**###**

**For this project, I’ve compiled code in coffeescript, a variant of javascript. There are several files related to this one, but this is the heart of the visualization. This file is then loaded into an html file. There is also an associated css file and a data file.**

**This is how the code would appear in a text editor, or on Github: https://github.com/EmptyCan/FEC-project.git**

**Comments in the code (which aren’t processed, are denoted by # marks.**

**###**

**#Start of code**

**#creates basic parameters of the chart.**

class BubbleChart

constructor: (data) ->

@data = data

@width = 940

@height = 600

@tooltip = CustomTooltip("gates\_tooltip", 240)

@center = {x: @width / 2, y: @height / 2}

**#these values, along with others, determine the physics of the d3 visualization.**

@layout\_gravity = -0.01

@damper = 0.1

@vis = null

@nodes = []

@force = null

@circles = null

**# These are the colors of the circles, here tied to which causes each group supports.**

@fill\_color = d3.scale.ordinal()

.domain(["Conservative", "None", "Liberal"])

.range(["#E3170D", "#424242", "#4169E1"])

**#This ties the max amount displayed in the visualization to the highest total listed in the data (d.total)**

max\_amount = d3.max(@data, (d) -> parseInt(d.total))

@radius\_scale = d3.scale.pow().exponent(0.5).domain([0, max\_amount]).range([2,99])

this.create\_nodes()

this.create\_vis()

**# In d3, nodes are data points. For a bubble chart, this**

**# means each node will be a circle. We pull in data from**

**# our csv file to create each specific node.**

create\_nodes: () =>

@data.forEach (d) =>

node = {

id: d.id

radius: @radius\_scale(parseInt(d.total))

value: d.total

name: d.group

leaning: d.leaning

x: Math.random() \* 900

y: Math.random() \* 800

}

@nodes.push node

@nodes.sort (a,b) -> b.value - a.value

**###**

**D3 uses CSS selector tags, in a similar way to jQuery. Here it selects a CSS tag called #vis and appends an svg to it. This will become our graphic. There is a .div in the html with this class id.**

**###**

create\_vis: () =>

@vis = d3.select("#vis").append("svg")

.attr("width", @width)

.attr("height", @height)

.attr("id", "svg\_vis")

@circles = @vis.selectAll("circle")

.data(@nodes, (d) -> d.id)

that = this

**###**

**One of the keys to d3. As data “enters” the svg, it can be acted on. Basically this code is saying give node without a circle, a circle and make that circle have this list of attributes. These circles have a radius of zero when they are made, but that quickly changes.**

**###**

@circles.enter().append("circle")

.attr("r", 0)

.attr("fill", (d) => @fill\_color(d.leaning))

.attr("stroke-width", 2)

.attr("stroke", (d) => d3.rgb(@fill\_color(d.leaning)).darker())

.attr("id", (d) -> "bubble\_#{d.id}")

.on("mouseover", (d,i) -> that.show\_details(d,i,this))

.on("mouseout", (d,i) -> that.hide\_details(d,i,this))

**# This transistion gives the circles their final radius,**

**# and creates the nifty animation when first viewing the**

**# grapic.**

@circles.transition().duration(2000).attr("r", (d) => d.radius)

charge: (d) ->

-Math.pow(d.radius, 2.0) / 6

**# Starts up the force layout (d3’s physic’s engine)**

**# with the default values**

start: () =>

@force = d3.layout.force()

.nodes(@nodes)

.size([@width, @height])

**# Sets up force layout to display**

**# all nodes in one circle.**

display\_group\_all: () =>

@force.gravity(@layout\_gravity)

.charge(this.charge)

.friction(0.9)

.on "tick", (e) =>

@circles.each(this.move\_towards\_center(e.alpha))

.attr("cx", (d) -> d.x)

.attr("cy", (d) -> d.y)

@force.start()

**# Moves all circles towards the @center**

**# of the visualization**

move\_towards\_center: (alpha) =>

(d) =>

d.x = d.x + (@center.x - d.x) \* (@damper + 0.03) \* alpha

d.y = d.y + (@center.y - d.y) \* (@damper + 0.03) \* alpha

# This creates the popup interactive box when you hover

# over each bubble.

show\_details: (data, i, element) =>

d3.select(element).attr("stroke", "black")

content = "<span class=\"name\">Title:</span><span class=\"value\"> #{data.name}</span><br/>"

content +="<span class=\"name\">Amount:</span><span class=\"value\"> $#{addCommas(data.value)}</span><br/>"

content +="<span class=\"name\">Leaning</span><span class=\"value\"> #{data.leaning}</span>"

@tooltip.showTooltip(content,d3.event)

hide\_details: (data, i, element) =>

d3.select(element).attr("stroke", (d) => d3.rgb(@fill\_color(d.leaning)).darker())

@tooltip.hideTooltip()

root = exports ? this

# I don’t quite understand this function, but it basically # puts the process in motion, I think. What I do know is # that this is where the data is drawn from.

$ ->

chart = null

render\_vis = (csv) ->

chart = new BubbleChart csv

chart.start()

root.display\_all()

root.display\_all = () =>

chart.display\_group\_all()

d3.csv "data/FEC\_short.csv", render\_vis