STAT718A SHAPE AND IMAGE ANALYSIS Spring Session 2009

Homework 2 : Solutions

Question 1:

```
> library(shapes)
Loading required package: scatterplot3d
Loading required package: rgl
> ans <- procGPA( gorf.dat )
> ans$size
 [1] 235.1797 238.9676 235.1678 242.8732 229.2441 240.4774 242.7609 240.3222
 [9] 237.9483 224.6049 243.0905 240.7094 241.1553 231.0463 228.3241 228.1830
[17] 246.1534 228.6140 236.4241 243.4913 229.9236 236.1977 238.1339 238.7001
[25] 244.6891 227.0449 232.5376 245.2027 240.9097 245.2588
>
> plot( ans$rho , ans$size , ylab="size", xlab = "rho")
```

See Figure 1 for the plot of size versus rho.

```
bsv <- bookstein2d( gorf.dat)

bsv$bshpv   contains the Bookstein shape variables

lines( bsv$mshape[c(1,5,4,3,2,8,7,6,1),1], bsv$mshape[c(1,5,4,3,2,8,7,6,1),2] , col=2)


proc<- procGPA( gorf.dat)

pmean <- bookstein.shpv( proc$mshape )

lines( pmean[c(1,5,4,3,2,8,7,6,1),1], pmean[c(1,5,4,3,2,8,7,6,1),2] , col=3)

# NB both means are so close you cannot see the difference - see Figure 2.

> riemdist( pmean, bsv$mshape)
[1] 0.0002295508

# the largest specimen using centroid size is the 17th.

> c(1:30)[ans$size == max(ans$size) ]
[1] 17
```

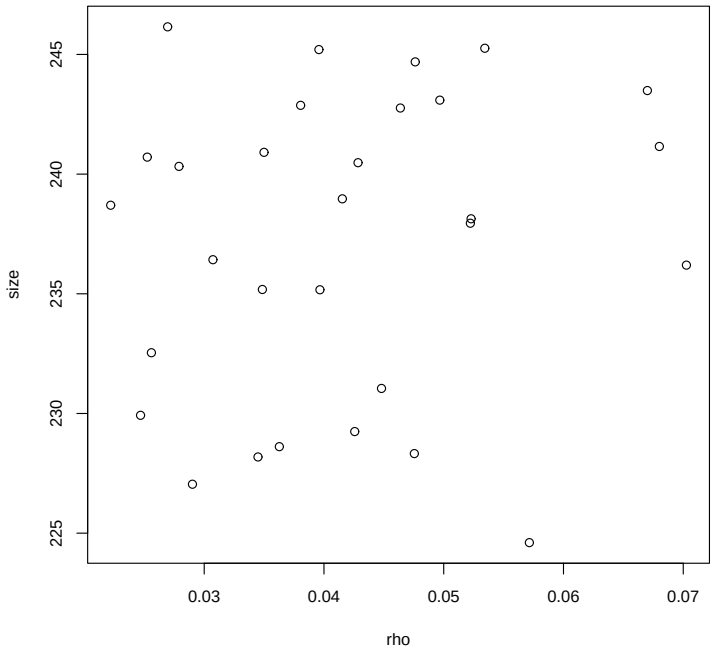

Figure 1: Plot of centroid size versus distance ρ to the mean, for the Female gorilla data.

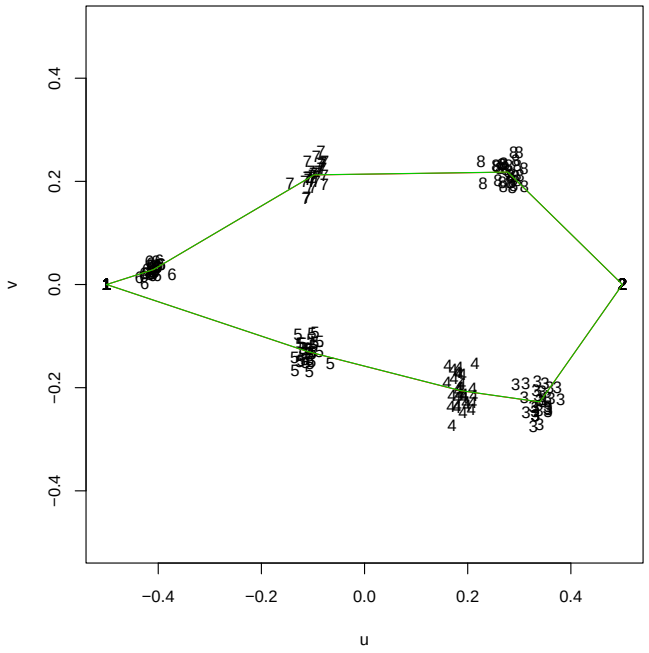

Figure 2: Scatter plot of Bookstein co-ordinates for the gorilla data, with the Bookstein mean superimposed in red and the full Procrustes mean in green.

```
> bsv$bshpv[,,17]
           [,1]        [,2]
[1,] -0.5000000  0.00000000
[2,]  0.5000000  0.00000000
[3,]  0.3361483 -0.21412477
[4,]  0.1737567 -0.18178705
[5,] -0.1068821 -0.13456027
[6,] -0.4026083  0.03255403
[7,] -0.1079997  0.20765362
[8,]  0.2691656  0.23211421

c(1:30)[ans$rho == max(ans$rho) ]
max(ans$rho) = 0.0702026451
```

The 22nd specimen is furthest from the mean in terms of rho

Question 2:

(a)

```
> male.eigen <- procGPA( gorm.dat , eigen2d=TRUE )
> female.eigen <- procGPA( gorf.dat , eigen2d=TRUE )
> male.gpa <- procGPA( gorm.dat , eigen2d=FALSE )
> female.gpa <- procGPA( gorf.dat , eigen2d=FALSE )
> riemdist(male.eigen$mshape, male.gpa$mshape )
[1] 0
> riemdist(female.eigen$mshape, female.gpa$mshape )
[1] 0

# yes they do give the same shape estimate

> male.eigen$percent
 [1] 4.220271e+01 1.796493e+01 1.237879e+01 7.053010e+00 5.236443e+00
 [6] 4.051309e+00 3.435191e+00 2.575969e+00 2.192304e+00 1.364635e+00
[11] 8.342580e-01 6.593374e-01 5.112160e-02 1.401730e-29 6.606944e-30
[16] 2.814214e-30
> male.gpa$percent
 [1] 4.220271e+01 1.796493e+01 1.237879e+01 7.053010e+00 5.236442e+00
 [6] 4.051310e+00 3.435188e+00 2.575968e+00 2.192304e+00 1.364634e+00
[11] 8.342578e-01 6.593374e-01 5.112161e-02 1.442065e-08 6.741667e-29
```

```
[16] 6.047677e-30

> female.eigen$percent
 [1] 3.479296e+01 2.290901e+01 1.125934e+01 8.841108e+00 6.062462e+00
 [6] 3.755552e+00 3.535783e+00 2.946377e+00 2.065423e+00 1.905920e+00
[11] 1.208520e+00 6.630631e-01 5.447876e-02 5.747008e-29 1.489944e-29
[16] 4.239026e-30
> female.gpa$percent
 [1] 3.479297e+01 2.290900e+01 1.125934e+01 8.841108e+00 6.062459e+00
 [6] 3.755554e+00 3.535782e+00 2.946375e+00 2.065424e+00 1.905921e+00
[11] 1.208521e+00 6.630634e-01 5.447879e-02 1.093015e-07 4.673833e-29
[16] 7.838134e-30
```

The percentages are almost identical (up to machine accuracy).

(b)

```
date()
for (i in 1:100){
ans <- procGPA( gorm.dat , eigen2d=FALSE )
}
date()
```

On my computer:

Females:

Eigenvector method : 5 seconds
Iterative method: 22 seconds

Males:

Eigenvector method : 5 seconds
Iterative method: 24 seconds

Hence, the eigenvector method is faster here.

Question 3.

One might guess that the full Procrustes mean is collinear, and in particular with Bookstein shape variables (0,0).

```
equ1 <- matrix(c(-0.5,0.5,0,0,0,sqrt(3)/2),3,2)
equ2 <- matrix(c(-0.5,0.5,0,0,0,-sqrt(3)/2),3,2)

equ<- array(0, c(3,2,2) )
equ[,,1] <- equu1
equ[,,2] <- equ2

ans <- procGPA( equ )

> ans$mshape
           [,1] [,2]
[1,] -0.4082483    0
[2,] -0.4082483    0
[3,]  0.8164966    0

collinear points, but not with Bookstein co-ordinates (0,0).

In fact the mean shape could be given by any collinear set of points.
Perhaps the easiest way to see this is to think of the data being
points on the sphere for triangle shapes. The two data points
are at the north and south pole of the sphere. The mean is given by any
triangle on the equator, i.e. any collinear shape.


Questions 4 and 5: see handwritten notes.


Question 6.

rho <- (0:100)*pi/2/100

dP <- sqrt(2) * sqrt( 1- cos(rho) )
dF <- sin(rho)

plot( rho, dP , type="l" , xlab="rho", ylab = "distance")
lines( rho, dF , type="l" , xlab="rho", ylab = "distance", lty=2)
```

See Figure 3 for the plot comparing the distances. Note d_P and ρ are very similar for most of the range. d_F is similar up to about 0.5, and then increases more slowly than d_P.

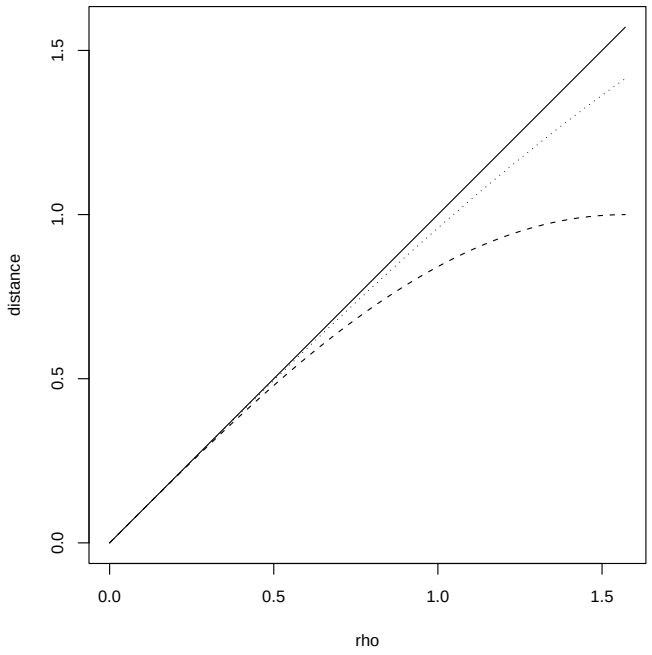

Figure 3: Plot of ρ vs ρ (solid), d_P versus ρ (dotted) and d_F versus ρ (dashed).