y - \beta x) + y - \hat{\beta}x - \hat{\beta}u \\ &= \epsilon + (\beta - \hat{\beta})x - \hat{\beta}u.\end{aligned}$$

The equation above shows that the residual $\hat{\epsilon}$ includes two sources of variation that are not present in the true error ϵ. The first source of variation is due to the bias of the estimator $\hat{\beta}$ towards zero, which means that even as the sample size grows, the difference between the estimated coefficient $\hat{\beta}$ and the true coefficient β will not disappear. The second source of variation is due to the measurement error in the regressor variable x, which introduces additional variance into the regression model. It is important to note that the three variables ϵ, x, and u in the equation are assumed to be uncorrelated. We therefore obtain for the estimated variance of the equation error

$$plim\ \hat{\sigma_\epsilon^2} = \sigma_\epsilon^2 + (1 - \lambda)^2\beta^2\sigma_x^2 + \lambda^2\beta^2\sigma_u^2.$$

For the estimate of the variance of $\sqrt{n}(\hat{\beta}\text{-}\beta)$, call it $\hat{s^2}$ (reminder: The OLS $\hat{\beta}$=(x'x)$^{-1}$(x'y)=$\frac{(x'y)}{(x'x)}$ and the variance is given by Var($\hat{\beta}$)=(x'x)$^{-1}\sigma_\epsilon^2$=$\frac{\sigma_\epsilon^2}{\sigma_x^2}$.

The proof is in the appendix A1, we have

$$
\begin{aligned}
plim\ \hat{s^2} &= plim\ \frac{\hat{\sigma_\epsilon^2}}{\hat{\sigma^2}_{\tilde{x}}} = \frac{\sigma_\epsilon^2 + (1-\lambda)^2\beta^2\sigma_x^2 + \lambda^2\beta^2\sigma_u^2}{\sigma_x^2 + \sigma_u^2} \\
&= \frac{\sigma_x^2}{\sigma_x^2+\sigma_u^2}(\frac{\sigma_\epsilon^2}{\sigma_x^2}) + \frac{\sigma_x^2}{\sigma_x^2+\sigma_u^2}(1-\lambda)^2\beta^2 + \frac{\sigma_u^2}{\sigma_x^2+\sigma_u^2}\lambda^2\beta^2 \\
&= \lambda(\frac{\sigma_\epsilon^2}{\sigma_x^2}) + \lambda(1-\lambda)^2\beta^2 + \lambda^2(1-\lambda)\beta^2 \\
&= \lambda s^2 + \lambda(1-\lambda)\beta^2.
\end{aligned}
$$

The first term in the equation suggests that the estimated variance will be lower than the true variance by a factor proportional to λ, which means the estimation is likely to be biased downwards.

However, the second term indicates that the bias could go either way, as it depends on the value of λ and β. If λ is small and β is large, the bias will be positive, which means the estimated variance will be overestimated. On the other hand, if λ is large and β is small, the bias will be negative, which means the estimated variance will be underestimated.

Therefore, it is not possible to determine the overall direction of the bias in the estimated variance without knowing the values of λ and β.

However, the t-statistic (Definition of the t=statistic: Let $\hat{\beta}$ be an estimator of parameter β in some statistical model. Then a t-statistic for this parameter is any quantity of the form $t = \frac{\hat{\beta}-\beta_0}{s.e.(\hat{\beta})}$, t-statistic with $\beta_0 = 0$ is used to test the significance of corresponding regressor) will be biased downwards (towards

zero). The t-ratio converges to

$$\frac{plim\ t}{\sqrt{n}} = \frac{plim\ \hat{\beta}}{plim\ \sqrt{\hat{s^2}}} = \frac{\lambda\beta}{\sqrt{\lambda s^2 + \lambda(1-\lambda)\beta^2}}$$
$$= \sqrt{\lambda}\frac{\beta}{\sqrt{s^2 + (1-\lambda)\beta^2}}$$

which is smaller than $\beta/\sqrt{s^2}$.

1.4 Measurment Error in Multivariate Regression Models

Multivariate OLS (Ordinary Least Squares) is a statistical method used to estimate the parameters of a linear regression model when there are multiple dependent variables or predictors. Measurement error occurs when the observed values of one or more variables are contaminated with error.
In the context of multivariate OLS with measurement error, the goal is to estimate the true parameters of the model, taking into account the measurement error in the observed variables. This is often referred to as the errors-in-variables (EIV) problem.

One approach is to use a simulation-based method such as the simulation and extrapolation (SIMEX) method (which we are going to analyze this method in Chapter 2) or the bootstrap method. These methods involve simulating data sets with known levels of measurement error, and then using

these simulated data sets to estimate the true parameters of the model.

Another approach to dealing with measurement error in multivariate OLS is to use instrumental variables (IV) regression. This involves finding one or more variables that are correlated with the true value of the independent variable(s) of interest, but not correlated with the measurement error. These instruments can then be used to obtain consistent estimates of the true parameters of the model. We will examine this method in the next section.
The OLS multivariate model with measurement error can be expressed as:

$$Y = X\beta + \epsilon,$$

where Y is an $n \times 1$ vector of dependent variable observations, X is an $n \times k$ matrix of independent variable observations, β is a $k \times 1$ vector of coefficients, and ϵ is an $n \times 1$ vector of error terms.
If we take into account the measurement error in X, we can write:

$$Y = (X + U)\beta + \epsilon,$$

where U is an $n \times k$ matrix of measurement error terms in X.
To estimate β, we need to use the information available from Y and the measured X, $\tilde{X}$. Assuming that the measurement error in X is uncorrelated with the error term ϵ, we can write:

$$\tilde{X} = X + U$$

and we can estimate β using the following equation:

$$\hat{\beta} = (X'X)^{-1}X'Y,$$

where X' is the transpose of X, and $(X'X)^{-1}$ is the inverse of the matrix $X'X$. This is the OLS estimator of β, and it minimizes the sum of squared errors between the predicted values of Y and the actual values of Y.

To adjust for the measurement error in X, we can use the following equation:

$$\hat{\beta}_{adj} = (X'X)^{-1}X'\tilde{Y},$$

where $\tilde{Y} = (X + U)\beta + \epsilon$ is the observed value of Y, using the measured value of X.

The estimator $\hat{\beta}_{adj}$ is generally not an unbiased estimator. The reason for this is that the measurement error in X causes attenuation bias, which biases the OLS estimator towards zero. The amount of bias depends on the magnitude of the measurement error and the correlation between X and U.

In the presence of measurement error, the OLS estimator is biased towards the regression coefficient of the measured variables, which is typically less than the true regression coefficient. The adjusted estimator $\hat{\beta}_{adj}$ reduces but does not completely eliminate this bias.

However, under certain assumptions, such as the classical measurement error model where the measurement error is uncorrelated with the true value of the variable, and the measurement error has a mean of zero and a constant variance, the adjusted estimator can be consistent, meaning that it converges

to the true value of β as the sample size increases. In practice, it is important to carefully consider the sources of measurement error and the assumptions underlying the model when interpreting the results of an analysis.

1.5 Instrumental Variables

Measurement error can lead to biased estimates in statistical analyses. One approach to address this issue is to use instrumental variables (IV) to estimate consistent estimators.

IVs are variables that are correlated the true value of the variable of interest with the measurement error but not with the measurement error. The idea is that the IV can be used to correct for the measurement error and obtain a consistent estimator.

There are several methods for constructing IVs in the context of measurement error. One approach is to use external validation data that is independent of the data used for the analysis. For example, if a researcher is interested in estimating the effect of a drug on blood pressure, they might use an IV that is correlated with the true blood pressure but not with the measurement error in the blood pressure data.

Another approach is to use a natural experiment that generates variation in the measurement error. For example, if a researcher is interested in esti-

mating the effect of education on income but there is measurement error in the education variable, they might use an IV that is correlated with the true education but not with the measurement error. This could be a policy change that affects the availability of education, such as a change in the compulsory schooling age.

Overall, the use of IVs can help to address the issue of measurement error and obtain more accurate and reliable estimates in statistical analyses. Lets consider an instrument z correlated with x but uncorrelated with u, which will identify the true coefficient since:

$$\hat{\beta}_{IV} = \frac{cov(y,z)}{cov(\tilde{x},z)} = \frac{cov(\beta x + \epsilon, z)}{cov(x+u,z)}$$
$$plim\ \hat{\beta}_{IV} = \frac{\beta\sigma_{xz}}{\sigma_{xz}} = \beta.$$

1.5.1 Popular Choices of Instruments in Measurment Error Models

Wald's method, Durbin's method, and Bartlett's method are three different approaches to constructing instrumental variables (IV) estimators.

Wald's method is a two-stage least squares (2SLS) estimator that uses a single instrumental variable for each endogenous variable in the regression model. The first stage involves regressing the endogenous variables on the instrumental variables to obtain the predicted values of the endogenous variables. In the second stage, the predicted values of the endogenous variables

are used as regressors in the main regression model. Wald's method is named after the statistician Abraham Wald.

Durbin's method is a generalization of Wald's method that allows for multiple instrumental variables for each endogenous variable. This method involves creating a set of instrumental variables that are uncorrelated with each other and using them in a multiple regression model to obtain the predicted values of the endogenous variables. Durbin's method is named after the statistician James Durbin.

Bartlett's method is another generalization of Wald's method that allows for multiple instrumental variables for each endogenous variable. However, it differs from Durbin's method in that it involves constructing a matrix of instrumental variables that is optimized to minimize the variance of the resulting IV estimator. Bartlett's method is named after the statistician Maurice Bartlett.

All three methods have their advantages and disadvantages, and the choice of which method to use depends on the specific research question and the characteristics of the data.

1.5.2 Example of IV in Multivariate OLS

Here is a simple example where we use IV to address endogeneity and measurement error in a multivariate OLS model:

Suppose we are interested in estimating the effect of education (E) and experience (X) on earnings (Y), but we suspect that education and experience are endogenous and affected by measurement error. To address this, we use an instrumental variable approach.

Let's assume that years of schooling of the father (F) is a valid instrument for education (E), meaning that F is correlated with E but uncorrelated with the error terms, including the measurement error in E. We also assume that the father's years of schooling is not directly related to the error term in earnings (Y) or the measurement error in experience (X).

Then, we can estimate the following IV regression model:

$$Y = \beta_0 + \beta_1 E + \beta_2 X + \epsilon$$

where Y is the dependent variable, E and X are the independent variables, β_0, β_1, and β_2 are the coefficients of interest, and ϵ is the error term. To estimate the coefficients, we use the following two-stage least squares (2SLS) estimator:

We assume that years of schooling of the father (F) is a valid instrumental variable for education (E), and we define the first-stage regression model:

$$E = \delta_0 + \delta_1 F + u,$$

where δ_0 and δ_1 are the coefficients of the first-stage regression, and u is the error term.

We can then estimate the first-stage regression using the following:

$$\hat{E} = \alpha_0 + \alpha_1 F,$$

where $\hat{E}$ is the predicted value of education from the first stage regression, and α_0 and α_1 are the estimated coefficients. Finally, we can estimate the second-stage regression using the predicted value of education as an instrument:

$$Y = \beta_0 + \beta_1 \hat{E} + \beta_2 X + \epsilon,$$

where $\hat{E}$ is the predicted value of education from the first-stage regression. This can be estimated using the following :

$$\hat{Y} = \gamma_0 + \gamma_1 \hat{E} + \gamma_2 X,$$

where γ_0, γ_1, and γ_2 are the estimated coefficients. This 2SLS estimator provides consistent estimates of the coefficients of interest, even in the presence of measurement error and endogeneity.

1.6 Measurment Error in Dummy variables

In OLS (ordinary least squares) regression, dummy variables are used to represent categorical variables in a regression equation. Dummy variables take on a value of 1 or 0 depending on whether the observation falls into a particular category or not.

When there is measurement error present in the data, the use of dummy vari-

ables can be affected. If the measurement error is random, meaning that it is not systematically related to any particular variable or group of variables, then the use of dummy variables in OLS regression should not be affected. The measurement error will simply add noise to the data, which may reduce the precision of the estimated coefficients but should not bias them.

However, if the measurement error is systematic, meaning that it is related to a particular variable or group of variables, then the use of dummy variables in OLS regression can be biased. For example, if the measurement error is larger for one category of a categorical variable compared to others, then the estimated coefficient for that category may be biased towards zero.

In order to address this issue, one approach is to use instrumental variables (IV) regression. Overall, when using dummy variables in OLS regression with measurement error, it is important to carefully examine the sources and nature of the measurement error to ensure that the estimated coefficients are not biased. If bias is suspected, alternative methods such as IV regression may need to be used.

For example, we want to study the effect of gender on income using OLS regression. However, we have reason to suspect that there may be measurement error in the gender variable due to self-reporting issues. To account for this, we will use an instrumental variables approach.
We will use the education level of the individual as an instrument for gender. Education level is likely correlated with gender, but is less likely to

be affected by measurement error. We first estimate the following equation, where the instrument (Education) is regressed on the suspect variable (Male):

$$Male = \alpha_0 + \alpha_1 Education + u$$

We can then use the predicted values of Male from this equation as our instrument in the OLS regression equation:

$$Income = \beta_0 + \beta_1 \widehat{Male} + \beta_2 Education + \beta_3 Experience + \epsilon$$

Here, $\widehat{Male}$ is the predicted value of Male from the first equation, β_0 is the intercept term, β_1 is the coefficient for the predicted value of Male, β_2 is the coefficient for Education, β_3 is the coefficient for Experience, and ϵ is the error term.

The estimated coefficient for the predicted value of Male (β_1) represents the difference in average income between males and females, controlling for education level and years of work experience, while accounting for the presence of measurement error in the gender variable.

Chapter 2

SIMEX

2.1 SIMEX

SIMEX (Simulation and Extrapolation method) is a statistical technique used to estimate the bias of an estimator in the presence of measurement error. The method was first proposed by Cook and Stefanski in 1994, in their paper "Simulation-extrapolation estimation in parametric measurement error models."

The SIMEX method works by simulating new datasets with different levels of measurement error and then extrapolating the results to estimate the bias of the estimator. This allows researchers to better understand the impact of measurement error on their results and to adjust their analyses accordingly.

Since its introduction, the SIMEX method has been widely used in various fields, including epidemiology, environmental science, and biostatistics.

It has been used to evaluate the performance of diagnostic tests, estimate the effects of exposure to environmental toxins, and evaluate the accuracy of measurement devices.

One of the main advantages of the SIMEX method is that it can provide more accurate estimates of bias than other methods, such as regression calibration, when the amount of measurement error is large or the relationship between the measurement error and the true value is complex. However, the SIMEX method also has some limitations, including the need for a large number of simulations and the potential for bias in the extrapolation process.

Overall, the SIMEX method has become a valuable tool for researchers seeking to improve the accuracy of their results in the presence of measurement error.

2.1.1 The SIMEX Method

The SIMEX (Simulation and Extrapolation) method is used when we have a predictor variable X that is not directly observable, but we observe a related variable W = X + U, where U = Normal(0, Σ_u), is a measurement error with a known covariance matrix Σ_u. Some components of U may equal zero indicating no measurement error in that component. Consider a set of values $0 = \lambda_1 < \lambda_2 < \cdots < \lambda_J$. In the SIMEX method, we take the following steps:

1. Simulation: We simulate additional measurement errors for W by adding independent errors with covariance matrix $\lambda_j \Sigma_u$ for each level of contamination j. This results in a set of contaminated data sets with increasing levels of measurement error. For the j-th data set, the total measurement error variance is $\Sigma_u + \lambda_j \Sigma_u = (1 + \lambda_j)\ \Sigma_u$.

2. Estimation: We estimate X from each contaminated data set using our favorite method (more details in the next page), assuming that there is no measurement error.

3. Averaging: We repeat steps 1 and 2 a large number B times and average the estimates obtained from each contaminated data set for each level of contamination λ_j. This gives us an average estimate for each level of contamination.

4. Extrapolation: We fit an extrapolation function to the averaged estimates (λ_j's) using a regression technique such as polynomial least squares. The extrapolation function estimates the relationship between the true X and the contaminated W data sets as the level of contamination approaches zero.

5.SIMEX estimate: We extrapolate to the ideal case of no measurement error ($\lambda = -1$), which gives us the SIMEX estimate of X.

In summary, the SIMEX method is a technique to estimate the value of an unobservable predictor variable X in the presence of measurement error in a related variable W. It involves simulating additional measurement errors for W, estimating X from each contaminated data set, averaging the estimates over many simulations, fitting an extrapolation function to the averaged estimates, and extrapolating to the ideal case of no measurement error to obtain

the SIMEX estimate of X.

The favorite method in each case:

In parametric problems, the commonly used approach for estimating parameters is to solve an estimating function. However, due to measurement errors in the explanatory variables, the underlying model may not be accurately specified, which can result in biased estimates. Fortunately, this is a well-known issue and does not pose a significant problem since estimating function methods are still effective even when the model is misspecified.

Likewise, in nonparametric regression, the preferred method for estimating the relationship between variables is to solve a local estimating equation. Again, this approach may be affected by model misspecification, but the general principles remain the same.

In both cases, the goal is to obtain estimates that are as accurate as possible given the available data, even if the model is not precisely specified. Therefore, it is important to use appropriate techniques and understand the limitations of the chosen approach to ensure that the resulting estimates are reliable.

2.2 Likelihood Functions and Estimators

In the SIMEX method, the likelihood function is used to estimate the model parameters and to simulate the true values of the predictor variable. The likelihood function is defined as the joint probability density function of the observed data, given the model parameters.

Let us consider a regression model with a predictor variable X, response variable Y, and measurement error in X, where we observe the related variable W = X + U, and U is a random error term with a mean of zero and covariance matrix Σ_u. The likelihood function for this model can be expressed as:

$$L(\beta, \sigma^2, \Sigma_u | W) = \prod_{i=1}^{n} f(y_i | w_i, \beta, \sigma^2) g(w_i | x_i, \Sigma_u),$$

where β represents the regression coefficients, σ^2 represents the error variance, and $f(y_i|w_i, \beta, \sigma^2)$ and $g(w_i|x_i, \Sigma_u)$ represent the conditional probability distributions of Y given W and of W given X and Σ_u, respectively.

The first term of the likelihood function represents the conditional probability of observing the response variable Y, given the observed values of W and the model parameters β and σ^2. This is typically assumed to be a normal distribution, such that:

$$f(y_i | w_i, \beta, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \beta^T w_i)^2}{2\sigma^2} \right).$$

The second term of the likelihood function represents the conditional proba-

bility of observing the related variable W, given the true values of X and the measurement error covariance matrix Σ_u. This is typically assumed to be a normal distribution, such that:

$$g(w_i|x_i, \Sigma_u) = \frac{1}{(2\pi)^{k/2}|\Sigma_u|^{1/2}} \exp\left(-\frac{1}{2}(w_i - x_i)^T \Sigma_u^{-1}(w_i - x_i)\right),$$

where k is the number of non-zero elements in the error covariance matrix Σ_u. Using the likelihood function, we can estimate the model parameters β, σ^2, and Σ_u by maximizing the likelihood function. This can be done using numerical optimization methods such as maximum likelihood estimation (MLE) or restricted maximum likelihood estimation (REML).

Once the model parameters have been estimated, we can simulate the true values of the predictor variable X by adding a random error term U, drawn from a normal distribution with mean zero and covariance matrix Σ_u, to the observed values of W. We can repeat this process many times to generate multiple datasets with simulated values of X.

Finally, we can re-estimate the model using the simulated values of X and the original values of W, and calculate the extrapolated effect of X by comparing the model coefficients from the original model to the model coefficients from the re-estimated model using the simulated values of X. This extrapolated effect provides an estimate of the true effect of X, adjusted for the measurement error in W.

2.3 SIMEX when X is Modeled Non-parametrically

In the SIMEX method, we can also use a non-parametric model to estimate the relationship between the predictor variable X and the response variable Y when X is not directly observable but is measured with error. In this case, we can use a smoothing technique, such as kernel smoothing or spline regression, to estimate the non-parametric relationship between X and Y.
Let us consider a non-parametric regression model, where the response variable Y is related to the true but unobserved predictor variable X by a non-parametric function $m(\cdot)$, such that $Y = m(X) + \epsilon$, where ϵ represents the random error term with mean zero and variance σ^2.
We observe a related variable W = X + U, where U is a random error term with mean zero and covariance matrix Σ_u. Similar to the parametric case, we assume that the error term U is normally distributed with mean zero and covariance matrix Σ_u.

To estimate the non-parametric relationship between X and Y, we can use a smoothing technique, such as kernel smoothing or spline regression, to estimate the conditional mean function $m(\cdot)$ given the observed values of W. The estimated function $\hat{m}(\cdot)$ can be obtained by minimizing the sum of squared errors:

$$\hat{m}(\cdot) = \arg\min_{m} \sum_{i=1}^{n} (Y_i - m(W_i))^2.$$

Once the non-parametric model has been estimated, we can simulate the true

values of the predictor variable X by adding a random error term U, drawn from a normal distribution with mean zero and covariance matrix Σ_u, to the observed values of W. We can repeat this process many times to generate multiple datasets with simulated values of X.

Finally, we can re-estimate the non-parametric model using the simulated values of X and the original values of W, and calculate the extrapolated effect of X by comparing the estimated function $\hat{m}(\cdot)$ from the original model to the estimated function $\tilde{m}(\cdot)$ from the re-estimated model using the simulated values of X. This extrapolated effect provides an estimate of the true effect of X, adjusted for the measurement error in W.

2.3.1 Standard Error Estimation for the Parametric Part

In the SIMEX method with non-parametric modeling of X, the standard error estimation for the parametric part can be challenging due to the complexity of the non-parametric model. However, one approach to estimate the standard error for the parametric part is the bootstrap method.

In the bootstrap method, we repeatedly resample the data and estimate the model to obtain multiple estimates of the parameter of interest. We can then calculate the standard error of the parameter estimate as the standard deviation of the bootstrapped estimates.

Specifically, to estimate the standard error of the parametric part in the SIMEX method with non-parametric modeling of X, we can perform the following steps:

1. Obtain the estimate of the non-parametric function $\hat{m}(\cdot)$ using the observed values of W.

2. Simulate the true values of X using the observed values of W and a random error term U drawn from a normal distribution with mean zero and covariance matrix Σ_u. We can repeat this process many times to generate multiple datasets with simulated values of X.

3. For each simulated dataset, re-estimate the non-parametric function $\tilde{m}(\cdot)$ using the simulated values of X and the original values of W.

4. Calculate the extrapolated effect of X as the difference between $\hat{m}(\cdot)$ and $\tilde{m}(\cdot)$.

5. Bootstrap the extrapolated effect by repeatedly resampling the observed data and simulating new datasets with simulated values of X. For each bootstrapped dataset, repeat steps 3 and 4 to obtain a bootstrapped estimate of the extrapolated effect.

6. Calculate the standard error of the extrapolated effect as the standard deviation of the bootstrapped estimates.

This bootstrap approach can provide an estimate of the standard error for the parametric part of the SIMEX method with non-parametric modeling of X. However, it can be computationally intensive, especially when dealing with large datasets or complex non-parametric models.

2.3.2 Standard Error Estimation for the Non-parametric Part

In the SIMEX method with non-parametric modeling of X, the standard error estimation for the non-parametric part can also be challenging due to the complexity of the non-parametric model. However, one common approach to estimate the standard error for the non-parametric part is the bootstrap-t method.

The bootstrap-t method is a variant of the bootstrap method that takes into account the variability in the estimate of the non-parametric function $\hat{m}(\cdot)$. Specifically, the bootstrap-t method involves the following steps:

1. Obtain the estimate of the non-parametric function $\hat{m}(\cdot)$ using the observed values of W.

2. Simulate the true values of X using the observed values of W and a random error term U drawn from a normal distribution with mean zero and covariance matrix Σ_u. We can repeat this process many times to generate multiple datasets with simulated values of X.

3. For each simulated dataset, re-estimate the non-parametric function $\tilde{m}(\cdot)$ using the simulated values of X and the original values of W.

4. Calculate the extrapolated effect of X as the difference between $\hat{m}(\cdot)$ and $\tilde{m}(\cdot)$.

5. Bootstrap the distribution of the extrapolated effect by repeatedly resampling the observed data and simulating new datasets with simulated values of X. For each bootstrapped dataset, repeat steps 3 and 4 to obtain a boot-

strapped estimate of the extrapolated effect.

6. Calculate the standard deviation of the bootstrapped estimates from step 5 as an estimate of the standard error of the extrapolated effect.

7. Use the studentized bootstrap-t method to adjust the standard error estimate for bias due to the estimation of the non-parametric function. Specifically, we divide the bootstrapped standard deviation from step 6 by a t-statistic that takes into account the degrees of freedom of the non-parametric estimate.

The bootstrap-t method can provide an estimate of the standard error for the non-parametric part of the SIMEX method with non-parametric modeling of X. However, like the bootstrap method for the parametric part, it can be computationally intensive, especially when dealing with large datasets or complex non-parametric models.

2.4 Simex when X is Modeled Parametrically

When X is modeled parametrically, the SIMEX method can be represented by the following equations:

1. Estimate the parametric model of interest using the observed values of W:

$$Y = \beta_0 + \beta_1 X + \beta_2 Z_2 + \cdots + \beta_k Z_k + \epsilon,$$

where Y is the outcome variable, X is the predictor variable with measurement error, $Z_2, \ldots, Z_k$ are other covariates, and ϵ is the error term. We esti-

mate the coefficients $\beta_0, \beta_1, \ldots, \beta_k$ using the observed values of $W = X + U$, where $U \sim N(0, \Sigma_u)$ is the measurement error with a known covariance matrix Σ_u.

2. Extrapolate the effect of X to the true values of X using the SIMEX method:

$$X^* = X + \gamma U,$$

where X^* is the extrapolated value of X, X is the observed value of X, $U \sim N(0, \Sigma_u)$ is the measurement error, and γ is a correction factor given by:

$$\gamma = \frac{\partial E(X|W)}{\partial X},$$

where $E(X|W)$ is the conditional expectation of the true value of X given the observed value of W.

3. Estimate the parametric model using the simulated values of X:

$$Y = \beta_0^* + \beta_1^* X^* + \beta_2^* Z_2 + \cdots + \beta_k^* Z_k + \epsilon,$$

where $\beta_0^*, \beta_1^*, \ldots, \beta_k^*$ are the estimated coefficients from the model fitted to the simulated values of X and the observed values of $Z_2, \ldots, Z_k$, and ϵ is the error term.

4. Calculate the extrapolated effect of X as the difference between the estimated parameters from the model fitted to the observed values of W and the average of the estimated parameters from the models fitted to the simulated datasets:

$$\hat{\tau} = \frac{1}{B}\sum_{b=1}^{B}(\beta_1^{*(b)} - \beta_1),$$

where $\beta_1^{(b)}$ is the estimated coefficient for X^* in the b-th simulated dataset, β_1 is the estimated coefficient for X in the original model, and B is the number of simulated datasets.

5. Bootstrap the extrapolated effect to estimate the standard error:

• Resample the observed data with replacement to create a new bootstrap sample.

• Repeat steps 2-4 using the bootstrap sample to estimate a bootstrapped value of $\hat{\tau}$.

• Repeat the above step many times to obtain a distribution of bootstrapped values of $\hat{\tau}$.

• Calculate the standard error of the extrapolated effect as the standard deviation of the bootstrapped values of $\hat{\tau}$.

This approach can provide a way to correct for measurement error in X and estimate the true effect of X on the outcome variable when X is modeled parametrically.

2.4.1 Standard Error Estimation for the Parametric and Non-Parametric Part

When X is modeled parametrically, the standard error estimation for the parametric part can be obtained by the following steps:

1. Simulate a new error vector U* using the estimated covariance matrix $\hat{\Sigma}_u$.

2. Generate a new predictor variable X* by adding the simulated error vector U* to the observed variable W:

$$X^* = W + U^*.$$

3. Fit a parametric model to the simulated data (X*, Y).

4. Repeat steps 1-3 a large number of times (e.g., 1000 times) to obtain multiple parametric estimates.

5.Calculate the mean and standard deviation of the parametric estimates:

$$\overline{\beta} = \frac{1}{M}\sum_{m=1}^{M}\beta_m$$
$$s(\beta) = \frac{1}{M-1}\sum_{m=1}^{M}(\beta_m - \overline{\beta})^2,$$

where β_m is the parameter estimate obtained from the m-th simulated dataset, M is the total number of simulations, and $s(\beta)$ is the estimated standard error of the parametric estimate.

Note that the standard error estimation for the nonparametric part is obtained using a similar procedure, but instead of fitting a parametric model in step 3, a nonparametric model is fitted to the simulated data and the step

5 is given by:

$$\overline{f}(x) = \frac{1}{M}\sum_{m=1}^{M} f_m(x)$$
$$s(x) = \frac{1}{M-1}\sum_{m=1}^{M}(f_m(x) - \overline{f}(x))^2,$$

where $f_m(x)$ is the nonparametric estimate obtained from the m-th simulated dataset.

2.5 SIMEX in Rstudio

Although the SIMEX method may seem easy, some people may have difficulty applying its steps to a simple regression problem. Fortunately, RStudio has the method pre-installed. By typing "?simex" in RStudio, a help window opens, which explains that the SIMEX method is used for dealing with measurement error. In the window that opens, we can see a figure as shown below: Figure2.1.

As we scroll down the help window we see that the basic syntax for using simex is as follows:

$$simex(formula, data, error.sd, num.simulations, seed)$$

where:

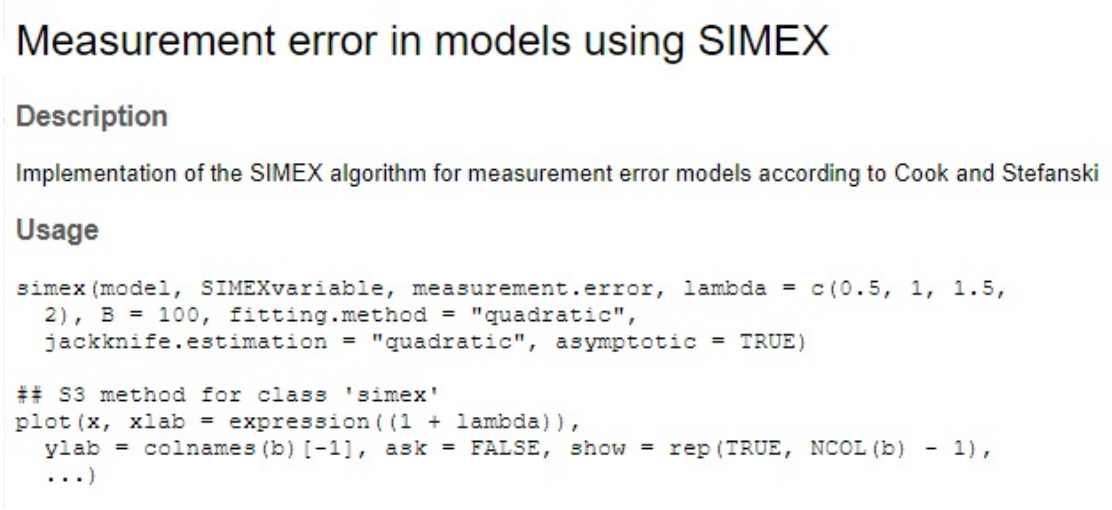

Figure 2.1: The SIMEX in Rstudio

• "formula" is a formula specifying the regression model

• "data" is the data frame containing the variables in the formula

• "error.sd" is the standard deviation of the measurement error

• "num.simulations" is the number of simulations to perform

• "seed" is an optional argument that can be used to set the random seed for reproducibility.

The simex() function returns a list object with the following components:

• "coefficients": a matrix containing the SIMEX-corrected coefficient estimates

• "std.errors": a matrix containing the standard errors of the SIMEX-corrected coefficient estimates

• "bias": a matrix containing the estimated biases of the coefficient estimates

• "conf.int": a matrix containing the confidence intervals for the SIMEX-corrected coefficient estimates.

If we continue scrolling down, we will find details about the function simex as well as generic methods that can be used, such as:

• "plot": Plot the simulation and extrapolation step
• "predict": Predict using simex correction
• "print": Print simex nicely
• "print": Print summary nicely
• "refit": Refits the model with a different extrapolation function
• summary: Summary of simulation and extrapolation

The last thing we see in the help window is some examples that are available by default in Rstudio. These examples can help us better understand the SIMEX method. In Rstudio, we can also use the SIMEX method for discrete data with misclassification by using the function "mcsimex" in a similar way to "simex".

2.5.1 SIMEX example in Rstudio

The following example demonstrates how the SIMEX method can be used to correct for measurement errors in Ordinary Least Squares (OLS) models in Rstudio. In order to run the SIMEX in our computer we need to install some packages in the Rstudio. We start by installing the packages

"simex" , "pacman" and "tibble" by easily typing install.packages("simex"), install.packages("pacman"), install.packages("tibble"). Then we call the libraries pacman, tibble and after that we type $p\_load(kirkegaard, simex, rms)$ $options(digits = 3)$, as we see in Figure 2.2.

In order to test the accuracy of models, we introduce different levels of

```
install.packages("simex")
install.packages("pacman")
install.packages("tibble")
library(pacman)
library(tibble)
p_load(kirkegaard, simex, rms)
options(digits = 3)
```

Figure 2.2: Our Rstudio input

errors to the actual measurements and analyze whether we can obtain the correct results by incorporating the degree of measurement error. Specifically, we simulate models with varying measurement errors added to the true variables and see if we can get the right results back if we plug in the amount of measurement error.

In this simulation study, we have considered three different models to assess the relationship between two uncorrelated predictors and an outcome variable. Firstly, we have a "true model" which is based on the unobserved predictors and serves as a benchmark for evaluating the performance of the other two models. Secondly, we have a "naive model" which only takes into account the observed variables and ignores any measurement reliability is-

sues. Finally, we have a "SIMEX model" that incorporates the reliability of the measurements to improve the estimation of the relationship between the predictors and the outcome, aiming to recover the results from the true model.

We have set the sample size to 5000 and have considered two predictors with constant slopes of 1, which are uncorrelated. We have also varied the reliability of the measurements to be 1.2 and 0.8. It is important to note that the true reliability of the measurements is known and used in the SIMEX model to improve the estimation accuracy. Overall, the simulation study aims to assess the performance of the different models under these settings and provide insights into the importance of accounting for measurement reliability in regression analysis.

We will generate a set of 5000 observations where a_1 and a_2 are values from the standard normal distribution and a_1x and a_2x are the corresponding obserations with the reliability at 1.2 and 0.8 respectively.
Input :

$sim1\_data = tibble(a1 = rnorm(5000),$
$a2 = rnorm(5000),$
$a1x = a1 + rnorm(5000) * 1.2,$
$a2x = a2 + rnorm(5000) * 0.8,$
$y = a1 + a2$
)

We now run the linear model with the variables $a1$ and $a2$

Fitment:

$(sim1\_truefit = lm(y \sim a1 + a2, data = sim1\_data))$

And Rstudio gives us the following results.

Output:

$Call$:

$lm(formula = y \sim a1 + a2, data = sim1\_data)$

$Coefficients$:

$(Intercept) \quad a1 \quad a2$

$7.02e-18 \;\; 1.00e+00 \;\; 1.00e+00$

As we can see from the output the constant term is equal to 0 and the slopes $a1$ and $a2$ are equal to 1 . We now do the same procedure for the naive model with the variables $a1x$ and $a2x$.

Input:

$(sim1\_naive\_fit = lm(y \sim a1x + a2x, data = sim1\_data, x = T))$

And Rstudio gives us the following results.

Output:

$Call$:

$lm(formula = y \sim a1x + a2x, data = sim1\_data, x = T)$

$Coefficients$:

$(Intercept)$ $a1x$ $a2x$

-0.01796 0.40419 0.62719

As we can see from the output the constant term is equal to -0.01796 and the slopes $a1x$ and $a2x$ are equal to 0.40419 and 0.62719 respectively.
We notice that the values of the constant terms as well as $a1$ and $a1x$ $a2$ and $a2x$ are different for the two above linear models. We would now like to apply simex to the naive model to see the results we get.
Input:

$(sim1\_simex\_fit = simex(sim1\_naive\_fit, SIMEXvariable = c(\text{"}a1x\text{"}, \text{"}a2x\text{"}),$
$measurement.error = cbind(1.2, 0.8), lambda = seq(.1, 2, by = .1)))$

And Rstudio gives us the following results.

Output:

$Naivemodel$:

$lm(formula = y \sim a1x + a2x, data = sim1\_data, x = T)$

$SIMEX - Variables : a1x, a2x$

$Number\ of\ Simulations : 100$

$Coefficients$:

$(Intercept)$ $a1x$ $a2x$

-0.00634 0.62955 0.88059

We can see that the value of the constant parameter is -0.00634, which indicates that SIMEX led us to the actual value of 0. Similarly, for the values of $a1x$ and $a2x$, they became 0.62955 and 0.88059, respectively, which shows that SIMEX led us to the actual values of 1 for the slopes $a1$ and $a2$. Initially, the constant was -0.01796 but it became -0.00634 after applying SIMEX, which is evidently closer to 0. Similarly, the values of $a1$ and $a2$ changed from 0.40419 and 0.62719 to 0.62955 and 0.88059, respectively, which are obviously closer to 1. Thus, we can conclude that the SIMEX method corrected our variables in the right direction.

But if we plot in Rstudio our results we are getting the following three plots Figure 2.3, Figure 2.4 and Figure 2.5 .

Input:

$plot(sim1\_simex\_fit)$

In order to visualize the values that Rstudio gave us we are making the following R code:

$truev = c(0, 1, 1)$

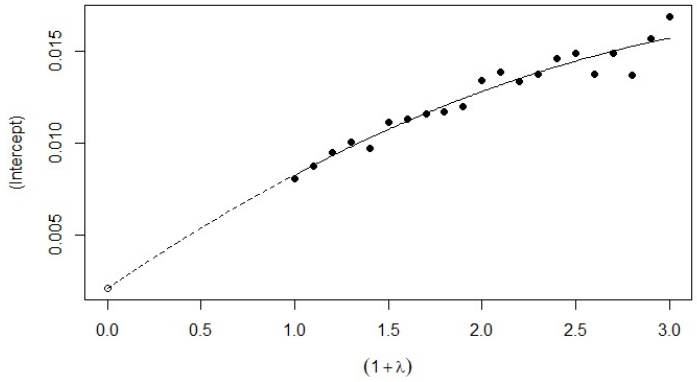

Figure 2.3: Plot 1

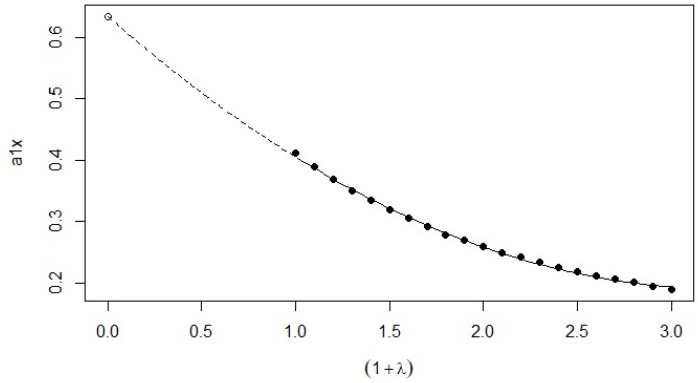

Figure 2.4: Plot 2

$namesv = c(0, 1, 2)$

$plot(namesv, truev, xlab = "intercept, a1, a2", ylab = "value", main = "Blue$
$for\ real\ values, red\ for\ naive\ values\ and\ green\ for\ simex\ values"$
$, pch = 15, col = "blue", cex = 1)$

$points(x = 0, y = -0.01796, pch = 15, col = "red", cex = 1)$

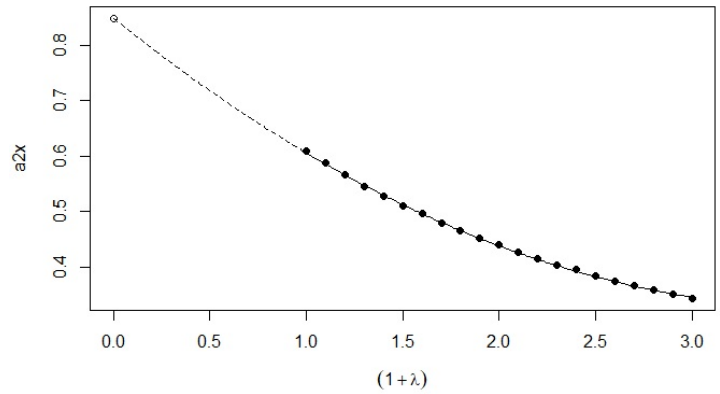

Figure 2.5: Plot 3

$points(x = 0, y = -0.00634, pch = 15, col = "green", cex = 1)$

$points(x = 1, y = 0.40419, pch = 15, col = "red", cex = 1)$

$points(x = 1, y = 0.62955, pch = 15, col = "green", cex = 1)$

$points(x = 2, y = 0.62719, pch = 15, col = "red", cex = 1)$

$points(x = 2, y = 0.88059, pch = 15, col = "green", cex = 1)$

The results we get are in the plot below: Figure 2.6.

Now we can clearly see that the green boxes are closer to blue boxes than the red ones.

The complete code for this simulation is in appendix A2.

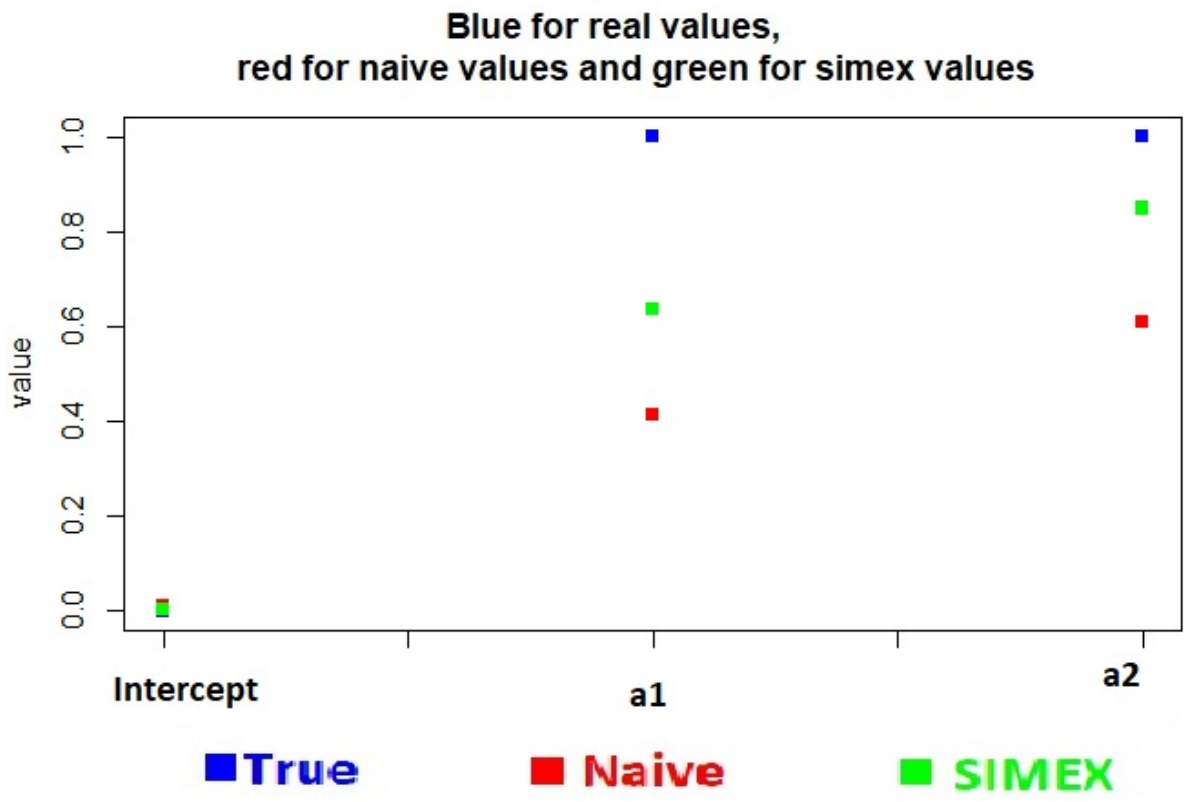

Figure 2.6: The visualized values

2.5.2 Extra example in Rstudio

To clarify, the OLS (ordinary least squares) method is commonly associated with a simple linear regression model that involves a single independent variable. For the purpose of demonstrating the regression line, we will perform an additional example based on the previous example, but this time we will only consider the variable $a1$ and disregard $a2$.

In this simulation, we are assuming that there is only one predictor variable, and the true relationship between the predictor and the outcome variable has a constant slope of 1. The reliability of the outcome measure is assumed to be constant and set at 1.2. We will use SIMEX to estimate the effect of measurement error on the slope estimate, and we know the true reliability of the measure that we will use in SIMEX. The sample size for the simulation is 5000. We will generate a set of 5000 observations where: $a1$ is the value from the standard normal distribution and $a1x$ is the corresponding obseration with the reliability at 1.2.

Input :

$$simextra = tibble(a1 = rnorm(5000), a1x = a1 + rnorm(5000) * 1.2, y = a1)$$

We now run the linear model:

$$(simextra\_true\_fit = lm(y \sim a1, data = simextra))$$

And Rstudio gives us the following results.

Output:

$Call$:

$lm(formula = y \sim a1, data = simextra)$

$Coefficients$:

$(Intercept)$ $a1$

$1.26e-16$ $1.00e+00$

As we see from the output the constant term is equal to 0 and the slope $a1$ is equal to 1. We now do the same procedure for the naive model with the variable $a1x$.
Input:

$(simextra\_naive\_fit = lm(y \sim a1x, data = simextra, x = T))$

And Rstudio gives us the following results.
Output:

$Call$:
$lm(formula = y \sim a1x, data = simextra, x = T)$
$Coefficients$:
$(Intercept)$ $a1x$
0.00577 0.40734

As we can see from the output, the constant term is equal to 0.00577 and the slope $a1$x is equal to 0.40734 . We notice that the values of the constant term, as well as $a1$ and $a1x$, are different for the two linear models above. We would now like to apply Simex to the naive model to see the results we get.
Input:

$(simextra\_simex\_fit = simex(simextra\_naive\_fit, SIMEXvariable = c("a1x"),$

$measurement.error = 1.2, lambda = seq(.1, 2, by = .1)))$

And Rstudio gives us the following results.

Output:

$Naivemodel$:

$lm(formula = y \sim a1x, data = simextra, x = T)$

$SIMEX - Variables : a1x$

$NumberofSimulations$: 100

$Coefficients$:

$(Intercept)$ a_1x

0.00372 0.62662

We can see that the constant parameter's value is 0.00372 , which indicates that SIMEX led us to the actual value of 0. Similarly, the value of $a1x$ changed from 0.40734 to 0.62662, which means that SIMEX led us to the actual value of 1 for the slope $a1$. The constant's initial value was 0.00577 , which became 0.00372 after applying SIMEX, appearing to be closer to 0. Likewise, $a1x$ changed from 0.40734 to 0.62662, which is evidently closer to 1. Therefore, from the above example, we can infer that SIMEX successfully corrected our variables in the right direction.

However, if we plot in Rstudio our results we are getting the following two plots Figure 2.7 and Figure 2.8 .

Input:

$plot(simextra\_simex\_fit)$

In order to plot the regression lines for the true, naive and simex models we

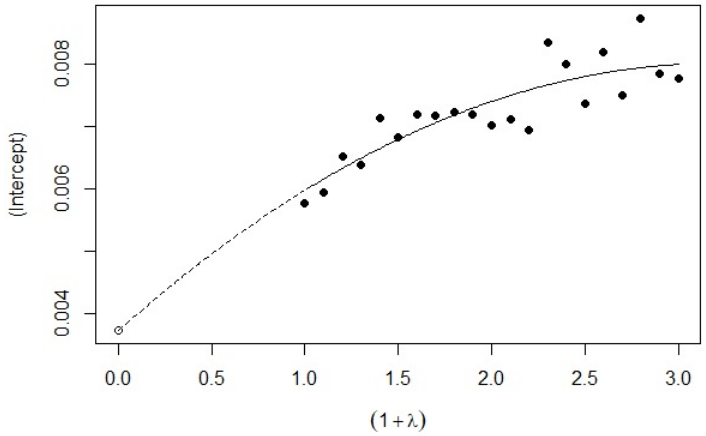

Figure 2.7: Plot extra 1

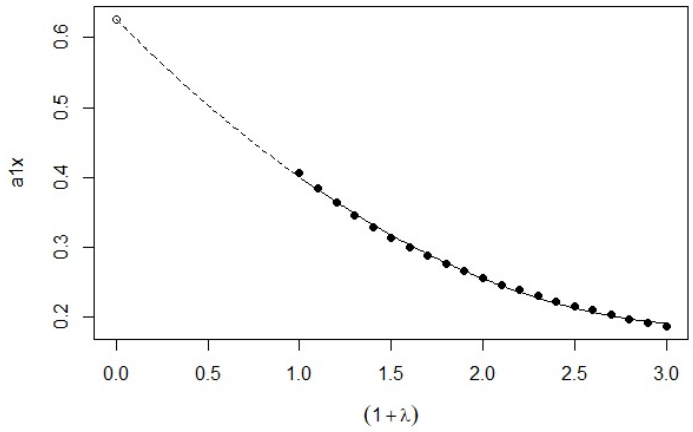

Figure 2.8: Plot extra 2

are using the following R code :

$plot(simextra\$a1, simextra\$y, xlab = "x", ylab = "y", main = "Blue\ for$

$real\ values, red\ for\ naive\ values\ and\ green\ for\ simex\ values")$

$abline(lm(simextra\$y \sim simextra\$a1), col = "blue")$

$abline(lm(simextra\$y \sim simextra\$a1x), col = "red")$

$abline(simextra\_simex\_fit, col = "green")$

The results are in the plot below: Figure 2.10:

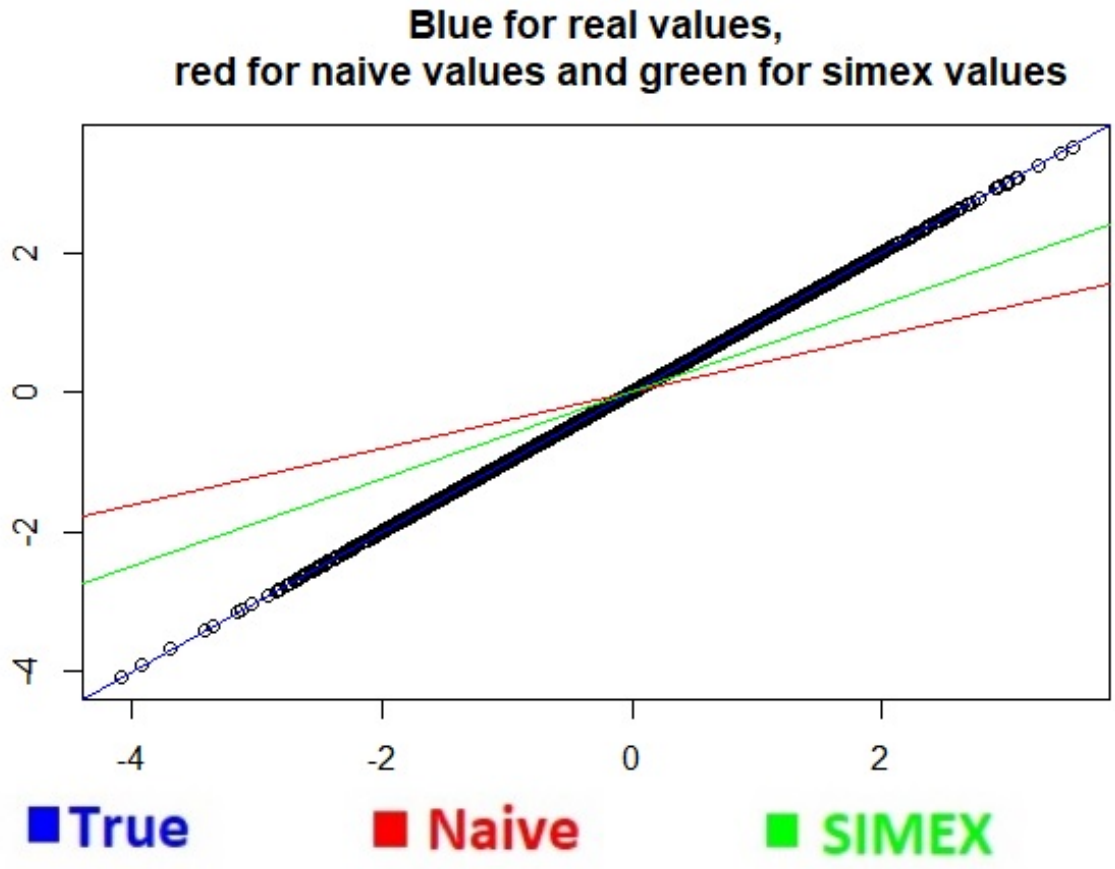

Figure 2.9: The regression lines of the true, naive and SIMEX simulation's

Now we can clearly see that the green (SIMEX) line is closer to the blue (True) line than the red (Naive) line.

It's important to note that in some cases, the intercept of the SIMEX line may not align with the desired direction. However, this doesn't necessarily mean that the SIMEX line won't be closer to the real line. When the slope of the SIMEX line is closer to the real slope, the SIMEX line will still be closer to the real line.

The complete code for this simulation is in the appendix A3.

2.6 Significant result for the multiple linear model

In this section, we will examine a case where the intercept of the regression model does not follow the desired path. After running the above example multiple times, we found that in some cases, the intercept does not tend towards zero, but instead, it increases the distance between the true value and the naive value.
A special case in the above extra example is, the following: Input :

$$simextra = tibble(a1 = rnorm(5000), a1x = a1 + rnorm(5000) * 1.2, y = a1)$$

We now run the linear model:

$(simextra\_true\_fit = lm(y \sim a1, data = simextra))$

And Rstudio gives us the following results.
Output:

$Call$:
$lm(formula = y \sim a1, data = simextra)$
$Coefficients$:
$(Intercept)$ $a1$
$6.28e-18$ $1.00e+00$

As we see from the output the constant term is equal to 0 and the slope $a1$ is equal to 1. We now do the same procedure for the naive model with the variable $a1x$.
Input:

$(simextra\_naive\_fit = lm(y \sim a1x, data = simextra, x = T))$

And Rstudio gives us the following results.
Output:

$Call$:
$lm(formula = y \sim a1x, data = simextra, x = T)$

$Coefficients$:

$(Intercept)$ $a1x$

-0.00603 0.41235

As we can see from the output, the constant term is equal to -0.00603 and the slope $a1$x is equal to 0.41235 . We notice that the values of the constant term, as well as $a1$ and $a1x$, are different for the two linear models above. We would now like to apply Simex to the naive model to see the results we get.

Input:

$(simextra\_simex\_fit = simex(simextra\_naive\_fit, SIMEXvariable = c("a1x"),$

$measurement.error = 1.2, lambda = seq(.1, 2, by = .1)))$

And Rstudio gives us the following results.

Output:

$Naivemodel$:

$lm(formula = y \sim a1x, data = simextra, x = T)$

$SIMEX - Variables : a1x$

$NumberofSimulations : 100$

$Coefficients$:

$(Intercept)$ a_1x

-0.00971 0.63746

We can see that the constant parameter's value is -0.00971. The value of $a1x$ changed from 0.41235 to 0.63746, which means that SIMEX led us to the actual value of 1 for the slope $a1$. The constant's initial value was -0.00603 and became -0.00971 after applying SIMEX, appearing to be more distant from 0. Instead, $a1x$ changed from 0.41235 to 0.63746, which is evidently closer to 1. Therefore, we can see that in that specific example, the intercept did not behave as we wanted. However, is that actually a problem? We will be plotting regression lines for the true, naive, and SIMEX models using the following R code:

$plot(simextra\$a1, simextra\$y, xlab = "x", ylab = "y", main = "Blue\ for\ real\ values, red\ for\ naive\ values\ and\ green\ for\ simex\ values")$

$abline(lm(simextra\$y \sim simextra\$a1), col = "blue")$

$abline(lm(simextra\$y \sim simextra\$a1x), col = "red")$

$abline(simextra\_simex\_fit, col = "green")$

The results are in the plot below: Figure 2.10:
Now we can clearly see that the green (SIMEX) line is closer to the blue (True) line than the red (Naive) line as before.

The example highlights that it is not essential for all variables in a model to converge to their true values, as long as the overall model converges to-

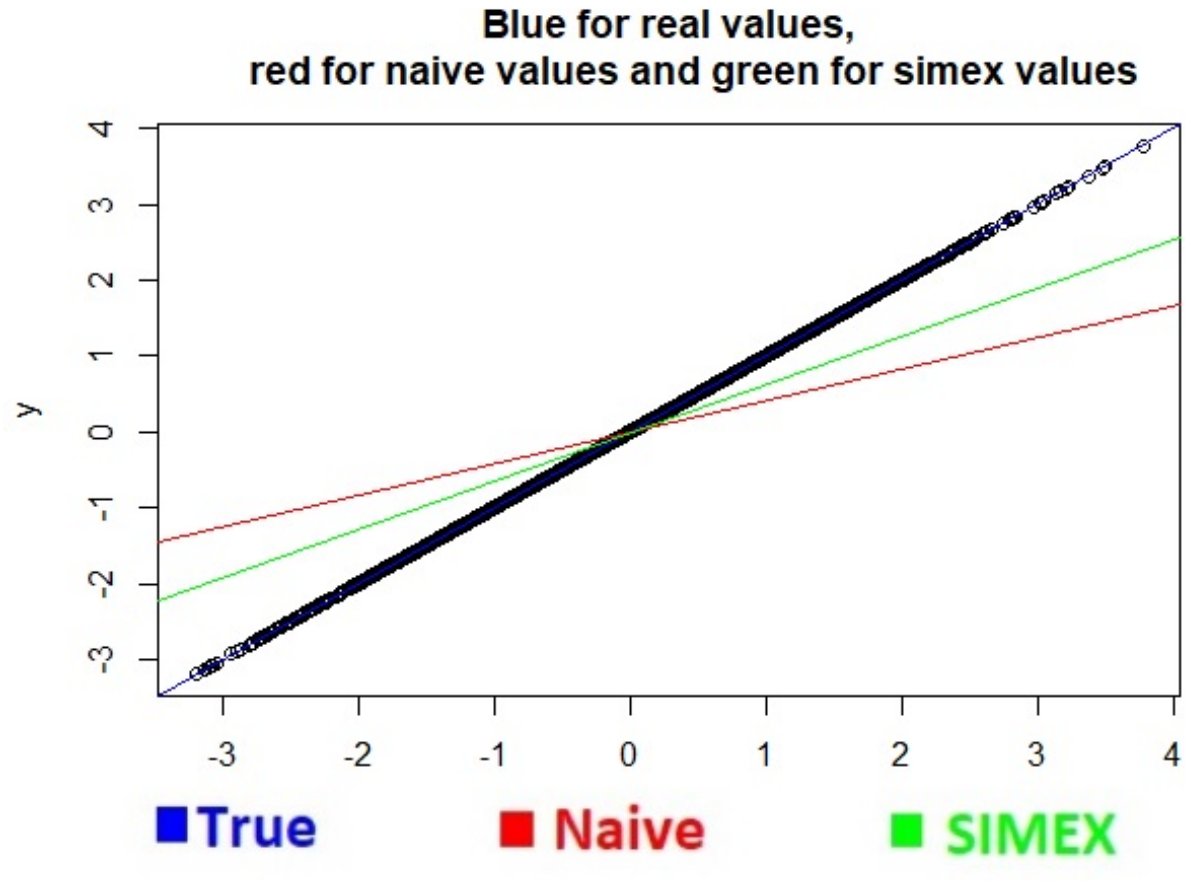

Figure 2.10: The regression lines of the true, naive and SIMEX simulation's

wards the true model of the data without measurement error. This idea can be extended to multiple linear regression models, where some variables may not converge to their true values, but simex will still lead towards the true regression that would exist in the absence of measurement error. Essentially, simex allows us to estimate the true model of the data by accounting for measurement error, even if some variables in the model do not converge to their true values. We can visualize this by imagining the simex (green) line converging towards the true (blue) regression line, even if not all variables in

the model converge to their true values.

Chapter 3

Large US Data Analysis

3.1 Introduction

We would like to investigate the data we have from a cardiology clinic as we know that the time of death is misreported (for example the patient died after the surgery and nobody declared his death), so obviously we have measurement error in our data. We also have a second database where the time of death is reported accurately. By comparing these two databases we will be able to find the measurement error present in the data where death is misreported. We will also use the SIMEX method to deal with the measurement error in our data and see how close the SIMEX method leads us to the real data. We will do our analysis in Rstudio.

3.2 Data Preview

The code used for the analysis is in Appendix A4.

In the data from the cardiologic clinic we have $n_1 = 9533$ persons and for each one we have 21 variables such as the ssn[1] (Social Security Number) if he reported , the day of birth, the day of admission, the day of surgery, the day of discharge , the day of follow-up etc. In the second "final" validated data set where there is no measurement error we have $n_2 = 4852$ persons and 237 variables for each one.

We would like to find in the two different data sets the same variables that express the same thing so that we can compare the data with measurement error to the validated data. First we see that the data sets have different number of people and different number of variables so we will look for the people present in both data sets and we do that via the ssn which is unique for every person. We find that n = 2822 individuals are in the two datasets. For them we would like to gather information such as the actual age at death, the wrongly reported age at death, the gender, the age at which the operation was performed, the actual survival time, the false survival time and whether or not they died.

Then we would like to use the built-in simex function found in Rstudio. So we would like to build a linear model with the variables of interest that

[1]The term Social Security Number (SSN) refers to a numerical identifier assigned to U.S. citizens and other residents to track income and determine benefits.

we have gathered from the two data sets. With the variables at our disposal we could make several different linear models and see how the simex method worked in reducing the measurement error.
But there's a problem with the simex function, it needs the measurement error which in our case is unknown. Without the availability of a validation subset or known variance for the outcome error, our method provides analysts with a new tool to perform sensitivity analyses that vary assumptions about the underlying measurement error variance and examine the robustness of results to random error in the event time.

In our first attempt we would like to replicate the idea as we used the simex function in the examples from the simex paragraph, specifically like the extra example in "SIMEX example 1 in Rstudio". From the two data sets we extract the data and we refer to the data without the measurement error as real and to the data with the measurement error included as naive. It is easy in this example to try several different measurement errors and see which one gets us closer to the numbers 0 for β_0 and 1 for β_1.

$y\_real = age\_real$

$(simex\_true\_age = lm(y\_real \sim age\_real))$

$(simex\_naive\_age = lm(y\_real \sim age\_naive, x = T))$

$(simex\_simex\_age = simex(simex\_naive\_age, SIMEXvariable = c("age\_naive"),$

$measurement.error = 0.8, lambda = seq(.1, 2, by = .5)))$

After many tests of the above code for different measurement errors we find that the value 0.8 performs better than all the others.
The outcome of the true model (is given through $simex\_true\_age$) is 0 for β_0 and 1 for β_1 by default, the outcome of the naive model (is given through $simex\_naive\_age$) is 0.4607 for β_0 and 0.9139 for β_1 (the coefficients). While the outcome of the simex method (is given through $simex\_simex\_age$) is 0.04628 for the β_0 and 0.91919 for β_1 and it's obvious that the simex method reduce's the bias.

This example is very basic and uses only one variable of the many that we obtained from the data sets, we can make other regressions with those variables like the following example.

$(simex\_true\_age111 = lm(age\_real \sim gender+age\_at\_surgery+survival\_real))$

$(simex\_naive\_age111 = lm(age\_real \sim gender+age\_at\_surgery+survival\_naive, x = T))$

$(simex\_simex\_age111 = simex(simex\_naive\_age111, SIMEXvariable = c(\text{"}survival\_naive\text{"}), measurement.error = 0.8, lambda = seq(.1, 2, by = .5)))$

The above example looks like any other example that we might like to use

in order to gather data like the coefficients from the linear model and their meanings.

For example someone wants to make the above linear model but he's not interested in the effect of the gender in the outcome survival age. Then easily he can make the following simulation by removing the variable gender in the above models.

$$(simex\_true\_age1111 = lm(age\_real \sim age\_at\_surgery + survival\_real))$$

$$(simex\_naive\_age1111 = lm(age\_real \sim age\_at\_surgery + survival\_naive, x = T))$$

$$(simex\_simex\_age1111 = simex(simex\_naive\_age1111, SIMEXvariable = c(\text{"}survival\_naive\text{"}), measurement.error = 0.8, lambda = seq(.1, 2, by = .5)))$$

But there is one more negative thing with the ready-made simex function that Rstudio has and is that it only accepts linear models and general linear models. Since we have survival data we would like to consider other models like the Cox proportional hazards model and Weibull parametric regression model to study the effects of random error in survival time T (T is the time starting from a defined point, which in our case is the operation, to the occurrence of a given event, which in our case is death).The COX model and Weibull model are both statistical models commonly used in survival analysis.

The COX model, also known as the proportional hazards model, is a type of regression model used to analyze the relationship between one or more predictor variables and a survival outcome. It assumes that the hazard function (the instantaneous rate at which events occur) is proportional across different levels of the predictor variables. This allows for the estimation of hazard ratios, which indicate the relative risk of an event occurring at one level of the predictor variable compared to another.

The Weibull model, on the other hand, is a parametric model that assumes that the hazard function follows a Weibull distribution. This model can be used to estimate the shape parameter and scale parameter of the distribution, which can be used to interpret the hazard function and make predictions about survival probabilities. The Weibull model is often used when the hazard rate changes over time, with a decreasing or increasing hazard rate over time. Both the COX model and Weibull model are commonly used in medical research and other fields where the analysis of survival data is necessary.

The Simulation and Extrapolation method (SIMEX) was developed by Cook and Stefanski to correct additive measurement error in the covariates. SIMEX has been applied to a wide variety of regression models and is generally implemented as an empirical method. It has been shown to be a useful tool for estimation in the presence of unbiased covariate measurement error in regression models for time-to-event outcomes, e.g., see Zhang et al. , He et al. , and Greene and Cai. The extended SIMEX approach addresses random

multiplicative error in the event time and we study whether this method can be applied to reduce bias in the regression coefficients. The Cox model is given by

$$\lambda(t) = \lambda_0(t)exp(X\beta)$$

where λ(t) is the hazard at time t given the p $\times$ 1 covariate vector X, λ_0(t) is the baseline hazard, and β is the vector of log hazard ratio parameters. The Weibull (AFT) model is given by

$$T = exp(\alpha_0 + X\alpha_1 + \sigma\epsilon) \tag{3.1}$$

where α_0 and α_1 are regression coefficients, σ is a shape parameter, and ϵ is the error term following an extreme value distribution. The model is also known as a linear transformation model, given by

$$log(T) = \alpha_0 + X\alpha_1 + \sigma\epsilon.$$

We are considering a situation where the survival time, denoted by T, is measured with error. Specifically, we observe a version of the survival time, denoted by T', that is subject to multiplicative error. This means that the observed survival time, T', is equal to the true survival time, T, multiplied by a random variable, $exp(v)$, where v is another random variable that represents the error in the measurement.

To clarify, if we knew the true value of v for a particular individual, we could correct the observed survival time by dividing by $exp(v)$ to obtain an estimate of the true survival time. However, since we don't know the true

value of v, we need to work with the observed survival time and account for the error. The variable v has a mean of 0 and a variance of σ_v^2. It is assumed to be independent of both the true survival time, T, and any covariates represented by the variable X. Then the error prone survival time on the log-scale is given by

$$log(T') = \alpha_0 + X\alpha_1 + \sigma\epsilon + v = log(T) + v. \tag{3.2}$$

We note that the form of the equation (3.2) is similar to the equation (1.2). In situations where there is outcome error, we can assess the performance of the Cox and Weibull models by comparing their ability to accurately capture the association between the predictor variable (X) and the outcome. Specifically, we can compare the log hazard ratio estimates from the Cox and Weibull models. This comparison is possible because the log hazard ratio can be estimated from both models, allowing for a direct comparison of their performance.

The log hazard ratio from the Cox model can be estimated from the Weibull model with the following relationship:

$$\beta = -\frac{\alpha_1}{\sigma}.$$

The linear equation for $log(T')$ mentioned in the statement is not affected by an additional error term, v, that is indepedent of the covariate X. This

means that the presence of v does not introduce any bias in the estimation of the acceleration parameter when using a typical linear regression model. We also note that the error equation in (3.2) has the same mathematical form as a log-linear event time model with an added frailty term, v.This means that the presence of v can be interpreted as a random effect or unobserved heterogeneity that affects the survival time, but it is not related to the covariates included in the model. Therefore, it is possible to model the effect of v using a frailty model to account for the unobserved heterogeneity and improve the estimation of the survival function.

Keiding et al. considered the AFT model for the setting of heterogeneity due to omitted covariates or frailties and observed that there is bias in the Cox model induced by erroneously ignoring an added frailty term v, whereas there is no expected bias in the acceleration parameter α_1.

One way to measure this bias is by using an "approximate attenuation factor," which takes into account the magnitude of the error in the survival function. This attenuation factor provides an estimate of the amount by which the true hazard ratio is attenuated, or reduced, by the presence of the error in the survival function when using the Cox model.
These authors also derived an approximate formula for the attenuation factor for the hazard ratio parameter in the Cox model, drawing connections between the log-linear model for the uncensored event time and the theoretical linear regression of $log(T)$ on X.
When adapted to our setting, the bias in $\hat{\beta}_{naive}$, the estimated hazard ratio

from naively applying the Cox model in the presence of the error in (3.2), is given by the approximate attenuation factor

$$\gamma = \frac{1}{\sqrt{1 + \sigma^{-2}\sigma_v^2}}. \tag{3.3}$$

That is , $\hat{\beta}_{naive} \approx \beta \times \gamma$.

The framework for survival models and outcome errors provides a basis for adjusting the estimation of a regression parameter, such as the log hazard ratio β, using the SIMEX method. While the SIMEX method was originally developed for addressing measurement error in the covariates, we can adapt it for the purpose of adjusting for multiplicative error by working with the log T. In doing so, we can convert the assumed multiplicative error into an additive scale that can be more easily addressed using the SIMEX method. Essentially, we are using a logarithmic transformation to reframe the problem and make it more amenable to analysis with the SIMEX method.

The method described is similar to SIMEX, a technique used to estimate the bias in the parameter estimates when measurement errors are present. In this case, we are interested in estimating the bias in the naive estimate of a parameter of interest, which ignores the measurement error. To do this, we start by estimating the relationship between the size of the measurement error, denoted as σ_v^2, and the bias in the naive estimate of the parameter.

Next, we simulate the effect of measurement error on the outcome variable

by adding extra measurement error to each outcome. Specifically, we draw random variables, denoted as ω, from a normal distribution with mean zero and variance $\lambda\sigma_v^2$. We add these random variables to the error-prone variable, log T, and exponentiate the result to obtain a new variable, denoted as $T\lambda_b'$. We repeat this process B times for a range of values of $\lambda \geq 0$. For each iteration of λ and b = 1,..., B, we refit the regression model using the new vector of error-prone measurements of the survival time, $T\lambda_b'$, to obtain a new estimate of the naive log hazard ratio, denoted as $\beta_{\lambda b}$ (or other parameter of interest, such as the acceleration parameter from the AFT model). Finally, we compute the new total measurement error variance in $\log T\lambda_b'$ using the formula given by

$$\sigma_v^2 + \lambda\sigma_v^2 = (1+\lambda)\sigma_v^2. \tag{3.4}$$

For illustration, we set B = 50 and λ in {0,0.5,1,1.5,2} to estimate new $\beta_{\lambda 1}$, which are shown as small circles in Figure 3.1 , and plot these naive $\beta_{\lambda 1}$ versus λ. In the Extrapolation step, we then fit a curve to the plot of $\beta_{\lambda 1}$ as a function of the λ 's. From this fitted model, we extrapolate back to λ = -1, which given the new total measurement error variance in equation (3.4) , should approximate the true coefficient value.

For the setting with covariate measurement error, Cook and Stefanski recommend a quadratic approximation due to good performance in most cases, but other extrapolation functions such as a linear, loglinear, or nonlinear function

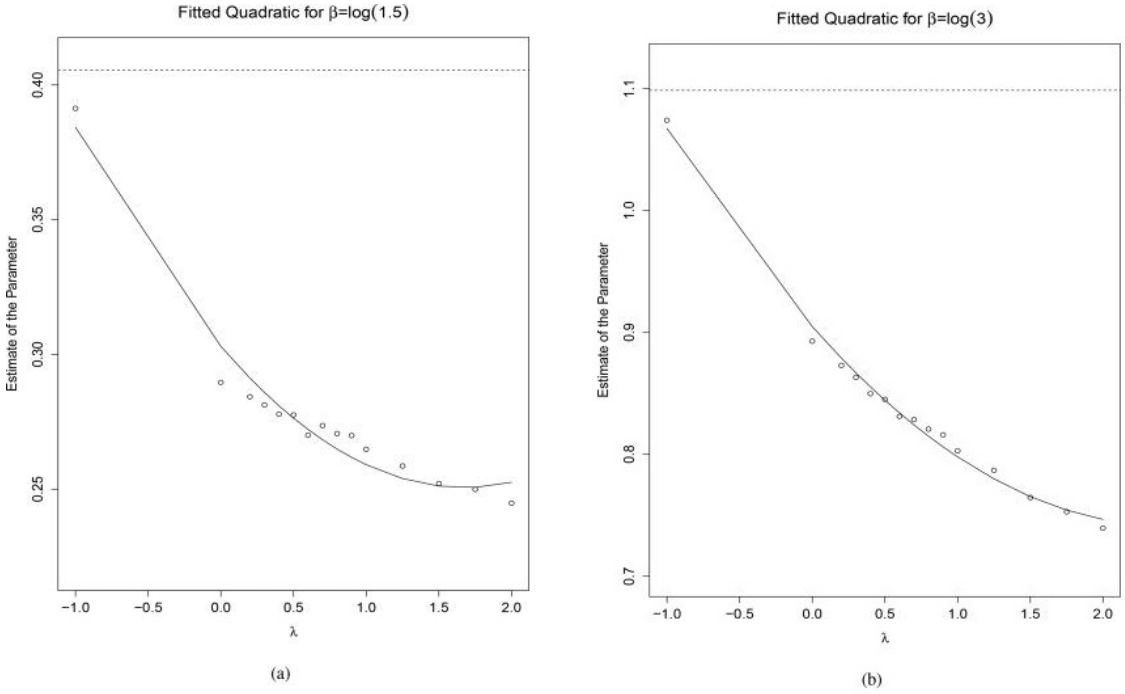

Figure 3.1: The quadratic approximations of the β parameters as a function of λ, extrapolated to λ = -1, with the dotted lines denoting the true β for $\beta = \log(1.5)$ (a) and $\beta = \log(3)$ (b)(arbitrarily chosen initial values for b as in: Oh EJ, Shepherd BE, Lumley T, Shaw PA. Considerations for analysis of time-to-event outcomes measured with error: Bias and correction with SIMEX.)

are possible. Based on our simulations, we found that the quadratic form performed better than the linear and loglinear approximations in our framework. In other words, when we tested these different mathematical models in our system, the quadratic form showed superior performance compared to the other models. In Figure 3.1, this extrapolation is shown by extending the curve to λ = - 1.

The given paragraph explains a method to assess the appropriateness of an extrapolation function used in a data application. The method involves

creating a plot by increasing the denseness of the vector λ and observing the resulting curve. However, this approach is only an approximation because we can only generate curves for which $\lambda \geq 0$ and it does not provide estimates for the curve in the region between [-1 and 0). To evaluate the sampling variability of the SIMEX estimates, a bootstrap technique is used to obtain standard errors. This involves repeatedly resampling the data to create multiple datasets and then calculating the SIMEX estimates for each dataset. By analyzing the variability in the estimates across the multiple datasets, we can determine the standard error of the estimates, which provides a measure of how much the estimates may vary due to sampling variation.

In certain situations, we may already have information about the variance of the measurement error, represented by σ_v^2, either from a validation study or a previous experiment. In such cases, we can assume that the value of σ_v^2 is known and use it in our analysis. However, there may be scenarios where we do not have an estimate of σ_v^2 or the true value of the measurement error variance is unknown. In such cases, we can still use the method by trying out different possible values for σ_v^2 and see how sensitive the estimated value of β is to the chosen value of σ_v^2. To illustrate this method, we can use an estimated value of σ_v^2 obtained from a validation subsample. We can apply this estimated value of σ_v^2 to our analysis and observe the effect it has on the estimated value of β.

We will now run simex with the Cox proportional hazard model for our data. To better understand the following lines of code, it would be beneficial

to simulate them on a computer using RStudio. The code is available in Appendix A4.

$true\_mod2 < -coxph(Surv(survival\_real, death\_occured) \sin age\_at\_surgery + gender)$

$naive\_mod2 < -coxph(Surv(survival\_naive, death\_occured) \sin age\_at\_surgery + gender)$

$simex\_finalmat < -cbind(simex\_betas, simex\_bias, simex\_CI\_lower, simex\_CI\_upper)$

For the thrue Cox model, 0.06481323 is the coefficient for the $age\_at\_surgery$ and -0.03977055 is the coefficient for the $gender$. For the naive Cox model, 0.06348539 is the coefficient for the $age\_at\_surgery$ and -0.05268084 is the coefficient for the $gender$ and for the simex betas in $simex\_finalmat$ we have that 0.06422506 is the coefficient for the $age\_at\_surgery$ and -0.04383477 is the coefficient for the $gender$.

This SIMEX "extended" method is not always as efficient as the built-in SIMEX method in Rstudio, as we estimate the measurement error from a random subset of our variables, the results will be different each time, depending on which subset is randomly chosen. This does not cease to extend the use of the SIMEX method to other models beyond the linear model.

Another problem that the bulit-in simex function has, is that it cannot take measurment error of any form. It is of note that the above simulations and data example all involved right-skewed error, with the mean error-prone T' larger than that of the true T, due to the nature of time-to-event data. Oh EJ, Shepherd BE, Lumley T, Shaw PA., investigated the performance of the method with left-skewed error for a small number of settings and found that SIMEX overestimated the true hazard ratio while still providing similar reduction in the magnitude of the bias for all settings.

Chapter 4

Appendix

A1 Proof that Var($\hat{\beta}$)=(x'x)$^{-1}\sigma_\epsilon^2$

$$\begin{aligned}
\text{Var}(\hat{\beta}) &= \text{Var}\left[(\mathbf{x}'\mathbf{x})^{-1}\mathbf{x}'\mathbf{y}\right] \\
&= (\mathbf{x}'\mathbf{x})^{-1}\mathbf{x}'\text{Var}(\mathbf{y})\mathbf{x}(\mathbf{x}'\mathbf{x})^{-1} \\
&= (\mathbf{x}'\mathbf{x})^{-1}\mathbf{x}'\text{Var}(\mathbf{x}\beta+\epsilon)\mathbf{x}(\mathbf{x}'\mathbf{x})^{-1} \\
&= (\mathbf{x}'\mathbf{x})^{-1}\mathbf{x}'\text{Var}(\epsilon)\mathbf{x}(\mathbf{x}'\mathbf{x})^{-1} \\
&= (\mathbf{x}'\mathbf{x})^{-1}\mathbf{x}'\sigma_\epsilon^2\mathbf{I}\mathbf{x}(\mathbf{x}'\mathbf{x})^{-1} \\
&= (\mathbf{x}'\mathbf{x})^{-1}\sigma_\epsilon^2\mathbf{x}'\mathbf{x}(\mathbf{x}'\mathbf{x})^{-1} \\
&= (\mathbf{x}'\mathbf{x})^{-1}\sigma_\epsilon^2.
\end{aligned}$$

A2 Code for the example 1 :

install.packages("*simex*")

$install.packages(\text{"}pacman\text{"})$

$install.packages(\text{"}tibble\text{"})$

$library(pacman)$

$library(tibble)$

$p_{l}oad(kirkegaard, simex, rms)$

$options(digits = 3)$

$sim1\_data = tibble($

$a1 = rnorm(5000),$

$a2 = rnorm(5000),$

$a1x = a1 + rnorm(5000) * 1.2,$

$a2x = a2 + rnorm(5000) * 0.8,$

$y = a1 + a2$

)

$fits$

$(sim1\_true\_fit = lm(y \sim a1 + a2, data = sim1\_data))$

$(sim1\_naive\_fit = lm(y \sim a1x + a2x, data = sim1\_data, x = T))$

$(sim1\_simex\_fit = simex(sim1\_naive\_fit, SIMEXvariable = c(\text{"}a1x\text{"}, \text{"}a2x\text{"}), measurement.e$
$cbind(1.2, 0.8), lambda = seq(.1, 2, by = .1)))$

$plot(sim1\_simex\_fit)$
$truev = c(0, 1, 1)$

$namesv = c(0, 1, 2)$

$plot(namesv, truev, xlab = \text{"}intercept, a1, a2\text{"}, ylab = \text{"}value\text{"}, main = \text{"}Blue$
$for\ real\ values, red\ for\ naive\ values\ and\ green\ for\ simex\ values\text{"}$
$, pch = 15, col = \text{"}blue\text{"}, cex = 1)$

$points(x = 0, y = -0.01796, pch = 15, col = \text{"}red\text{"}, cex = 1)$

$points(x = 0, y = -0.00634, pch = 15, col = \text{"}green\text{"}, cex = 1)$

$points(x = 1, y = 0.40419, pch = 15, col = \text{"}red\text{"}, cex = 1)$

$points(x = 1, y = 0.62955, pch = 15, col = "green", cex = 1)$

$points(x = 2, y = 0.62719, pch = 15, col = "red", cex = 1)$

$points(x = 2, y = 0.88059, pch = 15, col = "green", cex = 1)$

A3 Code for the extra part in the example 1 :

$simextra = tibble($

$a1 = rnorm(5000),$

$a1x = a1 + rnorm(5000) * 1.2,$

$y = a1)$

$(simextra\_true\_fit = lm(y \sim a1, data = simextra))$

$(simextra\_naive\_fit = lm(y \sim a1x, data = simextra, x = T))$

$(simextra\_simex\_fit = simex(simextra\_naive\_fit, SIMEXvariable = c("a1x"), measurement$
$1.2, lambda = seq(.1, 2, by = .1)))$

$plot(simextra\_simex\_fit)$

$plot(simextra\$a1, simextra\$y, xlab = "x", ylab = "y", main = "Blue\ for$
$real\ values, red\ for\ naive\ values\ and\ green\ for\ SIMEX\ values")$

$abline(lm(simextra\$y \sim simextra\$a1), col = "blue")$

$abline(lm(simextra\$y \sim simextra\$a1x), col = "red")$

$abline(simextra\_simex\_fit, col = "green")$

A4 $\#\#\#\#Example\ analysis\ of\ time-to-event\ dataset\ with\ error-$
$prone$
$\#\#\#times\ using\ SIMEX$

$install.packages("foreign")$

$library(foreign)$

$\#change\ working\ directory\ as\ needed$

$setwd("O : .....")$

$cardiak < -read.spss("O : ....."/Cardiac\_2020\_07\_Maths\_1989-2008\_9533\_1-$
$sensitivity - specificity\_2bull.sav")$

$cardiak1 < -read.spss("O : ....."/Cardiac\_2020\_07\_Maths\_1989-2008\_9533\_1-$
$sensitivity - specificity\_2bull.sav", to.data.frame = TRUE)$

$finaldata < -read.spss("O : ....."/01\_Circulation\_FINAL\_DATA\_no\_inhospital\_4852.sav")$

$finaldata1 < -read.spss("O : ....."/Desktop/01\_Circulation\_FINAL\_DATA\_no\_inhospital\_$
$TRUE)$

$\#The\ age\ is\ given\ by\ the\ following\ data$

$\#(dateofdeath - dateofbirth)/(365.25 * 24 * 60 * 60)$

$\#365.25\ days\ in\ a\ year$

$\#24\ hours\ in\ a\ day$

$\#60\ minutes\ in\ a\ hour$

$\#60\ seconds\ in\ a\ minute$

$ssn\_vector = c()$

$age\_real = c()$

$age\_naive = c()$

$gender = c()$

$age\_at\_surgery = c()$

$death\_occured = c()$

$survival\_real = c()$

$survival\_naive = c()$

$for(j\ in\ 1:9533)\{$

$if(cardiak1\$nossn[j] == "ssn")\{$

$for(i\ in\ 1:4852)\{$

$if(finaldata1\$SSN[i] == substr(cardiak1\$ssn[j], start = 0, stop = 12))\{$

$ssn\_vector <- append(ssn\_vector, finaldata1\$SSN[i])$

$age\_real <- append(age\_real, (finaldata1\$DOD[i] - cardiak1\$dob[j])/(365.25 * 24 * 60 * 60))$

$age\_naive <- append(age\_naive, (cardiak1\$dofollow[j] - cardiak1\$dob[j])/(365.25 * 24 * 60 * 60))$

$gender <- append(gender, finaldata1\$GENDER[i])$

$age\_at\_surgery <- append(age\_at\_surgery, cardiak1\$age[j])$

$death\_occured <- append(death\_occured, finaldata1\$ANCESTRY[i])$

$survival\_real <- append(survival\_real, (finaldata1\$SURVIVAL[i])/12)$

$survival\_naive <- append(survival\_naive, cardiak1\$years2018[j])$

}
}
}
}

ssn_vector

age_real

age_naive

$gender$

$age\_at\_surgery$

$death\_occured$

$survival\_real$

$survival\_naive$

$\#try1$
$y\_real = age\_real(simex\_true\_age = lm(y\_real \sim age\_real))$

$(simex\_naive\_age = lm(y\_real \sim age\_naive, x = T))$

$(simex\_simex\_age = simex(simex\_naive\_age, SIMEXvariable = c(\text{"}age\_naive\text{"}),$
$measurement.error = 0.8, lambda = seq(.1, 2, by = .5)))$

$\#try2$

$(simex\_true\_age111 = lm(age\_real \sim gender+age\_at\_surgery+survival\_real))$

$(simex\_naive\_age111 = lm(age\_real \sim gender+age\_at\_surgery+survival\_naive, x = T))$

$(simex\_simex\_age111 = simex(simex\_naive\_age111, SIMEXvariable = c(\text{"}survival\_naive\text{"}),$

$measurement.error = 0.5, lambda = seq(.1, 2, by = .5)))$

$\#try3$
$(simex\_true\_age1111 = lm(age\_real \sim age\_at\_surgery + survival\_real))$

$(simex\_naive\_age1111 = lm(age\_real \sim age\_at\_surgery+survival\_naive, x = T))$

$(simex\_simex\_age1111 = simex(simex\_naive\_age1111, SIMEXvariable = c(\text{"}survival\_naive\text{"}),$
$measurement.error = 0.8, lambda = seq(.1, 2, by = .5)))$

$library(survival)$

$\#$ *set up* $SIMEX$ *parameters (can be varied)*

$lambdavec < -c(0, 0.5, 1, 1.5, 2)$

$B < -50$

$\#\ set\ number\ of\ bootstrap\ iterations$

$Nboot < -100$

$true\_mod2 < -coxph(Surv(survival\_real, death\_occured) \sim age\_at\_surgery + gender)$

$true\_cox < -c(summary(true\_mod2)\$coef[, 1])$

$true\_se < -c(summary(true\_mod2)\$coef[, 3])$

$naive\_mod2 < -coxph(Surv(survival\_naive, death\_occured) \sim age\_at\_surgery + gender)$

$naive\_cox < -c(summary(naive\_mod2)\$coef[, 1])$

$naive\_se < -c(summary(naive\_mod2)\$coef[, 3])$

$n < -length(survival\_real)$

$\#\#\#\ run\ SIMEX\ procedure\ \#\#\#$

$\#\#\ Estimate\ simex\_sd\ from\ a\ validation\ subsample$

$\#\ select\ validation\ sample$

$R <- sample(1:n, 300, replace = FALSE) \#random\ sample\ of 300\ w/o\ replacement\ to\ be\ in\ validation\ sample$

$R\_survival\_real <- survival\_real[R] \#\ time\ for\ those\ in\ validation\ sample$

$R\_survival\_naive <- survival\_naive[R] \#\ timeprime\ for\ those\ in\ validation\ sample$

$R\_death\_occured <- death\_occured[R] \#\ event\ indicator\ for\ those\ in\ validation\ sample$

$\#\ covariates\ for\ those\ in\ validation\ sample$

$R\_age\_at\_surgery <- age\_at\_surgery[R]$

$R\_gender <- gender[R]$

$\#\ select\ subjects\ not\ in\ validation\ sample$

$noR\_survival\_real <- survival\_real[-R]$

$noR\_survival\_naive <- survival\_naive[-R]$

$noR\_death\_occured < -death\_occured[-R]$

$\#\ covariates\ for\ those\ not\ in\ validation\ sample$

$noR\_age\_at\_surgery < -age\_at\_surgery[-R]$

$noR\_gender < -gender[-R]$

$\#estimate\ simex\_sd\ from\ simulation\ validation\ subsample$

$simex\_sd1 < -sqrt(var(log(R\_survival\_naive) - log(R\_survival\_real)))$

$\#\ matrix\ to\ save\ new\ naive\ estimates$

$simex\_mat < -matrix(data = NA, nrow = length(lambdavec) * B, ncol = 3)$

$j < -1$

$for(lambda\ in\ lambdavec)\{$

$for(i\ in\ 1 : B)\{$

$simex\_err1 < -sqrt(lambda) * rnorm(n, 0, simex\_sd1)$

$survival\_simex <- -survival\_naive * exp(simex\_err1)$

$cox\_simex <- -coxph(Surv(survival\_simex, death\_occured) \sim age\_at\_surgery + gender)$

$simex\_mat[j,] <- -c(lambda, cox\_simex\$coef[1], cox\_simex\$coef[2])$

$j <- -j + 1$

}

}

$beta\_simex <- -simex\_mat[, 2:3]$

$lambda\_simex <- -simex\_mat[, 1]$

$\#\ fit\ quadratic\ function\ to\ the\ new\ naive\ estimates$
$simex\_formula\_age\_at\_surgery <- -lm(beta\_simex[, 1] \sim lambda\_simex + I(lambda\_simex^2))$

$simex\_formula\_gender <- -lm(beta\_simex[, 2] \sim lambda\_simex + I(lambda\_simex^2))$

$\#\ extrapolate\ quadratic\ function\ to\ lambda = -1\ to\ obtain\ simex\ estimates$

$simex\_estimate\_age\_at\_surgery <- predict(simex\_formula\_age\_at\_surgery, newdata = data.frame(lambda\_simex = -1))$

$simex\_estimate\_gender <- predict(simex\_formula\_gender, newdata = data.frame(lambda\_sim$
$-1))$

$simex\_betas <- c(simex\_estimate\_age\_at\_surgery, simex\_estimate\_gender)$

$\#\#\#\ bootstrap\ SIMEX\ procedure\ \#\#\#$

$\#\ define\ matrix\ to\ hold\ bootstrap\ SIMEX\ estimates$

$b\_simex\_betas <- matrix(NA, nrow = Nboot, ncol = 2)$

$for(f\ in\ 1 : Nboot)\{$

$\#\#\#\ stratified\ bootstrap\ of\ validation\ subsample\ to\ get\ bootstrap\ estimate$
$of\ error\ variance\#\#\#$
$\#\ select\ sample\ with\ replacement\ from\ those\ in\ validation\ subsample$

$R\_boot <- sample(1 : length(R\_survival\_real), length(R\_survival\_real), replace = TRUE)$

$R\_boot\_survival\_real <- R\_survival\_real[R\_boot]$

$R\_boot\_survival\_naive <- R\_survival\_naive[R\_boot]$

$R\_boot\_death\_occured <- R\_death\_occured[R\_boot]$

$\#\ bootstrap\ covariates\ from\ validation\ subsample$

$R\_boot\_age\_at\_surgery <- R\_age\_at\_surgery[R\_boot]$

$R\_boot\_gender <- R\_gender[R\_boot]$

$\#\ select\ sample\ with\ replacement\ from\ those\ not\ in\ validation\ subsample$
$noR\_boot <- sample(1 : length(noR\_survival\_real), length(noR\_survival\_real), replace = TRUE)$

$noR\_boot\_survival\_real <- noR\_survival\_real[noR\_boot]$

$noR\_boot\_survival\_naive <- noR\_survival\_naive[noR\_boot]$

$noR\_boot\_death\_occured <- noR\_death\_occured[noR\_boot]$

$\#\ bootstrap\ covariates\ from\ those\ not\ in\ validation\ subsample$

$noR\_boot\_age\_at\_surgery <- noR\_age\_at\_surgery[noR\_boot]$

$noR\_boot\_gender <- noR\_gender[noR\_boot]$

$\#\ estimate\ bootstrap\ simex\_sd$

$bsimex\_sd <- sqrt(var(log(R\_boot\_survival\_naive) - log(R\_boot\_survival\_real)))$

$\#\ get\ covariates\ and\ time,\ timeprime\ for\ all\ bootstrapped\ subjects$

$b\_survival\_real <- c(R\_boot\_survival\_real, noR\_boot\_survival\_real)$

$b\_survival\_naive <- c(R\_boot\_survival\_naive, noR\_boot\_survival\_naive)$

$b\_death\_occured <- c(R\_boot\_death\_occured, noR\_boot\_death\_occured)$

$b\_age\_at\_surgery <- c(R\_boot\_age\_at\_surgery, noR\_boot\_age\_at\_surgery) b\_gender <- c(R\_boot\_gender, noR\_boot\_gender)$

$\#\ set\ index\ for\ bootstrap\ SIMEX\ estimates$

$b\_simex\_mat <- matrix(data = NA, nrow = length(lambdavec) * B, ncol = 3)$

$l <- -1$

$for(lambda\ in\ lambdavec)\{$

$for(i\ in\ 1:B)\{$

$bsimex\_err <- -sqrt(lambda) * rnorm(n, 0, bsimex\_sd)$

$bsurvival\_simex <- -b\_survival\_naive * exp(bsimex\_err)$

$bcox\_simex <- -coxph(Surv(bsurvival\_simex, b\_death\_occured) \sim b\_age\_at\_surgery+ b\_gender)$

$b\_simex\_mat[l,] <- -c(lambda, bcox\_simex\$coef[1], bcox_simex\$coef[2])$

$l <- l + 1$

}

}

$b\_beta\_simex <- -b\_simex\_mat[, 2:3]$

$b\_lambda\_simex <- -b\_simex\_mat[, 1]$

$\#\ fit\ quadratic\ function\ to\ new\ bootstrap\ naive\ estimates$

$bsimex\_formula\_age\_at\_surgery <- -lm(b\_beta\_simex[, 1] \sim b\_lambda\_simex+ I(b\_lambda\_simex^2))$

$bsimex\_formula\_gender <- lm(b\_beta\_simex[,2] \sim b\_lambda\_simex + I(b\_lambda\_simex^2))$

$\#\ extrapolate\ quadratic\ function\ to\ lambda = -1\ to\ obtain\ SIMEX\ estimates$

$Bsimex\_age\_at\_surgery <- predict(bsimex\_formula\_age\_at\_surgery, newdata = data.frame(b\_lambda\_simex = -1))$

$Bsimex\_gender <- predict(bsimex\_formula\_gender, newdata = data.frame(b\_lambda\_simex = -1))$

$b\_simex\_betas[f,] <- c(Bsimex\_age\_at\_surgery, Bsimex\_gender)$

}

$naive\_bias <- round((naive\_cox - true\_cox)/true\_cox, 4) * 100$

$naive\_CI\_lower <- naive\_cox + qnorm(.025) * naive\_se$

$naive\_CI\_upper <- naive\_cox + qnorm(.975) * naive\_se$

$naive\_finalmat <- cbind(naive\_cox, naive\_bias, naive\_CI\_lower, naive\_CI\_upper)$

$\#\#\ calculate\ SIMEX\ HR\ bias\ and\ CIs$

$$simex\_bias < -round((simex\_betas - true\_cox)/true\_cox, 4) * 100$$

$$beta\_se < -apply(b\_simex\_betas, 2, sd)$$

$$simex\_CI\_lower < -simex\_betas + qnorm(.025) * beta\_se$$

$$simex\_CI\_upper < -simex\_betas + qnorm(.975) * beta\_se$$

$$simex\_finalmat < -cbind(simex\_betas, simex\_bias, simex\_CI\_lower, simex\_CI\_upper)$$

Chapter 5

References

Aigner 1973, Regression With a Binary Independent Variable Subject to Errors of Observation,Journal of Econometrics 1 ,49-90

Apanasovich TV, Carroll RJ, Maity A. SIMEX and standard error estimation in semiparametric measurement error models. Electron J Stat. 2009 Jan 1;3:318-348. doi: 10.1214/08-EJS341. PMID: 19609371; PMCID: PMC2710855.

Bound and Krueger (1991) The Extend of Measurement Error in Longitudinal Earnings Data: Do Two Wrongs Make a Right? Journal of Labor Economics 9, 1- 24.

Bound et.al. (1994) Evidence on the Validity of Cross-Sectional and Longitudinal Labor Market Data, Journal of Labor Economics 12, 345-368

Carroll, R. J., Ruppert, D., Stefanski, L. A., & Crainiceanu, C. M. (2006). Measurement error in nonlinear models: a modern perspective, Second Edition. Boca Raton, FL: Chapman & Hall/CRC.

Cook, John R., and Leonard A. Stefanski. Simulation-extrapolation estimation in parametric measurement error models. Journal of the American Statistical association

Emil O. W. Kirkegaard, The simulation and extrapolation method to measurement error in models

Griliches and Hausman (1986) Errors in Variables in Panel Data Journal of Econometrics 31, 93-118

He W, Yi GY, Xiong J. Accelerated failure time models with covariates subject to measurement error. Statistics in Medicine. 2007; 26(26):4817–4832.

Hyslop and Imbens (2001), Bias From Classical and Other Forms of Measurement Error,Journal of Business and Economic Statistics 19, 475-481.

Kane, Rouse, and Staiger (1999) Estimating Returns to Schooling When Schooling is Misreported, NBER Working Paper No. 7235

Keiding, Niels, Per Kragh Andersen, and John P. Klein. The role of frailty models and accelerated failure time models in describing heterogeneity due

to omitted covariates.

Klepper and Leamer (1984) Consistent Sets of Estimates for Regressions with Errors in All Variables, Econometrica 52, 163-183

Maddala (1977) Econometrics, New York: McGraw Hill

Oh EJ, Shepherd BE, Lumley T, Shaw PA. Considerations for analysis of time-to-event outcomes measured with error: Bias and correction with SIMEX. Stat Med.

Pischke, Steve, (2007) Lecture Notes on Measurement Error.

Stefanski, L. A., Cook, R. D., & Boos, D. D. (2002). Simulation-extrapolation: a general framework for extending statistical models. American Statistician, 56(4), 302-310.

Stefanski, L. A., & Boos, D. D. (2002). The calculus of M-estimation under random censorship. Statistical Science, 17(2), 219-237.

William M.K. Trochim, Research Methods Knowledge Base William M.K. Trochim