
Globant
we are ready

UI Engineering Studio. Day 9

Bootcamp: DATA VISUALIZATION

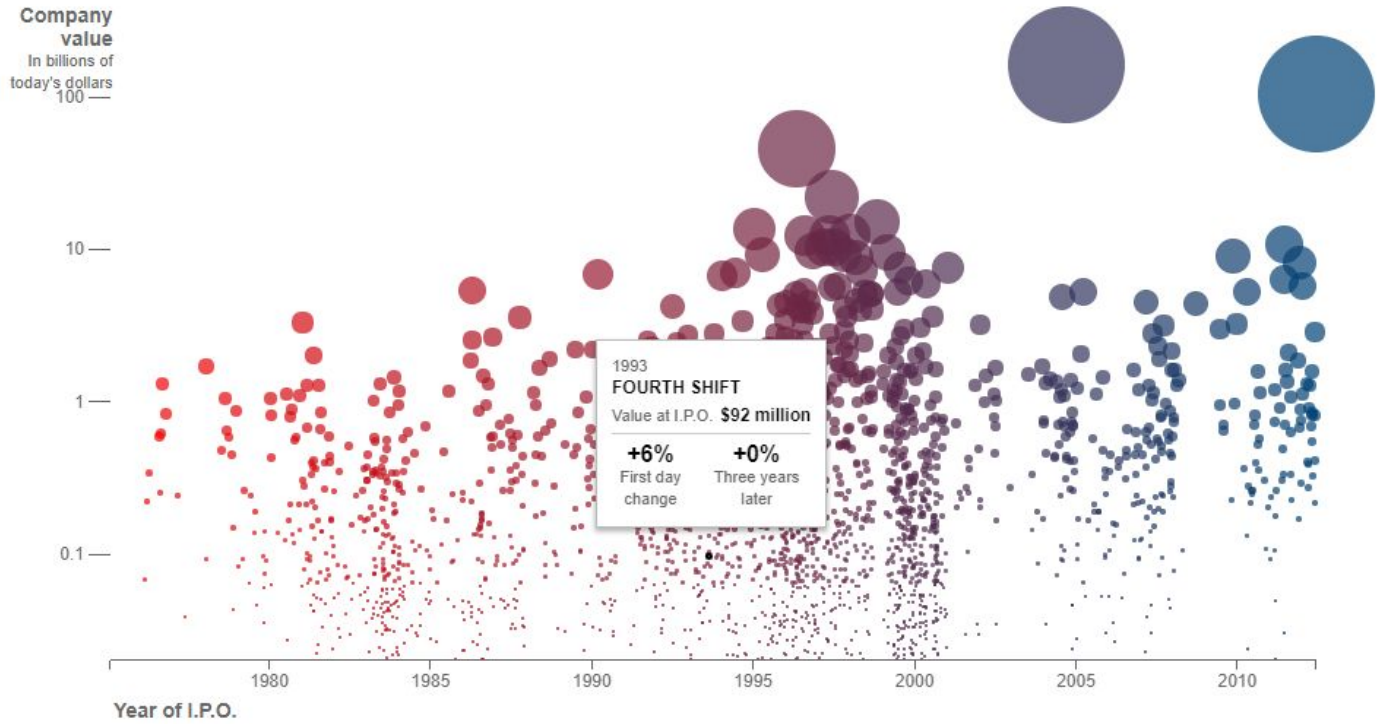
Globant

Data Visualization

Is viewed by many disciplines as a modern equivalent of visual communication

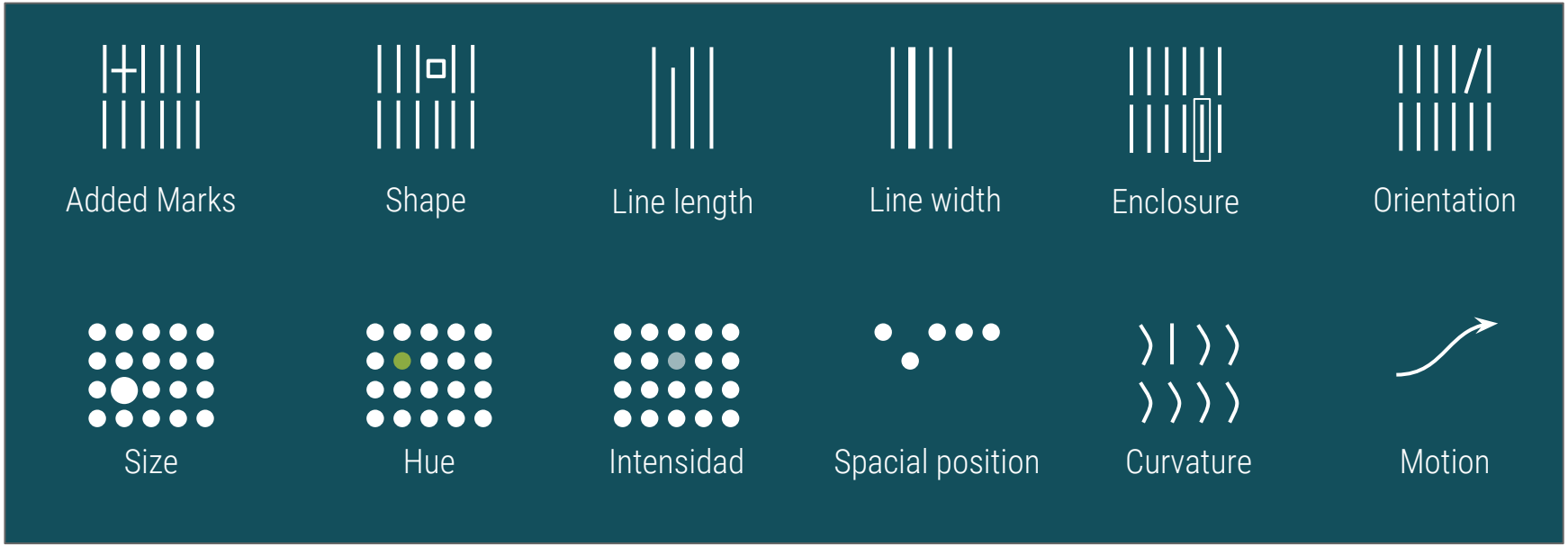
- Visual Encoding
- Intro to d3
- Data manipulation
- Social responsibility

UI Boot Camp: Visualization

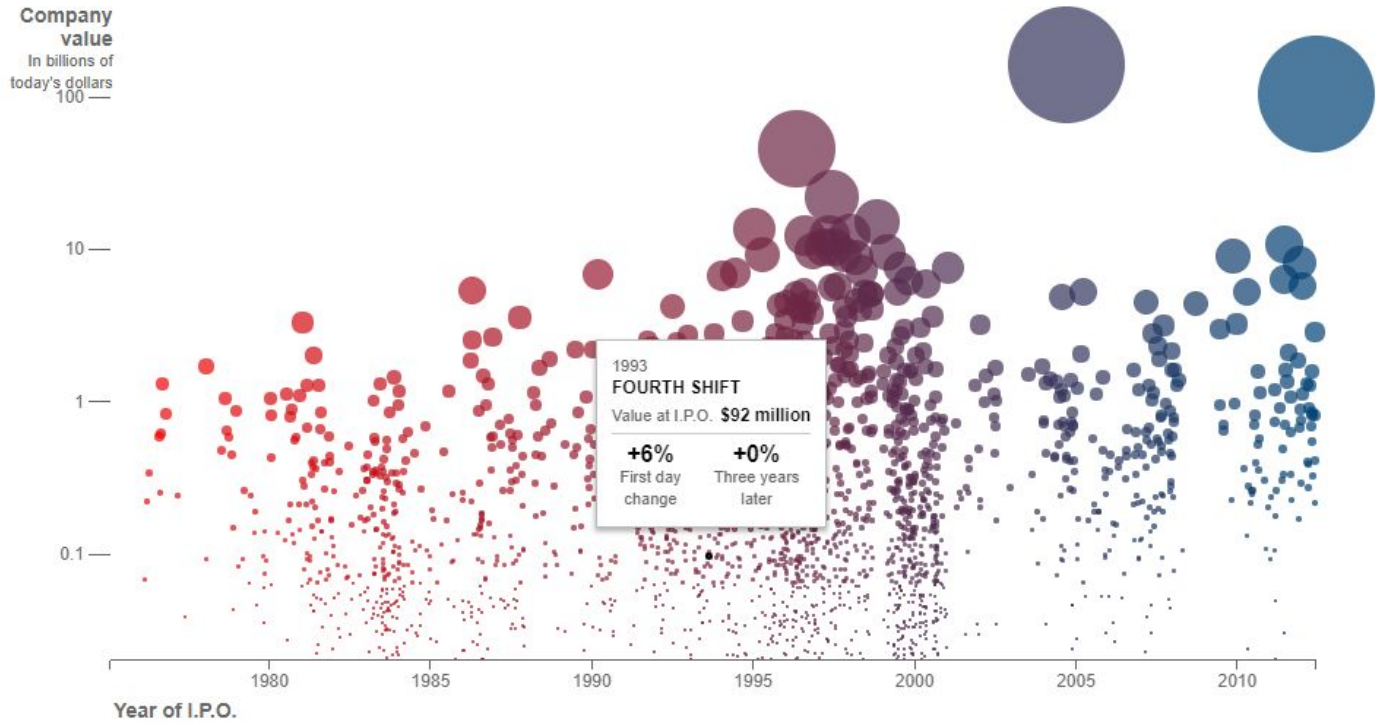


UI Boot Camp: Visualization

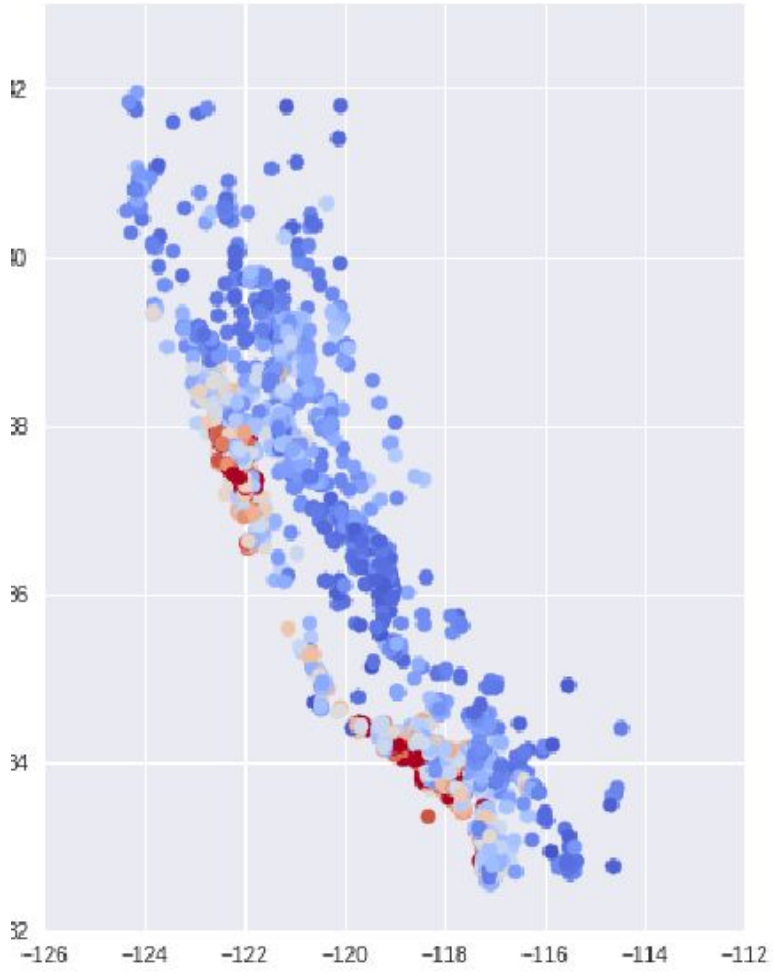
UI Boot Camp: Visualization



UI Boot Camp: Visualization



UI Boot Camp: Visualization



UI Boot Camp: Visualization

D3.js

D3.js is a JavaScript library for manipulating documents based on data

UI Boot Camp: Visualization

```
// Scales
const x = scaleLinear()
    .domain([270, 134000])
    .range([0, width])
```

UI Boot Camp: Visualization

```
// Scales
const x = scaleLinear()
    .domain([270, 134000])
    .range([0, width])

const y = scaleLinear()
    .domain([15, 85])
    .range([height, 0])

const r = scaleLinear()
    .domain([0, 99400000])
    .range([0, 10])
```

UI Boot Camp: Visualization

```
const r = scaleLinear()
    .domain([0, d3.max(countries, d ⇒ d.population)])
    .range([0, 10])
```

UI Boot Camp: Visualization

```
const r = scaleLinear()
    .domain([0, d3.max(countries, d ⇒ d.population)])
    .range([0, 10])
```

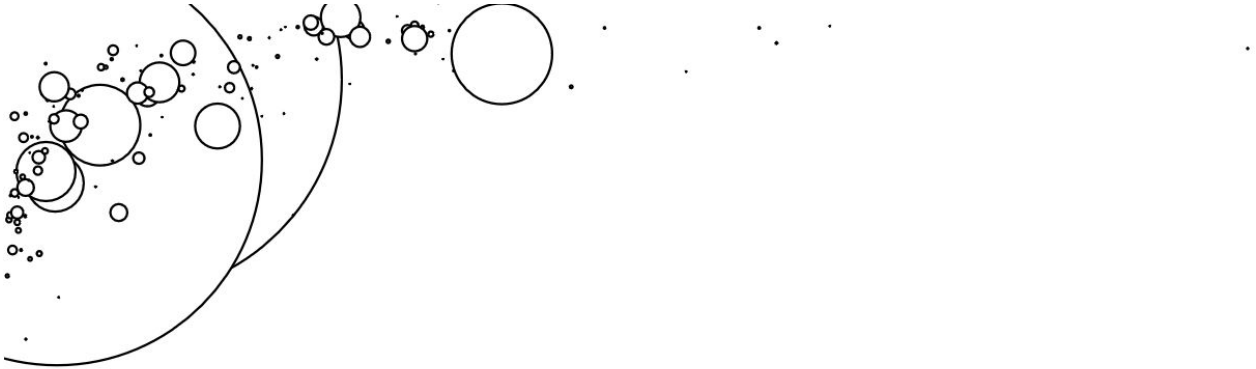
UI Boot Camp: Visualization

```
// Chart
const svg = d3.select('.chart')
svg.attr('width', width + xOffSet + 100)
    .attr('height', height + 2 * yOffSet)

const chart = svg.append('svg')
chart.attr('x', xOffSet).attr('y', yOffSet)
```

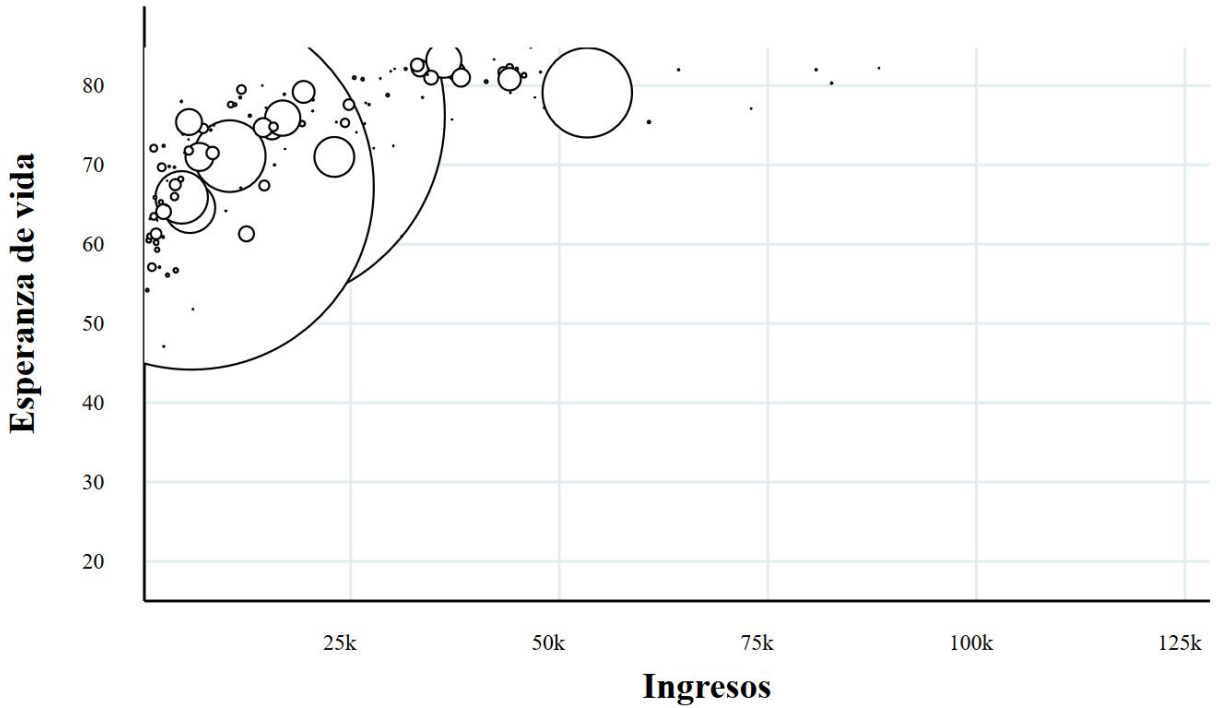
UI Boot Camp: Visualization

```
const circle = chart.selectAll('g')
        .data(countries)
    .enter().append('g')

circle.append('circle')
    .attr('class', d => `country
        country--continent-${toDashCase(d.continent)}
    )
    .attr('r', d => r(d.population))
    .attr('cx', d => x(d.income))
    .attr('cy', d => y(d.lifeExpectancy))
```

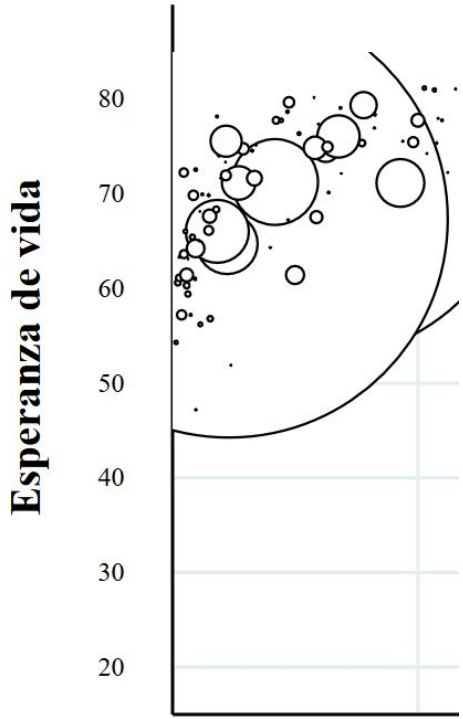
UI Boot Camp: Visualization



UI Boot Camp: Visualization



UI Boot Camp: Visualization




```
const labels = [20, 30, 40, 50, 60, 70, 80, 90]

const lifeExpLabels = svg.append('svg')
lifeExpLabels.attr('y', yOffSet)

const label = lifeExpLabels.selectAll('g')
        .data(labels)
    .enter().append('g')

label.append('text')
    .text(_.identity)
    .attr('class', 'axe-label--life-expectancy')
    .attr('x', xOffSet)
    .attr('y', d ⇒ y(d))
    .attr('dy', 5)
    .attr('dx', -45)

label.append('line')
    .attr('stroke', '#e3eef0')
    .attr('stroke-width', 2)
    .attr('x1', d ⇒ x(initialIncome) + xOffSet)
    .attr('y1', d ⇒ y(d))
    .attr('x2', d ⇒ x(128000) + xOffSet)
    .attr('y2', d ⇒ y(d))
```

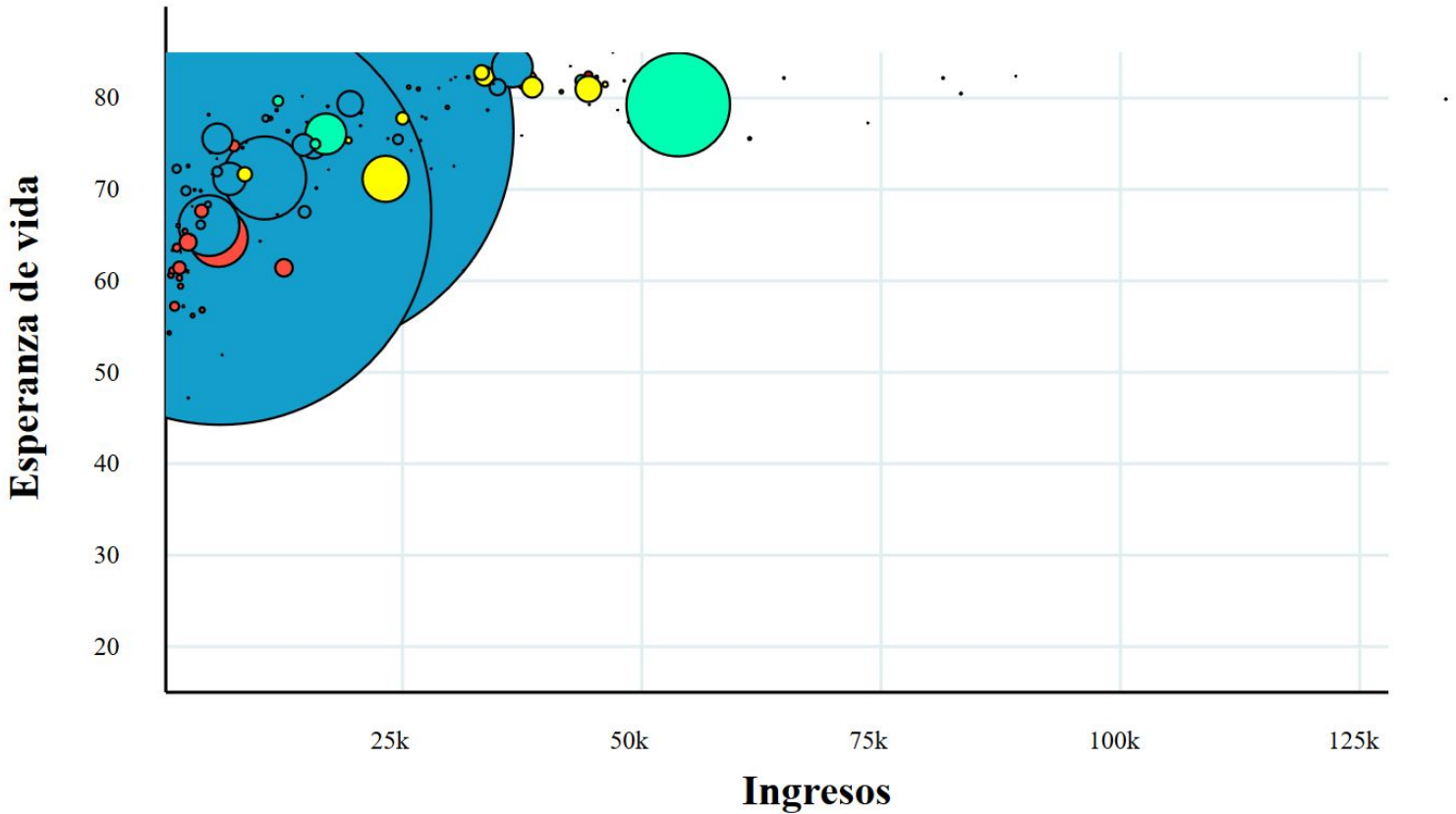
UI Boot Camp: Visualization

Using css on svg

```
svg.append('text')
    .text('Esperanza de vida')
    .attr('class', 'axe-title axe-title--life-expecta
    .attr('transform', `translate(${x(initialIncome)
        ${y(43) + yOffSet}) rotate(-90)`)
    .attr('dy', -80)
    .attr('dx', -40)
```

```
.country {
    stroke: black;
    stroke-width: 1.5;
    fill: white;

    &--continent {
        &-asia {
            fill: #149ECC;
        }

        &-australia {
            fill: blue;
        }

        &-south-america,
        &-north-america {
            fill: #00FFB3;
        }

        &-africa {
            fill: #FF4D40;
        }
```

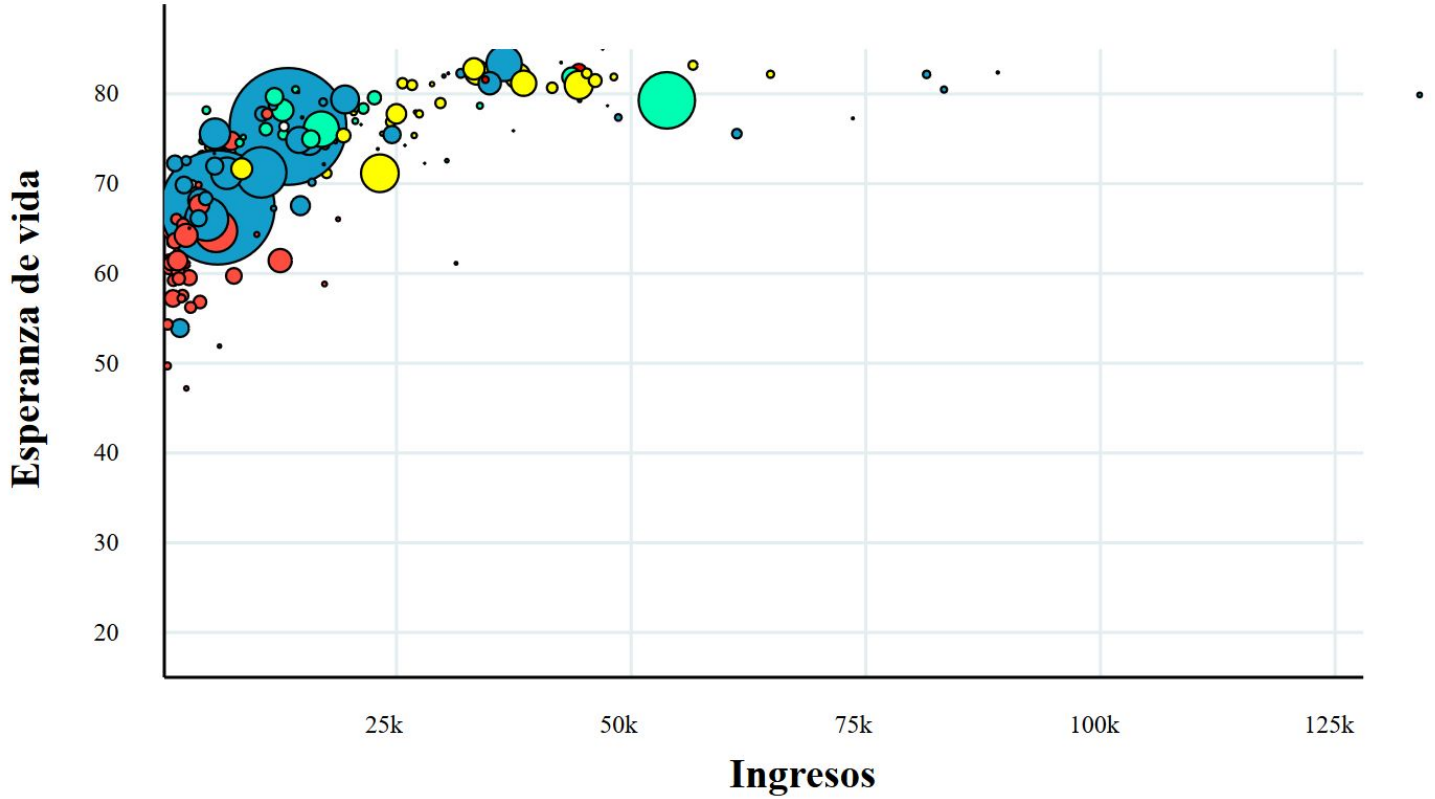
UI Boot Camp: Visualization



UI Boot Camp: Visualization

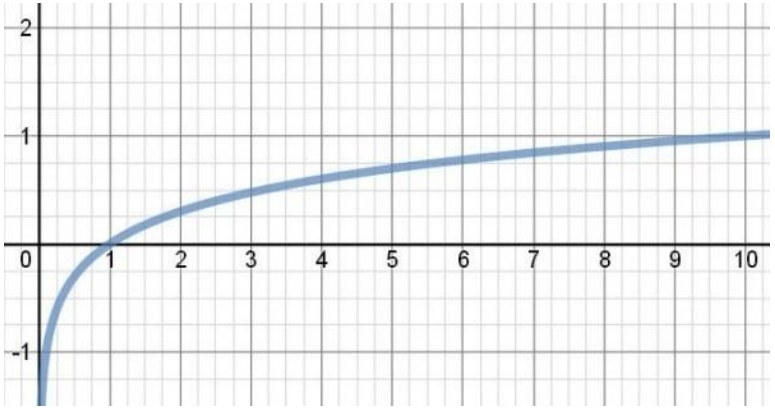
```
// Scales
const x = scaleLinear()
    .domain([initialIncome, d3.max(countries, d ⇒ d.in
    .range([0, width])

const y = scaleLinear()
    .domain([initialLifeExpectancy, d3.max(countries, d
    .range([height, 0])

const r = scaleSqrt()
    .domain([0, d3.max(countries, d ⇒ d.population)])
    .range([0, 10])
```

UI Boot Camp: Visualization

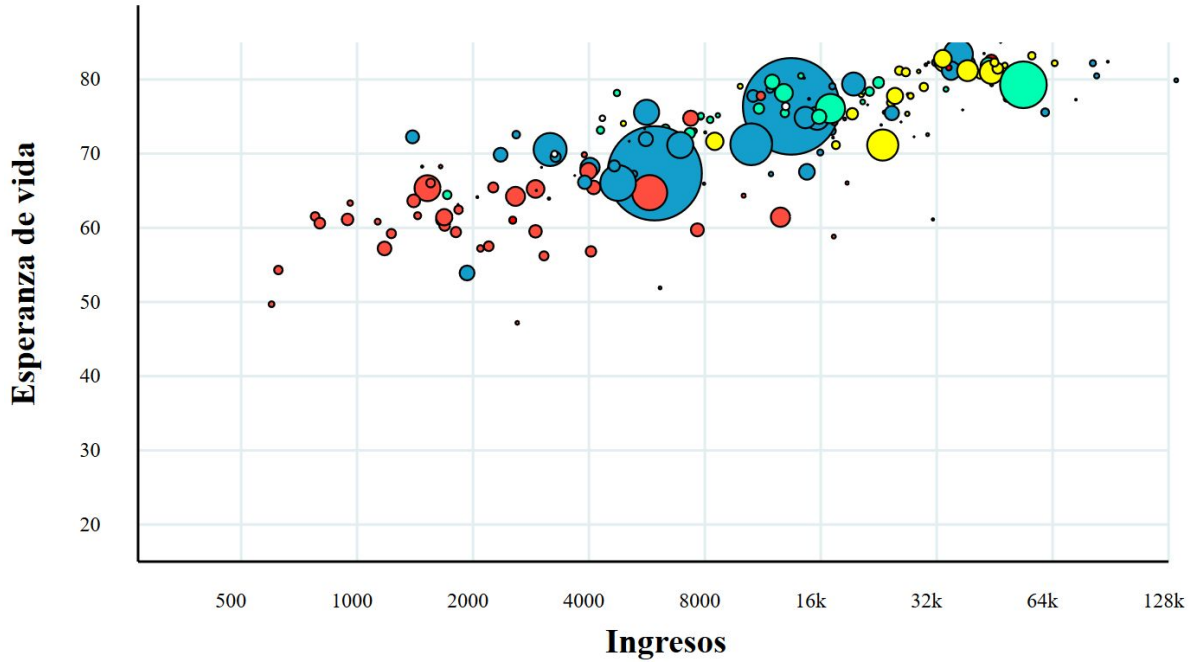


UI Boot Camp: Visualization

Logarithmic scale



UI Boot Camp: Visualization



UI Boot Camp: Visualization

```
// Scales
const x = scaleLog()
    .domain([initialIncome, d3.max(countries, d ⇒ d.in
    .range([0, width])

const y = scaleLinear()
    .domain([initialLifeExpectancy, d3.max(countries, d
    .range([height, 0])

const r = scaleSqrt()
    .domain([0, d3.max(countries, d ⇒ d.population)])
    .range([0, 10])
```

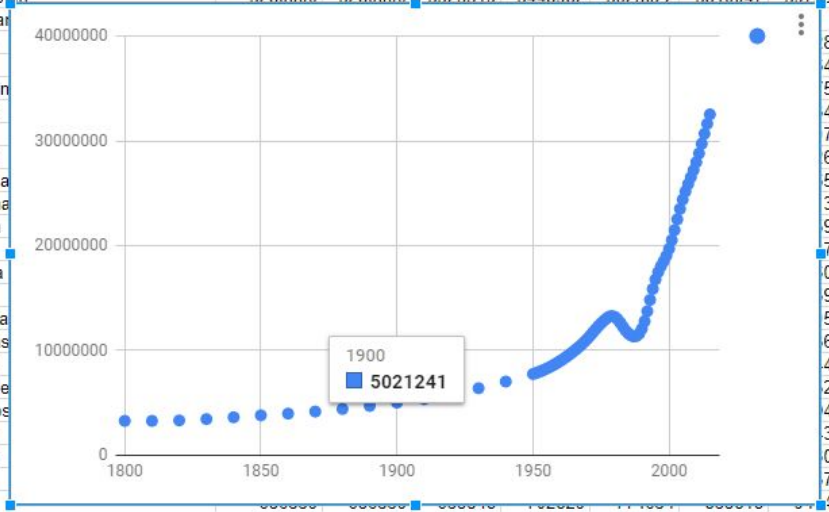
UI Boot Camp: Visualization

Data manipulation


```
[ …
  43:
    continent: "South America"
    country: "Chile"
    income: 0.01
    lifeExpectancy: "79.6"
    population: 0.01
  44:
    continent: "Asia"
    country: "China"
    income: 0.01
    lifeExpectancy: "76.5"
    population: 0.01
  47:
    continent: "South America"
    country: "Colombia"
    income: 0.01
    lifeExpectancy: "78.2"
    population: 0.01
  52:
    continent: "North America"
    country: "Cuba"
    income: 0.01
    lifeExpectancy: "78.3"
    population: 0.01
… ]
```

UI Boot Camp: Visualization

	Total population	1800	1810	1820	1830	1840	1850	1860	1870
2	Abkhazia	3280000	3280000	3323519	3448982	3625022	3810047	3973968	4169690
3	Afghanistan	3280000	3280000	3323519	3448982	3625022	3810047	3973968	4169690
4	Akrotiri ar								
5	Albania							800	610036
6	Algeria							468	3811028
7	American							564	7057
8	Andorra							436	3654
9	Angola							747	2285417
10	Anguilla							693	2888
11	Antigua a							532	35546
12	Argentina							333	1856886
13	Armenia							928	652450
14	Aruba							786	25063
15	Australia							048	1724213
16	Austria							926	4556658
17	Azerbaija							556	1391460
18	Bahamas							625	38787
19	Bahrain							469	65422
20	Banglade							237	24757994
21	Barbados							452	161206
22	Belarus							323	3733161
23	Belgium							088	5110891
24	Belize							755	25745
25	Benin							469	1038118



UI Boot Camp: Visualization

```
st getData = Promise.all([income, lifeExpectancy,
 .then(_.flow([
     cleanData,
     buildData,
     formatData
 ]))
```

UI Boot Camp: Visualization

```
const cleanData = ([income, lifeExpectancy, population]) => {
    // Remove unmatched countries
    let countries = _.flow([
        _.map(_.lowerCase),
        _.filter(x => !!x),
        _.groupBy(_.identity),
        _.filter(arr => arr.length === 3),
        _.keyBy(arr => arr[0]),
        _.mapValues(arr => arr[0])
    ])([
        ..._.map('country', income),
        ..._.map('country', lifeExpectancy),
        ..._.map('country', population)
    ])

    const selectedCountry = x => !!countries[x.country.toLowerCase()]
    income = income.filter(selectedCountry)
    lifeExpectancy = lifeExpectancy.filter(selectedCountry)
    population = population.filter(selectedCountry)

    return [income, lifeExpectancy, population]
}
```

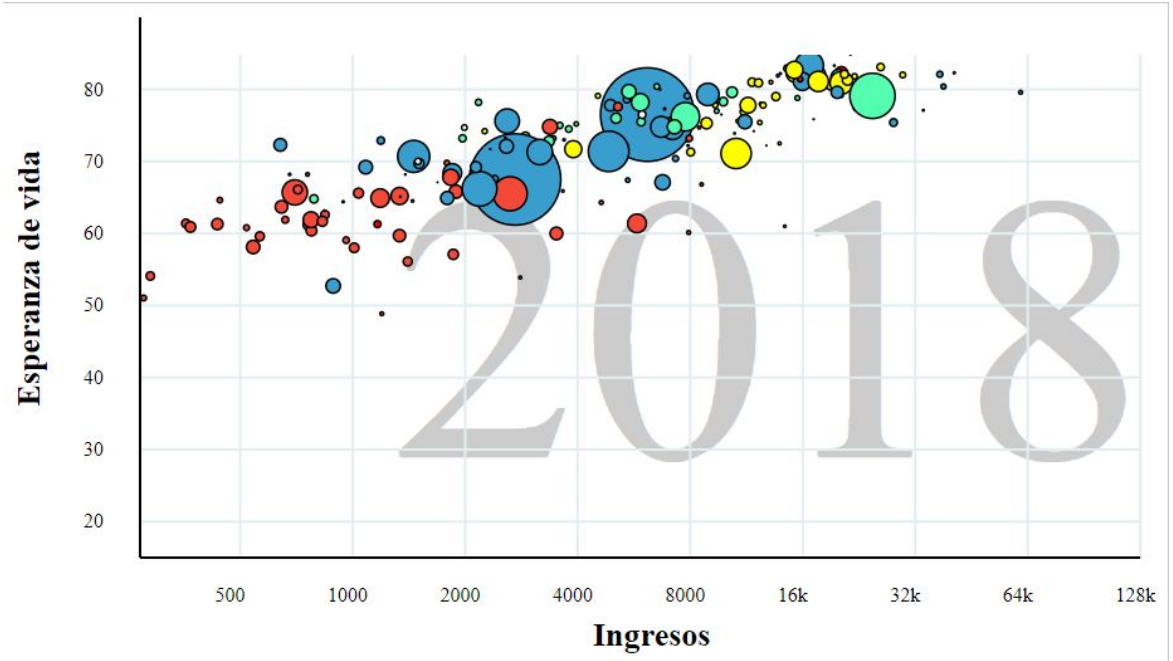
UI Boot Camp: Visualization

```
const buildData = ([income, lifeExpectancy, population]) => {

    const data = {}
    const readValues = (rows, key) => {
        rows.forEach(row => {
            for (let year in row) {
                const country = row.country
                let continent = 'unknown'
                if (countryContients[country.toLowerCase( any {
                    continent = countryContients[country.toLowerCase()]
                } else {
                    // console.warn(country.toLowerCase())
                }
                if (year !== 'country') {
                    const value = row[year]
                    data[year] = data[year] || {}
                    data[year][country] = data[year][country] || { country }
                    data[year][country]['continent'] = continent
                    data[year][country][key] = value
                }
            }
        })
    }

    // Put each country income, life expectancy and population
    readValues(income, 'income')
    readValues(lifeExpectancy, 'lifeExpectancy')
    readValues(population, 'population')

    return _.mapValues(_.toArray, data)
}
```

UI Boot Camp: Visualization

```
const formatData = (countries) => {
    for(let year in countries) {
        for(let country in countries[year]) {
            countries[year][country]['population'] = _.replace(/,/g, '')(
                countries[year][country]['population']
            )
            countries[year][country]['population'] = countries[year][country]['population'] != '' ?
                countries[year][country]['population'] : 0.01

            countries[year][country]['income'] = _.replace(/,/g, '')(
                countries[year][country]['income']
            )
            countries[year][country]['income'] = countries[year][country]['income'] != '' ?
                parseInt(countries[year][country]['income']) : 0.01
        }
    }
    return countries
}
```

UI Boot Camp: Visualization




Globant
we are ready