Create Thematic maps with **Bertin.js : : CHEAT SHEET**

• Installation :

• In bowser (latest version) :

```
<script src="https://cdn.jsdelivr.net/npm/bertin@1" charset="utf-8"></script>
```

• In Observable :

Latest version :

```
bertin = require("bertin")
```

• In Quarto :

You can use bertin with ojs cells. This allows to combine analyses in R or Python with visualizations made with bertin. An example is available here : https://neocarto.github.io/bertin/examples/quarto.html

• The data with bertin.js

For using your data, the file must be a geojson

• The draw function

For displaying content on the map, We use the function :

bertin.draw

• Gobal parameters

The global parameters of the map are specified in the bt.draw function: its size, projection, background color, etc.

```
bertin.draw({
  params: {
    projection: "Bertin1953",
    width: 750,
  },
  layers: [...]
})
```

Parameters :

- projection
- width
- extent
- margin
- background
- clip
- reverse

• Simple layer

For displaying a simple layer, we use: type: ***layer***.

```
bertin.draw({
  params: {
},
  layers: [
    {
      type: "layer",
      geojson: *a geojson here*,
      fill: "#e6acdf",
    }
  ]
})
```

• Output :

• Merge data

Bertin.js allows to merge data beetween a geojson and a CSV file

With ***bertin.match()*** we can check and vizualize the data Compatibility

With ***bertin.merge()***, the join is made

Example :

```
bertin.merge
(geojson, "ISO3", dataCSV, "id")
```

• Symbology layers :

There are several symbology typein bertin. We can choose it by setting the layer type.

Every symbology take parameters that affect the way the layers are displaying. These parameters are are sometimes specific to each layer type.

Proportional Symbols (stocks)

```
layers: [
  {
    type: "bubble",
    geojson: data,
    values: "pop",
    k: 50,
  },
],
```

type: "square" → Same with square

Choropleth (ratio)

```
layers: [
  {
    type: "layer",
    geojson: data,
    fill: {
      type: "choro",
      values: "gdppc",
      nbreaks: 5,
      method: "quantile",
      colors: "RdYlGn",
    },
],
```

Typology (categories)

```
layers: [
  {
    type: "layer",
    geojson: data,
    fill: {
      type: "typo",
      values: "region",
    },
  ]
```

Typo + prop

```
layers: [
  {
    type: "bubble",
    geojson: data,
    values: "pop",
    fill: {
      type: "typo",
      values: "region",
    },
  },
],
```

Choro + prop

```
layers: [
  {
    type: "bubble",
    geojson: data,
    values: "pop",
    fill: {
      type: "choro",
      method: "quantile",
      nbreaks: 5,
      values: "gdppc",
      pal: "RdYlGn",
    },
  },
],
```

Regular Grid (stocks)

```
layers: [
  {
    type: "regulargrid",
    geojson: countries,
    step:20,
    values: "pop",
    fill:{
      nbreaks: 6,
      method: "quantile",
      colors: "Blues"
    },
  ],
```

The regulargrid type is a way to transform polygon layer into a regular squared grid

Dot cartogram (stocks)

```
layers: [
  {
    type: "dotcartogram",
    geojson: data,
    onedot: 2000,
    iteration: 200,
    values: "gdp",
    radius: 4,
    span: 0.5,
    fill: "red",
  },
],
```

Regular Bubble

```
layers: [
  {
    type: "regularbubble",
    geojson: countries,
    step:20,
    values: "pop",
    k: 50,
  },
],
```

type: "regularsquare" → with square

The regularbubble type is used to draw a map by proportional circles in a regular grid, from absolute quantitative data.

Smooth (stocks)

```
layers: [
{
    type: "smooth",
    geojson: data,
    values: "pop",
    thresholds: 50,
    bandwidth: 25,
    colorcurve: 1,
  },
],
```

The smooth type is a way to produce a continuous representation from quantitative data

To have comlete overview of bertin's symbologies, have a look to the documentation here :https://github.com/neocarto/bertin

Bertin.js : : CHEAT SHEET

• Code explanation (bubble + choro map exemple) :

```
bertin.draw({
 params: { projection: d3.geoPolyhedralWaterman(), clip: true },
 layers: [
  {
   type: "bubble",
   geojson: data,
   k: 50,
   values: "pop",
   leg_x: 800,
   leg_y: 270
   leg_divisor: 1000000,
   leg_title: "Population",
   leg_round: 0,
   fill: {
    type: "choro",
    method: "jenks",
    nbreaks: 5,
    values: "gdppc",
    colors: pal,
    leg_title: "GDP per capita",
    leg_x: 100,
    leg_y: 270,
    leg_round: -2
   },
  },
  {
   type: "layer",
   geojson: data,
   stroke: "none",
   fill: "white",
   fillOpacity: 0.5
  },
  { type: "graticule" },
  { type: "outline" }
 ]
})
```

The ***bertin.draw*** function where all the symbology is set up with in

Define the parameter of the layers written bellow

Layers, define all the layers to be displayed.

The type of symbology is specified

Here the legend parameter of the bubble symbology is specified. They must be write inside the layer parameters bloc

The fill parameter of the bubble layer is used to display ***choro*** symbology in the bubbles.

Here are the others layers which compose the map.

• Order of the layers

The display order of the layers is defined by the writing order in the functions ***bertin.draw().***

You can reverse this logic by using:
params***(reverse : true)***
(default: ***false***)

• Set a base map :

The ***tiles*** layer's type allows you to set a raster background on your map.

Exemple of use :

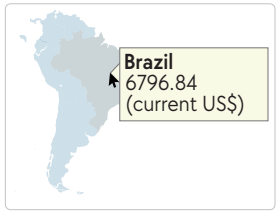
```
layers: [
 {
  type: "tiles",
  style: "worldphysical"
 },
],
```

The function can display several base map :

- openstreetmap
- opentopomap
- worldterrain
- worldimagery
- shadedreliefs
- oceanbasemap

• Tooltip :

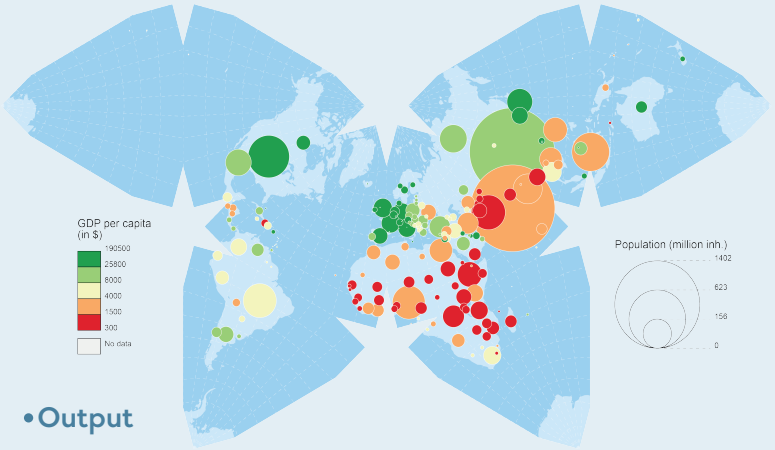
This allows you to display the characteristics of a data item by hovering it with the mouse.



toolip : parameter of a symbology layer

```
tooltip : ["$name", "$gdppc"
, "(current US$)"]
```

Tooltip can contain information from the data or added by hand.
For informations contained in the data, use «$».



• Output

• Legend

In bertin, legend are parameters of the layers. For display the legend of a layer you must use the params :
leg_x :*(value in px)*
leg_y : *(value in px)*
These parameters set up the location of the legend on your map.
For the legend title : ***leg_title :*** *" "*,

There are many other parameters that modify the legend: avalaible here

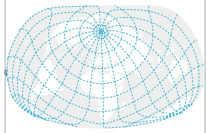
• Layout

There are several types of layers that can be add to the map.

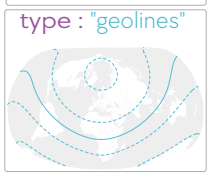
Layer types :

- "scale"
- "graticule
- "logo"
- "shadow"
- "rhumbs"
- "inner"
- "minimap"
- hatch
- geolines

type: "graticule"



type : "geolines"



• Texts

Type: ***text***, ***header***, and ***footer***, are used to display text on the map

```
bertin.draw({
 layers: [
  {
   type: "text",
   text: "This is my text",
   position: "bottomright",
   fontSize: 20,
   frame_stroke: "red",
   margin: 4,
  },
 ],
})
```