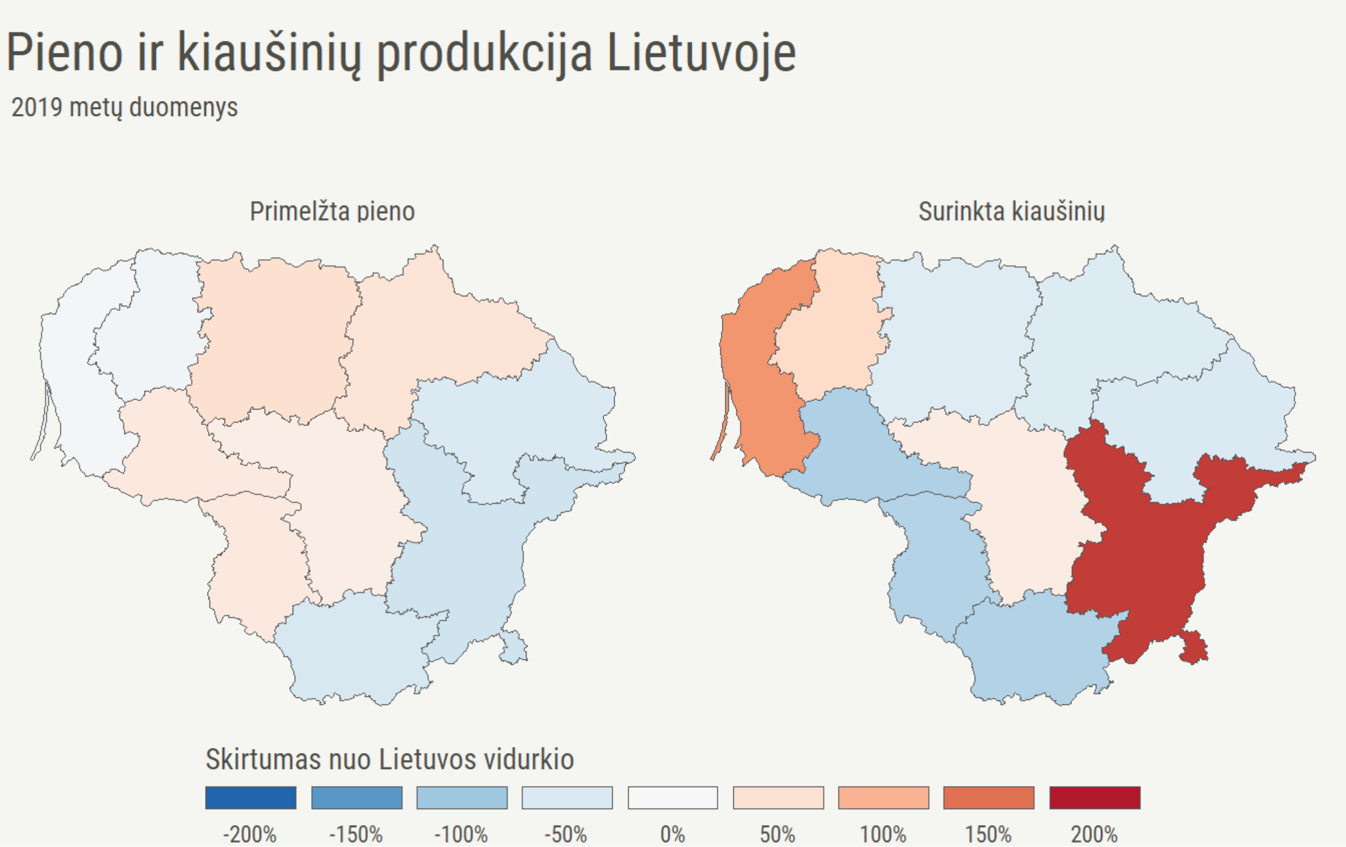
R Notebook

```r
library(sf)
library(tidyverse)
library(hrbrthemes)
library(countrycode)
library(cartogram)
library(rnaturalearth)
library(rgeos)

skaitymas <- function(pavadinimas){
  read_csv(pavadinimas) %>%
    janitor::clean_names() %>%
    select(-matavimo_vienetai) %>%
    filter(str_detect(administracine_teritorija,"apskritis"))
}
```

```r
latlongdata<- sf::read_sf("NUTS_RG_01M_2021_3857.geojson")
latlongdata<- latlongdata %>%
  filter(CNTR_CODE == "LT",LEVL_CODE == 3) %>%
  select(id, NAME_LATN, geometry) %>%
  rename(administracine_teritorija = NAME_LATN)

pienas<- skaitymas("pienas.csv") #y1
kiausiniai<- skaitymas("kiausiniai.csv") #y2
populiacija<- skaitymas("populiacija.csv") %>%
  select(laikotarpis,administracine_teritorija,reiksme) %>%
  rename(gyventojai=reiksme) #y3

rodikliai<-rbind(pienas,kiausiniai)
data<- merge(rodikliai,populiacija,by=c("laikotarpis","administracine_teritorija"))

data<- data %>%
  mutate(reiksmeweighted = reiksme/gyventojai*100000) %>%
  group_by(rodiklis,laikotarpis) %>%
  mutate(reiksmeweighted = reiksme/mean(reiksme)-1)

geo_data<-merge(data,latlongdata,by=c("administracine_teritorija")) #z
```

1

```
ggplot(filter(geo_data,laikotarpis==2019),aes(geometry=geometry,fill=reiksmeweighted)) +
  geom_sf(size=0.1) +
  scale_fill_distiller("Skirtumas nuo Lietuvos vidurkio",
                       palette = "RdBu",limits=c(-2,2),
                       breaks = c(-2,-1.5,-1,-0.5,0,0.5,1,1.5,2),
                       labels=c("-200%","-150%","-100%","-50%","0%","50%","100%","150%",
"200%"),
                       guide = guide_legend(
                         keyheight = unit(3, units = "mm"),
                         keywidth=unit(12, units = "mm"),
                         label.position = "bottom", title.position = 'top', nrow=1)
  ) +
  facet_wrap(vars(rodiklis)) +
  theme_void() +
  theme(
    text = element_text(color = "#4e4d47",family="Roboto Condensed"),
    plot.background = element_rect(fill = "#f5f5f2", color = NA),
    panel.background = element_rect(fill = "#f5f5f2", color = NA),
    legend.background = element_rect(fill = "#f5f5f2", color = NA),
    legend.position = "bottom",
    plot.title = element_text(size= 20, hjust=0.01, margin = margin(b = 0.2, t = 0.4, l
 = 2, unit = "cm")),
    plot.subtitle = element_text(size= 10, hjust=0.01, margin = margin(b = 1, t = 0, l =
2, unit = "cm")),
    plot.caption = element_text(size=8, margin = margin(b = 0.3, r=-99, unit = "cm")),
    strip.text = element_text(size = 10)) +
  labs( title = "Pieno ir kiaušinių produkcija Lietuvoje",
        subtitle = "2019 metų duomenys"
  )
```



Pieno ir kiaušinių produkcija Lietuvoje

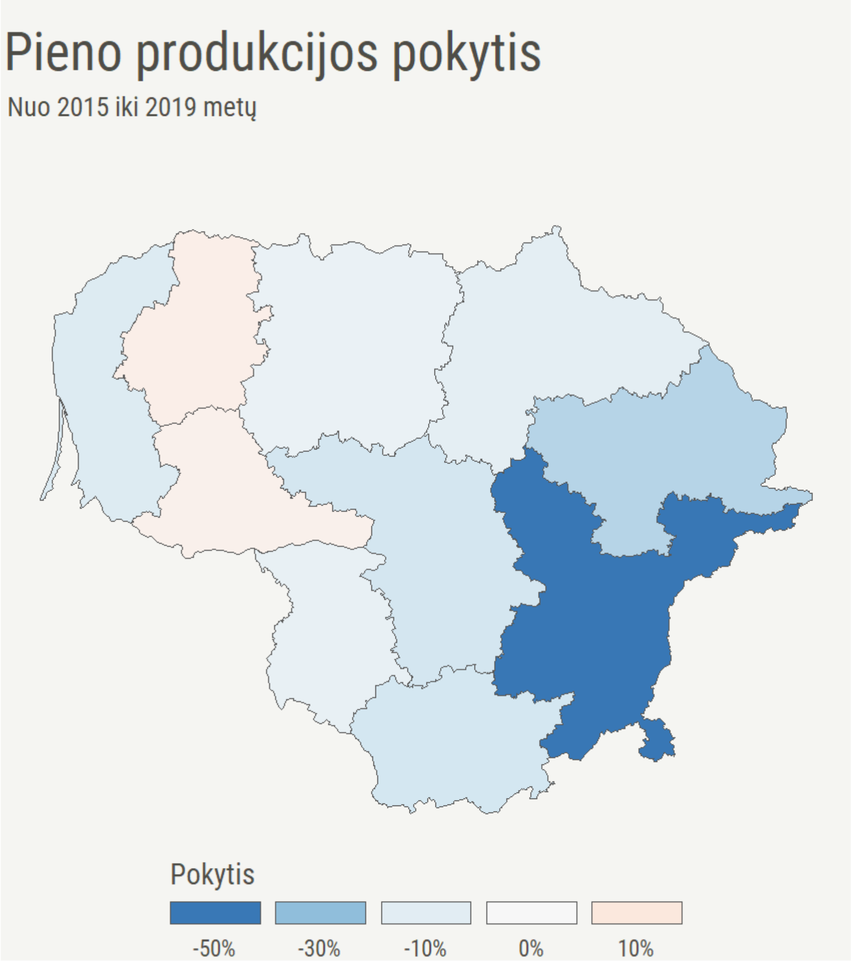
2019 metų duomenys

Primelžta pieno

Surinkta kiaušinių

2

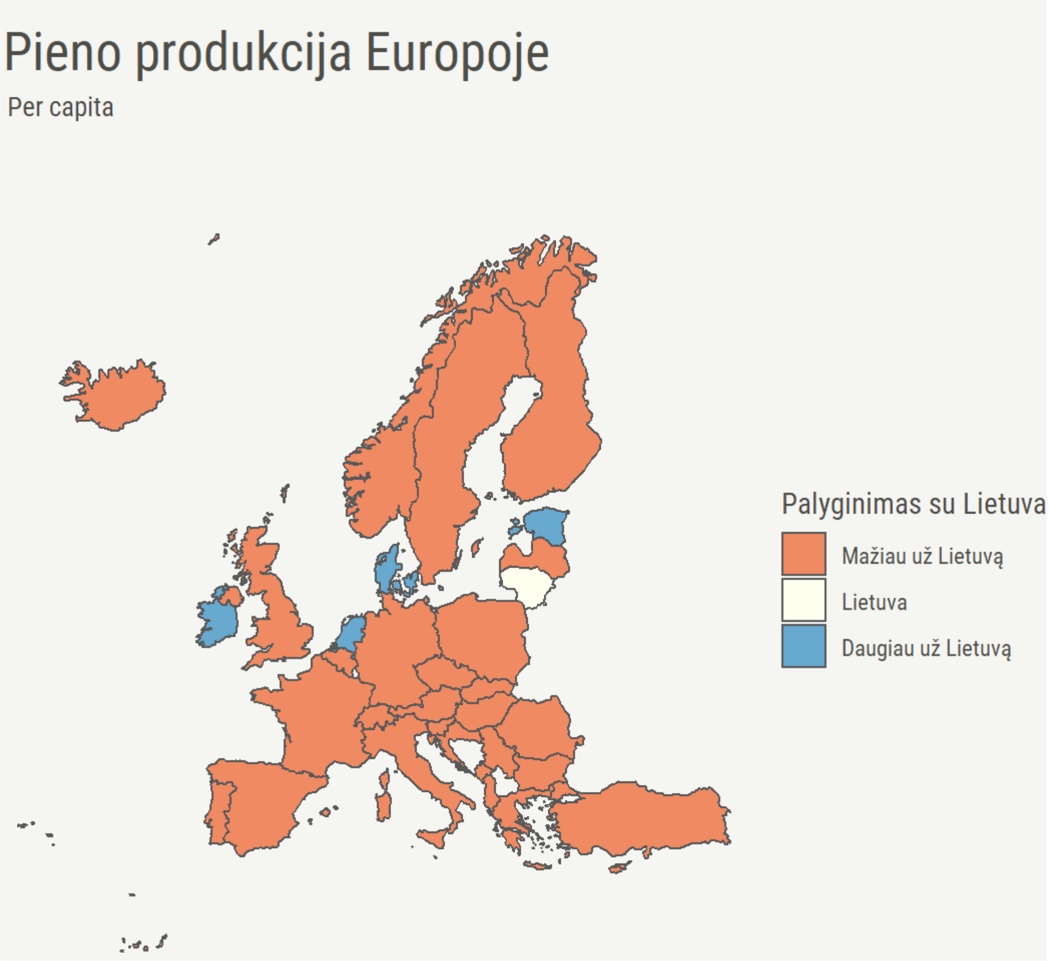
```r
second_data <- data %>% select(-reiksmeweighted,-gyventojai) %>% pivot_wider(names_from=
laikotarpis,values_from=reiksme) %>%
  mutate(diff = (`2019`-`2015`)/`2015`) %>%
  filter(rodiklis == "Primelžta pieno")

geo_second_data<-merge(second_data,latlongdata,by=c("administracine_teritorija"))

ggplot(filter(geo_second_data),aes(geometry=geometry,fill=diff)) +
  geom_sf(size=0.1) +
  theme_void() +
  scale_fill_distiller("Pokytis",
                       palette = "RdBu",limits=c(-0.55,0.55),
                       breaks = c(-0.50,-0.3,-0.1,0,0.1),
                       labels=c("-50%","-30%","-10%","0%","10%"),
                       guide = guide_legend(
                         keyheight = unit(3, units = "mm"),
                         keywidth=unit(12, units = "mm"),
                         label.position = "bottom", title.position = 'top', nrow=1)
  ) +
  theme(
    text = element_text(color = "#4e4d47",family="Roboto Condensed"),
    plot.background = element_rect(fill = "#f5f5f2", color = NA),
    panel.background = element_rect(fill = "#f5f5f2", color = NA),
    legend.background = element_rect(fill = "#f5f5f2", color = NA),
    legend.position = "bottom",
    plot.title = element_text(size= 20, hjust=0.01, margin = margin(b = 0.2, t = 0.4, l
 = 2, unit = "cm")),
    plot.subtitle = element_text(size= 10, hjust=0.01, margin = margin(b = 1, t = 0, l =
2, unit = "cm")),
    plot.caption = element_text(size=8, margin = margin(b = 0.3, r=-99, unit = "cm")),
    strip.text = element_text(size = 10)) +
  labs(title = "Pieno produkcijos pokytis",
       subtitle = "Nuo 2015 iki 2019 metų"
  )
```



3

```
third_data<- janitor::clean_names(read_csv("demo_pjan_1_Data.csv")) #y3
fourth_data<- janitor::clean_names(read_csv("apro_mk_cola_1_Data.csv")) #y4
```

```
third_data <- merge(third_data,fourth_data, by = c("geo","time"))
third_data<- third_data %>% merge(fourth_data,c("geo","time")) %>% filter(time == 2019)
 %>%
  mutate(value.x = as.numeric(str_replace_all(value.x,",","")),value.y=as.numeric(str_re
place_all(value.y,",",""))) %>%
  mutate(valueweighted = value.y/value.x) %>%
  drop_na() %>%
  mutate(code = countrycode(geo,"country.name","eurostat"))

weighted_lt <- third_data[["valueweighted"]][third_data$geo=="Lithuania"]
third_data <- third_data %>%
  mutate(compare = valueweighted>weighted_lt,
         compare =ifelse(geo=="Lithuania","Lithuania",compare))

world <- ne_countries(scale = "medium", returnclass = "sf")

world <- world %>%
  mutate(code = countrycode(name_long,"country.name","eurostat"))

geo_third<- merge(world,third_data,by=c("code"))

geo_third<-st_transform(geo_third,"EPSG:3857")

ggplot(geo_third,aes(fill=compare)) +
  geom_sf() +
  scale_fill_manual("Palyginimas su Lietuva",values = c("#ef8a62","ivory","#67a9cf"),
                    breaks=c("FALSE","Lithuania","TRUE"),
                    labels=c("Mažiau už Lietuvą","Lietuva","Daugiau už Lietuvą")) +
  theme_void() +
  coord_sf(xlim=c(-3000000,5000000),ylim=c(3500000,11500000)) +
  theme(
    text = element_text(color = "#4e4d47",family="Roboto Condensed"),
    plot.background = element_rect(fill = "#f5f5f2", color = NA),
    panel.background = element_rect(fill = "#f5f5f2", color = NA),
    legend.background = element_rect(fill = "#f5f5f2", color = NA),
    plot.title = element_text(size= 20, hjust=0.01, margin = margin(b = 0.2, t = 0.4, l
 = 2, unit = "cm")),
    plot.subtitle = element_text(size= 10, hjust=0.01, margin = margin(b = 1, t = 0, l =
2, unit = "cm")),
    plot.caption = element_text(size=8, margin = margin(b = 0.3, r=-99, unit = "cm")),
    strip.text = element_text(size = 10)) +
  labs(title = "Pieno produkcija Europoje",
       subtitle="Per capita")
```



4

```
geo_fourth<- drop_na(geo_third,value.y)
geo_fourth$value.y <- geo_fourth$value.y*1000000
geo_fourth<- cartogram_cont(geo_fourth, "value.y", itermax=7,threshold = 0.1)
```

```
ggplot(geo_fourth) + geom_sf() +
  theme_void() +
  coord_sf(xlim=c(-3000000,5500000),ylim=c(3500000,11500000)) +
  theme(
      text = element_text(color = "#4e4d47",family="Roboto Condensed"),
      plot.background = element_rect(fill = "#f5f5f2", color = NA),
      panel.background = element_rect(fill = "#f5f5f2", color = NA),
      legend.background = element_rect(fill = "#f5f5f2", color = NA),
      plot.title = element_text(size= 20, hjust=0.01, margin = margin(b = 0.2, t = 0.4,
 l = 2, unit = "cm")),
      plot.subtitle = element_text(size= 10, hjust=0.01, margin = margin(b = 1, t = 0, l
= 2, unit = "cm")),
      plot.caption = element_text(size=8, margin = margin(b = 0.3, r=-99, unit = "cm")),
      strip.text = element_text(size = 10)) +
  labs(title = "Pieno produkcijos kartograma",
       subtitle="2019 m. duomenys"
  )
```

Pieno produkcijos kartograma

2019 m. duomenys