

STAD37A3

Yulun Wu

21/11/2022

Q1

```
pp = read.table("profilePotato.txt", header=TRUE)
head(pp)
```

```
## texture flavor moist cookmeth
## 1      2.9      3.2   3.0         1
## 2      1.8      3.0   1.7         1
## 3      1.8      2.6   1.5         1
## 4      2.6      3.1   2.4         1
## 5      3.1      3.0   2.8         1
## 6      1.8      2.6   1.8         1
```

```
# Partition data into two group based on cookmeth
```

```
method1 = subset(pp,cookmeth==1,select=c("texture","flavor","moist"))
method2 = subset(pp,cookmeth==2,select=c("texture","flavor","moist"))
head(method1)
```

```
## texture flavor moist
## 1      2.9      3.2   3.0
## 2      1.8      3.0   1.7
## 3      1.8      2.6   1.5
## 4      2.6      3.1   2.4
## 5      3.1      3.0   2.8
## 6      1.8      2.6   1.8
```

```
head(method2)
```

```
## texture flavor moist
## 33      1.9      2.8   2.2
## 34      2.0      2.8   1.9
## 35      2.5      3.0   2.1
## 36      3.0      2.6   2.7
## 37      1.9      2.9   2.0
## 38      2.0      3.0   2.0
```

```
p = ncol(method1)
n1 = nrow(method1)
xbar1 = colMeans(method1) # Sample mean of method 1
xbar1
```

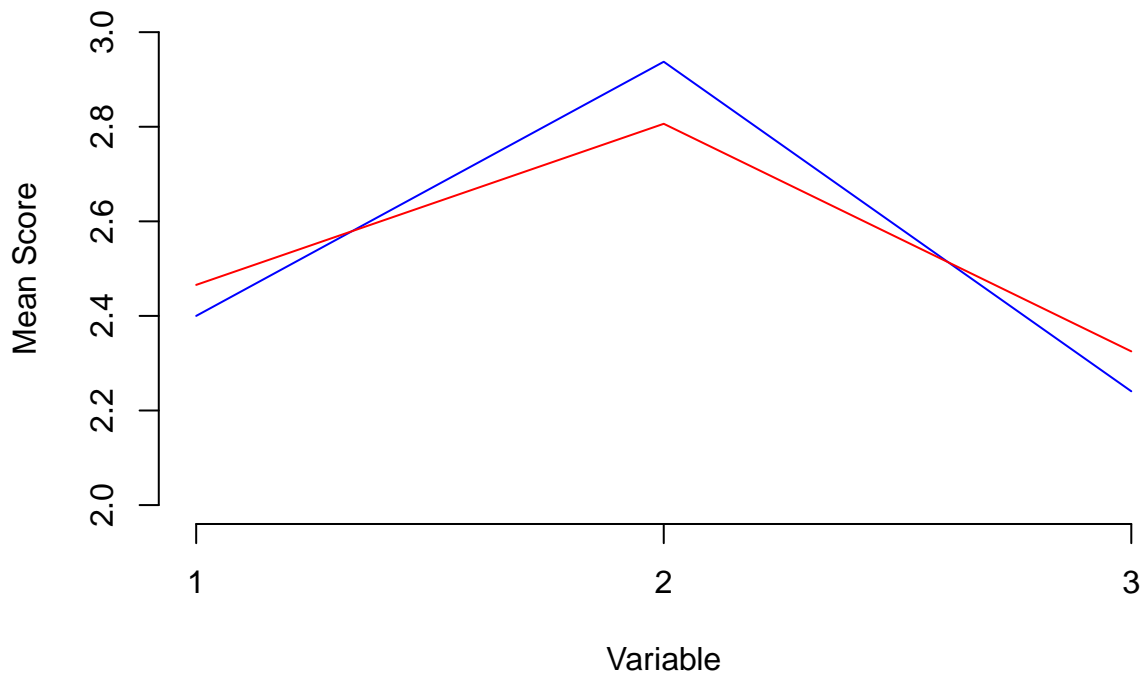
```
## texture flavor moist
## 2.400000 2.937500 2.240625
```

```
n2 = nrow(method2) # Sample mean of method 2
xbar2 = colMeans(method2)
xbar2
```

```
## texture flavor moist
## 2.465625 2.806250 2.325000
```

(a)

```
# Profile plot
Variable = c(1,2,3) # "texture","flavor","moist"
plot(Variable,xbar1,ylim=range(c(2,3)),type="l",col="blue",ylab="Mean Score",axes=FALSE) # All values of
axis(side = 1,at = Variable)
axis(side = 2)
lines(Variable,xbar2,col="red")
```



From profile plot, I can see that the two lines are not parallel but in similar shape and similar to coincident, therefore the two method maybe profile parallel and coincident, but clearly not level.

(b)

```
S1 = cov(method1) # Sample covariance of method 1
S1
```

```
## texture flavor moist
## texture 0.24774194 0.03354839 0.19612903
## flavor 0.03354839 0.08435484 0.02552419
## moist 0.19612903 0.02552419 0.21023185
```

```
S2 = cov(method2) # Sample covariance of method 2
S2
```

```
## texture flavor moist
## texture 0.2565222 0.03312500 0.15830645
```

```
## flavor 0.0331250 0.09415323 0.02758065
## moist 0.1583065 0.02758065 0.14322581
```

```
Sp = (1/(n1+n2-2))*((n1-1)*S1+(n2-1)*S2) # S_pooled
Sp
```

```
##          texture      flavor      moist
## texture 0.25213206 0.03333669 0.17721774
## flavor 0.03333669 0.08925403 0.02655242
## moist 0.17721774 0.02655242 0.17672883
```

```
C = matrix(c(-1,1,0,0,-1,1),byrow=TRUE,2,3)
C
```

```
##      [,1] [,2] [,3]
## [1,]  -1   1   0
## [2,]   0  -1   1
```

```
Cxbar = C%*(xbar1-xbar2) # C(xbar1-xbar2)
```

```
# (1)Test parallel profile
```

```
T_sq_parallel = (((1/n1)+(1/n2))^(-1))*t(Cxbar)%*%solve(C%*%Sp%*%t(C))%*%Cxbar
T_sq_parallel # test statistics for testing parallel
```

```
##      [,1]
## [1,] 3.527515
```

```
alpha = 0.05
```

```
cv_parallel = (((n1+n2-2)*(p-1))/(n1+n2-p))*qf(1-alpha,p-1,n1+n2-p)
cv_parallel # critical value for testing parallel
```

```
## [1] 6.398789
```

Since $3.527515 < 6.398789$, we fail to reject the null hypothesis that the two methods are profile parallel at level $\alpha = 0.05$.

```
S = cov(pp[,1:3]) # Sample covariance for whole dataset regardless method
S
```

```
##          texture      flavor      moist
## texture 0.24922371 0.03062004 0.17581101
## flavor 0.03062004 0.09221230 0.02331845
## moist 0.17581101 0.02331845 0.17573165
```

```
xbar = colMeans(pp[,1:3]) # Sample mean for whole dataset regardless method
xbar
```

```
## texture flavor moist
## 2.432812 2.871875 2.282812
```

```
# (2)Test coincident profile
```

```
one = as.matrix(c(rep(1,p)))
one
```

```
##      [,1]
## [1,]   1
## [2,]   1
## [3,]   1
```

```
T_sq_coincident = (((1/n1)+(1/n2))^(-1))*t(one)%*(xbar1-xbar2)%*%solve(t(one)%*%Sp%*%one)%*%t(one)%*(
T_sq_coincident # for testing coincident
```

```
##           [,1]
## [1,] 0.005668485
```

```
alpha = 0.05
cv_coincident = qf(1-alpha,1,n1+n2-2)
cv_coincident # for testing coincident
```

```
## [1] 3.995887
```

Since $0.005668485 < 3.995887$, we fail to reject the null hypothesis that the two methods are profile coincident at level $\alpha = 0.05$.

```
# (3)Test level profile
S = cov(pp[,1:3]) # Sample covariance for whole dataset regardless method
S
```

```
##           texture      flavor      moist
## texture 0.24922371 0.03062004 0.17581101
## flavor  0.03062004 0.09221230 0.02331845
## moist   0.17581101 0.02331845 0.17573165
```

```
xbar = colMeans(pp[,1:3]) # Sample mean for whole dataset regardless method
xbar
```

```
## texture flavor moist
## 2.432812 2.871875 2.282812
```

```
C = matrix(c(-1,1,0,0,-1,1),byrow=TRUE,2,3)
C
```

```
##           [,1] [,2] [,3]
## [1,]    -1     1     0
## [2,]     0    -1     1
```

```
T_sq_level = (n1+n2)*t(C%*%xbar)%*%solve(C%*%S%*%t(C))%*%(C%*%xbar)
T_sq_level # for testing level
```

```
##           [,1]
## [1,] 115.3216
```

```
alpha = 0.05
cv_level = (((n1+n2-1)*(p-1))/(n1+n2-p+1))*qf(1-alpha,p-1,n1+n2-p+1)
cv_level # for testing level
```

```
## [1] 6.391977
```

Since $115.3216 > 6.391977$, we reject the null hypothesis that the two methods are profile level at level $\alpha = 0.05$.

Q2

```
es = read.table("2sample_data.txt", header=TRUE)
head(es)
```

```
##      x1  x2  x3 x4 Period
## 1 131 138  89 49      1
## 2 125 131  92 48      1
## 3 131 132  99 50      1
## 4 119 132  96 44      1
```

```
## 5 136 143 100 54      1
## 6 138 137  89 56      1
```

```
# Partition data into two group based on Period
```

```
period1 = subset(es,Period==1,select=c("x1","x2","x3", "x4"))
period2 = subset(es,Period==2,select=c("x1","x2","x3", "x4"))
head(period1)
```

```
##      x1  x2  x3 x4
## 1 131 138  89 49
## 2 125 131  92 48
## 3 131 132  99 50
## 4 119 132  96 44
## 5 136 143 100 54
## 6 138 137  89 56
```

```
head(period2)
```

```
##      x1  x2  x3 x4
## 31 125 136  99 46
## 32 134 132  95 46
## 33 139 132  96 43
## 34 149 127 102 49
## 35 127 122  93 43
## 36 136 134  96 50
```

```
p = ncol(period1)
p
```

```
## [1] 4
```

```
n = nrow(es)
n
```

```
## [1] 60
```

```
n1 = nrow(period1)
n1
```

```
## [1] 30
```

```
n2 = nrow(period2)
n2
```

```
## [1] 30
```

(a)

```
xbar1 = colMeans(period1) # Observed sample mean of period 1
xbar1
```

```
##      x1      x2      x3      x4
## 131.36667 133.60000  99.16667  50.53333
```

```
xbar2 = colMeans(period2) # Observed sample mean of period 2
xbar2
```

```
##      x1      x2      x3      x4
## 133.36667 130.70000  97.06667  48.23333
```

(b)

```
S1 = cov(period1)
S1 # Observed sample covariance of period 1

##           x1           x2           x3           x4
## x1 26.309195  4.1517241  0.4540230  7.2459770
## x2  4.151724 19.9724138 -0.7931034  0.3931034
## x3  0.454023 -0.7931034 34.6264368 -1.9195402
## x4  7.245977  0.3931034 -1.9195402  7.6367816

S2 = cov(period2)
S2 # Observed sample covariance of period 2

##           x1           x2           x3           x4
## x1 23.136782  1.010345  4.7678161  1.8425287
## x2  1.010345 21.596552  3.3655172  5.6241379
## x3  4.767816  3.365517 18.8919540  0.1908046
## x4  1.842529  5.624138  0.1908046  8.7367816

Sp = (1/(n1+n2-2))*((n1-1)*S1+(n2-1)*S2) # S_pooled
Sp

##           x1           x2           x3           x4
## x1 24.722989  2.581034  2.6109195  4.5442529
## x2  2.581034 20.784483  1.2862069  3.0086207
## x3  2.610920  1.286207 26.7591954 -0.8643678
## x4  4.544253  3.008621 -0.8643678  8.1867816
```

(c)

```
# Box's M test to test for equality of the two p opulation covariance
# matrices
library(biotools)

## Loading required package: MASS

## ---
## biotools version 4.2

boxM(es[,1:4],es[, "Period"])
```

```
##
## Box's M-test for Homogeneity of Covariance Matrices
##
## data: es[, 1:4]
## Chi-Sq (approx.) = 10.143, df = 10, p-value = 0.428
```

Since $p_value = 0.428 > 0.05$, we fail to reject the null hypothesis that the two p opulation covariance matrices are equal at level $\alpha = 0.05$.

(d)

```
# Assume sigma1 = sigma2
delta0 = c(0,0,0,0)
# Test statistics
```

```
T_sq = t(xbar1-xbar2 - delta0)%*%solve(((1/n1)+(1/n2))*Sp)%*(xbar1-xbar2 - delta0)
T_sq
```

```
##           [,1]
## [1,] 24.59121
```

```
criticalValue = (((n1+n2-2)*p)/(n1+n2-p-1))*qf(1-0.05,p,n1+n2-p-1) # Critical Value
criticalValue
```

```
## [1] 10.71287
```

```
F_level = T_sq/(((n1+n2-2)*p)/(n1+n2-p-1)) # F value
F_level
```

```
##           [,1]
## [1,] 5.829813
```

```
p_value = 1 - pf(F_level,p,n1+n2-p-1) # P value of the test
p_value
```

```
##           [,1]
## [1,] 0.0005518371
```

Since $p_value = 0.0005518371 < 0.05$, we reject the null hypothesis that the mean vectors are equal at level $\alpha = 0.05$.

Find the linear combination most responsible for the rejection

```
lc_a = solve(Sp)%*(xbar1-xbar2)
```

lc_a # This is the linear combination most responsible for the rejection of null hypothesis

```
##           [,1]
## x1 -0.1658937
## x2  0.1038767
## x3  0.1008312
## x4  0.3454951
```

(e)

```
c = sqrt(criticalValue) # Square root of critical value
c
```

```
## [1] 3.273052
```

```
xbar1-xbar2
```

```
##   x1   x2   x3   x4
## -2.0  2.9  2.1  2.3
```

SCI for the components of $\mu_1 - \mu_2$

Test $\mu_{11} - \mu_{21} = 0$

```
SCI_LL1 = (xbar1[1] - xbar2[1]) - c*sqrt(((1/n1)+(1/n2))*Sp[1,1])
```

```
SCI_UL1 = (xbar1[1] - xbar2[1]) + c*sqrt(((1/n1)+(1/n2))*Sp[1,1])
```

```
SCI1 = c(SCI_LL1,SCI_UL1)
```

```
SCI1
```

```
##           x1           x1
## -6.202016  2.202016
```

Test $\mu_{12} - \mu_{22} = 0$

```
SCI_LL2 = (xbar1[2] - xbar2[2]) - c*sqrt(((1/n1)+(1/n2))*Sp[2,2])
```

```

SCI_UL2 = (xbar1[2] - xbar2[2]) + c*sqrt(((1/n1)+(1/n2))*Sp[2,2])
SCI2 = c(SCI_LL2,SCI_UL2)
SCI2

```

```

##          x2          x2
## -0.9528036  6.7528036

```

```

# Test mu13 - mu23 = 0
SCI_LL3 = (xbar1[3] - xbar2[3]) - c*sqrt(((1/n1)+(1/n2))*Sp[3,3])
SCI_UL3 = (xbar1[3] - xbar2[3]) + c*sqrt(((1/n1)+(1/n2))*Sp[3,3])
SCI3 = c(SCI_LL3,SCI_UL3)
SCI3

```

```

##          x3          x3
## -2.271634  6.471634

```

```

# Test mu14 - mu24 = 0
SCI_LL4 = (xbar1[4] - xbar2[4]) - c*sqrt(((1/n1)+(1/n2))*Sp[4,4])
SCI_UL4 = (xbar1[4] - xbar2[4]) + c*sqrt(((1/n1)+(1/n2))*Sp[4,4])
SCI4 = c(SCI_LL4,SCI_UL4)
SCI4

```

```

##          x4          x4
## -0.1180421  4.7180421

```

```

# BCI for the components mu1-mu2
c_BCI = qt(1-((0.05/p)/2),n1+n2-2) # c for BCI
c_BCI

```

```

## [1] 2.577988

```

```

# Test mu11 - mu21 = 0
BCI_LL1 = (xbar1[1] - xbar2[1]) - c_BCI*sqrt(((1/n1)+(1/n2))*Sp[1,1])
BCI_UL1 = (xbar1[1] - xbar2[1]) + c_BCI*sqrt(((1/n1)+(1/n2))*Sp[1,1])
BCI1 = c(BCI_LL1,BCI_UL1)
BCI1

```

```

##          x1          x1
## -5.309679  1.309679

```

```

# Test mu12 - mu22 = 0
BCI_LL2 = (xbar1[2] - xbar2[2]) - c_BCI*sqrt(((1/n1)+(1/n2))*Sp[2,2])
BCI_UL2 = (xbar1[2] - xbar2[2]) + c_BCI*sqrt(((1/n1)+(1/n2))*Sp[2,2])
BCI2 = c(BCI_LL2,BCI_UL2)
BCI2

```

```

##          x2          x2
## -0.1346243  5.9346243

```

```

# Test mu13 - mu23 = 0
BCI_LL3 = (xbar1[3] - xbar2[3]) - c_BCI*sqrt(((1/n1)+(1/n2))*Sp[3,3])
BCI_UL3 = (xbar1[3] - xbar2[3]) + c_BCI*sqrt(((1/n1)+(1/n2))*Sp[3,3])
BCI3 = c(BCI_LL3,BCI_UL3)
BCI3

```

```

##          x3          x3
## -1.343276  5.543276

```

```

# Test mu14 - mu24 = 0
BCI_LL4 = (xbar1[4] - xbar2[4]) - c_BCI*sqrt(((1/n1)+(1/n2))*Sp[4,4])

```



```
BCI_UL4 = (xbar1[4] - xbar2[4]) + c_BCI*sqrt(((1/n1)+(1/n2))*Sp[4,4])
BCI4 = c(BCI_LL4,BCI_UL4)
BCI4
```

```
##          x4          x4
## 0.395452 4.204548
```

The simultaneous 95% confidence interval for $\mu_{11}-\mu_{21}$ is (-6.202016,2.202016). The simultaneous 95% confidence interval for $\mu_{12}-\mu_{22}$ is (-0.9528036,6.7528036). The simultaneous 95% confidence interval for $\mu_{13}-\mu_{23}$ is (-2.271634,6.471634). The simultaneous 95% confidence interval for $\mu_{14}-\mu_{24}$ is (-0.1180421,4.7180421). The 95% Bonferroni simultaneous confidence interval for $\mu_{11}-\mu_{21}$ is (-5.309679,1.309679). The 95% Bonferroni simultaneous confidence interval for $\mu_{12}-\mu_{22}$ is (-0.1346243,5.9346243). The 95% Bonferroni simultaneous confidence interval for $\mu_{13}-\mu_{23}$ is (-1.343276,5.543276). The 95% Bonferroni simultaneous confidence interval for $\mu_{14}-\mu_{24}$ is (0.395452,4.204548).

(f)

```
x1234 = as.matrix(es[,1:4]) # data matrix with only 1 to 4 column
head(x1234)
```

```
##      x1  x2  x3 x4
## [1,] 131 138  89 49
## [2,] 125 131  92 48
## [3,] 131 132  99 50
## [4,] 119 132  96 44
## [5,] 136 143 100 54
## [6,] 138 137  89 56
```

```
Period = es[,5] # Pick only last column which is the group
head(Period)
```

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

```
fit = manova(x1234~Period) # fit using 1-way manova
summary(fit,test="Wilks") # Summary of Wilk's lambda test
```

```
##           Df    Wilks approx F num Df den Df    Pr(>F)
## Period      1 0.70225    5.8298      4    55 0.0005518 ***
## Residuals 58
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

P_value of Wilk's lambda test is $0.0005518 < 0.05$, so we reject the null hypothesis that the mean vectors are equal at level $\alpha = 0.05$. The p_value we get from Wilk's lambda test is the same as the p_value we get using Hotelling T^2 , because with $g=2$, Wilk's lambda test is just Hotelling T^2 .

Q3

```
Peanuts = read.table("t6_17.txt", header=TRUE)
head(Peanuts)
```

```
##   Fac1 Fac2   x1   x2   x3
## 1    1    5 195.3 153.1 51.4
## 2    1    5 194.3 167.7 53.7
## 3    2    5 189.7 139.5 55.5
```

```
## 4    2    5 180.4 121.1 44.4
## 5    1    6 203.0 156.8 49.8
## 6    1    6 195.9 166.0 45.8
```

```
p = 3
g = 2 # number of levels of factor 1
b = 3 # number of levels of factor 2
n = 2 # number of observation per treatment
```

```
fac1 = factor(Peanuts$Fac1) # factor 1
fac1
```

```
## [1] 1 1 2 2 1 1 2 2 1 1 2 2
## Levels: 1 2
```

```
fac2 = factor(Peanuts$Fac2) # factor 2
fac2
```

```
## [1] 5 5 5 5 6 6 6 6 8 8 8 8
## Levels: 5 6 8
```

```
x345 = as.matrix(Peanuts[,3:5]) # data matrix with only 3 to 5 column
head(x345)
```

```
##      x1      x2      x3
## [1,] 195.3 153.1 51.4
## [2,] 194.3 167.7 53.7
## [3,] 189.7 139.5 55.5
## [4,] 180.4 121.1 44.4
## [5,] 203.0 156.8 49.8
## [6,] 195.9 166.0 45.8
```

(a)

```
fit = manova(x345~fac1*fac2) # fit using 2-way manova
moreinfo = summary(fit)
SSP_fac1 = moreinfo$SS[1] # the sum of squares and cross products matrices for factor 1
SSP_fac1
```

```
## $fac1
##      x1      x2      x3
## x1  0.7008333 -10.6575    7.129167
## x2 -10.6575000 162.0675 -108.412500
## x3  7.1291667 -108.4125    72.520833
```

```
SSP_fac2 = moreinfo$SS[2] # the sum of squares and cross products matrices for factor 2
SSP_fac2
```

```
## $fac2
##      x1      x2      x3
## x1 196.1150 365.1825 42.6275
## x2 365.1825 1089.0150 414.6550
## x3 42.6275 414.6550 284.1017
```

```
SSP_int = moreinfo$SS[3] # the sum of squares and cross products matrices for interaction
SSP_int
```

```
## $`fac1:fac2`
```

```
##           x1           x2           x3
## x1 205.1017 363.6675 107.78583
## x2 363.6675 780.6950 254.22000
## x3 107.7858 254.2200 85.95167

SSP_res = moreinfo$SS[4] # the sum of squares and cross pro ducts matrices for residuals
SSP_res

## $Residuals
##           x1           x2           x3
## x1 104.205 49.365 76.480
## x2 49.365 352.105 121.995
## x3 76.480 121.995 94.835
```

(b)

```
fit2 = manova(x345~fac1+fac2) # This is the additive effects model (ie: no interaction term)
moreinfo2 = summary(fit2)
SSP2_fac1 = moreinfo2$SS[1] # the sum of squares and cross pro ducts matrices for factor 1
SSP2_fac1

## $fac1
##           x1           x2           x3
## x1 0.7008333 -10.6575 7.129167
## x2 -10.6575000 162.0675 -108.412500
## x3 7.1291667 -108.4125 72.520833

SSP2_fac2 = moreinfo2$SS[2] # the sum of squares and cross pro ducts matrices for factor 2
SSP2_fac2

## $fac2
##           x1           x2           x3
## x1 196.1150 365.1825 42.6275
## x2 365.1825 1089.0150 414.6550
## x3 42.6275 414.6550 284.1017

SSP2_res = moreinfo2$SS[3] # the sum of squares and cross pro ducts matrices for residuals
SSP2_res

## $Residuals
##           x1           x2           x3
## x1 309.3067 413.0325 184.2658
## x2 413.0325 1132.8000 376.2150
## x3 184.2658 376.2150 180.7867

# SSPint + SSPres in (a)
SSPintres = SSP_int$`fac1:fac2` + SSP_res$Residuals
SSPintres

##           x1           x2           x3
## x1 309.3067 413.0325 184.2658
## x2 413.0325 1132.8000 376.2150
## x3 184.2658 376.2150 180.7867
```

SSPfac1 and SSPfac2 are exactly the same in (a) and (b). SSPres in (b) = SSPins in (a) + SSPres in (a).