Missing Data Mechanisms & Imputation Methods

2022-06-28, Leo Murao Watson

Introduction

In this write-up, I will discuss the missing data mechanisms of MCAR, MAR, MNAR as well as the imputation methods of listwise deletion, pairwise deletion, mean imputation, stochastic/deterministic regression imputation, LOCF/BOCF, the missingness indicator method, and multiple imputation.

Missing Data Mechanisms

MCAR (Missing Completely at Random)

- A **MCAR** missing data mechanism is observed when missingness is some constant ϕ for all observations in the sample ([1]).
- Mathematically: $P(R_i|Y_i) = P(R_i) = \phi$ ([1]). Here, Y_i is the response for the i^{th} individual in our sample, R_i is an indicator for missingness ($R_i = 0$ if Y_i is not missing, $R_i = 1$ if Y_i is missing), ϕ is some parameter of the missing data model.
  - Note that covariates $(X_1, ..., X_n)$ may also be under a MCAR mechanism with anaolgous notation.

Example:

- 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 (Missing at Random)

- A **MAR** mechanism is observed when missingness is dependent on some observed variable of the data. In this case, the probability of missingness is dependent on groups defined by the data ([2]).
- Mathematically: $P(R_i|Y_i) = P(R_i|Y_{i,o})$. ([3]) I.e probability of missingness for the Y_i observation is dependent on some characteristic of the observed data for that observation.

Example

- 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.
  - Mathematically, can think of each student as an observation. Let whether they joined from the waitlist or not be binary covariate X_1, and first problem set submission be binary response Y. X_1 determines the probability that a student's first problem set is missing, hence this is a MAR mechanism example.

MNAR

- A **MNAR** mechanism is observed when missingness is dependent on the true value of the data point which we don't know for all subjects.
- Mathematically: $P(R_i|Y_i) \neq P(R_i|Y_{i,o})$ ([3]). I.e. probability of missingness doesn't depend on observed data; it depends on unobserved data which we don't know!

Example:

- Due to a catastrophic system failure, the spreadsheet corrupts causing the instructor to lose 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.

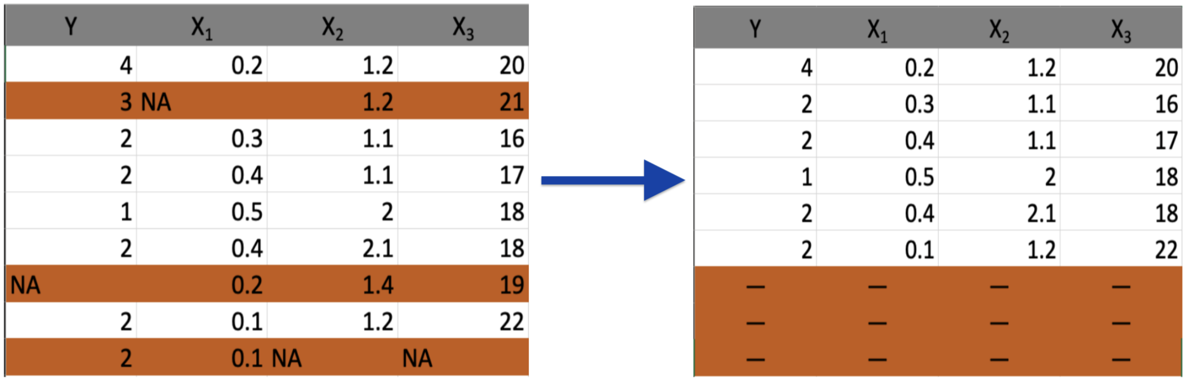
Imputation Methods

Listwise Deletion (Complete-Case Analysis)

Description

- Eliminates all observations containing ANY missing values in variables of interest ([1]).

Example

- 

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
–	–	–	–
–	–	–	–
–	–	–	–

Pros

- Highly convenient and computationally cheap.
- Robust model due to studying purely complete-cases.
- Unbiased estimates of means, variances, and regression coefficients under MCAR missingness ([1]).
- Can be efficient and exceedingly effective in specific scenarios:
  - (1): If probability to be missing does not depend on response Y; missing data rate may depend on any predictors X_i, with missing data in both Y/X allowed ([1]).
  - (2): If complete data model is logistic regression. If missing data is confined to one of Y or X_i, regression coefficients are unbiased if missingness depends only on Y and not on X_i ([1]).

Cons

- Wasteful, especially in cases where observations are only missing a small fraction of variable values. May be computationally cheap, but wasteful of expensive data.

- If data isn't MCAR, generates severely biased estimates of means, variances, and regression coefficients ([1]).

Pairwise Deletion

Description

- Pairwise Deletion seeks to utilize ALL available data.
- First, pairwise deletion calculates means of variables using observed data for each variable ([1]).
- Then, combines this with correlations/covariances between the different variables in the dataset to impute values ([1]).

```r
# First inspect initial dataset
data <- airquality[, c("Ozone", "Solar.R", "Wind")]
head(data)
```

Example

```
##   Ozone Solar.R Wind
## 1    41     190  7.4
## 2    36     118  8.0
## 3    12     149 12.6
## 4    18     313 11.5
## 5    NA      NA 14.3
## 6    28      NA 14.9
```

```
# Compute means for each column using observed values
mu <- colMeans(data, na.rm = TRUE)
mu
```

```
##      Ozone    Solar.R       Wind
##  42.129310 185.931507   9.957516
```

```
# Compute covariances for each variable, generating a covariance matrix.
cv <- cov(data, use = "pairwise")
cv
```

```
##              Ozone    Solar.R      Wind
## Ozone   1088.20052 1056.58346 -70.93853
## Solar.R 1056.58346 8110.51941 -17.94597
## Wind     -70.93853  -17.94597  12.41154
```

```
# Apply pairwise deletion using the Lavaan package to obtain our estimators: coefficients for
# Ozone w.r.t Wind, Solar.R in regression model.
fit <- lavaan("Ozone ~ 1 + Wind + Solar.R
              Ozone ~~ Ozone",
             sample.mean = mu, sample.cov = cv,
             sample.nobs = sum(complete.cases(data)))
help(fit)
coef(fit)[c(2:3)]
```

```
##    Ozone~Wind Ozone~Solar.R
##    -5.5449074     0.1180041
```

Pros

- Pairwise deletion works well if data $(Y, X_1, ..., X_n)$ approximately follows a multivariate normal distribution and the correlations between variables are low ([1]).
- If MCAR is plausible, provides unbiased estimates effectively and efficiently ([1]).

Cons

- Requires MCAR assumption for unbiased estimates.
- Correlation matrix obtained via pairwise deletion may give mathematically impossible results, and hence unable to perform further regression analysis ([4]). [see code below]
  - Consider 4x3 below data frame with some missingness

```
df <- data.frame(X1 = c(3,NA,2,4),
                 X2 = c(4,1,2,NA),
                 X3 = c(4,1,NA,3))
df
```

```
##   X1 X2 X3
## 1  3  4  4
## 2 NA  1  1
## 3  2  2 NA
## 4  4 NA  3
```

- Note that X_1, X_2 are perfectly correlated, X_2, X_3 are perfectly correlated and YET X_1, X_3 are perfectly negatively correlated. This is a physically impossible scenario.

```
# Creates correlation matrix
cor(df, use=  "pairwise.complete.obs")
```

```
##    X1 X2 X3
## X1  1  1 -1
## X2  1  1  1
## X3 -1  1  1
```

- Moreover, the covariance/correlation matrix may not be positive definite; this is a requirement for common multivariable procedures and further analysis, e.g. MANOVA: multivariate analysis of variance ([5]).
- The sample size will almost certainly vary between variables due to missingness. This is problematic for calculating standard error $= \frac{\sigma}{n}$ as the n value is ambiguous and hence standard error will almost certainly be underestimated/overestimated ([4]).

Mean Imputation

Description

- For each variable in the dataset, replaces missing values with mean of the observed values ([1]).
- Similar to pairwise deletion but without any correlation/covariance calculations.

Example

Get a sense of data of interest (note missingness).

```
head(airquality, n =10) %>% select(Ozone)
```

```
##    Ozone
## 1     41
## 2     36
## 3     12
## 4     18
## 5     NA
## 6     28
## 7     23
## 8     19
## 9      8
## 10    NA
```

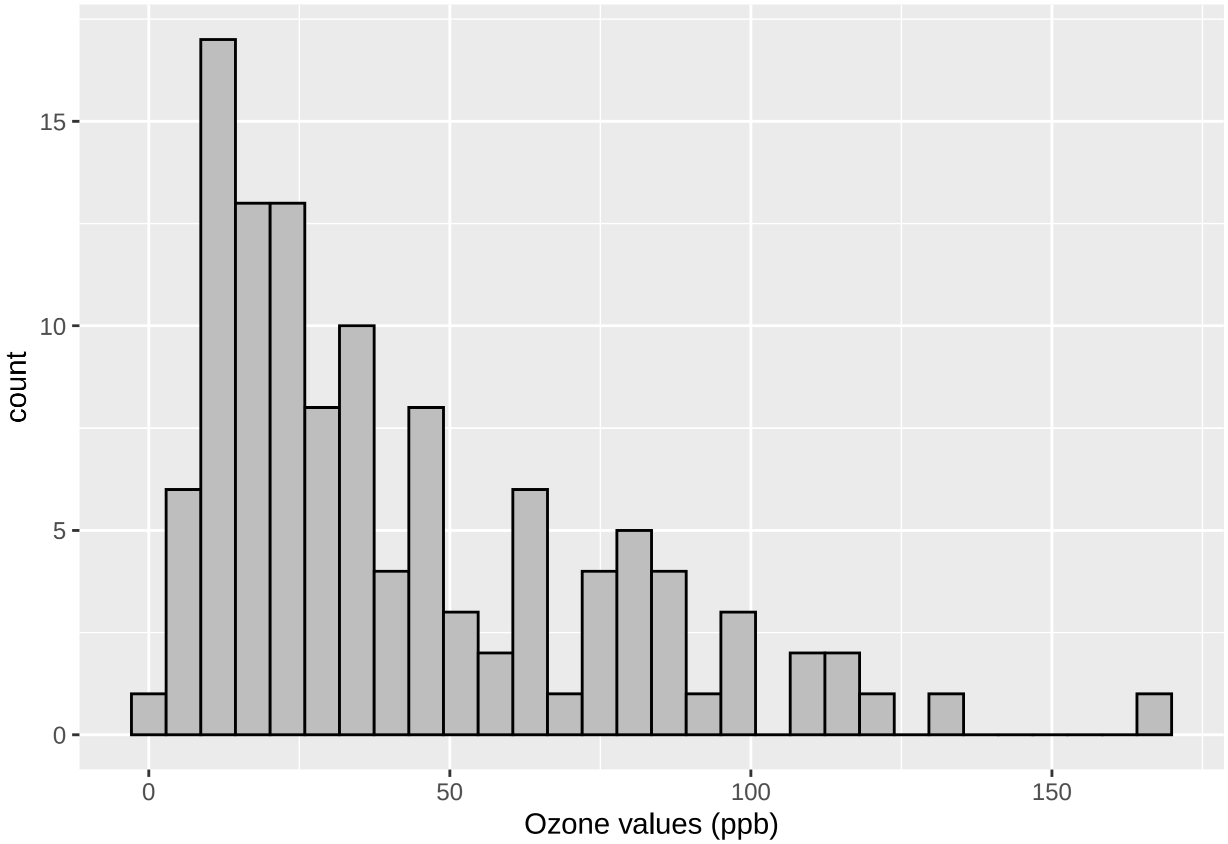
Compute mean of (observed) Ozone values.

```
mean_ozone <- mean(airquality$Ozone, na.rm = TRUE)
mean_ozone
```

```
## [1] 42.12931
```

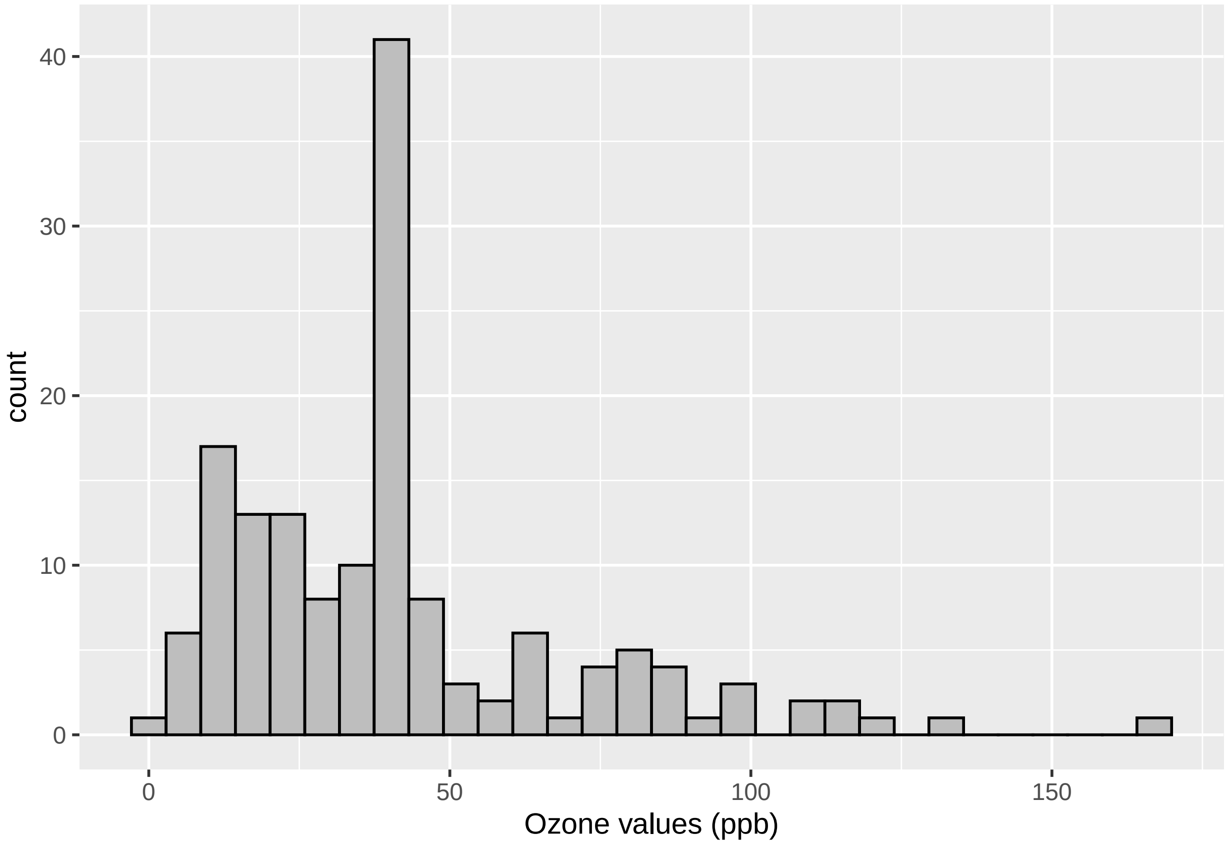
Create histogram of observed values.

```
ggplot(data = airquality, aes(x = Ozone)) +
  geom_histogram(color = "black",
                 fill = "gray",
                 bins = 30) +
  labs(x = "Ozone values (ppb)")
```



Impute missing values and create histogram of complete dataset.

```
imputed <- data.frame(airquality$Ozone)
imputed[is.na(imputed)] <- mean(airquality$Ozone, na.rm = TRUE)
ggplot(data = imputed, aes(x = airquality.Ozone)) +
  geom_histogram(color = "black",
                 fill = "gray",
                 bins = 30) +
  labs(x = "Ozone values (ppb)")
```



Note that the imputed values all equal the mean of 42.129, hence the large peak in the above histogram at that value. The distribution is now bimodal and the standard deviation has decreased from ~33.0 to ~28.7 as shown below.

```
sd(airquality$Ozone, na.rm = TRUE)
```

```
## [1] 32.98788
```

```
sd(imputed$airquality.Ozone)
```

```
## [1] 28.69337
```

Pros

- Conceptually straightforward.
- Computationally inexpensive.

Cons

- Can heavily distort distribution (e.g. unimodal distribution becomes bimodal like in example above)
- Underestimates the variance due to imputed values having zero contribution to variance ([1]).
- Biases almost all estimators irrespective of missing data mechanism ([1]).
- Even biases the mean estimator $\bar{X}$ if not MCAR ([1]).

(Deterministic) Regression Imputation

Description

- First, fits a regression model between observed values of predictor variables $(X_1, ..., X_n)$ and response variable Y.
- Then, imputes NA values with the exact predicted values from regression model ([1]).
  - Note that this method can be be used to impute missing values in predictor(s) X_i by fitting a regression model between observed $(X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n, Y)$ and X_i.

Example

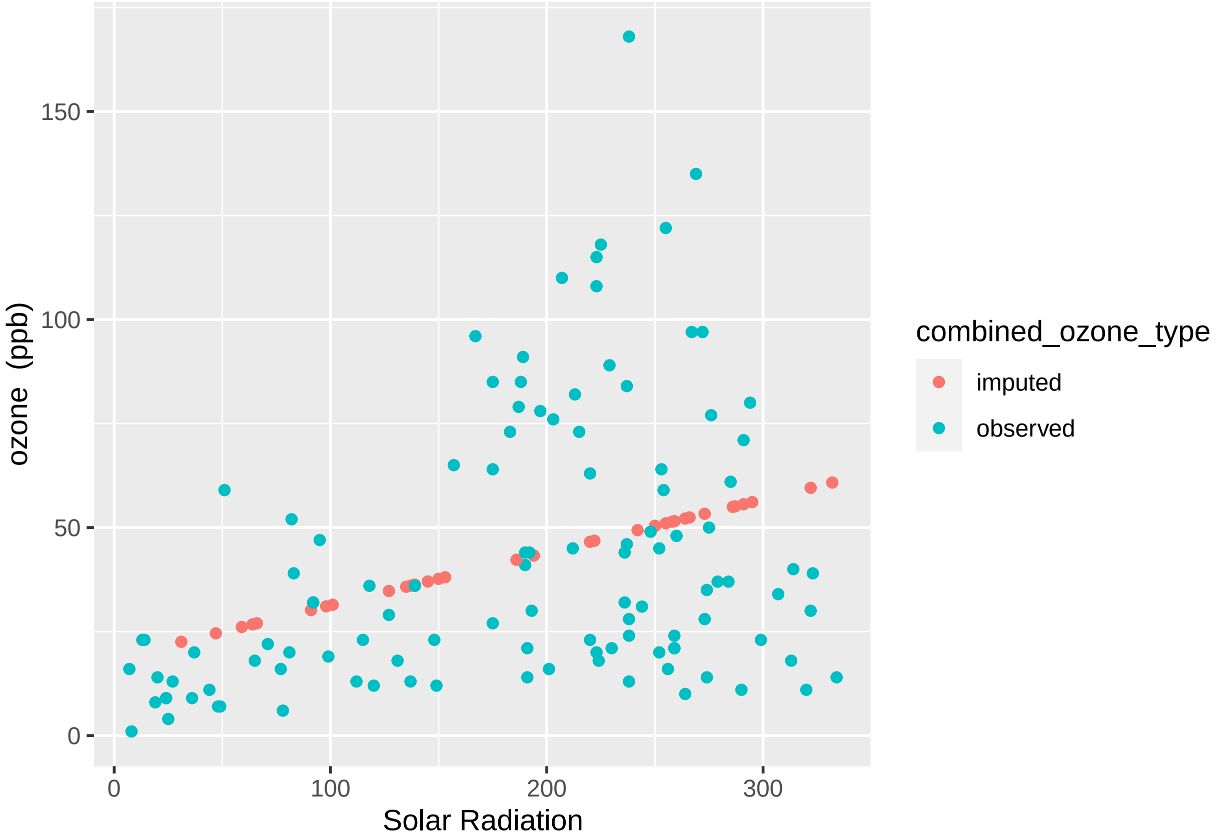
```
# Fit linear regression model to Ozone w.r.t Solar.R from airquality built-in dataset.
fit <- lm(Ozone ~ Solar.R, data = airquality)
# Applies fit to dataset.
pred <- predict(fit, airquality %>% select(Ozone, Solar.R))
pred_df <- as.data.frame(pred)
```

```
# Create new variable "model_ozone" that takes exact value given the fitted model,
# and new variable "combined_ozone" that takes observed "Ozone" value if it's not missing
# "model_ozone" value if missing.
airquality_processed <- airquality %>%
  select(Ozone, Solar.R) %>%
  mutate(model_ozone = pred_df$pred) %>%
  mutate(combined_ozone = if_else(is.na(Ozone), model_ozone, as.numeric(Ozone)
                                  , missing = NULL)) %>%
  mutate(combined_ozone_type = if_else(is.na(Ozone), "imputed", "observed"
                                       , missing = NULL)) %>%
  relocate(Solar.R, .after = combined_ozone) %>%
  relocate(combined_ozone)
head(airquality_processed, n = 25)
```

```
##    combined_ozone Ozone model_ozone Solar.R combined_ozone_type
## 1        41.00000    41    42.76013     190            observed
## 2        36.00000    36    33.60423     118            observed
## 3        12.00000    12    37.54635     149            observed
## 4        18.00000    18    58.40146     313            observed
## 5              NA    NA          NA      NA             imputed
## 6        28.00000    28          NA      NA            observed
## 7        23.00000    23    56.62114     299            observed
## 8        19.00000    19    31.18809      99            observed
## 9         8.00000     8    21.01487      19            observed
## 10       43.26879    NA    43.26879     194             imputed
## 11        7.00000     7          NA      NA            observed
## 12       16.00000    16    51.15304     256            observed
## 13       11.00000    11    55.47666     290            observed
## 14       14.00000    14    53.44201     274            observed
## 15       18.00000    18    26.86447      65            observed
## 16       14.00000    14    61.07193     334            observed
## 17       34.00000    34    57.63847     307            observed
## 18        6.00000     6    28.51762      78            observed
## 19       30.00000    30    59.54595     322            observed
## 20       11.00000    11    24.19400      44            observed
## 21        1.00000     1    19.61605       8            observed
## 22       11.00000    11    59.29161     320            observed
## 23        4.00000      4     21.77786       25              observed
## 24       32.00000     32     30.29793       92              observed
## 25       26.99164     NA     26.99164       66               imputed
```

- Note the imputed values in the scatter plot below are exactly the predicted values from the regression fit with no added randomness. This increases the correlation and decreases variance, as shown in the next chunks.

```
airquality_processed %>%
  ggplot(aes(x = Solar.R,y= combined_ozone, color = combined_ozone_type)) +
  geom_point() +
  labs(y = "ozone  (ppb)",
       x = "Solar Radiation")
```



```
# Correlation for observed values
cor.test(airquality_processed$Ozone, airquality_processed$Solar.R)$estimate
```

```
##       cor
## 0.3483417
```

```
# Correlation for observed/imputed values
cor.test(airquality_processed$combined_ozone, airquality_processed$Solar.R)$estimate
```

```
##       cor
## 0.3884392
```

```
# Variance for observed values
var(airquality_processed$Ozone, airquality_processed$Solar.R, na.rm = TRUE)
```

```
## [1] 1056.583
```

```
# Variance for observed/imputed values
var(airquality_processed$combined_ozone, airquality_processed$Solar.R, na.rm = TRUE)
```

```
## [1] 1031.376
```

Pros

- Yields unbiased mean estimates under MCAR, and regression weights of imputation model if the explanatory variables are complete ([1]).
- Yields unbiased estimators for regression weights under MAR, if the variable causing the MAR is part of the regression model ([1]); (e.g. if Solar.R was responsible for missingness in Ozone in the above example)

Cons

- Since imputed values are the exact predicted values from the regression model, correlation is overestimated and variability is underestimated ([6]).
- Imputations are realistic if predicted linear model is almost perfect, but deterministic regression is often too good to be true as the uncertainty in the imputed values is not accounted for ([1]).

Stochastic Regression Imputation

Description

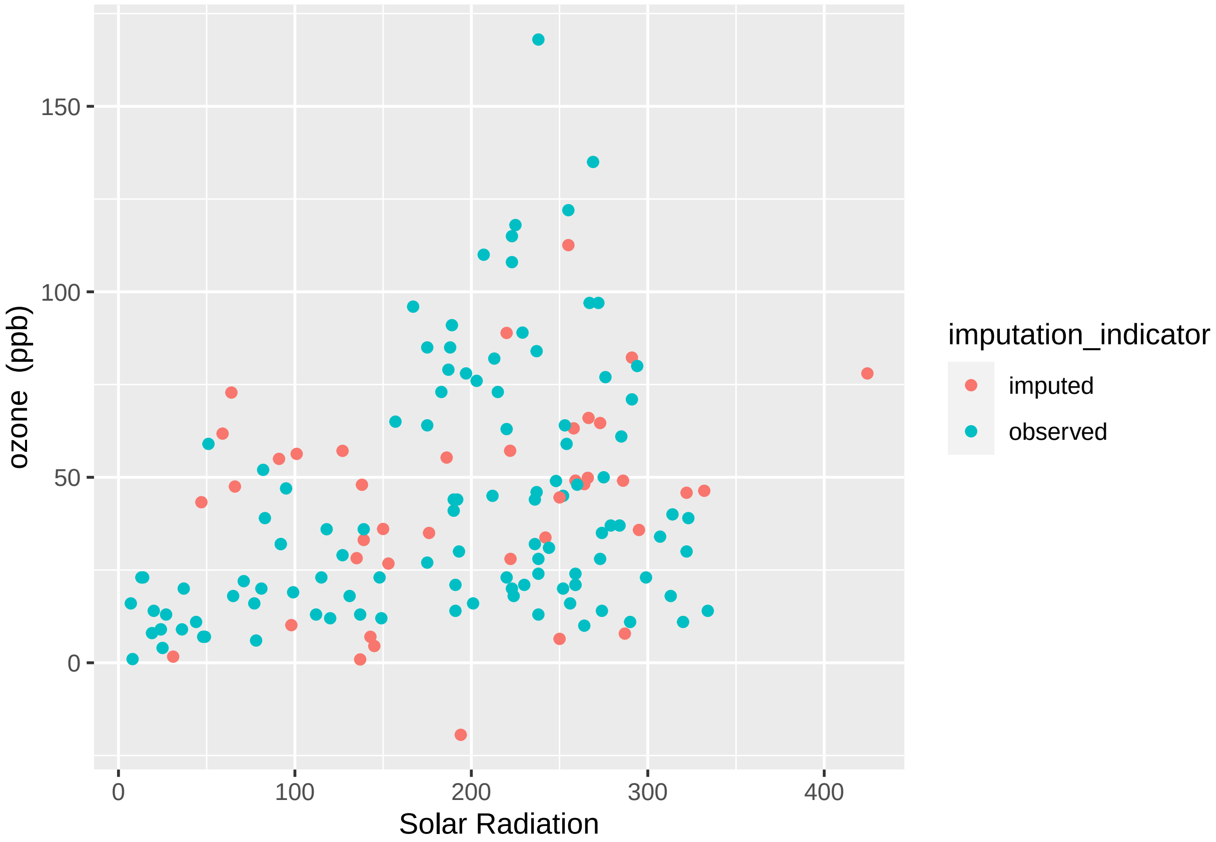
- Very similar to deterministic regression imputation but as an extra step adds noise (random draw from residual) to predictions to account for the uncertainty in imputation ([1]).

Example

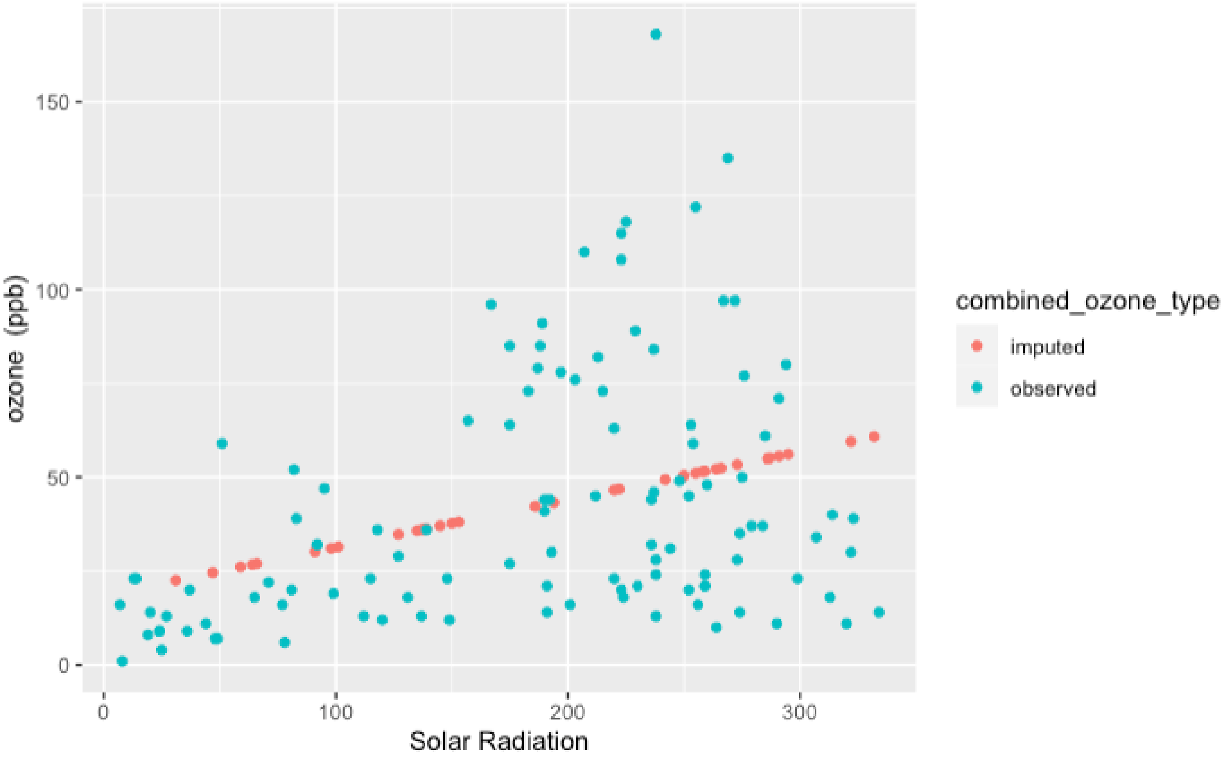
```
# remove rows with no information, add indicator variable for whether a row requires
# imputation or not.
data <- airquality[, c("Ozone", "Solar.R")]  %>%
  filter(!is.na(Ozone) | !is.na(Solar.R)) %>%
  mutate(imputed_or_not = if_else(is.na(Ozone) | is.na(Solar.R) , "imputed", "observed"))

# "method = norm.nob" applies imputation to "data" (w/o indicator) using stochastic
# regression. The other parameters in mice() aren't relevant for this section.
imp <- mice(data %>% select(Ozone, Solar.R), method = "norm.nob", m = 1, maxit = 1,
            seed = 1, print = FALSE)
imputed_airquality_dataset <- complete(imp) %>%
  add_column(data$imputed_or_not) %>%
  rename(imputation_indicator = "data$imputed_or_not")
```

```
imputed_airquality_dataset %>%
  ggplot(aes(x = Solar.R,y= Ozone, color = imputation_indicator)) +
  geom_point() +
  labs(y = "ozone  (ppb)",
       x = "Solar Radiation")
```



Recall the deterministic regression imputation plot (below for convenience) and note the random noise added to imputed values above:



Pros

- Unlike deterministic regression imputation, preserves correlation between variables ([6]).
- Initially counterintuitive to add random noise to our perfectly adequate prediction model, but this is in fact ideal for having the imputed values mimic the uncertainity in the unknown values.

Cons

- Can lead to extreme values such as negative values [e.g. negative ozone shown in example above], which are impossible in real-world.
- This model assumes equal dispersion of data for the entire linear model. Hence, not accurate for heteroscedastic (non-constant variance) distributions ([1]). In the example above, observed dispersion is more extreme in the 200~300Ly Solar.R band, but imputed values do not acocunt for this and follow same degree of dispersion throughout.

LOCF (longitudinal data imputation)

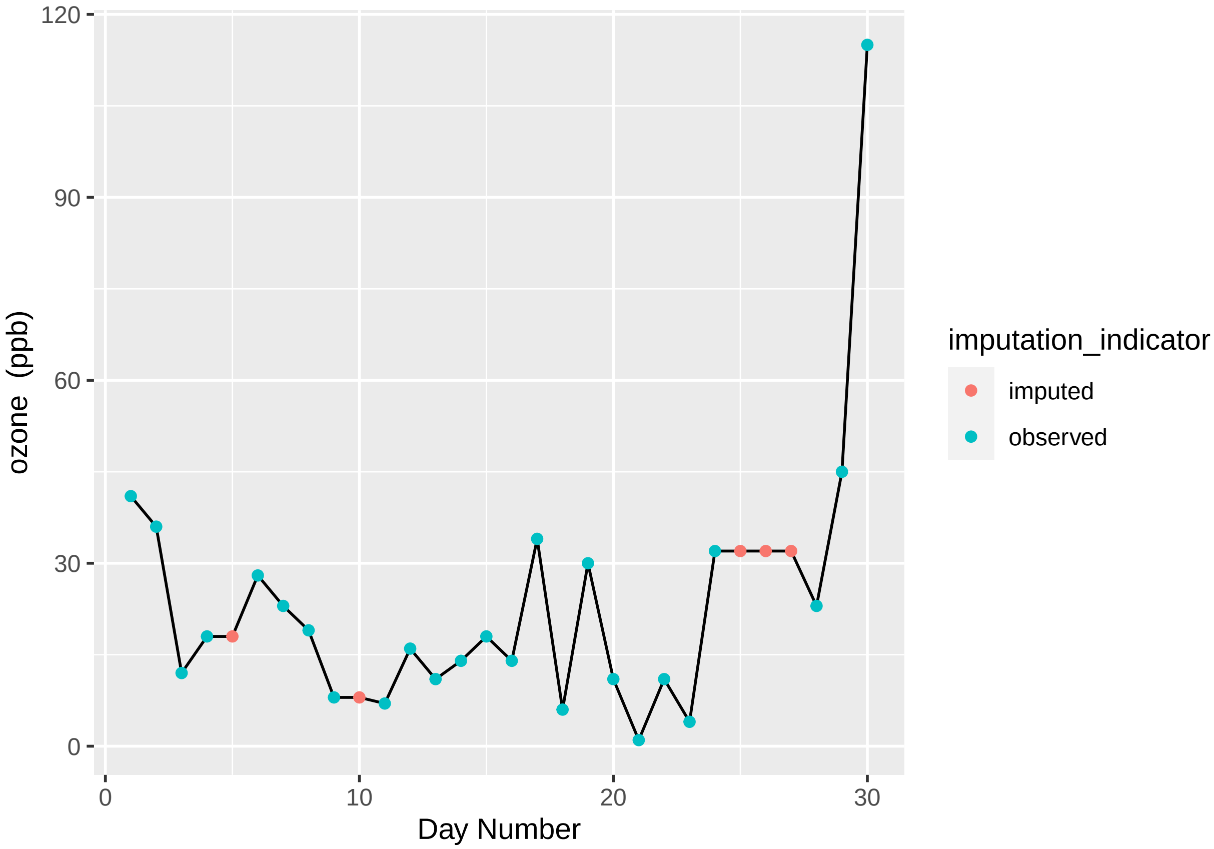
Description

- LOCF (Last Observation Carried Forward) is a way of imputing longitudinal data.
- It imputes missing values by replacing them with the latest observed value ([1]). Typically seen in clinical trials

Example

```
# add a missingness indicator for the Ozone variable using the airquality dataset.
airquality_indicator <- airquality %>%
  mutate(imputation_indicator = if_else(is.na(Ozone), "imputed", "observed"))

# Using the tidyr package, impute Ozone NA values with the previous observed value.
airquality2 <- tidyr::fill(airquality_indicator, Ozone)
# plot a line graph using the first 30 observations in the dataset.
# Notice imputed values have the same value as the previous observed value under LOCF.
head(airquality2, 30) %>% ggplot(aes(x = Day,y= Ozone, )) +
  geom_line() +
  geom_point(aes(color = imputation_indicator)) +
  labs(y = "ozone  (ppb)",
       x = "Day Number")
```



Pros

- Conceptually very simple to generate a complete dataset.
- Computationally cheap relative to other techniques that may be more statistically sound (e.g. multiple imputation).

Cons

- Should only be applied when we are confident in cases where we are certain what the missing values should be ([7], [8]).
- Can yield biased estimators even under MCAR ([7]).

BOCF (longitudinal data imputation)

Description

- BOCF (Base Observation Carried Forward) is a way of imputing longitudinal data.
- As per its name suggests, BOCF imputes missing values by replacing them with some predetermined baseline value; typically used in clinical trials where a "baseline value" can easily be established ([9]).

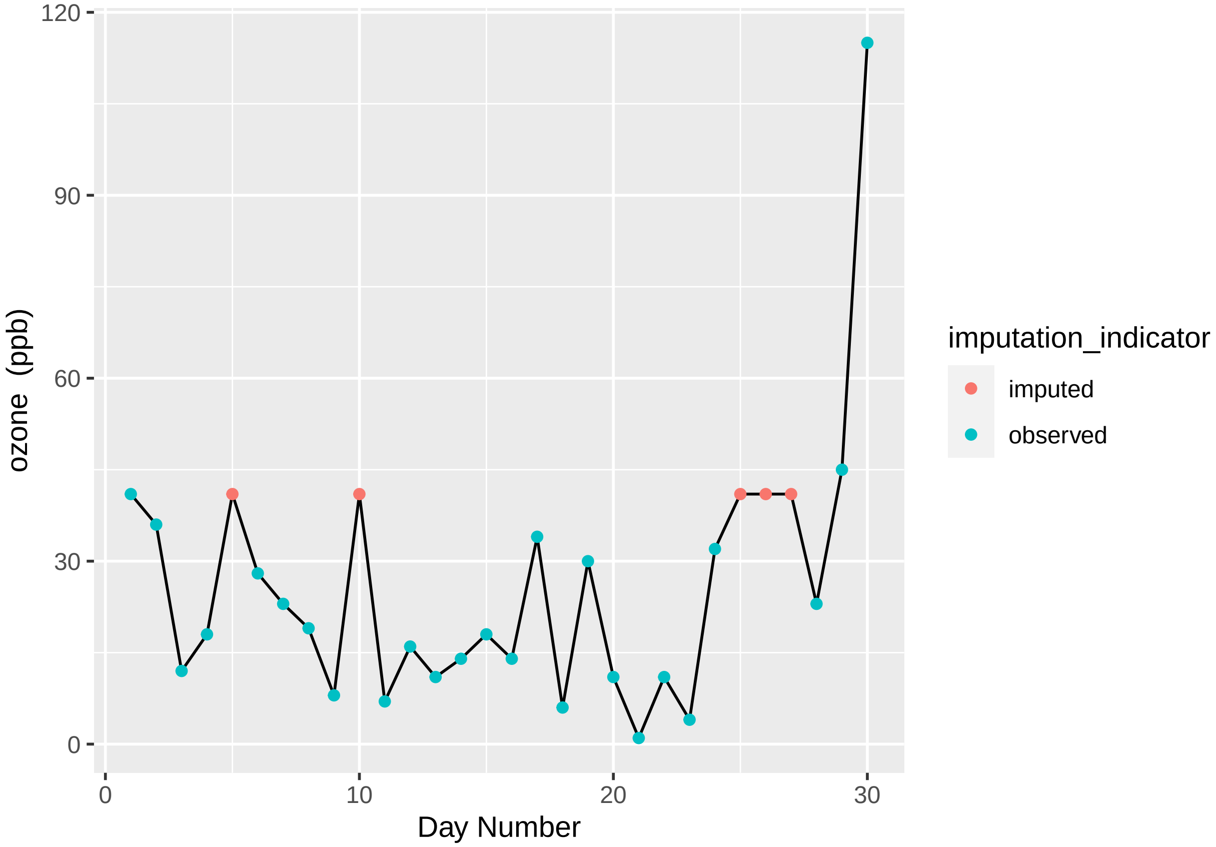
Example

```r
# Store baseline (first observation) of relevant variable [Ozone in this case]
baseline_Ozone = (airquality$Ozone)[1]

# Impute missing values with baseline observation as well as adding ozone
# missingness indicator variable.
airquality_LOCF <- airquality %>%
  mutate(imputation_indicator = if_else(is.na(Ozone), "imputed", "observed")) %>%
  replace_na(list(Ozone = baseline_Ozone))
```

- Notice in BOCF imputed values are equal to the baseline value observed on Day 1.

```
# Create line graph of ozone vs day for the first 30 observations,
# where imputed values are highlighted using red.
head(airquality_LOCF, 30) %>% ggplot(aes(x = Day,y= Ozone)) +
  geom_line() +
  geom_point( aes(color = imputation_indicator)) +
  labs(y = "ozone  (ppb)",
       x = "Day Number")
```



Pros

- Conceptually very simple in generating a complete dataset.
- Computationally cheap relative to other techniques that may be more statistically sound (e.g. multiple imputation).

Cons

- Should only be applied when we are confident in cases where we are certain what the missing values should be ([9]).
- Can yield biased estimators even under MCAR ([8]).
- The US National Reserach Council Panel on Handling Missing Data in Clinical Trials recommended in 2010 that BOCF shouldn't be used unless under very specific circumstances where BOCF assumptions

are justified.

Missing Indicator Method

Description

- Imputation technique applicable when explanatory variable(s) have missingness but response does not ([1]).
- Missing values are replaced by a fixed value (usually 0 or mean imputation), and a missingness indicator variable is added to each observation in the dataset for each incomplete variable ([1]).

Example

- Initial unaltered dataset. Note the 6 variables.

```
head(airquality)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

```
# Impute the airquality dataset, using observed mean as the fixed imputation value.
imp <- mice(airquality, method = "mean", m = 1,
            maxit = 1, print = FALSE)
# add column to airquality dataset indicating whether Ozone value is missing or not.
airquality2 <- cbind(complete(imp),
                     r.Ozone = is.na(airquality[, "Ozone"]))
# apply multiple linear regression
fit <- lm(Wind ~ Ozone + r.Ozone, data = airquality2)

# Note how airquality2 now has 7 variables with the addition of the Ozone indicator.

head(airquality2)
```

```
##      Ozone  Solar.R Wind Temp Month Day r.Ozone
## 1 41.00000 190.0000  7.4   67     5   1   FALSE
## 2 36.00000 118.0000  8.0   72     5   2   FALSE
## 3 12.00000 149.0000 12.6   74     5   3   FALSE
## 4 18.00000 313.0000 11.5   62     5   4   FALSE
## 5 42.12931 185.9315 14.3   56     5   5    TRUE
## 6 28.00000 185.9315 14.9   66     5   6   FALSE
```

```
fit$coefficients
```

```
## (Intercept)       Ozone r.OzoneTRUE
## 12.60842987 -0.06518884  0.39468779
```

Pros

- In randomized trials, can be used to generate unbiased estimators in MCAR and MAR covariate situations ([10]).
- Retains the whole observed dataset.

Cons

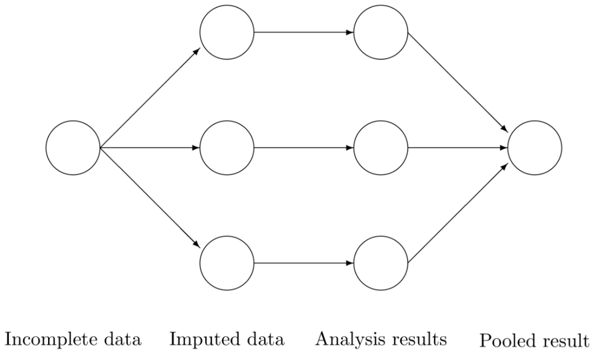
- In nonrandomized studies (e.g. observational studies), this method can result in biased estimators and associations between variables even in MCAR situations ([10]).
- Does not allow for missingness in response variable.

Multiple Imputation

Description:

Imputing one value for a missing datum cannot be correct in general, because we don't know what value to impute with certainty (if we did, it wouldn't be missing)— Donald B. Rubin

Mulitple Imputation Pipeline:



1. Imputation method for multivariate data that takes an incomplete dataset as input and creates multiple copies of the complete data ([1]).
2. Then, impute incomplete columns with plausible values given other columns through an iterative predictive method (methods include predictive mean matching, random forests, mean imputation, etc.) ([11]).
    - Iterative methods are used until imputed values converge (typically 5-10 iterations is sufficient) ([12]).
3. Next, obtain an estimate for the parameter of interest for each version of the imputed dataset ([1]).
    - This is done using regular analysis techniques similar to the other imputation procedures.
4. Finally, pool estimators together to create a single pooled estimate ([1]).

- Naively can visualize multiple imputation as applying stochastic regression imputation multiple times and summarizing results.
- Number of imputations should be equal to %-age of missingness as a rule of thumb ([1]).
- Note that multiple imputation has serious depth to it (changing predictor matrix, passive imputation, etc.) that I will not be discussing in detail here.

Example

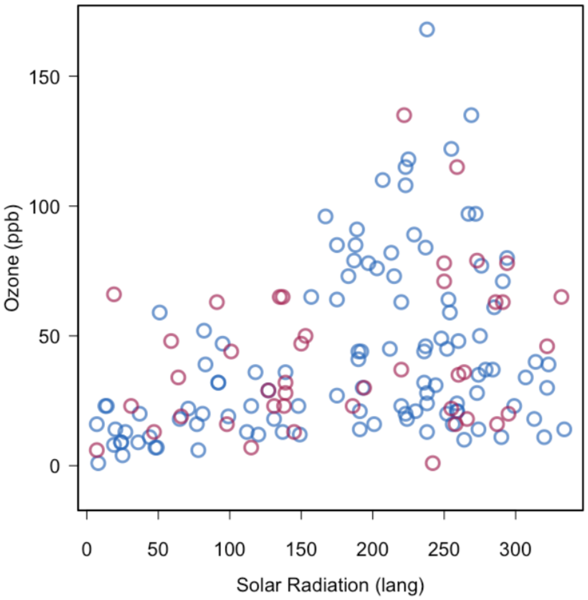
- The *mice()* function applies multiple imputation. The *m* parameter refers to the # of datasets we want to create, i.e. the # of circles in the 2nd and 3rd steps of the schematic above ([13]).

```
# [step 1 above]
incomplete_data <- airquality
# [Step 2 above] Apply multiple imputation to airquality to create m = 20 datasets
```

```
imp <- mice(incomplete_data, seed = 1, m = 20, print = FALSE)
# [step 3 above] Fits multiple linear regression model to predict Ozone using
# Wind, Temp, Solar.R as predictor vars
fit <- with(imp, lm(Ozone ~ Wind + Temp + Solar.R))
# [step 4 above] pools the twenty datasets' estimated parameters
# [B_0, B_1, B_2, B_3] to obtain a single pooled set of parameters
summary(pool(fit))
```

```
##          term     estimate   std.error statistic       df      p.value
## 1 (Intercept) -65.87829658 23.09377412 -2.852643 69.97033 5.696702e-03
## 2        Wind  -3.01897171  0.66252377 -4.556775 70.51194 2.125022e-05
## 3        Temp   1.63483547  0.25107557  6.511328 75.99913 7.203792e-09
## 4     Solar.R   0.05861581  0.02267832  2.584662 90.10797 1.135441e-02
```

```
# scatter plot for complete dataset of first imputation
knitr::include_graphics("images/mi_first_imp.png")
```



Pros:

- Gives unbiased and confidence-valid estimators under MAR, MCAR ([1]).
- Highly versatile and general technique. Can accommodate situations with low confidence in missing values by having a large number of imputed dataset copies m; similarly, can accomodate situations with high confidence in missing values by having a small number of imputed dataset copies m ([12]).
- Produces a suitable standard error value (listwise deletion produces too large standard error, other techniques produce too small standard error) close to true value ([1]).

Cons:

- Ad-hoc methods specified above may work better & faster in edge cases ([10]). For example, listwise deletion is equivalent and faster than multiple imputation if missing values occur only in the outcome.
  - This is because if missingness is only in the outcome, then missing data model is the same as the prediction model as no predictors are imputed ([1]). Hence, multiple imputation will simply perform complete-case analysis in a roundabout way.

- Multiple imputation has multiple parameters that can be varied (# of imputations m, # of iterations *maxit*, predictor matrix, imputation method, etc.) and so can be hard to work with & optimize for given situation.
- Doesn't create unbiased estimators under MNAR.

Works Cited

1. Van Buuren S. Flexible imputation of missing data. CRC press; 2018.
2. Ritvik. Missing data mechanisms. 2018.
3. Mia. Missing data assumptions (MCAR, MAR, MNAR). 2020.
4. Rönkkö M. Simple techniques for dealing with missing data. 2020.
5. Carey G. Important matrices for multivariate analysis. 1998.
6. Schork J. Regression imputation (stochastic vs. Deterministic & r example). 2021.
7. Liu X. Methods and applications of longitudinal data analysis. Elsevier; 2015.
8. Council NR et al. The prevention and treatment of missing data in clinical trials. 2010.
9. Deng Dr. Single imputation methods for missing data: LOCF, BOCF, LRCF (last rank carried forward), and NOCB (next observation carried backward). 2021.
10. Groenwold RH, White IR, Donders ART, Carpenter JR, Altman DG, Moons KG. Missing covariate data in clinical research: When and when not to use the missing-indicator method for analysis. Cmaj. 2012;184:1265–9.
11. Wilson S. The MICE algorithm. 2021.
12. Toshniwal R. Multivariate imputation by chained equations (MICE) algorithm for missing values | machine learning. 2021.
13. van Buuren S, Groothuis-Oudshoorn K. mice: Multivariate imputation by chained equations in r. Journal of Statistical Software. 2011;45:1–67.