УДК 378.1:004

Т. И. Воротницкая

Институт бизнеса БГУ, Минск, Беларусь, varatnitskaya@gmail.com

МЕТОДЫ ОБРАБОТКИ ПРОПУЩЕННЫХ ЗНАЧЕНИЙ С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА ПРОГРАММИРОВАНИЯ R

В статье рассматриваются основные подходы для обработки пропущенных значений в наборах данных с использованием языка программирования R. Приведены примеры использования встроенных пакетов, библиотек и функций для работы с пропусками в анализе данных.

Ключевые слова: *пропущенные значения, язык программирования R, анализ данных*

T. Varatnitskaya

School of Business of BSU, Minsk, Belarus, varatnitskya@gmail.com

METHODS FOR HANDLING MISSING VALUES IN DATA SETS USING THE R PROGRAMMING LANGUAGE

The main approaches for handling missing values in data sets using the R programming language are considered in the article. Examples of using built-in packages, libraries and functions for working with missing data are given.

Keywords: *missing values; R programming language; data analysis*

Для решения задач интеллектуального анализа данных используются многие языки программирования, но предпочтения отдаются тем, которые содержат много готовых инструментов, функций и библиотек для эффективной обработки данных. Самые популярные – это Python, R, Java, SQL, MatLab, Scala, C++ и прочие. В современных реалиях особую актуальность приобретает использование программного обеспечения с открытым кодом. Примером такого программного обеспечения может служить язык и среда программирования R [1]. Язык R распространен в научной среде, им пользуются математики, биологи, генетики, те, кто занимается статистическими исследованиями, анализом данным, построением математических моделей. R, с одной стороны, – язык программирования, а с другой – рабочая среда, в которую встроены средства статистического анализа, оптимизации, визуализации и др. Для R есть удобная среда разработки RStudio, есть приложение-блокнот Jupyter Notebook для создания и обмена программами на R прямо в браузере и дистрибутив Anaconda с коллекцией популярных библиотек. В настоящей статье излагаются основные подходы и алгоритмы обработки пропущенных значений, базирующиеся на использовании языка и среды программирования R.

В интеллектуальном анализе данных при решении большинства задач одним из первых этапов является предобработка исследуемого набора данных, который включает в себя обнаружение пропущенных значений и при необходимости замена их искусственными данными. При выборе методов борьбы с пропусками рекомендуется предварительно исследовать природу пропущенных значений. Механизмы формирования пропущенных значений впервые были систематизированы в работе [2]. Авторами были выделены три типа формирований пропущенных значений: MCAR (Missing Completely At Random), MAR (Missing At Random) и

MNAR (Missing Not At Random). Первый механизм формирования пропусков MCAR предполагает, что пропущенные значения полностью случайны, т. е. уточнить предсказание о пропущенных значениях при помощи имеющейся информации невозможно. Во втором механизме MAR считается, что данные обычно пропущены не случайно, а ввиду некоторых закономерностей, т. е. вероятность пропуска может быть определена на основе другой имеющейся в наборе данных информации, не содержащей пропуски. Как в первом, так и во втором случаях удаление или замена пропусков на искусственные значения не приведет к существенному искажению результатов. Третий тип формирования пропущенных значений MNAR предполагает, что данные отсутствуют в зависимости от неизвестных факторов и вероятность пропуска могла бы быть описана на основе других признаков, но информация по этим признакам в наборе данных отсутствует, т.е. пропущенные значения невозможно выразить на основе информации, содержащейся в наборе данных.

Одним из самых распространенных методов при работе с неполными данными – это удаление объектов, для которых значение хотя бы одного признака пропущено. Это быстрый метод борьбы с пропусками, применяемый во множестве прикладных пакетов как метод по умолчанию. Иногда предварительно рекомендуется избавиться от сильно разреженных признаков, т. е. удалить столбцы таблицы, содержащие большое количество пропусков. В случае механизма пропусков MCAR применение данного метода не приведет к существенному искажению параметров модели. Однако удаление строк приводит к тому, что при дальнейших вычислениях используется не вся доступная информация, стандартные отклонения возрастают, полученные результаты становятся менее репрезентативными.

В R [3] для идентификации недостающих данных можно использовать функцию ***is.na( )***, которая позволяет проверить данные на наличие пропущенных значений и возвращает объект такой же размерности, где элементы заменены на TRUE, если значение было пропущено и FALSE – в противном случае, или функцию ***complete.cases( ),*** возвращающую логический вектор, указывающий какие данные являются полными.

Функция ***na.omit( )*** удаляет все строки, в которых есть хотя бы одно пропущенное значение, формируя новый набор данных. При этом надо учесть, что нумерация строк сохраняется прежней.

Еще одним быстрым методом борьбы с пропусками является метод обработки, основанный на игнорировании пропусков в расчетах. Например, параметр ***na.rm = TRUE***, используемый в некоторых функциях, удаляет пропущенные значения перед вычислениями и позволяет применить функцию к оставшимся элементам (рис. 1):

```
> x<-c(2, 4, NA, 3)
> y<-sum(x)
> y
[1] NA
```

а

```
> x<-c(2,4, NA, 3 )
> y<-sum(x, na.rm=TRUE)
> y
[1] 9
```

б

Рис. 1. Пример вычисления суммы элементов вектора:
а – без использования параметра na.rm; *б* – с использованием параметра na.rm

Такие функции, как ***sum()***, ***prod()***, ***quantile()***, ***sd()*** и прочие, включают в себя данный параметр.

Преимущество данного подхода в том, что используется вся доступная информация. Статистические характеристики, такие как средние значения, стандартные отклонения, можно

рассчитать, используя все непропущенные значения для каждого из признаков. Недостаток заключается в том, что данные вычисления могут привести к некорректным результатам. Одномерные статистики теряют преимущество сравнимости, поскольку вычисляются на различных подвыборках в зависимости от распределения пропусков. Например, рассчитанные значения коэффициентов корреляции могут оказаться вне диапазона [-1; 1].

Следующий подход – это метод глобального заполнения, т. е. заполнения с использованием имеющихся данных. Часто используется при механизме формирования пропусков MAR. Например, заполнение средними показателями или медианными значениями. Недостатком такого метода является искажение показателей, характеризующие свойства распределения данных (кроме среднего значения), уменьшение дисперсии. При работе с временными рядами, когда последующие значения сильно взаимосвязаны с предыдущими, используется подход повторения результата последнего наблюдения LOCF (last observation carried forward). В языке программирования R замена пропущенных значений средним, медианой, модой осуществляется с помощью функции ***impute( )*** из библиотеки ***Hmisce***.

```
library(Hmisc)
x <- impute(x, fun = mean) # замена средним значением
x <- impute(x, fun = median) # замена медианой
x <- impute(x, fun = 10) # замена конкретным числовым значением
```

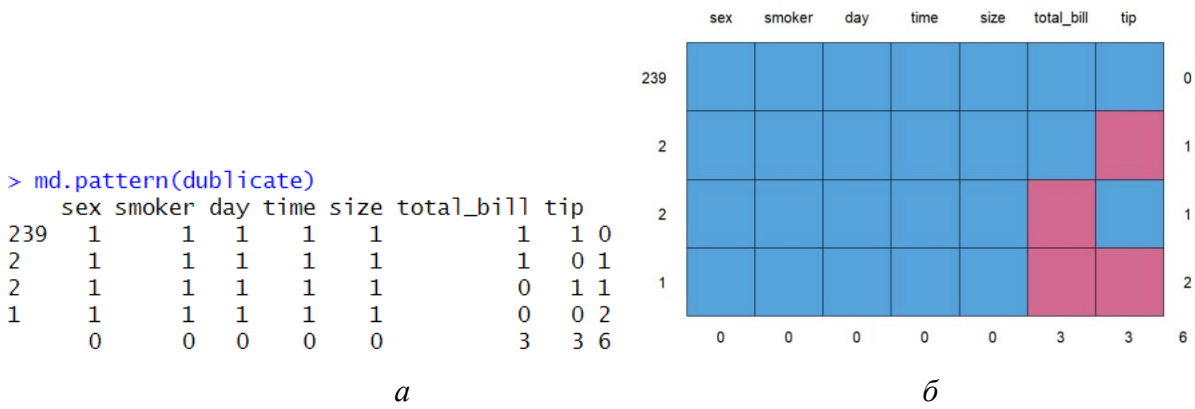
Рис. 2. Пример замены пропущенных значений

Восстановление пропусков на основе регрессионных моделей также является одним из известных методов. Он заключается в том, что пропущенные значения заполняются с помощью модели линейной регрессии, построенной на известных значениях набора данных. Достоинством данного метода является то, что получаются правдоподобно заполненные данные, а к недостаткам можно отнести потерю разброса значений реальных данных, уменьшение вариации значений восстанавливаемой характеристики, искусственное увеличение корреляции между характеристиками. В R построение модели линейной регрессии реализовано функцией ***lm()***. Далее с помощью функции ***predict()*** по построенной модели линейной регрессии можно спрогнозировать и заменить пропущенное значение.

Для более точного восстановления пропуска популярен метод k ближайших соседей (k-nearest neighbor). Оценка kNN строится следующим образом: для пропущенного значения, требующего замены, определяется k ближайших объектов со схожими признаками. В качестве меры сходства между объектами можно использовать, например, евклидово расстояние. По всем k объектам рассчитывается взвешенное среднее (по расстоянию), которое и заменяет пропуск. Нет четкого алгоритма выбора числа k схожих объектов, решение зависит от объема выборки и количества признаков. В R используется функция ***knnImputation(data, k = 10, scale = T, meth = "weighAvg", distData = NULL),*** где первым аргументом указывается набор данных, второй аргумент – это количество ближайших соседей (по умолчанию равно 10), далее указывается логический параметр, отвечающий за стандартизацию данных (по умолчанию TRUE), затем метод, используемый для расчета заменяющего значения доступные значения это медиана «median» или среднее значение «weighAvg» (по умолчанию). Последний параметр «distData» не обязателен, но можно указать здесь набор данных, который следует использовать для поиска соседей. Это полезно при заполнении пропущенных значений в тестовом наборе, где следует использовать только информацию из обучающего набора, по умолчанию значение параметра равно NULL, что означает поиск соседей в исходных данных.

Одним из известных пакетов в R для визуализации и заполнения пропусков является пакет MICE (Multivariate Imputation by Chained Equations) [4], который предоставляет сложные функции для работы с пропущенными значениями. Он использует способ оценки в два шага: ***mice()*** для построения модели и ***complete()*** для генерации данных. Функция mice() создает несколько полных копий набора данных, каждая со своей оценкой пропущенных данных. Функция ***complete()*** возвращает один или несколько наборов данных, набор по умолчанию будет первым. Алгоритм MICE может рассчитывать варианты для непрерывных, двоичных, неупорядоченных категориальных и упорядоченных категориальных данных.

Одна из функций пакета ***md.pattern()***, которая отображает структуру пропущенных значений в наборе данных в виде единиц и нулей в ячейках таблицы, где 0 обозначает пропущенное значение для данной переменной, а 1 – имеющееся значение. В качестве аргумента функции указывается имя набора данных (рис. 3)



а *б*

Рис. 3. Пример работы функции md.pattern():
а – табличное представление; *б* – матричная визуализация

На рисунке 3а видно, что первая строка содержит информацию о полных наблюдениях, вторая строка показывает число наблюдений равное 2, в которых нет пропущенных значений во всех переменных, кроме total_bill, в третьей указано, что в 2 наблюдениях пропущены значения в столбце tip, а единица в четвертой строке говорит об отсутствии данных в столбцах total_bill и tip одном наблюдении. Аналогичную информацию можно получить из графической матрицы на рисунке 3б.

Используя библиотеку ***library(VIM)*** можно визуализировать пропущенные данные. Например, с помощью функции ***aggr()*** графически отобразить число пропущенных наблюдений для каждой отдельной переменной и для каждой комбинации переменных в абсолютном или относительном виде, как показано на рис. 4. Функция ***matrixplot()*** графически отображает данные по каждой строке. Числовые данные нормализованы и находятся в интервале [0, 1]. Каждое значение представлено оттенком серого цвета, чем светлее оттенок, тем более низкому значению он соответствует. По умолчанию пропущенные значения обозначены красным. Такая матричная диаграмма позволяет увидеть, зависит ли наличие пропущенных значений в одной или нескольких переменных от значений других переменных. Недостатком такого вида визуализации является то, что она удобна для не очень больших по объему данных.

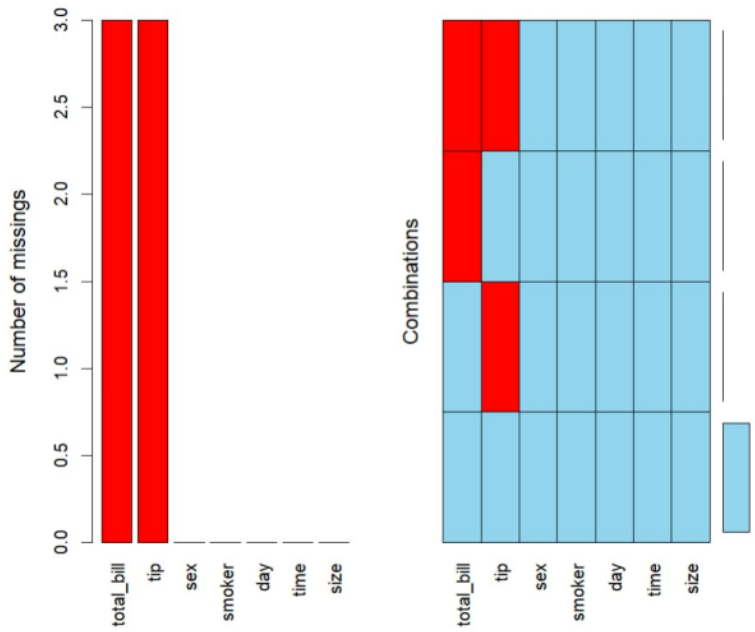


Рис. 4. Пример работы функции aggr()

В качестве других способов восстановления используют различные статистические подходы: EM-алгоритм, Full information maximum likelihood (FIML, полная оценка максимального правдоподобия) и другие. Данные подходы показывают неплохие результаты на больших выборках. Универсальной схемы для работы с пропущенными значениями не существует, но всегда можно определить их количество, характер пропусков и применить один из методов обработки пропущенных значений.

Список использованных источников

1. The Comprehensive R Archive Network [Electronic resource]. – Mode of access: http://https://cran.r-project.org. Date of access: 16.03.2024.

2. *Little, J. A. Roderick* Statistical Analysis with Missing Data / J. A. Roderick Little, Donald B. Rubin; 3rd edition. – Wiley, 2019. – 464 p.

3. *Kabacoff, Robert I.* R in Action. Manning Publications / Robert I. Kabacoff, 3rd edition. – 2022. – 656 p.

4. Multivariate Imputation by Chained Equations [Electronic resource]. – Mode of access: https://cran.r-project.org/web/packages/mice/mice.pdf. - Date of access: 16.03.2024.