World Wide Web (2021) 24:2099–2137
https://doi.org/10.1007/s11280-021-00950-y

Enhancing decision-making in user-centered web development: a methodology for card-sorting analysis

José A. Macías[1] · Alma L. Culén[2]

Received: 10 May 2021 / Revised: 4 August 2021 / Accepted: 10 September 2021 /
Published online: 7 October 2021

Abstract

The World Wide Web has become a common platform for interactive software development. Most web applications feature custom user interfaces used by millions of people every day. Information architecture addresses the structural design of information to build quality web applications with improved usability of content, navigation, and findability. One of the most frequently utilized information architecture methods is card sorting—an affordable, user-centered approach for eliciting and evaluating categories and navigable items. Card sorting facilitates decision-making during the development process based on users' mental models of a given application domain. However, although the qualitative analysis of card sorts has become common practice in information architecture, the quantitative analysis of card sorting is less widely applied. The reason for this gap is that quantitative analysis often requires the use of customized techniques to extract meaningful information for decision-making. To facilitate this process and support the structuring of information, we propose a methodology for the quantitative analysis of card-sorting results in this paper. The suggested approach can be systematically applied to provide clues and support for decisions. These might significantly impact the design and, thus, the final quality of the web application. Therefore, the approach includes proper goodness values that enable comparisons among the results of the methods and techniques used and ensure the suitability of the analyses performed. Two publicly available datasets were used to demonstrate the key issues related to the interpretation of card sorting results and the overall suitability and validity of the proposed methodology.

Keywords User-centered web application development · Information architecture · Quantitative analysis · Card sorting · User experience · Human–computer interaction

✉ José A. Macías
j.macias@uam.es

Alma L. Culén
almira@ifi.uio.no

[1] Computer Engineering Department, Escuela Politécnica Superior, Universidad Autónoma de Madrid, Tomás y Valiente 11, 28049 Madrid, Spain

[2] Department of Informatics, University of Oslo, P. Box 1080, 0316 Oslo, Norway

1 Introduction

The World Wide Web has become the main platform on which interactive applications are developed [1–4]. Initially, most web applications were produced as content blocks without consideration for specific quality issues. However, as the field of human–computer interaction (HCI) has moved toward user experience [5, 6], interactive web application design has changed to include a user-centered and experiential perspective [7]. Content, structure, and navigation are all considered crucial to a website's success [8]. Therefore, a positive user experience with web browsing is essential since efficient navigation and ease of access to good content increases perceived satisfaction and helps users find information that is of interest to them [9]. In contrast, deficiencies in content or navigation design negatively affect the user experience and may also affect the intelligent extraction algorithms that automatically produce knowledge [10] from content and page structure [9, 11].

Information architecture [9] is a discipline that focuses on the structural design of both content and navigation in web application development [6], aiming to provide positive user experiences with web applications. The latter implies a user-centered approach in which different methods, tools, and techniques are utilized to address usability issues [11]. In this environment, card sorting is a common and easy-to-use method [12]. It facilitates design decisions that align with users' mental models of an application area [13], thus increasing the possibility of creating positive experiences with the web application. In information architecture, card sorting is used to build and evaluate the navigation structure of a website [14] and to elicit the most convenient content categories, concepts, and information labeling [15]. The approach is particularly advantageous in the early design phases of a web development project [16]. However, card sorting is also utilized in other phases, for example, for summative evaluation and for evaluating and improving the quality of existing web applications [11].

Once the sorting of cards is complete, analysis is performed to gain insights relevant to the information structuring and navigation. While qualitative analysis of card sorting has become commonplace in information architecture, quantitative analysis requires further attention. The reason for this is that quantitative analysis often requires additional means of analysis, for example, clustering and scaling, which are very dependent on the way the data is understood [16]. Usually, quantitative analysis of a card sorts is more complex than qualitative, and care in applying it is essential for obtaining meaningful results that lead to good design. Therefore, in practice, most card sorts are still analyzed using custom spreadsheets to obtain only basic information about raw data [13]. This leaves room for increasing the understanding of the key issues related to interpreting card sorting results and creating a systematic approach, a methodology that supports quantitative analysis of card sorting results.

Even though many commercial tools have been made to facilitate the quantitative analysis of card sorting, the decision-making process still needs improvement. The existing tools often include visualizations that show the main outcomes and facilitate reasoning, enabling evaluators and participants to successfully carry out card-sorting tasks in different application domains, providing mechanisms that simplify the process. However, most such tools produce only basic quantitative results, addressing mainly hierarchical clustering and co-occurrence, which greatly predetermines the parameters and statistical techniques to be used and restricts the outcomes and the goodness observed. In addition, without adequate knowledge and correct initial analysis, a wrong selection of statistical techniques might be made when using commercial tools, for example, when proceeding without evaluating the validity of different conditions [17].

The above arguments allow the framing of the main research question discussed in this paper: *Is it possible to define a systematic method for carrying out a quantitative analysis of card-sorting data that provides instruments and goodness indicators for decision-making in web application development?*

A methodology for quantitative analysis of data obtained from card sorting is proposed to answer the above question. It is based on a top-down approach and consists of guidelines regarding the use of statistical techniques, visualizations, and goodness indicators to guide evaluators through the analysis process. Several existing statistics and algorithms from other disciplines have been selected, customized for card sorting, and integrated into the proposed approach. The methodology utilizes a range of methods and techniques, some that are essential and others that are complementary and can be used on-demand to provide better information for decision-making.

The proposed methodology contributes to information architecture by systematizing quantitative card-sorting into a comprehensive approach. Furthermore, the methodology aims to make quantitative card sorting available to a broader range of evaluators in other fields like design thinking, HCI, or service design, who currently use qualitative analysis because the techniques in quantitative analyses are too complex. Specifically, the methodology contributes by providing:

- Summary tables with main indicators and statistics for characterizing card-sorting variables based on the available raw data.
- Comprehensive modeling of the main data structures to be used as input for different techniques and algorithms, and the corresponding transformations needed to get the appropriate data structures.
- Summary tables with recommendations on algorithms and metrics to apply, including optimal configurations and parameters for a specific card-sorting design, depending on the variables involved.
- Visualizations to study and compare data density, categories that received more attention from participants in terms of sorts, the correlation among categories and cards, similarities using a graph-based representation, different multidimensional scaling configurations, and clustering solutions.
- Specific charts to evaluate goodness indicators, such as *Shepard*, *Stress-Per-Point*, and *Average Silhouette* diagrams. Also, appropriate values for goodness evaluation such as *Stress*, *Silhouette*, and *Cophenetic Correlation* coefficients are provided.
- Summary lists provided in each step to guide about the methods and techniques proposed, indicating which ones are essential, recommended, or optional.

The paper is structured as follows: Sect. 2 reports on related work, state of the art, and further information about card sorting and specific approaches; Sect. 3 describes the proposed methodology, including guidelines, statistics, and visualizations to apply in each case; Sect. 4 presents a discussion, including limitations and threats to validity related to the proposed methods, allowing us to answer the main research question in the affirmative. Finally, Sect. 5 provides conclusions and suggests possible future extensions of the work.

2 State of the art

The early practice of card sorting for research purposes can be traced to the field of social science, and especially, psychology [18, 19] where researchers frequently utilized the method as a psychological tool [20] in experiments involving the study of mental capacity and reaction time [21] or when making comparisons, for example, comparing the developmental states of individuals with cognitive challenges [20]. Card sorting has also been applied in linguistics to understand semantics [22], in marketing to understand the consumer's perspective [23] and to achieve pairwise similarity judgments [24], and also in criminology and other areas of the social sciences [25, 26]. In short, card sorting has been used in a plethora of research settings where participants are invited to perform tasks involving cards in which they must group, name, or categorize the objects represented by the cards. The use of card sorting has seen a steady increase, especially since the emergence of the World Wide Web [27].

Currently, card-sorting practices are mainly found in the fields of software engineering and HCI, and more specifically, in the fields of information architecture and user experience [9, 13, 28]. Card sorting gained prominence in the organization and evaluation of content and navigation in web design and in a range of other application areas where the understanding and experience of users within a situated context are central. As mentioned, card sorting can be useful in eliciting users' mental models or even for evaluating already existing information compositions according to the users' criteria [12, 13], especially in the early phases of a web development project.

However, there are many examples of a quite different usage. For instance, interaction design methods might be used to evaluate the usability of mobile applications concerning the experiences of blind people with those applications [29]. Moreover, within newer fields, such as design thinking [30, 31] and service design [32, 33], which are based on user- and human-centered principles, card sets and card sorting are often used to support design processes. The card sets might offer support by describing alternative methods to use in a process, providing images related to possible user experiences within a particular context, or suggesting choices of technologies [34]. Their intent is to stimulate creativity and innovative thinking, or else to focus on users' experiences and behavior in relation to products or services [35, 36]. A recent paper [37] used card sorting to categorize the first impressions of design card sets in relation to different formal qualities and content of sets. In addition, card sorting is often used to provide important clues to user experience researchers concerning brand alignments, emotions, goals, and workflows [38]. As discussed later in this paper, such categorizations might help to further the use of quantitative card sorting and analysis in areas where it is still underused as a methodology, for example, in design thinking, interaction design, or service design.

Despite differences in application domains and disciplines where card sorting is used, the practice of card-sorting is normally carried out similarly, using card-sorting studies. A card-sorting study comprises a set of sorting tasks proposed and evaluated by an expert (referred to as the evaluator), who also recruits participants for the study. Card-sorting tasks are then performed by those participants, face-to-face or online, where different stimuli (the cards) must be classified, based on the participants' subjective criteria, into different categories. The perceived relationships among the cards and their likelihood of being placed in the same category play an essential role in the participants' decisions when categorizing. Once the card-sorting tasks are finished, the evaluator carries out an analysis using the information obtained. At different stages of the study, and depending on the kind of study, the evaluator might wish to gather further information by using additional methods, for instance, by interviews or additional card-sorting tasks. The study's outcomes are then used to make decisions concerning, for example, structuring a website's content and navigation or city planning [39].

A card-sorting study (and its most important step—the analysis of card-sorts) typically utilizes one of the two different but complementary perspectives, the qualitative or the quantitative [13, 26]. Qualitative studies are frequently used to obtain information from participants' behavior in face-to-face card sorts. In contrast, quantitative studies analyze card-sorting data to obtain numerical evidence through different statistical techniques. Since the approaches are complementary and provide different sets of evidence, they can be combined to reinforce the conclusions of a study. In general, quantitative analysis requires a larger number of card sorts, and it fits well with online and tool-based card sorting, whereas qualitative analysis is useful for extracting on-site participants' opinions and feelings and for observing first-hand how participants perform their card sorts. Although quantitative studies have many advantages, such as the ease of performing online card-sorting at participants' convenience, the increased number of cards and tasks that could be used, and the existence of tools to facilitate the analysis, the main barrier to the broader use of the quantitative approach still lies in the difficulty of correctly implementing a quantitative analysis [16], as discussed in the Introduction.

In addition to choosing either a qualitative, quantitative, or mixed-methods study, other choices need to be made concerning card-sorting tasks [13]. Careful decisions concerning the sorting tasks are essential since they predetermine the selection of analytic tools that might lead to meaningful results. The main choices regarding the tasks are whether they are open or closed [13] and whether they are single or multiple card sorts [40], which might require pre-processing for normalization [12]. Furthermore, there are considerations regarding the use of single or nested categories that are important for the analysis. They might require pre-processing to identify dependencies between categories and subcategories [41].

Several commercial software tools aim to support non-skilled evaluators in performing card-sorting analyses. However, as discussed, they can over-simplify and create errors through the choice of techniques. Some of the more advanced tools, such as Cardsorting.net [42], provide graphical interactions for card-sorting tasks and allow for the export of information in different formats for further processing involving different tools and statistical packages. Finally, some commercial tools aim to facilitate more complex quantitative analyses, for example, Syntagm [43], XSort [44], UserZoom [45], Proven by Users [46], UsabiliTest [47], and OptimalSort [48]. These come with varying price tags and free reduced versions. These existing tools are highly functional approaches, with elaborate user interfaces that support the whole card-sorting process and facilitate the work of both evaluators and participants. However, most such tools provide specific data representations and run standard algorithms with customized parameters and settings, which intrinsically limit the utilization of alternative techniques and the gathering of enriched statistical outcomes, thus reducing the expressivity of the quantitative analysis.

Compared with the proposed methodology, described in the next section, most existing commercially available tools provide only basic analyses such as general statistics by sort and participant, dendrograms, frequency, and card-based classification matrix analyses. Advanced statistical and data mining techniques, such as the ones used in our comprehensive methodology, are rare and hard to find in the commercial tools mentioned above.

- Most of the existing tools are based on a card analysis. However, it is also important to conduct category analysis in normalization—i.e., detecting redundancy to simplify the set of categories. It is also interesting to analyze relationships among existing categories and see which received more attention from participants.
- Very few commercially available tools utilize more advanced analytical methods, like custom multi-dimensional scaling (MDS). Most existing tools use dendrograms as a clustering technique. However, dendrograms are based on an agglomerative clustering representation that needs to be analyzed with care, and the analysis depends on one important parameter called *height*. Some existing tools provide customized versions of dendrograms, which must be interpreted by the user (with the *height* parameter to consider the number of clusters), leading potentially to incorrect decisions. Also, when the number of items is large, dendrograms become cumbersome to work with, making other options such as k-means and principal component analysis (PCA) more convenient to use. While none of the existing tools offer such advanced alternatives, our methodology suggests alternatives, sometimes multiple.
- Heatmaps and graph-based visualizations are not found in any of the existing tools, even though they provide the big picture for the classification and the immediate visual feedback on relationships between cards and categories.
- None of the existing tools provide correlation analysis to consider linear relationships among variables. These can be useful for detecting items that may be related, not related, or even inversely related, enriching the relationship analysis.
- None of the existing tools provide advanced goodness indicators to evaluate the suitability of the techniques proposed.

In conclusion, the quality of the quantitative card-sorting analysis depends strongly on the number of card sorts [49, 50] and the nature of the data analyzed, but also on the methods, parameters, and algorithms selected in the context of a specific data type. A wrong selection and parametrization might produce inaccurate or misleading results [51], guiding the evaluator to make incorrect decisions [17]. While commercial tools to support card-sorting processes exist, they lack a comprehensive approach to the quantitative analysis of card sorting that might enhance decision-making significantly. A comprehensive approach suggested in this paper facilitates both the initial analysis of the card-sorting data (enabling selection of techniques and parameters leading to optimal results) and the determination and inclusion of principal goodness indicators for the benefit of decision-makers when interpreting the card-sorting results. Thus, a comprehensive approach facilitates these processes to a more significant extent than is possible with any existing tool.

3 Proposed methodology

A comprehensive methodology for the quantitative analysis of card-sorting data is proposed, aiming to mitigate the challenges mentioned in the previous section. The solution can be considered as a top-down approach. The analysis unfolds in four phases: (1) the initial analysis of raw data obtained from the card sorts, (2) data preparation for statistical analysis, (3) dissimilarity analysis proposing different metrics depending on the card-sorting data, and (4) multivariate analysis and calculation of the optimal number of clusters to consider. Figure 1 illustrates the steps of the proposed approach and main activities involved—the rectangles at the right indicate the main outputs (statistics and graphical

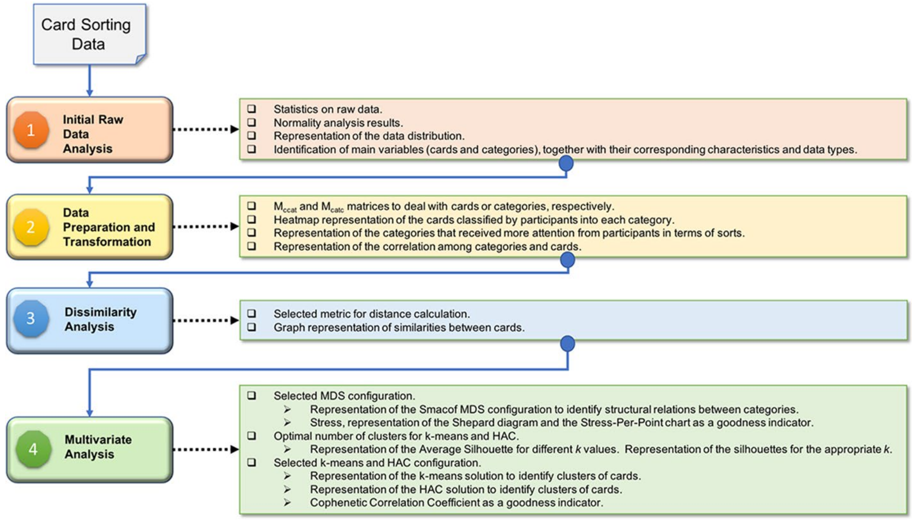
Card Sorting Data
1 Initial Raw Data Analysis
❑ Statistics on raw data.
❑ Normality analysis results.
❑ Representation of the data distribution.
❑ Identification of main variables (cards and categories), together with their corresponding characteristics and data types.
2 Data Preparation and Transformation
❑ M_{ccat} and M_{catc} matrices to deal with cards or categories, respectively.
❑ Heatmap representation of the cards classified by participants into each category.
❑ Representation of the categories that received more attention from participants in terms of sorts.
❑ Representation of the correlation among categories and cards.
3 Dissimilarity Analysis
❑ Selected metric for distance calculation.
❑ Graph representation of similarities between cards.
4 Multivariate Analysis
❑ Selected MDS configuration.
➢ Representation of the Smacof MDS configuration to identify structural relations between categories.
➢ Stress, representation of the Shepard diagram and the Stress-Per-Point chart as a goodness indicator.
❑ Optimal number of clusters for k-means and HAC.
➢ Representation of the Average Silhouette for different *k* values. Representation of the silhouettes for the appropriate *k*.
❑ Selected k-means and HAC configuration.
➢ Representation of the k-means solution to identify clusters of cards.
➢ Representation of the HAC solution to identify clusters of cards.
➢ Cophenetic Correlation Coefficient as a goodness indicator.

Fig. 1 The figure shows the steps of the proposed approach for quantitative analysis of card-sorting data in sequential execution order. Solid lines represent inputs, whereas dotted lines represent outputs

representations) generated. This way, for each step, the output (dotted line) represents the input (solid line) for the next step. The outputs are accumulative from one step to the following ones, as it might be necessary to re-visit the generated outputs in all steps.

Depending on the card-sorting data, specific parameters, goodness evaluation, and further recommendations are provided in all steps. Furthermore, it is worth mentioning that not all the recommended statistics and representations are strictly necessary for the majority of card-sorting analyses. Most techniques are complementary and, depending on the complexity of the data, might be used for further analysis. Therefore, to facilitate decision-making concerning the use of various techniques, selection criteria are provided at the end of each step. In general, the statistics and guidelines comprise broader support than necessary for most card sorting analyses. They provide a comprehensive approach generalizable to other practical situations and domains related to card sorting.

We start this section by briefly describing the datasets used to illustrate our approach and continue by discussing each of the steps of our approach in a separate subsection. To provide an easy summary in each step, we conclude the latter subsections with a list of discussed options and explicate which ones are essential, recommended, or optional.

3.1 Datasets

Two different datasets, briefly described below, were utilized to showcase the methodology and provide evidence and recommendations in each step. Both datasets are publicly available and can be downloaded from the source repository.

- *Dataset 1 (DS1)*: This is Donna Spencer's dataset [52], where card sorting was carried out to classify papers summited to the Information Architecture conference IA Summit into different topics [13] and create the web page for the conference. The card sorting used was open, single card sort, where 19 participants attempted to classify 99 cards, each into a single category. The participants created a total of 240 categories. Through a topic normalization process, Spencer reduced the number of categories to 57, thus raw data represented the relationship between each card and the number of times that it was classified into a particular normalized category.

- *Dataset 2 (DS2)*: This is a dataset published on the web page Cardsorting.net [53] as part of a tutorial [40]. The card-sorting task consisted of the classification of various food items into categories. In this open, single card sorting, 24 participants attempted to classify 40 cards into a single category. The participants created 240 categories, but no normalization process was carried out in this case. Thus, raw data represented the relationship between each card and the categories into which it was classified.

3.2 Initial raw data analysis

The raw data analysis of the card-sorting dataset as a whole is a necessary initial step to determine the type of raw data and identify principal variables (corresponding to cards and categories) and their characteristics to enable further analysis using advanced techniques. To this end, descriptive statistics based on the number of cards, categories, and possible card-category combinations are used.

It is worth mentioning that while in social-science-related card-sorting studies cards are considered stimuli (p) whereas categories are considered observations (n), in the present context p and n can represent both cards and categories. We consider p as the number of variables to analyze, and n as the number of observations of a variable. Table 1 shows the information extracted by exploring the raw datasets DS1 and DS2 described above.

As Table 1 shows, datasets are very different from each other in terms of raw data. While DS1's normalized categories have between 0 and 16 sorts observed per card, DS2 has 0 or 1. Neither dataset includes not available (NA) values (null or blank values related to non-usage). In general, it recommended that evaluators create only the cards necessary for a card sorting at hand and that participants use all the proposed cards. This simplifies the process and avoids NA values that can jeopardize the validity of statistics. When NA values are found, it is preferable to remove them from the data.

Table 1 Statistics for datasets DS1 and DS2

Statistics	DS1	DS2
Number of stimuli/cards (p)	99	40
Number of observations/categories (n)	57	240
Dataset size (card-category combinations)	5643	9600
Min value	0	0
Max value	16	1
Mean	0.33	0.09
SD	1.16	0.29
Variance	1.36	0.08
Median	0	0
NA (not available, null) values	0	0
Normality test at 95%	p-value < 0.05	p-value < 0.05

Both datasets contain a high number of 0 (zero) values, representing 84% and 90% of the data distribution for DS1 and DS2, respectively. This is a widespread occurrence in card-sorting raw data, especially when binary data is involved like in DS2, where each 0 represents the fact that a card has not been assigned to a concrete category.

In general, it is essential to choose a sufficient number of participants in quantitative card sorting studies. A higher number of participants usually yields more significant results and increases the power of the study. For DS1 and DS2, 19 and 24 participants (respectively) were recruited, which are acceptable sizes according to the literature [49, 50]. Also, the number of card sorts can be considered representative enough in both cases (i.e., 240 sorts normalized to 57 for DS1, and 240 sorts for DS2), generating a total of 5,643 and 9,600 card-category combinations for DS1 and DS2, respectively.

Furthermore, the normality of data needs to be verified. As Table 1 shows, univariate normality tests were carried out using the Shapiro–Wilk normality test for each variable and then the Anderson–Darling normality test for the whole dataset. The latter test is suitable for larger datasets (> 5,000). Since the normality tests for D1 and D2 returned *p-value* < 0.05, the results imply that data are not normally distributed. Therefore, non-parametric techniques should be utilized in further analysis.

A complementary analysis that might be done is to study the distribution of the data using kernel density charts, a non-parametric technique that does not assume any specific distribution for the data. Figure 2 shows the kernel density charts for DS1 and DS2, where data from each dataset have been analyzed as a univariate distribution. The data distribution demonstrates the non-normality of data and corroborates the previous analysis pointing to a high number of 0 values in both datasets compared with the frequency of the rest of the values. In addition, in DS2, the binary nature of the data is revelated, as the 'bumps' are specifically centered over 0 and 1.

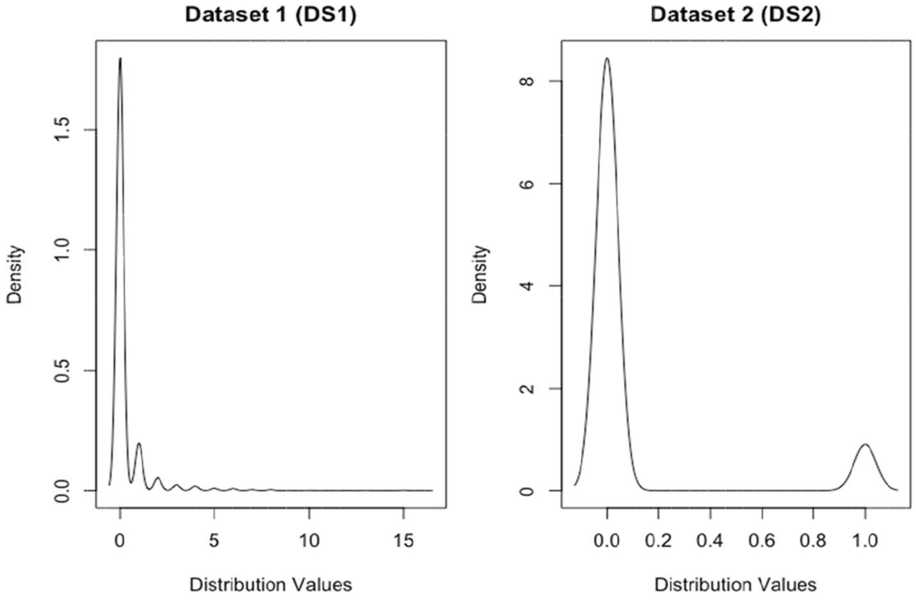

Fig. 2 Kernel density charts showing data distribution in datasets DS1 and DS2

Table 2 provides the summary of how different card-sorting designs (discussed in Sect. 2, such as open or closed sorts, single or multiple) provide clues that characterize the expected data. This information should be confirmed by the initial analysis of data and used in the subsequent steps of the analysis to select the suitable statistical techniques and their configurations.

It is easy to see from Table 2 that the expected data type greatly depends on the initial card sorting design. While closed card sorts mainly produce natural numbers (i.e., positive integers), open card sorts produce binary values (as long as the categories are not normalized).

As previously commented, normalization aims to remove duplicate categories or even merge similar ones to simplify and aggregate data. In such cases, positive integers rather than binary values are the expected data type. Since open card sorts are used much more frequently than closed ones, the data representation has very relevant implications on quantitative analysis.

Data based on positive integer values produce interval-scaled variables, as defined in the literature [54]. In the case of card sorting, this kind of data comes from the aggregation of binary matrices. The binary data produce two kinds of variables: symmetric and asymmetric (see Table 2). Symmetric binary variables codify two different states (e.g., member or non-member) having the same weight and preference (invariant characteristics). For such variables, the states are mutually exclusive. Thus, if a specific card is a member of a category, it is not a member of any other category. Symmetric binary variables are expected in open and single-category card sorts where categories have not been normalized. In contrast, asymmetric binary variables can codify more complex memberships since variables have different weights. In such a case, the lack of membership in a category is not the opposite of membership. Asymmetric binary variables can be found in open and nested-category card sorts, where a membership hierarchy is considered (e.g., a card classified into a subcategory belongs to both the subcategory and the parent category).

Furthermore, as Table 2 indicates, when open sort categories are normalized, variables represent aggregated values resulting in the same kinds of variables as closed card sorts. However, the distinction between symmetric and asymmetric variables is important, as asymmetric variables require non-invariant similarity measures [54] as detailed later on. For example, asymmetric positive integers resulting from the aggregation of asymmetric binary variables should be processed in a particular way to yield meaningful and comparable results. This might imply assigning specific weights or creating customized contingency tables and indexes for proximity values.

When card sort results in positive integers and larger values are identified, it might be advantageous to utilize z-scores (defined as $z = (x - \mu)/\sigma$, where x is the data, and μ and σ represent the mean and standard deviation) instead of the original data values to re-scale

Table 2 The table summarizes the expected data depending on the card-sorting design

Closed/open (*) card sort	Single/multiple card sorts	Single/nested category	Expected data
Closed	Single or multiple	Single	Positive integers
Closed	Multiple	Nested	Positive integers (asymmetric)
Open	Single or multiple	Single	Symmetric binary values
Open	Multiple	Nested	Asymmetric binary values

*Hybrid card sorts usually produce data comparable to open card sorts the dataset. This helps to smooth outliers and dominant values, which might affect the algorithms for multivariate analysis. However, the use of z-scores is not always necessary, as variables are usually expressed in the same units and have similar values within the same dataset. Therefore, data should be viewed comprehensively to decide on the advisability of using z-scores.

All the techniques mentioned above help carry out initial raw data analysis to understand the nature of card-sorting data. However, as previously commented, not all the techniques are strictly necessary. The likely selection criteria are:

- *Essential*: Statistics on raw data, which implies considering all the statistics presented in Table 1, except for the last row (normality test), to study and describe the card-sorting data.
- *Essential*: Identification of the main variables (cards and categories), their corresponding characteristics, and data types to carry out further analysis according to the guidelines shown in Table 2.
- *Recommended*: Normality analysis (the last row in Table 1) and, if needed, Anderson–Darling normality test, to use parametric or non-parametric statistics later on. This would be useful for the analysis of correlations if desired.
- *Optional*: Data distribution representation, such as kernel density charts (see Fig. 2), to observe the data density. This is complementary to the information shown in Table 1, and provides visual cues concerning the characteristics of the data.

3.3 Data preparation and transformation

As a rule, card sorting datasets have a similar appearance. The relationship between cards and categories is typically represented by one of the following matrix types:

- *Card-by-Participant Matrix* (M_{cp}) represents the categories assigned by participants to each card. Rows of the matrix represent cards as observations (n) and columns participants' sorts (p), where $M_{cp}(i,j)=c$ indicates that the card i has been classified into the category c by the participant j. M_{cp} is the most common representation used during the closed, single-category card sorting processes, or even when they are finished, as this representation makes the results easy to store, manipulate, and it helps when analyzing agreements among participants.
- *Participant-by-Card Matrix* (M_{pc}) represents the same information as M_{cp}, but it provides more flexibility if the same participant can appear more than once, for example, when representing data from multiple or nested-category sorts. In this matrix, rows represent participants as observation variables (n) and columns cards (p), where $M_{pc}(i,j)=c$ indicates that the participant i classified the card j into the category c.
- *Card-by-Category Matrix* (M_{ccat}) represents the classification of cards into categories. Rows represent cards as observations (n) and columns categories (p), where $M_{ccat}(i,j)=n$ indicates that the card i was placed into the category j by n participants. This representation is usually the result of the transformation of M_{cp} or M_{pc} matrices when sorting is done and in preparation for statistical analysis. Data from all card-sorting designs can be represented by this matrix. Note that when card-sorting variables are binary, it might be convenient to carry out the normalization of categories before creating the M_{ccat} matrix. This minimizes the number of categories and helps

the analysis. The representation is also suitable for analyzing the classification frequency, i.e., the categories in which a card has been classified most often.

- *Category-by-Card Matrix* ($\mathrm{M}_{\mathrm{catc)}}$ is the transpose of the matrix $\mathrm{M}_{\mathrm{ccat}}$, that is, $\mathrm{M}_{\mathrm{catc}} = \mathrm{M}^{\mathrm{T}}{}_{\mathrm{ccat}}$. It represents the classification of cards into categories, but in this case, rows represent categories as observations (*n*), and columns cards (*p*). The matrix entry $\mathrm{M}_{\mathrm{catc}}$ (*i,j*) = *n* indicates that the card *j* has been classified by *n* participants into the category *i*. The properties and the use cases described for the $\mathrm{M}_{\mathrm{ccat}}$ matrix also apply to $\mathrm{M}_{\mathrm{catc}}$.
- *Card-by-Card Matrix* (M_{cc}) represents the classification relationship among cards. This is a symmetric, square *c x c* matrix, where *c* is the number of cards. The entry M_{cc} (*i,j*) = *n* indicates that cards *i* and *j* have been classified *n* times into the same category. This kind of matrix is useful to study co-occurrences among cards, and it can be obtained from M_{cp}.
- *Category-by-Category Matrix* ($\mathrm{M}_{\mathrm{catcat}}$) represents the classification relationship among categories. This is a symmetric, square *cat x cat* matrix, where *cat* is the number of categories. The entry $\mathrm{M}_{\mathrm{catcat}}$ (*i,j*) = *n* indicates that the same *n* cards have been classified into categories *i* and *j*. This kind of matrix is useful to study co-occurrences among categories, locating similar ones that might be merged to reduce the total number of categories. $\mathrm{M}_{\mathrm{catcat}}$ can be obtained from $\mathrm{M}_{\mathrm{catc}}$.

Also, there are other matrices, less often used, that can be created on-demand for different analyses, for example, category-by-participant and participant-by-category matrices. However, for these matrices, categories are more readily represented than cards (which usually have longer labels) and results are more complex to use than when using the other matrices discussed above.

In general, $\mathrm{M}_{\mathrm{ccat}}$ and $\mathrm{M}_{\mathrm{catc}}$ comprise the more common input matrices for most statistical techniques, where cards and categories are clearly identified as variables for applying multivariate analysis. Nevertheless, M_{cp} and M_{pc} represent the most common structures to store the card sorting data, so a transformation step is necessary to create the referred matrices. An example of the procedure to transform M_{cp} into $\mathrm{M}_{\mathrm{ccat}}$ can be algorithmically represented as follows (c is the number of cards and pn is the number of participants):

```
Initialize $\mathrm{M}_{\mathrm{ccat}}$ to 0
For i = 1 to c
   For j = 1 to pn
      $\mathrm{M}_{\mathrm{ccat}}[i,\mathrm{M}_{\mathrm{cp}}[i,j]] = \mathrm{M}_{\mathrm{ccat}}[i,\mathrm{M}_{\mathrm{cp}}[i,j]] + 1$
```

The new matrix $\mathrm{M}_{\mathrm{ccat}}$ is a *c x cat* matrix, where *cat* is the number of categories (*cat* = *max*(M_{cp})). It is possible to obtain $\mathrm{M}_{\mathrm{catc}}$ by calculating the transpose of $\mathrm{M}_{\mathrm{ccat}}$, as commented previously.

As for DS1 and DS2 datasets, no specific transformations on raw data are needed. DS1 is readily represented by $\mathrm{M}_{\mathrm{ccat}}$ matrix, with interval-scaled variables and normalized categories, and $\mathrm{M}_{\mathrm{catc}}$ represents the dataset DS2, where categories are not normalized, and data consists of symmetric binary values. It is easy to transpose their respective matrices to carry out different statistical calculations on cards and categories in both cases.

$\mathrm{M}_{\mathrm{ccat}}$ and $\mathrm{M}_{\mathrm{catc}}$ matrices might be used to generate a heatmap for an initial overview of the classification and observe the number of cards classified by the participants

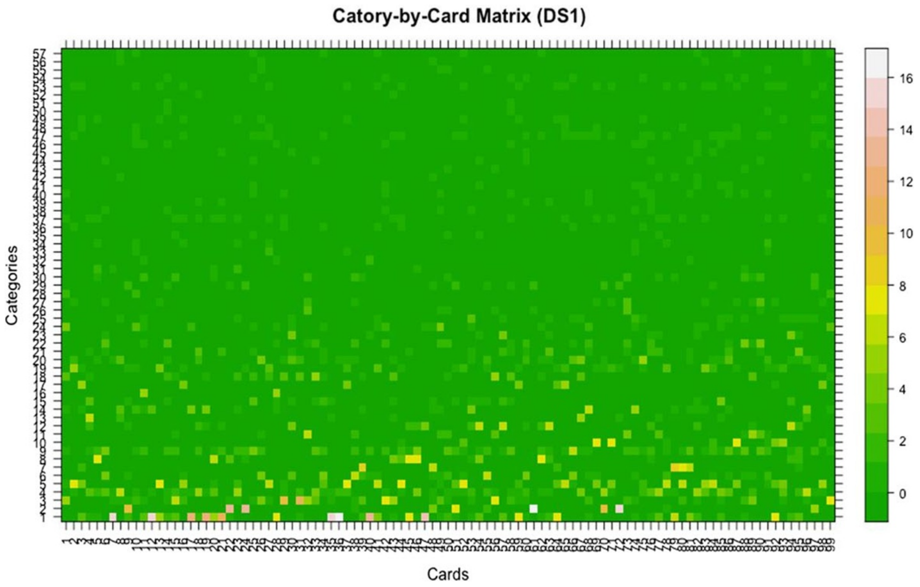

Fig. 3 Classification heatmap for DS1 obtained from its M_{ccat} matrix. Categories and cards have been assigned numerical values for the sake of the clarity of visualization

into each category. Figure 3 depicts such a heatmap for DS1, derived from its M_{catc} matrix. As the figure shows, the bottom half of the categories in the heatmap representation seem to contain higher numbers of classified cards (where 16 is the maximum value represented by the lightest color), whereas the top half of the categories seem to contain fewer cards. As DS2 contains only binary data, its heatmap is likely to be less meaningful.

M_{catc} matrices might be used for other complementary analyses. For example, as in Fig. 4, to show the number of participants' sorts per category. Categories have been arranged by the number of sorts, in decreasing order. Accumulative percentages (25, 50, 75, and 95%) are shown to ease reasoning around the numbers of sorts, categories, and cards.

In the case of DS1, one category (numbered 1) includes all 240 sorts. Furthermore, one can easily see that categories 1, 2, 3, 4, 5, 8, 9, and 20 comprise 50% of participants' sorts, while the categories to the right of the dashed 95% line represent less than 5% of sorts (starting with categories 41, 44, and further on the right). In the case of DS2, the figure shows that binary data produces a different chart. Here, category 143 represents a total of 14 sorts, which is the maximum number. Categories appearing on the left, from 143 to 125, represent 25% of participants' sorts, while categories from 60 onwards represent less than 5%. In both cases, the analysis helps to find the categories that have received more attention, higher engagement, and motivation from participants.

Furthermore, there is a correspondence between the number of sorts and the number of different cards classified into each category, which is especially useful when datasets are interval-scaled. For instance, for DS1, category 1 includes the highest number of sorts (240) and one of the higher numbers of distinct classified cards (50). The highest number of distinct classified cards (57) corresponds to category 4, which also had

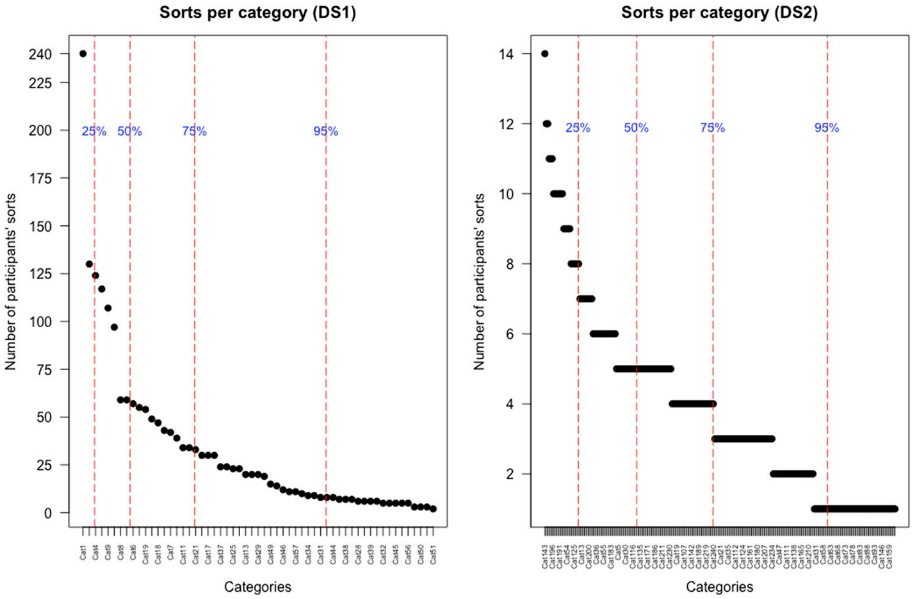

Fig. 4 Sorts per category for datasets DS1 (left) and DS2 (right). Category names have been shortened and assigned numerical values for the sake of the clarity of visualization

a high number of sorts. This indicates that the analysis helps identify categories into which high or low numbers of distinct cards were placed.

By contrast, for DS2, categories with higher numbers of participants' sorts have a one-to-one correspondence to categories with higher numbers of distinct cards due to the binary nature of relationship between cards and categories.

Another example of complementary analysis uses M_{cc} and M_{catcat} matrices (commonly known as item-by-item matrices) to analyze direct relationships among two variables of the same type. As previously commented, M_{cc} and M_{catcat} are symmetric matrices representing, respectively, the number of times that two cards have been classified into the same category and the number of times when the same cards were classified into two categories. These matrices might be obtained from M_{cp} and M_{catc}. However, a comparable relationship analysis can be achieved using correlation and proximity matrices obtained from M_{ccat} and M_{catc}. While proximity matrices are addressed in the following subsection, we focus on correlation matrices to look into relationships among cards and categories.

Correlation matrices are used to analyze dependencies between different variables simultaneously (i.e., cards or categories). A correlation matrix is a symmetric matrix containing correlation coefficients, which are real numbers between −1 and 1 (where −1 indicates a negative relationship, 0 no linear relationship, and 1 a positive relationship among variables). There are different techniques, both parametric (e.g., Pearson) and non-parametric (e.g., Spearman and Kendall), for finding correlation coefficients. As discussed earlier, variables in DS1 and DS2 are not normally distributed, so non-parametric techniques should be applied to calculate their corresponding correlation matrices. One of the most frequently used is the Spearman correlation. Figure 5 shows the correlation matrix, obtained from M_{ccat}, for different category variables in DS1, based on Spearman

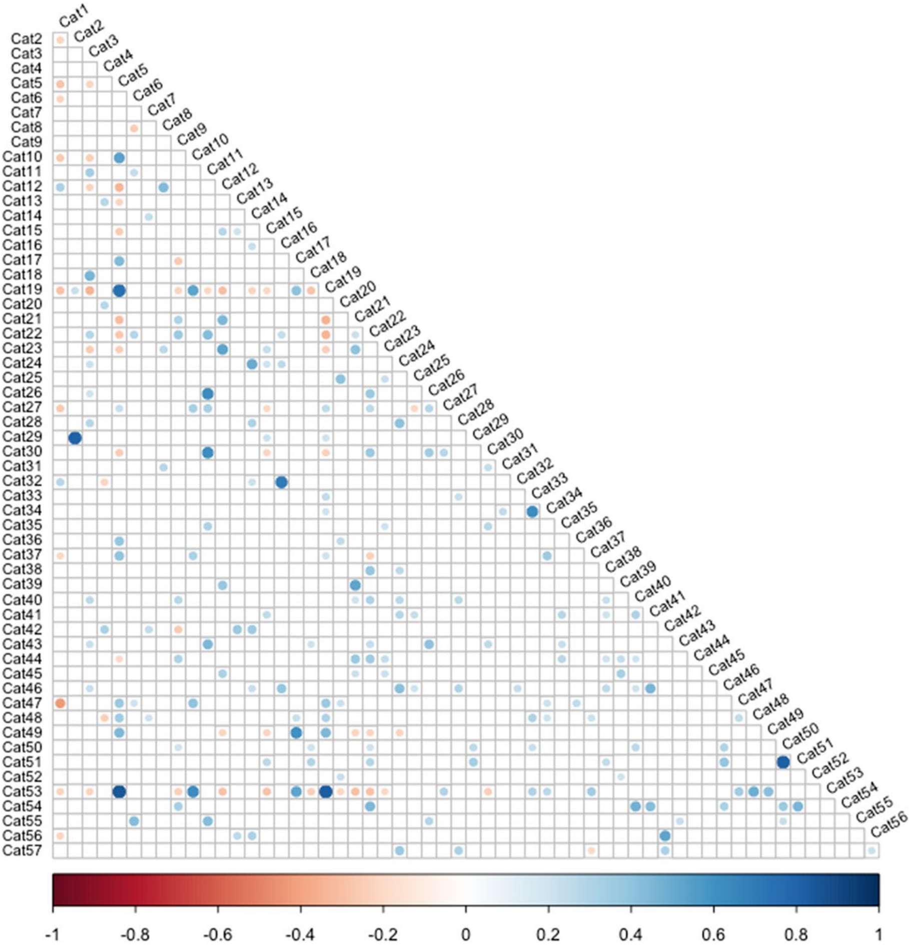

Fig. 5 Spearman correlation matrix at 95% obtained from M_{ccat} and representing relationships among categories in DS1. Only significant correlations (p-value < 0.05) are shown. Category names have been shortened and assigned numerical values for the sake of the clarity of visualization

correlation coefficients at 95%. Colored circles show degrees of correlation between categories in DS1, where the color bar gives a visual clue as to whether the correlation is positive, negative, or no correlation. Only significant correlations (p-value < 0.05) are shown. The figure shows strong linear correlations, for example, between categories 2 and 29, 5 and 19, 16 and 32, as well as cards classified into those categories. This analysis helps detect categories that classify the same cards or even the opposite ones (e.g., categories 1 and 47). The correlation could also be applied to card variables using the M_{catc} matrix.

The techniques described above are intended to prepare and transform card-sorting data into representations best suited for analysis. Again, not all of the mentioned techniques are strictly necessary. The following describes selection criteria at this step:

- *Essential*: Create (the most frequently used) Card-by-Category (M_{ccat}) and Category-by-Card (M_{catc}) matrices, suitable for most statistical techniques.

- *Optional*: Create Card-by-Participant (M_{cp}), Participant-by-Card (M_{pc}), Card-by-Card (M_{cc}), and Category-by-Category (M_{catcat}) matrices. These matrices are optional; however, M_{cp} and M_{pc} are common structures to store the card sorting data, and the transformation described above might be necessary to create the working matrices M_{ccat} and M_{catc}.
- *Essential*: Heatmap representation of the cards classified by participants in each category. This information is useful to study the classification's big picture, and see, at a glance, distribution and gradient of cards in various categories (as shown for DS1 in Fig. 3).
- *Recommended*: Create sorts by category representation (as in Fig. 4) to identify categories that received more attention from participants involving a higher number of attempts to classify the cards and indicating more motivation from participants, as previously discussed.
- *Optional*: Represent correlation among categories or cards. This is complementary to the heatmap representation and implies creating a visualization such as the one depicted in Fig. 5, analyzing dependencies between cards or categories simultaneously.

3.4 Dissimilarity analysis

A dissimilarity analysis focuses on understanding how far two variables are from each other. It uses a dissimilarity matrix, which can be defined as a square matrix where each cell contains information about the differences between two variables. Dissimilarity and similarity matrices are commonly known as proximity matrices, also referred to as resemblance matrices [54]. Such matrices are calculated from pairwise judgments based on similarities or dissimilarities.

In the specific case of card-sorting analysis, dissimilarity matrices, usually known as distance matrices, become more relevant, as they are commonly utilized as input for multivariate techniques. In general, the type of variable to analyze is relevant to determine the suitable metric for the distance calculation, which mainly depends on the data type, as commented in previous subsections.

As shown in Table 3, the choice of the metrics for distance calculations depends on the data type of the variables (see also Table 2). For example, while interval-scaled and symmetric binary variables require invariant measures, asymmetric binary variables require non-invariant ones to calculate the corresponding distance matrices.

For interval-scaled variables, it is common to use Manhattan and Euclidean metrics, where the Euclidean distance is the one that is most frequently used in card sorting. The Euclidian distance generates a metric space where the distance among two variables can be measured by the length of a *straight line* between them, that is, the minimum distance between two points. Although other metrics, such as Gower, could be applied in the case of symmetric binary variables, the Euclidean metric can also be used for such variables.

On the other hand, Jaccard metric can be used for asymmetric binary variables. One of the most utilized methods to calculate distances for asymmetric binary variables is based on the Jaccard coefficient (i.e., *S-coefficient*), which is best suited for card-sorting variables

Table 3 Recommended metrics for distance calculation in card sorting depending on the type of the variables involved

Variable type	Recommended metric for distance calculation
Interval-scaled / Symmetric Binary	Euclidean
Asymmetric Binary	Jaccard Coefficient

representing duplicated and nested categories [40]. As for symmetric binary variables, it is possible to work with binary variables as if they were interval-scaled [53]. This happens when data come from aggregated matrices, where a topic normalization process is carried out. For card-sorting analysis, this decision does not explicitly affect the distance calculations based on Euclidean distances. Although card sorting usually generates binary data, aggregated matrices that result from topic normalization processes can, to some extent, be considered as distance matrices.

Other ways to calculate dissimilarities are possible. For example, correlation matrices might be used. As previously discussed, bivariate correlations measure the degree of relatedness between two variables. The correlation coefficients can be transformed into dissimilarities using the formula: $d(v_1, v_2) = (1 - R(v_1, v_2))/2$, where $d(v_1, v_2)$ represents the dissimilarity between variables v_1 and v_2, and $R(v_1, v_2)$ represents the correlation coefficient (parametric or non-parametric) for variables v_1 and v_2. The formula shows that variables with a high positive correlation coefficient are considered to be very similar (dissimilarity value close to 0), whereas variables with a very negative correlation coefficient are considered dissimilar (dissimilarity value close to 1).

Dataset DS1 contains symmetric aggregated data. Thus, variables can be treated as symmetric interval-scaled variables. However, DS2 has non-aggregated data, implying symmetric binary variables. In general, aggregated matrices approximate distance matrices, but binary matrices might not fully capture the distance among variables. This drawback can be mitigated by having a representative number of participants [49], then geometrical distances can be calculated using Euclidean distance in both cases.

Dissimilarity matrices can also be represented using graphs. Such representations are helpful to visually analyze similarities and dissimilarities between variables (i.e., cards and categories). For example, Fig. 6 shows a graph where nodes represent card variables from

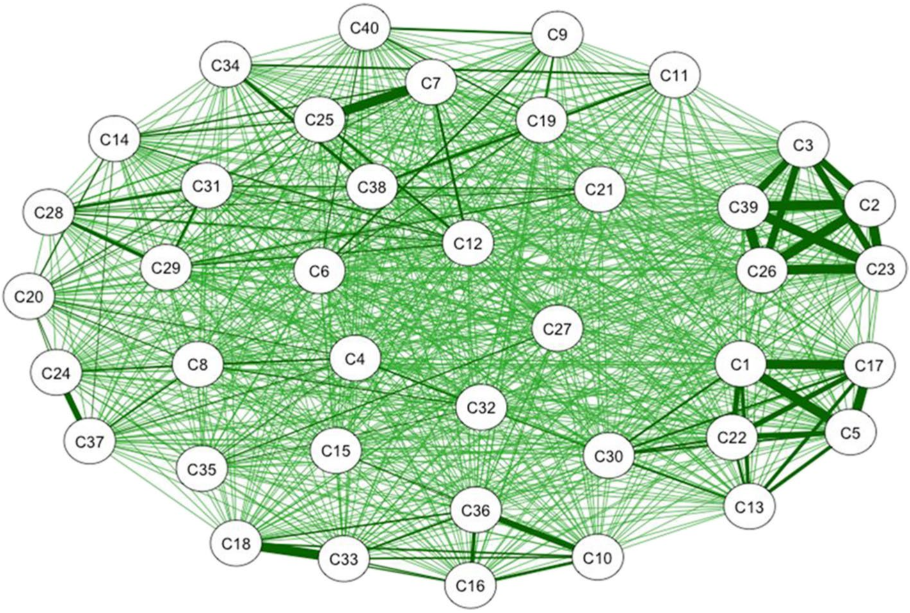

Fig. 6 Graph representing similarities among cards in DS2. Card names have been shortened and assigned numerical values for the sake of the clarity of visualization

dataset DS2 (M_{ccat} matrix). There, thicker lines (edges) represent higher similarities among nodes that, in this context, indicate that such cards are classified into similar categories. For instance, cards C2 (*apple*), C3 (*banana*), C23 (*orange*), C26 (*pineapple*), and C39 (*watermelon*) are considered to be very similar, as they have been classified into similar categories such as *fruit_and_veggie*, *fruits*, and similar.

The techniques presented in this sub-section are aimed at performing dissimilarity studies, which are useful for the multivariate analyses discussed later on. As previously commented, not all the techniques are strictly necessary. The likely selection criteria, in this case, can be the following:

- *Essential*: Select the metric for distance calculation. This implies determining the suitable metric for the distance calculation, which mainly depends on the data type. To this end, the information shown in Table 3 has to be considered.
- *Recommended*: Graph representation of similarities between cards. This can be useful to represent the dissimilarly matrices using a graph to visually analyze similarities and dissimilarities among cards, which can be complementary to the correlation and heatmap commented above. The graph depicted in Fig. 6 can be used for this purpose. In this case, relations are easy to detect, as thicker lines (edges) represent higher similarities among cards in this case.

3.5 Multivariate analysis

Multivariate analysis helps to identify structures and relations between different variables comprising the data and might include applications like dimensionality reduction, clustering, or multidimensional scaling, to name just a few.

For card sorting, the most common multivariate analysis methods are the principal component analysis [55], factorial analysis [56], multidimensional scaling [16, 57, 58], clustering methods such as hierarchical agglomerative clustering [15, 54, 59–61] and k-means [12, 54, 62]. The dendrogram [54, 63], derived from hierarchical agglomerative clustering, is probably the most popular agglomerative chart used to analyze card sorting results. Other multivariate techniques, such as multidimensional scaling and k-means, are increasingly used to study structural dependencies among variables (for example, categories) and topic normalization. In contrast, principal component analysis and factor analysis are used less due to their sensitivity to the input data. Sometimes, they suffer from singularity problems due to linear dependencies among variables, which often occur due to the nature of card-sorting data. Most of these statistical techniques are complementary; however, it is recommended to use more than one [13] since the result comparisons enrich the card-sorting analysis and help gain further evidence.

The subsections that follow provide guidance on how to apply multivariate techniques, together with the configuration of parameters and graphical representations that are suitable and frequently used in card-sorting analysis.

3.5.1 Multidimensional Scaling

Multidimensional scaling (MDS) is a multivariate technique that transforms a proximity matrix into a configuration of points (a map of coordinates) in an *n*-dimensional space, where distances among points maximize similarities among variables to the largest degree possible. The result is an *n*-dimensional classification, where all combinations of analyzed variables are considered. MDS helps uncover hidden structures among variables that are difficult to find using simple statistics and visualizations.

In the context of card sorting, MDS can be used to analyze similarities among card or category variables in an *n*-dimensional space, where *n* is usually 2 or 3 (that is, a two or three-dimensional configuration space). For this task, MDS implements algorithms that find the optimal representation of analyzed variables in the *n*-dimensional space, using a minimization function to evaluate different configurations and maximize the goodness of fit. In general, the graphical representation obtained from an MSD analysis provides the interrelation of the involved variables; the closer two variables are in the *n*-dimensional space, the higher correlation they have. These features make MDS a valuable tool to study the topology of the variables in the dataset and represent the results in a geometric space visually.

In general, the correct application of MDS greatly depends on the metric used to calculate the dissimilarity matrix and the precision of the configuration of parameters that different MDS approaches provide. Thus, a proper setup is necessary for obtaining relevant results. A range of MDS approaches might be used. For example, Torgerson represents the classical approach. However, Smacof, PROXCAL, and INDSCAL [57] are used more frequently. In particular, Smacof has been improved over time, providing interesting utilities to work with different metric and non-metric versions and distances computation. It is now likely the most frequently used approach. PROXCAL, based on an early consolidated version of Smacof, attempts to find the least-squares representation combined with an iterative majorization in a low-dimensional Euclidean space [64]. INDSCAL is a dimensional, weighted approach suitable for modeling individual differences in MDS, allowing evaluators to explore differences between groups of participants [65].

An indicator of goodness of fit, called *Stress* [66], might be used to determine the suitability of an MDS solution. It can be considered as a loss function (objective function). A perfect MDS solution is expected to have a *Stress* value of 0, that is to say when the distances in the *n*-dimensional space depict the data with no representation errors. *Stress* can be used to compare the goodness of fit for different MDS solutions. *Stress* is affected by the number of points—distances in MDS solutions grow as its function. The dimensionality of the MDS space also affects the *Stress* value: higher dimensionality results in lower *Stress*. Outliers also affect *Stress*, so it is preferable to eliminate them to reduce the total *Stress*. In general, the *Stress* should be evaluated according to the particular MDS accomplished.

The goodness of MDS can be represented graphically using a Shepard diagram [57]. A Shepard diagram is a scatter plot that shows how far apart are the data points before and after the transformation, indicating how well are the actual relations among variables reflected in the plot. In a Shepard diagram, dissimilarities are shown on the *x*-axis, and the fitted MDS distances are on the *y*-axis. Also, information about how dissimilarities and disparities (i.e., *d-hats*) are related to each other is added using an optimal scaling transformation that is usually a monotone regression in ordinal MDS and a linear transformation in interval MDS. In general, when evaluating an MDS fit, it is worth noting that a lower *Stress* does not always imply the best solution. Instead, it is necessary to analyze the Shepard diagram and even the *Stress-Per-Point* (SPP) chart [66], to simplify the model or select the one with a weaker dissimilarity-distance fit.

Unfolding is a related method that is better suited for dealing with ordinal data (e.g., ratings or preferences). While MDS deals with variable-based dissimilarities, *unfolding* can represent both observations and variables in a joint space. In the context of card sorting, *unfolding* can be directly applied by utilizing the $\mathrm{M_{cp}}$ matrix, which contains each participant's ratings (categories) for each card.

Table 4 Recommended MDS configurations for card sorting depending on the kind of variable

Variable type	Matrices needed	Distance metric	MDS configuration
Interval-scaled	Interval-scaled M_{ccat} or M_{catc} matrices to analyze cards or categories, respectively	Euclidean	Interval (metric)
Symmetric binary	Symmetric binary M_{ccat} or M_{catc} matrices to analyze cards or categories, respectively	Euclidean	Ordinal (non-metric)
Asymmetric binary	Asymmetric binary M_{ccat} or M_{catc} matrices to analyze cards or categories, respectively	Jaccard	Drift vector model
Ordinal	Ordinal M_{cp} matrix including card-by-participant data	–	Metric unfolding

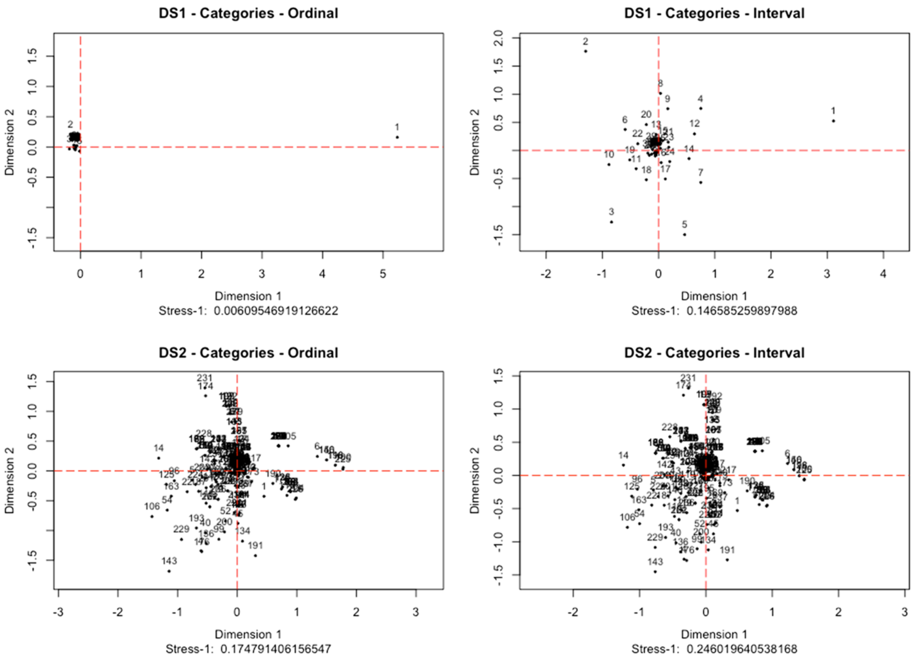

Fig. 7 Non-metric (ordinal) and metric (interval) MDS configurations for DS1 (top) and DS2 (bottom) datasets based on category variables. Category names have been assigned numerical values for the sake of the clarity of visualization

As Table 4 shows, MDS configuration depends on the card-sorting data. As mentioned, the Smacof approach is highly recommended for card sorting since it provides configuration facilities for different data types. Although some authors point out that ordinal and interval MDS often lead to similar results, interval MDS is preferred for interval-scaled data, as it provides more robust results than ordinal MDS that tends to over-fit the data. By contrast, ordinal MDS is best suited for symmetric binary variables. Euclidean distances are recommended since they guarantee the geometry of the information used as input for MDS (for example, Manhattan distance increases the risk of finding the false minimum). Some authors point out that the distance metric is sometimes irrelevant as long as the measures are linearly related. Also, for better computation, distances should be symmetric. Otherwise, it is recommended to use other approaches, such as the drift vector model where data is decomposed into separate symmetric and asymmetric data [57]. For high dimensional data, it is recommended to use other approaches, such as t-SNE (Distributed Stochastic Neighbor Embedding), a non-linear algorithm that reduces the dimensionality to tackle a high number of variables [67]. In general, M_{ccat} and M_{catc} matrices can be used for analyzing cards or categories, respectively. As commented before, *unfolding* can be also used to analyze data directly from the M_{cp} matrix (no distance matrix is needed), which provides information about the sorts carried out by each participant.

For DS1 and DS2 datasets, the two-dimensional Smacof (MDS) technique was applied to M_{ccat} or M_{catc}. First, dissimilarities matrices were calculated using Euclidean distances, then an MDS analysis on categories chosen to identify conceptual groupings.

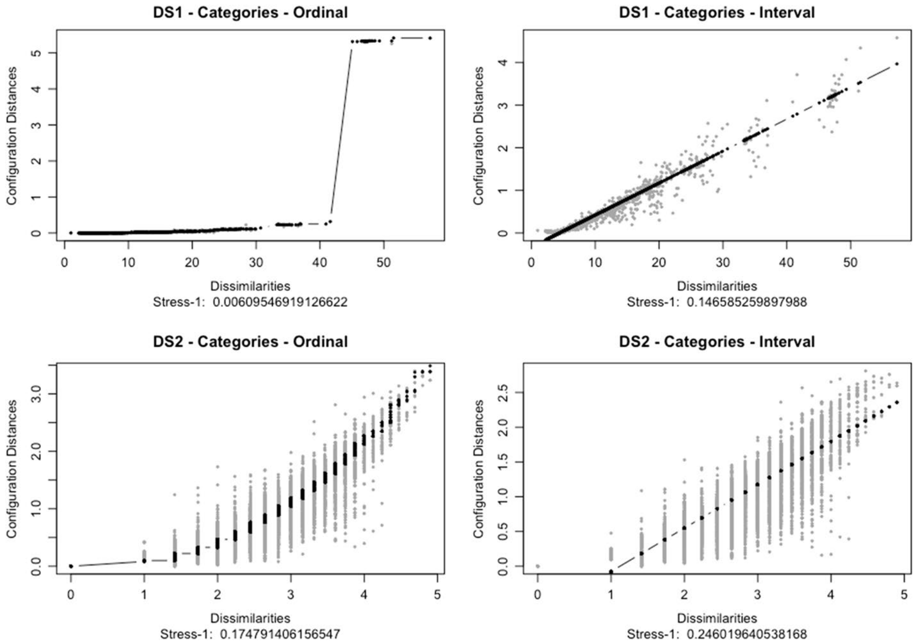

Fig. 8 Shepard diagrams for non-metric (ordinal) and metric (interval) MDS configurations for DS1 (top) and DS2 (bottom) datasets based on category variables

Figure 7 shows the resulting MDS configuration for DS1 (on the top) and DS2 (on the bottom), based on category variables (M_{catc} matrix) and using different MDS configurations (ordinal and interval). The *Stress* value is shown at the bottom of each chart.

As shown in Fig. 7, ordinal and interval MDS configurations provide similar results for DS2. However, the *Stress* in the ordinal MDS is lower (0.174). As for DS1, the ordinal MDS produces a degenerate solution (i.e., although having a rather lower *Stress*, the data seem dense and do not show properly in the chart), as is evidenced by the gross step in the chart.

To facilitate further analysis, Shepard diagrams (for the configurations shown in Fig. 7) are depicted in Fig. 8.

As Fig. 8 shows, the ordinal MDS for DS1 represents an overfit, whereas the interval MDS produces a more natural behavior over the linear transformation, which suggests that an interval MDS results in a better choice, with a *Stress* value of 0.146.

The last analysis consists of observing the *Stress-Per-Point* diagrams, shown in Fig. 9. SPP diagrams are generated from the MDS configurations and indicate the percentage of *Stress* contributed by each category.

(i.e., how they contribute to the misfit). Thus, the categories with a higher percentage should be studied, as they might provide evidence of mistakes or discrepancies with the remaining categories) and possibly lead to their removal to obtain a better configuration.

In the case of DS1, categories 2, 3, 4, and 5 seem to be outliers and misfit the model. Analyzing such categories, it is noted that most of them have higher mean and average values. After removing such categories and refitting the model, an improved *Stress* of 0.129 is obtained. For DS2, category 134 represents an outlier and also misfits the model according to the SPP chart. Removing such a category and refitting the model, the *Stress* is slightly

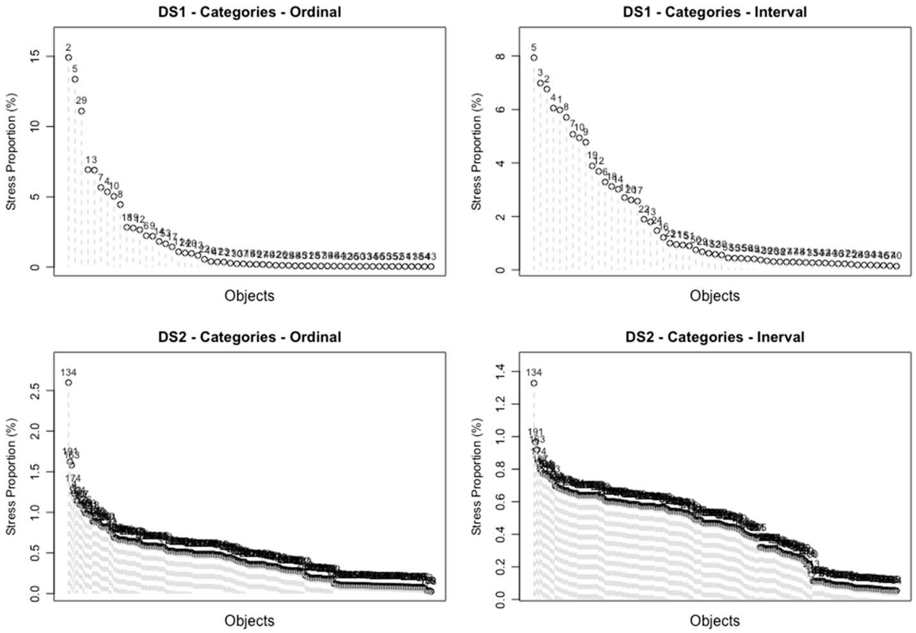

Fig. 9 Stress-Per-Point diagrams for non-metric (ordinal) and metric (interval) MDS configurations for DS1 (top) and DS2 (bottom) datasets based on category variables. Category names have been transformed into numbers for the sake of the clarity of visualization

improved to 0.172. In conclusion, using an interval configuration for DS1 and an ordinal configuration for DS2 seems to be the best choice, as recommended in Table 4.

The MDS configurations for DS1 and DS2 provide some recognizable groupings (see Fig. 7 showing interval categories for DS1 and ordinal for DS2). For instance, in DS2 dataset, categories 54 (*breads*), 106 (*baked_goods_breakfast*), 125 (*breakfast_foods*), and 96 (*break_fast_items*) have been placed closer to each another. After finding the MDS configuration, the next step is to look into scaling by giving the appropriate meaning to the two-dimensional partition, which generates four different quadrants that group different categories according to the MDS configuration obtained. This conceptual division is represented in upper-left to bottom-right separate spaces, thus generating the quadrants Q1 – Q4. The categories included in each quadrant can be analyzed according to the domain of each dataset. In the case of the DS1 dataset, Q1 might represent *fundamental* and *state-of-the-art* categories related to IA, Q2 those related to a *professional vision* of the IA, Q3 *specific* and *advanced IA research topics*, and Q4 categories related to *synergies among IA and the user* (here, provisional names of categories are italicized). This means that the Dimension 1 (D1) may represent the variation among *theoretical* (-D1) and *practical* (+D1) *IA concerns*, whereas Dimension 2 (D2) may represent the variation among *IA topics* that have a more impact on *user and research* (-D2) and those that can be meant as *more contextual* (+D2). For DS2 dataset, Q1 might represents categories related to *beverages and drinks*, Q2 to *dairy and healthy food*, Q3 *breakfast* and *pastries* categories, and Q4 includes categories that can be grouped as *snacks and sides*. This means that Dimension 1 (D1) may represent the variation among *fatty* (-D1) and *healthy* (+D1) *food*, whereas Dimension 2 (D2) may represent the variation among the *food that should be eaten less often* (-D2) and *more frequently* (+D2) *consumed*.

3.5.2 Finding the optimal number of clusters

MDS allows for scaling the data to two or three dimensions and represents a broad clustering mechanism that can be used to analyze the topology of the data and establish conceptual relationships among them. However, it is frequently necessary to find smaller groups to bring to light small or medium size relationships among the data. This mechanism is known as clustering, that is, gathering data (i.e., variables) with similar characteristics into groups.

As before, the selection of an appropriate clustering approach depends on the type of the variables and on the purpose of the clustering itself. Therefore, a configuration of parameters should be considered in advance [51, 54]. Most clustering approaches require an optimal number of clusters as input, and thus, this number should be determined first. Finding the right number of clusters is an optimization problem where the objective is to get a minimal k value that maximizes the clustering solution for a given dataset. The cluster validity should be considered to ensure goodness.

There are different approaches for determining the optimal number of clusters. The three most common ones are *elbow*, *average silhouette*, and *gap statistic* [54, 68]. *Elbow* is a heuristic based on the graphical representation of the explained variation (e.g., the total within-cluster sum of squares) as a function of the number of clusters (i.e., k), trying to find the elbow (or the knee) of the curve as the optimal k. The *gap statistic* method can also be applied to any clustering approach. It utilizes Monte Carlo simulations to generate an appropriate null reference distribution that is compared to the total within-cluster variation for different k values. The *average silhouette* method is the most frequently used one to analyze the cluster validity. Silhouette evaluates the partition of data regardless of the clustering approach utilized, thus being useful for comparing different clustering solutions obtained from different algorithms. A high average silhouette represents good clustering. Thus, the way to find the optimal number of clusters is based on obtaining the value k that maximizes the average silhouette. A silhouette index $s(i)$ indicates how similar the observation i is to others belonging to the same cluster. Commonly, $s(i)$ ranges from 1 to -1, where 1 indicates that observation i is well clustered and -1 that is poorly clustered (i.e., it should be moved to another cluster to improve the clustering solution). To find k, the average of silhouettes $\bar{s}(k)$ is used. This value is commonly used to measure the overall cluster validity through the silhouette coefficient, which is defined as the largest one of all over k ($SC = \max_{k} \bar{s}(k)$).

The *average silhouette* method can be applied to DS1 and DS2 datasets using the $\mathrm{M}_{\mathrm{ccat}}$ matrices, as the objective is to find clusters of cards. For DS1, z-scores for cards were used instead of the original data. Z-scores were a better option due to the variation in values included in the dataset DS1 (as discussed in Sect. 3.2) that affects the clustering algorithms and estimate of the optimal number of clusters k.

Figure 10 shows the $\bar{s}(k)$ scores for different k values ranging from 2 to 40. As shown, the maximum average silhouette value is obtained at $k = 15$ ($SC = 0.30$) for DS1. For DS2, the maximum average silhouette is at $k = 12$ ($SC = 0.39$).

Furthermore, the validity of the clustering should be analyzed, along with the opportunities for further improvements. Figure 11 shows the silhouette cluster validity for the optimal number of clusters for DS1 and DS2, $k = 15$ and $k = 12$, respectively. For DS1, 15 clusters of different sizes were found, containing (from left to right) a total of 9, 6, 2, 12, 5, 5, 10, 7, 9, 5, 4, 4, 4, 8 and 9 cards (which represented conference *papers* in this dataset). All silhouette values $s(i)$ ($i = 1, \ldots, 99$) were positive except for one card,

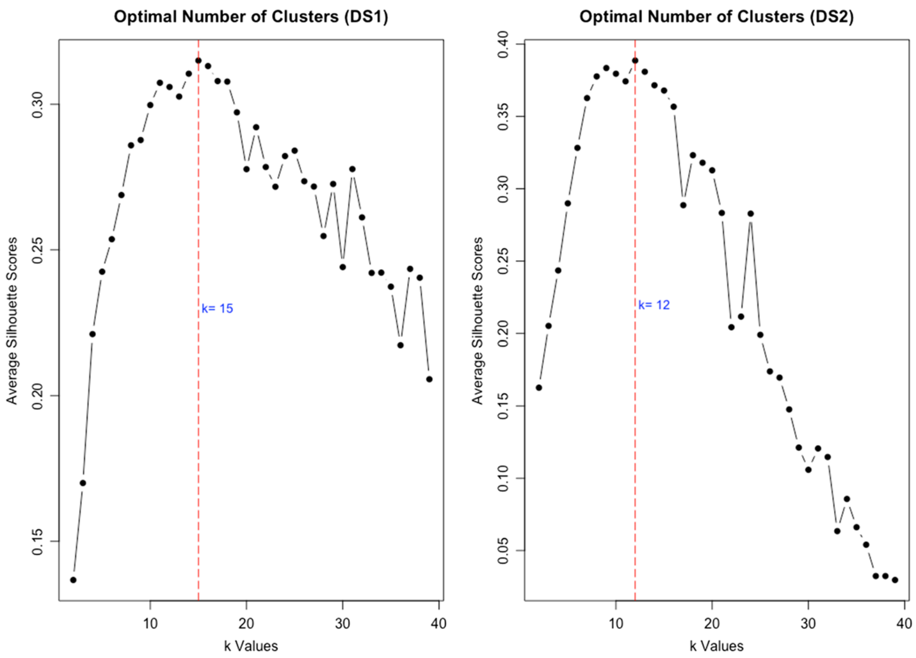

Fig. 10 Based on the *average silhouette* method, DS1 has the optimal number of clusters at $k = 15$ and DS2 at $k = 12$ (shown by dashed vertical lines)

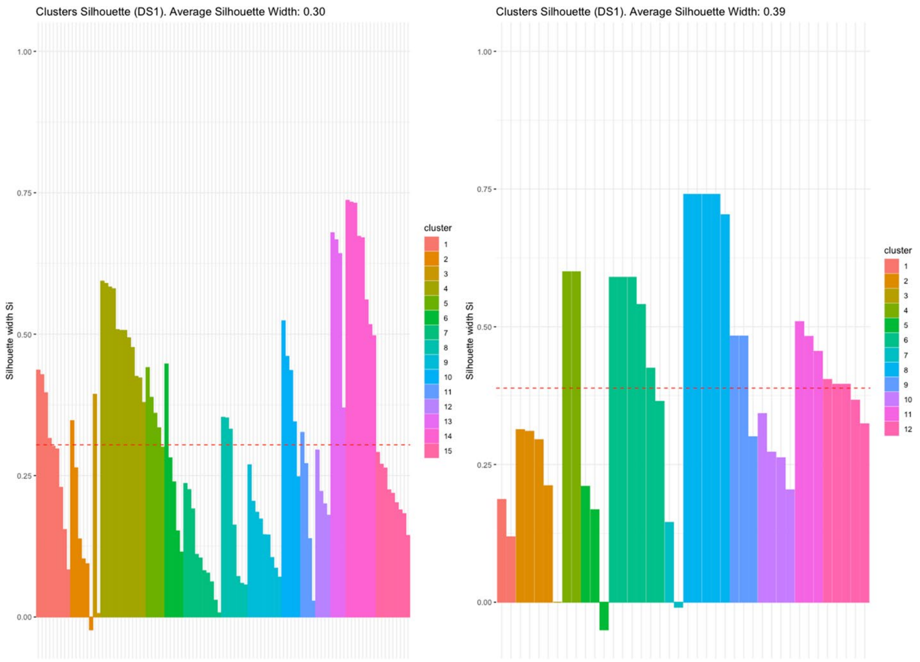

Fig. 11 Silhouette cluster validity for $k = 15$ (left) and $k = 12$ (right) based on the optimal number of clusters for DS1 and DS2, respectively. $\overline{s}(k)$ values are represented with a horizontal dashed line

89, which had a negative value of $s(89) = -0.02$. Although the negative value is small (close to zero), it might indicate that this card should be moved to another cluster to improve the overall clustering solution. Analyzing the clusters in more detail is needed to see if this would be an improvement—in this case, the card is best left in the same cluster, so no change is recommended. For DS2, 12 clusters were obtained, containing (from left to right) a total of 2, 4, 1, 2, 3, 6, 2, 5, 3, 4, 3, and 5 cards (representing *foods* in this dataset). Some clusters, as the number 8 (containing 5 cards), have high average $s(i)$ scores, indicating higher cohesion. Almost all $s(i)$ scores $\forall\ i=1, \ldots, 40$ are positive, with two exceptions, cards 8 (*cereal*) and 14 (*doughnuts*), where $s(8) = -0.050$ and $s(14) = -0.009$, respectively. As in DS1, both scores have small negative values, but it still might indicate that these cards should be moved to other clusters to improve the overall clustering solution. Analyzing the clusters closer, card 8 should stay in its cluster (classified together with cards 4 (*bread*) and 32 (*rice*)). However, if the card 14 (*doughnuts*) is moved together with the card 20 (*muffin*) to cluster 4, with cards 24 (*pancakes*) and 37 (*waffle*), the optimal number of clusters becomes 11 and gives a better overall clustering solution.

Considering the number of clusters obtained, probably some of them might be merged and others split up. This implies that the proposed values for k are only estimations that should be reinforced by studying the characteristics of each dataset. In general, it can be concluded that a suitable number of clusters for DS1 would be in the range 14–16. As for DS2, a number in the range 10 to 12 makes sense. To be more specific, and taking into account the aforementioned analysis, $k=15$ for DS1 and $k=11$ for DS2 could be considered as acceptable clustering solutions according to the cluster validity values obtained. Graphic representations of clusters, discussed in the following subsection, might help to confirm the validity of the analysis visually. In general, this kind of analysis helps reduce the original number of categories to a smaller number, which is an extraordinary gain in open card sorts. For instance, in DS2 a total of 240 categories were originally included in the dataset. Nevertheless, this number can be drastically reduced to 11 categories to classify a total of 40 cards. Concerning DS1, the dataset was previously normalized to already represent a reduced number of categories. However, and according to the analysis presented in this section, even the 57 categories can be conceptually reduced to a smaller number (e.g., 15) to classify the 99 cards.

3.5.3 Cluster analysis

Different techniques to carry out cluster analysis usually involve selecting an appropriate unsupervised algorithm based on the type of data and the purpose of clustering [54]. Indeed, it is recommended to use more than one technique and compare the results, taking the goodness of different solutions into account. The two main clustering approaches used for card sorting are data partitioning and hierarchical clustering.

K-means is probably the most widely utilized method for data partitioning. It is based on creating clusters by minimizing the average square distances among observations and finding centroids to group objects around them. K-means requires a k-value to generate the corresponding number of clusters. The standard and most commonly used k-means approach is Hartigan-Wong [54], which is based on the sum of squared Euclidean distances between items and the corresponding centroid to define the within-cluster variations. However, other approaches exist, such as k-medoid, which is based on minimizing the average dissimilarity of objects. This is an alternative to the standardization of the variables, and

Table 5 Recommended configuration for clustering in card sorting depending on each type of variable

Variable Type	Matrices Needed	Distance metric	Clustering Configuration
Interval-Scaled	Z-scores should be utilized: $\mathrm{M}_{\mathrm{ccat}}$ matrix for clustering cards or $\mathrm{M}_{\mathrm{ccat}}$ matrix for clustering categories	Euclidean	k-means (Hartigan-Wong's approach), with the selected *k* value for partitioning, and Ward linkage criterion for dendrograms
Symmetric Binary	$\mathrm{M}_{\mathrm{ccat}}$ or $\mathrm{M}_{\mathrm{catc}}$ for clustering cards or categories, respectively	Euclidean	k-means (Hartigan-Wong's approach), with the selected *k* value for partitioning, and Ward linkage criterion for dendrograms
Asymmetric Binary	$\mathrm{M}_{\mathrm{ccat}}$ or $\mathrm{M}_{\mathrm{catc}}$ for clustering cards or categories, respectively	–	Monothetic clustering algorithms

results might be more robust than those based on k-means due to the sensitivity to outliers that the latter techniques have [54]. For card sorting, k-means usually works well when clustering cards, and can be used to compare categories and reduce them in topic normalization [12].

Based on grouping observations as a hierarchy of clusters, hierarchical clustering mainly utilizes one of the two approaches: agglomerative or divisive [63]. Agglomerative approaches entail bottom-up algorithms where observations start in individual clusters and are combined as the algorithm moves up. By contrast, divisive algorithms are based on a top-down strategy, where there is an initial big cluster including all the observations, and this cluster is recursively split up as the algorithm moves down in the hierarchy. Hierarchical agglomerative clustering (HAC) is the most common approach. HAC does not require an initial value for *k*. Instead, it builds on the relationships among individual items, producing trees where each item is classified accordingly. The output is an agglomerative tree called a dendrogram. Different linkage criteria can be used when determining the tree, such as *single-linkage*, *complete-linkage*, *centroid-method*, Ward, etc. Previous studies have compared various HAC linkage criteria [17], and Ward's linkage criterion behaved better than others, being more sensitive to outliers and data bias. Thus, Ward is the most frequently used linkage for card sorting. It is worth mentioning that the dendrogram produced may not always fit the expected conceptual division due to the artificiality of the taxonomies involved [59]. Thus, keeping track of the process, interpreting and adjusting the information according to the domain, and the clustering purpose, is needed.

Other approaches exist for treating binary data specifically, for example, monothetic clustering that is based on analyzing only one variable at a time. A single-variable analysis, which can be particularly useful for asymmetric binary data, is commonly applied. However, in card-sorting analyses, all variables are often analyzed simultaneously, thus, a polythetic approach is more common to figure out the card or categories clusters based on all available observations.

The summary of clues discussed above is shown in Table 5. Depending on the types of variables, the table indicates the recommended configuration to carry out clustering strategies. For example, k-means, with Euclidean distances, work well for interval-scaled data, and also for symmetric binary data and Ward linkage for dendrograms is recommended. However, as explained in the previous section, the normalization of variables is necessary for interval-scaled card-sorting data to avoid biases due to data aggregation.

For asymmetric binary data, Jaccard distances might be used and then the previously commented clustering configurations. However, it is often advisable to utilize monothetic algorithms such as DIVCLUS-T [69] or MONA [54], using the asymmetric binary matrix directly as input.

The approaches discussed in the previous section can be used to evaluate and validate the clustering solutions. In general, a good cluster solution implies that between-cluster dissimilarities become much larger than the within-cluster ones [54]. The total within-cluster sum of squares indicates the compactness of the cluster and should be as smaller as possible (i.e., goodness indicator). Average silhouette can be applied to both k-means and HAC to analyze the goodness of clustering obtained for a certain *k*. Also, for HAC, the cophenetic correlation coefficient (CCC) can be used to analyze the goodness of a dendrogram solution [17]. CCC can be defined as the linear correlation between the dissimilarities of each pair of observations and their corresponding cophenetic distances. The cophenetic distance is an intergroup dissimilarity measure of two objects that were merged in the same cluster. This method has been used in biostatistics for a long time to measure how faithfully a dendrogram preserves the pairwise distances among the original unmodeled data points.

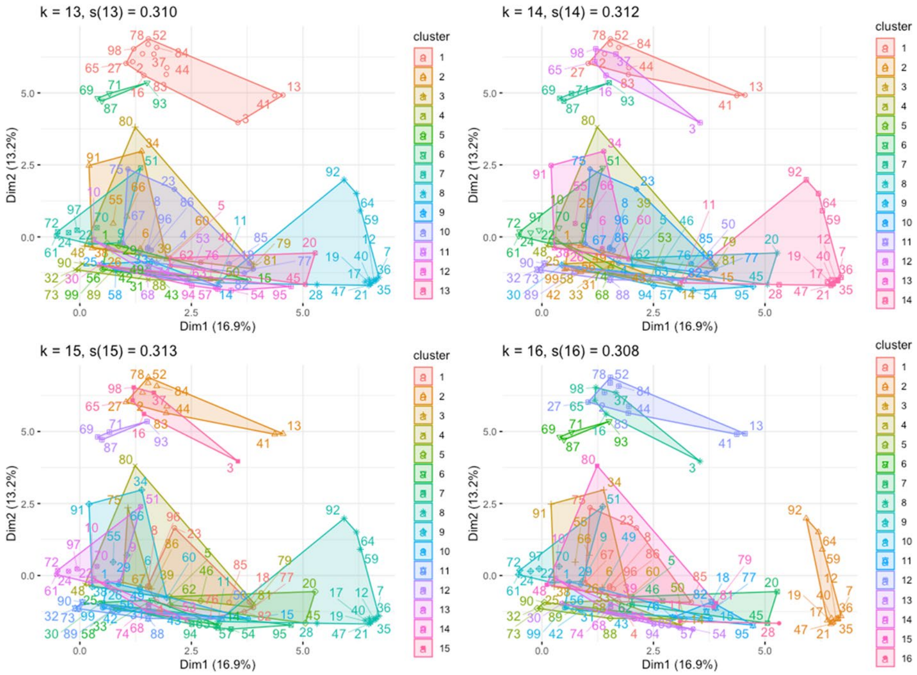

Fig. 12 PCA-based representation of DS1 k-means for different k values: 13 (top-left), 14 (top-right), 15 (bottom-left) and 15 (bottom-right), together with their corresponding $\bar{s}(k)$ scores for each k. Card names have been transformed into numbers for the sake of the clarity of visualization

It can be used to analyze whether a dendrogram represents an appropriate solution or not. Furthermore, a high correlation between the original distances and the cophenetic ones indicates high goodness for a given dendrogram, whereas a low value indicates that the dendrogram only represents a description of the output of the clustering algorithm.

For both DS1 and DS2 datasets, according to the parameters and configurations discussed in previous sections, k-means based on Hertingan-Wong's approach and Euclidean distances are good choices to use. Furthermore, both HAC with Ward's linkage can be utilized (with z-scores for DS1 to minimize the effect of aggregated data). M_{ccat} matrices have been used for both datasets since the idea was to analyze the clustering of observations (cards). In general, graphical representations for k-means results are quite complex when a high number of variables are involved. However, some approaches allow k-means to be also used in such cases. One of the most frequently used ones is the principal component analysis (PCA) approach mentioned in Sect. 2, where clusters are graphically depicted in two dimensions representing the two principal components that explain the majority of the variance [70]. Also, another possibility is to plot the MDS configuration and mark clusters calculated with k-means.

Figure 12 shows a PCA-based representation of k-means for DS1, including different k values and their corresponding $\bar{s}(k)$ scores for each k ranging from 13 to 16. As shown, $k = 15$ represents the optimal solution based on the average silhouette ($\bar{s}(15) = 0.313$), as discussed in the previous section. As Fig. 12 shows, there has been a transformation of clusters from $k = 13$ to $k = 15$. For example, in $k = 13$ case, cluster 1 (in light red, containing

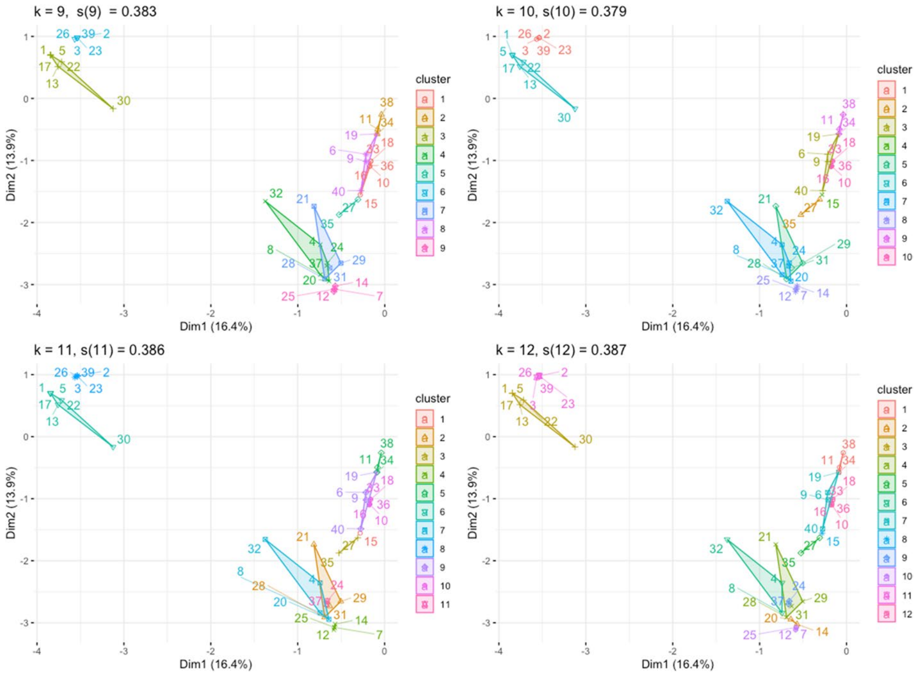

Fig. 13 PCA-based representation of DS2 k-means for different k values: 9 (top-left), 10 (top-right), 11 (bottom-left) and 12 (bottom-right), together with their corresponding $\overline{s}(k)$ scores for each k. Card names have been transformed into numbers for the sake of the clarity of visualization

14 cards, 3, 41, 13, etc.) was split up into two clusters: cluster 2 and cluster 15. These clusters appear on the bottom-left chart of Fig. 12, k = 15, where some conceptual groupings can be identified. For instance, while cluster 15 contains 5 cards representing *papers* that discuss *facet* issues, cluster 2 contains 9 cards representing *papers* that report on *taxonomies* and *controlled vocabularies.*

Similarly, Fig. 13 shows a PCA-based representation of k-means for DS2, including different k values and their corresponding $\overline{s}(k)$ scores for k ranging from 9 to 12. As shown, k = 12 represents the optimal solution based on the average silhouette ($\overline{s}(12)$ = 0.387). However, as discussed in the previous section, k = 11 can also be considered as a successful clustering. Figure 13 shows that there has been a transformation of clusters from k = 9 to k = 12, where for k = 9, cluster 4 (containing 6 cards: *bread*, *cereal*, *muffin*, *pancakes*, *rice,* and *waffle*) was split up into three different clusters: cluster 9 (*pancakes*, and *waffle*), cluster 2 (*doughnuts* and *muffin*) and cluster 6 (*bread*, *cereal,* and *rice*), shown in the figure for k = 12. In addition, on Fig. 13 (k = 12), some conceptual groupings appear. For instance, clusters 3 and 11, located on the top-left, represent *fruits* and *vegetables*, respectively. Cluster 11 contains cards such as 2 (*apple*), 3 (*banana*), 23 (*orange*), 26 (*pineapple*), and 39 (*watermelon*). On the other hand, cluster 3 includes cards such as 1 (*carrots*), 5 (*broccoli*), 13 (*corn*), 17 (*lettuce*), 22 (*onions*), and 30 (*potatoes*).

Figure 14 depicts the agglomerative dendrogram for dataset DS1. The dendrogram shows how similar cards were combined into branches, with the *height* of the tree growing along the x-axis. The *height* value is useful for analyzing how similar or

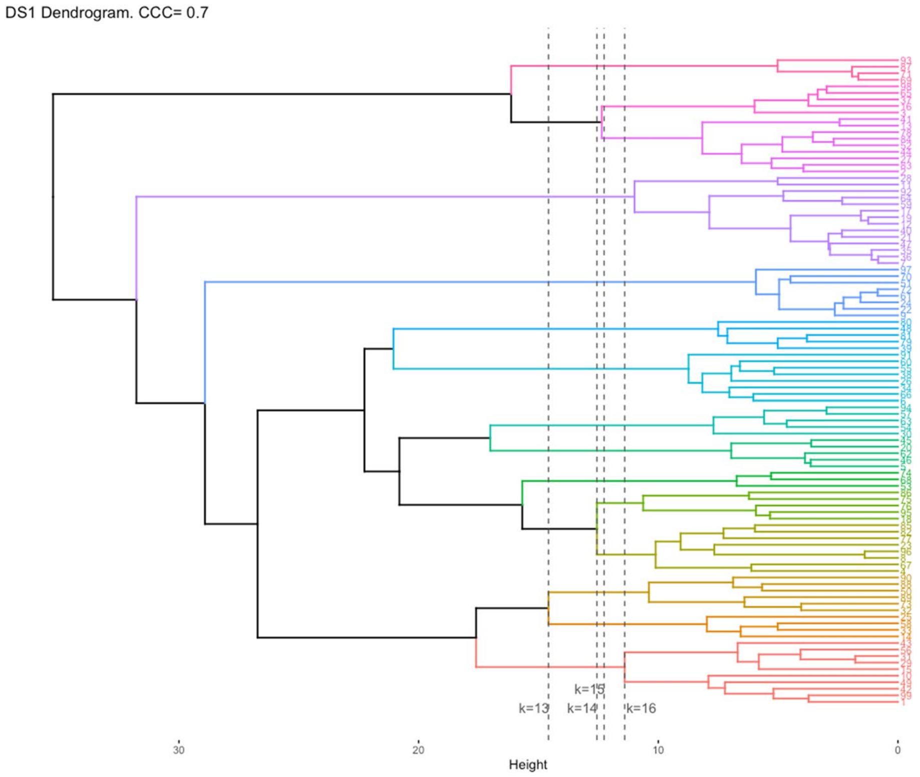

Fig. 14 Dendrogram based on cards clustering for DS1. Different $k = 13$, 14, 15 and 16 are represented as vertical dotted lines. The overall CCC is also provided. Card names have been transformed into numbers for the sake of the clarity of visualization

not two observations are and finding the point of fusion of the two. Specifically, the higher the *height* of the fusion, the less similar observations. However, this parameter also provides clues about the number of clusters that might be obtained at a certain *height* (called *height* cut). As indicated in Fig. 14, a CCC value of 0.7 was obtained, ensuring an acceptable goodness value for DS1 dendrogram. The intersections with branches are shown for different k values (13, 14, 15, and 16), similar to the previously analyzed k-means solutions. In this case, colors specify the configuration for $k = 15$. As can be seen, $k = 14$ and $k = 15$ represent similar outcomes; nevertheless, $k = 15$ might represent a better conceptual clustering since it facilitates the split of cluster 2 into two different conceptual ones (i.e., a cluster more related to *facets* and another one more related to *taxonomies* and *controlled vocabularies*), as discussed previously. For DS2, shown in Fig. 15, a value of 0.83 was obtained for CCC, ensuring an acceptable goodness value for this dendrogram. Figure 15 shows different k values (9–12) to indicate intersections with branches, like with DS1. As shown, $k = 11$ and $k = 12$ represent similar outcomes, but $k = 11$ may represent a better conceptual solution as it avoids the split of the *bakery*-related cards (i.e., *muffin*, *doughnuts*, *cookies*, *pie*, and *cake*) into two different branches.

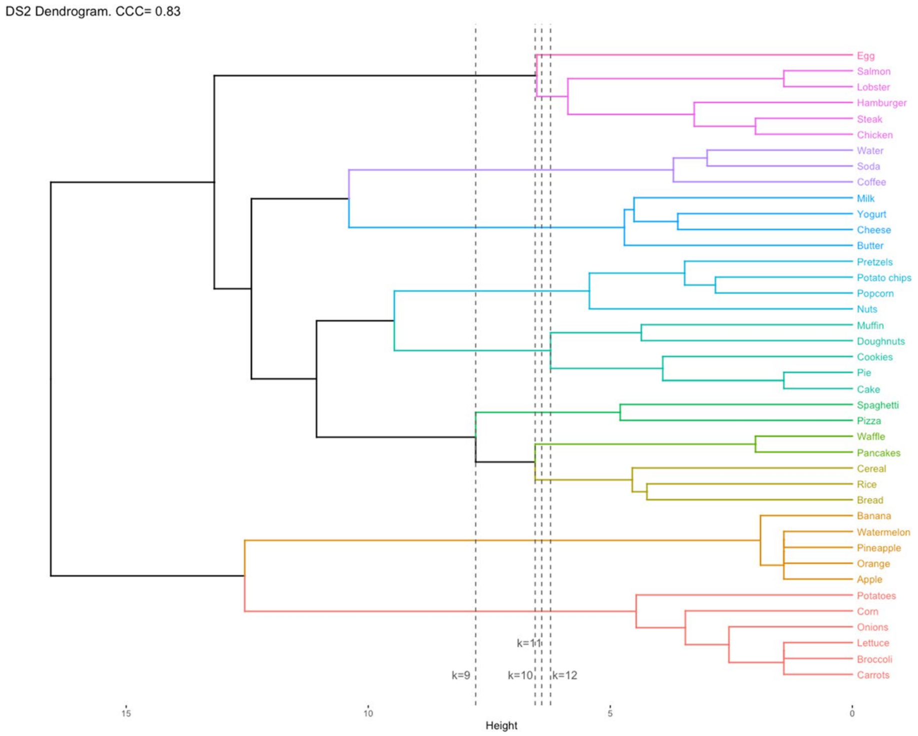

Fig. 15 Dendrogram based on cards clustering for DS2. Different $k=9$, 10, 11 and 12 are represented as vertical dotted lines. The overall CCC is also provided

The techniques presented in this sub-section describe how to carry out a multivariate analysis, using different statistics and goodness indicators to identify structures and relations of different card or category variables. Here too, not all the discussed techniques are strictly necessary for card-sorting analysis. The selection criteria for the techniques can be the following:

- *Essential*: MDS configuration is necessary to find the best configuration for the multidimensional scaling. The information shown in Table 4 has to be considered.
- *Essential*: Smacof MDS configuration to identify structural relations between categories. This identifies conceptual category groupings. Visualizations, like the one described in Fig. 7, should be used.
- *Recommended*: *Stress* might be computed, as well as the Shepard diagram and *Stress-Per-Point* charts (Figs. 8, 9) that are used as goodness indicators, in particular when further analysis is desired.
- *Recommended*: Optimal number of clusters for k-means and HAC might be desirable. This can be used to determine an exact number of clusters, instead of tying with different values to see how they fit, which is essential in HAC (dendrograms) representations, as they are more difficult to interpret without a proper *k* value.

- *Recommended*: Average silhouette for different *k* values should be determined. This helps to find a more accurate number of clusters. Visualizations, like the ones that shown in Figs. 10, 11, are useful for such purpose.
- *Essential*: Representation of the k-means solution to identify clusters of cards. As explained, MDS is suitable to analyze the topology of the data and establish conceptual relationships among items. However, most often, it is necessary to obtain smaller groups to bring to light small or medium size relationships among the data. To this end, k-means can be also used to compare both approaches and find other kinds of relations. Also, it is useful for comparing categories to reduce them (topic normalization). The information shown in Table 5 has to be considered for a better configuration. Also, the graphics shown in Figs. 12, 13 are convenient representations for clusters using k-means and PCA, which allows to graphically depict the clusters in two dimensions representing the two principal components that explain the majority of the variance, helping observe other kinds of clusters hard to find in the MDS approach.
- *Recommended*: HAC solution should be found to identify clusters of cards, and the corresponding dendrograms determined. Although convenient, dendrograms should be complemented with other representations, such as MDS and k-means, where clusters can be visually identified, providing the immediate feedback. In addition, dendrograms results are hard to interpret without a specific value of *k*, thus, evaluators should consider the *height* parameter to identify different clusters. If the number of items is high, the visualizations are even more complex. Figures 14, 15 can be considered as examples to follow in terms of suitable HAC representations of card-sorting results.
- *Optional*: CCC can be determined as a goodness indicator. It can be used with HAC to analyze the goodness of a dendrogram solution, if desired.

4 Discussion

The approach proposed in this paper comprises a set of steps for the systematic quantitative analysis of data obtained from card sorting. As suggested in the introduction, quantitative card sorting seldom realizes its full potential because it is often too complex for those not experienced in statistical analysis and visualization methods. As demonstrated through the analyses of datasets DS1 and DS2, a great variety of clues, statistics, and visual representations of data should be examined if decision-making processes are to be adequately supported in quantitative analysis of card sorts.

We have strived to include the best choices of methods and techniques relevant to these datasets and discuss their implications. As mentioned, card sorting often produces similar data to the ones in sets discussed. Thus, although a broader range of statistical and data-mining techniques could be used, the reasoning behind choices was carefully presented. The pathway to include other relevant methods and techniques was left open—which is one of the reasons for calling the approach a methodology. This comprehensive approach extends beyond applying prescribed steps in a liner-fashion, offering instead a way to create novel research designs for qualitative card sorting studies while maintaining the rigor in thinking and allowing for justifications and validations of different choices toward optimal analyses results.

One of the intents of this paper was to make the quantitative analysis of card sorting available to a broader range of researchers, including those in disciplines where extensive knowledge of statistical analysis is not necessarily expected (like experience design, service design, or design thinking). The explanations provided in this paper were given to rationalize choices rather than to provide the abundance of technical details, which are available in various articles cited throughout this paper. The visualizations and goodness indicators provided were to help guide researchers through options and possible pitfalls of the analysis.

Still, there were some limitations and threats to validity that need to be addressed.

4.1 Limitations

We highlight two issues concerning the limitations and how they might be resolved or mitigated. The first issue relates to the completeness of the suggested guidelines, clues, statistical methods, techniques, and visualizations used in the paper. While this could be seen as a limitation since not all possible combinations, even for the two datasets discussed, could be included, for the sake of brevity, the most frequently used ones were addressed, and the extent of information should be sufficient to try the approach successfully.

Moreover, a comprehensive analysis should utilize both cards and categories. Thus, instead of the analysis focusing on either cards or categories, we illustrate particular concerns that might arise in situations similar to the ones described. For instance, a specific kind of variable might be more appropriate than others in a particular sort. Then, we aimed to examine the corresponding details of decision-making. However, the analysis of cards and categories presented in this paper utilized only the $\mathrm{M}_{\mathrm{ccat}}$ or the $\mathrm{M}_{\mathrm{catc}}$ matrices. When using such an analysis, evaluators are encouraged to generate a larger number of visualizations, as these would increase opportunities to find evidence that might facilitate optimal decision-making.

Furthermore, the case of asymmetric binary variables has not been illustrated at the same level of detail as the case involving symmetric ones. This is because asymmetric binary datasets are seldom available. However, in the case of less common asymmetric card-sorting data, whether binary or numerically aggregated, the data are usually transformed into symmetric binary or aggregated variables by considering each nested category independently (i.e., like a multiple card-sorting design), where the same card can be classified into more than one individual category (linearizing the hierarchy). Thus, we paid greater attention to the analysis of prevalent, interval-scaled, and symmetric binary variables. However, all the necessary clues have been provided, including the main configurations, proposed techniques, algorithms, and parameters to guide the analysis in the case of asymmetric binary variables.

4.2 Threats to Validity

Regarding threats to validity, because the approach is illustrated by two distinct datasets, only a few external threats to validity might still apply. The chosen datasets have representative sample sizes, as previously justified, in terms of observations, variables, and the number of participants. The presented approach allows for a range of different possibilities and configurations, and attention has been paid to careful elaboration on how decisions were made at each step when applying the methods and techniques to the sample datasets. As card-sorting datasets do not usually vary too much from one another, the same process is readily applicable to other datasets. Thus, different combinations of data, techniques, parameters, and algorithms have been conveniently covered for other datasets also, thus minimizing the replication factor.

Another issue that might occur relates to the possibility that a particular feature of a dataset might be responsible for the effects obtained by the analysis and lead to limited generalizability of the findings. To mitigate this threat, the proposed method suggests the need for data characterization in the initial steps, as discussed in Sect. 3.2 and illustrated in sample datasets DS1 and DS2. Furthermore, recommendations for different data characteristics have been provided, including the normalization, or an alternative, of different data types. In addition, goodness indicators for all the statistical techniques suggested have been provided. Jointly, these features of the approach minimized the referred threat.

In summary, the presented approach produces results that minimize the threats to validity and are generalizable, enabling it to be applied in other domains. Thus, the main research question "*Is it possible to define a systematic method for carrying out a quantitative analysis of card-sorting data that provides instruments and goodness indicators for the decision-making in web application development*?" could be answered in the affirmative.

5 Conclusion

Modern web application development involves ensuring quality issues relating to content, structure, and navigation from a user-centered perspective [71]. To achieve the desired quality, it is crucial to have adequate methods, tools, and techniques that support design and evaluation during the engineering of web-based applications. Card sorting is a frequently used method to collect and analyze information from any domain, and it is helpful for understanding users' mental models concerning how information can be categorized and related. It has been successfully applied, for example, to find the optimal categories of items [13] when developing a web application or when creating menu items for navigating the application.

While card sorting has been applied in different research domains, information architecture is the field in which the method has been the most widely applied to categorize content and to develop navigational structures for web applications [5, 72], thereby ensuring the usability [11, 71] of applications and their overall quality [73, 74].

Since card sorting is easy to design and implement, it has become a common research practice in user experience and design thinking, among other fields [30]. It is one of the most suitable ways to elicit requirements in the early phases of web-based user-centered projects [75, 76], but it can also be used as an evaluation technique to test and refine advanced design solutions.

However, a quantitative analysis of card sorting results is not a simple process. It requires specific knowledge about the best statistical practices and their interpretation if meaningful results are to be obtained that can lead to making good decisions. Most card sorts are still analyzed utilizing custom spreadsheets that contain only the most basic information about the raw data. Admittedly, some commercial tools might cover both the implementation and analysis of card sorting, but most of them produce customized results and visualizations without proper goodness indicators. Especially with large datasets and complex sorts, these results are inadequate. In such cases, statistical packages might be used to provide more advanced visualizations and analyses. However, these are often difficult for unskilled users to interpret and could fail to assist with making appropriate choices [17]. In contrast, a precise understanding of the techniques, parameters, and algorithms implemented could result in complex but necessary steps to ensure the quality of the results [13].

This research proposes a methodology for the quantitative analysis of data obtained from card sorting, based on various statistics and visualizations. Although the statistical methods and techniques presented have been applied previously in other domains, this work brings them together to support better decision-making in the process of the quantitative analysis of card sorts.

The proposed approach utilizes a top-down process and unfolds in four sequential steps to systematically analyze card sorting data, identify key issues, and make informed decisions regarding the next step. In the first step, the raw data is comprehensively analyzed to characterize variables and their types. This step is crucial, as selecting the proper statistical techniques later strongly depends on the card sorting data. Next, specific data models are proposed (the working matrices) as inputs for the subsequent algorithms. Finally, in the further steps, technical details about dissimilarity analysis and multivariate statistics are provided, indicating how to obtain the optimal number of clusters from the data and how to represent the information using meaningful charts and visualizations to facilitate decision-making. Furthermore, several tables containing recommended parameters, configurations, and goodness indicators are provided to systematize the approach as far as possible and make it extensible to other card sorting problems. The entire process is carefully illustrated using two existing and publicly available datasets.

The approach is intended to guide both experienced and non-experienced evaluators through the process. Most of the presented techniques are complementary and not strictly necessary, but might be used for further analysis, depending on the complexity of the card sorting data. Of particular importance is the provision of recommendations and explanations of the most important algorithms and parameters, visualizations and principal goodness indicators to guide optimal choices when decision-making. Jointly, these are intended to make the quantitative analysis of card sorting more accessible to a broad range of evaluators, analysts, and usability engineers, among others. Consequently, the increased usability of the approach should help expand its use in fields such as user experience design, design thinking, and service design. In conclusion, the proposed approach makes it possible to answer the main research question in the affirmative.

Regarding future work, creating a supporting tool to implement this more systematic approach to card-sorting analysis and cover the interactive tasks in card sorting would be an appropriate next step. Such a tool would represent a more holistic solution that would support card sorting and its analysis, possibly including qualitative information from the card sorting sessions to enhance the overall analysis. It would be interesting to find more efficient solutions for extensive datasets than the matrices used in this work to pursue a different line of thought.

Acknowledgements This work was partially supported by the Spanish Government [grant number RTI2018-095255-B-I00]; and the Madrid Research Council [Grant Number P2018/TCS-4314].

Funding Included below, in the Acknowledgements heading.

Data availability Datasets are included as (web) references.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

1. Macías, J.A., Castells, P.: Tailoring dynamic ontology-driven web documents by demonstration. In: Proceedings of the international conference on information visualisation. pp. 535–540 (2002)
2. Macías, J.A., Castells, P.: Interactive design of adaptive courses. In: Computers and education. pp. 235–242. Kluwer Academic Publishers (2001)
3. Macías, J.A., Castells, P.: A generic presentation modeling system for adaptive web-based instructional applications. In: Conference on human factors in computing systems—proceedings. pp. 349–350 (2001)
4. Macías, J.A., Castells, P.: Adaptive hypermedia presentation modeling for domain ontologies. In: Proceedings of 10th International conference on human-computer interaction. In proceedings of 10th International conference on human-computer interaction (HCII'2001). New Orleans, Louisiana. (2001)
5. Macías, J.A.: Intelligent assistance in authoring dynamically generated web interfaces. World Wide Web **11**, 253–286 (2008). https://doi.org/10.1007/s11280-008-0043-3
6. Keller, M., Nussbaumer, M.: MenuMiner: Revealing the information architecture of large web sites by analyzing maximal cliques. In: WWW'12— Proceedings of the 21st Annual Conference on World Wide Web Companion. pp. 1025–1034 (2012)
7. Chinthakayala, K.C., Zhao, C., Kong, J., Zhang, K.: A comparative study of three social networking websites. World Wide Web **17**, 1233–1259 (2014). https://doi.org/10.1007/s11280-013-0222-8
8. Yuliang, W., Qi, Z., Fang, L., Xixian, H., Guodong, X., Bailing, W.: A novel approach for Web page modeling in personal information extraction. World Wide Web **22**, 603–620 (2019). https://doi.org/10.1007/s11280-018-0631-9
9. Rosenfeld, L., Morville, P.: Information Architecture for the World Wide Web, 3rd Edition - O'Reilly Media. (2001)
10. Castells, P., Macías, J.A.: Un sistema de presentación dinámica hipermedia para representaciones personalizadas del conocimiento. Intel. Artif. (2002). https://doi.org/10.4114/ia.v6i16.738
11. Cayola, L., Macías, J.A.: Systematic guidance on usability methods in user-centered software development. Inf. Softw. Technol. **97**, 163–175 (2018). https://doi.org/10.1016/j.infsof.2018.01.010
12. Paul, C.L.: Analyzing card-sorting data using graph visualization FLOSS usability view project IEEE VAST challenge view project. J. usability Stud. **9**, 87–104 (2014)
13. Spencer, D.: Card sorting: designing usable categories. Rosenfeld Media (2009)
14. Macías, J.A.: Enhancing interaction design on the semantic web: A case study. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **42**, 1365–1373 (2012)
15. Righi, C., James, J., Beasley, M., Day, D., Fox, J., Gieber, J., Howe, C., Ruby, L.: Card sort analysis best practices. J. Usability Stud. **8**, 69–89 (2013)
16. Paea, S., Baird, R.: Information Architecture (IA): using multidimensional scaling (MDS) and k-means clustering algorithm for analysis of card sorting data. J. Usability Stud. **13**, 138–157 (2018)
17. Saraçli, S., Doğan, N., Doğan, I.: Comparison of hierarchical cluster analysis methods by cophenetic correlation. J. Inequalities Appl. (2013). https://doi.org/10.1186/1029-242X-2013-203
18. Rosenberg, S., Nelson, C., Vivekananthan, P.S.: A multidimensional approach to the structure of personality impressions. J. Pers. Soc. Psychol. (1968). https://doi.org/10.1037/h0026086
19. Jastrow, J.: A sorting apparatus for the study of reaction-times. Psychol. Rev. **5**, 279–285 (1898). https://doi.org/10.1037/h0073343
20. Kline, L.W., Kellogg, C.E.: Cards as psychological apparatus. Science **39**, 657–659 (1914). https://doi.org/10.1126/science.39.1009.657
21. Shulman, C., Yirmiya, N., Greenbaum, C.W.: From categorization to classification: a comparison among individuals with autism, mental retardation, and normal development. J. Abnorm. Psychol. **104**, 601–609 (1995). https://doi.org/10.1037/0021-843X.104.4.601
22. Coxon, A.M.: Sorting Data Collection and Analysis. In: Sage University Series in Quantitative Application in the Social Science. p. 98 (1999)
23. Rao, V.R., Katz, R.: Alternative multidimensional scaling methods for large stimulus sets. J. Mark. Res. **8**, 488–494 (1971). https://doi.org/10.1177/002224377100800413

24. Bijmolt, T.H.A., Wedel, M.: The effects of alternative methods of collecting similarity data for multidimensional scaling. Int. J. Res. Mark. **12**, 363–371 (1995). https://doi.org/10.1016/0167-8116(95)00012-7
25. Feine, J., Gnewuch, U., Morana, S., Maedche, A.: A taxonomy of social cues for conversational agents. Int. J. Hum. Comput. Stud. **132**, 138–161 (2019). https://doi.org/10.1016/j.ijhcs.2019.07.009
26. Hudson, W.: Card sorting. The encyclopedia of human-computer interaction. Obtained from https//www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/card-sorting. (2014)
27. Blanchard, S.J., Banerji, I.: Evidence-based recommendations for designing free-sorting experiments. Behav. Res. Methods. **48**, 1318–1336 (2016). https://doi.org/10.3758/s13428-015-0644-6
28. Nielsen, J.: Card sorting to discover the users model of the information space. Obtained from https//www.nngroup.com/articles/usability-testing-1995-sun-microsystems-website/. (1995)
29. Robles, T. de J.Á., Rodríguez, F.J.Á., Benítez-Guerrero, E., Rusu, C.: Adapting card sorting for blind people: Evaluation of the interaction design in TalkBack. Comput. Stand. Interfaces. (2019). https://doi.org/10.1016/j.csi.2019.103356
30. Brown, T.: Design thinking. Harv. Bus. Rev. 86, (2008)
31. Culén, A.L., Gasparini, A.A.: Design thinking processes: card methodologies for non-designerse. In: Minaříková, P. and Suchá, L.Z. (eds.) Librarians as Designers. Case studies on the improvment of library services. pp. 73–85. Masarykova Univerzita (2016)
32. Polaine, A., Lovlie, L., Reason, B.: Service design from insight to implementation. Rosenfeld Media (2013)
33. Osterwalder, A., Pigneur, Y., Bernarda, G., Smith, A.: Value proposition design: how to create products and services customers want. Wiley (2015)
34. Culén, A.L., van der Velden, M.: Making context specific card sets—a visual methodology approach capturing user experiences with urban public transportation. Int. J. Adv. Intell. Syst. **8**, 17–26 (2015)
35. Clatworthy, S.: Service innovation through touch-points: development of an innovation toolkit for the first stages of new service development. Int. J. Des. **5**, 15–28 (2011)
36. Culén, A.L., Gasparini, A.A.: Find a book! Unpacking customer journeys at academic library. In: ACHI 2014— 7th International Conference on Advances in Computer-Human Interactions. pp. 89–95 (2014)
37. Aarts, T., Gabrielaitis, L.K., De Jong, L.C., Noortman, R., Van Zoelen, E.M., Kotea, S., Cazacu, S., Lock, L.L., Markopoulos, P.: Design card sets: Systematic literature survey and card sorting study. In: DIS 2020—Proceedings of the 2020 ACM Designing Interactive Systems Conference. pp. 419–428. Association for Computing Machinery, Inc (2020)
38. Veral, R., Macías, J.A.: Supporting user-perceived usability benchmarking through a developed quantitative metric. Int. J. Hum. Comput. Stud. **122**, 184–195 (2019). https://doi.org/10.1016/j.ijhcs.2018.09.012
39. Henriques, D.P., Dalton, R., Greenhalgh, P.: Measuring the impact of future visions through card sorting. In: Urban living labs for public space—a new generation of planning? (2017)
40. Blanchard, S.J., Aloise, D., DeSarbo, W.S.: The heterogeneous p-median problem for categorization based clustering. Psychometrika **77**, 741–762 (2012). https://doi.org/10.1007/s11336-012-9283-3
41. Capra, M.G.: Factor analysis of card sort data: an alternative to hierarchical cluster analysis. In: Proceedings of the human factors and ergonomics society. pp. 691–695 (2005)
42. Cardsorting.net: Cardsorting.net. http://cardsorting.net
43. Syntagm: Design for usability. http://www.syntagm.co.uk/design/index.shtml
44. XSort: Free card sorting application for Mac. https://xsortapp.com
45. UserZoom: UserZoom. https://www.userzoom.com/es/
46. ProvenByUsers: Online card sorting from proven by users. https://www.provenbyusers.com/
47. usabiliTEST: usabiliTEST: Usability testing tools for everyone. http://www.usabilitest.com/
48. Optimal Workshop: Optimal Workshop. https://www.optimalworkshop.com/
49. Lantz, E., Keeley, J.W., Roberts, M.C., Medina-Mora, M.E., Sharan, P., Reed, G.M.: Card sorting data collection methodology: how many participants is most efficient? J. Classif. **36**, 649–658 (2019). https://doi.org/10.1007/s00357-018-9292-8
50. Tullis, T., Investments, F., Wood, L., Young, B.: How Many Users Are Enough for a Card-Sorting Study? The Card-sorting Study. Proc. UPA. 0, 1–9 (2004)
51. Pawliczek, P., Dzwinel, W.: Interactive data mining by using multidimensional scaling. In: Procedia Computer Science. pp. 40–49 (2013)
52. Spencer, D.: Dataset based on paper submitted to IA Summit. https://rosenfeldmedia.com/books/card-sorting/details/resources/

53. Cardsorting.net: Dataset based on the classification of food ítems. http://cardsorting.net/tutorials/sumpi.html
54. Kaufman, L., Rousseeuw, P.J.: Finding groups in data. Wiley (2005)
55. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis. Chemom. Intell. Lab. Syst. **2**, 37–52 (1987). https://doi.org/10.1016/0169-7439(87)80084-9
56. Maxwell, A.E., Harman, H.H.: Modern factor analysis. J. R. Stat. Soc. Ser. A. **131**, 615 (1968). https://doi.org/10.2307/2343736
57. Borg, I., Groenen, P.J.F., Mair, P.: Applied multidimensional scaling and unfolding. Springer International Publishing, Cham (2018)
58. Whaley, A., Longoria, R.: Preparing card sort data for multidimensional scaling analysis in social psychological research: a methodological approach. J. Soc. Psychol. **149**, 105–115 (2009). https://doi.org/10.3200/SOCP.149.1.105-115
59. Hinkle, V.: Card-sorting: what you need to know about analyzing and interpreting card sorting results. Usability News. **10**, 1–6 (2008)
60. Romesburg, C.: Part I. Overview of cluster analysis. In: Cluster analysis for researchers. p. 334 (2004)
61. Baker, F.B., Hubert, L.J.: Measuring the power of hierarchical cluster analysis. J. Am. Stat. Assoc. **70**, 31–38 (1975). https://doi.org/10.1080/01621459.1975.10480256
62. Hartigan, J.A., Wong, M.A.: Algorithm AS 136: a K-means clustering algorithm. Appl. Stat. **28**, 100 (1979). https://doi.org/10.2307/2346830
63. Macías, J.A.: Enhancing card sorting dendrograms through the holistic analysis of distance methods and linkage criteria. J. Usability Stud. **16**, 73–90 (2021)
64. Busing, F., Commandeur, J.J.F., Heiser, W.J.: PROXSCAL: a multidimensional scaling program for individual differences scaling with constraints. Softstat. **97**, 67–74 (1997)
65. Roberts, M.C., Reed, G.M., Medina-Mora, M.E., Keeley, J.W., Sharan, P., Johnson, D.K., Mari, J.D.J., Ayuso-Mateos, J.L., Gureje, O., Xiao, Z., Maruta, T., Khoury, B., Robles, R., Saxena, S.: A global clinicians' map of mental disorders to improve ICD-11: Analysing meta-structure to enhance clinical utility. Int. Rev. Psychiatry. **24**, 578–590 (2012). https://doi.org/10.3109/09540261.2012.736368
66. Mair, P., Borg, I., Rusch, T.: Goodness-of-fit assessment in multidimensional scaling and unfolding. Multivariate Behav. Res. **51**, 772–789 (2016). https://doi.org/10.1080/00273171.2016.1235966
67. Van Der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**, 2579–2625 (2008)
68. Charrad, M., Ghazzali, N., Boiteau, V., Niknafs, A.: Nbclust: An R package for determining the relevant number of clusters in a data set. J. Stat. Softw. **61**, 1–36 (2014)
69. Chavent, M., Lechevallier, Y., Briant, O.: DIVCLUS-T: A monothetic divisive hierarchical clustering method. Comput. Stat. Data Anal. **52**, 687–701 (2007). https://doi.org/10.1016/j.csda.2007.03.013
70. Alboukadel, Kassambara, Fabian, M.: Factoextra: Extract and visualize the results of multivariate data analyses. R package. R Packag. version. 1, (2019)
71. Quintal, C., Macías, J.A.: Measuring and improving the quality of development processes based on usability and accessibility. Univers. Access Inf. Soc. **20**, 3 (2021). https://doi.org/10.1007/s10209-020-00726-7
72. Macías, J.A., Granollers, T., Andrérs, P.L.: New trends on human-computer interaction: Research, development, new tools and methods. (2009)
73. Seckler, M., Heinz, S., Forde, S., Tuch, A.N., Opwis, K.: Trust and distrust on the web: User experiences and website characteristics. Comput. Human Behav. **45**, 39–50 (2015). https://doi.org/10.1016/j.chb.2014.11.064
74. Biduski, D., Bellei, E.A., Rodriguez, J.P.M., Zaina, L.A.M., De Marchi, A.C.B.: Assessing long-term user experience on a mobile health application through an in-app embedded conversation-based questionnaire. Comput. Human Behav. (2020). https://doi.org/10.1016/j.chb.2019.106169
75. Rojas, L.A., Macías, J.A.: Toward collisions produced in requirements rankings: A qualitative approach and experimental study. J. Syst. Softw. (2019). https://doi.org/10.1016/j.jss.2019.110417
76. Borges, C.R., Macías, J.A.: Feasible database querying using a visual end-user approach. In: Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems—EICS'10. p. 187. ACM Press, New York, (2010)