Experiment Design:

Experiment Design II:

- Confounding
- Split Plot und Split Block Design
- Incomplete Block Designs, Gitter

Data:

- Cc
- Pc
- Sp
- Sb
- BIBD

Verwendete Pakete:

- crossdes
- ibd

Faktorielle Experimente:



Beispiel 22:

"Düngung"; "Bewässerung"

- Erbsenanbau ohne / mit NPK-Düngung
- ohne / mit künstlicher Bewässerung
- Anlage als Completely Randomized Design:

	"00"	"01"	"10"	"11"
Ertrag:	53.13	28.66	33.21	6.42



- Main Effects:
 - A: [00, 01] vs. [10, 11]
 - B: [00, 10] vs. [01, 11]
- Interaction:
 - AxB: [00, 11] vs. [01, 10]

Faktorielle Experimente:



Effects	"00"	"01"	"10"	"11"	S+	S-	S total
	232.49	105.96	148.03	84.27			
B: -/+ Wasser	+	-	+	-	380.52	190.23	190.29
A: -/+ Dünger	+	+	-	-	338.45	232.3	106.15
AXB	+	-	-	+	316.76	253.99	62.77
total	+	+	+	+	570.75		570.75

• Main Effects:

A: [00, 01] vs. [10, 11]B: [00, 10] vs. [01, 11]

• Interaction:

• AxB: [00, 11] vs. [01, 10]

Faktorielle Experimente:

Effects	"00"	"01"	"10"	"11"	S+	S-	S total
	232.49	105.96	148.03	84.27			
B: -/+ Wasser	+	-	+	-	380.52	190.23	190.29
A: -/+ Dünger	+	+	-	-	338.45	232.3	106.15
AXB	+	-	-	+	316.76	253.99	62.77
total	+	+	+	+	570.75		570.75

Warum ist 1+1 = 0?!

2² (2 Faktoren á 2 Level):

• Main effects: $A_{i=0}$: 00 01

A_{i=1}: 10 11

 $B_{j=0}$: 00 10

B_{i=1}: 01 11

• Interaktion: AB_{i+i=0}: 00 11

AB_{i+i=1}: 01 10

• Main Effects:

• A: [00, 01] vs. [10, 11]

• B: [00, 10] vs. [01, 11]

Interaction:

• AxB: [00, 11] vs. [01, 10]

Faktorielle Experimente:

Rinäi	rzahle	nsvst	em:
Dilla	Laine	เเองอเ	LEIII.

Dilla	Dinaizaniensystem.								
	Vierer	Zweier	Einer						
0:	0	0	0						
1:	0	0	1						
2:	0	1	0						
3:	0	1	1						

2² (2 Faktoren á 2 Level):

- Main effects: $A_{i=0}$: 00 01
 - A_{i=1}: 10 11
 - B_{j=0}: 00 10
 - B_{i=1}: 01 11
- Interaktion: $AB_{i+i=0}$: 00 11
 - AB_{i+i=1}: 01 10

Warum ist 1+1 = 0?!

Faktorielle Experimente:

Binärzahlensystem:

	Vierer	Zweier	Einer
0:	0	0	0
1:	0	0	1
2:	0	1	0
3.	0	1	1

2² (2 Faktoren á 2 Level):

"Restoperator"

Modulo:

- Main effects: $A_{i=0}$: 00 01
 - A_{i=1}: 10 11
 - B_{i=0}: 00 10
 - B_{i=1}: 01 11
- Interaktion: $AB_{i+i=0}$: 00 11
 - AB_{i+j=1}: 01 10

Alternative: Die Modulo-Notation

- (0+0) / 2 = 0, Rest: 0
- (0+1) / 2 = 0, Rest: 1
- (1+1) / 2 = 1, Rest: 0

Α	В	С	АВ	ВС	AC	ABC
0	0	0	0			
0	1	0	1			
0	0	1				
0	1	1				
1	0	0				
1	1	0				
1	0	1				
1	1	1				

Faktorielle Experiment	e:	2 ³	F	acto	rial
	_	_		_	

	Α	В	С	0+0+	0 = 0	AC	ABC
	0	0	0	0	0	0	0
		nzieller Rech 2 = "1"	enweg:	1	0	1	
)/2 = "0"		1	1	1
	0	1	1	1	0	1	0
Carr	1	0	0	1	0	1	1
_) / 2 = "0			0	1	0	0
		("0" + 1) / 2 =	= <i>"1"</i>	1		1	0
	1	1	1	0	0	0	1

Confounding:

Arbeiten mit unvollständigen Blöcken

- Experimente verkleinern: Probenzahlen reduzieren
- Varianzen zu mindern, Arbeitszeit sparen, Kosten senken

Beispiel:

- 23 hat 8 treatments: 000 100 010 110 001 101 011 111;
- * 8 Individuen pro Wiederholung (d.h. 32 Individuen für 4 Wiederholungen im CR)
- Im RCB mit 2 Blöcken: 16 Individuen pro Wiederholung (64 pro 4 WDH)
- Alternative: Reduktion auf unvollständige Blöcke á 4 treatment-Kombinationen:

Complete confounding:

 wenn ein bestimmter Effekt (z.B. AxBxC) nicht wichtig ist, kann er vollständig mit Effekten unvollständiger Blöcke überlagert werden. Der Effekt ist dann nicht bestimmbar.

Beispiel:

- Erbsenanbau
- je 2 Level N, P, K
- 2 Blöcke á 4 treatments
- 3 Replikate á 2 Blöcke

Error:

"block"-Effekt verschachtelt (nested) in Wiederholung

Complete confounding:

> summary(aov(yield~Error(Rep/block)+N*P*K, data=cc))

```
Error: Rep
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 2
                       88.9
             177.8
Error: Rep:block
         Df Sum Sq Mean Sq F value Pr(>F)
             37.0
                      37.00
                              0.576 0.527
Residuals 2 128.5
                      64.25
Error: Within
          Df Sum Sq Mean Sq F value Pr(>F)
Ν
          1 189.28 189.28 12.259 0.00437
Ρ
              8.40
                      8.40
                              0.544 0.47490
Κ
             95.20
                      95.20
                              6.166 0.02880 *
N:P
             21.28
                      21.28
                              1.378 0.26317
          1
N:K
             33.13
                      33.13
                              2.146 0.16865
                      0.48
P:K
              0.48
                              0.031 0.86275
          1
Residuals 12 185.29
                      15.44
```

Partial confounding:

- wenn teilweise Information zu allen Effekten gewünscht ist
- in verschiedenen Replikaten verschiedene Effekte überlagern
- alle Mittelwerte sind adjusted means: da alle Effekte abschätzbar sind, sind auch Block-Effekte abschätzbar. Die Mittelwerte werden um die Blockeffekte korrigiert.

• 3 main effects bes. wichtig, daher nicht überlagert. Beispiel: Therapieerfolg • 4 Interaktionen AB, AC, BC, ABC jeweils in 1 Rep. überlagert in nicht überlagerten Rep. sind Effekte bestimmbar! 2 Cytostatika Replikat 4 Replikat 1 Replikat2 Replikat 3 (5-Br-U, Taxol) 110 011 000 010 100 101 111 000 2 Anti-Epileptika (Diazepam, Lactosamid) 111 101 000 101 001 010 100 011 100 001 111 010 2 Altergruppen 110 000 001 101 (alt, jung) 011 110 001 100 011 111 010 110 4 Replikate á 2 halbe Blöcke BC_0 BC₁ AB_0 AB_1 AC_1 AC_0 ABC₁ ABC₀ (Krankenhäuser Aachen, Berlin)

Partial confounding:

> summary(aov(success~age*cytostaticum*antiepilepticum+Error(Rep/hospital), data=pc))

Error: Rep

Df Sum Sq Mean Sq F value Pr(>F)

Residuals 3 228.6 76.2

Error: Rep:hospital

 age:cytostaticum
 Df
 Sum Sq
 Mean Sq

 age:antiepileptikum
 1
 2.0
 2.0

 cytostaticum:antiepileptikum
 1
 24.5
 24.5

 age:cytostaticum:antiepileptikum
 1
 18.0
 18.0

 age:cytostaticum:antiepileptikum
 1
 276.1
 276.1

Error: Withir

F value Df Sum Sq Mean Sq Pr(>F) 54.609 1.06e-06 ** 1 1391.3 age 1391.3 504.0 cytostaticum 504.0 19.784 0.000353 *** 17.370 0.000645 *** antiepileptikum 442.5 442.5 age:cytostaticum 18.4 18.4 0.721 0.407542 age:antiepileptikum 3.4 3.4 0.132 0.720371 cytostaticum:antiepileptikum 0.0 0.0 0.002 0.968213 age:cytostaticum:antiepileptikum 24.0 24.0 0.942 0.345368 1 Residuals 17 433.1 25.5

Missing data:

- Bei Complete confounding Abschätzung wie für RCB
- Bei Partial confounding sehr kompliziert das Ergebnis wird stark beeinflusst

Split Plot Design:

a2 a0

Kino

1

b2	b2	b1	b1
b0	b0	b2	b2
b1	b1	b0	b0

a1

a1 a0 a3 a2

b₀

b1

b₂

b1

b2

b0

a3

a2 a3 a0 a1

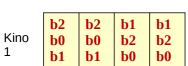
Kino 3

b2	b2	b0	b1
b1	b0	b2	b0
b0	b1	b1	b2

- wenn ein Faktor sehr große Layouts bedingt, können andere Faktoren als "Sub-Plots" darin angelegt werden
- z.B. Wirkung von Kinofilmen auf Zuschauergruppen unterschiedlicher Altersgruppen
 - Es ist nicht möglich, in einem Kino gleichzeitig mehrere Filme zu zeigen
 - Es können aber alle Altersgruppen im selben Kino sitzen.
 - Beispiel: 4 Filme; 3 Altersgruppen; 3 Kinos
 - Jeder Film wird einmal in jedem Kino gezeigt
 - Replikate für für a: 3; für b: 3 x 4=12

Split Plot Design:

a2 a0



a1

a3

a2

Kino

• Es handelt sich um ein faktorielles Experiment:

- a (Film) mit p levels
- b (Altersgruppe) mit q levels.

• a ist nicht zufällig angeordnet: "verschachtelt" im Plot-Effekt

Welcher Film ist der beste?

a1 a0 a3 a2 b0 b0 b1 b1 Kino b1 **b**2 b₂ **b0 b**2 **b0** b1

аЗ

In welchem Alter ist Kinobesuch beliebt? Wie wirkt welcher Film auf Vorteil: welche Altersgruppe?

• für B und AxB höhere Präzision als RCB

b2 b₂ **b0** b1

b1 **b0 b2 b0 b0 b1 b**2 b1

a0

a1

Nachteile:

(Kino)

- Präzision für A geringer als RCB
- großer Aufwand für fehlende Daten

Split Plot Design:

	a3	a1	a2	a0
Kino 1	b2 b0 b1	b2 b0 b1	b1 b2 b0	b1 b2 b0
	a1	a0	a3	a2
Kino 2	b1 b2 b0	b0 b2 b1	b0 b1 b2	b1 b2 b0
	a2	a3	a0	a1
Kino 3	b2 b1 b0	b2 b0 b1	b0 b2 b1	b1 b0 b2

ANOVA: source of var. df whole plot Replicate (r-1) = 2 A (p-1) 3 Error (a) (r-1)(p-1) split plot В (q-1)2 **AxB** (q-1)(p-1)= 6 Error (b) p(q-1)(r-1) = 16

aov(amusement ~ film * age + Error(kino/film)

Split Plot Design:

```
> summary(aov(amusement~film*age+Error(kino/film), data=sp))
Error: kino
         Df Sum Sq Mean Sq F value Pr(>F)
Residuals 2 8.167
                     4.083
Error: kino:film
         Df Sum Sq Mean Sq F value Pr(>F)
          3 23.19 7.731
                             0.368 0.779
Residuals 6 126.06 21.009
Error: Within
         Df Sum Sq Mean Sq F value
                                    Pr(>F)
         2 301.2 150.58 12.563 0.000525 ***
                             2.759 0.048978 *
film:age 6 198.4
                     33.06
Residuals 16 191.8
                     11.99
```

Split Plot Design: Identifikation

- Split Plot Designs werden häufig nicht erkannt und fälschlicherweise als RCB analysiert.
- Dabei wird die Trennschärfe für Faktor a überschätzt!

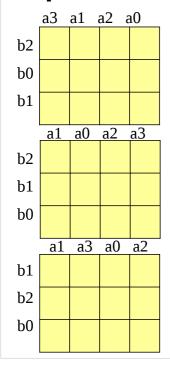
Beispiel: Zeitungsdruck

- Qualität der Bilder hängt ab von:
 - Papier
 - Walzendruck (high, low)
 - Druckgeschwindigkeit (s, m, l)
- Druck und Geschwindigkeit können leicht eingestellt werden
- Papierwechsel ist aufwändig

Erkennungszeichen:

- Immer wenn die Anordnung der Faktorkombinationen nicht völlig zufällig ist, handelt es sich nicht mehr um ein RCB Design!
- Wenn ein Faktor festgehalten wird, und andere "darin" zufällig angeordnet werden, liegt Split Plot vor
- aov(val ~ fix * var + Error(Rep/fix)

Split Block Design:



- wenn treatments a, und b, weniger wichtig sind als AxB
- a, in randomisierten Blocks oder latin square
- b_i wird in Streifen darüber gelegt

Weder a noch b werden völlig zufällig angeordnet!

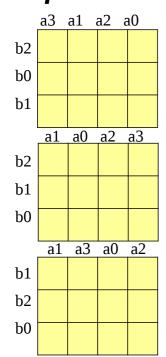
Vorteil:

- höhere Präzision für AxB als RCB
- Einheiten können klein sein

Nachteile:

- Präzision für A und B geringer als RCB
- Analyse aufwändiger

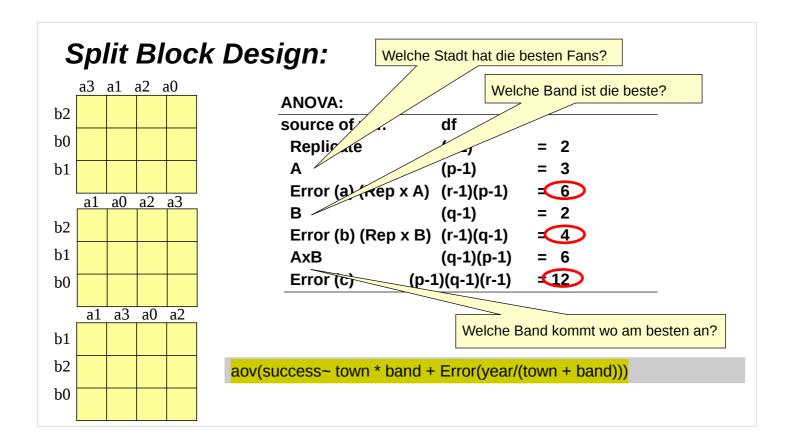
Split Block Design:



Beispiel: Tournee-Erfolg in Abhängigkeit von Tourplan und Programm-Abfolge

- Veranstaltungszeitraum: 3 Jahre (Replikate)
- 4 Städte in unterschiedlicher Reihenfolge (a)
- 3 Bands jeweils nacheinander (b)

aov(success~ town * band + Error(year/(town + band)))



Split Block Design:

```
> summary(aov(applause~town*band+Error(year/(town+band)), data=sb))
Error: year
         Df Sum Sq Mean Sq F value Pr(>F)
Residuals 2 202.2 101.1
Error: year:town
        Df Sum Sq Mean Sq F value Pr(>F)
          3 591.6 197.21
                           2.667 0.142
Residuals 6 443.6
                    73.94
Error: year:band
         Df Sum Sq Mean Sq F value Pr(>F)
         2 477.2 238.58
                          21.61 0.00718 **
Residuals 4 44.2
                    11.04
Signif. codes: 0 '*** 0.001 '** 0.01 '* 0.05 '.' 0.1 ' ' 1
Error: Within
        Df Sum Sq Mean Sq F value Pr(>F)
town:band 6 1521.3 253.55 3.194 0.0412 *
Residuals 12 952.7 79.391
```

Incomplete Block Designs:

- Wenn nicht genug Individuen vorhanden sind, um ein vollständiges RCB Design auszustatten
- * Wenn Blöcke aus sonstigen Gründen klein gehalten werden müssen

Bloc	c k 1	1	Block 2		Block3				
1	A		В			A			
]	В		С			С			
1	2		3		4	5		6	
A	A		A		В	В		С	
В	С		D		С	D)	D	
1			2		3			4	
A			В		С			D	
В			С		D			A	
С			D		A			В	

Balanced Incomplete Block Design:

- Die Anzahl der treatment levels (v) ist größer als die Größe der Blöcke (k): v > k
- Über alle Replikate (r) kommen alle level-Kombinationen gleich häufig vor: b*k = v*r
- Je nach Größe des Experiments ergibt sich
 I = r (k-1) / (v-1) als Anzahl der Kombinationen

Incomplete Block Designs:

- Wenn nicht genug Individuen vorhanden sind, um ein vollständiges RCB Design auszustatten
- Wenn Blöcke aus sonstigen Gründen klein gehalten werden müssen

V	b	r	k
4	4	3	3
4	6	3	2
5	5	4	4
5	10	4	2
5	10	6	3
•••	•••	•••	•••

BIBD in R:

- install.packages("crossdes")
- library(crossdes)
- find.BIB(treatments, blocks, blocksize)
- isGYD(find.BIB(treatments, blocks, blocksize)

Balanced Incomplete Block Designs:

Beispiel:

- 9 verschiedene Weine sollen verglichen werden
- Jeder Tester kann 3 Weine zuverlässig testen
- Mit 12 Testern kann ein BIBD gebildet werden
- Geschmacksnoten y: 0 100
- > summary(aov(Q~tester*wine, data=BIBD))

```
Df Sum Sq Mean Sq tester 11 1225.6 111.42 wine 8 1122.8 140.35 tester:wine 16 535.9 33.49
```

```
> find.BIB(9,12,3)
  [,1] [,2] [,3]
[1,]
        1
[2,]
        4
         5 8
3 5
2 6
1 3
1 6
2 4
[3,]
                   9
[4,]
 [5,]
                   7
 [6,]
                   4
                  9
8
[7,]
[8,]
[9,]
         1
            3
6
5
                  9
8
[10,]
         2
      2
3
4
[11,]
[12,]
```

Balanced Incomplete Block Designs:

```
> summary(aov(quality~wine+Error(tester/wine), data=BIBD))
Error: tester
      Df Sum Sq Mean Sq F value Pr(>F)
8 1133 141.58 4.569 0.119
wine
Residuals 3 93 30.99
Error: Within
        Df Sum Sq Mean Sq F value Pr(>F)
         8 1122.8 140.35 4.191 0.00714 **
Residuals 16 535.9
> summary(aov(quality~tester+Error(wine/tester), data=BIBD))
Error: wine
     Df Sum Sq Mean Sq
tester 8 908.1 113.5
Error: Within
       Df Sum Sq Mean Sq F value Pr(>F)
tester 11 1440.4 130.95 3.91 0.00692 **
Residuals 16 535.9 33.49
```

Balanced Incomplete Block Designs:

```
install.packages("ibd")
```

- library(ibd)
- aov.ibd(quality~wine+tester, data=BIBD)

Balanced Incomplete Block Designs:

Vorteile:

- viele treatments mit kleinen Blöcken erfassbar
- relativ einfache Analyse

Nachteile:

- * aufwändige Berechnung, wenn Blöcke fehlen
- nicht für alle Kombinationen v,b,r,k möglich
- problematisch, wenn Fehlervarianzen uneinheitlich
- wenn Mittelwerte der treatments sehr verschieden sind, kommt es zu Verzerrungen