

# TILS646

DEMO 1, 22.1.2019, klo 8:30.

Palauta R-tehtävien ratkaisut (R-koodit ja tulokset tulkintoineen) Koppaan (tehtävänpalautus) viimeistään demoja edeltävänä maanantaina klo 14:00 mennessä. Nimeä tiedosto etunimisukunimi\_demo1.R

Vain palautettuna tehtävistä saa demohyvityksiä. Jos ei pääse demoryhmään, demopisteistä saa vain puolet. Valmistaudu esittämään ratkaisusi demoryhmässä.

## 1. Aineistoon tutustuminen

Jyväskylän yliopiston bio- ja ympäristötieteiden laitoksen tutkijat Tapiro Mappes ja Mikael Puurtinen ovat tutkinneet altruismia eli epäitsekkyyttä erilaisten rahansijoituspelien avulla sekä kyselylomakkeella.

1. **tutkimuskysymys:** millaisia ihmisiä ihmisryhmiä epäitsekkyyden suhteeseen voidaan muodostaa kyselylomakkeen kysymysten perusteella?

Tutustutaan ensin altruismiin ja sitä kuvaaviin muuttuihin. Tietoa muuttujista ja tutkimuksesta löydät Kopasta tiedostoista [MarttilaToikkanenluktyo07skannattuversio.pdf](#) ja [altruismidatanmjat.pdf](#).

- Mitä altruismi on Marttilan ja Toikkasen kandidaatin tutkielman perusteella?
- Lue aineisto **altruismi.dat** R:ään ja käytä **str**-funktiota. Tutustu muuttuihin.
- Korjaa muuttujan B18E arvo 133.9 arvoksi 13.9 (M. Puurtinen). (Jatkuvilla muuttujilla maksimi aineistossa on n. 113mm (voi olla 120mm?) (M. Puurtinen).)

```
aineisto=read.table("altruismi.dat")
```

```
# 1a) altruismi Mattilan ja Toikkasen tutkielmaasta:  
# "Altruismi eli epäitsekkyys voidaan määritellä sellaisiksi teoiksi,  
# joista koituu itselle kustannuksia, mutta jotka ovat muilla hyödyksi."  
# 1b) Luetaan aineisto ja tutkitaan sitä  
aineisto=read.table("altruismi.dat", header=TRUE)  
names(aineisto)
```

```
## [1] "B7"   "B8"   "B9"   "B12"  "B13"  "B14"  "B15"  "B16"  "B18A" "B18B"  
## [11] "B18C" "B18D" "B18E" "B18F" "B19A" "B19B" "B20"
```

```
str(aineisto)
```

```
## 'data.frame': 192 obs. of 17 variables:  
## $ B7 : num 34.1 47.2 71.3 78.2 41.7 18.6 52.6 17.1 63.7 37 ...  
## $ B8 : Factor w/ 2 levels "No","Yes": 1 1 1 2 1 2 2 2 2 NA ...  
## $ B9 : num 56.1 7.8 17 26.4 64.7 59.7 54.3 24.7 75.2 16.2 ...  
## $ B12 : Factor w/ 3 levels "never","regularly",...: 1 1 3 1 3 3 1 1 1 1 ...  
## $ B13 : Factor w/ 2 levels "no","yes": 1 2 1 1 1 1 1 1 2 ...  
## $ B14 : Factor w/ 4 levels "I don't like",...: 2 2 4 2 2 4 4 1 2 2 ...  
## $ B15 : num 102 103.8 42.8 72.7 56.9 ...  
## $ B16 : Factor w/ 4 levels "I don't like",...: 2 2 2 1 1 4 1 1 4 4 ...  
## $ B18A: num 35 58.4 37.5 43.2 20.5 46.7 40.1 46.3 42.7 11.2 ...  
## $ B18B: num 19.2 18.4 10.9 36.8 7 34.9 27.9 47.6 32.6 53.3 ...
```

```

## $ B18C: num 13 6.1 12.2 10 8.2 2.2 26.6 46.1 73.9 10.1 ...
## $ B18D: num 18.3 6 12.6 5.5 9.4 43 11.7 61 34.8 52.4 ...
## $ B18E: num 27.6 3.9 14.2 26.5 9.7 35.6 10.3 23.2 35.9 53.3 ...
## $ B18F: num 53.8 51 44.7 38 33.1 82.7 53.5 51.5 53.7 37.4 ...
## $ B19A: num 55.4 8 51.8 47 24.6 ...
## $ B19B: num 11.8 64.8 50.9 79.9 32.1 57.2 85.6 52.9 70.1 92.2 ...
## $ B20 : num 49.8 75.5 84.1 52.5 81.6 75.5 51.9 53.2 83.5 75.6 ...

# head(aineisto)
# summary(aineisto)
# 1c) virheellisen arvon korjaus
index<-which(aineisto$B18E==133.9)
aineisto$B18E[index]<-13.9
# toinen ratkaisu
aineisto[, "B18E"] [aineisto$B18E == 133.9] <- 13.9

```

## 2. Muuttujien muunnokset

Pohdi, kannattaisiko joitakin faktoreita (eli nominaaliasteikollisia muuttujia) muuttaa järjestysasteikollisiksi muuttujiksi, jotta daisy-funktio käsittelisi niitä jatkuvina muuttujina. Toteuta muunnokset ja poista aineistosta päällekkäisyydet, jotta voit tehdä ryhmittelyanalyysin käyttäen kaikkia muuttujia.

```

names(aineisto)

## [1] "B7"    "B8"    "B9"    "B12"   "B13"   "B14"   "B15"   "B16"   "B18A"  "B18B"
## [11] "B18C"  "B18D"  "B18E"  "B18F"  "B19A"  "B19B"  "B20"

# korjataan samalla arvojen järjestys
aineisto$B12f <- ordered(aineisto$B12, levels = c("never", "sometimes", "regularly"))
aineisto$B14f <- ordered(aineisto$B14, levels = c("I don't like", "never", "sometimes", "regularly"))
aineisto$B16f <- ordered(aineisto$B16, levels = c("I don't like", "never", "sometimes", "regularly"))
aineisto<-aineisto[,c("B7", "B8", "B9", "B12f", "B13", "B14f", "B15",
                      "B16f", "B18A", "B18B", "B18C", "B18D", "B18E",
                      "B18F", "B19A", "B19B", "B20")]
names(aineisto)

## [1] "B7"    "B8"    "B9"    "B12f"  "B13"   "B14f"  "B15"   "B16f"  "B18A"  "B18B"
## [11] "B18C"  "B18D"  "B18E"  "B18F"  "B19A"  "B19B"  "B20"

```

## 3. pam- ja cluster.varstats-funktiot

- Tutustu funktionoon `pam` R-helppiin avulla. Kirjaa pääkohtia ylös: mm. mikä aineisto annetaan inputtina, muuttujien tyypivaatimuksista ja puuttuvien havaintojen käsittelystä.
- Tutustu `cluster.varstats`-funktioon R-helppiin avulla. Kirjaa pääkohtia ylös: Tutustu erityisesti argumentteihin: `vardata`, `tablevar`, `catvar`, `quantvar`, `catvarcats`. Tätä tietoa tarvitset seuraavassa tehtävässä.

```

library(cluster)
library(fpc)
?pam
# it minimizes a sum of dissimilarities instead of a sum of squared euclidean distances
# (k-means)

# x: input voi olla datakehikko, vaativat kuten dist-funktiolla:
# All variables must be numeric. Missing values (NAs) are allowed-as long as every
# pair of observations has at least one case not missing.

# x: input voi dissimilariiteettimatriisi, joka on laskettu daisy- tai
# dist-funktiolla, mutta dissimilariiteettimatriisin kohdalla ei saa olla puuttuvia

# metric: jos aineisto on datakehikko, metric annetaan erikseen

# stand: Measurements are standardized for each variable (column),
# by subtracting the variable's mean value and dividing by the variable's
# mean absolute deviation.

?cluster.varstats
# This function gives some helpful variable-wise information for cluster interpretation,
# given a clustering and a data set.
# vardata:
# data matrix or data frame of which variables are summarised.
# tablevar:
# vector of integers. Numbers of variables treated as categorical (i.e., no histograms
# and statistics, just tables) if clusterwise=TRUE. Note that an error will be
# produced by factor type variables unless they are declared as categorical here.
# catvar:
# vector of integers. Numbers of variables to be categorised by proportional quantiles
# for table computation. Recommended for all continuous variables.
# quantvar:
# vector of integers. Variables for which means, standard deviations and quantiles
# should be given out if clusterwise=TRUE.
# catvarcats:
# integer. Number of categories used for categorisation of variables specified in quantvar.

```

## 4. Altruismidatan ryhmittely K-medoids-menetelmällä, kun gower- etäisyys

- a) Kirjaa paperille käsin *daisy*-funktion erilaisuuusmitta samoilla merkinnöillä, kuten kalvomateriaalissaamme.
- b) Poista aineistosta havainnot, joilla puuttuvaa tietoa: Puuttuvat havainnot aiheuttavat e) kohdassa ongelmaa *cluster.varstat*-funktion kanssa, joten poista ne *complete.cases*-funktiolla.
- c) Laske dissimilariiteettimatriisi käyttäen *daisy*-funktiota ja gower-etaisyyttä.
- d) Toteuta aineistoon 3 ryhmän ryhmittely K-medoids-menetelmällä ja gower-etaisyydellä.
- e) Tulkitse ryhmät: Jatkuvatyypisten muuttujien kanssa *cluster.varstats*-funktio tulostaa perusmuodossaan ristiintaulukon jokaiselle jatkuvan muuttujan arvolle erikseen. Selvitä, pystyisikö funktion argumentteja: *vardata*, *tablevar*, *catvar*, *quantvar*, *catvarcats* muuttamaan siten, että jatkuviin muuttujien käsittely helpottuisi. Voit tehdä myös oman ratkaisun (*tapply*).

```

# 4b)
aineisto<-aineisto[complete.cases(aineisto),]
# 4c)
ddata<-daisy(aineisto, metric="gower")
# 4d)
pam3<-pam(ddata,3)
# 4e)
pam3stats<-cluster.varstats(clustering=pam3$clustering, vardata=aineisto,
tablevar=c(2,4,5,6,8), catvar=c(1,3,7,9:17),quantvar=c(1,3,7,9:17),
catvarcats=5, ask=FALSE, clusterwise=FALSE)
#pam3stats
#> pam3stats

```

Categorised B7

Cluster	1	2	3	4	5
1	17	16	23	11	18
2	12	12	7	12	12
3	7	7	6	11	6
Sum	36	35	36	34	36

B8

Cluster	No	Yes
1	69	16
2	55	0
3	1	36
Sum	125	52

Categorised B9

Cluster	1	2	3	4	5
1	19	15	14	16	21
2	9	11	13	13	9
3	8	9	8	6	6
Sum	36	35	35	35	36

B12f

Cluster	never	sometimes	regularly
1	25	48	12
2	8	30	17
3	10	20	7
Sum	43	98	36

B13

Cluster	no	yes
1	85	0
2	0	55
3	8	29
Sum	93	84

B14f

Cluster	I don't like	never	sometimes	regularly
1	4	53	23	5
2	0	33	16	6
3	1	18	13	5

Sum	5	104	52	16
-----	---	-----	----	----

Categorised B15

Cluster	1	2	3	4	5
1	22	21	12	17	13
2	6	8	15	12	14
3	8	6	8	6	9
Sum	36	35	35	35	36

B16f

Cluster	I don't like	never	sometimes	regularly
1	15	51	11	8
2	8	36	7	4
3	2	16	14	5
Sum	25	103	32	17

Categorised B18A

Cluster	1	2	3	4	5
1	15	17	19	14	20
2	11	7	11	14	12
3	10	11	5	7	4
Sum	36	35	35	35	36

Categorised B18B

Cluster	1	2	3	4	5
1	16	16	18	18	17
2	11	10	10	13	11
3	9	9	8	3	8
Sum	36	35	36	34	36

Categorised B18C

Cluster	1	2	3	4	5
1	20	16	15	17	17
2	9	9	13	10	14
3	8	9	8	7	5
Sum	37	34	36	34	36

Categorised B18D

Cluster	1	2	3	4	5
1	15	15	17	17	21
2	11	12	11	11	10
3	11	8	7	6	5
Sum	37	35	35	34	36

Categorised B18E

Cluster	1	2	3	4	5
1	13	19	17	19	17
2	10	9	9	12	15
3	13	8	9	3	4
Sum	36	36	35	34	36

Categorised B18F

Cluster	1	2	3	4	5
1	20	14	21	14	16

2	9	7	11	13	15
3	7	14	3	8	5
Sum	36	35	35	35	36

Categorised B19A

Cluster	1	2	3	4	5
1	17	18	14	19	17
2	12	10	15	6	12
3	7	7	6	10	7
Sum	36	35	35	35	36

Categorised B19B

Cluster	1	2	3	4	5
1	18	18	17	18	14
2	11	11	13	8	12
3	7	6	5	9	10
Sum	36	35	35	35	36

Categorised B20

Cluster	1	2	3	4	5
1	15	14	22	15	19
2	8	11	9	16	11
3	13	10	4	4	6
Sum	36	35	35	35	36

```
# karkeaa tulkintaa: ryhmäjaot eivät kovin selkeitä

#1. ryhmä: poliittisesti keskellä eniten, ei joukkuepeleille, hyväntekeväisyys (ei/joskus),
# ei ensiapukurssia, vapaaehtoistyötä suhteessa vähiten, liian paljon varoja
# kehitysyhteistyöhön, verenluovutusta suhteessa vähiten

# muut muuttujat aika tasaisan jakautuneita

#2. ryhmä: poliittisesti molemilla laidoilla (ammejakauma), ei joukkuepeleille,
# painopiste sosiaalisuuden puolella, hyväntekeväisyystyössä (joskus/säännöllisesti),
# kyllä ensiapukurssi, liian vähän rahaa kehitysyhteistyöhön, kyllä verenluovutusta

# muut muuttujat aika tasaisan jakautuneita

#3. ryhmä: poliittisesti kaikkea, kyllä joukkuepeleille, hyväntekeväisyys (ei/joskus),
# kyllä ensiapukurssi, vapaaehtoistyötä suhteessa eniten, kehitysyhteistyö tasajakauma,
# verenluovutusta suhteessa eniten, voittamisen haluisia suhteessa eniten,
# ystävällisyys eniten tärkeintä suhteessa muihin ryhmiin, pakolaisia
# Suomeen

# Huomataan, että ryhmittely ei ole kovin selkeä, mikä johtunee daisy-funktion
# tekemästä standardoinnista:

# ?daisy

# With that, each variable (column) is first standardized by dividing each entry
# by the range of the corresponding variable, after subtracting the minimum value;
# consequently the rescaled variable has range [0,1], exactly.
```

```
# sekä normalisoinnista:

# The contribution  $d(ij,k)$  of a nominal or binary variable to the total
# dissimilarity is 0 if both values are equal, 1 otherwise. The contribution of other
# variables is the absolute difference of both values, divided by the total range
# of that variable. Note that "standard scoring" is applied to ordinal variables,
# i.e., they are replaced by their integer codes 1:K. Note that this is not the
# same as using their ranks (since there typically are ties).
```

## 5. Ryhmittelyanalyysi K-medoids-menetelmällä, kun gower- etäisyys ja omat painot

- a) Toteuta aineistoon 3 ryhmän ryhmittely K-medoids-menetelmällä sekä gower-etaisyydellä ja omilla painoilla.
- b) Muuttuvatko ryhmien tulkinnat?

```
dim(aineisto)

## [1] 177 17

# 5a) painojen muodostus, tiedossani ei ole säätöä, millaisia painot pitäisi olla:
painot<-rep(1, 17)
names(aineisto)

## [1] "B7"    "B8"    "B9"    "B12f"  "B13"   "B14f"  "B15"   "B16f"  "B18A"  "B18B"
## [11] "B18C"  "B18D"  "B18E"  "B18F"  "B19A"  "B19B"  "B20"

painot[4]<-3 # hyväntekeväisyys
painot[8]<-3 # verenluovutus
ddata<-daisy(aineisto, metric="gower", weights=painot)
pam3<-pam(ddata,3)
# 5b)
pam3stats<-cluster.varstats(clustering=pam3$clustering, vardata=aineisto,
tablevar=c(2,4,5,6,8), catvar=c(1,3,7,9:17),quantvar=c(1,3,7,9:17),
catvarcats=5, ask=FALSE, clusterwise=FALSE)
pam3stats

##          Categorised  B7
## Cluster  1  2  3  4  5
##      1     18 15 23 10 17
##      2     12 11  9 15 13
##      3      6  9  4  9  6
##      Sum  36 35 36 34 36
##
##          B8
## Cluster  No Yes
##      1     63 20
##      2     39 21
```

```

##      3   23   11
##      Sum 125   52
##
##          Categorised  B9
## Cluster  1   2   3   4   5
##      1   16  14  13  18  22
##      2     9  12  12  15  12
##      3   11   9  10   2   2
##      Sum 36  35  35  35  36
##
##          B12f
## Cluster never sometimes regularly
##      1     29       52       2
##      2     14       46       0
##      3     0        0       34
##      Sum 43       98       36
##
##          B13
## Cluster no yes
##      1    83     0
##      2     0    60
##      3    10    24
##      Sum 93    84
##
##          B14f
## Cluster I don't like never sometimes regularly
##      1           4   51     25     3
##      2           1   39     12     8
##      3           0   14     15     5
##      Sum         5  104     52    16
##
##          Categorised  B15
## Cluster  1   2   3   4   5
##      1   21  23  13  14  12
##      2   12   8  17   9  14
##      3     3   4   5  12  10
##      Sum 36  35  35  35  36
##
##          B16f
## Cluster I don't like never sometimes regularly
##      1           14   49     14     6
##      2           6   37     11     6
##      3           5   17      7     5
##      Sum         25  103     32    17
##
##          Categorised  B18A
## Cluster  1   2   3   4   5
##      1   19  15  18  10  21
##      2   14  11  11  16   8
##      3     3   9   6   9   7
##      Sum 36  35  35  35  36
##
##          Categorised  B18B
## Cluster  1   2   3   4   5

```

```

##      1   15 17 16 17 18
##      2   14 11 13  8 14
##      3    7  7  7  9  4
## Sum 36 35 36 34 36
##
##          Categorised  B18C
## Cluster 1 2 3 4 5
##      1 21 19 13 13 17
##      2  9 11 14 11 15
##      3  7  4  9 10  4
## Sum 37 34 36 34 36
##
##          Categorised  B18D
## Cluster 1 2 3 4 5
##      1 17 16 16 14 20
##      2 13 12 11 13 11
##      3  7  7  8  7  5
## Sum 37 35 35 34 36
##
##          Categorised  B18E
## Cluster 1 2 3 4 5
##      1 14 20 16 18 15
##      2 13 13 10 10 14
##      3  9  3  9  6  7
## Sum 36 36 35 34 36
##
##          Categorised  B18F
## Cluster 1 2 3 4 5
##      1 18 17 20 13 15
##      2 11 13 12 12 12
##      3  7  5  3 10  9
## Sum 36 35 35 35 36
##
##          Categorised  B19A
## Cluster 1 2 3 4 5
##      1 14 15 12 21 21
##      2 14 14 13  8 11
##      3  8  6 10  6  4
## Sum 36 35 35 35 36
##
##          Categorised  B19B
## Cluster 1 2 3 4 5
##      1 18 17 14 20 14
##      2 10 12 14 11 13
##      3  8  6  7  4  9
## Sum 36 35 35 35 36
##
##          Categorised  B20
## Cluster 1 2 3 4 5
##      1 14 17 21 13 18
##      2 11 13 12 13 11
##      3 11  5  2  9  7
## Sum 36 35 35 35 36
##

```

```
# ryhmät muuttuvat ainakin siten, että painotettujen muuttujien arvojen jako  
# eri ryhmien välillä on selkeämpi
```

## 6. K-means-funktio

Tutustu funktioon `kmeans` R-helppoon avulla. Kirjaa päätökohtia ylös, mm. muuttujien tyyppeistä, erilaisuuksista, puuttuvien havaintojen käsittelystä, aloituskertojen määristä, outputista.

```
?kmeans  
# x  
# numeric matrix of data, or an object that can be coerced to such a matrix  
# (such as a numeric vector or a data frame with all numeric columns).  
  
# The data given by x are clustered by the k-means method, which aims to partition  
# the points into k groups such that the sum of squares from points to the assigned  
# cluster centres is minimized.  
  
# ei sano mitään puuttuvista, kokeilujen perusteella ei näytä toimivan puuttuvien kanssa,  
# joten ne on poistettava  
  
# trying several random starts (nstart > 1) is often recommended  
  
# output: W(C),B(C),T
```

## 7. Geeniaineiston ryhmittely K-means-menetelmällä

2. tutkimuskysymys: miten syöpäpotilaiden geenien perusteella tehdyt ryhmät ovat kytköksissä syöpätyypeihin?

Tarkastellaan geeniaineistoa `nci.data` ja syöpätyypiaineistoa `nci.label`, kuten Hastien et al. nettikirjassa (s. 512–514). Ryhmitellään geeniaineisto `nci.data` ( $N=64$ ,  $p=6830$ , mutta dimensiot eri pän kuin tavallisesti) K-means-menetelmällä kolmeen ryhmään ja tarkastellaan, miten eri syöpätyypit jakautuvat näiden ryhmien suhteeseen. Nettikirja on osoitteessa

<https://web.stanford.edu/~hastie/ElemStatLearn/>

a) Tutustu aineistoihin. <https://web.stanford.edu/~hastie/ElemStatLearn/datasets/>

```
nci<-read.table(url("https://web.stanford.edu/~hastie/ElemStatLearn/datasets/nci.data.csv"),  
                  sep=",",row.names=1,header=TRUE)  
label<-scan(url("https://web.stanford.edu/~hastie/ElemStatLearn/datasets/nci.label.txt"),  
            what="char")
```

b) Toteuta `k-means` aineistoon `nci.data` siten, että vaihdat rivit sarakkeiksi ja sarakkeet riveiksi.  
c) Vertaa klusteroinnin tuloksena saamiasi ryhmiä samojen henkilöiden syöpätyypeihin (`table`).

```

# 7a) Luetaan aineistot
nci.data <- read.table(url("https://web.stanford.edu/~hastie/ElemStatLearn/datasets/nci.data.csv"),
  sep=",",row.names=1,header=TRUE)
nci.label <- scan(url("https://web.stanford.edu/~hastie/ElemStatLearn/datasets/nci.label.txt"),
  what="char")
names(nci.data)

## [1] "s1"   "s2"   "s3"   "s4"   "s5"   "s6"   "s7"   "s8"   "s9"   "s10"  "s11"
## [12] "s12"  "s13"  "s14"  "s15"  "s16"  "s17"  "s18"  "s19"  "s20"  "s21"  "s22"
## [23] "s23"  "s24"  "s25"  "s26"  "s27"  "s28"  "s29"  "s30"  "s31"  "s32"  "s33"
## [34] "s34"  "s35"  "s36"  "s37"  "s38"  "s39"  "s40"  "s41"  "s42"  "s43"  "s44"
## [45] "s45"  "s46"  "s47"  "s48"  "s49"  "s50"  "s51"  "s52"  "s53"  "s54"  "s55"
## [56] "s56"  "s57"  "s58"  "s59"  "s60"  "s61"  "s62"  "s63"  "s64"

dim(nci.data)

## [1] 6830   64

class(nci.data)

## [1] "data.frame"

#LABELS suomeksi
#CNS= keskushermostosyöpä (syöpä aivoissa?)
#Renal= munuaissyöpä
#NSCLC= ei-pieni-solu keuhkosyöpä==keuhkosyöpä
#ovarian=munasarjasyöpä
#breast=rintasyöpä
#leukemia=leukemia & melanoma=melanoma
#colon=suolistosyöpä
#prostate=eturauhassyöpä
#reproto laboratorioissa tehtyjä kasvainvilkelmia

# 7b) Ryhmittely
tnci.data<-t(nci.data)
ryhmia3<-kmeans(tnci.data,3,nstart=100,iter.max=100)
ryhmia3$cluster

## s1   s2   s3   s4   s5   s6   s7   s8   s9   s10  s11  s12  s13  s14  s15  s16  s17  s18
## 3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3
## s19  s20  s21  s22  s23  s24  s25  s26  s27  s28  s29  s30  s31  s32  s33  s34  s35  s36
## 3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    1    1    1
## s37  s38  s39  s40  s41  s42  s43  s44  s45  s46  s47  s48  s49  s50  s51  s52  s53  s54
## 1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    3    1
## s55  s56  s57  s58  s59  s60  s61  s62  s63  s64
## 1    2    2    2    2    2    2    2    2    2

ryhmia3$tot.withinss

## [1] 215746.3

```

```
ryhmia3$betweenss
```

```
## [1] 52116.09
```

```
# 7c) Ryhmien ja syöpätyyppien vertailu
ncigeenienryhmatjasyopatyypit<-data.frame(ryhmia3$cluster,factor(nci.label))
taulu<-table(ncigeenienryhmatjasyopatyypit[,1],ncigeenienryhmatjasyopatyypit[,2])
taulu[3:1,]
```

```
##
##      BREAST CNS COLON K562A-repro K562B-repro LEUKEMIA MCF7A-repro
##    3      3   5     0          0          0          0          0
##    2      2   0     0          0          0          0          0
##    1      2   0     7          1          1          6          1
##
##      MCF7D-repro MELANOMA NSCLC OVARIAN PROSTATE RENAL UNKNOWN
##    3          0     1     7     6     2     9     1
##    2          0     7     0     0     0     0     0
##    1          1     0     2     0     0     0     0
```

```
taulu[c(3,1,2),]
```

```
##
##      BREAST CNS COLON K562A-repro K562B-repro LEUKEMIA MCF7A-repro
##    3      3   5     0          0          0          0
##    1      2   0     7          1          1          6          1
##    2      2   0     0          0          0          0          0
##
##      MCF7D-repro MELANOMA NSCLC OVARIAN PROSTATE RENAL UNKNOWN
##    3          0     1     7     6     2     9     1
##    1          1     0     2     0     0     0     0
##    2          0     7     0     0     0     0     0
```

```
# taulukko ei ole täsmälleen samanlainen kuin kirjassa, mutta näin voi olla, koska
# kyseessä lokaali optimi
```

## 8. Painot ja standardointi (teoriatehtävä)

Tarkastellaan erilaisuuusmittaa

$$d(x_i, x_{i'}) = \sum_{j=1}^p w_j d_j(x_{ij}, x_{i'j}).$$

Yksittäisen muuttujan  $x_j$  vaikutus havaintojen erilaisuuteen  $d(x_i, x_{i'})$  riippuu sen suhteellisesta vaikutuksesta valittuun kesiarvoiseen erilaisuuusmittaan yli kaikkien havaintoparien

$$\begin{aligned} \bar{d} &= \frac{1}{N^2} \sum_{i=1}^N \sum_{i'=1}^N d(x_i, x_{i'}) \\ &= \sum_{j=1}^p w_j \left[ \frac{1}{N^2} \sum_{i=1}^N \sum_{i'=1}^N d_j(x_{ij}, x_{i'j}) \right] = \sum_{j=1}^p w_j \bar{d}_j, \end{aligned}$$

missä

$$\bar{d}_j = \frac{1}{N^2} \sum_{i=1}^N \sum_{i'=1}^N d_j(x_{ij}, x_{i'j}).$$

a) Osoita, että erilaisuusmitalle  $d_j(x_{ij}, x_{i'j}) = (x_{ij} - x_{i'j})^2$

$$\bar{d}_j = \frac{1}{N^2} \sum_{i=1}^N \sum_{i'=1}^N (x_{ij} - x_{i'j})^2 = 2 \frac{1}{N} \sum_{i=1}^N (x_{ij} - \bar{x}_j)^2 = 2s_j^2,$$

missä  $s_j^2$  on muuttujan  $x_j$  otosvarianssi siten, että jakajana käytetään  $N - 1$  sijaan otoskokoa  $N$ .

b) Osoita, että standardoidun muuttujan tapauksessa ja painojen ollessa samat muuttujien vaikutukset ovat yhtä suuria.

## 9. Extraa: altruismidatan ryhmittely K-medoids-menetelmällä, kun manhattan-etäisyys

Huomattiin, että joissakin muuttujissa ei ollut eroa ryhmien välillä. Tutkitaan vielä sellaista ryhmittelyä, jossa ei ole tehty standardointeja.

```
pam3<-pam(aineisto,3,metric="manhattan")
pam3stats<-cluster.varstats(clustering=pam3$clustering, vardata=aineisto,
tablevar=c(2,4,5,6,8), catvar=c(1,3,7,9:17),quantvar=c(1,3,7,9:17),
catvarcats=5, ask=FALSE, clusterwise=FALSE)
pam3stats
```

```
##          Categorised  B7
## Cluster  1   2   3   4   5
##       1    11   9  21  20  20
##       2    18  18   9   5   4
##       3     7   8   6   9  12
##      Sum  36  35  36  34  36
##
##          B8
## Cluster No Yes
##      1    55  26
##      2    41  13
##      3    29  13
##      Sum 125  52
##
##          Categorised  B9
## Cluster  1   2   3   4   5
##       1    18  15  13  19  16
##       2    12  15  12   6   9
##       3     6   5  10  10  11
##      Sum  36  35  35  35  36
##
##          B12f
## Cluster never sometimes regularly
```

```

##      1     27     41     13
##      2      8     32     14
##      3      8     25      9
##    Sum    43     98     36
##
##          B13
## Cluster no yes
##      1    48    33
##      2    25    29
##      3    20    22
##    Sum   93    84
##
##          B14f
## Cluster I don't like never sometimes regularly
##      1        5    49     23      4
##      2        0    34     15      5
##      3        0    21     14      7
##    Sum      5   104     52     16
##
##          Categorised B15
## Cluster 1 2 3 4 5
##      1    26 15 20 10 10
##      2      2    7    4 20 21
##      3      8 13 11  5  5
##    Sum   36 35 35 35 36
##
##          B16f
## Cluster I don't like never sometimes regularly
##      1        18    43     13      7
##      2        7    28     10      9
##      3        0    32      9      1
##    Sum     25   103     32     17
##
##          Categorised B18A
## Cluster 1 2 3 4 5
##      1    25 22 20  9  5
##      2      2    3    4 19 26
##      3      9 10 11  7  5
##    Sum   36 35 35 35 36
##
##          Categorised B18B
## Cluster 1 2 3 4 5
##      1     8 14 17 23 19
##      2    25 12  8  3  6
##      3     3  9 11  8 11
##    Sum   36 35 36 34 36
##
##          Categorised B18C
## Cluster 1 2 3 4 5
##      1     9 15 17 19 21
##      2    21 16  9  3  5
##      3     7  3 10 12 10
##    Sum   37 34 36 34 36
##

```

```

##          Categorised  B18D
## Cluster  1  2  3  4  5
##      1    13  8 19 17 24
##      2    14 22 10  3  5
##      3    10  5  6 14  7
##      Sum 37 35 35 34 36
##
##          Categorised  B18E
## Cluster  1  2  3  4  5
##      1    12 15 17 19 18
##      2    20 16  5  4  9
##      3     4  5 13 11  9
##      Sum 36 36 35 34 36
##
##          Categorised  B18F
## Cluster  1  2  3  4  5
##      1    28 30 20  3  0
##      2     3  3 14 18 16
##      3     5  2  1 14 20
##      Sum 36 35 35 35 36
##
##          Categorised  B19A
## Cluster  1  2  3  4  5
##      1    21 23 20 14  3
##      2    13 11 10 12  8
##      3     2  1  5  9 25
##      Sum 36 35 35 35 36
##
##          Categorised  B19B
## Cluster  1  2  3  4  5
##      1    19 24 17 13  8
##      2    13  6 12  7 16
##      3     4  5  6 15 12
##      Sum 36 35 35 35 36
##
##          Categorised  B20
## Cluster  1  2  3  4  5
##      1    22 13 16 12 18
##      2     4 13 11 15 11
##      3    10  9  8  8  7
##      Sum 36 35 35 35 36
##

```

```
# Karkeaa tulkintaa:
```

```
#1. ryhmä: oikeistoa enemmän, enemmistö ei joukkuepelejä, sosiaalisuus tasajakauma,
# hyväntekeväisyys (eniten ei/joskus), ei ensiapukurssia, suhteessa vähiten palkatonta
# vapaaehtoistyötä ja verenluovutusta
```

```
#2. ryhmä: vasemmistoa enemmän, enemmistö ei joukkuepelejä, sosiaalisempia enemmän,
# hyväntekeväisyyttä (eniten joskus/säännöllisesti), kyllä ensiapukurssi,
# vapaaehtoistyötä ja verenluovutusta suhteessa enemmän kuin 1. ryhmässä, liian
# vähän rahaa kehitysyhteistyöhön
```

#3.ryhmä: oikeisto/vasemmisto, enemmistö ei joukkuepelejä, erakkoja enemmän,  
# hyväntekeväisyyttä (eniten ei/joskus), enimmäkseen ensiapukurssi,  
# vapaaehtoistyötä ja verenluovutusta suhteessa eniten, kehitysyhteistyön suhteeseen  
# tasajakauma

# ryhmien välillä hieman enemmän eroja voittamisen halun, myötätunnon, suvaitsevaisuuden,  
# oikeudenmukaisuuden, ystävällisyyden ja oman edun tavoittelun suhteeseen kuin aikaisemmissa  
# tehtävissä

# TILS646

DEMO 2, 29.1.2019, klo 8:30.

Palauta R-tehtävien ratkaisut (R-koodit ja tulokset tulkintoineen) Koppaan (tehtävänpalautus) viimeistään demoja edeltävänä maanantaina klo 18:00 mennessä. Nimeä tiedosto etunimisukunimi\_demo2.R

Vain palautettuina tehtävistä saa demohyvityksiä. Jos ei pääse demoryhmään, demopisteistä saa vain puolet. Valmistaudu esittämään ratkaisusi demoryhmässä.

## 1. Tutustu R:n helpin avulla seuraaviin funktioihin

- a) `agnes`, kirjaa päätökohtia ylös: mm. mikä aineisto annetaan inputtina, muuttujien tyypivaatimuksista, puuttuvien havaintojen käsitteelystä, mitä saadaan outputtina
- b) `plot.agnes`, joka kertoo, mitä piirretään, kun objektina on `agnes`-funktion antama objekti
- c) `summary.agnes`, kirjaa päätökohtia ylös: mitä tulostaa outputtina

```
library(cluster)
```

```
## Warning: package 'cluster' was built under R version 3.5.2
```

```
library(fpc)
```

```
## Warning: package 'fpc' was built under R version 3.5.2
```

```
library(clue)
```

```
## Warning: package 'clue' was built under R version 3.5.2
```

```
# 1a) agnes laskee hierarkkisen klusterointin,
# x: datamatriisi, datakehikko, dissimilariteettimatriisi
# tulostaa: agnes.object
```

```
# puuttuvia saa olla, jos datakehikko, mutta ei, jos dissimilariteettimatriisi
# helpin mukaan kaikki muuttujat pitäisi olla numeerisia
# (faktorit näyttävän käyvän)
```

```
# 1b)
# inputtina agnes-funktion objekti, tulostaa banner-plotin ja dendogrammin
# Banner-plot listaa etäisyydet, joilla havainnot ja klusterit yhdistetään.
# Havainnot tulisivat siinä järjestyksessä kuin agnes ne löytää. R-esimerkki:
agriculture
```

```
##      x     y
## B    16.8  2.7
## DK   21.3  5.7
## D    18.7  3.5
## GR   5.9  22.2
## E    11.4 10.9
## F    17.8  6.0
## IRL 10.9 14.0
## I    16.6  8.5
## L    21.0  3.5
## NL   16.4  4.3
## P    7.8  17.4
## UK   14.0  2.3
```

```

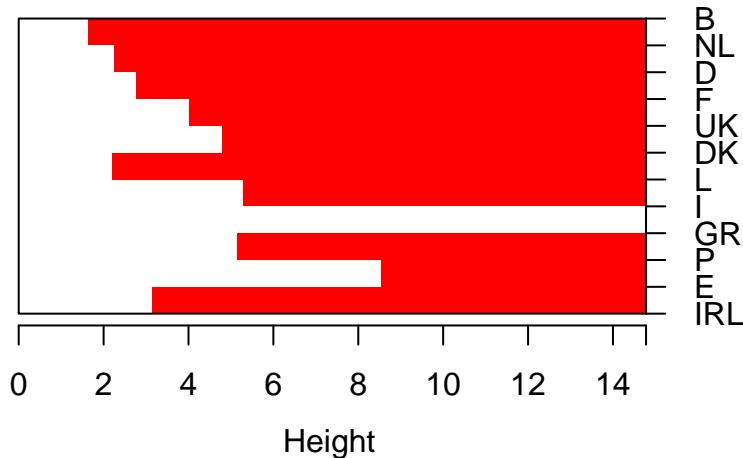
agnes(architecture)

## Call: agnes(x = agriculture)
## Agglomerative coefficient:  0.7818932
## Order of objects:
## [1] B   NL  D   F   UK  DK  L   I   GR  P   E   IRL
## Height (summary):
##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
##      1.649   2.509   4.027   4.966   5.228  14.780
##
## Available components:
## [1] "order"      "height"      "ac"          "merge"       "diss"        "call"
## [7] "method"     "order.lab"   "data"
agnes(architecture)$height

## [1] 1.649242 2.248356 2.769175 4.026768 4.788352 2.220360 5.294092
## [8] 14.779629 5.162364 8.550753 3.140064
plot(agnes(architecture), which.plots=1) # 1= banner, 2=dendrogrammi

```

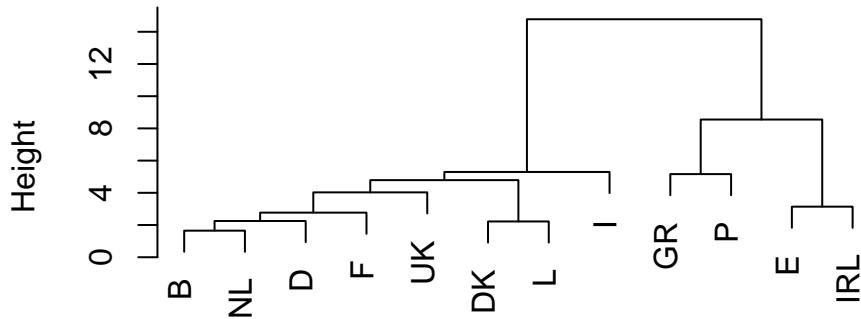
### Banner of agnes(x = agriculture)



Agglomerative Coefficient = 0.78

```
plot(agnes(architecture), which.plots=2)
```

## Dendrogram of agnes(x = agriculture)



agriculture  
Agglomerative Coefficient = 0.78

```
# 1c)
# ?summary.agnes
summary(agnes(agriculture))

## Object of class 'agnes' from call:
##   agnes(x = agriculture)
## Agglomerative coefficient:  0.7818932
## Order of objects:
## [1] B   NL  D   F   UK  DK  L   I   GR  P   E   IRL
## Merge:
##      [,1] [,2]
## [1,]    -1   -10
## [2,]    -2   -9
## [3,]     1   -3
## [4,]     3   -6
## [5,]    -5   -7
## [6,]     4   -12
## [7,]     6    2
## [8,]    -4   -11
## [9,]     7   -8
## [10,]    8    5
## [11,]    9   10
## Height:
## [1] 1.649242 2.248356 2.769175 4.026768 4.788352 2.220360 5.294092
## [8] 14.779629 5.162364 8.550753 3.140064
##
## 66 dissimilarities, summarized :
##   Min. 1st Qu. Median Mean 3rd Qu. Max.
##   1.649   4.357   7.987  9.594  13.250  24.035
## Metric : euclidean
## Number of objects : 12
##
```

```

## Available components:
## [1] "order"      "height"     "ac"          "merge"       "diss"        "call"
## [7] "method"     "order.lab"   "data"

# merge:
# an (n-1) by 2 matrix, where n is the number of observations. Row i of merge
# describes the merging of clusters at step i of the clustering. If a number
# j in the row is negative, then the single observation |j| is merged at
# this stage. If j is positive, then the merger is with the cluster formed
# at stage j of the algorithm.

# ac:
# the agglomerative coefficient, measuring the clustering structure of the dataset.
# For each observation i, denote by m(i) its dissimilarity to the first cluster it
# is merged with, divided by the dissimilarity of the merger in the final step of the
# algorithm. The ac is the average of all 1 - m(i). It can also be seen as
# the average width (or the percentage filled) of the banner plot. Because ac grows
# with the number of observations, this measure should not be used to compare datasets
# of very different sizes.

```

## 2. Tutustu R:n helpin avulla seuraaviin funktioihin

- a) `dist`, kirjaa pääkohtia ylös: mm. mikä aineisto annetaan inputtina, muuttujien tyypivatimuksista, puuttuvien havaintojen käsittelystä, mitä saadaan outputtina
- b) `diana`, kirjaa pääkohtia ylös: mm. mikä aineisto annetaan inputtina, muuttujien tyypivatimuksista, puuttuvien havaintojen käsittelystä, mitä saadaan outputtina
- c) `plot.diana`, joka kertoo, mitä piirretään, kun objektina on `diana`-funktion antama objekti
- d) `cluster.stats`, kirjaa pääkohtia ylös
- e) `cl_dissimilarity`, kirjaa pääkohtia ylös

```

# 2a)
# ?dist
# dist-funktiolle syötetään dataksi x=numeroerinen matriisi, datakehikko tai
# "dist"-objekti? ja palauttaa dist-objektin
# Puuttuvat havainnot ovat sallittuja, mutta laskennassa poistetaan koko rivi,
# jossa on puuttuvaa. Jos on suuria arvoja=Inf, niin nämä poistetaan,
# jos tulokseksi tulee NaN tai NA.
# Jos joitakin sarakkeita poistetaan, niin tietyillä etäisyysmitoilla summa skaalataan
# suhteessa sarakkeiden määrään.

# 2b)
# ?diana
# x: voi olla matriisi, datakehikko tai dissimilariteettimatriisi
#
# puuttuvia saa olla, jos datakehikko, mutta ei, jos dissimilariteettimatriisi
# helpin mukaan kaikki muuttujat pitäisi olla numeroisia
# (faktorit näyttävän käyvän)

# dc:
# the divisive coefficient, measuring the clustering structure of the dataset.
# For each observation i, denote by d(i) the diameter of the last cluster to
# which it belongs (before being split off as a single observation), divided
# by the diameter of the whole dataset. The dc is the average of all 1 - d(i).

```

```

# It can also be seen as the average width (or the percentage filled) of the banner
# plot. Because dc grows with the number of observations, this measure should
# not be used to compare datasets of very different sizes.

# 2c)
# ?plot.diana
# inputtina diana-funktion objekti, tulostaa banner-plotin ja dendogrammin
# samanlainen kuin plot.agnes

# 2d)
# ?cluster.stats
# Computes a number of distance based statistics, which can be used for
# cluster validation, comparison between clusterings and decision about the
# number of clusters: cluster sizes, cluster diameters, average distances within
# and between clusters, cluster separation, biggest within cluster gap, average
# silhouette widths, the Calinski and Harabasz index, a Pearson version of
# Hubert's gamma coefficient, the Dunn index and two indexes to assess the
# similarity of two clusterings, namely the corrected Rand index and Meila's VI.

# pearsongamma:
# correlation between distances and a 0-1-vector where 0 means same cluster, 1 means
# different clusters. "Normalized gamma" in Halkidi et al. (2001).

```

### 3. Jakavan menetelmän dendogrammista

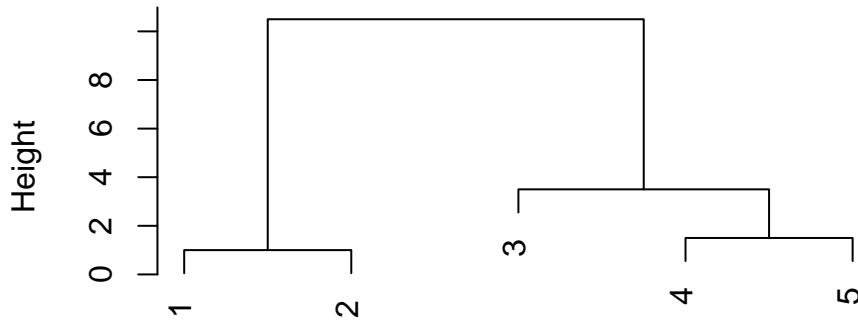
Lataa aineistoon luentokalvojen s. 46 X-matriisi, toteuta ryhmittely jakavalla menetelmällä (manhattan-täisyydellä) ja piirrä dendogrammi. Selvitä, miten dendogrammi on piirretty R:n helpin ja `dist`-funktion avulla.

```

X<-data.frame(x1=c(1,2,5,6,6.5),x2=c(1,1,4,5,6))
Xclust<- diana(X, metric="manhattan",keep.diss=T)
plot(Xclust, which.plots=2)

```

dendrogram of diana(x = X, metric = "manhattan", keep.d



X  
Divisive Coefficient = 0.84

```
dist(X[c(1,5),c(1,2)],method="manhattan")
```

```
##      1  
## 5 10.5
```

```
dist(X[c(3,5),c(1,2)],method="manhattan")
```

```
##      3  
## 5 3.5
```

```
dist(X[c(4,5),c(1,2)],method="manhattan")
```

```
##      4  
## 5 1.5
```

```
dist(X[c(1,2),c(1,2)],method="manhattan")
```

```
##      1  
## 2 1
```

#### 4. Altruismiaineiston ryhmittely hierarkkisilla menetelmillä

Tutkimuskysymys: millaisia ihmisiä ihmisiä etäisyyden suhteen voidaan muodostaa kyselylomakkeen kysymysten perusteella?

Käytetään erilaisuuusmittana manhattan-etäisyyttä. Toteuta ryhmittely ja piirrä dendrogrammi, kun yhdislevän menetelmän linkkifunktiona on

- a) keskimääräinen etäisyys
- b) lähin naapuri
- c) kaukaisin naapuri

ja kun menetelmää on

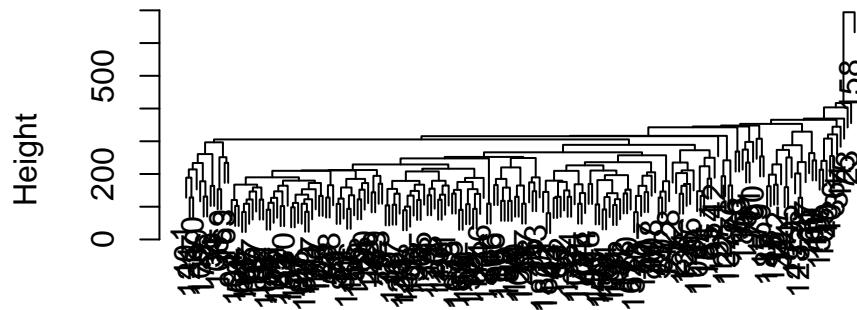
- d) Wardin menetelmä
- e) jakava menetelmä

- f) Miten ryhmittely onnistuu dendogrammien perusteella, mikä menetelmä tuottaisi selvän rakenteen?  
 g) Laske kofeneettiset korrelaatiokertoimet, CPCC:t, eri menetelmissä. Tulkitse arvoja.  
 h) Minkä hierarkkisen menetelmän valitsisit jatkoanalyyseihin?

```
altruismi=read.table("altruismi.dat", header=TRUE)
altruismi[, "B18E"] [altruismi$B18E == 133.9] <- 13.9
altruismi$B12f <- ordered(altruismi$B12, levels = c("never", "sometimes", "regularly"))
altruismi$B14f <- ordered(altruismi$B14, levels = c("I don't like", "never", "sometimes", "regularly"))
altruismi$B16f <- ordered(altruismi$B16, levels = c("I don't like", "never", "sometimes", "regularly"))
altruismi<-altruismi[,c("B7", "B8", "B9", "B12f", "B13", "B14f", "B15",
                       "B16f", "B18A", "B18B", "B18C", "B18D", "B18E",
                       "B18F", "B19A", "B19B", "B20")]

# a) keskiarvo
altave=agnes(altruismi, metric="manhattan", method="average", keep.diss=T)
plot(altave, which.plots=2)
```

**rogram of agnes(x = altruismi, metric = "manhattan",  
 "average", keep.diss = T)**



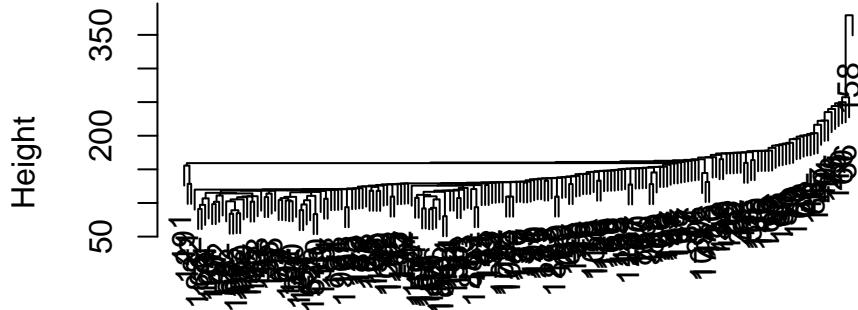
altruismi  
 Agglomerative Coefficient = 0.77

```
# ac=0.77
# summary(altave)

# ketjuttuu

# b) lähin naapuri
altsingle=agnes(altruismi, metric="manhattan", method="single", keep.diss=T)
plot(altsingle, which.plots=2)
```

```
lrogram of agnes(x = altruismi, metric = "manhattan",
"single", keep.diss = T)
```



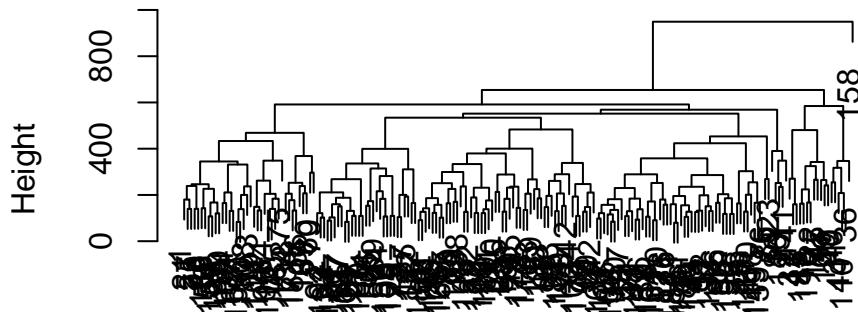
altruismi  
Agglomerative Coefficient = 0.64

```
# ac=0.64
# summary(altsingle)

# ketjuttuu

# c) kaukaisin naapuri
altcomp=agnes(altruismi, metric="manhattan", method="complete", keep.diss=T)
plot(altcomp, which.plots=2)
```

```
lrogram of agnes(x = altruismi, metric = "manhattan",
"complete", keep.diss = T)
```



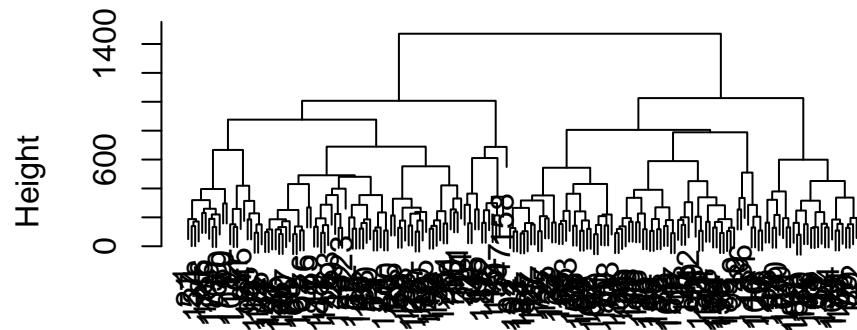
altruismi  
Agglomerative Coefficient = 0.83

```
# ac=0.83
# summary(altcomp)
```

```
# ei ketjutu

# d) Wardin menetelma
altward=agnes(altruismi, metric="manhattan", method="ward", keep.diss=T)
plot(altward, which.plots=2)
```

**Output of agnes(x = altruismi, metric = "manhattan", method = "ward", keep.diss = T)**



altruismi  
Agglomerative Coefficient = 0.89

```
# ac=0.89
# summary(altward)

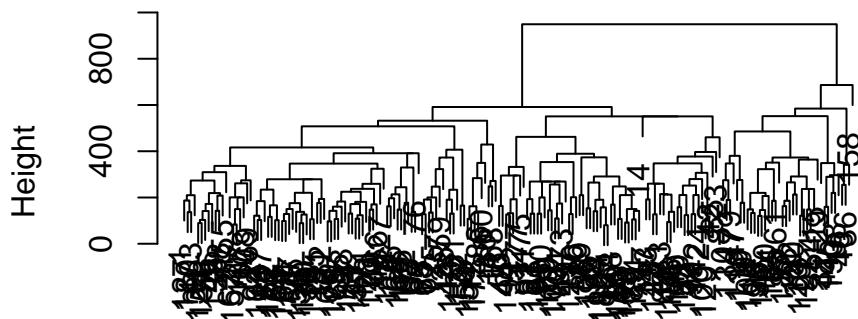
# ei ketjutu

# e) jakava menetelma
altdiana=diana(altruismi, metric="manhattan",keep.diss=T)
names(altdiana)

## [1] "order"      "height"     "dc"          "merge"       "diss"        "call"
## [7] "order.lab"  "data"

plot(altdiana, which.plots=2)
```

## Dendrogram of diana(x = altruismi, metric = "manhattan", keep.in.full = TRUE)



altruismi  
Divisive Coefficient = 0.81

```
# dc=0.81

# summary(altdiana)

# ei ketjutu, mutta jako ei niin selvä kuin valinnoilla complete ja ward

# f) kaukaisin ja Ward olisivat selkeimpiä rakenteeltaan

# g) kofeneettiset korrelaatiot
# library(clue)
#?cl_dissimilarity
# "cophenetic"
#  $1 - c \bar{2}$ , where  $c$  is the cophenetic correlation coefficient (i.e., the product-moment
# correlation of the ultrametrics).
sqrt(1-cl_dissimilarity(altave, altave$diss, method="cophenetic"))

## Cross-dissimilarities using cophenetic correlations:
##      [,1]
## [1,] 0.7284088
sqrt(1-cl_dissimilarity(altsingle, altsingle$diss, method="cophenetic"))

## Cross-dissimilarities using cophenetic correlations:
##      [,1]
## [1,] 0.7227473
sqrt(1-cl_dissimilarity(altcomp, altcomp$diss, method="cophenetic"))

## Cross-dissimilarities using cophenetic correlations:
##      [,1]
## [1,] 0.520954
sqrt(1-cl_dissimilarity(altward, altward$diss, method="cophenetic"))

## Cross-dissimilarities using cophenetic correlations:
```

```

##      [,1]
## [1,] 0.2865323
sqrt(1-cl_dissimilarity(altdiana, altdiana$diss, method="cophenetic"))

## Cross-dissimilarities using cophenetic correlations:
##      [,1]
## [1,] 0.4371298

# - Wardin menetelmän CPCC=0.29, muut isompiä
# - Wardin menetelmän ac-arvo suurin
# -> ristiriitaisia tuloksia CPCC:n vs. AC: & DC:n suhteeseen

# Näitten molempien perusteella ehkä kaukaisin menetelmä paras, mutta ryhmien jaot
# ja tulkinnat syttää katsoa

```

## 5. Satunnaisuuden testaus, kun Wardin menetelmä

Valitaan Wardin menetelmä satunnaisuuden testaukseen. Tee funktio, jolla voit simuloida altruismidatoja, joista lasket CPCC:n. Simuloi altruismidatoja ja CPCC:n jakaumaa. Mitä miettä olet satunnaisuuden hypoteesista simulointiesi perusteella?

```
summary(altruismi)
```

```

##      B7       B8       B9          B12f      B13
## Min.   : 0.10  No   :132  Min.   : 0.00  never   : 48  no  :102
## 1st Qu.:26.32 Yes  : 58  1st Qu.: 20.30 sometimes:102 yes: 90
## Median :49.65 NA's:  2   Median : 37.35 regularly: 41
## Mean   :45.56           Mean   : 41.44 NA's    :  1
## 3rd Qu.:58.52           3rd Qu.: 57.33
## Max.   :96.00           Max.   :108.50
## NA's   :4

##      B14f      B15          B16f      B18A
## I don't like: 6   Min.   : 0.00  I don't like: 29  Min.   : 0.00
## never        :111  1st Qu.: 56.65  never       :108  1st Qu.: 30.95
## sometimes     : 57  Median  : 65.60  sometimes    : 36  Median  : 43.70
## regularly     : 18  Mean    : 68.22  regularly    : 17  Mean    : 49.02
##                   3rd Qu.: 83.80  NA's      :  2  3rd Qu.: 69.40
##                   Max.   :112.60           Max.   :111.80
##                   NA's   :1           NA's   :1

##      B18B      B18C          B18D      B18E
## Min.   : 0.00  Min.   : 0.000  Min.   : 0.000  Min.   : 0.00
## 1st Qu.:12.10 1st Qu.: 6.175  1st Qu.: 3.775  1st Qu.: 5.55
## Median :22.70  Median :15.200  Median :13.850  Median :14.75
## Mean   :26.10  Mean   :20.153  Mean   :17.574  Mean   :18.55
## 3rd Qu.:33.45 3rd Qu.:26.675  3rd Qu.:23.900  3rd Qu.:26.82
## Max.   :114.00 Max.   :111.200 Max.   :111.200 Max.   :112.00
## NA's   :1

##      B18F      B19A          B19B      B20
## Min.   : 0.00  Min.   : 0.00  Min.   : 0.00  Min.   : 0.40
## 1st Qu.:37.85 1st Qu.: 28.70 1st Qu.: 44.75 1st Qu.: 59.60
## Median :56.40  Median :50.40  Median :64.60  Median :77.65
## Mean   :56.57  Mean   :48.61  Mean   :60.56  Mean   :75.47
## 3rd Qu.:76.85 3rd Qu.: 64.75 3rd Qu.: 80.10 3rd Qu.: 91.20
## Max.   :109.60 Max.   :112.00 Max.   :109.10 Max.   :112.20

```

```

##   NA's     :1           NA's     :1           NA's     :1
dim(altruismi)

## [1] 192 17
cpccward<-sqrt(1-cl_dissimilarity(altward, altward$diss, method="cophenetic"))
cpccward

## Cross-dissimilarities using cophenetic correlations:
##          [,1]
## [1,] 0.2865323

genaltruismidata <- function(N)
{# Funktiolla generoidaan yksi altruismidata, jossa on N henkeä.
# Muuttujien arvojen vaihteluvälit haetaan datasta.
sim <- data.frame(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17)
for(i in 1:N) { # yksi uusi satunnainen vastaus N kertaa
  B7 <- runif(1,0.10,96); B8 <- sample(1:2,1)
  B9 <- runif(1,0,108.50); B12 <- sample(1:3,1)
  B13 <- sample(1:2,1); B14 <- sample(1:4,1)
  B15 <- runif(1,0,112.60); B16 <- sample(1:4,1)
  B18A <- runif(1,0.0,111.80); B18B <- runif(1,0,114.0)
  B18C <- runif(1,0,111.2); B18D <- runif(1,0,111.2)
  B18E <- runif(1,0,112.0); B18F <- runif(1,0,109.60)
  B19A <- runif(1,0,112); B19B=runif(1,0,109.10); B20=runif(1,0.40,112.20)
  sim[i,] <- c(B7,B8,B9,B12,B13,B14,B15,B16,B18A, B18B, B18C,B18D,B18E,B18F,B19A,B18D,B18E)
}
sim
}

testausward <- function(m=100, N=192) {
# Funktiolla generoidaan m kappaletta uusia
# altruismidatoja, joissa on N henkeä.
# Klusterointi jokaiselle simuloidulle datalle
# Wardin menetelmällä. Tuloksesta lasketaan
# kofeneettinen korrelatiokerroin.
cpcc=NULL
for (i in 1:m) {
  simaltruismi=genaltruismidata(N=188)
  tulos=agnes(simaltruismi, metric="manhattan", method="ward", keep.diss=T)
  cpcc[i]=sqrt(1-cl_dissimilarity(tulos, tulos$diss,method="cophenetic"))
}
cpcc
}
testaustulos=testausward(m=100,N=192)
sort(testaustulos)

## [1] 0.3714578 0.3729975 0.3752974 0.3770315 0.3786688 0.3790475 0.3791858
## [8] 0.3863445 0.3873597 0.3879806 0.3921311 0.3955139 0.3959408 0.3961821
## [15] 0.3963134 0.3963498 0.3972205 0.3980170 0.3983691 0.3994578 0.3996932
## [22] 0.3997167 0.4003440 0.4009357 0.4018262 0.4022860 0.4029324 0.4032597
## [29] 0.4041094 0.4053774 0.4079612 0.4085975 0.4086963 0.4117889 0.4121359
## [36] 0.4121498 0.4123006 0.4129782 0.4130348 0.4132984 0.4133038 0.4133377
## [43] 0.4148380 0.4150836 0.4157483 0.4157665 0.4167933 0.4172150 0.4173746
## [50] 0.4182186 0.4189727 0.4190426 0.4192858 0.4193416 0.4198072 0.4201321

```

```

## [57] 0.4220506 0.4222275 0.4224016 0.4228432 0.4234803 0.4247086 0.4248708
## [64] 0.4250996 0.4253703 0.4271312 0.4273573 0.4276931 0.4289272 0.4298744
## [71] 0.4307764 0.4315628 0.4320683 0.4333504 0.4338209 0.4339781 0.4341618
## [78] 0.4344713 0.4371335 0.4371816 0.4382850 0.4385260 0.4392543 0.4407384
## [85] 0.4441474 0.4443279 0.4447306 0.4454808 0.4466850 0.4469657 0.4497529
## [92] 0.4521644 0.4529343 0.4534867 0.4556312 0.4591294 0.4634926 0.4649448
## [99] 0.4694065 0.4746700

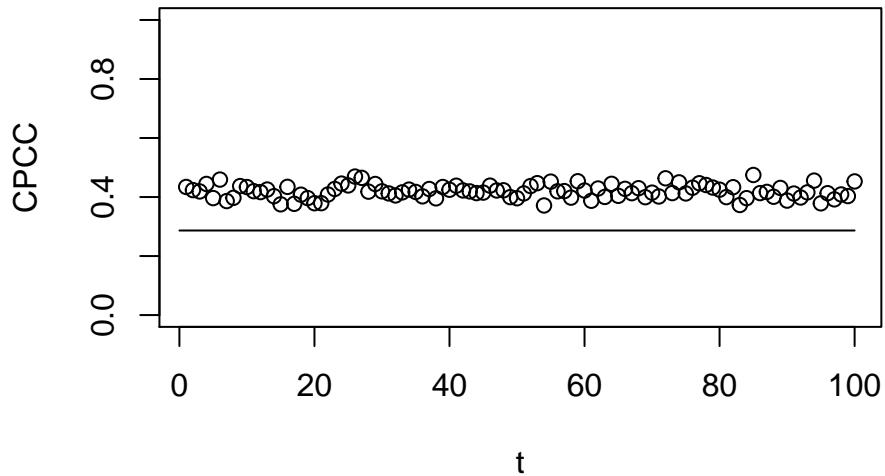
parvo<-length(testaustulos[testaustulos >= as.numeric(cpccward)])/100
parvo

## [1] 1

# p-arvon tulkinta: aineistossa ei olisi klusterirakennetta

plot(testaustulos, ylab="CPCC", xlab="t", ylim=c(0,1))
lines(c(0,100),c(as.numeric(cpccward), as.numeric(cpccward)))

```



```

# muokataan arvoalueita ja testataan, onko vaikutusta:

genaltruismidata2 <- function(N)
{# Funktiolla generoidaan yksi altruismidata, jossa on N henkeä.
# Muuttujien arvojen vaiheteluväli annettujen tietojen perusteella.
sim <- data.frame(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17)
for(i in 1:N) { # yksi uusi satunnainen vastaus N kertaa
  B7 <- runif(1,0,120); B8 <- sample(1:2,1)
  B9 <- runif(1,0,120); B12 <- sample(1:3,1)
  B13 <- sample(1:2,1); B14 <- sample(1:4,1)
  B15 <- runif(1,0,120); B16 <- sample(1:4,1)
  B18A <- runif(1,0,0,120); B18B <- runif(1,0,120)
  B18C <- runif(1,0,120); B18D <- runif(1,0,120)
  B18E <- runif(1,0,120); B18F <- runif(1,0,120)
  B19A <- runif(1,0,120); B19B=runif(1,0,120); B20=runif(1,0,120)
  sim[i,] <- c(B7,B8,B9,B12,B13,B14,B15,B16,B18A, B18B, B18C,B18D,B18E,B18F,B19A,B18D,B18E)
}
```

```

    }
sim
}

testausward2 <- function(m=100, N=192) {
# Funktiolla generoidaan m kappaletta uusia
# altruismidatajoa, joissa on N henkeä.
# Klusterointi jokaiselle simuloidulle datalle
# Wardin menetelmällä. Tuloksesta lasketaan
# kofeneettinen korrelaatiokerroin.
cpcc=NULL
for (i in 1:m) {
  simaltruismi=genaltruismidata2(N=188)
  tulos=agnes(simaltruismi, metric="manhattan", method="ward", keep.diss=T)
  cpcc[i]=sqrt(1-cl_dissimilarity(tulos, tulos$diss, method="cophenetic"))
}
cpcc
}

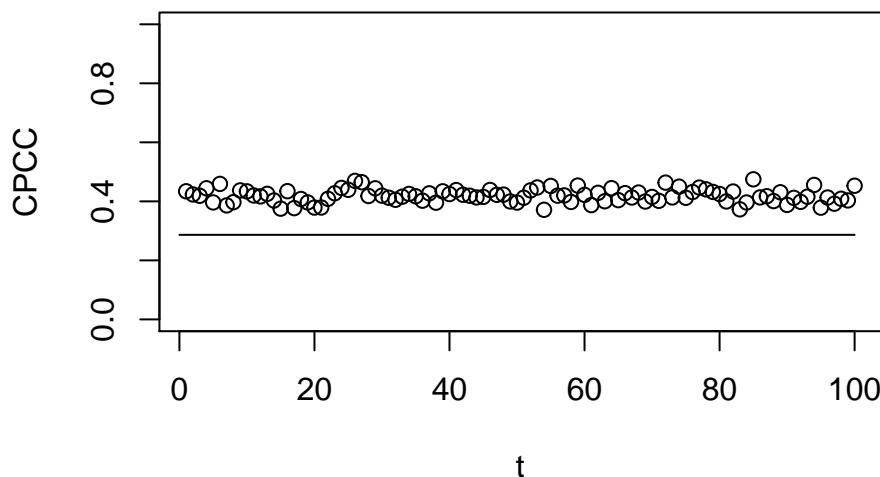
testaustulos2=testausward2(m=100,N=192)

parvo<-length(testaustulos[testaustulos >= as.numeric(cpccward)])/100
parvo

## [1] 1
# p-arvon tulkinta: aineistossa ei olisi klusterirakennetta

plot(testaustulos, ylab="CPCC", xlab="t", ylim=c(0,1))
lines(c(0,100),c(as.numeric(cpccward), as.numeric(cpccward)))

```



```
# Miten muut menetelmät?
```

```

cpccaukaisin<-sqrt(1-cl_dissimilarity(altcomp, altcomp$diss, method="cophenetic"))
cpccaukaisin

## Cross-dissimilarities using cophenetic correlations:
##          [,1]
## [1,] 0.520954

testauskaukaisin <- function(m=100, N=192) {
# Funktiolla generoidaan m kappaletta uusia
# altruismidatoja, joissa on N henkeä.
# Klusterointi jokaiselle simuloidulle datalle
# Wardin menetelmällä. Tuloksesta lasketaan
# kofeneettinen korrelaatiokerroin.
cpcc=NULL
for (i in 1:m) {
  simaltruismi=genaltruismidata(N=188)
  tulos=agnes(simaltruismi, metric="manhattan", method="complete", keep.diss=T)
  cpcc[i]=sqrt(1-cl_dissimilarity(tulos, tulos$diss, method="cophenetic"))
}
cpcc
}

testaustulos=testauskaukaisin(m=100,N=192)
sort(testaustulos)

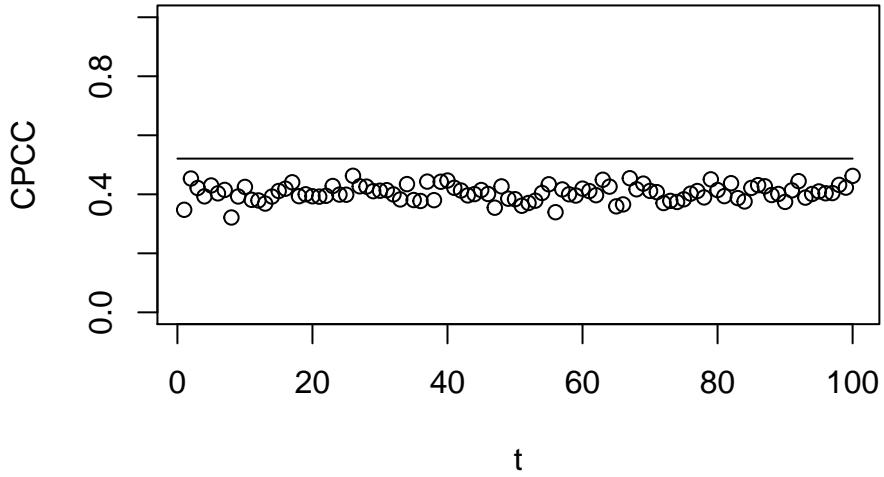
##      [1] 0.3214128 0.3393461 0.3475184 0.3545692 0.3592199 0.3612336 0.3658071
##      [8] 0.3685749 0.3704185 0.3705411 0.3740249 0.3742988 0.3757604 0.3773623
##     [15] 0.3776612 0.3783314 0.3792192 0.3801901 0.3805660 0.3812553 0.3825831
##     [22] 0.3831071 0.3834651 0.3851628 0.3870986 0.3887768 0.3893701 0.3920942
##     [29] 0.3923363 0.3926443 0.3930282 0.3933512 0.3936850 0.3938584 0.3949996
##     [36] 0.3958528 0.3962939 0.3970331 0.3975930 0.3986061 0.3990923 0.3995822
##     [43] 0.4000814 0.4001042 0.4007031 0.4010205 0.4010351 0.4011630 0.4022475
##     [50] 0.4031827 0.4034676 0.4042287 0.4044517 0.4073133 0.4097108 0.4106644
##     [57] 0.4109524 0.4111164 0.4112703 0.4115224 0.4121828 0.4128690 0.4133088
##     [64] 0.4140042 0.4142939 0.4144792 0.4148385 0.4164778 0.4169151 0.4190592
##     [71] 0.4191928 0.4213259 0.4216116 0.4221013 0.4226805 0.4249817 0.4260119
##     [78] 0.4260227 0.4268034 0.4270531 0.4271632 0.4282059 0.4299515 0.4310240
##     [85] 0.4319989 0.4339506 0.4345884 0.4362359 0.4376233 0.4407183 0.4425355
##     [92] 0.4430631 0.4445822 0.4466938 0.4493168 0.4506621 0.4534935 0.4544928
##     [99] 0.4623867 0.4627407

parvo<-length(testaustulos[testaustulos >= as.numeric(cpccaukaisin)])/100
parvo

## [1] 0
# p-arvon tulkinta: aineistossa olisi klusterirakennetta

plot(testaustulos, ylab="CPCC", xlab="t", ylim=c(0,1))
lines(c(0,100),c(as.numeric(cpccaukaisin), as.numeric(cpccaukaisin)))

```



```
cpccdiana<-sqrt(1-cl_dissimilarity(altdiana, altdiana$diss, method="cophenetic"))
cpccdiana
```

```
## Cross-dissimilarities using cophenetic correlations:
##      [,1]
## [1,] 0.4371298

testausdiana <- function(m=100, N=192) {
# Funktiolla generoidaan m kappaletta uusia
# altruismidatoja, joissa on N henkeä.
# Klusterointi jokaiselle simuloidulle datalle
# Wardin menetelmällä. Tuloksesta lasketaan
# kofeneettinen korrelaatiokerroin.
cpcc=NULL
for (i in 1:m) {
  simaltruismi=genaltruismidata(N=188)
  tulos=diana(simaltruismi, metric="manhattan", keep.diss=T)
  cpcc[i]=sqrt(1-cl_dissimilarity(tulos, tulos$diss, method="cophenetic"))
}
cpcc
}

testaustulos=testauskaukaisin(m=100, N=192)
sort(testaustulos)

## [1] 0.3315313 0.3502852 0.3571905 0.3585044 0.3587549 0.3618189 0.3627359
## [8] 0.3656670 0.3700766 0.3703400 0.3718508 0.3723163 0.3744713 0.3745878
## [15] 0.3750220 0.3751861 0.3769462 0.3781357 0.3795150 0.3795166 0.3804074
## [22] 0.3805935 0.3807224 0.3816222 0.3817733 0.3829926 0.3839921 0.3843450
## [29] 0.3857180 0.3864160 0.3870095 0.3885407 0.3886888 0.3887212 0.3888197
## [36] 0.3888507 0.3890064 0.3892321 0.3898778 0.3909725 0.3920150 0.3923132
## [43] 0.3929001 0.3938317 0.3938754 0.3946240 0.3951451 0.3961190 0.3965492
## [50] 0.3978792 0.3994449 0.3996585 0.4007616 0.4035029 0.4045470 0.4045638
```

```

## [57] 0.4049682 0.4050480 0.4054075 0.4055310 0.4057831 0.4060923 0.4092206
## [64] 0.4095137 0.4095241 0.4101304 0.4112703 0.4116596 0.4117612 0.4118231
## [71] 0.4130245 0.4134670 0.4139667 0.4162539 0.4164660 0.4166334 0.4174810
## [78] 0.4179643 0.4187433 0.4203777 0.4213202 0.4219504 0.4220559 0.4233769
## [85] 0.4252998 0.4271549 0.4285924 0.4325515 0.4329323 0.4352869 0.4368501
## [92] 0.4379469 0.4388497 0.4395420 0.4438457 0.4440959 0.4514849 0.4581510
## [99] 0.4595682 0.4641675

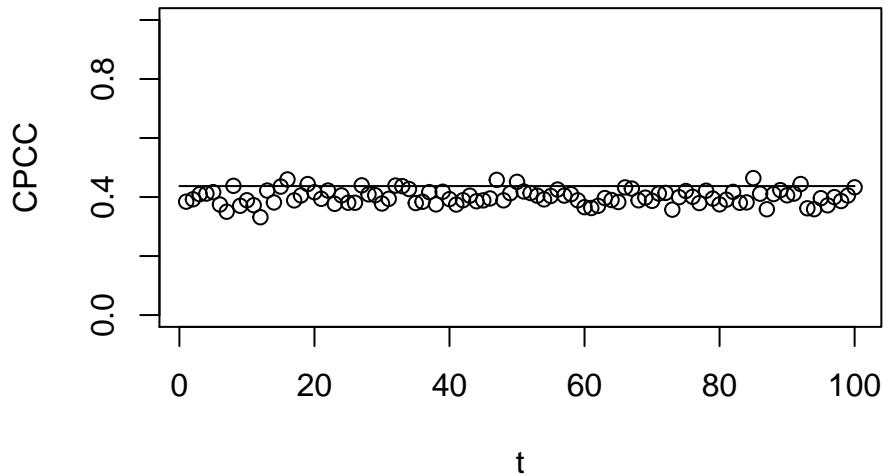
parvo<-length(testaustulos[testaustulos >= as.numeric(cpccdiana)])/100
parvo

## [1] 0.09

# p-arvon tulkinta: aineistossa voisi olla klusterirakennetta

plot(testaustulos, ylab="CPCC", xlab="t", ylim=c(0,1))
lines(c(0,100),c(as.numeric(cpccdiana), as.numeric(cpccdiana)))

```



## 7. Menetelmien vertailu

- a) Arvioi Wardin menetelmän tapauksessa sopivaa ryhmien määrää  $k$ . Valitse ryhmien määrä.
- b) Arvioi K-medoids-menetelmälle sopivaa ryhmien määrää  $k$ . Valitse ryhmien määrä.
- c) Kumman menetelmän valitsisit ja millä perusteella?

```

# Seuraavaksi tutkitaan katkaistuja hierarkioita:
# katkaisut cutree-funktiolla, ts. annetaan kullekin
# yksilölle ryhmän numero, esim.

```

```

altward7 <- cutree(altward, 7)
altward7

```

```

## [1] 1 2 3 4 2 5 4 1 1 4 5 4 4 6 6 3 3 3 7 4 1 6 4 5 4 4 4 4 3 7 6 2 2 3
## [36] 7 2 6 5 7 6 3 4 7 5 2 6 6 5 2 4 5 5 4 4 4 2 1 4 3 1 1 2 4 4 2 3 2 4 3 2

```

```

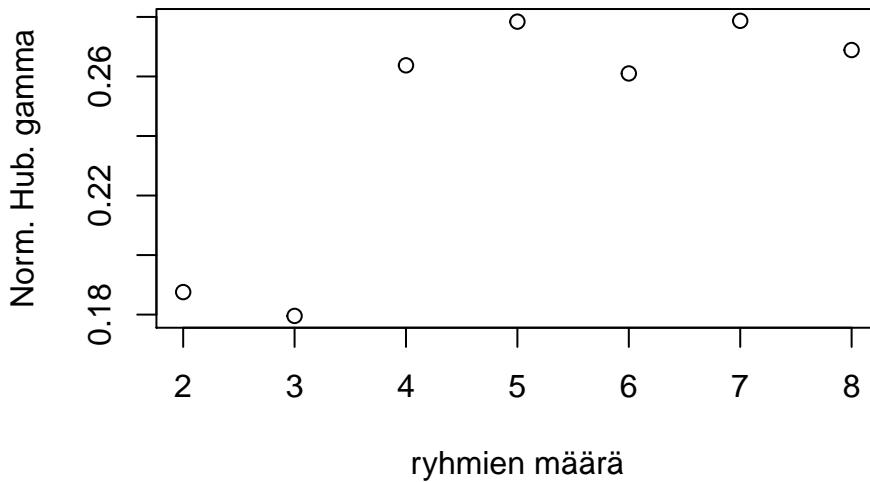
## [71] 4 2 3 3 1 4 5 4 1 4 2 4 4 1 4 4 5 3 5 3 3 2 3 1 5 3 5 7 1 2 5 5 4
## [106] 2 1 2 5 7 4 1 4 1 4 6 4 3 4 4 5 2 4 2 2 4 2 4 7 4 4 4 5 1 3 4 2 2 6 4
## [141] 4 5 3 5 6 4 4 3 3 2 2 3 3 2 4 4 2 6 3 5 4 4 2 3 4 3 4 4 1 5 1 4 3 3 1
## [176] 3 5 5 5 4 3 5 1 3 1 5 6 3 3 4 3 2

# a) Ward
pearsongammatward=function(aineisto,kmax=8)
{# Lasketaan pearsongammat Wardin menetelmälle, kun ryhmien määrä 2:8
# library(fpc)
# library(cluster)
pearsongammatward=NULL
altward=agnes(aineisto, metric="manhattan", method="ward", keep.diss=T)
for(i in 2:kmax){
  altwardi <- cutree(altward, i)
  validointitulos=cluster.stats(altward$diss, altwardi)
  pearsongammatward[i-1]=validointitulos$pearsongamma
}
list(modhubertingammat=pearsongammatward,k=2:kmax)
}

pearsongammatwardille=pearsongammatward(altruismi,kmax=8)
pearsongammatwardille

## $modhubertingammat
## [1] 0.1875577 0.1795481 0.2637372 0.2784247 0.2610203 0.2786957 0.2688791
##
## $k
## [1] 2 3 4 5 6 7 8
plot(2:8,pearsongammatwardille$modhubertingammat, xlab="ryhmien määrä",
      ylab="Norm. Hub. gamma")

```



```

# b) K-medoids
pearsongammatkmedoids=function(aineisto,kmax=8)

```

```

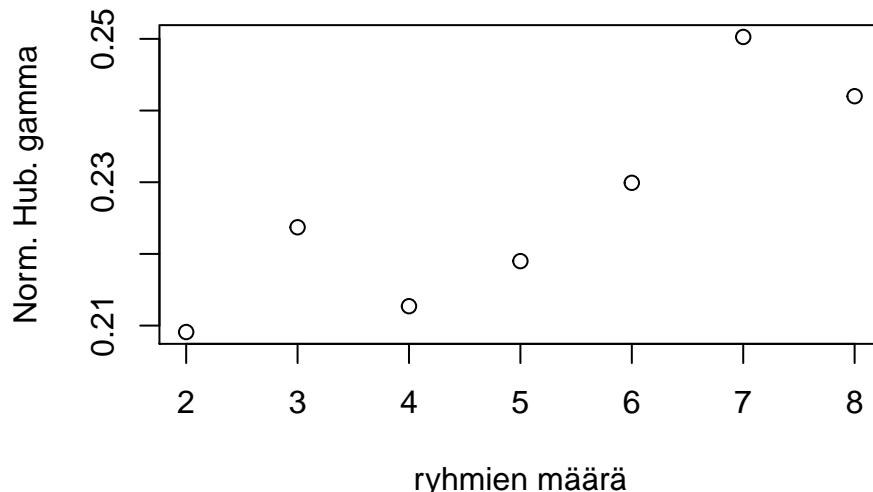
{
#library(fpc)
#library(cluster)
pearsongammatkmedoids=NULL
# Lasketaan laheisyysmatriisi:
d=dist(aineisto, method="manhattan") # euklidinen etaisyys oletus
# Lasketaan pearsongammatkmedoids
for(i in 2:kmax){
  klusttulos=pam(aineisto,i, metric="manhattan")
  validointitulos=cluster.stats(d, klusttulos$cluster)
  pearsongammatkmedoids[i-1]=validointitulos$pearsongamma
}
list(modhubertingammat=pearsongammatkmedoids, k=2:kmax)
}

pearsongammatkmedoids=pearsongammatkmedoids(altruismi, kmax=8)

## Warning in dist(aineisto, method = "manhattan"): NAs introduced by coercion
pearsongammatkmedoids

## $modhubertingammat
## [1] 0.2090967 0.2237140 0.2127023 0.2189931 0.2299176 0.2502690 0.2419999
##
## $k
## [1] 2 3 4 5 6 7 8
plot(2:8,pearsongammatkmedoids$modhubertingammat, xlab="ryhmien määrä",
      ylab="Norm. Hub. gamma")

```



```

# c) Kumpi on parempi?
# Wardin tapauksessa neljä ryhmää ja k-medoidsin kolme olisi hyvä määrä.
# Näistä ehkä mieluummin k-medoids, koska Wardin kohdalla ristiriitaisuuksia.

```

```

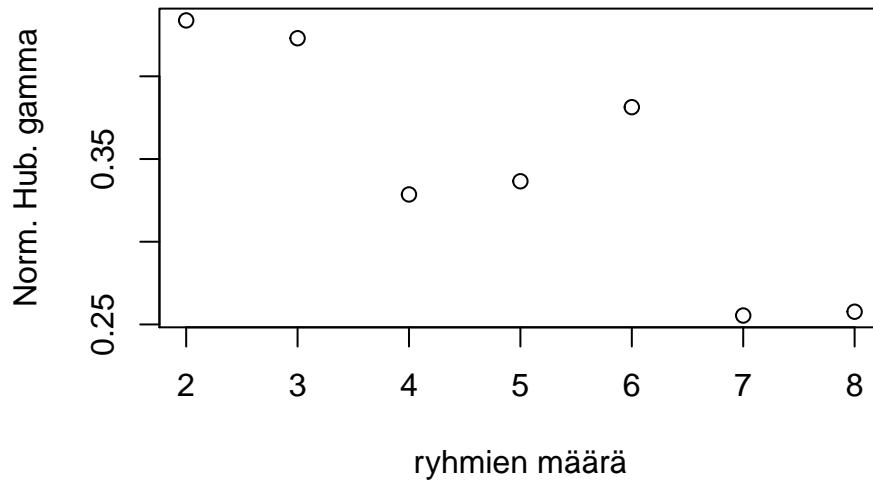
# Menetelmästä riippumatta kannattaa katsoa, onko ryhmien välillä
# eroja muuttujien arvoissa

# Extraa:
pearsongammatcomp=function(aineisto,kmax=8)
{# Lasketaan pearsongammat kaukaisimman, kun ryhmien määrä 2:8
# library(fpc)
# library(cluster)
pearsongammatcomp=NULL
alcomp=agnes(aineisto, metric="manhattan", method="complete", keep.diss=T)
for(i in 2:kmax){
  altcompi <- cutree(alcomp, i)
  validointitulos=cluster.stats(alcomp$diss, altcompi)
  pearsongammatcomp[i-1]=validointitulos$pearsongamma
}
list(modhubertingammat=pearsongammatcomp,k=2:kmax)
}

pearsongammatcompille=pearsongammatcomp(altruismi,kmax=8)
pearsongammatcompille

## $modhubertingammat
## [1] 0.4338026 0.4230224 0.3285836 0.3365586 0.3813582 0.2553909 0.2576776
##
## $k
## [1] 2 3 4 5 6 7 8
plot(2:8,pearsongammatcompille$modhubertingammat, xlab="ryhmien määrä",
      ylab="Norm. Hub. gamma")

```



#### 8. Paras menetelmä, kun 3 ryhmää

Tutki, mikä menetelmä (K-medoids- ja kaikki hierarkkiset menetelmät) olisi paras 3 ryhmän tapauksessa. Käytä kaikissa manhattan-etäisyyttä.

# hierarkkiset menetelmät

```

altave=agnes(altruismi, metric="manhattan", method="average", keep.diss=T)
altsingle=agnes(altruismi, metric="manhattan", method="single", keep.diss=T)
altcomp=agnes(altruismi, metric="manhattan", method="complete", keep.diss=T)
altward=agnes(altruismi, metric="manhattan", method="ward", keep.diss=T)
altdiana=diana(altruismi, metric="manhattan", keep.diss=T)

```

```
altave3=cutree(altave, 3)
```

altave3

```
tulos1=cluster.stats(altave$diss, altave3)
```

tulos1\$pearsongamma

```
## [1] 0.4013434
```

```
altsingle3=cutree(altsingle, 3)
```

- altsingle3

```
tulos2=cluster.stats(altsingle$diss, altsingle3)
```

tulos2\$pearsongamma

```
## [1] 0.4098327
```

```
altcomp3=cutree(altcomp, 3)
```

altcomp3

```
tulos3=cluster.stats(altcomp$diss, altcomp3)
```

tulos3\$pearsongamma

```
## [1] 0.4230224
```

```

altward3=cutree(altward, 3)
altward3

## [1] 1 2 2 1 2 3 1 1 1 1 3 1 1 1 1 2 2 2 2 1 1 1 1 3 1 1 1 1 1 1 1 1 1 2 2 2 1 2 2 2
## [36] 2 2 1 3 2 1 2 1 2 3 2 1 1 3 2 1 3 3 1 1 2 1 1 2 1 1 1 2 1 1 2 2 2 1 2 2 2 1 2 2
## [71] 1 2 2 2 1 1 3 1 1 1 1 2 1 1 1 1 1 1 3 2 3 2 2 2 2 1 3 2 3 2 1 2 3 3 1
## [106] 2 1 2 3 2 1 1 1 1 1 1 1 1 2 1 1 3 2 1 2 2 1 2 1 2 1 1 1 3 1 2 1 2 2 1 1
## [141] 1 3 2 3 1 1 1 2 2 2 2 2 1 1 2 1 2 3 1 1 2 2 1 2 1 1 1 3 1 1 2 1 2 2 1 1
## [176] 2 3 3 3 1 2 3 1 2 1 3 1 2 2 1 2 2

tulos4=cluster.stats(altward$diss, altward3)
tulos4$pearsongamma

## [1] 0.1795481

altdiana3=cutree(altdiana, 3)
altdiana3

## [1] 1 1 1 1 1 1 1 2 2 2 1 1 1 1 2 1 1 1 2 2 2 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1
## [36] 2 1 2 1 2 2 1 1 1 1 1 2 2 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1
## [71] 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1
## [106] 1 1 1 1 1 1 1 1 2 2 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1
## [141] 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [176] 1 1 1 1 1 1 1 1 2 1 2 1 1 2 1 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1

tulos5=cluster.stats(altdiana$diss, altdiana3)
tulos5$pearsongamma

## [1] 0.3550119

# k-medoids
altkmedoids3=pam(altruismi,3,metric="manhattan")
altkmedoids3$clustering

## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
## 1 2 3 3 1 1 3 1 3 1 2 3 3 3 3 3 2 3 2
## 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
## 3 3 1 3 3 2 3 3 3 2 3 3 1 3 3 2 3 1
## 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
## 1 3 1 3 3 2 3 3 2 1 1 3 2 1 3 2 1 3
## 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
## 3 1 1 3 1 1 3 1 3 3 1 3 1 3 3 2 2 3
## 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
## 3 2 1 3 2 3 3 2 2 3 3 1 3 1 3 3 2 2
## 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108
## 2 2 2 3 2 1 3 2 2 2 3 3 2 1 1 2 1 1
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
## 1 1 1 1 3 3 3 3 2 2 3 3 2 2 3 3 1 3
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
## 1 3 3 3 3 3 2 1 3 3 1 3 3 3 3 2 2 2
## 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162
## 3 3 3 2 2 1 1 2 2 1 3 2 1 3 2 2 3 3
## 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
## 1 2 3 2 3 3 1 2 3 1 3 3 2 2 1 2 2 3
## 181 182 183 184 185 186 187 188 189 190 191 192
## 2 2 1 2 3 2 3 3 2 3 2 2 1

```

```

pearsongamma kmedoids$modhubert ingamma [2]
## [1] 0.223714
# näistä ehkä k-medoids (eri periaate) ja kaukaisin, kaukaisimman kohdalla
# riittäisi kaksi ryhmää
# - myös valitulla menetelmällä saatujen ryhmien välillä olisi oltava eroja,
# jotta ryhmät ovat tulkittavissa

```

## 9. Geenidata ja hierarkkiset menetelmät

Analysoi geenidata uudelleen käyttäen hierarkisia menetelmiä ja euklidista etäisyyttä. Tutki, mikä menetelmä on paras (dendogrammien perusteella ja muiden tunnuslukujen perusteella), montako ryhmää tarvitaan ja vertaa ryhmittelyä syöpätyypeihin.

```

# Luetaan aineistot
nci.data <- read.table(url("https://web.stanford.edu/~hastie/ElemStatLearn/datasets/nci.data.csv"),
  sep=",", row.names=1, header=TRUE)
nci.label <- scan(url("https://web.stanford.edu/~hastie/ElemStatLearn/datasets/nci.label.txt"), what="ch"

# Ryhmittely
tnci.data<-t(nci.data)

nciave=agnes(tnci.data, metric="euclidean", method="average", keep.diss=T)
ncisingle=agnes(tnci.data, metric="euclidean", method="single", keep.diss=T)
ncicomp=agnes(tnci.data, metric="euclidean", method="complete", keep.diss=T)
nciward=agnes(tnci.data, metric="euclidean", method="ward", keep.diss=T)
ncidiana=diana(tnci.data, metric="euclidean", keep.diss=T)

# Kofeneettiset korrelaatiokertoimet
sqrt(1-cl_dissimilarity(nciave, nciave$diss, method="cophenetic"))

## Cross-dissimilarities using cophenetic correlations:
##      [,1]
## [1,] 0.7690221
sqrt(1-cl_dissimilarity(ncisingle, ncisingle$diss, method="cophenetic"))

## Cross-dissimilarities using cophenetic correlations:
##      [,1]
## [1,] 0.6829895
sqrt(1-cl_dissimilarity(ncicomp, ncicomp$diss, method="cophenetic"))

## Cross-dissimilarities using cophenetic correlations:
##      [,1]
## [1,] 0.6584
sqrt(1-cl_dissimilarity(nciward, ncisingle$diss, method="cophenetic"))

## Cross-dissimilarities using cophenetic correlations:
##      [,1]
## [1,] 0.5391595
sqrt(1-cl_dissimilarity(ncidiana, ncisingle$diss, method="cophenetic"))

## Cross-dissimilarities using cophenetic correlations:

```

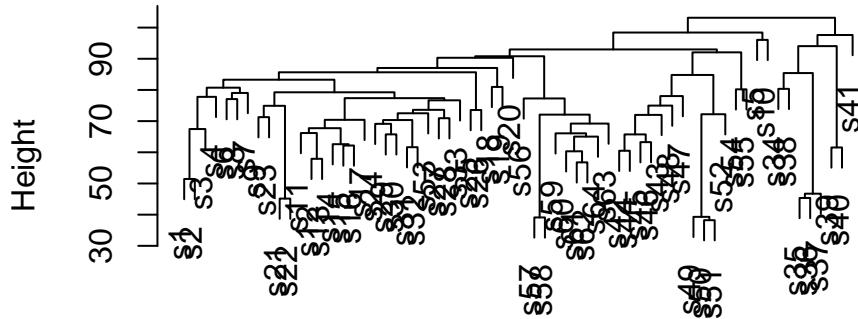
```

##          [,1]
## [1,] 0.6657222
# aika samoja

plot(nciave, which.plots=2)

```

**rogram of agnes(x = tnci.data, metric = "euclidean", r  
"average", keep.diss = T)**



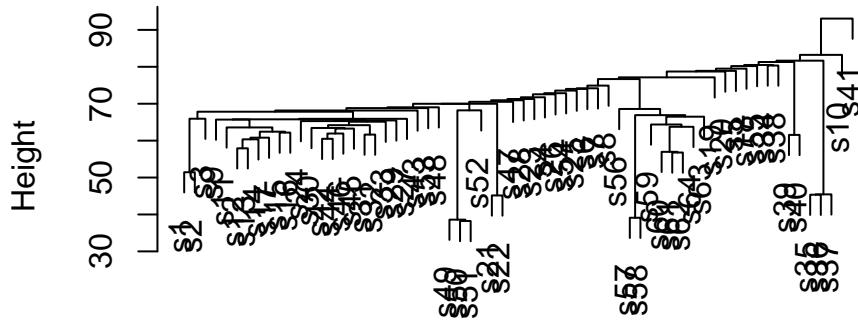
tnci.data  
Agglomerative Coefficient = 0.36

```

plot(ncisingle, which.plots=2)

```

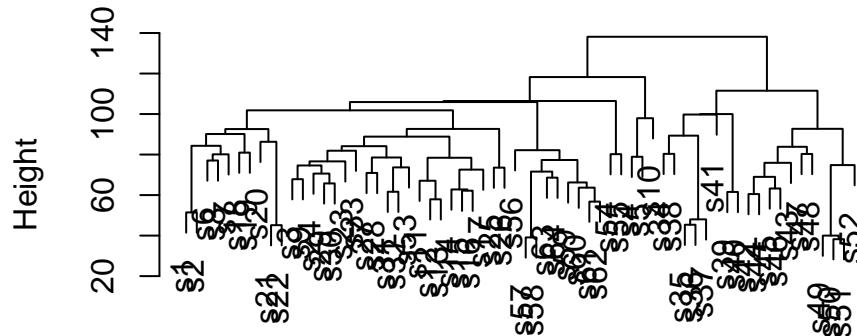
**rogram of agnes(x = tnci.data, metric = "euclidean", r  
"single", keep.diss = T)**



tnci.data  
Agglomerative Coefficient = 0.32

```
plot(ncicomp, which.plots=2)
```

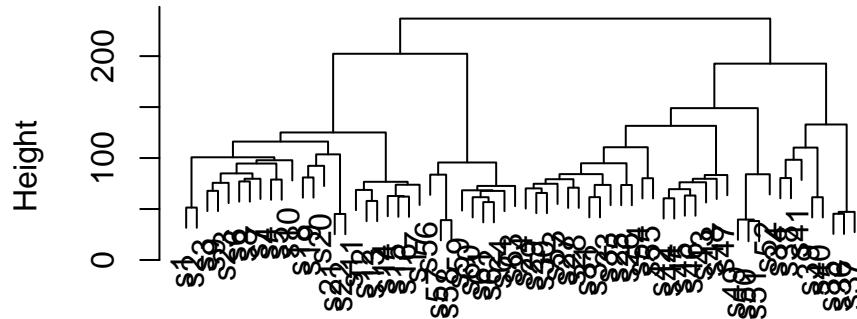
rogram of agnes(x = tnci.data, metric = "euclidean", r  
"complete", keep.diss = T)



tnci.data  
Agglomerative Coefficient = 0.52

```
plot(nciward, which.plots=2)
```

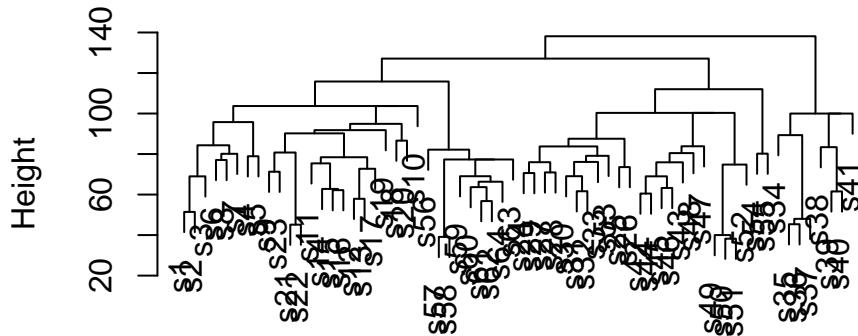
ram of agnes(x = tnci.data, metric = "euclidean", meth  
keep.diss = T)



tnci.data  
Agglomerative Coefficient = 0.72

```
plot(ncidiana, which.plots=2)
```

ogram of `diana(x = tnci.data, metric = "euclidean", keep.diss = T)`



tnci.data  
Divisive Coefficient = 0.51

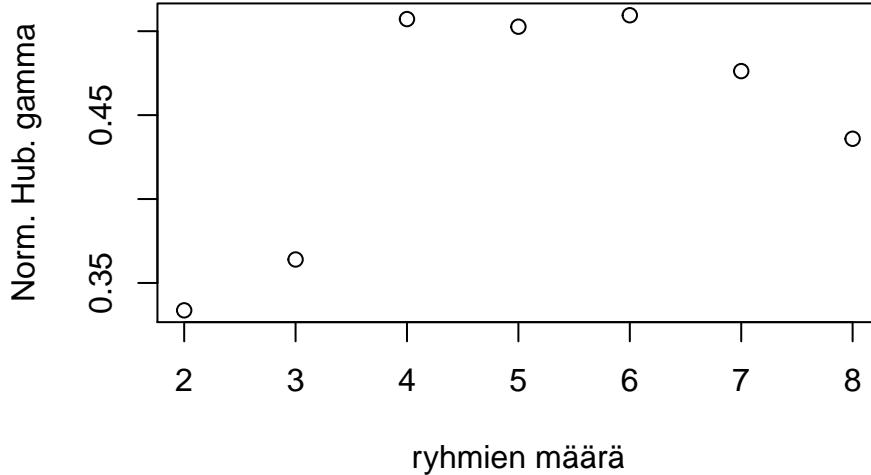
```

pearsongammatward=function(aineisto,kmax=8)
{# Lasketaan pearsongammaat Wardin menetelmälle, kun ryhmien määrä 2:8
# library(fpc)
# library(cluster)
pearsongammatward=NULL
altward=agnes(aineisto, metric="euclidean", method="ward", keep.diss=T)
for(i in 2:kmax){
  altwardi <- cutree(altward, i)
  validointitulos=cluster.stats(altward$diss, altwardi)
  pearsongammatward[i-1]=validointitulos$pearsongamma
}
list(modhubertingammaat=pearsongammatward,k=2:kmax)
}

pearsongammatwardille=pearsongammatward(tnci.data,kmax=8)
pearsongammatwardille

## $modhubertingammaat
## [1] 0.3336721 0.3639707 0.5072123 0.5026825 0.5095042 0.4762156 0.4358970
## 
## $k
## [1] 2 3 4 5 6 7 8
plot(2:8,pearsongammatwardille$modhubertingammaat, xlab="ryhmien määrä",
      ylab="Norm. Hub. gamma")

```



```

nciward4=cutree(nciward, 4)
nciward4

## [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
## [36] 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4

tulos4=cluster.stats(nciward$diss, nciward4)
tulos4$pearsongamma

## [1] 0.5072123

# Ryhmien ja syöpätyyppien vertailu
ncigeenienryhmatjasyopatyypit<-data.frame(nciward4,factor(nci.label))
taulu<-table(ncigeenienryhmatjasyopatyypit[,1],ncigeenienryhmatjasyopatyypit[,2])
taulu

##
##          BREAST CNS COLON K562A-repro K562B-repro LEUKEMIA MCF7A-repro
## 1          3    5     0          0          0          0          0
## 2          2    0     7          0          0          0          1
## 3          0    0     0          1          1          6          0
## 4          2    0     0          0          0          0          0
##
##          MCF7D-repro MELANOMA NSCLC OVARIAN PROSTATE RENAL UNKNOWN
## 1          0         1     3     1     0     9     1
## 2          1         0     6     5     2     0     0
## 3          0         0     0     0     0     0     0
## 4          0         7     0     0     0     0     0

pearsongammataave=function(aineisto,kmax=8)
{# Lasketaan pearsongammat Wardin menetelmälle, kun ryhmien määrä 2:8
  # library(fpc)
  # library(cluster)
  pearsongammataave=NULL
  altave=agnes(aineisto, metric="euclidean", method="single", keep.diss=T)
}

```

```

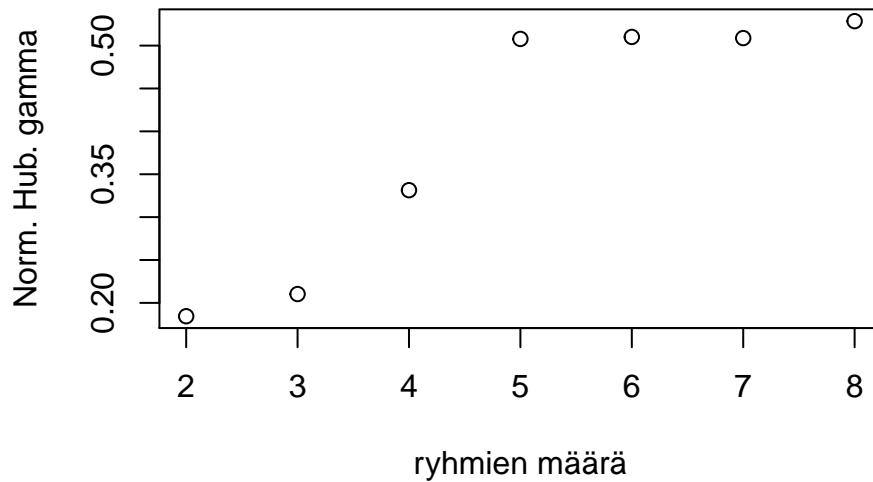
for(i in 2:kmax){
  altavei <- cutree(altave, i)
  validointitulos=cluster.stats(altave$diss, altavei)
  pearsongammatave[i-1]=validointitulos$pearsongamma
}
list(modhubertingammat=pearsongammatave,k=2:kmax)
}

pearsongammatavelle=pearsongammatave(tnci.data,kmax=8)
pearsongammatavelle

## $modhubertingammat
## [1] 0.1844202 0.2103134 0.3314155 0.5077862 0.5100962 0.5087790 0.5285272
##
## $k
## [1] 2 3 4 5 6 7 8

plot(2:8,pearsongammatavelle$modhubertingammat, xlab="ryhmien määrä",
      ylab="Norm. Hub. gamma")

```



taulu

```
##  
##      BREAST  CNS  COLON K562A-repro K562B-repro LEUKEMIA MCF7A-repro  
##  1      6    5     7          0          0          0          1  
##  2      1    0     0          0          0          0          0  
##  3      0    0     0          1          1          5          0  
##  4      0    0     0          0          0          1          0  
##  
##      MCF7D-repro MELANOMA NSCLC OVARIAN PROSTATE RENAL UNKNOWN  
##  1          1      8     8     6     2     9     1  
##  2          0      0     1     0     0     0     0  
##  3          0      0     0     0     0     0     0  
##  4          0      0     0     0     0     0     0
```

# TILS646

DEMO 3, 5.2.2019, klo 8:30.

Palauta R-tehtävien ratkaisut (R-koodit ja tulokset tulkintoineen) Koppaan (tehtävänpalautus) viimeistään demoja edeltävänä maanantaina klo 18:00 mennessä. Nimeä tiedosto etunimisukunimi\_demo3.R

Vain palautettuina tehtävistä saa demohyvityksiä. Jos ei pääse demoryhmään, demopisteistä saa vain puolet. Valmistaudu esittämään ratkaisusi demoryhmässä.

## 1. Aineiston luku

Lue pohjaeläinaineisto R:ään.

```
bugs<-read.table("bugs.dat",header=TRUE) # datan luku
#head(bugs)
names(bugs)

## [1] "Species"   "Area"      "Mean"       "StdDev"     "Mode"       "Perim"      "Width"
## [8] "Height"     "Major"     "Minor"      "Angle"      "Circ"      "Feret"      "IntDen"
## [15] "Median"    "Skew"      "Kurt"

#str(bugs)
```

## 2. Poimitaan aineistosta kaksi lajia ja kaksi muuttujaa

Poimi samat lajit kuin R3-harjoituksissa ja muuttujista (Area:n) lisäksi Height.

```
index1<-which(bugs$Species=="Baetisrhodani")
index2<-which( bugs$Species=="Hydropsychepellucidulla")
osadata<-bugs[c(index1,index2),]

# poistetaan ylimääräiset luokat
osadata$Species<-factor(osadata$Species, labels=c("Baetis","Hydro"))

# poimitaan Species, Area ja Height muuttujat
osadata<-osadata[,c("Species", "Area", "Height")]
names(osadata)

## [1] "Species"   "Area"      "Height"
```

	Species	Area	Height
1	Baetis	30813	503
2	Baetis	30600	317
3	Baetis	30551	427
4	Baetis	29713	547
5	Baetis	28546	383
6	Baetis	21492	359

## 3. Jako opetus- ja testiaineistoon

Jaa aineisto opetus- ja testiaineistoon, kuten R3-harjoituksissa. Käytä siemenlukua 123.

```

set.seed(123)
M<-ncol(osadata)
N<-nrow(osadata)
train.rows<-sample(1:N, 0.5*N)
osa.train<-osadata[train.rows,]
osa.test<-osadata[-train.rows,]

```

## 4. Tutkitaan opetusaineistoa, miten hyvin luokat ovat erillään

Laske tunnuslukuja ja määriä opetusaineistosta sekä piirrä hajontakuvio ryhmittäin samaan kuvaajaan.

```
tapply(osa.train$Area, osa.train$Species,summary)
```

```

## $Baetis
##   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
##   1341    11989   14646   14839   17052   30813
##
## $Hydro
##   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
##   5959   130431  193935  191231  266037  368784
tapply(osa.train$Height, osa.train$Species,summary)
```

```

## $Baetis
##   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
##   67     182     229     257     314     503
##
## $Hydro
##   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
##   149.0   511.0   729.0   675.4   843.5   1077.0
tapply(osa.train$Area, osa.train$Species, sd)
```

```

##   Baetis      Hydro
## 4547.461 93250.709
tapply(osa.train$Height, osa.train$Species, sd)
```

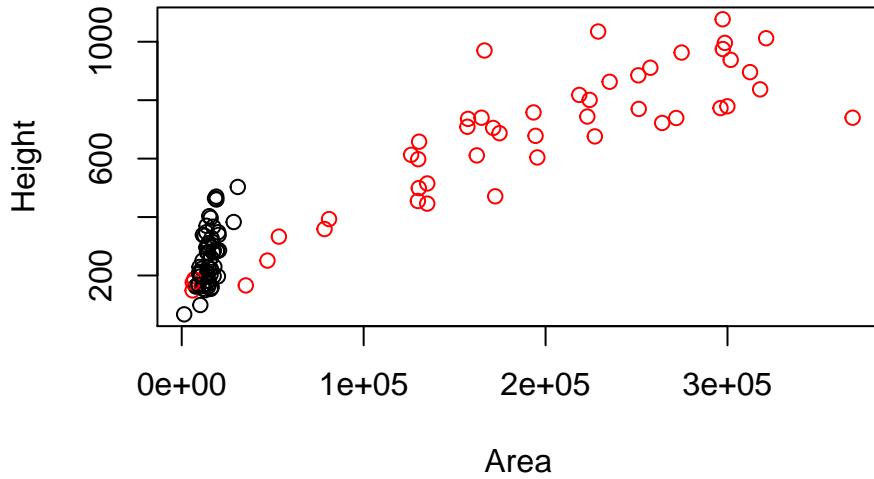
```

##   Baetis      Hydro
## 95.13884 246.78921

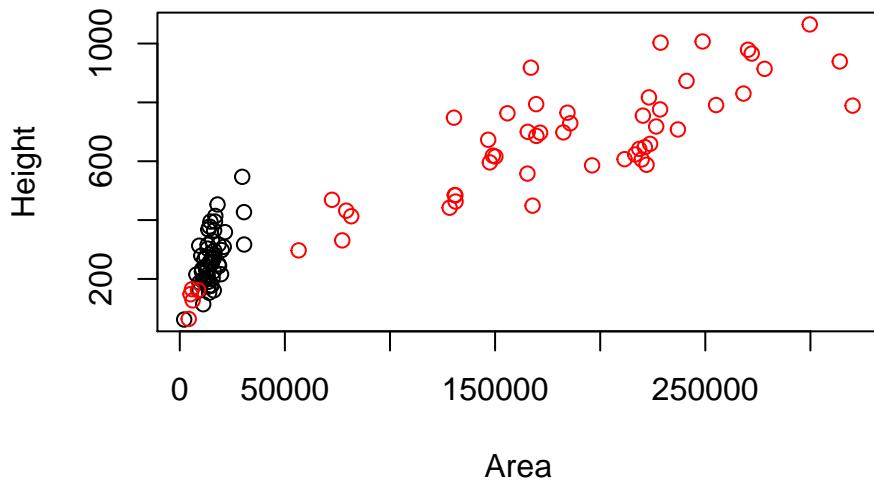
# talletetaan seuraavat tunnusluvut itse tehtäviä
# luokitteluja varten
baetis<-osa.train[osa.train$Species=="Baetis",]
covb<-cov(baetis[,c("Area","Height")])
hydro<-osa.train[osa.train$Species=="Hydro",]
covh<-cov(hydro[,c("Area","Height")])

mb<-c(mean(baetis[,c("Area")]),mean(baetis[, "Height"]))
nb<-nrow(baetis)
mh<-c(mean(hydro[,c("Area")]),mean(hydro[, "Height"]))
nh<-nrow(hydro)

# piirretään kuvaavia, ensin hajontakuvio opetusdatalle
plot(osa.train[,c("Area","Height")], col=osa.train$Species)
```



```
# ei kysytty, mutta piirretään silti vielä hajontakuvio testidatalle
plot(osa.test[,c("Area", "Height")], col=osa.test$Species)
```



## 5. Luokitellaan testidata

Luokittele testidata. Vaihtoehdot ovat

- a) euklidisen etäisyyden avulla
- b) Mahalanobis-etäisyyden avulla

- c) kaksiulotteisen normaalijakauman tiheysfunktion avulla olettaen yhtäsuuret priorit
- d) tiheysfunktion avulla, kuten c-kohdassa, mutta lisätään opetusdataasta estimoidut priorit (edellä olettettu yhtä suuriksi)
- e) tiheysfunktion avulla, kuten c-kohdassa, mutta kovarianssit oletetaan nolliksi (Naive Bayes)

```
# lisätään dataan
# 1) numeerinen laji-muuttuja Speciesint oikeaksi luokaksi
# 2) numeerinen laji-muuttuja Speciesluok luokitelluksi luokaksi
osa.test$Speciesint<-as.numeric(osa.test$Species)
ntest<-length(osa.test[,1])
osa.test$Speciesluok<-rep(0,ntest)
test.features<-osa.test[,c("Area","Height")]

# a) euklidisen etäisyyden avulla
# käytetään dist-funktioita

for (i in 1:ntest)
{
  osa.test$Speciesluok[i]<-2
  datab<-data.frame(rbind(test.features[i,],mb))
  datah<-data.frame(rbind(test.features[i,],mh))
  if (dist(datab)<=dist(datah))
    osa.test$Speciesluok[i]<-1
}
# cbind(osa.test$Area,osa.test$Speciesint,osa.test$Speciesluok)
table(osa.test$Speciesint,osa.test$Speciesluok)

##
##      1  2
##  1 55  0
##  2 10 44

# b) Mahalanobis-etaisyyden avulla
# käytetään omaa funktioita, olemassa myös mahalanobis-funktio
for (i in 1:ntest)
{
  osa.test$Speciesluok[i]<-2
  datab<-data.frame(rbind(test.features[i,],mb))
  datah<-data.frame(rbind(test.features[i,],mh))
  mahab<-as.matrix(datab[1,]-datab[2,])%*%solve(covb)%*%t(as.matrix(datab[1,]-datab[2,]))
  mahah<-as.matrix(datah[1,]-datah[2,])%*%solve(covh)%*%t(as.matrix(datah[1,]-datah[2,]))
  if (mahab<=mahah)
    osa.test$Speciesluok[i]<-1
}
#cbind(osa.test$Area,osa.test$Speciesint,osa.test$Speciesluok)
table(osa.test$Speciesint,osa.test$Speciesluok)

##
##      1  2
##  1 50  5
##  2  4 50

# c) kaksiulotteisen normaalijakauman tiheysfunktion avulla olettaen yhtäsuuret priorit
library(mvtnorm)
for (i in 1:ntest)
```

```

{
  osa.test$Speciesluok[i]<-2
  if (dmvnorm(test.features[i,],mb,covb)>=dmvnorm(test.features[i,],mh,covh))
    osa.test$Speciesluok[i]<-1
}
# cbind(osa.test$Area,osa.test$Speciesint,osa.test$Speciesluok)
table(osa.test$Speciesint,osa.test$Speciesluok)

## 
##      1  2
##  1 53  2
##  2  5 49

# d) tiheysfunktion avulla, kuten c-kohdassa, mutta lisätään opetusdatasta estimoidut
# priorit (edellä oletettu yhtä suuriksi)

pibaetis<-nb/(nb+nh)
pihydro<-1-pibaetis
for (i in 1:ntest)
{
  osa.test$Speciesluok[i]<-2
  if (pibaetis*dmvnorm(test.features[i,],mb,covb)>=pihydro*dmvnorm(test.features[i,],mh,covh))
    osa.test$Speciesluok[i]<-1
}
# cbind(osa.test$Area,osa.test$Speciesint,osa.test$Speciesluok)
table(osa.test$Speciesint,osa.test$Speciesluok)

## 
##      1  2
##  1 53  2
##  2  5 49

# e) tiheysfunktion avulla, kuten c-kohdassa, mutta kovarianssit oletetaan nolliksi
# (Naive Bayes)

# muokataan kovarianssimatriiseja siten, että ei-diagonaalit nollia
nbcovb<-covb
nbcovh<-covh
nbcovb[1,2]<-0
nbcovb[2,1]<-0
nbcovh[1,2]<-0
nbcovh[2,1]<-0

library(mvtnorm) # kirjasto, jossa moniulotteisen normaalijakauman funktioita
for (i in 1:ntest)
{
  osa.test$Speciesluok[i]<-2
  if (dmvnorm(test.features[i,],mb,nbcovb)>=dmvnorm(test.features[i,],mh,nbcovh))
    osa.test$Speciesluok[i]<-1
}
# cbind(osa.test$Area,osa.test$Speciesint,osa.test$Speciesluok)
table(osa.test$Speciesint,osa.test$Speciesluok)

## 
##      1  2

```

```
##   1 53  2
##   2  5 49
```

## 6. lda-funktio

- a) Tutustu R-helppiin avulla lda-funktioon.
- b) Toteuta testidatian luokittelu lda-funktioilla tasapriorilla ja aineistosta estimoiduilla prioreilla vertaa tulosta itse laskettuihin tuloksiin.

```
library(MASS)
# a)
# ?lda
# ?predict.lda
# Value: a list with components
# class:      The MAP classification (a factor)
# posterior: posterior probabilities for the classes
# x: the scores of test cases on up to dimen discriminant variables

# datan palautus alkuperäiseen muotoon
osa.train<-rbind(baetis,hydro)
osa.train$Species<-factor(osa.train$Species, labels=c("Baetis","Hydro"))
osa.test<-osadata[-train.rows,]
osa.test$Species<-factor(osa.test$Species, labels=c("Baetis","Hydro"))

# LDA, tasapriorit
LDA<-lda(Species ~ .,data=osa.train, prior=c(0.50,0.50))
LDA

## Call:
## lda(Species ~ ., data = osa.train, prior = c(0.5, 0.5))
##
## Prior probabilities of groups:
## Baetis  Hydro
##     0.5     0.5
##
## Group means:
##           Area    Height
## Baetis  14839.34 257.0328
## Hydro   191230.60 675.3542
##
## Coefficients of linear discriminants:
##             LD1
## Area  1.575638e-05
## Height 1.695434e-04

#plot(LDA) # tulostaa kanonisen erottelufunktion (tai -funktioiden)
# arvot luokittain
LDA.pred<-predict(LDA,newdata=osa.test)
table(osa.test$Species,LDA.pred$class)

##
##           Baetis Hydro
## Baetis      55     0
## Hydro       10    44
```

```

# LDA, estimoidut priorit oletuksena
LDA<-lda(Species ~ .,data=osa.train)
LDA

## Call:
## lda(Species ~ ., data = osa.train)
##
## Prior probabilities of groups:
##   Baetis    Hydro
## 0.559633 0.440367
##
## Group means:
##           Area   Height
## Baetis 14839.34 257.0328
## Hydro  191230.60 675.3542
##
## Coefficients of linear discriminants:
##           LD1
## Area  1.575638e-05
## Height 1.695434e-04

#plot(LDA)
LDA.pred<-predict(LDA,newdata=osa.test)
table(osa.test$Species,LDA.pred$class)

##
##           Baetis Hydro
## Baetis      55     0
## Hydro       10     44

# EXTRAA: toistetaan lda-funktion analyysia
# - tehdään uusi muuttuja d, jota kutsutaan kanoniseksi erottelufunktioksi
#   ja se on lineaarikombinaatio keskistetyistä alkuperäisistä muuttujista
# - erottelufunktioiden määrä on min(p,k-1), jossa p piirteiden määrä ja
#   k luokkien määrä
# - tässä tilanteessa vai yksi uusi muuttuja d, koska kaksi luokkaa
# - luokittelun tehdään muuttujan d avulla käyttäen euklidista etäisyyttä
# - R laskee posterioritodennäköisyydet
# Alla oleva koodi on peräisin monimuuttujamenetelmät-kurssilta (TIM/mmm/mmm5),
# jossa myös tietoa, miten lineaarikombinaation kertoimet teoriassa ja
# käytännössä lasketaan

# alkuperäiset muuttujat keskistetään
scaled.train<-scale(osa.train[,-1], center=TRUE, scale=FALSE)
mtrain<-colMeans(osa.train[,2:3])
# erottelufunktion laskemiseen tarvittavat kertoimet
raaka<-LDA$scaling
raaka

##
##           LD1
## Area  1.575638e-05
## Height 1.695434e-04

# kanonisen erottelufunktion arvot tallitetaan
d.train<-t(t(raaka) %*% t(scaled.train))
osa.train$d<-d.train

```

```

# lasketaan samat testidataalle siten, että laskenta perustuu
# opetusdataasta saatuihin tunnuslukuihin
scaled.test<-osa.test[,-1]
scaled.test$Area<-scaled.test$Area-mtrain[1]
scaled.test$Height<-scaled.test$Height-mtrain[2]
d.test<-t(t(raaka)%%t(scaled.test))
osa.test$d<-d.test

# verrataan R:n antamiin arvoihin
head(cbind(d.test,LDA.pred$x)) # saatiin samat kuin R

##          LD1         LD2
## 2 -0.9966412 -0.9966412
## 3 -0.9787635 -0.9787635
## 4 -0.9716222 -0.9716222
## 6 -1.1330296 -1.1330296
## 7 -1.1504564 -1.1504564
## 11 -1.1678256 -1.1678256

# toteutetaan luokittelu euklidisen etäisyyden avulla
# käyttääen muuttuja d

baetis<-osa.train[osa.train$Species=="Baetis",]
mbd<-mean(baetis$d)
hydro<-osa.train[osa.train$Species=="Hydro",]
mhd<-mean(hydro$d)

osa.test$Speciesint<-as.numeric(osa.test$Species)
ntest<-length(osa.test[,1])
osa.test$Speciesluok<-rep(0,ntest)

for (i in 1:ntest)
{
  osa.test$Speciesluok[i]<-2
  if (abs(osa.test$d[i]-mbd)<=abs(osa.test$d[i]-mhd))
    osa.test$Speciesluok[i]<-1
}
# cbind(osa.test$Area,osa.test$Speciesint,osa.test$Speciesluok)
table(osa.test$Speciesint,osa.test$Speciesluok)

##
##          1   2
## 1 55   0
## 2 10  44

# ainakin ristiintaulukot ovat samoja kuin 4a) ja 6. tehtävässä

```

## 7. qda-funktio

- Tutustu R-helppiin avulla qda-funktioon.
- Toteuta testidataan luokittelu qda-funktiolla tasapriorilla ja aineistosta estimoiduilla prioreilla vertaa tulosta itse laskettuihin tuloksiin.

```

# R Markdown antaa qda:han liittyvän virheilmoituksen, joka jäi
# ratkaisematta, tulokset saa kuitenkin laskettua Console-ikkunassa
# ?qda
# library(MASS)
# QDA
# QDA<-qda(Species~, data=osa.train, prior=c(0.50,0.50))
# QDA.pred<-predict(QDA, newdata=osa.test)
# table(osa.test$Species, QDA.pred$class)
#
#> table(osa.test$Species, QDA.pred$class)
#
#           Baetis Hydro
# Baetis      53     2
# Hydro        5    49
#

```

## 8. Suosiva tappiofunktio

Tutkitaan, miten luokittelua muuttuu, kun käytetään seuraavia tappiofunktion arvoja  $L(\text{ennustettu Baetis}|\text{Hydro})=1$  ja  $L(\text{ennustettu Hydro}|\text{Baetis})=2$ . Hyödynnä qda-funktion objektia posterior ja laske luokittelua uudelleen.

```

osa.test$Speciesint<-as.numeric(osa.test$Species)
ntest<-length(osa.test[,1])
osa.test$Speciesluok<-rep(0, ntest)

tappiomat=matrix(c(0,1,2,0), ncol=2)
tappiomat

##      [,1] [,2]
## [1,]     0     2
## [2,]     1     0
#head(QDA.pred$posterior)

#QDAatappio<- QDA.pred$posterior%*%tappiomat
#head(QDAatappio)

#for (i in 1:ntest)
#{ 
#  osa.test$Speciesluok[i]<-2
#  if (QDAatappio[i,1]<=QDAatappio[i,2])
#    osa.test$Speciesluok[i]<-1
#}

#table(osa.test$Speciesint, osa.test$Speciesluok)
#> table(osa.test$Speciesint, osa.test$Speciesluok)
#
#      1  2
# 1 54  1
# 2  5 49

```

# TILS646

DEMO 4, 12.2.2019, klo 8:30.

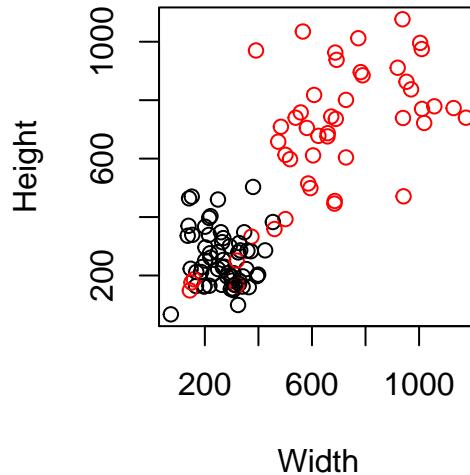
Palauta R-tehtävien ratkaisut (R-koodit ja tulokset tulkintoineen) Koppaan (tehtävänpalautus) viimeistään demoja edeltävänä maanantaina klo 20:00 mennessä. Nimeä tiedosto etunimisukunimi\_demo4.R

Vain palautettuina tehtävistä saa demohyvityksiä. Jos ei pääse demoryhmään, demopisteistä saa vain puolet. Valmistaudu esittämään ratkaisusi demoryhmässä.

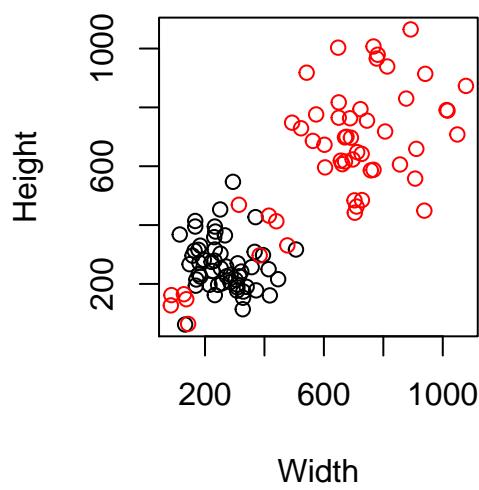
## 1. Aineiston luku ja muokkaus

Lue pohjaeläinaineisto R:ään. Poimi aineistosta lajit "Baetis" ja "Hydro" ja muuttujat "Width" ja "Height". Jaa aineisto kahtia opetus- ja testiaineistoon käyttäen siemenlukua 123. Laske tunnusluvut luokittain ja piirrä hajontakuvio opetusdataalle ja testidataalle siten, että sen akselit ovat yhtä pitkiä ja kuva on neliö.

```
bugs<-read.table("bugs.dat",header=TRUE) # datan luku
# poimitaan lajit ja muuttujat
index1<-which(bugs$Species=="Baetisrhodani")
index2<-which( bugs$Species=="Hydropsychepellucidulla")
osadata<-bugs[c(index1,index2),]
osadata$Species<-factor(osadata$Species, labels=c("Baetis","Hydro"))
osadata<-osadata[,c("Species", "Width", "Height")]
# jako opetus- ja testiaineistoon
set.seed(123)
M<-ncol(osadata)
N<-nrow(osadata)
train.rows<-sample(1:N,0.5*N)
osa.train<-osadata[train.rows,]
osa.test<-osadata[-train.rows,]
# piirretään kuvaa jaa:
# hajontakuvio opetusdataalle
par(pty="s")
plot(osa.train[,c("Width","Height")], col=osa.train$Species)
```



```
# hajontakuvio testidatalle
plot(osa.test[,c("Width","Height")], col=osa.test$Species)
```



```
# lasketaan tunnuslukuja
baetis<-osa.train[osa.train$Species=="Baetis",]
hydro<-osa.train[osa.train$Species=="Hydro",]
mb<-c(mean(baetis[,c("Width")]),mean(baetis[, "Height"]))
nb<-nrow(baetis)
mh<-c(mean(hydro[,c("Width")]),mean(hydro[, "Height"]))
nh<-nrow(hydro)
mb;mh
```

```

## [1] 267.0984 257.0328
## [1] 673.8958 675.3542
nb;nh

## [1] 61
## [1] 48
covb<-cov(baetis[,c("Width","Height")])
covh<-cov(hydro[,c("Width","Height")])
covb;covh

##           Width   Height
## Width  6553.5235 -710.0199
## Height -710.0199  9051.3989

##           Width   Height
## Width  65217.03  41021.23
## Height 41021.23  60904.91

```

## 2. Päätösalueiden piirto

Piirrä päätösalueet seuraavilla luokittelusäännöillä

- a) euklidinen etäisyys
- b) mahalanobis-etaisyys, kun kovarianssimatriisina on poolattu kovarianssimatriisi
- c) mahalanonis-etaisyys, kun kovarianssimatriisit on laskettu lajeittain

```

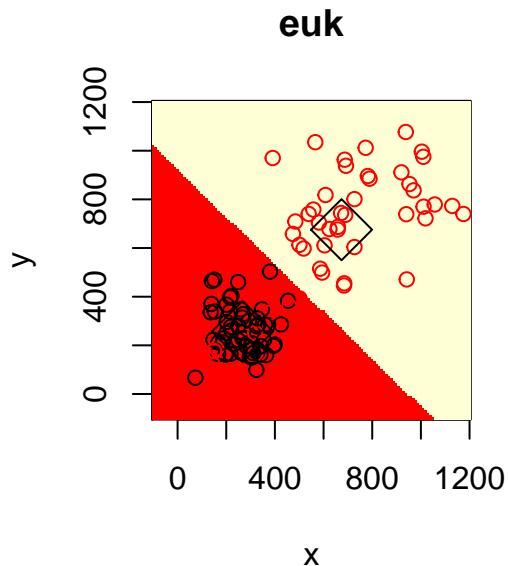
x<-seq(-100,1200,by=10)
y<-seq(-100,1200,by=10)
test.features <- mat.or.vec(length(x)*length(y), 2)
test.features<-data.frame("Width"=test.features[,1], "Height"=test.features[,2])
ntest<-length(test.features[,1])
ntest

## [1] 17161

for(j in 1:length(y)){
  for(i in 1:length(x)){
    test.features[i + (j-1)*length(x),1] = x[i]
    test.features[i + (j-1)*length(x),2] = y[j]
  }
}
# a) euklidinen etäisyys
luokate<-rep(2,length=ntest)
for (i in 1:ntest)
{
  datab<-data.frame(rbind(test.features[i,],mb))
  datah<-data.frame(rbind(test.features[i,],mh))
  if (dist(datab)<=dist(datah))
    luokate[i]<-1
}
z <- matrix(luokate, ncol=length(y))
par(pty="s")
image(x, y, z, main="euk")

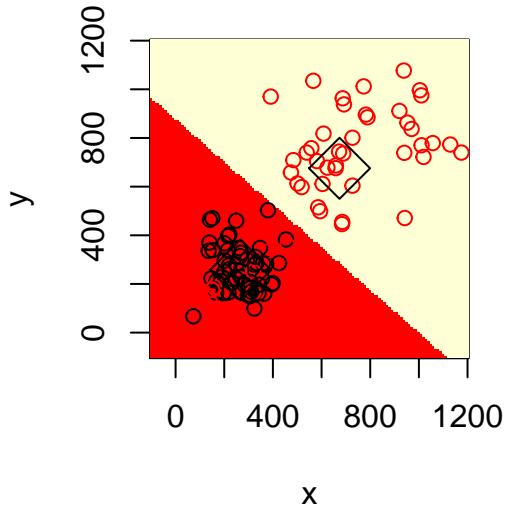
```

```
points(osa.train[,c("Width","Height")], col=osa.train$Species)
points(data.frame(rbind(mb,mh)),cex=3,pch=5)
```



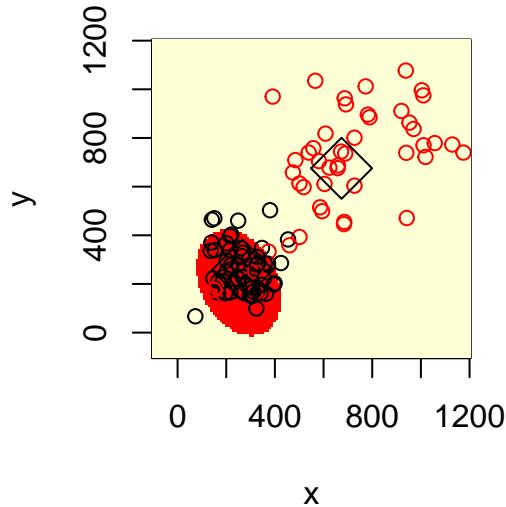
```
# b) mahalanobis-etaisyys, kun kovarianssimatriisina poolattu kovarianssimatriisi
N<-dim(osa.train)[1]
covp<-((nb-1)*covb+(nh-1)*covh)/(N-2)
luokatmp<-rep(2,length=ntest)
for (i in 1:ntest)
{
  db<-mahalanobis(test.features[i],mb,covp)
  dh<-mahalanobis(test.features[i],mh,covp)
  if (db<=dh)
    luokatmp[i]<-1
}
z <- matrix(luokatmp, ncol=length(y))
par(pty="s")
image(x, y, z, main="mahalanobis, kun poolattu Sigma")
points(osa.train[,c("Width","Height")], col=osa.train$Species)
points(data.frame(rbind(mb,mh)),cex=3,pch=5)
```

## mahanalobis, kun poolattu Sigma



```
# c) mahalanobis-etäisyys, kun kovarianssimatriiisit erit
luokatm<-rep(2,length=ntest)
for (i in 1:ntest)
{
  db<-mahalanobis(test.features[i,],mb,covb)
  dh<-mahalanobis(test.features[i,],mh,covh)
  if (db<=dh)
    luokatm[i]<-1
}
z <- matrix(luokatm, ncol=length(y))
par(pty="s")
image(x, y, z, main="mahanalobis, kun eri Sigmat")
points(osa.train[,c("Width","Height")], col=osa.train$Species)
points(data.frame(rbind(mb,mh)),cex=3,pch=5)
```

## mahanalobis, kun eri Sigmat



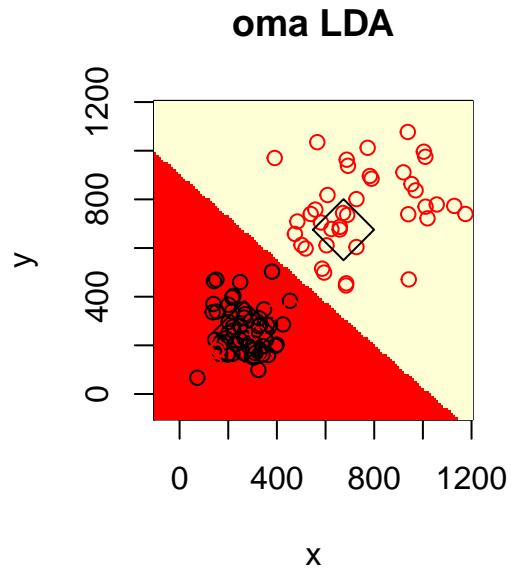
### 3. Päätösalueiden piirto

Piirrä päätösalueet seuraavilla luokittelusäännöillä:

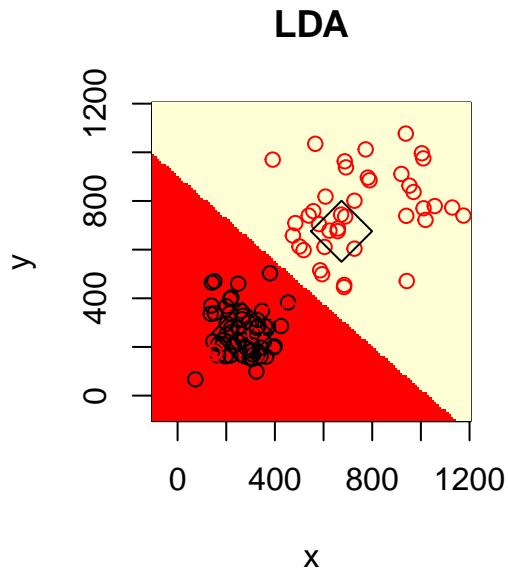
- a) posteriori, kun kovarianssimatriisi on poolattu matriisi ja priorit datasta
- b) lda-funktiolla
- c) posteriori, kun eri kovarianssimatriisit ja priorit datasta
- d) qda-funktiolla
- e) posteriori, kun eri kovarianssimatriisit, mutta kovariansit nollia ja priorit datasta (NB)
- f) naive bayes - valmiilla funktioilla

```
# a) normalijakauman tiheysfunktio posteriori, kovarianssimatriisina
# poolattu matriisi (matkii LDA:ta)
library(mvtnorm)
pi1<-nb/(nb+nh)
pi2<-nh/(nb+nh)
N<-dim(osa.train)[1]
covp<-((nb-1)*covb+(nh-1)*covh)/(N-2)
luokatp<-rep(2,length=ntest)
for (i in 1:ntest)
{
  db<-pi1*dmvnorm(test.features[i],mb,covp)
  dh<-pi2*dmvnorm(test.features[i],mh,covp)
  if (db>dh)
    luokatp[i]<-1
}
z <- matrix(luokatp, ncol=length(y))
par(pty="s")
image(x, y, z, main="oma LDA")
```

```
points(osa.train[,c("Width","Height")], col=osa.train$Species)
points(data.frame(rbind(mb,mh)),cex=3,pch=5)
```



```
# b) lda-funktiolla
library(MASS)
LDA<-lda(Species~.,data=osa.train)
luokatl<-predict(LDA, test.features)$class
luokatl<-as.numeric(luokatl)
z <- matrix(luokatl, ncol=length(y))
image(x, y, z, main="LDA")
par(pty="s")
points(osa.train[,c("Width","Height")], col=osa.train$Species)
points(data.frame(rbind(mb,mh)),cex=3,pch=5)
```



```

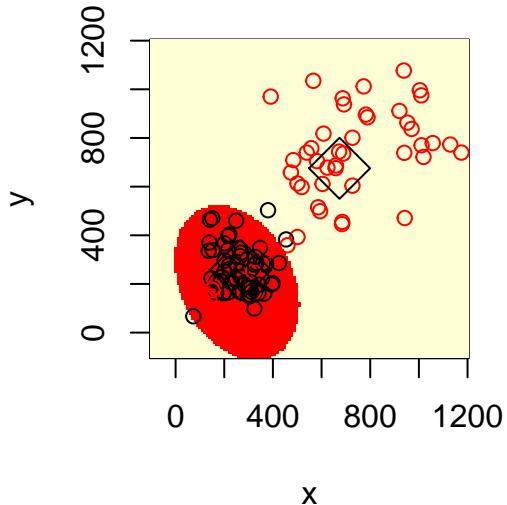
sum(luokatp-luokatl) # samat

## [1] 0

# c) normaalijakauman tiheysfunktioista posteriori (matkii QDA:ta)
luokatfp<-rep(2,length=ntest)
for (i in 1:ntest)
{
  db<-pi1*dmvnorm(test.features[i,],mb,covb)
  dh<-pi2*dmvnorm(test.features[i,],mh,covh)
  if (db>=dh)
    luokatfp[i]<-1
}
z <- matrix(luokatfp, ncol=length(y))
par(pty="s")
image(x, y, z, main="oma QDA")
points(osa.train[,c("Width","Height")], col=osa.train$Species)
points(data.frame(rbind(mb,mh)),cex=3,pch=5)

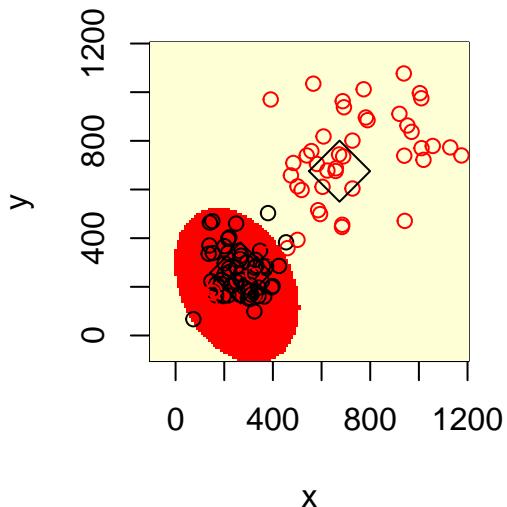
```

## oma QDA



```
# d) qda-funktiolla
library(MASS)
QDA<-qda(Species~, data=osa.train)
luokatq<-predict(QDA, test.features)$class
luokatq<-as.numeric(luokatq)
z <- matrix(luokatq, ncol=length(y))
image(x, y, z, main="QDA")
points(osa.train[,c("Width", "Height")], col=osa.train$Species)
points(data.frame(rbind(mb,mh)), cex=3, pch=5)
```

## QDA



```

sum(luokatq-luokatfp) # samat

## [1] 0

log(det(covh))-log(det(covb))

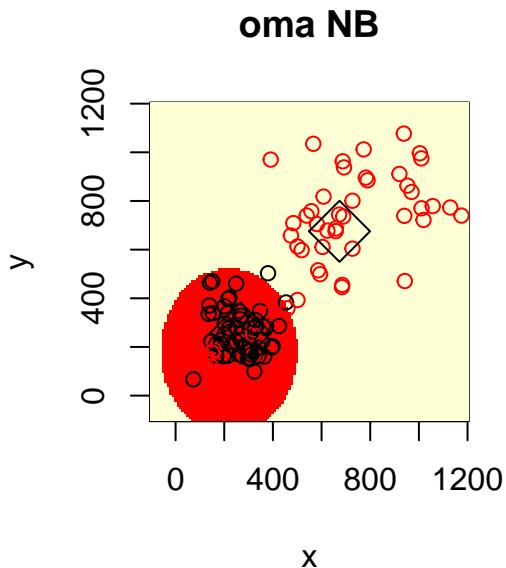
## [1] 3.661613

# ts. baetikseen enemmän kuin pelkällä
# mahalanobis-etäisyydellä

# e) posteriori, kun eri kovarianssimatriisit, mutta kovarianssit nollia
# (matkii NB:tä)
pi1<-nb/(nb+nh)
pi2<-nh/(nb+nh)
nbcovb<-covb
nbcovh<-covh
nbcovb[1,2]<-0
nbcovb[2,1]<-0
nbcovh[1,2]<-0
nbcovh[2,1]<-0

luokatnb<-rep(2,length=ntest)
for (i in 1:ntest)
{
  db<-pi1*dmvnorm(test.features[i,],mb,nbcovb)
  dh<-pi2*dmvnorm(test.features[i,],mh,nbcovh)
  if (db>=dh)
    luokatnb[i]<-1
}
z <- matrix(luokatnb, ncol=length(y))
par(pty="s")
image(x, y, z, main="oma NB")
points(osa.train[,c("Width","Height")], col=oso.train$Species)
points(data.frame(rbind(mb,mh)),cex=3,pch=5)

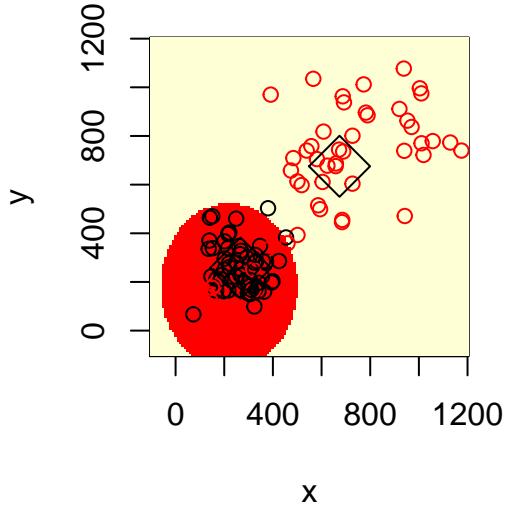
```



```
# f) naive bayes valmiilla funktiolla
library(naivebayes) # NB

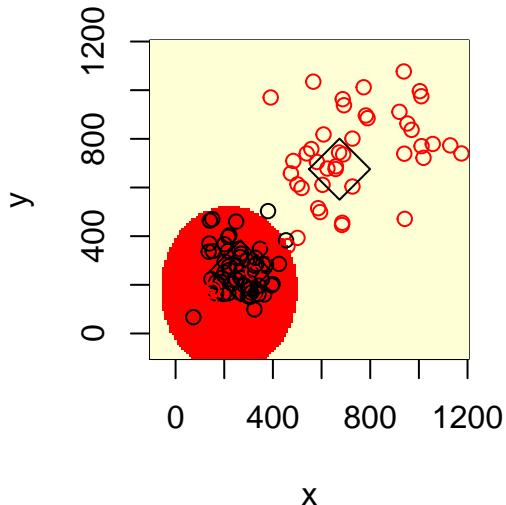
## Warning: package 'naivebayes' was built under R version 3.5.2
NB1<-naive_bayes(Species~, data=osa.train)
# plot(NB1) # NB
luokatnb1<-predict(NB1, test.features)
luokatnb1<-as.numeric(luokatnb1)
z <- matrix(luokatnb1, ncol=length(y))
par(pty="s")
image(x, y, z, main="NB, kun naive_bayes")
points(osa.train[,c("Width", "Height")], col=osa.train$Species)
points(data.frame(rbind(mb,mh)), cex=3,pch=5)
```

## NB, kun naive\_bayes



```
library(e1071) # NB
NB2<-naiveBayes(Species~, data=osa.train)
luokatnb2<-predict(NB2, test.features)
luokatnb2<-as.numeric(luokatnb2)
z <- matrix(luokatnb2, ncol=length(y))
par(pty="s")
image(x, y, z, main="NB, kun naiveBayes")
points(osa.train[,c("Width","Height")], col=osa.train$Species)
points(data.frame(rbind(mb,mh)),cex=3,pch=5)
```

## NB, kun naiveBayes



```

sum(luokatnb-luokatnb1) # samat, onneksi
## [1] 0
sum(luokatnb-luokatnb2) # samat, onneksi
## [1] 0
# kolmas funktio NaiveBayes klaR-kirjastosta ei toiminut jostain syystä
# - vaikuttaa siltä, että edellä olevissa NB-funktioissa on joku ongelma, kun piirteitä
# on yksi (R3vast.R)
# - aikaisempien kokemusten perusteella muistelisin, että tiedossa hankaluksia, kun
# aineistossa on piirteitä, joita ei käytetä luokittelusäänön rakentamisessa
# - tässä tehtävässä oli enemmän kuin yksi piirre, mutta ei muita piirteitä

```

## 4. Päättörajan laskenta käsin

Tavoitteena on jakaa kaksiulotteinen piirreavaruus kahteen päättösaluueeseen (luokkia oli kaksi), joiden perusteella uusi yksilö luokitellaan.

Olkoon  $\pi_1 = 0.50$ ,  $\pi_2 = 0.50$ . Opetusaineistosta on saatu 1. luokan keskiarvovektoriksi  $\bar{x}_1 = [3 \ 6]'$  ja 2. luokan  $\bar{x}_2 = [3 \ -2]'$  sekä kovarianssimatriiseiksi

$$\mathbf{S}_1 = \begin{pmatrix} 0.5 & 0 \\ 0 & 2 \end{pmatrix}, \quad \mathbf{S}_2 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}.$$

Tutki, milloin  $\hat{P}(\omega_1|\mathbf{x}) = \hat{P}(\omega_2|\mathbf{x})$ , kun  $\mathbf{x} = [\mathbf{x}_1 \ \mathbf{x}_2]'$  ja osoita edellä olevan yhtälön olevan voimassa, kun

$$x_2 = 3.514 - 1.125x_1 + 0.1875x_1^2.$$

## 5. Toistetaan edellinen

- a) Simuloi edellisen tehtävän mukaisesta jakaumasta havaintoja siten, että  $n_1 = 60$  ja  $n_2 = 60$ .
- b) Laske päättöraja käyttäen samaa jakaumaa ja värijää vastaava alue luokittelun mukaan.

```

library(mvtnorm)
# a) simulointi
m1<-c(3,6)
m2<-c(3,-2)
S1<-matrix(c(0.5,0,0,2),byrow=TRUE,ncol=2)
S2<-matrix(c(2,0,0,2),byrow=TRUE,ncol=2)

alustus<-rmvnorm(60,m1,S1)
simdata1<-data.frame("Species"=rep(1,60),"x1"=alustus[,1],"x2"=alustus[,2])
alustus<-rmvnorm(60,m2,S2)
simdata2<-data.frame("Species"=rep(2,60),"x1"=alustus[,1],"x2"=alustus[,2])

simdata<-rbind(simdata1, simdata2)

# b) alustetaan hila
x<-seq(-10,10,by=0.1)
y<-seq(-10,10,by=0.1)
test.features <- mat.or.vec(length(x)*length(y), 2)
test.features<-data.frame("x1"=test.features[,1],"x2"=test.features[,2])

```

```

ntest<-length(test.features[,1])
ntest

## [1] 40401

for(j in 1:length(y)){
  for(i in 1:length(x)){
    test.features[i + (j-1)*length(x),1] = x[i]
    test.features[i + (j-1)*length(x),2] = y[j]
  }
}

# ensimmäinen kuva käyttäen tunnettuja parametrien arvoja
# päätöskäyrän tulee olla samassa kohdassa kuin osoitettiin

# päätösalue omalla NB:llä
library(mvtnorm)
pi1<-0.50
pi2<-0.50
pi1;pi2

## [1] 0.5

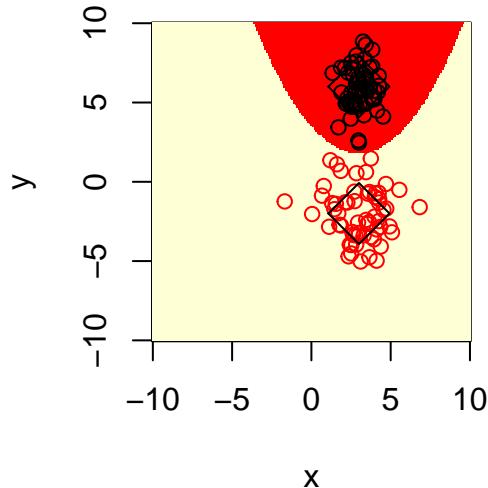
## [1] 0.5

luokatnb<-rep(2,length=ntest)
for (i in 1:ntest)
{
  db<-pi1*dmvnorm(test.features[i],m1,S1)
  dh<-pi2*dmvnorm(test.features[i],m2,S2)
  if (db>=dh)
    luokatnb[i]<-1
}

z <- matrix(luokatnb, ncol=length(y))
par(pty="s")
image(x, y, z, main="oma NB: teor. arvoilla (hila)")
points(simdata[,c("x1","x2")], col=simdata$Species)
points(data.frame(rbind(m1,m2)),cex=3,pch=5)

```

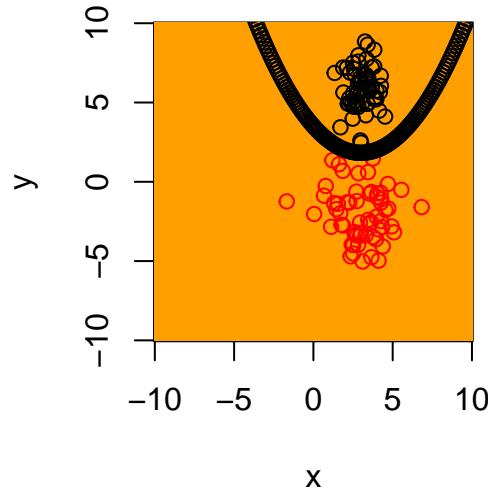
## oma NB: teor. arvoilla (hila)



```
#x1<-x
#x2<-3.514-1.125*x1+0.1875*x1^2
#points(data.frame(cbind(x1,x2)))

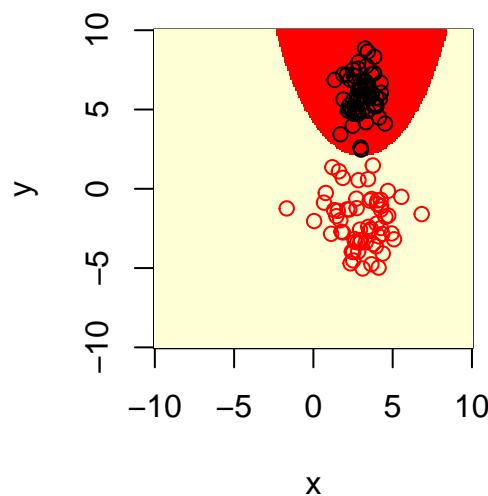
# piirretään
luokatpohja<-rep(2,length=ntest)
z <- matrix(luokatpohja, ncol=length(y))
x1<-x
x2<-3.514-1.125*x1+0.1875*x1^2
par(pty="s")
image(x, y, z, main="oma NB: teor.arvoilla (käyrä)")
points(simdata[,c("x1","x2")], col=simdata$Species)
points(data.frame(cbind(x1,x2)))
```

## oma NB: teor.arvoilla (käyrä)



```
NB1<-naive_bayes(factor(Species)~, data=simdata)
luokatnb1<-predict(NB1, test.features)
luokatnb1<-as.numeric(luokatnb1)
z <- matrix(luokatnb1, ncol=length(y))
par(pty="s")
image(x, y, z, main="estimoitu NB funktiolla naive_bayes")
points(simdata[,c("x1","x2")], col=simdata$Species)
points(data.frame(rbind(mb,mh)), cex=3, pch=5)
```

## estimoitu NB funktiolla naive\_bayes



```

sum(luokatnb-luokatnb1) # toinen tarkka ja toinen estimoitu
## [1] -1172

```

## 6. Neuroverkot

Tutkitaan tilannetta, jossa on yksi piilokerros ja siinä kaksi piiloyksikköä  $z_1$  ja  $z_2$  siten, että lajeja on kaksi ja piirremuuttuja on kaksi.

- Kirjoita muuttujille  $z_1$  ja  $z_2$  sekä todennäköisyyksille  $p_1(\mathbf{x}, \Psi)$  ja  $p_2(\mathbf{x}, \Psi)$  kaavat luentokalvoihin perustuen.
- Tutki samaa aineistoa kuin tehtävässä 1 ja toteuta siihen neuroverkko. Tulosta objekti `NN` ja hae sieltä samoja elementtejä kuin a) kohdassa.

```

#library(RWeka) # ajettu linuxilla
#MLP<-make_Weka_classifier("weka/classifiers/functions/MultilayerPerceptron")
#NN<-MLP(Species~, data=osa.train, control=Weka_control(V=20,S=1,E=10))
#NN
#Sigmoid Node 0
#   Inputs   Weights
#   Threshold      -5.788333839227147
#   Node 2       3.63646598619791
#   Node 3       5.0559425201677756
#Sigmoid Node 1
#   Inputs   Weights
#   Threshold      5.783559753350272
#   Node 2      -3.57127608724073
#   Node 3      -5.1154188983684765
#Sigmoid Node 2
#   Inputs   Weights
#   Threshold      -2.2794717293629083
#   Attrib Width    -7.689747153994943
#   Attrib Height   -1.9183998758908798
#Sigmoid Node 3
#   Inputs   Weights
#   Threshold      -2.950203619061042
#   Attrib Width     -9.552900559733773
#   Attrib Height    -1.9880158589314187
#Class Baetis
#   Input
#   Node 0
#Class Hydro
#   Input
#   Node 1

```

## 7. Päätöspuu

Tutustu alla olevaan koodiin ja mieti, miten sillä luokiteltaisiin testiaineisto ja toteuta se ainakin muutamalle testiaineiston yksikölle.

```

# muuttujien luokittelu
# install.packages("Hmisc")
library(Hmisc)

```

```

## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
## Loading required package: ggplot2
##
## Attaching package: 'Hmisc'
## The following object is masked from 'package:e1071':
##   impute
## The following objects are masked from 'package:base':
##   format.pval, units
osa.train$Widthluok<-cut2(osa.train$Width, cuts=median(osa.train$Width))
tw<-table(osa.train$Species,osa.train$Widthluok)
w1<-tw[,1]/sum(tw[,1])
w2<-tw[,2]/sum(tw[,2])
pip1<-sum(tw[,1])/(sum(tw[,1])+sum(tw[,2]))
pip2<-sum(tw[,2])/(sum(tw[,1])+sum(tw[,2]))
# Entropia H(Y/X) Width-muuttujalle
-(sum(w1*log(w1))*pip1 +sum(w2*log(w2))*pip2)

## [1] 0.4175376

osa.train$Heightluok<-cut2(osa.train$Height, cuts=median(osa.train$Height))
tw<-table(osa.train$Species,osa.train$Heightluok)
w1<-tw[,1]/sum(tw[,1])
w2<-tw[,2]/sum(tw[,2])
pip1<-sum(tw[,1])/(sum(tw[,1])+sum(tw[,2]))
pip2<-sum(tw[,2])/(sum(tw[,1])+sum(tw[,2]))
# Entropia H(Y/X) Height-muuttujalle
-(sum(w1*log(w1))*pip1 +sum(w2*log(w2))*pip2)

## [1] 0.4487504

# Height-muuttujan avulla tehdään ensimmäinen jako
osa.train$osajoukko1<- ifelse(osa.train$Height<=median(osa.train$Height),1,2)
lyhyet<-subset(osa.train,osajoukko1==1)
pitkat<-subset(osa.train,osajoukko1==2)

# Width-muuttujan avulla tehdään toinen jako
lyhyet$osajoukko2<-ifelse(lyhyet$Width<=median(osa.train$Width),1,2)
lyhyetkapeat<-subset(lyhyet, osajoukko2==1)
lyhyetleveat<-subset(lyhyet, osajoukko2==2)

pitkat$osajoukko2<-ifelse(pitkat$Width<=median(osa.train$Width),1,2)
pitkatkapeat<-subset(pitkat, osajoukko2==1)
pitkatleveat<-subset(pitkat, osajoukko2==2)

# neljä porukkaa, katsotaan, mitä niistä löytyy
summary(lyhyetkapeat) # testiyksilön luokka => baetis

##      Species          Width           Height        Widthluok

```

```

##  Baetis:42  Min.    : 73.0   Min.    : 67.0   [ 73, 341):46
##  Hydro  : 5   1st Qu.:198.0  1st Qu.:167.0  [ 341,1175]: 1
##                Median :265.0   Median :208.0
##                Mean   :247.1   Mean   :218.3
##                3rd Qu.:306.0  3rd Qu.:268.5
##                Max.   :341.0   Max.   :339.0
##                Heightluok osajoukko1 osajoukko2
## [ 67, 339):45  Min.    :1     Min.    :1
## [ 339,1077]: 2 1st Qu.:1     1st Qu.:1
##                Median :1     Median :1
##                Mean   :1     Mean   :1
##                3rd Qu.:1     3rd Qu.:1
##                Max.   :1     Max.   :1

```

```
summary(lyhyetleveat) # testiyksilön luokka => baetis
```

```

##      Species      Width       Height      Widthluok
##  Baetis:8   Min.    :342.0   Min.    :160.0   [ 73, 341):0
##  Hydro  :1   1st Qu.:359.0  1st Qu.:198.0  [ 341,1175]:9
##                Median :372.0   Median :222.0
##                Mean   :376.1   Mean   :240.8
##                3rd Qu.:395.0  3rd Qu.:285.0
##                Max.   :425.0   Max.   :333.0
##                Heightluok osajoukko1 osajoukko2
## [ 67, 339):9  Min.    :1     Min.    :2
## [ 339,1077]:0 1st Qu.:1     1st Qu.:2
##                Median :1     Median :2
##                Mean   :1     Mean   :2
##                3rd Qu.:1     3rd Qu.:2
##                Max.   :1     Max.   :2

```

```
summary(pitkatkapeat) # testiyksilön luokka => baetis
```

```

##      Species      Width       Height      Widthluok
##  Baetis:8   Min.    :139.0   Min.    :348.0   [ 73, 341):8
##  Hydro  :0   1st Qu.:148.5  1st Qu.:369.2  [ 341,1175]:0
##                Median :208.5   Median :400.0
##                Mean   :197.2   Mean   :409.9
##                3rd Qu.:228.0  3rd Qu.:461.0
##                Max.   :260.0   Max.   :470.0
##                Heightluok osajoukko1 osajoukko2
## [ 67, 339):0  Min.    :2     Min.    :1
## [ 339,1077]:8 1st Qu.:2     1st Qu.:1
##                Median :2     Median :1
##                Mean   :2     Mean   :1
##                3rd Qu.:2     3rd Qu.:1
##                Max.   :2     Max.   :1

```

```
summary(pitkatileveat) # testiyksilön luokka => hydro
```

```

##      Species      Width       Height      Widthluok
##  Baetis: 3   Min.    : 347.0   Min.    : 348.0   [ 73, 341): 0
##  Hydro :42   1st Qu.: 558.0  1st Qu.: 604.0  [ 341,1175]:45
##                Median : 684.0   Median : 739.0
##                Mean   : 712.5   Mean   : 719.8
##                3rd Qu.: 938.0  3rd Qu.: 863.0

```

```

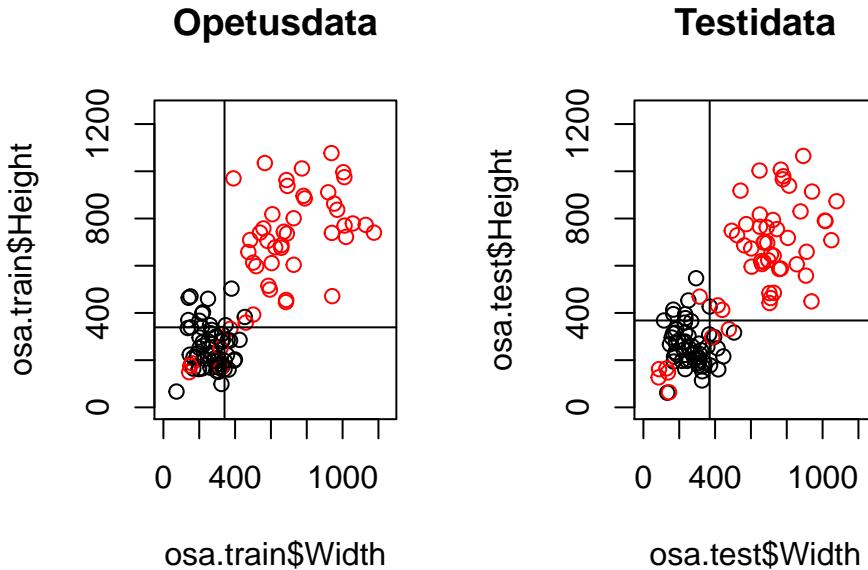
##          Max.    :1175.0    Max.    :1077.0
##      Heightluok   osajoukko1   osajoukko2
##  [ 67, 339): 0    Min.    :2    Min.    :2
##  [ 339,1077]:45  1st Qu.:2    1st Qu.:2
##                           Median :2    Median :2
##                           Mean    :2    Mean    :2
##                           3rd Qu.:2    3rd Qu.:2
##                           Max.    :2    Max.    :2

# tutkitaan jokoja tarkemmin
par(mfrow=c(1,2))

plot(osa.train$Width, osa.train$Height, col = osa.train$Species,
     main = "Opetusdata", xlim = c(0,1250), ylim = c(0,1250))
abline(h = median(osa.train$Height))
abline(v = median(osa.train$Width))

plot(osa.test$Width, osa.test$Height, col = osa.test$Species,
     main = "Testidata", xlim = c(0,1250), ylim = c(0,1250))
abline(h = median(osa.test$Height))
abline(v = median(osa.test$Width))

```



```

# Testidatan yksilöt luokitellaan sen mukaan, mihin ikkunaan ne osuvat. Huomataan, että osa
# menee väärin

```

## 8. Entropiasta (siirrettiin seuraaviin demoihin)

Päätöspuun yhteydessä hyödynnetään entropiaa. Näytä, että arvot  $\pi_l, l = 1, \dots, k$  ovat yhtäsuuria, kun entropia  $H(Y)$  on maksimissaan, ja  $H_{max} = \ln(k)$ . Ts. maksimoi  $H(Y)$ :tä:

$$\max_{\pi_1, \dots, \pi_{k-1}} H(Y) = \max_{\pi_1, \dots, \pi_{k-1}} \left\{ - \sum_{l=1}^{k-1} \pi_l \ln(\pi_l) - (1 - \sum_{l=1}^{k-1} \pi_l) \ln(1 - \sum_{l=1}^{k-1} \pi_l) \right\},$$

jossa  $\pi_k = (1 - \sum_{l=1}^{k-1} \pi_l)$ .

Vinkki: Vertaa, miten haetaan suurimman uskottavuuden estimaatit!

# TILS646

DEMO 5, 19.2.2019, klo 8:30.

Palauta R-tehtävien ratkaisut (R-koodit ja tulokset tulkintoineen) Koppaan (tehtävänpalautus) viimeistään demoja edeltävänä maanantaina klo 20:00 mennessä. Nimeä tiedosto etunimisukunimi\_demo5.R

Vain palautettuina tehtävistä saa demohyvityksiä. Jos ei pääse demoryhmään, demopisteistä saa vain puolet. Valmistaudu esittämään ratkaisusi demoryhmässä.

## 1. Entropiasta

Päätöspuun yhteydessä hyödynnetään entropiaa. Näytä, että arvot  $\pi_l, l = 1, \dots, k$  ovat yhtä suuria, kun entropia  $H(Y)$  on maksimissaan, ja  $H_{max} = \ln(k)$ . Ts. hae  $H(Y)$ :n maksimikohta  $\pi_l, l = 1, \dots, k$  suhteeseen.

$$\max_{\pi_1, \dots, \pi_{k-1}} H(Y) = \max_{\pi_1, \dots, \pi_{k-1}} \left\{ - \sum_{l=1}^{k-1} \pi_l \ln(\pi_l) - (1 - \sum_{l=1}^{k-1} \pi_l) \ln(1 - \sum_{l=1}^{k-1} \pi_l) \right\},$$

jossa  $\pi_k = (1 - \sum_{l=1}^{k-1} \pi_l)$ .

Vinkki: Vertaa, miten haetaan suurimman uskottavuuden estimaatit!

## 2. Vertailututkimus Err:n arvioimiseksi

Tutustu Lasse Moision pro gradu -tutkielmaan. Etsi vastaukset seuraaviin kysymyksiin

- Mitä uutta työssä on aikaisempiin tutkimuksiin verrattuna?
- Mikä on tutkimuksen tulos, kun luokittelija on qda?
- Miten käytetään stepclass-funktiota muuttujien valintaan?

```
# a) uutta
# -Bayes-virheen käyttö vertailuarvona
# -simuloidut aineistot monimutkaisempia: enemmän luokkia ja piirteitä
# b) abstraktin mukaan parhaita estimaattoreita Err:n estimoimiseksi olivat:
# -Err.632+bootstrap-estimaattori
# -Err.632+bootstrap-estimaattori
# -toistettu ristiinvaliointi K=10.
# -edellisen kanssa lähes yhtä hyvä: toistettu opetus- ja testiaineistojen
# jako jakosuhteella 90/10
```

## 3. Tukivektorikone

Tutustutaan tukivektorikoneisiin.

- Lue Kopasta aineistot, `circ_train.txt` ja `circ_test.txt`. Käytä tässä tehtävässä vasta kahta ensimmäistä muuttujaa (`x1` ja `x2`), joilla selität luokkaa `y`.
- Piirrä aineistosta kuva, jossa luokat ovat eri väreillä.
- Sovita aineistoon näillä kahdella piirteellä tukivektorikone, jossa ei vielä tehdä muunnoksia alkuperäisille piirteille (`kernel="linear"`).
- Piirrä kuva päättösrajasta ja aineistosta.

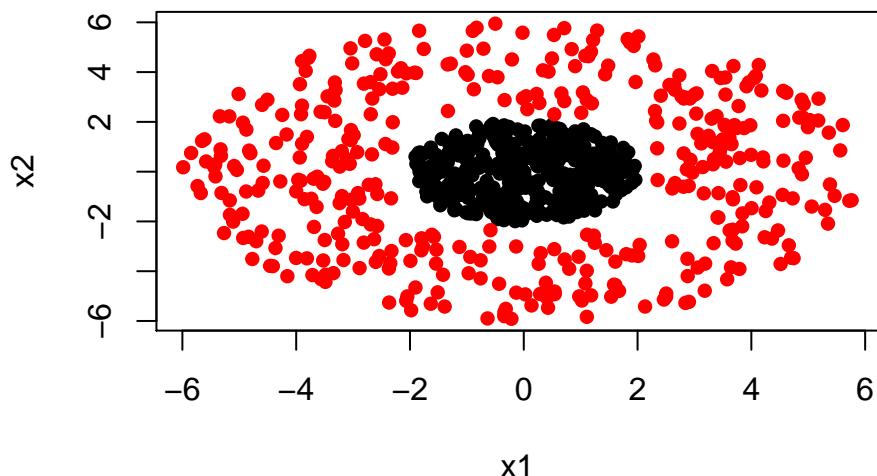
```

# a) aineistojen luku
opetus<-read.table("circ.train.txt",header=TRUE)
testi<-read.table("circ.test.txt",header=TRUE)

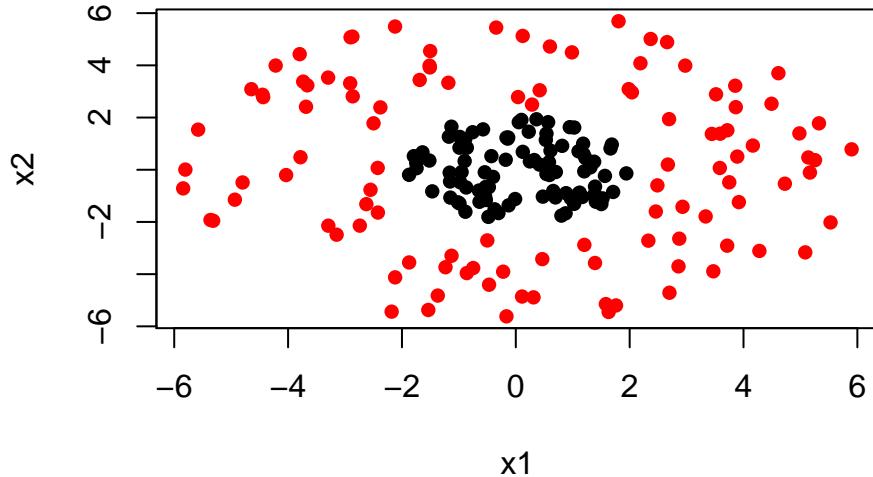
# y:stä faktori
opetus$y<-as.factor(opetus$y)
testi$y<-as.factor(testi$y)

#b) aineiston plottaus
plot(opetus$x1,opetus$x2,col=opetus$y,xlab="x1",ylab="x2",pch=16)

```



```
plot(testi$x1,testi$x2,col=testi$y,xlab="x1",ylab="x2",pch=16)
```



```
#c) tukivektorikoneen sovitus
library(e1071)
malli1<-svm(y~x1+x2,kernel="linear",data=opetus)
summary(malli1)

##
## Call:
## svm(formula = y ~ x1 + x2, data = opetus, kernel = "linear")
##
##
## Parameters:
##      SVM-Type:  C-classification
##      SVM-Kernel:  linear
##              cost:  1
##              gamma:  0.5
##
## Number of Support Vectors:  781
##
##  ( 390 391 )
##
##
## Number of Classes:  2
##
## Levels:
## -1 1

nrow(opetus)

## [1] 800
pred<-predict(malli1,newdata=testi)
1-sum(pred==testi$y)/nrow(testi)

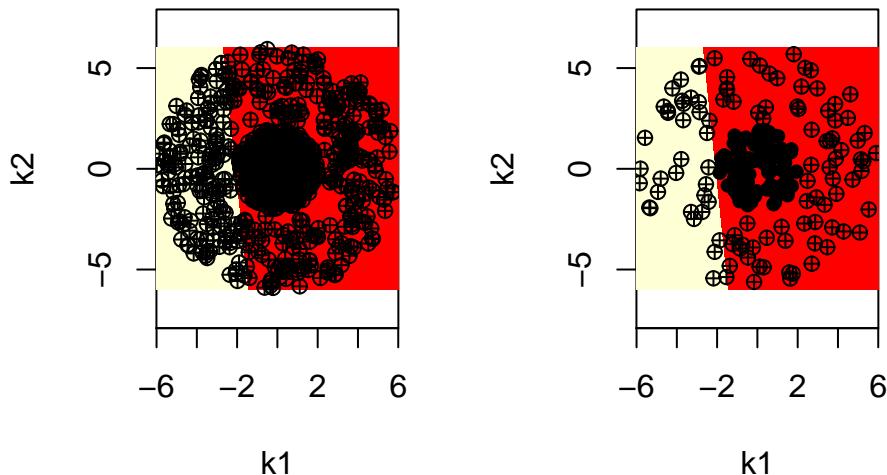
## [1] 0.36
```

```

# plot(malli1,opetus)

#d) päätösrajan piirto
k1<-seq(-6,6,by=0.01)
k2<-seq(-6,6,by=0.01)
test_vector<-mat.or.vec(length(k1)*length(k2),2)
for(j in 1:length(k2)){
  for(i in 1:length(k1)){
    test_vector[i+(j-1)*length(k1),1]<-k1[i]
    test_vector[i+(j-1)*length(k2),2]<-k2[j]
  }
}
k3<-matrix(as.numeric(as.matrix(predict(malli1,test_vector))),ncol=length(k2))
par(mfrow=c(1,2))
# opetusdata
image(k1,k2,k3,asp=1)
points(opetus[opetus$y==1,1],opetus[opetus$y==1,2],pch=10)
points(opetus[opetus$y==-1,1],opetus[opetus$y==-1,2],pch=10)
# testidata
image(k1,k2,k3,asp=1)
points(testi[testi$y==1,1],testi[testi$y==1,2],pch=10)
points(testi[testi$y==-1,1],testi[testi$y==-1,2],pch=16)

```



#### 4. Jatketaan saman aineiston ja tukivektorikoneen parissa

- Muuttuja  $x_3$  on muunnos kahdesta ensimmäisestä, joka projisoitiin kartion pinnalle, kuten luentomonisteessa. Sovita nyt tukivektorikone, jossa käytät kaikkia kolmea piirrettä. Paranevatko tulokset?
- Sovita aineistoon tukivektorikone, jossa selität luokkaa  $y$  kahdella ensimmäisellä piirteellä,  $x_1$  ja  $x_2$ , mutta käytä tällä kertaa joitain toista ydintä (esim. `kernel="radial"`).
- Visualisoi ensimmäisen tehtävän tapaan tämän luokittelijan tuottama päätösraja.

d) Selvitää, miten R tekee luokittelun, kun luokkia on enemmän kuin kaksi.

```
#a) mallin sovitus, kun piirteinä x1, x2, x3
malli2<-svm(y~x1+x2+x3,kernel="linear",data=opetus)
summary(malli2)

##
## Call:
## svm(formula = y ~ x1 + x2 + x3, data = opetus, kernel = "linear")
##
##
## Parameters:
##   SVM-Type: C-classification
##   SVM-Kernel: linear
##   cost: 1
##   gamma: 0.3333333
##
## Number of Support Vectors: 30
##
##  ( 16 14 )
##
##
## Number of Classes: 2
##
## Levels:
## -1 1
pred<-predict(malli2,newdata=testi)
1-sum(pred==testi$y)/nrow(testi)
```

```
## [1] 0
#b) mallin sovitus, kun piirteistä x1 ja x2 tehdään muunnos
malli3<-svm(y~x1+x2,data=opetus,kernel="radial")
summary(malli3)
```

```
##
## Call:
## svm(formula = y ~ x1 + x2, data = opetus, kernel = "radial")
##
##
## Parameters:
##   SVM-Type: C-classification
##   SVM-Kernel: radial
##   cost: 1
##   gamma: 0.5
##
## Number of Support Vectors: 59
##
##  ( 30 29 )
##
##
## Number of Classes: 2
##
## Levels:
## -1 1
```

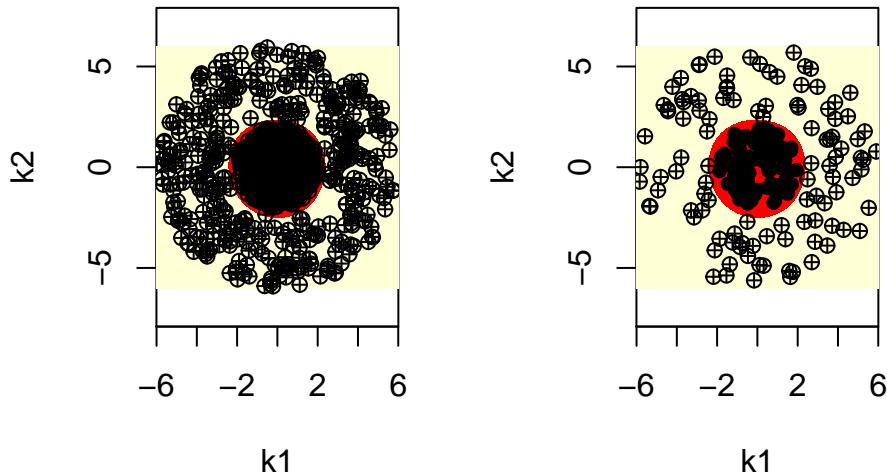
```

pred<-predict(malli3,newdata=testi)
1-sum(pred==testi$y)/nrow(testi)

## [1] 0

#c) päätösrajan piirto, kun malli3
k1<-seq(-6,6,by=0.01)
k2<-seq(-6,6,by=0.01)
test_vector<-mat.or.vec(length(k1)*length(k2),2)
for(j in 1:length(k2)){
  for(i in 1:length(k1)){
    test_vector[i+(j-1)*length(k1),1]<-k1[i]
    test_vector[i+(j-1)*length(k2),2]<-k2[j]
  }
}
k3<-matrix(as.numeric(as.matrix(predict(malli3,test_vector))),ncol=length(k2))
par(mfrow=c(1,2))
image(k1,k2,k3,asp=1)
points(opetus[opetus$y==1,1],opetus[opetus$y==1,2],pch=10)
points(opetus[opetus$y== -1,1],opetus[opetus$y== -1,2],pch=10)
image(k1,k2,k3,asp=1)
points(testi[testi$y==1,1],testi[testi$y==1,2],pch=10)
points(testi[testi$y== -1,1],testi[testi$y== -1,2],pch=16)

```



```

#d)
# ?sum
# For multiclass-classification with k levels, k>2, libsvm uses the
# 'one-against-one'-approach, in which k(k-1)/2 binary
# classifiers are trained; the #appropriate class is found by a voting scheme.
# degree:
# parameter needed for kernel of type polynomial (default: 3)
# gamma:
# parameter needed for all kernels except linear (default: 1/(data dimension))

```

## 5. Tukivektorikoneen parametrien tuunaus

Lue Kopasta aineisto `yeast.txt`. Luokkamuuttuja on `Class`.

- R-ohjelman `e1071`-kirjaston `svm`-funktiossa on valmis ristiinvalidointitoiminto. Selvitä, miten sitä käytetään.
- Arvo aineistosta 20% sivuun testiaineistoksi. Sovita aineistoon tukivektorikone, jossa käytät sädeperustaista (Gaussista) ydintä. Etsi ristiinvalidoinnin avulla optimaaliset parametrit  $C \in \{2^{-1} : 2^3\}$  ja  $\gamma \in \{2^{-6} : 2^{-3}\}$ .

`cost:`

`cost of constraints violation (default: 1)-it is the 'C'-constant of the regularization term in the Lagrange formulation.`

- Kun olet valinnut parametrien arvot, sovita malli niillä opetusaineistoon ja laske sen luokitteluvirhe erillisellä testiaineistolla.

```
# aineiston luku
aineisto<-read.table("yeast.txt",header=TRUE)
head(aineisto)

##      V1     V2     V3     V4     V5     V6     V7     V8 Class
## 1 0.58 0.61 0.47 0.13 0.5 0.0 0.48 0.22   MIT
## 2 0.43 0.67 0.48 0.27 0.5 0.0 0.53 0.22   MIT
## 3 0.64 0.62 0.49 0.15 0.5 0.0 0.53 0.22   MIT
## 4 0.58 0.44 0.57 0.13 0.5 0.0 0.54 0.22   NUC
## 5 0.42 0.44 0.48 0.54 0.5 0.0 0.48 0.22   MIT
## 6 0.51 0.40 0.56 0.17 0.5 0.5 0.49 0.22   CYT
nrow(aineisto)

## [1] 1484
table(aineisto$Class)

##
## CYT ERL EXC ME1 ME2 ME3 MIT NUC POX VAC
## 463    5   35   44   51  163  244  429   20   30

# a) parametrien tuunaus
library(e1071)
?tune.svm

## starting httpd help server ... done
?tune
# -annetaan samat datatiedot kuin sum()-funktiossa
# -annetaan optimoitaville parametriille vaihteluvälit
# -haetaan ristiinvalidoinnilla parametrien optimaaliset arvot

# b) 20% havainnoista erilleen lopullista testivirhettä varten
rivist<-sample(1:nrow(aineisto),0.8*nrow(aineisto))
opetus<-aineisto[rivist,]
testi<-aineisto[-rivit,]

# b) valitaan tukivektorikoneen parametrit C ja gamma ristiinvalidoinnilla
```

```

tuned_svm <- tune.svm(Class~, data = opetus, kernel="radial",
                       gamma = 2^{(-6:-3)}, cost = 2^{(-1:3)})
summary(tuned_svm)

##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##   gamma cost
##   0.0625    2
##
## - best performance: 0.3976855
##
## - Detailed performance results:
##   gamma cost   error dispersion
## 1  0.015625  0.5 0.4331007 0.04819314
## 2  0.031250  0.5 0.4128757 0.05138955
## 3  0.062500  0.5 0.4061031 0.05358813
## 4  0.125000  0.5 0.4061031 0.04635186
## 5  0.015625  1.0 0.4111879 0.04461449
## 6  0.031250  1.0 0.4010397 0.05054547
## 7  0.062500  1.0 0.4052628 0.04781999
## 8  0.125000  1.0 0.3985116 0.04645447
## 9  0.015625  2.0 0.4044296 0.04751745
## 10 0.031250  2.0 0.4060960 0.04876087
## 11 0.062500  2.0 0.3976855 0.04823697
## 12 0.125000  2.0 0.4027204 0.04304815
## 13 0.015625  4.0 0.3993591 0.04919518
## 14 0.031250  4.0 0.4018872 0.05103418
## 15 0.062500  4.0 0.4035536 0.04269560
## 16 0.125000  4.0 0.4111594 0.04517435
## 17 0.015625  8.0 0.4001994 0.05373187
## 18 0.031250  8.0 0.4001923 0.04908614
## 19 0.062500  8.0 0.4094645 0.04716468
## 20 0.125000  8.0 0.4229312 0.03624969

# c) mallin sovitus parhailla tuunausparametrien arvoilla
SVM<-svm(Class~, data = opetus, kernel="radial", gamma = 0.0625, cost = 1)
SVM.pred<-predict(SVM,testi)
table(SVM.pred,testi$Class)

##
## SVM.pred CYT ERL EXC ME1 ME2 ME3 MIT NUC POX VAC
##   CYT  54   0   1   0   0   2   18   37   1   2
##   ERL   0   0   0   0   0   0   0   0   0   0
##   EXC   0   0   7   3   1   0   0   2   0   0
##   ME1   0   0   0   5   1   0   0   0   0   0
##   ME2   1   0   1   0   5   1   4   0   0   0
##   ME3   1   0   0   0   1   29   1   3   0   1
##   MIT   9   0   1   0   1   0   26   5   1   0
##   NUC  21   0   0   0   0   1   4   43   0   0
##   POX   0   0   0   0   0   0   0   0   3   0

```

```

##      VAC    0    0    0    0    0    0    0    0    0    0
1 -sum(as.numeric(SVM.pred==testi$Class))/length(testi$Class)

## [1] 0.4208754

```

## 6. Bootstrap

Hyödynnetään bootstrap-estimaatteja luokitteluvirheiden vertailussa, kun aineistoa ei ole varaa jättää testaamista varten sivuun.

a) Lue Kopasta aineisto `cancer.txt`.

b) Laske  $\widehat{Err}^{.632+}$  lda:lle ja qda:lle.

```

# a) datan luku
cancer<-read.table("cancer.txt", header=TRUE)
dim(cancer)

## [1] 170  31

# head(cancer)
data_x<-cancer[,-1]
data_y<-cancer$Class

# b) odotettujen testivirheiden laskenta
library(sortinghat)

## Warning: package 'sortinghat' was built under R version 3.5.2

lda_wrapper <- function(object, newdata) {
  predict(object, newdata)$class
}
Err.632plus<-errorest(x=data_x, y=data_y,
                       estimator="632+",
                       train=MASS:::lda,
                       classify=lda_wrapper)
Err.632plus

## [1] 0.07282752

qda_wrapper <- function(object, newdata) {
  predict(object, newdata)$class
}
Err.632plus<-errorest(x=data_x, y=data_y,
                       estimator="632+",
                       train=MASS:::qda,
                       classify=qda_wrapper)
Err.632plus

## [1] 0.06367044

```

## 7. Err käyttäen opetusdatan- ja testidatan jakoa

Lue Kopasta aineisto `yeast.txt`.

a) Kirjoita funktio, joka 100 kertaa 1) valitsee satunnaisesti osan havainnosta opetusaineistoon ja osan testiaineistoon 2) sovittaa tukivektorikoneen opetusaineistoon ja ennustaa sillä luokat testiaineistolle ja

3) antaa arvion Err:lle.

- b) Miten aineistojen suhteet vaikuttavat virheisiin? Kokeile useaa eri jakoa, esim. 50%/50%, 70%/30% ja 90%/10%.

```
aineisto<-read.table("yeast.txt",header=TRUE)

# a) toistuva opetus- ja testiaineiston jako
errit<-function(data,pros){
  err.o<-NULL
  for(i in 1:100){
    rivot<-sample(1:nrow(data),pros/100*nrow(data))
    opetus<-data[rivot,]
    testi<-data[-rivot,]
    malli<-svm(Class~,data=opetus)
    pred<-predict(malli,newdata=testi)
    err.o[i]<-1-sum(pred==testi$Class)/nrow(testi)
  }
  list(testivirheet=err.o,Err=mean(err.o))
}
tulos5050<-errit(aineisto,50)
tulos6040<-errit(aineisto,60)
tulos8020<-errit(aineisto,80)
tulos9010<-errit(aineisto,90)

tulos5050$Err

## [1] 0.4057682
tulos6040$Err

## [1] 0.400202
tulos8020$Err

## [1] 0.4024579
tulos9010$Err

## [1] 0.4018121
# Mitä pienempi opetusaineisto, sitä suurempi ylisovittumisen vaara
```