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

```
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)
}
```

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

```
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 %")])
}
```

```
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
}
```

```
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.

```
# 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.

```
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
}
```

```
# 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
```

```r
  # 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 %")])
}
```

```r
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
}
```

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

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

```r
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))
imp_measures <- data.frame(RB, PB, CR, AW, RMSE)
```

```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
}
```

```
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 %")])
}
```

```
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
}
```

```
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)
MCAR_imp_measures <- data.frame(RB, PB, CR, AW, RMSE)

MCAR_imp_measures
```

```
##           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(trunc((MAR.res[,, "97.5 %"] - MAR.res[,, "2.5 %"])*10^3)/10^3)
RMSE <- sqrt(rowMeans((MAR.res[,, "estimate"] - true)^2))
MAR_imp_measures <- data.frame(RB, PB, CR, AW, RMSE)
```

```
means <- as.data.frame(apply(MNAR.res, c(1, 3), mean, na.rm = TRUE))
```

```
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 %"])
AW <- rowMeans(trunc((MNAR.res[,, "97.5 %"] - MNAR.res[,, "2.5 %"])*10^3)/10^3)
RMSE <- sqrt(rowMeans((MNAR.res[,, "estimate"] - true)^2))
MNAR_imp_measures <- data.frame(RB, PB, CR, AW, RMSE)
MNAR_imp_measures
```

```
##                       RB        PB   CR      AW       RMSE
## lightest-MNAR 0.01695680  1.695680 0.97 0.29198 0.06822474
## light-MNAR    0.01740392  1.740392 0.96 0.30569 0.07345546
## moderate-MNAR 0.02615325  2.615325 0.95 0.33087 0.08218141
## heavy-MNAR    0.04853644  4.853644 0.88 0.38751 0.12828771
## heaviest-MNAR 0.31462625 31.462625 0.57 0.80633 0.41423396
```

```
raw_data <- rbind(MAR_imp_measures, MNAR_imp_measures[c(1, 3, 5),])
processed_raw_data <- raw_data %>% mutate(mean = 1 + RB, lower = mean - AW, upper = mean + AW)
processed_raw_data
```

```
##                           RB          PB   CR      AW        RMSE      mean
## No_miss        0.0003582023  0.03582023 0.95 0.03877 0.009852852 1.0003582
## MCAR          -0.0220908076  2.20908076 0.97 0.10122 0.026032486 0.9779092
## lightMAR      -0.0231466261  2.31466261 0.91 0.10741 0.028181202 0.9768534
## moderateMAR   -0.0201198748  2.01198748 0.91 0.09444 0.026311825 0.9798801
## extremeMAR    -0.0158820761  1.58820761 0.90 0.08111 0.023626457 0.9841179
## lightest-MNAR  0.0169568023  1.69568023 0.97 0.29198 0.068224740 1.0169568
```

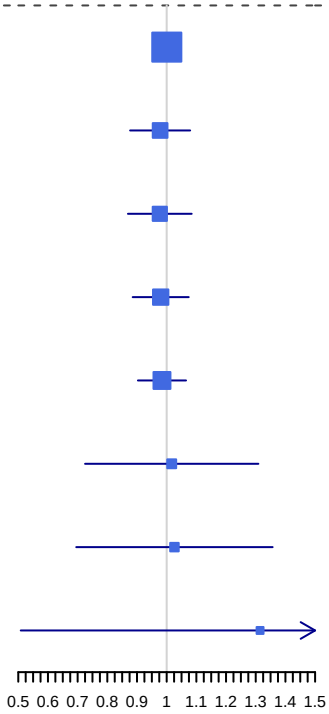
```
## moderate-MNAR  0.0261532452  2.61532452 0.95 0.33087 0.082181413 1.0261532
## heaviest-MNAR  0.3146262518 31.46262518 0.57 0.80633 0.414233961 1.3146263
##                    lower    upper
## No_miss        0.9615882 1.039128
## MCAR           0.8766892 1.079129
## lightMAR       0.8694434 1.084263
## moderateMAR    0.8854401 1.074320
## extremeMAR     0.9030079 1.065228
## lightest-MNAR 0.7249768 1.308937
## moderate-MNAR 0.6952832 1.357023
## heaviest-MNAR 0.5082963 2.120956
```

```
base_data <- processed_raw_data %>%
  mutate(study = c("Complete", "MCAR", "light-MAR", "moderate-MAR", "heavy-MAR", "light-MNAR", "moderat
         CR = as.character(CR),
         AW = as.character(AW)) %>%
  select(study, CR, AW, mean, lower, upper)



header <- tibble(study = c("Mechanism"),
                 CR = c("Coverage-Rate"),
                 AW = c("Average-Width"))

forest_data <- bind_rows(header, base_data)
forest_data
```

```
## # A tibble: 9 x 6
##   study         CR            AW              mean  lower upper
## * <chr>         <chr>         <chr>          <dbl>  <dbl> <dbl>
## 1 Mechanism     Coverage-Rate Average-Width NA     NA     NA
## 2 Complete      0.95          0.03877        1.00   0.962  1.04
## 3 MCAR          0.97          0.10122        0.978  0.877  1.08
## 4 light-MAR     0.91          0.10741        0.977  0.869  1.08
## 5 moderate-MAR  0.91          0.09444        0.980  0.885  1.07
## 6 heavy-MAR     0.9           0.08111        0.984  0.903  1.07
## 7 light-MNAR    0.97          0.29198        1.02   0.725  1.31
## 8 moderate-MNAR 0.95          0.33087        1.03   0.695  1.36
## 9 heavy-MNAR    0.57          0.80633        1.31   0.508  2.12
```

```
forest_data %>%
  forestplot(labeltext = c(study, CR, AW),
             # txt_gp =
             is.summary = FALSE,
             graph.pos = 2,
             hrzl_lines = list("2" = gpar(lty = 2)),
             clip = c(0.5, 1.5),
             zero = 1,
             col = fpColors(box = "royalblue",line = "darkblue", summary = "royalblue", hrz_lines = "#4
```

Mechanism	Coverage−Rate	Average−Width
Complete	0.95	0.03877
MCAR	0.97	0.10122
light−MAR	0.91	0.10741
moderate−MAR	0.91	0.09444
heavy−MAR	0.9	0.08111
light−MNAR	0.97	0.29198
moderate−MNAR	0.95	0.33087
heavy−MNAR	0.57	0.80633




```
{r} # knitr::kable(means, align = "lccrr") #
```

```
{r} # kable(imp_measures, caption = "MNAR performance measures")
#
```