

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**MATEMATIKOS IR GAMTOS MOKSLŲ FAKULTETAS**

**Vytautas Kraujalis**

6 variantas

Studijų modulio

**P160M101 DAUGIAMATĖ STATISTINĖS ANALIZĖ**

**3 laboratorinis darbas**

**KAUNAS, 2021**

# UŽDUOTIS Nr. 1. Hierarchinis klasterizavimas

1. Imties sudarymas. Atrinkite 12 miestų (miestelių, gyvenviečių), kurių pavadinimai prasideda ta pačia raide kaip ir jūsų pavardė. Sudarykite atstumų matricą naudodamiesi atstumais keliais (galite naudotis maps.lt, project-osrm.org, google.com/maps ar kitu jums patogiu žemėlapiu).
2. Pavaizduokite žemėlapį su pasirinktais miestais.
3. Atlikite hierarchinį imties klasterizavimą (naudokite R funkciją hclust()). Panaudokite keturis klasterizavimo metodus (keisdami parametrą method). Palyginkite gautas dendrogramas. Kuris iš gautų rezultatų labiausiai atitinką miestų grupąvimą lyginant su jūsų subjektyviu požiūriu. Atsakymą pakomentuokite.
4. Pasirinkę vieną iš hierarchinio klasterizavimo metodu ir naudodami ilgiausių dendrogramos šakų būdu parinkite klasterių kiekį. Pateikite miestų klasterius.

# UŽDUOTIS Nr. 2. Daugiamačių skalių metodas

1. Naudodami užduoties nr. 1. imtį (miestų atstumų lentelę), sudarykite ir pateikite daugiamačių skalių metodo rezultatus naudodami vieną metrinį ir vieną ne metrinį daugiamačių skalių metodą. Palyginkite gautus rezultatus su realiu žemlėlapio vaizdu.

**UŽDUOTIS NR. 3. GAUSO skirstinių MIŠINIo KLASTERIZAVIMAS**

1. Sugeneruokite duomenų imtis atitinkančias Gauso skirstinių mišinį nurodyą variantų lentelėje. Imties dydžiai – 100, 500, 3000 taškų. Generavimui galite naudotis bet kokiomis, savo pasirašytomis, ar paketuose pateikiamomis funkcijomis.

Variantai:

|  |  |
| --- | --- |
| 1. | Ambrutis Andrius |
| 2. | Babkauskas Dominykas |
| 3. | Baliūnaitė Gintarė |
| 4. | Dzidolikas Vilius |
| 5. | Jablonskaitė Kamilija |
| 6. | Kraujalis Vytautas |
| 7. | Levinskas Matas |
| 8. | Lugas Dovydas |
| 9. | Pacevičius Rytis |
| 10. | Petrauskaitė Vaiva |
| 11. | Sprainaitis Mantas |
| 12. | Uzdila Ernestas |

1. Įvertinkite mišinio paramterus naudodami mclust paketą. Kovariacinių matricų struktūros parinkimui naudokite BIC kriterijų. Paaiškinkte kokia klasterių kovariacinė struktūra buvo parinkta. Palyginkite ją su tikrosiomis reikšmėmis.

**UŽDUOTIS NR. 4. k-vidurkių klasterizavimas**

**Miestų duomenys.**

1. Pirmos užduoties duomenis papildykite pasirinktų miestų koordinatėmis.
2. Panaudokite k-vidurkių metodą miestams klasterizuoti į 2–5 klasterius. Pateikite gautus klasterius. Gautą rezultatą palyginkite su anksčiau gautais hierarchinio klasterizavimo rezultatais (jei skaidytume į 2–5 klasterius pagal hierarchinį klasterizavima).

**Gauso mišinių modelis.**

1. Panaudokite k-vidurkių metodą antros užduoties duomenims kai imties dydis yra 500, kai klasterių kiekis yra 2–5.
2. Vizualizuokite gautus klasterius (pvz., pasirinkdami dvi pirmąsias pagrindines komponentes).
3. Nubraižykite atstumų kvadratų sumos klasterių viduje (within cluster sum of squares). Panaudokita alkūnės metodą optimaliam klasterio kiekio parinkimui.
4. Panaudokite vidutinį silueto matą optimaliam klasterio kiekio parinkimui.

Užduoties ataskaita pateikiama per <http://moodle.ktu.lt/> Ataskaitoje reikia pateikti visas užduotis atitinkantį **pilną R kodą**, bei jo įvykdymo rezultatą (tekstinę ar grafinę), atsakyti į užduotyse nurodytus klausimus, aprašyti sunkumus su kuriais susidūrėte ir jų sprendimo būdus.

# SPRENDIMAS

# UŽDUOTIS Nr. 1. Hierarchinis klasterizavimas

1. Imties sudarymas. Atrinkite 12 miestų (miestelių, gyvenviečių), kurių pavadinimai prasideda ta pačia raide kaip ir jūsų pavardė. Sudarykite atstumų matricą naudodamiesi atstumais keliais (galite naudotis maps.lt, project-osrm.org, google.com/maps ar kitu jums patogiu žemėlapiu).

library(openxlsx)  
  
atstumai <- read.xlsx("atstumai.xlsx", rowNames = TRUE)  
  
knitr::kable(atstumai)

|  | Kaunas | Klaipeda | Kedainiai | Kaisiadorys | Kupiskis | Kalvarija | Krokialaukis | Kapciamiestis | Krosna | Kacergine | Karmelava | Kruonis |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Kaunas | 0.0 | 215 | 63.1 | 42.6 | 156 | 84.4 | 66.7 | 134.0 | 86.9 | 16.7 | 13.5 | 40.3 |
| Klaipeda | 215.0 | 0 | 171.0 | 249.0 | 286 | 254.0 | 272.0 | 315.0 | 266.0 | 223.0 | 220.0 | 247.0 |
| Kedainiai | 63.1 | 171 | 0.0 | 98.1 | 137 | 139.0 | 121.0 | 182.0 | 142.0 | 71.4 | 69.4 | 36.4 |
| Kaisiadorys | 42.6 | 249 | 98.1 | 0.0 | 161 | 127.0 | 84.4 | 125.0 | 101.0 | 58.9 | 38.0 | 20.2 |
| Kupiskis | 156.0 | 286 | 137.0 | 161.0 | 0 | 233.0 | 215.0 | 283.0 | 236.0 | 165.0 | 145.0 | 190.0 |
| Kalvarija | 84.4 | 254 | 139.0 | 127.0 | 233 | 0.0 | 47.1 | 61.9 | 22.5 | 83.9 | 98.4 | 97.5 |
| Krokialaukis | 66.7 | 272 | 121.0 | 84.4 | 215 | 47.1 | 0.0 | 68.5 | 24.9 | 70.0 | 73.3 | 66.3 |
| Kapciamiestis | 134.0 | 315 | 182.0 | 125.0 | 283 | 61.9 | 68.5 | 0.0 | 47.1 | 130.0 | 141.0 | 107.0 |
| Krosna | 86.9 | 266 | 142.0 | 101.0 | 236 | 22.5 | 24.9 | 47.1 | 0.0 | 90.7 | 101.0 | 83.3 |
| Kacergine | 16.7 | 223 | 71.4 | 58.9 | 165 | 83.9 | 70.0 | 130.0 | 90.7 | 0.0 | 30.9 | 57.3 |
| Karmelava | 13.5 | 220 | 69.4 | 38.0 | 145 | 98.4 | 73.3 | 141.0 | 101.0 | 30.9 | 0.0 | 41.1 |
| Kruonis | 40.3 | 247 | 36.4 | 20.2 | 190 | 97.5 | 66.3 | 107.0 | 83.3 | 57.3 | 41.1 | 0.0 |

Sudaryta atstumų matrica tarp 12 miestų. Atstumai buvo sudaryti naudojanti google maps.

1. Pavaizduokite žemėlapį su pasirinktais miestais.

Miestų žemėlapis:

Map

Description automatically generated

Čia

| 1|Kaunas |

| 2|Klaipeda |

| 3|Kedainiai |

| 4|Kaisiadorys |

| 5|Kupiskis |

| 6|Kalvarija |

| 7|Krokialaukis |

| 8|Kapciamiestis |

| 9|Krosna |

| 10|Kacergine |

| 11|Karmelava |

| 12|Kruonis |

1. Atlikite hierarchinį imties klasterizavimą (naudokite R funkciją hclust()). Panaudokite keturis klasterizavimo metodus (keisdami parametrą method). Palyginkite gautas dendrogramas. Kuris iš gautų rezultatų labiausiai atitinką miestų grupąvimą lyginant su jūsų subjektyviu požiūriu. Atsakymą pakomentuokite.

dist\_mat <- dist(atstumai, method = "euclidean")  
hclust\_average <- hclust(dist\_mat, method = "average")  
hclust\_complete <- hclust(dist\_mat, method = "complete")  
hclust\_centroid <- hclust(dist\_mat, method = "centroid")  
hclust\_single <- hclust(dist\_mat, method = "single")  
  
  
png(file = "1uzd\_3\_average.png", width = 1200, height = 850)  
plot(hclust\_average, main = "Vidutinio atstumo metodas" , cex = 2)  
dev.off()

Chart, diagram, schematic

Description automatically generated

Naudodamiesi vidutinio atstumo metodu, gauname 3 didesnius klasterius: {Klaipėda ir Kupiškis} (2 miestai išskirtys, atitolę nuo visų kitų, tačiau neesantys vienas šalia kito), {Kapčiamiestis, Krokialaukis, Kalvarija ir Krosna} (Pietų Lietuva), {Kėdainiai, Kačerginė, Kaunas, Karmėlava, Kaišiadorys, Kruonis} (Vidurio Lietuva, Kauno rajonas)

png(file = "1uzd\_3\_complete.png", width = 1200, height = 850)  
plot(hclust\_complete, main = "Tolimiausio kaimyno metodas" , cex = 2)  
dev.off()

Chart, box and whisker chart

Description automatically generated

Naudodamiesi tolimiausio kaimyno metodu, gauname gana panašų vaizdą, kaip ir vidutinio atstumo metodu.

png(file = "1uzd\_3\_centroid.png", width = 1200, height = 850)  
plot(hclust\_centroid, main = "Atstumo tarp centru metodas" , cex = 2)  
dev.off()

Diagram, schematic

Description automatically generated

Atstumo tarp centrų metodas jau duoda kitokius rezultatus, šis metodas atskiria Klaipėdą nuo Kupiškio. Kiti miestai patenka į tuos pačius jau prieš tai matytus klasterius.

png(file = "1uzd\_3\_single.png", width = 1200, height = 850)  
plot(hclust\_single, main = "Artimiausio kaimyno metodas" , cex = 2)  
dev.off()

Chart

Description automatically generated

Artimiausio kaimyno metodas, kaip ir atstumo tarp centrų metodas, atskyrė Klaipėdą nuo Kupiškio.

Vertinant šiuos 4 metodus ir lyginant su pateiktu žemėlapiu, atrodo, jog artimiausio kaimyno metodas pateikė arčiausiai tiesos esantį vaizdą, kadangi pirmiausia buvo suklasterizuota Pietų ir Centro Lietuva bei atskirta Klaipėda nuo Kupiškio.

1. Pasirinkę vieną iš hierarchinio klasterizavimo metodu ir naudodami ilgiausių dendrogramos šakų būdu parinkite klasterių kiekį. Pateikite miestų klasterius.

Naudojant artimiausio kaimyno metodą gauname 4 klasterius: {Klaipėda}, {Kupiškis}, {Kapčiamiestis, Krokialaukis, Kalvarija, Krosna} ir {Kėdainiai, Kačerginė, Kauas, Karmėlava, Kaišiadorys, Kruonis}.

# UŽDUOTIS Nr. 2. Daugiamačių skalių metodas

1. Naudodami užduoties nr. 1. imtį (miestų atstumų lentelę), sudarykite ir pateikite daugiamačių skalių metodo rezultatus naudodami vieną metrinį ir vieną ne metrinį daugiamačių skalių metodą. Palyginkite gautus rezultatus su realiu žemlėlapio vaizdu.

# Metric MDS  
fit <- cmdscale(dist\_mat)

# plot solution  
png(file = "2uzd\_1\_Metric.png", width = 1200, height = 850)  
plot(fit, xlab="Koord. 1", ylab="Koord. 2", main="Metrinis daugiamačių skalių metodas", type="n" , cex = 2)  
text(fit, labels = colnames(atstumai), cex=2)  
dev.off()

Chart

Description automatically generated

Naudodami metrinį daugiamačių skalių metodą gavome vaizdą, panašų į žemėlapį. T.y. Klaipėda yra toliausiai nuo visų miestų, tuomet seka ir atsiskyręs Kupiškis, Kėdainiai esantys toliau nuo Kauno ir čia šiek tiek tolėliau.

library(MASS)  
# Sammon  
fit <- sammon(dist\_mat)

# plot solution  
png(file = "2uzd\_1\_nonmetric.png", width = 1200, height = 850)  
plot(fit$points, xlab="Koord. 1", ylab="Koord. 2", main="Ne metrinis daugiamačių skalių metodas", type="n", cex = 2)  
text(fit$points, labels = colnames(atstumai), cex = 2)   
dev.off()

Chart

Description automatically generated with medium confidence

Naudojantis ne metriniu daugiamačių skalių metodu, gavome gana panašų vaizdą į metrinį metodą, tačiau dabar ne taip arti vienas kito yra dauguma miestų („nesugrūsta“ į vieną vietą).

**UŽDUOTIS NR. 3. GAUSO skirstinių MIŠINIo KLASTERIZAVIMAS**

1. Sugeneruokite duomenų imtis atitinkančias Gauso skirstinių mišinį nurodyą variantų lentelėje. Imties dydžiai – 100, 500, 3000 taškų. Generavimui galite naudotis bet kokiomis, savo pasirašytomis, ar paketuose pateikiamomis funkcijomis.

p1 <- p2 <- p3 <- 1/3  
  
m1 <- c(0, 0, 0)  
m2 <- c(-9, -5, 1)  
m3 <- c(3.99618, 0, 6.930407)  
  
R1 <- matrix(  
 c(  
 3.8571, 0, -0.2162,  
 0, 8.344, 0,  
 -0.2162, 0, 4.8989  
 ),  
 nrow = 3, ncol = 3  
)  
R2 <- matrix(  
 c(  
 3.83, 2.281, 0,  
 2.281, 3.55, 0,  
 0, 0, 7.573  
 ),  
 nrow = 3, ncol = 3  
)  
R3 <- matrix(  
 c(  
 9.935, 0, 0,  
 0, 7.6078, -0.3615,  
 0, -0.3615, 7.7082  
 ),  
 nrow = 3, ncol = 3  
)

library(mvtnorm)  
library(mclust)  
  
n1 <- sum(runif(100) < p1)  
n2 <- sum(runif(100) < p1)  
n3 <- 100 - n1 - n2  
  
gauso\_ad\_100 <- rbind(rmvnorm(n1, m1, R1), rmvnorm(n2, m2, R2), rmvnorm(n3, m3, R3))  
  
n1 <- sum(runif(500) < p1)  
n2 <- sum(runif(500) < p1)  
n3 <- 500 - n1 - n2  
  
gauso\_ad\_500 <- rbind(rmvnorm(n1, m1, R1), rmvnorm(n2, m2, R2), rmvnorm(n3, m3, R3))  
  
n1 <- sum(runif(3000) < p1)  
n2 <- sum(runif(3000) < p1)  
n3 <- 3000 - n1 - n2  
  
gauso\_ad\_3000 <- rbind(rmvnorm(n1, m1, R1), rmvnorm(n2, m2, R2), rmvnorm(n3, m3, R3))

1. Įvertinkite mišinio paramterus naudodami mclust paketą. Kovariacinių matricų struktūros parinkimui naudokite BIC kriterijų. Paaiškinkte kokia klasterių kovariacinė struktūra buvo parinkta. Palyginkite ją su tikrosiomis reikšmėmis.

Kai N = 100

# Klasterizavimas  
fit <- Mclust(gauso\_ad\_100)  
  
# Rezultatai  
fit$BIC

## Top 3 models based on the BIC criterion:   
## VII,3 EII,3 VEI,3   
## -1644.927 -1647.320 -1649.828

Pagal BIC kriterijų modelio struktūra buvo parinkta VII

fit$parameters

## $mean  
## [,1] [,2] [,3]  
## [1,] -0.3329913 -8.465281 4.6595946  
## [2,] 0.4449511 -4.605904 -0.2719289  
## [3,] 0.2130196 1.204083 7.1331373

Palyginus su tikromis reikšmėmis:

Calendar

Description automatically generated with medium confidence (transponuoti tik reikėtų)

Gauti vidurkiai yra gana panašūs į tikrąsias reikšmes.

## $variance  
## $variance$modelName  
## [1] "VII"  
##   
## [,1] [,2] [,3]  
## [1,] 5.577353 0.000000 0.000000  
## [2,] 0.000000 5.577353 0.000000  
## [3,] 0.000000 0.000000 5.577353  
##   
## , , 2  
##   
## [,1] [,2] [,3]  
## [1,] 4.21924 0.00000 0.00000  
## [2,] 0.00000 4.21924 0.00000  
## [3,] 0.00000 0.00000 4.21924  
##   
## , , 3  
##   
## [,1] [,2] [,3]  
## [1,] 8.711706 0.000000 0.000000  
## [2,] 0.000000 8.711706 0.000000  
## [3,] 0.000000 0.000000 8.711706

Kovariacinių matricų struktūra buvo parinkta pagal VII (sferinė, nelygus tūris)

Vadinasi grafiškai turėtumėme ne vienodo dydžio „apskritimus“.

Tikrosios reikšmės:

A picture containing text, scoreboard, battery

Description automatically generated

Kaip matom pagal tikrąsias reikšmes, gautos kovariacinės matricos ne visai sutampa dėl parinktos kovariacinių matricų struktūros.

Kai N = 500

# Klasterizavimas  
fit <- Mclust(gauso\_ad\_500)  
  
# Rezultatai  
fit$BIC  
## Top 3 models based on the BIC criterion:   
## VVE,3 VVI,3 VVV,3   
## -8182.435 -8186.094 -8200.593

Pagal BIC kriterijų modelio struktūra buvo parinkta VVE

fit$parameters  
##   
## $mean  
## [,1] [,2] [,3]  
## [1,] 4.4473328 0.1464948 -8.885036  
## [2,] -0.0831896 -0.1837244 -5.025990  
## [3,] 7.3787449 0.2635029 1.150784  
##

Palyginus su tikromis reikšmėmis:

Calendar

Description automatically generated with medium confidence (transponuoti tik reikėtų)

Gauti vidurkiai yra tikrai gana panašūs į tikrąsias reikšmes. (kitaip išdėstyti tik)

## $variance  
## $variance$modelName  
## [1] "VVE"  
##   
## [,1] [,2] [,3]  
## [1,] 9.50944995 -0.33739615 -0.01957925  
## [2,] -0.33739615 9.56299681 0.02728906  
## [3,] -0.01957925 0.02728906 8.43931815  
##   
## , , 2  
##   
## [,1] [,2] [,3]  
## [1,] 6.61452139 0.09297222 -0.01024406  
## [2,] 0.09297222 6.59942888 0.02008945  
## [3,] -0.01024406 0.02008945 5.57884732  
##   
## , , 3  
##   
## [,1] [,2] [,3]  
## [1,] 3.6875298 1.7717342 0.1065395  
## [2,] 1.7717342 3.4064250 -0.1498762  
## [3,] 0.1065395 -0.1498762 9.6239162

Kovariacinių matricų struktūra buvo parinkta pagal VVE (elipsinė, lygi kryptis)

Vadinasi grafiškai turėtumėme nebūtinai vienodo dydžio elipses tačiau venodai „susiplojusias“ į tą pačią pusę.

Tikrosios reikšmės:

A picture containing text, scoreboard, battery

Description automatically generated

Kaip matom pagal tikrąsias reikšmes, gautos kovariacinės matricos sutampa geriau, nei su 100 stebinių, tačiau vis dar skirtumai matosi. (irgi reikia atkreipti dėmesį, kad kovariacinės matricos išdėstytos ne ta pačia tvarka kaip mūsų tikrosios).

Kai N = 3000

# Klasterizavimas  
fit <- Mclust(gauso\_ad\_3000)  
  
# Rezultatai  
fit$BIC  
## Top 3 models based on the BIC criterion:   
## VVV,3 VVE,4 VVE,3   
## -47418.41 -47457.22 -47472.41

Pagal BIC kriterijų modelio struktūra buvo parinkta VVV

fit$parameters

##   
## $mean  
## [,1] [,2] [,3]  
## [1,] 3.90108535 0.08003846 -9.029992  
## [2,] -0.05507408 -0.05166000 -5.005980  
## [3,] 7.04893743 -0.06535605 1.079669  
##

Palyginus su tikromis reikšmėmis:

Calendar

Description automatically generated with medium confidence (transponuoti tik reikėtų)

Gauti vidurkiai yra beveik identiški tikriesiems vidurkiams (kitaip išdėstyti tik).

## $variance  
## $variance$modelName  
## [1] "VVV"  
##   
##   
## $variance$sigma  
## , , 1  
##   
## [,1] [,2] [,3]  
## [1,] 9.4610654 -0.1875229 -0.2992911  
## [2,] -0.1875229 7.6655327 -0.2179045  
## [3,] -0.2992911 -0.2179045 7.7573256  
##   
## , , 2  
##   
## [,1] [,2] [,3]  
## [1,] 4.0056658 -0.21706320 -0.28465563  
## [2,] -0.2170632 8.17609994 0.03971236  
## [3,] -0.2846556 0.03971236 5.05975660  
##   
## , , 3  
##   
## [,1] [,2] [,3]  
## [1,] 3.6390846 2.16579745 -0.25241486  
## [2,] 2.1657975 3.43463410 -0.09844315  
## [3,] -0.2524149 -0.09844315 7.63504270  
##

Kovariacinių matricų struktūra buvo parinkta pagal VVV (elipsinė, skirtingi tūriai, pavidalai ir kryptis)

Vadinasi grafiškai turėtumėme elipses, tačiau įvairių formų, krypčių ir t.t.

Tikrosios reikšmės:

A picture containing text, scoreboard, battery

Description automatically generated

Kaip matom pagal tikrąsias reikšmes, gautos kovariacinės matricos sutampa labai panašiai su tikrosiomis reikšmėmis. (irgi reikia atkreipti dėmesį, kad kovariacinės matricos išdėstytos ne ta pačia tvarka kaip mūsų tikrosios).

**UŽDUOTIS NR. 4. k-vidurkių klasterizavimas**

**Miestų duomenys.**

1. Pirmos užduoties duomenis papildykite pasirinktų miestų koordinatėmis.

library(dplyr)  
koordinates <- read.xlsx("koordinates.xlsx", rowNames = TRUE) %>%   
 mutate\_all(as.numeric)  
  
knitr::kable(koordinates)

Miestų koordinatės:

|  | Pirma\_Koord | Antra\_Koord |
| --- | --- | --- |
| Kaunas | 54.89838 | 23.94513 |
| Klaipeda | 55.72537 | 21.15102 |
| Kedainiai | 55.28855 | 23.96746 |
| Kaisiadorys | 54.86270 | 24.46709 |
| Kupiskis | 55.83987 | 24.98481 |
| Kalvarija | 54.41565 | 23.22712 |
| Krokialaukis | 54.43655 | 23.76704 |
| Kapciamiestis | 54.00249 | 23.65649 |
| Krosna | 54.37853 | 23.53013 |
| Kacergine | 54.93359 | 23.72001 |
| Karmelava | 54.96913 | 24.06590 |
| Kruonis | 54.75844 | 24.24072 |

koordinates <- scale(koordinates)  
  
knitr::kable(koordinates)

Miestų koordinates „suscale‘inam“:

|  | Pirma\_Koord | Antra\_Koord |
| --- | --- | --- |
| Kaunas | 0.0415059 | 0.2342620 |
| Klaipeda | 1.5600000 | -2.7652696 |
| Kedainiai | 0.7579187 | 0.2582310 |
| Kaisiadorys | -0.0240006 | 0.7945996 |
| Kupiskis | 1.7702244 | 1.3503825 |
| Kalvarija | -0.8448543 | -0.5365378 |
| Krokialaukis | -0.8064697 | 0.0430848 |
| Kapciamiestis | -1.6034746 | -0.0755945 |
| Krosna | -0.9130118 | -0.2112462 |
| Kacergine | 0.1061680 | -0.0074080 |
| Karmelava | 0.1714297 | 0.3639121 |
| Kruonis | -0.2154357 | 0.5515842 |

1. Panaudokite k-vidurkių metodą miestams klasterizuoti į 2–5 klasterius. Pateikite gautus klasterius. Gautą rezultatą palyginkite su anksčiau gautais hierarchinio klasterizavimo rezultatais (jei skaidytume į 2–5 klasterius pagal hierarchinį klasterizavima).

2 klasteriai:

library(ggpubr)  
library(tibble)  
kmeans\_2 <- kmeans(koordinates, centers = 2)  
df <- koordinates %>%   
 as.data.frame() %>%   
 rownames\_to\_column("Miestas") %>%   
 bind\_cols(klasteris = factor(kmeans\_2$cluster))  
  
ggscatter(  
 df, x = "Pirma\_Koord", y = "Antra\_Koord",   
 color = "klasteris", palette = "npg", ellipse = TRUE, ellipse.type = "convex",  
 shape = "klasteris", size = 1.5, legend = "right", ggtheme = theme\_minimal(),  
 label = "Miestas", repel = TRUE, label.rectangle = TRUE, font.label = c(14)  
) +  
 stat\_mean(aes(color = klasteris), size = 4)

ggsave(filename = "4uzd\_2\_2klasteriai.png", width = 14, height = 7, units = "in", bg = "white")

A picture containing timeline

Description automatically generated

Gauname 2 klasterius: Klaipėdą ir Kiti, tą gautume ir naudodamiesi artimiausių kaimynų hierarchinį metodą.

3 klasteriai:

kmeans\_3 <- kmeans(koordinates, centers = 3)  
df <- koordinates %>%   
 as.data.frame() %>%   
 rownames\_to\_column("Miestas") %>%   
 bind\_cols(klasteris = factor(kmeans\_3$cluster))  
  
ggscatter(  
 df, x = "Pirma\_Koord", y = "Antra\_Koord",   
 color = "klasteris", palette = "npg", ellipse = TRUE, ellipse.type = "convex",  
 shape = "klasteris", size = 1.5, legend = "right", ggtheme = theme\_minimal(),  
 label = "Miestas", repel = TRUE, label.rectangle = TRUE, font.label = c(14)  
) +  
 stat\_mean(aes(color = klasteris), size = 4)

ggsave(filename = "4uzd\_2\_3klasteriai.png", width = 14, height = 7, units = "in", bg = "white")

Chart

Description automatically generated

Gauname 3 klasterius: {Klaipėda}, {Krosnas, Krokialaukis, Kalvarija, Kapčiamiesti} ir Kiti, tokių klasterių su prieš tai naudotais hierarchiniais metodais negautume, kadangi šiame metode Kupiškis priskiriamas prie Kauno klasterio, ko hierarchiniame klasterizavime neturėjome.

4 klasteriai:

kmeans\_4 <- kmeans(koordinates, centers = 4)  
df <- koordinates %>%   
 as.data.frame() %>%   
 rownames\_to\_column("Miestas") %>%   
 bind\_cols(klasteris = factor(kmeans\_4$cluster))  
  
ggscatter(  
 df, x = "Pirma\_Koord", y = "Antra\_Koord",   
 color = "klasteris", palette = "npg", ellipse = TRUE, ellipse.type = "convex",  
 shape = "klasteris", size = 1.5, legend = "right", ggtheme = theme\_minimal(),  
 label = "Miestas", repel = TRUE, label.rectangle = TRUE, font.label = c(14)  
) +  
 stat\_mean(aes(color = klasteris), size = 4)

ggsave(filename = "4uzd\_2\_4klasteriai.png", width = 14, height = 7, units = "in", bg = "white")

Chart, scatter chart

Description automatically generated

Naudodami 4 klasterius, gauname: {Klaipėda}, {Kupiškis} ir 2 kiti klasteriai. Tokį klasterizavimą gauname ir naudodamiesi artimiausio kaimyno metodu, kuomet 2 tolimiausi miestai yra atskiriami nuo Pietų Lietuvos ir Centro Lietuvos klasterių.

5 klasteriai:

kmeans\_5 <- kmeans(koordinates, centers = 5)  
df <- koordinates %>%   
 as.data.frame() %>%   
 rownames\_to\_column("Miestas") %>%   
 bind\_cols(klasteris = factor(kmeans\_5$cluster))  
  
ggscatter(  
 df, x = "Pirma\_Koord", y = "Antra\_Koord",   
 color = "klasteris", palette = "npg", ellipse = TRUE, ellipse.type = "convex",  
 shape = "klasteris", size = 1.5, legend = "right", ggtheme = theme\_minimal(),  
 label = "Miestas", repel = TRUE, label.rectangle = TRUE, font.label = c(14)  
) +  
 stat\_mean(aes(color = klasteris), size = 4)

ggsave(filename = "4uzd\_2\_5klasteriai.png", width = 14, height = 7, units = "in", bg = "white")

**Chart, scatter chart

Description automatically generated**

Naudodami 5 klasterius, gauname: {Klaipėda}, {Kupiškis}, {Kapčiamiestis} ir 2 kiti klasteriai. Tokio klasterizavimo hierarchiniu metodu negauname, kadangi šiuo atveju Kapčiamiestis yra atskiriamas nuo Krosnos klasterio.

**Gauso mišinių modelis.**

1. Panaudokite k-vidurkių metodą antros užduoties duomenims kai imties dydis yra 500, kai klasterių kiekis yra 2–5.

kmeans\_2 <- kmeans(gauso\_ad\_500, centers = 2)  
kmeans\_3 <- kmeans(gauso\_ad\_500, centers = 3)  
kmeans\_4 <- kmeans(gauso\_ad\_500, centers = 4)  
kmeans\_5 <- kmeans(gauso\_ad\_500, centers = 5)

1. Vizualizuokite gautus klasterius (pvz., pasirinkdami dvi pirmąsias pagrindines komponentes).

2 klasteriai:

df <- gauso\_ad\_500 %>%   
 as.data.frame() %>%   
 bind\_cols(klasteris = factor(kmeans\_2$cluster))  
  
ggscatter(  
 df, x = "V1", y = "V2",   
 color = "klasteris", palette = "npg", ellipse = TRUE, ellipse.type = "convex",  
 shape = "klasteris", size = 1.5, legend = "right", ggtheme = theme\_minimal()  
) +  
 stat\_mean(aes(color = klasteris), size = 4)

ggsave(filename = "4uzd\_4\_2klasteriai.png", width = 14, height = 7, units = "in", bg = "white")

Chart, scatter chart

Description automatically generated

Naudodami 2 klasterius, gauname pakankamai „gražų“ klasterių atskyrimą. 2 klasteriai šiuo atveju turėtų būti optimalus atsakymas.

3 klasteriai:

df <- gauso\_ad\_500 %>%   
 as.data.frame() %>%   
 bind\_cols(klasteris = factor(kmeans\_3$cluster))  
  
ggscatter(  
 df, x = "V1", y = "V2",   
 color = "klasteris", palette = "npg", ellipse = TRUE, ellipse.type = "convex",  
 shape = "klasteris", size = 1.5, legend = "right", ggtheme = theme\_minimal()  
) +  
 stat\_mean(aes(color = klasteris), size = 4)

ggsave(filename = "4uzd\_4\_3klasteriai.png", width = 14, height = 7, units = "in", bg = "white")

Chart, scatter chart

Description automatically generated

Naudodami 3 klasterius, vienas klasteris visiškai atsiskiria nuo kitų, tačiau likę 2 klasteriai persipina vienas su kitu.

4 klasteriai:

df <- gauso\_ad\_500 %>%   
 as.data.frame() %>%   
 bind\_cols(klasteris = factor(kmeans\_4$cluster))  
  
ggscatter(  
 df, x = "V1", y = "V2",   
 color = "klasteris", palette = "npg", ellipse = TRUE, ellipse.type = "convex",  
 shape = "klasteris", size = 1.5, legend = "right", ggtheme = theme\_minimal()  
) +  
 stat\_mean(aes(color = klasteris), size = 4)

ggsave(filename = "4uzd\_4\_4klasteriai.png", width = 14, height = 7, units = "in", bg = "white")

Chart, scatter chart

Description automatically generated

Naudodami 4 klasterius matome tą pačią situaciją, vienas klasteris atsiskiria gražiai, tačiau likę 3 klasteriai persipina vieni su kitais.

5 klasteriai:

df <- gauso\_ad\_500 %>%   
 as.data.frame() %>%   
 bind\_cols(klasteris = factor(kmeans\_5$cluster))  
  
ggscatter(  
 df, x = "V1", y = "V2",   
 color = "klasteris", palette = "npg", ellipse = TRUE, ellipse.type = "convex",  
 shape = "klasteris", size = 1.5, legend = "right", ggtheme = theme\_minimal()  
) +  
 stat\_mean(aes(color = klasteris), size = 4)

ggsave(filename = "4uzd\_4\_5klasteriai.png", width = 14, height = 7, units = "in", bg = "white")

Chart, scatter chart

Description automatically generated

Naudodami 5 klasterius jau gauname visus klasterius persimaišius, 2 klasteriai beveik sutampa vienas su kitu.

1. Nubraižykite atstumų kvadratų sumos klasterių viduje (within cluster sum of squares). Panaudokita alkūnės metodą optimaliam klasterio kiekio parinkimui.

kvad\_sumos <- NULL  
klasteriai <- 1:10  
  
for(i in 1:length(klasteriai)){  
 kvad\_sumos[i] <- kmeans(gauso\_ad\_500, centers = i)$tot.withinss  
}  
  
png(file = "4uzd\_5.png", width = 1200, height = 850)  
plot(klasteriai, kvad\_sumos, type = "l", cex = 2)  
dev.off()

Chart, line chart

Description automatically generated

Naudodamiesi alkūnės metodu, sakytume, jog optimalus klasterių kiekis turėtų būti 3.

1. Panaudokite vidutinį silueto matą optimaliam klasterio kiekio parinkimui.

library(factoextra)  
  
png(file = "6uzd.png", width = 1200, height = 850)  
fviz\_nbclust(gauso\_ad\_500, kmeans, method = "silhouette")  
dev.off()

Chart, line chart

Description automatically generated

Vidutinio silueto mato metodu gautas optimalus klasterių kiekis – 2. Tokią išvadą ir padarėme žiūrėdami į sklaidos diagramas.

Literatūra

1. mclust package homepage. <https://sites.stat.washington.edu/mclust/>
2. Luca Scrucca. A quick tour of mclust, 2020. <https://cran.r-project.org/web/packages/mclust/vignettes/mclust.html>
3. Luca Scrucca, Michael Fop, T. Brendan Murphy and Adrian E. Raftery (2016). mclust 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models. *The R Journal*, Vol. 8, No. 1, p. 289 – 317. <https://journal.r-project.org/archive/2016/RJ-2016-021/RJ-2016-021.pdf>
4. K-means Cluster Analysis. UC Business Analytics R Programming Guide. https://uc-r.github.io/kmeans\_clustering
5. Kisung You (2019). *L2 distance between Gaussian Mixture Models*. <http://kyoustat.com/pdf/note004gmml2.pdf>