Missingness Mechanisms

Comparing multiple imputation for a dataset under MCAR, MAR, MNAR mechanisms with varying degrees of missingness.

- Dataset is a linear model $y_i = \alpha + x_i\beta + \epsilon_i$ where parameter beta (true value = 1) is estimated
- Will observe if imputation strength decreases going from MCAR -> MAR -> MNAR w/ low missingness -> MNAR w/ high missingness as expected.

MCAR

```r
MCAR.create.data <- function(beta = 1, sigma2 = 1, n = 200,
                             run = 1) {
  set.seed(seed = run)
  x <- rnorm(n)
  y <- beta * x + rnorm(n, sd = sqrt(sigma2))
  cbind(x = x, y = y)
}
```

```r
MCAR.make.missing <- function(data, p = 0.5){
  rx <- rbinom(nrow(data), 1, p)
  data[rx == 0, "x"] <- NA
  data
}
```

```r
MCAR.test.impute <- function(data) {
  imp <- mice(data, print = FALSE)
  fit <- with(imp, lm(y ~ x))
  tab <- summary(pool(fit), "all", conf.int = TRUE)
  as.numeric(tab[2, c("estimate", "2.5 %", "97.5 %")])
}
```

```r
MCAR.simulate <- function(runs = 10) {
  res <- array(NA, dim = c(1, runs, 3))
  dimnames(res) <- list(c("MCAR"),
                        as.character(1:runs),
                        c("estimate", "2.5 %","97.5 %"))
  for(run in 1:runs) {
    data <- MCAR.create.data(run = run)
    data <- MCAR.make.missing(data)
    res[1, run, ] <- MCAR.test.impute(data)
  }
  res
}
```

```r
MCAR.res <- MCAR.simulate(100)
```

MCAR code inspiration courtesy of)

MAR

- MAR mechanisms involve different degrees of missingness for groups dependent on an observed variable in the dataset.
- I will run simulations for varied differences in missingness between the groups ($|p_2 - p_1|$) and see how multiple imputation performs.

```r
# Creates function that create a dataset with observations for a
# MUTLIPLE linear regression model with a categorical variable.
MAR.create.data <- function(beta = 1, sigma2 = 1, n = 10000,
                            run = 1, beta2 = 10, categorical_var = "gender" ,
                            category_one = "M", category_zero = "F") {
  set.seed(seed = run)
  x <- rnorm(n)
  # p represents probability of being in some group (E.g. Male = 1 vs Female = 0)
  z <- rbinom(n, 1, p = 0.5)
  temp <- cbind(x = x, z = z)
  # Note the beta2 term for categorical variable effect.
  y <- beta * x + rnorm(n, sd = sqrt(sigma2)) + beta2 * z
  z[z == 0] <- category_zero
  z[z == 1] <- category_one
  # cat_var = categorical_var
  data <- cbind(x = x, y = y, categorical_var = z)
  # colnames(data) <- c("x", "y", categorical_var)
  # sort df by categorical variable.
  df <- as.data.frame(data) %>%
    arrange(categorical_var)
}
```

- Creates function that removes p_0 % of data from category zero,
-

p_1 % of data from category_one. Note this is effectively MCAR within each group (definition of MAR). Note if p_0 = p_1, have a MAR mechanism.

```r
MAR.make.missing <- function(data, p_0 = 0.4, p_1 = 0.6 ){
  # don't generate nrow(data) times, generate count of category times.
  counts <- count(as.data.frame(data), categorical_var)
  num_cat_zero <-counts[1,2]
  num_cat_one <- counts[2,2]
  r_zero <- rbinom(counts[1,2], 1, p_0)
  r_one <- rbinom(counts[2,2], 1, p_1)
  data[1:num_cat_zero,1][r_zero == 1] <- NA
  data[(num_cat_zero +1):nrow(data), 1][r_one ==1] <- NA
  data
}
```

```r
# Function that calls mice (applying imputation) and applies Rubin's Rules,
# and creates 95% confidence intervals for parameter
MAR.test.impute <- function(data) {
  # Convert numerical vars to doubles (from character).
  data_num <- as.data.frame(apply(data[,c(1:2)], 2, as.numeric))
  # create copy
  data_num_complete <- data_num
  # add the categorical var
  data_num_complete$categorical_var <- data[,3]
  imp <- mice(data_num_complete, print = FALSE)
  fit <- with(imp, lm(y ~ x + categorical_var))
  tab <- summary(pool(fit), "all", conf.int = TRUE)
  as.numeric(tab[2, c("estimate", "2.5 %", "97.5 %")])
}
```

```
simulate <- function(runs = 10) {
  res <- array(NA, dim = c(5, runs, 3))
  dimnames(res) <- list(c("No_miss", "MCAR", "lightMAR", "moderateMAR", "extremeMAR"),
                        as.character(1:runs),
                        c("estimate", "2.5 %","97.5 %"))
  for(run in 1:runs) {
    data <- MAR.create.data(run = run)
    none_data <- MAR.make.missing(data, p_0 = 0, p_1 = 0)
    MCAR_data <- MAR.make.missing(data, p_0 = 0.5, p_1 = 0.5)
    light_data <- MAR.make.missing(data, p_0 = 0.4, p_1 = 0.6)
    moderate_data <- MAR.make.missing(data, p_0 = 0.3, p_1 = 0.7)
    heavy_data <- MAR.make.missing(data, p_0 = 0.2, p_1 = 0.8)
    res[1, run, ] <- MAR.test.impute(none_data)
    res[2, run, ] <- MAR.test.impute(MCAR_data)
    res[3, run, ] <- MAR.test.impute(light_data)
    res[4, run, ] <- MAR.test.impute(moderate_data)
    res[5, run, ] <- MAR.test.impute(heavy_data)

  }
  res
}
```

```
MAR.res <- simulate(100)
```

```
apply(MAR.res, c(1, 3), mean, na.rm = TRUE)
```

```
##              estimate     2.5 %   97.5 %
## No_miss     1.0003582 0.9807380 1.019978
## MCAR        0.9779092 0.9270569 1.028761
## lightMAR    0.9768534 0.9228824 1.030824
## moderateMAR 0.9798801 0.9323819 1.027378
## extremeMAR  0.9841179 0.9433072 1.024929
```

```
true <- 1
RB <- rowMeans(MAR.res[,, "estimate"]) - true
PB <- 100 * abs((rowMeans(MAR.res[,, "estimate"]) - true)/ true)
CR <- rowMeans(MAR.res[,, "2.5 %"] < true & true < MAR.res[,, "97.5 %"])
AW <- rowMeans(MAR.res[,, "97.5 %"] - MAR.res[,, "2.5 %"])
RMSE <- sqrt(rowMeans((MAR.res[,, "estimate"] - true)^2))
data.frame(RB, PB, CR, AW, RMSE)
```

```
##                        RB         PB   CR         AW        RMSE
## No_miss      0.0003582023 0.03582023 0.95 0.03924041 0.009852852
## MCAR        -0.0220908076 2.20908076 0.97 0.10170455 0.026032486
## lightMAR    -0.0231466261 2.31466261 0.91 0.10794194 0.028181202
## moderateMAR -0.0201198748 2.01198748 0.91 0.09499644 0.026311825
## extremeMAR  -0.0158820761 1.58820761 0.90 0.08162145 0.023626457
```

```r
MNAR.create.data <- function(beta = 1, sigma2 = 1, n = 200,
                        run = 1) {
  set.seed(seed = run)
  x <- rnorm(n)
  y <- beta * x + rnorm(n, sd = sqrt(sigma2))
  as.data.frame(cbind(x = x, y = y))
}
```

```r
# Create missingness in x values greater than median with specified probability.
MNAR.make.missing <- function(data, prob_missing_higher = 0.2){
  higher <- data$x[data$x > median(data$x)]
  data$x[data$x > median(data$x)] = ifelse(sample(
    c(T, F), length(data$x[data$x > median(data$x)]), replace=T,
    prob=c(prob_missing_higher, 1 - prob_missing_higher)),
    NA,
    data$x[data$x > median(data$x)])
  data
}
```

```r
MNAR.test.impute <- function(data, m = 5) {
  imp <- mice(data, m = m, print = FALSE)
  fit <- with(imp, lm(y ~ x))
  tab <- summary(pool(fit), "all", conf.int = TRUE)
  as.numeric(tab[2, c("estimate", "2.5 %", "97.5 %")])
}
```

```r
simulate <- function(runs = 10) {
  res <- array(NA, dim = c(5, runs, 3))
  dimnames(res) <- list(c("lightest_MNAR", "light_MNAR", "moderate_MNAR",
                        "heavy_MNAR", "heaviest_MNAR"),
                        as.character(1:runs),
                        c("estimate", "2.5 %","97.5 %"))
  for(run in 1:runs) {
    data <- MNAR.create.data(run = run)
    lightest_data <- MNAR.make.missing(data, prob_missing_higher = 0.2)
    lighter_data <- MNAR.make.missing(data, prob_missing_higher = 0.4)
    moderate_data <- MNAR.make.missing(data, prob_missing_higher = 0.6)
    heavier_data <- MNAR.make.missing(data, prob_missing_higher = 0.8)
    heaviest_data <- MNAR.make.missing(data, prob_missing_higher = 1.0)
    res[1, run, ] <- MNAR.test.impute(lightest_data)
    res[2, run, ] <- MNAR.test.impute(lighter_data)
    res[3, run, ] <- MNAR.test.impute(moderate_data)
    res[4, run, ] <- MNAR.test.impute(heavier_data)
    res[5, run, ] <- MNAR.test.impute(heaviest_data)
    # print(paste("run", run,  "completed"))
  }
  res
}
```

```r
MNAR.res <- simulate(100)
```

MNAR

```
apply(MCAR.res, c(1, 3), mean, na.rm = TRUE)
```

Results

```
##      estimate    2.5 %   97.5 %
## MCAR 1.013443 0.845978 1.180909
```

```
true <- 1
RB <- mean(MCAR.res[,, "estimate"]) - true
PB <- 100 * abs((mean(MCAR.res[,, "estimate"]) - true)/ true)
CR <- mean(MCAR.res[,, "2.5 %"] < true & true < MCAR.res[,, "97.5 %"])
AW <- mean(MCAR.res[,, "97.5 %"] - MCAR.res[,, "2.5 %"])
RMSE <- sqrt(mean(MCAR.res[,, "estimate"] - true)^2)
data.frame(RB, PB, CR, AW, RMSE)
```

```
##           RB       PB CR        AW       RMSE
## 1 0.01344329 1.344329  1 0.3349306 0.01344329
```

```
apply(MAR.res, c(1, 3), mean, na.rm = TRUE)
```

```
##              estimate     2.5 %   97.5 %
## No_miss     1.0003582 0.9807380 1.019978
## MCAR        0.9779092 0.9270569 1.028761
## lightMAR    0.9768534 0.9228824 1.030824
## moderateMAR 0.9798801 0.9323819 1.027378
## extremeMAR  0.9841179 0.9433072 1.024929
```

```
true <- 1
RB <- rowMeans(MAR.res[,, "estimate"]) - true
PB <- 100 * abs((rowMeans(MAR.res[,, "estimate"]) - true)/ true)
CR <- rowMeans(MAR.res[,, "2.5 %"] < true & true < MAR.res[,, "97.5 %"])
AW <- rowMeans(MAR.res[,, "97.5 %"] - MAR.res[,, "2.5 %"])
RMSE <- sqrt(rowMeans((MAR.res[,, "estimate"] - true)^2))
data.frame(RB, PB, CR, AW, RMSE)
```

```
##                        RB         PB   CR         AW        RMSE
## No_miss      0.0003582023 0.03582023 0.95 0.03924041 0.009852852
## MCAR        -0.0220908076 2.20908076 0.97 0.10170455 0.026032486
## lightMAR    -0.0231466261 2.31466261 0.91 0.10794194 0.028181202
## moderateMAR -0.0201198748 2.01198748 0.91 0.09499644 0.026311825
## extremeMAR  -0.0158820761 1.58820761 0.90 0.08162145 0.023626457
```

```
apply(MNAR.res, c(1, 3), mean, na.rm = TRUE)
```

```
##                estimate     2.5 %   97.5 %
## lightest_MNAR 1.016957 0.8707358 1.163178
## light_MNAR    1.017404 0.8643218 1.170486
## moderate_MNAR 1.026153 0.8604851 1.191821
## heavy_MNAR    1.048536 0.8545357 1.242537
## heaviest_MNAR 1.314626 0.9111934 1.718059
```

```
true <- 1
RB <- rowMeans(MNAR.res[,, "estimate"]) - true
PB <- 100 * abs((rowMeans(MNAR.res[,, "estimate"]) - true)/ true)
CR <- rowMeans(MNAR.res[,, "2.5 %"] < true & true < MNAR.res[,, "97.5 %"])
```

```r
AW <- rowMeans(MNAR.res[,, "97.5 %"] - MNAR.res[,, "2.5 %"])
RMSE <- sqrt(rowMeans((MNAR.res[,, "estimate"] - true)^2))
data.frame(RB, PB, CR, AW, RMSE)
```

```
##                        RB        PB   CR        AW       RMSE
## lightest_MNAR 0.01695680  1.695680 0.97 0.2924419 0.06822474
## light_MNAR    0.01740392  1.740392 0.96 0.3061643 0.07345546
## moderate_MNAR 0.02615325  2.615325 0.95 0.3313363 0.08218141
## heavy_MNAR    0.04853644  4.853644 0.88 0.3880015 0.12828771
## heaviest_MNAR 0.31462625 31.462625 0.57 0.8068658 0.41423396
```