README: miWQS

Introduction

We have integrated WQS regression into the MI framework in a flexible $\mathcal{R}$ package called **miWQS** to meet a wide variety of needs (Figure 1). The data used in this package consist of a mixture of correlated components that share a common outcome while adjusting for other covariates. The correlated components in the set, X, may be complete or interval-censored between zero and low thresholds, or detection limits, that may be different across the components. The common outcome, y, may be modeled as binary, continuous, count-based, or rate-based and can be adjusted by the `family` and `offset` arguments of `estimate.wqs()`. Additional covariates, $\mathbf{Z}$, may be used in the bootstrap imputation and WQS models. If X is interval-censored, the choice of the imputation technique depends on the majority vote of BDL values among the components. When most chemicals have 80% of its values BDL, we suggest to use the BDLQ1 approach. When most chemicals have less than 80% of its values BDL, the user should perform Bayesian or bootstrapping multiple imputation. Previous literature suggests to ignore any chemicals that have greater than 80% of its values BDL. The **miWQS** package, though, still allows the user to perform single imputation. Regardless of the technique used, researchers may use the **miWQS** package in order to detect an association between the mixture and the outcome and to identify the important components in that mixture.

Installation

You can install the released version of miWQS from CRAN with:

```r
install.packages("miWQS")
```

Do you have missing components?

yes

no

Do you impute and know the detection limits?

WQS Regression: *estimate.wqs()*

yes

no

How many chemicals have under 80% of their values BDL?

many

Use BDLQ1 approach: *estimate.wqs()*

few

Do you want to create one or many imputed datasets?

one

many (recommended)

Single Imputation (not recommended)
1. Select your imputation method of choice, bootstrap or Bayesian *impute.boot()* or *impute.univariate.bayesian.mi()*
2. Do WQS: *estimate.wqs()*

Multiple Imputation (recommended)
1. Select your imputation method of choice, bootstrap or Bayesian: *impute.boot()* or *impute.univariate.bayesian.mi()*
2. Do WQS: *do.many.wqs()*
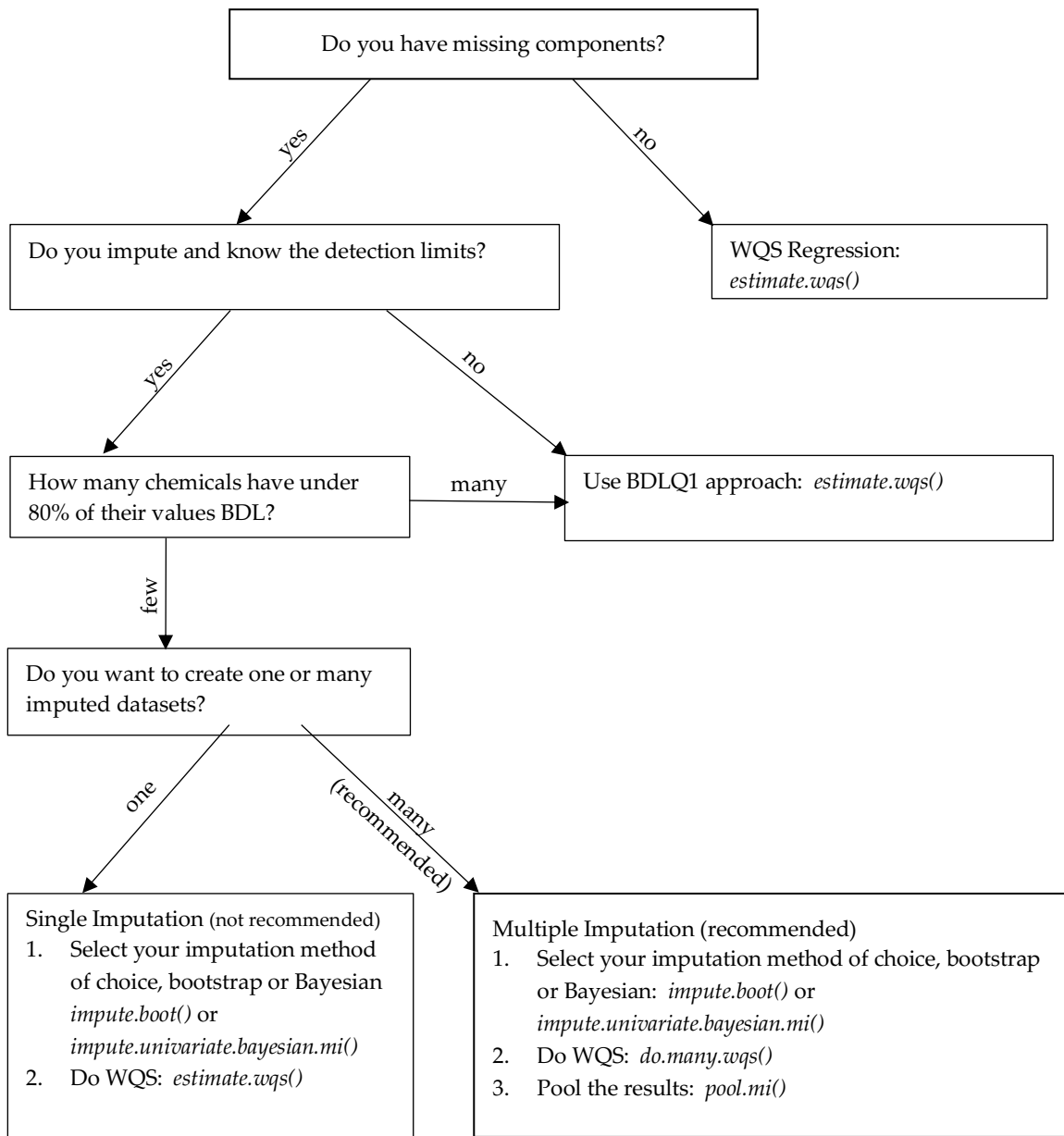3. Pool the results: *pool.mi()*

Figure 1: A decision tree to help researchers in using the miWQS package. The package is flexible and can meet a wide range of needs.