D3.JS

DATA-DRIVEN DOCUMENTS

Presented by Georgi Krastev

Code available on GitHub

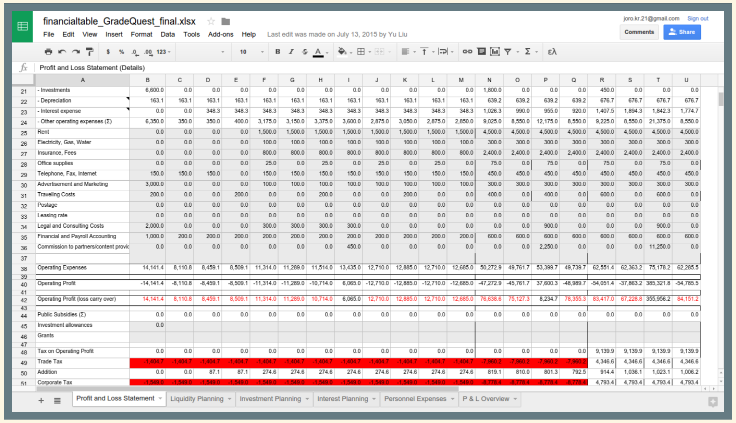
WHAT IS IT?

D3.js is an open-source library for data visualization written in JavaScript and based on SVG (Scalable Vector Graphics).

Disclaimer: Scalable as *"looks good on different screen sizes"*, not as *"supports millions of elements"*

MOTIVATION

It's all about the data. But what does it mean?



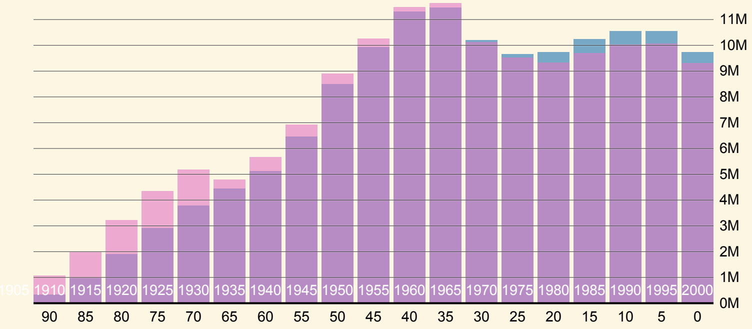
financialtable_GradeQuest_final.xlsx

Last edit was made on July 13, 2015 by Yu Liu

fx Profit and Loss Statement (Details)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
21	- Investments	6,600.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1,800.0	0.0	0.0	0.0	450.0	0.0	0.0	0.0
22	- Depreciation	163.1	163.1	163.1	163.1	163.1	163.1	163.1	163.1	163.1	163.1	163.1	163.1	639.2	639.2	639.2	639.2	676.7	676.7	676.7	676.7
23	- Interest expense	0.0	0.0	348.3	348.3	348.3	348.3	348.3	348.3	348.3	348.3	348.3	348.3	1,026.3	990.0	955.0	920.0	1,407.5	1,894.3	1,842.3	1,774.7
24	- Other operating expenses (Σ)	6,350.0	350.0	350.0	400.0	3,175.0	3,150.0	3,375.0	3,600.0	2,875.0	3,050.0	2,875.0	2,850.0	9,025.0	8,550.0	12,175.0	8,550.0	9,225.0	8,550.0	21,375.0	8,550.0
25	Rent	0.0	0.0	0.0	0.0	1,500.0	1,500.0	1,500.0	1,500.0	1,500.0	1,500.0	1,500.0	1,500.0	4,500.0	4,500.0	4,500.0	4,500.0	4,500.0	4,500.0	4,500.0	4,500.0
26	Electricity, Gas, Water	0.0	0.0	0.0	0.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0
27	Insurance, Fees	0.0	0.0	0.0	0.0	800.0	800.0	800.0	800.0	800.0	800.0	800.0	800.0	2,400.0	2,400.0	2,400.0	2,400.0	2,400.0	2,400.0	2,400.0	2,400.0
28	Office supplies	0.0	0.0	0.0	0.0	25.0	0.0	25.0	0.0	25.0	0.0	25.0	0.0	75.0	0.0	75.0	0.0	75.0	0.0	75.0	0.0
29	Telephone, Fax, Internet	150.0	150.0	150.0	0.0	150.0	150.0	150.0	150.0	150.0	150.0	150.0	150.0	450.0	450.0	450.0	450.0	450.0	450.0	450.0	450.0
30	Advertisement and Marketing	3,000.0	0.0	0.0	0.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0	300.0
31	Traveling Costs	200.0	0.0	0.0	200.0	0.0	0.0	200.0	0.0	0.0	200.0	0.0	0.0	400.0	0.0	400.0	0.0	600.0	0.0	600.0	0.0
32	Postage	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
33	Leasing rate	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
34	Legal and Consulting Costs	2,000.0	0.0	0.0	0.0	300.0	300.0	300.0	300.0	0.0	0.0	0.0	0.0	0.0	0.0	900.0	0.0	0.0	0.0	900.0	0.0
35	Financial and Payroll Accounting	1,000.0	200.0	200.0	200.0	200.0	200.0	200.0	200.0	200.0	200.0	200.0	200.0	600.0	600.0	600.0	600.0	600.0	600.0	600.0	600.0
36	Commission to partners/content provic	0.0	0.0	0.0	0.0	0.0	0.0	0.0	450.0	0.0	0.0	0.0	0.0	0.0	0.0	2,250.0	0.0	0.0	0.0	11,250.0	0.0
37																					
38	Operating Expenses	14,141.4	8,110.8	8,459.1	8,509.1	11,314.0	11,289.0	11,514.0	13,435.0	12,710.0	12,885.0	12,710.0	12,685.0	50,272.9	49,761.7	53,399.7	49,739.7	62,551.4	62,363.2	75,178.2	62,285.5
39																					
40	Operating Profit	-14,141.4	-8,110.8	-8,459.1	-8,509.1	-11,314.0	-11,289.0	-10,714.0	6,065.0	-12,710.0	-12,885.0	-12,710.0	-12,685.0	-47,272.9	-45,761.7	37,600.3	-48,989.7	-54,051.4	-37,863.2	385,321.8	-54,785.5
41																					
42	Operating Profit (loss carry over)	14,141.4	8,110.8	8,459.1	8,509.1	11,314.0	11,289.0	10,714.0	6,065.0	12,710.0	12,885.0	12,710.0	12,685.0	76,638.6	75,127.3	8,234.7	78,355.3	83,417.0	67,228.8	355,956.2	84,151.2
43																					
44	Public Subsidies (Σ)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
45	Investment allowances	0.0																			
46	Grants																				
47																					
48	Tax on Operating Profit	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	9,139.9	9,139.9	9,139.9	9,139.9
49	Trade Tax	-1,404.7	-1,404.7	-1,404.7	-1,404.7	-1,404.7	-1,404.7	-1,404.7	-1,404.7	-1,404.7	-1,404.7	-1,404.7	-1,404.7	-7,960.2	-7,960.2	-7,960.2	-7,960.2	4,346.6	4,346.6	4,346.6	4,346.6
50	Addition	0.0	0.0	87.1	87.1	274.6	274.6	274.6	274.6	274.6	274.6	274.6	274.6	819.1	810.0	801.3	792.5	914.4	1,036.1	1,023.1	1,006.2
51	Corporate Tax	-1,549.0	-1,549.0	-1,549.0	-1,549.0	-1,549.0	-1,549.0	-1,549.0	-1,549.0	-1,549.0	-1,549.0	-1,549.0	-1,549.0	-8,778.4	-8,778.4	-8,778.4	-8,778.4	4,793.4	4,793.4	4,793.4	4,793.4

vs



Credits: mbostock's block #4062085

```
<script id="motivation"/>
```

```
(function() {
var margin = {top: 20, right: 40, bottom: 30, left: 20},
    width = 400 - margin.left - margin.right,
    height = 200 - margin.top - margin.bottom,
    barWidth = Math.floor(width / 19) - 1;

var x = d3.scale.linear()
    .range([barWidth / 2, width - barWidth / 2]);

var y = d3.scale.linear()
    .range([height, 0]);

var yAxis = d3.svg.axis()
    .scale(y)
    .orient("right")
    .tickSize(-width)
    .tickFormat(function(d) { return Math.round(d / 1e6) + "M"; });
```

OUTLINE FOR TODAY

1. Brief overview
2. How to program
3. Examples
4. Discussion

SOME HISTORY

- 2009 - based on the experience of developing and utilizing Prefuse and Flare, Prof. Jeff Heer, Ph.D. Mike Bostock, and M.S. Vadim Ogievetsky of the former Stanford Visualization Group (now the Interactive Data Lab at UW) created Protovis, a JavaScript library to generate SVG graphics from data.
- 2011 - the development of Protovis was stopped to focus on a new project, D3.js that applied the lessons learned from Protovis.

WHERE TO START?

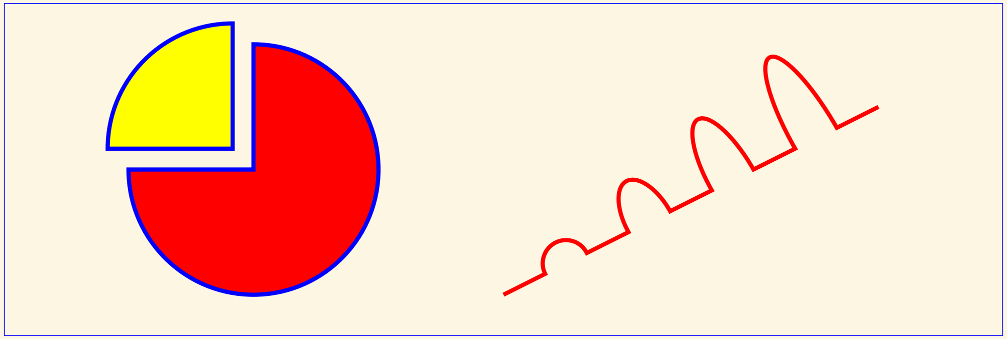
- Documentation - check the wiki
- Learn by example - Mike Bostock's blocks
- 18,139 questions on StackOverflow as of time of writing
- A lot of plugins available
- Reference paper by Bostock, Ogievetsky and Heer
- License: BSD

FIRST - THE BASICS

- SVG
- Selections
- Transitions
- Scales
- ... and then some

SVG (IS XML)

```
<svg width="500" viewBox="0 0 1200 400" xmlns="http://www.w3.org/2000/svg
  <title>Example arcs01 - arc commands in path data</title>
  <desc>Picture of a pie chart with two pie wedges and
        a picture of a line with arc blips</desc>
  <rect x="1" y="1" width="1198" height="398" fill="none" stroke="blue" s

  <path d="M300,200 h-150 a150,150 0 1,0 150,-150 z" fill="red" stroke="b
  <path d="M275,175 v-150 a150,150 0 0,0 -150,150 z" fill="yellow" stroke

  <path d="M600,350 l 50,-25
           a25,25 -30 0,1 50,-25 l 50,-25
           a25,50 -30 0,1 50,-25 l 50,-25
           a25,75 -30 0,1 50,-25 l 50,-25
           a25,100 -30 0,1 50,-25 l 50,-25" fill="none" stroke="red" stro
</svg>
```



D3.SELECT

DOM/SVG manipulation and data binding

```
d3.selectAll("p") // select all <p> elements
    .style("color", "lavender") // set style "color" to "lavender"
    .classed("squares")         // set "class" to "squares"
    .attr("x", 50);             // set attribute "x" to 50px

// create SVG container
var svg = d3.select("#hook").append("svg")
    .attr("width", 640).attr("height", 480)
    .style("fill", "#D0D0D0");

// create SVG elements from data
svg.selectAll("circle") // create virtual circle template
    .data(data)         // bind data
    .enter()            // for each row in the data ...
    .append("circle")   // ... set the attributes accordingly
    .attr("id", function(d) { return d.name })
    .attr("fill",function(d) { return d.color });
```

D3.TRANSITION

Animating style and data changes

defghjklopqrstuvwxy

```
d3.selectAll("p") // select all paragraphs
  .transition("trans-color")   // transition with name "trans-color"
    .delay(0)                  // starting righ after trigger
    .duration(500)             // lasting 500ms
    .ease("linear")            // with linear easing progression ...
    .style("color", "pink");   // ... to color:pink
```

Credits: mbostock's block #3808234

<script id="transitions"/>

```
(function () {
var alphabet = "abcdefghijklmnopqrstuvwxyz".split("");

var width = 800,
    height = 100;

var svg = d3.select("#transition-example").append("svg")
    .attr("width", width)
    .attr("height", height)
  .append("g")
    .attr("transform", "translate(32," + (height / 2) + ")");

function update(data) {

  // DATA JOIN
  // Join new data with old elements, if any.
  var text = svg.selectAll("text")
```

D3.SCALE

Mapping data points (domain) to style properties (range)

- Quantitative
  - `.linear().domain([20,80]).range([0,120]);`
  - `.quantize().domain([0,10]).range([0,3,7]);`
  - `.quantile().domain(dataset).range([0,100]);`
- Ordinal
  - `.ordinal().domain(["R","G","B"]).rangeBands([0,60]);`
  - `.category20();` ●●●●●●●●●●●●●●●●●●●●

... AND MORE

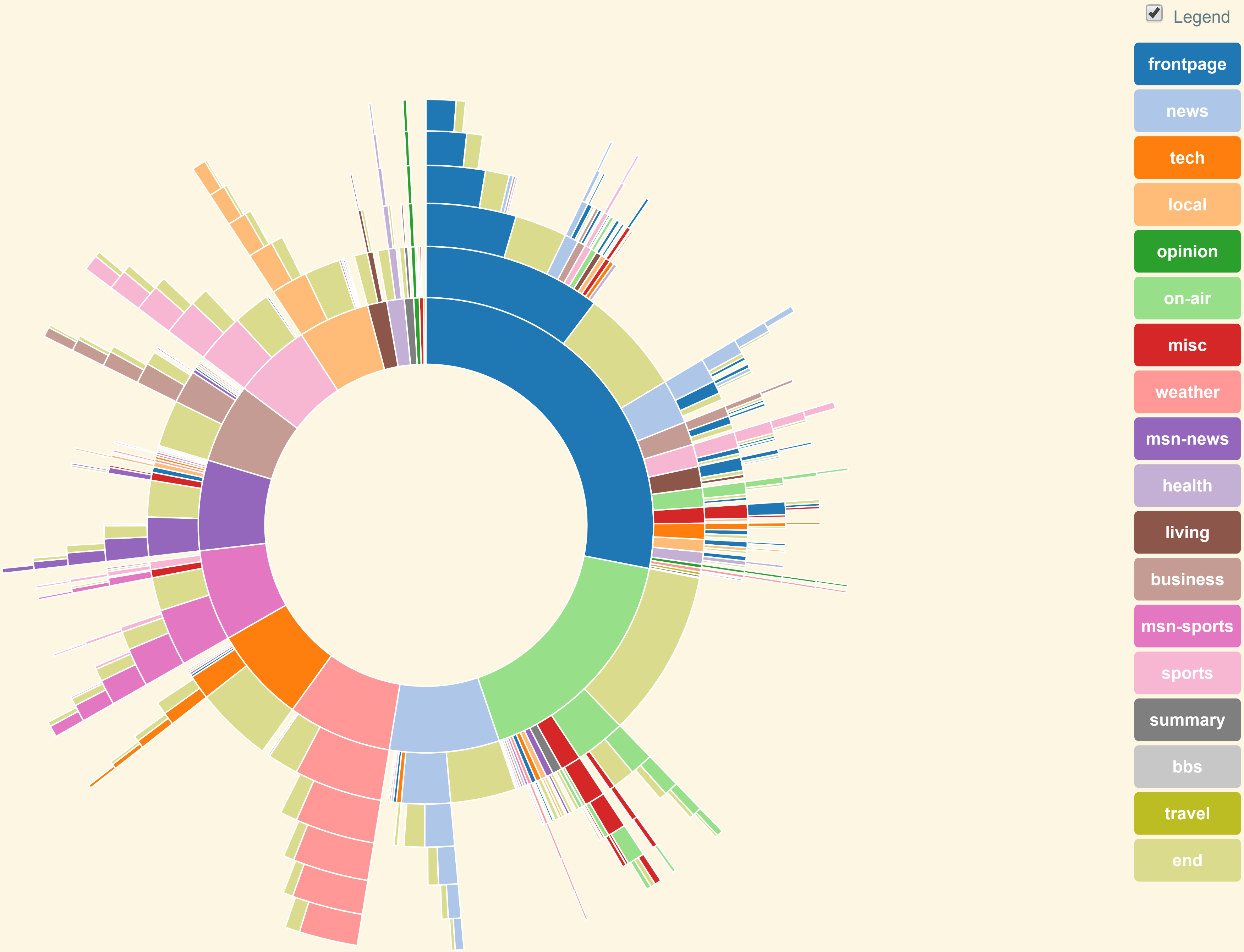
- Support for `.json`, `.csv` and `.tsv` data
- Force-directed graphs
- Integration with various map APIs
- Again, check the wiki

CLICK STREAM ANALYSIS

MSNBC.com anonymous web data set

Data Set Characteristics	Sequential	**Number of Instances**	989818	**Area**	Computer
Attribute Characteristics	Categorical	**Number of Attributes**	N/A	**Date Donated**	N/A
Associated Tasks	N/A	**Missing Values?**	N/A	**Number of Web Hits**	41119

Credits: kerryrodden's block #7090426


☑ Legend
frontpage
news
tech
local
opinion
on-air
misc
weather
msn-news
health
living
business
msn-sports
sports
summary
bbs
travel
end

```
<script id="sunburst"/>
```

```
(function () {
// Dimensions of sunburst.
var width = 750;
var height = 600;
var radius = Math.min(width, height) / 2;

// Breadcrumb dimensions: width, height, spacing, width of tip/tail.
var b = {
  w: 75, h: 30, s: 3, t: 10
};

// Mapping of step names to colors.
var category20 = d3.scale.category20();
var colors = "frontpage news tech local opinion on-air misc weather msn
-news health living business msn-sports sports summary bbs travel end"
    .split(' ').reduce(function(prev, curr, i) { prev[curr] = category2
0(i); return prev; }, {});
```

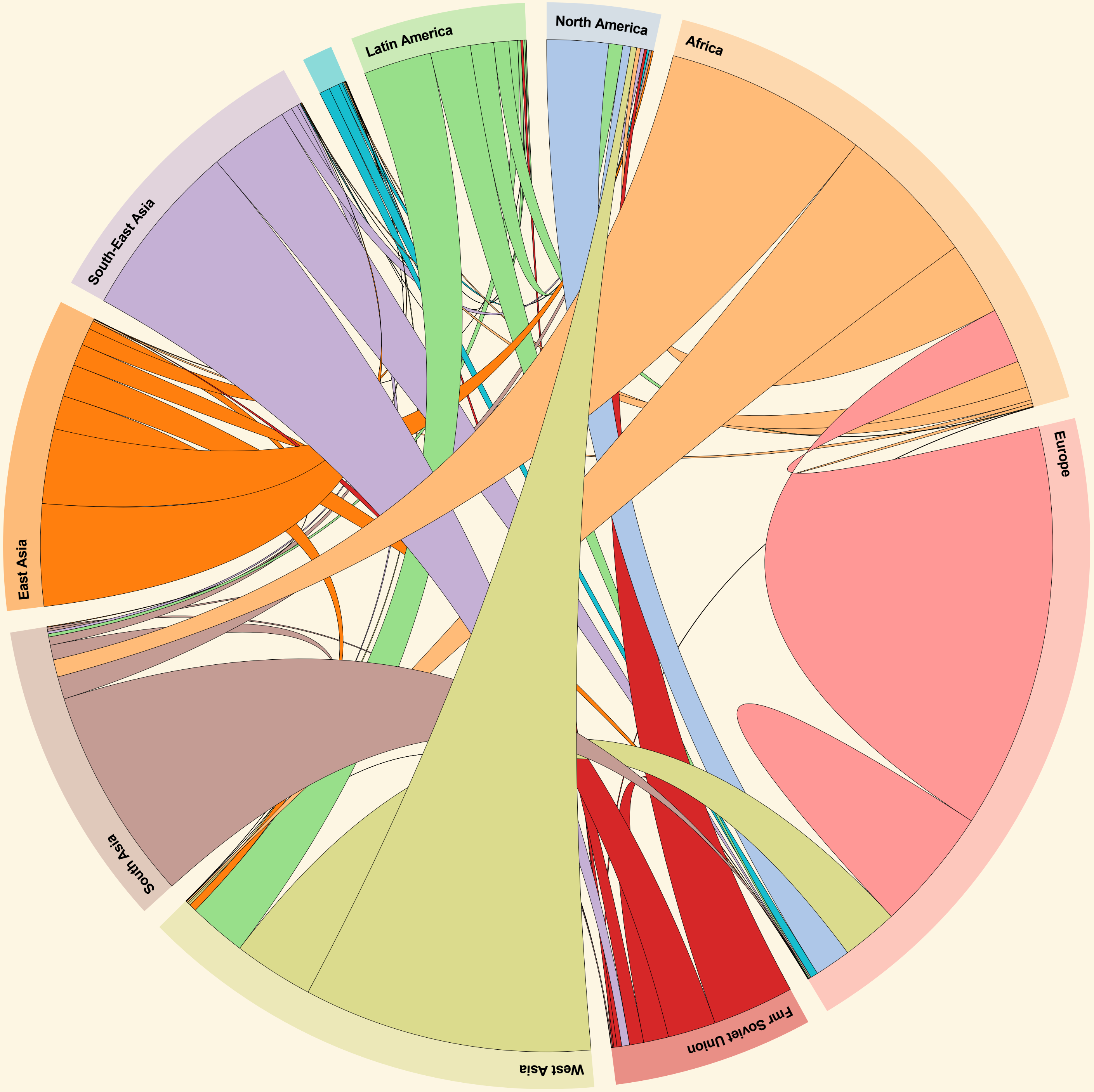
PRE-PROCESSING WITH FLINK

```
object MSNBC extends App {

  // set up the execution environment
  val env = ExecutionEnvironment.getExecutionEnvironment
  env.getConfig.disableSysoutLogging()

  val categories = "frontpage news tech local opinion on-air misc weather
    " business msn-sports sports summary bbs travel end" split ' '

  val sequences = env.readTextFile(
    "/media/georgy/work/Data/msnbc/msnbc990928.seq")

  val clickStream = sequences.map { seq =>
    s"${seq.trim} 18".split(' ')
      .take(6).map(_.toInt - 1)
      .map(categories).mkString(" ") -> 1
  }.groupBy("_1").sum("_2").filter(_._2 > 1)
```

WORLD POPULATION MIGRATION

- Bilateral flows between 196 countries estimated from sequential stock tables
- In the timespan 1990-2010 over 5 year periods
- Provided by the World Bank
- Reference: The Global Flow of People

Credits: mbostock's block uberdata


Latin America
North America
Africa
Europe
Fmr Soviet Union
West Asia
South Asia
East Asia
South-East Asia

<script id="migration"/>

```
(function () {
var width = 720,
    height = 720,
    outerRadius = Math.min(width, height) / 2 - 10,
    innerRadius = outerRadius - 24;

var formatPercent = d3.format(".1%");

var arc = d3.svg.arc()
    .innerRadius(innerRadius)
    .outerRadius(outerRadius);

var layout = d3.layout.chord()
    .padding(.04)
    .sortSubgroups(d3.descending)
    .sortChords(d3.ascending);
```

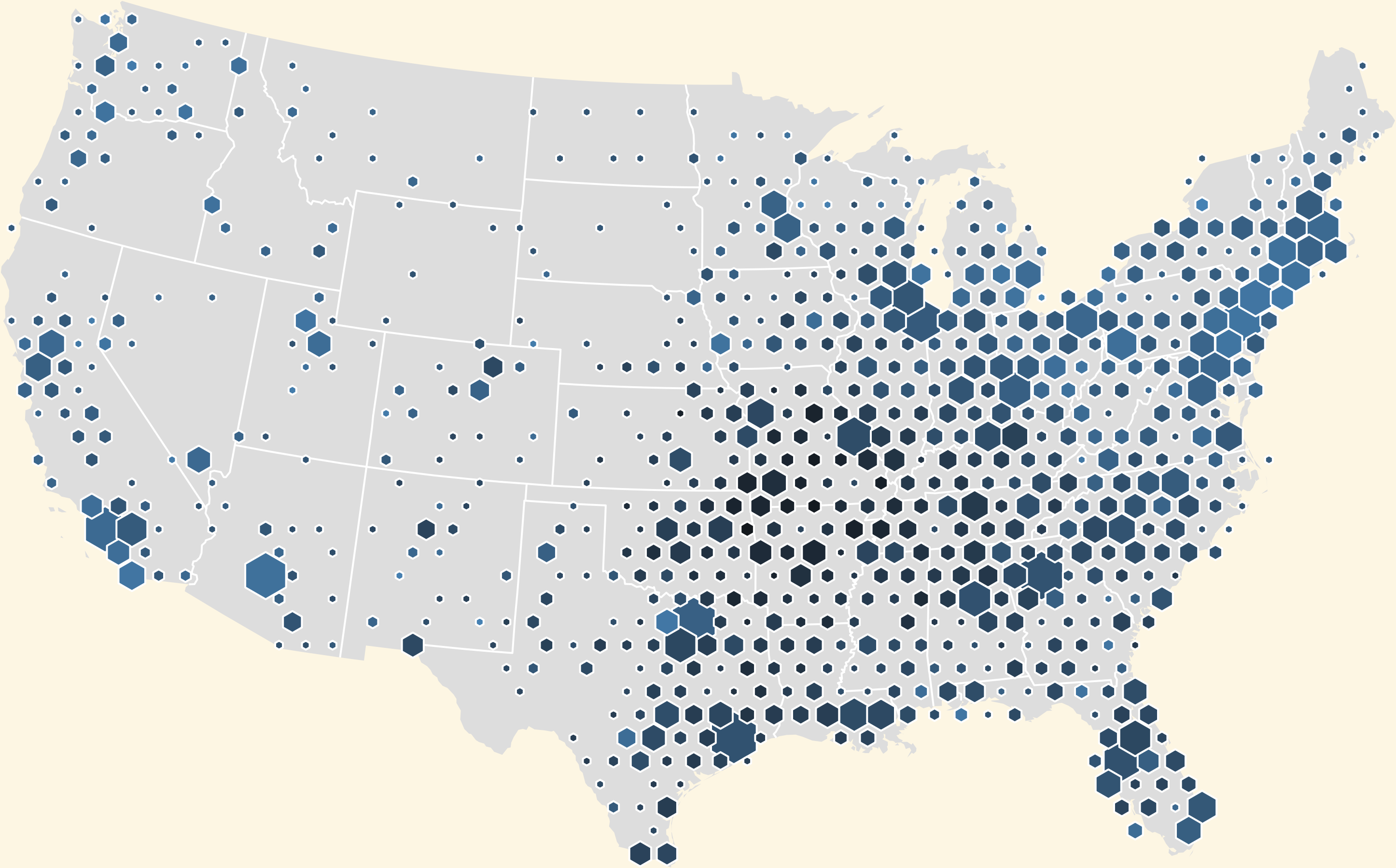
PRE-PROCESSING WITH SCALA

```
object Migration extends App {

  val data = Source.fromFile("/home/georgy/work/edu/d3-reveal/data/migrat
    .getLines().toVector.tail.map { line =>
    val record = line.replaceAll("\"", "").split(',')
    val orig = record(1).toInt - 1
    val dest = record(3).toInt - 1
    val migrants = record.slice(10, 12).map(_.toDouble).sum
    (orig, dest, migrants)
  }.distinct.sorted

  val total = data
    .map(_._3).sum

  val fractions = data
    .map { case (orig, dest, migrants) =>
      (orig, dest, migrants / total)
    }.groupBy(_._1).values
```

WALMART STORES IN THE US

- Geolocation data
- Example of a binning technique
- Size of the bins encodes number of stores
- Color shade encodes the average store age

Credits: mbostock's block #4330486



`<script id="map"/>`

```
(function () {
var width = 960,
    height = 500,
    parseDate = d3.time.format("%x").parse;

var color = d3.time.scale()
     .domain([new Date(1962, 0, 1), new Date(2006, 0, 1)])
     .range(["black", "steelblue"])
     .interpolate(d3.interpolateLab);

var hexbin = d3.hexbin()
     .size([width, height])
     .radius(8);

var radius = d3.scale.sqrt()
     .domain([0, 12])
     .range([0, 8]);
```

DISCUSSION

THE GOOD

- Interactive - unlike many other alternatives
- The best documented library for visualization
- Standard web frontend - SVG, CSS and JavaScript
- A lot of examples available online

DISCUSSION

THE BAD

- Scalability issues - SVG is not hardware-accelerated
- Cannot visualiza streams - update protocol missing
- Frontend programming can be tricky sometimes
- Integration with other data analysis tools still lacking

TAKE AWAY

- Take the time to pre-process your data! Seariously, do it.
- Think out of the box - try out something other than plots and histograms.
- Get your hands dirty - making the graphic work might reveal something more about the data.

THE END

THANK YOU!

ANY QUESTIONS?