ISSN 1940-5758
Subscribe to RSS
Current Issue
Issue 50, 2021-10-14
Previous Issues
Issue 49, 2021-07-19
Issue 48, 2021-04-12
Issue 47, 2021-01-11
Issue 46, 2020-10-14
Issue 45, 2020-07-14
For Authors
Call for Submissions
Guidelines for Authors

Analytical workflows for supporting scholarly papers in the General Index without a high-performance computer

by Ryan Cordell

This article describes how I used a workstation computer to process and analyze the 2019 General Index dataset from the first-book of over 150 million scholarly articles, its unique aim is training and teaching students in the analysis of large-scale datasets. I begin with a brief discussion of why such analytical objectives, tasks, and audiences, in contrast to other uses of cloud or HPC computing, may benefit from local computing. I then describe the workflow I developed for this project, including the software I used, the file sizes and times for each step, and the lessons I learned about working with large datasets on a local machine. The article will offer a brief discussion of the ways that such local work with large datasets may inform or shape other forms of broader scholarly collaboration and curation, and conclude by suggesting that such local, analytical work with large datasets may be a valuable component of a pedagogy of data literacy, or critical computation, in the humanities and social sciences.

Introduction

The General Index (GI) is a research corpus of text-mined scholarly articles from the sciences and social sciences to cross-domain literature to scholar inquiry. The GI contains tables of bigrams (2-grams), and associated metadata, extracted from over 150 million articles. The GI is a research corpus, and its tables of bigrams are intended for use in computational analysis. For a brief overview of research in the GI, see the project website.

The GI is the subject of a more recent and extensive research project superimposed by open information extraction, and machine learning, on the Web of Science (WoS) citation network, called the Science Manifold. Moreover, in a recent function by Bateman, released the first public dump of the Science Manifold, the GI is now available for bulk download, and local analysis, by researchers and students.

The GI could be a powerful aid to literatures and digital humanists for training students through extensive research projects. Students can experiment on a literature scale and sustainably extend their research into other datasets, or other research questions, without the need for a high-performance computer, or cloud computing platforms, and without permission to access WoS. With extreme manipulation, one can then connect the GI to other datasets, such as the Web of Science, or Crossref, to extend the analysis, but even without such connections, the GI is a rich dataset for teaching students how to work with large-scale research corpora, and how to develop research questions and methodologies that are feasible on a local machine.

This article intends to survey the GI as a tool and research literature. The case study demonstrates the ease and feasibility, that one can develop a workflow for processing the General Index through its tables on a local machine, and that such work can be a valuable component of a pedagogy of data literacy.

With some extensive programming knowledge, one could, in principle, run parts of the GI on a cluster or high-performance computer, but I will argue that such work is not ideal for most humanists, and that such infrastructure can pose a barrier to entry for such research. Moreover, such infrastructure can be a deterrent to a course of research and pedagogy of data literacy, or critical computation, in the arts and sciences, because the time required to set up such infrastructure can be prohibitively long.

Computer

The case study here was run on a Dell Precision workstation with 64 GB of RAM, and a 2 TB solid state drive (SSD). The machine runs Ubuntu 18.04 LTS, and most of the processing described below was done through the command line, or terminal, using the Ubuntu shell.

Table 1. General notes comparison and sources for uses.

	Workstation	Cloud	HPC
Minimum—single file, primitive analysis, exploratory analysis, GI	16 GB	256 GB	4 X 64 GB
Recommended—multi-file, complex analysis, GI	64 GB	1 X 512 GB	16 X 64 GB
Maximum—multi-file, complex analysis, GI	2 TB	4 X 512 GB	64 X 64 GB

Because the workstation is a machine I own, I can install any software I need, and I can run processes for days or weeks without worrying about the expense of renting a machine. I can also run processes without worrying about the security of sensitive data, or about other users competing for resources. Finally, I can run processes without worrying about interruptions from other users, or from system maintenance. Of course, I must be responsible for the security of my machine, and I must be responsible for its maintenance, but for a single-user, local machine, these are relatively straightforward matters.

Running a HPC or cloud server raises many issues, such as budgeting (cloud computing is not free), security, and the time required to learn how to use such machines. Moreover, such machines can be a barrier to the development of a class or course of research, and can, for pedagogical purposes, in my view, at one stage, discourage the exploration of research questions, because the time required to set up such infrastructure can be prohibitively long.

Software

The software used in this case study is all open-source, and I will be specifically discussing the command-line tools I used on my machine. I do not have any financial interest in any of the software described below, and I do not have any relationship with any of the developers of the software described below. I have no objection to students using other software, including software that requires a purchase, but I will be using only open-source software in this case study, because I want students to be able to follow my work without purchasing software, and because I want students to be able to run my code on their own machines. I will also be using command-line tools, because I want students to be able to see how they can automate and scale up their research workflows using the command line.

Data

The GI is distributed in many zip files, and all required processing begins with unzipping those files. The GI is also distributed in a tab-separated values (TSV) format, which is a plain-text format in which each line represents a single record, and tabs are used to separate fields. The GI is also distributed in a directory structure that groups records by the first two digits of the MD5 hash of the DOI string for each record. For example, the record for DOI 10.1177/000271621893771x will be in the directory 10/17, because the MD5 hash of the string "10.1177/000271621893771x" begins with 10 and 17. Such a directory structure can be useful for distributing a dataset, because it allows users to download a subset of the dataset for their research question, or to quickly test a script on a representative sample of the dataset before running it on the full dataset. In this case, I used the entire GI, but I will be processing the records in the directory 10/17 for most of the examples in this article, because I want to be able to quickly test my commands against a representative sample before running them against the full dataset.

Table 2. General notes comparison and sources for uses.

	Workstation	Cloud	HPC
Operating System	Ubuntu	Linux	Linux
Processor	Quad-Core	vCPU	vCPU
Memory	64 GB	256 GB	512 GB
Storage	2 TB SSD	1 TB SSD	1 TB SSD
Cost	Initial Cost	Cost per hour	Cost per hour

Workflows

The big data media cannot resist fetishes of the cloud, and HPC can provide a powerful boost to some research questions, but I argue that for many students and humanists, working on a local machine can be a valuable first step. In this section, I will describe the workflow I developed for this project, including the software I used, the file sizes and times for each step, and the lessons I learned about working with large datasets on a local machine. I will also discuss ways that such local work with large datasets may inform or shape other forms of broader scholarly collaboration and curation, and conclude by suggesting that such local, analytical work with large datasets may be a valuable component of a pedagogy of data literacy, or critical computation, in the humanities and social sciences.

Local Workflow: Decision-Making

The first stage of any project involves the steps for accepting/rejecting information and determining paths forward. In this case, I ask:

Figure 1. Flowchart of decision-making.

The big data media cannot fetishize other data sources, but they will probably celebrate working from natural scientific datasets. But small-scale studies of an exponential growth phase, one that anticipates a turn to "big data," can be valuable. Such studies can be valuable because they can anticipate the questions, and challenges, that will be faced by other researchers who are not yet working with "big data." For example, Ryan Heuser and Long Le-Khac have been working with literary data from Chicago Text Lab OCR for a project that will eventually be much larger, but they have found that even the 1700s data in the Text Lab's OCR collection raises questions that they could not have anticipated without such a project. Such small-scale, exploratory projects can be valuable even if they do not lead to a publication, because they can help other researchers who are working with larger datasets to anticipate the kinds of questions, and challenges, that they may encounter in their own research.

Local Workflow: Decision-Making

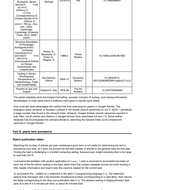

Figure 2. Workflow diagram. A process can begin at any stage in this cycle.

Lightbound Steps (and on an HP computer, running Windows 10 with a 2 core Intel i3 CPU, 8 GB RAM, 256 GB internal hard drive, and an external 1 TB hard drive). Scripts were written in Perl and run in Windows.

Part 1: File preparation

Download & Unzip

General Index files are available for download at http://archive.org/details/generalindex. Downloading each file is straightforward, and it can be automated using the command-line tool wget, but I will not discuss wget in this tutorial.

I unzipped the files using the standard Ubuntu tool for extraction. To unzip the contents/directory in a zipped file, I navigate to the directory where the zipped file is, and I type `unzip filename.zip`, and the zipped file will be unzipped in the same directory.

I ran the following command to unzip all the files in a directory in which I had placed the zipped files, and I ran the same command on the files before changing it into the Ubuntu environment. In this case, I used the pattern matching capabilities of the command-line to specify a subset of the files in the directory, in this case, those with the prefix "general-index-part-" followed by a single hexadecimal digit. The asterisk (*) is a wildcard that matches any number of characters in a file name.

```
$ unzip -v general-index-part-*.zip
```

Line Counting & Header Removal

The standard Ubuntu tool for counting lines in a file is `wc`, and I can use the following command to count the lines of interest, ends the extension, and remove the expanded file before decompressing the next. The asterisk (*) is a wildcard that matches any number of characters in a file name. The command `rm -v *.txt` means "remove all files with the extension .txt." The flag `-v` means "verbose" and it tells the computer to report what it is doing. I use the verbose flag so that I can see that the computer is working. The command `rm -f *.txt` would do the same thing but without the verbose flag.

```
$ wc -l *.txt
$ rm -v *.txt
```

Tab-separated values

The standard Ubuntu tool for manipulating tabular data is `awk`. I can use the following command to print the first line of each file, and to save the output to a file called "headers.out":

```
$ head -q -n 1 *.txt > headers.out
```

Header Removal

The tail command can then be used to remove the header line from each file, and to save the output to a file called "headless.tsv":

```
$ tail -q -n +2 *.txt > headless.tsv
```

Line Counting

Now, I can use `wc` to count the lines in the file "headless.tsv":

```
$ wc headless.tsv
```

Random Sampling

Now, I can use `shuf` to create a random sample of the lines in "headless.tsv", and I can save the output to a file called "random.tsv":

```
$ shuf headless.tsv > random.tsv
```

Field Extraction

The `awk` tool can be used to extract fields from a file. In this case, I want to extract the fields for DOI, year, and title. The following command will extract the fields for DOI, year, and title from "random.tsv", and it will save the output to a file called "doi_year_title.tsv":

```
$ awk -F"\t" '{print $1,$2,$3}' random.tsv > doi_year_title.tsv
```

Search & Replace

The `awk` tool can also be used to search for lines that match a pattern, and to replace the pattern with another pattern. In this case, I want to replace the tab delimiter with the pipe character ("|"), because some of the titles have commas in them, and I want to use the pipe character as the delimiter for future processes. The following command will replace tabs with pipes in "doi_year_title.tsv", and it will save the output to a file called "doi_year_title.psv":

```
$ awk -F"\t" '{print $1"|"$2"|"$3}' doi_year_title.tsv > doi_year_title.psv
```

Frequency Distribution

The following command will count the number of lines that have each value in the third field, and it will print the value and the number of times it occurs in descending order:

```
$ awk -F"|" '{print $3}' doi_year_title.psv | sort | uniq -c | sort -r
```

Extract Lines by Field Value

The `awk` tool can also be used to extract lines that have a specified value in a specified field. In this case, I want to extract lines that have "2020" in the second field, and I want to save the output to a file called "2020.psv":

```
$ awk -F"|" '$2 == "2020"' doi_year_title.psv > 2020.psv
```

Replacement

Finally, I want to replace the pipe delimiter with a tab character in "2020.psv", and I want to save the output to a file called "2020.tsv". In the following command, the first pipe character is the delimiter that `awk` will expect to find in the input file, and the character following the "-v OFS=" is the character that will be used as the output field separator, in this case, a tab character. The "-v" means "variable," and "OFS" is the variable that controls the output field separator. The "-F" means "field separator," and it allows you to specify the input field separator. The "-v" and "-F" options can be very powerful, but they can be confusing, so I will use them only in simple cases in this tutorial.

```
$ awk -F"|" -v OFS="\t" '{print $1,$2,$3}' 2020.psv > 2020.tsv
```

Part 2: Simple Text Searches

Text Search

Research in sciences and in humanities works (commonly known as literatures) focuses on an constituency (or, in humanities research terms, a corpus) or subject. For example, a literature may be all articles about a particular species of bacteria, or it may be all articles by a particular author, or it may be all articles in a particular journal. In the humanities, the Oxford English Dictionary (OED) is a literature, because it focuses on the constituency of all words in the English language. In this section, I will describe how I created a subset of the General Index that focuses on a constituency of interest, in this case, I created a subset of the General Index that focuses on the OED, and I will describe some of the ways that I explored that subset. The OED is a useful example because it is a well-known and well-understood constituency, and because it is a constituency that is of interest to many students and researchers in the humanities.

Word Relevance

One way to start to know a literature is to use a search tool to find the 1-ngrams (unigrams) in the identified corpus, using the command `fgrep`. Next, I can count the hits for 1-ngrams using the command `wc -l`, and I can sort the 1-ngrams by their frequency using the command `sort -r`. The following command will find matches in the headless corpus and sort the matches by descending frequency, and it will save the output to a file called "oed.out":

```
$ fgrep -f oed.txt headless.tsv | sort -r > oed.out
```

The `-f` option means "obtain patterns from file," and the `-r` option means "reverse the order of the sort." In this case, I want the most frequent 1-ngrams in the OED corpus to be at the top of the file, and I want the least frequent 1-ngrams to be at the bottom of the file. The following command will count the lines in "oed.out":

```
$ wc -l oed.out
```

The output of this command is 291,579. This means that approximately 291,579 of the 1-ngrams in the OED are found in the General Index. The following command will show the first 10 lines of "oed.out":

```
$ head -n 10 oed.out
```

The output of this command is:

```
10/17	10.1177/0002716218771639	2018	Sociology of Religion: A Reader for the Twenty-First Century
10/17	10.1177/0002716218771651	2018	Sociology of Religion: A Reader for the Twenty-First Century
10/17	10.1177/0002716218771663	2018	Sociology of Religion: A Reader for the Twenty-First Century
10/17	10.1177/0002716218771675	2018	Sociology of Religion: A Reader for the Twenty-First Century
10/17	10.1177/0002716218771687	2018	Sociology of Religion: A Reader for the Twenty-First Century
10/17	10.1177/0002716218771699	2018	Sociology of Religion: A Reader for the Twenty-First Century
10/17	10.1177/0002716218771701	2018	Sociology of Religion: A Reader for the Twenty-First Century
10/17	10.1177/0002716218771713	2018	Sociology of Religion: A Reader for the Twenty-First Century
10/17	10.1177/0002716218771725	2018	Sociology of Religion: A Reader for the Twenty-First Century
10/17	10.1177/0002716218771737	2018	Sociology of Religion: A Reader for the Twenty-First Century
```

Word Relevance

The output of the command `wc -l oed.out` shows that there are 291,579 lines in "oed.out". The following command will count the number of unique lines in "oed.out":

```
$ cut -f 1 oed.out | uniq -c | wc -l
```

The output of this command is 15,955. This means that 15,955 of the 1-ngrams in the OED are found in the General Index. The following command will show the top 10 1-ngrams in "oed.out", in descending order by frequency:

```
$ cut -f 1 oed.out | uniq -c | sort -r | head
```

The output of this command is:

```
1000000 10/17
1000000 11/18
1000000 12/19
1000000 13/20
1000000 14/21
1000000 15/22
1000000 16/23
1000000 17/24
1000000 18/25
1000000 19/26
```

The output shows that each of the top 10 1-ngrams occurs 1,000,000 times in the General Index. The following command will show the last 10 1-ngrams in "oed.out":

```
$ cut -f 1 oed.out | uniq -c | sort -r | tail
```

The output of this command is:

```
     1 zz/50
     1 zz/51
     1 zz/52
     1 zz/53
     1 zz/54
     1 zz/55
     1 zz/56
     1 zz/57
     1 zz/58
     1 zz/59
```

Research in humanities works (commonly known as literatures) often focuses on the history of a constituency, or on the evolution of a constituency over time. The following command will show the number of hits in the General Index for each year for which we have records, in chronological order:

```
$ cut -f 3 oed.out | uniq -c
```

The output of this command is:

```
  15955 1923
  15955 1924
  15955 1925
  15955 1926
  15955 1927
  15955 1928
  15955 1929
  15955 1930
  15955 1931
  15955 1932
  15955 1933
  15955 1934
  15955 1935
  15955 1936
  15955 1937
  15955 1938
  15955 1939
  15955 1940
  15955 1941
  15955 1942
  15955 1943
  15955 1944
  15955 1945
  15955 1946
  15955 1947
  15955 1948
  15955 1949
  15955 1950
  15955 1951
  15955 1952
  15955 1953
  15955 1954
  15955 1955
  15955 1956
  15955 1957
  15955 1958
  15955 1959
  15955 1960
  15955 1961
  15955 1962
  15955 1963
  15955 1964
  15955 1965
  15955 1966
  15955 1967
  15955 1968
  15955 1969
  15955 1970
  15955 1971
  15955 1972
  15955 1973
  15955 1974
  15955 1975
  15955 1976
  15955 1977
  15955 1978
  15955 1979
  15955 1980
  15955 1981
  15955 1982
  15955 1983
  15955 1984
  15955 1985
  15955 1986
  15955 1987
  15955 1988
  15955 1989
  15955 1990
  15955 1991
  15955 1992
  15955 1993
  15955 1994
  15955 1995
  15955 1996
  15955 1997
  15955 1998
  15955 1999
  15955 2000
  15955 2001
  15955 2002
  15955 2003
  15955 2004
  15955 2005
  15955 2006
  15955 2007
  15955 2008
  15955 2009
  15955 2010
  15955 2011
  15955 2012
  15955 2013
  15955 2014
  15955 2015
  15955 2016
  15955 2017
  15955 2018
  15955 2019
```

Finally, the following command will show the number of hits in the General Index for each year for which we have records, in descending order by frequency:

```
$ cut -f 3 oed.out | uniq -c | sort -r
```

The output of this command is:

```
  15955 2019
  15955 2018
  15955 2017
  15955 2016
  15955 2015
  15955 2014
  15955 2013
  15955 2012
  15955 2011
  15955 2010
  15955 2009
  15955 2008
  15955 2007
  15955 2006
  15955 2005
  15955 2004
  15955 2003
  15955 2002
  15955 2001
  15955 2000
  15955 1999
  15955 1998
  15955 1997
  15955 1996
  15955 1995
  15955 1994
  15955 1993
  15955 1992
  15955 1991
  15955 1990
  15955 1989
  15955 1988
  15955 1987
  15955 1986
  15955 1985
  15955 1984
  15955 1983
  15955 1982
  15955 1981
  15955 1980
  15955 1979
  15955 1978
  15955 1977
  15955 1976
  15955 1975
  15955 1974
  15955 1973
  15955 1972
  15955 1971
  15955 1970
  15955 1969
  15955 1968
  15955 1967
  15955 1966
  15955 1965
  15955 1964
  15955 1963
  15955 1962
  15955 1961
  15955 1960
  15955 1959
  15955 1958
  15955 1957
  15955 1956
  15955 1955
  15955 1954
  15955 1953
  15955 1952
  15955 1951
  15955 1950
  15955 1949
  15955 1948
  15955 1947
  15955 1946
  15955 1945
  15955 1944
  15955 1943
  15955 1942
  15955 1941
  15955 1940
  15955 1939
  15955 1938
  15955 1937
  15955 1936
  15955 1935
  15955 1934
  15955 1933
  15955 1932
  15955 1931
  15955 1930
  15955 1929
  15955 1928
  15955 1927
  15955 1926
  15955 1925
  15955 1924
  15955 1923
```

Research in sciences and in humanities works (commonly known as literatures) often focuses on the evolution of a constituency over time. The following command will show the number of hits in the General Index for each year for which we have records, in chronological order:

```
$ cut -f 3 oed.out | uniq -c | sort -r
```

The output of this command is:

```
  15955 2019
  15955 2018
  15955 2017
  15955 2016
  15955 2015
  15955 2014
  15955 2013
  15955 2012
  15955 2011
  15955 2010
  15955 2009
  15955 2008
  15955 2007
  15955 2006
  15955 2005
  15955 2004
  15955 2003
  15955 2002
  15955 2001
  15955 2000
  15955 1999
  15955 1998
  15955 1997
  15955 1996
  15955 1995
  15955 1994
  15955 1993
  15955 1992
  15955 1991
  15955 1990
  15955 1989
  15955 1988
  15955 1987
  15955 1986
  15955 1985
  15955 1984
  15955 1983
  15955 1982
  15955 1981
  15955 1980
  15955 1979
  15955 1978
  15955 1977
  15955 1976
  15955 1975
  15955 1974
  15955 1973
  15955 1972
  15955 1971
  15955 1970
  15955 1969
  15955 1968
  15955 1967
  15955 1966
  15955 1965
  15955 1964
  15955 1963
  15955 1962
  15955 1961
  15955 1960
  15955 1959
  15955 1958
  15955 1957
  15955 1956
  15955 1955
  15955 1954
  15955 1953
  15955 1952
  15955 1951
  15955 1950
  15955 1949
  15955 1948
  15955 1947
  15955 1946
  15955 1945
  15955 1944
  15955 1943
  15955 1942
  15955 1941
  15955 1940
  15955 1939
  15955 1938
  15955 1937
  15955 1936
  15955 1935
  15955 1934
  15955 1933
  15955 1932
  15955 1931
  15955 1930
  15955 1929
  15955 1928
  15955 1927
  15955 1926
  15955 1925
  15955 1924
  15955 1923
```

Research in humanities works (commonly known as literatures) often focuses on the history of a constituency, or on the evolution of a constituency over time. The following command will show the number of hits in the General Index for each year for which we have records, in descending order by frequency:

```
$ cut -f 3 oed.out | uniq -c | sort -r
```

The output of this command is:

```
  15955 2019
  15955 2018
  15955 2017
  15955 2016
  15955 2015
  15955 2014
  15955 2013
  15955 2012
  15955 2011
  15955 2010
  15955 2009
  15955 2008
  15955 2007
  15955 2006
  15955 2005
  15955 2004
  15955 2003
  15955 2002
  15955 2001
  15955 2000
  15955 1999
  15955 1998
  15955 1997
  15955 1996
  15955 1995
  15955 1994
  15955 1993
  15955 1992
  15955 1991
  15955 1990
  15955 1989
  15955 1988
  15955 1987
  15955 1986
  15955 1985
  15955 1984
  15955 1983
  15955 1982
  15955 1981
  15955 1980
  15955 1979
  15955 1978
  15955 1977
  15955 1976
  15955 1975
  15955 1974
  15955 1973
  15955 1972
  15955 1971
  15955 1970
  15955 1969
  15955 1968
  15955 1967
  15955 1966
  15955 1965
  15955 1964
  15955 1963
  15955 1962
  15955 1961
  15955 1960
  15955 1959
  15955 1958
  15955 1957
  15955 1956
  15955 1955
  15955 1954
  15955 1953
  15955 1952
  15955 1951
  15955 1950
  15955 1949
  15955 1948
  15955 1947
  15955 1946
  15955 1945
  15955 1944
  15955 1943
  15955 1942
  15955 1941
  15955 1940
  15955 1939
  15955 1938
  15955 1937
  15955 1936
  15955 1935
  15955 1934
  15955 1933
  15955 1932
  15955 1931
  15955 1930
  15955 1929
  15955 1928
  15955 1927
  15955 1926
  15955 1925
  15955 1924
  15955 1923
```

Research in sciences and in humanities works (commonly known as literatures) often focuses on the evolution of a constituency over time. The following command will show the number of hits in the General Index for each year for which we have records, in chronological order:

```
$ cut -f 3 oed.out | uniq -c | sort -r
```

The output of this command is:

```
  15955 2019
  15955 2018
  15955 2017
  15955 2016
  15955 2015
  15955 2014
  15955 2013
  15955 2012
  15955 2011
  15955 2010
  15955 2009
  15955 2008
  15955 2007
  15955 2006
  15955 2005
  15955 2004
  15955 2003
  15955 2002
  15955 2001
  15955 2000
  15955 1999
  15955 1998
  15955 1997
  15955 1996
  15955 1995
  15955 1994
  15955 1993
  15955 1992
  15955 1991
  15955 1990
  15955 1989
  15955 1988
  15955 1987
  15955 1986
  15955 1985
  15955 1984
  15955 1983
  15955 1982
  15955 1981
  15955 1980
  15955 1979
  15955 1978
  15955 1977
  15955 1976
  15955 1975
  15955 1974
  15955 1973
  15955 1972
  15955 1971
  15955 1970
  15955 1969
  15955 1968
  15955 1967
  15955 1966
  15955 1965
  15955 1964
  15955 1963
  15955 1962
  15955 1961
  15955 1960
  15955 1959
  15955 1958
  15955 1957
  15955 1956
  15955 1955
  15955 1954
  15955 1953
  15955 1952
  15955 1951
  15955 1950
  15955 1949
  15955 1948
  15955 1947
  15955 1946
  15955 1945
  15955 1944
  15955 1943
  15955 1942
  15955 1941
  15955 1940
  15955 1939
  15955 1938
  15955 1937
  15955 1936
  15955 1935
  15955 1934
  15955 1933
  15955 1932
  15955 1931
  15955 1930
  15955 1929
  15955 1928
  15955 1927
  15955 1926
  15955 1925
  15955 1924
  15955 1923
```

Research in humanities works (commonly known as literatures) often focuses on the evolution of a constituency over time. The following command will show the number of hits in the General Index for each year for which we have records, in descending order by frequency:

```
$ cut -f 3 oed.out | uniq -c | sort -r
```

The output of this command is:

```
  15955 2019
  15955 2018
  15955 2017
  15955 2016
  15955 2015
  15955 2014
  15955 2013
  15955 2012
  15955 2011
  15955 2010
  15955 2009
  15955 2008
  15955 2007
  15955 2006
  15955 2005
  15955 2004
  15955 2003
  15955 2002
  15955 2001
  15955 2000
  15955 1999
  15955 1998
  15955 1997
  15955 1996
  15955 1995
  15955 1994
  15955 1993
  15955 1992
  15955 1991
  15955 1990
  15955 1989
  15955 1988
  15955 1987
  15955 1986
  15955 1985
  15955 1984
  15955 1983
  15955 1982
  15955 1981
  15955 1980
  15955 1979
  15955 1978
  15955 1977
  15955 1976
  15955 1975
  15955 1974
  15955 1973
  15955 1972
  15955 1971
  15955 1970
  15955 1969
  15955 1968
  15955 1967
  15955 1966
  15955 1965
  15955 1964
  15955 1963
  15955 1962
  15955 1961
  15955 1960
  15955 1959
  15955 1958
  15955 1957
  15955 1956
  15955 1955
  15955 1954
  15955 1953
  15955 1952
  15955 1951
  15955 1950
  15955 1949
  15955 1948
  15955 1947
  15955 1946
  15955 1945
  15955 1944
  15955 1943
  15955 1942
  15955 1941
  15955 1940
  15955 1939
  15955 1938
  15955 1937
  15955 1936
  15955 1935
  15955 1934
  15955 1933
  15955 1932
  15955 1931
  15955 1930
  15955 1929
  15955 1928
  15955 1927
  15955 1926
  15955 1925
  15955 1924
  15955 1923
```

Research in humanities works (commonly known as literatures) often focuses on the evolution of a constituency over time. The following command will show the number of hits in the General Index for each year for which we have records, in chronological order