

# GADMTTools - Graphics

*jp.decorps@epiconcept.fr - Epiconcept*

*2019-06-26*

---

Epiconcept is made up of a team of doctors, epidemiologists, data scientists and digital specialists. For more than 20 years, Epiconcept has been contributing to the improvement of public health programs by providing software, epidemiological studies, counseling, evaluation and training to better prevent, detect and treat people.

Epiconcept delivers software and services in the following areas :

- Software for managing public health programs,
- Secure cloud solutions for health data collection, reporting and processing,
- The implementation of research projects on measuring the effectiveness and impact of vaccines,
- Services in the field of epidemiology (protocols, analyzes, training, etc.),
- Expertise in data analysis,
- Counseling, coaching and assistance to project owners for public health programs,
- Training (short introductory modules, training through long-term practice).

To achieve such goals Epiconcept :

- Recognized research organization,
- Certified datacenter for hosting personal health data,
- Training organisation.

Epiconcept relies on :

- Its expertise in epidemiology
- Its IT expertise,
- Ethical values rooted in practice (responsibility and quality of services, data security and confidentiality, scientific independence, etc.),
- Capabilities to answer and anticipate tomorrow's challenges (Research - evaluation, e-health, Big Data, IoT, etc.),
- A desire to build long-term relationships with its clients and partners.

Its current customers and partners include some of the greatest names in the world such as : Santé Publique France (and many public health organizations around the world), WHO, eCDC, AFD, MSF, World Bank, etc.

---

# Graphics

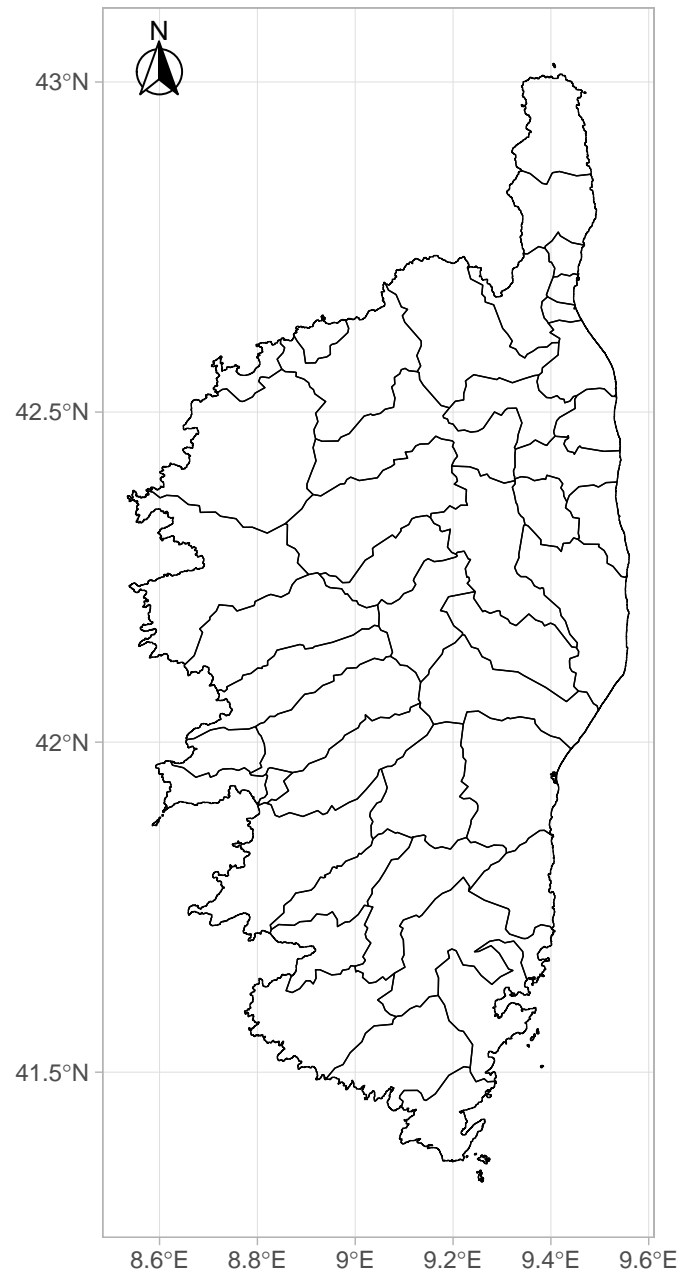
## functions

Function	Description
choropleth	Draws a choropleth on selected regions
classDots	Plots dots on a map with values between different fixed classestries
dots	Plots dots on a map
dotDensity	Draws a dot-density map
isopleth	Draws an isopleth on selected regions
gadm_showNorth	Displays a north arrow on a plot
gadm_showScale	Displays a scale on a plot

## Display a north arrow on a plot

```
library(GADMTTools)
data("Corsica")

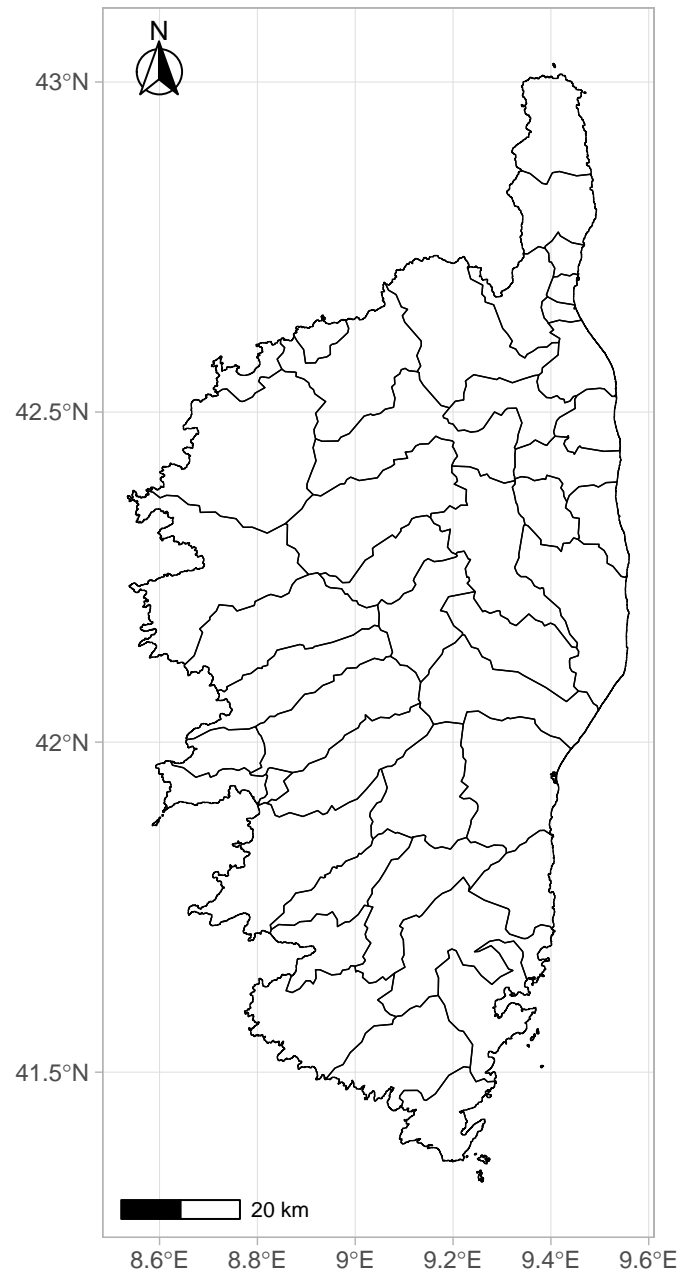
plotmap(Corsica) %>% gadm_showNorth("t1")
```



## Display a scale on a plot

```
library(GADMTTools)
data("Corsica")

plotmap(Corsica) %>% gadm_showNorth("t1") %>% gadm_showScale("b1")
```



## Plotting dots on a map

```
# Preparing data.frame
# -----
data("Corsica")

Corse <- gadm_union(Corsica, 0)
Cantons <- listNames(Corsica, 4)
L <- length(Cantons)
Pop <- floor(runif(L, min=15200, max=23500))

longitude <- runif(6, min=8.74, max = 9.25)
latitude <- runif(6, min=41.7, max = 42.6)
Cases <- floor(runif(6, 25, 80))

Type <- rep(c("TYPE A", "TYPE B", "TYPE C"), 6, length.out = 6)
DAT <- data.frame(longitude, latitude, Cases)
```

```
# Simple dots
```

```
#-----
```

```
dots(Corsica, DAT, color="red", size = 3)
```

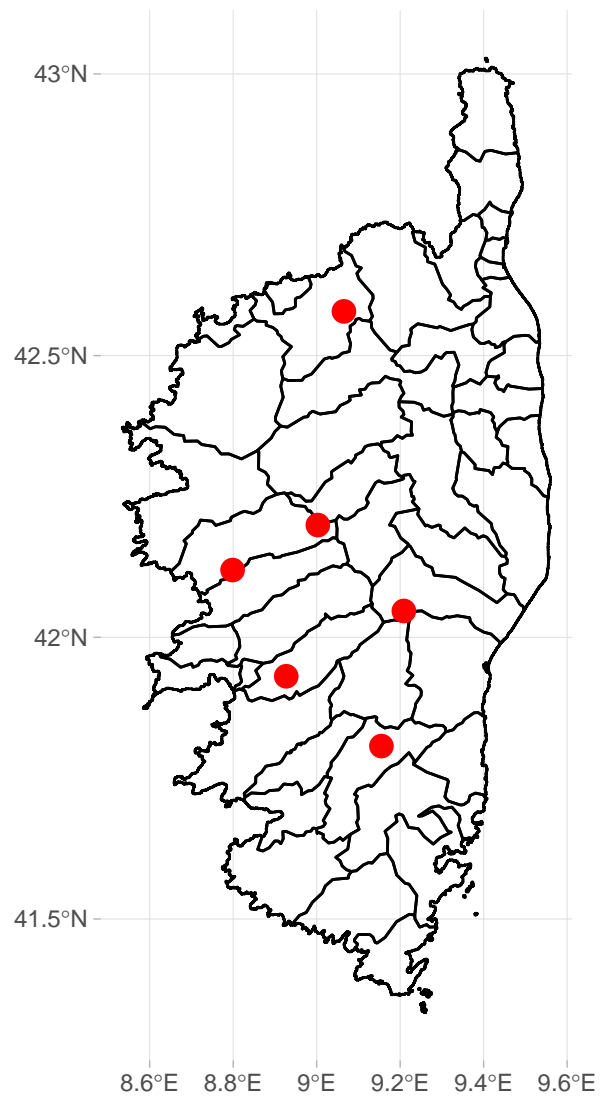


Figure 1: Simple dots

```
# Classified dots
```

```
#-----
```

```
dots(Corse, points = DAT,  
      palette = "Reds",  
      value="Cases")
```

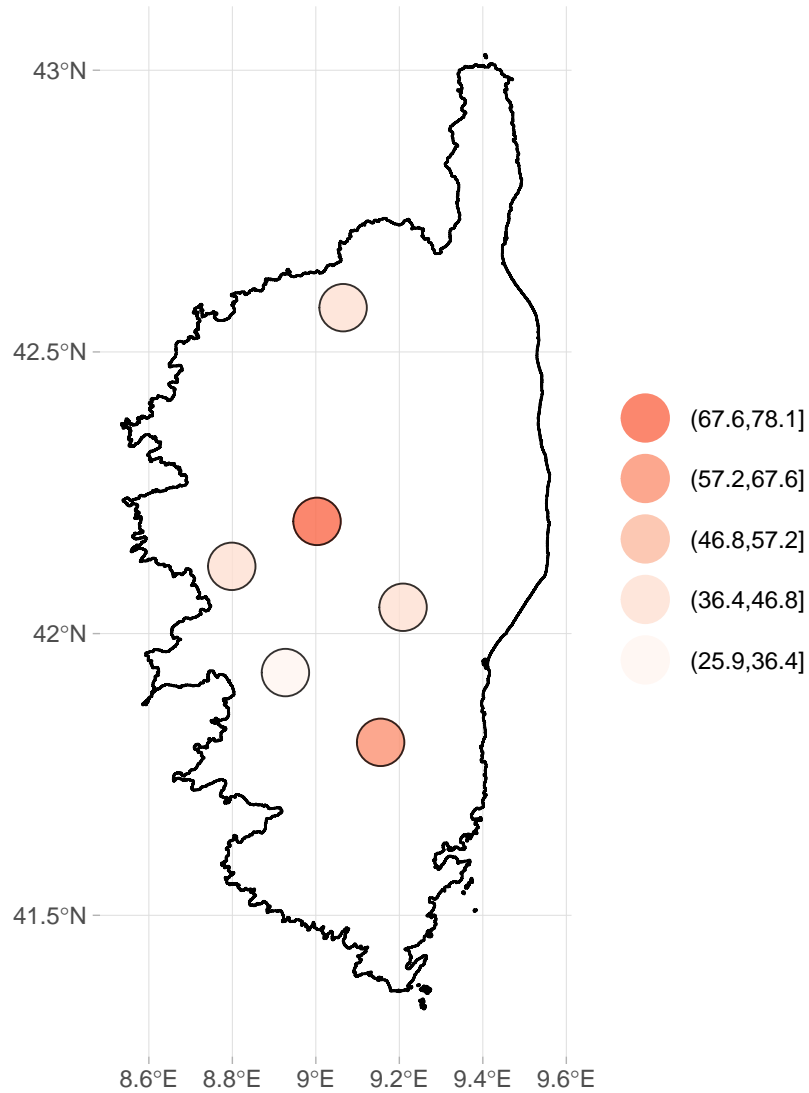


Figure 2: Classified dots

```
# Typed points
```

```
#-----
```

```
DAT2 <- data.frame(longitude, latitude, Type)
```

```
dots(Corse, points = DAT2,  
      color = "#ee00ee",  
      strate="Type")
```

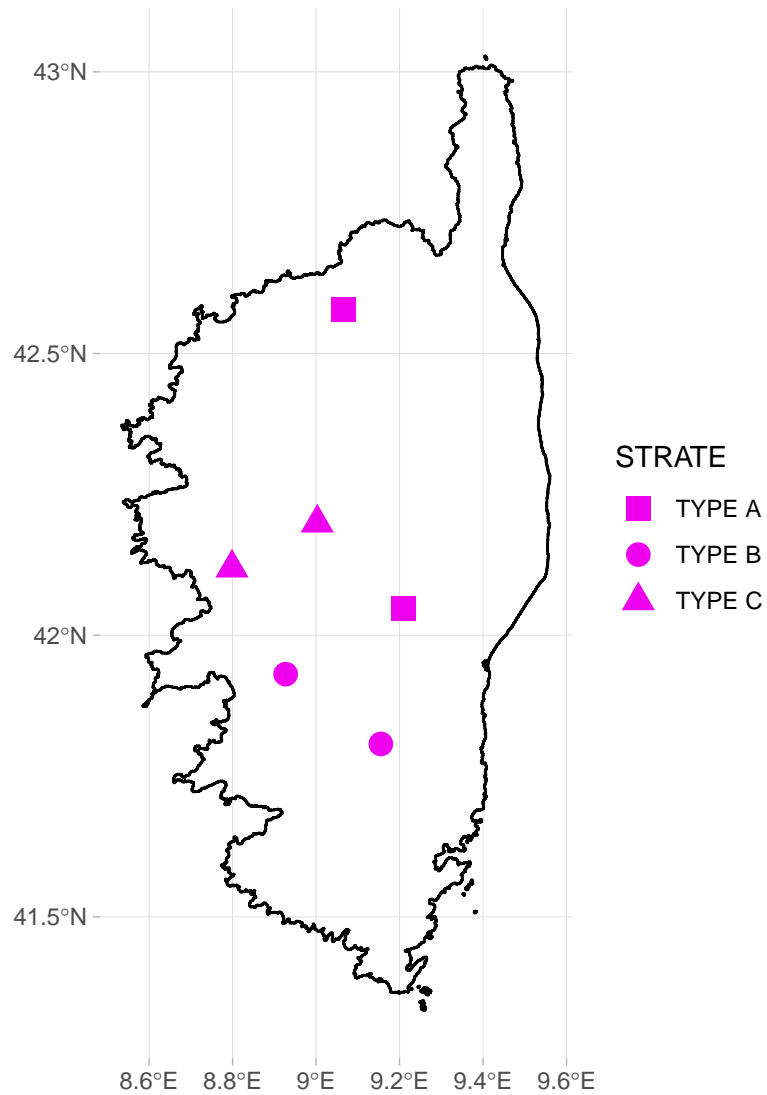


Figure 3: Typed dots



## Plotting proportionals dots

```
# Test of propDots with default parameters  
# -----  
propDots(Corse,  
  data = DAT,  
  value="Cases",  
  color = "blue")
```

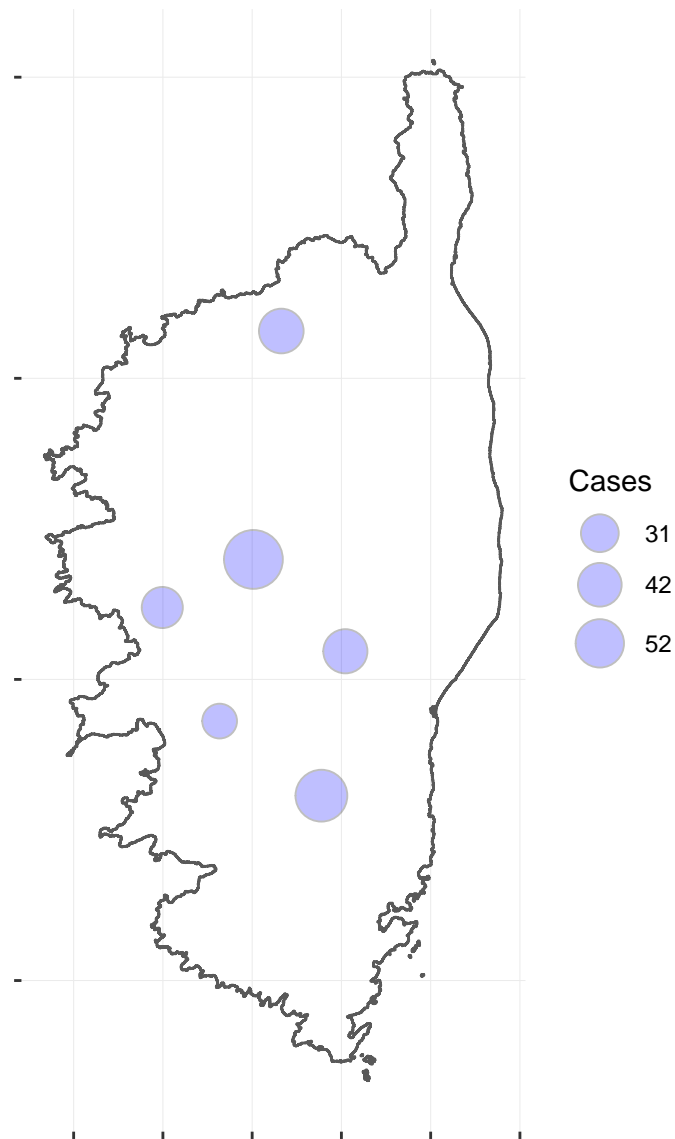


Figure 4: propDots

```
# Test of propDots with defined breaks
```

```
# -----
```

```
propDots(Corse,  
  data = DAT,  
  value="Cases", breaks=c(30, 40, 50, 70, 100),  
  color = "blue")
```

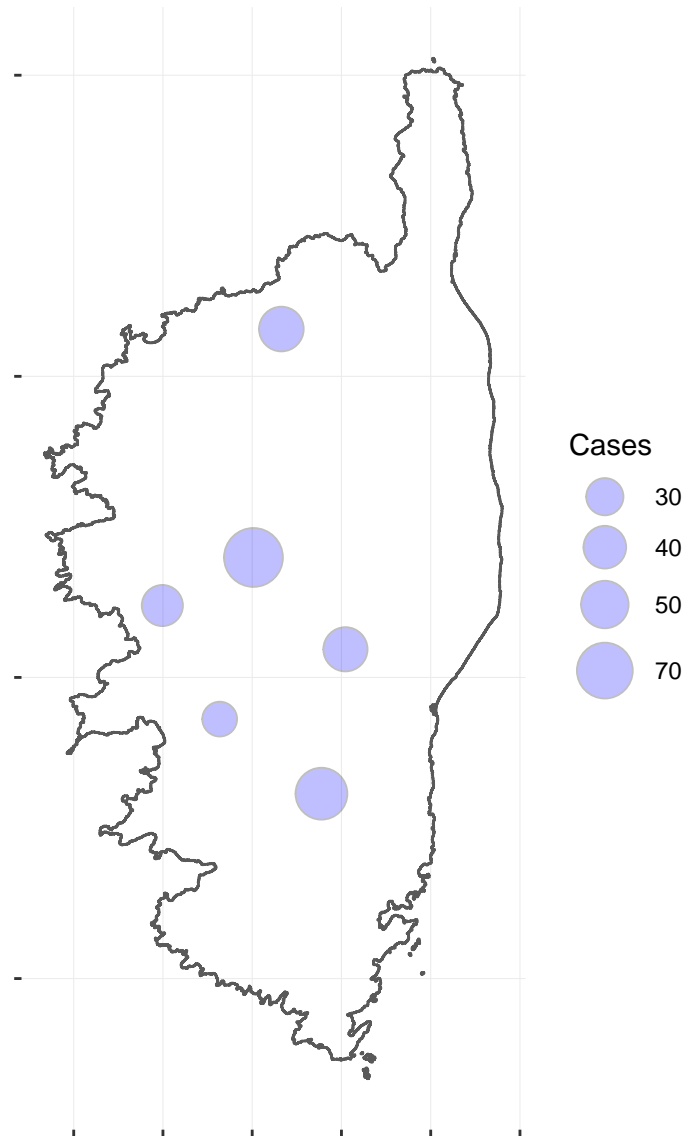


Figure 5: propDots with provided breaks

```
# Test of propDots with forced range of breaks
```

```
# -----
```

```
propDots(Corse, data = DAT, value="Cases",  
         breaks=c(0, 25, 50, 75, 100),  
         range = c(25, 100))
```

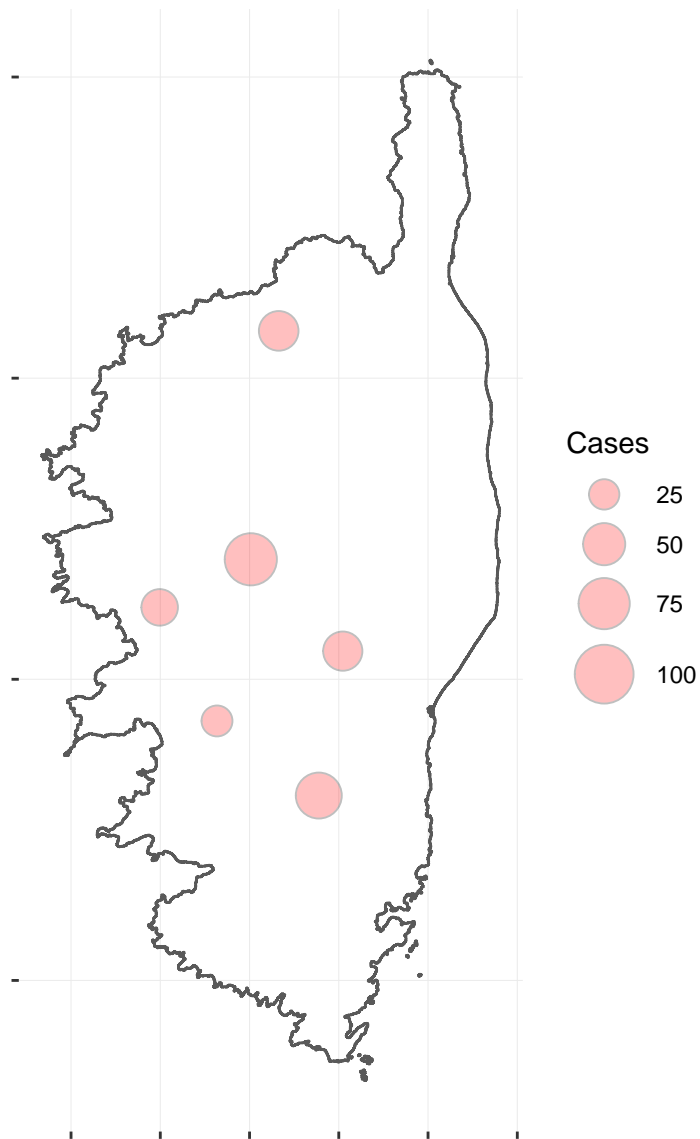


Figure 6: propDots

## Plotting dots with classified size

```
library(GADMTools)

classDots(Corse, DAT, color="blue", value = "Cases", steps = 4)
```

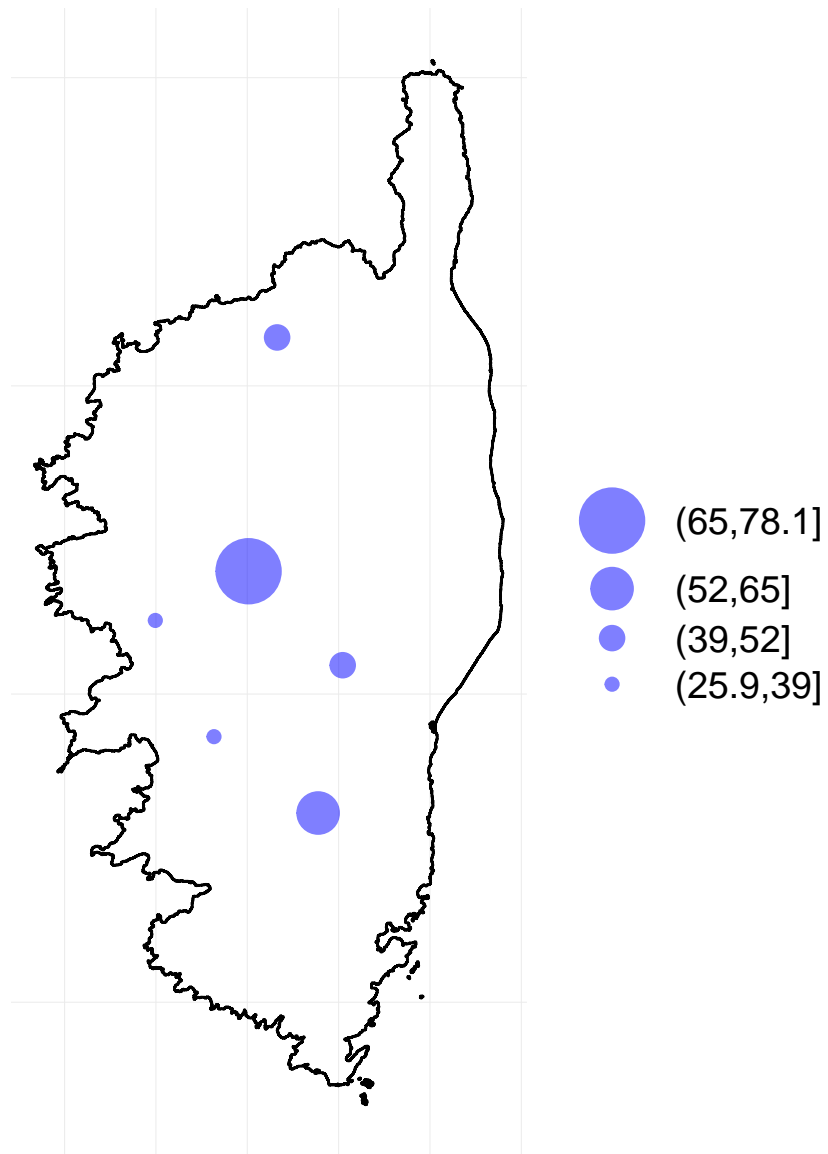


Figure 7: classDots

## Dot-Density

```
library(GADMTools)
data("Corsica")

# Creates test data.frame -----
# -----
VAR_1 <- as.integer(runif(n = 43, min = 800, max = 15800))
VAR_2 <- as.integer(runif(n = 43, min = 1000, max = 15800))
VAR_3 <- as.integer(runif(n = 43, min = 1500, max = 15800))
Cantons <- listNames(Corsica, 4)
DF <- data.frame(Cantons, VAR_1, VAR_2, VAR_3, stringsAsFactors = FALSE)

dotDensity(Corsica,
  DF, adm.join="Cantons", dot.size = 0.5, cases.by.dots = 1000,
  values = c("VAR_1", "VAR_2", "VAR_3"),
  labels = c("H1N1", "H1N2", "H2N2"),
  palette = c("#ffff00", "#ffaa00", "#FF3200"))
```

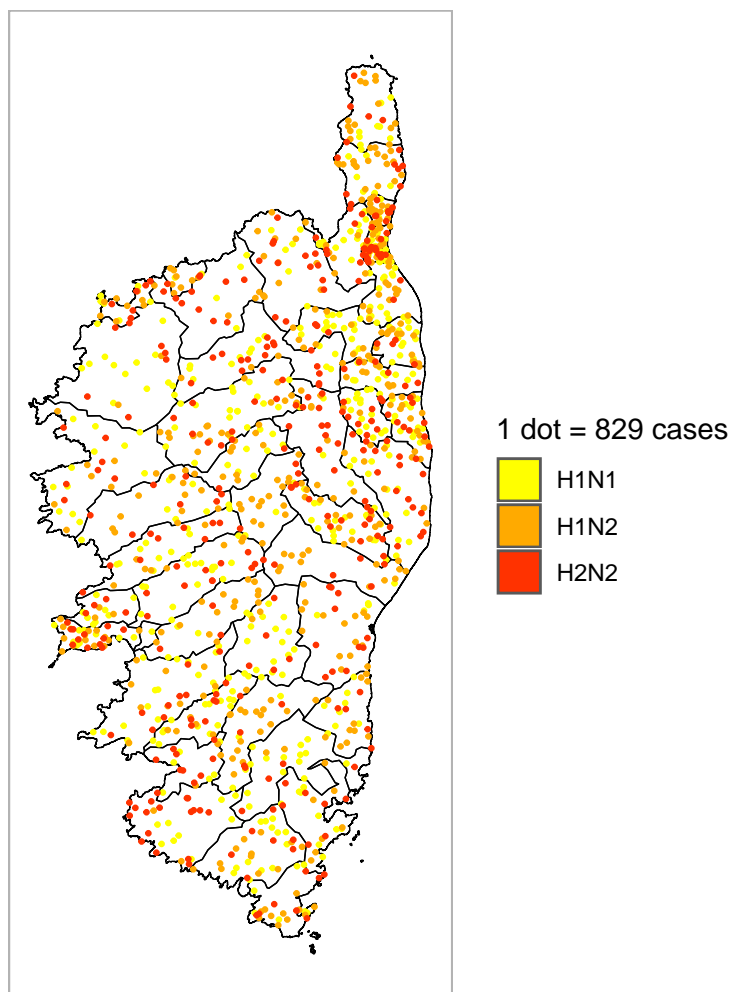


Figure 8: Dot-Density

## Plotting density

```
library(GADMTools)

isopleth(Corse, data = DAT, palette = "Blues")
```

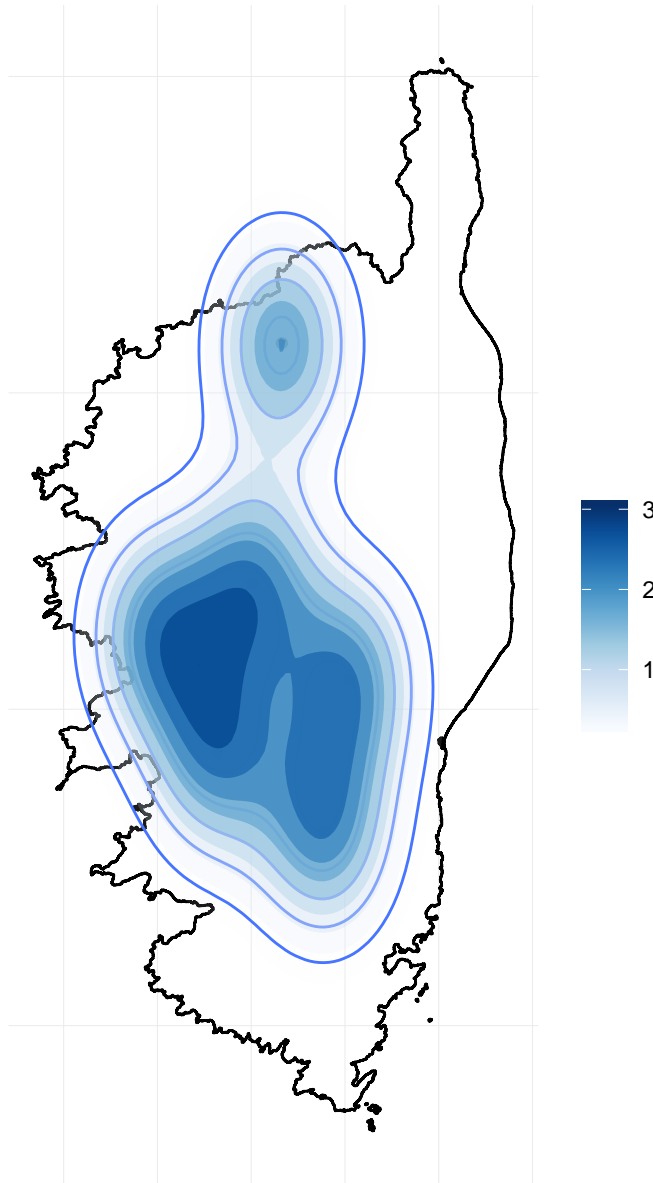


Figure 9: Isopleth

## Plotting a choropleth

```
DAT <- data.frame(Cantons, Pop, stringsAsFactors = FALSE)
choropleth(Corsica, data = DAT, value = "Pop", adm.join = "Cantons",
           breaks = "sd", palette = "Greens")
```

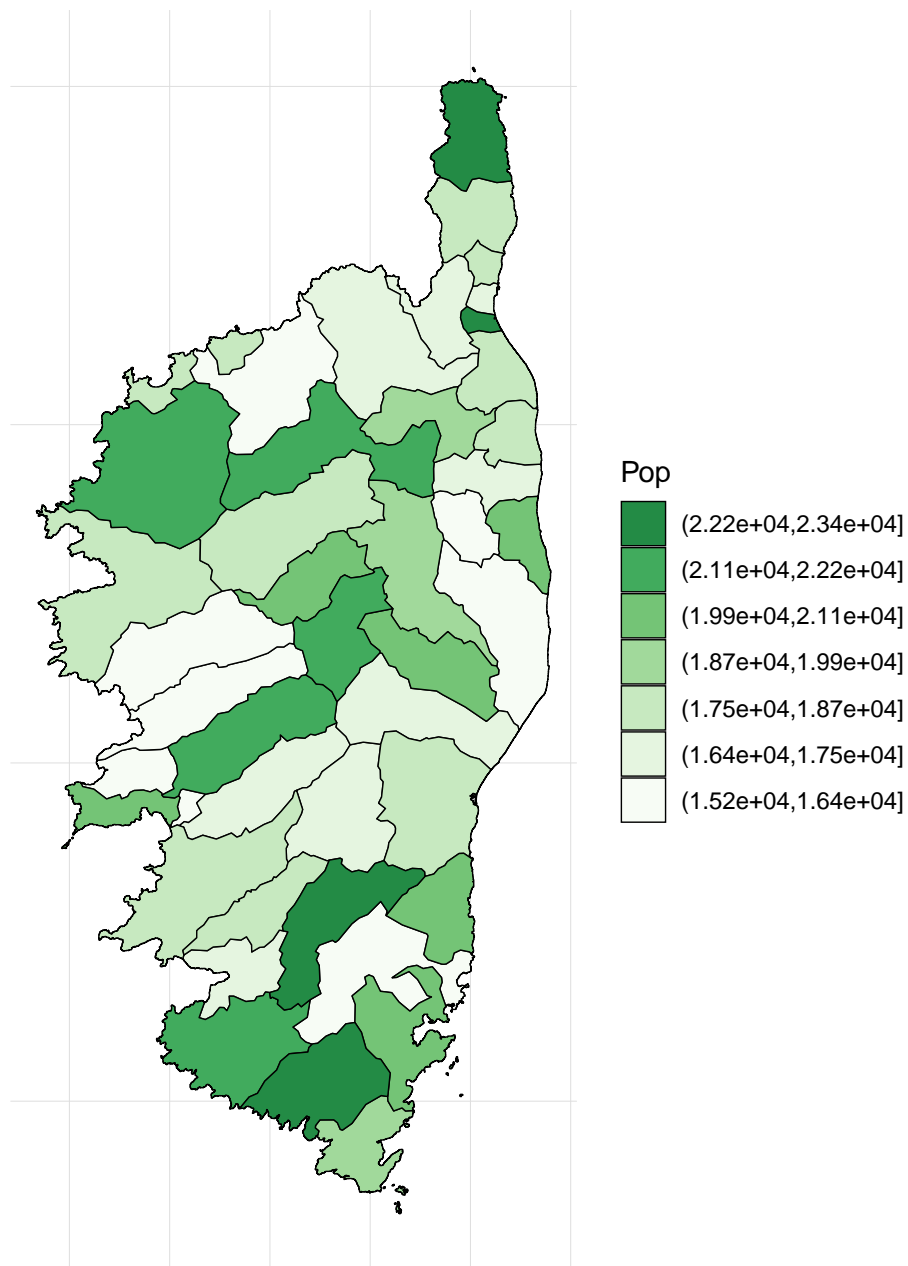


Figure 10: Choropleth

**fast.choropleth()**

```
fast.choropleth(  
  x, data, value=NULL,  
  
  breaks = NULL, steps = 5,  
  
  adm.join=NULL, legend = NULL,  
  
  labels = NULL,  
  
  palette=NULL, title=""  
)
```

Parameter	Description
<b>x</b>	<b>Object</b> GADMWrapper
<b>data</b>	<b>data.frame</b> - data to plot
<b>value</b>	<b>String</b> - the name of the column in the data.frame we want to plot (eg: an incidence in epidemiology studies)
<b>breaks</b>	
<b>steps</b>	<b>Integer</b> - number of breaks. Default = 5. If breaks is NOT NULL this value is used internally with cut().
<b>adm.join</b>	<b>String</b> - the name in GADM spdf dataset which will be joined with a column of the data.
<b>legend</b>	<b>String</b> - legend title. Default NULL.
<b>labels</b>	<b>String vector</b> labels for the legend. Default NULL
<b>palette</b>	<b>String</b> - An RColorBrewer palette name or a String vector vector of colors. Default NULL.
<b>title</b>	<b>String</b> - Title of the plot. Default is an empty string.



## Chlamydia incidence by Belgian district (2003)



Figure 11: drawing a fast.choropleth

### Example

```
MAP <- gadm_sp_loadCountries("BEL", level = 3, simplify=0.01)
DAT = read.csv2("BE_clamydia_incidence.csv")

# Rewriting District names
# -----
DAT$district <- as.character(DAT$district)
DAT[7,1] = "Brussel"
DAT[20,1] <- "Liège"
DAT[22,1] = "Marche-en-Famenne"
DAT[27,1] = "Neufchâteau"
DAT <- rename(DAT, NAME_3 = district)

fast.choropleth(MAP, DAT,
  adm.join = "NAME_3",
  value = "rate03",
  steps = 4,
  breaks = "jenks",
  palette="Greens",
  legend = "Incidence",
  title="Chlamydia incidence by Belgian district (2003)")
```