Study on Open Data Visualisation

Author

- Jesús Jiménez Sánchez

Introduction

This project considers three public datasets, obtained from https://github.com/awesomedata/awesome-public-datasets. Using these sets, analysing its content, I have created five (four for one of them) different graphics of their different information. The public datasets used in the project are the next ones:

- Gutenberg eBooks List: http://www.gutenberg.org/wiki/Gutenberg:Offline_Catalogs
- IMDb Database: http://www.imdb.com/interfaces
- NASA Exoplanet Archive: http://exoplanetarchive.ipac.caltech.edu/

Gutenberg eBooks List

The Gutenberg eBooks List is a list of all the books added to its database of free books until now. It has 57100 entries and looks like this:

```
TITLE and AUTHOR                                                     EBOOK NO.
The Clue of the Gold Coin, by Helen Wells                                57100
 [Subtitle: Vicki Barr, Flight Stewardess, #12]

Miss Crespigny, by Frances Hodgson Burnett                               57099

British Museum (Natural History) General Guide, by Various               57098

The Icknield Way, by Edward Thomas                                       57097
 [Illustrator: A. L. Collins]

The Battle of Talavera, by John Wilson Croker                            57096

A Chronicle of Jails, by Darrell Figgis                                  57095

The Autobiography of Lieutenant-General Sir Harry Smith, Baronet         57094
 of Aliwal, on the Sutlej G.C.B.

De Dochter van de Zeekapitein, door D'Arbez                              57093
 [Subtitle: Een Histories Verhaal]
 [Language: Dutch]
```

As we can see, the information it gives is:

- Title of the book
- Name of the author
- Language (if it’s not English)
- The order in which every book was added to the list
- Some extra information like subtitle, illustrator, composer…

This information is parsed from the text file using a script written in Python. The parser looks like this:

```
first_line = True
books = list()
authors = dict()
book_code = dict()
languages = list()

re_book_author = re.compile("(.+)(, by |, por |, mennessä |, door |, di )(.+)
(  *)([0-9]+)")
re_book_language = re.compile("(.*\[Language: )([a-zA-Z]+)( ?\].*)")

for line in gutenberg_file:
    if first_line:
        first_line = False
    elif line[0] == "\n":
        1+1
    elif line[0] != ' ':
        if re_book_author.match(line):
            book_name = re_book_author.search(line).group(1)
            books.append(book_name)
            aux_author = re_book_author.search(line).group(3)

            i = len(aux_author)
            i -= 1

            while i > 0:
                if aux_author[i] == ' ':
                    aux_author = aux_author[0:len(aux_author)-1]
                    i -= 1
                else:
                    i = -1

            authors[book_name] = aux_author
            book_code[book_name] = re_book_author.search(line).group(5)
    elif re_book_language.match(line):
        languages.append(re_book_language.search(line).group(2))

gutenberg_file.close()
```

After parsing and filtering all the information, it is written in csv files to be processed by the programs or algorithms that created the graphics.

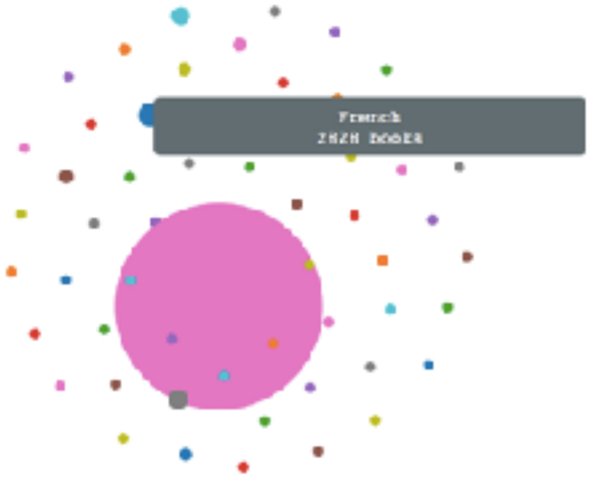
These csv files look like this:

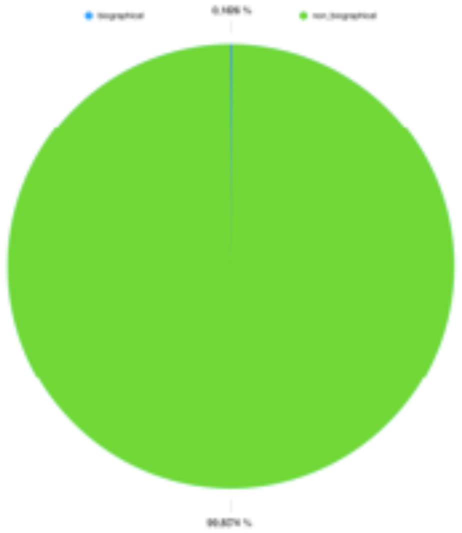
```
Author,Num_books
W.W. Jacobs,88
Anonymous,714
Charles Dickens,146
Jack London,88
Jules Verne,124
George Meredith,108
William Shakespeare,190
G. A. Henty,81
```

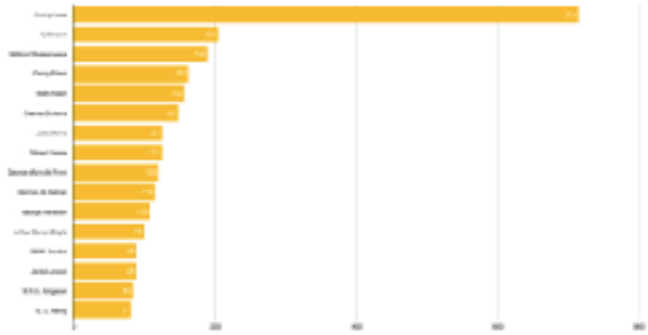
```
Language,Num_books
Swedish,178
Icelandic,8
Estonian,1
Telugu,6
Gascon,1
Romanian,2
English,39655
```

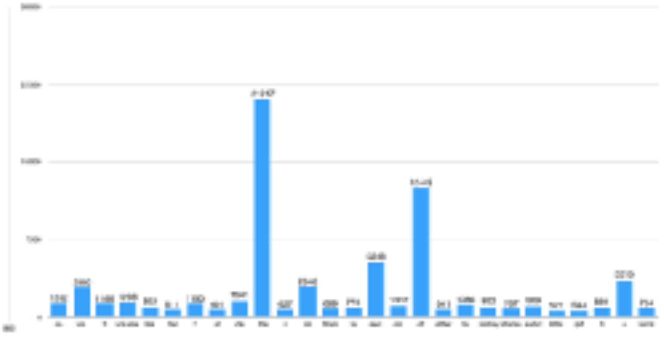
These csv files were used in a series of programs, being d3js (a Javascript library) for displaying the number of books per language and Apple iWork Numbers for the rest of graphics.

These are the obtained graphics:









IMDb Database

The IMDb Database is a list of all the movies, shorts, TV series, TV movies, TV shorts, TV mini series, videos and video games added to their database until now. It has 4990866 entries and looks like this:

```
tconst      titleType   primaryTitle      originalTitle     isAdult
startYear   endYear     runtimeMinutes    genres
tt0000001   short Carmencita  Carmencita  0     1894  \N    1
Documentary,Short
tt0000002   short Le clown et ses chiens  Le clown et ses chiens  0     1892  \N
5     Animation,Short
tt0000003   short Pauvre Pierrot    Pauvre Pierrot    0     1892  \N    4
Animation,Comedy,Romance
tt0000004   short Un bon bock Un bon bock 0     1892  \N    \N
Animation,Short
tt0000005   short Blacksmith Scene  Blacksmith Scene  0     1893  \N    1
Short
tt0000006   short Chinese Opium Den Chinese Opium Den 0     1894  \N    1
Short
tt0000007   short Corbett and Courtney Before the Kinetograph     Corbett and
Courtney Before the Kinetograph     0     1894  \N    1     Short,Sport
tt0000008   short Edison Kinetoscopic Record of a Sneeze    Edison Kinetoscopic
Record of a Sneeze      0     1894  \N    1     Documentary,Short
tt0000009   movie Miss Jerry  Miss Jerry  0     1894  \N    45    Romance
```

As we can see, the information it gives is:

- Code identifier
- Title
- Check if it’s an adult title
- Start and end year
- Length
- Genre

This information is parsed from the text file using a script written in Python. The parser looks like this:

```
first_line = True
titles = list()
types = set()
years = set()
genres = set()
title_type = dict()
title_year = dict()
title_length = dict()
title_genres = dict()

re_line = re.compile("(.*)\t(.+)\t(.+)\t(.*)\t(.*)\t(.+)\t(.*)\t(.+)\t(.+)")

for line in titles_basics_file:
    if first_line:
        first_line = False
    elif re_line.match(line):
        if re_line.search(line).group(2) != "tvEpisode":
            type = re_line.search(line).group(2)
            title = re_line.search(line).group(3)
            year = re_line.search(line).group(6)
            length = re_line.search(line).group(8)
            genre = re_line.search(line).group(9)

            titles.append(title)
            types.add(type)
            years.add(year)

            if genre != "\\N":
                for elem in genre.split(","):
                    genres.add(elem)
            else:
                genres.add("unknown")

            title_type[title] = type

            if year != "\\N":
                title_year[title] = year
            else:
                title_year[title] = "unknown"

            if length != "\\N":
                title_length[title] = length
            else:
                title_length[title] = ""

            if genre != "\\N":
                title_genres[title] = genre.split(",")
            else:
                title_genres[title] = ["unknown"]

titles_basics_file.close()
```

After parsing and filtering all the information, it is written in csv files to be processed by the programs or algorithms that created the graphics.

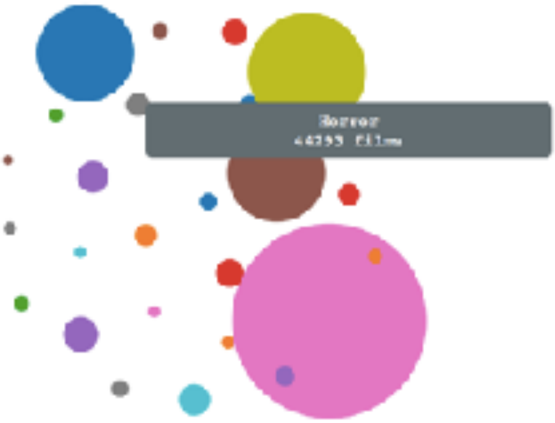
These csv files look like this:

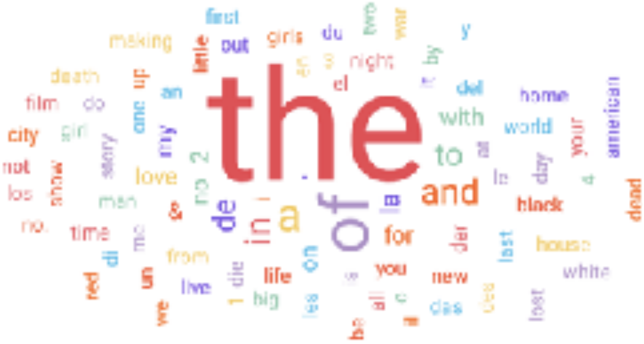
```
Genre,Num_films
Sci-Fi,26113
Crime,39800
Romance,52945
Animation,55143
Music,64857
Comedy,245418
War,10599
Horror,44293
```

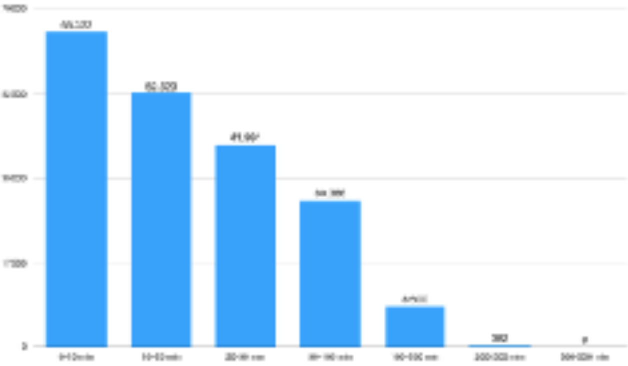
```
Year,Num_films
1948,2131
1949,2346
1942,1884
1943,1660
1940,1931
1941,1953
1946,1756
1947,1896
```

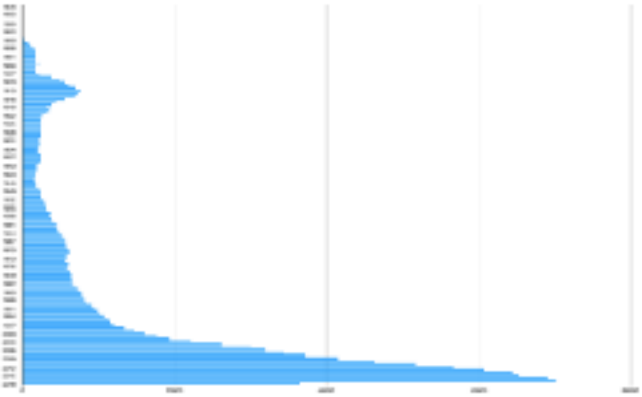
These csv files were used in a series of programs, being d3js (a Javascript library) for displaying the number of movies per genre, Infogram to make the most used words *word cloud* and Apple iWork Numbers for the rest of graphics.

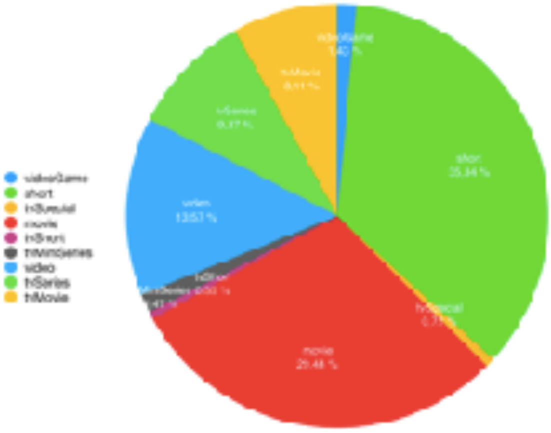
These are the obtained graphics:











NASA Exoplanet Archive

The NASA Exoplanet Archive is a list of all the discovered until now. It has 3725 entries and looks like this:

```
loc_rowid   pl_discmethod     pl_pnum     pl_orbper   st_dist     pl_name     
pl_facility
1      Radial Velocity   1     326.03000000      110.62      11 Com b    Xinglong 
Station
2      Radial Velocity   1     516.21997000      119.47      11 UMi b    
Thueringer Landessternwarte Tautenburg
3      Radial Velocity   1     185.84000000      76.39 14 And b    Okayama 
Astrophysical Observatory
4      Radial Velocity   1     1773.40002000     18.15 14 Her b    W. M. Keck 
Observatory
5      Radial Velocity   1     798.50000000      21.41 16 Cyg B b  Multiple 
Observatories
6      Radial Velocity   1     993.30000000      73.10 18 Del b    Okayama 
Astrophysical Observatory
7      Imaging     1           145.00      1RXS J160929.1-210524 b Gemini 
Observatory
8      Radial Velocity   1     30.35060000 97.75 24 Boo b    Okayama 
Astrophysical Observatory
```

As we can see, the information it gives is:

- Discovery method
- Number of planets in the system
- Time it takes to orbit around its star (in days)
- Distance
- Name
- Discovery facility

This information is parsed from the text file using a script written in Python. The parser looks like this:

```
first_line = True
pl_name = list()
pl_discmethod = dict()
pl_pnum = dict()
pl_orbper = dict()
st_dist = dict()
pl_facility = dict()

discmethods = set()
facilities = set()

re_line = re.compile("(.*)\t(.*)\t(.*)\t(.*)\t(.*)\t(.*)\t(.*)")

for line in exoplanets_file:
    if first_line:
        first_line = False
    elif re_line.match(line):
        name = re_line.search(line).group(6)
        discmethod = re_line.search(line).group(2)
        pnum = re_line.search(line).group(3)
        orbper = re_line.search(line).group(4)
        dist = re_line.search(line).group(5)
        facility = re_line.search(line).group(7)

        pl_name.append(name)
        discmethods.add(discmethod)
        facilities.add(facility)

        pl_discmethod[name] = discmethod
        pl_pnum[name] = pnum

        if orbper != "":
            pl_orbper[name] = float(orbper)
        else:
            pl_orbper[name] = 9999999999999

        st_dist[name] = dist
        pl_facility[name] = facility

exoplanets_file.close()
```

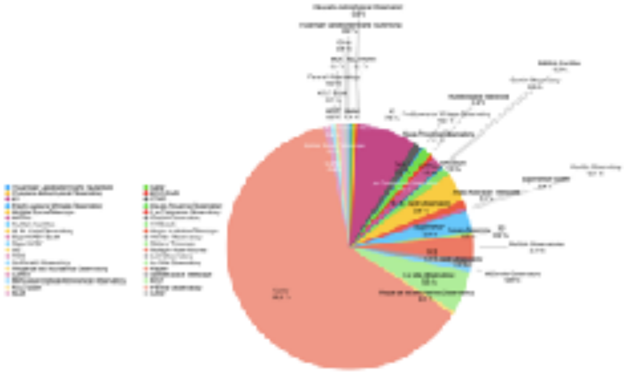
After parsing and filtering all the information, it is written in csv files to be processed by the programs or algorithms that created the graphics.

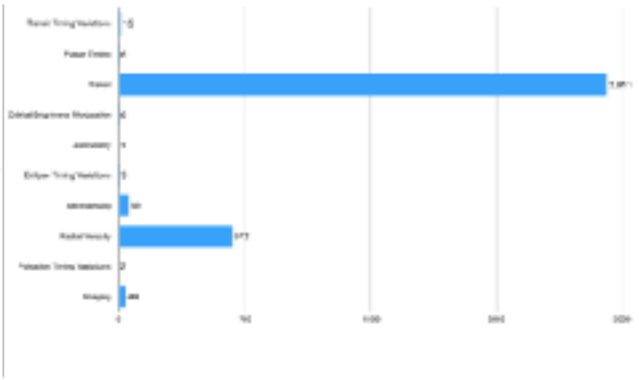
These csv files look like this:

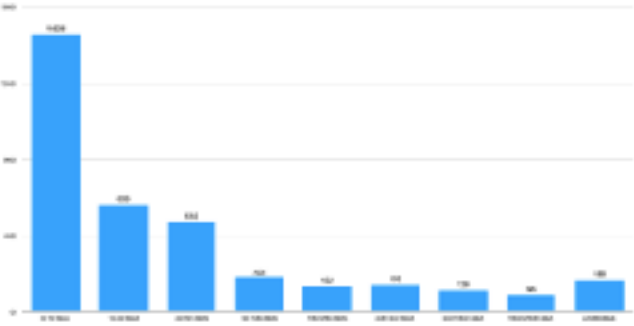
```
Facility,Times
Thueringer Landessternwarte Tautenburg,8
Qatar,5
Oak Ridge Observatory,1
Okayama Astrophysical Observatory,21
Acton Sky Portal Observatory,1
Parkes Observatory,2
KELT-North,5
Palomar Observatory,2

Planet,Orbital period
HD 25171 b,1845.0
Kepler-1540 b,125.4131177
Kepler-1026 b,36.5156053
Kepler-1118 b,38.6715075
Kepler-449 b,12.58242
K2-86 b,8.77683
HAT-P-13 c,446.27
HAT-P-13 b,2.91625
```

These csv files were used in a series of programs, being draw.io for displaying the number of planets in the system and Apple iWork Numbers for the rest of graphics.

These are the obtained graphics:







100
2160
796
59
6
7
204
408