QUIZ 3

Amanjit Gill

November 14, 2022

```
library(CCA)
```

```
## Warning: package 'CCA' was built under R version 4.2.2
## Loading required package: fda
## Warning: package 'fda' was built under R version 4.2.2
## Loading required package: splines
## Loading required package: fds
## Warning: package 'fds' was built under R version 4.2.2
## Loading required package: rainbow
## Warning: package 'rainbow' was built under R version 4.2.2
## Loading required package: MASS
## Loading required package: pcaPP
## Loading required package: RCurl
## Loading required package: deSolve
## Warning: package 'deSolve' was built under R version 4.2.2
##
## Attaching package: 'fda'
## The following object is masked from 'package:graphics':
##
##     matplot
## Loading required package: fields
## Warning: package 'fields' was built under R version 4.2.2
## Loading required package: spam
## Warning: package 'spam' was built under R version 4.2.2
## Spam version 2.9-1 (2022-08-07) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
##
## Attaching package: 'spam'
```

```
## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: viridis

## Warning: package 'viridis' was built under R version 4.2.2

## Loading required package: viridisLite

##
## Try help(fields) to get started.
```

```
library(ggm)
```

```
## Warning: package 'ggm' was built under R version 4.2.2
```

```
library(CCP)

# QUESTION 3

meat <- read.csv("price-cons.csv")[,2:5]

P <- meat[,1:2]
C <- meat[,3:4]

pc.cc <- cc(P, C)
pc.cc[c("cor", "xcoef", "ycoef")]
```

```
## $cor
## [1] 0.8466626 0.7291162
##
## $xcoef
##                  [,1]       [,2]
## steer_price 0.4848242  0.6466666
## hog_price   0.4095221 -0.4438449
##
## $ycoef
##                         [,1]        [,2]
## beef_consumption -0.05040897 -0.22478073
## pork_consumption -0.14820099  0.01749222
```

```
p.perm(P, C, type="Wilks")
```

```
##  Permutation resampling using Wilks 's statistic:
##      stat0     mstat nboot nexcess p
##  0.1326303 0.7876272   999       0 0
```

```
# QUESTION 4

n <- nrow(meat)
p <- ncol(P)
q <- ncol(C)

p.asym(pc.cc$cor, n, p, q, tstat="Wilks")
```

```
## Wilks' Lambda, using F-approximation (Rao's F):
##               stat   approx df1 df2      p.value
## 1 to 2:  0.1326303 13.96689   4  32 1.069829e-06
```

```
## 2 to 2:  0.4683896 19.29457   1  17 3.974817e-04
```

```r
p.asym(pc.cc$cor, n, p, q, tstat="Hotelling")
```

```
##  Hotelling-Lawley Trace, using F-approximation:
##              stat   approx df1 df2      p.value
## 1 to 2:  3.666517 13.74944   4  30 1.759234e-06
## 2 to 2:  1.134975 19.29457   1  34 1.037369e-04
```

```r
p.asym(pc.cc$cor, n, p, q, tstat="Pillai")
```

```
##  Pillai-Bartlett Trace, using F-approximation:
##               stat   approx df1 df2      p.value
## 1 to 2:  1.2484480 14.11986   4  34 6.897527e-07
## 2 to 2:  0.5316104 13.75738   1  38 6.624381e-04
```

```r
p.asym(pc.cc$cor, n, p, q, tstat="Roy")
```

```
##  Roy's Largest Root, using F-approximation:
##               stat   approx df1 df2      p.value
## 1 to 1:  0.7168376 21.51811   2  17 2.199391e-05
##
##  F statistic for Roy's Greatest Root is an upper bound.
```

```r
# QUESTION 6

meat.cor <- cor(meat)

# cor34.12 using pcor

cor34.12 <- pcor(c(3,4,1,2), meat.cor)
cor34.12
```

```
## [1] -0.1881614
```

```r
# cor34.12 using formula, with values from pcor

cor34.2 <- pcor(c(3,4,2), meat.cor)
cor31.2 <- pcor(c(3,1,2), meat.cor)
cor41.2 <- pcor(c(4,1,2), meat.cor)

cor34.12 <- cor34.12 <- (cor34.2 - cor31.2*cor41.2) /
    sqrt((1 - cor31.2**2)*(1 - cor41.2**2))

cor34.12
```

```
## [1] -0.1881614
```

```r
# hypothesis test

pcor.test(cor34.12, 2, 20)
```

```
## $tval
## [1] -0.7663339
##
## $df
## [1] 16
##
## $pvalue
```

```
## [1] 0.4546362
```

```r
# linear regression

meat.mlm <- lm(
    cbind(beef_consumption, pork_consumption) ~ steer_price+hog_price,
    data=meat)

summary(meat.mlm)
```

```
## Response beef_consumption :
##
## Call:
## lm(formula = beef_consumption ~ steer_price + hog_price, data = meat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7148 -2.4360  0.5312  2.3087  4.5577
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  66.3336     6.1292  10.822  4.8e-09 ***
## steer_price  -2.2535     0.5839  -3.859  0.00126 **
## hog_price     1.2252     0.4363   2.808  0.01209 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.149 on 17 degrees of freedom
## Multiple R-squared:  0.5342, Adjusted R-squared:  0.4794
## F-statistic: 9.746 on 2 and 17 DF,  p-value: 0.001514
##
##
## Response pork_consumption :
##
## Call:
## lm(formula = pork_consumption ~ steer_price + hog_price, data = meat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2913 -1.7762 -0.3153  2.5485  7.3788
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 107.6163     7.4913  14.365 6.14e-11 ***
## steer_price  -2.0033     0.7136  -2.807   0.0121 *
## hog_price    -2.7563     0.5332  -5.169 7.71e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.849 on 17 degrees of freedom
## Multiple R-squared:  0.708,  Adjusted R-squared:  0.6736
## F-statistic: 20.61 on 2 and 17 DF,  p-value: 2.859e-05
```

```r
# QUESTION 7
```

```r
# see whether standardised data change the hypothesis test results
# re-use meat data for this

P.std <- scale(P)
C.std <- scale(C)

pc.cc.std <- cc(P.std, C.std)
pc.cc.std[c("cor", "xcoef", "ycoef")]
```

```
## $cor
## [1] 0.8466626 0.7291162
##
## $xcoef
##                  [,1]       [,2]
## steer_price 0.6099651  0.8135815
## hog_price   0.6895407 -0.7473324
##
## $ycoef
##                        [,1]       [,2]
## beef_consumption -0.2199921 -0.9809757
## pork_consumption -0.9984102  0.1178428
```

```r
p.perm(P.std, C.std, type="Wilks")
```

```
##  Permutation resampling using Wilks 's statistic:
##      stat0    mstat nboot nexcess p
##  0.1326303 0.790945   999       0 0
```

```r
p.asym(pc.cc.std$cor, n, p, q, tstat="Wilks")
```

```
## Wilks' Lambda, using F-approximation (Rao's F):
##               stat   approx df1 df2      p.value
## 1 to 2:  0.1326303 13.96689   4  32 1.069829e-06
## 2 to 2:  0.4683896 19.29457   1  17 3.974817e-04
```

```r
p.asym(pc.cc.std$cor, n, p, q, tstat="Hotelling")
```

```
##  Hotelling-Lawley Trace, using F-approximation:
##              stat   approx df1 df2      p.value
## 1 to 2:  3.666517 13.74944   4  30 1.759234e-06
## 2 to 2:  1.134975 19.29457   1  34 1.037369e-04
```

```r
p.asym(pc.cc.std$cor, n, p, q, tstat="Pillai")
```

```
##  Pillai-Bartlett Trace, using F-approximation:
##               stat   approx df1 df2      p.value
## 1 to 2:  1.2484480 14.11986   4  34 6.897527e-07
## 2 to 2:  0.5316104 13.75738   1  38 6.624381e-04
```

```r
p.asym(pc.cc.std$cor, n, p, q, tstat="Roy")
```

```
##  Roy's Largest Root, using F-approximation:
##               stat   approx df1 df2      p.value
## 1 to 1:  0.7168376 21.51811   2  17 2.199391e-05
##
##  F statistic for Roy's Greatest Root is an upper bound.
```

```r
# QUESTION 9

S <- matrix(
    c(
        177, 40, -14, 4,
        40, 98, -37, -4,
        -14, -37, 314, 5,
        4, -4, 5, 1
    ),
    4, 4)

# canonical correlations between (X1,X2) and (X3,x4)
# using Cholesky decomposition for S22_neg_sqrt

S11 <- S[1:2, 1:2]
S12 <- S[1:2, 3:4]
S21 <- S[3:4, 1:2]
S22 <- S[3:4, 3:4]

S22_neg_sqrt <- chol(solve(S22))

prod <- S22_neg_sqrt %*% S21 %*% solve(S11) %*% S12 %*% t(S22_neg_sqrt)

sqrt(eigen(prod)$values)
```

```
## [1] 0.6011624 0.1590759
```

```r
# hypothesis test for zero correlation b/w (X1,X2) and (X3,x4)

n <- 30
p <- 2
q <- 2

m <- n - 3/2 - (p+q)/2
s <- sqrt((p**2 * q**2 - 4)/(p**2 + q**2 - 5))

df1 <- p*q
df2 <- m*s - p*q/2 + 1

(big_lambda <- det(S) / (det(S11) * det(S22)))
```

```
## [1] 0.6224438
```

```r
# check big_lambda using Q1 and Q2

Q1 <- S22 - S21 %*% solve(S11) %*% S12
Q2 <- S21 %*% solve(S11) %*% S12
det(Q1 %*% solve(Q1+Q2))
```

```
## [1] 0.6224438
```

```r
# final figures for hypothesis test

(F_stat <- (1-sqrt(big_lambda))/sqrt(big_lambda) * df2/df1)
```

```
## [1] 3.477574
```

```r
(p_value <- pf(F_stat,df1, df2, lower.tail = FALSE))
```

```
## [1] 0.01365727
```