Practica1

Álvaro Ribot & Jose Pérez Cano

10/02/2020

Exercici 2

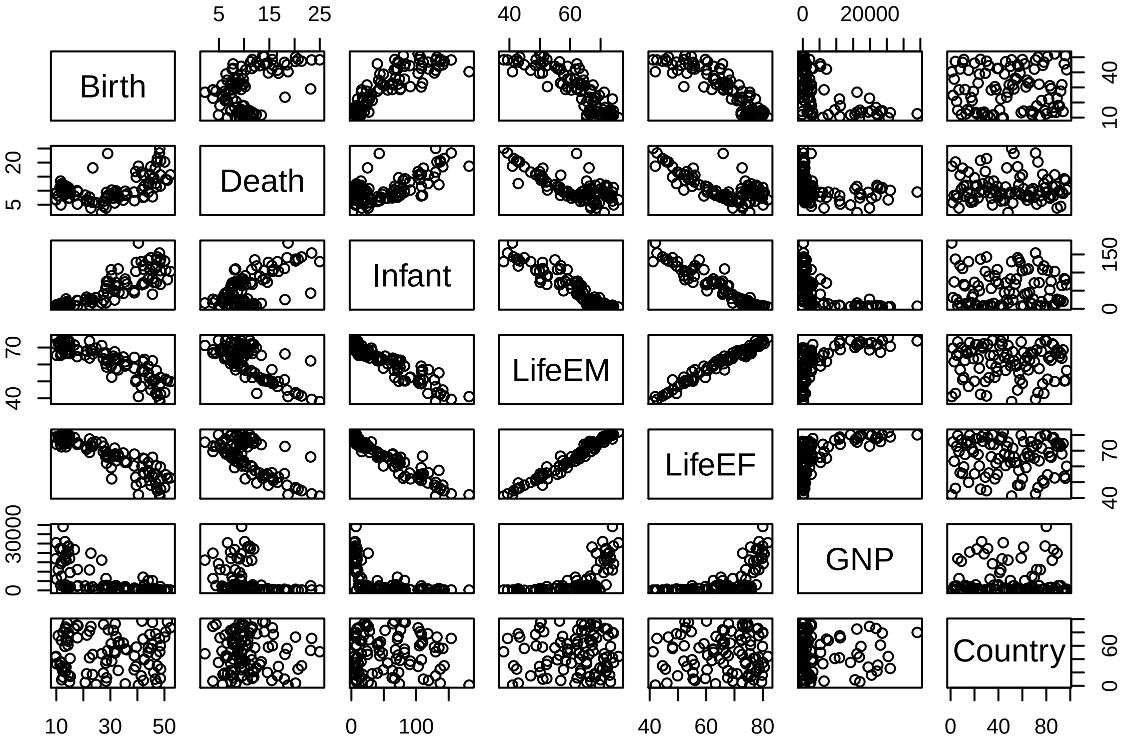
a)

```
dd2 <- read.table("http://www-eio.upc.es/~jan/Data/MVA/PovertyStudy.dat", header=T)
head(dd2)
```

```
##   Birth Death Infant LifeEM LifeEF  GNP           Country
## 1  24.7   5.7   30.8   69.6   75.5  600           Albania
## 2  12.5  11.9   14.4   68.3   74.7 2250          Bulgaria
## 3  13.4  11.7   11.3   71.8   77.7 2980    Czechoslovakia
## 4  12.0  12.4    7.6   69.8   75.9  -99 Former_E._Germany
## 5  11.6  13.4   14.8   65.4   73.8 2780           Hungary
## 6  14.3  10.2   16.0   67.2   75.7 1690            Poland
```

b) Relations between variables

```
pairs(dd2)
```



c) Missing values

```
dd2$GNP
```

```
##  [1]   600  2250  2980   -99  2780  1690  1640   -99  2242  1880  1320
## [12]  2370   630  2680  1940  1260   980   330  1110  1160  2560  2560
## [23]  2490 15540 26040 22080 19490 22320  5990  9550 16830 17320 23120
## [34]  7600 11020 23660 34064 16100 17000 25430 20470 21790   168  6340
## [45]  2490  3020 10920  1240 16150   -99  5220  7050  1630 19860   210
## [56]   -99   380 14210   350   570   -99  2320   110   170   380   730
## [67] 11160   470  1420   -99  2060   610  2040  1010   600   120   390
## [78]   260   390   370  5310   200   960    80  1030   360   240   120
## [89]  2530   480   810  1440   220   110   220   420   640
```

```
sprintf("There are %i countries without GNP", sum(dd2$GNP == -99))
```

```
## [1] "There are 6 countries without GNP"
```

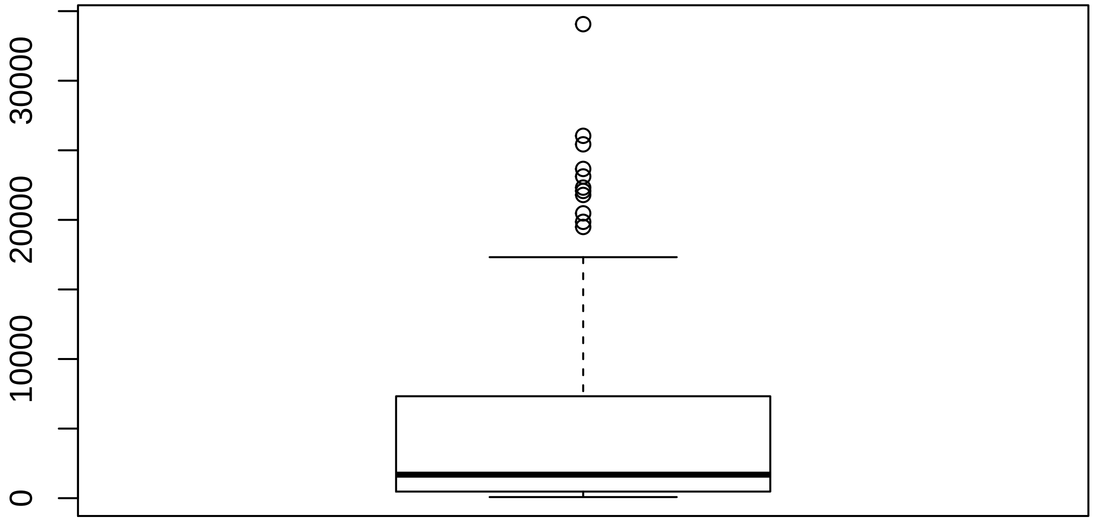
It seems the missing values are coded with -99.

d) Substituting by NAs

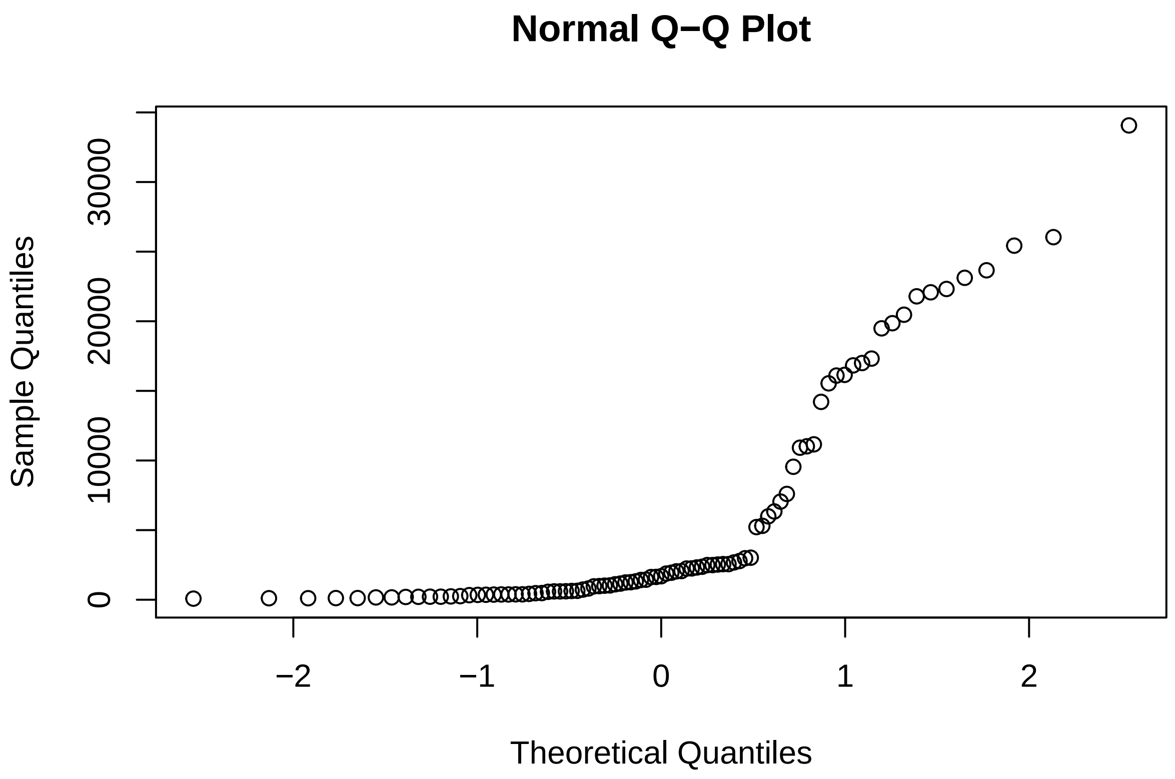
```
dd2$GNP[dd2$GNP == -99] <- NA
```

e) Boxplot & Q-Q plot

```
with(dd2, boxplot(GNP))
```

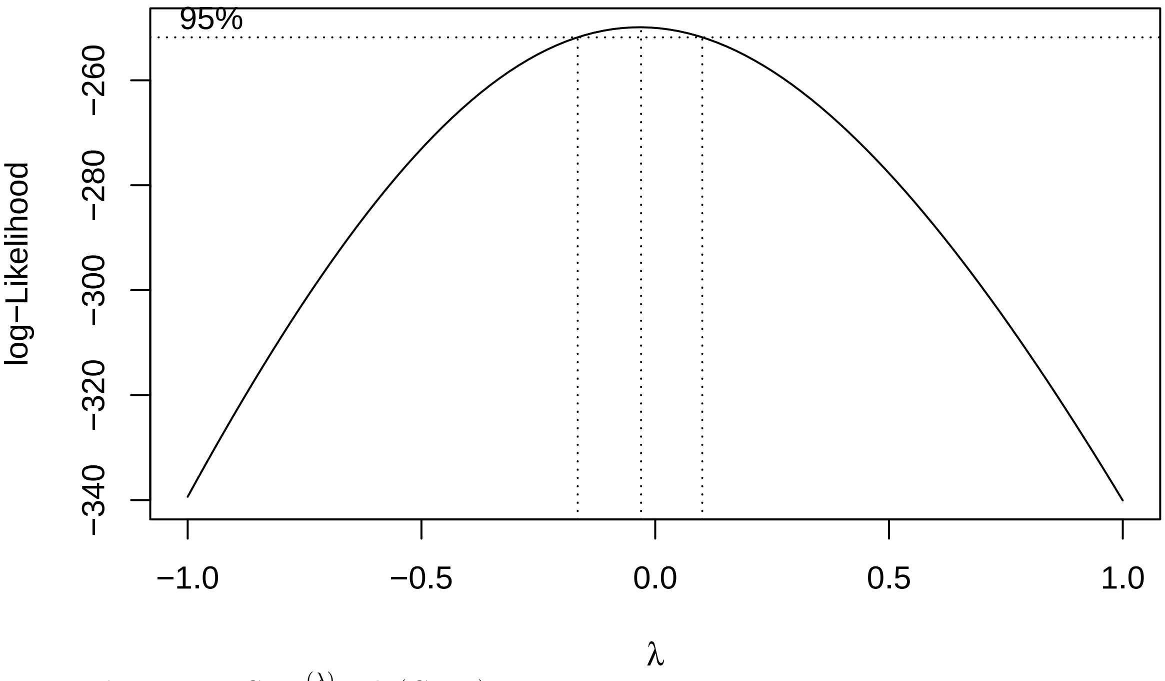


```
with(dd2, qqnorm(GNP))
```



f) BoxCox

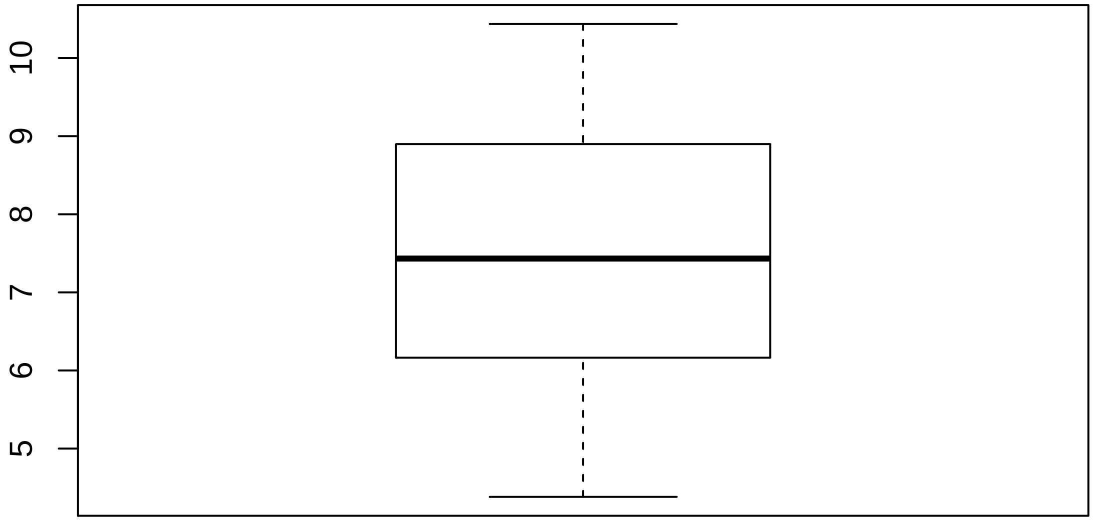
```
boxcox(lm(GNP~1, dd2),lambda = seq(-1, 1, by=0.1))
```



We can suppose $\lambda = 0 \implies GNP_i^{(\lambda)} = ln(GNP_i)$

g) Boxplot of transformed variable

```r
GNPmod <- log(dd2$GNP)
boxplot(GNPmod)
```



Now it follows a normal distribution as we can see a symmetric boxplot.

h) Linear regression

```r
m2 <- lm(GNPmod~Birth+Death+Infant+LifeEM+LifeEF, dd2)
anova(lm(GNPmod~1), m2)
```

```
## Analysis of Variance Table
##
## Model 1: GNPmod ~ 1
## Model 2: GNPmod ~ Birth + Death + Infant + LifeEM + LifeEF
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1     90 243.415
## 2     85  75.547  5    167.87 37.774 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
sprintf("The percentage of variance of the data explained by the model is %.2f%%", summary(m2)$r.square
```

```
## [1] "The percentage of variance of the data explained by the model is 68.96%"
```

i) Predicted values for missing values

```r
GNPmancant <- dd2[is.na(dd2$GNP),]
predict(m2, newdata = GNPmancant)
```

```
##        4        8       50       56       61       70
## 8.917075 8.510013 7.592047 5.523770 8.911541 7.790786
```

j) Variance of residuals

```r
aaa <- anova(lm(GNPmod~1),m2)
sprintf("The residual variance is %.2f", resvar <- aaa$RSS[2]/aaa$Df[2])
```

```
## [1] "The residual variance is 15.11"
```

k) Predictions with gaussian noise

```
set.seed(123)
noise <- rnorm(n = 6, sd = sqrt(resvar))
predict(m2, newdata = GNPmancant) + noise
```

```
##         4         8        50        56        61        70
##  6.738458  7.615293 13.650881  5.797842  9.414094 14.457392
```

Exercise 5

Data

a) Read data

```
dd <- read.table("http://www-eio.upc.es/~jan/Data/MVA/kernels.dat", header=T)
head(dd)
```

```
##    area perimeter compactness length width asymmetry groove
## 1 15.26     14.84      0.8710  5.763 3.312     2.221  5.220
## 2 14.88     14.57      0.8811  5.554 3.333     1.018  4.956
## 3 14.29     14.09      0.9050  5.291 3.337     2.699  4.825
## 4 13.84     13.94      0.8955  5.324 3.379     2.259  4.805
## 5 16.14     14.99      0.9034  5.658 3.562     1.355  5.175
## 6 14.38     14.21      0.8951  5.386 3.312     2.462  4.956
```

First questions

b) Means

```
apply(dd, MARGIN=2, FUN=mean)
```

```
##        area   perimeter compactness      length       width   asymmetry
##   14.334429   14.294286    0.880070    5.508057    3.244629    2.667403
##      groove
##    5.087214
```

c) Centered dataframe

```
ddc <- scale(dd, scale=FALSE)
head(ddc)
```

```
##             area   perimeter compactness      length      width
## [1,]  0.92557143  0.54571429    -0.00907  0.25494286 0.06737143
## [2,]  0.54557143  0.27571429     0.00103  0.04594286 0.08837143
## [3,] -0.04442857 -0.20428571     0.02493 -0.21705714 0.09237143
## [4,] -0.49442857 -0.35428571     0.01543 -0.18405714 0.13437143
## [5,]  1.80557143  0.69571429     0.02333  0.14994286 0.31737143
## [6,]  0.04557143 -0.08428571     0.01503 -0.12205714 0.06737143
##        asymmetry      groove
## [1,] -0.44640286  0.13278571
## [2,] -1.64940286 -0.13121429
## [3,]  0.03159714 -0.26221429
## [4,] -0.40840286 -0.28221429
## [5,] -1.31240286  0.08778571
## [6,] -0.20540286 -0.13121429
```

d) Covariance matrix

They aren't comparable because they are in different scales.

```r
cov(dd)
```

```
##                     area    perimeter   compactness        length
## area         1.477935176  0.684437267  0.0073032652  0.2349442360
## perimeter    0.684437267  0.332448033  0.0015396232  0.1229654037
## compactness  0.007303265  0.001539623  0.0002621462 -0.0005483954
## length       0.234944236  0.122965404 -0.0005483954  0.0535959677
## width        0.194349350  0.082169731  0.0019169046  0.0226387317
## asymmetry   -0.072043578 -0.036510244  0.0007024395 -0.0099707683
## groove       0.231122660  0.120674141 -0.0005596341  0.0528775818
##                    width     asymmetry        groove
## area         0.194349350 -0.0720435781  0.2311226605
## perimeter    0.082169731 -0.0365102443  0.1206741408
## compactness  0.001916905  0.0007024395 -0.0005596341
## length       0.022638732 -0.0099707683  0.0528775818
## width        0.031547280 -0.0055611207  0.0209413126
## asymmetry   -0.005561121  1.3780442298 -0.0034101557
## groove       0.020941313 -0.0034101557  0.0695370114
```

```r
sprintf("The variable with more variance is %s", names(which.max(apply(dd, MARGIN=2, FUN=var))))
```

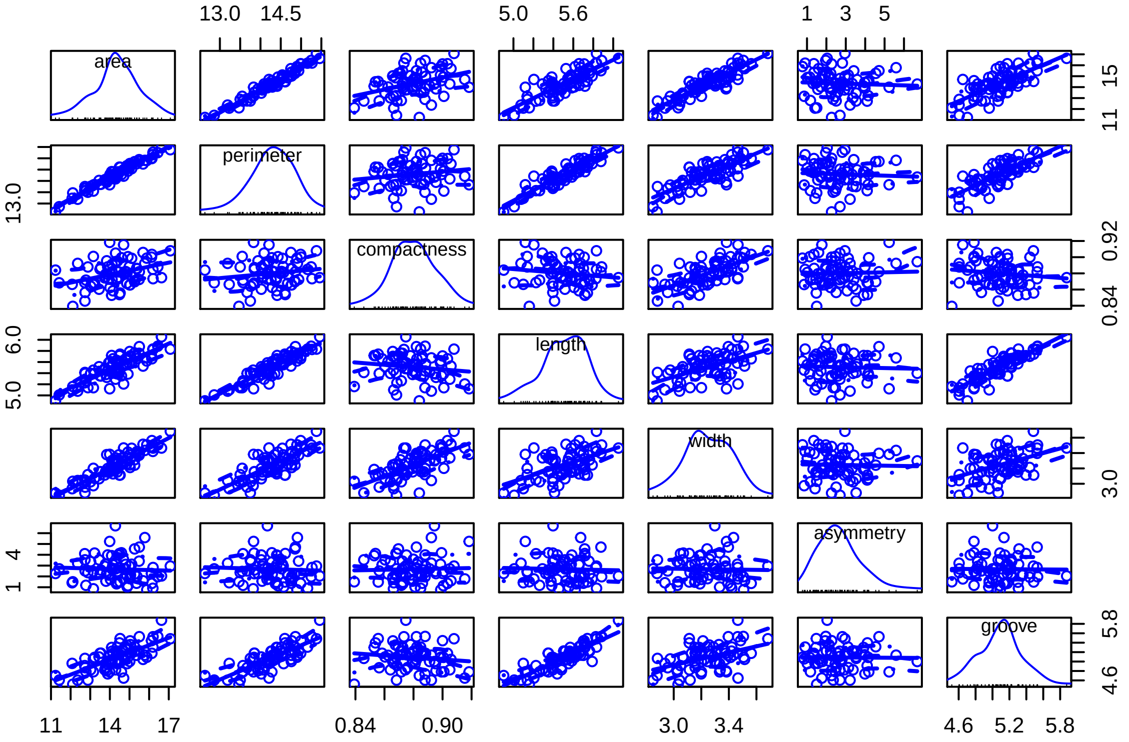
```
## [1] "The variable with more variance is area"
```

e) Correlation matrix

```r
cor(dd)
```

```
##                    area   perimeter compactness      length       width
## area         1.00000000  0.97643665  0.37103733  0.83477809  0.90006617
## perimeter    0.97643665  1.00000000  0.16492283  0.92120227  0.80235953
## compactness  0.37103733  0.16492283  1.00000000 -0.14630391  0.66657308
## length       0.83477809  0.92120227 -0.14630391  1.00000000  0.55056053
## width        0.90006617  0.80235953  0.66657308  0.55056053  1.00000000
## asymmetry   -0.05048194 -0.05394128  0.03695775 -0.03668859 -0.02667164
## groove       0.72095279  0.79367796 -0.13107635  0.86615879  0.44711056
##               asymmetry      groove
## area        -0.05048194  0.72095279
## perimeter   -0.05394128  0.79367796
## compactness  0.03695775 -0.13107635
## length      -0.03668859  0.86615879
## width       -0.02667164  0.44711056
## asymmetry    1.00000000 -0.01101627
## groove      -0.01101627  1.00000000
```

```r
scatterplotMatrix(dd)
```



```
abs(cor(dd)) > 0.5 # Strong linear correlation
```

```
##              area perimeter compactness length width asymmetry groove
## area         TRUE      TRUE       FALSE   TRUE  TRUE     FALSE   TRUE
## perimeter    TRUE      TRUE       FALSE   TRUE  TRUE     FALSE   TRUE
## compactness FALSE     FALSE        TRUE  FALSE  TRUE     FALSE  FALSE
## length       TRUE      TRUE       FALSE   TRUE  TRUE     FALSE   TRUE
## width        TRUE      TRUE        TRUE   TRUE  TRUE     FALSE  FALSE
## asymmetry   FALSE     FALSE       FALSE  FALSE FALSE      TRUE  FALSE
## groove       TRUE      TRUE       FALSE   TRUE FALSE     FALSE   TRUE
```

f) Standardized data frame

```
dds <- scale(dd)
head(dds)
```

```
##             area  perimeter compactness     length     width   asymmetry
## [1,]  0.76134631  0.9464626 -0.56019021  1.1012268 0.3793104 -0.38027291
## [2,]  0.44877011  0.4781866  0.06361587  0.1984504 0.4975433 -1.40506095
## [3,] -0.03654556 -0.3543040  1.53975104 -0.9375793 0.5200639  0.02691635
## [4,] -0.40670159 -0.6144574  0.95300275 -0.7950357 0.7565297 -0.34790221
## [5,]  1.48520698  1.2066159  1.44093027  0.6476789 1.7868449 -1.11798400
## [6,]  0.03748564 -0.1461814  0.92829756 -0.5272264 0.3793104 -0.17497456
##           groove
## [1,]  0.5035509
## [2,] -0.4975917
## [3,] -0.9943707
## [4,] -1.0702149
## [5,]  0.3329016
## [6,] -0.4975917
```

g)

We observe it is equal to the correlation matrix of the original dataframe.

```
cov(dds)
```

```
##                    area   perimeter compactness      length       width
## area         1.00000000  0.97643665  0.37103733  0.83477809  0.90006617
## perimeter    0.97643665  1.00000000  0.16492283  0.92120227  0.80235953
## compactness  0.37103733  0.16492283  1.00000000 -0.14630391  0.66657308
## length       0.83477809  0.92120227 -0.14630391  1.00000000  0.55056053
## width        0.90006617  0.80235953  0.66657308  0.55056053  1.00000000
## asymmetry   -0.05048194 -0.05394128  0.03695775 -0.03668859 -0.02667164
## groove       0.72095279  0.79367796 -0.13107635  0.86615879  0.44711056
##               asymmetry      groove
## area        -0.05048194  0.72095279
## perimeter   -0.05394128  0.79367796
## compactness  0.03695775 -0.13107635
## length      -0.03668859  0.86615879
## width       -0.02667164  0.44711056
## asymmetry    1.00000000 -0.01101627
## groove      -0.01101627  1.00000000
```

h) Euclidean distance

```
as.matrix(dist(dd[1:5,]))
```

```
##          1        2         3         4        5
## 1 0.000000 1.333578 1.4534352 1.7882615 1.274149
## 2 1.333578 0.000000 1.8684695 1.7597174 1.410420
## 3 1.453435 1.868469 0.0000000 0.6495716 2.519256
## 4 1.788262 1.759717 0.6495716 0.0000000 2.737101
## 5 1.274149 1.410420 2.5192564 2.7371013 0.000000
```

i) Centered / Standardize euclidean distance

```
print("Centered")
```

```
## [1] "Centered"
```

```
as.matrix(dist(ddc[1:5,]))
```

```
##          1        2         3         4        5
## 1 0.000000 1.333578 1.4534352 1.7882615 1.274149
## 2 1.333578 0.000000 1.8684695 1.7597174 1.410420
## 3 1.453435 1.868469 0.0000000 0.6495716 2.519256
## 4 1.788262 1.759717 0.6495716 0.0000000 2.737101
## 5 1.274149 1.410420 2.5192564 2.7371013 0.000000
```

```
print("Standardized")
```

```
## [1] "Standardized"
```

```
as.matrix(dist(dds[1:5,]))
```

```
##          1        2         3         4        5
## 1 0.000000 1.894091 3.6502730 3.5080553 2.712256
## 2 1.894091 0.000000 2.5876565 2.2839181 2.477507
## 3 3.650273 2.587656 0.0000000 0.8783193 3.457194
## 4 3.508055 2.283918 0.8783193 0.0000000 3.583076
## 5 2.712256 2.477507 3.4571945 3.5830763 0.000000
```

j)

The transformation in question is $f(\mathbf{x}) = \frac{\mathbf{x}-\overline{\mathbf{x}}}{\sqrt{\mathbf{Var}(\mathbf{x})}} + \overline{\mathbf{x}}$ where $\overline{x}$ is the vector where all entries are the mean of the vector x.

```
ddn <- t(t(dds)+apply(dd, MARGIN=2, FUN=mean))
head(ddn)
```

```
##           area perimeter compactness   length    width asymmetry   groove
## [1,] 15.09577  15.24075   0.3198798 6.609284 3.623939  2.287130 5.590765
## [2,] 14.78320  14.77247   0.9436859 5.706508 3.742172  1.262342 4.589623
## [3,] 14.29788  13.93998   2.4198210 4.570478 3.764692  2.694319 4.092844
## [4,] 13.92773  13.67983   1.8330727 4.713021 4.001158  2.319501 4.016999
## [5,] 15.81964  15.50090   2.3210003 6.155736 5.031474  1.549419 5.420116
## [6,] 14.37191  14.14810   1.8083676 4.980831 3.623939  2.492428 4.589623
```

```
apply(ddn, MARGIN=2, FUN=mean)
```

```
##        area   perimeter compactness      length       width   asymmetry 
##   14.334429   14.294286    0.880070    5.508057    3.244629    2.667403 
##      groove 
##    5.087214
```

```
abs(apply(ddn, MARGIN=2, FUN=mean)-apply(dd, MARGIN=2, FUN=mean)) < 1e-7
```

```
##        area   perimeter compactness      length       width   asymmetry 
##        TRUE        TRUE        TRUE        TRUE        TRUE        TRUE 
##      groove 
##        TRUE
```

```
apply(ddn, MARGIN=2, FUN=var)
```

```
##        area   perimeter compactness      length       width   asymmetry 
##           1           1           1           1           1           1 
##      groove 
##           1
```