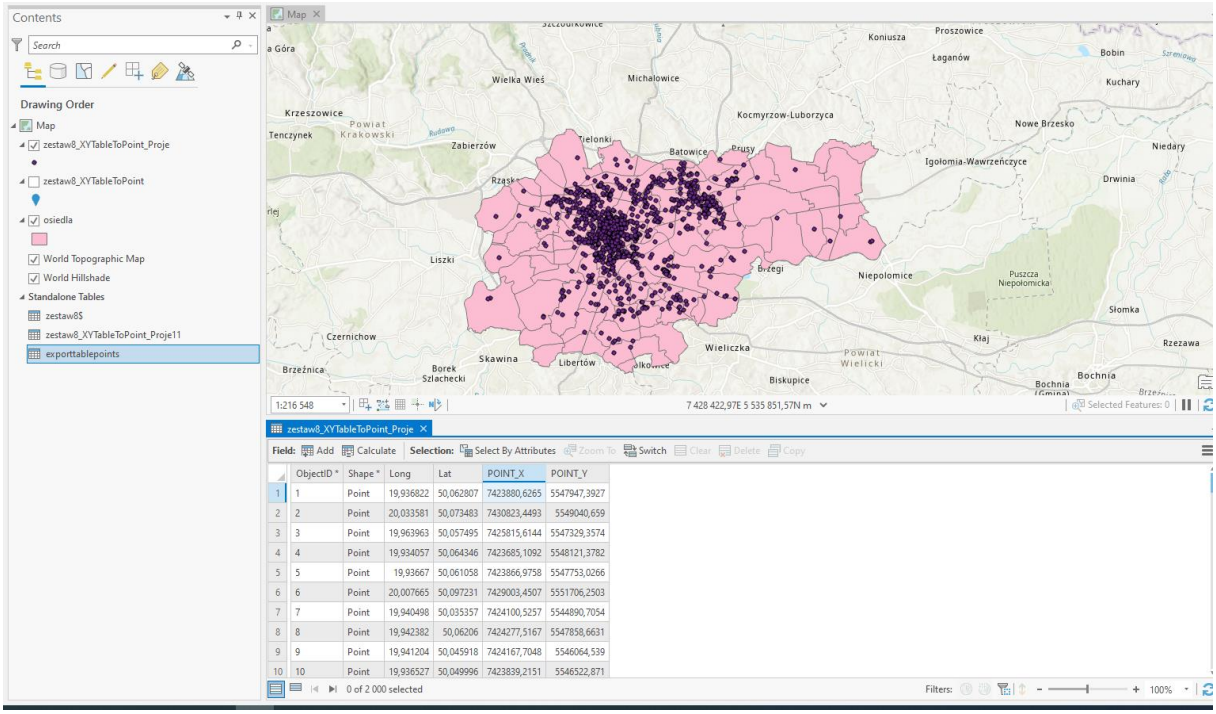
SPRAWOZDANIE

Opis i cel projektu

Celem projektu jest analiza skupień(cluster analysis), czyli eksploracja danych, której celem jest ułożenie obiektów w grupy w taki sposób, aby stopień powiązania obiektów z obiektami należącymi do tej samej grupy był jak największy, a z obiektami z pozostałych grup jak najmniejszy. Dane wykorzystane w projekcie dotyczą wykroczeń na terenie Krakowa.

Współrzędne osiedli oraz współrzędne zgłoszonych wykroczeń przedstawione w jednakowym układzie współrzędnych ETRS 1989 Poland CS2000 Zone 7(ArcGIS Pro)

***Wczytanie danych punktowych i nadanie im układu współrzędnych ETRS 1989 Poland CS2000 Zone 7 zostało zrobione w programie ArcGIS Pro na zajęciach. Dane ze zmienionymi współrzędnymi zostały zapisane do pliku "zestaw8_XYTableToPoint_Proje.shp".



Pakiety, z których korzystałam

```
library("dbscan")
library("sf")
library("devtools")
library("rgdal")
library("ggplot2")
```

Wczytanie danych do R

```
Data<-st_read("zestaw8_XYTableToPoint_Proje.shp")
Data<-st_drop_geometry(Data) #pozbywam się kolumny geometry
Data <- subset( Data, select = -c(1,2) ) #wybieram potrzebne mi kolumny z danymi
class(Data)
```

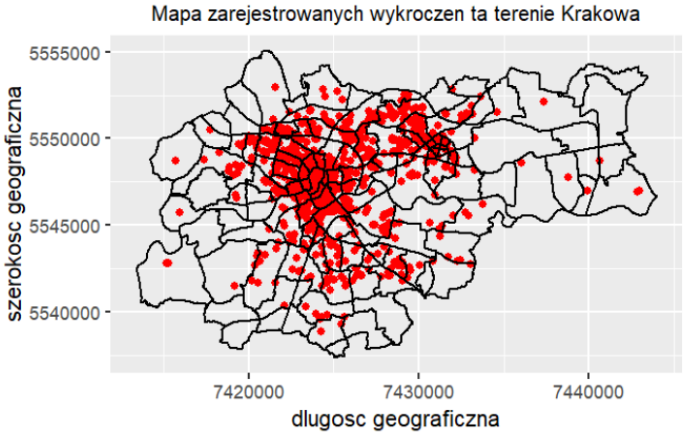
```
> class(Data)
[1] "data.frame"
```

```
head(Data)
```

```
> head(Data)
      Long      Lat POINT_X POINT_Y
1 19.93682 50.06281 7423881 5547947
2 20.03358 50.07348 7430823 5549041
3 19.96396 50.05750 7425816 5547329
4 19.93406 50.06435 7423685 5548121
5 19.93667 50.06106 7423867 5547753
6 20.00766 50.09723 7429003 5551706
```

Mapa dzielnic Krakowa z naniesionymi punktami wykonana w R

```
shp <- readOGR(dsn ="osiedla.shp", stringsAsFactors = F)
map <- ggplot() +
  geom_point(aes(x=Data$POINT_X, y=Data$POINT_Y),colour = 'red')+
  geom_polygon(data = shp, aes(x = long, y = lat, group = group),fill = NA, colour = "black")+
  ggtitle("Mapa zarejestrowanych wykroczen ta terenie Krakowa")+
  theme(plot.title = element_text(size=10, hjust = 0.5))+
  labs(x="dlugosc geograficzna",y="szerokosc geograficzna")+coord_fixed()
map
```



Opis każdego z analizowanych algorytmów klasteryzacji- zasada działania, wady, zalety

algorytm grupowania gęstościowego: DBSCAN

SPOSÓB DZIAŁANIA

Algorytm DBSCAN jest algorytmem zaliczanym do klasy algorytmów gęstościowych.
Szacuje on gęstość wokół każdego punktu danych, licząc liczbę punktów w określonym przez użytkownika sąsiedztwie (eps) i stosuje określone przez użytkownika progi (minPts) do identyfikacji punktów rdzenia, granicy i szumu. W drugim kroku punkty rdzenia są łączone w klaster, jeśli są osiągalne pod względem gęstości (tzn. Istnieje łańcuch punktów rdzeniowych, w których jeden wpada w sąsiedztwo eps następnego). Następnie punkty graniczne są przypisywane do klastrów. Algorytm potrzebuje parametrów: eps(maksymalna odległość osiągalności-promień sąsiedztwa) i minPts (minimalna liczba punktów osiągalności - punktów wymaganych w sąsiedztwie eps)

ZALETY	WADY
• Znakomicie radzi sobie z grupami o niewypukłym kształcie. • Daje dobre rezultaty – zachowując przy tym relatywnie szybkie działanie. • Daje możliwość definiowania wielu miar odległości • Nie wymaga wcześniejszego określenia liczby klastrów • Kolejność punktów w bazie danych jest niewrażliwa • Obsługuj hałas i wartości odstające, jest odporny na wartości odstające i potrafi je wykryć.	• Nie daje możliwości definiowania a priori liczby segmentów – liczba segmentów zależy od liczby obserwacji i dobranych parametrów. • Dobór odpowiednich parametrów bywa dosyć problematyczny – ich optymalizacja bywa długa i uciążliwa, gdyż nie ma jednej sprawdzonej metody • Nie działa dobrze przy dużych różnicach gęstości • Nie nadaje się, gdy występują różne gęstości

algorytm grupowania gęstościowego: HDBSCAN

SPOSÓB DZIAŁANIA

HDBSCAN to algorytm klastrowania, który jest rozszerzoną wersją DBSCAN, konwertując go na hierarchiczny algorytm klastrowania, a następnie używając techniki wyodrębniania płaskiego klastrowania opartego na stabilności klastrów. Wykorzystuje podejście oparte na gęstości-wyszukuje obszary danych o większej gęstości niż otaczająca je przestrzeń.
Jako parametr używa minPts-określa on, jaki rozmiar musi mieć klaster, aby został utworzony

ZALETY	WADY
• jest lepszy dla danych o różnej gęstości • szybszy niż zwykły DBScan • HDBScan ma parametr minPts, który określa, jak duży musi być klaster, aby mógł się utworzyć, jest to bardziej intuicyjne niż korzystanie dodatkowo z parametru eps • pozwala zdefiniować, które klastry są ważne na podstawie rozmiaru	• umieszcza część szumu w klastrach • bardziej skomplikowany w zrozumieniu, przez dużą ilość operacji, które algorytm wykonuje w trakcie działania

algorytm grupowania gęstościowego: OPTICS

SPOSÓB DZIAŁANIA

OPTICS jest algorytmem opartym na idei gęstości. Liniowo porządkuje punkty danych w taki sposób, że punkty, które są najbliższe przestrzennie, stają się sąsiadami w kolejności. Najbliższym analogiem tego uporządkowania jest dendrogram w hierarchicznym klastrowaniu z jednym łączem. Algorytm oblicza również odległość osiągalności dla każdego punktu. ‘plot()’ tworzy wykres osiągalności, który pokazuje każdy punkt odległości osiągalności, gdzie punkty są sortowane według OPTICS. Doliny reprezentują skupiska (im głębsza dolina, tym gęstsza gromada), a wysokie punkty wskazują punkty między gromadami. OPTICS generuje ściśle określone uporządkowanie obiektów, które reprezentuje zagęszczenie elementów ze zbioru danych. Dla każdego obiektu wyznaczana jest tzw. jego wewnętrzna (ang. core-distance) oraz osiągalna odległość (ang. reachability-distance). Na podstawie tych parametrów, generowany i zapisywany jest porządek, w jakim obiekty są przetwarzane. Mimo, że opisywana technika nie generuje wprost podziału na grupy, można w relatywnie prosty sposób podzielić elementy zbioru danych na skupienia.

ZALETY	WADY
• Nie wymaga parametrów gęstości. • Kolejność klastrowania jest użyteczna do wyodrębnienia podstawowych informacji o klastrowaniu.	• Tworzy tylko kolejność klastrów • Nie obsługuje danych o dużych wymiarach

Wyniki klasteryzacji z wnioskami:

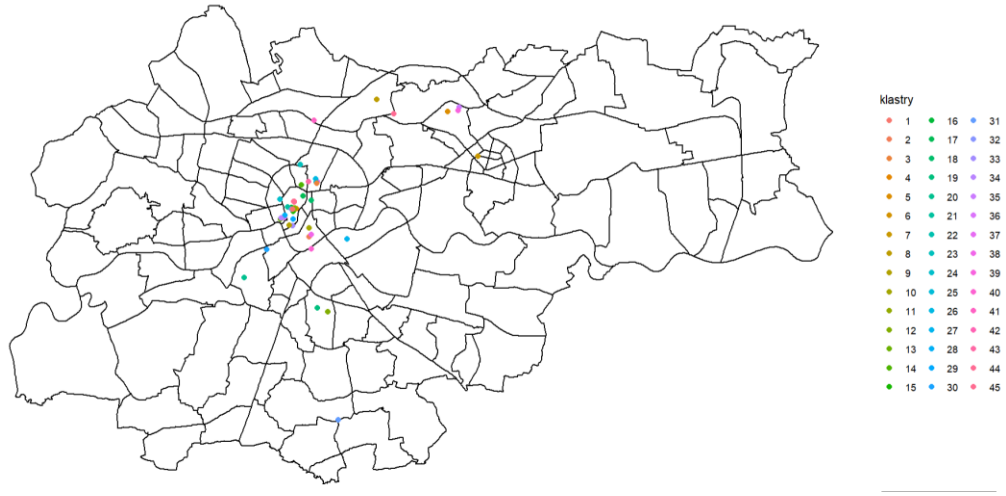
- **DBSCAN**

*eps=10, minPts=5

```
db <- dbscan(Data, eps =10, minPts = 5)
db
```

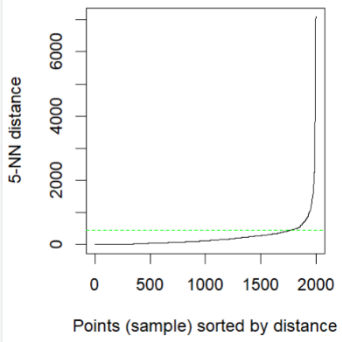
```
Parameters: eps = 10, minPts = 5
The clustering contains 45 cluster(s) and 1597 noise points.

   0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24   25
1597    9    8   13    9    6   18    9    8   25   10    9    7    5   12   20   17   16    9    7    9    5    5    6    7    5
  26   27   28   29   30   31   32   33   34   35   36   37   38   39   40   41   42   43   44   45
  17    8    7   10   12    7    8    5    8    5    5    6    6    6    7    6    5    9    7    5
```

```
cluster_dbscan<- db$cluster
groups_dbscan<-Data[cluster_dbscan != 0,]
noise_dbscan<-Data[cluster_dbscan == 0,]
clusters_dbscan<-as.factor(cluster_dbscan)
color_dbscan<-clusters_dbscan[clusters_dbscan!=0]


ggplot() +
  geom_polygon(data = shp, aes(x = long, y = lat, group = group),fill=NA, color = "black")+
  theme_void()+
  geom_point(aes(groups_dbscan$POINT_X, groups_dbscan$POINT_Y, color = color_dbscan),pch=16,size=2)+
  #geom_point(aes(noise_dbscan$POINT_X, noise_dbscan$POINT_Y),pch=20,size=2,color='grey' )+
  labs(color="klastry")+ggtitle("Wykres klasteryzacji DBSCAN z parametrami eps=10 min Pts=5")+
  theme(plot.title = element_text(size=10, hjust = 0.5))+coord_fixed()
```



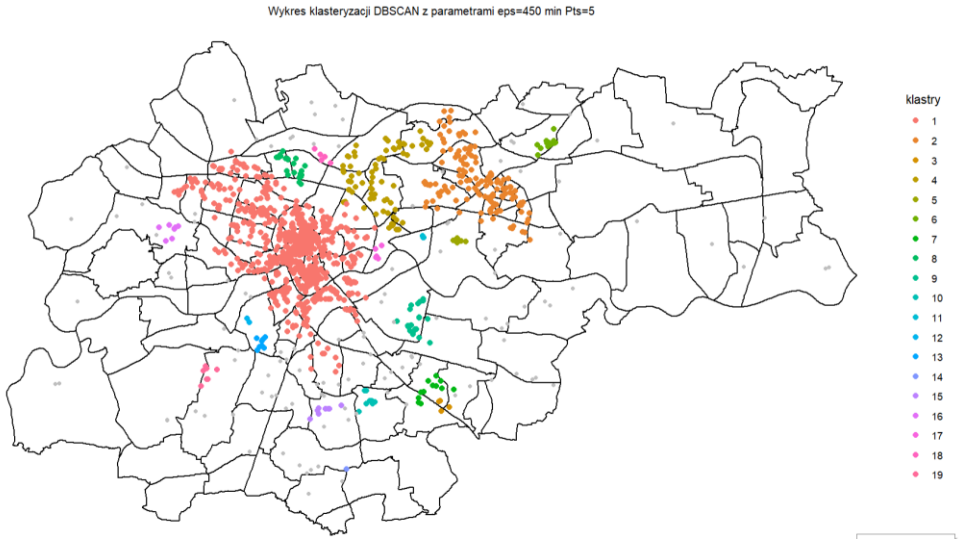
*eps=450, minPts=5

```
> dim(Data)
[1] 2000    2
## Find suitable DBSCAN parameters using KNN:
#k=2*dim-1
#min Pts=k+1
kNNdistplot(Data, k = 5)
abline(h=450, col = "green", lty=2)
```




```
db <- dbscan(Data, eps =450, minPts = 5)
db
#eps - maksymalny promień sąsiedztwa.
#minPts - minimalna liczba obiektów w regionie określonego eps. Domyślnie 5 punktów.


cluster_dbscan<- db$cluster
groups_dbscan<-Data[cluster_dbscan != 0,]
noise_dbscan<-Data[cluster_dbscan == 0,]
clusters_dbscan<-as.factor(cluster_dbscan)
color_dbscan<-clusters_dbscan[clusters_dbscan!=0]


ggplot() +
  geom_polygon(data = shp, aes(x = long, y = lat, group = group),fill=NA, color = "black")+
  theme_void()+
  geom_point(aes(groups_dbscan$POINT_X, groups_dbscan$POINT_Y, color = color_dbscan),pch=16,size=2)+
  geom_point(aes(noise_dbscan$POINT_X, noise_dbscan$POINT_Y),pch=20,size=2,color='grey' )+
  labs(color="klastry")+ggtitle("Wykres klasteryzacji DBSCAN z parametrami eps=450 min Pts=5")+
  theme(plot.title = element_text(size=10, hjust = 0.5))+coord_fixed()
```

```
> db
DBSCAN clustering for 2000 objects.
Parameters: eps = 450, minPts = 5
The clustering contains 19 cluster(s) and 127 noise points.

   0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19
 127 1381  201    6   99    6   19   13   35   26   16    5   10    8    7    9    8    6   11    7

Available fields: cluster, eps, minPts
```

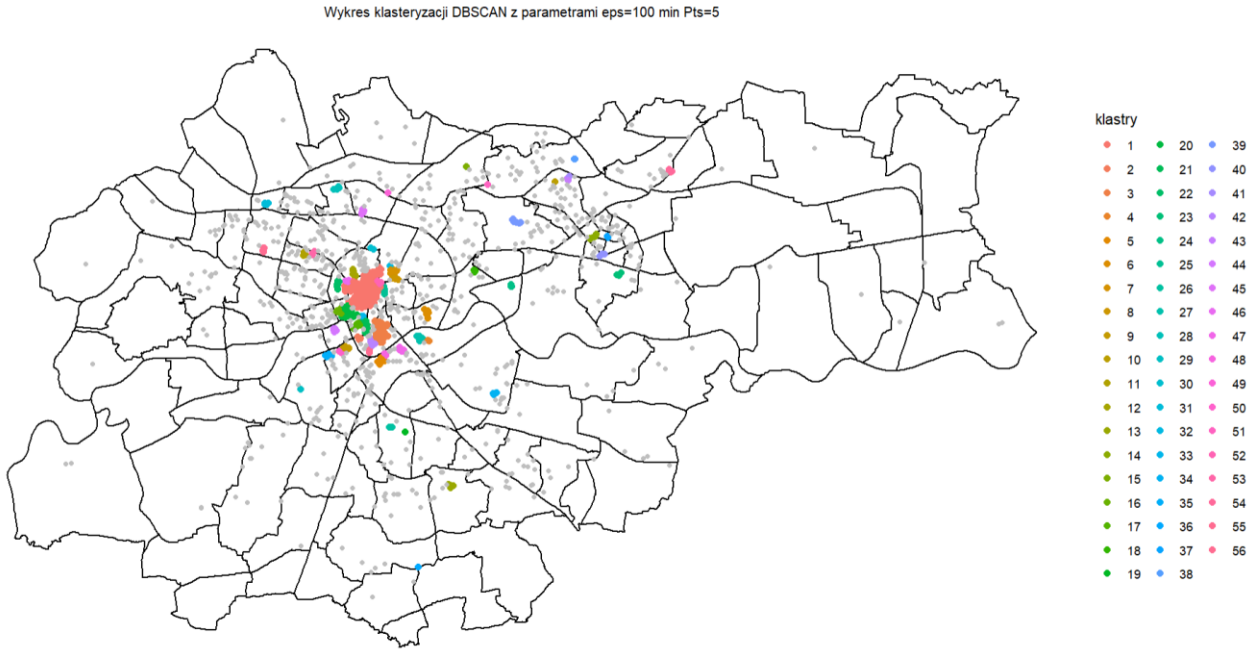


eps=100, minPts=5

```
db <- dbscan(Data, eps =100, minPts = 5)
db
```

```
DBSCAN clustering for 2000 objects.
Parameters: eps = 100, minPts = 5
The clustering contains 56 cluster(s) and 874 noise points.

  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32
874 438   6 120  12  10  10  33   5   9   6   7   8   8  15   8  12  40   6   7  50  18   8   6   5  22  10  10   7   9   7   5  17
 33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56
 11   7  16   7   5   7  11   6   5  20   7   9  10   6   9   8   6   5   7   7   6   5   7   5
```

```
cluster_dbscan<- db$cluster
groups_dbscan<-Data[cluster_dbscan != 0,]
noise_dbscan<-Data[cluster_dbscan == 0,]
clusters_dbscan<-as.factor(cluster_dbscan)
color_dbscan<-clusters_dbscan[clusters_dbscan!=0]


ggplot() +
  geom_polygon(data = shp, aes(x = long, y = lat, group = group),fill=NA, color = "black")+
  theme_void()+
  geom_point(aes(groups_dbscan$POINT_X, groups_dbscan$POINT_Y, color = color_dbscan),pch=16,size=2)+
  geom_point(aes(noise_dbscan$POINT_X, noise_dbscan$POINT_Y),pch=20,size=2,color='grey' )+
  labs(color="klastry")+ggtitle("Wykres klasteryzacji DBSCAN z parametrami eps=100 min Pts=5")+
  theme(plot.title = element_text(size=10, hjust = 0.5))+coord_fixed()
```

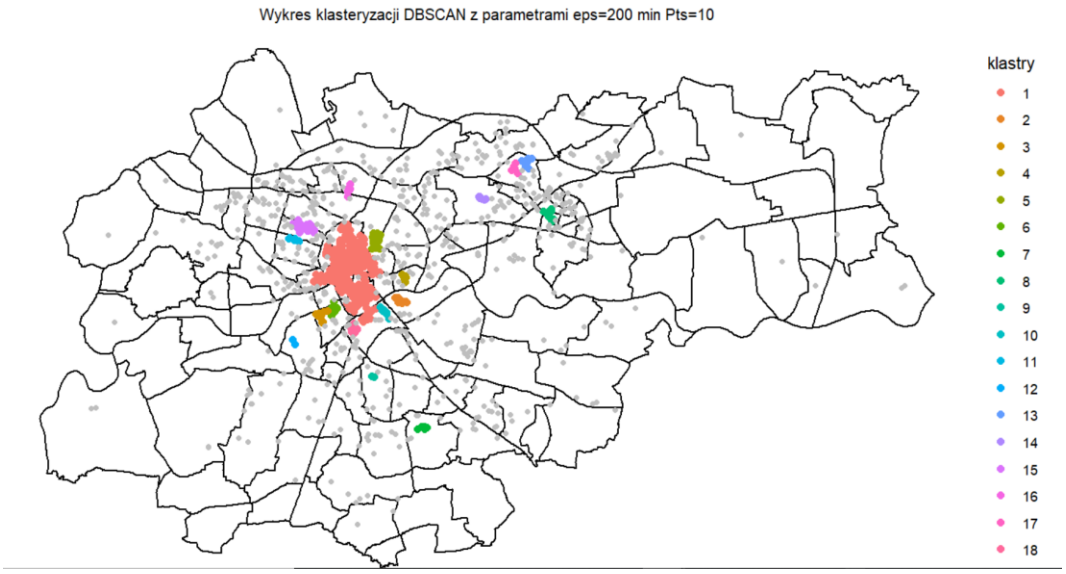


eps=200 minPts=10

```
db <- dbscan(Data, eps =200, minPts = 10)
db
```

```
> db
DBSCAN clustering for 2000 objects.
Parameters: eps = 200, minPts = 10
The clustering contains 18 cluster(s) and 828 noise points.

  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
828 875  23  21  10  64  16  12  21  10  13  10  10  17  11  26  12  11  10
```

```
cluster_dbscan<- db$cluster
groups_dbscan<-Data[cluster_dbscan != 0,]
noise_dbscan<-Data[cluster_dbscan == 0,]
clusters_dbscan<-as.factor(cluster_dbscan)
color_dbscan<-clusters_dbscan[clusters_dbscan!=0]


ggplot() +
  geom_polygon(data = shp, aes(x = long, y = lat, group = group),fill=NA, color = "black")+
  theme_void()+
  geom_point(aes(groups_dbscan$POINT_X, groups_dbscan$POINT_Y, color = color_dbscan),pch=16,size=2)+
  geom_point(aes(noise_dbscan$POINT_X, noise_dbscan$POINT_Y),pch=20,size=2,color='grey' )+
  labs(color="klastry")+ggtitle("Wykres klasteryzacji DBSCAN z parametrami eps=200 min Pts=10")+
  theme(plot.title = element_text(size=10, hjust = 0.5))+coord_fixed()
```

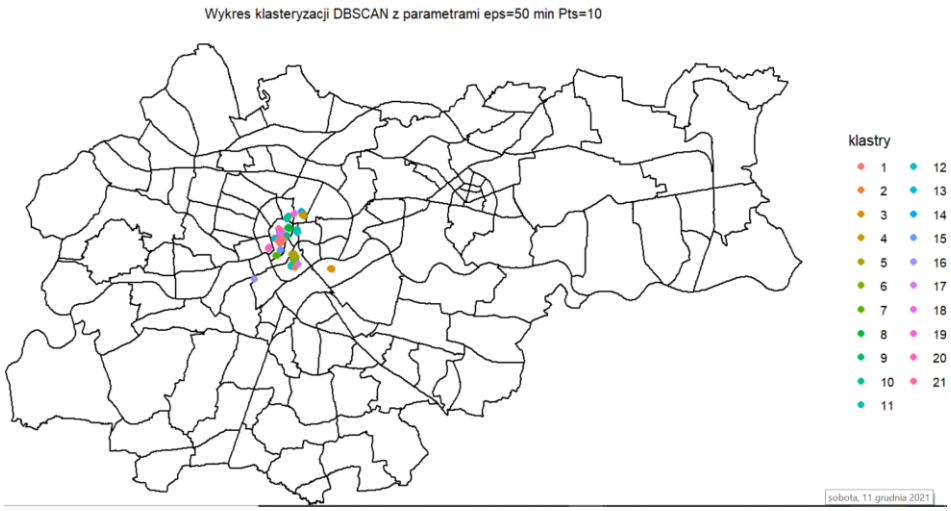


eps=50, minPts=10

```
db <- dbscan(Data, eps =50, minPts = 10)
db
```

```
DBSCAN clustering for 2000 objects.
Parameters: eps = 50, minPts = 10
The clustering contains 21 cluster(s) and 1538 noise points.

   0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20   21
1538  133   18   12   30   19   26   28   21   12   18   20   11   11   17   11   12   13   19   11   10   10
```

```
cluster_dbscan<- db$cluster
groups_dbscan<-Data[cluster_dbscan != 0,]
noise_dbscan<-Data[cluster_dbscan == 0,]
clusters_dbscan<-as.factor(cluster_dbscan)
color_dbscan<-clusters_dbscan[clusters_dbscan!=0]


ggplot() +
  geom_polygon(data = shp, aes(x = long, y = lat, group = group),fill=NA, color = "black")+
  theme_void()+
  geom_point(aes(groups_dbscan$POINT_X, groups_dbscan$POINT_Y, color = color_dbscan),pch=16,size=2)+
  #geom_point(aes(noise_dbscan$POINT_X, noise_dbscan$POINT_Y),pch=20,size=2,color='grey' )+
  labs(color="klastry")+ggtitle("Wykres klasteryzacji DBSCAN z parametrami eps=50 min Pts=10")+
  theme(plot.title = element_text(size=10, hjust = 0.5))+coord_fixed()
```



WNIOSKI DBSCAN:

1. eps=10,minPts=5

dużo pojedyńczych klastrów, największe skupisko można zauważyć w dzielnicy stare miasto, skupiska klastrów występują również w dzielnicach: Dębniki, Warszawskie, Prądnik Czerwony i Bieńczyce

2. eps=450,minPts=5

Przy tych parametrach widzimy 3 duże klastry o dużej gęstości: różowy- rozciąga się przez dzielnice Stare Miasto, Krowodrza i lekko wkracza na teren dzielnicy Bronowice, żółty-występuje na terenie Rakowic, pogranicza Debnic i Czyżyn oraz Olszy i Prądnika Czerwonego, pomarańczowy- występuje na terenie Mistrzejowic, Bieńczyc i pogranicza Nowej Huty i Czyżyn.Występuje dużo klastrów o mniejszej gęstości występują na terenach osiedli Wola Justowska-Chełm, Zakrzówek,Wzgórza Krzesławickie,Kurdwanów,Piaski ,Rżąka, Nowy Prokocim i Płaszów

3. eps=100, minPts=5

Widoczne dużo skupisko klastrów w dzielnicy Stare miasto. Klastry o niskiej gęstości widoczne na terenie osiedli:Zakrzówek, Ludwinów,Swoszowice, Piaski,Wola Duchacka Zachód, na terenie dzielnicy Nowa Chuta przy granicy z dzielnicą Czyżyny, oraz w okolicach dzielnicy Krowodrza

4. eps=200, minPts=10

Widoczny jeden duży klaster o dużej gęstości na terenie dzielnicy stare miasto, klastry o niskiej gęstości na terenie osiedli:Piaski,Wola Duchocka Zachód, Zakrzówek, Ludwinów, Nowa Wieś Południe, Krowodrza-Nowa Wieś, Krowodrza Wschód, Czyżyny Lotnisko, Bieńczyce Nowe

5. eps=50,minPts=1

Duże skupisko klastrów o małej gęstości na terenie dzielnicy stare miasto, pojedynczy klaster widoczny na terenie osiedla Zabłocie

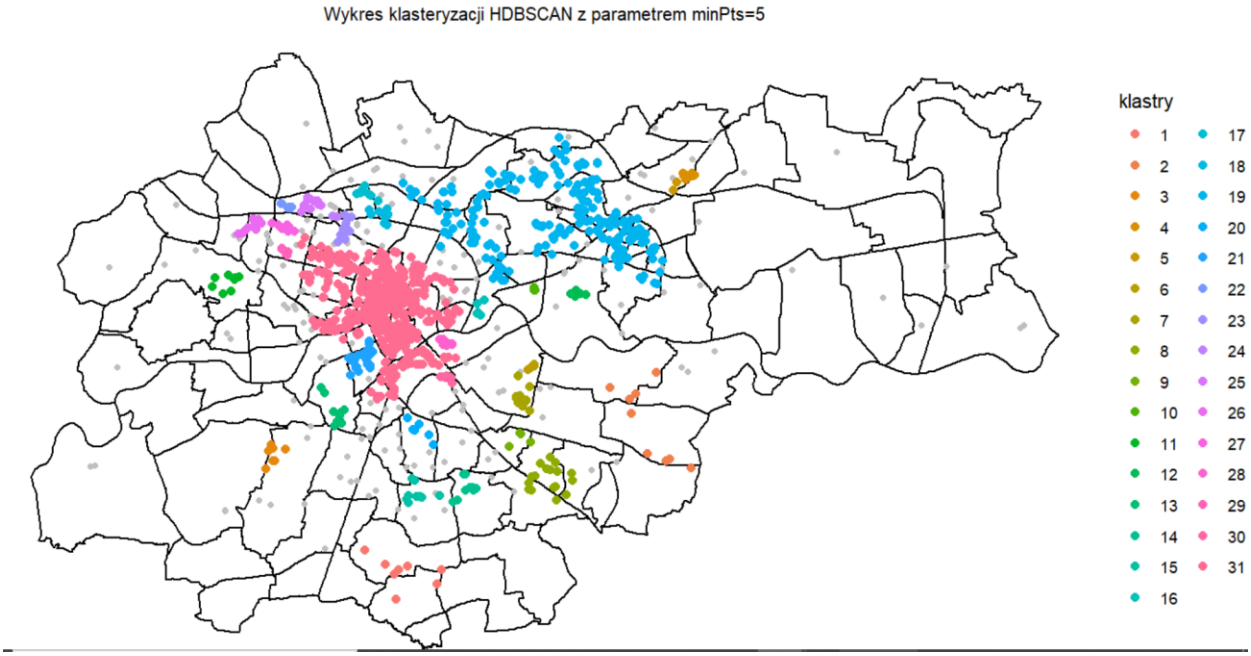
- **HDBSCAN**

MinPts=5

```
hdbscan<-hdbscan(Data, minPts=5)
hdbscan
```

```
The clustering contains 31 cluster(s) and 175 noise points.

   0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24   25
 175   14   13    7   11    6    5   17   20    5    5   10    6   18   17   10    6   13   20  297   22   46    7   12    5   11
  26   27   28   29   30   31
  11   10   23    9    8 1161
```

```
cluster_hdbscan <- hdbscan$cluster
groups_hdbscan<-Data[cluster_hdbscan != 0,]
noise_hdbscan<-Data[cluster_hdbscan == 0,]
clusters_hdbscan<-as.factor(hdbscan$cluster)
color_hdbscan<-clusters_hdbscan[clusters_hdbscan!=0]


ggplot() +
  geom_polygon(data = shp, aes(x = long, y = lat, group = group),fill=NA, color = "black")+
  theme_void()+
  geom_point(aes(noise_hdbscan$POINT_X, noise_hdbscan$POINT_Y),pch=20,size=2,color='grey' )+
  geom_point(aes(groups_hdbscan$POINT_X, groups_hdbscan$POINT_Y, color = color_hdbscan),pch=16,size=2)+
  labs(color="klastry")+ggtitle("Wykres klasteryzacji HDBSCAN z parametrem minPts=5")+
  theme(plot.title = element_text(size=10, hjust = 0.5))+coord_fixed()
```

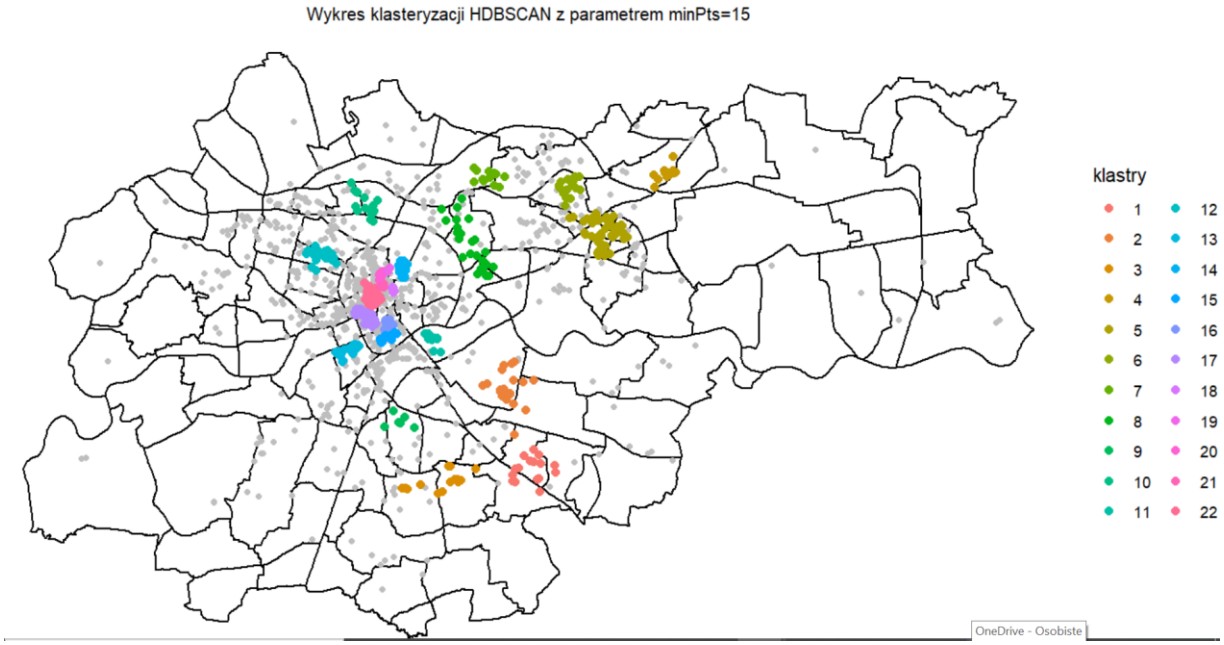


minPts=15

```
hdbscan<-hdbscan(Data, minPts=15)
hdbscan
```

```
Parameters: minPts = 15
The clustering contains 22 cluster(s) and 1013 noise points.

   0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20   21   22
1013   19   26   23   19   71   32   27   33   22   27   25   31   36   60   54   61  108   21   16   32   24  220

Available fields: cluster, minPts, cluster_scores, membership_prob, outlier_scores, hc
```

```
cluster_hdbscan <- hdbscan$cluster
groups_hdbscan<-Data[cluster_hdbscan != 0,]
noise_hdbscan<-Data[cluster_hdbscan == 0,]
clusters_hdbscan<-as.factor(hdbscan$cluster)
color_hdbscan<-clusters_hdbscan[clusters_hdbscan!=0]


ggplot() +
  geom_polygon(data = shp, aes(x = long, y = lat, group = group),fill=NA, color = "black")+
  theme_void()+
  geom_point(aes(noise_hdbscan$POINT_X, noise_hdbscan$POINT_Y),pch=20,size=2,color='grey' )+
  geom_point(aes(groups_hdbscan$POINT_X, groups_hdbscan$POINT_Y, color = color_hdbscan),pch=16,size=2)+
  labs(color="klastry")+ggtitle("Wykres klasteryzacji HDBSCAN z parametrem minPts=15")+
  theme(plot.title = element_text(size=10, hjust = 0.5))+coord_fixed()
```

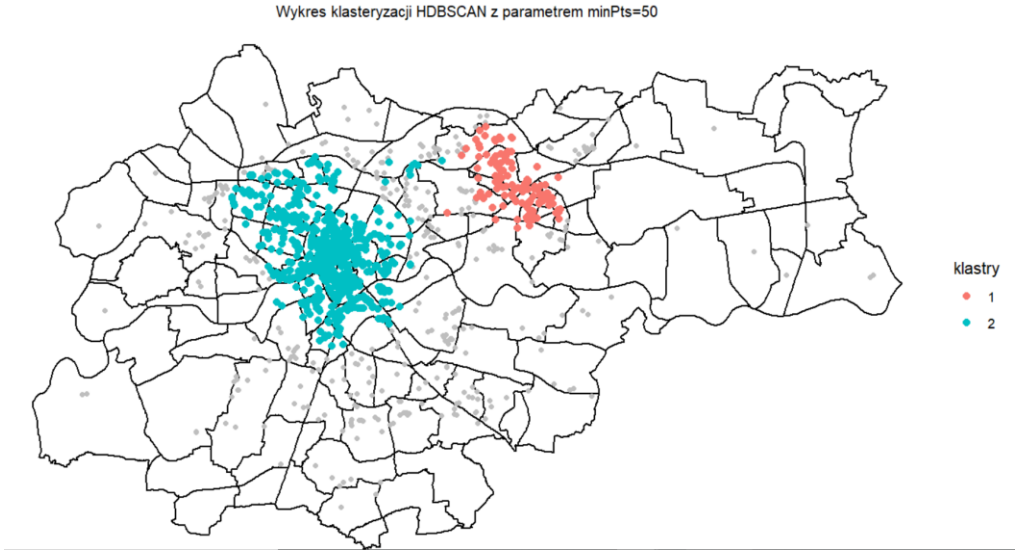


minPts=50

```
hdbscan<-hdbscan(Data, minPts=50)
hdbscan
```

```
> hdbscan
HDBSCAN clustering for 2000 objects.
Parameters: minPts = 50
The clustering contains 2 cluster(s) and 448 noise points.

   0    1    2
 448  172 1380
```

```
cluster_hdbscan <- hdbscan$cluster
groups_hdbscan<-Data[cluster_hdbscan != 0,]
noise_hdbscan<-Data[cluster_hdbscan == 0,]
clusters_hdbscan<-as.factor(hdbscan$cluster)
color_hdbscan<-clusters_hdbscan[clusters_hdbscan!=0]


ggplot() +
  geom_polygon(data = shp, aes(x = long, y = lat, group = group),fill=NA, color = "black")+
  theme_void()+
  geom_point(aes(noise_hdbscan$POINT_X, noise_hdbscan$POINT_Y),pch=20,size=2,color='grey' )+
  geom_point(aes(groups_hdbscan$POINT_X, groups_hdbscan$POINT_Y, color = color_hdbscan),pch=16,size=2)+
  labs(color="klastry")+ggtitle("Wykres klasteryzacji HDBSCAN z parametrem minPts=50")+
  theme(plot.title = element_text(size=10, hjust = 0.5))+coord_fixed()
```

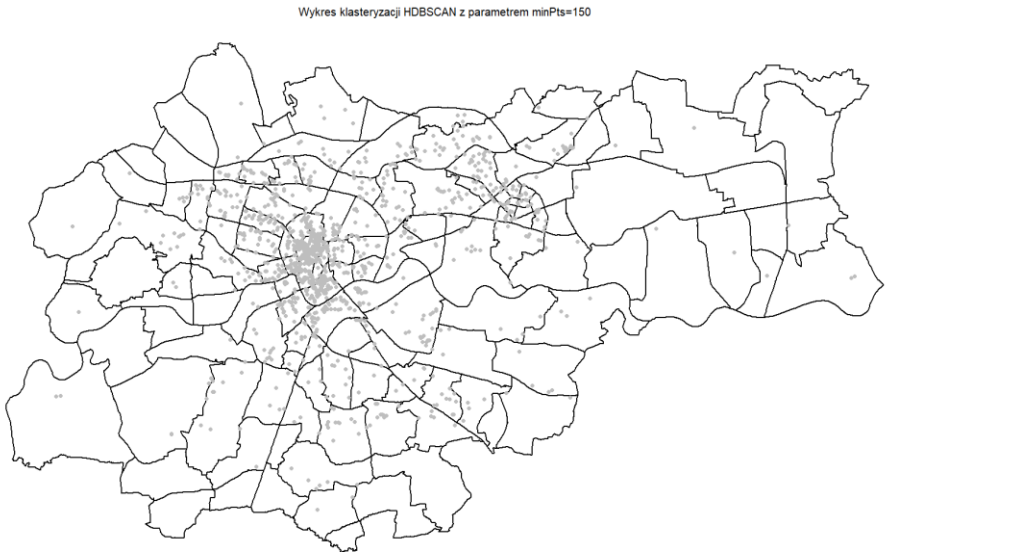


minPts=150

```
hdbscan<-hdbscan(Data, minPts=150)
hdbscan
```

```
Parameters: minPts = 150
The clustering contains 0 cluster(s) and 2000 noise points.

   0
2000
```

```
cluster_hdbscan <- hdbscan$cluster
groups_hdbscan<-Data[cluster_hdbscan != 0,]
noise_hdbscan<-Data[cluster_hdbscan == 0,]
clusters_hdbscan<-as.factor(hdbscan$cluster)
color_hdbscan<-clusters_hdbscan[clusters_hdbscan!=0]


ggplot() +
  geom_polygon(data = shp, aes(x = long, y = lat, group = group),fill=NA, color = "black")+
  theme_void()+
  geom_point(aes(noise_hdbscan$POINT_X, noise_hdbscan$POINT_Y),pch=20,size=2,color='grey' )+
  geom_point(aes(groups_hdbscan$POINT_X, groups_hdbscan$POINT_Y, color = color_hdbscan),pch=16,size=2)+
  labs(color="klastry")+ggtitle("Wykres klasteryzacji HDBSCAN z parametrem minPts=150")+
  theme(plot.title = element_text(size=10, hjust = 0.5))+coord_fixed()
```



WNIOSKI HDBSCAN:

1. minPts=5

Duża ilość klastrów. Widoczne dwa duże klastry: różowy obejmujący dzielnicę Stare Miasto, Grzegórzki i Krowodrza oraz niebieski-rozciągający się na terenie dzielnic Czyżyny i okolic pogranicza z Nową Hutą, Prądnik Czerwony, Mistrzejowice, Bieńczyce Jest też sporo klastrów o niskiej gęstości na terenie osiedli : Wola Justowska-Chełm, Kobierzyn, Zakrzówek, Ludwinów, Kurdwanów, Piaski, Swoszowice, Wróblowice, klaster na osiedlach Nowy Prokocim i Nowy Bieżanów, klaster o bardzo małej gęstości na terenie dzielnicy Podgaje i Wzgórza Krzesławickie

2. minPts=15

W porównaniu do mapy z poprzednim parametrem, tutaj nie utworzył się żaden duży klaster, za to mamy klastry mniejszych rozmiarów zlokalizowane „w grupach” tzn. wytworzyło się sporo klastrów o gęściejszych skupiskach. Największe skupisko kilku klastrów występuje na terenie dzielnicy Stare Miasto. Gęste skupiska klastrów występują na terenie dzielnic Bieńczyce i okolic granicznych z Nową Hutą. Klastry o niższej gęstości są zlokalizowane w obrębie osiedli: Rakowice, Dębie, Płaszów, Nowy Bieżanów i Nowy Prokocim

3. minPts=50

Utworzyły się 2 duże klastry. Niebieski o większej gęstości i różowy. Niebieski położony jest na terenie dzielnic:Stare Miasto, Grzegórzki i Krowodrza i niewielka część Prądnika Białego. Różowy zajmuje dzielnice Mistrzejowic, Bieńczyc i delikatnie zahacza o dzielnicę Nowej Huty.

4. minPts=150

klaster nie utworzył się

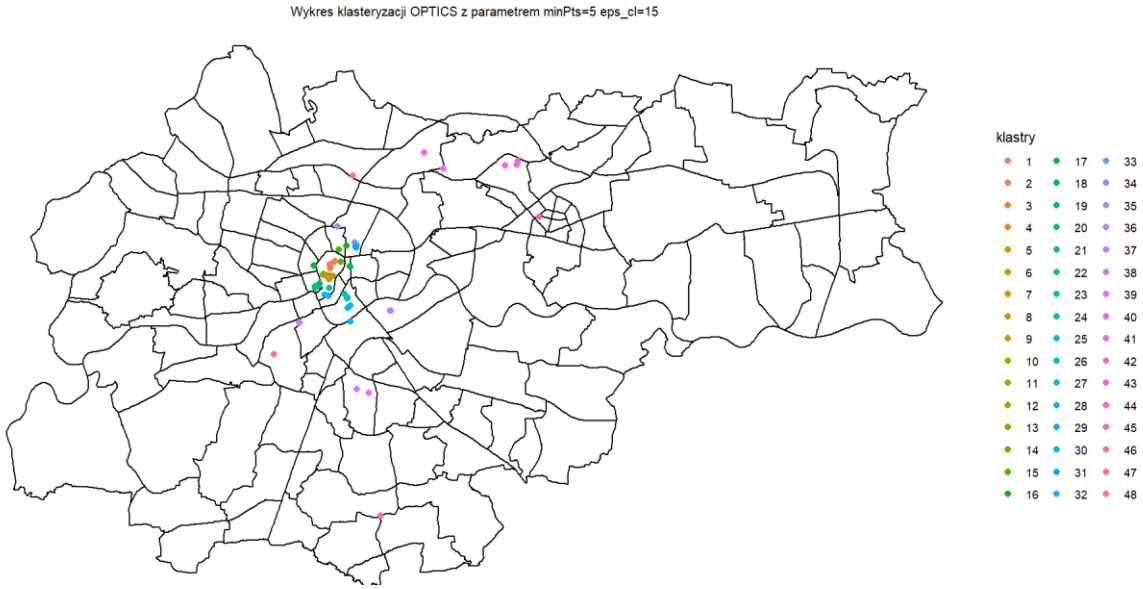
- **OPTICS**

minPts=5 ,eps_cl=15

```
optics<-optics(Data, minPts =5)
optics <- extractDBSCAN(optics, eps_cl = 15)
optics
```

```
Parameters: minPts = 5, eps = 7090.14557302626, eps_cl = 15, xi = NA
The clustering contains 48 cluster(s) and 1578 noise points.

   0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24   25
1578    5    6    5   18   17    9    5    8   20    9    9    5    5   16   12    9    7    9    7    5    5    8   10   11    5
  26   27   28   29   30   31   32   33   34   35   36   37   38   39   40   41   42   43   44   45   46   47   48
   8    6    5   25    7   13    5    6   17    6   10   12    7    7    7    8    6    6    5    9    6    9    7
```

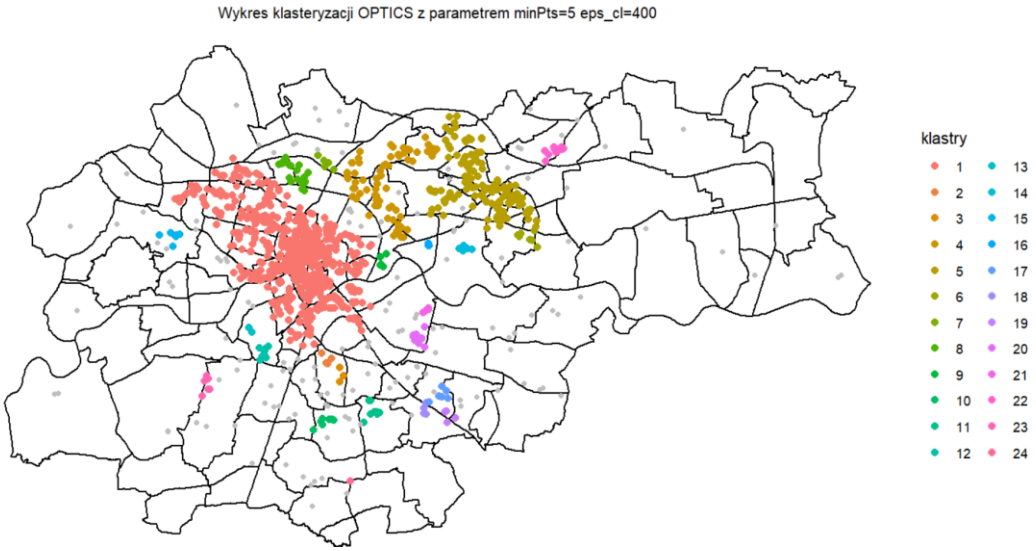
```
cluster_optics<- optics$cluster
groups_optics<-Data[cluster_optics != 0,]
noise_optics<-Data[cluster_optics == 0,]
clusters_optics<-as.factor(cluster_optics)
color_optics<-clusters_optics[clusters_optics!=0]


ggplot() +
  geom_polygon(data = shp, aes(x = long, y = lat, group = group),fill=NA, color = "black")+
  theme_void()+
  geom_point(aes(groups_optics$POINT_X, groups_optics$POINT_Y, color = color_optics),pch=16,size=2)+
  #geom_point(aes(noise_optics$POINT_X, noise_optics$POINT_Y),pch=20,size=2,color='grey' )+
  labs(color="klastry")+ggtitle("Wykres klasteryzacji OPTICS z parametrem minPts=5 eps_cl=15")+
  theme(plot.title = element_text(size=10, hjust = 0.5))+coord_fixed()
```



minPts=5, eps_cl=400

```
optics<-optics(Data, minPts =5)
#extractDBSCAN() extracts a clustering from an OPTICS ordering
optics <- extractDBSCAN(optics, eps_cl = 400)
optics
```

```
cluster_optics<- optics$cluster
groups_optics<-Data[cluster_optics != 0,]
noise_optics<-Data[cluster_optics == 0,]
clusters_optics<-as.factor(cluster_optics)
color_optics<-clusters_optics[clusters_optics!=0]


ggplot() +
  geom_polygon(data = shp, aes(x = long, y = lat, group = group),fill=NA, color = "black")+
  theme_void()+
  geom_point(aes(groups_optics$POINT_X, groups_optics$POINT_Y, color = color_optics),pch=16,size=2)+
  geom_point(aes(noise_optics$POINT_X, noise_optics$POINT_Y),pch=20,size=2,color='grey' )+
  labs(color="klastry")+ggtitle("Wykres klasteryzacji OPTICS z parametrem minPts=5 eps_cl=400")+
  theme(plot.title = element_text(size=10, hjust = 0.5))+coord_fixed()
```

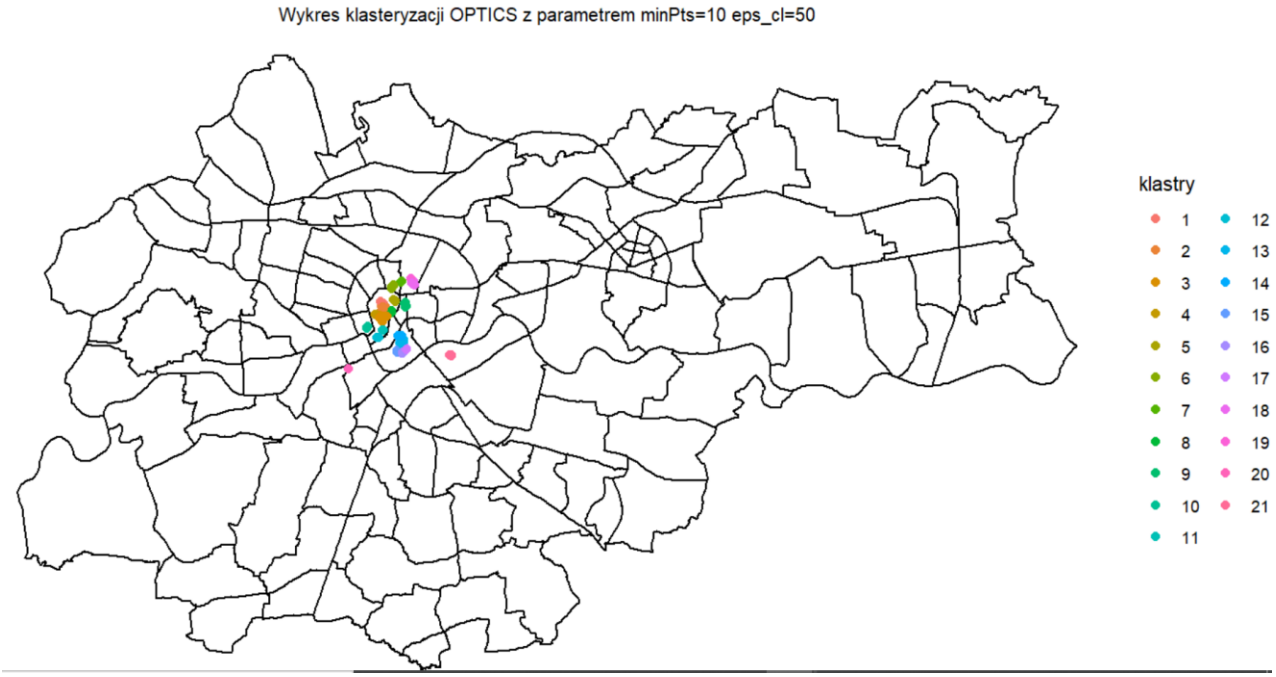


minPts=10,eps_cl=50

```
optics<-optics(Data, minPts =10)
optics <- extractDBSCAN(optics, eps_cl = 50)
optics
```

```
Parameters: minPts = 10, eps = 10105.251907727, eps_cl = 50, xi = NA
The clustering contains 21 cluster(s) and 1547 noise points.

   0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20   21
1547   10   18  131   10   20   18   11   12   20    8   11   28   26   19    9   18   13   30   17   12   12
```

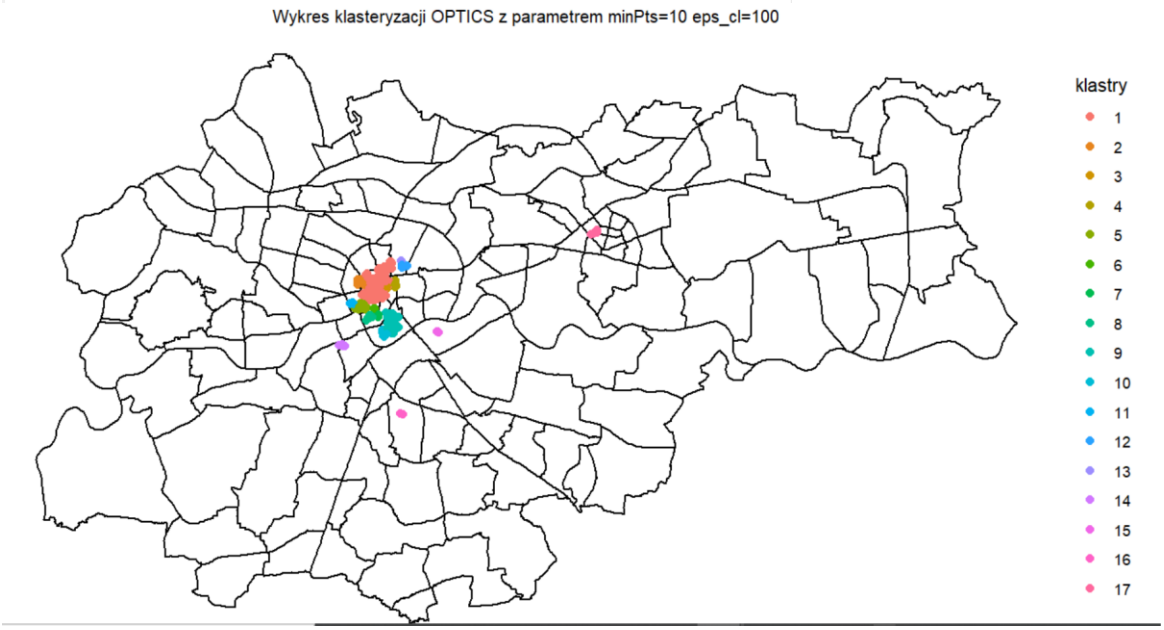
```
cluster_optics<- optics$cluster
groups_optics<-Data[cluster_optics != 0,]
noise_optics<-Data[cluster_optics == 0,]
clusters_optics<-as.factor(cluster_optics)
color_optics<-clusters_optics[clusters_optics!=0]


ggplot() +
  geom_polygon(data = shp, aes(x = long, y = lat, group = group),fill=NA, color = "black")+
  theme_void()+
  geom_point(aes(groups_optics$POINT_X, groups_optics$POINT_Y, color = color_optics),pch=16,size=2)+
  #geom_point(aes(noise_optics$POINT_X, noise_optics$POINT_Y),pch=20,size=2,color='grey' )+
  labs(color="klastry")+ggtitle("Wykres klasteryzacji OPTICS z parametrem minPts=10 eps_cl=50")+
  theme(plot.title = element_text(size=10, hjust = 0.5))+coord_fixed()
```



minPts=10,eps_cl=100

```
optics<-optics(Data, minPts =10)
optics <- extractDBSCAN(optics, eps_cl = 100)
optics
```

```
cluster_optics<- optics$cluster
groups_optics<-Data[cluster_optics != 0,]
noise_optics<-Data[cluster_optics == 0,]
clusters_optics<-as.factor(cluster_optics)
color_optics<-clusters_optics[clusters_optics!=0]


ggplot() +
  geom_polygon(data = shp, aes(x = long, y = lat, group = group),fill=NA, color = "black")+
  theme_void()+
  geom_point(aes(groups_optics$POINT_X, groups_optics$POINT_Y, color = color_optics),pch=16,size=2)+
  #geom_point(aes(noise_optics$POINT_X, noise_optics$POINT_Y),pch=20,size=2,color='grey' )+
  labs(color="klastry")+ggtitle("Wykres klasteryzacji OPTICS z parametrem minPts=10 eps_cl=100")+
  theme(plot.title = element_text(size=10, hjust = 0.5))+coord_fixed()
```



WNIOSKI OPTICS:

1. minPts=5 ,eps_cl=15

Bardzo dużo pojedynczych klastrów. Brak większych klastrów. Zagęszczenie klastrów występuje w okolicach Starego Miasta.

2. minPts=5, eps_cl=400

Utworzyły się 3 duże klastry: różowy(na terenie Starego Miasta i Krowodrzy),pomarańczowy(Prądnik Czerwony) i musztardowy(dzielnica Mistrzejowice, Bieńczyce i okolice tych dzielnic). Pozostałe klastry są bardzo małe o niewielkim zagęszczeniu, występują m.in. na terenie dzielnic: Wola Justowska-Chełm, Wzgórza Krzesławickie, Czyżyny Łęg, Płaszów, Zakrzówek

3. minPts=10,eps_cl=50

W porównaniu efektów z parametrami minPts=5 ,eps_cl=15, pojedyncze punkty znikły, zostało tylko skupisko zagęszczonej ilości klastrów w okolicach Starego Miasta.

4. minPts=10,eps_cl=100

Obserwujemy skupisko dużej ilości pojedynczych klastrów w dzielnicy Stare Miasto. Pojawiły się pojedyncze punkty na terenie osiedli: Wola Duchocka-Zachód, Zakrzówek, Zabłocie i na terenie dzielnicy Nowa Huta

PODSUMOWANIE

Najlepsze efekty uzyskałam stosując algorytm HDBSCAN, mimo że spodziewałam się że to OPTICS lepiej sobie poradzi z danymi, ponieważ jest ulepszoną wersją DBSCAN. Z otrzymanych rezultatów klasteryzacji mogę stwierdzić, że najwyższa przestępczość występuje w rejonach dzielnic :Stare Miasto,Grzegórzki i Krowodrza, Mistrzejwice, Bieńczyć i osiedla w dzielnicy Nowej Huty(na rys.)-to na tych terenach utworzyły się dwa duże klastry

Rys.

Niewielka ilość przestępstw została zlokalizowane w obrębie osiedli: Rakowice, Dębie,Płaszów,Nowy Bieżanów i Nowy Prokocim-tam występowały pojedyncze punkty. Jednoznacznie można natomiast stwierdzić, że bezkonkurencyjnie najwięcej przestępstw jest na terenie Starego Miasta, na każdej mapie w tej lokalizacji tworzył się dość duży klaster.