

PROJECT REPORT

CSCE 413 MIDTERM

Nasimul Gani

NUID: 13243939

November 23, 2019

Contents

1	Importing the data	2
2	Scrubbing the Data	3
3	Interesting Queries	7
3.1	Insertion	7
3.2	Find	9
3.3	Aggregation	11
3.4	Update	13
3.5	Deletion	13
4	Data Visualization Model	14
4.1	Setting up the chart area	14
4.2	Creating the Axes	15
4.3	The update() Function	17
5	Complete Webpage	19
6	References and Glossary	19

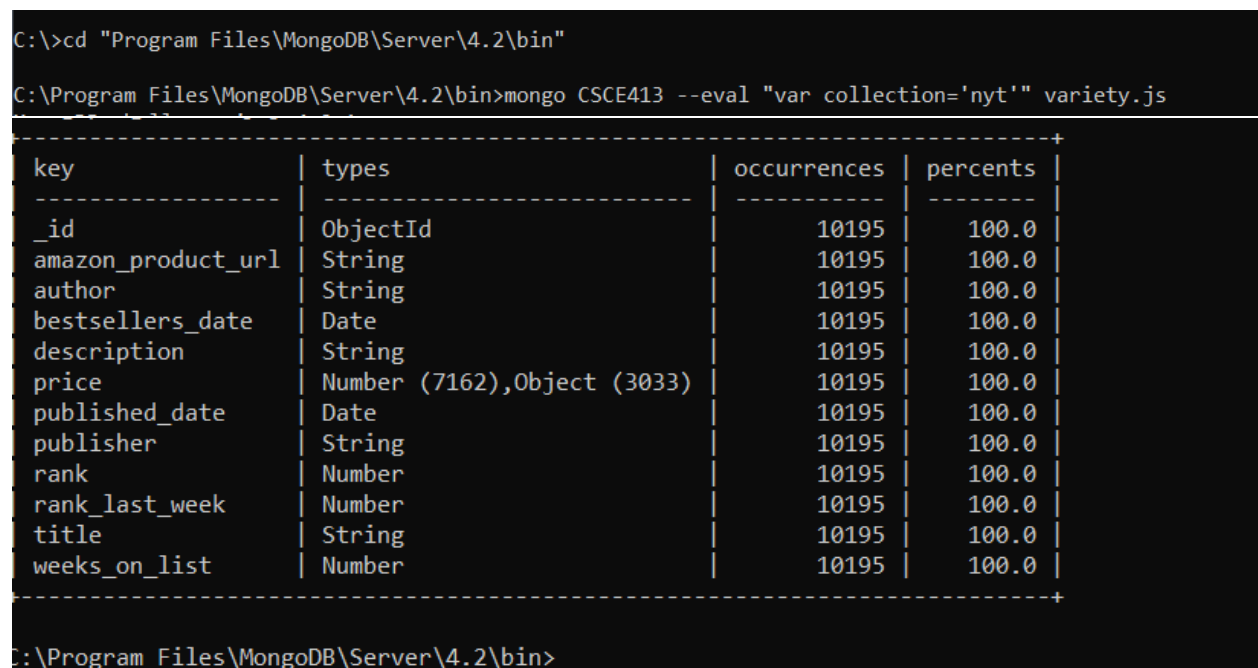
Introduction

The purpose of the project is to visualize a given dataset using D3.js(Data Driven Documents) library that uses SVG(Scalable Vector Graphics) components to render customizable, interactive data visualizations. The dataset for the project called "New York Times Best Sellers from 2008 to 2018" contained data about the rankings of different books in the top 20 Bestsellers list published by New York Times between 2008 and 2018. The list is published every week with updates to the rankings of books from previous weeks and contains details about the books.

The following sections enumerate and explain the steps taken to complete the webpage for the data visualization.

1 Importing the data

The data was first imported to a mongoDB database using Studio3T. An initial summary of the collection using a JavaScript library called Variety.js yielded the result in Figure 1.



```
C:\>cd "Program Files\MongoDB\Server\4.2\bin"
C:\Program Files\MongoDB\Server\4.2\bin>mongo CSCE413 --eval "var collection='nyt'" variety.js
```

key	types	occurrences	percents
_id	ObjectId	10195	100.0
amazon_product_url	String	10195	100.0
author	String	10195	100.0
bestsellers_date	Date	10195	100.0
description	String	10195	100.0
price	Number (7162),Object (3033)	10195	100.0
published_date	Date	10195	100.0
publisher	String	10195	100.0
rank	Number	10195	100.0
rank_last_week	Number	10195	100.0
title	String	10195	100.0
weeks_on_list	Number	10195	100.0

```
C:\Program Files\MongoDB\Server\4.2\bin>
```

Figure 1: Analysis of the collection

This analysis provided the range of values in the dataset along with the names and data types of the keys in a collection.

2 Scrubbing the Data

The original data was properly formatted with an `_id` for each of the documents in the collection. The other key names were also self-explanatory. After a long and intensive search for all the types of charts that could model this collection, the "filter group line chart" model was chosen.

The filter group model allows us to smoothly transition between groups in a line chart with d3.js. In this model, group's values are stored in the same column instead of having one distinct column each. An example of this model can be found at https://www.d3-graph-gallery.com/graph/line_filter.html. Details about this model is explained in section 4.

In lieu of the model that was selected, the original data had to be modified. First, the following query was used to get the books with the most number of appearances in the bestsellers list

```
db.getCollection("nyt").aggregate([
  {
    "$group" : {
      "_id" : {
        "title" : "$title"
      },
      "COUNT(_id)" : {
        "$sum" : NumberInt(1)
      }
    }
  },
  {
    "$project" : {
      "title" : "$_id.title",
      "COUNT(_id)" : "$COUNT(_id)",
      "_id" : NumberInt(0)
    }
  },
  {
    "$sort" : {
      "COUNT(_id)" : NumberInt(-1)
    }
  }
]);
```

which gave the following result:

Result> title	
title	COUNT(_id)
ALL THE LIGHT WE CANNOT SEE	141
THE HELP	114
A DANCE WITH DRAGONS	106
GONE GIRL	87
THE NIGHTINGALE	87
THE GIRL ON THE TRAIN	80
THE GIRL WHO KICKED THE HORNET'S NEST	80
THE HOST	68
THE GOLDFINCH	68
A GENTLEMAN IN MOSCOW	63
THE PARIS WIFE	52
THE HUSBAND'S SECRET	42
BEFORE WE WERE YOURS	42
LITTLE FIRES EVERYWHERE	41
THE STORY OF EDGAR SAWTELLE	40
THE GUERNSEY LITERARY AND POTATO PEEL PIE SOCIETY	40
THE INVENTION OF WINGS	39
GO SET A WATCHMAN	38
THE UNDERGROUND RAILROAD	38
INFERNO	33
THE LOST SYMBOL	29
THE GIRL WHO PLAYED WITH FIRE	29
AND THE MOUNTAINS ECHOED	28
AFTER YOU	28
DEAD AND GONE	28

Figure 2: Query Result

Upon further investigation, the result, however, was incorrect because in the dataset some of the books appeared in the collections but did not have ranking associated with it. So, using the following query the correct number of appearances of each book was computed. The field "weeks_on_list" is a cumulative count of the number of times the book was ranked in the bestsellers list. So, the max of that field for a particular document yields the correct number of times the book appeared in the list.

```
db.getCollection("nyt").aggregate(
```

```

[
  {
    "$group" : {
      "_id" : {
        "title" : "$title"
      },
      "MAX(weeks_on_list)" : {
        "$max" : "$weeks_on_list"
      }
    },
    {
      "$project" : {
        "title" : "$_id.title",
        "MAX(weeks_on_list)" : "$MAX(weeks_on_list)",
        "_id" : NumberInt(0)
      }
    },
    {
      "$sort" : {
        "MAX(weeks_on_list)" : NumberInt(-1)
      }
    }
  ]
);

```

The query gave the following result:

nyt		
title	MAX(weeks_on_list)	
" ALL THE LIGHT WE CANNOT SEE	134	
" THE HELP	108	
" A DANCE WITH DRAGONS	86	
" THE NIGHTINGALE	82	
" GONE GIRL	80	
" THE GIRL WHO KICKED THE HORNET'S NEST	79	
" THE GIRL ON THE TRAIN	79	
" THE GOLDFINCH	60	
" THE HOST	59	
" A GENTLEMAN IN MOSCOW	58	
" BEFORE WE WERE YOURS	42	
" LITTLE FIRES EVERYWHERE	41	
" THE STORY OF EDGAR SAWTELLE	39	
" THE UNDERGROUND RAILROAD	37	
" GO SET A WATCHMAN	35	
" THE GUERNSEY LITERARY AND POTATO PEEL PIE SOCIETY	33	
" THE PARIS WIFE	31	
" INFERNO	29	
" THE LOST SYMBOL	29	
" THE INVENTION OF WINGS	27	
" THE HUSBAND'S SECRET	26	
" AND THE MOUNTAINS ECHOED	24	
" ORIGIN	24	
" THE WOMAN IN THE WINDOW	23	
" DEAD AND GONE	22	

Figure 3: Updated Query Result

Using this query result as a guide, a list of the first 20 book titles was created and then the following query was executed to generate the final data set to be used in the d3.js line chart.

```

db.getCollection("nyt").find({title:{$in:[
"ALL THE LIGHT WE CANNOT SEE","THE HELP","A DANCE WITH DRAGONS",
"THE NIGHTINGALE","GONE GIRL","THE GIRL ON THE TRAIN","THE GIRL WHO KICKED THE
HORNET'S NEST","THE GOLDFINCH","THE HOST","A GENTLEMAN IN MOSCOW","BEFORE WE WERE
YOURS","LITTLE FIRES EVERYWHERE","THE STORY OF EDGAR SAWTELLE","THE UNDERGROUND
RAILROAD","GO SET A WATCHMAN","THE GUERNSEY LITERARY AND POTATO PEEL PIE
SOCIETY","THE PARIS WIFE","THE LOST SYMBOL","INFERNO","THE INVENTION OF
WINGS","THE HUSBAND'S SECRET","ORIGIN","AND THE MOUNTAINS ECHOED","SMALL GREAT
THINGS","DEAD AND GONE","THE WOMAN IN THE WINDOW","THE LAST SONG","THE
WHISTLER","SYCAMORE ROW","THE GIRL WHO PLAYED WITH FIRE","DOCTOR
SLEEP","11/22/63","TWO BY TWO","AFTER YOU","ROGUE LAWYER","GRAY MOUNTAIN"
]}},
{_id:0,bestsellers_date:1, title:1,
rank:1,amazon_product_url:1,author:1,publisher:1,
description:1})
.sort({bestsellers_date:1});

```

The result of the query gave the following documents:

bestsellers_date	amazon_product_url	author	description	publisher	title
2008-05-24T00:00:00.000Z	http://www.amazo	Stephenie Meyer	Aliens have taken c	Little, Brown	THE HOST
2008-05-31T00:00:00.000Z	http://www.amazo	Stephenie Meyer	Aliens have taken c	Little, Brown	THE HOST
2008-06-07T00:00:00.000Z	http://www.amazo	Stephenie Meyer	Aliens have taken c	Little, Brown	THE HOST
2008-06-14T00:00:00.000Z	http://www.amazo	Stephenie Meyer	Aliens have taken c	Little, Brown	THE HOST
2008-06-14T00:00:00.000Z	http://www.amazo	David Wroblewski	A young mute who	Ecco	THE STORY OF EDGAR SAWTELLE
2008-06-21T00:00:00.000Z	http://www.amazo	Stephenie Meyer	Aliens have taken c	Little, Brown	THE HOST
2008-06-21T00:00:00.000Z	http://www.amazo	David Wroblewski	A mute takes refug	Ecco	THE STORY OF EDGAR SAWTELLE
2008-06-28T00:00:00.000Z	http://www.amazo	Stephenie Meyer	Aliens have taken c	Little, Brown	THE HOST
2008-06-28T00:00:00.000Z	http://www.amazo	David Wroblewski	A mute takes refug	Ecco	THE STORY OF EDGAR SAWTELLE
2008-07-05T00:00:00.000Z	http://www.amazo	David Wroblewski	A mute takes refug	Ecco	THE STORY OF EDGAR SAWTELLE
2008-07-05T00:00:00.000Z	http://www.amazo	Stephenie Meyer	Aliens have taken c	Little, Brown	THE HOST
2008-07-12T00:00:00.000Z	http://www.amazo	David Wroblewski	A mute takes refug	Ecco	THE STORY OF EDGAR SAWTELLE
2008-07-12T00:00:00.000Z	http://www.amazo	Stephenie Meyer	Aliens have taken c	Little, Brown	THE HOST
2008-07-19T00:00:00.000Z	http://www.amazo	David Wroblewski	A mute takes refug	Ecco	THE STORY OF EDGAR SAWTELLE
2008-07-19T00:00:00.000Z	http://www.amazo	Stephenie Meyer	Aliens have taken c	Little, Brown	THE HOST
2008-07-26T00:00:00.000Z	http://www.amazo	David Wroblewski	A mute takes refug	Ecco	THE STORY OF EDGAR SAWTELLE
2008-07-26T00:00:00.000Z	http://www.amazo	Stephenie Meyer	Aliens have taken c	Little, Brown	THE HOST
2008-08-02T00:00:00.000Z	http://www.amazo	Stephenie Meyer	Aliens have taken c	Little, Brown	THE HOST
2008-08-02T00:00:00.000Z	http://www.amazo	David Wroblewski	A mute takes refug	Ecco	THE STORY OF EDGAR SAWTELLE
2008-08-02T00:00:00.000Z	http://www.amazo	Mary Ann Shaffer a	After World War II,	Dial	THE GUERNSEY LITERARY AND POTATO PEEL PIE SOCIETY
2008-08-09T00:00:00.000Z	http://www.amazo	Mary Ann Shaffer a	After World War II,	Dial	THE GUERNSEY LITERARY AND POTATO PEEL PIE SOCIETY
2008-08-09T00:00:00.000Z	http://www.amazo	Stephenie Meyer	Aliens have taken c	Little, Brown	THE HOST
2008-08-09T00:00:00.000Z	http://www.amazo	David Wroblewski	A mute takes refug	Ecco	THE STORY OF EDGAR SAWTELLE

Figure 4: Final Query Result

3 Interesting Queries

3.1 Insertion

```

db.getCollection("nyt").insertOne(
{ "bestsellers_date" : ISODate("2019-11-21T00:00:00.000+0000"),
"published_date" : ISODate("2019-10-15T00:00:00.000+0000"),
"amazon_product_url" : "https://www.amazon.com/Blue-Moon-Jack-Reacher-Novel

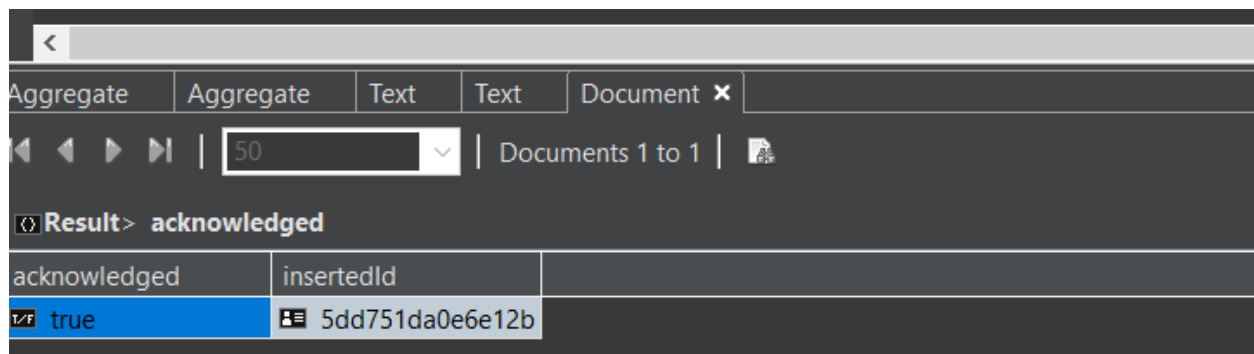
```



```

    /dp/B07QYWMKVJ
    /ref=sr_1_1?keywo
    rds=blue+moon&qid=1574392182&s=audible&sr=1-1",
    "author" : "John Grisham",
    "description" : "In the small north Florida town of Seabrook, a young lawyer named
    Keith Russo was shot dead at his desk as he worked late one night. The killer left
    no clues behind. ",
    "price" : NumberInt(21),
    "publisher" : "Knopf",
    "title" : "THE GUARDIANS",
    "rank" : NumberInt(2),
    "rank_last_week" : NumberInt(0),
    "weeks_on_list" : NumberInt(1) }
)

```



```

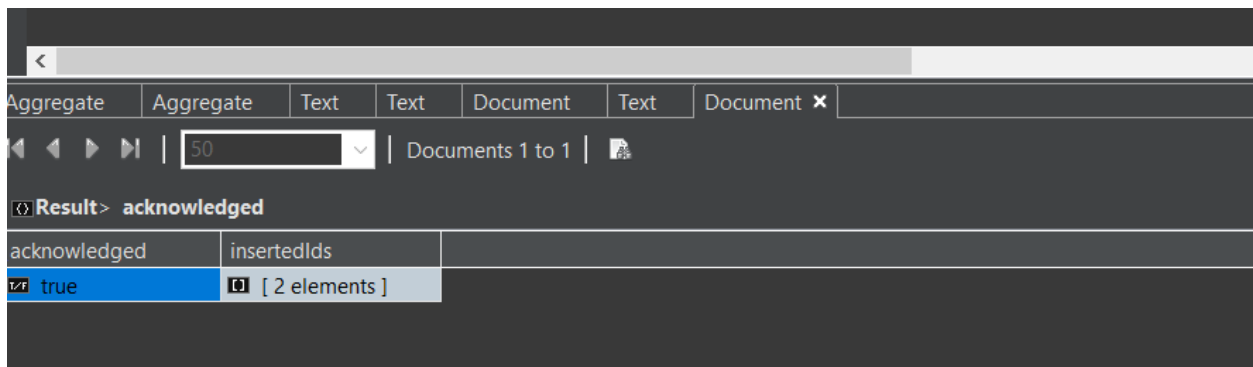
db.getCollection("nyt").insertMany([
  { "bestsellers_date" : ISODate("2019-11-21T00:00:00.000+0000"),
    "published_date" : ISODate("2019-10-15T00:00:00.000+0000"),
    "amazon_product_url" : "https://www.amazon.com/Guardians-Novel-John-Grisham-ebook/dp/
    /B07MYLK9FP/ref=sr_1_1
    ?keywords=the+guardians&qid=1574392657&s=books&sr=1-1",
    "author" : "John Grisham",
    "description" : "In the small north Florida town of Seabrook, a young lawyer named
    Keith Russo was shot dead at his desk as he worked late one night. The killer left
    no clues behind. ",
    "price" : NumberInt(21),
    "publisher" : "Knopf",
    "title" : "THE GUARDIANS",
    "rank" : NumberInt(2),
    "rank_last_week" : NumberInt(0),
    "weeks_on_list" : NumberInt(1) },
  { "bestsellers_date" : ISODate("2019-11-21T00:00:00.000+0000"),
    "published_date" : ISODate("2019-11-05T00:00:00.000+0000"),
    "amazon_product_url" :

```

```

    "https://www.amazon.com/Starless-Sea-Novel-Erin-Morgenstern/dp/038554121X/ref=sr_1_1?crid=3QV8TPYS4JWUC&keywords=the+starless+sea&qid=1574392707&s=books&srefix=the+starless%2Cstripbooks%2C176&sr=1-1",
    "author" : "Erin Morgenstern",
    "description" : "Zachary Ezra Rawlins is a graduate student in Vermont when he discovers a mysterious book hidden in the stacks.",
    "price" : NumberInt(15),
    "publisher" : "Knopf",
    "title" : "THE STARLESS SEA",
    "rank" : NumberInt(3),
    "rank_last_week" : NumberInt(0),
    "weeks_on_list" : NumberInt(1) }
]);

```



3.2 Find

```

db.getCollection("nyt").find(
  {
    "weeks_on_list" : {
      "$gt" : NumberLong(90)
    }
  },
  {
    "title" : "$title",
    "rank" : "$rank",
    "bestsellers_date" : "$bestsellers_date",
    "weeks_on_list" : "$weeks_on_list",
    "_id" : NumberInt(0)
  }
);

```

nyt				
bestsellers_date	title	rank	weeks_on_list	
2010-12-26T00:00:00.000Z	THE HELP	8	91	
2011-01-02T00:00:00.000Z	THE HELP	11	92	
2011-01-09T00:00:00.000Z	THE HELP	5	93	
2011-01-16T00:00:00.000Z	THE HELP	6	94	
2011-01-23T00:00:00.000Z	THE HELP	8	95	
2011-01-30T00:00:00.000Z	THE HELP	5	96	
2011-02-05T00:00:00.000Z	THE HELP	3	97	
2011-02-12T00:00:00.000Z	THE HELP	4	98	
2011-02-19T00:00:00.000Z	THE HELP	4	99	
2011-02-26T00:00:00.000Z	THE HELP	8	100	
2011-03-05T00:00:00.000Z	THE HELP	12	101	
2011-03-12T00:00:00.000Z	THE HELP	14	102	
2011-03-19T00:00:00.000Z	THE HELP	12	103	
2011-08-13T00:00:00.000Z	THE HELP	11	104	
2011-08-20T00:00:00.000Z	THE HELP	4	105	
2011-08-27T00:00:00.000Z	THE HELP	8	106	
2011-09-03T00:00:00.000Z	THE HELP	9	107	
2011-09-10T00:00:00.000Z	THE HELP	13	108	
2016-01-30T00:00:00.000Z	ALL THE LIGHT WE CANNOT SEE	4	91	
2016-02-06T00:00:00.000Z	ALL THE LIGHT WE CANNOT SEE	3	92	
2016-02-13T00:00:00.000Z	ALL THE LIGHT WE CANNOT SEE	5	93	
2016-02-20T00:00:00.000Z	ALL THE LIGHT WE CANNOT SEE	2	94	
2016-02-27T00:00:00.000Z	ALL THE LIGHT WE CANNOT SEE	1	95	

```

db.getCollection("nyt").find(
  {
    "author" : "Lee Child",
    "weeks_on_list" : {
      "$gt" : NumberLong(5)
    }
  },
  {
    "title" : "$title",
    "weeks_on_list" : "$weeks_on_list",
    "_id" : NumberInt(0)
  }
);

```

nyt		
title	weeks_on_list	
61 HOURS	6	
THE AFFAIR	6	
THE AFFAIR	7	
A WANTED MAN	6	
NEVER GO BACK	6	
NEVER GO BACK	7	
PERSONAL	6	
PERSONAL	7	
PERSONAL	8	
PERSONAL	9	
PERSONAL	10	
MAKE ME	6	
MAKE ME	7	
NIGHT SCHOOL	6	
NIGHT SCHOOL	7	
NIGHT SCHOOL	8	
NIGHT SCHOOL	9	
NIGHT SCHOOL	10	
NIGHT SCHOOL	11	
NO MIDDLE NAME	6	
THE MIDNIGHT LINE	6	
THE MIDNIGHT LINE	7	

3.3 Aggregation

```
db.getCollection("nyt").aggregate([
  {
    "$group" : {
      "_id" : {
        "title" : "$title"
      },
      "MAX(weeks_on_list)" : {
        "$max" : "$weeks_on_list"
      }
    }
  },
  {
    "$project" : {
      "title" : "$_id.title",
      "MAX(weeks_on_list)" : "$MAX(weeks_on_list)",
      "_id" : NumberInt(0)
    }
  },
  {
    "$sort" : {
      "MAX(weeks_on_list)" : NumberInt(-1)
    }
  }
]);
```

title	weeks_on_list
61 HOURS	6
THE AFFAIR	6
THE AFFAIR	7
A WANTED MAN	6
NEVER GO BACK	6
NEVER GO BACK	7
PERSONAL	6
PERSONAL	7
PERSONAL	8
PERSONAL	9
PERSONAL	10
MAKE ME	6
MAKE ME	7
NIGHT SCHOOL	6
NIGHT SCHOOL	7
NIGHT SCHOOL	8
NIGHT SCHOOL	9
NIGHT SCHOOL	10
NIGHT SCHOOL	11
NO MIDDLE NAME	6
THE MIDNIGHT LINE	6
THE MIDNIGHT LINE	7

```

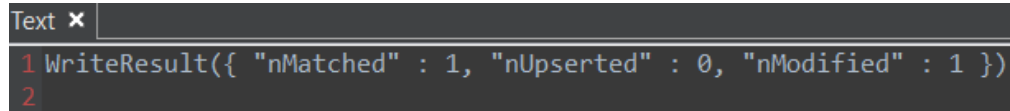
db.getCollection("nyt").aggregate(
[
  {
    "$group" : {
      "_id" : {
        "publisher" : "$publisher"
      },
      "COUNT(_id)" : {
        "$sum" : NumberInt(1)
      }
    }
  },
  {
    "$project" : {
      "publisher" : "$_id.publisher",
      "COUNT(_id)" : "$COUNT(_id)",
      "_id" : NumberInt(0)
    }
  }
],
{
  "allowDiskUse" : true
}
);

```

nyt		
title	weeks_on_list	
61 HOURS	6	
THE AFFAIR	6	
THE AFFAIR	7	
A WANTED MAN	6	
NEVER GO BACK	6	
NEVER GO BACK	7	
PERSONAL	6	
PERSONAL	7	
PERSONAL	8	
PERSONAL	9	
PERSONAL	10	
MAKE ME	6	
MAKE ME	7	
NIGHT SCHOOL	6	
NIGHT SCHOOL	7	
NIGHT SCHOOL	8	
NIGHT SCHOOL	9	
NIGHT SCHOOL	10	
NIGHT SCHOOL	11	
NO MIDDLE NAME	6	
THE MIDNIGHT LINE	6	
THE MIDNIGHT LINE	7	

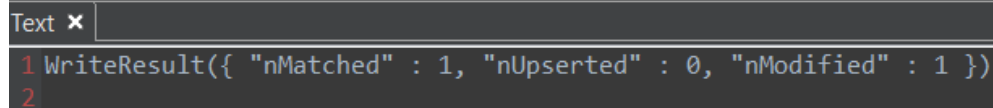
3.4 Update

```
db.getCollection("nyt")
.update({publisher:"Knopf Publishing Group"},
{$set:{publisher:"Knopf"}});
```



The screenshot shows a MongoDB console window with a tab labeled 'Text'. The output of the update operation is displayed as a JSON object: `WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })`. The first line is highlighted in red.

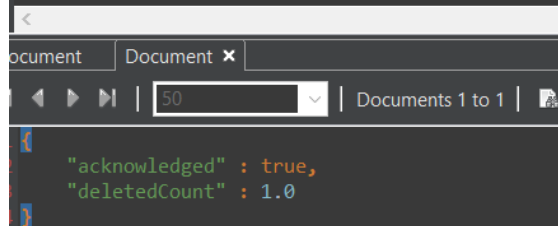
```
db.getCollection("nyt").update(
{tile:"THE HOST"},
{$set:{price:NumberDecimal("18.19")}});
```



The screenshot shows a MongoDB console window with a tab labeled 'Text'. The output of the update operation is displayed as a JSON object: `WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })`. The first line is highlighted in red.

3.5 Deletion

```
db.getCollection("nyt").deleteOne({ "bestsellers_date" : { $lt: ISODate("2008-06-01T00:00:00Z") } } )
```



The screenshot shows a MongoDB console window with a tab labeled 'Document'. The output of the delete operation is displayed as a JSON object: `{ "acknowledged" : true, "deletedCount" : 1 }`. The first line is highlighted in red.

```
1 db.getCollection("nyt").deleteMany({publisher:"ROC"})
```



```
Document x
```

◀ ◀ ▶ ▶ | 50 ▼ | Documents 1 to 1 | 📄

```
1 {
2   "acknowledged" : true,
3   "deletedCount" : 3.0
4 }
5
```

4 Data Visualization Model

The d3.js line chart can be customized to accomodate multiple line charts that change upon changing a selection from the dropdown menu. The final data is used to plot points on the line chart using "rank" as the Y-axis and "bestsellers_date" as the X-axis. This would help visualize the rise and drop in rankings of the 20 books that appeared the most number of times in the bestseller list. The rankings would be a function of the time period during which the book was featured in the list. Hence, each book will have a different time period. This warrants the need for a dynamic X-axis along with a dynamic line chart. The problem was solved by creating an update function which updates the X-axis and the line upon selection of a particular book.

The example from https://www.d3-graph-gallery.com/graph/line_filter.html uses a linear scale for the X-axis which had to be changed to a time scale for the project domain.

4.1 Setting up the chart area

The chart area was created using SVG elements and maipulating the DOM of the webpage using d3.js functions. Here's a snippet:

```
1 // set the dimensions and margins of the graph
2 var margin = { top: 10, right: 30, bottom: 50, left: 60 },
3     width = 1400 - margin.left - margin.right,
4     height = 450 - margin.top - margin.bottom;
5
6 // append the svg object to the body of the page
```

```

7   var svg = d3
8     .select("#my_dataviz")
9     .append("svg")
10    .attr("width", width + margin.left + margin.right)
11    .attr("height", height + margin.top + margin.bottom)
12    .append("g")
13    .attr("transform", "translate(" + margin.left + "," + margin.top + "
    )");

```

Listing 1: Create Chart Area

4.2 Creating the Axes

The CSV(Comma Separated Values) file "final_data.csv" was parsed and then the necessary fields were extracted from the file. These arrays served as the data to update the chart every time the selection was changed. Here's a snippet:

```

1   // A color scale: one color for each group
2   var myColor = d3
3     .scaleOrdinal()
4     .domain(allGroup)
5     .range(d3.schemeSet2);
6
7   // add X axis
8   var x = d3
9     .scaleTime()
10    .domain(
11      d3.extent(data, function(d) {
12        if (d.name == allGroup[0]) {
13          return new Date(d.year);
14        }
15      })
16    )
17    .range([0, width])
18    .nice();
19
20   var x_axis = d3.axisBottom().scale(x);
21
22   svg
23     .append("g")
24     .attr("class", "x axis")
25     .attr("transform", "translate(0," + height + ")")
26     .call(x_axis);
27
28   //axis label
29   svg
30     .append("text")
31     .attr(
32       "transform",
33       "translate(" + width / 2 + " ," + (height + margin.top + 25) + "
    )");

```



```

34     )
35     .style("text-anchor", "middle")
36     .style("stroke", "#ffa500")
37     .text("Dates");
38
39 // Add Y axis
40 var y = d3
41     .scaleLinear()
42     .domain([20, 1])
43     .range([height, 0]);
44
45 svg.append("g").call(
46     d3
47     .axisLeft()
48     .scale(y)
49     .ticks(19)
50 );
51
52 //axis label
53 svg
54     .append("text")
55     .attr("transform", "rotate(-90)")
56     .attr("y", 0 - margin.left)
57     .attr("x", 0 - height / 2)
58     .attr("dy", "1em")
59     .style("text-anchor", "middle")
60     .style("stroke", "#ffa500")
61     .text("Ranks");
62
63 // Initialize line with first group of the list
64 var line = svg
65     .append("g")
66     .append("path")
67     .datum(
68         data.filter(function(d) {
69             return d.name == allGroup[0];
70         })
71     )
72     .attr(
73         "d",
74         d3
75         .line()
76         .x(function(d) {
77             return x(new Date(d.year));
78         })
79         .y(function(d) {
80             return y(+d.n);
81         })
82     )
83     .attr("stroke", function(d) {
84         return myColor("valueA");
85     })
86     .style("stroke-width", 3)

```

```
87 .style("fill", "none");
```

Listing 2: Create the axes

4.3 The update() Function

The update() function was responsible for updating the chart area, axes and the book information at the bottom of the chart. The d3.select() function allows us to select HTML elements on the webpage and update or append new data onto them. Here's a snippet:

```
1
2 // A function that updates the chart
3 function update(selectedGroup) {
4   // Create new data with the selection
5   var dataFilter = data.filter(function(d) {
6     return d.name == selectedGroup;
7   });
8
9   let link = d3.extent(dataFilter, function(d) {
10     return d.link;
11   })[0];
12
13   restInfo = d3.extent(data, function(d) {
14     if (d.name == selectedGroup) {
15       return new Array(d.author, d.publisher, d.desc);
16     }
17   });
18
19   x.domain(
20     d3.extent(dataFilter, function(d) {
21       return new Date(d.year);
22     })
23   );
24
25   svg
26     .select(".x")
27     .transition()
28     .call(x_axis);
29
30   // Give these new data to update line
31   line
32     .datum(dataFilter)
33     .transition()
34     .duration(1000)
35     .attr(
36       "d",
37       d3
38         .line()
39         .x(function(d) {
40           return x(new Date(d.year));
41         })
```

```

42         .y(function(d) {
43             return y(+d.n);
44         })
45     )
46     .attr("stroke", function(d) {
47         return myColor(selectedGroup);
48     });
49
50     linkButton.attr("href", link);
51
52     auth.text("Author: " + restInfo[0][0]);
53
54     pub.text("Publisher: " + restInfo[0][1]);
55
56     desc.text("Short Description: " + restInfo[0][2]);
57 }
58
59 // When the button is changed, run the updateChart function
60 d3.select("#selectButton").on("change", function(d) {
61     // recover the option that has been chosen
62     var selectedOption = d3.select(this).property("value");
63     // run the updateChart function with this selected option
64     update(selectedOption);
65 });
66 });

```

Listing 3: update()

5 Complete Webpage

Below is a screenshot of the complete webpage that displays the chart with the dropdown menu and a link to the book's Amazon page. It also shows some preliminary details about the selected book. The webpage can be found [here](#).

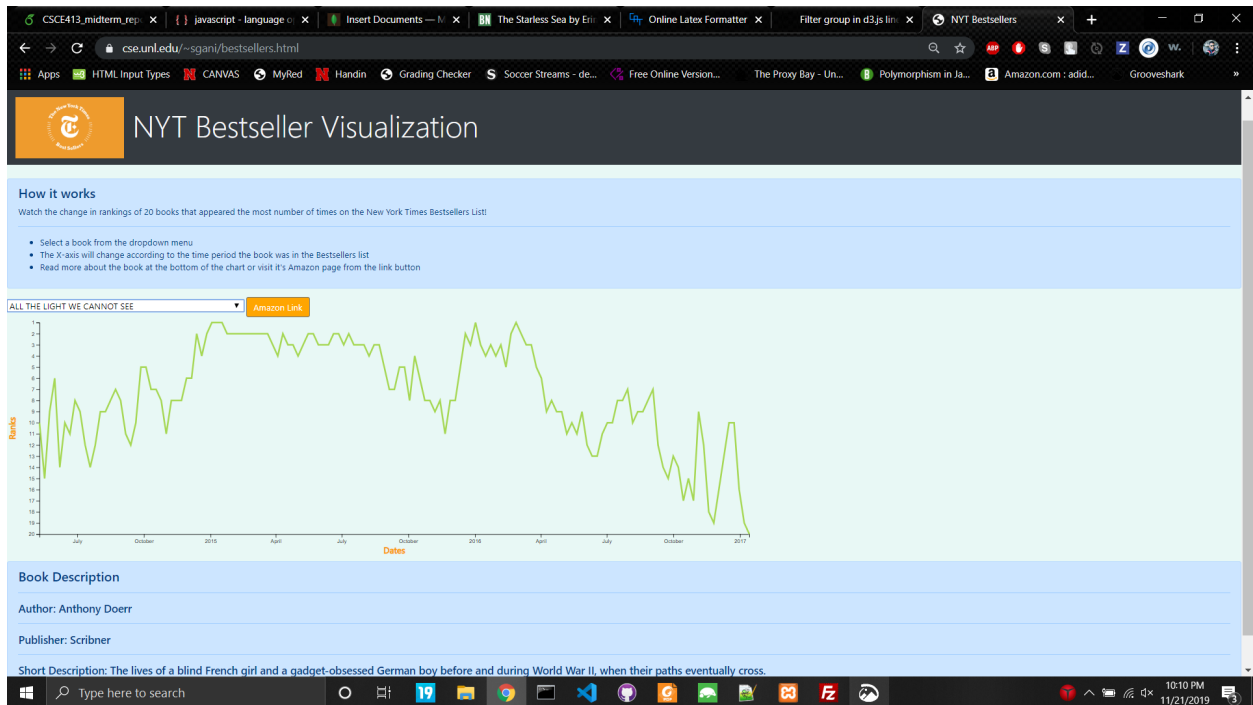


Figure 5: Webpage Screenshot

6 References and Glossary

<https://observablehq.com/@d3/line-chart>

<https://bl.ocks.org/pstuffa/26363646c478b2028d36e7274cedefa6>

https://www.d3-graph-gallery.com/graph/line_filter.html

HTML: Hyper Text Markup Language

DOM: Document Object Model

SVG: Scalable Vector Graphics