MatchThem::CHEAT SHEET

Matching Multiply Imputed Datasets

Pishgar et al. | October 2019

0::

Missing values are a common problem in analyzing data of observational studies. There are 3 kinds of missing values: (1) missing-completely-at-random, meaning that a random subset of observations have missing values, (2) missing-at-random, meaning that observations with specific values in other variables have higher chances to have missing value in a variable, and (3) missing-not-at-random, meaning that observations with specific values in a variable have higher chances to have missing value in the same variable.

Multiply imputing the missing values (i.e. replacing the missing values with a set of plausible values) is a credible approach to account for missing values when missingness is completely-at-random or at-random.

2::MATCHING::

In this step, you match observations in the imputed datasets using the `matchthem()` function. `matchthem()` offers a variety of options for this step:

```
>   models <- matchthem(formula, datasets,
                        approach = "within",
                        method = "cem", ...)
```

When `formula` specifies the model (e.g. `z ~ x1 + x2 + x3`) that should be used for the matching, `datasets` specifies the multiply imputed datasets, `approach` specifies that matching should be performed within each imputed dataset (`within`) or an averaged propensity score across the datasets should be used for matching (`across`), and `method` specifics the method that should be used for matching. `matchthem()` supports several matching methods including `nearest` (nearest neighbor matching), `exact` (exact matching), and `cem` (coarsened exact matching).

The output will be saved in a `mimids` class object. The extent of the balance of covariates in the datasets after matching can be reviewed with `summary()` or `plot ()` functions or the `bal.tab()` function from the `cobalt` package:
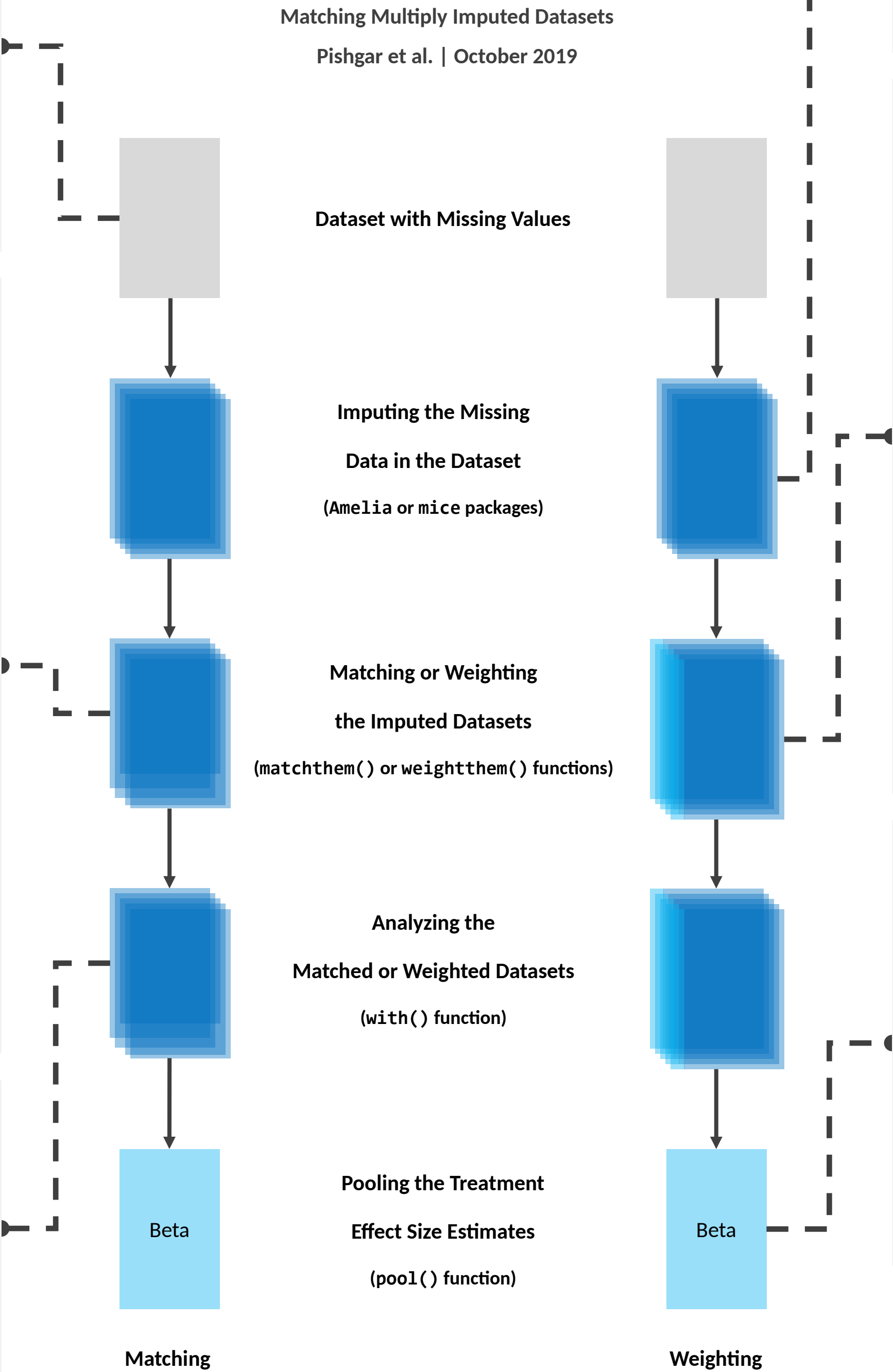
```
>   library(cobalt)
>   bal.tab(models)
```

3::

In this step, you should estimate the treatment effect size in each matched or weighted dataset using the `with()` function:

```
>   results <- with(models, glm(y ~ z,
                        family = binomial))
```

When `models` specifies the `mimids` or `wimids` object. `with()` function supports expression like `glm()`, `lm()`, `coxph()`, `svyglm()`, `svycoxph()`, etc. The output will be saved in a `mira` class object.

Dataset with Missing Values

↓

Imputing the Missing Data in the Dataset

(Amelia or `mice` packages)

↓

Matching or Weighting the Imputed Datasets

(`matchthem()` or `weightthem()` functions)

↓

Analyzing the Matched or Weighted Datasets

(`with()` function)

↓

Pooling the Treatment Effect Size Estimates

(`pool()` function)

Beta

Matching   Weighting

1::

In the first step, you multiply impute the missing values in your dataset. There are several R packages to do this, including the **Amelia**, **Hmisc**, **mi**, **mice**, and **missForest** packages. **MatchThem**, in this version, is compatible with imputed datasets by **Amelia** and **mice** packages (please see vignettes of these packages for details).

2::WEIGHTING::

In this step, you estimate weights of observations in the imputed datasets using the `weightthem()` function. `weightthem()` offers a variety of options for this step:

```
>   models <- weightthem(formula, datasets,
                        approach = "within",
                        method = "ps", ...)
```

When `formula` specifies the model (e.g. `z ~ x1 + x2 + x3`) that should be used for weighting, `datasets` specifies the multiply imputed datasets, `approach` specifies that weights should be calculated within each imputed dataset (`within`) or calculated using an averaged propensity score across the datasets (`across`), and `method` specifics the method that should be used for calculating the weights (`ps`, `gbm`, `cbps`, etc.).

The output will be saved in a `wimids` class object. The extent of the balance of covariates in the datasets within the `wimids` object can be reviewed with `summary()` function or the `bal.tab()` function from the `cobalt` package:

```
>   library(cobalt)
>   bal.tab(models)
```

4::

In this step, you should pool the treatment effect sizes (obtained from analyzing each matched or weighted dataset in the `mira` object) using the `pool()` function:

```
>   output <- pool(results, dfcom = NULL)
```

When `results` specifies the output of the `with()` function and `dfcom` specifies the degree of freedom in the analysis (when not specified, it is usually extracted from the dataset with the lowest number of observations).

The output will be saved in a `mipo` class object and can be reviewed with the `summary()` function:

```
>   summary(output, conf.int = TRUE )
```

MatchThem (version 0.8.2) imports functions from several other packages, including **MatchIt**, **mice**, and **WeightIt** | Please see the package **GitHub page** (github.com/FarhadPishgar/MatchThem) for more details