Political Landscape Python/D3 Visualizations

A tutorial

By
Ryan Gregg, Harel Kopelman,
Nan Wang, Huanhuan Xu

Why Use Python/D3?

Allows user to build easily-customizable data

  visualizations without proprietary software

    Can display in Python terminal or any web browser

D3 - Written in Javascript (JSON)

D3 - Attach HTML, SVG, CSS3

    Endless customization options

Introduction

The 2016 presidential primaries are underway. Primary results dominate news-cycles and Americans are coming out in droves to vote.

Campaign contributions has attributed to the discourse surrounding this election: who takes money from Wall Street? Who gets it from "real" Americans? How do states contribute and vote?

Another hot topic has been the delegate system- who has the most pledged delegates? The most unpledged?

We answer these questions through data visualization

D3 Steps

Data -> Create Folder for D3 with d3.json, d3.min.js.json ->

Outline

Today we will perform a tutorial outlining how to use D3 to create meaningful visualizations:

1. A bar chart of total contributions to each candidate

2. A packed bubble chart showing the number of Democrat-pledged delegates by state

3. A weighted tree node showing Republican vs. Democrat disposition based off of contributions

4. A United States geographic map showing contribution sums and likelihood of individuals to donate, by state

Data

Original data we utilized contained 15 columns and 1,170,161 rows

We added a "Party" column and deleted 10 columns

Total of 6 columns left

Candidate name, party, contribution amount, contributor zip code, contributor state, contributor occupation

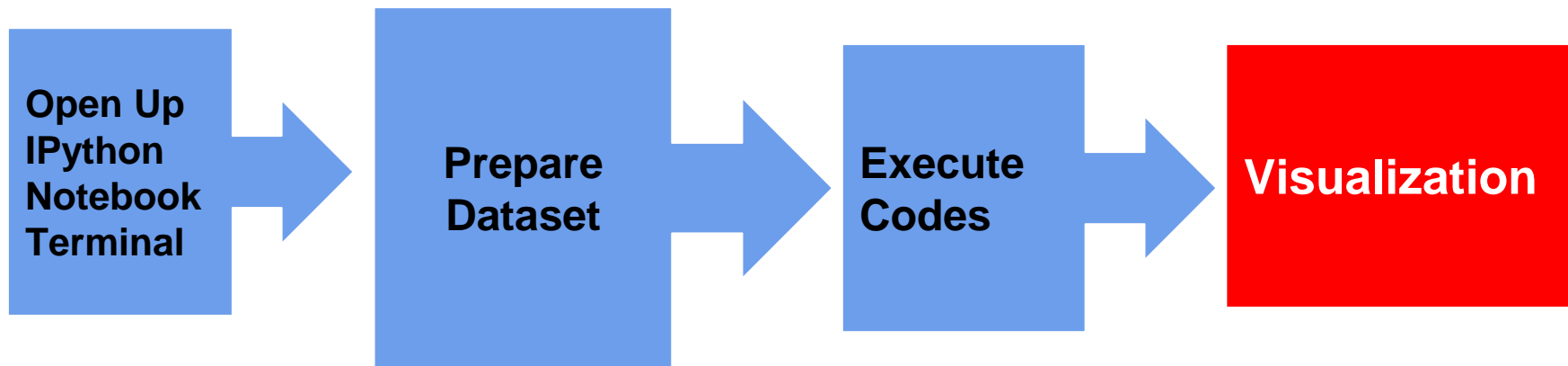# Bar chart - Matplotlib

Bar chart

Created using iPython Notebook

Utilized matplotlib, a Python 2D plotting library

Hillary Clinton, Ted Cruz and Jeb Bush received the most
   money

A large variance of received contributions among candidates
   is found

Bar chart

Bar chart Code

Data Preparation

```python
import pandas as pd
import matplotlib.pyplot as plt
```

```python
df = pd.read_csv("dataoriginal.csv",low_memory=False,index_col = False)
```

```python
df.head()
```

	cmte_id	cand_id	cand_nm	contbr_nm	contbr_city	contbr_st	contbr_zip	contbr_employer	contbr_occupation	contb_receipt_amt	co
0	C00458844	P60006723	Rubio, Marco	HEFFERNAN, MICHAEL	APO	AE	09096	RETIRED	RETIRED	100.0	18
1	C00458844	P60006723	Rubio, Marco	HEFFERNAN, MICHAEL	APO	AE	09096	RETIRED	RETIRED	220.5	15
2	C00458844	P60006723	Rubio, Marco	HEFFERNAN, MICHAEL	APO	AE	09096	RETIRED	RETIRED	225.0	29
3	C00458844	P60006723	Rubio, Marco	CALLESEN JR., RICHARD O.	FPO	AE	096230023	RETIRED	RETIRED	50.0	14
4	C00458844	P60006723	Rubio, Marco	KIBBLE, KUMAR	DPO	AE	092131903	U.S. DEPARTMENT OF HOMELAND SECURITY	LAW ENFORCEMENT	500.0	27

```python
list = df.columns.tolist()
for i,v in enumerate(list):--
    print i,v
```
...

```python
df = df.iloc[:,[2,5,6,8,9]]
df = df.iloc[:,[0,4]]
```

```python
z = df.groupby('cand_nm').sum()
```

```python
z = z.sort('contb_receipt_amt',ascending = False)
```

```
C:\Users\Miya\AppData\Local\Enthought\Canopy\User\lib\site-packages\ipykernel\__main__.py:1: FutureWarning: sort(colu
mns=....) is deprecated, use sort_values(by=.....)
  if __name__ == '__main__':
```
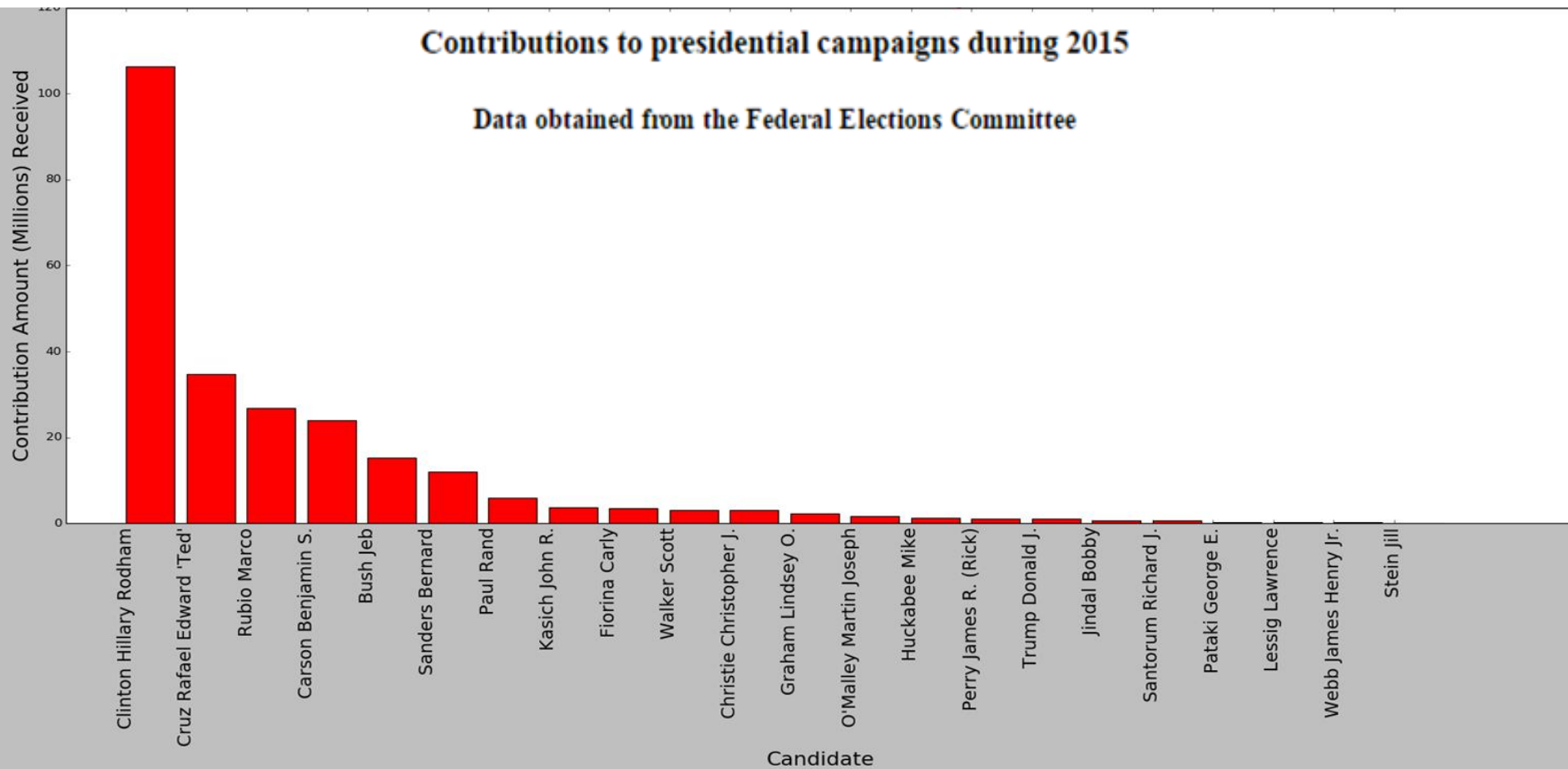
```python
z.head()
```

	contb_receipt_amt
cand_nm	
Clinton, Hillary Rodham	1.062859e+08
Cruz, Rafael Edward 'Ted'	3.482153e+07
Bush, Jeb	3.158145e+07
Rubio, Marco	2.691562e+07
Sanders, Bernard	2.650937e+07

Visualization

```python
plt.bar(x,y,color = 'red')
plt.xticks(x,labels,rotation = 'vertical',fontsize = 18)
plt.subplots_adjust(bottom=0.3)
plt.ylabel('Contribution Amount Received',fontsize = 20)
plt.xlabel('Candidate', fontsize=20)
plt.title('Contribution Amount by Candidate',fontsize = 25,color =
        'red')
plt.show()
```

Most Work in the Data Preparation!

Total Amount of Contributions Received by Each Candidate

Contributions to presidential campaigns during 2015

Data obtained from the Federal Elections Committee

What is a Pledged Delegate?

Elected or chosen at the state or local level, with the understanding that they will support a particular candidate at the convention.

Pledged delegates are, however, not actually bound to vote for that candidate, thus the candidates are allowed to periodically review the list of delegates and eliminate any of those they feel would not be supportive. Currently there are 4,051 pledged delegates.

Packed Bubble Code, data

JSON - DOM defined

```json
{
  "name": "bubbles",
  "children": [
    {
      "name": "States",
      "children": [
        {
          "name": "A",
          "children": [
            {"name": "Alabama", "size": 53},
            {"name": "Arizona", "size": 75},
            {"name": "Arkansas", "size": 32},
            {"name": "Alaska", "size": 16}
          ]
        },
        {
          "name": "C",
          "children": [
            {"name": "California", "size": 475},
            {"name": "Colorado", "size": 66},
            {"name": "Connecticut", "size": 55}
          ]
        },
        {
          "name": "D",
          "children": [
            {"name": "D.C.", "size": 20},
            {"name": "Delaware", "size": 21}
          ]
        }
      ]
    },
    {
      "name": "F-G",
      "children": [
        {"name": "Florida", "size": 214},
        {"name": "Georgia", "size": 102},
        {
          "name": "H-I",
          "children": [
            {"name": "Hawaii", "size": 25},
            {"name": "Idaho", "size": 23},
            {"name": "Illinois", "size": 156},
            {"name": "Indiana", "size": 83},
            {"name": "Iowa", "size": 44}
          ]
        }
```

HTML - svg are defined

```html
<!DOCTYPE html>
<meta charset="utf-8">
<style>

text {
  font: 10px sans-serif;
}

</style>
<body>
<script src="d3.min.js"></script>
<script>

var diameter = 960,
    format = d3.format(",d"),
    color = d3.scale.category20c();

var bubble = d3.layout.pack()
    .sort(null)
    .size([diameter, diameter])
    .padding(1.5);

var svg = d3.select("body").append("svg")
    .attr("width", diameter)
    .attr("height", diameter)
    .attr("class", "bubble");

d3.json("bubble2.json", function(error, root) {
  if (error) throw error;

  var node = svg.selectAll(".node")
      .data(bubble.nodes(classes(root))
      .filter(function(d) { return !d.children; }))
    .enter().append("g")
      .attr("class", "node")
      .attr("transform", function(d) { return "translate(" + d.x + "," + d.y + ")"; });

  node.append("title")
      .text(function(d) { return d.className + ": " + format(d.value); });

  node.append("circle")
      .attr("r", function(d) { return d.r; })
      .style("fill", function(d) { return color(d.packageName); });

  node.append("text")
      .attr("dy", ".3em")
      .style("text-anchor", "middle")
      .text(function(d) { return d.className.substring(0, d.r / 3); });
});
```

Packed Bubble Chart of Democratic Party Pledged Delegates

Tree diagram - d3

Tree diagram

Utilizes D3 and is served onto a web browser

Found top ten occupation contributors

    Measured how many times each had contributed

Mapped Retired, Not Employed and Currently Employed

Tree diagram code, data

Executed HTML

```
.node {
  cursor: pointer;
}

.node circle {
  fill: #fff;
  stroke: steelblue;
  stroke-width: 1.5px;
}

.node text {
  font: 15px sans-serif;
}

.link {
  fill: none;
  stroke: #ccc;
  stroke-width: 1.5px;
}

</style>
<body>
<script type="text/javascript" src="d3.min.js"></script>
<script>

var margin = {top: 20, right: 120, bottom: 20, left: 120},
    width = 960 - margin.right - margin.left,
    height = 800 - margin.top - margin.bottom;

var i = 0,
    duration = 750,
    root;

var tree = d3.layout.tree()
    .size([height, width]);
```
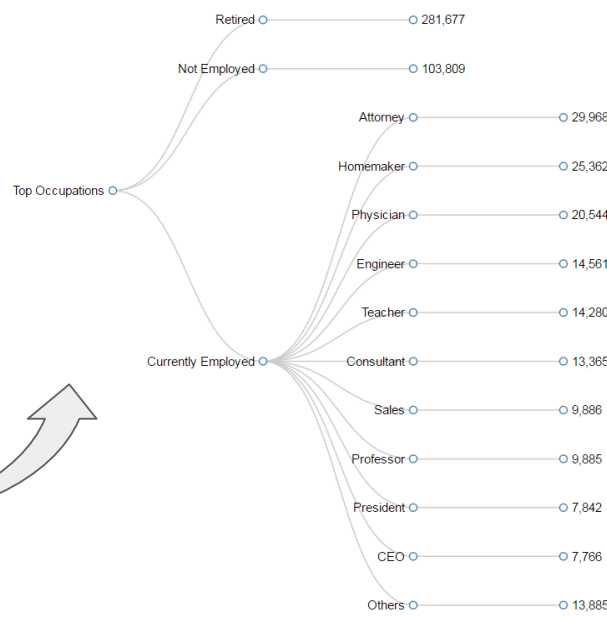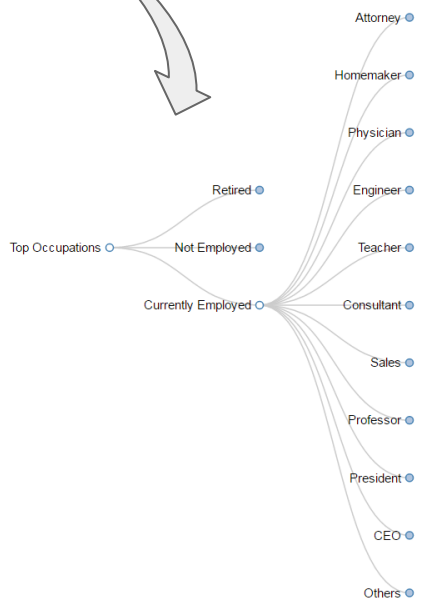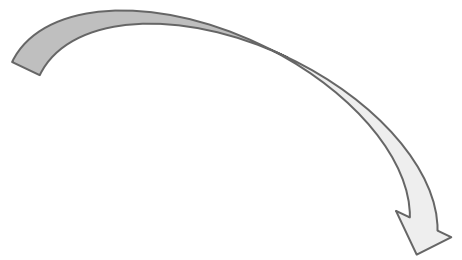
JSON Data

```
{
 "name": "Top Occupations",
 "children": [
  {
   "name": "Retired",
   "children": [
    {
     "name": "281,677"
    }
   ]
  },
  {
   "name": "Not Employed",
   "children": [
    {
     "name": "103,809"
    }
   ]
  },
  {
   "name": "Currently Employed",
   "children": [
    {
     "name": "Attorney",
     "children": [
      {"name": "29,968", "size": 1302}
     ]
    },
```

Top Occupations
- Retired
- Not Employed
- Currently Employed

Top Occupations
- Retired
- Not Employed
- Currently Employed
  - Attorney
  - Homemaker
  - Physician
  - Engineer
  - Teacher
  - Consultant
  - Sales
  - Professor
  - President
  - CEO
  - Others

Top Occupations
- Retired — 281,677
- Not Employed — 103,809
- Currently Employed
  - Attorney — 29,968
  - Homemaker — 25,362
  - Physician — 20,544
  - Engineer — 14,561
  - Teacher — 14,280
  - Consultant — 13,365
  - Sales — 9,886
  - Professor — 9,885
  - President — 7,842
  - CEO — 7,766
  - Others — 13,885

States Maps - d3.geomap

States maps

Utilizes `D3.geomap`, a library written in JavaScript built upon D3

    Executed in web browser, reads CSV only

These types of maps are known as choropleths

    Differences in shading, coloring, or the placing of symbols within predefined areas indicate the property or quantity in those areas

Three different types of States Maps are created:

    Total contributions by state

    Percent of people who contributed in each state

Percentile ranking of each percent of people who contributed by state

States maps code, data

Javascript, CSV

```javascript
var map = d3.geomap.choropleth()
    .geofile('/d3-geomap/topojson/countries/USA.json')
    .projection(d3.geo.albersUsa)
    .column('2012')
    .unitId('fips')
    .scale(1000)
    .legend(true);

d3.csv('/data/venture-capital.csv', function(error, data) {
    d3.select('#map')
        .datum(data)
        .call(map.draw, map);
});
```

Executing HTML

```html
<html>
    <head>
        <meta charset="utf-8">
        <link href="/d3-geomap/css/d3.geomap.css" rel="stylesheet">
        <script src="/d3-geomap/vendor/d3.geomap.dependencies.min.js"></script>
        <script src="/d3-geomap/js/d3.geomap.min.js"></script>
        <script src="/app1.js"></script>
    </head>
    <body>
        <div id="map"></div>
    </body>
</html>
```

```
,State,1998,1999,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,fips
0,Alabama,0.77,0.59,2.3,0.67,0.46,0.23,0.27,0.13,0.12,0.19,0.14,0.26,0.0,0.02,0.13,US01
1,Alaska,0.0,0.0,0.14,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,US02
2,Arizona,1.51,2.14,0.21,1.16,1.11,0.39,0.51,0.06,1.1,0.78,0.8,0.44,0.34,0.89,0.83,US04
3,Arkansas,0.11,0.39,9.15,0.15,0.13,0.01,0.04,1.39,0.42,0.0,0.0,0.0,0.05,0.0,0.05,US05
4,California,7.14,19.14,32.61,12.46,6.87,5.86,5.95,6.52,6.99,7.88,7.51,5.06,5.95,7.92,7.08,US06
...
```
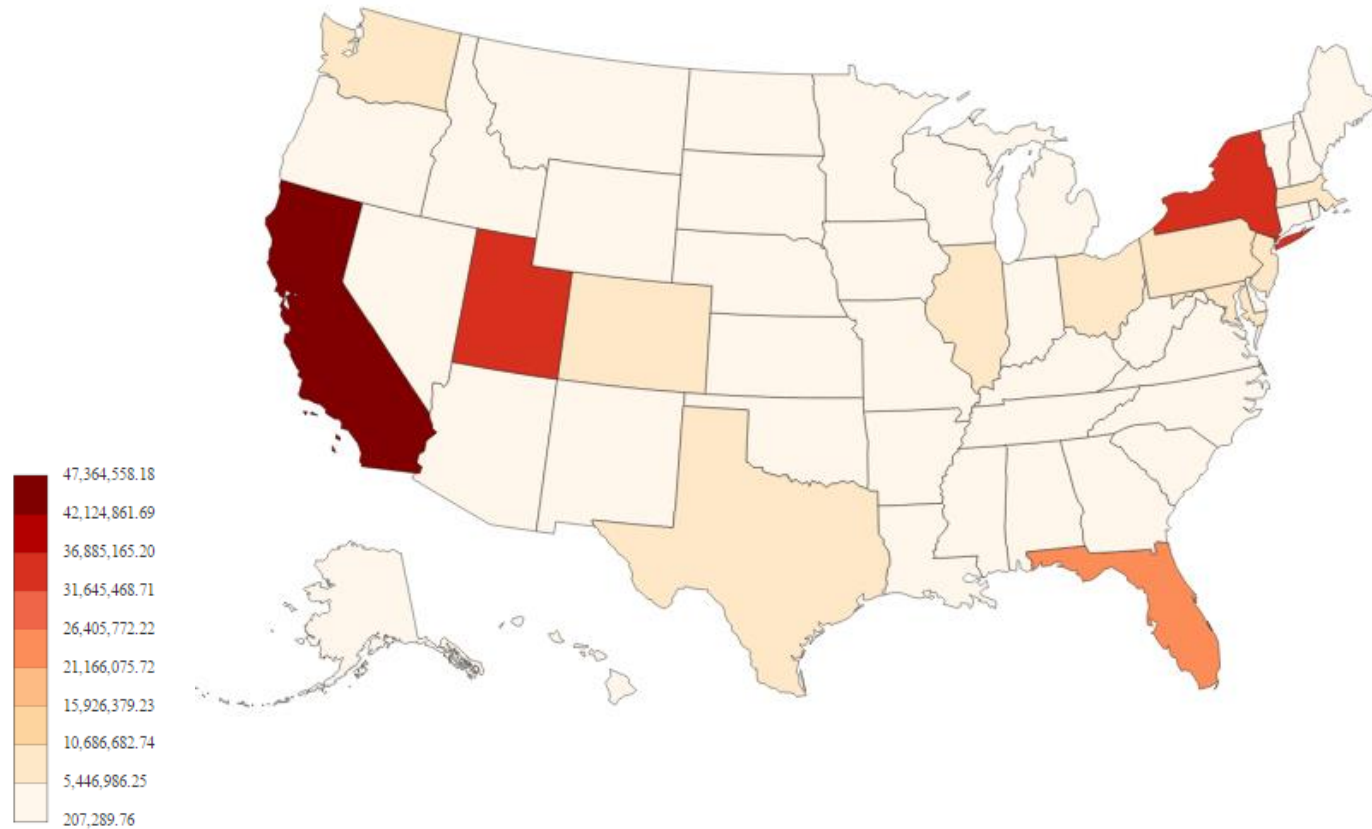
Amount in Millions Each State Donated to Presidential Campaigns

Contributions to presidential campaigns during 2015

Data obtained from the Federal Elections Committee

	47,364,558.18
	42,124,861.69
	36,885,165.20
	31,645,468.71
	26,405,772.22
	21,166,075.72
	15,926,379.23
	10,686,682.74
	5,446,986.25
	207,289.76

Likelihood of Individual Donation to Presidential Candidates by State

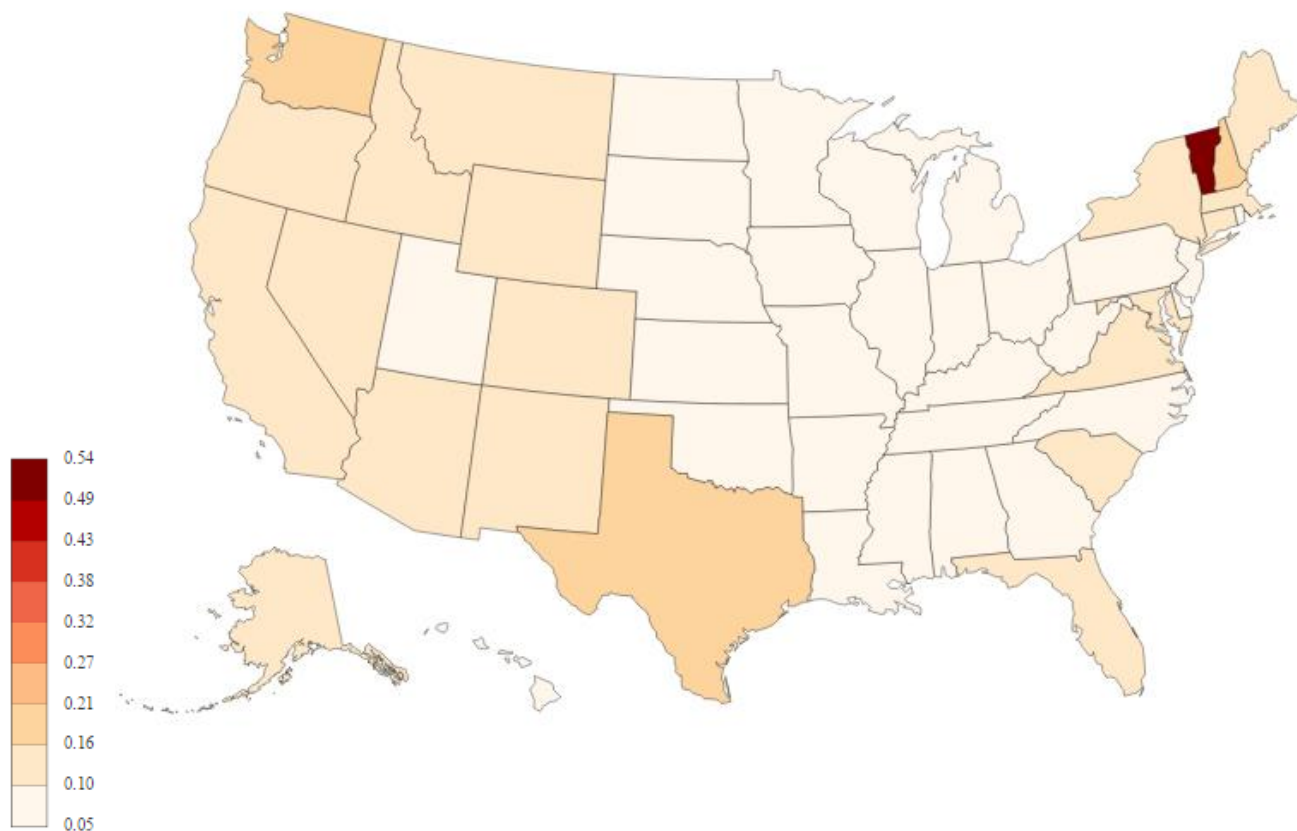## Contributions to presidential campaigns during the primary season
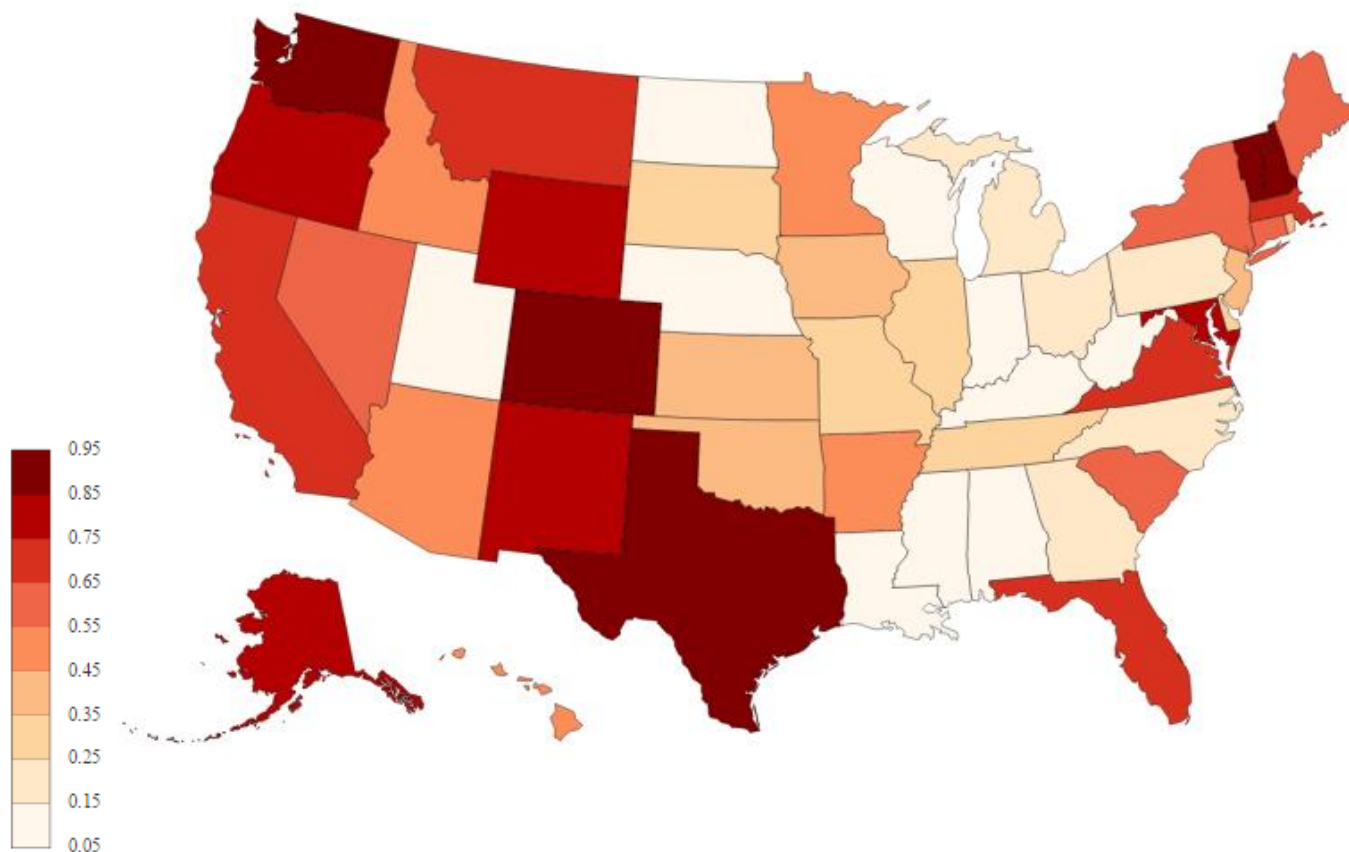
Data obtained from the Federal Elections Committee

Percentile Likelihood of Individual Donation to Presidential Candidates by State

Contributions to presidential campaigns during the primary season

Likelihood measure generated by dividing the number of contributions given (December, 2015) by the state population and then finding the percentile of each likelihood measure

The end