INFO7250 33478 Engineering Big-Data Systems

Amish Garhwal- 001860961

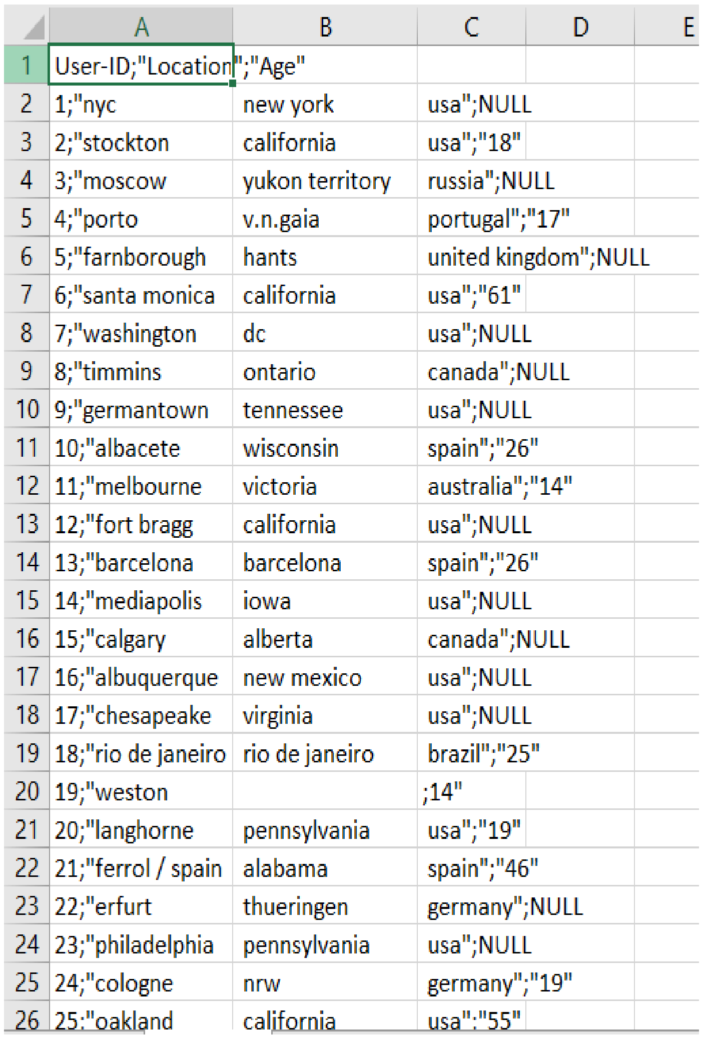
PROJECT Dataset

Link to Dataset : http://www2.informatik.uni-freiburg.de/~cziegler/BX/

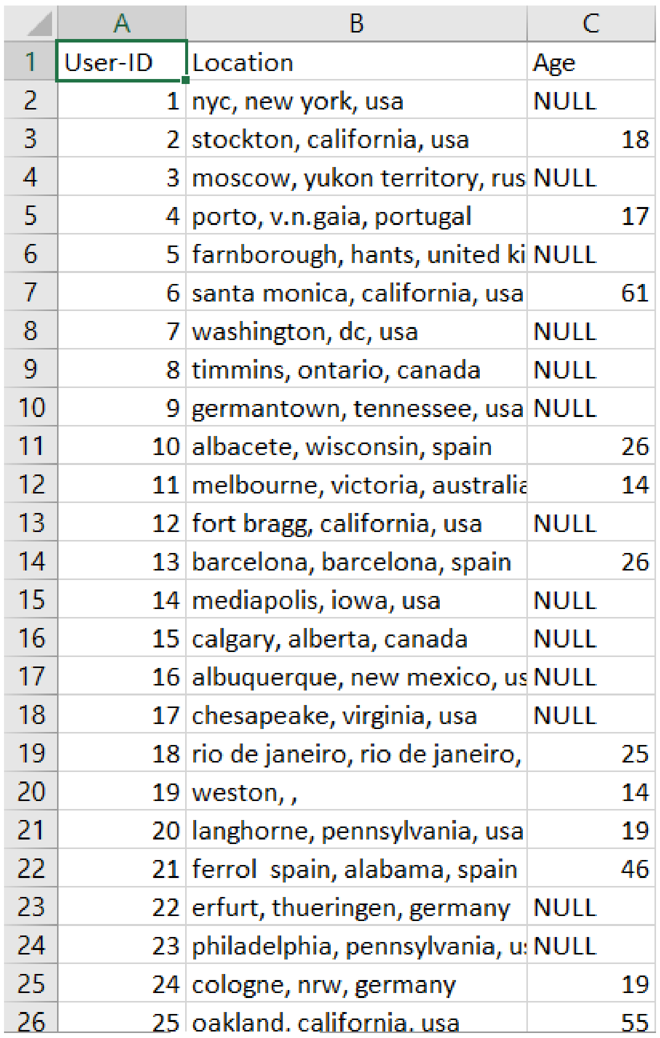
The Book-Crossing dataset comprises 3 tables.

• **BX-Users**

Contains the users. Note that user IDs (`*User-ID*`) have been anonymized and map to integers. Demographic data is provided **(`*Location*`, `*Age*`)** if available. Otherwise, these fields contain *NULL*-values. This is what it looked like when I simply opened it in Excel.



	A	B	C	D	E
1	User-ID;"Location";"Age"				
2	1;"nyc	new york	usa";NULL		
3	2;"stockton	california	usa";"18"		
4	3;"moscow	yukon territory	russia";NULL		
5	4;"porto	v.n.gaia	portugal";"17"		
6	5;"farnborough	hants	united kingdom";NULL		
7	6;"santa monica	california	usa";"61"		
8	7;"washington	dc	usa";NULL		
9	8;"timmins	ontario	canada";NULL		
10	9;"germantown	tennessee	usa";NULL		
11	10;"albacete	wisconsin	spain";"26"		
12	11;"melbourne	victoria	australia";"14"		
13	12;"fort bragg	california	usa";NULL		
14	13;"barcelona	barcelona	spain";"26"		
15	14;"mediapolis	iowa	usa";NULL		
16	15;"calgary	alberta	canada";NULL		
17	16;"albuquerque	new mexico	usa";NULL		
18	17;"chesapeake	virginia	usa";NULL		
19	18;"rio de janeiro	rio de janeiro	brazil";"25"		
20	19;"weston		;14"		
21	20;"langhorne	pennsylvania	usa";"19"		
22	21;"ferrol / spain	alabama	spain";"46"		
23	22;"erfurt	thueringen	germany";NULL		
24	23;"philadelphia	pennsylvania	usa";NULL		
25	24;"cologne	nrw	germany";"19"		
26	25;"oakland	california	usa";"55"		

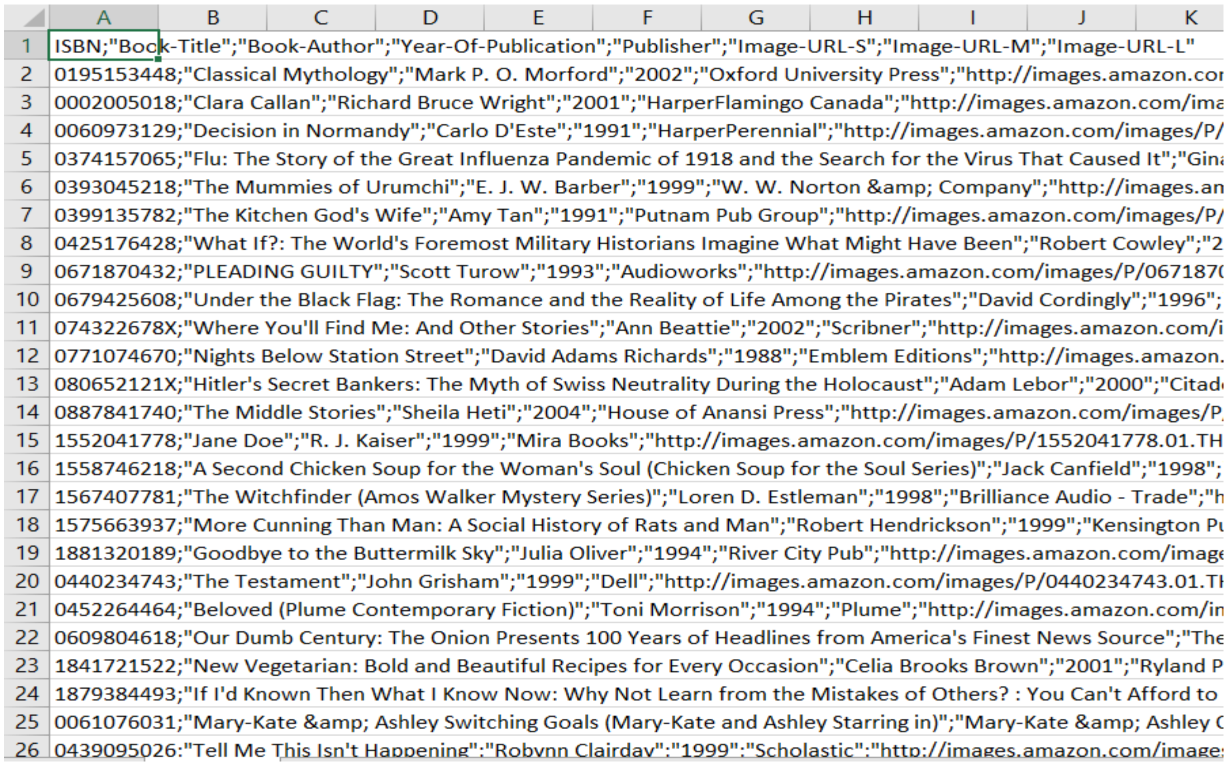


	A	B	C
1	User-ID	Location	Age
2	1	nyc, new york, usa	NULL
3	2	stockton, california, usa	18
4	3	moscow, yukon territory, rus	NULL
5	4	porto, v.n.gaia, portugal	17
6	5	farnborough, hants, united ki	NULL
7	6	santa monica, california, usa	61
8	7	washington, dc, usa	NULL
9	8	timmins, ontario, canada	NULL
10	9	germantown, tennessee, usa	NULL
11	10	albacete, wisconsin, spain	26
12	11	melbourne, victoria, australia	14
13	12	fort bragg, california, usa	NULL
14	13	barcelona, barcelona, spain	26
15	14	mediapolis, iowa, usa	NULL
16	15	calgary, alberta, canada	NULL
17	16	albuquerque, new mexico, us	NULL
18	17	chesapeake, virginia, usa	NULL
19	18	rio de janeiro, rio de janeiro,	25
20	19	weston, ,	14
21	20	langhorne, pennsylvania, usa	19
22	21	ferrol spain, alabama, spain	46
23	22	erfurt, thueringen, germany	NULL
24	23	philadelphia, pennsylvania, u:	NULL
25	24	cologne, nrw, germany	19
26	25	oakland, california, usa	55

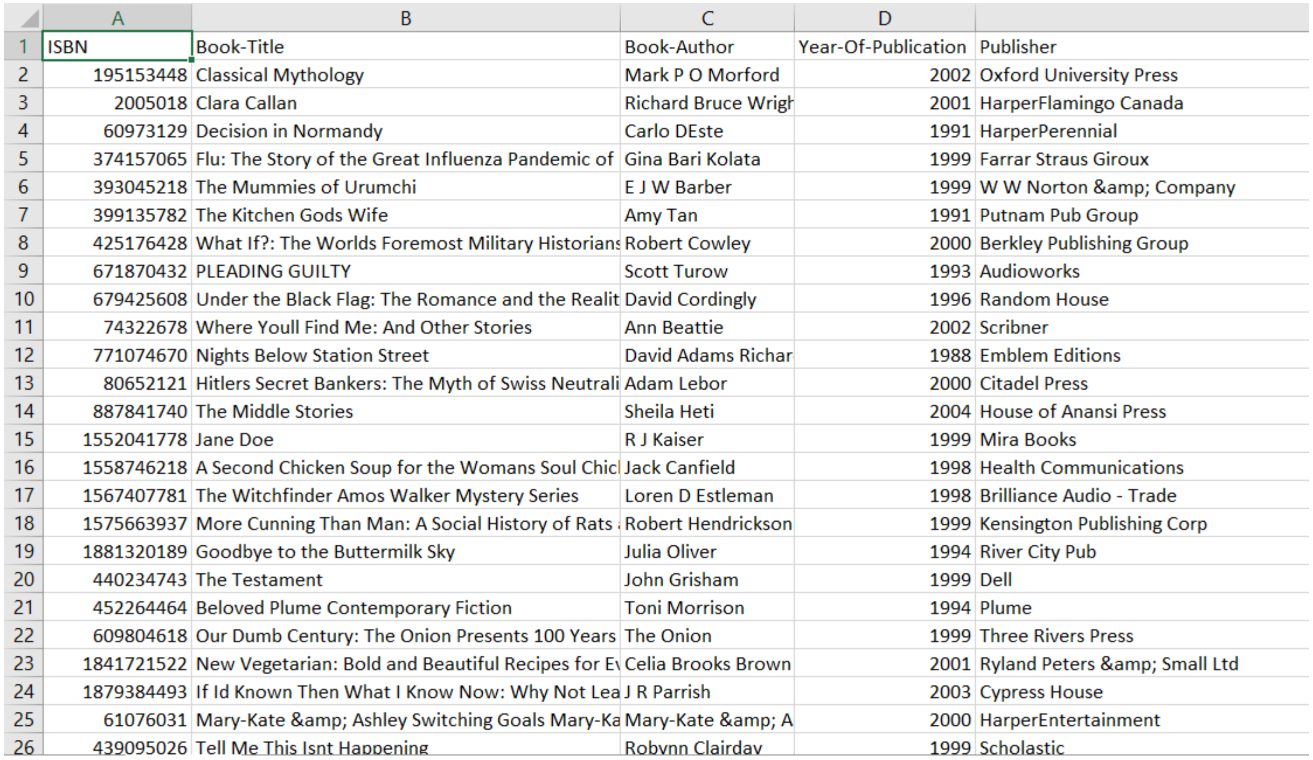
This is how it looks now.

- **BX-Books**

Books are identified by their respective ISBN. Invalid ISBNs have already been removed from the dataset. Moreover, some content-based information is given (**`*Book-Title*`, `*Book-Author*`, `*Year-Of-Publication*`, `*Publisher*`**), obtained from Amazon Web Services. Note that in case of several authors, only the first is provided. URLs linking to cover images are also given, appearing in three different flavors (**`*Image-URL-S*`, `*Image-URL-M*`, `*Image-URL-L*`**), i.e., small, medium, large. These URLs point to the Amazon web site.



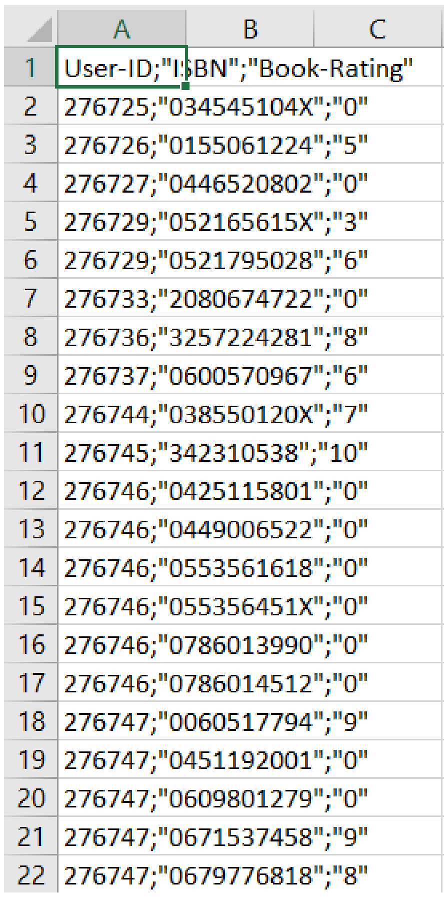
	A
1	ISBN;"Book-Title";"Book-Author";"Year-Of-Publication";"Publisher";"Image-URL-S";"Image-URL-M";"Image-URL-L"
2	0195153448;"Classical Mythology";"Mark P. O. Morford";"2002";"Oxford University Press";"http://images.amazon.co
3	0002005018;"Clara Callan";"Richard Bruce Wright";"2001";"HarperFlamingo Canada";"http://images.amazon.com/ima
4	0060973129;"Decision in Normandy";"Carlo D'Este";"1991";"HarperPerennial";"http://images.amazon.com/images/P/
5	0374157065;"Flu: The Story of the Great Influenza Pandemic of 1918 and the Search for the Virus That Caused It";"Gin
6	0393045218;"The Mummies of Urumchi";"E. J. W. Barber";"1999";"W. W. Norton &amp; Company";"http://images.a
7	0399135782;"The Kitchen God's Wife";"Amy Tan";"1991";"Putnam Pub Group";"http://images.amazon.com/images/P/
8	0425176428;"What If?: The World's Foremost Military Historians Imagine What Might Have Been";"Robert Cowley";"2
9	0671870432;"PLEADING GUILTY";"Scott Turow";"1993";"Audioworks";"http://images.amazon.com/images/P/067187
10	0679425608;"Under the Black Flag: The Romance and the Reality of Life Among the Pirates";"David Cordingly";"1996";
11	074322678X;"Where You'll Find Me: And Other Stories";"Ann Beattie";"2002";"Scribner";"http://images.amazon.com/i
12	0771074670;"Nights Below Station Street";"David Adams Richards";"1988";"Emblem Editions";"http://images.amazon.
13	080652121X;"Hitler's Secret Bankers: The Myth of Swiss Neutrality During the Holocaust";"Adam Lebor";"2000";"Citad
14	0887841740;"The Middle Stories";"Sheila Heti";"2004";"House of Anansi Press";"http://images.amazon.com/images/P
15	1552041778;"Jane Doe";"R. J. Kaiser";"1999";"Mira Books";"http://images.amazon.com/images/P/1552041778.01.TH
16	1558746218;"A Second Chicken Soup for the Woman's Soul (Chicken Soup for the Soul Series)";"Jack Canfield";"1998";
17	1567407781;"The Witchfinder (Amos Walker Mystery Series)";"Loren D. Estleman";"1998";"Brilliance Audio - Trade";"h
18	1575663937;"More Cunning Than Man: A Social History of Rats and Man";"Robert Hendrickson";"1999";"Kensington P
19	1881320189;"Goodbye to the Buttermilk Sky";"Julia Oliver";"1994";"River City Pub";"http://images.amazon.com/image
20	0440234743;"The Testament";"John Grisham";"1999";"Dell";"http://images.amazon.com/images/P/0440234743.01.T
21	0452264464;"Beloved (Plume Contemporary Fiction)";"Toni Morrison";"1994";"Plume";"http://images.amazon.com/i
22	0609804618;"Our Dumb Century: The Onion Presents 100 Years of Headlines from America's Finest News Source";"The
23	1841721522;"New Vegetarian: Bold and Beautiful Recipes for Every Occasion";"Celia Brooks Brown";"2001";"Ryland P
24	1879384493;"If I'd Known Then What I Know Now: Why Not Learn from the Mistakes of Others? : You Can't Afford to
25	0061076031;"Mary-Kate &amp; Ashley Switching Goals (Mary-Kate and Ashley Starring in)";"Mary-Kate &amp; Ashley
26	0439095026;"Tell Me This Isn't Happening";"Robynn Clairday";"1999";"Scholastic";"http://images.amazon.com/image



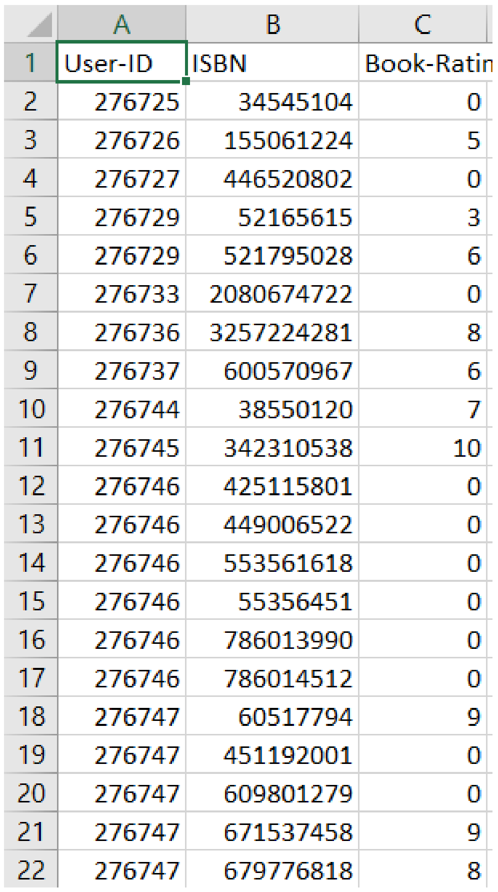
	A	B	C	D	
1	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher
2	195153448	Classical Mythology	Mark P O Morford	2002	Oxford University Press
3	2005018	Clara Callan	Richard Bruce Wrigh	2001	HarperFlamingo Canada
4	60973129	Decision in Normandy	Carlo DEste	1991	HarperPerennial
5	374157065	Flu: The Story of the Great Influenza Pandemic of	Gina Bari Kolata	1999	Farrar Straus Giroux
6	393045218	The Mummies of Urumchi	E J W Barber	1999	W W Norton &amp; Company
7	399135782	The Kitchen Gods Wife	Amy Tan	1991	Putnam Pub Group
8	425176428	What If?: The Worlds Foremost Military Historians	Robert Cowley	2000	Berkley Publishing Group
9	671870432	PLEADING GUILTY	Scott Turow	1993	Audioworks
10	679425608	Under the Black Flag: The Romance and the Realit	David Cordingly	1996	Random House
11	74322678	Where Youll Find Me: And Other Stories	Ann Beattie	2002	Scribner
12	771074670	Nights Below Station Street	David Adams Richar	1988	Emblem Editions
13	80652121	Hitlers Secret Bankers: The Myth of Swiss Neutrali	Adam Lebor	2000	Citadel Press
14	887841740	The Middle Stories	Sheila Heti	2004	House of Anansi Press
15	1552041778	Jane Doe	R J Kaiser	1999	Mira Books
16	1558746218	A Second Chicken Soup for the Womans Soul Chic	Jack Canfield	1998	Health Communications
17	1567407781	The Witchfinder Amos Walker Mystery Series	Loren D Estleman	1998	Brilliance Audio - Trade
18	1575663937	More Cunning Than Man: A Social History of Rats	Robert Hendrickson	1999	Kensington Publishing Corp
19	1881320189	Goodbye to the Buttermilk Sky	Julia Oliver	1994	River City Pub
20	440234743	The Testament	John Grisham	1999	Dell
21	452264464	Beloved Plume Contemporary Fiction	Toni Morrison	1994	Plume
22	609804618	Our Dumb Century: The Onion Presents 100 Years	The Onion	1999	Three Rivers Press
23	1841721522	New Vegetarian: Bold and Beautiful Recipes for Ev	Celia Brooks Brown	2001	Ryland Peters &amp; Small Ltd
24	1879384493	If Id Known Then What I Know Now: Why Not Lea	J R Parrish	2003	Cypress House
25	61076031	Mary-Kate &amp; Ashley Switching Goals Mary-Ka	Mary-Kate &amp; A	2000	HarperEntertainment
26	439095026	Tell Me This Isnt Happening	Robynn Clairday	1999	Scholastic

• **<u>BX-Book-Ratings</u>**

Contains the book rating information. Ratings (**`*Book-Rating*`**) are either explicit, expressed on a scale from 1-10 (higher values denoting higher appreciation), or implicit, expressed by 0.



	A	B	C
1	User-ID;"ISBN";"Book-Rating"		
2	276725;"034545104X";"0"		
3	276726;"0155061224";"5"		
4	276727;"0446520802";"0"		
5	276729;"052165615X";"3"		
6	276729;"0521795028";"6"		
7	276733;"2080674722";"0"		
8	276736;"3257224281";"8"		
9	276737;"0600570967";"6"		
10	276744;"038550120X";"7"		
11	276745;"342310538";"10"		
12	276746;"0425115801";"0"		
13	276746;"0449006522";"0"		
14	276746;"0553561618";"0"		
15	276746;"055356451X";"0"		
16	276746;"0786013990";"0"		
17	276746;"0786014512";"0"		
18	276747;"0060517794";"9"		
19	276747;"0451192001";"0"		
20	276747;"0609801279";"0"		
21	276747;"0671537458";"9"		
22	276747;"0679776818";"8"		

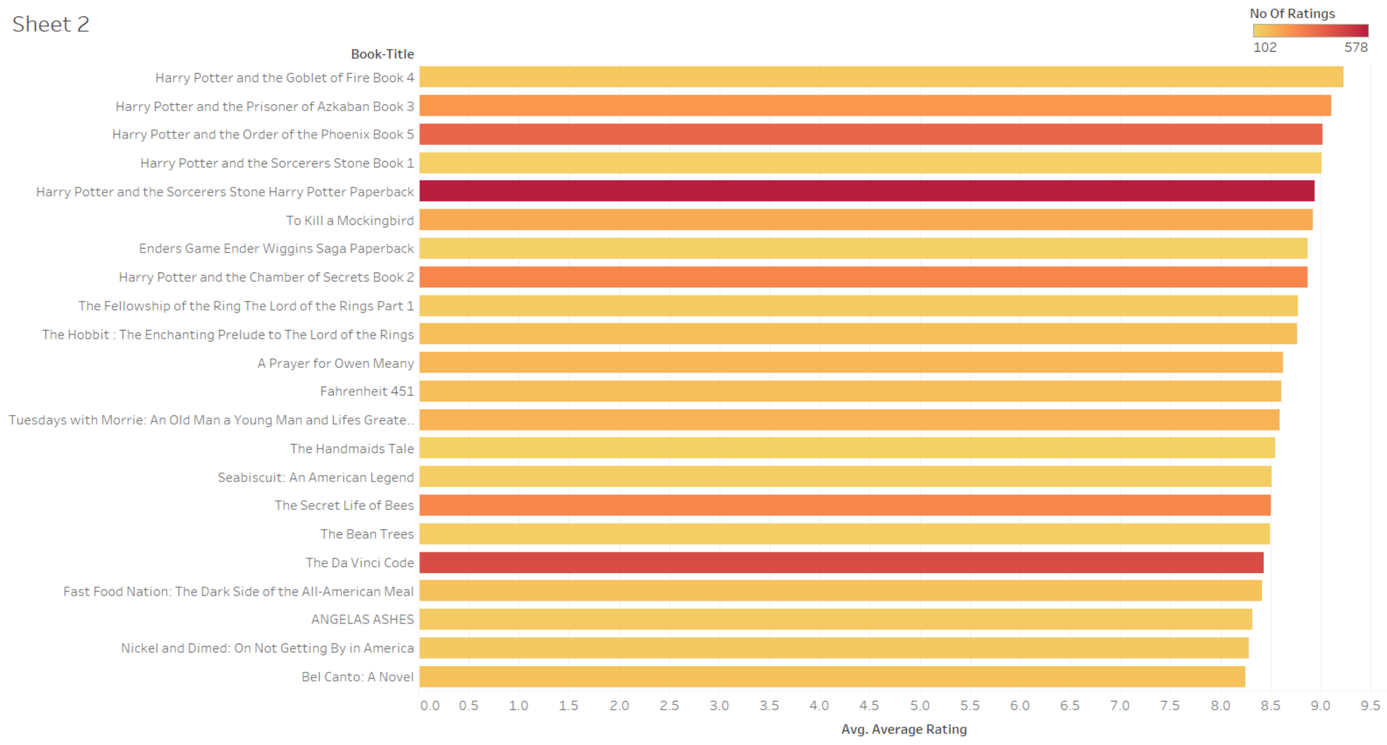


	A	B	C
1	User-ID	ISBN	Book-Ratin
2	276725	34545104	0
3	276726	155061224	5
4	276727	446520802	0
5	276729	52165615	3
6	276729	521795028	6
7	276733	2080674722	0
8	276736	3257224281	8
9	276737	600570967	6
10	276744	38550120	7
11	276745	342310538	10
12	276746	425115801	0
13	276746	449006522	0
14	276746	553561618	0
15	276746	55356451	0
16	276746	786013990	0
17	276746	786014512	0
18	276747	60517794	9
19	276747	451192001	0
20	276747	609801279	0
21	276747	671537458	9
22	276747	679776818	8

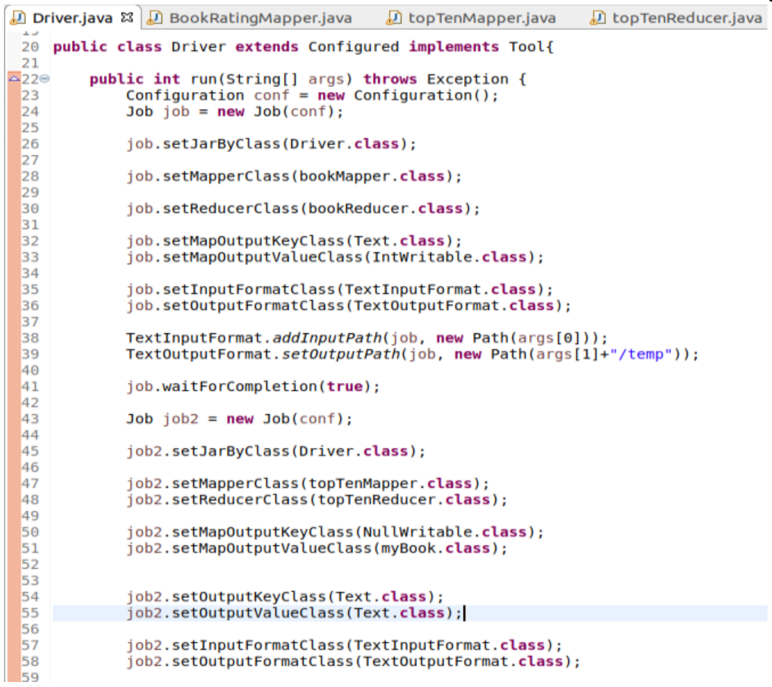
ANALYSIS

MAPREDUCE JOBS:

1. Average Rating Per Book:

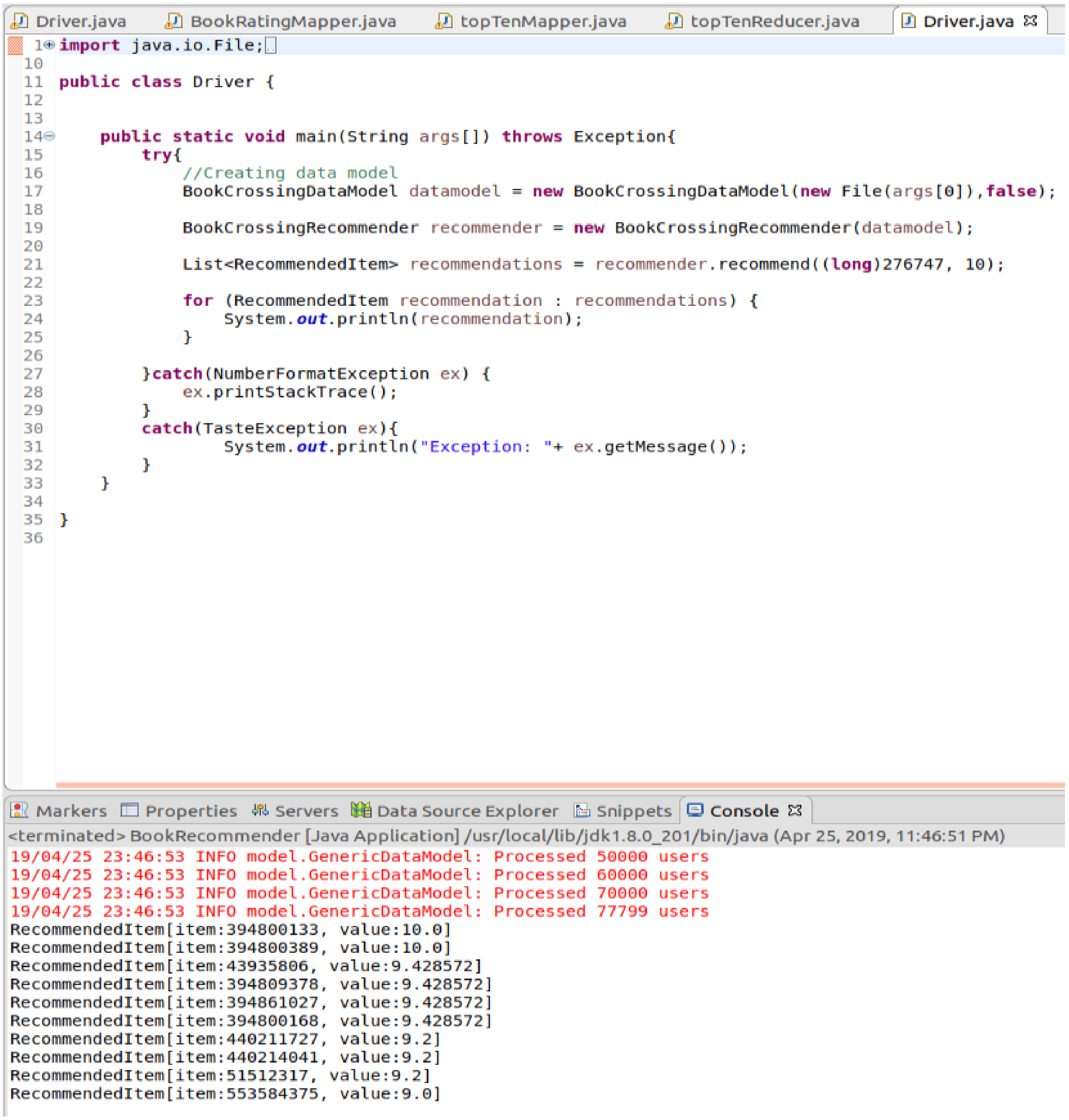


- Used **Job Chaining** to Aggregate the **Top Ten Results**.
- Use of Book Objects to store **Count** and Overall **Average Rating** of the Books.
- Limited the minimum number of reviews for each book to 100 to obtain reliable results
- Books with their bars are darker color has a greater number of reviews.



```java
public class Driver extends Configured implements Tool{

    public int run(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = new Job(conf);

        job.setJarByClass(Driver.class);

        job.setMapperClass(bookMapper.class);

        job.setReducerClass(bookReducer.class);

        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(IntWritable.class);

        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);

        TextInputFormat.addInputPath(job, new Path(args[0]));
        TextOutputFormat.setOutputPath(job, new Path(args[1]+"/temp"));

        job.waitForCompletion(true);

        Job job2 = new Job(conf);

        job2.setJarByClass(Driver.class);

        job2.setMapperClass(topTenMapper.class);
        job2.setReducerClass(topTenReducer.class);

        job2.setMapOutputKeyClass(NullWritable.class);
        job2.setMapOutputValueClass(myBook.class);


        job2.setOutputKeyClass(Text.class);
        job2.setOutputValueClass(Text.class);

        job2.setInputFormatClass(TextInputFormat.class);
        job2.setOutputFormatClass(TextOutputFormat.class);

```

2. Book Recommender:



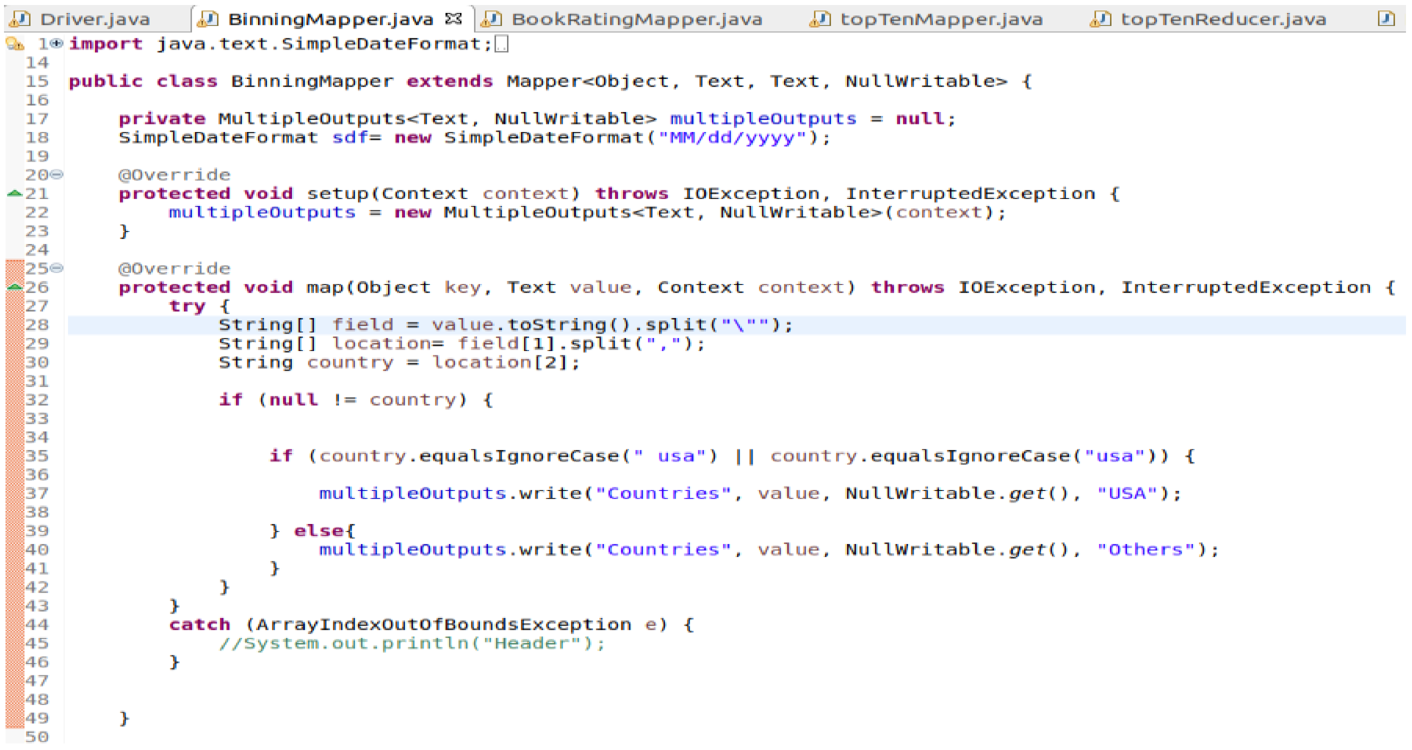
```java
import java.io.File;

public class Driver {


    public static void main(String args[]) throws Exception{
        try{
            //Creating data model
            BookCrossingDataModel datamodel = new BookCrossingDataModel(new File(args[0]),false);

            BookCrossingRecommender recommender = new BookCrossingRecommender(datamodel);

            List<RecommendedItem> recommendations = recommender.recommend((long)276747, 10);

            for (RecommendedItem recommendation : recommendations) {
                System.out.println(recommendation);
            }

        }catch(NumberFormatException ex) {
            ex.printStackTrace();
        }
        catch(TasteException ex){
                System.out.println("Exception: "+ ex.getMessage());
        }
    }

}

```

```
<terminated> BookRecommender [Java Application] /usr/local/lib/jdk1.8.0_201/bin/java (Apr 25, 2019, 11:46:51 PM)
19/04/25 23:46:53 INFO model.GenericDataModel: Processed 50000 users
19/04/25 23:46:53 INFO model.GenericDataModel: Processed 60000 users
19/04/25 23:46:53 INFO model.GenericDataModel: Processed 70000 users
19/04/25 23:46:53 INFO model.GenericDataModel: Processed 77799 users
RecommendedItem[item:394800133, value:10.0]
RecommendedItem[item:394800389, value:10.0]
RecommendedItem[item:43935806, value:9.428572]
RecommendedItem[item:394809378, value:9.428572]
RecommendedItem[item:394861027, value:9.428572]
RecommendedItem[item:394800168, value:9.428572]
RecommendedItem[item:440211727, value:9.2]
RecommendedItem[item:440214041, value:9.2]
RecommendedItem[item:51512317, value:9.2]
RecommendedItem[item:553584375, value:9.0]
```

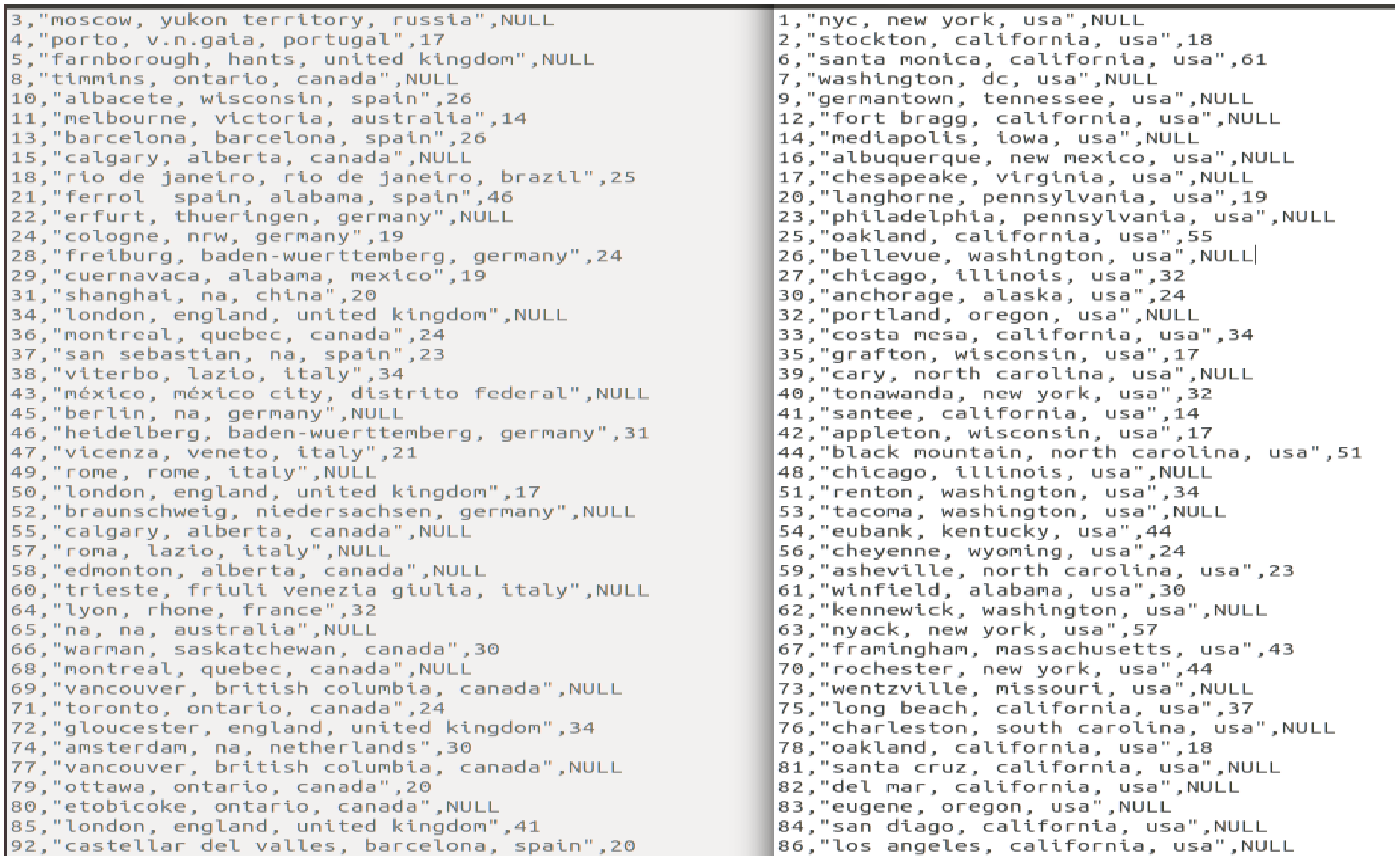
- Implemented **Mahout** to Find Book Recommendations.
- Used **BookCrossingDataModel** and **BookCrossingRecommender** to predict the recommended Books.
- Program gives decently accurate recommendations.

3. Country Binning Program

- Segregates the Input values based on the countries.
- It could be used to individually analyze the dataset and provide meaningful results.
- I separated the inputs into two categories based on countries of users, **USA** and **Others**.

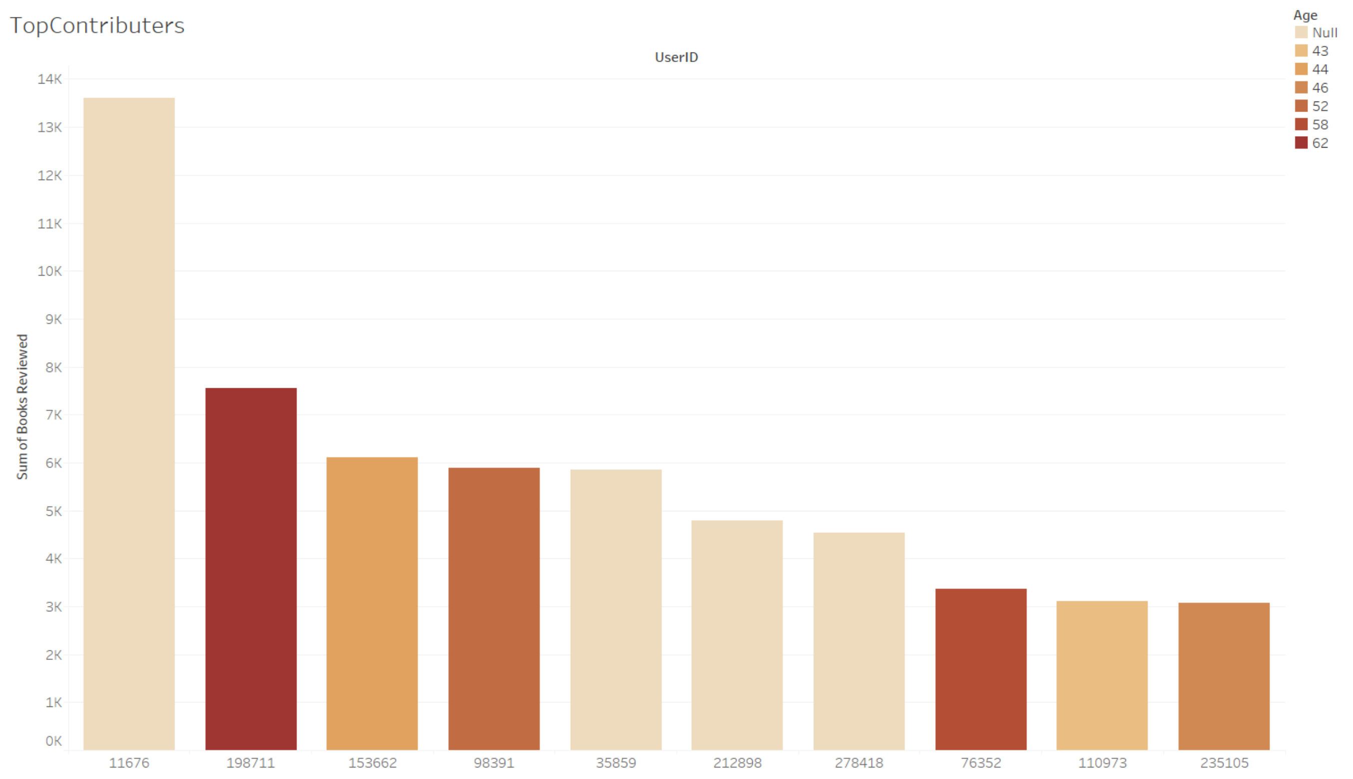

Driver.java | BinningMapper.java | BookRatingMapper.java | topTenMapper.java | topTenReducer.java

```java
import java.text.SimpleDateFormat;

public class BinningMapper extends Mapper<Object, Text, Text, NullWritable> {

    private MultipleOutputs<Text, NullWritable> multipleOutputs = null;
    SimpleDateFormat sdf= new SimpleDateFormat("MM/dd/yyyy");

    @Override
    protected void setup(Context context) throws IOException, InterruptedException {
        multipleOutputs = new MultipleOutputs<Text, NullWritable>(context);
    }

    @Override
    protected void map(Object key, Text value, Context context) throws IOException, InterruptedException {
        try {
            String[] field = value.toString().split("\"");
            String[] location= field[1].split(",");
            String country = location[2];

            if (null != country) {


                if (country.equalsIgnoreCase(" usa") || country.equalsIgnoreCase("usa")) {

                    multipleOutputs.write("Countries", value, NullWritable.get(), "USA");

                } else{
                    multipleOutputs.write("Countries", value, NullWritable.get(), "Others");
                }
            }
        }
        catch (ArrayIndexOutOfBoundsException e) {
            //System.out.println("Header");
        }


    }

```

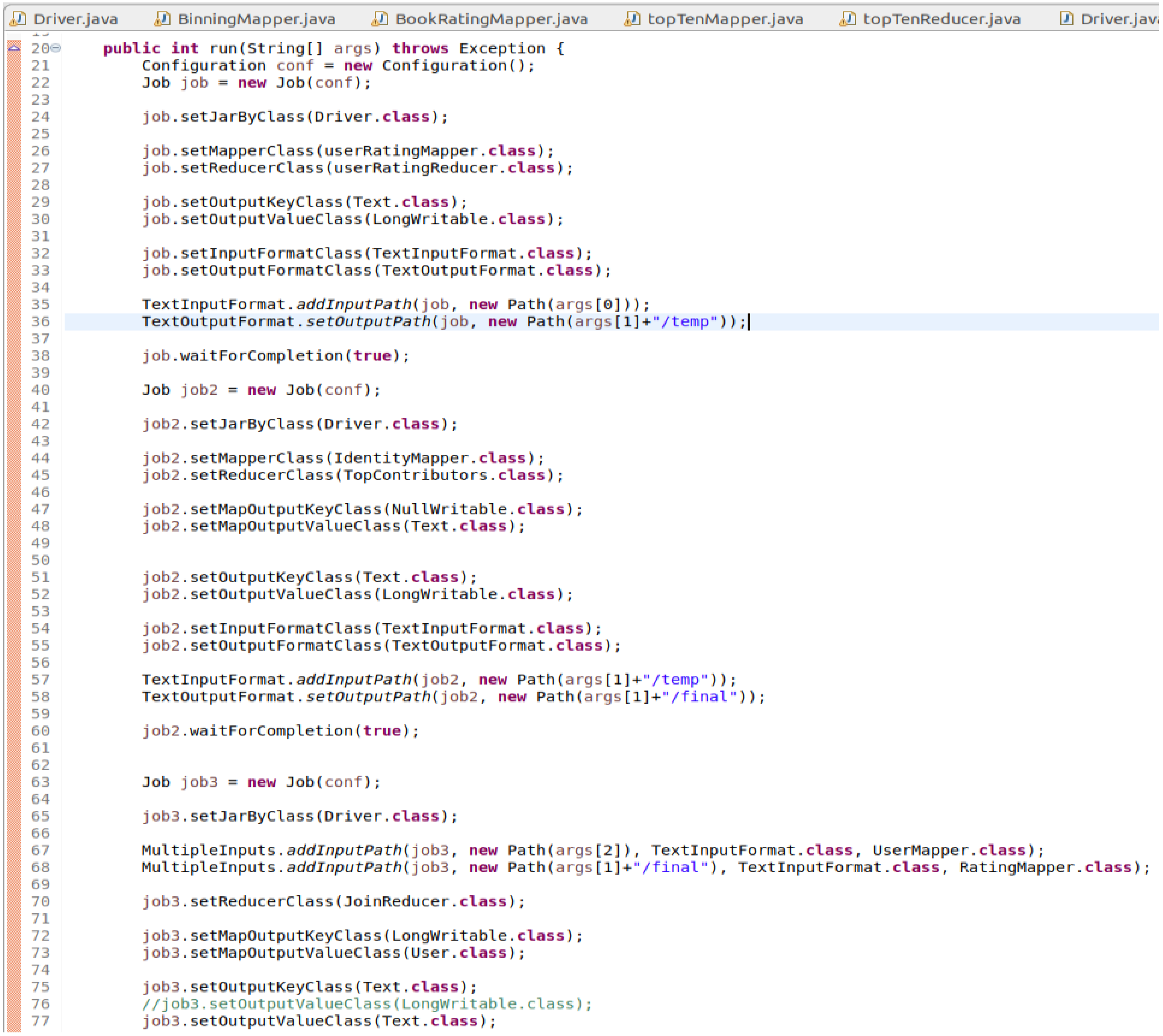

```
3,"moscow, yukon territory, russia",NULL
4,"porto, v.n.gaia, portugal",17
5,"farnborough, hants, united kingdom",NULL
8,"timmins, ontario, canada",NULL
10,"albacete, wisconsin, spain",26
11,"melbourne, victoria, australia",14
13,"barcelona, barcelona, spain",26
15,"calgary, alberta, canada",NULL
18,"rio de janeiro, rio de janeiro, brazil",25
21,"ferrol  spain, alabama, spain",46
22,"erfurt, thueringen, germany",NULL
24,"cologne, nrw, germany",19
28,"freiburg, baden-wuerttemberg, germany",24
29,"cuernavaca, alabama, mexico",19
31,"shanghai, na, china",20
34,"london, england, united kingdom",NULL
36,"montreal, quebec, canada",24
37,"san sebastian, na, spain",23
38,"viterbo, lazio, italy",34
43,"méxico, méxico city, distrito federal",NULL
45,"berlin, na, germany",NULL
46,"heidelberg, baden-wuerttemberg, germany",31
47,"vicenza, veneto, italy",21
49,"rome, rome, italy",NULL
50,"london, england, united kingdom",17
52,"braunschweig, niedersachsen, germany",NULL
55,"calgary, alberta, canada",NULL
57,"roma, lazio, italy",NULL
58,"edmonton, alberta, canada",NULL
60,"trieste, friuli venezia giulia, italy",NULL
64,"lyon, rhone, france",32
65,"na, na, australia",NULL
66,"warman, saskatchewan, canada",30
68,"montreal, quebec, canada",NULL
69,"vancouver, british columbia, canada",NULL
71,"toronto, ontario, canada",24
72,"gloucester, england, united kingdom",34
74,"amsterdam, na, netherlands",30
77,"vancouver, british columbia, canada",NULL
79,"ottawa, ontario, canada",20
80,"etobicoke, ontario, canada",NULL
85,"london, england, united kingdom",41
92,"castellar del valles, barcelona, spain",20
```

```
1,"nyc, new york, usa",NULL
2,"stockton, california, usa",18
6,"santa monica, california, usa",61
7,"washington, dc, usa",NULL
9,"germantown, tennessee, usa",NULL
12,"fort bragg, california, usa",NULL
14,"mediapolis, iowa, usa",NULL
16,"albuquerque, new mexico, usa",NULL
17,"chesapeake, virginia, usa",NULL
20,"langhorne, pennsylvania, usa",19
23,"philadelphia, pennsylvania, usa",NULL
25,"oakland, california, usa",55
26,"bellevue, washington, usa",NULL
27,"chicago, illinois, usa",32
30,"anchorage, alaska, usa",24
32,"portland, oregon, usa",NULL
33,"costa mesa, california, usa",34
35,"grafton, wisconsin, usa",17
39,"cary, north carolina, usa",NULL
40,"tonawanda, new york, usa",32
41,"santee, california, usa",14
42,"appleton, wisconsin, usa",17
44,"black mountain, north carolina, usa",51
48,"chicago, illinois, usa",NULL
51,"renton, washington, usa",34
53,"tacoma, washington, usa",NULL
54,"eubank, kentucky, usa",44
56,"cheyenne, wyoming, usa",24
59,"asheville, north carolina, usa",23
61,"winfield, alabama, usa",30
62,"kennewick, washington, usa",NULL
63,"nyack, new york, usa",57
67,"framingham, massachusetts, usa",43
70,"rochester, new york, usa",44
73,"wentzville, missouri, usa",NULL
75,"long beach, california, usa",37
76,"charleston, south carolina, usa",NULL
78,"oakland, california, usa",18
81,"santa cruz, california, usa",NULL
82,"del mar, california, usa",NULL
83,"eugene, oregon, usa",NULL
84,"san diago, california, usa",NULL
86,"los angeles, california, usa",NULL
```

4. Most Contributions by Users:



- Graph shows **top 10** User based on the number of books they have reviewed.
- The user rating dataset is **joined** with user dataset to combine the details of the top users.
- The bars with different colors indicate users with different ages, darker ones being greater in age.
- Used Job chaining of **3 jobs**.
- Used **multiple inputs** for joining of different inputs.

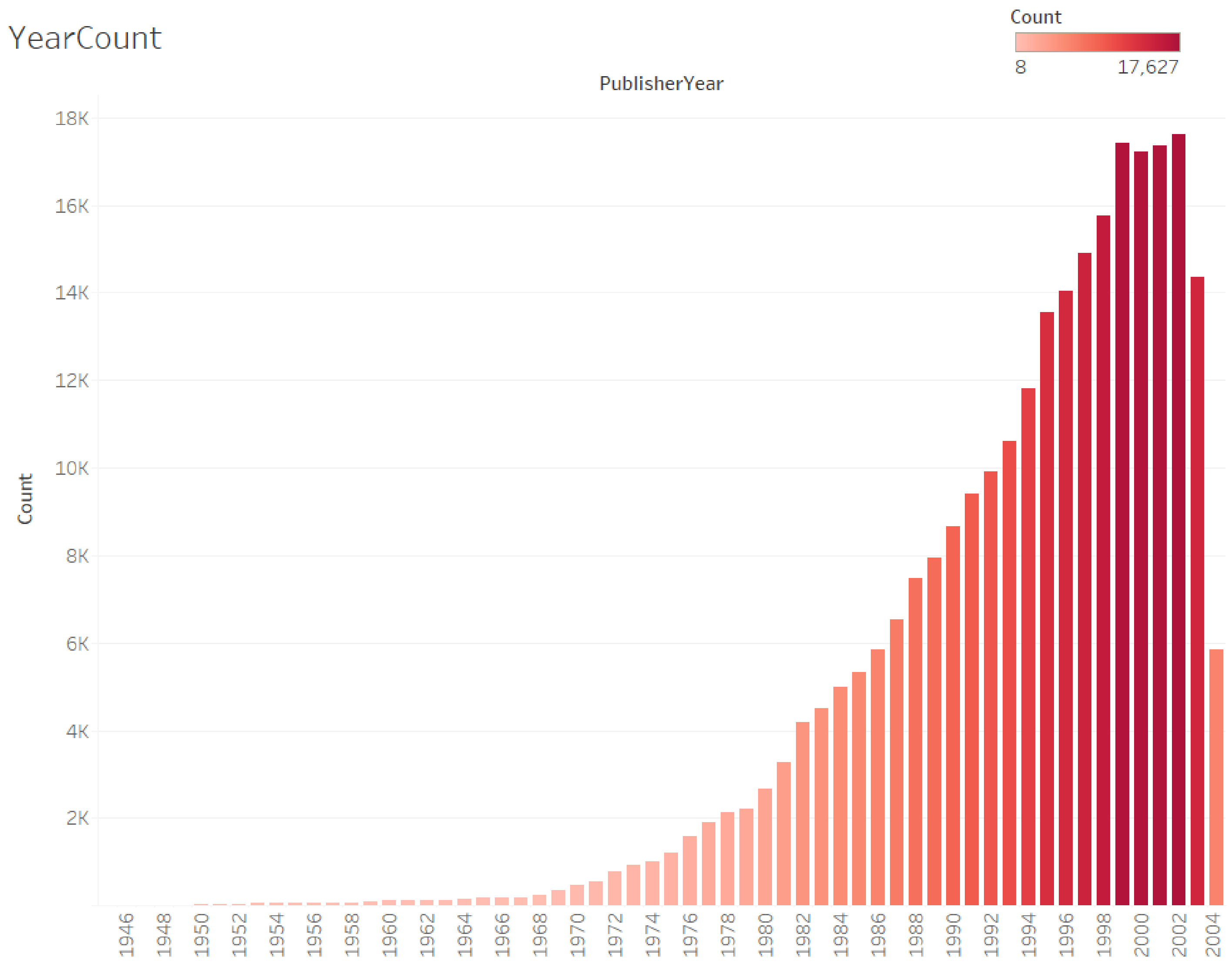


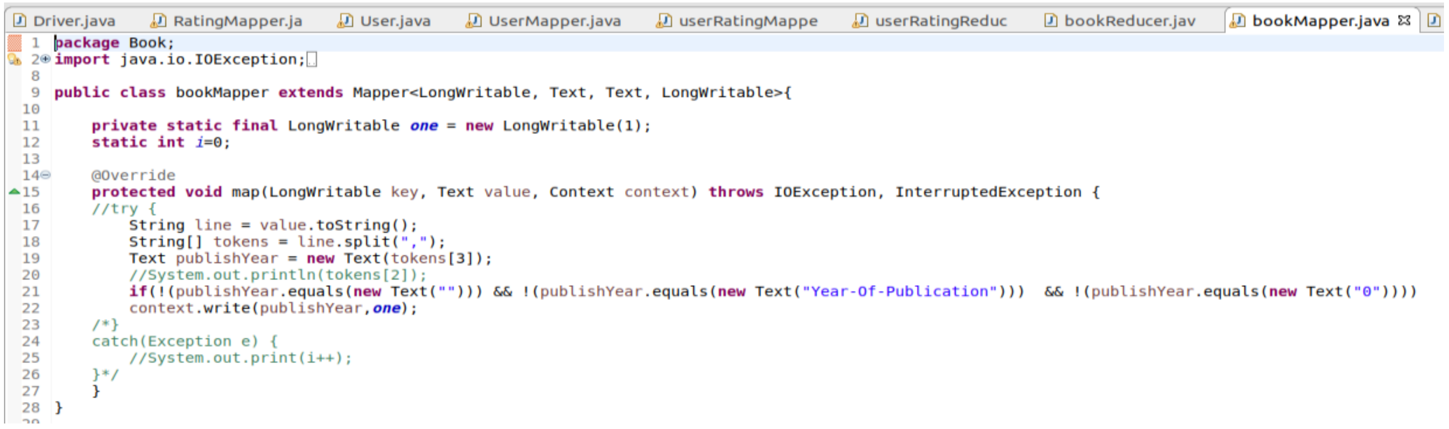
Driver.java   BinningMapper.java   BookRatingMapper.java   topTenMapper.java   topTenReducer.java   Driver.jav

```java
    public int run(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = new Job(conf);

        job.setJarByClass(Driver.class);

        job.setMapperClass(userRatingMapper.class);
        job.setReducerClass(userRatingReducer.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(LongWritable.class);

        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);

        TextInputFormat.addInputPath(job, new Path(args[0]));
        TextOutputFormat.setOutputPath(job, new Path(args[1]+"/temp"));

        job.waitForCompletion(true);

        Job job2 = new Job(conf);

        job2.setJarByClass(Driver.class);

        job2.setMapperClass(IdentityMapper.class);
        job2.setReducerClass(TopContributors.class);

        job2.setMapOutputKeyClass(NullWritable.class);
        job2.setMapOutputValueClass(Text.class);


        job2.setOutputKeyClass(Text.class);
        job2.setOutputValueClass(LongWritable.class);

        job2.setInputFormatClass(TextInputFormat.class);
        job2.setOutputFormatClass(TextOutputFormat.class);

        TextInputFormat.addInputPath(job2, new Path(args[1]+"/temp"));
        TextOutputFormat.setOutputPath(job2, new Path(args[1]+"/final"));

        job2.waitForCompletion(true);


        Job job3 = new Job(conf);

        job3.setJarByClass(Driver.class);

        MultipleInputs.addInputPath(job3, new Path(args[2]), TextInputFormat.class, UserMapper.class);
        MultipleInputs.addInputPath(job3, new Path(args[1]+"/final"), TextInputFormat.class, RatingMapper.class);

        job3.setReducerClass(JoinReducer.class);

        job3.setMapOutputKeyClass(LongWritable.class);
        job3.setMapOutputValueClass(User.class);

        job3.setOutputKeyClass(Text.class);
        //job3.setOutputValueClass(LongWritable.class);
        job3.setOutputValueClass(Text.class);
```

5. Publications per Year:

- The graph displays the curve in the increase in the number of publications of book by year.
- It makes it easy to study the rate of growth.
- Discarded the outlier value where the year was less than 1945 and greater than 2005 where the value was too sparse.



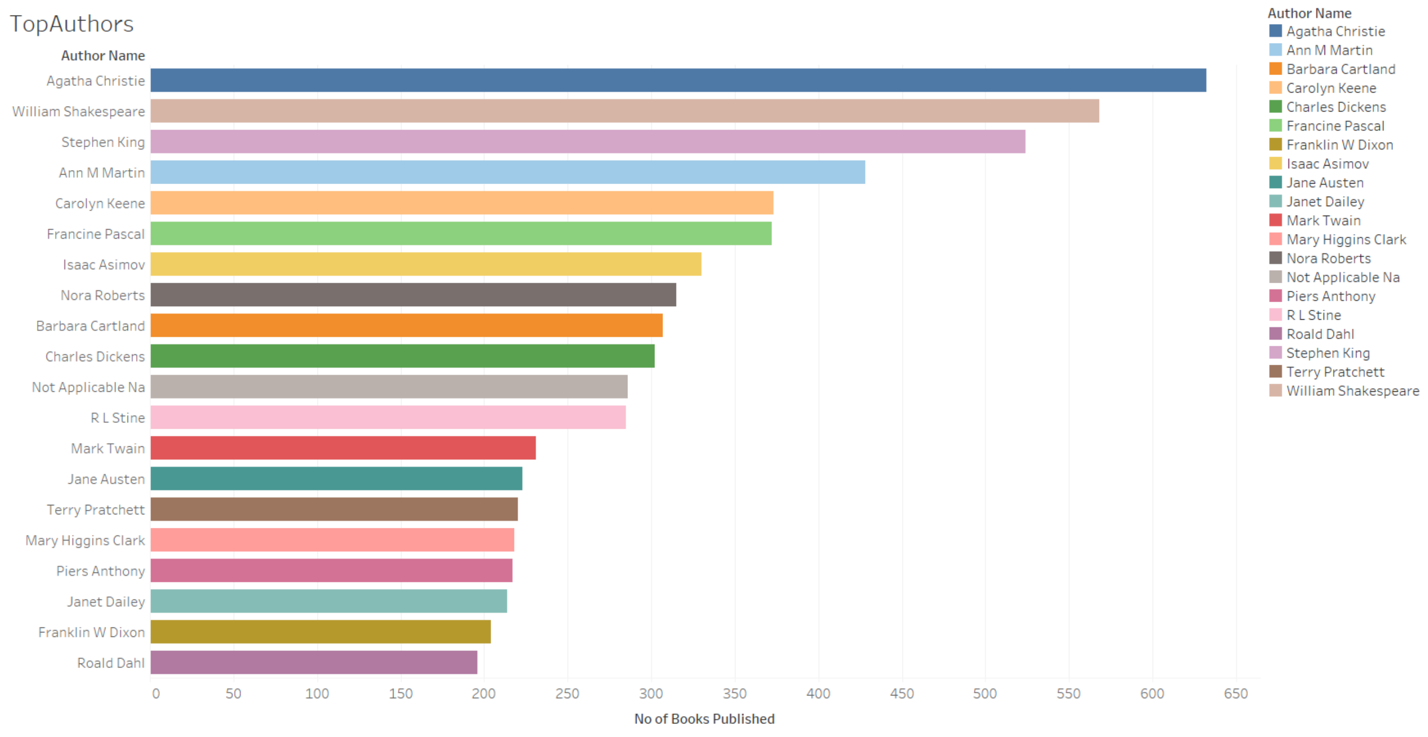


```
package Book;
import java.io.IOException;

public class bookMapper extends Mapper<LongWritable, Text, Text, LongWritable>{

    private static final LongWritable one = new LongWritable(1);
    static int i=0;

    @Override
    protected void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
    //try {
        String line = value.toString();
        String[] tokens = line.split(",");
        Text publishYear = new Text(tokens[3]);
        //System.out.println(tokens[2]);
        if(!(publishYear.equals(new Text(""))) && !(publishYear.equals(new Text("Year-Of-Publication")))  && !(publishYear.equals(new Text("0"))))
        context.write(publishYear,one);
    /*}
    catch(Exception e) {
        //System.out.print(i++);
    }*/
    }
}
```

PIG SCRIPTS:

1. TOP 25 AUTHORS

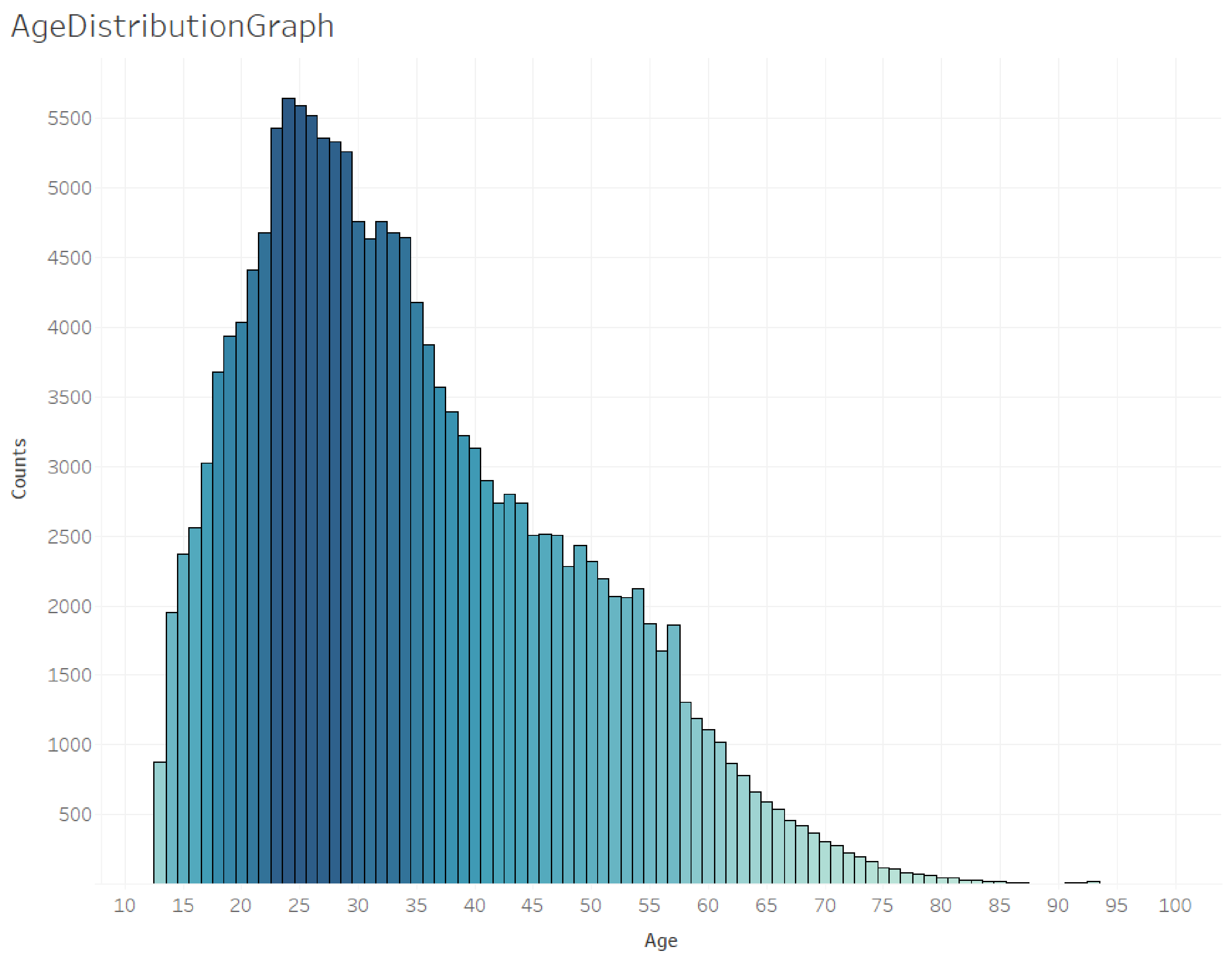
The graph shows the names and comparison between the top 25 authors based on the number of books they have published.



```
BookRatings = LOAD 'input/BX-Book-Ratings.csv' USING PigStorage(',') as (userid:chararray, isbn:int, rating:float);

Books = LOAD 'input/BX-Books.csv' USING PigStorage(',') as (isbn,title,author,year,publisher);

booksfilter = FILTER BookRatings by isbn is not null;

isbn_group = GROUP booksfilter by (isbn);

isbn_count = FOREACH isbn_group GENERATE group, AVG(booksfilter.rating) as rating;

isbnDistinct = DISTINCT isbn_count;

joined = JOIN isbnDistinct by group, Books by isbn;

sorted = order joined by rating desc;

top_25 = limit sorted 25;

myOutput = FOREACH top_25 GENERATE $0, $1, $3;

store myOutput into 'finalOutput5';
```

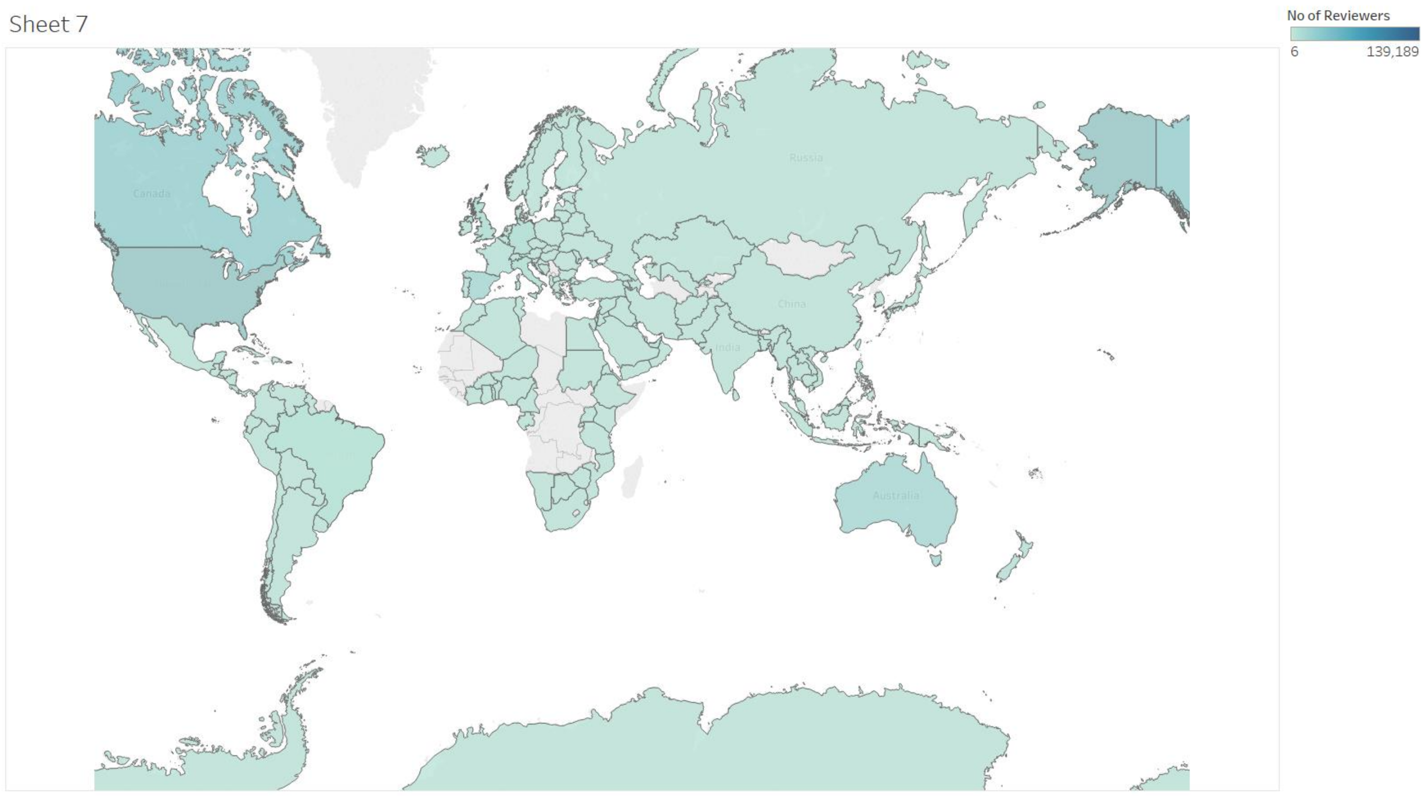
2. USER AGE DISTRIBUTION

The graph below shows the age distribution of the users who gave the reviews.



```
UserAge_temp = LOAD 'input/BX-Users.csv' USING PigStorage(',') as (userid:chararray, area:chararray, city:chararray, country:chararray, age:int);

UserAge = FOREACH UserAge_temp GENERATE userid, REPLACE(country,'\\"',''),age;

UserFilter = FILTER UserAge BY (age is not null) AND (age < 100) AND (age > 12);

AgeGroup = GROUP UserFilter by (age);

AgeCount = FOREACH AgeGroup GENERATE group, COUNT(UserFilter) as counts;

SortedByAge = ORDER AgeCount BY group;

STORE SortedByAge into 'AgeOutput2';
```

3. USER DISTRIBUTION BY COUNTRY

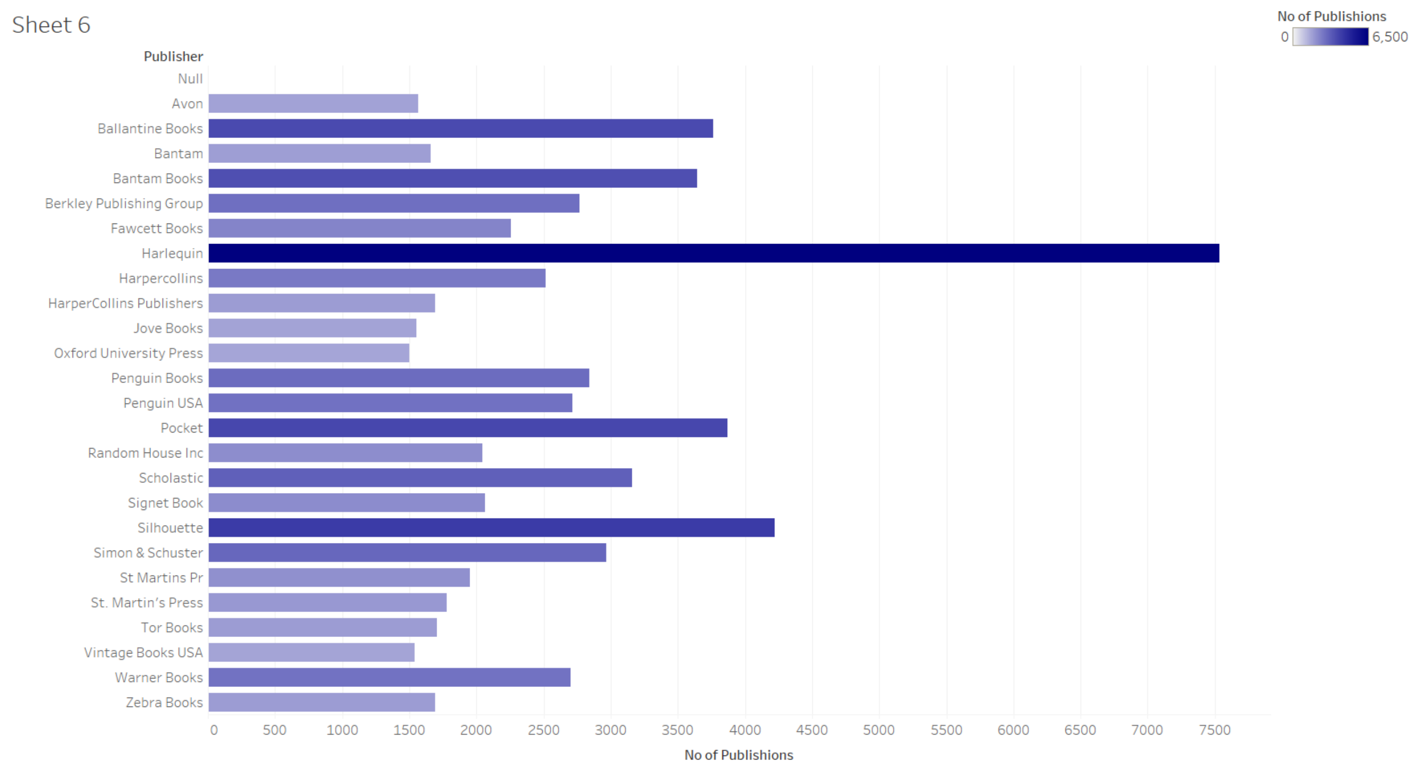


```
UserAge_temp = LOAD 'input/BX-Users.csv' USING PigStorage(',') as (userid:chararray, area:chararray, city:chararray, country:chararray, age:int);

UserAge = FOREACH UserAge_temp GENERATE userid, REPLACE(country,'\\"',''),age;

UserFilter = FILTER UserAge BY (age is not null) AND (age < 100) AND (age > 12);

AgeGroup = GROUP UserFilter by (age);

AgeCount = FOREACH AgeGroup GENERATE group, COUNT(UserFilter) as counts;

SortedByAge = ORDER AgeCount BY group;

STORE SortedByAge into 'AgeOutput2';
```

HIVE SCRIPT:

1. TOP 25 PUBLISHERS

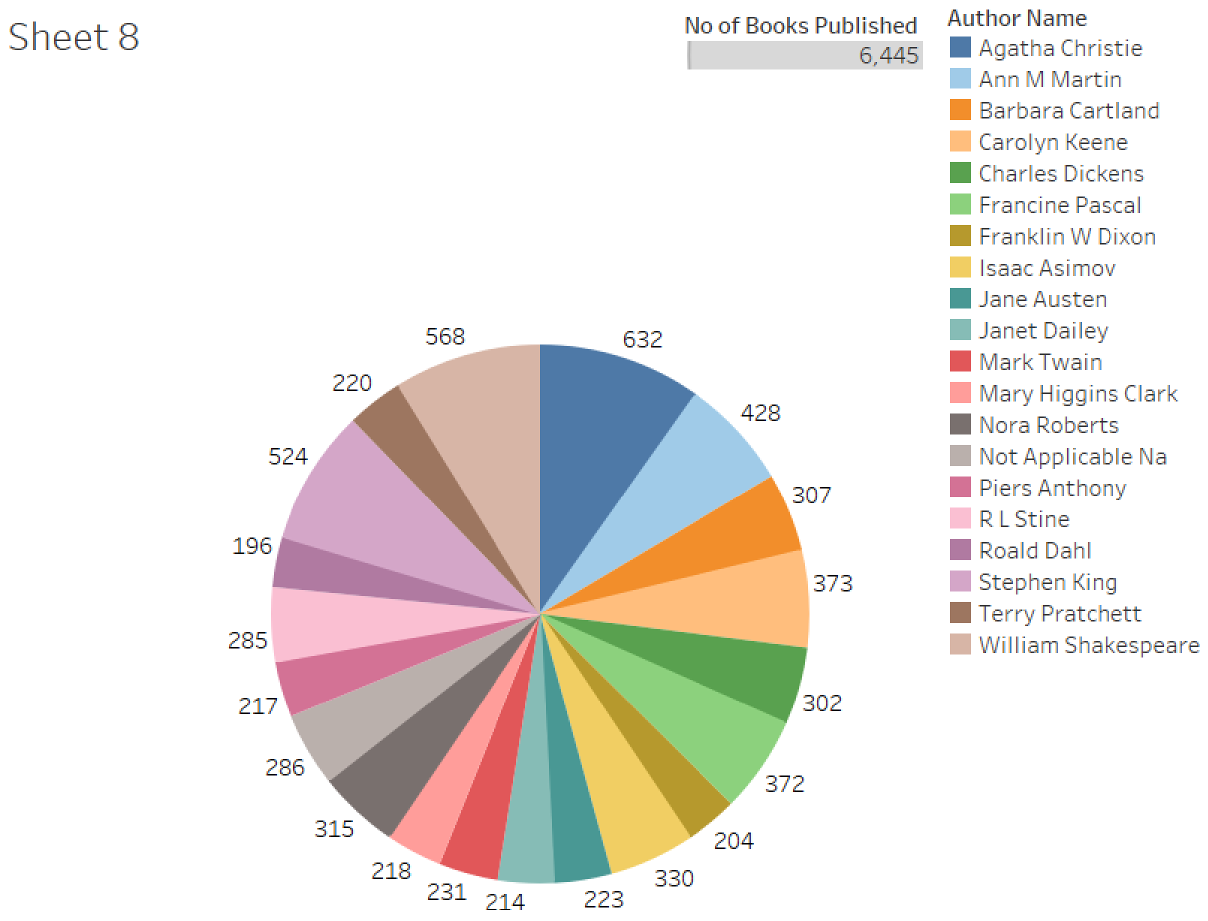
- Graph shows top 25 publishers and their number of publications.

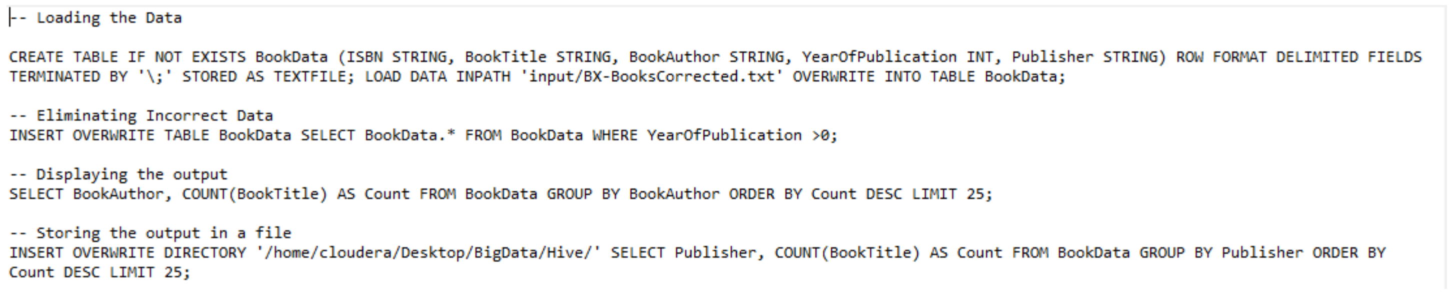


```
-- Loading the Data

CREATE TABLE IF NOT EXISTS BookData (ISBN STRING, BookTitle STRING, BookAuthor STRING, YearOfPublication INT, Publisher STRING) ROW FORMAT DELIMITED FIELDS
TERMINATED BY '\;' STORED AS TEXTFILE; LOAD DATA INPATH 'input/BX-BooksCorrected.txt' OVERWRITE INTO TABLE BookData;

-- Eliminating Incorrect Data
INSERT OVERWRITE TABLE BookData SELECT BookData.* FROM BookData WHERE YearOfPublication >0;

-- Displaying the output
SELECT Publisher, COUNT(BookTitle) AS Count FROM BookData GROUP BY Publisher ORDER BY Count DESC LIMIT 25;

-- Storing the output in a file
--INSERT OVERWRITE DIRECTORY '/home/cloudera/Desktop/BigData/Hive/Output/output.txt' SELECT Publisher, COUNT(BookTitle) AS Count FROM BookData GROUP BY
Publisher ORDER BY Count DESC LIMIT 25;
```

2. TOP 25 AUTHORS

- Hive implementation of the previous use case.





```
-- Loading the Data

CREATE TABLE IF NOT EXISTS BookData (ISBN STRING, BookTitle STRING, BookAuthor STRING, YearOfPublication INT, Publisher STRING) ROW FORMAT DELIMITED FIELDS
TERMINATED BY '\;' STORED AS TEXTFILE; LOAD DATA INPATH 'input/BX-BooksCorrected.txt' OVERWRITE INTO TABLE BookData;

-- Eliminating Incorrect Data
INSERT OVERWRITE TABLE BookData SELECT BookData.* FROM BookData WHERE YearOfPublication >0;

-- Displaying the output
SELECT BookAuthor, COUNT(BookTitle) AS Count FROM BookData GROUP BY BookAuthor ORDER BY Count DESC LIMIT 25;

-- Storing the output in a file
INSERT OVERWRITE DIRECTORY '/home/cloudera/Desktop/BigData/Hive/' SELECT Publisher, COUNT(BookTitle) AS Count FROM BookData GROUP BY Publisher ORDER BY
Count DESC LIMIT 25;
```