

MA50259: Statistical Design of Investigations

Dr. Sandipan Roy

Lecture 6: Blocking in Factorial Designs

Factorial Designs in Blocks

- ▶ Blocking is even more effective when combined with a factorial design in treatment factors
- ▶ The number of experimental units in each block must be equal to the product of levels of all the factors. This is called a **randomized complete block factorial (RCBF)**
- ▶ The model for analysis of a two-factor factorial in a randomized block design is given by

$$y_{ijk} = \mu + b_i + \alpha_j + \beta_k + (\alpha\beta)_{jk} + \epsilon_{ijk}$$

- ▶ Note there is an **interaction between the two factorial factors** (α_j and β_k)
- ▶ There is **no interactions between treatment factors and the block factors** b_i as there will be zero degrees of freedom for ssE

Efficiency of factorial designs in Blocks

```
# data in RCBF data.frame
mod.blocks <- aov( response ~ block + factor1* factor2, data = RCBF)
summary(mod.blocks)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
block	1	47.6	47.6	18.372	0.00363 **
factor1	3	33.0	11.0	4.240	0.05274 .
factor2	1	422.3	422.3	162.961	4.19e-06 ***
factor1:factor2	3	40.3	13.4	5.189	0.03368 *
Residuals	7	18.1	2.6		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```
mod.no.blocks <- aov( response ~ factor1 * factor2, data = RCBF)
summary(mod.no.blocks)
```

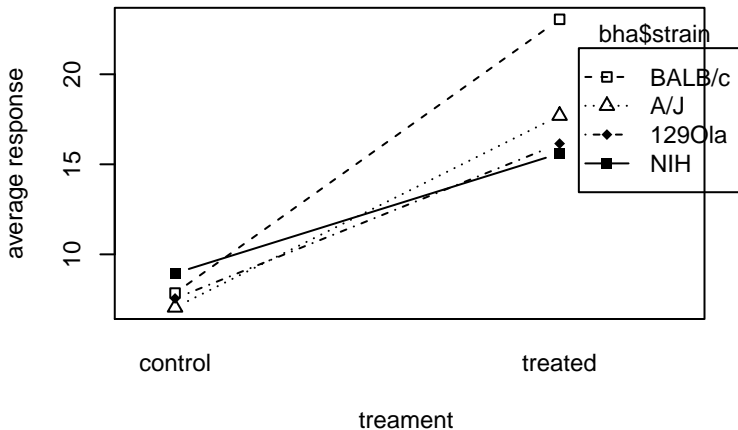
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
factor1	3	33.0	11.0	1.337	0.329
factor2	1	422.3	422.3	51.383	9.54e-05 ***
factor1:factor2	3	40.3	13.4	1.636	0.257
Residuals	8	65.8	8.2		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

- ▶ The estimated variance has been reduced from 8.2 to 2.6 so that $8.2/2.6 = 3.15$
- ▶ It would take more than three times as many EUs to have the same power if the experiments were not blocked

Factorial Designs in Blocks

```
interaction.plot(bha$treat, bha$strain, bha$y, type = "b", pch = c(18,24,22,15), leg.bty = "o",  
main = "", xlab = "treatment", ylab = "average response")
```



Generalized Complete Block Design

- ▶ When experimental units (EUs) represent physical entities, smaller blocks of EUs usually result in greater homogeneity.
- ▶ The larger the block, the more likely it is to have many heterogeneous EUs. Then is not advised to have blocked designs with more the number of treatment levels or number of combination of levels of treatment factors
- ▶ However, sometimes the EUs represent trials rather than physical entities and the runs can be made quickly, then larger block sizes may not increase the variability of EUs within a block
- ▶ In that case, a design with replicates of each treatment level within a block, **a generalized complete block design (GCB)**, can be used

Generalized Complete Block Design

- ▶ Since there are replicate EUs for each treatment in each block, it is possible to fit the model with interactions between blocks and treatment factors. For an RCB we can use model:

$$y_{ijk} = \mu + b_i + \tau_j + (b\tau)_{ij} + \epsilon_{ijk}$$

- ▶ Interactions are meaningful if block factors are of interest!
- ▶ **Problem:** If block factors are not interest and the interaction between blocks and treatment factor is significant, its interpretation would imply that treatment effect is different for different blocks!
- ▶ In that case we need to use a **special test statistic** for the null hypothesis of no treatment effect $H_0 : \tau_i = 0$ for all i
- ▶ the test should use the **block by treatment interaction mean square as the denominator of the F-test!**

Generalized Complete Block Design

```
# data in GCB data.frame
inter<-interaction(GCB$block,GCB$treatment,sep=":")
X1<-model.matrix(~block-1,GCB);X2<-model.matrix(~treatment-1,GCB);X3<-model.matrix(~inter-1)
X<-cbind(1,X1,X2,X3); X0<-cbind(1,X1,X2); X00<-cbind(1,X1)

G<-ginv(t(X)%*%X); G0<-ginv(t(X0)%*%X0); G00<-ginv(t(X00)%*%X00)

y<-GCB$response; n<-length(y)
resid<-y-X%*%G%*%t(X)%*%y; ssE<-t(resid)%*%resid %>% as.vector;
ssE
[1] 7340.752
resid0<-y-X0%*%G0%*%t(X0)%*%y;ssE0<-t(resid0)%*%resid0 %>% as.vector;
ssE0-ssE
[1] 2356.1
resid00<-y-X00%*%G00%*%t(X00)%*%y; ssE00<-t(resid00)%*%resid00 %>% as.vector;
ssE00-ssE0
[1] 1723.932

Fval<-((ssE00-ssE0)/(2))/((ssE0-ssE)/(16));Fval
[1] 5.85351
p_value<-pf(Fval,2,16,lower.tail = FALSE);p_value
[1] 0.01236634

# this analysis is incorrect
aov(response~block*treatment,GCB) %>% summary
      Df Sum Sq Mean Sq F value    Pr(>F)
block      8 124741    15593  229.406 < 2e-16 ***
treatment    2   1724      862  12.682 1.13e-05 ***
block:treatment 16   2356      147   2.166  0.0102 *
Residuals   108   7341       68
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Generalized Complete Block Design

Alternative codes

```
aov(response ~ treatment + Error(block/treatment), data = GCB) %>% summary
```

```
Error: block
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Residuals	8	124741	15593		

```
Error: block:treatment
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
treatment	2	1724	862.0	5.854	0.0124 *
Residuals	16	2356	147.3		

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Error: Within
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Residuals	108	7341	67.97		

```
cellmeans <- tapply( GCB$response, list(GCB$block, GCB$treatment), mean)
```

```
# tidyverse version
```

```
summ_GCB <- GCB %>%
```

```
group_by(block, treatment) %>% summarise(response = mean(response))
```

```
aov( response ~ block + treatment, summ_GCB) %>% summary
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
block	8	24948	3118.5	105.888	2.19e-12 ***
treatment	2	345	172.4	5.854	0.0124 *
Residuals	16	471	29.5		

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```