hw3

Huang Fang ID:913439658

January 30, 2016

**Q23.13**

kidney = read.csv("/Users/fangh/academic/cpp/kidney.csv", header = TRUE, sep = ',')  
kidney = kidney[ , -c(1:2)]  
head(kidney)

## Days Duration Weight X.2  
## 1 0 1 1 1  
## 2 2 1 1 2  
## 3 1 1 1 3  
## 4 3 1 1 4  
## 5 0 1 1 5  
## 6 2 1 1 6

kidney$Duration = factor(kidney$Duration, levels = c(1, 2), labels = c('short', 'long'))  
kidney$Weight = factor(kidney$Weight, levels = c(1, 2, 3), labels = c('mild', 'moderate', 'substantial'))  
kidney$Days = log(kidney$Days + 1)/log(10)  
kidney\_new = kidney[!(kidney$Duration == "long" & kidney$Weight == "mild"), ]  
kidney\_new$XAij1 = rep(-1, nrow(kidney\_new))  
kidney\_new$XAij1[kidney\_new$Duration == "short"] = 1  
kidney\_new$XBij1 = rep(0, nrow(kidney\_new))  
kidney\_new$XBij1[kidney\_new$Weight == "mild"] = 1  
kidney\_new$XBij1[kidney\_new$Weight == "substantial"] = -1  
kidney\_new$XBij2 = rep(0, nrow(kidney\_new))  
kidney\_new$XBij2[kidney\_new$Weight == "moderate"] = 1  
kidney\_new$XBij2[kidney\_new$Weight == "substantial"] = -1

**a.**  
Full regression model:  
  
where equals 1 when i = 1, equals -1 when Duration is i = 2; equals 1 when j = 1, equals 0 when j = 2, equals -1 when j = 3; equals 0 j = 1, equals 1 when j = 2, equals -1 when j = 3;

Reduced model to test factor A's main effect:  
, the definition of is just the same as the full model.

Reduced model to test factor B's main effect:  
, the definiation of is just the same the full model.

kidney\_new2 = kidney\_new[,c(1, 5, 6, 7)]  
#Full model  
fit\_full = lm(Days ~ ., data = kidney\_new2)  
fit\_full

##   
## Call:  
## lm(formula = Days ~ ., data = kidney\_new2)  
##   
## Coefficients:  
## (Intercept) XAij1 XBij1 XBij2   
## 0.66939 0.11733 -0.34323 0.02608

#Reduced for A  
fit\_A = lm(Days ~ .-XAij1, data = kidney\_new2)  
fit\_A

##   
## Call:  
## lm(formula = Days ~ . - XAij1, data = kidney\_new2)  
##   
## Coefficients:  
## (Intercept) XBij1 XBij2   
## 0.70850 -0.26502 -0.01303

#Reduced for B  
fit\_B = lm(Days ~ XAij1, data = kidney\_new2)  
fit\_B

##   
## Call:  
## lm(formula = Days ~ XAij1, data = kidney\_new2)  
##   
## Coefficients:  
## (Intercept) XAij1   
## 0.75520 0.03152

For full model:  
  
Reduced model for testing A's main effect:  
  
Reduced model for testing B's main effect:

**b.**

summary(fit\_full)

##   
## Call:  
## lm(formula = Days ~ ., data = kidney\_new2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.57814 -0.20825 0.02878 0.26140 0.51076   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.66939 0.04940 13.551 < 2e-16 \*\*\*  
## XAij1 0.11733 0.04940 2.375 0.021766 \*   
## XBij1 -0.34323 0.08067 -4.255 0.000102 \*\*\*  
## XBij2 0.02608 0.06377 0.409 0.684511   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3124 on 46 degrees of freedom  
## Multiple R-squared: 0.3722, Adjusted R-squared: 0.3313   
## F-statistic: 9.091 on 3 and 46 DF, p-value: 7.758e-05

summary(aov(Days ~ Weight + Duration, data = kidney\_new))

## Df Sum Sq Mean Sq F value Pr(>F)   
## Weight 2 2.111 1.0557 10.817 0.000141 \*\*\*  
## Duration 1 0.551 0.5506 5.641 0.021766 \*   
## Residuals 46 4.490 0.0976   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

For factor Duration:  
  
 = ((SSE(B) - SSE(A, B))/(df(R) - df(F)))/((SSE(A, B))/(df(F))) = 0.5551/0.0976 = 5.6875 if > F(0.95, 1, 46), we conclude , otherwise we conclude   
 = 5.6875 > F(0.95, 1, 46) = 4.05, so we conclude , the main effect of Duration is significant.  
The P-value is 0.0218

For factor Weight:  
  
 = ((SSE(A) - SSE(A, B))/(df(R) - df(F)))/(SSE(A, B)/df(F)) = 1.3072/0.0976 = 13.393  
if > F(0.95, 2, 46), we conclude , otherwise we conclude   
 = 13.393 > F(0.95, 2, 46) = 3.200, so we conclude , the main effect of Weight is significant.  
The P-value is 2.61e-05

**Q23.19**

element = c(0.73, 0.86, 0.94, 1.40, 1.62, 0.67, 0.75, 0.81, 1.32, 1.41, 0.15, 0.21, 0.26, 0.75, 0.78)  
block = rep(1:5, 3)  
fat = rep(1:3, each = 5)  
data = as.data.frame(cbind(block, fat, element))  
data$block = as.factor(data$block)  
data$fat = as.factor(data$fat)  
data2 = data[!(data$block == 1 & data$fat == 3) & !(data$block == 5 & data$fat == 1), ]

The ANOVA model is:

The equivalent regression model:  
  
where when i = 5, = 1 when i = k, otherwise = 0; = -1 when j = 3, = 1 when j = k, otherwise = 0.

names\_A = paste0("XAij", as.character(c(1:4)))  
names\_B = paste0("XBij", as.character(c(1:2)))  
data2[, names\_A] = rep(0, nrow(data2))  
data2[, names\_B] = rep(0, nrow(data2))  
for(i in 1:4){  
 data2[, names\_A[i]][data2$block == i] = 1  
 data2[, names\_A[i]][data2$block == 5] = -1  
}  
for(i in 1:2){  
 data2[, names\_B[i]][data2$fat == i] = 1  
 data2[, names\_B[i]][data2$fat == 3] = -1  
}  
data\_new = data2[, -c(1, 2)]  
fit = lm(element ~ ., data = data\_new)  
fit

##   
## Call:  
## lm(formula = element ~ ., data = data\_new)  
##   
## Coefficients:  
## (Intercept) XAij1 XAij2 XAij3 XAij4   
## 0.8294 -0.3361 -0.2227 -0.1594 0.3273   
## XBij1 XBij2   
## 0.2508 0.1626

The Full model:

**b.**  
The reduced model:  
  
The definition of is just the same as the full model.

fit\_r = lm(element ~ ., data = data\_new[, c(1:(ncol(data\_new) - 2))])  
fit\_r

##   
## Call:  
## lm(formula = element ~ ., data = data\_new[, c(1:(ncol(data\_new) -   
## 2))])  
##   
## Coefficients:  
## (Intercept) XAij1 XAij2 XAij3 XAij4   
## 0.8457 -0.1457 -0.2390 -0.1757 0.3110

**c.**

anova(fit)

## Analysis of Variance Table  
##   
## Response: element  
## Df Sum Sq Mean Sq F value Pr(>F)   
## XAij1 1 0.15603 0.15603 267.027 3.344e-06 \*\*\*  
## XAij2 1 0.16835 0.16835 288.129 2.673e-06 \*\*\*  
## XAij3 1 0.05024 0.05024 85.975 8.896e-05 \*\*\*  
## XAij4 1 0.34543 0.34543 591.186 3.181e-07 \*\*\*  
## XBij1 1 0.77223 0.77223 1321.629 2.889e-08 \*\*\*  
## XBij2 1 0.17845 0.17845 305.399 2.252e-06 \*\*\*  
## Residuals 6 0.00351 0.00058   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#With   
summary(aov(element ~ block + fat , data = data2))

## Df Sum Sq Mean Sq F value Pr(>F)   
## block 4 0.7200 0.1800 308.1 4.53e-07 \*\*\*  
## fat 2 0.9507 0.4753 813.5 4.96e-08 \*\*\*  
## Residuals 6 0.0035 0.0006   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#With no missing observation  
summary(aov(element ~ block + fat, data = data))

## Df Sum Sq Mean Sq F value Pr(>F)   
## block 4 1.4190 0.3547 146.9 1.61e-07 \*\*\*  
## fat 2 1.3203 0.6601 273.4 4.33e-08 \*\*\*  
## Residuals 8 0.0193 0.0024   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

= ((0.77223 + 0.17845)/2)/(0.00058) = 813.5 If > F(0.95, 2, 6), we conclude , otherwise we conclude .  
 = 819.5517 > F(0.95, 2, 6) = 5.14, so we conclude H\_a, the mean reduction in lipid level differ for the three diets.  
The result is the same as the condition when there is no missing oberservation. Although the F value increased drastically.

**d.**

summary(fit)

##   
## Call:  
## lm(formula = element ~ ., data = data\_new)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.026984 -0.014127 0.002487 0.014127 0.026984   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.829407 0.006900 120.20 2.24e-11 \*\*\*  
## XAij1 -0.336127 0.015648 -21.48 6.64e-07 \*\*\*  
## XAij2 -0.222741 0.012825 -17.37 2.34e-06 \*\*\*  
## XAij3 -0.159407 0.012825 -12.43 1.66e-05 \*\*\*  
## XAij4 0.327259 0.012825 25.52 2.39e-07 \*\*\*  
## XBij1 0.250847 0.010252 24.47 3.07e-07 \*\*\*  
## XBij2 0.162593 0.009304 17.48 2.25e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.02417 on 6 degrees of freedom  
## Multiple R-squared: 0.9979, Adjusted R-squared: 0.9958   
## F-statistic: 476.6 on 6 and 6 DF, p-value: 9.153e-08

mse\_root = 0.02417  
X = as.matrix(data\_new[, -1])  
#Covariance matrix  
cov\_co = solve(t(X) %\*% X) \* mse\_root^2

The Covariance matrix is:

cov\_co

## XAij1 XAij2 XAij3 XAij4  
## XAij1 2.412624e-04 -4.956754e-05 -4.956754e-05 -4.956754e-05  
## XAij2 -4.956754e-05 1.628648e-04 -3.186485e-05 -3.186485e-05  
## XAij3 -4.956754e-05 -3.186485e-05 1.628648e-04 -3.186485e-05  
## XAij4 -4.956754e-05 -3.186485e-05 -3.186485e-05 1.628648e-04  
## XBij1 -3.641697e-05 -3.540539e-06 -3.540539e-06 -3.540539e-06  
## XBij2 -1.062162e-05 7.081078e-06 7.081078e-06 7.081078e-06  
## XBij1 XBij2  
## XAij1 -3.641697e-05 -1.062162e-05  
## XAij2 -3.540539e-06 7.081078e-06  
## XAij3 -3.540539e-06 7.081078e-06  
## XAij4 -3.540539e-06 7.081078e-06  
## XBij1 1.046988e-04 -4.248647e-05  
## XBij2 -4.248647e-05 8.497293e-05

c = matrix(c(0, 0, 0, 0, 2, 1))  
t(c) %\*% cov\_co %\*% c

## [,1]  
## [1,] 0.0003338222

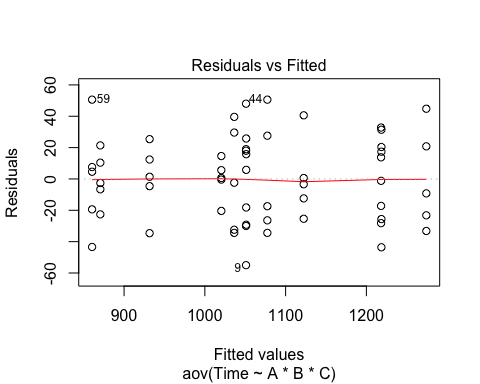
L = 0.664287 - qt(0.99, 6) \* 0.0182708 = 0.6068679  
U = 0.664287 + qt(0.99, 6) \* 0.0182708 = 0.7217061  
So our confidence interval for is (0.6068679, 0.7217061)

**Q24.12**  
Residuals:

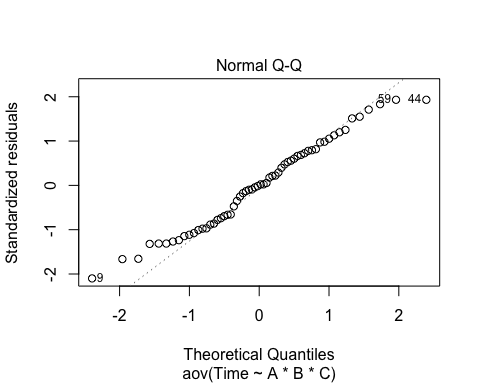
data = read.csv("~/hw3.csv", header = TRUE)  
data$A = as.factor(data$A)  
data$B = as.factor(data$B)  
data$C = as.factor(data$C)  
aov\_out = aov(Time ~ A\*B\*C, data = data)  
aov\_out$residuals

## 1 2 3 4 5 6 7 8 9 10 11 12   
## 31.4 -43.6 17.4 20.4 -25.6 -30.0 48.0 18.0 -55.0 19.0 44.8 -23.2   
## 13 14 15 16 17 18 19 20 21 22 23 24   
## -33.2 20.8 -9.2 -3.4 -12.4 0.6 -25.4 40.6 -1.2 -28.2 -17.2 13.8   
## 25 26 27 28 29 30 31 32 33 34 35 36   
## 32.8 -18.2 15.8 5.8 25.8 -29.2 29.6 39.6 -32.4 -34.4 -2.4 -6.6   
## 37 38 39 40 41 42 43 44 45 46 47 48   
## -22.6 10.4 21.4 -2.6 27.6 -34.4 -26.4 50.6 -17.4 -4.6 12.4 25.4   
## 49 50 51 52 53 54 55 56 57 58 59 60   
## -34.6 1.4 0.6 -0.4 14.6 -20.4 5.6 -19.4 4.6 -43.4 50.6 7.6

plot(aov\_out, which = 1)

 There is no obvious nonlinear pattern in our residuals vs fitted value plot, so the ANOVA model seems to be approperiate.

plot(aov\_out, which = 2)

 From this plot, we can find that the distribution of residuals is slightly heavy tailed, and the normality assumption holds.

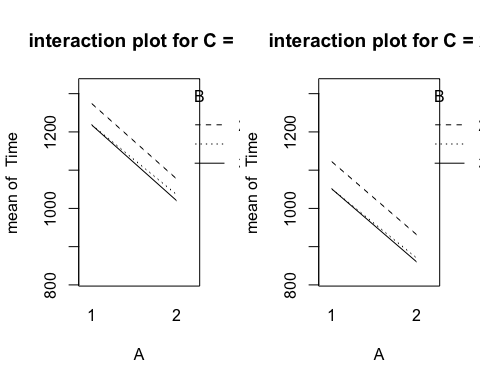
expvalue = rep(1, 60)  
for(i in 1:60)  
{  
 expvalue[i] = qnorm((i - 0.375)/(60 + 0.25), 0, 1)  
}  
ordered\_res = sort(aov\_out$residuals, decreasing = FALSE)  
cor(expvalue, ordered\_res)

## [1] 0.9916214

The coefficient of correlation between the ordered residuals and their expected value is 0.9916214, which is comparatively high, suggest that the normality assumption is reasonable.

**Q24.13**  
**a.**

par(mfrow = c(1, 2))  
with(data[data$C == 1, ], interaction.plot(A, B, Time, ylim = range(data$Time), main = "interaction plot for C = 1"))  
with(data[data$C == 2, ], interaction.plot(A, B, Time, ylim = range(data$Time), main = "interaction plot for C = 2"))

 It seems that the lines in our plot are parallel, so that there seems to be no interaction effect. But the main effect of both A, B and C seem to be significant.

**b.**  
The ANOVA Table.

summary(aov\_out)

## Df Sum Sq Mean Sq F value Pr(>F)   
## A 1 540361 540361 629.760 < 2e-16 \*\*\*  
## B 2 49320 24660 28.740 6.22e-09 \*\*\*  
## C 1 382402 382402 445.668 < 2e-16 \*\*\*  
## A:B 2 543 271 0.316 0.730   
## A:C 1 91 91 0.106 0.746   
## B:C 2 911 456 0.531 0.591   
## A:B:C 2 19 10 0.011 0.989   
## Residuals 48 41186 858   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

**c.**  
  
F^\* = MSABC/MSE = 10/858 = 0.01165501  
If   
 = 0.01165501 < F(0.95, 2, 48) = 3.190727, so we conclude , the three-factor interaction is insignificant.  
The P-value is 0.989.

**d.**  
Since our case is balanced, so the adjusted sums of squares and sequential sums of squares are the same.

options(contrasts = c("contr.sum","contr.poly"))  
aov\_out\_adj = aov(Time ~ A\*B\*C, data = data)  
drop1(aov\_out, .~., test="F")

## Single term deletions  
##   
## Model:  
## Time ~ A \* B \* C  
## Df Sum of Sq RSS AIC F value Pr(>F)   
## <none> 41186 415.89   
## A 1 82992 124178 480.11 96.7227 4.358e-13 \*\*\*  
## B 2 10379 51565 425.38 6.0482 0.004544 \*\*   
## C 1 70224 111410 473.60 81.8426 6.055e-12 \*\*\*  
## A:B 2 381 41567 412.44 0.2222 0.801596   
## A:C 1 4 41190 413.90 0.0047 0.945512   
## B:C 2 401 41587 412.47 0.2336 0.792578   
## A:B:C 2 19 41205 411.92 0.0111 0.988973   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#Since our case is balanced, so the adjusted sums of squares and sequential sums of squares are the same.

For AB:  
  
 = MSAB/MSE = 0.316  
If > F(0.95, 2, 48), then we conclude , otherwise we conclude   
 = 0.316 < F(0.95, 2, 48) = 3.19, so we conlcude , there is no interaction effect for .  
The P-value is 0.73

For AC:  
  
 = MSAC/MSE = 0.106  
If > F(0.95, 1, 48), then we conclude , otherwise we conclude   
 = 0.106 < F(0.95, 1, 48) = 4.04, so we conclude H\_0, there is no interaction effect for A.  
The P\_value is 0.746

For BC:  
  
 = MSBC/MSE = 0.531  
If > F(0.95, 2, 48), then we conclude , otherwise we conclude   
 = 0.531 < F(0.95, 2, 48) = 3.19, so we conclude , there is no interaction effect for BC.  
The P-value is 0.591

**e.**  
For A:  
  
 = MSA/MSE = 629.760  
If > F(0.95, 1, 48), then we conclude , otherwise we conclude   
 = 629.760 > F(0.95, 1, 48) = 4.04, so we conclude , the main effect of A is significant.  
The P-value < 2e-16

For B:  
  
 = MSB/MSE = 28.740  
If > F(0.95, 2, 48), then we conclude , otherwise we conclude   
 = 28.740 > F(0.95, 2, 48) = 3.19, so we conclude , the main effect of B is significant.  
The P-value is 6.22e-09

For C:  
  
 = MSC/MSE = 445.668  
If > F(0.95, 1, 48), then we conclude , otherwise we conclude   
 = 445.668 > F(0.95, 1, 48) = 4.04, so we conclude , the main effect of C is significant.  
The P-value < 2e-16

**f.**  
Conclusions:  
  
  
exist , exist , exist   
According to Kimball inequality.  
 < 1 - (1 - 0.05)^7 = 0.3016627 So the upper bound for the family level of significance for the set of tests is 0.3016627

**g.**  
The result from part(f) confirm my graphic analysis in part(a).

**Q24.14**  
**a.**

with(data, tapply(Time, list(A, B, C), mean))

## , , 1  
##   
## 1 2 3  
## 1 1218.6 1274.2 1218.2  
## 2 1036.4 1077.4 1020.4  
##   
## , , 2  
##   
## 1 2 3  
## 1 1051.0 1122.4 1051.2  
## 2 870.6 931.6 860.4

with(data, tapply(Time, list(A), mean))

## 1 2   
## 1155.9333 966.1333

with(data, tapply(Time, list(B), mean))

## 1 2 3   
## 1044.15 1101.40 1037.55

with(data, tapply(Time, list(C), mean))

## 1 2   
## 1140.867 981.200

= 1155.9333 - 966.1333 = 189.8  
 = 1044.15 - 1101.40 = -57.25  
 = 1044.15 - 1037.55 = 6.6  
 = 1101.40 - 1037.55 = 63.85  
 = 1140.867 - 981.200 = 159.667

mse\_root = sqrt(858)   
D1 = 189.8   
D2 = -57.25  
D3 = 6.6   
D4 = 63.85   
D5 = 159.667   
s1 = mse\_root\*sqrt(2)/sqrt(5\*3\*2)  
s2 = mse\_root\*sqrt(2)/sqrt(5\*2\*2)  
s3 = s1  
s1

## [1] 7.563068

s2

## [1] 9.262829

s3

## [1] 7.563068

B = qt(1 - 0.1/(2\*5), 48)  
B

## [1] 2.406581

c1 = c(D1-s1\*B, D1+s1\*B)  
c2 = c(D2-s2\*B, D2+s2\*B)  
c3 = c(D3-s2\*B, D3+s2\*B)  
c4 = c(D4-s2\*B, D4+s2\*B)  
c5 = c(D5-s3\*B, D5+s3\*B)

The confidence interval for is (171.5989, 208.0011)  
For is (-79.54175, -34.95825)  
For is (-15.69175, 28.89175)  
For is (41.55825, 86.14175)  
For is (141.4659, 177.8681)

**b.**  
 = 1020.4

#The interval  
s = mse\_root  
s = mse\_root/sqrt(5)  
mu = 1020.4  
c(mu - s\*qt(0.975, 48), mu + s\*qt(0.975, 48))

## [1] 994.0615 1046.7385

So the confidence interval for is (994.0615, 1046.7385)