

**AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE**

**AGH UNIVERSITY OF SCIENCE
AND TECHNOLOGY**

AGH

Biblioteka leaflet – Java w służbie kartografii

Mgr inż. Damian Biel

Kraków 12.1.2018

Leaflet – co to takiego?

- Biblioteka JavaScript
- Tworzenie interaktywnych map
- Szeroko wykorzystywany przez prasę, serwisy społecznościowe, OSM itp.





Leaflet – co oferuje?


- Markowanie, pop-up na podkładzie mapowym
- Wgrywanie danych wektorowych i rastrowych
- Analizy i zapytania atrybutowe
- Współpraca z bazami danych


Przygotowanie


- Zebranie odpowiednich bibliotek


 acs_2.1.2.zip


 curl_3.1.zip


 geojson_0.2.0.zip


 htmlwidgets_0.9.zip


 jsonlite_1.5.zip


 maps_3.2.0.zip


 mime_0.5.zip


 R6_2.2.2.zip


 Rcpp_0.12.14.zip


 rmapshaper_0.3.0.zip


 shiny_1.0.5.zip


 tmap_1.11.zip


 XML_3.98-1.9.zip


 colorspace_1.3-2.zip


 devtools_1.13.4.zip


 geojsonio_0.5.0.zip


 httpuv_1.3.5.zip


 leaflet_1.1.0.zip


 mapview_2.2.0.zip


 munsell_0.4.3.zip


 raster_2.6-7.zip


 rgdal_1.2-16.zip


 scales_0.5.0.zip


 sp_1.2-6.zip


 tmaptools_1.2-2.zip


 xtable_1.8-2.zip


 crosstalk_1.0.0.zip


 digest_0.6.13.zip


 htmltools_0.3.6.zip


 jq_1.0.0.zip


 magrittr_1.5.zip


 markdown_0.8.zip


 plyr_1.8.4.zip


 RColorBrewer_1.1-2.zip

 rgeos_0.3-26.zip

 sf_0.6-0.zip

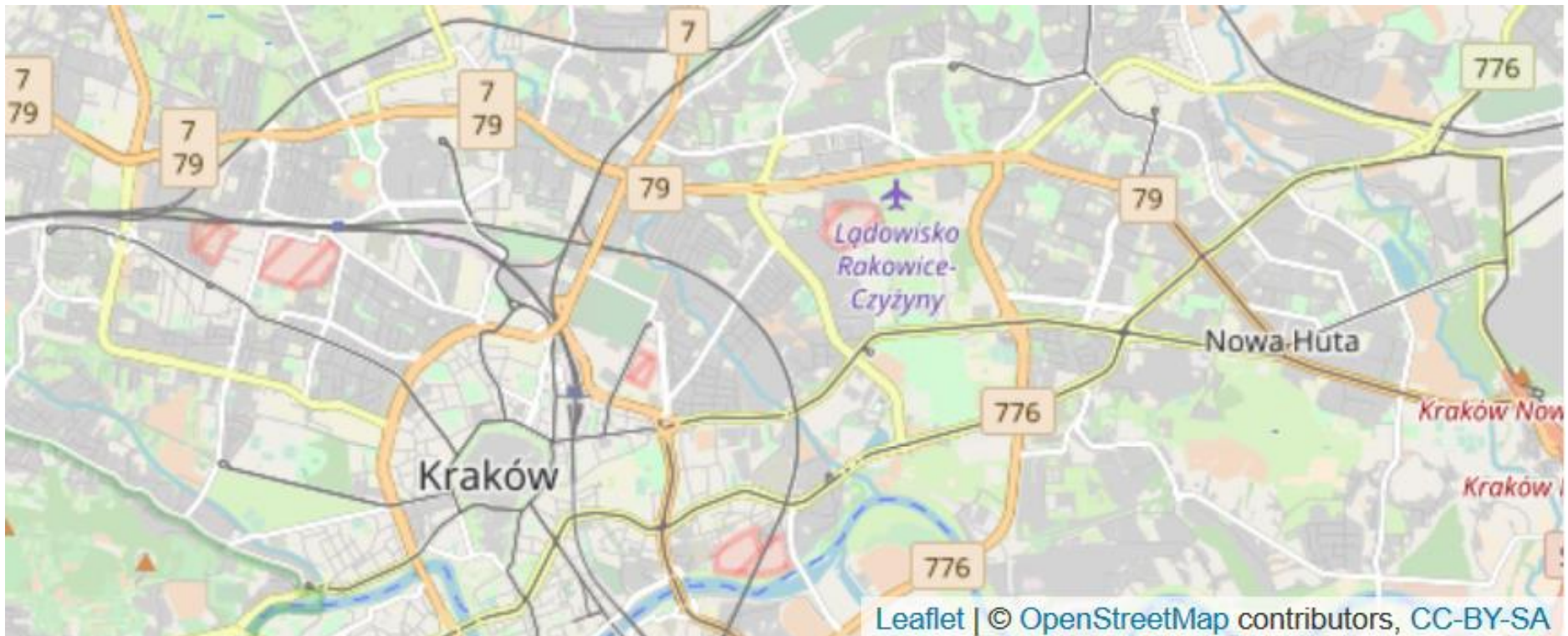
 tigris_0.6.2.zip

 V8_1.5.zip

 yaml_2.1.16.zip

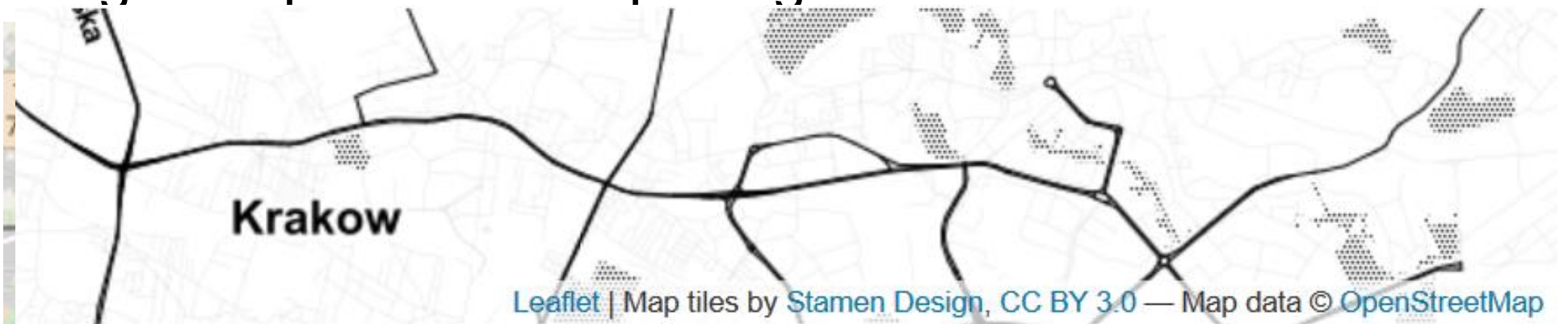
Przygotowanie

- Zebranie odpowiednich bibliotek
- Wgranie podkładu mapowego – `leaflet()` i `addTiles()`



Przygotowanie

- Zebranie odpowiednich bibliotek
- Wgranie podkładu mapowego



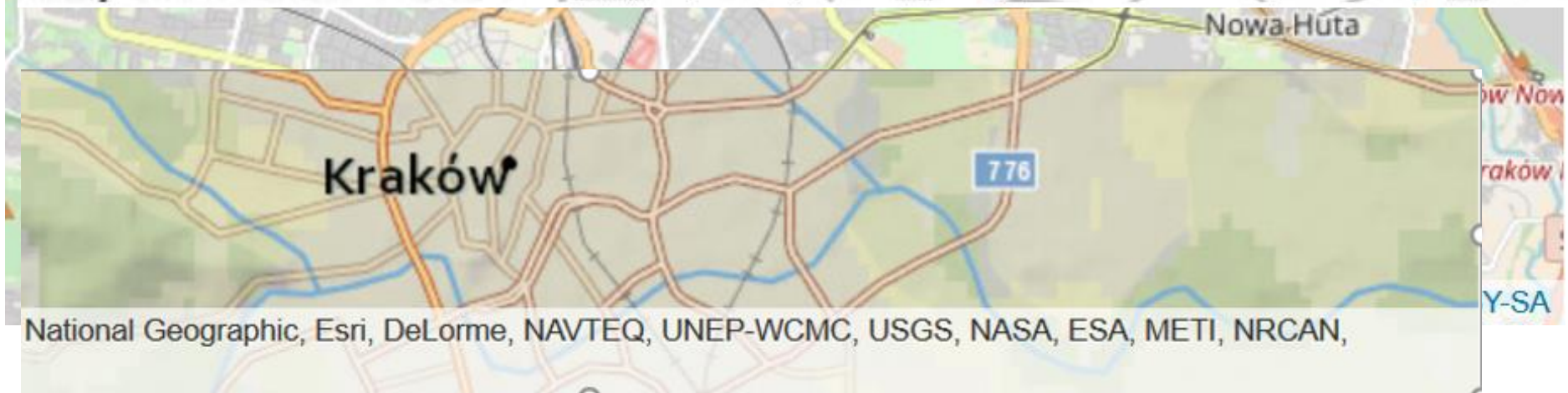
Przygotowanie

```
m <- leaflet() %>% setView(lng = 20.0000, lat = 50.0000, zoom = 12)
```

```
m %>% addTiles()
```

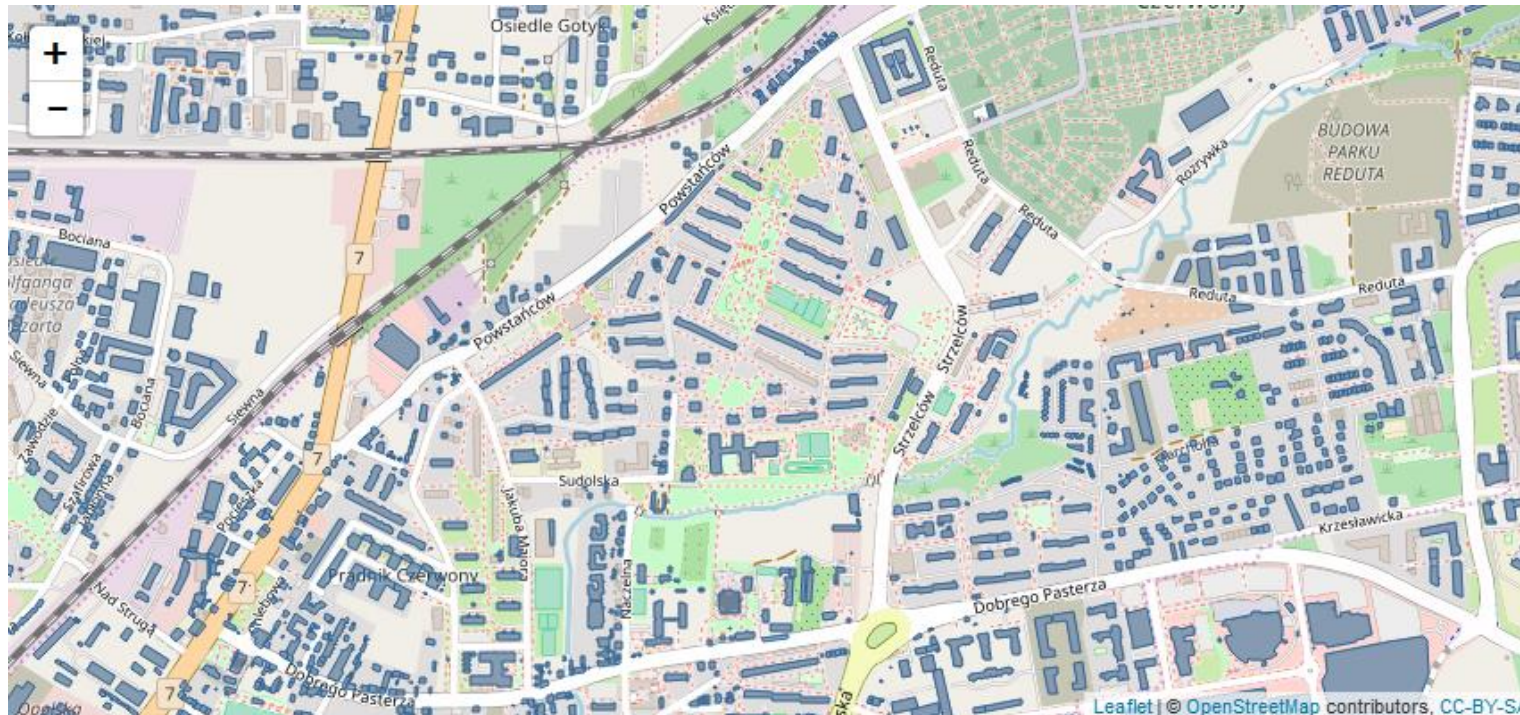
```
m %>% addProviderTiles(providers$Stamen.Toner)
```

```
m %>% addProviderTiles(providers$Esri.NatGeoWorldMap)
```



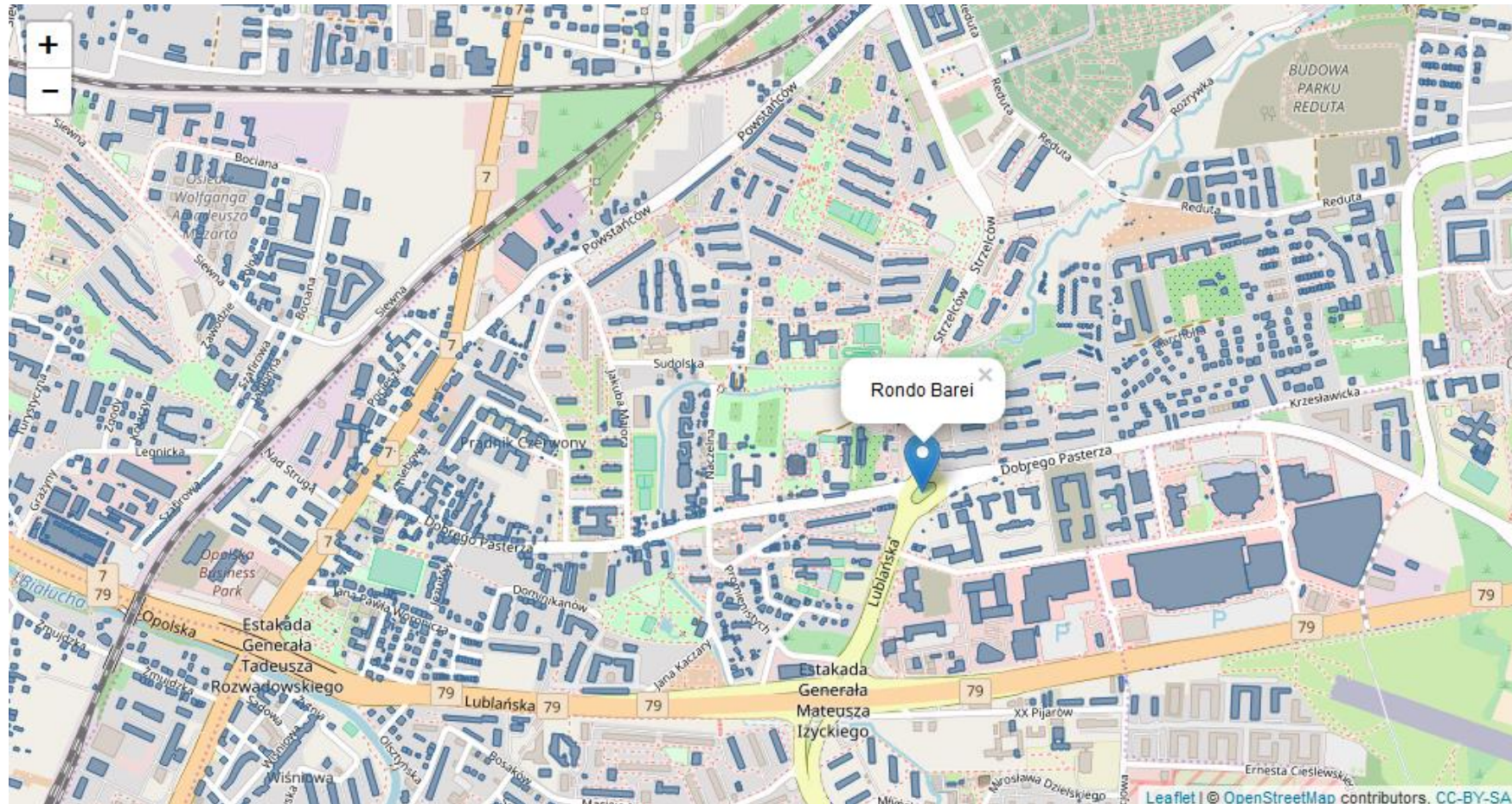
Przygotowanie

- Zebranie odpowiednich bibliotek
- Wgranie podkładu mapowego
- Wgranie danych wektorowych



Przygotowanie

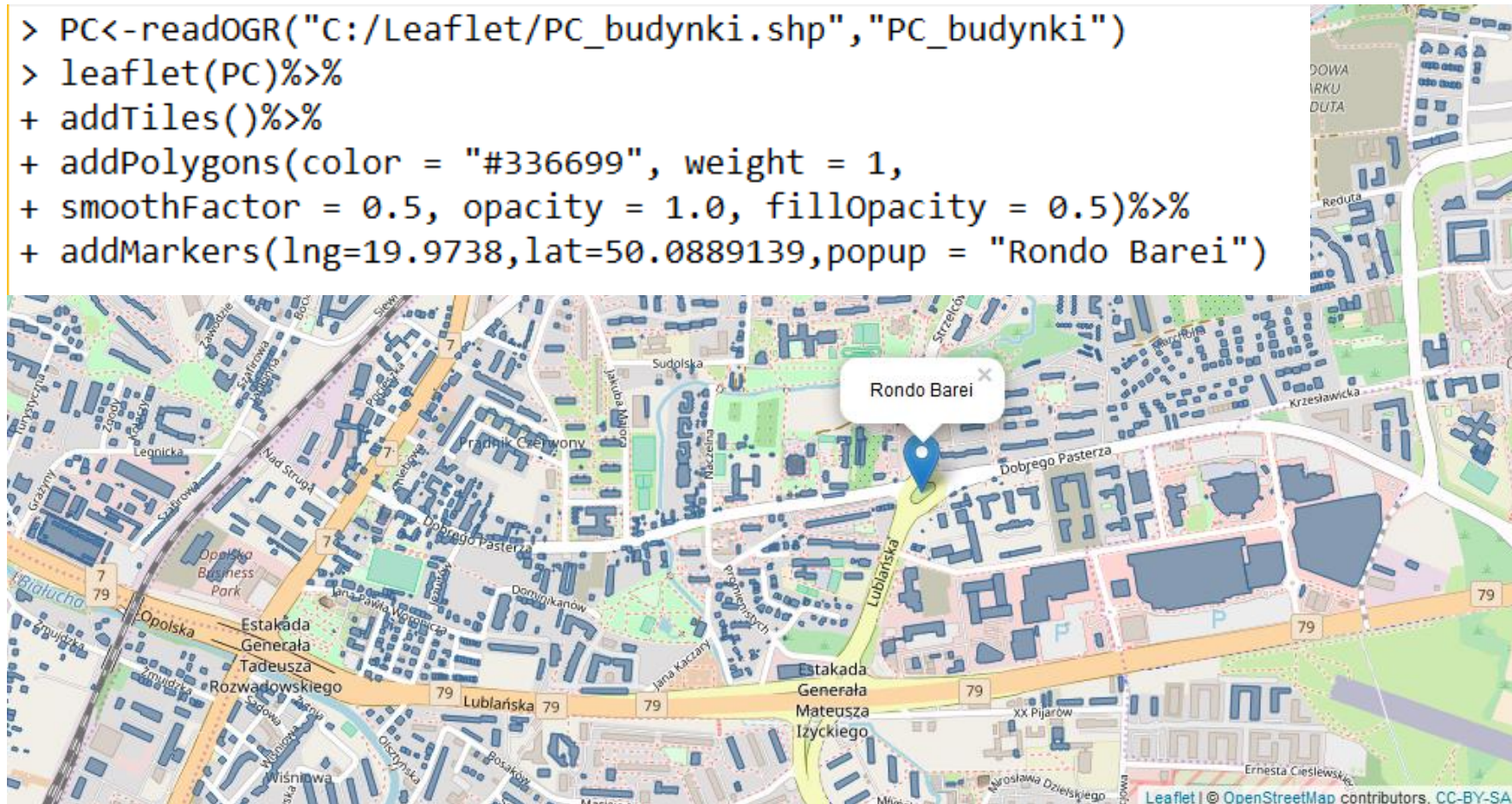
- Wgranie danych wektorowych
- Zaznaczenie interesującego miejsca - addMarkers()



Przygotowanie

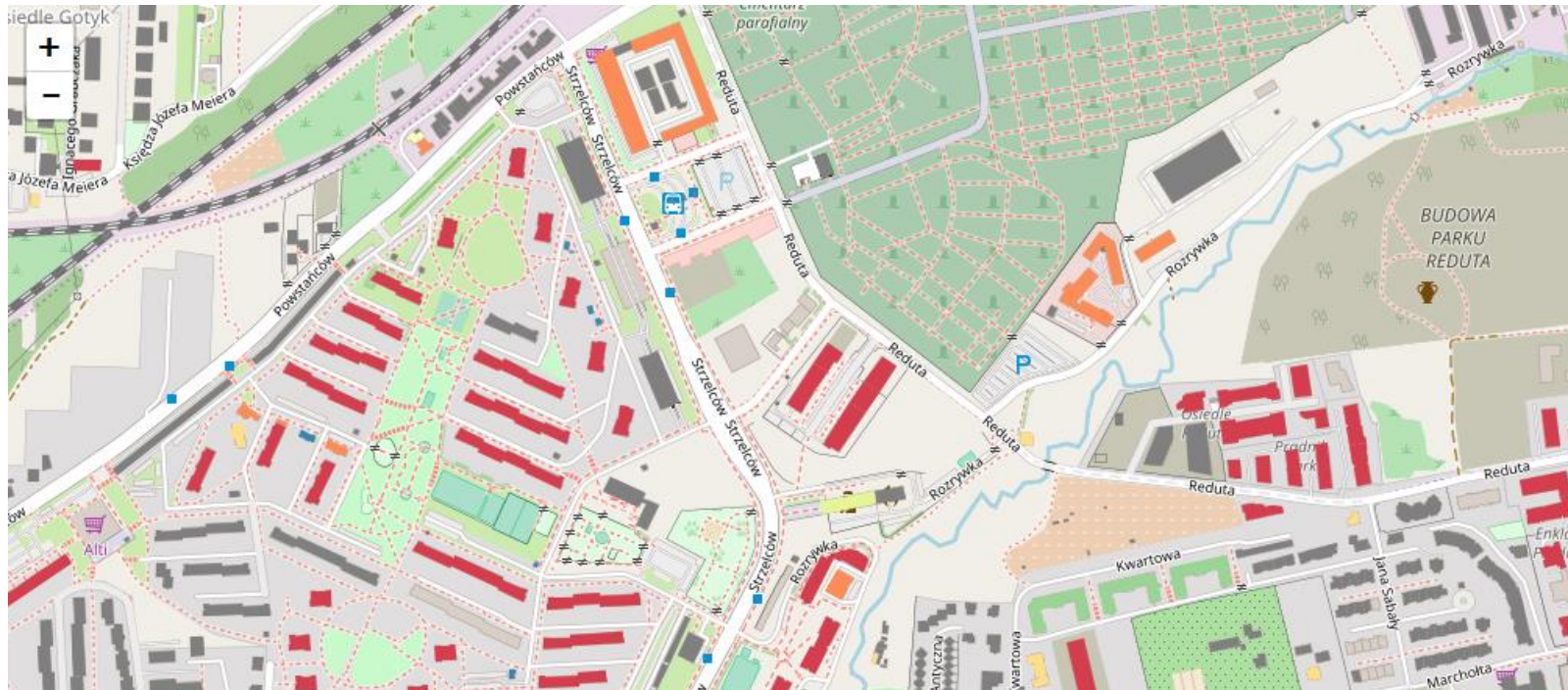
- Wgranie danych wektorowych
- Zaznaczenie interesującego miejsca

```
> PC<-readOGR("C:/Leaflet/PC_budynki.shp", "PC_budynki")  
> leaflet(PC)%>%  
+ addTiles()%>%  
+ addPolygons(color = "#336699", weight = 1,  
+ smoothFactor = 0.5, opacity = 1.0, fillOpacity = 0.5)%>%  
+ addMarkers(lng=19.9738, lat=50.0889139, popup = "Rondo Barei")
```



Przygotowanie

- Wgranie danych wektorowych
- Zapytanie atrybutowe – kolor wg funkcji budynku



```
> pal <- colorFactor("Spectral", domain = Budynki$building)
> map%>%
+ addPolygons(stroke = FALSE, smoothFactor = 0.2, fillOpacity = 1, color = ~pal(building))
```


Przykłady zastosowania

Zaznaczenie epicentrum trzęsienia ziemi wraz z podziałem na kolory wg magnitudy (ostatnie 20 zarejestrowanych)

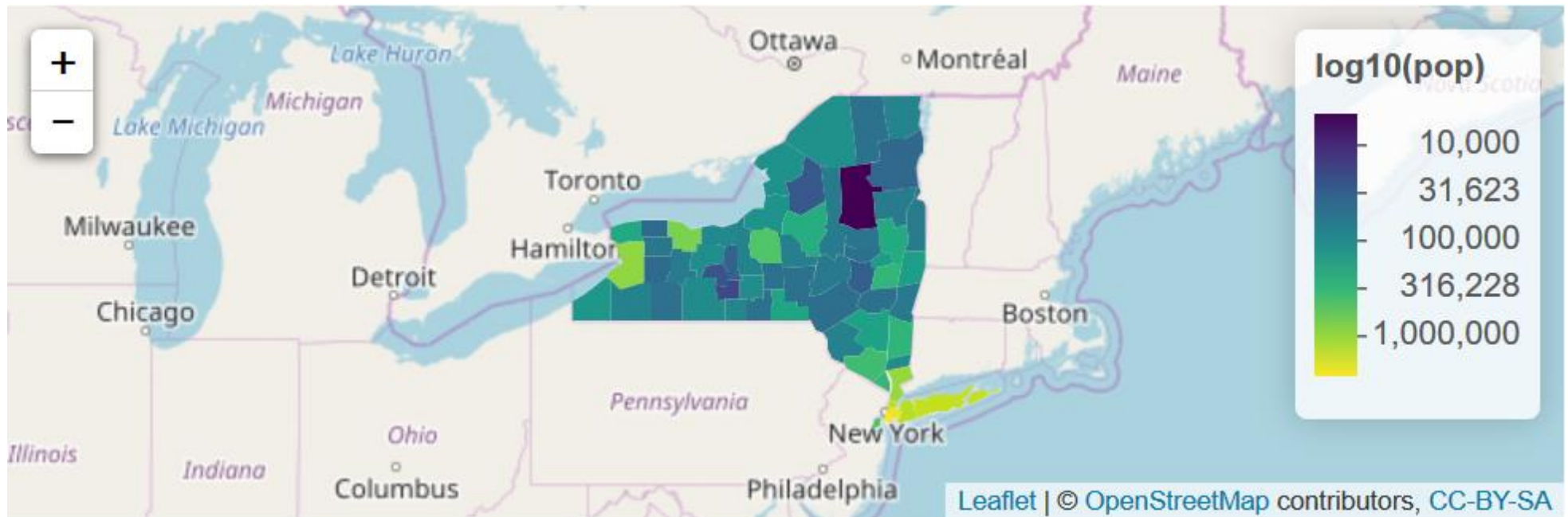


```
df.20 <- quakes[1:20,]
getColor <- function(quakes) { sapply(quakes$mag, function(mag) {
  if(mag <= 4) {"green"} else if(mag <= 5) {"orange"} else {"red"}})}
icons <- awesomeIcons(
  icon = 'ios-close',
  iconColor = 'black',
  library = 'ion',
  markerColor = getColor(df.20))
leaflet(df.20) %>% addTiles() %>%
  addAwesomeMarkers(~long, ~lat, icon=icons, label=~as.character(mag))
```

© OpenStreetMap contributors, CC-BY-SA

Przykłady zastosowania

Ukazanie liczby ludności hrabstw w stanie Nowy Jork



```
nycounties <- geojsonio::geojson_read("json/nycounties.geojson", what = "sp")
pal <- colorNumeric("viridis", NULL)
leaflet(nycounties) %>%
  addTiles() %>%
  addPolygons(stroke = FALSE, smoothFactor = 0.3, fillOpacity = 1, fillColor = ~pal(log10
(pop)), label = ~paste0(county, ": ", formatC(pop, big.mark = ","))) %>% addLegend(pal =
pal, values = ~log10(pop), opacity = 1.0, labFormat = labelFormat(transform = function(x)
round(10^x)))
```

Przykłady zastosowania

Ukazanie dnia i nocy w czasie rzeczywistym (~9:00)



```
> leaflet() %>% addTiles() %>% addTerminator()
```


Przykłady zastosowania

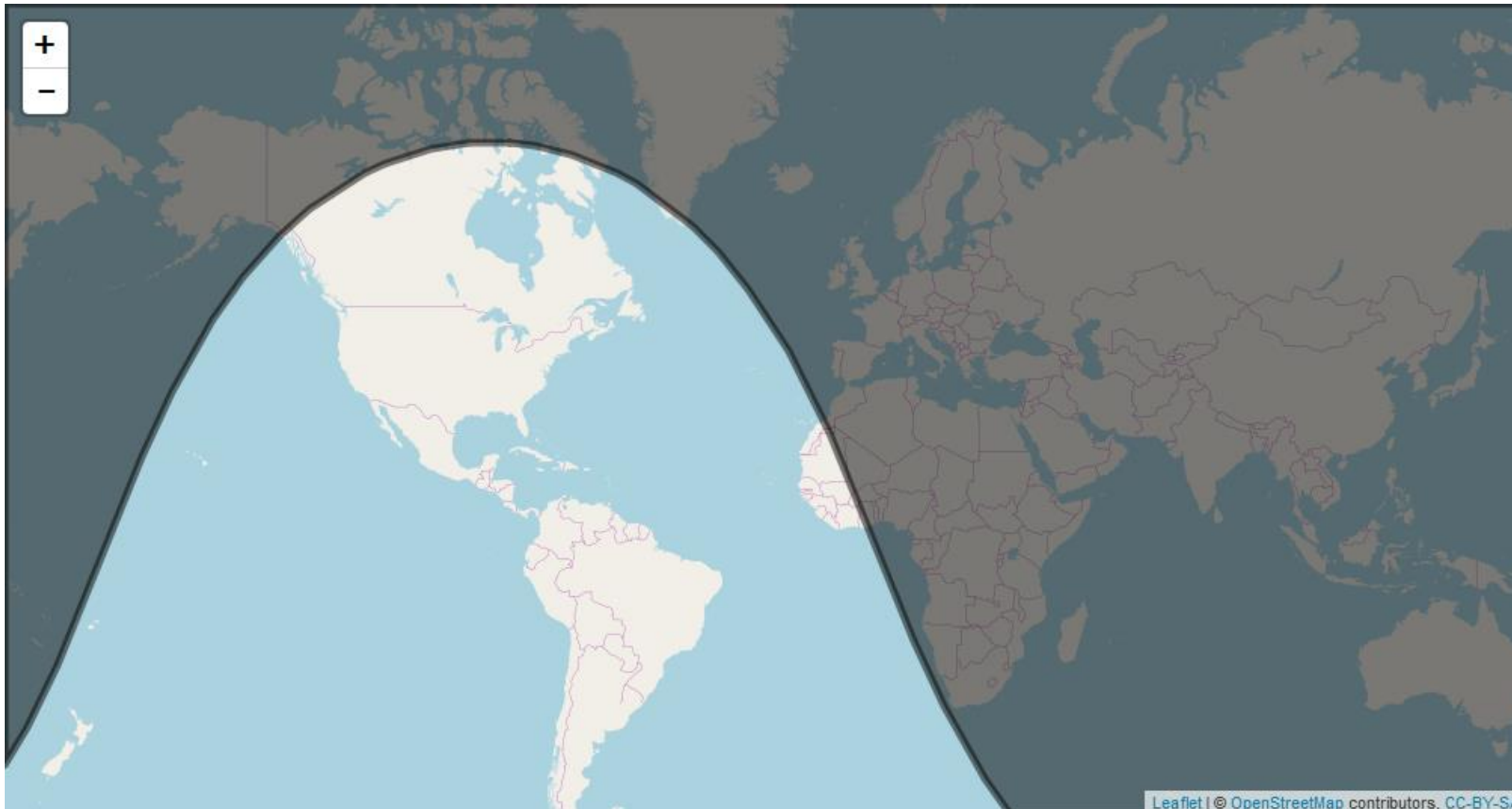
Ukazanie dnia i nocy w czasie rzeczywistym (11:30)



```
> leaflet() %>% addTiles() %>% addTerminator()
```

Przykłady zastosowania

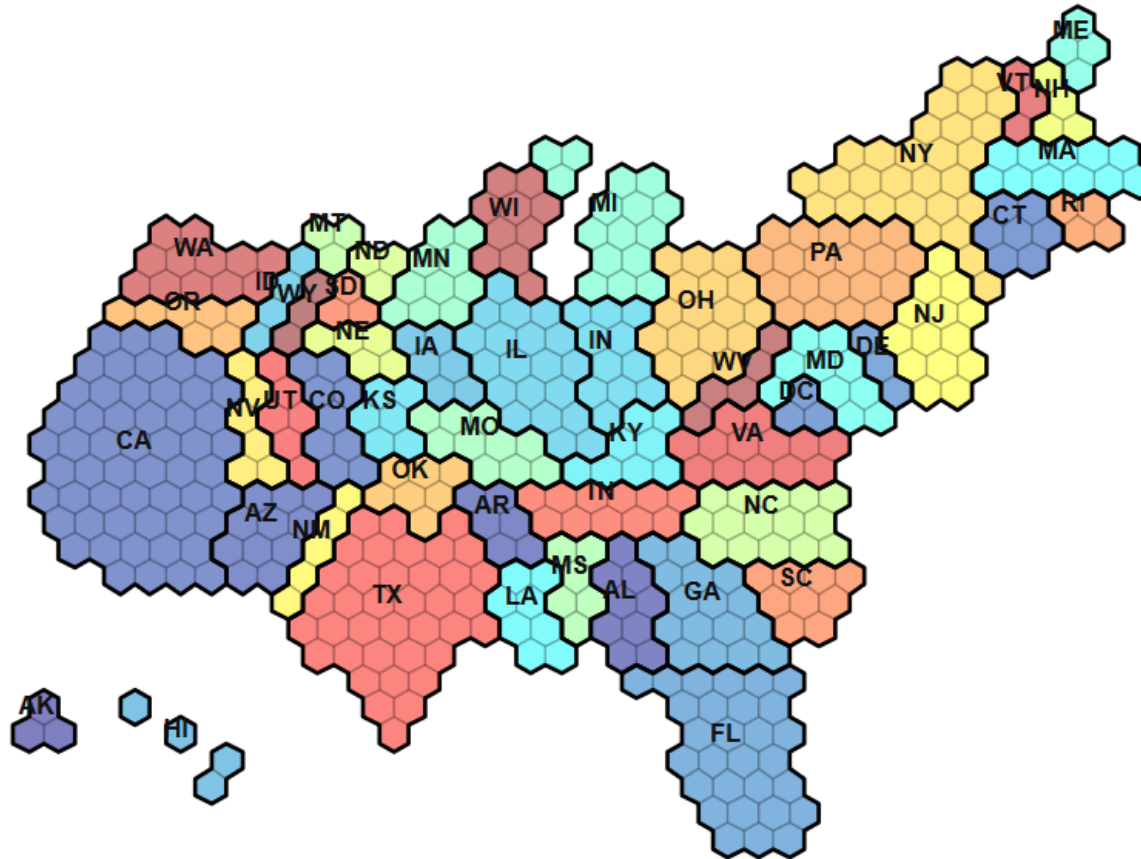
Ukazanie dnia i nocy w czasie rzeczywistym (19:00)



```
> leaflet() %>% addTiles() %>% addTerminator()
```

Przykłady zastosowania

Przestawianie map w różnych odwzorowaniach – mapa USA z skalowaniem wielkości stanów wg liczby elektorów



Przykłady zastosowania

Przestawianie map w różnych odwzorowaniach

```
l <- leaflet(  
  options=leafletOptions(  
    crs = leafletCRS("L.CRS.Simple"),  
    minZoom = -2, maxZoom = -2,  
    dragging = FALSE, zoomControl = FALSE, attributionControl = FALSE)) %>%  
  addPolygons(  
    data=FiveThirtyEightElectoralCollege,  
    weight=1,color='#000000', fillOpacity = 0.5, opacity=0.2,  
    fillColor= ~factpal(state)) %>%  
  addPolygons(  
    data=FiveThirtyEightElectoralCollege.states, group = 'states',  
    weight=2,color='#000000',  
    fill = T, opacity = 1, fillOpacity = 0,  
    highlightOptions = highlightOptions(weight = 4)) %>%  
  addLabelOnlyMarkers(  
    data=FiveThirtyEightElectoralCollege.centers,  
    label = ~as.character(state),  
    labelOptions = labelOptions(  
      noHide = 'T', textOnly = T,  
      offset=c(-8,-20), textsize = '12px'))
```

Bibliografia

- <https://www.rstudio.com/resources/videos/mapping-in-r-with-leaflet/>
- <https://rstudio.github.io/leaflet/>
- OpenStreetMaps
- Własny kod