MA50259: Statistical Design of Investigations

Lab sheet 3: Estimability in designs of less than full rank

In this practical you will learn more of how to analyse an experimental design that does not have full rank and also the basics of the corresponding hypothesis testing. Take the time to run each of the following commands and analyse the displayed results to understand what the code is doing.

Estimation in the treatment effects model

Consider the treatment effects model of a completely randomized design (CRD) with $t = 3$ levels and $r_i = 4$ replicates

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\boldsymbol{\epsilon} \sim MVN(\boldsymbol{0}, \sigma^2\boldsymbol{I})$ and

$$\boldsymbol{y} = \begin{pmatrix} y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{21} \\ y_{22} \\ y_{23} \\ y_{24} \\ y_{31} \\ y_{32} \\ y_{33} \\ y_{34} \end{pmatrix}, \quad \boldsymbol{X} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \mu \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix}, \quad \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{13} \\ \epsilon_{14} \\ \epsilon_{21} \\ \epsilon_{22} \\ \epsilon_{23} \\ \epsilon_{24} \\ \epsilon_{31} \\ \epsilon_{32} \\ \epsilon_{33} \\ \epsilon_{34} \end{pmatrix}$$

1. Load the corresponding packages to use in this tutorial

```
library(tidyverse)
library(MASS)
```

2. As done in Lab 2, simulate the response with a normal distribution. Also, construct the design matrix $\boldsymbol{X}$.

```
t<-3; # the number of treatment levels
r<-4; # the number of replicates
n<-t*r
levels<-c("level 1","level 2","level 3");
fact<-gl(t,r,labels=levels)
crd <- tibble(  treatment=fact )
set.seed(13579)
mu<-500 # reference value
tau<-c(50,0,-30) # treatment effects= differences wrt to mu
sd<-10 # overall standard deviation
```

```r
means<-mu+tau %>% rep(each=r) # vector of means
y<-rnorm(n,mean=means,sd=sd)
crd$response<-y
crd
```

```
## # A tibble: 12 x 2
##    treatment response
##    <fct>        <dbl>
##  1 level 1       538.
##  2 level 1       537.
##  3 level 1       547.
##  4 level 1       535.
##  5 level 2       511.
##  6 level 2       525.
##  7 level 2       508.
##  8 level 2       502.
##  9 level 3       460.
## 10 level 3       467.
## 11 level 3       477.
## 12 level 3       475.
```

```r
# construct the design matrix X
Z <- model.matrix(~ fact-1)
Z
```

```
##    factlevel 1 factlevel 2 factlevel 3
## 1            1           0           0
## 2            1           0           0
## 3            1           0           0
## 4            1           0           0
## 5            0           1           0
## 6            0           1           0
## 7            0           1           0
## 8            0           1           0
## 9            0           0           1
## 10           0           0           1
## 11           0           0           1
## 12           0           0           1
## attr(,"assign")
## [1] 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$fact
## [1] "contr.treatment"
```

```r
X<-cbind(1,as.matrix(Z))
X
```

```
##      factlevel 1 factlevel 2 factlevel 3
## 1  1           1           0           0
## 2  1           1           0           0
## 3  1           1           0           0
## 4  1           1           0           0
## 5  1           0           1           0
## 6  1           0           1           0
## 7  1           0           1           0
## 8  1           0           1           0
## 9  1           0           0           1
## 10 1           0           0           1
## 11 1           0           0           1
## 12 1           0           0           1
```

```
colnames(X)<-c("reference","effect 1","effect 2","effect 3")
```

3. **(H) Definition:** A generalized inverse of the symmetric matrix $\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}$ is a square matrix $\boldsymbol{G}$ satisfying

$$\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{G}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X} = \boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}$$

The command `ginv` in the package `MASS` computes one type of generalised inverse called the Moore-Penrose generalised inverse (here we call it $\boldsymbol{G}_1$). Another type of generalized inverse can be calculated for our specific case of $\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}$ as follows

$$\boldsymbol{G}_2 = \begin{pmatrix} 0 & 0 \\ 0 & (\boldsymbol{Z}^{\boldsymbol{T}}\boldsymbol{Z})^{-1} \end{pmatrix}$$

where $\boldsymbol{Z}$ is the design matrix of the means model. Compute two solutions, $\tilde{\boldsymbol{\beta}}_1$ and $\tilde{\boldsymbol{\beta}}_2$ to the normal equations

$$\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\beta = \boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{y}$$

in our example using these two types of generalised inverse. That is, compute $\tilde{\boldsymbol{\beta}}_1 = \boldsymbol{G}_1\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{y}$ and $\tilde{\boldsymbol{\beta}}_2 = \boldsymbol{G}_2\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{y}$

```
# Q1
ginv1 <- ginv(t(X) %*% X)
ginv2 <- rbind(0, cbind(0, solve(t(Z) %*% Z)))
cat("G-Inverse2:\n")
```

```
## G-Inverse2:
```

```
ginv2
```

```
##                factlevel 1 factlevel 2 factlevel 3
##              0        0.00        0.00        0.00
## factlevel 1 0        0.25        0.00        0.00
## factlevel 2 0        0.00        0.25        0.00
## factlevel 3 0        0.00        0.00        0.25
```

```
beta1 <- ginv1 %*% t(X) %*% y
beta2 <- ginv2 %*% t(X) %*% y
cat("Beta 1:\n")
```

```
## Beta 1:
```

```
beta1
```

```
##            [,1]
## [1,] 380.13679
## [2,] 159.19078
## [3,] 131.24750
## [4,]  89.69851
```

```
cat("Beta 2:\n")
```

```
## Beta 2:
```

```
beta2
```

```
##                   [,1]
##                 0.0000
## factlevel 1 539.3276
## factlevel 2 511.3843
## factlevel 3 469.8353
```

4. **Definition:** A linear combination $\boldsymbol{\lambda^T\beta}$ of the parameter vector $\boldsymbol{\beta}$ is said to be **estimable** if it can be estimated unbiasedly with a linear combination of the observations, that is, if there exists a vector $\boldsymbol{a}$ of length n such that

$$E[\boldsymbol{a^TY}] = \boldsymbol{\lambda^T\beta}\,, \qquad \text{for all } \boldsymbol{\beta}$$

- It is easy to check that $\boldsymbol{\lambda^T\beta}$ is estimable if and only if there exists a vector $\boldsymbol{a}$ of length n such that $\boldsymbol{\lambda} = \boldsymbol{X^Ta}$, that is, if $\boldsymbol{\lambda}$ belongs to the space generated by the rows of $\boldsymbol{X}$. Which ones of the following linear combinations are estimable?
- $\boldsymbol{\lambda_1^T}\beta = \mu + \tau_1$
- $\boldsymbol{\lambda_2^T}\beta = \tau_1 - \tau_2$
- $\boldsymbol{\lambda_3^T}\beta = \mu - \tau_1$
- $\boldsymbol{\lambda_4^T}\beta = \tau_1$

```
lambda1<-c(1,1,0,0)
lambda2<-c(0,1,-1,0)
lambda3<-c(1,-1,0,0)
lambda4<-c(0,1,0,0)
qr(rbind(X,lambda1))$rank
```

```
## [1] 3
```

```
qr(rbind(X,lambda2))$rank
```

```
## [1] 3
```

```
qr(rbind(X,lambda3))$rank
```

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

```r
qr(rbind(X,lambda4))$rank
```

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

<u>My answer</u>

- $\boldsymbol{\lambda_1^T\beta} = \mu + \tau_1$ and $\boldsymbol{\lambda_2^T\beta} = \tau_1 - \tau_2$ are estimable, $\boldsymbol{\lambda_3^T\beta} = \mu - \tau_1$ and $\boldsymbol{\lambda_4^T\beta} = \tau_1$ are not estimable.
- You check for estimability by noting the rank of the extended matrix.
- Since $\boldsymbol{\lambda_1^T\beta}$ and a$\boldsymbol{\lambda_2^T\beta} = \tau_1 - \tau_2$ are full rank with rank $= 3$, they are estimable as they are not linearly dependant on column 1 of X due to being a 3x3 matrix (Both are actually derived from Z matrix), unlike $\boldsymbol{\lambda_3^T\beta} = \mu - \tau_1$ and $\boldsymbol{\lambda_4^T\beta} = \tau_1$ which are rank 4 and therefore linearly dependant on column 1 of X due to being a 4x4 matrix.

---

5. **(H)** If $\boldsymbol{\lambda^T\beta}$ is estimable, then we can estimate it with any solution to the normal equations $\tilde{\boldsymbol{\beta}}$ by simply using

$$\widehat{\boldsymbol{\lambda^T\beta}} := \boldsymbol{\lambda^T}\tilde{\boldsymbol{\beta}} = \boldsymbol{\lambda^T G X^T y} = \boldsymbol{a^T X G X^T y}$$

and the estimator does not depend on the particular choice of generalised inverse $\boldsymbol{G}$. Verify that this last statement is true for the linear combinations $\boldsymbol{\lambda_1^T\beta} = \mu + \tau_1$ and $\boldsymbol{\lambda_2^T\beta} = \tau_1 - \tau_2$ using the two solutions of the normal equations found in question 3.

```r
# For lambda1 = mu + tau1
test1 <- lambda1%*%beta1
test2 <- lambda1%*%beta2
check1 <- mu + tau[1]
difference1 <- test1 - check1

# For lambda2 = tau1 - tau2
test3 <- lambda2%*%beta1
test4 <- lambda2%*%beta2
check2 <- tau[1] - tau[2]
difference2 <- test4 - check2

cat("test1: ", test1)
```

```
## test1:  539.3276
```

```r
cat("test2: ", test2)
```

```
## test2:  539.3276
```

```r
cat("test3: ", test3)
```

```
## test3:  27.94328
```

```
cat("test4: ", test4)
```

```
## test4:  27.94328
```

```
cat("mu + tau1: ", check1)
```

```
## mu + tau1:  550
```

```
cat("tau1 - tau2: ", check2)
```

```
## tau1 - tau2:  50
```

```
cat("Difference between lambda1 and check1:", difference1)
```

```
## Difference between lambda1 and check1: -10.67244
```

```
cat("Difference between lambda2 and check2:", difference2)
```

```
## Difference between lambda2 and check2: -22.05672
```

My answer

- $\boldsymbol{\lambda_1^T \beta_1}$ and $\boldsymbol{\lambda_1^T \beta_2}$ return approximately same value, as do $\boldsymbol{\lambda_2^T \beta_1}$ and $\boldsymbol{\lambda_2^T \beta_2}$. This is because the estimator does not depend on the particular choice of generalised inverse $\boldsymbol{G}$ being used.
- Difference between $\boldsymbol{\lambda_1^T \beta_1}$ and $\boldsymbol{\lambda_1^T \beta_2}$ versus $\boldsymbol{\mu + \tau_1}$ is -10.67 and difference between $\boldsymbol{\lambda_2^T \beta_1}$ and $\boldsymbol{\lambda_2^T \beta_2}$ versus $\boldsymbol{\tau_1 - \tau_2}$ is -22.06.

---

6. The reason why the 2 estimators above: one based on $\boldsymbol{G}_1$ and the other based on $\boldsymbol{G}_2$, do not depend on the particular choice of generalised inverse is because the product

$$\boldsymbol{XGX^T}$$

is invariant to the choice of generalised inverse $\boldsymbol{G}$. Verify this product does not depend on the choice of $\boldsymbol{G}$ using the two generalised inverses $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ computed above, that is, verify that

$$\boldsymbol{XG_1X^T} = \boldsymbol{XG_2X^T}$$

```
# Output of both commands indicate a near zero difference
X%*%ginv1%*%t(X)-X%*%ginv2%*%t(X)
```

```
##              1            2            3            4             5
## 1  0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  4.163336e-17
## 2  0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  4.163336e-17
## 3  0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  4.163336e-17
## 4  0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00  4.163336e-17
## 5  1.110223e-16 1.110223e-16 1.110223e-16 1.110223e-16  0.000000e+00
## 6  1.110223e-16 1.110223e-16 1.110223e-16 1.110223e-16  0.000000e+00
## 7  1.110223e-16 1.110223e-16 1.110223e-16 1.110223e-16  0.000000e+00
## 8  1.110223e-16 1.110223e-16 1.110223e-16 1.110223e-16  0.000000e+00
## 9  5.551115e-17 5.551115e-17 5.551115e-17 5.551115e-17 -1.387779e-17
## 10 5.551115e-17 5.551115e-17 5.551115e-17 5.551115e-17 -1.387779e-17
## 11 5.551115e-17 5.551115e-17 5.551115e-17 5.551115e-17 -1.387779e-17
## 12 5.551115e-17 5.551115e-17 5.551115e-17 5.551115e-17 -1.387779e-17
##                6             7             8             9            10
## 1   4.163336e-17  4.163336e-17  4.163336e-17  5.551115e-17  5.551115e-17
## 2   4.163336e-17  4.163336e-17  4.163336e-17  5.551115e-17  5.551115e-17
## 3   4.163336e-17  4.163336e-17  4.163336e-17  5.551115e-17  5.551115e-17
## 4   4.163336e-17  4.163336e-17  4.163336e-17  5.551115e-17  5.551115e-17
## 5   0.000000e+00  0.000000e+00  0.000000e+00 -1.387779e-17 -1.387779e-17
## 6   0.000000e+00  0.000000e+00  0.000000e+00 -1.387779e-17 -1.387779e-17
## 7   0.000000e+00  0.000000e+00  0.000000e+00 -1.387779e-17 -1.387779e-17
## 8   0.000000e+00  0.000000e+00  0.000000e+00 -1.387779e-17 -1.387779e-17
## 9  -1.387779e-17 -1.387779e-17 -1.387779e-17  5.551115e-17  5.551115e-17
## 10 -1.387779e-17 -1.387779e-17 -1.387779e-17  5.551115e-17  5.551115e-17
## 11 -1.387779e-17 -1.387779e-17 -1.387779e-17  5.551115e-17  5.551115e-17
## 12 -1.387779e-17 -1.387779e-17 -1.387779e-17  5.551115e-17  5.551115e-17
##               11            12
## 1   5.551115e-17  5.551115e-17
## 2   5.551115e-17  5.551115e-17
## 3   5.551115e-17  5.551115e-17
## 4   5.551115e-17  5.551115e-17
## 5  -1.387779e-17 -1.387779e-17
## 6  -1.387779e-17 -1.387779e-17
## 7  -1.387779e-17 -1.387779e-17
## 8  -1.387779e-17 -1.387779e-17
## 9   5.551115e-17  5.551115e-17
## 10  5.551115e-17  5.551115e-17
## 11  5.551115e-17  5.551115e-17
## 12  5.551115e-17  5.551115e-17
```

```
near(X%*%ginv1%*%t(X),X%*%ginv2%*%t(X))
```

```
##       1    2    3    4    5    6    7    8    9   10   11   12
## 1  TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## 2  TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## 3  TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## 4  TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## 5  TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## 6  TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## 7  TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## 8  TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## 9  TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## 10 TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## 11 TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## 12 TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

7. **(H)** For the treatment effects model in a CRD, there can be at most t linearly independent estimable linear combinations, in other words, the space of estimable linear combinations is of dimension t, the

number of levels of the treatment effect. Given two estimable linear combinations $\boldsymbol{\lambda_1^T\beta}$ and $\boldsymbol{\lambda_2^T\beta}$, they are said to be linearly independent if $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ are linearly independent. Verify that $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ in question 5 are linearly independent. Here, the dimension is $t = 3$ so find a third linearly independent combination $\boldsymbol{\lambda_3^T\beta}$

```
# A linear combination lambda_T is estimable if and only if there exists a solution R to the system X^T
r_1 <- ginv1 %*% lambda1
r_1
```

```
##         [,1]
## [1,]  0.0625
## [2,]  0.1875
## [3,] -0.0625
## [4,] -0.0625
```

```
r_2 <- ginv1 %*% lambda2
r_2
```

```
##               [,1]
## [1,] -1.214306e-17
## [2,]  2.500000e-01
## [3,] -2.500000e-01
## [4,]  8.326673e-17
```

```
# How to find r_3?

# Chatgpt
qr(cbind(X, lambda1))$rank
```

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

```
qr(cbind(X, lambda2))$rank
```

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

```
qr(cbind(X, lambda3))$rank
```

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

```
# Once we confirm lambda1 and lambda2 are linearly independent, we can find a third linearly independen
# This third vector lambda3 cannot be expressed as a linear combination of lambda1 and lambda2, and it
# But rank is 3 for both, isn't it non-full rank and thus not linearly independent?

# strange thing is r(bind(X, lambda1, 2, and 3))$rank is 3, cbind gets me 4 maybe this is the answer?
```

8. Any linear model of less than full rank (such as the treatment effects model in a CRD) can always be reparametrised into a linear model of full rank. Given t linearly independent estimable combinations $\boldsymbol{\lambda_1^T\beta}, \ldots, \boldsymbol{\lambda_t^T\beta}$. Let $\boldsymbol{\Lambda^T} = (\boldsymbol{\lambda}_1, \cdots, \boldsymbol{\lambda}_t)$ be the matrix of dimension $(t+1) \times t$ whose columns are

$\{\boldsymbol{\lambda}_i\}$. Let γ be a vector of length $t+1$ such that the square matrix $\boldsymbol{U}^T = (\boldsymbol{\lambda}_1, \cdots, \boldsymbol{\lambda}_t, \gamma) = (\boldsymbol{\Lambda} \,|\, \gamma)$ is non-singular with inverse $\boldsymbol{W} = \boldsymbol{U}^{-1} = (\boldsymbol{W}_1 \,|\, \eta)$. If $\boldsymbol{W}$ diagonalises $\boldsymbol{X^T X}$, that is

$$\boldsymbol{W}^T \boldsymbol{X^T X W} = diag(v_1, \ldots, v_t, 0)$$

for some constants $v_1, \ldots, v_t$. Then $\boldsymbol{X\eta} = \boldsymbol{0}$ and we can write

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon} = (\boldsymbol{X W_1})(\boldsymbol{\Lambda\beta}) + (\boldsymbol{X\eta})(\gamma^{\boldsymbol{T}}\boldsymbol{\beta}) + \boldsymbol{\epsilon} = \boldsymbol{Z\delta} + \boldsymbol{\epsilon}$$

where $\boldsymbol{Z} = \boldsymbol{XW_1}$ has rank t and $\boldsymbol{\delta} = \boldsymbol{\Lambda\beta}$ so this model is full rank. The reparametrised full rank model is not unique. The commands below construct two different reparametrised full rank models both starting from the non full rank treatment effect model. Verify that the first one corresponds to the means model.

```
lambda1<-c(1,1,0,0)
lambda2<-c(1,0,1,0)
lambda3<-c(1,0,0,1)
Lambda=rbind(lambda1,lambda2,lambda3)
gama<-c(1,1,0,1) # arbitrary vector as long as U is non-singular
#gama<-c(1,0,1,1) # alternative 1
# gama<-c(0,0,1,1) # alternative 2
U<-rbind(Lambda,gama)
U
```

```
##         [,1] [,2] [,3] [,4]
## lambda1    1    1    0    0
## lambda2    1    0    1    0
## lambda3    1    0    0    1
## gama       1    1    0    1
```

```
qr(U)$rank
```

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

```
W<-solve(U) # use solve() to get the inverse
W
```

```
##      lambda1 lambda2 lambda3 gama
## [1,]       1       0       1   -1
## [2,]       0       0      -1    1
## [3,]      -1       1      -1    1
## [4,]      -1       0       0    1
```

```
W1<-W[,1:3]
eta<-W[,4]
X%*%eta # should be zero vector
```

```
##    [,1]
## 1     0
## 2     0
## 3     0
## 4     0
## 5     0
## 6     0
## 7     0
## 8     0
## 9     0
## 10    0
## 11    0
## 12    0
```

```
Z1<-X%*%W1 # new design matrix
qr(Z1)$rank # should have full column rank
```

```
## [1] 3
```

```
t(W)%*%t(X)%*%X%*%W # check that it diagonalises X'X
```

```
##         lambda1 lambda2 lambda3 gama
## lambda1       4       0       0    0
## lambda2       0       4       0    0
## lambda3       0       0       4    0
## gama          0       0       0    0
```

```
# second reparametrisation
# uses eigenvectors to guarantee diagonalisation

E<-t(eigen(t(X)%*%X)$vectors)
E
```

```
##           [,1]       [,2]       [,3]       [,4]
## [1,] 0.8660254  0.2886751  0.2886751  0.2886751
## [2,] 0.0000000 -0.5773503 -0.2113249  0.7886751
## [3,] 0.0000000 -0.5773503  0.7886751 -0.2113249
## [4,] 0.5000000 -0.5000000 -0.5000000 -0.5000000
```

```
Lambda=E[1:3,]
gama<-E[4,]
U<-rbind(Lambda,gama)
U
```

```
##           [,1]       [,2]       [,3]       [,4]
##      0.8660254  0.2886751  0.2886751  0.2886751
##      0.0000000 -0.5773503 -0.2113249  0.7886751
##      0.0000000 -0.5773503  0.7886751 -0.2113249
## gama 0.5000000 -0.5000000 -0.5000000 -0.5000000
```

```
qr(U)$rank
```

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

```
W<-solve(U)
W
```

```
##                                              gama
## [1,] 0.8660254  3.204938e-17  3.204938e-17  0.5
## [2,] 0.2886751 -5.773503e-01 -5.773503e-01 -0.5
## [3,] 0.2886751 -2.113249e-01  7.886751e-01 -0.5
## [4,] 0.2886751  7.886751e-01 -2.113249e-01 -0.5
```

```
W1<-W[,1:3]
eta<-W[,4]
X%*%eta # should be zero vector
```

```
##             [,1]
## 1  0.000000e+00
## 2  0.000000e+00
## 3  0.000000e+00
## 4  0.000000e+00
## 5  2.220446e-16
## 6  2.220446e-16
## 7  2.220446e-16
## 8  2.220446e-16
## 9  2.775558e-16
## 10 2.775558e-16
## 11 2.775558e-16
## 12 2.775558e-16
```

```
Z2<-X%*%W1 # new design matrix
qr(Z2)$rank # should have full column rank
```

```
## [1] 3
```

```
t(W)%*%t(X)%*%X%*%W # check that it diagonalises X'X
```

```
##                                                       gama
##       1.600000e+01 1.415592e-16 -6.331015e-17 3.877810e-15
##       3.538981e-17 4.000000e+00  6.780619e-16 9.694524e-16
##      -4.278674e-16 6.546001e-16  4.000000e+00 5.253632e-16
## gama  2.307555e-15 6.879111e-16  4.658665e-16 4.807121e-31
```

9. The error sum of squares is defined as

$$ssE := \|\boldsymbol{y} - \boldsymbol{X}\tilde{\boldsymbol{\beta}}\|^2$$

where $\tilde{\boldsymbol{\beta}}$ is a solution to the normal equations. This can be written as

$$ssE := \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{G}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{y}\|^2$$

and therefore is invariant to the choice of generalised inverse $\boldsymbol{G}$ since $\boldsymbol{X}\boldsymbol{G}\boldsymbol{X}^{\boldsymbol{T}}$ is also invariant. Verify, using our observed sample that

$$\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{G_1}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{y}\|^2 - \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{G_2}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{y}\|^2$$

```
# enough to check the vectors
d1<-y-X%*%ginv1%*%t(X)%*%y
d2<-y-X%*%ginv2%*%t(X)%*%y
d1-d2
```

10. To test the hypothesis

$$H_0 : \tau_1 = \cdots = \tau_t$$

we can compute the error sum of squares in the reduced model where H_0 is true. Note this reduced model is given by

$$\boldsymbol{Y} = \boldsymbol{1}\mu + \boldsymbol{\epsilon}$$

where $\boldsymbol{1}$ is the column vector on ones of dimension n. Clearly this model is of full rank and the solution to the normal equations

$$\boldsymbol{1^T 1}\mu = \boldsymbol{1^T y} = \sum_{i=1}^{n} y_i$$

is $\hat{\mu} = \bar{y} = \boldsymbol{1^T y}/n$ and the corresponding error sum of squares is

$$ssE_0 := \|\boldsymbol{y} - \boldsymbol{1}\hat{\mu}\|^2 = \|\boldsymbol{y} - \boldsymbol{1}\bar{y}\|^2$$

The test statistic commonly used is

$$F = \frac{n-t}{t-1}\frac{SSE_0 - SSE}{SSE}$$

Compute F for our data and compare to the one given using the `aov` command in R given by `aov(response treatment,crd)`.

```
SSE<-t(d1)%*%d1
d0<-y-mean(y)
SSE0<-t(d0)%*%d0
FF<-((n-t)/(t-1))*(SSE0-SSE)/SSE
FF
aov1<-aov(response~treatment,crd)
summary(aov1)
```