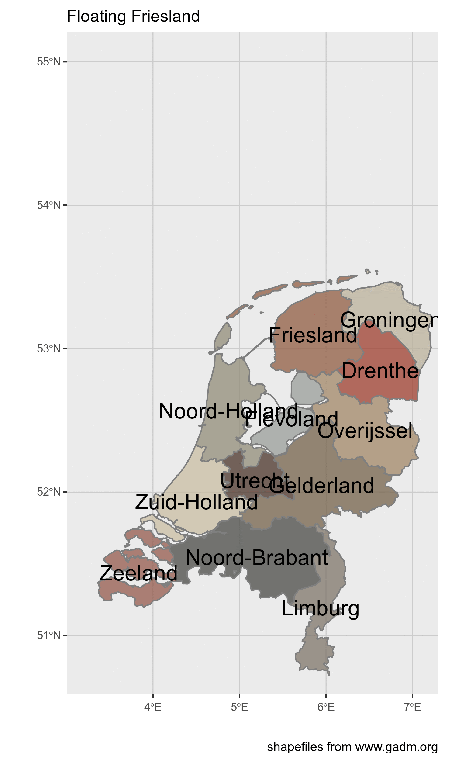
We love making maps, We also love making gifs.  
In this short post I make an animated gif of parts of a map moving. In this case the parts of the map only move in the xy direction, but [you can also turn them, and make them bigger or smaller](https://r-spatial.github.io/sf/articles/sf3.html#affine-transformations).  
Today I show you how I made a part of the Netherlands ‘float away’. It is part of a larger nonsense project (I have many silly projects), and mostly just to document my path to learning about spatial analytics.

End result :



**General principles**

* make small functions that do one thing well (not that small in this case)
* combine those
* make imagemagick canvas
* write to the canvas
* animate the canvas

**loading libraries and data**

suppressMessages(library(tidyverse)) # ggplot2, dplyr, purrr, etc.

library(magick)

library(sf)

library(paletti) # thanks @edwinthoen

# colorscheme

dutchmasters\_fill <- get\_scale\_fill(get\_pal(dutchmasters))

# the data

NLD <- read\_sf("data/NLD\_adm1.shp") # I cannot redistribute the data from GADM, but you can download and use it for your projects

**basic functions**

I created a function that takes a name, uses that to filter the data and apply a transformation on that part only. (a mutate\_if() could also work, but I didn’t know how). And also one that uses that function to plot. The final function takes a matrix of xy values and sequentially applies every row to the plotting function.

* modify data
* plot a single ggplot version
* loop or apply over range

# basic function that moves an a province

move\_province <- function(provincename, movement){

mov <- quo(movement)

rest <- NLD %>%

filter(NAME\_1 != !!provincename) %>%

filter(TYPE\_1 != "Water body")

#rest %>% st\_centroid() %>% st\_as\_text()

province <- NLD %>%

filter(NAME\_1 == !!provincename) %>%

mutate(geometry = geometry + !!mov) %>%

st\_set\_crs("+proj=longlat +datum=WGS84 +no\_defs")

data1 <-

rbind(province, rest)

centroids <-

data1 %>% st\_centroid() %>% st\_coordinates()

cbind(data1, centroids)

}

# make function to create plot

# using the previous function to move the province

plot\_netherlands <- function(province, movement){

plotunit <- move\_province(provincename = province, movement = movement) %>%

ggplot()+

geom\_sf(aes(fill = NAME\_1),color = "grey50", alpha = 3/4)+

geom\_text(aes(X,Y, label = NAME\_1), size = 6)+

lims(x = c(3.2,7.1), y = c(50.8,55))+

labs(x="", y = "", caption = "shapefiles from www.gadm.org", title = "Floating Friesland")+

dutchmasters\_fill("little\_street")+

theme( legend.position = "empty", # we already labeled the provinces

panel.grid.major = element\_line(colour = "grey80"))

print(plotunit) # you have to explicitly tell it to print so the image is captured

}

# go over every frame and print

plot\_province\_over\_range <-

function(offset\_matrix, province = "Friesland", debug = FALSE){

if(any(is.na(offset\_matrix))){stop("I cannot handle empty movements, there are NA's in movement\_matrix")}

if(NCOL(offset\_matrix) != 2) stop("movement\_matrix needs to have exactly 2 columns")

actionsframe <- data\_frame(x = offset\_matrix[,1], y = offset\_matrix[,2]) %>%

mutate(rownumber = row\_number())

actionsframe$name <- paste0(formatC(actionsframe$rownumber, flag = 0,width = 4))

pb <- progress\_estimated(NROW(actionsframe))

walk(actionsframe$name, ~{

pb$tick()$print()

vars <- filter(actionsframe, name == .x)

if(debug){

message("using values from: ",vars)

}

plot\_netherlands(province = province,movement = c(vars$x[[1]], vars$y[[1]]))

}) # ends the walk action

}

**The plotting and saving**

Nothing happened before the next step (except loading data). All the action and calculation happens here.

## then the creation starts with the movement

Friesland\_moves <- rbind(

matrix(c(c(0,-.1,-.2,-.2,-.3), c(0,.03,.05,.1,.15)) ,ncol = 2),

matrix(c(seq(from = -.3, by = -.1, length.out = 14),seq(from = .2, by = .1, length.out = 14)), ncol = 2)

)

# set up print location

frames <- image\_graph(width = 1500, height = 2500, res = 300, pointsize = 5)

plot\_province\_over\_range(offset\_matrix = Friesland\_moves, province = "Friesland")

# animate

image\_animate(frames, 1) %>%

image\_write(path = "friesland.gif")