|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Age | Height | Weight | BMI | Symptom severity | Medicine type | Recovery time |
| Emma | 55.5 | 159 | 60 | 23.7 | NaN | B | 13 |
| Isabella | NaN | 164 | 65 | 24.2 | 8 | B | NaN |
| Liam | 37.2 | 186 | NaN | NaN | NaN | A | 5 |
| Evelyn | NaN | 156 | 56 | 23.0 | 4 | B | 15 |
| Joey | 37.0 | 188 | 87 | 24.6 | NaN | A | 5 |
| Charlotte | 46.5 | 166 | NaN | NaN | 2 | A | 8 |
| Logan | 42.1 | 186 | 83 | 24.0 | 5 | B | NaN |

**How to manage missing data, for data scientists**

By Eefje Poppelaars

Whether it be because of technical difficulties, branching data collection, or excluded outliers, most datasets have missing data. Since statistical calculations do not work with missing data, they need to be managed somehow. But what is the best way to manage missing data? I will discuss common options, going from most simple to most complicated (but not necessarily better):

1. **Prevention**

As always, the best intervention is prevention. When you are involved during the data collection, try to assure that there is the least amount of missing data as possible.

1. **Complete case analysis**

The easiest way to handle missing data is to remove all observations that contain missing data, and only use complete observations for the statistical calculations (a.k.a. listwise deletion). This simple approach has the advantage that significance levels and standard errors (though larger) will be correct.

However, because this approach is so wasteful, it is only possible when the dataset is big enough to maintain enough statistical power. Additionally, this is only a valid strategy when data is missing completely at random (MCAR, i.e., the probability to be missing is constant for all observations), as opposed to missing at random (MAR, i.e., the probability to be missing depends on observed data) or missing not at random (MNAR, i.e., the probability to be missing depends on unobserved data). In other words, when data is missing according to some pattern, simply removing the missing data will bias your results.

Take for example the case of calculating average BMI based on self-reported height and weight. In the case that people with more extreme BMI’s don’t want to disclose their weight. The missingness of the data here depends on BMI (i.e., MNAR). Thus, complete case analysis will give the illusion that the average BMI is healthier than reality.

There is however one exception to the rule of biased results under complete case analysis with MAR data: when calculating a regression model where X has missing values but Y does not, the prediction of Y based on X will not be biased, as long as the missingness of X does not depend on Y. In contrast, for logistic regression: when either X or Y has missing values, the prediction of Y on X will not be biased, as long as the missingness depends only on Y and not on X.



1. **Pair-wise deletion**

Similar to listwise deletion (or complete case analysis), pairwise deletion can be used. In this approach, only the relevant observations with missing data are deleted that are needed for each individual analysis. This provides similar advantages to complete case analysis, such as the significance levels and standard errors (though larger) being correct, but complicates the results by having a different sample size for each analysis. Additionally, it will again only work for MCAR data.

1. **Mean imputation**

Another easy way to handle missing data is to simply replace any missing values with the mean (or median, or mode).

The mean-substitution approach has the advantage that the mean itself will not be influenced. However, it has many serious disadvantages, such as disturbing the distribution, underestimating the variance and standard errors, decreasing correlations with other variables, and again biasing the results when the data are missing according to some pattern. Additionally, it will only work for MCAR data. Thus, this approach is best avoided.

Take for example the case of calculating the effect of medicine A vs. B on recovery time, while patients only get prescribed one medicine, and medicine B is more often prescribed to patients with more severe disease symptoms. This data is missing depending on symptom severity. In this case, mean imputation will give the illusion of medicine B being less effective than medicine A. Thus, the mean imputation approach can give biased results for MAR and MNAR data.



1. **Regression imputation**

Another popular approach to handle missing data is to use prediction, or regression imputation. This works by fitting a regression model with complete cases to predict Y from other variable(s) in the dataset that happen(s) to reasonably accurately predict Y (e.g., X). The missing values are then predicted based on the regression model (Y = intercept + regression weight of X \* X), and replaced by this prediction. This approach has the advantage that the regression coefficients will be unbiased for MAR data, and can provide a good approximation of the missing data when the explained variance by the regression model is high. However, it will not work for MCAR data.

Going back to the example of effectiveness of medicine A vs. B, depending on symptom severity. If symptom severity was measured, the data are missing at random, and estimates of medicine effectiveness can be corrected for by using an additional covariate of symptom severity. However, if symptom severity was not measured, the data are missing not at random, and the results will be biased to the point that the research question cannot be answered.

However, the regression imputation approach also has serious disadvantages, such as underestimating the variance and standard errors, increasing correlations with predictor variables, and too optimistic p-values and confidence intervals. Thus, although better than mean imputation, regression imputation is also not recommended.



1. **Stochastic regression imputation**

Since regression imputation underestimates the variance, you can use stochastic regression imputation instead to alleviate this issue. It works in the same way as regression imputation, but adds appropriate noise to the predictions to reflect uncertainty (Y = intercept + regression weight of X \* X + error). This approach has the advantage of preserving the distribution and the correlation with other variables, besides also providing unbiased regression coefficients for MAR data and providing a good approximation of the missing data when the explained variance by the regression model is high. However, the induced noise is limited to symmetric and constant error, and subsequent analyses do not take the uncertainty of the predictions into account, treating the predicted missing data as real values instead, with variance and standard errors again underestimated.



1. **Indicator method**

In an attempt to improve upon the stochastic regression imputation approach by alleviating the disadvantage that predicted missing data is treated as real data, the indicator method (a.k.a. dummy variable adjustment) can be used. Using this approach, a missingness indicator is added into the regression model. Practically, two new variables need to be calculated: a missingness indicator M (if missing: M = 1, if not missing: M = 0) and a mean-imputed variable Z (if missing: Z = mean(Y), if not missing: Z = Y). Then use the following regression model to predict the missing data: Y = intercept + regression weight of M \* M + regression weight of Z \* Z + error.

This approach has the advantage of providing an even better approximation of the missing data when the explained variance by the regression model is high, besides again preserving the distribution and the correlation with other variables. However, since it is essentially based on mean-imputation, the disadvantage is that it provides biased regression coefficients, even for MCAR data, as well as underestimating variance and standard errors and providing incorrect p-values and confidence intervals. Thus, this approach is essentially worse than the stochastic regression imputation and is not recommended.

1. **Multiple imputation**

Multiple imputation is a state-of-the-art method to manage missing data, and has been demonstrated to work well for both simulations as well as for real data, and works for both MCAR and MAR data, and for all types of variables (continuous, binary, count, etc.). Essentially, this approach takes missing data and replaces them with appropriate values multiple times, after which you can analyze the results for each imputed dataset and then pool those results. This approach accounts both for sampling uncertainty (within-data variance, i.e., conventional variability) as well as for the missing data uncertainty (between-dataset variance). Thereby, the p-values and standard errors are correct and the distribution remains intact.

I will now walk you through the practical steps involved in doing multiple imputations:

* You first need to create a predictor matrix, where you specify which predictors should be used for which variable. Take care not to have too many predictors at once (as a rule: <10) and do not use predictors with very high correlations to avoid multicollinearity (the algorithm will kick them out automatically anyway). Due to its fully conditional specification algorithm, it can handle missing data in both X and Y. Doing passive imputation for some variables is also possible, where you don’t predict certain variables from other variables but rather from a predetermined formula (e.g., predicting BMI from height and weight that are also present in the data according to the formula kg/m2, again to prevent multicollinearity).
* Then you need to decide how many imputed datasets you need. The advice is to develop the model using five imputed datasets, and also check whether running the analyses with different seeds causes large differences (this implies that the data contain little information for those parameters).
* You also need to decide the number of iterations the model is allowed to reach convergence. 5-10 Iterations is usually adequate for most problems, but you might need more iterations for more difficult problems.
* Importantly, you need to decide which method of prediction to use. The standard is predictive mean matching, which works kind of like stochastic regression imputation but with more variability in the error. Predictive mean matching also works well for non-normally distributed data and most other cases. There are however many other options available, such as: logistic regression, linear regression using bootstrap, Bayesian linear regression, a random sample from observed values, random forest imputations, etc.
* The quality of the imputations needs to be inspected using both the header as well as visualizations (e.g., strip plot, boxplot, density plot). Also inspect the trace line plots for convergence (this looks like overlapping lines).
* When you are satisfied with the model, do a final run with as many imputed datasets as the percentage of incomplete cases (e.g., 43% of my observations contain missing values, so I need 43 imputed datasets).
* Use the imputed datasets to run your statistical analyses for each imputed dataset, and then pool the results according to Rubin’s rules. Although multiple imputation is certainly the superior method to manage missing data, analyzing the complicated format of imputed datasets comes with its own challenges, as standard statistical methods are not equipped to deal with such data. Luckily, the most common statistical methods are already implemented in R and are as easy to run as their standard counterparts. For more uncommon methods and for uses in Python, however, you will need to hard-code the analyses yourself, and you will need a firm grasp on Rubin’s rules in order to pool the estimates and variances. Warning: it is incorrect to first average the imputed data and then run the statistical analyses as usual because the estimates and variance will not take the between-imputed data variance into account!

Multiple imputation is implemented in the R package ‘mice’ (https://cran.r-project.org/web/packages/mice/index.html), with several add-on R packages such as ‘miceadds’, ‘mitools’, ‘mitml’, ‘micemd’, and ‘MKmisc’. In Python, it is implemented in ‘skikit-learn’ in a limited fashion, inspired by the ‘mice’ R package (https://scikit-learn.org/stable/modules/impute.html). The ‘mice’ package is also accompanied by the free e-book ‘Flexible imputation of missing data’ (https://stefvanbuuren.name/fimd/).

I have personally used multiple imputation in my research into associations between stress responses, read more about this research here: [link].



**Conclusion**

So which approach is best to handle missing data? It depends on several things: whether the data is MCAR, MAR, or MNAR; the size of the dataset; the proportion of missing data; the purpose of your analyses; etc. In general: the prevention of missing data is the best method, followed by multiple imputation. For easier solutions, the best approaches are probably (in descending order): complete case analysis (only for MCAR), pairwise deletion (only for MCAR), stochastic regression imputation, and regression imputation. Other discussed methods are not recommended, but now you at least know why to avoid them.

One final tip: when you are involved during the data collection and you suspect that the data will be missing at random depending on some other confounding variable, try to make sure that the suspected confounding variable will also be measured. This way, you can check and correct for its confounding influence later.

All R scripts that were used to produce the plots, as well as a dashboard, can be found here: https://github.com/ESPoppelaars/MissingDataManagement.