Hexagon tessellation using the great [geogrid](https://cran.r-project.org/package=geogrid) package.

The [*départements*](https://en.wikipedia.org/wiki/Departments_of_France) are the second level of administrative government in France. They neither have the same area nor the same population and this heterogeneity provides a few challenges for a fair and accurate map representation

However if we are just interested in the *départements* as *units*, we can use a regular grid for visualisation. Since France is often called the hexagon, we could even use an hexagon tiling (a fractal map !)…

Creating the grid and conserving minimal topological relations and the general shape can be time consuming, but thanks to Geogrid it’s quite easy.We will reuse our code of the [COVID19 animation](http://r.iresmi.net/2020/04/01/covid-19-decease-animation-map/).

# Carto décès COVID 19 hexagones

# France métro. + DOM

# Animation France métro.

# DONNEES SPF

# packages ----------------------------------------------------------------

library(tidyverse)

library(httr)

library(fs)

library(sf)

library(readxl)

library(janitor)

library(glue)

library(tmap)

library(grid)

library(classInt)

library(magick)

library(geogrid)

# sources -----------------------------------------------------------------

# <https://www.data.gouv.fr/fr/datasets/donnees-hospitalieres-relatives-a-lepidemie-de-covid-19/>

fichier\_covid <- "donnees/covid.csv"

url\_donnees\_covid <- "<https://www.data.gouv.fr/fr/datasets/r/63352e38-d353-4b54-bfd1-f1b3ee1cabd7>"

# <https://www.insee.fr/fr/statistiques/2012713#tableau-TCRD_004_tab1_departements>

fichier\_pop <- "donnees/pop.xls"

url\_donnees\_pop <- "<https://www.insee.fr/fr/statistiques/fichier/2012713/TCRD_004.xls>"

# Adminexpress : à télécharger manuellement

# <https://geoservices.ign.fr/documentation/diffusion/telechargement-donnees-libres.html#admin-express>

aex <- path\_expand("~/Downloads/ADMIN-EXPRESS\_2-2\_\_SHP\_\_FRA\_2020-02-24/ADMIN-EXPRESS/1\_DONNEES\_LIVRAISON\_2020-02-24")

# config ------------------------------------------------------------------

force\_download <- FALSE # retélécharger même si le fichier existe et a été téléchargé aujourd'hui ?

# téléchargement ------------------------------------------------------

if (!dir\_exists("donnees")) dir\_create("donnees")

if (!dir\_exists("resultats")) dir\_create("resultats")

if (!dir\_exists("resultats/animation\_spf\_hex")) dir\_create("resultats/animation\_spf\_hex")

if (!file\_exists(fichier\_covid) |

file\_info(fichier\_covid)$modification\_time < Sys.Date() |

force\_download) {

GET(url\_donnees\_covid,

progress(),

write\_disk(fichier\_covid, overwrite = TRUE)) %>%

stop\_for\_status()

}

if (!file\_exists(fichier\_pop)) {

GET(url\_donnees\_pop,

progress(),

write\_disk(fichier\_pop)) %>%

stop\_for\_status()

}

# données -----------------------------------------------------------------

covid <- read\_csv2(fichier\_covid)

# adminexpress prétéléchargé

dep <- read\_sf(path(aex, "ADE\_2-2\_SHP\_LAMB93\_FR/DEPARTEMENT.shp")) %>%

clean\_names() %>%

mutate(surf\_ha = st\_area(geometry) \* 10000) %>%

st\_set\_crs(2154)

# grille hexagonale

dep\_cells\_hex <- calculate\_grid(shape = dep, grid\_type = "hexagonal", seed = 3)

dep\_hex <- assign\_polygons(dep, dep\_cells\_hex) %>%

st\_set\_crs(2154)

# Pour les DOM on duplique et déplace un département existant

d971 <- dep\_hex[dep\_hex$insee\_dep == "29", ]

d971$geometry[[1]] <- d971$geometry[[1]] + st\_point(c(0, -150000))

d971$insee\_dep = "971"

d972 <- dep\_hex[dep\_hex$insee\_dep == "29", ]

d972$geometry[[1]] <- d972$geometry[[1]] + st\_point(c(0, -250000))

d972$insee\_dep = "972"

d973 <- dep\_hex[dep\_hex$insee\_dep == "29", ]

d973$geometry[[1]] <- d973$geometry[[1]] + st\_point(c(0, -350000))

d973$insee\_dep = "973"

d974 <- dep\_hex[dep\_hex$insee\_dep == "2A", ]

d974$geometry[[1]] <- d974$geometry[[1]] + st\_point(c(0, 250000))

d974$insee\_dep = "974"

d976 <- dep\_hex[dep\_hex$insee\_dep == "2A", ]

d976$geometry[[1]] <- d976$geometry[[1]] + st\_point(c(0, 350000))

d976$insee\_dep = "976"

dep\_hex <- rbind(dep\_hex, d971, d972, d973, d974, d976)

# population

pop <- read\_xls(fichier\_pop, skip = 2) %>%

clean\_names()

# lignes de séparation DOM / métropole

encarts <- st\_multilinestring(

list(st\_linestring(matrix(c(1100000, 6500000,

1100000, 6257000,

1240000, 6257000), byrow = TRUE, nrow = 3)),

st\_linestring(matrix(c(230000, 6692000,

230000, 6391000), byrow = TRUE, nrow = 2)))) %>%

st\_sfc() %>%

st\_sf(id = 1, geometry = .) %>%

st\_set\_crs(2154)

# traitement --------------------------------------------------------------

# jointures des données

creer\_df <- function(territoire, date = NULL) {

territoire %>%

left\_join(pop, by = c("insee\_dep" = "x1")) %>%

left\_join(

covid %>%

filter(jour == if\_else(is.null(date), max(jour), date),

sexe == 0) %>%

rename(deces = dc,

reanim = rea,

hospit = hosp),

by = c("insee\_dep" = "dep")) %>%

mutate(incidence = deces / x2020\_p \* 100000)

}

incidence <- creer\_df(dep\_hex)

set.seed(1234)

classes <- classIntervals(incidence$incidence, n = 6, style = "kmeans", dataPrecision = 0)$brks

# carto -------------------------------------------------------------------

# décès cate du dernier jour dispo

carte <- tm\_layout(title = glue("COVID-19\nFrance\n{max(covid$jour)}"),

legend.position = c("left", "bottom"),

frame = FALSE) +

tm\_shape(incidence) +

tm\_polygons(col = "incidence", title = "décés\ncumulés pour\n100 000 hab.",

breaks = classes,

palette = "viridis",

legend.reverse = TRUE,

legend.format = list(text.separator = "à moins de",

digits = 0)) +

tm\_text("insee\_dep", size = .8) +

tm\_shape(encarts) +

tm\_lines(lty = 3) +

tm\_credits(glue("<http://r.iresmi.net/>

classif. kmeans

données départementales Santé Publique France,

INSEE RP 2020, d'après IGN Adminexpress 2020"),

position = c(.6, 0),

size = .5)

fichier\_carto <- glue("resultats/covid\_hex\_fr\_{max(covid$jour)}.png")

tmap\_save(carte, fichier\_carto, width = 900, height = 900, scale = .4)

# animation ---------------------------------------------------------------

image\_animation <- function(date) {

message(glue("\n\n{date}\n==========\n"))

m <- creer\_df(dep\_hex, date) %>%

tm\_shape() +

tm\_polygons(col = "incidence", title = "décés\ncumulés pour\n100 000 hab.",

breaks = classes,

palette = "viridis",

legend.reverse = TRUE,

legend.format = list(text.separator = "à moins de",

digits = 0)) +

tm\_text("insee\_dep", size = .8) +

tm\_shape(encarts) +

tm\_lines(lty = 3) +

tm\_layout(title = glue("COVID-19\nFrance\n{date}"),

legend.position = c("left", "bottom"),

frame = FALSE) +

tm\_credits(glue("<http://r.iresmi.net/>

classif. kmeans

données départementales Santé Publique France,

INSEE RP 2020, d'après IGN Adminexpress 2020"),

position = c(.6, 0),

size = .5)

tmap\_save(m, glue("resultats/animation\_spf\_hex/covid\_fr\_{date}.png"),

width = 800, height = 800, scale = .4,)

}

unique(covid$jour) %>%

walk(image\_animation)

animation <- glue("resultats/deces\_covid19\_fr\_hex\_spf\_{max(covid$jour)}.gif")

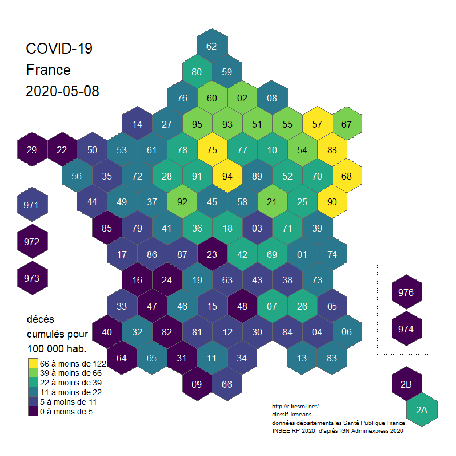
dir\_ls("resultats/animation\_spf\_hex") %>%

map(image\_read) %>%

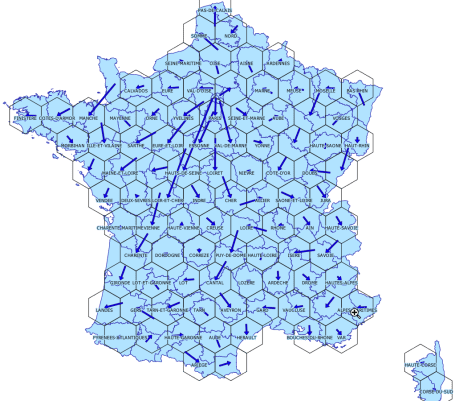
image\_join() %>%

image\_animate(fps = 2, optimize = TRUE) %>%

image\_write(animation)



COVID decease

The global shape and relations are quite well rendered. Deformations are quite important for the small *départements* around Paris, but it’s quite legible. 

Shift