applied sciences

MDPI

Review

R Packages for Data Quality Assessments and Data Monitoring: A Software Scoping Review with Recommendations for Future Developments

Joany Mariño [1,*,†], Elisa Kasbohm [1,*,†], Stephan Struckmann [1], Lorenz A. Kapsner [2,3] and Carsten O. Schmidt [1]

[1] Unit Quality in the Health Sciences (QIHS), Department SHIP-KEF, Institute for Community Medicine, University Medicine Greifswald, 17475 Greifswald, Germany; stephan.struckmann@uni-greifswald.de (S.S.); carsten.schmidt@uni-greifswald.de (C.O.S.)
[2] Medical Center for Information and Communication Technology, Universitätsklinikum Erlangen, Friedrich-Alexander-University Erlangen-Nürnberg (FAU), 91054 Erlangen, Germany; lorenz.kapsner@uk-erlangen.de
[3] Institute of Radiology, Universitätsklinikum Erlangen, Friedrich-Alexander-University Erlangen-Nürnberg (FAU), 91054 Erlangen, Germany
* Correspondence: joany.marino@uni-greifswald.de (J.M.); elisa.kasbohm@uni-greifswald.de (E.K.)
† These authors contributed equally to this work.

Abstract: Data quality assessments (DQA) are necessary to ensure valid research results. Despite the growing availability of tools of relevance for DQA in the R language, a systematic comparison of their functionalities is missing. Therefore, we review R packages related to data quality (DQ) and assess their scope against a DQ framework for observational health studies. Based on a systematic search, we screened more than 140 R packages related to DQA in the Comprehensive R Archive Network. From these, we selected packages which target at least three of the four DQ dimensions (integrity, completeness, consistency, accuracy) in a reference framework. We evaluated the resulting 27 packages for general features (e.g., usability, metadata handling, output types, descriptive statistics) and the possible assessment's breadth. To facilitate comparisons, we applied all packages to a publicly available dataset from a cohort study. We found that the packages' scope varies considerably regarding functionalities and usability. Only three packages follow a DQ concept, and some offer an extensive rule-based issue analysis. However, the reference framework does not include a few implemented functionalities, and it should be broadened accordingly. Improved use of metadata to empower DQA and user-friendliness enhancement, such as GUIs and reports that grade the severity of DQ issues, stand out as the main directions for future developments.

Keywords: data quality; data quality monitoring; data reporting; exploratory data analysis; initial data analysis; R project for statistical computing

Citation: Mariño, J.; Kasbohm, E.; Struckmann, S.; Kapsner, L.A.; Schmidt, C.O. R Packages for Data Quality Assessments and Data Monitoring: A Software Scoping Review with Recommendations for Future Developments. *Appl. Sci.* **2022**, *12*, 4238. https://doi.org/10.3390/app12094238

Academic Editor: Federico Divina

Received: 9 March 2022
Accepted: 18 April 2022
Published: 22 April 2022

1. Introduction

The assessment of data quality (DQ) has received increasing attention over the past years. Several DQ frameworks have been developed in the health sciences, most of them related to administrative data [1–5]. Among the exceptions, one targets observational studies specifically [6], and another covers different data and study types [7]. Despite their differences, all these frameworks share a multidimensional approach to DQ by addressing various issues related to missing data and data correctness. Yet, the structure of their concepts and proposed DQ indicators diverge considerably as the frameworks take specific demands from the underlying data sources into account.

Numerous tools have been developed to facilitate and partially automate data inspections and data quality assessments (DQA), ranging from stand-alone software solutions (e.g., [8–11]) to packages within existing programming environments, such as the R language (e.g., [12–18]). These developments are of great practical relevance, as they address the phases of data screening, data cleaning, and data preparation, which may even be more time-consuming than the actual statistical modeling itself [19,20].

Of particular relevance to DQA are developments in the R language [21]. Being a freely available programming environment, R has become one of the most popular statistical data processing environments worldwide. The base distribution of R contains functionalities for many statistical procedures; additional modules or extensions (termed "packages") are available for multiple purposes [22], such as exploratory data analysis and DQA. A review by Staniak et al. [20] included fifteen R packages that facilitate automated exploratory data analysis. Package features comprised, for example, univariate and multivariate data exploration, whole dataset summaries, checks on data validity, data cleaning options, and data transformation. Nevertheless, their scope varied considerably.

While exploratory data analysis overlaps with DQA, they are not identical. The former focuses on data properties and data descriptions in the broadest sense. In contrast, DQA is more concerned with measuring the "degree to which a set of inherent characteristics of data fulfils requirements" (ISO 8000 [23]). Thus, to perform DQA, it is not sufficient to rely on a complete data frame with study data. In addition, extensive metadata is needed to characterize further expected and required data properties (e.g., admissible values or ranges), as well as to identify deviations from the requirements [24]. Hence, software evaluation for exploratory data analysis provides only limited guidance on their ability to conduct DQA. Such a review of specific DQA capabilities of R packages is still missing.

This paper, therefore, contributes to existing works in several ways. Our primary objective is to provide a structured overview of the features of active R packages for assessing DQ. The focus is on the scope of possible DQA, using a DQ framework for observational studies [6] as a point of reference and example data from a cohort study as a test on real-world data. By doing so, we aim to identify potential enhancements to both the existing packages and the DQ framework. We also target links to DQ concepts, ease of use, metadata handling, and output type.

We start by describing our systematic search and defining the eligibility criteria (Section 2.1). Then, we explain how we evaluated and compared the packages (Section 2.2), including the reference framework (Section 2.2.1), general evaluation criteria (Section 2.2.2), and example data (Section 2.3). In the results, we report the process of selecting packages (Section 3.1), then we provide a comparison of DQ capabilities (Section 3.2), general features (Section 3.3), and package characteristics (Section 3.4). We discuss our findings considering the trade-offs between obtaining fast results and comprehensive analysis (Section 4.1) as well as the coverage of DQ aspects (Section 4.2). We provide a perspective in current real-world applications by contrasting our findings with other approaches to DQ in observational health studies (Section 4.3) and then discuss our approach's strengths and possible limitations (Section 4.4). Finally, we outline some future directions aiming to guide package development and DQA (Section 5).

2. Methods

As a first step, we checked whether there was sufficient progress over existing packages, as reported in [20], to merit a full review; hence, we did not set up a review protocol in advance. Upon a positive result, we formalized the approach and followed the PRISMA-ScR guidelines [25] to report our systematic search, assessment, and findings.

2.1. Search Strategy and Package Selection

We used a dual procedure to search and select packages, simultaneously conducting a manual and automated strategy to identify relevant packages. In the manual approach, we compiled a list of R packages related to DQA that were known to the authors, covered in

previous comparisons in the literature [16,18,20,26], or that were found by web searches while researching specific packages. For the automated process, we developed a script using R [21] and the package `pkgsearch` [27] to retrieve potential packages for assessing DQ from the Comprehensive R Archive Network (CRAN).

We refined the automated search during the process of package identification. Our final search strategy included six queries aiming at DQA, metadata use, and related concepts, which might incorporate DQ aspects (such as validating data, pre-processing, and exploratory tools). The search queries are reported in Appendix A. Duplicates within the search hits were removed using the R package `dplyr` [28]. To reduce the number of irrelevant packages, we compiled a list of exclusion terms (e.g., "air quality") in an iterative process, with caution to not accidentally exclude relevant packages, and applied it to the description texts of the previously identified packages. The R script with our final search strategy and filtering steps is available at https://github.com/JoanyMarino/RPackages4DQA (search performed on 18 January 2022). One reviewer (E.K.) screened these search results based on the available metadata to exclude packages unequivocally unrelated to DQA (i.e., packages that address none of the domains of the reference DQ framework), packages for a specific type of data or single quality indicators (e.g., only to check digit distributions), as well as archived packages. However, we did not exclude packages for which these assessments were inconclusive at this step.

We combined the results from the manual search approach and the automated search, and packages were subsequently screened in more detail by two reviewers (J.M. and E.K.) to revise their eligibility according to the pre-defined criteria (see list below). For this, we used the metadata and reference manuals available on CRAN. The details on the extracted information are given in Section 2.2.

We defined the following eligibility criteria for R packages:

1. The package is hosted on CRAN. CRAN is a global network of web servers that store up-to-date official releases of the R distribution, contributed packages, and documentation. By including only CRAN packages, we filter for packages that have passed basic technical quality control on different operating systems, i.e., Windows, macOS (Intel and ARM), and Linux.
2. The package is active on CRAN. CRAN runs regular tests on the hosted packages to ensure their technical functionality and stability across platforms and R versions. Packages that do not pass these tests and are not timely maintained are removed from CRAN. Considering active CRAN packages also means that they are ready-to-use, and the users can install them directly with base R (i.e., via the `install.packages` function or through the RStudio interface). We considered a package active if it was not "archived" on CRAN.
3. The package either explicitly targets DQ or has functionalities that are suitable for DQA. To evaluate and compare the scope of the packages, we matched their functionalities to a reference DQ framework for observational health research data [6]. We included packages that target at least three dimensions and four domains of the reference DQ framework (see Section 2.2.1 for explanations on the framework).
4. The package is not restricted to a specific field of application (e.g., air or water quality) nor a particular type of data (e.g., RNA-sequencing data, process data) to ensure applicability to a broad audience.
5. The package does not produce errors on basic output, such as a wrong number of observations when applied to real-world data, or stops unexpectedly due to possible internal errors.

2.2. Package Assessment and Feature Comparison

Before starting a detailed assessment of the packages, we devised a list of evaluation criteria (see Sections 2.2.1 and 2.2.2) in a spreadsheet and refined it as we conducted the assessment. For this, we extracted information from CRAN, such as the description, the date of the latest update, and links to related websites or publications. We assessed

functionalities of relevance based on the reference manual, the output, and, if available, the vignettes or website of the package. R Shiny applications were also installed and run in R. We conducted our evaluation using the latest (and active) release of a package, conducting the necessary updates until 7 March 2022. The assessment of each package was independently performed in parallel by two reviewers (J.M. and E.K.). Discussion with the other authors (S.S., C.O.S., and L.A.K.) solved disagreements and inconclusive cases, and ensured that the assessments conformed to the definitions of the reference DQ framework. To ensure consistency in the assessment across packages, four reviewers (S.S., L.A.K., J.M., and E.K.) re-assessed the evaluation criteria across all the packages included in this review. For the final synthesis of results, we report the number of packages which fulfil each evaluation criterion.

2.2.1. Data Quality Framework

We followed a framework for harmonized DQA in observational studies [6] to assess the scope of possible DQA. The detailed structure of the framework and definitions are provided in Appendix B.

The framework has been designed based on a literature review of DQ concepts, an empirical assessment of an existing DQ framework [7], and experiences conducting partially automated DQ assessments in cohort studies. Its hierarchical taxonomy consists of three levels: dimensions, domains, and DQ indicators (Table 1). The dimension level distinguishes qualitatively different aspects of DQ: integrity ("the degree to which the data conforms to structural and technical requirements"), completeness ("the degree to which expected data values are present"), consistency ("the degree to which data values are free of breaks in conventions or contradictions"), and accuracy ("the degree of agreement between observed and expected distributions and associations") [6]. Each dimension is divided into different DQ domains. These differ mainly in terms of the methodology used to evaluate DQ. For example, the consistency dimension distinguishes range and value violations from contradictions. The range and value violations domain compares single data values against metadata (e.g., range limits). In contrast, the contradictions domain jointly evaluates two or more different data values against a given rule that describes inadmissible combinations. Currently, a total of 10 domains are distinguished across the four dimensions [6].

The different DQ *indicators* are defined within domains. DQ indicators quantify detected or potential DQ issues (e.g., the number or proportion of data fields with range violations). With few exceptions, the computation of indicators requires the provision of metadata, which represent requirements on the data. However, based on a manual inspection, users can also use many statistics and graphs to infer DQ issues. For example, users can identify range violations with a histogram. Therefore, in addition to indicators, the DQ framework also defines *descriptors*, which provide insights on relevant data properties without quantitatively generating a measure of a deviation from a particular requirement. Having only a descriptor implies a manual processing step to conclude whether a DQ issue is present. Hence, having an indicator is generally preferable to a descriptor. During the package evaluation, we assessed which DQ dimensions and domains were covered by DQ indicators or descriptors. We display our findings at the level of dimensions and domains for a concise overview; in Appendix C, we provide a full summary of indicators and descriptors.

Table 1. Reference framework for harmonized data quality assessments [6,29] used to compare the selected R packages.

Dimension	Domain	Indicator
Integrity		
	Structural dataset error	
		1001: Unexpected data elements
		1002: Unexpected data records
		1003: Duplicates
	Dataset combination error	
		1004: Data record mismatch
		1005: Data element mismatch
	Value format error	
		1006: Data type mismatch
		1007: Inhomogeneous value formats
		1008: Uncertain missingness status
Completeness		
	Crude missingness	
		2001: Missing values
	Qualified missingness	
		2002: Non-response rate
		2003: Refusal rate
		2004: Drop-out rate
		2005: Missing due to specified reason
Consistency		
	Range and value violations	
		3001: Inadmissible numerical values (hard limits)
		3002: Inadmissible time-date values
		3003: Inadmissible categorical values
		3004: Inadmissible standardized vocabulary
		3005: Inadmissible precision
		3006: Uncertain numerical values (soft limits)
		3007: Uncertain time-date values
	Contradictions	
		3008: Logical contradictions
		3009: Empirical contradictions
Accuracy		
	Unexpected distribution	
		4001: Univariate outliers
		4002: Multivariate outliers
		4003: Unexpected locations
		4004: Unexpected shape
		4005: Unexpected scale
		4006: Unexpected proportions
	Unexpected association	
		4007: Unexpected association strength
		4008: Unexpected association direction
		4009: Unexpected association form
	Disagreement of repeated measurements	
		4011: Inter-Class reliability
		4012: Intra-Class reliability
		4013: Disagreement with gold standard

In the digital version, each indicator number links to its definition on the web page of the data quality framework, which contains further information and examples.

2.2.2. General Evaluation Criteria

We evaluated basic and pertinent aspects for conducting DQA related to the output type and the ease with which users can achieve at least elementary results. These comprised the following elements:

1. To assess whether DQ was at the root of a package, we checked if (i) the package description mentioned DQ in general and (ii) if the package was developed following a specific DQ concept.
2. Given the central role metadata has in DQA to compute indicators and not only descriptors, it is essential to evaluate if and how a package can handle metadata. We classify metadata, for instance, as any decision rule for a given data element (e.g., admissible values or ranges), the definition of missing value codes, or expectations on distributions. We looked at whether users can (iii) enter metadata through function calls or (iv) by importing a separate file. Entering metadata through a function call normally requires more programming skills compared to the use of separate files, where metadata could, for example, be provided in a spreadsheet type format.
3. As R users have different backgrounds and needs, we considered the mode of operation a key feature of the packages. We checked whether the packages (v) offer a graphical user interface (GUI), (vi) if they can be fully used via coding and enable a reproducible workflow, and (vii) allow triggering extensive output based on a single function.
4. We further evaluated the output formats for the DQA results, highlighting (viii) whether automatically generated reports can be produced. As reports, we considered all stand-alone files that the user can also view outside of R or RStudio. Useful features of such reports might include (ix) a dataset overview, (x) descriptive summary statistics, (xi) univariate graphs (e.g., histograms), and (xii) multivariate graphs (e.g., scatter plots and correlation heat maps).
5. Another desirable feature of a DQ report is (xiii) a grading or scoring of DQ issues to judge their severity automatically. A grading requires either preset or user defined categorization rules (e.g., the proportion of range violations per variable) to make a decision on whether the encountered number of findings is considered a problem.
6. We further noted whether a package offers (xiv) functionalities to handle string properties, such as checks for string lengths and upper or lower case.

2.3. Data and Application Example

Each of the identified R packages was applied to a publicly available dataset from the Study of Health in Pomerania (SHIP), a population-based cohort study [30,31]. This dataset comprises variables from a medical interview and from physical examinations, including: height, weight, waist circumference, systolic and diastolic blood pressure, smoking, intake of contraceptives, and marital status. The dataset is an anonymized 50% random subset (N = 2154) of the original sample from the baseline assessment of SHIP-0 from the years 1997 to 2001 [32]. This data application allowed for a better comparison of packages and evaluation of their functionalities, and furthermore, facilitated the detection of errors. All created scripts and reports, as well as the example data, are available at https://github.com/JoanyMarino/RPackages4DQA.

3. Results

3.1. Selected Data Quality R Packages

We present the number of packages considered during each phase of the selection process in Figure 1. Our systematic search for packages on CRAN via `pkgsearch` resulted in a list of 3921 packages. We first filtered this list automatically by keywords (excluding 769 duplicate and 2360 out of scope hits). Then, we manually screened the subset, excluding 669 out of scope hits (i.e., packages addressing none of the domains of the reference DQ framework, packages intended for DQA for a specific type of data or single quality indicators). Combining the remaining 123 search hits from the CRAN search with the

list of 54 previously known to the authors or identified from the literature resulted in 145 distinct packages. We conducted a detailed screening of these. The majority of them did not provide enough functionalities to assess DQ according to our criteria. For example, this applied to packages that were developed specifically for data management, data cleaning, or missing value imputation. Similarly, we excluded packages that solely focused on generating a descriptive data summary or summary tables in R without any graphs. Only 33 packages met the first 4 inclusion criteria. Out of these, `xplorerr` [33] had to be excluded, because it could only be partially executed, and we obtained errors when testing it with our example dataset (in the Shiny application for descriptive statistics, the overview for single variables gave the wrong number of observations; additionally, figures could only be partially generated). Likewise, `analyzer` [34] and `mdapack` [35] were excluded because they did not properly carry out the analysis of the sample dataset (it was not possible to generate a report for the SHIP data using `analyzer` as there were multiple issues with different data types causing errors that could not be resolved easily; the package `mdapack` excluded numeric variables coded as integers from the data overview, and a function for univariate analysis threw an error if less or more than four columns were included). Two additional packages were excluded because they were succeeded by other packages covered in this review: `editrules` [36] and `dataMaid` [16].

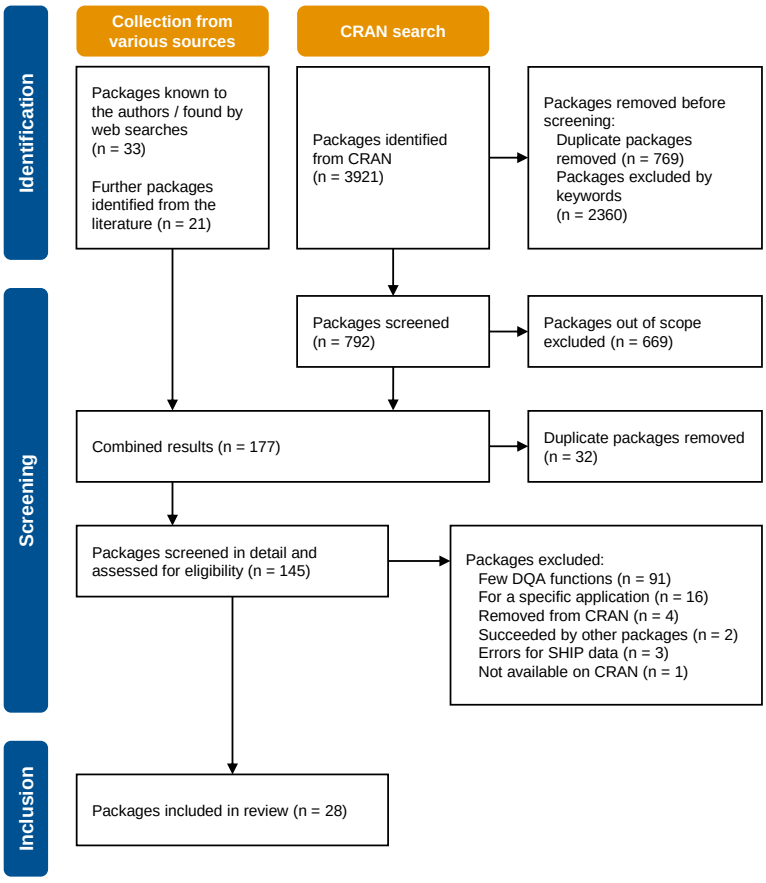

Figure 1. PRISMA flow diagram showing the number of packages considered during the three phases of our review process (Identification, Screening, and Inclusion). The Identification step comprises two independent starting points (collection from various sources and CRAN search) because we conducted two searches for packages in parallel. During the Screening phase, we combined the packages identified from both approaches. The main reason for excluding packages after screening was not meeting our inclusion criteria in the scope of functionalities for data quality assessments. In total, 28 packages remained at the end of the search, but we regard one of them (`errorlocate`) as an addition to another package (`validate`) and report their assessment together.

Our final comparison includes the following packages:

`assertable` [37]	`assertive` [38]	`assertr` [39]	`clickR` [40]
`DataExplorer` [41]	`dataquieR` [42]	`dataReporter` [16]	`DescTools` [43]
`dlookr` [44]	`DQAstats` [45]	`errorlocate` [46]	`ExPanDaR` [47]
`explore` [48]	`funModeling` [49]	`inspectdf` [50]	`IPDFileCheck` [51]
`MOQA` [52]	`mStats` [53]	`pointblank` [54]	`sanityTracker` [55]
`skimr` [56]	`SmartEDA` [57]	`StatMeasures` [58]	`summarytools` [59]
`testdat` [60]	`validate` [18]	`visdat` [61]	`xray` [62]

The package `errorlocate` builds on the package `validate`; therefore, we assessed them jointly. The majority of functions of interest were provided by `validate`. Thus, we report our findings for a total of 27 R packages. The results of our assessment are included in detail in Appendix C.

3.2. Data Quality Capabilities Comparison

The coverage of DQ domains across the packages varies considerably (Figures 2 and 3); a detailed overview is provided in Tables A1–A7 of the Appendix. While some domains are covered by almost all packages, such as crude missingness or range and value violations, other domains receive little coverage of package built-in functionalities, such as qualified missingness or contradictions (Figure 3). Only one of the packages contains functionalities to directly target repeated measurements. Packages that are based on rule checking (category 3 in Section 3.3) incorporate on average more indicators than packages focusing on descriptive data overviews and data exploration (category 2 in Section 3.3). Figure 3 indicates that the packages cover at most eight of the ten domains of the framework, while almost half cover at least six domains.

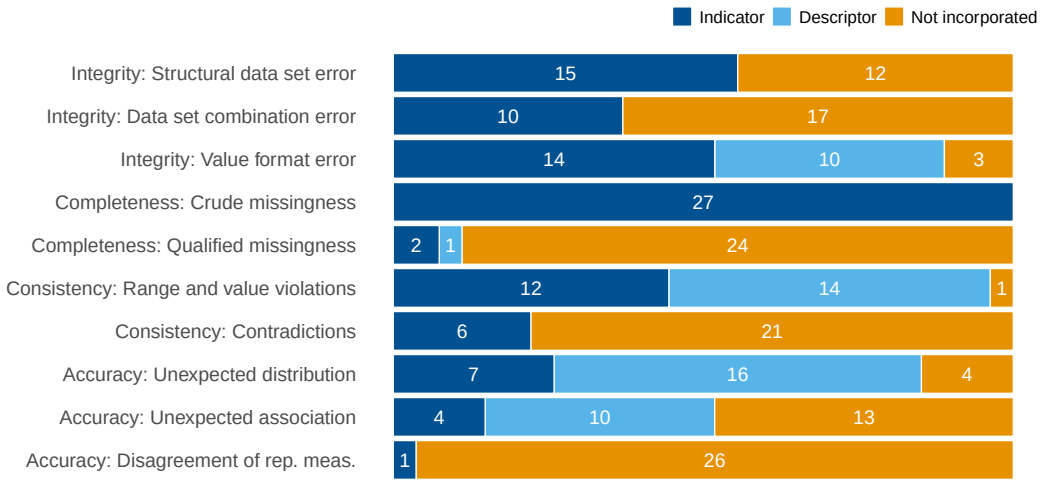

Figure 2. Number of packages with relevant output to detect issues in distinct data quality domains (Section 2.2.1) with respect to the total 27 packages included in this review.

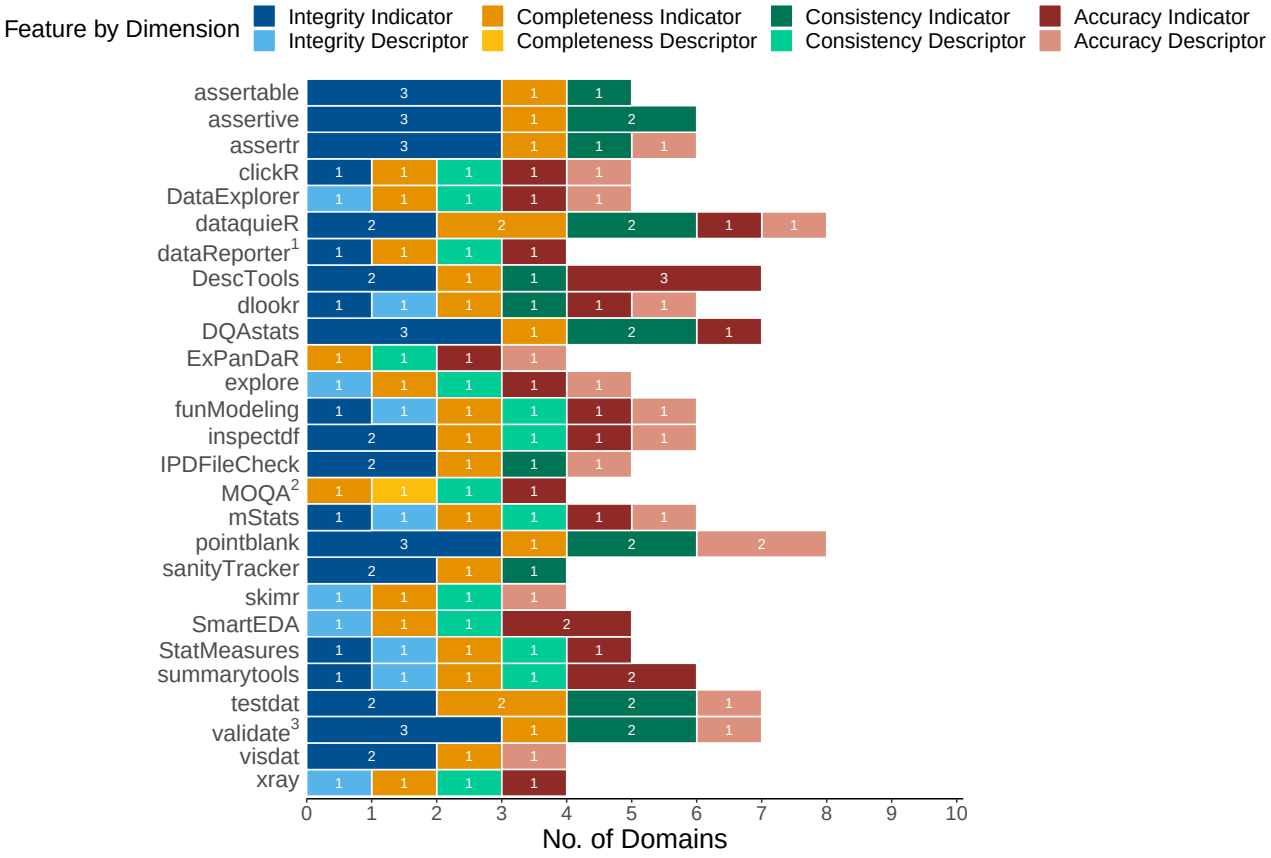

Figure 3. Number of data quality dimensions and domains for which inferences can be made based on the output of each of the 27 packages included in this review. For each package, the numbers correspond to the number of covered domains within each dimension. If a domain is covered by indicators and descriptors, the descriptors are not mentioned separately, yet Tables A1–A7 provide the detailed information. [1] identical with `dataMaid`, [2] first published as `mosaicQA`, [3] includes integration with `errorlocate`.

3.3. General Feature Comparison

The packages vary greatly regarding most aspects of the comparison, such as theoretical background, programming requirements, data handling, and output (Figure 4). A detailed table with the description of each package is available in Appendix C.

In total, nine packages explicitly refer to DQ, but only three were developed with reference to a published DQ framework (`dataquieR` [6], `DQAstats` [17], and `MOQA` [7]). Merely three packages offer a GUI, yet about half (n = 15) provide an option to use a single function to generate comprehensive output. For the most part, programming knowledge is necessary to perform a wide range of DQ checks. A report outside an R console, for example as PDF or HTML files, is provided by 13 packages.

While the majority of packages (n = 16) allows at least for some metadata to be provided through functions, only few are able to cover more than four domains with at least some indicators (illustrated in Section 3.3). Only four packages handle compiled metadata in a separate file. Six packages provide a grading of DQ issues for at least selected indicators.

Most packages (n = 19) provide summaries of a given dataset; for example, the number of variables and data records, data types, and the percentage of missing values. Many (n = 18) also compute summary statistics and (n = 17) univariate descriptive statistics graphs (e.g., mean, median, mode, standard deviation, quartiles, and distributional characteristics) that may serve as descriptors, but only a third (n = 10) provides multivariate visualizations. All packages handle numerical values, but only a minority (n = 13) offers functionalities to

describe and assess strings, such as consistent upper and lower case usage or data fields with blanks.

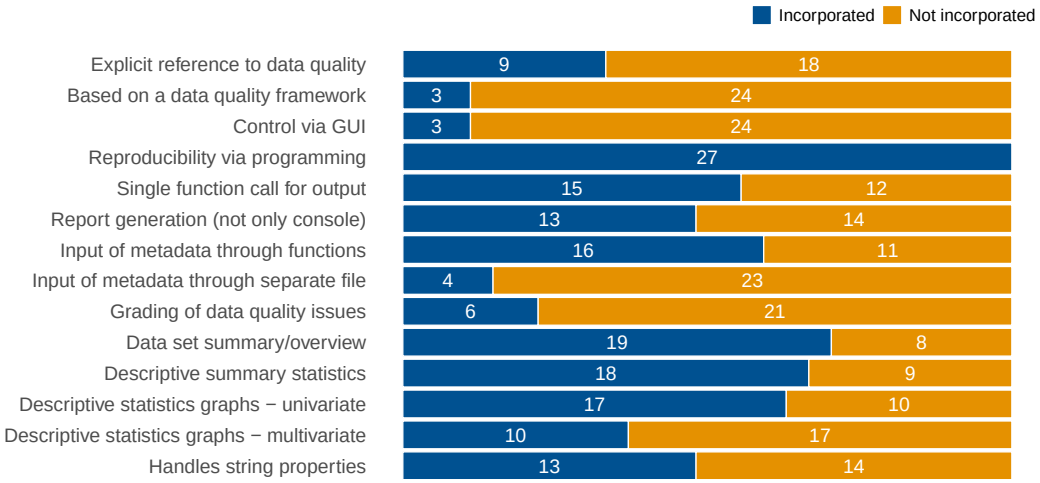

Figure 4. General feature comparison across the 27 packages included in this review (Section 2.2.2).

3.4. Package Characteristics

The R packages in our review can be assigned to one of three categories:

1. Packages that combine descriptive features with targeted DQ checks (Figure 5A);
2. Packages that focus on statistical overviews and on data exploration (Figure 5B);
3. Packages that perform highly focused checks based on an input of rules (Figure 5C).

The first category comprises packages that combine a wide range of descriptive features with the possibility to perform targeted checks. They explicitly focus on DQA and are mostly suitable for users with limited programming skills. `dataquieR`, `dataReporter`, `DQAstats`, `MOQA`, and `pointblank` belong to this category, with the latter being the most popular package (Figure 5A). `pointblank` offers two different types of reports: a descriptive report, which can be produced through a single function, and a report on DQ checks, which have to be defined by the user first. The latter approach is similar to the rule-checking packages (in category 3) and enables customized DQ reports creation. The reports generated by `dataReporter` can be compiled in different formats (R Markdown, PDF, HTML, and MS-Word) and, with additional programming, extended to include further checks and visualizations. `dataquieR` can automatically generate reports in the form of an interactive dashboard, but manual reports of varying scope may also be generated using R Markdown. For reporting, `dataquieR` uses a large set of helper functions to ensure its robustness with deficient study data and metadata. `MOQA` produces a separate PDF report for each variable, potentially resulting in many files. Only `DQAstats` and `dataquieR` enable users to import metadata from a separate file (i.e., a CSV file with a specific structure). `dataquieR` and `MOQA` allow users to specify missing value codes. However, these implementations differ: in `MOQA`, only one general missing threshold for numerical variables and only one list of valid categories for categorical variables can be defined, while in `dataquieR`, these can be variable-specific. `MOQA` crashed if inadmissible categorical values exist in the data, but manual pre-processing can circumvent this issue. In this group, the packages `pointblank` and `dataquieR` cover the largest number of domains.

Most packages belong to the second category (Figure 5B). Two of them offer a GUI: `ExPanDaR` and `explore`. The former exports reports as R Markdown Notebooks, while the latter generates HTML reports using R Markdown. The package `dlookr` offers two different types of reports with detailed information on DQ issues and exploratory data analysis, in the form of PDF or HTML files. In contrast, `summarytools` generates a brief but concise HTML report, which can be triggered by a single function. `DataExplorer` has a similar feature, but produces a more comprehensive report than `summarytools`, including

a wider range of figures, such as correlation heat maps. The package `SmartEDA` produces reports comparable to that of `DataExplorer`, but includes pairwise scatter plots instead of a correlation heat map and a more extensive data overview. The packages `clickR`, `mStats`, `skimr`, `StatMeasures`, and `xray` mainly give a descriptive variable overview, but do not include stand-alone reports. The package `visdat` is similar in this regard, but follows a unique approach by providing mainly graphical output. While the package `inspectdf` equips each of its functions with a matching plot, it differs from `visdat` in providing the output also in a tabular format.

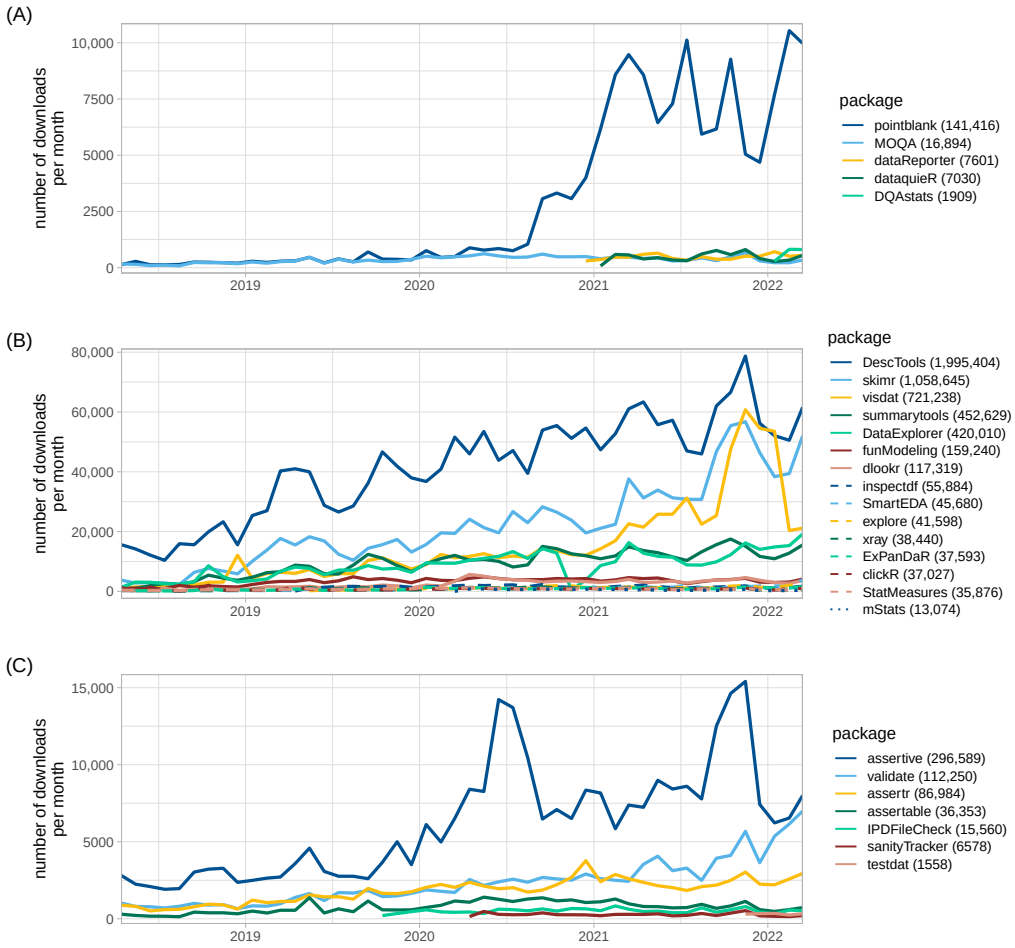

Figure 5. CRAN downloads per month from April 2018 to March 2022 for (**A**) packages with focus on DQA, (**B**) packages for descriptive statistics, data exploration, and statistical analysis, and (**C**) rule-based packages. Numbers in the legend report the total number of downloads in this time period (numbers of downloads retrieved using `cranlogs` [63]).

Regarding additional functionalities for packages in the second category, `inspectdf` and `visdat` are able to compare two datasets, but `inspectdf` offers more functionalities in this regard. The largest collection of functions is provided by `DescTools`, which includes descriptive data overviews, data cleaning, visualizations, and statistical tests. It has no function to generate a complete report automatically, but it offers functions to export R output to MS Word and PowerPoint. The packages `ExPanDaR` and `explore` also include additional features for statistical inference, which were not assessed in our comparison, such as regression analysis and decision trees, whereas `funModeling` provides additional functions to assist with variable selection and data preparation for statistical modeling. A principal component analysis is included in the report produced by `DataExplorer`.

According to CRAN download statistics, `DescTools`, `skimr`, and `visdat` are the most popular packages of our review (Figure 5B).

The third category of packages follows the idea of a rule-based checking, similar to software unit-test-functions [64]. For example, `sanityTracker` provides functions to check if values in a data column lie within a certain range or conform to a set of predefined values. With a specific function call, all previously defined checks can be executed, and the output indicates which checks on the data failed. Since the user has to set up the rules, these packages can usually be applied to any kind of data. One exception may be `IPDFileCheck`, which is intended for patient data and provides specialized functions to check age and gender data columns, optionally considering a code for missing values. However, it is not possible to input a list of several missing value codes indicating different reasons for missingness. The only package of this category which offers this feature is `testdat`. The package `assertive` provides numerous specialized check functions, among them tests for specific string patterns (email addresses, ISBN codes, US postal codes, etc.). Some packages do not only report if a rule was violated, but also prompt an error message or warning (e.g., `assertable`, `assertive`, and `assertr`), and some (e.g., `testdat`) provide options for integration in test frameworks such as `testthat` [65], which can run a set of tests and summarize their results. The package `validate` is comparable to `testdat`, but can import rules from other files (e.g., YAML and SDMX) and comprises notification management options. Together with the add-on packages `errorlocate` and `lumberjack` [66], `validate` can be used for traceable data curation. The most frequently downloaded packages from this last category are `assertive`, `validate`, and `assertr` (Figure 5C).

4. Discussion

This comprehensive review inspected R packages relevant to DQA and systematically mapped their functionalities against a DQ framework for observational health studies [6]. Our review identified many R packages that efficiently support DQA, covering all framework dimensions and domains. However, few packages offer comprehensive options to assess a wide range of DQ indicators and the scope of packages varies considerably in terms of functionalities and ease of use. Output is often descriptive and requires a manual inspection to identify issues of concern, thus hampering automated workflows. This shortcoming reflects a lack of appropriate metadata management in many current studies to describe requirements on the data [19,24]. The DQ framework could incorporate the package's functionalities at the dimension and domain level. However, some expansions at the level of indicators are indicated.

4.1. Fast Results vs. Thorough Analysis

There are different approaches to targeting DQA. One important use case is to get a quick insight into a dataset with little programming skills and without using much formal information about data properties and requirements. Packages such as `DataExplorer` [41] follow this approach, and are a viable option to obtain informative insights. Accessibility is also supported with GUIs, but this may impair reproducible workflows if the selected options or pre-processing steps, such as filtering data, are not recorded. However, one package (`ExPanDaR` [47]) aims at reconciling a GUI with reproducibility by providing an interactive notebook.

In this first use case, simplicity usually comes at the expense of having no or few options to explicitly check the requirements on the targeted dataset. Without an explicit definition of permissible values, reference distributions, or expected associations, a package may only use generic algorithms to numerically or graphically describe data properties. The person inspecting the results must then decide whether the finding truly reflects a DQ problem. This way, judgements on DQ findings are informal, based on implicit assumptions, and therefore, they are prone to error [6]. However, a lack of metadata usage may reflect that only limited related information is available in the data dictionaries underlying most observational studies [24].

A second use case is to obtain a highly focused insight on rule violations in a dataset. However, the additional information required to perform the checks in this approach commonly conflicts with ease of use. For example, while `validate` [18] and `errorlocate` [46] allow for calculating many indicators, the user must explicitly write the so-called "rules" (equivalent to metadata) to evaluate each indicator for the corresponding variable. Similar packages are `testdat` [60] and `sanityTracker` [55]. Moreover, for some packages, the user must understand their particular grammar. An example is `pointblank` [54]. In this package, the user needs to be familiar with specific conventions to perform the assessments: the indicators are stored in an "agent" that needs to be "interrogated" to perform the evaluation. Despite the detailed documentation provided with `pointblank`, understanding the grammar is time-intensive for any user. The main advantages of these packages are the transparency and reproducibility of their assessments, as well as the possibility of extending the range of checks with user-defined requirements. Moreover, these features allow integrating DQA into automated workflows, which is impossible with packages from the first use case. A disadvantage is the low interoperability of check rules, as the syntax may need to be tailored to fit individual package requirements.

From the previous considerations, it is clear that meaningful metadata in a machine-readable format is a key to comprehensive DQ reports [24]. Therefore, a third use case reconciles ease-of-use with setting up extensive rules. One possibility of achieving this is by separating the metadata setup from formal programming, an approach followed by `dataquieR` [42] and `DQAstats` [45,67]. Both packages can use metadata for DQ in a basic spreadsheet format, which lowers the technical barriers to adding and modifying content. Subsequently, basic code suffices to conduct extensive reports. However, agreeing on common data formats for storing metadata remains an open issue. For this reason, it is advisable to take into account existing standards in the health sciences, such as HL7 FHIR and CDISC-ODM [68], as well as common data dictionary formats, for example REDCap [69].

4.2. Coverage of Data Quality Aspects

Using an existing DQ framework [6] as a point of reference for our comparison enabled us to structure the packages' features and to identify existing gaps. Overall, the functionalities of the packages could be fitted into the framework, and there was no need to expand in dimensions or domains. However, not all functionalities could be accommodated well at the indicator level. This concerns the capability to check for unexpected variable names, loners (i.e., values that occur only once for a data element), and the occurrence of specific and maybe unwanted values (zeros, negatives, and infinity). For example, zero values may be regarded as inadmissible for a given data element. In this case, the proportion of zero values is an indicator for inadmissible numerical values (ID 3001). If an excess number of zero values is of concern, the indicator "Unexpected Shape" (ID 4004) applies. Furthermore, there may be multiple ways to calculate metrics for virtually all the indicators of the DQ framework. A need to expand the framework resulted from the capabilities of some packages to check for correct value formats, such as email addresses, postal codes, or telephone numbers. While fitting within the integrity domain "value format error", none of the three existing indicators of the framework provides a good match. Thus, we recommend introducing a novel indicator, "value format mismatch", which refers to a mismatch between an observed and an expected value format that is not captured by a "data type mismatch". We discuss further issues in Section 4.4.

4.2.1. Missing Related Implementations

Ideally, any missing value in a dataset has been assigned a code that indicates the reason for why it is missing (i.e., "missing by design", "technical error", and "refusal"). This is the only way to correctly assess key figures, such as response rates or refusal rates. In contrast, representing missing values only with *NAs*, which is frequently the case, can be misleading. Consider, for example, observing *NAs* in an item on the number of cigarettes

smoked per day for a person who previously stated to be a non-smoker. The value is in fact not missing but zero. Therefore, the DQ framework provides a distinction between "crude missingness", where DQA cannot rely on missing codes, as opposed to "qualified missingness", which computes indicators under the precondition of knowing missing reasons [6]. According to our review, almost all packages handle only the crude missingness approach. Consequently, more common standards are needed to code missing values and make them available in packages [70]. Note that qualified missing codes are not the same as value codes of categorical variables, since the former must be identified as such by a package. Thus, a package supporting value labels for a bar chart does not qualify for the qualified missingness indicator, as it does not interpret these values differently from "normal" categories. Another way to improve missing assessments may be to incorporate statistical methods which assess missing data structures, such as clustering methods [71,72].

4.2.2. Consistency-Related Implementations

The reference DQ framework distinguishes two classes of indicators in the range and value violations domain [6]. The first corresponds to inadmissible values, meaning data values that are not permissible. The second refers to uncertain values, which are unlikely data values because they are outside the expected ranges. Despite their entirely different meaning, their computation is identical, and a distinction between the two can only be achieved through the metadata. Thus, during our evaluation, when a package had a function to check for values outside a defined range, we have included it as fulfilling both indicators. This is justifiable particularly for rule-based programs, where users may simply apply rules in the intended context.

Regarding contradictions, some value combinations may be impossible by the very nature of the targeted variables (e.g., follow-up examination date must be after the baseline examination date; date of birth must always be the same). Others may be very unlikely to happen (e.g., if the gender is male, no pregnancies should be listed, but maybe a study participant did undergo gender transitioning). The DQ framework classifies the former as logical and the latter as empirical contradictions. Although this distinction is solely based on the semantics of such rules, the method is the same for both contradiction types. Hence, we counted both in our assessment if a package is capable of checking these types of rules.

4.2.3. Accuracy-Related Implementations

Since most packages comprise univariate statistics and almost half also include multivariate statistics, users may trivially use their output to detect some accuracy issues. For example, a histogram may be used to identify unexpected locations, shapes or scales, and aspects of other dimensions, such as range violations. Yet, these descriptors rely on the user inspecting the results, and the quality of the output strongly influences options to detect issues reliably. Despite this, indicators within the accuracy dimension were the least covered by the packages' functionalities. This is likely the case because most related checks are not of a simple Boolean type, but require a range of statistical analyses and metadata on expected distributions and associations to make decisions on the presence of some issue.

4.3. Data Quality in Electronic Health Records

A goal of many research networks in the medical domain is to leverage the potential of electronic health records by providing solutions that enable joint analyses [73–77]. In this way, information that was initially gathered for a specific purpose (e.g., clinical documentation or billing) can be used for a different goal (e.g., scientific studies). Such secondary data usage raises the concern that the data may harbour potential flaws that could influence the findings because it was not systematically collected [78]. This concern imposes further DQ requirements that should be met when analysing such data [1–3,5]. The international Observational Health Data Science and Informatics collaborative [14], for example, provides two R-based solutions [79,80] for assessing descriptive statistics and conducting DQA of the data stored in their common data model (CDM) "OMOP" [81].

Compared to other solutions for storing and managing data for observational research studies, these functionalities are very practical. However, users must convert the data to comply with the OMOP CDM to use these tools. In contrast, the R packages presented in our review can be applied to a wide variety of source data without such requirements.

4.4. Strengths and Limitations

According to CRAN, more than 18,000 packages are available [82], with many more being published every week. We have tried to cover all potentially relevant packages by combining different search strategies. However, we may have overlooked some, particularly as R packages are hard to review systematically on CRAN due to limited search functionalities. Our search hits, using the package `pkgsearch`, were based on the title and (short) description of the packages as provided by the package authors. Therefore, we additionally curated a list of relevant packages from the literature and other sources as outlined in the methods. Excluding packages for DQA in specific fields might potentially have led to excluding well-developed packages with additional functionalities.

We classified packages into three broad groups according to their main features and description. This classification is straightforward for most packages. The packages that were developed following a DQ concept (i.e., `MOQA`, `DQAstats`, and `dataquieR`) are unequivocally in the category of packages focused on DQA (category 1 in Section 3.4). Similarly, packages based on assertion checks (e.g., `testdat` and `assertr`) can be trivially classified as rule-based (category 3 in Section 3.4). However, other packages that provide diverse features could fit multiple categories. A prominent example is `pointblank` because it is built as a rule-based package, but many of its functions and output are specifically designed for DQA. This also applies to `validate`, yet the packages were classified differently because of the DQ-specific reporting features provided by `pointblank`. The packages `dlookr` and `DescTools` incorporate various checks for DQ as well, but their scope is broader and extends to data exploration and statistical analyses (category 2 in Section 3.4). We decided the final categorization of packages in these uncertain cases based on their description and documentation. Nevertheless, we emphasize that this classification should be considered broadly and flexible.

There was some ambiguity regarding the classification of package functionalities concerning the reference DQ framework. We resolved disagreements through extensive consensus processes. However, certain aspects may remain debatable. This issue becomes particularly evident with packages that perform rule-based checking, such as `assertr` or `validate`. With additional base R input, experienced users may effectively use these packages to address all framework indicators. However, this disagrees with the idea of having packages that facilitate DQA. Therefore, our classification focused on functionalities provided within packages, allowing only additional basic input, such as regular expressions, while excluding base R. Another example is in the "disagreement of repeated measures" domain. We expected the package to recognize and handle repeated measurements to assign a given functionality to this domain. However, any scatter plot or correlation trivially provides information about repeated measurements, and we could also have decided to qualify such functionalities as descriptors. Nevertheless, our rationale was to focus on features directly implemented by the package without relying on user customization to avoid these nuances.

Furthermore, functionalities are implemented differently across packages, but their binary mapping against the reference DQ framework did not capture such heterogeneity. For example, regarding the indicator "inadmissible standardized vocabulary", `DescTools` handles a specific check for ISO 3166-1 country codes, while `validate` can import and contrast any SDMX code list from the web. We consider both implementations as standardized vocabulary because they are based on explicit ontologies, but their difference in breadth is evident. Another example is the varying quality of the graphical output across packages. Similarly, some packages provide very specific implementations. For instance, `assertive` implements an indicator for dates that are in the past, and while this can be considered nar-

row for inadmissible or uncertain time-date values, it is nevertheless an indicator, and we considered these cases as such. We did not rate better or worse implementations, as this is subject to the individual use case, but rather concentrated on the coverage of different aspects of the DQ framework. Likewise, we did not target all package features of potential relevance. For example, the user-friendliness of a GUI is important, but we did not include it in our assessment. Our focus was on DQ, not on exploratory data analysis [20] or initial data analysis in general [19]. Accordingly, we did not assess all options of data exploration as offered by some packages.

A final concern is the robustness of package functionalities, but thoroughly assessing this aspect would require different datasets with a systematic variation of errors. While our application to SHIP data worked smoothly with most packages, some deficits became transparent.

5. Conclusions

Many R packages are available to examine data properties in an automated and efficient fashion. Therefore, we strongly recommend using them before setting up basic R code for data screening and DQA from the beginning. However, most functionalities of the assessed packages are exploratory. The outputs on metrics of deviations between observed and expected data characteristics are limited, and extensive programming may be necessary to obtain them. This reflects a deficiency in our science, where most studies offer a narrow scope of metadata to describe expected or required data properties. Developing common metadata standards for DQA is an important step towards overcoming this limitation. This could be supported by R routines to manage and validate metadata (see, for example, `dccvalidator` [83]). Conceptually, while the DQ framework could easily incorporate package functionalities at the dimension and domain level, our work illustrated the need to expand the scope of indicators.

Author Contributions: Conceptualization, J.M., E.K. and C.O.S.; methodology, E.K. and J.M.; validation, C.O.S.; formal analysis, J.M. and E.K.; writing—original draft preparation, J.M., E.K. and C.O.S.; writing—review and editing, J.M., E.K., S.S., L.A.K. and C.O.S.; visualization, J.M. and E.K.; supervision, C.O.S.; project administration, J.M., E.K. and C.O.S.; funding acquisition, C.O.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the DFG (NFDI 13/1, SCHM 2744/9-1, SCHM 2744/3-1), by the European Union's Horizon 2020 research and innovation programme under grant agreement No 825903 (euCanSHare project), as well as in part by the German Federal Ministry of Education and Research (BMBF) within the Medical Informatics Initiative (MIRACUM Consortium, FKZ: 01ZZ1801A).

Data Availability Statement: Publicly available datasets were analyzed in this study. The used dataset can be found here: https://dfg-qa.ship-med.uni-greifswald.de/ExampleDataDescription.html.

Acknowledgments: We express our gratitude to Johannes Darms for their critical feedback on the manuscript. We also thank two anonymous reviewers for their insightful comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest. However, we note that the data quality framework was developed in the Greifswald working group as well as the dataquieR package. L.A.K. developed the DQAstats package. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

CRAN	Comprehensive R Archive Network
CDM	Common Data Model
DQ	Data Quality
DQA	Data Quality Assessment
GUI	Graphical User Interface

Appendix A. Search Queries

We used the following search queries to find DQA-relevant packages on CRAN based on their titles and descriptions:

1. (data OR dataset) AND quality
2. quality AND indicator
3. (data OR dataset OR quality) AND (assessment OR control OR check OR monitor OR manage OR report OR summary OR summarise OR curation OR screening OR visualise)
4. (data OR dataset) AND (clean OR validate OR preprocess OR process OR consistent OR inconsistent)
5. exploration OR exploratory
6. metadata

Since the search server uses word stems, it suffices to search for example for "clean" to also match "cleaning", and in the same way "summarise" also matches with "summarize". However, some related search phrases still resulted in different search results in our test runs. That is why, for instance, we included both the terms "exploration" and "exploratory", or "summary" and "summarise".

Appendix B. Reference Data Quality Framework

Our reference data quality framework, by Schmidt et al. [6], is hierarchically structured into three levels: dimensions, domains, and indicators. The four dimensions of the framework are defined in [6,29] as follows:

Integrity: The degree to which the data conforms to structural and technical requirements.
Completeness: The degree to which expected data values are present.
Consistency: The degree to which data values are free of breaks in conventions or contradictions.
Accuracy: The degree of agreement between observed and expected distributions and associations.

The integrity dimension entails three domains:

Structural dataset error: The observed structure of a dataset differs from the expected structure.
Dataset combination error: The observed correspondence between different datasets differs from the expected correspondence.
Value format error: The technical representation of data values within a dataset does not conform to the expected representation.

The completeness dimension consists of two domains:

Crude missingness: Metrics of missing data values that ignore the underlying reasons for missing data.
Qualified missingness: Metrics of missing data values that use reasons underlying missing data.

The consistency dimension entails the following two domains:

Range and value violations: Observed data values do not comply with admissible data values or value ranges.
Contradictions: Observed data values appear in impossible or improbable combinations.

Lastly, the accuracy dimension contains three domains:

Unexpected distribution: Observed distributional characteristics differ from expected distributional characteristics.
Unexpected association: Observed associations differ from expected associations.
Disagreement of repeated measurements: Disagreement between repeated measurements of the same or similar objects under specified conditions.

Table 1 summarizes the structure of the data quality framework and provides links to the definitions of the indicators.

Appendix C. Package Assessment Results

Table A1. Data quality assessment capabilities by package.

Criteria	assertable	assertive	assertr	clickR
Explicit reference to data quality				yes
Based on a data quality framework				
Control via GUI				
Reproducibility via programming	yes	yes	yes	yes
Single function call for output				
Report generation (not only console)				
Input of metadata through functions	yes	yes	yes	yes
Input of metadata through separate file				
Grading of data quality issues				
Dataset summary/overview				yes
Descriptive summary statistics				yes
Descriptive statistics graphs—univariate				yes
Descriptive statistics graphs—multivariate				
Handles string properties		yes		yes
Integrity: Structural dataset error	1001, 1002, 1003, 100X	1003	1001, 1003, 100X	
Integrity: Dataset combination error	1005	1004, 1005	1005	
Integrity: Value format error	1006	1006, 100Y, 1007	1006	1006, 1006D, 1007
Completeness: Crude missingness	2001	2001	2001	2001
Completeness: Qualified missingness				
Consistency: Range and value violations	3001, 3002, 3003, 3006, 3007	3001, 3002, 3003, 3005, 3006, 3007	3001, 3003, 3006	3001D, 3002D, 3003D, 3006D, 3007D
Consistency: Contradictions		3008, 3009		
Accuracy: Unexpected distributions			4001D, 4002D	4001, 4001D, 4002, 4003D, 4004D, 4005D, 400X
Accuracy: Unexpected associations				4007D, 4008D
Accuracy: Disagreement of rep. meas.				

Data quality indicator numbers followed by D mark functionalities which are not included as an indicator, but only as a descriptor. Extending the reference data quality framework, we included three additional codes: the code 100X denotes that the package provides functions to detect unexpected variable names, and 100Y denotes functions to detect data format mismatches. The code 400X indicates a functionality to detect loners (i.e., values which occur only once).

Table A2. Data quality assessment capabilities by package (continued).

Criteria	DataExplorer	dataquieR	dataReporter [1]	DescTools
Explicit reference to data quality		yes	yes	
Based on a data quality framework		yes [6]		
Control via GUI				
Reproducibility via programming	yes	yes	yes	yes
Single function call for output	yes	yes	yes	
Report generation (not only console)	yes	yes	yes	yes
Input of metadata through functions		yes	yes	yes
Input of metadata through separate file		yes		
Grading of data quality issues	yes	yes		
Dataset summary/overview	yes	yes	yes	yes
Descriptive summary statistics			yes	yes
Descriptive statistics graphs—univariate	yes	yes	yes	yes
Descriptive statistics graphs—multivariate	yes	yes		yes
Handles string properties			yes	yes
Integrity: Structural dataset error		1001, 100X		
Integrity: Dataset combination error				1004, 1005
Integrity: Value format error	1006D	1006, 1008	1006D, 100Y, 1007	1006, 1006D, 100Y, 1007
Completeness: Crude missingness	2001	2001	2001	2001
Completeness: Qualified missingness		2002D, 2003D, 2004D, 2005		
Consistency: Range and value violations	3001D, 3003D, 3006D	3001, 3002, 3003, 3006, 3007	3001D, 3002D, 3003D, 3006D, 3007D	3001, 3002D, 3003, 3004D, 3005, 3006, 3007D
Consistency: Contradictions		3008, 3009		
Accuracy: Unexpected distributions	4001D, 4003D, 4004D, 4005D, 4006D, 400XD	4001, 4001D, 4002, 4002D, 4003, 4003D, 4004, 4004D, 4005, 4005D, 4006, 4006D	4001D, 4003D, 4004D, 4005D, 4006D, 400X	4001D, 4002, 4003, 4003D, 4004D, 4005D, 4006, 4006D, 400XD
Accuracy: Unexpected associations	4007D, 4008D, 4009D	4007D, 4008D, 4009D		4007, 4007D, 4008D, 4009D
Accuracy: Disagreement of rep. meas.				4011, 4012, 4013

Data quality indicator numbers followed by D mark functionalities which are not included as an indicator, but only as a descriptor. Extending the reference data quality framework, we included three additional codes: the code 100X denotes that the package provides functions to detect unexpected variable names, and 100Y denotes functions to detect data format mismatches. The code 400X indicates a functionality to detect loners (i.e., values which occur only once). [1] identical with `dataMaid`.

Table A3. Data quality assessment capabilities by package (continued).

Criteria	dlookr	DQAstats	ExPanDaR	explore
Explicit reference to data quality	yes	yes		
Based on a data quality framework		yes [17]		
Control via GUI		yes *	yes	yes
Reproducibility via programming	yes	yes	yes	yes
Single function call for output	yes	yes	yes	yes
Report generation (not only console)	yes	yes	yes	yes
Input of metadata through functions			yes	
Input of metadata through separate file		yes	yes	
Grading of data quality issues	yes			
Dataset summary/overview	yes	yes	yes	yes
Descriptive summary statistics	yes	yes	yes	yes
Descriptive statistics graphs—univariate	yes		yes	yes
Descriptive statistics graphs—multivariate	yes		yes	yes

Table A3. *Cont.*

Criteria	dlookr	DQAstats	ExPanDaR	explore
Handles string properties		yes		yes
Integrity: Structural dataset error	1003	1003		
Integrity: Dataset combination error		1004D, 1005		
Integrity: Value format error	1006D	1006D, 100Y, 1007, 1008		1006D
Completeness: Crude missingness	2001	2001	2001	2001
Completeness: Qualified missingness				
Consistency: Range and value violations	3001, 3002D, 3003D, 3005D, 3006D, 3007D	3001, 3002, 3003, 3005, 3006, 3007	3001D, 3003D, 3006D	3001D, 3002D, 3003D, 3006D, 3007D
Consistency: Contradictions		3008, 3009		
Accuracy: Unexpected distributions	4001D, 4003D, 4004D, 4005D, 4006D, 400X	4001D, 4003D, 4005D, 4006D, 400XD	4001D, 4003D, 4004D, 4005D, 4006D, 400XD	4001D, 4003D, 4004D, 4005D, 4006D, 400XD
Accuracy: Unexpected associations	4007D, 4008D, 4009D		4007D, 4008D, 4009D	4007D, 4008D, 4009D
Accuracy: Disagreement of rep. meas.				

Data quality indicator numbers followed by D mark functionalities which are not included as an indicator, but only as a descriptor. Extending the reference data quality framework, we included three additional codes: the code 100X denotes that the package provides functions to detect unexpected variable names, and 100Y denotes functions to detect data format mismatches. The code 400X indicates a functionality to detect loners (i.e., values which occur only once). [∗] `DQAstats` was equipped with a GUI by an add-on package (`DQAgui` [67]), which was first published on CRAN at 11 February 2022. The functionalities provided by this new package were not evaluated, because it had been published after our final search on CRAN.

Table A4. Data quality assessment capabilities by package (continued).

Criteria	funModeling	inspectdf	IPDFileCheck	MOQA [2]
Explicit reference to data quality				yes
Based on a data quality framework				yes [7]
Control via GUI				
Reproducibility via programming	yes	yes	yes	yes
Single function call for output				yes
Report generation (not only console)				yes
Input of metadata through functions			yes	yes
Input of metadata through separate file				
Grading of data quality issues				
Dataset summary/overview	yes		yes	
Descriptive summary statistics	yes	yes	yes	yes
Descriptive statistics graphs—univariate	yes	yes		yes
Descriptive statistics graphs—multivariate	yes			
Handles string properties	yes		yes	
Integrity: Structural dataset error			1001, 100X	
Integrity: Dataset combination error	1004, 1005	1004D, 1005, 1005D		
Integrity: Value format error	1006D	1006, 1006D	1006	
Completeness: Crude missingness	2001	2001	2001	2001
Completeness: Qualified missingness				2005D
Consistency: Range and value violations	3001D, 3002D, 3003D, 3006D, 3007D	3001D, 3003D, 3006D	3001, 3003, 3005, 3006	3001D, 3003D, 3006D

Table A4. *Cont.*

Criteria	funModeling	inspectdf	IPDFileCheck	MOQA [2]
Consistency: Contradictions				
Accuracy: Unexpected distributions	4001D, 4003D, 4004D, 4005D, 4006D, 400XD	4001D, 4003D, 4004D, 4005D, 4006D, 400XD	4003D, 4005D, 4006D	4001D, 4003D, 4004D, 4005D, 4006D, 400XD
Accuracy: Unexpected associations	4007D, 4008D	4007D, 4008D		
Accuracy: Disagreement of rep. meas.				

Data quality indicator numbers followed by D mark functionalities which are not included as an indicator, but only as a descriptor. Extending the reference data quality framework, we included three additional codes: the code 100X denotes that the package provides functions to detect unexpected variable names, and 100Y denotes functions to detect data format mismatches. The code 400X indicates a functionality to detect loners (i.e., values which occur only once). [2] first published as `mosaicQA`.

Table A5. Data quality assessment capabilities by package (continued).

Criteria	mStats	pointblank	sanityTracker	skimr
Explicit reference to data quality		yes		
Based on a data quality framework				
Control via GUI				
Reproducibility via programming	yes	yes	yes	yes
Single function call for output	yes	yes		yes
Report generation (not only console)		yes		
Input of metadata through functions		yes	yes	
Input of metadata through separate file				
Grading of data quality issues		yes		
Dataset summary/overview	yes	yes		yes
Descriptive summary statistics	yes	yes		yes
Descriptive statistics graphs—univariate	yes	yes		yes
Descriptive statistics graphs—multivariate	yes			
Handles string properties		yes		yes
Integrity: Structural dataset error	1003	1001, 1002, 1003, 100X	1003	
Integrity: Dataset combination error		1004, 1005	1004, 1005	
Integrity: Value format error	1006D	1006, 1006D, 100YD, 1007, 1007D		1006D
Completeness: Crude missingness	2001	2001	2001	2001
Completeness: Qualified missingness				
Consistency: Range and value violations	3001D, 3003D, 3006D	3001, 3002, 3003, 3005, 3006, 3007	3001, 3003, 3006	3001D, 3002D, 3003D, 3006D, 3007D
Consistency: Contradictions		3008, 3009		
Accuracy: Unexpected distributions	4001D, 4002D, 4003D, 4004D, 4005D, 4006D	4001D, 4002D, 4003D, 4005D, 4006D		4003D, 4004D, 4005D, 4006D
Accuracy: Unexpected associations	4007, 4007D, 4008, 4008D, 4009D	4007D, 4008D, 4009D		
Accuracy: Disagreement of rep. meas.				

Data quality indicator numbers followed by D mark functionalities which are not included as an indicator, but only as a descriptor. Extending the reference data quality framework, we included three additional codes: the code 100X denotes that the package provides functions to detect unexpected variable names, and 100Y denotes functions to detect data format mismatches. The code 400X indicates a functionality to detect loners (i.e., values which occur only once).

Table A6. Data quality assessment capabilities by package (continued).

Criteria	SmartEDA	StatMeasures	summarytools	testdat
Explicit reference to data quality		yes		
Based on a data quality framework				
Control via GUI				
Reproducibility via programming	yes	yes	yes	yes
Single function call for output	yes	yes	yes	
Report generation (not only console)	yes		yes	yes
Input of metadata through functions	yes			yes
Input of metadata through separate file				
Grading of data quality issues				yes
Dataset summary/overview	yes	yes	yes	
Descriptive summary statistics	yes	yes	yes	
Descriptive statistics graphs—univariate	yes		yes	
Descriptive statistics graphs—multivariate	yes			
Handles string properties				yes
Integrity: Structural dataset error		1003	1003	1003
Integrity: Dataset combination error				
Integrity: Value format error	1006D	1006D	1006D	100Y, 1007, 1008
Completeness: Crude missingness	2001	2001	2001	2001
Completeness: Qualified missingness				2005
Consistency: Range and value violations	3001D, 3003D, 3006D	3001D, 3002D, 3003D, 3006D, 3007D	3001D, 3002D, 3003D, 3006D, 3007D	3001, 3002, 3003, 3005, 3006, 3007
Consistency: Contradictions				3008, 3009
Accuracy: Unexpected distributions	4001, 4001D, 4002D, 4003D, 4004D, 4005D, 4006D, 400XD	4001D, 4003D, 4005D, 4006, 4006D	4003D, 4004D, 4005D, 4006D, 400XD	4006D
Accuracy: Unexpected associations	4007, 4007D, 4008D, 4009D		4007, 4008	
Accuracy: Disagreement of rep. meas.				

Data quality indicator numbers followed by D mark functionalities which are not included as an indicator, but only as a descriptor. Extending the reference data quality framework, we included three additional codes: the code 100X denotes that the package provides functions to detect unexpected variable names, and 100Y denotes functions to detect data format mismatches. The code 400X indicates a functionality to detect loners (i.e., values which occur only once).

Table A7. Data quality assessment capabilities by package (continued).

Criteria	validate [3]	visdat	xray
Explicit reference to data quality	yes		
Based on a data quality framework			
Control via GUI			
Reproducibility via programming	yes	yes	yes
Single function call for output			yes
Report generation (not only console)			
Input of metadata through functions	yes	yes	
Input of metadata through separate file	yes		
Grading of data quality issues			yes
Dataset summary/overview		yes	yes
Descriptive summary statistics			yes
Descriptive statistics graphs—univariate			yes
Descriptive statistics graphs—multivariate		yes	
Handles string properties	yes	yes	
Integrity: Structural dataset error	1002, 1003	1001, 1002	

Table A7. *Cont.*

Criteria	validate [3]	visdat	xray
Integrity: Dataset combination error	1004		
Integrity: Value format error	100Y, 1007	1006, 1006D, 1007	1006D
Completeness: Crude missingness	2001	2001	2001
Completeness: Qualified missingness			
Consistency: Range and value violations	3001, 3002, 3003, 3004, 3005, 3006, 3007		3001D, 3002D, 3003D, 3006D, 3007D
Consistency: Contradictions	3008, 3009		
Accuracy: Unexpected distributions	4001D		4003D, 4004D, 4005D, 4006D, 400XD
Accuracy: Unexpected associations		4007D, 4008D	
Accuracy: Disagreement of rep. meas.			

Data quality indicator numbers followed by D mark functionalities which are not included as an indicator, but only as a descriptor. Extending the reference data quality framework, we included three additional codes: the code 100X denotes that the package provides functions to detect unexpected variable names, and 100Y denotes functions to detect data format mismatches. The code 400X indicates a functionality to detect loners (i.e., values which occur only once). [3] includes integration with `errorlocate`.

References

1. Kahn, M.G.; Brown, J.S.; Chun, A.T.; Davidson, B.N.; Meeker, D.; Ryan, P.B.; Schilling, L.M.; Weiskopf, N.G.; Williams, A.E.; Zozus, M.N. Transparent reporting of data quality in distributed data networks. *EGEMS* **2015**, *3*, 1052. [CrossRef] [PubMed]
2. Kahn, M.G.; Callahan, T.J.; Barnard, J.; Bauck, A.E.; Brown, J.; Davidson, B.N.; Estiri, H.; Goerg, C.; Holve, E.; Johnson, S.G.; et al. A harmonized data quality assessment terminology and framework for the secondary use of electronic health record data. *EGEMS* **2016**, *4*, 1244. [CrossRef] [PubMed]
3. Lee, K.; Weiskopf, N.; Pathak, J. A Framework for Data Quality Assessment in Clinical Research Datasets. *AMIA Annu. Symp. Proc.* **2017**, *2017*, 1080–1089. [PubMed]
4. Liaw, S.T.; Guo, J.G.N.; Ansari, S.; Jonnagaddala, J.; Godinho, M.A.; Borelli, A.J.; de Lusignan, S.; Capurro, D.; Liyanage, H.; Bhattal, N.; et al. Quality assessment of real-world data repositories across the data life cycle: A literature review. *J. Am. Med. Inform. Assoc.* **2021**, *28*, 1591–1599. [CrossRef]
5. Weiskopf, N.G.; Bakken, S.; Hripcsak, G.; Weng, C. A data quality assessment guideline for electronic health record data reuse. *EGEMS* **2017**, *5*, 14. [CrossRef] [PubMed]
6. Schmidt, C.O.; Struckmann, S.; Enzenbach, C.; Reineke, A.; Stausberg, J.; Damerow, S.; Huebner, M.; Schmidt, B.; Sauerbrei, W.; Richter, A. Facilitating harmonized data quality assessments. A data quality framework for observational health research data collections with software implementations in R. *BMC Med. Res. Methodol.* **2021**, *21*, 63. [CrossRef]
7. Nonnemacher, M.; Nasseh, D.; Stausberg, J. *Datenqualität in der medizinischen Forschung: Leitlinie zum adaptiven Management von Datenqualität in Kohortenstudien und Registern*; MWV Medizinisch Wissenschaftliche Verlagsgesellschaft: Berlin, Germany, 2014.
8. Kandel, S.; Parikh, R.; Paepcke, A.; Hellerstein, J.M.; Heer, J. Profiler: Integrated statistical analysis and visualization for data quality assessment. In Proceedings of the International Working Conference on Advanced Visual Interfaces, Capri Island, Italy, 21–25 May 2012; pp. 547–554.
9. Golling, T.; Hayward, H.; Onyisi, P.; Stelzer, H.; Waller, P. The ATLAS data quality defect database system. *Eur. Phys. J. C* **2012**, *72*, 1–6. [CrossRef]
10. Fillbrunn, A.; Dietz, C.; Pfeuffer, J.; Rahn, R.; Landrum, G.A.; Berthold, M.R. KNIME for reproducible cross-domain analysis of life science data. *J. Biotechnol.* **2017**, *261*, 149–156. [CrossRef]
11. Tute, E.; Scheffner, I.; Marschollek, M. A method for interoperable knowledge-based data quality assessment. *BMC Med. Informatics Decis. Mak.* **2021**, *21*, 93. [CrossRef]
12. De Jonge, E.; Van Der Loo, M. *An Introduction to Data Cleaning with R*; Statistics Netherlands: Heerlen, The Netherlands, 2013.
13. Eaton, J.; Painter, I.; Olson, D.; Lober, W.B. Visualizing the quality of partially accruing data for use in decision making. *Online J. Public Health Inform.* **2015**, *7*, e226. [CrossRef]
14. Hripcsak, G.; Duke, J.D.; Shah, N.H.; Reich, C.G.; Huser, V.; Schuemie, M.J.; Suchard, M.A.; Park, R.W.; Wong, I.C.K.; Rijnbeek, P.R.; et al. Observational Health Data Sciences and Informatics (OHDSI): Opportunities for observational researchers. *Stud. Health Technol. Inform.* **2015**, *216*, 574. [PubMed]
15. Bialke, M.; Rau, H.; Schwaneberg, T.; Walk, R.; Bahls, T.; Hoffmann, W. mosaicQA-A General Approach to Facilitate Basic Data Quality Assurance for Epidemiological Research. *Methods Inf. Med.* **2017**, *56*, e67–e73. [CrossRef] [PubMed]
16. Petersen, A.H.; Ekstrøm, C.T. dataMaid: Your Assistant for Documenting Supervised Data Quality Screening in R. *J. Stat. Softw.* **2019**, *90*, 1–38. [CrossRef]

17. Kapsner, L.A.; Kampf, M.O.; Seuchter, S.A.; Kamdje-Wabo, G.; Gradinger, T.; Ganslandt, T.; Mate, S.; Gruendner, J.; Kraska, D.; Prokosch, H.U. Moving towards an EHR data quality framework: The MIRACUM approach. In *German Medical Data Sciences: Shaping Change–Creative Solutions for Innovative Medicine*; IOS Press: Amsterdam, The Netherlands, 2019; pp. 247–253.
18. van der Loo, M.P.J.; de Jonge, E. Data Validation Infrastructure for R. *J. Stat. Softw.* **2021**, *97*, 1–31. 097.i10. [CrossRef]
19. Huebner, M.; le Cessie, S.; Schmidt, C.O.; Vach, W. A contemporary conceptual framework for initial data analysis. *Obs. Stud.* **2018**, *4*, 171–192. [CrossRef]
20. Staniak, M.; Biecek, P. The Landscape of R Packages for Automated Exploratory Data Analysis. *R J.* **2019**, *11*, 347. [CrossRef]
21. R Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing, Vienna, Austria, 2020.
22. Hornik, K. R FAQ. 2021. Available online: https://cran.r-project.org/doc/FAQ/R-FAQ.html (accessed on 8 March 2022).
23. *Standard ISO 8000-2:2017*; Data Quality—Part 2: Vocabulary. International Organization for Standardization: Geneva, Switzerland, 2017.
24. Richter, A.; Schössow, J.; Werner, A.; Schauer, B.; Radke, D.; Henke, J.; Struckmann, S.; Schmidt, C.O. Data quality monitoring in clinical and observational epidemiologic studies: The role of metadata and process information. *MIBE* **2019**, *15*. [CrossRef]
25. Tricco, A.C.; Lillie, E.; Zarin, W.; O'Brien, K.K.; Colquhoun, H.; Levac, D.; Moher, D.; Peters, M.D.; Horsley, T.; Weeks, L.; et al. PRISMA Extension for Scoping Reviews (PRISMA-ScR): Checklist and Explanation. *Ann. Intern. Med.* **2018**, *169*, 467–473. [CrossRef]
26. Putatunda, S.; Ubrangala, D.; Rama, K.; Kondapalli, R. SmartEDA: An R Package for Automated Exploratory Data Analysis. *J. Open Source Softw.* **2019**, *4*, 1509. [CrossRef]
27. Csárdi, G.; Salmon, M. pkgsearch: Search and Query CRAN R Packages; R Package Version 3.0.3. 2020. Available online: https://CRAN.R-project.org/package=pkgsearch (accessed on 18 January 2022).
28. Wickham, H.; François, R.; Henry, L.; Müller, K. dplyr: A Grammar of Data Manipulation; R Package Version 1.0.7. 2021. Available online: https://CRAN.R-project.org/package=dplyr (accessed on 18 January 2022).
29. Schmidt, C.O.; Richter, A.; Struckmann, S. Data Quality Concept. Available online: https://dataquality.ship-med.uni-greifswald.de/DQconceptNew.html (accessed on 9 March 2022).
30. Völzke, H.; Alte, D.; Schmidt, C.O.; Radke, D.; Lorbeer, R.; Friedrich, N.; Aumann, N.; Lau, K.; Piontek, M.; Born, G.; et al. Cohort Profile: The Study of Health in Pomerania. *Int. J. Epidemiol.* **2011**, *40*, 294–307. [CrossRef]
31. Völzke, H.; Schössow, J.; Schmidt, C.O.; Jürgens, C.; Richter, A.; Werner, A.; Werner, N.; Radke, D.; Teumer, A.; Ittermann, T.; et al. Cohort Profile Update: The Study of Health in Pomerania (SHIP). *Int. J. Epidemiol.* **2022**, dyac034. [CrossRef] [PubMed]
32. Standards and Tools for Data Quality Assessment in Epidemiological Studies. Available online: https://dataquality.ship-med.uni-greifswald.de/ (accessed on 1 October 2021).
33. Hebbali, A. xplorerr: Tools for Interactive Data Exploration; R Package Version 0.1.2. 2021. Available online: https://CRAN.R-project.org/package=xplorerr (accessed on 7 March 2022).
34. Priyam, A. Analyzer: Data Analysis and Automated R Notebook Generation; R Package Version 1.0.1. 2020. Available online: https://CRAN.R-project.org/package=analyzer (accessed on 7 March 2022).
35. Nanji, H.; Chernbumroong, S. mdapack: Medical Data Analysis Pack; R Package Version 0.0.2. 2020. Available online: https://CRAN.R-project.org/package=mdapack (accessed on 7 March 2022).
36. de Jonge, E.; van der Loo, M. editrules: Parsing, Applying, and Manipulating Data Cleaning Rules. R Package Version 2.9.3. 2018. Available online: https://CRAN.R-project.org/package=editrules (accessed on 7 March 2022).
37. Nguyen, G. assertable: Verbose Assertions for Tabular Data (Data.frames and Data.tables); R Package Version 0.2.8. 2021. Available online: https://CRAN.R-project.org/package=assertable (accessed on 7 March 2022).
38. Cotton, R. assertive: Readable Check Functions to Ensure Code Integrity; R Package Version 0.3-6. 2020. Available online: https://CRAN.R-project.org/package=assertive (accessed on 7 March 2022).
39. Fischetti, T. Assertr: Assertive Programming for R Analysis Pipelines; R Package Version 2.8. 2021. Available online: https://CRAN.R-project.org/package=assertr (accessed on 7 March 2022).
40. Marin, D.H. clickR: Semi-Automatic Preprocessing of Messy Data with Change Tracking for Dataset Cleaning; R Package Version 0.8.0. 2021. Available online: https://CRAN.R-project.org/package=clickR (accessed on 7 March 2022).
41. Cui, B. DataExplorer: Automate Data Exploration and Treatment; R Package Version 0.8.2. 2020. Available online: https://CRAN.R-project.org/package=DataExplorer (accessed on 7 March 2022).
42. Richter, A.; Schmidt, C.O.; Struckmann, S. dataquieR: Data Quality in Epidemiological Research; R Package Version 1.0.9. 2021. Available online: https://CRAN.R-project.org/package=dataquieR (accessed on 7 March 2022).
43. Signorell, A.; Aho, K.; Alfons, A.; Anderegg, N.; Aragon, T.; Arachchige, C.; Arppe, A.; Baddeley, A.; Barton, K.; Bolker, B.; et al. DescTools: Tools for Descriptive Statistics; R Package Version 0.99.44. 2021. Available online: https://CRAN.R-project.org/package=DescTools (accessed on 7 March 2022).
44. Ryu, C. dlookr: Tools for Data Diagnosis, Exploration, Transformation; R Package Version 0.5.4. 2021. Available online: https://CRAN.R-project.org/package=dlookr (accessed on 7 March 2022).

45. Kapsner, L.A.; Mang, J.M.; Mate, S.; Seuchter, S.A.; Vengadeswaran, A.; Bathelt, F.; Deppenwiese, N.; Kadioglu, D.; Kraska, D.; Prokosch, H.U. Linking a Consortium-Wide Data Quality Assessment Tool with the MIRACUM Metadata Repository. *Appl. Clin. Inf.* **2021**, *12*, 826–835. doi: 10.1055/s-0041-1733847. [CrossRef] [PubMed]
46. de Jonge, E.; van der Loo, M. errorlocate: Locate Errors with Validation Rules; R Package Version 0.9.9. 2021. Available online: https://CRAN.R-project.org/package=errorlocate (accessed on 7 March 2022).
47. Gassen, J. ExPanDaR: Explore Your Data Interactively; R Package Version 0.5.3. 2020. Available online: https://CRAN.R-project.org/package=ExPanDaR (accessed on 7 March 2022).
48. Krasser, R. explore: Simplifies Exploratory Data Analysis; R Package Version 0.8.0. 2022. Available online: https://CRAN.R-project.org/package=explore (accessed on 7 March 2022).
49. Casas, P. funModeling: Exploratory Data Analysis and Data Preparation Tool-Box; R Package Version 1.9.4. 2020. Available online: https://CRAN.R-project.org/package=funModeling (accessed on 7 March 2022).
50. Rushworth, A. inspectdf: Inspection, Comparison and Visualisation of Data Frames; R Package Version 0.0.11. 2021. Available online: https://CRAN.R-project.org/package=inspectdf (accessed on 7 March 2022).
51. Krishnan, S.M. IPDFileCheck: Basic Functions to Check Readability, Consistency, and Content of an Individual Participant Data File. R Package Version 0.7.5. 2022. Available online: https://CRAN.R-project.org/package=IPDFileCheck (accessed on 7 March 2022).
52. Bialke, M.; Schwaneberg, T.; Walk, R. MOQA: Basic Quality Data Assurance for Epidemiological Research; R Package Version 2.0.0. 2017. Available online: https://CRAN.R-project.org/package=MOQA (accessed on 7 March 2022).
53. Oo, M.M. mStats: Epidemiological Data Analysis; R Package Version 3.4.0. 2020. Available online: https://CRAN.R-project.org/package=mStats (accessed on 7 March 2022).
54. Iannone, R.; Vargas, M. pointblank: Data Validation and Organization of Metadata for Local and Remote Tables; R Package Version 0.10.0. 2022. Available online: https://CRAN.R-project.org/package=pointblank (accessed on 7 March 2022).
55. Scheer, M. sanityTracker: Keeps Track of all Performed Sanity Checks; R Package Version 0.1.0. 2020. Available online: https://CRAN.R-project.org/package=sanityTracker (accessed on 7 March 2022).
56. Waring, E.; Quinn, M.; McNamara, A.; Arino de la Rubia, E.; Zhu, H.; Ellis, S. skimr: Compact and Flexible Summaries of Data; R Package Version 2.1.3. 2021. Available online: https://CRAN.R-project.org/package=skimr (accessed on 7 March 2022).
57. Dayanand Ubrangala.; R, K.; Prasad Kondapalli, R.; Putatunda, S. SmartEDA: Summarize and Explore the Data; R Package Version 0.3.8. 2021. Available online: https://CRAN.R-project.org/package=SmartEDA (accessed on 7 March 2022).
58. Jain, A. StatMeasures: Easy Data Manipulation, Data Quality and Statistical Checks; R Package Version 1.0. 2015. Available online: https://CRAN.R-project.org/package=StatMeasures (accessed on 7 March 2022).
59. Comtois, D. summarytools: Tools to Quickly and Neatly Summarize Data; R Package Version 1.0.0. 2021. Available online: https://CRAN.R-project.org/package=summarytools (accessed on 7 March 2022).
60. Smith, D.; Behr, K. testdat: Data Unit Testing for R; R Package Version 0.4.0. 2022. Available online: https://CRAN.R-project.org/package=testdat (accessed on 7 March 2022).
61. Tierney, N. visdat: Visualising Whole Data Frames. *JOSS* **2017**, *2*, 355. [CrossRef]
62. Seibelt, P. xray: X Ray Vision on Your Datasets; R Package Version 0.2. 2017. Available online: https://CRAN.R-project.org/package=xray (accessed on 7 March 2022).
63. Csárdi, G. cranlogs: Download Logs from the 'RStudio' 'CRAN' Mirror; R Package Version 2.1.1. 2019. Available online: https://CRAN.R-project.org/package=cranlogs (accessed on 5 April 2022).
64. Hamill, P. *Unit Test Frameworks: Tools for High-Quality Software Development*; O'Reilly Media: Newton, MA, USA, 2004.
65. Wickham, H. testthat: Get Started with Testing. *R J.* **2011**, *3*, 5–10. [CrossRef]
66. van der Loo, M.P.J. Monitoring Data in R with the lumberjack Package. *J. Stat. Softw.* **2021**, *98*, 1–13. 098.i01. [CrossRef]
67. Kapsner, L.A.; Mang, J.M. DQAgui: Graphical User Interface for Data Quality Assessment; R Package Version 0.1.9. 2022. Available online: https://CRAN.R-project.org/package=DQAgui (accessed on 7 March 2022).
68. Rinaldi, E.; Thun, S. From OpenEHR to FHIR and OMOP Data Model for Microbiology Findings. *Stud. Health Technol. Inf.* **2021**, *281*, 402–406. [CrossRef]
69. Cheng, A.C.; Duda, S.N.; Taylor, R.; Delacqua, F.; Lewis, A.A.; Bosler, T.; Johnson, K.B.; Harris, P.A. REDCap on FHIR: Clinical Data Interoperability Services. *J. Biomed. Inf.* **2021**, *121*, 103871. [CrossRef]
70. Hoevenaar-Blom, M.P.; Guillemont, J.; Ngandu, T.; Beishuizen, C.R.L.; Coley, N.; Moll van Charante, E.P.; Andrieu, S.; Kivipelto, M.; Soininen, H.; Brayne, C.; et al. Improving data sharing in research with context-free encoded missing data. *PLoS ONE* **2017**, *12*, e0182362. [CrossRef]
71. Dinh, D.T.; Huynh, V.N.; Sriboonchitta, S. Clustering mixed numerical and categorical data with missing values. *Inf. Sci.* **2021**, *571*, 418–442. [CrossRef]
72. Gao, K.; Khan, H.A.; Qu, W. Clustering with Missing Features: A Density-Based Approach. *Symmetry* **2022**, *14*, 60. [CrossRef]
73. Holve, E.; Segal, C.; Lopez, M.H.; Rein, A.; Johnson, B.H. The Electronic Data Methods (EDM) Forum for Comparative Effectiveness Research (CER). *Med. Care* **2012**, *50*, S7–S10. [CrossRef] [PubMed]
74. McMurry, A.J.; Murphy, S.N.; MacFadden, D.; Weber, G.; Simons, W.W.; Orechia, J.; Bickel, J.; Wattanasin, N.; Gilbert, C.; Trevvett, P.; et al. SHRINE: Enabling Nationally Scalable Multi-Site Disease Studies. *PLoS ONE* **2013**, *8*, e55811. [CrossRef] [PubMed]

75. van Ommen, G.J.B.; Törnwall, O.; Bréchot, C.; Dagher, G.; Galli, J.; Hveem, K.; Landegren, U.; Luchinat, C.; Metspalu, A.; Nilsson, C.; et al. BBMRI-ERIC as a Resource for Pharmaceutical and Life Science Industries: The Development of Biobank-Based Expert Centres. *Eur. J. Hum. Genet.* **2015**, *23*, 893–900. [CrossRef]
76. Semler, S.; Wissing, F.; Heyder, R. German Medical Informatics Initiative: A National Approach to Integrating Health Data from Patient Care and Medical Research. *Methods Inf. Med.* **2018**, *57*, e50–e56. [CrossRef] [PubMed]
77. Bahls, T.; Pung, J.; Heinemann, S.; Hauswaldt, J.; Demmer, I.; Blumentritt, A.; Rau, H.; Drepper, J.; Wieder, P.; Groh, R.; Hummers, E.; et al. Designing and Piloting a Generic Research Architecture and Workflows to Unlock German Primary Care Data for Secondary Use. *J. Transl. Med.* **2020**, *18*, 394. [CrossRef] [PubMed]
78. Hersh, W.R.; Weiner, M.G.; Embi, P.J.; Logan, J.R.; Payne, P.R.; Bernstam, E.V.; Lehmann, H.P.; Hripcsak, G.; Hartzog, T.H.; Cimino, J.J.; et al. Caveats for the Use of Operational Electronic Health Record Data in Comparative Effectiveness Research. *Med. Care* **2013**, *51*, S30–S37. [CrossRef]
79. DeFalco, F.; Ryan, P.; Schuemie, M.; Huser, V.; Knoll, C.; Londhe, A.; Abdul-Basser, T.; Molinaro, A. Achilles: Generates Descriptive Statistics for an OMOP CDM Instance; R Package Version 1.7. 2021. Available online: https://github.com/OHDSI/Achilles (accessed on 7 March 2022).
80. Blacketer, C.; Schuemie, F.J.; Ryan, P.B.; Rijnbeek, P. Increasing trust in real-world evidence through evaluation of observational data quality. *J. Am. Med. Inform. Assoc.* **2021**, *28*, 2251–2257. [CrossRef]
81. OMOP Common Data Model. Available online: http://ohdsi.github.io/CommonDataModel/ (accessed on 5 April 2022).
82. Ooms, J. METACRAN. Available online: https://www.r-pkg.org/ (accessed on 9 March 2022).
83. Woo, K.; Kauer, N.; Montgomery, K. dccvalidator: Metadata Validation for Data Coordinating Centers; R Package Version 0.3.0. 2020. Available online: https://CRAN.R-project.org/package=dccvalidator (accessed on 28 February 2022).