# Exploration of the effect of missing data on statistical analysis

Leo Watson, Nathalie Moon

UNIVERSITY OF TORONTO

## Abstract

Analysis of **missing data mechanisms** and **modern approaches to handling missing data**. **Designing R simulations** to investigate hypotheses about imputation technique.

## Introduction

### Motivations

- Interested in what scenarios different imputation techniques should be used to reduce runtime without sacrificing bias, error, and other performance measures.
- Determine the types of missing data in the real world

### Definitions

#### Missing Data Mechanisms

**MCAR**: When probability of missingness for data points in a dataset is constant.

- Each student's mark is stored in a spreadsheet by the instructor but following a computer update 10% of the data is deleted at random.

**MAR**: When probability of missingness is dependent on some observed variable of the dataset.

- Most students joined a class from day 1, but some students joined late from the waitlist due to capacity restrictions. 10% of students who joined on time had a missing submission for the first problem set, while 30% of students who joined late missed the first problem set.

**MNAR**: When probability of missingness is dependent on the true value of the data point which we don't know for all subjects.

• Due to a system failure, the instructor loses all the students' marks. Left with no choice, the instructor requests the students to calculate and share their true final marks to the instructor. If they don't, the instructor will input that they got a B.

**–** If a student's true mark is an A, they are 90% likely to state their true mark. **–** If a student's true mark is a B, they are 70% likely to state their true mark. **–** If a student's true mark is a C, they are 50% likely to state their true mark.

#### Imputation Techniques

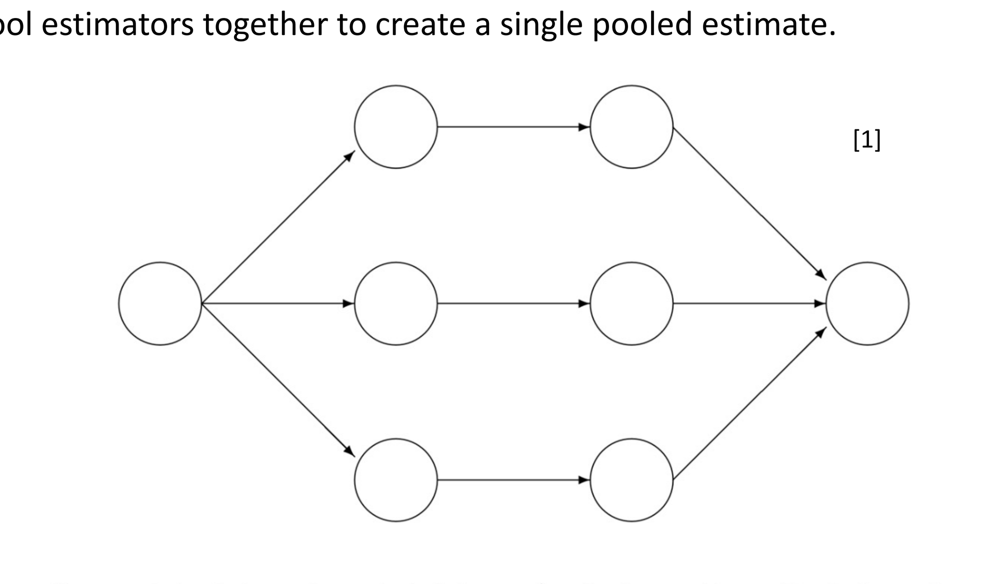
**Listwise Deletion**: Eliminates all observations containing ANY missing values in variables of interest

| Y | $X_1$ | $X_2$ | $X_3$ |
|---|---|---|---|
| 4 | 0.2 | 1.2 | 20 |
| 3 NA | | 1.2 | 21 |
| 2 | 0.3 | 1.1 | 16 |
| 2 | 0.4 | 1.1 | 17 |
| 1 | 0.5 | 2 | 18 |
| 2 | 0.4 | 2.1 | 18 |
| NA | 0.2 | 1.4 | 19 |
| 2 | 0.1 | 1.2 | 22 |
| 2 | 0.1 NA | | NA |

| Y | $X_1$ | $X_2$ | $X_3$ |
|---|---|---|---|
| 4 | 0.2 | 1.2 | 20 |
| 2 | 0.3 | 1.1 | 16 |
| 2 | 0.4 | 1.1 | 17 |
| 1 | 0.5 | 2 | 18 |
| 2 | 0.4 | 2.1 | 18 |
| 2 | 0.1 | 1.2 | 22 |
| – | – | – | – |
| – | – | – | – |
| – | – | – | – |

**Multiple Imputation**:

1. Takes incomplete dataset and creates multiple copies of it.
2. Impute incomplete columns with plausible values through an iterative predictive method for each copy
3. Obtain estimate for parameter of interest for each copy
4. Pool estimators together to create a single pooled estimate.




## INVESTIGATIONS

### 1) Comparing multiple imputation under varying degrees of MCAR, MAR, MNAR

### Simulation

```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)
}
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
}
```

**Simulate determining $\beta_1 = 1$**

1. MCAR: $y_i = x_i\beta_{1,} + \epsilon_i$
2. MAR: $y_i = x_{1,i}\beta_1 + x_{2,i}\beta_2 + \epsilon_i$
3. MNAR: $y_i = x_i\beta_{1,} + \epsilon_i$

```r
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
```

```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))
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
```

| | Estimate | PB | CR | AW |
|---|---|---|---|---|
| MCAR | 0.9779 | 2.209 | 0.97 | 0.102 |
| MAR-light | 0.9768 | 2.315 | 0.91 | 0.108 |
| MAR-moderate | 0.9799 | 2.011 | 0.91 | 0.095 |
| MAR-heavy | 0.9841 | 1.588 | 0.90 | 0.082 |
| MNAR-light | 1.0174 | 1.740 | 0.96 | 0.306 |
| MNAR-moderate | 1.0262 | 2.615 | 0.95 | 0.331 |
| MNAR-heavy | 1.0485 | 4.853 | 0.88 | 0.388 |

### 2) When Listwise Deletion Outperforms Multiple Imputation

Hypothesis 2a:
Missing Data only in Response *Y*

Simulation

Results

Hypothesis 2b:
Probability of missingness doesn't depend on *Y*

Simulation

Results

Hypothesis 2c:
Data follows Logistic Regression, probability of missingness depends only on *Y*

Simulation

Results

## CONCLUSION

## References

# Exploration of the effect of missing data on statistical analysis

Leo Watson, Nathalie Moon

UNIVERSITY OF TORONTO

## Abstract

Analysis of **missing data mechanisms** and **modern approaches to handling missing data**. **Designing R simulations** to investigate hypotheses about imputation technique.

## Introduction

### Motivations

- Interested in what scenarios different imputation techniques should be used to reduce runtime without sacrificing bias, error, and other performance measures.
- Determine the types of missing data in the real world

### Definitions

#### Missing Data Mechanisms

**MCAR**: When probability of missingness for data points in a dataset is constant.

- Each student's mark is stored in a spreadsheet by the instructor but following a computer update 10% of the data is deleted at random.

**MAR**: When probability of missingness is dependent on some observed variable of the dataset.

- Most students joined a class from day 1, but some students joined late from the waitlist due to capacity restrictions. 10% of students who joined on time had a missing submission for the first problem set, while 30% of students who joined late missed the first problem set.

**MNAR**: When probability of missingness is dependent on the true value of the data point which we don't know for all subjects.

• Due to a system failure, the instructor loses all the students' marks. Left with no choice, the instructor requests the students to calculate and share their true final marks to the instructor. If they don't, the instructor will input that they got a B.

– If a student's true mark is an A, they are 90% likely to state their true mark. – If a student's true mark is a B, they are 70% likely to state their true mark. – If a student's true mark is a C, they are 50% likely to state their true mark.

#### Imputation Techniques

**Listwise Deletion**: Eliminates all observations containing ANY missing values in variables of interest

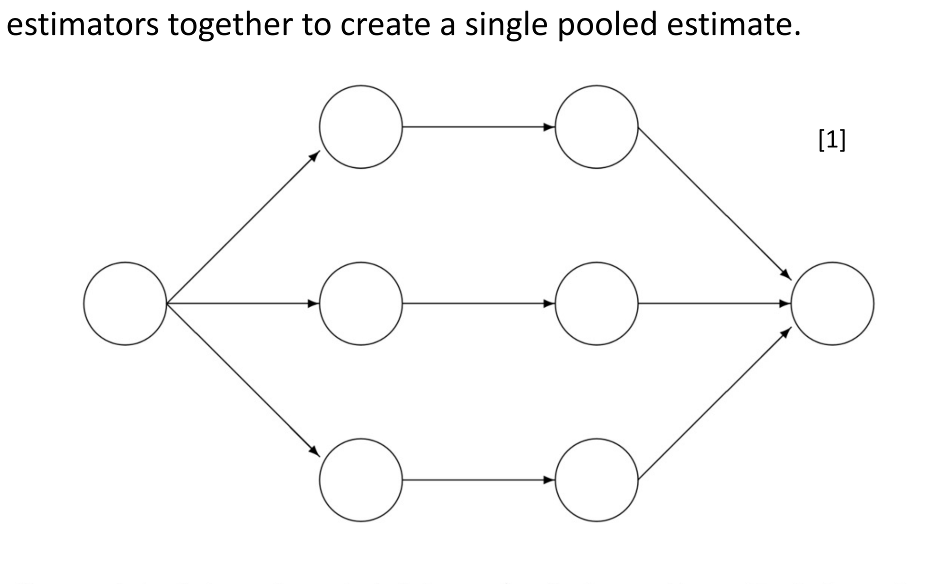
| Y | $X_1$ | $X_2$ | $X_3$ |
|---|---|---|---|
| 4 | 0.2 | 1.2 | 20 |
| 3 | NA | 1.2 | 21 |
| 2 | 0.3 | 1.1 | 16 |
| 2 | 0.4 | 1.1 | 17 |
| 1 | 0.5 | 2 | 18 |
| 2 | 0.4 | 2.1 | 18 |
| NA | 0.2 | 1.4 | 19 |
| 2 | 0.1 | 1.2 | 22 |
| 2 | 0.1 | NA | NA |

→

| Y | $X_1$ | $X_2$ | $X_3$ |
|---|---|---|---|
| 4 | 0.2 | 1.2 | 20 |
| 2 | 0.3 | 1.1 | 16 |
| 2 | 0.4 | 1.1 | 17 |
| 1 | 0.5 | 2 | 18 |
| 2 | 0.4 | 2.1 | 18 |
| 2 | 0.1 | 1.2 | 22 |
| – | – | – | – |
| – | – | – | – |
| – | – | – | – |

**Multiple Imputation**:

1. Takes incomplete dataset and creates multiple copies of it.
2. Impute incomplete columns with plausible values through an iterative predictive method for each copy
3. Obtain estimate for parameter of interest for each copy
4. Pool estimators together to create a single pooled estimate.




## INVESTIGATIONS

### 1) Multiple imputation under varying levels of MCAR, MAR, MNAR

```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)
}
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
 }
```

**Simulate determining $\boldsymbol{\beta_1 = 1}$**

1. MCAR: $y_i = x_i\beta_{1,} + \epsilon_i$
2. MAR: $y_i = x_{1,i}\beta_1 + x_{2,i}\beta_2 + \epsilon_i$
3. MNAR: $y_i = x_i\beta_{1,} + \epsilon_i$

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

##                 estimate      2.5 %   97.5 %
## No_miss        1.0003582 0.9807380 1.019978
MAR.res <- simulate(100)
## lightMAR     0.9768534 0.9228824 1.030824
## moderateMAR 0.9798801 0.9323819 1.027378
## extremeMAR  0.9841179 0.9433072 1.024929
```

```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))
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
```

| | Estimate | PB | CR | AW |
|---|---|---|---|---|
| MCAR | 0.9779 | 2.209 | 0.97 | 0.102 |
| MAR-light | 0.9768 | 2.315 | 0.91 | 0.108 |
| MAR-moderate | 0.9799 | 2.011 | 0.91 | 0.095 |
| MAR-heavy | 0.9841 | 1.588 | 0.90 | 0.082 |
| MNAR-light | 1.0174 | 1.740 | 0.96 | 0.306 |
| MNAR-moderate | 1.0262 | 2.615 | 0.95 | 0.331 |
| MNAR-heavy | 1.0485 | 4.853 | 0.88 | 0.388 |

### (2) When Listwise Deletion Outperforms Multiple Imputation

#### Case 1: Missing Data only in Response *Y*

**Determining $\boldsymbol{\beta_1 = 1}$, $\boldsymbol{\beta_2}$ = 2, $\boldsymbol{\beta_3}$ = 3, $\boldsymbol{\beta_4}$ = 4** in model $y_i = x_{1,i}\beta_1 + x_{2,i}\beta_2 + x_{3,i}\beta_3 + x_{4,i}\beta_4 + \epsilon_i$

| | Y | X_1 | X_2 | X_3 | X_4 |
|---|---|---|---|---|---|
| 1 | NA | -0.96193342 | 0.726838902 | 0.61735005 | -1.29134932 |
| 2 | NA | -0.29252572 | -0.809440902 | -0.40507751 | 2.63504537 |
| 3 | 26.986093 | 0.25878822 | 0.267085116 | 1.05310376 | 0.48707229 |
| 4 | NA | -1.15213189 | -1.737263711 | 0.60228425 | 0.85389232 |
| 5 | 24.321907 | 0.19578283 | -1.411425136 | 1.01746118 | 1.08844271 |

Multiple Imputation

```r
##                     RB         PB   CR        AW      RMSE
## Intercept -0.004632882 0.02316441 1.00 0.5360155 0.0955964
## X_1       -0.063330448 6.33304482 0.96 0.5788813 0.1180301
## X_2       -0.085072564 4.25362822 0.97 0.5297375 0.1353696
## X_3       -0.202638162 6.75460540 0.64 0.5033990 0.2373411
## X_4       -0.103835898 2.59589744 0.90 0.4892921 0.1475618
times <- res_MI2[[2]]
mean(times)

## [1] 0.07354221
```

Listwise Deletion

```r
##                   RB         PB   CR        AW       RMSE
## Intercept 0.01412967 0.07064837 1.00 0.3649604 0.04946551
## X_1       0.02335495 2.33549497 1.00 0.3977020 0.05633452
## X_2       0.06959586 3.47979321 0.98 0.3617599 0.08869656
## X_3       0.02198526 0.73284197 1.00 0.3417499 0.04601087
## X_4       0.03357291 0.83932276 1.00 0.3353844 0.05205810
times_LD <- result_LD[[2]]
mean(times_LD)

## [1] 0.007305567
```

#### Case 2: Missing Data independent of response *Y*

**Determining $\boldsymbol{\beta_1 = 1}$, $\boldsymbol{\beta_2}$ = 2, $\boldsymbol{\beta_3}$ = 3, $\boldsymbol{\beta_4}$ = 4** in model $y_i = x_{1,i}\beta_1 + x_{2,i}\beta_2 + x_{3,i}\beta_3 + x_{4,i}\beta_4 + \epsilon_i$

| | Y | X_1 | X_2 | X_3 | X_4 |
|---|---|---|---|---|---|
| 95 | 18.06782258 | 0.558486426 | 0.134447661 | -1.10891000 | 0.14510663 |
| 96 | 27.50100053 | NA | 0.128855402 | 0.57562029 | 1.08715025 |
| 97 | 21.65300743 | -0.461644730 | 0.099078487 | -0.93590256 | 1.29675557 |
| 98 | 11.33108355 | 0.716707476 | 0.082965734 | NA | -1.52315577 |
| 99 | 12.44660145 | -1.015847465 | 0.045010598 | -0.43610692 | -1.22499116 |
| 100 | 21.17633208 | NA | 0.004398704 | NA | NA |
| 101 | 12.46882475 | NA | 0.002131860 | NA | NA |
| 102 | 23.94489455 | NA | -0.005344028 | NA | NA |
| 103 | 18.02430387 | NA | -0.008309014 | NA | NA |
| 104 | 33.60898116 | NA | -0.013399523 | NA | NA |

Multiple Imputation

```r
##                   RB        PB   CR        AW      RMSE
## Intercept  0.13268692 0.6634346 0.95 1.1691412 0.2328769
## X_1        0.09318012 9.3180118 0.98 0.7916511 0.1608131
## X_2       -0.02529328 1.2646638 0.97 1.1818267 0.1804150
## X_3        0.07662874 2.5542914 0.97 0.6610958 0.1331688
## X_4        0.04258306 1.0645764 0.98 0.6218411 0.1099079
times <- res_MI2[[2]]
mean(times)

## [1] 0.191239
```

Listwise Deletion

```r
##                   RB        PB   CR        AW       RMSE
## Intercept  0.11429526 0.5714763 0.98 1.0689402 0.21563689
## X_1        0.06477740 6.4777396 1.00 0.6403192 0.13364648
## X_2       -0.03368863 1.6844316 1.00 1.0811044 0.17563507
## X_3        0.01466790 0.4889301 1.00 0.5801619 0.09709261
## X_4       -0.01747858 0.4369644 1.00 0.5405804 0.08590829
times_LD <- result_LD[[2]]
mean(times_LD)

## [1] 0.007702417
```

#### Case 3: Logistic regression model & probability to be missing depends *only* on *Y*

**Simulate determining $\boldsymbol{\beta_1 = 1}$, $\boldsymbol{\beta_2}$ = 2** in logistic regression model. MNAR missingness where *Y = 0* observations have greater missingness probability for predictors than *Y = 1* cases

| | Y | x1 | x2 |
|---|---|---|---|
| 98 | 1 | 1.10992029 | 0.45623640 |
| 99 | 1 | -0.49929202 | NA |
| 100 | 1 | 0.09458353 | 0.88425082 |
| 101 | 1 | NA | 1.50390061 |
| 102 | 1 | -1.25127136 | 0.84573154 |
| 103 | 0 | NA | NA |
| 104 | 0 | 1.71506499 | -0.47624689 |
| 105 | 0 | 0.46091621 | -0.78860284 |
| 106 | 0 | NA | NA |
| 107 | 0 | NA | NA |

Multiple Imputation

```r
##                   RB       PB   CR        AW      RMSE
## Intercept 0.07608593      Inf 0.99 0.8060835 0.1511027
## x1        0.11169267 11.16927 0.98 1.4481633 0.2403467
## x2        0.21261059 10.63053 0.98 1.7733321 0.3379104
times <- res_MI2[[2]]
mean(times)

## [1] 0.2123698
```

Listwise Deletion

```r
##                   RB        PB   CR        AW      RMSE
## Intercept 0.92154738       Inf 0.00 0.5545185 0.9290863
## x1        0.14804061 14.804061 0.96 0.6491966 0.1979282
## x2        0.05219365  2.609682 1.00 0.8593025 0.1655004
times_LD <- result_LD[[2]]
mean(times_LD)

## [1] 0.007046299
```

### Investigation (2) Summarized Results:

| | Listwise Del (s) | Multiple Imp (s) | Rate |
|---|---|---|---|
| Case 1 | 0.00731 | 0.07354 | 10x faster |
| Case 2 | 0.00770 | 0.19124 | ~25x faster |
| Case 3 | 0.00705 | 0.21237 | ~30x faster |

```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
}
```

| | Estimate | PB | CR | AW |
|---|---|---|---|---|
| MCAR | 0.9779 | 2.209 | 0.97 | 0.102 |
| MAR-light | 0.9768 | 2.315 | 0.91 | 0.108 |
| MAR-moderate | 0.9799 | 2.011 | 0.91 | 0.095 |
| MAR-heavy | 0.9841 | 1.588 | 0.90 | 0.082 |
| MNAR-light | 1.0174 | 1.740 | 0.96 | 0.306 |
| MNAR-moderate | 1.0262 | 2.615 | 0.95 | 0.331 |
| MNAR-heavy | 1.0485 | 4.853 | 0.88 | 0.388 |

```r
create.data <- function(alpha = 20, beta_1 = 1, beta_2 = 2, beta_3 = 3, beta_4 = 4,
                        sigma2 = 1, n = 50, run = 1) {
  set.seed(seed = run)
  x_1 <- rnorm(n)
  x_2 <- rnorm(n)
  x_3 <- rnorm(n)
  x_4 <- rnorm(n)
  y <- beta_1 * x_1 + beta_2 * x_2 + beta_3 * x_3 + beta_4 * x_4 + alpha +
    rnorm(n, sd = sqrt(sigma2))
  cbind("Y" = y, "X_1"= x_1, "X_2" = x_2, "X_3" = x_3, "X_4" = x_4)
}
```

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

```r
simulate_MI2 <- function(runs = 100) {
  res <- array(NA, dim = c(5, runs, 3))
  times <- array(NA, dim = c(100, 1, 1))
  dimnames(res) <- list(c("Intercept", "X_1", "X_2", "X_3", "X_4"),
                        as.character(1:runs), c("estimate", "2.5%", "97.5%"))
  sim_dataset <- as.data.frame(create.data(n = 200))
  for (run in 1:runs){
      # Note that time is only measured for the MI/imp steps
      # (i.e. filtering, predicting)
    missingness_sim_dataset <- MCAR.make.missing(sim_dataset, p = 0.5)
    start_time <- Sys.time()
    imp_MI <- mice(missingness_sim_dataset, print = FALSE)
```

```r
    fit <- with(imp_MI, lm(Y ~ X_1 + X_2 + X_3 + X_4))
    end_time <- Sys.time()
    tab <- summary(pool(fit), "all", conf.int = TRUE)
    res[1, run, ] <- as.numeric(tab[1, c("estimate", "2.5 %", "97.5 %")])
    res[2, run, ] <- as.numeric(tab[2, c("estimate", "2.5 %", "97.5 %")])
    res[3, run, ] <- as.numeric(tab[3, c("estimate", "2.5 %", "97.5 %")])
    res[4, run, ] <- as.numeric(tab[4, c("estimate", "2.5 %", "97.5 %")])
    res[5, run, ] <- as.numeric(tab[5, c("estimate", "2.5 %", "97.5 %")])

    times[run, 1, 1] <- as.numeric(end_time - start_time)
  }
  list(res, times)
}
```

```r
result_LD <- simulate_LD()
```

```r
# Obtain confidence intervals & estimates for all coefficients, intercept.
apply(result_LD[[1]], c(1, 3), mean, na.rm = TRUE)
```

```
##            estimate       2.5%     97.5%
## Intercept 20.014130 19.8316495 20.196610
## X_1        1.023355  0.8245039  1.222206
## X_2        2.069596  1.8887159  2.250476
## X_3        3.021985  2.8511103  3.192860
## X_4        4.033573  3.8658807  4.201265
```

```r
# Evaluating imputation method performance for estimating
# all parameters of interest.
res <- res_MI2[[1]]
true <- c(20, 1, 2, 3, 4)
RB <- rowMeans(res[,, "estimate"]) - true
PB <- 100 * abs((rowMeans(res[,, "estimate"]) - true)/ true)
CR <- rowMeans(res[,, "2.5%"] < true & true < res[,, "97.5%"])
AW <- rowMeans(res[,, "97.5%"] - res[,, "2.5%"])
RMSE <- sqrt(rowMeans((res[,, "estimate"] - true)^2))
data.frame(RB, PB, CR, AW, RMSE)
```

```
##                     RB         PB   CR        AW      RMSE
## Intercept -0.004632882 0.02316441 1.00 0.5360155 0.0955964
## X_1       -0.063330448 6.33304482 0.96 0.5788813 0.1180301
## X_2       -0.085072564 4.25362822 0.97 0.5297375 0.1353696
## X_3       -0.202638162 6.75460540 0.64 0.5033990 0.2373411
## X_4       -0.103835898 2.59589744 0.90 0.4892921 0.1475618
```

```
## X_3       0.02198526 0.73284197 1.00 0.3417499 0.04601087
## X_4       0.03357291 0.83932276 1.00 0.3353844 0.05205810
```

```r
# Mean time for 100 instances of LD
times_LD <- result_LD[[2]]
mean(times_LD)
```

```
## [1] 0.007305567
```

```r
# Mean time for the multiple imputation instances
times <- res_MI2[[2]]
mean(times)
```

```
## [1] 0.07354221
```