Daugiamatė Statistinė Analizė Lab 3, 6 variantas

Vytautas Kraujalis

2021-12-04

Table of Contents

# 1 Uzd. 1

## 1.1 1

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 |

## 1.2 3

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()

## png   
## 2

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

## png   
## 2

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

## png   
## 2

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

## png   
## 2

# 2 Uzd. 2

## 2.1 1

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

## [,1] [,2]  
## Kaunas 100.89619 -99.26631  
## Klaipeda -516.58514 94.34312  
## Kedainiai -42.27535 -122.38483  
## Kaisiadorys 74.26353 -81.78268  
## Kupiskis -303.22271 -103.32623  
## Kalvarija 93.56191 135.24235  
## Krokialaukis 133.60965 76.92615  
## Kapciamiestis 56.70982 217.05769  
## Krosna 120.04777 139.25792  
## Kacergine 87.73094 -76.67180  
## Karmelava 83.19294 -110.58741  
## Kruonis 112.07045 -68.80797

# 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()

## png   
## 2

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

## Initial stress : 0.03201  
## stress after 10 iters: 0.01127, magic = 0.461  
## stress after 20 iters: 0.01034, magic = 0.500  
## stress after 30 iters: 0.01029, magic = 0.500

fit

## $points  
## [,1] [,2]  
## Kaunas 124.28692 -120.28729  
## Klaipeda -521.12938 150.98382  
## Kedainiai -45.65293 -142.61948  
## Kaisiadorys 50.58703 -53.14605  
## Kupiskis -336.27692 -181.07456  
## Kalvarija 79.38121 139.42624  
## Krokialaukis 147.62128 85.88543  
## Kapciamiestis 15.77810 235.18675  
## Krosna 127.06088 150.31379  
## Kacergine 111.04734 -87.24227  
## Karmelava 91.97892 -127.31131  
## Kruonis 155.31754 -50.11507  
##   
## $stress  
## [1] 0.01028675  
##   
## $call  
## sammon(d = 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()

## png   
## 2

# 3 Uzd. 3

## 3.1 1

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))

## 3.2 2

### 3.2.1 N = 100

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

## Bayesian Information Criterion (BIC):   
## EII VII EEI VEI EVI VVI EEE  
## 1 -1764.181 -1764.181 -1741.649 -1741.649 -1741.649 -1741.649 -1684.134  
## 2 -1673.285 -1642.666 -1663.741 -1634.318 -1666.050 -1643.104 -1653.566  
## 3 -1629.303 -1628.659 -1634.786 -1631.367 -1639.147 -1643.728 -1644.050  
## 4 -1632.179 -1640.439 -1635.223 -1643.597 -1652.473 -1662.315 -1639.109  
## 5 -1645.336 -1659.012 -1648.179 -1661.253 -1663.117 -1676.272 -1635.415  
## 6 -1660.274 -1673.484 -1654.282 -1667.627 -1679.766 -1710.595 -1646.861  
## 7 -1675.475 -1684.095 -1675.817 -1686.633 -1704.682 -1718.693 -1678.662  
## 8 -1693.566 -1691.679 -1685.045 -1700.270 -1732.020 -1726.000 -1680.301  
## 9 -1705.491 -1711.679 -1696.376 -1711.018 -1756.318 -1758.692 -1688.280  
## VEE EVE VVE EEV VEV EVV VVV  
## 1 -1684.134 -1684.134 -1684.134 -1684.134 -1684.134 -1684.134 -1684.134  
## 2 -1632.443 -1646.374 -1621.172 -1638.319 -1629.369 -1644.889 -1629.590  
## 3 -1639.971 -1638.522 -1636.778 -1626.378 -1652.460 -1641.846 -1659.435  
## 4 -1643.317 -1628.140 -1646.592 -1654.162 -1668.489 -1658.231 -1673.106  
## 5 -1651.318 -1638.584 -1664.682 -1661.421 -1694.844 -1682.544 -1691.907  
## 6 -1666.289 -1670.457 -1692.577 -1692.392 -1709.252 -1716.939 -1719.003  
## 7 -1680.544 -1677.812 -1719.409 -1715.864 -1744.992 -1752.574 -1763.752  
## 8 -1709.539 -1716.880 -1740.761 -1741.569 -1738.684 -1785.050 -1770.912  
## 9 -1716.324 -1726.103 NA -1760.518 -1771.255 -1805.789 -1788.946  
##   
## Top 3 models based on the BIC criterion:   
## VVE,2 EEV,3 EVE,4   
## -1621.172 -1626.378 -1628.140

fit$parameters

## $pro  
## [1] 0.6050763 0.3949237  
##   
## $mean  
## [,1] [,2]  
## [1,] 1.51618738 -8.859875  
## [2,] -0.04445203 -4.963482  
## [3,] 2.67472157 1.141836  
##   
## $variance  
## $variance$modelName  
## [1] "VVE"  
##   
## $variance$d  
## [1] 3  
##   
## $variance$G  
## [1] 2  
##   
## $variance$sigma  
## , , 1  
##   
## [,1] [,2] [,3]  
## [1,] 9.31289830 -0.00283276 6.048632  
## [2,] -0.00283276 7.99957526 2.211384  
## [3,] 6.04863218 2.21138361 19.320099  
##   
## , , 2  
##   
## [,1] [,2] [,3]  
## [1,] 4.6438977 1.8964204 0.1236338  
## [2,] 1.8964204 2.2365968 -0.3676914  
## [3,] 0.1236338 -0.3676914 5.5416047  
##   
##   
## $variance$scale  
## [1] 10.326061 3.326017  
##   
## $variance$shape  
## [,1] [,2]  
## [1,] 0.5778840 1.719470  
## [2,] 0.7961347 0.349012  
## [3,] 2.1735657 1.666345  
##   
## $variance$orientation  
## [,1] [,2] [,3]  
## [1,] 0.7740142 -0.47875287 0.4143643  
## [2,] 0.4666890 0.87364451 0.1376468  
## [3,] -0.4279059 0.08683869 0.8996419  
##   
##   
## $Vinv  
## NULL

### 3.2.2 N = 500

# Klasterizavimas  
fit <- Mclust(gauso\_ad\_500)  
  
# Rezultatai  
fit$BIC

## Bayesian Information Criterion (BIC):   
## EII VII EEI VEI EVI VVI EEE  
## 1 -8873.401 -8873.401 -8725.212 -8725.212 -8725.212 -8725.212 -8417.570  
## 2 -8326.091 -8209.535 -8262.086 -8140.374 -8273.717 -8151.118 -8199.914  
## 3 -8075.708 -8052.798 -8084.917 -8057.788 -8073.148 -8036.610 -8096.715  
## 4 -8086.673 -8062.248 -8094.345 -8065.652 -8070.868 -8077.606 -8104.773  
## 5 -8082.870 -8075.101 -8080.275 -8075.399 -8096.125 -8096.849 -8062.614  
## 6 -8100.687 -8079.815 -8103.955 -8089.481 -8113.808 -8083.942 -8082.699  
## 7 -8078.130 -8101.112 -8089.406 -8113.059 -8109.926 -8118.443 -8081.946  
## 8 -8093.680 -8098.389 -8100.779 -8103.746 -8138.393 -8123.060 -8111.788  
## 9 -8110.115 -8117.572 -8109.858 -8118.582 -8163.905 -8127.173 -8103.175  
## VEE EVE VVE EEV VEV EVV VVV  
## 1 -8417.570 -8417.570 -8417.570 -8417.570 -8417.570 -8417.570 -8417.570  
## 2 -8097.735 -8161.864 -8058.819 -8184.109 -8061.011 -8151.698 -8021.507  
## 3 -8069.189 -8022.533 -7977.212 -8047.712 -8008.317 -8035.943 -7990.197  
## 4 -8074.224 -8039.844 -7996.686 -8044.522 -8027.059 -8050.197 -8041.930  
## 5 -8068.624 -8038.768 -8027.359 -8046.642 -8054.925 -8076.211 -8084.408  
## 6 -8071.802 -8063.403 -8058.735 -8083.664 -8092.385 -8126.189 -8125.763  
## 7 -8089.900 -8065.601 -8096.673 -8092.459 -8124.285 -8149.461 -8169.303  
## 8 -8110.760 -8102.524 -8122.512 -8135.440 -8160.767 -8204.191 -8209.967  
## 9 -8108.972 -8150.464 -8139.197 -8157.710 -8206.765 -8210.371 -8263.556  
##   
## Top 3 models based on the BIC criterion:   
## VVE,3 VVV,3 VVE,4   
## -7977.212 -7990.197 -7996.686

fit$parameters

## $pro  
## [1] 0.3269863 0.3713448 0.3016689  
##   
## $mean  
## [,1] [,2] [,3]  
## [1,] 0.3036309 3.8699504 -9.209576  
## [2,] 0.1240036 -0.2317915 -5.289481  
## [3,] -0.0548821 6.8243924 1.089706  
##   
## $variance  
## $variance$modelName  
## [1] "VVE"  
##   
## $variance$d  
## [1] 3  
##   
## $variance$G  
## [1] 3  
##   
## $variance$sigma  
## , , 1  
##   
## [,1] [,2] [,3]  
## [1,] 6.6953864 -0.8538646 -0.2360190  
## [2,] -0.8538646 6.9221277 -0.3133368  
## [3,] -0.2360190 -0.3133368 3.6099535  
##   
## , , 2  
##   
## [,1] [,2] [,3]  
## [1,] 8.44750435 -1.09745281 0.05010496  
## [2,] -1.09745281 8.76026854 -0.01067991  
## [3,] 0.05010496 -0.01067991 7.68418585  
##   
## , , 3  
##   
## [,1] [,2] [,3]  
## [1,] 3.85290442 2.3562857 0.05010543  
## [2,] 2.35628565 3.1909038 0.19782929  
## [3,] 0.05010543 0.1978293 6.92040515  
##   
##   
## $variance$scale  
## [1] 5.468212 8.239486 3.597180  
##   
## $variance$shape  
## [,1] [,2] [,3]  
## [1,] 1.4029690 1.1788659 0.3169527  
## [2,] 1.0988548 0.9090880 1.6330956  
## [3,] 0.6486518 0.9331032 1.9319411  
##   
## $variance$orientation  
## [,1] [,2] [,3]  
## [1,] 0.65541419 0.7477731 0.1061492  
## [2,] -0.75500058 0.6449264 0.1185077  
## [3,] 0.02015848 -0.1578144 0.9872630  
##   
##   
## $Vinv  
## NULL

### 3.2.3 N = 3000

# Klasterizavimas  
fit <- Mclust(gauso\_ad\_3000)  
  
# Rezultatai  
fit$BIC

## Bayesian Information Criterion (BIC):   
## EII VII EEI VEI EVI VVI EEE  
## 1 -53041.50 -53041.50 -52054.40 -52054.40 -52054.40 -52054.40 -49899.36  
## 2 -49782.62 -49050.76 -49331.35 -48570.33 -49339.11 -48558.62 -48775.82  
## 3 -48074.90 -47887.38 -48062.21 -47840.44 -47857.77 -47617.83 -48020.33  
## 4 -48061.80 -47845.28 -48039.70 -47781.39 -47904.95 -47342.00 -47945.28  
## 5 -48061.30 -47689.23 -48066.82 -47703.82 -47878.12 -47385.55 -47978.15  
## 6 -47912.20 -47705.31 -47820.96 -47719.30 -47720.00 -47332.36 -47592.94  
## 7 -47855.56 -47725.61 -47921.86 -47759.30 -47690.03 -47380.74 -47577.83  
## 8 -47865.48 -47643.21 -47947.98 -47606.25 -47696.77 -47429.31 -47609.70  
## 9 -47849.24 -47678.76 -47765.69 -47554.85 -47659.02 -47483.34 -47527.24  
## VEE EVE VVE EEV VEV EVV VVV  
## 1 -49899.36 -49899.36 -49899.36 -49899.36 -49899.36 -49899.36 -49899.36  
## 2 -48099.34 -48541.69 -47877.00 -48578.86 -47913.87 -48415.50 -47678.28  
## 3 -47733.73 -47580.08 -47285.46 -47657.11 -47445.54 -47490.29 -47208.82  
## 4 -47615.39 -47535.81 -47249.91 -47656.90 -47452.38 -47543.47 -47287.78  
## 5 -47519.83 -47582.75 -47299.73 -47711.25 -47419.64 -47615.71 -47361.46  
## 6 -47413.46 -47483.46 -47341.72 -47479.58 -47440.80 -47491.83 -47437.17  
## 7 -47386.21 -47445.57 -47412.94 -47474.57 -47460.54 -47513.31 -47515.28  
## 8 -47395.20 -47494.09 -47434.28 -47532.00 -47508.18 -47619.13 -47587.57  
## 9 -47405.68 -47520.45 -47482.93 -47536.00 -47562.73 -47629.66 -47649.85  
##   
## Top 3 models based on the BIC criterion:   
## VVV,3 VVE,4 VVE,3   
## -47208.82 -47249.91 -47285.46

fit$parameters

## $pro  
## [1] 0.3094083 0.3440968 0.3464949  
##   
## $mean  
## [,1] [,2] [,3]  
## [1,] 4.121913108 -9.0056279 0.12256144  
## [2,] -0.000482941 -4.9517478 -0.06650492  
## [3,] 7.046947853 0.9712856 0.09815046  
##   
## $variance  
## $variance$modelName  
## [1] "VVV"  
##   
## $variance$d  
## [1] 3  
##   
## $variance$G  
## [1] 3  
##   
## $variance$sigma  
## , , 1  
##   
## [,1] [,2] [,3]  
## [1,] 9.8089688 0.4258644 -0.6188314  
## [2,] 0.4258644 8.0773424 -0.2106697  
## [3,] -0.6188314 -0.2106697 7.1718823  
##   
## , , 2  
##   
## [,1] [,2] [,3]  
## [1,] 3.8652253 2.41670260 0.09235440  
## [2,] 2.4167026 3.72515550 0.05769822  
## [3,] 0.0923544 0.05769822 7.36726398  
##   
## , , 3  
##   
## [,1] [,2] [,3]  
## [1,] 3.78993113 -0.4387998 -0.07778984  
## [2,] -0.43879983 8.3220330 0.20260231  
## [3,] -0.07778984 0.2026023 4.99186266  
##   
##   
## $variance$cholsigma  
## , , 1  
##   
## [,1] [,2] [,3]  
## [1,] -3.131927 -0.1359752 0.19758806  
## [2,] 0.000000 -2.8388119 0.06474631  
## [3,] 0.000000 0.0000000 2.66995302  
##   
## , , 2  
##   
## [,1] [,2] [,3]  
## [1,] 1.966018 1.229238 4.697537e-02  
## [2,] 0.000000 1.487996 -3.068601e-05  
## [3,] 0.000000 0.000000 2.713864e+00  
##   
## , , 3  
##   
## [,1] [,2] [,3]  
## [1,] -1.946775 0.2253984 0.03995832  
## [2,] 0.000000 2.8759744 0.06731485  
## [3,] 0.000000 0.0000000 -2.23287588

# 4 Uzd. 4

## 4.1 1

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

|  | 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 |

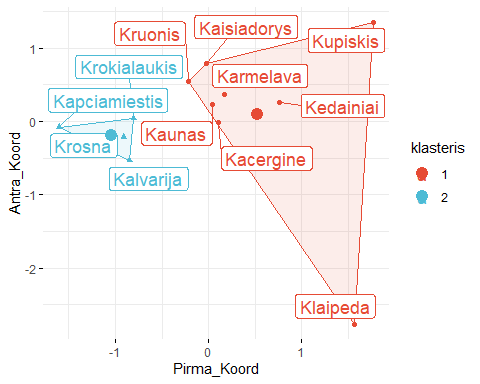
koordinates <- scale(koordinates)  
  
knitr::kable(koordinates)

|  | 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 |

## 4.2 2

### 4.2.1 2 klasteriai

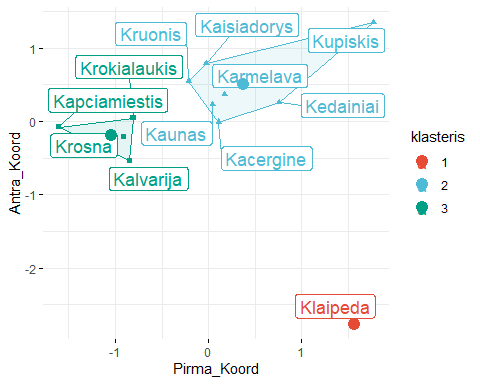
library(ggpubr)  
library(tibble)  
kmeans\_2 <- kmeans(koordinates, centers = 2)  
df <- koordinates %>%   
 as.data.frame() %>%   
 rownames\_to\_column("Miestas") %>%   
 bind\_cols(klasteris = factor(kmeans\_2$cluster))  
  
ggscatter(  
 df, x = "Pirma\_Koord", y = "Antra\_Koord",   
 color = "klasteris", 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")

### 4.2.2 3 klasteriai

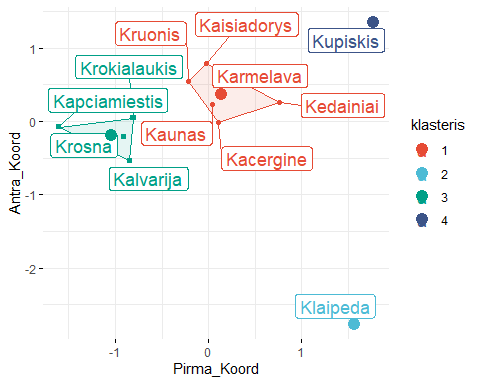
kmeans\_3 <- kmeans(koordinates, centers = 3)  
df <- koordinates %>%   
 as.data.frame() %>%   
 rownames\_to\_column("Miestas") %>%   
 bind\_cols(klasteris = factor(kmeans\_3$cluster))  
  
ggscatter(  
 df, x = "Pirma\_Koord", y = "Antra\_Koord",   
 color = "klasteris", palette = "npg", ellipse = 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")

### 4.2.3 4 klasteriai

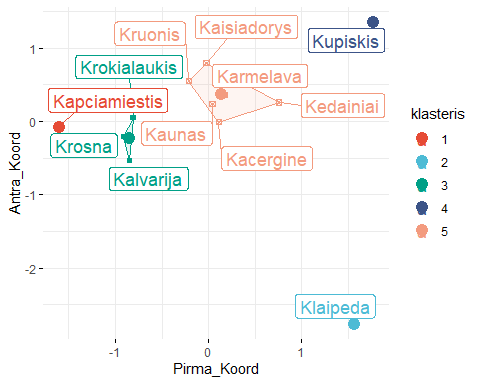
kmeans\_4 <- kmeans(koordinates, centers = 4)  
df <- koordinates %>%   
 as.data.frame() %>%   
 rownames\_to\_column("Miestas") %>%   
 bind\_cols(klasteris = factor(kmeans\_4$cluster))  
  
ggscatter(  
 df, x = "Pirma\_Koord", y = "Antra\_Koord",   
 color = "klasteris", palette = "npg", ellipse = 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")

### 4.2.4 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")

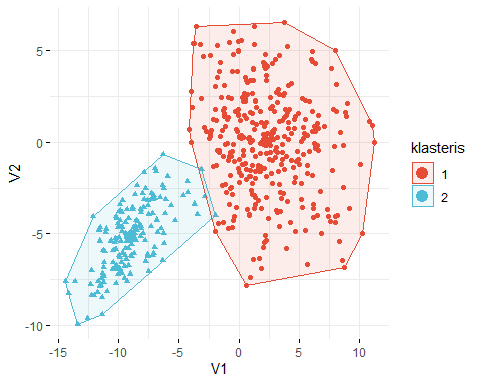
## 4.3 3

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)

## 4.4 4

### 4.4.1 2 klasteriai

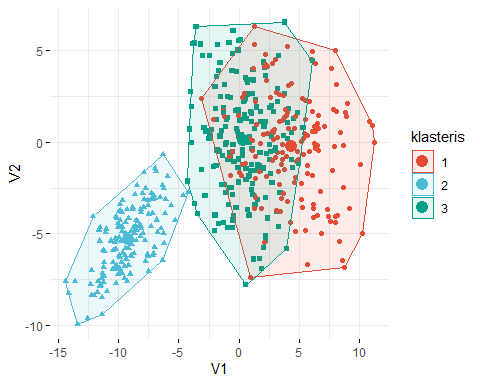
df <- gauso\_ad\_500 %>%   
 as.data.frame() %>%   
 bind\_cols(klasteris = factor(kmeans\_2$cluster))  
  
ggscatter(  
 df, x = "V1", y = "V2",   
 color = "klasteris", palette = "npg", ellipse = TRUE, ellipse.type = "convex",  
 shape = "klasteris", size = 1.5, legend = "right", ggtheme = theme\_minimal()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



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

### 4.4.2 3 klasteriai

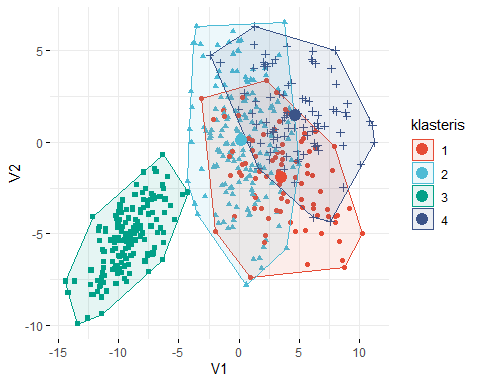
df <- gauso\_ad\_500 %>%   
 as.data.frame() %>%   
 bind\_cols(klasteris = factor(kmeans\_3$cluster))  
  
ggscatter(  
 df, x = "V1", y = "V2",   
 color = "klasteris", palette = "npg", ellipse = TRUE, ellipse.type = "convex",  
 shape = "klasteris", size = 1.5, legend = "right", ggtheme = theme\_minimal()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



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

### 4.4.3 4 klasteriai

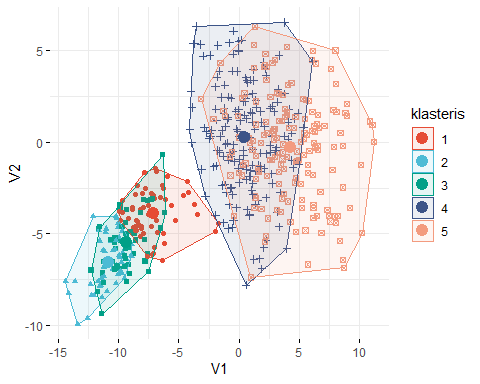
df <- gauso\_ad\_500 %>%   
 as.data.frame() %>%   
 bind\_cols(klasteris = factor(kmeans\_4$cluster))  
  
ggscatter(  
 df, x = "V1", y = "V2",   
 color = "klasteris", palette = "npg", ellipse = TRUE, ellipse.type = "convex",  
 shape = "klasteris", size = 1.5, legend = "right", ggtheme = theme\_minimal()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



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

### 4.4.4 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")

## 4.5 5

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()

## png   
## 2

## 4.6 6

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

## png   
## 2