**JSM2024 Short Course: Understanding and tackling measurement error:**

**A review of modern practical methods: Hands On Session #1**

1. Consider the Regression calibration example from RC Case 1b (slide 56), where 2 observations of X\* with classical measurement error (X\* = X + u) are observed instead of X and we are interested in estimates of the coefficients for the simple logistic regression for the binary outcome Y: logit( Prob(Y=1)) = b0 + b1 X. On the Github page: <https://github.com/PamelaShaw/CovarError-Hands-on-2024>

you can download the code that generated the data for this example, file “Session2\_RC\_20240802.R”

1. Use the following approach to fitting the calibration model: regress the 2nd observation of X\* on the first observation and obtain a new estimate of and perform regression calibration to get an estimate of b1
2. Is the approach in a) valid for estimating , and why?
3. Obtain the bootstrap standard errors for the regression coefficients for the regression calibration approach to fitting the outcome model and compare them to those in lecture. How do they compare?
4. Use the observed values of and to obtain an estimate of the standard error for the measurement error in X\*
5. Fit the naïve outcome model using X\* instead of X, and then use the answer in d to apply the SIMEX method to correct the coefficients in this model. How do these regression estimates compare to the answer in a?
6. Try different assumed estimates of the measurement error variance in X\*, including the true value, and see how sensitive your answer to e is to having a good estimate of that variance.
7. Download the simulated WHI dataset from GitHub. In this dataset the biomarker values are assumed to have classical measurement error and the self-reported values for the dietary exposures are assumed to have linear measurement error

### true log energy intake = logEn

### self-reported log energy intake = logEn\_sr

### biomarker log energy intake = logEn\_bm

### repeat measures of these variables have a 2 after them

### Physical activity = step\_1000 (units 1000 steps)

### Body mass index = BMI

### age at baseline = age

### race = race

### time = censored event time to first cancer

### delta = observed event indicator

1. Fit the calibration model for the self-reported log energy intake exposure, assuming the linear measurement model for the self-reported value and classical measurement error for the biomarker value. Hint: the calibration model should have same covariates as outcome model (see 1b).
2. Fit the following Cox model based on the true dietary exposure

True.model<-coxph(Surv(time,delta)~logEn+age+BMI+step\_1000+myrace, data=data)

1. Suppose you don’t observe the true logEn and only observe the self-reported value. Fit a similar outcome model to 2b, but consider the naïve model that ignores the error in the self report.
2. Now take the regression calibration approach to fit the outcome model, including getting standard errors that account for the uncertainty in estimating the calibrated exposure, and compare your estimates and standard errors to those of the naïve model and the true model.
3. Is the SIMEX method appropriate in this setting?
4. Regardless of your answer to 2e, apply the SIMEX method to estimate the outcome model and consider its performance relative to regression calibration in terms of how well it does to capture the true model. Will need the simexaft package to use SIMEX for survival data. Is SIMEX an improvement over the naïve method in this case?

**R code hint for Question 1f:** SIMEX for regressions with no other covariates in model I think had to be run this way

model.simex <-simex(naive.model, SIMEXvariable = "xstarbar", measurement.error = sd\_me, **asymptotic = FALSE**)

**Some R code hints for Question 2:** reading in data, formatting data, and AFTSIMEX for 2f

**library(openxlsx) ### For reading in excel data**

**library(survival) ### For survival regressions**

**library(simexaft) ### simex for the accelerated failure time model**

**library(boot) ## For bootstrap SE**

**#> Loading required package: uncomment this line to install if don’t have already**

**#install\_github("lboe23/sandwich2stage", subdir="pkg")**

**library(sandwich2stage)**

**### Reed in simulated dataset.Inspired by WHI dietary data**

**### Need to update the directory name to be where you saved the data file**

**data<-read.xlsx("~/Library/CloudStorage/OneDrive-KaiserPermanente/Meetings/JSM 2023/simulated\_whidata.xlsx")**

**names(data)**

**### Note sandwich2stage and simexFit did not seem to like factor variables, so create the binary dummary variables for each race**

**data$black<-ifelse(data$race=="black",1,0)**

**data$other<-ifelse(data$race=="other",1,0)**

**data$white<-ifelse(data$race=="white",1,0)**

**### turn race into a factor and make biggest group the reference category**

**data$myrace<-relevel(factor(data$race),ref="white")**

**###fit a AFT model and use simex for accelerated failure time models-- simexaft package**

**set.seed(120)**

**### Dont have repeats of the self reported diet, so could guess the measurement error variance is = to total variance in exposure**

**ind <- c("logEn\_sr")**

**err.mat <- var(data$logEn\_sr)**

**### saving some typing by creating formula variables**

**true.formula<- "Surv(time, delta) ~ logEn+ age + BMI + step\_1000+black+other"**

**#### Simex for survival time uses the accelerated failure time model**

**#### The AFT is a proportional hazards model but need to convert the paraemters to a hazard ratio**

**## Since data are computer generated we can consider the true exposure, for a benchmark**

**trueAFT<-survreg(formula = formula(true.formula), data = data, dist = "weibull")**

**### Compare hazard ratios estimated by the Cox model and AFT model**

**### Hazard ratio (HR) estimated by true Cox model**

**trueCox<-coxph(as.formula(true.formula), data=data)**

**exp(trueCox$coef)**

**### HR estimated by true AFT model, similar to Cox model**

**exp(-trueAFT$coef/trueAFT$scale)**

**#### #### Naive model using errorprone self-reported energy with no correction for error**

**naive.formula<- "Surv(time, delta) ~ logEn\_sr + age + BMI + step\_1000+black+other"**

**naiveAFT<-survreg(formula = formula(naive.formula), data = data, dist = "weibull")**

**### Look at SIMEX for weibull model, using quadratic extrapolation**

**simexFit <- simexaft(formula = formula(naive.formula), data = data, SIMEXvariable = ind, repeated = FALSE, err.mat = err.mat, B = 50, lambda = seq(0, 2, 0.1),extrapolation = "quadratic", dist = "weibull")**

**summary(simexFit)**

**plotsimexaft(simexFit,"logEn\_sr",ylimit=c(-3,1),extrapolation="quadratic")**

**### HR estimated by simex AFT model: Does terribly because exposure has systematic and not classical error in the self-reported energy variable**

**### Some inflation here, due to the error being more complicated than the simple classical error SIMEX assumes. Also had to guess at error variance.**

**### SIMEX estimated HR**

**exp(-simexFit$coef/simexFit$scale)**