Characteristics of Open Data CSV Files

Johann Mitlöhner,* Sebastian Neumaier,† Jürgen Umbrich,‡ and Axel Polleres§
Vienna University of Economics and Business, Vienna, Austria
Email: *johann.mitloehner@wu.ac.at, †sebastian.neumaier@wu.ac.at, ‡juergen.umbrich@wu.ac.at, §axel.polleres@wu.ac.at

***Abstract*—This work analyzes an Open Data corpus containing 200K tabular resources with a total file size of 413 GB from a data consumer perspective. Our study shows that ∼10 % of the resources in Open Data portals are labelled as a tabular data of which only 50 % can be considered CSV files. The study inspects the general shape of these tabular data, reports on column and row distribution, analyses the availability of (multiple) header rows and if a file contains multiple tables. In addition, we inspect and analyze the table column types, detect missing values and report about the distribution of the values.**

I. Motivation

The following work presents a data corpus consisting of tabular Open Data sources and studies the characteristics and properties of the files from a consumers point of view. Earlier reports showed that the CSV (comma-separated values) format is the predominant format in the Open Data landscape [8]. The main reason is the simplicity and independence of this format: it stores tabular data in plain text where each line of the file is a data record. Each record consists of one or more fields which are separated by a delimiter, typically a comma.

Comma-separated values as a file format has been used for decades for interchanging database information between machines and pre-dates the Open Data initiative. Historically, the CSV format developed out of a need for such an exchange format without an initial formalization; therefore, there are many variations in use and there is not a single, fully specified, "CSV" format. In 2005 the IEFT published a first attempt to standardize CSV by proposing a strict dialect and implementation:

- Each record is located on a line, delimited by a line break.
- The use of an header line (appearing as the first line of the file) is optional, however, "the presence or absence of the header line should be indicated via the optional header parameter of this MIME type" [7].
- The fields of a record are separated by a comma and each line contains the same number of fields.
- A field may be enclosed in double quotes so that commas and and line breaks can be used within fields. If double quotes are used in already double-quotes enclosed fields, it must be escaped by another double quote (e.g., the field "b""b" contains one double quote within the two b's).

However, one can observe many variants of this specification, and nowadays CSV stands more for "character-separated-values". This work studies and analyses the characteristics of 200K CSV files listed in 232 Open Data portals, attributing to a total file size of 413 GB. To the best of our knowledge, this is the first large-scale study of Open Data CSV files. Section II introduces the statistics of our data corpus, detailing the availability of the files per portal software and publisher domain and reporting on the file size distribution. Next, Section III details the amount of files that can be parsed as CSV files (using straight forward heuristics to detect delimiters, comment and header lines) and reports about typical CSV dialects and table shapes. In Section IV, we inspect the tables in more detail and report about column data types and various measures on numerical data columns using the statistics package R. A discussion of related efforts and the implications for data consumers is presented in Section V. Section VI concludes our work by presenting the core findings.

TABLE I: Data corpus statistics

	all	CKAN	Socrata	OpenDataSoft
resources	3 571 085	3 436 288	125 514	9283
valid http	3 558 823	3 424 227	125 514	9082
unique	1 995 742	1 898 804	109 650	9082
labelled as csv	200 939	185 946	18 275	2166
HTTP 200 OK	141 738	126 776	12 864	2098
parsed as csv	104 826	95 249	10 612	1918

II. Data Corpus

We collected our data corpus from 232 Open Data portals to study the characteristics of Open Data CSV files. The list of data portals and their meta data was extracted from the Open Data Portal Watch framework [8], using the snapshot of the second week of May 2016.

A. Corpus statistics

We parsed 950 117 dataset descriptions and extracted 1 995 742 distinct resources. In total, 200 939 unique resources are annotated as a "CSV" files or contain either the tokens "*.csv*" in the URL or the token "csv" in the query parameters (assuming that the URL defines the export format to be CSV). 141 694 files were successfully downloaded of which 104 826 are considered to be CSV files (cf. Section III). The basic statistics are summarized in Table I.

B. Download statistics

We tried to downloaded the 200 939 resources and stored the HTTP response header information. In total, we successfully downloaded 141 694 documents (73 %) with a total of 413 GB content. In Table II we show the distribution of received HTTP Status codes and exceptions which are grouped into typical error classes. A set of 44 838 (23 %) documents are

TABLE III: Header content-type distribution

№	MIME-TYPE	COUNT	
1	application/octet-stream	63 350	(44.71 %)
2	text/csv	49 451	(34.90 %)
3	application/zip	9636	(6.80 %)
4	text/html	8118	(5.73 %)
5	text/plain	2825	(1.99 %)
6	application/csv	2453	(1.73 %)
7	application/vnd.ms-excel	2423	(1.71 %)
8	text/x-comma-separated-values	752	(0.53 %)
9	text/xml	582	(0.41 %)
10	application/x-zip-compressed	496	(0.35 %)

not available any more for download (more details about this in the following).

The majority of the downloaded content has a per-file size of less than 100 kB, the biggest file had a file size of 25 GB (cf. Figure 1).

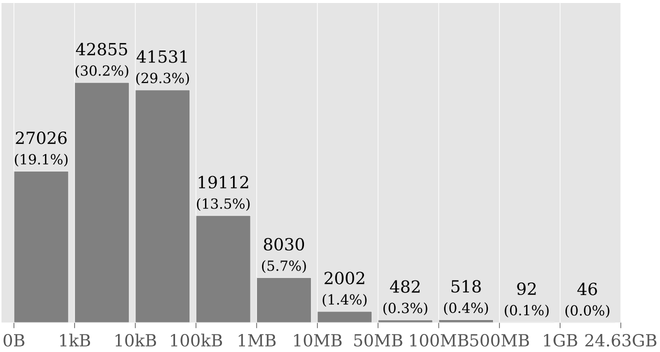


Fig. 1: File size distribution

Table III provides an overview of the reported content-types in the HTTP Response headers with status code 200. Interestingly, we see that the header content-type cannot really be considered as an indication if the file is in CSV format.

C. Domain statistics

Next we group the 200 939 extracted resource URIs by their domain and analyzed the percentage of available resources per domain. In total we discovered 2301 unique domains. The 10 domains with the most resources are listed in Table IV together with their ratio of available resources. To our surprise, we see three domains with the availability ratio of less than 10 percent. To get the full overview, we plotted the availability for each of the 2301 domains in Figure 2. Another view is provided as a dot plot in Figure 3. The domains are binned by their availability ratio (y-axis). Next, for each bin, we ordered the domains by their size and split them into 10 equal size groups (ten dots on the x-axis). Each dot represents the average size of 1/10 of the domain for that bin. The two plots show that there exists a couple of smaller domains for which the files are not available, but also some larger domains (e.g. the dot representing ten domains with an average availability of 0.4 -0.5 and an average document number of 3000, cf. Figure 3).

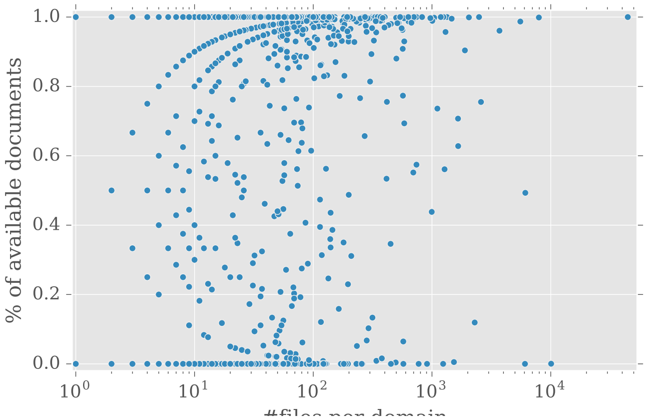


Fig. 2: Availability of documents on domains

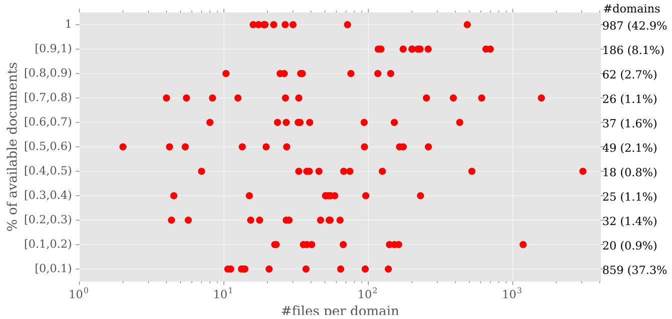


Fig. 3: Availability of documents on domains as dotplot. Each dot represents the average over 1/10 of the total elements in the bin.

III. Parsed CSV files

This section reports on the challenges in parsing our corpus using standard Python libraries. As a first step, we tried to identify the correct encoding of the file using the file-magic library (which uses the underlying Unix file library).[1] Next, we assumed the properly encoded content is CSV-like and guessed the delimiter, line ending and used quotation char using the heuristics in the standard CSV library.[2]

In order to properly detect the shape of tables in CSV documents we checked if the documents contained any preceding comment lines or consisted of multiple tables. For instance, consider Table V (found on an Italian Open Data portal[3]): here we observed a CSV document with leading comment lines, typically indicated by a single field containing a value. Table V also consists of multiple tables within one CSV file: the document holds three tables which are separated by repeated line breaks. In other documents we also observed multi-tables where the number of delimiters remain consistent for each line

[1] http://pubs.opengroup.org/onlinepubs/9699919799/utilities/file.html

[2] https://docs.python.org/2/library/csv.html\#csv.Sniffer

[3] http://dati.veneto.it/dataset/fb289cb7-ba04-4eb8-a2cf-11a8a0fb0e3c/resource/72e9df90-a114-40a7-9889-b907064e2e39/download/c0201030concinquinantecoinfo.csv, last accessed 2015-06-06

TABLE II: Download statistics.

	OK	NA	Connection	SSL	URI	Sever	Content	IP	Intern
count	141 738	44 838	2350	1386	959	651	613	530	132
	73.36%	23.21%	1.22%	0.72%	0.50%	0.34%	0.32%	0.27%	0.07%

TABLE IV: Top-10 domains and their availability ratio.

№	domain	count	avail
1	cdn1.sdlabs.ru	44 558.0	(100.00 %)
2	ec.europa.eu	10 075.0	(0.04 %)
3	www20.statcan.gc.ca	7946.0	(99.87 %)
4	www.gov.uk	6108.0	(49.31 %)
5	www.e-stat.go.jp	6067.0	(0.00 %)
6	opendata.socrata.com	5544.0	(98.70 %)
7	webarchive.nationalarchives.gov.uk	3700.0	(96.05 %)
8	www.landesdatenbank.nrw.de	2585.0	(75.51 %)
9	aimis-simia.agr.gc.ca	2489.0	(100.00 %)
10	www.statistik.sachsen.de	2289.0	(11.93 %)

(i.e., the tables are separated by empty records/cell values).

Comment lines, multi-tables and multiple header rows potentially occur in documents produced by spreadsheet applications such as Microsoft Excel. With the help of such a tool it is easy to keep an overview even if there are multiple tables within one sheet (cf. Table VI which displays the content of Table V in Microsoft Excel). Next, we discuss our heuristics to detect preceding comment rows, multiple header rows, and multiple tables in a single CSV document.

A. Header detection heuristic

Most CSV files provide header rows to describe a specific column. However, even in the RFC specification [7] the use of an header row is optional. In certain cases we observed multiple header rows in a document. In particular, this can be the case when exporting a spreadsheet with joined fields to CSV.

An absolutely accurate detection of header rows is impossible, due to the lack of syntactic description of CSVs. In order to get an overview over missing headers, and multi-headers respectively, we implemented a heuristic header detection algorithm which is based on two primitive cell types, i.e., NUMERIC (integer or float value), STRING:

```
def detectHeader(table):
    i = 0
    // we assume string values as headers
    if NUMERIC in first row:
        return 0
    for row in table:
        i = i + 1
        if types(row) all STRING and
        types(next row) not all STRING:
            return i
        // we assume at most two header rows
        if i > 2:
            return 1
    return i
```

Algorithm 1: Header detection.

In the above algorithm we assume a missing header row if there are numeric values in the first row. Obviously, there are cases where numeric values in the first row can make sense: for instance, consider a transposed table which holds the header values in the first column instead of the first row, or a table describing data for different years, using the years as headers. However, the algorithm shall serve as a heuristic in order to get an indication for the use of header rows.

Next our header-detection-algorithm checks if the primitive types of the current row differ from the next row. If the current row contains only strings but the next row contains some numeric cell values we assume that we found the end of some multi-header lines. However, in case of more than two detected header rows our algorithm terminates and uses the first row as the default header. This is for instance the case if all cells in all the first rows consist of string values.

B. Multi-table and comment line detection heuristic

Simple heuristics are used to detect comment lines and multi-tables. The algorithm considers as comment lines all lines at the beginning of a file with zero or one delimiter. This heuristic was developed based on a manual inspection of randomly selected CSV files.

Regarding multi-table detection, the developed heuristic first parses the files and builds groups for consecutive lines with the same column number (e.g. (*rows*,*cols*)). For instance, a table with 20 rows and 10 columns would be represented as one single group (20,10), while a multi-table with the first table having 20 rows and 10 columns and the second table having 10 rows and 5 columns would result in the groups ((20,10), (10,5)). The algorithm considers a CSV file containing multiple tables if there exists more than one group with more than one row and different cell/column numbers.

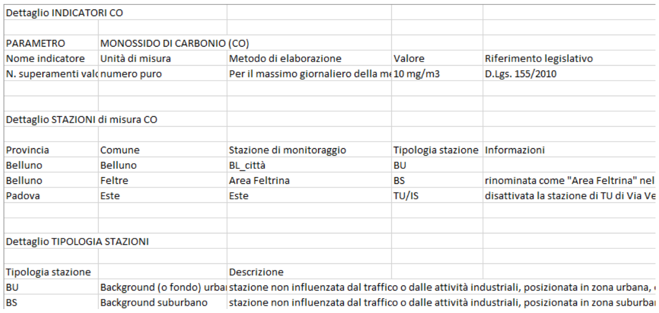
C. Corpus Results

In order to convert our corpus of CSV files to a consistent representation, i.e., a single header table of regular shape without comment lines, we applied the introduced heuristics. Out of a total of 141 738 files, which are marked as CSV in the metadata, we successfully parsed 104 826 CSV documents. As possible delimiters we allowed the following characters: , \t ; # : | ^ . The Python standard CSV library also uses a single whitespace character as a possible delimiter. We excluded this choice since it is an extremely uncommon delimiter with an high rate of incorrectly guessed files. Table VII shows the results of this parsing process in which 73.9 % of the downloaded files could be successfully parsed as a table (or multi-table with less than 5 tables). The majority of the valid CSV files contain a single table. The main parsing error was that the delimiter could not be detected, followed by

TABLE V: Example CSV document.

```
Dettaglio INDICATORI CO;;;;;

PARAMETRO;MONOSSIDO DI CARBONIO (CO);;;;
Nome indicatore;Unit  di misura;Metodo di elaborazione;Valore;Riferimento l ...
N. superamenti valore limite protezione salute umana (media mobile 8h);numer ...

Dettaglio STAZIONI di misura CO;;;;;

Provincia;Comune;Stazione di monitoraggio;Tipologia stazione;Informazioni;
Belluno;Belluno;BL_citt ;BU;;
Belluno;Feltre;Area Feltrina;BS;"rinominata come ""Area Feltrina"" nel 2010 ...
Padova;Este;Este;TU/IS;disattivata la stazione di TU di Via Versori in data ...

Dettaglio TIPOLOGIA STAZIONI;;;;

Tipologia stazione;;Descrizione;;
BU;Background (o fondo) urbano;stazione non influenzata dal traffico o dalle ...
BS;Background suburbano;stazione non influenzata dal traffico o dalle attivi ...
```

TABLE VI: Viewed with spreadsheet.



Dettaglio INDICATORI CO				
PARAMETRO	MONOSSIDO DI CARBONIO (CO)			
Nome indicatore	Unità di misura	Metodo di elaborazione	Valore	Riferimento legislativo
N. superamenti valo	numero puro	Per il massimo giornaliero della me	10 mg/m3	D.Lgs. 155/2010
Dettaglio STAZIONI di misura CO				
Provincia	Comune	Stazione di monitoraggio	Tipologia stazione	Informazioni
Belluno	Belluno	BL_città	BU	
Belluno	Feltre	Area Feltrina	BS	rinominata come "Area Feltrina" nel
Padova	Este	Este	TU/IS	disattivata la stazione di TU di Via Ve
Dettaglio TIPOLOGIA STAZIONI				
Tipologia stazione		Descrizione		
BU	Background (o fondo) urba	stazione non influenzata dal traffico o dalle attività industriali, posizionata in zona urbana,		
BS	Background suburbano	stazione non influenzata dal traffico o dalle attività industriali, posizionata in zona suburba		

compressed (zip) files, which we did not handle and parse in this study.

TABLE VII: Overall parse process statistics

	COUNT	
parsed	141 738	100 %
without errors	104 826	73.9 %
single tables	102 210	97.5 %
two tables	2279	2.2 %
three tables	337	0.3 %
with errors	36 912	26.1 %
ignored (zip files)	9925	26.88 %
too many tables	1717	4.6 %
no delimiter	19 511	52.8 %
others	5759	15.6 %

a) CSV dialect: Table VIII shows the distribution of the detected delimiter in total and grouped by the underlying portal software. The most common delimiter symbol is the comma (~ 70%) followed by the semi-colon (~ 8%). In addition to these two delimiters there are also 1153 tab-separated-value files and only a minute proportion of files using other delimiter symbols (e.g., | and #).

The OpenDataSoft software integrates and displays tabular data in the framework and allows the export of these tables in different formats. Surprisingly, the OpenDataSoft framework mainly returns semicolon-separated files when exporting CSV. A possible reason is that OpenDataSoft is mainly deployed in France (7 out of 11 monitored portals) where semicolon is the commonly used delimiter (e.g., Excel saves a spreadsheet as semicolon separated file under French location settings).

In order to further look into the deviating use of delimiter in different countries we grouped the portals by their origin location. In Table IX we list the top 5 countries and their use of comma and semicolon delimiters. Beside the French portals also in German and Austrian portals there are more semicolon-separated files. In principle, it can be assumed that this is highly influenced by the use of comma as the decimal mark.

b) Multi-tables & comment lines: By applying our multi-table detection algorithm (cf. Section III-B) we observed 102 210 CSV files containing a single tables, 2279 files with two and 337 with three tables (cf. Table VII), which results in a total of 107 779 tables.

TABLE VIII: Distribution of delimiter

	All	CKAN	Socrata	OpenDataSoft
,	96 580	83 063	13 515	2
;	11 011	9123	1	**1887**
\t	1153	1152	1	-
:	251	202	27	22
|	194	194	-	-
#	62	61	1	-
ˆ	3	3	-	-

TABLE IX: Use of semicolon vs. comma in top-5 countries

	comma (,)	**semicolon (;)**
RUS	45 181	175
GBR	17 590	8
USA	16 331	148
DEU	377	4551
AUS	3655	10

In Table X we list the tables with a certain number of detected comment lines and header rows. As the results show, the majority of the tables have no comment line and one header row. Around 11k documents have no detectable header row (Section III-A), 92 970 have one header row and 3002 tables contain two header rows.

TABLE X: Comment and header rows

	HEADER ROWS		
COMMENT LINES	0	1	2
0	11 065	86 846	1251
1	279	3565	638
2	66	706	105
3	26	231	27
4	78	352	108
5	293	1270	873

c) Columns-Rows shape: Table XI contains the descriptive statistics for the row and column counts (excluding tables with more rows/columns than the 95% quantile). This covers 88% of the tables. We can see that the average Open Data CSV table has around 379 rows and 14 columns. Figure 4 shows the distribution of the tables for various row/column shapes (rows are binned). Surprisingly, we see a large number of tables with exactly one 1 row and different columns. A manual inspection of randomly selected files with 1 row indicates that these are exports from Socrata portals with test data.[4]

TABLE XI: Statistics about number of rows and columns (max 95% quantile)

Statistics	COUNT	MEAN	STD	MIN	25%	50%	75%	MAX
rows	94 958	379	1091	1	5	25	146	8684
columns	94 958	14	7	2	8	19	20	26

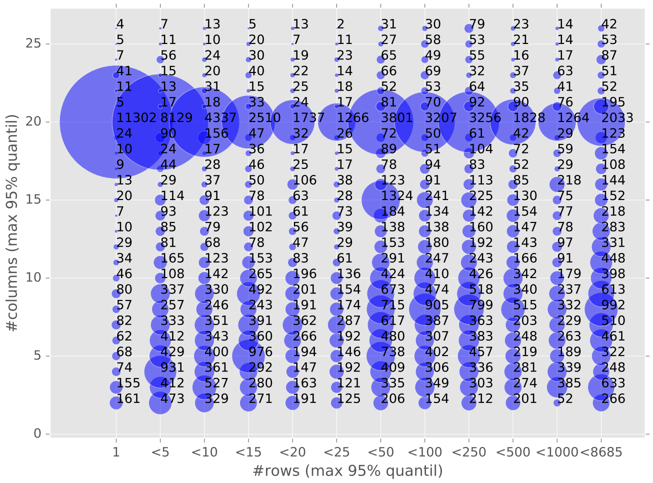


Fig. 4: Row (binned) vs. column count, including number of tables

d) Readability of headers: In order to get better insights into the readability of the header values we analyzed the structure and composition of the CSVs' headers. The results in Table XII are based on 92 970 CSVs where we detected a single header row, consisting of a total of 1.7M values. In Table XII we distinguish between values consisting of *multiple* words, values containing *underscores*, values written in *camel case* and the remaining values, which we assume consist of a *single* word.

TABLE XII: Composition of header values.

Header	Count	
Total (single row header)	1 735 807	
Underscore	707 558	(40.7%)
Single word	578 088	(33.3%)
Multiple words	302 474	(17.4%)
Camel Case	147 687	(8.5%)
WordNet Mapping	186 531	(10.7%)

Interestingly, about 50% of the inspected header values (855k) were composed of camel case and underscore separated words. This fact suggests that many of these headers spring from dumps of relational databases and therefore some of the headers might also exist as labels in other tables. However, this stringing together of words impedes an automated mapping and linkage to potential entities and concepts.

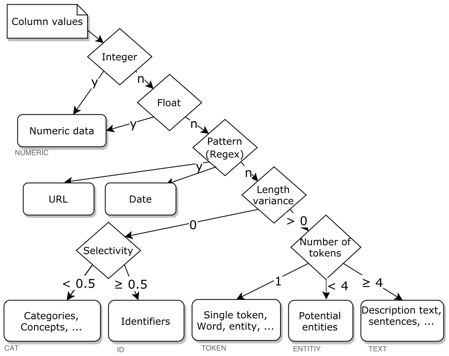


Fig. 5: Column type heuristic

Additionally, we experimentally investigated the readability of the headers by a simple mapping of the values to WordNet [4], a lexical database of English words grouped into sets of synonyms. Prior to the WordNet lookup we split the headers on underscores and camel case and apply an automated stemming (i.e., reducing words to their word stem). This resulted in 186 531 mappings (~ 11%, cf. Table XII).

D. Classifying column values

To get an idea of the distribution of general data types in the columns, we applied a basic type classification heuristic as depicted in Figure 5. As a first step of this algorithm we check if all values of a column are integers or decimals (i.e., floating point numbers). We also allow the comma as a decimal mark. If yes we classify the column as "numeric", otherwise we apply regular expressions which check if the values are valid URLs or date values.[5]

If the column does not consist of URLs or dates we compute the length variance of the string values, i.e., the varying length of the values. If there is no variance, we additionally consider the selectivity of the values. In case the values are rather unique (we use a selectivity of 0.5 as threshold) we classify the column as ID-column, otherwise we call it CAT-column (meaning categories or concepts).

If we observe higher length variance of the values we also take the number of tokens/words into account. Here, we use 4 words as a threshold to consider a column as "potentially entities"-column or as a column holding description text and sentences.

Table XIII lists the results of this column classification heuristic over our corpus. Surprisingly, we see that the majority of the columns are containing either `numeric` or `id` data.

[4] One of those files is https://performance.smcgov.org/api/views/mncj-7pjs/rows.csv?accesstype=download

[5] Date parsing is done by using the Python dateutil package, https://pypi.python.org/pypi/python-dateutil.

TABLE XIII: Column type distribution

NUMERIC	687 896
ID	626 568
TOKEN	411 976
ENTITY	327 223
TEXT	158 956
DATE	62 589
URL	40 586

First Column	Remaining Columns	Number of Tables
Character	Character	8410
Character	Mixed	26573
Numeric	Mixed	45549
Date	Mixed	5588
Numeric	Numeric	601

TABLE XIV: A selection of structural table types derived from data types and positions of columns

IV. CSV Data Characteristics

In this section the characteristics of the data read from the CSV files are described by studying data types and various measures on numerical data. For this purpose data columns are analyzed individually as they were read in from the files. No attempt is made here to look for patterns indicating correspondences of columns within the same CSV file, or among different CSV files.

After the significant amount of pre-processing described in the previous sections the results presented below show the success or failure of accessing the data via the statistics package R. The function read.csv() was used on only those CSV files that were already conforming to single header format. The emphasis here is on the point of view of the statistics package i.e. how the data are processed by R.

For performance reasons some very large CSV files were omitted, and the following constraints were set for the subsequent computations:

- rows $\leq$ 10 000
- columns $\leq$ 100

Problem cases such as varying number of entries per line or multiple tables within the same file were no longer present, therefore practically all CSV files were successfully read; only 11 failures were reported. This resulted in a total of 87 875 tables with 1 345 626 columns being analyzed.

In the following listings the data types integer and double as assigned by R were not distinguished but are both listed as type 'numeric'.

A. Table Types

To gain a little more insight into the structure of tables in the repository a survey on the column types and positions is presented in table XIV. The more frequent specific table structures among all the combinations of positions and types are listed:

- Tables containing only character columns make up about 10% of the total. For these tables no further properties were analyzed in this work.
- Tables containing a character column in the first position and mixed data types in the remaining columns may be interpreted as describing the text values in the first column; these tables make up about 30% of the CSV tables analyzed.
- Tables containing a number in the first column and mixed data in the remaining columns make up the largest fraction of about 52% of the tables. These tables may seem like textbook examples of consecutive numeric keys and corresponding description; however, the small amount of columns containing consecutive numbers refutes this idea (see below).
- A similar category to the above is the case with date columns at the first position; these make up about 6% of the tables.
- Another category of interest are all numeric tables which occur rather infrequently, making up less than 1%.

Individual columns are further analyzed in the following section.

Type	Number	Complete	$n = 0$	$n = 1$	$n = 2$	$n \geq 3$
logical	982	935	0	33	8	941
empty	38230	0	38230	0	0	0
date	59645	58874	0	4965	3135	51545
character	601923	601025	0	37024	22251	542648
numeric	644846	562129	0	63695	39144	542007

TABLE XV: Columns by type and number of cases

B. Column Data by Types and Number of Cases

To study the data properties of the individual columns in the CSV tables statistics were collected on data type and number of cases as recognized by the statistics package R; the results are presented in table XV.

Note that here the number of cases and hence the terms complete and empty in table XV refer to the number of values successfully read into the statistics package, with unreadable CSV cell entries marked as NA and removed.

- The number of logical value columns is very small, which is not surprising for CSV data published in Open Data portals. This type of data, if published at all, might be encoded in numeric values 0 and 1, or character strings other than TRUE/FALSE. In order to further investigate this idea the numeric columns were tested for containing 0/1 only; as expected, this pattern was also rarely found (see below).
- About 3% of the data columns were found to be completely empty. Those columns that contained any values at all were found to be complete in the large majority for all data types i.e. columns missing only some values were relatively rare.
- About 4% of the data columns contained date values only. These data were found by applying a very simple regular expression checking for a small number of patterns like yyyy-mm-dd and various combinations with commonly used separation characters.

n	Columns	Sorted	Consecutive	Identical	All 0/1
3	28405	16578	62	5336	374
4-10	109778	25089	311	13148	1109
11-100	219232	21218	888	17254	1538
>100	184592	14765	491	12115	2973
Totals	542007	77650	1752	47853	5994

TABLE XVI: Apparent patterns for numeric columns with number of cases ≥ 3

n	Columns	Md Cases	Md Skewness	Md Kurtosis
3	22993	3	0.299	1.500
4-10	96226	6	0.419	2.270
11-100	200800	30	1.041	4.312
>100	170927	447	1.643	7.883
Totals	490946	39	0.798	3.382

TABLE XVII: Medians for Cases, Skewness, and Kurtosis for columns with no apparent pattern

- Character columns made up 45% of the total number of data columns. No further analysis was made on these values. However, in an attempt to gain some insight into the function of these values the tables were categorized according to the position of these character columns, as described in the next section.
- Numeric columns made up the largest share and were detected in about 48% of the data columns. Probably not surprisingly these columns showed the largest amount of missing values. Numeric columns are further studied below.

C. Properties of Numeric Values

In order to study the properties of the numeric columns only those columns containing at least three cases were studied, since most of the measures calculated would not be meaningful with fewer cases.

The columns containing numeric values were analyzed for apparent patterns by taking a number of properties and tabulating them in table XVI.

- About 14% of the data columns turned out to be in sorted order. This may be the result of automated generation by database query with an ORDER BY clause.
- Only 0.3% of the data columns contained values all in consecutive order. Such values would be indicative of a description, providing associated records for a range of key values.
- About 9% of the numeric columns contained all identical values. This may indicate the fact that the data described is actually split over several CSV files, which would then contain different values for this column; however, no attempt was made in this work to study this idea.
- About 1% of the data columns contained values all 0 or 1. These may be interpreted as logical values in many cases, which would add to the very small number of occurrence for logical data types in table XV.

Further restricting the columns to those that did not contain consecutive values, were not all identical, and not all zero or one, results in the values shown in table XVII.

- The median number of cases was 39. This seems to support the assumption that CSV files provided in open data portals are not generally very large or detailed, but tend to provide a more aggregate view with fewer records.
- To study the distribution of the data two measures were calculated: skewness and kurtosis. The median of these measures is listed to provide an aggregate indication that is more robust against outliers than the mean.
- The median skewness of the data was about 0.8 which indicates that the tails on the right side of the distribution are longer or fatter than on the left side. This is generally taken as a sign of an asymmetric distribution.
- The median kurtosis of about 3.4 means slightly heavier tails than the normal distribution which has a kurtosis of 3; values for kurtosis > 3 indicate more outliers than in the normal distribution.

V. Discussion

This section discusses the implications for a CSV data publisher and consumer and briefly reports on related efforts (mainly centered around the analysis of Web HTML tables).

A. Implications for data consumers & data publishers

From the point of view of the data consumer i.e. someone aiming to use the huge amount of data provided in Open Data portals for interactive or automated analysis the following wish list is addressed to data providers.

Technical problems arising from different formats can usually be solved by application-specific programming; however, the cost is often prohibitive. Adherence to a few rules would improve the accessibility of open data and allow for easier knowledge discovery and analysis, which after all is one of the main purposes of open data.

- Number of columns: CSV files with much more than a few dozen columns become very difficult to handle in an exploratory mode of analysis.
- Number of header lines: interactively any number of header lines can be dealt with easily, but given the huge amount of data available in open data portals a comprehensive interactive approach is not generally feasible. Statistics packages are more easily scripted for CSV files containing a single header line.
- Choice of names in headers: ideally the headers would not only provide meaningful descriptions of the content of the data columns but would also be suited for use as variable names in the subsequent statistical analysis, since many statistics packages (including R) provide simple means to attach the header names and use them directly as variables.
- Multiple tables in one file: this adds an additional level of complexity without appreciable benefit; putting each table into a separate CSV file facilitates access to the data.

B. *Open Data CSV files listings*

In order to facilitate subsequent studies, a directory of the CSV files used in this work will be provided at the following URL: http://data.wu.ac.at/opencsv. These listings provide direct links to the files contained in the various open data portals analyzed in this work. The listings also include some of the characteristics described here, such as the number of header lines, rows, and columns, as detected by the algorithms described here.

C. *Related Studies*

Most existing work on analysis and statistics over corpora of tabular data use HTML/Web tables which are extracted by crawling the Web or a specific domain (e.g., tables found on Wikipedia) [3], [5], [6], [1], [9].

The largest attempt regarding the analysis of tabular data is to the best of our knowledge the Web Data Commons (WDC) project.[6] It presents statistics of 233 million Web tables with a total size of 165 Gigabyte. In order to generate these statistics, the project tried to detect the orientation of these tables, the header rows and entity columns, and extracts some context data of the tables. Ritze et al. [5], [6] use the WDC Web tables corpus in order to run additional analysis and to explore the potential of linking this corpus to the DBpedia knowledge base.

Crestan and Pantel [1] performed a large scale analysis of Web tables on a crawl of the Web and proposed a table type taxonomy, e.g., HTML formatting table, horizontal and vertical listings of entities, or enumerations. This automated taxonomy uses HTML tags and “Lexical features” for the categorization and therefore cannot be applied to CSV syntax.

Wang et al. [9] present approaches to “understand” Web tables in terms of schema and entities. To this end, the paper describes an header detection (based on HTML tags and formatting) and entity column detection algorithm. Again, these algorithms are tailored to Web tables and only partially applicable to CSVs.

Related to our analysis, in [2] the authors formalize a canonical form of tabular data consisting of a single header row and corresponding data rows and define three deviation levels from this canonical form: table level (e.g., metadata/comments are embedded), header level (e.g., header missing), data level (e.g., empty cells or rows). Similar to our data, the underlying corpus for the analysis of [2] is tabular Open Data (100 randomly selected CSVs).

VI. Conclusion

In this work, a corpus of 200k tabular Open Data resources from 232 portals have been analyzed from a consumers point of view. Our analysis highlights some very specific characteristics for tabular Open Data and the challenges for data consumers. The core findings of the study can be summarized as follows:

- 200k (10 %) of the resources in 232 Open Data portals are labelled as CSV, of which only 100k files can be actually parsed.
- Only 50 % of the actual CSV files specify the correct format in the HTTP response header.
- Only 10 % of the header values in single header tables could be mapped to entries in the English WordNet dictionary.
- The majority of the CSV files use the correct comma (,) delimiter.
- An average CSV Open Data file contains 365 rows and 14 columns. However, 10 % of the tables have only one row, possibly indicating that these are dummy/test tables.
- 50 % of the columns in our tables contain either numerical values or IDs (same length values, low selectivity).

Overall, our analysis revealed a number of insights into the shapes and content of Open Data portals. In future work, we will further investigate the structure and shape of CSV tables, investigating header or column path (e.g. multi-headers and columns with CAT or ID types). Furthermore, the type detection algorithm will be extended to allow for finer granular type detection. Eventually, the header and character columns will be further analyzed (language detection and applying concept/class and entity mappings, e.g. using the BabelNet service).

Acknowledgements: This work has been supported by the Austrian Research Promotion Agency (FFG) under the project ADEQUATe (grant no. 849982).

References

[1] Eric Crestan and Patrick Pantel. Web-scale table census and classification. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*, pages 545–554, 2011.

[2] Ivan Ermilov, Sören Auer, and Claus Stadler. User-driven semantic mapping of tabular data. In *I-SEMANTICS 2013 - 9th International Conference on Semantic Systems, ISEM ’13, Graz, Austria, September 4-6, 2013*, pages 105–112, 2013.

[3] Oktie Hassanzadeh, Michael J. Ward, Mariano Rodriguez-Muro, and Kavitha Srinivas. Understanding a large corpus of web tables through matching with knowledge bases - an empirical study. In *Proceedings of the Tenth International Workshop on Ontology Matching (OM-2012)*, October 2015.

[4] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

[5] Dominique Ritze, Oliver Lehmberg, and Christian Bizer. Matching html tables to dbpedia. In *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics*, WIMS ’15, pages 10:1–10:6, New York, NY, USA, 2015. ACM.

[6] Dominique Ritze, Oliver Lehmberg, Yaser Oulabi, and Christian Bizer. Profiling the potential of web tables for augmenting cross-domain knowledge bases. In *Proceedings of the 25th International Conference on World Wide Web*, WWW ’16, pages 251–261, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee.

[7] Y. Shafranovich. Common Format and MIME Type for Comma-Separated Values (CSV) Files. RFC 4180 (Informational), October 2005.

[8] Jürgen Umbrich, Sebastian Neumaier, and Axel Polleres. Quality assessment & evolution of open data portals. In *IEEE International Conference on Open and Big Data*, Rome, Italy, August 2015.

[9] Jingjing Wang, Haixun Wang, Zhongyuan Wang, and Kenny Qili Zhu. Understanding tables on the web. In *Conceptual Modeling - 31st International Conference ER 2012, Florence, Italy, October 15-18, 2012. Proceedings*, pages 141–155, 2012.

[6] http://webdatacommons.org/webtables/2015/relationalStatistics.html