Praktikum 8
Klaster- ja peakomponentanalüüs

¤) Salvestage arvutisse Eesti puude andmestik *R*-i failina:

http://www.eau.ee/~ktanel/DK_0007/puud.rda

¤) Avage *R* ja käivitage lisamoodul *Rcmdr*.

¤) Võtke *R*-is kasutusele salvestatud puude andmestik

- *R Commanderis* käsud *Data -> Load data set …*
- või skripti aknas käsk `load`, näiteks
  `load("C:/Documents and Settings/Tanel/DK_0007/puud.rda")`

¤) **`attach(puud)`** # edasiste käskude lihtsama esituse huvides

¤) Puude andmestikus vastab üks rida ühele puule, veergudes on vastavalt:

veerus nimega 'A' puu vanused aastates;

veerus 'D' puu diameetersentimeetrites;

veerus 'H' puu kõrgus meetrites;

veerus 'ARENGUKL' puu arenguklass väärtustega A – lage, N – noorendikud, L – latimets (noorendikust järgmine), K – keskealised, V – valmiv, Y – küps, S – selguseta, – puuduv väärtus;

veerus 'PE' puu liik (HB – haab, KS – kask, KU – kuusk, LV – hall lepp, …);

veerus 'KKT' kasvukohatüüp (AN – angervaksa, JM – mustika, …)[1];

…

1. Klasteranalüüs

Kaks peamist klasterdamisega seotud funktsiooni *R*-is on

`hclust` (hierarhiline klasterdamine) ja

`kmeans` (k-keskmiste meetod).

R Commanderis on mõlemad meetodid leitavad menüüdest

Statistics -> Dimensional analysis -> Cluster analysis ->

Nagu *R Commanderi* puhul tavaline, on menüüdest valitud analüüside tulemusel kirjutatav programm pisut teistsugune – *R Commander* kasutab lihtsalt vähe erinevat süntaksit, samas töötavad *R Commanderi* skripti aknas ka tavalisele *R*-ile omased käsud `hclust` ja `kmeans`.

Kuna hierarhiline klasterdamine on märksa töömahukam võrreldes k-keskmiste meetodiga, kasutatakse esimest enamasti väikeste ja teist suurte andmestike korral.

[1] Kasvukohatüüp (ka metsakasvukohatüüp) on mullastikult ja taimestikult ühtlane metsaala. Nimetus tuleb enamesineva taime järgi (näiteks kasvab naadi kasvukohas väga palju naate, jänesekapsa kasvukohatüübis jänesekapsaid jne) – http://et.wikipedia.org/wiki/Kasvukohatüüp.
Täpsemalt vt näiteks http://www.hot.ee/sinumets3/sinumets03-08.pdf.

1.1. Püüdke jagada üksikud puud klastritesse kasutades andmeid puude vanuse ja diameetri kohta.

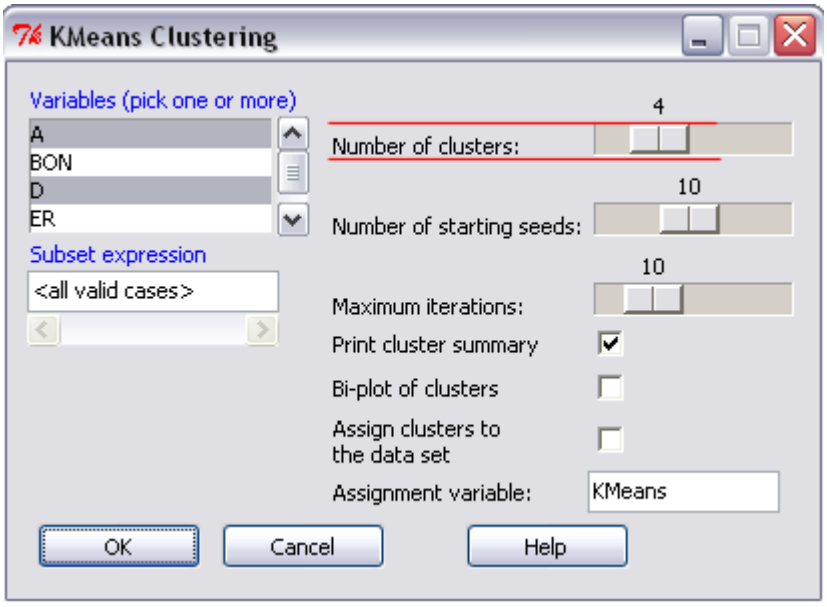
Kuna puid on palju (*n*=29554), siis on mõistlik kasutada k-keskmiste meetodit, ja jagada võiks puud näiteks 4 klastrisse.

1. võimalus – käsuna skriptiaknas:

```
tree_kmean1 = kmeans(cbind(A, D), 4)
```

2. võimalus – *R Commander*'i menüüdest:

Statistics -> Dimensional analysis -> Cluster analysis -> kmeans cluster analysis ...



Eelkõige on oluline määrata soovitav klastrite arv,

ülejäänud parameetrid (nn algseemnete arv ja maksimaalne iteratsioonide arv) on seotud iteratiivse klasterdamisalgoritmi tööga ja võivad enamusel juhtudel jääda muutmata (sama käsku skriptiaknas käivitades võib vastavad argumendid – `iter.max` ja `num.seeds` – üldse ära jätta).

Valikuga *'Print cluster summary'* tellitakse olulisem klasterdamise tulemuste kohta käiv info,

valiku *'Bi-plot clusters'* tulemusena teostatakse ette antud tunnustega peakomponentide analüüs ja joonistatakse kahe esimese peakomponendi põhjal hajuvusdiagramm, kus iga objekti märgib klastri number, millesse vastav objekt on määratud – mõningatel juhtudel võib taoline graafik anda täpsema ettekujutuse sellest, milliste omadustega objektid mingisse klastrisse kuuluvad (et üksnes kahe tunnuse alusel teostatud klasteranalüüsi tulemuste illustreerimiseks on ka lihtsamaid võimalusi, on antud näites see valik tühjaks jäetud),

kolmas lisavalik võimaldab lisada klastrinumbrid algsesse andmebaasi eraldi veeruna (mille nime saab ka eraldi määrata).

Tulemuseks on programm kujul

```
.cluster <- KMeans(model.matrix(~-1 + A + D, puud), centers=4, iter.max=10, num.seeds=10)
.cluster$size # Cluster Sizes
.cluster$centers # Cluster Centroids
.cluster$withinss # Within Cluster Sum of Squares
.cluster$tot.withinss # Total Within Sum of Squares
.cluster$betweenss # Between Cluster Sum of Squares
remove(.cluster)
```

Need käsud on valiku *'Print cluster summary'* tulemus.

Kuna *R Commander*

1) ei võimalda omistada teostatud klasteranalüüsi tulemustele oma nime ja

2) kustutab analüüsi tulemustest moodustatud muutujad peale tellimisaknas valitud analüüside teostamist (automaatselt genereeritav käsk `remove(.cluster)`),

siis tuleb täiendavate analüüside teostamiseks *R Commanderi* poolt genereeritud programm uuesti käivitada, muutes igaks juhuks ka nime, mille all analüüsi tulemusi hoitakse
(et mõni *R Commanderi* abil edaspidi teostatav klasteranalüüs tulemusi üle ei kirjutaks):

```
tree_kmean1_RC <- KMeans(model.matrix(~-1 + A + D, puud), centers = 4)
```

Järgnevate käskude puhul on ükskõik, kas kasutada käsuga `kmeans` teostatud klasteranalüüsi tulemusi (muutuja `tree_kmean1`) või *R Commanderi* käsuga `KMeans` teostatud klasteranalüüsi tulemusi (muutuja `tree_kmean1_RC`).

¤) Info, mis klastrisse mingi puu kuulub, on kirjas muutujas `tree_kmean1$cluster`.
Näiteks 100 esimese puu klastrid on välja trükitavad käsuga

```
tree_kmean1$cluster[1:100]
```

¤) Klastrite suurused:

```
table(tree_kmean1$cluster)
```

Sama, mis eelnev käsk, üksnes ilma klastrite numbriteta (sama käsk sisaldub ka *R Commanderi* väljastatavas klasteranalüüsi kokkuvõttes)

```
tree_kmean1$size
```

¤) Klastrite keskpunktid (ka see käsk sisaldub *R Commanderi* klasteranalüüsi kokkuvõttes)

```
tree_kmean1$centers
```

```
> tree_kmean1$centers
          A        D
1  75.372441 21.59089
2  46.909646 14.08725
3   9.851673  2.01572
4 114.754876 26.32926
```

Klastri number

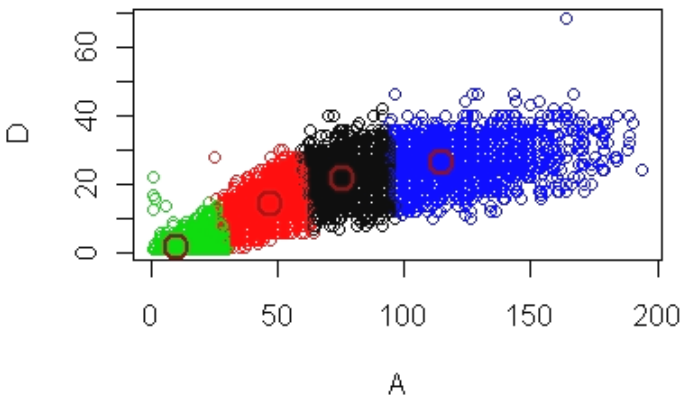
Klastrisse kuuluvate puude keskmine vanus

Klastrisse kuuluvate puude keskmine diameeter

¤) Klasterdades vaatluseid (=puid) vaid kahe tunnuse alusel, on toimuvat võimalik üsna hästi illustreerida graafiliselt (siin esimese käsk joonistab puude vanuse ja diameetri hajuvusdiagrammi, kus erinevatesse klastritesse kuuluvad puud on tähistatud erineva värviga, teine käsk lisab klastrite keskpunktid ringidena):

```
plot(A, D, col=tree_kmean1$cluster)
points(tree_kmean1$centers, cex=2, col="red4", bg=1:4, pch=21, lwd=2)
```

Tulemus:



Veendumaks, et nii käsuga `kmeans` kui ka *R Commanderi* käsuga `KMeans` teostatud klasteranalüüsi tulemused on samad, võite mõlemate kohta tellida joonised ühele lehele:

```
par(mfrow=c(2,1))
plot(A, D, col=tree_kmean1$cluster)
points(tree_kmean1$centers, cex=2, col="red4", bg=1:4, pch=21, lwd=2)

plot(A, D, col=tree_kmean1_RC$cluster)
points(tree_kmean1_RC$centers, cex=2, col="red4", bg=1:4, pch=21, lwd=2)
par(mfrow=c(1,1))
```

¤) Miks tundub joonist vaadates, et klastrid on tehtud puu (tunnus 'A') vanuse järgi?

Vastus – puud paigutati klastritesse eelkõige vanuse alusel põhjusel, et puude vanust märkivad arvud (eelkõige nende varieeruvus) on suuremad, kui diameetrit märkivad arvud. Soovides, et klasterdamisalgoritm peaks puu diameetrit sama oluliseks kui vanust, tuleb klasterdamiseks kasutatavad tunnused eelnevalt standardiseerida (lahutada kõigist väärtustest keskmine ja jagada standardhälbega).

Tunnuseid saab standardiseerida käsu `scale` abil:

```
tree_kmean2 = kmeans(scale(cbind(A, D)), 4)
```

Sama käsku saab kasutada ka muutmaks *R Commanderi* käsku skriptiaknas:

```
tree_kmean2_RC <- KMeans(model.matrix(~-1 + scale(A) + scale(D), puud), centers = 4)
```

Veel ühe alternatiivina võib standardiseeritud tunnused lisada uute tunnustena andmestikku, mida saab *R Commanderis* tellida menüüdest

Data -> Manage variables in active dataset -> Standardize variables ...

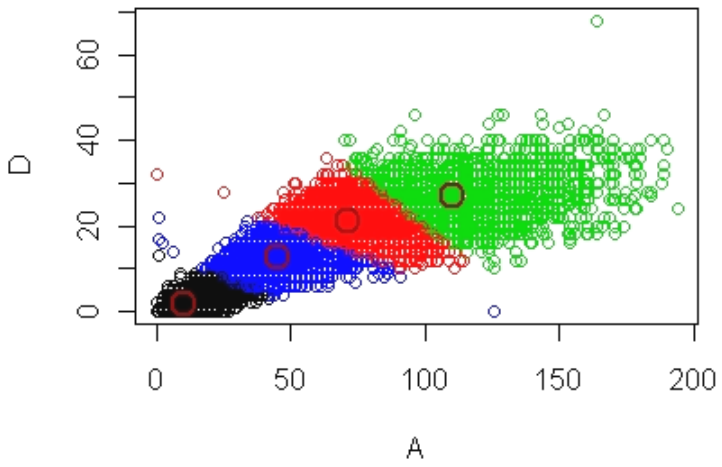
ja rakendada seejärel klasteranalüüsi juba standardiseeritud tunnustele.

Tulemuste illustreerimiseks:

```
plot(A, D, col=tree_kmean2$cluster)
```

Klastrite keskpunktide joonisele kandmiseks tuleb aga teha täiendavaid arvutusi, sest klasteranalüüsiga on leitud klastrite keskpunktid standardiseeritud tunnuste jaoks.

```
meanA = tree_kmean2$centers[,1]*sd(A) + mean(A)
meanD = tree_kmean2$centers[,2]*sd(D) + mean(D)
points(meanA, meanD, cex=2, col="red4", bg=1:4, pch=21, lwd=2)
```



Nagu näha, on klasterdamisprogramm peale standardiseerimist käsitlenud puude vanust ja diameetrit samaväärsetena (klastrite piirid sõltuvad mõlemast tunnusest).

¤) Vaatame veelkord, kuidas puud on klasterdunud – sedakorda sõltuvalt arenguklassist.

```
table(tree_kmean2$cluster, puud$ARENGUKL)
```

```
> table(tree_kmean2$cluster, puud$ARENGUKL)

      -    A    K    L    N    S    V    Y
  1   3    0  989    0    1    0  753 2297
  2   2    0 4643 2977    6    0  257  210
  3  23 1193   46 1044 3774 1682   44    0
  4   1    0 6199    1    0    0 1704 1705
```

Nagu näha, kuuluvad arenguklassid A, N, S ja osaliselt ka L ühte klastrisse (pisitillukesed ja noorukesed puud); klastrisse 2 kuuluvatest puudest on enamus arenguklassidest K ja L (küpsust saavutavad puud); klastrisse 1 kuuluvate puude hulgas on märgatavalt palju arenguklassi Y kuuluvaid puid, märkimisväärselt ka arenguklassi V puid ja natuke ka arenguklassi K puid (suured ja vanad puud); 4. klastrisse kuuluvad arenguklasside K ja V enamus, veidi ka arenguklassi Y puid – samuti suured ja vanad, kenasti raieküpsed puud.

1.2. Püüame klasterdada puuliike.

Puuliikide klasterdamiseks peame esmalt moodustama andmetabeli, kus objektideks on puuliigid. Lisaks peame otsustama, millised tunnused sellesse andmetabelisse kaasata. Võtame nendeks tunnusteks puuliigi keskmise diameetri 20 ja 50 aasta vanuses ning diameetri juurdekasvu vanuses 60-80.

Seega tahame jõuda andmetabelini kujul

	D50	D20	D60_80
HB	19.49209	6.098114	5.535394
KS	15.44849	4.799597	4.449109
KU	15.59229	4.680480	5.265263
LH	20.27949	6.810392	4.777650
LM	16.56126	6.799664	4.510453
LV	16.62557	6.259825	0.000000
MA	13.73487	4.201077	4.890281
RE	22.64630	6.816267	0.000000
SA	16.02317	4.926619	4.814877
TA	17.33763	5.815752	5.491468

Kuidas sellist andmetabelit koostada? Kuna meil algses andmetabelis selliseid tunnuseid pole, tuleb nende väärtused prognoosida olemasolevate tunnuste baasil. Näiteks võime iga puuliigi tarvis püüda prognoosida puu diameetrit sõltuvalt tema vanusest 4. järku polünoomi abil (see peaks olema piisavalt täpne võimaldamaks modelleerida ka mittelineaarset juurdekasvu), leitud mudeli alusel saab juba omakorda prognoosida kõigi vajalike tunnuste väärtused.

¤) Kirjeldatu teostamiseks on üks variant teha arvutused iga puuliigi jaoks eraldi:

```
model_KU = lm(D~A+I(A^2)+I(A^3)+I(A^4), data=puud[PE==“KU“,])
predict(model_KU, data.frame(A=c(20,50,60,80)))
```

Järgnevalt tuleks tulemused välja kirjutada ja jätkata uue puuliigiga.

¤) Alternatiivne variant on lasta kõik teha arvutil.

Et liiga väikese esinemissagedusega puude puhul ei pruugi prognoosid eriti usaldusväärsed tulla, jätame analüüsist välja liigid, mille kohta on vähem kui 10 mõõtmist.

```
species = names(table(PE)[table(PE)>10])
```

Järgnevalt programmeerime tsükli üle kõigi puuliikide (mille arv on tähistatud *n*-ga), moodustame esmalt tühjad vektorid kõigi tunnuste tarvis ja täidame need rida realt konkreetsele puuliigile prognoositud väärtustega (konkreetset puuliiki ja vastavat rida tunnuste vektoreis märgib muutuja *i*). Vältimaks ebareaalseid prognoose puuliikidele, mille puhul 80-aastaseid puid polegi, leiame täiendavalt ka maksimaalse vanuse iga puuliigi tarvis ja võrdsustame diameetri juurdekasvu vanuses 60-80 nendel liikidel, kelle maksimaalne vanus <80, nulliga.

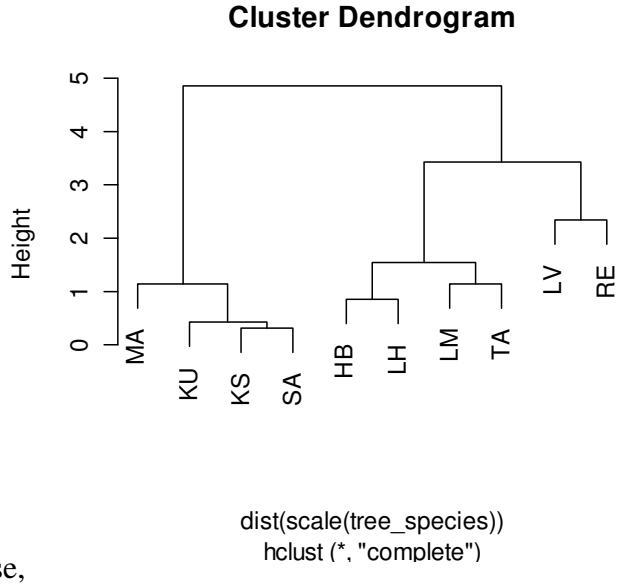
```
n = length(species)
D50=rep(NA,n); D20=rep(NA,n); D60_80=rep(NA,n); AMAX=rep(NA,n)
for (i in 1:n) {
 m1 = lm(D~A+I(A^2)+I(A^3)+I(A^4), data=puud[PE==species[i],])
 D50[i] = predict(m1, data.frame(A=50))
 D20[i] = predict(m1, data.frame(A=20))
 D60_80[i] = predict(m1, data.frame(A=80)) - predict(m1, data.frame(A=60))
 AMAX[i] = max(A[PE==species[i]])
}
D60_80[AMAX<80]=0
```

Koondame moodustatud vektorid üheks andmestikuks:

```
tree_species = data.frame(D50, D20, D60_80)
rownames(tree_species)=species
```

¤) Teostame uue andmestikuga hierarhilise klasteranlüüsi ja tellime tulemusena dendrogrammi (joonis, mis esitab klasterduse tulemused puu kujul).

```
tree_hclust1 = hclust(dist(scale(tree_species)))
plot(tree_hclust1)
```



Kui soovime puuliigid jagada kolme klastrisse, saame seda teha käsuga `cutree`:

```
tree_hclust1_3 = cutree(tree_hclust1, 3)
tree_hclust1_3
```

```
HB KS KU LH LM LV MA RE SA TA
 1  2  2  1  1  3  2  3  2  1
```

Esimesse klastrisse kuuluvad puuliigid haab, tamm, …..; teise klastrisse kask, kuusk, mänd ja saar; kolmandasse klastrisse hall lepp ja remmelgas.

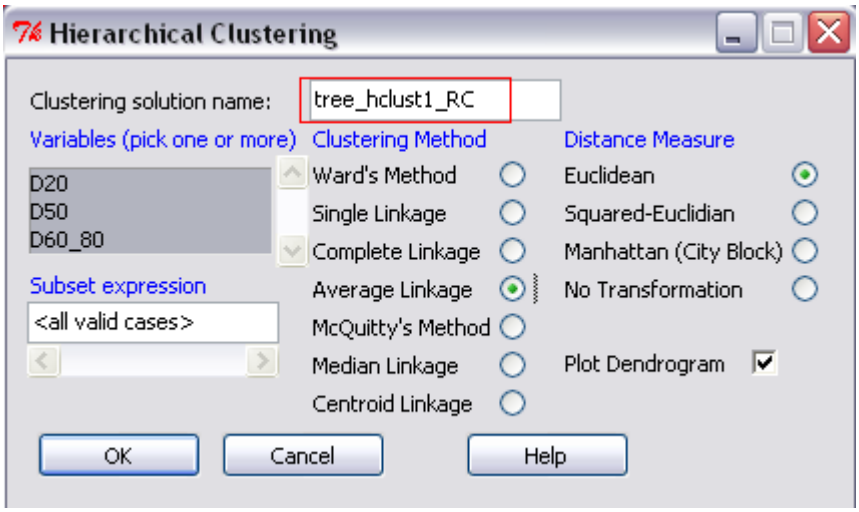
¤) Järgnevalt võite püüda muuta klasterdamismeetodit ja jälgida, kas ja kuidas tulemused muutuvad.

```
tree_hclust1 = hclust(dist(scale(tree_species)), method="ave")
tree_hclust2 = hclust(dist(scale(tree_species)), method="complete")
tree_hclust3 = hclust(dist(scale(tree_species)), method="single")
tree_hclust4 = hclust(dist(scale(tree_species)), method="centroid")

par(mfrow=c(2,2))
plot(tree_hclust1); plot(tree_hclust2); plot(tree_hclust3); plot(tree_hclust4)
par(mfrow=c(1,1))
```

¤) Analoogse ülesande võib lahendada *R Commanderi* abil, kusjuures erinevalt k-keskmiste meetodist saab hierarhilise klasterdamise korral omistada tulemusi sisaldavale muutujale ka nime, mille alusel saab konkreetse analüüsi kohta hiljem lisaanalüüse tellida.

Statistics
-> Dimensional analysis
-> Cluster analysis
-> Hierarchical cluster analysis ...



Hoolimata sellest, et hierarhilisel klasterdamisel kasutab *R Commander* funktsiooni `hclust` nii nagu ka *R*-i baasversioon, võivad tulemused siiski erinevad tulla. Võrdle näiteks:

```
tree_hclust1 = hclust(dist(scale(tree_species)), method="ave")
tree_hclust1_RC = hclust(dist(model.matrix(~-1 + D20+D50+D60_80,
tree_species)), method= "average")

par(mfrow=c(2,1))
plot(tree_hclust1); plot(tree_hclust1_RC)
par(mfrow=c(1,1))
```

Põhjus on selles, et *R Commander* ei kasuta vaikimis tunnuste standardiseerimist ega võimalda seda teha ka analüüsi tellimise käigus – lahenduseks on kas tunnuste eelnev standardiseerimine või siis *R Commanderi* poolt väljastatud programmi käsitsi täiendamine. Viimane tähendab funktsiooni `scale` lisamist kõigi tunnuste ette:

```
tree_hclust1_RCsc = hclust(dist(model.matrix(~-1+scale(D20)
  +scale(D50)+scale(D60_80), tree_species)), method= "average")
```

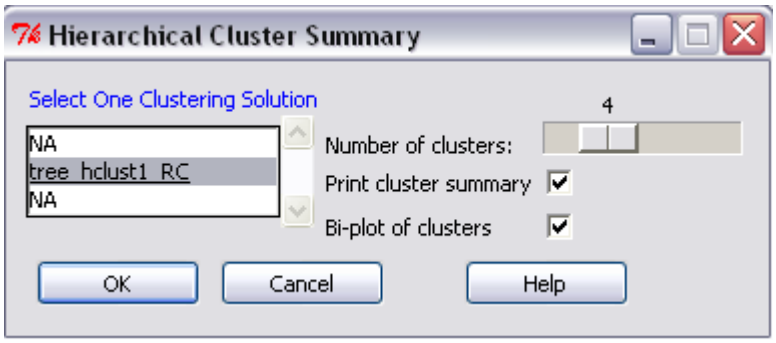
Nüüd peaksid tulemused tulema juba analoogsed.

¤) Lisaks võimaldab *R Commander* valida menüüdest mõningaid lisaanalüüse hierarhilise klasterdamise tulemuste alusel. Halb on see, et aktsepteerib *R Commander* vaid enese poolt teostatud analüüse, skripti aknas jooksutatud klasteranalüüside kohta menüüdest midagi lisaks tellida ei saa. Küll aga on võimalik menüüdest tellitud lisaanalüüside programmides asendada muutujate nimesid ja selliselt muudetud programmid skripti aknas uuesti käivitada …

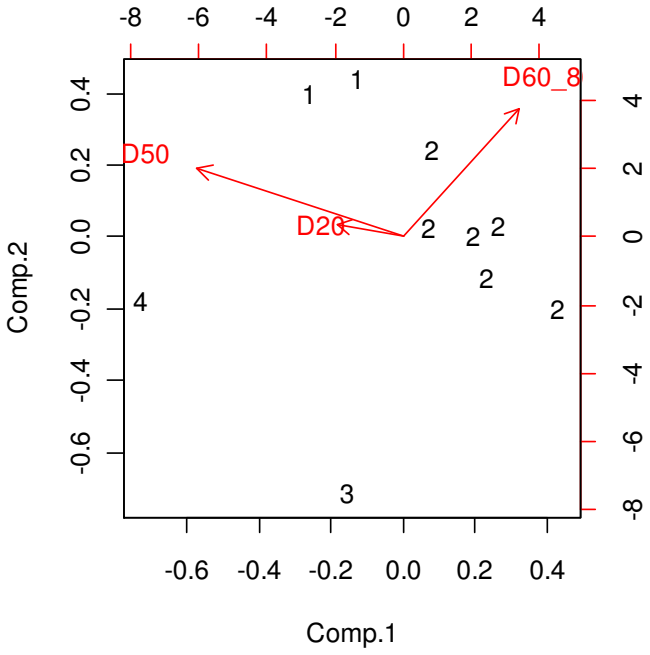
Näiteks *R Commanderiga* teostatud klasteranalüüsi tulemuste `tree_hclust1_RC` kohta lisaanalüüside tellimiseks tuleb menüüst

Statistics -> Dimensional analysis -> Cluster analysis

valida käsk
Summarize Hierarchical clustering ...



Jagades puuliigid 4 klastrisse saame lisavaliku '*Print cluster summary*' tulemusena puliikide arvud ja keskmised klasterdamise aluseks olnud tunnuste väärtused klastrite kaupa;
lisavalik '*Bi-plot clusters*' väljastab **peakomponentanalüüsi** tulemusena leitud kahe esimese peakomponendi korrelatsioone uuritavate tunnustega illustreeriva joonise, millel on erinevate numbritega ka erinevatesse klastritesse kuuluvate puuliikide asetsus peakomponentide mõistes.



Esimene peakomponent on negatiivselt seotud diameetriga 20 ja 50 aasta vanuses ning positiivselt juurdekasvuga vanuses 60-80 omades seega väikeseid väärtusi juhul, kui puuliik kasvab kiiresti, saavutades maksimumi lähedase läbimõõdu juba 50.-ks eluaastaks (juurdekasv vanuses 60-80 on väike), ning suuri väärtusi juhul, kui algne kasv on suhteliselt aeglane, aga kasv vanuses 60-80 märgatav.

Teine peakomponent väljendab üldist kasvukiirust olles positiivselt korreleeritud kõigi tunnustega.

Klastrite paigutus näitab seda, et 3. ja 4. klastrisse kuuluvad liigid, mis on suhteliselt kiirekasvulised ja ei ela enamasti kaua – hall lepp ja remmelgas (vt `tree_hclust1_RC` kohta joonistatud dendrogrammi või siis uuri liikide klastritesse kuuluvust käsuga `cutree(tree_hclust1_RC, k=4)`); klastrisse 2 kuuluvad liigid kasvavad vanuses 60-80 kõige rohkem, ja klastrisse 1 kuuluvad liigid võib liigitada keskmise kasvukiirusega liikide hulka.