Mājas darbs 2-1: MDS, ISOMAP

1. Uzdevums: MDS, pilsētu savstarpējie attālumi

Vispirms jāielādē dati. Šajā gadījumā izvēlēta datu kopa WG22 ("West Germany"), kur apkopoti attālumi starp 22 Rietumvācijas pilsētām (datu kopa acīmredzot nav tā jaunākā), pieejama Floridas Štata universitātes mājas lapā kopā ar citām pilsētu attālumu datu kopām.

```
dist.de <- read.table("wg22_dist.txt")
names <- read.table("wg22_name.txt")
```

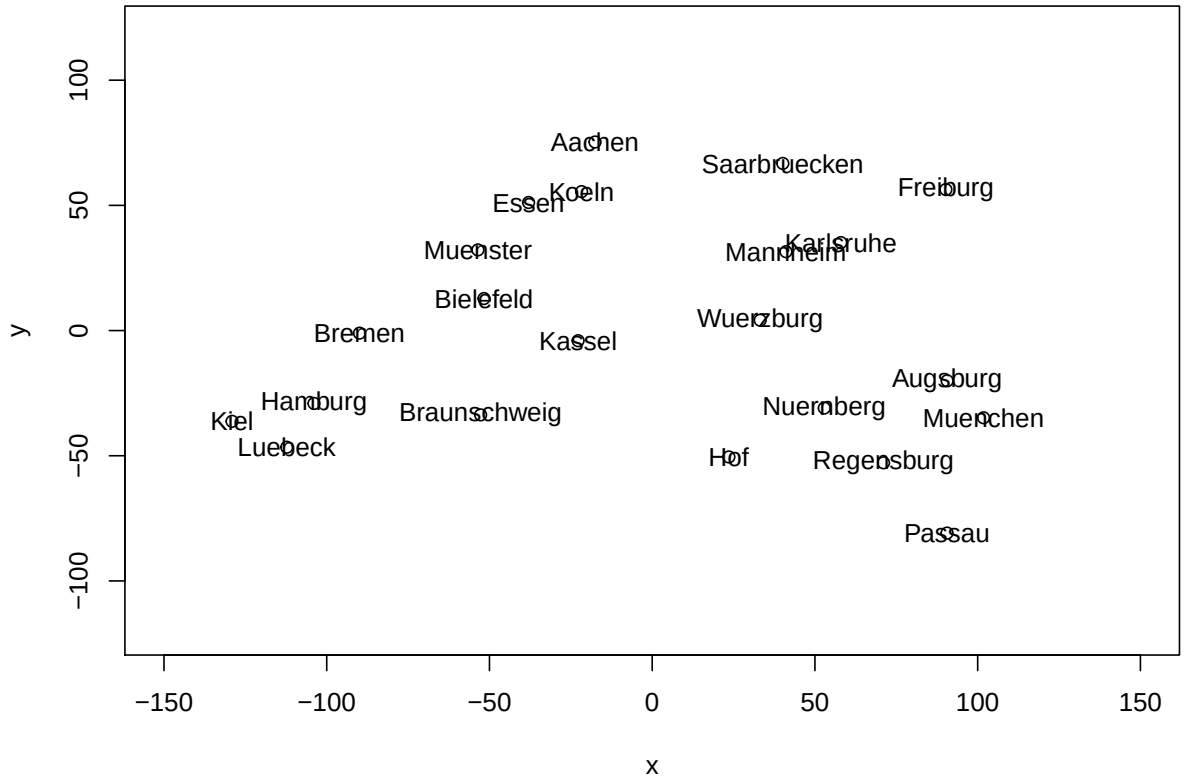
Pēc tam veic dimensiju redukciju:

```
fit <- cmdscale(dist.de, eig = TRUE, k = 2)
```

Iznesot iegūtos punktus 2-dimensiju koordinātu sistēmā un grafiski attēlojot, redzams, ka to izkārtojums ir līdzīgs Vācijas kartei, taču iespējams, nepieciešams pagriezt (samainīt x-y asis) un/vai kādu no asīm apgriezt otrādi (reizināt ar -1).

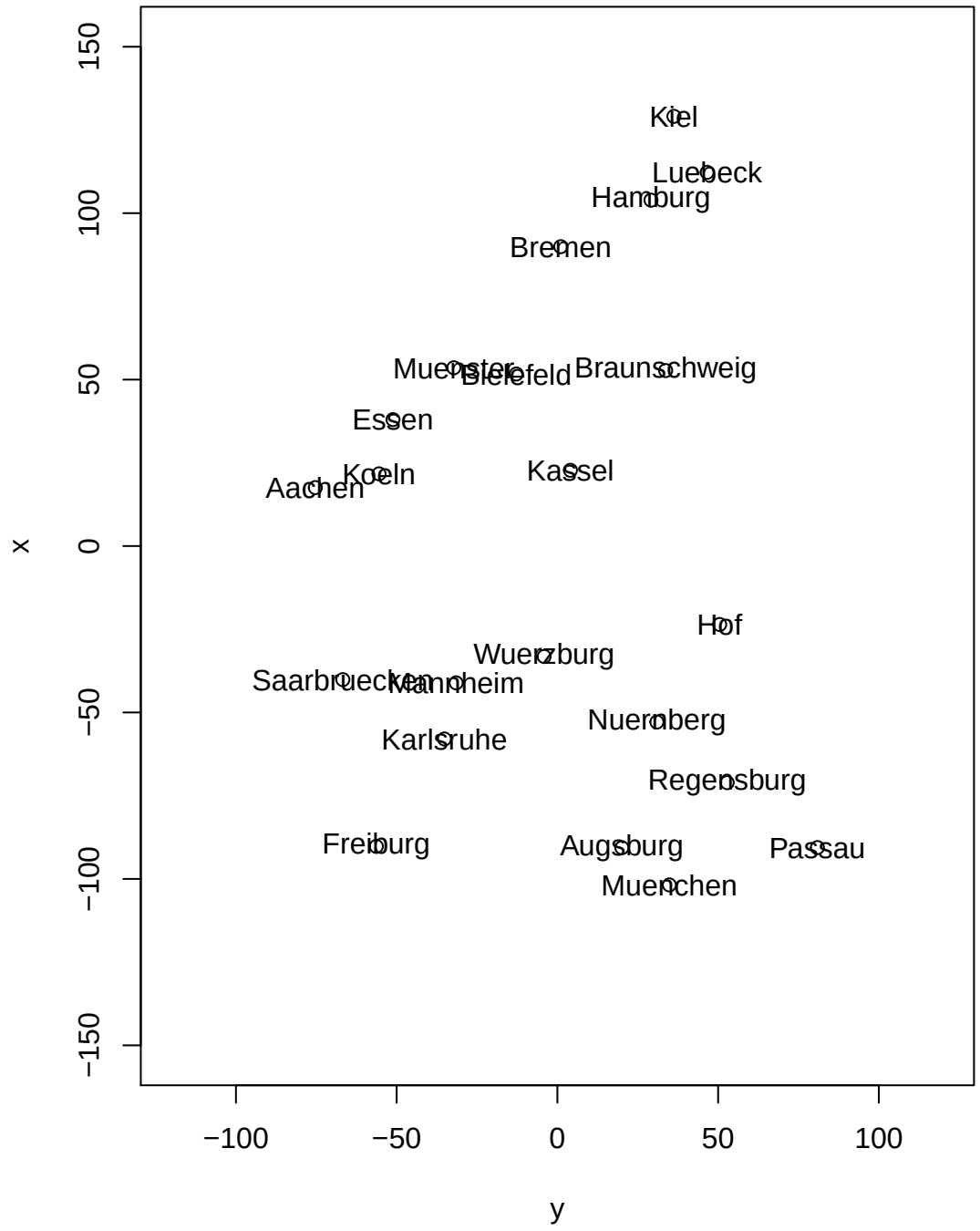
Figure 1: Vācijas karte no Google Maps

```
x <- fit$points[, 1]
y <- fit$points[, 2]
plot(x,y,xlim=c(-150,150),ylim=c(-120,120))
text(x,y,labels=names$V1)
```



Veicot korekcijas:

```
x <- -x
y <- -y
plot(y,x,xlim=c(-120,120),ylim=c(-150,150))
text(y,x,labels=names$V1)
```


Kiel
Luebeck
Hamburg
Bremen
Muenster
Bielefeld
Braunschweig
Essen
Koeln
Aachen
Kassel
Hof
Wuerzburg
Saarbruecken
Mannheim
Nuernberg
Karlsruhe
Regensburg
Freiburg
Augsburg
Passau
Muenchen
x
y
150
100
50
0
−50
−100
−150
−100
−50
0
50
100

2. Uzdevums: ISOMAP

Kļūda kodā

Vispirms, piebilde par kļūdu "ProjectionBasedClustering" modulī. Kļūda atrodama izejas koda failā Isomap.R (Github, ProjectionBasedClustering), 44 rindā:

```
res=vegan::isomap(Distances,ndim = 2,k=40,fragmentedOK=T, path = "shortest")
```

Tā vietā, lai izmantotajai funkcijai padotu lietotāja dotos parametrus, tiek izmantotas konstantes. To var labot, nokopējot visu funkciju savā koda failā un nomainot rindu uz

```
res=vegan::isomap(Distances,ndim = OutputDimension,k=k,fragmentedOK=T, path = "shortest")
```

vai arī tiešā veidā iekļaujot bibliotēku "vegan" un iekopējot tikai vajadzīgās rindas no funkcijas (izlaižot dažādas pārbaudes un grafika zīmēšanu):

```
library(vegan)
Isomap  = function(Distances,k,OutputDimension=2){
  res=vegan::isomap(Distances,ndim = OutputDimension,k=k,fragmentedOK=T, path = "shortest")
  ProjectedPoints=res$points
  return(list(ProjectedPoints=ProjectedPoints))
}
```

Datu apstrāde

Datus ielasa ar 'foreign' moduļa palīdzību:

```
library(foreign)
data <- read.arff('unbalanced.arff')
```

Lai varētu klases "Active" un "Inactive" datos attēlot kā krāsas, jāuzraksta funkcija, kas ģenerē attēlu:

```
colors <- function(status) {
  sapply(status, function(i) {
    if (i == "Active"){
      "red"
    } else {
      "blue"
    }
  })
}
color_row <- colors(data$Outcome)
```

No datiem izgriež klases rindu

```
trimmed <- data[,1:length(data[1,])-1]
```

un aprēķina pilno Eiklīda distanču matricu:

```
distances <- as.matrix(dist(as.matrix(trimmed)))
```

No distanču matricas var iegūt ISOMAP dimensiju redukcijas algoritma rezultātu. Koda paraugs (k=5):

```
fit <- Isomap(distances, 5, 3)
x <- fit$ProjectedPoints[,1]
y <- fit$ProjectedPoints[,2]
z <- fit$ProjectedPoints[,3]
```

Iespējams ģenerēt 3-dimensionālus, interaktīvus grafikus, taču tos dokumentā tiešā veidā iekļaut nevar, tāpēc katrai k vērtībai iekļauts iepriekš ģenerēts attēls. Kods ir sekojošs:

```
library(rgl)
plot3d(x,y,z,
       col=color_row,
       type="s",
       radius=3.5)
```

Grafiki ģenerēti k vērtībām 1,2,3,5,10,40. Ar k vērtību 1 tuvāko kaimiņu grafs nav sakarīgs, tāpēc Isomap funkcija automātiski atgriež tikai rezultātu lielākajai sakarīgai komponentei, kurā visi elementi ir "Active":

```
## Warning in isomapdist(dist, ...): data are fragmented: taking the largest
## fragment
```

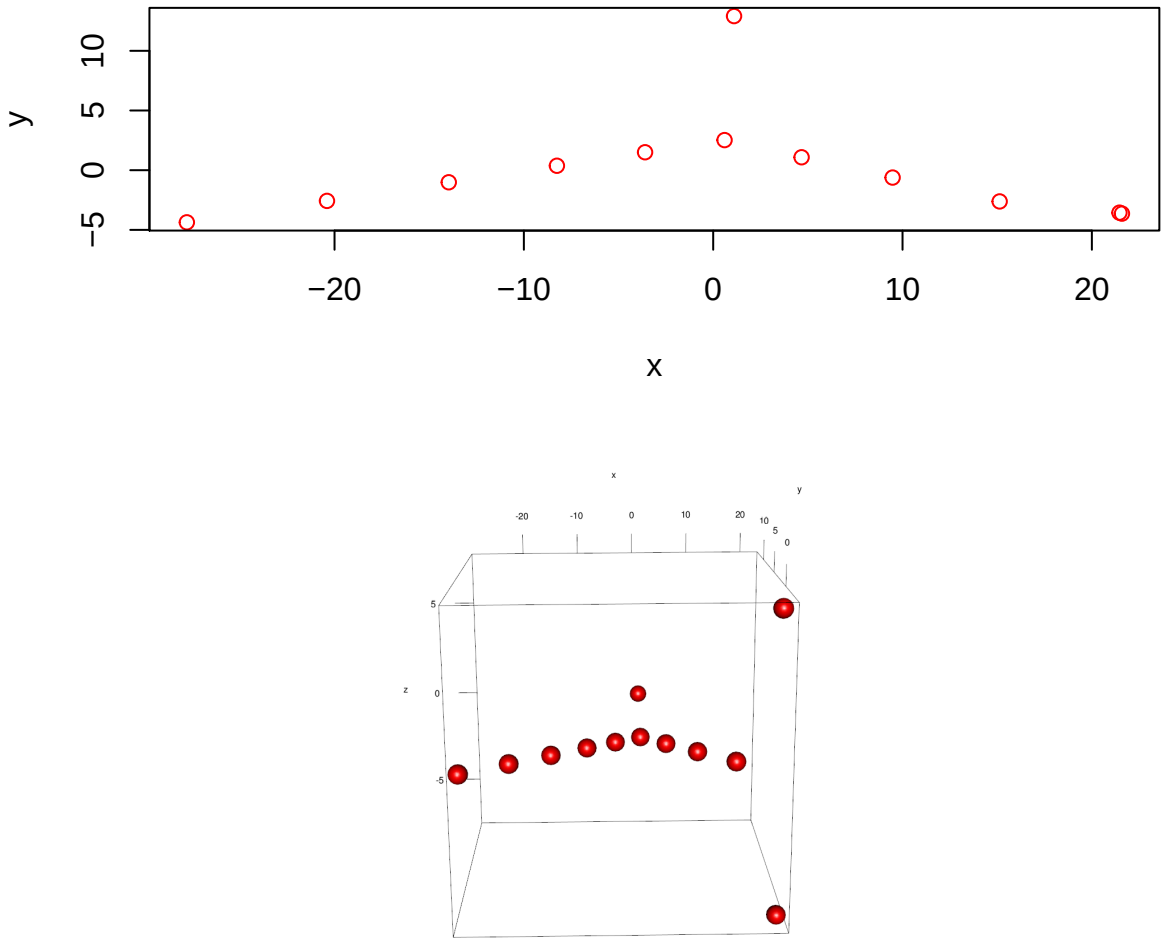
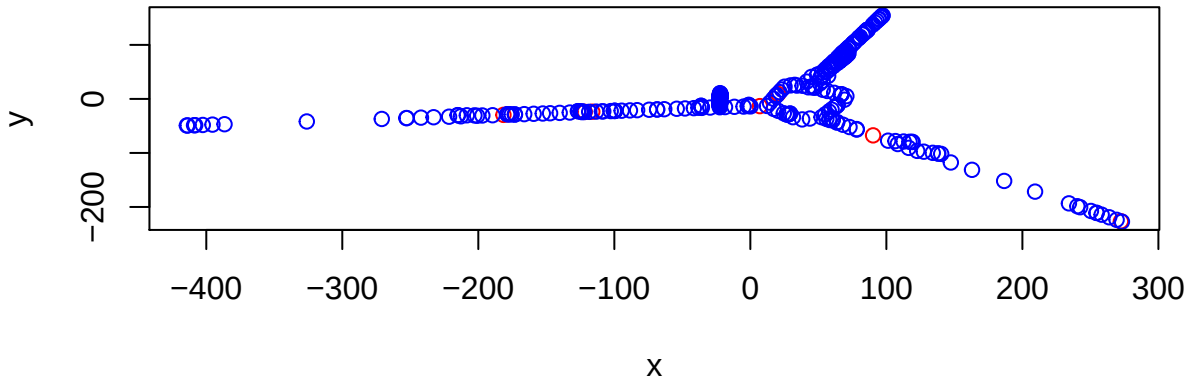


Figure 2: 3-dimensiju attēls k=1

K = 2, vēl joprojām grafs nav sakarīgs taču tagad iekļauti arī daudzi “Inactive” elementi:

```
## Warning in isomapdist(dist, ...): data are fragmented: taking the largest
## fragment
```



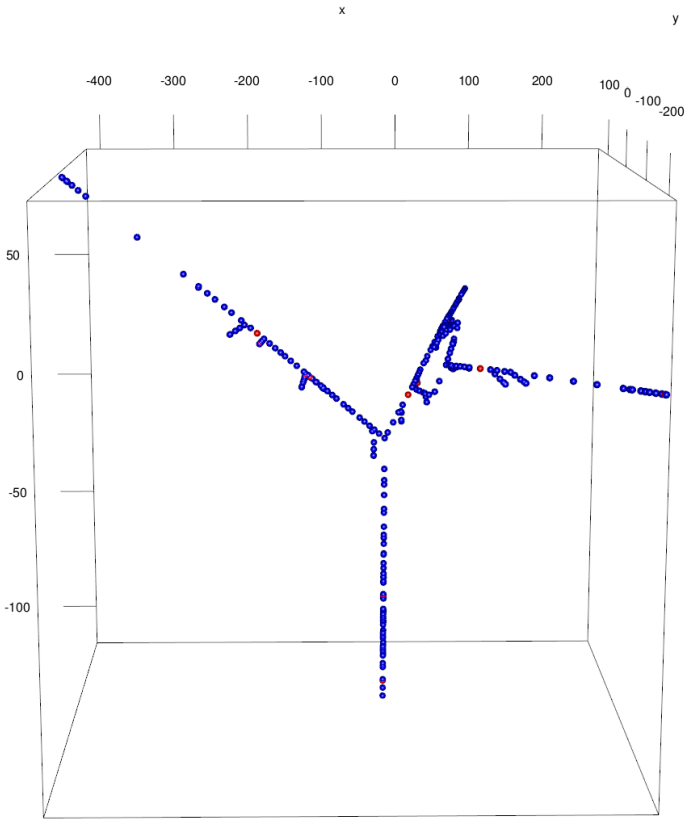
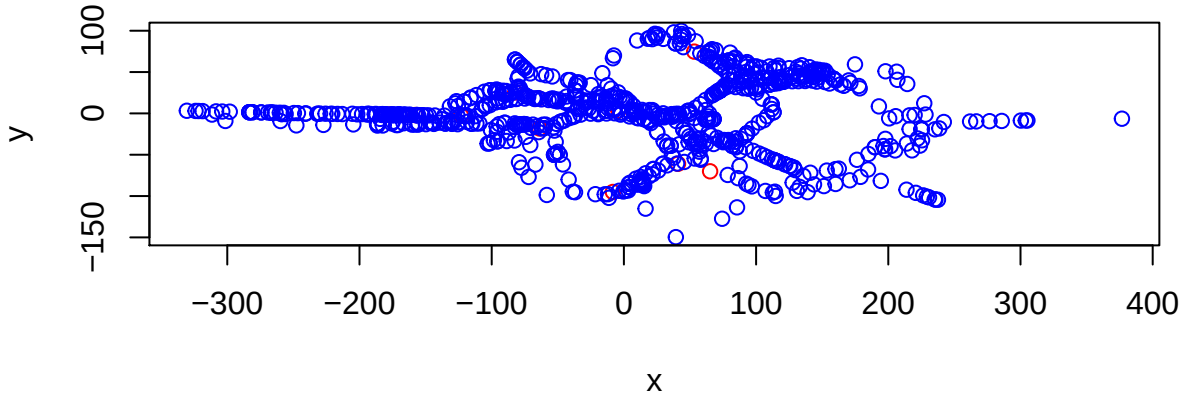


Figure 3: 3-dimensiju attēls k=2

K = 3, šoreiz grafs jau ir sakarīgs, taču projekcijas struktūra vēl arvien ir izteikti "izstiepta", t.i., elementi veido garas rindas, jo dominē grafu distances, nevis telpiskās:



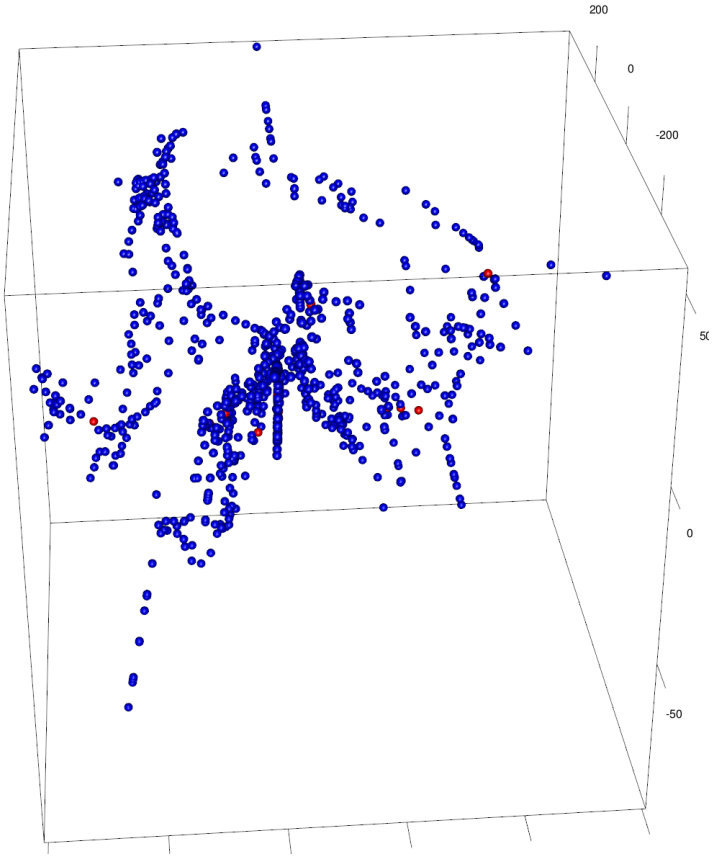
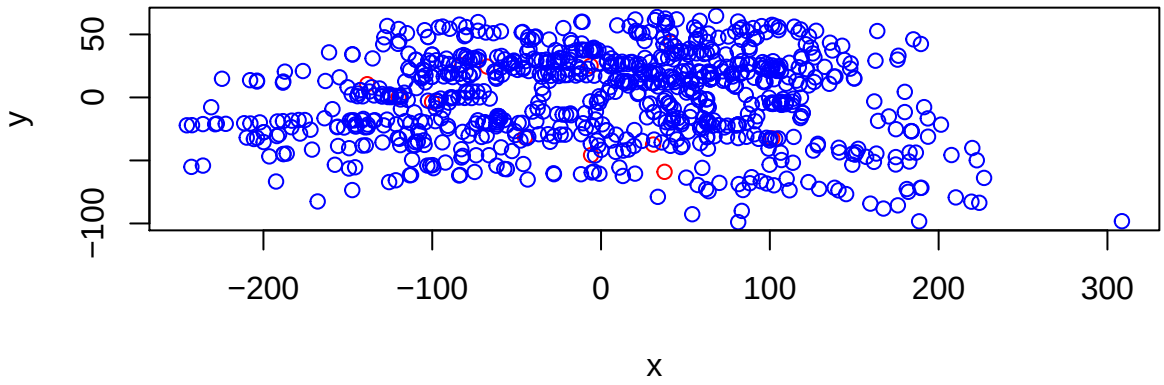


Figure 4: 3-dimensiju attēls k=3

K = 5, redzams, ka projekcija sāk telpiski "izplūst", lielāka nozīme lokāli ir Eiklīda, nevis telpiskām distancēm, veidojas nepārtraukta virsma:



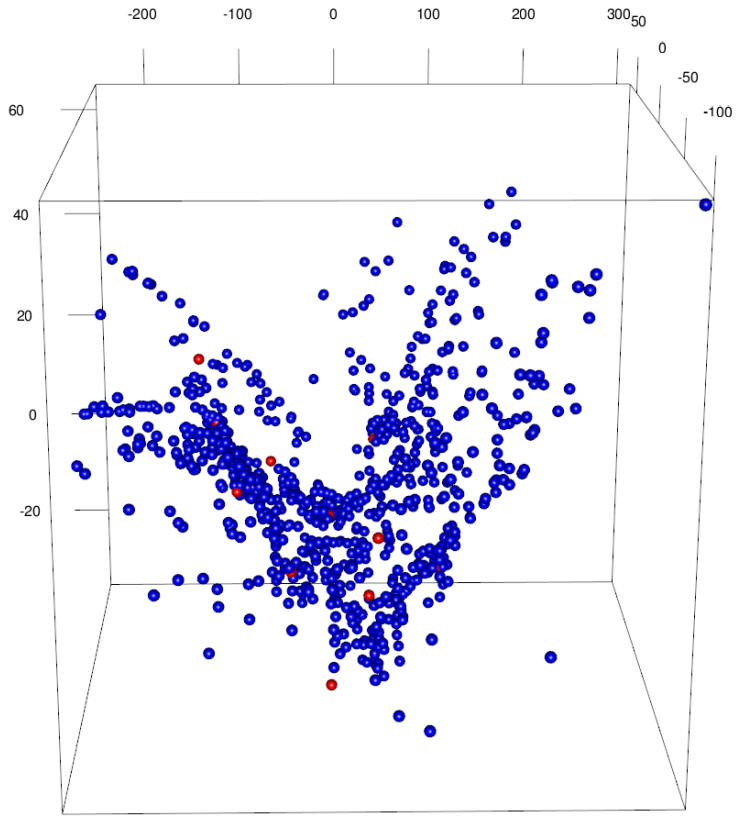
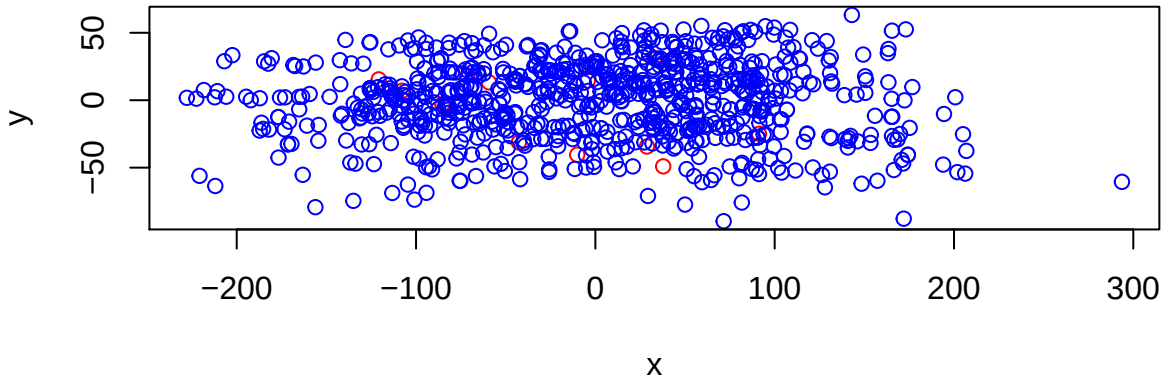


Figure 5: 3-dimensiju attēls k=5

K = 10, virsma kļūst līdzenāka:



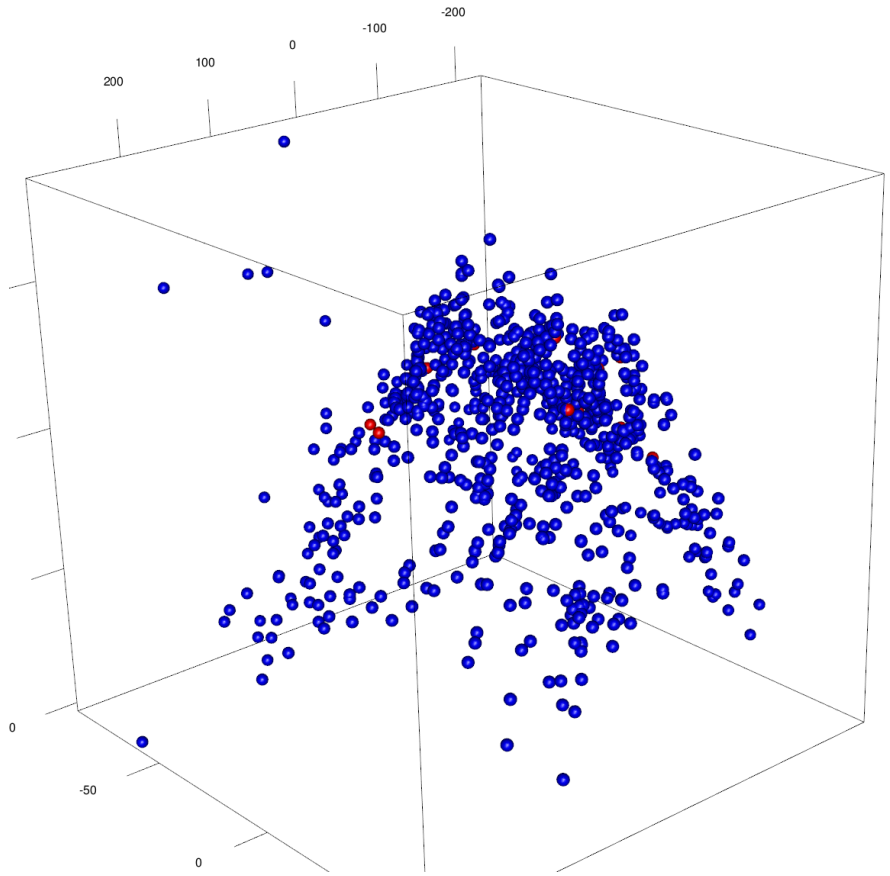
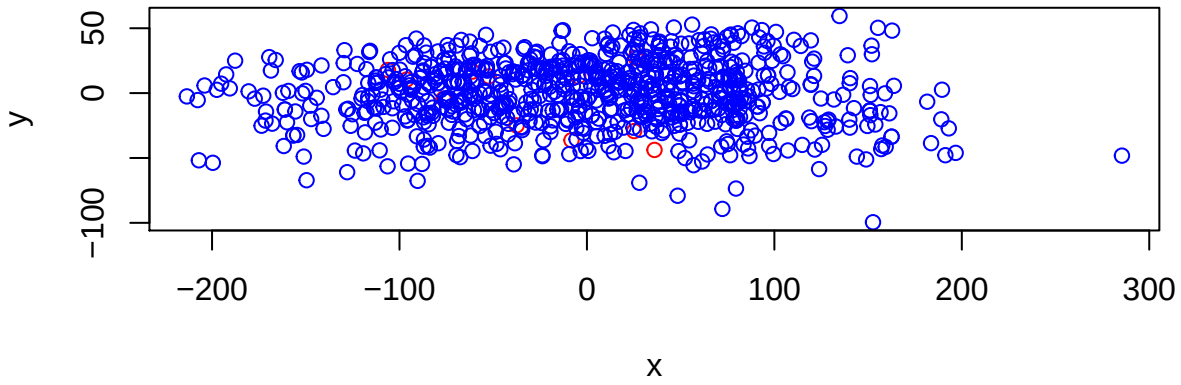


Figure 6: 3-dimensiju attēls k=10

K = 40, projicēto punktu kopas “grafa” pazīmes vairs nav redzamas:



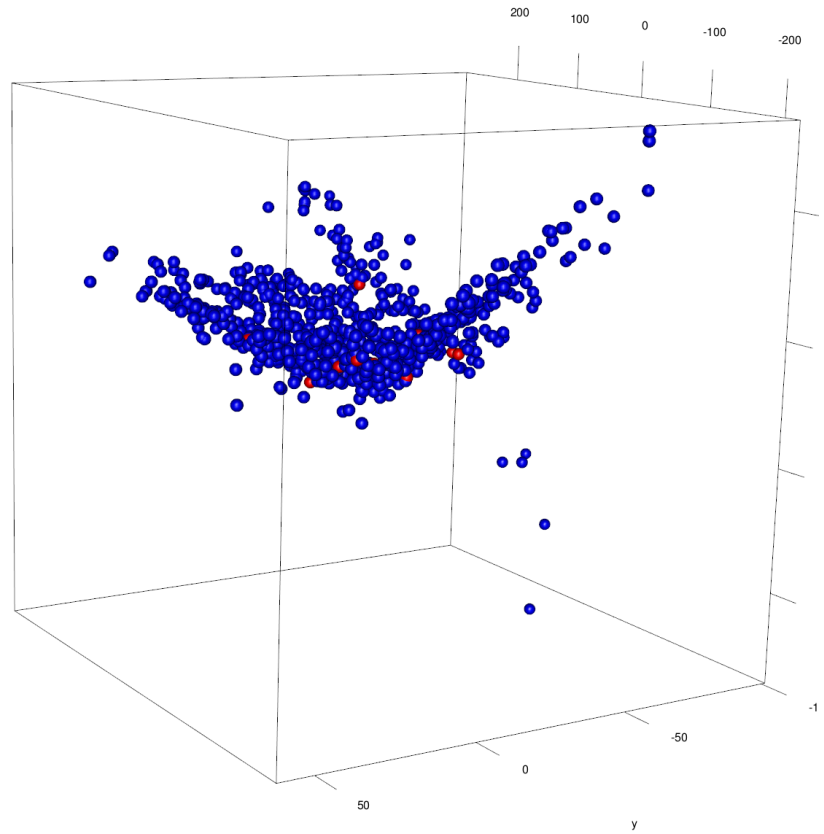


Figure 7: 3-dimensiju attēls k=40

Secinājumi

Interesants fenomens ir tas, ka ar k=1, lielākā sakarīgā punktu kopa ir visi 12 klases "Active" punkti, un iekļauts nav neviens "Inactive" punkts. Taču tiklīdz k>1, "Active" punkti tiek šķietami izkaisīti "Inactive" punktu jūrā. Šis fakts liek palauzīt galvu, taču ticamākais izskadrojums varētu būt šāds: kaut kur daudzdimensionālā telpā var atrast 2 hipervirsmas, starp kurām attālums visur ir lielāks nekā tas starp katriem kopas active "kaimiņiem" (tuvākajiem punktiem) taču mazāks nekā attālums starp punktiem, kas neatrodas blakus. Uz vienas no virsmām atrodas visi "Active" punkti, uz otras - "Inactive". Tāpēc ar k=1 "Active" punktu kaimiņi ir tikai citi kopas "Active" punkti, taču ar k>1 var atrast daudzus "Inactive" punktus, kas ir tuvāki nekā nākamie tuvākie "Active" punkti, jo "Inactive" kopa ir daudzkārt lielāka un "biezāka" nekā "Active" kopa.

Piemērs kā šāda struktūra varētu izskatīties divās dimensijās:

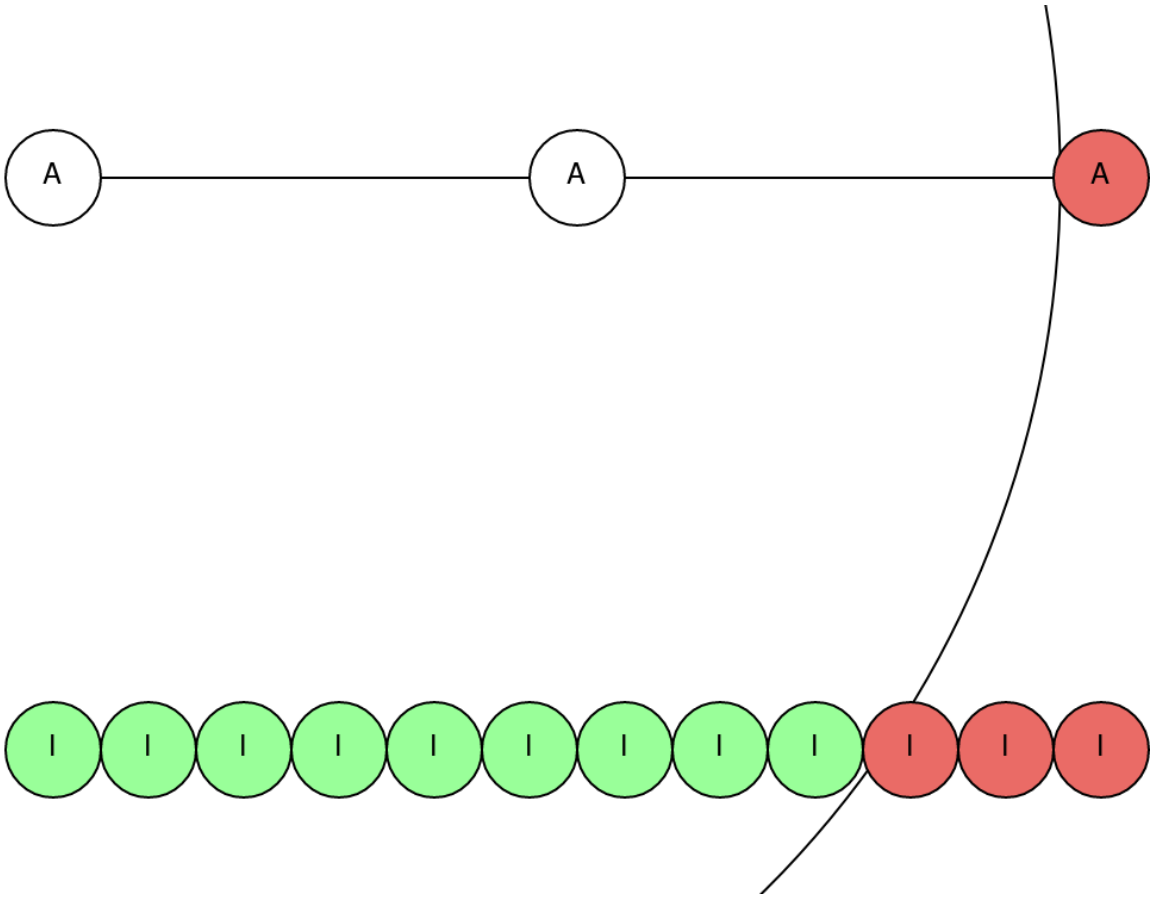


Figure 8: Divas 1-dimensionālas hipervirsmas 2-dimensiju telpā

Savukārt fakts, ka šī struktūra nav redzama ne 2-dimensiju, ne 3-dimensiju projekcijās varētu būt skaidrojams ar to, ka šo struktūru var "redzēt" tikai augstākās dimensijās - piemēram var salīdzināt 2-d un 3-d attēlu k=1 gadījumā (5.lpp), kur divi punkti ar ievērojamu attālumu tiek saspiesti vienā, ja tiek atmests īpašvektors, kas ir lielākā komponente to starpībā (punkti labajā apakšējā 2-d attēla stūrī saspiežas gandrīz kopā, kad tiek atmesta z-ass).