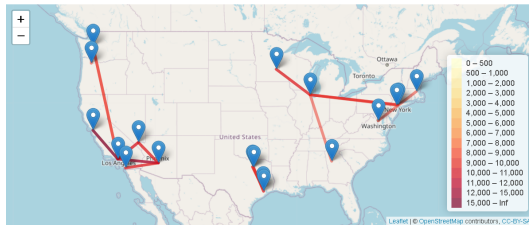


Techniki Wizualizacji Danych. HW 3

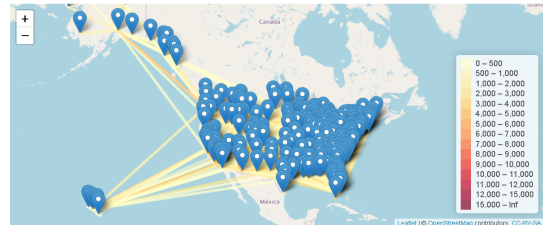
Jan Cwalina

dane: <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/HG7NV7>
<https://ourairports.com/data/>

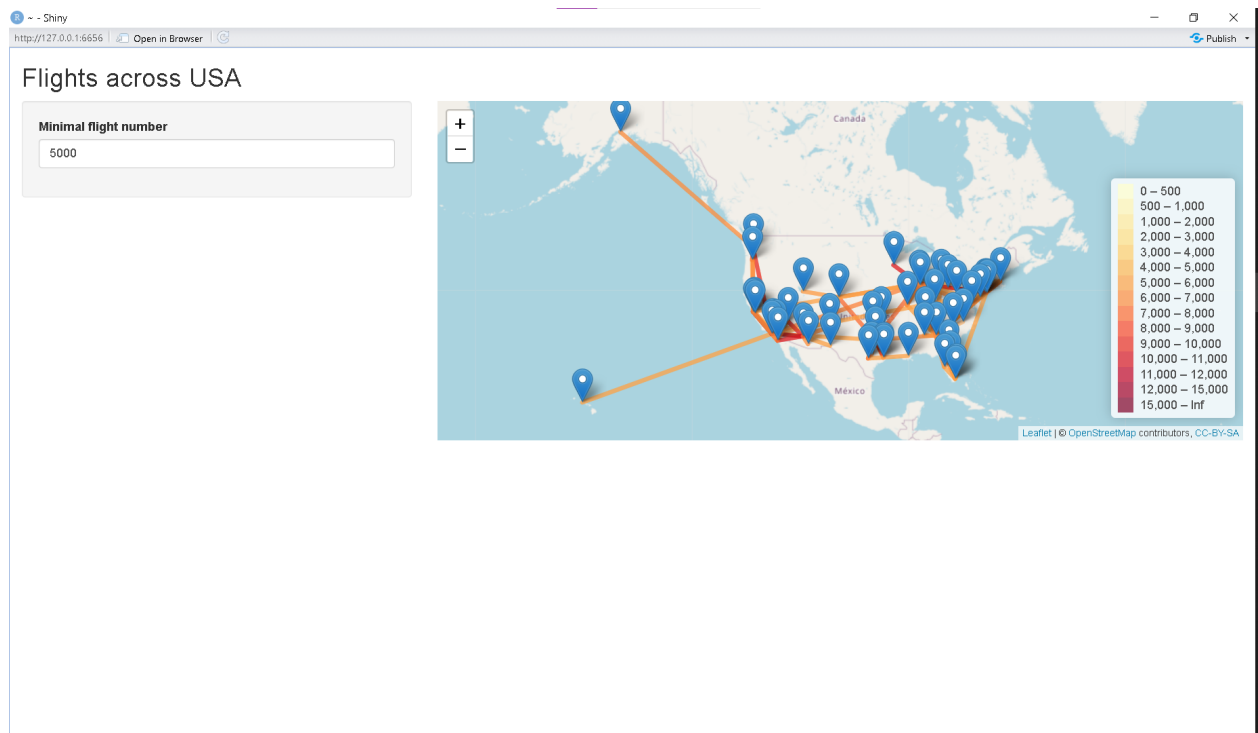
Mapka prezentuje loty komercyjne na terenie USA z roku 1990 ich ilość, trasę w linii prostej łączącą lotniska. Mapa jest interaktywna, po kliknięciu na trasę pokazuje “linia lotu” (Lotnisko1 <-> Lotnisko2) oraz ilość lotów w tym roku. Dodatkowo w oknie aplikacji SHINY możliwe jest ustawienie od jakiej ilość lotów w skali roku chcemy obserwować dane, dzięki czemu ,jeżeli chcemy obserwować te “linie” na których ilość lotów wynosiła co najmniej 9000 w roku, to widzimy to:



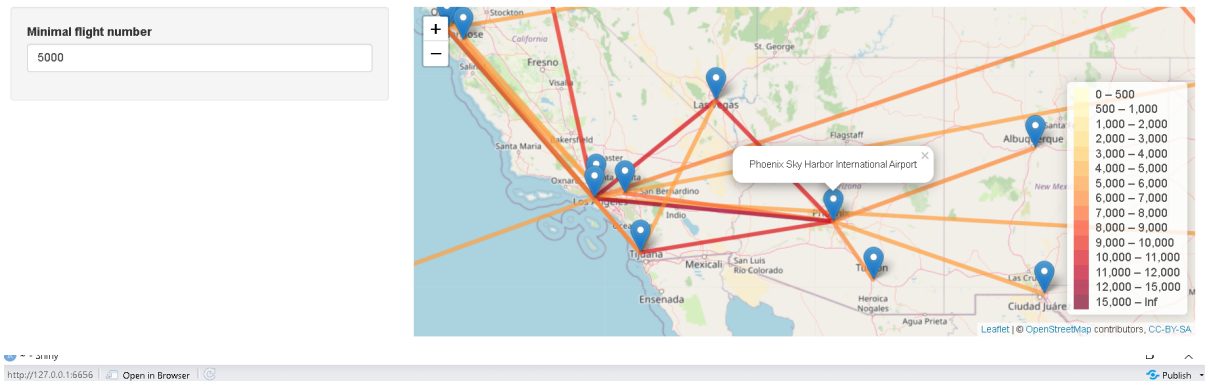
,a nie to:



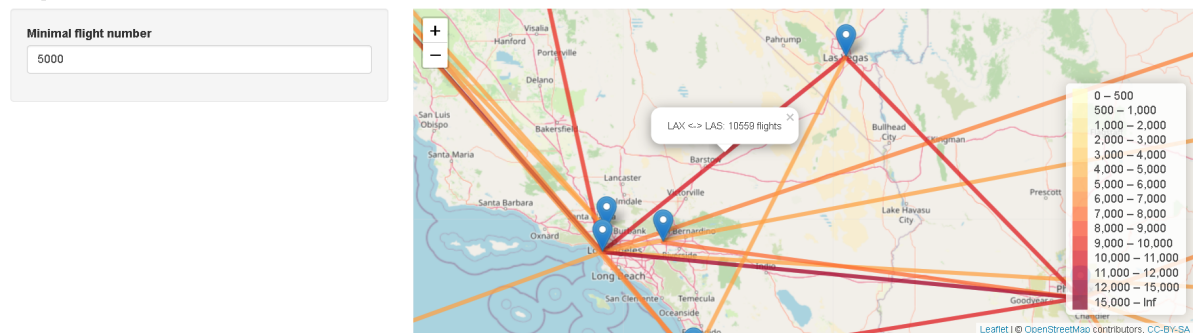
Zbędne lotniska z, którymi nie ma połączeń też nie są wyświetlane w zależności od zmiennej ilości lotów.



Flights across USA



Flights across USA



kod:

```
library(ggplot2)
library(maps)
library(dplyr)
library(leaflet)
library(mapdata)
library(shiny)
library(geojsonio)
```

```
airports <- read.csv("./R/PROJEKT_2/airports.csv")
Year1990 <- read.csv("./R/PROJEKT_2/1990.csv.bz2")
```

```
uniqueAirports <- unique(c(Year1990$Origin, Year1990$Dest))
```

```
AP2 <- airports %>%
  filter(iso_country == "US" & local_code %in% uniqueAirports) %>%
  select(local_code, longitude_deg, latitude_deg)
```

```
AirportsFromTo <- Year1990 %>% group_by(Origin, Dest) %>% summarise(n = n())
```

```

colnames(AirportsFromTo) <- c('Origin','Dest','n')
AirPortsTemp1 <- merge(AirportsFromTo ,AP2, by.x = "Dest", by.y = "local_code")
colnames(AirPortsTemp1) <- c("Dest","Origin","n","long_D","lat_D")
AirPortsTemp2 <- merge(AirPortsTemp1 ,AP2, by.x = "Origin", by.y = "local_code")
colnames(AirPortsTemp2) <- c("Origin","Dest","n","long_D","lat_D","long_O","lat_O")

Lotniska <- airports %>%
  filter(iso_country == "US" & local_code %in% uniqueAirports) %>%
  select(local_code,longitude_deg,latitude_deg,name)

bins <- c(0, 500, seq(1000,12000,by = 1000),15000,Inf)
pal <- colorBin("YlOrRd", domain = AirPortsTemp2$n, bins = bins)

ui <- fluidPage(
  titlePanel("Flights across USA"),
  sidebarLayout(
    sidebarPanel(
      numericInput(inputId = 'fn', 'Minimal flight number', 5000, min = 0, max = 27000)
    ),
    mainPanel(
      leafletOutput(outputId = "distPlot")
    )
  )
)
server <- function(input, output) {
  output$distPlot <- renderLeaflet({

    APT <- AirPortsTemp2 %>% filter(n > input$fn)

    wektorLiniLong <- c(rbind(APT$long_O,APT$long_D,rep(NA,length(APT$long_O))))
    wektorLiniLat <- c(rbind(APT$lat_O,APT$lat_D,rep(NA,length(APT$lat_O))))
    polaczenia <- paste(APT$Origin,APT$Dest,sep = " <-> ")

    Airports <- Lotniska %>% filter(Lotniska$local_code %in%
unique(c(APT$Origin,APT$Dest)))

    leaflet() %>% addPolylines(
      lat = wektorLiniLat,
      lng = wektorLiniLong,
      color = pal(APT$n),
      popup = paste(polaczenia, paste(as.character(APT$n), c(" flights"), sep = ""),sep = ": ")
    ) %>%
    addMarkers(
      lat = Airports$latitude_deg,
      lng = Airports$longitude_deg,
      popup = Airports$name
    ) %>% setView(-100, 37.8, 3) %>% addTiles() %>%
    addLegend(pal = pal, values = AirPortsTemp2$n, opacity = 0.7, title = NULL,

```

```
position = "bottomright")
```

```
}}
```

```
shinyApp(ui = ui, server = server)
```

```
runApp("my_app")
```