

## 1. Setting things up.

- (a)
- Solution:**
- The R codes for completing the above task are the following:

```
> data painters)
> # after looking at the data set, we know that the four ordinal columns are
> # the first four columns in that data set
> cov1 <- cov(painters[1:10, 1:4])
> mu1 <- c(11, 16, 8, 8)
> obs40 <- mvrnorm(n = 40, mu = mu1, Sigma = cov1)
> colnames(obs40) <- c("A", "B", "C", "D")
```

- (b)
- Solution:**
- The R codes for completing the above task are the following:

```
> mu2=c(8,9,16,4)
> tm=c(16.8, 9, 0.8, 6, 9, 11, 2.5, 5.7, 0.8, 2.5, 2.1, 0.6, 6, 5.7, 0.6, 5.7)
> cov2=matrix(tm,4,4,byrow=T)
> obs30 <- mvrnorm(n = 30, mu = mu2, Sigma = cov2)
> colnames(obs30) <- c("A", "B", "C", "D")
```

- (c)
- Solution:**
- The R code for completing the above task are the following. The estimated mean vector is given as the last output in the code below.

```
> library(MASS)
> X <- rbind(obs40, obs30)
> gr <- c(rep(0, 40), rep(1, 30))
> colmean <- apply(X, 1, mean)
> round(colmean, 2)
 [1] 10.50 14.30 13.93  3.61 13.26 10.62 11.33 10.78 11.59  6.96 13.97 12.93
[13]  8.10 12.94 14.26  6.92 13.98 13.03 15.56 16.33 11.69 12.82 12.31 13.07
[25] 10.05  7.60  5.05 10.58  9.76 12.49  5.16  9.01  9.48 10.33  8.51  2.40
[37] 11.66  5.61 11.19 15.36  9.97  9.16 11.96 12.05  7.09  9.28  5.14  8.30
[49] 10.42 11.18  8.57  8.76  7.40  8.81  4.60 11.36 10.72 11.54  7.66  9.76
[61]  5.83 10.46  9.95  8.51  5.47 10.36 11.61 12.29 11.22  7.99
```

- (d)
- Solution:**
- The R codes for completing the above task are the following. By calculating the means of the columns of
- $X_c$
- , we know that all manipulation gives the desired result.

```
> Xc <- scale(X, scale = F)
> round(apply(Xc, 2, mean), 3)
A B C D
0 0 0 0
```

- (e)
- Solution:**
- First, we should note that given
- $Y$
- and
- $D = \frac{1}{n}I$
- , we have
- $P = Y(Y'DY)^{-1}Y'D = Y(Y'Y)^{-1}Y$
- . Hence, we don't need to worry about
- $D$
- anymore. The group membership matrix
- $Y$
- is a
- $70 \times 2$
- matrix, with each row
- $(0, 1)$
- if it belongs to the first group and
- $(1, 0)$
- otherwise. With this group membership matrix, we have the total variance
- $T = X_c'X_c$
- since
- $X_c$
- is centered data matrix, the between group variance
- $B = X_c'PX_c$
- and the within group variance
- $W = X_c'(I - P)X_c$
- . The key point in obtaining the above formulae is the observation that

$$Y = \begin{pmatrix} \frac{1}{40}I_{40} & 0 \\ 0 & \frac{1}{30}I_{30} \end{pmatrix}.$$

The R codes for completing the above task are the following:

```

> Y <- matrix(c(gr, -(gr-1)), ncol = 2)
> P <- Y %>% solve(t(Y) %>% Y) %>% t(Y)
> T <- t(Xc) %>% Xc
> I <- diag(1, 70)
> W <- t(Xc) %>% (I - P) %>% Xc
> B <- t(Xc) %>% P %>% Xc
> round(T - (W + B), 2)
  A B C D
A 0 0 0 0
B 0 0 0 0
C 0 0 0 0
D 0 0 0 0

```

## 2. The lda function.

**Solution:** The R codes for completing the above task are the following. The first linear discriminant function explains the between group variance in the total variance. The coefficients of the linear discriminant direction is  $(-0.0329, -0.2865, 0.1541, 0.1172)'$ .

```

> ld <- lda(Xc, gr)
> ld
Call:
lda(Xc, grouping = gr)

Prior probabilities of groups:
      0      1
0.5714286 0.4285714

Group means:
      A      B      C      D
0  1.607663  2.905604 -3.132814  1.155055
1 -2.143551 -3.874139  4.177085 -1.540074

Coefficients of linear discriminants:
      LD1
A -0.03287111
B -0.28653572
C  0.15407874
D  0.11722492

```

## 3. The svd function.

**Solution:** The singular values of the centered data matrix are 69.83, 38.92, 20.83 and 15.77. They are the square roots of the 4 eigenvalues of the matrix  $X_c'X_c$ . The R codes for completing the task are the following:

```

> XcSVD <- svd(Xc)
> XcSVD$d
[1] 69.82922 38.91741 20.83014 15.77625

```

## 4. Question 1

**Solution:** First of all, we compute the discriminant analysis in `ade4` package as the following and compare the result with that we obtained via `lda`. We can see that the two procedures give the same result up to a scale factor.

```

> library(ade4)
> pca <- dudi.pca(Xc, scale = F, scannf = F, nf = 4)

```

```
> dudilda <- discrimin(pca, as.factor(gr), scannf = F, nf = 4)
> dudilda$fa ## result obtained using ade4
      DS1
A -0.01898603
B -0.16550025
C  0.08899439
D  0.06770798
> ld$scaling ## result obtained using lda
      LD1
A -0.03287111
B -0.28653572
C  0.15407874
D  0.11722492
```

## Question 2

**Solution:** The list output of a call to `dudi.pca` contains the following items: `tab`, `cw`, `lw`, `eig`, `rank`, `nf`, `c1`, `l1`, `co`, `li`, `call`, `cent`, and `norm`. `rank` is the rank of the data matrix; `cw` is the vector of column weights; `center` is the mean vector (in  $p$ -dim) of rows in the  $n \times p$  data matrix and `li` is the principal components.