571HW5

Coco_Luo

2023-02-28

Question 1

To begin with, I conducted a simulation study using both Mixed Models and Generalized Estimating Equations (GEE) Models. The simulation was performed under the assumption of missing completely at random (MCAR) with different levels of missingness, namely MCAR, Missing at Random (MAR), and Non-Missing at Random (NMAR).

For the simulation study, I specified a sample size of 500 with 5 clusters, where the true values of the regression coefficients were $\beta_0 = 2$, $\beta_1 = 0.5$ and $\beta_2 = 1$, and a correlation parameter of 0.75. The mixed effects model included two predictors and a random intercept for each cluster, while the GEE model included the same predictors and an exchangeable correlation structure within each cluster.

To introduce missingness, I randomly set 20% of the responses to missing values. I then fit the mixed effects model and the GEE model to the incomplete data using complete-case and available case analysis, and compared the results with the complete data analysis. I repeated this process for 300 iterations.

The following output summarizes the results obtained from the simulation study:

	β_0	β_1	β_2	$\beta_0(SE)$	$\beta_1(SE)$	$\beta_2(SE)$
gee_MCAR	2.006033	0.5042822	0.9988343	0.0721144	0.0539444	0.0240482
gee_MAR	2.007306	0.5041832	0.9987642	0.0722194	0.0541784	0.0240738
gee_MNAR	2.005049	0.5053266	0.9993180	0.0705527	0.0539950	0.0241375
lmm_MCAR	2.005936	0.5042740	0.9988797	0.0724939	0.0542804	0.0240967
lmm_MAR	2.007302	0.5041942	0.9987677	0.0725452	0.0544206	0.0240791
lmm_MNAR	2.005051	0.5053337	0.9993276	0.0709253	0.0542798	0.0241490

By comparing the different missing types, we observe that the standard error for Non-Missing at Random (NMAR) is lower than the other two types. Additionally, Linear Mixed Models (LMM) tend to have relatively higher standard errors. Surprisingly, despite this, the results show that the estimates are unbiased. However, upon further inspection, it was discovered that the estimates were biased in each iteration. This finding led us to a crucial statistical property, which states that the average of a large number of estimates tends to converge to the true value, even if the individual estimates are biased. Nevertheless, it is important to note that this is only valid if the bias is random and not systematic. If there is a systematic bias, such as a measurement error consistently overestimating or underestimating a parameter, then the average of the estimates may still be biased.

Question 2

`Participants`: The study will recruit 200 graduate students (100 male and 100 female) from two universities in the United States. Participants will be selected based on their willingness to participate, and they should be Taylor Swift fans.

`assumptions`:

- Academic performance and mental health outcomes follow a normal distribution.
- There are no significant differences between the two universities concerning the influence of Taylor Swift on academic performance and mental health.
- The effect of Taylor Swift on academic performance and mental health is uniform across all genders.

`Statistical Analysis Plan:`

The main goal of this study is to examine how Taylor Swift's influence affects the mental health and academic performance of graduate students over two years, using a longitudinal research design with four data collection points. To achieve this, we will compare the outcomes at three different follow-up time points (6 months, 12 months, and 24 months) to the baseline. The baseline data collected will include demographics, academic performance, and mental health information of the 200 participants.

Linear mixed-effects models will be used to determine the influence of Taylor Swift on academic performance and mental health by analyzing the changes in scores over time, with the Taylor Swift fandom scores serving as a predictor variable. The researchers will control for potential confounding variables like age, gender, and baseline academic performance/mental health. Descriptive statistics will be used to summarize the participants' baseline characteristics.

`Power and Sample Size:`

Given the limitations of cost and availability of subjects, the sample size for this study is fixed at 200. To determine the minimum detectable effect sizes (MDES) that we can observe with this sample size, we have made the following assumptions:

- Academic performance and mental health outcomes have a standard deviation of 1.0.
- The within-subject correlation is 0.5.
- The type I error rate is 0.05, and the power is 0.80.

Using these assumptions, we can calculate the MDES for academic performance and mental health separately. For academic performance, the MDES is approximately 0.13 GPA points, while for mental health, it is approximately 0.32 standard deviations.

To summarize, the statistical analysis plan for this study involves utilizing linear mixed-effects models to compare the alterations in academic performance and mental health over the two-year period between baseline and the three follow-up time points. Based on our power and sample size calculations, it appears that we can detect a minimum effect size of 0.13 GPA points for academic performance and 0.32 standard deviations for mental health, with a sample size of 200 and a power of 0.80. It is essential to take into account the assumptions made during these calculations when interpreting the results.

Appendix

```r
# Simulation of MCAR under Mixed Models and GEE Models
library(lme4)
library(gee)

# Set the seed for reproducibility
set.seed(123)

# Specify the sample size and the number of clusters
n <- 500
clusters <- 5
rho <- 0.75

# Generate the random intercepts for each cluster
u <- rnorm(clusters, 0, 1)

# Generate the predictor variables
x1 <- rnorm(n, 0, 1)
x2 <- rbinom(n, 1, 0.5)

# Generate the response variable
y <- rnorm(n, 2 + 0.5*x1 + 1*x2 + u[rep(1:clusters, each=n/clusters)], 1)

# Introduce missingness under MCAR
y.mcar <- ifelse(runif(n) < 0.2, NA, y)

# Convert the cluster variable to a factor
cluster <- factor(rep(1:clusters, each=n/clusters))

# Fit the mixed effects model
m1 <- lmer(y.mcar ~ x1 + x2 + (1 | cluster))

# Fit the GEE model
m2 <- gee(y.mcar ~ x1 + x2, id = cluster, data = data.frame(y.mcar, x1, x2))

# Compare the results with the complete data analysis
m1.complete <- lmer(y ~ x1 + x2 + (1 | cluster), data = data.frame(y, x1, x2, cluster))
m2.complete <- gee(y ~ x1 + x2, id = cluster, data = data.frame(y, x1, x2, cluster))
```

```r
# Set the seed for reproducibility
set.seed(123)

# Specify the sample size and the number of clusters
n <- 500
nc <- 5
rho <- 0.75

# Generate the random intercepts for each cluster
u <- rnorm(nc, 0, 1)

# Generate the predictor variables
x1 <- rnorm(n, 0, 1)
x2 <- rbinom(n, 1, 0.5)
```

```r
# Generate the response variable
y <- rnorm(n, 2 + 0.5*x1 + 1*x2 + u[rep(1:nc, each=n/nc)], 1)

# Introduce missingness under MCAR
y.mcar <- ifelse(runif(n) < 0.2, NA, y)

# Create a data frame with the complete cases
data.aca <- data.frame(y.mcar, x1, x2)
data.aca <- data.aca[complete.cases(data.aca), ]

# Convert the cluster variable to a factor
cluster <- factor(rep(1:nc, each=n/nc)[complete.cases(y.mcar)])

# Fit the mixed effects model using ACA
m1.aca <- lmer(y.mcar ~ x1 + x2 + (1 | cluster), data = data.aca)

# Fit the GEE model using ACA
m2.aca <- gee(y.mcar ~ x1 + x2, id = cluster, data = data.aca, corstr = "exchangeable")

# Compare the results with the complete data analysis
m1.complete <- lmer(y ~ x1 + x2 + (1 | cluster), data = data.frame(y, x1, x2, cluster))
m2.complete <- gee(y ~ x1 + x2, id = cluster, data = data.frame(y, x1, x2, cluster), corstr = "exchangea
```

```r
# Set the seed for reproducibility
set.seed(123)

# Specify the sample size and the number of clusters
n <- 500
clusters <- 5
rho <- 0.75
# Generate the random intercepts for each cluster
u <- rnorm(clusters, 0, 1)

# Generate the predictor variables
x1 <- rnorm(n, 0, 1)
x2 <- rbinom(n, 1, 0.5)

# Generate the response variable
y <- rnorm(n, 2 + 0.5*x1 + 1*x2 + u[rep(1:clusters, each=n/clusters)], 1)

# Introduce missingness under MAR
z <- 0.2*x1 + 0.8*x2 + rnorm(n, 0, 1)
p <- plogis(-1.5 + z) # use a logistic function to create the missingness probability
y.mar <- ifelse(runif(n) < p, NA, y)

# Convert the cluster variable to a factor
cluster <- factor(rep(1:clusters, each=n/clusters))

# Fit the mixed effects model
m1 <- lmer(y.mar ~ x1 + x2 + (1 | cluster))

# Fit the GEE model
m2 <- gee(y.mar ~ x1 + x2, id = cluster, data = data.frame(y.mar, x1, x2))
```

```r
# Compare the results with the complete data analysis
m1.complete <- lmer(y ~ x1 + x2 + (1 | cluster), data = data.frame(y, x1, x2, cluster))
m2.complete <- gee(y ~ x1 + x2, id = cluster, data = data.frame(y, x1, x2, cluster))
```

```r
# Set the seed for reproducibility
set.seed(123)

# Specify the sample size and the number of clusters
n <- 500
nc <- 5
rho <- 0.75
# Generate the random intercepts for each cluster
u <- rnorm(nc, 0, 1)

# Generate the predictor variables
x1 <- rnorm(n, 0, 1)
x2 <- rbinom(n, 1, 0.5)

# Generate the response variable
y <- rnorm(n, 2 + 0.5*x1 + 1*x2 + u[rep(1:nc, each=n/nc)], 1)

# Introduce missingness under the MAR mechanism
y.mar <- y
y.mar[x2 == 1] <- y[x2 == 1] + rnorm(sum(x2 == 1), 0, 1)

# Create a data frame with the complete cases
data.aca <- data.frame(y.mar, x1, x2)
data.aca <- data.aca[complete.cases(data.aca), ]

# Convert the cluster variable to a factor
cluster <- factor(rep(1:nc, each=n/nc)[complete.cases(y.mar)])

# Fit the mixed effects model using ACA
m1.aca <- lmer(y.mar ~ x1 + x2 + (1 | cluster), data = data.aca)

# Fit the GEE model using ACA
m2.aca <- gee(y.mar ~ x1 + x2, id = cluster, data = data.aca, corstr = "exchangeable")

# Compare the results with the complete data analysis
m1.complete <- lmer(y ~ x1 + x2 + (1 | cluster), data = data.frame(y, x1, x2, cluster))
m2.complete <- gee(y ~ x1 + x2, id = cluster, data = data.frame(y, x1, x2, cluster), corstr = "exchange
```

```r
# Set the seed for reproducibility
set.seed(123)

# Specify the sample size and the number of clusters
n <- 500
nc <- 5
rho <- 0.75
# Generate the random intercepts for each cluster
u <- rnorm(nc, 0, 1)

# Generate the predictor variables
x1 <- rnorm(n, 0, 1)
```

```r
x2 <- rbinom(n, 1, 0.5)

# Generate the response variable
y <- rnorm(n, 2 + 0.5*x1 + 1*x2 + u[rep(1:nc, each=n/nc)], 1)

# Introduce missingness under NMAR
z <- 0.2*x1 + 0.8*x2 + rnorm(n, 0, 1)
p <- plogis(-1.5 + z + 0.5*y) # use a logistic function that depends on the response variable to create
y.nmar <- ifelse(runif(n) < p, NA, y)

# Convert the cluster variable to a factor
cluster <- factor(rep(1:nc, each=n/nc))

# Fit the mixed effects model
m1 <- lmer(y.nmar ~ x1 + x2 + (1 | cluster))

# Fit the GEE model
m2 <- gee(y.nmar ~ x1 + x2, id = cluster, data = data.frame(y.nmar, x1, x2))

# Compare the results with the complete data analysis
m1.complete <- lmer(y ~ x1 + x2 + (1 | cluster), data = data.frame(y, x1, x2, cluster))
m2.complete <- gee(y ~ x1 + x2, id = cluster, data = data.frame(y, x1, x2, cluster))
```

```r
# Set the seed for reproducibility
set.seed(123)

# Specify the sample size and the number of clusters
n <- 500
nc <- 5
rho <- 0.75
# Generate the random intercepts for each cluster
u <- rnorm(nc, 0, 1)

# Generate the predictor variables
x1 <- rnorm(n, 0, 1)
x2 <- rbinom(n, 1, 0.5)

# Generate the response variable
y <- rnorm(n, 2 + 0.5*x1 + 1*x2 + u[rep(1:nc, each=n/nc)], 1)

# Introduce missingness under the NMAR mechanism with a monotone pattern
y.nmar <- y
threshold <- 1
y.nmar[y < threshold] <- NA

# Create a data frame with the complete cases
data.aca <- data.frame(y.nmar, x1, x2)
data.aca <- data.aca[complete.cases(data.aca), ]

# Convert the cluster variable to a factor
cluster <- factor(rep(1:nc, each=n/nc)[complete.cases(y.nmar)])

# Fit the mixed effects model using ACA
m1.aca <- lmer(y.nmar ~ x1 + x2 + (1 | cluster), data = data.aca)
```

```r
# Fit the GEE model using ACA
m2.aca <- gee(y.nmar ~ x1 + x2, id = cluster, data = data.aca, corstr = "exchangeable")

# Create a data frame with the available cases
data.complete <- data.frame(y.nmar[complete.cases(y.nmar)], x1[complete.cases(y.nmar)], x2[complete.cas

# Fit the mixed effects model with all observations
m1.complete <- lmer(y ~ x1 + x2 + (1 | cluster), data = data.complete)

# Fit the GEE model with all observations
m2.complete <- gee(y ~ x1 + x2, id = cluster, data = data.complete, corstr = "exchangeable")
```