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

MAR

MNAR

Imputation Techniques

Listwise Deletion

Multiple Imputation

Investigations

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

Simulation

```
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: $y_i = x_i\beta_{1,} + \epsilon_i$

```
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
```

	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

```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