Assignment3

Student number:1407870

2023-09-26

Subject name: A First Course In Statistical Learning
Subject code: MAST90104

Q1

```r
#(a)
fat=read.csv('/Users/guanmuhan/Downloads/fat.csv')
fat$Breed=as.factor(fat$Breed)
fat$Age=as.factor(fat$Age)
fat_model_additive=lm(Butterfat~Breed+Age,contrasts = list(Breed='contr.sum',Age='contr.sum'),data=fat)
levels(fat$Age)
```

```
## [1] "2year"  "Mature"
```

```r
levels(fat$Breed)
```

```
## [1] "Ayrshire"          "Canadian"          "Guernsey"          "Holstein-Fresian"
## [5] "Jersey"
```

```r
summary(fat_model_additive)
```

```
##
## Call:
## lm(formula = Butterfat ~ Breed + Age, data = fat, contrasts = list(Breed = "contr.sum",
##     Age = "contr.sum"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0202 -0.2373 -0.0640  0.2617  1.2098
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.48210    0.04138 108.321  < 2e-16 ***
## Breed1      -0.42210    0.08276  -5.101 1.75e-06 ***
## Breed2      -0.04360    0.08276  -0.527    0.600
## Breed3       0.46790    0.08276   5.654 1.68e-07 ***
## Breed4      -0.81260    0.08276  -9.819 4.45e-16 ***
## Age1        -0.05230    0.04138  -1.264    0.209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4138 on 94 degrees of freedom
## Multiple R-squared:  0.6825, Adjusted R-squared:  0.6656
## F-statistic: 40.41 on 5 and 94 DF,  p-value: < 2.2e-16
```

Convert the two features into 'factor' type and reparameterize the model to full rank model by sum to zero contrast.

```r
#(b)
age_difference=2*coef(fat_model_additive)[['Age1']]
age_difference
```

```
## [1] -0.1046
```

In the model converted by the method of’sum to zero contrast’, the meaning of the coefficient of ‘Age1’ is (mu_2year-mu_population), and the opposite number of the coefficient of ‘Age1’ is (mu_mature-mu_population), so the difference between the butter fat content of two-year old cows and mature cows is 2*coefficient of ‘Age1’ in the model.

So the result is -0.1046, which the means of the butter fat content of 2-year old cows is lower than that of the mature cows.

```
#(c)
library(gmodels)
fat_model_interaction=lm(Butterfat~Breed*Age,data = fat)
# fit the model with interaction by default parameters (contast.treatment)
summary(fat_model_interaction)
```

```
##
## Call:
## lm(formula = Butterfat ~ Breed * Age, data = fat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0190 -0.2720 -0.0430  0.2372  1.3170
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       3.9660     0.1316  30.143  < 2e-16 ***
## BreedCanadian                     0.5220     0.1861   2.805  0.00616 **
## BreedGuernsey                     0.9330     0.1861   5.014 2.65e-06 ***
## BreedHolstein-Fresian            -0.3030     0.1861  -1.628  0.10693
## BreedJersey                       1.1670     0.1861   6.272 1.22e-08 ***
## AgeMature                         0.1880     0.1861   1.010  0.31503
## BreedCanadian:AgeMature          -0.2870     0.2631  -1.091  0.27834
## BreedGuernsey:AgeMature          -0.0860     0.2631  -0.327  0.74457
## BreedHolstein-Fresian:AgeMature  -0.1750     0.2631  -0.665  0.50773
## BreedJersey:AgeMature             0.1310     0.2631   0.498  0.61982
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.4161 on 90 degrees of freedom
## Multiple R-squared:  0.6926, Adjusted R-squared:  0.6619
## F-statistic: 22.53 on 9 and 90 DF,  p-value: < 2.2e-16
```

```
#[mu_(mature,Jersey)-mu_(mature,Guernsey)] we calculate by estimable function :
estimable(fat_model_interaction, c(0,0,-1,0,1,0,0,-1,0,1), conf.int=0.95)
```

```
##                         Estimate Std. Error  t value DF   Pr(>|t|)   Lower.CI
## (0 0 -1 0 1 0 0 -1 0 1)    0.451  0.1860713 2.423803 90 0.01735937 0.08133698
##                         Upper.CI
## (0 0 -1 0 1 0 0 -1 0 1) 0.820663
```

```
#CI for 95% : [0.08133698,0.820663]
```

```r
#(d)
library(car)
```

```
## Loading required package: carData
```

```r
C = matrix(c(0,0,0,0,0,1,1,0,0,0), 1, 10, byrow=T)
linearHypothesis(fat_model_interaction, C)
```

```
## Linear hypothesis test
##
## Hypothesis:
## AgeMature  + BreedCanadian:AgeMature = 0
##
## Model 1: restricted model
## Model 2: Butterfat ~ Breed * Age
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     91 15.629
## 2     90 15.580  1  0.049005 0.2831  0.596
```

Specifically,we are asked to test whether the equation :[mu_(mature,Canadian)-mu_(2year,Canadian)]=0 is significant or not

The P-value is large,so we do not have enough evidence to reject the Null hypothesis that the butter fat content of Canadian cows has no dependence on age.

```r
#(e)
anova(fat_model_additive,fat_model_interaction)
```

```
## Analysis of Variance Table
##
## Model 1: Butterfat ~ Breed + Age
## Model 2: Butterfat ~ Breed * Age
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     94 16.094
## 2     90 15.580  4   0.51387 0.7421 0.5658
```

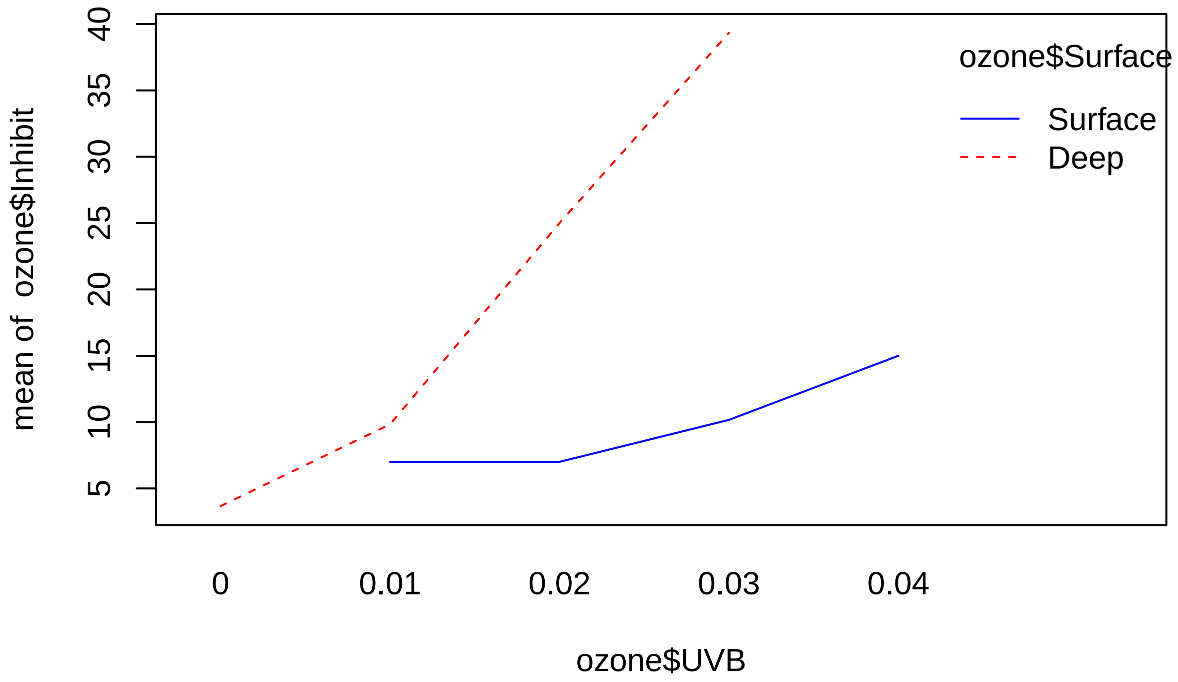
Interaction term is not significant ,so we can remove the interaction term and then fit an additive model.

Q2

```
#2(a)
ozone=read.csv('/Users/guanmuhan/Downloads/ozone.csv')
ozone$Surface=as.factor(ozone$Surface)
levels(ozone$Surface)
```

```
## [1] "Deep"    "Surface"
```

```
interaction.plot(ozone$UVB,ozone$Surface ,ozone$Inhibit,col = c('red','blue'))
```



The two lines are not parallel,specifically, the change in the mean of **response** when **UVB** change from **0.01 to 0.02(or 0.03)** given the depth level is **'deep'** is not equal to the change in the mean of **response** when **UVB** change from **0.01 to 0.02(or 0.03)** given the surface is **'surface'** , hence which means there is evidence for interaction between two factor.

```
#2(b)

y=as.matrix(ozone$Inhibit)
X.treatment=model.matrix(~Surface*UVB,data=ozone)
#get model matrix by treatment contrasts
X.treatment
```

```
##    (Intercept) SurfaceSurface  UVB SurfaceSurface:UVB
## 1            1              0 0.00               0.00
## 2            1              0 0.00               0.00
## 3            1              0 0.01               0.00
## 4            1              1 0.01               0.01
## 5            1              1 0.02               0.02
## 6            1              1 0.03               0.03
## 7            1              1 0.04               0.04
## 8            1              0 0.01               0.00
## 9            1              0 0.00               0.00
## 10           1              1 0.03               0.03
## 11           1              1 0.03               0.03
## 12           1              0 0.01               0.00
## 13           1              0 0.03               0.00
## 14           1              1 0.04               0.04
## 15           1              0 0.02               0.00
## 16           1              0 0.03               0.00
## 17           1              0 0.03               0.00
## attr(,"assign")
## [1] 0 1 2 3
## attr(,"contrasts")
## attr(,"contrasts")$Surface
## [1] "contr.treatment"
```

```
cat('----------------------------------------------------' ,"\n")
```

```
## ----------------------------------------------------
```

```
b=solve(t(X.treatment)%*%X.treatment)%*%t(X.treatment)%*%y
cat('Tntercept:' , b[1],"\n")
```

```
## Tntercept: 1.180556
```

```
cat('Surface:' , b[2],"\n")
```

```
## Surface: 1.277778
```

```
cat('UVB' , b[3],"\n")
```

```
## UVB 1226.389
```

```
cat('SurfaceSurface:UVB' , b[4],"\n")
```

```
## SurfaceSurface:UVB -939.9306
```

```
cat('dimension of matrix' , dim(X.treatment),"\n")
```

```
## dimension of matrix 17 4
```

```
Hat_matrix=X.treatment%*%solve(t(X.treatment)%*%X.treatment)%*%t(X.treatment)
SSRes=t(y)%*%y-t(y)%*%Hat_matrix%*%y
cat('Sum of squares of Residues:' , SSRes,"\n")
```

```
## Sum of squares of Residues: 1014.311
```

```
s2=SSRes/(17-4)
cat('S2:' , s2,"\n")
```

```
## S2: 78.0239
```

```
cat('-----------------------------------------------------' ,"\n")
```

```
## ----------------------------------------------------
```

```
summary(lm(ozone$Inhibit~Surface*UVB,data=ozone))
```

```
##
## Call:
## lm(formula = ozone$Inhibit ~ Surface * UVB, data = ozone)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.9722  -3.9444  -0.1806   1.4479  21.0278
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.181      4.292   0.275 0.787599
## SurfaceSurface         1.278     11.066   0.115 0.909837
## UVB                 1226.389    232.773   5.269 0.000152 ***
## SurfaceSurface:UVB -939.931    409.839  -2.293 0.039134 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 8.833 on 13 degrees of freedom
## Multiple R-squared:  0.7086, Adjusted R-squared:  0.6414
## F-statistic: 10.54 on 3 and 13 DF,  p-value: 0.000868
```

x

```
#2(c)
x=c(1,0,0.02,0)
y_predict=x%*%b
cat('the predicted percent inhibition of primary phytoplankton production in water ' , y_predict,"\n")
```

```
## the predicted percent inhibition of primary phytoplankton production in water  25.70833
```

Q3

```
#3(a)
conc = c(0.8,1,1.2,1.4,1.6,2.5)
m = c(7,5,6,6,4,2)
satis = c(1,1,4,4,4,2)
anaesthetic= data.frame(conc = conc,m= m,satis = satis)
#eta=beta0+beta1*concentration
l <- function(tha, y,n, conc) {
eta <- tha[1] + tha[2]*conc
return(sum(y*eta - n*log(1 + exp(eta))))
}
(betahat <- optim(c(10, -0.1), l, y = anaesthetic$satis,
conc = anaesthetic$conc,n = anaesthetic$m,
control = list(fnscale = -1,reltol=1e-16))$par)
```

```
## [1] -6.468675  5.566762
```

```
cat('try different original parameters:'  ,"\n")
```

```
## try different original parameters:
```

```
(betahat <- optim(c(0, 0), l, y = anaesthetic$satis,
conc = anaesthetic$conc,n = anaesthetic$m,
control = list(fnscale = -1,reltol=1e-16))$par)
```

```
## [1] -6.468675  5.566762
```

```
# so the beta0_hat is -6.468675, beta1_hat 5.566762
```

```
#3(b)
ilogit = function(x) 1/(1+exp(-x))
phat = ilogit(betahat[1]+betahat[2]*anaesthetic$conc)#estimation of reponse
#fisher information matrix:
I11= sum(anaesthetic$m*phat*(1 - phat))
I12 = sum(anaesthetic$m*anaesthetic$conc*phat*(1 - phat))
I22 = sum(anaesthetic$m*anaesthetic$conc^2*phat*(1 - phat))
Iinv = solve(matrix(c(I11, I12, I12, I22), 2, 2))
Iinv
```

```
##           [,1]      [,2]
## [1,]  5.849531 -4.848630
## [2,] -4.848630  4.176661
```

```
cat('----------------------------------------------------' ,"\n")
```

```
## ----------------------------------------------------
```

```r
#matrix(c(I11, I12, I12, I22), 2, 2)
sdp =c(sqrt(Iinv[1,1]),sqrt(Iinv[2,2]))

cat('Standrad deviation for beta0 and beta1' , sdp,"\n")
```

```
## Standrad deviation for beta0 and beta1 2.41858 2.043688
```

```r
q95=qnorm(0.975)
betahat_1_lower=betahat[1]-q95*sdp[1]
betahat_1_upper=betahat[1]+q95*sdp[1]
cat('95%CI for beta0:' , betahat_1_lower, betahat_1_upper,"\n")
```

```
## 95%CI for beta0: -11.20901 -1.728344
```

```r
betahat_2_lower=betahat[2]-q95*sdp[2]
betahat_2_upper=betahat[2]+q95*sdp[2]
cat('95%CI for beta1:',betahat_2_lower, betahat_2_upper)
```

```
## 95%CI for beta1: 1.561207 9.572317
```

```r
#3(c)
x=cbind(anaesthetic$m,anaesthetic$conc)
y=anaesthetic$satis
n=anaesthetic$m
#generate the explainatory variables and response and n
ylogxy=function(x,y) ifelse(y==0, 0, y*log(x/y))
#Deviance for the full model
D=-2*sum(ylogxy(n*phat,y)+ylogxy(n*(1-phat),n-y))
# DF for the full model
df=length(y)-length(betahat)

phatN=sum(y)/sum(n)
# Deviance for the null model
DN = -2*sum(ylogxy(n*phatN, y) + ylogxy(n*(1-phatN), n - y))
# DF for the null model
DfN=length(y)-1
#Likelihood Ratio Test
pchisq(DN-D,DfN-df,lower=FALSE)
```

```
## [1] 0.0002143095
```

Firstly we need to calculate deviances, and then take the difference of deviances and evaluate if it follows a chi-squared distribution. Under the assumption that the H0 is correct, this difference follows a chi-squared distribution with n-k degrees of freedom, where n is the number of observations and k is the number of parameters.

```r
#3(d)
q95=qnorm(0.975)
phat_predict= ilogit(betahat[1]+betahat[2]*2)
cat('estimation of reponse:',phat_predict, '\n')
```

```
## estimation of reponse: 0.9906673
```

```
si2=matrix(c(1,2),1,2)%*%Iinv%*%matrix(c(1,2),2,1)
cat('estimated variance of linear predictor estimate at concentration equals to 2:',si2, '\n')
```

```
## estimated variance of linear predictor estimate at concentration equals to 2: 3.161655
```

```
phat_predict
```

```
## [1] 0.9906673
```

```
#95% ci lower
ilogit(betahat[1]+betahat[2]*2-q95*sqrt(si2))
```

```
##           [,1]
## [1,] 0.764917
```

```
#95% ci upper
ilogit(betahat[1]+betahat[2]*2+q95*sqrt(si2))
```

```
##            [,1]
## [1,] 0.9997113
```

```
#95% CI [0.764917,0.9997113]
```