

```
library(sp)
library("readxl")
library(spatstat)
```

```
## Ładowanie wymaganego pakietu: spatstat.data
```

```
## Ładowanie wymaganego pakietu: spatstat.geom
```

```
## spatstat.geom 2.3-0
```

```
## Ładowanie wymaganego pakietu: spatstat.core
```

```
## Ładowanie wymaganego pakietu: nlme
```

```
## Ładowanie wymaganego pakietu: rpart
```

```
## spatstat.core 2.3-1
```

```
## Ładowanie wymaganego pakietu: spatstat.linnet
```

```
## spatstat.linnet 2.3-0
```

```
##
## spatstat 2.2-0      (nickname: 'That's not important right now')
## For an introduction to spatstat, type 'beginner'
```

```
library(dbscan)
library(tidyverse)
```

```
## Registered S3 method overwritten by 'cli':
## method      from
## print.boxx  spatstat.geom
```

```
## ——— Attaching packages ——— tidyverse 1.3.1 ———
```

```
## ✓ ggplot2 3.3.5   ✓ purrr  0.3.4
## ✓ tibble  3.1.5   ✓ dplyr  1.0.7
## ✓ tidyr   1.1.4   ✓ stringr 1.4.0
## ✓ readr   2.0.2   ✓ forcats 0.5.1
```

```
## ——— Conflicts ——— tidyverse_conflicts() ———
## x dplyr::collapse() masks nlme::collapse()
## x dplyr::filter()   masks stats::filter()
## x dplyr::lag()      masks stats::lag()
```

```
library(ggplot2)
library(raster)
```

```
##
## Dołączanie pakietu: 'raster'
```

```
## Następujący obiekt został zakryty z 'package:dplyr':
##
## select
```

```
## Następujący obiekt został zakryty z 'package:nlme':
##
## getData
```

Ndadanie wczytanym danym układu współrzędnych ETRS 1989 Poland CS2000 Zone 7 - EPSG:2178

```
data <- read_excel("zestaw8.xlsx", col_types = c("numeric", "numeric"))
head(data)
```

```
## # A tibble: 6 × 2
##   Long  Lat
##   <dbl> <dbl>
## 1 19.9 50.1
## 2 20.0 50.1
## 3 20.0 50.1
## 4 19.9 50.1
## 5 19.9 50.1
## 6 20.0 50.1
```

```
dim(data)
```

```
## [1] 2000  2
```

```
coord <- SpatialPoints(cbind(data$Long, data$Lat), proj4string = CRS("+proj=longlat"))
coordUTM <- spTransform(coord, CRS("+init=epsg:2178"))
```

```
## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO", prefer_proj = prefer_proj): Discarded datum Unknown based on GRS80 ellipsoid in Proj4 definition,
## but +towgs84= values preserved
```

```
dataUTM <- data.frame(coordUTM)
head(dataUTM)
```

```
##   coords.x1 coords.x2
## 1 7423881 5547947
## 2 7430823 5549041
## 3 7425816 5547329
## 4 7423685 5548121
## 5 7423867 5547753
## 6 7429003 5551706
```

```
colnames(dataUTM) <- c("Lon", "Lat")
write.csv2(dataUTM, "data_out.csv", row.names = F)
```

## Wczytanie danych po zmianie układu współrzędnych oraz osiedla.shp

```
data <- read.csv2("data_out.csv")
data_dist <- shapefile("osiedla.shp")
```

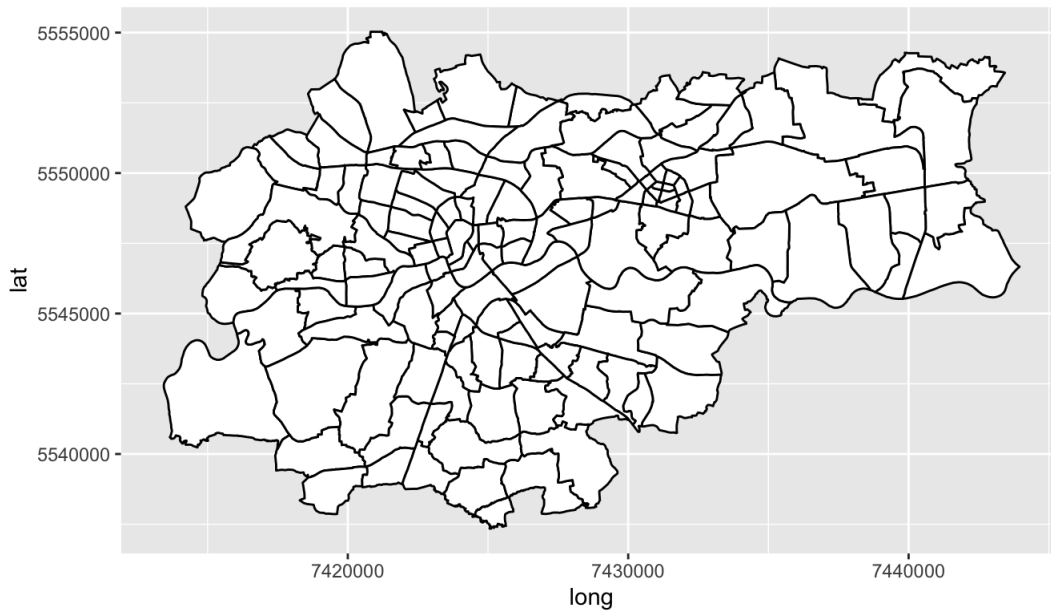
```
## Warning in OGRSpatialRef(dsn, layer, morphFromESRI = morphFromESRI, dumpSRS = dumpSRS, : Discarded datum European_Terrestrial_Reference_System_1989 in Proj4 definition: +proj=tmerc +lat_0=0 +lon_0=21 +k=0.999923 +x_0=7500000 +y_0=0 +ellps=GRS80 +towgs84=0,0,0,0,0,0 +units=m +no_defs,
## but +towgs84= values preserved
```

## Stworzenie mapy Krakowa z podziałem na dzielnice

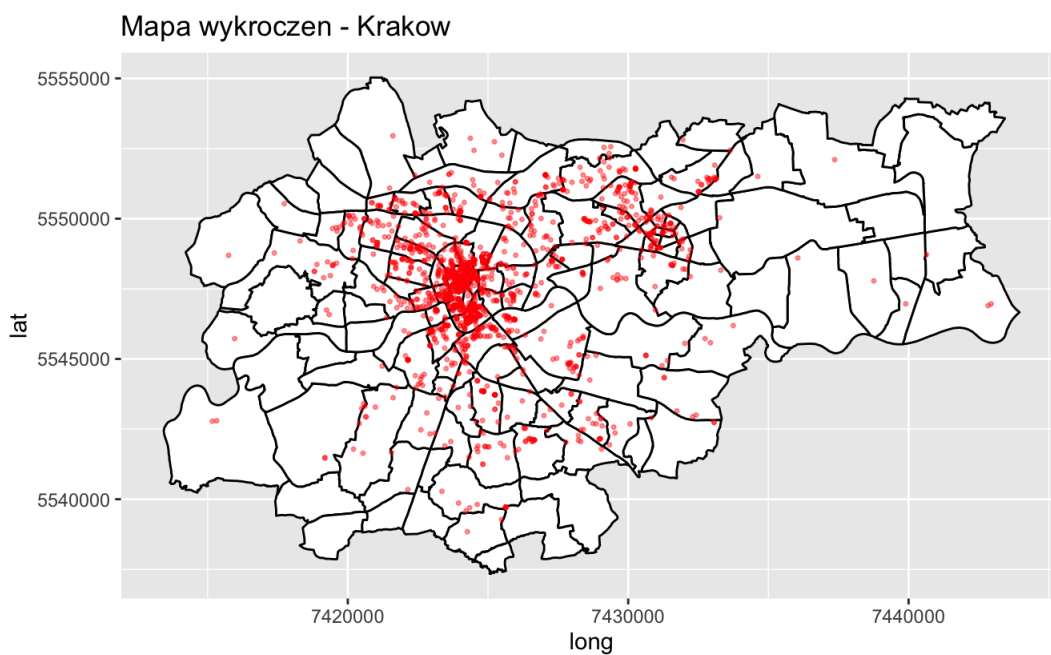
```
cra_map <- ggplot() + geom_polygon(data=data_dist, aes(x = long, y = lat, group = group),
  show.legend = FALSE, color = "black", fill = "white") + coord_fixed()
```

```
## Regions defined for each Polygons
```

```
cra_map
```



```
cra_map + geom_point(data=data, aes(x=Lon, y=Lat), alpha=0.4, size = 0.6, colour="red") +  
  ggtitle("Mapa wykroczen - Krakow")
```



## DBSCAN

- opiera sie na dwóch parametrach wejściowych:

\* epsilon (Eps) – promień sąsiedztwa

\* minPts – minimalna liczba obserwacji potrzebna by wybrana obserwacja została uznana za punkt centralny danej grupy # (punkt centralny również jest liczony).

Działanie algorytmu:

- dla kazdej obserwacji znajdujemy jej sąsiadów

- kazda obserwacja ktora ma co najmniej MinPts sąsiadów w odległości mniejszej niż epsilon to punkt centralny

- wszystkie obserwacje spełniające warunki z powyższego punktu łączone są w grupy

- obserwacje które znajdują się w odległości epsilon, a nie są punktami centralnymi, zostają przyłączone do istniejących grup

- obserwacje, które należą do grup, lecz w ich zasięgu epsilon nie znajduje się żadna nowa obserwacja,

nazywane są obserwacjami granicznymi danej grupy

- wszystkie obserwacje, które nie zostały przyłączone do żadnej z grup, stają się obserwacjami odstającymi.

## Zalety:

- odporny na wpływ obserwacji odstających
- dobrze radzi sobie z grupami o niewypukłym kształcie
- szybkie działanie i relatywnie dobre wyniki
- daje możliwość definiowania wielu miar odległości

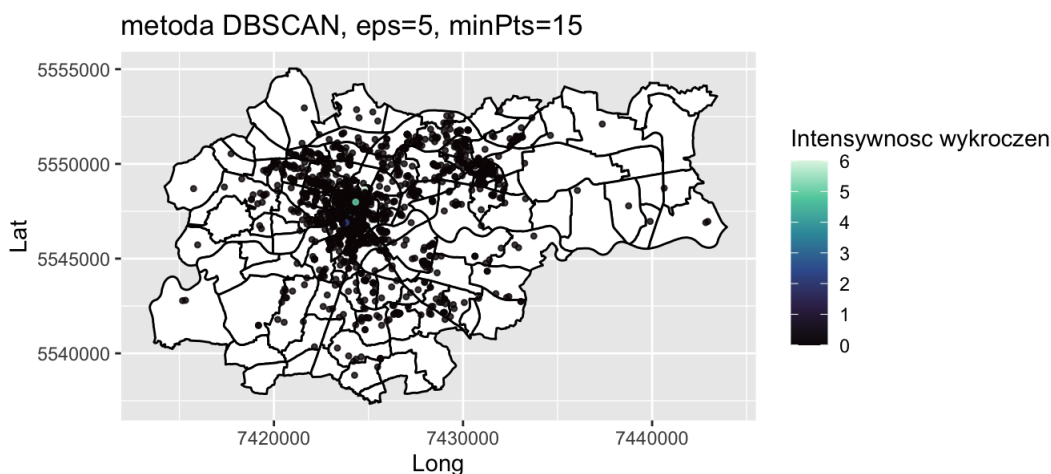
## Wady:

- nie daje możliwości definiowania a priori liczby segmentów – liczba segmentów zależy od liczby obserwacji i dobranych parametrów

- dobór odpowiednich parametrów bywa dość problematyczny – optymalizacja bywa długa i uciążliwa

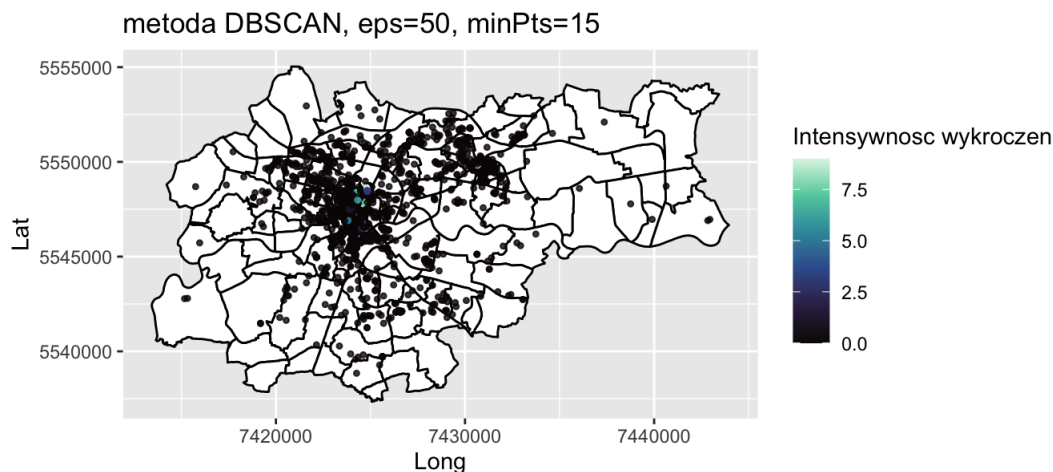
dla wartości:  $\text{eps} = 5$ ,  $\text{minPts} = 15$

```
dbscan_1 <- dbscan(data, eps=5, minPts = 15)
cra_map + geom_point(data=data, aes(x=Lon, y=Lat, color = dbscan_1$cluster), alpha=0.8, size = 0.9) +
  scale_colour_viridis_c(option = "mako", name = "Intensywnosc wykroczen") + ggtitle("metoda DBSCAN, eps=5, minPts=15") +
  labs(x = "Long", y = "Lat")
```



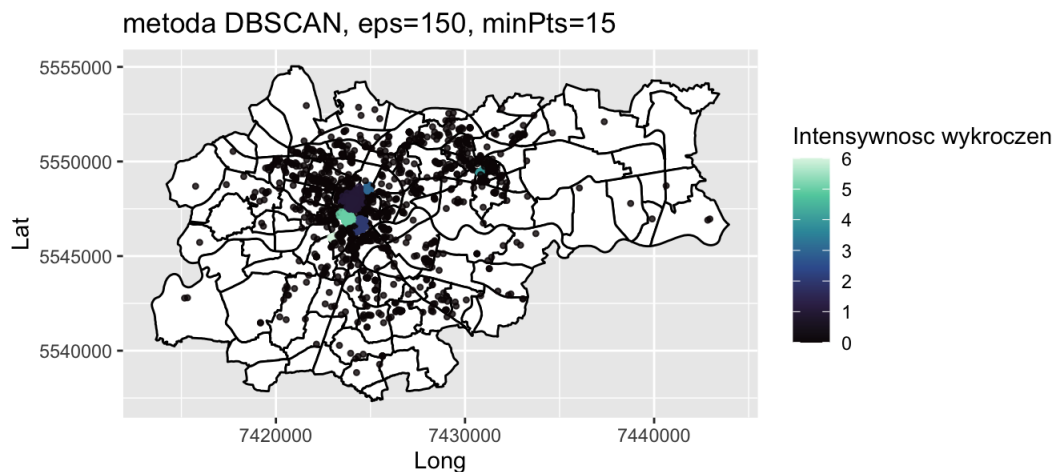
dla wartości:  $\text{eps} = 50$ ,  $\text{minPts} = 15$

```
dbscan_2 <- dbscan(data, eps = 50, minPts = 15)
cra_map + geom_point(data=data, aes(x=Lon, y=Lat, color = dbscan_2$cluster), alpha=0.8, size = 0.9) +
  scale_colour_viridis_c(option = "mako", name = "Intensywnosc wykroczen") + ggtitle("metoda DBSCAN, eps=50, minPts=15") +
  labs(x = "Long", y = "Lat")
```



dla wartosci: eps=30, minPts = 5

```
dbscan_3 <- dbscan(data, eps=150, minPts = 15)
cra_map + geom_point(data=data, aes(x=Lon, y=Lat, color = dbscan_3$cluster), alpha=0.8, size = 0.9) +
  scale_colour_viridis_c(option = "mako", name = "Intensywnosc wykroczen") + ggtitle("metoda DBSCAN, eps=150, minPts=15") +
  labs(x = "Long", y = "Lat")
```



## Wnioski:

Wraz ze zwiększaniem wartosci promienia(epsilon) rosnie ilosc pojawiajacych sie klastrow na mapie, szczególnie w centrum Krakowa, poniewaz tam zagesczenie punktow jest najwieksze

## HDBSCAN

Rozszerza algorytm DBSCAN, przekształcając go w hierarchiczny algorytm grupowania, # a następnie wykorzystuje technikę wyodrębniania płaskiego klastrowania w oparciu o stabilność klastrow. # Przyjmuje jeden argument minPts - minimalna liczba próbek

## Zalety:

- lepszy niż dbscan dla danych o różnej gęstości

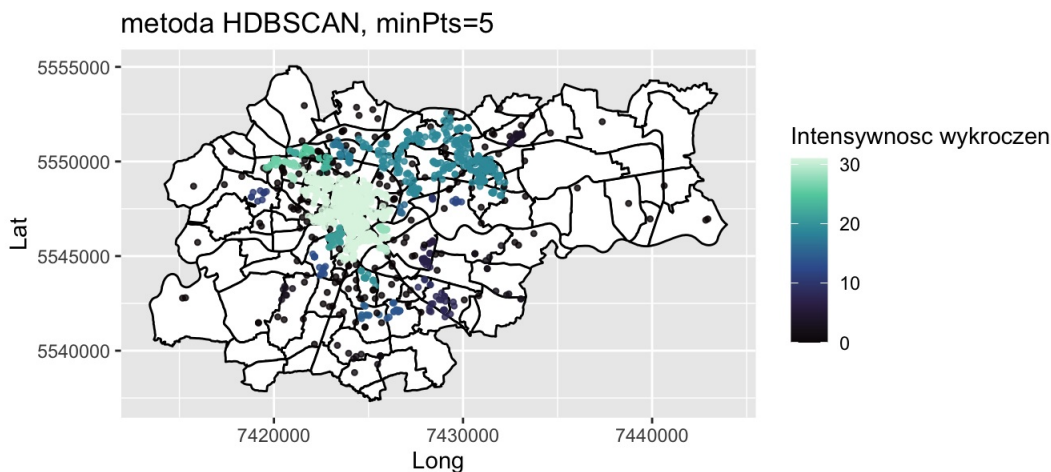
- szybszy niz dbscan

## Wady:

- skomplikowanie algorytmu

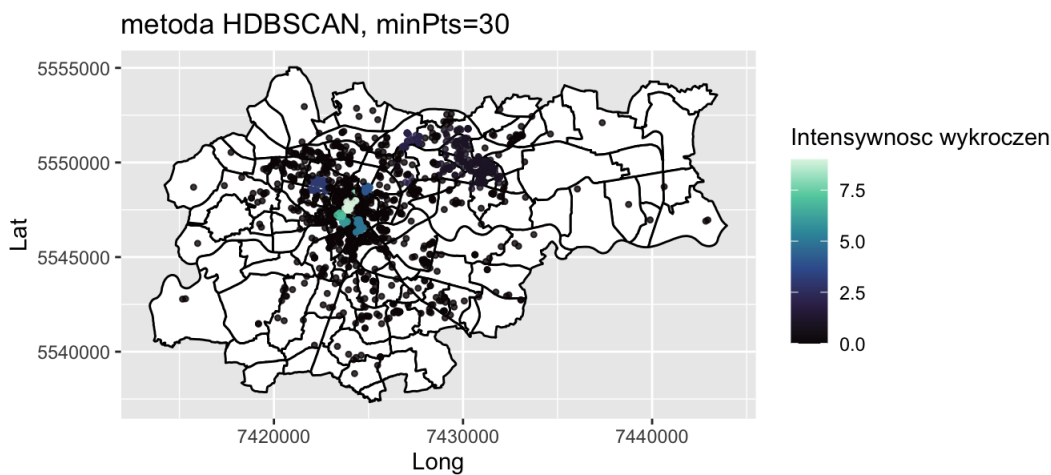
dla wartosci: minPts = 5

```
hdbscan_1 <- hdbscan(data, minPts = 5)
cra_map + geom_point(data=data, aes(x=Lon, y=Lat, color = hdbscan_1$cluster), alpha=0.8, size = 0.9) +
  scale_colour_viridis_c(option = "mako", name = "Intensywnosc wykroczen") + ggtitle("metoda HDBSCAN, minPts=5") +
  labs(x = "Long", y = "Lat")
```



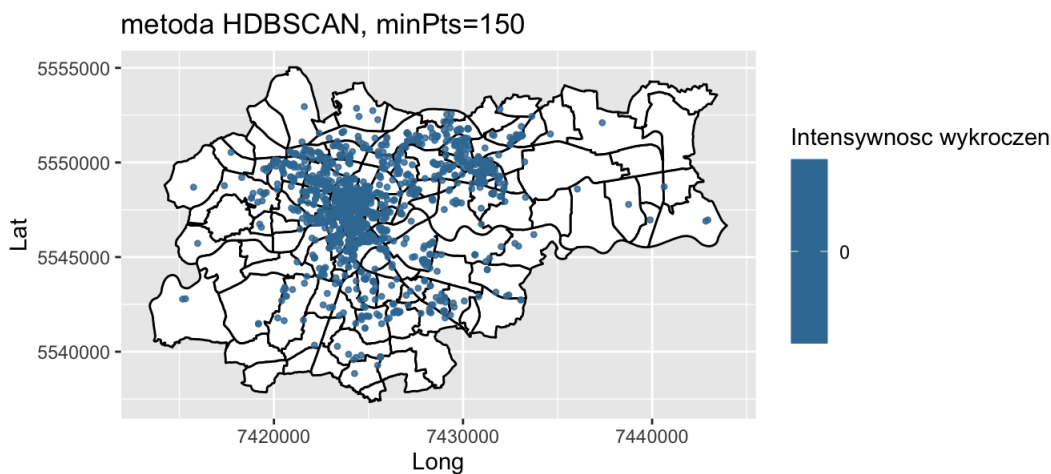
dla wartosci: minPts = 10

```
hdbscan_2 <- hdbscan(data, minPts = 30)
cra_map + geom_point(data=data, aes(x=Lon, y=Lat, color = hdbscan_2$cluster), alpha=0.8, size = 0.9) +
  scale_colour_viridis_c(option = "mako", name = "Intensywnosc wykroczen") + ggtitle("metoda HDBSCAN, minPts=30") +
  labs(x = "Long", y = "Lat")
```



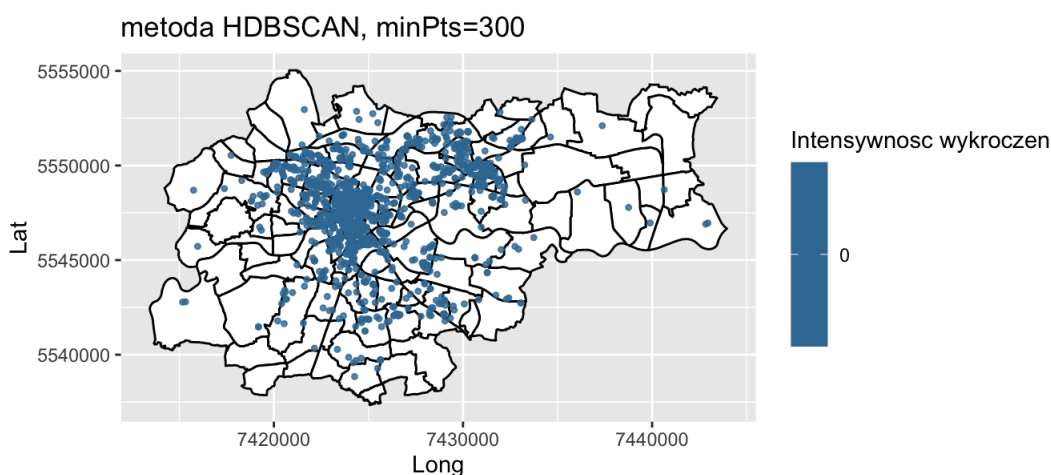
dla wartosci: minPts = 30

```
hdbscan_3 <- hdbscan(data, minPts = 150)
cra_map + geom_point(data=data, aes(x=Lon, y=Lat, color = hdbscan_3$cluster), alpha=0.8, size = 0.9) +
  scale_colour_viridis_c(option = "mako", name = "Intensywnosc wykroczen") + ggtitle("metoda HDBSCAN, minPts=150") +
  labs(x = "Long", y = "Lat")
```



dla wartosci: minPts = 300

```
hdbscan_4 <- hdbscan(data, minPts = 300)
cra_map + geom_point(data=data, aes(x=Lon, y=Lat, color = hdbscan_4$cluster), alpha=0.8, size = 0.9) +
  scale_colour_viridis_c(option = "mako", name = "Intensywnosc wykroczen") + ggtitle("metoda HDBSCAN, minPts=300") +
  labs(x = "Long", y = "Lat")
```



## Wnioski:

Wraz ze zwiekszaniem wartosci minPts maleje liczba kalstrow, poniewaz dla wartosci minPts=5 tworzy sie kilka kalstrow rozlozonych sotunkowo rownomiennie wokol centrum Krakowa, natomiast im wartosc minPts jest wieksza tym kalstry mniejsze kalstry sa bardziej skupione w centrum.

## OPTICS

Podstawa dzialania algorytmu jest podobna do DBSCAN, jednek usuwa on powazna wade dzialania tego

algorytmu - nie ma problemów z rozpoznawaniem klastrow o bardzo roznej gestosci, rownoczesnie znacznie zmniejszana jest wrazlowosc na parametry epsilon oraz minPts, ktore rowniez wystepuja.

## Zalety:

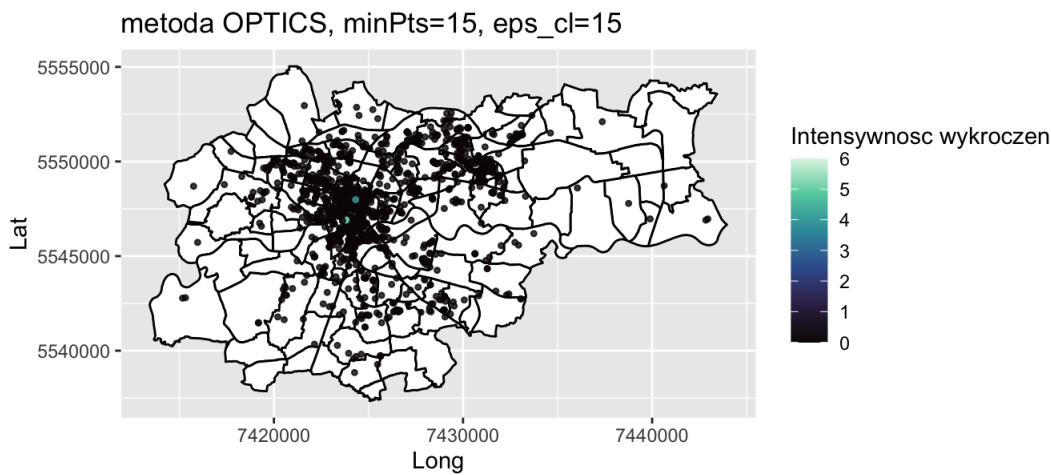
- brak koniecznosci nadawania z gory wielkosci promienia
- nie wymaga parametrow gestosci

## Wady:

- algorytm zle radzi sobie z wielowymiarowymi danymi
- tworzy tylko porzadek klastrowy

dla wartosci: minPts = 15, eps\_cl = 15

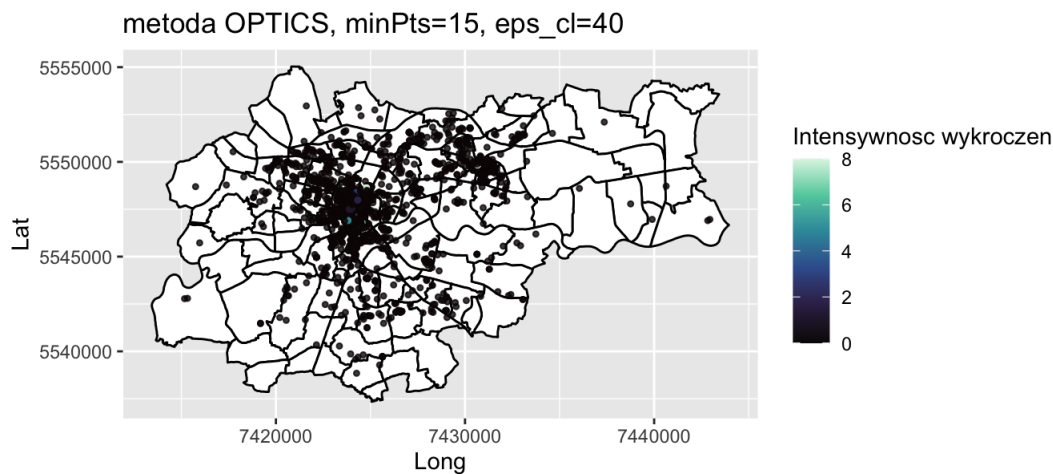
```
out_1 <- optics(data, minPts = 15)
optics_1 <- extractDBSCAN(out_1, eps_cl = 15)
cra_map + geom_point(data=data, aes(x=Lon, y=Lat, color = optics_1$cluster), alpha=0.8, size = 0.9) +
  scale_colour_viridis_c(option = "mako", name = "Intensywnosc wykroczen") + ggtitle("metoda OPTICS, minPts=15, eps_cl=15") +
  labs(x = "Long", y = "Lat")
```



dla wartosci: minPts = 15, eps\_cl = 40

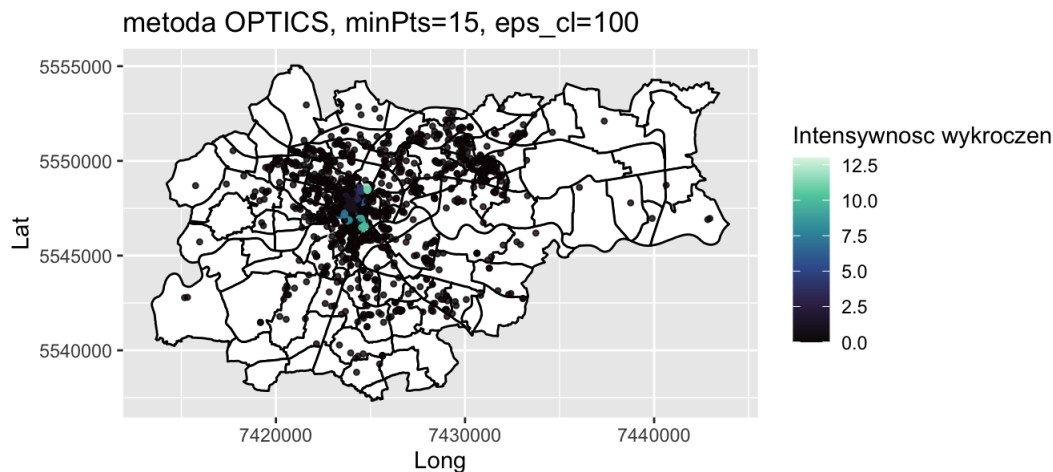
```
out_2 <- optics(data, minPts = 15)
optics_2 <- extractDBSCAN(out_2, eps_cl = 40)
cra_map + geom_point(data=data, aes(x=Lon, y=Lat, color = optics_2$cluster), alpha=0.8, size = 0.9) +
  scale_colour_viridis_c(option = "mako", name = "Intensywnosc wykroczen") + ggtitle("metoda OPTICS, minPts=15, eps_cl=40") +
  labs(x = "Long", y = "Lat")
```





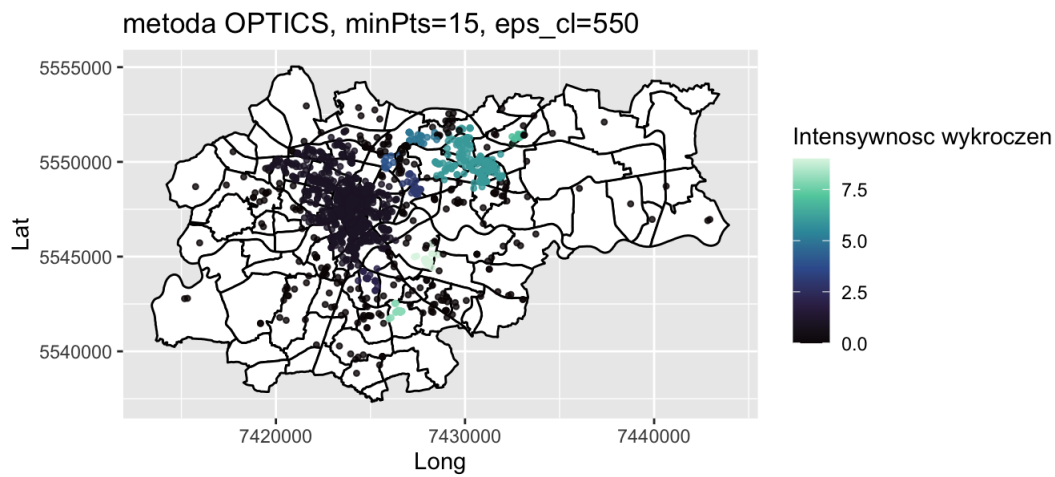
dla wartosci: minPts = 15, eps\_cl = 100

```
out_3 <- optics(data, minPts = 15)
optics_3 <- extractDBSCAN(out_3, eps_cl = 100)
cra_map + geom_point(data=data, aes(x=Lon, y=Lat, color = optics_3$cluster), alpha=0.8, size = 0.9) +
  scale_colour_viridis_c(option = "mako", name = "Intensywnosc wykroczen") + ggtitle("metoda OPTICS, minPts=15, eps_cl=100") +
  labs(x = "Long", y = "Lat")
```



dla wartosci: minPts = 15, eps\_cl = 550

```
out_4 <- optics(data, minPts = 15)
optics_4 <- extractDBSCAN(out_4, eps_cl = 550)
cra_map + geom_point(data=data, aes(x=Lon, y=Lat, color = optics_4$cluster), alpha=0.8, size = 0.9) +
  scale_colour_viridis_c(option = "mako", name = "Intensywnosc wykroczen") + ggtitle("metoda OPTICS, minPts=15, eps_cl=550") +
  labs(x = "Long", y = "Lat")
```



## Wnioski:

Zmienialem jedynie wartosc dla epsilon - minPts = 15 (stale), liczba klastrow rozlozonych wokol centrum miasta rosla, najmniej klastrow, skupinyh w scislum centrum bylo dla najmniejszej wartosci epsilon=15. Im wieksza wartosc epsilon tym wielkosc klastrow rosla. Najwieksze klaster - w centrum Krakowa oraz w polnocno-wschodniej czesci miasta.