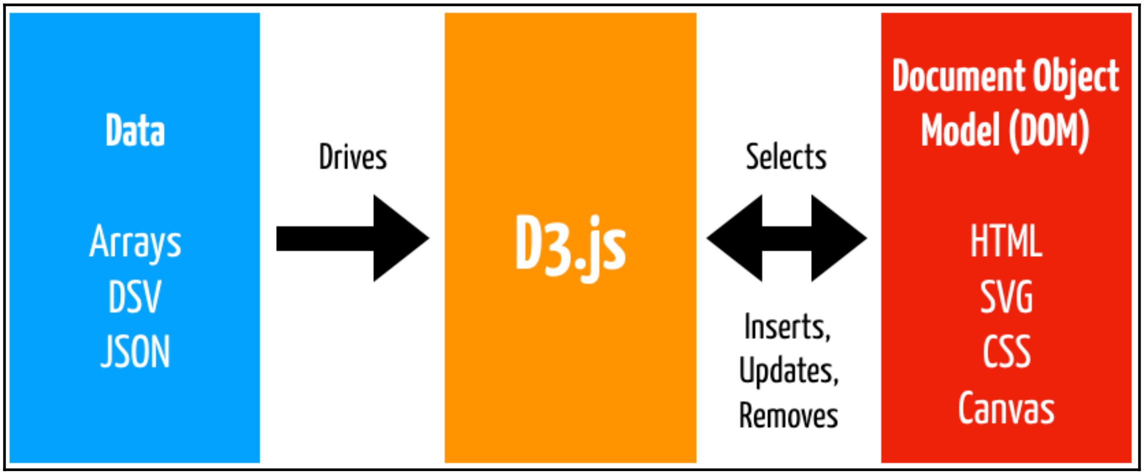
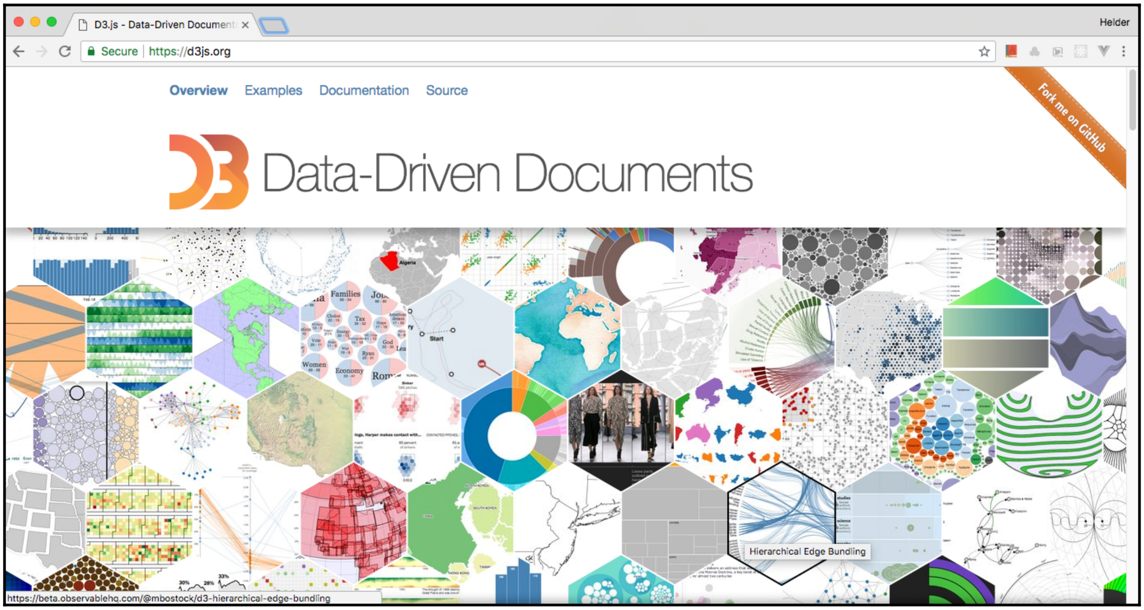
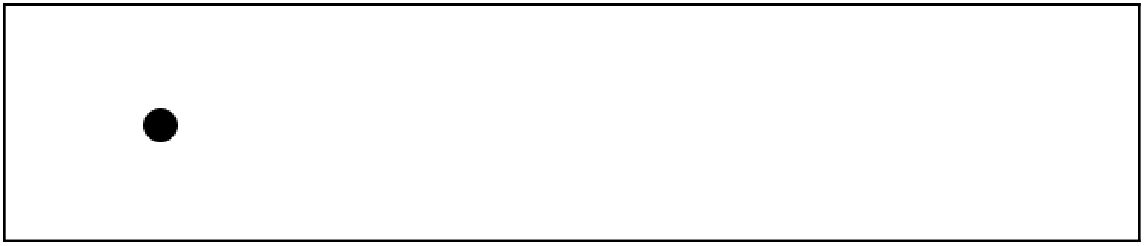
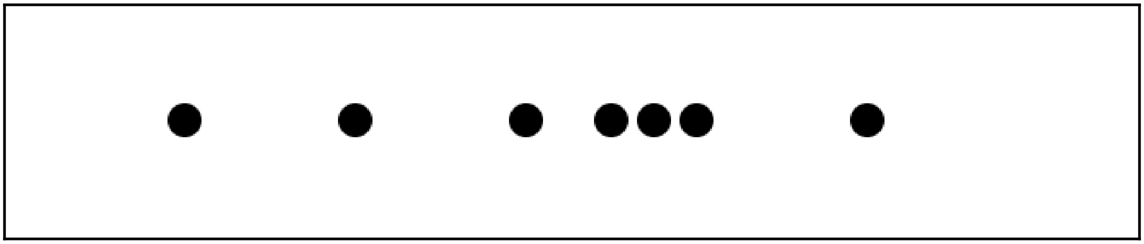
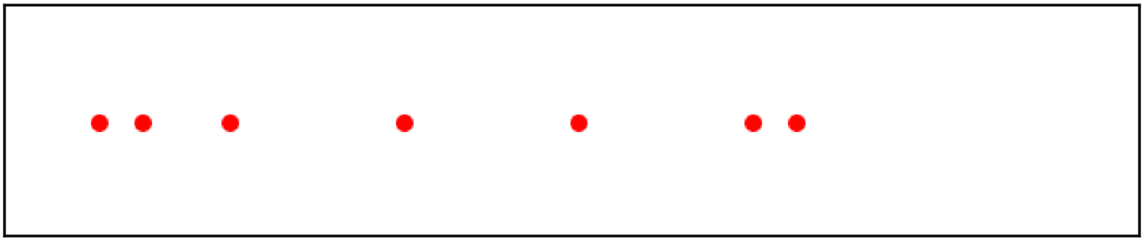
Chapter 1: Introduction

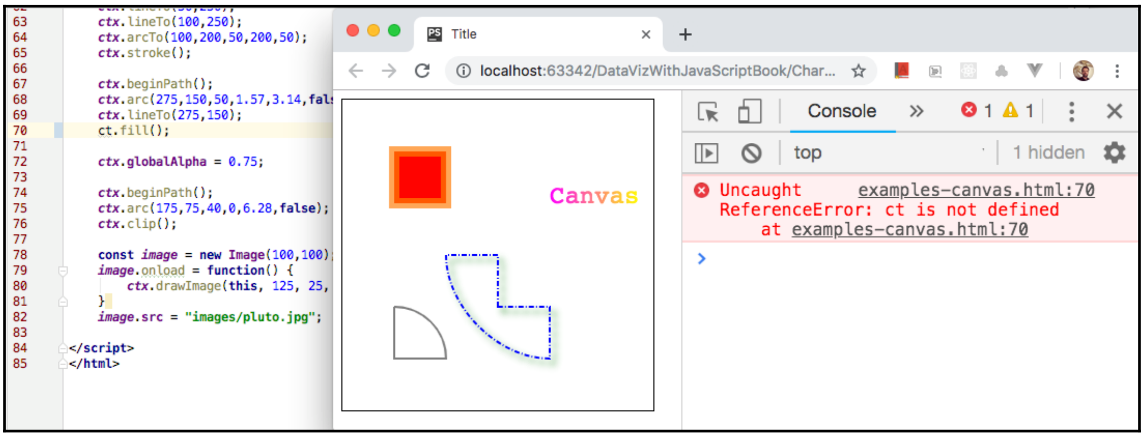




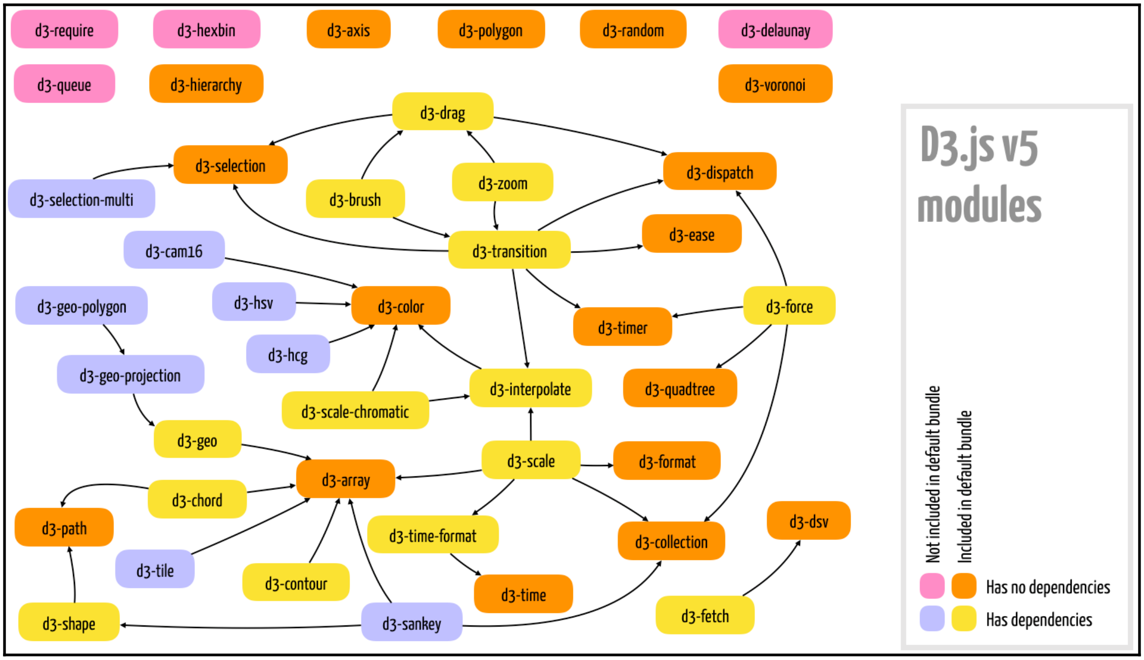






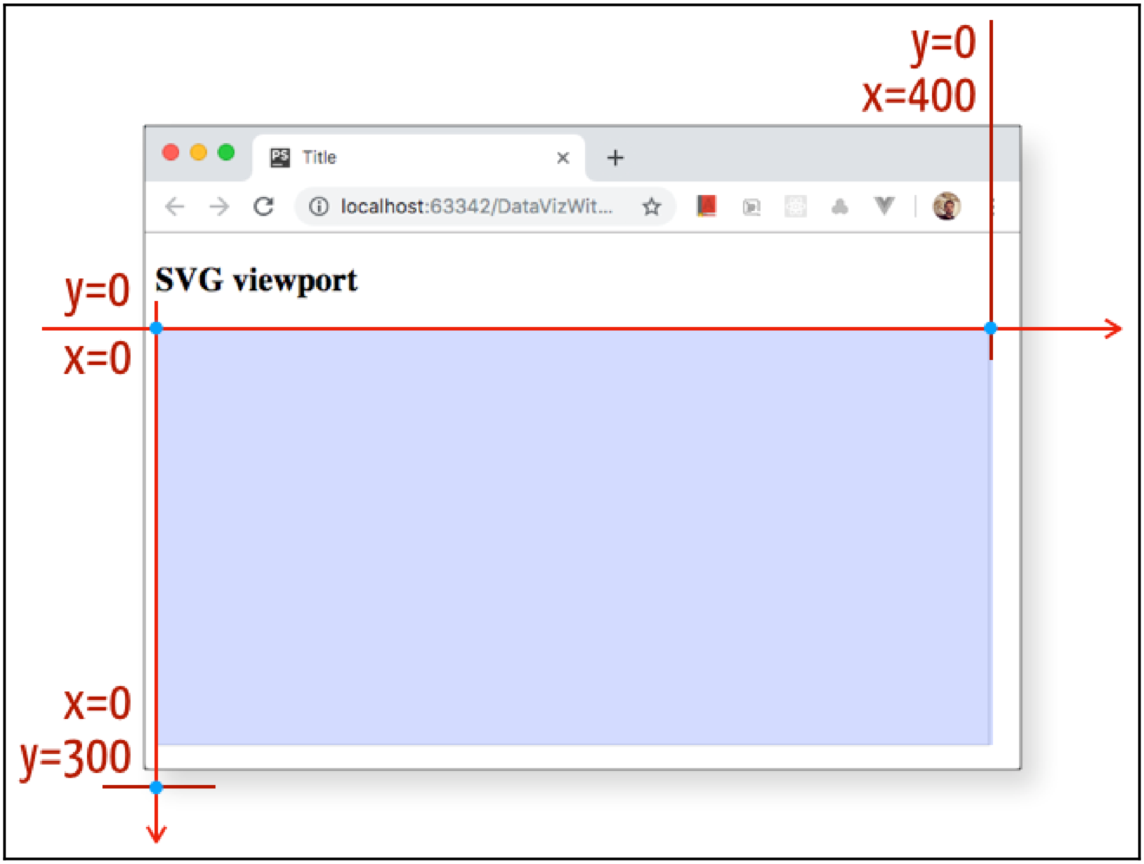


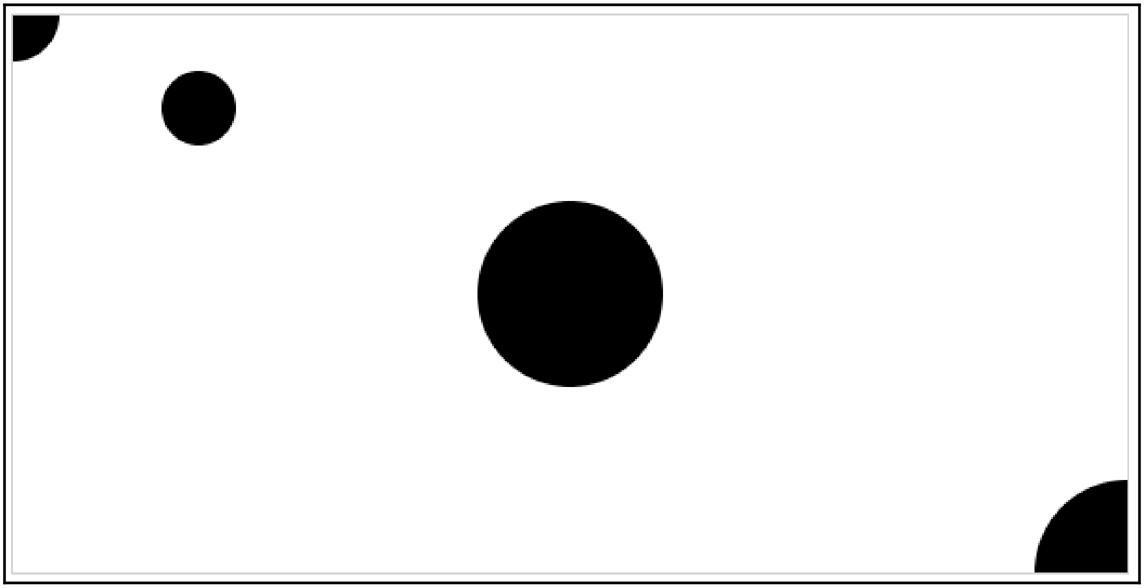
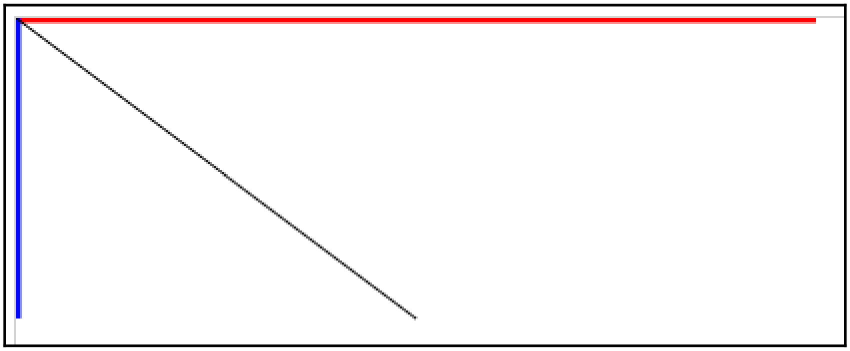
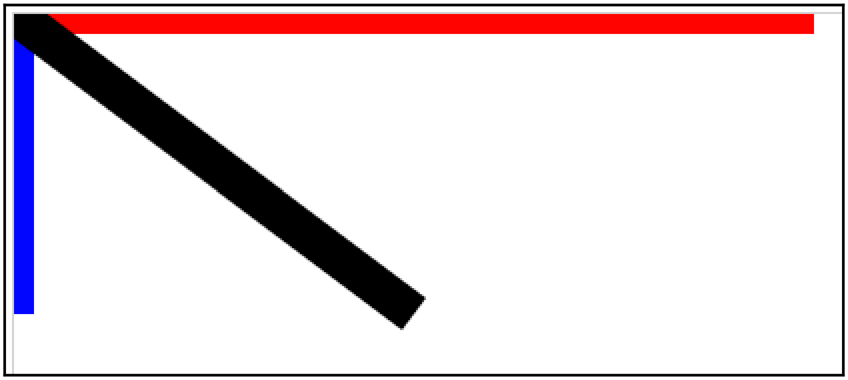
ctx.lineTo(100,250);
ctx.arcTo(100,200,50,200,50);
ctx.stroke();
ctx.beginPath();
ctx.arc(275,150,50,1.57,3.14,fal
ctx.lineTo(275,150);
ct.fill();
ctx.globalAlpha = 0.75;
ctx.beginPath();
ctx.arc(175,75,40,0,6.28,false);
ctx.clip();
const image = new Image(100,100)
image.onload = function() {
ctx.drawImage(this, 125, 25,
}
image.src = "images/pluto.jpg";
</script>
</html>
Title
localhost:63342/DataVizWithJavaScriptBook/Char...
Canvas
Console
top
1 hidden
Uncaught examples-canvas.html:70
ReferenceError: ct is not defined
at examples-canvas.html:70

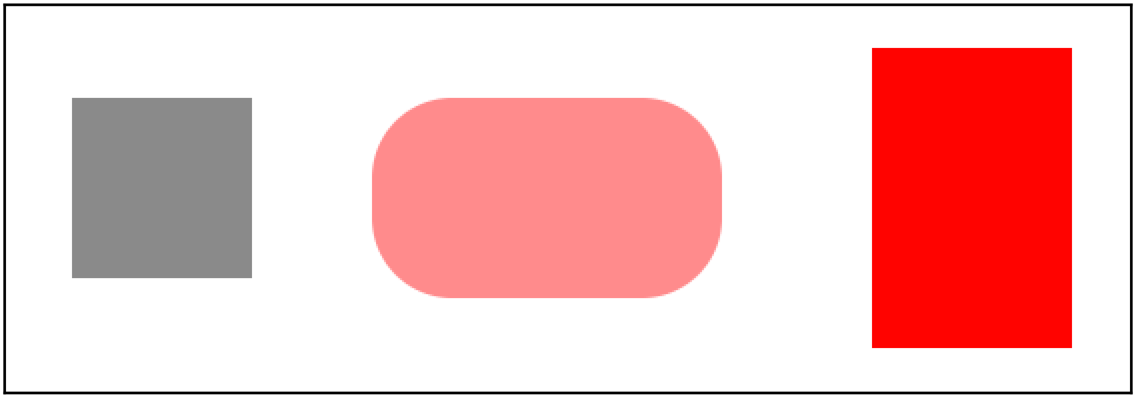


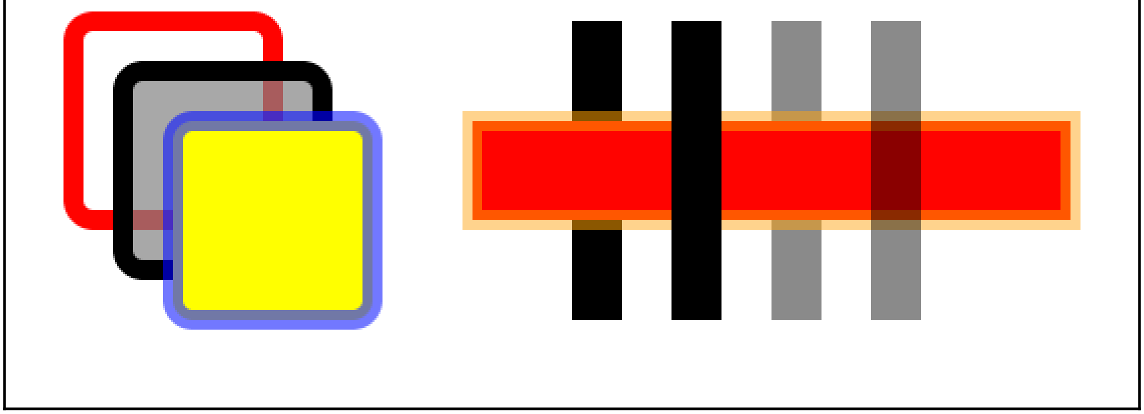
D3.js v5 modules
d3-require
d3-hexbin
d3-axis
d3-polygon
d3-random
d3-delaunay
d3-queue
d3-hierarchy
d3-voronoi
d3-drag
d3-selection
d3-zoom
d3-dispatch
d3-selection-multi
d3-brush
d3-transition
d3-ease
d3-cam16
d3-geo-polygon
d3-hsv
d3-color
d3-timer
d3-force
d3-hcg
d3-geo-projection
d3-scale-chromatic
d3-interpolate
d3-quadtree
d3-geo
d3-scale
d3-format
d3-array
d3-chord
d3-path
d3-time-format
d3-collection
d3-dsv
d3-tile
d3-contour
d3-time
d3-fetch
d3-shape
d3-sankey
Not included in default bundle
Included in default bundle
Has no dependencies
Has dependencies

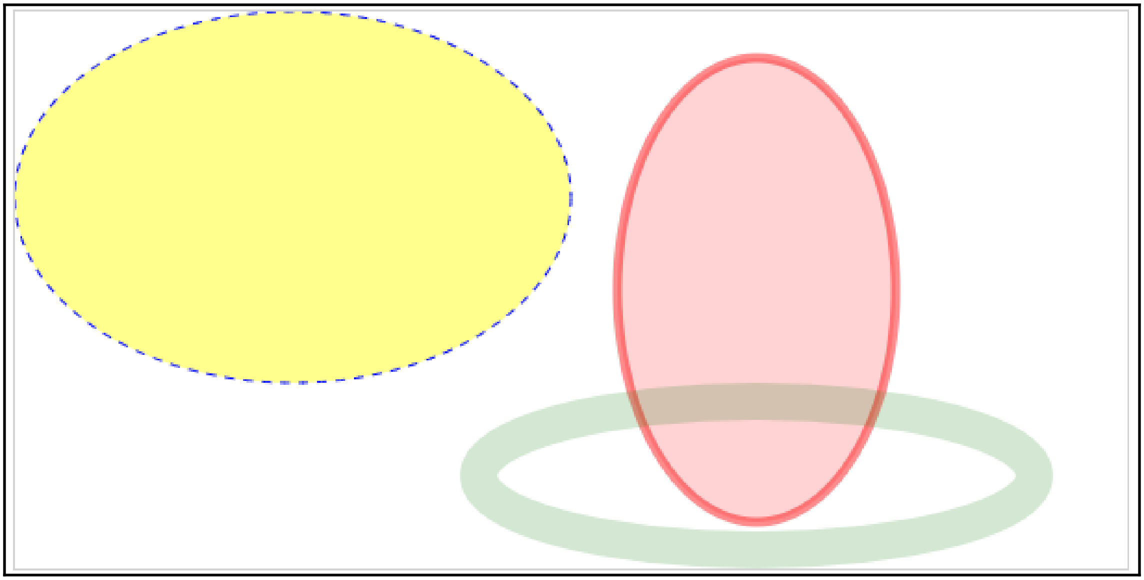
Chapter 2: Technical Fundamentals

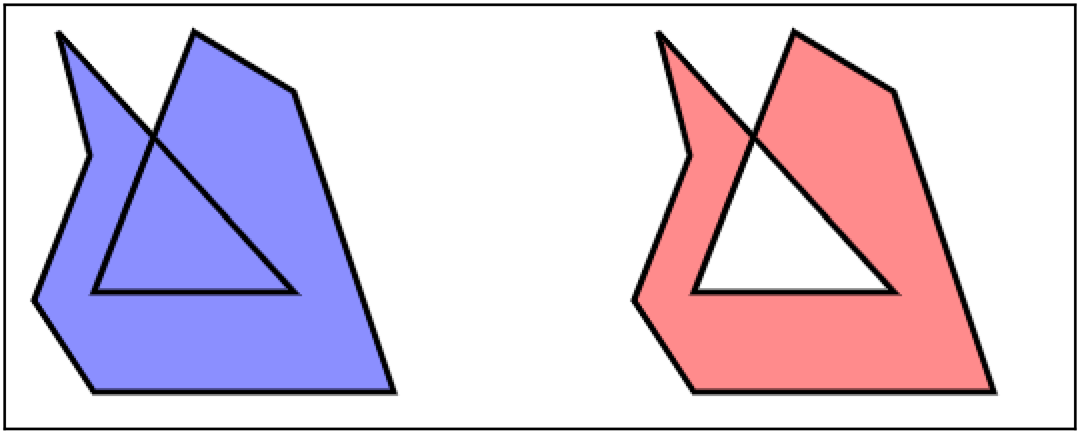


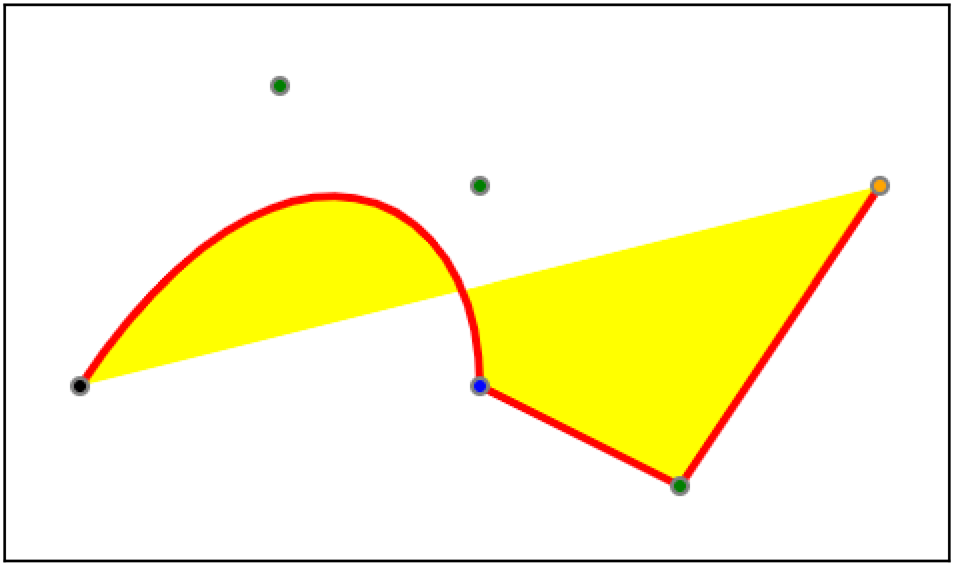




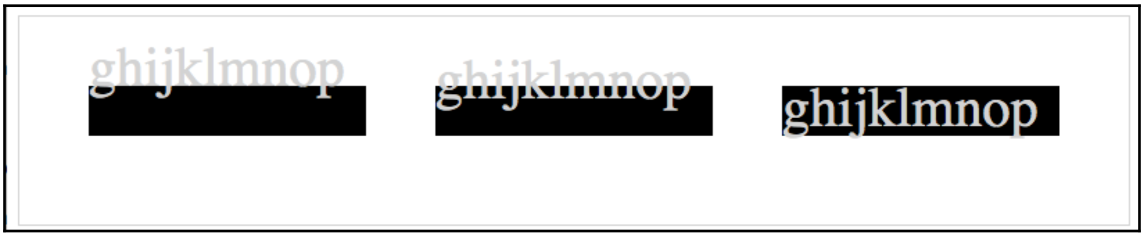


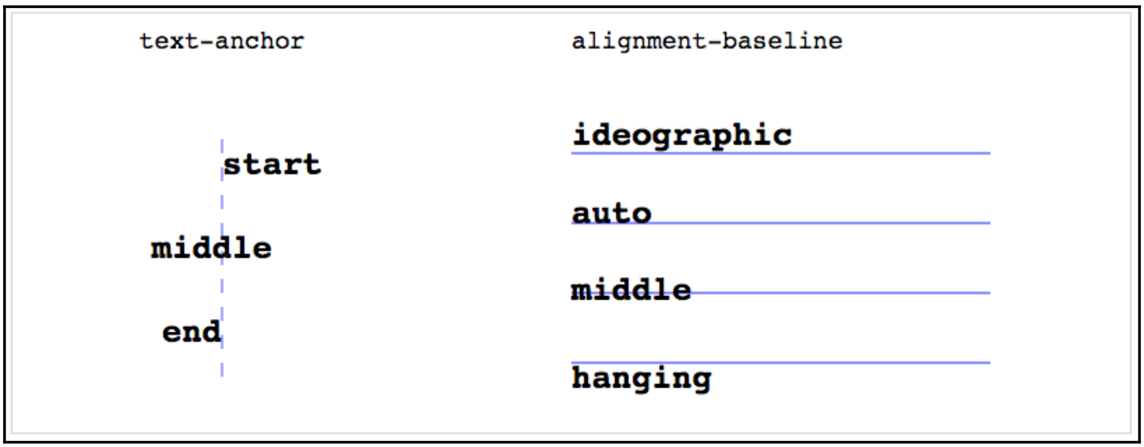


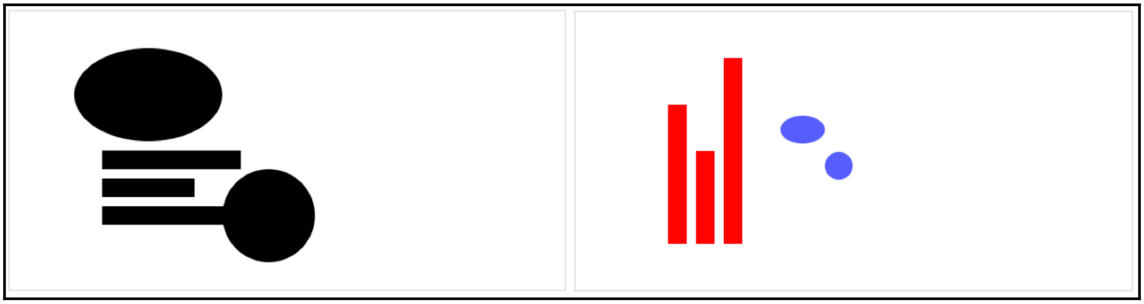


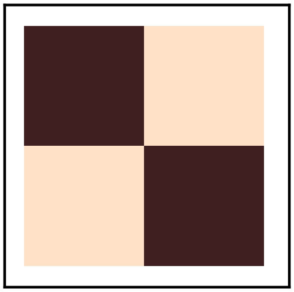


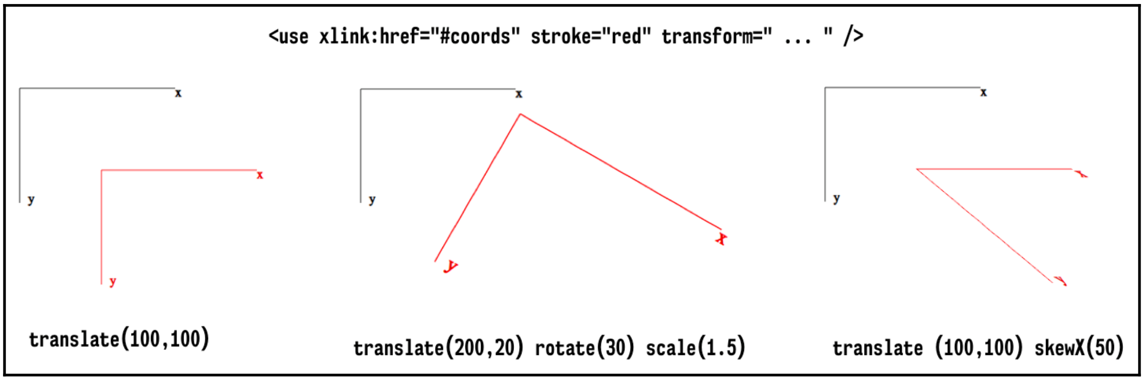


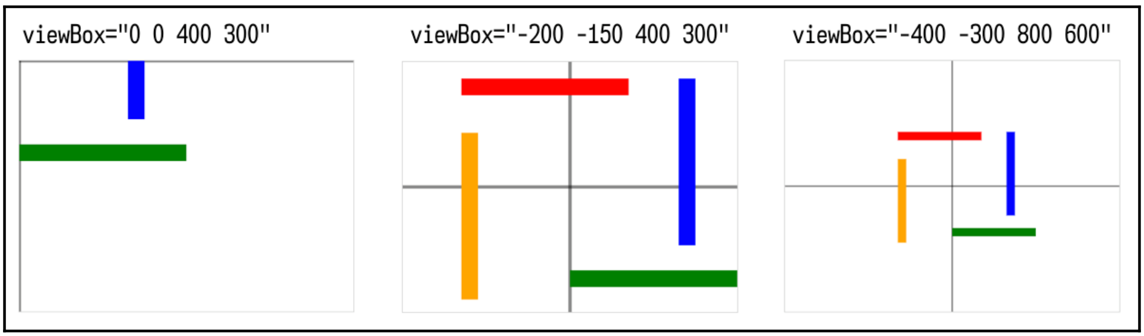

ghijklmnop
ghijklmnop
ghijklmnop

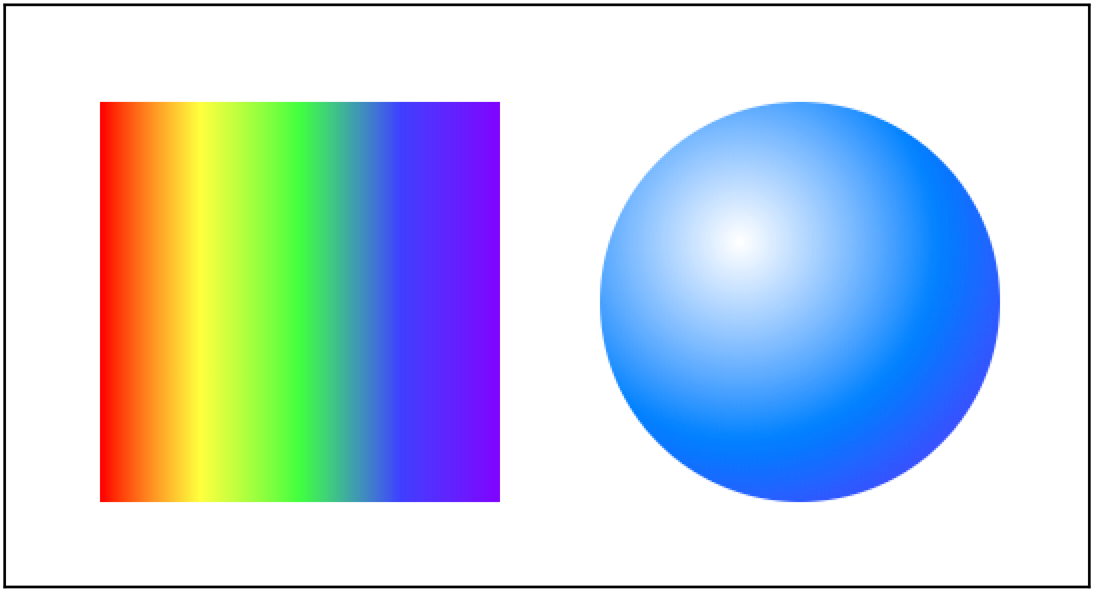

text-anchor
alignment-baseline
start
middle
end
ideographic
auto
middle
hanging

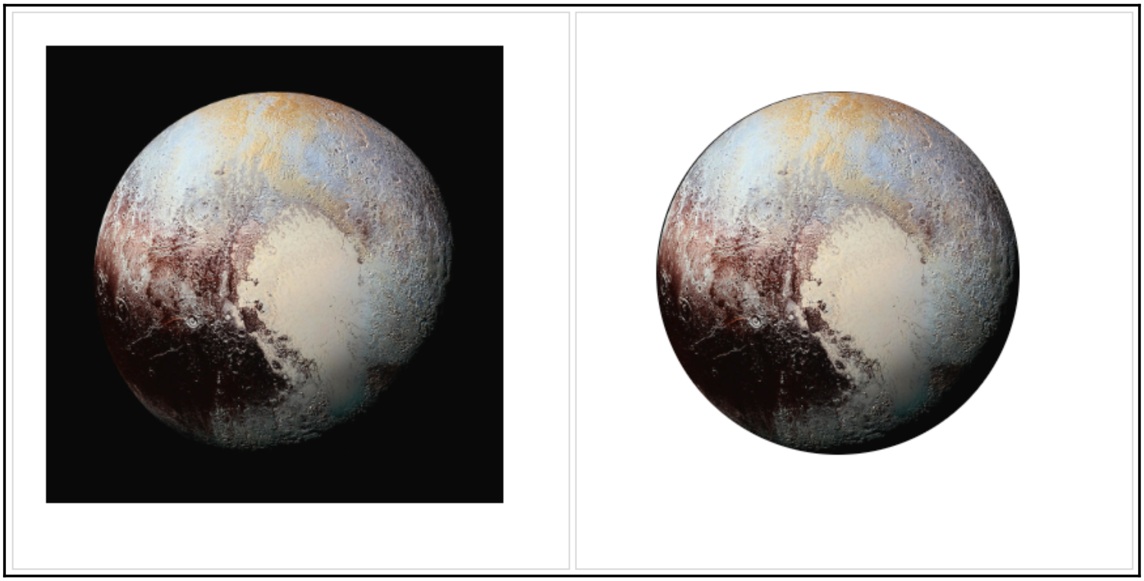


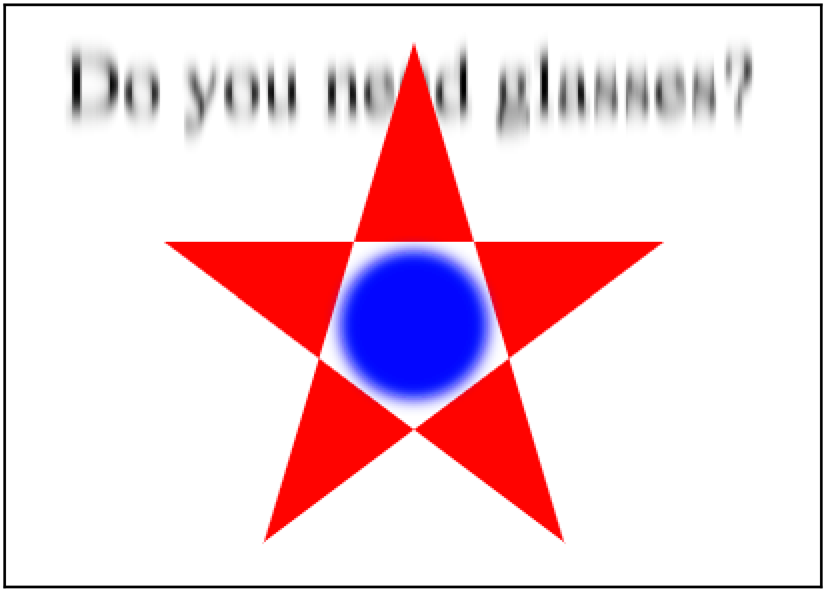


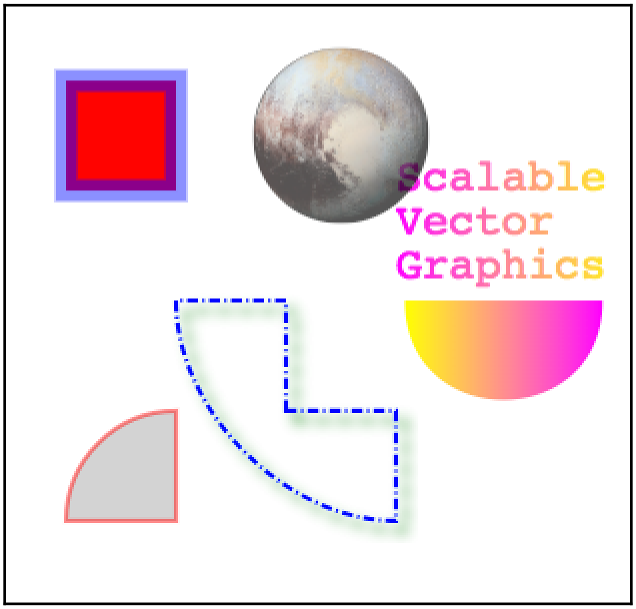

<use xlink:href="#coords" stroke="red" transform=" ... " />
x
y
x
y
x
y
x
y
x
y
translate(100,100)
translate(200,20) rotate(30) scale(1.5)
translate (100,100) skewX(50)

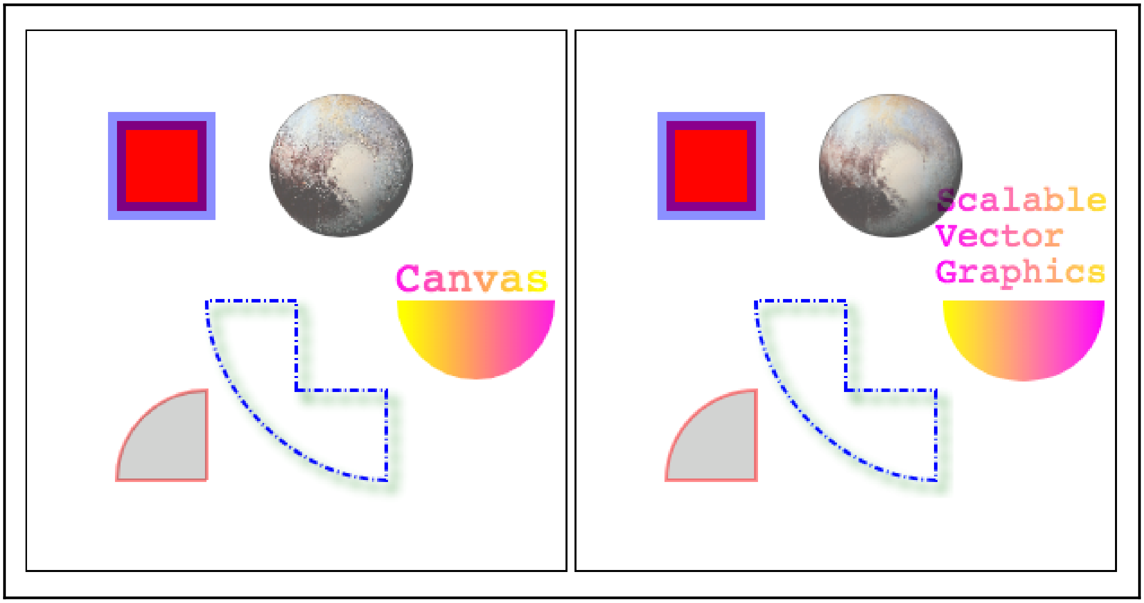

viewBox="0 0 400 300"
viewBox="-200 -150 400 300"
viewBox="-400 -300 800 600"



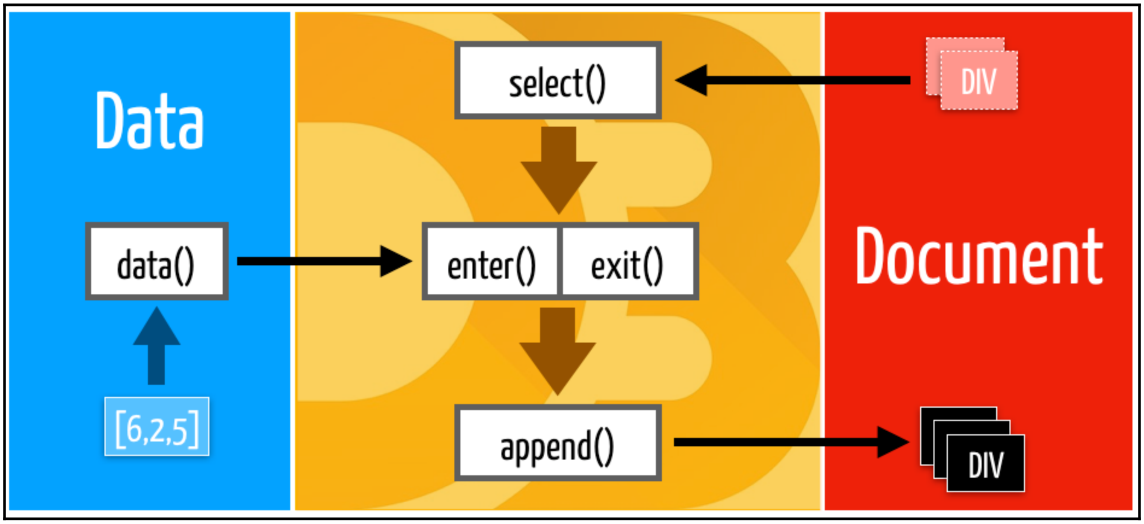


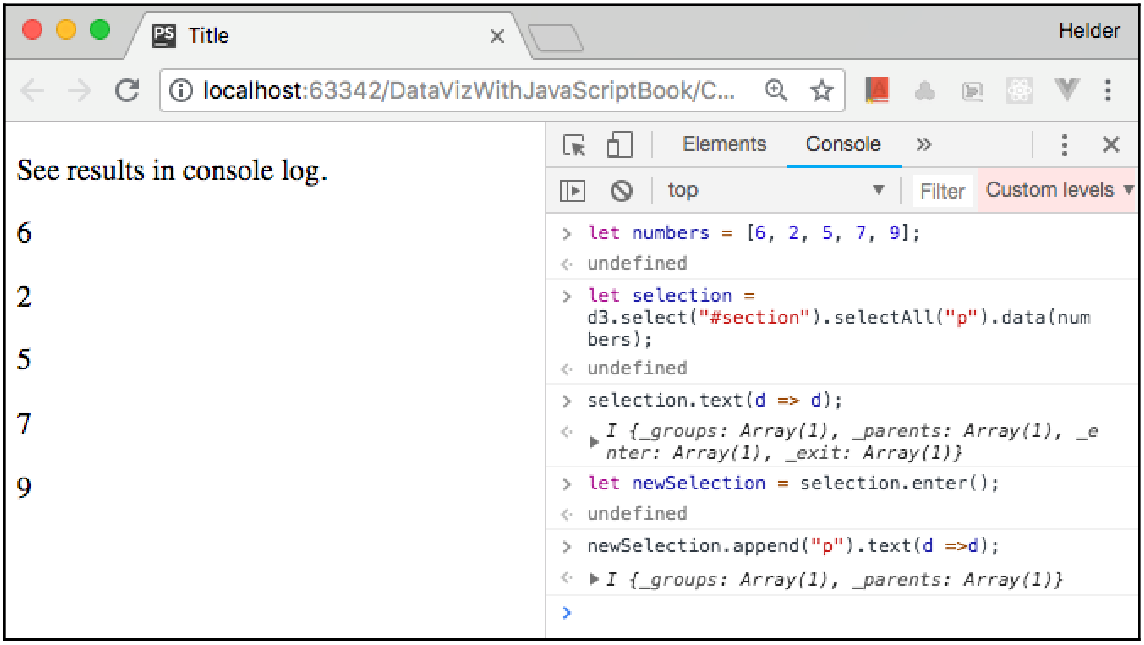

Do you need glasses?

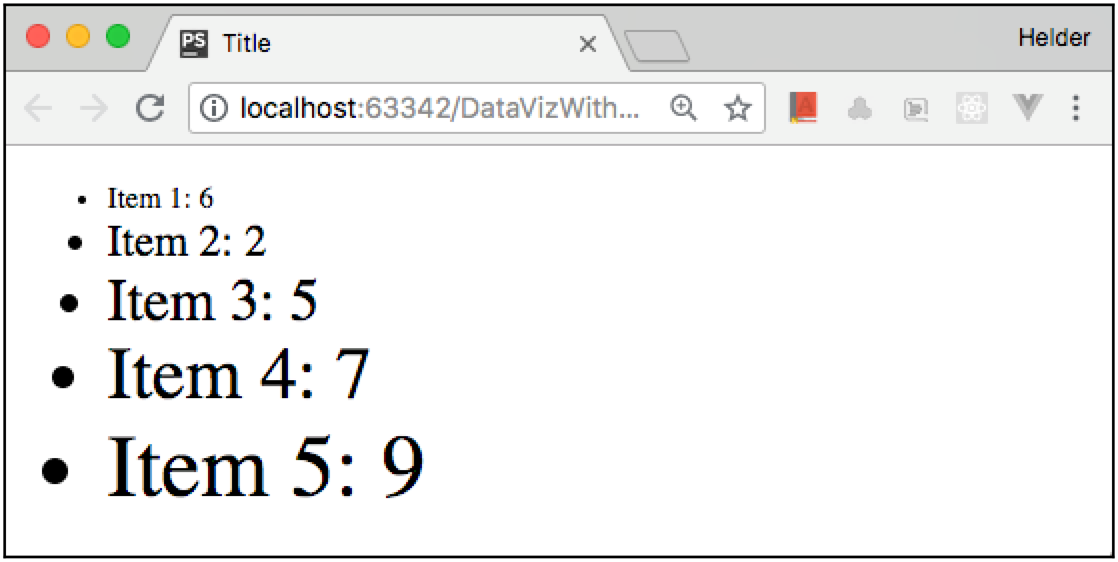

Scalable
Vector
Graphics

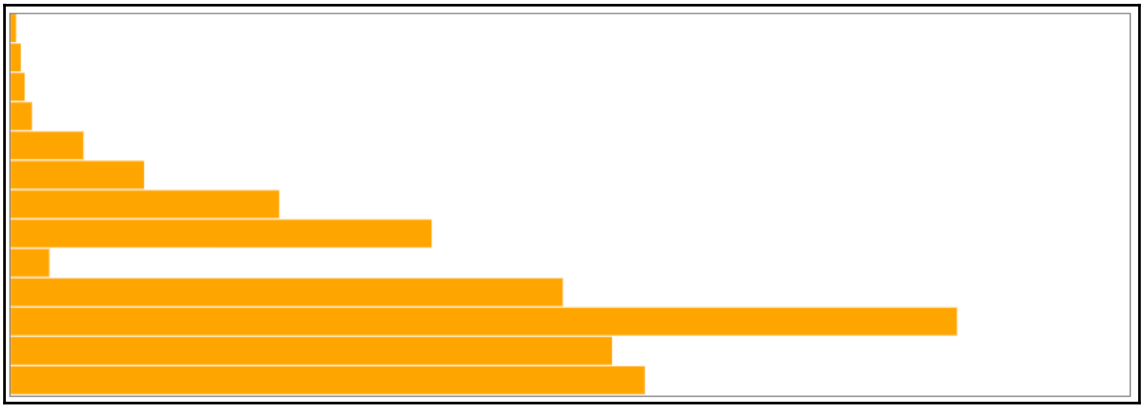

Canvas
Scalable
Vector
Graphics

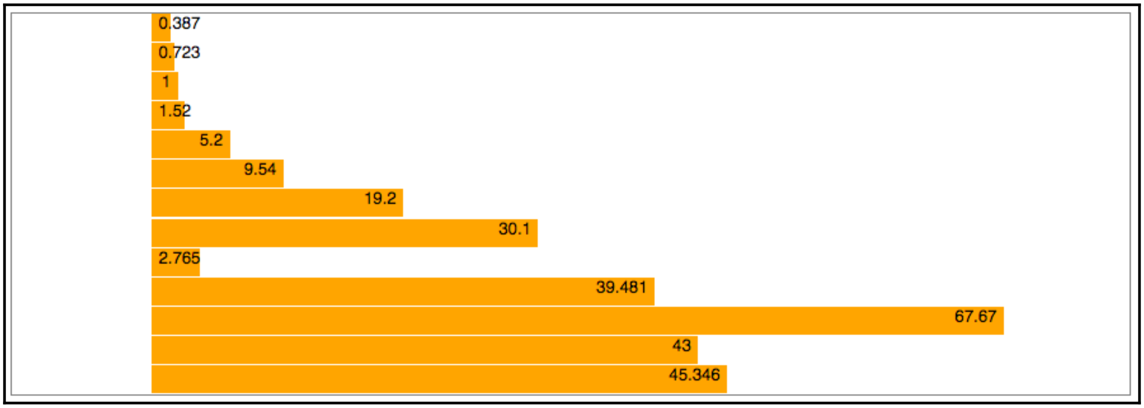
Chapter 3: Quick Start

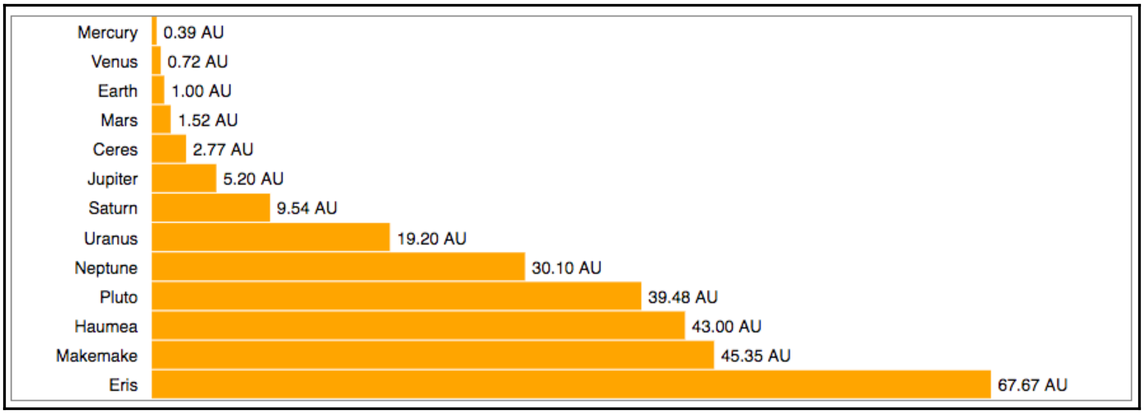


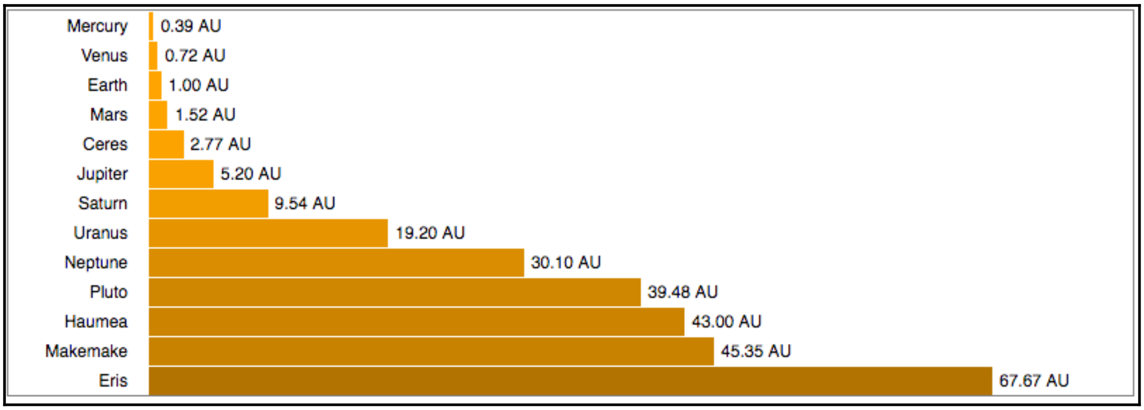


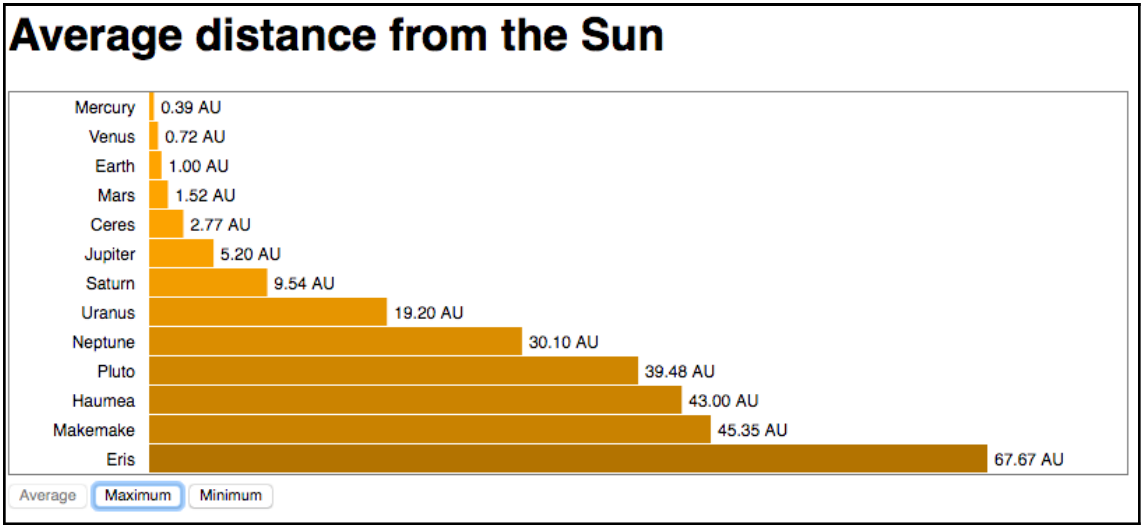

Title
Helder
localhost:63342/DataVizWith...
Item 1: 6
Item 2: 2
Item 3: 5
Item 4: 7
Item 5: 9

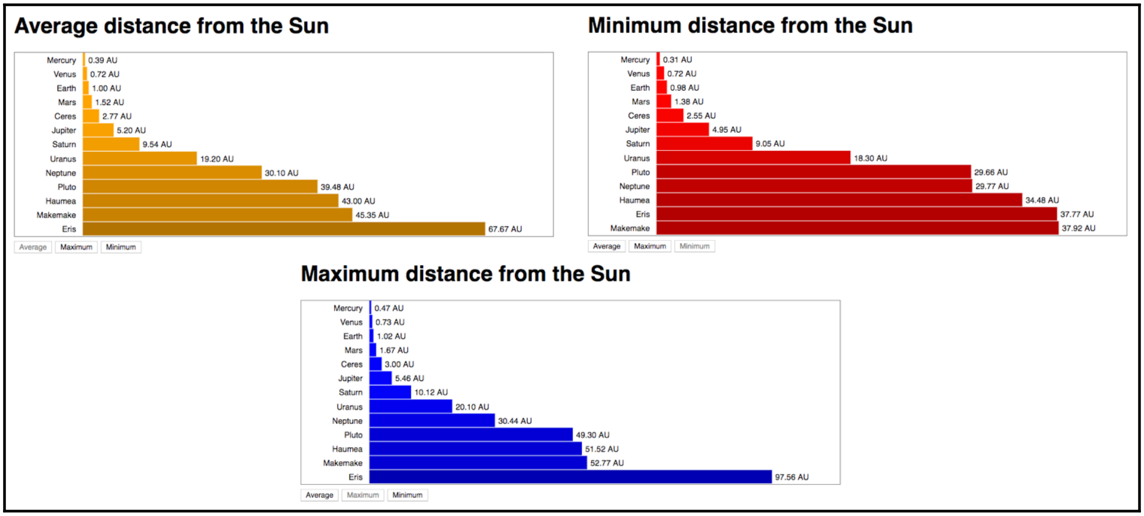


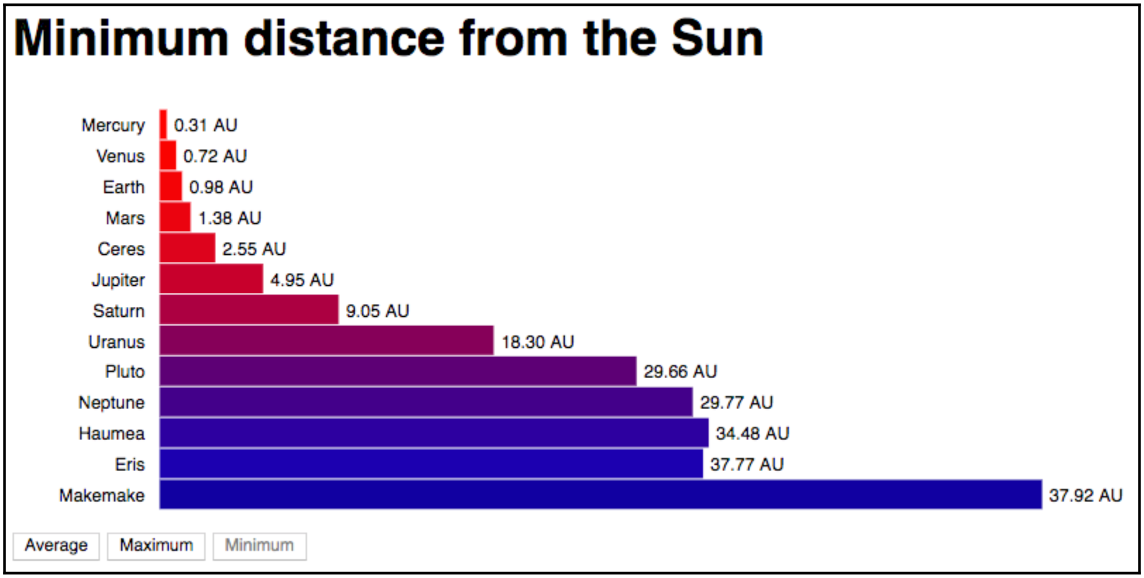

0.387
0.723
1
1.52
5.2
9.54
19.2
30.1
2.765
39.481
67.67
43
45.346


Mercury 0.39 AU
Venus 0.72 AU
Earth 1.00 AU
Mars 1.52 AU
Ceres 2.77 AU
Jupiter 5.20 AU
Saturn 9.54 AU
Uranus 19.20 AU
Neptune 30.10 AU
Pluto 39.48 AU
Haumea 43.00 AU
Makemake 45.35 AU
Eris 67.67 AU

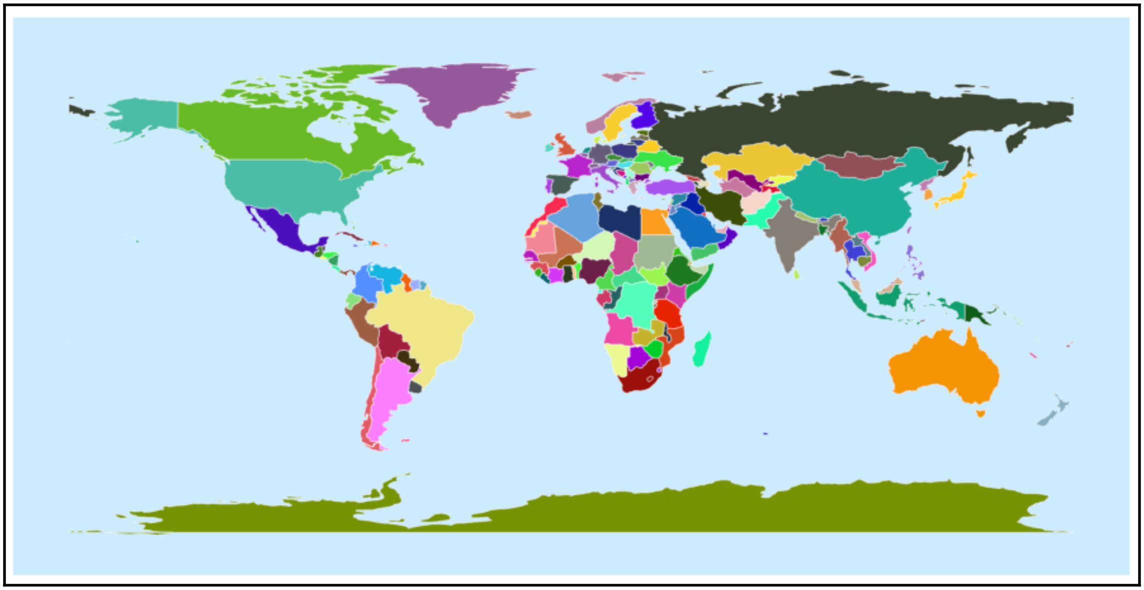

Mercury 0.39 AU
Venus 0.72 AU
Earth 1.00 AU
Mars 1.52 AU
Ceres 2.77 AU
Jupiter 5.20 AU
Saturn 9.54 AU
Uranus 19.20 AU
Neptune 30.10 AU
Pluto 39.48 AU
Haumea 43.00 AU
Makemake 45.35 AU
Eris 67.67 AU


Average distance from the Sun
Mercury 0.39 AU
Venus 0.72 AU
Earth 1.00 AU
Mars 1.52 AU
Ceres 2.77 AU
Jupiter 5.20 AU
Saturn 9.54 AU
Uranus 19.20 AU
Neptune 30.10 AU
Pluto 39.48 AU
Haumea 43.00 AU
Makemake 45.35 AU
Eris 67.67 AU
Average
Maximum
Minimum

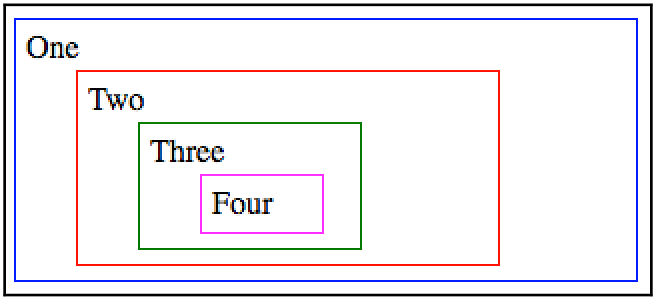

Average distance from the Sun
Mercury 0.39 AU
Venus 0.72 AU
Earth 1.00 AU
Mars 1.52 AU
Ceres 2.77 AU
Jupiter 5.20 AU
Saturn 9.54 AU
Uranus 19.20 AU
Neptune 30.10 AU
Pluto 39.48 AU
Haumea 43.00 AU
Makemake 45.35 AU
Eris 67.67 AU
Average
Maximum
Minimum
Minimum distance from the Sun
Mercury 0.31 AU
Venus 0.72 AU
Earth 0.98 AU
Mars 1.38 AU
Ceres 2.55 AU
Jupiter 4.95 AU
Saturn 9.05 AU
Uranus 18.30 AU
Pluto 29.66 AU
Neptune 29.77 AU
Haumea 34.48 AU
Eris 37.77 AU
Makemake 37.92 AU
Average
Maximum
Minimum
Maximum distance from the Sun
Mercury 0.47 AU
Venus 0.73 AU
Earth 1.02 AU
Mars 1.67 AU
Ceres 3.00 AU
Jupiter 5.46 AU
Saturn 10.12 AU
Uranus 20.10 AU
Neptune 30.44 AU
Pluto 49.30 AU
Haumea 51.52 AU
Makemake 52.77 AU
Eris 97.56 AU
Average
Maximum
Minimum

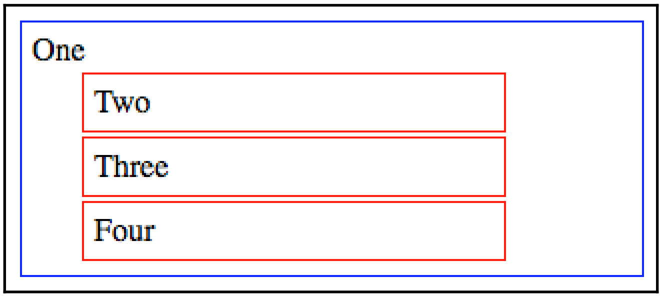


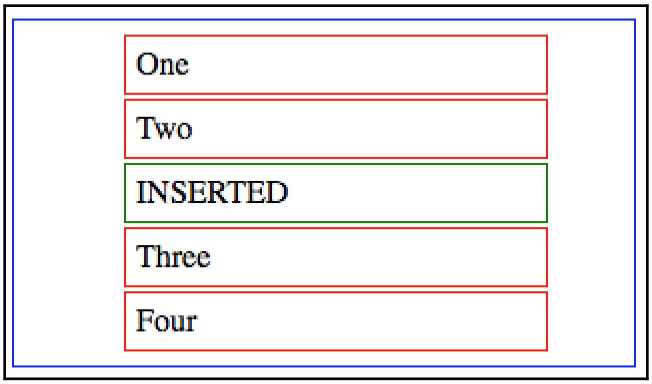
Minimum distance from the Sun
Mercury
0.31 AU
Venus
0.72 AU
Earth
0.98 AU
Mars
1.38 AU
Ceres
2.55 AU
Jupiter
4.95 AU
Saturn
9.05 AU
Uranus
18.30 AU
Pluto
29.66 AU
Neptune
29.77 AU
Haumea
34.48 AU
Eris
37.77 AU
Makemake
37.92 AU
Average
Maximum
Minimum



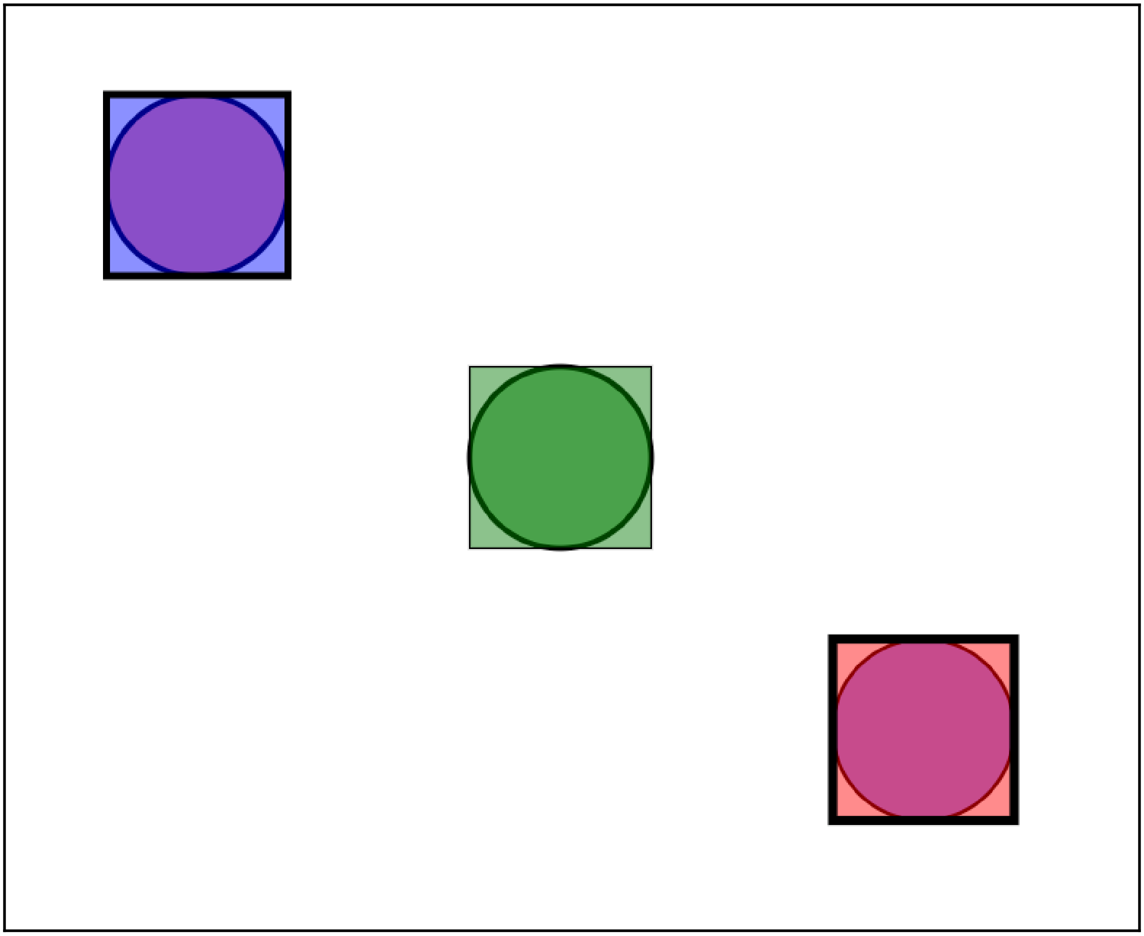
Chapter 4: Data Binding

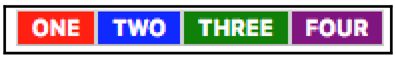


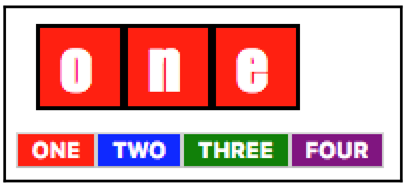


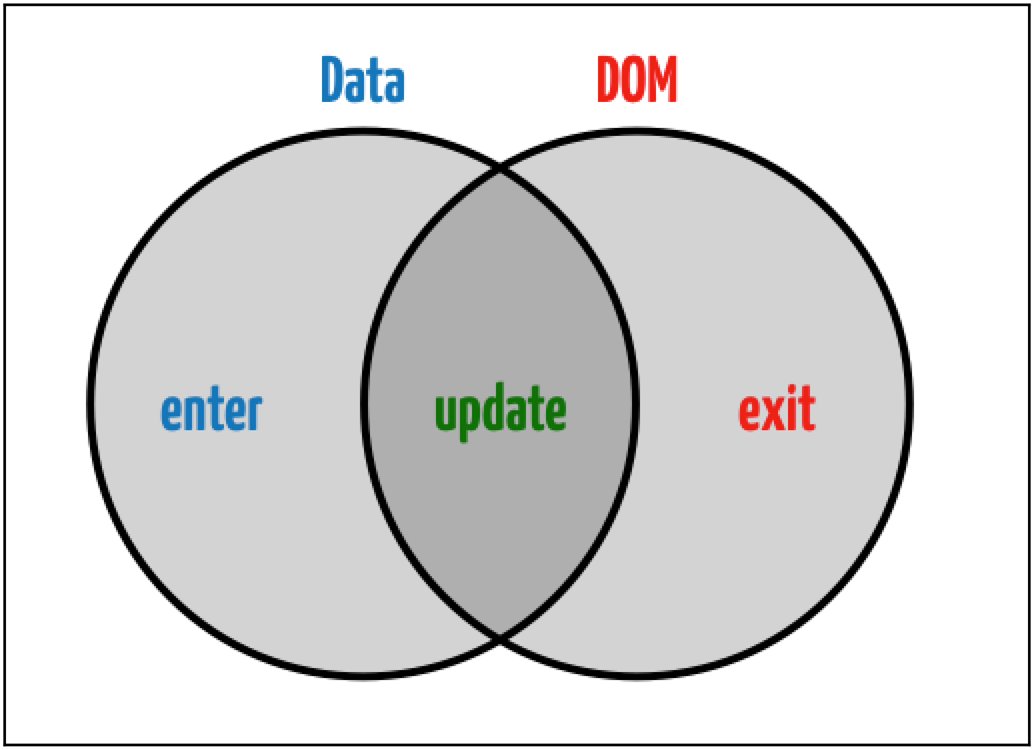


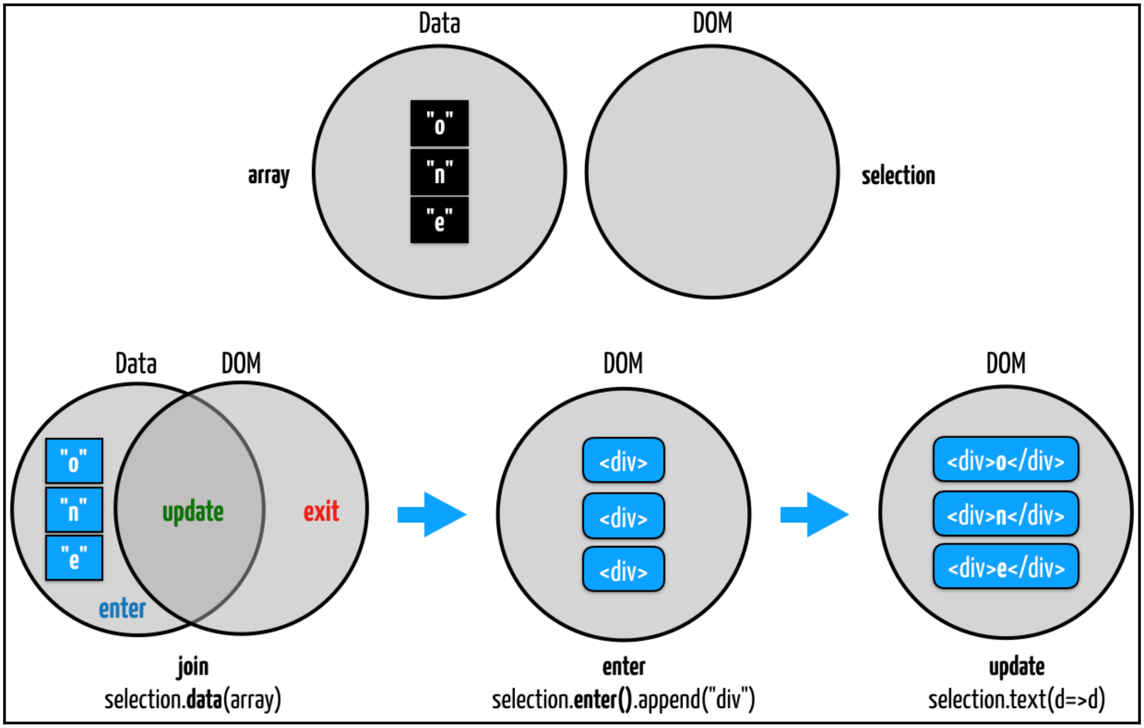
Initial data [1,2,3,4,5]	key function: **index** update: [2,4,6] enter: [2,4,6]	key function: **datum** update: [2,4,6] enter: [2,4,6]
datum: 1, i:0	updated: 2,j:0	datum: 1, i:0
datum: 2, i:1	updated: 4,j:1	updated: 2,j:0
datum: 3, i:2	updated: 6,j:2	datum: 3, i:2
datum: 4, i:3	datum: 4, i:3	updated: 4,j:1
datum: 5, i:4	datum: 5, i:4	datum: 5, i:4
	appended: 2,j:5	appended: 6,j:2
	appended: 4,j:6	
	appended: 6,j:7	

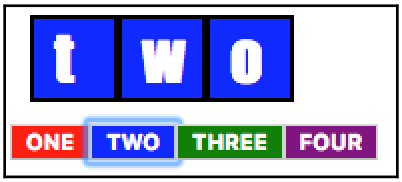


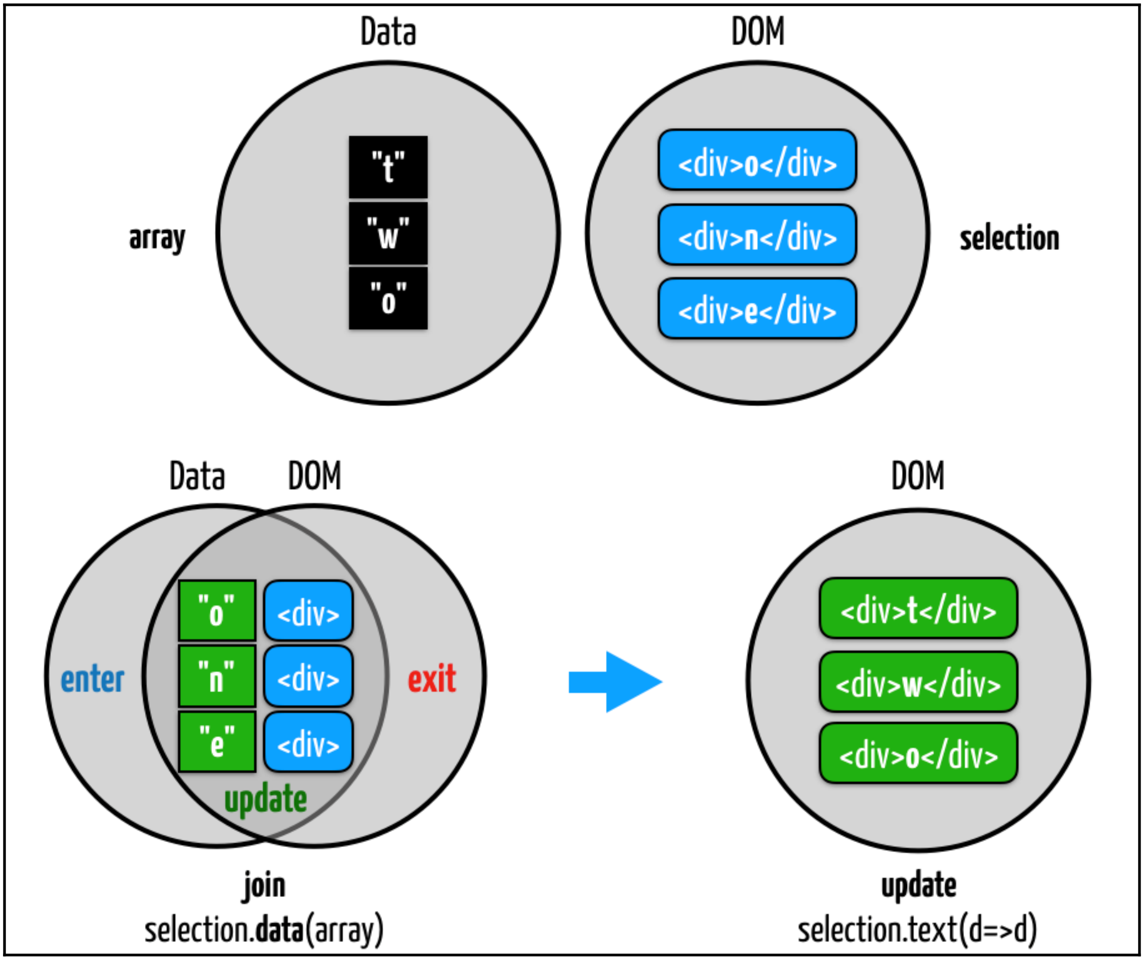

ONE
TWO
THREE
FOUR


o n e
ONE
TWO
THREE
FOUR

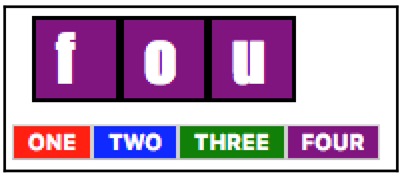

Data
DOM
enter
update
exit


Data
DOM
array
"o"
"n"
"e"
selection
Data
DOM
"o"
"n"
"e"
update
exit
enter
DOM
<div>
<div>
<div>
DOM
<div>o</div>
<div>n</div>
<div>e</div>
join
selection.data(array)
enter
selection.enter().append("div")
update
selection.text(d=>d)

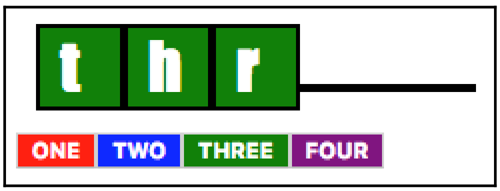

t
w
o
ONE
TWO
THREE
FOUR



Data
DOM
array
"t"
"w"
"o"
<div>o</div>
<div>n</div>
<div>e</div>
selection
Data
DOM
enter
"o"
"n"
"e"
<div>
<div>
<div>
exit
update
DOM
<div>t</div>
<div>w</div>
<div>o</div>
join
selection.data(array)
update
selection.text(d=>d)

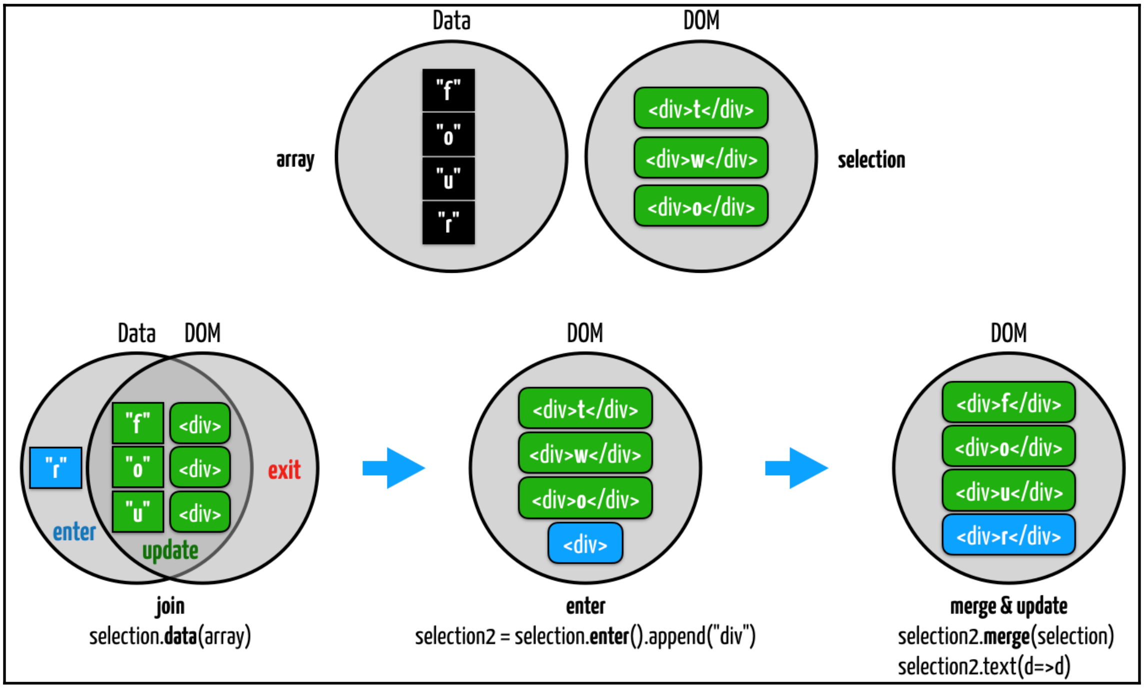


f
o
u
ONE
TWO
THREE
FOUR

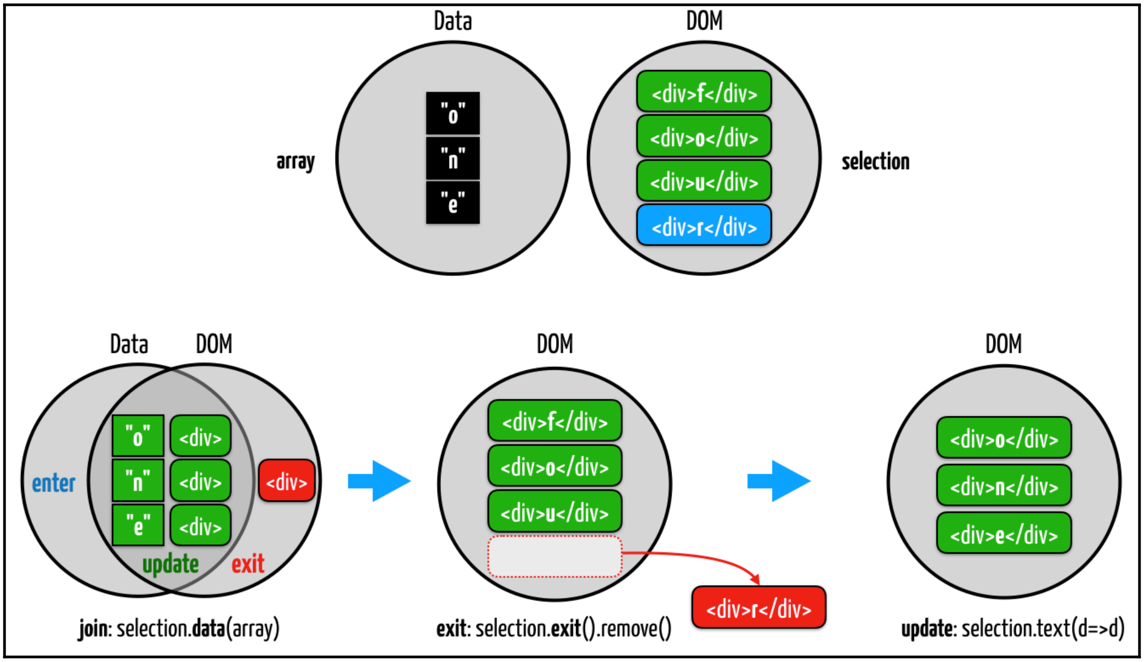


t
h
r
ONE
TWO
THREE
FOUR

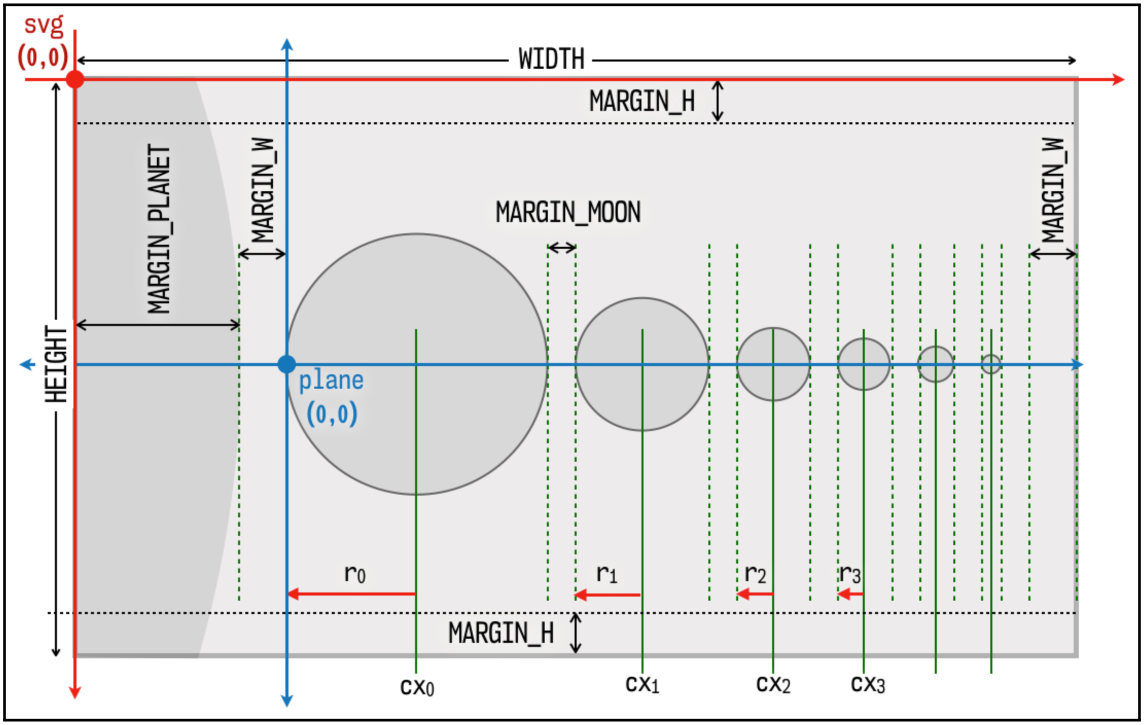

t h r e e
ONE
TWO
THREE
FOUR

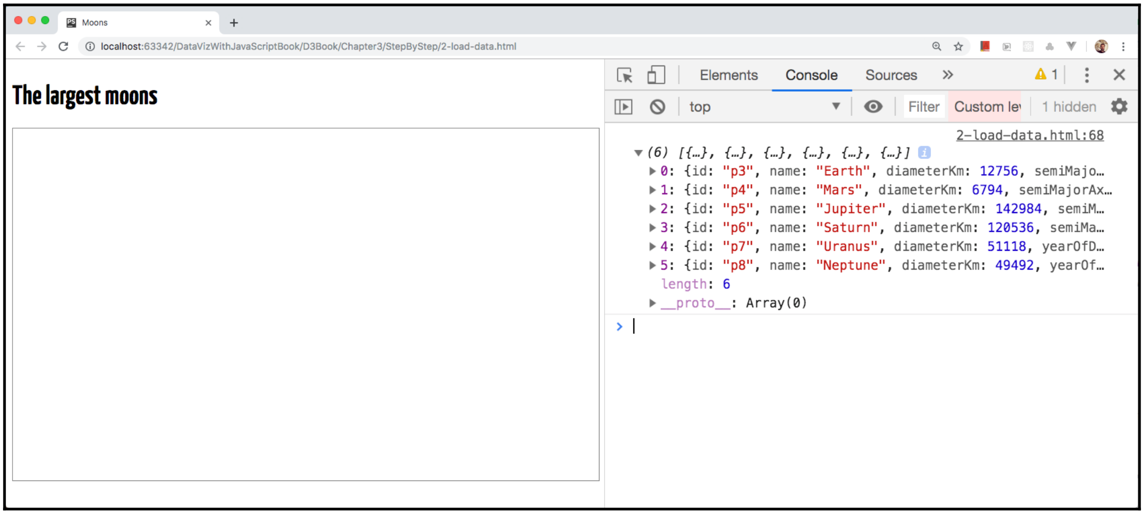

Data
DOM
array
"f"
"o"
"u"
"r"
<div>t</div>
<div>w</div>
<div>o</div>
selection
Data
DOM
"r"
"f"
"o"
"u"
<div>
<div>
<div>
exit
enter
update
DOM
<div>t</div>
<div>w</div>
<div>o</div>
<div>
DOM
<div>f</div>
<div>o</div>
<div>u</div>
<div>r</div>
join
selection.data(array)
enter
selection2 = selection.enter().append("div")
merge & update
selection2.merge(selection)
selection2.text(d=>d)


o n e r
ONE
TWO
THREE
FOUR

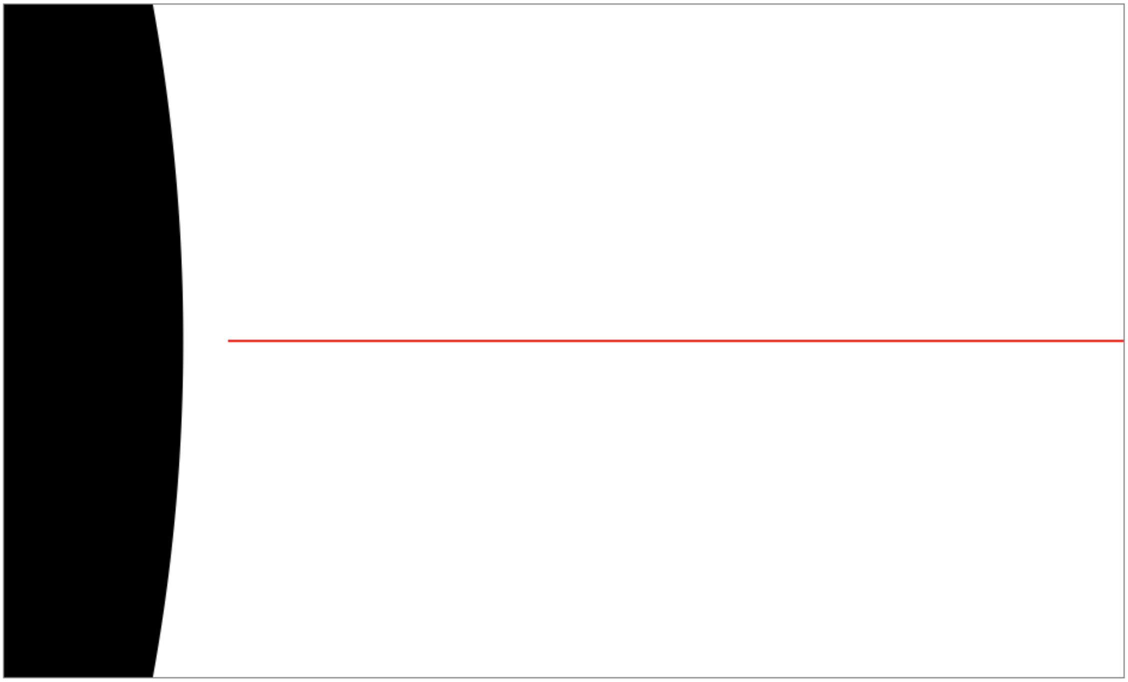


Data
DOM
array
"o"
"n"
"e"
<div>f</div>
<div>o</div>
<div>u</div>
<div>r</div>
selection
Data
DOM
enter
"o"
"n"
"e"
<div>
<div>
<div>
<div>
update
exit
join: selection.data(array)
DOM
<div>f</div>
<div>o</div>
<div>u</div>
<div>r</div>
exit: selection.exit().remove()
DOM
<div>o</div>
<div>n</div>
<div>e</div>
update: selection.text(d=>d)

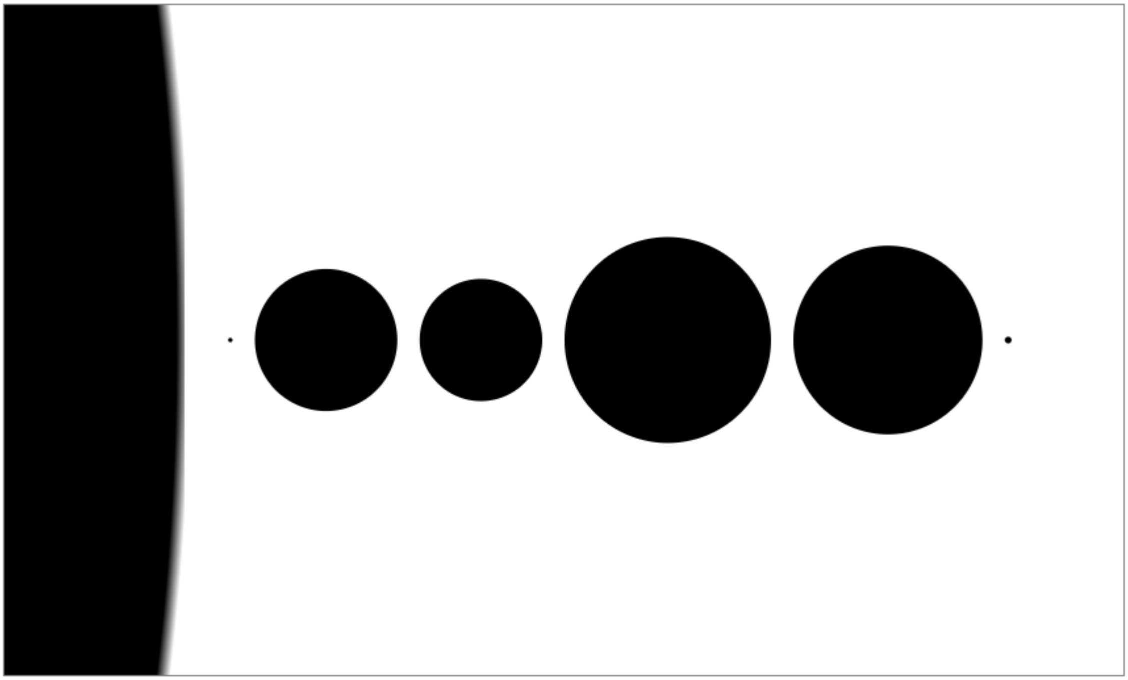

svg
(0,0)
WIDTH
MARGIN_H
MARGIN_PLANET
MARGIN_W
MARGIN_MOON
MARGIN_W
HEIGHT
plane
(0,0)
r0
r1
r2
r3
MARGIN_H
cx0
cx1
cx2
cx3


Moons
localhost:63342/DataVizWithJavaScriptBook/D3Book/Chapter3/StepByStep/2-load-data.html
The largest moons
Elements
Console
Sources
top
Filter
Custom le
1 hidden
2-load-data.html:68
(6) [{…}, {…}, {…}, {…}, {…}, {…}]
0: {id: "p3", name: "Earth", diameterKm: 12756, semiMajo…
1: {id: "p4", name: "Mars", diameterKm: 6794, semiMajorAx…
2: {id: "p5", name: "Jupiter", diameterKm: 142984, semiM…
3: {id: "p6", name: "Saturn", diameterKm: 120536, semiMa…
4: {id: "p7", name: "Uranus", diameterKm: 51118, yearOfD…
5: {id: "p8", name: "Neptune", diameterKm: 49492, yearOf…
length: 6
__proto__: Array(0)

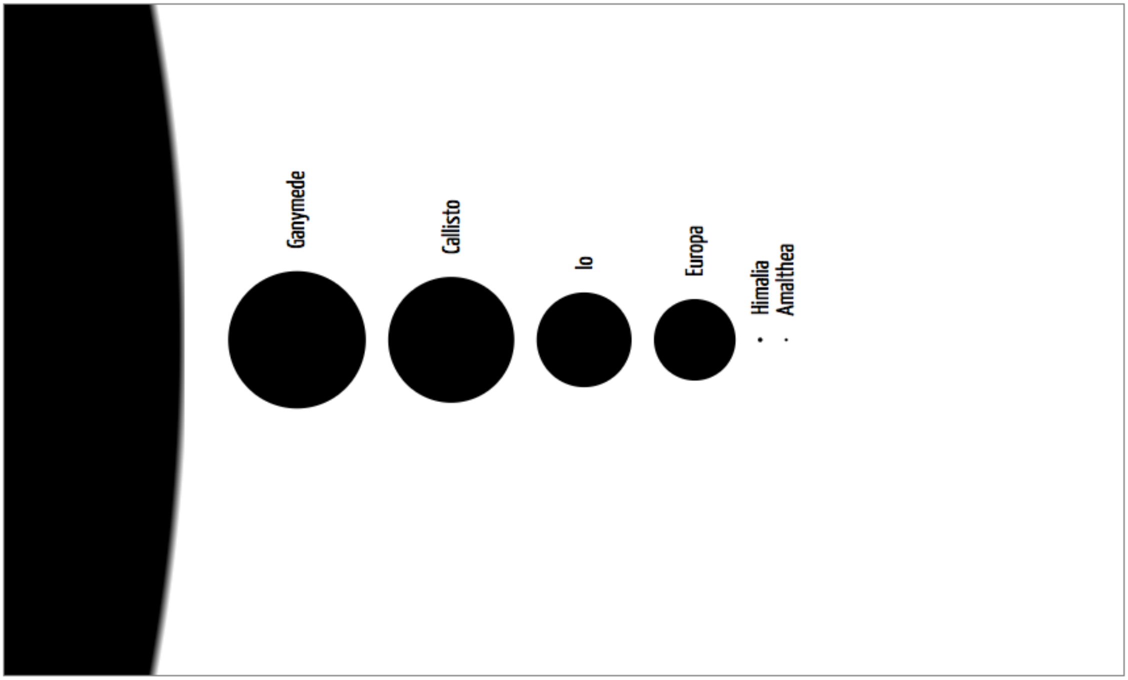
The largest moons of Jupiter



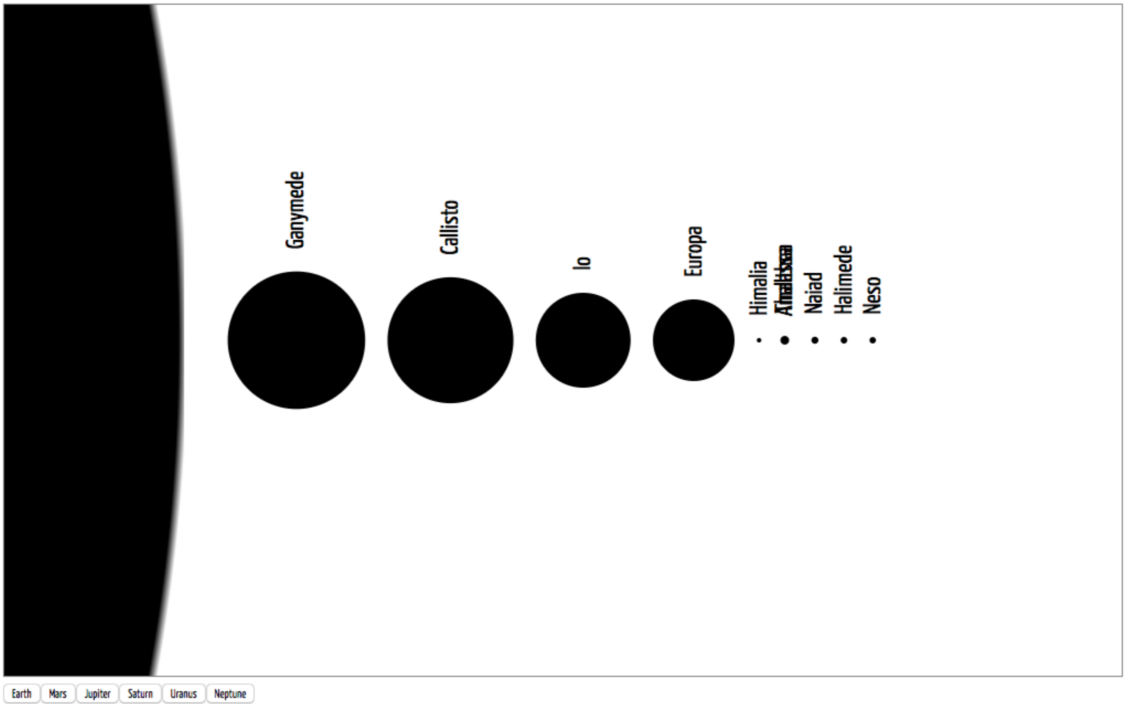
The largest moons of Jupiter



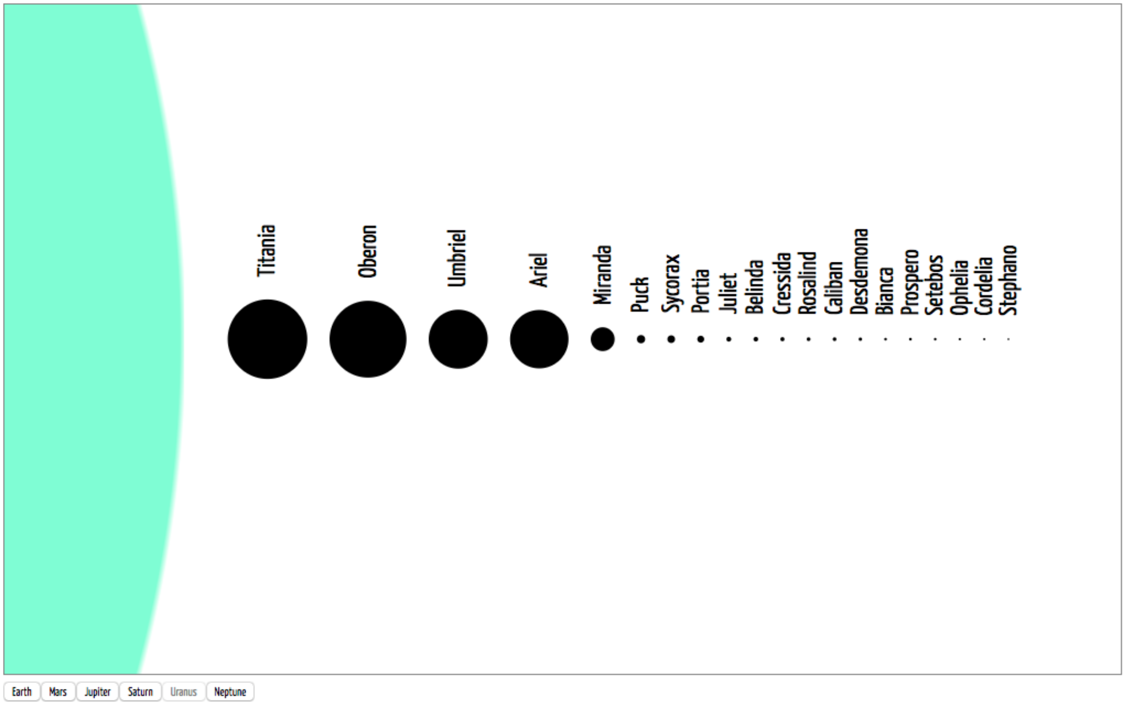
The largest moons of Jupiter

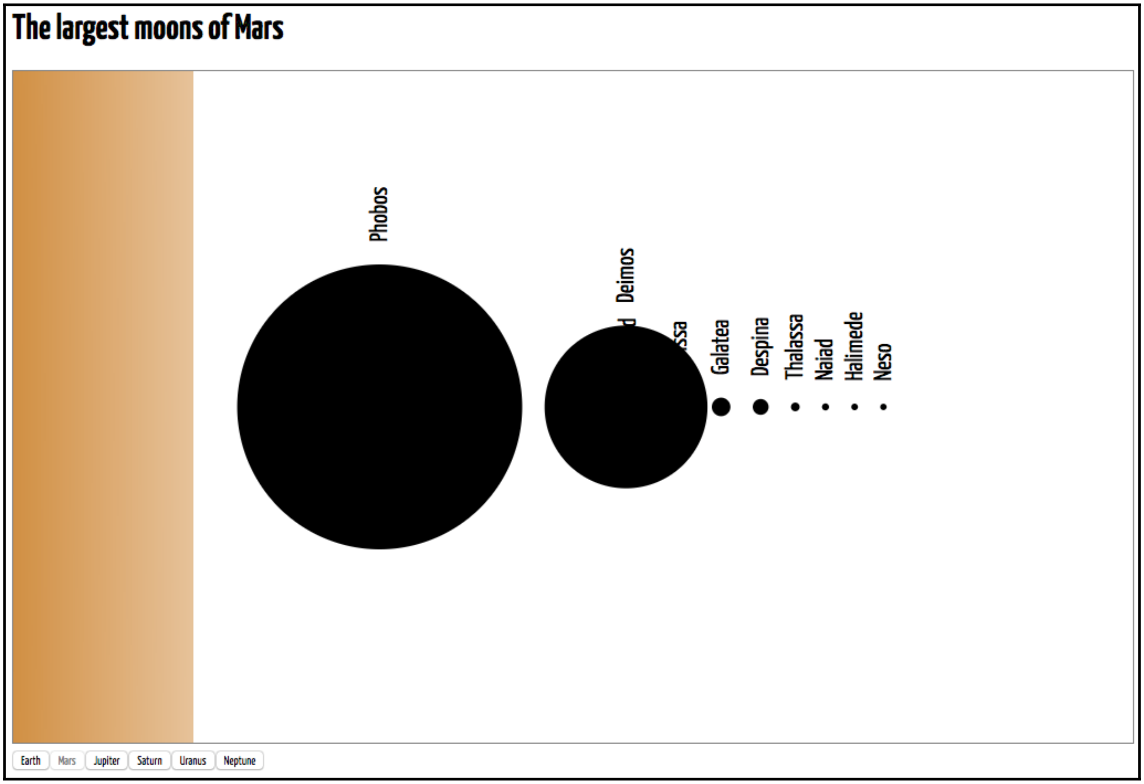


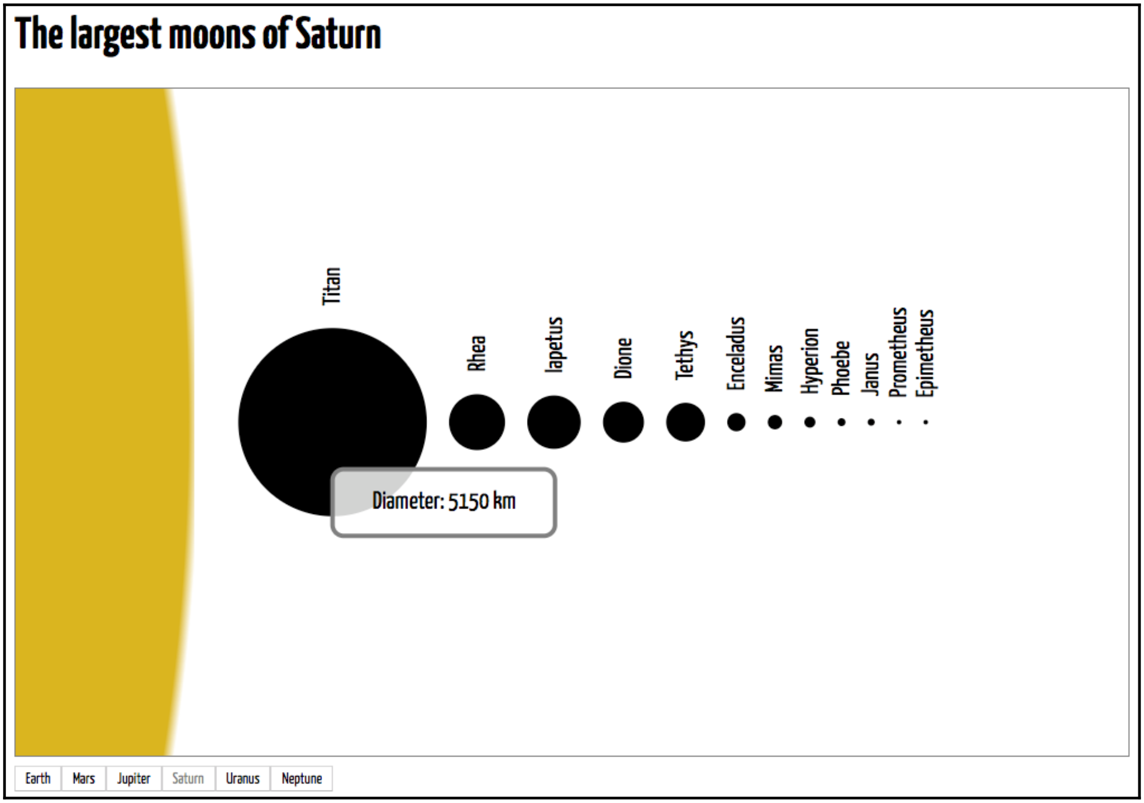
The largest moons of Neptune



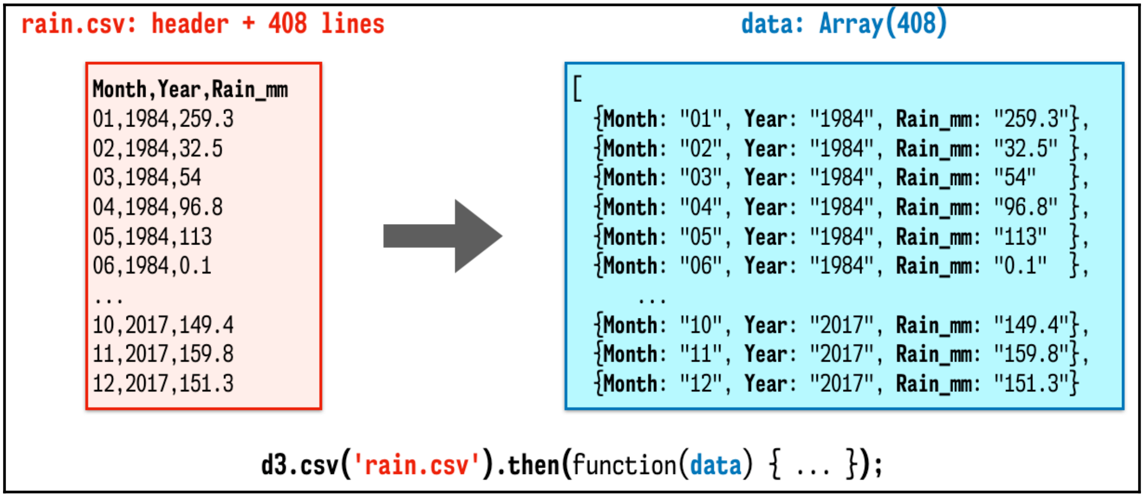
The largest moons of Uranus

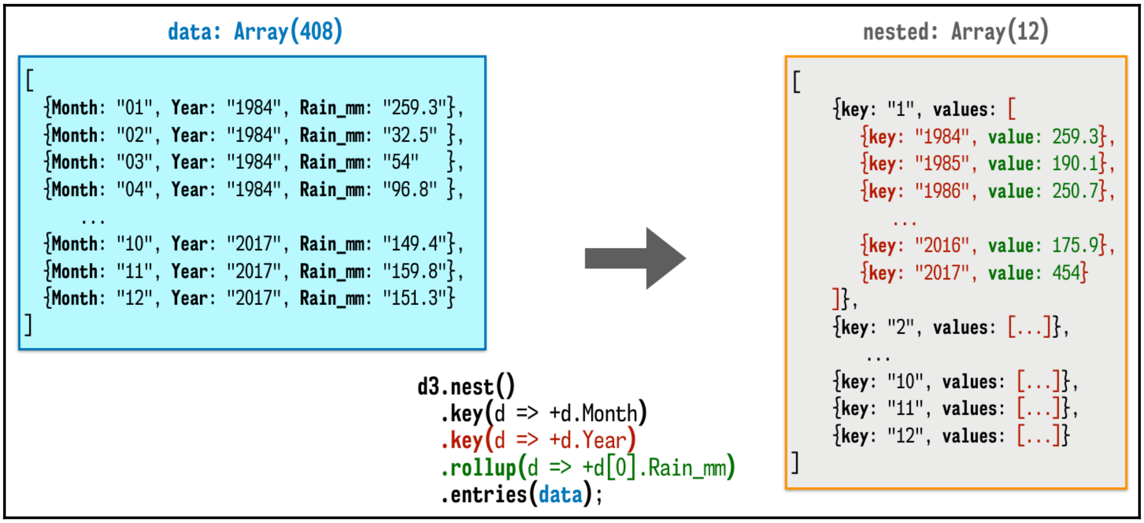


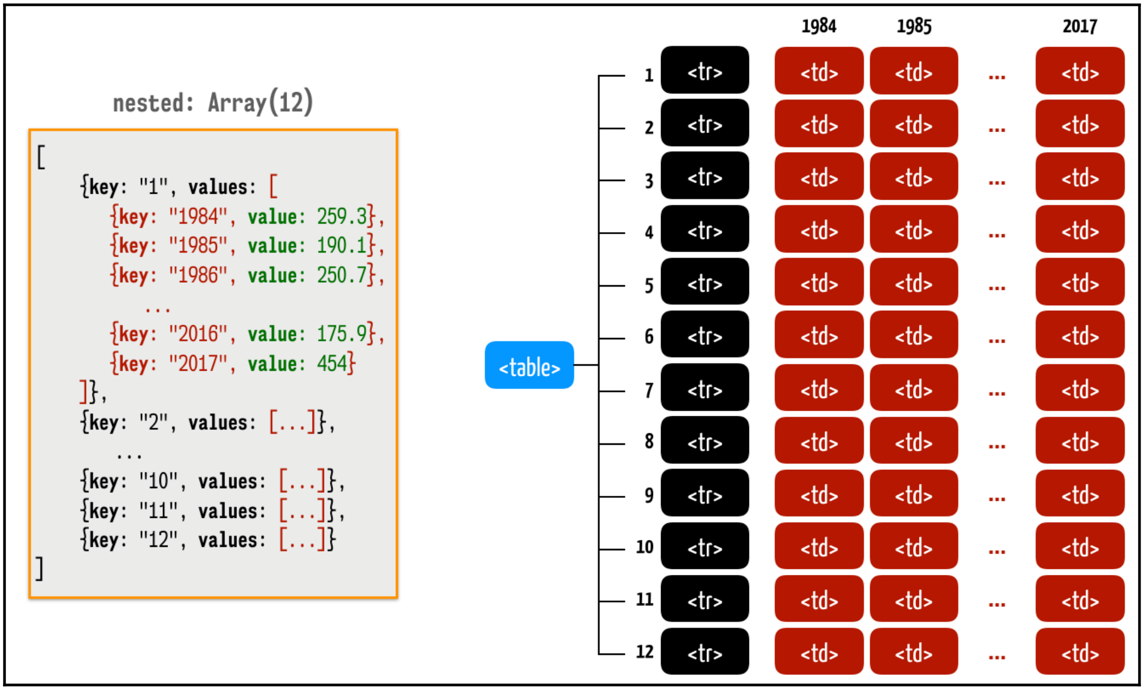

The largest moons of Mars
Phobos
Deimos
Galatea
Despina
Thalassa
Naiad
Halimede
Neso
Earth
Mars
Jupiter
Saturn
Uranus
Neptune


The largest moons of Saturn
Titan
Rhea
Iapetus
Dione
Tethys
Enceladus
Mimas
Hyperion
Phoebe
Janus
Prometheus
Epimetheus
Diameter: 5150 km
Earth
Mars
Jupiter
Saturn
Uranus
Neptune

Chapter 5: Manipulating Data and Formatting







259.3	190.1	250.7	442.3	293.4	359.2	280.7	270.7	203.4	248.7	231.9	379	354.9	313.3	200.6	319.1	328.3
32.5	281.4	268.4	177.5	284.7	326.5	152.8	358	184	265.6	170.8	445.5	305.6	181.1	394.2	369.8	246.6
54	166.9	230.6	178.5	203.6	185.1	228.8	451.3	273.7	92.5	252.3	304.1	396.8	76	252.9	187.3	100.1
96.8	31.4	84.9	91.5	173.4	96.2	77.1	169	59.3	118.4	99.9	64.9	81.7	71.6	92.8	35.8	6.3
113	107.2	84.6	241.2	210.8	30.4	52.8	34.2	71.4	101.2	40.7	74.4	33.1	75	130	35.5	9.3
0.1	14.2	2.4	195.9	58.8	44.4	39.2	85.8	18.6	57.5	31.9	58.7	54.1	122.5	16.1	84.3	12.2
26.7	0.6	28.1	10.7	2.2	144.7	121	26.2	39.1	15.2	26.1	47.1	7.2	10	10.1	25.6	65.6
110.6	21.6	97.2	17.4	2.6	32.3	49.6	39.3	25.8	52.1	3.1	16.7	31.6	22.3	42.5	1.4	86.3
162.5	110.1	34.7	59.2	34.7	82.2	96.1	65.6	180.6	206.7	1.3	41.5	178.9	140.9	95	80.1	111.1
31.1	11.2	43.9	89.9	170.1	65.4	117.6	153.9	177.5	148.5	126.8	229.9	154	96.1	216.4	62	59
140	86	155.6	96.1	85.7	82.4	76	48	213.8	96.2	122.8	97.7	93.9	220.1	38.6	101.4	186
139.6	176.3	383.2	146.3	279.6	121.7	124.7	220.6	201	180.3	311.1	202	331.4	255.4	241.2	74.6	249.7

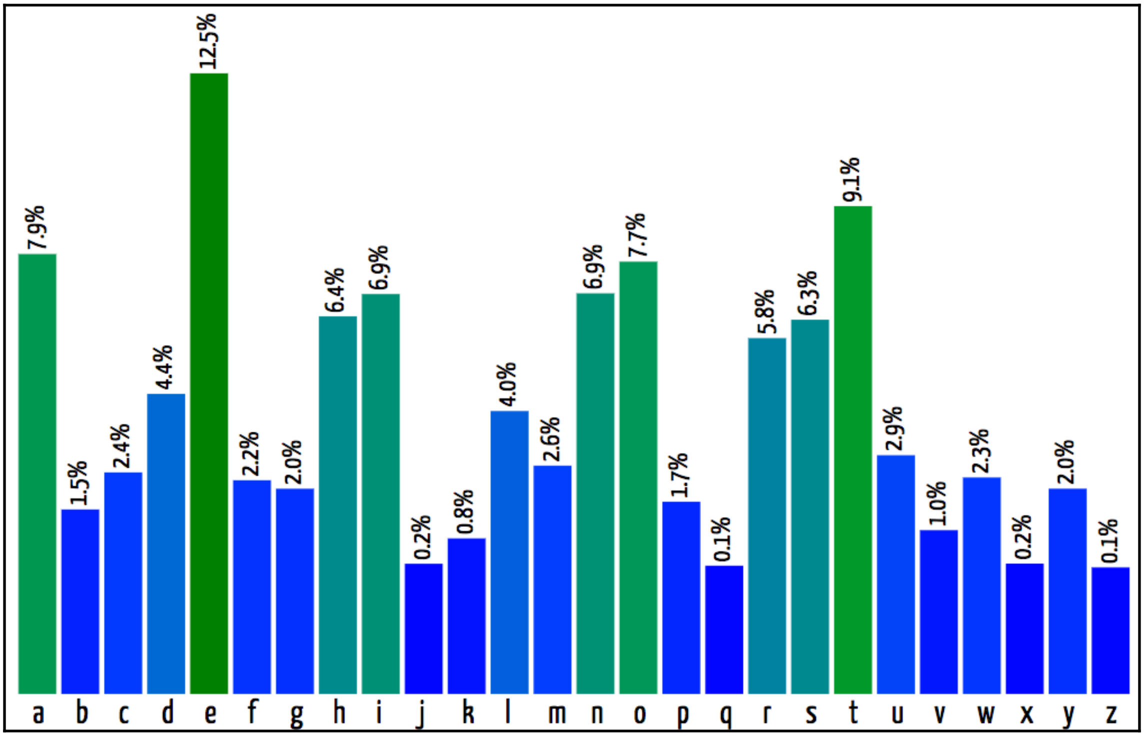
144.2	284.8	317.2	284.5	312.1	348	213.5	318.5	351.8	480.5	493.7	332.6	169.2	237.9	156.2	175.9	454
290.1	172.5	109.4	335.6	99.9	166	285.9	236.3	200.2	296.5	311.5	224.2	278	197.6	273	275.1	127.3
138.5	326.5	126.5	130.7	286.6	607.9	185.4	180.8	125.7	184.5	164	187.6	174.5	226.9	332.7	256.6	160.4
32.9	53.5	45.6	123.4	133.2	51.1	124.8	96.6	69.6	124.5	133	155.9	70.7	79.7	108.1	2.4	143.1
86.1	93	33.1	60.1	199	15	59	80.3	50	65.2	30.4	82.7	42.5	46	50.7	105.7	153.4
24.5	1.3	16	66.8	30.4	24.2	30.7	78.2	49.8	13.1	81.6	233.7	143.2	9.7	20.3	206.8	102.9
41	22.9	19	97.4	13.7	71	148.3	0	179.7	93.5	4.5	74.7	90.9	21.4	65.1	6.4	0.8
32.8	46.4	25.3	2.7	9.5	5.8	0	78.5	102.8	0.4	46.3	0.3	7.7	29.6	31.6	82.4	60.5
88.5	54.8	34.5	9.3	138.8	77.7	15.7	43.9	192.2	104.8	7.4	19.2	81.3	58.7	201.7	22.2	11.1
204.3	124.9	126.7	97.4	172.1	100.4	109.3	161.4	154.5	70.1	149.6	128.3	126.6	25.2	92.1	104.1	149.4
185.2	226.5	99.3	173.6	106.1	230.7	219.9	165.1	177.3	109.6	141.3	91.6	123.6	117.5	247.2	166.8	159.8
187	235.3	139.8	262.9	228.2	311.5	230.9	220.2	363.7	343.1	136.8	401.9	83.1	203.1	318.1	165.4	151.3

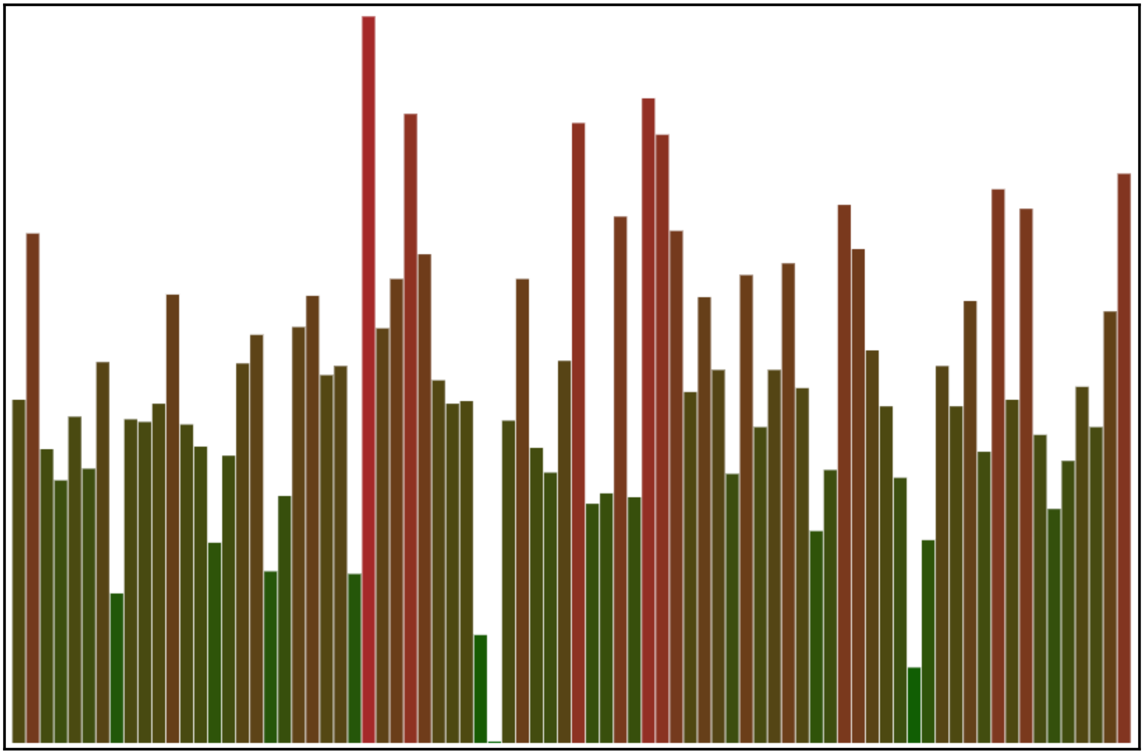
Rainfall in São Paulo, Brazil (in mm) from 1984 to 2017

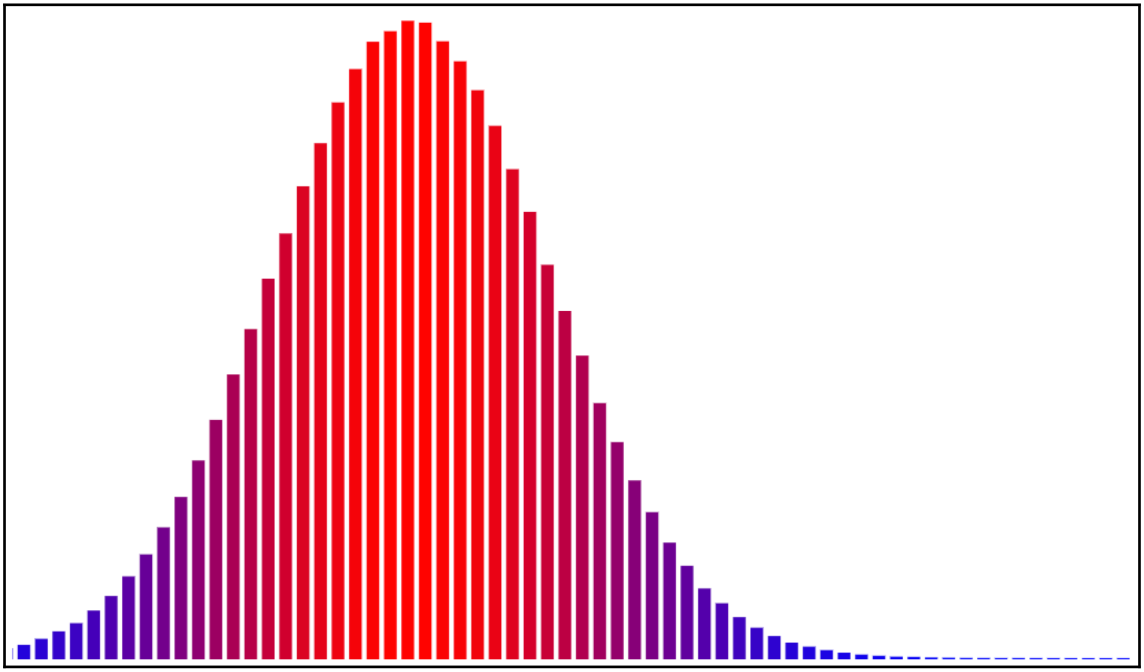
	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000
Jan	259.3	190.1	250.7	442.3	293.4	359.2	280.7	270.7	203.4	248.7	231.9	379	354.9	313.3	200.6	319.1	328.3
Feb	32.5	281.4	268.4	177.5	284.7	326.5	152.8	358	184	265.6	170.8	445.5	305.6	181.1	394.2	369.8	246.6
Mar	54	166.9	230.6	178.5	203.6	185.1	228.8	451.3	273.7	92.5	252.3	304.1	396.8	76	252.9	187.3	100.1
Apr	96.8	31.4	84.9	91.5	173.4	96.2	77.1	169	59.3	118.4	99.9	64.9	81.7	71.6	92.8	35.8	6.3
May	113	107.2	84.6	241.2	210.8	30.4	52.8	34.2	71.4	101.2	40.7	74.4	33.1	75	130	35.5	9.3
Jun	0.1	14.2	2.4	195.9	58.8	44.4	39.2	85.8	18.6	57.5	31.9	58.7	54.1	122.5	16.1	84.3	12.2
Jul	26.7	0.6	28.1	10.7	2.2	144.7	121	26.2	39.1	15.2	26.1	47.1	7.2	10	10.1	25.6	65.6
Aug	110.6	21.6	97.2	17.4	2.6	32.3	49.6	39.3	25.8	52.1	3.1	16.7	31.6	22.3	42.5	1.4	86.3
Sep	162.5	110.1	34.7	59.2	34.7	82.2	96.1	65.6	180.6	206.7	1.3	41.5	178.9	140.9	95	80.1	111.1
Oct	31.1	11.2	43.9	89.9	170.1	65.4	117.6	153.9	177.5	148.5	126.8	229.9	154	96.1	216.4	62	59
Nov	140	86	155.6	96.1	85.7	82.4	76	48	213.8	96.2	122.8	97.7	93.9	220.1	38.6	101.4	186
Dec	139.6	176.3	383.2	146.3	279.6	121.7	124.7	220.6	201	180.3	311.1	202	331.4	255.4	241.2	74.6	249.7

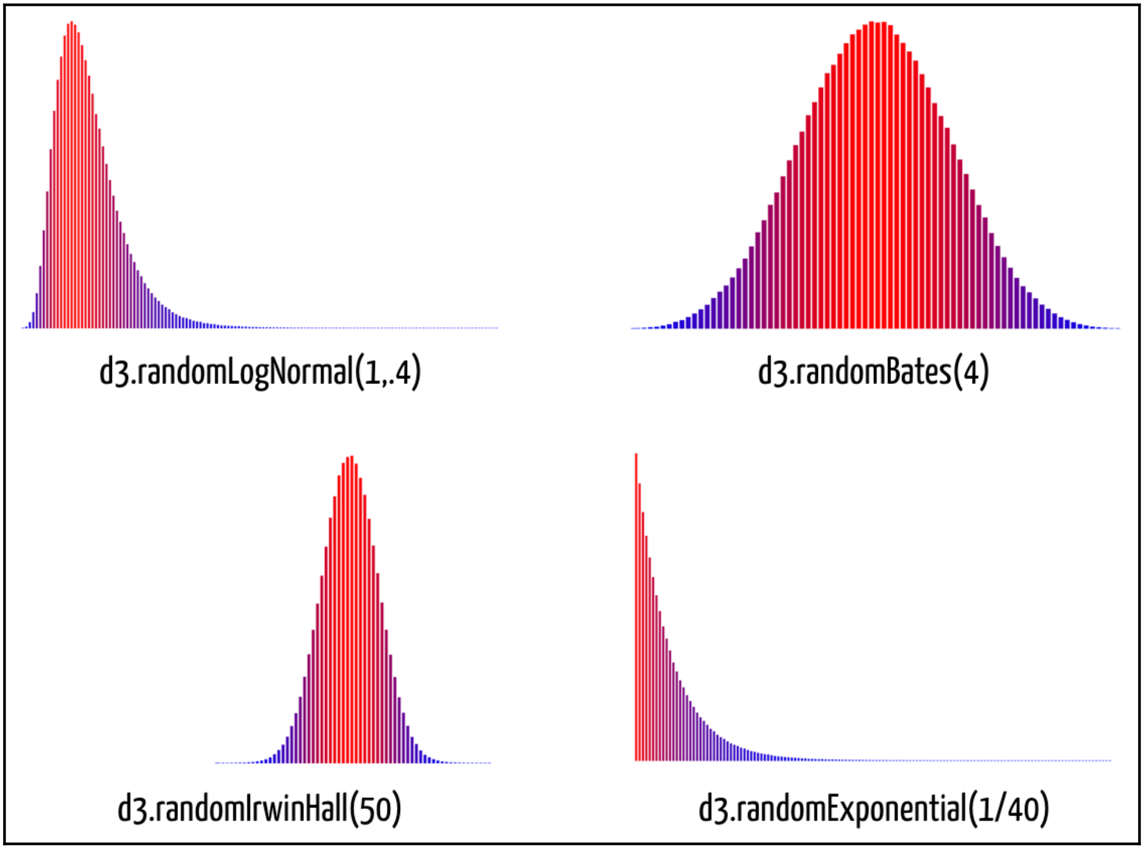
	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
Jan	144.2	284.8	317.2	284.5	312.1	348	213.5	318.5	351.8	480.5	493.7	332.6	169.2	237.9	156.2	175.9	454
Feb	290.1	172.5	109.4	335.6	99.9	166	285.9	236.3	200.2	296.5	311.5	224.2	278	197.6	273	275.1	127.3
Mar	138.5	326.5	126.5	130.7	286.6	607.9	185.4	180.8	125.7	184.5	164	187.6	174.5	226.9	332.7	256.6	160.4
Apr	32.9	53.5	45.6	123.4	133.2	51.1	124.8	96.6	69.6	124.5	133	155.9	70.7	79.7	108.1	2.4	143.1
May	86.1	93	33.1	60.1	199	15	59	80.3	50	65.2	30.4	82.7	42.5	46	50.7	105.7	153.4
Jun	24.5	1.3	16	66.8	30.4	24.2	30.7	78.2	49.8	13.1	81.6	233.7	143.2	9.7	20.3	206.8	102.9
Jul	41	22.9	19	97.4	13.7	71	148.3	0	179.7	93.5	4.5	74.7	90.9	21.4	65.1	6.4	0.8
Aug	32.8	46.4	25.3	2.7	9.5	5.8	0	78.5	102.8	0.4	46.3	0.3	7.7	29.6	31.6	82.4	60.5
Sep	88.5	54.8	34.5	9.3	138.8	77.7	15.7	43.9	192.2	104.8	7.4	19.2	81.3	58.7	201.7	22.2	11.1
Oct	204.3	124.9	126.7	97.4	172.1	100.4	109.3	161.4	154.5	70.1	149.6	128.3	126.6	25.2	92.1	104.1	149.4
Nov	185.2	226.5	99.3	173.6	106.1	230.7	219.9	165.1	177.3	109.6	141.3	91.6	123.6	117.5	247.2	166.8	159.8
Dec	187	235.3	139.8	262.9	228.2	311.5	230.9	220.2	363.7	343.1	136.8	401.9	83.1	203.1	318.1	165.4	151.3

source: inmet.gov.br

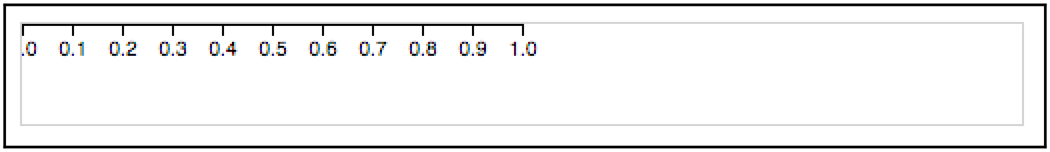


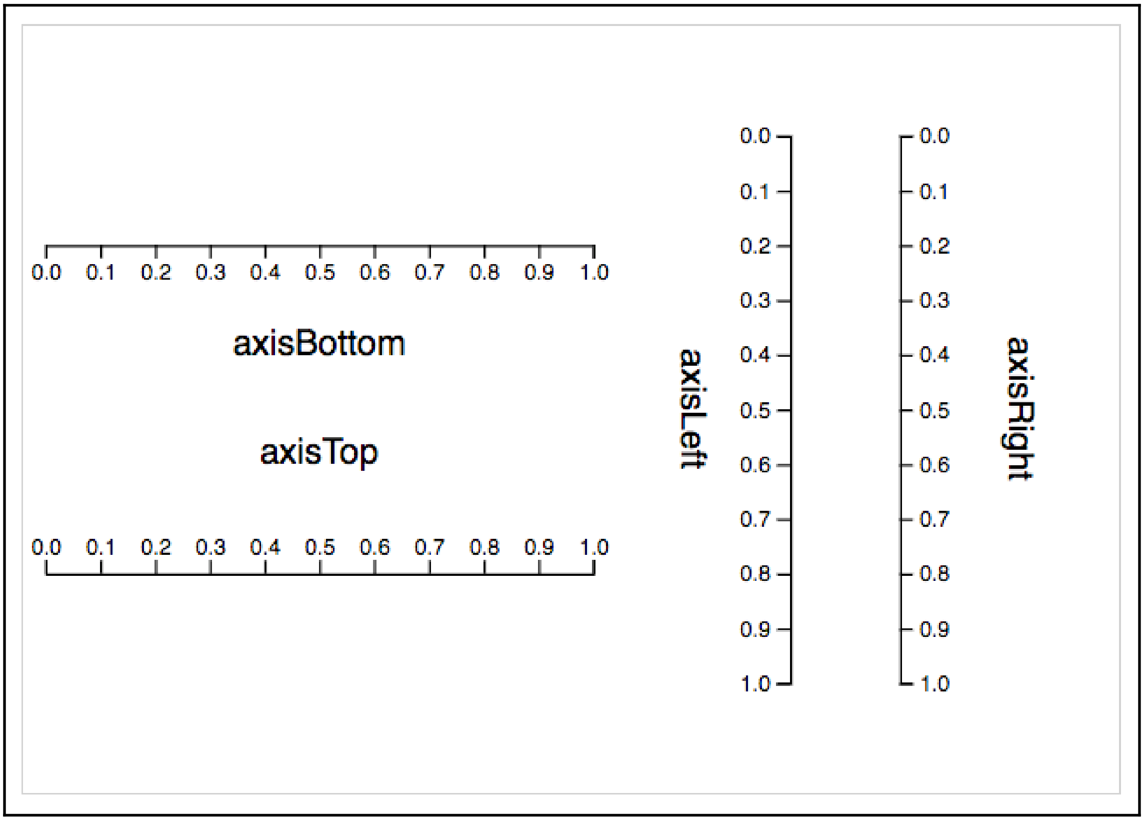


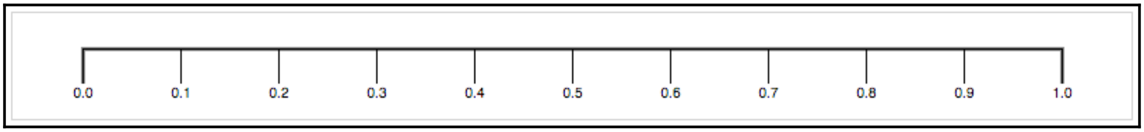


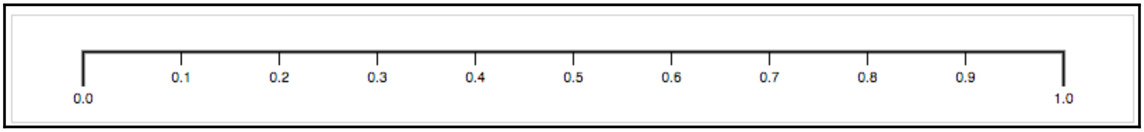

d3.randomLogNormal(1,.4)
d3.randomBates(4)
d3.randomIrwinHall(50)
d3.randomExponential(1/40)

Chapter 6: Scales, Axes, and Colors

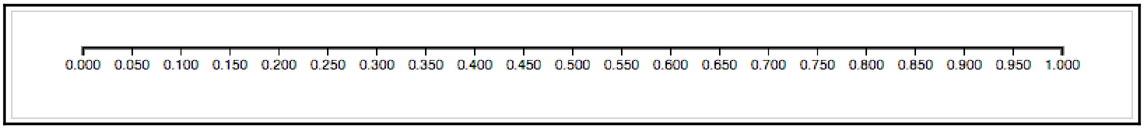


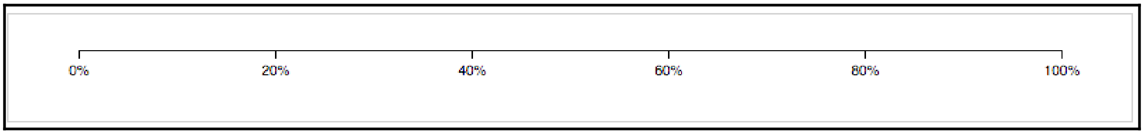


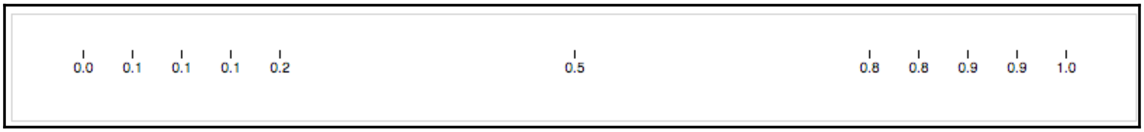


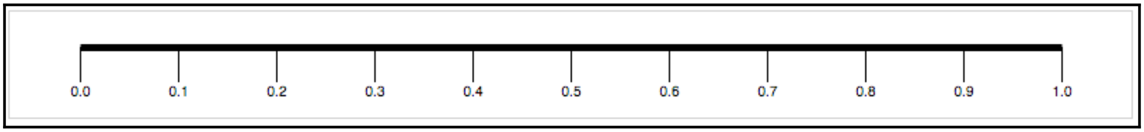

0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

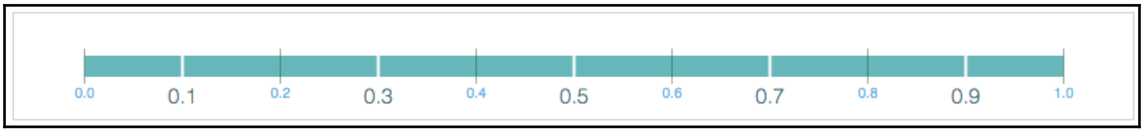

0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

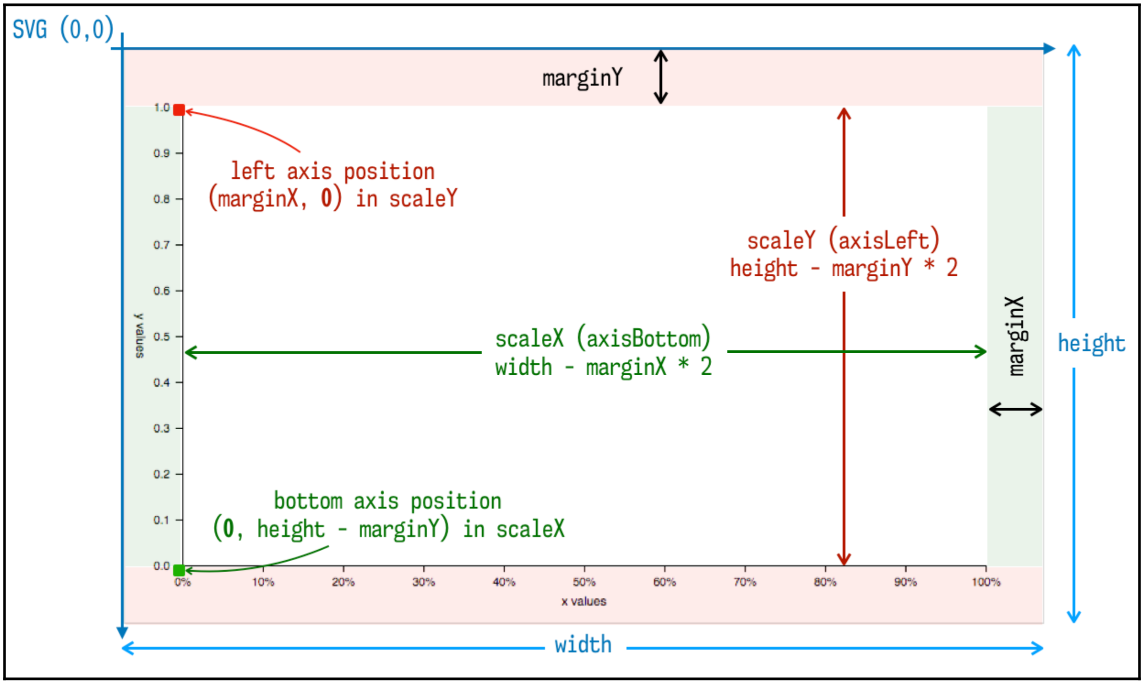

0.000 0.050 0.100 0.150 0.200 0.250 0.300 0.350 0.400 0.450 0.500 0.550 0.600 0.650 0.700 0.750 0.800 0.850 0.900 0.950 1.000


0% 20% 40% 60% 80% 100%

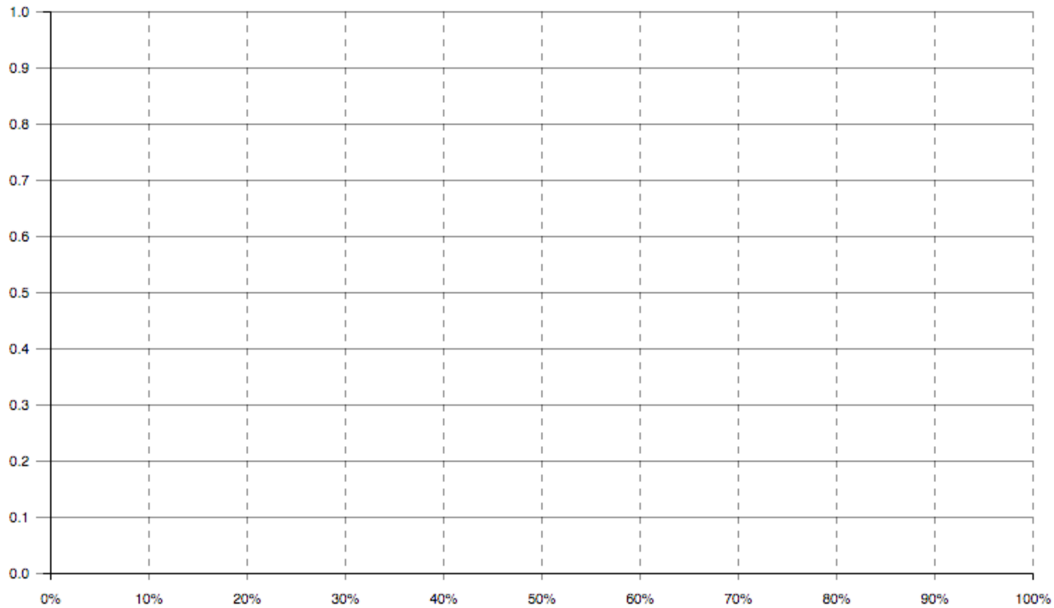

0.0 0.1 0.1 0.1 0.2 0.5 0.8 0.8 0.9 0.9 1.0

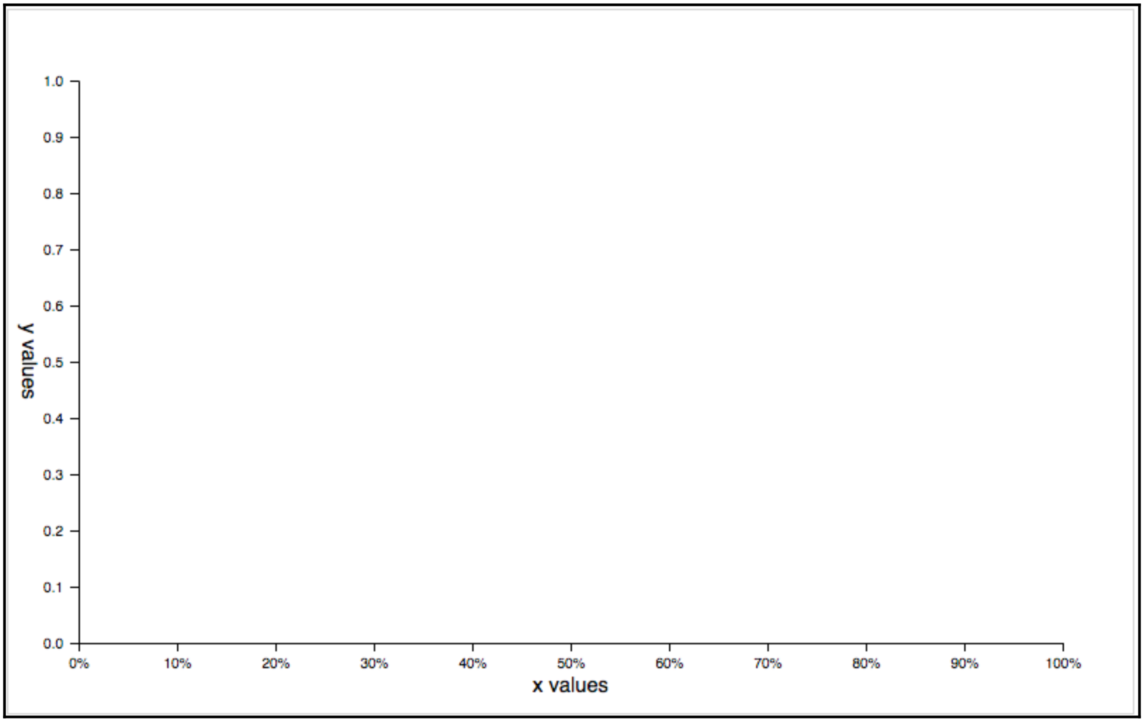

0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

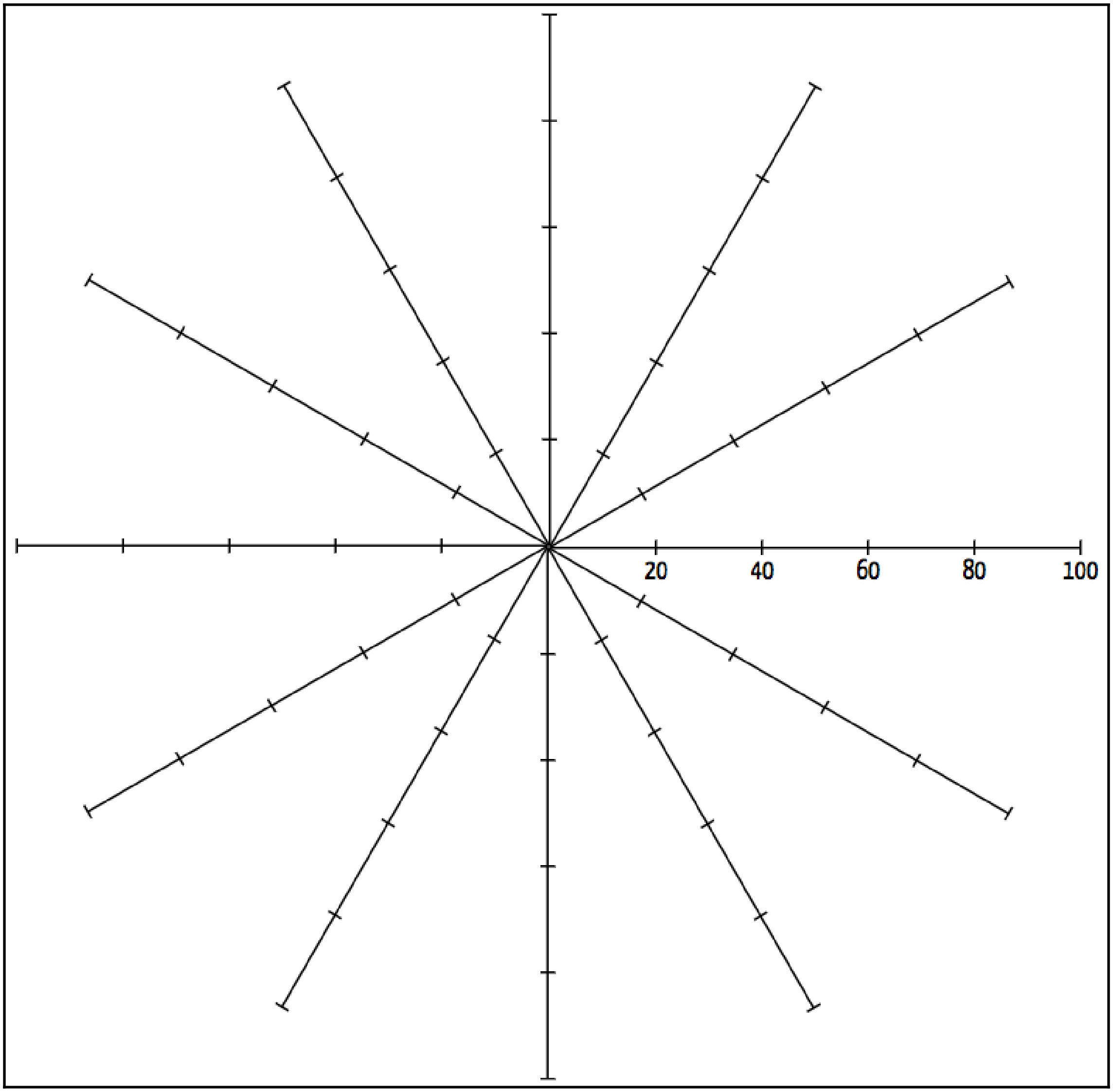

0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

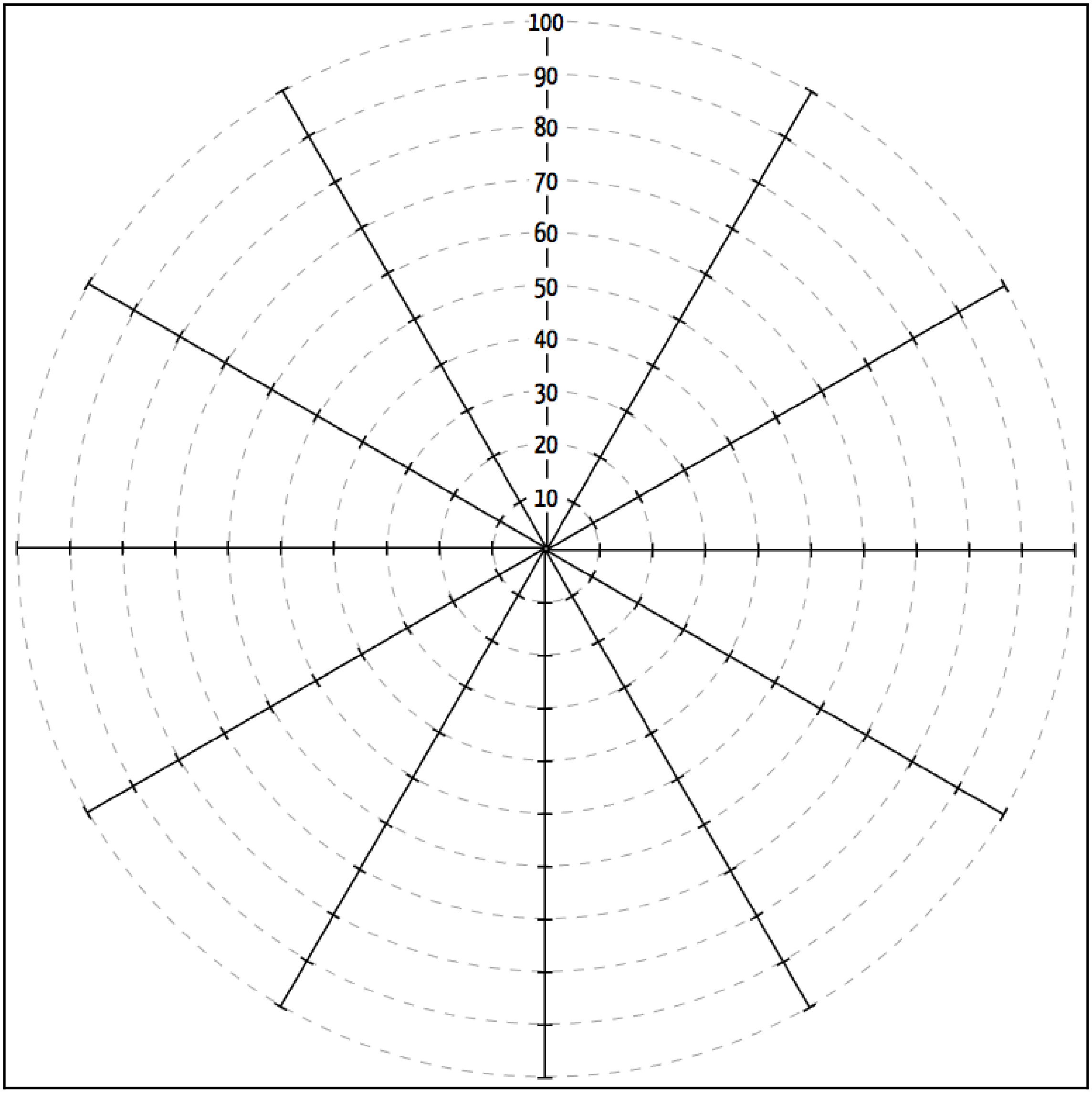


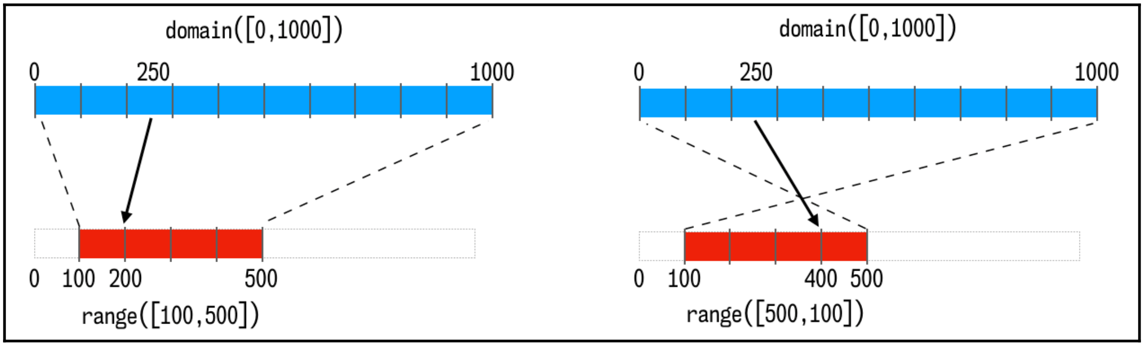
SVG (0,0)
marginY
left axis position
(marginX, 0) in scaleY
scaleY (axisLeft)
height - marginY * 2
scaleX (axisBottom)
width - marginX * 2
marginX
height
bottom axis position
(0, height - marginY) in scaleX
y values
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
x values
width

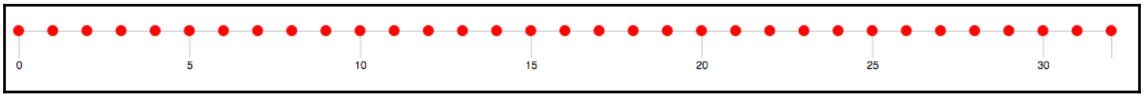

1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%

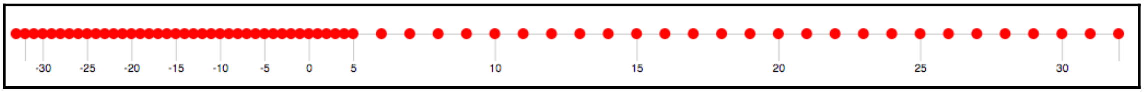

1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
y values
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
x values

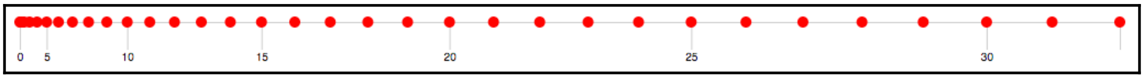

20
40
60
80
100

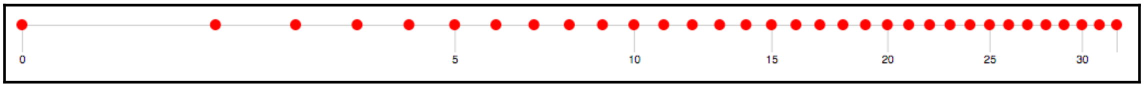

100
90
80
70
60
50
40
30
20
10

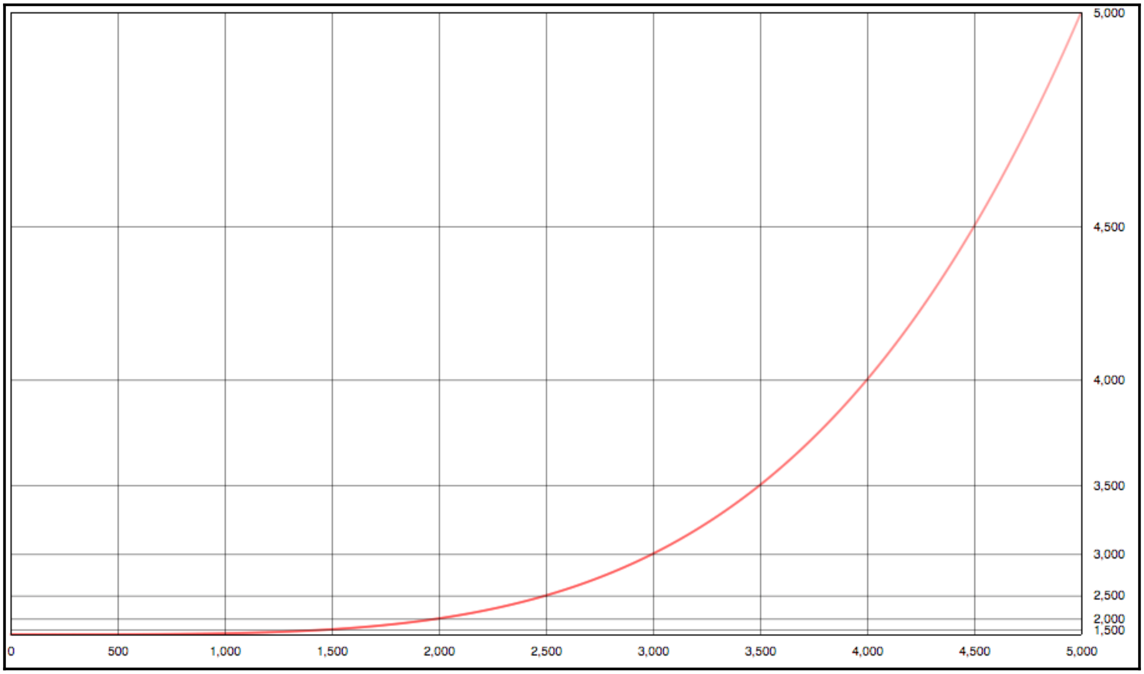

domain([0,1000])
0
250
1000
0
100
200
500
range([100,500])
domain([0,1000])
0
250
1000
0
100
400
500
range([500,100])

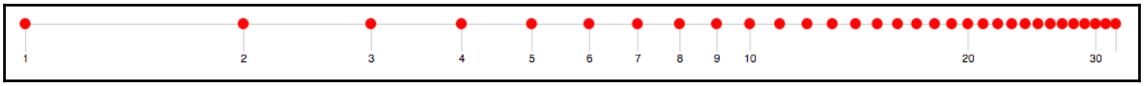

0 5 10 15 20 25 30

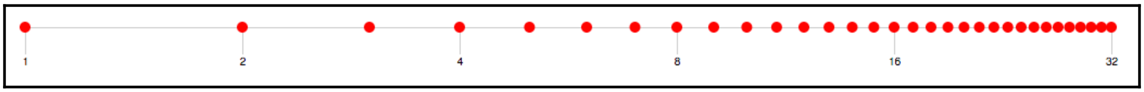

-30 -25 -20 -15 -10 -5 0 5 10 15 20 25 30

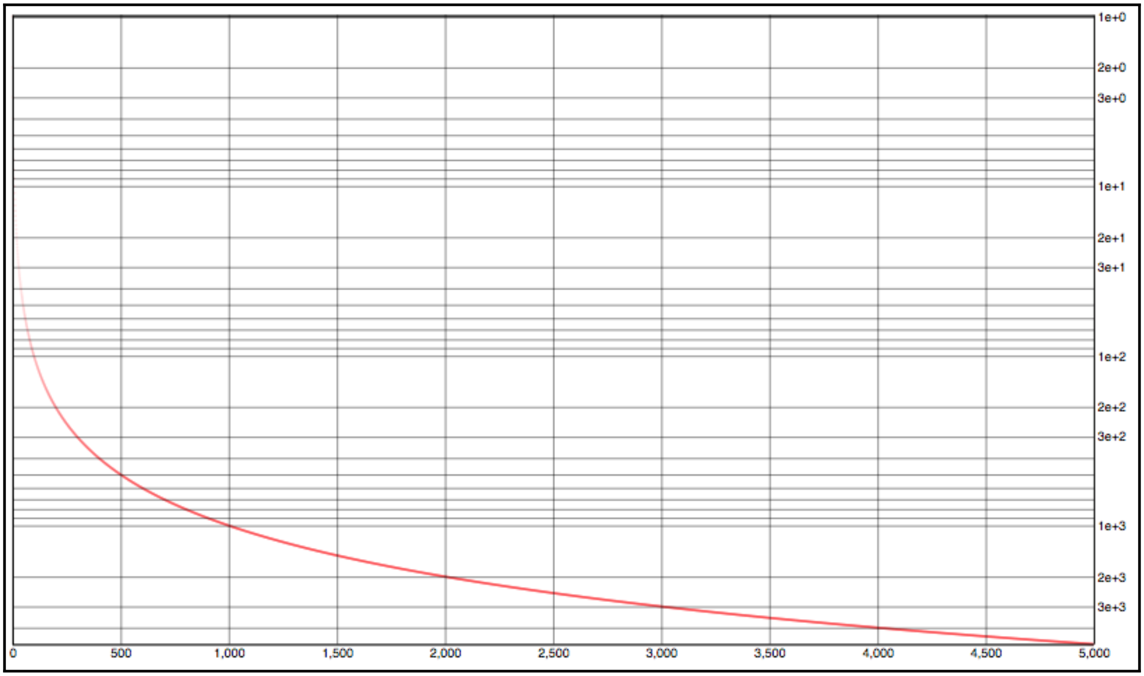

0 5 10 15 20 25 30

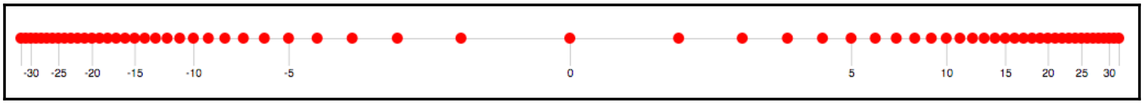

0 5 10 15 20 25 30

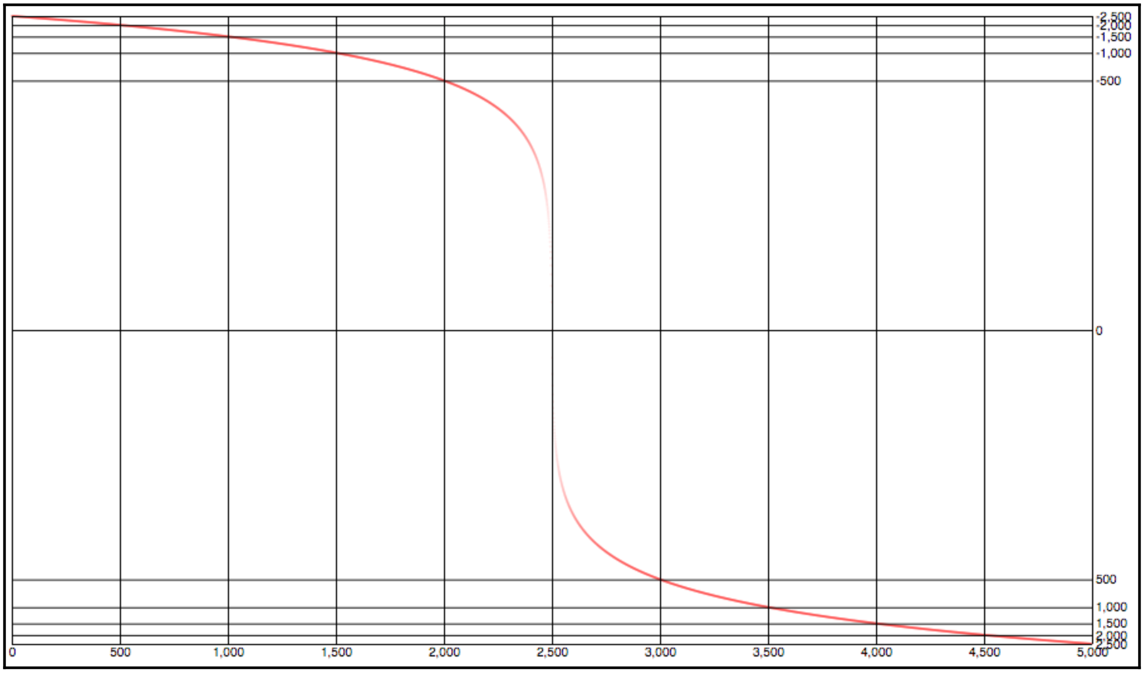

5,000
4,500
4,000
3,500
3,000
2,500
2,000
1,500
0
500
1,000
1,500
2,000
2,500
3,000
3,500
4,000
4,500
5,000

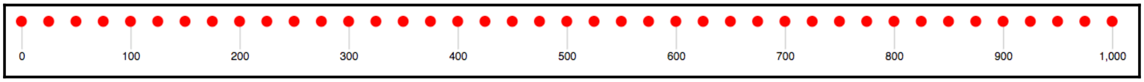

1
2
3
4
5
6
7
8
9
10
20
30

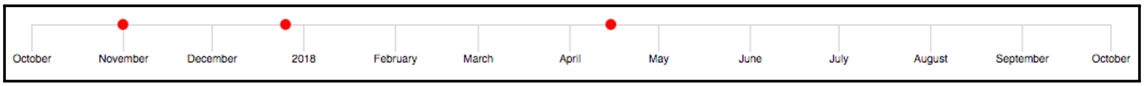

1
2
4
8
16
32

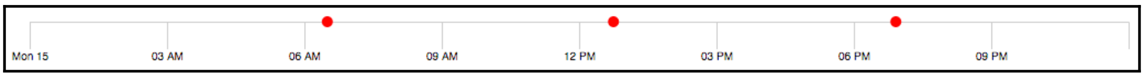

1e+0
2e+0
3e+0
1e+1
2e+1
3e+1
1e+2
2e+2
3e+2
1e+3
2e+3
3e+3
0
500
1,000
1,500
2,000
2,500
3,000
3,500
4,000
4,500
5,000

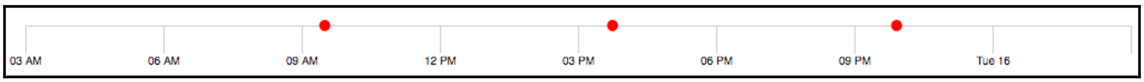

-30
-25
-20
-15
-10
-5
0
5
10
15
20
25
30

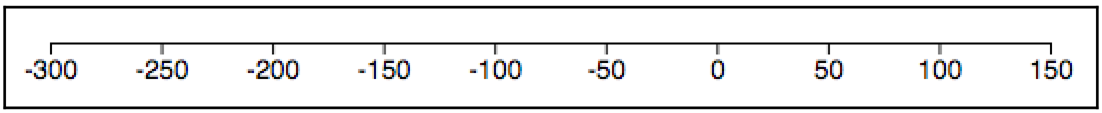

-2,500
-2,000
-1,500
-1,000
-500
0
500
1,000
1,500
2,000
2,500
0
500
1,000
1,500
2,000
2,500
3,000
3,500
4,000
4,500
5,000

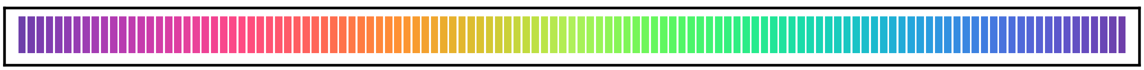

0
100
200
300
400
500
600
700
800
900
1,000

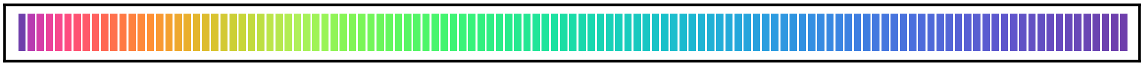

October
November
December
2018
February
March
April
May
June
July
August
September
October

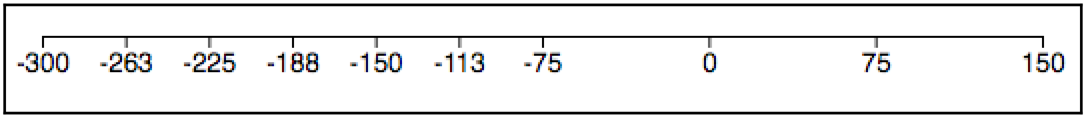

Mon 15
03 AM
06 AM
09 AM
12 PM
03 PM
06 PM
09 PM

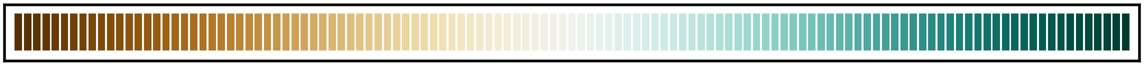

03 AM
06 AM
09 AM
12 PM
03 PM
06 PM
09 PM
Tue 16

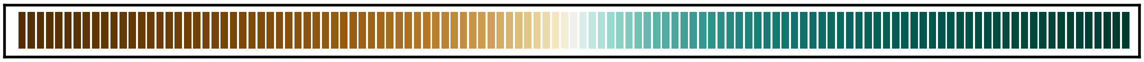

-300
-250
-200
-150
-100
-50
0
50
100
150

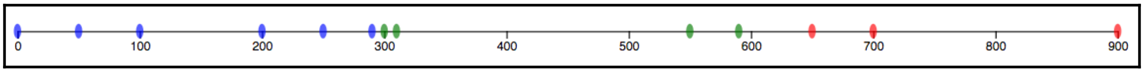


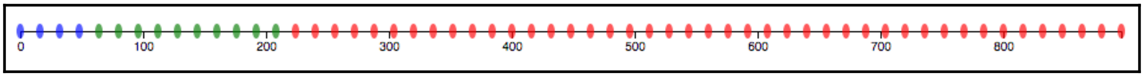


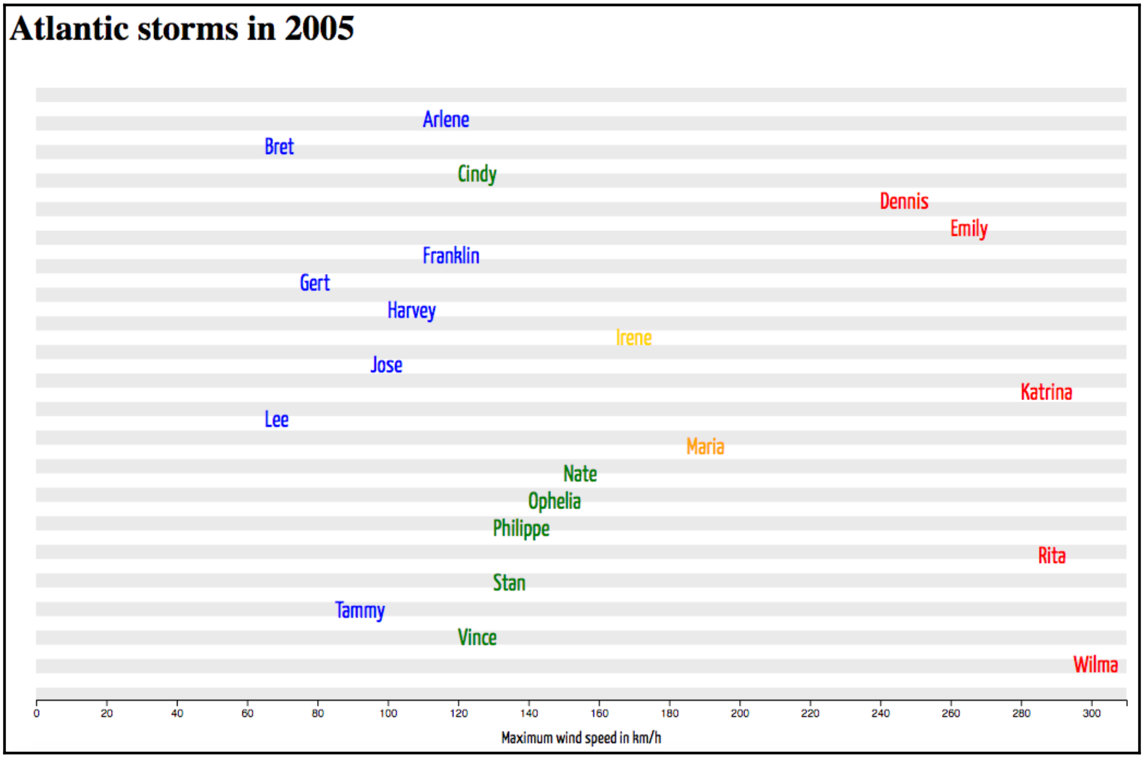

-300
-263
-225
-188
-150
-113
-75
0
75
150



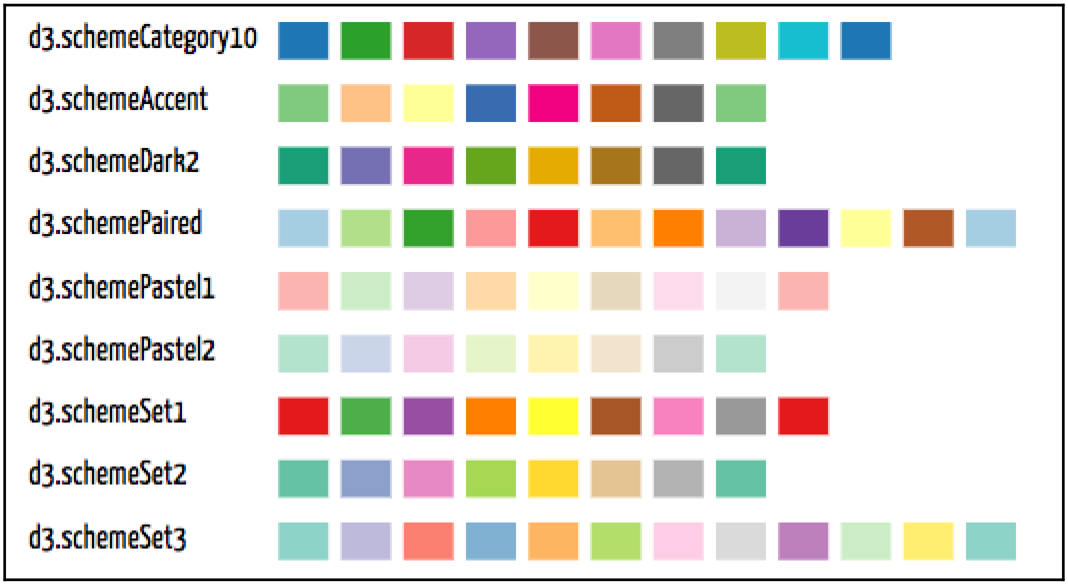


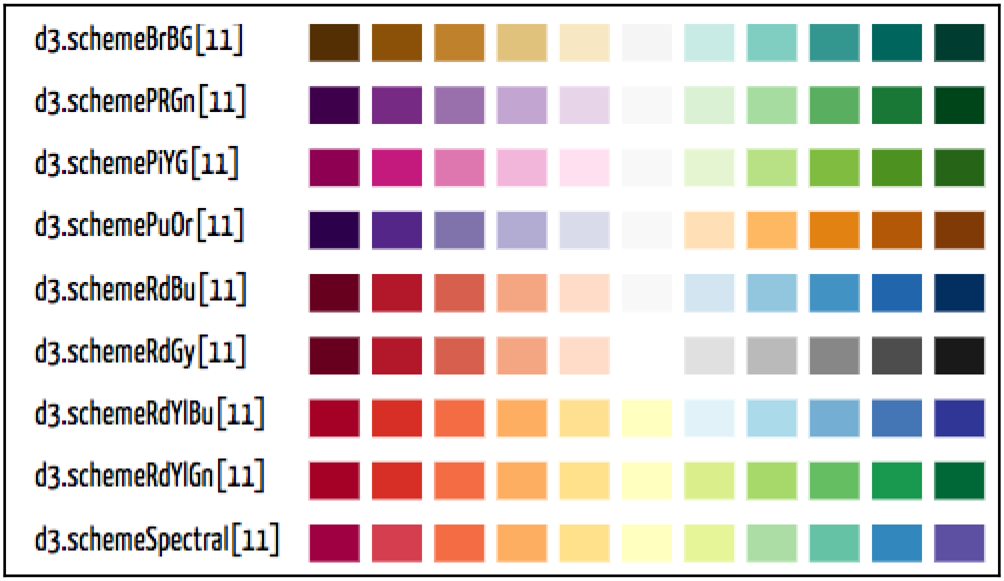

0
100
200
300
400
500
600
700
800
900

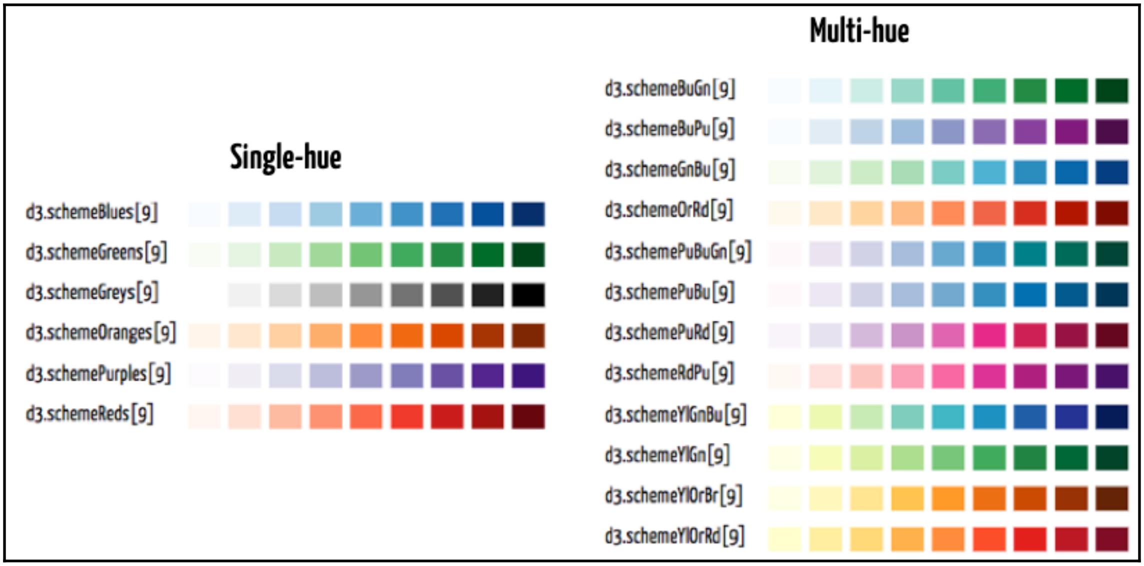

0
100
200
300
400
500
600
700
800

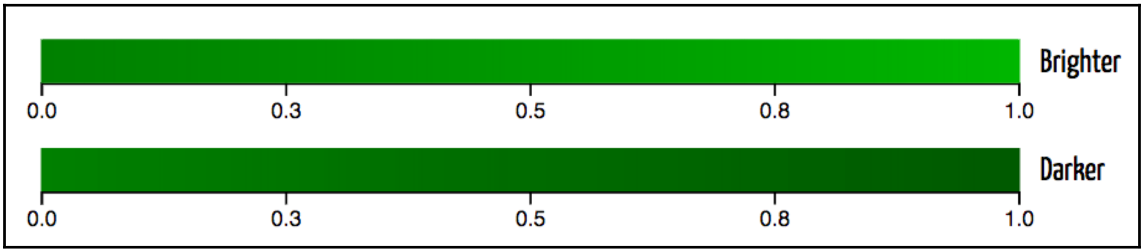

Atlantic storms in 2005
Arlene
Bret
Cindy
Dennis
Emily
Franklin
Gert
Harvey
Irene
Jose
Katrina
Lee
Maria
Nate
Ophelia
Philippe
Rita
Stan
Tammy
Vince
Wilma
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
Maximum wind speed in km/h

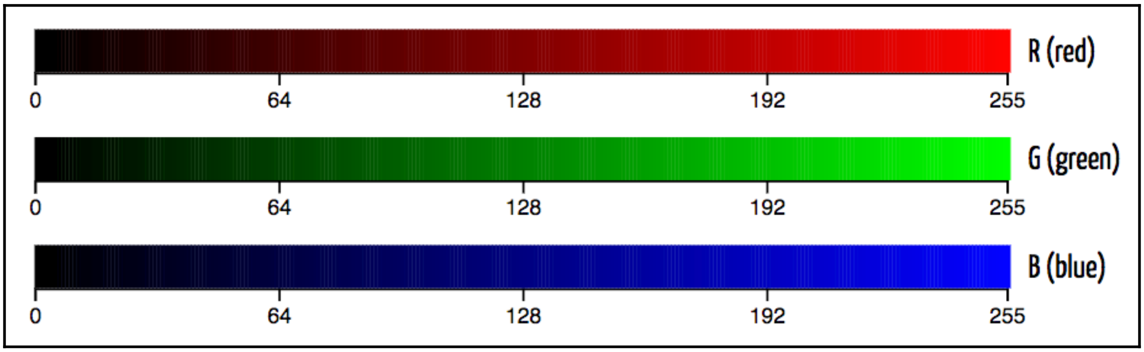

0
0.25
0.5
0.75
1
1.25
1.5
1.75
2
2.25
2.5
2.75
3
3.25
3.5
3.75
4
4.25
4.5
4.75
5

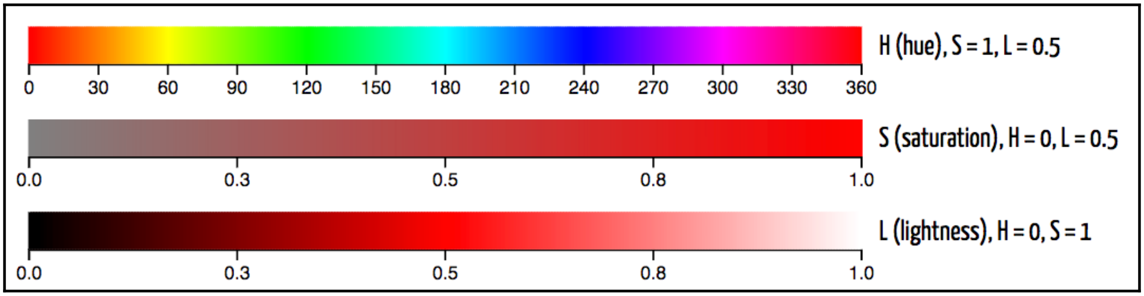

d3.schemeCategory10
d3.schemeAccent
d3.schemeDark2
d3.schemePaired
d3.schemePastel1
d3.schemePastel2
d3.schemeSet1
d3.schemeSet2
d3.schemeSet3

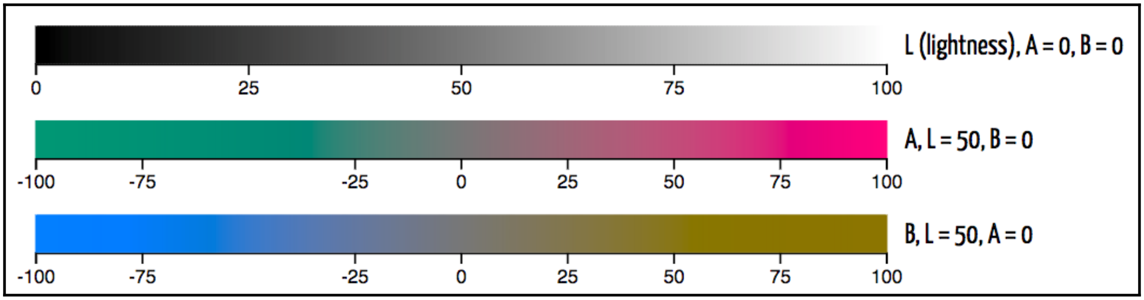

d3.schemeBrBG[11]
d3.schemePRGn[11]
d3.schemePiYG[11]
d3.schemePuOr[11]
d3.schemeRdBu[11]
d3.schemeRdGy[11]
d3.schemeRdYlBu[11]
d3.schemeRdYlGn[11]
d3.schemeSpectral[11]

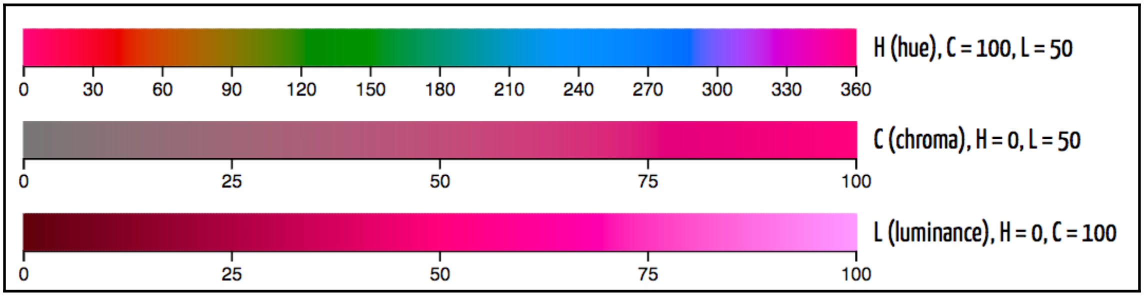

Single-hue
d3.schemeBlues[9]
d3.schemeGreens[9]
d3.schemeGreys[9]
d3.schemeOranges[9]
d3.schemePurples[9]
d3.schemeReds[9]
Multi-hue
d3.schemeBuGn[9]
d3.schemeBuPu[9]
d3.schemeGnBu[9]
d3.schemeOrRd[9]
d3.schemePuBuGn[9]
d3.schemePuBu[9]
d3.schemePuRd[9]
d3.schemeRdPu[9]
d3.schemeYlGnBu[9]
d3.schemeYlGn[9]
d3.schemeYlOrBr[9]
d3.schemeYlOrRd[9]

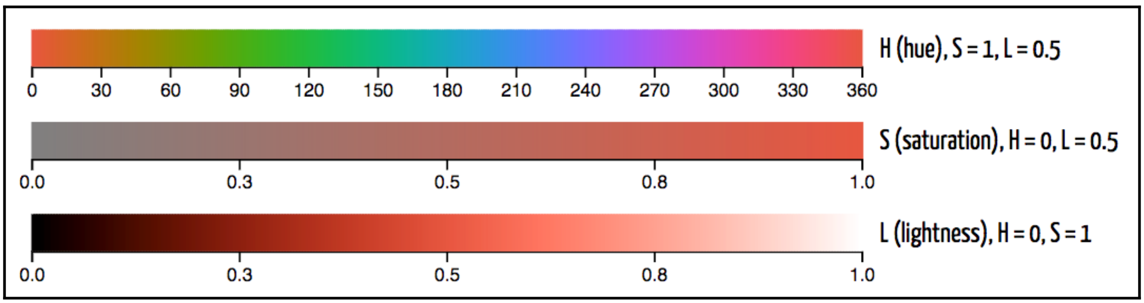

Brighter
0.0
0.3
0.5
0.8
1.0
Darker
0.0
0.3
0.5
0.8
1.0

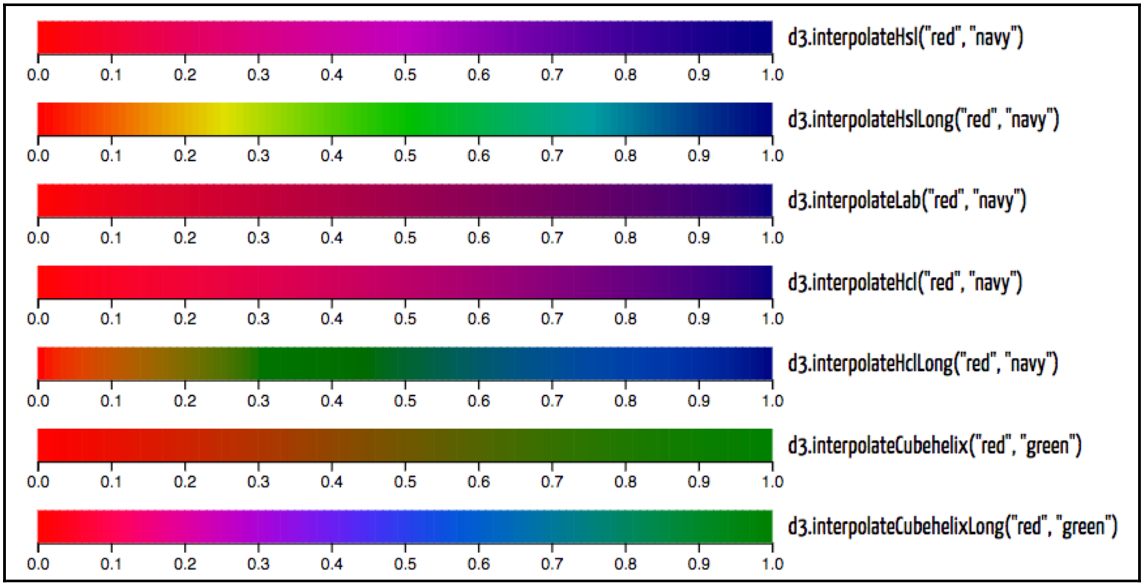

R (red)
0
64
128
192
255
G (green)
0
64
128
192
255
B (blue)
0
64
128
192
255

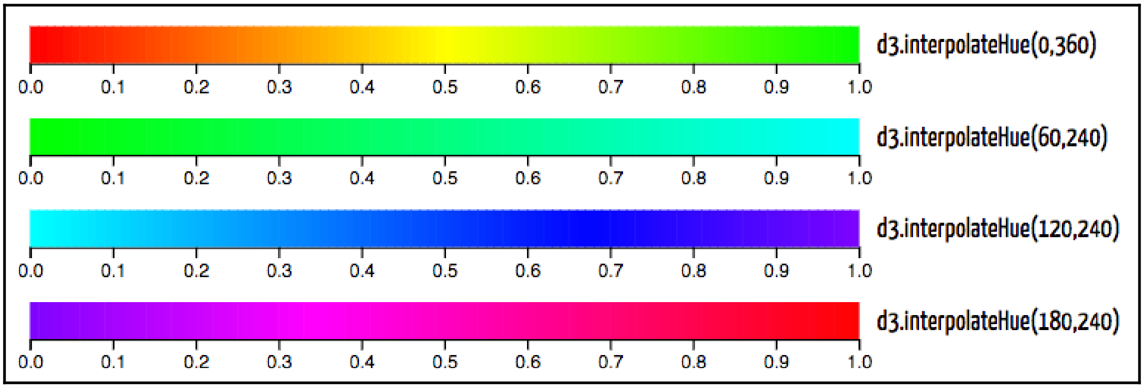

H (hue), S = 1, L = 0.5
0 30 60 90 120 150 180 210 240 270 300 330 360
S (saturation), H = 0, L = 0.5
0.0 0.3 0.5 0.8 1.0
L (lightness), H = 0, S = 1
0.0 0.3 0.5 0.8 1.0

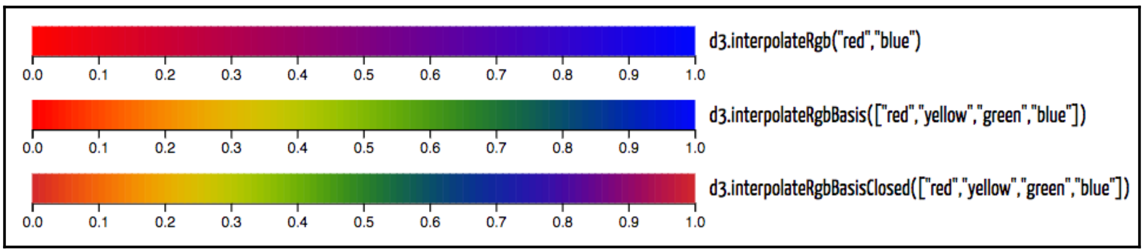

L (lightness), A = 0, B = 0
0 25 50 75 100
A, L = 50, B = 0
-100 -75 -25 0 25 50 75 100
B, L = 50, A = 0
-100 -75 -25 0 25 50 75 100

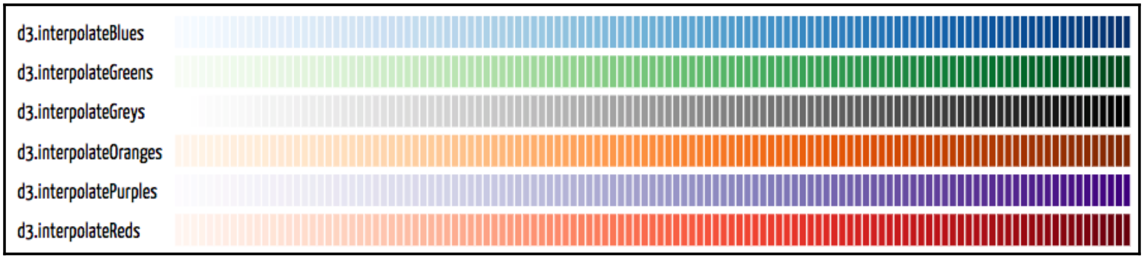

H (hue), C = 100, L = 50
0 30 60 90 120 150 180 210 240 270 300 330 360
C (chroma), H = 0, L = 50
0 25 50 75 100
L (luminance), H = 0, C = 100
0 25 50 75 100

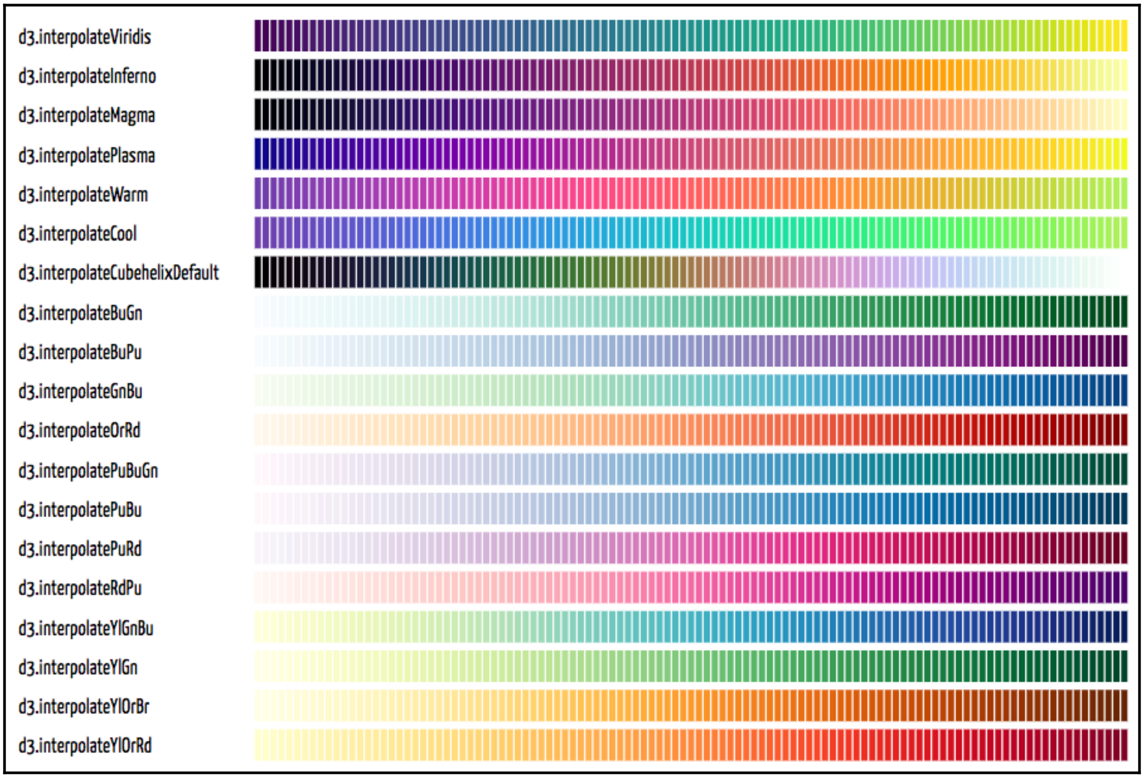

H (hue), S = 1, L = 0.5
0 30 60 90 120 150 180 210 240 270 300 330 360
S (saturation), H = 0, L = 0.5
0.0 0.3 0.5 0.8 1.0
L (lightness), H = 0, S = 1
0.0 0.3 0.5 0.8 1.0

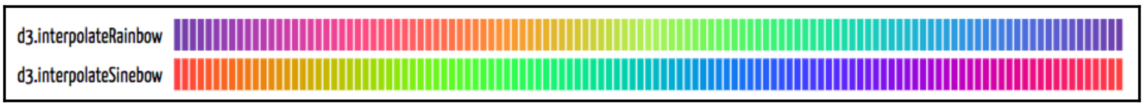

d3.interpolateHsl("red", "navy")
d3.interpolateHslLong("red", "navy")
d3.interpolateLab("red", "navy")
d3.interpolateHcl("red", "navy")
d3.interpolateHclLong("red", "navy")
d3.interpolateCubehelix("red", "green")
d3.interpolateCubehelixLong("red", "green")
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

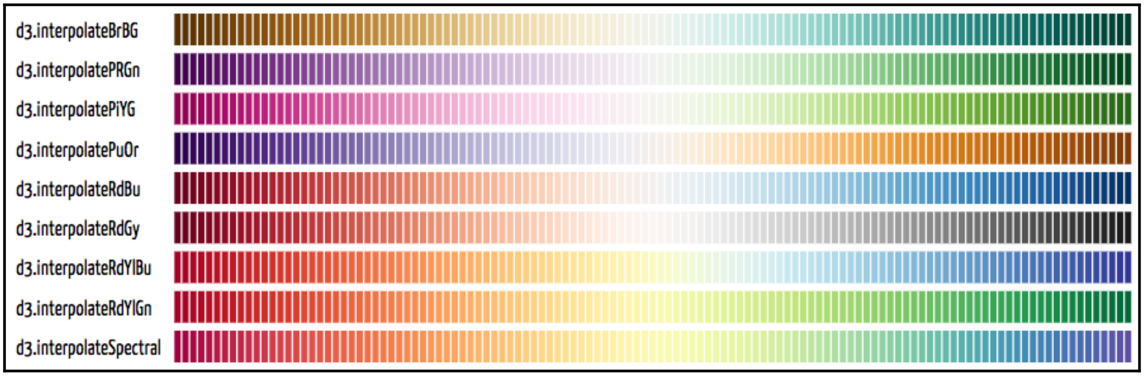

d3.interpolateHue(0,360)
d3.interpolateHue(60,240)
d3.interpolateHue(120,240)
d3.interpolateHue(180,240)
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

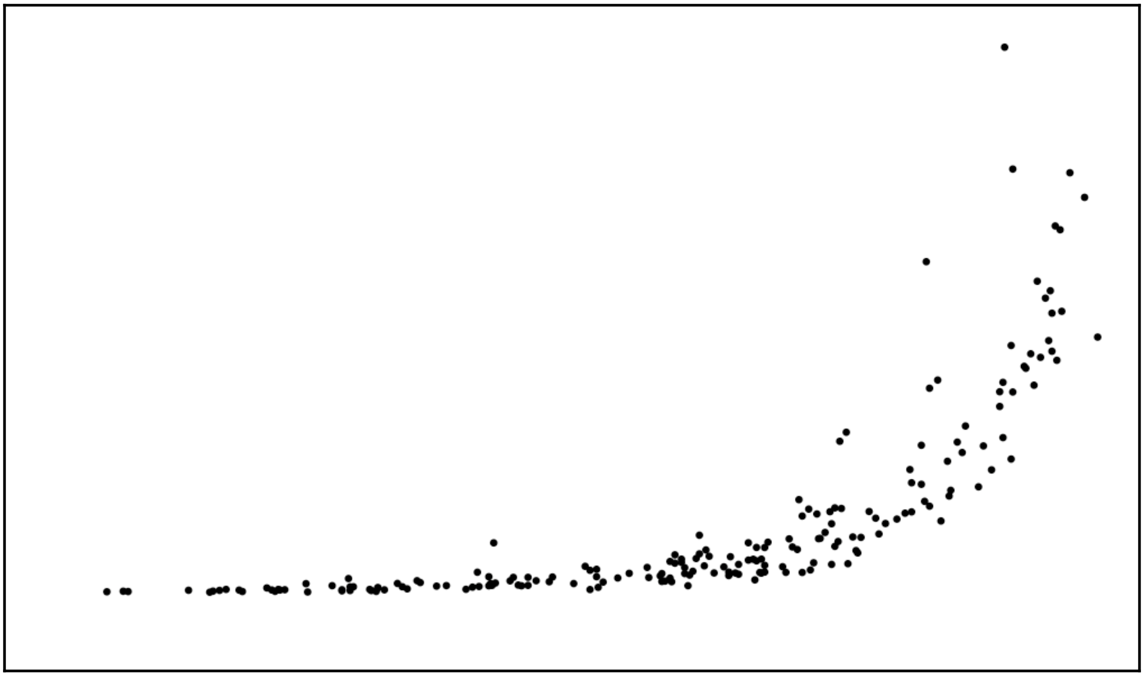

d3.interpolateRgb("red","blue")
d3.interpolateRgbBasis(["red","yellow","green","blue"])
d3.interpolateRgbBasisClosed(["red","yellow","green","blue"])
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

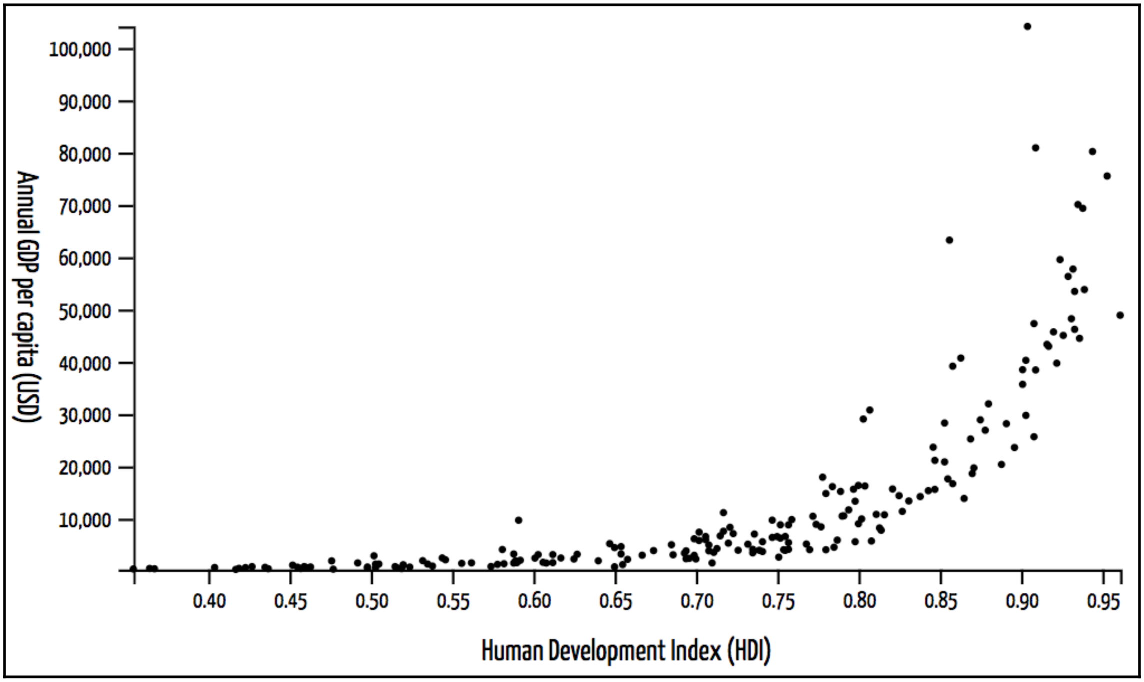

d3.interpolateBlues
d3.interpolateGreens
d3.interpolateGreys
d3.interpolateOranges
d3.interpolatePurples
d3.interpolateReds


d3.interpolateViridis
d3.interpolateInferno
d3.interpolateMagma
d3.interpolatePlasma
d3.interpolateWarm
d3.interpolateCool
d3.interpolateCubehelixDefault
d3.interpolateBuGn
d3.interpolateBuPu
d3.interpolateGnBu
d3.interpolateOrRd
d3.interpolatePuBuGn
d3.interpolatePuBu
d3.interpolatePuRd
d3.interpolateRdPu
d3.interpolateYlGnBu
d3.interpolateYlGn
d3.interpolateYlOrBr
d3.interpolateYlOrRd

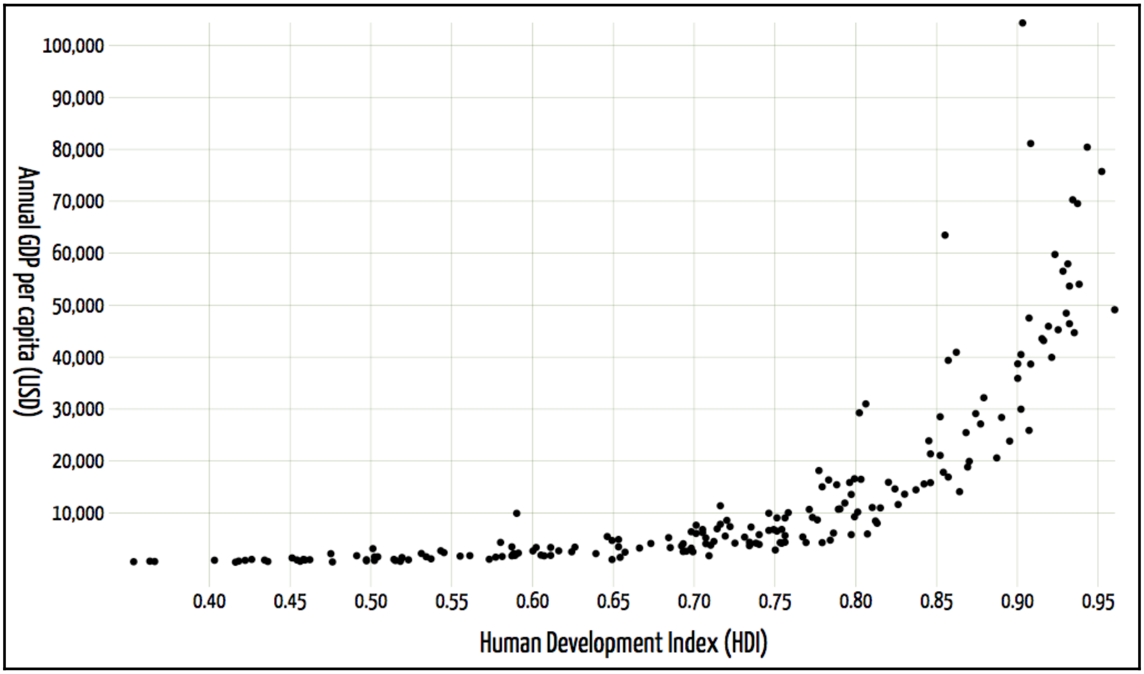

d3.interpolateRainbow
d3.interpolateSinebow


d3.interpolateBrBG
d3.interpolatePRGn
d3.interpolatePiYG
d3.interpolatePuOr
d3.interpolateRdBu
d3.interpolateRdGy
d3.interpolateRdYlBu
d3.interpolateRdYlGn
d3.interpolateSpectral

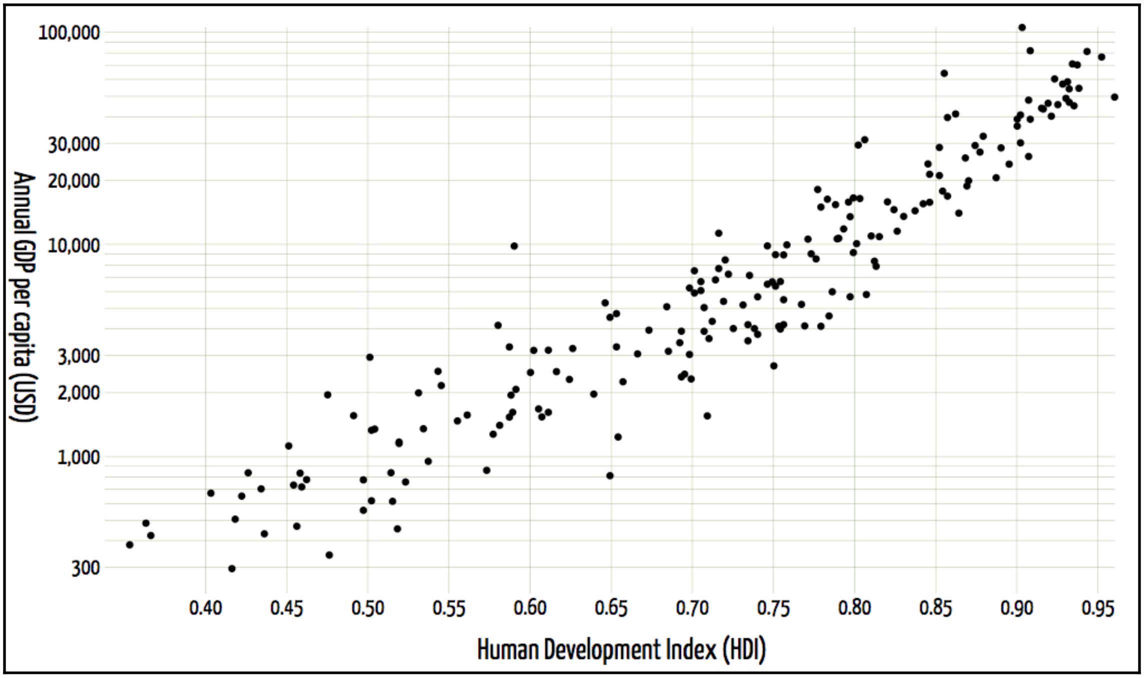


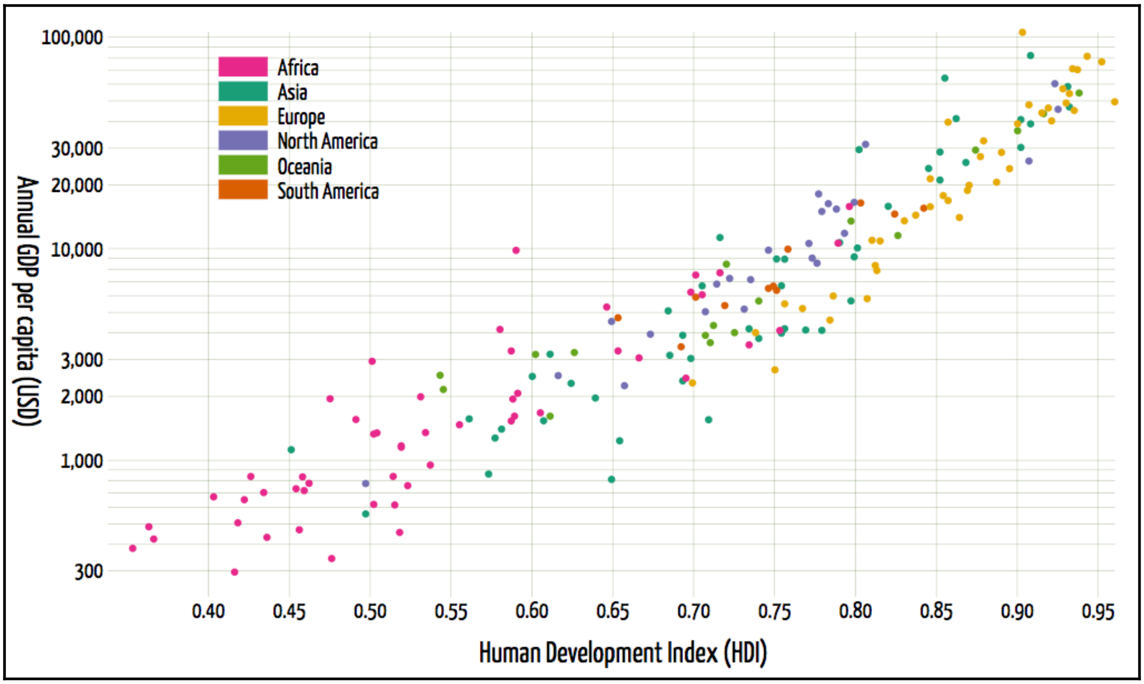


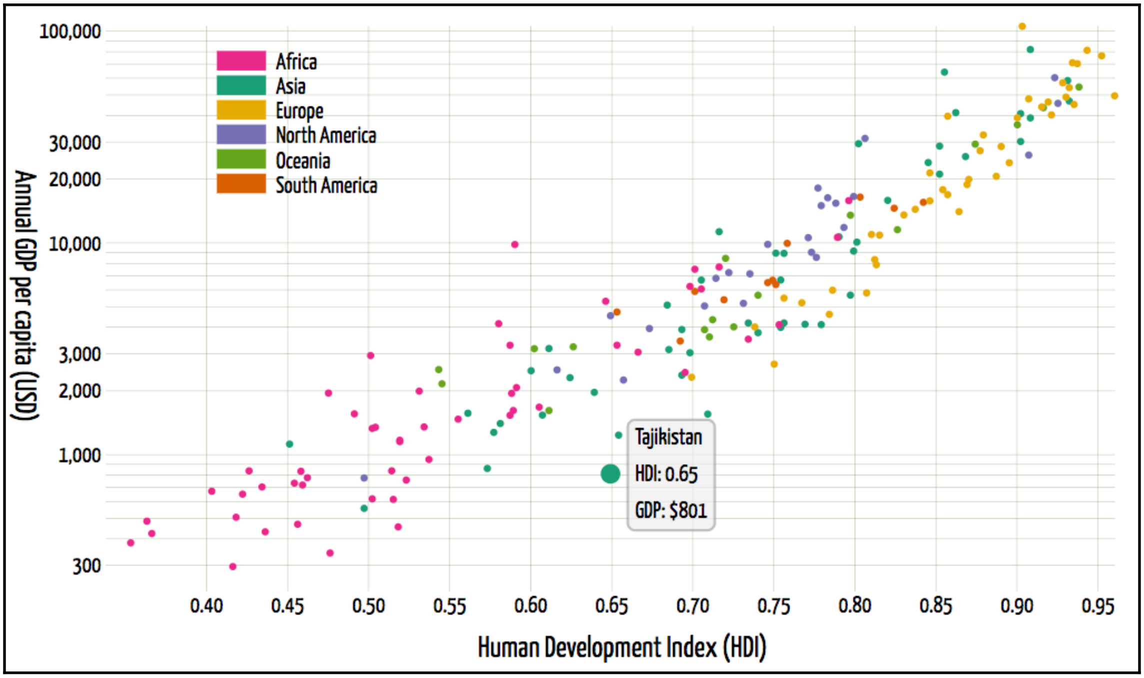
100,000
90,000
80,000
70,000
60,000
50,000
40,000
30,000
20,000
10,000
Annual GDP per capita (USD)
0.40
0.45
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
Human Development Index (HDI)

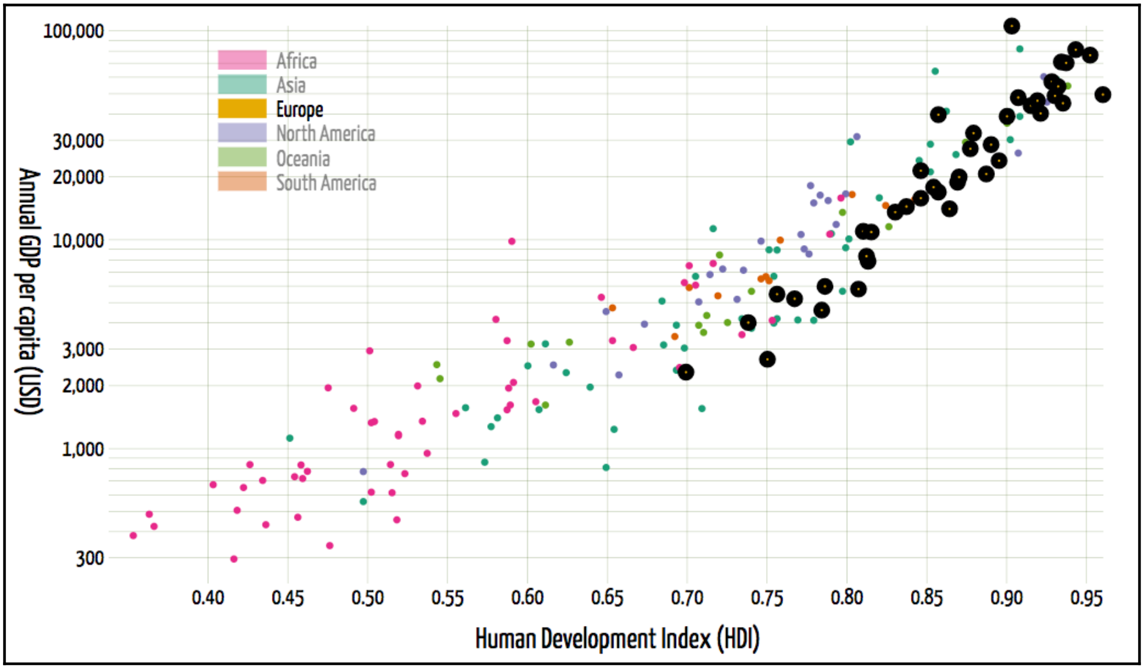


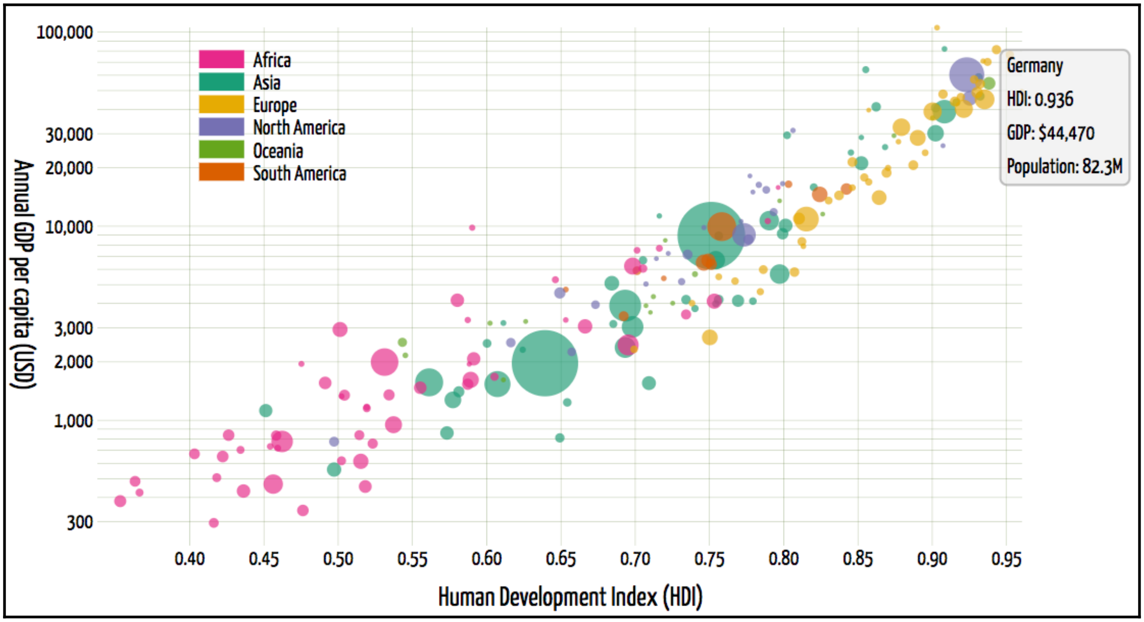
100,000
90,000
80,000
70,000
60,000
50,000
40,000
30,000
20,000
10,000
Annual GDP per capita (USD)
0.40
0.45
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
Human Development Index (HDI)


100,000
30,000
20,000
10,000
3,000
2,000
1,000
300
Annual GDP per capita (USD)
0.40
0.45
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
Human Development Index (HDI)

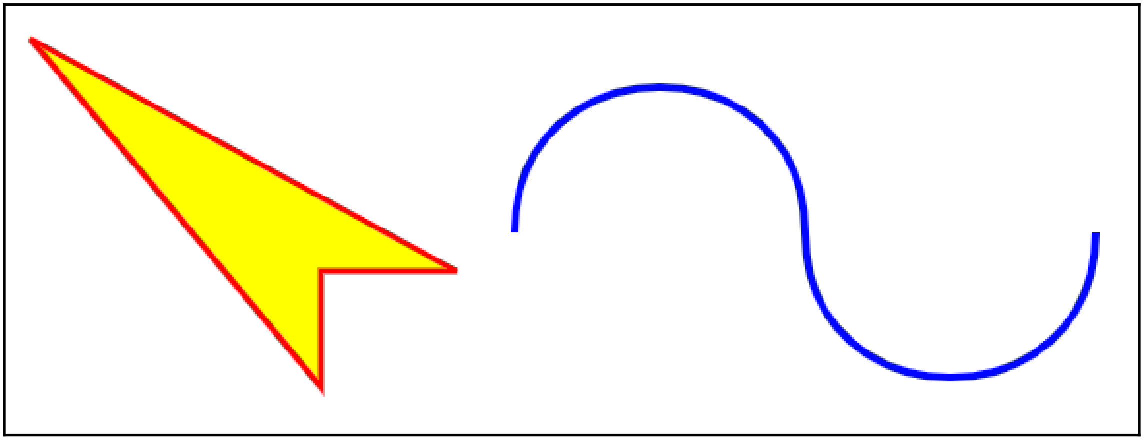

Africa
Asia
Europe
North America
Oceania
South America
100,000
30,000
20,000
10,000
3,000
2,000
1,000
300
Annual GDP per capita (USD)
0.40
0.45
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
Human Development Index (HDI)

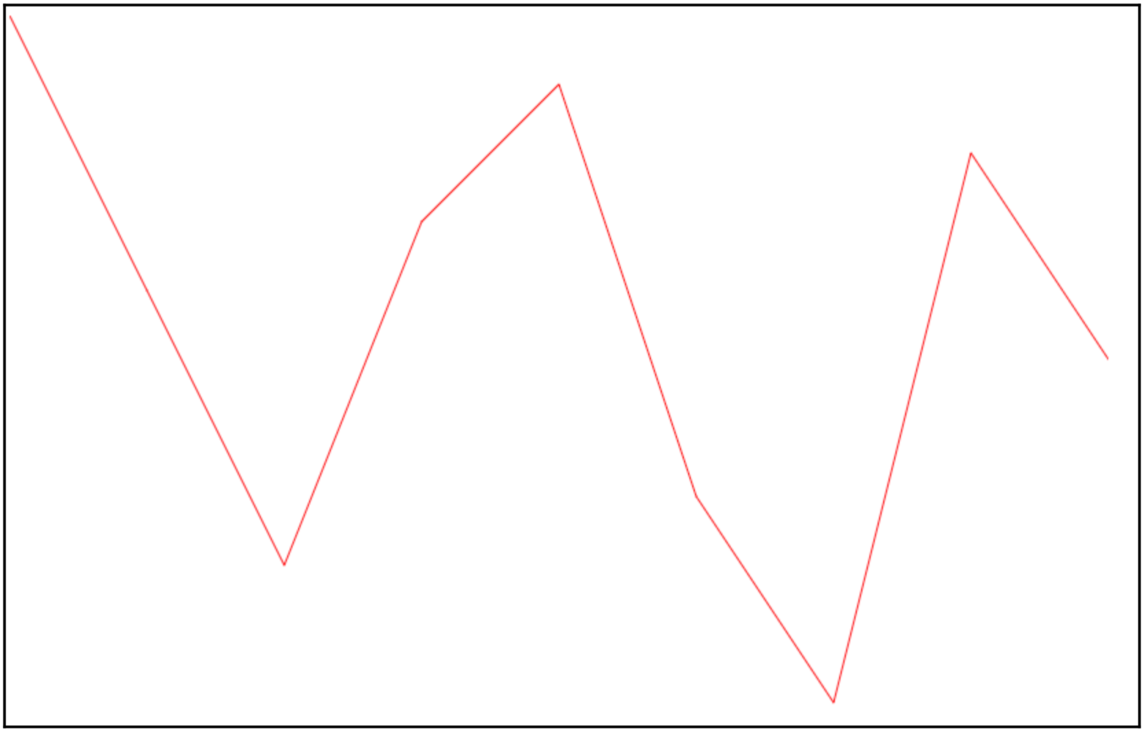

100,000
30,000
20,000
10,000
3,000
2,000
1,000
300
Annual GDP per capita (USD)
Africa
Asia
Europe
North America
Oceania
South America
Tajikistan
HDI: 0.65
GDP: $801
0.40
0.45
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
Human Development Index (HDI)

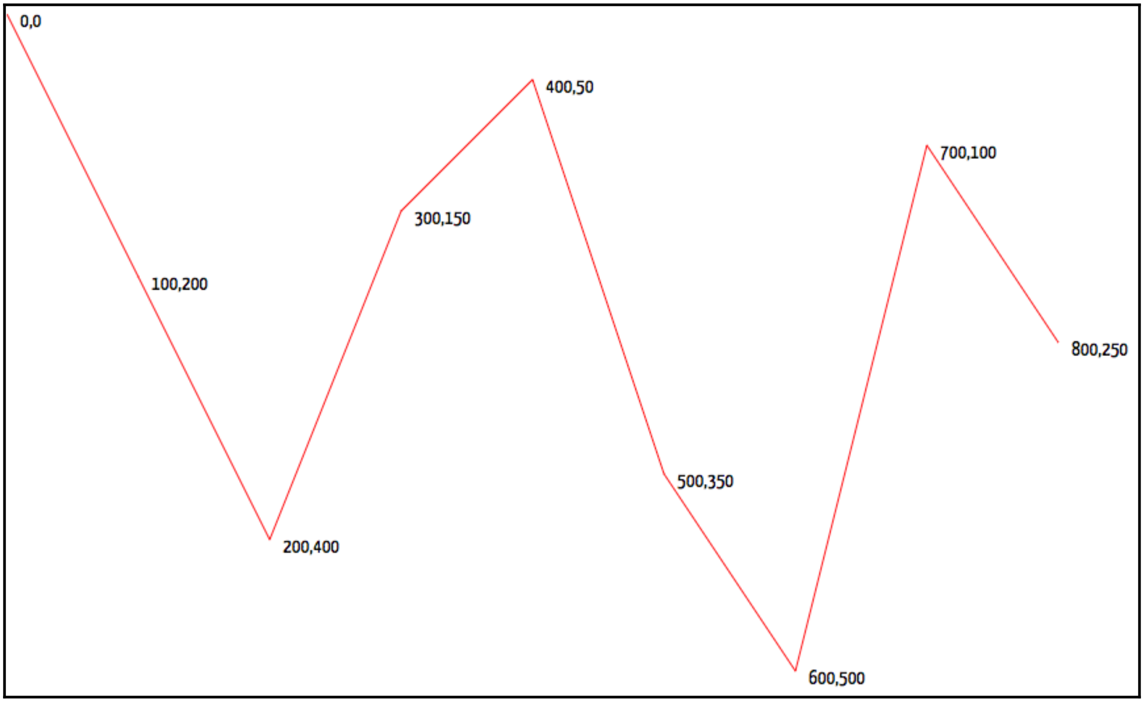

100,000
30,000
20,000
10,000
3,000
2,000
1,000
300
Annual GDP per capita (USD)
Africa
Asia
Europe
North America
Oceania
South America
0.40
0.45
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
Human Development Index (HDI)

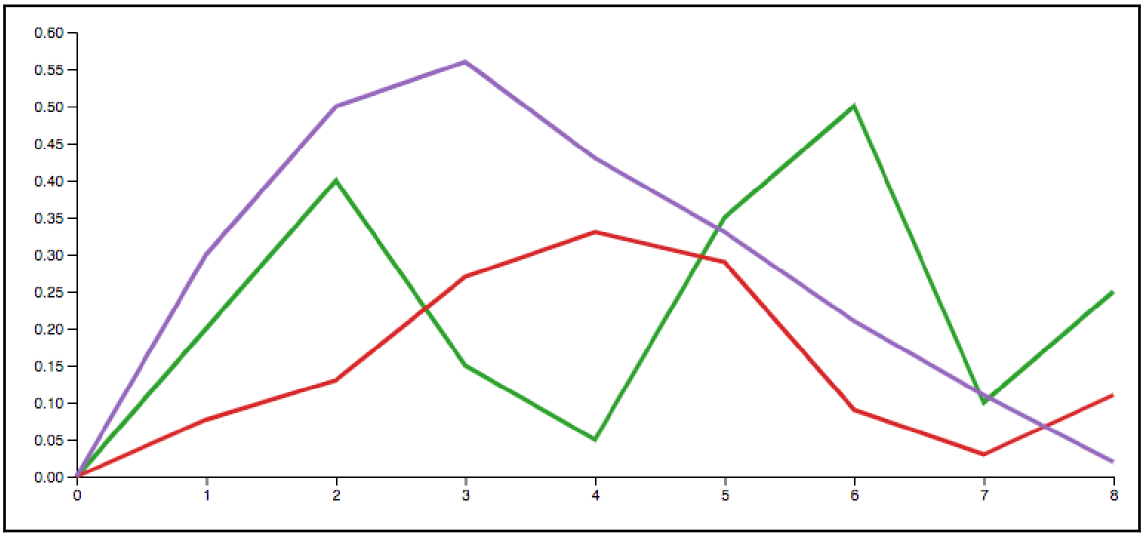

Africa
Asia
Europe
North America
Oceania
South America
Germany
HDI: 0.936
GDP: $44,470
Population: 82.3M
100,000
30,000
20,000
10,000
3,000
2,000
1,000
300
Annual GDP per capita (USD)
0.40
0.45
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
Human Development Index (HDI)

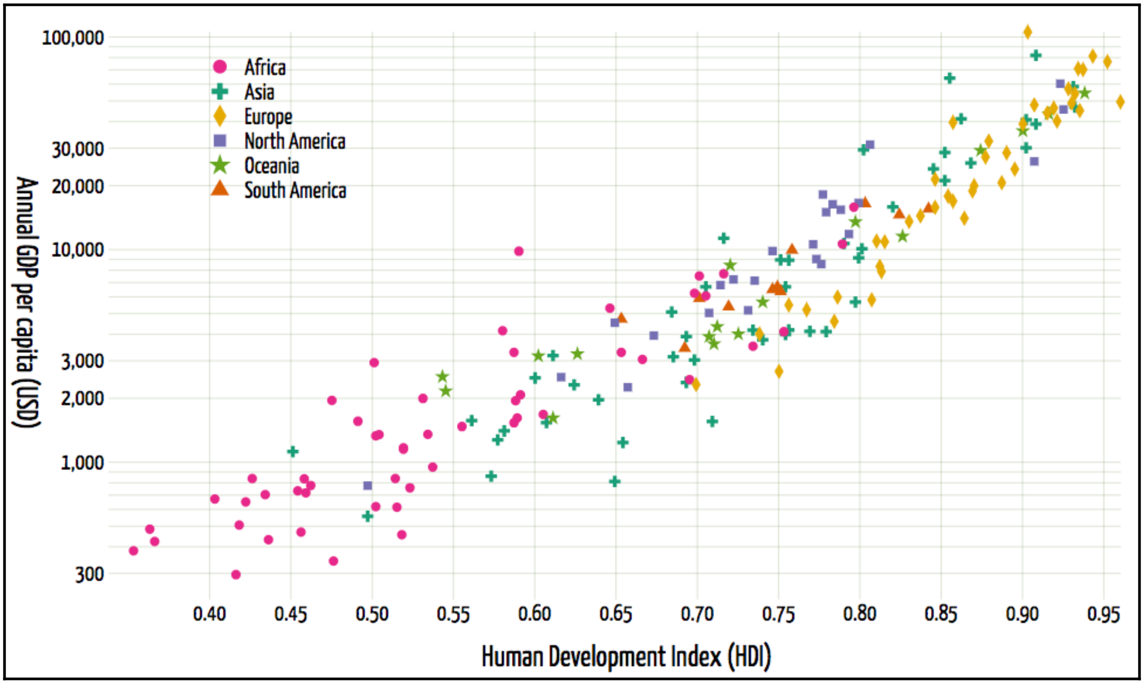
Chapter 7: Shape and Layout Generators

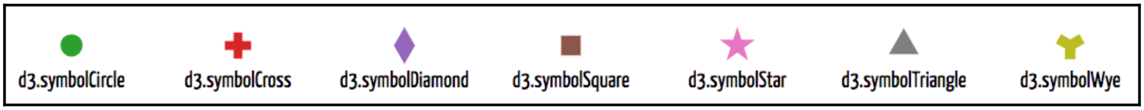


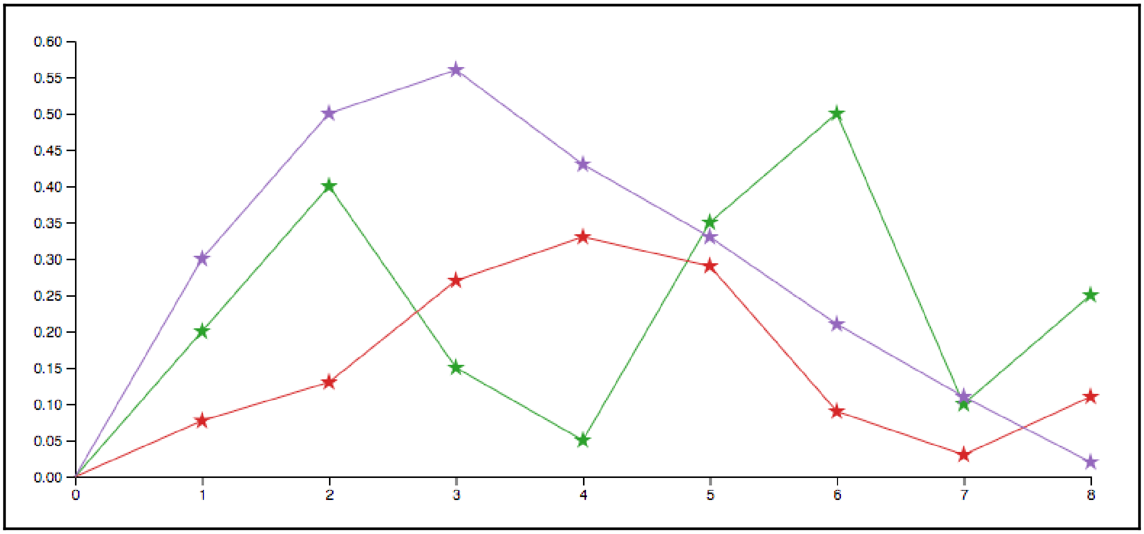


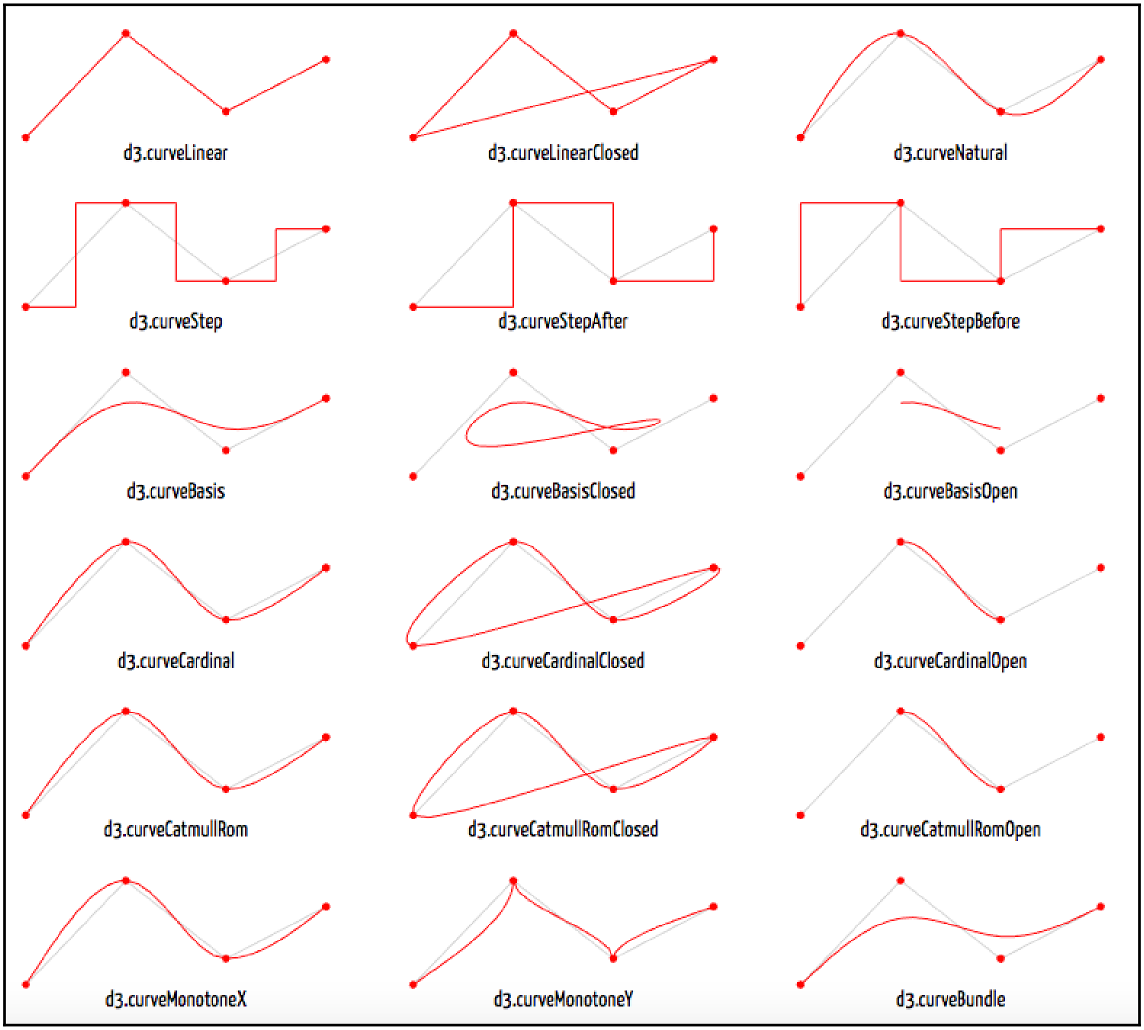

0,0
100,200
200,400
300,150
400,50
500,350
600,500
700,100
800,250

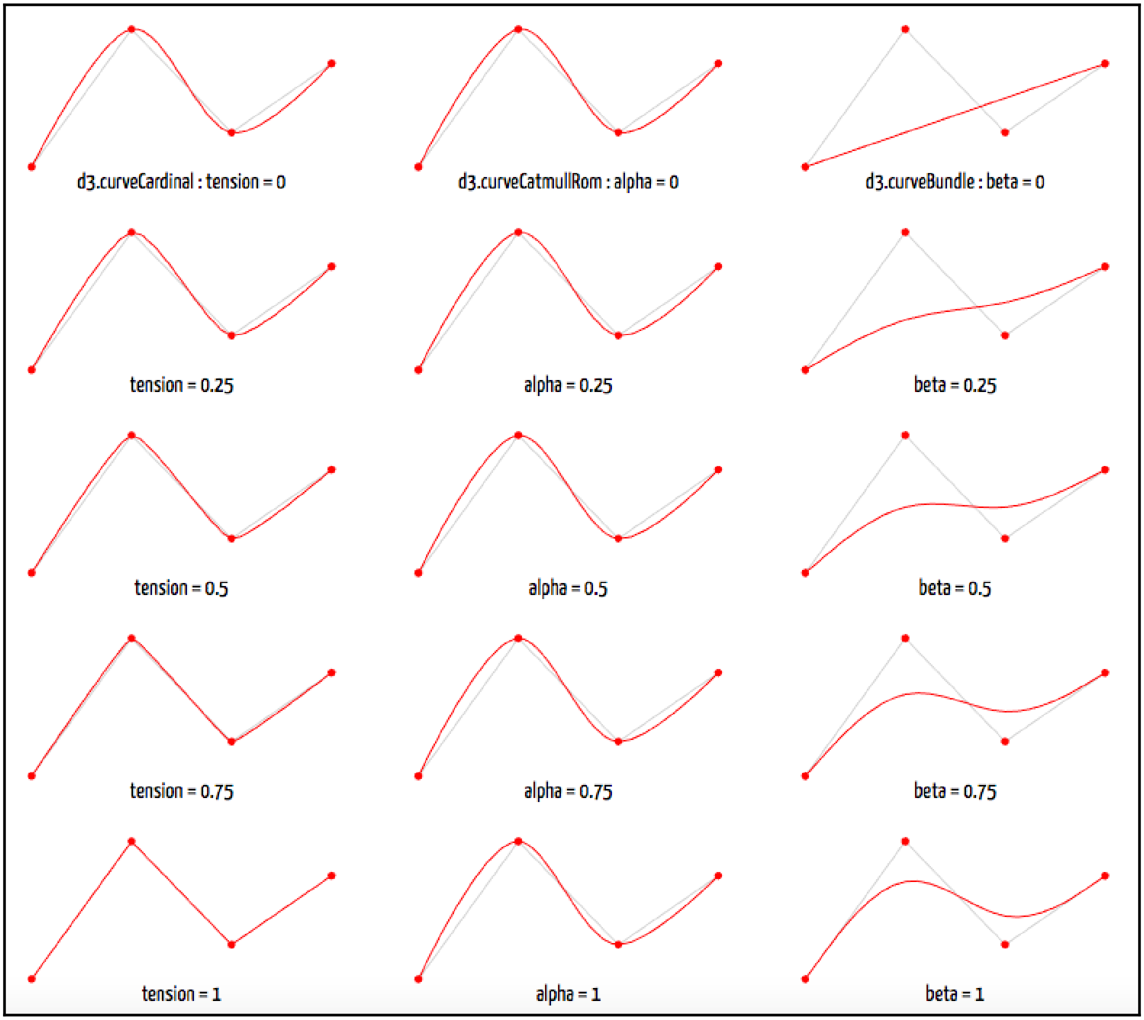

0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
1
2
3
4
5
6
7
8

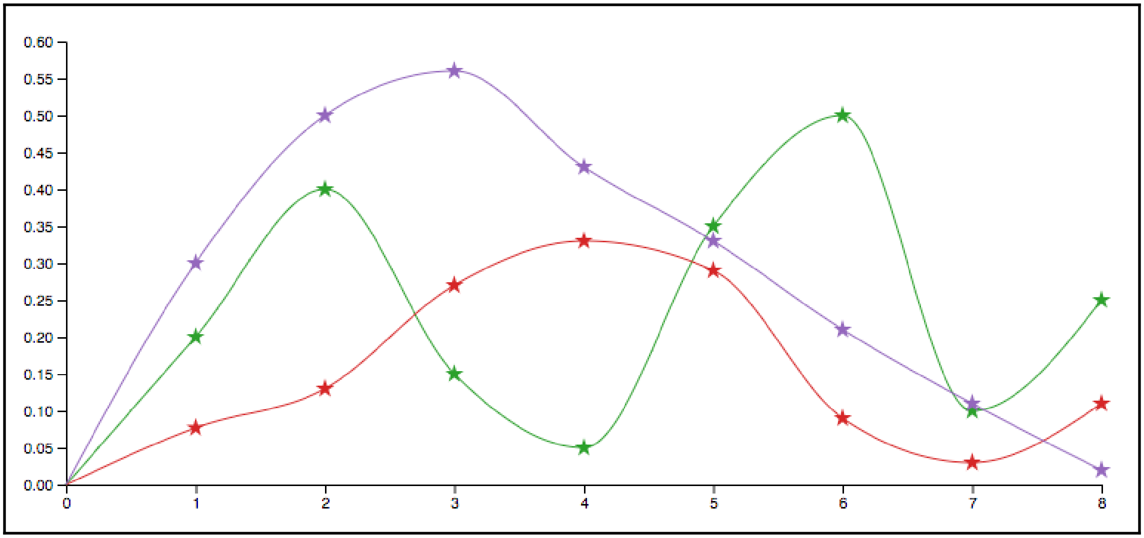

Africa
Asia
Europe
North America
Oceania
South America
Annual GDP per capita (USD)
100,000
30,000
20,000
10,000
3,000
2,000
1,000
300
0.40
0.45
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
Human Development Index (HDI)

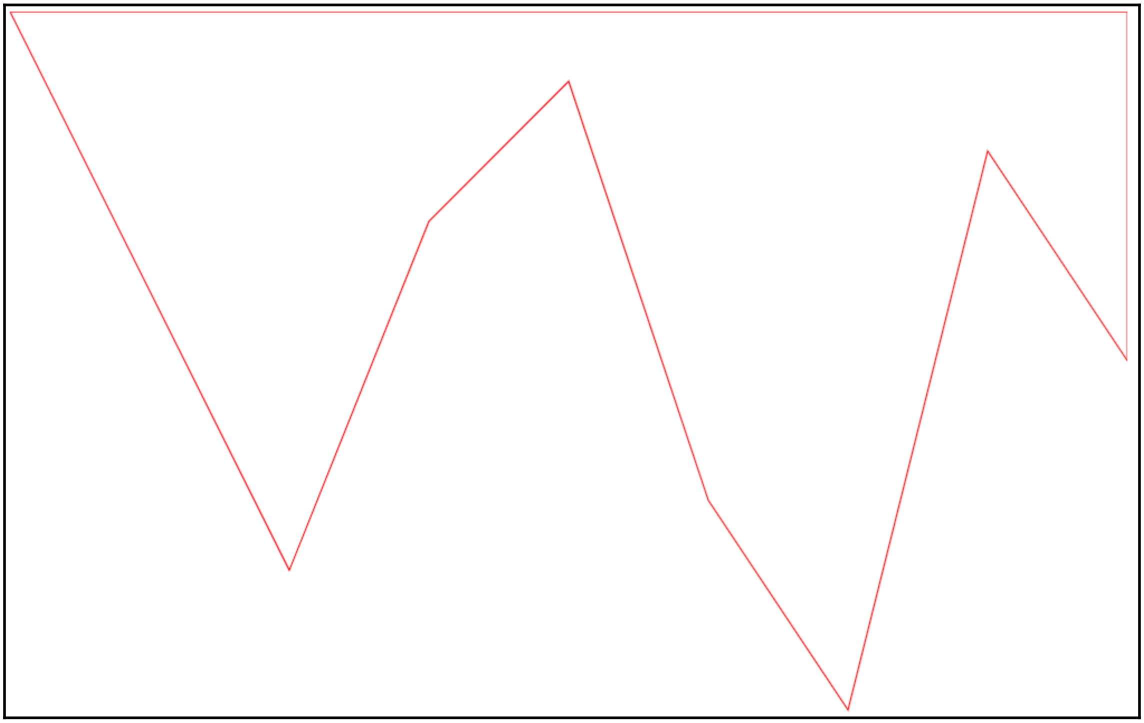

d3.symbolCircle
d3.symbolCross
d3.symbolDiamond
d3.symbolSquare
d3.symbolStar
d3.symbolTriangle
d3.symbolWye

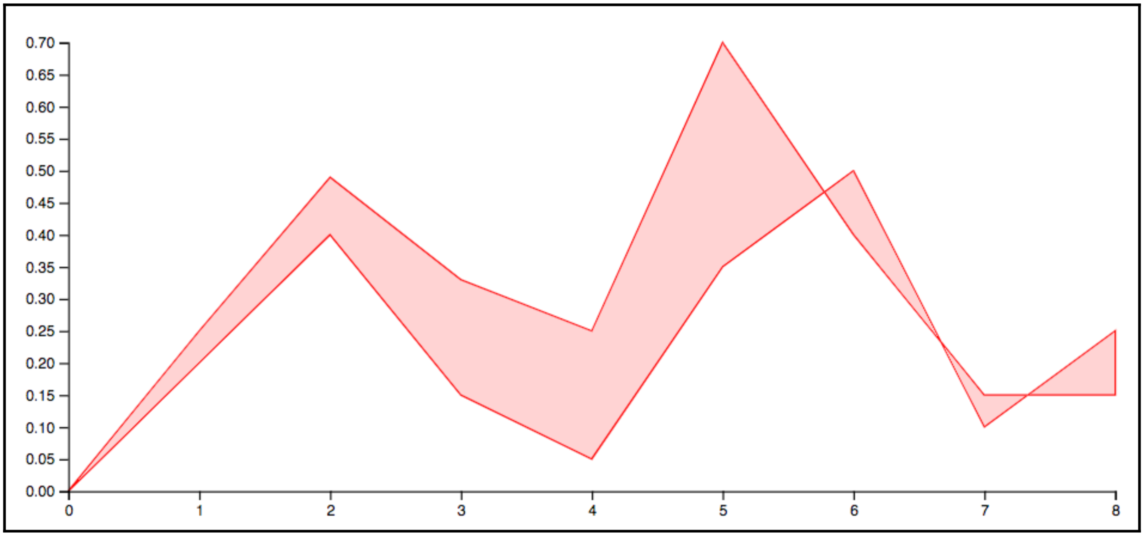

0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
1
2
3
4
5
6
7
8

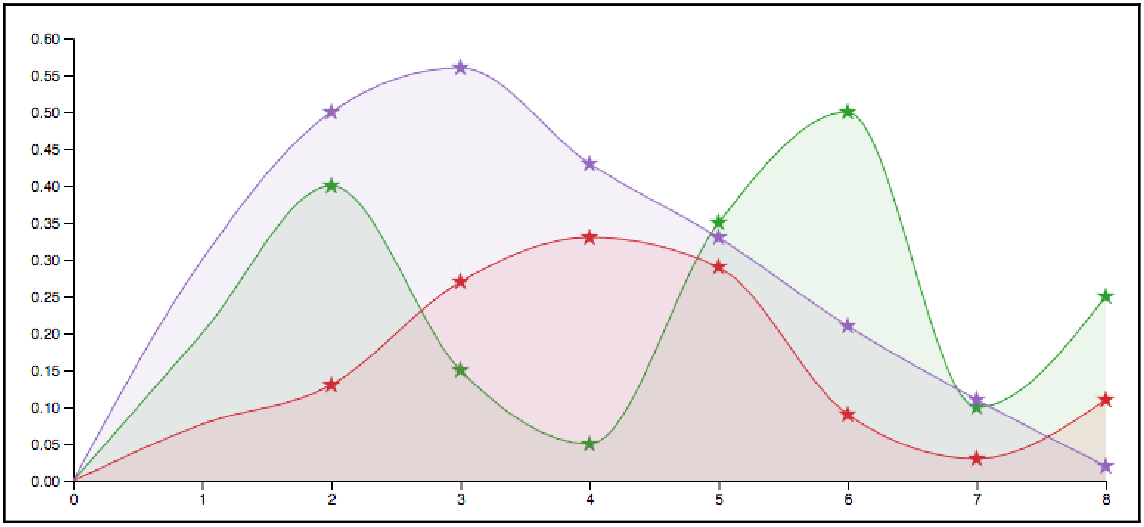

d3.curveLinear
d3.curveLinearClosed
d3.curveNatural
d3.curveStep
d3.curveStepAfter
d3.curveStepBefore
d3.curveBasis
d3.curveBasisClosed
d3.curveBasisOpen
d3.curveCardinal
d3.curveCardinalClosed
d3.curveCardinalOpen
d3.curveCatmullRom
d3.curveCatmullRomClosed
d3.curveCatmullRomOpen
d3.curveMonotoneX
d3.curveMonotoneY
d3.curveBundle

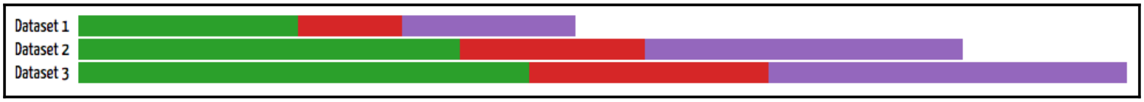

d3.curveCardinal : tension = 0
d3.curveCatmullRom : alpha = 0
d3.curveBundle : beta = 0
tension = 0.25
alpha = 0.25
beta = 0.25
tension = 0.5
alpha = 0.5
beta = 0.5
tension = 0.75
alpha = 0.75
beta = 0.75
tension = 1
alpha = 1
beta = 1

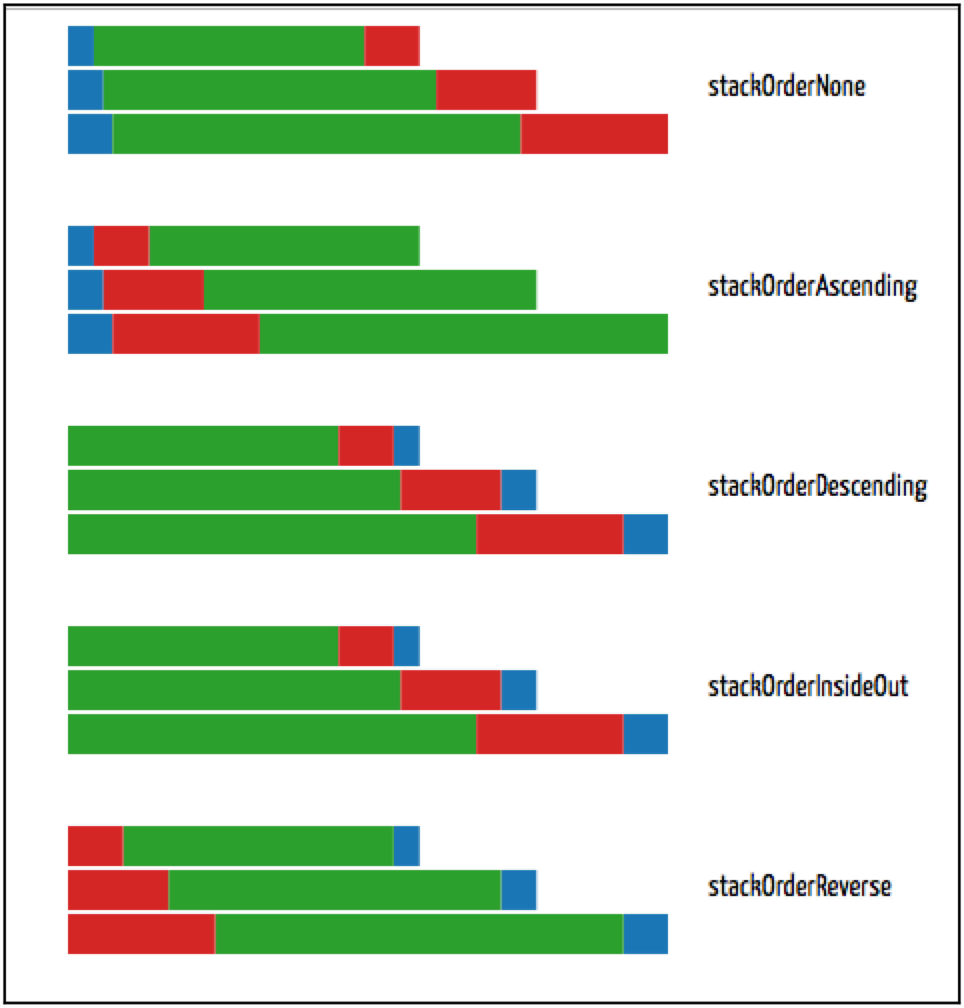

0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
1
2
3
4
5
6
7
8



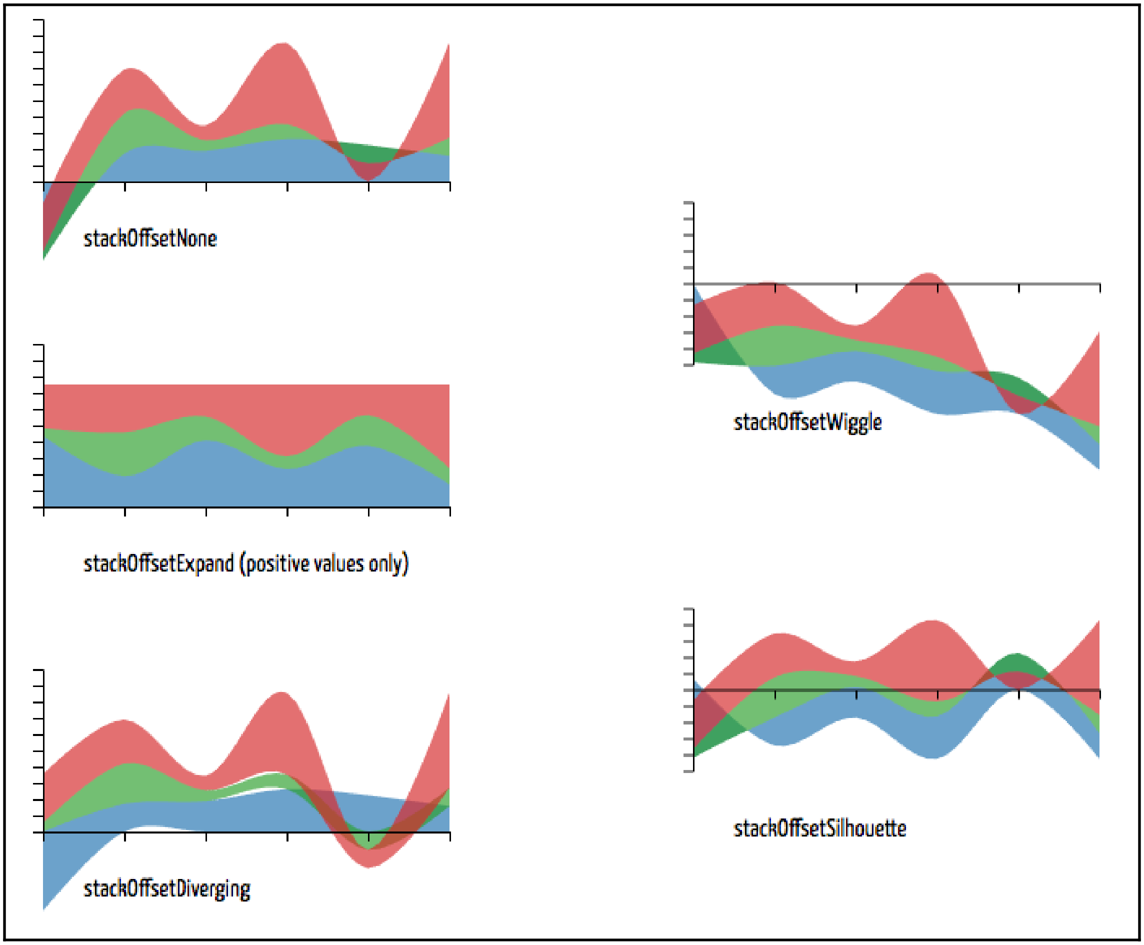

0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0 1 2 3 4 5 6 7 8

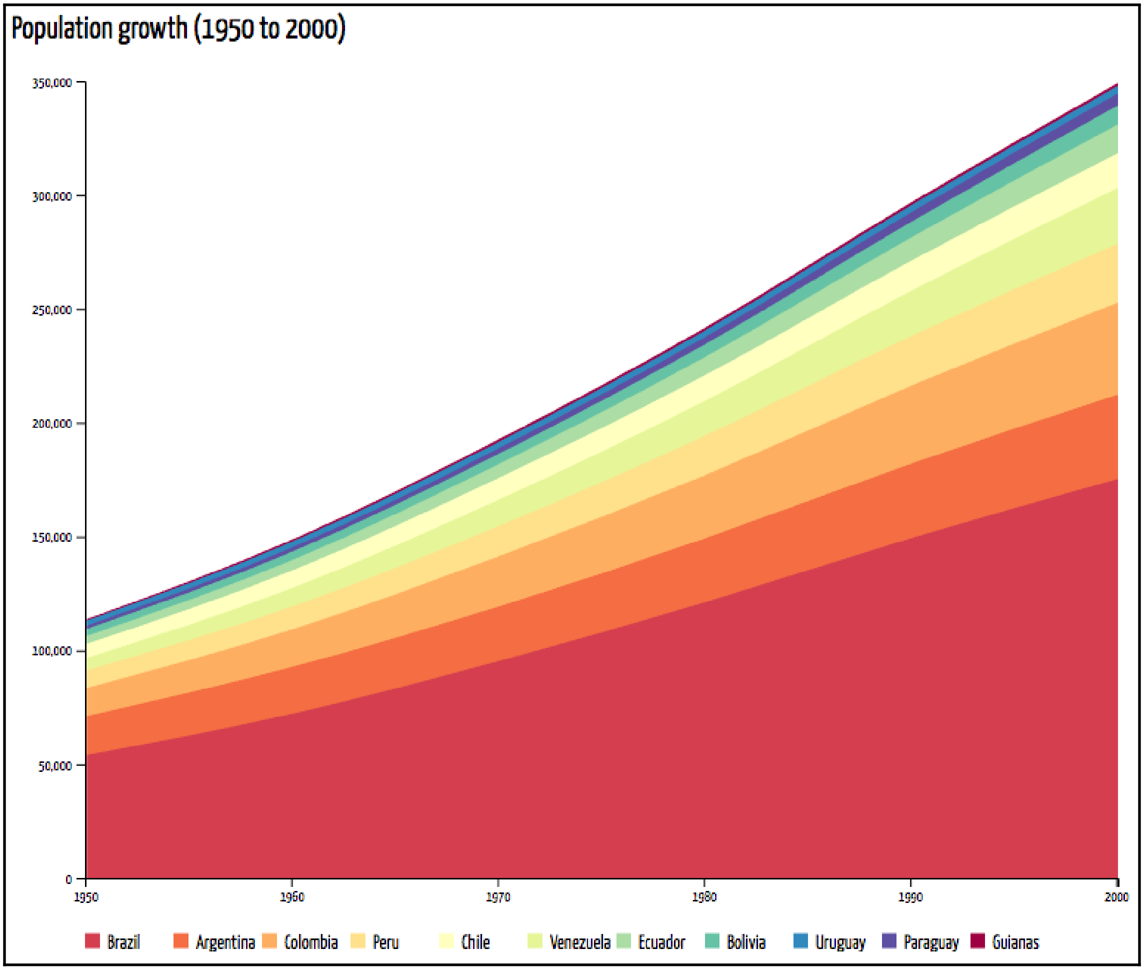

0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0 1 2 3 4 5 6 7 8


Dataset 1
Dataset 2
Dataset 3

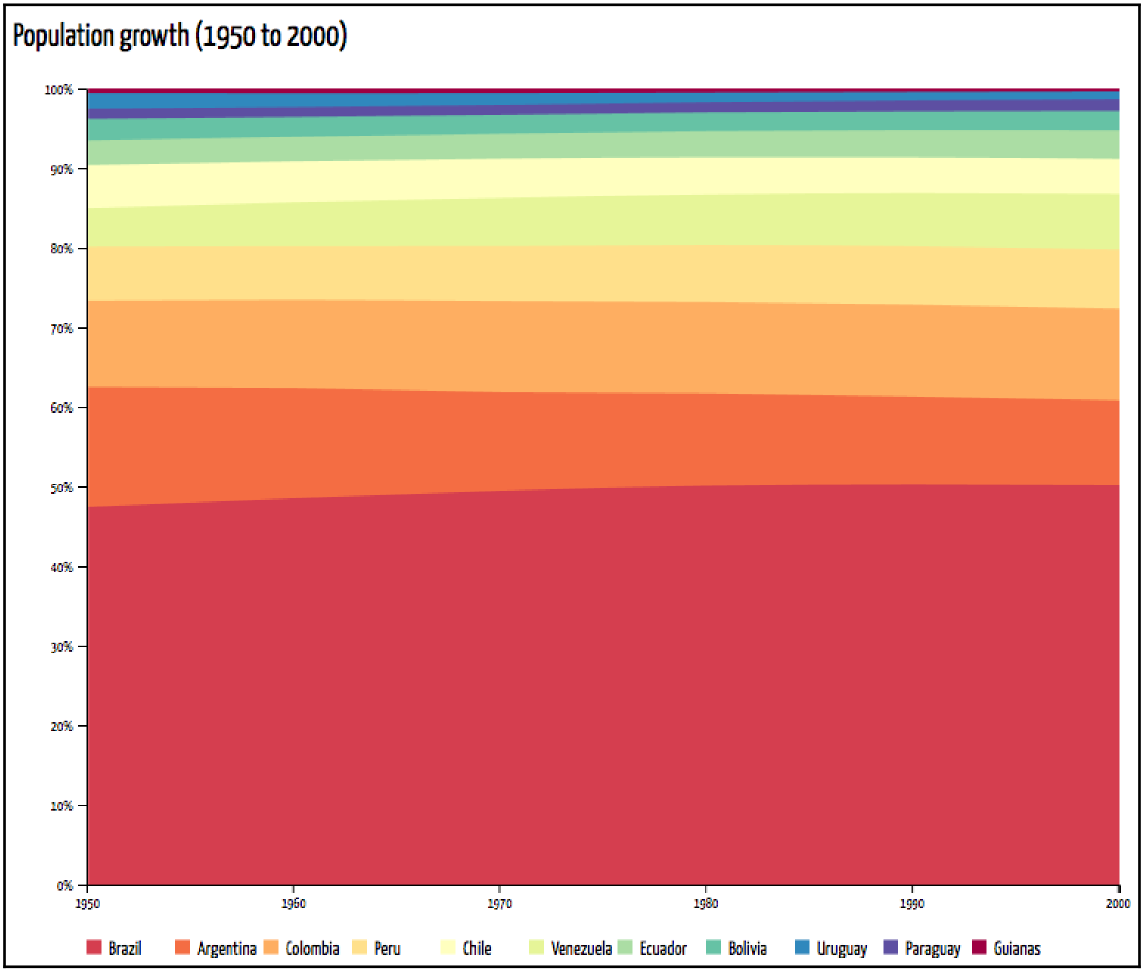


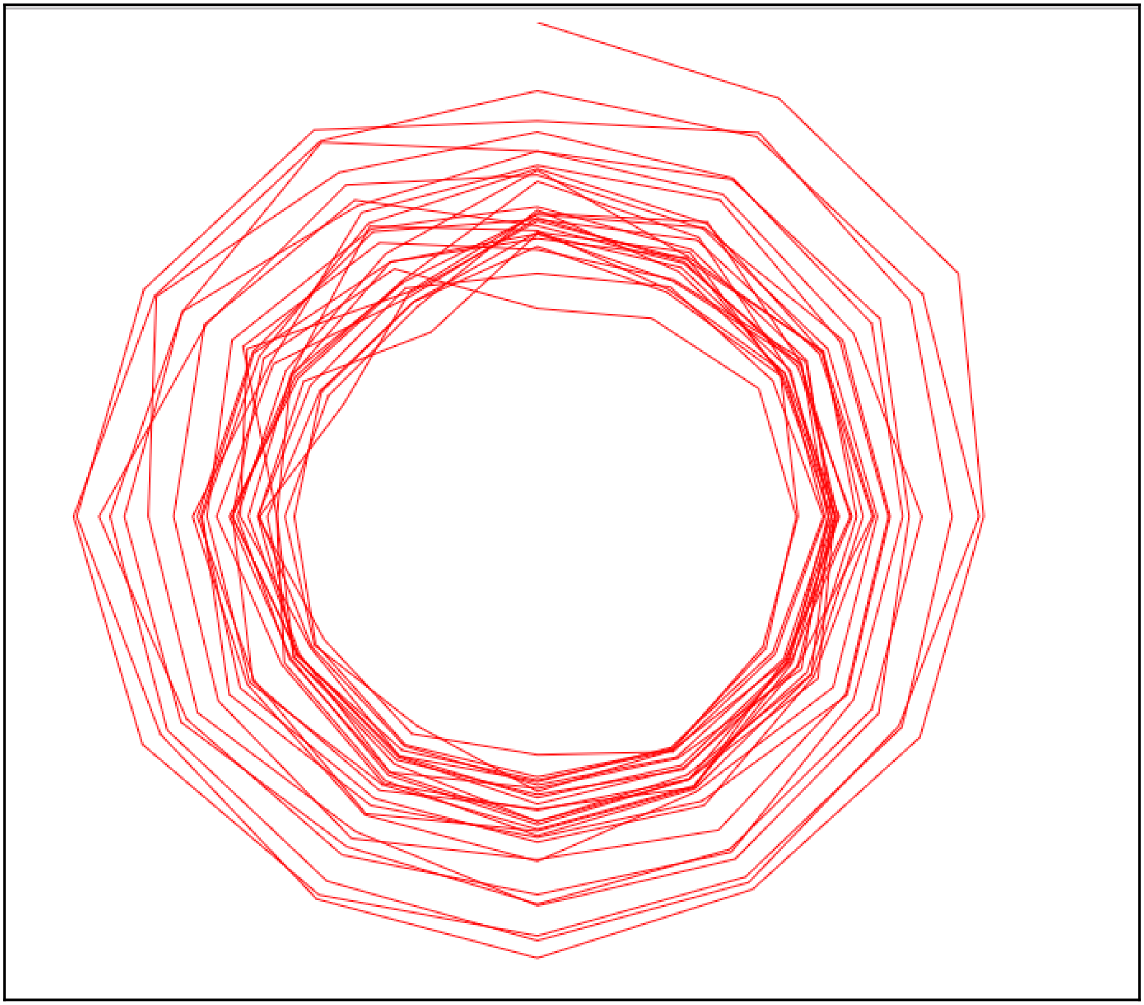
stackOrderNone
stackOrderAscending
stackOrderDescending
stackOrderInsideOut
stackOrderReverse

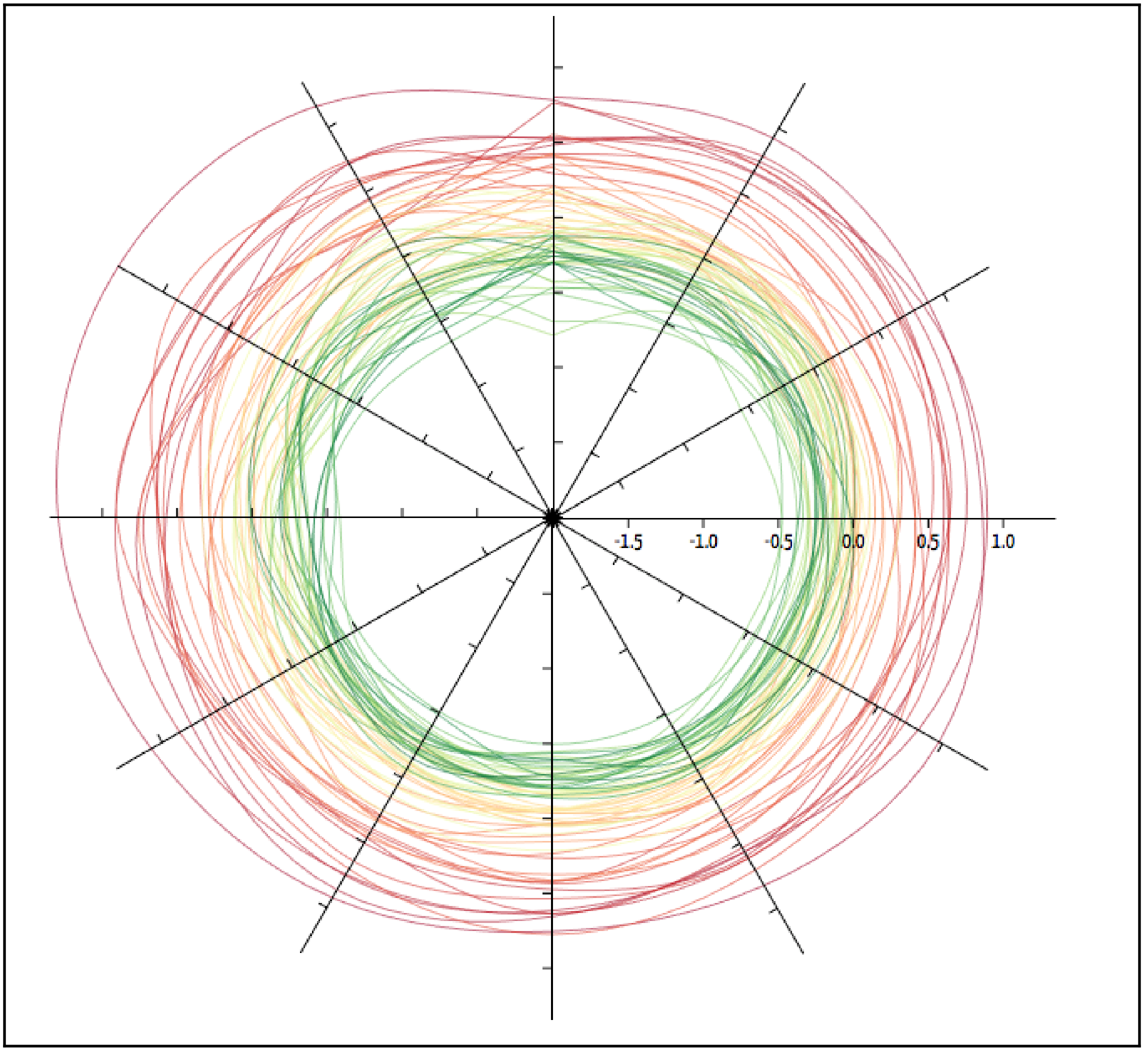

stackOffsetNone
stackOffsetWiggle
stackOffsetExpand (positive values only)
stackOffsetSilhouette
stackOffsetDiverging



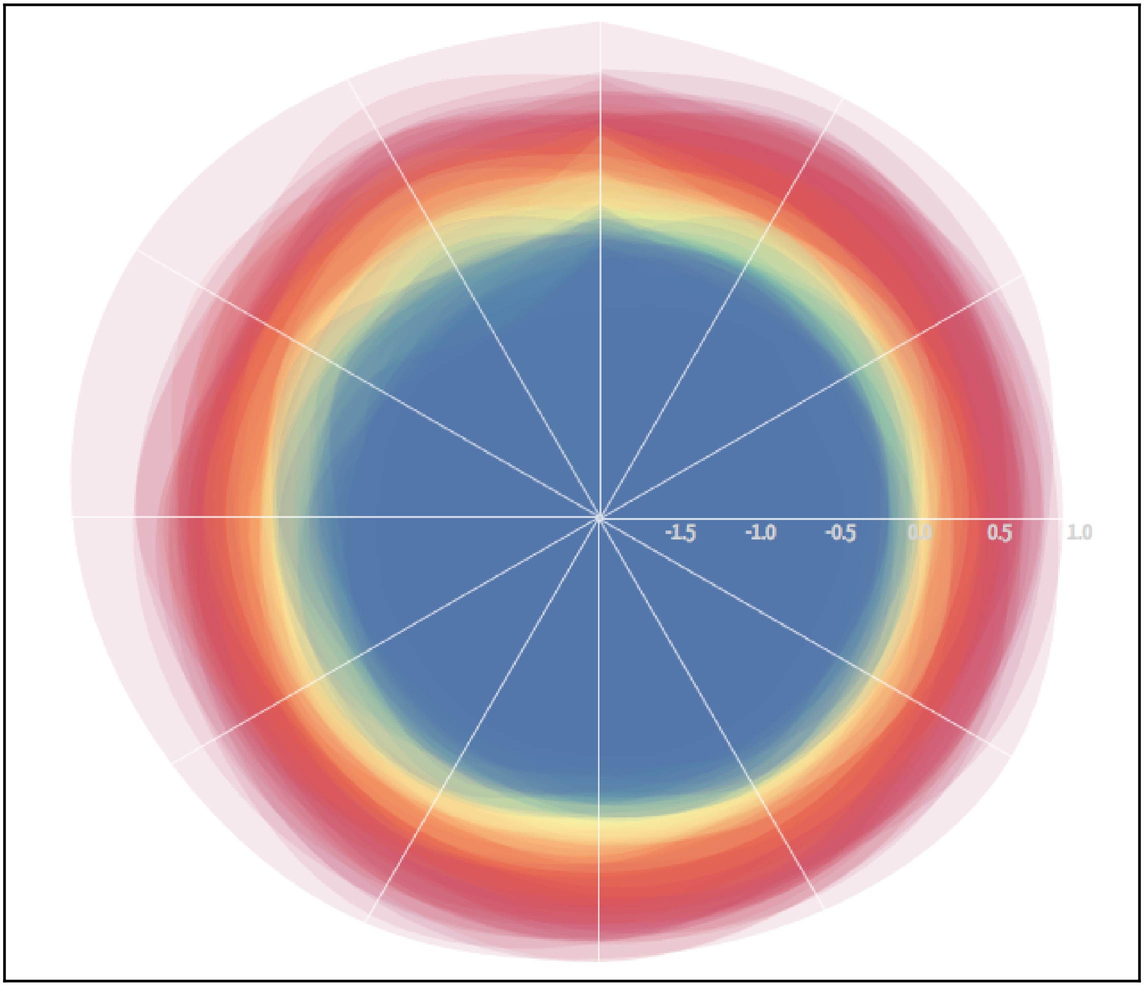
Population growth (1950 to 2000)
350,000
300,000
250,000
200,000
150,000
100,000
50,000
0
1950
1960
1970
1980
1990
2000
Brazil
Argentina
Colombia
Peru
Chile
Venezuela
Ecuador
Bolivia
Uruguay
Paraguay
Guianas

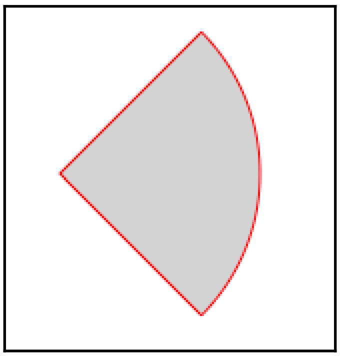

Population growth (1950 to 2000)
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
1950
1960
1970
1980
1990
2000
Brazil
Argentina
Colombia
Peru
Chile
Venezuela
Ecuador
Bolivia
Uruguay
Paraguay
Guianas

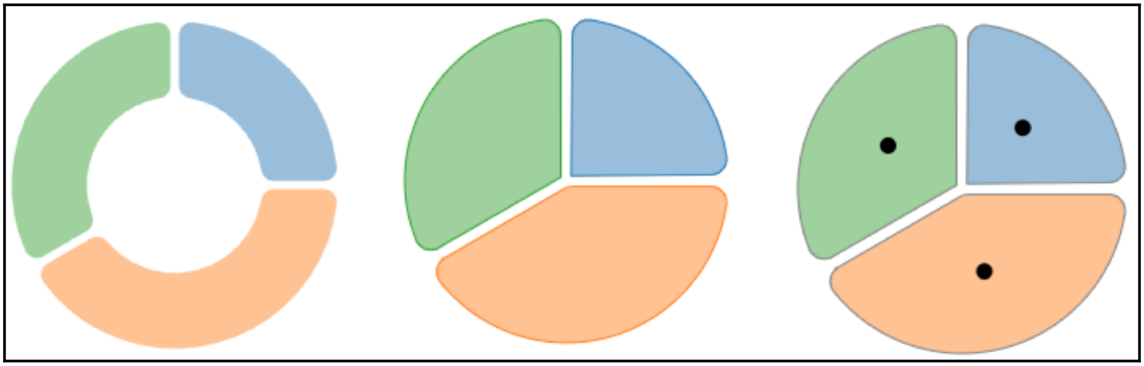


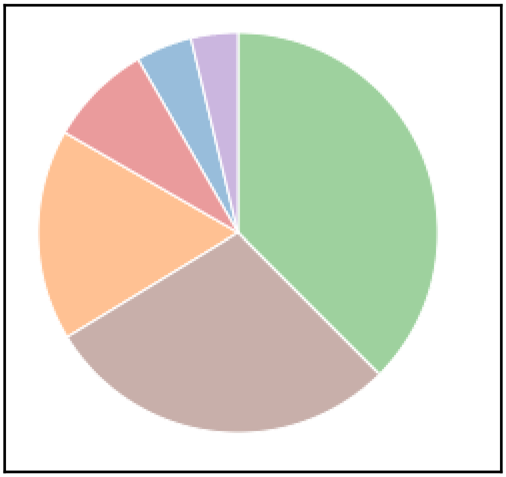


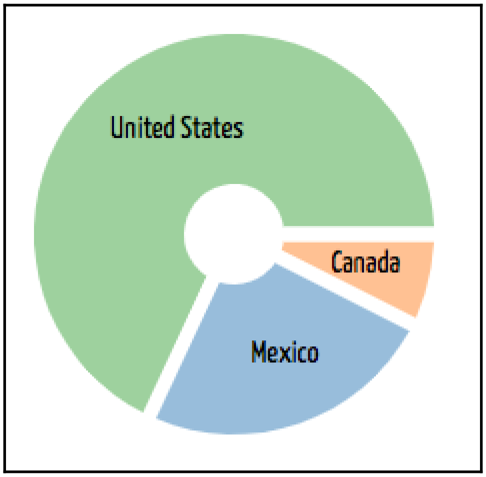
-1.5
-1.0
-0.5
0.0
0.5
1.0


-1.5
-1.0
-0.5
0.0
0.5
1.0

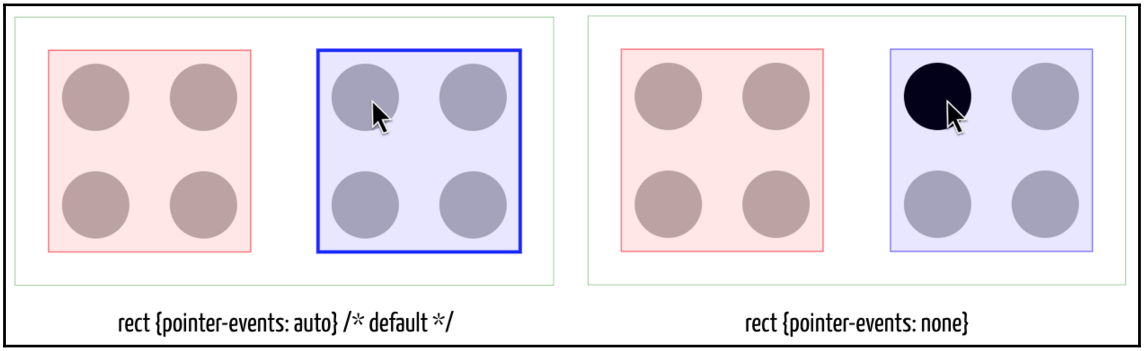







United States
Canada
Mexico

Chapter 8: Animation and Interactivity



document root, origin at (0,0) browser viewport coordinates

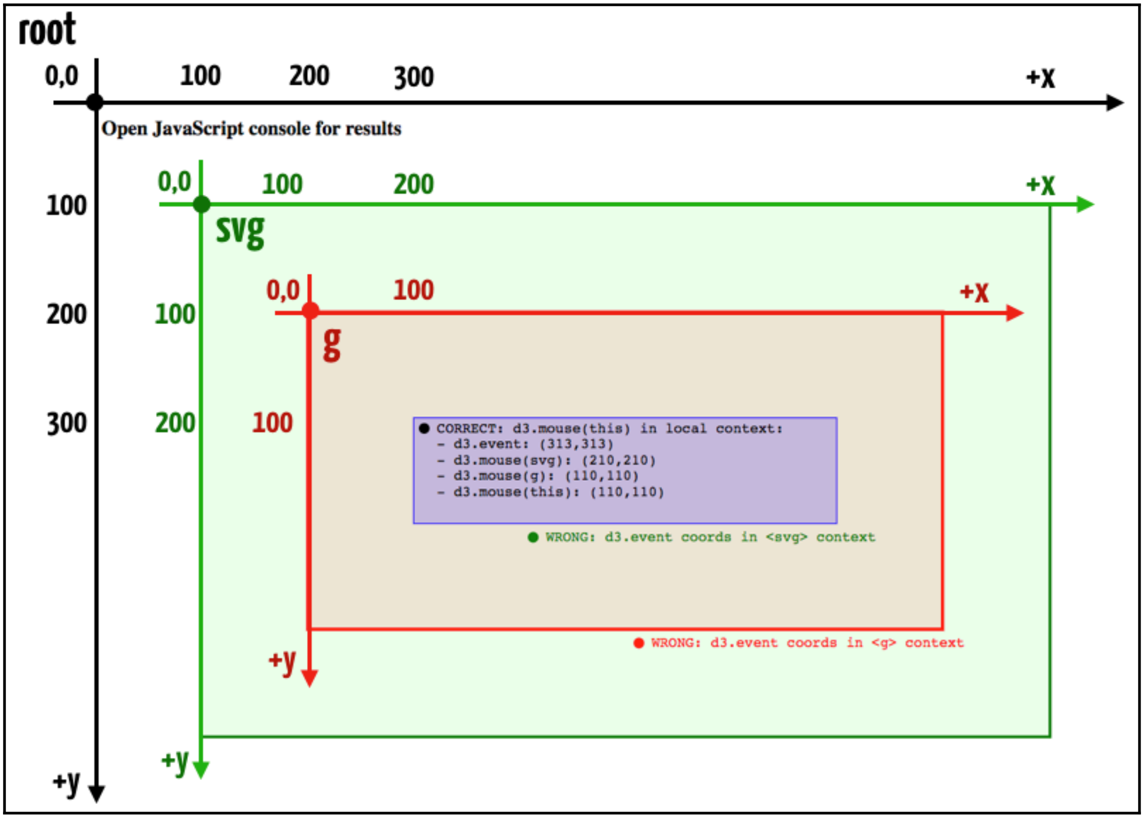
body

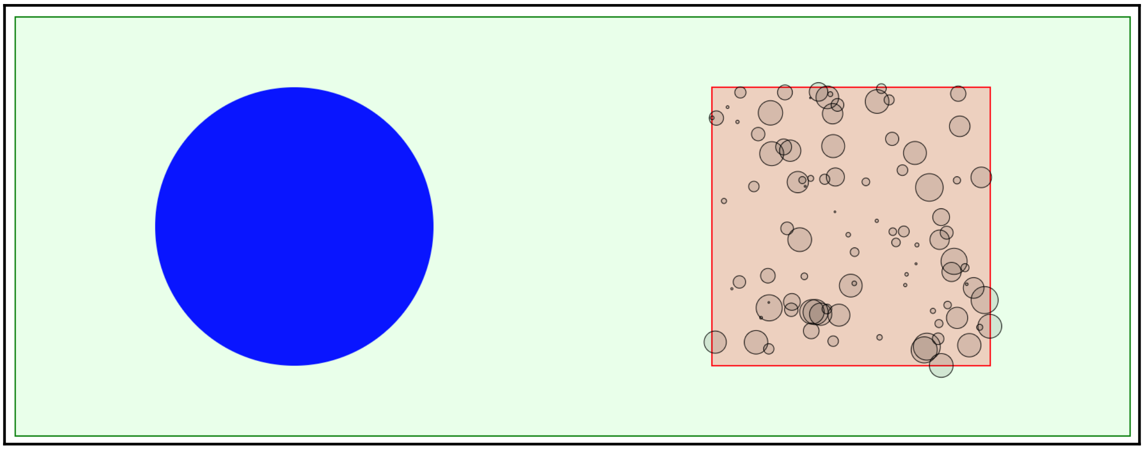
svg: 800x500 px, svg origin at (100,100) browser viewport coordinates

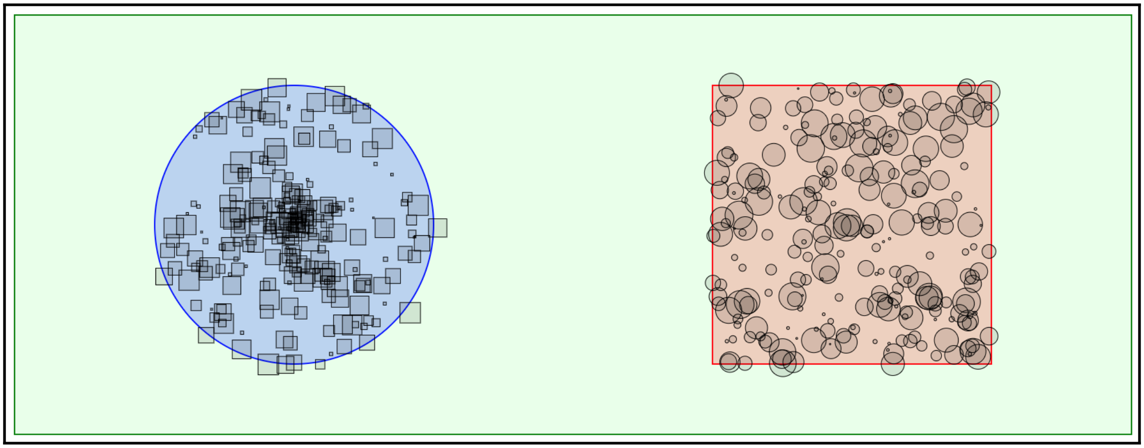
g: group, group origin at (100,100) svg coordinates

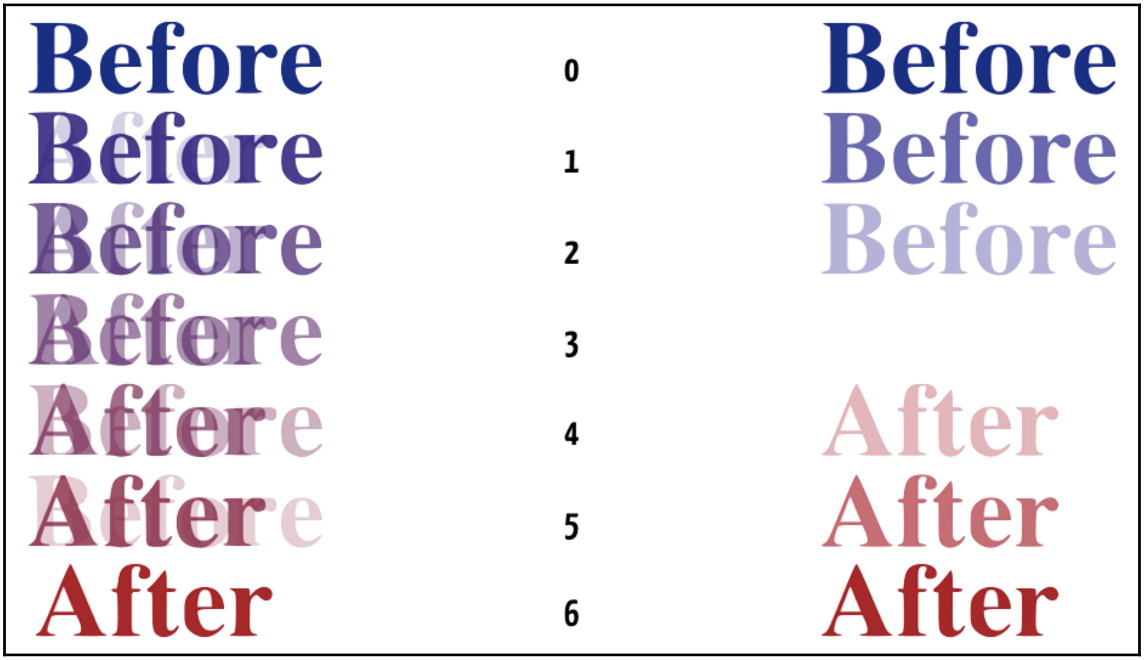
rect: 600x300 px, at (0,0) group coordinates

rect: 400x100 px, at (100,100) group coordinates

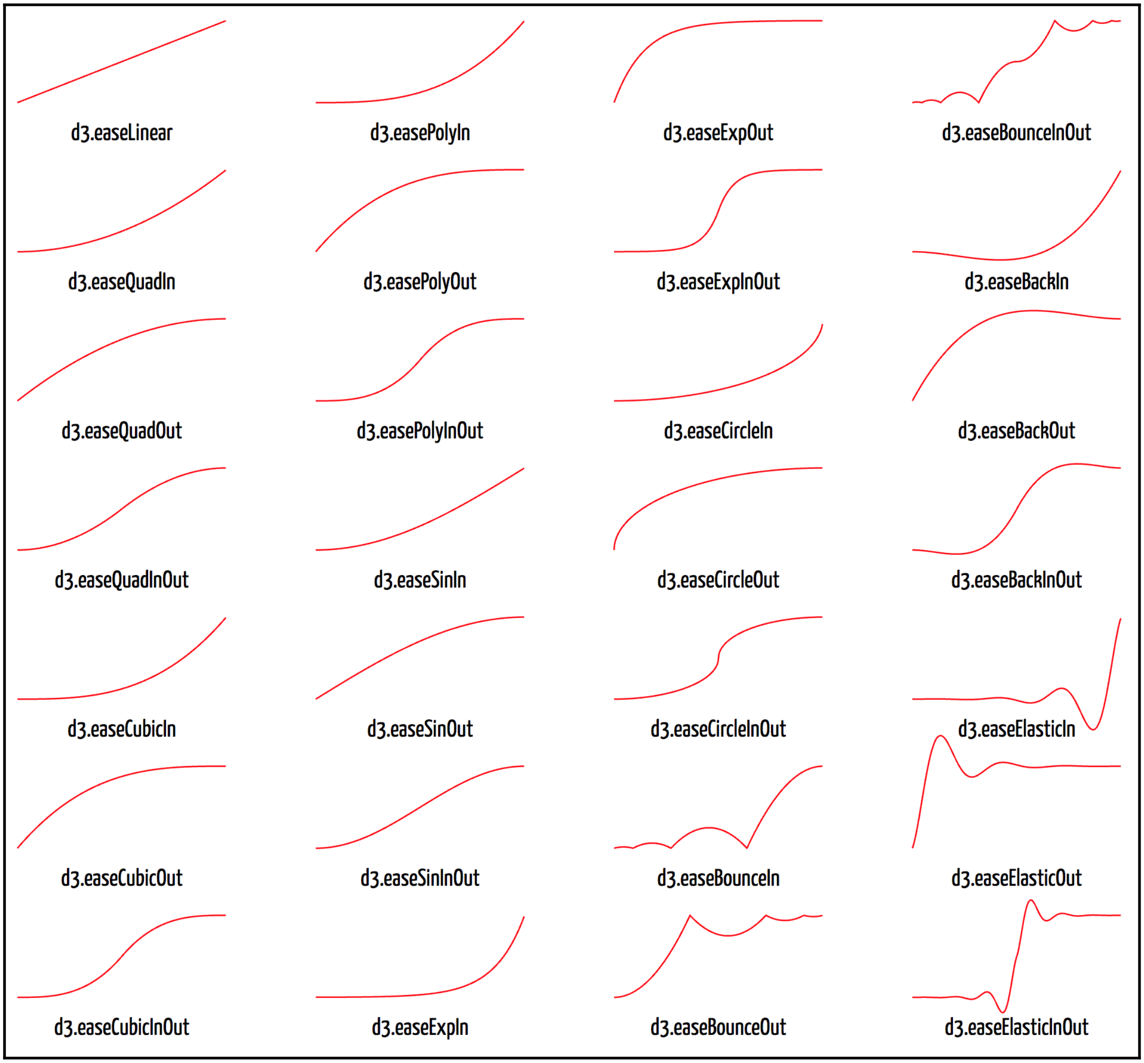

root
0,0
100
200
300
+x
Open JavaScript console for results
100
200
300
+y
0,0
100
200
+x
svg
100
200
+y
0,0
100
+x
g
100
+y
CORRECT: d3.mouse(this) in local context:
- d3.event: (313,313)
- d3.mouse(svg): (210,210)
- d3.mouse(g): (110,110)
- d3.mouse(this): (110,110)
WRONG: d3.event coords in <svg> context
WRONG: d3.event coords in <g> context

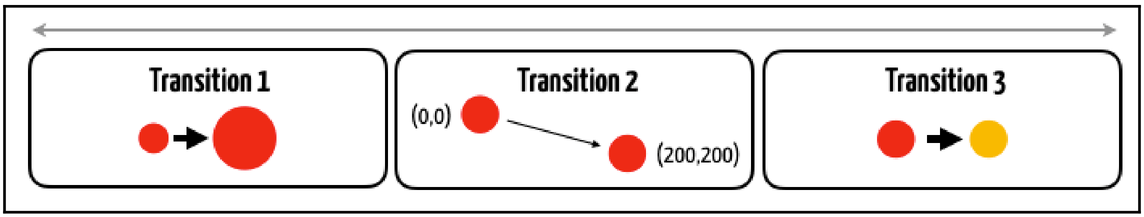


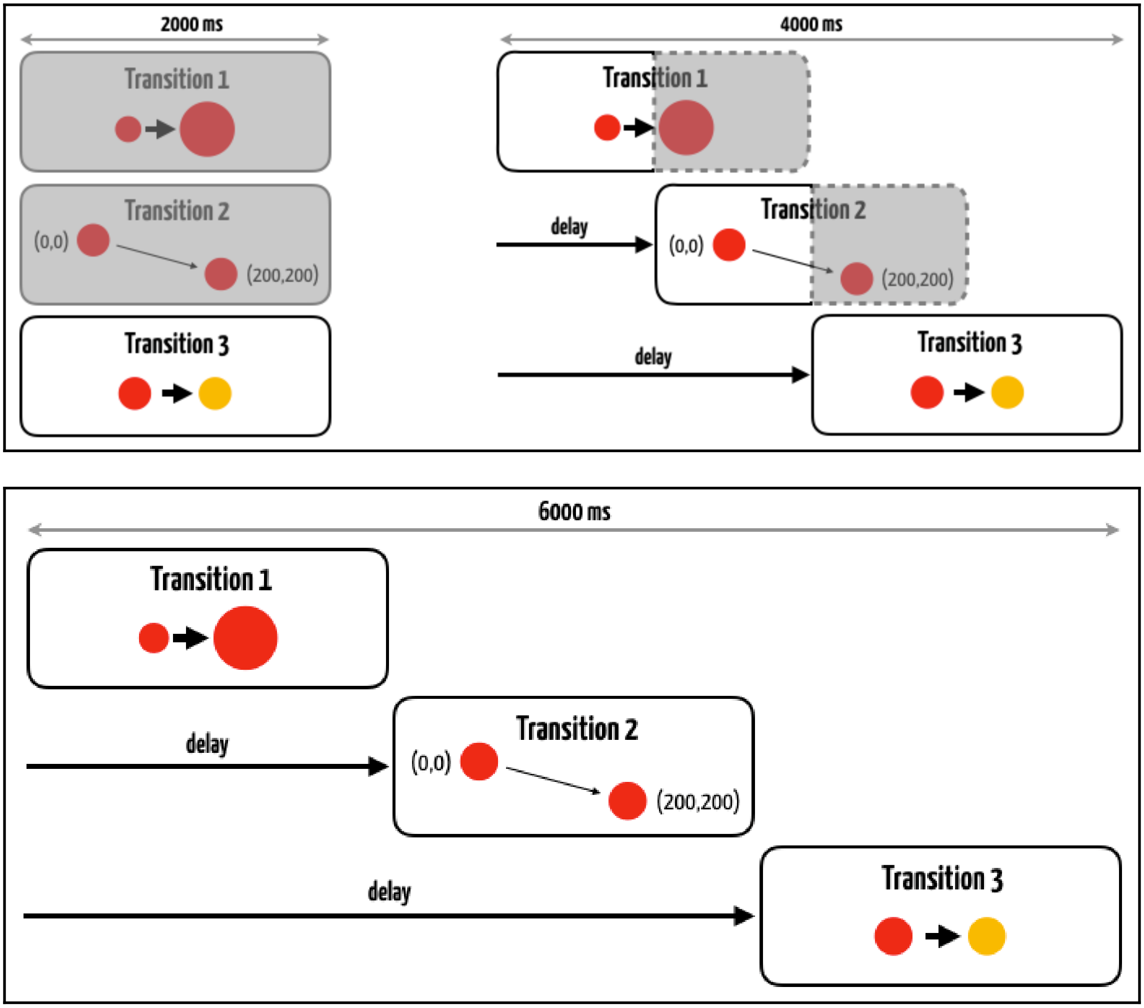


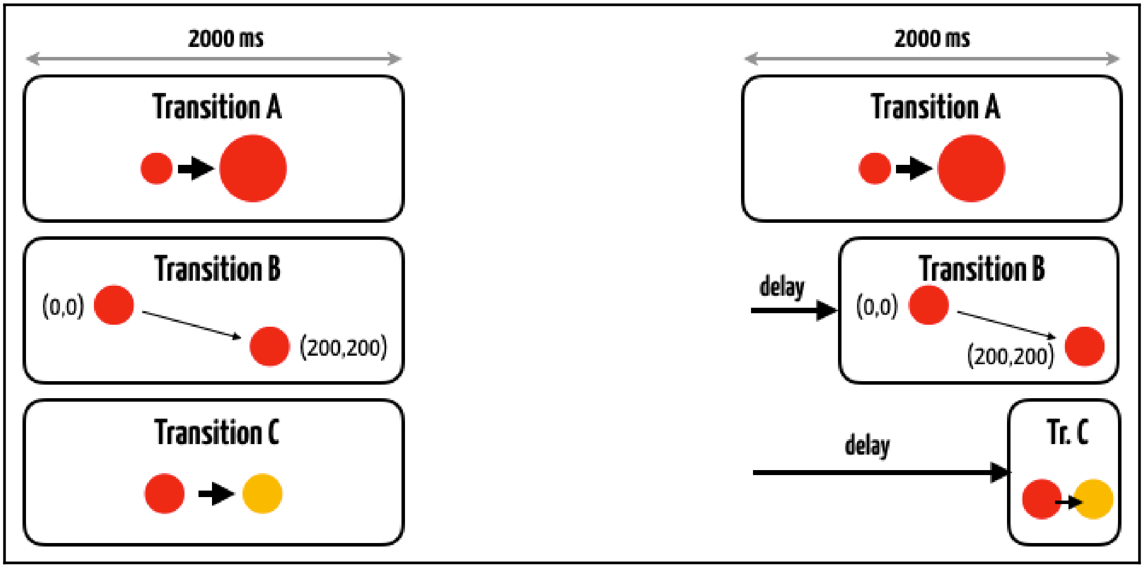


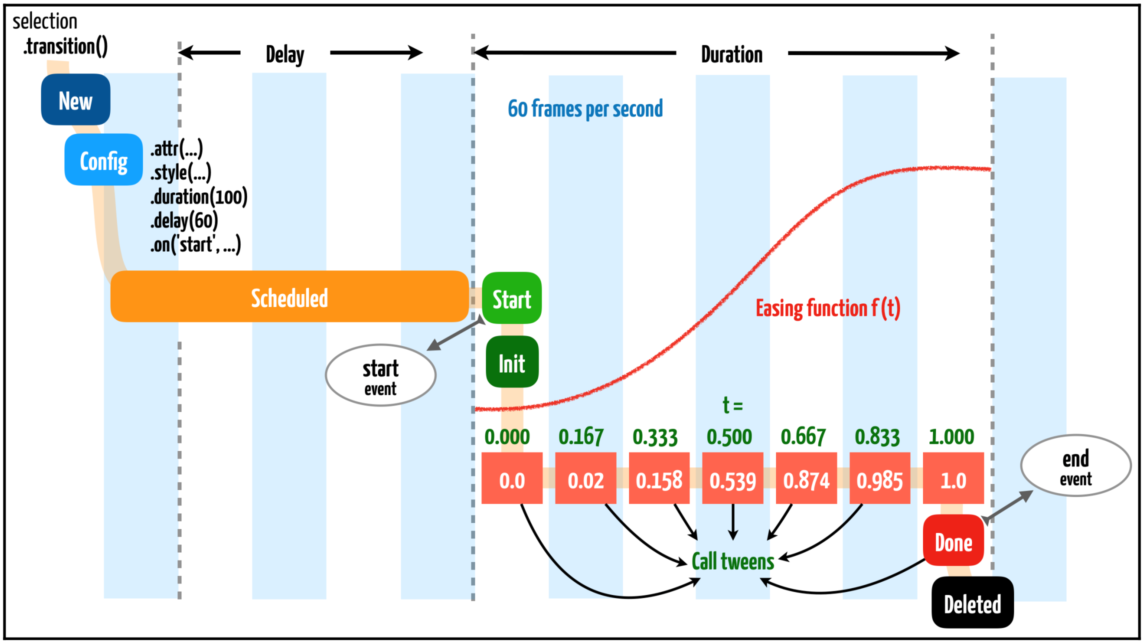
Before
0
Before
Before
1
Before
Before
2
Before
Before
3
After
4
After
After
5
After
After
6
After

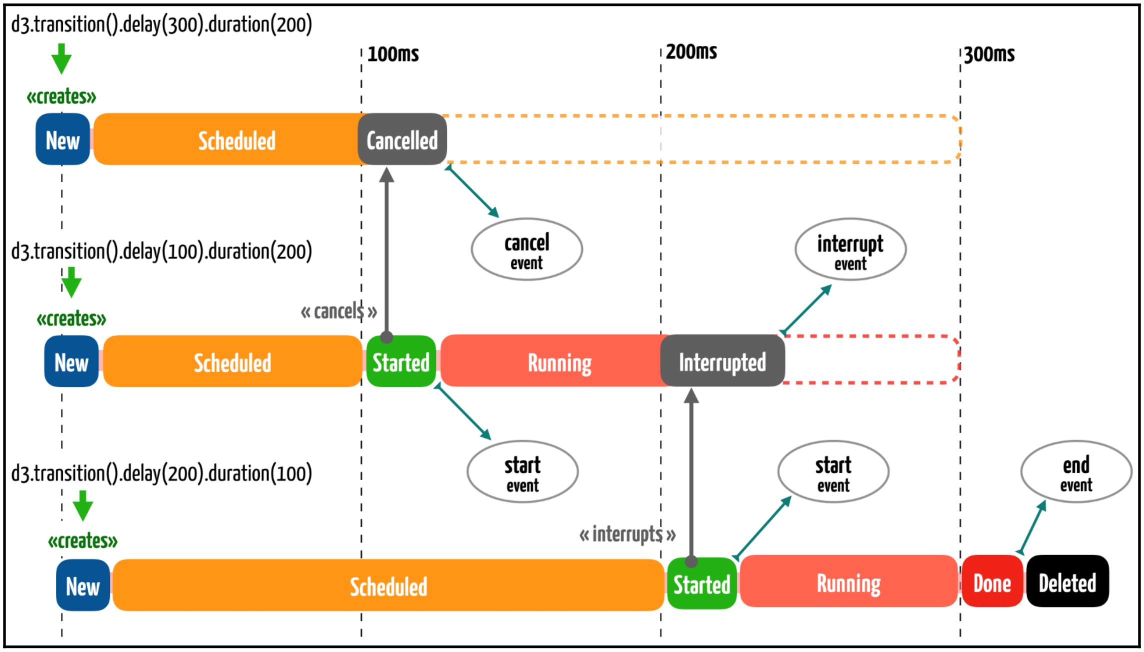

d3.easeLinear
d3.easePolyIn
d3.easeExpOut
d3.easeBounceInOut
d3.easeQuadIn
d3.easePolyOut
d3.easeExpInOut
d3.easeBackIn
d3.easeQuadOut
d3.easePolyInOut
d3.easeCircleIn
d3.easeBackOut
d3.easeQuadInOut
d3.easeSinIn
d3.easeCircleOut
d3.easeBackInOut
d3.easeCubicIn
d3.easeSinOut
d3.easeCircleInOut
d3.easeElasticIn
d3.easeCubicOut
d3.easeSinInOut
d3.easeBounceIn
d3.easeElasticOut
d3.easeCubicInOut
d3.easeExpIn
d3.easeBounceOut
d3.easeElasticInOut


Transition 1
Transition 2
(0,0)
(200,200)
Transition 3

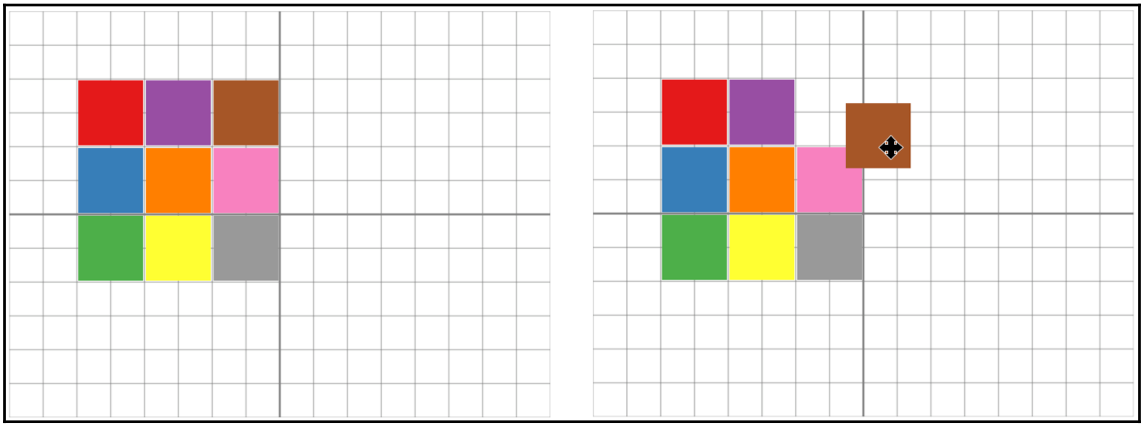

2000 ms
Transition 1
Transition 2
(0,0)
(200,200)
Transition 3
4000 ms
Transition 1
delay
Transition 2
(0,0)
(200,200)
delay
Transition 3
6000 ms
Transition 1
delay
Transition 2
(0,0)
(200,200)
delay
Transition 3

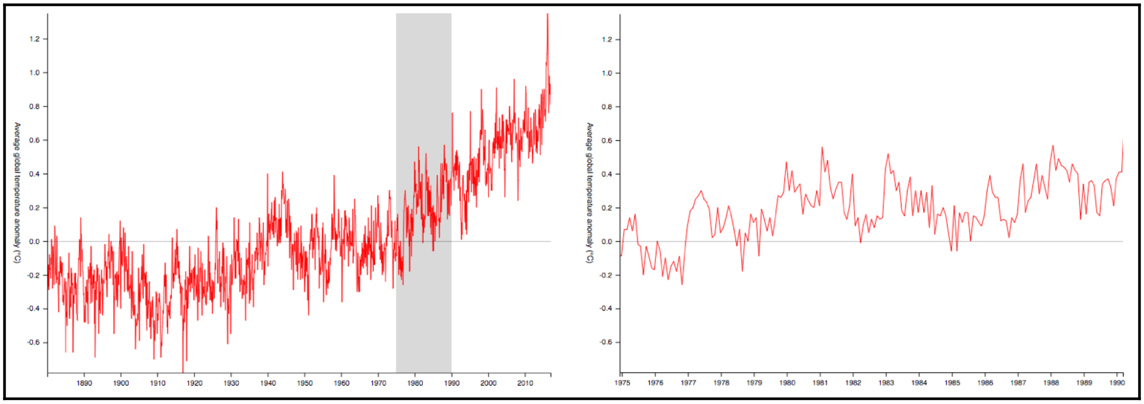

2000 ms
Transition A
Transition B
(0,0)
(200,200)
Transition C
2000 ms
Transition A
delay
Transition B
(0,0)
(200,200)
delay
Tr. C


selection
.transition()
Delay
Duration
New
Config
.attr(...)
.style(...)
.duration(100)
.delay(60)
.on('start', ...)
60 frames per second
Scheduled
Start
start
event
Init
Easing function f (t)
t =
0.000
0.167
0.333
0.500
0.667
0.833
1.000
0.0
0.02
0.158
0.539
0.874
0.985
1.0
end
event
Done
Call tweens
Deleted

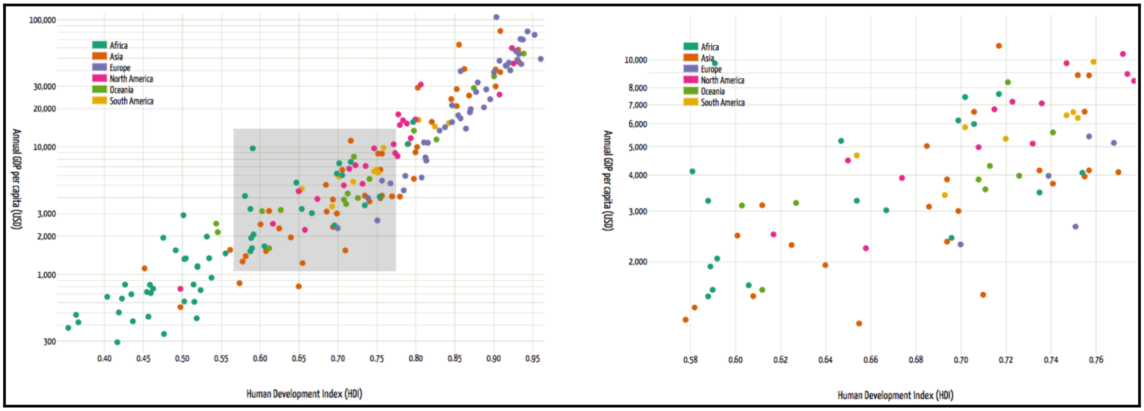


d3.transition().delay(300).duration(200)
«creates»
New
Scheduled
Cancelled
100ms
200ms
300ms
cancel
event
d3.transition().delay(100).duration(200)
«creates»
« cancels »
New
Scheduled
Started
Running
Interrupted
interrupt
event
start
event
d3.transition().delay(200).duration(100)
«creates»
« interrupts »
New
Scheduled
Started
Running
Done
Deleted
start
event
end
event

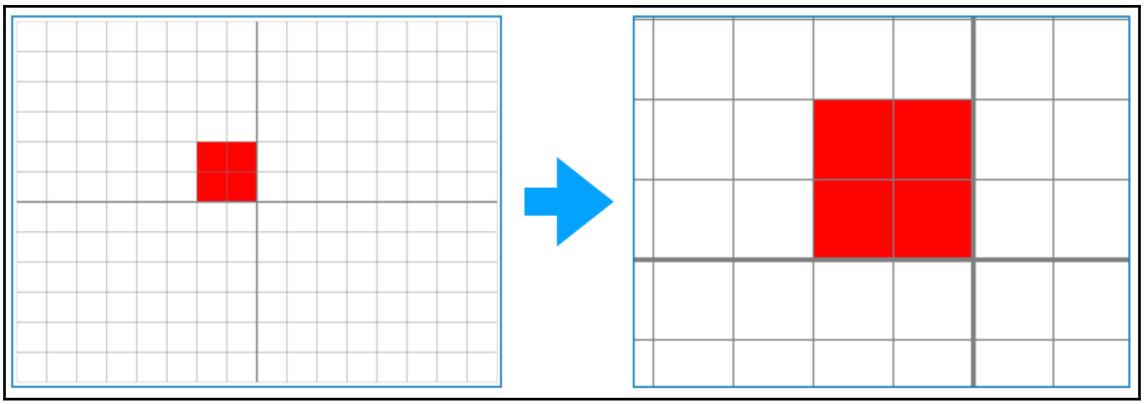


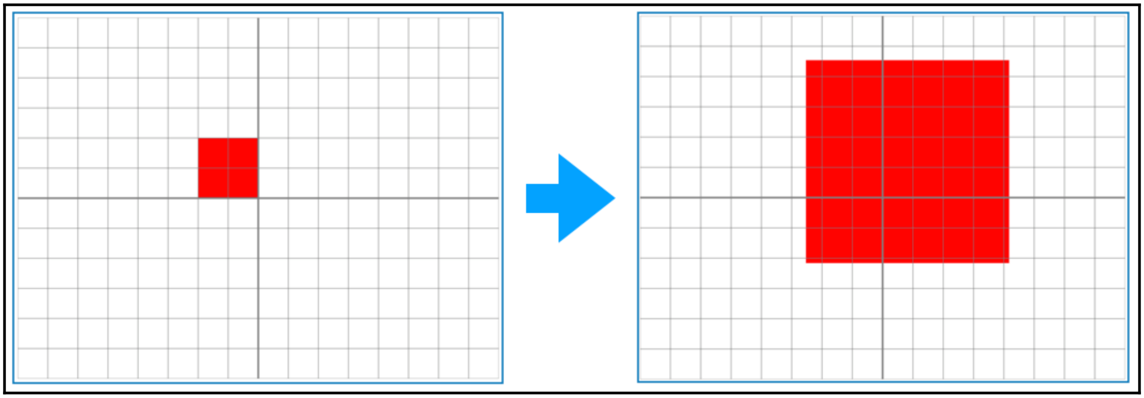


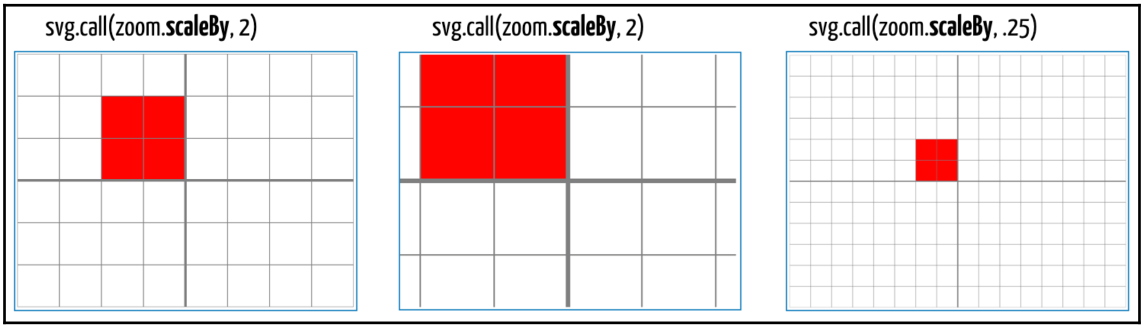
Average global temperature anomaly (°C)
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
1890 1900 1910 1920 1930 1940 1950 1960 1970 1980 1990 2000 2010
1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990

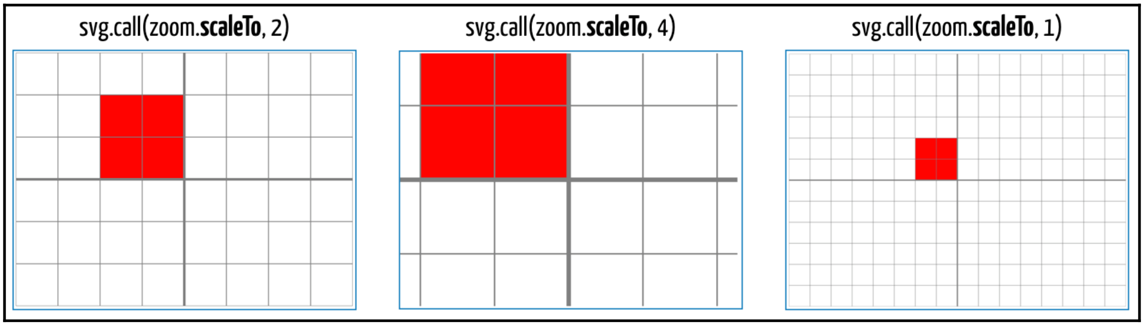


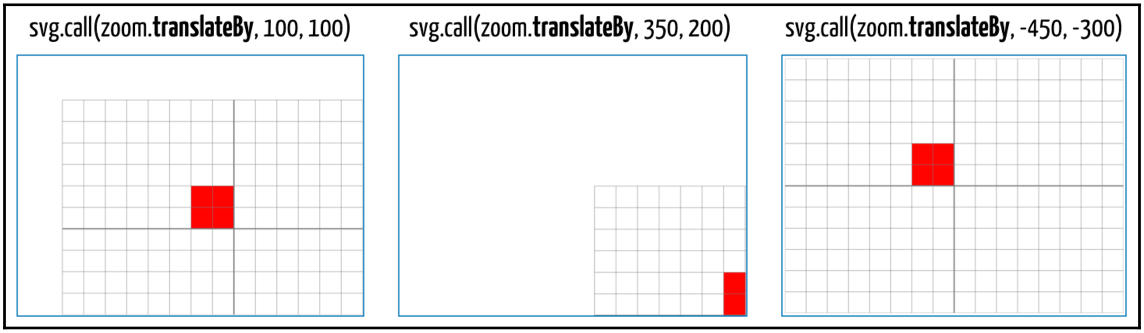
Africa
Asia
Europe
North America
Oceania
South America
Annual GDP per capita (USD)
100,000
30,000
20,000
10,000
3,000
2,000
1,000
300
0.40 0.45 0.50 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95
Human Development Index (HDI)
10,000
8,000
7,000
6,000
5,000
4,000
3,000
2,000
0.58 0.60 0.62 0.64 0.66 0.68 0.70 0.72 0.74 0.76



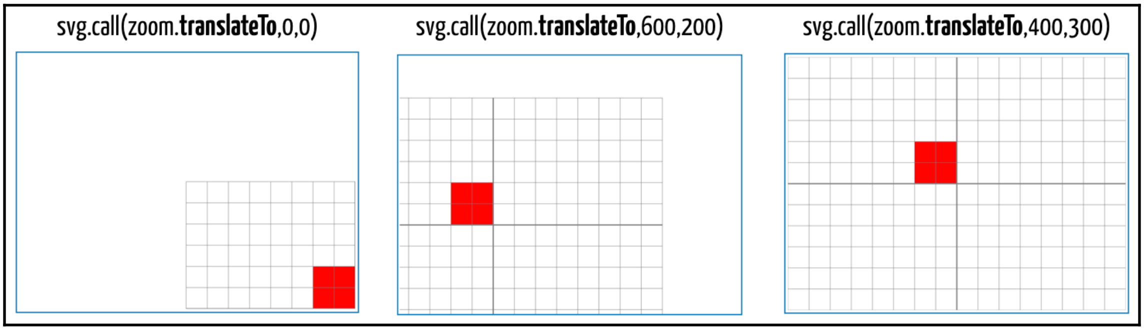



svg.call(zoom.scaleBy, 2)
svg.call(zoom.scaleBy, 2)
svg.call(zoom.scaleBy, .25)

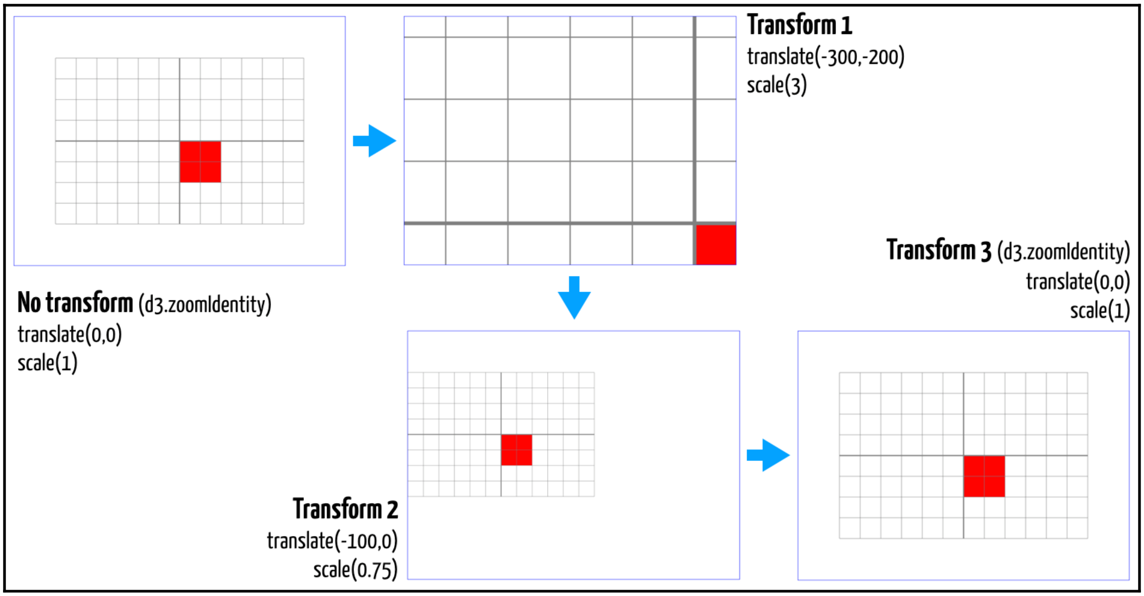

svg.call(zoom.scaleTo, 2)
svg.call(zoom.scaleTo, 4)
svg.call(zoom.scaleTo, 1)



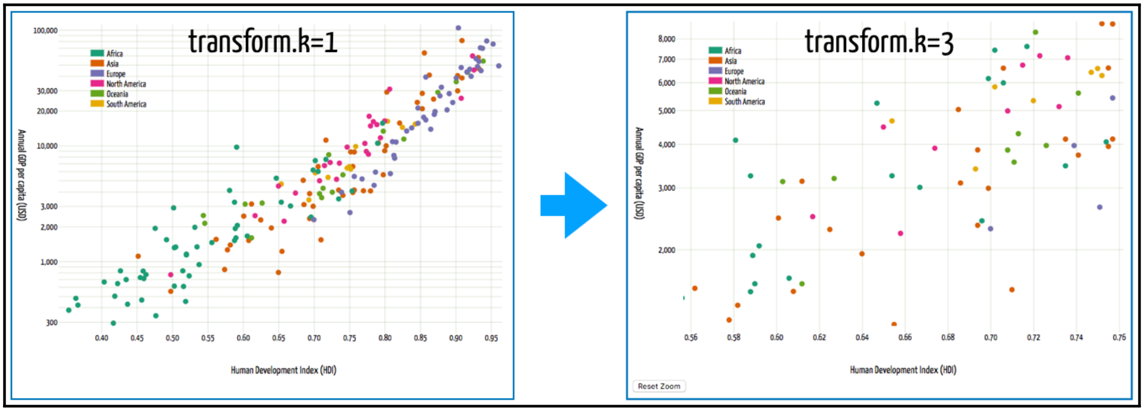
svg.call(zoom.translateBy, 100, 100)
svg.call(zoom.translateBy, 350, 200)
svg.call(zoom.translateBy, -450, -300)

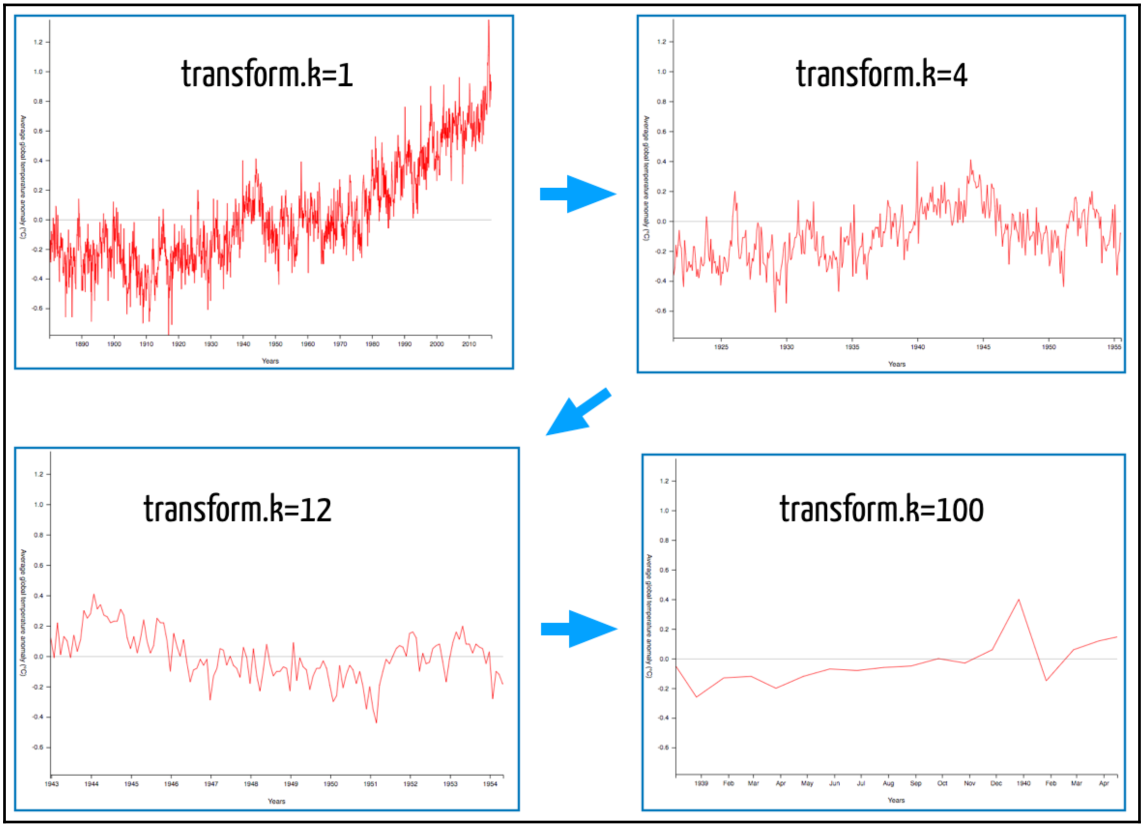


svg.call(zoom.translateTo,0,0)
svg.call(zoom.translateTo,600,200)
svg.call(zoom.translateTo,400,300)

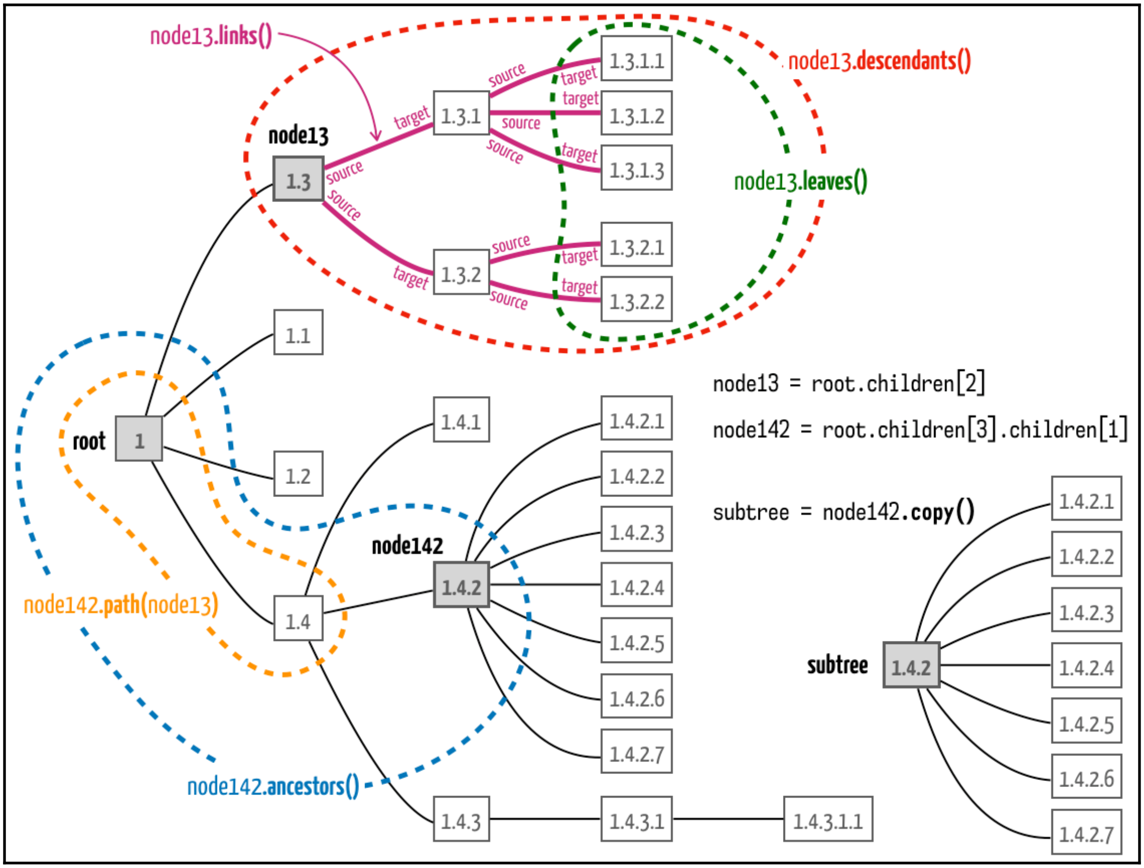


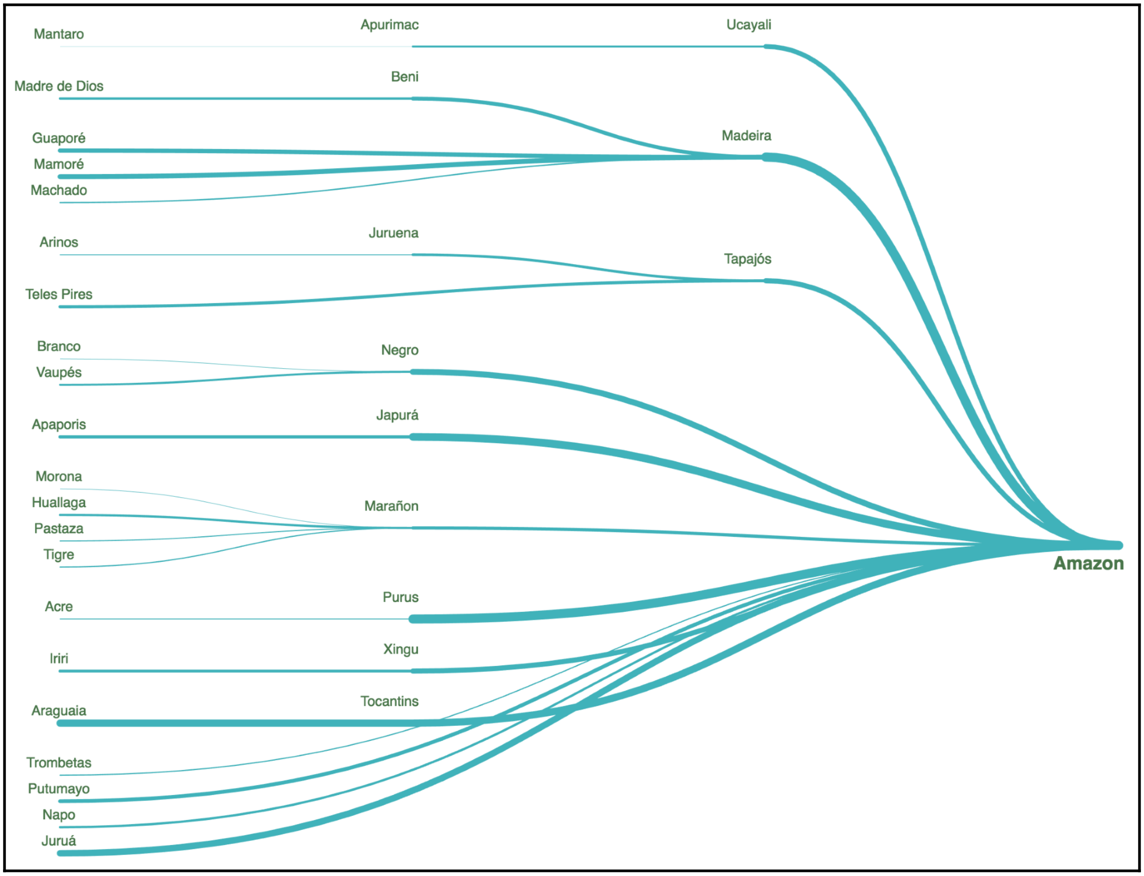
Transform 1
translate(-300,-200)
scale(3)
Transform 3 (d3.zoomIdentity)
translate(0,0)
scale(1)
No transform (d3.zoomIdentity)
translate(0,0)
scale(1)
Transform 2
translate(-100,0)
scale(0.75)

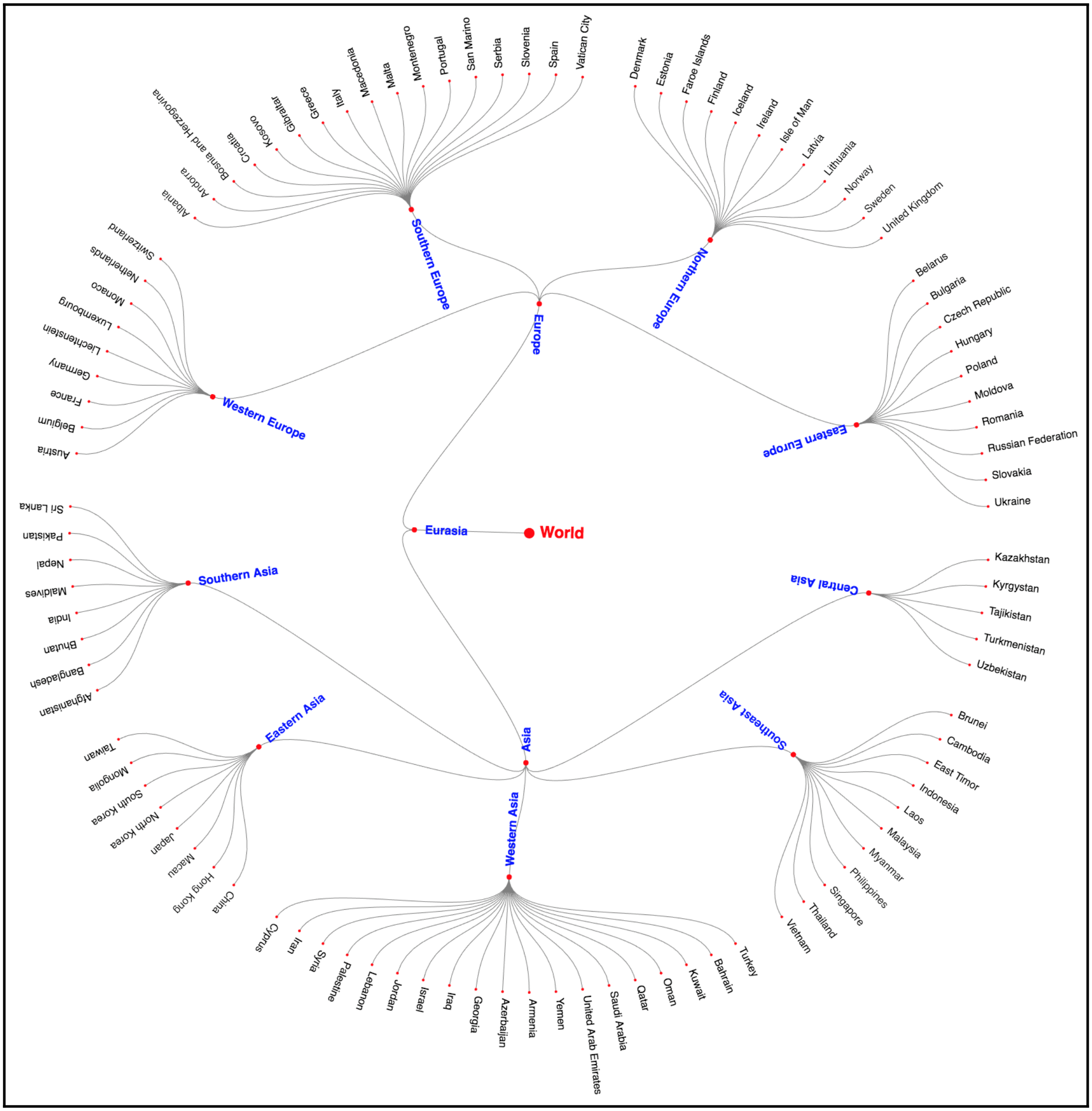

transform.k=1
Africa
Asia
Europe
North America
Oceania
South America
Annual GDP per capita (USD)
Human Development Index (HDI)
transform.k=3
Reset Zoom

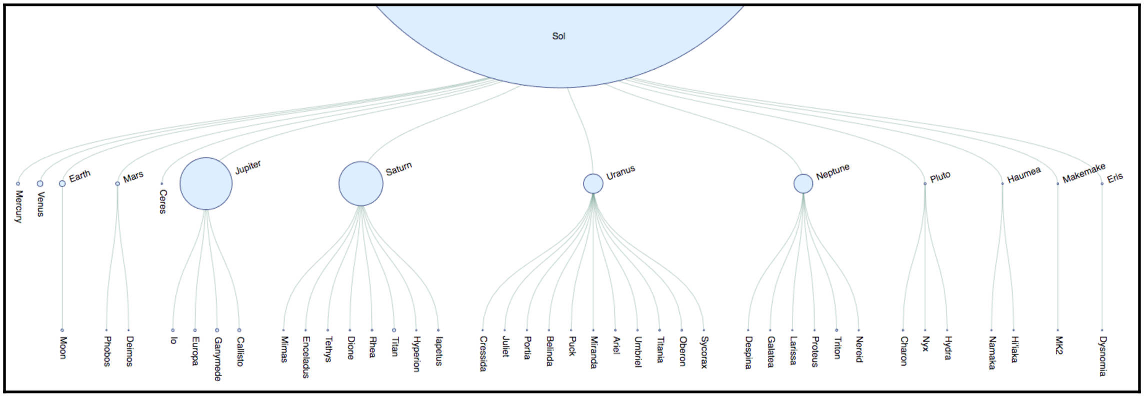

transform.k=1
transform.k=4
transform.k=12
transform.k=100
Average global temperature anomaly (°C)
Years

Chapter 9: Visualizing Hierarchical Data

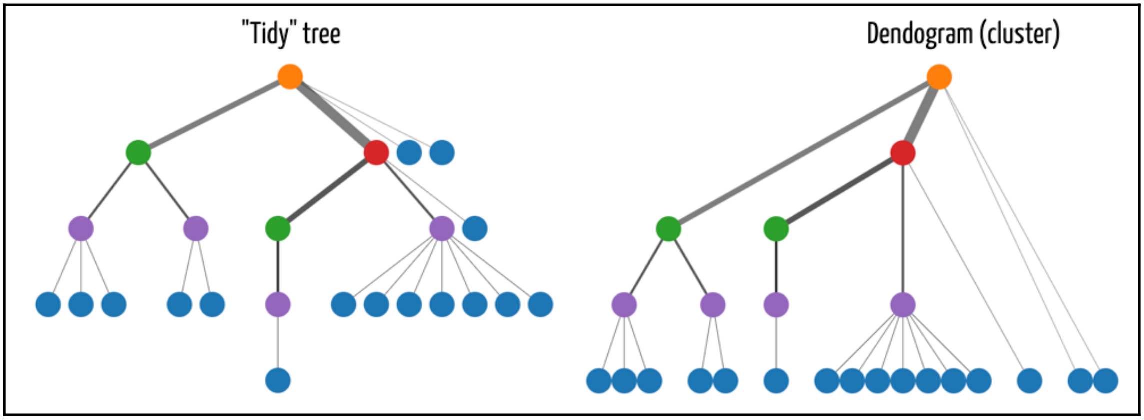



Mantaro
Apurimac
Ucayali
Madre de Dios
Beni
Guaporé
Madeira
Mamoré
Machado
Arinos
Juruena
Tapajós
Teles Pires
Branco
Negro
Vaupés
Apaporis
Japurá
Morona
Huallaga
Marañon
Pastaza
Tigre
Amazon
Acre
Purus
Iriri
Xingu
Araguaia
Tocantins
Trombetas
Putumayo
Napo
Juruá


World
Eurasia
Europe
Asia
Southern Europe
Albania
Andorra
Bosnia and Herzegovina
Croatia
Kosovo
Gibraltar
Greece
Italy
Macedonia
Malta
Montenegro
Portugal
San Marino
Serbia
Slovenia
Spain
Vatican City
Northern Europe
Denmark
Estonia
Faroe Islands
Finland
Iceland
Ireland
Isle of Man
Latvia
Lithuania
Norway
Sweden
United Kingdom
Western Europe
Switzerland
Netherlands
Monaco
Luxembourg
Liechtenstein
Germany
France
Belgium
Austria
Eastern Europe
Belarus
Bulgaria
Czech Republic
Hungary
Poland
Moldova
Romania
Russian Federation
Slovakia
Ukraine
Southern Asia
Sri Lanka
Pakistan
Nepal
Maldives
India
Bhutan
Bangladesh
Afghanistan
Central Asia
Kazakhstan
Kyrgystan
Tajikistan
Turkmenistan
Uzbekistan
Eastern Asia
Taiwan
Mongolia
South Korea
North Korea
Japan
Macau
Hong Kong
China
Southeast Asia
Brunei
Cambodia
East Timor
Indonesia
Laos
Malaysia
Myanmar
Philippines
Singapore
Thailand
Vietnam
Western Asia
Cyprus
Iran
Syria
Palestine
Lebanon
Jordan
Israel
Iraq
Georgia
Azerbaijan
Armenia
Yemen
United Arab Emirates
Saudi Arabia
Qatar
Oman
Kuwait
Bahrain
Turkey

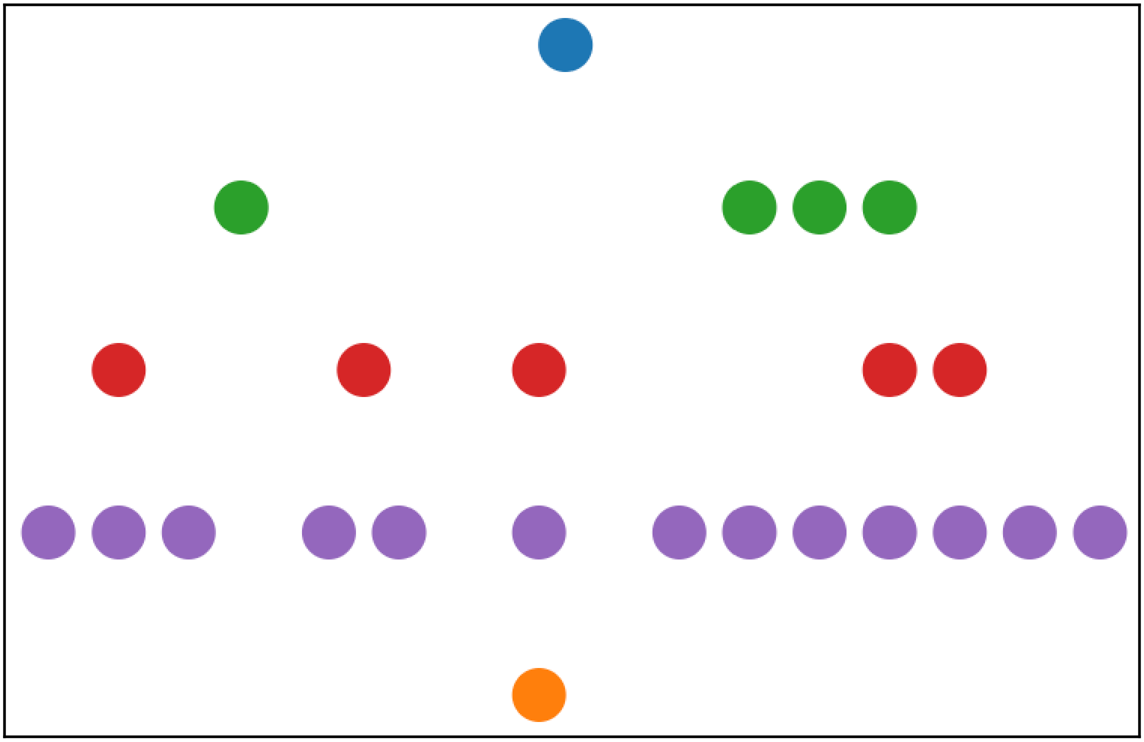


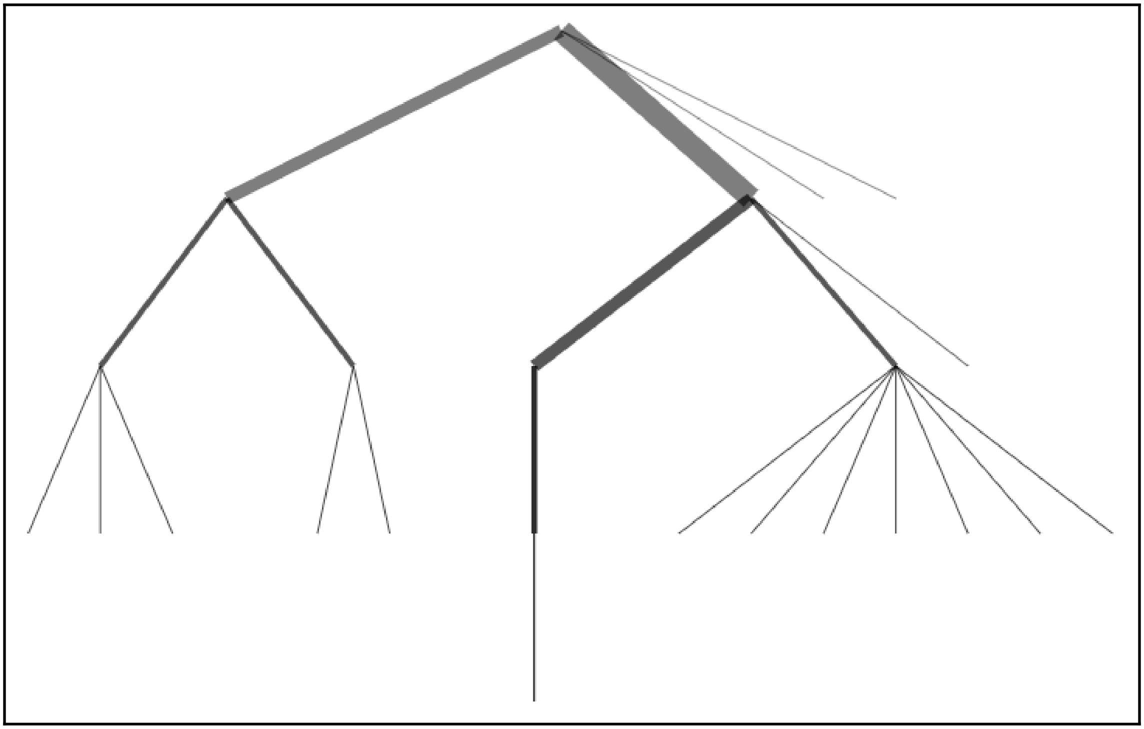
Sol
Mercury
Venus
Earth
Mars
Ceres
Jupiter
Saturn
Uranus
Neptune
Pluto
Haumea
Makemake
Eris
Moon
Phobos
Deimos
Io
Europa
Ganymede
Callisto
Mimas
Enceladus
Tethys
Dione
Rhea
Titan
Hyperion
Iapetus
Cressida
Juliet
Portia
Belinda
Puck
Miranda
Ariel
Umbriel
Titania
Oberon
Sycorax
Despina
Galatea
Larissa
Proteus
Triton
Nereid
Charon
Nyx
Hydra
Namaka
Hi'iaka
MK2
Dysnomia

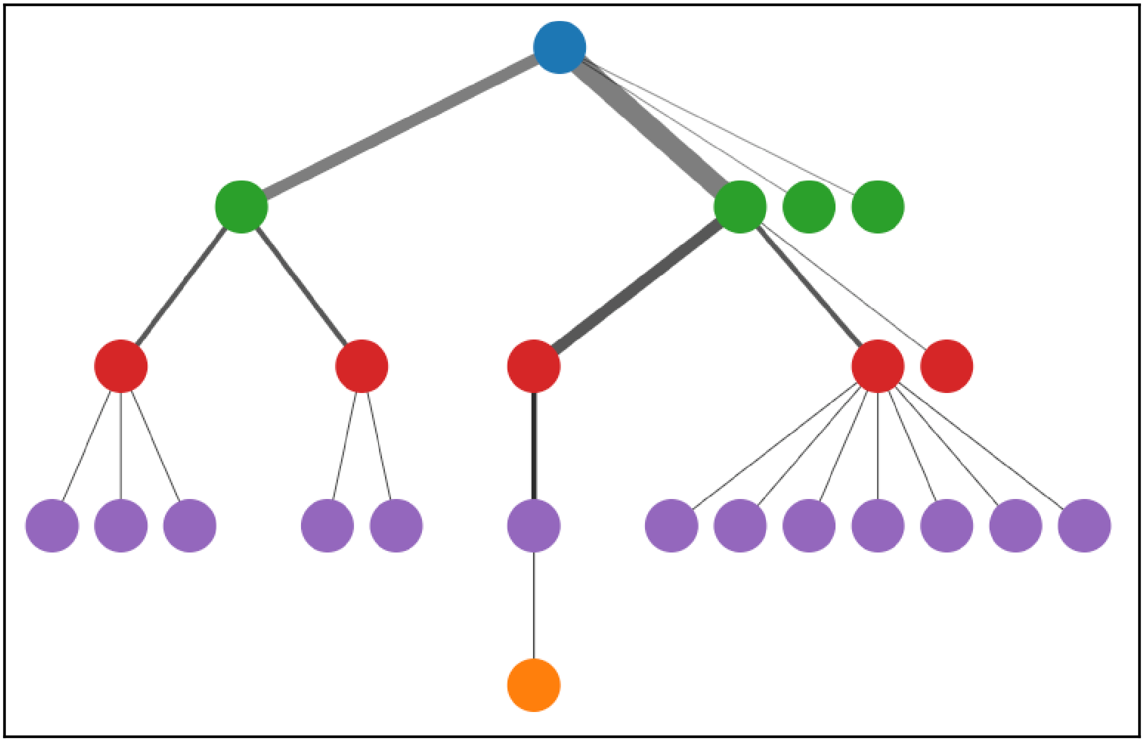


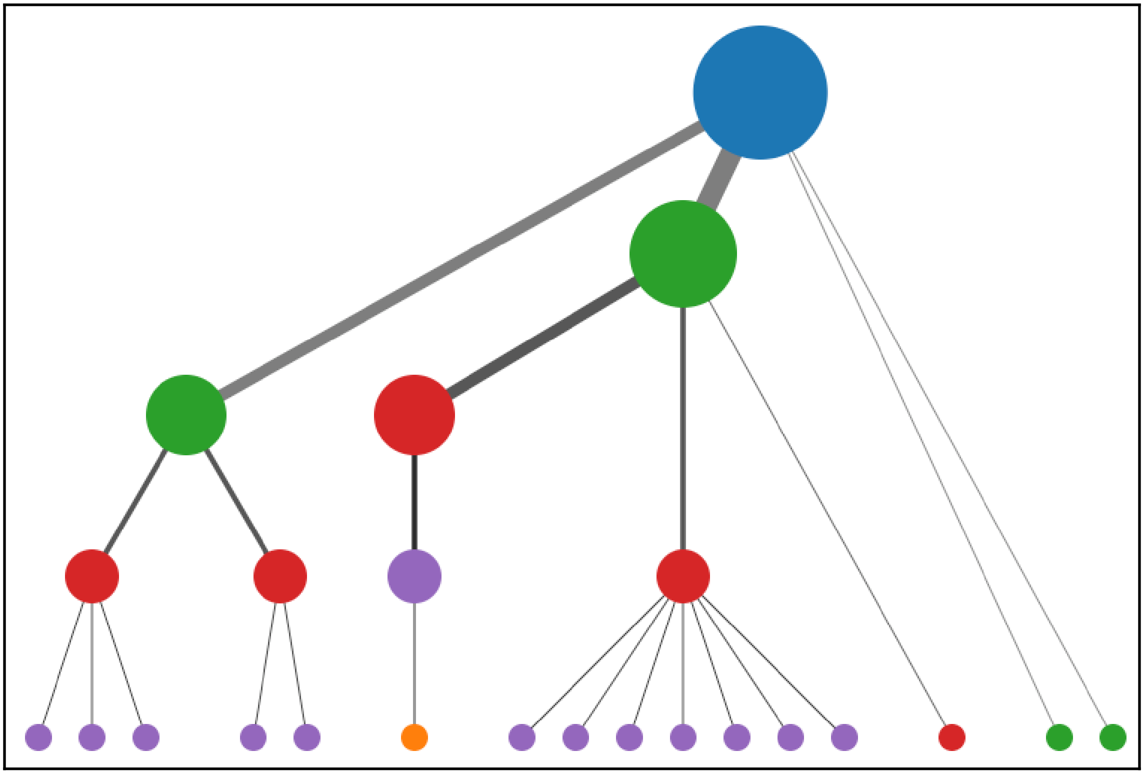
"Tidy" tree
Dendogram (cluster)

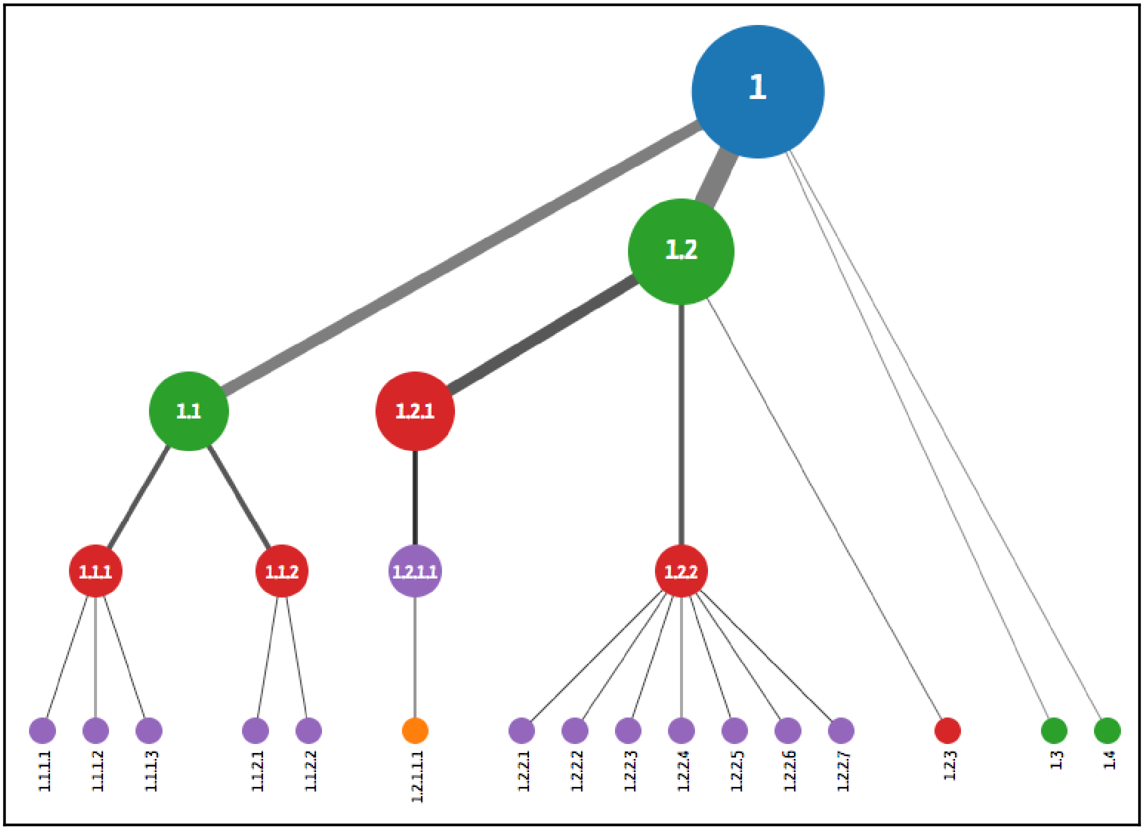
```
▼hl
  ▼children: Array(4)
    ▼0: hl
      ▶children: (2) [hl, hl]
      ▶data: {id: "group_1", name: "First Group", children: Array(2)}
       depth: 1
       height: 2
      ▶parent: hl {data: {…}, height: 4, depth: 0, parent: null, children: Array(4), …}
       x: 194.11764705882354
       y: 120
      ▶__proto__: Object
    ▶1: hl {data: {…}, height: 3, depth: 1, parent: hl, children: Array(3), …}
    ▶2: hl {data: {…}, height: 0, depth: 1, parent: hl, x: 621.1764705882354, …}
    ▶3: hl {data: {…}, height: 0, depth: 1, parent: hl, x: 672.9411764705883, …}
     length: 4
    ▶__proto__: Array(0)
  ▶data: {id: "root", name: "Root Level", children: Array(4)}
   depth: 0
   height: 4
   parent: null
   x: 433.5294117647059
   y: 0
```



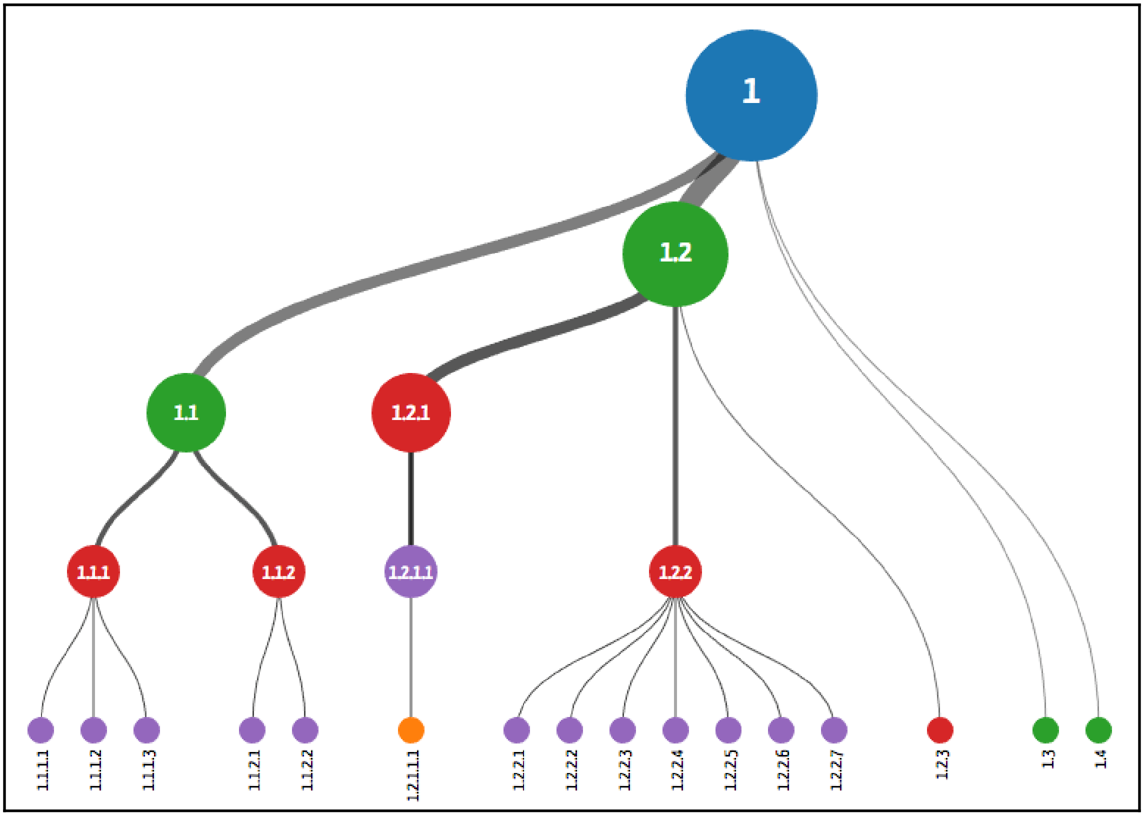


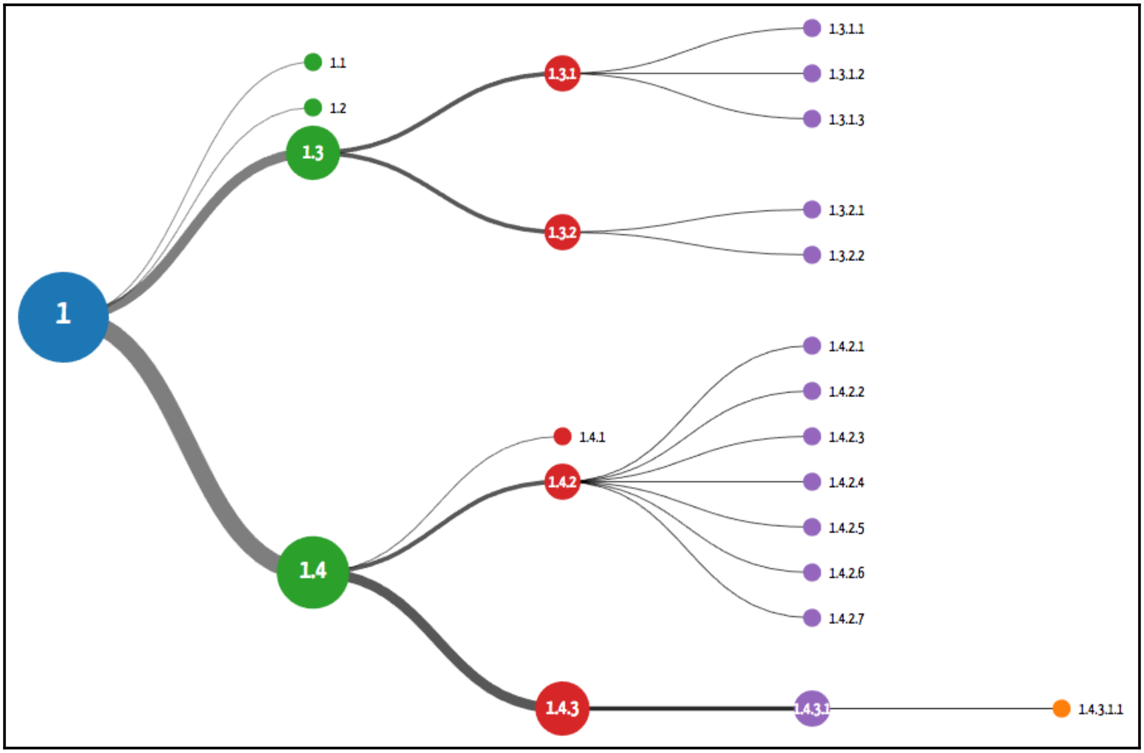




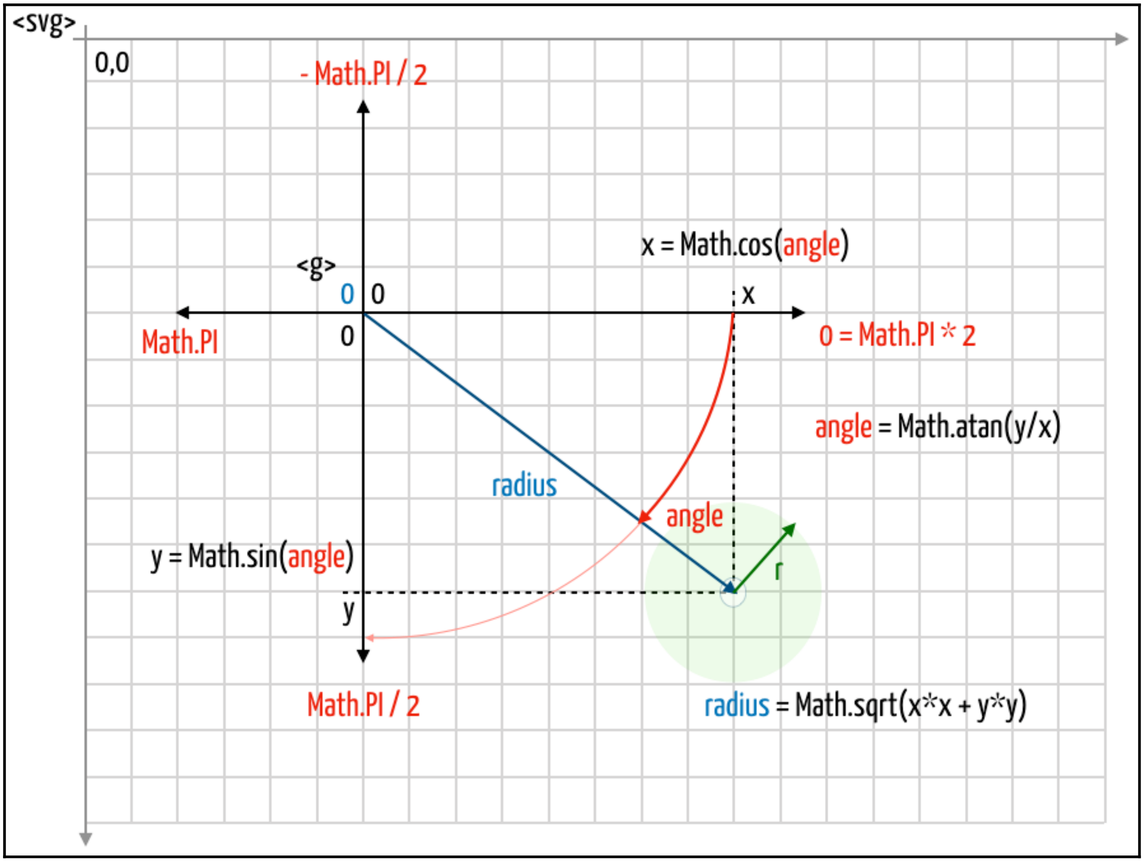


1
1.2
1.1
1.2.1
1.1.1
1.1.2
1.2.1.1
1.2.2
1.1.1.1
1.1.1.2
1.1.1.3
1.1.2.1
1.1.2.2
1.2.1.1.1
1.2.2.1
1.2.2.2
1.2.2.3
1.2.2.4
1.2.2.5
1.2.2.6
1.2.2.7
1.2.3
1.3
1.4

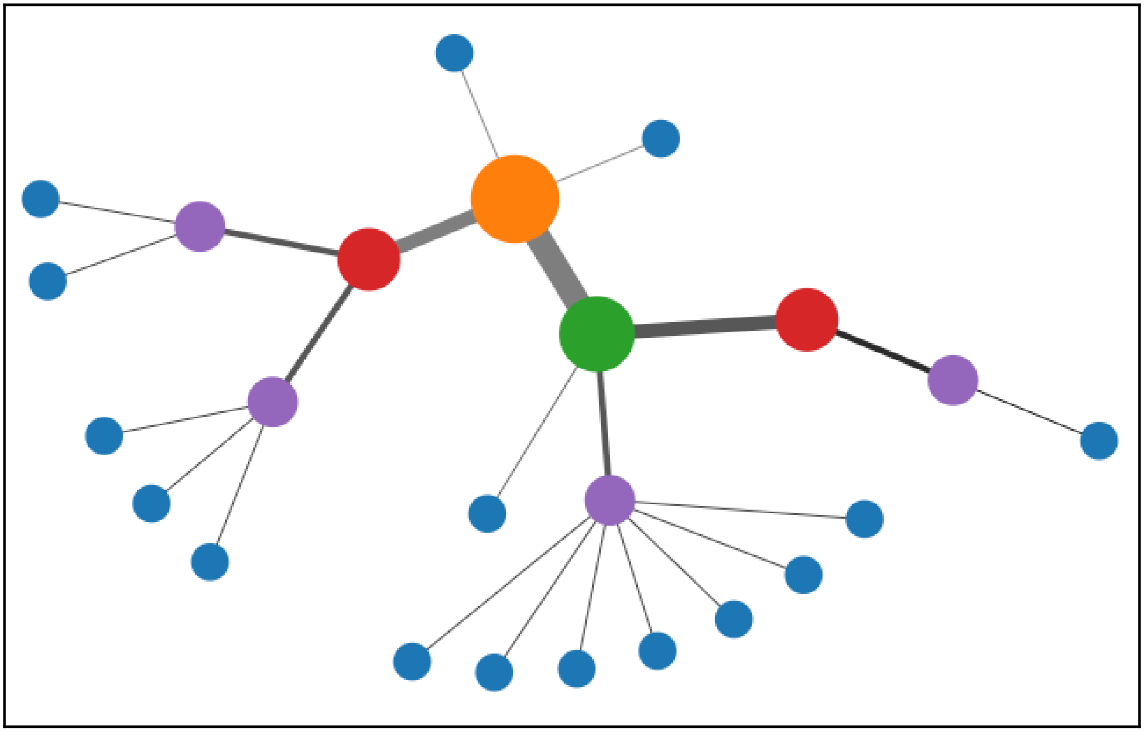

1
1.1
1.2
1.2.1
1.1.1
1.1.2
1.2.1.1
1.2.2
1.1.1.1
1.1.1.2
1.1.1.3
1.1.2.1
1.1.2.2
1.2.1.1.1
1.2.2.1
1.2.2.2
1.2.2.3
1.2.2.4
1.2.2.5
1.2.2.6
1.2.2.7
1.2.3
1.3
1.4

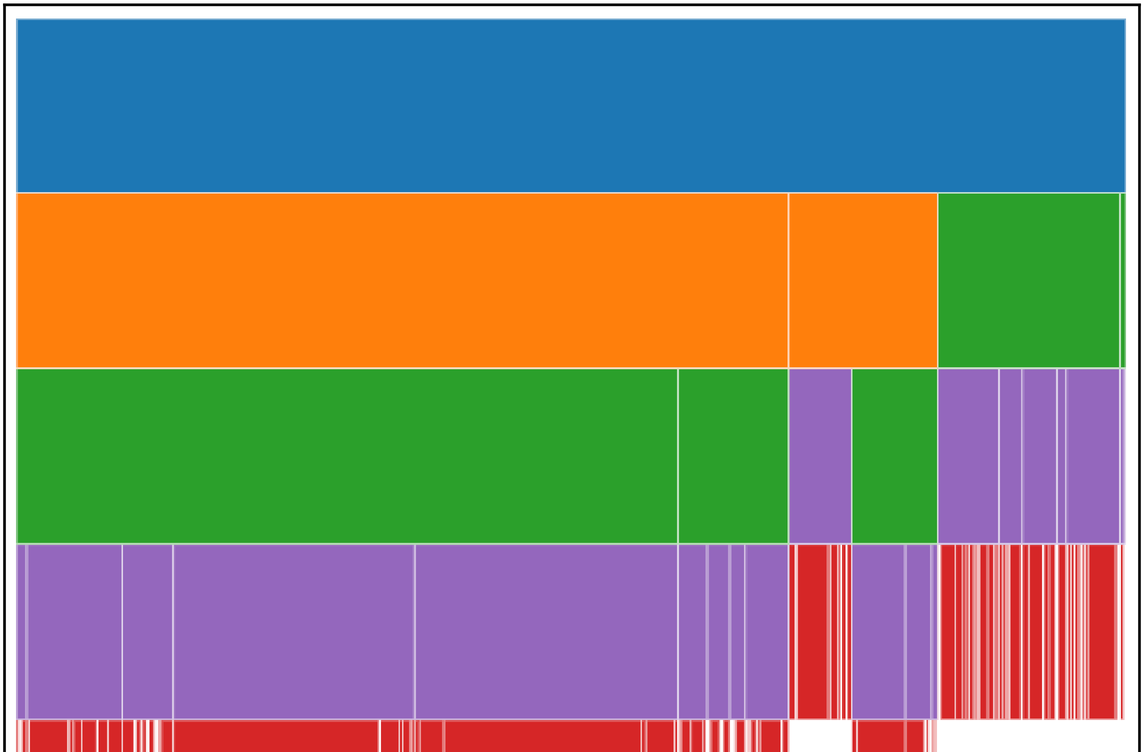


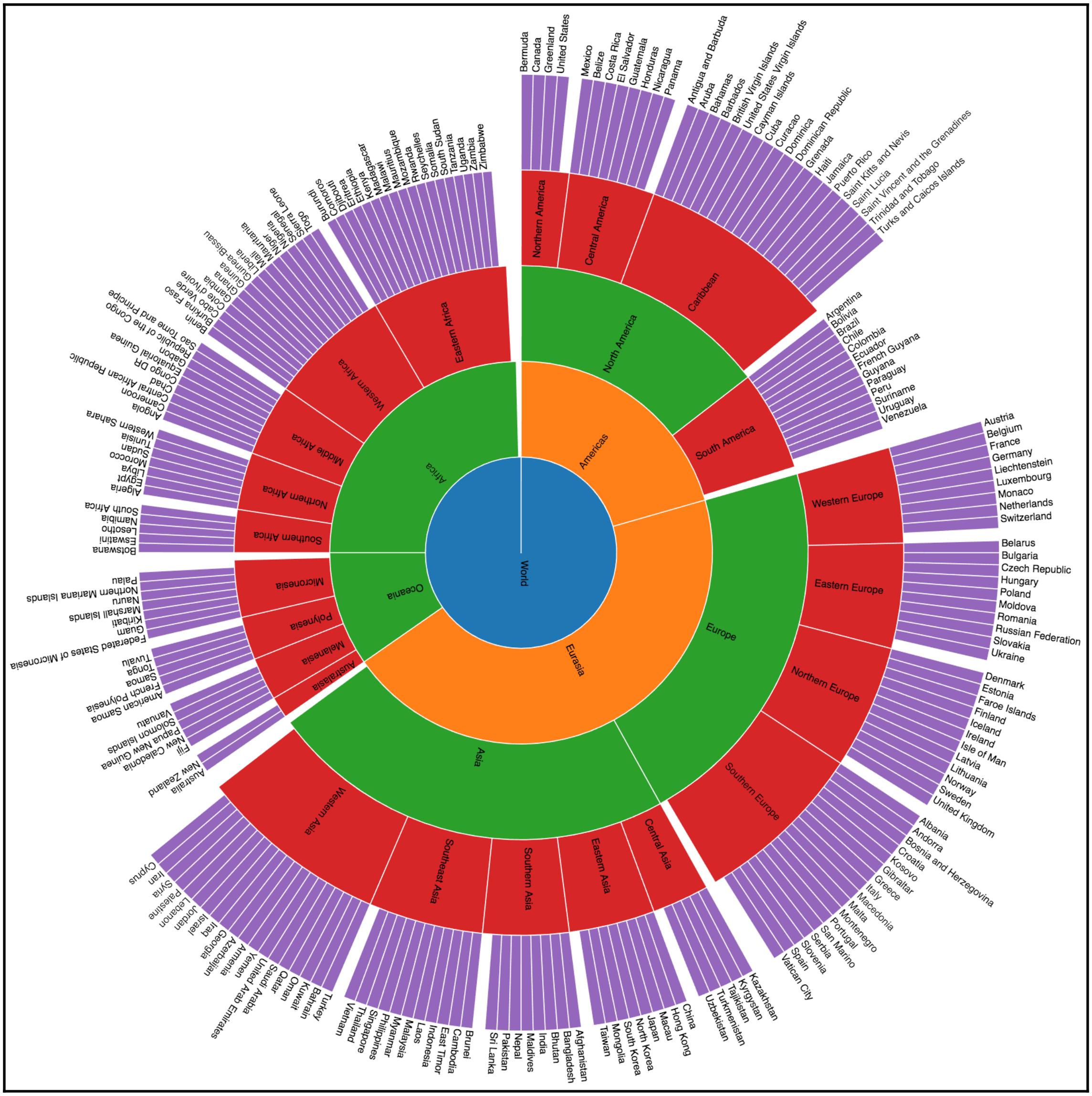
1
1.1
1.2
1.3
1.4
1.3.1
1.3.2
1.4.1
1.4.2
1.4.3
1.3.1.1
1.3.1.2
1.3.1.3
1.3.2.1
1.3.2.2
1.4.2.1
1.4.2.2
1.4.2.3
1.4.2.4
1.4.2.5
1.4.2.6
1.4.2.7
1.4.3.1
1.4.3.1.1



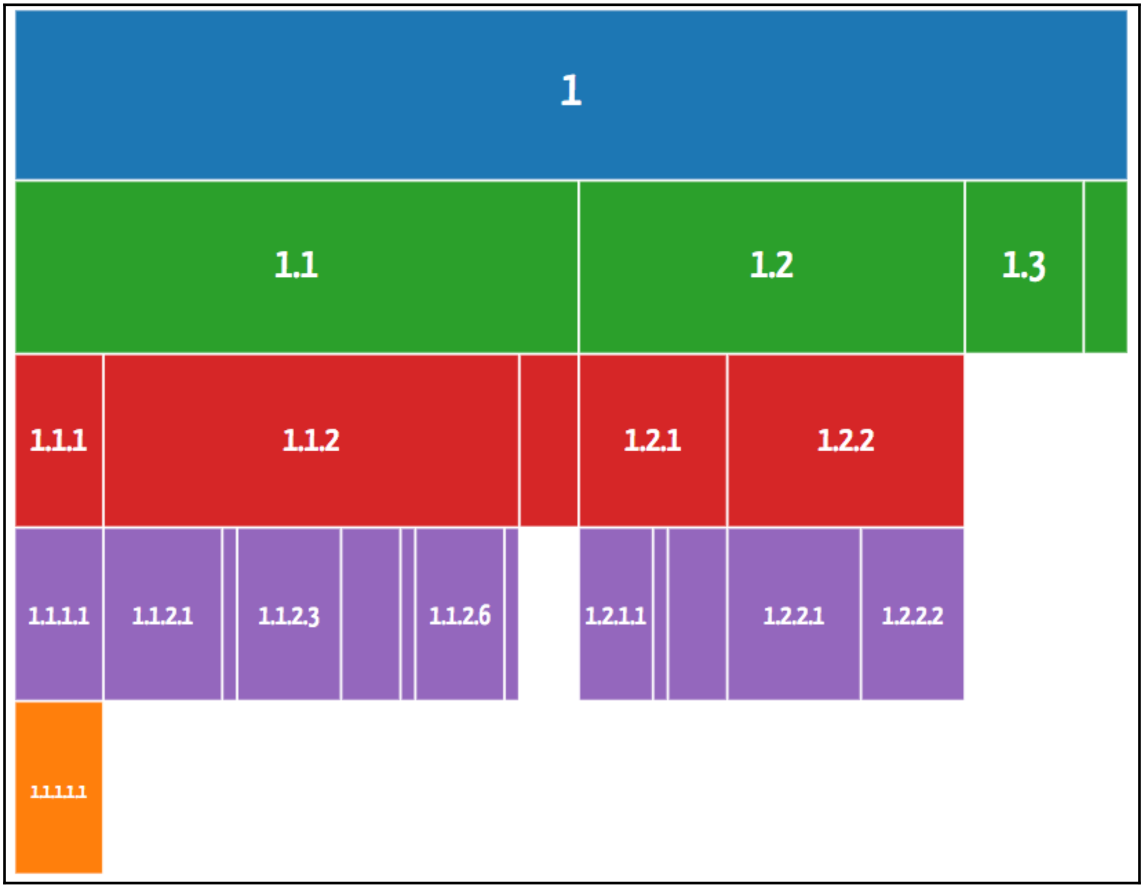
<svg>
0,0
- Math.PI / 2
x = Math.cos(angle)
<g>
0
0
x
Math.PI
0
0 = Math.PI * 2
angle = Math.atan(y/x)
radius
angle
y = Math.sin(angle)
r
y
Math.PI / 2
radius = Math.sqrt(x*x + y*y)

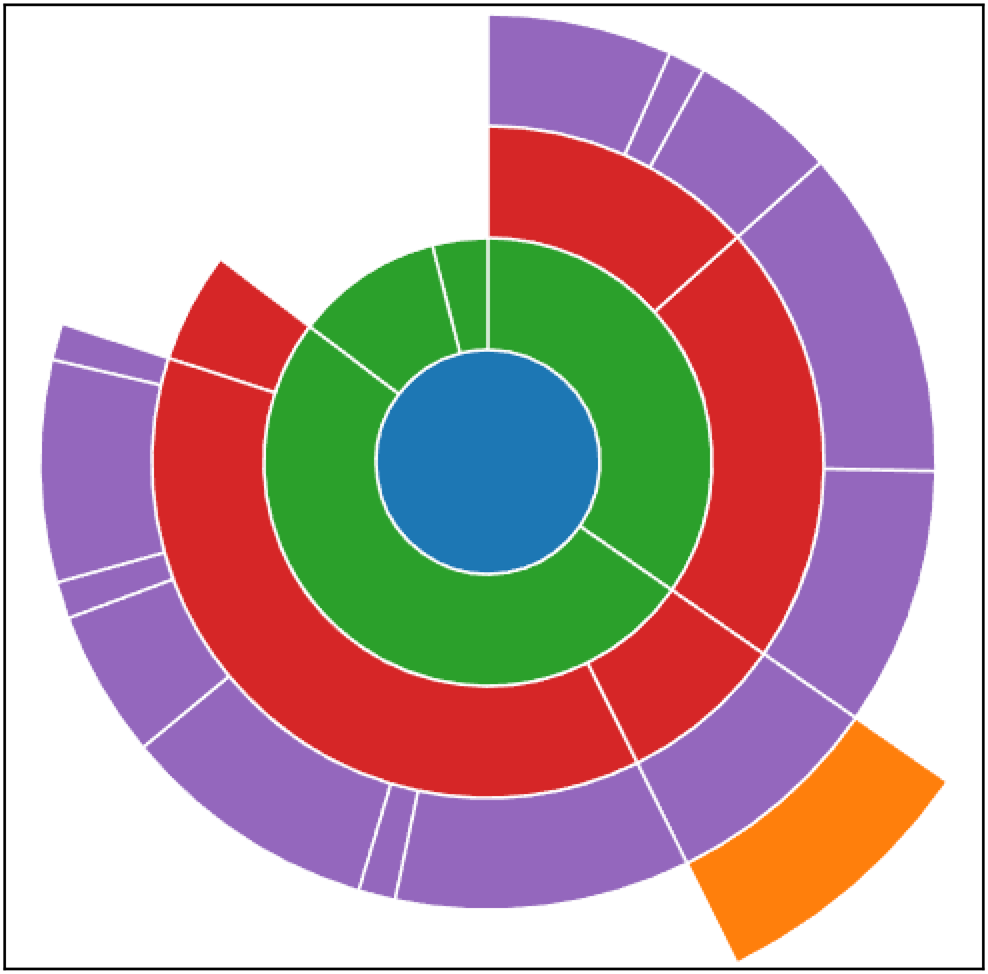


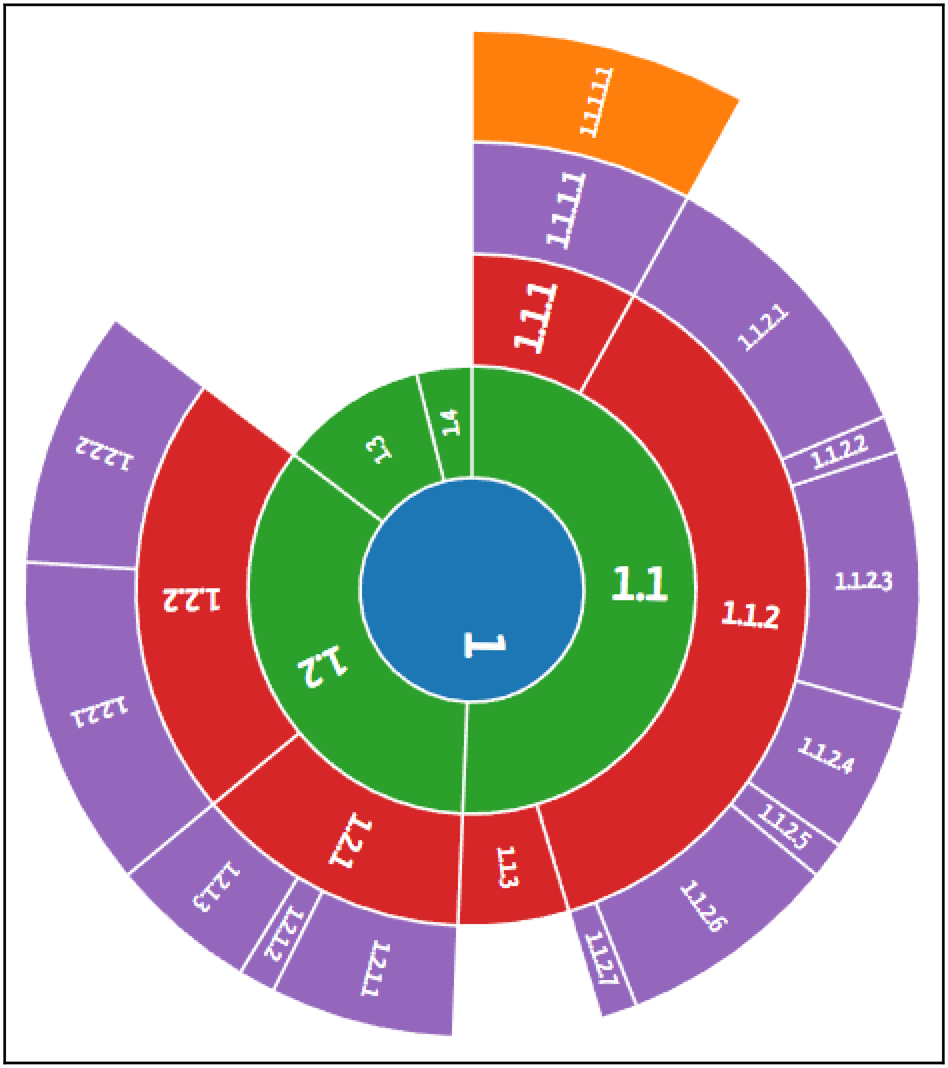


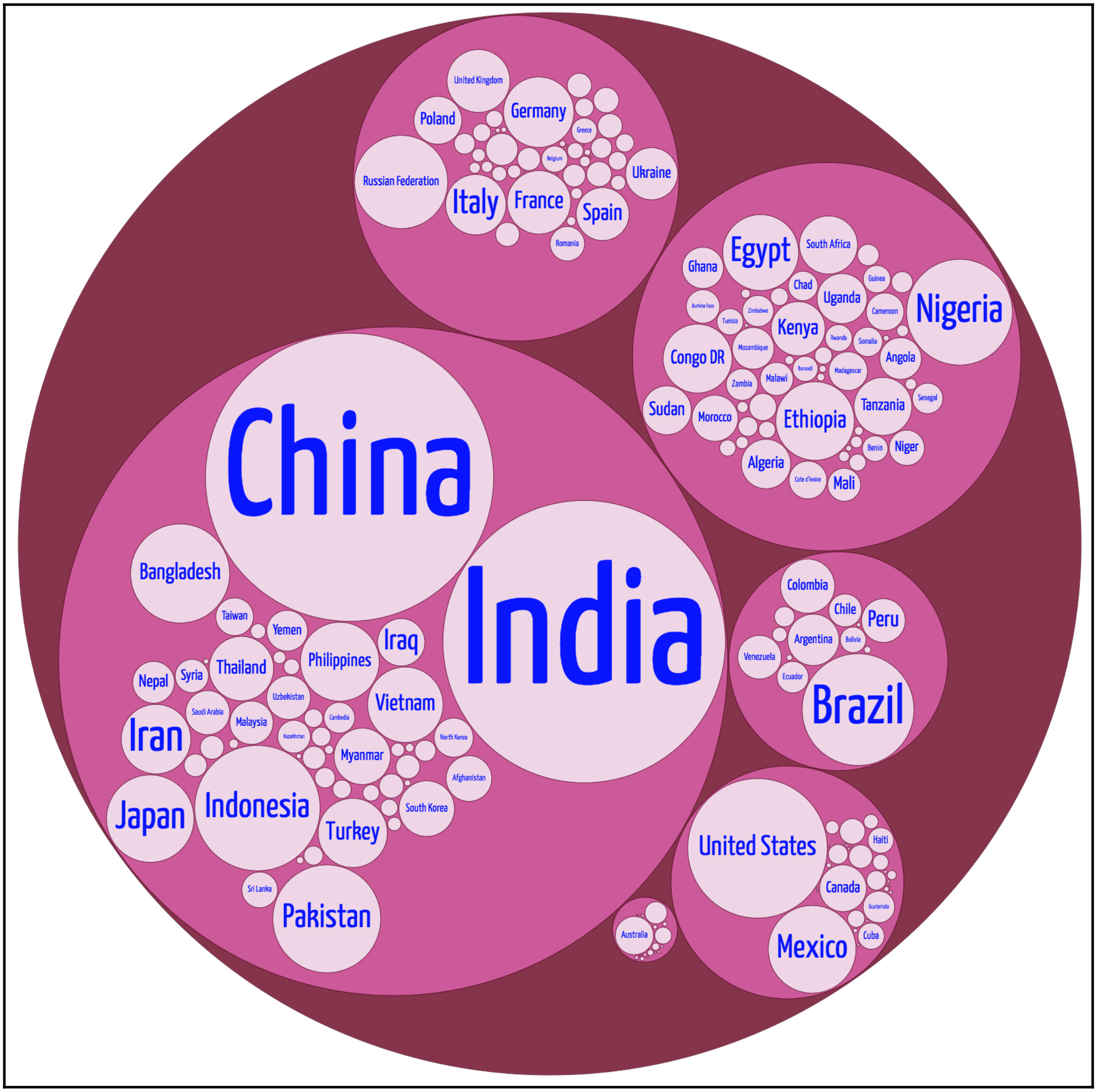

World
Americas
Eurasia
Africa
Oceania
North America
South America
Europe
Asia
Northern America
Central America
Caribbean
Western Europe
Eastern Europe
Northern Europe
Southern Europe
Central Asia
Eastern Asia
Southern Asia
Southeast Asia
Western Asia
Australasia
Melanesia
Polynesia
Micronesia
Southern Africa
Northern Africa
Middle Africa
Western Africa
Eastern Africa
Bermuda
Canada
Greenland
United States
Mexico
Belize
Costa Rica
El Salvador
Guatemala
Honduras
Nicaragua
Panama
Antigua and Barbuda
Aruba
Bahamas
Barbados
British Virgin Islands
United States Virgin Islands
Cayman Islands
Cuba
Curacao
Dominica
Dominican Republic
Grenada
Haiti
Jamaica
Puerto Rico
Saint Kitts and Nevis
Saint Lucia
Saint Vincent and the Grenadines
Trinidad and Tobago
Turks and Caicos Islands
Argentina
Bolivia
Brazil
Chile
Colombia
Ecuador
French Guyana
Guyana
Paraguay
Peru
Suriname
Uruguay
Venezuela
Austria
Belgium
France
Germany
Liechtenstein
Luxembourg
Monaco
Netherlands
Switzerland
Belarus
Bulgaria
Czech Republic
Hungary
Poland
Moldova
Romania
Russian Federation
Slovakia
Ukraine
Denmark
Estonia
Faroe Islands
Finland
Iceland
Ireland
Isle of Man
Latvia
Lithuania
Norway
Sweden
United Kingdom
Albania
Andorra
Bosnia and Herzegovina
Croatia
Kosovo
Gibraltar
Greece
Italy
Macedonia
Malta
Montenegro
Portugal
San Marino
Serbia
Slovenia
Spain
Vatican City
Kazakhstan
Kyrgystan
Tajikistan
Turkmenistan
Uzbekistan
China
Hong Kong
Macau
Japan
North Korea
South Korea
Mongolia
Taiwan
Afghanistan
Bangladesh
Bhutan
India
Maldives
Nepal
Pakistan
Sri Lanka
Brunei
Cambodia
East Timor
Indonesia
Laos
Malaysia
Myanmar
Philippines
Singapore
Thailand
Vietnam
Turkey
Bahrain
Kuwait
Oman
Qatar
Saudi Arabia
United Arab Emirates
Yemen
Armenia
Azerbaijan
Georgia
Iraq
Israel
Jordan
Lebanon
Palestine
Syria
Iran
Cyprus
Australia
New Zealand
New Caledonia
Fiji
Papua New Guinea
Solomon Islands
Vanuatu
American Samoa
French Polynesia
Samoa
Tonga
Tuvalu
Federated States of Micronesia
Guam
Kiribati
Marshall Islands
Nauru
Northern Mariana Islands
Palau
Botswana
Eswatini
Lesotho
Namibia
South Africa
Algeria
Egypt
Libya
Morocco
Sudan
Tunisia
Western Sahara
Angola
Cameroon
Central African Republic
Chad
Congo DR
Equatorial Guinea
Gabon
Republic of the Congo
Sao Tome and Principe
Benin
Burkina Faso
Cabo Verde
Cote d'Ivoire
Gambia
Ghana
Guinea
Guinea-Bissau
Liberia
Mali
Mauritania
Niger
Nigeria
Senegal
Sierra Leone
Togo
Burundi
Comoros
Djibouti
Eritrea
Ethiopia
Kenya
Madagascar
Malawi
Mauritius
Mozambique
Rwanda
Seychelles
Somalia
South Sudan
Tanzania
Uganda
Zambia
Zimbabwe

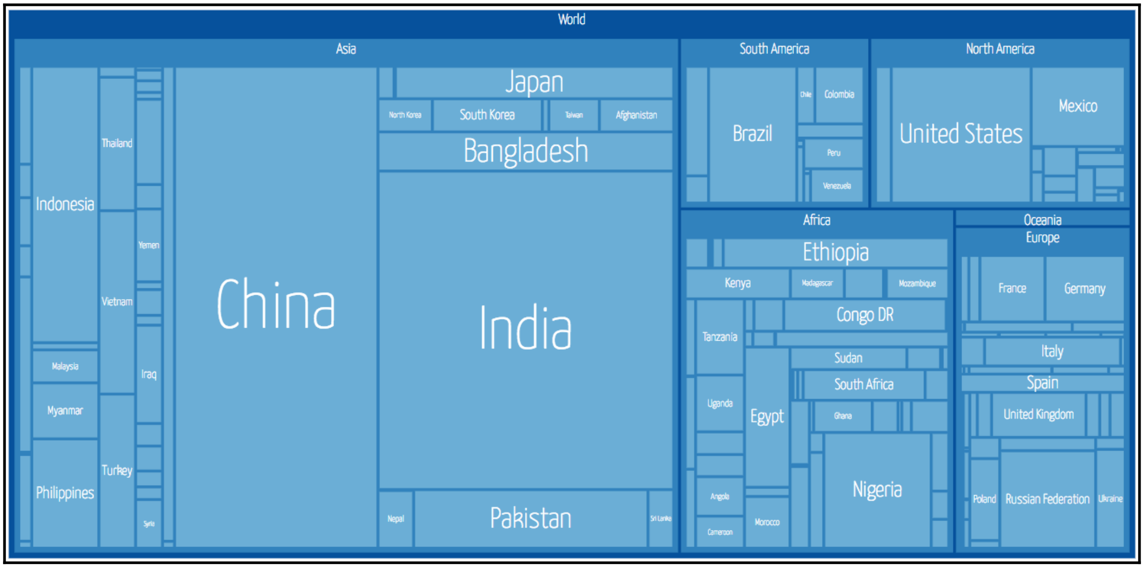
```
▼hl ℹ
  ▼children: Array(4)
    ▼0: hl
      ▶children: (3) [hl, hl, hl]
      ▶data: {id: "group_2", name: "Second Group", children: Array(3)}
       depth: 1
       height: 3
       number: "1.1"
      ▶parent: hl {data: {…}, height: 4, depth: 0, parent: null, children: Array(4), …}
       value: 38
       x0: 2
       x1: 454.9866666666666
       y0: 140
       y1: 278
      ▶__proto__: Object
    ▶1: hl {data: {…}, height: 2, depth: 1, parent: hl, children: Array(2), …}
    ▶2: hl {data: {…}, height: 0, depth: 1, parent: hl, value: 8, …}
    ▶3: hl {data: {…}, height: 0, depth: 1, parent: hl, value: 3, …}
     length: 4
    ▶__proto__: Array(0)
  ▶data: {id: "root", name: "Root Level", children: Array(4)}
   depth: 0
   height: 4
   number: 1
   parent: null
   value: 75
   x0: 2
   x1: 898
   y0: 2
   y1: 138
```

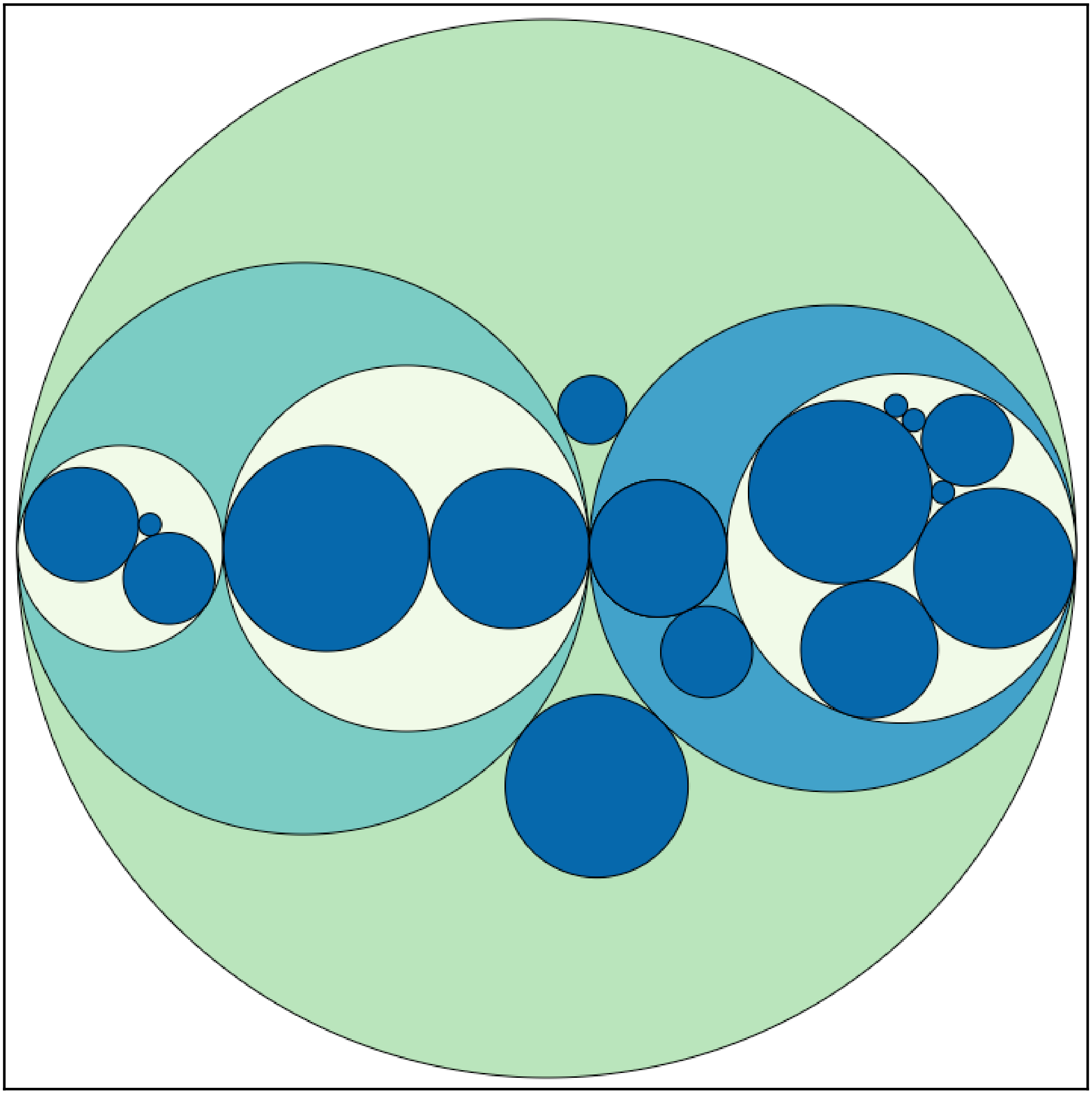

1
1.1
1.2
1.3
1.1.1
1.1.2
1.2.1
1.2.2
1.1.1.1
1.1.2.1
1.1.2.3
1.1.2.6
1.2.1.1
1.2.2.1
1.2.2.2
1.1.1.1.1

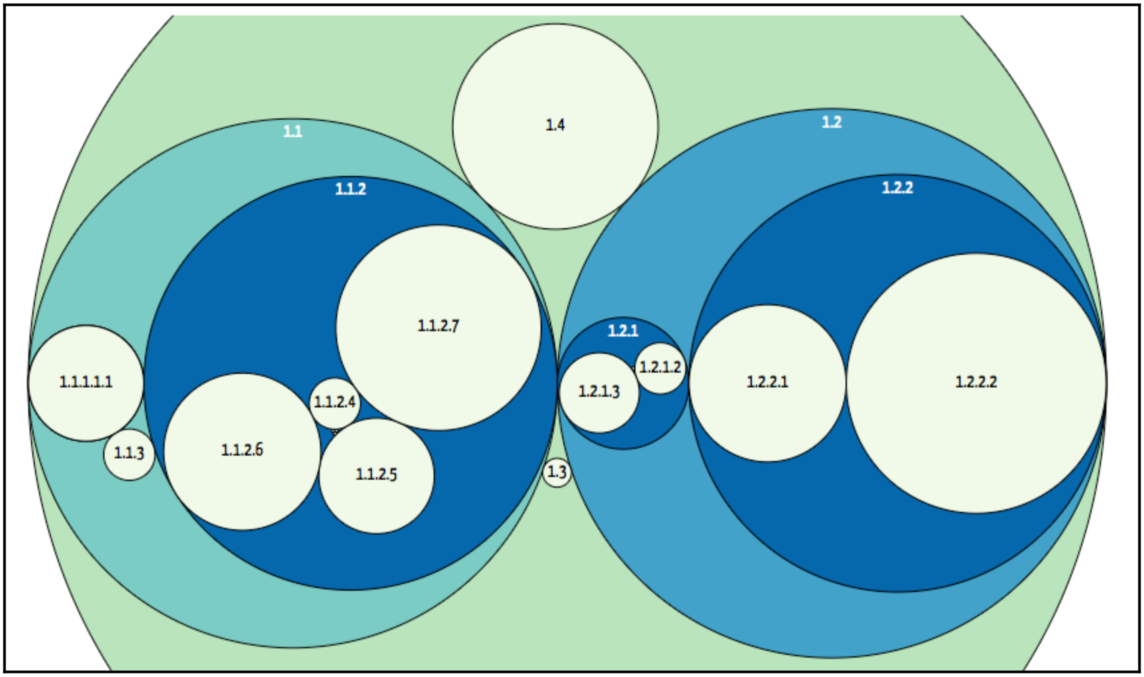


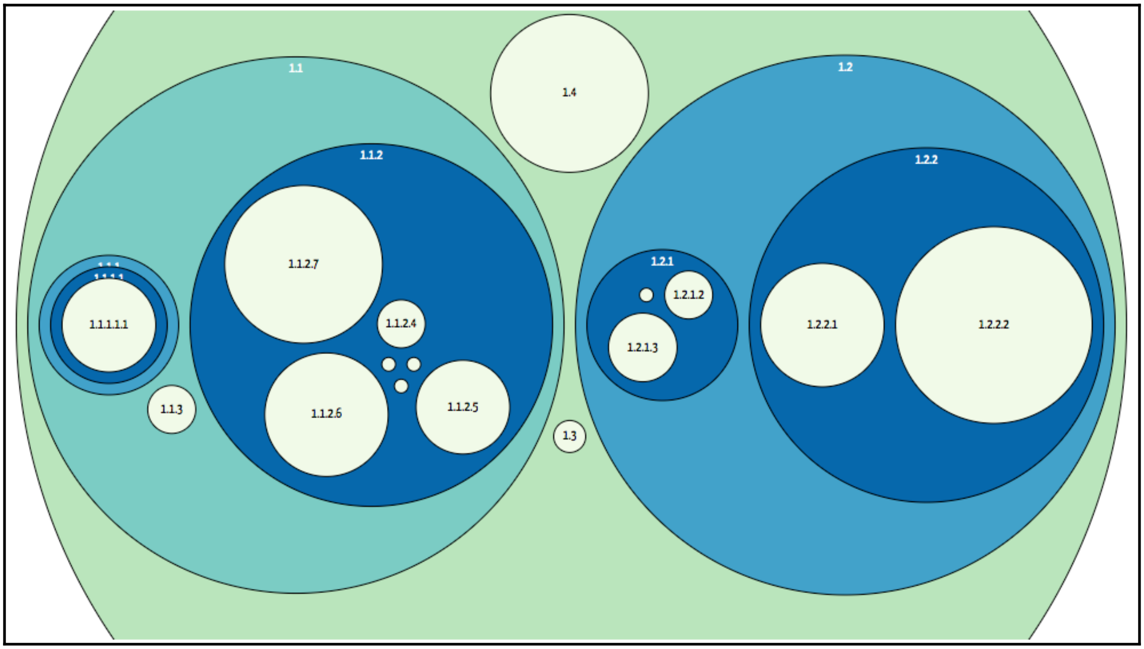

1
1.1
1.2
1.3
1.4
1.1.1
1.1.2
1.1.3
1.2.1
1.2.2
1.1.1.1
1.1.1.1.1
1.1.2.1
1.1.2.2
1.1.2.3
1.1.2.4
1.1.2.5
1.1.2.6
1.1.2.7
1.2.1.1
1.2.1.2
1.2.1.3
1.2.2.1
1.2.2.2

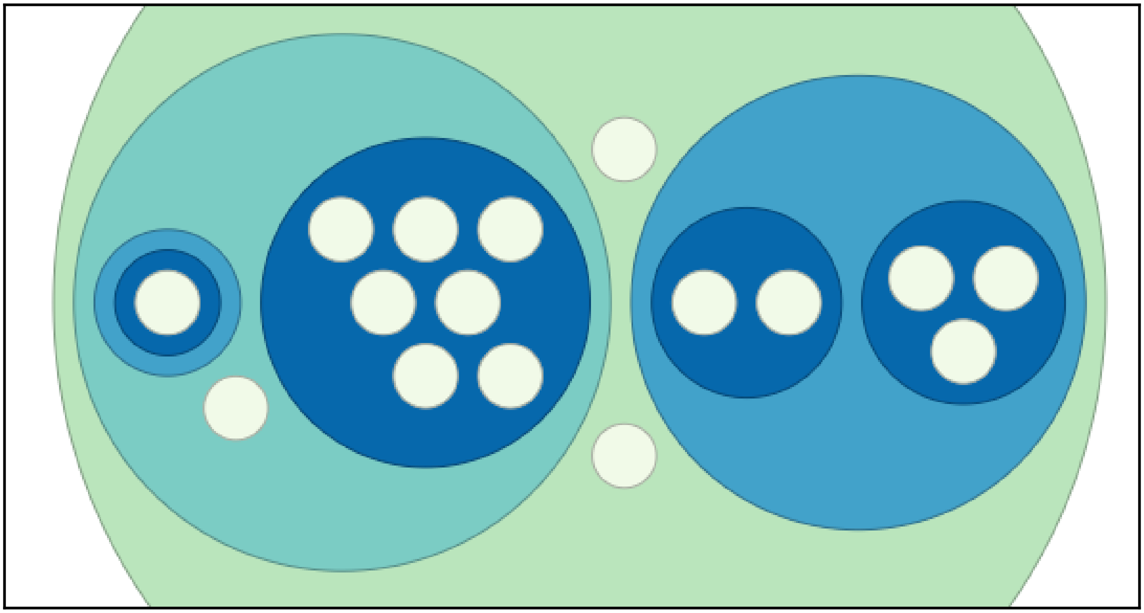

United Kingdom
Germany
Poland
Greece
Belgium
Russian Federation
Ukraine
Italy
France
Spain
Romania
Egypt
South Africa
Ghana
Chad
Guinea
Uganda
Nigeria
Burkina Faso
Zimbabwe
Tunisia
Kenya
Cameroon
Rwanda
Somalia
Congo DR
Mozambique
Angola
Burundi
Madagascar
Zambia
Malawi
Senegal
Sudan
Morocco
Ethiopia
Tanzania
Benin
Niger
Algeria
Cote d'Ivoire
Mali
China
India
Bangladesh
Taiwan
Yemen
Iraq
Philippines
Thailand
Nepal
Syria
Vietnam
Uzbekistan
Saudi Arabia
Cambodia
Iran
Malaysia
Kazakhstan
North Korea
Myanmar
Afghanistan
Indonesia
Japan
South Korea
Turkey
Sri Lanka
Pakistan
Colombia
Chile
Peru
Argentina
Bolivia
Venezuela
Ecuador
Brazil
United States
Haiti
Canada
Guatemala
Australia
Cuba
Mexico

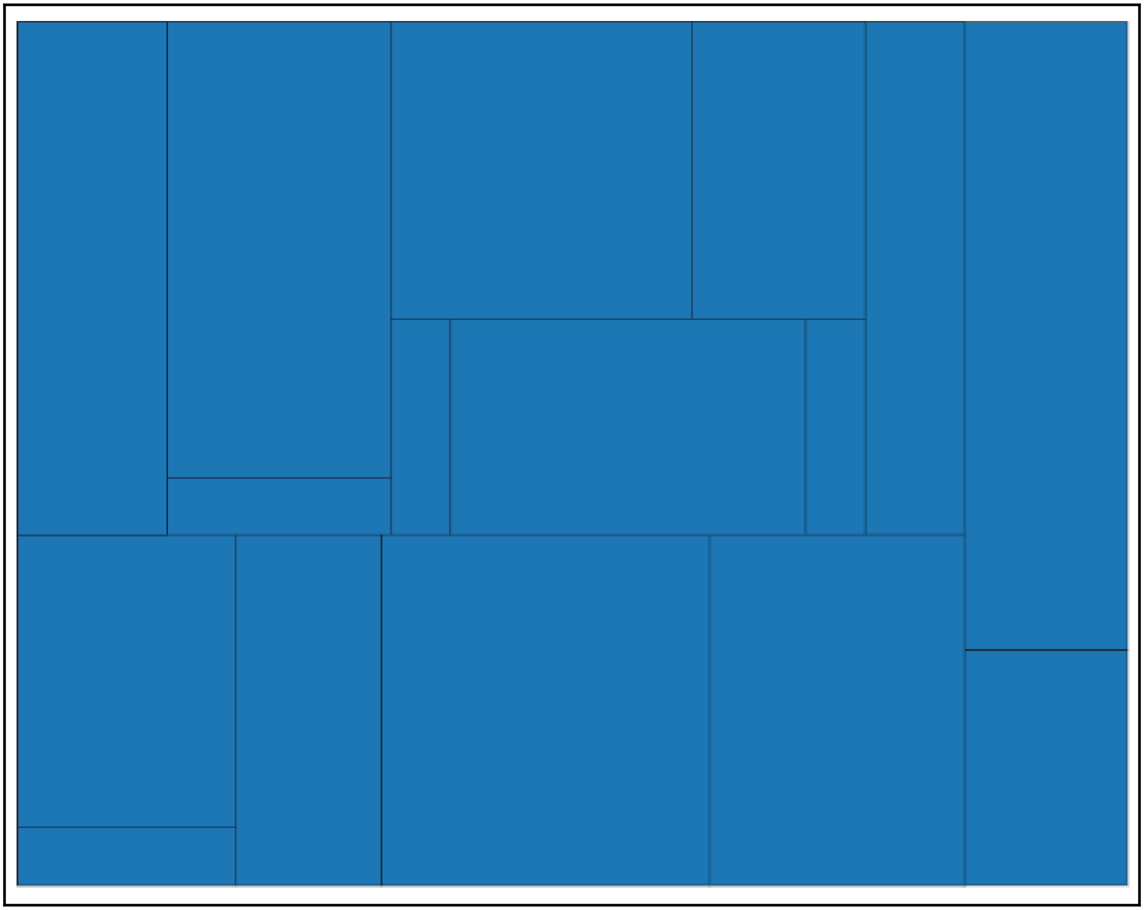

World
Asia
South America
North America
Japan
North Korea
South Korea
Taiwan
Afghanistan
Bangladesh
Thailand
Indonesia
Yemen
China
India
Vietnam
Iraq
Malaysia
Myanmar
Turkey
Philippines
Syria
Nepal
Pakistan
Sri Lanka
Brazil
Chile
Colombia
Peru
Venezuela
United States
Mexico
Africa
Oceania
Europe
Ethiopia
Kenya
Madagascar
Mozambique
Congo DR
Tanzania
Sudan
South Africa
Uganda
Egypt
Ghana
Nigeria
Angola
Cameroon
Morocco
France
Germany
Italy
Spain
United Kingdom
Poland
Russian Federation
Ukraine

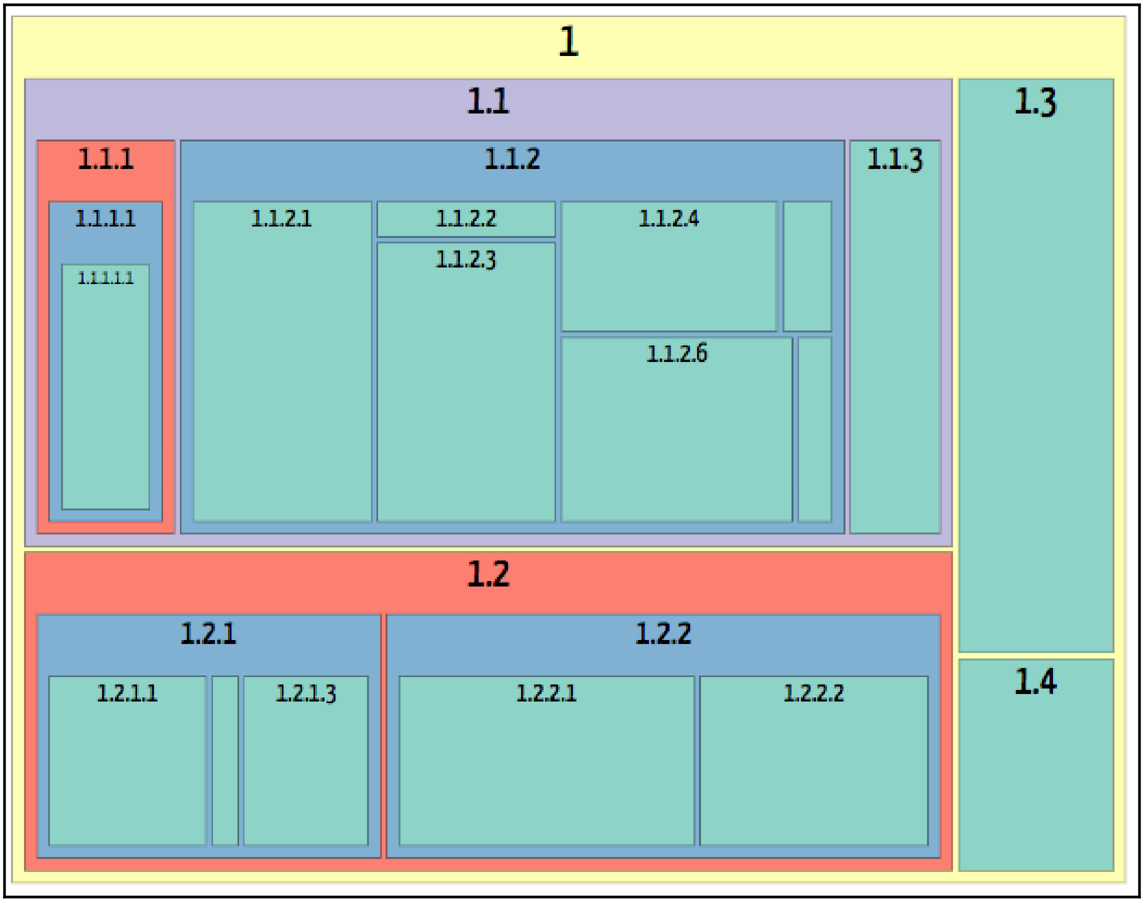


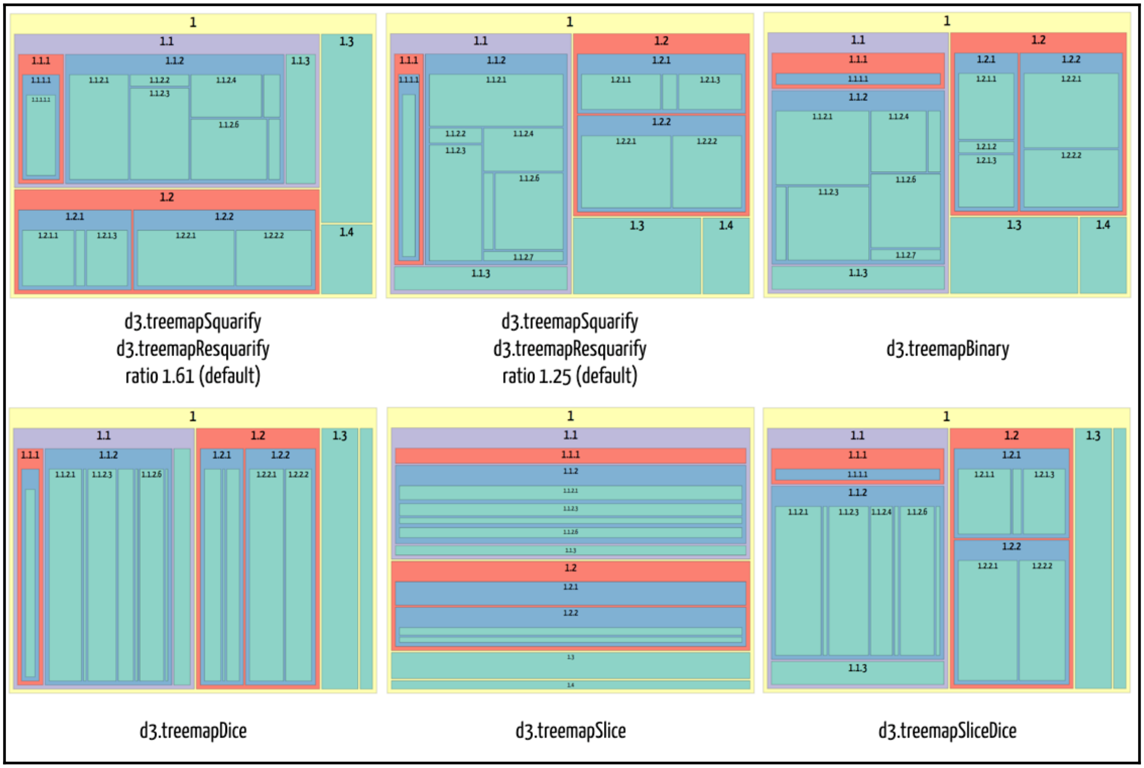

1.1
1.4
1.2
1.1.2
1.2.2
1.1.2.7
1.2.1
1.1.1.1.1
1.2.1.2
1.2.1.3
1.2.2.1
1.2.2.2
1.1.2.4
1.1.3
1.1.2.6
1.1.2.5
1.3


1.1
1.4
1.2
1.1.2
1.2.2
1.1.2.7
1.2.1
1.1.1.1.1
1.1.2.4
1.2.1.2
1.2.2.1
1.2.2.2
1.2.1.3
1.1.3
1.1.2.6
1.1.2.5
1.3






1
1.1
1.1.1
1.1.1.1
1.1.1.1.1
1.1.2
1.1.2.1
1.1.2.2
1.1.2.3
1.1.2.4
1.1.2.6
1.1.3
1.2
1.2.1
1.2.1.1
1.2.1.3
1.2.2
1.2.2.1
1.2.2.2
1.3
1.4



d3.treemapSquarify
d3.treemapResquarify
ratio 1.61 (default)

d3.treemapSquarify
d3.treemapResquarify
ratio 1.25 (default)

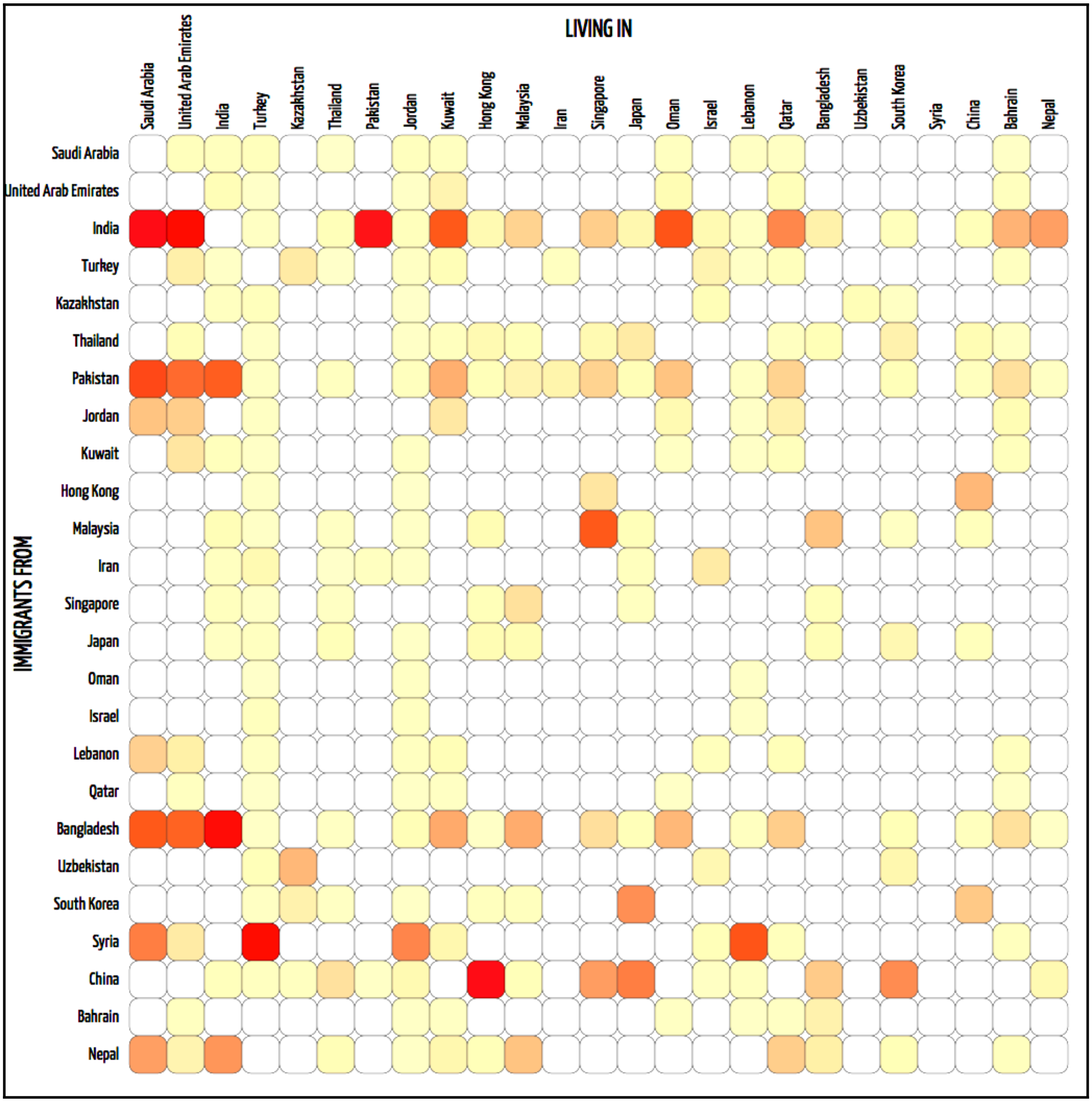
d3.treemapBinary

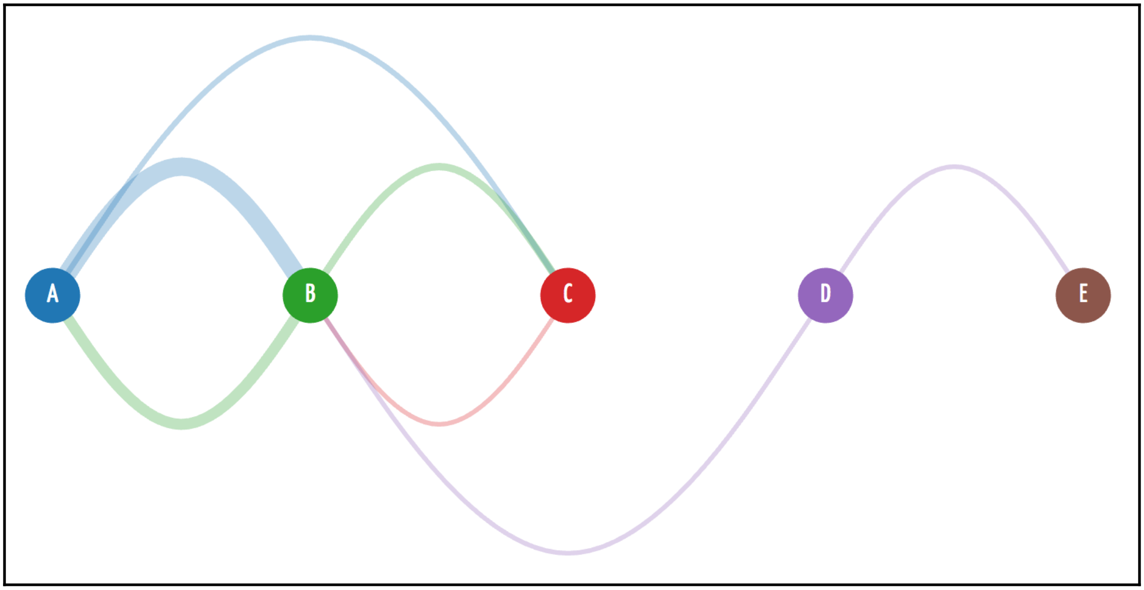
d3.treemapDice

d3.treemapSlice

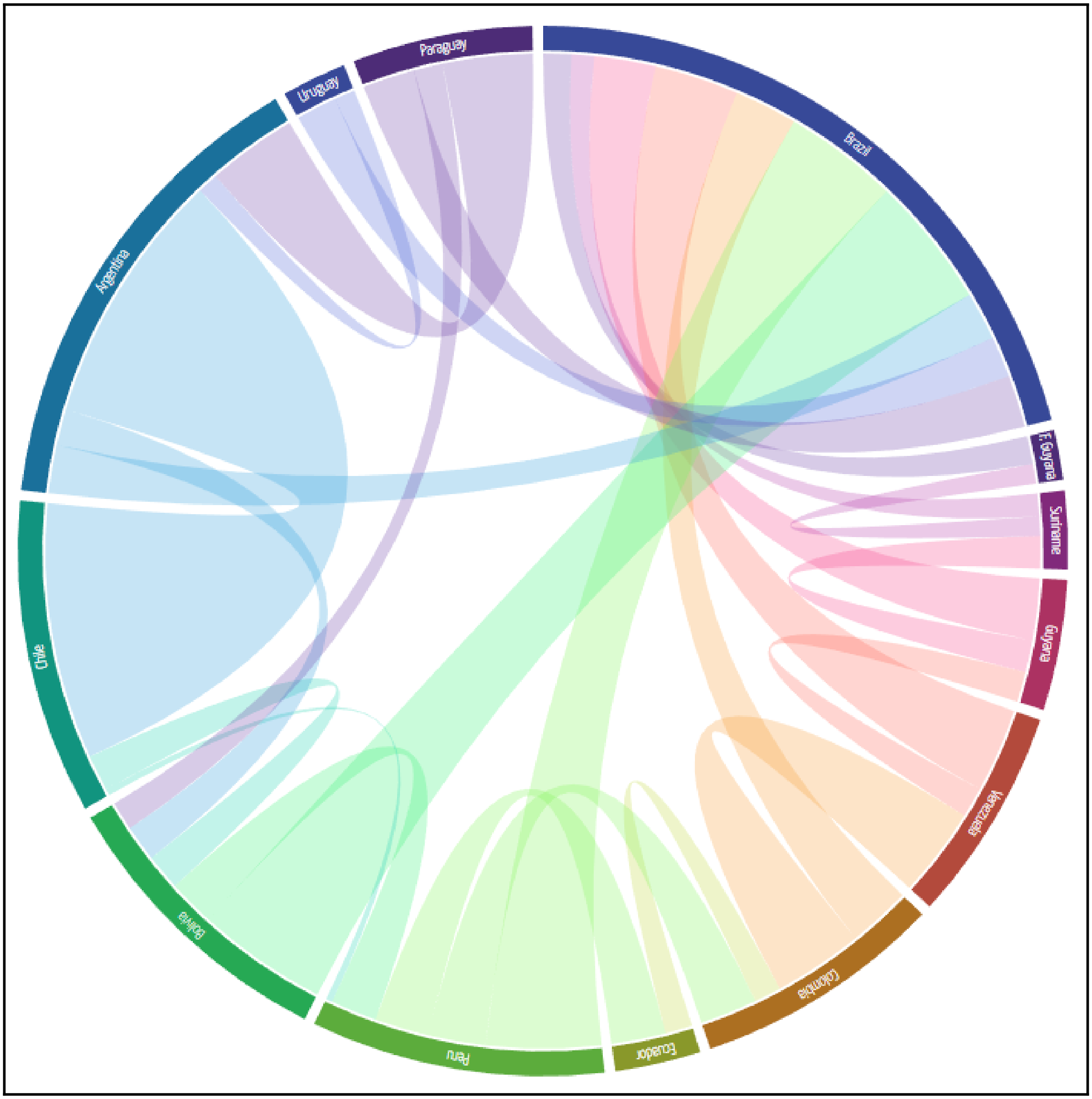
d3.treemapSliceDice

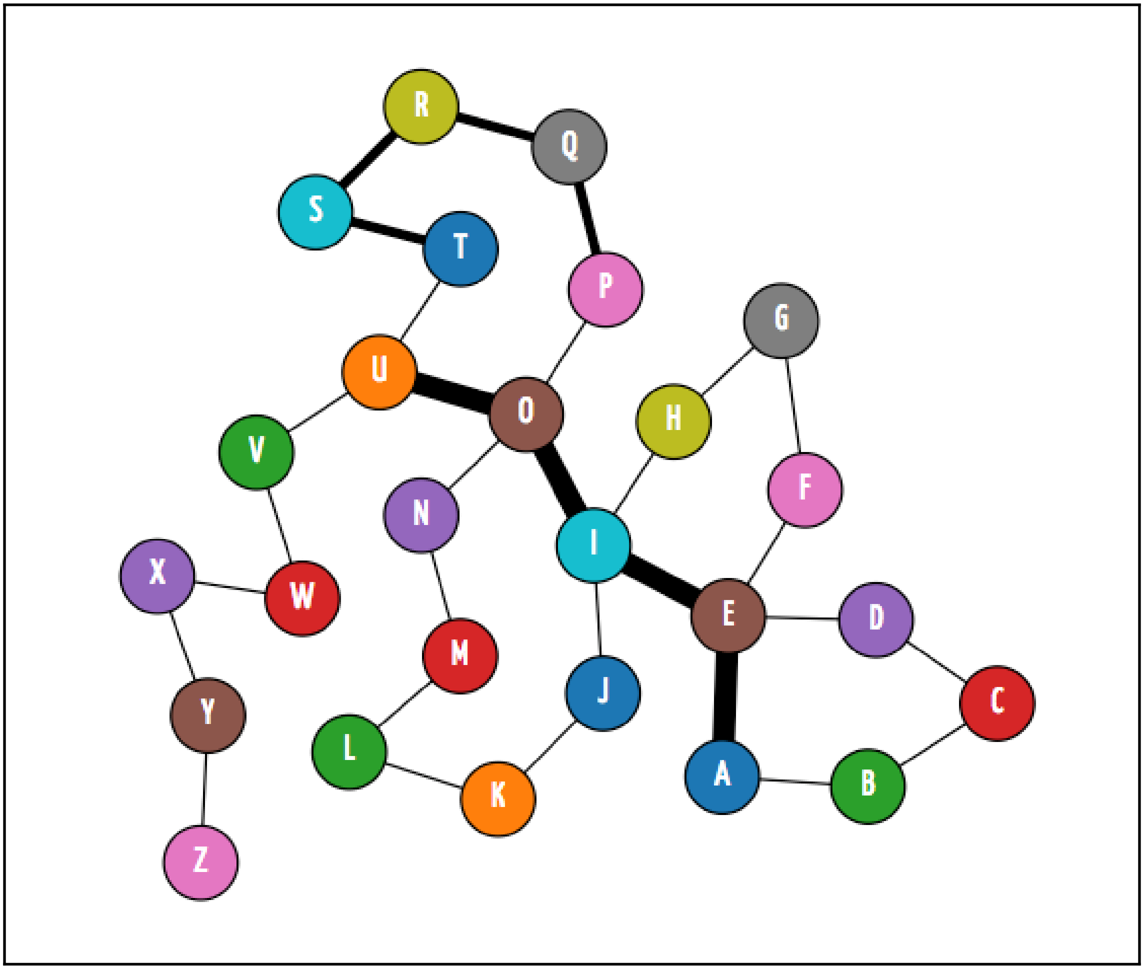
Chapter 10: Visualizing Flows and Networks

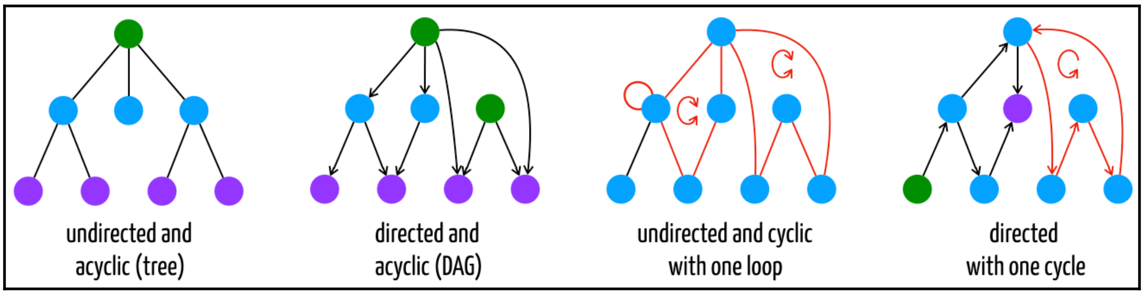


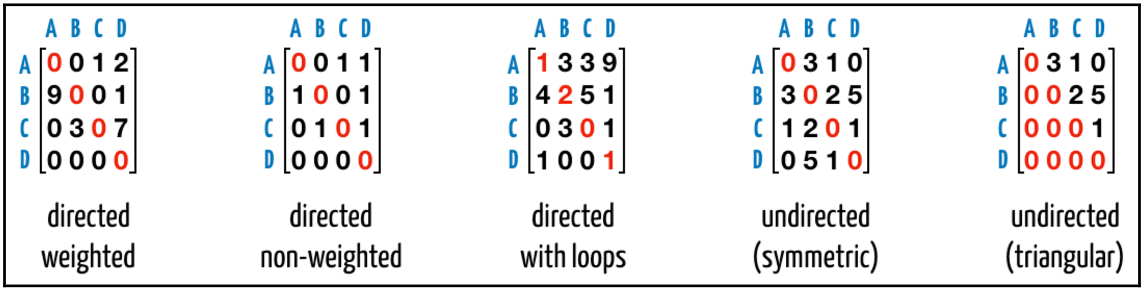


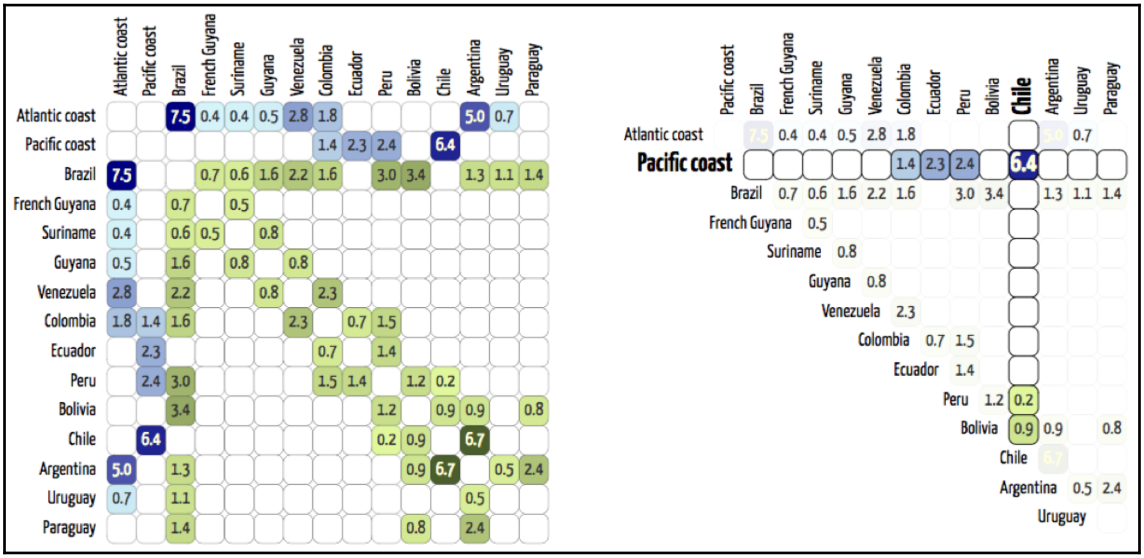
A
B
C
D
E

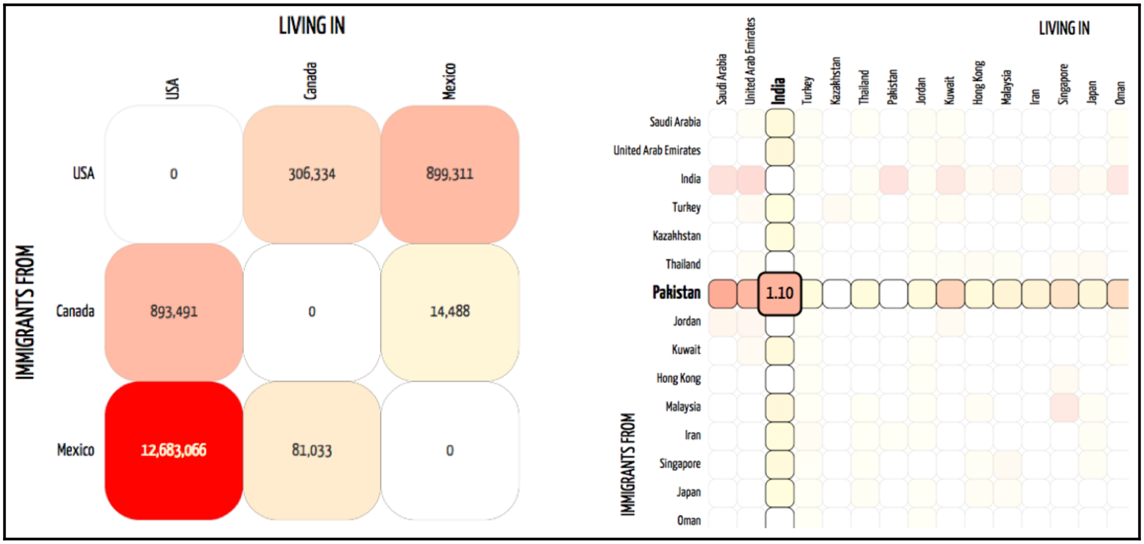

Brazil
F. Guyana
Suriname
Guyana
Venezuela
Colombia
Ecuador
Peru
Bolivia
Chile
Argentina
Uruguay
Paraguay


R
Q
S
T
P
G
U
O
H
V
N
F
I
X
W
E
D
M
J
C
Y
L
A
B
K
Z

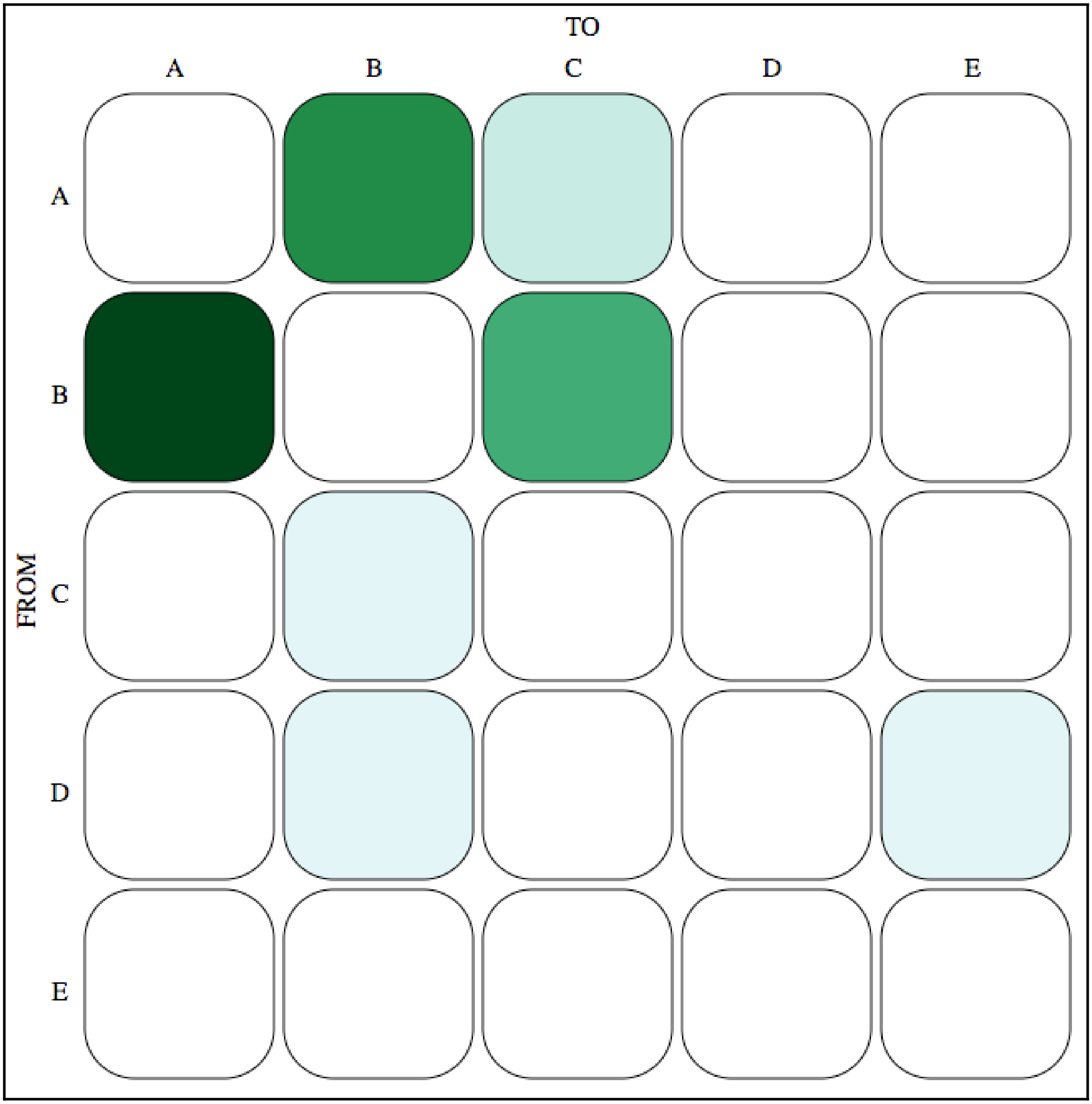

undirected and
acyclic (tree)
directed and
acyclic (DAG)
undirected and cyclic
with one loop
directed
with one cycle

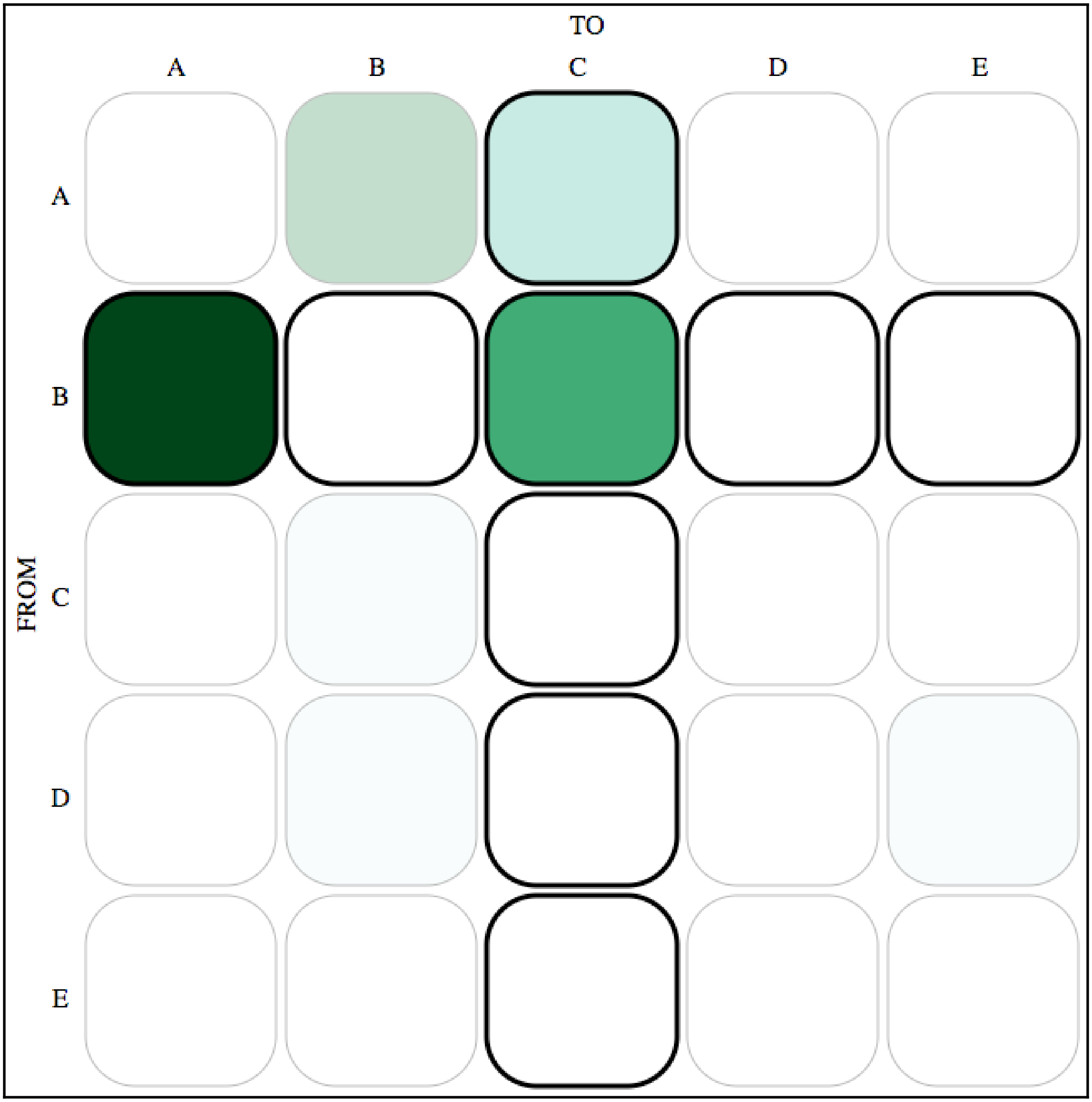

A B C D
A [0 0 1 2]
B [9 0 0 1]
C [0 3 0 7]
D [0 0 0 0]
directed weighted
A B C D
A [0 0 1 1]
B [1 0 0 1]
C [0 1 0 1]
D [0 0 0 0]
directed non-weighted
A B C D
A [1 3 3 9]
B [4 2 5 1]
C [0 3 0 1]
D [1 0 0 1]
directed with loops
A B C D
A [0 3 1 0]
B [3 0 2 5]
C [1 2 0 1]
D [0 5 1 0]
undirected (symmetric)
A B C D
A [0 3 1 0]
B [0 0 2 5]
C [0 0 0 1]
D [0 0 0 0]
undirected (triangular)

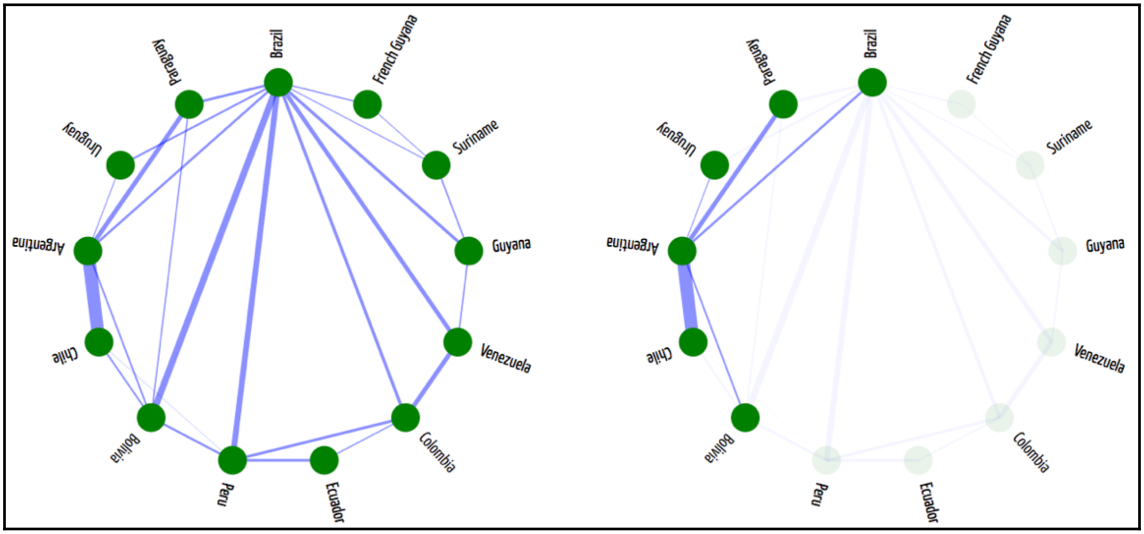

Atlantic coast
Pacific coast
Brazil
French Guyana
Suriname
Guyana
Venezuela
Colombia
Ecuador
Peru
Bolivia
Chile
Argentina
Uruguay
Paraguay

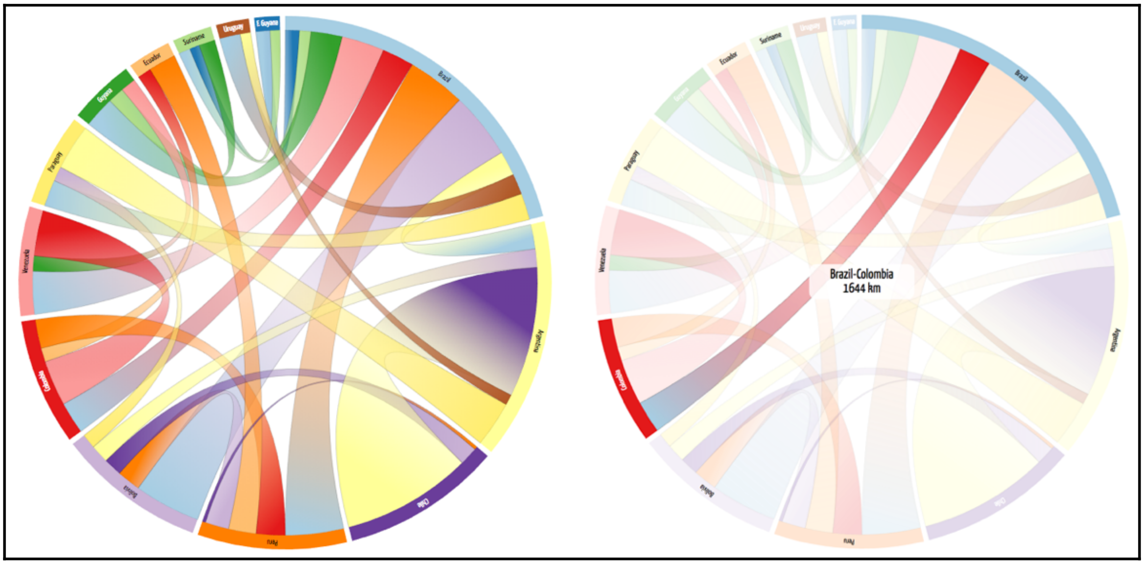


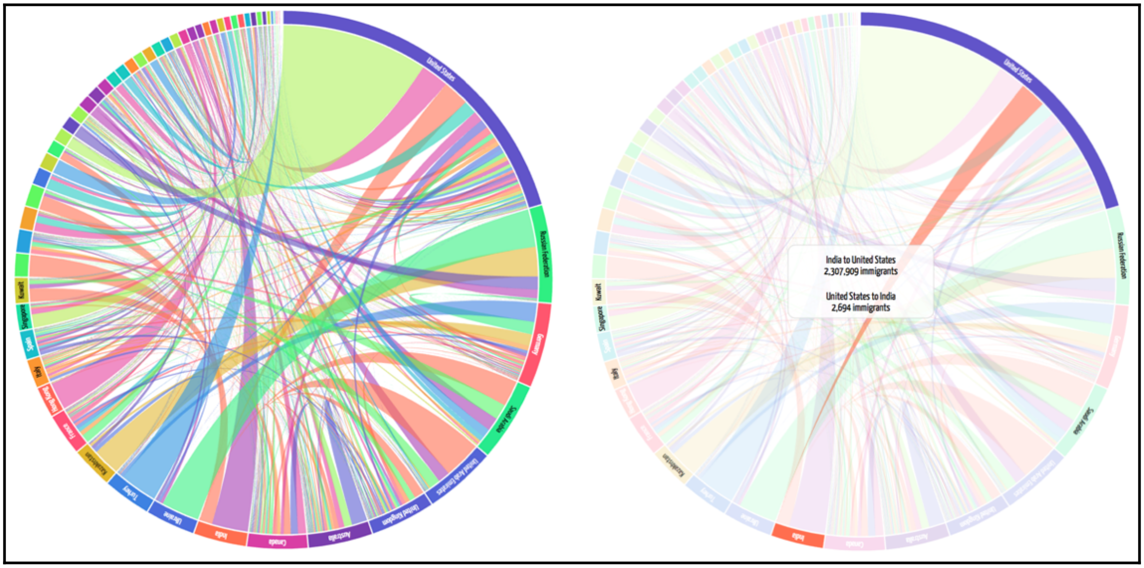
LIVING IN
USA
Canada
Mexico
IMMIGRANTS FROM
USA
0
306,334
899,311
Canada
893,491
0
14,488
Mexico
12,683,066
81,033
0
LIVING IN
Saudi Arabia
United Arab Emirates
India
Turkey
Kazakhstan
Thailand
Pakistan
Jordan
Kuwait
Hong Kong
Malaysia
Iran
Singapore
Japan
Oman
IMMIGRANTS FROM
Saudi Arabia
United Arab Emirates
India
Turkey
Kazakhstan
Thailand
Pakistan
Jordan
Kuwait
Hong Kong
Malaysia
Iran
Singapore
Japan
Oman
1.10

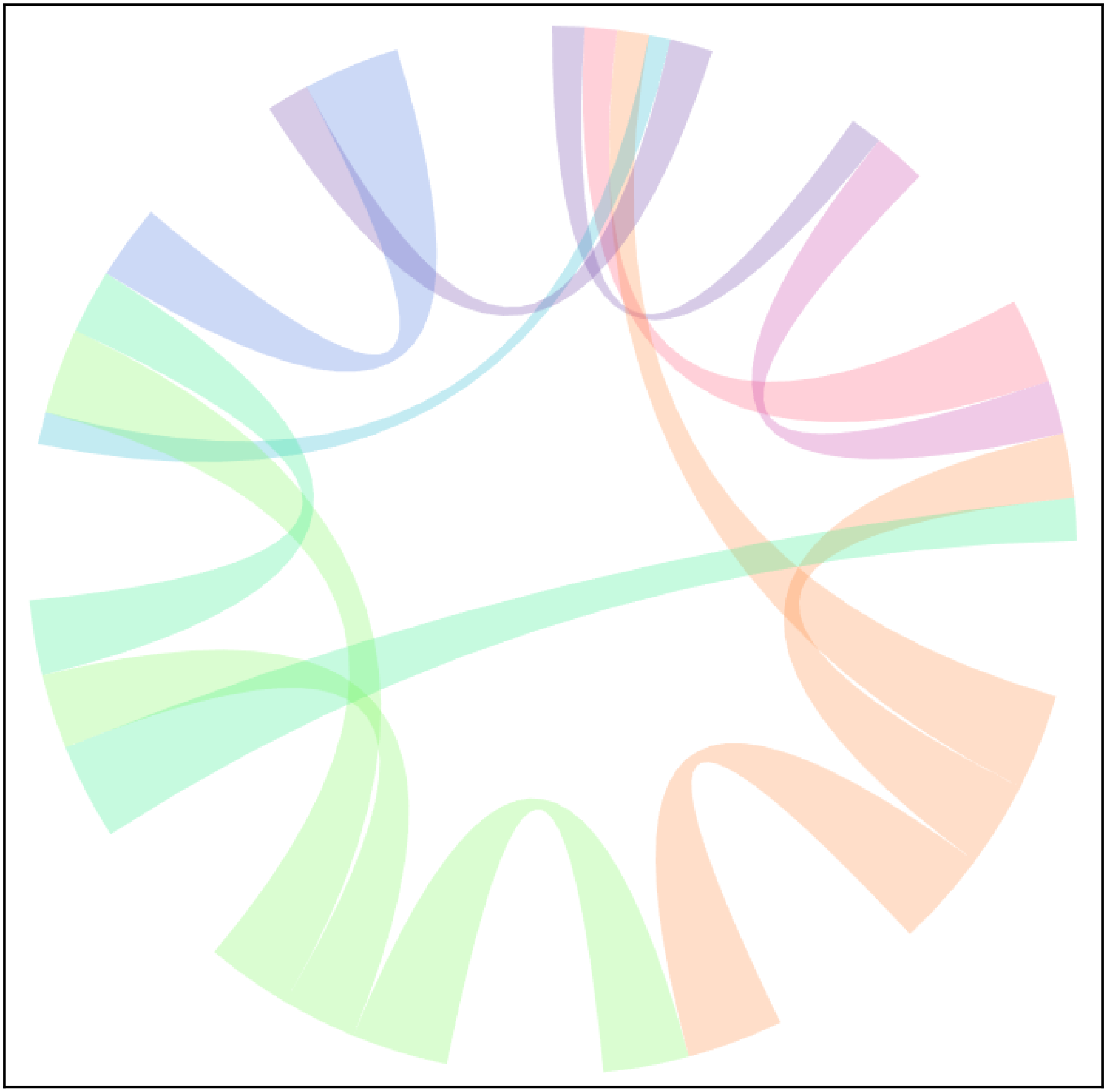

TO
A
B
C
D
E
FROM
A
B
C
D
E

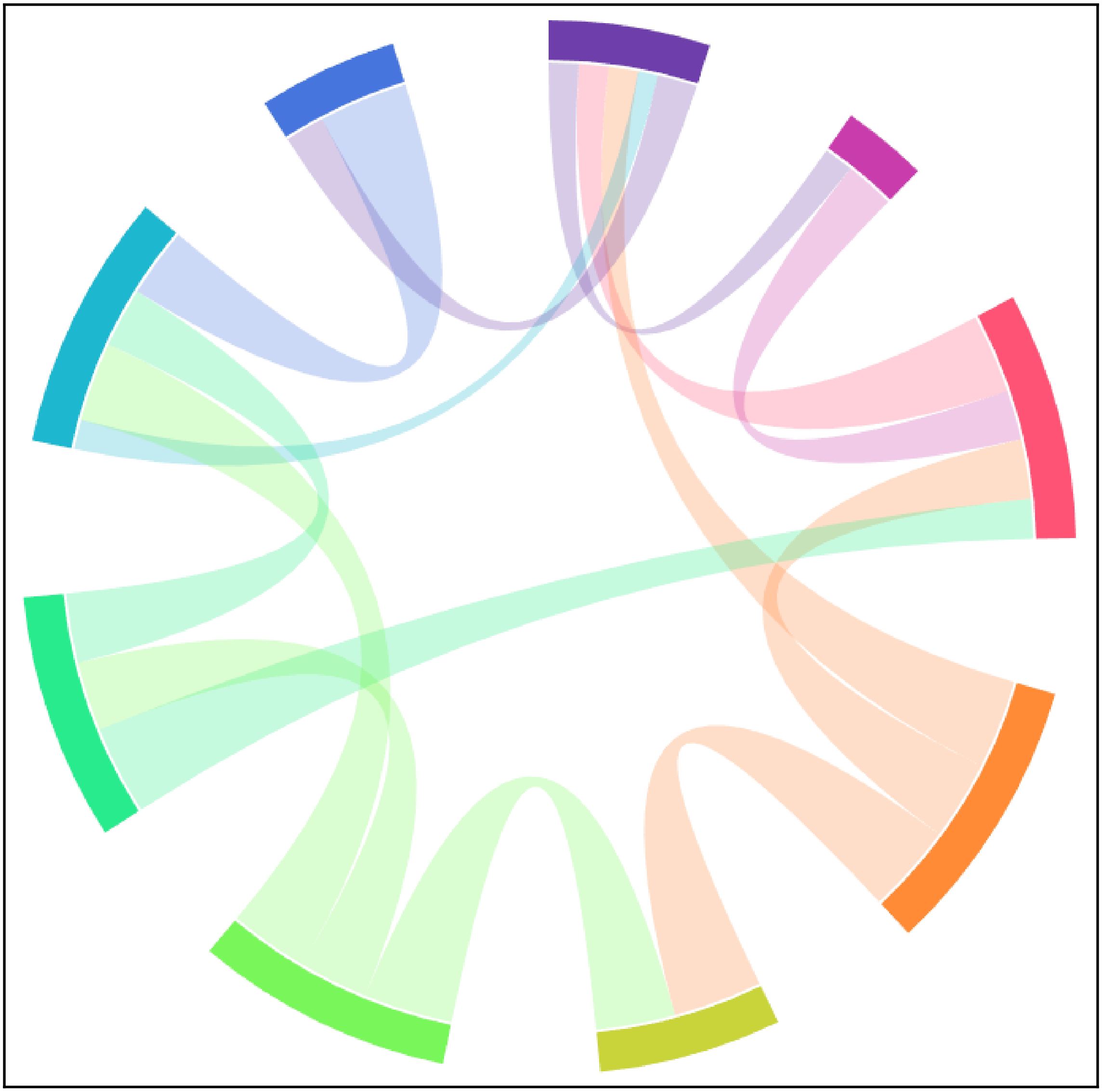

TO
A
B
C
D
E
FROM
A
B
C
D
E

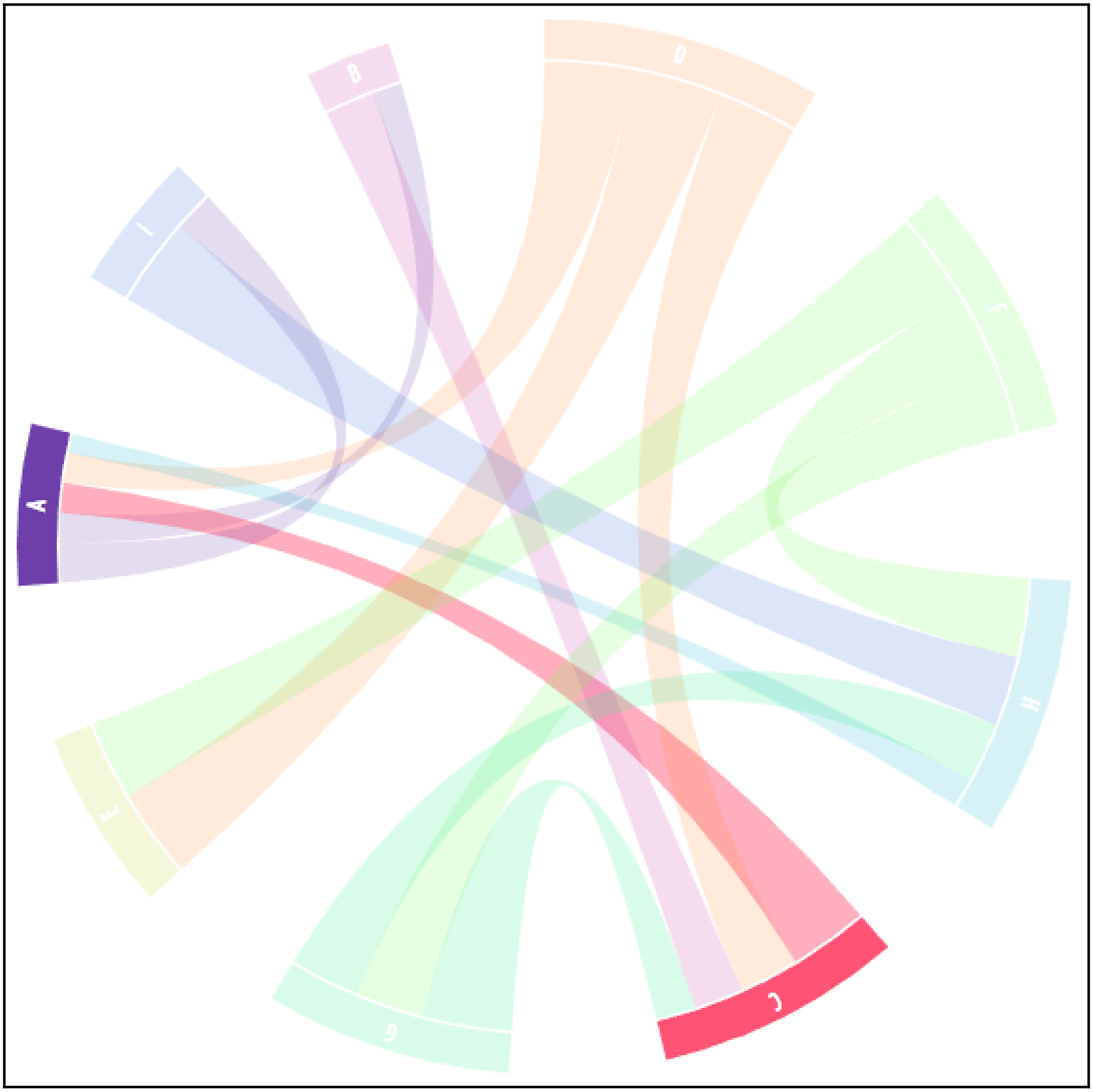

Brazil
French Guyana
Suriname
Guyana
Venezuela
Colombia
Ecuador
Peru
Bolivia
Chile
Argentina
Uruguay
Paraguay

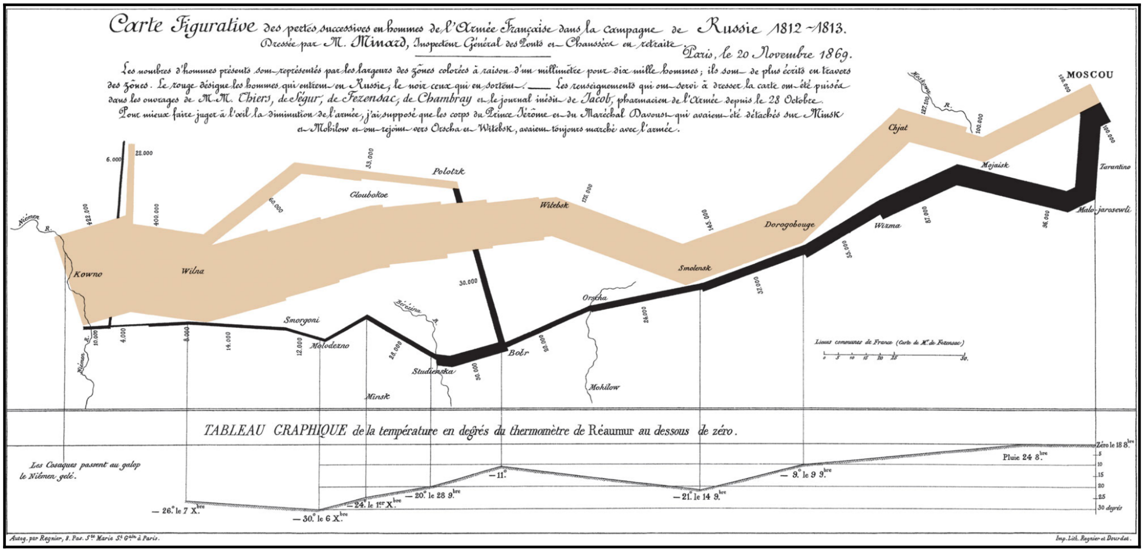

Brazil-Colombia
1644 km

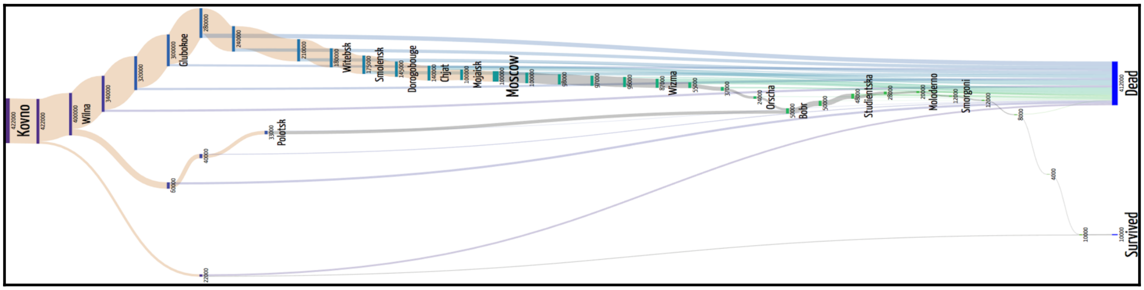

United States
Russian Federation
Germany
Saudi Arabia
United Arab Emirates
United Kingdom
Australia
Canada
India
Ukraine
Turkey
Kazakhstan
France
Italy
Spain
Singapore
Kuwait
India to United States
2,307,909 immigrants
United States to India
2,694 immigrants

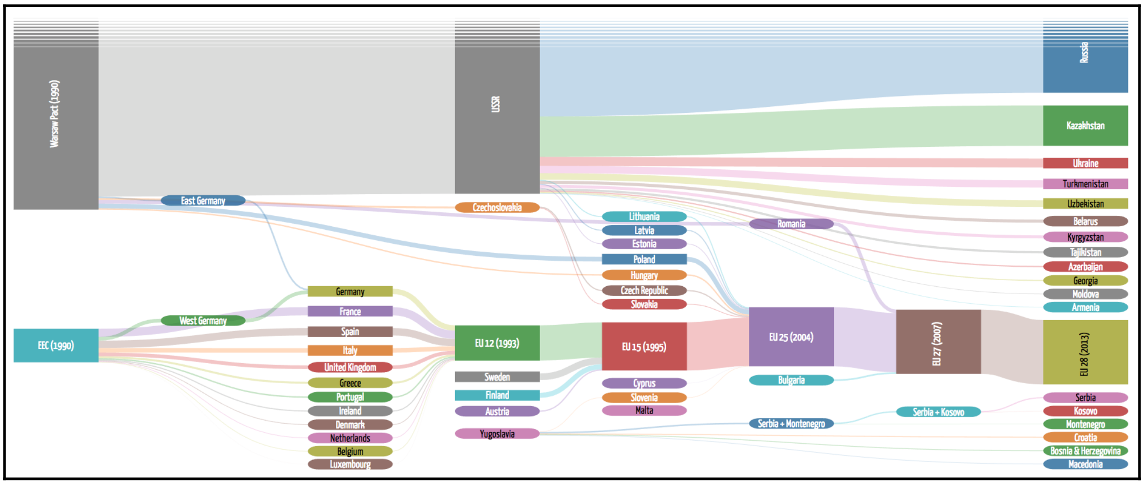


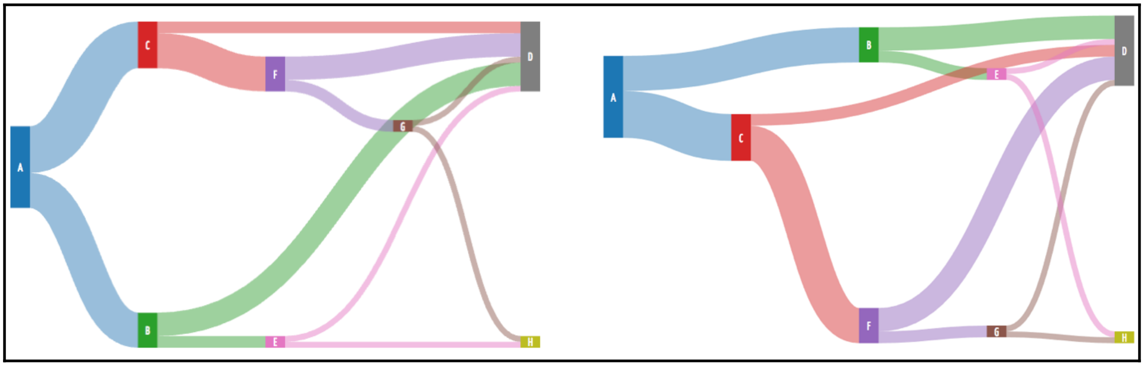


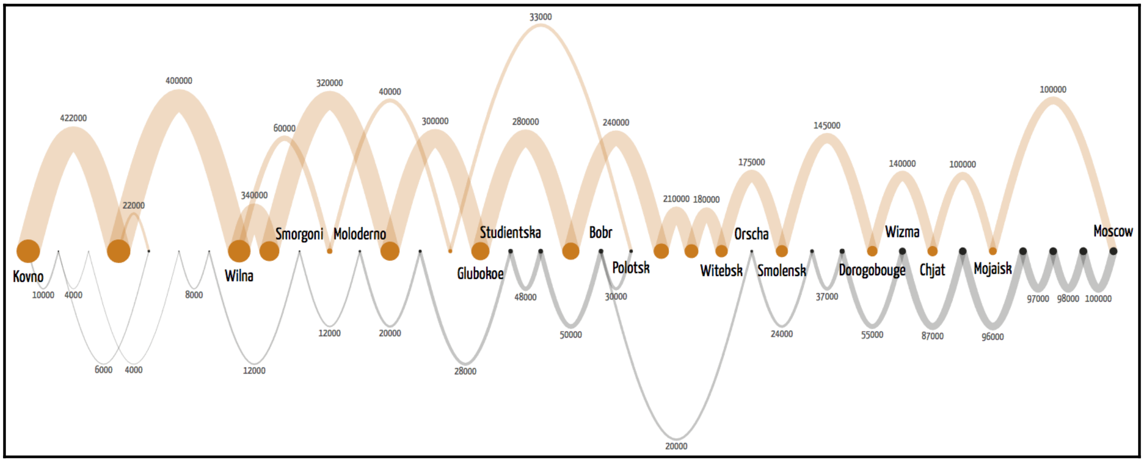

A
B
C
D
E
F
G
H
I

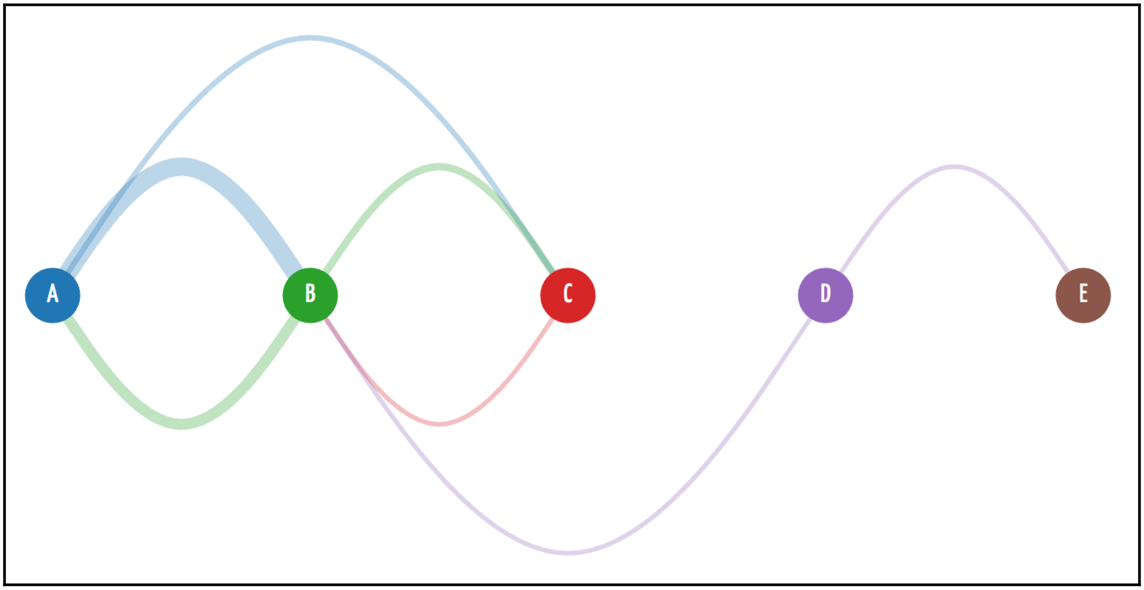

Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812-1813.
Dressée par M. Minard, Inspecteur Général des Ponts et Chaussées en retraite.
Paris, le 20 Novembre 1869.
Les nombres d'hommes présents sont représentés par les largeurs des zônes colorées à raison d'un millimètre pour dix mille hommes; ils sont de plus écrits en travers des zônes. Le rouge désigne les hommes qui entrent en Russie, le noir ceux qui en sortent. Les renseignements qui ont servi à dresser la carte ont été puisés dans les ouvrages de M.M. Thiers, de Ségur, de Fezensac, de Chambray et le journal inédit de Jacob, pharmacien de l'Armée depuis le 28 Octobre.
Pour mieux faire juger à l'œil la diminution de l'armée, j'ai supposé que les corps du Prince Jérôme et du Maréchal Davoust qui avaient été détachés sur Minsk et Mobilow et ont rejoint vers Orscha et Witebsk, avaient toujours marché avec l'armée.
MOSCOU
Kowno
Wilna
Polotzk
Gloubokoe
Witebsk
Smorgoni
Molodezno
Minsk
Studienska
Botr
Orscha
Mohilow
Smolensk
Dorogobouge
Wizma
Chjat
Mojaisk
Tarantino
Malo-jarosewli
TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.
Les Cosaques passent au galop le Niémen gelé.
— 26° le 7 X.bre
— 30° le 6 X.bre
— 24° le 1er X.bre
— 20° le 28 9.bre
— 11°
— 21° le 14 9.bre
— 8° le 9 9.bre
Pluie 24 8.bre
Zéro le 18 8.bre
Autog. par Regnier, 8. Pas. Ste Marie St Gain à Paris.
Imp. Lith. Regnier et Dourdet.

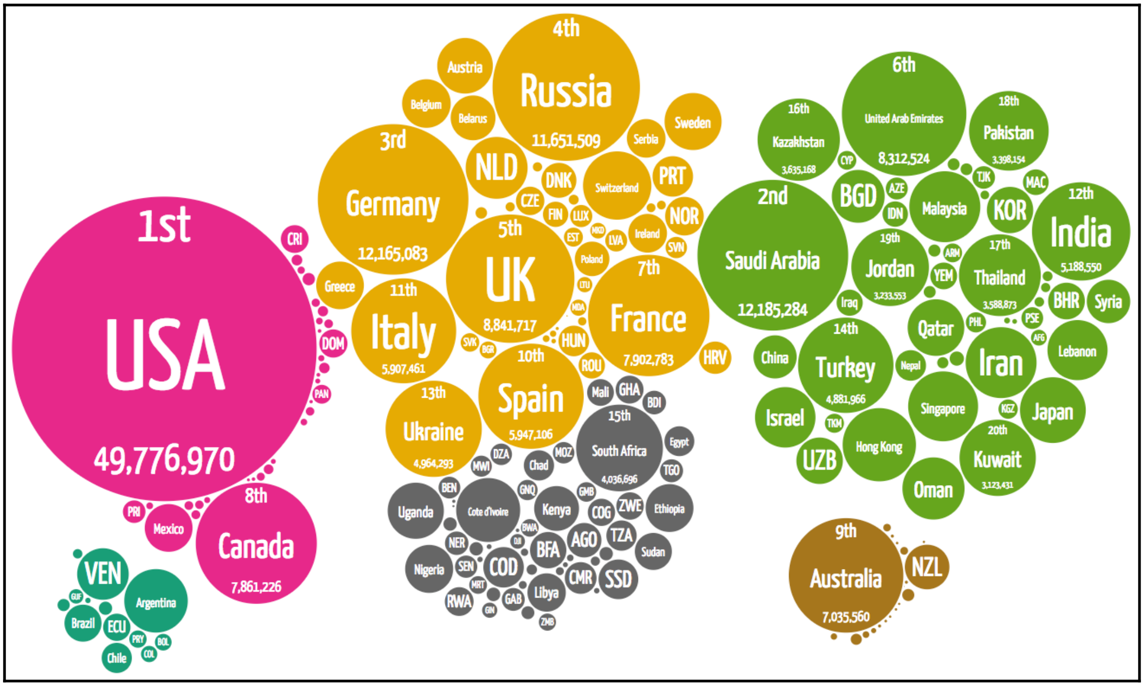

Kovno
Wilna
Gloubokoe
Polotsk
Witebsk
Smolensk
Dorogobouge
Chjat
Mojaisk
Moscow
Wizma
Orscha
Bobr
Studienska
Molodezno
Smorgoni
Dead
Survived

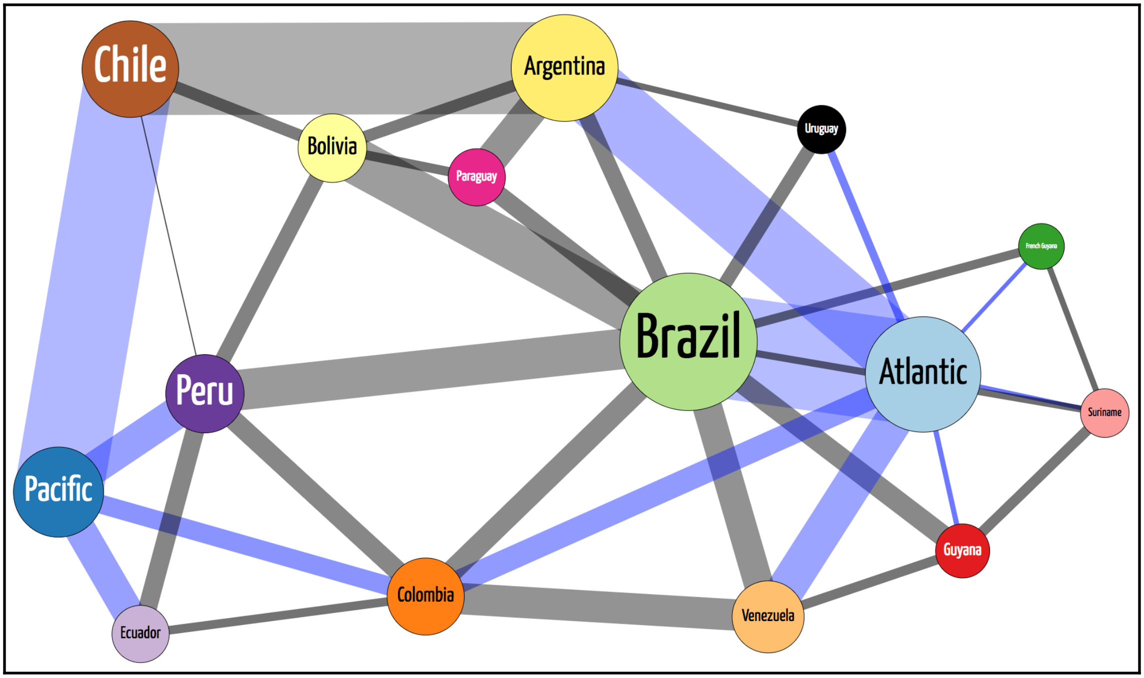

Warsaw Pact (1990)
USSR
Russia
Kazakhstan
Ukraine
Turkmenistan
Uzbekistan
Belarus
Kyrgyzstan
Tajikistan
Azerbaijan
Georgia
Moldova
Armenia
East Germany
Czechoslovakia
Lithuania
Latvia
Estonia
Poland
Hungary
Czech Republic
Slovakia
Romania
EEC (1990)
West Germany
Germany
France
Spain
Italy
United Kingdom
Greece
Portugal
Ireland
Denmark
Netherlands
Belgium
Luxembourg
EU 12 (1993)
Sweden
Finland
Austria
Yugoslavia
EU 15 (1995)
Cyprus
Slovenia
Malta
EU 25 (2004)
Bulgaria
Serbia + Montenegro
EU 27 (2007)
Serbia + Kosovo
EU 28 (2013)
Serbia
Kosovo
Montenegro
Croatia
Bosnia & Herzegovina
Macedonia

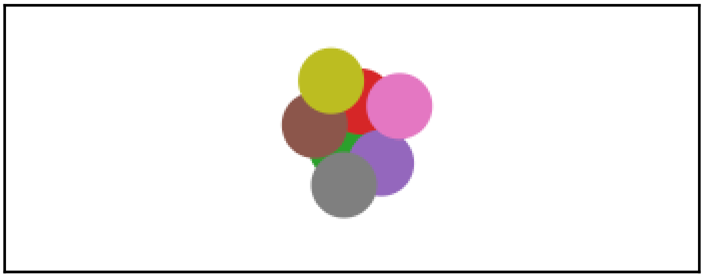

A
B
C
D
E
F
G
H

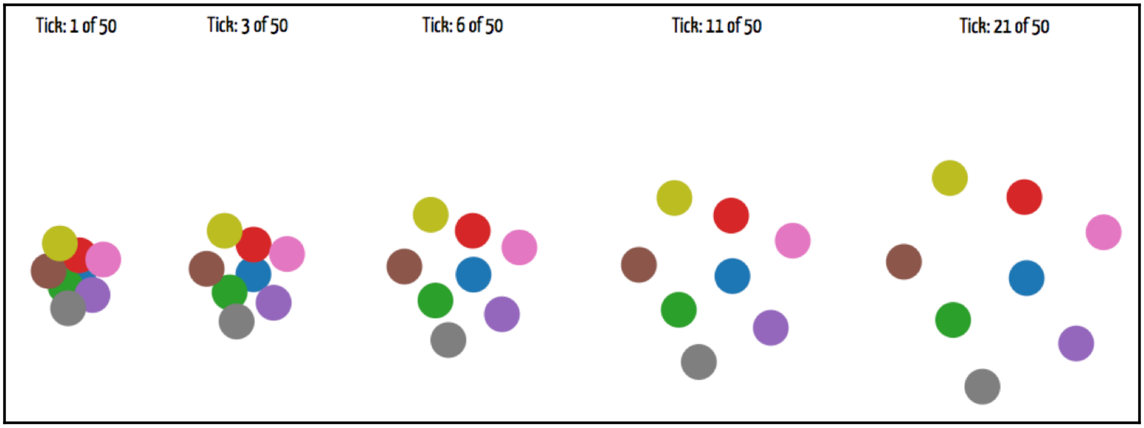

Kovno
Wilna
Smorgoni
Moloderno
Studientska
Glubokoe
Bobr
Polotsk
Witebsk
Orscha
Smolensk
Dorogobouge
Wizma
Chjat
Mojaisk
Moscow
422000
400000
22000
340000
60000
320000
40000
300000
33000
280000
240000
210000
180000
175000
145000
140000
100000
100000
10000
4000
6000
4000
8000
12000
12000
20000
28000
48000
50000
30000
20000
24000
37000
55000
87000
96000
97000
98000
100000

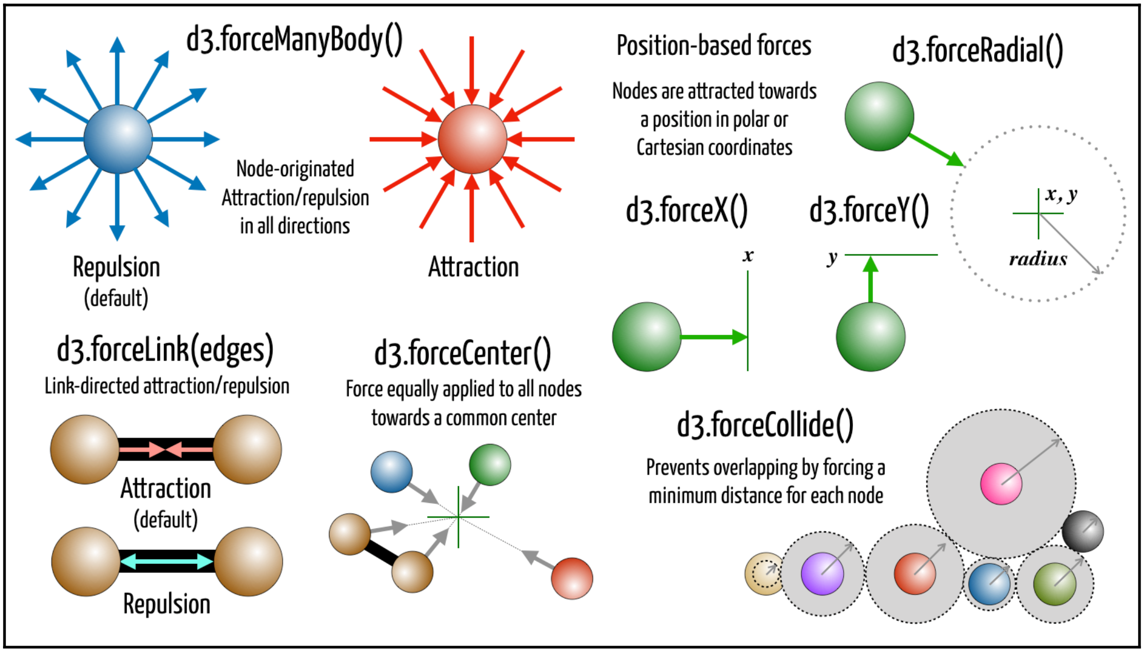

A
B
C
D
E

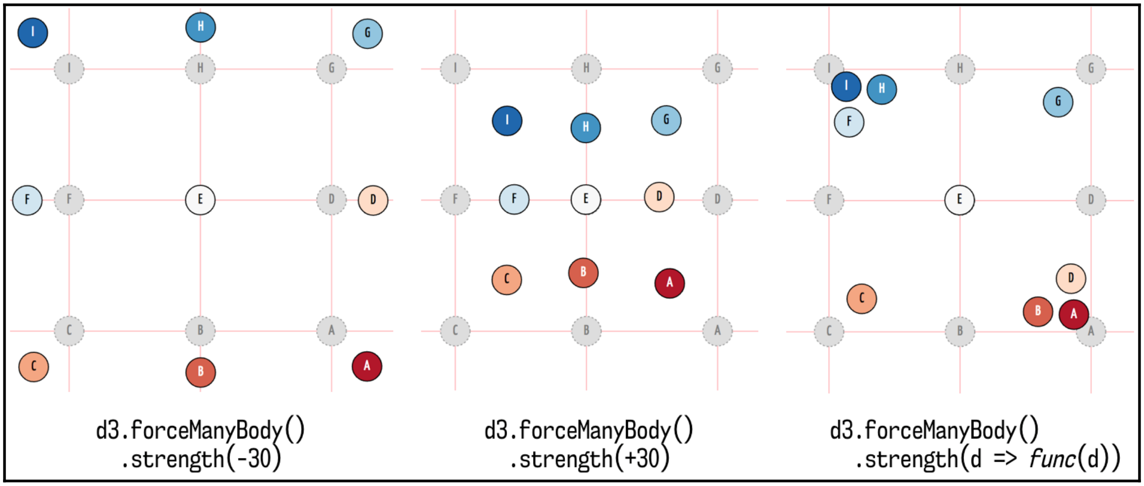

1st
USA
49,776,970
8th
Canada
7,861,226
CRI
DOM
PAN
PRI
Mexico
VEN
Argentina
Brazil
ECU
Chile
PRY
BOL
COL
GUF
3rd
Germany
12,165,083
4th
Russia
11,651,509
5th
UK
8,841,717
7th
France
7,902,783
10th
Spain
5,947,106
11th
Italy
5,907,461
13th
Ukraine
4,964,293
Austria
Belgium
Belarus
Sweden
Serbia
NLD
DNK
Switzerland
PRT
CZE
FIN
LUX
NOR
Ireland
EST
LVA
SVN
Poland
Greece
LTU
HUN
ROU
HRV
SVK
BGR
15th
South Africa
4,036,696
Mali
GHA
BDI
Egypt
TGO
MOZ
Chad
DZA
MWI
BEN
Uganda
Cote d'Ivoire
Kenya
GNQ
GMB
COG
ZWE
Ethiopia
NER
BFA
AGO
TZA
Sudan
Nigeria
SEN
COD
CMR
SSD
RWA
GAB
Libya
MRT
GIN
ZMB
2nd
Saudi Arabia
12,185,284
6th
United Arab Emirates
8,312,524
12th
India
5,188,550
14th
Turkey
4,881,966
16th
Kazakhstan
3,635,168
17th
Thailand
3,588,873
18th
Pakistan
3,398,154
19th
Jordan
3,233,553
20th
Kuwait
3,123,431
CYP
BGD
AZE
IDN
Malaysia
TJK
MAC
KOR
ARM
YEM
Iraq
BHR
Syria
PSE
PHL
Qatar
Iran
Lebanon
China
Nepal
AFG
Israel
TKM
Singapore
KGZ
Japan
UZB
Hong Kong
Oman
9th
Australia
7,035,560
NZL

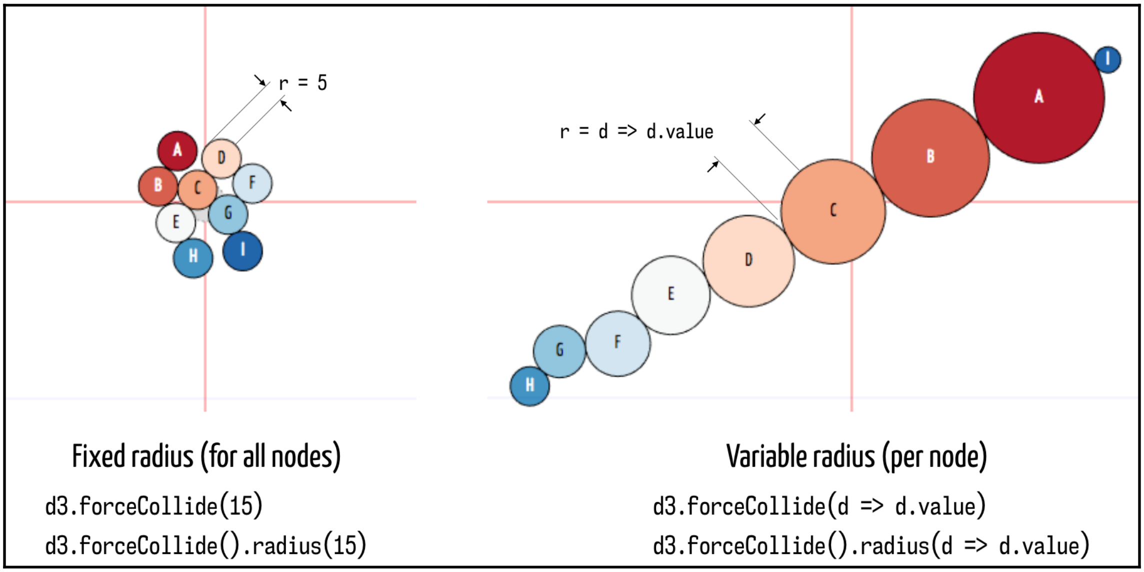

Chile
Argentina
Bolivia
Paraguay
Uruguay
French Guyana
Brazil
Atlantic
Peru
Pacific
Suriname
Guyana
Colombia
Venezuela
Ecuador

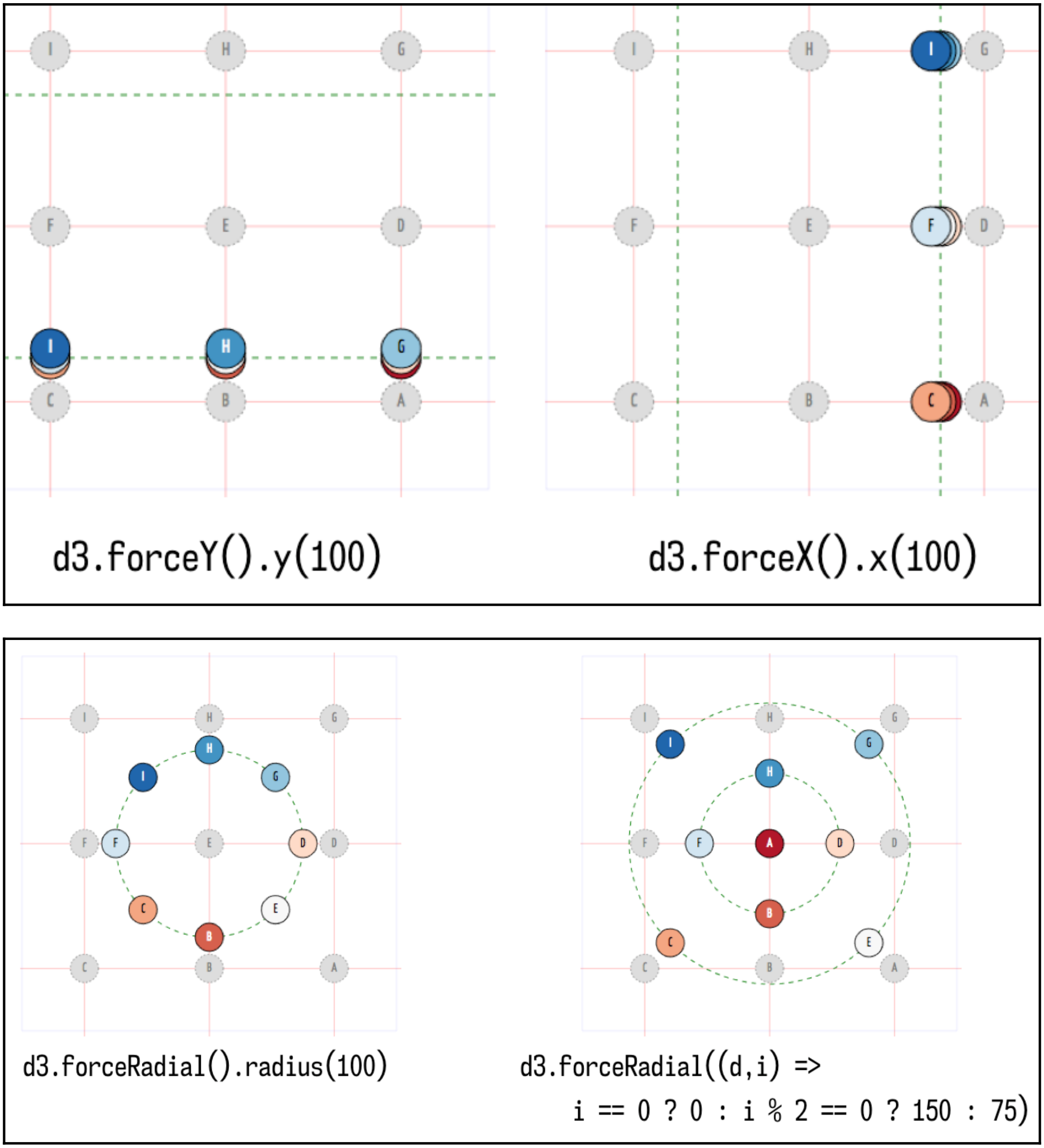



Tick: 1 of 50
Tick: 3 of 50
Tick: 6 of 50
Tick: 11 of 50
Tick: 21 of 50

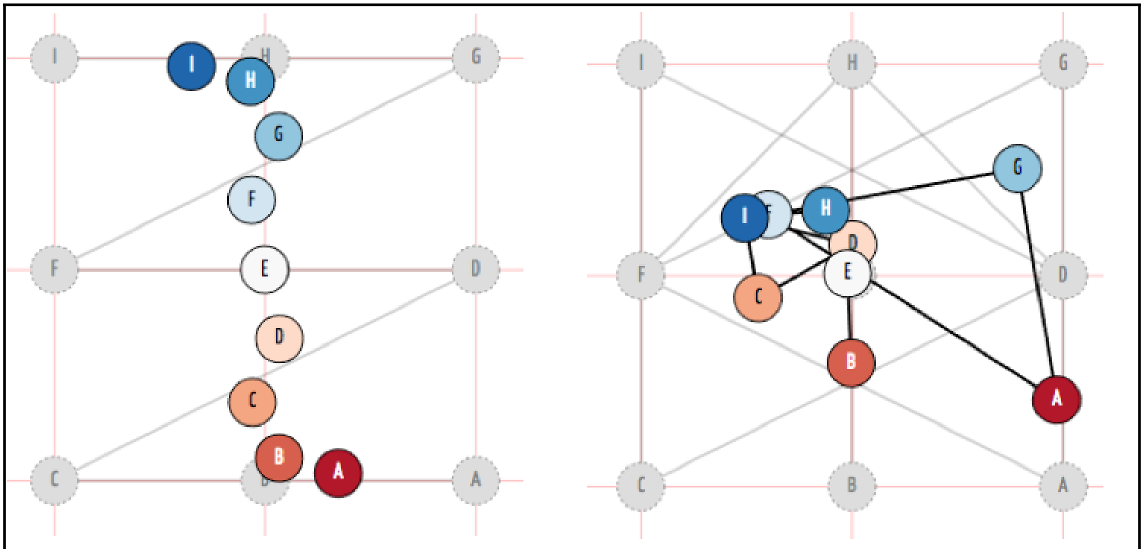

d3.forceManyBody()
Node-originated Attraction/repulsion in all directions
Repulsion (default)
Attraction
Position-based forces
Nodes are attracted towards a position in polar or Cartesian coordinates
d3.forceRadial()
x, y
radius
d3.forceX()
d3.forceY()
x
y
d3.forceLink(edges)
Link-directed attraction/repulsion
Attraction (default)
Repulsion
d3.forceCenter()
Force equally applied to all nodes towards a common center
d3.forceCollide()
Prevents overlapping by forcing a minimum distance for each node


d3.forceManyBody()
.strength(-30)
d3.forceManyBody()
.strength(+30)
d3.forceManyBody()
.strength(d => func(d))


r = 5
r = d => d.value
Fixed radius (for all nodes)
d3.forceCollide(15)
d3.forceCollide().radius(15)
Variable radius (per node)
d3.forceCollide(d => d.value)
d3.forceCollide().radius(d => d.value)



d3.forceY().y(100)
d3.forceX().x(100)
d3.forceRadial().radius(100)
d3.forceRadial((d,i) =>
i == 0 ? 0 : i % 2 == 0 ? 150 : 75)

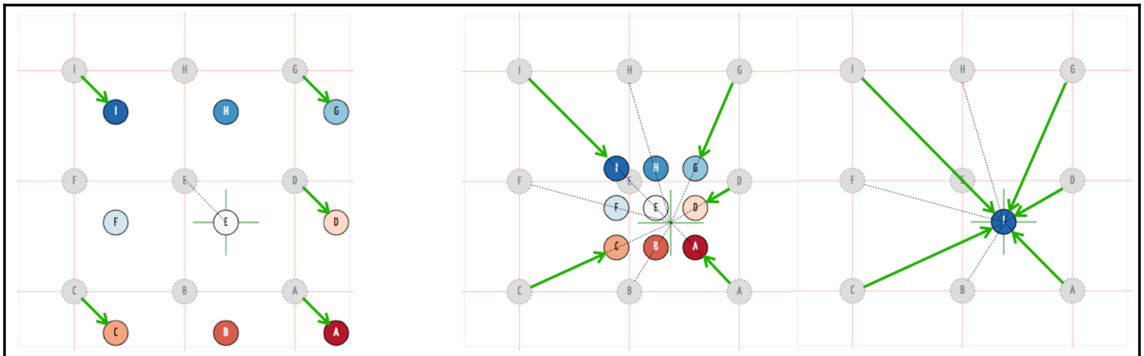


Same strength for all nodes

```
d3.forceLink(edges)
  .strength(.01)
```

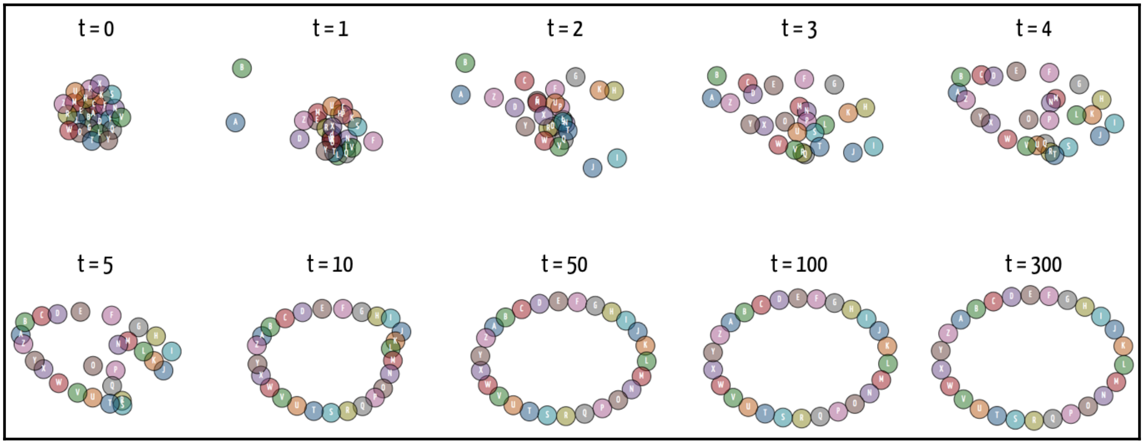
Variable strength per node

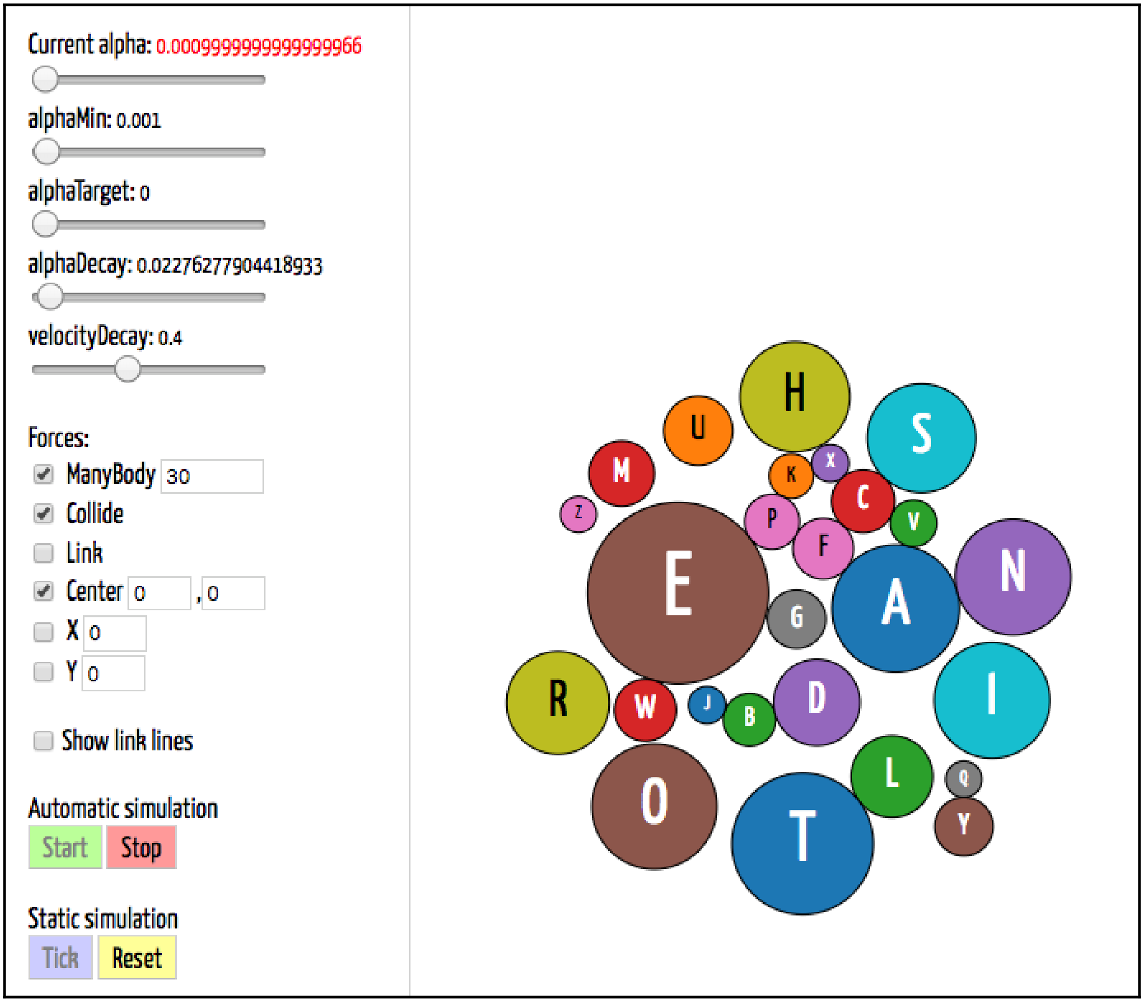
```
d3.forceLink(edges)
  .strength(d => d.index/200)
```

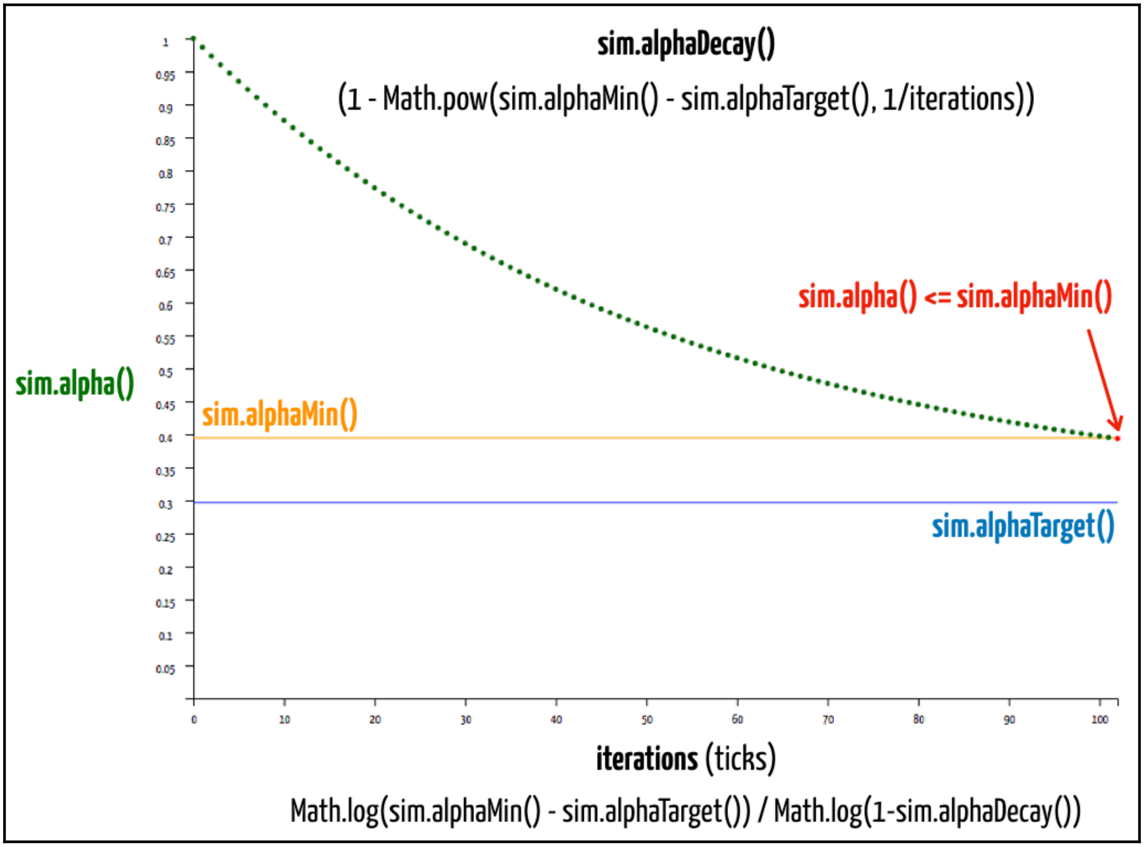


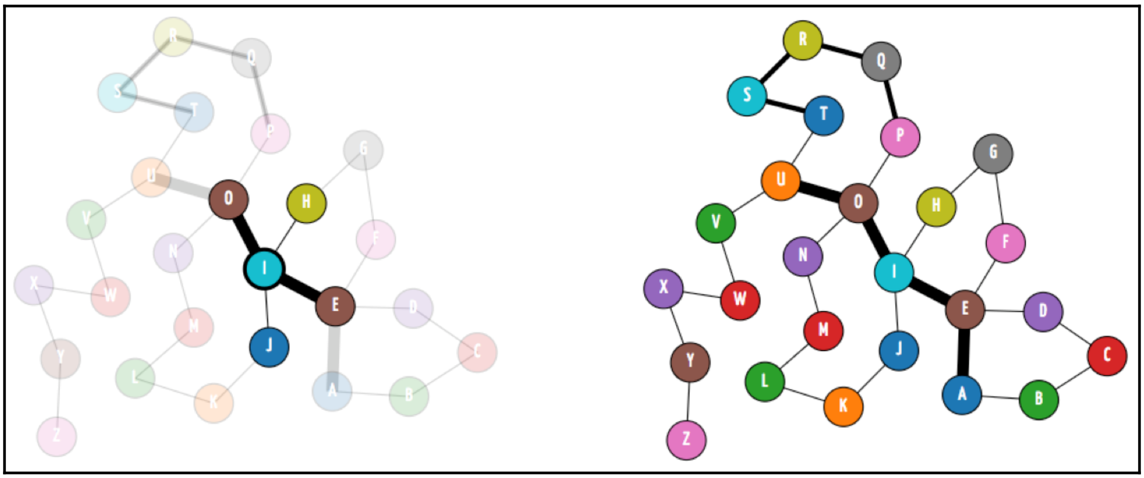
```
d3.forceCenter().x(50).y(50)
```

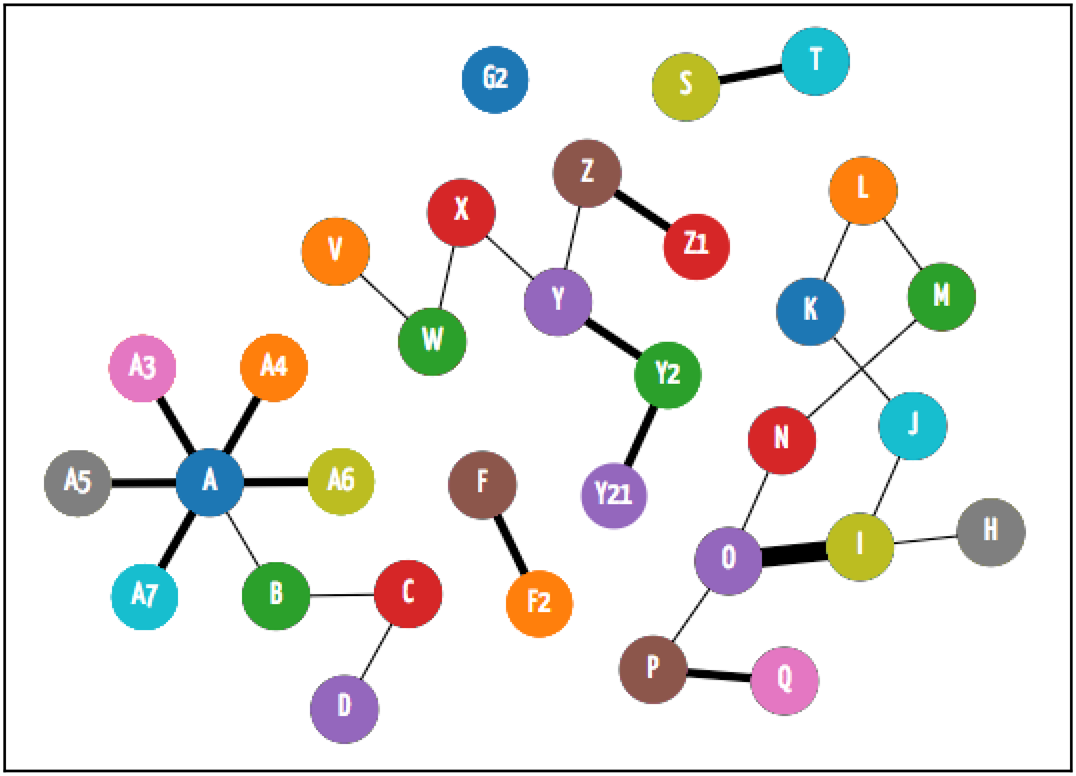
```
d3.forceY().y(50)
d3.forceX().x(50)
```

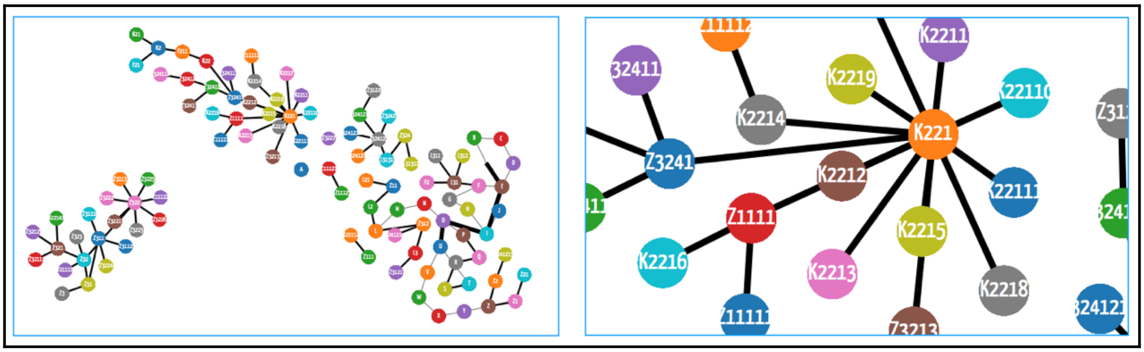

t = 0
t = 1
t = 2
t = 3
t = 4
t = 5
t = 10
t = 50
t = 100
t = 300


Current alpha: 0.0009999999999999966
alphaMin: 0.001
alphaTarget: 0
alphaDecay: 0.02276277904418933
velocityDecay: 0.4
Forces:
ManyBody 30
Collide
Link
Center 0 , 0
X 0
Y 0
Show link lines
Automatic simulation
Start
Stop
Static simulation
Tick
Reset
H
U
S
M
X
K
C
Z
P
V
F
N
E
A
G
I
R
W
J
B
D
L
Q
O
T
Y

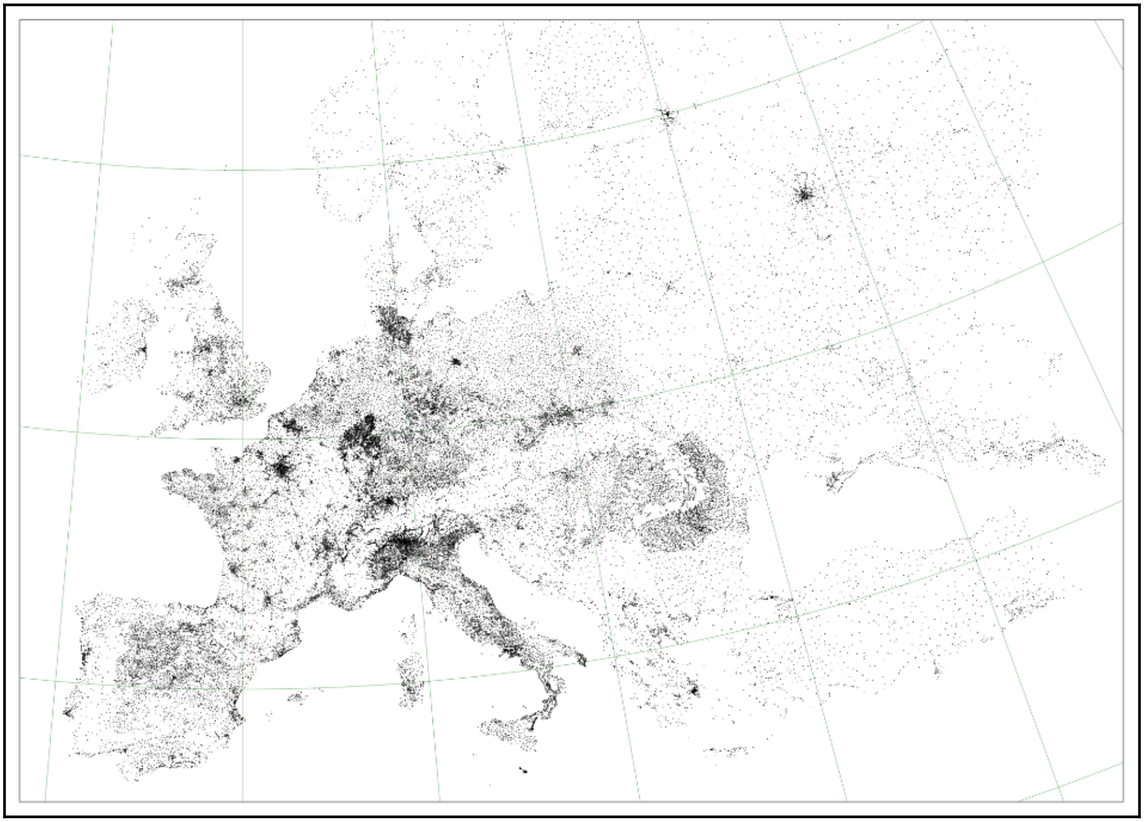

sim.alphaDecay()
(1 - Math.pow(sim.alphaMin() - sim.alphaTarget(), 1/iterations))
sim.alpha() <= sim.alphaMin()
sim.alpha()
sim.alphaMin()
sim.alphaTarget()
iterations (ticks)
Math.log(sim.alphaMin() - sim.alphaTarget()) / Math.log(1-sim.alphaDecay())

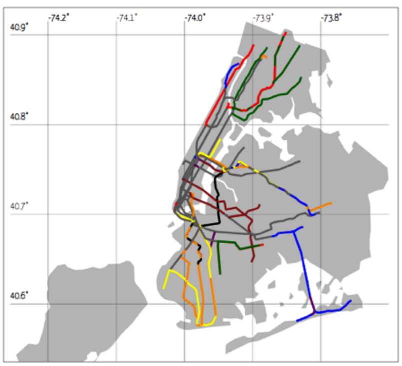


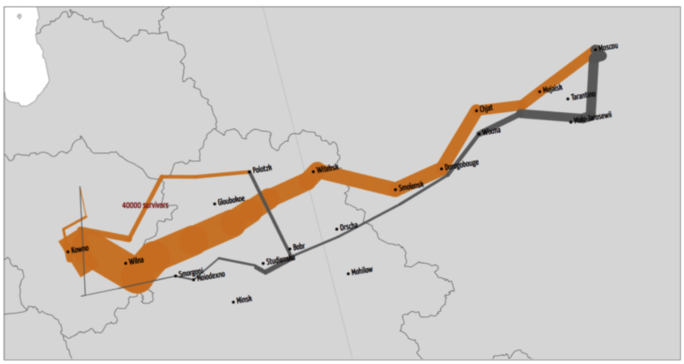

G2
S
T
Z
X
Z1
L
V
Y
M
K
W
A3
A4
Y2
N
J
A5
A
A6
F
Y21
H
O
I
A7
B
C
F2
P
Q
D

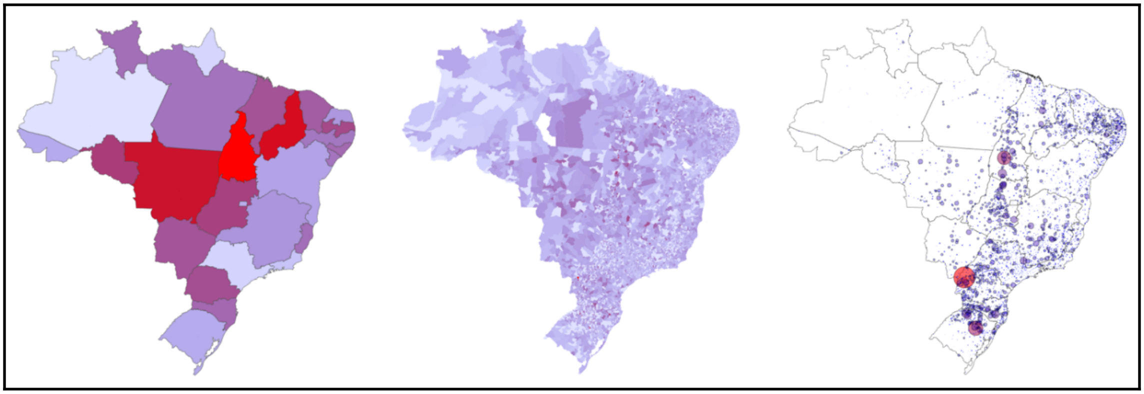

K2211
K2219
K2214
K221
Z3241
K2212
Z1111
K2215
K2216
K2213
K2218

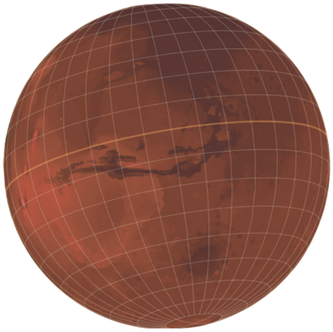
Chapter 11: Visualizing Geographical Data



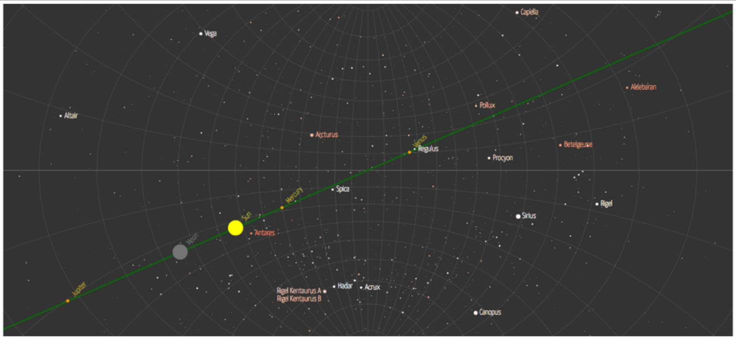




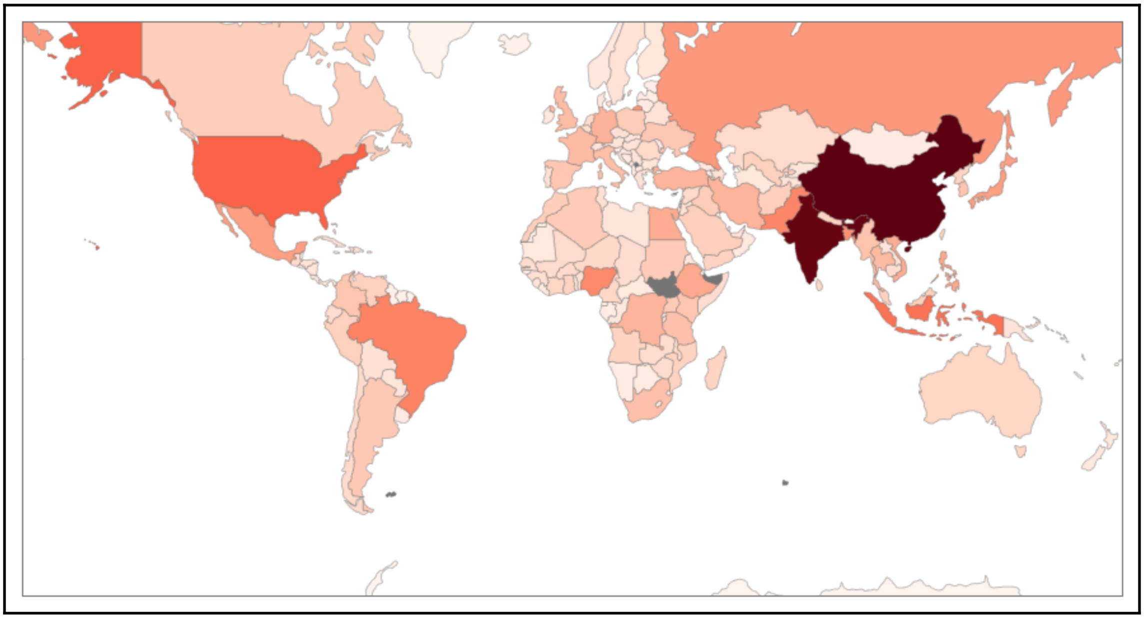


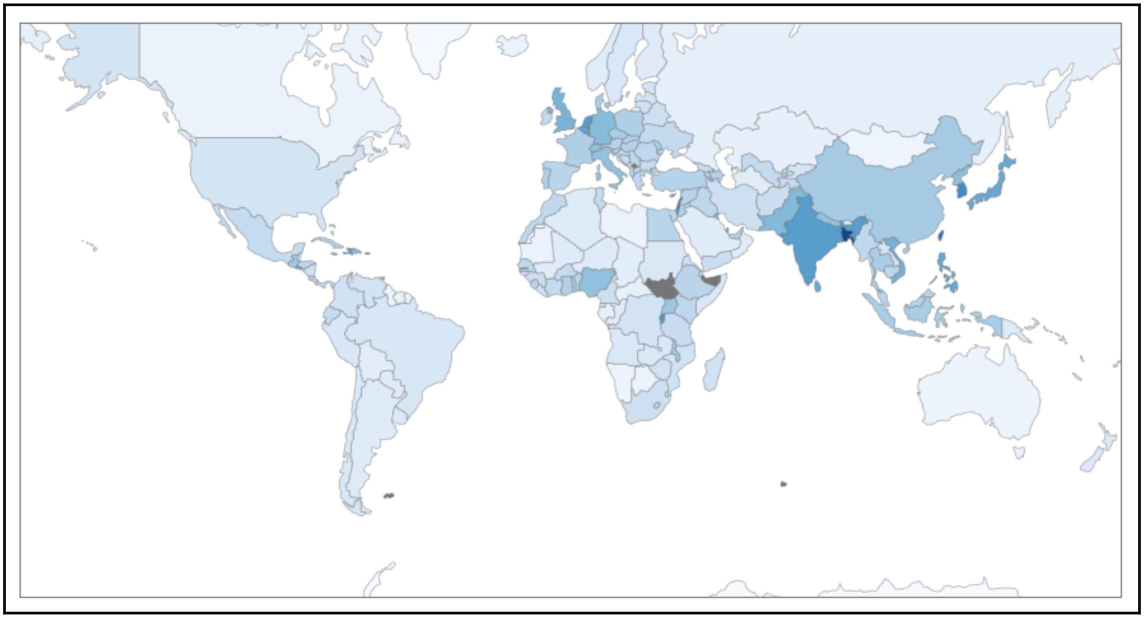


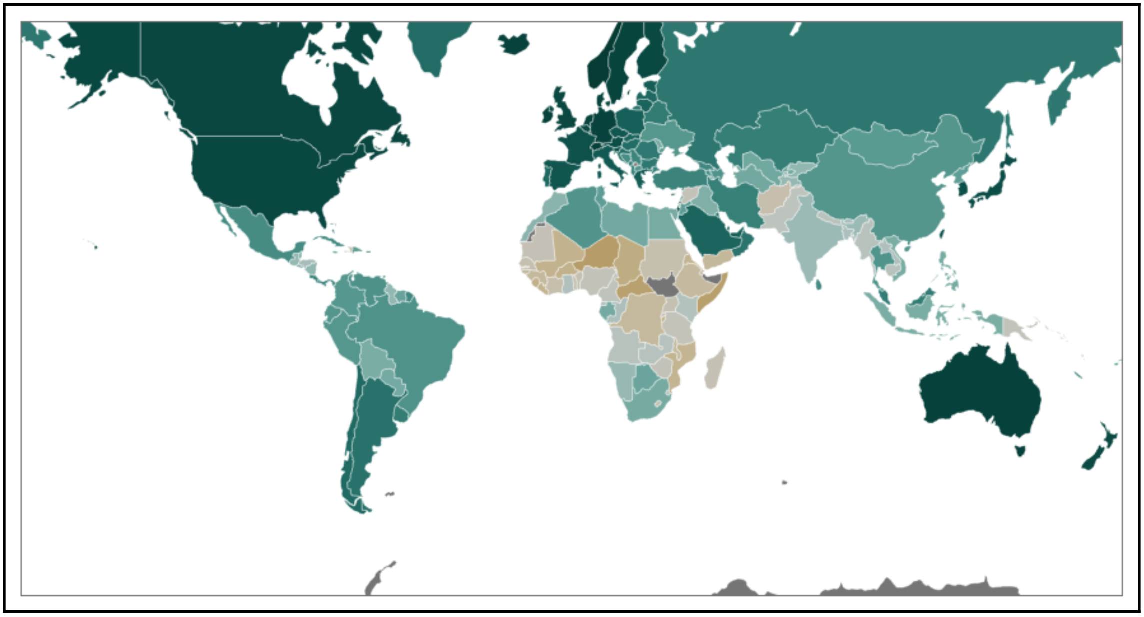
Mars using **d3.geoOrthographic()**

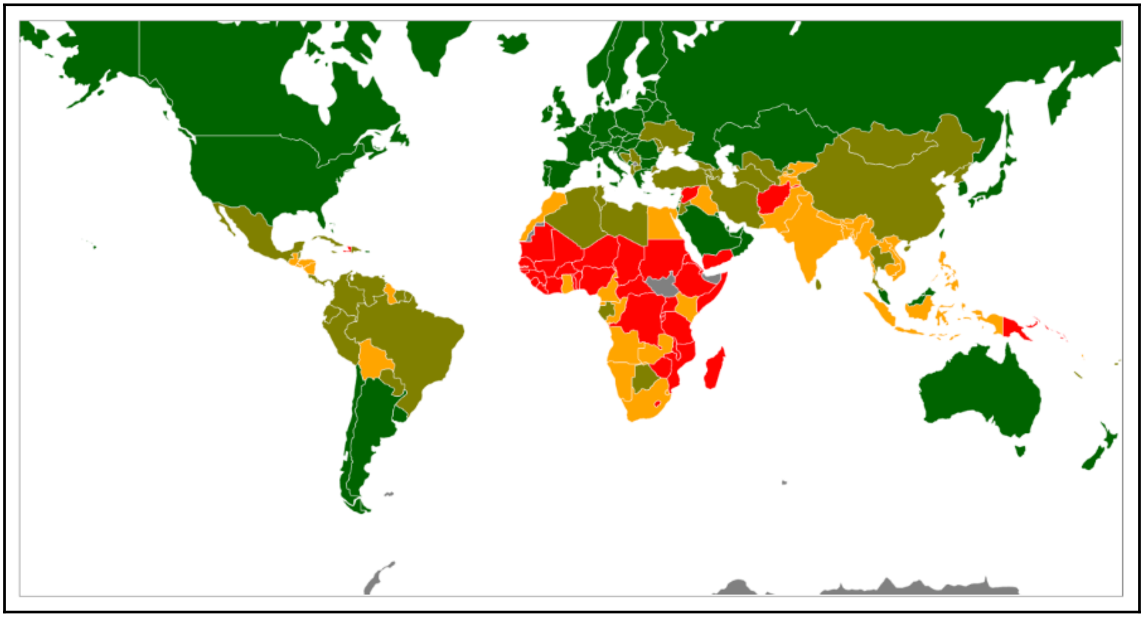


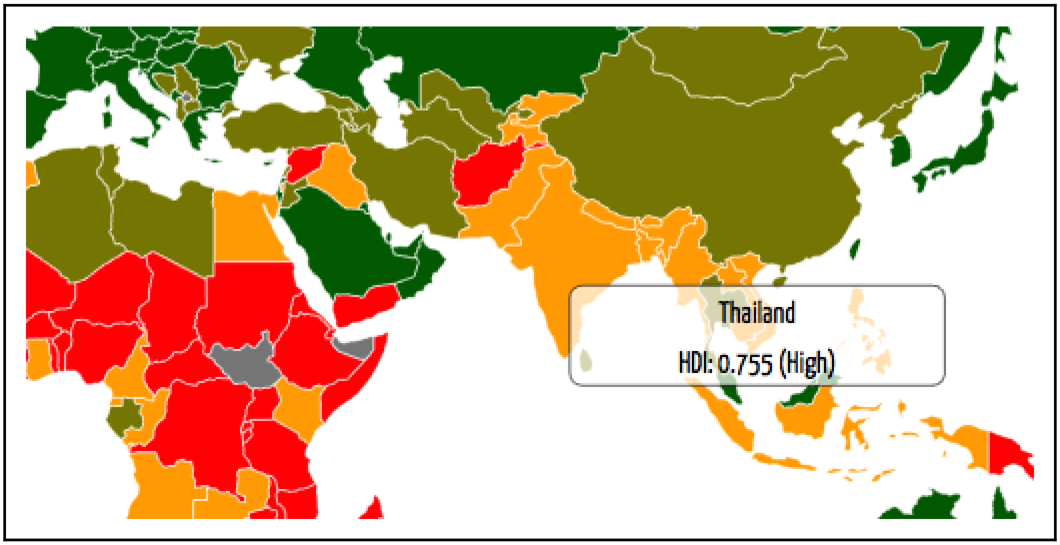
Stars, planets and the ecliptic using **d3.geoStereographic()**

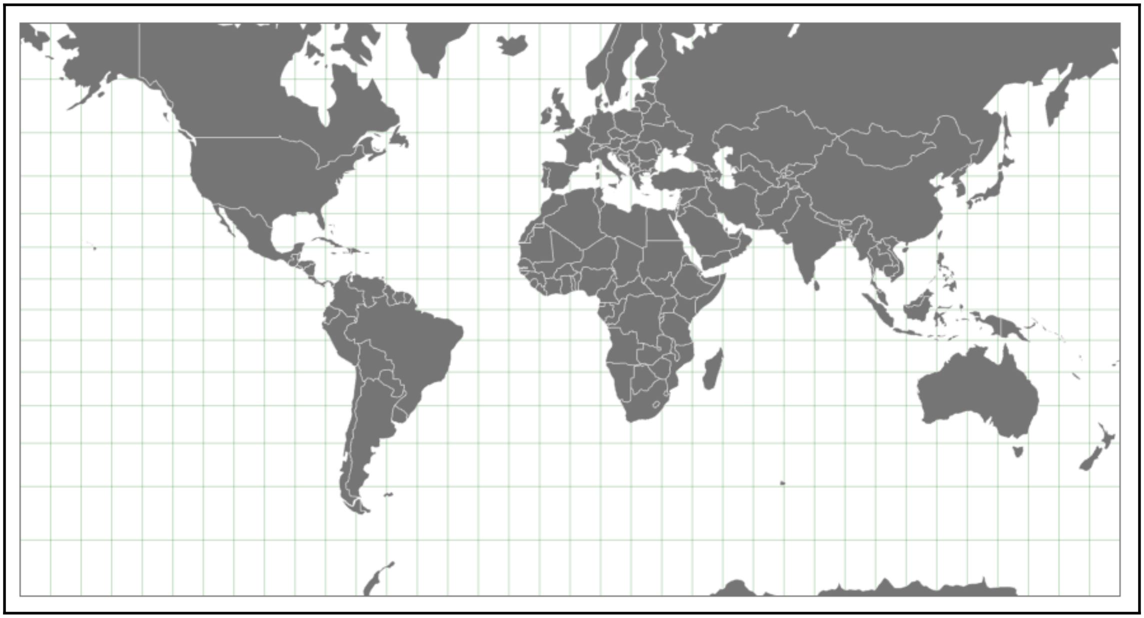


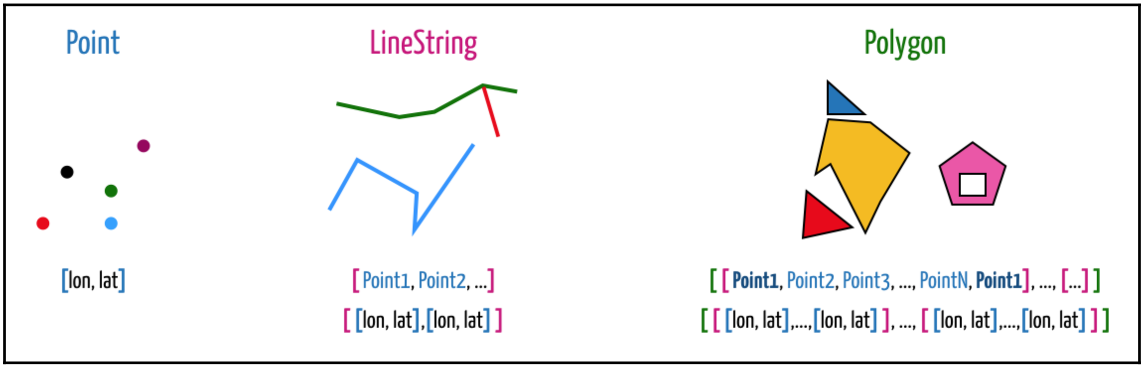


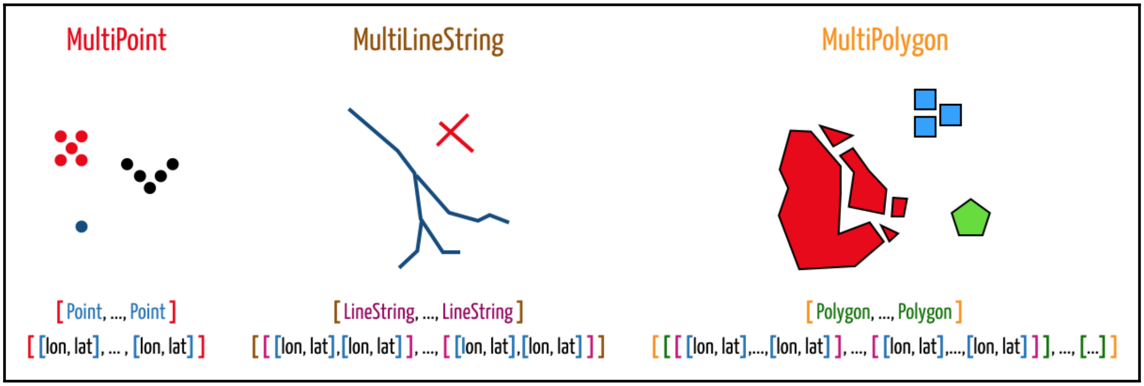


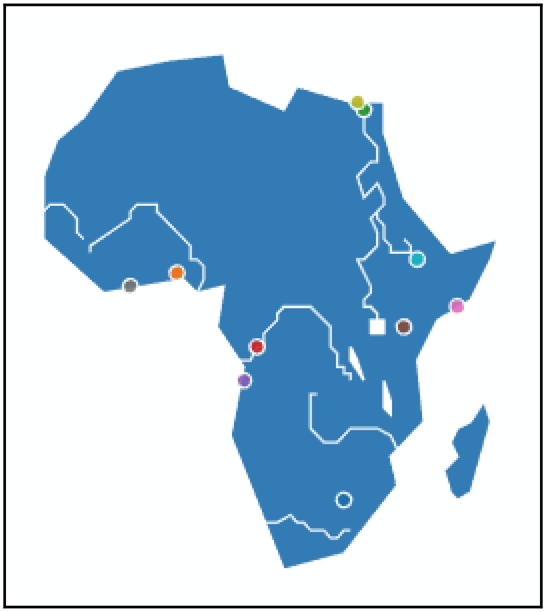


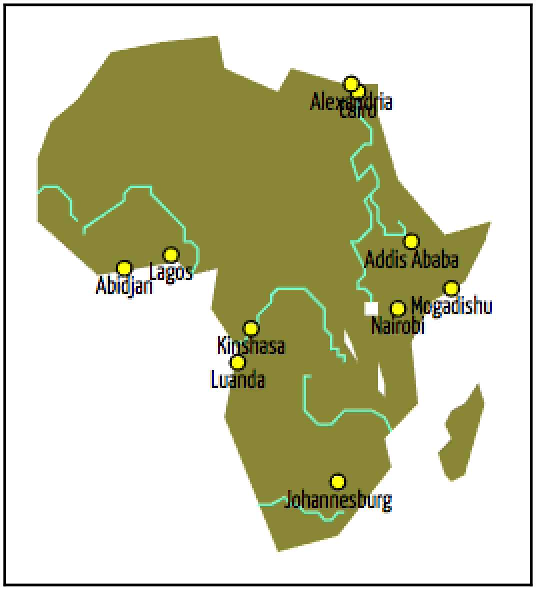

Thailand
HDI: 0.755 (High)

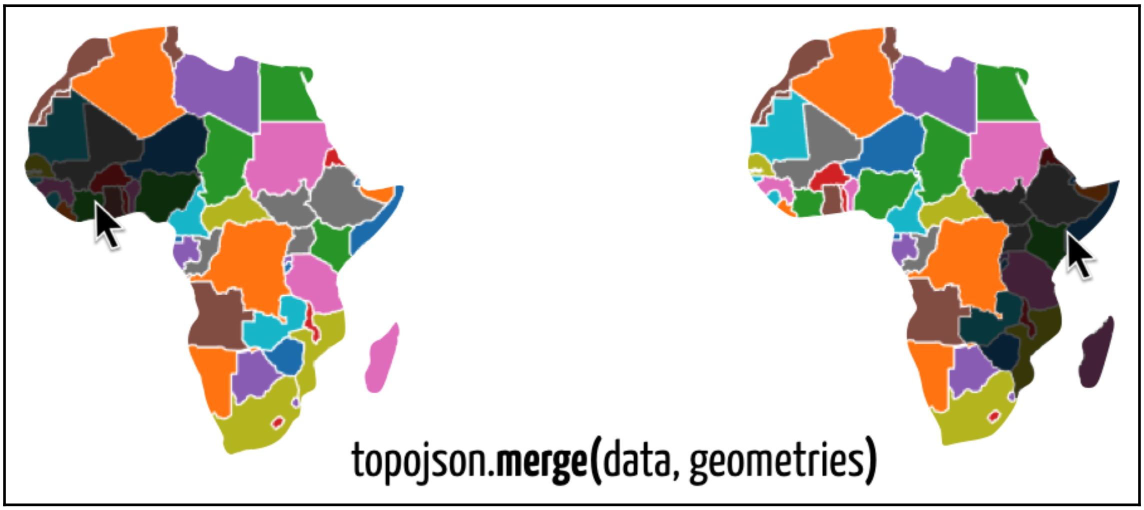


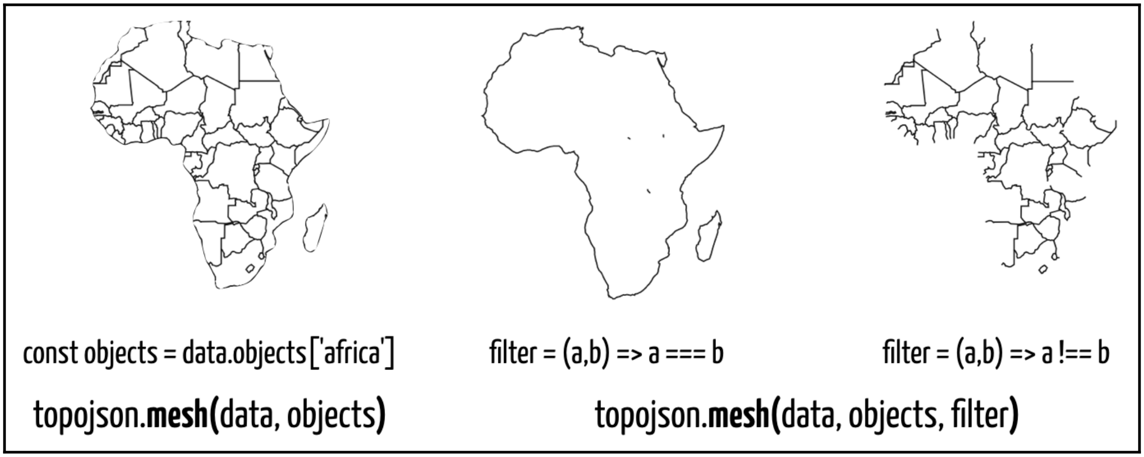

Point
LineString
Polygon
[lon, lat]
[Point1, Point2, ...]
[[lon, lat],[lon, lat]]
[[Point1, Point2, Point3, ..., PointN, Point1], ..., [...]]
[[[lon, lat],....,[lon, lat]], ..., [[lon, lat],....,[lon, lat]]]

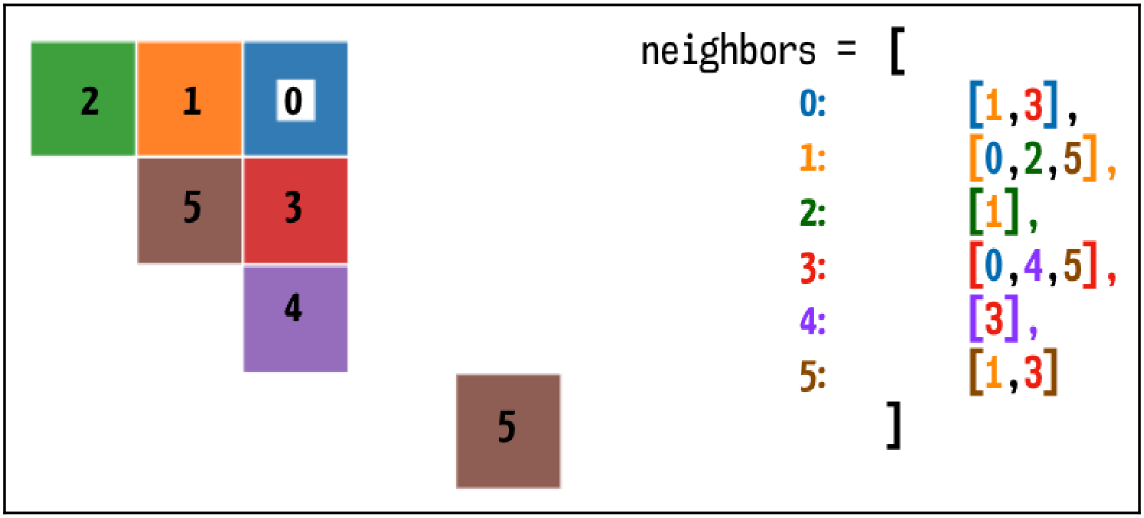

MultiPoint
MultiLineString
MultiPolygon
[Point, ..., Point]
[[lon, lat], ... , [lon, lat]]
[LineString, ..., LineString]
[[[lon, lat],[lon, lat]], ..., [[lon, lat],[lon, lat]]]
[Polygon, ..., Polygon]
[[[[lon, lat],...,[lon, lat]], ..., [[lon, lat],...,[lon, lat]]], ..., [...]]

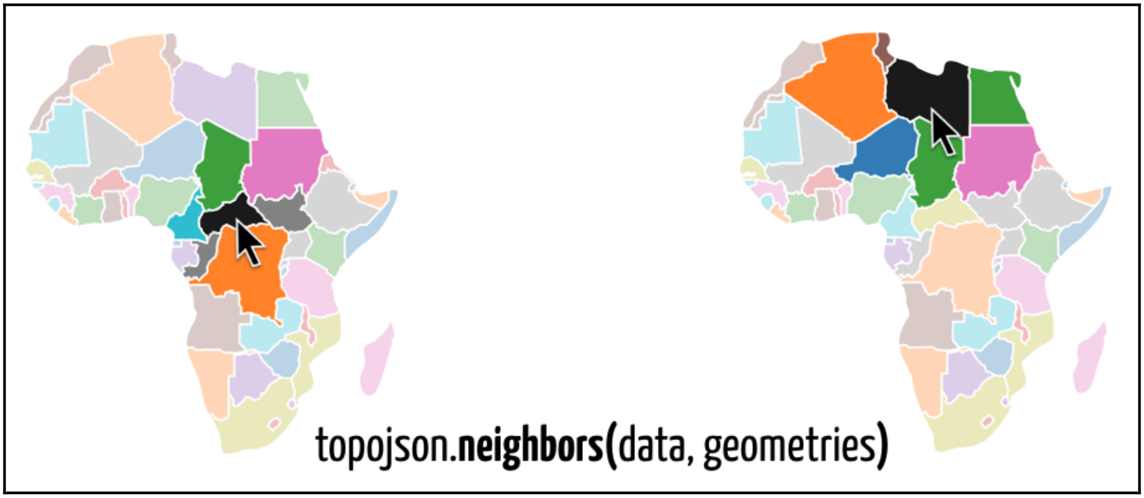



Alexandria
Cairo
Lagos
Abidjan
Addis Ababa
Mogadishu
Nairobi
Kinshasa
Luanda
Johannesburg

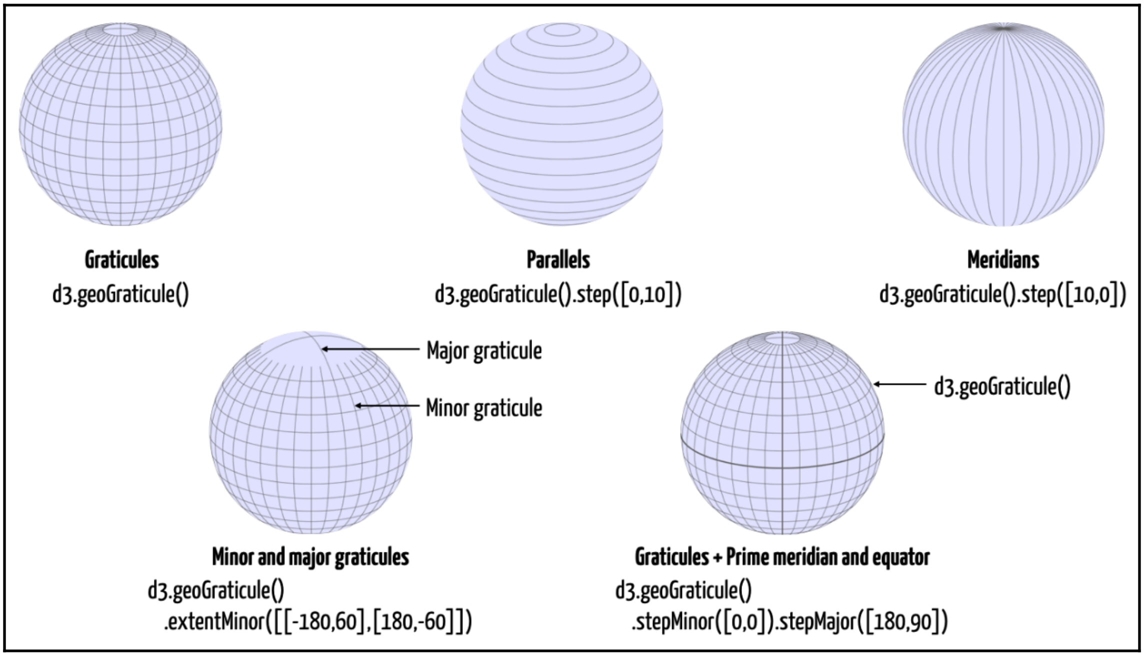

topojson.**merge**(data, geometries)


const objects = data.objects['africa']
filter = (a,b) => a === b
filter = (a,b) => a !== b
topojson.**mesh**(data, objects)
topojson.**mesh**(data, objects, filter)

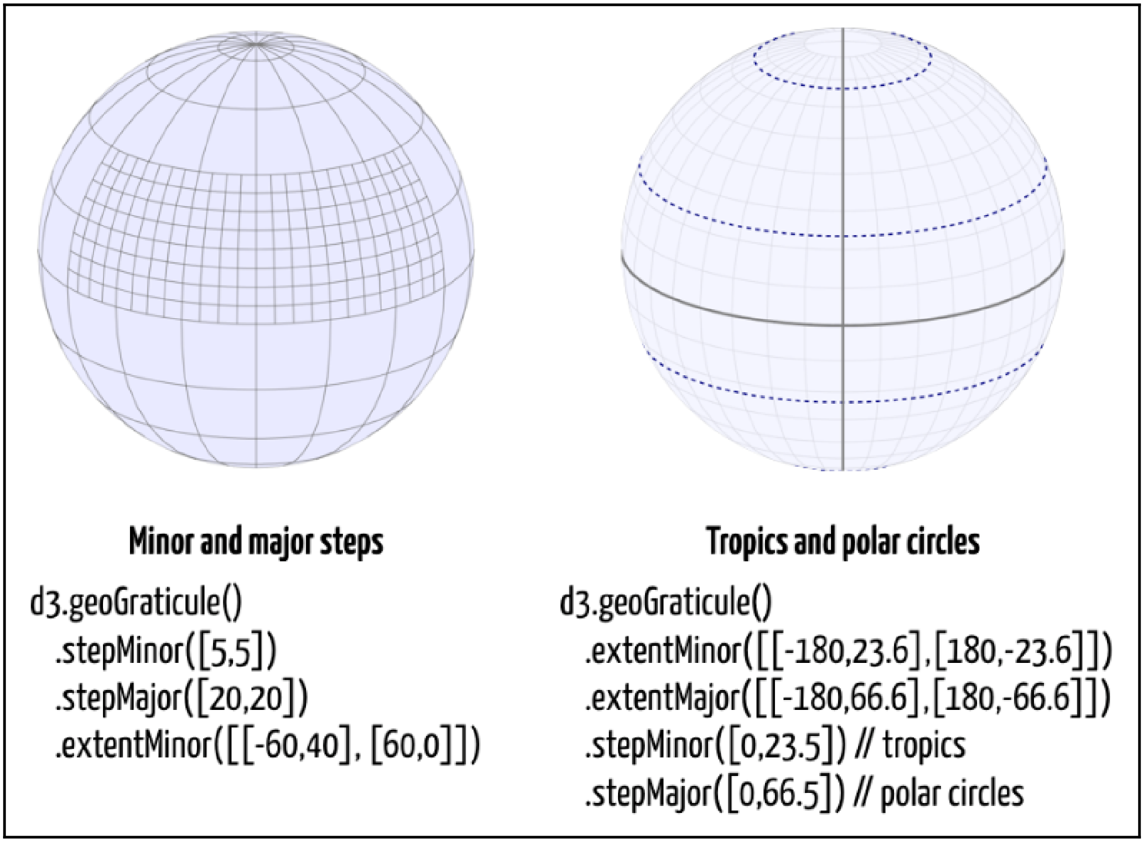

2
1
0
5
3
4
5
neighbors = [
0: [1,3],
1: [0,2,5],
2: [1],
3: [0,4,5],
4: [3],
5: [1,3]
]

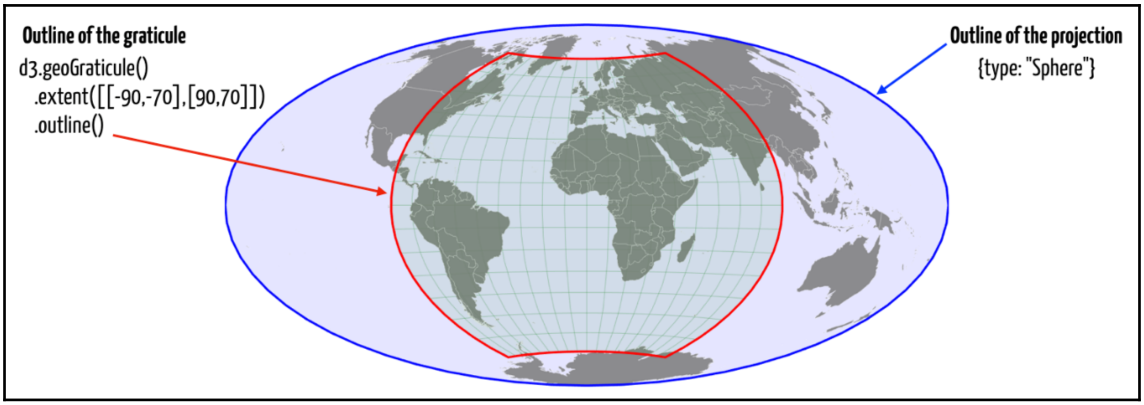


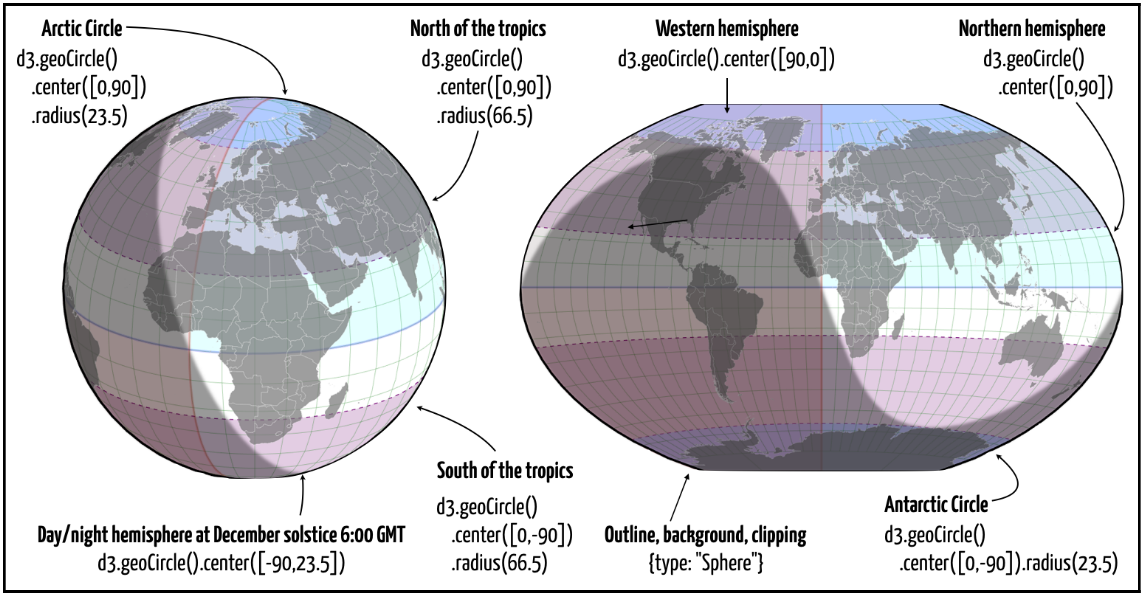
topojson.neighbors(data, geometries)

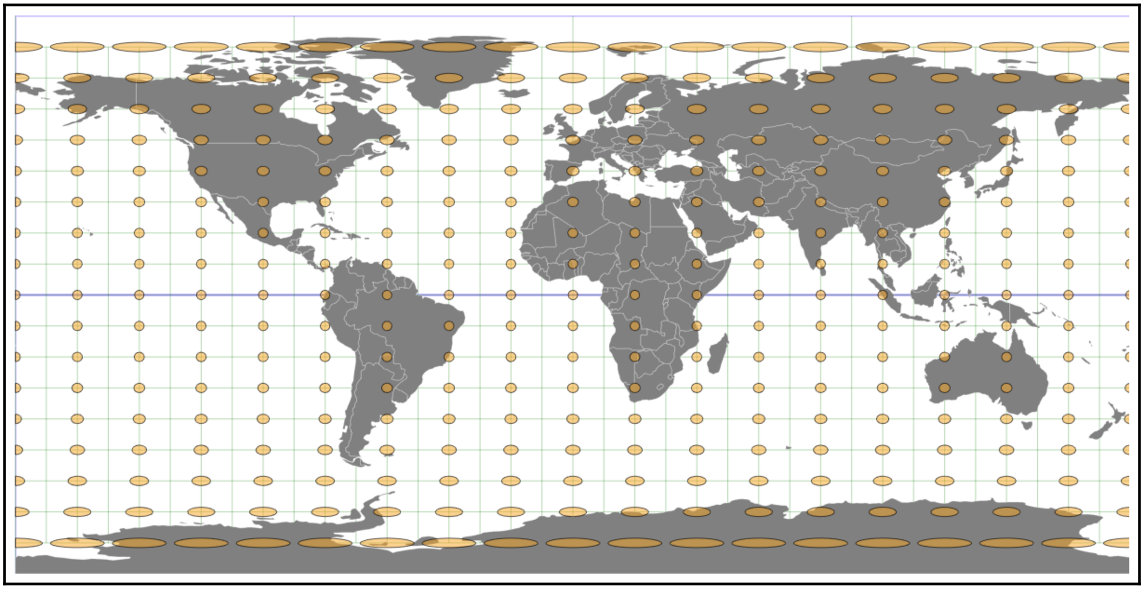


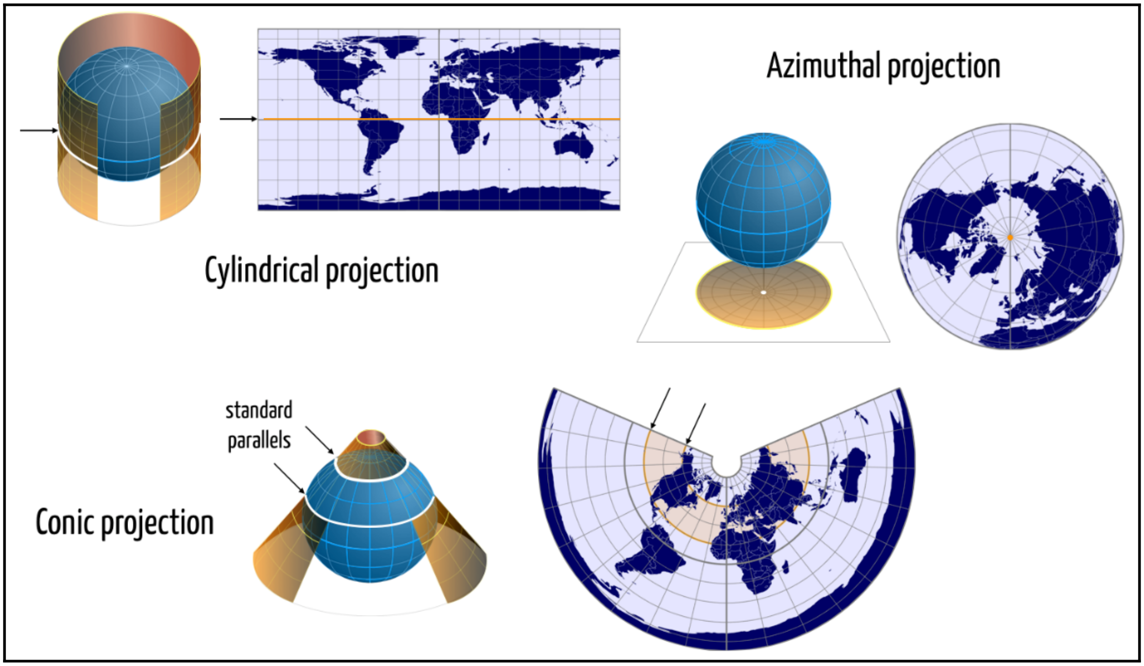
Graticules
d3.geoGraticule()
Parallels
d3.geoGraticule().step([0,10])
Meridians
d3.geoGraticule().step([10,0])
Major graticule
Minor graticule
d3.geoGraticule()
Minor and major graticules
d3.geoGraticule()
.extentMinor([[-180,60],[180,-60]])
Graticules + Prime meridian and equator
d3.geoGraticule()
.stepMinor([0,0]).stepMajor([180,90])

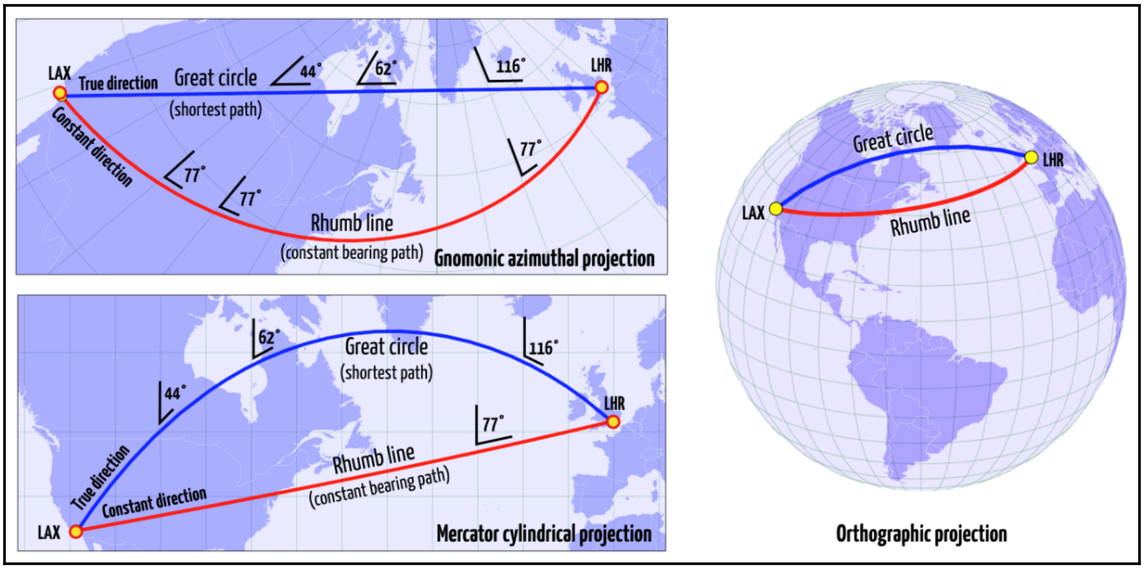

Minor and major steps
d3.geoGraticule()
.stepMinor([5,5])
.stepMajor([20,20])
.extentMinor([[-60,40], [60,0]])
Tropics and polar circles
d3.geoGraticule()
.extentMinor([[-180,23.6],[180,-23.6]])
.extentMajor([[-180,66.6],[180,-66.6]])
.stepMinor([0,23.5]) // tropics
.stepMajor([0,66.5]) // polar circles

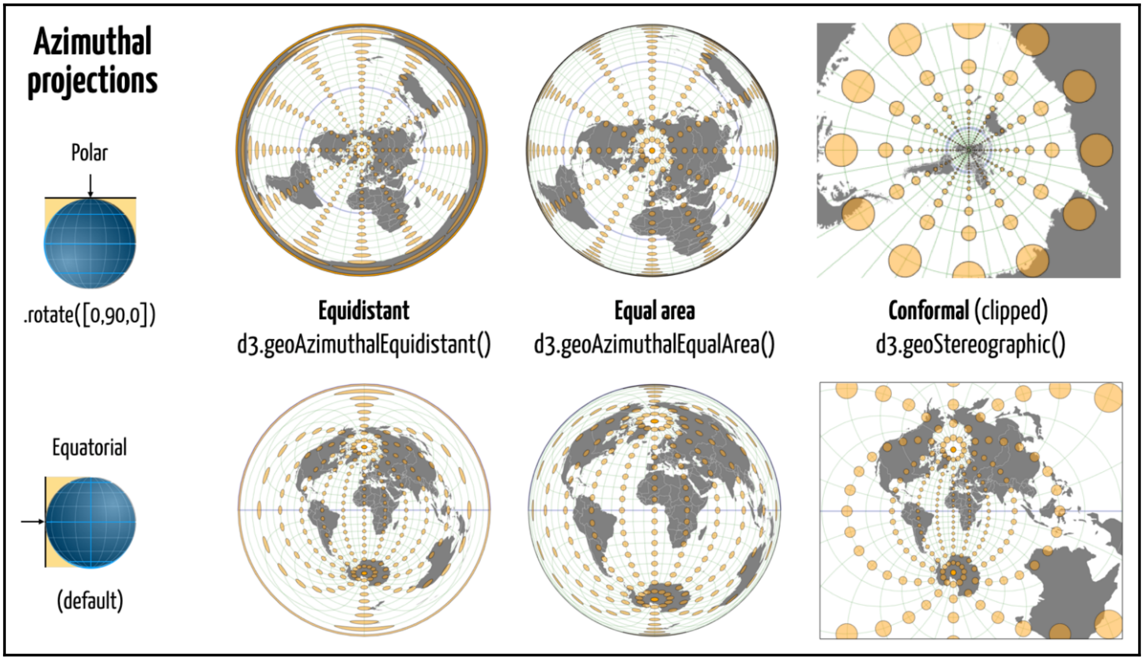

Outline of the graticule
d3.geoGraticule()
.extent([[-90,-70],[90,70]])
.outline()
Outline of the projection
{type: "Sphere"}

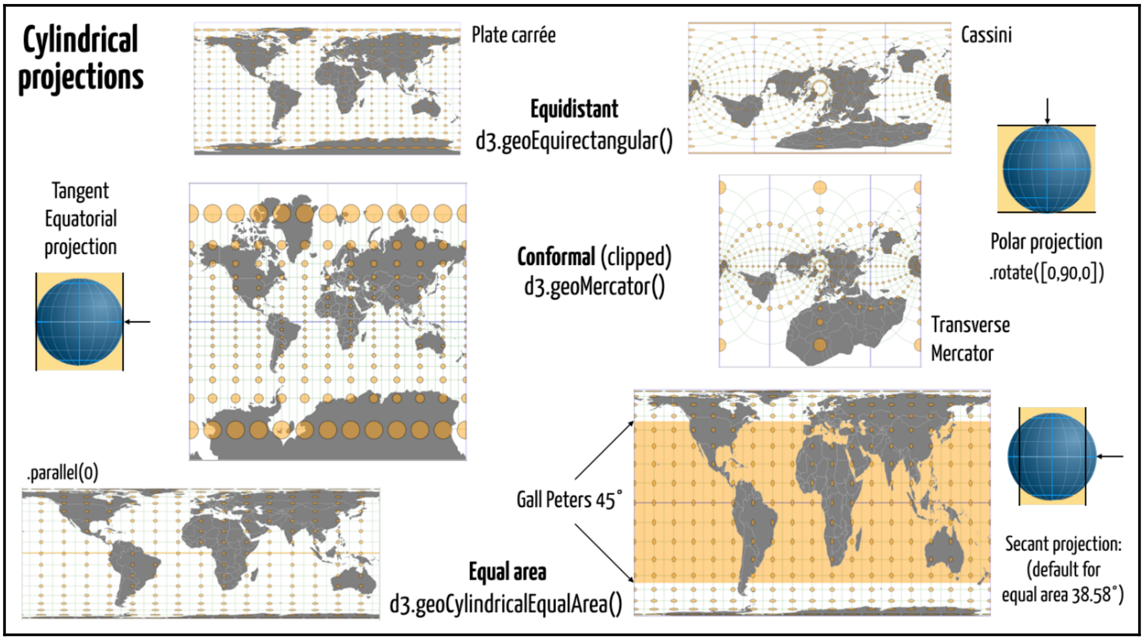

Arctic Circle
d3.geoCircle()
.center([0,90])
.radius(23.5)
North of the tropics
d3.geoCircle()
.center([0,90])
.radius(66.5)
Western hemisphere
d3.geoCircle().center([90,0])
Northern hemisphere
d3.geoCircle()
.center([0,90])
South of the tropics
d3.geoCircle()
.center([0,-90])
.radius(66.5)
Day/night hemisphere at December solstice 6:00 GMT
d3.geoCircle().center([-90,23.5])
Outline, background, clipping
{type: "Sphere"}
Antarctic Circle
d3.geoCircle()
.center([0,-90]).radius(23.5)

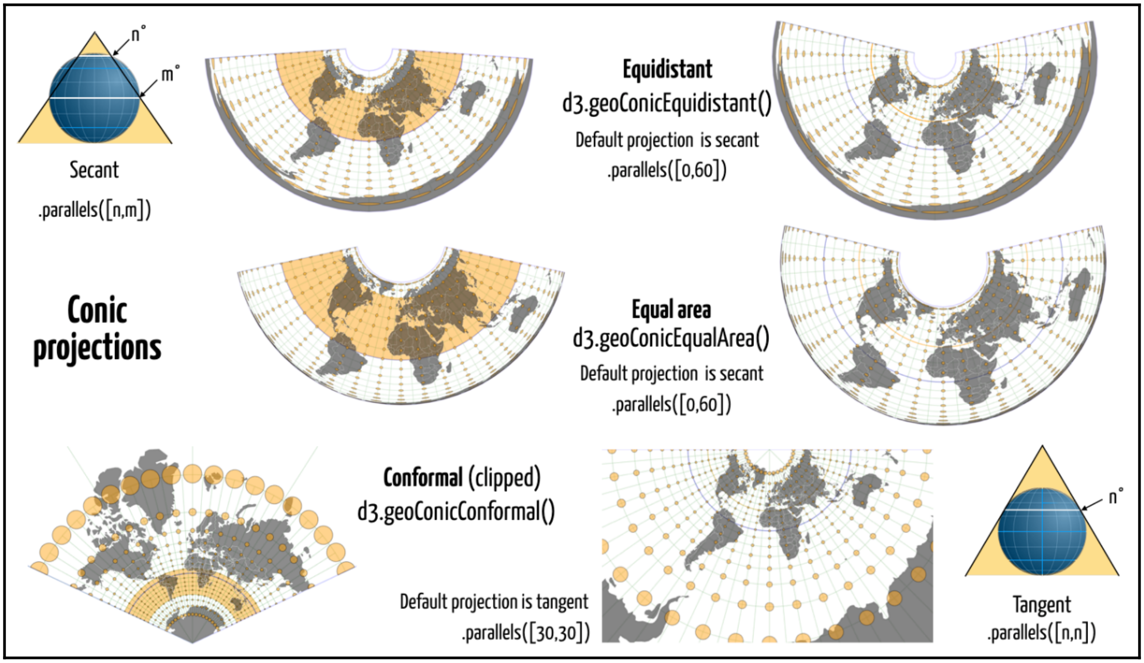


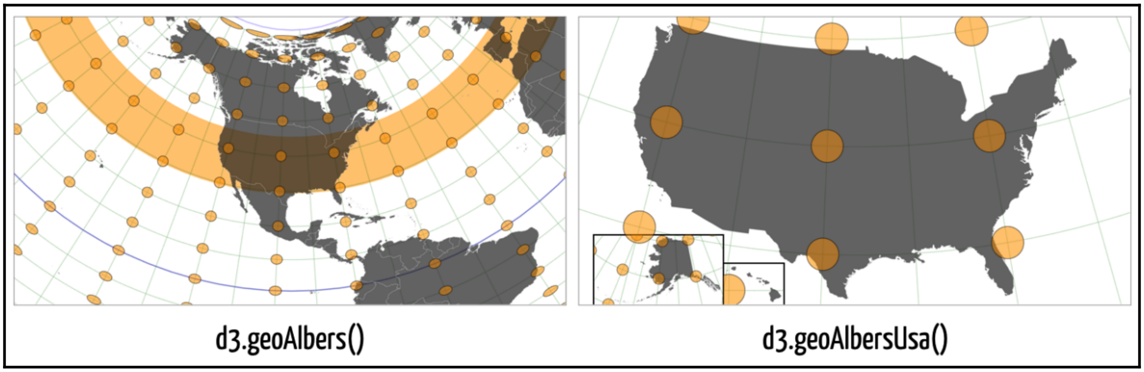

Azimuthal projection
Cylindrical projection
standard
parallels
Conic projection

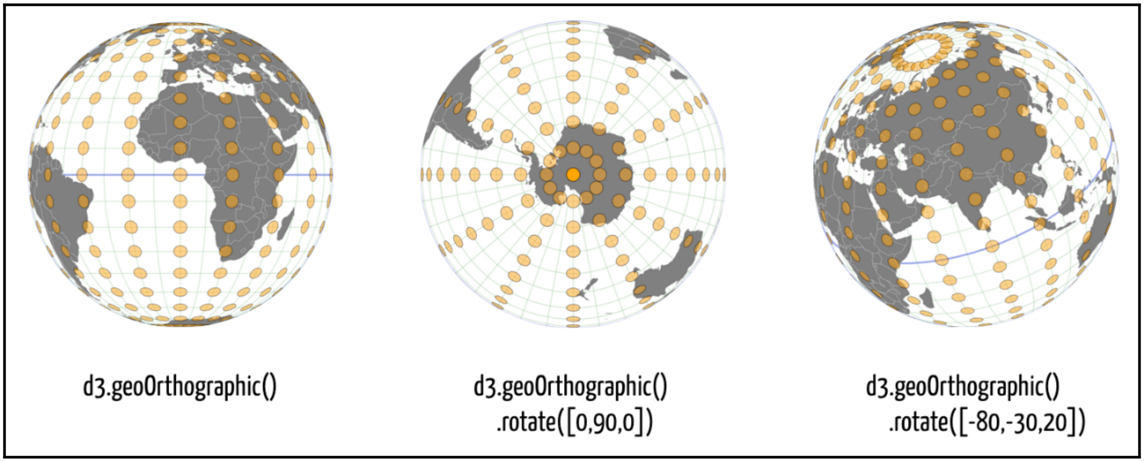

LAX
True direction
Great circle
(shortest path)
44°
62°
116°
LHR
Constant direction
77°
77°
77°
Rhumb line
(constant bearing path)
Gnomonic azimuthal projection
62°
Great circle
(shortest path)
116°
44°
LHR
77°
Rhumb line
(constant bearing path)
True direction
Constant direction
LAX
Mercator cylindrical projection
Great circle
LHR
LAX
Rhumb line
Orthographic projection

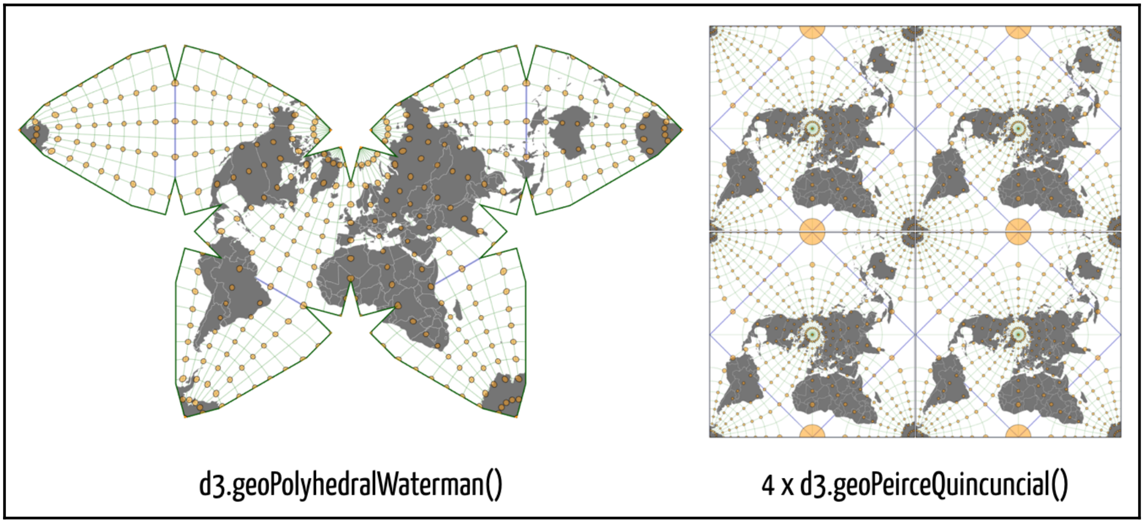

Azimuthal projections
Polar
.rotate([0,90,0])
Equidistant
d3.geoAzimuthalEquidistant()
Equal area
d3.geoAzimuthalEqualArea()
Conformal (clipped)
d3.geoStereographic()
Equatorial
(default)

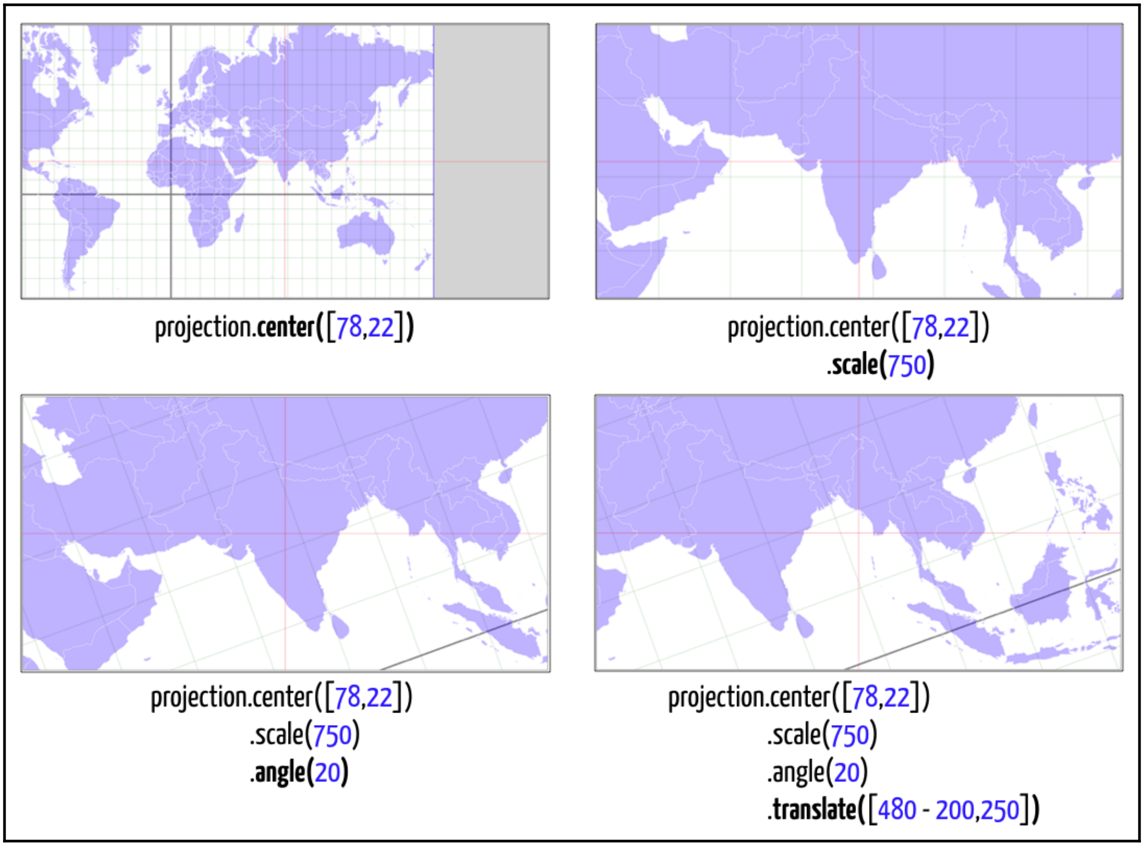

Cylindrical projections
Plate carrée
Equidistant
d3.geoEquirectangular()
Cassini
Polar projection
.rotate([0,90,0])
Tangent
Equatorial
projection
Conformal (clipped)
d3.geoMercator()
Transverse
Mercator
.parallel(0)
Gall Peters 45°
Equal area
d3.geoCylindricalEqualArea()
Secant projection:
(default for
equal area 38.58°)

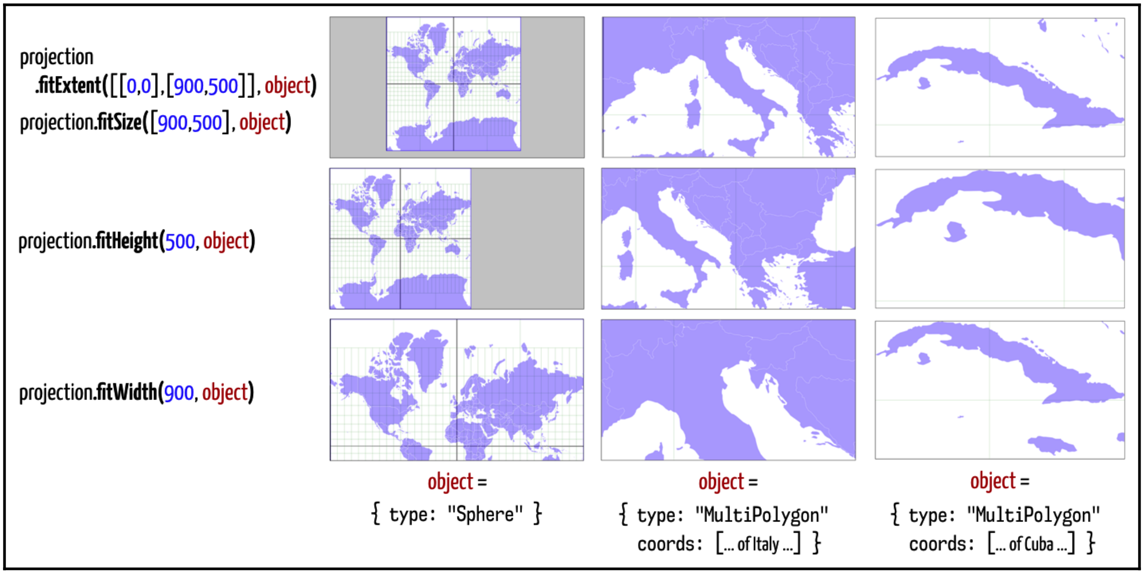

n°
m°
Secant
.parallels([n,m])
Equidistant
d3.geoConicEquidistant()
Default projection is secant
.parallels([0,60])
Conic projections
Equal area
d3.geoConicEqualArea()
Default projection is secant
.parallels([0,60])
Conformal (clipped)
d3.geoConicConformal()
Default projection is tangent
.parallels([30,30])
n°
Tangent
.parallels([n,n])

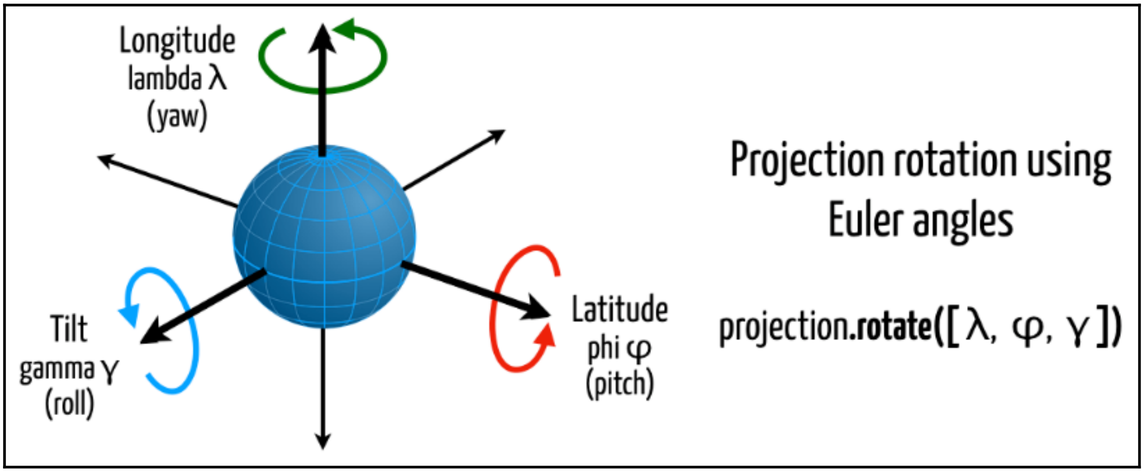

d3.geoAlbers()
d3.geoAlbersUsa()

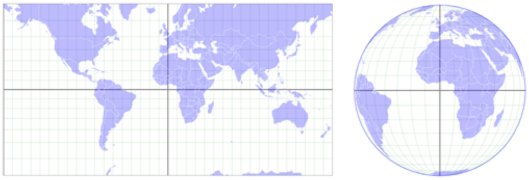

d3.geoOrthographic()
d3.geoOrthographic()
.rotate([0,90,0])
d3.geoOrthographic()
.rotate([-80,-30,20])


d3.geoPolyhedralWaterman()
4 x d3.geoPeirceQuincuncial()

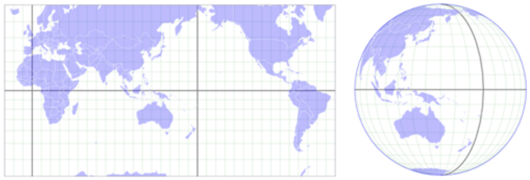

projection.center([78,22])
projection.center([78,22])
.scale(750)
projection.center([78,22])
.scale(750)
.angle(20)
projection.center([78,22])
.scale(750)
.angle(20)
.translate([480 - 200,250])


projection
.fitExtent([[0,0],[900,500]], object)
projection.fitSize([900,500], object)
projection.fitHeight(500, object)
projection.fitWidth(900, object)
object =
{ type: "Sphere" }
object =
{ type: "MultiPolygon"
coords: [... of Italy ...] }
object =
{ type: "MultiPolygon"
coords: [... of Cuba ...] }


Longitude
lambda λ
(yaw)
Tilt
gamma γ
(roll)
Latitude
phi φ
(pitch)
Projection rotation using
Euler angles
projection.rotate([λ, φ, γ])



No rotation: projection.**rotate(**[0,0,0]**)**

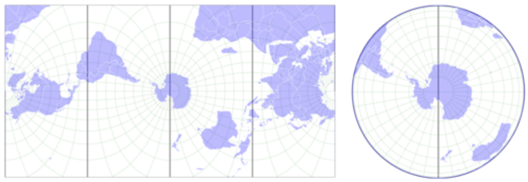


Yaw: projection.**rotate(**[-150,0,0]**)**

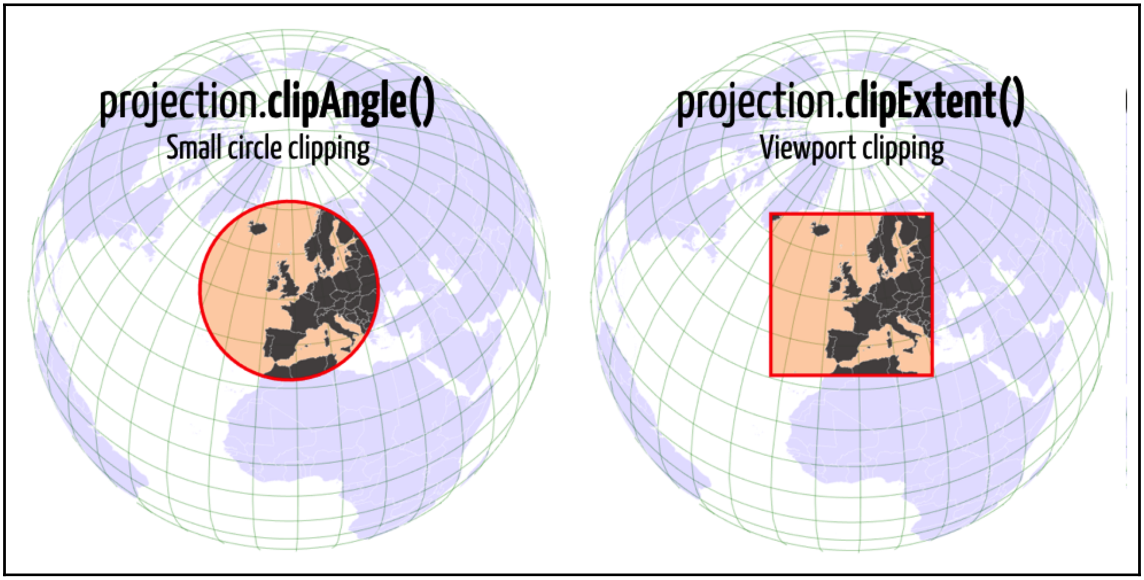
Pitch: projection.**rotate(**[0,-30,0]**)**

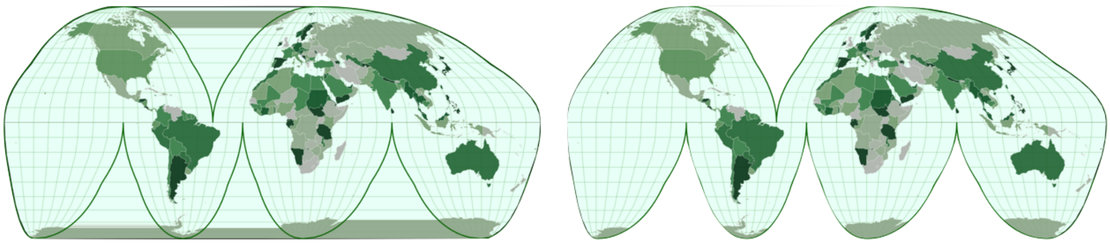
Roll: projection.**rotate(**[0,0,-30]**)**

Combined: projection.**rotate(**[-150,-30,-30]**)**



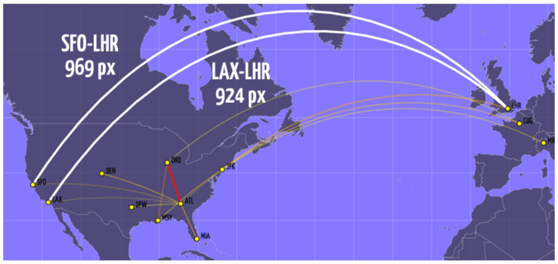
South pole: projection.**rotate(**[0,0,-30]**)**





Projected map with no clipping

With SVG clipping applied to the map



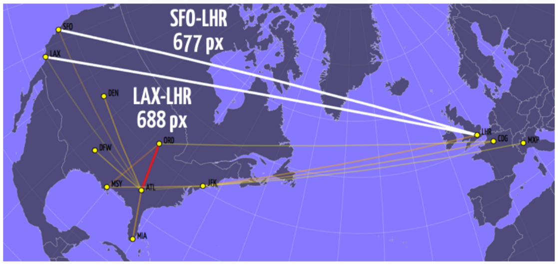
In the **Mercator conformal projection** the projected distance from London (LHR) to San Francisco (SFO) is **longer** than to Los Angeles (LAX)

geoPath.**measure**(d)
returns the projected length of the line in pixels

LAX-LHR: 8,769 km
or 1.3749 radians

>

SFO-LHR: 8,626 km
or 1.3524 radians



d3.geoLength(d) or
d3.geoDistance(p1,p2)
returns the circular length of the arc in radians

In the **Azimuthal equidistant projection** the projected distance from London (LHR) to San Francisco (SFO) is **shorter** than London to Los Angeles (LAX)

In the **Mercator projection** Greenland is 15x larger than the Democratic Republic of the Congo

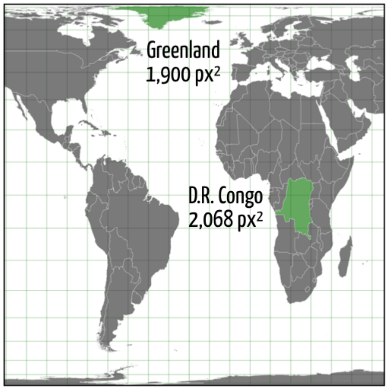
geoPath.**area**(d)
returns **projected** area, in pixels²

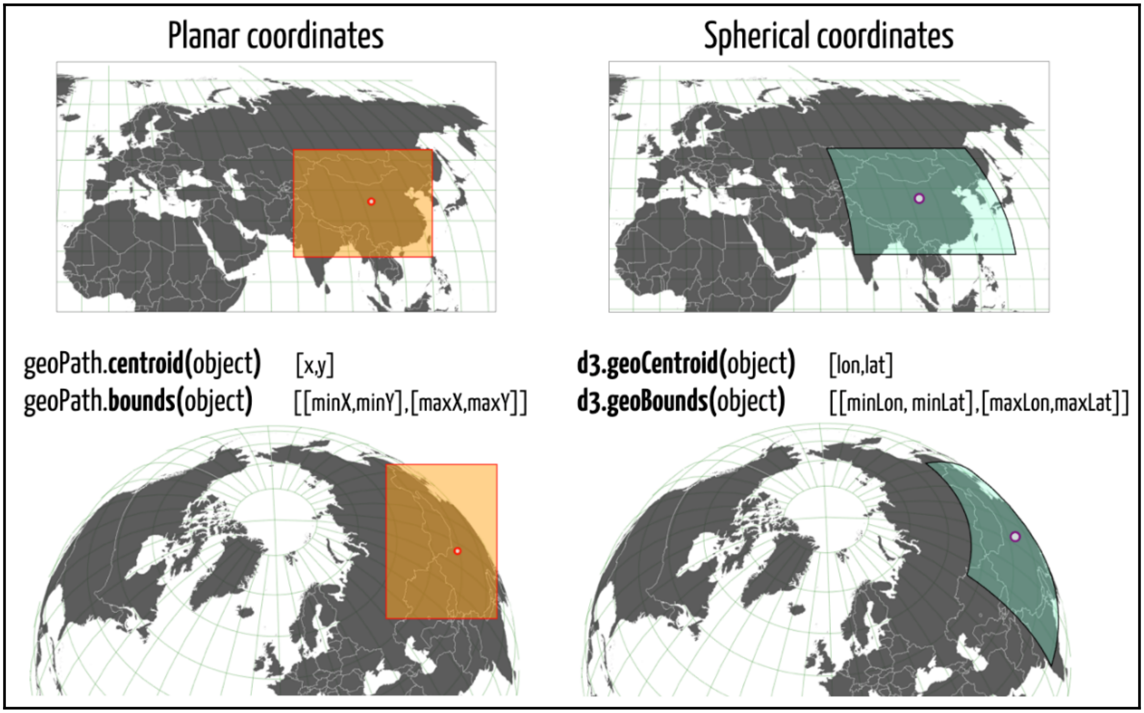
d3.geoArea(d)
returns **spherical** area in radians² (steradians)

D.R. Congo: 2,335,484 km²
or 0.0575 steradians

Greenland: 2,146,343 km²
or 0.0529 steradians

In the **cylindrical equal area projection** Greenland appears smaller than the Democratic Republic of the Congo




Planar coordinates
Spherical coordinates
geoPath.centroid(object)
[x,y]
geoPath.bounds(object)
[[minX,minY],[maxX,maxY]]
d3.geoCentroid(object)
[lon,lat]
d3.geoBounds(object)
[[minLon, minLat],[maxLon,maxLat]]