Module 2

Data preparation and cleaning

We will start by loading all the librarieswe will need.

In [1]:
```
import sys
import time
import bibtexparser
import itertools
import requests
import re
import matplotlib as plt
import pandas as pd
from bs4 import BeautifulSoup
import numpy as np
```

In [2]:
```
bibtex_file = open('gap-publishednicer.bib.txt', encoding='utf-8')
bib_data = bibtexparser.load(bibtex_file)
bib = bib_data.entries # we prepare the GAP Bibliography file, ready to be loadec
```

Here are the 3 datasets we will start with.

In [3]:
```
bib_df = pd.DataFrame.from_dict(bib) # Large one from the Bibliography
review_df = pd.read_csv('no_citation_text.csv', dtype='str') # MR numbers who car
corpus_df = pd.read_csv('gap_citations_corpus.csv', dtype='str') # CItations scrc
```

Larger dataset from Bibliography

We will start by filtering the data, let us look at all the columns at our disposal.

In [4]:
```
bib_df.columns
```

Out[4]:
```
Index(['printedkey', 'doi', 'url', 'mrreviewer', 'mrnumber', 'mrclass', 'issn',
       'fjournal', 'pages', 'year', 'volume', 'journal', 'title', 'author',
       'ENTRYTYPE', 'ID', 'number', 'school', 'booktitle', 'isbn', 'note',
       'publisher', 'day', 'keywords', 'month', 'series', 'annote', 'type',
       'address', 'institution', 'howpublished', 'editor', 'bookeditor',
       'edition', 'key', 'organization'],
      dtype='object')
```

We only need some of these columns, hence we `drop` the rest.

In [5]:
```
bib_df.drop(bib_df.columns[[0, 1, 2, 3, 6, 7, 8, 10, 12, 15, 16, 17, 18, 19, 20,
```

In [6]:
```
bib_df.columns
```

Out[6]:
```
Index(['mrnumber', 'mrclass', 'year', 'journal', 'author', 'ENTRYTYPE'], dtype
='object')
```

We reorder the columns. Then we format the names accordingly. We change the `mrnumber` coulmn name to `MR` so we cane later merge this dataframe with the other one.

In [7]:
```
bib_df = bib_df[['mrnumber', 'author', 'journal', 'year', 'ENTRYTYPE', 'mrclass']
bib_df.columns = ['MR', 'Author', 'Journal', 'Year', 'Publication Type', 'MSC']
bib_df
```

Out[7]:

	MR	Author	Journal	Year	Publication Type	MSC
0	4056124	Abas, M. and Vetrík, T.	Theoret. Comput. Sci.	2020	article	05C25 (05C20 20F05)
1	3942387	Abbas, A. and Assi, A. and García-Sánchez, P. A.	Rev. R. Acad. Cienc. Exactas Fís. Nat. Ser. A ...	2019	article	13F20 (05E15 14H50)
2	NaN	Abdeljaouad, I.	RAIRO-INF THEOR APPL	1999	article	NaN
3	3354065	Abdolghafourian, A. and Iranmanesh, M. A.	Comm. Algebra	2015	article	05C25 (20B30 20E45)
4	3646312	Abdolghafourian, A. and Iranmanesh, M. A. and ...	J. Pure Appl. Algebra	2017	article	20G40 (05C25)
...	...	...	...	...	...	...
3362	2647300	Zusmanovich, P.	J. Geom. Phys.	2010	article	17B60
3363	2735394	Zusmanovich, P.	J. Algebra	2010	article	17B40
3364	3201064	Zusmanovich, P.	J. Algebra	2014	article	17B40
3365	3598575	Zusmanovich, P.	Linear Algebra Appl.	2017	article	17C10 (17-08 17A30 17C55)
3366	3089327	Zvezdina, M. A.	Sibirsk. Mat. Zh.	2013	article	20D05 (05C25)

3367 rows × 6 columns

We can inspect Data types and count of non-null values for each column.

```
In [8]: bib_df.info(show_counts  = True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3367 entries, 0 to 3366
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   MR                3159 non-null   object
 1   Author            3367 non-null   object
 2   Journal           3047 non-null   object
 3   Year              3367 non-null   object
 4   Publication Type  3367 non-null   object
 5   MSC               3252 non-null   object
dtypes: object(6)
memory usage: 79.0+ KB
```

Looking at a single entry from the `MRN` column, some cells contain `NaN`

```
In [9]: bib_df.iloc[3274]
```

```
Out[9]: MR                              NaN
        Author                   Wegner, A.
        Journal                         NaN
        Year                           1989
        Publication Type     mastersthesis
        MSC                          Thesis
        Name: 3274, dtype: object
```

- this method is used usually for numerical columns but we can try it to get an overview of our data

```
In [10]: bib_df.describe()
```

Out[10]:

	MR	Author	Journal	Year	Publication Type	MSC
count	3159	3367	3047	3367	3367	3252
unique	3158	2511	384	43	10	2268
top	3656296	Sambale, B.	J. Algebra	2017	article	Thesis
freq	2	18	387	188	2976	99

- this gives us an overview of a column, displaying top 5 most frequent values and the 5 least frequent, with their counts

```
In [11]: bib_df['MSC'].value_counts()
```

```
Out[11]: Thesis                 99
         20C15                  36
         20C20                  33
         20N05                  30
         20D15                  22
                                ..
         42C15 (05C50 05C90)     1
         16E20 (16G20 16S99)     1
         05C25 (05E30 20B25)     1
         14L35 (20G40 20G41)     1
         20E18 (20D15 20F40)     1
         Name: MSC, Length: 2268, dtype: int64
```

We will process the `year` column. There are several anomalies and we need just 4 digits in each cell.

```
In [12]: bib_df.sort_values('Year', ascending=False)
```

Out[12]:

	MR	Author	Journal	Year	Publication Type	MSC
3165	3973299	Then, H.	NaN	[2019] \copyright 2019	incollection	11F12 (11R06)
1546	3898507	Greer, M.	NaN	[2019] \copyright 2019	incollection	20N05
3133	3898514	Stuhl, I. and Vojtěchovský, P.	NaN	[2019] \copyright 2019	incollection	20N05 (57M27)
3147	3782458	Swinarski, D.	NaN	[2018] \copyright 2018	incollection	30F20 (14H37 14H45 14Q05)
1981	4167659	Kaushik, R. and Yadav, M. K.	J. Algebra	2021	article	20D15 (20F12)
...	...	...	...	...	...	...
3274	NaN	Wegner, A.	NaN	1989	mastersthesis	Thesis
2689	NaN	Niemeyer, A.	NaN	1988	mastersthesis	Thesis
2673	NaN	Nickel, W.	NaN	1988	mastersthesis	Thesis
3018	NaN	Schönert, M.	NaN	1987	mastersthesis	Thesis
2459	NaN	Meier, J.	NaN	1987	mastersthesis	Thesis

3367 rows × 6 columns

We will use `.str` and a regular expresion `(r'^(\d{4})'` which first converts all year cells to strings then takes the first 4 digits from each. We then replace the old values with the filtered ones.

```
In [13]: bib_df['Year'] = bib_df['Year'].str.extract(r'^(\d{4})', expand=False)
```

In [14]:
```
bib_df['Year'].value_counts() # to inspect the results
```

Out[14]:
```
2017    188
2013    175
2018    168
2020    166
2019    165
2010    163
2015    162
2016    158
2014    154
2011    152
2012    142
2007    142
2008    132
2004    131
2005    128
2009    124
2006    118
2001    107
2003    101
2002     84
1999     84
2000     78
1997     76
1998     58
1995     56
2021     39
1996     34
1994     28
1993     25
1992     13
1991      5
1987      2
1989      2
1988      2
1990      1
Name: Year, dtype: int64
```

We inspect the result and no more anomalies are visible. Data type is integer which is exactly what we need for futher operations.

Now we will focus on the other two input files, produced from Module 1 - the Web-scraping tool.

We already loaded them at the beginning of the notbook. We will only work with the main data - `corpus_df` .
The other file `no_citation_text.csv` containing few anomalies we will not handle in this project, in real-life scenario each entry there will be manually investigated by staff who work in the GAP Group, or whichever is the institution or company we are working with.

In [15]:
```
corpus_df
```

Out[15]:

	MR	Citation
0	MR4056124	GAP – Groups, algorithms, programming - a syst...
1	MR3942387	Delgado, M., García-Sánchez, P.A., Morais, J.:...
2	MR3942387	The GAP Group: GAP—groups, algorithms, and pro...
3	MR3354065	The GAP – Groups, Algorithms and Programming. ...
4	MR3646312	The \ssf$GAP$ Group, \ssf$GAP$–Groups, Alg...
...	...	...
3537	MR3988630	M. Delgado, P. A. García-Sánchez and J. Morais...
3538	MR1801202	L.H. Soicher, GRAPE: a system for computing wi...
3539	MR2558870	L. Bartholdi, Functionally recursive groups, h...
3540	MR2824780	X. Sun, C. Liu, D. Li and J. Gao, On duality g...
3541	MR1981371	Schönert M. et al., Groups, Algorithms and Pro...

3542 rows × 2 columns

In [16]:
```
pd.options.display.max_colwidth = 157 # increasing column width for better readab
```

We start by defining two functions, to help us browse the data by MR number. The base for the functions was borrowed from the second year Python course CS2006 by Dr Konovalov, but they were modifed to better fir this project. The first function displays just Citation text and Version. The second function displays the whole row for given MR number.

In [17]:
```
# Python lectures by Dr Konovalov
# https://studres.cs.st-andrews.ac.uk/CS2006/Lectures/Python/L08-dataset.pdf

def get_citation(mrno):
    r = corpus_df[corpus_df['MR'] == mrno]
    return r.at[r.index[0],'Citation'], r.at[r.index[0],'Version']
```

In [18]:
```
# Python lectures by Dr Konovalov
# https://studres.cs.st-andrews.ac.uk/CS2006/Lectures/Python/L08-dataset.pdf
# slightly modified so it can return all citations with the specified MRN
# on the other hand the result is a dataframe and if we want to read the full cit
def get_c(mrno):
    r = corpus_df[corpus_df['MR'] == mrno]
    return r
```

Version

Version is a very important feature and we need to have it in a separate column. We will achieve this by parsing each citation cell with a Regex and extracting the version, where provided.

- First we create the version column.

In [19]:
```
corpus_df.insert(loc=2, column='Version', value=' ')
```

This is the function that we will use to parse each citation and extract the version.
It is based on the lectures from CS2006 by Dr Konovlov.
It also prints the outputs, which was used while testing and modifying the function until it worked fine for our purposes.

In [20]:
```
# Python lectures by Dr Konovalov
# https://studres.cs.st-andrews.ac.uk/CS2006/Lectures/Python/L08-dataset.pdf

unknown_ver = []

def get_version(s):
    match = re.search("(?:(\d+\.(?:\d+\.)*\d+))", s, re.IGNORECASE)
    if match != None:
        return match.group(1)
        print('* VERSION FOUND *')
    else:
        print('* No VERSION found *', s)
        unknown_ver.append(s)
        return 'Unknown'
```

We appl it to the `Citation` column.

In [21]:
```
corpus_df['Version'] = corpus_df['Citation'].map(get_version)
```

```
* No VERSION found * GAP – Groups, algorithms, programming - a system for com
putational discrete algebra, www.gap-system.org.
* No VERSION found * Delgado, M., García-Sánchez, P.A., Morais, J.: "Numerica
l Sgps", A GAP package for numerical semi-groups. https://gap-packages.githu
b.io/numericalsgps. (https://gap-packages.github.io/numericalsgps.) Accessed
 19 Aug 2017
MR3493240
* No VERSION found * M. Schönert et al. GAP - Groups, Algorithms, and Program
ming (Lehrsthul D für Mathematik, Reinisch-Westflische Technische Hochschule,
Aachen, Germany, fifth ed., 1995.)
* No VERSION found * W. Nickel, NQ, 1998, A refereed GAP 4 package, see [10].
* No VERSION found * W. Nickel, NQ, 1998, A refereed GAP 4 package, see [8].
* No VERSION found * Gamble, G., Nickel, W., O'Brien, E.A.: ANU p-Quotient-p-
Quotient and p-Group Generation Algorithms (2006). An accepted GAP 4 package,
available also in MAGMA
* No VERSION found * M. Schönert et al, GAP: groups, algorithm and programmin
g, © 1992 by Lehrstuhl D für Mathematik, distributed with the GAP software vi
a ftp from samson.math.rwth-aachen.de.
* No VERSION found * M. Delgado, P. A. García-Sánchez and J. Morais, "numeric
algsps": a GAP [12] package on numerical semigroups. (http://www.gap-system.o
```

In [22]:

```
corpus_df['Version'].value_counts() # to inspect results
```

Out[22]:

```
Unknown     895
4.4         460
4.4.12      310
4.3         232
4.4.10      136
           ... 
10.2140       1
0.6.5         1
2.22          1
4.4.2006      1
0.9.4         1
Name: Version, Length: 197, dtype: int64
```

Then we will further process the `Version` column by finding and labelling GAP Packages. Packages are connected to GAP, but technically is a separate piece of software, having its own Version tree. Therefore, in entries citing GAP package there is no version of GAP and we will fill the `Version` cell with the string `Package` . We will create and apply a function which chekcs if it is a case of citing GAP Package. It will search citations for the word "package" in order to determine if they are citing GAP or a GAP Package, in the latter case the `Version` cell value will be replaced with 'Package'.

- First we create a list of all GAP Package names, adding the ones already out of use, just in case.

In [23]:

```
f = open('packages.txt', 'r')
pac_name = []
for line in f:
        mat = line.split(" ",1)[0]
        pac_name.append(mat)
pac_name.append('magma')
pac_name.append('anu')
pac_name.append('Carat')
pac_name.append('Citrus')
pac_name.append('Convex')
pac_name.append('Gpd')
pac_name.append('MONOID')
pac_name.append('NQL')
pac_name.append('ParGAP')
pac_name.append('PolymakeInterface')
pac_name.append('QaoS')
pac_name.append('recogbase')
pac_name.append('RAMEGA')
#-fr modules
```

- We use a regex expression combined with the list we compiled so the function searches citations either for the word "Package" ignoring case or fo any of the Package names.
  We also add a case if the citation contains "manual" - in such cases it is not package, but counts as a GAP citation and we leave the Version unchanged. This function also prints the output, which was used in the tuning, debugging and polishing the function to perfection.

In [24]:

```
def is_package(series):
    mrno = series['MR']
    citation = series['Citation']
    version = series['Version']
    manu = re.search("manual", citation, re.IGNORECASE)
    m = re.search(r"(?=(\b" + '\\b|\\b'.join(pac_name) + r"\b))", citation, re.I
    if re.search("package", citation, re.IGNORECASE) != None:
        print('***Package***:', mrno, citation)
        return 'Package'
    elif manu != None:
        print('& Manual &', citation, version)
        return series['Version']
    elif m != None:
        print('* Package *:', mrno, citation, version)
        return 'Package'
    else:
        print('***Not a Package***:', mrno, citation, version)
        return series['Version']
```

In [25]:

```
corpus_df['Version'] = corpus_df.apply(is_package,axis=1)
```

```
***Not a Package***: MR4056124 GAP – Groups, algorithms, programming - a syst
em for computational discrete algebra, www.gap-system.org. Unknown
***Package***: MR3942387 Delgado, M., García-Sánchez, P.A., Morais, J.: "Nume
rical Sgps", A GAP package for numerical semi-groups. https://gap-packages.gi
thub.io/numericalsgps. (https://gap-packages.github.io/numericalsgps.) Access
ed 19 Aug 2017
MR3493240
***Not a Package***: MR3942387 The GAP Group: GAP—groups, algorithms, and pro
gramming, version 4.7.5 (2014). http://www.gap-system.org. (http://www.gap-sy
stem.org.) Accessed 19 Aug 2017 4.7.5
***Not a Package***: MR3354065 The GAP – Groups, Algorithms and Programming.
 Version 4.4.12, 2008. www.gap-system.org. 4.4.12
***Not a Package***: MR3646312 The $\ssf{GAP}$ Group, $\ssf{GAP}$–Groups, Alg
orithms, and Programming, 4.7.8, 2015, http://www.gap-system.org. (http://ww
w.gap-system.org.) 4.7.8
***Not a Package***: MR1864795 M. Schönert et al. GAP - Groups, Algorithms, a
nd Programming (Lehrstuhl D für Mathematik, Reinisch-Westflische Technische H
ochschule, Aachen, Germany, fifth ed., 1995.) Unknown
***Not a Package***: MR2287843 The GAP Group, GAP - Groups, Algorithms, and P
```

In [26]:

```
corpus_df['Version'].value_counts() # for overview on the results
```

Out[26]:

```
Package       819
Unknown       493
4.4           454
4.4.12        310
4.3           212
             ...
4.08.10         1
4.46            1
1405.5063       1
4.6.12          1
1804.09707      1
Name: Version, Length: 84, dtype: int64
```

Version Filter

We need to filter out some anomalies in the version column, such as too long versions which are usually `arXiv` numbers, dates connected with version or other organizations' serial numbers.
THe following function isolates any entries with Version value longer than 6 characters, then replaces it with the string 'Not GAP citation'.
It also prints the output and we can see there are not many such entries, so we will inspect them manually.

In [27]:

```
def version_filter(series):
    mrno = series['MR']
    citation = series['Citation']
    version = series['Version']
    ind = series.name
    if version != 'Package' and version != 'Unknown' and len(version) > 6:
        print(ind, 'Too long Version *', mrno, citation)
        return 'Not GAP citation'
    else:
        return series['Version']
```

In [28]:

```
corpus_df['Version'] = corpus_df.apply(version_filter, axis=1)
```

```
125 Too long Version * MR4170882 F. Ali, M. Al-Kadhi, A. Aljouiee, M.A.F. Ibr
ahim, 2-Generations of finite simple groups in GAP, in: 2016 International Co
nference on Computational Science and Computational Intelligence (CSCI), IEEE
Conf. Proc., 249, IEEE, Las Vegas, NV, 2016, pp. 1339–1344 (doi:10.1109/CSCI.
2016.0250).
366 Too long Version * MR2422501 The GAP Group. (2005). GAP - Groups, Algorit
hms, and Programming, version 4.4.10.2007. http://www.gap-system.org. (htt
p://www.gap-system.org.)
371 Too long Version * MR3272384 John Bamberg, S.P. Glasby, Eric Swartz, AS-c
onfigurations and skew-translation generalised quadrangles (including support
ing GAP code), arXiv:1405.5063v2.
645 Too long Version * MR4193641 GAP – Groups, Algorithms, and Programming.
 (2018). Version 4.08.10. https://www.gap-system.org. (https://www.gap-syste
m.org.)
651 Too long Version * MR2422303 T. Breuer, GAP computations concerning proba
bilistic generation of finite simple groups, arXiv:0710.3267.
655 Too long Version * MR2669683 T. Breuer, `GAP computations concerning Hami
ltonian cycles in the generating graphs of finite groups', Preprint, 2009, ar
Xiv:0911.5589.
```

We have a list of anomalies here which we inspect manually in the cell above. We will only look at the genuine GAP citations with typing errors conencting version and year - these we will fix manually with our function `fix_version` .
Others are not GAP citations but rather citing articles connected to GAP and have other organizational numbers such as `arXiv:0710.3267` which fooled our version hunter function - these we will remove from our data once we finish the manual fixing as they are not citations of GAP software or its packages.

In [29]:
```
# https://studres.cs.st-andrews.ac.uk/CS2006/Lectures/Python/L08-dataset.pdf
def fix_version(mrno,version):
 r = corpus_df[corpus_df['MR'] == mrno]
 corpus_df.at[r.index[0],'Version']=version
```

We start with MR2422501 which is version 4.4 accidentaly connected with the year, we will manually fix it below.

In [30]:
```
get_c('MR2422501')
```

Out[30]:

	MR	Citation	Version
366	MR2422501	The GAP Group. (2005). GAP - Groups, Algorithms, and Programming, version 4.4.10.2007. http://www.gap-system.org.	Not GAP citation

In [31]:
```
fix_version('MR2422501', '4.4')
```

Next is MR4193641 which should be 4.8.10 instead of 4.08.10. Fixed manually below.

In [32]:
```
get_c('MR4193641')
```

Out[32]:

	MR	Citation	Version
645	MR4193641	GAP – Groups, Algorithms, and Programming. (2018). Version 4.08.10. https://www.gap-system.org.	Not GAP citation

In [33]:
```
fix_version('MR4193641', '4.8.10')
```

Next we have version 4.4 accidentally connected with the year again, easy fix below.

In [34]:
```
get_c('MR2526731')
```

Out[34]:

	MR	Citation	Version
1839	MR2526731	The GAP Group, GAP–Groups, Algorithms, and Programming, Version 4.4.2006. http://www.gap-system.org.	Not GAP citation

In [35]:
```
fix_version('MR2526731', '4.4')
```

This citation has a long number before the version which was captured by our version checker and used as version. The real version is 4.4.12 which we will manually assign below.

In [36]:
```
get_c('MR2928559')
```

Out[36]:

	MR	Citation	Version
2315	MR2928559	L. R. Ford, Automorphic functions, Chelsea, 1951. Zbl 55.0810.04 GAP - groups, algorithms, and programming, Version 4.4.12, The GAP Group, St. Andrews, F...	Not GAP citation

In [37]:
```
fix_version('MR2928559', '4.4.12')
```

All the rest are anomalies citing other sources but not GAP.

Once we manually fixed all the genuine citations versions, we will delete all the remaining records with version labelled 'Not GAP citation' with the following line of code.

In [38]:
```
corpus_df = corpus_df[corpus_df['Version'] != 'Not GAP citation']
```

In [39]:
```
corpus_df.loc[corpus_df['MR'] == 'MR3957957']
```

Out[39]:

	MR	Citation	Version
1150	MR3957957	The GAP Group, GAP – Groups, Algorithms, and Programming, http://www.gap-system.org.	Unknown
1151	MR3957957	D.F. Holt, The \ssf$GAP$ package \ssf$kbmag$, Knuth-Bendix on monoids and automatic groups, https://www.gap-system.org/Packages/kbmag.html.	Package
1152	MR3957957	M. Neunhöffer, Á. Seress, et al., The \ssf$GAP$ package \ssf$recog$, A collection of group recognition methods, http://gap-packages.github.io/recog/.	Package

- Now we will investigate the versions a little bit more manually.

Versions from 4 onwards are fine, we will focus on the older ones between 1 and 3, as they might be anomalies which are not GAP citations at all.

In [40]:
```
ver_list = corpus_df['Version'].unique()
ver_list = np.sort(ver_list)
ver_list # list of versions we have in the data
```

Out[40]:
```
array(['1.0', '1.1', '1.9.6', '3.0', '3.1', '3.2', '3.3', '3.4', '3.4.3',
       '3.4.4', '4.1', '4.10', '4.10.0', '4.10.1', '4.10.2', '4.11',
       '4.11.0', '4.2', '4.3', '4.4', '4.4.10', '4.4.11', '4.4.12',
       '4.4.2', '4.4.3', '4.4.4', '4.4.5', '4.4.6', '4.4.7', '4.4.9',
       '4.46', '4.49', '4.5', '4.5.3', '4.5.4', '4.5.5', '4.5.6', '4.5.7',
       '4.6', '4.6.1', '4.6.12', '4.6.2', '4.6.3', '4.6.4', '4.6.5',
       '4.6.9', '4.7', '4.7.2', '4.7.4', '4.7.5', '4.7.6', '4.7.7',
       '4.7.8', '4.7.9', '4.8', '4.8.1', '4.8.10', '4.8.2', '4.8.3',
       '4.8.4', '4.8.5', '4.8.6', '4.8.7', '4.8.8', '4.8.9', '4.9',
       '4.9.0', '4.9.1', '4.9.2', '4.9.3', '5.7', 'Package', 'Unknown'],
      dtype=object)
```

Versions 1.0, 1.0.0 and 1.1 have less than 10 records and we will check them all manually. We will start with 1.0, as we can see below it is a GAP manual which is early practice of GAP citation and we will keep it in the data so we can investigate how this early practice dissapeared over time.

In [41]:
```
corpus_df[corpus_df['Version'] == '1.0']
```

Out[41]:

	MR	Citation	Version
2222	MR2111596	Breuer, T. (2001). Manual for the GAP Character Table Library, Version 1.0. Lehrstuhl D für Mathematik; RWTH Aachen, Germany.	1.0

In [42]:
```
fix_version('MR2111596','Package')
```

All the six records with version 1.1 are actually for the "Character Table Library" which is a GAP package, but escaped the Regex expression because its full name was used here. I will fix these manually.

In [43]:
```
corpus_df[corpus_df['Version'] == '1.1']
```

Out[43]:

	MR	Citation	Version
389	MR2308856	Thomas Breuer, Manual for the GAP character table library, Version 1.1 (Lehrstuhl D für Mathematik, Rheinisch Westfälische Technische Hochschule, Aachen,...	1.1
735	MR3007647	T. Breuer, Manual for the GAP character table library, version 1.1 (RWTH, Aachen, 2004).	1.1
738	MR2684423	T. Breuer, Manual for the GAP Character Table Library, Version 1.1, RWTH Aachen, 2004.	1.1
741	MR3219555	T. Breuer, Manual for the GAP Character Table Library, Version 1.1, RWTH Aachen, 2004.	1.1
966	MR2805443	Breuer, T.: Manual for the GAP Character Table Library, Version 1.1, Lehrstuhl D für Mathematik, Rheinisch Westfälische Technische Hochschule, Aachen, Ge...	1.1
1742	MR2326329	T. Breuer, Manual for the GAP Character Table Library Version 1.1 (Lehrstuhl D für Mathematik, Rheinisch West-fälische Hochschule, Aachen, 2004).	1.1

In [44]:
```
fix_version('MR2308856', 'Package')
fix_version('MR3007647', 'Package')
fix_version('MR2684423', 'Package')
fix_version('MR3219555', 'Package')
fix_version('MR2805443', 'Package')
#fix_version('MR2326329', 'Package')
```

In [45]:
```
corpus_df[corpus_df['MR'] == 'MR2326329']
```

Out[45]:

	MR	Citation	Version
1741	MR2326329	The GAP Group, gap—Groups, Algorithms, Programming, Version 4.4.7, 2006 (http://www.gap-system.org).	4.4.7
1742	MR2326329	T. Breuer, Manual for the GAP Character Table Library Version 1.1 (Lehrstuhl D für Mathematik, Rheinisch West-fälische Hochschule, Aachen, 2004).	1.1

The last entry `MR2326329` has two citations and our `fix_version` function wrongly apllies itself on the firt one. Therefore, we will use the manual fix below instead.

In [46]:
```
corpus_df.loc[1742]['Version']='Package'
```

There is a single entry with version 1.9.6. After discussing with Dr Konovalov, we were both unable to access the paper and it is definitely some sort of error as there is no such early GAP release, we have decided to exclude this record from the analysis.

In [47]:
```
corpus_df[corpus_df['Version'] == '1.9.6']
```

Out[47]:

	MR	Citation	Version
2824	MR2747149	The GAP Group, Welcome to GAP – Groups, Algorithms and Programming: a system for computational discrete algebra. Version 1.9.6, URL www.gap-system.org/. ...	1.9.6

In [48]:
```
corpus_df[corpus_df['MR'] == 'MR2747149']
```

Out[48]:

	MR	Citation	Version
2824	MR2747149	The GAP Group, Welcome to GAP – Groups, Algorithms and Programming: a system for computational discrete algebra. Version 1.9.6, URL www.gap-system.org/. ...	1.9.6

In [49]:
```
#corpus_df.drop(2824, inplace=True)
corpus_df.drop(corpus_df[corpus_df['MR'] == 'MR2747149'].index, inplace=True)
```

```
c:\users\fliqp_000\appdata\local\programs\python\python38-32\lib\site-packages
\pandas\core\frame.py:4305: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  return super().drop(
```

We have one entry with version 3.0 which is another example of early GAP citation practice by Martin Schönert. He is one of the initial authors, who created the GAP language.

```
In [50]: corpus_df[corpus_df['Version'] == '3.0']
```

Out[50]:

	MR	Citation	Version
1601	MR1195429	M. Schönert (Editor), GAP 3.0 manual, Lehrstuhl D für Mathematik, RWTH Aachen, 1991.	3.0

Two entries with version 3.1, again by Martin Schönert, they will remain in the data to help us analyse early GAP citation practice.

```
In [51]: corpus_df[corpus_df['Version'] == '3.1']
```

Out[51]:

	MR	Citation	Version
2985	MR1176715	M. Schönert et al., GAP 3.1 manual, March 1992, Lehrstuhl D für Mathematik, RWTH Aachen.	3.1
3235	MR1213836	M. Schönert (ed.), GAP: groups, algorithms and programming. Manual (version 3.1), Lehrstuhl D für Mathematik, RWTH Aachen, 1992.	3.1

One entry for version 3.2, again it remains in the data as it is genuine early GAP citation.

```
In [52]: corpus_df[corpus_df['Version'] == '3.2']
```

Out[52]:

	MR	Citation	Version
3517	MR1425323	M. Schönert et al (eds), \sc Gap: groups, algorithms, and programming, Manual, release 3.2, Lehrstuhl D für Mathematik, RWTH Aachen, 1993.	3.2

With version 3.3 two more early GAP citation practice examples.

```
In [53]: corpus_df[corpus_df['Version'] == '3.3']
```

Out[53]:

	MR	Citation	Version
872	MR1468940	M. Schönert et al., GAP Groups, Algorithms and Programming 3.3, Lehrstuhl D für Mathematik, RWTH Aachen, 1993.	3.3
3519	MR1624797	M. SCHO NERT (ed.), Gap-3.3 manual (RWTH Aachen, 1993).	3.3

We have quite a few examples with version 3.4 again by Marting Schönert and we will keep them in the data, as there are no anomalies here.

In [54]:
```
corpus_df[corpus_df['Version'] == '3.4']
```

Out[54]:

	MR	Citation	Version
409	MR1626409	M. Schönert et al., GAP version 3.4, 4th edition, Lehrstuhl D für Mathematik, RWTH Aachen, 1995.	3.4
632	MR1743630	M. Schönert, GAP: Groups Algorithms and Programming, version 3.4, Lehrstuhl D für Mathematik, RWTH Aachen, 1994.	3.4
633	MR1842416	Schönert, M. et al. GAP 3.4 Manual (Groups, Algorithms, and Programming); RWTH Aachen, 1994.	3.4
835	MR1443190	Martin Schönert et al., GAP - Groups, Algorithms, and Programming, Release 3.4, Lehrstuhl D für Mathematik, Rheinisch-Westfälische Technische Hochschule,...	3.4
837	MR1482983	M. Schönert et al., "GAP—Groups, Algorithms, and Programming," Release 3.4, Lehrstuhl D für Mathematik, Rheinisch-Westfälische Technische Hochschule, Aac...	3.4
855	MR1831996	M. Schönert et al., GAP - Groups, Algorithms and Programming, Release 3.4, Lehrstuhl D für Mathematik, Rheinisch-Westfälische Technische Hochschule, Aach...	3.4
1488	MR1968456	M. Schönert, et al., GAP 3.4, patchlevel 4, School of Mathematical and Computational Sciences, University of St. Andrews, Scotland, 1997.	3.4
1613	MR1800032	Schönert, M. et al. (1997). GAP 3.4, patchlevel 4. School of Mathematical and Computational Sciences, University of St Andrews, Scotland.	3.4
1877	MR1772517	M. Schönert et al. GAP—Groups, Algorithms and Programming. Lehrstuhl D für Mathematik, RWTH Aachen, 3.4 edition, 1994.	3.4
1887	MR1673415	M. Schönert et al., GAP, version 3.4, 4th edn (D für Mathematik, RWTH Aachen, 1995).	3.4
2039	MR1610479	M. SCHO NERT et al., GAP 3.4 manual (groups, algorithms, and programming) (Lehrstuhl D fu r Mathematik, RWTH Aachen, 1994).	3.4
2215	MR1800033	Schönert, M. et al. (1997). GAP 3.4, patchlevel 4. School of Mathematical and Computational Sciences, University of St Andrews, Scotland.	3.4
2257	MR1915094	M. Schönert, et al., GAP 3.4 Manual, RWTH, Aachen, 1994.	3.4
2258	MR2078933	M. Schönert et al., GAP 3.4 Manual, RWTH Aachen, Aachen, 1994.	3.4
2287	MR2007740	M. Schönert et al. GAP 3.4 Manual (Groups, algorithms and programming) (RWTH Aachen, 1994).	3.4
2290	MR2014018	M. Schönert et al., GAP 3.4 Manual (Groups, Algorithms and Programming), RWTH Aachen, Aachen, Germany, 1994.	3.4
2436	MR1476055	M. Schönert et. al., GAP: Groups, Algorithms and Programming (version 3.4), Lehrstuhl D für Mathematik, RWTH Aachen, Germany, 1994.	3.4
2475	MR1806213	Schönert, M. et al. (1994). GAP—Groups, Algorithms and Programming, 3.4 edn. Lehrstuhl D für Mathematik, RWTH Aachen.	3.4
2641	MR1800751	M. Schönert et al., "GAP 3.4 Manual (Groups, Algorithms and Programming)," RWTH Aachen, Aachen, Germany, 1994.	3.4
2643	MR1825823	M. Schönert et al. GAP 3.4 Manual (Groups, Algorithms and Programming). RWTH Aachen, 1994.	3.4
2644	MR2031331	M. Schönert et al., GAP 3.4 Manual (Groups, Algorithms and Programming), RWTH Aachen, Aachen, Germany, 1994.	3.4
2645	MR1921730	M. Schönert et al., GAP 3.4 Manual (Groups, Algorithms and Programming). RWTH Aachen 1994.	3.4

	MR	Citation	Version
2646	MR1997749	M. Schönert et al., GAP 3.4 Manual (Groups, Algorithms and Programming) (RWTH Aachen, Aachen, Germany, 1994).	3.4
2873	MR1888424	M. Schönert et al., GAP 3.4 manual (Groups, Algorithms, and Programming), Lehrstuhl D für Mathematik, RWTH Aachen, 1994.	3.4
2937	MR1423329	M. Schönert, et al., GAP (Groups Algorithms and Programming) Version 3.4, RWTH Aachen.	3.4
2938	MR1807659	M. Schönert et al., "GAP: Groups Algorithms and Programming," Ver. 3.4, Lehrstuhl D für Mathematik, RWTH Aachen, 1994.	3.4
3094	MR2098769	M. Schönert, GAP: Groups Algorithms and Programming, version 3.4, Lehrstuhl D für Mathematik, RWTH Aachen, 1994.	3.4
3284	MR1765312	M. Schönert et al., GAP 3.4 Manual (Groups, Algorithms, and Programming), RWTH Aachen, 1994.	3.4
3513	MR1658168	M. Schönert (ed.), Gap-3.4, manual, RWTH Aachen, 1994.	3.4
3514	MR1769294	M. Schönert (Ed.), "Gap-3.4, Manual," RWTH Aachen, 1994.	3.4

We have 3 records for 3.4.3 and 9 records for 3.4.4 all of them genuine early GAP citations, which we will gladly keep in the data.

In [55]:

```
corpus_df[corpus_df['Version'] == '3.4.3']
```

Out[55]:

	MR	Citation	Version
2204	MR1863400	M. Schönert et al., GAP 3.4.3 manual (Groups, Algorithms and Programming) Lehrstuhl D für Mathematik, RWTH Aachen, 1996.	3.4.3
3478	MR1764578	M. Schönert et al., "Gap: Groups, Algorithms and Programming, 3.4.3," RWTH Aachen, 1996.	3.4.3
3486	MR1807270	M. Schönert et. al., Gap: groups, algorithms and programming, 3.4.3, RWTH Aachen, 1996.	3.4.3

In [56]:
```
corpus_df[corpus_df['Version'] == '3.4.4']
```

Out[56]:

	MR	Citation	Version
635	MR2049015	The GAP group. GAP: Groups, Algorithms and Programming. (Version 3.4.4, 1997; Version 4.2, 2001.)	3.4.4
848	MR1704676	The GAP Group, Lehrstuhl D für Mathematik, RWTH, Aachen, Germany, and School of Mathematical and Computational Sciences, University of St. Andrews, Scotl...	3.4.4
990	MR1837963	Schönert M. et al. GAP–Groups, algorithms and programming. Version 3.4.4. Lehrstuhl D für Mathematik, RWTH Aachen, and School of Mathematical and Computa...	3.4.4
1037	MR1946634	Schönert, M. (together with, Bessche, H. U. et al.), (1997). updated by S. A. Linton, GAP: Groups Algorithms and Programming v. 3.4.4. Distributed Electr...	3.4.4
1671	MR3550870	The GAP Group, GAP — Groups, Algorithms, and Programming, Version 3.4.4, http://www.gap-system.org, 1997.	3.4.4
2095	MR1960300	The GAP Group, Lehrstuhl D für Mathematik, RWTH Aachen, Germany and School of Mathematical and Computational Sciences, U. St. Andrews, Scotland. GAP—Grou...	3.4.4
3050	MR3184410	The GAP Group, GAP - Groups. Algorithms, and Programming, Version 3.4.4; 1997. (http://www.gap-system.org)	3.4.4
3520	MR2143203	M. Schönert et al., Gap: groups, algorithms, and programming, in: Lehrstuhl D für Mathematik, 3.4.4 ed., RWTH Aachen, 1997.	3.4.4
3528	MR1695079	M. Schönert et al., "Gap: groups, algorithms, and programming," Lehrstuhl D für Mathematik, RWTH Aachen, 3.4.4 edition, 1997.	3.4.4

There is one entry with version 5.7, after manual inspection we can see this is a typing error. We will use our function to manually fix the version of such anomalies.

In [57]:
```
corpus_df[corpus_df['Version'] == '5.7']
```

Out[57]:

	MR	Citation	Version
316	MR4052374	The GAP Group, GAP-Groups, Algorithms, and Programming, Version 4–5.7, http://www.GAP-system.org (2012).	5.7

In [58]:
```
fix_version('MR4052374','4.5.7')
```

In [59]:
```
corpus_df[corpus_df['Version'] == '5.7'] # now the anomaly is gone
```

Out[59]:

MR	Citation	Version

Versions 4.46 and 4.49 are typing errors and we will correct them to 4.4.6 and 4.4.9

In [60]:
```
corpus_df[corpus_df['Version'] == '4.46']
```

Out[60]:

	MR	Citation	Version
3376	MR2537368	The GAP Group, GAP—Groups, Algorithms, and Programming, Version 4.46; Aachen, Braunschweig, Fort Collins and St Andrews, 2006. http://www.gap-system.org/.	4.46

In [61]:
```
fix_version('MR2537368','4.4.6')
```

In [62]:
```
corpus_df[corpus_df['Version'] == '4.49']
```

Out[62]:

	MR	Citation	Version
3464	MR2548919	The GAP Group, GAP-Groups, Algorithms, and Programming, Version 4.49, 2006, http://www.gap-system.org.	4.49
3465	MR2606860	The GAP Group, GAP-Groups, Algorithms, and Programming. Version 4.49, 2006; http://www.gap-system.org	4.49

In [63]:
```
fix_version('MR2548919','4.4.9')
fix_version('MR2606860','4.4.9')
```

Website

Now we will create a `website` coulmn to indicate if such is provided in each entry.
Then we fill each cell using a Regex to search citations for the GAP website.
It will be a binary column with Yes and No cells.
The function below iterates over Citation cells and searches for "www" or ".net" or "http" - these are the website characteristic strings, isolated after testing. If the search returns positive `Website` cell is populated with "Yes" nad if not then it is filled with "No".
Again we add a "print" statement to teach case of the loop so we can manually inspect results.

In [64]:
```
def website_check(series):
    mrno = series['MR']
    citation = series['Citation']
    version = series['Version']
    if re.search("www|\.net|http", citation, re.IGNORECASE) != None:
        print('***Provided Website***:', mrno, citation)
        return 'Yes'
    else:
        print('***Not Provided***:', mrno, citation)
        return 'No'
```

In [65]:
```
corpus_df.insert(loc=3, column='Website', value=' ') # we apply it to our data
```

```
In [66]: corpus_df['Website'] = corpus_df.apply(website_check, axis=1)
```

```
***Provided Website***: MR4056124 GAP – Groups, algorithms, programming - a s
ystem for computational discrete algebra, www.gap-system.org.
***Provided Website***: MR3942387 Delgado, M., García-Sánchez, P.A., Morais,
J.: "Numerical Sgps", A GAP package for numerical semi-groups. https://gap-pa
ckages.github.io/numericalsgps. (https://gap-packages.github.io/numericalsgp
s.) Accessed 19 Aug 2017
MR3493240
***Provided Website***: MR3942387 The GAP Group: GAP—groups, algorithms, and
 programming, version 4.7.5 (2014). http://www.gap-system.org. (http://www.ga
p-system.org.) Accessed 19 Aug 2017
***Provided Website***: MR3354065 The GAP – Groups, Algorithms and Programmin
g. Version 4.4.12, 2008. www.gap-system.org.
***Provided Website***: MR3646312 The $\ssf{GAP}$ Group, $\ssf{GAP}$–Groups,
 Algorithms, and Programming, 4.7.8, 2015, http://www.gap-system.org. (htt
p://www.gap-system.org.)
***Not Provided***: MR1864795 M. Schönert et al. GAP - Groups, Algorithms, an
d Programming (Lehrsthul D für Mathematik, Reinisch-Westflische Technische Ho
chschule, Aachen, Germany, fifth ed., 1995.)
***Provided Website***: MR2287843 The GAP Group, GAP - Groups, Algorithms, an
```

Merging the two dataframes with the equivalent of SQL `join`

- The `MR` column in `corpus_df` dataframe has the letters "MR" preceeding each number, first we will remove these letters, using Regex, so the the MR number format is the same in both datasets.

In [67]:
```
corpus_df['MR'] = corpus_df['MR'].str.extract('(\d+)', expand=False)
corpus_df
```

```
<ipython-input-67-2ab7a7ba2ac5>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  corpus_df['MR'] = corpus_df['MR'].str.extract('(\d+)', expand=False)
```

Out[67]:

	MR	Citation	Version	Website
0	4056124	GAP – Groups, algorithms, programming - a system for computational discrete algebra, www.gap-system.org.	Unknown	Yes
1	3942387	Delgado, M., García-Sánchez, P.A., Morais, J.: "Numerical Sgps", A GAP package for numerical semi-groups. https://gap-packages.github.io/numericalsgps. A...	Package	Yes
2	3942387	The GAP Group: GAP—groups, algorithms, and programming, version 4.7.5 (2014). http://www.gap-system.org. Accessed 19 Aug 2017	4.7.5	Yes
3	3354065	The GAP – Groups, Algorithms and Programming. Version 4.4.12, 2008. www.gap-system.org.	4.4.12	Yes
4	3646312	The \ssf$GAP$ Group, \ssf$GAP$–Groups, Algorithms, and Programming, 4.7.8, 2015, http://www.gap-system.org.	4.7.8	Yes
...	...	...	...	...
3537	3988630	M. Delgado, P. A. García-Sánchez and J. Morais. Numericalsgps: a \ssf$gap$ package on numerical semigroups, (http://www.gap-system.org/Packages/numeri...	Package	Yes
3538	1801202	L.H. Soicher, GRAPE: a system for computing with graphs and groups, in: L. Finkelstein and W.M. Kantor, eds., Groups and Computation, DIMACS Series in Di...	Package	Yes
3539	2558870	L. Bartholdi, Functionally recursive groups, http://www.gap-systems.org/Manuals/pkg/fr/doc/manual.pdf.	Unknown	Yes
3540	2824780	X. Sun, C. Liu, D. Li and J. Gao, On duality gap in binary quadratic programming, Available from: http://www.optimization-online.org/DB_FILE/2010/01/2512...	Unknown	Yes
3541	1981371	Schönert M. et al., Groups, Algorithms and Programming (1997), http://www-gap.dcs.st-and.ac.uk/gap.	Unknown	Yes

3532 rows × 4 columns

```
In [68]: corpus_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3532 entries, 0 to 3541
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   MR        3532 non-null   object
 1   Citation  3532 non-null   object
 2   Version   3532 non-null   object
 3   Website   3532 non-null   object
dtypes: object(4)
memory usage: 179.8+ KB
```

The data from GAP Bibliography has Null values across the columns, this is indicated by the difference in the count of Non-Null entries in each coilumn. However this issue will be sorted by the merge process, as we will use `corpus_df` MR numbers as a base column to join the two data-frames on.

```
In [69]: bib_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3367 entries, 0 to 3366
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   MR                3159 non-null   object
 1   Author            3367 non-null   object
 2   Journal           3047 non-null   object
 3   Year              3363 non-null   object
 4   Publication Type  3367 non-null   object
 5   MSC               3252 non-null   object
dtypes: object(6)
memory usage: 79.0+ KB
```

With the following code we are joining the two datasets on the `MR` column and using `corpus_df` as a base.
The resulting dataset will have as many lines as `corpus_df` but all columns from `bib_df` will be added, hence we will have much more information to work with.
Rows that were in `bib_df` but had no matching MR number in `corpus_df` will be left behind, because we would not have Citation text for them, hence they are not useful for further analysis.

```
In [70]: merged_df = pd.merge(bib_df, corpus_df, on='MR', how='right', indicator=True)
```

In [71]:

```
merged_df.info() # to inspect for Null values and data-types of each column.
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3533 entries, 0 to 3532
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype   
---  ------            --------------  -----   
 0   MR                3533 non-null   object  
 1   Author            3526 non-null   object  
 2   Journal           3430 non-null   object  
 3   Year              3521 non-null   object  
 4   Publication Type  3526 non-null   object  
 5   MSC               3526 non-null   object  
 6   Citation          3533 non-null   object  
 7   Version           3533 non-null   object  
 8   Website           3533 non-null   object  
 9   _merge            3533 non-null   category
dtypes: category(1), object(9)
memory usage: 155.3+ KB
```

We need to remove any rows not containing Year value as they will be also of little use for our analysis. We will also correct they `Year` column data type to Integer, again.

In [72]:

```
type(merged_df['Year'][3])
```

Out[72]:

```
str
```

In [73]:

```
merged_df = merged_df.dropna(subset=['Year'])
merged_df['Year'] = merged_df['Year'].astype(np.int64)
```

In [74]:

```
merged_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3521 entries, 0 to 3532
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype   
---  ------            --------------  -----   
 0   MR                3521 non-null   object  
 1   Author            3521 non-null   object  
 2   Journal           3430 non-null   object  
 3   Year              3521 non-null   int64   
 4   Publication Type  3521 non-null   object  
 5   MSC               3521 non-null   object  
 6   Citation          3521 non-null   object  
 7   Version           3521 non-null   object  
 8   Website           3521 non-null   object  
 9   _merge            3521 non-null   category
dtypes: category(1), int64(1), object(8)
memory usage: 168.6+ KB
```

In [75]:

```
type(merged_df['Year'][3])
```

Out[75]:

```
numpy.int64
```

We can use the following iteration loop to browse the resulting merged dataframe. By borwsing the raw data we can make sure everything is alright and spot any remaining issues or anomalies. In our case there are some remaining special characters, which we will remove as best as we can.

In [76]:
```
for index, row in merged_df.iterrows():
    print(row['MR'], row['Citation'])
```

```
4056124 GAP – Groups, algorithms, programming - a system for computational di
screte algebra, www.gap-system.org.
3942387 Delgado, M., García-Sánchez, P.A., Morais, J.: "Numerical Sgps", A GA
P package for numerical semi-groups. https://gap-packages.github.io/numerical
sgps. (https://gap-packages.github.io/numericalsgps.) Accessed 19 Aug 2017
MR3493240
3942387 The GAP Group: GAP—groups, algorithms, and programming, version 4.7.5
(2014). http://www.gap-system.org. (http://www.gap-system.org.) Accessed 19 A
ug 2017
3354065 The GAP – Groups, Algorithms and Programming. Version 4.4.12, 2008. w
ww.gap-system.org.
3646312 The $\ssf{GAP}$ Group, $\ssf{GAP}$–Groups, Algorithms, and Programmin
g, 4.7.8, 2015, http://www.gap-system.org. (http://www.gap-system.org.)
1864795 M. Schönert et al. GAP - Groups, Algorithms, and Programming (Lehrsth
ul D für Mathematik, Reinisch-Westflische Technische Hochschule, Aachen, Germ
any, fifth ed., 1995.)
2287843 The GAP Group, GAP - Groups, Algorithms, and Programming, Version 4.
3; 2002, (http://www.gap-system.org).
2175389 The GAP Group, GAP-Groups, Algorithms, and programming, Version 4.3;
```

We use Regex to further purify the `Citation` column, removing some remaining special characters, that we noticed during manual scrolling over the data.

In [77]:
```
merged_df['Citation'] = merged_df['Citation'].str.replace(r'[\\\$\{\}\^]', '')
merged_df['Citation'] = merged_df['Citation'].str.replace(r'(ssf)', '')
```

```
<ipython-input-77-e299e3edd306>:1: FutureWarning: The default value of regex wi
ll change from True to False in a future version.
  merged_df['Citation'] = merged_df['Citation'].str.replace(r'[\\\$\{\}\^]',
'')
<ipython-input-77-e299e3edd306>:2: FutureWarning: The default value of regex wi
ll change from True to False in a future version.
  merged_df['Citation'] = merged_df['Citation'].str.replace(r'(ssf)', '')
```

We remove the unnecessary `merge` column and add a `Length` column to reflect the character lenght of each citation.

In [78]:

```
merged_df = merged_df.drop(['_merge'], axis=1)
merged_df['Length'] = merged_df['Citation'].apply(len)
merged_df = merged_df.dropna()
merged_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3430 entries, 0 to 3532
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   MR                3430 non-null   object
 1   Author            3430 non-null   object
 2   Journal           3430 non-null   object
 3   Year              3430 non-null   int64 
 4   Publication Type  3430 non-null   object
 5   MSC               3430 non-null   object
 6   Citation          3430 non-null   object
 7   Version           3430 non-null   object
 8   Website           3430 non-null   object
 9   Length            3430 non-null   int64 
dtypes: int64(2), object(8)
memory usage: 187.6+ KB
```

Creating the Accuracy Score column

I have decided to award each citation with one accuracy point for:

- providing some kind of version (either GAP version or some sort of package version)
- providing a website (either the official GAP website or a package website)
- Citation longer than 90 characters (because too short citations do not contain enough information)
  First we create the column, then we apply to it a function, which checks `Version` , `Website` , and `Length` columns and awards points accordingly.

In [79]:

```
merged_df['Accuracy Score'] = 0
merged_df['Accuracy Score'] = merged_df['Accuracy Score'].astype(int)
```

In [80]:
```
def accuracy_calculator(series):
    mrno = series['MR']
    citation = series['Citation']
    version = series['Version']
    website = series['Website']
    score = series['Accuracy Score']
    dal = series['Length']

    if version != 'Unknown':
        score += 1

    if website != 'No':
        score += 1

    if dal >= 90:
        score += 1

    return score
```

In [81]:
```
merged_df['Accuracy Score'] = merged_df.apply(accuracy_calculator, axis=1)
```

In [82]:
```
merged_df['Accuracy Score'].value_counts() # overview of the results
```

Out[82]:
```
3    2671
2     376
1     359
0      24
Name: Accuracy Score, dtype: int64
```

Now we split the extended dataset in two dataframes for further analysis

Pure GAP citations - citing GAP software, not a Package.

In [83]:
```
gap_df = merged_df[merged_df['Version'] != 'Package']
gap_df = gap_df.dropna()
gap_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2645 entries, 0 to 3532
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   MR                2645 non-null   object
 1   Author            2645 non-null   object
 2   Journal           2645 non-null   object
 3   Year              2645 non-null   int64 
 4   Publication Type  2645 non-null   object
 5   MSC               2645 non-null   object
 6   Citation          2645 non-null   object
 7   Version           2645 non-null   object
 8   Website           2645 non-null   object
 9   Length            2645 non-null   int64 
 10  Accuracy Score    2645 non-null   int64 
dtypes: int64(3), object(8)
memory usage: 165.3+ KB
```

In [84]:
```
versions_cited = gap_df['Version'].unique() # the same as ver_list but for the g
versions_cited = np.sort(versions_cited)
versions_cited
```

Out[84]:
```
array(['3.0', '3.1', '3.2', '3.3', '3.4', '3.4.3', '3.4.4', '4.1', '4.10',
       '4.10.0', '4.10.1', '4.10.2', '4.11', '4.11.0', '4.2', '4.3',
       '4.4', '4.4.10', '4.4.11', '4.4.12', '4.4.2', '4.4.3', '4.4.4',
       '4.4.5', '4.4.6', '4.4.7', '4.4.9', '4.5', '4.5.3', '4.5.4',
       '4.5.5', '4.5.6', '4.5.7', '4.6', '4.6.1', '4.6.12', '4.6.2',
       '4.6.3', '4.6.4', '4.6.5', '4.6.9', '4.7', '4.7.2', '4.7.4',
       '4.7.5', '4.7.6', '4.7.7', '4.7.8', '4.7.9', '4.8', '4.8.1',
       '4.8.10', '4.8.2', '4.8.3', '4.8.4', '4.8.5', '4.8.6', '4.8.7',
       '4.8.8', '4.8.9', '4.9', '4.9.0', '4.9.1', '4.9.2', '4.9.3',
       'Unknown'], dtype=object)
```

We will add two more columns that we will need later in the analysis `ReleaseYear` and `Delay` . Below is a dictionary we manually assembled with the help of Dr Konovalov and the GAP website. The dictionary contains the release year for each version we have in the data.

```
In [85]: release_dates = {
             # dates from archive timestamps
             '4.11.1': 2021,
             '4.11.0': 2020,
             '4.11': 2020,
             '4.10.2': 2019,
             '4.10.1': 2019,
             '4.10.0': 2018,
             '4.10': 2018,
             '4.9.3': 2018,
             '4.9.2': 2018,
             '4.9.1': 2018,
             '4.9.0': 2018,
             '4.9': 2018,
             '4.8.10': 2017, # assumption
             '4.8.9': 2017,
             '4.8.8': 2017,
             '4.8.7': 2017,
             '4.8.6': 2016,
             '4.8.5': 2016,
             '4.8.4': 2016,
             '4.8.3': 2016,
             '4.8.2': 2016, # 2016/02/20
             '4.8.1': 2016,
             '4.8': 2016,
             '4.7.9': 2015, # 2015/11/29
             '4.7.8': 2015, # 2015/06/09
             '4.7.7': 2015, # 2015/02/13
             '4.7.6': 2014, # 2014/11/15
             '4.7.5': 2014, # 2014/05/24
             '4.7.4': 2014, # 2014/02/20
             '4.7.3': 2013, # 2014/02/15
             '4.7.2': 2013, # 2013/12/01
             '4.7': 2013,
             '4.6.9': 2013,
             '4.6.5': 2013, # 2013/07/20
             '4.6.4': 2013, # 2013/05/04
             '4.6.3': 2013, # 2013/03/18
             '4.6.2': 2013, # 2013/02/02
             '4.6.12': 2013,
             '4.6.1': 2013,
             '4.6': 2013,
             '4.5.7': 2012, # 2012/12/14
             '4.5.6': 2012, # 2012/09/16
             '4.5.5': 2012, # 2012/07/16
             '4.5.4': 2012, # 2013/06/04
             '4.5.3': 2012,
             '4.5': 2012, # https://www.gap-system.org/Doc/History/history.html
             # dates below from file creation
             '4.4.12': 2008, # 2008/12/16
             '4.4.11': 2008, # 2008/12/08
             '4.4.10': 2007, # 2007/10/05
             '4.4.9': 2006,  # 2006/11/02
             '4.4.8': 2006,  # 2006/09/29
             '4.4.7': 2006,  # 2006/03/17
             '4.4.6': 2005,  # 2005/09/02
```

```
    '4.4.5': 2005,  # 2005/05/13
    '4.4.4': 2004,  # 2004/12/22
    # dates below from http://www.gap-system.org/Download/Updates/index.html
    '4.4.3': 2004,   # May 2004
    '4.4.2': 2004,  # April 2004
    # dates from http://www.gap-system.org/Doc/History/history.html
    # if not stated otherwise
    '4.4': 2004, # https://www.gap-system.org/Doc/History/history.html
    '4.3': 2002, # https://www.gap-system.org/Doc/History/history.html
    '4.2': 2000, # http://www.gap-system.org/ForumArchive/Linton.1/Steve.1/Releas
    '4.1': 1999, # https://www.gap-system.org/Doc/History/history.html
    '3.4.4': 1997, # https://www.gap-system.org/Doc/History/history.html
    '3.4.3': 1994, # https://www.gap-system.org/ForumArchive/Schoener.1/Martin.1/
    '3.4': 1994, # https://www.gap-system.org/ForumArchive/Schoener.1/Martin.1/GA
    '3.3': 1993, # https://www.gap-system.org/ForumArchive/Schoener.1/Martin.1/GA
    '3.2': 1993, # https://www.gap-system.org/Doc/History/history.html
    '3.1': 1991, # https://www.gap-system.org/Doc/History/history.html
    '3.0': 1991, # "M. Schönert (Editor), GAP 3.0 manual, Lehrstuhl D für Mathema

}
```

The following loop checks for versions that we have in the data but do not have in our Release Year dictionary.

In [86]:

```
for x in versions_cited:
    if not x in release_dates.keys():
        print(x)
```

```
Unknown
```

The following function we will use to populate the cells in the `Release Year` column.

In [87]:

```
def release_year(version):
    if version in release_dates.keys():
        return release_dates[version]
    else:
        return 'Unknown'
```

In [88]:

```
release_year('3.4')
```

Out[88]:

```
1994
```

In [89]:

```
gap_df['ReleaseYear'] = gap_df['Version'].map(release_year) # applying the functi
```

In [90]:
```
gap_df.head() # inspect results
```

Out[90]:

	MR	Author	Journal	Year	Publication Type	MSC	Citation	Version	Websit
0	4056124	Abas, M. and Vetrík, T.	Theoret. Comput. Sci.	2020	article	05C25 (05C20 20F05)	GAP – Groups, algorithms, programming - a system for computational discrete algebra, www.gap-system.org.	Unknown	Ye
2	3942387	Abbas, A. and Assi, A. and García-Sánchez, P. A.	Rev. R. Acad. Cienc. Exactas Fís. Nat. Ser. A Mat. RACSAM	2019	article	13F20 (05E15 14H50)	The GAP Group: GAP—groups, algorithms, and programming, version 4.7.5 (2014). http://www.gap-system.org. Accessed 19 Aug 2017	4.7.5	Ye
3	3354065	Abdolghafourian, A. and Iranmanesh, M. A.	Comm. Algebra	2015	article	05C25 (20B30 20E45)	The GAP – Groups, Algorithms and Programming. Version 4.4.12, 2008. www.gap-system.org.	4.4.12	Ye
4	3646312	Abdolghafourian, A. and Iranmanesh, M. A. and Niemeyer, A. C.	J. Pure Appl. Algebra	2017	article	20G40 (05C25)	The GAP Group, GAP–Groups, Algorithms, and Programming, 4.7.8, 2015, http://www.gap-system.org.	4.7.8	Ye
5	1864795	Abdollahi, A.	Houston J. Math.	2001	article	20F45 (20D60 20F19)	M. Schönert et al. GAP - Groups, Algorithms, and Programming (Lehrsthul D für Mathematik, Reinisch-Westflische Technische Hochschule, Aachen, Germany, fi...	Unknown	N

Delay column

- we will use later to analyse the difference between publication year and the year of GAP release cited by this publication.

In [91]:

```
gap_df['Delay'] = 0 # create the column, with 0 as default value for each cell
```

The following function we will use to populate `Delay` column. It will give us the difference between year of publication and year when the cited GAP version was released.

In [92]:

```
def set_delay(series):
    rel_year = series['ReleaseYear']
    year = series['Year']
    delay = series['Delay']
    if rel_year != 'Unknown':
        #print('***Package***:')
        delay = year - rel_year
    return delay
```

In [93]:

```
gap_df['Delay'] = gap_df.apply(set_delay, axis=1) # we apply it to our data
```

In [94]:

```
gap_df.info() # we can see the new column at the bottom
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2645 entries, 0 to 3532
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   MR                2645 non-null   object
 1   Author            2645 non-null   object
 2   Journal           2645 non-null   object
 3   Year              2645 non-null   int64 
 4   Publication Type  2645 non-null   object
 5   MSC               2645 non-null   object
 6   Citation          2645 non-null   object
 7   Version           2645 non-null   object
 8   Website           2645 non-null   object
 9   Length            2645 non-null   int64 
 10  Accuracy Score    2645 non-null   int64 
 11  ReleaseYear       2645 non-null   object
 12  Delay             2645 non-null   int64 
dtypes: int64(4), object(9)
memory usage: 196.3+ KB
```

GAP Packages Citations - all rows that have "Package" in the `Version` column cell.

This subset of our data we will later use to perform some specific analysis of Package citations and give a brief overview of GAP Package citation practices.

In [95]:

```
pac_df = merged_df[merged_df['Version'] == 'Package']
pac_df = pac_df.dropna()
pac_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 785 entries, 1 to 3529
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   MR                785 non-null    object
 1   Author            785 non-null    object
 2   Journal           785 non-null    object
 3   Year              785 non-null    int64 
 4   Publication Type  785 non-null    object
 5   MSC               785 non-null    object
 6   Citation          785 non-null    object
 7   Version           785 non-null    object
 8   Website           785 non-null    object
 9   Length            785 non-null    int64 
 10  Accuracy Score    785 non-null    int64 
dtypes: int64(3), object(8)
memory usage: 49.1+ KB
```

Once all the data is cleaned and prepared, we can several random samples to ensure it is all good before we pass it to Module 3 for analysis and visualisation.

In [96]:

```
# we can see the count of citations by specified length, for example
sma = gap_df[gap_df['Length'] < 90]
big = gap_df[gap_df['Length'] > 90]
print(len(sma))
print(len(big))
```

```
183
2447
```

In [97]:

```
get_c('3092787') # using this function conveniently displays all records with the
```

Out[97]:

	MR	Citation	Version	Website
354	3092787	Ballester-Bolinches A., Cosme-Llópez E., Esteban–Romero R., Permut: A GAP4 package to deal with permutability, v.0.03, available at http://personales.upv...	Package	Yes
355	3092787	The GAP Group, GAP–Groups, Algorithms, Programming, v. 4.5.7, 2012	4.5.7	No

In [98]:
```
merged_df.loc[354] # thus we can display a single row by specified index
```

Out[98]:
```
MR
3092787
Author              Ballester-Bolinches, A. and Cosme-Llópez, E. and Esteban-Ro
mero, R.
Journal                                                             Cent. Eur.
J. Math.
Year
2013
Publication Type
article
MSC                                                                       20D10
(20D20)
Citation             The GAP Group, GAP-Groups, Algorithms, Programming, v. 4.
5.7, 2012
Version
4.5.7
Website
No
Length
66
Accuracy Score
1
Name: 354, dtype: object
```

Exporting the pre-processed data to `CSV` files to be picked up by the final *Data Visualisations and Analysis* notebook.

In [99]:
```
merged_df.to_csv('full.csv', index=False, encoding='utf-8')
gap_df.to_csv('gap.csv', index=False, encoding='utf-8')
pac_df.to_csv('pac.csv', index=False, encoding='utf-8')
```