Book Crossing Dataset

It is book crossing data having all the details of books published in each year.

Understanding Data

The data format is comma separated values.

It contains 8 columns made up of following:
ISBN
Book-Title
Book-Author
Year-Of-Publication
Publisher
Image-URL-S
Image-URL-M
Image-URL-L

1) Load the Book crossing dataset file

```
grunt> load_bookdata = LOAD
'file:///home/cloudera/khasimbabu/PIG_Excercise/CaseStudy2/Book_Crossing_Dataset.txt' USING
PigStorage(',') as
(ISBN:chararray,BookTitle:chararray,BookAuthor:chararray,YearOfPub:int,Publisher:chararray,image
_url_s:chararray,image_url_m:chararray,image_url_l:chararray);
```

2) Generate only ISBN, BookTitle, BookAuthor, YearOfPublication, Publisher fields/columns

```
grunt> books = FOREACH load_bookdata GENERATE ISBN, BookTitle, BookAuthor, YearOfPub,
Publisher;
```

3) Count the number of books published in each year

```
grunt> grp_books_byyear = GROUP books BY YearOfPub;

grunt> describe grp_books_byyear;

grp_books_byyear: {group: int,books: {(ISBN: chararray,BookTitle: chararray,BookAuthor:
chararray,YearOfPub: int,Publisher: chararray)}}

grunt> books_count_byyear = FOREACH grp_books_byyear GENERATE group as YearOfPub,
COUNT($1) as BookCount;
```

4) Split the books_info on the basis of year, So, if the year is greater than equal to 1990 put it in another file and if it is less than 1990 put it in another file

```
grunt> SPLIT load_bookdata into gtyear IF YearOfPub>=1990, ltyear IF YearOfPub<1990;
grunt> dump gtyear;
grunt> dump ltyear;
```

5) Display those records which doesn't have year in it.

```
grunt> books_year_null = FILTER load_bookdata by YearOfPub is null;
```

6) Union gtyear and ltyear
grunt> union_gtyear_ltyear = UNION gtyear,ltyear;

7) Split all the Publisher column data into individual words
grunt> publisher_token = FOREACH load_bookdata generate TOKENIZE(Publisher);

grunt> describe publisher_token;

publisher_token: {bag_of_tokenTuples_from_Publisher: {tuple_of_tokens: (token: chararray)}}

grunt> dump publisher_token;