Identifying the steps in an exploratory data analysis: a process-oriented approach

Seppe Van Daele[1] and Gert Janssenswillen[1][0000−0002−7474−2088]

UHasselt - Hasselt University, Faculty of Business Economics
Agoralaan, 3590 Diepenbeek, Belgium
gert.janssenswillen@uhasselt.be

Abstract. Best practices in (teaching) data literacy, specifically Exploratory Data Analysis, remain an area of tacit knowledge until this day. However, with the increase in the amount of data and its importance in organisations, analysing data is becoming a much-needed skill in today's society. Within this paper, we describe an empirical experiment that was used to examine the steps taken during an exploratory data analysis, and the order in which these actions were taken. Twenty actions were identified. Participants followed a rather iterative process of working step by step towards the solution. In terms of the practices of novice and advanced data analysts, few relevant differences were yet discovered.

Keywords: Process Mining · Deliberate Practice · Learning Analytics

1 Introduction

Data is sometimes called the new gold, but is much better compared to gold-rich soil. As with gold mining, several steps are needed to go through in order to get to the true value. With the amount and importance of data in nearly every industry [14, 13, 15], data analysis is a vital skill in the current job market, not limited to profiles such as data scientists or machine learning engineers, but equally important for marketing analysts, business controllers, as well as sport coaches, among others.

However, best practices in data literacy, and how to develop them, mainly remains an area of tacit knowledge until this day, specifically in the area of Exploratory Data Analysis (EDA). EDA is an important part in the data analysis process where interaction between the analyst and the data is high [3]. While there are guidelines on how the process of data analysis can best be carried out [15, 18, 21], these steps typically describe what needs to be done at a relatively high level, and do not precisely tell how best to perform them in an actionable manner. Which specific steps take place during an exploratory data analysis, and how they are structured in an analysis has not been investigated.

The goal of this paper is to refine the steps underlying exploratory data analysis beyond high-level categorisations such as transforming, visualising, and modelling. In addition, we analyse the order in which these actions are performed. The results of this paper form a first step towards better understanding

the detailed steps in a data analysis, which can be used in future research to analyse difference between novices and experts in data analysis, and create better data analysis teaching methods focussed on removing these differences.

The next section will discuss related work, while Section 3 will discuss the methodology used. The identified steps are described in the subsequent section, while an analysis of the recorded data is provided in Section 5. Section 6 concludes the paper.

2 Related work

A number of high-level tasks to be followed while performing a data analysis have already been defined in the literature [15, 18], which can be synthesised as 1) the collection of data, 2) processing of data, 3) cleaning of data, 4) exploratory data analysis, 5) predictive data analysis, and 6) communicating the results. In [21] this process is elaborated in more detail, applied to the `R` language. Here the process starts with importing data and cleaning. The actual data analysis is subsequently composed of the cycle of transforming, visualising and modelling data, and is thus slightly more concrete than the theoretical exploratory and prescriptive data analysis. The concluding communication step is similar to [15, 18].

That the different steps performed in a data analysis have received little attention, has also been recognised by [23], specifically focused on process analysis. In this paper, an empirical study has been done to understand how process analysts follow different patterns in analysing process data, and have different strategies to explore event data. Subsequent research has shown that such analysis can lead to the identification of challenges to improve best practices [24].

Breaking down a given action into smaller steps can reduce cognitive load when performing the action [20]. Cognitive load is the load that occurs when processing information. The more complex this information is, the higher the cognitive load is. Excessive cognitive load can overload working memory and thus slow down the learning process. Creating an instruction manual addresses The Isolated Elements Effect [4], when there is a reduction in cognitive load by isolating steps, and only then looking at the bigger picture [20]. In [5], this theory was applied using *The Structured Process Modeling Theory*, to reduce the cognitive load when creating a process model. Participants who followed structured steps, thus reducing their cognitive load, generally made fewer syntax errors and created better process models [5]. Similarly, in [10], participants were asked to build an event log, where the test group was provided with the event log building guide from [11]. The results showed that the event logs built by the test group outperformed those of the control group in several areas.

An additional benefit of identifying smaller steps is that these steps can be used in the creation of a *deliberate practice* — a training course that meets the following conditions [1, 6] :

1. Tasks with a defined objective
2. Immediate feedback on the task created
3. Opportunity to repeat this task multiple times

4. Motivation to actually get better

Karl Ericsson [6] studied what the training of experts in different fields had in common [2], from which the concept of deliberate practice emerged. It was already successfully applied, for example, in [7] where a physics course, reworked to deliberate practise principles, resulted in higher attendance and better grades.

In addition to studying what kind of training experts use to acquire their expertise, it has also been studied why experts are better at a particular field than others. In [6], it is concluded that experts have more sophisticated mental representations that enable them to make better and/or faster decisions. Mental representations are internal models about certain information that become more refined with training [6]. Identifying actions taken in a data analysis can help in mapping mental representations of data analysis experts. This can be done by comparing the behaviour of experts with that of beginners. Knowing why an expert performs a certain action at a certain point can have a positive effect on the development of beginners' mental models. In fact, using mental representations of experts was considered in [19] as a crucial first step in designing new teaching methods.

3 Methodology

In order to analyse the different steps performed during an exploratory analysis, and typical flows between them, an experiment was conducted. The experiments and further data processing and analysis steps are described below.

Experiment Cognitive Task Analysis (CTA) [22] was used as overall methodology for conducting the experiment described in this paper, with the aim to uncover (hidden) steps in a participant's process of exploratory data analysis. Participants were asked to make some simple analyses using supplied data and to make a screen recording of this process. The tasks concerned analysing the distribution of variables, the relationship between variables, as well as calculating certain statistics.

As certain steps can be taken for granted due to developed automatisms [8], the actual analysis was followed by an interview, in which the participants were asked to explain step by step what decisions and actions were taken. By having the interview take place after the data analysis, interference with the participants' usual way of working was avoided. For example, asking questions before or during the data analysis could have caused participants to hesitate, slow down, or even make different choices.

The general structure of the experiment was as follows:

1. **Participants:** The participans for this experiment were invited by mail from three groups with different levels of experience: undergraduate students, graduate students, and PhD students, from the degree Business and Information systems engineering. These students received an introductory course on data analysis in their first bachelor year, where they work with the language R, which was subsequently chosen as the language to be used in the experiment.

In the end, 11 students were convinced to participate in this experiment: two undergraduate students, four graduate students and 5 PhD students. The 11 participants each performed the complete analysis of three assignments, and thus results from 33 assignments were collected.
While having participants with different levels of experience is expected to result in a broader variety in terms of behaviour, the scale of the experiment and the use of student participants only will not allow a detailed analysis of the relationship between experience-level and analysis behaviour. Furthermore, disregarding the different level of students, the once accepting the invitation to participate might also be the more confident about their skills.

2. **Survey:** Before participants began the data analysis, they were asked to complete an introductory survey to gain insight into their own perceptions of their data analysis skill (in R).
3. **Assignment:** The exploratory analysis was done in the R programming language, and consisted of three independent tasks about data from a housing market: 2 involving data visualisation and 1 specific quantitative question. The analysis was recorded by the participants.
4. **Interview:** The recording of the assignment was used during the interview to find out what steps, according to the participants themselves, were taken. Participants were asked to actively tell what actions were taken and why.

Transcription The transcription of the interviews was done manually. Because most participants actively narrated the actions taken, a question-answer structure was not chosen. If a question was still asked, it was placed in italics between two dashes when transcribed.

Coding and categorization To code the transcripts of the interviews, a combination of descriptive and process coding was used in the first iteration. Descriptive coding looks for nouns that capture the content of the sentence [16]. Process coding, in turn, attempts to capture actions by encoding primarily action-oriented words (verbs) [16]. These coding techniques were applied to the transcripts by highlighting the words and sentences that met them. A second iteration used open coding (also known as initial coding) where the marked codes from the first iteration were grouped with similarly marked codes [9, 17]. These iterations were performed one after the other for the same transcription before starting the next transcription.

These resulting codes were the input for constructing the categories. In this process, the codes that had the same purpose were taken together and codes with a similar purpose were grouped together and given an overarching term. This coding step is called axial coding [9].

Event log construction Based on the screen recording and the transcription, the actions found were transformed into an event log. In addition, if applicable, additional information was also stored to enrich the data such as the location where a certain action was performed (e.g. in the console, in a script, etc.), what exactly happened in the action (e.g. what was filtered on) and then an attribute how this happened (e.g. search for a variable using $CTRL+F$). Timestamps for the event log where based on the screen recordings.

Event log analysis The frequency of activities, and typical activity flows were subsequently analysed. Next to the recorded behaviour, also the quality of the execution was assessed, by looking at both the duration of the analysis, as well as the correctness of the results. For each of these focus points, participants with differing levels of experiences where also compared.

For the analysis of the event log, the R package `bupaR` was used [12]. Because there were relatively few cases present in the event log, the analysis also consisted largely of qualitative analysis of the individual traces.

4 Identified actions

Before analysing the executed actions and flows in relation to the different experiences, duration and correctness, this section describes the identified actions, which have been subdivided in the categories preparatory, analysis, debugging, and other actions.

Preparatory actions. Actions are considered preparatory steps if they occurred mainly prior to the actual analysis itself. For the purpose of this experiment, actions were selected that had a higher relative frequency among the actions performed before the first question than during the analysis. An overview of preparatory actions is shown in Table 1.

Table 1. Preparatory actions

Action	Description
Check data	Check if the data met their expectations, if the data was tidy (each row is an observation and each column is a variable [21]).
Explore data	Viewing the data itself, e.g., in the IDE or Excel, or by consulting the data description. Whereas data checking is really exploring the quality of the data, the act of data exploring looks at the content of the data.
Load data	Checking what file type the data source had, whether column names were present, what the separator was if any, and in what directory the data file was present. This operation corresponds to importing data from [21].
Load library	In *R*, packages must be loaded before they can be used.
Read assignment	Studying the assignment. This activity was performed both at the start of the assignment, as well as during the analysis.

Analysis actions. The steps covered within this category are actions that can be performed to accomplish a specific task, and are listed in Table 2. These are actions directly related to solving the data analysis task and not, for example, emergency actions that must be performed such as solving an error message.

Debugging actions. Debugging is the third category of operations that was identified. Next to the actual debugging of the code, this category include the activities that (might) trigger debugging, which are *errors*, *warnings*, and *messages*.

Table 2. Analysis actions.

Action	Description
Evaluate results	Reflection on (intermediate) results. Is this the result I expect? Does it answer the question?
Execute code	Executing the written code
Manipulation data	This step covers the preparation of the data for a specific assignment. Eight types of data manipulation were identified. – Data grouping: looking at aggregate statistics – Data filtering: selecting rows in the data. – Data selection: selecting columns in the data. – Data joining – Data transformation: pivoting a dataset – Mutate data: add a column with calculated variables. – Change data type: changing the data type of a column. – Create object: e.g. to store intermediate results.
Prepare plot	Determine the type of graph and data mapping.
Search variable	Identifying a particular requested variable, by looking at the description file or the data itself.
Show plot	Graph formatting.
Summarize data	Calculating summary statistics such as frequency, centrality measures, and measures of variance.

Executing the code 77 times out of 377 resulted in an error. Debugging is a (series of) action(s) taken after receiving an error or warning. Most of these errors were fairly trivial to resolve. In twenty percent of the loglines registered during debugging, however, additional information was consulted on, for example, the Internet.

Other Actions The last category of actions includes adding structure, reasoning, reviewing the assignments, consulting information, and trial-and-error. Except for the review of the assignments, which was performed after completing all the assignments, these actions are fairly independent of the previous action and thus were performed at any point in the analysis. An overview of these actions can be found in Table 3. Note that as trial-and-error is a method rather than a separate action, it was not coded separately in the event log, but can be identified in the log as a pattern.

5 Analysis

In the experiment, a total of 1674 activity instances were recorded. An overview of the identified actions together with summary statistics is provided in Table 4. It can be seen that the most often observed actions are *Execute code*, *Consult information*, *Prepare data* and *Evaluate results*. Twelve of the identified actions were performed by all 11 participants at some point. Looking at the summary statistics, we observe quite significant differences in the execution frequency of actions, such as the consultation of information (ranging from 4 to 63) and the

Table 3. Other actions

Action	Description
Add structure	Adding intermediate steps and comments and structuring code in chunks.
Consult information	Four different sources were used: documentation of programming functions used, examples included in function documentations, returning to previous analyses, and consulting relevant programming course materials.
Reasoning	Thinking about performing a task was undoubtedly performed by all participants, though only seven participants cited actively thinking at certain points during the analysis.
Review solution	Before finishing, checking all the solutions whether they are correct and met the assignments.
Trial-and-error	Experimenting, by just trying out some things or comparing the outcome of different types of joins.

execution of code (ranging from 16 to 48), indicating important individual differences. Table 5 shows for each participant the total processing time (minutes) together with the total number of actions, and the number of actions per category.

Flows A first observation is that the log records direct repetitions of a certain number of actions. This is a natural consequence of the fact that information is stored in additional attributes. As such, when a participant is, for instance, consulting different sources of information directly after one another, this will not be regarded as a single "Consulting information" action, but as a sequence of smaller actions. Information of these repetitions is shown in Table 6. Because these length-one loops might clutter the analysis, it was decided to *collapse* them into single activity instances. After doing so, the number of activity instances was reduced from 1674 to 1572.

That the process of data analysis is flexible attests Figure 1, which contains a directly-follows matrix of the log. While many different (and infrequent) flows can be observed, some interesting insights can be seen. Within the analysis actions, we can see 2 groups: actions related to manipulation of data, and actions related to evaluation and visualising data. Furthermore, it can be seen that some analysis actions are often performed before or after preparatory actions, while most are not.

Duration In Figure 2, the total time spent on each of the 4 categories is shown per participant, divided in undergraduate, graduate and PhD participants. The dotted vertical lines in each group indicates the average time spent. While the limited size of the experiment does not warrant generalizable results with respect to different experience levels, it can be seen that Undergraduates spent the least time overall, while graduate spent the most time. In the latter group, we can however see a large amount of variation between participants. What is notable is that both graduate participants and PhDs spent a significantly larger amount

Table 4. Summary statistics of the identified actions.

Category	Action	#part.	Total freq.	Min. freq.	Avg. freq.	Max. freq.
Preparatory	Check data	7	11	1	1.57	3
	Explore data	10	52	2	5.20	12
	Load data	10	35	2	3.50	6
	Load library	11	39	2	3.55	9
	Read assignment	11	84	4	7.64	14
Analysis	Evaluate results	11	182	5	16.55	33
	Execute code	11	377	16	34.27	48
	Manipulate data	11	195	6	17.73	34
	Prepare plot	11	70	2	6.36	14
	Search variable	11	81	4	7.36	10
	Show plot	8	40	1	5.00	12
	Summarize data	11	44	1	4.00	8
Debugging	Debug	11	48	1	4.36	12
	Error	11	77	2	7.00	14
	Message	1	2	2	2.00	2
	Warning	3	8	1	2.67	4
Other	Add structure	11	69	3	6.27	10
	Consult information	11	229	4	20.82	63
	Reasoning	7	17	1	2.43	4
	Review solution	9	14	1	1.56	3

Table 5. Statistics per participant

Participant	Proc. time	#actions	Preparatory	Analysis	Debugging	Other
1	26.20	139	19	75	26	19
2	32.87	159	16	110	12	21
3	42.98	172	15	117	19	21
4	52.63	172	31	88	6	47
5	39.67	151	21	87	8	35
6	43.15	155	19	93	14	29
7	38.08	155	14	109	10	22
8	36.17	104	11	54	8	31
9	17.52	97	23	55	5	14
10	38.75	170	28	112	12	18
11	71.52	200	24	89	15	72

Table 6. Direct repetitions of actions

Action	Number of repetitions	Action	Number of repetitions
Consult information	54.00	Load data	7.00
Prepare data	23.00	Load library	6.00
Search variable	19.00	Debug	2.00
Add structure	14.00	Check data	1.00
Execute code	13.00	Read assignment	1.00
Explore data	7.00	Review solution	1.00

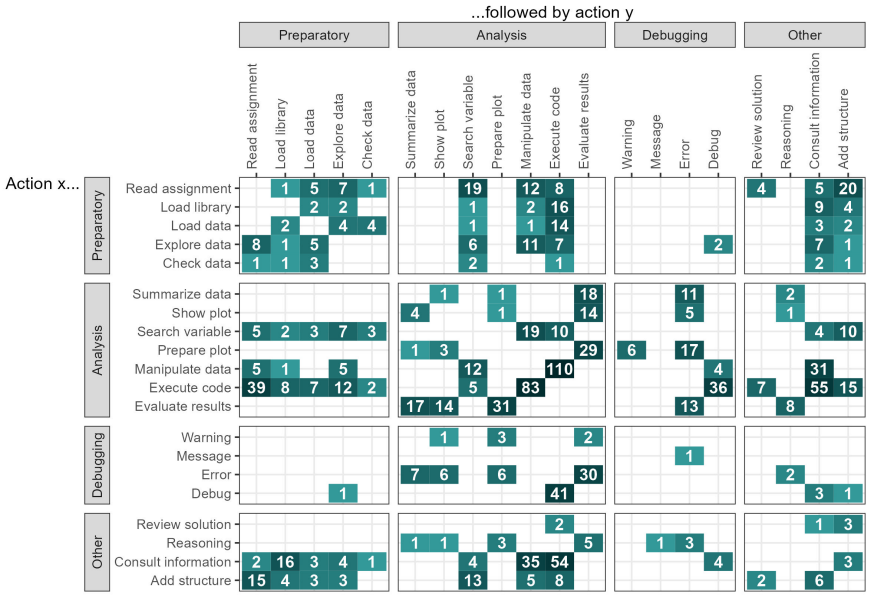

Fig. 1. Precedence flows between actions.

of time on preparatory steps, compared to undergraduate students. On average, graduate students spent more time on other actions than the other groups. Predominantly, this appeared to be the consultation of information. This might be explained by the fact that for these students, data analysis (specifically the course in R) was further removed in the past compared to undergraduate students. On the other hand, PhDs might have more expertise about usage of R and data analysis readily available.

Correctness After the experiment, the results where also scored for correctness. Table 7 shows the average scores in each group, on a scale from zero to 100%. While the differences are small, and still noting the limited scope of the experiment, a slight gap can be observed between undergraduates on the one hand, and PhDs and graduates on the other. The gap between the latter two is less apparent.

Table 7. Average scores per group.

Group	Avg score (out of 2)
Undergraduate	83.5
Graduate	91.5
PhD	93.5

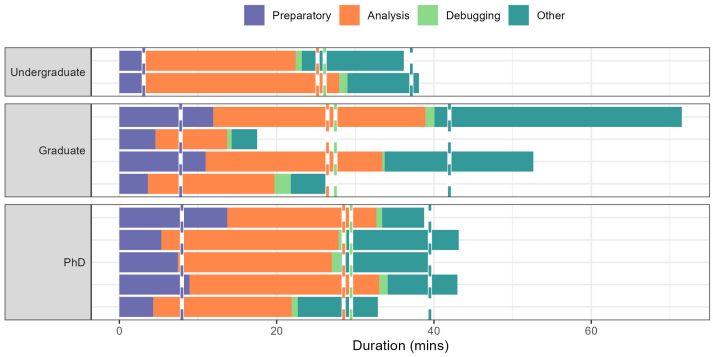

Fig. 2. Duration per category for each participant in each experience level.

Figure 3 shows a correlation matrix between the scores, the number of actions in each category, and the time spent on each category. Taking into account the small data underlying these correlations, it can be seen that no significant positive correlations with the score can be observed. However, the score is found to have a moderate negative correlation with both the amount and duration of debugging actions, as well as the duration of analysis actions. While the former seems logical, the latter is somewhat counter-intuitive. Given that no relation is found between with the number of analysis actions, the average duration of an analysis task seems to relevant. This might thus indicate that the score is negatively influenced when the analysis takes place slower, which might be a sign of inferior skills.

6 Conclusion

The steps completed during an exploratory data analysis can be divided into four categories: the preparatory steps, the analysis steps, the debug step, and finally the actions that do not belong to a category but can be used throughout the analysis process. By further breaking down the exploratory data analysis into these steps, it becomes easier to proceed step by step and thus possibly obtain better analyses. The data analysis process performed by the participants appeared to be an iterative process that involved working step-by-step towards the solution.

The experiment described in this paper clearly is only a first step towards understanding the behaviour of data analysts. Only a small amount of people participated and the analysis requested was a relatively simple exercise. As a result, the list of operations found might not be exhaustive. Furthermore, the use of *R* and *RStudio* will have caused that some of the operations are specifically related to R. While R was chosen because all participants had a basic knowledge of *R* through an introductory course received in the first bachelor year, future research is needed to see whether these steps are also relevant with respect to other programming languages or tools. Moreover, this course may have already

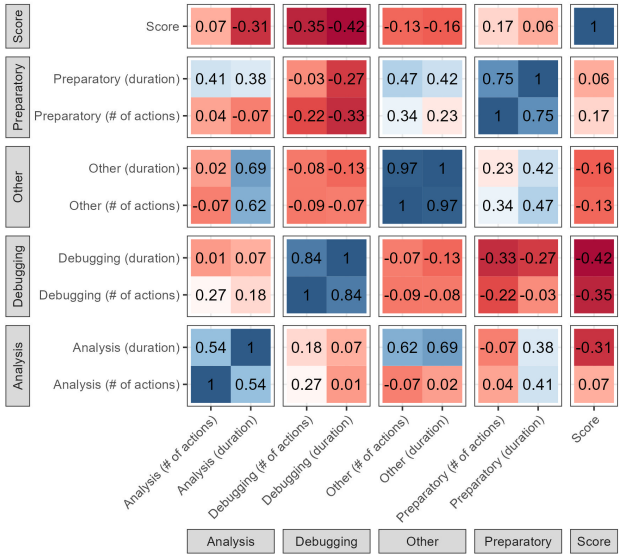

Fig. 3. Correlations between score, number of (distinct) actions in each category, and duration of each category.

taught a certain methodology, which might not generalize to other data analyst. Additionally, the fact that the participants participated voluntarily, might mean they feel more comfortable performing a data analysis in R than their peers, especially among novices.

It is recommended that further research is conducted on both the operations, the order of these operations as well as the practices of experts and novices. By using more heterogeneous participants, a more difficult task and different programming languages, it is expected that additional operations can be identified as well as differences in practices between experts and beginners. These can be used to identify the mental representations of experts and, in turn, can be used to design new teaching methods [19]. In addition, an analysis at the sub-activity level could provide insights about frequencies and a lower-level order, such as in what order the sub-activities in the act of preparing data were usually performed.

References

1. Anders Ericsson, K.: Deliberate practice and acquisition of expert performance: a general overview. Academic emergency medicine **15**(11), 988–994 (2008)
2. Anders Ericsson, K., Towne, T.J.: Expertise. Wiley Interdisciplinary Reviews: Cognitive Science **1**(3), 404–416 (2010)
3. Behrens, J.T.: Principles and procedures of exploratory data analysis. Psychological Methods **2**(2), 131 (1997)
4. Blayney, P., Kalyuga, S., Sweller, J.: Interactions between the isolated–interactive elements effect and levels of learner expertise: Experimental evidence from an accountancy class. Instructional Science **38**(3), 277–287 (2010)

5. Claes, J., Vanderfeesten, I., Gailly, F., Grefen, P., Poels, G.: The structured process modeling theory (spmt) a cognitive view on why and how modelers benefit from structuring the process of process modeling. Information Systems Frontiers **17**(6), 1401–1425 (2015)
6. Ericsson, A., Pool, R.: Peak: Secrets from the new science of expertise. Random House (2016)
7. Ericsson, K.A., et al.: The influence of experience and deliberate practice on the development of superior expert performance. The Cambridge handbook of expertise and expert performance **38**(685-705), 2–2 (2006)
8. Hinds, P.J.: The curse of expertise: The effects of expertise and debiasing methods on prediction of novice performance. Journal of experimental psychology: applied **5**(2), 205 (1999)
9. Holton, J.A.: The coding process and its challenges. The Sage handbook of grounded theory **3**, 265–289 (2007)
10. Jans, M., Soffer, P., Jouck, T.: Building a valuable event log for process mining: an experimental exploration of a guided process. Enterprise Information Systems **13**(5), 601–630 (2019)
11. Jans, M.: From relational database to valuable event logs for process mining purposes: a procedure. Tech. rep., Technical report, Hasselt University (2017)
12. Janssenswillen, G., Depaire, B., Swennen, M., Jans, M., Vanhoof, K.: bupar: Enabling reproducible business process analysis. Knowledge-Based Systems **163**, 927–930 (2019)
13. Kitchin, R.: The data revolution: Big data, open data, data infrastructures and their consequences. Sage (2014)
14. Mayer-Schoenberger, V., Cukier, K.: The rise of big data: How it's changing the way we think about the world. Foreign affairs **92**(3), 28–40 (2013)
15. O'Neil, C., Schutt, R.: Doing data science: Straight talk from the frontline. " O'Reilly Media, Inc." (2013)
16. Saldaña, J.: Coding and analysis strategies. The Oxford handbook of qualitative research pp. 581–605 (2014)
17. Saldaña, J.: The coding manual for qualitative researchers. The coding manual for qualitative researchers pp. 1–440 (2021)
18. Saltz, J.S., Shamshurin, I.: Exploring the process of doing data science via an ethnographic study of a media advertising company. In: 2015 IEEE international conference on big data (Big Data). pp. 2098–2105. IEEE (2015)
19. Spector, J.M., Ohrazda, C.: Automating instructional design: Approaches and limitations. In: Handbook of research on educational communications and technology, pp. 681–695. Routledge (2013)
20. Sweller, J.: Cognitive load theory: Recent theoretical advances. (2010)
21. Wickham, H., Grolemund, G.: R for data science: import, tidy, transform, visualize, and model data. " O'Reilly Media, Inc." (2016)
22. Yates, K.A., Clark, R.E.: Cognitive task analysis. International Handbook of Student Achievement. New York, Routledge (2012)
23. Zerbato, F., Soffer, P., Weber, B.: Initial insights into exploratory process mining practices. In: International Conference on Business Process Management. pp. 145–161. Springer (2021)
24. Zimmermann, L., Zerbato, F., Weber, B.: Process mining challenges perceived by analysts: An interview study. In: International Conference on Business Process Modeling, Development and Support, International Conference on Evaluation and Modeling Methods for Systems Analysis and Development. pp. 3–17. Springer (2022)