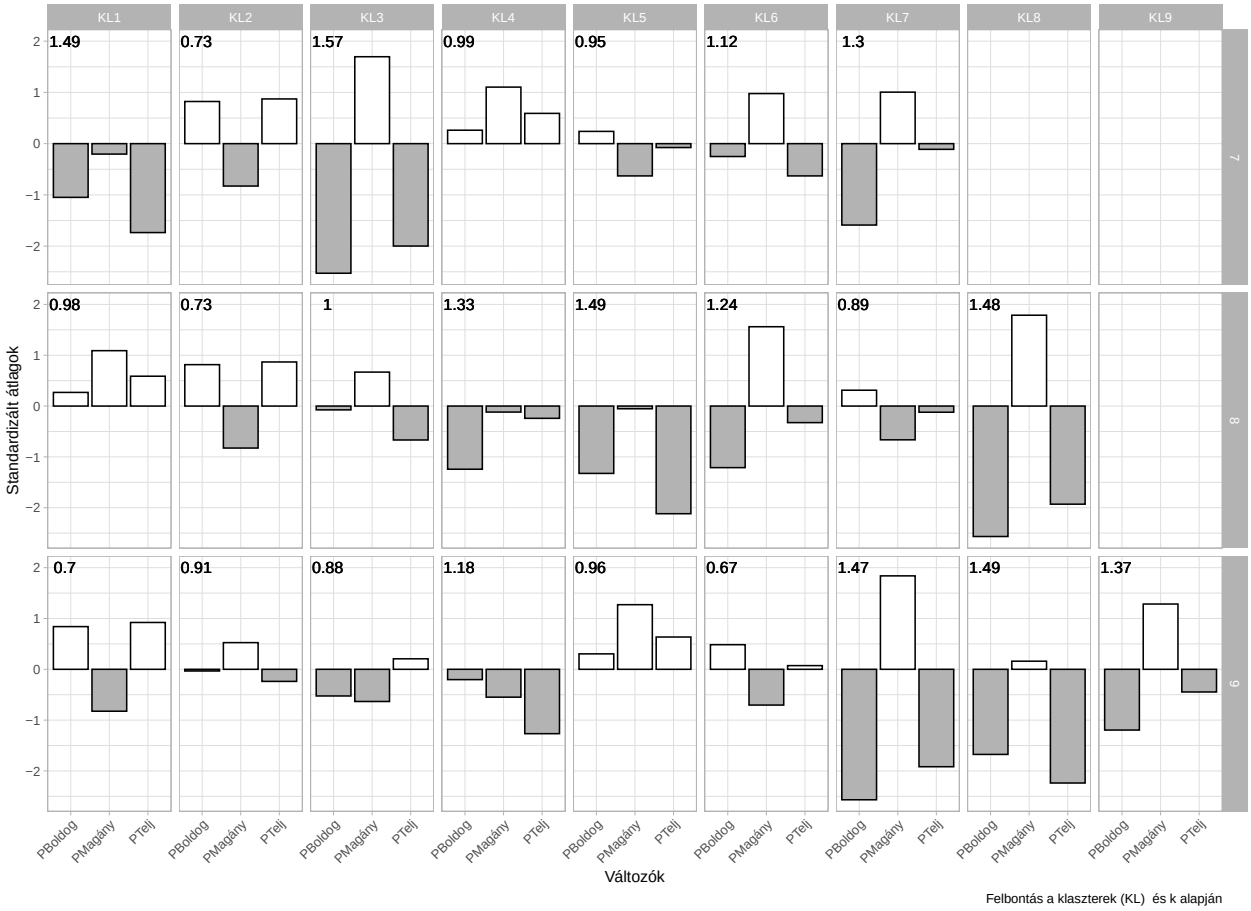
Házi feladatok megoldása 9.

k-középpontú klaszteranalízis R-ben

Smahajcsik-Szabó Tamás, M9IJYM

1. Végezz k-közép elemzést R-ben a PTELJ, Pboldog, Pmagány input változókkal, outlier kiszűréssel, standardizálással k = 7 és 9 között!

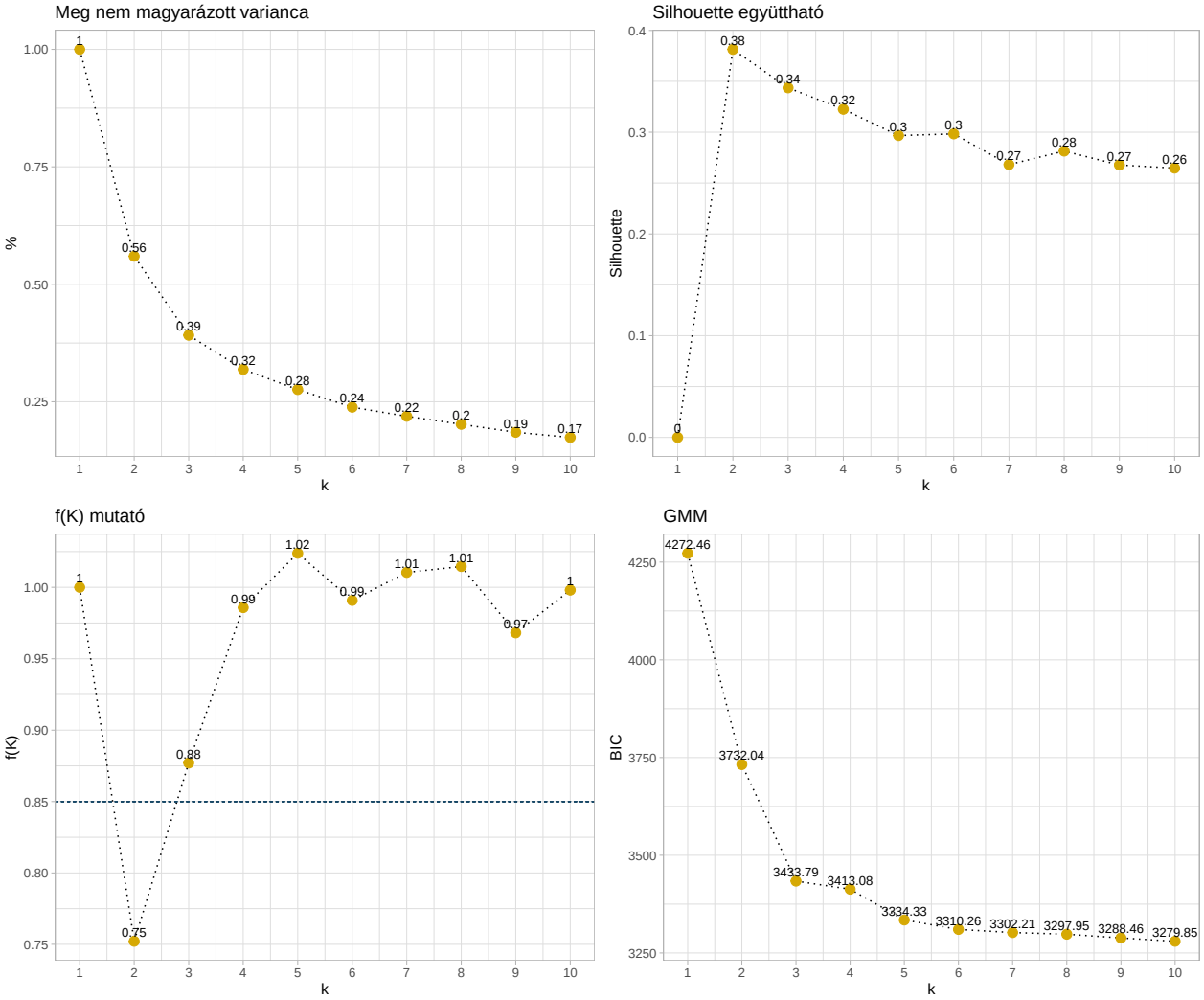
A k-közép elemzéseket 5 kezdeti centroid struktúrával, maximális 20 iterációval végeztem (MacQueen-féle algoritmussal). Az eredményekről az alábbi áttekintő ábra tájékoztat. A képződött klaszterek standard átlagait, és a homogenitási együtthatókat is feltüntettem.



1.ábra A klaszterstruktúra áttekintése

2. Hány klaszteres megoldás tűnik a legjobbnak az 1. feladat változói esetében a 6.4.1. alpontban leírt R-beli módszerek alapján (vö. 6.6-6.10. ábrák)?

A fenti ábra kedvezőtlenebb homogenitási indexeivel összhangban, a k=7 és k=9 közötti megoldás nem optimális a segítő ábrák alapján sem.



2. ábra: Segítő ábrák

A meg nem magyarázott varianca könyökábrája (bal felül), illetve a Silhouette együttható (jobb felül) egy k=2 megoldás fölényét erősíti a k = 7, 8 illetve 9 megoldásokkal szemben. Az *f(K)* mutató (bal alul) a k=2 megoldást emeli ki, k=2-nél ereszkedik f(k) értéke a 0.85 küszöb alá. A k=7 és k=9 megoldások közül a k=9 esetén kedvezőbb kissé a mutató, de mindegyikre nézve suboptimális a jelzés. GMM függvénnyel tesztelve az adatokat a BIC értéke egyaránt alacsony k=7 és k=9 között, de egyrészt nem optimális a BIC ezen struktúrák mellett, másrészt minden más segítő ábra a k=7, 8 vagy 9 megoldások nem kielégítő voltát erősíti.

3. Mentsd el az 1. feladat klaszterváltozóit k = 7 és 9 között, tedd át ROPstatba és számítsd ki a Validálás modullal a főbb QC mutatókat! Melyik klaszterszám megoldása tűnik a legjobbnak?

A *Validálás* modullal végzett számítások eredményét az alábbi táblázat összegzi.

EESS%	Pontbisz	XBmod	Sil.eh.	HCatlag	CLdelta	GDI24	HCmin-HCmax	k
78.39	0.355	0.494	0.661	0.440	0.893	0.384	0.21-1.11	7
79.75	0.347	0.518	0.659	0.413	0.894	0.412	0.21-1.02	8
81.34	0.313	0.353	0.639	0.382	0.875	0.283	0.18-1.01	9

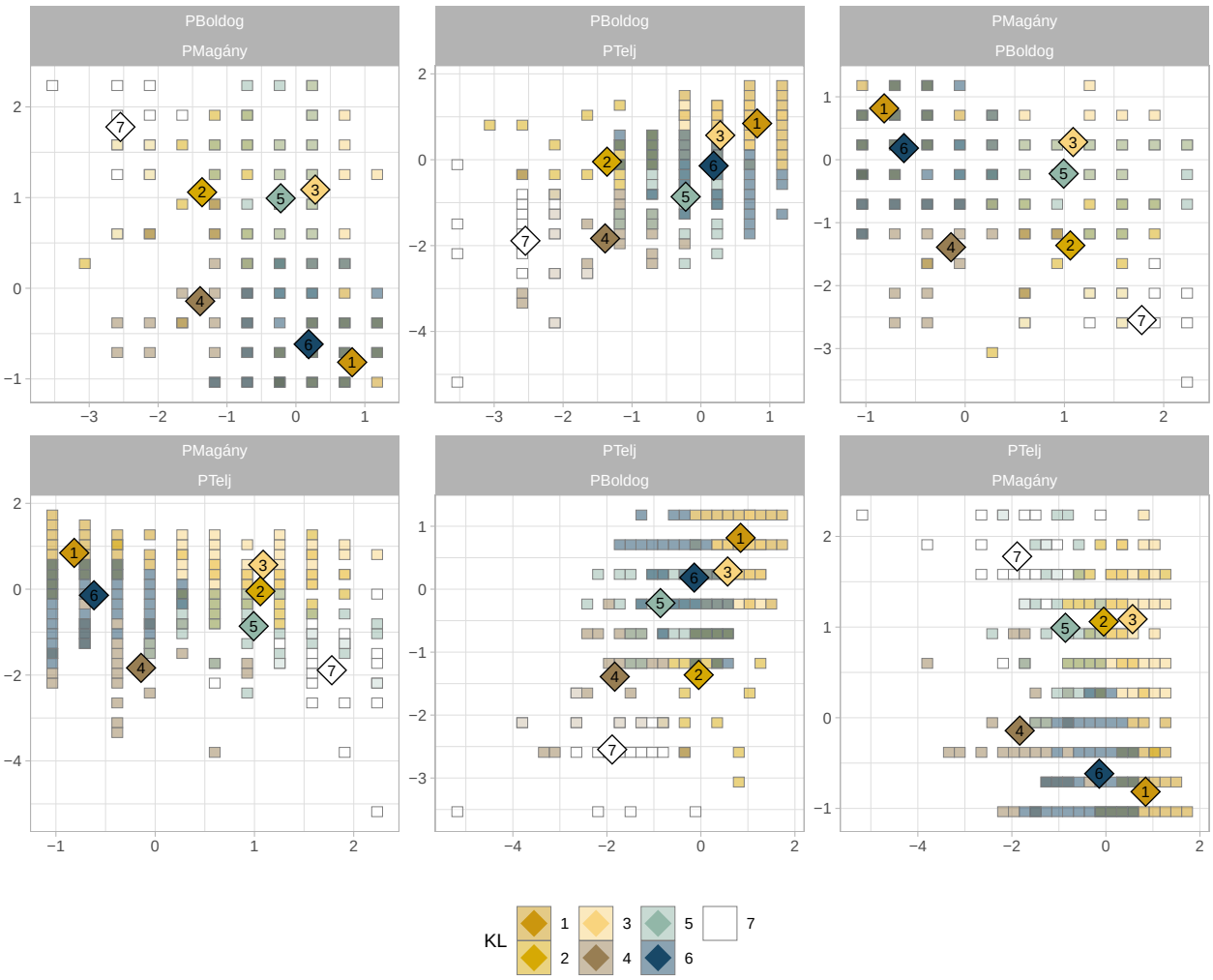
1.táblázat Klaszterstruktúra validitás mérése ROPStattal, k-középpontú klaszterelemzéshez

A magyarázott variancia (EESS%) növekszik **k** értékének emelkedésével, de nem láthatunk kiugró javulást. A Pontbiszeriális együttható értéke csökken, a klaszterszám emelkedésével. Optimális szintjént k=7-nél éri el. k=9 esetén nemcsak a PB-index, de a módosított Xie-Beni mutató is kedvezőtlen, 0.353 értéket vesz fel, legjobb k=8 esetén. A Silhouette index mindegyik megoldás esetén kedvező, k=7-nél tetőzik, majd folyamatosan csökken. A CLdelta k=8 esetén a legjobb a három struktúra közül, elfogadható szinten. A HCÁtlag természetszerűleg csökken **k** emelkedésével. A GDI24 index mindegyik megoldás esetén kedvezőtlen struktúrát jelez.

Mindezek alapján a k=7 vagy k=8 struktúra tűnik megfelelőnek. A hét klaszteres megoldást erősíti egy egyedi, *clustering_plot* nevű R függvényem, mely a **clusterCrit** csomag *IntCriterion*, illetve *BestCriterion* függvényei segítségével **k** különböző értékei mentén teszteli a **clusterCrit** R csomag adta tetszőleges illeszkedési mutatókat, és a *BestCriterion* függvény adta szavazatokat gyűjtve, numerikusan kifejezi, mely k lehet a legmegfelelőbb. A legjobb **k** értéknek azt veszi a megadott vektorból, mely a legtöbb szavazat alapján többséget szerez (*majority voting*). Az összes (42) illeszkedési mutatót bevonva a hét klaszteres megoldás tűnik megfelelőnek.

A függvény a változók egyes párjai mentén mutatja az adatok eloszlását, a klaszterbe tartozást adott k érték mentén színkóddal jelöli; továbbá rombusz (gyémánt) alakú ponttal tünteti fel a klaszter középpontját (k-középpont esetén a centroid, medoid elemzésnél a medoid).

A függvény jelenlegi forráskódját a **Függelékben** tüntettem fel. Az eredményt a függvény adta ábrával foglalom össze alább.



3. ábra A klaszterek változó-páronkénti megoszlása a centroidokkal mint rombuszokkal.

Az illeszkedési mutatók összesített szavazatait az alábbi táblázat foglalja össze. Az összes szavazatok száma 41, mert egy mutató nem volt alkalmazható az adatokra.

k	votes
7	18
8	13
9	10

2. táblázat A szavaztok eloszlása k értékei mentén a **clusterCrit** programból nyert adekvációs mutatókkal

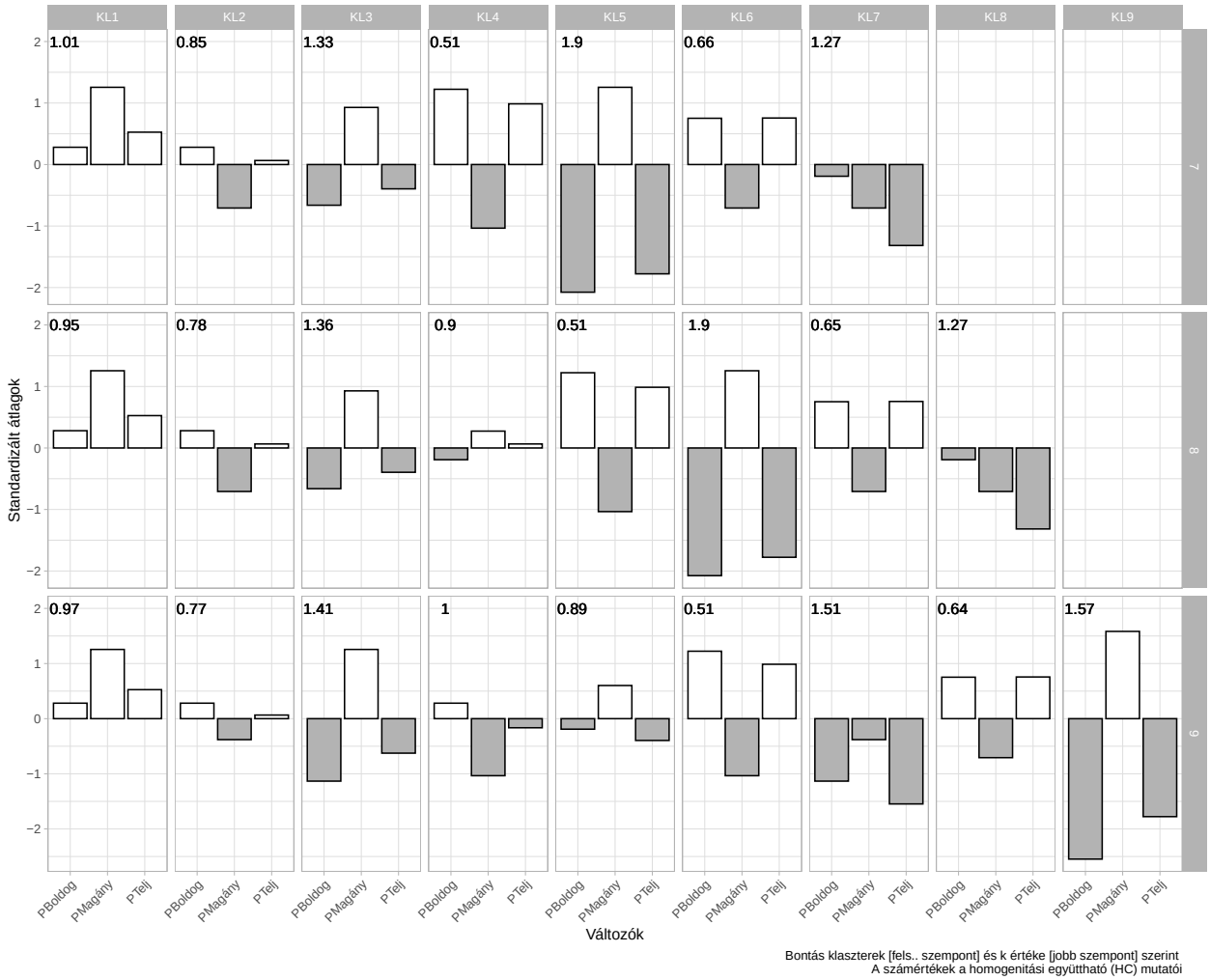
A homogenitási indexekről az alábbi táblázat tájékoztat.

k	1	2	3	4	5	6	7	8	9
7	0.7474445	1.2640905	0.9753942	1.521568	1.1560123	0.9946769	1.4771426	0.000000	0.000000
8	0.9084502	0.9856584	1.3510225	1.406028	0.7181216	1.1806257	0.8423153	1.813457	0.000000
9	0.8946394	1.3831566	0.7490864	1.813457	1.1734826	0.6764511	0.7414283	1.018271	1.351023

3.táblázat A homogenitási indexek klaszterenként k-középpontú elemzésben

4. Végezz k-medoid elemzést R-ben a PTELJ, Pboldog, Pmagány input változókkal, outlier kiszűréssel, standardizálással k = 7 és 9 között!

k-medoid elemzést végeztem a *pam* **R** függvénnyel, euklideszi távolság paraméterrel. Az eredmények közlését a standard átlagok klaszterenkénti megoszlásával kezdem az alábbi ábrán.



4. ábra A k-medoid módszerrel képzett klaszterek standardizált átlagainak és homogenitásának áttekintése

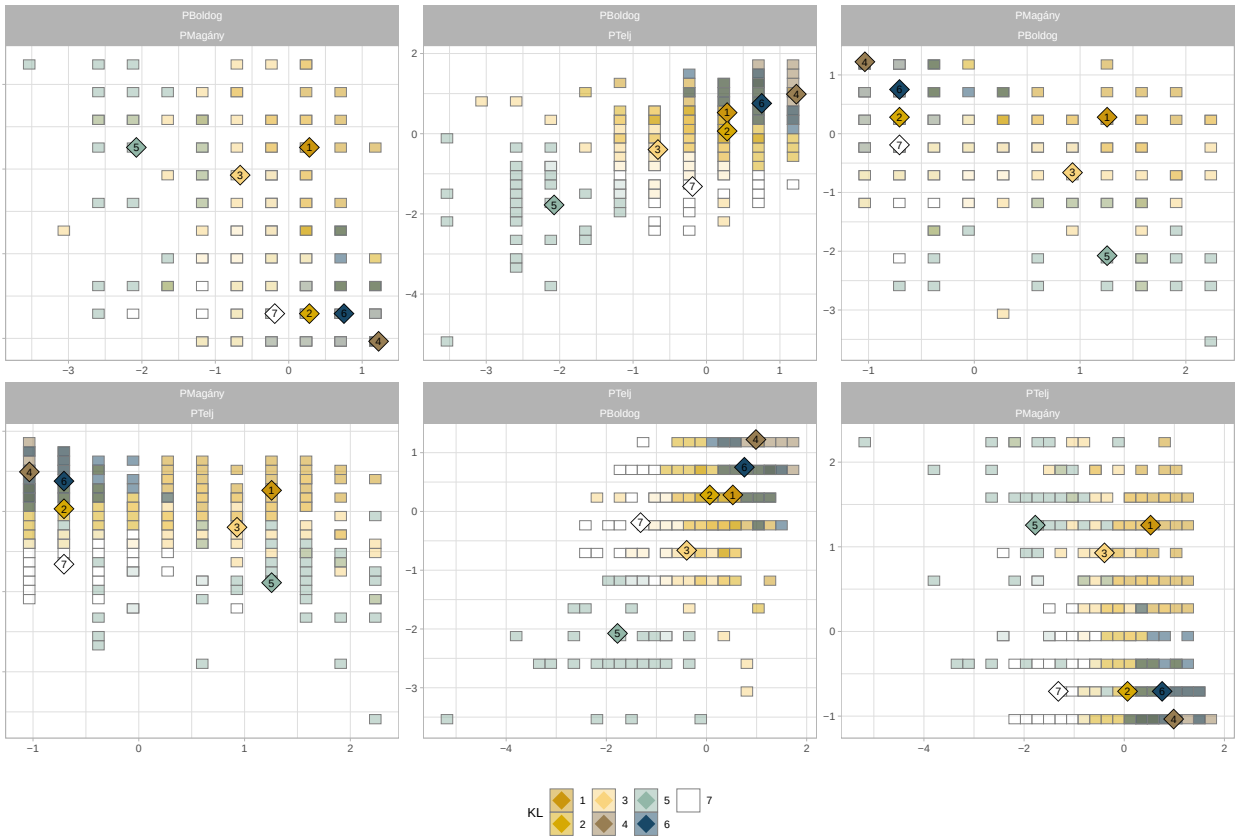
5. Mentsd el a 4. feladat klaszterváltozóit k = 7 és 9 között, tedd át ROPstatba és számítsd ki a Validálás modullal a főbb QC mutatókat! Melyik klaszterszám megoldása tűnik a legjobbnak?

EESS%	Pontbisz	XBmod	Sil.eh.	HCatlag	CLdelta	GDI24	HCmin-HCmax	k
76.32	0.298	-0.404	0.583	0.480	0.815	0.095	0.12-1.48	7
77.88	0.279	-0.375	0.571	0.450	0.812	0.093	0.12-1.48	8
80.08	0.270	-0.300	0.567	0.407	0.823	0.125	0.12-1.11	9

4.táblázat Klaszterstruktúra validitás mérése ROPStattal, k-medoid klaszterelemzéshez

A magyarázott variancia (EESS%) növekszik **k** értékének emelkedésével, de nem láthatunk kiugró javulást. A Pontbiszeriális mutatók értéke kedvezőtlen, k=7 esetén még az elfogadható 0.3 szint alatt, de azt nagyon megközelítő értéket látunk. A módosított Xien Beni mutatók nem mutatnak jó illeszkedést. A Silhouette index a k=7 esetén jelzi a legjobb illeszkedést 0.583 értékkel. A CLdelta k=8 esetén a legjobb a három struktúra közül, elfogadható szinten (0.823), mögötte a k=7 struktúra áll. A HCÁtlag természetszerűleg csökken **k** emelkedésével A GDI24 index mindegyik megoldás esetén kedvezőtlen struktúrát jelez. Felvethető a medoidok érvényességének alacsony szintje az adott adatokon.

Mindezek alapján a k=7 struktúra tűnik a legjobbnak a háromból.



5. ábra A k-medoid módszerrel képzett klaszterek eloszlásának és medoidjainak vizuális áttekintése

Függelék

A *clustering_plot* függvény áttekintése. A függvény alapesetben egy meglévő skálázott adattáblával dolgozik (*dataset* paraméter). A *method* paraméterrel jelenleg a *kmeans* vagy a *pam* függvényeket tudja alkalmazni, **k** adott értékeire, melyeket a *k_range* paraméterrel tudunk szabályozni. Az *autotune* opció *TRUE* esetén a függvény a **clusterCrit** szerinti, a *criteria_list* paraméterben megadott adekvációs mutatók mentén tesztelést végez *k* értékei mentén, és többségi szavazás révén kiválasztja a (fontos!) mennyiségi szempontból legjobbnak tűnő *k*-t. Tehát csupán azt veszi figyelembe, *k* mely értéke kapta a legtöbb szavazatot a *bestCriterion* függvénytől.

Második esetben, ha már rendelkezünk képzett klaszterstruktúrával, a *plot_data* paraméterrel tudjuk azt a függvénynek, mint inputot megadni.

autotune helyett előre rögzített *k*-t a *selected_k* paraméterrel tudunk megadni.

```
clustering_plot <- function(dataset, method = "pam", Nvar = 3,
                            k_range = 7:9, autotune = TRUE,
                            selected_k = 7, plot_data = NA,
                            plot_data_medoid = NA,
                            criteria_list = "all", ...) {
  library(clusterCrit)
  if (is.na(plot_data) || is.na(plot_data_medoid)) {
    medoids <- tibble()
    memberships <- tibble()
    criteria <- tibble()

    for (k in k_range) {
      set.seed(2324234)
      if (method == "pam") {
        medoid_fit <- pam(dataset, k = k, metric = "euclidean")
        medoid <- tibble(data.frame(medoid_fit$medoids))
        members <- medoid_fit$clustering
        medoid$HC <- hc(dataset, membership = members, max_k = k)
        membership <- tibble(data.frame(c = members), k = k)
      } else if (method == "kmeans") {
        medoid_fit <- kmeans(dataset, centers = k, iter.max = 20, nstart = 5,
                             algorithm = "MacQueen")
        members <- medoid_fit$cluster
        medoid <- tibble(data.frame(medoid_fit$centers))
        medoid$HC <- hc(dataset, membership = members, max_k = k)
        membership <- tibble(data.frame(c = members), k = k)
      }
      actual_criteria <- tibble(data.frame(intCriteria(dataset, members, criteria_list)))
      actual_criteria$k <- k
      criteria <- bind_rows(criteria, actual_criteria)
      medoid$Klaszter <- paste0("KL", 1:k)
      medoid$k <- k
      medoid$type <- "stand"
      medoids <- bind_rows(medoids, medoid)
      memberships <- bind_rows(memberships, membership)
    }
    HCs <- medoids %>%
      dplyr::select(k, HC) %>%
      unique()

    if (autotune) {
```

```
  criteria <- filter_out_nan(criteria)
  votes <- tibble(k = criteria$k, votes = 0)
  for (qc in colnames(criteria)[-ncol(criteria)]) {
    actual_qc <- unname(unlist(criteria[qc]))
    best_qc <- bestCriterion(actual_qc, qc)
    votes[best_qc, 2] <- votes[best_qc, 2][[1]] + 1
  }
  selected_k <- votes[votes$votes == max(votes$votes, na.rm = TRUE), ]$k
}

varnames <- expand.grid(names(medoids)[1:Nvar], names(medoids)[1:Nvar]) %>% filter(Var1 != Var2)
medoid_plot_data <- medoids %>%
  filter(k %in% selected_k) %>%
  mutate(c = as.numeric(str_sub(Klaszter, 3, 3)))
selected_members <- memberships %>% filter(k %in% selected_k)
plot_data <- tibble(data.frame(dataset))
plot_data_prep <- bind_cols(plot_data, selected_members)

plot_data <- tibble()
for (i in 1:nrow(varnames)) {
  actual_pair <- as.character(unname(unlist(varnames[i, ])))
  actual_df <- tibble(
    var1 = rep(actual_pair[1], nrow(plot_data_prep)),
    var2 = rep(actual_pair[2], nrow(plot_data_prep)),
    x = unname(unlist(plot_data_prep[, actual_pair[1]])),
    y = unname(unlist(plot_data_prep[, actual_pair[2]])),
    KL = plot_data_prep$c
  )
  colnames(actual_df) <- c("var1", "var2", "x", "y", "KL")
  plot_data <- bind_rows(plot_data, actual_df)
}
plot_data_medoid <- tibble()
for (i in 1:nrow(varnames)) {
  actual_pair <- as.character(unname(unlist(varnames[i, ])))
  actual_df <- tibble(
    var1 = rep(actual_pair[1], nrow(medoid_plot_data)),
    var2 = rep(actual_pair[2], nrow(medoid_plot_data)),
    x = unname(unlist(medoid_plot_data[, actual_pair[1]])),
    y = unname(unlist(medoid_plot_data[, actual_pair[2]])),
    KL = medoid_plot_data$c
  )
  colnames(actual_df) <- c("var1", "var2", "x", "y", "KL")
  plot_data_medoid <- bind_rows(plot_data_medoid, actual_df)
}

plot_data <- plot_data %>%
  mutate(KL = as.factor(KL))
plot_data_medoid <- plot_data_medoid %>%
  mutate(KL = as.factor(KL))

# saveRDS(plot_data, "plot_data.RDS")
# saveRDS(plot_data_medoid, "plot_data.RDS")
```

```
    # plot_data <- readRDS("plot_data.RDS")
    # plot_data_medoid <- readRDS("plot_data_medoid.RDS")
  }

  CLplot <- ggplot() +
    #  geom_point(data = plot_data, aes(x, y),color="black", size = 5) +
    #  geom_point(data = plot_data, aes(x, y, color = KL), size = 4) +
    geom_bin2d(data = plot_data, aes(x, y, fill = KL), alpha = 1 / 2, color = "grey50") +
    geom_point(data = plot_data_medoid, aes(x, y), color = "black",
               size = 8, shape = 18) +
    geom_point(data = plot_data_medoid, aes(x, y, color = KL),
               size = 7, shape = 18) +
    geom_text(data = plot_data_medoid, aes(x, y, label = KL),
              size = 3, shape = 18) +
    facet_wrap(~var1 ~ var2, scales = "free") +
    scale_color_manual(values = c("#CB960E", "#D6A904",
                                  "#F9D47E", "#987E53", "#92B7A8", "#184867",
                                  "white", "coral", "coral4")) +
    scale_fill_manual(values = c("#CB960E", "#D6A904", "#F9D47E",
                                 "#987E53", "#92B7A8", "#184867", "white",
                                 "coral", "coral4")) +
    labs(x = "", y = "") +
    theme_light() +
    theme(legend.position = "bottom")

  if (exists("votes")) {
    list(
      "plot" = CLplot,
      "votes" = votes,
      "medoids" = medoids,
      "homogenity" = HCs
    )
  } else {
    CLplot
  }
}
```