```
#LinearDiscriminantAnalysis
 x=read.table('T6-10.dat',header=FALSE)
 x=read.table('T6-
10.dat',header=FALSE,col.names=c("Fuel","Repair","Capital","Type"))
 library(MASS)
 fit_lda=lda(x[,4]~x[,1]+x[,2]+x[,3])
 fit_lda
 #Butthecutoffpointcouldnotgethere
 x=read.table('T6-10.dat',header=FALSE)
 group1=x[x[,4]=='gasoline',][,1:3]
 group2=x[x[,4]=='diesel',][,1:3]
 n1=nrow(group1)
 n2=nrow(group2)
 group1.means=apply(group1,2,mean)
 group2.means<-apply(group2,2,mean)
 w1=(n1-1)*var(group1)
 w2=(n2-1)*var(group2)
 sp1<-1/(n1+n2-2)*(w1+w2)
 cutoff=.5*(group1.means-group2.means)%*%solve(sp1)%*%(group1.means+
 group2.means)
 cutoff
 x0=matrix(c(20,10,19),3,1,byrow=T)
 y_hat=(group1.means-group2.means)%*%solve(sp1)%*%x0
 if (y_hat>=cutoff){
 predict_class='gasoline'
 print(predict_class)
 }else{predict_class='diesel'
 print(predict_class)}
 #CanonicalCorrelation
 #Example10.1
 R11=matrix(c(1,.4,.4,1),byrow=T,nrow=2)
 R22=matrix(c(1,.2,.2,1),byrow=T,nrow=2)
 R12=matrix(c(.5,.6,.3,.4),byrow=T,nrow=2)
 R21=t(R12)
 R11m12=((eigen(R11)$values[1])^(-
1/2))*eigen(R11)$vectors[,1]%*%t(eigen(R11)$vectors[,
 1])+((eigen(R11)$values[2])^(-
1/2))*eigen(R11)$vectors[,2]%*%t(eigen(R11)$vectors[,2])
 R22m12=((eigen(R22)$values[1])^(-
1/2))*eigen(R22)$vectors[,1]%*%t(eigen(R22)$vectors[,
 1])+((eigen(R22)$values[2])^(-
1/2))*eigen(R22)$vectors[,2]%*%t(eigen(R22)$vectors[,2])
solve(R11)
 solve(R22)
 A=R11m12%*%R12%*%solve(R22)%*%R21%*%R11m12
```

```r
e1=eigen(A)$vectors[,1]
e2=eigen(A)$vectors[,2]
CCor_1st=sqrt(eigen(A)$values[1])##0.7387273
CCor_2nd=sqrt(eigen(A)$values[2])##0.03014884
B=R22m12%*%R21%*%solve(R11)%*%R12%*%R22m12
f1=eigen(B)$vectors[,1]
f2=eigen(B)$vectors[,2]
##1stPairofCanonicalvariables
a1=t(e1)%*%R11m12##U1=-0.8559647*Z1_1-0.2777371*Z2_1
b1=t(f1)%*%R22m12##V1=0.5448119*Z1_2+0.7366455*Z2_2
##1stPairofCanonicalvariables
a2=t(e2)%*%R11m12##U2=0.6766097*Z1_1-1.055148*Z2_1
b2=t(f2)%*%R22m12##V2=-0.863045*Z1_2+0.7064135*Z2_2
#####Example10.5
R11=matrix(c(1.00,0.49,0.53,0.49,0.51,0.49,1,0.57,0.46,0.53,0.53,0.57,1,0.48,0.57,.
49,.46,.48,1,.57,
.51, .53,.57,.57,1),5,5,byrow=T)
R21=matrix(c(0.33,0.30,0.31,0.24,0.38,0.32,0.21,0.23,0.22,0.32,0.20,0.16,0.14,0.12,0.17,
0.19,0.08,0.07,0.19,0.23,0.30,0.27,0.24,0.21,0.32,0.37,0.35,0.37,0.29,0.36,0.21,0.20,
0.18,0.16,0.27),7,5,byrow=T)
R12=t(R21)
R22=matrix(c(1,.43,.27,.24,.34,.37,.40,.43,1,.33,.26,.54,.32,.58,.27,.33,1,.25,.46,.29,.45,.
24,.26,.25,1,.28,.30,.27,.34,.54,.46,.28,1,.35,.59,.37,.32,.29,.30,.35,1,.31,0.40,.58,.45,.27,.59,.3
1,1.00),7,7,byrow=T)
#CalculatingR11^(-1/2)
emat=eigen(R11)
R11_12=emat$vectors%*%diag(emat$values^(-1/2))%*%t(emat$vectors)
#CalculatingR22^(-1/2)
emat2=eigen(R22)
R22_12=emat2$vectors%*%diag(emat2$values^(-1/2))%*%t(emat2$vectors)
#FindingtheE1andE2matrices:
E1=R11_12%*%R12%*%solve(R22)%*%R21%*%R11_12
E2=R22_12%*%R21%*%solve(R11)%*%R12%*%R22_12
eigen(E1);eigen(E2)
#Thecanonicalcorrelationsare:
canon.corr<-sqrt(eigen(E1)$values)
canon.corr
#Firstpairofcanonicalvariatesare
a1=eigen(E1)$vectors[,1]%*%R11_12
b1=eigen(E2)$vectors[,1]%*%R22_12
```

```
#2ndpairofcanonicalvariatesare
a2=eigen(E1)$vectors[,2]%*%R11_12
b2=eigen(E2)$vectors[,2]%*%R22_12
A=rbind(a1,a2)
B=rbind(b1,b2)
p=nrow(R11)
q=nrow(R22)
###SamplecorrelationofUandX1
R_UX1=a1%*%R11%*%diag(p)
R_UX1
####FActorAnalysis
#Createascatterplotmatrixofthestandardizeddata
x=read.table('T8-4.dat',header=FALSE)
x=read.table('T8-
4.dat',header=FALSE,col.names=c("Allied","Dupont","Carbide","Exxon",
"Texaco"))
x_x<-scale(x,center=T,scale=T)
pairs(x_x,labels=c("Allied","Dupont","Carbide","Exxon",
"Texaco"),panel=function(x,y){panel.smooth(x,y)
abline(lsfit(x,y),lty=2)})
x=read.table('T8-4.dat',header=FALSE)
x_x=scale(x,center=T,scale=T)
s.cor=var(x_x)
R=cor(x)###=cov(x_x)
##LoadingsestimatedbyPrincipleComponent
L1=sqrt(eigen(R)$values[1])*eigen(R)$vectors[,1]
L2=sqrt(eigen(R)$values[2])*eigen(R)$vectors[,2]
L=cbind(L1,L2)
Q=1-(L[,1]^2+L[,2]^2)#SpecificVariances
####Testforfactorloadings
n=100
p=5
m=2
det(L%*%t(L)+diag(Q))
det(R)
cal_value=(n-1-(2*p+4*m+5)/6)*log(det(L%*%t(L)+diag(Q))/det(R))
df=((p-m)^2-p-m)/2
cv=qchisq(.95,1)##3.841459
cv1=qchisq(.975,1)###5.023886
cv2=qchisq(.025,1)###0.0009820691
###FactoranalysisusingPCA
fac=factanal(x,factors=1,method='PCA',scale=T,center=T)
est_sigma=(fac$loadings[,1])%*%t(fac$loadings[,1])+diag(fac$uniquenesses)
###FactoranalysisusingPCAandVarimaxrotation
fac1=factanal(x,factors=2,method='PCA',scale=T,center=T,rotation="varimax")
```

```
fac1
####FactoranalysisusingMLE
fac2=factanal(x,factors=2,method='mle',scale=T,center=T)
fac2
####FactoranalysisusingcorrelationmatrixR
fac3=factanal(covmat=R,factors=2,method='mle',scale=T,center=T)
fac3
####Factorscoresusingweightedleast-squares:needargument(scores="Bartlett")
fac4=factanal(x,factors=2,method='mle',scale=T,center=T,scores="Bartlett")
fac4
####FactorscoresusingRegression:needargument(scores="regression")
fac5<-factanal(x,factors=2,method='pc',scale=T,center=T,scores="regression")
boxplot(fac5$scores)
#PrincipalComponents
x=read.table('T6-9.dat',header=FALSE)
x
turtles=log(x[25:48,1:3])
xbar=colMeans(turtles)
S=cov(turtles)
R=cor(turtles)
fit<-prcomp(S) #princomp(S)
summary(fit)
screeplot(fit)
screeplot(fit,npcs=3,type="lines")
PC1_S=-0.7616419*turtles[,1]-0.4550873*turtles[,2]-0.4612996*turtles[,3]
fit2=prcomp(R)
fit2
PC1_R=-0.1306064*turtles[,1]-0.6913349*turtles[,2]+0.71063216*turtles[,3]
boxplot(cbind(PC1_S,PC1_R))
#####Example8.9
g_curle=(4-1)^2*(1-(1-.6855)^2)/(4-(4-2)*(1-.6855)^2)
T_test=((150-1)/(1-.6855)^2)*(((.7501-.6855)^2+(.6329-.6855)^2+(.6363-
.6855)^2+(.6925-.
6855)^2+(.7386-.6855)^2+(.6625-.6855)^2)-g_curle*((.6731-.6855)^2+(.7271-
.6855)^2+(.6
626-.6855)^2+(.6791-.6855)^2))
qchisq(.95,5)
#Comparisionofseveralmeans
#Example6.8
data1=matrix(c(9,6,9,0,2,3,1,2),ncol=1)
data2=matrix(c(3,2,7,4,0,8,9,7),ncol=1)
group=c(rep(1,3),rep(2,2),rep(3,3))
group=as.factor(group)
mm1=manova(cbind(data1,data2)~group)
mmFull=summary(mm1)
```

```r
mmFull$SS[1]
B=mmFull$SS[1]#MANOVASSforgroup
W=mmFull$SS[2]#MANOVASSforResiduals
B=matrix(c(78,-12,-12,48),byrow=T,nrow=2)
W=matrix(c(10,1,1,24),byrow=T,nrow=2)
B+W
LS=det(W)/det(B+W)
2*(1-sqrt(LS))/sqrt(LS)##8.19886
CV=qf(.95,4,8)
CV##3.837853
summary(mm1,test='Wilks')
summary(mm1,test='Roy')
summary.aov(mm1)
####ProfileAnalysis
x1_bar=matrix(c(6.833,7.033,3.967,4.700),nrow=4)
x2_bar=matrix(c(6.633,7.000,4.000,4.533),nrow=4)
Sp=matrix(c(.606,.262,.066,.161,.262,.637,.173,.143,.066,.173,.810,.029,.161,.14
3,.029,.306
),byrow=T,nrow=4)
C=matrix(c(-1,1,0,0,0,-1,1,0,0,0,-1,1),byrow=T,nrow=3)
###TestforParallelProfiles
T2=t(x1_bar-x2_bar)%*%t(C)%*%solve((1/30+1/30)*C%*%Sp%*%t(C))%*%C%*%(x1_bar-x2_
bar)
n1=30
n2=30
p=4
n1+n2-p
c2=((n1+n2-2)*(p-1)/(n1+n2-p))*qf(.95,df1=3,df2=56)
c2
c2=((n1+n2-2)*(p-1)/(n1+n2-p))*qf(.975,df1=3,df2=56)
c2
c21=((n1+n2-2)*(p-1)/(n1+n2-p))*qf(.025,df1=3,df2=56)
##ForCoincidentProfile
oneM=matrix(c(1,1,1,1),nrow=4,ncol=1)
T2=t(oneM)%*%(x1_bar-x2_bar)%*%solve((1/30+1/30)*t(oneM)%*%Sp%*%oneM)%*%t(one
M)%*%(x1_bar-x2_bar)
T2
c2=qf(.975,df1=1,df2=58)
c21=qf(.025,df1=1,df2=58)
####ForLevelTest
X_bar=(x1_bar-x2_bar)
T2=(n1+n2)*t(X_bar)%*%t(C)%*%solve(C%*%Sp%*%t(C))%*%C%*%X_bar
T2##4.021441
C2=((n1+n2-1)*(p-1))/(n1+n2-p+1)*qf(0.975,3,57)##10.41811
C2=((n1+n2-1)*(p-1))/(n1+n2-p+1)*qf(0.025,3,57)##0.2218554
```

```r
HotellingT2
###Women'snutritiondata
mu0=matrix(c(1000,15,60,800,75),nrow=5,byrow=T)
X_bar=matrix(c(624,11.1,65.8,839.6,78.9),nrow=5,byrow=T)
S=matrix(c(157829.4,940.1,6075.8,102411.1,6701.6,940.1,35.8,114.1,2383.2,137.7,6
075.8,
114.1,934.9,7330.1,477.2,102411.1,2383.2,7330.1,2668452.4,22063.3,6701.6,137.7,
477.2,22063.3,5416.3),nrow=5,byrow=T)
n=737
p=5
T2=n*t(X_bar-mu0)%*%solve(S)%*%(X_bar-mu0)
(n-1)*p/(n-p)
##.05/2=.025
##1-.05/2=1-.025=.975
qf(.975,p,(n-p))#2.5838Upper-tail
qf(.025,p,(n-p))##0.1659967Lower-tail
Lcv=((n-1)*p/(n-p))*qf(.025,p,(n-p))
Ucv=((n-1)*p/(n-p))*qf(.975,p,(n-p))
##ifT2<LcvorT2>Ucv2,thenH0willberejected
if (T2<Lcv|T2>Ucv)print("H0isrejected")elseprint("H0isnotrejected")
##T2=1768.458,Ucv=12.98959
#HotellingT2:Example5.2
X=read.table('T5-1.dat',header=FALSE)
mu0=matrix(c(4,50,10),nrow=3,ncol=1)
xbar=colMeans(X)
xbar=matrix(xbar,nrow=3,ncol=1)
S=cov(X)
S_inv=solve(S)
n=nrow(X)
p=ncol(X)
T2=n*t(xbar-mu0)%*%S_inv%*%(xbar-mu0)##T2=9.738773
##alpha=.10
Lcv=((n-1)*p/(n-p))*qf(.05,p,(n-p))
Ucv=((n-1)*p/(n-p))*qf(.95,p,(n-p))
##ifT2<LcvorT2>Ucv2,thenH0willberejected
if (T2<Lcv|T2>Ucv)print("H0isrejected")elseprint("H0isnotrejected")
##T2=9.738773,Ucv=10.7186,Lcv=0.3861545
#Q-Qplot
x=read.table("T4-1.dat",header=F)
x
dim(x)
class(x)
dim(t(t(x)))
x=t(t(x))
xj=sort(x,decreasing=FALSE)
```

```r
xj
n=length(xj)
n
j=matrix(seq(1,n),nrow=n,ncol=1)
j
qj=qnorm((j-0.5)/n)
qj
plot(qj,xj)
###Chi-squarePlot
x=read.table("T4-3.dat",header=F)
n=length(x[,1])
x=matrix(x,)
S=cov(x)
solve(S)
x_bar=colMeans(x)
x_bar=matrix(colMeans(x),nrow=1,ncol=5)
d_sq=matrix(0,nrow=n,ncol=1)
for(iin1:n){
d_sq[i]=(x[i,]-x_bar)%*%solve(S)%*%t(x[i,]-x_bar)
}
t(x[1,]-colMeans(x))%*%solve(S)%*%(x[1,]-colMeans(x))
A=(x[1,]-x_bar)
B=solve(S)
C=(t(x[1,]-x_bar))
A%*%B%*%C
dim(x_bar)
dim(x[1,])
###ThecorrectedCodeneedsas.matrixfunctiontoconvertadata.frameasamatrix
###Chi-squarePlot
x=read.table("T4-3.dat",header=F)
n=length(x[,1])
S=cov(x)
solve(S)
x_bar=colMeans(x)
x_bar=matrix(colMeans(x),nrow=1,ncol=5)
d_sq=matrix(0,nrow=n,ncol=1)
for(iin1:n){
d_sq[i]=as.matrix(x[i,]-x_bar)%*%solve(S)%*%t(x[i,]-x_bar)
}
d_sq
###Or
###Chi-squarePlot
x=read.table("T4-3.dat",header=F)
x=as.matrix(x)
n=length(x[,1])
```

```r
S=cov(x)
solve(S)
x_bar=colMeans(x)
x_bar=matrix(colMeans(x),nrow=1,ncol=5)
d_sq=matrix(0,nrow=n,ncol=1)
for(iin1:n){
d_sq[i]=(x[i,]-x_bar)%*%solve(S)%*%t(x[i,]-x_bar)
}
d_sq
#Q-Qplot
x=read.table("T4-1.dat",header=F)
x=as.matrix(x)
xj=sort(x,decreasing=FALSE)
xj
n=length(xj)
n
j=matrix(seq(1,n),nrow=n,ncol=1)
j
qj=qnorm((j-0.5)/n)
qj
plot(qj,xj)
rQ=cor(xj,qj)##0.9279049
Tab_value=0.9737
###Chi-squarePlot
x=read.table("T4-3.dat",header=F)
x=as.matrix(x)
n=length(x[,1])
S=cov(x)
solve(S)
x_bar=colMeans(x)
x_bar=matrix(colMeans(x),nrow=1,ncol=5)
d_sq=matrix(0,nrow=n,ncol=1)
for(iin1:n){
d_sq[i]=(x[i,]-x_bar)%*%solve(S)%*%t(x[i,]-x_bar)
}
d_sq
dj=sort(d_sq,decreasing=FALSE)
dj
j=matrix(seq(1,n),ncol=1,nrow=n)
qj=qchisq((j-0.5)/n,5)
qj
plot(qj,dj)
###ellipse
x=read.table("T4-3.dat",header=F)
x=as.matrix(x)
```

```r
n=length(x[,1])
Y=x[,1:2]
S=cov(Y)
solve(S)
Y_bar=colMeans(Y)
Y_bar=matrix(colMeans(Y)
A=matrix(c(2,1,0,1,3,3,0,3,8),byrow=T,nrow=3)
eigen(A)$values
eigen(A)$vectors[,1]
##spectraldecomposition
eigen(A)$values[1]*eigen(A)$vectors[,1]%*%t(eigen(A)$vectors[,1])+
eigen(A)$values[2]*eigen(A)$vectors[,2]%*%t(eigen(A)$vectors[,2])+
eigen(A)$values[3]*eigen(A)$vectors[,3]%*%t(eigen(A)$vectors[,3])
print(A)
##FindA^(1/2)usingspectraldecomposition
A^(1/2)
(eigen(A)$values[1])^(1/2)*eigen(A)$vectors[,1]%*%t(eigen(A)$vectors[,1])+
(eigen(A)$values[2])^(1/2)*eigen(A)$vectors[,2]%*%t(eigen(A)$vectors[,2])+
(eigen(A)$values[3])^(1/2)*eigen(A)$vectors[,3]%*%t(eigen(A)$vectors[,3])
```