Stat 212 Notes

Table of contents

Course Goals for STAT 212

STAT 212 is an introduction to the statistical aspects of study design. We'll consider all stages of the experimental process. By the end of the course, you will:

- understand the differences between designed experiments and observational studies, including the resulting conclusions that can be drawn.
- be familiar with major experimental and treatment design structures, and what to consider when choosing an appropriate design for a study.
- understand the impact of sample size, variation, and effect size on power.
- be able to write down a linear additive model, correctly specify fixed effects, random effects, interactions, and model assumptions for the error terms.
- sketch a skeleton ANOVA table from a description of a study.
- understand data collection considerations
- be able to analyze data resulting from major experimental and treatment design structures, including estimating main and simple effects, pairwise comparisons, contrasts, and model parameters.
- be able to clearly write up the results of an analysis so that a non-statistician could understand the major findings of the study.

1 Introduction to Design

> In interpreting and in presenting experimental results there is no adequate substitute for thought– thought about the questions to be asked, thought about the nature and weight of evidence the data provide on those questions, and thought about how the story can be told with clarity and full honesty to a reader. Statistical techniques must be chosen and used to aid, but not to replace, relevant thought.
> - Bryan-Jones and Finney, 1983

There are two aspects to any design of experiments problem: the actual design and the analysis of the data from the experiment. We'll be discussing both aspects at length is semester in about equal measure.

Experimentation is one of the most common activities that people engage in, because it allows an investigator to find out what happens to a response when settings of another variable are purposefully changed. The results of the experiment provide a basis for selecting optimum settings or determine a plan of action. Experimentation is carried out by everyone, in everyday activities. For example, observing what happens to the taste of brownies if you change the material of the baking dish or the oven temperature. Changing the time you visit the dining hall to determine if it is less crowded. Changing a pet's food to see if they eat better/more. All of these experiments.

Can you think of others?

The goal of STAT 212 is to fully explore the "relevant thought" in the Bryan-Jones and Finney quote above, and to help you become independent in this thought. By the end of the semester you should be able to:

1. Optimally design simple studies to answer a research question.

2. Compute and interpret the analysis of data resulting from these designs.

3. Communicate the results of your analysis to non-statisticians both verbally and in writing.

4. Converse intelligently about more complex designs with someone who has more training.

Well-designed experiments allow an investigator to conduct better studies more efficiently, analyze data effectively, and make the connections between the conclusions from the analysis and the original research objectives. In every experiment, there are seven basic steps:

1.

2.

3.

4.

5.

6.

7.

Throughout these steps, we must always keeps the following in mind:

1.

2.

3.

4.

To look at these in more detail, let's consider an example (and state a few definitions along the way).

Example: Two students at Queensland University of Technology, as a project for their statistics class, carried out an experiment to test the effect certain factors such as refrigeration, stem length, and water content have on the life of a cut rose. The students considered

- Stem length (15 cm or 25 cm)
- Water content (tap water or tap water + citric acid)
- Temperature (refrigerated or room temperature)

The response measured was the number of days until death, and the goal is to determine the conditions that will extend rose life.

In this example, there are 3 **factors**.

Definition:

In order to study the effect of the factor on the response, two or more values of the factor are considered. These values are called **levels**.

A **treatment** is a combination of factor levels. In this experiment:

- Factors:

- Levels:

- Treatments:

Definition: The **treatment design**

Definition: The **experimental design**

There are three main principles of **experimental design** that must be addressed with every design. They are:

1. **Replication:**

Why it's important:

Each rose in the experiment above, each replicate, is an **experimental unit. An experimental unit is the smallest physical entity to which a treatment can be INDEPENDENTLY applied.**

The **sampling unit** is the entity that is actually observed or measured.

Pseudo-replication occurs when subsets of the experimental units (often sampling units) are mistakenly identified as experimental units. This is the **most common serious design error.**

Examples:

2. **Randomization:**

Why it's important:

3. **Control/Blocking:**

Why it's important:

To make sure everyone's on the same page, let's review the two-sample case. Back to the example.

Example: Let's consider just the data on refrigeration vs room temperature. There were 16 roses: 8 stored at room temperature and 8 stored in a refrigerator. Let

$$\mu_1 =$$

$$\mu_2 =$$

We have observations

$$y_{ij} =$$

We can express each observation as a combination of the treatment mean and the characteristics unique to each individual rose.

In STAT 102/318 we assume:

We can further decompose μ_i:

The Central Limit Theorem says that the sample means $\overline{y}_{i\cdot}$ will be (approximately) normally distributed with mean μ_i and variance σ^2/n. We also have to estimate σ^2, the variance of e_{ij} and hence the variance of y_{ij}.

To determine if refrigeration and room temperature produce different lifetimes of cut roses we can test the null hypothesis $H_0 : \mu_1 = \mu_2$ or we can estimate $\mu_1 - \mu_2$ with confidence using t-procedures.

We could also use the Analysis of Variance. The ANOVA table would look like:

Source	df	Sum of Sq	Mean Sq	F	p-value
Model		SSTrt	MSTrt	MSTrt/MSE	
Error		SSError	MSE		
Total		SSTotal			

There are two mistakes we could make carrying out this test.

- **Type I Error:**

- **Type II Error:**

A new concept we'll talk about this semester is closely related to Type II Error: **power**.

Definition: Power

All of these concepts of treatment design, experimental design, power, factors, and levels will be revisited many times over the course of the semester with increasingly complex treatment and experimental designs. We'll start with the simplest experimental design: the **Completely Randomized Design** (CRD).

2 The Completely Randomized Design

The first **experimental design** we'll consider is the **completely randomized design** or CRD. The CRD is an experimental design because

The CRD is characterized by

The CRD may be combined with several different **treatment designs**. To explore the CRD in more detail, we'll start with the simplest treatment design, the **one-way design**. The one-way design is so named because

Within one-way designs there are four basic treatment structures:

1. Unstructured
2. Control versus other treatments
3. Quantitative
4. Other structure

Example: A donut manufacturer wants to see if the type of fat used to fry the donuts has any impact on the amount of fat absorbed by the donuts. The manufacturer has two types of animal fat and two types of vegetable fat that they would like to compare. They also have available 4 fryers, which can each fry 1 batch of 18 donuts at a time. They plan to measure the amount of fat absorbed in each batch.They have the resources to test 24 total batches of donuts.

- **Treatment Design:**
  - Factor:
  - Levels:
- **Experimental Design:**
  - Run 6 batches of each fat in a Completely Randomized Design

There are two possible scenarios for the CRD. Discuss the pros and cons of each, and decide which one you would use and why.

- Scenario 1: Randomly assign fat to a fryer and prepare 6 batches.

- Scenario 2: Randomly select 6 of the 24 batches for each type of fat.

Here's one possible experimental layout

			Batch			
Fryer	1	2	3	4	5	6
1	1	2	2	4	2	4
2	1	1	4	1	3	1
3	4	1	3	2	4	2
4	3	3	3	3	4	2

2.1 CRD Model and Basic Analysis

The **CRD Model** can be written in two different ways.

- y_{ij} = amount of fat absorbed by the j^{th} batch of donuts prepared using the i^{th} fat
- μ = overall mean fat absorbed by a batch of donuts
- τ_i = treatment effect of fat i = additional amount of fat absorbed on average by batches prepared using fat type i
- ϵ_{ij} = random error = additional amount of fat absorbed by the j^{th} batch of donuts prepared using fat i

Some consequences of these assumptions:

Expected value:

Dispersion:

Distribution:

For this particular treatment design, there are several hypothesis tests that may be of interest. Write out in the symbols the null and alternative hypotheses for the following specified objectives. Reminder: Fats 1 and 2 are animal fats and Fats 3 and 4 are vegetable fats.

1. Are there differences among the four fats with respect to the amount of fat absorbed?

2. Do the vegetable fats differ from the animal fats in the amount of fat absorbed?

3. Are there differences between the two animal fats? Are there differences between the two vegetable fats?

Let's try out using SAS. We'll start by reading in the data.

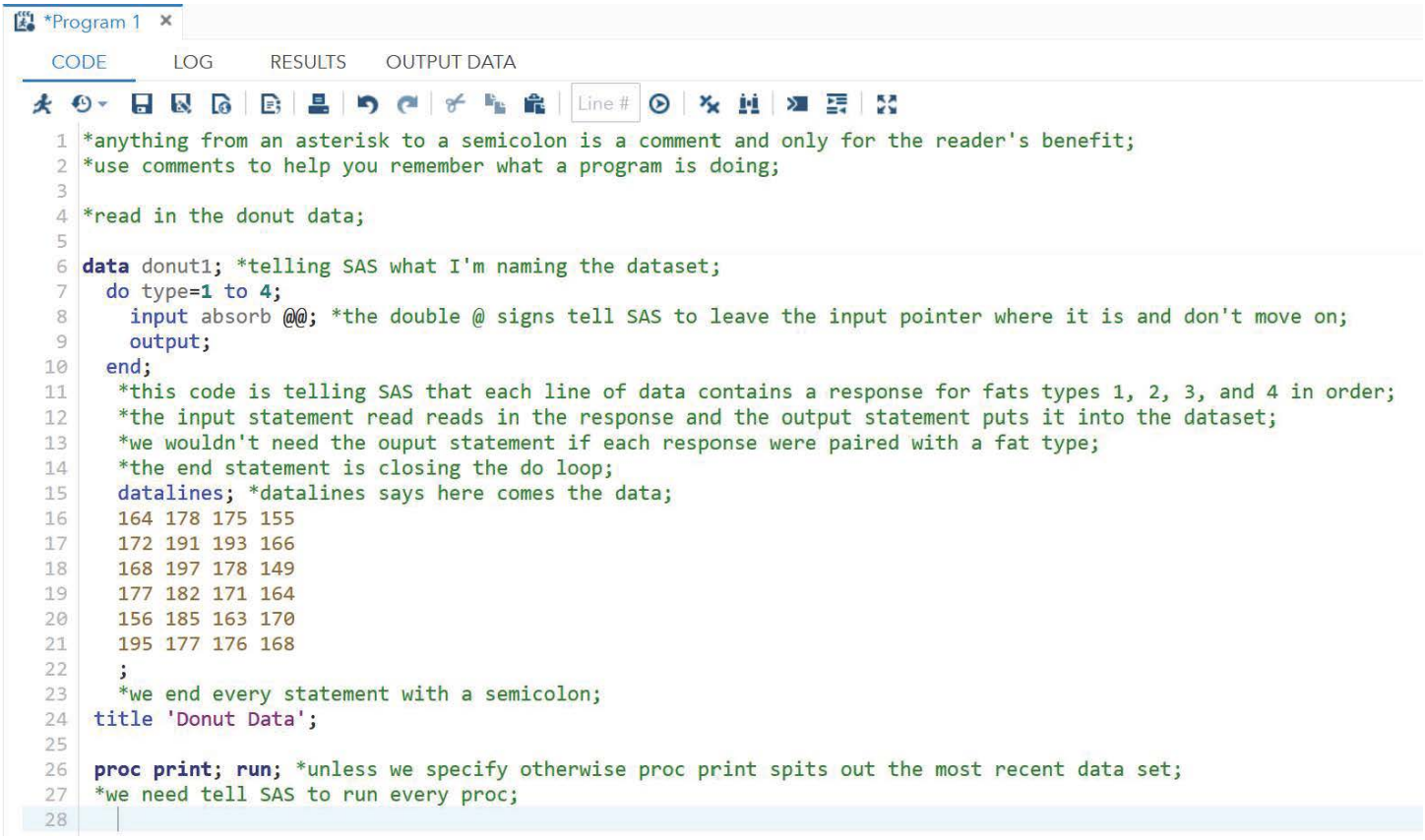
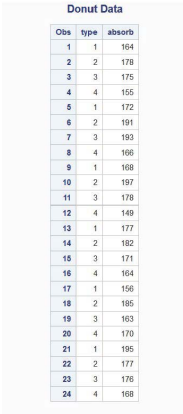


Figure 2.1: SAS code for reading in the donut data set

Here are the results of `proc print`:



Donut Data

Obs	type	absorb
1	1	164
2	2	178
3	3	175
4	4	155
5	1	172
6	2	191
7	3	193
8	4	166
9	1	168
10	2	197
11	3	178
12	4	149
13	1	177
14	2	182
15	3	171
16	4	164
17	1	156
18	2	185
19	3	163
20	4	170
21	1	195
22	2	177
23	3	176
24	4	168

Figure 2.2: Donut data set

We can read the data in other ways as well. We'll see other data set options as we encounter different types of variables.

Let's start by plotting the data.

```
proc plot data=donut1; *specifying the data set I want to plot;
  plot absorb*type; *y-axis*x-axis;
run;
```

That's not very pretty. Graphics are never going to be as pretty as they are in R, but we can do better.

```
proc gplot data=donut1;
  plot absorb*type='dot'; *'dot' is the symbol I want to use;
run;
```

Now let's construct the ANOVA table:

Source of Variation	df	SS	MS	Expected MS	F
Fat Type	$t-1=3$	SSTrt	MST	$\sigma^2+\frac{n}{t-1}\sum_{i=1}^{4}\tau_i^2$	MST/MSE
Error	$t(n-1)=20$	SSError	MSE	σ^2	
Total	$nt-1=23$	SSTotal			

Under the null hypothesis, $\mathrm{H}_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4$, or equivalently $\mathrm{H}_0 : \tau_i = 0$ for all i.

So, if the null hypothesis is true

and F follows the F distribution with $\mathrm{num}_{df} = t-1$ and $\mathrm{den}_{df} = t(n-1)$.

Let's fit this model in SAS. There are a couple of different procedures ('proc's) we can use to do so. We'll start with `proc glimmix`. This gives a LOT of output.

```
proc glimmix data=donut1;
  class type; *telling SAS that fat type is the treatment factor;
  model absorb=type; *model response = treatment factor;
  lsmeans type/pdiff cl; *let's add some fanciness;
run;
```

```
           The GLIMMIX Procedure
            Model Information

Data Set                      WORK.DONUT1
Response Variable             absorb
Response Distribution         Gaussian
Link Function                 Identity
Variance Function             Default
Variance Matrix               Diagonal
Estimation Technique          Restricted Maximum Likelihood
Degrees of Freedom Method     Residual


     Class Level Information

    Class    Levels    Values
    type          4    1 2 3 4

Number of Observations Read          24
Number of Observations Used          24

            Dimensions

Covariance Parameters          1
Columns in X                   5
Columns in Z                   0
Subjects (Blocks in V)         1
Max Obs per Subject           24


      Optimization Information

Optimization Technique    None
Parameters                5
Lower Boundaries          1
Upper Boundaries          0
Fixed Effects             Not Profiled
```

Fit Statistics

-2 Res Log Likelihood	156.21
AIC (smaller is better)	166.21
AICC (smaller is better)	170.49
BIC (smaller is better)	171.19
CAIC (smaller is better)	176.19
HQIC (smaller is better)	167.18
Pearson Chi-Square	2018.00
Pearson Chi-Square / DF	100.90

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
type	3	20	5.41	0.0069

type Least Squares Means

type	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
1	172.00	4.1008	20	41.94	<.0001	0.05	163.45	180.55
2	185.00	4.1008	20	45.11	<.0001	0.05	176.45	193.55
3	176.00	4.1008	20	42.92	<.0001	0.05	167.45	184.55
4	162.00	4.1008	20	39.50	<.0001	0.05	153.45	170.55

Differences of type Least Squares Means

type	_type	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
1	2	-13.0000	5.7994	20	-2.24	0.0365	0.05	-25.0974	-0.9026
1	3	-4.0000	5.7994	20	-0.69	0.4983	0.05	-16.0974	8.0974
1	4	10.0000	5.7994	20	1.72	0.1001	0.05	-2.0974	22.0974
2	3	9.0000	5.7994	20	1.55	0.1364	0.05	-3.0974	21.0974
2	4	23.0000	5.7994	20	3.97	0.0008	0.05	10.9026	35.0974
3	4	14.0000	5.7994	20	2.41	0.0255	0.05	1.9026	26.0974

There's a lot of stuff here, but what's missing?

Let's try `proc mixed`.

```
proc mixed data=donut1;
  class type;
  model absorb=type;
run;
```

One more try:

```
proc mixed data=donut1 method=type3;
  class type;
  model absorb=type;
run;
```

Finally!

Type 3 Analysis of Variance

Source	DF	Sum of Squares	Mean Square	Expected Mean Square	Error Term	Error DF
type	3	1636.500000	545.500000	Var(Residual) + Q(type)	MS(Residual)	20
Residual	20	2018.000000	100.900000	Var(Residual)	.	.

Type 3 Analysis of Variance

Source	F Value	Pr > F
type	5.41	0.0069
Residual	.	.

Let's go back to the `glimmix` output and look more carefully at some components. The output below comes from the `lsmeans type/cl;` statement.

type Least Squares Means

type	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
1	172.00	4.1008	20	41.94	<.0001	0.05	163.45	180.55
2	185.00	4.1008	20	45.11	<.0001	0.05	176.45	193.55
3	176.00	4.1008	20	42.92	<.0001	0.05	167.45	184.55
4	162.00	4.1008	20	39.50	<.0001	0.05	153.45	170.55

Now let's look at what's provided by the `lsmeans type/pdiff;` statement.

Differences of type Least Squares Means

type	_type	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
1	2	-13.0000	5.7994	20	-2.24	0.0365	0.05	-25.0974	-0.9026
1	3	-4.0000	5.7994	20	-0.69	0.4983	0.05	-16.0974	8.0974
1	4	10.0000	5.7994	20	1.72	0.1001	0.05	-2.0974	22.0974
2	3	9.0000	5.7994	20	1.55	0.1364	0.05	-3.0974	21.0974
2	4	23.0000	5.7994	20	3.97	0.0008	0.05	10.9026	35.0974
3	4	14.0000	5.7994	20	2.41	0.0255	0.05	1.9026	26.0974

2.2 Treatment Comparisons and Contrasts

We can see in the results above that we may reject the overall hypothesis that the four treatments produce the same mean fat absorption ($F = 5.41$, p-value$= 0.0069$). But, this doesn't address the hypotheses you constructed earlier. Remember, we also considered:

- Do the vegetable fats differ from the animal fats in the amount of fat absorbed?
- Are there differences between the two animal fats?
- Are there differences between the two vegetable fats?

The output above allows us to address some of these questions, but not the one regarding vegetable fats versus animal fats. Let's look at a more general way to construct treatment comparisons.

Contrasts

A well-thought-out treatment design's objectives can usually be stated in terms of a set of **contrasts**. This is usually an important goal in planning the design, and contrasts are constructed before data are collected.

A **contrast** is

Estimates of the contrast are obtained by substituting in the sample means

We may also obtain standard errors of the contrast estimate

Standard errors may then be used to carry out tests and construct confidence intervals.

The contrasts of interest depend on the basic treatment design structure and the goals of the experiment. Remember, the four basic structures are

1. Unstructured
2. Control versus other treatments
3. Quantitative
4. Other structure

Let's first consider Unstructured designs, because these are the simplest.

2.2.1 Unstructured Treatment Designs and All Pairwise Comparisons

Example: The *New England Journal of Medicine* published a study investigating the effects of different exercise programs on postural stability in Parkinson's patients. The three exercise programs compared were: tai chi, resistance training, and stretching. 65 patients with Parkinson's were randomly assigned to each program, and change in functional reach was measured after 24 weeks.

- **Treatment Design:**
  - Factor:
  - Levels:
- **Experimental Design:**
  -

In designs like this without structure, we are typically interested in **all pairwise comparisons**.

There are multiple methods for making such comparisons. The simplest is the **least significant difference** (LSD), also called the unprotected LSD. It's easy, but the Type I error rate can be badly inflated (we'll talk more about this in a bit).

A (slightly) more conservative option is **Fisher's protected LSD**.

We've already seen these, but let's add some fanciness!

```
proc glimmix data=reach;
  class group;
  model reach=group;
  lsmeans group/pdiff cl lines plot=diffplot;
run;
```

This gives (in part) the output:

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Group	2	192	11.10	<.0001

Group Least Squares Means

Group	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
Resistan	2.3400	0.6130	192	3.82	0.0002	0.05	1.1310	3.5490
Stretchi	0.8569	0.6130	192	1.40	0.1637	0.05	-0.3521	2.0660
Tai_Chi	4.8938	0.6130	192	7.98	<.0001	0.05	3.6848	6.1029

Differences of Group Least Squares Means

Group	_Group	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
Resistan	Stretchi	1.4831	0.8669	192	1.71	0.0887	0.05	-0.2268	3.1929
Resistan	Tai_Chi	-2.5538	0.8669	192	-2.95	0.0036	0.05	-4.2637	-0.8440
Stretchi	Tai_Chi	-4.0369	0.8669	192	-4.66	<.0001	0.05	-5.7468	-2.3271

T Grouping for Group Least Squares Means (Alpha=0.05)

LS-means with the same letter are not significantly different.

Group	Estimate	
Tai_Chi	4.8938	A
Resistan	2.3400	B
		B
Stretchi	0.8569	B

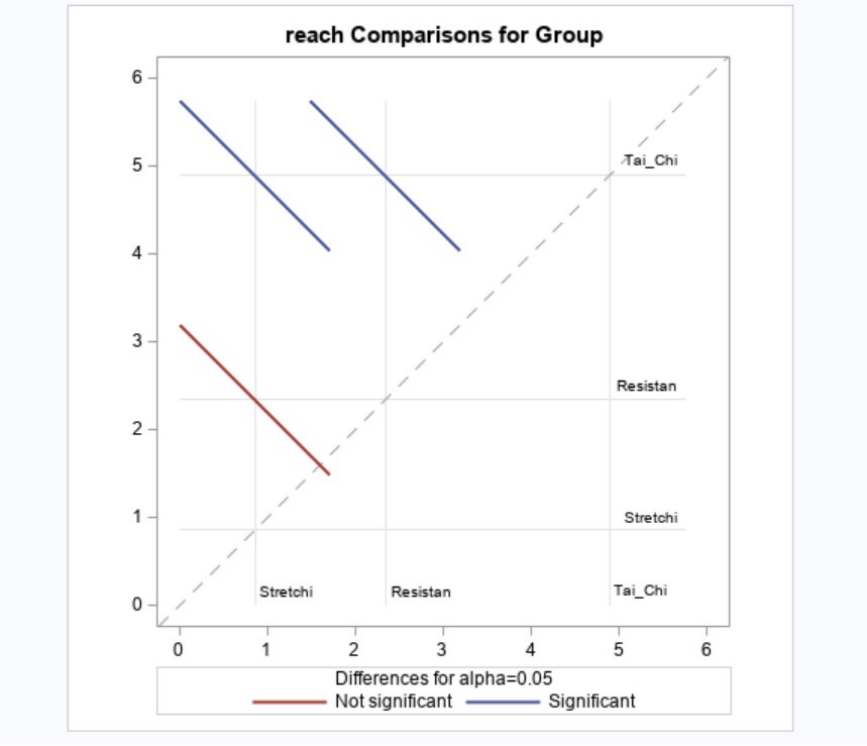


Figure 2.3: Diffogram for the reach data

This plot is called a **diffogram** and is a way to visualize differences among the treatments.

So these plots are awesome, and the output is easy to interpret! Why do we care about anything other than the LSD? The big issue is Type I error rate, and it can be a concern for pairwise comparisons as well as more complicated contrasts.

Multiple Comparisons

If more than one comparison is made among the treatment means, then we have multiple comparisons which can lead to the problem of **multiplicity**.

Definition: Multiplicity is

For a single test, the significance level of a Type I error is called a **comparison-wise** error rate. This means

But, if we have multiple tests, the Type I errors for these tests accumulate. This accumulated rate is the called the **experiment-wise** error rate. This is

But, the errors don't just add up. They accumulate in a power-type relationship. Consider a situation with a comparison-wise error rate of α and c independent comparisons. Then, the experiment-wise error rate is

For example, consider a situation with $\alpha = 0.05$ and 5 independent comparisons (there are as many independent comparisons as there are df for treatment). In that case:

We can control the experiment-wise error rate by setting it to a pre-specified value α (maybe 0.05) and then solving for the comparison-wise error rate, assuming c independent comparisons. So, for example, if $\alpha = 0.05$ and $c = 5$,

We'd then use this as the critical value (cut-off) value for our independent treatment comparisons.

But here's another issue. If the comparisons are not independent (which they aren't in all pairwise-comparisons, and often aren't in pre-planned contrasts of interest), then the experiment-wise error rate is actually even bigger than we see above. What can we do?

There are a multitude of multiple comparison procedures which control the overall experiment-wise error rate, which have different pros and cons. We're only going to the talk about a few.

Tukey's HSD: Tukey's Honestly Significant Difference (HSD) procedure is based on the studentized range statistic. To get this HSD from SAS:

```
proc glimmix data=reach;
  class group;
  model reach=group;
  lsmeans group/pdiff cl adjust=tukey plot=diffplot;
run;
```

Differences of Group Least Squares Means
Adjustment for Multiple Comparisons: Tukey

Group	_Group	Estimate	Standard Error	DF	t Value	Pr > |t|	Adj P	Alpha
Resistan	Stretchi	1.4831	0.8669	192	1.71	0.0887	0.2038	0.05
Resistan	Tai_Chi	-2.5538	0.8669	192	-2.95	0.0036	0.0101	0.05
Stretchi	Tai_Chi	-4.0369	0.8669	192	-4.66	<.0001	<.0001	0.05

Differences of Group Least Squares Means
Adjustment for Multiple Comparisons: Tukey

Group	_Group	Lower	Upper	Adj Lower	Adj Upper
Resistan	Stretchi	-0.2268	3.1929	-0.5645	3.5307
Resistan	Tai_Chi	-4.2637	-0.8440	-4.6015	-0.5062
Stretchi	Tai_Chi	-5.7468	-2.3271	-6.0845	-1.9893

The diffogram is also adjusted.

Pros/Cons of the HSD:

The other multiple comparison procedures we'll discuss are used with other treatment design structures. The three other one-way treatment design structures are:

1. Control versus other treatments
2. Quantitative (we'll put a pin in this one for now)
3. Other structure

2.2.2 Control versus other treatments

In some scenarios, one of the factor levels acts as a control treatment for some or all of the remaining levels. Often, we are interested in comparing all of the treatments against the control but not against each other. This means there are

Dunnett's procedures is a modification to the two-sample t test that is used when comparing all treatments against a control.

Example: Sections of tomato plant tissue were grown in culture with differing amounts and types of sugars with five replications of four treatments. The treatments were: control, 3% glucose, 3% fructose, and 3% sucrose.

- **Treatment Design:**
  - Factor:
  - Levels:
- **Experimental Design:**
  -

In a situation like this, we may be interested in comparing each of the sugar treatments to the control.

```
proc glimmix data=tomato;
  class trt;
  model growth=trt;
  lsmeans trt/diff=control('control') cl adjust=dunnett plot=controlplot;
run;
```

Note that unless otherwise specified, SAS will assume the first treatment level (alphabetically or numerically) is the control.

trt Least Squares Means

trt	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
control	42.2000	1.1726	16	35.99	<.0001	0.05	39.7142	44.6858
fructose	27.6000	1.1726	16	23.54	<.0001	0.05	25.1142	30.0858
glucose	29.0000	1.1726	16	24.73	<.0001	0.05	26.5142	31.4858
sucrose	34.0000	1.1726	16	29.00	<.0001	0.05	31.5142	36.4858

Differences of trt Least Squares Means
Adjustment for Multiple Comparisons: Dunnett

trt	_trt	Estimate	Standard Error	DF	t Value	Pr > |t|	Adj P	Alpha
fructose	control	-14.6000	1.6583	16	-8.80	<.0001	<.0001	0.05
glucose	control	-13.2000	1.6583	16	-7.96	<.0001	<.0001	0.05
sucrose	control	-8.2000	1.6583	16	-4.94	0.0001	0.0004	0.05

Differences of trt Least Squares Means
Adjustment for Multiple Comparisons: Dunnett

trt	_trt	Lower	Upper	Adj Lower	Adj Upper
fructose	control	-18.1155	-11.0845	-18.8990	-10.3010
glucose	control	-16.7155	-9.6845	-17.4990	-8.9010
sucrose	control	-11.7155	-4.6845	-12.4990	-3.9010

We get a new plot!

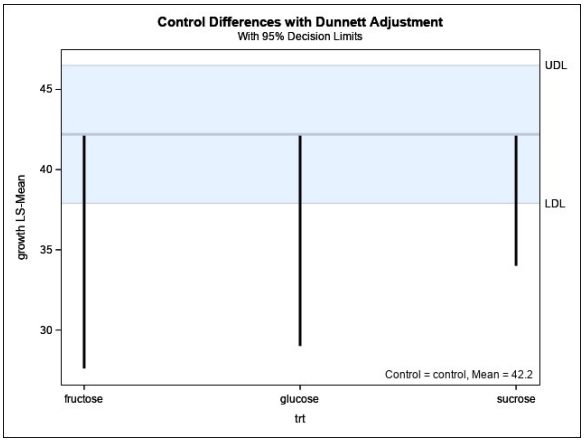


Figure 2.4: Control plot for the tomato plant data

Pros/Cons of Dunnett's Test:

2.2.3 Treatment Designs with (other) Structure

This is where the donut example fits in. There isn't a true control, but we also may not care about all pairwise comparisons. Instead, we had some specific, pre-planned comparisons of interest:

- Do the vegetable fats differ from the animal fats in the amount of fat absorbed?
- Are there differences between the two animal fats?
- Are there differences between the two vegetable fats?

Why pre-plan comparisons?

Earlier, we wrote out the hypotheses of interest corresponding to these comparisons:

There are three options available in SAS to test these hypotheses and/or construct confidence intervals:

- `contrast` statement
- `estimate` statement
- `lsmestimate` statement

All three statements involve specifying the coefficients of the treatment effects/treatment means. Let's look at the comparison of vegetable and animal fats.

and two different contrast statements we could write:

```
proc glimmix data=donut1;
  class type;
  model absorb=type;
  contrast "animal vs veg" type 1 1 -1 -1;
  contrast "animal vs veg 2" type 0.5 0.5 -0.5 -0.5;
run;
```

Both give the same results!

```
                      Contrasts

Num      Den
Label                 DF       DF    F Value    Pr > F

animal vs veg          1       20       5.37    0.0313
animal vs veg 2        1       20       5.37    0.0313
```

Let's try them with `estimate` statements, and add a third option:

```
estimate "animal vs veg" type 1 1 -1 -1;
estimate "animal vs veg 2" type 0.5 0.5 -0.5 -0.5;
estimate "animal vs veg 3" type 1 1 -1 -1/divisor=2;
```

```
                              Estimates

                              Standard
Label              Estimate       Error       DF    t Value    Pr > |t|

animal vs veg       19.0000      8.2016       20       2.32      0.0313
animal vs veg 2      9.5000      4.1008       20       2.32      0.0313
animal vs veg 3      9.5000      4.1008       20       2.32      0.0313

                  type Least Squares Means

                    Standard
type    Estimate       Error       DF    t Value    Pr > |t|

1         172.00      4.1008       20      41.94      <.0001
2         185.00      4.1008       20      45.11      <.0001
3         176.00      4.1008       20      42.92      <.0001
4         162.00      4.1008       20      39.50      <.0001
```

What's going on?

Suppose for some reason we wanted to test whether fats 1-3 (collectively) were different from fat 4.

The way we write the `estimate` statement really matters here:

```
estimate "first 3 vs last" type 0.33 0.33 0.33 -1;
estimate "first 3 vs last" type 1 1 1 -3/divisor=3;
```

```
                              Estimates

                              Standard
Label              Estimate       Error       DF    t Value    Pr > |t|

first 3 vs last     Non-est           .        .        .         .
first 3 vs last     15.6667      4.7352       20       3.31      0.0035
```

We do still have a multiplicity issue, because we are interested in three pre-planned contrasts. We can use the **Sidak** adjustment to control experiment-wise error rate:

```
estimate "1 vs 2" type 1 -1 0 0,
         "3 vs 4" type 0 0 1 -1,
         "animal vs veg" type 0.5 0.5 -0.5 -0.5/adjust=sidak;
```

```
                              Estimates

                                Standard
Label                Estimate       Error       DF    t Value    Pr > |t|

animal vs veg 2        9.5000      4.1008       20       2.32      0.0313
```

```
                              Estimates
                    Adjustment for Multiplicity: Sidak

                              Standard
Label              Estimate       Error       DF    t Value    Pr > |t|     Adj P

1 vs 2            -13.0000      5.7994       20      -2.24      0.0365    0.1055
3 vs 4             14.0000      5.7994       20       2.41      0.0255    0.0745
animal vs veg       9.5000      4.1008       20       2.32      0.0313    0.0909
```

Finally, we can use the `lsmestimate` statement. `lsmestimate` basically does the same thing as **estimate** but it allows for more complicated models than we have yet encountered. For a CRD, the output of the two should be identical, though `lsmestimate` does have some additional options (and slightly different syntax).

```
lsmestimate type "1 vs 2" 1 -1 0 0,
                 "3 vs 4" 0 0 1 -1,
                 "animal vs veg"  0.5 0.5 -0.5 -0.5/joint;
```

The `joint` option gives a joint test for whether the LSMeans are the same, which is the same as the overall test in the simple designs like the CRD. There are also multiple comparison adjustments available in `lsmestimate`.

What happens if you don't pre-plan? Ideally, comparisons are set up ahead of time based on specific research questions. If comparisons are selected after examining the data, most researchers construct tests that correspond to large differences in the means. These differences could be due to a real treatment effect, or they could be due to random error. Picking the largest differences to compare will inflate Type I error. If you do want to look at comparisons suggested by the data (post hoc comparisons), then you should replace the t test with a VERY conservative test called the **Scheffé** test. Scheffè works for pairwise comparisons or contrasts. We request it by adding the `adjust=scheffe` option.

To see how conservative Scheffè is, let's look at the comparison of Fats 1 vs 2 (and pretend that Fat 1 is a control, just for illustration.

Adjustment Type	p-value	Lower CL	Upper CL
Unadjusted	0.0365	-25.0974	-0.9026
Tukey	0.1462	-29.2320	3.2320
Dunnett	0.0908	-27.7326	1.7326
Scheffè	0.2044	-30.6813	4.6813

What do you notice?

Which one to use? It depends. Is it more important to control the comparison-wise error rate or experiment-wise error rate? That will depend on the situation. Keep in mind that the more conservative the adjustment, the lower the power. That is, the more likely you are to make a Type II error.

A Note on Independent Comparisons: As mentioned above, there can be as many independent comparisons as there are df for treatment. However, just because the number of planned comparisons equals the number of treatment df does not mean they are independent.

Independent contrasts are also called **orthogonal contrasts**. Orthogonality means that one contrast conveys no information about the other. We can check whether contrasts are orthogonal.

Definition: Two contrasts with coefficients c_i and d_j are orthogonal if

Let's consider the three contrasts in the donut example.

Practice: Determine if the following set of contrasts for the donut example are orthogonal.

- $H_0 : \tau_1 - \tau_2 = 0$
- $H_0 : \tau_1 - \tau_3 = 0$
- $H_0 : \tau_1 - \tau_4 = 0$

2.3 Model Adequacy

Everything we've done so far is based on the assumptions that the observations are adequately described by the model

If these assumptions are not valid, then the estimates of the treatment means and tests of significance from the ANOVA will be affected. We typically use **residuals** as a basis of our diagnostic tools.

The **residual** for observation j in treatment i is defined as:

Examining residuals should be an automatic part of the analysis of variance, and can be used to check the assumptions of common variance and normality of the error term. The assumptions can be checked using a visual inspection or formally through tests, and SAS makes it very easy to do so.

There's a lot of code here, but we'll examine it piece-by-piece.

```
proc glimmix data=donut1 plot=residualpanel;
  class type;
  model absorb=type;
  random _residual_/group=type;
  covtest homogeneity;
  output out=donutout pred=pred residual=resid;
run;
```

Here's what the options are doing:

- `plot=residualpanel` produces a set of residual plots
- `random _residual_/group=type` tells SAS you want to estimate a residual variance for each treatment group (i.e., get separate estimates of σ^2 from each treatment group)
- `covtest` produces a hypothesis test for comparing variances, and `homogeneity` says you want to test whether they are all equal
- `output` produces a new data set (called `donutout`) which contains the observed residuals (`resid`) and predicted values (`pred`)

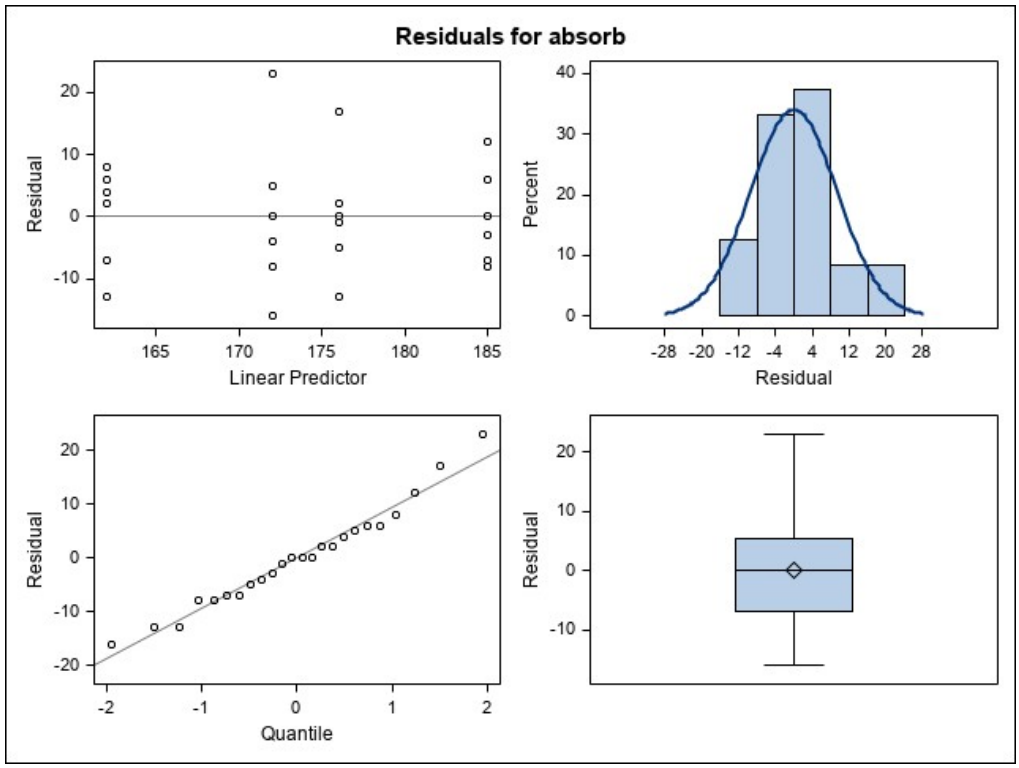


Figure 2.5: Residual panel for the donut data

The upper left hand plot shows

The other three plots all deal with the normality assumption.

We can also use `proc univariate` to check normality, using the `donoutout` data set we created above.

```
proc univariate data=donutout plot normal;
  var resid;
run;
```

Here's part of the output

```
                        Tests for Normality

Test                  --Statistic---    -----p Value------

Shapiro-Wilk          W     0.972165    Pr < W      0.7205
```

The Shapiro-Wilk test is the most commonly used test for normality. A highly significant p-value would indicate there may be a problem with non-normality.

What happens if we do see a large departure from normality?

The other assumption we can check with residuals is the constant variance assumption, also called the assumption of homogeneous variances. From the SAS output

Covariance Parameter Estimates

Cov Parm	Group	Estimate	Standard Error
Residual (VC)	type 1	178.00	112.58
Residual (VC)	type 2	60.4000	38.2003
Residual (VC)	type 3	97.6000	61.7277
Residual (VC)	type 4	67.6000	42.7540

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
type	3	20	8.39	0.0008

Tests of Covariance Parameters
Based on the Restricted Likelihood

Label	DF	-2 Res Log Like	ChiSq	Pr > ChiSq	Note
Homogeneity	3	156.21	1.90	0.5942	DF

DF: P-value based on a chi-square with DF degrees of freedom.

The covariance parameter estimates are the estimates of the variances for each of the four treatments, along with their standard errors. What do you notice?

The Tests of Covariance Parameters is testing the null hypothesis that the four variances are equal, versus the alternative that at least one is different.

In general, moderate departures from normality are of little concern, especially with the CRD. Nonconstant variance can be a bigger issue, but there are things we can do (like transformations) to stabilize the variance.

2.4 Power for the Completely Randomized Design

With multiple comparisons, we talked about Type I error and its probability. We defined Type I error as rejecting the null hypothesis when it is, in fact, true. If P(Type I error) = α, then P(no Type I error) = $1 - \alpha$.

There is another kind of error – Type II error. A Type II error occurs when H_0 is not rejected, but H_0 is actually false. In earlier courses, we summarized these two types of errors in a table:

Earlier in STAT 212, as well as other classes, we've said that we can "set" the probability of a Type I error. Anytime we say we'll reject H_0 if the p-value $< \alpha$, we're setting P(Type I error) = α. What about Type II error? We generally call the probability of a Type II error β, P(Type II error) = β. The problem is we can't "set" both β and α without some other complications.

We are typically interested in the **power** of a test:

Let's explore this via simulation. Consider a two-sample t-test. In Canvas, there is an R file called `simulation example.R`.

- In the program, we're generating $n_1 = 20$ normal random variables with $\mu_1 = 10$, $\sigma^2 = 25$ and $n_2 = 20$ normal random variables with $\mu_2 = 10$, $\sigma^2 = 25$.
- Carry out the t-test (code already included) and observe the p-value. Using $\alpha = 0.10$, what is your decision?
- Run the program 9 more times, so you have a total of 10 p-values. How many times out of 10 did you reject H_0?

  What does this estimate?

- Edit the program to generate data with $\mu_1 = 10$ and $\mu_2 = 12$ (still with $n_1 = n_2 = 20$ and $\sigma^2 = 25$). Run the program 10 times total. How many times did you reject H_0, using $\alpha = 0.10$?
- Edit the program to generate data with $\mu_1 = 10$ and $\mu_2 = 15$ (still with $n_1 = n_2 = 20$ and $\sigma^2 = 25$). Run the program 10 times total. How many times did you reject H_0, using $\alpha = 0.10$?
- Edit the program to generate data with $\mu_1 = 10$ and $\mu_2 = 20$ (still with $n_1 = n_2 = 20$ and $\sigma^2 = 25$). Run the program 10 times total. How many times did you reject H_0, using $\alpha = 0.10$?

What do you observe as μ_1 and μ_2 get further apart?

Now let's try the following:

- Edit the program to generate data with $\mu_1 = 10$ and $\mu_2 = 12$, but with $\sigma^2 = 1$ (still with $n_1 = n_2 = 20$). Run the program 10 times total. How many times did you reject H_0, using $\alpha = 0.10$?
- Edit the program to generate data with $\mu_1 = 10$ and $\mu_2 = 20$, but with $\sigma^2 = 625$ (still with $n_1 = n_2 = 20$). Run the program 10 times total. How many times did you reject H_0, using $\alpha = 0.10$?

What do observe as σ^2 gets larger or smaller?

Power is the probability of rejecting H_0 when it is really false. It is a function of several quantities:

-
-
-
-

Power analyses usually focus on calculating the sample size required to achieve a particular power. What do you think would happen if instead of using $n_1 = n_2 = 20$ we used $n_1 = n_2 = 10$?

What do you think would happen if instead of using $n_1 = n_2 = 20$ we used $n_1 = n_2 = 40$?

What do you think would happen if instead of using $n_1 = n_2 = 20$ we used $n_1 = 10$ and $n_2 = 30$?

In some simple situations, we can SAS procs to do power calculations. There are two: `PROC POWER` and `PROC GLMPOWER`. In some more complicated situations we'll need to write our own code. We'll start with `PROC POWER`. This proc will do power calculations for two sample t tests and ANOVA.

For a two-sample t test, the basic code is:

```
proc power;
  twosamplemeans test=diff
  alpha=
  stddev=
  meandiff=
  npergroup=
  power=               ;
run;
```

We'll need to supply values for alpha, stddev, and meandiff. We can either supply a value for npergroup and use `power=.` or supply a value for power and use `npergroup=.`

We could also add the lines

- `plot x=power min=0.5 max=0.95;` (for `ntotal=.`)
- `x=n min= max= ;` (for `power=.`)

Let's try this, going back to our example with $\mu_1 = 10$, $\mu_2 = 12$, and $\sigma^2 = 25$.

We can also use PROC POWER for ANOVA and contrasts. This time, the basic code is:

```
proc power;
  onewayanova
  alpha=
  stddev=
  groupmeans=  |   |
  ntotal=
  power=
  contrast= (     );
run;
```

Let's go back to the donut data. In that example, the MSE was 100.90, so we'll use $\sigma = 10$ as a guess for future experiments. We observed sample means of $\overline{y}_{1\cdot} = 172$, $\overline{y}_{2\cdot} = 185$, $\overline{y}_{3\cdot} = 176$, and $\overline{y}_{4\cdot} = 162$. We can certainly use these as guesses for future experiments. We'll consider the contrast testing animal fats versus vegetable fats. We could also look at the overall test.

We can also add plot statements here.

But, we don't actually have to have guesses for the treatment means. We do have to have an idea of how large a difference we want to be able to detect. With our example data, we had a animal fat mean of 178.5 and a vegetable fat mean of 169. This is a difference of 9.5.

We could also the consider potential differences we might observe in pairwise differences.

There are situations, however, where PROC POWER doesn't work. It can't handle more than one factor, a treatment design we'll see in a couple of weeks. And, it can't handle any model with a random effect other than e_{ij}, a consequence of experimental designs we'll see shortly.

Let's go back to the donut scenario. In that case, the ANOVA table was

Source of Variation	df	SS	MS	Expected MS	F
Fat Type	$t-1=4-1=3$	SSTrt	MST	$\sigma^2+\frac{n}{t-1}\sum_{i=1}^{4}\tau_i^2$	MST/MSE
Error	$t(n-1)=$ $4(6-1)=20$	SSError	MSE	σ^2	
Total	$nt-1=$ $(4)(6)-1=23$	SSTotal			

When we considered this ANOVA table back in Chapter 3, we observed

But, power is about detecting when the null hypothesis is NOT true. It turns out that if H_0 is not true, the F ratio follows a different distribution:

The term SSHyp means

We can determine the power of an overall F test or contrast under a specified alternative by tricking `PROC GLIMMIX` into computing λ and then using it to compute power. There are three basic steps.

1. Generate data set where all y_{ij} are set to the alternative value μ_i (the "true" μ_i MUST be specified by the researcher). A "true" value of σ^2 must also be specified.

2. Run `PROC GLIMMIX` on these data, but fix σ^2 at the value specified by the researcher (don't let SAS estimate it). Use `GLIMMIX` to calculate the overall F and F values for any contrasts. This means the `contrast` statement must be used, not `estimate` or `lsmestimate`. The product of the numerator degrees of freedom and the resulting overall ANOVA and contrast F values are the λs.
3. Use the results from (2) in the SAS F probability functions to compute power.

Let's go back to the donuts.

1. Step 1: Generate or simulate the data of "true" means.

```
data donutpower;
  input trt mu;
  do eu=1 to 6; *6 batches per treatment;
    output;
  end;
datalines;
1 172
2 185
3 176
4 162
;
```

2. Step 2: Run `GLIMMIX`, fixing the true value of σ^2 at 100.

```
proc glimmix data=donutpower;
  class trt;
  model mu=trt;
  parms (100)/hold=1;
  contrast 'animal vs veg' trt 0.5 0.5 -0.5 -0.5;
ods output tests3=b;
ods output contrasts=c;
run;

proc print data=b;
proc print data=c; run;
```

This produces the output

The SAS System

Obs	Effect	NumDF	DenDF	FValue	ProbF
1	trt	3	20	5.46	0.0066

The SAS System

Obs	Label	NumDF	DenDF	FValue	ProbF
1	animal vs veg	1	20	5.42	0.0306

Figure 2.6: The results of Step 2 in the power calculation.

3. Step 3: Use the results from `GLIMMIX` to find the power.

```
data powerval;
 set b c;
 do alpha=0.10, 0.05, 0.01;
   lambda=numdf*fvalue;
   fcrit=finv(1-alpha, numdf, dendf, 0);
   power=1-probf(fcrit, numdf, dendf,lambda);
   output;
 end;
proc print; run;
```

Which gives

The SAS System

Obs	Effect	NumDF	DenDF	FValue	ProbF	Label	alpha	lambda	fcrit	power
1	trt	3	20	5.46	0.0066		0.10	16.365	2.38009	0.94158
2	trt	3	20	5.46	0.0066		0.05	16.365	3.09839	0.88195
3	trt	3	20	5.46	0.0066		0.01	16.365	4.93819	0.66653
4		1	20	5.42	0.0306	animal vs veg	0.10	5.415	2.97465	0.72657
5		1	20	5.42	0.0306	animal vs veg	0.05	5.415	4.35124	0.60044
6		1	20	5.42	0.0306	animal vs veg	0.01	5.415	8.09596	0.33017

Figure 2.7: The results of Step 3 in the power calculation.

2.5 Quantitative Levels of a Factor

As a reminder, we've considered four basic treatment structures that may be used in a one-way treatment design:

1. Unstructured
2. Control versus other treatments
3. Quantitative
4. Other structure

The last of the four treatment structures we still need address is the quantitative factor. Up to now, we've only considered qualitative factors. A **qualitative factor** is

We've seen many of these examples:

On the other hand, a **quantitative factor** is

When we look at the initial experimental design (how the treatments are assigned to experimental units) and analysis of a study, it doesn't really matter whether our factor is quantitative or qualitative. The researcher wants to know whether the response differs based on the treatment levels.

However, many studies that have a quantitative factor are conducted because

The general approach to fitting these models is regression analysis. The functional relationship between the quantitative factor and the response may be linear or non-linear, and a polynomial model is often used to describe the functional relationship. It is the simplest of the linear functions.

The general form of a **polynomial model of** p degrees is

The objective is to determine

Example: A researcher is interested in studying the tensile strength of a new synthetic fiber to use in a cotton blend for shirts. The researcher knows the strength of material is affected by the percent (by weight) of cotton used in the blend and a range between 10 and 40% cotton is necessary to meet other characteristics criteria, like accepting a no-iron treatment. The researcher decides to test a range of percentages of cotton: 15, 20, 25, 30, and 35% and has the resources to test five samples at each level. The data are:

	Obs				
Cotton %	1	2	3	4	5
15	7	7	15	11	9
20	12	17	12	18	18
25	15	19	19	21	19
30	19	25	22	19	23
35	7	10	11	15	11

- **Treatment Design:**
  - Factor:
  - Levels:
- **Experimental Design:**
  -

Let's first look at a plot of the treatment means by treatment level. We'll use this code:

```
proc glimmix data=cotton;
  class percent; *treating percent as a qualitative var only to get the trt means;
  model strength=percent;
  lsmeans percent/plot=meanplot(join); *connect the dots;
run;
```

This produces the plot

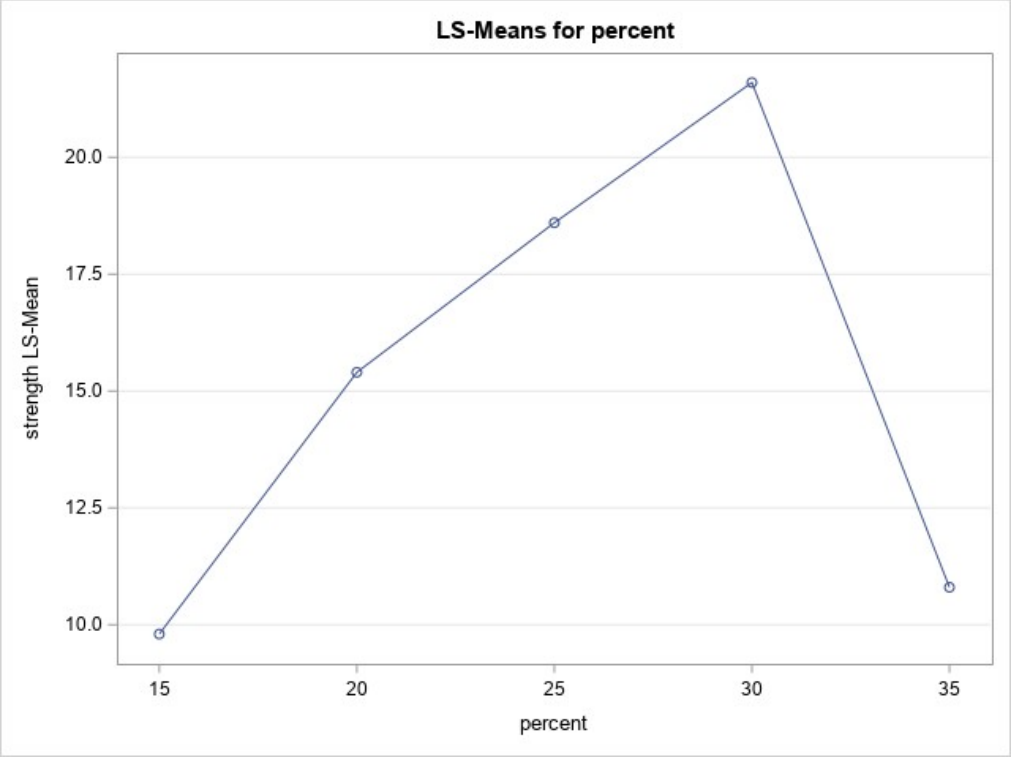


Figure 2.8: Plot of the lsmeans corresponding to the levels of percent cotton.

Anytime we're fitting a polynomial model, the number of degrees p that can be fit is

and each degree corresponds to 1 treatment degree of freedom. If we fit all terms, we get a complete partition of the treatment effect into its one degree of freedom components.

In the case, the highest order model we can fit is

From the plot, it appears

The first model we'll try fitting is a quartic model:

where

There are two models/methods for fitting the polynomial model:

- direct regression
- orthogonal polynomials

Direct regression

The **direct regression** approach considers the treatment/explanatory variable as a regression variable, not an ANOVA variable (though we know they're really the same model at their core!). Because we are treating our explanatory varibale as a regression variable, it *must not* appear in the `class` statement. The following code fits a quartic model to the data:

```
proc glimmix data=cotton;
  model strength=percent percent*percent percent*percent*percent
                 percent*percent*percent*percent/htype=1,3;
/*alternate model statement */
* model strength=percent|percent|percent|percent/htype=1,3;
run;
```

A few notes:

- We're asking for both Type I and Type III tests.

- Note the alternate `model` statement.

Here's part of the output

Type I Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
percent	1	20	4.12	0.0559
percent*percent	1	20	47.66	<.0001
percen*percen*percen	1	20	7.96	0.0105
perc*perc*perc*perce	1	20	2.19	0.1549

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
percent	1	20	1.57	0.2242
percent*percent	1	20	1.67	0.2115
percen*percen*percen	1	20	1.88	0.1857
perc*perc*perc*perce	1	20	2.19	0.1549

The **Type I Tests**

So, it looks like

The **Type III Tests**

So now we can rerun the analyses, keeping the terms up to the highest order that has a significant contribution. In our example

We'll use the code:

```
proc glimmix data=cotton;
  model strengt=percent|percent|percent/solution htype=1;
run;
```

Note we've added the `solution` option, which will provide the parameter estimates for the effects. Here's the output:

Parameter Estimates

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	59.5257	38.2927	21	1.55	0.1350
percent	-8.7257	5.0052	21	-1.74	0.0959
percent*percent	0.4757	0.2081	21	2.29	0.0327
percen*percen*percen	-0.00760	0.002768	21	-2.75	0.0121
Scale	8.6205	2.6604	.	.	.

Type I Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
percent	1	21	3.90	0.0616
percent*percent	1	21	45.12	<.0001
percen*percen*percen	1	21	7.54	0.0121

We can use these estimates to construct the fitted model:

We can also plot the predicted values from our equation

```
proc glimmix data=cotton;
  model strength=percent|percent|percent/solution htype=1;
  output out=cottonout pred=predicted;
run;

proc print data=cottonout; run;
```

Which gives us the output data set that I've called `cottonout`:

Obs	percent	strength	predicted
1	15	7	10.0257
2	15	7	10.0257
3	15	15	10.0257
4	15	11	10.0257
5	15	9	10.0257
6	20	12	14.4971
7	20	17	14.4971
8	20	12	14.4971
9	20	18	14.4971
10	20	18	14.4971
11	25	15	19.9543
12	25	19	19.9543
13	25	19	19.9543
14	25	21	19.9543
15	25	19	19.9543
16	30	19	20.6971
17	30	25	20.6971
18	30	22	20.6971
19	30	19	20.6971
20	30	23	20.6971
21	35	7	11.0257
22	35	10	11.0257
23	35	11	11.0257
24	35	15	11.0257
25	35	11	11.0257

To plot the predicted values there are few options

```
title "Regression model for cotton data";
symbol1 interpol=none value=dot color=black;
symbol2 interpol=join value=diamond color=red;
proc gplot data=cottonout;
  plot strength*percent predicted*percent/overlay;
run;

title "Regression model for cotton data";
symbol1 interpol=none value=dot color=black;
symbol2 interpol=RC value=diamond color=red; *can only go up to a cubic;
proc gplot data=cottonout;
  plot strength*percent predicted*percent/overlay;
run;

title "Regression model for cotton data";
symbol1 interpol=none value=dot color=black;
symbol2 interpol=spline value=diamond color=red;
proc gplot data=cottonout;
  plot strength*percent predicted*percent/overlay;
run;
```

Let's go to SAS to see the differences among these three plots.

Notice that at the percents outside the inference space we get nonsense strength values. Only values between 15 and 35 percent are within the inference space.

Orthogonal polynomials

Quantitative factors levels are one of the few cases where a full set of **orthogonal contrasts** can be useful. However, this approach does require

and involves testing orthogonal contrasts among the treatment factor levels. These contrasts allow us to evaluate the importance of each polynomial component with a specific contrast.

Here are the steps:

-
-
-
-
-

The coefficients for polynomial contrasts are not obvious. It's easiest to look them up (google orthogonal polynomial coefficients table). For a design with 5 equally spaced levels:

$v = 5$

Trend	c_1	c_2	c_3	c_4	c_5
Linear	−2	−1	0	1	2
Quadratic	2	−1	−2	−1	2
Cubic	−1	2	0	−2	1
Quartic	1	−4	6	−4	1

Figure 2.9: Orthogonal polynomial coefficients for 5 equally spaced levels.

So, our code would look like:

```
proc glimmix data=cotton;
  class percent;
  model strength=percent;
  contrast 'linear' percent -2 -1 0 1 2;
  contrast 'quadratic' percent 2 -1 -2 -1 2;
  contrast 'cubic' percent -1 2 0 -2 1;
  contrast 'quartic' percent 1 -4 6 -4 1;
run;
```

Which gives the output:

```
       Type III Tests of Fixed Effects

              Num      Den
Effect         DF       DF    F Value    Pr > F

percent         4       20      15.48    <.0001


                  Contrasts

              Num      Den
Label         DF       DF    F Value    Pr > F

linear         1       20       4.12    0.0559
quadratic      1       20      47.66    <.0001
cubic          1       20       7.96    0.0105
quartic        1       20       2.19    0.1549
```

Note that these results are the same as we got with the direct regression method!

3 Factorial Treatment Designs

Often we are interested in more than one treatment factor. For example, we may want to look at

A **factorial arrangement of treatments** is a **treatment design** that is the most efficient approach for this type of experiment. For example, we've already seen

The goal is to describe *as simply as possible* how these factors affect our response variable.

Why use Factorials?

We could investigate a single factor as a time:

-
-
-
-
-

Alternatively, we could investigate every possible combination in a single experiment:

-
-
-

3.1 Introduction and Model

We'll start by looking at the general set-up for a 2-factor design. Designate one factor as Factor A. Factor A has a levels. Designate the other factor as Factor B. Factor B has b levels. This means there are a total of $a \times b$ treatment combinations. We will still be thinking about this factorial treatment design in the context of the CRD experimental design.

Example: In lab, you explored some data resulting from an experiment designed to study the effects of four treatments on the inches of fabric burned after a flame test. In the lab problem you were told

- Treatment A = Cotton fabric, fire-retardant additive X
- Treatment B = Polyester fabric, fire-retardant additive X
- Treatment C = Cotton fabric, fire-retardant additive Y
- Treatment D = Polyester fabric, fire-retardant additive Y

So really, rather than 1 factor (Treatment) with 4 levels, we have

Let's plot the data. Does it look like there is a difference among the treatments? Can we (easily) see if there are differences among different levels of the two factors?

Your group was asked to come up with two contrasts. What contrasts did you use?

If we treat this experiment as the factorial treatment design that it is, we can look at the factors individually and in combination. We'll need to start by adjusting the model and our notation.

The treatment combinations and their means can be summarized in a table

Table 3.1: Treatment Combinations and Means

		Fabric	
Additive		Cotton	Polyester
	X		
	Y		

These are fixed, unknown values. The observed treatment means and marginal means are:

Table 3.2: Observed Cell and Marginal Means

		Fabric		
Additive		Cotton	Polyester	Additive Means
	X			
	Y			
Fabric Means				

With a single factor, we could write the model in one of two ways:

With two factors, we can still write the model in one of two ways:

Before we can see how this new model gets translated into an ANOVA table, we need to explore the **effect** of a factor.

3.2 Effects

Definition: The **effect** of a factor

There are three types of effects in a factorial experiment:

-
-
-

To demonstrate these effects, we'll go back to the pajamas example.

3.2.1 Simple Effects

Simple effects give the effect of one factor when the other(s) is held constant. So, a simple effect is the difference between two levels of a factor at a given level of the remaining factor(s).

In our pajama example there are 4 simple effects:

- The simple effects comparing Cotton to Polyester:
  - given Additive X
  - given Additive Y
- The simple effects comparing Additive X to Additive Y:
  - given Cotton fabric
  - given Polyester fabric

We can estimate the simple effects using our observed treatment means:

- The **estimated** simple effects comparing Cotton to Polyester:
  - given Additive X
  - given Additive Y
- The **estimated** simple effects comparing Additive X to Additive Y:
  - given Cotton fabric
  - given Polyester fabric

3.2.2 Main Effects

Main effects are averages of simple effects. Main effects average over a variety of conditions.

In our pajama example there are two main effects:

- Main effect for Fabric type:

- Main effect for Additive:

Main effects can also be thought of as differences between marginal means for a given factor. Let's look more closely at Fabric type effect:

We could also re-write the main effect for Additive in a similar way.

3.2.3 Interaction Effects

Interaction Effects are differences of simple effects, and interaction effects ask the question:

We do this by looking at the difference in the simple effects for one factor at two different levels of the second factor.

For example,

- The simple effect of Fabric given Additive X
- The simple effect of Fabric given Additive Y

Is the effect of Fabric the same for both Additives?

What about the effect of Additive for both Fabrics?

Presence or absence of interaction effects can be illustrated graphically. We plot the response mean against one of the factors for both levels of the other factor. If the lines are approximately parallel, there is no interaction. If the lines are not parallel, there is interaction.

Important Note: The presence or absence of main effects tells us NOTHING about the presence or absence of interaction.The presence or absence of interaction tells us NOTHING about the presence or absence of main effects. The presence of interaction DOES tell us something about the homogeneity of simple effects. When factors interact, a single factor experiment will lead to disconnected and possibly misleading results.

Why do we care?

One thing we noted at the beginning of this section

The presence or absence of interaction helps inform how we can best report the findings of the experiment.

- Interaction is non-significant:

- Interaction is significant

It's never **wrong** to report simple effects,

How do we determine significant? That brings us to . . .

3.3 Model and Analysis

We'll start with the simplest type of factorial design with 2 factors. As we mentioned earlier, we'll assume a levels of Factor A and b levels of Factor B arranged in a **factorial treatment design**. That is, each replication of the experiment contains all $a \times b$ treatment combinations. In general, there are n replicates. We call this a

Recall that y_{ijk} represents

In general (assuming equal sample sizes, which is not necessary), the data can be represented as

We've been assuming that μ_{ij} represents the true mean for cell (i, j). This implies the model

This is the **cell means model**. Like the the one-factor model, we can also write this using a **treatment effects model**:

Just as our model grows bigger, so does our ANOVA table:

We can formally test for interaction by testing the hypotheses

If there is no interaction, we look at main effects. For the main effects we can test hypotheses about the equality of Factor A treatment effects:

and the equality of Factor B effects:

and the main effects can be used to summarize the experiment.

If interaction is present, we must summarize the experiment using simple effects. Main effects would not adequately represent the effect of the factors.

Example: Back to the pajamas data. We had two factors each with two levels, and 4 observations at each treatment combination. The linear model would be

Let's sketch the ANOVA table:

From the interaction plot we saw earlier, it was unclear whether interaction is present or not. We can fit the ANOVA model using the code:

```
proc glimmix data=pajamas;
   class fabric additive;
     model burn=fabric additive fabric*additive;
  run;
```

And here is the output:

Fit Statistics

-2 Res Log Likelihood	67.28
AIC (smaller is better)	77.28
AICC (smaller is better)	87.28
BIC (smaller is better)	79.71
CAIC (smaller is better)	84.71
HQIC (smaller is better)	76.38
Pearson Chi-Square	120.50
Pearson Chi-Square / DF	10.04

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
fabric	1	12	4.88	0.0474
additive	1	12	75.31	<.0001
fabric*additive	1	12	4.21	0.0627

What do you think?

We can output the estimated effects and get a plot using

```
lsmeans fabric*additive/diff plot=diffplot;
```

which gives

fabric*additive Least Squares Means

fabric	additive	Estimate	Standard Error	DF	t Value	Pr > |t|
C	X	40.5000	1.5844	12	25.56	<.0001
C	Y	30.0000	1.5844	12	18.93	<.0001
P	X	47.2500	1.5844	12	29.82	<.0001
P	Y	30.2500	1.5844	12	19.09	<.0001

Differences of fabric*additive Least Squares Means

fabric	additive	_fabric	_additive	Estimate	Standard Error	DF	t Value	Pr > |t|
C	X	C	Y	10.5000	2.2407	12	4.69	0.0005
C	X	P	X	-6.7500	2.2407	12	-3.01	0.0108
C	X	P	Y	10.2500	2.2407	12	4.57	0.0006
C	Y	P	X	-17.2500	2.2407	12	-7.70	<.0001
C	Y	P	Y	-0.2500	2.2407	12	-0.11	0.9130
P	X	P	Y	17.0000	2.2407	12	7.59	<.0001

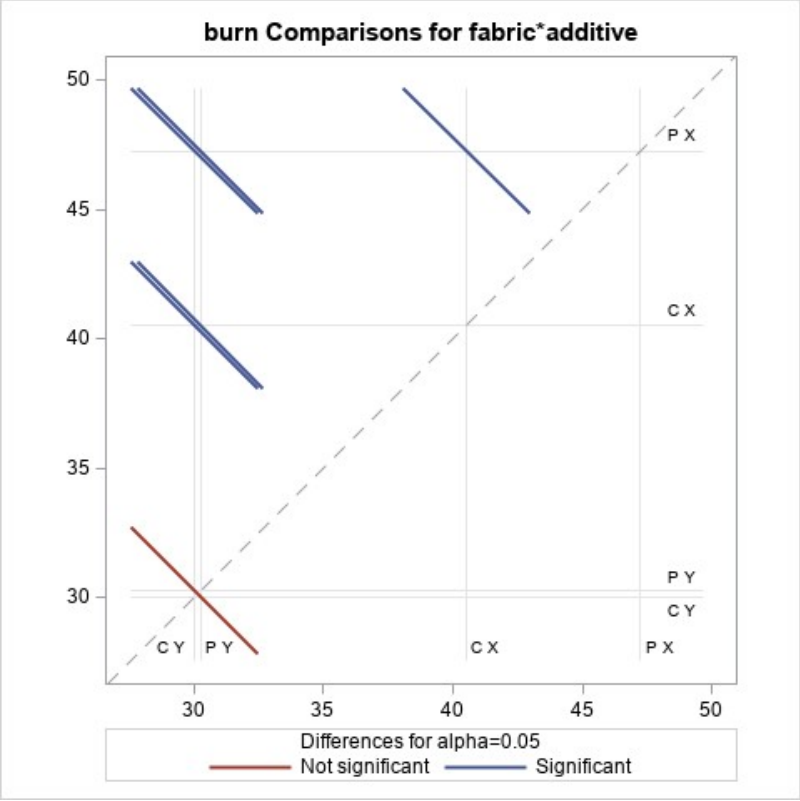


Figure 3.1: Diffogram for pajama lsmeans

What does this plot show us?

Although we generally worry about testing the interaction effect, there may be times where we want to estimate the interaction. That is, we may be interested in the magnitude of the difference between the simple effects. We can do this by adding an `estimate` statement to the `PROC GLIMMIX` program:

```
estimate 'fabric x additive' fabric*additive 1 -1 -1 1/alpha=0.1;
```

This gives the output

Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
fabric x additive	-6.5000	3.1689	12	-2.05	0.0627	0.1	-12.1478	-0.8522

`PROC GLIMMIX` has both `slice` and `slicediff` options available that allow us to test one factor at each level of the other. The `slice` option is similar to a contrast statement, in that it only gives us the F stats and p-values. The `slicediff` option allows us to construct confidence intervals. Let's see what they provide:

`slice` option:

```
lsmeans fabric*additive/slice=fabric slice=additive;
```

Tests of Effect Slices for fabric*additive
Sliced By fabric

fabric	Num DF	Den DF	F Value	Pr > F
C	1	12	21.96	0.0005
P	1	12	57.56	<.0001

Tests of Effect Slices for fabric*additive
Sliced By additive

additive	Num DF	Den DF	F Value	Pr > F
X	1	12	9.07	0.0108
Y	1	12	0.01	0.9130

slicediff option:

```
lsmeans fabric*additive/slicediff=fabric slicediff=additive cl;
```

fabric*additive Least Squares Means

fabric	additive	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
C	X	40.5000	1.5844	12	25.56	<.0001	0.05	37.0478	43.9522
C	Y	30.0000	1.5844	12	18.93	<.0001	0.05	26.5478	33.4522
P	X	47.2500	1.5844	12	29.82	<.0001	0.05	43.7978	50.7022
P	Y	30.2500	1.5844	12	19.09	<.0001	0.05	26.7978	33.7022

Simple Effect Comparisons of fabric*additive Least Squares Means By fabric

Simple Effect Level	additive	_additive	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha
fabric C	X	Y	10.5000	2.2407	12	4.69	0.0005	0.05
fabric P	X	Y	17.0000	2.2407	12	7.59	<.0001	0.05

Simple Effect Comparisons of fabric*additive Least Squares Means By fabric

Simple Effect Level	additive	_additive	Lower	Upper
fabric C	X	Y	5.6179	15.3821
fabric P	X	Y	12.1179	21.8821

Simple Effect Comparisons of fabric*additive Least Squares Means By additive

Simple Effect Level	fabric	_fabric	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha
additive X	C	P	-6.7500	2.2407	12	-3.01	0.0108	0.05
additive Y	C	P	-0.2500	2.2407	12	-0.11	0.9130	0.05

Simple Effect Comparisons of fabric*additive Least Squares Means By additive

Simple Effect Level	fabric	_fabric	Lower	Upper
additive X	C	P	-11.6321	-1.8679
additive Y	C	P	-5.1321	4.6321

We may also construct confidence intervals for the marginal and cell means in a factorial experiment. To do this, we need the standard error of those means.

- Marginal mean of Factor A

- Marginal mean of Factor B

- Cell means

So, a confidence interval for

- Marginal mean

- Cell mean

We could also get a confidence interval for an interaction effect. Recall that an interaction effect for a 2×2 factorial is a contrast with the coeefficients 1 -1 -1 1. The standard error for a contrast is

So for the pajamas example, the standard error would be

which would give a confidence interval

Using our pajamas example, we'll use the following SAS program to see how to calculate estimates of the main effects, simple effects, and interaction effects to compare with what we did by hand.

```
proc glimmix data=pajamas;
    class fabric additive;
    model burn=fabric additive fabric*additive;
    lsmeans fabric additive fabric*additive/diff cl;
    estimate 'main effect cotton fabric' intercept 2 fabric 2 0 additive 1 1
                fabric*additive 1 1 0 0/divisor=2 cl;
    estimate 'main effect polyester fabric' intercept 2 fabric 0 2 additive 1 1
                fabric*additive 0 0 1 1/divisor=2 cl;
    estimate 'main effect additive X' intercept 2 additive 2 0 fabric 1 1
                fabric*additive 1 0 1 0/divisor=2 cl;
    estimate 'main effect additive Y' intercept 2 additive 0 2 fabric 1 1
                fabric*additive 0 1 0 1/divisor=2 cl;
    estimate 'effect of fabric with add X' fabric 1 -1 fabric*additive 1 0 -1 0/cl;
    estimate 'effect of fabric with add Y' fabric 1 -1 fabric*additive 0 1 0 -1/cl;
    estimate 'effect of additive in cotton' additive 1 -1 fabric*additive 1 -1 0 0/cl;
    estimate 'effect of additive in polyester' additive 1 -1 fabric*additive 0 0 1 -1/cl;
    estimate 'fabric x additive' fabric*additive 1 -1 -1 1/cl;
    contrast 'fabric with add X' fabric 1 -1 fabric*additive 1 0 -1 0;
    contrast 'fabric with add Y' fabric 1 -1 fabric*additive 0 1 0 -1;
    contrast 'additive in cotton' additive 1 -1 fabric*additive 1 -1 0 0;
    contrast 'additive in polyester' additive 1 -1 fabric*additive 0 0 1 -1;
    contrast 'interaction' fabric*additive 1 -1 -1 1;
run;
```

Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha
main effect cotton fabric	35.2500	1.1204	12	31.46	<.0001	0.05
main effect polyester fabric	38.7500	1.1204	12	34.59	<.0001	0.05
main effect additive X	43.8750	1.1204	12	39.16	<.0001	0.05
main effect additive Y	30.1250	1.1204	12	26.89	<.0001	0.05
effect of fabric with add X	-6.7500	2.2407	12	-3.01	0.0108	0.05
effect of fabric with add Y	-0.2500	2.2407	12	-0.11	0.9130	0.05
effect of additive in cotton	10.5000	2.2407	12	4.69	0.0005	0.05
effect of additive in polyester	17.0000	2.2407	12	7.59	<.0001	0.05
fabric x additive	-6.5000	3.1689	12	-2.05	0.0627	0.05

Estimates

Label	Lower	Upper
main effect cotton fabric	32.8089	37.6911
main effect polyester fabric	36.3089	41.1911
main effect additive X	41.4339	46.3161
main effect additive Y	27.6839	32.5661
effect of fabric with add X	-11.6321	-1.8679
effect of fabric with add Y	-5.1321	4.6321
effect of additive in cotton	5.6179	15.3821
effect of additive in polyester	12.1179	21.8821
fabric x additive	-13.4044	0.4044

Contrasts

Label	Num DF	Den DF	F Value	Pr > F
fabric with add X	1	12	9.07	0.0108
fabric with add Y	1	12	0.01	0.9130
additive in cotton	1	12	21.96	0.0005
additive in polyester	1	12	57.56	<.0001
interaction	1	12	4.21	0.0627

fabric Least Squares Means

fabric	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
C	35.2500	1.1204	12	31.46	<.0001	0.05	32.8089	37.6911
P	38.7500	1.1204	12	34.59	<.0001	0.05	36.3089	41.1911

Differences of fabric Least Squares Means

fabric	_fabric	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
C	P	-3.5000	1.5844	12	-2.21	0.0474	0.05	-6.9522	-0.04782

additive Least Squares Means

additive	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
X	43.8750	1.1204	12	39.16	<.0001	0.05	41.4339	46.3161
Y	30.1250	1.1204	12	26.89	<.0001	0.05	27.6839	32.5661

Differences of additive Least Squares Means

additive	_additive	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
X	Y	13.7500	1.5844	12	8.68	<.0001	0.05	10.2978	17.2022

fabric*additive Least Squares Means

fabric	additive	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
C	X	40.5000	1.5844	12	25.56	<.0001	0.05	37.0478	43.9522
C	Y	30.0000	1.5844	12	18.93	<.0001	0.05	26.5478	33.4522
P	X	47.2500	1.5844	12	29.82	<.0001	0.05	43.7978	50.7022
P	Y	30.2500	1.5844	12	19.09	<.0001	0.05	26.7978	33.7022

Differences of fabric*additive Least Squares Means

fabric	additive	_fabric	_additive	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha
C	X	C	Y	10.5000	2.2407	12	4.69	0.0005	0.05
C	X	P	X	-6.7500	2.2407	12	-3.01	0.0108	0.05
C	X	P	Y	10.2500	2.2407	12	4.57	0.0006	0.05
C	Y	P	X	-17.2500	2.2407	12	-7.70	<.0001	0.05
C	Y	P	Y	-0.2500	2.2407	12	-0.11	0.9130	0.05
P	X	P	Y	17.0000	2.2407	12	7.59	<.0001	0.05

Differences of fabric*additive Least Squares Means

fabric	additive	_fabric	_additive	Lower	Upper
C	X	C	Y	5.6179	15.3821
C	X	P	X	-11.6321	-1.8679
C	X	P	Y	5.3679	15.1321
C	Y	P	X	-22.1321	-12.3679
C	Y	P	Y	-5.1321	4.6321
P	X	P	Y	12.1179	21.8821

Let's look more closely at the `SAS` code to see where those coefficients are coming from. Specifically, we'll look at `estimate` statements 1 and 2 in detail.

The value provided by the `estimate` statement is constructed based on the estimates of the effects in the model: the estimates of

We've used cell means (treatment combination means) to see where main, simple, and interaction effects come from, but they can also be expressed as a function of the model effects.

For the pajama example, the `lsmeans` in `SAS` are

and the treatment effects model is

- `estimate` statement #1: main effect for Cotton fabric

- `estimate` statement #2: (simple) effect of fabric with Additive X

Why does learning this matter, if we're duplicating results from `lsmeans` statements?

- For practice, try `estimate` statement #3: main effect for Additive X

- For practice, try `estimate` statement #4: (simple) effect of additive with Polyester fabric

3.4 More Than Two Levels per Treatment Factor

In the pajama example, we had two levels for each of our two factors. Fabric was either cotton or polyester, and Additive was either X or Y. Now we'll consider a factorial with more than two levels for each factor.

Example: An experiment was conducted to aid in developing a product that can be used as a substrate for making ribbons. The experiment was designed to investigate the effects of base polymer and additive on the tensile strength of the resulting ribbon. There are two factors of interest: (1) base polymer: mylar, nylon, and polyethylene and (2) additive: c1, c2, c3, c4, and c5. There are 3×5 treatment combinations, and the researchers plan to test each treatment combination 6 times, for a total of 90 observations. The data are in the `ribbon.sas` file.

The model is

and the ANOVA table is

Let's look at an interaction plot to see what's going on. We can also get an interaction plot using the `lsmeans` statement!

```
proc glimmix data=ribbon;
  class polymer add;
  model strength=polymer add polymer*add;
  lsmeans polymer*add/plots=meanplot(sliceby=polymer join);
run;
```

Which gives

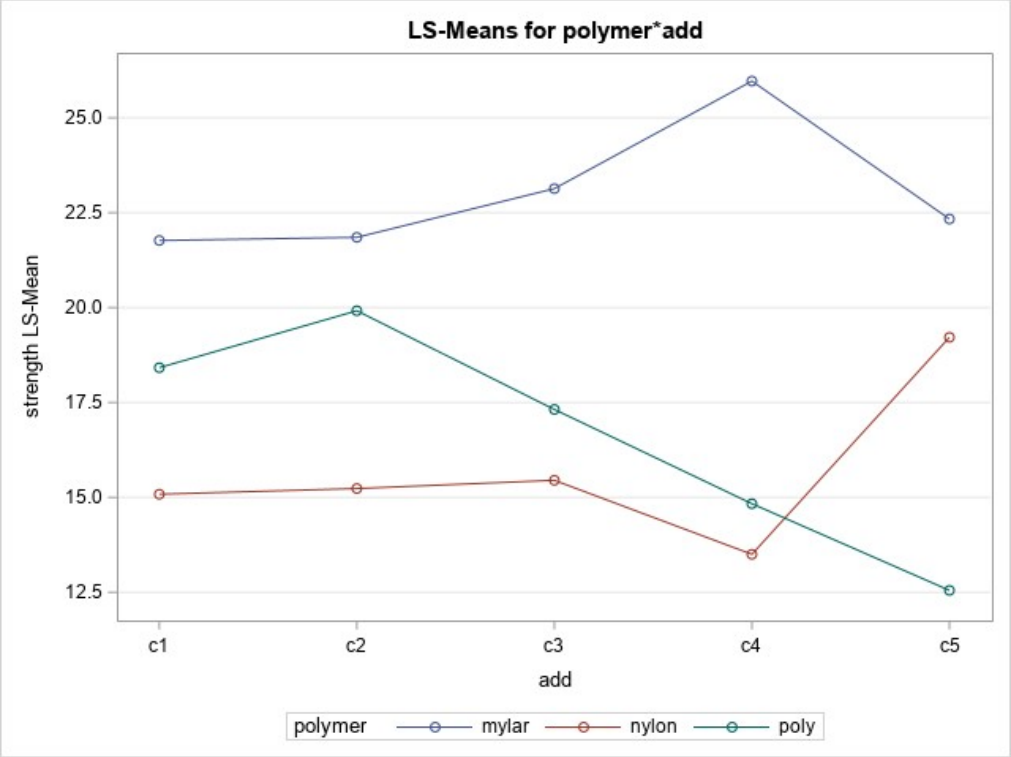


Figure 3.2: Interaction plot for the ribbon data

So it looks like we have interaction!

We'll also formally test for interaction

```
           Type III Tests of Fixed Effects
                Num      Den
Effect           DF       DF    F Value    Pr > F
polymer           2       75      24.25    <.0001
add               4       75       0.14    0.9648
polymer*add       8       75       2.38    0.0241
```

This verifies what we see in the plot.

So, we must look at simple effects and not main effects. We'll use the same `slice` and `slicediff` statements we used with the pajama example:

```
proc glimmix data=ribbon;
  class polymer add;
  model strength=polymer add polymer*add;
  lsmeans polymer*add/slice=polymer slice=add slicediff=(polymer add);
run;
```

This is going to produce a LOT of output. Let's look at the output for `slice=add` and `slicediff=add`.

Tests of Effect Slices for polymer*add Sliced By add

add	Num DF	Den DF	F Value	Pr > F
c1	2	75	3.41	0.0383
c2	2	75	3.53	0.0341
c3	2	75	4.90	0.0100
c4	2	75	14.31	<.0001
c5	2	75	7.63	0.0010

Simple Effect Comparisons of polymer*add Least Squares Means By add

Simple Effect Level	polymer	_polymer	Estimate	Standard Error	DF	t Value	Pr > |t|
add c1	mylar	nylon	6.6833	2.5593	75	2.61	0.0109
add c1	mylar	poly	3.3500	2.5593	75	1.31	0.1945
add c1	nylon	poly	-3.3333	2.5593	75	-1.30	0.1968
add c2	mylar	nylon	6.6167	2.5593	75	2.59	0.0117
add c2	mylar	poly	1.9333	2.5593	75	0.76	0.4524
add c2	nylon	poly	-4.6833	2.5593	75	-1.83	0.0712
add c3	mylar	nylon	7.6833	2.5593	75	3.00	0.0036
add c3	mylar	poly	5.8167	2.5593	75	2.27	0.0259
add c3	nylon	poly	-1.8667	2.5593	75	-0.73	0.4681
add c4	mylar	nylon	12.4667	2.5593	75	4.87	<.0001
add c4	mylar	poly	11.1333	2.5593	75	4.35	<.0001
add c4	nylon	poly	-1.3333	2.5593	75	-0.52	0.6039
add c5	mylar	nylon	3.1167	2.5593	75	1.22	0.2271
add c5	mylar	poly	9.7833	2.5593	75	3.82	0.0003
add c5	nylon	poly	6.6667	2.5593	75	2.60	0.0111

The p-values given are for the LSD mean comparisons. We're doing a lot of comparisons!

We can try to adjust for multiplicity by using the Tukey adjustment:

```
lsmeans polymer*add/slicediff=(polymer add) cl adjust=tukey;
```

Simple Effect Comparisons of polymer*add Least Squares Means By add
Adjustment for Multiple Comparisons: Tukey

Simple Effect Level	polymer	_polymer	Estimate	Standard Error	DF	t Value	Pr > |t|	Adj P	Alpha
add c1	mylar	nylon	6.6833	2.5593	75	2.61	0.0109	0.0290	0.05
add c1	mylar	poly	3.3500	2.5593	75	1.31	0.1945	0.3947	0.05
add c1	nylon	poly	-3.3333	2.5593	75	-1.30	0.1968	0.3983	0.05
add c2	mylar	nylon	6.6167	2.5593	75	2.59	0.0117	0.0310	0.05
add c2	mylar	poly	1.9333	2.5593	75	0.76	0.4524	0.7313	0.05
add c2	nylon	poly	-4.6833	2.5593	75	-1.83	0.0712	0.1669	0.05
add c3	mylar	nylon	7.6833	2.5593	75	3.00	0.0036	0.0101	0.05
add c3	mylar	poly	5.8167	2.5593	75	2.27	0.0259	0.0659	0.05
add c3	nylon	poly	-1.8667	2.5593	75	-0.73	0.4681	0.7469	0.05
add c4	mylar	nylon	12.4667	2.5593	75	4.87	<.0001	<.0001	0.05
add c4	mylar	poly	11.1333	2.5593	75	4.35	<.0001	0.0001	0.05
add c4	nylon	poly	-1.3333	2.5593	75	-0.52	0.6039	0.8614	0.05
add c5	mylar	nylon	3.1167	2.5593	75	1.22	0.2271	0.4465	0.05
add c5	mylar	poly	9.7833	2.5593	75	3.82	0.0003	0.0008	0.05
add c5	nylon	poly	6.6667	2.5593	75	2.60	0.0111	0.0295	0.05

Simple Effect Comparisons of polymer*add Least Squares Means By add
Adjustment for Multiple Comparisons: Tukey

Simple Effect Level	polymer	_polymer	Lower	Upper	Adj Lower	Adj Upper
add c1	mylar	nylon	1.5849	11.7817	0.5637	12.8029
add c1	mylar	poly	-1.7484	8.4484	-2.7696	9.4696
add c1	nylon	poly	-8.4317	1.7651	-9.4529	2.7863
add c2	mylar	nylon	1.5183	11.7151	0.4971	12.7363
add c2	mylar	poly	-3.1651	7.0317	-4.1863	8.0529
add c2	nylon	poly	-9.7817	0.4151	-10.8029	1.4363
add c3	mylar	nylon	2.5849	12.7817	1.5637	13.8029
add c3	mylar	poly	0.7183	10.9151	-0.3029	11.9363
add c3	nylon	poly	-6.9651	3.2317	-7.9863	4.2529
add c4	mylar	nylon	7.3683	17.5651	6.3471	18.5863
add c4	mylar	poly	6.0349	16.2317	5.0137	17.2529
add c4	nylon	poly	-6.4317	3.7651	-7.4529	4.7863
add c5	mylar	nylon	-1.9817	8.2151	-3.0029	9.2363
add c5	mylar	poly	4.6849	14.8817	3.6637	15.9029
add c5	nylon	poly	1.5683	11.7651	0.5471	12.7863

Even with the Tukey adjustment, we need to remember the total number of comparisons we're doing far exceeds the df for treatment. This means

A better way to try to keep the experiment-wise error rate reasonable is to look at only a small set of comparisons that are planned before carrying out the experiment. `contrast` and `estimate` statements (or `lsmestimate`) can be used to look at these pre-planned comparisons.

Planned Comparisons

In our example, we were not specific about the goals of the experiment, beyond considering these two factors. Typically, a researcher has more information about the specific questions of interest. Suppose nylon and polyethylene have been used by the company in the past, and they are interest in how mylar compares as a potentially new option. This leads to some specific hypotheses the company may want to test.

So, in general, the company may be interested in

But there are several ways to look at this comparison

-
-
-

Not all of these will be appropriate comparisons! It depends on whether or not interaction is present. If there is interaction, we have to use simple effects or a compromise. If there is no interaction, then we can use the main effects.

When we would use the compromise? When there are subsets of additives where the polymer does not interact with additive.

We'll now see how to we can use `estimate`, `contrast`, and `lsmestimate` statements to determine which of the comparisons above are important. The `estimate` statement is constructed based on the estimates of the effects in the model. The `lsmestimate` statement is constructed based on the treatment `lsmeans` (the cell means).

Here's a three step procedure to construct the `contrast` and `estimate` statements:

1. Write the linear combination you want to test or estimate in terms of the cell means, μ_{ij}
2. Convert means into model parameters
3. Gather like terms

This will give you the coefficients to use in the statements. We're going to start with `lsmestimate`, because this is where it really gets to shine.

Before determining the linear combinations, we need to know what order SAS has the `lsmeans`–this will tell us the order in which they go in the statements. The order is determined by the order the factors appear in the `class` statment (not the `model` statement) and then in alphanumeric order within each factor.

For example:

```
class polymer add;
model strength=polymer|add;
model strength=add|polymer;
```

will put the factors and levels in the same order.

We have three polymers with 5 additives, so the order of the coefficients would be

However, if we use

```
class add polymer;
model strength=polymer|add;
model strength=add|polymer;
```

we'd get a different order. Now, the order of the coefficients would be

Let's stick with `class polymer add` order. We want to find the coefficients that we need to compare mylar vs the average of nylon and polyethylene in additive 1.

We want to test

which maps to coefficients

Mc1	Mc2	Mc3	Mc4	Mc5	Nc1	Nc2	Nc3	Nc4	Nc5	Pc1	Pc2	Pc3	Pc4	Pc5

We can now add the `lsmestimate` statement to our program

```
lsmestimate polymer*add 'Mylar vs Nylon,Poly in Add c1' 1 0 0 0 0 -0.5 0 0 0 0 -0.5 0 0 0 0;
lsmestimate polymer*add 'Mylar vs Nylon,Poly in Add c1' 2 0 0 0 0 -1 0 0 0 0 -1 0 0 0 0/divisor=2;
```

We could additional statements to look at this same comparison with the other additives. Note that I'm not looking at this comparison in Additive c5, because we already saw there is a difference between nylon and polyethylene in c5 so it doesn't make sense to look at this.

```
proc glimmix data=ribbon;
  class polymer add;
  model strength=polymer|add;
  lsmestimate polymer*add
    'Mylar vs Nylon,Poly in Add c1' 2 0 0 0 0 -1 0 0 0 0 -1 0 0 0 0,
    'Mylar vs Nylon,Poly in Add c2' 0 2 0 0 0 0 -1 0 0 0 0 -1 0 0 0,
    'Mylar vs Nylon,Poly in Add c3' 0 0 2 0 0 0 0 -1 0 0 0 0 -1 0 0,
    'Mylar vs Nylon,Poly in Add c4' 0 0 0 2 0 0 0 0 -1 0 0 0 0 -1 0/divisor=2;
run;
```

Least Squares Means Estimates

Effect	Label	Estimate	Standard Error	DF	t Value	Pr > |t|
polymer*add	Mylar vs Nylon,Poly in Add c1	5.0167	2.2164	75	2.26	0.0265
polymer*add	Mylar vs Nylon,Poly in Add c2	4.2750	2.2164	75	1.93	0.0575
polymer*add	Mylar vs Nylon,Poly in Add c3	6.7500	2.2164	75	3.05	0.0032
polymer*add	Mylar vs Nylon,Poly in Add c4	11.8000	2.2164	75	5.32	<.0001

We see the estimates differ for the different additives, indicating a potential interaction between Mylar vs (Nylon & Poly) and Additive. We can investigate this further by comparing the simple effects. But for this, we need to use `estimate`.

Again the difference between `lsmestimate` and `estimate` is

Using `estimate`, we'll need to go through the 3 step method outlined above.

1. State the comparison in terms of the cell means

2. Convert cell means into treatment effects

3. Gather like terms

So, our estimate statement will only involve the polymer and polymer*additive estimates.

```
proc glimmix data=ribbon;
  class polymer add;
  model strength=polymer|add;
  estimate 'Mylar vs Nylon,Poly in Add c1'
    polymer 2 -1 -1 polymer*add 2 0 0 0 0 -1 0 0 0 0 -1 0 0 0 0/divisor=2;
run;
```

which gives

Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > |t|
Mylar vs Nylon,Poly in Add c1	5.0167	2.2164	75	2.26	0.0265

We could also look at all four comparisons at once like we did with `lsmestimate` and we could also add the `adjust=` option to control multiplicity.

```
proc glimmix data=ribbon;
  class polymer add;
  model strength=polymer|add;
  estimate 'Mylar vs Nylon,Poly in Add c1'
      polymer 2 -1 -1 polymer*add 2 0 0 0 0 -1 0 0 0 0 -1 0 0 0 0,
   'Mylar vs Nylon,Poly in Add c2'
      polymer 2 -1 -1 polymer*add 0 2 0 0 0 0 -1 0 0 0 0 -1 0 0 0,
   'Mylar vs Nylon,Poly in Add c3'
      polymer 2 -1 -1 polymer*add 0 0 2 0 0 0 0 -1 0 0 0 0 -1 0 0,
   'Mylar vs Nylon,Poly in Add c4'
      polymer 2 -1 -1 polymer*add 0 0 0 2 0 0 0 0 -1 0 0 0 0 -1 0/divisor=2;
run;
```

These give us the same results we saw with `lsmestimate`! So why bother going through all of this mess if we could get them more easily with `lsmestimate`? So now that we know how to build the `estimate` statements we can do so to explore contrasts to see if there is significant interaction. We can see there is from the Type III tests, but interaction now has 8 df–there's a lot more going on with interaction than there was in a 2×2 model, and we need to tease it apart. Right now, we're specifically interested in whether there is interaction between the mylar vs nylon/poly comparison and additive.

Let's write out the null hypothesis for this interaction.

Rather than a single equation for interaction like we saw in the 2×2, we actually have a set of 4 equations. This is a

How can we come up with the 4 equations to see what goes in the contrast statement?

We'll do the same thing we did in the three step process–write the means in terms of their treatment effects and gather like terms. Let's look at c1 vs c5:

So the main effects cancel out and the only term we need in our contrast statement is polymer*add. That would give us coefficients:

Mc1	Mc2	Mc3	Mc4	Mc5	Nc1	Nc2	Nc3	Nc4	Nc5	Pc1	Pc2	Pc3	Pc4	Pc5

and our contrast statement would look like this:

```
proc glimmix data=ribbon;
  class polymer add;
  model strength=polymer|add;
  contrast 'Mylar vs Nylon,Poly by Add interaction'
    polymer*add 1 0 0 0 -1  -0.5 0 0 0 0.5  -0.5 0 0 0 0.5,
    polymer*add 0 1 0 0 -1  0 -0.5 0 0 0.5  0 -0.5 0 0 0.5,
    polymer*add 0 0 1 0 -1  0 0 -0.5 0 0.5  0 0 -0.5 0 0.5,
    polymer*add 0 0 0 1 -1  0 0 0 -0.5 0.5  0 0 0 -0.5 0.5;
run;
```

which gives

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
polymer	2	75	24.25	<.0001
add	4	75	0.14	0.9648
polymer*add	8	75	2.38	0.0241

Contrasts

Label	Num DF	Den DF	F Value	Pr > F
Mylar vs Nylon,Poly by Add interaction	4	75	1.76	0.1450

So it looks like there is potentially interaction between this comparison of polymers and additive. If we wanted to present results for this comparison, we should do it separately for the 5 additives. If the structure of the factors lends itself to specific comparisons, we could also tease apart that 4 df interaction test.

Suppose we do have additional knowledge about the factor structures, and Additives c1 and c2 are similarly formulated as are Additives c3 and c4. We already know that polymers N and P are different in c5, so so we can't report M versus (N/P) for c5. We can further tease apart c1, c2, c3, and c4 and potentially report some more general effects. We'll consider (c1 and c2) and (c3 and c4), and break apart that 4 df interaction test into 4 separate 1 df tests.

1. H_0 : No interaction between M vs (N/P) and Additives (c1 and c2)

2. H_0 : No interaction between M vs (N/P) and Additives (c3 and c4)

At this point, if we were reporting our results, we'd stop. We can report M vs (N/P) averaged over (c1 and c2), but must report M vs (N/P) separately for the other 3 additives. Where do the other 2 df live?

Let's **suppose** (this is not true), that both contrast 1 and 2 had been non-significant. We could then move on to:

3. Could we combine (c1 and c2) and (c3 and c4)?

The remaining df is:

4. M vs (N/P) × [(c1/c2/c3/c4) vs c5]

3.5 Multifactor Experiments

So far, we've discussed factorial experiments with two factors, A and B, both of which have 2 or more levels, a and b. We've referred to this as an $a \times b$ factorial treatment design. We're still considering the completely randomized (CRD) experimental design. If we add a factor C (with c levels) to our $a \times b$ design, we have an $a \times b \times c$ factorial design. If all three factors have a levels, we can alternatively describe this as an a^3 factorial. The total number of treatments is $a \times b \times c$.

Example: The following experiment was actually carried out in an experimental design class at Arizona State University, and considered some of the many different ways to bake brownies. The purpose of the experiment was to determine how the pan material, brand of brownie mix, and the stirring method affect the scrumptiousness of brownies. The factors and levels were:

Factor	Levels
A= Pan Material	Glass, Aluminum
B= Stirring Method	Spoon, Mixer
C= Brand of Mix	Expensive, Cheap

This is a $2 \times 2 \times 2 = 2^3$ experiment. There are 8 treatment combinations:

	Factors		
Treatment	Pan	Stirring	Mix
1	Glass	Spoon	Expensive
2	Aluminum	Spoon	Expensive
3	Glass	Mixer	Expensive
4	Aluminum	Mixer	Expensive
5	Glass	Spoon	Cheap
6	Aluminum	Spoon	Cheap
7	Glass	Mixer	Cheap
8	Aluminum	Mixer	Cheap

The response variable was scrumptiousness, a subjective measured derived from a questionnaire given to the subjects who sampled each batch of brownies. The questionnaire included questions related to taste, appearance, consistency, aroma, etc. Eight batches of each treatment combination were rated, for a total of $8 \times 8 = 64$ experimental units.

Before we look at the data, let's consider how we must adapt our notation, model, and ANOVA table for an additional factor.

First, consider the cell mean μ_{ijk}

Our response now have 4 subscripts: y_{ijkl}

The treatment effects model is:

And a sketch of the ANOVA table:

It gets a little more complicated to make a table of cell means

	Stirring	Method	Stirring	Method
	Spoon	Spoon	Mixer	Mixer
	Pan	Material	Pan	Material
Mix Brand	Glass	Aluminum	Glass	Aluminum
Expensive	μ_{111}	μ_{211}	μ_{121}	μ_{221}
Cheap	μ_{112}	μ_{212}	μ_{122}	μ_{222}

The data are in the file `brownie.sas`. Let's put in the observed cell means

	Stirring	Method	Stirring	Method
	Spoon	Spoon	Mixer	Mixer
	Pan	Material	Pan	Material
Mix Brand	Glass	Aluminum	Glass	Aluminum
Expensive				
Cheap				

We can also think about main effect marginal means

and treatment combination marginal means

3.5.1 Simple Effects and Interaction in a Multifactor Experiment

We can again look at simple effects but this time, instead of looking at the effect of one factor with the other held constant, simple effects describe the effect of one factor with the other two (or more) held constant.

Example: We can consider the simple effect of pan material, given stirring with a spoon and expensive mix:

We can use simple effects to determine if there is **conditional interaction**. Conditional interaction means

This is denoted as $A \times B|C_k$ or $A \times B|C$. Determining whether $A \times B|C_k$ is nonzero is the same as asking

We can also construct interaction plots to explore this. For the brownie example, suppose we are interested in the interaction between pan material and stirring method, given the cheap brand.

We can also examine the conditional interaction between pan material and stirring method, given the expensive mix.

These two plots together can be examined to see if there is a three-way interaction present. Three-way interaction occurs when

In this case, it does not appear there is significant three-way interaction. What would three-way interaction look like?

3.5.2 Analysis for Multifactor Experiments

We'll now explore the analysis of a three-way factorial treatment design by looking first at the analysis of the brownie data, and then two additional example data sets with increasingly complex interactions among the three factors.

The basic steps in analyzing data arising from a three-factor design are:

1. Include all main effects, two-way, and three-way interaction terms in the model and test the three-way interaction.
2. If the three-way interaction term is not significant, test the two-way interactions. If the three-way interaction is significant, test the two-way interactions at each level of the third factor (conditional interactions).
3. Depending on the results of (1) and (2),
    - Test main effects (for those main effects free of any interaction)
    - Test simple effects of one factor conditional on the second but free of the third (when there are two-way interactions, but no three-way)
    - Test simple effects of one factor conditional on combinations of the other two (when there is three-way interaction)

This can be conducted using a two-stage approach in `SAS`.

Stage 1: Run the basic analysis including all main effects, two-way interactions, and three-way interaction to see what is significant.

```
proc glimmix;
  class a b c;
  model y=a|b|c; *or model y=a b c a*b a*c b*c a*b*c;
run;
```

Stage 2: Add needed two-way interaction and/or main effect contrasts, `lsmeans`, `slice`, and `slicediff` statements to test various hypotheses and estimate various differences, depending on which thee-way or two-way interactions are significant (and therefore which hypotheses are legitimate to test).

3.5.2.1 Brownie Example

Stage 1: We'll start using the basic program

```
proc glimmix data=brownies;
  class pan stir mix;
  model y=pan|stir|mix;
run;
```

which gives

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
pan	1	56	11.95	0.0010
stir	1	56	2.99	0.0894
pan*stir	1	56	0.01	0.9194
mix	1	56	0.01	0.9194
pan*mix	1	56	0.26	0.6132
stir*mix	1	56	0.17	0.6858
pan*stir*mix	1	56	0.04	0.8396

Stage 2: Based on the output above, we only need to further investigate the (main) effects of pan material and (potentially) stirring method. We can refit the model without the interactions and add an `lsmeans` statements

```
proc glimmix data=brownies;
  class pan stir mix;
  model y=pan stir mix;
  lsmeans pan stir/diff;
run;
```

which gives

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
pan	1	60	12.70	0.0007
stir	1	60	3.17	0.0798
mix	1	60	0.01	0.9169

pan Least Squares Means

pan	Estimate	Standard Error	DF	t Value	Pr > |t|
aluminum	12.6250	0.4217	60	29.94	<.0001
glass	10.5000	0.4217	60	24.90	<.0001

Differences of pan Least Squares Means

pan	_pan	Estimate	Standard Error	DF	t Value	Pr > |t|
aluminum	glass	2.1250	0.5963	60	3.56	0.0007

stir Least Squares Means

stir	Estimate	Standard Error	DF	t Value	Pr > |t|
mixer	12.0938	0.4217	60	28.68	<.0001
spoon	11.0313	0.4217	60	26.16	<.0001

Differences of stir Least Squares Means

stir	_stir	Estimate	Standard Error	DF	t Value	Pr > |t|
mixer	spoon	1.0625	0.5963	60	1.78	0.0798

3.5.2.2 Data Set #2 Example

The data are in the file `multifactor examples.sas`.

Let's first explore the cell means and sketch the interaction plots.

a*b*c Least Squares Means

a	b	c	Estimate	Standard Error	DF	t Value	Pr > |t|
1	1	1	221.83	4.8769	16	45.49	<.0001
1	1	2	243.50	4.8769	16	49.93	<.0001
1	2	1	239.77	4.8769	16	49.16	<.0001
1	2	2	247.73	4.8769	16	50.80	<.0001
2	1	1	254.87	4.8769	16	52.26	<.0001
2	1	2	270.40	4.8769	16	55.44	<.0001
2	2	1	232.07	4.8769	16	47.58	<.0001
2	2	2	245.67	4.8769	16	50.37	<.0001

Let's plot B $\times$ C for each level of A.

Stage 1: We'll start with the basic program

```
proc glimmix data=example2;
  class a b c;
  model y=a|b|c;
run;
```

```
           Fit Statistics

Pearson Chi-Square / DF        71.35
```

```
     Type III Tests of Fixed Effects
               Num      Den
Effect          DF       DF    F Value    Pr > F

a                1       16      13.23    0.0022
b                1       16       3.38    0.0846
a*b              1       16      25.53    0.0001
c                1       16      18.15    0.0006
a*c              1       16       0.00    0.9715
b*c              1       16       1.28    0.2738
a*b*c            1       16       0.73    0.4062
```

Based on this, we can look at the main effects of C and the simple effects of A given B (averaged over C) and B given A (averaged over C).

Stage 2:

```
proc glimmix data=example2;
  class a b c;
  model y=a b a*b c;
  lsmeans c/diff;
  lsmeans a*b/slicediff=a slicediff=b;
run;
```

Fit Statistics

Pearson Chi-Square / DF	67.65

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
a	1	19	13.95	0.0014
b	1	19	3.57	0.0743
a*b	1	19	26.93	<.0001
c	1	19	19.14	0.0003

c Least Squares Means

c	Estimate	Standard Error	DF	t Value	Pr > |t|
1	237.13	2.3743	19	99.87	<.0001
2	251.83	2.3743	19	106.06	<.0001

Differences of c Least Squares Means

c	_c	Estimate	Standard Error	DF	t Value	Pr > |t|
1	2	-14.6917	3.3578	19	-4.38	0.0003

a*b Least Squares Means

a	b	Estimate	Standard Error	DF	t Value	Pr > |t|
1	1	232.67	3.3578	19	69.29	<.0001
1	2	243.75	3.3578	19	72.59	<.0001
2	1	262.63	3.3578	19	78.22	<.0001
2	2	238.87	3.3578	19	71.14	<.0001

Simple Effect Comparisons of a*b Least Squares Means By a

Simple Effect Level	b	_b	Estimate	Standard Error	DF	t Value	Pr > |t|
a 1	1	2	-11.0833	4.7486	19	-2.33	0.0307
a 2	1	2	23.7667	4.7486	19	5.00	<.0001

Simple Effect Comparisons of a*b Least Squares Means By b

Simple Effect Level	a	_a	Estimate	Standard Error	DF	t Value	Pr > |t|
b 1	1	2	-29.9667	4.7486	19	-6.31	<.0001
b 2	1	2	4.8833	4.7486	19	1.03	0.3167

We can also construct interaction plots to visualize the interaction between A and B:

```
proc glimmix data=example2;
  class a b c;
  model y=a b a*b c;
  lsmeans a*b/plots=meanplot(sliceby=b join);
run;
```

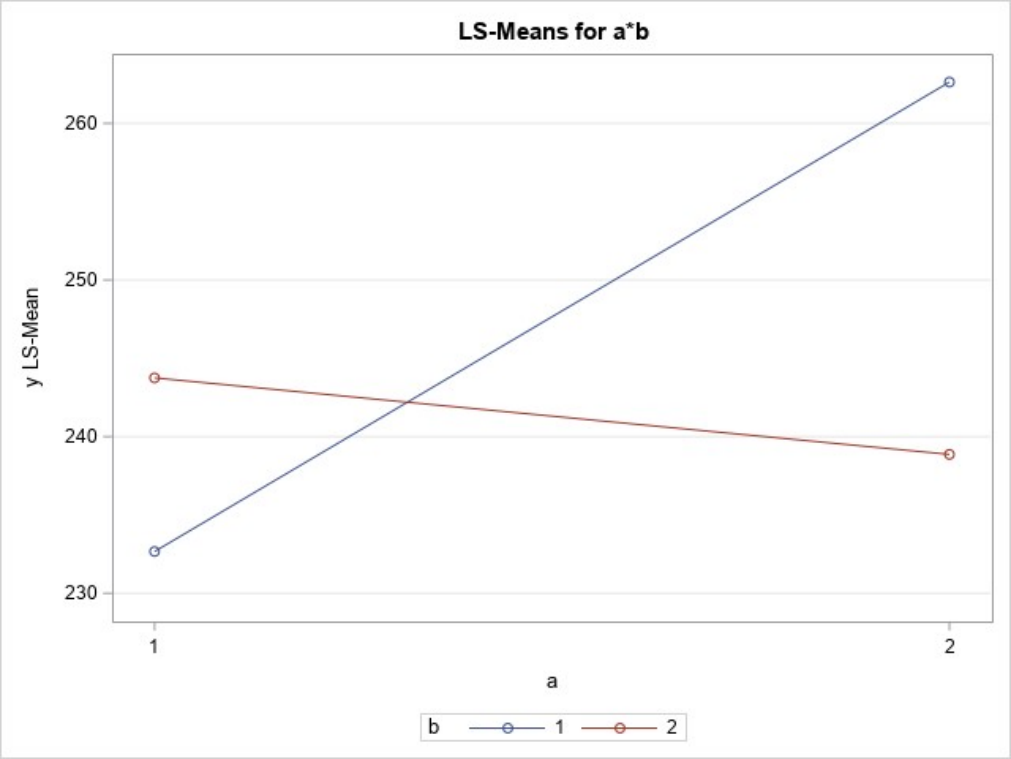


Figure 3.3: Interaction plot for A x B, averaged over C

Let's verify that this really is averaging over C.

3.5.2.3 Data Set #3 Example

The data are in the file `multifactor examples.sas`.

Let's first explore the cell means and sketch the interaction plots.

a*b*c Least Squares Means

a	b	c	Estimate	Standard Error	DF	t Value	Pr > |t|
1	1	1	223.13	6.2094	16	35.93	<.0001
1	1	2	234.53	6.2094	16	37.77	<.0001
1	2	1	247.77	6.2094	16	39.90	<.0001
1	2	2	259.57	6.2094	16	41.80	<.0001
2	1	1	245.67	6.2094	16	39.56	<.0001
2	1	2	275.30	6.2094	16	44.34	<.0001
2	2	1	265.33	6.2094	16	42.73	<.0001
2	2	2	249.90	6.2094	16	40.25	<.0001

Let's plot B $\times$ C for each level of A.

Stage 1: We'll start with the basic program

```
proc glimmix data=example2;
class a b c;
model y=a|b|c;
run;
```

```
              Fit Statistics
    Pearson Chi-Square / DF       115.67
```

```
     Type III Tests of Fixed Effects
               Num      Den
Effect          DF       DF    F Value    Pr > F

a                1       16      16.43    0.0009
b                1       16       6.26    0.0236
a*b              1       16       9.95    0.0061
c                1       16       4.53    0.0491
a*c              1       16       0.26    0.6153
b*c              1       16       6.47    0.0217
a*b*c            1       16       6.70    0.0198
```

So our suspicion of three-way interaction is verified. Let's explore this a bit more using plots. We've seen the B $\times$ C for each level of A. To illustrate how to use SAS to construct the interaction plots, let's consider A $\times$ B for each level of C.

```
proc glimmix data=example3;
  class a b c;
  model y=a|b|c;
  lsmeans a*b*c/plot=meanplot (sliceby=b plotby=c join);
run;
```

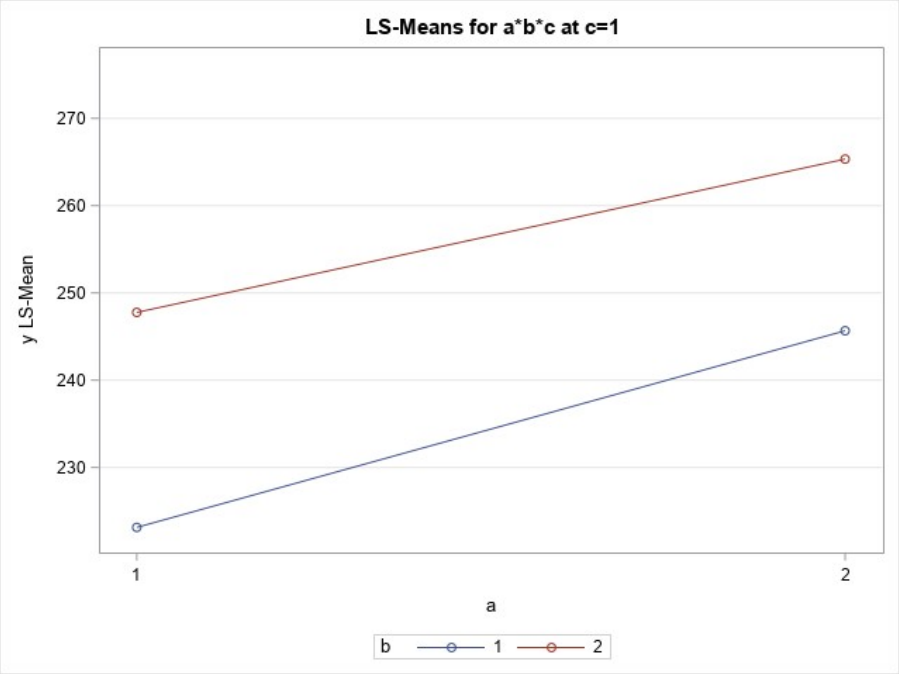


Figure 3.4: Interaction plot for A x B at level C=1

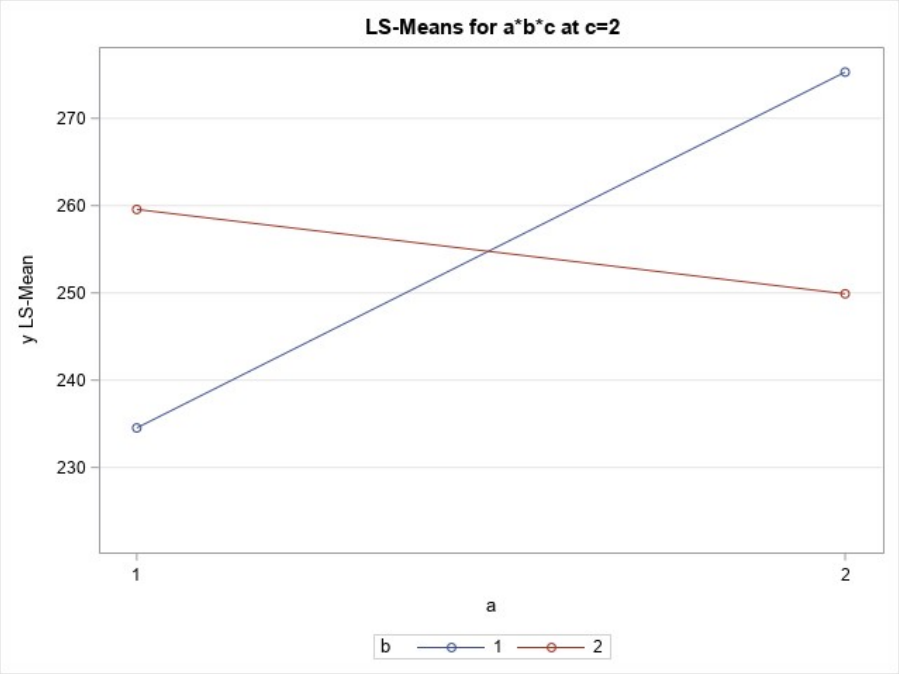


Figure 3.5: Interaction plot for A x B at level C=2.

So it looks like there is no interaction between A and B at C=1.

Stage 2: We can formally test the interaction between A and B at C=1, and proceed based on the results. We'll use the same approach we did in the last section:

1. Write the linear combination you want to test or estimate in terms of the cell means, μ_{ij}
2. Convert means into model parameters
3. Gather like terms

In this case, we're interested in $\mu_{111} - \mu_{121} - \mu_{211} + \mu_{221}$.

This leads to the `contrast` statement

```
contrast 'a*b at c=1' a*b 1 -1 -1 1 a*b*c 1 0 -1 0 -1 0 1 0;
```

```
                    Contrasts
                Num       Den
Label            DF        DF    F Value    Pr > F

a*b at c=1        1        16       0.16    0.6945
```

As suspected, there is no A × B interaction at C=1. So, we can examine the 'main' effects of A and B at C=1.

```
proc glimmix data=example3;
  class a b c;
  model y=a|b|c;
  lsmeans a*c/slicediff=c;
  lsmeans b*c/slicediff=c;
run;
```

```
                     a*c Least Squares Means
                       Standard
a    c    Estimate        Error       DF    t Value    Pr > |t|

1    1      235.45       4.3907       16      53.62      <.0001
1    2      247.05       4.3907       16      56.27      <.0001
2    1      255.50       4.3907       16      58.19      <.0001
2    2      262.60       4.3907       16      59.81      <.0001

         Simple Effect Comparisons of a*c Least Squares Means By c
Simple
Effect                               Standard
Level     a    _a    Estimate        Error       DF    t Value    Pr > |t|

c 1       1    2     -20.0500       6.2094       16      -3.23      0.0052
c 2       1    2     -15.5500       6.2094       16      -2.50      0.0235
```

b*c Least Squares Means

b	c	Estimate	Standard Error	DF	t Value	Pr > |t|
1	1	234.40	4.3907	16	53.39	<.0001
1	2	254.92	4.3907	16	58.06	<.0001
2	1	256.55	4.3907	16	58.43	<.0001
2	2	254.73	4.3907	16	58.02	<.0001

Simple Effect Comparisons of b*c Least Squares Means By c

Simple Effect Level	b	_b	Estimate	Standard Error	DF	t Value	Pr > |t|
c 1	1	2	-22.1500	6.2094	16	-3.57	0.0026
c 2	1	2	0.1833	6.2094	16	0.03	0.9768

However, it does look like there is an interaction between A and B if C=2. We'll check this formally. This time, we're interested in $\mu_{112} - \mu_{122} - \mu_{212} + \mu_{222}$.

Practice: Find the coefficients necessary to test this contrast.

```
                    Contrasts
                  Num       Den
Label              DF        DF    F Value    Pr > F

a*b at c=2          1        16      16.49    0.0009
```

So, the A × B interaction is significant. So, we must look at simple effects.

```
lsmeans a*b*c/slicediff=b*c slicediff=a*c;
```

```
                      a*b*c Least Squares Means
                                Standard
a    b    c    Estimate          Error       DF    t Value    Pr > |t|

1    1    1      223.13         6.2094       16      35.93      <.0001
1    1    2      234.53         6.2094       16      37.77      <.0001
1    2    1      247.77         6.2094       16      39.90      <.0001
1    2    2      259.57         6.2094       16      41.80      <.0001
2    1    1      245.67         6.2094       16      39.56      <.0001
2    1    2      275.30         6.2094       16      44.34      <.0001
2    2    1      265.33         6.2094       16      42.73      <.0001
2    2    2      249.90         6.2094       16      40.25      <.0001


          Simple Effect Comparisons of a*b*c Least Squares Means By b*c
Simple
Effect                                Standard
Level       a    _a    Estimate          Error       DF    t Value    Pr > |t|

b*c 1 1     1    2     -22.5333         8.7815       16      -2.57      0.0207
b*c 1 2     1    2     -40.7667         8.7815       16      -4.64      0.0003
b*c 2 1     1    2     -17.5667         8.7815       16      -2.00      0.0627
b*c 2 2     1    2       9.6667         8.7815       16       1.10      0.2873

          Simple Effect Comparisons of a*b*c Least Squares Means By a*c

Simple
Effect                                Standard
Level       b    _b    Estimate          Error       DF    t Value    Pr > |t|

a*c 1 1     1    2     -24.6333         8.7815       16      -2.81      0.0127
a*c 1 2     1    2     -25.0333         8.7815       16      -2.85      0.0116
a*c 2 1     1    2     -19.6667         8.7815       16      -2.24      0.0397
a*c 2 2     1    2      25.4000         8.7815       16       2.89      0.0106
```

So, how would we write an overall summary of the results of Example 3? We explored:

- 'Main' effects of A and B at C=1
- Simple effects of A and B at C=2

4 Block Designs

At this point in the course, we make a distinct change of focus. Up to now, we've concentrated on analyzing data coming from various **treatment** designs. We've considered multiple flavors of one-way designs (unstructured, control vs others, regression, other structure) and factorials (2 factors, 3 factors, could have more than 3). In all cases though, we've been using the same **experimental** design: the completely randomized design (CRD). Now, we change our focus to other experimental designs. The treatment designs will be those we've seen before, and we'll continue to analyze treatment effects using methods we've already discussed.

With a shift to experimental designs, we'll be considering

This also means

We're going to start with the simplest experimental design (besides the CRD): the **randomized complete block design** (RCBD) and then move on to other block designs.

Consider an experiment in which we are interested in comparing six different lab activities for teaching the central limit theorem. Based on a power analysis, we believe four replications per treatment is sufficient, and so we need a total of 24 lab teams. We've got two options:

-
-

Which do you pick? Why? What are pros and cons of each?

Suppose you decide to use teams in different classes (or you don't have a choice). How will you assign treatments to teams?

Suppose we allocate treatments to teams completely at random (CRD), and by chance four out of six teams in one class are assigned to treatment 1. Would you be okay with this?

One of the main problems with the CRD is a possible 'conditional' bias. That is, treatment assignment is not balanced relative to any systematic variation/gradient. In this experiment, the gradient is

When this happens, treatment effect is confounded with gradient. Is any effect we observe really due to treatment, or is it due to the effect of the class? The other problem with the CRD is variance inflation. Suppose there is a gradient among the experimental units, with response increasing as you go up a gradient:

Even if all the same treatment is applied throughout, the variance among the experimental units (the residuals) will be composed of two quantities:

This means it will appear larger than it actually is. The most common solution to these problems of confounding and variance inflation is **blocking**.

The idea of blocking is:

Blocking allows us to reconcile two somewhat opposing aims of experimental design.

-
-

In summary, the general idea of blocking is to organize experimental units into groups that are as uniform as possible. We want to

Blocks usually represent naturally occurring differences not related to treatments. If we block 'correctly' then the design accounts for block variation, and allows us to pull it out and isolate the usual random error due to experimental units. If we block 'incorrectly' then we get a weaker experiment.

How Do We Block?

There are two basic steps in blocking an experiment:

1. Organize the experimental units into subsets (blocks) according to gradient
2. Restrict the randomization so that each treatment is assigned to one or fewer (zero) experimental units in each block.

4.1 The Randomized Complete Block Design

The simplest block design is the **randomized complete block design** (RCBD). In this design

In the RCBD, we carry out the two steps referenced on the previous page as

1. Divide the experimental units into blocks of homogeneous units.
2. Randomly assign treatments to units within blocks, using a separate randomization for each block. Every treatment will appear in every block.

Again, we carry out step 1 with the goal

The model for the RCBD helps point out some considerations for choosing blocks. The model (assuming a one-way treatment design) is:

The ANOVA table looks like

Example: An experiment was carried out to evaluate the effect of elevated CO_2 on rice grain yield. Four blocks of 2 rice paddies each (each block owned by a different farmer, who used different fertilizer regimes and management practices over the years) are available for the experiment. In each paddy there is a 12 m diameter circular plot. In one plot in each block there is a ring of tubing around the plot emitting CO_2 at a rate of 300 ppm above ambient level. In the other plot, no CO_2 is emitted. The grain yield is measured at 3 locations in each plot at the end of the season, and the response is the average of the 3 locations.

What is the experimental unit here?

What does the assumption of no block $\times$ treatment interaction mean in this example?

We can check it with an interaction plot. Here are the means for each plot

Block	Ambient CO_2	Elevated CO_2
1	6.21	6.41
2	6.25	6.42
3	6.10	6.26
4	6.14	6.30

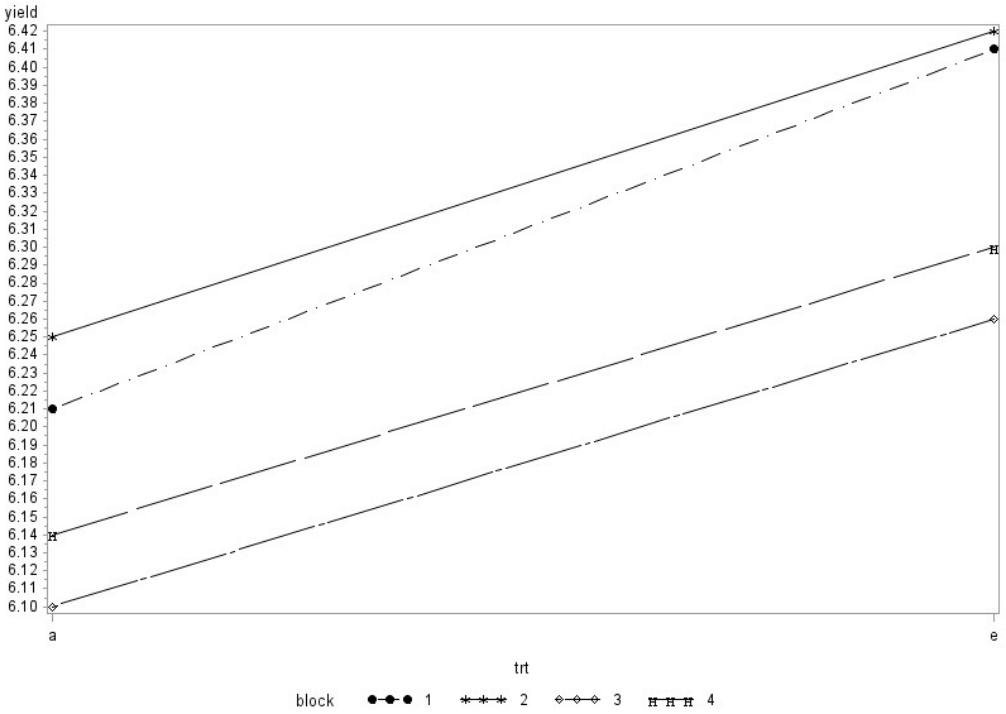


Figure 4.1: Interaction plot for treatment x block.

If you had been presented with this data in STAT 102 (or STAT 318), how would you have analyzed it?

Block	Ambient CO_2	Elevated CO_2
1	6.21	6.41
2	6.25	6.42
3	6.10	6.26
4	6.14	6.30

4.1.1 Selecting Blocks

Remember that the RCBD is an **experimental** design, not a treatment design. It can be used with any treatment design. So, we might see RCBD layouts that look like:

Block 1	Block 2	Block 3
Control	Trt2	Trt4
Trt2	Control	Trt3
Trt3	Trt4	Trt2
Trt4	Trt3	Control

Block 1	Block 2	Block 3
20	60	80
60	40	20
80	80	40
40	20	60

Block 1	Block 2	Block 3
A1 & B1	A1 & B1	A2 & B2
A2 & B1	A2 & B1	A1 & B1
A2 & B2	A1 & B2	A1 & B2
A1 & B2	A2 & B2	A2 & B2

When you write a report, both the treatment design and the experimental design need to be described in the methods section.

Tips for Choosing Blocks:

- We want to maximize differences between blocks and minimize differences within blocks

- Block size should not be excessively large

- Keep in the mind the no block $\times$ treatment interaction assumption

Common Criteria for Blocking:

- gradients that occur in the field, in greenhouses, in growth chambers
- weight groups in animal experimentation, litters, cage positions in a room
- occasion (day, month, year)
- location (barn, different fields, different rooms, different states)
- subjects (each subject serves as their own control)

4.1.2 RCBD Model and Analysis

Let y_{ij} be

Earlier we stated the model

We do have another choice to make. We can consider the block effect to either be a **fixed effect** or a **random effect**.

Our choice will have implications in the ANOVA table (in the Expected Mean Squares) and in the standard errors of the cell means.

To estimate the difference between two treatment means ($\mu_{i\cdot} - \mu_{i'\cdot}$), we use $\overline{y}_{i\cdot} - \overline{y}_{i'\cdot}$. To figure out the variance (or estimate of the variance), let's look at what $\overline{y}_{i\cdot}$ is actually estimating:

Now let's explore the variance of this quantity.

Now let's consider $\overline{y}_{i\cdot} - \overline{y}_{i'\cdot}$:

and its variance

If we want to construct confidence intervals for treatment means or differences, they'll have the form

where the standard error is

For example, we use MSE as our estimate of σ^2, so confidence intervals for the difference between two means is

RCBDs in SAS

We can still use `PROC GLIMMIX` to fit the model if our experimental design is the RCBD. The basic program for **fixed blocks** is

```
proc glimmix data=dataset;
  class block trt;
  model y = block trt;
run;
```

Note:

The basic program for **random blocks** is

```
proc glimmix data=dataset;
  class block trt;
  model y = trt;
  random block;
run;
```

Note:

Example: This experiment is looking at the emergence rate of soybean seeds treated with four different chemical treatments and a control.

Treatment Number	Treatment Name
1	Control
2	Arasan
3	Spergon
4	Semesan
5	Fermate

Experimental Layout: The field is located on a slope, and blocks are formed based on elevation. There are five plots at each elevation, and five blocks.

Treatment Design:

100 seeds were planted in each plot, and the response is the number of plants that emerge out of the 100.

Model:

Analysis with Blocks Fixed:

If we assume blocks are fixed, we use the code

```
proc glimmix data=seeds;
  class block chem;
  model emerge=block chem;
run;
```

which gives

```
               Fit Statistics

    Pearson Chi-Square / DF         5.41

         Type III Tests of Fixed Effects
                   Num      Den
    Effect          DF       DF    F Value    Pr > F

    block            4       16       2.30    0.1032
    chem             4       16       3.87    0.0219
```

The follow-up analyses don't change from what we've done so far. In this case, the treatment design is one-way treatment-versus-control, so comparing all treatments to the control is appropriate and we can use the Dunnett adjustment.

```
    lsmeans chem/diff=control('Control') adjust=dunnett;

                      Chem Least Squares Means

                         Standard
    Chem       Estimate       Error       DF    t Value    Pr > |t|

    Arasan      93.8000      1.0402       16      90.18      <.0001
    Control     89.2000      1.0402       16      85.75      <.0001
    Fermate     94.2000      1.0402       16      90.56      <.0001
    Semesan     93.4000      1.0402       16      89.79      <.0001
    Spergon     91.8000      1.0402       16      88.25      <.0001
```

Differences of Chem Least Squares Means
Adjustment for Multiple Comparisons: Dunnett

Chem	_Chem	Estimate	Standard Error	DF	t Value	Pr > |t|	Adj P
Arasan	Control	4.6000	1.4711	16	3.13	0.0065	0.0218
Fermate	Control	5.0000	1.4711	16	3.40	0.0037	0.0125
Semesan	Control	4.2000	1.4711	16	2.86	0.0115	0.0375
Spergon	Control	2.6000	1.4711	16	1.77	0.0962	0.2680

Analysis with Blocks Random:

If we assume blocks are random, we use the code

```
proc glimmix data=seeds;
  class block chem;
  model emerge=chem;
  random block;
  lsmeans chem/diff=control('Control') adjust=dunnett;
run;
```

which gives

Fit Statistics

Gener. Chi-Square / DF	5.41

Covariance Parameter Estimates

Cov Parm	Estimate	Standard Error
block	1.4100	1.8032
Residual	5.4100	1.9127

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Chem	4	16	3.87	0.0219

Chem Least Squares Means

Chem	Estimate	Standard Error	DF	t Value	Pr > |t|
Arasan	93.8000	1.1679	16	80.31	<.0001
Control	89.2000	1.1679	16	76.38	<.0001
Fermate	94.2000	1.1679	16	80.66	<.0001
Semesan	93.4000	1.1679	16	79.97	<.0001
Spergon	91.8000	1.1679	16	78.60	<.0001

Differences of Chem Least Squares Means
Adjustment for Multiple Comparisons: Dunnett-Hsu

Chem	_Chem	Estimate	Standard Error	DF	t Value	Pr > |t|	Adj P
Arasan	Control	4.6000	1.4711	16	3.13	0.0065	0.0218
Fermate	Control	5.0000	1.4711	16	3.40	0.0037	0.0125
Semesan	Control	4.2000	1.4711	16	2.86	0.0115	0.0375
Spergon	Control	2.6000	1.4711	16	1.77	0.0962	0.2680

The results are the same whether we used fixed blocks or random blocks. This is because our data are **balanced**– we had the same number of observations in each block, and all treatments appear in all blocks. If our data had not been balanced, the results would be different.

Let's go back to the ANOVA table to see where the estimate of σ_R^2 is coming from

We just said that the analysis based on the treatment design will not change. Let's consider an example with a one-way regression/quantitative factor levels treatment design.

Example: In this experiment, seven soil samples were taken from the Canary Islands. Each soil sample was split into five subsamples and a phosphate (Na_2,PO_4H) was added in amounts 0, 50, 100, 150, 200 ppm to these five subsamples (note even spacing). At the end of the experiment, the amount of exchangeable calcium was measured on each subsample. The seven samples serve as blocks. There are five levels of phosphate so

Let's first consider blocks fixed (it doesn't matter here, since the experiment is balanced). First we have to decide the order of the polynomial:

```
proc glimmix data=soil;
  class block;
  model calcium=block phos phos*phos phos*phos*phos phos*phos*phos*phos/htype=1;
run;
```

which gives output:

```
             Fit Statistics
Pearson Chi-Square / DF         0.13

             Type I Tests of Fixed Effects
                          Num      Den
Effect                     DF       DF    F Value    Pr > F

block                       6       24     109.80    <.0001
phos                        1       24       7.08    0.0137
phos*phos                   1       24       1.62    0.2147
phos*phos*phos              1       24       1.88    0.1829
phos*phos*phos*phos         1       24       0.48    0.4940
```

Now that we've determined the appropriate polynomial, we can get the fitted model:

```
proc glimmix data=soil;
  class block;
  model calcium=block phos/htype=1 solution;
run;
```

which gives

Parameter Estimates

Effect	block	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept		1.9737	0.1854	27	10.65	<.0001
block	1	1.4520	0.2312	27	6.28	<.0001
block	2	2.2280	0.2312	27	9.64	<.0001
block	3	0.8540	0.2312	27	3.69	0.0010
block	4	0.4420	0.2312	27	1.91	0.0666
block	5	5.0280	0.2312	27	21.75	<.0001
block	6	0.9960	0.2312	27	4.31	0.0002
block	7	0	.	.	.	.
phos		0.002283	0.000874	27	2.61	0.0145
Scale		0.1336	0.03637	.	.	.

So the fitted model is:

Do we really care about the effect Block 1 has on exchangeable calcium? Probably not. The seven soil samples were likely selected at random, so it really makes sense to treat blocks as random. Let's go back to the most complicated possible model and start over.

```
proc glimmix data=soil;
  class block;
  model calcium=phos phos*phos phos*phos*phos phos*phos*phos*phos/htype=1;
  random block;
run;
```

Covariance Parameter Estimates

Cov Parm	Estimate	Standard Error
block	2.8049	1.6343
Residual	0.1289	0.03721

```
             Type I Tests of Fixed Effects
                          Num      Den
Effect                     DF       DF    F Value    Pr > F

phos                        1       24       7.08    0.0137
phos*phos                   1       24       1.62    0.2147
phos*phos*phos              1       24       1.88    0.1829
phos*phos*phos*phos         1       24       0.48    0.4940
```

Now let's get the final model

```
proc glimmix data=soil;
  class block;
  model calcium=phos/htype=1 solution;
  random block;
run;
```

```
 Covariance Parameter Estimates
                         Standard
Cov Parm    Estimate       Error

block         2.8040      1.6343
Residual      0.1336     0.03637

                  Solutions for Fixed Effects
                        Standard
Effect       Estimate       Error       DF    t Value    Pr > |t|

Intercept      3.5451      0.6419        6       5.52      0.0015
phos         0.002283    0.000874       27       2.61      0.0145

        Type I Tests of Fixed Effects
              Num      Den
Effect         DF       DF    F Value    Pr > F

phos            1       27       6.83    0.0145
```

So, our fitted model is

Note that when we treated blocks as either fixed or random

- We ended up picking the same model
- The estimate for the slope was the same

Our goal was to describe the relationship between the treatment and the response. The blocks just give us a way to eliminate excess variance.

4.2 Did Blocking Work?

When we treated blocks as fixed effects, we get a p-value associated with block but it is completely meaningless because there is no valid hypothesis test for evaluating the effect of block. However, we can check the **efficiency** of the block design relative to a competing design.

Suppose we have t treatments and rt experimental units available for our experiment. We have two possible experimental designs:

-

-

The only difference between these is whether we group the experimental units into blocks before randomly assigning the treatments. Efficiency gives us a way to compare the variance of two competing designs–we want to select the design that gives us the smaller variance of estimated treatment differences.

- CRD:

- RCBD

So the choice between these two designs comes down to a comparison of σ^2_{CRD} and σ^2_{RCBD}. We can compare variances using a ratio called the **relative efficiency**.

If RE > 1

Once we've conducted an RCBD experiment we can look and see whether we did the right thing when we used blocks.

Note there is a difference in the error degrees of freedom between the CRD and RCBD which can have an impact. We can adjust for this difference by calculating the **adjusted relative efficiency**:

The correction factor is always less than 1, and usually won't make much difference. It can make a difference if the number of treatments and reps is small.

Example: Rice paddies In this example, there were 4 blocks and two treatments. We'll fit the model both with blocks and without.

- RCBD:

- CRD:

Example: Seed Emergence In this example, there were five blocks and five treatments. Again, we'll fit the model both with blocks and without.

- RCBD:

- CRD:

4.3 Blocking and Power

We've used several methods to determine power and sample size for CRDs: `proc power`, `proc glmpower`, and tricking `proc glimmix` into calculating it for us. We can use the third option to incorporate blocking into our power calculations. These often require a pilot data set so we can get some idea about how much variance we expect among the blocks.

Example: Researchers are interested in exploring the effect of three types of insecticide on the count of plant seedlings. They know there will be variability due to location in the greenhouse, so they plan on using an RCBD. They have pilot data from a previous study in the greenhouse, with 4 samples from 6 greenhouse benches selected at random. The data are in 'block power.sas.'

The researchers want to treat block as random, so we need estimates of both σ^2 and σ^2_R. We can get estimates of both of these based on our pilot data, as well as 75% confidence intervals for the variances.

We can use the following code:

```
proc glimmix noprofile data=seedlings;
  class plot;
  model count = ;
  random plot;
  covtest / cl(type=plr alpha=0.25);
run;
```

Covariance Parameter Estimates

Cov Parm	Estimate	Standard Error	Profile Likelihood 75% Confidence Bounds ------- Lower ------ Bound	Pr > Chisq	------- Upper ------ Bound	Pr > Chisq
plot	44.5903	29.1831	22.5497	0.2500	104.00	0.2500
Residual	6.1806	2.0602	4.3102	0.2500	9.3093	0.2500

Now we've got estimates for the variances, and can turn our attention to the differences we want to be able to detect. The researchers are interested in detecting differences of 1, 2, and 3 units. We're going to start with 6 blocks, and see if that's enough power. They want $\alpha = 0.05$ and are aiming for a power of 0.80.

We can modify the previous code we used to trick `glimmix` as follows:

```
*set up example data set with initial parameters;
data seedlingsa;
  nblock=6;
  input insecticide mu;
  do block=1 to nblock;
    output;
  end;
datalines;
1 79
2 82
3 80
;

proc glimmix data=seedlingsa;
  class block insecticide;
  model mu=insecticide;
  random block;
  parms(104)(9.3093)/hold=1,2;
  contrast '1 unit diff' insecticide 1 0 -1;
  contrast '2 unit diff' insecticide 0 1 -1;
  contrast '3 unit diff' insecticide 1 -1 0;
  lsmeans  insecticide/diff cl;
  ods output tests3=overallF1 contrasts=contrastF1;
run;
```

```
/* Power computation & Print step */
data power1;
  set overallF1 contrastF1;
  alpha=0.05;
  ncparm=numDF*Fvalue;
  Critical_F=Finv(1-alpha,numDF,denDF,0);
  Power=1-probF(Critical_F,numDF,denDF,ncparm);

proc print data=power1;
  var effect label numdf dendf alpha critical_F ncparm Power;
  title 'power - RCBD';
run;
```

which gives the following:

power - RCBD

Obs	Effect	Num Label	DF	DenDF	alpha	Critical_ F	ncparm	Power
1	insecticide		2	10	0.05	4.10282	3.00774	0.24820
2		1 unit diff	1	10	0.05	4.96460	0.32226	0.08092
3		2 unit diff	1	10	0.05	4.96460	1.28903	0.17731
4		3 unit diff	1	10	0.05	4.96460	2.90033	0.33778

So it doesn't look like 6 blocks is enough. Even with a 3 unit difference, the power is only 0.33778. Let's see what happens if we increase the block size to

- 20 blocks:

- 40 blocks:

- 150 blocks:

4.4 Row-Column Designs

Sometimes there are two blocking variables that can be used to reduce extraneous variability, rather than one factor. If it possible to block on both of the variables we may be able to increase precision while using the same number of experimental units as in a randomized complete block design.

Designs with two blocking criteria are called **row-column designs**. There are several different types of row-column designs,and we'll discuss one of them, specifically the Latin square.

Row-column designs tend to be situations in which there are two gradients:

The generic model for row-column designs is

```
observation = row + column + treatment + error
```

We assume there is no row×column interaction, no row×treatment interaction, and no column×treatment interaction.

Example: Suppose we want to compare five kinds of glue, and glue is affected by temperature and humidity. Temperature and humidity vary across days and over time throughout the day, so two possible blocking criteria are day and hour. A **Latin Square** for five treatments blocks on day and hour looks like this

Each day has a full set of five treatments and each hour has a full set of five treatments, so both day and hour could be considered blocks. The two types of blocking are superimposed on each other.

The Latin Square design requires

By using a Latin square arrangement, we are able to control for 5 rows $\times$ 5 columns $\times$ 5 treatments $=$ 125 combinations using only 25 experimental units. Because of this feature, Latin squares have the potential for large gains in efficiency without using more experimental material.

The model for a Latin square is

Just like with RCBDs we can consider rows and columns to be either fixed or random. The corresponding ANOVA table is

Example: Consider an experiment to study yield of four wheat varieties (A, B, C, D). There are gradients running parallel to both sides of the field

To account for both gradients, the experiment was conducted as a 4×4 Latin square. Yields are in kg per plot. The plot layout with the data:

	Column			
Row	1	2	3	4
1	10.5 (C)	7.7 (D)	12.0 (B)	13.2 (A)
2	11.1 (B)	12.0 (A)	10.3 (C)	7.5 (D)
3	5.8 (D)	12.2 (C)	11.2 (A)	13.7 (B)
4	11.6 (A)	12.3 (B)	5.9 (D)	10.2 (C)

In this example, the model is

and ANOVA table

The analysis in SAS is very similar to the RCBD:

```
proc glimmix data=wheat;
  class row col variety;
  model yield=variety;
  random row col;
run;
```

In this example, we are considering rows and columns as random effects. The output is:

```
  Covariance Parameter Estimates
                        Standard
Cov Parm    Estimate       Error

row          0.04958      0.1482
col           0.4533      0.4673
Residual      0.4533      0.2617

       Type III Tests of Fixed Effects
              Num      Den
Effect         DF       DF    F Value    Pr > F

variety         3        6      58.03    <.0001
```

It looks like there is a difference among the four varieties. We could follow up with whatever treatment design analysis is most appropriate (LSD, Tukey, maybe contrasts) just as before. A Latin square is a experimental design and like the RCBD can be paired with any treatment design.

The downside to the Latin square is that it can be restrictive, since # treatments = # columns = # rows. This means

-
-

The randomization of Latin squares is quite involved, and details will be left to a more advanced design course. There are other row-column designs as well which offer more flexibility than the Latin square. These include Latin rectangles and replicated Latin squares. We'll also leave those to a second design course.

4.5 Incomplete Block Designs

When we discussed RCBDs, we noted that we needed enough experimental units in each block to accommodate all treatments. However, in some situations, the number of treatments exceeds block size. This can happen because of the physical size of the block or shortages of experimental equipment or facilities. In instances like this, we can use randomized block designs in which every treatment is not present in every block. These designs are called **incomplete block designs**.

Example: A baker is testing different cookie recipes, and has 9 recipes to test. Due to oven size and baking time available, the baker can only test 6 recipes per day. However, they know it's important to block on day since oven temperature can vary day to to day. The RCDB would say use blocks of 1.5 days, and ignore the fact that days might not be homogeneous within a block. A better way is to use incomplete blocks of size six (one day). Consider the following set up:

	Treatments
Day 1	4, 5, 6, 7, 8, 9
2	2, 3, 5, 6, 8, 9
3	2, 3, 4, 6, 7, 8
4	2, 3, 4, 5, 7, 9
5	1, 3, 5, 6, 7, 8
6	1, 3, 4, 6, 7, 9
7	1, 3, 4, 5, 8, 9
8	1, 2, 5, 6, 8, 9
9	1, 2, 4, 6, 8, 9
10	1, 2, 4, 5, 7, 8
11	1, 2, 3, 7, 8, 9
12	1, 2, 3, 4, 5, 6

This design is a **balanced incomplete block** (BIB) design. It's balanced because

Standard notation for incomplete blocks is

-
-
-
-
-

These values are not independent, but must satisfy three conditions:

1.

2.

3.

Additionally,

BIBs do not exist for all combinations of blocks sizes, number of treatments, and number of replications. For example, with the cookie scenario, there is no BIB design with few than 8 replications. These restrictions means BIBs are not always feasible. However, when they are an option, they can very useful.

Example: Suppose we have $t = 6$ treatments and block size that can accommodate $k = 4$ observations per block.

So if BIBs are not always feasible, why are they still useful? BIB designs are **optimal** in that

If you are willing to give up balance, we can find a design with far fewer replicates that is nearly balanced and nearly as efficient as a BIB would be if it existed. **Partially balanced incomplete block** (PBIB) designs share many features with BIBs. In PBIBs

Not all incomplete block designs qualify as partially balanced. A PBIB must satisfy three criteria:

1.

2.

3.

Consider the following set-up:

Block		
1	**2**	**3**
1	2	3
4	5	6
2	3	1
5	6	4

The analysis of incomplete block designs is just like the analysis for an RCBD with missing data. We didn't discuss the implications of missing data on the RCBD, but missing data means that we don't get the same results for random and fixed blocks and we have to carefully think about which choice is most appropriate. Missing data also means that cell (arithmetic) means won't be the same as `lsmeans`, and we should be sure to look at `lsmeans`.

Example: An experiment was designed to test the wearing quality of $t = 7$ types of cloth. The machine used to test the wearing quality can only process four pieces of cloth at the same time. Machine run is considered the block. Here's a layout of the design

			Cloth	**Type**			
Block	A	B	C	D	E	F	G
1		627		248		563	252
2	344		233			442	226
3			251	211	160		297
4	337	537			195		300
5		520	278		199	595	
6	369			196	1985	606	
7	396	602	240	273			

Let's check that this is a BIB design.

The data are in `fabric.sas`. The model is

and the ANOVA table is

We can use the same code we did with RCBDs:

```
proc glimmix data=fabric;
  class block type;
  model y=type;
  random block;
  lsmeans type/pdiff adjust=tukey;
run;
```

It seems like the 7 machine runs we used could be thought of as a random sample of all runs we could ever do on the machine, so I'm treating blocks as random. The output is

```
  Covariance Parameter Estimates
                         Standard
Cov Parm    Estimate       Error

block         273.40      428.98
Residual     1471.43      537.29

        Type III Tests of Fixed Effects
              Num      Den
Effect         DF       DF    F Value    Pr > F

type            6       15      62.73    <.0001
```

type Least Squares Means

type	Estimate	Standard Error	DF	t Value	Pr > |t|
A	363.84	20.6074	15	17.66	<.0001
B	566.49	20.6074	15	27.49	<.0001
C	252.61	20.6074	15	12.26	<.0001
D	227.19	20.6074	15	11.02	<.0001
E	184.03	20.6074	15	8.93	<.0001
F	553.22	20.6074	15	26.85	<.0001
G	273.13	20.6074	15	13.25	<.0001

Differences of type Least Squares Means
Adjustment for Multiple Comparisons: Tukey-Kramer

type	_type	Estimate	Standard Error	DF	t Value	Pr > |t|	Adj P
A	B	-202.65	27.8771	15	-7.27	<.0001	<.0001
A	C	111.23	27.8771	15	3.99	0.0012	0.0160
A	D	136.65	27.8771	15	4.90	0.0002	0.0029
A	E	179.80	27.8771	15	6.45	<.0001	0.0002
A	F	-189.38	27.8771	15	-6.79	<.0001	<.0001
A	G	90.7093	27.8771	15	3.25	0.0053	0.0630
B	C	313.88	27.8771	15	11.26	<.0001	<.0001
B	D	339.30	27.8771	15	12.17	<.0001	<.0001
B	E	382.46	27.8771	15	13.72	<.0001	<.0001
B	F	13.2728	27.8771	15	0.48	0.6408	0.9988
B	G	293.36	27.8771	15	10.52	<.0001	<.0001
C	D	25.4242	27.8771	15	0.91	0.3762	0.9648
C	E	68.5788	27.8771	15	2.46	0.0265	0.2407
C	F	-300.61	27.8771	15	-10.78	<.0001	<.0001
C	G	-20.5159	27.8771	15	-0.74	0.4731	0.9878
D	E	43.1546	27.8771	15	1.55	0.1425	0.7142
D	F	-326.03	27.8771	15	-11.70	<.0001	<.0001
D	G	-45.9401	27.8771	15	-1.65	0.1201	0.6566
E	F	-369.18	27.8771	15	-13.24	<.0001	<.0001
E	G	-89.0946	27.8771	15	-3.20	0.0060	0.0700
F	G	280.09	27.8771	15	10.05	<.0001	<.0001

5 Split-Plot Designs

Suppose we are carrying out an experiment to explore the effect of cold storage conditions on food quality. There are two treatment factors: storage temperature (2, 4, 8 degrees C) and container type (sealed plastic, open plastic, glass, and aluminum foil). A food product will be placed in one of the containers and stored in a small temperature controlled freezer for fixed period of time. There are 12 small freezers available for the experiment. Each freezer can hold 4 containers.

What kind of treatment design is this?

How many treatment combinations are there?

What is the experimental unit?

What's the problem?

One thing we can try is randomly assign storage temperatures to the freezers and randomly assign food products to storage containers within each freezer. The storage temperature is applied to all food samples in a freezer.

Experimental unit for storage temperature:

The storage container is randomized separately, so the experimental unit for storage container:

So, we have two different sizes of experimental units for the two different factors in the experiment. Designs in which there are two sizes of experimental unit are called **split-plot designs**.

Split-plot designs are frequently used for factorial experiments, and may use CRD, RCBD, row-column, or incomplete block designs. The underlying principle of split-plots is

This means each whole unit becomes a block for the subunit treatments.

Consider the following design structure. We have Factor A (four levels) laid out in 3 randomized complete blocks.

Table 5.1: Factor A laid out in 3 randomized complete blocks

(a) Block 1

A4	A1	A2	A3

(b) Block 2

A2	A1	A4	A3

(c) Block 3

A1	A2	A4	A3

A second Factor B (2 levels), can be superimposed by dividing each A unit into two subunits and assigning the two B treatment levels to these subunits.

Table 5.2: Factor A laid out in 3 randomized complete blocks, with each A unit subdivided into two subunits.

(a) Block 1

A4	A1	A2	A3
b2	b2	b1	b2
b1	b1	b2	b1

(b) Block 2

A2	A1	A4	A3
b1	b2	b1	b1
b2	b1	b2	b2

(c) Block 3

A1	A2	A4	A3
b1	b2	b2	b1
b2	b1	b1	b2

Here the A units are the whole units and the B units are the subunits.

Note the randomization happens in two stages.

Each whole plot may be considered as a block for Factor B, but it is not a complete block for the full set of treatments.

Why might we use split-plot designs?

- Treatments associated with levels of one (or more) of the factors require larger amounts of experimental material than do treatments for other factors.

- An additional factor could be added to the experiment to increase its scope.

- From previous experiments or literature, we know that larger differences are expected among one factor than another.

- We need greater precision for comparisons among some factors than we do for comparison among others.

Big picture summary: variation among split-plot units is expected to be less than variation among whole plot units. This implies

As mentioned earlier, we can use split-plots in combination with any experimental design.

- **CRD:**

A1	A2	A1	A1	A2	A2
B2	B1	B1	B1	B2	B2
B1	B2	B2	B2	B1	B1

- **RCBD:**

Block	1	Block	2	Block	3
A1	A2	A2	A1	A2	A1
B2	B1	B1	B1	B2	B2
B1	B2	B2	B2	B1	B1

- **Latin Square:**

Row	Col				
	1		2		3	
1	A1		A2		A3	
	B1	B2	B2	B1	B2	B1
2	A2		A3		Aa	
	B1	B2	B1	B2	B1	B2
3	A3		A1		A2	
	B2	B1	B2	B1	B1	B2

- **BIB:**

Location	1	Location	2	Location	3
A1	A2	A1	A3	A2	A3
B2	B1	B1	B1	B2	B2
B1	B2	B2	B2	B1	B1

5.1 Model and Analysis of Variance

To determine the appropriate model, we'll start with the whole/main plot. Consider the following whole plot arrangement

A1	A2	A1	A1	A2	A2

Let's sketch the ANOVA table for this arrangement

Now let's look only at the subplot arrangement:

B1	B2	B2	B1	B2	B1
B2	B1	B1	B2	B1	B2

And sketch its ANOVA table

In order to get the main/whole unit analysis, the "block" is partitioned. The error from the subplot analysis can also be partitioned as each component times B, so we can examine the A $\times$ B interaction. If we put these two analyses together, we get (for the CRD):

If the main/whole plot is laid out in randomized complete blocks, the ANOVA table looks like this:

The model for the split-plot design incorporates both the whole plot and the sub plot. Without blocks, the model is

Tests of hypothesis in the split-plot will be constructed differently than we've seen before. To see this, let's review the ANOVA table (for the CRD) for a regular factorial:

Now let's compare this with the ANOVA table for the split-plot (still CRD) with the expected mean squares:

This leads to different F statistics.

For testing the null hypothesis $\mathrm{H}_0 : \alpha_1 = \alpha_2 = \cdots = \alpha_a = 0$

For testing the null hypothesis $\mathrm{H}_0 : \beta_1 = \beta_2 = \cdots = \beta_b = 0$

For testing the null hypothesis $\mathrm{H}_0 : (\alpha\beta)_{11} = \cdots = (\alpha\beta)_{ab} = 0$

Using these split units tends to mean

- Tests of main plot effects

- Tests of sub plot effects

5.2 Analysis in SAS

To correctly analyze the data in SAS (or any software), we have to identify the model terms correctly. Consider the following (very small) example:

	Replication	(sub	unit	level)	
1	2	3	4	5	6
A1 (R1, wp)	A2 (R1, wp)	A1 (R2, wp)	A1 (R3, wp)	A2 (R2,wp)	A2 (R3, wp)
B2 (5.22)	B1 (6.52)	B1 (6.13)	B1 (5.77)	B2 (5.81)	B2 (5.49)
B1 (5.61)	B2 (5.78)	B2 (6.14)	B2 (6.23)	B1 (6.43)	B1 (4.60)

What can we tell about the experiment by looking at this layout?

We'll need to identify the whole plot experimental unit for SAS. Let's first use `proc mixed`, just so we can see the expected mean squares and make sure we've correctly identified the whole plot experimental unit.

```
proc mixed method=type3;
  class rep A B;
  model y=A B A*B;
  random rep(A);
run;
```

Type 3 Analysis of Variance

Source	DF	Sum of Squares	Mean Square	Expected Mean Square	Error Term	Error DF	F Value	Pr > F
A	1	0.935208	0.935208	Var(Residual) + 2 Var(rep(A)) + Q(A,A*B)	MS(rep(A))	4	2.98	0.1592
B	1	0.000675	0.000675	Var(Residual) + Q(B,A*B)	MS(Residual)	4	0.00	0.9600
A*B	1	0.095408	0.095408	Var(Residual) + Q(A*B)	MS(Residual)	4	0.40	0.5602
rep(A)	4	1.254133	0.313533	Var(Residual) + 2 Var(rep(A))	MS(Residual)	4	1.32	0.3964
Residual	4	0.947867	0.236967	Var(Residual)	.	.	.	.

Covariance Parameter Estimates

Cov Parm	Estimate
rep(A)	0.03828
Residual	0.2370

Fit Statistics

-2 Res Log Likelihood	16.7
AIC (Smaller is Better)	20.7
AICC (Smaller is Better)	23.1
BIC (Smaller is Better)	20.3

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
A	1	4	2.98	0.1592
B	1	4	0.00	0.9600
A*B	1	4	0.40	0.5602

Figure 5.1: SAS output from PROC MIXED for the analysis of a split-plot

So the F value for Factor A is $F = 2.98$. This comes from

For the A $\times$ B interaction, the F value is $F = 0.40$. This comes from

Example: A researcher is studying the absorption times of a particular type of antibiotic capsule. There are three dosage strengths and 4 capsule wall thicknesses, so 12 total treatment combinations. The researcher has decided on four replicates and it is necessary to run each replicate on a different day, so the researcher plans to block on day. On each day, each dosage strength is formulated. Once a particular dosage strength is formulated, all four wall thicknesses are tested at that strength. Then another dosage strength is selected, and all four wall thicknesses are tested. Finally, the third dosage strength and the four wall thicknesses are tested.

The experimental layout looks like this (everything would be appropriately randomized):

	Day 1			Day 2			Day 3	
Dose1	Dose2	Dose3	Dose1	Dose2	Dose3	Dose1	Dose2	Dose3
Wall1	Wall1	Wall1	Wall1	Wall1	Wall1	Wall1	Wall1	Wall1
Wall2	Wall2	Wall2	Wall2	Wall2	Wall2	Wall2	Wall2	Wall2
Wall3	Wall3	Wall3	Wall3	Wall3	Wall3	Wall3	Wall3	Wall3
Wall4	Wall4	Wall4	Wall4	Wall4	Wall4	Wall4	Wall4	Wall4

- Block:
- Main plot treatment factor:
- Main plot experimental unit:
- Subplot treatment factor:
- Subplot experimental unit:

So, we'll put

in the random statement to identify main plot error.

Again, we'll use `proc mixed`, just so we can see the expected mean squares and make sure we've correctly identified the whole plot experimental unit.

```
proc mixed data=antibiotic method=type3;
  class day dosage wall;
  model time=day dosage wall dosage*wall;
  random day*dosage;
run;
```

Type 3 Analysis of Variance

Source	DF	Sum of Squares	Mean Square	Expected Mean Square	Error Term	Error DF	F Value	Pr > F
day	3	1.895833	0.631944	Var(Residual) + 4 Var(day*dosage) + Q(day)	MS(day*dosage)	6	0.05	0.9833
dosage	2	7810.166667	3905.083333	Var(Residual) + 4 Var(day*dosage) + Q(dosage,dosage*wall)	MS(day*dosage)	6	315.92	<.0001
wall	3	1096.729167	365.576389	Var(Residual) + Q(wall,dosage*wall)	MS(Residual)	27	48.94	<.0001
dosage*wall	6	196.333333	32.722222	Var(Residual) + Q(dosage*wall)	MS(Residual)	27	4.38	0.0033
day*dosage	6	74.166667	12.361111	Var(Residual) + 4 Var(day*dosage)	MS(Residual)	27	1.65	0.1707
Residual	27	201.687500	7.469907	Var(Residual)	.	.	.	.

Covariance Parameter Estimates

Cov Parm	Estimate
day*dosage	1.2228
Residual	7.4699

Fit Statistics

-2 Res Log Likelihood	185.7
AIC (Smaller is Better)	189.7
AICC (Smaller is Better)	190.1
BIC (Smaller is Better)	190.7

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
day	3	6	0.05	0.9833
dosage	2	6	315.92	<.0001
wall	3	27	48.94	<.0001
dosage*wall	6	27	4.38	0.0033

Figure 5.2: PROC MIXED analysis of the antibiotic capsule data

So the F value for dosage strength is $F = 315.92$. This comes from

For the wall thickness effect, the F value is $F = 48.94$. This comes from

We could also consider day as a random effect. We'll go back to `proc glimmix`, since we no longer need the expected mean squares:

```
proc glimmix data=antibiotic;
  class day dosage wall;
  model time=dosage wall dosage*wall;
  random day day*dosage;
run;
```

Here's part of the output

```
Estimated G matrix is not positive definite.

    Covariance Parameter Estimates
                            Standard
  Cov Parm      Estimate       Error

  day                  0           .
  day*dosage      0.2454      1.1182
  Residual        7.4699      2.0330

        Type III Tests of Fixed Effects
                  Num      Den
  Effect           DF       DF    F Value    Pr > F

  dosage            2        6     462.06    <.0001
  wall              3       27      48.94    <.0001
  dosage*wall       6       27       4.38    0.0033
```

These don't match what we had with fixed days, and weird things are happening! What do you notice?

Let's try something else

```
proc glimmix data=antibiotic nobound;
  class day dosage wall;
  model time=dosage wall dosage*wall;
  random day day*dosage;
run;
```

```
Estimated G matrix is not positive definite.

      Covariance Parameter Estimates
                              Standard
 Cov Parm      Estimate       Error

 day            -0.9774      0.5963
 day*dosage      1.2228      1.8552
 Residual        7.4699      2.0331

         Type III Tests of Fixed Effects
                  Num      Den
 Effect            DF       DF    F Value    Pr > F

 dosage             2        6     315.92    <.0001
 wall               3       27      48.94    <.0001
 dosage*wall        6       27       4.38    0.0033
```

What do you notice now?

Let's treat blocks as fixed, just for the sake of demonstrating how we should be estimating treatment means and differences in a split-plot experiment.

The model with blocks:

The main plot treatment mean is $\mu_{i\cdot}$, and the natural estimator is $\overline{y}_{i\cdot\cdot} = \frac{1}{rb}\sum_{k=1}^{b}\sum_{j=1}^{r} y_{ijk}$. For the antibiotic data with dosage strength 1, we get

$$\begin{aligned}\overline{y}_{i\cdot\cdot} &= \frac{1}{(4)(4)}[(95+104+101+108)+(95+106+103+109)\\ &\quad +(96+105+106+113)+(90+100+102+114)]\\ &= \frac{1}{16}(408+413+420+406) = 102.9375\end{aligned}$$

The hardest part is obtaining the estimate of the variance of the mean. First, keep in mind

-
-

To figure out the variance of $\overline{y}_{i\cdot\cdot}$, let's think about what it's actually estimating (assuming balanced data and fixed blocks):

$$\overline{y}_{i\cdot\cdot} = \frac{1}{rb}\sum_{k=1}^{b}\sum_{j=1}^{r} y_{ijk} = \frac{1}{rb}\sum_{k=1}^{b}\sum_{j=1}^{r}(\mu_{ij} + R_j + w_{ij} + e_{ijk})$$

So,

Let's check this in SAS

dosage	Estimate	Standard Error	DF	t Value	Pr > |t|
1	102.94	0.8790	6	117.11	<.0001

What about differences between two whole plot treatment means? We estimate $\mu_{i\cdot} - \mu_{i'\cdot}$ with $\overline{y}_{i\cdot\cdot} - \overline{y}_{i'\cdot\cdot}$, and estimate the variance of the difference with

Checking in SAS:

Differences of dosage Least Squares Means

dosage	_dosage	Estimate	Standard Error	DF	t Value	Pr > |t|
1	2	21.1250	1.2430	6	16.99	<.0001
1	3	-9.3750	1.2430	6	-7.54	0.0003
2	3	-30.5000	1.2430	6	-24.54	<.0001

Estimating the variance of the difference between simple effects gets even more complicated, so we'll just let SAS do it. But, we do need to be aware that depending on the form of the standard error, the default degrees of freedom calculated by SAS may or may not be correct. Any time the standard error cannot be estimated by a single mean square (for many simple effects), the degrees of freedom must be estimated. We can do this by asking SAS to use the Kenward-Roger approximation to the degrees of freedom.

```
proc glimmix data=antibiotic;
  class day dosage wall;
  model time=day dosage wall dosage*wall/ddfm=kr;
  random day*dosage;
  lsmeans dosage/pdiff;
  lsmeans dosage*wall/pdiff;
run;
```

This gives (in part)

Differences of dosage Least Squares Means

dosage	_dosage	Estimate	Standard Error	DF	t Value	Pr > |t|
1	2	21.1250	1.2430	6	16.99	<.0001
1	3	-9.3750	1.2430	6	-7.54	0.0003
2	3	-30.5000	1.2430	6	-24.54	<.0001

dosage*wall Least Squares Means

dosage	wall	Estimate	Standard Error	DF	t Value	Pr > |t|
1	1	94.0000	1.4742	27.44	63.76	<.0001
1	2	103.75	1.4742	27.44	70.38	<.0001
1	3	103.00	1.4742	27.44	69.87	<.0001
1	4	111.00	1.4742	27.44	75.30	<.0001
2	1	71.7500	1.4742	27.44	48.67	<.0001
2	2	82.7500	1.4742	27.44	56.13	<.0001
2	3	86.0000	1.4742	27.44	58.34	<.0001
2	4	86.7500	1.4742	27.44	58.85	<.0001
3	1	108.50	1.4742	27.44	73.60	<.0001
3	2	110.50	1.4742	27.44	74.96	<.0001
3	3	115.00	1.4742	27.44	78.01	<.0001
3	4	115.25	1.4742	27.44	78.18	<.0001

Differences of dosage*wall Least Squares Means

dosage	wall	_dosage	_wall	Estimate	Standard Error	DF	t Value	Pr > |t|
1	1	1	2	-9.7500	1.9326	27	-5.05	<.0001
1	1	1	3	-9.0000	1.9326	27	-4.66	<.0001
1	1	1	4	-17.0000	1.9326	27	-8.80	<.0001
1	1	2	1	22.2500	2.0848	27.44	10.67	<.0001
1	1	2	2	11.2500	2.0848	27.44	5.40	<.0001
1	1	2	3	8.0000	2.0848	27.44	3.84	0.0007
1	1	2	4	7.2500	2.0848	27.44	3.48	0.0017
1	1	3	1	-14.5000	2.0848	27.44	-6.96	<.0001
1	1	3	2	-16.5000	2.0848	27.44	-7.91	<.0001
1	1	3	3	-21.0000	2.0848	27.44	-10.07	<.0001
1	1	3	4	-21.2500	2.0848	27.44	-10.19	<.0001
1	2	1	3	0.7500	1.9326	27	0.39	0.7010
1	2	1	4	-7.2500	1.9326	27	-3.75	0.0009
1	2	2	1	32.0000	2.0848	27.44	15.35	<.0001
1	2	2	2	21.0000	2.0848	27.44	10.07	<.0001
1	2	2	3	17.7500	2.0848	27.44	8.51	<.0001
1	2	2	4	17.0000	2.0848	27.44	8.15	<.0001
1	2	3	1	-4.7500	2.0848	27.44	-2.28	0.0307
1	2	3	2	-6.7500	2.0848	27.44	-3.24	0.0031

5.3 More Complicated Split-Plot Designs

The concept of split-plot designs can be extended to

- Split-split-plot designs (or more splits). Sub-plot units are themselves further divided into sub-sub-plot units.

- Split block/Strip-split plot designs. These lead to 3 different experimental units: for A, B, and A $\times$ B.

- Repeated measures designs. One of the factors is time, and measurements are taken repeatedly on the same experimental unit.