MAST90104: A First Course in Statistical Learning

Week 8 Practical and Workshop

1 Practical questions

1. Consider the filter question in Week 7. Recall that we are interested in comparing the lifespan of 5 different types of filters. Six filters of each type are tested, and the time to failure in hours is given in the dataset (on the website) `filters` (in `csv` format).

   Read the data. Then convert the `type` component into a factor. Last week fit a one-way classification model using the treatment contrast (R code below).

```
> library(Matrix)
> filters <- read.csv("filters.csv")
> filters$type <- factor(filters$type)
> y <- filters$life
> X.treatment <- matrix(0,length(y),6)
> X.treatment [,1] <- 1
> for (i in 1:5) { X.treatment [filters$type==i,i+1] <- 1 }
```

   (a) Calculate a 95% confidence interval for the difference in lifespan between filter types 3 and 4.
   Solution: After reparametrization, the model become

$$Y = \widetilde{X}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$

   so the parameters are $\boldsymbol{\gamma} = (\mu+\tau_1, \tau_2-\tau_1, \tau_3-\tau_1, \tau_4-\tau_1, \tau_5-\tau_1)^T$. The difference between filter type 3 and type 4 would be $\tau_3-\mu_3-\mu_4 = \mathbf{t}^T\boldsymbol{\gamma}$ where $\mathbf{t} = (0,0,1,-1,0)^T$. The reparameterized model is full rank so we can use the results in Lecture 4 part II.

```
> BigC <- matrix(0,nrow = ncol(X.treatment), ncol = rankMatrix(X.treatment)[1])
> BigC[1,1] <- 1
> BigC[2:ncol(X.treatment),2:(rankMatrix(X.treatment)[1])] <- contr.treatment(5)
> Xtilde <- X.treatment%*%BigC
>
> gammahat <- c(solve(t(Xtilde )%*%Xtilde )%*%t(Xtilde )%*%y)
> s2 <- sum((y - c(Xtilde%*%gammahat))^2)/(length(y) - rankMatrix(X.treatment)[1])
> Treatment.contrast.est <- gammahat[3] - gammahat[4]
> tt <- c(0,0,1,-1,0)
> c(Treatment.contrast.est - qt(0.975,df = (length(y) - rankMatrix(X.treatment)[1]))
*sqrt(s2*t(tt)%*% solve(t(Xtilde)%*%Xtilde) %*% tt ),
Treatment.contrast.est + qt(0.975,df = (length(y) - rankMatrix(X.treatment)[1]))
*sqrt(s2*t(tt)%*% solve(t(Xtilde)%*%Xtilde) %*% tt ))
[1] -338.43399  -44.23268
```

   (b) Show that the hypothesis that the filters all have the same lifespan is testable.
   Solution: The hypothesis is true if and only if the differences between the means of the levels are 0.

$$H_0 : \tau_1 = \ldots = \tau_5.$$

   We show in the notes that the difference are contrasts and hence estimable. We can express the hypothesis as $H_0 : L\boldsymbol{\beta} = 0$, where

$$L = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

   $\boldsymbol{\beta}$ is parameter vector under original parameterisation. The rows of L are obviously independent and hence H_0 is testable.

(c) Test this hypothesis, using matrix theory.

Solution

```
> L <- rbind(c(0,1,-1,0,0,0),c(0,0,1,-1,0,0),c(0,0,0,1,-1,0), c(0,0,0,0,1,-1) )
> Ltilde <- cbind(0,diag(1,4))
> Lbetahat <- Ltilde%*% gammahat
> Fstat.numerator <- t(Lbetahat)%*%
solve(Ltilde%*%solve(t(Xtilde)  %*%Xtilde)%*%t(Ltilde))
%*%Lbetahat/rankMatrix(Ltilde)[1]
> Fstat.denominator <- sum(c((y - Xtilde%*%gammahat)^2))/(n - rankMatrix(Xtilde)[1])
> Fstat <- Fstat.numerator/Fstat.denominator
> pval <- pf(Fstat,rankMatrix(Ltilde)[1],(n - rankMatrix(Xtilde)[1]), lower.tail = FALSE)
> Fstat
[,1]
[1,] 3.318776
> pval
[,1]
[1,] 0.02599945
```

(d) Test the same hypothesis using the `linearHypothesis` function from the `car` package.

Solution

```
> model.fit <- lm(life~type, data=filters)
> library(car)
> linearHypothesis(model.fit, Ltilde, rep(0,4))
Linear hypothesis test

Hypothesis:
type2 = 0
type3 = 0
type4 = 0
type5 = 0

Model 1: restricted model
Model 2: life ~ type

Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     29 585770
2     25 382605  4    203165 3.3188  0.026 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

(e) Repeat part d using the sum-to-0 contrast (`contr.sum`)

Solution:

```
> model.sum.fit <- lm(life~type, data=filters, contrasts = list(type = 'contr.sum'))
> library(car)
> #In this case, Ltilde coincidentally is the same matrix
> linearHypothesis(model.sum.fit, Ltilde, rep(0,4))
Linear hypothesis test

Hypothesis:
type2 = 0
type3 = 0
type4 = 0
type5 = 0

Model 1: restricted model
Model 2: life ~ type
```

```
Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     29 585770
2     25 382605  4    203165 3.3188  0.026 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

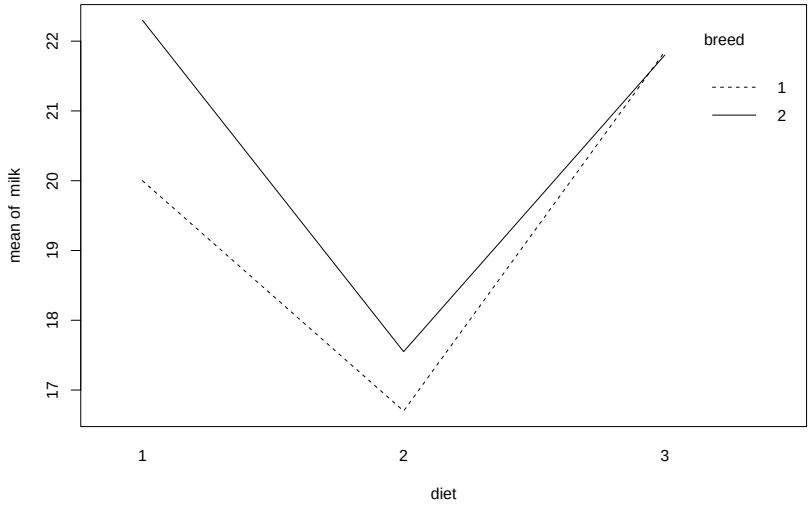

Figure 1: Interaction plot between breed and diet

2. We study the effect of various breeds and diets on the milk yield of cows. A study is conducted on 9 cows and the following data obtained:

	Diet		
Breed	1	2	3
1	18.8	16.7	19.8
	21.2		23.9
2	22.3	15.9	21.8
		19.2	

(a) Input this data into R. Plot an interaction plot between breed and diet.

Solution

```
>milk <- data.frame(milk=c(18.8,21.2,16.7,19.8,23.9,22.3,15.9,19.2,21.8),
diet=factor(c(1,1,2,3,3,1,2,2,3)),
breed=factor(c(1,1,1,1,1,2,2,2,2)))
> with(milk, interaction.plot(diet, breed, milk))
```

(b) Fit an additive model. What is the estimated amount of milk produced from breed 2 and diet 3 now?

Solution:

```
> amodel <- lm(milk ~ breed + diet, data=milk)
> amodel$coeff
(Intercept)      breed2       diet2       diet3
  20.422222    1.033333   -3.844444    1.066667
> c(1,1,0,1)%*%amodel$coefficients
         [,1]
[1,] 22.52222
```

(c) Test the hypothesis (under the additive model) that the 2nd and 3rd diets are equivalent in terms of milk produced.

Solution

```
> linearHypothesis(amodel, c(0,0,1,-1),0)
Linear hypothesis test

Hypothesis:
```

```
diet2 - diet3 = 0

Model 1: restricted model
Model 2: milk ~ breed + diet

  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1      6 52.000
2      5 18.604  1    33.396 8.9752 0.03024 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

(d) Find a 95% confidence interval, under the additive model, for the amount of milk produced from breed 2 and diet 3. Use both matrix calculations and the `estimable` function from the `gmodels` package.

Solution Use matrix calculation

```
> n <- 9
> Xtilde <- model.matrix(~breed+diet,data=milk)
  (Intercept) breed2 diet2 diet3
1           1      0     0     0
2           1      0     0     0
3           1      0     1     0
4           1      0     0     1
5           1      0     0     1
6           1      1     0     0
7           1      1     1     0
8           1      1     1     0
9           1      1     0     1
attr(,"assign")
[1] 0 1 2 2
attr(,"contrasts")
attr(,"contrasts")$breed
[1] "contr.treatment"

attr(,"contrasts")$diet
[1] "contr.treatment"

> y <- milk$milk
> XtildetXtildeinv <- solve(t(Xtilde) %*% Xtilde)
> gammahat <- XtildetXtildeinv %*% t(Xtilde) %*% y
> r <- rankMatrix(Xtilde)
> s2 <- sum((y - Xtilde %*% gammahat)^2)/(n - r)
> t <- c(1,1,0,1)
> mu23 <- t(t) %*% gammahat
> wdth <- qt(.975, n - r)*sqrt(s2 * t(t) %*% XtildetXtildeinv %*% t)
> c(mu23 - wdth, mu23 + wdth)
[1] 18.82634 26.21811
```

Use the function `estimable`

```
> library(gmodels)
> help("estimable")
> estimable(amodel, c(1,1,0,1), conf.int=0.95)
          Estimate Std. Error  t value DF     Pr(>|t|) Lower.CI Upper.CI
(1 1 0 1) 22.52222   1.437762 15.66477  5 1.927104e-05 18.82634 26.21811
```