

Math650 Homework 10

Yu Huang

11-10-2006

Abstract

Question 4,5,6,10 of chapter 13, question 12 of chapter 13

1 question 13.4

The randomized block experiment is set to study the effects of treatments. If the purpose is to study block difference, controls from each block are enough.

2 question 13.5

In a balanced design, averages of row and column provide simple estimates of parameters. It also facilitates the study of the effect of each individual treatment or interaction because the block difference will be easily cancelled during addition or subtraction.

3 question 13.6

That's plainly what it means. Avoid talking about the effects of individual treatments because it's not supported.

4 question 13.10

In a randomized block design, the random allocation of treatment levels (certain levels and combinations) is carried out separately for each block. While in a completely randomized design, the random allocation of treatment levels is carried out to all experimental units at once. The levels correspond to all possible combinations of levels of distinct treatments.

In the randomized block design, there is generally no expectation for interaction and little interest in block difference. Purpose is straightforward. Just certain treatments. While in a complete randomized design, there is interest in effects of individual treatments, interactions and blocks.

5 question 13.12

5.1 common code to load data

```
data1 = read.csv("/usr/local/doc/statistical_sleuth/ASCII/case1301.csv")
```

```
cover_perc = 0.01*data1$COVER
LOGIT_COVER = log(cover_perc/(1- cover_perc))
data1$LOGIT_COVER = LOGIT_COVER
```

5.2 part a

```
mean_ls = c()
for (i in levels(data1$TREAT))
mean_ls = c(mean_ls, mean(data1[data1$TREAT==i,]$LOGIT_COVER))
msq_treatment = var(mean_ls)*16 #16 is dim(data1[data1$TREAT==i,])[1]
cat("mean square of treatment is ", msq_treatment, "\n")
```

The mean square of treatment is 19.39864.

5.3 part b

```
block_mean_ls = c()
for (i in levels(data1$BLOCK))
block_mean_ls = c(block_mean_ls, mean(data1[data1$BLOCK==i,]$LOGIT_COVER))
msq_block = var(block_mean_ls)*12 #12 is dim(data1[data1$BLOCK==i,])[1]
cat("mean square of block is ", msq_block, "\n")
```

The mean square of block is 10.89123.

5.4 part c

```
group_mean_ls = c()
for (i in levels(data1$BLOCK))
{
for (j in levels(data1$TREAT))
{
group_mean_ls = c(group_mean_ls, mean(data1[data1$BLOCK==i & data1$TREAT==j,]$LOGIT_COVER))
}
}
msq_between_groups = var(group_mean_ls)*2 #2 is number of replicates in each plot
cat("mean square between groups is ", msq_between_groups, "\n")
```

The mean square between groups is 4.009835.

5.5 part d

```
rss_interaction = msq_between_groups*47 - msq_treatment*5 - msq_block*7
cat("Interactions Sum of Squares is ", rss_interaction, "\n")
```

The interactions Sum of Squares is 15.23041.

6 check the orthogonality between factors for the Graeco-Latin square design (take $k = 4$)

For a Graeco-Latin square, there're 2 blocking factors (R for Row and C for Column) and 2 treatment factors (G for Graeco and L for Latin).

To check orthogonality between any two factors, count how many entries belong to the combination of these two factors (which is always 1 in this case) and count how many entries belong to either of the two factors (which is always 4 in this case).

For example, check orthogonality between row-blocking factor and G ttt factor. For each combination of row-blocking factor and G ttt factor, there's only 1 entry. There're 4 entries for each row-blocking factor and 4 for each G ttt factor. So $\frac{1}{16} = \frac{4}{16} * \frac{4}{16}$, which satisfies **proportional frequency**.

Basically, the rule of **proportional frequency** is

$$P(\text{factor1} = \text{value1}, \text{factor2} = \text{value2}) = P(\text{factor1} = \text{value1}) * P(\text{factor2} = \text{value2})$$

In this Graeco-Latin square design, it's always met.