Universidade do Minho
Escola de Engenharia
Departamento de Informática

José Miguel Silva Dias

Humanized Data Cleaning

March 2021

Universidade do Minho
Escola de Engenharia
Departamento de Informática

José Miguel Silva Dias

Humanized Data Cleaning

Master dissertation
Integrated Master in Informatics Engineering

Dissertation supervised by
Jácome Cunha
Rui Pereira

March 2021

Despacho RT - 31 /2019 - Anexo 3

Declaração a incluir na Tese de Doutoramento (ou equivalente) ou no trabalho de Mestrado

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

ACKNOWLEDGEMENTS

Agradeço desde já a todos os que contribuíram para que eu hoje esteja aqui.

Antes de mais gostava de agradecer aos meus orientadores Jácome Cunha e Rui Pereira pelo suporte e pela ajuda que provaram ser ao longo deste ano. Pela disponibilidade que demonstraram e pelo conhecimento que partilharam. Sem eles esta dissertação não seria o que é hoje.

Um especial agradecimento á minha família pelos 23 anos em que cuidaram do mim e da minha educação. Agradecimento pelo apoio e incentivo e por me terem guiado até aqui hoje.

Agradeço também aos meus amigos por me terem acompanhado até aqui.

Por ultimo gostava de agradecer a todos pela companhia especialmente durante este ano especialmente difícil.

ABSTRACT

Data science has started to become one of the most important skills someone can have in the modern world, due to data taking an increasingly meaningful role in our lives. The accessibility of data science is however limited, requiring complicated software or programming knowledge. Both can be challenging and hard to master, even for the simpler tasks.

Currently, in order to clean data you need a data scientist. The process of data cleaning, consisting of removing or correcting entries of a data set, usually requires programming knowledge as it is mostly performed using programming languages such as Python and R (kag). However, data cleaning could be performed by people that may possess better knowledge of the data domain, but lack the programming background, if this barrier is removed.

We have studied current solutions that are available on the market, the type of interface each one uses to interact with the end users, such as a control flow interface, a tabular based interface or block-based languages. With this in mind, we have approached this issue by providing a new data science tool, termed Data Cleaning for All (DCA), that attempts to reduce the necessary knowledge to perform data science tasks, in particular for data cleaning and curation. By combining Human-Computer Interaction (HCI) concepts, this tool is: *simple* to use through direct manipulation and showing transformation previews; allows users to *save time* by eliminate repetitive tasks and automatically calculating many of the common analyses data scientists must perform; and suggests data transformations based on the contents of the data, allowing for a *smarter* environment.

Keywords – Data Cleaning, Data Science

RESUMO

A ciência de dados tornou-se uma das capacidades mais importantes que alguém pode possuir no mundo moderno, devido aos dados serem cada vez mais importantes na nossa sociedade. A acessibilidade da ciência de dados é no entanto limitada, requer *software* complicado ou conhecimentos de programação. Ambos podem ser desafiantes e dificeis de aprender bem, mesmo para tarefas simples.

Atualmente, para efetuar a limpeza de dados é necessário um *Data Scientist*. O processo de limpeza de dados, que consiste em remover ou corrigir entradas de um *dataset*, é normalmente efetuado utilizando linguagens de programação como Python e R (kag). No entanto, a limpeza de dados poderia ser efetuada por profissionais que possuam melhor conhecimento sobre o dominio dos dados a tratar, mas que não possuam uma formação em ciencias da computação

Estudamos soluções que estão presentes no mercado e o tipo de interface que cada uma usa para interagir com o utilizador, seja através de diagramas de fluxo de controlo, interfaces tabulares ou recorrendo a linguagens de programação baseadas em blocos. Com isto em mente, abordamos o problema através do desenvolvimento de uma nova plataforma onde podemos efetuar tarefas de ciências de dados com o nome *Data Cleaning for All (DCA)*. Com esta ferramenta esperamos reduzir os conhecimentos necessários para efetuar tarefas nesta área, especialmente na área da limpeza de dados. Através da combinação de conceitos de HCI, a plataforma é: simples de usar através da manipulação direta dos dados e da demonstração de pré-vizualizações das transformações; permite aos utilizadores poupar tempo através da eliminação de tarefas repetitivas ao calcular muitas das métricas que *Data Scientist* têm de calcular; e sugere transformações dos dados baseadas nos conteudos dos mesmos, permitindo um ambiente mais inteligente.

Palavras chave – Ciencia de Dados, Limpeza de Dados

Despacho RT - 31 /2019 - Anexo 4

Declaração a incluir na Tese de Doutoramento (ou equivalente) ou no trabalho de Mestrado

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF LISTINGS

LIST OF ABBREVIATIONS

CSCW Computer-Supported Cooperative Work. 1, 2, 83
csv Comma Separated Values. 1, 26

DC Data Cleaning. viii, 1, 2, 15, 20, 22, 33, 41, 43, 45
DCA Data Cleaning for All. iii, iv, vi–ix, 1, 12, 18–20, 23–27, 30–33, 35–43, 45, 46, 56–58, 60–62, 64
DS Data Science. viii, 1, 12, 20, 28, 33, 41, 45

HCI Human-Computer Interaction. iii, iv, 1, 2, 74

MEI Mestrado em Engenharia Informática. 1

SUS System Usability Scale. ix, xi, 1, 38–41, 44, 45

TPB Tableau Prep Builder. ix, x, 1, 28, 30–33, 35–41, 45, 46, 58–60, 62–64

UM Universidade do Minho. 1

1

INTRODUCTION

"The key word in data science is not "data", it is "science". Data science is only useful when the data is used to answer a question. That is the science part of the equation" (Leek, 2013).

Nowadays we live in a world overflowing with data and with each passing year it takes a more central role in our development. However data by itself is of little use, as such, there is a higher demand for data scientists and their skills in data analysis. Nowadays it is at least required someone with a background in programming as the task of data processing is usually executed using programming languages. A study by IBM, Business-Higher Education Forum and Burning Glass expects a growth of 15% from 2015 to 2020 in the number of job listings for data science and analytics jobs just in the US (Markow et al., 2017). In order to answer to the current high demand for data scientists, the Portuguese government issued the Contract for the Higher Education Legislature for 2020-2023, where it advocates the need for all students and graduates to have an increased capacity for data processing. This means that 100% of university students should have the opportunity to learn data science (Government, 2019).

1.1 MOTIVATION

Data Scientists are relied upon due to their skills in data processing, which are closely tied to their programming knowledge. A study performed by Kaggle[1] suggests that the programming language (PL) Python[2] was, in 2017, the most used tool for Data Science with R[3] showing popularity among Statisticians (kag). One of the main reasons as for why we need solutions that attempt to reduce the need of programming qualifications and try to simplify the process of data science is connected to the emergence of big data and the lack of qualified data scientists (Lopes et al., 2018). Additionally, it would also help users with little to no knowledge in computer science become more capable to perform data science tasks. By reducing the programming qualifications required, we can allow people with better data domain knowledge to perform a data science task. They may have a better understanding of the underlying data and in turn be more suited when analyzing the data and detecting incorrect values. As an example, a biologist would be better suited in dealing with biology related data as they would have a better understanding of what the values they would observe represent, while a data scientist would struggle.

According to the study by Cunha et al. (2020), in order to increase the number of data scientists there are two options. On one hand, as suggested by governments and companies, the academia

[1] Subsidiary company of Google LLC and online community of data scientists
[2] https://www.python.org/
[3] https://www.r-project.org/

should prepare courses and degrees to teach the younger generation. On the other hand researchers and the industry should attempt to build tools for non programmers to be capable of performing data science activities. The study by Cunha et al. (2020) defines this latter alternative as humanized data science.

The main motivation for this thesis is to provide users with a humanized data science tool so that other professionals, those that do not have a computer science or programming background, can access and transform data. As data science becomes more relevant and inseparable from our daily lives it is important that professionals, of any background, are capable of performing data science tasks.

This thesis will focus on only one of the areas of data science: data cleaning. Data cleaning consists of, as the name suggests, eliminating "dirty" entries in the data set, some examples include removing null values, or removing different representations of the same values. During the data cleaning process we keep finding recurring errors, such as syntax errors, missing values and duplicate entries, among others. These problems and their implications within the context of data quality are explored by Müller and Freytag (2003). Our goal with this thesis is not to innovate how data is transformed behind the scenes, but it is to improve how we interact with data.

1.2 KEY CONTRIBUTIONS

With the conclusion of this thesis, we have developed the first steps towards a tool for data cleaning for end users. An end users is anyone that creates a program to support their work or a hobby, they can be secretaries, teachers, scientists among others (Ko et al., 2011). It provides end users with a tool for data cleaning tasks with a low bar of entry. In this thesis we have conducted an empirical validation of our tool in order to check the viability of our approach and validate our tool while comparing with a currently available market solution. Our results have shown that users prefer working with our tool in order to perform most tasks.

We have also contributed through:

- A paper for HCI International 2020 (Appendix E);
- A paper for the Interrogating Data Science CSCW 2020 Workshop (Appendix F).

1.3 RESEARCH QUESTIONS

A few important questions arose during this thesis work, relative to the design and implementation of a Data Cleaning (DC) tool for the average end user:

- **RQ1** - *Is is possible to design and implement a tool that can allow end users to do data cleaning?* Programming languages are still a mainstay as tools for DC, why are there no tools that are easier to use more popular. In order to make DC more approachable we need to study and improve upon current solutions.
- **RQ2** - *Does our solution improve productivity of users and their user experience?* A complete solution for this problem needs to improve the level of productivity of the users or at least keep it the same and improve the users experience mainly its accessibility and making it easier to understand.

During this thesis we worked in order to answer all of these questions. We also acknowledge that, in order to answer the last question, we require an empirical study.

1.4 STRUCTURE OF THE THESIS

This thesis is structured as follows:

- **Section** 2 - contains the State of the Art, here we will look at existing solutions and some of their features that are useful for the data cleaning process and focus on the way they interact with the user.
- **Section** 3 - contains the Solution Design, here we will look at the design choices made, the solutions chosen and the problems they resolve.
- **Section** 4 - contains the Development of the Tool, here we will look at how we have developed the solutions presented in the previous chapter.
- **Section** 5 - contains the Empirical Validation, here we will explore the design of the study, the execution and analysis of the results.
- **Section** 6 - contains the Conclusion, here we will look at some concluding remarks and future work.

2

STATE OF THE ART

This chapter will focus on the tools with useful features or interesting methods of interaction. Additionally we will study and divide such tools according to their method of interaction with the end-user.

We will start by looking at control flows (Section 2.1), then tabular based interfaces (Section 2.2) and finally block-based languages (Section 2.3).

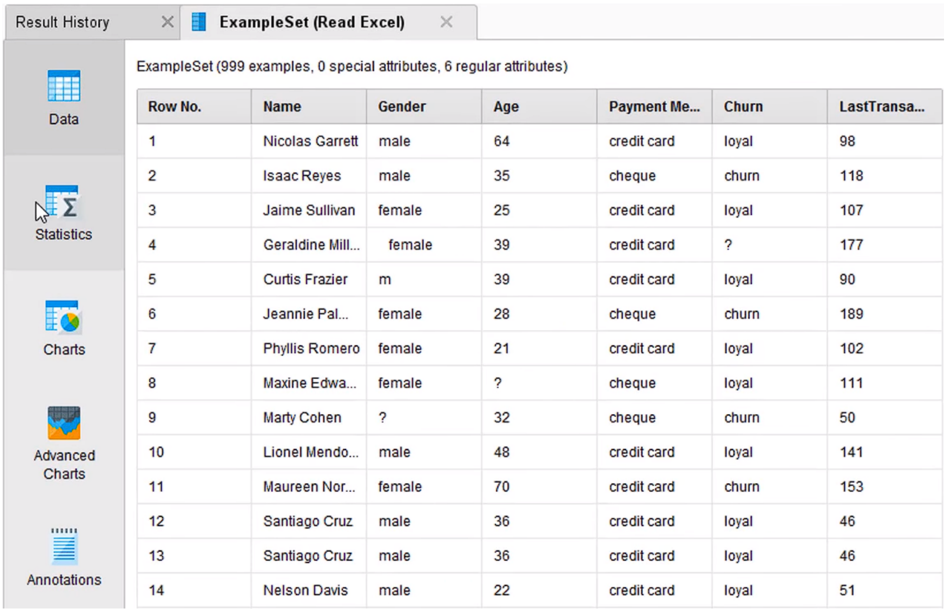
2.1 CONTROL FLOW

There are various tools which use a control flow interface, some of these tools include Tableau Prep (Software), RapidMiner (rap), Knime (kni), Alteryx (alt), Azure ML (Microsoft), Orange (of Ljubljana) and Weka (wek) among others. Each one uses a similar way of interaction, providing a tabular view of the data and some statistics, but most of the transformations are applied through a sequence of instructions represented in a control flow. As an example, we will use RapidMiner to present this method of interaction with the user.

RapidMiner (rap) is one of the tools that can be used in data science and fulfills every step in the process of data processing. When starting off you will first be prompted to import a data set from the supported files. Afterwards RapidMiner will allow the visualization of the data set in a table format as shown in Figure 1 and some statistics for each attribute such as unique values, the most common, among others, as shown in Figure 2. RapidMiner also uses a control flow type of interface in order to perform data cleaning as shown in Figure 3.

In Figure 1 we can see how RapidMiner shows us data. This figure details costumer churn in an example store. This table format allows the user to be able to easily identify errors and "dirty" data, so long as the error is present among the entries that we can see at any given time. In Figure 1 we can quickly see some missing values and typos in the gender column. For example, we can see female written with a leading space in the fourth row , male being represented as a single m in the fifth row and a missing gender in the ninth row.

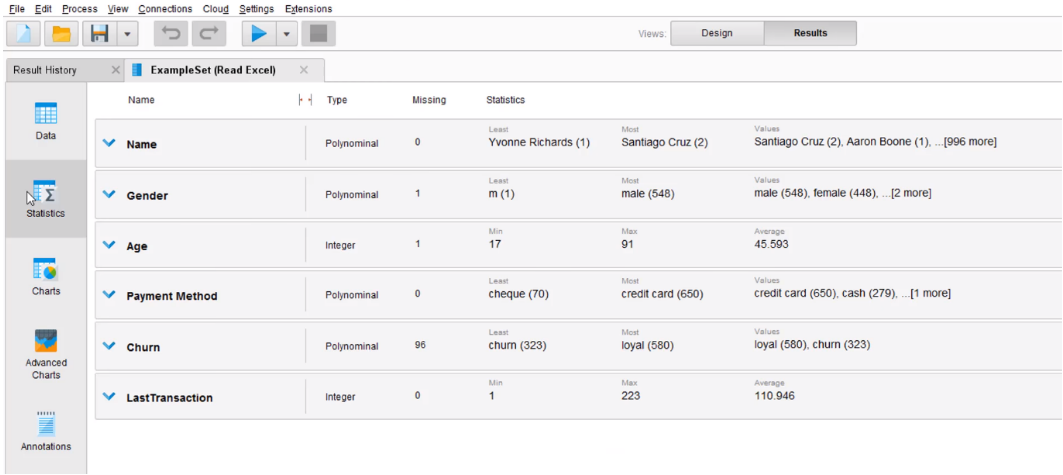
Shown in Figure 2 are several statistics that are available in RapidMiner. Statistics, such as the ones presented, allow the user to, most of the times, easily identify some of the most common errors such as unnormalized data, null values in certain attributes, and duplicate entries. For example, looking at Figure 2 we can see that the name "Santiago Cruz" appears twice, which may suggest a duplicate entry. Looking at genders we can see they are not uniformly categorized since 4 different gender values can be found ("female", " female", "male", "m").



ExampleSet (999 examples, 0 special attributes, 6 regular attributes)

Row No.	Name	Gender	Age	Payment Me...	Churn	LastTransa...
1	Nicolas Garrett	male	64	credit card	loyal	98
2	Isaac Reyes	male	35	cheque	churn	118
3	Jaime Sullivan	female	25	credit card	loyal	107
4	Geraldine Mill...	female	39	credit card	?	177
5	Curtis Frazier	m	39	credit card	loyal	90
6	Jeannie Pal...	female	28	cheque	churn	189
7	Phyllis Romero	female	21	credit card	loyal	102
8	Maxine Edwa...	female	?	cheque	loyal	111
9	Marty Cohen	?	32	cheque	churn	50
10	Lionel Mendo...	male	48	credit card	loyal	141
11	Maureen Nor...	female	70	credit card	churn	153
12	Santiago Cruz	male	36	credit card	loyal	46
13	Santiago Cruz	male	36	credit card	loyal	46
14	Nelson Davis	male	22	credit card	loyal	51

Figure 1: Visual representation of a data set in RapidMiner



Name	Type	Missing	Statistics		
Name	Polynominal	0	Least Yvonne Richards (1)	Most Santiago Cruz (2)	Values Santiago Cruz (2), Aaron Boone (1), ...[996 more]
Gender	Polynominal	1	Least m (1)	Most male (548)	Values male (548), female (448), ...[2 more]
Age	Integer	1	Min 17	Max 91	Average 45.593
Payment Method	Polynominal	0	Least cheque (70)	Most credit card (650)	Values credit card (650), cash (279), ...[1 more]
Churn	Polynominal	96	Least churn (323)	Most loyal (580)	Values loyal (580), churn (323)
LastTransaction	Integer	0	Min 1	Max 223	Average 110.946

Figure 2: Data set statistics in RapidMiner

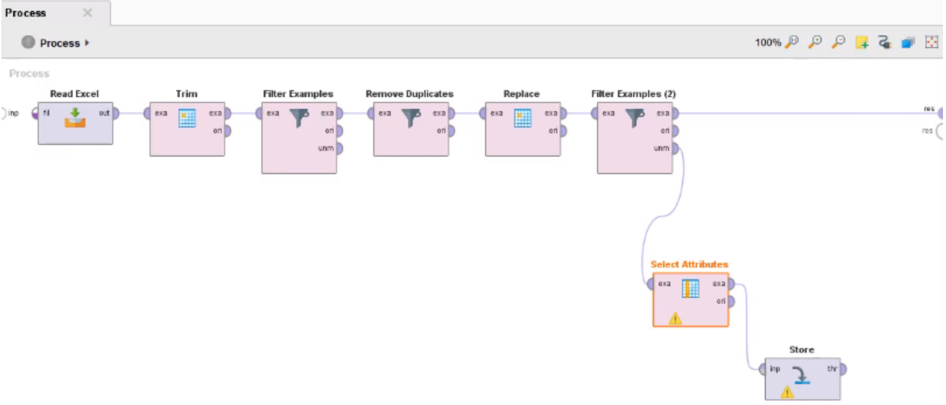


Figure 3: Data cleaning control flow in RapidMiner

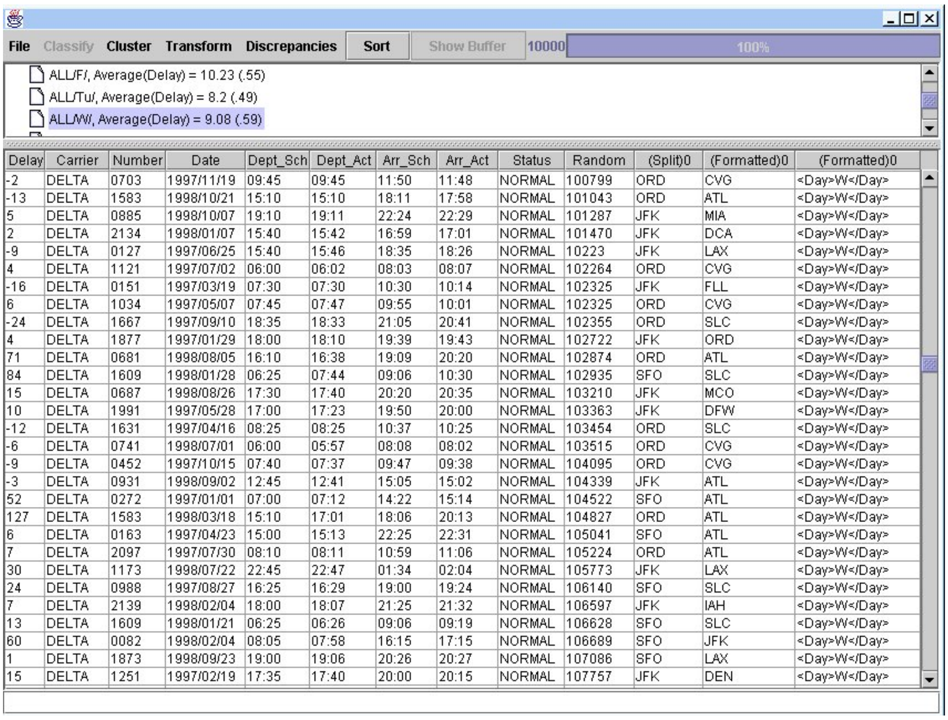
With RapidMiner, the data cleaning process is shown as a sequence of blocks (a control flow) as shown in Figure 3. Each block represents an operation that will be used to process the data set, following the order of the flow itself. The operations include altering a column in order to normalize values with a replace function, by using regular expressions, which may not be friendly to non-programmers. Some other operations we can see in the Figure 3 above include:

- Trim - remove white spaces before or after the values
- Filter - filter entries that match a certain criteria, it can also be used to split a data set.
- Remove Duplicates - removes duplicate entries when it verifies matching values
- Replace - replaces anything in a specified column that matches the criteria specified, supports regular expressions.

Presented by Lopes et al. (2018), their work provides a browser based application that allows for the construction of workflows composed of sequential tasks and presents it in a web interface. One of the methods this solution uses to ease the burden on the user is by attempting to stop the users from adding impossible tasks to the workflow given its current state. Tasks can be added while the workflow is running as it can be stopped at any point during its execution (Lopes et al., 2018).

2.2 TABULAR BASED INTERFACE

In the context of data science, a tabular based interface is any interface where it is possible to interact directly with data. One tool which most people should be familiar with that has a tabular based interface is Microsoft Excel.



ALL/F/, Average(Delay) = 10.23 (.55)
ALL/Tu/, Average(Delay) = 8.2 (.49)
ALL/W/, Average(Delay) = 9.08 (.59)

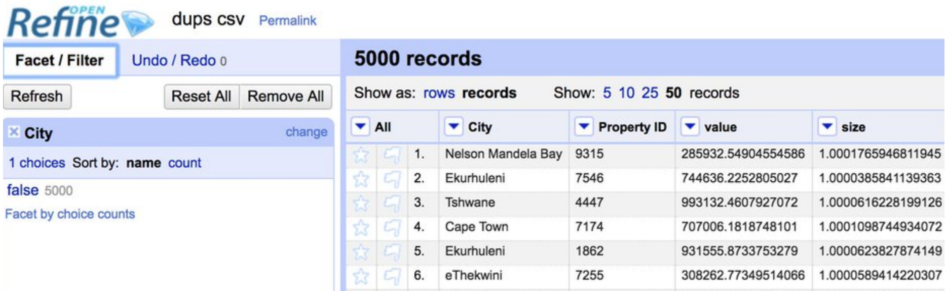
Delay	Carrier	Number	Date	Dept_Sch	Dept_Act	Arr_Sch	Arr_Act	Status	Random	(Split)0	(Formatted)0	(Formatted)0
-2	DELTA	0703	1997/11/19	09:45	09:45	11:50	11:48	NORMAL	100799	ORD	CVG	<Day>W</Day>
-13	DELTA	1583	1998/10/21	15:10	15:10	18:11	17:58	NORMAL	101043	ORD	ATL	<Day>W</Day>
5	DELTA	0885	1998/10/07	19:10	19:11	22:24	22:29	NORMAL	101287	JFK	MIA	<Day>W</Day>
2	DELTA	2134	1998/01/07	15:40	15:42	16:59	17:01	NORMAL	101470	JFK	DCA	<Day>W</Day>
-9	DELTA	0127	1997/06/25	15:40	15:46	18:35	18:26	NORMAL	10223	JFK	LAX	<Day>W</Day>
4	DELTA	1121	1997/07/02	06:00	06:02	08:03	08:07	NORMAL	102264	ORD	CVG	<Day>W</Day>
-16	DELTA	0151	1997/03/19	07:30	07:30	10:30	10:14	NORMAL	102325	JFK	FLL	<Day>W</Day>
6	DELTA	1034	1997/05/07	07:45	07:47	09:55	10:01	NORMAL	102325	ORD	CVG	<Day>W</Day>
-24	DELTA	1667	1997/09/10	18:35	18:33	21:05	20:41	NORMAL	102355	ORD	SLC	<Day>W</Day>
4	DELTA	1877	1997/01/29	18:00	18:10	19:39	19:43	NORMAL	102722	JFK	ORD	<Day>W</Day>
71	DELTA	0681	1998/08/05	16:10	16:38	19:09	20:20	NORMAL	102874	ORD	ATL	<Day>W</Day>
84	DELTA	1609	1998/01/28	06:25	07:44	09:06	10:30	NORMAL	102935	SFO	SLC	<Day>W</Day>
15	DELTA	0687	1998/08/26	17:30	17:40	20:20	20:35	NORMAL	103210	JFK	MCO	<Day>W</Day>
10	DELTA	1991	1997/05/28	17:00	17:23	19:50	20:00	NORMAL	103363	JFK	DFW	<Day>W</Day>
-12	DELTA	1631	1997/04/16	08:25	08:25	10:37	10:25	NORMAL	103454	ORD	SLC	<Day>W</Day>
-6	DELTA	0741	1998/07/01	06:00	05:57	08:08	08:02	NORMAL	103515	ORD	CVG	<Day>W</Day>
-9	DELTA	0452	1997/10/15	07:40	07:37	09:47	09:38	NORMAL	104095	ORD	CVG	<Day>W</Day>
-3	DELTA	0931	1998/09/02	12:45	12:41	15:05	15:02	NORMAL	104339	JFK	ATL	<Day>W</Day>
52	DELTA	0272	1997/01/01	07:00	07:12	14:22	15:14	NORMAL	104522	SFO	ATL	<Day>W</Day>
127	DELTA	1583	1998/03/18	15:10	17:01	18:06	20:13	NORMAL	104827	ORD	ATL	<Day>W</Day>
6	DELTA	0163	1997/04/23	15:00	15:13	22:25	22:31	NORMAL	105041	SFO	ATL	<Day>W</Day>
7	DELTA	2097	1997/07/30	08:10	08:11	10:59	11:06	NORMAL	105224	ORD	ATL	<Day>W</Day>
30	DELTA	1173	1998/07/22	22:45	22:47	01:34	02:04	NORMAL	105773	JFK	LAX	<Day>W</Day>
24	DELTA	0988	1997/08/27	16:25	16:29	19:00	19:24	NORMAL	106140	SFO	SLC	<Day>W</Day>
7	DELTA	2139	1998/02/04	18:00	18:07	21:25	21:32	NORMAL	106597	JFK	IAH	<Day>W</Day>
13	DELTA	1609	1998/01/21	06:25	06:26	09:06	09:19	NORMAL	106628	SFO	SLC	<Day>W</Day>
60	DELTA	0082	1998/02/04	08:05	07:58	16:15	17:15	NORMAL	106689	SFO	JFK	<Day>W</Day>
1	DELTA	1873	1998/09/23	19:00	19:06	20:26	20:27	NORMAL	107086	SFO	LAX	<Day>W</Day>
15	DELTA	1251	1997/02/19	17:35	17:40	20:00	20:15	NORMAL	107757	JFK	DEN	<Day>W</Day>

Figure 4: Potter's Wheel spreadsheet like interface

Potter's Wheel is a software platform developed in the year 2000 and seems to no longer be supported. The purpose of Potter's Wheel was to clean, analyze and transform data as mentioned in (pwa).

Potter's Wheel uses a interface inspired in spreadsheets (Figure 4) and shows live updates, displaying immediate feedback on performed transformations. Error detection is done automatically in the background. As a user, you can also define custom domains and define the algorithms to enforce domain constraints. In order to simplify users' experience, Potter's Wheel allows one to define expected results from examples, and automatically infers regular expressions describing the domain format (Müller and Freytag, 2003). Another feature Potter's Wheel has is the ability to store sequences of transformations as macros in order to apply them to similar data sets (Raman and Hellerstein, 2001).

OpenRefine (ope), formerly GoogleRefine, is another option when it comes to performing data cleaning tasks. OpenRefine still requires some programming knowledge, however, it allows the user to see a preview of the expressions written on a small sample of the actual data. This is a different approach and helps users while preventing mistakes. OpenRefine's interface is as shown in Figure 5.



City change

1 choices Sort by: name count

false 5000

Facet by choice counts

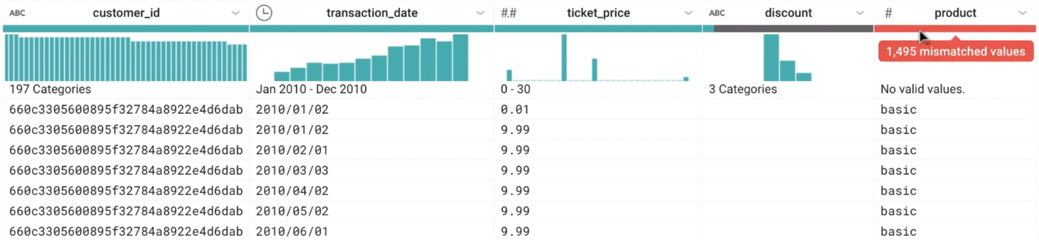
5000 records

Show as: rows records Show: 5 10 25 50 records

All			City	Property ID	value	size
☆		1.	Nelson Mandela Bay	9315	285932.54904554586	1.0001765946811945
☆		2.	Ekurhuleni	7546	744636.2252805027	1.0000385841139363
☆		3.	Tshwane	4447	993132.4607927072	1.0000616228199126
☆		4.	Cape Town	7174	707006.1818748101	1.0001098744934072
☆		5.	Ekurhuleni	1862	931555.8733753279	1.0000623827874149
☆		6.	eThekwini	7255	308262.77349514066	1.0000589414220307

Figure 5: OpenRefine Interface

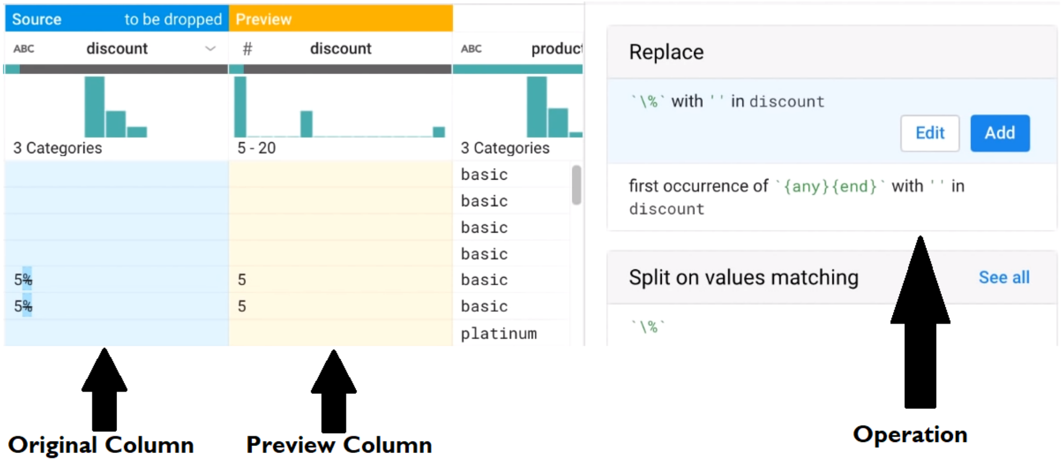
Trifacta Wrangler (Trifacta) is the free plan for the Trifacta program solutions (Wrangler, Wrangler Pro and Wrangler Enterprise). Trifacta Wrangler uses a similar control flow interface to RapidMiner, the one shown in Figure 3. However it also allows for a tabular view of the data that provides some useful information on each of the columns' data as seen in Figure 6. Trifacta, similar to OpenRefine also allows users to see a preview column of any operation before it is actually performed as shown in Figure 7. Trifacta calls a sequence of transformations a recipe and it allows backtracking inside a recipe and add new steps.



customer_id	transaction_date	ticket_price	discount	product
197 Categories	Jan 2010 - Dec 2010	0 - 30	3 Categories	No valid values.
660c3305600895f32784a8922e4d6dab	2010/01/02	0.01		basic
660c3305600895f32784a8922e4d6dab	2010/01/02	9.99		basic
660c3305600895f32784a8922e4d6dab	2010/02/01	9.99		basic
660c3305600895f32784a8922e4d6dab	2010/03/03	9.99		basic
660c3305600895f32784a8922e4d6dab	2010/04/02	9.99		basic
660c3305600895f32784a8922e4d6dab	2010/05/02	9.99		basic
660c3305600895f32784a8922e4d6dab	2010/06/01	9.99		basic

Figure 6: Trifacta Table View

Another interesting approach to data cleaning is Winpure (win). Winpure provides a tabular view of the data and a data statistics section as shown in the bottom left and right halves of Figure 8



Source ABC discount	Preview # discount	ABC product
3 Categories	5 - 20	3 Categories
		basic
		basic
		basic
		basic
5%	5	basic
5%	5	basic
		platinum

Replace

`\%` with '' in discount

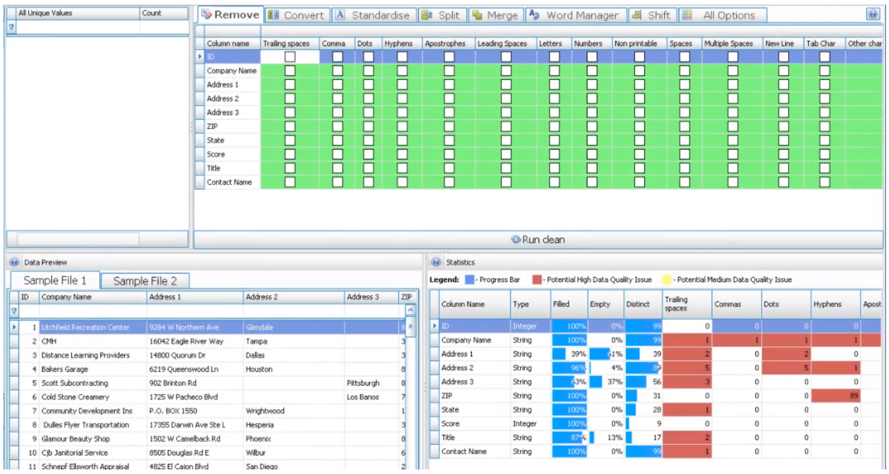
first occurrence of `{any}{end}` with '' in discount

Split on values matching

`\%`

Figure 7: Trifacta Preview Column

respectively. Winpure also provides what is referred to as a cleaning matrix, shown in the upper half of Figure 8. The cleaning matrix contains the various columns of the current table along with some possible operations we can apply to each individual column. A possible operation, but not the only one, is the conversion of a column to all uppercase or lowercase.



All Unique Values	Count

Column name	Trailing spaces	Comma	Dots	Hyphens	Apostrophes	Leading Spaces	Letters	Numbers	Non printable	Spaces	Multiple Spaces	New Line	Tab Char	Other char
ID	□	□	□	□	□	□	□	□	□	□	□	□	□	
Company Name	□	□	□	□	□	□	□	□	□	□	□	□	□	
Address 1	□	□	□	□	□	□	□	□	□	□	□	□	□	
Address 2	□	□	□	□	□	□	□	□	□	□	□	□	□	
Address 3	□	□	□	□	□	□	□	□	□	□	□	□	□	
ZIP	□	□	□	□	□	□	□	□	□	□	□	□	□	
State	□	□	□	□	□	□	□	□	□	□	□	□	□	
Score	□	□	□	□	□	□	□	□	□	□	□	□	□	
Title	□	□	□	□	□	□	□	□	□	□	□	□	□	
Contact Name	□	□	□	□	□	□	□	□	□	□	□	□	□	

ID	Company Name	Address 1	Address 2	Address 3	ZIP
1	Litchfield Recreation Center	9284 W Northern Ave	Glendale		8
2	CMH	16042 Eagle River Way	Tampa		3
3	Distance Learning Providers	14800 Quorum Dr	Dallas		3
4	Bakers Garage	6219 Queenswood Ln	Houston		8
5	Scott Subcontracting	902 Brinton Rd		Pittsburgh	8
6	Cold Stone Creamery	1725 W Pacheco Blvd		Los Banos	7
7	Community Development Ins	P.O. BOX 1550	Wrightwood		1
8	Dulles Flyer Transportation	17355 Darwin Ave Ste L	Hesperia		3
9	Glamour Beauty Shop	1502 W Camelback Rd	Phoenix		8
10	Cjb Janitorial Service	8505 Douglas Rd E	Wilbur		6
11	Schnepf Ellsworth Appraisal	4825 El Cajon Blvd	San Diego		2

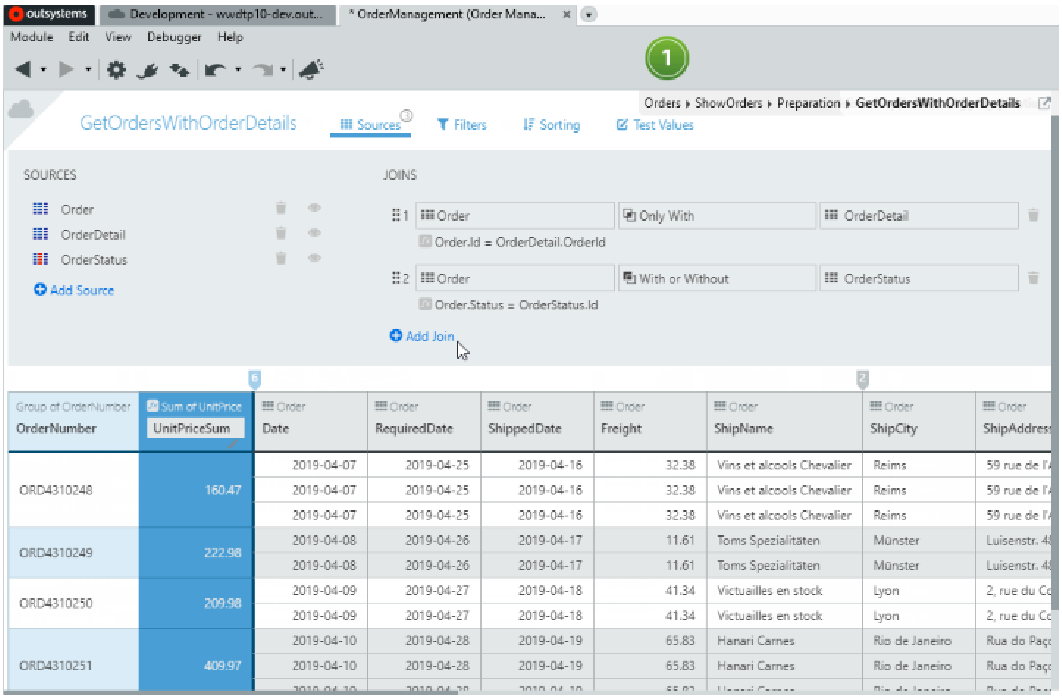
Legend: 🟦 - Progress Bar 🟥 - Potential High Data Quality Issue 🟨 - Potential Medium Data Quality Issue

Column Name	Type	Filled	Empty	Distinct	Trailing spaces	Commas	Dots	Hyphens	Apost
ID	Integer	100%	0%	99	0	0	0	0	
Company Name	String	100%	0%	99	1	1	1	1	
Address 1	String	39%	61%	39	2	0	2	0	
Address 2	String	96%	4%	89	5	0	5	1	
Address 3	String	63%	37%	56	3	0	0	0	
ZIP	String	100%	0%	31	0	0	0	89	
State	String	100%	0%	28	1	0	0	0	
Score	Integer	100%	0%	9	0	0	0	0	
Title	String	87%	13%	17	2	0	0	0	
Contact Name	String	100%	0%	99	1	0	0	0	

Figure 8: Winpure Cleaning

The OutSystems platform (OutSystems, a) is a low code platform for the development of software solutions. Here we're interested in the tool that the platform includes for database integration and interaction (OutSystems, b). This tool allows the end user to query and aggregate data visually, which in turn means that developers with any skill set can work with the data. Shown in Figure 9 is the tabular view provided in the platform for data interaction.

SpreadDB is a spreadsheet-based user interface for typical users who do not know SQL in order to query and update data of relational databases (Chatvichienchai and Kawasaki, 2018). SpreadDB is built to operate on Microsoft Excel and will keep the spreadsheet updated with the current data from the back end database (Chatvichienchai and Kawasaki, 2018).


outsystems
Development - wwdtp10-dev.out...
* OrderManagement (Order Mana...
Module
Edit
View
Debugger
Help
1
Orders ▸ ShowOrders ▸ Preparation ▸ GetOrdersWithOrderDetails
GetOrdersWithOrderDetails
Sources
Filters
Sorting
Test Values
Add Source
Add Join

SOURCES

- Order
- OrderDetail
- OrderStatus

JOINS

1	Order	Only With	OrderDetail
	Order.Id = OrderDetail.OrderId		
2	Order	With or Without	OrderStatus
	Order.Status = OrderStatus.Id		

Group of OrderNumber OrderNumber	Sum of UnitPrice UnitPriceSum	Order Date	Order RequiredDate	Order ShippedDate	Order Freight	Order ShipName	Order ShipCity	Order ShipAddres
ORD4310248	160.47	2019-04-07	2019-04-25	2019-04-16	32.38	Vins et alcools Chevalier	Reims	59 rue de l'A
		2019-04-07	2019-04-25	2019-04-16	32.38	Vins et alcools Chevalier	Reims	59 rue de l'A
		2019-04-07	2019-04-25	2019-04-16	32.38	Vins et alcools Chevalier	Reims	59 rue de l'A
ORD4310249	222.98	2019-04-08	2019-04-26	2019-04-17	11.61	Toms Spezialitäten	Münster	Luisenstr. 48
		2019-04-08	2019-04-26	2019-04-17	11.61	Toms Spezialitäten	Münster	Luisenstr. 48
ORD4310250	209.98	2019-04-09	2019-04-27	2019-04-18	41.34	Victuailles en stock	Lyon	2, rue du Co
		2019-04-09	2019-04-27	2019-04-18	41.34	Victuailles en stock	Lyon	2, rue du Co
ORD4310251	409.97	2019-04-10	2019-04-28	2019-04-19	65.83	Hanari Carnes	Rio de Janeiro	Rua do Paço
		2019-04-10	2019-04-28	2019-04-19	65.83	Hanari Carnes	Rio de Janeiro	Rua do Paço

Figure 9: OutSystems Interface for database interaction (OutSystems, b)

2.3 BLOCK-BASED LANGUAGES

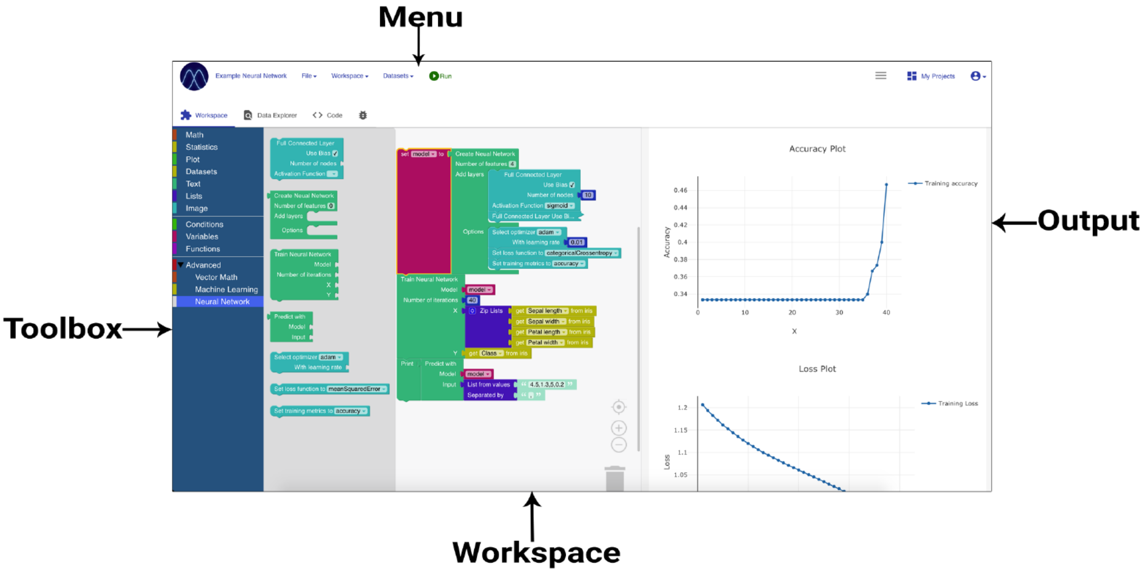
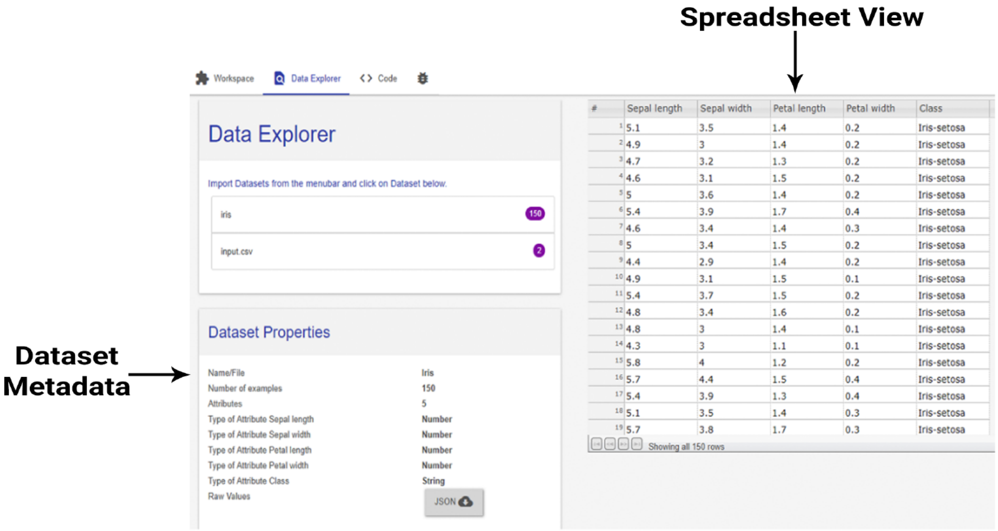


Figure 10: Milo Workspace (Rao et al., 2018)

A more recent approach to data science is block based programming. Block-based languages attempt to provide an introductory language that reduces probability of error by providing entire statements as blocks of code with very specific places which the user can edit and write their own code (Bart et al., 2017).

The main objective of Rao et al. (2018) is building a platform - Milo - that is easy to interact with for non computer science majors and would allow them to self learn concepts for data science and machine learning. Milo provides a web based visual programming environment. It is mainly divided in 2 parts: the workspace and the data explorer. Milo uses block based languages, more specifically Blockly, to build the workspace - Figure 10. In the workspace, access to building blocks is through folders which represent different block categories. In the data explorer it provides a spreadsheet view of the data and the metadata of the dataset - Figure 11.



Data Explorer

Import Datasets from the menubar and click on Dataset below.

iris	150
input.csv	2

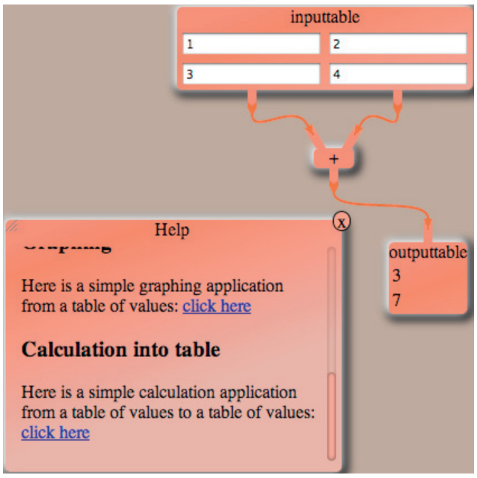
Dataset Properties

Name/File	Iris
Number of examples	150
Attributes	5
Type of Attribute Sepal length	Number
Type of Attribute Sepal width	Number
Type of Attribute Petal length	Number
Type of Attribute Petal width	Number
Type of Attribute Class	String
Raw Values	JSON

#	Sepal length	Sepal width	Petal length	Petal width	Class
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3	1.4	0.1	Iris-setosa
14	4.3	3	1.1	0.1	Iris-setosa
15	5.8	4	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa
17	5.4	3.9	1.3	0.4	Iris-setosa
18	5.1	3.5	1.4	0.3	Iris-setosa
19	5.7	3.8	1.7	0.3	Iris-setosa

Showing all 150 rows

Figure 11: Milo Data Explorer (Rao et al., 2018)



1	2
3	4

3
7

Here is a simple graphing application from a table of values: click here

Calculation into table

Here is a simple calculation application from a table of values to a table of values: click here

Figure 12: Proposed example interface by Mason (2013)

A study by Mason (2013) is also focused in a block-based language and defines itself as a "visual programming environment for sporadic users/programmers - not necessarily novice, but not classically trained". This solution is loosely based on Scratch (Resnick), with the objective of popularizing data by providing an environment with a low bar of entry for accessing, analyzing, and extracting meaning from data. This low bar of entry would be impossible to achieve with normal programming languages such as Python (van Rossum) or R (Ihaka and Gentleman) as it requires someone with a programming background and others would need to spend time to learn these programming languages which is a very challenging task (Cunha et al., 2020).

3

SOLUTION DESIGN

As mentioned in Chapter 1, the number of data science job openings is increasing, but not the number of data scientists. With this thesis we hope that by reducing the requirements for some parts of that job (by reducing the programming knowledge required) we help fill in those gaps. Thus, by accomplishing this we hope that data science becomes more accessible to the end user.

In this chapter we will explain and elaborate on our design choices and how we might expect each of them to help end users while they are using our tool. Initially we will start by our design overview where we will discuss what we want to achieve with this tool and present the main ideas for the development of the tool (Section 3.1). Following this, we will have two sections that correspond to the main uses of the tool: "Visualization of Data"(Section 3.2) and "Manipulation of Data"(Section 3.3). In each of these sections we will look at the general problem and further specify in the respective subsections. Lastly we will look at how everything comes together to form the tool: Data Cleaning for All (DCA) (Section 3.4).

3.1 DESIGN OVERVIEW

Currently, even though there are tools that attempt to remove the need for previous knowledge or background in programming, it is still Python and R that are the most used tools for Data Science (DS) (kag). This means that the tools that we have studied in Section 2 may not be enough to allow users with no background in those areas to do the necessary tasks. This seems to be true as many of these tools still require a background in computer science as we mentioned in Section 2. These tools require the user to abstract from the data and use control flows, block-based languages or others to manipulate data. Tools are critical to the data scientists effectiveness (Pereira et al., 2020) and we believe that abstracting from data is not the ideal way and may be a detriment in order to involve non programmers with data science, and that a tool that would allow for direct manipulation of the data would be more suitable. With this solution we want to as much as possible remove the need for programming or abstraction and provide users with a direct way of managing data. According to a study performed by Pereira et al. (2020) participants that regularly used programming languages reported that most of their analysis was through MS Excel since it allows for more immediate results. MS Excel provides almost no abstraction with most interactions being through direct manipulation which seems corroborate our beliefs.

In accordance to what we have previously discussed, we believe there are several paths one may take when developing a visual environment for the manipulation of data. We have developed a prototype of a humanized data cleaning tool, termed Data Cleaning for All[1][2]. The tool we developed uses a spreadsheet like view of the data with suggestions and transformation previews in order to allow the users to interact with the data. The dataset present in the figures along this chapter represents Android smartphone usage information (Matalonga et al., 2019).

3.2 VISUALIZATION OF DATA

Problem: *Some users might not feel comfortable working with abstractions of data as it is not an easy task.*

Since we are proposing methodologies and tools for data science, it seems natural that data should be represented in a way users can actually see and manipulate it using some tabular format, e.g., resembling Excel. With this in mind we have chosen to always present the user with a spreadsheet like view of the data at all times.

3.2.1 *Transformation Previews*

Problem: *Understanding how a transformation affects the current data.*

Along chapter 2 we have seen multiple tools and explored some of the features present. One feature we have deemed as important in order to help users understand how they are affecting the data is through the use of *previews*. By presenting the user with the state before and after applying a transformation we can cull possible mistakes by the end user. Indeed, shown in Figure 13 in the bottom table, we have the original and unaltered dataset shown at all times, allowing the end user to better accompany their transformations. All such transformations would be shown and previewed in the top table in Figure 13. This side-by-side look at the dataset before and after applying changes aims to help remove a level of abstraction of how data will be changed, and directly present such actions.

3.2.2 *Confirming and Committing Changes*

Problem: *The user might add inconsistencies or errors into the data by mistake.*

In order to commit the changes currently applied to the *Preview dataset* the user must use the buttons between both tables (*Apply to Data* & *Reset Changes* - Figure 14). The button *Apply to Data* commits the changes from the *Preview dataset* to the *Original dataset*. The other button, *Reset Changes*, does the opposite and makes the *Preview dataset* equal to the *Original dataset*.

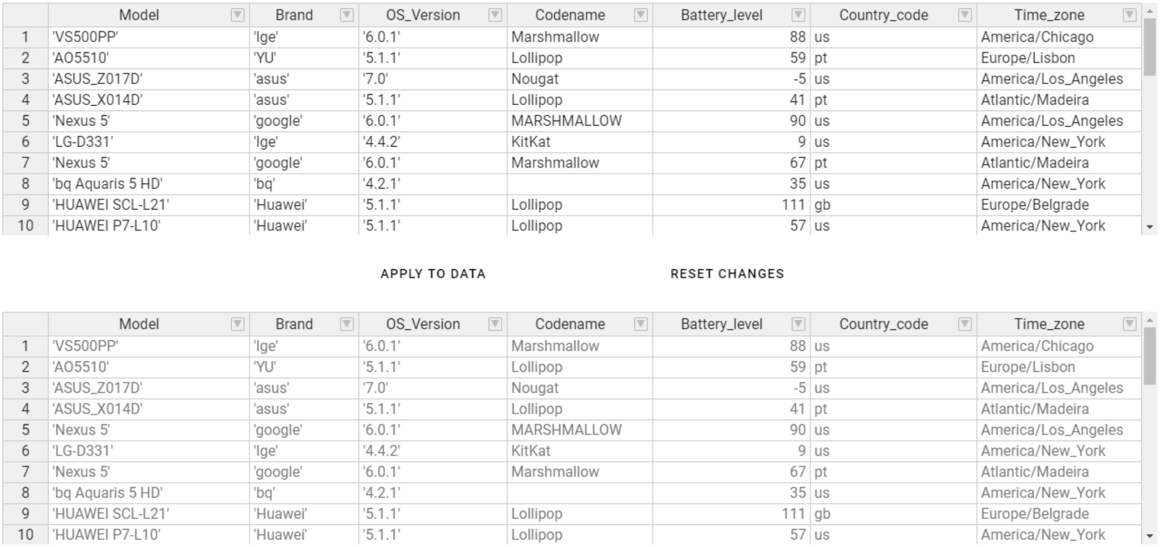
These buttons present the user with the opportunity to double check their desired changes instead of automatically applying them to the *Original dataset*. With this if they did not work as intended the user can always just reset the current transformations to the *Preview dataset*.

3.2.3 *Statistics*

Problem: *Difficulty identifying problems/inconsistencies in the data or just understanding what is in the data.*

[1] Data Cleaning for All code can be found at https://github.com/Zamreg/DeployDCA.

[2] Data Cleaning for All website can be found at https://data-cleaning-for-all.herokuapp.com/.



	Model	Brand	OS_Version	Codename	Battery_level	Country_code	Time_zone
1	'VS500PP'	'lge'	'6.0.1'	Marshmallow	88	us	America/Chicago
2	'AO5510'	'YU'	'5.1.1'	Lollipop	59	pt	Europe/Lisbon
3	'ASUS_Z017D'	'asus'	'7.0'	Nougat	-5	us	America/Los_Angeles
4	'ASUS_X014D'	'asus'	'5.1.1'	Lollipop	41	pt	Atlantic/Madeira
5	'Nexus 5'	'google'	'6.0.1'	MARSHMALLOW	90	us	America/Los_Angeles
6	'LG-D331'	'lge'	'4.4.2'	KitKat	9	us	America/New_York
7	'Nexus 5'	'google'	'6.0.1'	Marshmallow	67	pt	Atlantic/Madeira
8	'bq Aquaris 5 HD'	'bq'	'4.2.1'		35	us	America/New_York
9	'HUAWEI SCL-L21'	'Huawei'	'5.1.1'	Lollipop	111	gb	Europe/Belgrade
10	'HUAWEI P7-L10'	'Huawei'	'5.1.1'	Lollipop	57	us	America/New_York

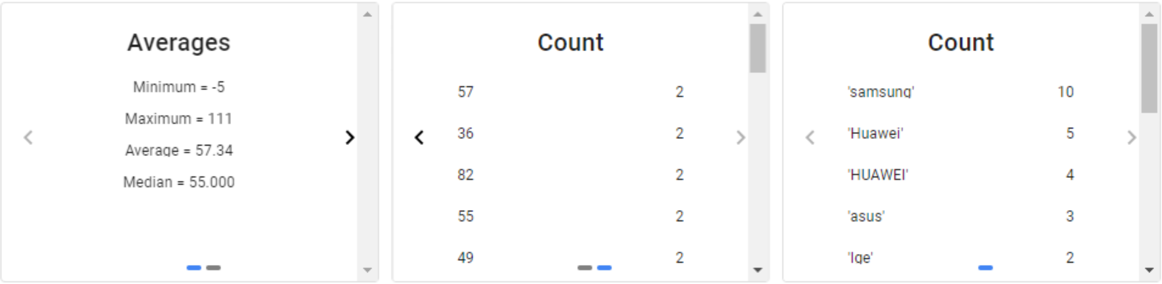
	Model	Brand	OS_Version	Codename	Battery_level	Country_code	Time_zone
1	'VS500PP'	'lge'	'6.0.1'	Marshmallow	88	us	America/Chicago
2	'AO5510'	'YU'	'5.1.1'	Lollipop	59	pt	Europe/Lisbon
3	'ASUS_Z017D'	'asus'	'7.0'	Nougat	-5	us	America/Los_Angeles
4	'ASUS_X014D'	'asus'	'5.1.1'	Lollipop	41	pt	Atlantic/Madeira
5	'Nexus 5'	'google'	'6.0.1'	MARSHMALLOW	90	us	America/Los_Angeles
6	'LG-D331'	'lge'	'4.4.2'	KitKat	9	us	America/New_York
7	'Nexus 5'	'google'	'6.0.1'	Marshmallow	67	pt	Atlantic/Madeira
8	'bq Aquaris 5 HD'	'bq'	'4.2.1'		35	us	America/New_York
9	'HUAWEI SCL-L21'	'Huawei'	'5.1.1'	Lollipop	111	gb	Europe/Belgrade
10	'HUAWEI P7-L10'	'Huawei'	'5.1.1'	Lollipop	57	us	America/New_York

Figure 13: Preview and Original Tables

Figure 14: Buttons to Commit and Reset Changes

Statistics are an important part when performing data cleaning. They have a major impact on the ease of finding *dirty* entries in the data. As such we are presenting them together with the data instead of presenting it on a separate tab. This maintains our approach of presenting everything important to the user upfront.

Statistic calculations depend on the type of data present in the column the user has selected at a particular moment. Numeric columns have the list of unique values and how many times each appears and the statistical values (minimum, maximum, average and median). Meanwhile on text columns only the count of unique values is present. In Figure 15 we see examples of the statistics mentioned.



Averages

Minimum = -5
Maximum = 111
Average = 57.34
Median = 55.000

Count

57	2
36	2
82	2
55	2
49	2

Count

'samsung'	10
'Huawei'	5
'HUAWEI'	4
'asus'	3
'lge'	2

Figure 15: Examples of Statistics Present - Averages, Numeric Count, Text Count

3.3 MANIPULATION OF DATA

Problem: *How to approach a given problem in DC? Some users might struggle finding out the correct approach to a problem.*

In previous sections we have discussed transformations, however we did not discuss how those transformations happen. The manipulation of data - transformations - are done through either direct manipulation of data or by using suggestions we present through cards. These alternatives provide the user with different methods to approach a problem. With Suggestions we allow the user to deal with general problems that the tool can find using data inference. Meanwhile using direct data manipulation we present an option for more specific problems such as changing a specific value in a cell or removing a column.

3.3.1 *Using Suggestions*

Problem: *Errors, such as multiple values out of bounds or the presence of null/empty cells, that span across multiple rows are sometimes difficult to correct.*

The main way a user will interact with the data in this tool will be through the *Suggestion Cards*. Similar to the statistics, depending on some factors in the column, we present the user with relevant suggestions to the data present. Currently the suggestions depend on the presence of empty cells/*null* values or the type of data in each column, but could be expanded in the future in order to fit smarter suggestions.

Remove or Replace Null Values

This suggestion is used to remove the row where a null value is present or to replace it by an user input. It should only be visible when the current column has an empty cell/null value present.

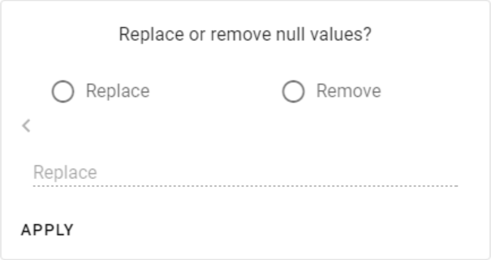


Figure 16: Remove or Replace Null Values

Find Value and Replace it

This suggestion is used to replace a specific value for another. It is present as a suggestion in every non numeric column.

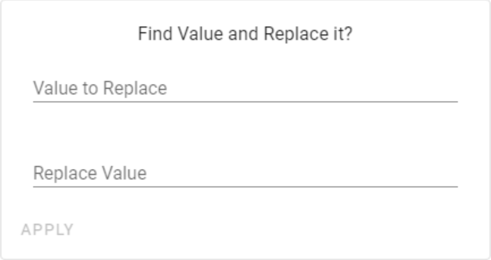


Figure 17: Find Value and Replace it

Split on First Instance

This suggestion is used to split a column into two new columns by a specific character. The old column is removed and 2 new columns are formed. It is present as a suggestion in every non numeric column.

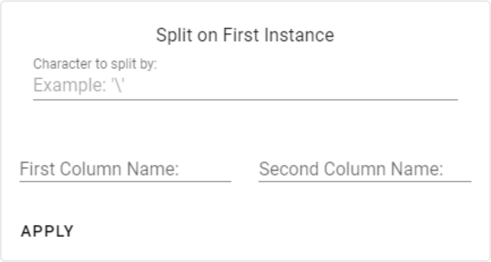


Figure 18: Split on First Instance

Find and Remove Matches - Text

This suggestion is used to filter the values of a column, removing those that match the condition chosen. It is present as a suggestion in every non numeric column. In Figure 19 we provide an example with the condition "Is Equal To".

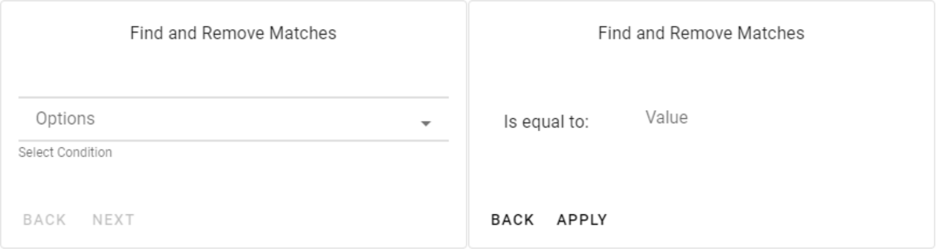


Figure 19: Find and Remove Matches - Is Equal To

Normalize String Case

This suggestion is used to normalize the case of every entry in the column. It is present as a suggestion in every non numeric column.

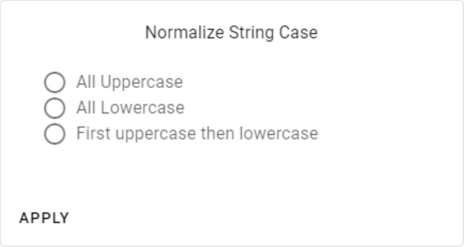


Figure 20: Normalize String Case

Define Bounds and Remove Outliers

This suggestion is used to remove entries outside of a certain interval of values in the column. It is present as a suggestion in every numeric column.

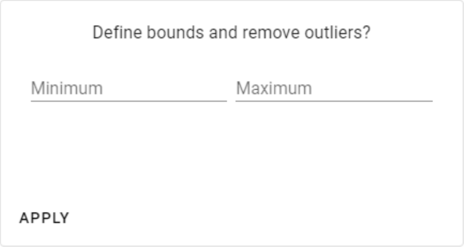


Figure 21: Define Bounds and Remove Outliers

Find and Remove Matches - Numeric

This suggestion is used to filter the values of a column, removing those that match the condition chosen. It is present as a suggestion in every numeric column. In Figure 22 we provide an example with the condition ”Greater than or equal to”.

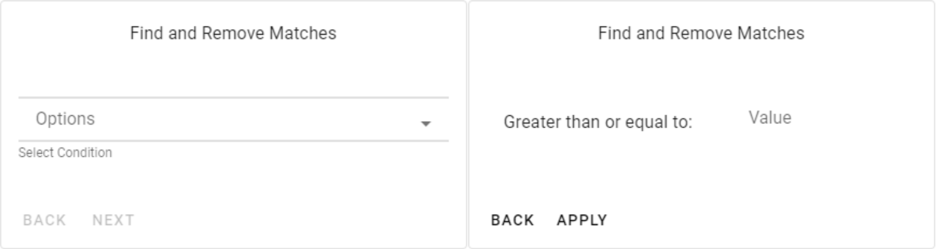
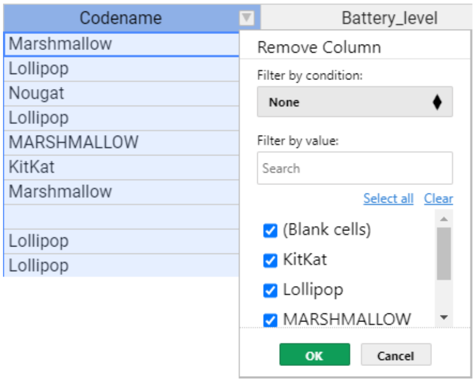


Figure 22: Find and Remove Matches - Greater than or equal to

3.3.2 *Direct Manipulation of Data*

Problem: *Spotting an error in a cell in the spreadsheet data might be hard to correct using more generic means that affect the whole column/table.*



Codename	Battery_level
Marshmallow	
Lollipop	
Nougat	
Lollipop	
MARSHMALLOW	
KitKat	
Marshmallow	
Lollipop	
Lollipop	

Filter by condition: None

Filter by value: Search

☑ (Blank cells)
☑ KitKat
☑ Lollipop
☑ MARSHMALLOW

Figure 23: Column Dropdown Menu

Another possible way to interact with data, instead of using the available suggestions, is through the direct manipulation of data. This includes directly updating values and removing whole columns. Direct manipulation of data is done through the use of the *Preview Table*, to do so it is the same process as in a spreadsheet, double click a cell and change its value. Each column has its own drop down menu beside its header, visible in Figure 23. Through the drop down menu we can remove the column or instead apply a visual filter. The visual filter has options similar to the ones present in the suggestion ”Find and Remove Matches”, however it does not change the data itself, only changes which rows are visible.

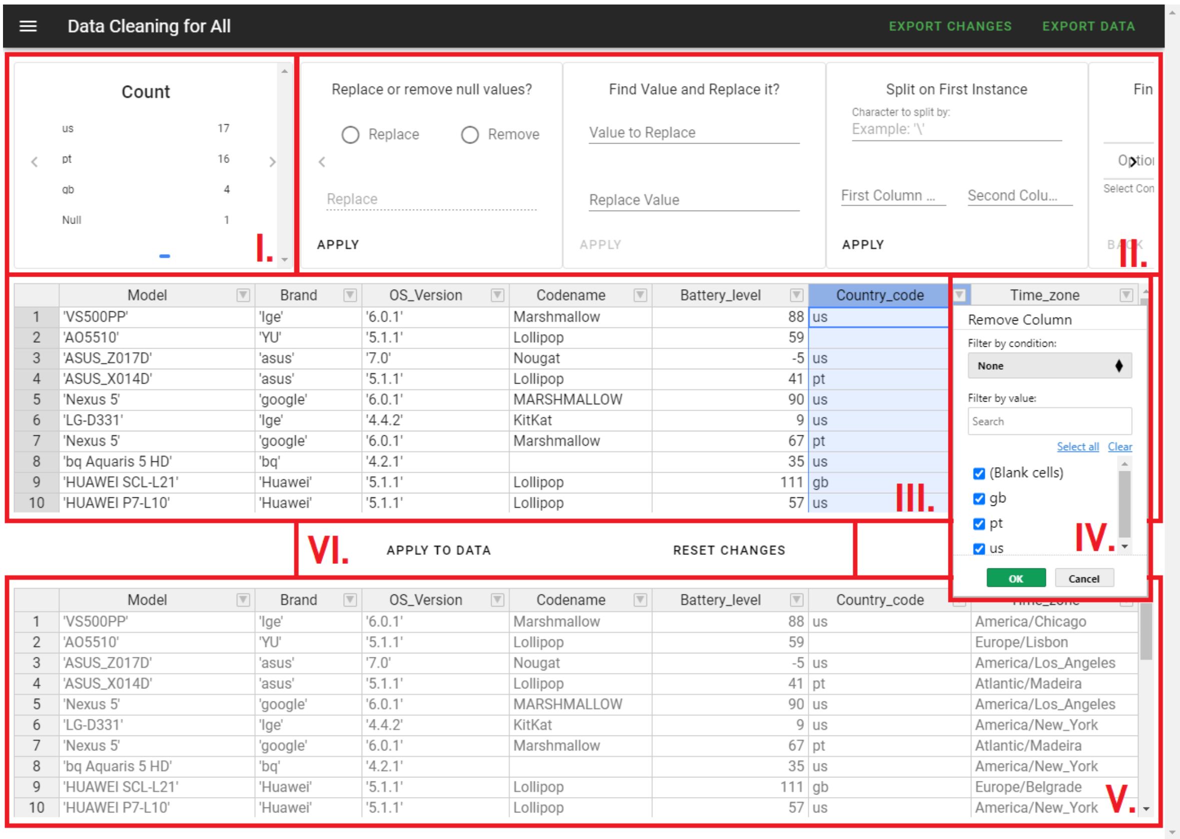
3.4 INTERFACE OVERVIEW

When we put everything mentioned in this chapter together we get the interface presented in Figure 24.

This is the main hub for everything the user will perform using the tool developed - DCA. Here the user will have everything important visible at all times. Indeed, shown in Figure 24 - V (*Original dataset*), we have the original and unaltered dataset shown at all times, allowing the end user to better accompany the transformations. All such transformations would be shown and previewed in Figure 24 - III (*Preview dataset*). This side-by-side look at the dataset before and after applying changes aims to help remove a level of abstraction of how data will be changed, and directly present such actions. In order to commit the changes from the *Preview dataset* to the *Original dataset* the user will use the *actions* present in Figure 24 - VI.

At any point, the user may directly manipulate the data within the *Preview dataset*, such as updating cell values, or through a drop down menu (as shown in Figure 24 - IV) to remove a column or applying a visual filter to the data on a specific column. For many operations related to data cleaning/curation (Muller et al., 2019) this should be sufficient.

When the user selects one specific column, a *statistics card* is displayed in order to help summarize the contents of the chosen column. An example is shown in Figure 24 - I, where the Country_code



Count

us	17
pt	16
gb	4
Null	1

☑ (Blank cells)
☑ gb
☑ pt
☑ us

	Model	Brand	OS_Version	Codename	Battery_level	Country_code	Time_zone
1	'VS500PP'	'lge'	'6.0.1'	Marshmallow	88	us	
2	'AO5510'	'YU'	'5.1.1'	Lollipop	59		
3	'ASUS_Z017D'	'asus'	'7.0'	Nougat	-5	us	
4	'ASUS_X014D'	'asus'	'5.1.1'	Lollipop	41	pt	
5	'Nexus 5'	'google'	'6.0.1'	MARSHMALLOW	90	us	
6	'LG-D331'	'lge'	'4.4.2'	KitKat	9	us	
7	'Nexus 5'	'google'	'6.0.1'	Marshmallow	67	pt	
8	'bq Aquaris 5 HD'	'bq'	'4.2.1'		35	us	
9	'HUAWEI SCL-L21'	'Huawei'	'5.1.1'	Lollipop	111	gb	
10	'HUAWEI P7-L10'	'Huawei'	'5.1.1'	Lollipop	57	us	

	Model	Brand	OS_Version	Codename	Battery_level	Country_code	Time_zone
1	'VS500PP'	'lge'	'6.0.1'	Marshmallow	88	us	America/Chicago
2	'AO5510'	'YU'	'5.1.1'	Lollipop	59		Europe/Lisbon
3	'ASUS_Z017D'	'asus'	'7.0'	Nougat	-5	us	America/Los_Angeles
4	'ASUS_X014D'	'asus'	'5.1.1'	Lollipop	41	pt	Atlantic/Madeira
5	'Nexus 5'	'google'	'6.0.1'	MARSHMALLOW	90	us	America/Los_Angeles
6	'LG-D331'	'lge'	'4.4.2'	KitKat	9	us	America/New_York
7	'Nexus 5'	'google'	'6.0.1'	Marshmallow	67	pt	Atlantic/Madeira
8	'bq Aquaris 5 HD'	'bq'	'4.2.1'		35	us	America/New_York
9	'HUAWEI SCL-L21'	'Huawei'	'5.1.1'	Lollipop	111	gb	Europe/Belgrade
10	'HUAWEI P7-L10'	'Huawei'	'5.1.1'	Lollipop	57	us	America/New_York

Figure 24: DCA Labeled Interface

column is selected and a *statistics card* detailing the different data entries (and their quantification) is shown. In addition to displaying a *statistics card*, a collection of *suggestion cards* are automatically displayed (shown in Figure 24 - II), where each presents a data transformation action, based on the statistics and data inference. Following our example, the system detects the presence of null or empty values, and suggests either replacing them with a new value or removing such data entries.

Such *statistic cards* and *suggestion cards* aim to remove another layer of complexity in data cleaning by automatically presenting common statistical information which users otherwise have to calculate, and by suggesting transformations based on their data. In both cases, the user would have to resort to either programming or using complex tools to gather the statistical information and apply their transformation.

4

DEVELOPMENT OF DATA CLEANING FOR ALL

As mentioned in Chapter 1, data is becoming more important with each passing year. As such, making tools which allow the manipulating and analysis of data available to a broader audience is important. Also our intent is to develop an open source tool that can be then looked at by the scientific community and further extended and expanded upon.

In this chapter we will discuss our approach and how we are developing a tool that is easier for end users to understand and use.

4.1 OVERVIEW

We have developed our solution as a web application since it is something readily accessible to most people.

According to the design feature we defined in the previous section, we have opted to develop a tool with a tabular interface and use suggestions in order to help end users participate in DS. We developed DCA with JavaScript using the Vue.js[1] framework with Vuetify[2]. Due to our focus on human interaction and not so much on performance, we have replaced a separate backend with the Vuex library, a state management library for Vue.js applications.

As we see in Figure 25 we have 2 main pages: home and import. Import serves a single purpose; importing data, which will then be sent to Vuex. Home is where the users will perform all DC tasks. It is divided in 3 main components: the suggestion bar and 2 tables. The suggestion bar includes 2 new components: the statistics and a the suggestions. The statistics component gets each columns data from Vuex and then calculates the statistics to display. Suggestions correspond to the main way the user

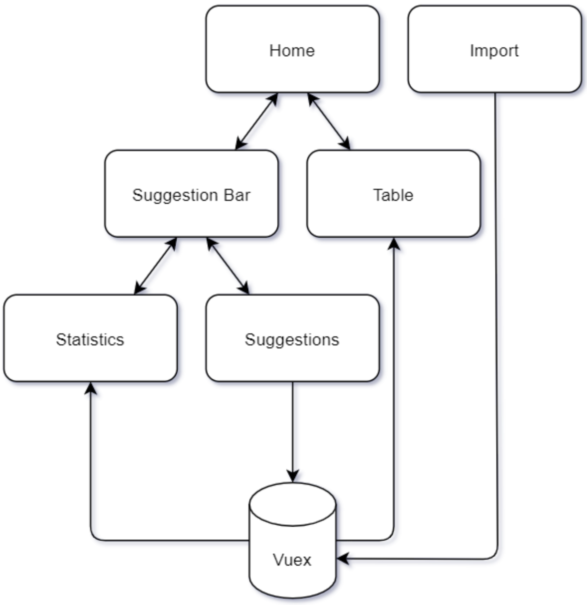


Figure 25: Component Architecture Overview

[1] Found at: https://vuejs.org/

[2] Fount at: https://vuetifyjs.com/en/

will interact with data and they communicate with Vuex through actions. The two tables correspond to the preview table and original table discussed in the previous section. Each table gets the data they will display from the state in Vuex.

4.2 Backend - Vuex Store

The Vuex store runs in the server and manages the state of the application. It keeps track of the current and past states of the data. The Vuex store is divided in a few sections: state, getters, mutations and actions.

4.2.1 *State*

The state of our store is as follows:

```
state:{
    changeCounter:0,
    colHeaders:[],
    colHeaders2: [],
    columns:[],
    columns2:[],
    trans: [],
    finalTrans:[],
    data:[],
    data2:[],
    dataHistory:[]
}
```

Listing 4.1: Store State

Each variable in the state has its own purpose. The changeCounter, as its name suggests, serves as the key to render some of our components. This means that every time the value changes the components it keys are re-rendered. There are variables for the state of our *Preview Table* and the *Original Table*. Variables corresponding to the *Original Table* have the numeral 2 at the end of its name as it is the second table (Example: data is for the *Preview Table* and data2 is for the *Original Table*). The rest of the variables are as follows:

- The colHeaders has the name for each column.
- The columns has the metadata such as type of data on the column (string or numeric)
- The trans keeps track of all transformations applied to the *Preview Table*
- The finalTrans keeps track of all transformations applied to the *Original Table*
- The data is the current state of the data we're cleaning
- The dataHistory keeps track of the all the previous states of the data

4.2.2 *Getters*

The store getters are computed variables calculated through the variables present in the state. Simple getters can give us the number of columns in the table (Listing 4.2), more complex ones can return an array with all the row indexes where a null value is present (Listing 4.3).

```
getNumberOfCols: (state) => {
  return state.colHeaders.length
}
```

Listing 4.2: Getter - Get Number of Columns

```
getNulls: (state) => (col) => {//return row index where null values appear
  var v = []
  var i = 0
  while ( i !=-1){
    i=˙.findIndex(state.data,function(array){
      if (array[col] == null && array[col] == '') return true
      else return false
    })
    v.push(i)
  }
  return v
}
```

Listing 4.3: Getter - Get Null Rows

4.2.3 *Mutations*

Mutations are synchronous operations that mutate the state. Here is where all DC operations are executed. Since we are working with arrays and arrays of arrays in the state we are using the Lodash library in order to manipulate data and also help with performance. For example:

```
removeAboveEq (state, payload) {
  ˙.remove(state.data, function(array){
    if (parseFloat(array[payload.col]) >= payload.val) return true
    else return false
  })
  state.dataHistory.push(state.data)
  state.changeCounter++
}
```

Listing 4.4: Mutations - Remove Above or Equal

4.2.4 *Actions*

Actions are what our components will call in order to change the data. Components will call an action which will then call a mutation. Actions can have computing tasks before calling a mutation, they are not used exclusively to call mutations. For example:

```
nullFilter (state,payload) {
  state.commit('commitTrans',payload)
  switch(payload.job){
    case "Is empty":
      state.commit('removeNull',payload)
      break;
    case "Is not empty":
      state.commit('keepNull',payload)
      break;
    default: break;
  }
}
```

Listing 4.5: Actions - Null Filter

4.3 IMPORTING DATA

Importing data is done through accessing https://data-cleaning-for-all.herokuapp.com/import or through clicking on Import File in the navigation drawer. The user will be presented with what is seen in Figure 26.

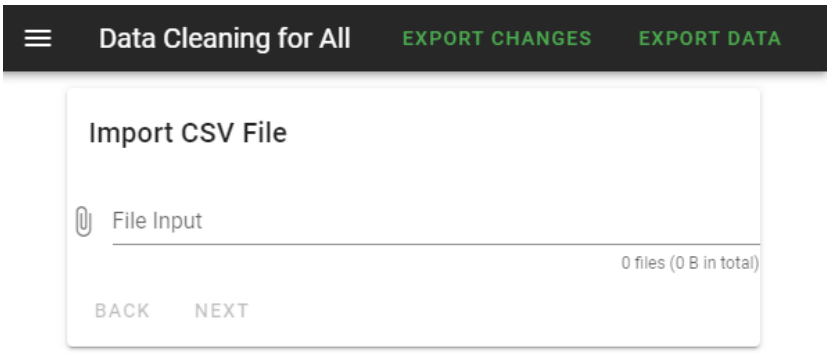


Figure 26: DCA Import File Interface

Once a file is selected we use the vue-papa-parse package, which is a wrapper for the Papa Parse[3] built for Vue.js. The parsing of data from the file is done according to the Listing 4.6. As we see, when papa parse finishes it saves the result on the local variable data.

```
getFiles: function(){
```

[3] Found at: https://www.papaparse.com/

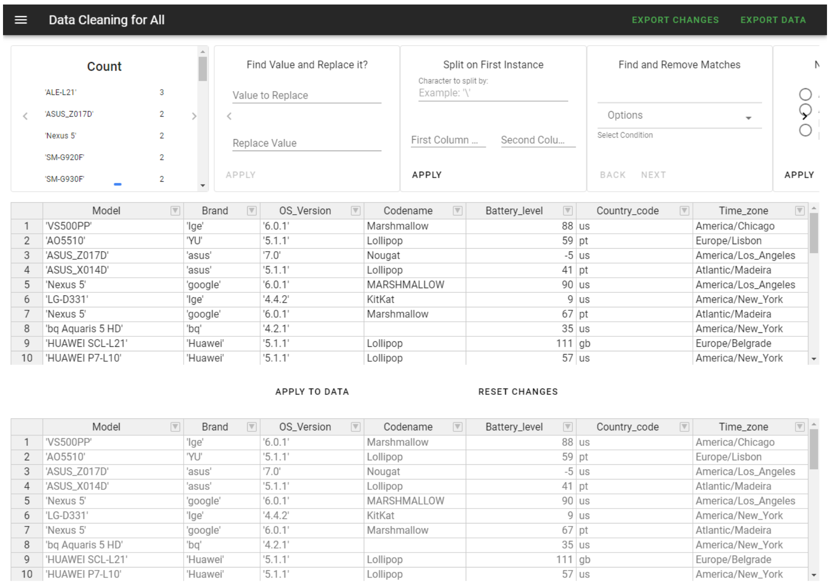
```
      if(this.files != null){
        this.$papa.parse(this.files[0],{
          complete: (result) =¿ {
            this.data = ˙.cloneDeep(result.data)
            this.update++
          }
        })
      }
    },
```

Listing 4.6: Import - Get Files

Once all the above is done the data is then converted in order to fit our needs by defining headers and removing quotation marks that serve as string separators.

4.4 Data Cleaning for All (DCA) home

The home page for the application is divided in 4 main parts: the *Suggestion Bar*, the *Preview Table*, the actions and the *Original Table*. The main page is represented in Figure 27.



Count

'ALE-L21'	3
'ASUS_Z017D'	2
'Nexus 5'	2
'SM-G920F'	2
'SM-G930F'	2

	Model	Brand	OS_Version	Codename	Battery_level	Country_code	Time_zone
1	'VS500PP'	'lge'	'6.0.1'	Marshmallow	88	us	America/Chicago
2	'AO5510'	'YU'	'5.1.1'	Lollipop	59	pt	Europe/Lisbon
3	'ASUS_Z017D'	'asus'	'7.0'	Nougat	-5	us	America/Los_Angeles
4	'ASUS_X014D'	'asus'	'5.1.1'	Lollipop	41	pt	Atlantic/Madeira
5	'Nexus 5'	'google'	'6.0.1'	MARSHMALLOW	90	us	America/Los_Angeles
6	'LG-D331'	'lge'	'4.4.2'	KitKat	9	us	America/New_York
7	'Nexus 5'	'google'	'6.0.1'	Marshmallow	67	pt	Atlantic/Madeira
8	'bq Aquaris 5 HD'	'bq'	'4.2.1'		35	us	America/New_York
9	'HUAWEI SCL-L21'	'Huawei'	'5.1.1'	Lollipop	111	gb	Europe/Belgrade
10	'HUAWEI P7-L10'	'Huawei'	'5.1.1'	Lollipop	57	us	America/New_York

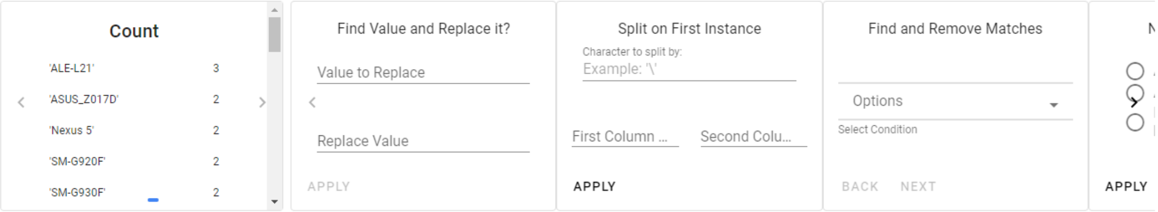
	Model	Brand	OS_Version	Codename	Battery_level	Country_code	Time_zone
1	'VS500PP'	'lge'	'6.0.1'	Marshmallow	88	us	America/Chicago
2	'AO5510'	'YU'	'5.1.1'	Lollipop	59	pt	Europe/Lisbon
3	'ASUS_Z017D'	'asus'	'7.0'	Nougat	-5	us	America/Los_Angeles
4	'ASUS_X014D'	'asus'	'5.1.1'	Lollipop	41	pt	Atlantic/Madeira
5	'Nexus 5'	'google'	'6.0.1'	MARSHMALLOW	90	us	America/Los_Angeles
6	'LG-D331'	'lge'	'4.4.2'	KitKat	9	us	America/New_York
7	'Nexus 5'	'google'	'6.0.1'	Marshmallow	67	pt	Atlantic/Madeira
8	'bq Aquaris 5 HD'	'bq'	'4.2.1'		35	us	America/New_York
9	'HUAWEI SCL-L21'	'Huawei'	'5.1.1'	Lollipop	111	gb	Europe/Belgrade
10	'HUAWEI P7-L10'	'Huawei'	'5.1.1'	Lollipop	57	us	America/New_York

Figure 27: DCA Home Page Interface

The following sections will provide a more in-depth description of each of the main parts.

4.4.1 *Suggestion Bar*

The suggestion bar corresponds to the top part of the interface shown in Figure 27, specifically what is shown in the Figure 28. It is composed by two carousel components built with Hooper[4]. The two components are the *Statistics* (on the left) and the *Suggestions* (on the right).



Count	
'ALE-L21'	3
'ASUS_Z017D'	2
'Nexus 5'	2
'SM-G920F'	2
'SM-G930F'	2

Figure 28: DCA Suggestion Bar

Statistics

The statistics are calculated when a column of the data is selected and every time there is a change to the data they are updated.

Currently we identify data as one of two types: ”Text” or ”Number”. According to the type of data in each column different statistics are calculated.

For columns with the data type ”Number” we present up front the minimum, maximum, average and median values. For these calculations we disregard empty cells or null values. In columns with either data type, ”Text” or ”Number”, we present a sorted view of the data ordered by the most common unique value to the least common and display how many times each one is represented.

Currently, in order to add a new statistic card we need to follow these steps:

1. Create the respective component.
2. In the new component, access Vuex and get the column values.
3. Calculate the statistics inside the component.
4. Add it to the statistics slider with the necessary conditions (Ex. Only display when the selected column is of type “number”).

Suggestions

The suggestions are part of a sliding carousel component which displays multiple slides at once. A suggestion is present in each slide and consists of the main way the user interacts and changes the data present in the *Preview Table*. The currently available suggestions are as follows:

- BoundsCard:
  - Suggestion card used to define bounds and remove outliers in a column with the data type ”Number”.
  - Always present if the data type is ”Number”.
- NullCard:
  - Suggestion card used if there’s an empty cell present in the currently select column.

[4] Found at: https://baianat.github.io/hooper/

  - Present if column contains an empty cell.
- FindReplace:
  - Suggestion card used to replace a unique value for a new value.
  - Always present if the data type is ”Text”.
- SplitCard:
  - Suggestion card used to split a column into 2 new ones and removes the old column.
  - Always present if the data type is ”Text”.
- FilterNumericCard:
  - Suggestion card used to filter values in a numeric column by removing all matches to the applied filter.
  - Always present if the data type is ”Number”.
- FilterTextCard:
  - Suggestion card used to filter values in a text column by removing all matches to the applied filter.
  - Always present if the data type is ”Text”.
- CaseCard:
  - Suggestion card change the case of the current column.
  - Always present if the data type is ”Text”.

Similar to the process of adding new statistics, in order to add a new suggestion we follow a similar process:

1. Create the respective component.
2. Add the new component to the suggestion slider.
3. Add the condition for the new suggestion (Ex. “There are letters in a column that is mostly numbers”).
4. Create a method that represents the intended condition.

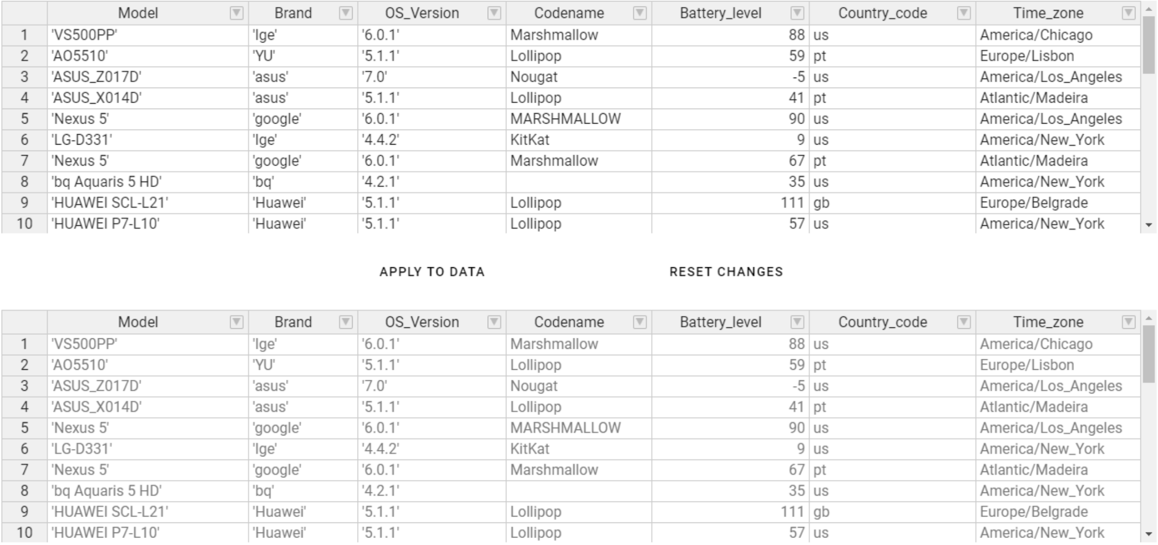
4.4.2 *Tables and Actions*

The Home view presents apart from the *Suggestion Bar* 2 similar tables, the *Preview Table* and the *Original Table*. Both are present in order to provide the user with a quick way to compare and see how his actions affect the data. The data tables can be seen in Figure 29, with the top one referring to the *Preview Table* and the bottom one referring to the *Original Table*. Between the tables we see the actions to apply or reset the current changes.

Both tables were built with the Handsontable[5] component. Handsontable is ”a JavaScript data grid that feels like a spreadsheet”. The Handsontable has many useful features that help with what we want to do with data, from column sorting and native filters to export to csv. The tables differ between one another as only the *Preview Table* allows for the direct manipulation of data. The only way to change the data in the *Original Table* is through the action APPLY TO DATA that can be seen in Figure 29.

The actions APPLY TO DATA and RESET CHANGES are 2 different buttons. The APPLY TO DATA functions as a way to commit the changes we’ve made to the *Preview Table* to the *Original Table*. The RESET CHANGES does the opposite and serves as a way to cancel all changes since the last time that

[5] Available at: https://handsontable.com/



	Model	Brand	OS_Version	Codename	Battery_level	Country_code	Time_zone
1	'VS500PP'	'lge'	'6.0.1'	Marshmallow	88	us	America/Chicago
2	'AO5510'	'YU'	'5.1.1'	Lollipop	59	pt	Europe/Lisbon
3	'ASUS_Z017D'	'asus'	'7.0'	Nougat	-5	us	America/Los_Angeles
4	'ASUS_X014D'	'asus'	'5.1.1'	Lollipop	41	pt	Atlantic/Madeira
5	'Nexus 5'	'google'	'6.0.1'	MARSHMALLOW	90	us	America/Los_Angeles
6	'LG-D331'	'lge'	'4.4.2'	KitKat	9	us	America/New_York
7	'Nexus 5'	'google'	'6.0.1'	Marshmallow	67	pt	Atlantic/Madeira
8	'bq Aquaris 5 HD'	'bq'	'4.2.1'		35	us	America/New_York
9	'HUAWEI SCL-L21'	'Huawei'	'5.1.1'	Lollipop	111	gb	Europe/Belgrade
10	'HUAWEI P7-L10'	'Huawei'	'5.1.1'	Lollipop	57	us	America/New_York

	Model	Brand	OS_Version	Codename	Battery_level	Country_code	Time_zone
1	'VS500PP'	'lge'	'6.0.1'	Marshmallow	88	us	America/Chicago
2	'AO5510'	'YU'	'5.1.1'	Lollipop	59	pt	Europe/Lisbon
3	'ASUS_Z017D'	'asus'	'7.0'	Nougat	-5	us	America/Los_Angeles
4	'ASUS_X014D'	'asus'	'5.1.1'	Lollipop	41	pt	Atlantic/Madeira
5	'Nexus 5'	'google'	'6.0.1'	MARSHMALLOW	90	us	America/Los_Angeles
6	'LG-D331'	'lge'	'4.4.2'	KitKat	9	us	America/New_York
7	'Nexus 5'	'google'	'6.0.1'	Marshmallow	67	pt	Atlantic/Madeira
8	'bq Aquaris 5 HD'	'bq'	'4.2.1'		35	us	America/New_York
9	'HUAWEI SCL-L21'	'Huawei'	'5.1.1'	Lollipop	111	gb	Europe/Belgrade
10	'HUAWEI P7-L10'	'Huawei'	'5.1.1'	Lollipop	57	us	America/New_York

Figure 29: DCA Tables

APPLY TO DATA was executed. If we have yet to execute APPLY TO DATA once, then it reverts the data to its original state. These actions are performed through the use of the deep clone method[6] present in Lodash between the respective variables [Section 4.2.1].

4.4.3 *Navigation Bar*

The navigation bar is built with the App Bar and Navigation Drawer components found in Vuetify. In the Navigation Drawer we find the Home and Import buttons which are used to swap the view of the app from one to another. It is in the App Bar where we also find the export changes and export data buttons. These buttons are used to export the data contained in the state variable finalTrans and the data on the *Original Table* respectively.

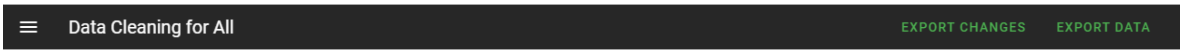


Figure 30: DCA Navigation Bar

[6] Described in: https://lodash.com/docs/4.17.15#cloneDeep

5

EMPIRICAL VALIDATION

In order to validate the developed application, we have performed an empirical study. This study was made in order to understand if the end user was faster at understanding and performing several common tasks in data cleaning.

In this chapter we will describe this study. First, in Section 5.1 we will explain the design choices. Afterwards, in Section 5.2 we will go into how the study was performed. Later on, we will look at results in Section 5.3. Following that we will take a look at our interpretation of the results and present some possible threats in Section 5.4. Finally we will provide some concluding remarks and discussion of the study in Section 5.5. For this study we have followed what was presented by Wohlin et al. (2012) towards experimentation in software engineering.

5.1 DESIGN

As stated above, our aim with this study is to evaluate the user experience while performing a set of tasks in our application and compare with the same set of tasks on a tool that is currently on the market. In Section 2 we have mentioned tools that already attempt to increase accessibility of data science software through the to reduction of the need for programming knowledge. As our aim is to do the same, we've chosen to use Tableau Prep Builder (TPB) in order to compare task execution times and its difficulty with the developed tool: Data Cleaning for All. We have chosen TPB due to the similarities present with our own tool in both the approach and presentation of data. Both use a spreadsheet-like view of the data and present statistics relevant to each column. TPB also presents some suggestions/recommendations in one of its menus. TPB is also a very popular tool according to a study performed by Thomas (2017) where 20% of the participants reported that they use Tableau. In the same study, all the tools that had a higher response percentages were programming languages (Python, R, SQL,etc.).

The target audience for this study are end users, in this case they correspond to users with little to no amount of DS experience.

In order to perform this study we've made a questionnaire which can be found in the Appendix D. The questionnaire has 8 versions which change which tool is used in each of the 3 datasets. In this study, participants performed a set of 5 tasks across 2 datasets and a final set of 1 task on a final dataset. Each tool contained its own tutorial which included similar tasks but in a different dataset.

Due to the current pandemic of Covid-19 in Portugal we've conducted this study by contacting participants through their e-mail address. We have obtained the email addresses through email distribution and word of mouth between work colleagues and shared through multiple forums/communities[1]. If there were any doubts or the participant felt they needed to be monitored we made ourselves available through a voice call in Jistsi Meet[2]. We chose Jistsi Meet due to its functionalities (private messages; sharing the screen; no registration required; no call time limit; no relevant user limit).

5.1.1 *Hypothesis*

Our aim is to prove that using Data Cleaning for All helps users perform tasks faster and that performing the same kind of tasks they feel that their difficulty is lower. However, we need to confirm this through the study, as such we can hypothesize the following:

1. Users spent less time performing the same tasks in Data Cleaning for All than when performing them in Tableau Prep Builder
2. Users felt that the tasks were easier to execute in Data Cleaning for All than when using Tableau Prep Builder

In total 2 hypothesis are being tested: H_t for the time the end user takes to perform a set of tasks and H_d for the difficulty that the user gave each task.

5.1.2 *Variables*

The independent variables are: for H_t the time the user takes to perform the tasks (Δ_t), for H_d the difficulty rating the user gave each task (d).

5.1.3 *Subjects and Objects*

The subjects for this study are end users. We want to explore results from users without prior programming and data science knowledge and from users that could need to parse data, but do not have the background in computer and data science.

The study was performed with both university students across multiple areas and also multiple professionals participated in this study such as doctors, nurses and secretaries.

There was no prior selection of participants due to there not being any heavy restrictions in place. However, we've prefaced our questionnaire with some questions related to their personal experience with computers and data science (Appendix D.1). The questionnaire is in Portuguese.

The objects of this study are 4 distinct datasets which will be further explained in section 5.1.4. One dataset is used during the tutorial for both tools, explaining how to perform tasks with each tool. Two other datasets were used with each tool in separate versions of the questionnaire, these were used to test both tools and each had a set of 5 tasks that the user had to perform. The last dataset was used at the end of the questionnaire in only one of the tools (varied between versions), this dataset only had a single task which was meant to have the users explore the tool (with the knowledge they have acquired) and clean the data as best as they could.

[1] Through Reddit communities such as: r/portugal, r/brasil and r/cienciadedados

[2] The call was accessible through the link: https://meet.jit.si/Dissertacao2020Teste2

5.1.4 *Instrumentation*

As mentioned above (Section 5.1.3) we will be using four datasets[3] in order to perform this study. The dataset used during the tutorials for both tools is named "MaxTempPerCity.csv" and consists of the highest average temperature registered in a city (Dataset_T). The two datasets used to test the tools were the Android dataset (named: "android.csv") and the Movies dataset (named: "movies.csv"). The Android dataset (Dataset_A) contained Android smartphone usage information such as battery level, OS version, model, brand etc. The Android dataset comes from the GreenHub repository (Matalonga et al., 2019). The Movies dataset (Dataset_M) contained information about a variety of movies such as their imdb score, title, color scheme, content rating. The last dataset was the IGN dataset (named: "ign-3k.csv"). The IGN dataset (Dataset_{IGN}) had information related to the review scores of a variety of video games in the IGN website [4] such as title name, review score and platform.

The study was performed through a single Google Forms questionnaire which would guide and help participants through the study. There are 8 versions of the questionnaire. The versions swap the tools and datasets used with each tool. One version of the questionnaire can be found at Appendix D. Examples:

- Version A: DCA and Dataset_M, TPB and Dataset_A, DCA and Dataset_{IGN}.
- Version B: TPB and Dataset_M, DCA and Dataset_A, DCA and Dataset_{IGN}.
- Version C: TPB and Dataset_M, DCA and Dataset_A, TPB and Dataset_{IGN}.
- Version D: TPB and Dataset_A, DCA and Dataset_M, TPB and Dataset_{IGN}.

Prior to submitting the questionnaire, participants were asked to send via email the files they created during the process.

5.1.5 *Data Collecting Procedure*

The study is divided in various sections. The questionnaire presented in the appendix D represents one of the possible 8 versions. The sections are as follows:

1. Questionnaire about Personal Info (Appendix D.1)
2. Data Cleaning for All Tutorial (Appendix D.2)
3. Movies Questionnaire (Appendix D.3)
4. Tableau Prep Builder Tutorial (Appendix D.4)
5. Android Questionnaire (Appendix D.5)
6. IGN Questionnaire (Appendix D.7)
7. System Usability Scale Questionnaire (Appears twice, once for each tool) (Appendix D.6 & D.8)

The questionnaire versions change which tool is used when and which dataset is used with each tool giving us every single possible combination and order of tool and dataset usage within the study. (As mentioned in section 5.1.4).

[3] Datasets can be accessed here: https://tinyurl.com/y6zboayu

[4] IGN website: ign.com

5.1.6 *Analysis Procedure and Evaluation of Validity*

The analysis of collected data is made possible due to the multiple versions of the questionnaire (Appendix D). In the questionnaire, for every task, the user needs to register:

- the time at which he started;
- the time at which he finished;
- the difficulty felt executing what was required for the task.

With this it is possible to compare both the difficulty (d) and the time consumption (Δ_t) of each task with both tools (DCA & TPB).

To compare the time (Δ_t) we will:

1. calculate the average time taken in each task in one of the tools;
2. calculate the average time taken for the same task in the other tool;
3. compare both the calculated averages.

To compare the difficulty (d), which was rated from 1 (very easy) to 5 (very hard), we use a similar process to the one employed to compare the time. For this, we calculate the average difficulty for a task on each tool and then compare them one another.

In order to make sure the end user could perform each task we presented multiple methods to support the participant. The participant could be accompanied and could have supervision during the whole questionnaire through a voice call on Jitsi Meet[5]. The participant also had access to tutorials on both tools where the necessary knowledge was presented.

5.2 EXECUTION

As mentioned in Section 5.1, the study was distributed through email and word of mouth between work colleagues and shared through multiple forums/communities. Due to the current pandemic the communication and supervision of the study was achieved through a voice call on Jitsi Meet accessible through the following url: https://meet.jit.si/Dissertacao2020Teste2. The call was available throughout multiple weeks where participants would be able to join.

The participants were asked to install TPB beforehand through its free trial and evaluation period, which lasted 14 days. I was present in the voice call during set periods during the day so that if someone showed up they could be supervised. If the participant chose to be supervised then when they joined the voice call they could make use of Jitsi Meet screen share feature in order for us to better understand where and what were his difficulties and clarify any doubts.

To start, participants had to fill a personal information questionnaire (Appendix D.1) where they would register their identifier, age, working area (computer science, biology, physics etc.) , if they were a student or not and their experience with both data science and computers. The identifier is a number which represents which version of the questionnaire the user had and allowed us to match the files we would receive at the end with the answers they made on the google forms questionnaire.

After registering their personal information, the participant would then be presented with a tutorial in one of the tools (TPB or DCA) where it explained how to perform certain tasks. The tutorial had its own dataset. Afterwards, the participant had to perform a set of tasks in the same tool they had just performed the tutorial with, but using a different dataset. In these tasks, the user would register

[5] The call was accessible through the link: https://meet.jit.si/Dissertacao2020Teste2.

their starting time, when they finished and the difficulty they felt for each task. Following this, the participants would repeat a tutorial using the alternative tool, and afterwards were presented new set of tasks to perform with a new dataset (See version examples in Section 5.1.4).

Lastly the participant would then have a final dataset where they were tasked to clean the data as best they could. This final task was to test what the users would be able to find when there was no specific guidance given. This final dataset cleaning could be performed in either tool, depending on the version of the questionnaire.

The participants also had to fill a questionnaire termed the "System Usability Scale" for each of the tools. They would fill this questionnaire when they were finished with a tool.

After each dataset the participants were tasked with exporting the necessary files in each tool (dataset and changes in DCA and exporting the encapsulated flow[6] in TPB). Then, before the participants were able to submit their answers, they were asked to email the files generated during the study back to us.

5.3 ANALYSIS -DESCRIPTIVE STATISTICS

In order to have an analysis of the study, we have collected the answers of 15 different people, most of them professionals in other areas such as doctors and secretaries.

5.3.1 *Subjects*

We have collected basic information about each participant namely: gender, age, student status, training area, computer experience and data science/cleaning experience. Currently, from the 15 subjects they are 54.5% Female and 45.5% Male with the following age distribution (Figure 31).

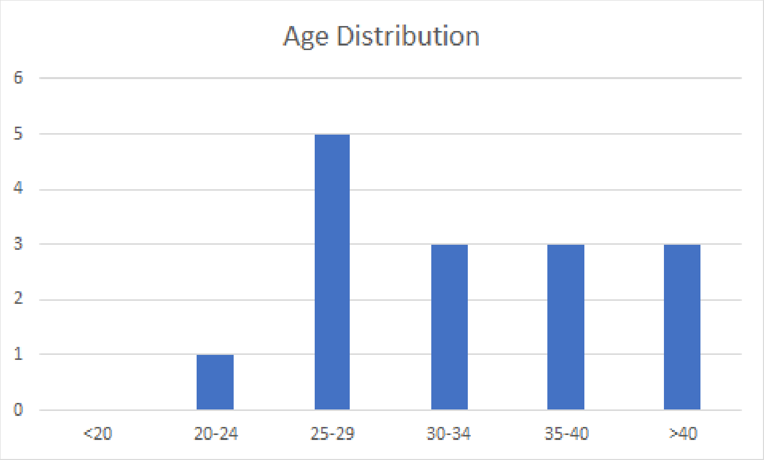


Figure 31: Subjects Age Distribution

The subjects came from multiple different training areas such as: Healthcare professionals (Doctors and Nurses), Management, Law, Biology, Food Engineering, IT & Multimedia, Science, Bioinformatics. Subjects were asked to auto-evaluate their experience experience in using a computer and their

[6] includes the dataset and all transformations

experience with DS and DC on a scale of 1 to 5 with 1 being none and 5 a professional. From the 15 subjects only 3 classified as a 3 or higher in DS/DC experience and none classified as a 5 (Figure 32).

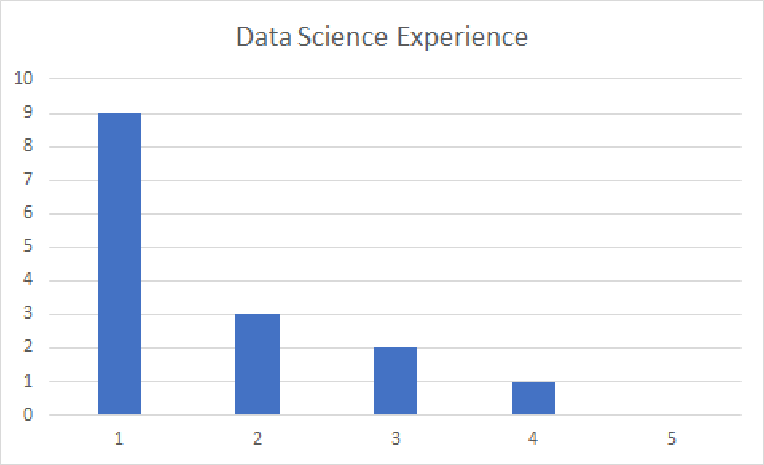


Figure 32: DS & DC Experience Distribution - 1=None; 5=Professional

More information about the subjects can be found at the Appendix A.

5.3.2 *Difficulty Grading*

For each task the user recorded their difficulty rating. The difficulty rating was from 1 to 5, with 1 being very easy and 5 very hard. We can compare the difficulty on each tool for the same task by comparing the following figures found in Appendix B:

- Figure 43-48: Android Tasks Difficulty Comparison.
- Figure 49-54: Movies Tasks Difficulty Comparison.
- Figure 55 and 56: IGN Tasks Difficulty Comparison.

In the figures mentioned above (Figures 43-56) the horizontal axis corresponds to the difficulty grade and the vertical axis corresponds to the number of answers.

With the IGN dataset we must also take into account which tasks the user performed in order to clean the data, or if any task at all was performed.

When comparing tasks we have used results from different users as a user that performs a task in one tool will not do that same task on the other tool.

When we look at the two graphs presented above (Figure 33 and 34) we see two different stories. On one hand we see in the Movies Comparison (Figure 33) that every task was harder to perform when using TPB, meanwhile on the Android dataset (Figure 34) it is more split with some tasks being considered harder in TPB while others harder on DCA.

When we look at the last graph (Figure 35) we see that users once again felt that importing data was a bit more complicated than they would wish in TPB. Meanwhile the difficulty for the last task, which consists of the user finding and cleaning ”dirty” data entries, was slightly harder on DCA.

When we look at the graphs as a whole we can see that no task was considered exceedingly difficult as only one task even touched the grade 3 for difficulty on a scale of 1 to 5.

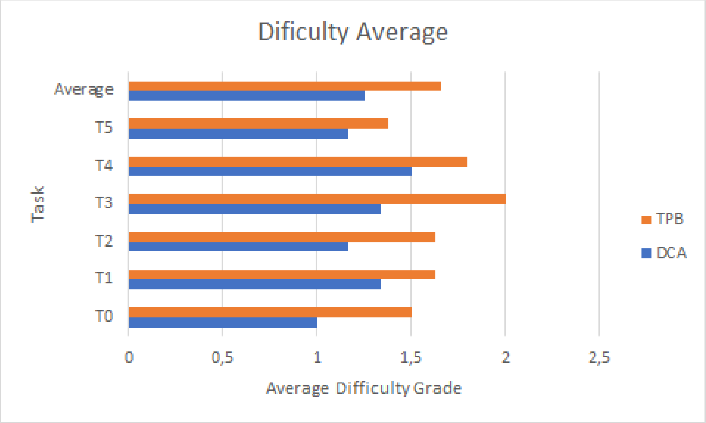


Figure 33: Movies Average Task Difficulty Comparison

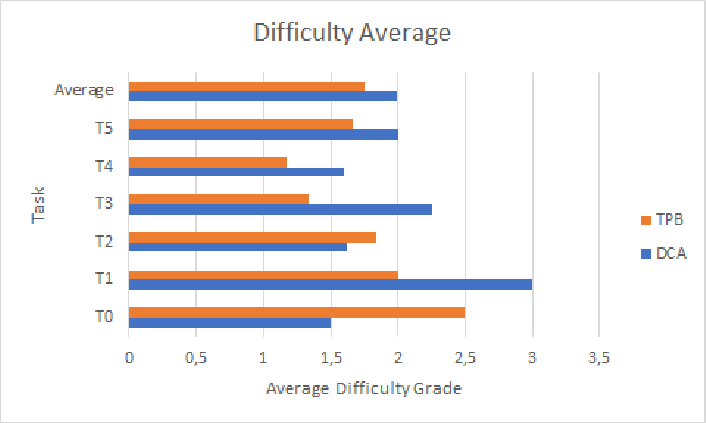


Figure 34: Android Average Task Difficulty Comparison

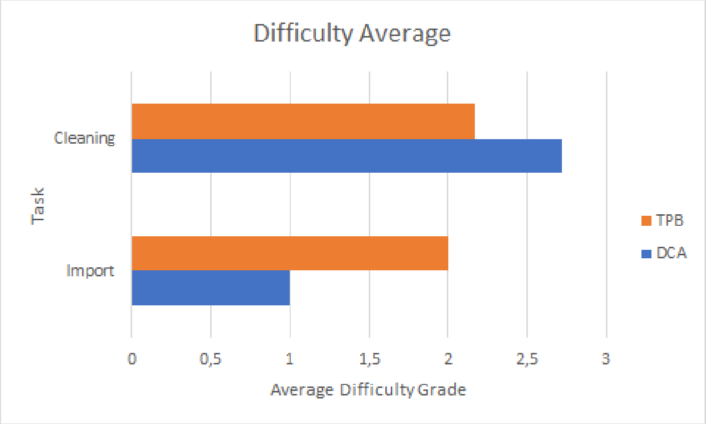


Figure 35: IGN Average Task Difficulty Comparison

5.3.3 *Time Spent*

For each task the user recorded the time when he started and finished every task. With these values we calculated the average time a user takes on each individual task in both tools. The figures in Appendix C correspond to the time each individual took in a task according to their id. In these figures (Figure 57-82) the horizontal axis corresponds to the id for a participant and the vertical axis corresponds to the time in minutes.

- Figure 57-62: DCA Android Tasks Time Distribution
- Figure 63-68: TPB Android Tasks Time Distribution
- Figure 69-74: DCA Movies Tasks Time Distribution
- Figure 75-80: TPB Movies Tasks Time Distribution
- Figure 81: DCA IGN Tasks Time Distribution
- Figure 82: TPB IGN Tasks Time Distribution

From the graphs presented in Appendix C, we can see that the time consumed for each task is similar when compared with the same task in a different tool. There are some anomalies in some answers where the user finishes a task at a set time and starts the next task before the time he previously registered. These results are not included for calculation of the average time taken in each task.

In order to understand what each task consists of, refer to Appendixes D.3, D.5 and D.7.

The following graphs (Figures 36, 37 and 38) contain information about the average time for each task and the overall average for all tasks in each tool DCA and TPB. In these figures (Figures 36-38) the horizontal axis contains the task name (T0,T1,etc.) and the vertical axis corresponds to the time in minutes.

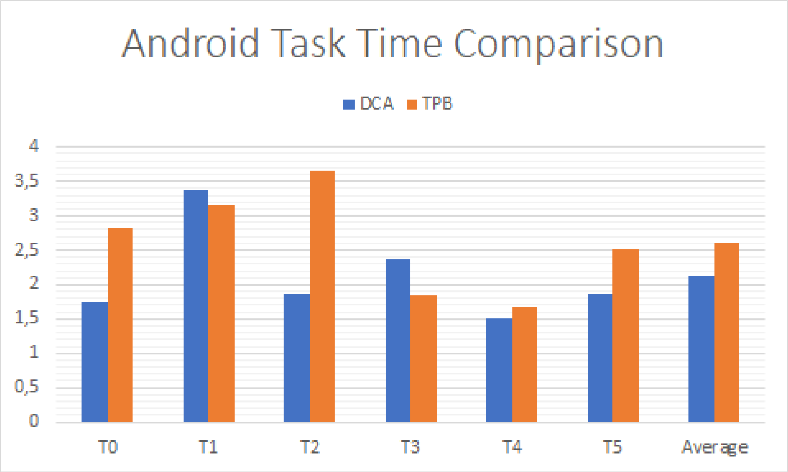


Figure 36: Android Average Task Time

When comparing the tasks present for the Android dataset (Figure 36) we can see that some tasks take almost twice as long in TPB. There is no task where TPB clearly outperforms DCA, it is always a negligible difference of less or equal to a fourth of a minute.

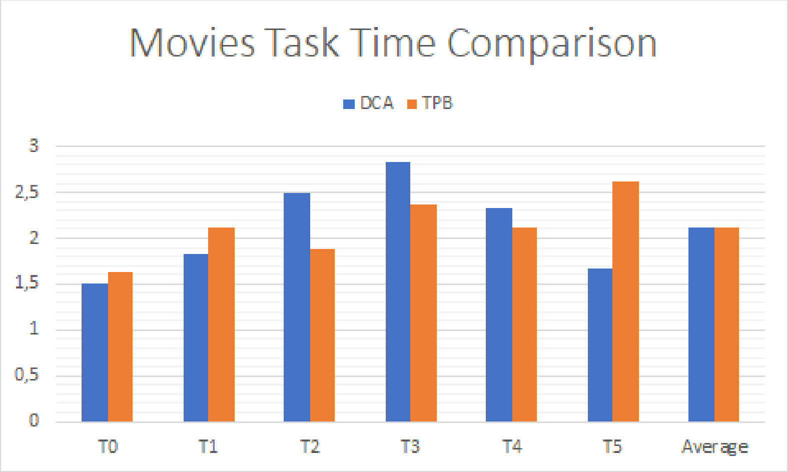


Figure 37: Movies Average Task Time

When we advance to the Movies dataset (Figure 37) we can see that both tools perform similarly as seen by the average column. Tasks 0, 1 and 5 are the ones where DCA outperforms. These tasks are where the user is tasked with importing data, removing nulls and removing a column. The others where TPB outperforms are where the user is tasked with removing negative numbers, replacing nulls and filtering values.

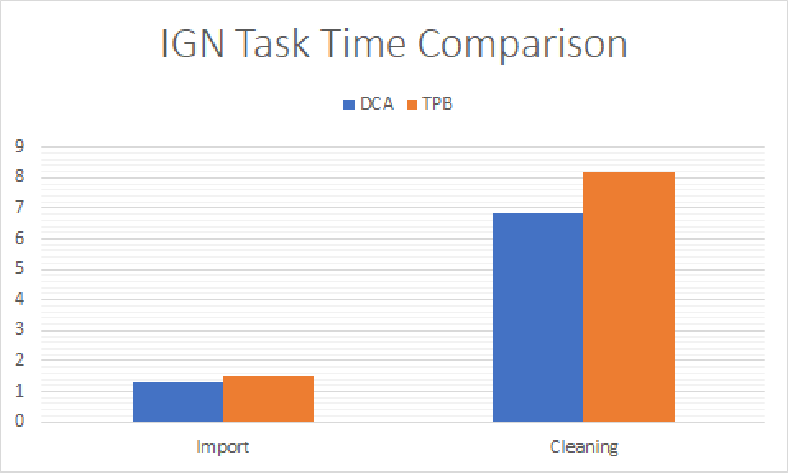


Figure 38: IGN Average Task Time

When we look at the IGN dataset (Figure 38) we see that the average times in DCA are much lower. The IGN dataset only had a single task that was meant to see if the user identified any ”dirty” data entries. With this in mind it is less relevant how much time it took but how many ”dirty” entries were found and then fixed.

Considering just the time, it seems that with the tool developed - DCA - we have made end users perform data cleaning tasks slightly faster.

5.3.4 *IGN Dataset Grading*

As mentioned before, there are no specific tasks for this dataset as there were with previous datasets. We introduced this dataset in order to try to identify which tool would allow users to identify more errors and actually correct them. We wanted users to correct 6 errors in this dataset:

1. Normalize the ”score_phrase” column;
2. Normalize the ”platform” column;
3. Define bounds for the ”score” column (minimum of 0 and a maximum 10);
4. Remove or Replace Nulls in the ”genre” column;
5. Group values with the same meaning in ”editors_choice” column;
6. Remove or Replace Nulls in the ”release_month” and ”release_day” columns.

Due to multiple issues, from participants not sending the files to others forgetting to save their changes in DCA, we could not obtain results from all participants. Only 9 participants sent the necessary files for this analysis and 5 used TPB and the other 4 used DCA.

IGN Grades		
TPB	Minimum	3.5
	Maximum	6.0
	Average	4.9
DCA	Minimum	2.0
	Maximum	4.0
	Average	3.125

Table 1: IGN Average Grades

In Table 1 we have the average grades for the dataset. The numerical value corresponds to the number of errors detected and corrected successfully. Half a point was given to users that identified a problem and tried to correct it, but did not correct it completely, for example on problem 3 only defining the minimum bound of 0.

IGN Errors Found				
Error	Tool	Corrected	Incomplete	Not Found
1.	TPB	3	0	2
	DCA	3	1	0
2.	TPB	4	0	1
	DCA	1	0	3
3.	TPB	3	2	0
	DCA	4	0	0
4.	TPB	5	0	0
	DCA	0	0	4
5.	TPB	5	0	0
	DCA	3	0	1
6.	TPB	3	1	1
	DCA	0	2	2

Table 2: IGN Errors Found

- Corrected: Found and corrected the error;
- Incomplete: Found but did not complete the correction or applied the wrong correction;
- Not Found: Did not find or found but did not attempt to correct the error.

Incomplete corrections range from the user not defining the correct bounds for error 3 or removing the entire column when they only had to remove or replace empty cells.

5.3.5 *System Usability Scale Scores*

After the user was finished with a tool they were tasked with answering 10 questions relative to that tool, the System Usability Scale (SUS). SUS is the the most used questionnaire for measuring perceptions of usability. It was released in 1986 by John Brooke and has since become an industry standard with references in over 600 publications. SUS is technology independent and has since been tested on hardware, consumer software, websites, cell-phones, IVRs and even the yellow-pages (Sauro, 2011).

The 10 questions served to evaluate the users opinion of each tool. The scale for each question is from 1 to 5, where 1 means strongly disagree and 5 means strongly agree. In Table 3 we present the mean scores each question had in each tool, these values are going to enter in the calculation of the global SUS score. The global SUS scores can be seen in Table 4.

When we compare the values for each tool in Table 3 we see an overall preference across the users towards DCA.

Afterwards we apply the calculations to measure the final score where we arrive at the final values and get a result on a scale from 0 to 100. Here the higher the score the better. In Figure 39 we display the distribution of SUS scores across 500 studies where the average SUS score is 68 (Sauro, 2011).

In Table 4, where we have the calculation of the global SUS score metrics (Minimum, Maximum and Average), we continue to see the values pointing favorably towards DCA. When we combine the results from Figure 39 and Table 4 we can see that both tools have an above average usability score in our study.

Questions	TPB Mean	DCA Mean
1.I think that I would like to use this system frequently.	3.47	3.53
2.I found the system unnecessarily complex.	2.06	1.80
3.I thought the system was easy to use.	3.73	4.07
4.I think that I would need the support of a technical person to be able to use this system.	2.47	2.07
5.I found the various functions in this system were well integrated.	4.13	4.13
6.I thought there was too much inconsistency in this system.	1.67	1.67
7.I would imagine that most people would learn to use this system very quickly.	3.80	3.93
8.I found the system very cumbersome to use.	2.20	1.73
9.I felt very confident using the system.	3.87	4.13
10.I needed to learn a lot of things before I could get going with this system.	2.47	2.27

Table 3: SUS Questions Mean Scores

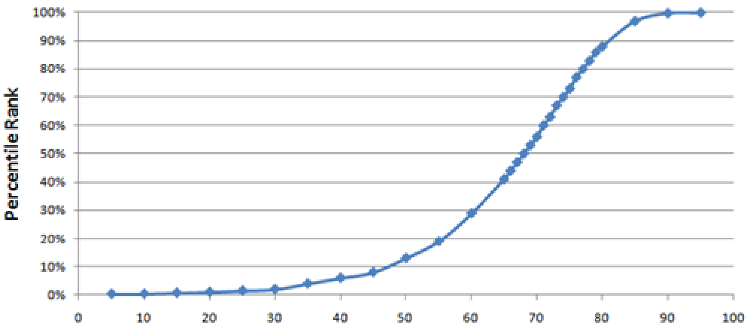


Figure 39: SUS Percentile Distribution (Sauro, 2011)

System Usability Scale Scores		
TPB	Minimum	27.5
	Maximum	92.5
	Average	70.3
DCA	Minimum	32.5
	Maximum	100
	Average	75.7

Table 4: SUS Score Metrics

We then divided the answers we got from SUS and matched them according to the participants academic background. We then calculated the average score for each background and the absolute number of participants which prefer each tool in each area (Table 5). They were divided as follows:

- Health Professionals: 3 Doctors, 2 Nurses and another Health professional.
- Science Professionals: 2 Biologists, 1 Food engineer and another science professional.
- IT and Computer Science: 1 IT and multimedia professional and 1 Bioinformatics professional.

- Other: 1 High School Graduate.

		TPB	DCA
Health Professionals	Average	66.7	69.2
	Absolute	2	3
	Total	6	
Science Professionals	Average	75	62.5
	Absolute	2	2
	Total	4	
IT and Computer Science	Average	45	90
	Absolute	0	2
	Total	2	
Law and Management	Average	86.25	95
	Absolute	0	2
	Total	2	
Other	Average	92.5	100
	Absolute	0	1
	Total	1	
All	Average	70.3	75.7
	Absolute	4	10
	Total	15	

Table 5: SUS Score Metrics by Academic Background

When looking at Table 5 we can see that only with participants from a science background does TPB have a higher average score. If we look at the absolutes then it is either even or more people favor DCA. When the total number is different from the sum of both absolutes meaning that the participants missing gave the same score to both tools.

5.4 INTERPRETATION

The results of our analysis show that using DCA makes users complete their work faster, which also improves productivity. Looking at the Android Dataset tasks we see that there is a 30 second improvement on average per task. When looking at specific tasks we see that on task 2, which consists of splitting a column into 2 new ones is almost two times faster with DCA. This seems to come from the fact that in TPB there are a lot more steps involved in order to split a column, rename new columns and remove the old column. In DCA these are done simultaneously and the user only needs to fill a single suggestion card.

When we turn the sights to the Movies Dataset we see that everything is very close. The difference for the average time per task is less than 2 seconds with DCA standing at 2 minutes and 7 seconds and the TPB at 2 minutes and 8 seconds. As we turn towards specific tasks we see that only two tasks provide a significant difference in the time consumed, tasks 2 and 5. Task 2 consists of removing negative values on a column. The gap in task 2 could be due to the user using the wrong card to perform the task in DCA as they were presented with two different suggestions "Find and Remove

Matches" and "Define Bounds and Remove Outliers". This seems to be the result of some users believing that they could use the suggestion "Define Bounds and Remove Outliers" without defining an upper bound, which is not true. However, as long as the user sets an upper bound that was high enough it could be accomplished using this suggestion. The intended approach was to use "Find and Remove Matches" with the condition "is lower" with the value 0.

As we move towards the next and final dataset, if we look at the time and difficulty grading, we can see that importing in TPB is more difficult and takes longer than the single-step process implemented in DCA. From our use of TPB this seems to be due to importing data taking more than a single step in order to reach the data cleaning environment. As far as the cleaning process we can see that although the time taken was higher in TPB we need to take into account that there was a participant that took 27 minutes in order to complete the task. If we remove the participant that took 27 minutes we arrive at an average time of 5.2 minutes taken instead of the current 8.3 minutes, a whole 3 minutes lower than the previous TPB average. This new average time is lower than the DCA average time, which sits at approximately 6.9 minutes. When we then take into considerations the errors found for this dataset, which we looked at in Section 5.3.4 we find that even though the time is lower it does not mean that the participants did not find and correct the errors with an average of 4.9 errors corrected in TPB against the 3.1 of DCA. This advantage of TPB seems to come from their statistics as they are interactive and more visually appealing than what is available in DCA. Another part that we can take into consideration here is the performance of both tools with TPB clearly outperforming DCA in this regard, which can impact the times we have obtained. However the differences seem meaningful enough that we can not attribute it only to the performance of the tools.

5.5 Discussion

The empirical study show promising results for DCA. However, due to some of the threats we have mentioned in Section 5.6 such as the reduced number of participants we can not be sure that this study is representative or that it could be replicated in other environments. Nevertheless, the results obtained, even with the concerns before, are promising. Even though there is a low number of participants they come from multiple academic backgrounds that coincide with the intended population for the study, end users.

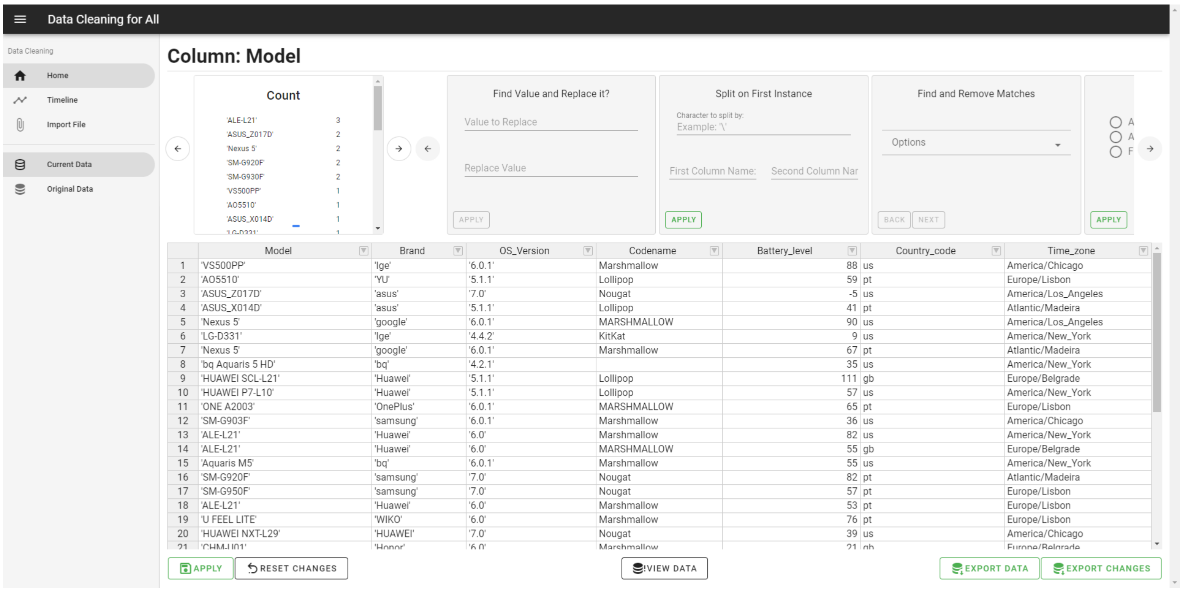
Despite the fact that most of the subjects of the study had no prior experience with DS and DC they performed the tasks without issues. As seen in Section 5.3.3, when given specific tasks the users were on average at least equal or faster while using DCA. In terms of difficulty both tools seem very comparable with mostly less than a 0.5 difference between the average scores on each task.

When we come to the IGN dataset, as we have mentioned in Section 5.4, it seems clear that TPB does a better job in terms of providing information that helps users identify and correct errors. However, with the previous results and SUS scores, it seems that the method of utilizing suggestions and presenting the users with the options available instead of using menus, as TPB does, seems to indicate that it is an approach worth considering and improving upon.

Finally looking at the SUS it seems to reinforce the previous idea that the tools are comparable. The final SUS scores and mean answers for each question even slightly favor DCA.

As a whole, this study, even with the concerns described, seems to provide favorable results towards this concept of work.

5.5.1 *DCA Iteration - Implementation of Feedback*



Column: Model

Count

'ALE-L21'	3
'ASUS_Z017D'	2
'Nexus 5'	2
'SM-G920F'	2
'SM-G930F'	2
'VS500PP'	1
'AO5510'	1
'ASUS_X014D'	1
'LG-D331'	1

	Model	Brand	OS_Version	Codename	Battery_level	Country_code	Time_zone
1	'VS500PP'	'lge'	'6.0.1'	Marshmallow	88	us	America/Chicago
2	'AO5510'	'YU'	'5.1.1'	Lollipop	59	pt	Europe/Lisbon
3	'ASUS_Z017D'	'asus'	'7.0'	Nougat	-5	us	America/Los_Angeles
4	'ASUS_X014D'	'asus'	'5.1.1'	Lollipop	41	pt	Atlantic/Madeira
5	'Nexus 5'	'google'	'6.0.1'	MARSHMALLOW	90	us	America/Los_Angeles
6	'LG-D331'	'lge'	'4.4.2'	KitKat	9	us	America/New_York
7	'Nexus 5'	'google'	'6.0.1'	Marshmallow	67	pt	Atlantic/Madeira
8	'bq Aquaris 5 HD'	'bq'	'4.2.1'		35	us	America/New_York
9	'HUAWEI SCL-L21'	'Huawei'	'5.1.1'	Lollipop	111	gb	Europe/Belgrade
10	'HUAWEI P7-L10'	'Huawei'	'5.1.1'	Lollipop	57	us	America/New_York
11	'ONE A2003'	'OnePlus'	'6.0.1'	MARSHMALLOW	65	pt	Europe/Lisbon
12	'SM-G903F'	'samsung'	'6.0.1'	Marshmallow	36	us	America/Chicago
13	'ALE-L21'	'Huawei'	'6.0'	Marshmallow	82	us	America/New_York
14	'ALE-L21'	'Huawei'	'6.0'	MARSHMALLOW	55	gb	Europe/Belgrade
15	'Aquaris M5'	'bq'	'6.0.1'	Marshmallow	55	us	America/New_York
16	'SM-G920F'	'samsung'	'7.0'	Nougat	82	pt	Atlantic/Madeira
17	'SM-G950F'	'samsung'	'7.0'	Nougat	57	pt	Europe/Lisbon
18	'ALE-L21'	'Huawei'	'6.0'	Marshmallow	53	pt	Europe/Lisbon
19	'U FEEL LITE'	'WIKO'	'6.0'	Marshmallow	76	pt	Europe/Lisbon
20	'HUAWEI NXT-L29'	'HUAWEI'	'7.0'	Nougat	39	us	America/Chicago
21	'CHM-U01'	'Honor'	'6.0'	Marshmallow	21	gb	Europe/Belgrade

Figure 40: DCA Iteration Interface

After conversations with the participants and after analyzing the results it was clear that some improvements could be made to our interface. Figure 40 showcases the iteration developed while taking into consideration the feedback received. This version is not currently available on the website.

One of the most received criticism and things noticed during some of the sessions was that the user would sometimes not know in which column they were working on. This stems from the fact that previously once the user clicked anywhere outside the table there was no indication of which column was selected, so we have added a label at the top of the page which indicates the selected column.

Another issue that affected some participants was being able to identify what were suggestion cards and statistic cards. This came from the fact that both cards were extremely similar. As such we have changed suggestion cards, both their background color and the buttons, to add a differentiating factor between them. Also related to the cards, there was also the fact that the buttons that allowed to change slides were sometimes obscured by text and hard to click and specially on the statistics messed with the scrolling inside the card. In order to fix this we have changed them and placed them outside each component.

Moving into the tables the only issue was that users did not at all use the original dataset table (Figure 24 - V). Since this is the case we have moved the table to a new page and increased the space for the preview dataset table (Figure 24 - III). The preview dataset table now occupies the space that both tables used to occupy, this also allows the user to see more data entries at once.

Users also spent some time looking for certain actions as they were not always easy to locate, as some were in the Navigation Bar others in between tables. In order to make the actions easier to identify we have moved all of them to the bottom of the page and changed some of the button colors. We have also added a button which points towards the new page that displays what was the original dataset table.

With this iteration we have mainly focused on user interaction problems and did not implement any additional features.

5.6 THREATS TO VALIDITY

The goal of this study was to see if end users could efficiently use DCA and perform DC tasks when compared to other in-market solutions. Multiple threats to validity exist and they were analyzed and divided into four categories (Wohlin et al., 2012)].

5.6.1 *Internal Validity*

Testing."If the test is repeated, the subjects may respond differently at different times since they know how the test is conducted. If there is a need for familiarization to the tests, it is important that the results of the test are not fed back to the subject, in order not to support unintended learning" (Wohlin et al., 2012). Even though we have mentioned that the same tasks are performed in both tools we are comparing, this threat is not an issue as they are not performed by the same users. Even though users worked with both tools, the same user does not perform the same tasks in both tools. In another words if "User A" performed "Task A" in "Tool A" he would not perform "Task A" in "Tool B".

As the study was conducted online we have created different versions of the questionnaire so that if there was a learning effect it would affect both tools. As the questions for each dataset were different we could have different difficulties for each dataset and the user might be biased towards the tool where he worked on the dataset with the easier tasks. By using different tasks for each dataset this also meant the user could not take his experience performing a task on a tool to the next tool. The versions of the questionnaire that the users had were attributed at random, only taking into account how many answers each version had in order to keep the answers spread across all versions of the questionnaire. As the learning effect of starting in one tool or one dataset affects the results for both tools we can neglect it.

5.6.2 *Conclusion Validity*

A concern is the low amount of participants, which leads to a lower statistical power for the study. However, by using mostly non students and working adults we expect representative results.

Fishing is a possible threat as one may be searching for particular results, as the researcher might influence the result (Wohlin et al., 2012). We have minimized this threat by having the same set of tasks for both tools and giving random versions of the questionnaire and before having any feedback from the user in terms of their experience or background.

Random irrelevancies in experimental setting."Elements outside the experimental setting may disturb the results, such as noise outside the room or a sudden interrupt in the experiment" (Wohlin et al., 2012). This is a viable threat as the circumstances did not allow for any other method apart from an online survey. Since it was online, the subjects answered in their own spaces where other variables could come into play, such as family members or other things interrupting them. We have attempted to minimize this issue by providing optional voice and video sessions where the subject was accompanied and where we tried to keep them focused on the tasks at hand.

5.6.3 *Construct Validity*

Inadequate preoperational explication of constructs is a possible issue related to the constructs not being well defined prior to being measured (Wohlin et al., 2012). In our case we wish to measure the usability of the platform we have developed and compare it to currently available market solutions. The measurement of usability of each tool was done with the SUS which has become an industry standard with references in over 600 publications (Sauro, 2011).

Another possible issue is the *mono-operation bias* concerned with the under-representation of a construct. As we are comparing our tool with another and used different datasets and interchanged them between each different subject this is not an issue.

5.6.4 *External Validity*

In general, in this category of threats it is paramount to report the characteristics of the experiment in order to understand its applicability to other contexts (Wohlin et al., 2012).

A common external validity threat is termed *interaction of selection and treatment* meaning the population chosen is not representative or limits the potential generalization of the study (Wohlin et al., 2012). We accepted all interested participants while trying to avoid entries from subjects with a background in computer science or programming. The participants we accrued are from multiple academic backgrounds, from doctors to law (Section 5.3.1). With the span of backgrounds we have from our participants we believe this study could be applied with subjects from multiple areas.

Another threat in this category is the *pre-test-treatment interaction* which says that the interaction with the subject during the pre test, sensitizing users towards aspects of the treatment, may influence the post test scores. During the execution of the study the interaction with the users prior to their participation was a single email where we do the following:

1. Explain what tools the subject will need to install on their machine;
2. Attached the required data for the participation in the study;
3. Explain how to contact us for supervision during the questionnaire.

6

CONCLUSION

In this last chapter we present some concluding remarks in section 6.1 and future work in section 6.2. We also look to provide an answer to the research questions that were presented in Section 1.3.

6.1 CONCLUDING OBSERVATIONS

DS is a very complicated process with multiple steps. With our thesis we have attempted to make the first step in DC more accessible to the average end user, which has no background in programming/computer science.

DCA works as a web browser application that does not require any installation. DCA helps users perform DC tasks by removing any need at all for programming knowledge by using suggestions and descriptive statistics together with a non abstract way to visualize data with a spreadsheet like view of the data. These features together with the use of a preview table allow the user to better understand the data and what each transformation does.

With the results we have obtained in the empirical study (Section 5), where we compared our tool with TPB, we can say that we have matched and sometimes surpass one of the main tools currently found in the market (TPB). The results indicate that for simpler tasks DCA outperforms or matches TPB.

There are a number of possible transformations in a data cleaning environment such as joining datasets, multi column transformations among others. Due to this, it must be noted that not all were implemented in our tool. Our purpose has always been to prove that this approach could be successful and together with the results from the empirical validation we can say that we obtained a favorable result.

In order to answer the research questions we have presented in Section 1.3 we present the following:

- **RQ1** - *Is is possible to design and implement a tool that can allow end users to do data cleaning?*

  We have developed a tool which allowed end users from various backgrounds to perform multiple DC tasks.

- **RQ2** - *Does our solution improve productivity of users and their user experience?*

  According to the results obtained in the study in Chapter 5, we have favorable results that show that using DCA and in specific tasks users complete them faster and as such achieve higher productivity. The results also show that, according to the SUS scores we achieved, users generally prefer DCA.

6.2 FUTURE WORK

Even though we have obtained favorable results during the empirical validation we still learned a lot and understand that some improvements could still be made.

We have already addressed some of the issues and things we have learned during the study in Section 5.5.1, however not everything was addressed. Some other possible improvements to DCA include the following:

1. One improvement should be the addition of loading indicators in order to help the user understand that something is being processed.

2. A different implementation of previews with a side by side comparison of columns before and after a transformation.

3. Not all processes used in data cleaning were implemented. One possible improvement here could be the addition of multi column tasks.

4. Statistics could be displayed in an easier way using graphs or other options and made interactive. Similar to the way statistics are presented in Trifacta Wrangler (Trifacta) and TPB.

5. Implemented suggestions correspond to the base level needed and could be further improved through the use of machine learning or other studies in order to identify more possible suggestions.

BIBLIOGRAPHY

Alteryx. https://www.alteryx.com/platform/. Last Visited: 20-01-2020.

The state of data science & machine learning. https://www.kaggle.com/surveys/2017. Last Visited: 20-01-2020.

Knime. https://www.knime.com/. Last Visited: 20-01-2020.

OpenRefine. https://openrefine.org/. Last Visited: 29-01-2020.

Potter's Wheel A-B-C: An Interactive Tool for Data Analysis, Cleansing, and Transformation. http://control.cs.berkeley.edu/abc/. Last Visited: 20-01-2020.

RapidMiner. https://rapidminer.com/. Last Visited: 20-01-2020.

Weka. https://www.cs.waikato.ac.nz/ml/weka/. Last Visited: 20-01-2020.

Winpure. https://winpure.com/. Last Visited: 20-01-2020.

Austin Cory Bart, Javier Tibau, Eli Tilevich, Clifford A. Shaffer, and Dennis Kafura. BlockPy: An Open Access Data-Science Environment for Introductory Programmers. *Computer*, 50(5):18–26, 2017. ISSN 00189162. doi: 10.1109/MC.2017.132.

Somchai Chatvichienchai and Yu Kawasaki. Spreaddb: Spreadsheet-based user interface for querying and updating data of external databases. *International Journal of Advanced Trends in Computer Science and Engineering*, 7(2):6–10, 2018. ISSN 22783091. doi: 10.30534/ijatcse/2018/01722018.

Jácome Cunha, João Paulo Fernandes, and Paula Pereira. Humanized Data Science. 2020. Submitted for CHI 2020.

Portuguese Government. Contrato para a legislatura com o ensino superior para 2020–2023. https://www.portugal.gov.pt/download-ficheiros/ficheiro.aspx?v=d2607a18-51c9-489c-a61c-1ff420dab2f0, Nov 2019.

Ross Ihaka and Robert Gentleman. R: The r project for statistical computing. https://www.r-project.org/. Last Visited: 21-01-2020.

Andrew J. Ko, Robin Abraham, Laura Beckwith, Alan Blackwell, Margaret Burnett, Martin Erwig, Chris Scaffidi, Joseph Lawrance, Henry Lieberman, Brad Myers, Mary Beth Rosson, Gregg Rothermel, Mary Shaw, and Susan Wiedenbeck. The state of the art in end-user software engineering. *ACM Comput. Surv.*, 43(3), April 2011. ISSN 0360-0300. doi: 10.1145/1922649.1922658. URL https://doi.org/10.1145/1922649.1922658.

Jeff Leek. The key word in'data science'is not data, it is science. *Simply Statistics*, 12, 2013.

Bruno Leonel Lopes, Artur Pedroso, Jorge Cardoso, and Rui Pedro Paiva. DataScience4NP - A Data Science Service for Non-Programmers. *In 10º Simpósio da Informática - INForum2018*, pages 1–11, 2018.

Will Markow, Soumya Braganza, Bledi Taska, Steven Miller, and Debbie Hughes. The Quant Crunch: How the Demand For Data Science Skills is Disrupting the Job Market. *Burning Glass Technologies*, page 25, 2017. URL http://web.archive.org/web/20170627143049/https://www.ibm.com/analytics/us/en/technology/data-science/quant-crunch.html.

Dave Mason. Data programming for non-programmers. *Procedia Computer Science*, 21:68–74, 2013. ISSN 18770509. doi: 10.1016/j.procs.2013.09.011. URL http://dx.doi.org/10.1016/j.procs.2013.09.011.

Hugo Matalonga, Bruno Cabral, Fernando Castor, Marco Couto, Rui Pereira, Simao Melo de Sousa, and Joao Paulo Fernandes. Greenhub farmer: real-world data for android energy mining. In *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*, pages 171–175. IEEE, 2019.

Microsoft. Azure ML studio. https://studio.azureml.net/. Last Visited: 20-01-2020.

Heiko Müller and Johann-christoph Freytag. Problems, Methods, and Challenges in Comprehensive Data Cleansing. *Challenges*, (HUB-IB-164):1–23, 2003. URL http://www.dbis.informatik.hu-berlin.de/fileadmin/research/papers/techreports/2003-hub_ib_164-mueller.pdf.

Michael Muller, Ingrid Lange, Dakuo Wang, David Piorkowski, Jason Tsay, Q. Vera Liao, Casey Dugan, and Thomas Erickson. How data science workers work with data: Discovery, capture, curation, design, creation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450359702. doi: 10.1145/3290605.3300356.

University of Ljubljana. Orange. https://orange.biolab.si/. Last Visited: 20-01-2020.

OutSystems. Outsystems. https://www.outsystems.com/, a. Last Visited: 20-01-2020.

OutSystems. Use outsystems with existing databases. https://www.outsystems.com/evaluation-guide/%20use-outsystems-with-existing-databases/#1, b. Last Visited: 20-01-2020.

P. Pereira, J. Cunha, and J. P. Fernandes. On understanding data scientists. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 1–5, 2020. doi: 10.1109/VL/HCC50065.2020.9127269.

Vijayshankar Raman and Joseph M. Hellerstein. Potter's wheel: An interactive data cleaning system. *VLDB 2001 - Proceedings of 27th International Conference on Very Large Data Bases*, pages 381–390, 2001.

Arjun Rao, Ayush Bihani, and Mydhili Nair. Milo: A visual programming environment for data science education. *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC*, 2018-Octob:211–215, 2018. ISSN 19436106. doi: 10.1109/VLHCC.2018.8506504.

Mitchel Resnick. Scratch - imagine, program, share. https://scratch.mit.edu/. Last Visited: 21-01-2020.

Jeff Sauro. Measuring usability with the system usability scale (sus), 2011. URL https://measuringu.com/sus/. Last Visited: 16-12-2020.

Tableau Software. Tableau prep. https://www.tableau.com/products/prep. Last Visisted: 20-01-2020.

Amber Thomas. Kaggle 2017 survey results. https://www.kaggle.com/amberthomas/kaggle-2017-survey-results, 2017. Last Visited: 28-01-2021.

Trifacta. Trifacta wrangler. https://www.trifacta.com/products/wrangler-editions/. Last Visited: 20-01-2020.

Guido van Rossum. Python. https://www.python.org/. Last Visited: 21-01-2020.

Claes Wohlin, Per Runeson, Magnus C. Ohlsson Martin Höst, Björn Regnell, and Anders Wesslén. *Experimentation in Software Engineering*. Springer, 2012.

A

PRESONAL INFORMATION GRAPHS

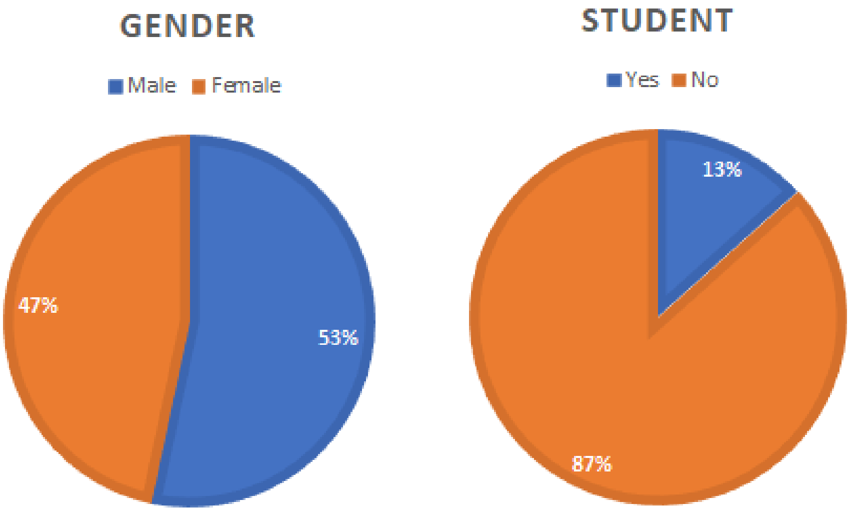


Figure 41: Gender and Student Distribution

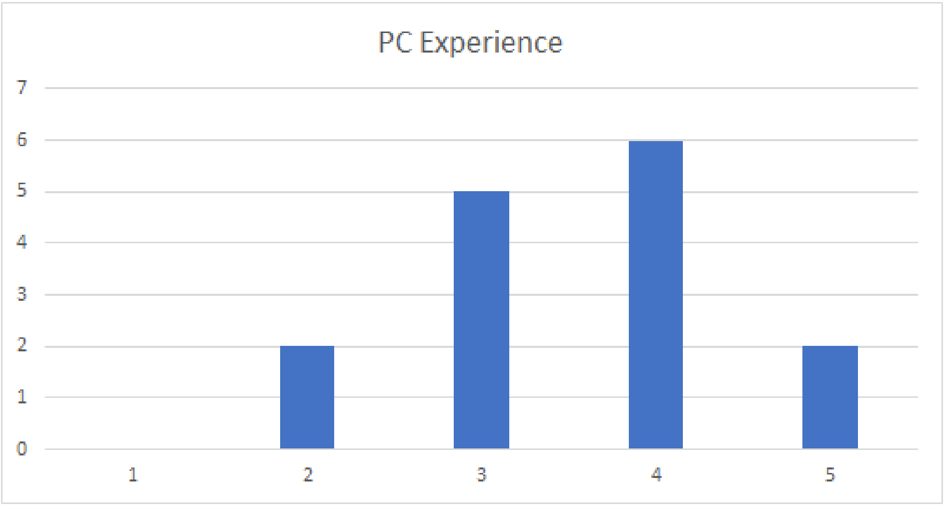


Figure 42: PC Experience Distribution - 1=None; 5=Professional

B

DIFFICULTY GRAPHS

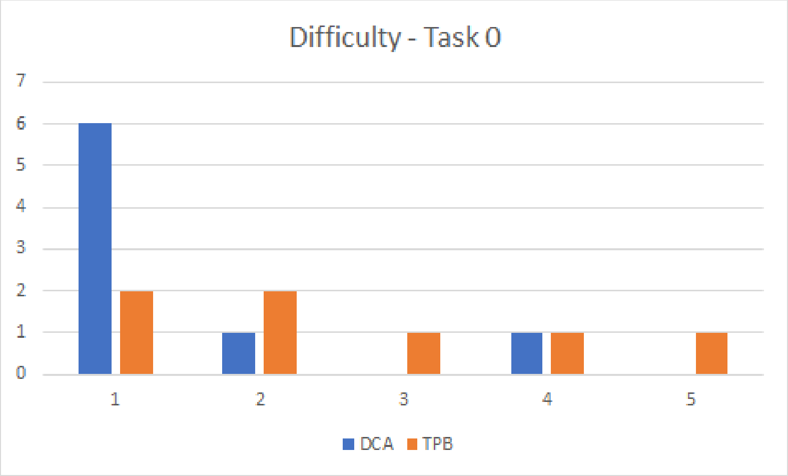


Figure 43: Android Tasks 0 - Difficulty Comparison

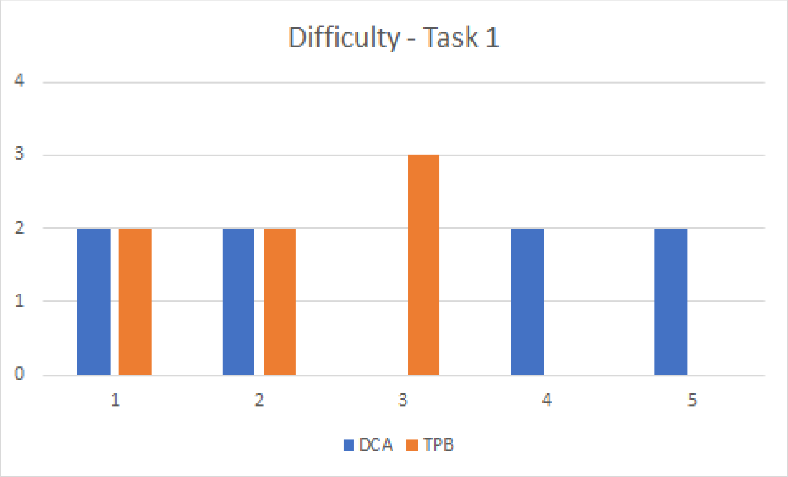


Figure 44: Android Tasks 1 - Difficulty Comparison

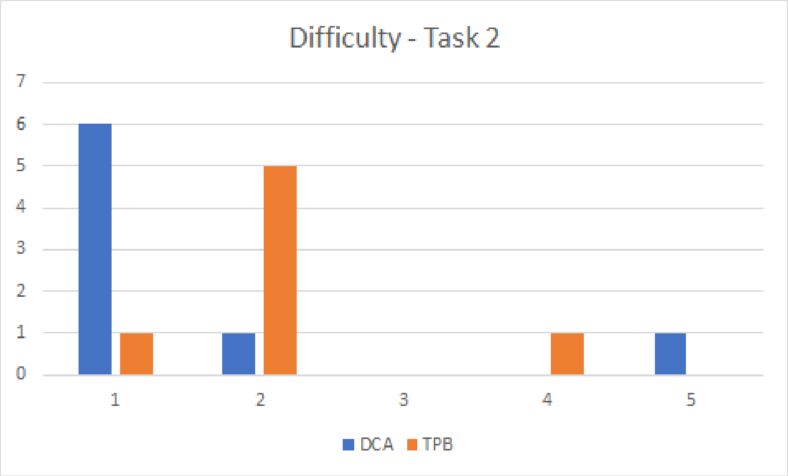


Figure 45: Android Tasks 2 - Difficulty Comparison

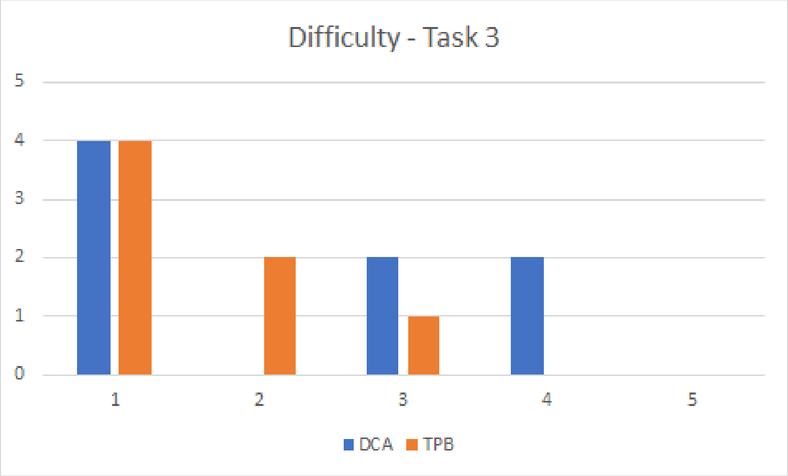


Figure 46: Android Tasks 3 - Difficulty Comparison

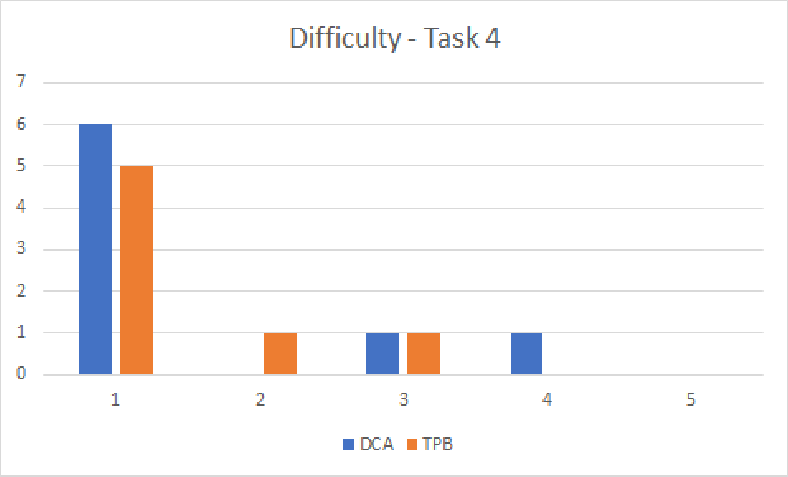


Figure 47: Android Tasks 4 - Difficulty Comparison

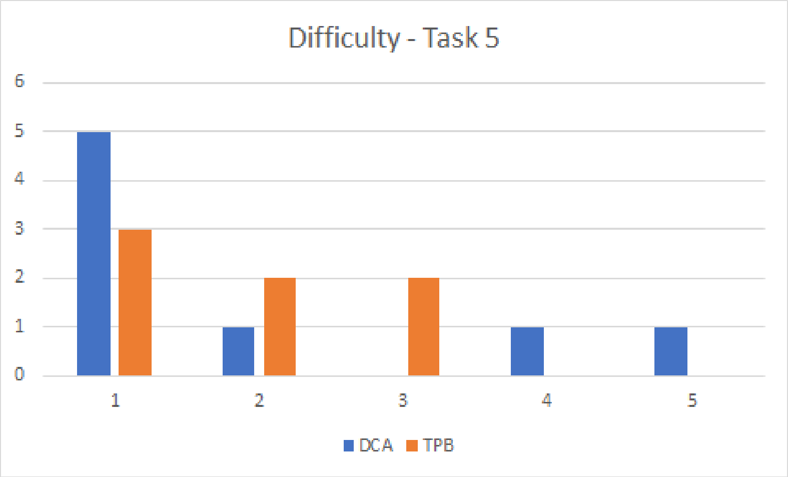


Figure 48: Android Tasks 5 - Difficulty Comparison

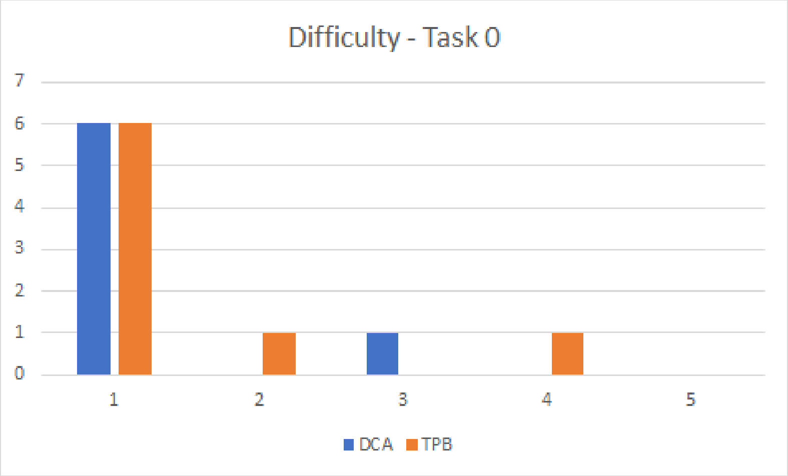


Figure 49: Movies Tasks 0 - Difficulty Comparison

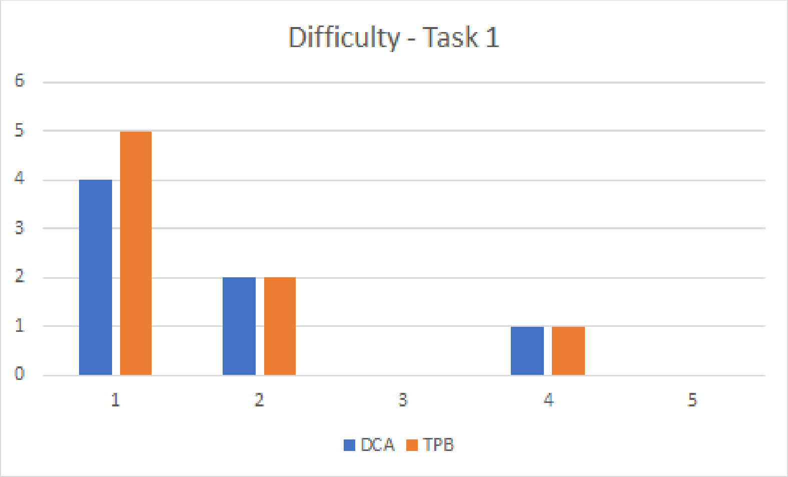


Figure 50: Movies Tasks 1 - Difficulty Comparison

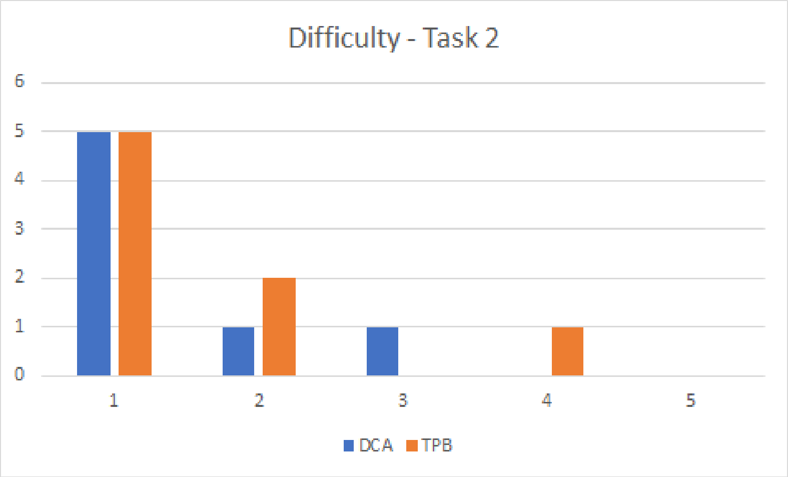


Figure 51: Movies Tasks 2 - Difficulty Comparison

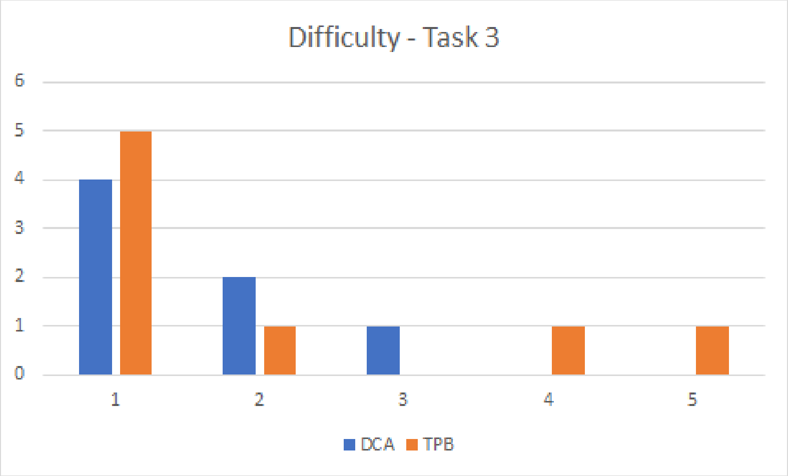


Figure 52: Movies Tasks 3 - Difficulty Comparison

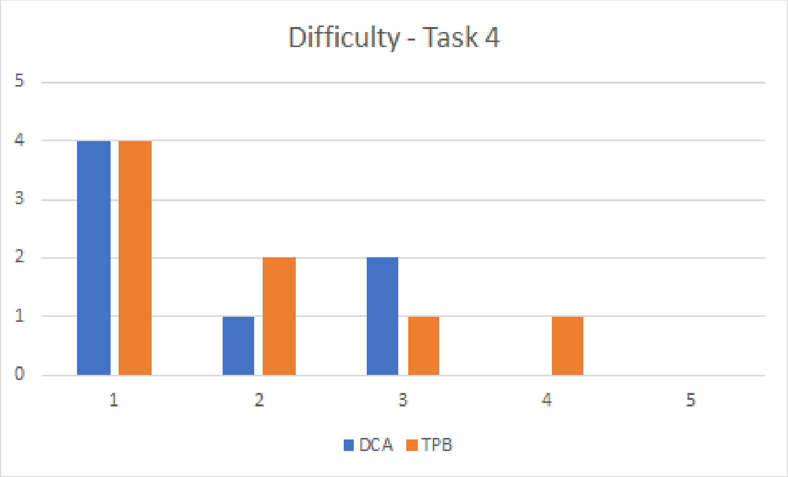


Figure 53: Movies Tasks 4 - Difficulty Comparison

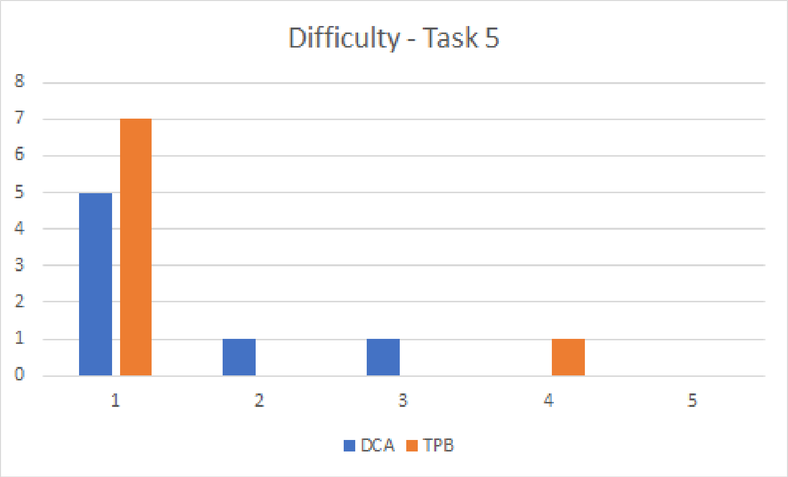


Figure 54: Movies Tasks 5 - Difficulty Comparison

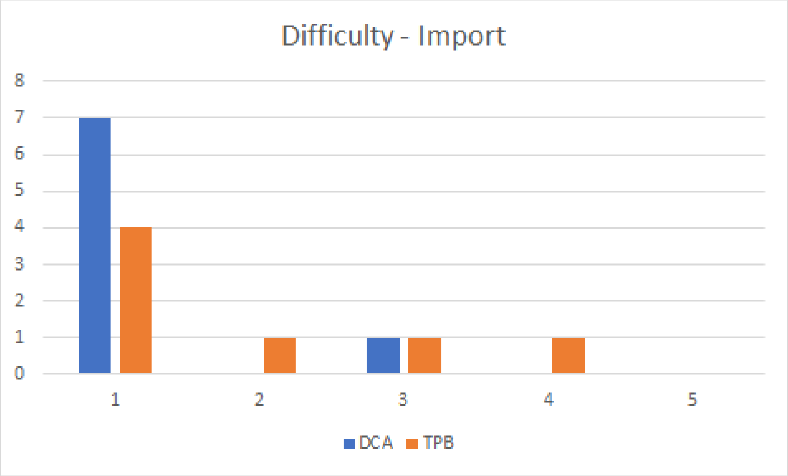


Figure 55: IGN Import Task - Difficulty Comparison

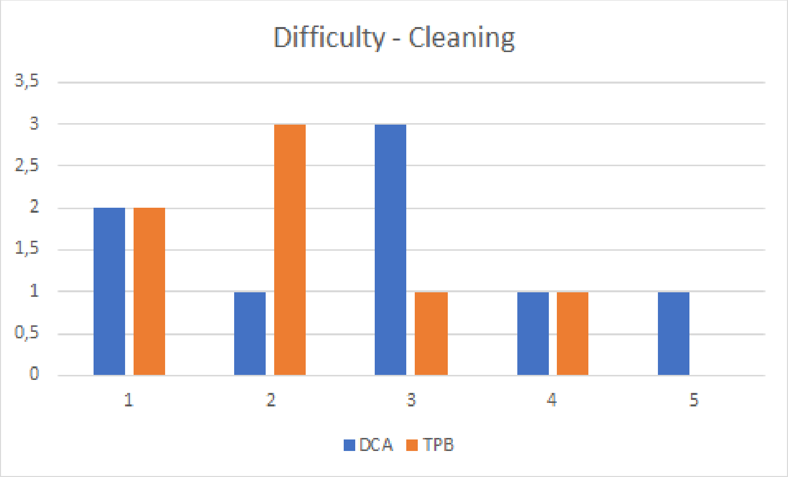


Figure 56: IGN Cleaning Task - Difficulty Comparison

C

TIME GRAPHS

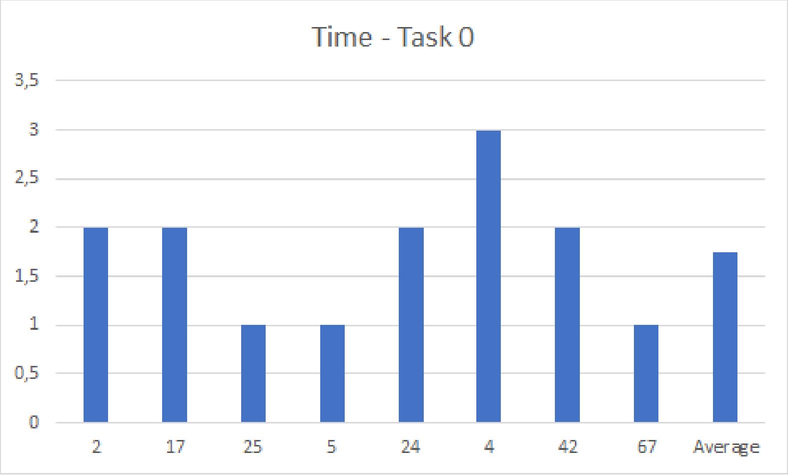


Figure 57: DCA Android Task 0 Time Distribution

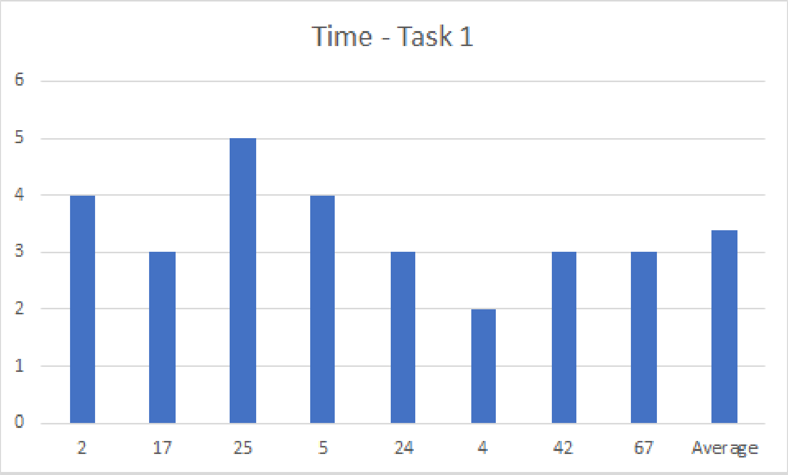


Figure 58: DCA Android Task 1 Time Distribution

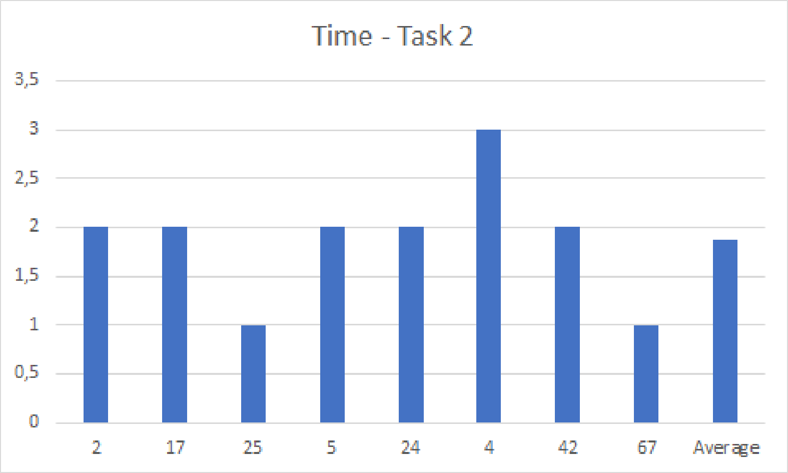


Figure 59: DCA Android Task 2 Time Distribution

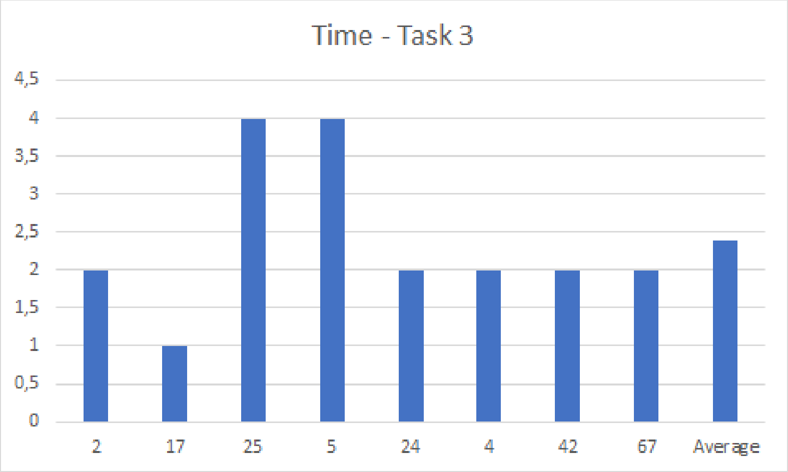


Figure 60: DCA Android Task 3 Time Distribution

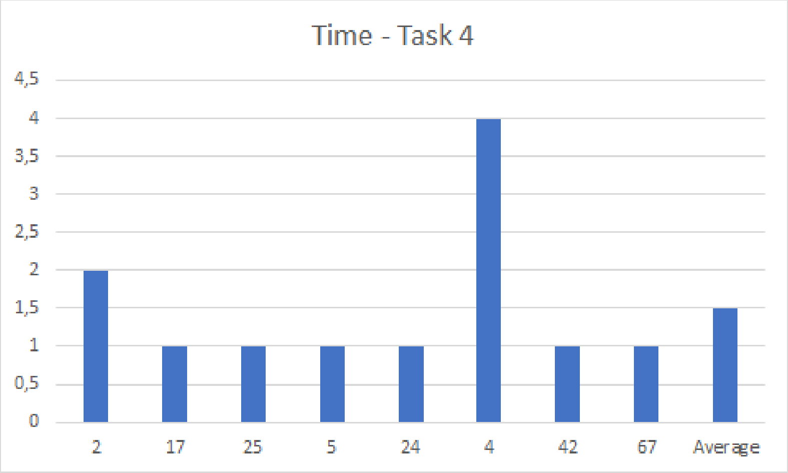


Figure 61: DCA Android Task 4 Time Distribution

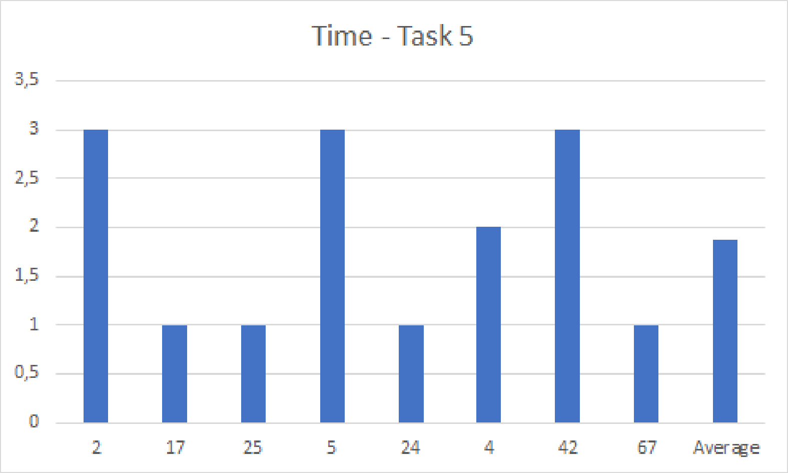


Figure 62: DCA Android Task 5 Time Distribution

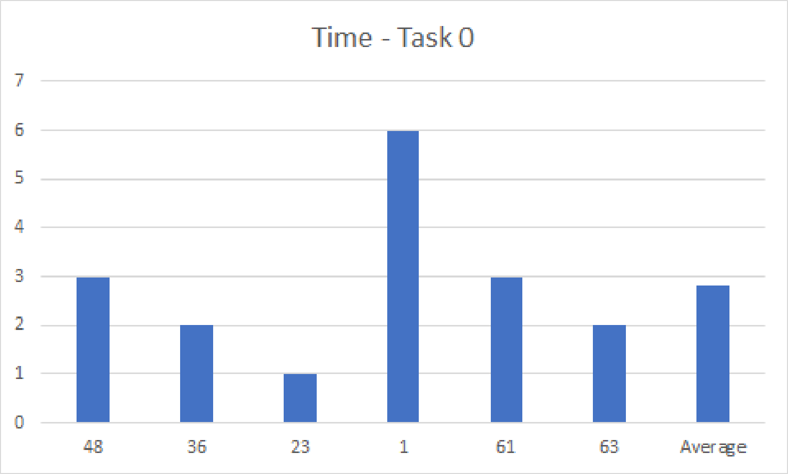


Figure 63: TPB Android Task 0 Time Distribution

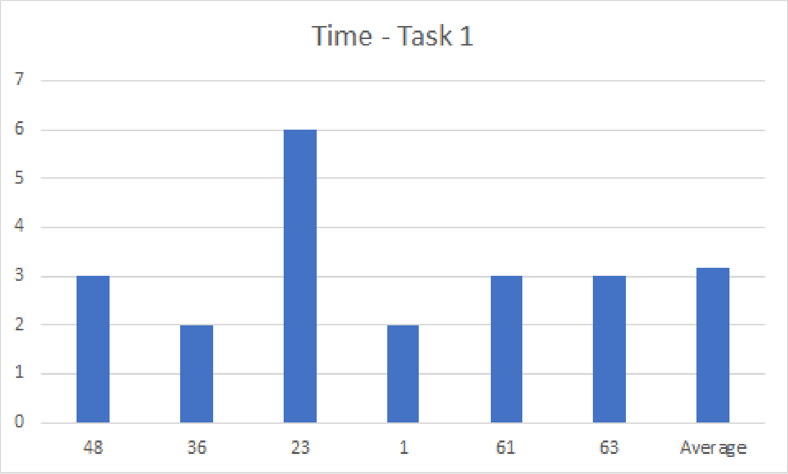


Figure 64: TPB Android Task 1 Time Distribution

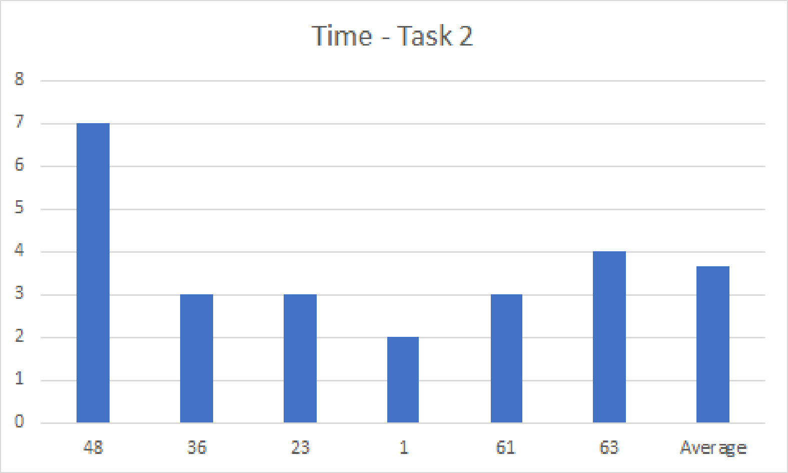


Figure 65: TPB Android Task 2 Time Distribution

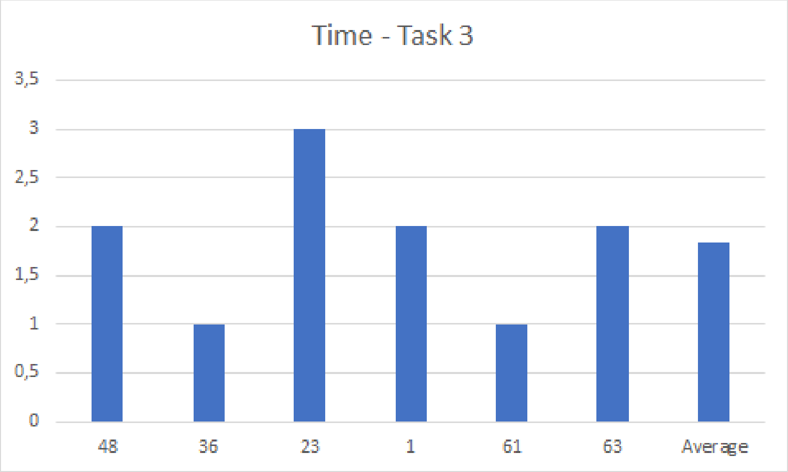


Figure 66: TPB Android Task 3 Time Distribution

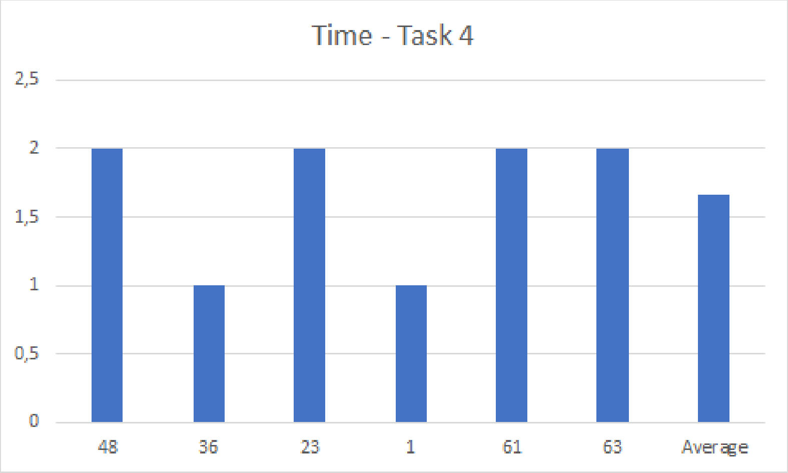


Figure 67: TPB Android Task 4 Time Distribution

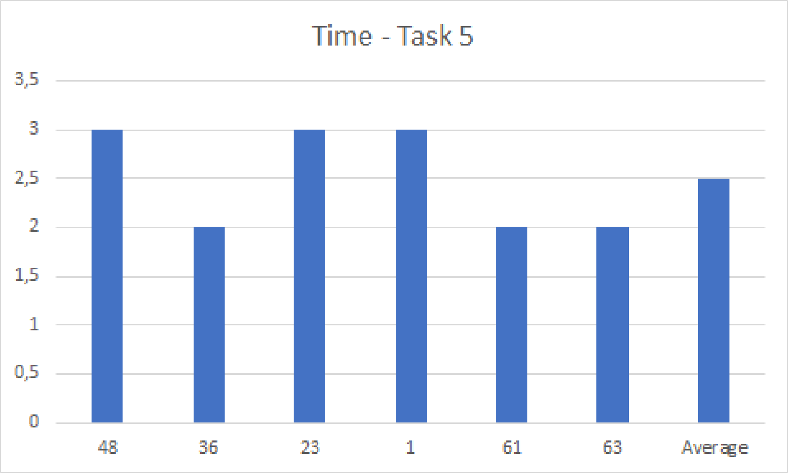


Figure 68: TPB Android Task 5 Time Distribution

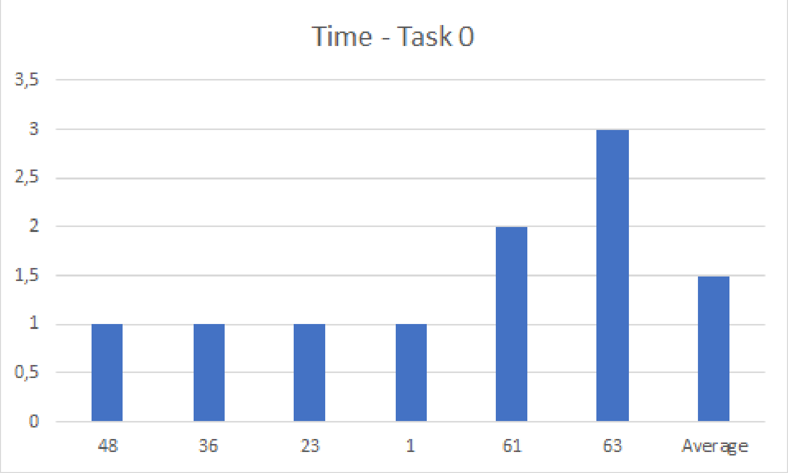


Figure 69: DCA Movies Task 0 Time Distribution

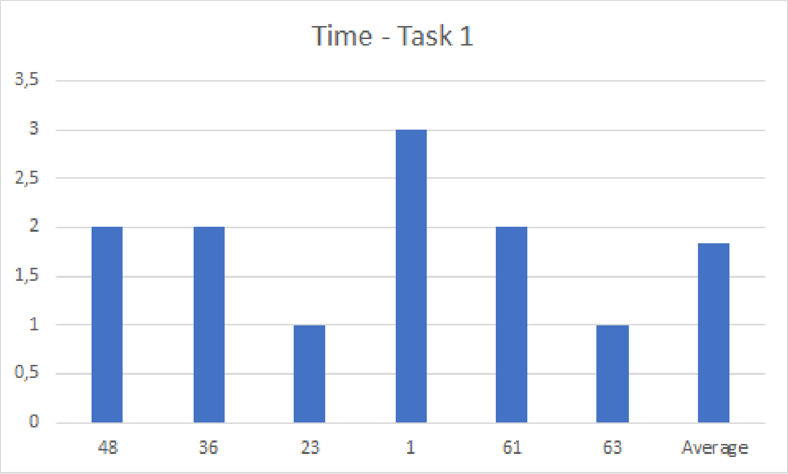


Figure 70: DCA Movies Task 1 Time Distribution

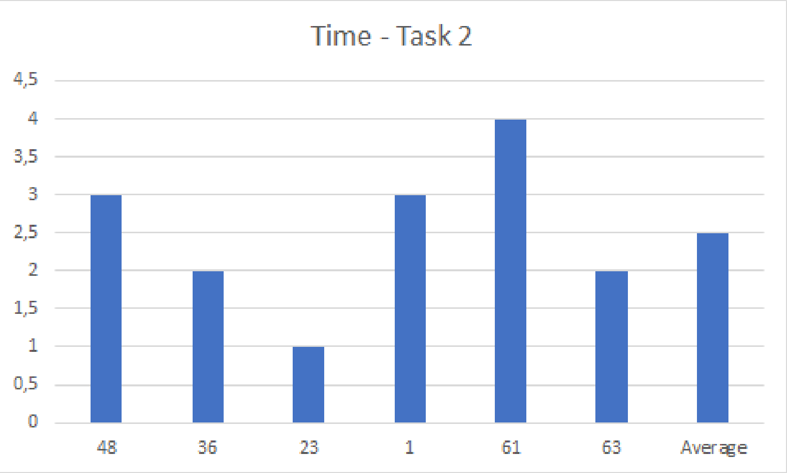


Figure 71: DCA Movies Task 2 Time Distribution

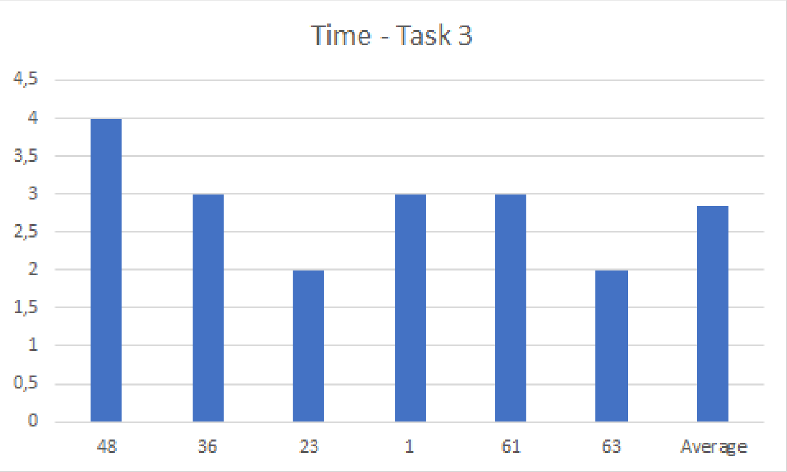


Figure 72: DCA Movies Task 3 Time Distribution

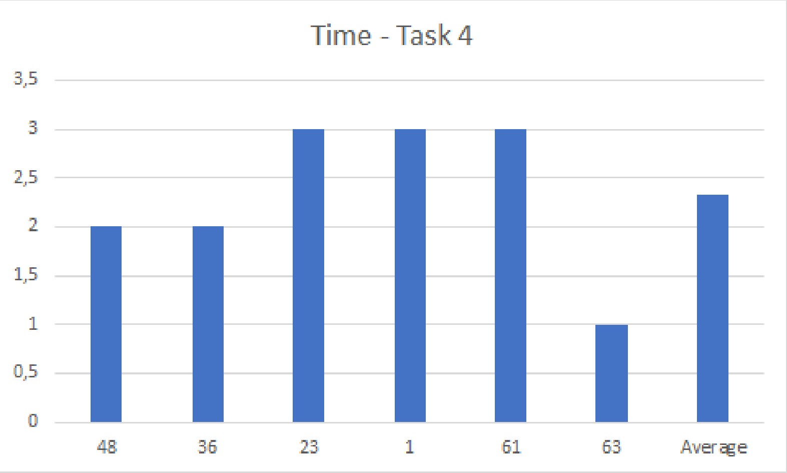


Figure 73: DCA Movies Task 4 Time Distribution

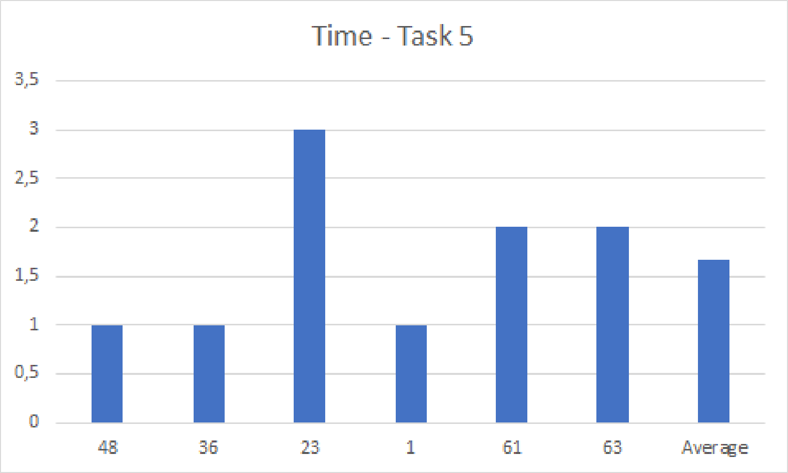


Figure 74: DCA Movies Task 5 Time Distribution

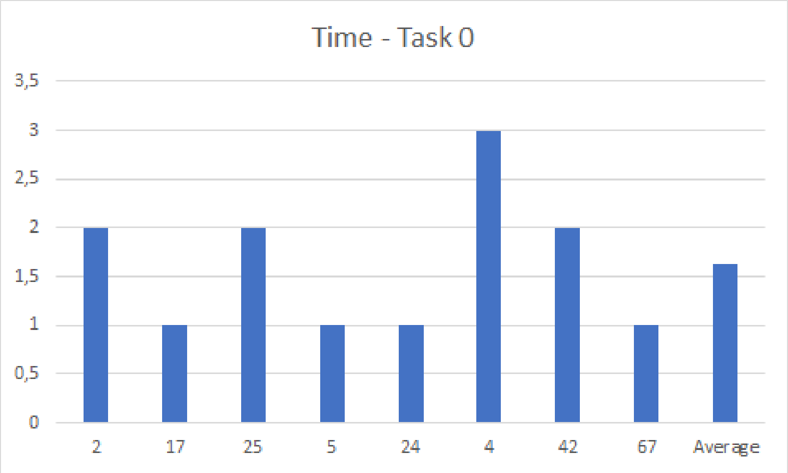


Figure 75: TPB Movies Task 0 Time Distribution

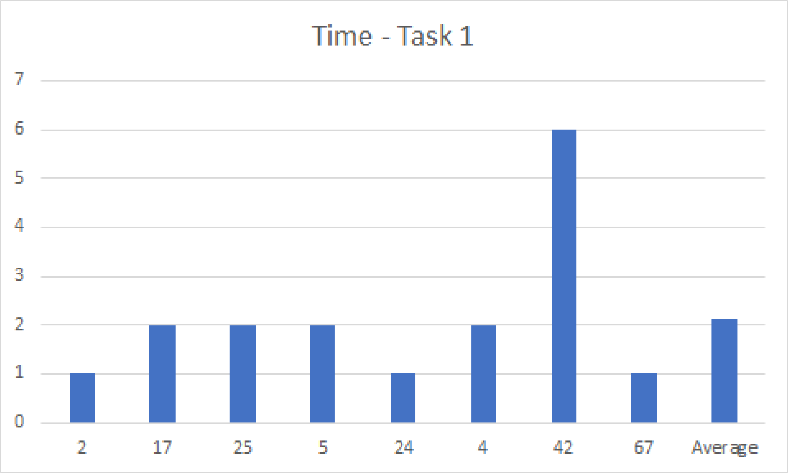


Figure 76: TPB Movies Task 1 Time Distribution

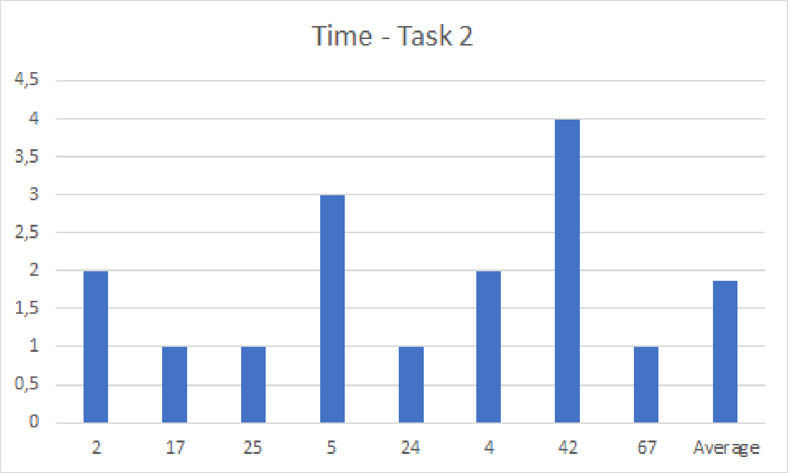


Figure 77: TPB Movies Task 2 Time Distribution

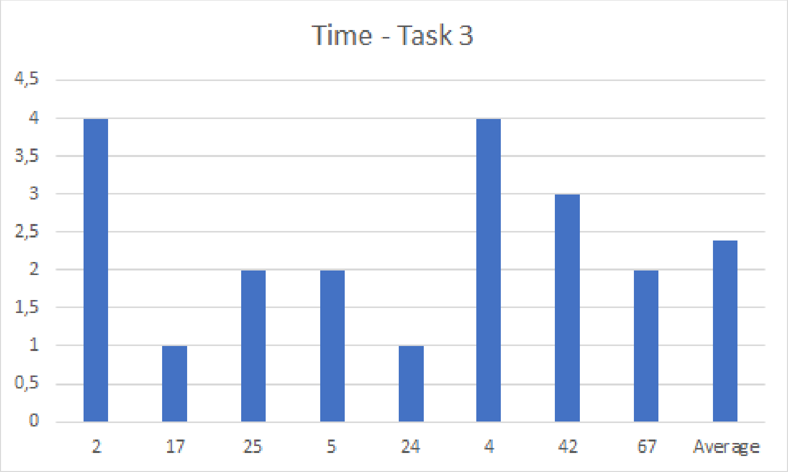


Figure 78: TPB Movies Task 3 Time Distribution

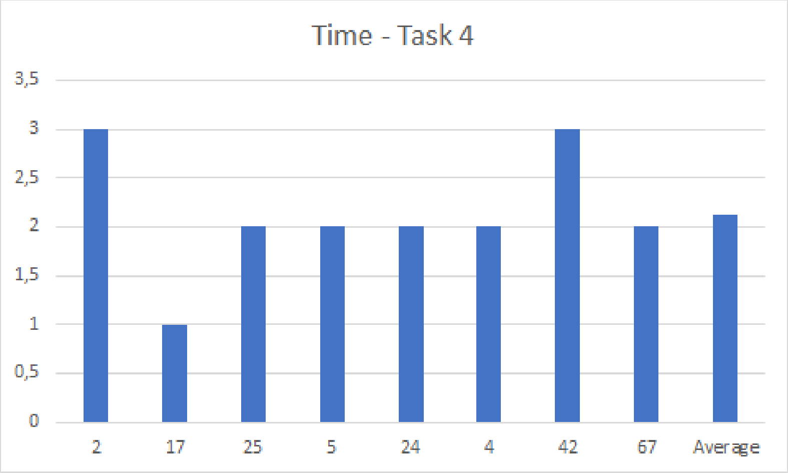


Figure 79: TPB Movies Task 4 Time Distribution

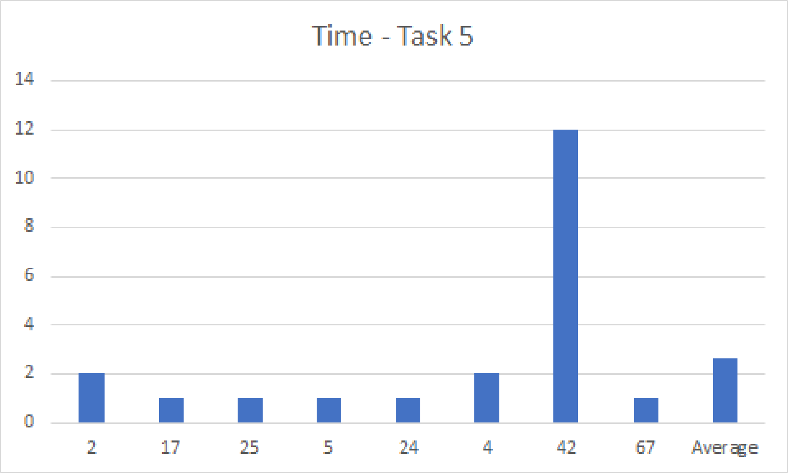


Figure 80: TPB Movies Task 5 Time Distribution

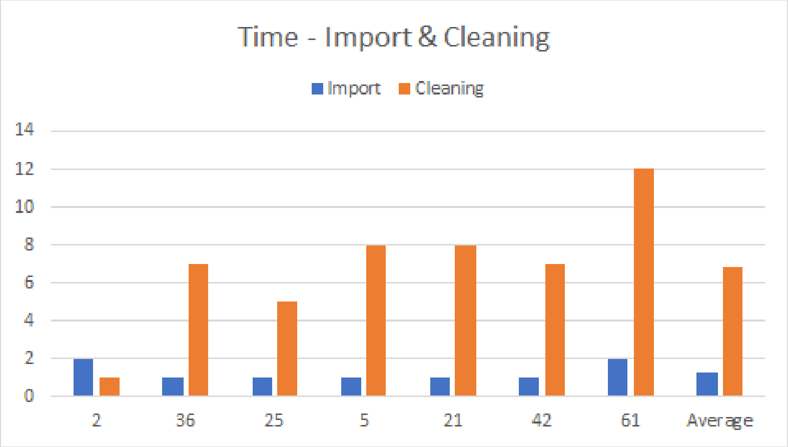


Figure 81: DCA IGN Import and Cleaning Time Distribution

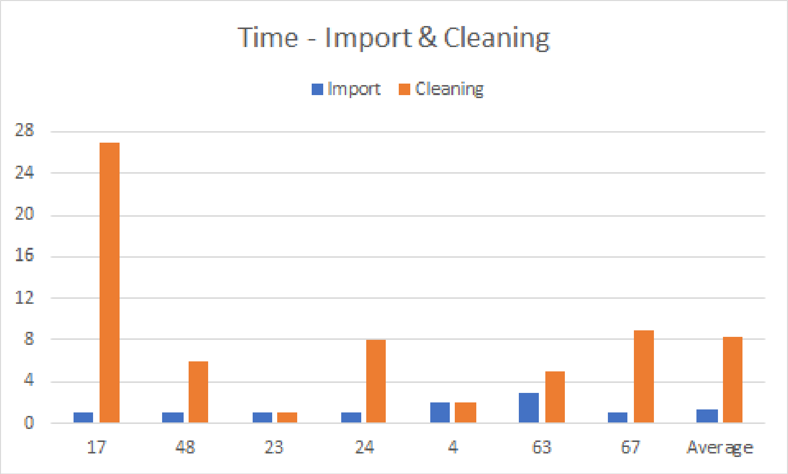


Figure 82: TPB IGN Import and Cleaning Time Distribution

D

QUESTIONNAIRE

D.1 PERSONAL INFO

Identificador: ______________________________

Sexo:

□ Masculino
□ Feminino

Idade: ______________________________

É Estudante:

□ Sim
□ Não

Experiência em utilização de computadores (1 - Nenhuma, 5 - Profissional):

□ 1 □ 2 □ 3 □ 4 □ 5

Experiência em Data Science/Data Cleaning (1 - Nenhuma, 5 - Profissional):

□ 1 □ 2 □ 3 □ 4 □ 5

D.2 DATA CLEANING FOR ALL TUTORIAL

Importar Dados

1. Aceda a: https://data-cleaning-for-all.herokuapp.com/
2. No canto superior esquerdo clique nos 3 traços horizontais.
3. Selecione a opção "Import File" no menu que abriu
4. Na nova pagina clique em "File Input" e selecione o ficheiro pretendido (Para este tutorial é "MaxTempPerCity.csv")
5. Clique em "Next"
6. Clique em "Apply"

Questões Tutorial

1. Find And Replace - Substitua o valor "Congo (Democratic Republic Of The)" por "Congo".

– Selecione a coluna "Country" e através da carta com o titulo "Find Value and Replace it?" substitua as entradas com o valor "Congo (Democratic Republic Of The)" por "Congo".

2. Split - Divida a coluna dt em 3 novas colunas ″Month″, ″Day″ e ″Year″.
– Selecione a coluna ″dt″ e através da carta com o título ″Split on First Instance″ divida a coluna pelo caracter “ / ” em duas novas colunas “Month” e “Day/Year”. Repita a tarefa na nova coluna “Day/Year” divida a coluna pelo caracter “ / ” em duas novas colunas “Day” e “Year”.
3. Find And Remove - Exclua todas as entradas que nao pertençam ao ano mais representado.
– Selecione a nova coluna ″Year″ e através da carta com o título ″Find and Remove Matches″ selecione a opção “Is not equal to” e insira o valor ″2010″.
4. Normalize String Case - Normalize a coluna ″City″, alterando todos os valores para letra maiúscula.
– Selecione a coluna “City” e através da carta com o título “Normalize String Case” selecione a opção “All Uppercase” e aplique.
5. Bounds Card - Remova entradas onde a ″AverageTemperatureUncertainty″ não seja entre 0 e 1.
– Selecione a coluna “AverageTemperatureUncertainty” e através da carta com o título “Define bounds and remove outliers?” insira o valor 0 como mínimo e 1 como máximo.
6. Remove Column – Remova a coluna que criou com o nome “Day”.
– No cabeçalho da coluna com o nome “Day” clique na seta e selecione a opção “Remove Column″.
7. Remove Empty Cells/Nulls - Remova entradas onde não se encontra registada alguma ″AverageTemperature″.
– Selecione a coluna “AverageTemperature” e na carta com o título “Replace or remove null values?” selecione a opção “Remove”.
8. Find And Remove - Na coluna ″AverageTemperature″ remova entradas que possuem valores inferiores á media geral.
– Selecione a coluna “AverageTemperature” e verifique o valor médio através da carta de estatísticas. Posteriormente, na carta com o título “Find and Remove Matches” selecione a opção “Less than”, clique em “Next” e insira o valor médio que verificou.
9. Replace Empty Cells/Nulls - Substitua células vazias/nulos na coluna ″City″ pelo valor ″No City Recorded″
– Selecione a coluna “City” e na carta com o título “Replace or remove null values?” selecione a opção “Replace” e insira o valor “No City Recorded”

Exportar Dados e Operações

1. Clique em ″Apply to Data″
2. Clique em ″Export Data to CSV″ e guarde o ficheiro.
3. Clique em ″Export changes to file″ e guarde o ficheiro

Nome dos Ficheiros:

- Nome do ficheiro criado quando clica em export changes: ″ChangesDCATutorial″
- Nome do ficheiro criado quando clica em export data: ″DataDCATutorial″

D.3 QUESTIONNAIRE MOVIES

Tarefa 0

Importe os dados do ficheiro ″movies.csv″ na ferramenta Data Cleaning for All

Inicio: ____

Fim:

Dificuldade (1 - Muito Fácil, 5 - Muito Dificil):

□ 1 □ 2 □ 3 □ 4 □ 5

Tarefa 1

Remova entradas com valores nulos/células vazias na coluna “content_rating” .

Inicio:

Fim:

Dificuldade (1 - Muito Fácil, 5 - Muito Dificil):

□ 1 □ 2 □ 3 □ 4 □ 5

Tarefa 2

Remova entradas com números negativos na coluna ”num_user_for_review”.

Inicio:

Fim:

Dificuldade (1 - Muito Fácil, 5 - Muito Dificil):

□ 1 □ 2 □ 3 □ 4 □ 5

Tarefa 3

Substitua os valores nulos/células vazias na coluna ”movie_imdb_link” por “No Website Found”.

Inicio:

Fim:

Dificuldade (1 - Muito Fácil, 5 - Muito Dificil):

□ 1 □ 2 □ 3 □ 4 □ 5

Tarefa 4

Remova todas as entradas que possuem ”title_year” inferior a 2010

Inicio:

Fim:

Dificuldade (1 - Muito Fácil, 5 - Muito Dificil):

□ 1 □ 2 □ 3 □ 4 □ 5

Tarefa 5

Remova a coluna “director_facebook_likes”.

Inicio:

Fim:

Dificuldade (1 - Muito Fácil, 5 - Muito Dificil):

□ 1 □ 2 □ 3 □ 4 □ 5

Guardar Ficheiros

Export Data:
Guarde com o nome: ”DataDCAMovies”

Export Changes:
Guarde com o nome: ”ChangesDCAMovies”

D.4 TABLEAU PREP BUILDER TUTORIAL

Instalação caso ainda não tenha sido efetuada:

1. Aceda a: https://www.tableau.com/products/prep/download
2. Insira o seu email (p.e. o seu email institucional)
3. Verifique que esta a efetuar o download do Tableau Prep Builder
4. Clique em ″Start your free trial″
5. Guarde o instalador
6. Execute o instalador
7. Após acabar a instalação execute o Tableau Prep

Importar Dados no Tableau Prep

1. Clique na seta do lado esquerdo para abrir o menu lateral
2. Deve ver apenas uma opção ″Conexões″ com um pequeno sinal ″+″
3. Clique no ″+″ e seleccione no menu seguinte a opção ″Arquivo de texto″
4. Selecione o ficheiro de dados pretendido. (Para este tutorial é o ficheiro com o nome ″MaxTempPerCity.csv″)

Questões Tutorial

1. Find And Replace - Substitua o valor ″Congo (Democratic Republic Of The)″ por ″Congo″.

– Na carta de estatísticas da coluna “country”, selecione a lupa e pesquise pelo valor que pretendemos alterar. Após encontrar a linha “Congo (Democratic Republic Of The)” faça duplo clique na linha para alterar todos os valores e altere para “Congo”

2. Split - Divida a coluna dt em 3 novas colunas ″Month″, ″Day″ e ″Year″.

– Na carta de estatísticas da coluna “dt” selecione o icone azul no canto superior esquerdo e altere o tipo de dados para “Cadeia de caracteres”/″string”. Clique nos 3 pontos e selecione “Dividir -¿ Divisão personalizada”. Insira o caractere apropriado (” - ”) e a opção de divisão “Todos”. Renomeie as novas colunas para o seu nome apropriado “Year” “Month” “Day”.

3. Remove Column – Remova a coluna que criou com o nome “Day” e a antiga coluna ″dt”.

– Para eliminar uma coluna selecione na carta de estatísticas dela os “3 pontos” e a opção “Remover”

4. Find And Remove - Exclua todas as entradas que nao pertençam ao ano mais representado.

– Na carta de estatísticas clique no botão com 3 barras, isso irá ordenar por ordem de presença nos dados os valores. Clique até ficar em ordem decrescente. Quando encontrar a barra com o ano mais representado clique com o botão direito em cima da respetiva barra e selecione a opção “Manter Apenas”.

5. Normalize String Case - Normalize a coluna ″City″, alterando todos os valores para letra maiúscula.

– Na carta de estatísticas clique no botão com 3 pontos e selecione a opção “Limpar -¿ Tornar as letras maiúsculas”

6. Bounds Card - Remova entradas onde a ″AverageTemperatureUncertainty″ não seja entre 0 e 1.

– Na carta de estatísticas clique no botão com 3 pontos e selecione a opção “Filtrar -¿ Intervalo de valores”. Insira os valores apropriados para ambos os limites e clique em “Concluido”

7. Remove Column – Remova a coluna que criou com o nome “Day”.

–

8. Remove Empty Cells/Nulls - Remova entradas onde não se encontra registada alguma ”AverageTemperature”.

– Na carta de estatísticas procure a barra correspondente a valores nulos e clique com o botão direito e selecione a opção “Excluir”

9. Filter And Drop - Na coluna ”AverageTemperature” remova entradas que possuem valores inferiores a 25.

– Na carta de estatísticas clique no botão com 3 pontos e selecione a opção “Filtrar -¿ Intervalo de valores”. Selecione a opção “Mínimo” e insira o valor apropriado para o limite e clique em “Concluído”

10. Replace Empty Cells/Nulls - Substitua células vazias/nulos na coluna ”City” pelo valor ”No City Recorded”

– Na carta de estatísticas procure a barra correspondente a valores nulos e faça duplo clique para editar todos os valores nulos na tabela. Insira o valor indicado.

Criar ficheiro de saída

1. Nos menus no topo do ecrã selecione ”Arquivo”
2. No submenu de ”Arquivo” selecione ”Exportar fluxo encapsulado...”
3. Guarde com o nome indicado. Para o este guiao guarde como: ”TPBTutorial”

D.5 QUESTIONNAIRE ANDROID

Introdução - Android

O dataset “Android” representa uma coleção de amostras retiradas de telemóveis. Para cada telemóvel foi recolhido o sei modelo, marca, versão do sistema operativo, nível de bateria, código do país e o seu fuso horário. Cada dispositivo individual está representado apenas uma vez.

Tarefa 0

Importe os dados do ficheiro ”android.csv” na ferramenta Tableau Prep Builder

Inicio: ______________________________

Fim: ______________________________

Dificuldade (1 - Muito Fácil, 5 - Muito Dificil):

□ 1 □ 2 □ 3 □ 4 □ 5

Tarefa 1

Caso eles existam, edite os valores da coluna “os_version” que não sejam do tipo “Número.Número.Número” (Ex: “7.1.1”) ou “Número.Número” (Ex: “6.0”) ou ”Número” (Ex: ”9”).

Inicio: ______________________________

Fim: ______________________________

Dificuldade (1 - Muito Fácil, 5 - Muito Dificil):

□ 1 □ 2 □ 3 □ 4 □ 5

Tarefa 2

Divida a Informação da coluna “timezone” em 2 novas colunas – Continente e Cidade

Inicio: ______________________________

Fim: ______________________________

Dificuldade (1 - Muito Fácil, 5 - Muito Dificil):
□ 1 □ 2 □ 3 □ 4 □ 5

Tarefa 3

Filtre os dados de modo a conter apenas os valores do continente mais representado.
Inicio: ____________________
Fim: ____________________
Dificuldade (1 - Muito Fácil, 5 - Muito Dificil):
□ 1 □ 2 □ 3 □ 4 □ 5

Tarefa 4

Normalize os valores da coluna "brand" modificando todos os valores para Letra Maiúscula.
Inicio: ____________________
Fim: ____________________
Dificuldade (1 - Muito Fácil, 5 - Muito Dificil):
□ 1 □ 2 □ 3 □ 4 □ 5

Tarefa 5

Caso existam, remova entradas onde, na coluna "battery_level", estão representados valores impossíveis. (Valores negativos ou superiores a 1)
Inicio: ____________________
Fim: ____________________
Dificuldade (1 - Muito Fácil, 5 - Muito Dificil):
□ 1 □ 2 □ 3 □ 4 □ 5

Exportar Fluxo Encapsulado

Guarde com o nome "TPBAndroid"

D.6 SYSTEM USABILITY SCALE QUESTIONNAIRE - TABLEAU PREP BUILDER

Gostaria de usar esta ferramenta frequentemente (1 - Discordo Fortemente, 5 - Concordo Fortemente):
□ 1 □ 2 □ 3 □ 4 □ 5

Acho que a ferramenta é desnecessariamente complexa (1 - Discordo Fortemente, 5 - Concordo Fortemente):
□ 1 □ 2 □ 3 □ 4 □ 5

Achei uma ferramenta fácil de utilizar (1 - Discordo Fortemente, 5 - Concordo Fortemente):
□ 1 □ 2 □ 3 □ 4 □ 5

Penso que precisarei do suporte de um técnico para utilizar esta ferramenta (1 - Discordo Fortemente, 5 - Concordo Fortemente):
□ 1 □ 2 □ 3 □ 4 □ 5

As várias funções desta ferramenta encontram-se bem integradas (1 - Discordo Fortemente, 5 - Concordo Fortemente):

□ 1 □ 2 □ 3 □ 4 □ 5

Existem demasiadas inconsistências na ferramenta (1 - Discordo Fortemente, 5 - Concordo Fortemente):

□ 1 □ 2 □ 3 □ 4 □ 5

Imagino que outras pessoas serão capazes de aprender a trabalhar com esta ferramenta facilmente (1 - Discordo Fortemente, 5 - Concordo Fortemente):

□ 1 □ 2 □ 3 □ 4 □ 5

Achei a ferramenta demasiado complicada para a utilizar (1 - Discordo Fortemente, 5 - Concordo Fortemente):

□ 1 □ 2 □ 3 □ 4 □ 5

Senti-me confiante ao utilizar a ferramenta (1 - Discordo Fortemente, 5 - Concordo Fortemente):

□ 1 □ 2 □ 3 □ 4 □ 5

Precisei de aprender várias coisas antes de poder utilizar a ferramenta (1 - Discordo Fortemente, 5 - Concordo Fortemente):

□ 1 □ 2 □ 3 □ 4 □ 5

Comentários e Sugestões: ______________________________

____

____

D.7 QUESTIONNAIRE IGN

Tarefa 0

Importe os dados do ficheiro "ign-3k.csv" na ferramenta Data Cleaning for All

Inicio: ______________________________

Fim: ______________________________

Dificuldade (1 - Muito Fácil, 5 - Muito Dificil):

□ 1 □ 2 □ 3 □ 4 □ 5

Tarefa 1

Utilizando os conhecimentos que possui efetue a limpeza dos dados até se sentir confortável com o resultado.

Inicio: ______________________________

Fim: ______________________________

Dificuldade (1 - Muito Fácil, 5 - Muito Dificil):

□ 1 □ 2 □ 3 □ 4 □ 5

Guardar Ficheiros

Export Data:
Guarde com o nome: "DataDCAign"

Export Changes:
Guarde com o nome: "ChangesDCAign"

D.8 SYSTEM USABILITY SCALE QUESTIONNAIRE - DATA CLEANING FOR ALL

Gostaria de usar esta ferramenta frequentemente (1 - Discordo Fortemente, 5 - Concordo Fortemente):

□ 1 □ 2 □ 3 □ 4 □ 5

Acho que a ferramenta é desnecessariamente complexa (1 - Discordo Fortemente, 5 - Concordo Fortemente):

□ 1 □ 2 □ 3 □ 4 □ 5

Achei uma ferramenta fácil de utilizar (1 - Discordo Fortemente, 5 - Concordo Fortemente):

□ 1 □ 2 □ 3 □ 4 □ 5

Penso que precisarei do suporte de um técnico para utilizar esta ferramenta (1 - Discordo Fortemente, 5 - Concordo Fortemente):

□ 1 □ 2 □ 3 □ 4 □ 5

As várias funções desta ferramenta encontram-se bem integradas (1 - Discordo Fortemente, 5 - Concordo Fortemente):

□ 1 □ 2 □ 3 □ 4 □ 5

Existem demasiadas inconsistências na ferramenta (1 - Discordo Fortemente, 5 - Concordo Fortemente):

□ 1 □ 2 □ 3 □ 4 □ 5

Imagino que outras pessoas serão capazes de aprender a trabalhar com esta ferramenta facilmente (1 - Discordo Fortemente, 5 - Concordo Fortemente):

□ 1 □ 2 □ 3 □ 4 □ 5

Achei a ferramenta demasiado complicada para a utilizar (1 - Discordo Fortemente, 5 - Concordo Fortemente):

□ 1 □ 2 □ 3 □ 4 □ 5

Senti-me confiante ao utilizar a ferramenta (1 - Discordo Fortemente, 5 - Concordo Fortemente):

□ 1 □ 2 □ 3 □ 4 □ 5

Precisei de aprender várias coisas antes de poder utilizar a ferramenta (1 - Discordo Fortemente, 5 - Concordo Fortemente):

□ 1 □ 2 □ 3 □ 4 □ 5

Comentários e Sugestões: ____

____

____

D.9 SENDING FILES

Envio de ficheiros

Envio de todos os ficheiros:

Envie em anexo todos os ficheiros gerados ao longo deste questionário para o seguinte email: ”dca2020.answers@gmail.com”. No assunto coloque ”Entrega de ficheiros: [identificador]” (Substitua o ”[identificador]” pelo identificador que lhe foi atribuído.)

Já enviou os ficheiros:

□ Sim

□ Não

E

HCI PUBLISHED ARTICLE

In the following pages is the article submitted and accepted as a short research paper for HCI 2020.

Data Curation: Towards a Tool for All*

José Dias[1], Jácome Cunha[1,2], and Rui Pereira[2]

[1] University of Minho, Portugal
a78494@alunos.uminho.pt jacome@di.uminho.pt
[2] HASLab/INESC Tec, Portugal
ruipereira@di.uminho.pt

Abstract. Data science has started to become one of the most important skills one can have in the modern world, due to data taking an increasingly meaningful role in our lives. The accessibility of data science is however limited, requiring complicated software or programming knowledge. Both can be challenging and hard to master, even for the simple tasks.

With this in mind, we have approached this issue by providing a new data science platform, termed *DS4All.Curation*, that attempts to reduce the necessary knowledge to perform data science tasks, in particular for data cleaning and curation. By combining HCI concepts, this platform is: *simple* to use through direct manipulation and showing transformation previews; allows users to *save time* by eliminate repetitive tasks and automatically calculating many of the common analyses data scientists must perform; and suggests data transformations based on the contents of the data, allowing for a *smarter* environment.

Keywords: Human-Centered Data Science · Data Cleaning · Data Curation.

1 Introduction

The use of data cannot be dissociated from our daily lives - data supports, e.g., social media, is fundamental to guide us in traffic and is being used in precision medicine by promising health-care avenues. In order to support all these data-based services, the amount of data which are produced these days are tremendous, and are still expected to increase significantly within the near future. For example, Facebook experiences about 2.5 billion likes and 300 million photo uploads on a regular day [22]. Of course data by itself, even if in massive amounts, has very little value. Indeed, it is the information extracted from data which has the potential to change and improve our lives. However, the information extraction process is complex, requiring cleaning, transforming, understanding, analyzing and interpreting data [21]. This is what is currently called Data Science (DS) [3], and one incorrect or inaccurate decision in any step of the

* This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project UIDB/50014/2020.

process is sufficient enough to compromise the extracted information [7]. However, the challenge for any data scientist is that performing these steps requires a variety of skills including mathematics, statistics, machine learning (ML), data structures, algorithms, and correlation or causation [9]. Nevertheless, there is a worldwide movement towards pushing everyone to have DS skills. For instance, a study by IBM advocates that academia must ensure data literacy for any student in any field of education [5]. Similarly, the Portuguese Government has also defined that until the end of 2023 all students with higher education must have the opportunity to learn DS [15]. In fact, many other countries have defined national strategies for DS [3]. However, to teach advanced techniques and tools to an entire academic community is challenging, tedious, and difficult to entirely fulfil. Indeed, a study by Kaggle, with more than 16.000 answers from DS practitioners, shows that textual programming languages (PLs) such as Python or R are the most used tools (76.3% and 59.2%, respectively) [6]. Unfortunately, programming is a very challenging task, taking years to train and master. While there are other tools targeting inexperienced users, such as Tableau or Excel, these are much less used (20.4% and 13.7%, respectively [6]). Moreover, there is no empirical evidence of their efficiency and efficacy amongst non-expert users.

Human-computer interaction (HCI) related communities have been proposing several methodologies to aid users in developing their own software. These users are usually termed end users, i.e. computer users with no (or little) software development background, yet still need to develop software, i.e. end-user programming [10]. The proposed methodologies include visual programming [2], programming by example [4] or direct manipulation [19].

In this work we build on such works to further design methodologies and a tool (termed DS4All.Curation) that can be productively used by any end user for performing DS, particularly focusing on data cleaning and curation. The curation and transformation of data is generally a very complex and time consuming process for an experienced data scientist [14, 7]. Oftentimes, several tools or programming languages (a PL can also be seen as a tool) are used for this. But to do so, data scientists must properly learn to use these. This is a larger issue for end user data scientists, with their limited (or inexistent) computer science background.

Thus, we believe that a visual development environment for data science (DS) direct manipulation will help diminish such difficulties and limitations. Naturally, data should be represented in a way that (end) users can actually see and manipulate it using some tabular format, e.g., resembling Excel. Whenever a user wishes to apply a certain transformation, they should also be able to see a preview of how their data will be altered. Such a side-by-side look at the dataset, prior and post changes, aims to help remove a level of abstraction of how data will be changed. Additionally, a user should be able to, at any point, directly manipulate the data within such a dataset previewer, such as updating cell values, or through a drop down menu to allow changes or filtering data on a specific column. For many operations related to data curation [13] this should be sufficient. In essence, this environment must be *simple*.

Such a visual environment must also help guide the user to more efficiently perform their work. Indeed, prior studies suggest DS environments should guide their users [21]. For example, it is very common to calculate the statistical information (average, min, max, etc.) or grouping/clustering of data prior to manipulating the data [18]. Such statistics help data scientists summarize the contents of their data, understanding if there are any outliers present, or if something appears to be incorrect. For such operations, data scientists have to repeatedly turn to using programming or complex tools to perform such common tasks each time and every time they tackle a new dataset. We propose that such common tasks should be automatically performed within our visual environment, in order to facilitate the end user data scientists' work, and in turn *save time*.

We propose to go one step further and use such information to automatically present suggestions of common (or uncommon) transformations to the user, which can be automatically applied by the system. An example would be, in a column representing gender, when detecting similar values such as `FEMALE` and `female`, to suggest replacing one entry by the other or by a new value. Another example would be for columns inferred as numerical, where a suggestion to remove data entries based on minimum and maximum bounds may be presented. The system should also learn with the user, by understanding what operations they repeatedly need and/or use, and intelligently offer suggestions. Offering both statistical information on the data and suggested operations to be performed will lower the amount of time taken to perform such tasks, reduce errors, and also reduce the possibility of incorrectly programming the tasks. As such, the final requirement of a data science environment for any use is that it must be *smart*.

In summary, we propose that a visual data science development environment must be *simple*, *saves time*, and is *smart*. Section 2 presents our initial steps in providing data science end users with an environment adhering to these three principals. In Section 3 we discuss related work and in Section 4 we summarize out contribution and discuss future work.

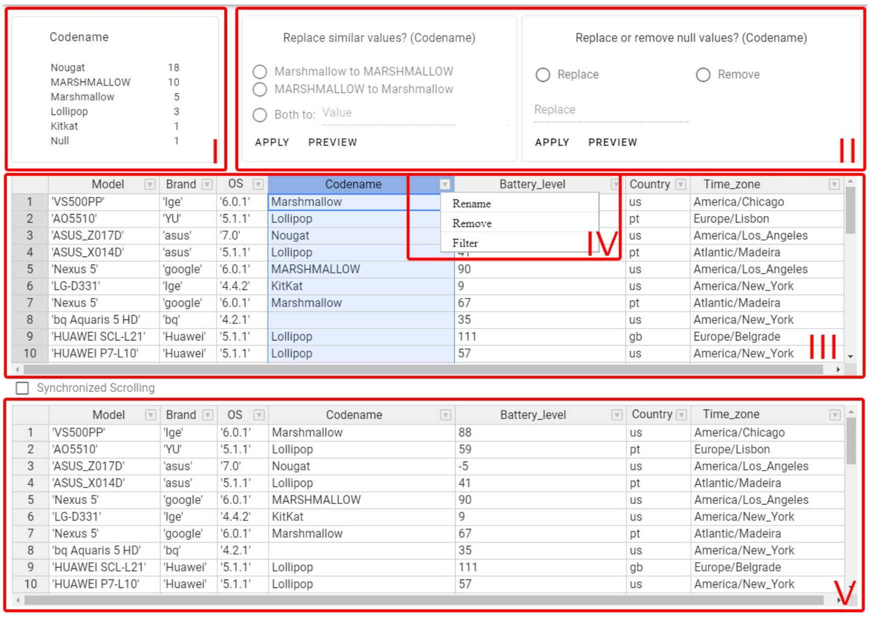
2 DS4All.Curation: A Data Curation Tool for All

In accordance to what we have previously discussed, we believe there are several paths one may take when developing a visual environment for the direct manipulation of data. We have developed a prototype of a humanized data cleaning tool, termed DS4All.Curation[3], shown in Figure 1, that we now describe. The dataset represents Android smartphone usage information [12].

Since we are proposing methodologies and tools for data science, it seems natural that data should be represented in a way users can actually see and manipulate it using some tabular format, e.g., resembling Excel. Indeed, shown in Figure 1 - V (*Original dataset*), we have the original and unaltered dataset shown at all times, allowing the end user to better accompany their transformations. All

[3] DS4All.Curation can be found at https://github.com/Zamreg/HDC.

such transformations would be shown and previewed in Figure 1 - III (*Preview dataset*). This side-by-side look at the dataset before and after applying changes aims to help remove a level of abstraction of how data will be changed, and directly present such actions. At any point, the user may directly manipulate the data within the *Preview dataset*, such as updating cell values, or through a drop down menu (as shown in Figure 1 - IV) to allow changes or filtering data on a specific column. For many operations related to data cleaning/curation [13] this should be sufficient.



Codename	
Nougat	18
MARSHMALLOW	10
Marshmallow	5
Lollipop	3
Kitkat	1
Null	1

Replace similar values? (Codename)

○ Marshmallow to MARSHMALLOW
○ MARSHMALLOW to Marshmallow
○ Both to: Value

Replace or remove null values? (Codename)

○ Replace ○ Remove
Replace

	Model	Brand	OS	Codename	Battery_level	Country	Time_zone
1	'VS500PP'	'lge'	'6.0.1'	Marshmallow		us	America/Chicago
2	'AO5510'	'YU'	'5.1.1'	Lollipop		pt	Europe/Lisbon
3	'ASUS_Z017D'	'asus'	'7.0'	Nougat		us	America/Los_Angeles
4	'ASUS_X014D'	'asus'	'5.1.1'	Lollipop		pt	Atlantic/Madeira
5	'Nexus 5'	'google'	'6.0.1'	MARSHMALLOW	90	us	America/Los_Angeles
6	'LG-D331'	'lge'	'4.4.2'	KitKat	9	us	America/New_York
7	'Nexus 5'	'google'	'6.0.1'	Marshmallow	67	pt	Atlantic/Madeira
8	'bq Aquaris 5 HD'	'bq'	'4.2.1'		35	us	America/New_York
9	'HUAWEI SCL-L21'	'Huawei'	'5.1.1'	Lollipop	111	gb	Europe/Belgrade
10	'HUAWEI P7-L10'	'Huawei'	'5.1.1'	Lollipop	57	us	America/New_York

□ Synchronized Scrolling

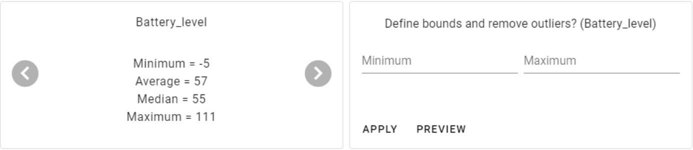
	Model	Brand	OS	Codename	Battery_level	Country	Time_zone
1	'VS500PP'	'lge'	'6.0.1'	Marshmallow	88	us	America/Chicago
2	'AO5510'	'YU'	'5.1.1'	Lollipop	59	pt	Europe/Lisbon
3	'ASUS_Z017D'	'asus'	'7.0'	Nougat	-5	us	America/Los_Angeles
4	'ASUS_X014D'	'asus'	'5.1.1'	Lollipop	41	pt	Atlantic/Madeira
5	'Nexus 5'	'google'	'6.0.1'	MARSHMALLOW	90	us	America/Los_Angeles
6	'LG-D331'	'lge'	'4.4.2'	KitKat	9	us	America/New_York
7	'Nexus 5'	'google'	'6.0.1'	Marshmallow	67	pt	Atlantic/Madeira
8	'bq Aquaris 5 HD'	'bq'	'4.2.1'		35	us	America/New_York
9	'HUAWEI SCL-L21'	'Huawei'	'5.1.1'	Lollipop	111	gb	Europe/Belgrade
10	'HUAWEI P7-L10'	'Huawei'	'5.1.1'	Lollipop	57	us	America/New_York

Fig. 1. Humanized Data Cleaning Example Interface

When the user selects one specific column, a *statistics card* is displayed in order to help summarize the contents of the chosen column. An example is shown in Figure 1 - I, where the `Codename` column is selected and a *statistics card* detailing the different data entries (and their quantification) is shown. In addition to displaying a *statistics card*, a collection of *suggestion cards* are automatically displayed (shown in Figure 1 - II), where each presents a data transformation action, based on the statistics and data inference. Following our example, the system detects two very similar values: `Marshmallow` and `MARSHMALLOW`, and thus suggests replacing one data value by the other or by a new value. In the same example, it also detected the presence of `null` or empty values, and suggests either replacing them with a new value or removing such data entries. Shown in Figure 2, is another example of such cards if one would choose the

`Battery_level` column. In this case, as the column is inferred to be numerical, a set of common numerical metrics are shown, followed by a suggestion to remove data entries based on minimum and maximum bounds. Knowing that a smartphone's battery level could not be higher than 100% nor lower than 1%, such data entries might present themselves as dirty data and could accordingly be removed through the *suggestion card*.

Such *statistic cards* and *suggestion cards* aim to remove another layer of complexity in data cleaning by automatically presenting common statistical information which users otherwise have to calculate, and by suggesting transformations based on their data. In both cases, the user would have to resort to either programming or using complex tools to gather the statistical information and apply their transformation.



Battery_level

Minimum = -5
Average = 57
Median = 55
Maximum = 111

Define bounds and remove outliers? (Battery_level)

Fig. 2. Numerical statistics and suggestion card example

3 Related Work

Several authors have proposed related approaches to make DS more accessible. Potter's Wheel provides an interactive data transformation and cleaning system that allows users to define transforms through graphical operations or examples and see the effects instantly, making it easy to experiment with different transformations [16]. Unfortunately, the project ended about 20 years ago and does not seem to have been evaluated with users.

Milo [17] and BlockPy [1] are tools that offer a block-based language for users, but focus on different aspects. While Milo aims to help users with no computer science background to only perform machine learning techniques, we propose a tool for data cleaning. BlockPy is a visual interface for the Python programming language to motivate students to start learning how to program. In our case, our visual environment is designed for data cleaning, and not a programming language interface.

Wallace et al. propose a tool to allow users with less statistical skills to make use of advanced models written using the R language [20]. Their motivation is similar to ours although their goal is to provide a graphical user interface for a given R model whilst we provide a tool specific for data cleaning tasks.

DataScience4NP is a web platform aiming to provide an intuitive user interface for users to build sequential DS workflows [11]. This system intends to perform all the steps of extracting knowledge from data, which includes data insertion, pre-processing, transformation, mining and interpretation/evaluation of results, without requiring users to program. However, similarly to Milo, this platform is focused on data mining techniques whilst ours focus on data cleaning.

Industry and open-source communities have also proposed several tools for DS. Popular tools include Microsoft PowerBi[4], Tableau (Prep)[5], Jupyter (notebooks)[6], and RapidMiner[7]. These tools allow their users to make data exploration, data mining, visualization and reporting tasks through visual interactive dashboards. However, there does not seem to exist any scientific evidence of their effectiveness amongst end user data scientists. In fact, Jupyter notebooks have been found to be messy by some users [8].

4 Conclusions

In this work we propose a platform for data cleaning/curation intended for end user data scientists. We achieve this by relying on suggestions and direct data manipulation. Currently as we're still improving upon what we have we plan to explore suggestions further and explore programming by example as a way to transform data where the user can specify input and output examples. This has the potential to easily allow users to normalize data, map it to other representations, and further remove a layer of abstraction of data and mental work for our end user data scientists. We also intend to empirically evaluate our tool comparing its usability (effectiveness, efficiency and satisfaction) against other popular tools.

References

1. Bart, A.C., Tibau, J., Tilevich, E., Shaffer, C.A., Kafura, D.: BlockPy: An Open Access Data-Science Environment for Introductory Programmers. Computer **50**(5), 18–26 (may 2017). https://doi.org/10.1109/MC.2017.132
2. Burnett, M.M.: Visual Programming. In: Wiley Encyclopedia of Electrical and Electronics Engineering. John Wiley & Sons, Inc. (dec 1999). https://doi.org/10.1002/047134608x.w1707
3. Cao, L.: Data science: A comprehensive overview. ACM Comput. Surv **50**(43) (2017). https://doi.org/10.1145/3076253
4. Gulwani, S.: Programming by examples (and its applications in data wrangling). In: Dependable Software Systems Engineering, vol. 45, pp. 137–158. IOS Press (apr 2016). https://doi.org/10.3233/978-1-61499-627-9-137
5. IBM and Business-Higher Education Forum and Burning Glass: The Quant Crunch: How the Demand for Data Science Skills Is Disrupting the Job Market (2017), https://www.ibm.com/downloads/cas/3RL3VXGA
6. Kaggle Inc.: The State of Data Science & Machine Learning (2017), https://www.kaggle.com/surveys/2017

[4] https://powerbi.microsoft.com

[5] http://tableau.com

[6] https://jupyter.org

[7] https://rapidminer.com

7. Kandel, S., Paepcke, A., Hellerstein, J.M., Heer, J.: Enterprise Data Analysis and Visualization: An Interview Study. IEEE Transactions on Visualization and Computer Graphics **18**(12), 2917–2926 (dec 2012). https://doi.org/10.1109/TVCG.2012.219
8. Kery, M.B., Radensky, M., Arya, M., John, B.E., Myers, B.A.: The Story in the Notebook. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18. vol. 2018-April, pp. 1–11. ACM Press, New York, New York, USA (apr 2018). https://doi.org/10.1145/3173574.3173748
9. Kim, M., Zimmermann, T., DeLine, R., Begel, A.: The emerging role of data scientists on software development teams. Proceedings - International Conference on Software Engineering pp. 96–107 (2016). https://doi.org/10.1145/2884781.2884783
10. Ko, A.J., Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwig, M., Scaffidi, C., Lawrance, J., Lieberman, H., Myers, B., Rosson, M.B., Rothermel, G., Shaw, M., Wiedenbeck, S.: The state of the art in end-user software engineering. ACM Computing Surveys **43**(3) (apr 2011). https://doi.org/10.1145/1922649.1922658
11. Lopes, B., Pedroso, A., Correia, J., Araujo, F., Cardoso, J., Paiva, R.P.: DataScience4NP -A Data Science Service for Non-Programmers. In: 10º Simpósio de Informática – INForum 2018 (2018)
12. Matalonga, H., Cabral, B., Castor, F., Couto, M., Pereira, R., de Sousa, S.M., Fernandes, J.P.: Greenhub farmer: real-world data for android energy mining. In: 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR). pp. 171–175. IEEE (2019)
13. Muller, M., Lange, I., Wang, D., Piorkowski, D., Tsay, J., Liao, Q.V., Dugan, C., Erickson, T.: How data science workers work with data: Discovery, capture, curation, design, creation. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. CHI '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3290605.3300356
14. Pereira, P., Cunha, J., Fernandes, J.P.: On Understanding Data Scientists. In: IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC) (2020), to appear
15. Portuguese Government: Contrato para a Legislatura com o Ensino Superior para 2020–2023 (2019), https://www.portugal.gov.pt/download-ficheiros/ficheiro.aspx?v=d2607a18-51c9-489c-a61c-1ff420dab2f0
16. Raman, V., Hellerstein, J.M.: Potter's wheel: An interactive data cleaning system. VLDB 2001 - Proceedings of 27th International Conference on Very Large Data Bases pp. 381–390 (2001)
17. Rao, A., Bihani, A., Nair, M.: Milo: A visual programming environment for Data Science Education. In: 2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC). vol. 2018-Octob, pp. 211–215. IEEE (oct 2018). https://doi.org/10.1109/VLHCC.2018.8506504
18. Refaat, M.: Data Preparation for Data Mining Using SAS. Elsevier (2007). https://doi.org/10.1016/B978-0-12-373577-5.X5000-5
19. Shneiderman, B.: Direct Manipulation: A Step Beyond Programming Languages. Computer **16**(8), 57–69 (aug 1983). https://doi.org/10.1109/MC.1983.1654471
20. Wallace, B.C., Dahabreh, I.J., Trikalinos, T.A., Lau, J., Trow, P., Schmid, C.H.: Closing the Gap between Methodologists and End-Users: R as a Computational Back-End. Journal of Statistical Software **49**(5), 1–15 (jun 2012). https://doi.org/10.18637/jss.v049.i05

21. Wongsuphasawat, K., Liu, Y., Heer, J.: Goals, Process, and Challenges of Exploratory Data Analysis: An Interview Study. arXiv preprint arXiv:1911.00568 (nov 2019)
22. Zikopoulos, P.C., DeRoos, D., Parasuraman, K., Deutsch, T., Corrigan, D., Giles, J.: Harness the power of Big Data : the IBM Big Data platform. McGraw-Hill (2013)

F

CSCW PUBLISHED ARTICLE

In the following pages is the article submitted and accepted as a short research paper for Interrogating Data Science CSCW 2020 Workshop.

Data Science For All

Jácome Cunha
jacome@di.uminho.pt
University of Minho & INESC Tec/HASLab,
Portugal

José Dias,
Paula Pereira
{a78494,a77672}@alunos.uminho.pt
University of Minho, Portugal

João P. Fernandes
jpf@dei.uc.pt
University Coimbra, Portugal

Rui Pereira
ruipereira@di.uminho.pt
INESC Tec/HASLab, Portugal

ABSTRACT

Data is everywhere and in everything we do and in many cases in massive amounts. While on its own data has little value its analysis under the lenses of data science currently supports valuable functions and systems. The problem is that the amount of data generated and the fast-growing need to analyze it is not compatible with the number of workers with the necessary skills. A possible way to mitigate this issue is to propose methodologies and tools that more people, with less programming skills can still use. In this paper we propose our vision to create such methodologies and tools.

1. INTRODUCTION

Data by itself, even if massive, has little value [11, 23, 30]. Indeed, it is the extracted information from data that has the potential to keep changing and improving our lives. However, the extracting process is quite complex and requires several tasks[1] [8], such tasks make up what is called *data science* [5]. These steps are challenging as they require a variety of skills including mathematics, statistics, machine learning, algorithms, correlation or causation [6], with experts with all such skills being hard to come by. In fact, data science related job openings stay unfilled about 10% more time than the market average [12]. Studies have shown that by 2020 the number of positions for data scientists in the USA will be of 2.7 million [12], while also advocating that academia must ensure for data literacy for all in any field of education. In fact, many countries have defined national strategies regarding data science [5]. As widely suggested by companies and governments [5, 9, 12] academia should prepare courses and degrees to capacitate the next generation of data scientists with data science skills. Additionally, researchers and industry should create methodologies and tools for non-programmers or end users to be capable of performing such activities. In this paper we pursue the latter proposal discussing and proposing ways to achieve – **data science for all** (DS4All).

[1] Such tasks include cleaning, transforming, understanding, analyzing and interpreting data

2. WHO ARE THE DATA SCIENTISTS?

A *Data Scientist*'s job is relatively recent, being recognized as such for little over a decade (although many have performed tasks similar to what data scientists nowadays do) [5]. Many working on the topic have a CS background as they have the skills and know the tools (e.g. PLs) to manipulate data. Indeed, in many job offers, employers ask for skills such as SQL, Java, or Unix [12] which are tools common to be known among workers with CS background. However, given the rapid growth of data science job openings [12], many hire workers with different backgrounds [21]. For instance, physicists and other highly quantitative disciplines are being hired for financial quantitative analysts [12].

Given this eclectic scenario we argue that the research community needs a better understanding of the people doing data science. CHI 2019 held a workshop raising this concern [19], with other researchers studying the field in detail [20, 27, 29]. We have also recently presented results of interviews with data science professions in order to further understand their skills, methodologies, difficulties and needs [21]. Such studies are fundamental to understand who are today's data scientists.

3. MINDING THE GAP BETWEEN HUMANS AND TOOLS

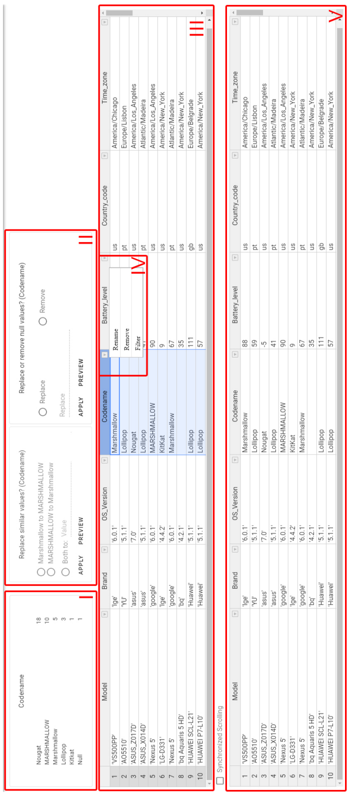
Beyond understanding data scientist it is also necessary to know the gap between them (skills and needs) and the existing tools (capabilities and abstraction).

For a CS background worker, the best tool s/he uses may be a PL like Python. Nevertheless, it may be possible to create abstractions to help improve her/is productivity. In fact, many Python programmers use Pandas [18] for data science related tasks, as this library has abstractions to help programmers become more productive. For non-CS workers the gap and need for stronger abstractions may be wider. The industry has proposed tools such as RapidMiner [22] or KNIME [1] allowing users to define data science tasks using control flow visual languages. Tableau [25] goes even further by allowing users to manipulate data through drag-and-drop actions and other intuitive graphical interactions. However, there is little scientific knowledge about the effectiveness of these approaches.

We thus propose to study the gap between data science workers and their currently used tools. It is necessary to create users' profiles characterized by skills and tool knowledge. Depending on their purpose different solutions may arise. General purpose tools, i.e. PLs, require very different skills and abstraction power when compared to tools that are closer to direct manipulation of the data, such as Tableau. We thus need to understand the users skills' and needs, as well as the tools and their requirements, and map these two sets to be able to propose impactful solutions.

4. TOWARDS HUMANIZED DATA SCIENCE

Data scientists must master several tools which can be challenging. This is even more of an issue for no-CS background workers as having end users developing software is a very well known problem [4, 14].



Codename

Nougat	18
MARSHMALLOW	10
Marshmallow	5
Lollipop	3
Kitkat	1
Null	1

Replace similar values? (Codename)

○ Marshmallow to MARSHMALLOW
○ MARSHMALLOW to Marshmallow
○ Both to: Value

Replace or remove null values? (Codename)

○ Replace ○ Remove

Replace

	Model	Brand	OS_Version	Codename	Battery_level	Country_code	Time_zone
1	'VS500PP'	'lge'	'6.0.1'	Marshmallow		us	America/Chicago
2	'AO5510'	'YU'	'5.1.1'	Lollipop		pt	Europe/Lisbon
3	'ASUS_Z017D'	'asus'	'7.0'	Nougat		us	America/Los_Angeles
4	'ASUS_X014D'	'asus'	'5.1.1'	Lollipop	41	pt	Atlantic/Madeira
5	'Nexus 5'	'google'	'6.0.1'	MARSHMALLOW	90	us	America/Los_Angeles
6	'LG-D331'	'lge'	'4.4.2'	KitKat	9	us	America/New_York
7	'Nexus 5'	'google'	'6.0.1'	Marshmallow	67	pt	Atlantic/Madeira
8	'bq Aquaris 5 HD'	'bq'	'4.2.1'		35	us	America/New_York
9	'HUAWEI SCL-L21'	'Huawei'	'5.1.1'	Lollipop	111	gb	Europe/Belgrade
10	'HUAWEI P7-L10'	'Huawei'	'5.1.1'	Lollipop	57	us	America/New_York

	Model	Brand	OS_Version	Codename	Battery_level	Country_code	Time_zone
1	'VS500PP'	'lge'	'6.0.1'	Marshmallow	88	us	America/Chicago
2	'AO5510'	'YU'	'5.1.1'	Lollipop	59	pt	Europe/Lisbon
3	'ASUS_Z017D'	'asus'	'7.0'	Nougat	-5	us	America/Los_Angeles
4	'ASUS_X014D'	'asus'	'5.1.1'	Lollipop	41	pt	Atlantic/Madeira
5	'Nexus 5'	'google'	'6.0.1'	MARSHMALLOW	90	us	America/Los_Angeles
6	'LG-D331'	'lge'	'4.4.2'	KitKat	9	us	America/New_York
7	'Nexus 5'	'google'	'6.0.1'	Marshmallow	67	pt	Atlantic/Madeira
8	'bq Aquaris 5 HD'	'bq'	'4.2.1'		35	us	America/New_York
9	'HUAWEI SCL-L21'	'Huawei'	'5.1.1'	Lollipop	111	gb	Europe/Belgrade
10	'HUAWEI P7-L10'	'Huawei'	'5.1.1'	Lollipop	57	us	America/New_York

Figure 1: Humanized Data Cleaning DS4ALL Interface

Computer science is challenging for several reasons and in particular because of its need for abstraction, something paramount for students yet very difficult [16]. In general there does not seem to exist a course specifically on abstraction, but it is mastered by practising it in software development and math courses [16]. In fact, not all computer users are willing to create abstractions as it heavily involves both investment and risk, yet programmers tend to do it more than end users [2].

On the other hand, direct manipulation offers “visibility of the object of interest; rapid, reversible, incremental actions; and replacement of complex command language syntax by direct manipulation of the object of interest” [24]. This makes users feel they master the system under use, ease the learning process, and increases the desire to explore more powerful aspects [24].

Another interesting approach to include in a data science tool for all is to allow users to define their tasks by example. Programming by example has been extensively explored by the research community with very good results [17]. It allows users to give a set of examples of the results one wishes to achieve and have a program synthesized that generalizes the results for any given input.

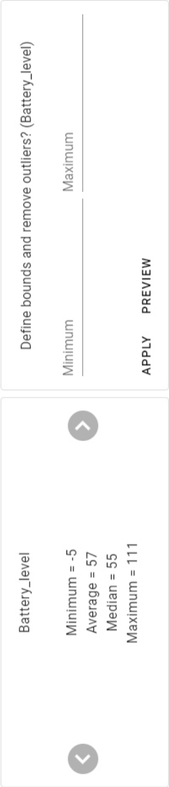
More generally, all these approaches have the common characteristic of easing the development of software, specially for end users or non-programmers, which is the case of many data science workers.

Based on the following facts: a) programming is difficult in part due to abstraction [2]; b) learning to abstract is difficult [16]; c) direct manipulation allows for mastering a system in use [24]; and d) visual programming, PBE, and live programming intend to ease programming [3, 10, 26]; **we advocate that a visual environment for direct manipulation of data is the best tool one can desire for allowing anyone to perform data science, that is, to achieve a data science for all**.

5. TOWARDS A HUMANIZED DATA SCIENCE TOOL

Several paths may be taken when developing a visual environment for the direct manipulation of data. We have previously proposed a prototype, shown in Figure 1, for a humanized data cleaning interface [7]. We believe that data should naturally be represented in a way users can actually see and manipulate it using some tabular format. Indeed, presented in Figure 1 - V (*Original dataset*), is the original and unaltered dataset shown at all times, allowing the end user to better accompany their transformations. All such transformations are shown and previewed in Figure 1 - III (*Preview dataset*). This side-by-side look at the dataset before and after applying changes aims to help remove a level of abstraction of how data will be changed, and directly present such actions. At any point, the user may directly manipulate the data within the *Preview dataset*, such as updating cell values, or through a drop down menu (as shown in Figure 1 - IV) to allow changes or filtering data on a specific column.

When selecting a column, a *statistics card* is displayed to help summarize the contents of the chosen column, as shown in Figure 1 - I, where the Codename column is selected and details of the different data entries are shown. Additionally, a collection of *suggestion cards* are automatically displayed (shown in Figure 1 - II), where each presents a data transformation action, based on statistics and



Battery_level

Minimum = -5
Average = 57
Median = 55
Maximum = 111

Define bounds and remove outliers? (Battery_level)

Figure 2: Numerical statistics and suggestion card example

data inference. For example, here the system detects two very similar values: `Marshmallow` and `MARSHMALLOW`, and thus suggests replacing one data value by the other or by a new value. Figure 2 shows another example of such cards if choosing the `Battery_level` numerical column.

Such *statistic cards* and *suggestion cards* aim to remove a layer of complexity in data cleaning by automatically presenting common statistical information, which users otherwise have to calculate, and by suggesting transformations. In both cases, the user would have to resort to either programming or using complex tools to gather the statistical information and apply their transformation.

6. COMMUNICATION, DOCUMENTATION, REUSE AND MORE

Data science tends to be an iterative activity [28]. For instance, after a first cleaning phase, one may still find data quality issues during analysis, and thus the need for more cleaning. If the cleaning and analysis phases are done by different teams, then they need to **communicate** with each other. On one hand, the analysis team needs to tell the cleaning team that some issues still need work, while on the other hand, the cleaning team needs to describe what was changed from the original data set. Additionally, the performed operations must be documented if at any point one may need to debug or understand how the data reached the current state. This is specially relevant if someone other who performed such changes wants to understand what happened[2]. Thus, a data scientist needs to additionally document the transformations. In fact, PL Notebooks mix code, execution results and text annotations. However, studies show that such Notebooks have been found to be messy by users [13]. Thus, a tool designed for end users should provide a proper way (e.g. a language, possibly visual) to easily allow the description of data changes to be communicated back and forth between different teams. Studies have also shown that data scientists tend to **reuse** code [15]. However, in some tools this is the common copy&paste [15] which is dangerous as one duplicates code. Thus, good tools need to provide support for proper reuse by allowing users to define some kind of function instead of promoting code duplication.

In fact, we argue these issues – communicate, document, and reuse – are quite connected. In fact, they can be seen as different perspectives over the same needs. To communicate between the different teams, documentation is needed. If the documentation of the operations is performed using a language with proper semantics, then these operations can be reused. Thus, we propose that a tool for end user data scientists should provide a language to document such operations, or the ones that need to be done, so everyone can understand what was done to the data. The language should have a semantics so it can be used to re-execute the operations. Such a language could be inspired by block-based languages which are being used with quite success among novice programmers. As the operations are being executed in the data, the tool could build a block-based program with the operation. This could be used by everyone to read what happened or to specify what should be done. Moreover, could be used to re-execute a set of operations.

[2] For example when a new member comes to the team

REFERENCES

[1] KNIME AG. last visited 1/6/2020. KNIME. www.knime.com.

[2] Alan F. Blackwell. 2001. See What You Need: Helping End-users to Build Abstractions. *Journal of Visual Languages & Computing* 12, 5 (2001), 475 – 499. https://doi.org/10.1006/jvlc.2001.0216

[3] Margaret M Burnett. 2001. Visual programming. *Wiley Encyclopedia of Electrical and Electronics Engineering* (2001).

[4] Margaret M. Burnett and Brad A. Myers. 2014. Future of End-User Software Engineering: Beyond the Silos. In *Proceedings of the on Future of Software Engineering* (Hyderabad, India) *(FOSE 2014)*. ACM, New York, NY, USA, 201–211. https://doi.org/10.1145/2593882.2593896

[5] Longbing Cao. 2017. Data Science: A Comprehensive Overview. *ACM Computing Surveys (CSUR)* 50, 3 (June 2017), 42. https://doi.org/10.1145/3076253

[6] Vasant Dhar. 2013. Data Science and Prediction. *Commun. ACM* 56, 12 (Dec. 2013), 64–73. https://doi.org/10.1145/2500499

[7] Jose Dias, Jacome Cunha, and Rui Pereira. 2020. Data Curation: Towards a Tool for All. In *Proceedings of the 22nd HCI International Conference on Human-Computer Interaction* (Compenhagen, Denmark) *(HCII '20)*.

[8] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. 1996. The KDD Process for Extracting Useful Knowledge from Volumes of Data. *Commun. ACM* 39, 11 (Nov. 1996), 27–34. https://doi.org/10.1145/240455.240464

[9] Portuguese Government. 2019. Contrato para a Legislatura com o Ensino Superior para 2020–2023. https://www.portugal.gov.pt/download-ficheiros/ficheiro.aspx?v=d2607a18-51c9-489c-a61c-1ff420dab2f0.

[10] Sumit Gulwani. 2016. Programming by Examples - and its applications in Data Wrangling. *Dependable Software Systems Engineering* 45 (2016), 137–158. https://doi.org/10.3233/978-1-61499-627-9-137

[11] Tim Hoyland, Chris Spafford, and Andrew Medland. 2016. Oliver Wyman's 2016 MRO Survey. https://www.oliverwyman.com/our-expertise/insights/2016/apr/mro-survey-2016.html.

[12] IBM, Business-Higher Education Forum, and Burning Glass. 2017. The Quant Crunch: How the Demand for Data Science Skills Is Disrupting the Job Market. https://www.ibm.com/downloads/cas/3RL3VXGA.

[13] Mary Beth Kery, Marissa Radensky, Mahima Arya, Bonnie E. John, and Brad A. Myers. 2018. The Story in the Notebook: Exploratory Data Science Using a Literate Programming Tool. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3173574.3173748

[14] Andrew J. Ko, Robin Abraham, Laura Beckwith, Alan Blackwell, Margaret Burnett, Martin Erwig, Chris Scaffidi, Joseph Lawrance, Henry Lieberman, Brad Myers, and et al. 2011. The State of the Art in End-User Software Engineering. *ACM Comput. Surv.* 43, 3, Article 21 (April 2011), 44 pages. https://doi.org/10.1145/1922649.1922658

[15] A. P. Koenzen, N. A. Ernst, and M. D. Storey. 2020. Code Duplication and Reuse in Jupyter Notebooks. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 1–9.

[16] Jeff Kramer. 2007. Is abstraction the key to computing? *Commun. ACM* 50, 4 (2007), 36–42.

[17] Henry Lieberman. 2001. *Your wish is my command: Programming by example.* Morgan Kaufmann.

[18] Wes McKinney. last visited 1/6/2020. Pandas - Python Data Analysis Library. https://pandas.pydata.org.

[19] Michael Muller, Melanie Feinberg, Timothy George, Steven J. Jackson, Bonnie E. John, Mary Beth Kery, and Samir Passi. 2019. Human-Centered Study of Data Science Work Practices. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI EA '19)*. Association for Computing Machinery, New York, NY, USA, Article W15, 8 pages. https://doi.org/10.1145/3290607.3299018

[20] Michael Muller, Ingrid Lange, Dakuo Wang, David Piorkowski, Jason Tsay, Q. Vera Liao, Casey Dugan, and Thomas Erickson. 2019. How Data Science Workers Work with Data: Discovery, Capture, Curation, Design, Creation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for

Computing Machinery, New York, NY, USA, Article 126, 15 pages. https://doi.org/10.1145/3290605.3300356

[21] Paula Pereira, Jácome Cunha, and Joao Paulo Fernandes. 2020. On Understanding Data Scientists. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 1–5.

[22] RapidMiner. last visited 1/6/2020. RapidMiner. rapidminer.com.

[23] David Reinsel, John Gantz, and John Rydning. 2018. The Digitization of the World – From Edge to Core. https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf.

[24] Ben Shneiderman. 1983. Direct Manipulation: A Step Beyond Programming Languages. *Computer* 16, 8 (August 1983), 57–69.

[25] Tableau Software. last visited 1/6/2020. Tableau Desktop. https://www.tableau.com/products/desktop.

[26] Steven L. Tanimoto. 2013. A Perspective on the Evolution of Live Programming. In *Proceedings of the 1st International Workshop on Live Programming* (San Francisco, California) *(LIVE '13)*. IEEE Press, 31–34.

[27] Dakuo Wang, Justin D. Weisz, Michael Muller, Parikshit Ram, Werner Geyer, Casey Dugan, Yla Tausczik, Horst Samulowitz, and Alexander Gray. 2019. Human-AI Collaboration in Data Science: Exploring Data Scientists' Perceptions of Automated AI. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 211 (Nov. 2019), 24 pages. https://doi.org/10.1145/3359313

[28] Kanit Wongsuphasawat, Yang Liu, and Jeffrey Heer. 2019. Goals, Process, and Challenges of Exploratory Data Analysis: An Interview Study. *arXiv* 1911.00568 (2019). http://idl.cs.washington.edu/papers/eda-goals-process

[29] Amy X. Zhang, Michael Muller, and Dakuo Wang. 2020. How do Data Science Workers Collaborate? Roles, Workflows, and Tools. arXiv. arXiv:2001.06684 http://arxiv.org/abs/2001.06684

[30] Paul Zikopoulos, Dirk Deroos, Krishnan Parasuraman, Thomas Deutsch, James Giles, and David Corrigan. 2012. *Harness the power of big data The IBM big data platform*. McGraw Hill Professional.