Ch 12. Question 6.

Solution 1. *For part (a):*

The model for this Latin Square design is:

$$Y_{hqi} = \mu + \theta_h + \phi_q + \tau_i + \epsilon_{hqi}$$

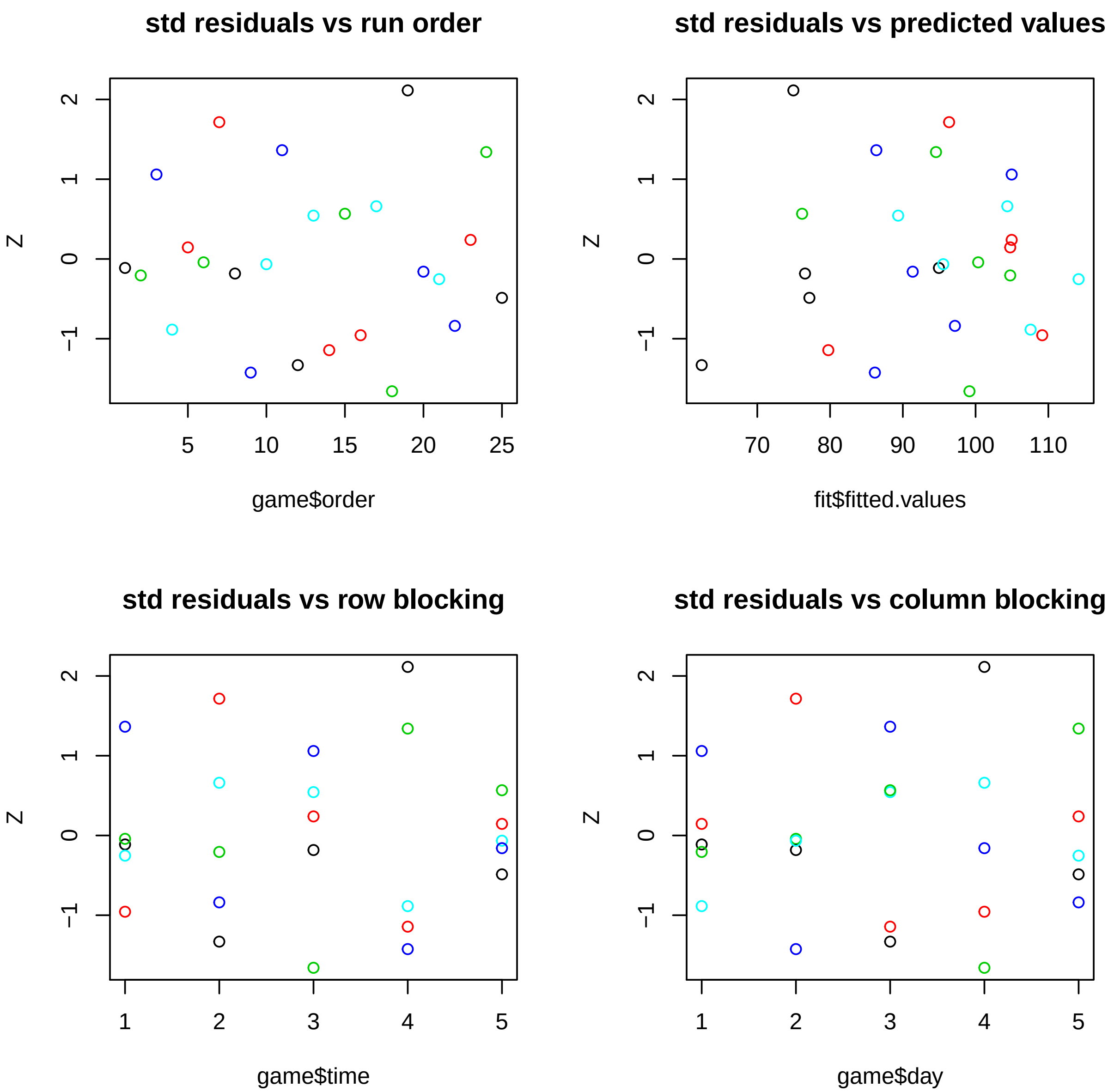
while $\epsilon_{hqi} \sim N(0, \sigma^2), i.i.d..$ *Also* $h = 1, \ldots, b; q = 1, \ldots, c; i = 1, \ldots, v; (h, q, i)$ *in the design.*

In our case, we have $b = c = v = 5$.

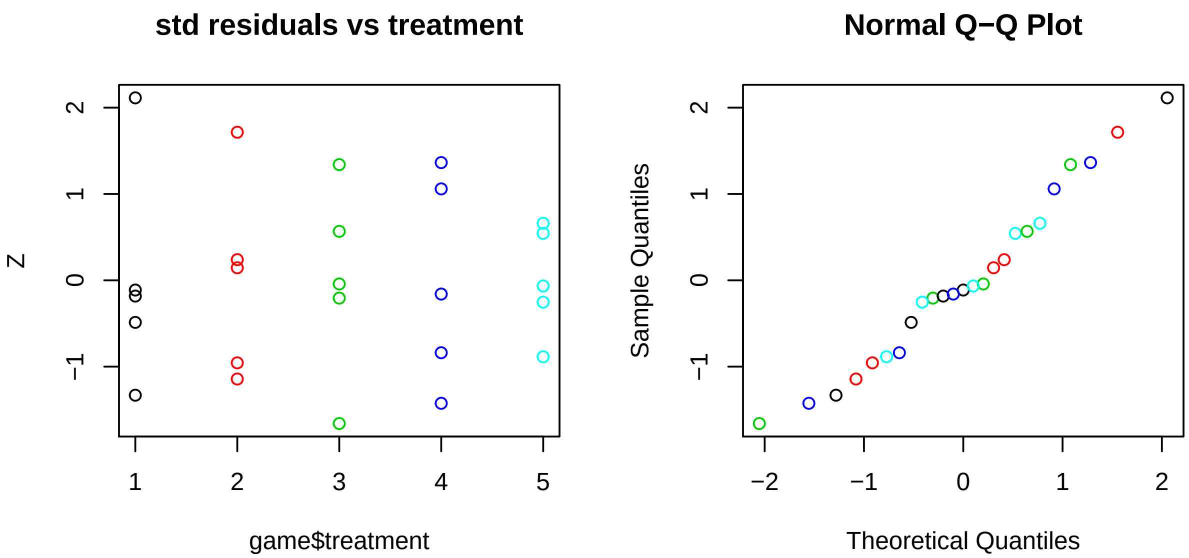
```
#read in the data
setwd("C:\\Akira\\R")
game = read.csv("game.csv", header = TRUE)

#fit the model
fit <- lm(response ~ factor(day) + factor(time) +
            factor(treatment), data = game)
anova(fit)

## Analysis of Variance Table
##
## Response: response
##                   Df  Sum Sq Mean Sq F value  Pr(>F)
## factor(day)        4 1711.44  427.86  2.9360 0.06611 .
## factor(time)       4  514.24  128.56  0.8822 0.50328
## factor(treatment)  4 1869.04  467.26  3.2064 0.05229 .
## Residuals         12 1748.72  145.73
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

to check model assumptions, we need the following plots:

```
#plot for model diagnostic
par(mfrow = c(2, 2))
SSE <- sum((game$response - fit$fitted.values)^2)
Z = fit$residuals/sqrt(SSE/24)
cols = factor(game$treatment)
plot(game$order, Z, col=as.character(cols))
title(main = "std residuals vs run order")
plot(fit$fitted.values, Z, col=as.character(cols))
title(main = "std residuals vs predicted values")
plot(game$time, Z, col = as.character(cols))
title(main = "std residuals vs row blocking")
plot(game$day, Z, col = as.character(cols))
title(main = "std residuals vs column blocking")
```




```
plot(game$treatment, Z, col = as.character(cols))
title(main = "std residuals vs treatment")
qqnorm(Z, col=as.character(cols))
```



To check independence, we look at residuals against run order(run order is from day 1 *time* 1 *until day* 5 *time* 5*). Since there is no pattern or trend, we consider the independence assumption valid.*

To check equal variance, we look at residuals against predicted value and residuals against treatment. Both of them do show a sign of non-equal variance. But keep in mind this is a latin square design and for either row block or column block it is randomized complete block design, so we have really small sample compared to blocks. Thus it is hard to tell.

To check normality, we look at QQ plot, and it does give a well formed straight line, so we consider normality assumption holds.

Finally, to see the effect of row or column blocks on the response, we look at residuals against row or column blocking. Both graphs shows no apparent difference of residuals among different row/column blocking factors, which implies that the row and column design here may not be necessary. In fact it matches with our observation from the anova table above. Both sum of squares for row and column blocks are smaller than the error sum of square. By rule of thumb we think the row and column blocks should not have effect on the response.

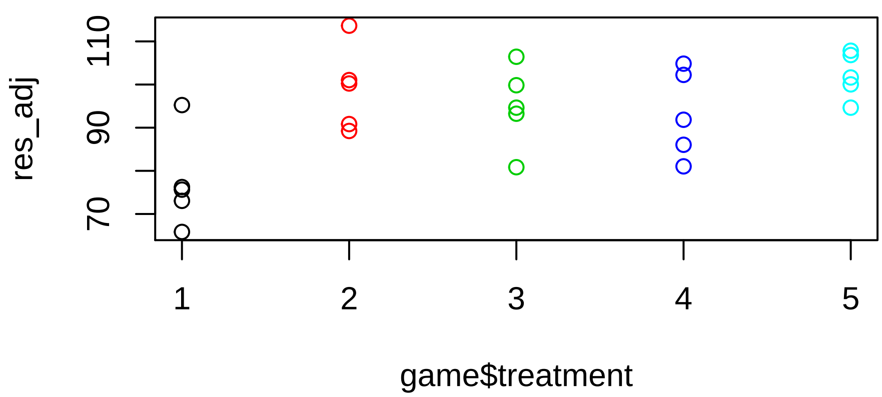
For part (b):

Under the latin square model, the adjusted data is:

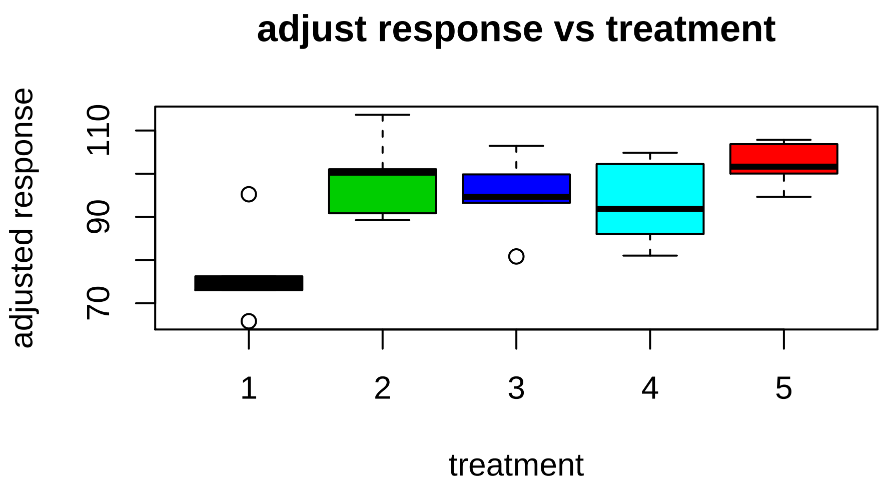
$$y^*_{hqi} = y_{hqi} - \left(\frac{1}{v}B_h - \frac{1}{v^2}G\right) - \left(\frac{1}{v}C_q - \frac{1}{v^2}G\right)$$

We compute and plot this adjusted value against treatment factor:

```
v <- 5
B <- as.vector(tapply(game$response, game$time, sum))
C <- as.vector(tapply(game$response, game$day, sum))
G <- sum(B)
res_adj <- game$response - (1/v*B[game$time] - 1/v^2*G) -
  (1/v*C[game$day] - 1/v^2*G)
plot(game$treatment, res_adj, col= as.character(cols))
```




```
boxplot(res_adj~game$treatment,main="adjust response vs treatment",
        xlab="treatment", ylab="adjusted response", col=as.character(cols))
```



From the plots above we see that there might be a treatment effect such that under the first treatment, the game performence is worse than the rest, and under the last treatment, the game performence is higher htan the rest. The anova table from part (*a*) *shows a marginal difference too* (*p value slightly larger than* 0.05).

For part (*c*):

For ANOVA table, we already did it in part (*a*).

For prat (*d*):

To check the effectiveness of block, we need to see the adjusted blocking effect.

For row blocking effect (adjust time order):

```
fit <- lm(response ~ factor(day) +
            factor(treatment) + factor(time), data = game)
anova(fit)

## Analysis of Variance Table
##
## Response: response
##                   Df  Sum Sq Mean Sq F value  Pr(>F)
## factor(day)        4 1711.44  427.86  2.9360 0.06611 .
## factor(treatment)  4 1869.04  467.26  3.2064 0.05229 .
## factor(time)       4  514.24  128.56  0.8822 0.50328
## Residuals         12 1748.72  145.73
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since sum square of time order is smaller than error sum of square, by rule of thumb we consider the time order does not have effect on the game performance.

For column blocking effect (adjust day):

```
fit <- lm(response ~ factor(time) + factor(treatment) +
            factor(day), data = game)
anova(fit)

## Analysis of Variance Table
##
## Response: response
##                   Df  Sum Sq Mean Sq F value  Pr(>F)
## factor(time)       4  514.24  128.56  0.8822 0.50328
## factor(treatment)  4 1869.04  467.26  3.2064 0.05229 .
## factor(day)        4 1711.44  427.86  2.9360 0.06611 .
## Residuals         12 1748.72  145.73
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since the sum square of day is smaller than error sum of square, by rule of thumb we consider the day order does not have effect on the game performance.

For part (e):

Under the Latin square model(without replication), the least square estimate for any contrast $\sum_i d_i\tau_i$ *is:*

$$\sum_i d_i\hat{\tau}_i = \frac{1}{v}\sum d_iT_i$$

We use Scheffe method here, and the $100(1-\alpha)\%$ *confidence interval is:*

$$\sum d_i\tau_i \in \Big(\frac{1}{v}\sum d_iT_i \pm \sqrt{(v-1)F_{v-1,(v-2)(v-1),\alpha}}\sqrt{msE\Big(\frac{\sum d_i^2}{v}\Big)}\Big)$$

So we have:

```
a<-c(1, -1, 0, 0, 0)
b<-c(1, 0, -1, 0, 0)
c<-c(1, 0, 0, -1, 0)
d<-c(1, 0, 0, 0, -1)
e<-c(0, 1, -1, 0, 0)
f<-c(0, 1, 0, -1, 0)
g<-c(0, 1, 0, 0, -1)
h<-c(0, 0, 1, -1, 0)
k<-c(0, 0, 1, 0, -1)
l<-c(0, 0, 0, 1, -1)
m<-c(0.33, 0.33, 0.33, -0.5, -0.5)
o<-c(0.25, 0.25, 0.25, 0.25, -1)
```

```
contrast <- list(a, b, c, d, e, f, g, h, k, l, m, o)
n <- length(contrast)
msE <- anova(fit)[[3]][[4]]
msE

## [1] 145.7267

#compute T_i
T <- as.vector(tapply(game$response, game$treatment, sum))
#compute Scheffe critical coefficient
w_S <- sqrt(4*qf(0.05, 4,12, lower.tail = FALSE))
w_S

## [1] 3.610632

#compute 95% confidence interval
for (i in 1:n){
#compute least square estimate
lse <- sum(contrast[[i]]*T)/5
#compute msd
msd <- w_S*sqrt(msE*sum(contrast[[i]]^2)/5)
#compute lower and upper bound for CI
ci_lower<- lse-msd
ci_upper<- lse+msd
#print the result of CI
cat("The CI for (", contrast[[i]], ") is (",
    ci_lower<- lse-msd, ", ", ci_upper<- lse+msd, ")", "\n")
}

## The CI for ( 1 -1 0 0 0 ) is ( -49.36657 ,  5.766574 )
## The CI for ( 1 0 -1 0 0 ) is ( -45.36657 ,  9.766574 )
## The CI for ( 1 0 0 -1 0 ) is ( -43.56657 ,  11.56657 )
## The CI for ( 1 0 0 0 -1 ) is ( -52.56657 ,  2.566574 )
## The CI for ( 0 1 -1 0 0 ) is ( -23.56657 ,  31.56657 )
## The CI for ( 0 1 0 -1 0 ) is ( -21.76657 ,  33.36657 )
## The CI for ( 0 1 0 0 -1 ) is ( -30.76657 ,  24.36657 )
## The CI for ( 0 0 1 -1 0 ) is ( -25.76657 ,  29.36657 )
## The CI for ( 0 0 1 0 -1 ) is ( -34.76657 ,  20.36657 )
## The CI for ( 0 0 0 1 -1 ) is ( -36.56657 ,  18.56657 )
## The CI for ( 0.33 0.33 0.33 -0.5 -0.5 ) is ( -25.92718 ,  9.519185 )
## The CI for ( 0.25 0.25 0.25 0.25 -1 ) is ( -32.89329 ,  10.69329 )
```

All the contrast above, the confidence intervals include 0*, from which we can conclude: there is no treatment difference between any two single treatments, and there is no treatment difference between the gameing with music versus without, and no difference between gaming with sound versus* *no sound.*

For part (f):

Based on what we analyzed in part $(a)-(e)$, we conclude that different backgorund music, or having music or not, having game sound or not, does not make a statistically significant difference on the performence on the game. So the professor should pick any arbitrary sound mode to play.