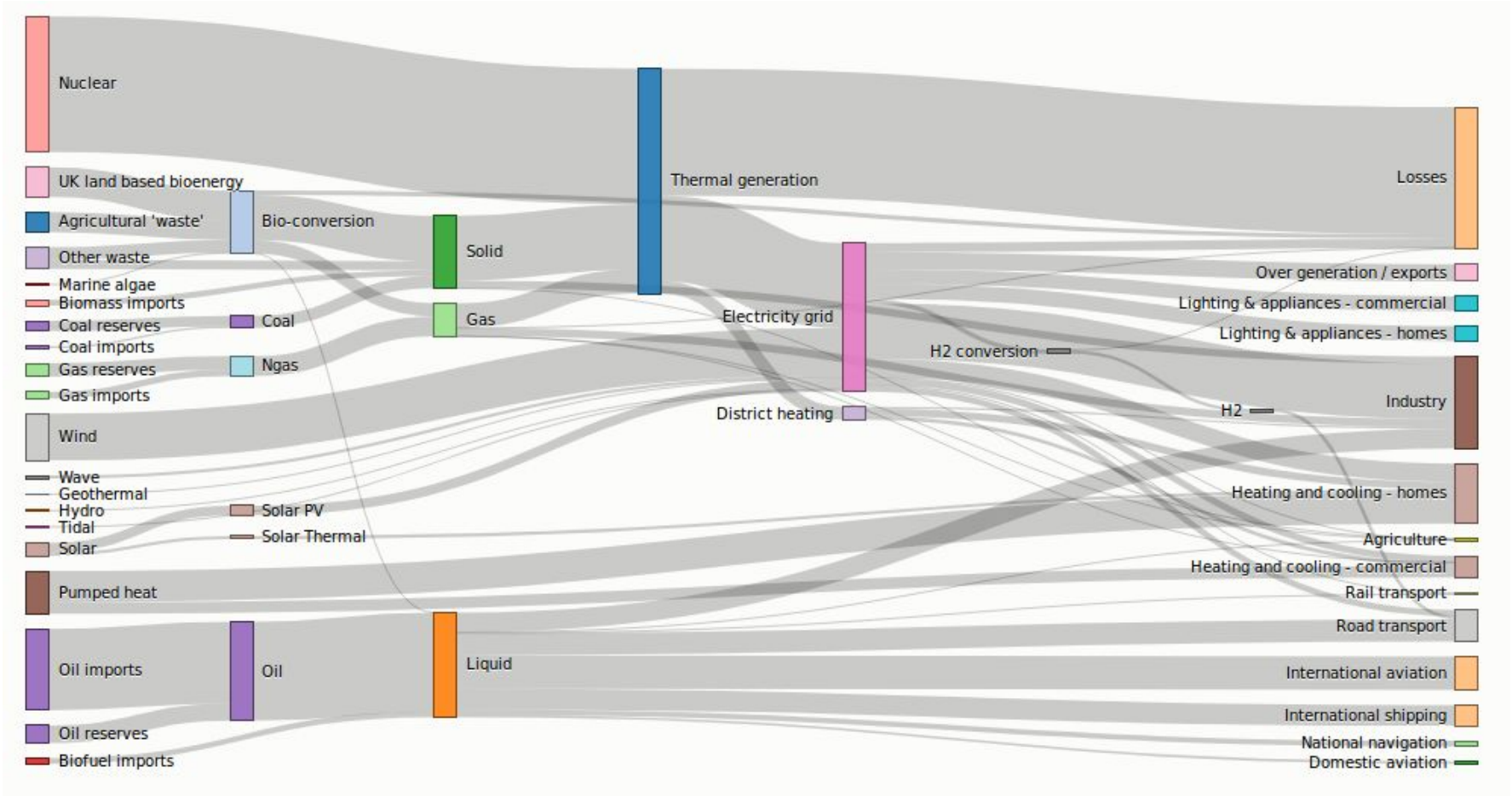
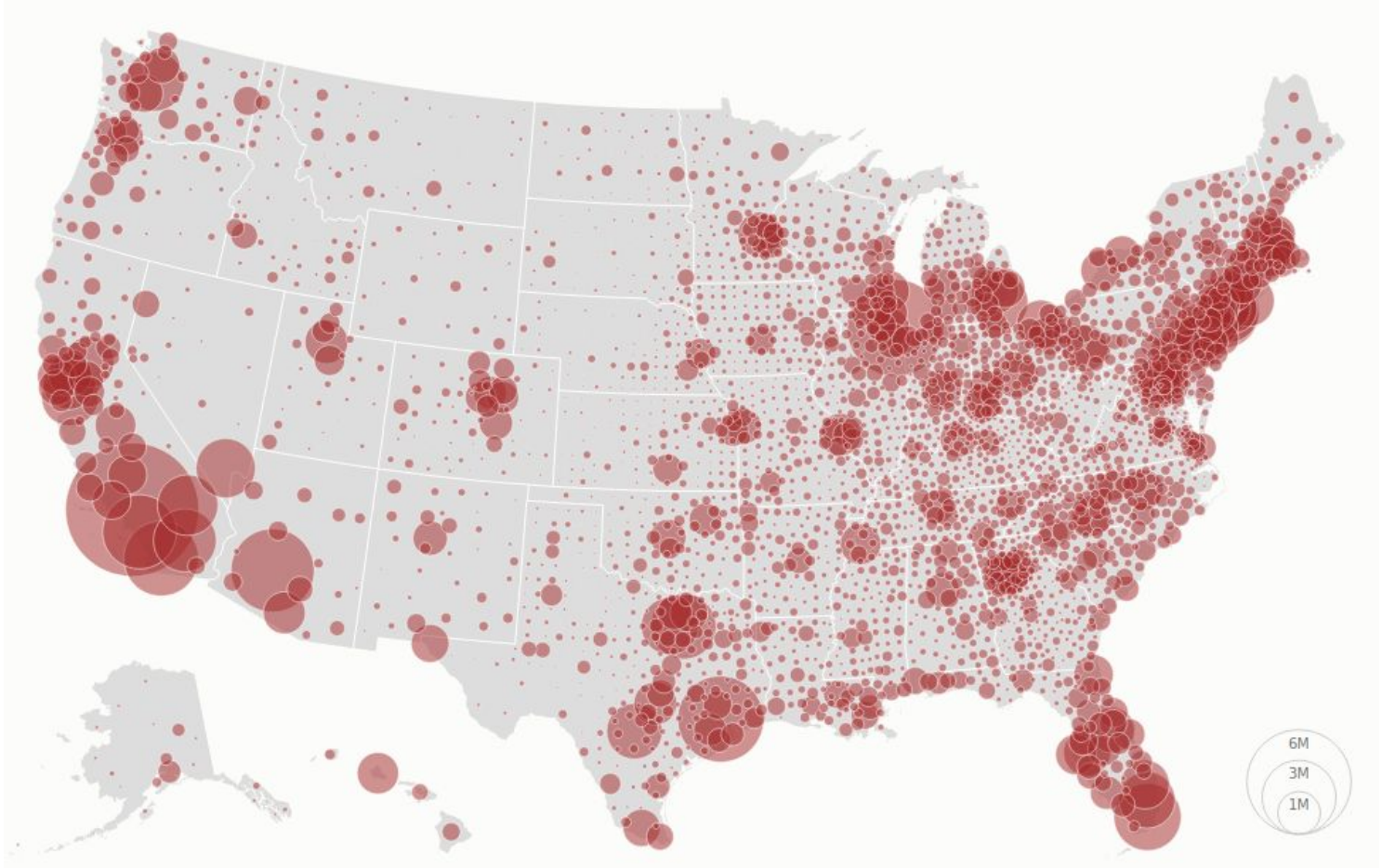
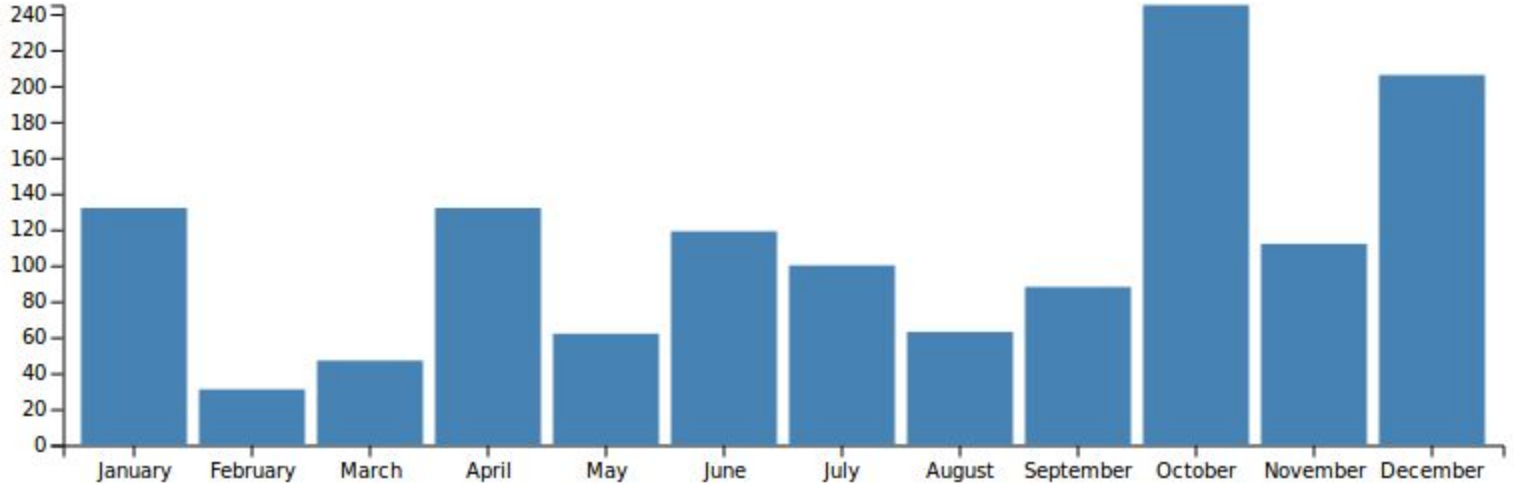
Hands on D3.js

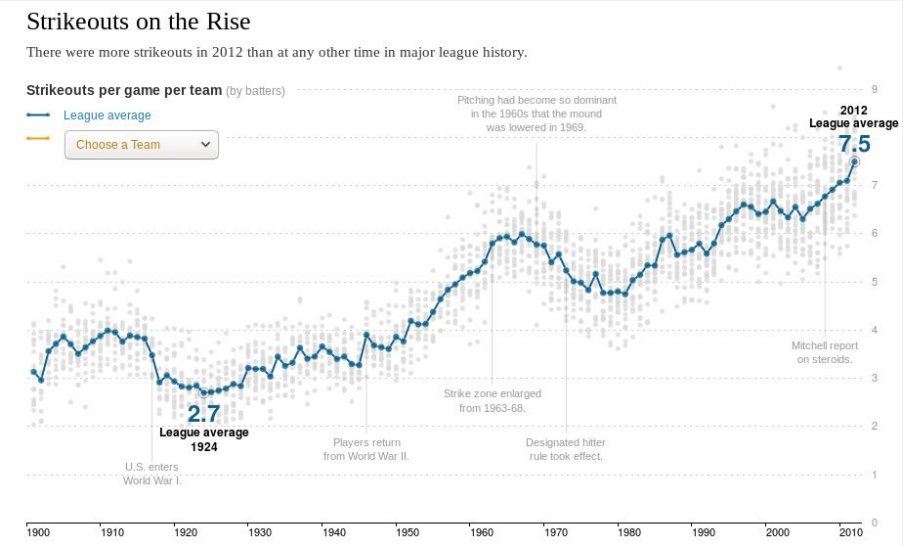
What can I do with D3?


vis.operator.layout
vis
analytics.cluster
analytics.graph
analytics.optimization
animate
animate.interpolate
data.converters
data
display
flex
physics
query
query.methods
scale
util
util.heap
util.math
util.palette
vis.axis
vis.controls
vis.data
vis.data.render
vis.events
vis.legend
is.operator.distortion
vis.operator.encoder
vis.operator.filter
vis.operator
vis.operator.label


Nuclear
UK land based bioenergy
Agricultural 'waste'
Other waste
Marine algae
Biomass imports
Coal reserves
Coal imports
Gas reserves
Gas imports
Wind
Wave
Geothermal
Hydro
Tidal
Solar
Pumped heat
Oil imports
Oil reserves
Biofuel imports
Bio-conversion
Coal
Ngas
Solar PV
Solar Thermal
Oil
Solid
Gas
Liquid
Thermal generation
Electricity grid
District heating
H2 conversion
Losses
Over generation / exports
Lighting & appliances - commercial
Lighting & appliances - homes
Industry
H2
Heating and cooling - homes
Agriculture
Heating and cooling - commercial
Rail transport
Road transport
International aviation
International shipping
National navigation
Domestic aviation


6M
3M
1M


240
220
200
180
160
140
120
100
80
60
40
20
0
January
February
March
April
May
June
July
August
September
October
November
December



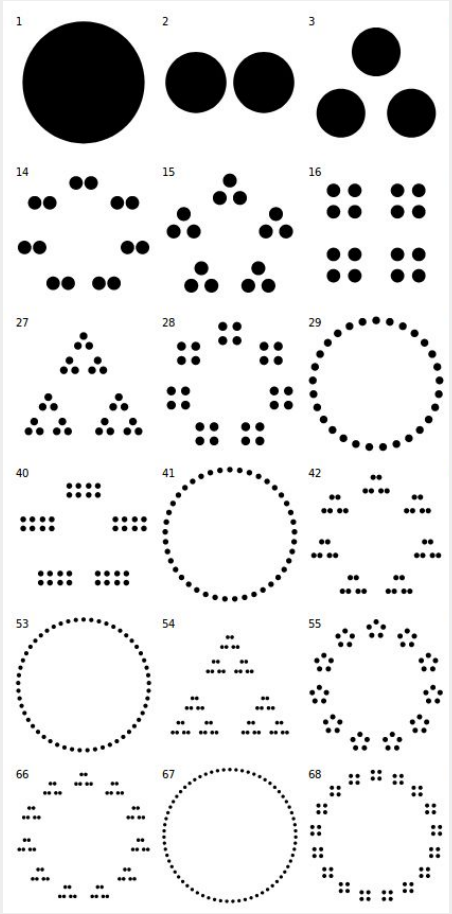
What is D3?

- it's not a charting library (!= highcharts)
- it's like jQuery
- it's like lodash
- data manipulating tools that output html or svg
- provide graphical primitives
- provide helper function
- SVG?
  - vectorial chart
  - xhtml

What does SVG look like?

```
<svg width="800" height="300">
  <g transform="translate(0,260)" fill="none" font-size="10" font-family="sans-serif" text-anchor="middle">…</g>
  <g transform="translate(40,0)" fill="none" font-size="10" font-family="sans-serif" text-anchor="end">…</g>
  <g>
    <rect x="49" y="141.46938775510205" width="53" height="118.53061224489795" fill="steelblue"></rect>
    <rect x="108" y="232.16326530612244" width="53" height="27.83673469387756" fill="steelblue"></rect>
    <rect x="167" y="217.79591836734693" width="53" height="42.20408163265307" fill="steelblue"></rect>
    <rect x="226" y="141.46938775510205" width="53" height="118.53061224489795" fill="steelblue"></rect>
    <rect x="285" y="204.3265306122449" width="53" height="55.67346938775509" fill="steelblue"></rect>
    <rect x="344" y="153.14285714285714" width="53" height="106.85714285714286" fill="steelblue"></rect>
    <rect x="403" y="170.20408163265307" width="53" height="89.79591836734693" fill="steelblue"></rect>
    <rect x="462" y="203.42857142857144" width="53" height="56.571428571428555" fill="steelblue"></rect>
    <rect x="521" y="180.9795918367347" width="53" height="79.0204081632653" fill="steelblue"></rect>
    <rect x="580" y="40" width="53" height="220" fill="steelblue"></rect>
    <rect x="639" y="159.42857142857144" width="53" height="100.57142857142856" fill="steelblue"></rect>
    <rect x="698" y="75.0204081632653" width="53" height="184.9795918367347" fill="steelblue"></rect>
  </g>
</svg>
```

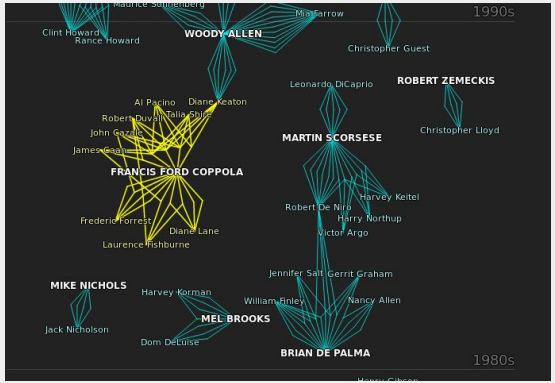
D3 API



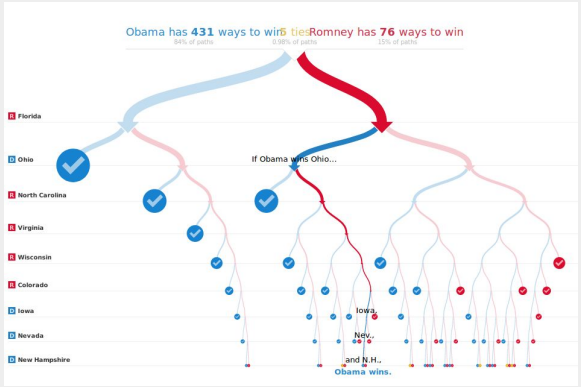
Manipulating Element

- d3|myElement.select()
- d3|myElement.selectAll()
- myElement.append()
- myElement.attr()
- myElement.text()
- myElement.style()

Manipulating Data



- d3.tsv("path to file", function to transform data)
- myElement.data(myArray)
- .enter(): after .data() or .tsv(), let you iterate on row
- d3.min(myArray, function(d) { return d.myValue})
- d3.max(myArray, function(d) { return d.myValue})



Scale and Axis

- d3.scaleBand()
  - myScale.domain([value1, value2, ...])
  - myScale.bandwidth()
- d3.scaleLinear()
  - myScale.domain([minValue, maxValue])

- myScale.range([min, max]) // range size in pixel
- myScale.rangeRound([min, max]) // range size in pixel
- myScale.padding(aFloat)
- myScale(aNumber) // convert a number from value to pixel

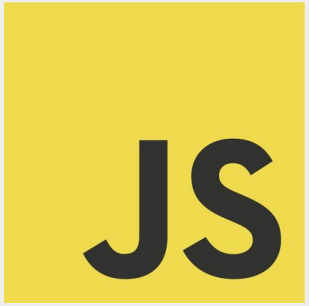
- d3.axisLeft(myScale) - draw a vertical scale with label and tick on left
- d3.axisBottom(myScale) - draw an horizontal scale with label and tick on bottom

Tips

- read instruction carefully
- zero is on top left
- you draw from 0
- don't forget to add padding between element
- use "transform" attribute with "translate()" to move element on the correct place.
  - `.attr("transform", "translate(0,12)")`
- you can chain function but beware of order
- lambda in js: `function(d) { return d + 1; }`
- don't forget ";" really put it everywhere
- undefined is not a function

Your Turn

Bar Charts



Step 0

1. Open a web browser
2. Go To: https://jsfiddle.net/19tcwseg/
3. Do a bar chart. 
4. D3 API: https://github.com/d3/d3/blob/master/API.md

Step 1

manipulating element

- Create a div tag
  - inside body
  - width: 800
  - height: 300
  - text: hello world
- you will need
  - select()
  - append()
  - attr()
  - text()

Step 2

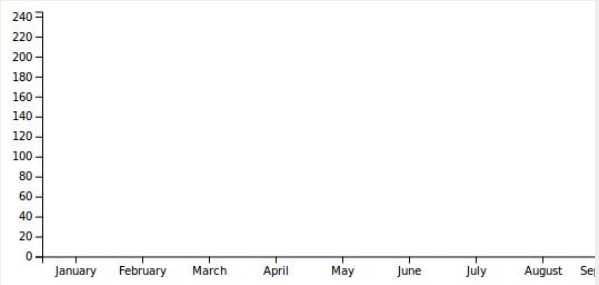
init your chart

- replace the div tag by a svg tag
- remove text
- add var for width height padding
  - width = 800
  - height = 300
  - padding = 20

Step 3

add scale and axis

- create xScale
  - scaleBand()
  - rangeRound([start, end]) // pixel value
  - domain(monthLabels)
- create yScale
  - scaleLinear()
  - range([start, end]) //pixel value
  - domain([0, maxRain])
- draw xAxis
  - append(“g”)
  - call(d3.axisBottom(xScale))
- draw yAxis
  - append(“g”)
  - call(d3.axisLeft(yScale)



step 4

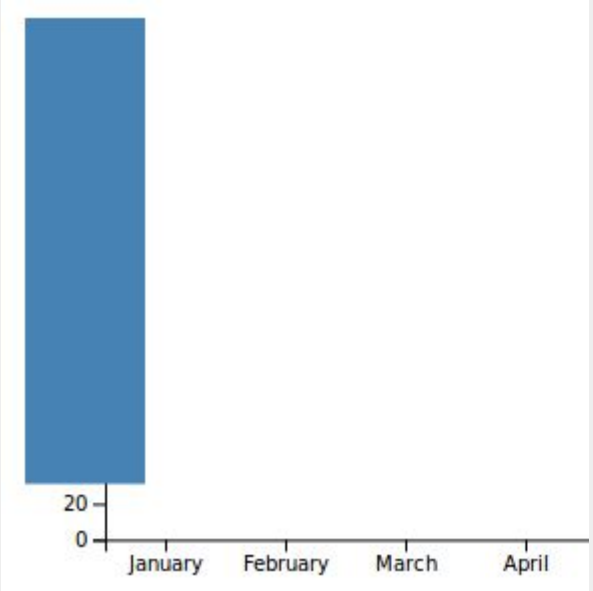
position the axis correctly

- xAxis element
  - translate the axis to bottom with transform attribute
  - add padding on height when you translate
- yAxis element
  - add padding (use transform translate)
- change xScale rangeRound to take padding in account
  - add padding on left
  - add padding on right
- change yScale range to take padding in account
  - add padding on top
  - add padding on bottom
- if necessary invert start and end in yScale.range

Step 5

init bar chart container

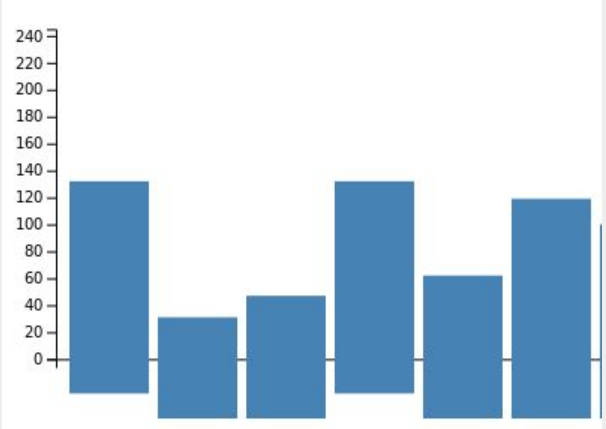
- on your svg object
  - append(“g”)
  - selectAll(“rect”)
  - data(“myData”)
  - enter()
  - append(“rect”)
- inspect the generated HTML to see:
  - where are rect tag?
  - how many rect tag have been created?

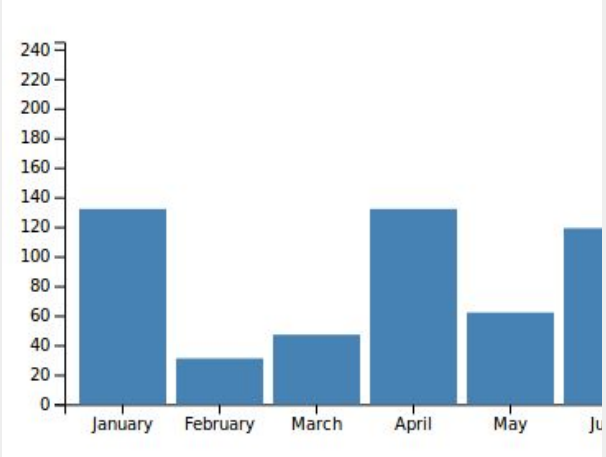


Step 6

draw bar

- add attributes to rectangle
  - fill = steelblue
  - width = x scale bandwidth
- add height attribute to rectangle
  - you can pass a one parameter function to attr. the parameter will be one row of your data
  - use yScale to convert value to pixel (yScale(myValue))
- inspect the HTML and check that all attributes are correctly set





Step 7

move bars on the correct place

- add x attribute to rect
  - like heigth you can pass a function
  - use xScale to convert month string to pixel value
- add y attribute to rect
  - use yScale to convert rain in pixel value
- change height attribute
  - you draw the bar from the top
  - yScale give you the number of pixel between the top of your scale and the top of your bar
  - don't forget that you have padding on the bottom of the chart
- add space between bar
  - call padding(aFloat) on xScale

Let's Continue

Pie Chart

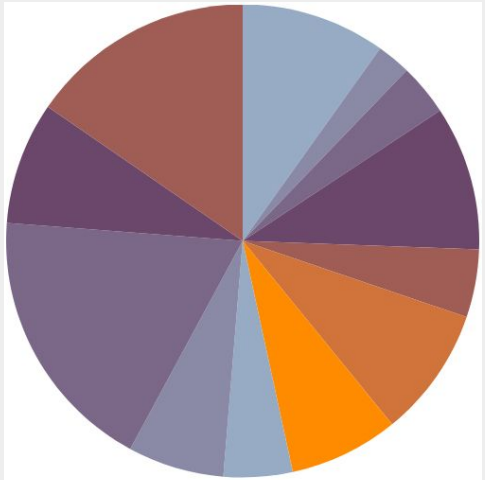
Step 0

1. open a web browser
2. go to: https://jsfiddle.net/27pxv5fj/

Step 1

prepare

- add a “g” element inside svg
  - translate it in the center of the chart
- add on ordinale scale of your favorite color
  - mine are: ["#98abc5", "#8a89a6", "#7b6888", "#6b486b", "#a05d56", "#d0743c", "#ff8c00"]
- prepare a d3.pie():
  - .sort(null)
  - .value(function where parameter return rain value)
- d3.pie()
  - compute startAngle and endAngle for your pie chart
  - your data are available in “data” key of each row



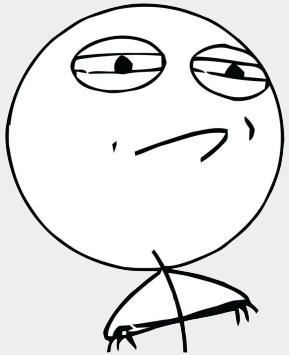
Step 2

make a pie

- pie are constructed from the center
- a slice of pie is made of ”path” element
  - “d” attribute contain the vector to draw the path
- d3.arc() can help to compute the path of an arc
  - .outerRadius()
  - .innerRadius()
  - .startAngle()
  - .endAngle()
  - don’t forget to call() to return the path
- fill the slice with some color from the ordinal scale
  - don’t worry if you don’t have enough color

You are To Fast

CHALLENGE ACCEPTED



Challenge

maybe you shouldn't has been so fast.

1. do a donut chart (this will make the presenter angry)
2. bar chart: add axis title
3. bar chart: add chart title (center, top)
4. bar chart: add a grid on bar chart (white grid over blue bar would be nice)
5. bar chart: add rain quantity on top of each bar
6. do a line chart
7. add label foreach slice of your pie chart
8. add a sorting functionality on pie chart
   - ▽ by month
   - ▽ by rain
9. do a grouped bar chart
10. find your own challenge (tooltips, animation, filter, draw a unicorn, ...)

Thank you

Bar chart Solution

Step 1: Nope
Step 2: Nope
Step 3: https://jsfiddle.net/19tcwseg/3/
Step 4:https://jsfiddle.net/19tcwseg/4/
Step 5:https://jsfiddle.net/19tcwseg/5/
Step 6: https://jsfiddle.net/19tcwseg/6/
Step 7: https://jsfiddle.net/19tcwseg/7/
Bonus 2-5: https://jsfiddle.net/19tcwseg/8/

Donut Chart Solution

Step 1: https://jsfiddle.net/27pxv5fj/1
Step 2: https://jsfiddle.net/27pxv5fj/2
Bonus 1: https://jsfiddle.net/27pxv5fj/3/