Lecture 15
Topic 11: Unbalanced Designs (missing data)

In the real world, things fall apart:

plants are destroyed/trampled/eaten

animals get sick

volunteers quit

assistants are sloppy

accidents happen

The assumptions:

Data loss is due to *accidents*, not to treatments.

Missing values (e.g. Y_{ij}) follow the same mathematical model as all other observations in the experiment (e.g. $Y_{ij} = \mu + \tau_i + \varepsilon_{ij}$).

Missing data in single-factor designs = not a big deal

$$w = q_{\alpha,p,df_{MSE}} \sqrt{\frac{MSE}{r}} \quad \text{for equal r}$$

Tukey's Studentized Range (HSD) Test for Nlevel

Minimum Significant Difference 5.0499

Tukey Grouping		Mean	N	Culture
	A	28.800	5	3DOk1
B	A	23.940	5	3DOk5
B	C	19.880	5	3DOk7
D	C	18.700	5	Comp
D	E	14.600	5	3DOk4
	E	13.260	5	3DOk13

$$w = q_{\alpha,p,df_{MSE}} \sqrt{\frac{MSE}{2}\left(\frac{1}{r_1}+\frac{1}{r_2}\right)} \quad \text{for unequal r}$$

Tukey's Studentized Range (HSD) Test for Nlevel

Comparisons significant at the 0.05 level are indicated by *.**

Culture Comparison	Difference Between Means	Simultaneous 95% Confidence Limits		
3DOk1 - 3DOk5	4.450	-2.052	10.952	
3DOk1 - Comp	9.325	3.305	15.345	***
3DOk1 - 3DOk7	10.450	3.077	17.823	***
3DOk1 - 3DOk4	13.250	7.539	18.961	***
3DOk1 - 3DOk13	14.850	8.830	20.870	***
3DOk5 - 3DOk1	-4.450	-10.952	2.052	
3DOk5 - Comp	4.875	-1.627	11.377	
3DOk5 - 3DOk7	6.000	-1.772	13.772	
3DOk5 - 3DOk4	8.800	2.583	15.017	***
3DOk5 - 3DOk13	10.400	3.898	16.902	***
Comp - 3DOk1	-9.325	-15.345	-3.305	***
Comp - 3DOk5	-4.875	-11.377	1.627	
Comp - 3DOk7	1.125	-6.248	8.498	
Comp - 3DOk4	3.925	-1.786	9.636	
Comp - 3DOk13	5.525	-0.495	11.545	
3DOk7 - 3DOk1	-10.450	-17.823	-3.077	***
3DOk7 - 3DOk5	-6.000	-13.772	1.772	
3DOk7 - Comp	-1.125	-8.498	6.248	
3DOk7 - 3DOk4	2.800	-4.323	9.923	
3DOk7 - 3DOk13	4.400	-2.973	11.773	
3DOk4 - 3DOk1	-13.250	-18.961	-7.539	***
3DOk4 - 3DOk5	-8.800	-15.017	-2.583	***
3DOk4 - Comp	-3.925	-9.636	1.786	
3DOk4 - 3DOk7	-2.800	-9.923	4.323	
3DOk4 - 3DOk13	1.600	-4.111	7.311	
3DOk13 - 3DOk1	-14.850	-20.870	-8.830	***
3DOk13 - 3DOk5	-10.400	-16.902	-3.898	***
3DOk13 - Comp	-5.525	-11.545	0.495	
3DOk13 - 3DOk7	-4.400	-11.773	2.973	
3DOk13 - 3DOk4	-1.600	-7.311	4.111	

Missing data in crossed designs = a big deal

Loss of symmetry

Loss of orthogonal partitioning of sums of squares

Loss of simplicity of analysis

The two-factor case…the historical approach

Example: Comparing the yields of four breeding lines of wheat (RCBD).

Line	Block 1	2	3	4	5	Means	Totals
A	32.3	34.0	34.3	35.0	36.5	34.42	172.1
B	33.3	33.0	36.3	36.8	34.5	34.78	173.9
C	30.8	34.3	35.3	32.3	35.8	33.70	168.5
D		26.0	29.8	28.0	28.8	28.15	112.6
Means	32.13	31.83	33.93	33.03	33.90	32.76	
Totals	96.4	127.3	135.7	132.1	135.6		627.1

The basic strategy:
Replace the missing value with its best estimate and analyze the data

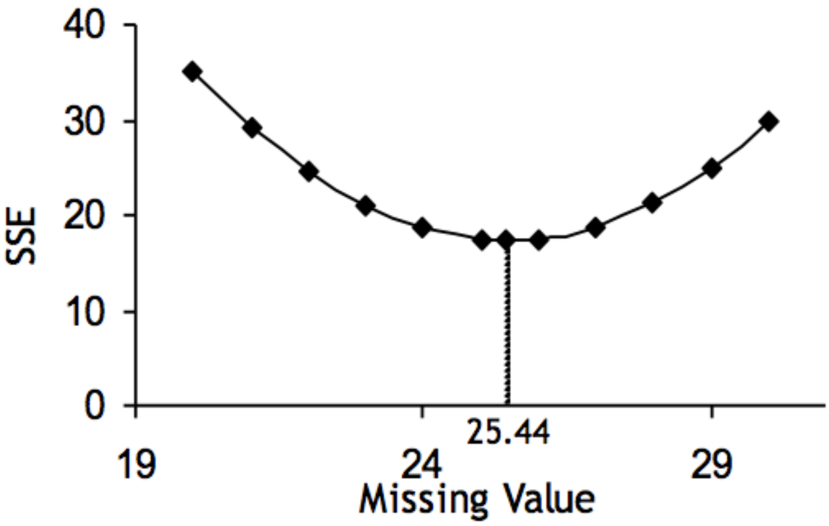
If the missing value is in row i and column j, and "t" is the number of treatments and "b" is the number of blocks, the best estimate is given by the following formula:

$$\boxed{\text{Estimated } Y_{ij} = (tY_{i.} + bY_{.j} - Y_{..}) / [(t-1)(b-1)]}$$

For this particular dataset, the value to be inserted is:

$$\text{Estimated } Y_{41} = [4 * 112.6 + 5 * 96.4 - 627.1] / (3 * 4) = \mathbf{25.44}$$

This is called the "least-squares" estimate of the missing value because it minimizes the SSE.



25.44 is the least-squares estimate of the missing value.

Once this value has been determined, it is entered into the data table and the ANOVA is computed as usual:

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	206.1710600	29.4530086	20.39	<.0001
Error	12	17.3299600	1.4441633		
Corrected Total	19	223.5010200			
Block	4	35.2113200	8.8028300	6.10	0.0065
Treatment	3	170.9597400	56.9865800	39.46	<.0001

At this point, two additional corrections are required:

1. **df_{Total} and df_{Error} must be corrected**

2. **SST and SSB must be corrected**

$$\text{Correction for SST} = [Y_{.j} - (t\text{-}1)^*\text{estimated } Y_{ij}]^2 / t^*(t\text{-}1)$$
$$\text{Correction for SSB} = [Y_{i.} - (b\text{-}1)^*\text{estimated } Y_{ij}]^2 / b^*(b\text{-}1)$$

In this case:

Correction for SST = $[96.4 - 3^*25.44]^2 / 4^*3 = 33.601$
So, Corrected SST = 170.95974 – 33.601 = 137.36

Correction for SSB = $[112.6 - 4^*25.44]^2 / 5^*4 = 5.875$
So, Corrected SSB = 35.21132 – 5.875 = 29.34

And the corrected ANOVA:

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	206.1710600	29.4530086	20.39	<.0001
Error	**11**	17.3299600	**1.5754509**		
Corrected Total	**18**	223.5010200			
Block	4	**29.34**	**7.335**	**4.66**	**0.0191**
Treatment	3	**137.36**	**45.79**	**29.06**	**<.0001**

The two-factor case…the modern approach

Missing data are indicated in R by "NA"

The data, in a form R can interpret (shown here in a table -- saved as the .csv file):

trtmt	block	yield
1	1	32.3
1	2	34.0
1	3	34.3
1	4	35.0
1	5	36.5
2	1	33.3
2	2	33.0
2	3	36.3
2	4	36.8
2	5	34.5
3	1	30.8
3	2	34.3
3	3	35.3
3	4	32.3
3	5	35.8
4	1	NA
4	2	26.0
4	3	29.8
4	4	28.0
4	5	28.8

Some results to consider:

```
miss1_mod<-lm(yield ~ trtmt + block, miss_dat)
anova(miss1_mod)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
trtmt	3	122.46	40.82	25.911	2.75e-05	***
block	4	29.34	7.33	4.655	0.0192	*
Residuals	11	17.33	1.58			

```
miss2_mod<-lm(yield ~ block + trtmt, miss_dat)
anova(miss2_mod)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
block	4	14.44	3.61	2.291	0.125	
trtmt	3	137.36	45.79	29.062	1.59e-05	***
Residuals	11	17.33	1.58			

Recall, from the historical, "least squares" approach:

Block	4	29.34	7.335	4.66	0.0191
Treatment	3	137.36	45.79	29.06	<.0001

For the factor listed last in each model, we obtain *exactly the same result* as when we replaced the missing value with its least-squares estimate!

Type I SS vs. Type II SS

Type I = *sequential* or *incremental* SS

Any variance that is common to two or more variables will be attributed to *one* variable (either the first variable listed in the model or the variable of lowest order [e.g. main effects before interactions]).

Type II = *partial* SS

The effect of each variable is evaluated *after* all other factors have been accounted for.

In R, partial (or Type II) SS can be obtained with the Anova() function in the "car" package:

```
#library(car)
miss1_mod<-lm(yield ~ trtmt + block, miss_dat)
Anova(miss1_mod, type=2)
```

	Sum Sq	Df	F value	Pr(>F)	
trtmt	137.4	3	29.0624	1.586e-05	***
block	29.3	4	4.6552	0.01919	*
Residuals	17.3	11			

```
miss2_mod<-lm(yield ~ block + trtmt, miss_dat)
Anova(miss2_mod, type=2)
```

	Sum Sq	Df	F value	Pr(>F)	
block	29.3	4	4.6552	0.01919	*
trtmt	137.4	3	29.0624	1.586e-05	***
Residuals	17.3	11			

Recall, from the historical, "least squares" approach:

Block	4	29.34	7.335	4.66	0.0191
Treatment	3	137.36	45.79	29.06	<.0001

Effects of unbalanced data on the estimation of differences between means

Missing data → Loss of balance → Loss of orthogonality

(i.e. SS become contaminated by other parameters in the model)

Data		**B**		
		1	**2**	
A	**1**	7, 9	5	7
	2	8	4, 6	6
		8	5	

$7 - 6 = 1 \neq 0$

There is an effect of factor A.

Means		**B**		
		1	**2**	
A	**1**	8	5	6.5
	2	8	5	6.5
		8	5	

$6.5 - 6.5 = 0$

There isn't an effect of factor A.

The design is unbalanced

Orthogonality is broken

Factor B influences the calculation of the effect of factor A

Imagine the underlying linear model:

$$y_{ij} = \mu + \alpha_i + \beta_j$$

Data	B 1	2
A 1	$7 = \mu + \alpha_1 + \beta_1$ $9 = \mu + \alpha_1 + \beta_1$	$5 = \mu + \alpha_1 + \beta_2$
A 2	$8 = \mu + \alpha_2 + \beta_1$	$4 = \mu + \alpha_2 + \beta_2$ $6 = \mu + \alpha_2 + \beta_2$

Mean A_1 – Mean A_2 $= 1/3\ (7 + 9 + 5) - 1/3\ (8 + 4 + 6)$

$= 1/3\ [(\alpha_1 + \beta_1) + (\alpha_1 + \beta_1) + (\alpha_1 + \beta_2)]$
$\quad - 1/3[(\alpha_2 + \beta_1) + (\alpha_2 + \beta_2) + (\alpha_2 + \beta_2)]$

$= (\alpha_1 - \alpha_2) + \mathbf{1/3\ (\beta_1 - \beta_2)}$

The difference between the marginal means for the two levels of A is a measure of the effect of factor A *PLUS* an effect of factor B.

The null hypothesis about A we typically wish to test:

H_0: $\alpha_1 - \alpha_2 = 0$.

The null hypothesis actually tested using the Type I SS for factor A:

H_0: $\alpha_1 - \alpha_2 + 1/3\ (\beta_1 - \beta_2) = 0$

The problem with unbalanced designs in multifactor analyses:

The factors get mixed up with each other in the calculations

Effects of unbalanced data on the estimation of marginal means

We usually want to estimate the marginal means of A:

$$(\mu_{11} + \mu_{12}) / 2 \qquad \text{and} \qquad (\mu_{21} + \mu_{22}) / 2$$

In this particular dataset, the A marginal means actually estimate:

$$(2\mu_{11} + \mu_{22}) / 3 \qquad \text{and} \qquad (\mu_{21} + 2\mu_{22}) / 3$$

The expected marginal mean for A_1:

$$[(\mu + \alpha_1 + \beta_1) + (\mu + \alpha_1 + \beta_1) + (\mu + \alpha_1 + \beta_2)] / 3$$

$$[3\mu + 3\,\alpha_1 + 2\beta_1 + \beta_2] / 3 = \mu + \alpha_1 \mathbf{+ 2/3\beta_1 + 1/3\beta_2}$$

Consider the following table of calculated means, under two situations:

	Missing value as "25.44166"		Missing value as "NA"
Treatment A	34.4200		34.4200
Treatment B	34.7800		34.7800
Treatment C	33.7000		33.7000
Treatment D	27.6083	≠	28.1500
Block 1	30.4604	≠	32.1333
Block 2	31.8250		31.8250
Block 3	33.9250		33.9250
Block 4	33.0250		33.0250
Block 5	33.9000		33.9000

The means on the left are referred to as "least-squares means" (LSMeans). They are least-squares estimates of the means, adjusted for contamination due to loss of balance/orthogonality.

The means on the right are simple, unadjusted means, cross-contaminated by other factors due to the loss of balance/orthogonality.

Means of unbalanced data are functions of cell frequencies;
LSMeans are not

Computing LSMeans in R

To compute LSMeans in R, use the lsmeans() function found in the "lsmeans" library:

```
#library(lsmeans)
lsmeans(miss1_mod, "trtmt")
lsmeans(miss1_mod, "block")

trtmt     lsmean        SE df lower.CL upper.CL
 1      34.42000 0.5613289 11 33.18452 35.65548
 2      34.78000 0.5613289 11 33.54452 36.01548
 3      33.70000 0.5613289 11 32.46452 34.93548
 4      27.60833 0.6481668 11 26.18173 29.03494

block     lsmean        SE df lower.CL upper.CL
 1      30.46042 0.7469753 11 28.81634  32.1045
 2      31.82500 0.6275848 11 30.44370  33.2063
 3      33.92500 0.6275848 11 32.54370  35.3063
 4      33.02500 0.6275848 11 31.64370  34.4063
 5      33.90000 0.6275848 11 32.51870  35.2813
```

Easy. Now if we compare everything in one table:

	Missing value as "25.44166"		**Missing value as "NA"**	
	Means	**LS Means**	Means	**LS Means**
Treatment A	34.4200	34.4200	34.4200	34.4200
Treatment B	34.7800	34.7800	34.7800	34.7800
Treatment C	33.7000	33.7000	33.7000	33.7000
Treatment D	27.6083	27.6083	28.1500	≠ 27.6083
Block 1	30.4604	30.4604	32.1333	≠ 30.4604
Block 2	31.8250	31.8250	31.8250	31.8250
Block 3	33.9250	33.9250	33.9250	33.9250
Block 4	33.0250	33.0250	33.0250	33.0250
Block 5	33.9000	33.9000	33.9000	33.9000

Left columns: The design is "balanced," so Means = LSMeans.

Right columns: The design is unbalanced, so Means ≠ LSMeans for affected classes.

Comparing LSMeans in R

Since ordinary means are contaminated by other factors in unbalanced crossed designs, all comparisons should be made among *adjusted* means. The "lsmeans" package makes all this fairly convenient:

For such means comparison, you must first assign the computed lsmeans to an object. Here, I assign it to an object called "miss.lsm":

```
miss_lsm <- lsmeans(miss1_mod, "trtmt")
```

This object can then be acted upon by the `contrast`() function within the "lsmeans" package.

To perform Tukey (HSD) pairwise comparisons among the adjusted means:

```
contrast(miss_lsm, method = "pairwise", adjust = "tukey")
```

contrast	estimate	SE	df	t.ratio	p.value
1 - 2	-0.360000	0.7938390	11	-0.453	0.9676
1 - 3	0.720000	0.7938390	11	0.907	0.8016
1 - 4	6.811667	0.8574441	11	7.944	<.0001
2 - 3	1.080000	0.7938390	11	1.360	0.5469
2 - 4	7.171667	0.8574441	11	8.364	<.0001
3 - 4	6.091667	0.8574441	11	7.104	0.0001

To perform a Dunnett test (comparisons to a control) among the adjusted means:

```
contrast(miss_lsm, method = "trt.vs.ctrl")
```

contrast	estimate	SE	df	t.ratio	p.value
2 - 1	0.360000	0.7938390	11	0.453	0.9604
3 - 1	-0.720000	0.7938390	11	-0.907	0.7661
4 - 1	-6.811667	0.8574441	11	-7.944	<.0001

To compare the adjusted means using orthogonal contrasts (group comparisons):

```
contrast(miss_lsm, list("A vs. B" = c(1,-1,0,0), "AB vs. CD" = c(1,1,-
      1,-1), "C vs. D" = c(0,0,1,-1)))
```

contrast	estimate	SE	df	t.ratio	p.value
A.vs..B	-0.360000	0.7938390	11	-0.453	0.6590
AB.vs..CD	7.891667	1.1684993	11	6.754	<.0001
C.vs..D	6.091667	0.8574441	11	7.104	<.0001

In all cases, notice that the estimates are the differences in the LSMeans.

To conduct a trend analysis among the levels of the factor of interest:

```
contrast(miss.lsm, method = "poly")
```

contrast	estimate	SE	df	t.ratio	p.value
linear	-21.515000	2.692039	11	-7.992	<.0001
quadratic	-6.451667	1.168499	11	-5.521	0.0002
cubic	-3.571667	2.531172	11	-1.411	0.1859

Package "multcomp"

One can obtain the same results using the "multcomp" (multiple comparisons) package in R. Within this package, the glht() function (general linear hypotheses) can be used to compare LSMeans in unbalanced datasets. Sample code is below:

```
#Comparing LSMeans, using the "multcomp" package (function glht())
#library(multcomp)
#library(sandwich)
summary(glht(miss1_mod,linfct=mcp(trtmt="Tukey")))
summary(glht(miss1_mod,linfct=mcp(trtmt="Dunnett")))
```

In the above two lines, the glht() function is being used to carry out pairwise comparisons of LSMeans. mcp refers to "multiple comparison procedures," of which Tukey and Dunnett are options. linfct is a specification of the linear hypothesis to be tested, in this case pairwise comparisons.

You may also perform group comparisons and trend analyses using orthogonal contrasts. In the example below, I am building a contrast matrix K, containing 3 separate orthogonal contrasts. This contrast matrix is then called by the mcp option within glht:

```
K<-rbind("A vs. B"=c(1,-1,0,0), "AB vs. CD" = c(1,1,-1,-1), "C vs.
      D"=c(0,0,1,-1))
summary(glht(miss1_mod,linfct=mcp(trtmt=K)))
```

The Problem

Loss of balance leads to the contamination of SS, means, and differences among means by effects of other factors in the model.

The Solutions

Adjust factor SS to remove contaminating effects by using **partial sums of squares [**Anova()**]**.

Adjust means to remove the contaminating effects by using **least-squares (adjusted) means [**lsmeans()**]**.